ESW Project

Indoor Air Pollution

Team Name: MR VR

Project Mentor: Prof. Sachin Chaudhari
Project TA: Rishikesh Bosh

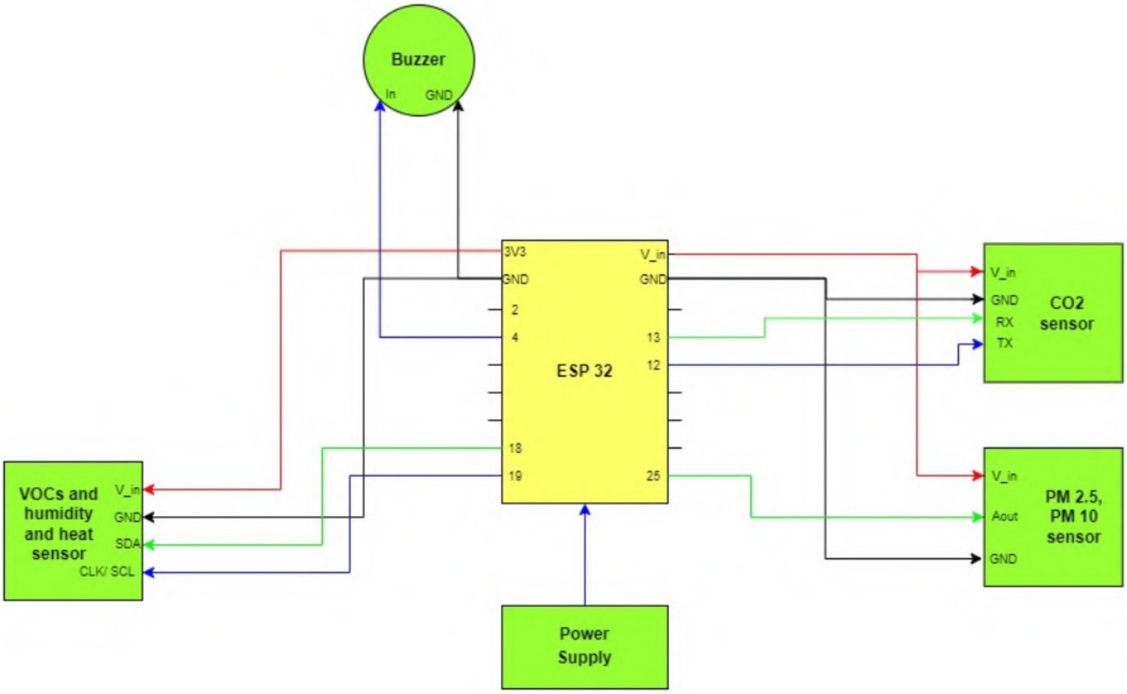
Team Members:
Mayank Shukla
Rishabh Srivastava
Rishabh Jain
Vaibhav Singh Tomar

Motivation:

During Covid-19, We’ve kept ourselves safe and healthy by careful protection yet without effective information. Now it’s time to be out of the woods and learn indoor air quality solution. We understand about outdoor air pollution and get aware of that and often concern about that and neglect the indoor air pollution. But sometimes it becomes more dangerous and is it confined within closed doors. People spend 90% of their time indoors and so the health problems that result from exposure to indoor pollution increases, the contaminants in indoor air could damage people’s health.

To overcome and detect this problem, we are creating a indoor air pollution detector so that we can analyse the indoor data and extent of pollution level there. This will be helpful in deciding which factor is affecting the indoor environment and can be rectified without further ignorance.

Circuit Diagram of Hardware:



CIRCUIT DIAGRAM

Methodology:

1. Understanding the purpose:

Our first task was to understand the purpose of our project. We get to know the importance indoor air pollution detector and what all factors it will depend on.

2. Project Strategy:

After understanding the project, we strategize our project in which we decided what will be our final deliverables and what will be the route to reach the final deliverable.

Work division was done in this phase and tentative timeline was give for each work.

3. Hardware Requirements:

Collect the required hardware for the project.

i. Sensors:

* Particulate Matter (PM) Sensor which will measure the indoor PM2 and PM10.
* Carbon Dioxide (CO_2) Sensor to measure the CO2 concentration.
* Temperature, Humidity and Volatile Organic Solids (VOCs) Sensor to measure the indoor tempereature, humidity and VOCs.

ii. Zero PCB

iii. Breadboard

iv. ESP32 as microcontroller.

v. Wires

vi. Soldering materials.

4. Arduino Coding:

Requirement for all the sensors are coded.

5. Code and Circuit design checking:

Uploaded the code in ESP32 and connect the circuit and checked. Resolved many issues came in this phase like Tx and Rx pin connection, SDA and SCL connection and other similar stuffs.

6. PCB Design:

After checking code and circuit we Created the proper design and based on that be embedded all the sensors in zero PCB and soldered all sensors on zero PCB.

7. Sending data in Thingspeak:

We send the collected data in thingspeak to visualize the data.

8. Connecting to oneM2M:

We connected our device to IIIT oneM2M server and uploaded data on that server.

9. Creating Dashboard:

To visualize the data and make accessible to public we created a simple but informative dashboard. Dashboard shows the live sensor values and plot the graph on collected data.

10. Data Analysis and Security:

Encrypted the data before sending it to oneM2M. Hence, data cannot be breached in between.
Data is analyzed and inferences are made.

11. 3D-enclosure:

Put the final zero PCB into a 3D enclosure which we created by 3D printing.

12. Deployment:

We deployed our final device in Kadamb mess and analyzed data over there.

Deployment Campaign:

This was one of the toughest phase of project. We created our device and was working properly before deployment. When we enclosed it in a 3D enclosure, it sometime stopped working. Again we have to take out of the box and check the entire circuit and sensors.
Many times WiFi connection goes down and we don’t understand the real problem and start checking the entire circuit.

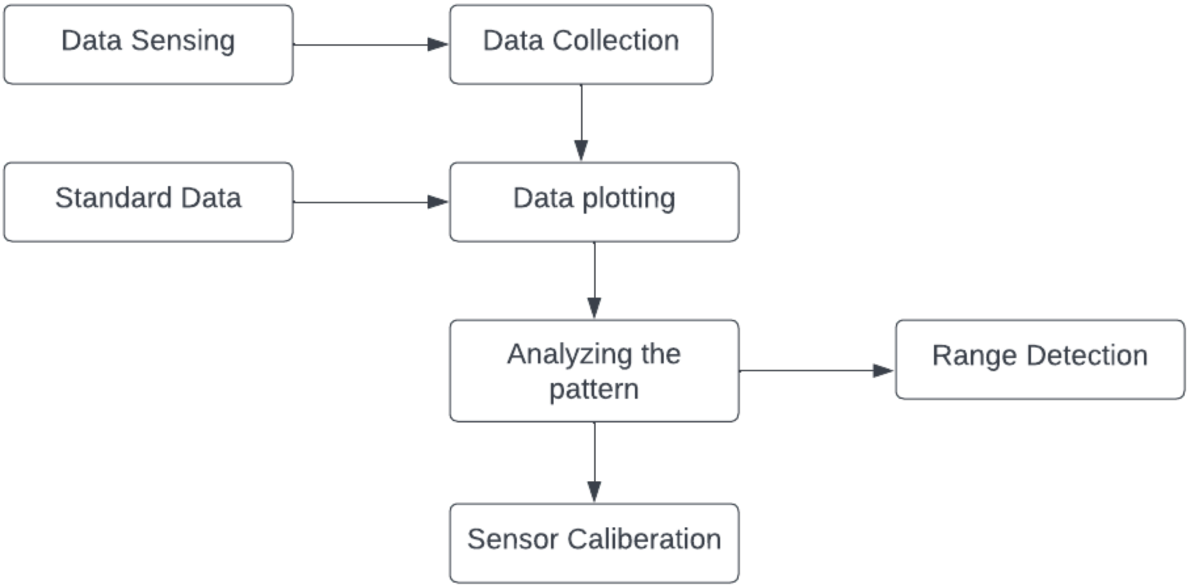
Finally we searched the suitable place for deployment where we can collect the actual data. We installed our device in mess along with Airtel WiFi node which gives the stable connection.
This becomes the successful campaign and now we are collecting data.

Data Validation:

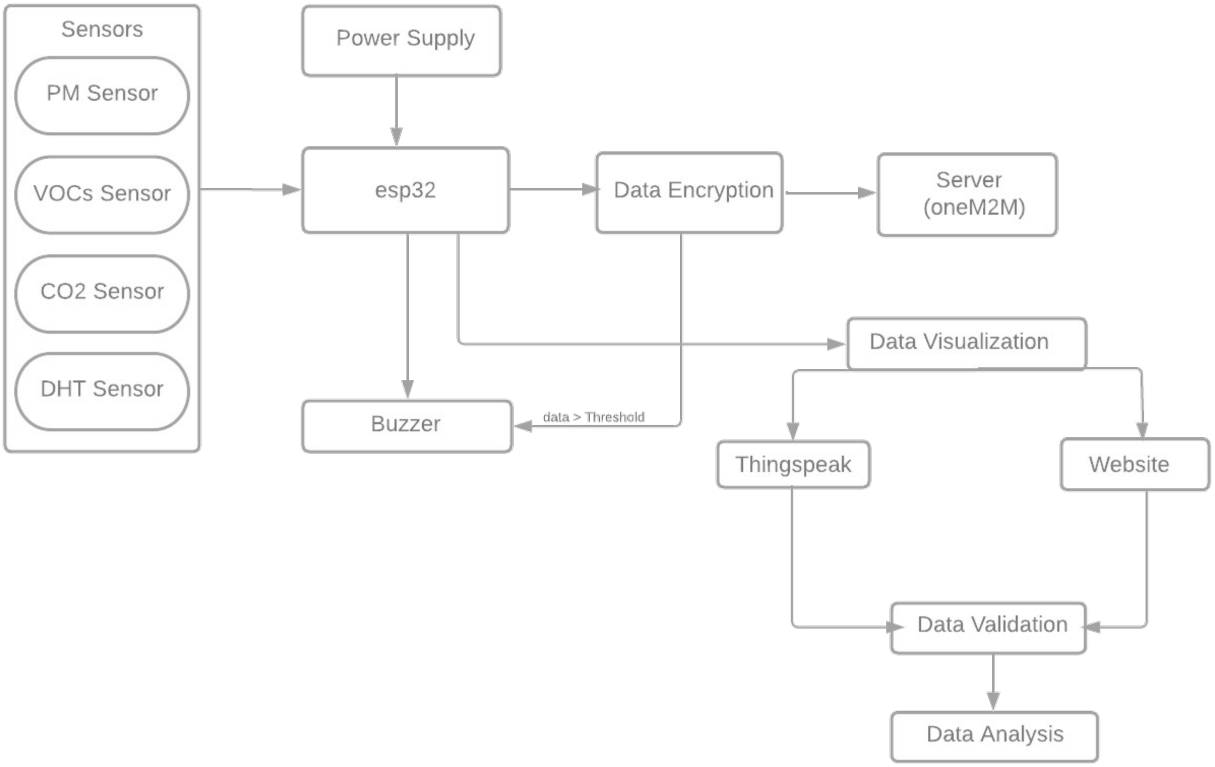
We plotted our data with standard data and checked if we are getting the same data pattern in both. This helped in sensor caliberation.
This phase can help in understanding the sensor defects, circuit defects or code bug if any.

In this phase, we got the clarity about
1. Data types of each sensor values.
2. Data ranges.
3. Data fluctuation and consistency.
4. Null values detection.

Data validation Pipeline:



Work Flow Diagram:



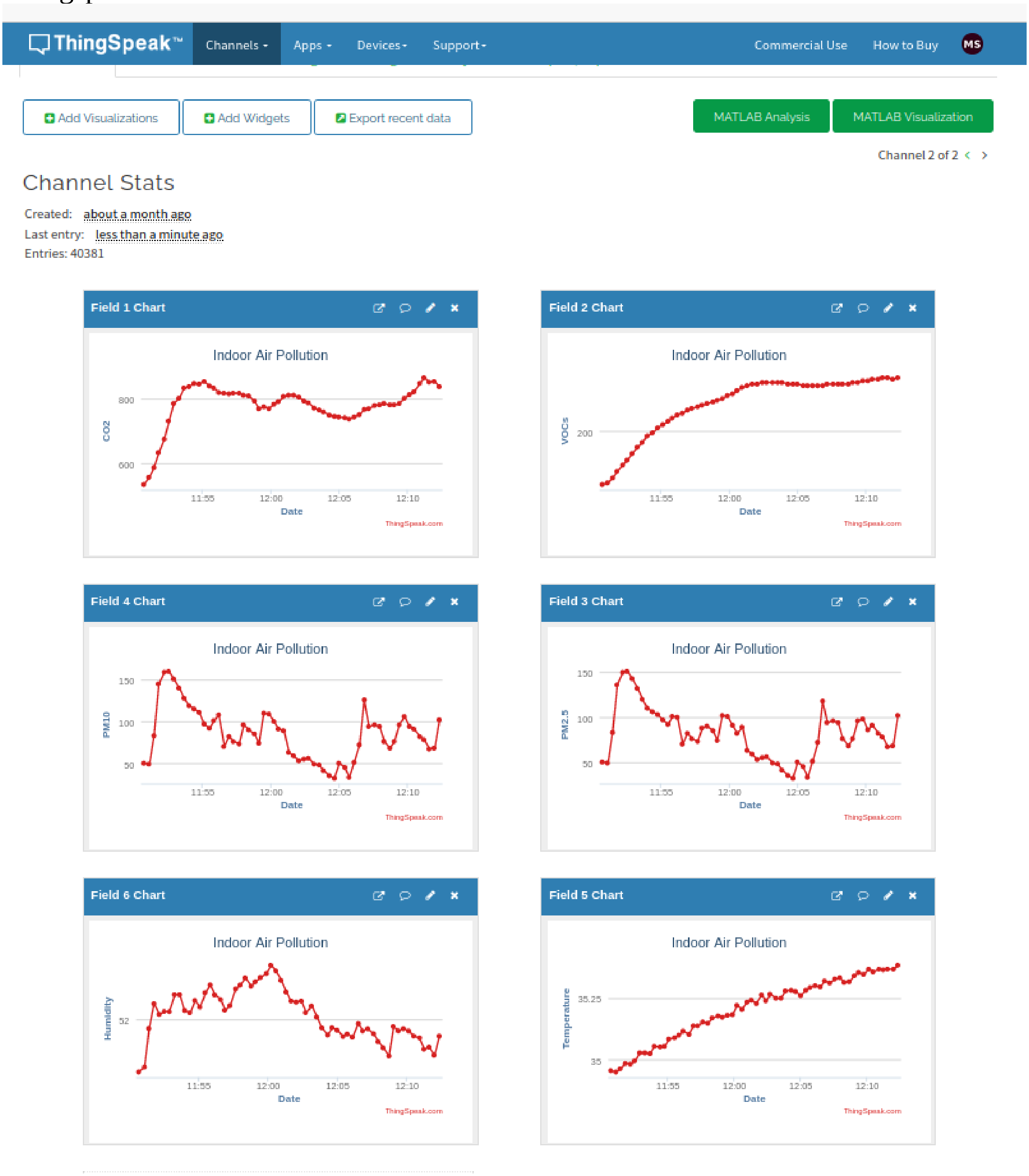
Communication to Thingspeak and oneM2M:

Data is send to both thingspeak and oneM2M.

Data in oneM2M can be accessed by sending the GET request in oneM2M using the API: http://esw-onem2m.iiit.ac.in:443/~/in-cse/in-name/Team-6/Node-1/Data?rcn=4 when using IIIT wifi or LAN

Else use the API for your internet connection:
API: https://esw-onem2m.iiit.ac.in/~/in-cse/in-name/Team-6/Node-1/Data?rcn=4

Thingspeak:



Channel Stats

Created: about a month ago

Last entry: less than a minute ago

Entries: 40381

Development of Dashboard:

Tools and Technologies used:

1. Flask
2. HTML
3. CSS
4. Javascript
5. Highcharts
6. SaSS
7. Thingspeak for database.
8. Heroku for deployment of dashboard.

Functionalites of Dashboard:

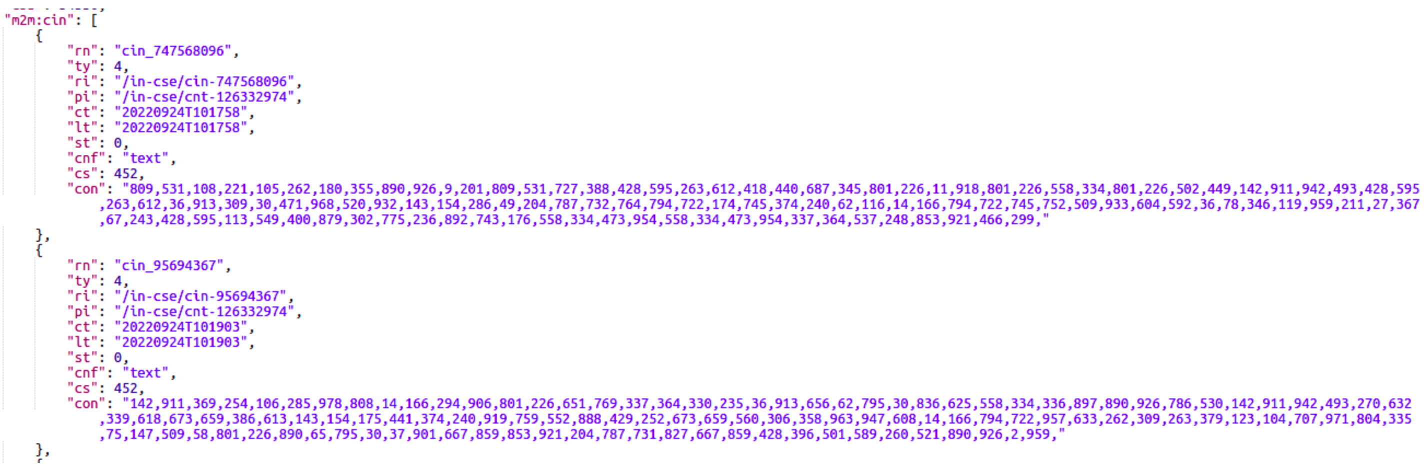
1. Shows the live data of all the sensors.
2. Plot the graph for all the sensors helps in visualization.
3. It contains the Time filter where we can filter data based on given time frame.

IoT security tools:

Used the ECC encryption scheme to encrypt the data before sending it to oneM2M. This scheme is better that most of the other scheme like RSA, AES etc. Because of its resource consumption feature. It takes very less number of bits to give same level of security as compared to others.

It takes 256 bits to give same level of security as given by RSA algorithm which takes 3072 bits.

Data is stored in oneM2M in encrypted format:

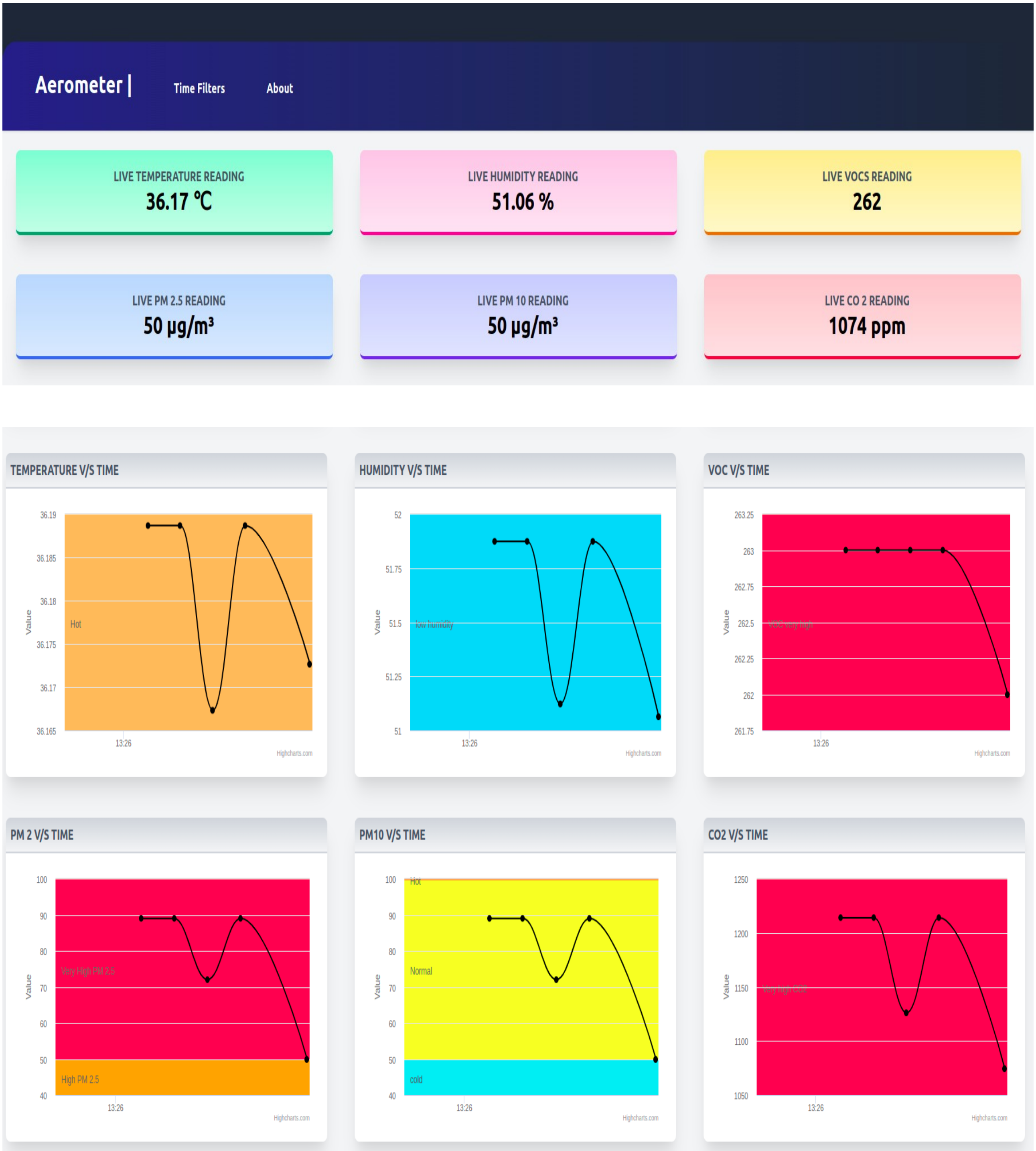


```
"m2m:cin": [
    {
        "rn": "cin_747568096",
        "ty": 4,
        "ri": "/in-cse/cin-747568096",
        "pi": "/in-cse/cnt-126332974",
        "ct": "20220924T101758",
        "lt": "20220924T101758",
        "st": 0,
        "cnf": "text",
        "cs": 452,
        "con": "809,531,108,221,105,262,180,355,890,926,9,201,809,531,727,388,428,595,263,612,418,440,687,345,801,226,11,918,801,226,558,334,801,226,502,449,142,911,942,493,428,595
            ,263,612,36,913,309,30,471,968,520,932,143,154,286,49,204,787,732,764,794,722,174,745,374,240,62,116,14,166,794,722,745,752,509,933,604,592,36,78,346,119,959,211,27,367
            ,67,243,428,595,113,549,400,879,302,775,236,892,743,176,558,334,473,954,558,334,473,954,337,364,537,248,853,921,466,299,"
    },
    {
        "rn": "cin_95694367",
        "ty": 4,
        "ri": "/in-cse/cin-95694367",
        "pi": "/in-cse/cnt-126332974",
        "ct": "20220924T101903",
        "lt": "20220924T101903",
        "st": 0,
        "cnf": "text",
        "cs": 452,
        "con": "142,911,369,254,106,285,978,808,14,166,294,906,801,226,651,769,337,364,330,235,36,913,656,62,795,30,836,625,558,334,336,897,890,926,786,530,142,911,942,493,270,632
            ,339,618,673,659,386,613,143,154,175,441,374,240,919,759,552,888,429,252,673,659,560,306,358,963,947,608,14,166,794,722,957,633,262,309,263,379,123,104,707,971,804,335
            ,75,147,509,58,801,226,890,65,795,30,37,901,667,859,853,921,204,787,731,827,667,859,428,396,501,589,260,521,890,926,2,959,"
    },
```

Data Visualization:

We use the thingspeak and dashboard for visualization.

Dashboard Visualization:



LIVE TEMPERATURE READING
36.17 ℃

LIVE HUMIDITY READING
51.06 %

LIVE VOCS READING
262

LIVE PM 2.5 READING
50 µg/m³

LIVE PM 10 READING
50 µg/m³

LIVE CO 2 READING
1074 ppm

TEMPERATURE V/S TIME

HUMIDITY V/S TIME

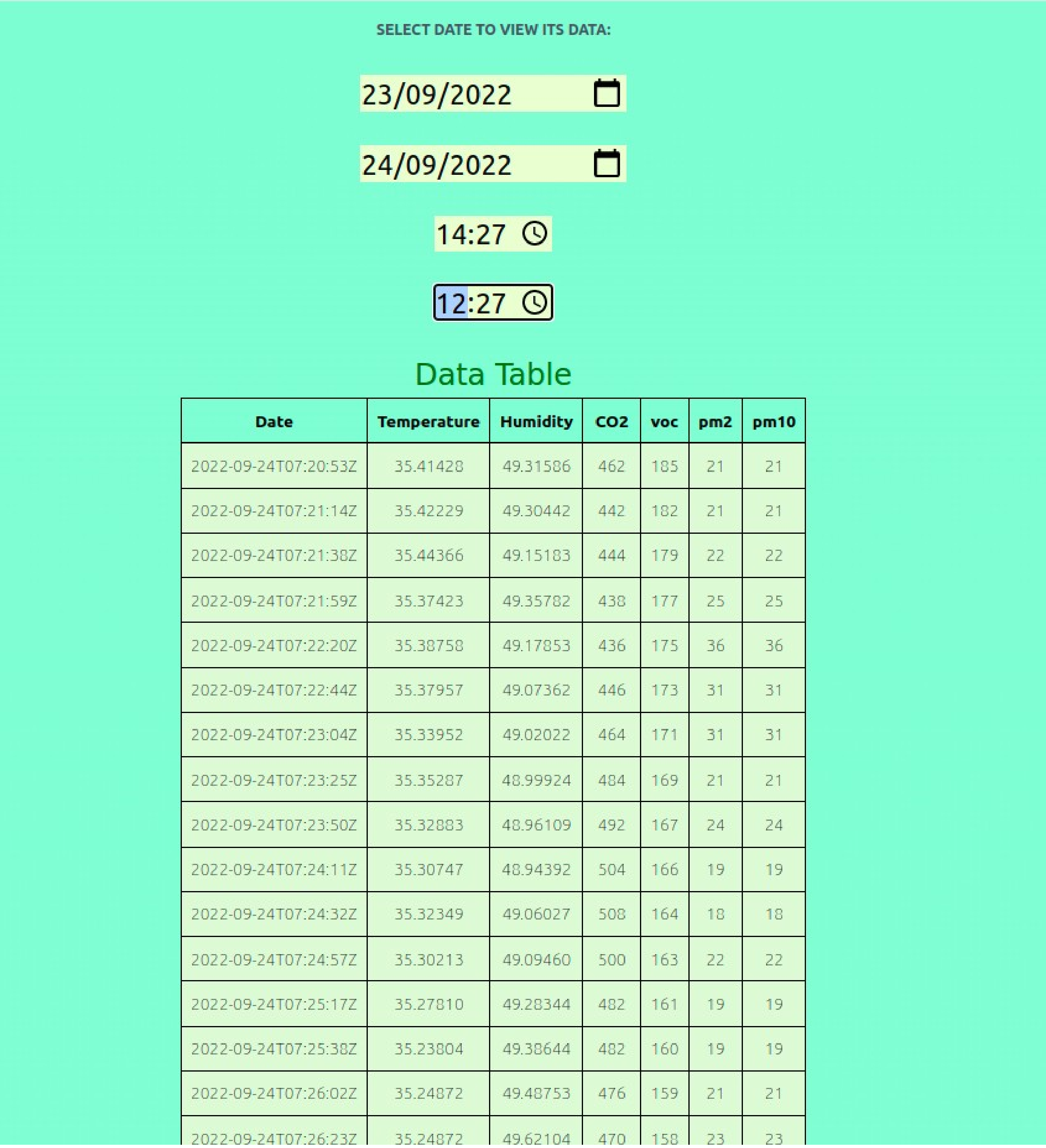
VOC V/S TIME

PM 2 V/S TIME

PM10 V/S TIME

CO2 V/S TIME

Time Filter:



SELECT DATE TO VIEW ITS DATA:

23/09/2022

24/09/2022

14:27

12:27

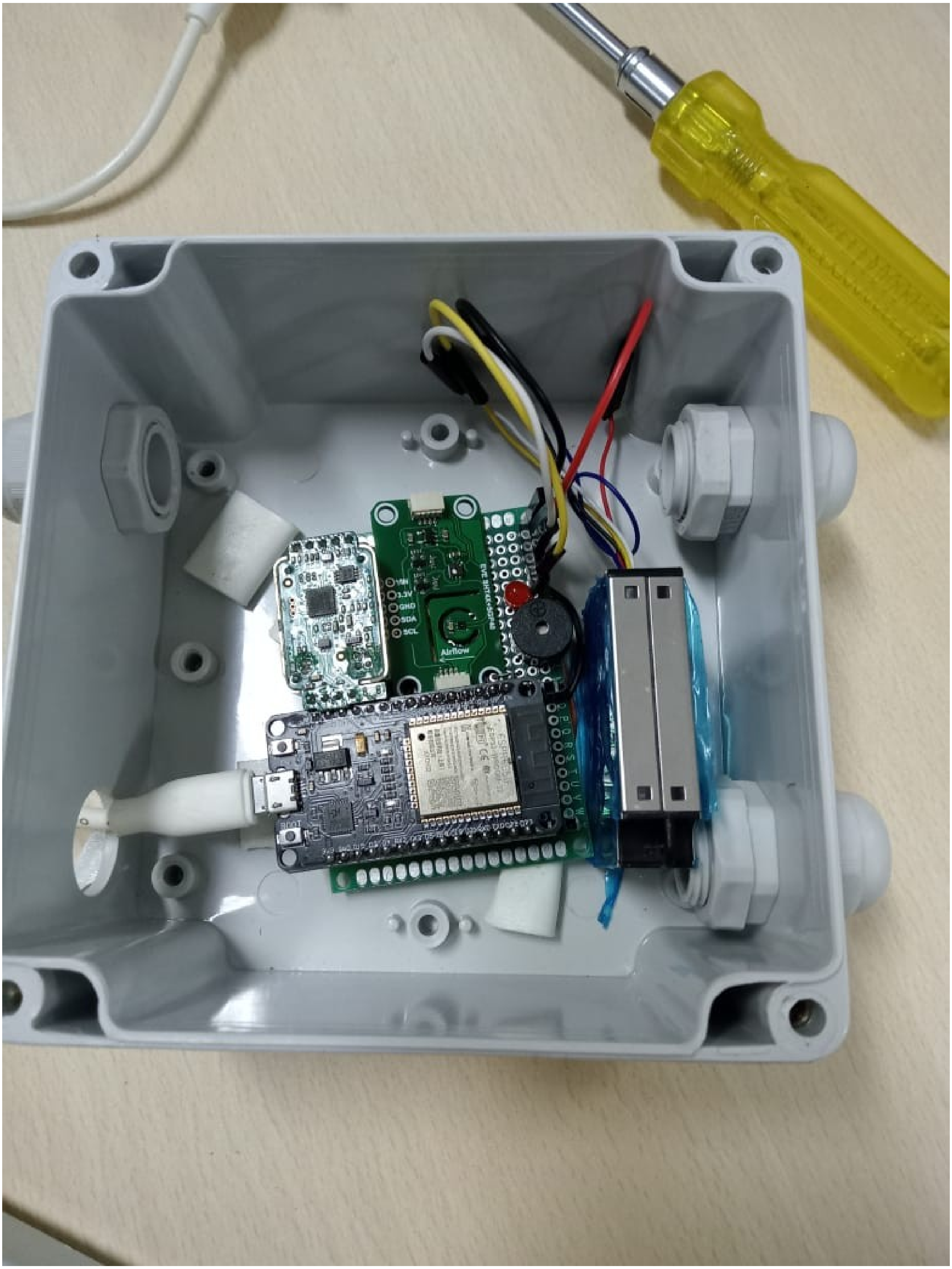
Data Table

Date	Temperature	Humidity	CO2	voc	pm2	pm10
2022-09-24T07:20:53Z	35.41428	49.31586	462	185	21	21
2022-09-24T07:21:14Z	35.42229	49.30442	442	182	21	21
2022-09-24T07:21:38Z	35.44366	49.15183	444	179	22	22
2022-09-24T07:21:59Z	35.37423	49.35782	438	177	25	25
2022-09-24T07:22:20Z	35.38758	49.17853	436	175	36	36
2022-09-24T07:22:44Z	35.37957	49.07362	446	173	31	31
2022-09-24T07:23:04Z	35.33952	49.02022	464	171	31	31
2022-09-24T07:23:25Z	35.35287	48.99924	484	169	21	21
2022-09-24T07:23:50Z	35.32883	48.96109	492	167	24	24
2022-09-24T07:24:11Z	35.30747	48.94392	504	166	19	19
2022-09-24T07:24:32Z	35.32349	49.06027	508	164	18	18
2022-09-24T07:24:57Z	35.30213	49.09460	500	163	22	22
2022-09-24T07:25:17Z	35.27810	49.28344	482	161	19	19
2022-09-24T07:25:38Z	35.23804	49.38644	482	160	19	19
2022-09-24T07:26:02Z	35.24872	49.48753	476	159	21	21
2022-09-24T07:26:23Z	35.24872	49.62104	470	158	23	23

Data Analysis:

Outlier detection is done.
Outlier is removed.
Data cleaning is done.
Inferences are made based on data visualization.

Circuit Photo:



VIN
3.3V
GND
SDA
SCL
Airflow

Conclusion:

Our final device is monitoring indoor air pollution. It sense the data at the deployed region and sends it to thingspeak and oneM2M. We can visualize the data in thingspeak. Take the data from there and do data validation and perform analysis. Final data is visualized in our dashboard which is publicly available where anyone can scan the QR code and access the dashboard.

Final Code base:

https://github.com/ed-mayank/ESW-Project