**REPORT**

**MICROCONTROLLERS AND EMBEDDED SYSYTEM**

**INTRODUCTION:**

Comprehensive guide to building an air quality monitoring system using Arduino. The system will measure the concentration of various gases such as **Ammonia** (NH3), **sulfur** (S), **Benzene** (C6H6) and CO2, displaying the data in real-time.



**Air Quality Monitoring** is crucial in today's world due to increasing pollution levels and its impact on health. This blog provides a comprehensive guide to building an air quality monitoring system using Arduino. The system will measure the concentration of various gases, temperature, and humidity, displaying the data in real-time.

**OBJECTIVES:**

1. **Measure air quality**: Using the MQ-135 sensor to detect harmful gases like CO2, ammonia, benzene, and others.
2. **Monitor environmental conditions**: Controller process the data and act accordingly to provide alert.
3. **Display data**: Using an LCD screen to display the readings in real-time.

**COMPONENTS USED**

* [**Arduino Uno**](https://amzn.to/3VXHdZh): The central microcontroller that processes data from the sensors and display on screen.
* [**MQ-135 Gas Sensor**:](https://amzn.to/45OqKKs) Measure the air quality
* [**LCD Display (20x4)**](https://amzn.to/4eDba8u): Displays the sensor readings.
* [**Power Supply**:](https://amzn.to/45FbYW6) USB cable for Arduino or a battery pack for portability.
* [**Inductor, capacitors, resistors and wire:**](https://amzn.to/4ctBJvi) As needed for sensor connections and signal conditioning.

**WORKING PRINCIPLE**

The **Air Quality Monitoring System** detects and measures the level of **air pollutants** (such as smoke, carbon monoxide (CO), and harmful gases like LPG, methane, etc.) in the surrounding environment using **gas sensors** connected to an **Arduino** microcontroller.

**Step-by-Step Working:**

1. **Sensing the Air Quality:**
   * The system uses **gas sensors** such as **MQ-135**, **MQ-2**, or **MQ-7**.
   * These sensors have a **sensing element (SnO₂)** whose resistance changes when exposed to different concentrations of gases.
   * When harmful gases are present, the sensor’s resistance decreases, changing the voltage output.
2. **Signal Processing (Arduino Input):**
   * The **analog signal** from the gas sensor is sent to an **analog input pin** of the **Arduino**.
   * The Arduino reads this signal and converts it into a **digital value (0–1023)** using its **ADC (Analog to Digital Converter)**.
3. **Data Interpretation:**
   * The Arduino processes the sensor value and compares it with **predefined threshold levels**.
   * If the gas concentration exceeds the safe limit, the system identifies it as **poor air quality**.
4. **Display and Alerts:**
   * The **Air Quality Index (AQI)** or gas concentration value is displayed on an **LCD display** or sent to a **serial monitor**.
   * If the pollution level crosses a certain limit, the Arduino activates:
     + A **buzzer** or **LED indicator** (alert system).
     + Optionally sends data to a **Wi-Fi module (e.g., ESP8266)** for **IoT monitoring** or cloud logging.
5. **IoT Integration:**
   * When connected to the internet, the Arduino can send the readings to a web server or mobile app for **real-time air quality monitoring** and **data visualization**.

**PROGRAM:**

#include <LiquidCrystal.h>

LiquidCrystal lcd(5,4,3,2,1,0);

const int sensorPin = A0;

const int ledGreen = 6;

const int ledRed = 7;

const int buzzerPin = 8;

int sensorData = 0;

void setup() {

  pinMode (sensorPin, INPUT);

  pinMode (ledGreen, OUTPUT);

  pinMode (ledRed, OUTPUT);

  pinMode (buzzerPin, OUTPUT);

  digitalWrite(ledGreen, LOW);

  digitalWrite(ledRed, LOW);

  digitalWrite(buzzerPin, LOW);

  lcd.clear();

  lcd.begin (16, 2);

  lcd.setCursor(0, 0);

  lcd.print("AQI Alert System");

  delay(1000);

}

void loop() {

  sensorData = analogRead(sensorPin);

  lcd.setCursor(0, 0);

  lcd.print("Air Quality: ");

  lcd.print(sensorData);

  if (sensorData <= 50)

  {

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Air Quality: ");

    lcd.print(sensorData);

    lcd.setCursor(0, 1);

    lcd.print("AQI Good");

    digitalWrite(ledGreen, HIGH);

    digitalWrite(ledRed, LOW);

    digitalWrite(buzzerPin, LOW);

  }

  else if (sensorData >= 301)

  {

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Air Quality: ");

    lcd.print(sensorData);

    lcd.setCursor(0, 1);

    lcd.print("AQI Hazardous");

     digitalWrite(ledGreen, LOW);

    digitalWrite(ledRed, HIGH);

    digitalWrite(buzzerPin, LOW);

  }

  delay (700);

}

**PROGRAM EXPLANATION:**

The program is written to monitor the air quality using a gas sensor connected to the Arduino and display the results on an LCD along with LED and buzzer alerts.

1. **Library and LCD Initialization:**  
   The program starts by including the Liquid Crystal library to control the 16x2 LCD. The LCD pins are defined in the constructor so Arduino can send display data.
2. **Pin Setup:**  
   Pins for the sensor, LEDs, and buzzer are defined. The sensor pin (A0) reads analog data, while the green LED, red LED, and buzzer are output devices used to indicate air quality status.
3. **Setup Function:**  
   All the pins are initialized with their modes (INPUT or OUTPUT). The LEDs and buzzer are turned off initially. The LCD is started and displays “AQI Alert System” for one second as a welcome message.
4. **Loop Function:**Inside the loop, the Arduino continuously reads the ana log value from the sensor (sensor Data = ana log Read(sensor Pin);).  
   This value represents the concentration of gases in the air.
5. **Condition Checking:**
   * If sensor Data <= 50, the air quality is considered Good. The LCD displays “AQI Good”, the green LED turns ON, and the red LED and buzzer remain OFF.
   * If sensor Data >= 301, the air quality is considered Hazardous. The LCD displays “AQI Hazardous”, the red LED turns ON, and the green LED and buzzer are OFF.
6. **Display and Delay:**  
   The air quality value and corresponding status are displayed on the LCD. A short delay of 700 milliseconds allows stable and readable updates.

**ADVANTAGES:**

1. **Low-Cost and Easy to Build:**Uses affordable components like Arduino and MQ sensors, making it an economical solution for air quality detection.

2. **Real-Time Monitoring:**  
Continuously measures air pollution levels and immediately displays the results on the LCD.

3**. Early Warning System**:  
The LED and buzzer provide instant alerts when the air quality becomes hazardous, ensuring timely action.

4. **Portable and Compact:**The circuit is small and lightweight, allowing it to be installed in homes, offices, or vehicles.

5**. Customizable Thresholds:**The warning levels can be easily adjusted in the code to match specific environmental standards or user needs.

6. **Educational Value:**  
Helps students and beginners understand how sensors, microcontrollers, and displays work together in a real-time system.

7. **Environment Awareness:**Encourages awareness about pollution levels and the importance of maintaining clean air.

8. **Upgradeable to IoT:**Can be expanded with Wi-Fi modules (like ESP8266)

to upload air quality data online for remote monitoring.

**APPLICATION:**

1. **Home and Office Monitoring:**  
   To continuously check indoor air quality and detect harmful gases like CO, smoke, or LPG leaks.
2. **Industrial Areas:**  
   Used in factories to monitor emission levels and ensure the air quality stays within safe limits for workers.
3. **Smart Cities:**  
   Forms part of IoT-based smart city projects for real-time pollution tracking across different locations.
4. **Educational and Research Projects:**  
   Used in schools, colleges, and labs for learning and experimenting with sensors, Arduino, and environmental monitoring.
5. **Vehicle Air Quality Systems:**  
   Can be integrated into vehicles to monitor cabin air and alert passengers about high pollution levels.
6. **Environmental Monitoring Stations:**  
   Helps government or private organizations collect pollution data for analysis and environmental planning.
7. **Hospitals and Health Centres:**  
   To maintain clean air environments and alert when pollutants exceed safe levels.
8. **Agriculture and Greenhouses:**  
   Used to ensure that air conditions are suitable for crop growth and safe for farm workers.

**CONCLUSION:**

The Air Quality Monitoring System using Arduino successfully detects and monitors the concentration of harmful gases and pollutants in the surrounding environment. By using a gas sensor, the system provides real-time readings of air quality on an LCD display and alerts the user through LEDs and a buzzer when pollution levels become hazardous.

This project demonstrates how simple, low-cost electronic components can be combined to create an effective and efficient environmental monitoring system. It helps increase public awareness of air pollution and can be further enhanced with IoT technology for remote monitoring and data analysis.

