PROJECT TITLE: AIR QUALITY MONITORING

Phase 5: Project Documentation & Submission

INTRODUCTION:

The fundamental target of IoT air quality Monitoring System is that the Air contamination is a rising issue nowadays. It is obligatory to screen air quality and monitor it for a more beneficial future and solid living for all. Web of things (IoT) is picking up prominence step by step as it can change life making it simpler for people.

In this project we can measure air quality by using *Raspberry pi*, temperature and humidity sensor, gas sensor, dust sensor. Sensors have been used to detect the presence of harmful gases/compounds, which are continually transmitted to a controller. Air quality monitoring and controlling system is proposed in this project, which enable us to monitor and check real time quality or the air temperature, humidity in specific region through IOT . In this project we can also control the quality of air pollution by using air filtering which absorb the carbon in the air and produce a fresh air.

OBJECTIVES:

- Main objective of this project is to get a clean pure air. In this project will be designing a device which can detect air pollution in the environment.
- This IoT devices will be having microcontroller as well as air quality sensor charges mq2 sensor. This IoT device will be continuously monitoring air quality and upload the data to server.

- We will make a air purifier which works on dry air purifier concept. This air purifier will be started once the air quality has been decreased. once the air quality has been restored this air purifier will be switched off.

CODE:

WOKWi SAVE SHARE

sketch.ino | diagram.json | libraries.txt | Library Manager

```
#define BLYNK_TEMPLATE_ID "TMPLwToQUqRw"
#define BLYNK_TEMPLATE_NAME "Air Quality Monitoring"
#define BLYNK_AUTH_TOKEN "C8Y7T0Fr54QF8pdfQ5dZsdfhhSdiQBFLj8mYe"

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

#include <DHT.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

  byte degree_symbol[8] =
                    {
                      0b00111,
                      0b00101,
                      0b00111,
                      0b00000,
                      0b00000,
                      0b00000,
                      0b00000,
                      0b00000
                    };

char auth[] = BLYNK_AUTH_TOKEN;
```

```
29  char ssid[] = "WiFi Username";  // type your wifi name
30  char pass[] = "WiFi Password";  // type your wifi password
31
32  BlynkTimer timer;
33
34  int gas = 32;
35  int sensorThreshold = 100;
36
37  #define DHTPIN 2 //Connect Out pin to D2 in NODE MCU
38  #define DHTTYPE DHT11
39  DHT dht(DHTPIN, DHTTYPE);
40
41
42  void sendSensor()
43  {
44
45
46    float h = dht.readHumidity();
47    float t = dht.readTemperature(); // or dht.readTemperature(true) for
48
49
50      if (isnan(h) || isnan(t)) {
51     Serial.println("Failed to read from DHT sensor!");
52     return;
53    }
54     int analogSensor = analogRead(gas);
55    Blynk.virtualWrite(V2, analogSensor);
```

```
54      int analogSensor = analogRead(gas);
55     Blynk.virtualWrite(V2, analogSensor);
56     Serial.print("Gas Value: ");
57       Serial.println(analogSensor);
58     // You can send any value at any time.
59     // Please don't send more that 10 values per second.
60       Blynk.virtualWrite(V0, t);
61       Blynk.virtualWrite(V1, h);
62
63       Serial.print("Temperature : ");
64       Serial.print(t);
65       Serial.print("    Humidity : ");
66       Serial.println(h);
67
68
69   }
70   void setup()
71   {
72
73    Serial.begin(115200);
74
75    //pinMode(gas, INPUT);
76     Blynk.begin(auth, ssid, pass);
77     dht.begin();
78     timer.setInterval(30000L, sendSensor);
79
80    //Wire.begin();
81      lcd.begin();
```

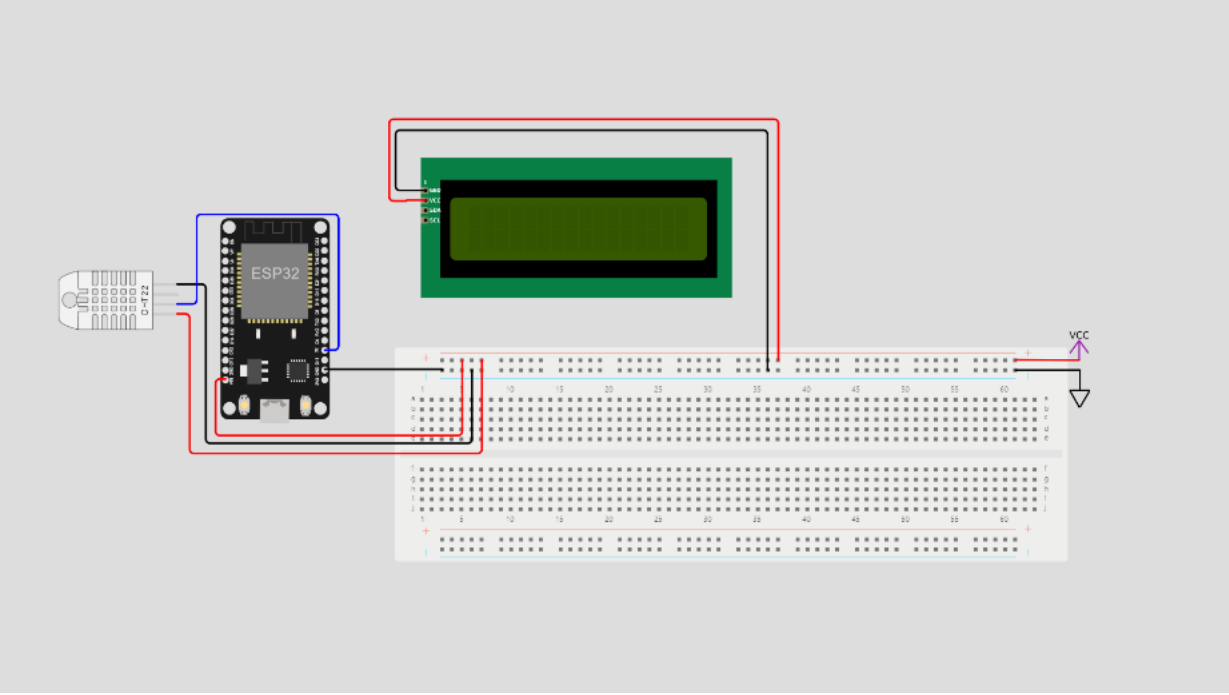
```
78    timer.setInterval(30000L, sendSensor);
79
80   //Wire.begin();
81     lcd.begin();
82
83
84  //  lcd.backlight();
85   // lcd.clear();
86    lcd.setCursor(3,0);
87    lcd.print("Air Quality");
88    lcd.setCursor(3,1);
89    lcd.print("Monitoring");
90    delay(2000);
91    lcd.clear();
92    }
93
94  void loop()
95  {
96    Blynk.run();
97    timer.run();
98   float h = dht.readHumidity();
99    float t = dht.readTemperature(); // or dht.readTemperature(true) for
100     int gasValue = analogRead(gas);
101   lcd.setCursor(0,0);
102   lcd.print("Temperature ");
103   lcd.setCursor(0,1);
104   lcd.print(t);
```

```
lcd.setCursor(0,1);
lcd.write(1);
lcd.createChar(1, degree_symbol);
lcd.setCursor(7,1);
lcd.print("C");
delay(4000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Humidity ");
lcd.print(h);
lcd.print("%");
delay(4000);
lcd.clear();
//lcd.setCursor(0,0);
// lcd.print(gasValue);
// lcd.clear();
Serial.println("Gas Value");
Serial.println(gasValue);
if(gasValue<1200)
{
  lcd.setCursor(0,0);
  lcd.print("Gas Value: ");
  lcd.print(gasValue);
  lcd.setCursor(0, 1);
  lcd.print("Fresh Air");
  Serial.println("Fresh Air");
  delay(4000);
  lcd.clear();
```

```
    lcd.print("Fresh Air");
    Serial.println("Fresh Air");
    delay(4000);
    lcd.clear();
  }
  else if(gasValue>1200)
  {
    lcd.setCursor(0,0);
    lcd.print(gasValue);
    lcd.setCursor(0, 1);
    lcd.print("Bad Air");
    Serial.println("Bad Air");
    delay(4000);
    lcd.clear();
  }

  if(gasValue > 1200){
    //Blynk.email("shameer50@gmail.com", "Alert", "Bad Air!");
    Blynk.logEvent("pollution_alert","Bad Air");
  }
}
```

SETUP:



PROJECT DESCRIPTION:

MQ135 sensor can sense NH3, NOx, alcohol, Benzene, smoke, CO2 and some other gases, so it is perfect gas sensor for our Air Quality Monitoring System. When we will connect it to Arduino then it will sense the gases, and we will get the Pollution level in PPM (parts per million). MQ135 gas sensor gives the output in form of voltage levels and we need to convert it into PPM. So for converting the output in PPM, here we have used a library for MQ135 sensor, it is explained in detail in “Code Explanation” section below.

Sensor was giving us value of 90 when there was no gas near it and the safe level of air quality is 350 PPM and it should not exceed 1000 PPM. When it exceeds the limit of 1000 PPM, then it starts cause Headaches, sleepiness and stagnant, stale, stuffy air and if exceeds beyond 2000 PPM then it can cause increased heart rate and many other diseases. When the value will be less than 1000 PPM, then the LCD and webpage will display “Fresh Air”.

Whenever the value will increase 1000 PPM, then the buzzer will start beeping and the LCD and webpage will display “Poor Air, Open Windows”. If it will increase 2000 then the buzzer will keep beeping and the LCD and webpage will display “Danger! Move to fresh Air”.

The code has been computed successfully. It is user friendly, and had required options, which can be utilized by the user to perform the desired operations The code need to be dumped in the Arduino IDE software. The goals that are achieved by the code.

· Less number of human involvement

· Efficient management of water usage

· Cost effective

PYTHON CODE:

```
from MQ135 import MQ135
import serial
import time
import RPi.GPIO as GPIO
import Adafruit_DHT
import requests

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 4

MQ135_PIN = 17

GPIO.setup(MQ135_PIN, GPIO.IN)

ser = serial.Serial('/dev/ttyS0', 9600, timeout=1)

def sendData(command, timeout, debug):
    ser.write(command.encode('utf-8'))
    if debug:
        print('Sent: ' + command)
    time.sleep(timeout / 1000)
    while ser.inWaiting() > 0:
        if debug:
            print('Response: ' + ser.readline().decode('utf-8').strip())

def readDHT():
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
    return humidity, temperature

def readMQ135():
    return GPIO.input(MQ135_PIN)
```

```
def sendToServer(air_quality):
    url = 'http://your_server_url'
    data = {'air_quality': air_quality}
    response = requests.post(url, data=data)
    print(response.text)

def setup():
    sendData('AT+RST\r\n', 2000, True)
    sendData('AT+CWMODE=2\r\n', 1000, True)
    sendData('AT+CIFSR\r\n', 1000, True)
    sendData('AT+CIPMUX=1\r\n', 1000, True)
    sendData('AT+CIPSERVER=1,80\r\n', 1000, True)

def loop():
    air_quality = readMQ135()
    sendToServer(air_quality)
    time.sleep(1)

if __name__ == '__main__':
    setup()
    while True:
        loop()

webpage = "<h1>IOT Air Pollution Monitoring System</h1>"
webpage += "<p><h2>"
webpage += " Air Quality is "
webpage += str(air_quality)
webpage += " PPM"
webpage += "<p>"
if air_quality <= 1000:
    webpage += "Fresh Air"
elif air_quality <= 2000 and air_quality >= 1000:
    webpage += "Poor Air"
elif air_quality >= 2000:
```

```
    webpage += "Danger! Move to Fresh Air"
webpage += "</h2></p></body>"
cipSend = "AT+CIPSEND="
cipSend += str(connectionId)
cipSend += ","
cipSend += str(len(webpage))
cipSend += "\r\n"
sendData(cipSend, 1000, DEBUG)
sendData(webpage, 1000, DEBUG)
cipSend = "AT+CIPSEND="
cipSend += str(connectionId)
cipSend += ","
cipSend += str(len(webpage))
cipSend += "\r\n"
closeCommand = "AT+CIPCLOSE="
closeCommand += str(connectionId)
closeCommand += "\r\n"
sendData(closeCommand, 3000, DEBUG)

lcd.setCursor(0, 0)
lcd.print("Air Quality is ")
lcd.print(air_quality)
lcd.print(" PPM ")
lcd.setCursor(0, 1)
if air_quality <= 1000:
    lcd.print("Fresh Air")
    digitalWrite(8, LOW)
elif air_quality >= 1000 and air_quality <= 2000:
    lcd.print("Poor Air, Open Windows")
    digitalWrite(8, HIGH)
elif a
lcd.print("Danger! Move to Fresh Air")
digitalWrite(8, HIGH)   # turn the LED on
```

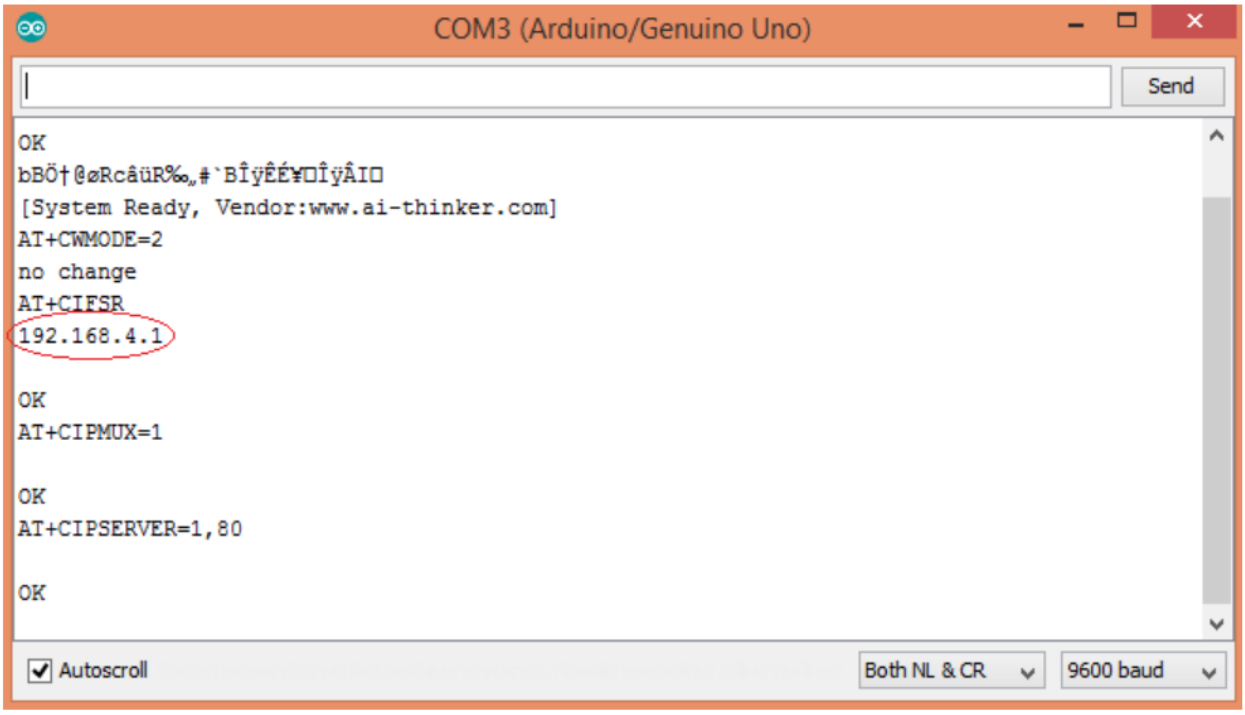
```
lcd.scrollDisplayLeft()
delay(1000)

def sendData(command, timeout, debug):
    response = ""
    esp8266.print(command)  # send the read character to the esp8266
    time = millis()
    while (time + timeout) > millis():
        while esp8266.available():
            # The esp has data so display its output to the serial window
            c = esp8266.read()  # read the next character.
            response += c
    if debug:
        Serial.print(response)
    return response
```

Testing and Output of the Project:

Before uploading the code, make sure that you are connected to the Wi-Fi of your ESP8266 device. After uploading, open the serial monitor and it will show the IP address like shown below.

The critical got the opportunity to screen air quality is very obvious, in light of expanded mechanical exercises over the previous years. Individuals got the chance to perceive the degree that their exercises affect air quality [4]. This undertaking proposes air contamination observing framework. The framework was created utilizing the Arduino microcontroller. The contamination recognition framework was intended to watch and dissect air quality in period and log data to a faraway server, keeping the data refreshed over the net.



Type this IP address in your browser, it will show you the output as shown below. You will have to refresh the page again if you want to see the current Air Quality Value in PPM.

IOT Air Pollution Monitoring System

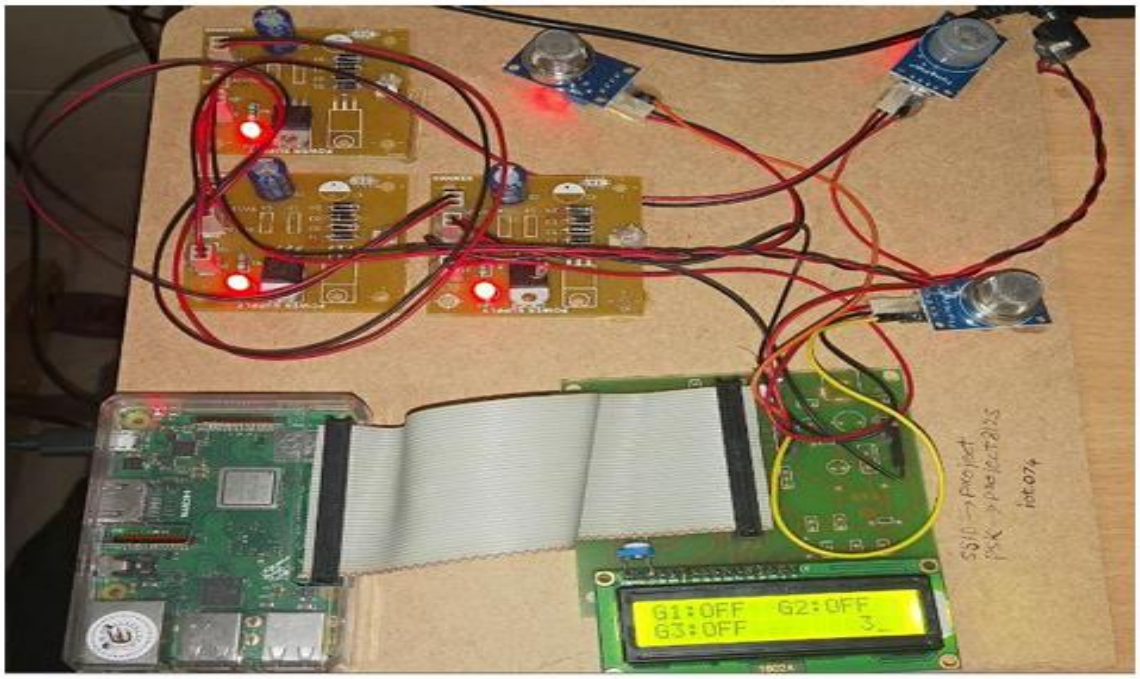
Air Quality is 977 PPM

Good Air

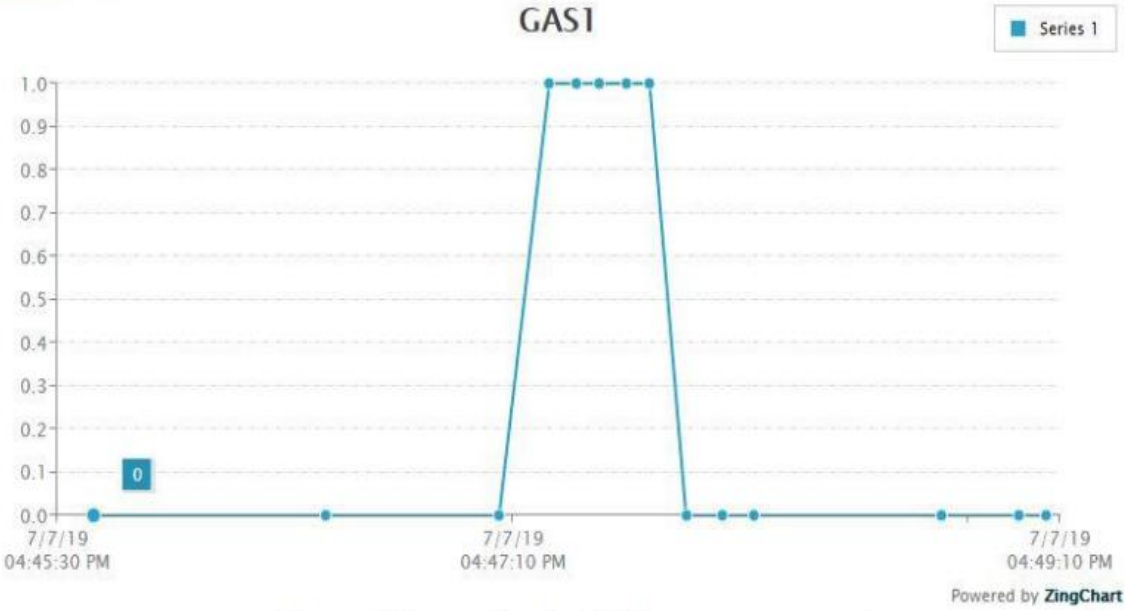
We have setup a local server to demonstrate its working, you can check the **Video** below. But to monitor the air quality from anywhere in the world, you need to **forward the port 80 (used for HTTP or internet) to your local or private IP address** (192.168*) of you device. After port forwarding all the incoming connections will be forwarded to this local address and you can open above shown webpage by just entering the public IP address of your internet from anywhere. You can forward the port by logging into your router (192.168.1.1) and find the option to setup the port forwarding.

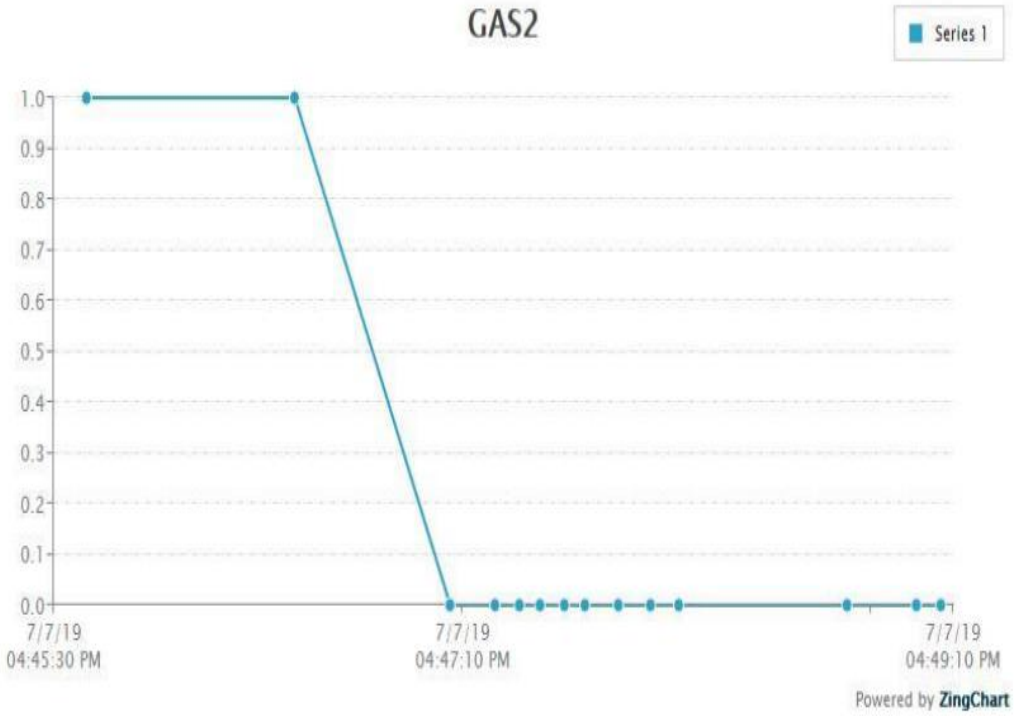
OVERVIEW:

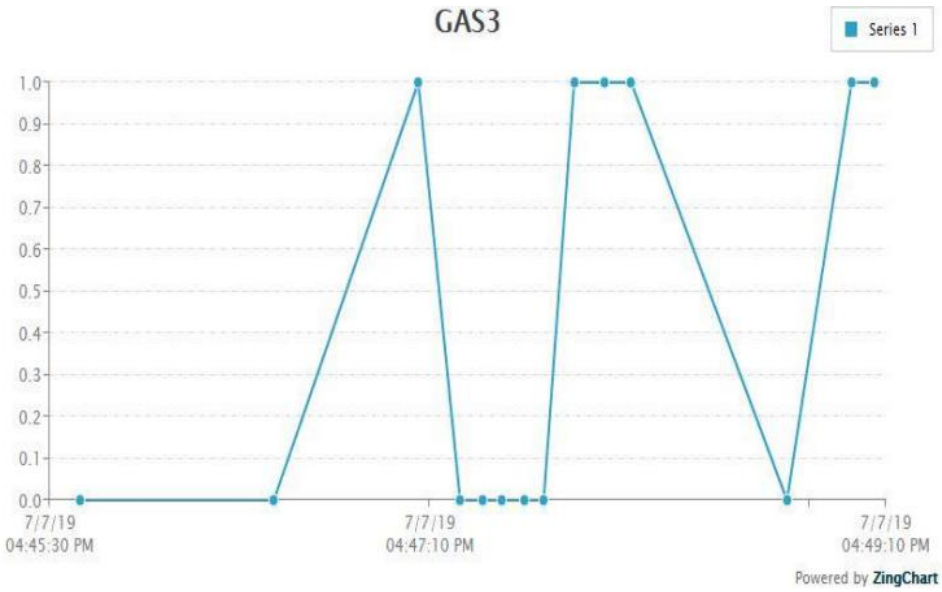
This Shows the complete setup of the Air Quality Monitoring System Based on the Iot Using Raspberry pi that contains Mq-2 Gas sensor , Mq-7 Gas Sensor and Mq-135 Gas Sensor and finally placed on the board for easy to use and Convince



Hello.. iot074
Logout Switch to Table View
Previous Page 4 of 4







APPLICATIONS:

- **Prevent smog dangers:** This is one of the most important reasons why people use the air pollution control.
- **Protect their health:** This is part of the reason why most people install the air pollution control. Most of these chemicals could damage the lungs.
- **Improve their indoor air quality:** People use it to improve the air when they are indoors.

ADVANTAGES:

- **Air pollution control helps to protect the human health:** To every human on earth, health is wealth; health is paramount for the running of our day to day activities, without which we would have to rely on others to live. Air Pollution Scrubber Suppliers come to help protect our health, which is very valuable.

- **Air pollution control helps prevent economic wastes**: With air pollution control, the wastes accrued from dead crops and bad water will be limited or stopped.
- **Increased worker productivity:** No matter how strong the immune system is, there are times when it fails, especially when there is excess air pollution. As pollution is controlled, workers can now work for a longer period of time.
- **Helps improve indoor air quality:** Air pollution control helps to secure the quality of the air inside your house.
- **Prevent smog dangers:** This is one of the most important reasons why people use the air pollution control. Smog can be very hazardous, which is why air pollution control should be installed at an early stage to prevent smog.
- **Protect their health:** This is part of the reason why most people install the air pollution control. Most of these chemicals could damage the lungs.
- **Improve their indoor air quality:** People use it to improve the air when they are indoors.

CONCLUSION:

In this project, the integrated IoT air quality monitoring system is developed to overcome the issues in air quality. The sensors mainly sense the various dangerous gases present in the environment. Safety efforts can be upgraded to secure the information that is being sent through the segments by presenting new conventions.

The air monitoring system can help in the innovation of new practices to overcome the problems of the highly-polluted areas, which is a major issue. It supports the newtechnology and effectively supports the healthy life concept.

REFERENCE LINKS:

-Phase1 link-[IOT_Phase1]

https://github.com/Aishu241203/Naan-Mudhalvan/blob/09635a6d503f8dff8a19cc9f367ab9d586749142/IOT_Phase1.docx

- Phase 2 link –[IOT_Phase2]

https://github.com/Aishu241203/Naan-Mudhalvan/blob/09635a6d503f8dff8a19cc9f367ab9d586749142/IOT_Phase2.docx

- Phase 3 link- [IOT_Phase3]

https://github.com/Aishu241203/Naan-Mudhalvan/blob/09635a6d503f8dff8a19cc9f367ab9d586749142/IOT_Phase3.pdf

- Phase 4 link – [IOT_Phase4]

https://github.com/Aishu241203/Naan-Mudhalvan/blob/09635a6d503f8dff8a19cc9f367ab9d586749142/IOT_Phase4.pdf

-README link:

https://github.com/Aishu241203/Naan-Mudhalvan/blob/main/README.md