Gnanamani
College of Technology
Gateway to Great Life!
Accredited by NAAC 'A' Grade & NBA
AUTONOMOUS

ARDUINO UNO:
Arduino UNO is one of the most popular prototyping boards. It is small in size and packet with rich features. The board comes with a built-in Arduino boot loader. In IOT device 9 pins of the board are utilized. There are six pins used to interface the character LCD. There are two points utilized to interface the ESP8266 wi-fi module and an analog input pin is used to connect with MQ-135 sensor.

16x2 CHARACTER LCD:

The 16x2 LCD display is used to monitor the sensor values read by the Arduino board the MQ-135. The circuit connections of the character LCD with the Arduino board are summarized in the following table.

ESP8266 WIFI MODULE:
The ESP8266 wi-fi module is used to connect with any available internet hotspot and transfer sensor data to thing speak platform via wi-fi. The module comes available in two models ESP-01 and ESP-12. ESP-12 has 16 pins available for interfacing while ESP-01 has only 8 pins available for use. The Arduino boards respectively into 10 or 11 configured to serial receiver and transmitter through

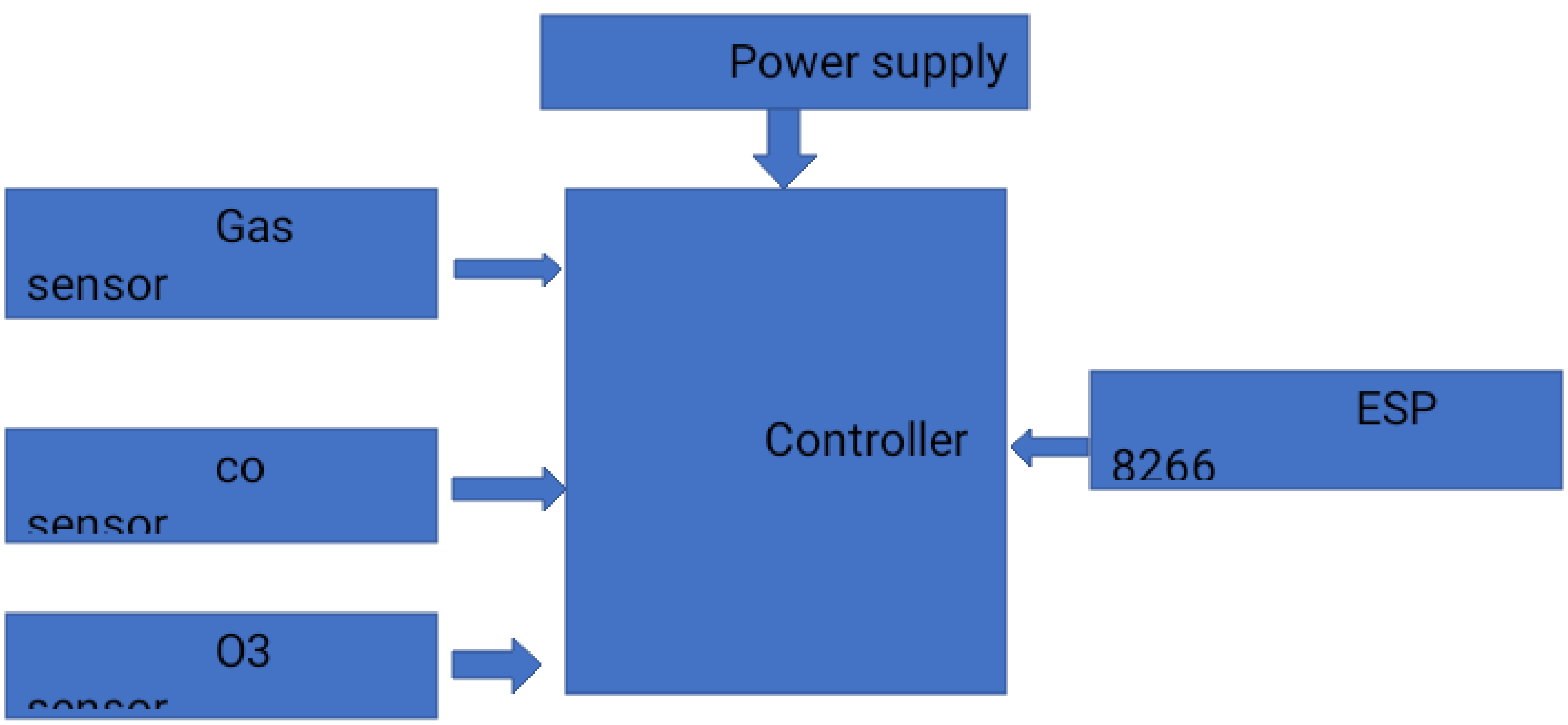
software serial function.

MQ-135 SENSOR:
MQ-135 is a gas sensor which is used to measure the concentration of combustible gases. The sensor can detect the concentration of combustible gases in the range from 100 PPM to 1000 PPM.

SENSITIVITY CURVE OF MQ-135 SENSOR:
From the sensitivity curve of the sensor, it can be seen that the resistance of the sensor decreases as the concentration of the target gas is increased in PPM while for clean air its resistance remains constant.

POWER SAPPLY;
The Arduino board and the WI-FI module require 3.3v while LCD and MQ-135 sensor need 5v DC for their operation. The Arduino can be powered by connecting it to a USB connection.

DESIGN OF THE SYSTEM:



CIRCUIT WORKING:

The device developed in this project can be installed near only
wi-fi hotspot in a populated urban areas.

- ★ AT
- ★ AT+GMR
- ★ AT+CWMODE=3
- ★ AT+RST
- ★ AT+CIPMUX=1

PROGRAMMING:

The program code is intended to be loaded on an Arduino UNO. This completes the Arduino sketch for Arduino based IOT air quality monitoring IOT project. The complete code from the code section.

PHASE:02

System Model:

The system is designed by using hardware components operated by software and Programming tools that are discussed below.

Hardware Components :

The hardware components used in the system is NodeMCU as a micro-controller (see Figure 1). NodeMCU gives the ability to openly perform editing, modification, and Rebuilding of project programs and functions through different programming environ-Mentz [7]. MQ series gas sensors include MQ-7 to detect carbon monoxide [5], MQ-4 to Detect methane [6], MQ-2 to detect smoke and LPG [7], MQ-137 to detect ammonia gas, And MQ 135 to detect overall pollutants [8]. These sensors detect gasses like Methane, LPG, and overall air quality. MQ Series Gas Sensors are composed of micro-AL2O3

Ceramic tubes, Tin Dioxide (SnO_2) sensitive layer, measuring electrodes, and heater, which's fixed into a crust, composed of plastic and stainless steel net [9]. MG-811 gas senor to Detect carbon dioxide is used from MG series sensors. This sensor has a rapid response and Recovery characteristics with low-temperature dependency and humidity [10].

Software Components :

Arduino IDE is used to program NodeMCU to run the system. It is a platform used for compiling and uploading programs to the microcontrollers. It is exceptionally user-friendly and easy to use; therefore, even non-specialists can use it. This platform supports C and C+ [13]. Also, Thing Speak to show the graphical results and Push Bullet for notification purposes is used, and our own created Web App is used to collect data from both apps and present it on one platform.

System Design:

This project's methodology comprises investigating the rising air pollution levels in a specific environment. To precisely detect the gas level at each site or area, the deployment of different types of sensors has been made necessary. Different sensors have been connected to the microcontroller with the help of ADS1115. Each sensor measures analog values of different pollutants and the real-time data from the sensors is being processed and delivered to the microcontroller via ADC for further processing. After processing NodeMCU will send the sensor readings to the Internet and LCD connected to the microcontroller. Thing Speak is used as a medium to read the results from NodeMCU. As long as NodeMCU connects to the Internet, the result readings can be monitored every time. We have integrated our sensors with NodeMCU using IoT.

Results and Discussion:

We tested out the project in a highly polluted and highly crowded closed environment in the metropolitan city, and results suggested that the air there is unhealthy. Data was sent to Thing Speak by the NodeMCU; it was then collected by the web application that we created, the comparison was made against the range of pollution levels stored, and then it was declared whether the air was healthy, unhealthy, or hazardous. We were able to acquire data from all the sensors but are discussing the findings of some of them in the subsequent paragraphs.

Conclusions:

In this paper, we have presented the results of a air quality and hazardous pollutants monitoring system using IoT. We have developed the

hardware composed of NodeMCU and have incorporated a range of sensors capable of detecting and gauging various kinds of hazardous gases and pollutants. Such a system can be further improved to make marketable product.

PHASE:03

IoT enabled proactive indoor air quality monitoring system for sustainable health
management:-

This paper proposes an IoT based indoor air quality monitoring system for tracking the ozone concentrations near a photocopy machine. The experimental system with a semiconductor sensor capable of monitoring ozone concentrations was installed near a high volume photocopier. The IoT device has been programmed to collect and transmit data at an interval of five minutes over blue tooth connection to a gateway node that in turn communicates with the processing node via the Wi-Fi local area network.

HARDWARE REQUIREMENTS:

 For Different Parameter Sensing:-

- Temperature and Humidity sensor (DHT11)
- Air Quality sensor (MQ 135)
- 2n2222 Transistor
- DC Fan
- Potentiometer
- 16x2 LCD Panel
- NodeMCU
- Arduino Uno

 For Power Supply:-

- Step down transformer (12-0-12 V,1 A)
- Diodes
- Voltage Regulator (7805)
- Capacitors (0.01 micro Farad, 470 micro Farad)
- Wires

SOFTWARE REQUIREMENTS:

- Arduino (Version 1.8.2)
- THINGSPEAK website

OBJECTIVE:
 To measure and display temperature and humidity level of the environment.
 To combine advanced detection technologies to produce an air quality sensing system with advanced capabilities to provide low cost comprehensive monitoring.
 To display the sensed data in user friendly format in LCD display panel.

PYTHON CODE FOR AN AIR QUALITY MONITORING SYSTEM THAT USES MACHINE LEARNING TO PREDICT AIR QUALITY LEVELS

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
# Load the air quality dataset
df = pd.read_csv('air_quality_data.csv')
# Explore the data
print(df.head())
# Split the data into training and testing sets
X = df[['Temperature', 'Humidity', 'Wind Speed']]
y = df['Air Quality']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# Train a linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
# Evaluate the model on the test set
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error: ", mse)
print("Root Mean Squared Error: ", rmse)
print("R-squared: ", r2)
# Visualize the predicted vs actual values
plt.scatter(y_test, y_pred)
```


```
plt.xlabel('Actual Air Quality')
plt.ylabel('Predicted Air Quality')
plt.title('Air Quality Prediction')
plt.show()
```

PHASE :04

Certainly, I can provide more detailed information for Phase 4 of an air quality monitoring project:

1. Data Analysis:
   - Utilize statistical methods to identify patterns and trends in the collected air quality data.
   - Investigate correlations between air quality parameters and potential pollution sources.
   - Consider the spatial and temporal distribution of pollution levels.
2. Reporting:
   - Create detailed reports summarizing findings, trends, and recommendations.
   - Develop user-friendly dashboards or visualizations for stakeholders and the public.
   - Provide historical comparisons and future projections.
3. Quality Assurance:
   - Regularly calibrate and maintain monitoring equipment to ensure accuracy.
   - Conduct quality control checks to identify and rectify data anomalies.
   - Establish protocols for data validation and verification.
4. Public Awareness:
   - Organize public seminars, webinars, or workshops to educate the community on air quality issues.
   - Promote actions that individuals can take to improve air quality, such as reducing emissions and using air purifiers.
   - Utilize social media and public information campaigns.
5. Technology Upgrades:
   - Explore advanced monitoring technologies like remote sensors and

real-time data transmission.

- Consider integrating machine learning or AI for predictive modeling and anomaly detection.
- Evaluate the cost-effectiveness of potential upgrades.

6. Regulatory Compliance:
   - Ensure that the project aligns with local, state, and federal environmental regulations.
   - Maintain records of compliance efforts and any required permits.
7. Research and Innovation:
   - Collaborate with researchers to stay updated on the latest developments in air quality monitoring.
   - Investigate emerging pollutants and health impacts.
   - Seek funding for innovative projects or pilot studies.
8. Collaboration:
   - Engage with environmental agencies, universities, and community organizations to share data and insights.
   - Seek partnerships for joint research efforts or funding opportunities.
   - Foster relationships with experts in the field for guidance.
9. Sustainability:
   - Develop a sustainability plan to secure funding and resources for ongoing monitoring.
   - Consider transitioning to community-led monitoring programs.
   - Explore revenue generation opportunities through data services or consultancy.

Each of these aspects is crucial for the success of Phase 4 in your air quality monitoring project.

PHASE:05

1.Data Collection and Analysis:

Focus on collecting real-time data from monitoring stations and sensors.

Ensure data accuracy and consistency.
2.Quality Assurance:
Implement quality control measures to identify and rectify data anomalies or Certainly, here are some key points to consider for Phase 5 of an air quality monitoring project:
3. Data Visualization:
Develop effective data visualization tools and dashboards to make the information accessible and understandable for stakeholders.
4. Report Generation:
Automate the generation of periodic reports summarizing air quality trends, pollutant levels, and any notable events.
5. Integration with Public Information:
Consider sharing data with the public through websites or apps to raise awareness and enable individuals to make informed decisions.
6.Data Security:
Ensure the security and privacy of collected data to protect against unauthorized access.
7.Calibration and Maintenance:
Regularly calibrate and maintain monitoring equipment to ensure data accuracy.
8:Alerting System:
Implement an alerting system that notifies relevant authorities and the public when air quality reaches critical levels.
9. Regulatory Compliance:
Stay up-to-date with air quality regulations and ensure compliance with relevant standards.
10. Stakeholder Engagement:
Engage with local government, environmental organizations, and the community to gather feedback and improve the monitoring system.
11. Research and Development:
Explore new technologies and methodologies to enhance the effectiveness and efficiency of air quality monitoring.
12. Data Storage and Backup:
Establish robust data storage and backup systems to prevent data loss and ensure continuity of the project.
13. Community Outreach:
Educate the public about the importance of air quality monitoring and involve them in the project.
14. Budget Management:

Carefully manage the budget to cover operational costs, maintenance, and potential upgrades.

15. Future Planning:

Plan for the future, considering expansion, potential partnerships, and the integration of emerging technologies.