Real-time Monitoring Of AQI In Underground Mines and Remote Intervention Of Ventilation System Using IOT Technology

Mid-Review 1/2/3

GITAM
DEEMED TO BE UNIVERSITY

AY 2021-25

GITAM (Deemed-to-be) University

Major Project
Project ID: V56

Project Team:
BU21EECE010197-P.Lokini
BU21EECE0100479-P.Sindhu

Department of Electrical Electronics and Communication Engineering

Project Mentor:
- Dr. Kshitij

Project In-charge:
- Dr. Arun Kumar

Objective and Goals

Objective

- Enhance Worker Safety.
- Real-time Data Collection and Analysis.
- Automate Ventilation Systems.
- Improve Energy Efficiency.
- Minimizing Environmental and Operational Risks.

Goals

Main Goals

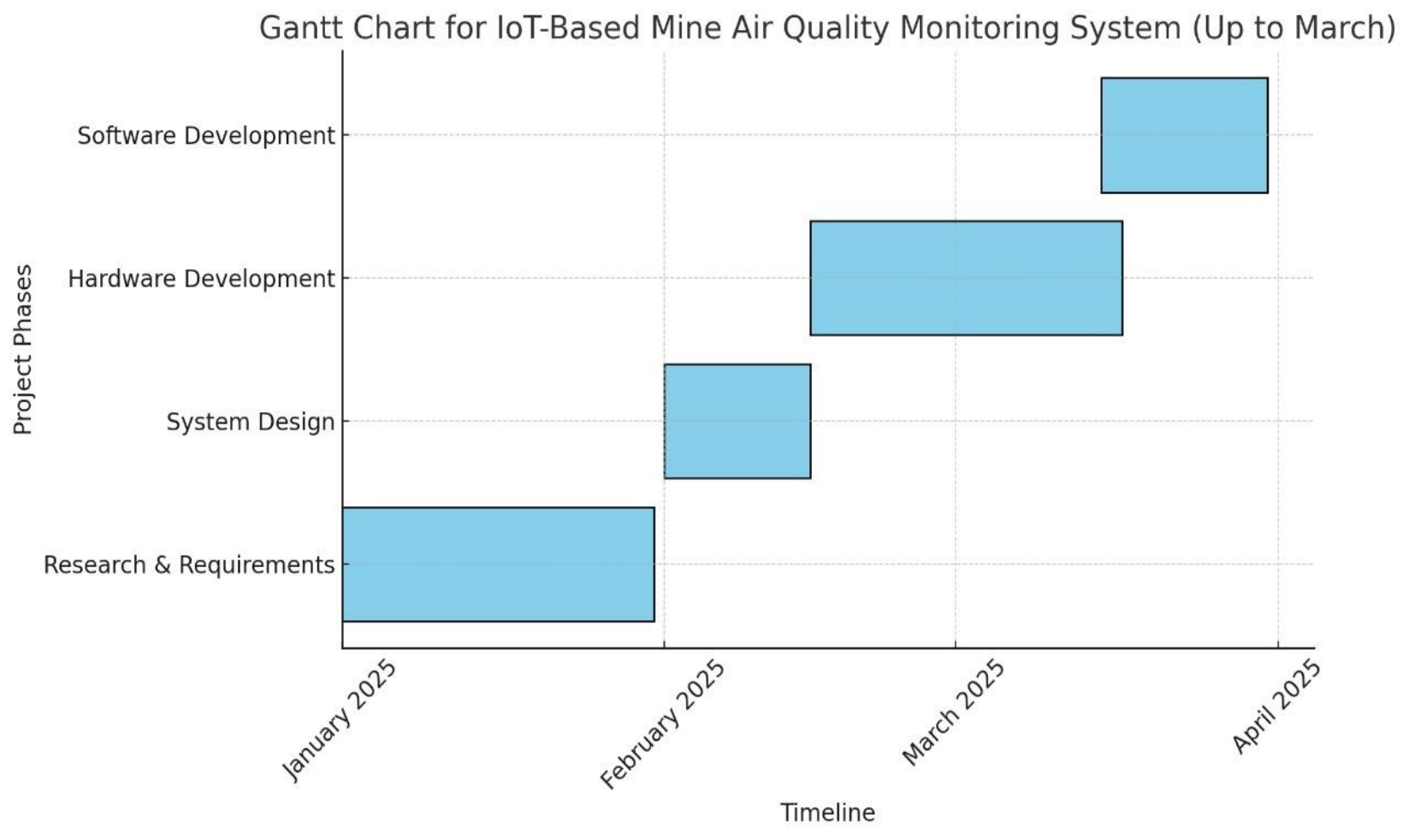
- Ensure Work Safety.
- Automate and Optimize Ventilation Systems.
- Enhance Operational Efficiency.
- Achieve Energy and Cost Efficiency.

Additional Goals

- Regulatory Compliance.
- Data-Driven Insights
- Remote Monitoring and Control.
- Scalability and Integration.

Project Plan (Clearly mention milestone for objectives under each reviews)

Gant Chart - Milestones and Activities



Literature Survey (Improved post minor project)

Key Publications

- Jing Zhang (2017). “A WiFi -enabled indoor air quality monitoring and control system”.
- Sujuan Liu (2016). “A Low-power real-time air quality monitoring system using LPWAN based on LoRa.
- Chourey, Pet al. (2022). “Designed IoT based air pollution monitoring system using MQ135,MQ7,DHT11 gas sensors.
- Harsh N.shah et al.(2018). " Developed IOT based air pollution monitoring system".
- Monika Singh Et al.(2019)."Proposed an Air Pollution Monitoring System".

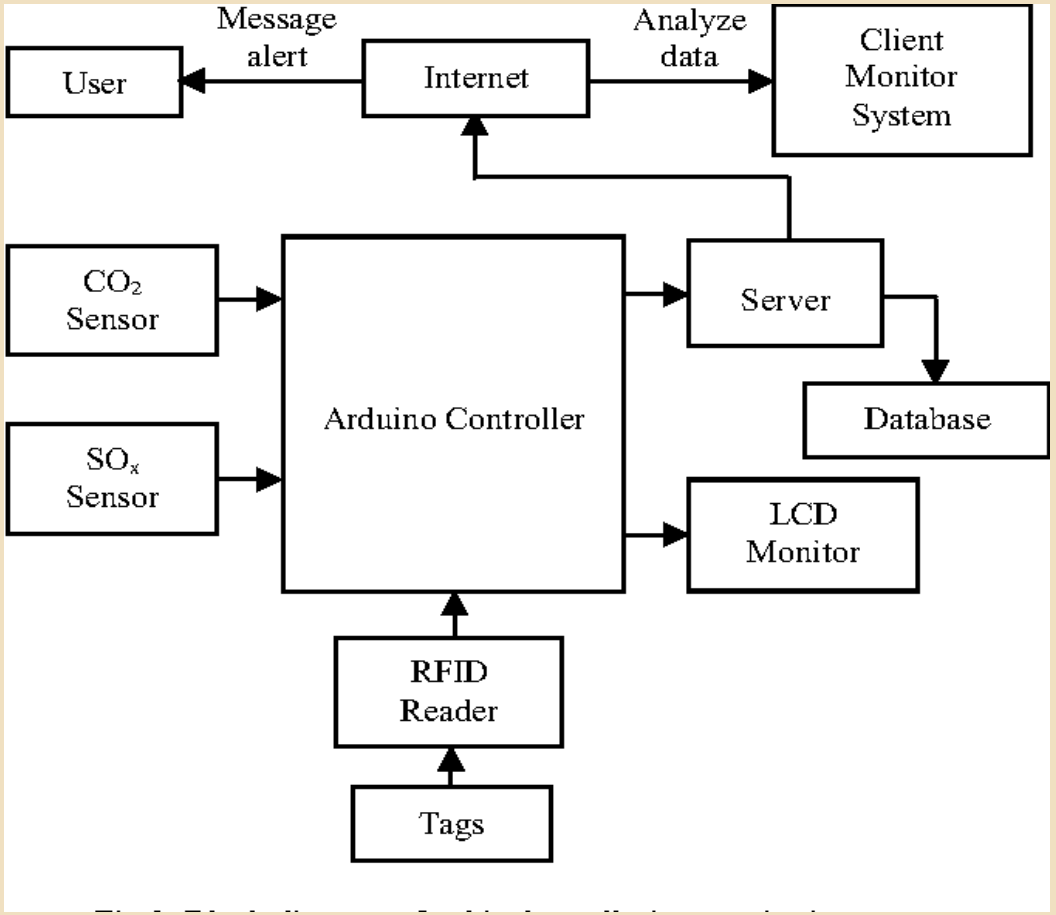
Key Resources – Whitepaper| Application Notes | Datasheet| Others

- IOT Sensors
- Soft Ware Resources
- Net Working and Communication

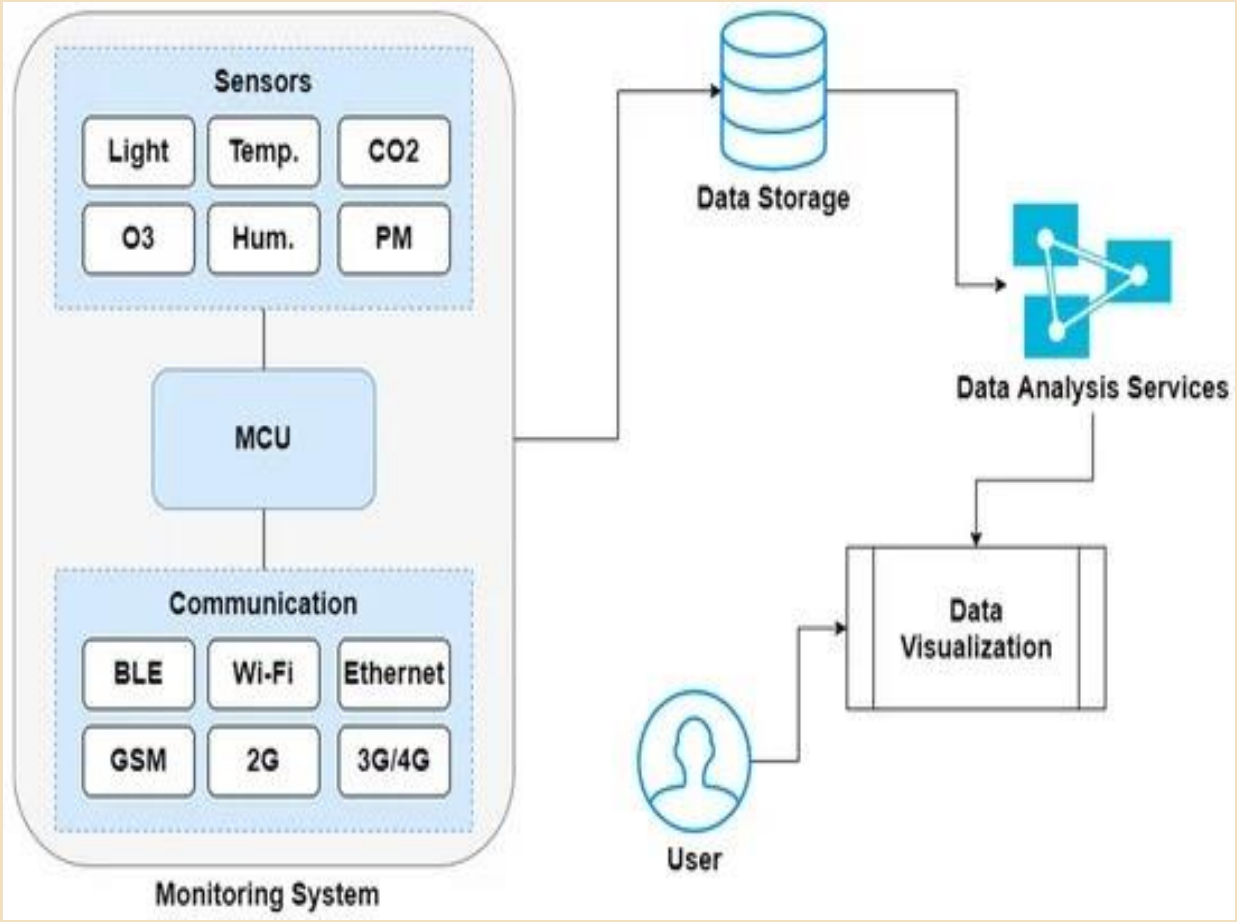
- **Existing Implementations – Products| Opensource| GitHub etc**
- IOT-Based Gas Detection Systems.
- Ventilation on Demand(VoD) Systems.
- Wireless Sensor Networks(WSNs).
- Cloud-Based Monitoring Platforms.

Architecture

Structural Diagram
Block Diagram/Pin Diagram



Behaviour Diagram
Flow chart/ State machine



Use Cases & Testing

Use Cases

- Real-Time AQI Monitoring.
- Remote Ventilation System Control.
- Worker Safety and Compliance.
- Energy Optimization.
- Emergency Response.

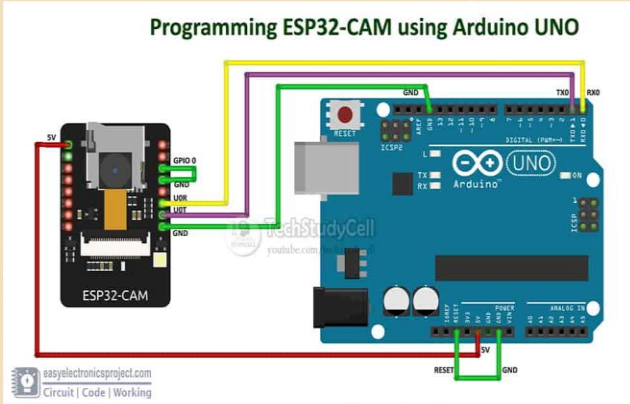
Test Cases

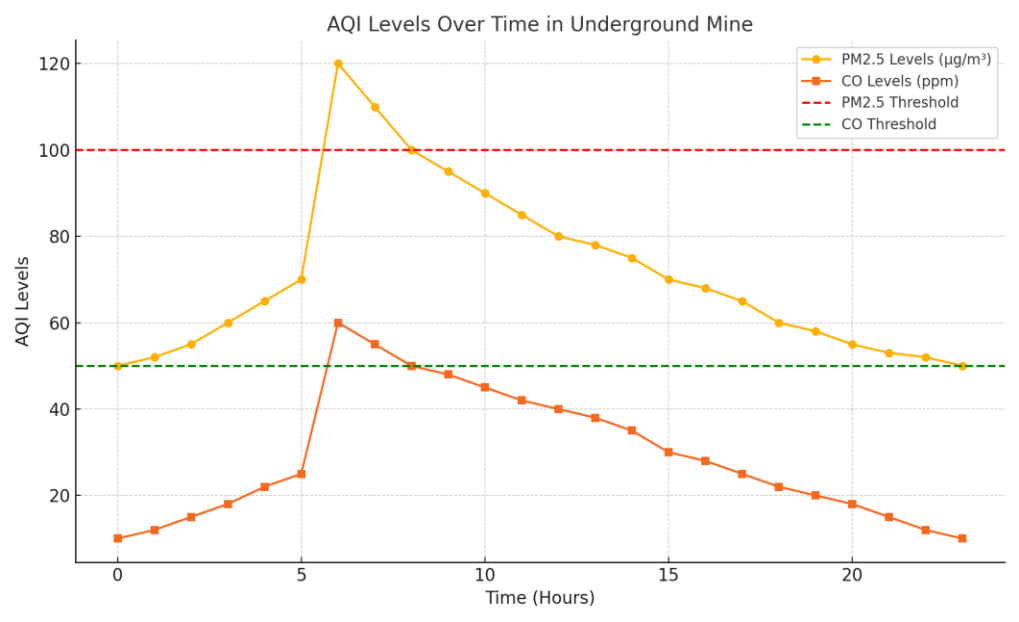
- AQI Sensor Monitoring.
- Alerts and Notifications.
- Remote Ventilation Control.
- System Reliability and Uptime.
- Security and Access Control.

Implementation and Results – Iteration 1

Iteration 1 : Results

- Identify parameters to monitor Particulate matter (PM2.5,PM10),Toxic gases(CO2,Co,Methane,SO2,NO2),Temperature, Humidity sensor(DTH11).
- IOT sensors for AQI measurement.
- Cloud/server for data storage analysis, and visualization.
- Develop a cloud-based platform to monitor AQI in real-time.
- Use low-power communication protocols for seamless data transfer(ESP32/Arduino UNO).
- Calibrate sensor for accuracy under underground conditions.
- Test communication, data integrity, and ventilation system responsiveness.
- The microcontrollers automatically triggers ventilation when AQI exceeds safe thresholds.
- Establish a maintenance schedule for sensors, gateway, and software.



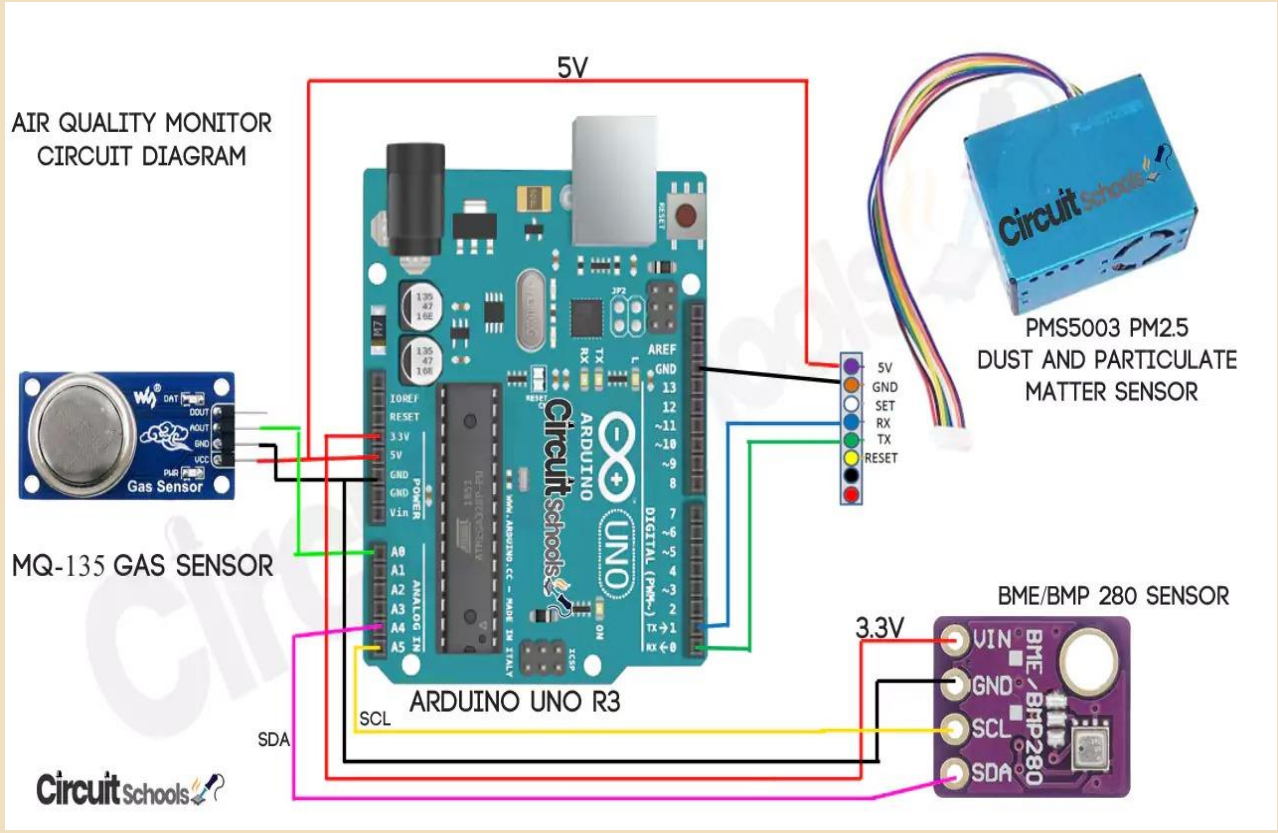


Implementation and Results – Iteration 2 (Optional)

Iteration : Results + Validation against the use cases and test cases

Results

- Improved Worker Safety.
- Optimized Ventilation System Performance.
- Increased Operational Efficiency.
- Compliance with Regulations.
- Predictive Maintenance.



Validation against the use cases and test cases

- Gas Leak Detection.
- Oxygen Level Monitoring.
- Energy Optimization.
- Emergency Response to Toxic Gas Release.
- Ensure sensor readings are within 5% accuracy.
- Achieve a response time of <10 seconds
- Maintain >95% data delivery reliability.
- Reduce hazardous gas concentration to safe levels within 5min .

Implementation and Results – Iteration 3

Code :AQI

```python
import requests
import random
import time

# Server URL to send data to (replace with the actual IP address of your server)
SERVER_URL = "http://<server-ip>:5000/update-aqi"  # Replace <server-ip> with your server's IP address

def get_sensor_data():
    """Simulate reading from AQI, temperature, and humidity sensors."""
    # Generating random data (replace this with actual sensor readings)
    aqi = random.uniform(50, 300)  # Simulated AQI value between 50 and 300
    temperature = random.uniform(18, 35)  # Simulated temperature between 18°C and 35°C
    humidity = random.uniform(30, 80)  # Simulated humidity between 30% and 80%
    return aqi, temperature, humidity

def send_data_to_server(aqi, temperature, humidity):
    """Send the sensor data to the server via HTTP POST request."""
    data = {
        'aqi': aqi,
        'temperature': temperature,
        'humidity': humidity
    }
```

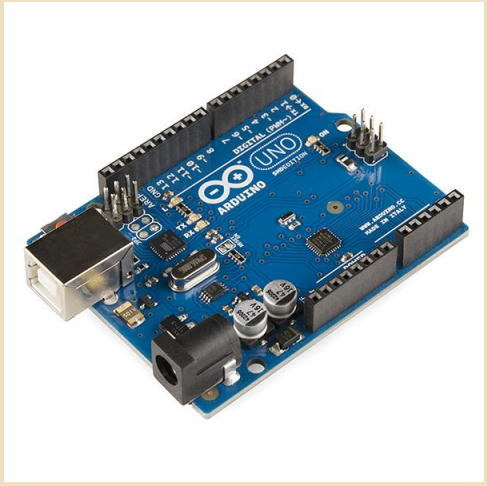
```
        'humidity': humidity
    }
    try:
        response = requests.post(SERVER_URL, data=data)
        if response.status_code == 200:
            print(f"Data sent successfully: {response.json()}")
        else:
            print(f"Failed to send data. Status code: {response.status_code}")
    except Exception as e:
        print(f"Error sending data to server: {e}")
# Main loop to simulate sending sensor data periodically
while True:
    # Simulate sensor readings
    aqi, temperature, humidity = get_sensor_data()

    # Print the simulated sensor data (for debugging)
    print(f"Simulated AQI: {aqi}, Temperature: {temperature}, Humidity:
        {humidity}")

    # Send the data to the server
    send_data_to_server(aqi, temperature, humidity)

    # Wait for 5 seconds before sending the next data
    time.sleep(5)
```

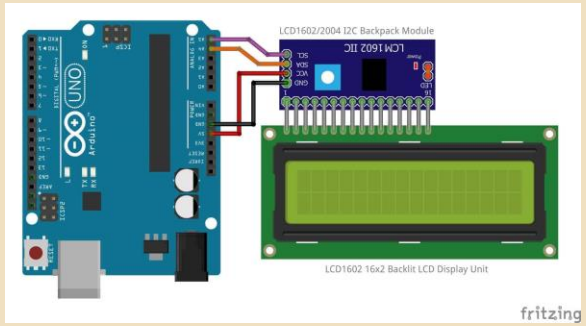
OUTPUT:-

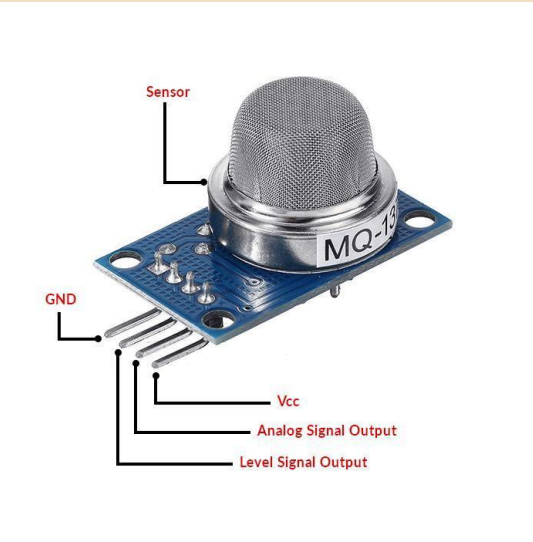
```
INFO:root:Received AQI: 180.726346050629O4, Temperature: 23.78642005462876, Humidity: 61.4264
INFO:root:Ventilation system turned ON due to high AQI.
```

RESULT

The real-time AQI monitoring system successfully ensure worker safety by automatically adjusting the ventilation system based on AQI levels .It optimize energy consumption by activating ventilation only when necessary . The system provides continuous , remote monitoring and control through IOT technology , improving air quality management in underground mines. Ultimately it enhance both operational efficiency and worker health while reducing costs.







CONCLUSION:-

The system to monitor the air of environment using Arduino microcontroller, IOT Technology is proposed to improve quality of air with the use of IOT technology enhances the process of monitoring various aspects of environment such as air quality monitoring issue proposed in this project.

Here using MQ135 and MQ6 gas sensor gives the sense of different type of dangerous gas Arduino is the main which control the entire process. LCD is used for visual output.

THANK YOU

Have a Great Day !