**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](https://www.who.int/data/gho/data/themes/air-pollution/total-burden-of-disease-from-household-and-ambient-air-pollution), 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 (O3)**  
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.

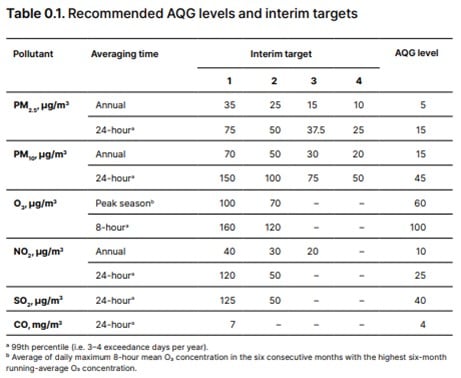
**Nitrogen dioxide (NO2)**  
NO2 is a gas that is commonly released from the combustion of fuels in the transportation and industrial sectors.

**Sulfur dioxide (SO2)**SO2 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](https://www.who.int/publications-detail-redirect/9789240034228)(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.



**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**
2. **Set Up the Microcontroller**
3. **IoT Platform Setup**
4. **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" ] ],

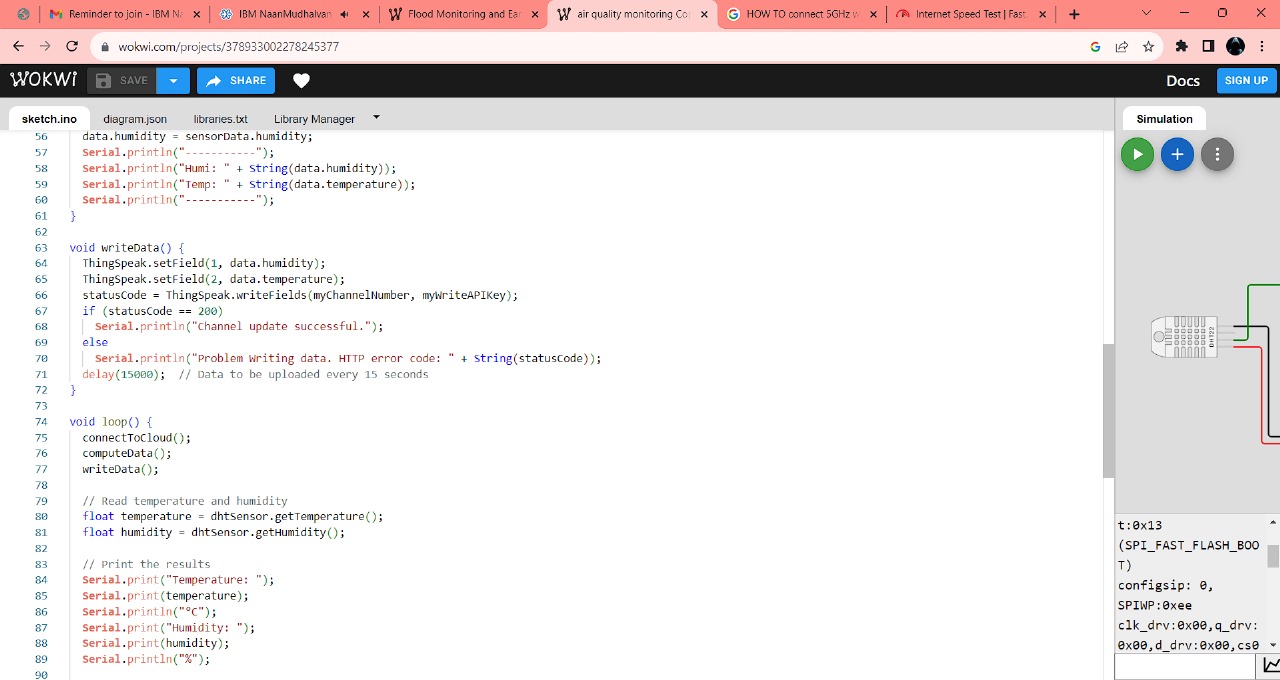
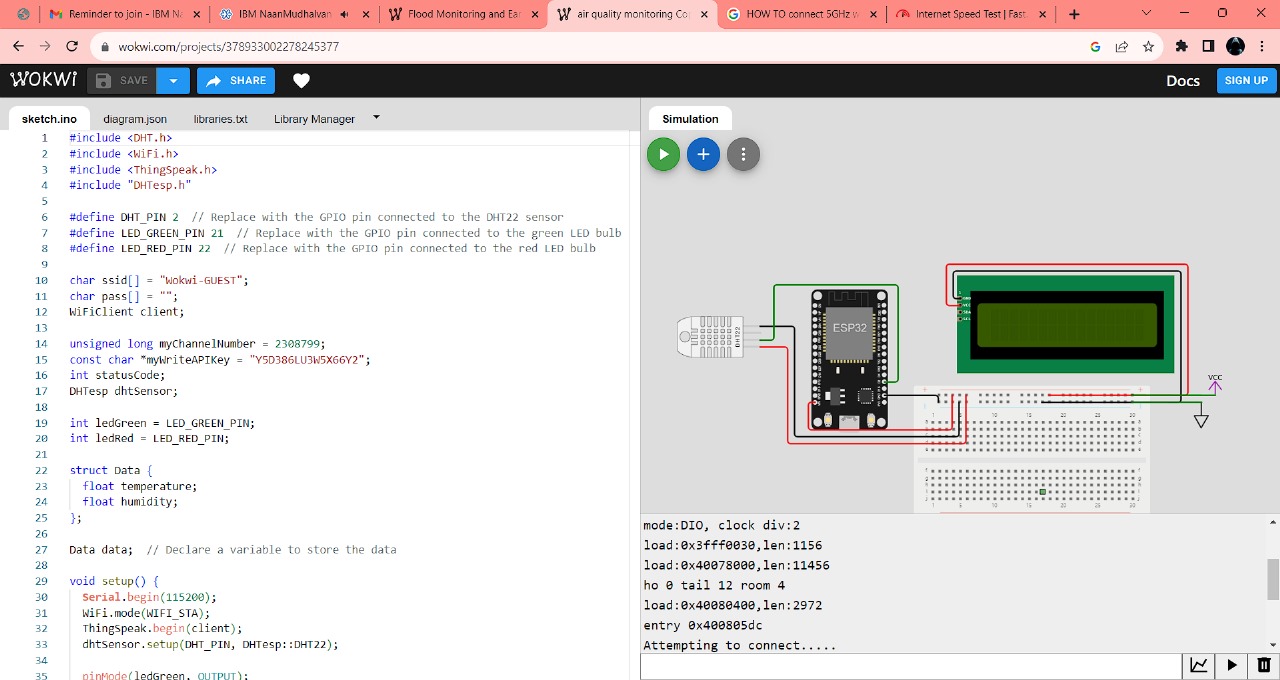
[ "bb1:tp.25", "vcc1:VCC", "green", [ "v0" ] ],

[ "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.

GITHUB LINK : https://github.com/Saravanankodi/Naan-mudhalvan-IOT-group-4-Team-6.

*SUBMITTED*

*BY*

*TEAM MEMBERS*

|  |
| --- |
| SARAVANAN.K(au821121104048) |
| MURUGANANTHAM.P(au821121104036) |
| MOHEMMED ASICK.A(au8211211032) |
| MANIBHARATHI.V.S(au821121104029) |
| GOVINDHAVASAN.K(au821121104301) |