DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING SHARDA SCHOOL OF ENGINEERING AND TECHNOLOGY SHARDA UNIVERSITY, GREATER NOIDA

PROJECT TITLE

Indoor Air-Quality Determination using IoT

A project submitted

in partial fulfillment of the requirements for the degree of

Bachelor of Technology in Computer Science and Engineering

by

ABHIJEET SINGH (2019622666)

SIDHANTH SUMAN(2019619997)

VINEET VATS(2019503184)

Supervised by:

Ms. Jyoti Pruthi (Asst. Prof.)

MARCH, 2023

CERTIFICATE

This is to certify that the report entitled **"Air Quality Determination using IOT"** submitted by **"**ABHIJEET SINGH (2019622666), SIDHANTH SUMAN(2019619997) and VINEET VATS (2019503184)**"** to Sharda University, towards the fulfillment of requirements ofthe degree of **"Bachelor of Technology"** is record of bonafide final year Project workcarried out by them in the **"**Department of Computer Science & Engineering, Sharda School of Engineering and Technology, Sharda University**"**.

The results/findings contained in this Project have not been submitted in part or full to any other University/Institute forward of any other Degree/Diploma.

Signature of the Guide
Name: Jyoti Pruthi
Designation: Asst Prof.

Signature of Head of Department

Name: Prof.(Dr.)Nitin Rakesh
Place: Sharda University
Date:

Signature of External Examiner
Date:

ACKNOWLEDGEMENT

A major project is a golden opportunity for learning and self-development. We consider ourselves very lucky and honored to have so many wonderful people lead us through in completion of this project.

First and foremost we would like to thank Dr. Nitin Rakesh, HOD, CSE who gave us an opportunity to undertake this project.

We are grateful to Ms. Jyoti Pruthi for her guidance in our project work. Ms.Jyoti Pruthi, who in spite of being extraordinarily busy with academics, took timeout to hear, guide and keep us on the correct path. We do not know where we would have been without her help.

The CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Name and signature of Students:
ABHIJEET SINGH (2019622666)

SIDHANTH SUMAN(2019619997)

VINEET VATS(2019503184)

ABSTRACT

The rapid growth of urbanization and industrialization has led to a significant increase in air pollution, posing severe threats to human health and the environment. This paper presents an IoT-based air quality monitoring and low-cost air filter system that aims to provide real-time air quality data and purify the surrounding air. The system employs a DHT11 sensor, ESP8266 WiFi module, Arduino microcontroller, MQ135 gas sensor, MQ7 carbon monoxide sensor, and the ThingSpeak platform for data visualization and analysis. The proposed system offers an affordable and efficient solution for monitoring air quality and mitigating the adverse effects of air pollution.

Keywords: IoT, air quality monitoring, low-cost air filter, DHT11, ESP8266, Arduino, MQ135, MQ7, ThingSpeak

SUMMARY

The system deals with monitoring and controlling the environmental conditions like temperature, relative humidity, and CO level present in an area with sensors and sends the information to the web page and then plots the sensor data as graphical statistics.

The main aim of this project is to design and implement an efficient monitoring system through which the required parameters are monitored remotely using the internet and the data gathered from the sensors are stored in the cloud and to project the estimated trend on the web browser.

This project is based on IoT (Internet of Things), which is an emerging field in which all the devices are connected to a channel made by self (private channel). The channel is used to view the pollutants parameters with the unique API key of the channel of a particular user. Every channel has both Read and Write API keys to get access. Wi-Fi module, temperature, humidity, gas, and dust sensors are interfaced with the nano Arduino. The user is prompted to provide the API key of the channel. ESP8266-01 reads the key and sends it to the Arduino nano. If the key is matched, then the data transmission can be carried out between the channel and the microcontroller. The module is connected to the Wi-Fi through some AT Commands.

LIST OF FIGURES

TABLE OF CONTENTS

1.INTRODUCTION

Small solid and liquid particles suspended in the air are referred to as "fine particulate matter" (FPM). Compared to other air pollutants, they are the main cause of air pollution and have the biggest negative effects on individuals. Poor air quality can have negative effects on health and is bad for all living things. It's important to be aware of the air quality in a certain region in order to prevent things that could make health problems worse. As a result, it's important to regularly check the local air quality. Temperature, humidity, and air-contained chemicals including ozone, sulfur dioxide,carbon monoxide, and particles can all be measured as part of an air quality monitoring process. We can currently measure the proficiency level metrics to assess the state of the air quality thanks to advancements in sensor technology. IoT (Internet of Things) advancements have also aided remote monitoring technology. In this study, we develop an IoT-based monitoring system to continuously track the air quality in a specific location. Monitoring is carried out by gathering information from numerous remote-accessible sensors. This study was based on a number of earlier ones that had already been done. There have been studies in the past that have controlled and monitored the air quality in the space. This study is also based on our investigation into the design of remote communication for tracking air quality and offering a remedy by setting up a purification system. Several airborne substances, including O3, SO2, CO, and particulates, as well as temperature and humidity, will be measured by the system. Remote monitoring of the air quality will be done using websites and purification is done based on input received and the input is based on a certain value that we fixed as a threshold in our programming that we will get by using the web browser on the real time basis.

2.PROBLEM STATEMENT

To show the weather parameters of an area and warn users of the quality of air of that location. It is intended to keep the people updated about the changing air quality of a particular area.

An effective natural observing framework is essential to screen and estimate the conditions in the event of surpassing the endorsed level of the parameter (for example, commotion, CO, and radiation levels). At the point when the items like condition furnished with sensor gadgets, smaller scale controllers and different programming applications turn into a self – securing and self-observing condition.

The ESP8266 Wi-Fi module is a self-contained SOC with an integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. The ESP8266 module is pre-programmed with an AT command set firmware. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application-specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. It contains a self-calibrated RF allowing it to work under all operating conditions and requires no external RF parts.

3.HARDWARE SPECIFICATIONS

Components required

• Nano Arduino
• WIFI Module (ESP8266)
• Temperature Sensor (DHT11)
• MQ6 Sensor
• Gas Sensor(MQ135)
• LCD Display
• Battery
• Purifier

- **Components overview**

1. ARDUINO UNO

The Arduino UNO is a standard board of Arduino. Here UNO means 'one' in Italian. It was named as UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is considered as the powerful board used in various projects. Arduino.cc developed the Arduino UNO board.

Arduino UNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits.

FIG 3.1 Arduino Uno

2. MQ-6

Gasses like butane and LPG can be detected or measured by the MQ-6 Gas monitor. When you only need to identify one particular gas, the MQ-6 sensor module's built-in Digital Pin allows it to function without the aid of a microcontroller. The analog pin, which is also TTL driven and operates on 5V and can be used with the majority of popular microcontrollers, must be used to measure the gas in ppm.

FIG3.2 MQ-6

3. ESP8266 WIFI MODULE:

The ESP8266 Wi-Fi module is a self-contained SOC with an integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

FIG 3.3 ESP8266

Each ESP8266 module comes pre-programmed with an AT command set firmware. This module has a powerful enough on-board processing and storage capability that allows it to

be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry.

4. DHT11

The DHT11 sensor comes in a single row 4-pin package and operates from a 3.5 to 5.5V power supply. It can measure temperature from 0-50 °C with an accuracy of ±2°C and relative humidity ranging from 20-95% with an accuracy of ±5%. The sensor provides fully calibrated digital outputs for the two measurements. It has its own proprietary 1- wire protocol, and therefore, the communication between the sensor and a microcontroller is not possible through a direct interface with any of its peripherals.

Fig 3.4 DHT11

5. MQ135

A hazardous gas detection apparatus for the family, the environment, suitable for ammonia, aromatic compounds, sulfur, benzene vapor, smoke, and other gasses harmful gas detection, gas-sensitive element test concentration. Air quality sensor is for detecting a wide range of gasses, including NH3, NOx, alcohol, benzene, smoke, and CO2.
The MQ-135 gas sensor senses gasses like CO2, ammonia nitrogen, oxygen, alcohols, aromatic compounds, sulfide, and smoke. In the atmosphere, we can find polluting gasses, but the conductivity of the gas sensor increases as the concentration of polluting gas increases.

FIG 3.5 MQ-135 Sensor

4. SOFTWARE SPECIFICATIONS:-

1. Arduino(1.8.0)

2. Android studio.

3. THINGSPEAK

- ThingSpeak is an analytic IoT platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud.

- **ThingSpeak** is an open data platform for the Internet of Things that allows you to collect data in your channel and get data from other channels using the API

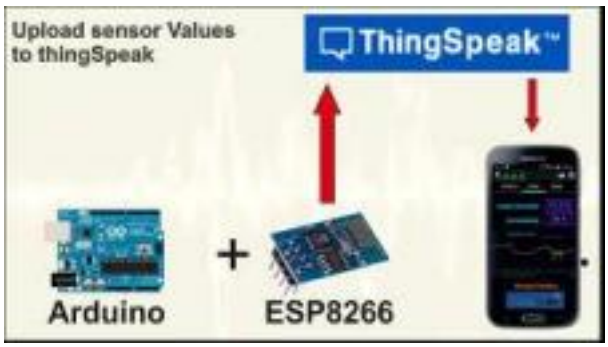

FIG 4.1 THINGSPEAK

5.BLOCK DIAGRAM

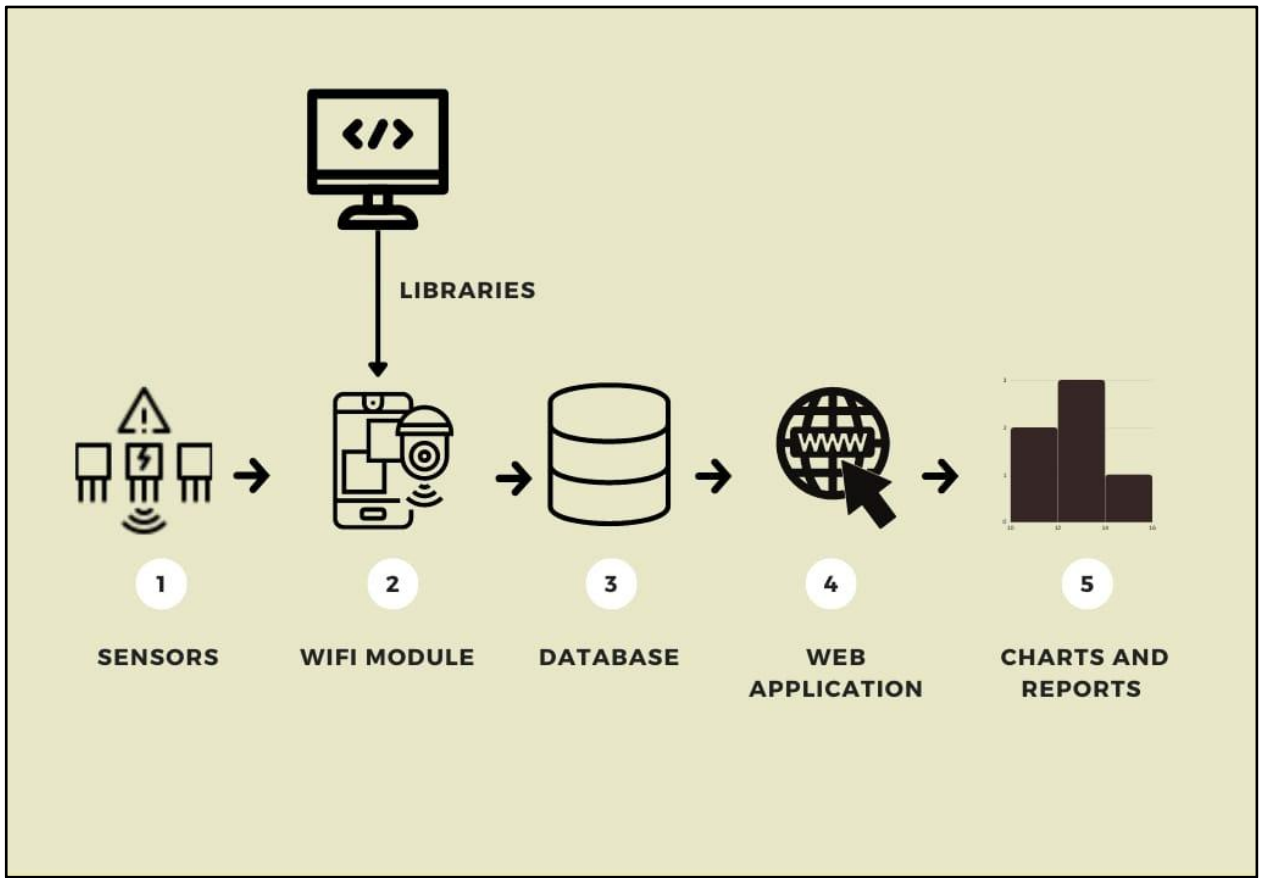

FIG 5.1 Process Flow

6.FLOWCHART

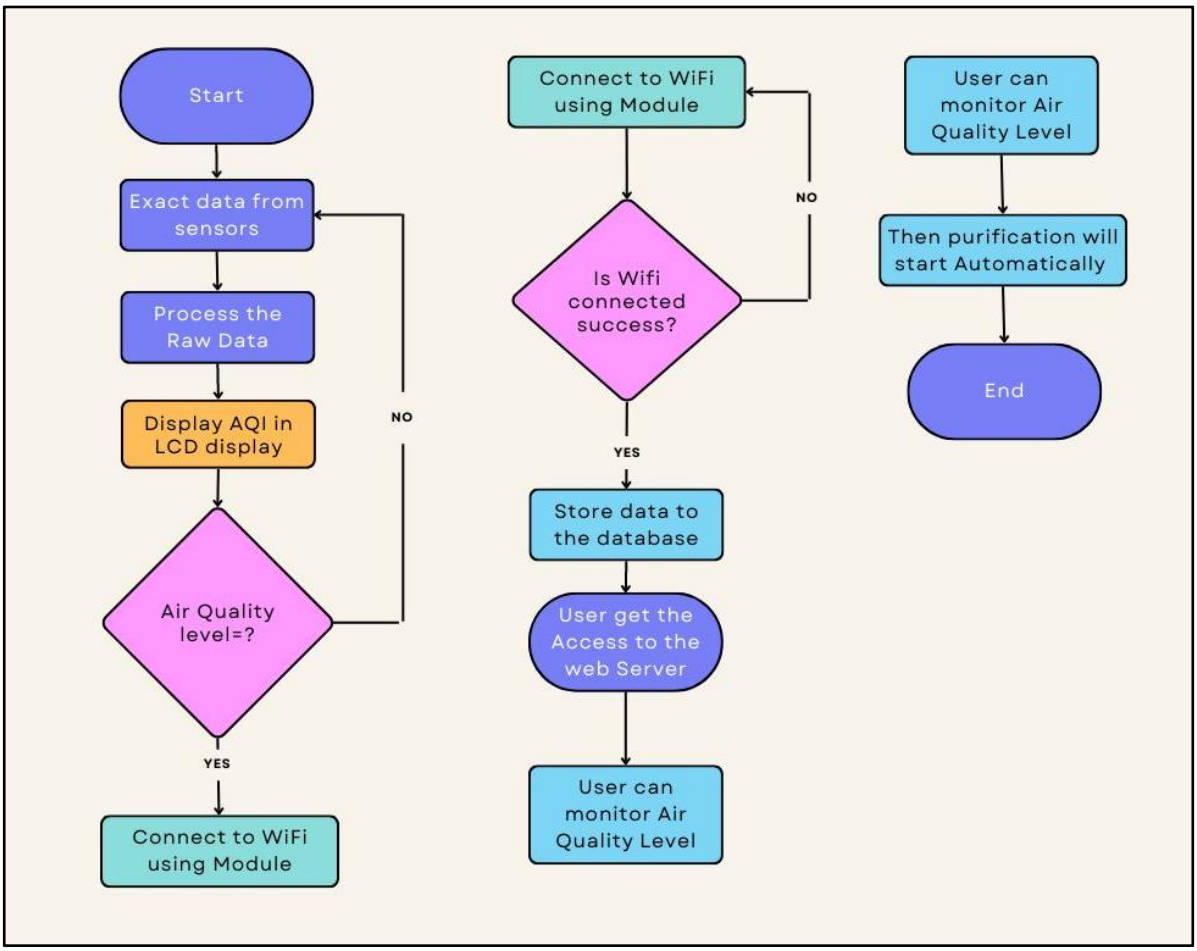

FIG 6.1 Implementation Flowchart

7.WHAT'S NEW

The proposed IOT based project is for monitoring pollution parameters like temperature**, humidity and CO levels** in the atmosphere to make the environment intelligent or interactive with the objects through wireless communication.

This proposed project monitors pollution parameters along with air quality. The proposed model is more adaptable and distributive to monitor the environmental parameters.

Monitoring gives air pollution concentrations, which can then be analyzed, interpreted, and presented. Regular analysis of monitoring data allows us to assess how bad air pollution is from day to day.

The air quality is becoming adverse these days majorly due to the concentration of dust particles, which is like a silent killer as it is extremely small in size and enters into the nose tract and leads to many respiratory problems. Our project includes an optical dust sensor to measure PM2.5 which leads to the precise measuring of air quality.

8.PROGRESS

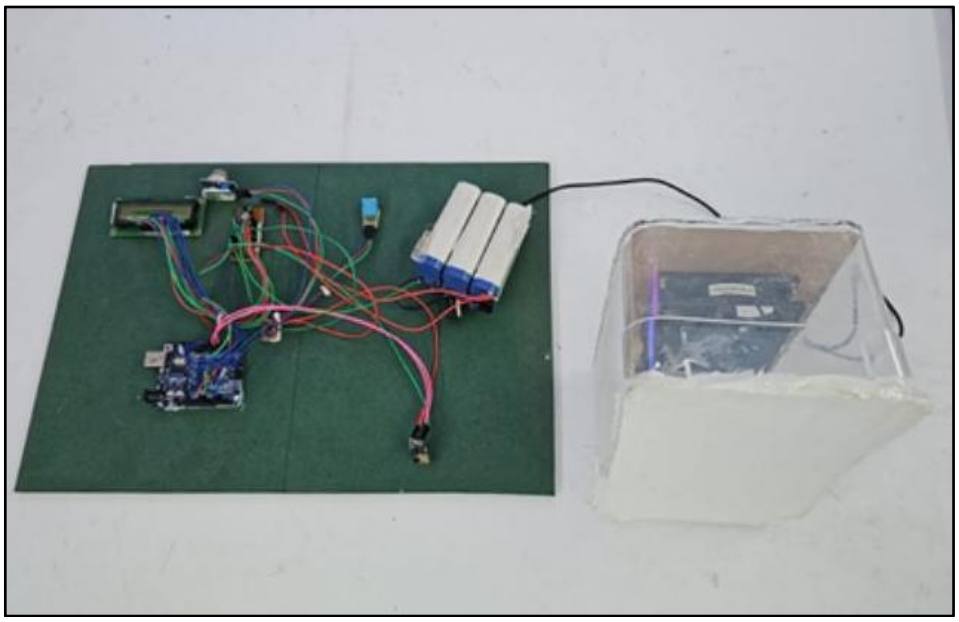

FIG 8.1: PROGRESS

- The status of the proposed Air Quality Determination and Purification Using IOTsystem is on track.
- The circuit designing and all software related work to the proposed project has been done.
- The hardware work for air quality monitoring is completed. Coding for the project has also been finished.

9.PROGRAMMING

```
#include <SoftwareSerial.h>
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
#include <stdlib.h>
#include <dht.h>
#define dht_dpin A0
int mq135;
int mq6;
int i, j;
dht DHT;

char buf1[16];
char buf2[16];
char buf3[16];
char buf4[16];
char buf5[16];
String strmq135;
String strmq6;

String stri;
String strj;

// replace with your channel's thingspeak API key
String apiKey = "2ZH86WTL0GGYXEX5";
```

```
SoftwareSerial ser(5, 6); // RX, TX

// this runs once
void setup() {
  // initialize the digital pin as an output.
   lcd.begin(16, 2);
lcd.setCursor(0, 0);
 lcd.print("IOT Sensor ");

 delay(2000);
 lcd.clear();

analogReference(DEFAULT);
  // enable debug serial
  Serial.begin(9600);
  // enable software serial
  ser.begin(115200);
// reset ESP8266
  ser.println("AT+RST");
   delay(500);
  ser.println("AT+CWMODE=3");
   delay(500);
  ser.println("AT+CWJAP=\"project\",\"12345678\"");
  delay(500);
  pinMode(2, OUTPUT);
  digitalWrite(2, HIGH);
    pinMode(3, OUTPUT);
  digitalWrite(3, LOW);
```

```
  pinMode(A2, INPUT);
    pinMode(A3, INPUT);
      pinMode(A4, INPUT);
      digitalWrite(A4, HIGH);
}


// the loop
void loop() {
   DHT.read11(dht_dpin);


 i=DHT.humidity;
      j =DHT.temperature;

mq6 = analogRead(A1); //mq6
mq135 = analogRead(A3); //mq135
if(i>70 || j>39 || mq6>300 || mq135>300)
{
  digitalWrite(2, LOW);
digitalWrite(3, HIGH);
  delay(500);
 digitalWrite(3, LOW);
}
else
{
   digitalWrite(2, HIGH);

   digitalWrite(3, LOW);
}
```

```
   lcd.setCursor(0, 0);
    lcd.print("mq6:");
 lcd.print(mq6);
 lcd.print(" ");

   lcd.setCursor(8, 0);
    lcd.print("m35:");
 lcd.print(mq135);
 lcd.print(" ");
  lcd.setCursor(0, 1);
    lcd.print("H:");
 lcd.print(i);
 lcd.print(" ");
  lcd.setCursor(8, 1);
   lcd.print("T:");
 lcd.print(j);
 lcd.print(" ");

// convert to string
  strmq135 = dtostrf(mq135, 4, 1, buf1);
 strmq6 = dtostrf(mq6, 4, 1, buf2);

  // convert to string

  stri = dtostrf(i, 4, 1, buf3);
strj = dtostrf(j, 4, 1, buf4);


   Serial.print(strmq135);
```

```
    Serial.print(" ");
   Serial.print(strmq6);
    Serial.print(" ");
   Serial.print(stri);
  Serial.print(" ");
   Serial.print(strj);

  Serial.println(" ");
String cmd = "AT+CIPSTART=\"TCP\",\"";
  cmd += "184.106.153.149"; // api.thingspeak.com
  cmd += "\",80";
  ser.println(cmd);

  if(ser.find("Error")){
   Serial.println("AT+CIPSTART error");
   return;
  }


 // prepare GET string
  String getStr = "GET /update?api_key=";
  getStr += apiKey;
  getStr +="&field4=";
  getStr += String(strmq135);
  getStr += "\r\n\r\n";




// send data length
```

```
  cmd = "AT+CIPSEND=";
  cmd += String(getStr.length());
  ser.println(cmd);


   if(ser.find(">")){
    ser.print(getStr);
  }
   else{
    ser.println("AT+CIPCLOSE");
    // alert user
    Serial.println("AT+CIPCLOSE");
  }
  delay(16000);
   DHT.read11(dht_dpin);


 i=DHT.humidity;
      j =DHT.temperature;

mq6 = analogRead(A1); //mq6
mq135 = analogRead(A3); //mq135
if(i>70 || j>39 || mq6>300 || mq135>300)
{
  digitalWrite(2, LOW);
digitalWrite(3, HIGH);
  delay(500);
 digitalWrite(3, LOW);
}
else
```

```
{
  digitalWrite(2, HIGH);

  digitalWrite(3, LOW);
}

   lcd.setCursor(0, 0);
    lcd.print("mq6:");
lcd.print(mq6);
lcd.print(" ");

   lcd.setCursor(8, 0);
    lcd.print("m35:");
lcd.print(mq135);
lcd.print(" ");
 lcd.setCursor(0, 1);
    lcd.print("H:");
lcd.print(i);
lcd.print(" ");
 lcd.setCursor(8, 1);
  lcd.print("T:");
lcd.print(j);
lcd.print(" ");

// convert to string
 strmq135 = dtostrf(mq135, 4, 1, buf1);
strmq6 = dtostrf(mq6, 4, 1, buf2);

 // convert to string
```

```
  stri = dtostrf(i, 4, 1, buf3);
strj = dtostrf(j, 4, 1, buf4);


   Serial.print(strmq135);
     Serial.print(" ");
   Serial.print(strmq6);
     Serial.print(" ");
    Serial.print(stri);
  Serial.print(" ");
   Serial.print(strj);

  Serial.println(" ");


  String cmd1 = "AT+CIPSTART=\"TCP\",\"";
  cmd1 += "184.106.153.149"; // api.thingspeak.com
  cmd1 += "\",80";
  ser.println(cmd1);

  if(ser.find("Error")){
    Serial.println("AT+CIPSTART error");
    return;
  }


 // prepare GET string
  String getStr1 = "GET /update?api_key=";
  getStr1 += apiKey;
  getStr1 +="&field1=";
```

```
  getStr1 += String(strmq6);
  getStr1 += "\r\n\r\n";




// send data length
  cmd1 = "AT+CIPSEND=";
  cmd1 += String(getStr1.length());
  ser.println(cmd1);



   if(ser.find(">")){
    ser.print(getStr1);
  }
   else{
    ser.println("AT+CIPCLOSE");
    // alert user
    Serial.println("AT+CIPCLOSE");
  }

  // thingspeak needs 15 sec delay between updates
  delay(16000);
 DHT.read11(dht_dpin);



 i=DHT.humidity;
      j =DHT.temperature;

mq6 = analogRead(A1); //mq6
```

```
mq135 = analogRead(A3); //mq135
if(i>70 || j>39 || mq6>300 || mq135>300)
{
  digitalWrite(2, LOW);
digitalWrite(3, HIGH);
  delay(500);
 digitalWrite(3, LOW);
}
else
{
   digitalWrite(2, HIGH);

   digitalWrite(3, LOW);
}

    lcd.setCursor(0, 0);
     lcd.print("mq6:");
 lcd.print(mq6);
 lcd.print(" ");

    lcd.setCursor(8, 0);
     lcd.print("m35:");
 lcd.print(mq135);
 lcd.print(" ");
  lcd.setCursor(0, 1);
     lcd.print("H:");
 lcd.print(i);
 lcd.print(" ");
  lcd.setCursor(8, 1);
   lcd.print("T:");
```

```
 lcd.print(j);
 lcd.print(" ");

// convert to string
  strmq135 = dtostrf(mq135, 4, 1, buf1);
 strmq6 = dtostrf(mq6, 4, 1, buf2);

  // convert to string

  stri = dtostrf(i, 4, 1, buf3);
strj = dtostrf(j, 4, 1, buf4);



   Serial.print(strmq135);
     Serial.print(" ");
   Serial.print(strmq6);
     Serial.print(" ");
    Serial.print(stri);
  Serial.print(" ");
   Serial.print(strj);

  Serial.println(" ");

   // TCP connection
  String cmd2 = "AT+CIPSTART=\"TCP\",\"";
  cmd2 += "184.106.153.149"; // api.thingspeak.com
  cmd2 += "\",80";
  ser.println(cmd2);

  if(ser.find("Error")){
```

```
  Serial.println("AT+CIPSTART error");
  return;
 }


// prepare GET string
 String getStr2 = "GET /update?api_key=";
 getStr2 += apiKey;
 getStr2 +="&field2=";
 getStr2 += String(stri);
 getStr2 += "\r\n\r\n";




// send data length
 cmd2 = "AT+CIPSEND=";
 cmd2 += String(getStr2.length());
 ser.println(cmd2);


  if(ser.find(">")){
   ser.print(getStr2);
 }
  else{
   ser.println("AT+CIPCLOSE");
   // alert user
   Serial.println("AT+CIPCLOSE");
 }
```

```
 // thingspeak needs 15 sec delay between updates
 delay(16000);

 DHT.read11(dht_dpin);


i=DHT.humidity;
    j =DHT.temperature;

mq6 = analogRead(A1); //mq6
mq135 = analogRead(A3); //mq135
if(i>70 || j>39 || mq6>300 || mq135>300)
{
  digitalWrite(2, LOW);
digitalWrite(3, HIGH);
delay(500);
 digitalWrite(3, LOW);

}
else
{
   digitalWrite(2, HIGH);

   digitalWrite(3, LOW);
}

    lcd.setCursor(0, 0);
     lcd.print("mq6:");
```

```
 lcd.print(mq6);
 lcd.print(" ");

    lcd.setCursor(8, 0);
     lcd.print("m35:");
 lcd.print(mq135);
 lcd.print(" ");
  lcd.setCursor(0, 1);
     lcd.print("H:");
 lcd.print(i);
 lcd.print(" ");
  lcd.setCursor(8, 1);
   lcd.print("T:");
 lcd.print(j);
 lcd.print(" ");

// convert to string
  strmq135 = dtostrf(mq135, 4, 1, buf1);
 strmq6 = dtostrf(mq6, 4, 1, buf2);

  // convert to string

  stri = dtostrf(i, 4, 1, buf3);
strj = dtostrf(j, 4, 1, buf4);



   Serial.print(strmq135);
     Serial.print(" ");
   Serial.print(strmq6);
     Serial.print(" ");
```

```
  Serial.print(stri);
 Serial.print(" ");
  Serial.print(strj);

 Serial.println(" ");
  // TCP connection
 String cmd3 = "AT+CIPSTART=\"TCP\",\"";
 cmd3 += "184.106.153.149"; // api.thingspeak.com
 cmd3 += "\",80";
 ser.println(cmd3);

 if(ser.find("Error")){
  return;
 }


// prepare GET string
 String getStr3 = "GET /update?api_key=";
 getStr3 += apiKey;
 getStr3 +="&field3=";
 getStr3 += String(strj);
 getStr3 += "\r\n\r\n";




// send data length
 cmd3 = "AT+CIPSEND=";
 cmd3 += String(getStr3.length());
 ser.println(cmd3);
```

```
  if(ser.find(">")){
   ser.print(getStr3);
  }
   else{
   ser.println("AT+CIPCLOSE");
   // alert user
   Serial.println("AT+CIPCLOSE");
  }


  // thingspeak needs 15 sec delay between updates
  delay(16000);
 DHT.read11(dht_dpin);


 i=DHT.humidity;
     j =DHT.temperature;

mq6 = analogRead(A1); //mq6
mq135 = analogRead(A3); //mq135
if(i>70 || j>39 || mq6>300 || mq135>300)
{
  digitalWrite(2, LOW);

  digitalWrite(3, HIGH);
  delay(500);
}
else
```

```
{
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
}

   lcd.setCursor(0, 0);
    lcd.print("mq6:");
 lcd.print(mq6);
 lcd.print(" ");

   lcd.setCursor(8, 0);
    lcd.print("m35:");
 lcd.print(mq135);
 lcd.print(" ");
  lcd.setCursor(0, 1);
    lcd.print("H:");
 lcd.print(i);
 lcd.print(" ");
  lcd.setCursor(8, 1);
   lcd.print("T:");
 lcd.print(j);
 lcd.print(" ");

// convert to string
  strmq135 = dtostrf(mq135, 4, 1, buf1);
 strmq6 = dtostrf(mq6, 4, 1, buf2);

  // convert to string

  stri = dtostrf(i, 4, 1, buf3);
```

```
strj = dtostrf(j, 4, 1, buf4);

   Serial.print(strmq135);
     Serial.print(" ");
   Serial.print(strmq6);
     Serial.print(" ");
    Serial.print(stri);
  Serial.print(" ");
   Serial.print(strj);

  Serial.println(" ");
}
```

10.ADVANTAGES OF THE PROJECT

- The IoT monitoring system project using Arduino uno is fully automated.
- It does not require any human attention.
- We can get information about AQI on thingspeak.
- The low cost and effort are less in this system .
- Accuracy is high.
- A smart way to monitor Pollution level.
- Efficient

11.APPLICATIONS OF THE PROJECT

The fact that indoor air pollution causes more than 3.8 million deaths each year is quite alarming. Particulate matter and hazardous gases lower air quality, which when breathed can result in serious illnesses like asthma, lowered lung function, and even cancer.

Although both the industrial and the commercial sectors are covered by the data, the effect of air pollutants on workers is greater because of the higher concentration of contaminants. Thus, to keep the AQI under control, the indoor air quality monitoring device aids businesses in creating a healthier work environment. Companies can ensure sufficient ventilation, manage the production of pollutants in their facility, and maintain temperature & humidity levels in a comfortable range by comparing the real-time air quality data with ideal conditions.

12.FUTURE SCOPE OF THE PROJECT

- One can implement a few more sensors and connect them to the satellite as a global feature of this system.
- Adding more sensor to monitor other environmental parameters such as CO2,Pressure, and Oxygen Sensor
- In aircraft, navigation, and military there is a great scope of this real-time system.
- It can also be implemented in hospitals or medical institutes for the research & study in “Effect of Weather on Health and Diseases”, hence providing better precaution alerts.
- We may add a weather monitoring system integrating in the existing system.

13.REFERENCE

[1].Marin Berov Marinov, Dimitar Iliev Iliev, Todor Stoynov Djamiykov, Ivan Vladimirov Rachev, Katya Konstantinova Asparuhova Department of Electronics, Faculty of Electronic Engineering and Technologies Technical University of Sofia 8 Kliment Ohridski Blvd., 1756 Sofia, Bulgaria, mbm@tu-sofia.bg

[2].Siavash Esfahani , Piers Rollins, Jan Peter Specht , Marina Cole , Julian W.Gardner School of Engineering, University of Warwick, Coventry, UK

[3].J. M. Pinter and M. L. Kiss, "Determination and measurement of parameters affecting indoor comfort," 2019 20th International Carpathian Control Conference (ICCC), 2019.

[4]. Ahmed Samy Moursi1 · Nawal ElFishawy1 · Soufene Djahel2 · Marwa Ahmed Shouman1 Received: 21 November 2020 / Accepted: 16 July 2021 / Published online: 29 July 2021 © The Author(s) 2021

[5].DANIEL IBASETA1 , JULIO MOLLEDA2 , FIDEL DÍEZ1 & JUAN C. GRANDA21 CTIC Technological Centre, Spain 2 Department of Computer Science and Engineering, University of Oviedo, Spain

[6] Guang Yang 1, HwaMin Lee 2,and Giyeol Lee 3,*1Department of Computer Science, Soonchunhyang University, Asan 31538, Korea; taffiffiffic@outlook.com 2Department of Computer Software & Engineering, Soonchunhyang University, Asan 31538, Korea;leehm@sch.ac.kr 3Department of Landscape Architecture, Chonnam National University, Gwangju 61186, Korea

[7] Matthew Meli (Contact Author),Edward Gatt ,Owen Casha ,Ivan Grech ,Joseph Micallef Department of Microelectronics and Nanoelectronics,University of Malta,Msida, Maltajoseph.micallef@um.edu.mt

[8] S. McGrath, C. Flanagan, L. Zeng and C. O'Leary, "IoT Personal Air Quality Monitor," 2020 31st Irish Signals and Systems Conference (ISSC), 2020..

[9] JunHo Jo ,1 ByungWan Jo ,1 JungHoon Kim ,1 SungJun Kim,1 and WoonYong Han2 1Department of Civil and Environmental Engineering, Hanyang University, 04763 Seoul, Republic of Korea2Smart IS, 22101 Incheon, Republic of Korea Correspondence should be addressed to ByungWan Jo; joycon@hanmail.net

[10] R. K. Jha, "Air Quality Sensing and Reporting System Using IoT," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020