DOCUMENTATION(PHASE-05)

AIR QUALITY MONITORING SYSTEM

PROBLEM STATEMENT :

In this rapidly developing environment, the breathing air is very much important to us. That much important air should be in a sufficient pure level, it must rich in the purity level. But ,nowadays the air concentration is becoming very harmful to us. By the harmful gases are merging into the atmospheric air, it's not good? Right!. So we are emerging the solution for this environment.

Who has this problem? :

All living Oraganism suffer from serious disease relating to respiration and lungs.

Why should this problem be solved? :

To save human life and to control haxardous air pollution that is increasing day by day.

How will we know this problem has been solved? :

With the help of the monitoring air pollutants with real-time data, can take preventive measeres which gradually help in controlling the air pollution.

Background Information :

- The model is using the VGA Monitor interfacing with microcontroller for directly displaying the monitored data, also helps to replace bulky and costly displays.
- The increase in popullation and infrastructures alongside increases the pollution wgich can severlygarm the living being and the environment.
- On daily news relating to increased air pollution and irsharmfull effect made a matter of concern for our research. The following references is also been used:

- Daily NEWS
- Prohest mentor experience
- Internet

THE STEPS INVOLVING ARE AS FOLLOWS:

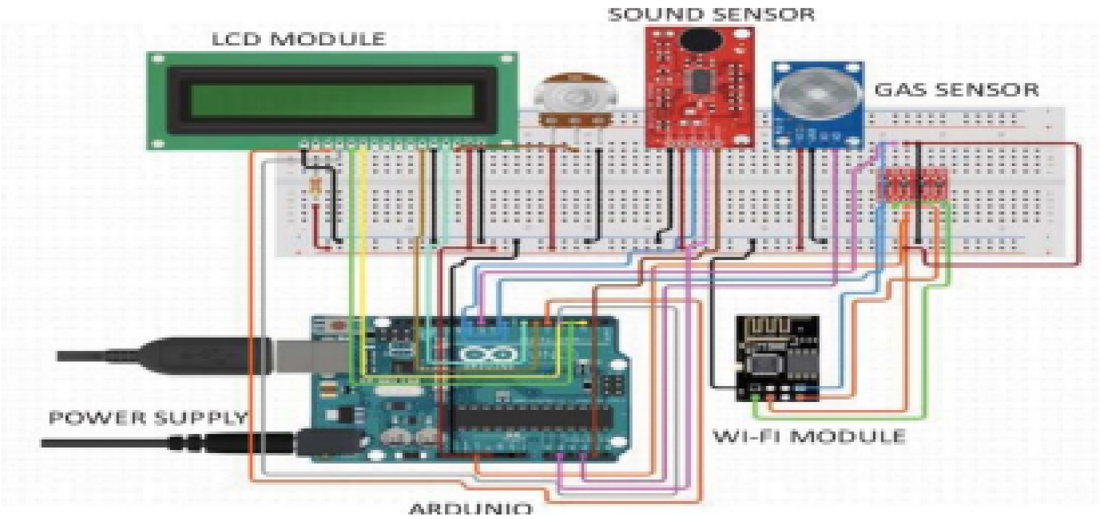
Step 1: WORKABLE SOLUTIONS

- This model requires microcontroller and gaseous sensors.
- The final project model will take a form of portable device for plug-n-play at any VGA display monitor.
- This device need to be mounted over vehicles for monitoring wide area and providing a real time data.

Step 2: THE PROTOTYPE

- ❖ Choose the best workable solution and create a plan to bulid a prototype.
- ❖ What materials will you use for your prototype?
  The prototype be a model and also be further worked on to make more compact.

Air quality monitoring device

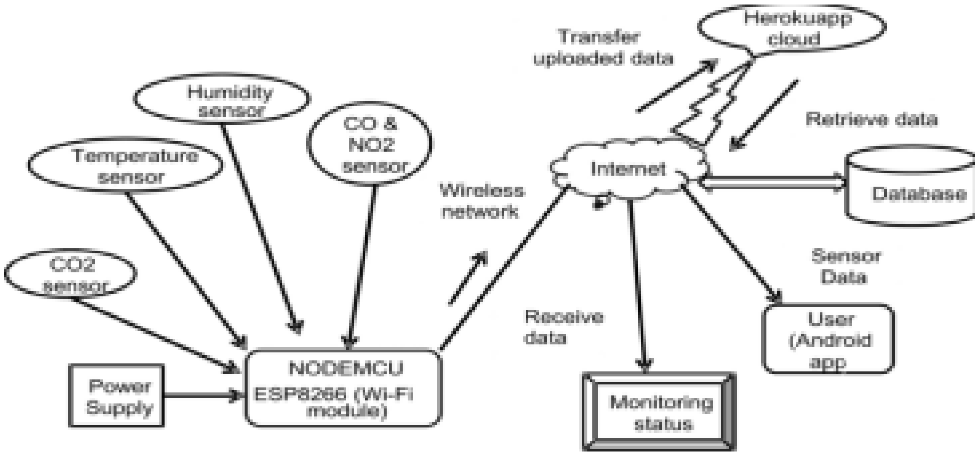


AIR POLLUTION MONITORING

Step 3: SENSORS DETAILS

SENSOR	**SPECIFICATION**
DHT22	TEMPERATURE , HUMIDITY
MQ-7	CARBON MONOXIDE
MQ-135	AIR QUALITY SENSOR
DS3231	DATE AND TIME SENSOR
SDS021	PARTICULATE MATTER

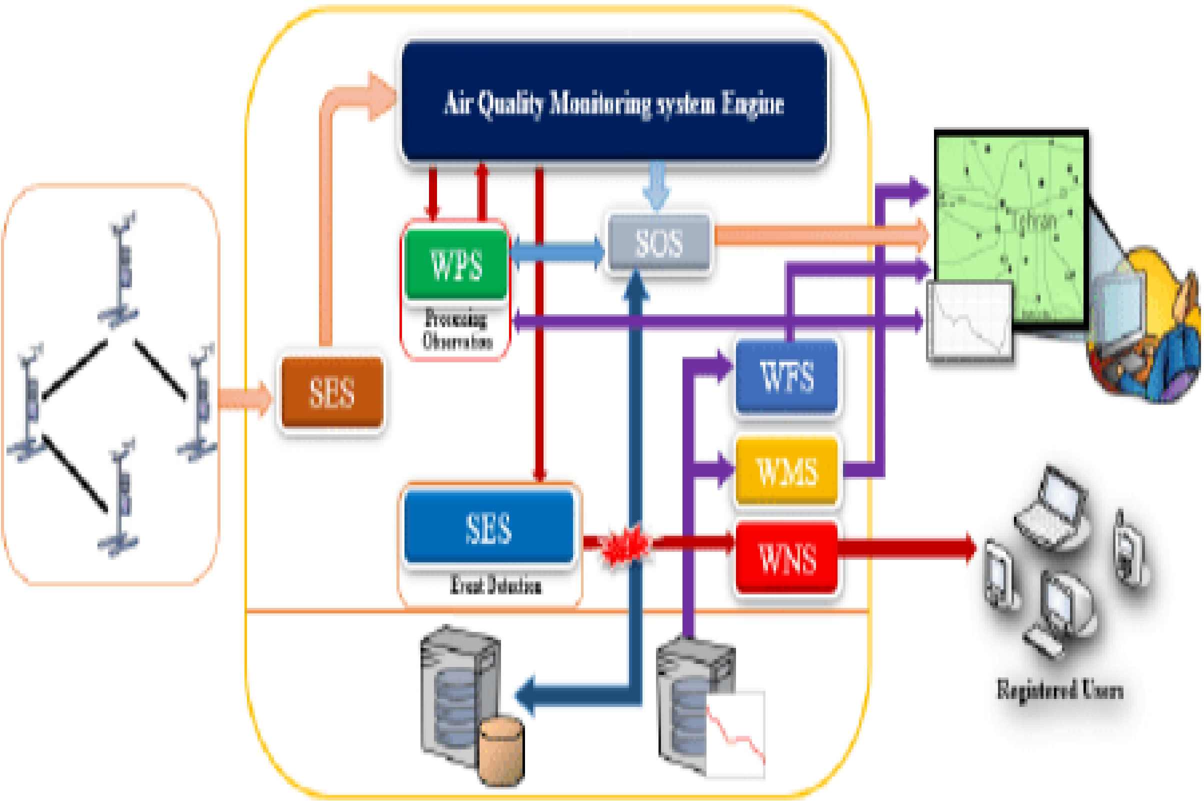
Step 4: WORKING



Step 5 : TESTING THE PROTOTYPE

- **What worked? What did not work ? why?**
- **What materials need to be changed and/or kept? Why?**
- **Did the size of your prototype five you enough information to move forward with a final product? Why or why not ?**
- **Did you have other people test your design and give you feedback? Why or why not?**

Step 6: HOW?



Step 7: HOW TO USE

Basically the data stored over cloud will be accessible to all as we are prociding a web sevice to them from which they can gather any information related to air pollutants in their vicinity.

Step 8: REDESIGNING THE PROTOTYPE

- **Use the data from your testing to redesign with…**
  - **Different materials**
  - **Different builds**
  - **Different sizes**

FINAL RESULT :

The paper presents a detailed survey on the tools and techniques employed in existing smart water quality and air pollution monitoring systems. Also, a low cost, less complex water quality monitoring system is proposed. The implementation enables sensor to provide online data to consumers. The experimental setup can be improved by incorporating algorithms for

anomaly detections in environment.

- ✔ **This proposed Air and water quality early warning system is hoped to play a key role in future for its accuracy and effectiveness,**
- ✔ **This system mainly consists of two parts: prediction model and evaluation model.**
- ✔ **The results expecting that the proposed model has the best accuracy and stability compared to the other existing models**

DESIGNING AN IOT-BASED AIR QUALITY MONITORING SYSTEM

Abstract:

The air quality in an area very influential on the state of the population in an area because of the quality can affect both the health of humans, animals or plants. Therefore, it is necessary to periodically monitor air quality conditions in an area. In this study an IoT-based air quality monitoring system was designed to determine the air quality conditions in an area. The system will monitor using sensors to see the levels of several substances in the aur including 03, SO2, CO and particulates. Reading sensor data using an Arduino microcontroller. Then the data sent to the ThingSpeak cloud system uses a WIFI module on Arduino by accessing the API provided by the ThingSpeak cloud service. The monitoring results will be visible through a web page provided by the Thing Speak cloud service.

1. Introduction

The air quality in an area very influential on the state of the population in an area because of the quality can affect both the health of humans, animals or plants[1-3]. Poor air quality can impact on health is not good for all living organisms. To prevent something that can worsen health conditions, it is necessary to know the condition of air quality in an area. Therefore, it is necessary to periodically monitor air quality conditions in an area.

Air quality monitoring can be done by measuring several parameters including temperature, humidity and air-contained compounds including 03 (Ozone), SO2 (Sulfur Dioxide), CO (Carbon Monoxide) and particulates [4–6]. At this time, advances in sensor technology can help us to measure the proficiency level parameters to determine the condition of the air quality[7,8]. In addition, the development of IoT (Internet of things) also helps with remote monitoring technology[9,10]. In this study we design an IoT- based monitoring system to monitor air quality in an area on a regular basis. Monitoring is done by collecting data from several sensors that can be accessed remotely. This research was conducted based on several previous existing studies. Previously there were studies that conducted control and monitoring of air quality in the room[11,12]. This research is also based on our research in designing remote communication for monitoring air quality[13]. In contrast to previous studies we designed an outdoor air quality monitoring system. The system will measure several compounds in the air including 03, SO2, CO and particles. Air quality will be monitored remotely from web pages.

2. Method

2.1. System overview

General overview designed system can be seen in the block diagram in Figure 1. Figure 1 shows a system designed to collect air quality data from various sensors that will measure the levels of 03. SO2, CO and particles in the air. The measurement results of these sensors will be read by the Arduino Uno controller. Arduino Uno is an ATmega328 based microcontroller board which is widely used for automatic control and monitoring[14,15,16]. Arduino Uno has 14 digital input/output pins (of which 6 can be used as PWM output), 6 analog inputs, 16 MHz ceramic resonator, USB connection, power jack, ICSP header, and reset button[17,18].

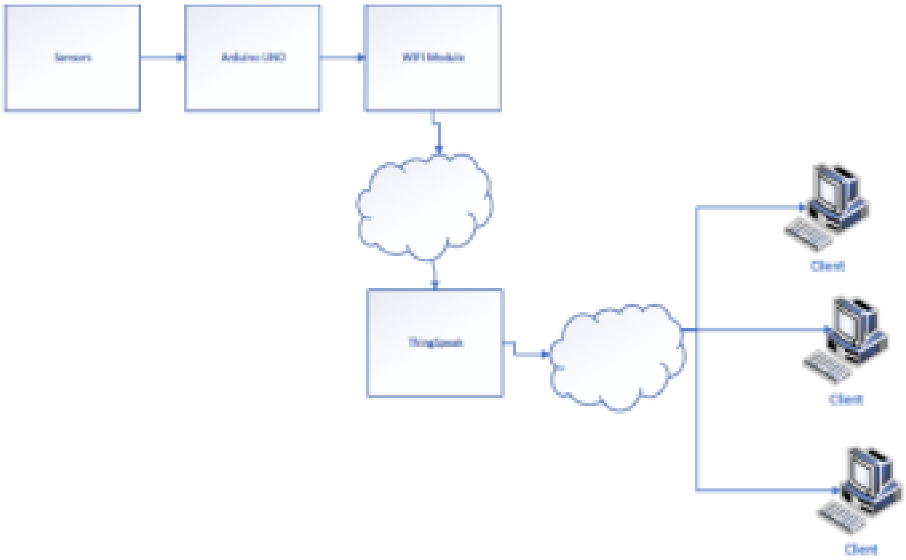


Figure 1. Block diagram of the system

Furthermore, Arduino Uno will send data to the cloud system using WIFI module. We use ThingSpeak for cloud systems. ThingSpeak is an open source IoT application platform and API for storing and retrieving data from things using the HTTP protocol over the Internet or through a Local Area Network[19]. The monitoring results can be seen on the web page provided by ThingSpeak in graphical form.

2.2. The design of the system

The first thing to do in system design is to make a schematic range as shown in Figure 2. We use 4 types of sensors to measure several types that are in the air, namely MQ-7, MQ1-131, MQ-135 and Pm10. Then do the coding on Arduino Uno using Arduino IDE. Arduino IDE is an editor used to write programs to Arduino [20]. Furthermore, create a channel in ThingSpeak to collect measurement data sent by the Arduino. Data transmission from Arduino is done by accessing the ThingSpeak API URL obtained after we create a channel to get data. Arduino accesses the ThingSpeak API using the WIFI ESP8266 module[21].

The MQ-7 sensor is used to read the levels of CO compounds in the air. MQ-7 is an analog sensor. The features of the MQ7 gas sensor are having high sensitivity to CO, stable, and long life[22]. This sensor uses heater power supply. SV AC/DC and uses circuit power supply: SVDC, measuring distance: 20- 2000 ppm to be able to measure carbon monoxide gas. In Figure 2 MQ-7 is connected to the A2 analog pin on Arduino.

To measure Ozone levels, we use MQ-131 which is an analog sensor to measure Ozone levels in the air[23]. MQ-131 works with a power supply of 5V (VCC) which is connected to a VCC pin on the microcontroller. If the sensor detects 03 gas in the air, the output voltage on the sensor will increase, so that the gas concentration will decrease and a deoxidation process will occur. 03 gas concentration value is obtained by calculating the ratio between the sensor resistance value and sensor resistance when the air is clean. In Figure 2 the MQ-131 is connected to the A0 analog pin on Arduino.

SO2 levels in the air are measured using the MQ-135 sensor which is connected to an analog Al pin on Arduino. MQ-135 sensor is a gas sensor whose working principle is to detect the quality of air in the room so that the characteristics of the sensor as a resistor will turn into a semiconductor or voltage conductor[24] Besides SO2 MQ-135 can also be used to measure ammonia, aromatics, benzene vapor, and other gases.

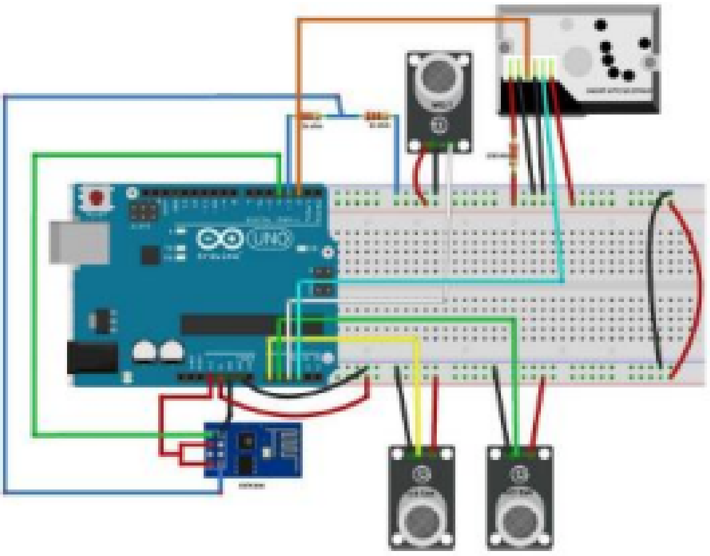


Figure 2. Schematic circuit

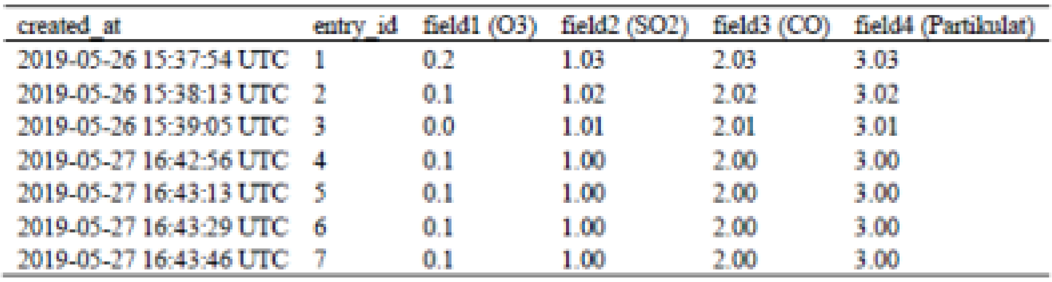
In addition to measuring the levels of some gases, we also measure particles in the air using PM10. PM10 sensor is a sensor that can measure airbome particles smaller than 10 microns (micrometres) [25]. In the schematic of the circuit in Figure 2 the Pm10 sensor is connected to an A3 analog pin on Arduino.

3. Results and discussion

From the results of the design that has been done, a system trial has been carried out that has been designed. The trial was carried out in the campus environment to monitor air quality conditions on the campus. From the results of the trial monitored data of compounds in the air that were measured were data on levels of Ozone, SO2, CO and PM10 particles. Data is received by ThingSpeak every one minute. Data received by Thing Speak can be monitored on the channel pages provided. On the web page data is presented in graphical form. Web pages can be accessed from the internet network by accessing the monitoring channel web pages as shown in Figure 3. The monitoring results can also be exported to table forms in the excel, xml or Json file format.

Examples of export data in tables can be seen in Table 1.

Table 1. Data from monitoring



created_at	entry_id	field1 (O3)	field2 (SO2)	field3 (CO)	field4 (Partikulat)
2019-05-26 15:37:54 UTC	1	0.2	1.03	2.03	3.03
2019-05-26 15:38:13 UTC	2	0.1	1.02	2.02	3.02
2019-05-26 15:39:05 UTC	3	0.0	1.01	2.01	3.01
2019-05-27 16:42:56 UTC	4	0.1	1.00	2.00	3.00
2019-05-27 16:43:13 UTC	5	0.1	1.00	2.00	3.00
2019-05-27 16:43:29 UTC	6	0.1	1.00	2.00	3.00
2019-05-27 16:43:46 UTC	7	0.1	1.00	2.00	3.00

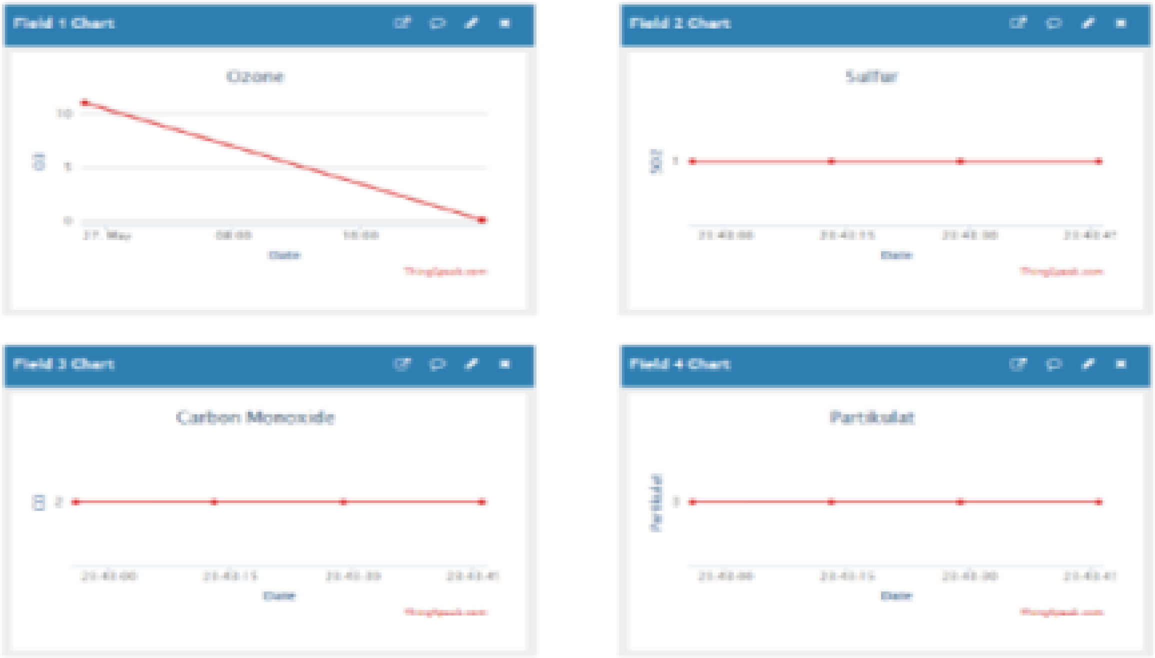


Figure 3. Graph data from monitoring

4. Conclusion

From the trials conducted, the results of the design pan work well. It seems that the data obtained from the sensor can be seen from the webpage provided. Data changes can be monitored every once a minute. From the results obtained, further development can be done by making a mobile-based client application

Data can also be exported into Excel data format, XML or in Json

This is supported because ThingSpeak can also provide API access to get data that has been collected

on the server.

DEVELOPMENT PART OF AIR QUALITY MONITORING SYSTEM

Overview

The IoT-based air pollution monitoring system provides several benefits over traditional air pollution monitoring systems. It can collect real-time data from multiple locations, which then analyzed to identify the sources of pollution. It helps to take necessary measures to reduce it.

The system can also alert the users if the air quality reaches a dangerous level, allowing them to take precautions to protect themselves.

Outdoor air pollution is a major environmental health problem affecting everyone in low-, middle-, and high-income countries.

Ambient (outdoor) air pollution in both cities and rural areas was estimated to cause 4.2 million premature deaths worldwide per year in 2019; this mortality is due to exposure to fine particulate matter, which causes cardiovascular and respiratory disease, and cancers.

WHO estimates that in 2019, some 37% of outdoor air pollution-related premature deaths were due to ischaemic heart disease and stroke, 18% and 23% of deaths were due to chronic obstructive pulmonary disease and acute lower respiratory infections respectively, and 11% of deaths were due to cancer within the respiratory tract.

People living in low- and middle-income countries disproportionately experience the burden of outdoor air pollution with 89% (of the 4.2 million premature deaths) occurring in these areas. The greatest burden is found in the WHO South-East Asia and Western Pacific Regions. The latest burden estimates reflect the significant role air pollution plays in cardiovascular illness and death.

Policies reducing air pollution

Addressing air pollution, which is the second highest risk factor for noncommunicable diseases, is key to protecting public health.

Most sources of outdoor air pollution are well beyond the control of individuals and this demands concerted action by local, national and regional level policy-makers working in sectors like energy, transport, waste management, urban planning and agriculture.

There are many examples of successful policies that reduce air pollution:

- **for industry:** clean technologies that reduce industrial smokestack emissions; improved management of urban and agricultural waste, including capture of methane gas emitted from waste sites as an alternative to incineration (for use as biogas);
- **for energy:** ensuring access to affordable clean household energy solutions for cooking, heating and lighting;
- **for transport:** shifting to clean modes of power generation; prioritizing rapid urban transit,

walking and cycling networks in cities as well as rail interurban freight and passenger travel; shifting to cleaner heavy-duty diesel vehicles and low-emissions vehicles and fuels, including fuels with reduced sulfur content;

- **for urban planning:** improving the energy efficiency of buildings and making cities more green and compact, and thus energy efficient;
- **for power generation:** increased use of low-emissions fuels and renewable combustion-free power sources (like solar, wind or hydropower); co-generation of heat and power; and distributed energy generation (e.g. mini-grids and rooftop solar power generation);
- **for municipal and agricultural waste management:** strategies for waste reduction, waste separation, recycling and reuse or waste reprocessing, as well as improved methods of biological waste management such as anaerobic waste digestion to produce biogas, are feasible, low-cost alternatives to the open incineration of solid waste – where incineration is unavoidable, then combustion technologies with strict emission controls are critical; and
- **for health-care activities:** putting health services on a low-carbon development path can support more resilient and cost-efficient service delivery, along with reduced environmental health risks for patients, health workers and the community. In supporting climate friendly policies, the health sector can display public leadership while also improving health service delivery.

Pollutants

Particulate matter (PM)
PM is a common proxy indicator for air pollution. There is strong evidence for the negative health impacts associated with exposure to this pollutant. The major components of PM are sulfates, nitrates, ammonia, sodium chloride, black carbon, mineral dust and water.

Carbon monoxide (CO)
Carbon monoxide is a colourless, odourless and tasteless toxic gas produced by the incomplete combustion of carbonaceous fuels such as wood, petrol, charcoal, natural gas and kerosene.

Ozone (O_3)
Ozone at ground level – not to be confused with the ozone layer in the upper atmosphere – is one of the major constituents of photochemical smog and it is formed through the reaction with gases in the presence of sunlight.

Nitrogen dioxide (NO_2)
NO_2 is a gas that is commonly released from the combustion of fuels in the transportation and industrial sectors.

Sulfur dioxide (SO_2)
SO_2 is a colourless gas with a sharp odour. It is produced from the burning of fossil fuels (coal and oil) and the smelting of mineral ores that contain sulfur.

Air quality guidelines

The WHO Global air quality guidelines (AQG) offer global guidance on thresholds and limits for key air pollutants that pose health risks. These guidelines are of a high methodological quality and are developed through a transparent, evidence-based decision-making process. In addition to the guideline values, the *WHO Global air quality guidelines* provide interim targets to promote a gradual shift from high to lower concentrations.

The guidelines also offer qualitative statements on good practices for the management of certain types of particulate matter (PM), for example black carbon/elemental carbon, ultrafine particles, and particles originating from sand and dust storms, for which there is insufficient quantitative evidence to derive AQG levels.

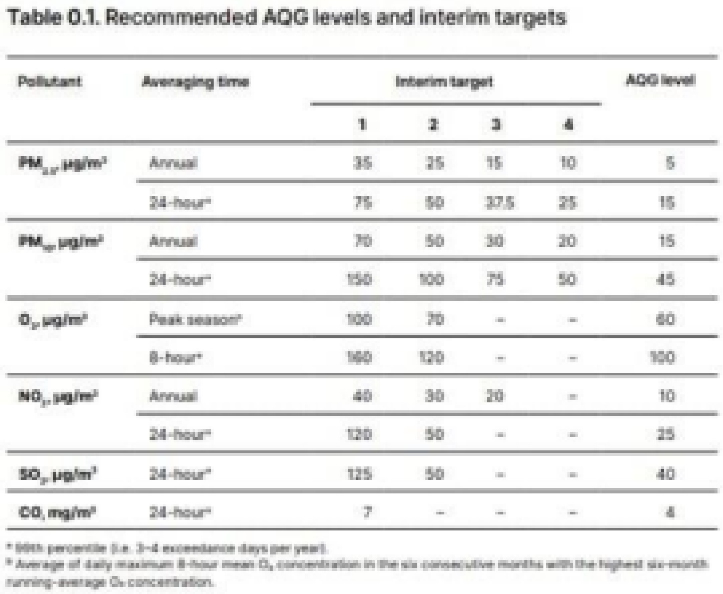


Table 0.1. Recommended AQG levels and interim targets

Pollutant	Averaging time	Interim target				AQG level
		1	2	3	4	
$PM_{2.5}$, µg/m^3	Annual	35	25	15	10	5
	24-hour[a]	75	50	37.5	25	15
PM_{10}, µg/m^3	Annual	70	50	30	20	15
	24-hour[a]	150	100	75	50	45
O_3, µg/m^3	Peak season[b]	100	70	–	–	60
	8-hour[a]	160	120	–	–	100
NO_2, µg/m^3	Annual	40	30	20	–	10
	24-hour[a]	120	50	–	–	25
SO_2, µg/m^3	24-hour[a]	125	50	–	–	40
CO, mg/m^3	24-hour[a]	7	–	–	–	4

[a] 99th percentile (i.e. 3–4 exceedance days per year).
[b] Average of daily maximum 8-hour mean O_3 concentration in the six consecutive months with the highest six-month running-average O_3 concentration.

HARDWARE REQUIREMENTS

IoT-based air pollution monitoring systems comprise several components that work together to collect and analyze air quality data. The components include:

1. **Sensors**: Sensors are the primary components of IoT-based air pollution monitoring systems. They measure various air quality parameters such as particulate matter, carbon monoxide, sulfur dioxide, and nitrogen oxides. The sensors can be classified into two categories: physical and chemical sensors. Physical sensors measure parameters such as temperature, humidity, and pressure, while chemical sensors measure air pollutants.
2. **Microcontroller**: The microcontroller is the brain of IoT-based air pollution monitoring systems. It receives data from the sensors, processes it, and sends it to the cloud server. The microcontroller is usually a microprocessor such as Arduino, Raspberry Pi, or similar devices.
3. **Communication Module**: The communication module is responsible for transmitting data from the microcontroller to the cloud server. Communication modules can use various wireless technologies such as Wi-Fi, Bluetooth, or cellular networks.
4. **Cloud Server**: The cloud server is a centralized platform for storing, analyzing, and sharing air quality data. It collects data from the communication module and stores it in a database. The cloud server also provides web and mobile applications for users to access the data.
5. **Power Supply**: IoT-based air pollution monitoring systems require a power supply to operate. In case of permanent installations external power supply is provided and batteries are provided for portable devices.
6. **Enclosure**: The enclosure is the outer covering that protects the components from environmental factors such as dust, water, and temperature.

SOFTWARE REQUIREMENTS:

1. **IoT Platform:** Choose an IoT platform for data storage and analysis. Common choices include

AWS IoT, Azure IoT, Google Cloud IoT, or open-source platforms like MQTT with Node- RED.

2. **Python Script:** Develop a Python script to read sensor data, process it, and send it to the loT platform. This script will run on the microcontroller.

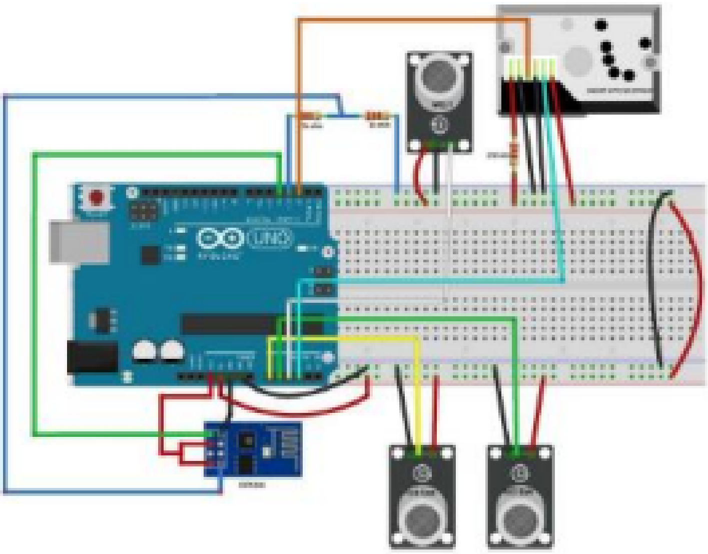
Building the loT System:

1. **Connect the Sensors:** Wire the flow rate and pressure sensors to the microcontroller. Ensure you have the necessary libraries or drivers to interface with these sensors. Sensor datasheets and online resources can guide you through this process.

2. **Set Up the Microcontroller**. Install the necessary software (e.g., Raspbian for Raspberry Pi, Arduino IDE for Arduino, MicroPython for ESP8266/ESP32) on your microcontroller. Write the code to read data from the sensors.

3. **IoT Platform Setup:**

- Choose anloT platform and create an account.
-Set up a new loT device or thing in the platform's console.
-Note down the credentials and connection details provided by the platform (eg, device ID, access tokens, endpoints)



4. Python Script Development:

Here's a simplified example of a Python script to read sensor data and send it to an lot platform. This example uses MQTT as the communication protocol. Make sure to adapt it to your specific sensors and platform.

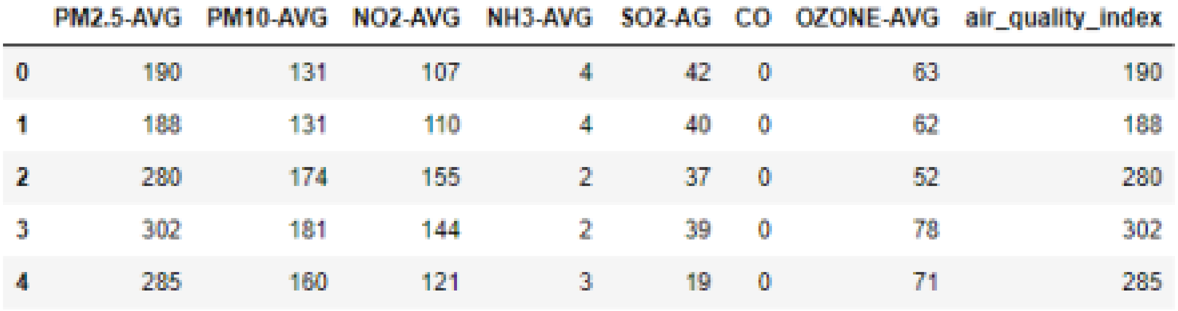
PYTHON CODE :

```
# importing pandas module for data frame
import pandas as pd

# loading dataset and storing in train variable
train=pd.read_csv('AQI.csv')

# display top 5 data
train.head()
```

Output:



	PM2.5-AVG	PM10-AVG	NO2-AVG	NH3-AVG	SO2-AG	CO	OZONE-AVG	air_quality_index
0	190	131	107	4	42	0	63	190
1	188	131	110	4	40	0	62	188
2	280	174	155	2	37	0	52	280
3	302	181	144	2	39	0	78	302
4	285	160	121	3	19	0	71	285

```
# importing Randomforest
fromsklearn.ensemble import AdaBoostRegressor
fromsklearn.ensemble import RandomForestRegressor

# creating model
m1 = RandomForestRegressor()

# separating class label and other attributes
train1 = train.drop(['air_quality_index'], axis=1)
target = train['air_quality_index']

# Fitting the model
m1.fit(train1, target)
'''RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                                                  max_depth=None, max_features='auto',
max_leaf_nodes=None,
                                                  max_samples=None, min_impurity_decrease=0.0,
                                                  min_impurity_split=None, min_samples_leaf=1,
                                                   min_samples_split=2, min_weight_fraction_leaf=0.0,
                                                  n_estimators=100, n_jobs=None, oob_score=False,
                                                  random_state=None, verbose=0, warm_start=False)'''

# calculating the score and the score is 97.96360799890066%
m1.score(train1, target) * 100

# predicting the model with other values (testing the data)
# so AQI is 123.71
m1.predict([[123, 45, 67, 34, 5, 0, 23]])

# Adaboost model
# importing module
# defining model
m2 = AdaBoostRegressor()

# Fitting the model
m2.fit(train1, target)

'''AdaBoostRegressor(base_estimator=None, learning_rate=1.0, loss='linear',
                                    n_estimators=50, random_state=None)'''

# calculating the score and the score is 96.15377360010211%
```

```
m2.score(train1, target)*100
# predicting the model with other values (testing the data)
# so AQI is 94.42105263
m2.predict([[123, 45, 67, 34, 5, 0, 23]])
```

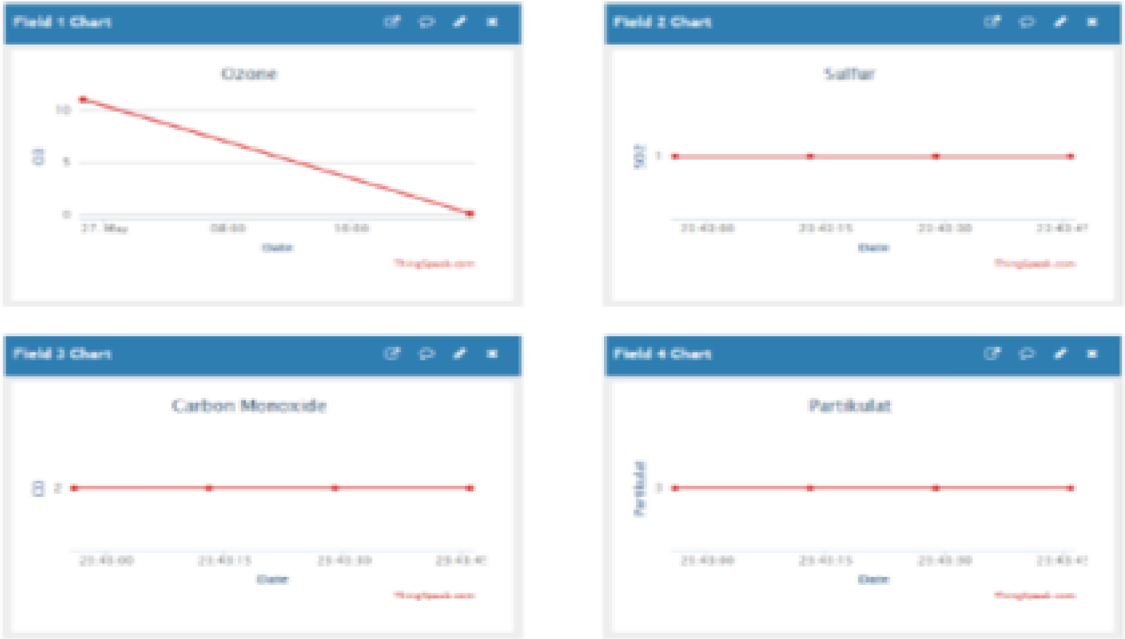


Figure 3. Graph data from monitoring

5. Data Visualization and Analysis:

Set up dashboards or analysis tools on your chosen loT platform to monitor and analyze the data collected from the water fountains.

6. Power and Maintenance:

Ensure a reliable power source and periodic maintenance for the loT sensors to keep the systern running smoothly

CONCLUSION :

In conclusion, an IoT-based air pollution monitoring system is a revolutionary solution that can provide accurate and real-time data about the air quality in a particular area. It can help identify the sources of pollution and take necessary measures to reduce it, protecting the environment and human health.

From the trials conducted, the results of the design pan work well. It seems that the data obtained from the sensor can be seen from the webpage provided. Data changes can be monitored every once a minute. From the results obtained, further development can be done by making a mobile-based client applicationData can also be exported into Excel data format, XML or in Json This is supported because ThingSpeak can also provide API access to get data that has been collectedon the server.

DEVELOPMENT PART – 2 OF AIR QUALITY MONITORING SYSTEM

Overview

The IoT-based air pollution monitoring system provides several benefits over traditional air pollution monitoring systems. It can collect real-time data from multiple locations, which then analyzed to identify the sources of pollution. It helps to take necessary measures to reduce it.

The system can also alert the users if the air quality reaches a dangerous level, allowing them to take precautions to protect themselves.

Outdoor air pollution is a major environmental health problem affecting everyone in low-, middle-, and high-income countries.

Ambient (outdoor) air pollution in both cities and rural areas was estimated to cause 4.2 million premature deaths worldwide per year in 2019; this mortality is due to exposure to fine particulate matter, which causes cardiovascular and respiratory disease, and cancers.

WHO estimates that in 2019, some 37% of outdoor air pollution-related premature deaths were due to ischaemic heart disease and stroke, 18% and 23% of deaths were due to chronic obstructive pulmonary disease and acute lower respiratory infections respectively, and 11% of deaths were due to cancer within the respiratory tract.

People living in low- and middle-income countries disproportionately experience the burden of outdoor air pollution with 89% (of the 4.2 million premature deaths) occurring in these areas. The greatest burden is found in the WHO South-East Asia and Western Pacific Regions. The latest burden estimates reflect the significant role air pollution plays in cardiovascular illness and death.

Policies reducing air pollution

Addressing air pollution, which is the second highest risk factor for noncommunicable diseases, is key to protecting public health.

Most sources of outdoor air pollution are well beyond the control of individuals and this demands concerted action by local, national and regional level policy-makers working in sectors like energy, transport, waste management, urban planning and agriculture.

There are many examples of successful policies that reduce air pollution:

- **for industry:** clean technologies that reduce industrial smokestack emissions; improved management of urban and agricultural waste, including capture of methane gas emitted from waste sites as an alternative to incineration (for use as biogas);
- **for energy:** ensuring access to affordable clean household energy solutions for cooking, heating and lighting;
- **for transport:** shifting to clean modes of power generation; prioritizing rapid urban transit, walking and cycling networks in cities as well as rail interurban freight and passenger travel; shifting to cleaner heavy-duty diesel vehicles and low-emissions vehicles and fuels, including fuels with reduced sulfur content;
- **for urban planning:** improving the energy efficiency of buildings and making cities more green and compact, and thus energy efficient;
- **for power generation:** increased use of low-emissions fuels and renewable combustion-free power sources (like solar, wind or hydropower); co-generation of heat and power; and distributed energy generation (e.g. mini-grids and rooftop solar power generation);
- **for municipal and agricultural waste management:** strategies for waste reduction, waste separation, recycling and reuse or waste reprocessing, as well as improved methods of biological waste management such as anaerobic waste digestion to produce biogas, are

feasible, low-cost alternatives to the open incineration of solid waste – where incineration is unavoidable, then combustion technologies with strict emission controls are critical; and

- **for health-care activities:** putting health services on a low-carbon development path can support more resilient and cost-efficient service delivery, along with reduced environmental health risks for patients, health workers and the community. In supporting climate friendly policies, the health sector can display public leadership while also improving health service delivery.

Pollutants

Particulate matter (PM)
PM is a common proxy indicator for air pollution. There is strong evidence for the negative health impacts associated with exposure to this pollutant. The major components of PM are sulfates, nitrates, ammonia, sodium chloride, black carbon, mineral dust and water.

Carbon monoxide (CO)
Carbon monoxide is a colourless, odourless and tasteless toxic gas produced by the incomplete combustion of carbonaceous fuels such as wood, petrol, charcoal, natural gas and kerosene.

Ozone (O_3)
Ozone at ground level – not to be confused with the ozone layer in the upper atmosphere – is one of the major constituents of photochemical smog and it is formed through the reaction with gases in the presence of sunlight.

Nitrogen dioxide (NO_2)
NO_2 is a gas that is commonly released from the combustion of fuels in the transportation and industrial sectors.

Sulfur dioxide (SO_2)
SO_2 is a colourless gas with a sharp odour. It is produced from the burning of fossil fuels (coal and oil) and the smelting of mineral ores that contain sulfur.

Air quality guidelines

The WHO Global air quality guidelines (AQG) offer global guidance on thresholds and limits for key air pollutants that pose health risks. These guidelines are of a high methodological quality and are developed through a transparent, evidence-based decision-making process. In addition to the guideline values, the *WHO Global air quality guidelines* provide interim targets to promote a gradual shift from high to lower concentrations.

The guidelines also offer qualitative statements on good practices for the management of certain types of particulate matter (PM), for example black carbon/elemental carbon, ultrafine particles, and particles originating from sand and dust storms, for which there is insufficient quantitative evidence to derive AQG levels.

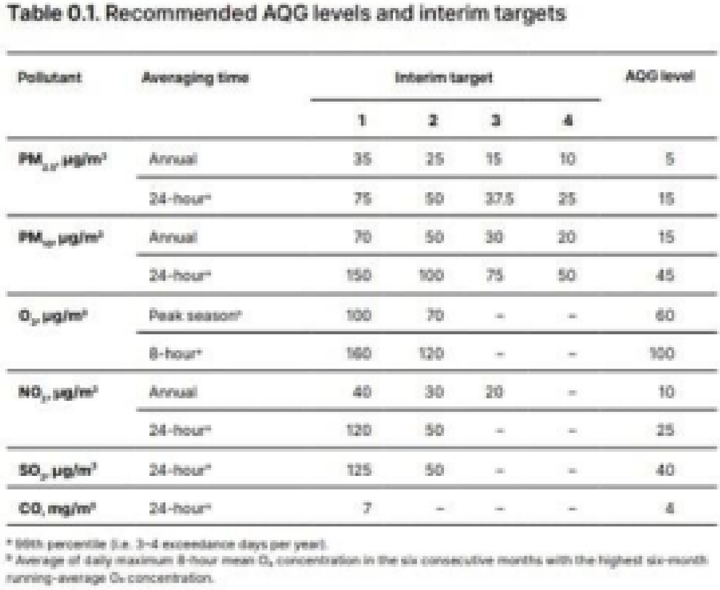


Table 0.1. Recommended AQG levels and interim targets

Pollutant	Averaging time	Interim target				AQG level
		1	2	3	4	
$PM_{2.5}$, µg/m^3	Annual	35	25	15	10	5
	24-hour^a	75	50	37.5	25	15
PM_{10}, µg/m^3	Annual	70	50	30	20	15
	24-hour^a	150	100	75	50	45
O_3, µg/m^3	Peak season^b	100	70	–	–	60
	8-hour^a	160	120	–	–	100
NO_2, µg/m^3	Annual	40	30	20	–	10
	24-hour^a	120	50	–	–	25
SO_2, µg/m^3	24-hour^a	125	50	–	–	40
CO, mg/m^3	24-hour^a	7	–	–	–	4

[a] 99th percentile (i.e. 3–4 exceedance days per year).
[b] Average of daily maximum 8-hour mean O_3 concentration in the six consecutive months with the highest six-month running-average O_3 concentration.

HARDWARE REQUIREMENTS

IoT-based air pollution monitoring systems comprise several components that work together to collect and analyze air quality data. The components include:

7. **Sensors**: Sensors are the primary components of IoT-based air pollution monitoring systems. They measure various air quality parameters such as particulate matter, carbon monoxide, sulfur dioxide, and nitrogen oxides. The sensors can be classified into two categories: physical and chemical sensors. Physical sensors measure parameters such as temperature, humidity, and pressure, while chemical sensors measure air pollutants.
8. **Microcontroller**: The microcontroller is the brain of IoT-based air pollution monitoring systems. It receives data from the sensors, processes it, and sends it to the cloud server. The microcontroller is usually a microprocessor such as Arduino, Raspberry Pi, or similar devices.
9. **Communication Module**: The communication module is responsible for transmitting data from the microcontroller to the cloud server. Communication modules can use various wireless technologies such as Wi-Fi, Bluetooth, or cellular networks.
10. **Cloud Server**: The cloud server is a centralized platform for storing, analyzing, and sharing air quality data. It collects data from the communication module and stores it in a database. The cloud server also provides web and mobile applications for users to access the data.
11. **Power Supply**: IoT-based air pollution monitoring systems require a power supply to operate. In case of permanent installations external power supply is provided and batteries are provided for portable devices.
12. **Enclosure**: The enclosure is the outer covering that protects the components from environmental factors such as dust, water, and temperature.

SOFTWARE REQUIREMENTS:

2. **IoT Platform:** Choose an IoT platform for data storage and analysis. Common choices include AWS IoT, Azure IoT, Google Cloud IoT, or open-source platforms like MQTT with Node- RED.

2. **Python Script:** Develop a Python script to read sensor data, process it, and send it to the loT platform. This script will run on the microcontroller.

Building the loT System:

2. **Connect the Sensors**
3. **Set Up the Microcontroller**
4. **IoT Platform Setup**
5. **Python Script Development:**

WORKING ON WOKWI SIMULATOR:
The coding part for simulating the air quality monitoring system in the wokwi platform. CODE:-

```
#include <DHT.h>
#include <WiFi.h>
#include <ThingSpeak.h>
#include "DHTesp.h"

#define DHT_PIN 2 // Replace with the GPIO pin connected to the DHT22 sensor #define
LED_GREEN_PIN 21 // Replace with the GPIO pin connected to the green LED bulb #define
LED_RED_PIN 22 // Replace with the GPIO pin connected to the red LED bulb

char ssid[] = "Wokwi-GUEST";
char pass[] = "";
WiFiClient client;

unsigned long myChannelNumber = 2308799;
const char *myWriteAPIKey = "Y5D386LU3W5X66Y2";
int statusCode;
DHTesp dhtSensor;

int ledGreen = LED_GREEN_PIN;
int ledRed = LED_RED_PIN;

struct Data {
 float temperature;
 float humidity;
};

Data data; // Declare a variable to store the data

void setup() {
 Serial.begin(115200);
 WiFi.mode(WIFI_STA);
 ThingSpeak.begin(client);
 dhtSensor.setup(DHT_PIN, DHTesp::DHT22);

 pinMode(ledGreen, OUTPUT);
 pinMode(ledRed, OUTPUT);
 }

void connectToCloud() {
 if (WiFi.status() != WL_CONNECTED) {
```

```
 Serial.print("Attempting to connect");
 while (WiFi.status() != WL_CONNECTED) {
 WiFi.begin(ssid, pass);
 for (int i = 0; i < 5; i++) {
 Serial.print(".");
 delay(1000);
 }
 }
 Serial.println("\nConnected.");
 }
 }

void computeData() {
 TempAndHumidity sensorData = dhtSensor.getTempAndHumidity();
data.temperature = sensorData.temperature;
 data.humidity = sensorData.humidity;
 Serial.println("-----------");
 Serial.println("Humi: " + String(data.humidity));
 Serial.println("Temp: " + String(data.temperature));
 Serial.println("-----------");
 }
```

LIBRARIES WHICH INCLUDED IN THE SIMULATOR,

```
# Wokwi Library List

# Automatically added based on includes:
DHT sensor library
LiquidCrystal I2C

Blynk
ThingSpeak
DHT sensor library for ESPx
```

FOR THE DIAGRAM,

```
 {
 "version": 1,
 "author": "M.induja",
 "editor": "wokwi",
 "parts": [
 { "type": "wokwi-breadboard-half", "id": "bb1", "top": 93, "left": 185.2, "attrs": {} },
{ "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -43.3, "left": 43, "attrs": {} },  {
 "type": "wokwi-dht22",
 "id": "dht1",
 "top": -33.6,
 "left": -113.7,
 "rotate": 270,
 "attrs": {}
 },
 {
 "type": "wokwi-lcd1602",
 "id": "lcd1",
 "top": -60.8,
 "left": 245.6,
 "attrs": { "pins": "i2c" }
 },
 { "type": "wokwi-vcc", "id": "vcc1", "top": 77.56, "left": 595.2, "attrs": {} },
{ "type": "wokwi-gnd", "id": "gnd1", "top": 115.2, "left": 575.4, "attrs": {} }
],
 "connections": [
 [ "esp:TX0", "$serialMonitor:RX", "", [] ],
 [ "esp:RX0", "$serialMonitor:TX", "", [] ],
 [ "dht1:VCC", "bb1:tp.5", "red", [ "h38.4", "v134.4", "h259.2" ] ],  [
"dht1:GND", "bb1:tn.4", "black", [ "h48", "v153.6", "h230.4" ] ],  [
"dht1:SDA", "esp:D2", "green", [ "h19.2", "v-76.9", "h172.8", "v134.4" ] ],  [
"esp:VIN", "bb1:tp.3", "red", [ "h-9.6", "v38.4", "h153.6" ] ],
 [ "esp:GND.1", "bb1:tn.1", "black", [ "h0" ] ],
 [ "lcd1:GND", "bb1:tn.14", "black", [ "h-9.6", "v-38.4", "h316.8", "v182.4" ] ],  [
"lcd1:VCC", "bb1:tp.15", "red", [ "h-19.2", "v-57.5", "h336", "v182.4", "h-192" ] ],  [
"bb1:tp.25", "vcc1:VCC", "green", [ "v0" ] ],
 [ "bb1:tn.25", "gnd1:GND", "green", [ "v0" ] ]
 ],
 "dependencies": {}
 }
```

CONCLUSION :

In conclusion, an IoT-based air pollution monitoring system is a revolutionary solution that can provide accurate and real-time data about the air quality in a particular area. It can help identify the sources of pollution and take necessary measures to reduce it, protecting the environment and human health.

From the trials conducted, the results of the design pan work well. It seems that the data obtained from the sensor can be seen from the webpage provided. Data changes can be monitored every once a minute. From the results obtained, further development can be done by making a mobile-based client application Data can also be exported into Excel data format, XML or in Json This is supported because ThingSpeak can also provide API access to get data that has been collectedon the server.

SUBMITTED

BY

TEAM MEMBERS

DHARANI. R(au821121104012)
ASHVITHA.D (autk23cs01)
BHARATH.P.A (au821121104009)
MURUGESHWARI.A(au821121104037)
SHARMIKA.R(au821121104051)

AARTH S (au821121104003)