Air Quality Monitoring System

Professional Practice/Seminar (IT890) Project Report

Submitted in partial fulfilment of the requirements for the degree of

MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

by

MOHD ASIF KHAN KHAISHAGI (202200)

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE -575025

MAY 2021

DECLARATION

I hereby declare that the Professional Practice/Seminar (IT890) Project Work Report of the M.Tech.(IT) entitled Indoor Air Quality Monitoring System which is being submitted to the National Institute of Technology Karnataka Surathkal, in partial fulfilment of the requirements for the award of the Degree of Master of Technology in the department of Information Technology, is a **bonafide report of the work carried out by me**. The material contained in this project report has not been submitted to any University or Institution for the award of any degree.

Asif

(Signature of Student)

MOHD ASIF KHAN KHAISHAGI (202200)
(Name And Registration Number of Student)

PLACE : NITK, SURATHKAL

DATE : 22-05-2021

CERTIFICATE

This is to certify that the Professional Practice/Seminar (IT890) Project Work Report entitled INDOOR AIR QUALITY MONITORING SYSTEM submitted by MOHD ASIF KHAN KHAISHAGI, (Register Number: 202200) as the record of the work carried out by him/her, is accepted as the Professional Practice/Seminar (IT890) Project Work Report submission in partial fulfilment of the requirements for the award of degree of Master of Technology in the Department of Information Technology.

EXAMINER NAME GUIDE NAME

SIGNATURE WITH DATE SIGNATURE WITH DATE

TABLE OF CONTENTS

AIR QUALITY MONITORING SYSTEM

1 Abstract

Air is one the most important aspect of human life, but now a days due to smoke from vehicles, harmful gases from air-conditioners and refridgerators, burning of crops and forests have degraded air quality in our country as well as globally. Air pollution is among the most dangerous issues to human life. To monitor the quality of air in our surroundings is important in getting information about pollution and harmful it is to our health. In this project, I have made an indoor air quality monitor which will show what is the air quality in our surrounding as well as what is what is danger level associated with it.

2 Introduction

Air inside the house may seem harmless but there can be colorless and odourless substances in air like carbon monoxide etc. which can affect human health. So for this project, I made Indoor Air Quality Monitoring System which consists of hardware , IOT data aggregation and visualisation platform and an Android App which will show the Air quality status at real time.

2.1 ABOUT AQI

As referenced from [5, 6] suggest that standard for Air quality monitoring is AQI.

What is the U.S. Air Quality Index (AQI)?

The U.S. AQI is EPA's index for reporting air quality.

How does the AQI work?

Think of the AQI as a yardstick that runs from 0 to 500. The higher the AQI value, the greater the level of air pollution and the greater the health concern. For example, an AQI value of 50 or below represents good air quality, while an AQI value over 300 represents hazardous air quality.

For each pollutant an AQI value of 100 generally corresponds to an ambient air concentration that equals the level of the short-term national ambient air quality standard for protection of public health. AQI values at or below 100 are generally thought of as satisfactory. When AQI values are above 100, air quality is unhealthy: at first for certain sensitive groups of people, then for everyone as AQI values get higher.

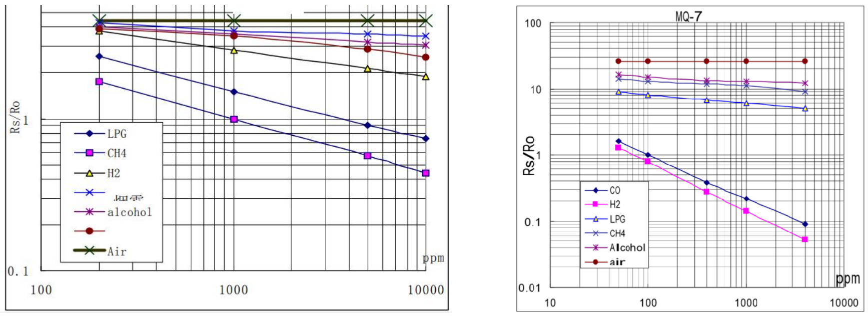
The AQI is divided into six categories. Each category corresponds to a different level of health concern. Each category also has a specific color. The color makes it easy for people to quickly determine whether air quality is reaching unhealthy levels in their communities.

AQI Table		
Levels of Concern	Values of Index	Description of Air Quality
Good	0 to 50	Air quality is satisfactory, and air pollution poses little or no risk.
Moderate	51 to 100	Air quality is acceptable. However, there may be a risk for some people, particularly those who are unusually sensitive to air pollution.
Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general public is less likely to be affected.
Unhealthy	151 to 200	Some members of the general public may experience health effects; members of sensitive groups may experience more serious health effects.
Very Unhealthy	201 to 300	Health alert: The risk of health effects is increased for everyone.
Hazardous	301 and higher	Health warning of emergency conditions: everyone is more likely to be affected.

2.2 CALCULATING AQI

The MQ sensors basically check for the PPM value by the change in the resistance as discussed in [3].

From Figure 1, we try to derive the following calculations for both the sen-



(a) Gas Sensor (b) CO Sensor

Figure 1: Datasheet-Change in Resistance vs change in PPM

sors.

$$y = mx + b \tag{1}$$

For a log-log scale, the formula looks like this:

$$\log y = m \log x + b \tag{2}$$

Let's find the slope. To do so, we need to choose 2 points from the graph. In our case, we chose the points (200,2.6) and (10000,0.75) from the LPG (Liquified Petroleum gas)line from fig 1a. The LPG line is a result of sensor under testing with various level of LPG as input. The formula to calculate slope m(here) is the following:

$$m = \frac{\log y - \log y_0}{\log x - \log x_0} \tag{3}$$

If we apply the logarithmic quotient rule we get the following:

$$m = \frac{\log (y/y_0)}{\log (x/x_0)} \tag{4}$$

Now we substitute the values for x, x_0 , y, and y_0 :

$$m = \frac{\log(0.75/2.6)}{\log(10000/200)} \tag{5}$$

$$m = -0.318 \tag{6}$$

Now that we have m, we can calculate the y intercept. To do so, we need to choose one point from the graph (once again from the LPG line). In our case, we chose (5000,0.9)

$$\log(y) = m * \log(x) + b \tag{7}$$

$$b = \log(0.9) - (-0.318) * \log(5000) \tag{8}$$

$$b = 1.13 \tag{9}$$

Now that we have m and b, we can find the gas concentra- tion for any ratio with the following formula:

$$\log(x) = \frac{\log(y) - b}{m} \tag{10}$$

$$x = 10^{\frac{\log(y)-b}{m}} \tag{11}$$

Using equations 11, we will be able to convert the sensor output values into PPM (Parts per Million)[7].

3 Literature Survey

3.1 Toward SATVAM: An IoT Network for Air Quality Monitoring

Air pollution is ranked as the second most serious risk for public health in India after malnutrition. The lack of spatially and temporally distributed air quality information prevents a scientific study on its impact on human health and on the national economy. In this paper, we present our initial efforts toward SATVAM, Streaming Analytics over Temporal Variables for Air quality Monitoring, that aims to address this gap. We introduce the multi-disciplinary, multi-institutional project and some of the key IoT technologies used. These cut across hardware integration of gas sensors with a wireless mote packaging, design of the wireless sensor network using 6LoWPAN and RPL, and integration with a cloud backend for data acquisition and analysis. The outcome of our initial deployment will inform an improved design that will enable affordable and manageable monitoring at the city scale. This should lead to data-driven policies for urban air quality management.

3.2 SATVAM: Toward an IoT Cyber-Infrastructure for Low-Cost Urban Air Quality Monitoring

Air pollution is a public health emergency in large cities. The availability of commodity sensors and the advent of Internet of Things (IoT) enable the deployment of a city-wide network of 1000's of low-cost real-time air quality monitors to help manage this challenge. This needs to be supported by an IoT cyber-infrastructure for reliable and scalable data acquisition from the edge to the Cloud. The low accuracy of such sensors also motivates the need for data-driven calibration models that can accurately predict the science variables from the raw sensor signals. Here, we offer our experiences with designing and deploying such an IoT software platform and calibration models, and validate it through a pilot field deployment at two mega-cities, Delhi and Mumbai. Our edge data service is able to even-out the differential bandwidths from the sensing devices and to the Cloud repository, and recover from transient failures. Our analytical models reduce the errors of the sensors from a best-case of 63% using the factory baseline to as low as 21%, and substantially advances the state-of-the-art in this domain.

3.3 IOT based Air Quality Monitoring System Using MQ135 and MQ7 with Machine Learning Analysis

This paper deals with measuring the Air Quality using MQ135 sensor along with Carbon Monoxide CO using MQ7 sensor. Measuring Air Quality is an important element for bringing awareness to take care of the future generations and for a healthier life. Based on this, Government of India has already taken certain measures to ban Single Stroke and Two Stroke Engine based motorcycles which are emitting high pollution. We are trying to implement a system using IoT platforms like Thingspeak or Cayenne in order to bring awareness to every individual about the harm we are doing to our environment. Already, New Delhi is remarked as the most pollution city in the world recording Air Quality above 300 PPM. We have used easiest platform like Thingspeak and set the dashboard to public such that everyone can come to know the Air Quality at the location where the system is installed. Machine Learning analysis brings us a lot of depth in understanding the information that we obtained from the data. Moreover, we are proviing a reducement of the cost of components versus the state of the art.

4 Methodology Followed

4.1 Hardware Components

This are the hardware components which i used for gathering real world air quality data.

4.1.1 Arduino UNO

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.[1] The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

4.1.2 MQ135 Gas 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, 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. If you need to measure the gases in PPM the analog pin need to be used. The analog pin is TTL driven and works on 5V and so can be used with most common microcontrollers.

4.1.3 MQ7 Carbon Monoxide Sensor

This is a simple-to-use Carbon Monoxide (CO) sensor, suitable for sensing CO concentrations in the air. The MQ-7 can detect CO-gas concentrations anywhere from 10 to 500ppm.

This sensor has a high sensitivity and fast response time. The sensor's output is an analog resistance. The drive circuit is very simple; all you need to do is power the heater coil with 5V, add a load resistance, and connect the output to an ADC.

4.1.4 ESP8266 Chip

ESP8266 (technically ESP8266EX) is a WiFi Module based on Cadence Tensilica L106 32-bit MCU manufactured by Espressif Systems. The ESP8266 SoC contains a fully functional WiFi Stack and TCP/IP Stack that allows any Microcontroller to get connected to WiFi Network.

With Software Development Kits (SDKs), you can directly program the ESP8266's on-chip Microcontroller, without the need for an external Microcontroller.

4.1.5 Assembling Hardware

Here i have combined all this hardware along with breadboard and jumper wires. After assembling the hardware components look like this.

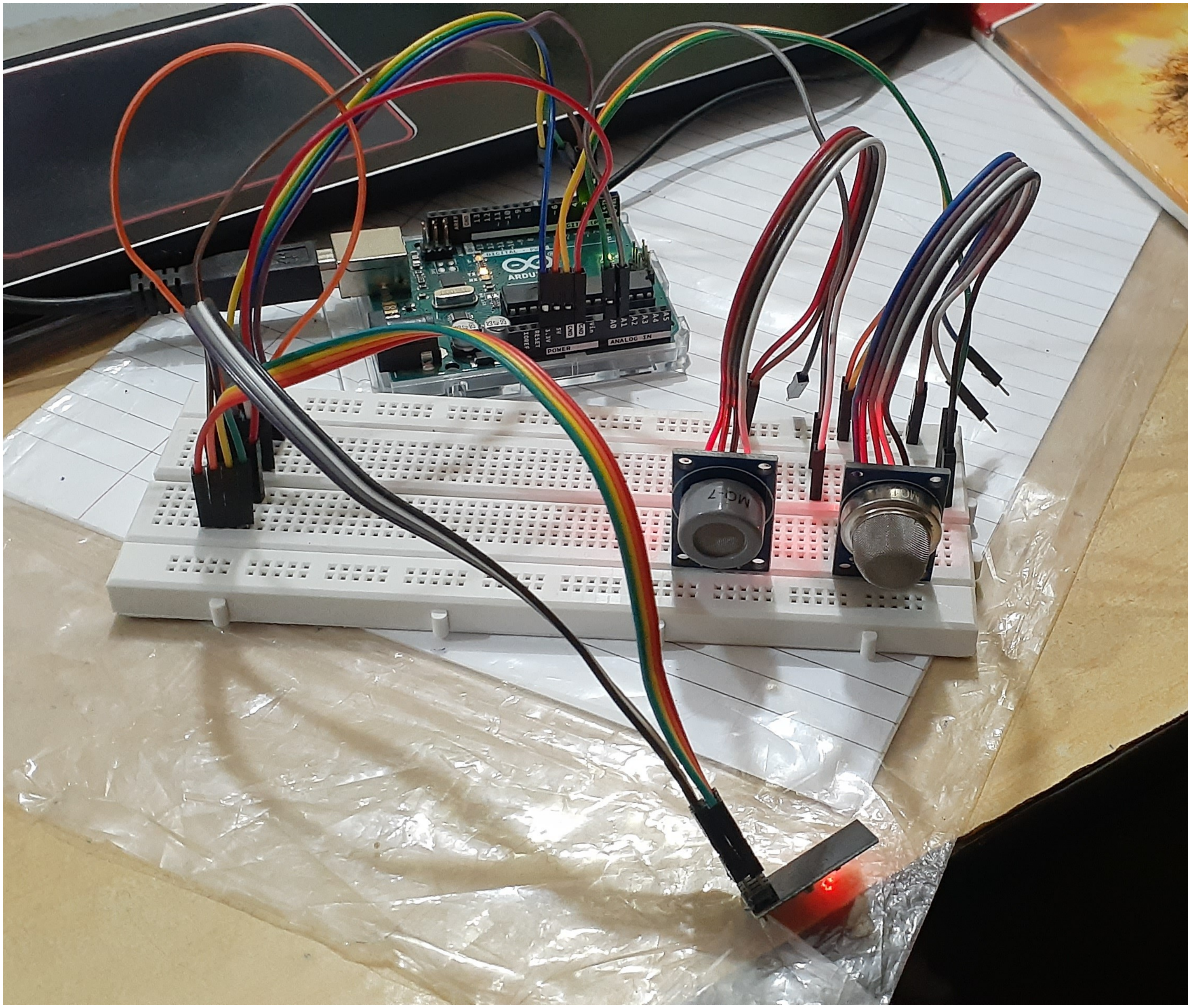


Figure 2: Working Air Quality Sensors

4.2 Writing Code For Sensor

4.2.1 Arduino IDE

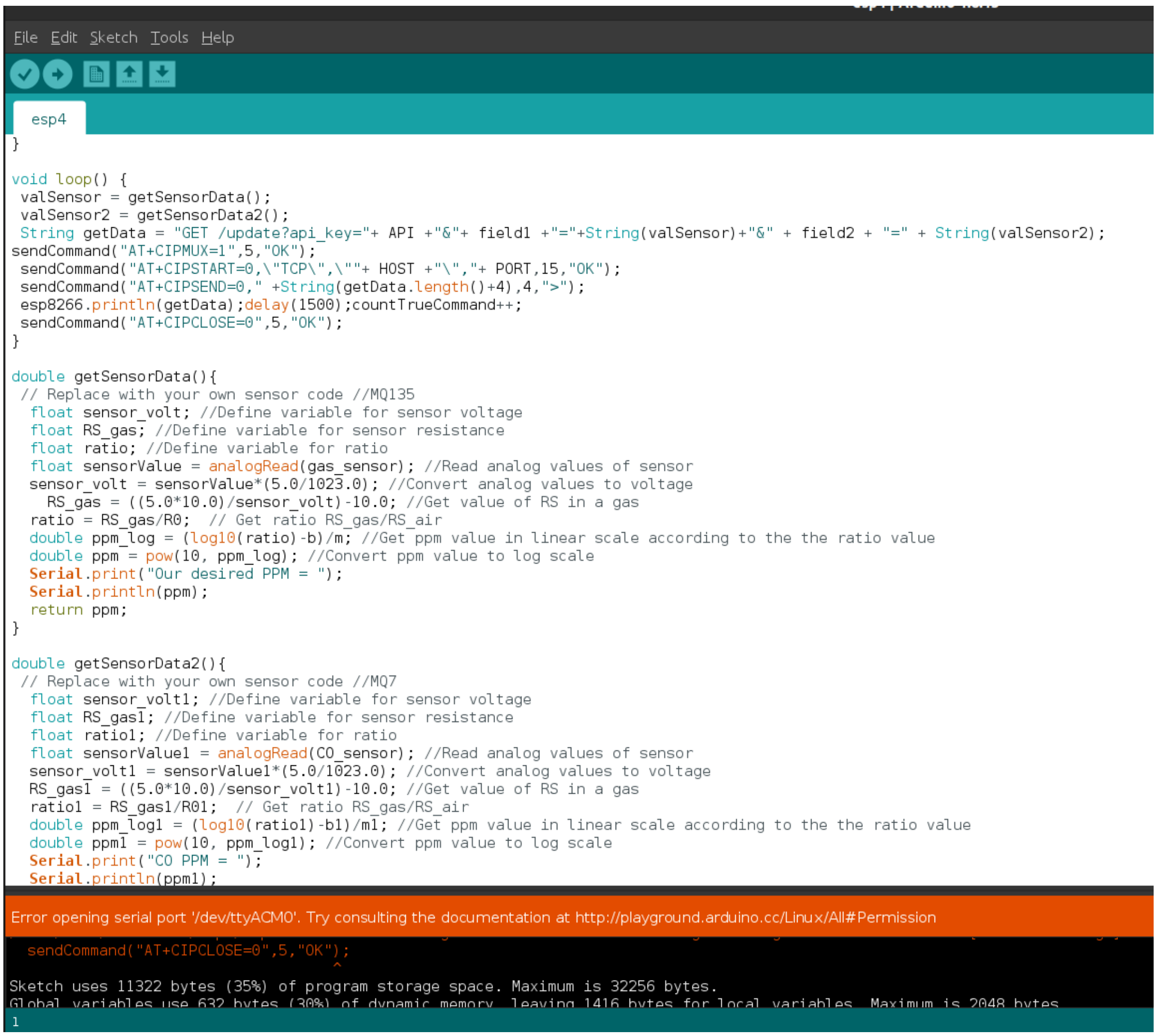
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

Figure 3: Arduino IDE

4.2.2 Arduino Coding

For reading sensor values, i have written the following arduino code and uploaded it arduino controller. It basically connects to wifi first, and then read sensor values and push them to thingspeak channel.




```
}

void loop() {
 valSensor = getSensorData();
 valSensor2 = getSensorData2();
 String getData = "GET /update?api_key="+ API +"&"+ field1 +"="+String(valSensor)+"&" + field2 + "=" + String(valSensor2);
sendCommand("AT+CIPMUX=1",5,"OK");
 sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
 sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");
 esp8266.println(getData);delay(1500);countTrueCommand++;
 sendCommand("AT+CIPCLOSE=0",5,"OK");
}

double getSensorData(){
 // Replace with your own sensor code //MQ135
  float sensor_volt; //Define variable for sensor voltage
  float RS_gas; //Define variable for sensor resistance
  float ratio; //Define variable for ratio
  float 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*10.0)/sensor_volt)-10.0; //Get value of RS in a gas
  ratio = RS_gas/R0;  // Get ratio RS_gas/RS_air
  double ppm_log = (log10(ratio)-b)/m; //Get ppm value in linear scale according to the the ratio value
  double ppm = pow(10, ppm_log); //Convert ppm value to log scale
  Serial.print("Our desired PPM = ");
  Serial.println(ppm);
  return ppm;
}

double getSensorData2(){
 // Replace with your own sensor code //MQ7
  float sensor_volt1; //Define variable for sensor voltage
  float RS_gas1; //Define variable for sensor resistance
  float ratio1; //Define variable for ratio
  float sensorValue1 = analogRead(CO_sensor); //Read analog values of sensor
  sensor_volt1 = sensorValue1*(5.0/1023.0); //Convert analog values to voltage
  RS_gas1 = ((5.0*10.0)/sensor_volt1)-10.0; //Get value of RS in a gas
  ratio1 = RS_gas1/R01;  // Get ratio RS_gas/RS_air
  double ppm_log1 = (log10(ratio1)-b1)/m1; //Get ppm value in linear scale according to the the ratio value
  double ppm1 = pow(10, ppm_log1); //Convert ppm value to log scale
  Serial.print("CO PPM = ");
  Serial.println(ppm1);
```

```
Error opening serial port '/dev/ttyACM0'. Try consulting the documentation at http://playground.arduino.cc/Linux/All#Permission
  sendCommand("AT+CIPCLOSE=0",5,"OK");
                                     ^
Sketch uses 11322 bytes (35%) of program storage space. Maximum is 32256 bytes.
Global variables use 632 bytes (30%) of dynamic memory, leaving 1416 bytes for local variables. Maximum is 2048 bytes.
1
```

Figure 4: Arduino Code

4.3 ThingSpeak

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualization of live data, and send alerts.

According to its developers, "ThingSpeak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates".

ThingSpeak was originally launched by ioBridge in 2010 as a service in support of IoT applications.

ThingSpeak has integrated support from the numerical computing software MATLAB from MathWorks, allowing ThingSpeak users to analyze and visualize uploaded data using Matlab without requiring the purchase of a Matlab license from Mathworks.

4.3.1 ThingSpeak Channel

For data aggregation I have made a thingspeak channel where i am collecting gas sensor and carbon monoxide sensor values.

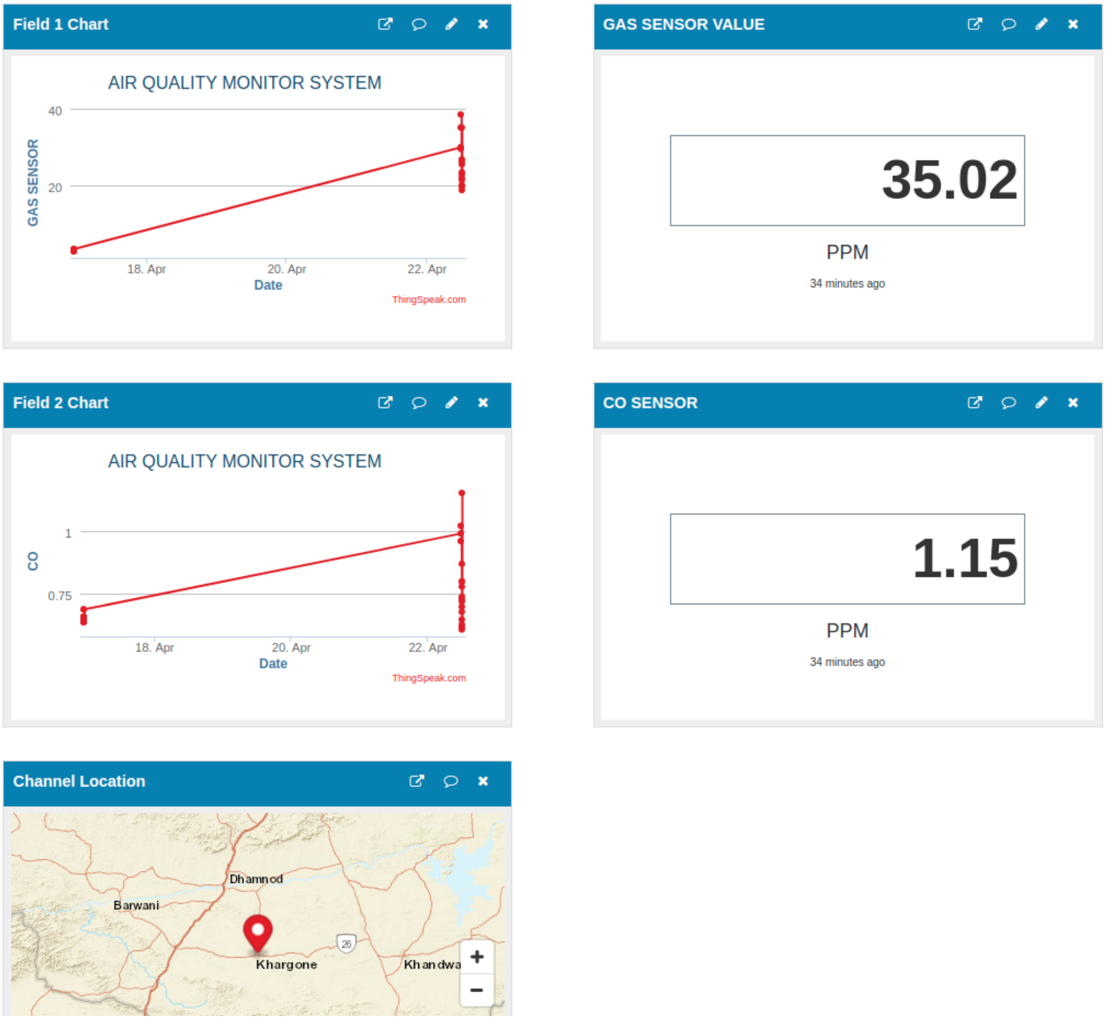


Figure 5: Thingspeak Channel

4.4 Android

4.4.1 Android Studio IDE

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- 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
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

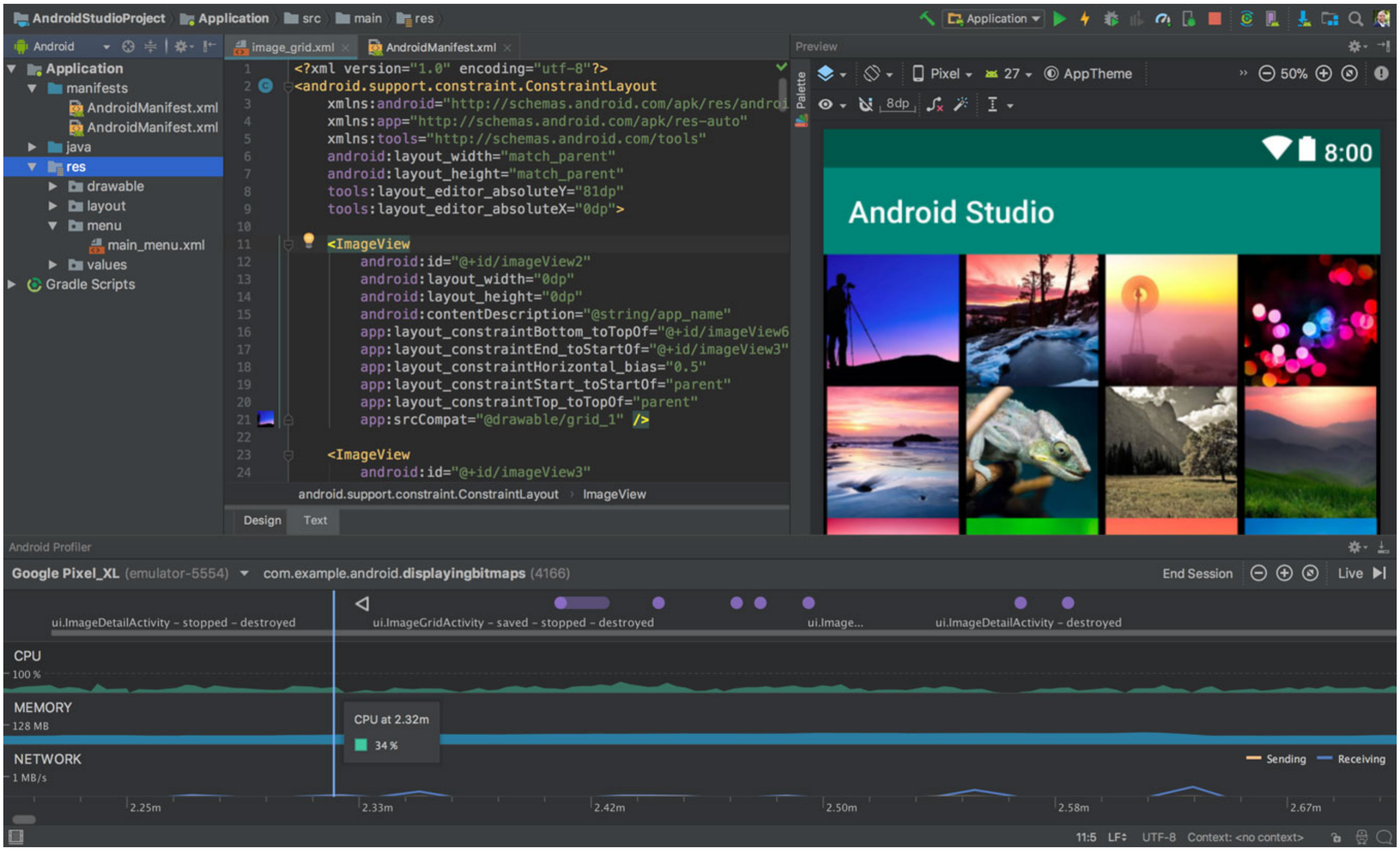


Figure 6: Android Studio

4.4.2 Android App

I build the android app on Android Studio IDE using Java, XML. It basically shows the air quality in ppm. Based on air quality it shows status also whether air is good,moderate,unhealthy,very unhealthy and hazardous. You can also view current reading of gas sensor as well as Carbon Monoxide sensor as well as the previous trend graph of the data. I have also added text to speech feature in gas sensor pages. I have attached screenshots of all the screens.

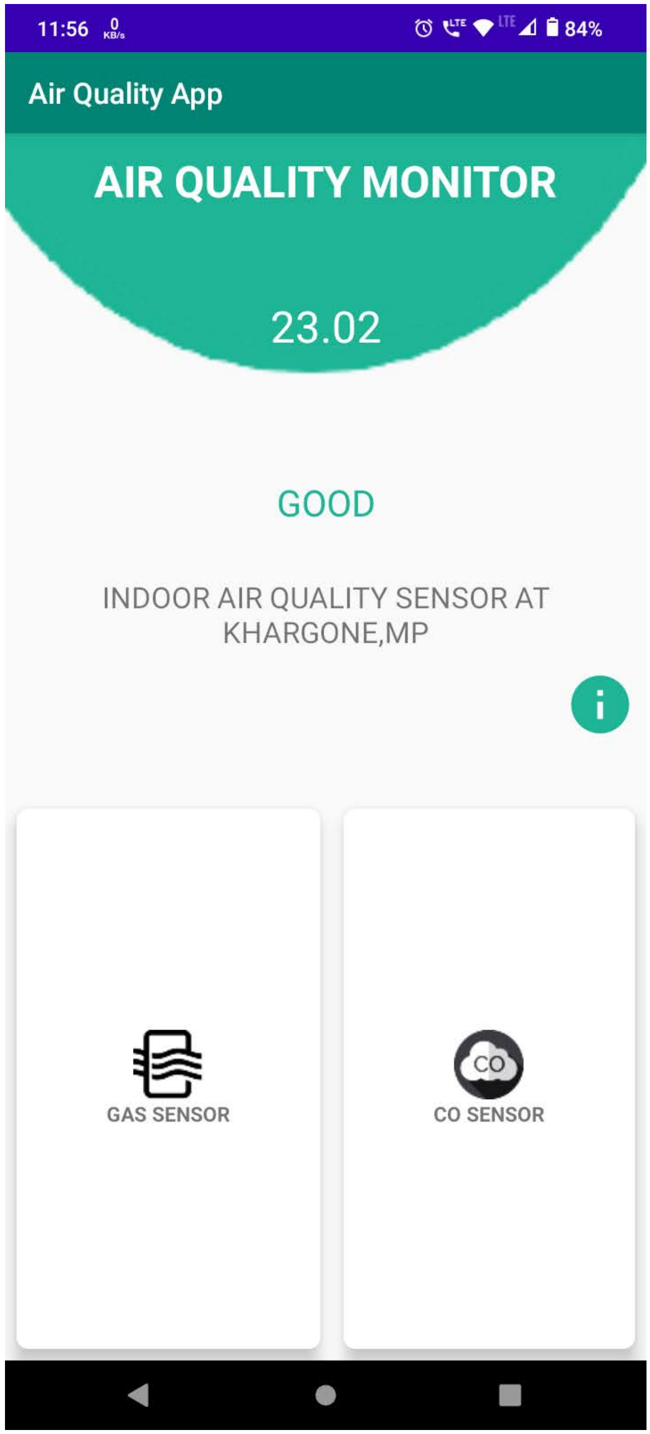


Figure 7: Home Page

Figure 8: Info Page

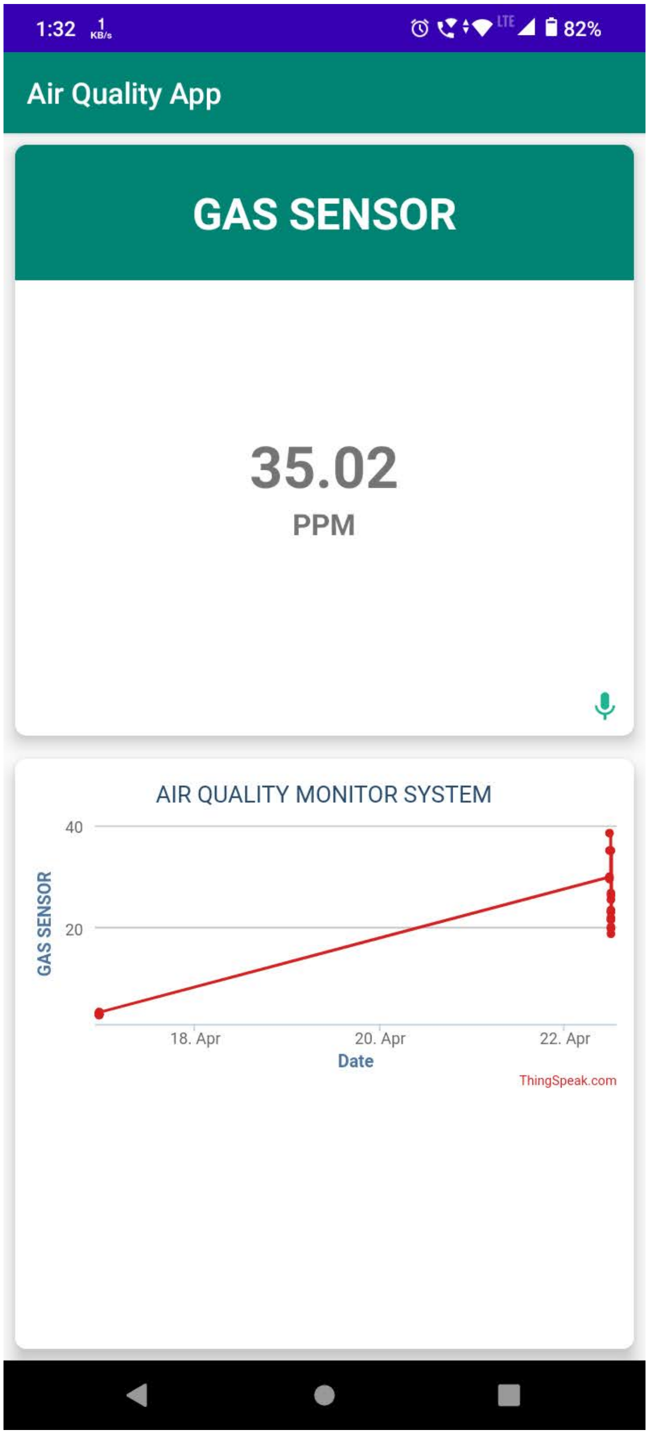


Figure 9: Gas Sensor Page

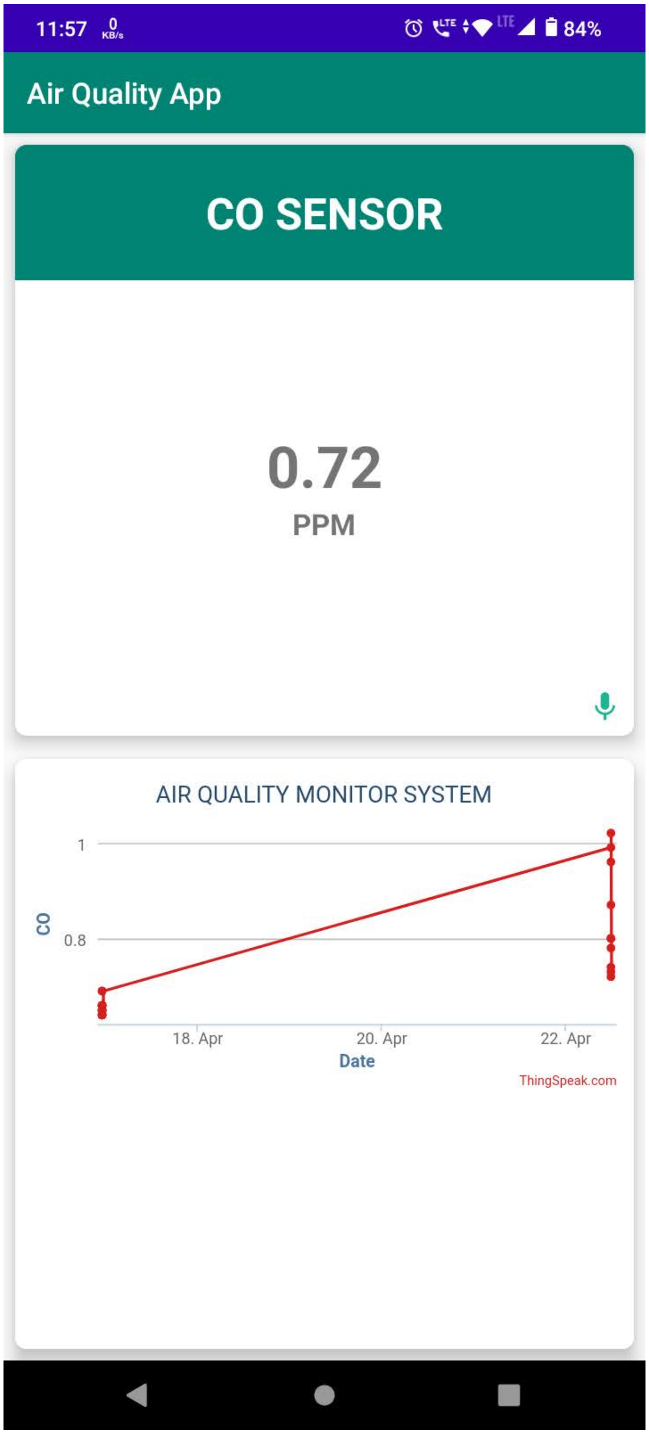


Figure 10: Carbon Monoxide Sensor Page

5 Results Comparison

5.1 Analysis from data from sensors

I collected instances of sensor values and wanted to know min, max and other details. Here they are available in the table given below in different environments.

5.1.1 INDOOR DATA

Here i collected the data for 400 instances keeping the device inside my home.

	entry_id	GAS_SENSOR	CO_SENSOR
count	400.000000	400.000000	400.000000
mean	337.500000	3.812200	0.223975
std	115.614301	1.505959	0.033820
min	138.000000	1.490000	0.150000
25%	237.750000	2.400000	0.200000
50%	337.500000	3.390000	0.230000
75%	437.250000	5.130000	0.240000
max	537.000000	8.130000	0.320000

Figure 11: Statistics about data

GAS SENSOR READINGS

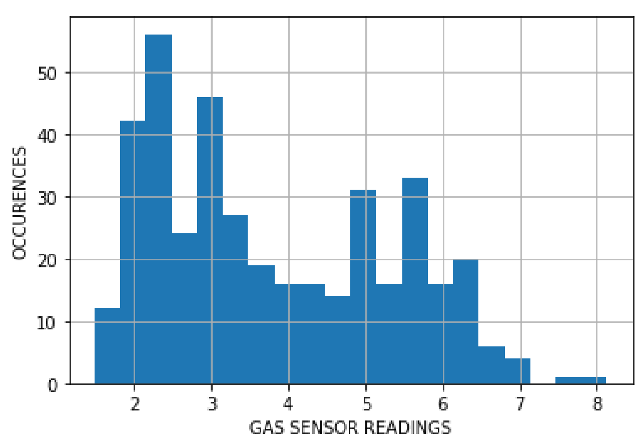


Figure 12: GAS SENSOR READINGS HISTOGRAM

CO SENSOR READINGS

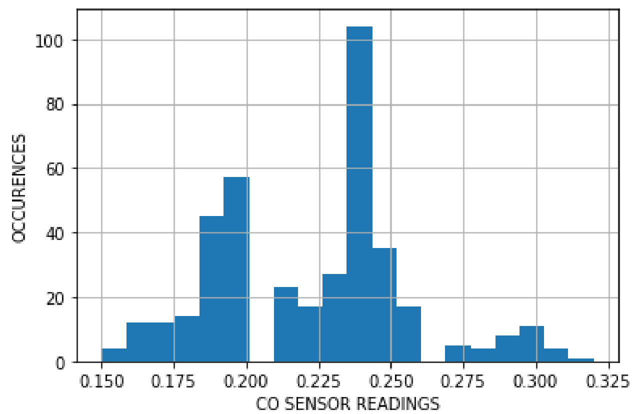


Figure 13: CARBON MONOXIDE SENSORS READINGS HISTOGRAM

5.1.2 OUTDOOR DATA

Here i collected the data of 400 instances keeping the device outside my home.

-	entry_id	GAS_SENSOR	CO_SENSOR
count	400.000000	400.000000	400.000000
mean	737.500000	14.895150	0.546800
std	115.614301	6.027452	0.077183
min	538.000000	9.390000	0.470000
25%	637.750000	11.375000	0.490000
50%	737.500000	12.820000	0.510000
75%	837.250000	15.700000	0.570000
max	937.000000	55.710000	0.870000

Figure 14: Statistics about data

GAS SENSOR READINGS

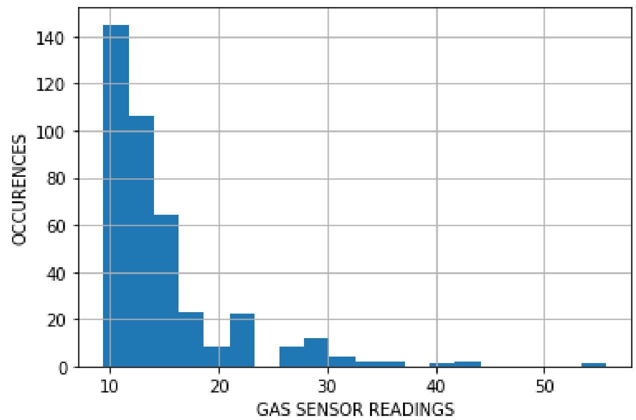


Figure 15: GAS SENSOR READINGS HISTOGRAM

CO SENSOR READINGS

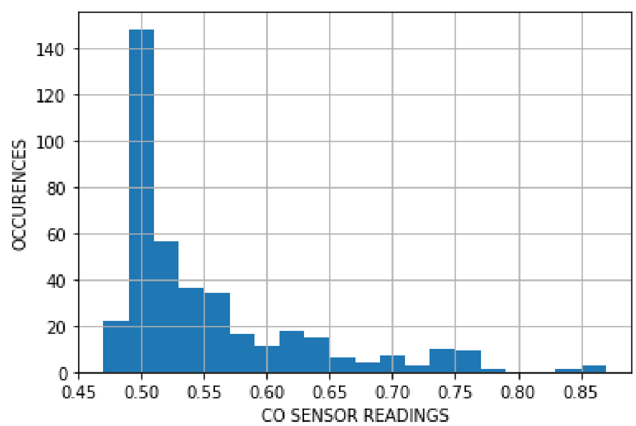


Figure 16: CARBON MONOXIDE SENSORS READINGS HISTOGRAM

5.1.3 SMOKE ENVIRONMENT DATA

Here i collected the data of 30 instances after doing smoke inside my home.

	entry_id	GAS_SENSOR	CO_SENSOR
count	30.000000	30.000000	30.000000
mean	965.500000	67.918333	1.442000
std	8.803408	30.965186	0.782139
min	951.000000	28.510000	0.620000
25%	958.250000	42.730000	0.865000
50%	965.500000	57.755000	1.080000
75%	972.750000	85.670000	2.080000
max	980.000000	163.150000	3.960000

Figure 17: Statistics about data

GAS SENSOR READINGS

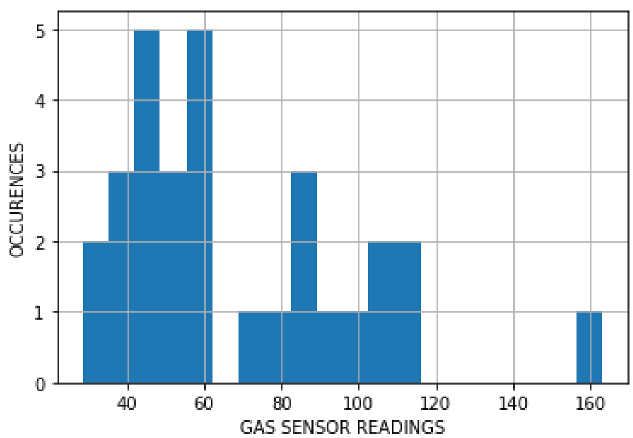


Figure 18: GAS SENSOR READINGS HISTOGRAM

CO SENSOR READINGS

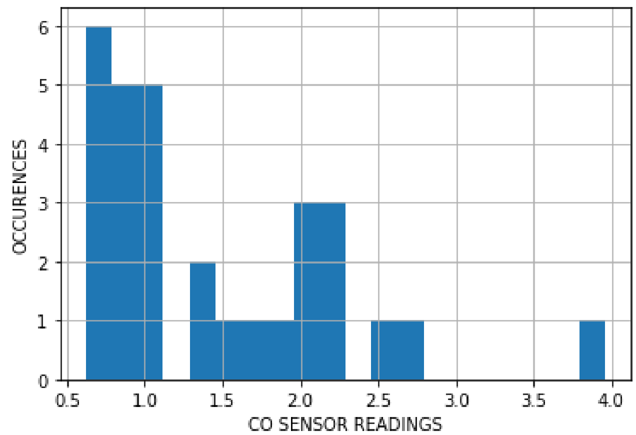


Figure 19: CARBON MONOXIDE SENSORS READINGS HISTOGRAM

5.1.4 COMPARISON OF SENSORS IN DIFFERENT ENVIRONMENTS

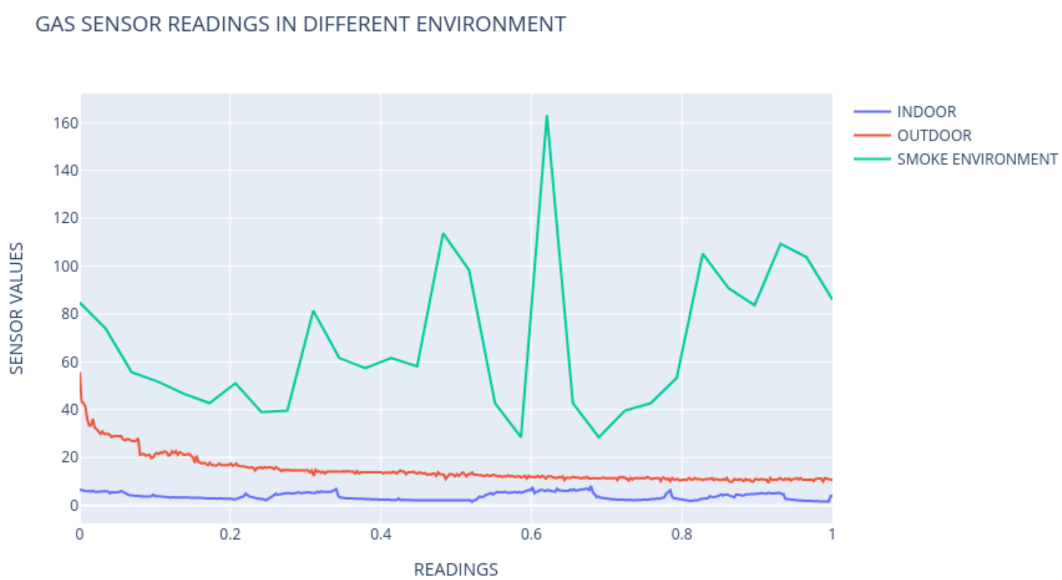


Figure 20: Comparison on values of gas sensor on different environments

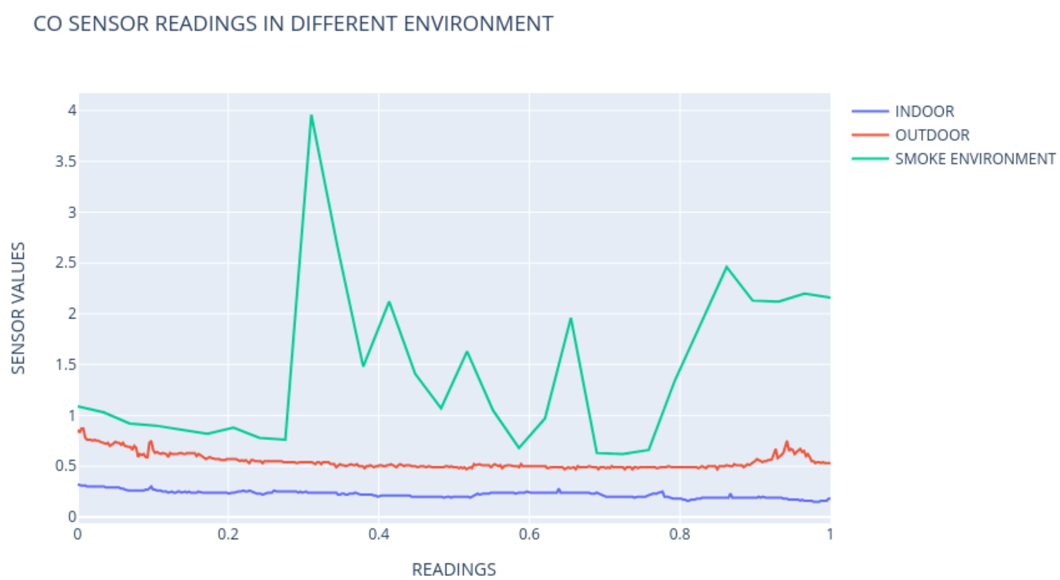


Figure 21: Comparison on values of CO sensor on different environments

Here as we can see the values of gas sensor as well as CO sensor are minimum inside and generally more in the outside and highest incase of smoke environment. Incase of smoke the readings were in the range of 67 to 163 with their level of concern ranging between moderate, unhealthy for sensitive groups and unhealty.

5.1.5 Thingspeak data pushing performance

I took data of 400 consecutive uploads on thingspeak and found out on average it was taking 43.5 seconds to push data to cloud meanwhile i was sending data after every 1.5 seconds so thingspeak is not suitable when when we want to use high velocity data upload on the internet.

5.1.6 Base Paper Comparison

- Sensors Cost : In base paper they have used Alphasense brand sensors which are very costly compared to the Gas sensors in this project.
- Power Supply : In base paper power supply was 0.5 Watt using solar panel battery and in this project power supply is 5V from USB port and power supply is 2.5 Watt using laptop USB.
- Sensors Used : In base paper sensors used are NO2 and O3 and here i used Gas and CO sensor.
- Microcontroller Used : In base paper they have used raspberry pi and in this project Arduino Board was used.
- Cloud Service : In base paper they have used KairosDB & Graphana and in this project ThingSpeak was used.
- Total Cost of Node : 500$ for complete setup in base paper and 50$ in this project for one node.
- Web Monitoring : Both projects have web monotoring.
- Mobile App Monitoring : In base paper they have not provided with mobile app for quick monitoring and in this project mobile app is also there.

Comparison		
Property	Base Paper	Ours
Sensors Cost	50$	5$
Power Supply	0.5Watt	2.5Watt
Sensors Used	NO2,O3	Gas,CO
Microcontroller	raspberry pi	Arduino UNO
Cloud Service	KairosDB	ThingSpeak
	Graphana	
Total Cost of Node	500$	50$

5.2 Market Analysis

Comparison			
Device Name	Price (Rupees)	Mobile App	Link
Kaiterra Laser Egg 2 Air Quality Monitor	9,990	No	tinyurl.com/34wdpsr6
Prana Air Quality Monitor	6,490	Yes	tinyurl.com/vtvptew7
Xuuxu Air Quality Monitor	3599	No	tinyurl.com/mae64426
SMILEDRIVE® Portable Air Quality Pollution Meter	7299	No	tinyurl.com/wsp5wsdm
AIRATOM Air Quality Monitor	8,999	No	tinyurl.com/kfz6nkvs
HealthInnovative G7015	5,149	No	tinyurl.com/hcparbmd
Crusaders AQI	4999	No	tinyurl.com/3vydt6pv
IQAir Air Visual Air Quality Monitor	25,300	No	tinyurl.com/4j7nt84x
Airveda Air Quality	9,499	Yes	tinyurl.com/mcx3reh6
Airveda Air Quality With humidity	19,008	Yes	tinyurl.com/3vyzb7ev
Our Air Quality Monitor	3,000	Yes	

6 Conclusion

I have made an end to end air quality monitoring system. Now a days, there is a need of this kind of sensors because pollution is increasing day by day.

The cost of the sensors should be reduced to as minimum as possible. Here the cost of overall system is very less as compared to any other air monitoring system in the market. I have used Arduino with esp8266 in comparison to Raspberri Pi in base paper, so cost for board is reduced. Most of the commercial air quality monitoring devices doesn't give you monitoring via an android app. I have tried to keep the hardware expenses at minimum but it can be further reduced by using cheaper hardware or different programmable boards. Now a days, There is age of smart systems and with the advancement of IOT we want everything at our fingertips so an Android App is ideal choice for monitoring purpose. Also I used thingspeak as data aggregation platform. This platform is easy to use, but it is little bit slow. Also i used the free version of thingspeak which has a daily limit of fixed messages.

Furthermore, this project can be used by people in cities where Air Pollution is very high. In the price range this is the one of the best option for air quality monitoring. Also the monitoring happens in real time.

7 Applications

If we extend this system possible applications can be :

- Smart monitoring of house in case of gas leak which can cause fire.
- Smart notifications for people who are sensitive of bad air quality can be notified.
- Smoke detection from crop burning in nearby areas.

- Connect with smart devices to close your windows automatically if smoke is coming from outside.
- Air Quality monitoring at large scale in case of smart cities along with humidity and temperature.

8 Limitations & Future Work

Cost of hardware can be reduced further by using cheaper hardware or other NodeMCU based devices which can bring down cost further.

For Monitoring of multiple sensors situated at multiple locations the current thingspeak channel will not scale. A Database for handling data from all the different air quality sensor monitor units can be created.

References

[1] Rashmi Ballamajalu, Srijith Nair, Shayal Chhabra, Sumit K. Monga, Anand SVR, Malati Hegde, Yogesh Simmhan, Anamika Sharma, Chandan M. Choudhary, Ronak Sutaria, Rajesh Zele, Sachchida N. Tripathi.(2018) *Toward SATVAM: An IoT Network for Air Quality Monitoring.* .CoRR, abs/1811.07847., .

[2] Y. Simmhan (2019)*SATVAM: Toward an IoT Cyber-Infrastructure for Low-Cost Urban Air Quality Monitoring* .doi: 10.1109/eScience.2019.00014.

[3] Kinnera, Bharath Kumar Sai Subbareddy, Somula & Luhach, Ashish. (2019). Scalable Computing: Practice and Experience. *IOT based Air Quality Monitoring System Using MQ135 and MQ7 with Machine Learning Analysis.* 20. 599-606. 10.12694/scpe.v20i4.1561.

[4] Sirdah MM, Al Laham NA, El Madhoun RA. Possible health effects of liquefied petroleum gas on workers at filling and distribution stations of Gaza governorates. East Mediterr Health J. 2013 Mar;19(3):289-94. PMID: 23879082.

[5] Cho, R. (2018, June 26). What you should know about air quality alerts. Retrieved from https://phys.org/news/2018-06-air-quality.html

[6] Air Quality Index (AQI) Basics
`https://www.airnow.gov/aqi/aqi-basics/`

[7] Rushikesh, R., Sivappagari, C. M. R. (2015, October). Development of IoT based vehicular pollution monitoring system. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 779-783). IEEE.

[8] Arduino UNO Wikipedia
`https://en.wikipedia.org/wiki/Arduino_Uno`

[9] MQ135 Gas Sensor

`https://components101.com/sensors/mq135-gas-sensor-for-air-quality`

[10] MQ7 CO Sensor

`https://www.sparkfun.com/products/9403`

[11] ESP8266

`https://www.electronicshub.org/esp8266-arduino-interface/`

[12] Arduino IDE

`https://www.arduino.cc/en/Guide/Environment`

[13] Android Studio IDE

`https://developer.android.com/studio`