# AIR QUALITY MONITORING

## PHASE 4: DEVELOPMENT PART 2

## CONTINUE BUILDING THE PROJECT BY DEVELOPING THE DATA – SHARING PLATFORM.


**PROJECT: AIR QUALITY MONITORING**

## INTRODUCTION:

Air is a basic requirement for the survival and development of all lives on Earth. It affects healthand influences the development of the economy. Today, due to the development of industrialization,the increase in the number of private cars, and the burning of fossil fuels, air quality is decreasing, withincreasingly serious air pollution. There are many pollutants in the atmosphere, such as SO2,

$NO_2$,$CO_2$, NO, CO, $NO_x$, PM2.5, and PM10. Internationally, a large number of scholars have conductedresearch on air pollution and air quality forecasts, concentrating on the forecasting of contaminants.Air pollution affects the life of a society, and even endangers the survival of mankind. During theIndustrial Revolution, there was a dramatic increase in coal use by factories and households, andthe smog caused significant morbidity and mortality, particularly when combined with stagnantatmospheric conditions. During the Great London Smog of 1952, heavy pollution for 5 days caused atleast 4000 deaths [1,2]. This episode highlighted the relationship between air pollution and humanhealth, yet air pollution continues to be a growing problem in cities and households around the world.

Air pollution is made up of a mixture of gases and particles in harmful amounts that are releasedinto the atmosphere due to either natural or human activities [3]. The sources of pollutants can bedivided into two categories:

1) Natural sources
2) Anthropogenic sources

(1) Natural sources:

Natural pollution sources are natural phenomena that discharge harmful substances or haveharmful effects on the environment. Natural phenomena, such as volcanic eruptions and forest fires,will result in air pollutants, including $SO_2$, $CO_2$, $NO_2$, CO, and sulfate.

(2) Anthropogenic (man-made) sources:

Man-made sources such as the burning of fuels, discharges from industrial production processes,and transportation emissions are the main sources of air pollution. There are many kinds of pollutantsemitted by man-made pollution sources, including hydrogen, oxygen,nitrogen, sulfur, metal compounds,and particulate matter.With the increasing world population and the developing world economy, the demand for energyin the world has increased dramatically. The large-scale use of fossil energy globally has also ledto a series of environmental problems that have received much attention due to their detrimentaleffects on human health and the environment [3–5]. Air pollution is a fundamental problem in manyparts of the world, with two important concerns: the impact on human health, such as cardiovasculardiseases, and the impact on the environment, such as acid rain, climate change, and global warming.

Air pollution is one of the biggest threats to the present-day environment. Everyone is beingaffected by air pollution day by day including humans, animals, crops, cities, forests and aquaticecosystems. Besides that, it should be controlled at a certain level to prevent the increasing rate ofglobal warming. This project aims to design an IOT-based air pollution monitoring systemusing the internet from anywhere using a computer or mobile to monitor the air quality of thesurroundings and environment. There are various methods and instruments available for themeasurement and monitoring quality of air. The IoT-based air pollution monitoring systemwould not only help us to monitor the air quality but also be able to send alert signals whenever the air quality deteriorates and goes down beyond a certain level.In this system, NodeMCU plays the main controlling role. It has been programmed in a manner,such that, it senses the sensory signals from the sensors and shows the quality level via ledindicators. Besides the harmful gases (such as CO2, CO, smoke, etc) temperature and humidity canbe monitored through the temperature and humidity sensor by this system. Sensor responses are fedto the NodeMCU which displays the monitored data in the ThingSpeak cloud which can be utilizedfor analyzing the air quality of that area.The following simple flow diagramindicates the working mechanism of the IoT-based Air Pollution Monitoring System.

## IOT BASED AIR QUALITY MONITORING SYSTEM

Air is getting polluted because of the release of toxic gases by industries, vehicle emissions and increased concentration of harmful gases and particulate matter in theatmosphere.

The level of pollution is increasing rapidly due to factors like industries, urbanization,increase in population, vehicle use which can affect human health. Particulate matter isone of the most important parameters having a significant contribution to the increase inair pollution. This creates a need for measurement and analysis of real-time air qualitymonitoring so that appropriate decisions can be taken in a timely period.This paper presents real-time standalone air quality monitoring. Internet of Things (IoT)inowadays finding profound use in each and every sector, plays a key role in our air qualitymonitoring system too. The setup will show the air quality in PPM on the webpage so thatwe can monitor it very easily.

## AIM OF THE PROJECT:

Air is getting polluted because of the release of toxic gases by industries, vehicleemissions and increased concentration of harmful gases and particulate matter in theatmosphere.The level of pollution is increasing rapidly due to factors like industries, urbanization,increase in population, vehicle use which can affect human health. Particulate matter isone of the most important parameters having a significant contribution to the increase inair pollution. This creates a need for measurement and analysis of real-time air qualitymonitoring so that appropriate decisions can be taken in a timely period.This paper presents real-time standalone air quality monitoring. Internet of Things (IoT)isnowadays finding profound use in each and every sector, plays a key role in our air qualitymonitoring system too. The setup will show the air quality in PPM on the webpage so thatwe can monitor it very easily.

In this IoT project, we can monitor the pollution level from anywhere using your computer
or mobile.

## Components Used:

### Hardware Components

1. NodeMCU V3
2. DHT11 Sensor Module
3. MQ-135 Gas Sensor Module
4. Veroboard(KS100)
5. Breadboard
6. Connecting Wires
7. AC-DC Adapters
8. LEDs emitting green, yellow and red colours
9. Resistors

### SOFTWARE COMPONENTS:

1. ThinkSpeak Cloud
2. Arduino IDE

2.3 Brief Description of the Components

### NodeMCU V3:

NodeMCU V3 is an open-source ESP8266 development kit, armed with the CH340G USBTTLSerial chip. It has firmware that runs on ESP8266 Wi-Fi SoC from Espressif Systems.Whilst cheaper, CH340 is super reliable even in industrial applications. It is tested to bestable on all supported platforms as well. It can be simply coded in Arduino IDE. It has avery low current consumption between 15 μA to 400 mA.

The pinout Diagram of NodeMC3

## Pinout Diagram of NodeMCU V3

### MQ-135 Gas Sensor Module:

The material of MQ135 is $SnO_2$, it is a special material: when exposed to clean air, it ishardly being conducted, however, when put in an environment with combustible gas, it has apretty performance of conductivity. Just make a simple electronic circuit, and convert thechange of conductivity to a corresponding output signal. MQ135 gas sensor is sensitive toAmmonia, Sulphide, Benzene steam, smoke and other harmful gases. Used for family,surrounding environment noxious gas detection device, apply to ammonia, aromatics,sulphur, benzene vapor, and other harmful gases/smoke, gas detection, tested concentrationrange: 10 to 1000ppm. In a normal environment, the environment which doesn't havedetected gas set the sensor's output voltage as the reference voltage, the analog outputvoltage will be about 1V, when the sensor detects gas, harmful gas concentration increases by20ppm per voltage increase by 0.1V.

### Veroboard (KS100):

Veroboard is the original prototyping board.Sometimes referred to as 'stripboard' or 'matrix board' these offer total flexibility for hard wiringdiscrete components. Manufactured from acopper clad laminate board or Epoxy basedsubstrate, it is offered in both single and doublesided formats. Vero boards are available in a wide range of board sizes and in both imperialand metric pitch – Veroboard is an ideal base forcircuit construction and offers even greater adaptability using our range of terminal pins andassemblies. As with other stripboards, in usingVeroboard, components are suitably positioned and soldered to the conductors to form therequired circuit. Breaks can be made in the tracks, usually around holes, to divide the stripsinto multiple electrical nodes enabling increased circuit complexity. This type of wiringboard may be used for initial electronic circuit development, to construct prototypes forbench testing or in the production of complete electronic units in small quantities.

## AC-DC Power Adapter:

An AC-DC power supply or adapter is an electrical device that obtains electricity from agrid-based power supply and converts it into adifferent current, frequency, and voltage. AC-DCpower supplies are necessary to provide the rightpower that an electrical component needs. The ACDCpower supply delivers electricity to devices thatwould typically run-on batteries or have no other power source.

## LED (Red, Green & Yellow):

A light-emitting diode (LED) is a semiconductor light source that emits light when currentflows through it. Electrons in the semiconductor recombine with electron holes, releasingenergy in the form of photons. The colour of the light (corresponding to the energy of thephotons) is determined by the energy required for electrons to cross the band gap of thesemiconductor. White light is obtained by using multiple semiconductors or a layer of lightemittingphosphor on the semiconductor device. LEDs have many advantages overincandescent light sources, including lower power consumption, longer lifetime, improvedphysical robustness, smaller size, and faster switching. In exchange for these generallyfavourable attributes, disadvantages of LEDsinclude electrical limitations to low voltage andgenerally to DC (not AC) power, inability toprovide steady illumination from a pulsing DCor an AC electrical supply source, and lessermaximum operating temperature and storagetemperature. In contrast to LEDs, incandescentlamps can be made to intrinsically run at virtually any supply voltage, can utilize either AC or DCcurrent interchangeably, and will provide steady illumination when powered by AC orpulsing DC even at a frequency as low as 50 Hz. LEDs usually need electronic supportcomponents to function, while an incandescent bulb can and usually does operate directlyfrom an unregulated DC or AC power source.

## Resistors:

A resistor is a passivetwo-terminalelectricalcomponent that implements electrical resistance asa circuit element. In electronic circuits, resistors areused to reduce current flow, adjust signal levels, todivide voltages, bias active elements, and terminatetransmission lines, among other uses. High-powerresistors that can dissipate many watts of electricalpower as heat may be used as part of motor controls, in power distribution systems, or as testloads for generators. Fixed resistors have resistancesthat only change slightly with temperature, time or operating voltage.

## Arduino IDE:

The Arduino IDE is open-sourcesoftware, which is used to writeand upload code to the Arduinoboards. The IDE application is suitable for different operatingsystems such as Windows, MacOS X, and Linux. It supports theprogramming languages C andC++. Here, IDE stands forIntegrated DevelopmentEnvironment. The program or code written in the Arduino IDE is often called sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software.The sketch is saved with the extension '.ino.'

## ThingSpeak Cloud:

ThingSpeak is open-source software writtenin Ruby which allows users to communicatewith internet-enabled devices. It facilitatesdata access, retrieval and logging of data byproviding an API to both the devices andsocial network websites. ThingSpeak wasoriginally launched by ioBridge in 2010 as aservice in support of IoT applications.ThingSpeak has integrated support from thenumerical computing software MATLABfrom MathWorks, allowing ThingSpeak usersto analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from MathWorks.

# Working Procedures:

NodeMCU plays the main controlling role in this project. It has been programmed in a manner,such that, it senses the sensory signals from the sensors and shows the quality level via ledindicators. The DHT11 sensor module is used to measure the temperature and the humidity of thesurroundings.With the help of the MQ-135 gas sensor module, air quality is measured in ppm.

These data are fed to the ThinkSpeak cloud over the internet. We have also provided LED indicators to indicate the safety levels.

STEP 1. Firstly, the calibration ofthe MQ-135 gas sensor module is done. The sensor is set to preheat for 24 minutes. Then the software code is uploaded to the NodeMCU followed by the hardware circuit to calibrate the sensor has been performed.

STEP 2. Then, the DHT11 sensor is set to preheat for 10 minutes.

STEP 3. The result of calibration found in STEP 1 is used to configure the final working code.

STEP 4. The final working code is then uploaded to the NodeMCU.

STEP 5. Finally, the complete hardware circuit is implemented.
The software codes and the hardware circuits are described in the following chapters

## SOFTWARE CODE for Calibration of MQ135 Sensor:

```
void setup()
{
Serial.begin(9600); //Baud rate
```



```
pinMode(A0,INPUT);
}
void loop()
{
float sensor_volt; //Define variable for sensor voltage
float RS_air; //Define variable for sensor resistance
float R0; //Define variable for R0
float sensorValue=0.0; //Define variable for analog readings
Serial.print("Sensor Reading = ");
Serial.println(analogRead(A0));
for(int x = 0 ; x < 500 ; x++) //Start for loop
{
sensorValue = sensorValue + analogRead(A0); //Add analog values of sensor 500
times
}
sensorValue = sensorValue/500.0; //Take average of readings
sensor_volt = sensorValue*(5.0/1023.0); //Convert average to voltage
RS_air = ((5.0*1.0)/sensor_volt)-1.0; //Calculate RS in fresh air
R0 = RS_air/3.7; //Calculate R0
Serial.print("R0 = "); //Display "R0"
Serial.println(R0); //Display value of R0
delay(1000); //Wait 1 second


}
```

## Execution of the Main Program:

```
#include <ESP8266WiFi.h>
#include <DHT.h>
#include <ThingSpeak.h>
DHT dht(D5, DHT11);
#define LED_GREEN D2
#define LED_YELLOW D3
#define LED_RED D4
#define MQ_135 A0
int ppm=0;
float m = -0.3376; //Slope
float b = 0.7165; //Y-Intercept
float R0 = 3.12; //Sensor Resistance in fresh air from previous code
WiFiClient client;
long myChannelNumber = 123456; // Channel id
```

```
const char myWriteAPIKey[] = "API_Key";
void setup() {
// put your setup code here, to run once:
Serial.begin(9600);
pinMode(LED_GREEN,OUTPUT);
pinMode(LED_YELLOW,OUTPUT);
pinMode(LED_RED,OUTPUT);
pinMode(MQ_135, INPUT);
WiFi.begin("WiFi_Name", "WiFi_Password");
while(WiFi.status() != WL_CONNECTED)
{
delay(200);
Serial.print(".");
}
Serial.println();
Serial.println("NodeMCU is connected!");
Serial.println(WiFi.localIP());
dht.begin();
ThingSpeak.begin(client);
}
void loop() {
float sensor_volt; //Define variable for sensor voltage
float RS_gas; //Define variable for sensor resistance
float ratio; //Define variable for ratio
int sensorValue;//Variable to store the analog values from MQ-135
float h;
float t;
float ppm_log; //Get ppm value in linear scale according to the the ratio value
float ppm; //Convert ppm value to log scale
h = dht.readHumidity();
delay(4000);
t = dht.readTemperature();
delay(4000);
sensorValue = analogRead(gas_sensor); //Read analog values of sensor
sensor_volt = sensorValue*(5.0/1023.0); //Convert analog values to voltage
RS_gas = ((5.0*1.0)/sensor_volt)-1.0; //Get value of RS in a gas
ratio = RS_gas/R0; // Get ratio RS_gas/RS_air
ppm_log = (log10(ratio)-b)/m; //Get ppm value in linear scale according to the ratio
value
ppm = pow(10, ppm_log); //Convert ppm value to log scale
Serial.println("Temperature: " + (String) t);
Serial.println("Humidity: " + (String) h);
```



```
Serial.println("Our desired PPM = "+ (String) ppm);
ThingSpeak.writeField(myChannelNumber, 1, t, myWriteAPIKey);
delay(20000);
ThingSpeak.writeField(myChannelNumber, 2, h, myWriteAPIKey);
delay(20000);
ThingSpeak.writeField(myChannelNumber, 3, ppm, myWriteAPIKey);
```

```
delay(20000);
if(ppm<=100)
{
digitalWrite(LED_GREEN,HIGH);
digitalWrite(LED_YELLOW,LOW);
digitalWrite(LED_RED,LOW);
}
else if(ppm<=200)
{
digitalWrite(LED_GREEN,LOW);
digitalWrite(LED_YELLOW,HIGH);
digitalWrite(LED_RED,LOW);
}
else
{
digitalWrite(LED_GREEN,LOW);
digitalWrite(LED_YELLOW,LOW);
digitalWrite(LED_RED,HIGH);
}
delay(2000);


}
```

## CUNCLUTION AND FUTURE WORK (PHASE 4):

### Project conclution:

In the phase 4 coclution, we wil summarize the key finding and i nsides from

the advanced regression techniques. We will reiterate the impact of these technique s on

improving the Air quality monitoring.

**THE END...**