AIR POLLUTION MONITORING SYSTEM

Submitted in partial fulfillment of the requirements

for the subject of

Sensor Network Lab

by

Ankitsingh Gusain

Roll no:- 08

Tushar Bhosale

Roll No:- 47

Darshan Jadhav

Roll no:- 54

Under the Supervision of-

Prof. J.M.Hatwar

DEPARTMENT OF INFORMATION TECHNOLOGY

KONKAN GYANPEETH COLLEGE OF ENGINEERING

KARJAT-410201

Certificate

This is to certify that the project entitled **Air Pollution Monitoring System** is a bonafide work of **Ankitsingh Gusain**(Roll no:- 08) , **Tushar Bhosale**(Roll no:- 47) , **Darshan Jadhav** (Roll no:- 54) submitted to the **Department of Information Technology** in partial fulfillment of the requirement for the subject of **Sensor Network Lab.**

Supervisor/Guide	Head of Department
Prof. J.M.Hatwar	Prof. A.W.Kale
Department of Information Technology	Department of Information Technology

Project report approval

This project report entitled **Air Pollution Monitoring System**, by **Ankitsingh Gusain**(Roll no:- 08), **Tushar Bhosale**(Roll no:- 47) , **Darshan Jadhav** (Roll no:- 54) is approved for the partial fulfillment of the requirement for the subject of Sensor Network lab.

Examiners

1……………………………………….

2……………………………………….

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where other ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact /source in our submission. We understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the source which have thus nor been properly cited or from whom proper permission has not been taken when needed.

Signature.

Ankitsingh Gusain

(Roll no.:08)

Signature.

Tushar Bhosale
(Roll no.:47)

Signature

Darshan Jadhav

(Roll no:- 54)

Date:

Place:

ACKNOWLEDGEMENT

It has come out to be a sort of great pleasure and experience for me to work on the project "**Air Pollution Monitoring System**". we wish to express our ideas to those who helped us i.e. faculty of our Institute Dr. Anil Kale(HOD) & Mr. J.M.Hatwar (Project Guide) the preparation of the manual script of this text. This would not have been made successful without their help and precious suggestions. Finally, we also warmly thank all my colleagues who encouraged us to an extent, which made the project successful.

Table of Contents

Abstract

As the world's population is becoming increasingly urban, the cities are under pressure to remain livable. In recent years, the air quality of the cities has become one of the major cause of concern around the world. Air pollution causes respiratory and other health problems among the people. It is necessary to constantly monitor the air quality index of a city to make it smart and livable. Hence monitoring the quality of air is very essential as it helps to take necessary and preventive measures to reduce harmful gases and thus save many lives.

This project presents a proposal for an Air Pollution Monitoring System. The real-time data of the air quality is accessed through the smart devices and analyzed to measure the impact on city dwellers. The smart devices are capable of measuring the Air Quality Index, Temperature, Humidity, Pressure, and Dew Point in the atmosphere. The gathered data is accessible globally through an Android Application as well as on ThingSpeak platform in the form of graphs.

The system is implemented using 3 different sensors such as MQ135 to detect Air Quality Index, DHT11 sensor to measure Humidity level and BMP280 sensor to measure Temperature and Pressure in the surrounding air. Also we have used NodeMCU ESP8266 module to interface the sensors. After the sensors collect the data, it is being uploaded to ThingSpeak platform and this data is then retrieved in the Android Application that we have made using Android Studio and the Firebase real-time database. So this application displays the values for parameters such as Air Quality Index, Temperature, Humidity, Pressure, and Dew Point in the atmosphere and it shows whether the Air is good, moderate, unhealthy, very unhealthy or hazardous. This Air Pollution Monitoring System is implemented and tested in real-life working condition.

1. Introduction

Air pollution is both an environmental and a social problem, as it leads to a multitude of adverse effects on human health, ecosystems and the climate. Air pollution is one of the largest environmental health risks. Quality of the air in city and urban areas is the most important factor that directly influences the incidence of diseases and decreases the quality of life.

Taking appropriate decisions in a timely period depends on the measurement and analysis of the parameters of the air, which creates the need for the development of real time air quality monitoring. The use of multi-parameter air quality monitoring systems makes it possible to do a detailed level analysis of major pollutants and their sources. These monitoring systems are important components in many smart city projects for monitoring air quality in urban areas.

Clean air is vital need for every human being. Polluted air causes many health problems and several damages. Therefore to make any step ahead of controlling the pollution rate it is necessary to monitor the air quality which may help us to make a right decision at right time.

Here we present an approach for cost-effective measurement of Air Pollution, based on different sensors, NodeMCU ESP8266, ThingSpeak and an Android Application to show the results. It costs around 750 rupees, anyone can afford it. This Air Pollution Monitoring System collects real time data using sensor integrated with NodeMCU and the graphs are displayed on ThingSpeak platform & this data is then retrieved from the ThingSpeak platform to Firebase real-time database and displayed on the UI of the application

2. Aim and Objectives

The aim of the project is to develop an Air Pollution Monitoring System using different sensors (MQ135 sensor, DHT11 sensor and BMP280 sensor), NodeMCU, ThingSpeak platform and an Android application which will monitor real time data and display the air quality on the android application as well as on ThingSpeak platform.

The core objectives are:

- Gather system requirements
- Evaluate and study the platform required for the system.
- Evaluate and study suitable development language, technologies and tools.
- Evaluate Methods of Interface.
- Program NodeMCU.
- Interface board with sensors.
- Collection of real time data using sensors.
- Evaluate and test the system.
- Display the Air Quality in the App UI and on ThingSpeak.
- Maintain system.

3. Problem Definition

Air pollution monitoring is though old but very useful concept in day to day life. Air pollution monitoring start from traditional way to the most sophisticated computer has been used to monitor the air quality, however the fresh air is necessary for all human being, for that various technology has been used and some of this technology is really useful in order to provide a real time air quality data.

So, to monitor the air quality we have proposed a system which will monitor the Air Pollution. This Air Pollution Monitoring System should be able to collect real time data about the air pollution and display the result about the air quality. Different Sensors, NodeMCU ESP8266 module, ThingSpeak and an Android application can be used to make this system. It should be cost-effective, so that anyone can buy it.

The system must be able to collect different values using sensors like MQ135 sensor (to measure Air Quality Index), DHT11 sensor (to measure Humidity) and BMP280 sensor (to measure Temperature & Pressure) integrated with NodeMCU ESP8266 module over Wi-Fi and display the graphs on ThingSpeak platform. Then this data should be retrieved from ThingSpeak platform to Firebase real-time database and displayed on the UI of the Application made using Android Studio. The UI shows the values like Temperature, Pressure, Humidity, Dew Points and the Air Quality Index with status of air quality.

4. Proposed System

4.1 Block Diagram

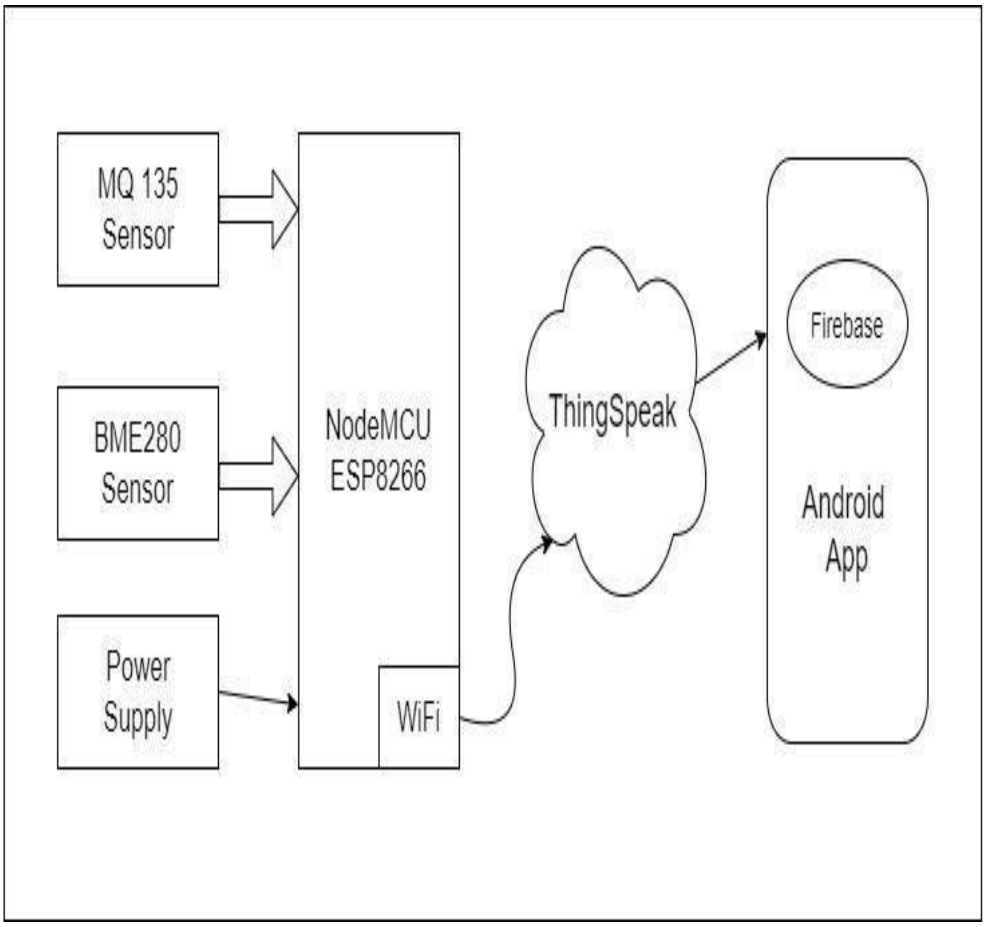


Figure 4.1 Block Diagram

4.2 Flow Chart

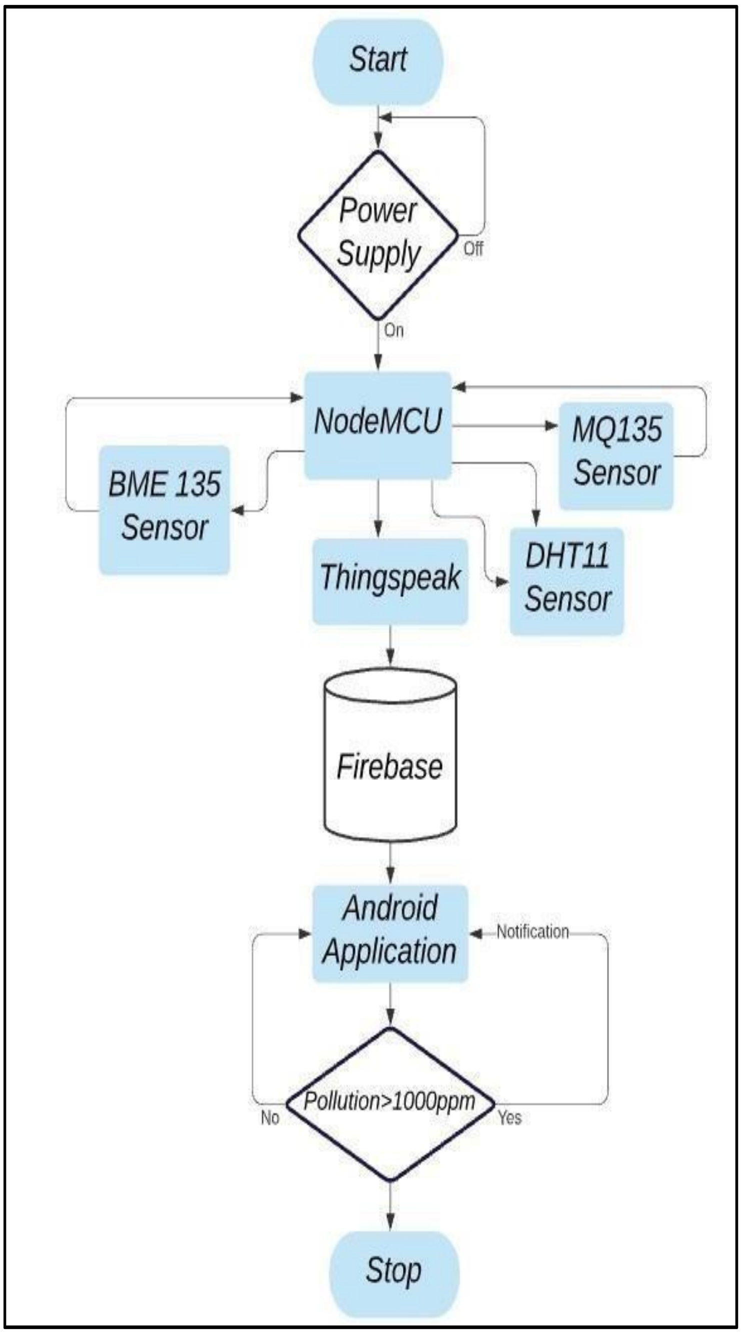


Figure 4.2 Flow Chart

5. Components

5.1 Hardware:

5.1.1 NodeMCU

NodeMCU v3 is a development board which runs on the ESP8266 with the Espress if Non-OS SDK, and hardware based on the ESP-12 module. The device features 4MB of flash memory, 80MHz of system clock, around 50k of usable RAM and an on chip WIFI Transceiver.

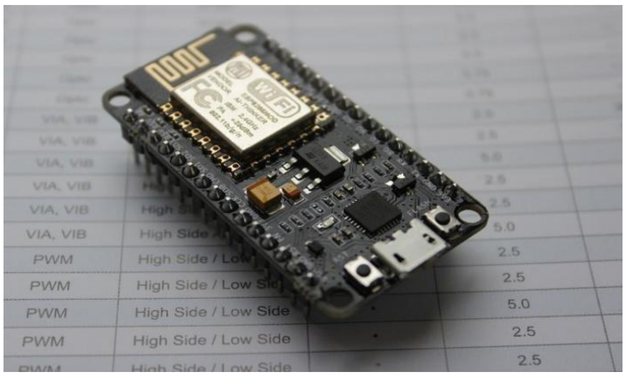


Figure 5.1.1A NODEMCU

Features:

Microcontroller: Tensilica 32-bit RISC CPU Xtensa

LX106 Operating Voltage: 3.3V Input Voltage: 7-
12V

Digital I/O Pins (DIO): 16	Analog Input Pins (ADC): 1
UARTs: 1	SPIs: 1
Flash Memory: 4 MB	SRAM: 64 KB
Clock Speed: 80 Mhz	Wi-Fi: IEEE 802.11 b/g/n:

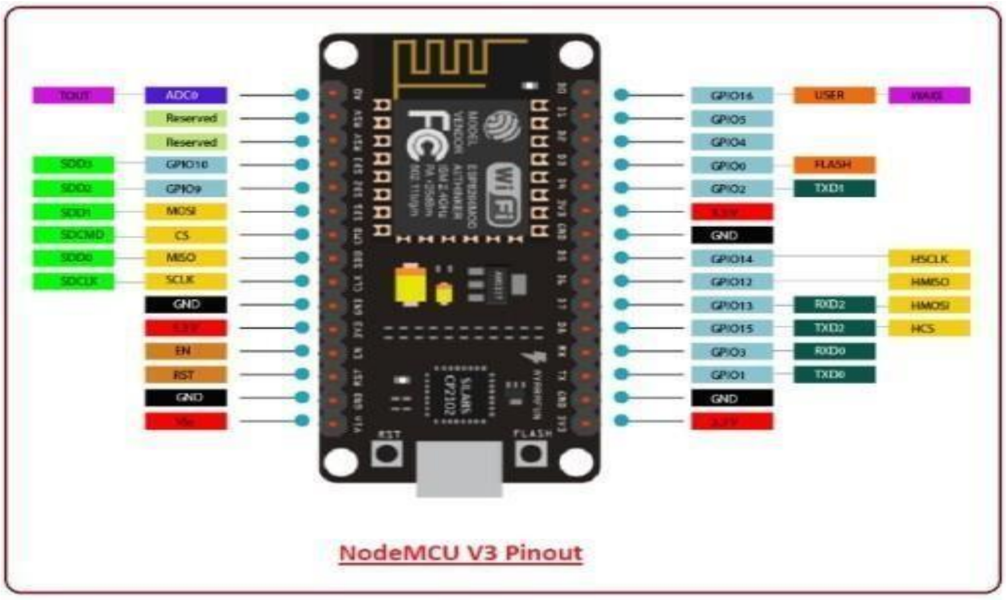


Figure 5.1.1B NodeMCU PIN Diagram

MQ135 Sensor

The MQ-135 Gas sensors are used in air quality control equipments and are suitable for detecting or measuring of NH3, NOx, Alcohol, Benzene, Smoke, and CO2. The MQ-135 sensor module comes with a Digital Pin which makes this sensor to operate even without a microcontroller and that comes in handy when you are only trying to detect one particular gas.

Figure 5.1.2 MQ135 Sensor

5.1.1 DHT11 Sensor

DHT11 sensor has a moisture holding component sandwiched between two electrodes. A minute variation in humidity leads to change in the conductivity of the component that results in the resistance change between the electrodes which in turn is measured by an Integrated Circuit (IC). DHT11 is preferred over DHT22 as the sampling rate of DHT11 is much better than that of DHT22. DHT11 is a Digital Sensor consisting of two different sensors in a single package. The sensor contains an NTC (Negative Temperature Coefficient) Temperature Sensor, a Resistive-type Humidity Sensor and an 8-bit Microcontroller to convert the analog signals from these sensors and produce a Digital Output.

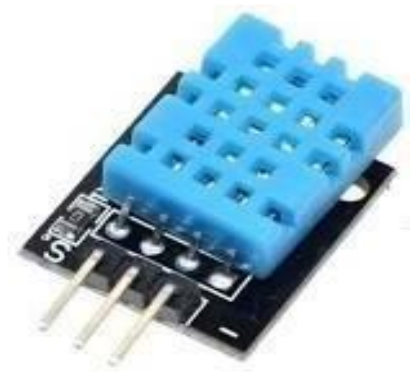


Figure 5.1.3 DHT11 Sensor

BMP280 Sensor

The BMP280 is an absolute barometric pressure sensor, which is especially feasible for mobile applications. Its small dimensions and its low power consumption allow for the implementation in battery powered devices such as mobile phones, GPS modules or watches.

Figure 5.1.4 BMP280 Sensor

Breadboard

A breadboard is a solder less device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connects the holes on the top of the board. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.

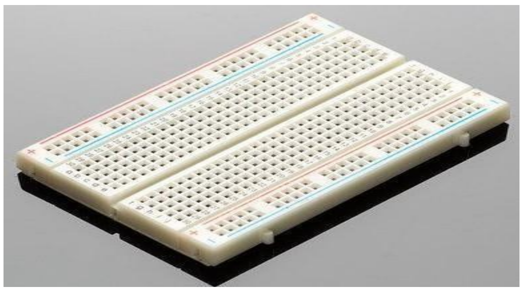


Figure 5.1.5 Breadboard

5.1.2 Jumper Wires

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other withoutsoldering. Jumper wires are used to connect two points in a circuit. All Electronics stocks jumper wire in a variety of lengths and assortments. Frequently used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.

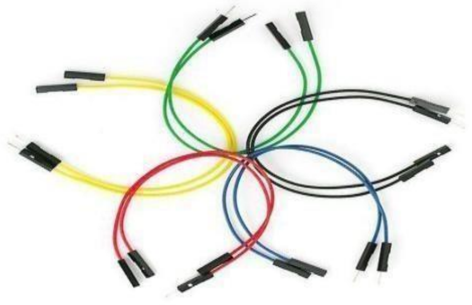


Figure 5.1.6 Jumper Wires

Software

5.1.3 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

Figure 5.2.1 Arduino IDE

ThingSpeak

ThingSpeak is IoT Cloud platform where you can send sensor data to the cloud. You can also analyze and visualize your data with MATLAB or other software, including making your own applications. The ThingSpeak service is operated by MathWorks. In order to sign up for ThingSpeak, you must create a new MathWorks Account or log in to your existing MathWorks Account.

Figure 5.2.2 ThingSpeak

Android Studio

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. A unified environment where you can develop for all *Android* devices. Apply Changes to push code and resource changes to your running app without restarting your app. Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development.[13] Java is still supported, as is C

Figure 5.2.2 Android Studio

Firebase

Firebase is Google's mobile platform that helps you quickly develop high-quality apps and grow your business. Firebase helps you build and run successful apps. Backed by Google, loved by developers. Accelerate app development with fully managed backend infrastructure. Firebase is a Backend-as-a-Service (Baas). Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents. Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

Figure 5.2.3 Firebase

5. LOGIC

In this Air Pollution Monitoring System, we are collecting the parameters like Temperature, Humidity, Pressure, Air Quality Index and calculating the Dew Points.

MQ135 Sensor gives us Air Quality Index and we have set levels such as

- 0-50: Air Quality is Good
- 51-100: Air Quality is Moderate
- 101-150: Air Quality is Unhealthy for Sensitive Groups
- 151-200: Air Quality is Unhealthy
- 201-300: Air Quality is Very Unhealthy
- 301-500: Air Quality is Hazardous

On the application we get AQI value with status of air quality.

BMP280 sensor collects temperature and pressure (hPA-hectoPascal). Temperature is collected in degree Celsius form we are converting it to degree Fahrenheit using formula = t*1.8+32.0 (t- temperature). Humidity (in %) is collected by DHT11 sensor.

The dew point is the temperature to which air must be cooled to become saturated with water vapor. When cooled further, the airborne water vapor will condense to form liquid water (dew). The higher the dew points, the higher the moisture content of the air at a given temperature. Dew points are calculated using formula: dp = t-((100.0-h)/5) (t- temperature, h- humidity).

6. Code

```
#include
<ESP8266WiFi.h>
#include
<Adafruit_Sensor.h>
#include
<Adafruit_BMP280.h>
#include "MQ135.h"
#include
<Arduino.h>
#include
"DHT.h"

#define D1 5
#define D2 4
#define D4 2
#define D3 0
#define DHTPIN
14 #define
DHTTYPE
DHT11

// assign the SPI bus
to pins #define
BMP_SCK D1
#define
BMP_MISO D4
#define
BMP_MOSI D2
#define BMP_CS
```

```
D3
#define SEALEVELPRESSURE_HPA (1013.25)

Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI,
BMP_MISO, BMP_SCK); DHT dht(DHTPIN,
DHTTYPE);
unsigned long
delayTime; float h,
t, p, dp;
char
temperatureFString[
6]; char dpString[6];
char
humidityString[6
]; char
pressureString[7]
;
String apiKey =
"T628M0MQYLRH4JY1";
const char* ssid = "siddhi";
const char* password = "siddhi14";
const char* server =
"api.thingspeak.com";
WiFiClient client;

void setup() {
 Serial.begin(11
 5200);
 delay(10);
 Serial.println();
 Serial.print("Connecti
 ng to ");
 Serial.println(ssid);
 WiFi.begin(ssid, password);
  while (WiFi.status() !=
  WL_CONNECTED){
   delay(500);
   Serial.print(".");
  }
 Serial.println("");
 Serial.println("WiFi
 connected");
 // Printing the ESP IP
 address
 Serial.println(WiFi.lo
 calIP()); delay(500);
 dht.begin();
 if (!bmp.begin()) {
   Serial.println("Could not find a valid BME280
   sensor, check wiring!"); return;
  }
 }

void loop() {
    MQ135 gasSensor = MQ135(A0);
    float air_quality =
    gasSensor.getPPM();
    Serial.print("Air Quality: ");
    Serial.print(air_quality);
    Serial.println(" PPM");
    Serial.println();
    //delay(2000);
    h = dht.readHumidity();
    t =
    bmp.readTemperatur
    e(); t = t*1.8+32.0;
    dp = t-((100.0-h)/5);
    p =
    bmp.readPressure()/100.0
```

```
        F; dtostrf(t, 5, 1,
        temperatureFString);
        dtostrf(h, 5, 1, humidityString);
        dtostrf(p, 6, 1, pressureString);
        dtostrf(dp, 5, 1,
        dpString);
        Serial.print("Temperature
        = ");
        Serial.println(temperature
        FString);
        Serial.print("Humidity =
        "); Serial.println(h,1);
        Serial.print("Pressure =
        ");
        Serial.println(pressureStri
        ng); Serial.print("Dew
        Point = ");
        Serial.println(dp,1);
        Serial.println("........................... ");

        if (client.connect(server,80)) // "184.106.153.149" or api.thingspeak.com)
   {
         String postStr =
         apiKey; postStr
         +="&field1=";
         postStr +=
         String(air_quality);
         postStr
         +="&field2=";
         postStr +=
         String(pressureString);
         postStr  +="&field3=";
         postStr +=
        String(humidityString);
        postStr +="&field4=";
        postStr +=
        String(dpString);
        postStr
        +="&field5=";
        postStr +=
        String(temperatureFString);
        postStr += "\r\n\r\n";
   {

        client.print("POST /update
        HTTP/1.1\n");
        client.print("Host:
        api.thingspeak.com\n");
        client.print("Connection:
        close\n");
        client.print("X-THINGSPEAKAPIKEY:
        "+apiKey+"\n"); client.print("Content-Type:
        application/x-www-form-urlencoded\n");
        client.print("Content-Length: ");
        client.print(postStr.le
        ngth());
        client.print("\n\n");
        client.print(postStr);
      }
      client.stop();
    }
```

8. Implementation

8.1 Working

When the system is turned on, all the sensors will collect the data from the surrounding air. MQ135 sensor will help to find Air Quality Index, DHT11 sensor will measure the Humidity level in the air and BMP280 sensor will collect the values of Temperature and Pressure. These sensors are connected to NodeMCU. So, the collected data values are sent to ThingSpeak through NodeMCU over the Wi-Fi. On ThingSpeak different graphs will be shown according to the collected data. This data is then retrieved in Firebase real-time database using API keys and then these retrieved values are shown in the Android Application. The interface of the application shows values of Temperature, Pressure, Humidity and Dew Points with the AQI index of the air as well as status of the air. And the application shows last five at the same time with current values.

8.2 Circuit Diagram

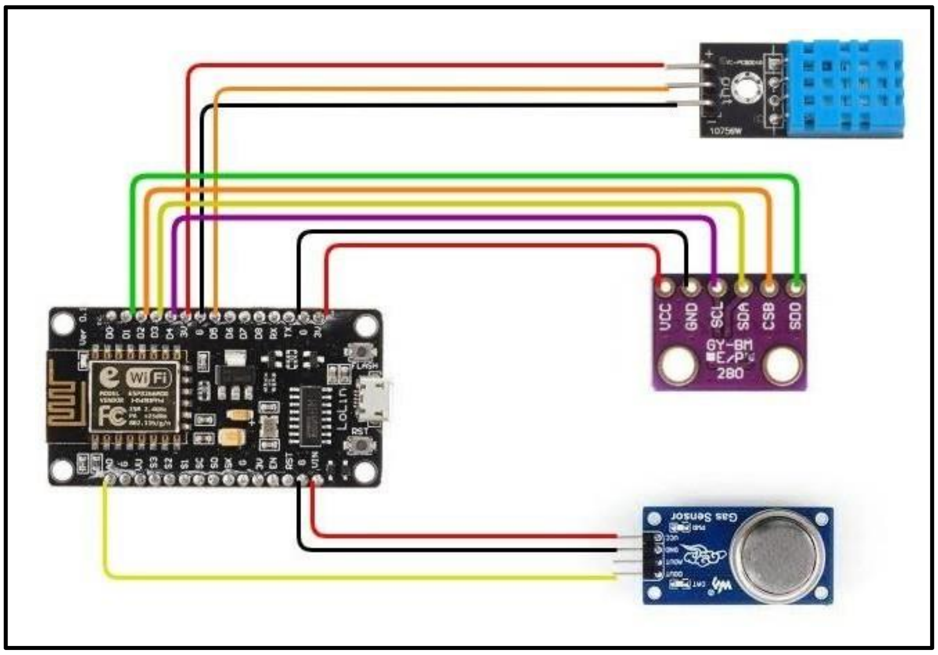


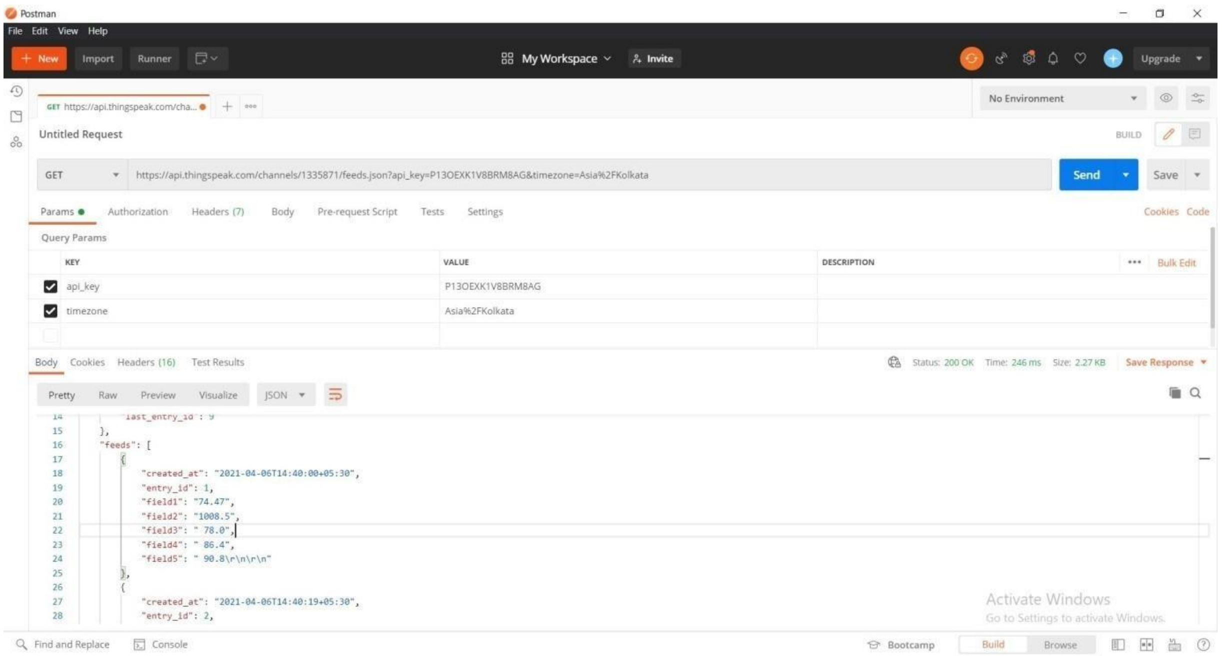
Figure 8.2 Circuit Diagram

9. Deployment of Testing

As mentioned in the report, we have created an android application for showing the values of different parameters collected by the sensors. The values are getting stored on ThingSpeak platform. To show the values on the mobile application we have used the API keys from ThingSpeak and retrieved those values in the application.

Now to check the format of data that we have retrieved from ThingSpeak using API keys and to check whether the retrieved data is correct or not, we have used the Postman platform. Testing of the data format and the correct data is shown in figure below.

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster.



Untitled Request

```
GET https://api.thingspeak.com/channels/1335871/feeds.json?api_key=P13OEXK1V8BRM8AG&timezone=Asia%2FKolkata
```

Query Params

KEY	VALUE	DESCRIPTION
✓ api_key	P13OEXK1V8BRM8AG	
✓ timezone	Asia%2FKolkata	

Status: 200 OK Time: 246 ms Size: 2.27 KB

```
        "last_entry_id": 9
    },
    "feeds": [
        {
            "created_at": "2021-04-06T14:40:00+05:30",
            "entry_id": 1,
            "field1": "74.47",
            "field2": "1008.5",
            "field3": " 78.0",
            "field4": " 86.4",
            "field5": " 90.8\r\n\r\n"
        },
        {
            "created_at": "2021-04-06T14:40:19+05:30",
            "entry_id": 2,
```

Figure 9.1 Postman

10. Conclusion & Future Scope

The aim of this project was to propose a cost effective Air Pollution Monitoring System. The approach discussed in the project was successfully implemented. This system is highly reliable and efficient. It can be implemented anywhere and it is very useful for highly polluted areas for monitoring the air quality. This system is very cheap, so anyone can afford it easily. Also it consumes less space.

The future scope for this system can be huge. There are many factors to improve on to make it more powerful, intelligent, scalable, and to become better overall for monitoring and controlling the air pollution. For example, we can notify the user if the air quality goes beyond the specified range with different safety measures. To make the system respond fasterdifferenttechnologiescan be implemented.Well,no system is ever perfect. It always has a scope for improvement. One just needs to put onathinkingcapandtryandmakethe system better.

BIBLIOGRAPHY

[1] Frances Moore, "Climate Change and Air Pollution: Exploring the Synergies and Potential for Mitigation in Industrializing Countries", Sustainability, 2009. Vol. 1(1). pp. 43-54.

[2] Harsh G., Dhaanjay B., Himanshu A., Vinay T.. Arun Kumar, "An IOT Based Air Pollution Monitoring System for Smart Cities", ICSETS 2019.

[3] Brook RD, FB.. Cascio W. Hong Y. Howard G. Lipsett M. Luepker R. Mittleman M. Samet J. Smith SC Jr. Tager 1: "Air pollution and cardiovascular disease a statement for healthcare professionals from the Expert Panel on Population and Prevention Science of the American Heart Association". Circulation. Vol. 109 (21) June 1, 2004

[4] H Ah, JK Soe, and S.R. Weller. "A real-time ambient air quality monitoring wireless sensor network for schools smart cities" In the ICSETS 2019 176Proceedings of the IEEE First International Smart Cities Conference (ISC2 15): 25-28 Oct 2015 Goadalajara, Mexico

151 Sarath K. Guttikundaab. R. "Health impacts of particulate pollution in a megacity Delhi India" Environmental Development. April 2013 Vol 6. pp. 8-20