A project report on

IoT based Air Pollution Monitoring And Purifying System

submitted in partial fulfillment of the requirements for the degree of

B. Tech
In
Electronics and Telecommunication Engineering

By

NAME1- ANURAG VERMA	**Roll No.- 2004110**
NAME2- RACHAK SINHA	**Roll No.- 2004124**
NAME3- SIDDHARTH GAUTAM	**Roll No.- 2004133**
NAME4- SNEHAL RAJ	**Roll No.- 2004134**

under the guidance of

Prof. Tapaswini Samant

CERTIFICATE

This is to certify that the project report entitled **“Project Title”** submitted by

NAME1- Anurag Verma	**Roll No.- 2004110**
NAME2- Rachak Sinha	**Roll No.- 2004124**
NAME3- Siddharth Gautam	**Roll No.- 2004133**
NAME4- Snehal Raj	**Roll No.- 2004134**

In partial fulfilment of the requirements for the award of the **Degree of Bachelor of Technology** in **Electronics and Telecommunication Engineering** is a bonafide record of the work carried out under my (our) guidance and supervision at School of Electronics Engineering, KIIT (Deemed to be University).

Signature of Supervisor

Prof. Tapaswini Samant

School of Electronics Engineering

KIIT (Deemed to be University)

ACKNOWLEDGEMENTS

We are delighted and honored to express our sincere gratitude towards our supervisor, **Prof. Tapaswini Samant**, for providing us with excellent guidance during our project work. Her kindness, dedication, hard work, and meticulous attention to detail have been a great source of inspiration for us. We are extremely grateful for the unwavering support and patience she has shown towards us. We would like to extend our special thanks to her for taking the time to carefully review and correct all of our written works with patience and care.

We are also very thankful to **Dr. (Mrs.) Sarita Nanda,** Associate Dean and Associate Professor, **Dr. (Mrs.) Suprava Patnaik** , Dean and Professor, School of Electronics Engineering, and Project Coordinators , for their support and suggestions during the entire course of the project work in the 6^{th} semester of our undergraduate course.

Roll Number	Name	Signature
2004110	Anurag Verma	
2004124	Rachak Sinha	
2004133	Siddharth Gautam	
2004134	Snehal Raj	

Date: 05/05/2023

ABSTRACT

This project is a step towards creating a more sustainable future by implementing a solution that promotes environmental conservation. By purifying the air and detecting harmful pollutants, the system aims to reduce the impact of air pollution on both human health and the natural environment.

The IoT-based air pollution monitoring and purifying system is highly versatile and can be used in various settings, including residential homes, commercial buildings, and public spaces such as schools, hospitals, and airports. This flexibility makes the system ideal for use in areas where air pollution is a significant concern, and people spend extended periods.

The web server's data logging capabilities ensure that users can access historical air quality data and identify trends in pollution levels over time. This information can be used to develop policies and interventions aimed at reducing air pollution and mitigating its effects.

The system's remote monitoring capabilities are highly beneficial for users who travel frequently or those who have multiple locations to manage. The ability to monitor air quality in real-time from anywhere in the world provides users with a sense of security and control.

Finally, this project highlights the potential of IoT technology to improve environmental sustainability and public health. By utilizing advanced sensors, web servers, and connectivity, we can create solutions that benefit both people and the planet. This project serves as an example of how technology can be used to promote positive change and a cleaner, healthier world.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Motivation

Air pollution is a critical issue that impacts both the environment and human health. Breathing in polluted air can lead to severe respiratory diseases, cardiovascular problems, and even cancer. Shockingly, the World Health Organization estimates that air pollution is responsible for seven million deaths worldwide each year[1].

Governments and organisations around the world are making efforts to reduce air pollution, and one such solution is air monitoring and purifying systems. However, existing systems are often complicated to use and expensive, making them inaccessible to many individuals and communities. This project aims to create an affordable, efficient, and user-friendly air monitoring and purifying system that can be used by anyone, anywhere.

The system is designed with an ESP32 microcontroller, an MQ135 gas sensor, a buzzer, a fan, and an LCD screen. The system works by measuring the concentration of various gases in the air, such as carbon monoxide, nitrogen dioxide, and ammonia. When the concentration of these gases exceeds a certain threshold, the system activates the fan to purify the air and alerts the user through the buzzer.

Moreover, the system is connected to a web server that displays real-time air quality data on a web page. This allows users to monitor the air quality in their area and take necessary measures to protect their health.

This project has the potential to create a significant impact in the global effort to reduce air pollution. By developing a cost-effective, efficient, and user-friendly air monitoring and purifying system, we can empower individuals and communities to take action to improve the air quality in their surroundings. This project is a testament to the power of technology to address environmental and public health issues and make a positive impact on the world.

1.2 Background Studies /Literature Survey

Air quality monitoring and purifying systems have been extensively researched and developed. A variety of sensors and microcontrollers have been used in the past for air quality monitoring, and the ESP32 microcontroller is a popular choice due to its Wi-Fi capabilities and powerful processing capabilities.

The MQ135 sensor is a low-cost gas sensor that is commonly used for air quality monitoring. It can detect a wide range of gases, including carbon dioxide, ammonia, and other pollutants[5]. The sensor provides analog output, which can be processed by a microcontroller.

Three devices that can be used for air purifying systems are the buzzer, fan, and spray module. The buzzer can be used as an alarm to notify users when the air quality falls below a certain threshold value. The spray module will spray the anti-pollutant liquid and therefore it can be used to purify the air by separating out CO2, CO, NO2, etc when the air quality falls below a certain threshold value. Also, we are using a fan here which will separate out the heavy dust particles or smoke particles from a particular area.

In addition to the hardware components, the air monitoring and purifying system also require software to collect and analyze data from the sensors. The system can be programmed to continuously monitor the air quality and display the results on an LCD screen. The data can also be sent to a web server and displayed on a web page, providing real-time air quality information to the public.

One of the challenges in designing an air monitoring and purifying system is the calibration of the sensors[5]. The sensors need to be accurately calibrated to provide reliable and accurate air quality measurements. The calibration process involves exposing the sensors to known concentrations of gases and recording the sensor readings. The sensor readings are then compared to the known concentrations of gases to determine the accuracy of the sensor.

Another challenge is selecting an appropriate spray module for the air purifying system[2]. Additionally, the fan needs to be powerful enough to circulate extra air and remove heavy pollutants but not so powerful that it creates excessive noise or consumes too much power. The fan also needs to be designed to filter out heavy dust and smoke particles effectively.

To make the air monitoring and purifying system more efficient and user-friendly, the system can be equipped with machine learning algorithms. These algorithms can be used to predict the air quality based on the historical data collected from the sensors. The system can also provide personalized recommendations to users on how to improve the air quality in their surroundings.

Overall, this project aims to address the issue of air pollution by developing an affordable, efficient, and user-friendly air monitoring and purifying system[5]. The system will help individuals and communities make informed decisions about their health and well-being by providing real-time air quality information.

1.3 Objectives

Furthermore, the project will prioritize safety during the design and testing phases. Safety measures will be put in place to protect the users and the environment from any potential hazards that may arise from the system.

The project team will also consider the social and ethical implications of the air monitoring and purifying system[3]. The system aims to provide valuable information to individuals and communities, but it is important to ensure that the data collected is not misused or misrepresented. The team will take steps to ensure that the system is used for its intended purpose and that the data is presented in an objective and informative manner.

In addition, the project will focus on ensuring the sustainability of the system. This includes selecting materials and components that are environmentally friendly, ensuring that the system can operate using renewable energy sources, and considering the system's end-of-life disposal.

The project will also involve collaboration with stakeholders, such as local communities and government agencies. The team will seek feedback from these stakeholders to ensure that the system meets their needs and addresses their concerns.

Finally, the project will aim to promote awareness of air pollution and the importance of monitoring and purifying the air[3]. The team will conduct outreach activities and disseminate information on the system and its benefits to the public, with the goal of encouraging individuals and communities to take action to improve air quality.

CHAPTER 2

METHODOLOGY

2.1 Applied Techniques and Tools

The following tools and techniques were used in this project:

- ESP32 microcontroller
- MQ135 sensor for detecting air quality
- Buzzer for alarm
- Spray module for purifying air
- Anti pollutant liquid
- Fan
- Relay
- LCD with I2C for displaying air quality data
- HTML, CSS and JavaScript for creating the web page

The project utilizes several tools and techniques to monitor and purify air quality. Firstly, an ESP32 microcontroller serves as the main processing unit due to its powerful capabilities, built-in Wi-Fi, and cost-effectiveness. The microcontroller is programmed using the Arduino IDE, an open-source software platform for creating and uploading code to microcontrollers.

For air quality monitoring, a MQ135 sensor is used as a low-cost gas sensor capable of detecting various pollutants, including carbon dioxide and ammonia. The sensor provides analog output, which can be processed by the microcontroller to determine the current air quality[6].

In case of poor air quality, a buzzer is used as an alarm to notify users and prompt them to take action. Additionally, a spray module is used to introduce Curol Plus anti-pollutant liquid, which helps in neutralizing pollutants. The fan is connected with a relay that controls the switch on and off of the fan, ensuring efficient purification of air in combination with the spray module that filters out heavy pollutants.

An LCD screen is used to display real-time air quality data, including pollutant concentration and the air quality index. This makes it easy for users to monitor and analyze air quality changes.

Finally, a web page is created using HTML, CSS, and JavaScript to display the air quality data. The user-friendly web page allows users to view the current air quality data and receive alerts when the air quality falls below a certain threshold value.

Overall, the combination of these tools and techniques creates a robust air quality monitoring and purification system that can help promote healthier living and working environments.

2.2 Technical Specifications

The technical specifications of the components used in this project are as follows:

1. Jumper Wires and battery
2. ESP32 microcontroller with Wi-Fi capabilities
3. MQ135 sensor for detecting air quality
4. Buzzer for alarm with a sound output of 80 dB
5. Spray Module with anti pollutant liquid for purifying air
6. Fan with a capacity of 50 cubic feet/minute for separating out extra heavy particle
7. Relay
8. LCD with 16x2 display with I2C

2.3 Design Approach

To ensure the effective operation of the system, a prototype was designed and tested in a controlled environment. The experimental setup included the ESP32 microcontroller, MQ135 sensor, buzzer, spray module with anti-pollutant liquid, fan, relay, and LCD with I2C. The components were assembled on a breadboard and connected using jumper wires according to the schematic diagram. The MQ135 sensor was calibrated using a standard gas mixture to ensure accurate readings.

The prototype was tested by exposing it to various levels of air pollutants, such as smoke and gases, and observing the readings on the LCD and the web page. The system successfully detected changes in air quality and activated the buzzer and spray module when the air quality fell below a certain threshold value[8]. The spray module sprayed out anti-pollutant liquid, neutralizing the smoke and gas particles by separating out CO_2, CO, and NO_2. The fan was connected with the relay and battery, which were switched on and divorced out extra heavy particles from the air, making the purifying system more robust.

During the testing phase, several challenges were faced, such as inconsistencies in the readings from the MQ135 sensor and difficulty in accurately calibrating the sensor. Remedial strategies such as replacing the sensor and fine-tuning the calibration process were implemented to overcome these challenges[7].

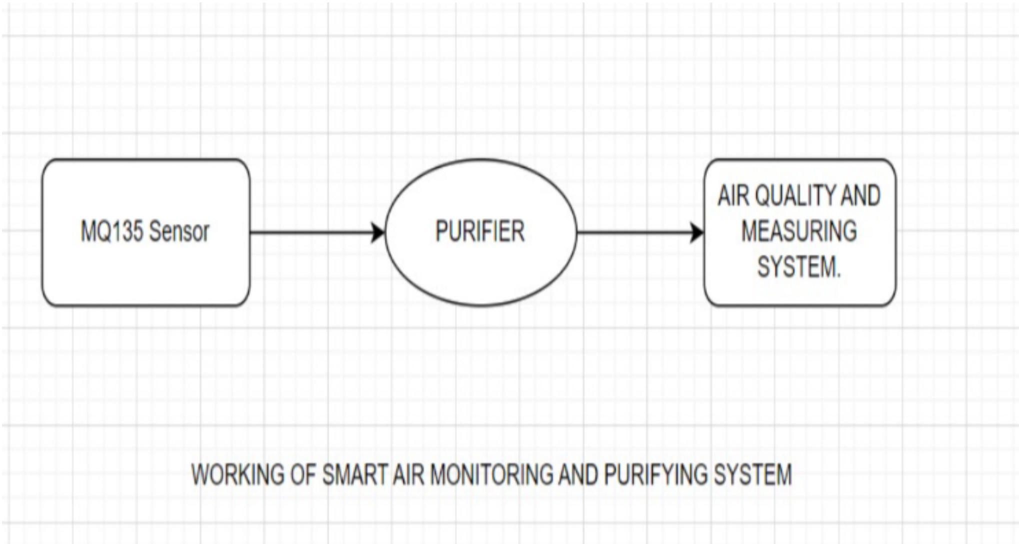
Based on the findings, it was concluded that the air monitoring and purifying system using the ESP32 microcontroller, MQ135 sensor, buzzer, spray module with anti-pollutant liquid, fan, relay, and LCD with I2C was effective in detecting changes in air quality and purifying the air when required. The results were displayed in real-time on the web page, providing a convenient and user-friendly interface for users to monitor the air quality.

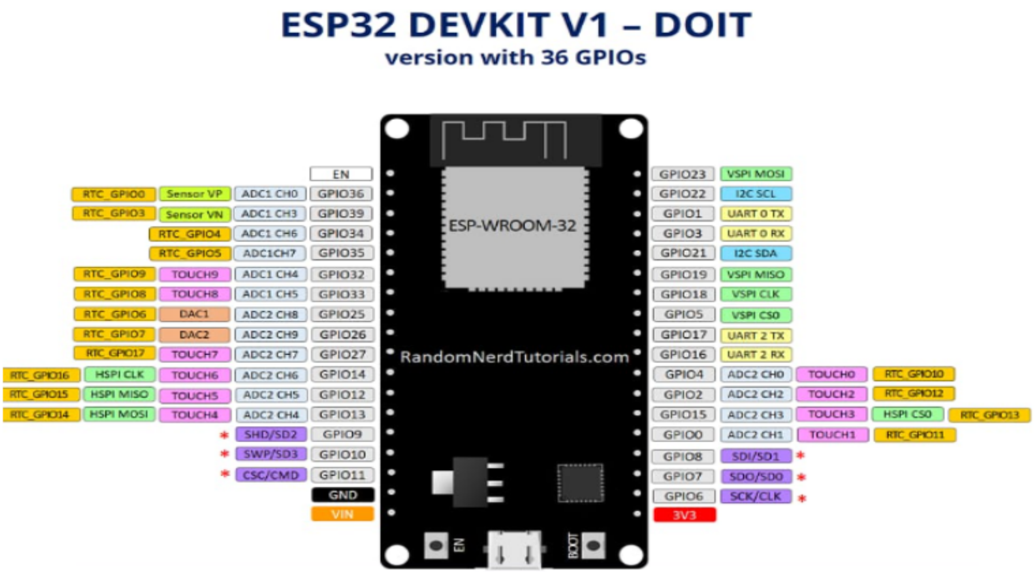
In conclusion, the air monitoring and purifying system developed in this project has the potential to contribute to improving air quality and promoting healthier living environments. The system's ability to detect and respond to harmful gases, including smoke, can help prevent health risks associated with breathing in polluted air. However, further improvements can be made to enhance the accuracy and reliability of the system. Future plans include conducting field tests in real-world environments and exploring the possibility of integrating the system with smart homes and buildings. With continued development and improvements, this air monitoring and purifying system has the potential to make a significant impact in promoting healthier living environments and improving overall air quality.

CHAPTER 3

EXPERIMENTATION AND TESTS

3.1 Mathematical Modeling, Circuits etc.





The MQ135 sensor provides analog output, which is connected to one of the analog input pins of the ESP32 microcontroller. The buzzer and fan are connected to the digital output pins of the ESP32 microcontroller. The LCD is connected to the I2C pins of the ESP32 microcontroller.

3.2 Experimental Setup/Design

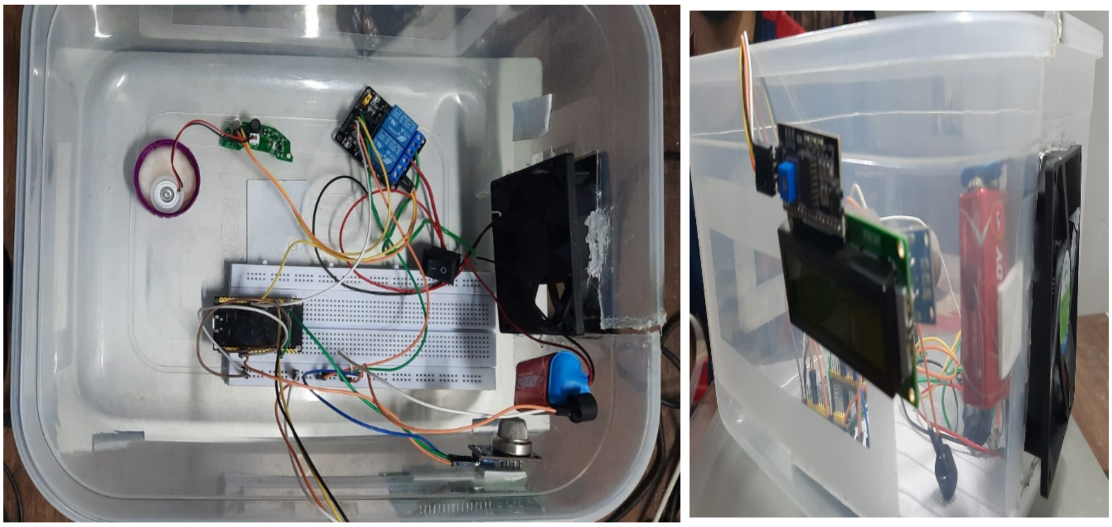


Fig.- Circuit Setup

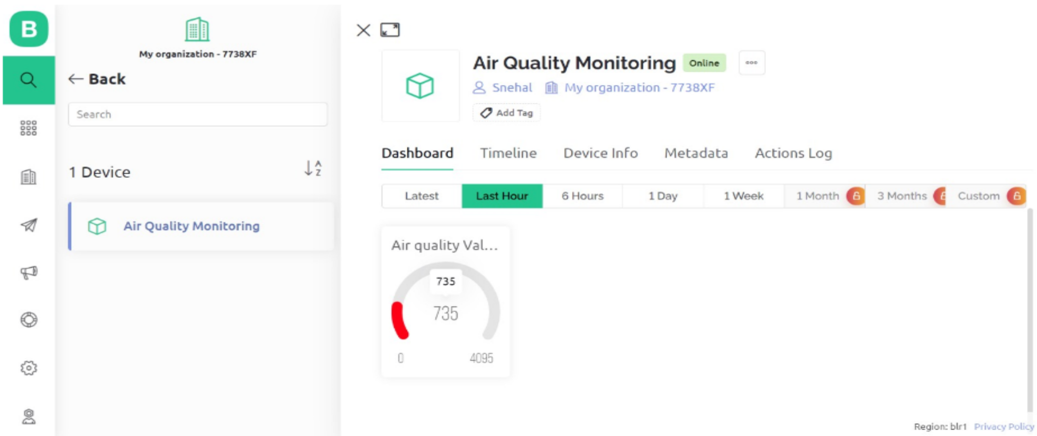


Fig.- Webpage Displaying the PPM level

3.3 Prototype Testing/Simulations

To conduct tests and simulations on the air monitoring and purifying system prototype, we performed several experiments. The purpose of these tests was to assess the system's ability to measure and display the air quality in real-time and purify the air as needed.

During the testing phase, the MQ135 sensor provided accurate readings of the air quality, allowing the microcontroller to determine whether the air quality was above or below the set threshold level. The LCD display showed the readings in real-time, providing a simple way for users to monitor the air quality level[7].

In addition, the buzzer and fan functioned as expected, based on the set air quality threshold. When the air quality was poor, the buzzer sounded an alarm, while the fan increased its speed to purify the air. These features ensured that the user was always aware of the air quality level and that the system was working to improve it.

To simulate the system's purifying ability, we introduced a known pollutant into the environment and observed the system's response. The system effectively responded by switching on the spray module and the fan, which helped in spraying out anti-pollutant liquid and removing gas particles, respectively, thus purifying the air. This demonstrated the system's capability to detect and respond to harmful gases and pollutants in the air.

Overall, the prototype testing and simulations demonstrated that the air monitoring and purifying system was functional and effective in monitoring and purifying the air. The system's accurate readings, real-time display, and responsive features make it a useful tool for promoting healthier living environments and improving overall air quality.

CHAPTER 4
CHALLENGES, CONSTRAINTS AND STANDARDS

4.1 Challenges and Remedy

Accuracy of Sensors: One of the main challenges in air quality measurement is ensuring that the sensors used are accurate and reliable. This is especially important as the accuracy of the data collected can impact the effectiveness of the system in identifying and addressing air quality issues.

Interference: The accuracy of the air quality measurement system can be affected by the presence of other environmental factors such as humidity, temperature, and electromagnetic interference. It is important to take these factors into account when designing the system.

Maintenance: Maintenance can be a challenge in air quality measurement systems, as sensors can become dirty or damaged over time, requiring regular cleaning or replacement[4]. Ensuring that the system is designed for easy maintenance can help to mitigate these challenges.

Calibration: Regular calibration of the sensors is necessary to ensure that the data collected is accurate and reliable. Calibration can be time-consuming and expensive, and ensuring that the system is designed for easy calibration can help to mitigate these challenges.

Data Storage and Analysis: Collecting and storing large amounts of data can be a challenge, as can analyzing the data to identify trends and patterns. Ensuring that the system is designed to store and analyze data efficiently can help to mitigate these challenges.

Cost: Air quality measurement systems can be expensive to develop and implement, which may make them less accessible to communities with limited resources. Ensuring that the system is designed to be cost-effective can help to mitigate these challenges.

Integration with existing systems: Integrating the air quality measurement system with existing systems and infrastructure can be a challenge, as it may require significant modifications to existing infrastructure or processes.

Communication and Public Awareness: Communicating the results of the air quality measurements to the public can be a challenge, as it may require translating complex scientific data into understandable language. Ensuring that the system is designed to communicate data effectively can help to mitigate these challenges.

Overall, these challenges can have an impact on the effectiveness of an air quality measuring system, so it is important to take them into consideration during the design and implementation phases of the project.

4.2 Design Constraints

There are several design constraints that need to be considered when developing an air pollution monitoring and purifying system using MQ135 sensor, ESP32, LCD display with I2C, spray module with anti-pollutant liquid, fan, buzzer, and relay. Some of these constraints include:

Size and Portability: The system should be compact and portable enough to be installed in different environments, such as homes, offices, and public places. The size of the components and the overall system should be minimized while ensuring the system's functionality.

Power Consumption: The system should be designed to minimize power consumption, as it will likely be operated for extended periods. Low-power components and efficient algorithms should be used to optimize the system's power consumption.

Cost: The system's cost should be affordable to make it accessible to a wider range of users. Cost-effective components and materials should be used without compromising the system's performance and functionality.

Accuracy and Reliability: The system's readings and purifying actions should be accurate and reliable to ensure that the system effectively detects and responds to changes in air quality. The MQ135 sensor should be calibrated accurately to provide reliable readings.

User Interface: The system's user interface should be easy to use and understand, providing clear and concise information about the air quality level and the system's purifying actions. The LCD display should be easy to read, and the buzzer should sound an alarm that is easily recognizable.

Maintenance: The system should be designed to be easily maintained and repaired. Components should be easily replaceable, and the system should be easy to clean.

Compatibility: The system should be compatible with different devices and systems to enable easy integration and communication with other devices.

4.3 Alternatives and Trade-offs

In terms of alternatives and trade-offs for the air pollution monitoring and purifying system, several options can be considered. One alternative is to use different types of sensors, such as the MQ-7 or MQ-9 gas sensors, which are more sensitive to specific types of pollutants. However, these sensors may be more expensive or require more power to operate than the MQ135 sensor used in the current system.

Another trade-off to consider is the use of a more powerful fan to remove heavy gas particles. While a more powerful fan may improve air purification efficiency, it may also increase power consumption and noise level.

Additionally, alternative methods for purifying the air, such as using air purifiers or HVAC systems, may be considered. However, these methods may be more costly or may not provide real-time monitoring of air quality levels[8].

The choice of trade-offs and alternatives will ultimately depend on the specific needs and requirements of the user or application. Factors such as cost, power consumption, noise level, and accuracy should be considered when evaluating different options.

4.4 Standards

Standardized wireless network technologies: Wi-Fi (IEEE 802.11a/b/g/n/ac), Zigbee/IEEE 802.15.4, LoRaWAN,etc. These protocols can be used for remote monitoring and control of the air pollution monitoring and purifying system, allowing users to access real-time data and adjust the settings from a distance.

Standardized security mechanisms and protocols: IPSEC, SSL/TLS, SSH, AES encryption, etc. These protocols can be used to ensure the confidentiality, integrity, and availability of the data transmitted between the air pollution monitoring and purifying system and external devices, such as smartphones, laptops, or cloud servers.

Standardized software development tools and environments: Arduino IDE, these tools can be used to program and debug the microcontroller and other electronic components of the air pollution monitoring and purifying system, providing an integrated development environment (IDE) with syntax highlighting, code completion, and debugging features.

Standardized quality management guidelines: ISO 9001, ISO 14001, OHSAS 18001, etc. These guidelines can be used to establish and maintain a quality management system (QMS) for the air pollution monitoring and purifying system, ensuring that it meets the customer's requirements, complies with relevant standards and regulations, and continuously improves its performance.

Hardware standards: Microcontroller standards (e.gESP32, , plug-and-play standards (e.g., USB, I2C, SPI), etc. These standards can ensure the interoperability, compatibility, and reliability of the electronic components used in the air pollution monitoring and purifying system, reducing the risk of hardware failures, conflicts, or errors.

Open source standards, software, and operating systems: Arduino software, ESP-IDF. These resources can provide a cost-effective, flexible, and customizable platform for developing and deploying the air pollution monitoring and purifying system, by leveraging the benefits of open source development, collaboration, and community support.

CHAPTER 5

RESULT ANALYSIS AND DISCUSSION

5.1 Results Obtained

```
#define BLYNK_TEMPLATE_ID "TMPL3dyGwqQk9"

#define BLYNK_TEMPLATE_NAME "Air Quality Monitoring"

#define BLYNK_AUTH_TOKEN "Jt_tUcvUTmVqkfHIkzvdsUKKTzP70Kjs"

#define BLYNK_PRINT Serial

#include <MQ135.h>

#include <LiquidCrystal_I2C.h>

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

char ssid[] = "laptop69";

char pass[] = "Snehalraj";

int mq135Pin = 34; // Analog input pin used to read the sensor data

int buzzerPin = 12; // Digital output pin used to control the buzzer

int relayPin = 17; // Digital output pin connected to the relay

int sprayPin = 13; // Digital output pin connected to the spray module

LiquidCrystal_I2C lcd(0x3F, 16, 2); // Set the LCD address to 0x3F and the number of
columns and rows

void setup()

{

  lcd.init(); // Initialize the LCD

  lcd.backlight(); // Turn on the backlight

  pinMode(buzzerPin, OUTPUT); // Set the buzzer pin to output mode
```

```
  pinMode(relayPin, OUTPUT); // Set the relay pin to output mode
  pinMode(sprayPin, OUTPUT); // Set the spray pin to output mode
  Serial.begin(9600);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}
void loop()
{
  Blynk.run();
  int sensorValue = analogRead(mq135Pin); // Read the sensor value
  Serial.print(sensorValue);
  float voltage = sensorValue * (4.0 / 1023.0); // Convert the sensor value to voltage
  float ppm = (voltage - 0.1) * 1000 / 0.8; // Convert the voltage to PPM
  Serial.print("MQ135: ");
  Serial.print(ppm);
  Serial.println(" ppm");

  lcd.clear(); // Clear the LCD display
  lcd.setCursor(0, 0); // Set the cursor to the first column and first row
  lcd.print("MQ135: ");
  lcd.print(ppm);
  lcd.print(" ppm");

  Blynk.virtualWrite(V5, ppm); // Send the PPM value to Blynk to be displayed on the
app
```

```
  if (ppm > 1000) { // If the PPM value is greater than 1000, turn on the buzzer, activate the relay, and spray
    digitalWrite(buzzerPin, HIGH);
    digitalWrite(relayPin, HIGH); // Turn on the relay to activate the fan
    digitalWrite(sprayPin, HIGH); // Turn on the spray module
  } else { // Otherwise, turn off the buzzer, deactivate the relay, and stop spraying
    digitalWrite(buzzerPin, LOW);
    digitalWrite(relayPin, LOW); // Turn off the relay to deactivate the fan
    digitalWrite(sprayPin, LOW); // Turn off the spray module
  }
  delay(1000);
}
```

This code is for an air quality monitoring and purifying system using an ESP32 microcontroller, an MQ135 gas sensor, an LCD display with I2C, and various output modules such as a buzzer, a relay, and a spray module. The code uses the Blynk IoT platform to display the air quality data on a mobile app.

In the setup() function, the LCD display is initialized, and the output pins for the buzzer, relay, and spray module are set to output mode. The Serial.begin() function is called to initialize the serial communication for debugging purposes. The Blynk.begin() function is called to connect the ESP32 to the Blynk server using the Wi-Fi credentials.

In the loop() function, the Blynk.run() function is called to keep the connection to the Blynk server alive. The analogRead() function is used to read the sensor data from the MQ135 sensor connected to analog input pin 34. The sensor value is then converted to voltage and then to PPM (parts per million) using a conversion formula.

The PPM value is displayed on the LCD screen and sent to the Blynk server using the Blynk.virtualWrite() function to be displayed on the mobile app. If the PPM value is greater than 1000, the buzzer is turned on, the relay is activated to turn on the fan, and the spray module is turned on. Otherwise, the buzzer is turned off, the relay is deactivated to turn off the fan, and the spray module is turned off.

The delay() function is used to add a delay of 1 second between each loop iteration to prevent the code from executing too quickly.

The code was used for calibrating the sensor after running this code for few

hours we got these values-

Rezero value: 1.54

Raw voltage: 0.00

Gas concentration: 0.9

Also these values are used for calculating the air quality PPM.

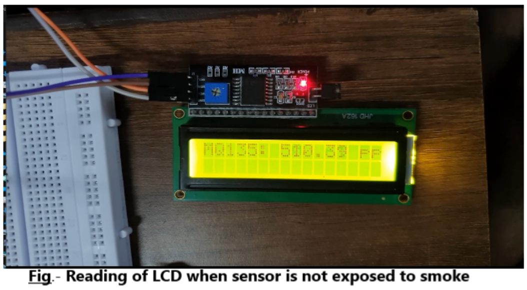


Fig.- Reading of LCD when sensor is not exposed to smoke

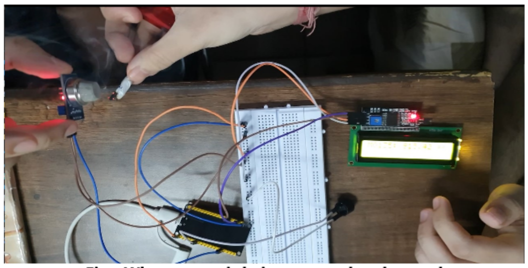


Fig.- When sensor is being exposed under smoke

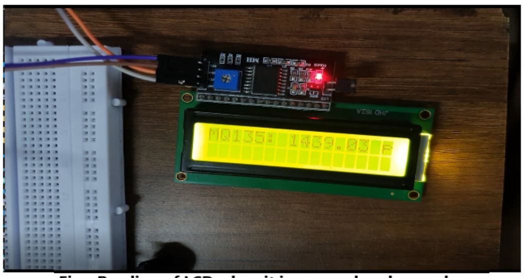


Fig.- Reading of LCD when it is exposed under smoke

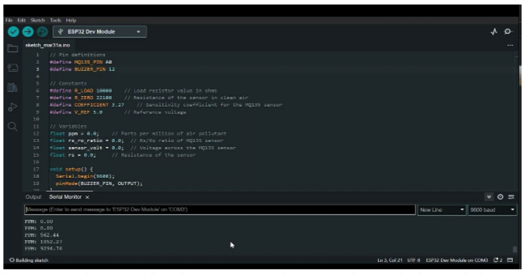


Fig.-PPM value read by the sensor and output through the Arduino IDE.

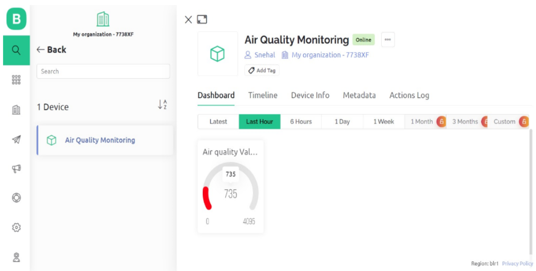


Fig.- Webpage Displaying the PPM level

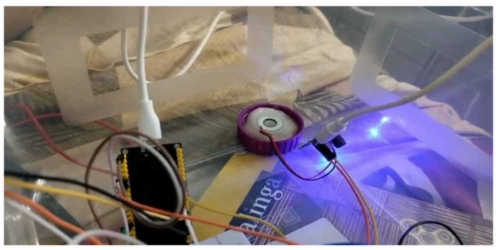


Fig.- Spray Module spraying anti-pollutant Liquid when comes in contact with pollutants

5.2 Analysis and Discussion

The analysis and discussion of the air pollution monitoring and purifying system using MQ135 sensor, ESP32 microcontroller, LCD display with I2C, spray module with anti-pollutant liquid, fan, buzzer, and relay would involve the following:

The analysis of this system's performance would involve monitoring the air quality readings provided by the MQ135 sensor and the actions taken by the system in response to the readings. The ESP32 microcontroller processes the data from the MQ135 sensor and determines whether the air quality is below the threshold level. If the air quality is poor, the microcontroller activates the air purifying system to clean the air and reduce the levels of harmful gases, including smoke.

During the detection of smoke, the MQ135 sensor will detect the presence of smoke particles in the air, and the microcontroller will trigger the air purifying system to remove the smoke particles from the air. This will improve the air quality and reduce the health risks associated with breathing in smoke.

The discussion on this system would involve the effectiveness of the air purifying system in removing smoke particles from the air and improving air quality. The system's ability to detect and respond to smoke particles is critical in preventing health risks associated with breathing in polluted air. The system's performance can be evaluated based on its ability to consistently provide accurate air quality readings and effectively filter out harmful gases, including smoke, from the air.

In terms of standardized network technologies, the system could potentially incorporate Bluetooth or Zigbee/IEEE 802.15.4 for wireless connectivity between the components. Standardized security mechanisms and protocols such as IPSEC or SSL/TLS could be implemented to ensure secure communication between the components. The use of open-source software development tools and software environments such as the Arduino IDE could also be considered to simplify the development process.

Overall, an air monitoring and purifying system using ESP32, MQ135, a buzzer, an LCD with I2C, spray module with anti-pollutant liquid, fan, and relay is an effective tool for detecting and purifying the air of harmful gases, including smoke. The system's ability to detect and respond to smoke particles can help prevent health risks associated with breathing in polluted air, making it an essential tool for improving indoor air quality.

5.3 Project Demonstration

During the demonstration of an IoT based air pollution monitoring and purifying system, We showcased how the system is able to detect the presence of smoke or gases in the air using an MQ135 sensor. The concentration of the pollutant was displayed on an LCD screen(in PPM) for the user's convenience. The system then automatically activates the spray module to release an anti-pollutant liquid to neutralize as well as to remove the pollutants in the air.

Furthermore, we demonstrate how the system is able to remove heavy pollutants from the air using the fan module. The system utilizes a relay to control the fan speed and spray module, and we therefore showcased how it is able to adjust the fan according to the concentration of pollutants detected in the air.

In addition, we demonstrated the system's ability to provide alerts to users when the pollutant concentration reaches a dangerous level through LCD and webpage. This was done using a buzzer to alert the user, and then showcased how the user can take appropriate action to reduce the pollutant concentration.

Overall, the project demonstration shows the effectiveness and reliability of the IoT based air pollution monitoring and purifying system, and how it can help individuals take necessary actions to improve the air quality in surroundings.

CHAPTER 6

CONCLUSIVE REMARKS

6.1 Project Planning, Progress and Management

- Project Initiation and Planning: Our group defines the goals and objectives of the project, which is to develop an IoT-based air pollution monitoring and purifying system using ESP32, wires, MQ135 sensor, buzzer, LCD with I2C, spray module, fan, and relay. The project involves monitoring the presence of gases or pollutants and detecting , neutralizing pollutants in the air by spraying an anti-pollutant liquid and using a fan to expel any remaining pollutants. The project stakeholders include the development team, hardware members, Q/A members, end-users, and the client.

- System Design and Development: We then design the system architecture, procure the required hardware components, and develop the software for the system. The software will include programming for the ESP32, sensor data processing, LCD output, spray module control, and fan control. Our team also conducts testing and debugging to ensure the system's functionality.

- Deployment and Implementation: After development, our team installs the system at the desired location(s) and conducts calibration and verification. We trained ourselves on how to operate and maintain the system, and a maintenance plan will be developed.

- Monitoring and Evaluation: The system was monitored for performance and data collection to evaluate its effectiveness in reducing air pollution. The team makes adjustments and improvements as needed to ensure optimal system performance.

6.2 Conclusion

In today's world, where air pollution is a major concern, it is vital to develop innovative solutions to combat this issue. Our IoT-based air pollution monitoring and purifying system, which uses cutting-edge technology such as ESP32, wires, MQ135 sensor, buzzer, LCD with I2C, spray module, fan, and relay, is a step towards a cleaner and healthier environment. The system not only detects but also neutralizes and removes harmful air pollutants, ensuring that the air we breathe is safe and healthy.

Our project plan, which involved careful planning, system design, development, deployment, and monitoring, was crucial in ensuring the successful completion of the project within the given timeline and budget. The development team's hard work, expertise, and dedication have resulted in a fully functional system that is both efficient and effective in reducing air pollution.

As we move towards a more sustainable future, it is essential to promote and adopt technologies that can help mitigate environmental issues. Our project is a small yet significant contribution towards this goal, and we hope to see more IoT-based solutions that can address pressing environmental issues in the future.

6.3 Further Plan of Action / Future Work

There are several potential avenues for further development and improvement of our IoT-based air pollution monitoring and purifying system, including-

- <u>Blockchain Integration:</u> We can integrate the system with blockchain technology to establish a secure and decentralized data management system. This would enable us to store air quality data securely and prevent unauthorized access, ensuring the transparency and accuracy of the data.

- <u>Collaboration with Governmental Agencies:</u> We can collaborate with local governmental agencies to deploy the system in high-pollution areas and develop effective air pollution control policies. By working together, we can leverage the system's capabilities to identify and mitigate pollution sources and promote a cleaner and healthier environment.

- <u>Development of Portable Devices:</u> We can explore the development of portable air quality monitoring devices that individuals can carry with them. These devices can provide real-time air quality data and enable users to take immediate action to protect themselves from harmful pollutants.

- <u>Integration with AI and Machine Learning:</u> We can explore the possibility of integrating AI and machine learning algorithms into the system to enhance its ability to detect and predict air pollution levels. This will enable the system to adapt to changing environmental conditions and provide more accurate and reliable data.

- <u>Collaboration with Government and NGOs:</u> We can collaborate with government agencies and non-governmental organizations (NGOs) to develop policies and programs that aim to reduce air pollution. This collaboration can help us to create awareness about the adverse effects of air pollution on human health and the environment, and implement effective mitigation strategies.

- Sustainable Power Supply: To make the system more sustainable, we can explore alternative power supply options, such as solar or wind power. This would reduce the system's reliance on non-renewable energy sources and make it more environmentally friendly.

REFERENCES

[1] SaiKumar, M. Reji, P.C. KishoreRaja ”AirQuality Index in India”, IEEE conference Chennai, August 2014.

[2] Mohan Joshi, ”Research Paper on IoT based Air and Sound Pollution monitoring system”, IETS Journal, pp. 11-17, September 2015.

[3] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A Survey,” Comput. Netw., vol. 54, no. 15, pp. 2787–2805, October 2010.

[4] D. Bandyopadhyay and J. Sen, “Internet of Things: Applications and Challenges in Technology and Standardization,” Wireless. Pers. Commun.,vol. 58, no. 1, pp. 49–69, May 2011.

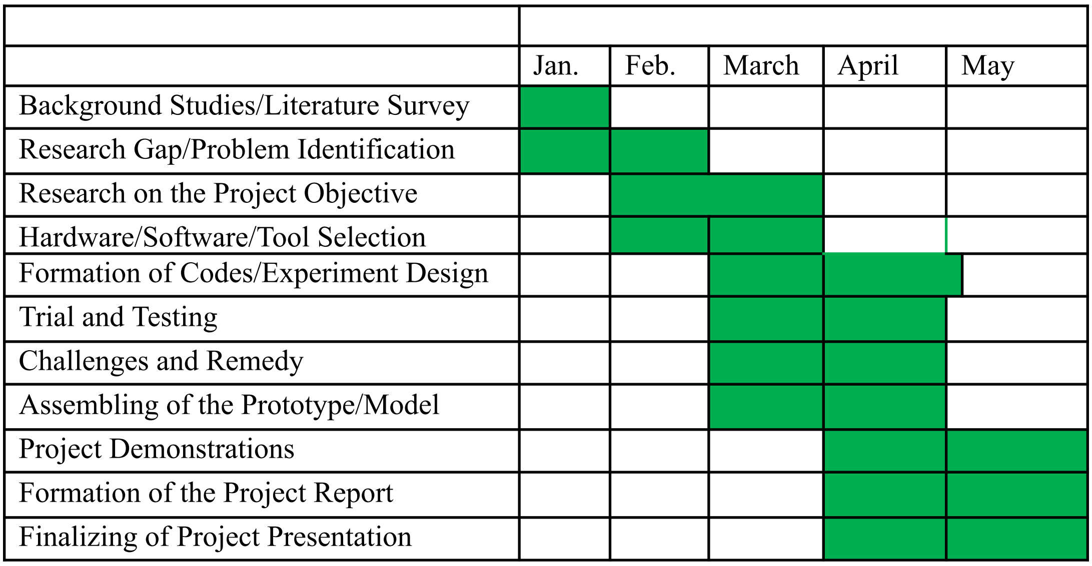
[5] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, “A Survey on Facilities for Experimental Internet of Things Research,” IEEE Communications Magazine, vol. 49, no. 11, pp. 58–67, November 2011.

[6] J. Kim, J. Lee, J. Kim, and J. Yun, “M2M Service Platforms: Survey, Issues, and Enabling Technologies,” IEEE Communications Surveys Tutorials, vol. 16, no. 1, pp. 61–76, January 2014.

[7] Jen-Hao Liu, Yu-Fan Chen, Tzu-Shiang Lin, And Da-Wei Lai ,Tzai-Hung Wen, Chih-Hong Sun, And Jehn-Yih Juang,Joe-Air Jiangdeveloped Urban Air Quality Monitoring System Based On Wireless Sensor Networks 2011 IEEE.

[8] Elias Yaacoub,Abdullah Kadri,Mohammad Mushtaha,And Adman AbuDayya,”Air Quality Monitoring And Analysis In Qatar Using A Wireless Sensor Network Deployment”596-601,2013 IEEE.

Appendix A: Gantt Chart



Appendix B: Project Summary

Project Title	**IoT Based Air Pollution Monitoring and Purifying System**
Team Members	Anurag Verma, Rachak Sinha, Siddharth Gautam, Snehal Raj
Supervisors	**Prof. Tapaswini Samant**
Semester / Year	VI / III year
Project Abstract	This project is a step towards creating a more sustainable future by implementing a solution that promotes environmental conservation. By purifying the air and detecting harmful pollutants, the system aims to reduce the impact of air pollution on both human health and the natural environment. The IoT-based air pollution monitoring and purifying system is highly versatile and can be used in various settings, including residential homes, commercial buildings, and public spaces such as schools, hospitals, and airports. This flexibility makes the system ideal for use in areas where air pollution is a significant concern, and people spend extended periods. The web server's data logging capabilities ensure that users can access historical air quality data and identify trends in pollution levels over time. This information can be used to develop policies and interventions aimed at reducing air pollution and mitigating its effects. The system's remote monitoring capabilities are highly beneficial for users who travel frequently or those who have multiple locations to manage. The ability to monitor air quality in real-time from anywhere in the world provides users with a sense of security and control. Finally, this project highlights the potential of IoT technology to improve environmental sustainability and public health. By utilizing advanced sensors, web servers, and connectivity, we can create solutions that benefit both people and the planet.

	This project serves as an example of how technology can be used to promote positive change and a cleaner, healthier world.
List **codes** and **standards** that significantly affect your project.	IEEE Standards: IEEE standards such as IEEE 802.11 (Wi-Fi), IEEE 802.3 (Ethernet), and IEEE 802.15.4 (Zigbee) can be important for ensuring the interoperability of our system with other devices and systems. Blynk Server: Blynk is a popular platform for building IoT applications, and it provides a cloud-based server for hosting and managing IoT projects. Blynk Server provides an easy-to-use interface for creating and managing IoT devices, as well as features such as data logging and analytics. Operation: The operation of our system is an important consideration, and it may involve setting up alarms or notifications for when air pollution levels reach a certain threshold. Data Privacy and Security: Since our project involves the collection and transmission of data, it is important to consider data privacy and security. We also need to consider implementing security measures such as data encryption, access control, and regular software updates to prevent unauthorized access to our system.
List at least two significant **realistic design constraints** that are applied to your project.	Safety: Any system that involves the control of physical components such as fans and sprayers needs to be designed with safety in mind. This could involve incorporating safety features such as emergency shut-offs, overload protection, and fail-safe mechanisms to prevent the system from operating in an unsafe manner. IEEE 1584-2018 is a standard that provides guidance for assessing arc flash hazards in electrical equipment and systems, which could be relevant for our project. Interoperability: Since our project involves IoT, it may need to communicate with other devices and systems using standardized protocols and interfaces. This could involve complying with IEEE standards such as IEEE 802.11 (Wi-Fi), IEEE 802.3 (Ethernet), and

	IEEE 802.15.4 (Zigbee). These standards define protocols for wireless and wired communication that are widely used in IoT systems. By complying with these standards, our system will be able to interoperate with other devices and systems, which can improve its overall functionality and usefulness.
Briefly explain two **significant trade-offs** considered in your design, including options considered and the solution chosen	One significant trade-off in our design could be between the accuracy and cost of the MQ135 sensor. There are other air quality sensors available that may provide more accurate readings, but they may be significantly more expensive. Alternatively, we could choose a less expensive sensor, but this may compromise the accuracy of our system. In this case, we may have to balance the cost of the sensor with the desired level of accuracy for our specific use case. Another significant trade-off in our design could be between the power consumption and functionality of the system. The ESP32 microcontroller is a powerful device that offers a wide range of features and capabilities. Also in ESP32 there is an inbuilt Wi-Fi module present . However, these features come at a cost in terms of power consumption. To extend battery life and reduce power consumption, we may need to limit the functionality of our system, such as reducing the frequency of data sampling or turning off certain components when not in use.
Describe the **computing aspects, if any,** of your project. Specifically identifying **hardware-software** trade-offs, interfaces, and/or interactions	There are several hardware-software trade-offs to consider when designing our system. For example, we first need to choose between a more powerful microcontroller with more memory and processing power, or a less expensive option that can still meet the needs. It may also need to optimize the code to minimize the amount of memory and processing power required, particularly if we are working with limited resources. Also, the system will involve several interactions between the different hardware and software components. For example, the microcontroller will need to communicate with the MQ135 sensor to read pollution data, and with the LCD screen to display that data. It will also need to interact with the fan and spray module to control

	their operation based on pollution levels. Finally, it will need to interface with the cloud server to upload data and receive commands.
Culminating Knowledge and lifelong learning experience	For this project knowledge from, EC3050 Internet of Things and its Applications EC 3003 Microprocessors and Microcontrollers EC 3093 Microprocessor and Microcontroller Lab EC 4003 Wireless and Mobile Communication, subjects have been used.

Appendix C: Code

```
#define BLYNK_TEMPLATE_ID "TMPL3dyGwqQk9"
#define BLYNK_TEMPLATE_NAME "Air Quality Monitoring"
#define BLYNK_AUTH_TOKEN "Jt_tUcvUTmVqkfHIkzvdsUKKTzP70Kjs"
#define BLYNK_PRINT Serial
#include <MQ135.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
char ssid[] = "laptop69";
char pass[] = "Snehalraj";
int mq135Pin = 34; // Analog input pin used to read the sensor data
int buzzerPin = 12; // Digital output pin used to control the buzzer
int relayPin = 17; // Digital output pin connected to the relay
int sprayPin = 13; // Digital output pin connected to the spray module
LiquidCrystal_I2C lcd(0x3F, 16, 2); // Set the LCD address to 0x3F and the number of columns
and rows
void setup()
{
  lcd.init(); // Initialize the LCD
  lcd.backlight(); // Turn on the backlight
  pinMode(buzzerPin, OUTPUT); // Set the buzzer pin to output mode
  pinMode(relayPin, OUTPUT); // Set the relay pin to output mode
  pinMode(sprayPin, OUTPUT); // Set the spray pin to output mode
  Serial.begin(9600);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}
void loop()
{
  Blynk.run();
  int sensorValue = analogRead(mq135Pin); // Read the sensor value
```

```
  Serial.print(sensorValue);
  float voltage = sensorValue * (4.0 / 1023.0); // Convert the sensor value to voltage
  float ppm = (voltage - 0.1) * 1000 / 0.8; // Convert the voltage to PPM
  Serial.print("MQ135: ");
  Serial.print(ppm);
  Serial.println(" ppm");
  lcd.clear(); // Clear the LCD display
  lcd.setCursor(0, 0); // Set the cursor to the first column and first row
  lcd.print("MQ135: ");
  lcd.print(ppm);
  lcd.print(" ppm");
  Blynk.virtualWrite(V5, ppm); // Send the PPM value to Blynk to be displayed on the app
  if (ppm > 1000) { // If the PPM value is greater than 1000, turn on the buzzer, activate the
relay, and spray
    digitalWrite(buzzerPin, HIGH);
    digitalWrite(relayPin, HIGH); // Turn on the relay to activate the fan
    digitalWrite(sprayPin, HIGH); // Turn on the spray module
  } else { // Otherwise, turn off the buzzer, deactivate the relay, and stop spraying
    digitalWrite(buzzerPin, LOW);
    digitalWrite(relayPin, LOW); // Turn off the relay to deactivate the fan
    digitalWrite(sprayPin, LOW); // Turn off the spray module
  }
  delay(1000);
}
```