# AIR QUALITY MONITORING SYSTEM USING ARDUINO UNO AND MQ135 GAS SENSOR


## ABSTRACT

The Air Quality Monitoring System is a significant innovation in the realm of environmental sensing and data visualization. This project leverages the power of an Arduino UNO microcontroller, an MQ-135 gas sensor, a breadboard, connecting jumper wires, an LED, a buzzer, a 16x2 LCD monitor, and an array of components to create a comprehensive solution for monitoring air quality. Air quality has a profound impact on our health and well-being, making it vital to keep a close eye on its status. This system not only detects air quality but also provides real-time feedback, indicating whether the air quality falls within acceptable levels.


## INTRODUCTION

The Air Quality Monitoring System is designed to continuously assess air quality and present the results in a user-friendly manner. By employing an MQ-135 gas sensor, which measures the concentration of harmful gases in the environment, the system can accurately determine the air quality. The project is built on an Arduino UNO platform and employs a range of components such as a breadboard, connecting jumper wires, an LED, a buzzer, and a 16x2 LCD monitor to create a comprehensive solution.

The system's operation is intuitive and user-friendly. It displays the air quality in a visual and informative way, using a threshold value as a reference. When the air quality is within an acceptable range, it displays "Air Quality Level Good" on the LCD screen. Conversely, if the air quality exceeds the defined threshold, it displays "Air Quality Level Poor," alerting the user to take necessary precautions. Additionally, the system prints the exact air quality value, allowing for precise data logging and analysis.

The threshold value used in this project is determined based on the MQ-135 gas sensor's measurements. It ensures that the system is sensitive enough to detect air quality variations accurately. The Air Quality Monitoring System is an invaluable tool for homeowners, workplaces, and public spaces, enabling everyone to monitor and maintain a safe and healthy environment.

## LITERATURE SURVEY

### (1) IOT Based Air Pollution Monitoring System Using Arduino

Poonam Pal1, Ritik Gupta2, Sanjana Tiwari3, Ashutosh Sharma

The level of pollution has increased with times by lot of factors like the increase in population, increased vehicle use, industrialization and urbanization which results in harmful effects on human wellbeing by directly affecting health of population exposed to it. In order to monitor In this project we are going to make an IOT Based Air Pollution Monitoring System in which we will monitor the Air Quality over a web server using internet and will trigger a alarm when the air quality goes down beyond a certain level, means when there are sufficient amount of harmful gases are present in the air like CO2, smoke, alcohol, benzene and NH3. It will show the air quality in PPM on the LCD and as well as on webpage so that we can monitor it very easily. In this IOT project, you can monitor the pollution level from anywhere using your computer or mobile.

### (2) IOT Based Air Pollution Monitoring System

Harsh N. Shah 1, Zishan Khan 2, Abbas Ali Merchant 3, Moin Moghal 4, Aamir Shaikh 5, Priti Rane 6 1, 2, 3, 4,5Student, Diploma in Computer Engineering, BGIT, Mumbai Central, India 6Assistant Professor, BGIT, Mumbai Central, India

Air pollution is the biggest problem of every nation, whether it is developed or developing. Health problems have been growing at faster rate especially in urban areas of developing countries where industrialization and growing number of vehicles leads to release of lot of gaseous pollutants. Harmful effects of pollution include mild allergic reactions such as irritation of the throat, eyes and nose as well as some serious problems bronchitis, heart diseases, pneumonia, lung and aggravated asthma. According to survey, due to air pollution 50,000 to 100,000 premature deaths per year occur in the U.S. alone. Whereas in EU number reaches to 300,000 and over 3,000,000 worldwide. IOT Based Air Pollution Monitoring System monitors the Air quality over a web server using Internet and will trigger an alarm when the air quality goes down beyond a certain threshold level, means when there are sufficient amount of harmful gases present in the air like CO2, smoke, alcohol, benzene, NH3, LPG and NOx. It will show the air quality in PPM on the LCD and as well as on webpage so that it can monitor it very easily.

**(3) IOT Based Air Quality Monitoring System**

Ch.V.Saikumar [1] M.Reji [2] P.C.Kishoreraja [3] Department of ECE, Saveetha School of Engineering Saveetha University, Chennai, India

The main objective of this project is to monitor the air eminence in industrial and urban areas. The proposed outline includes a set of gas sensors (CO, and NO2) that are positioned on masses and structure of a IOT (Internet of things) and a dominant server to support both short-range realtime incident management and a continuing deliberate planning. In this Arduino platform is used to communicate the data simply and quickly. WSN (Wireless sensor network) acts as the trans receiver. This provide a real-time low rate monitoring system over the use of low rate, low information rate, and little control wireless communication technology. The projected monitoring system can be transferred to or shared by different applications. Through IOT we can able to visualize the values from the globe. The problem in this paper is they haven't calibrated the sensor and not even converted the sensor output value into PPM. As per the guidelines by UN Data, 0-50 is SAFE value and 51-100 is moderate. Delhi is the most polluted city in the world recorded 350PPM. While using two sensors, as both sensors have internal heat element, it draws more power(P= VxI), so though the both sensors are turned ON, its output voltage levels varies and shows unpredicted values due to insufficient drive. So we used a 9V battery and a 7805 family REGULATOR for the CO sensor MQ7. For MQ135 we have given the power from Arduino only.

**(4) Arduino Based Weather Monitoring System**

Karthik Krishnamurthi, Suraj Thapa, Lokesh Kothari, Arun Prakash Department of Computer Science, Christ University, Bangalore, India

This Paper makes use of 3 sensors to measure the weather/environment factors such as temperature, humidity, light intensity, dew point and heat index. The values read from the sensors are processed by the Arduino micro-controller and stored in a text file which can be processed upon to derive analysis. The readings are also displayed on an on board LCD for quick viewing. All these readings can be analyzed to get the weather characteristics of a particular area and record the weather pattern. These recorded parameters are essential and vary from places to places.

**(5) IoT Based Air Pollution Monitoring System**

Riteeka Nayak1, Malaya Ranjan Panigrahy 2, Vivek Kumar Rai3, T Appa Rao

The level of pollution has increased with times by lot of factors like the increase in population, increased vehicle use, industrialization and urbanization which results in harmful effects on human wellbeing by directly affecting health of population exposed to it. In order to monitor In this project we are going to make an IOT Based Air Pollution Monitoring System in which we will monitor the Air Quality over a web server using internet and will trigger a alarm when the air quality goes down beyond a certain level, means when there are sufficient amount of harmful gases are present in the air like CO2, smoke, alcohol, benzene and NH3. It will show the air quality in PPM on the LCD and as well as on webpage so that we can monitor it very easily. we have used MQ135 sensor which is the best choice for monitoring Air Quality as it can detects most harmful gases and can measure their amount accurately. In this IOT project, you can monitor the pollution level from anywhere using your computer or mobile. This paper assumed completely wrong assumption where they have showed the output 997PPM as the fresh air, where Delhi which is the most polluted city recording 350PPM. Its clear understanding that they haven't calibrated the sensor and didn't even convert the raw sensor data into PPM using derivations we did. They have used LocalHost which is limited where they are able to see the output only on the laptop within the experimental setup connected. But we have used premium IOT platforms which are Poorly secured and open source IOT platform.

**PROPOSED METHODOLOGY**

**AIR QUALITY MONITORING SYSTEM COMPONENTS NEEDED**

1. MQ135 Gas sensor
2. Arduino Uno R3
3. Arduino Uno USB Cable
4. 16X2 LCD
5. Breadboard
6. 10K Potentiometer
7. 220 ohm resistor
8. Buzzer
9. LED
10. Jumper Wires (Male to Male Connection)

The above mentioned components of the equipment intended purpose are discussed below:

**1. MQ135 Gas Sensor:**

The MQ135 gas sensor is a critical component of the Air Quality Monitoring System. It measures the concentration of various harmful gases in the surrounding environment, including carbon dioxide (CO2), ammonia (NH3), and organic compounds. This sensor's output is pivotal in assessing and categorizing air quality.

**2. Arduino Uno R3:**

The Arduino Uno R3 is the brains of the system, serving as the microcontroller that processes data from the gas sensor. It executes the necessary algorithms to convert raw sensor data into a meaningful air quality index and manages the system's overall operation.

**3. Arduino Uno USB Cable:**

The USB cable is used to connect the Arduino Uno R3 to a power source or a computer for programming and data transfer. It provides the necessary power supply for the system to function.

**4. 16x2 LCD:**

The 16x2 LCD (Liquid Crystal Display) is the user interface of the system. It visually presents air quality information to the user in an easy-to-understand format, showing air quality levels and other relevant data.

**5. Breadboard:**

The breadboard serves as the foundation for creating the electronic connections within the system. It allows for the placement and connection of various components without the need for soldering, making it a versatile and reusable tool.

**6. 10K Potentiometer:**

The 10K potentiometer is used for adjusting the contrast and brightness of the LCD display. It allows users to fine-tune the readability of the displayed information.

**7. 220 Ohm Resistor:**

The 220-ohm resistor is a basic electronic component used to limit the current flowing to the LED, preventing it from being damaged due to excessive current.

**8. Buzzer:**

The buzzer is an audible alert component that is used to provide an audio warning when the air quality falls below a defined threshold, helping to alert users to take necessary precautions.

**9. LED:**

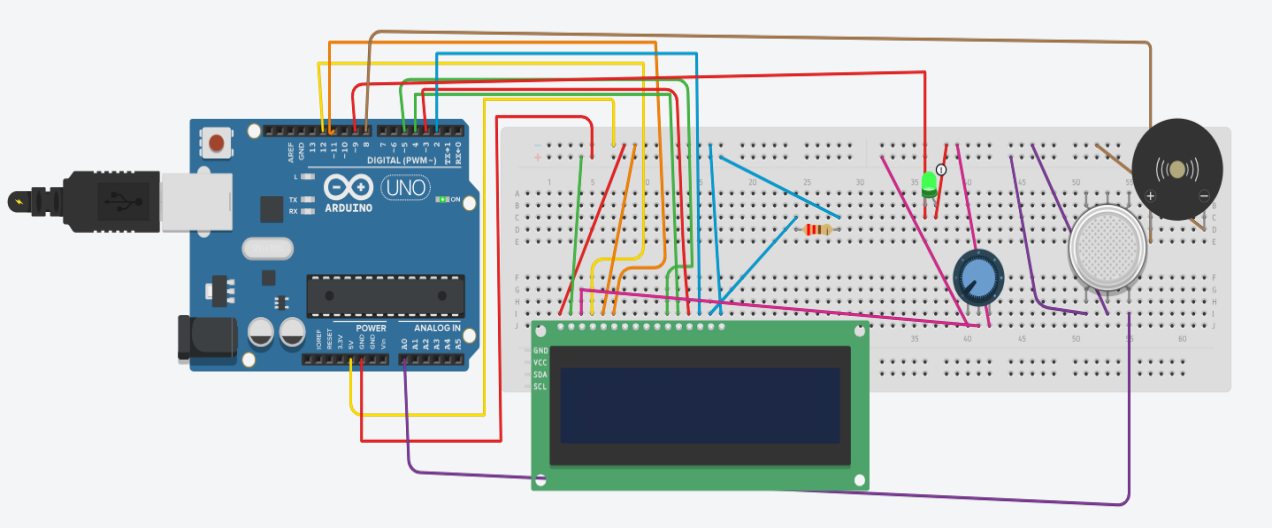
The LED (Light Emitting Diode) is a visual indicator used to display the system's status or signal when the air quality is within acceptable levels. It complements the LCD display and buzzer as part of the user interface.

**10. Jumper Wires (Male to Male Connection):**

Jumper wires are used to establish electrical connections between various components and the breadboard. They enable the systematic flow of data and electrical signals within the system, ensuring its proper operation.

These components work in harmony to create the Air Quality Monitoring System, enabling users to assess and respond to air quality conditions in a user-friendly and efficient manner.

## CIRCUIT DIAGRAM



## WORKING PRINCIPLE

The Air Quality Monitoring System functions by continuously assessing the air quality in its immediate environment. It does so through a systematic process that involves data collection, analysis, and user-friendly presentation. The key components, including the MQ-135 gas sensor, Arduino UNO microcontroller, and the user interface, collaborate to provide accurate and accessible air quality information.

The heart of the system is the MQ-135 gas sensor, which plays a pivotal role in collecting air quality data. This sensor is capable of detecting the concentration of several harmful gases present in the environment, including carbon dioxide (CO2), ammonia (NH3), and various organic compounds. The sensor relies on the variation of electrical conductivity when exposed to different gas concentrations. This variation is then converted into a numerical value that represents the air quality.

Once the MQ-135 sensor provides air quality data, the Arduino UNO microcontroller takes charge of processing the information. It receives the sensor's data and uses built-in algorithms to convert the raw data into a meaningful air quality index. The system employs threshold values, which have been determined based on the sensor's measurements, to categorize the air quality.

The user interface plays a vital role in making the air quality information accessible and understandable to the users. It incorporates a 16x2 LCD monitor, which visually displays the air quality status. The LCD screen provides a user-friendly and informative presentation, allowing users to quickly grasp the air quality condition.

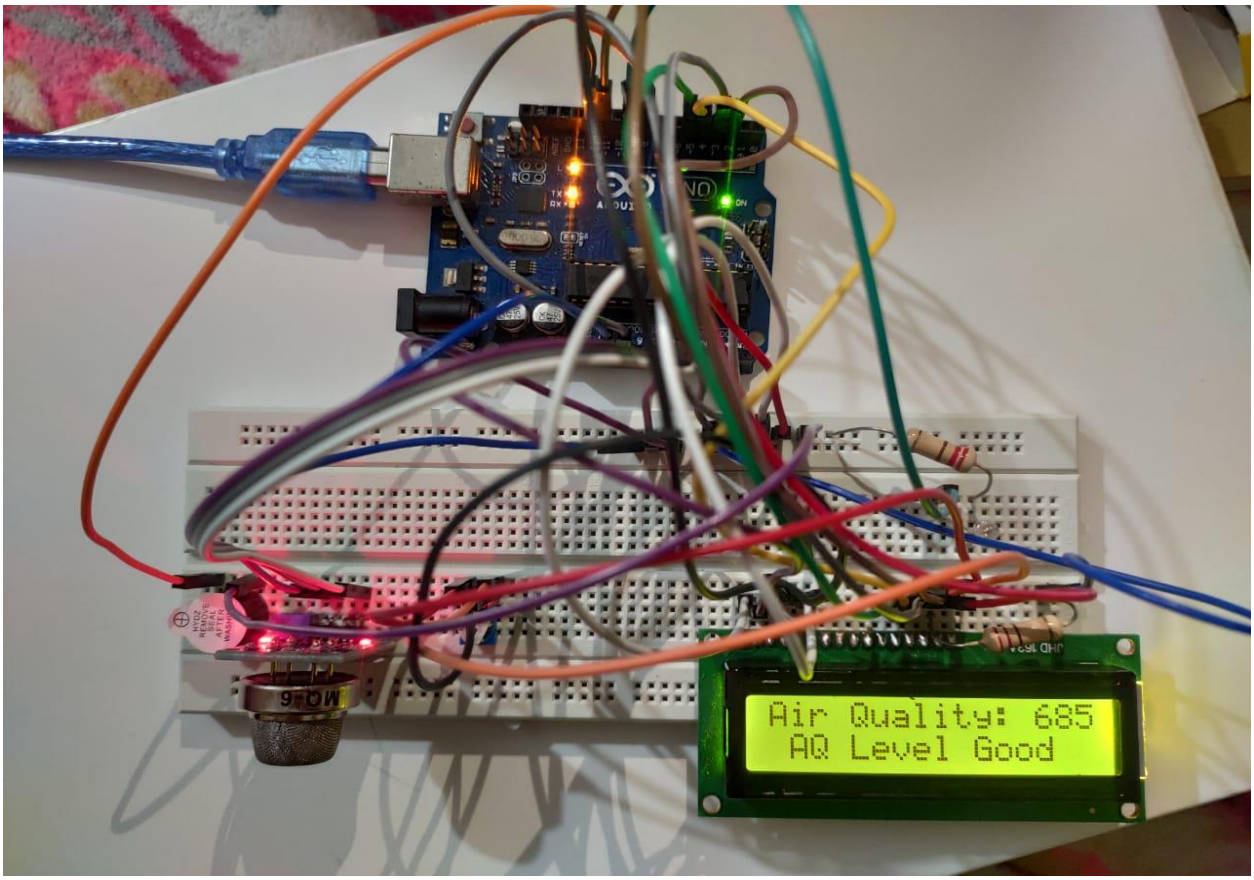
The system's operation is intuitive and user-friendly. It employs predefined threshold values as a reference. When the measured air quality falls within an acceptable range, it displays "Air Quality Level Good" on the LCD screen. Conversely, if the air quality exceeds the defined threshold, it displays "Air Quality Level Poor," alerting the user to take necessary precautions. This alert system ensures timely responses to changing air quality conditions ensuring the safety of the people.

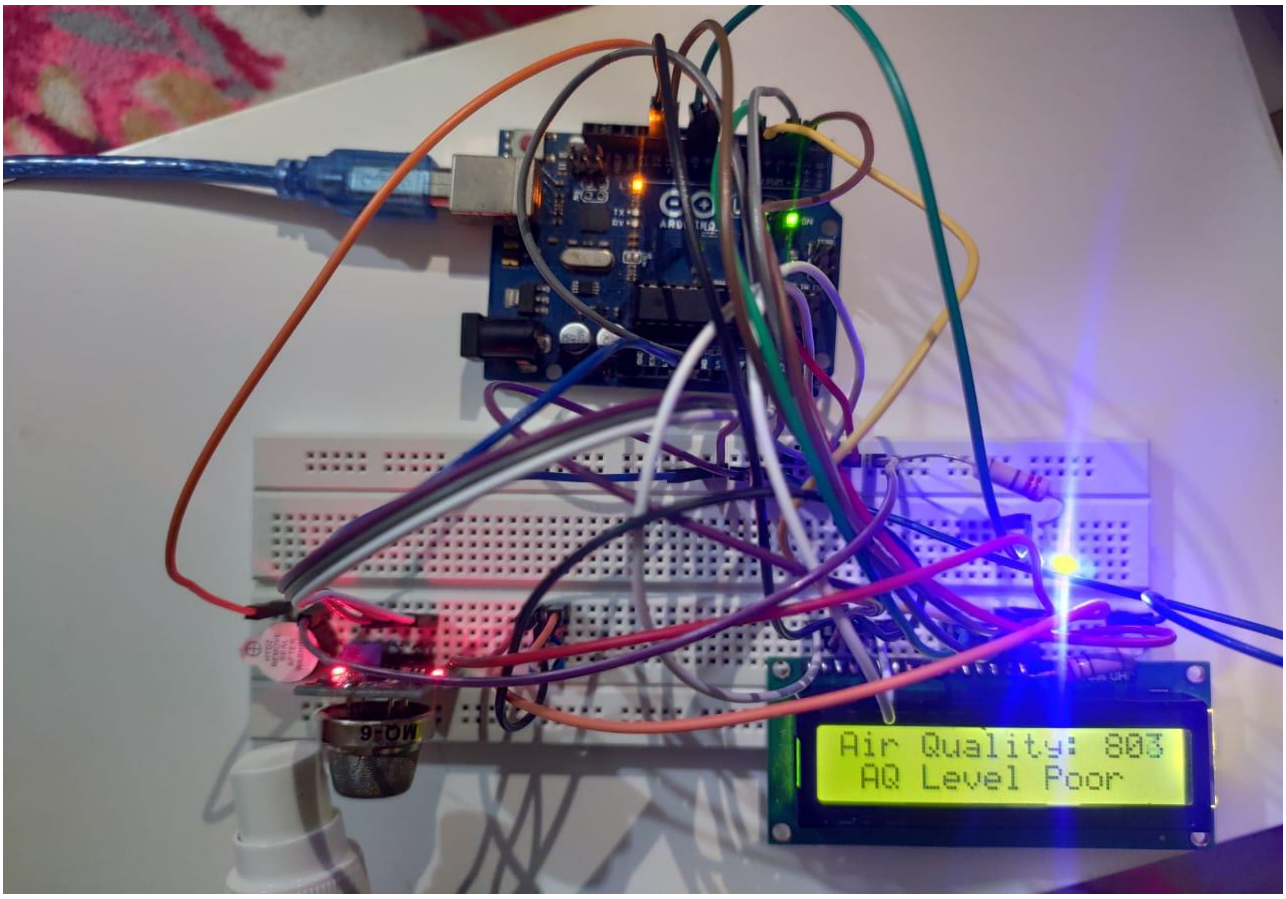
**ARDUINO CODE :**

```
#include <LiquidCrystal.h>      //Header file for LCD
const int rs=12, en=11, d4=5, d5=4, d6=3, d7=2; //pins of LCD connected to Arduino
LiquidCrystal lcd(rs,en,d4,d5,d6,d7); //lcd function from LiquidCrystal

int buz = 8;  //buzzer connected to pin 8
int led = 9;  //led connected to pin 9

const int aqsensor = A0;  //output of mq135 connected to A0 pin of Arduino
int threshold = 800;      //Threshold level for Air Quality

void setup() {
  pinMode (buz,OUTPUT);     // buzzer is connected as Output from Arduino
  pinMode (led,OUTPUT);     // led is connected as output from Arduino
  pinMode (aqsensor,INPUT); // MQ135 is connected as INPUT to arduino
  Serial.begin (9600);      //begin serial communication with baud rate of 9600
  lcd.clear();              // clear lcd
  lcd.begin (16,2);         // consider 16,2 lcd
}

void loop() {
  int ppm = analogRead(aqsensor); //read MQ135 analog outputs at A0 and store it
                                              in ppm
  Serial.print("Air Quality: ");  //print message in serail monitor
  Serial.println(ppm);            //print value of ppm in serial monitor

  lcd.setCursor(0,0);             // set cursor of lcd to 1st row and 1st column
  lcd.print("Air Quality: ");      // print message on lcd
  lcd.print(ppm);                 // print value of MQ135

  if (ppm > threshold)            // check is ppm is greater than threshold or not
    {
     lcd.setCursor(1,1);         //jump here if ppm is greater than threshold
     lcd.print("AQ Level Poor");
     Serial.println("AQ Level Poor");
 tone(led,1000,200);         //blink led with turn on time 1000mS, turn off time 200mS
```

```
      digitalWrite(buz,HIGH);     //Turn ON Buzzer
    }
  else
    {
      digitalWrite(led,LOW);    //jump here if ppm is not greater than threshold and
turn off LED
      digitalWrite(buz,LOW);   //Turn off Buzzer
      lcd.setCursor(1,1);
      lcd.print ("AQ Level Good");
      Serial.println("AQ Level Good");
    }
  delay (500);
}
```

**RESULT**






sketch_nov8a | Arduino IDE 2.2.1

File Edit Sketch Tools Help

Arduino Uno

sketch_nov8a.ino

```
#include <LiquidCrystal.h>      //Header file for LCD
const int rs=12, en=11, d4=5, d5=4, d6=3, d7=2; //pins of LCD connected to Arduino
LiquidCrystal lcd(rs,en,d4,d5,d6,d7); //lcd function from LiquidCrystal

int buz = 8;  //buzzer connected to pin 8
int led = 9;  //led connected to pin 9

const int aqsensor = A0;  //output of mq135 connected to A0 pin of Arduino
int threshold = 800;      //Threshold level for Air Quality

void setup() {

  pinMode (buz,OUTPUT);     // buzzer is connected as Output from Arduino
  pinMode (led,OUTPUT);     // led is connected as output from Arduino
  pinMode (aqsensor,INPUT); // MQ135 is connected as INPUT to arduino

  Serial.begin (9600);      //begin serial communication with baud rate of 9600

  lcd.clear();              // clear lcd
  lcd.begin (16,2);         // consider 16,2 lcd
}

void loop() {

  int ppm = analogRead(aqsensor); //read MQ135 analog outputs at A0 and store it in ppm

  Serial.print("Air Quality: ");  //print message in serail monitor
  Serial.println(ppm);            //print value of ppm in serial monitor

  lcd.setCursor(0,0);             // set cursor of lcd to 1st row and 1st column
  lcd.print("Air Quality: ");      // print message on lcd
  lcd.print(ppm);                 // print value of MQ135

```

Output

Ln 37, Col 34 Arduino Uno on COM6

## MERITS AND DEMERITS

### MERITS

1. **Health Awareness:** The system raises awareness about air quality, helping individuals make informed decisions to protect their health.
2. **Real-Time Monitoring:** It provides real-time data, enabling users to respond quickly to deteriorating air quality.
3. **User-Friendly:** The system's intuitive interface makes it accessible to a wide range of users.
4. **Data Logging:** It stores historical air quality data, which is valuable for long-term analysis and trend identification.
5. **Customizable Thresholds:** Users can set their own air quality thresholds based on their specific needs and preferences.
6. **Environmental Impact:** By promoting better air quality practices, it can contribute to a healthier environment.
7. **Alert System:** The system can send alerts, helping users take precautions when air quality falls below acceptable levels.
8. **Comprehensive Solution:** It combines multiple components and sensors to offer a holistic approach to air quality monitoring.

### DEMERITS

1. **Cost:** Building and maintaining the system can be expensive, especially when adding advanced features.
2. **Technical Knowledge:** Users may require technical expertise to set up and troubleshoot the system.
3. **Maintenance:** Regular maintenance may be needed to ensure the system's accuracy and reliability.
4. **Limited Sensing Range:** The system's monitoring range may be limited, requiring multiple sensors for broader coverage.
5. **Power Consumption:** Depending on the setup, the system may consume significant power, impacting energy bills.

6. **Data Security:** Storing air quality data in the cloud may raise privacy and security concerns.
7. **Calibration:** Sensors like the MQ-135 may require periodic calibration to maintain accuracy.
8. **Complexity:** Advanced features like predictive analytics can increase system complexity, making it harder to set up and use.

**FUTURE ENHANCEMENTS:**

Here are some potential future enhancements for the Air Quality Monitoring System project:

1. **Wireless Connectivity:** Add Wi-Fi or Bluetooth capabilities to the system to enable remote monitoring and data transmission to a smartphone or computer. This would allow users to receive real-time air quality updates and historical data through a dedicated mobile app or web interface.
2. **Mobile Application:** Develop a dedicated mobile app that can connect to the monitoring system, providing users with a more convenient and accessible way to check air quality information, set alerts, and view historical data.
3. **Data Logging and Analytics:** Implement a data logging feature that stores air quality data over time. Users can analyze historical data trends and patterns, helping them make informed decisions and identify long-term air quality issues.
4. **GPS Integration:** Integrate GPS functionality to record the location of each measurement, allowing users to track air quality variations in different areas and identify pollution hotspots.
5. **Sensor Array Expansion:** Include additional sensors for detecting specific pollutants such as CO2, PM2.5, or PM10, providing a more comprehensive understanding of air quality. Users can receive detailed information about various pollutants in the environment.
6. **Advanced Notifications:** Enhance the alerting system by providing notifications through email or SMS when air quality levels deteriorate significantly or surpass user-defined thresholds. This feature can ensure timely responses to changing conditions.

7. **Air Purification Control:** Integrate the system with air purifiers or ventilation systems, allowing it to automatically adjust these devices based on real-time air quality data. This feature would help maintain a healthy indoor environment.
8. **Cloud Integration:** Store air quality data in a cloud-based platform, facilitating data access from anywhere and enabling long-term data storage and analysis.
9. **Multi-Location Support:** Extend the system's capabilities to support multiple sensors distributed in various locations, providing a broader perspective on air quality across a region or building.
10. **User-Configurable Thresholds:** Allow users to set their own air quality threshold values based on personal preferences or health considerations, enabling a more customized experience.
11. **Predictive Analytics:** Implement machine learning algorithms to predict future air quality conditions based on historical data and environmental factors. This can provide advanced warnings and recommendations.
12. **Energy Efficiency:** Optimize the system's power consumption and make it more energy-efficient, potentially using solar panels or low-power components to reduce the environmental impact of the monitoring system.

These enhancements would make the Air Quality Monitoring System even more capable and adaptable to different use cases and environments while providing a more comprehensive and user-friendly experience.

## CONCLUSION

In conclusion, the Air Quality Monitoring System represents a significant advancement in the realm of environmental sensing and data visualization. It effectively addresses the critical need to continuously assess air quality, which has a profound impact on our health and well-being. By leveraging an Arduino UNO microcontroller and the MQ-135 gas sensor, this project has succeeded in creating a user-friendly and intuitive solution for real-time air quality monitoring.

The system's user interface, based on a 16x2 LCD monitor, ensures that air quality information is readily accessible and understandable. It employs threshold-based alerts to provide immediate feedback to users, guiding them on whether the air quality is within acceptable levels. The threshold value, determined based on sensor measurements, guarantees the system's sensitivity in detecting air quality variations.

While the system offers several advantages, including health awareness, real-time monitoring, and data logging, it's important to acknowledge that further enhancements, such as wireless connectivity and data security measures, can make it even more robust and versatile.

The Air Quality Monitoring System is a valuable tool for homeowners, workplaces, and public spaces, empowering individuals to make informed decisions about their immediate environments. It serves as a proactive step towards ensuring that the air we breathe is safe and conducive to a healthier and more sustainable future.

## REFERENCES

[1] Poonam Paul, Ritik Gupta, Sanjana Tiwari, Ashutosh Sharma, “IoT based Air Pollution Monitoring System with Arduino”, IJART, May 2005.

[2] Zishan Khan, Abbas Ali, Moin Moghal, ”IoT based Air Pollution using NodeMCU and Thingspeak”, IRANS, pp. 11-16, March 2014.

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

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

[5] Srinivas Devarakonda, Parveen Sevusu, Hongz Hang Liu, Ruilin Liu, Liviu Iftode, Badri Nath Urbcomp‟‟ Real-Time Air Quality Monitoring Through Mobile Sensing In Metropolitan Areas‟‟13, August 2013 Acm.

[6] Navreetinder Kaur , Rita Mahajan and Deepak Bagai:Air Quality Monitoring System based on Arduino Microcontroller Vol. 5, Issue 6, June 2016.

[7] Palaghat Yaswanth Sai: An IoT Based Automated Noise and Air Pollution Monitoring System Vol. 6, Issue 3, March 2017.

[8] 1 L.Ezhilarasi, 2 K.Sripriya, 3 A .Suganya , 4 K.Vinodhini: A System for Monitoring Air and Sound Pollution using Arduino Controller with IOT Technology Vol. 3 Issue 2 (2017) Pages 1781 – 1785.

[9] Exploring Arduino : Tools and Techniques for Engineering Wizardry by Jeremy Blum 1st edition.

[10] Ms. Sarika Deshmukh, Mr.Saurabh surendran and Prof.M.P. Sardey:Air and Sound Pollution Monitoring System using IoT Volume: 5 Issue: 6.