Sistem de monitorizare a calității aerului

Autor:
Diana-Elena Popescu

Motivatie

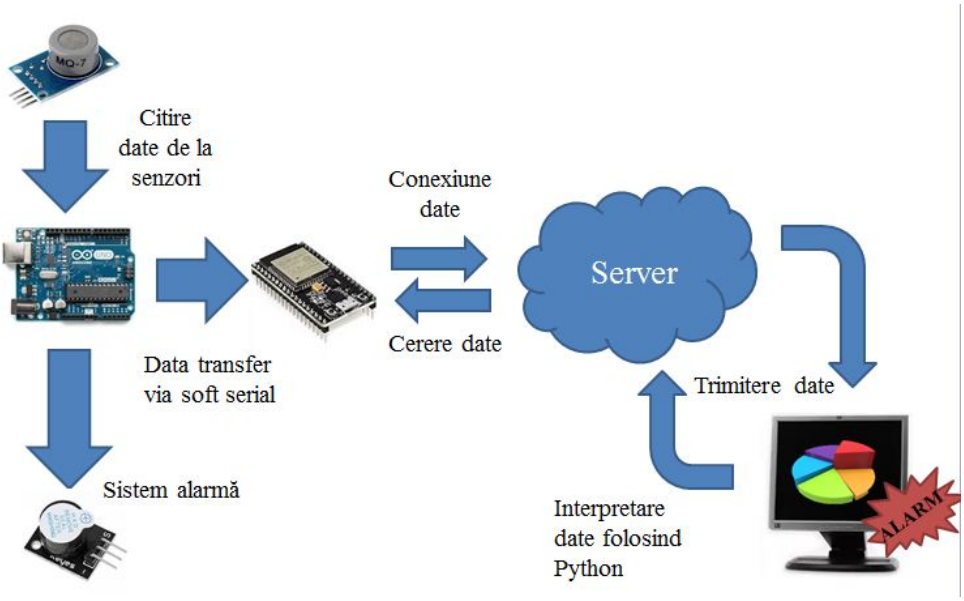
Una dintre marile probleme ale lumii moderne este poluarea aerului, care ne afectează viața și activitatea zilnica, provocând de la oboseala, la amețeli și stări de rău. Prin faptul că suntem nevoiți să ne desfășuram activitățile înăuntru, mi-am amintit cat de mult contează calitatea aerului din mediul în care lucram și trăim. Ceea ce trebuia sa fie un modul de monitorizare a calității aerului și a nivelului de poluare s-a adaptat izolării la domiciliu și a devenit un ansamblu de senzori meniți să măsoare nivelul de CO2, CO din încăpere, dar totodată să declanșeze un mecanism de alarmă în cazul în care se detectează fum sau scurgeri de gaz.

Introducere

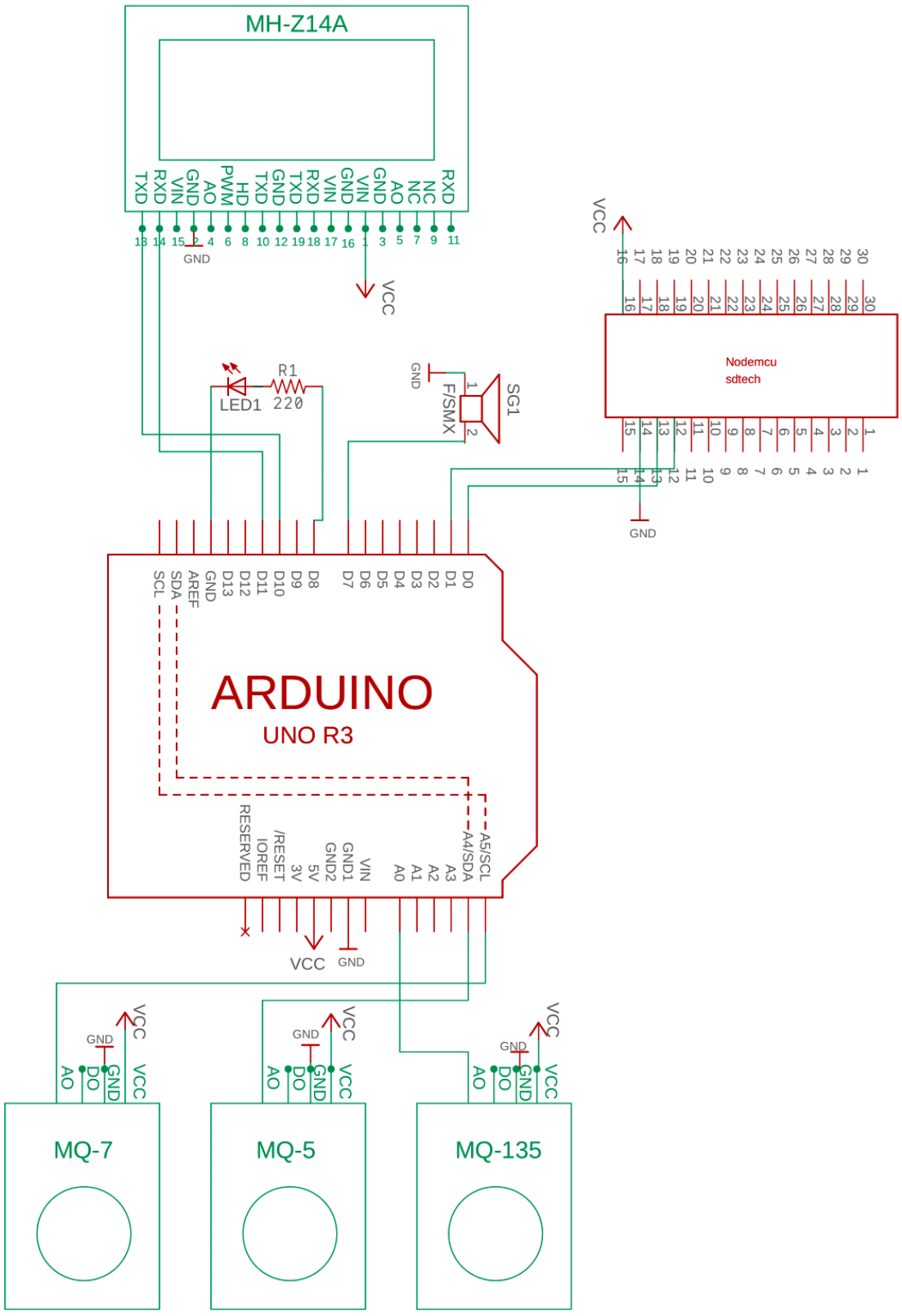
Mi-am dorit sa realizez un modul complet de monitorizare a calității aerului nu doar din punct de vedere al componenței chimice, cât și din perspectiva siguranței persoanelor din mediul respectiv. Astfel datele colectate pot fi accesate remote, printr-o interfață prietenoasă, dar si local. Modulul are integrat un sistem de avertizare sonoră in cazul in care unul dintre senzori detectează depășirea limitei normale.

Descriere generală

Sistemul creat este împărțit ideatic în mai multe componente: senzori și citirea datelor, conexiunea plăcută ArduinoUno și NodeMCU și transmiterea datelor, conexiune server, obținerea și prelucrarea datelor, sistem de alarmă.



Hardware Design



Componentă	Număr
Arduino Uno	1
NodeMCU	1
Senzor MQ5	1
Senzor MQ7	1
Senzor MQ135	1
Senzor MH-Z14A	1

Led rosu	1
Buzzer activ	1
Baterie externa 5V	1
Rezistență 1k	1

Senzori folositi

MH-Z14A NDIR CO2 Sensor

Senzorul MH-Z14A are o precizie destul de mare la un pret accesibil. Range-ul de valori pe care il poate masura este de 0-10000ppm (parts per million). Senzorul dispune de 3 optiuni de citire al output-ului : UART, DAC, PWM. Dupa testarea celor 3 variante am ajuns la urmatoarele concluzii:

1. Conform DAC, camera mea [1)] are un nivel de CO2 comparabil cu o intersectie din Bucuresti la ora de varf.
2. Folosind PWM se obtin date corecte, desi 1/20 de citiri este eronata (foarte mica fata de celelalte citiri, uneori negativa).
3. Interfata seriala (UART) a avut rezultatele cele mai bune si consistente, fara diferente semnificative intre citiri consecutive.
4*. Desi in datasheet este specificat timpul de preheat ca fiind 3 min, valorile obtinute incep sa se apropie de adevar si sa se stabilizeze abia dupa ~1h.

Verificare: Am expirat deasupra senzorului aproximativ 1 minut si valorile au crescut vizibil. In 5 minute acesta isi revine.
Am decis deci, sa folosesc Interfata Seriala pentru a face citiri pentru acest proiect.
Acest senzor costa ~20 dolari si se poate achizitiona de pe https://www.aliexpress.com.
Datasheet-ul se poate gasi aici.

MQ-7

Senzorul MQ-7 masoara concentratia de particule de CO (monoxid de carbon) din aer.
Range-ul de valori pe care il poate masura senzorul MQ7 este de 0-10000ppm CO.
Citirile se fac folosind pinul analogic.
Valorile normale sunt de 9-10 ppm pentru exterior si maxim 40-50 ppm in interior. Daca se depaseste aceasta valoare, expunerea prelungita poate produce dureri de cap, ameteli, eventual moarte (1600ppm cu expunere 1h).
Acest senzor costa ~1$ si se poate achizitiona de aici.
Datasheet-ul se poate gasi aici.

MQ-5

Senzorul MQ-5 este sensibil la prezenta gazului natural, LPG.
De asemenea, el are un răspuns rapid, este stabil si de lunga durata, iar integrarea lui in circuit este destul de simplă.

Acest senzor costa ~1 dolar si se poate achizitiona de aici.
Datasheet-ul se poate gasi aici.

MQ-135

Acest senzor se foloseste pentru a masura “calitatea aerului”, la general vorbind. Este sensibil la mai multe tipuri de gaze, dar nu poate face o diferentiere precisa. Detectează NH3, NOx, alcool, Benzen, fum, CO2. Outputul acestuia creste odata cu ratia particulelor din aerul la care este expus.
Acest senzor costa ~1 dolar si se poate achizitiona de aici.
Datasheet-ul se poate gasi aici.

Software Design

Pentru cod, am folosit ca mediu de dezvoltare Arduino IDE.

Biblioteci utilizate au fost:

Arduino	NodeMCU
SoftwareSerial.h	ESP8266WiFi.h
ArduinoJson.h	SoftwareSerial.h
	ArduinoJson.h
	SPI.h

- **Senzori si citirea datelor**

Pregătirea înainte de citire a datelor: Senzorii MQ au nevoie de preheating înainte de citirea datelor, dar și de calibrare. Aceasta a constat în aplicarea unor tensiuni de 1,4V si 5 V pe pinii analogici ai senzorilor pentru intervale de 1 min, respectiv 90 de secunde înainte de realizarea citirii datelor. Senzorul MH-Z14A sta la preheat 1h. Acesta nu necesită calibrare.

```
val135 = analogRead(pinMQ135);
val7 = analogRead(pinMQ7);
val5 = analogRead(pinMQ5);
MHZ14A = readPPMSerial(); \\ folosim seriala pt citirea datelor
```

- **Conexiunea plăcuță ArduinoUno și NodeMCU și transmiterea datelor**

Conexiunea a fost realizată prin Software Serial, conectând pinul RX, respectiv TX al plăcuței ArduinoUno la pinul TX, respectiv RX al NodeMCU-ului. Pentru a evita scrieri si citiri succesive asincrone, predispuse la întârzieri, datele au fost trimise sub forma unui JSON. Integritatea structurii este verificată la NodeMcu. Dacă este corupta/invalidă, se așteaptă următoarea structură.
ArduinoUno

```
StaticJsonBuffer<5000> jsonBuffer;
JsonObject& root = jsonBuffer.createObject();
root["val 135"] = val135;
root["val 7"] = val7;
root["val 5"] = val5;
root["MH-Z14A"] = CO2;
if (Serial.available() > 0) {
  root.printTo(Serial);
}
```

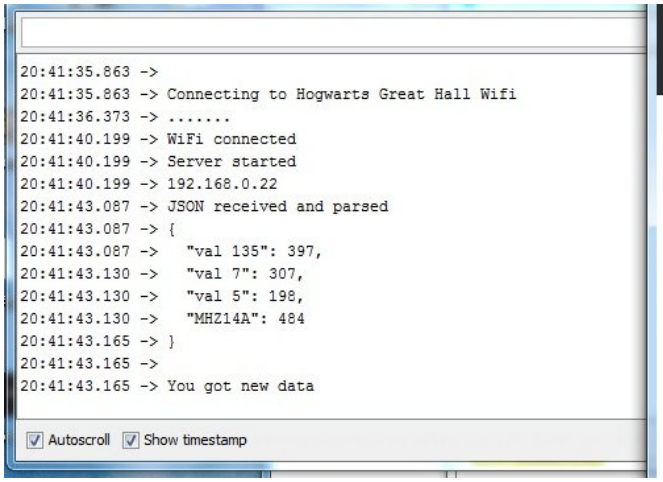
NodeMCU

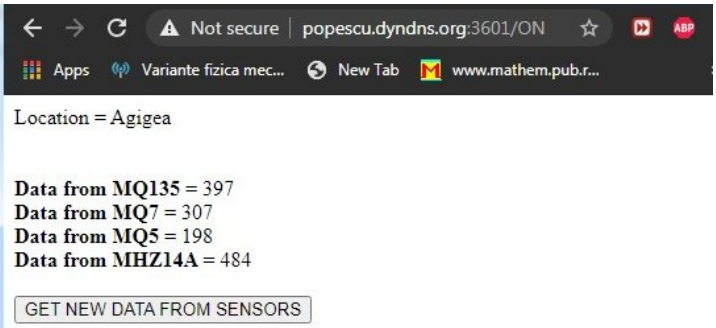
```
if (Serial.available() > 0) {
    StaticJsonBuffer<5000> jsonBuffer;
    JsonObject& root = jsonBuffer.parseObject(Serial);
    if (root == JsonObject::invalid()){
      JsonObject& root = jsonBuffer.parseObject(Serial);
      return;
```

```
    }
```

- **Conexiunea NodeMCU - Server**

NodeMCU a fost conectat la WiFi-ul personal. Din setările routerului am aflat range-ul DHCP-ului și am atribuit plăcuței un IP înafara acestuia. IP-ului modulului i-am alocat un port care a fost deschis ulterior folosind funcția „Port Forwarding” a router-ului. Am achiziționat un abonament la DynDNS pentru a face posibilă accesarea IP-ului public printr-un domeniu ușor de memorat. Pentru a avea o reprezentare vizuală la nivelul modulului, am integrat un LED care se aprinde de fiecare dată când este realizat un request din partea serverului.





- **Obținerea și prelucrarea datelor**

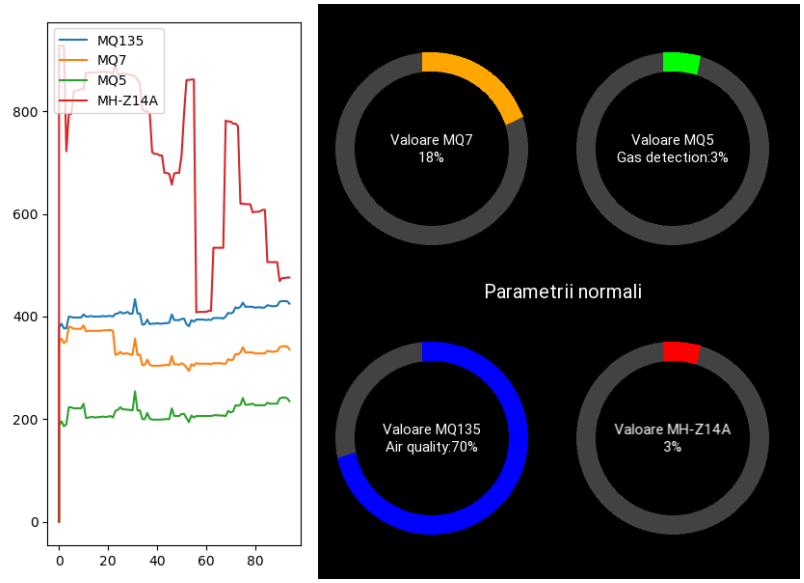
Aplicația ce prelucrează datele a fost scrisă în limbajul Python, utilizând framework-ul Kivy și modulul Selenium. Există două părți principale:

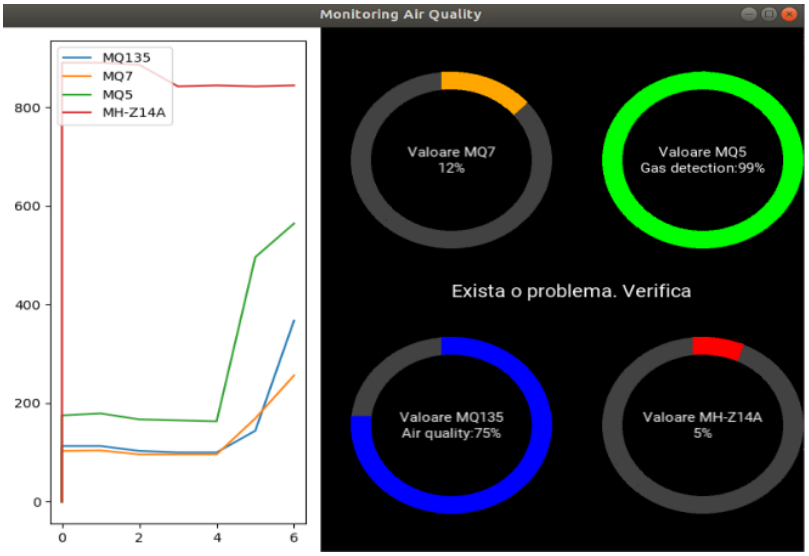
- Preluarea datelor de la server:

presupune conectarea la serverul unde sunt stocate datele, folosind Chrome WebDriver rulat în modul headless. Datele sunt parsate, cu ajutorul unui regex, apoi valorile sunt atașate unui fișier CSV, pentru a menține un log în cazul în care apar probleme în cadrul funcționării părții de interpretare a datelor, acestea putând fi ușor fiind interpretate ulterior. Această măsură de siguranță permite și citirea unor date deja primite de la server, pentru eventuale verificări ulterioare.

- Analiza datelor de la server:

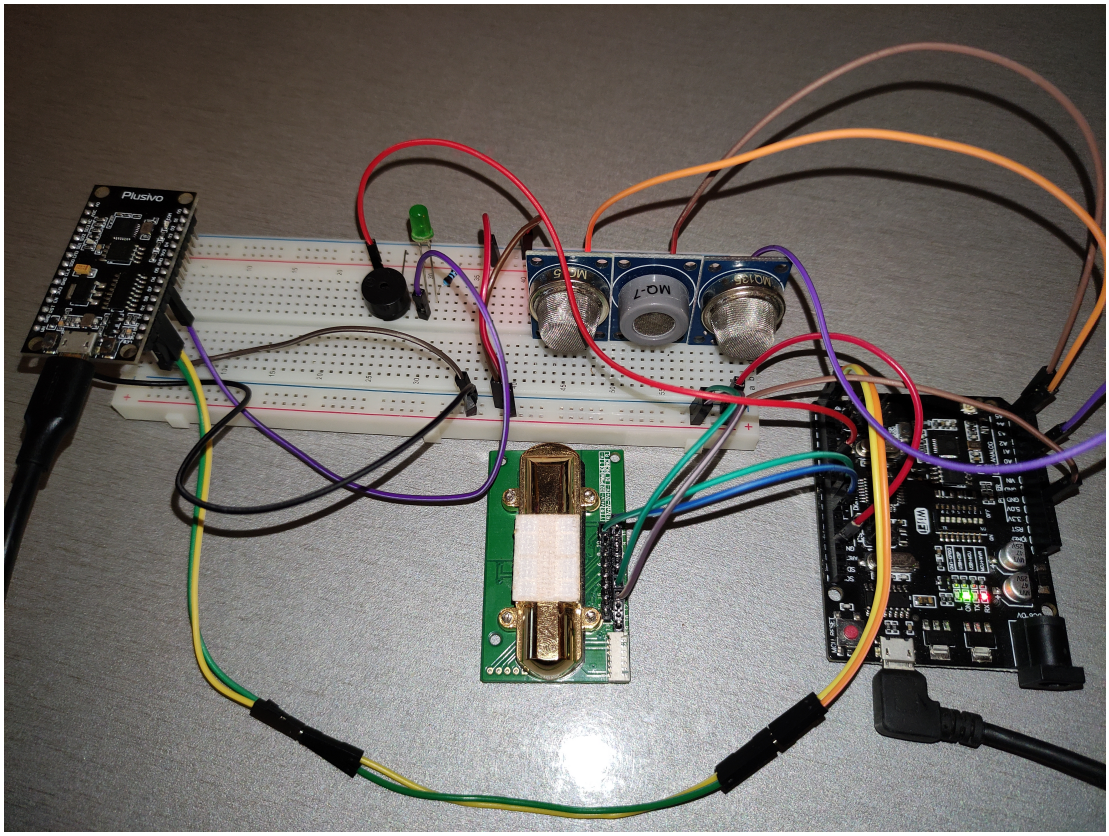
presupune o interfață grafică cu utilizatorul, în care sunt prezentate datele din fișierul CSV / date real time, dacă preluarea datelor de la server este activată. Datele sunt dispuse într-un mod ce permite utilizatorului să observe valorile atât pe un grafic, cât și procentual, pe un progress bar ce se modifică constant în funcție de ultima intrare primită. Procentele sunt afișate în funcție de anumite valori standard (măsurate în funcție de rezistența senzorului sau PPM). În cazul în care se detectează o valoare foarte mare la unul din senzori, acest lucru i se aduce la cunoștință utilizatorului printr-un semnal sonor relevant, ce va rămâne pornit până când senzorul va măsura din nou o valoare normală. Verificarea corectitudinii acestui lucru se face prin apropierea, unei brichete, de exemplu și observarea creșterii rapide a senzorului ce măsoară CO2. în acest caz, se va afișa pe ecran și un mesaj concludent.





Concluzii

În concluzie, lucrarea a prezentat o modalitate de a lua date de la senzori aflați la distanță prin folosirea unei plăcuțe ArduinoUno, care transmite datele la un server local. Astfel, un utilizator poate vedea în timp real care este concentrația de CO2 și CO, precum și detecția de fum din apropierea senzorului. Aceste date ajută persoana aflată la distanță să verifice daca totul este in regula în spatiul in care se afla montați senzorii. Prin folosirea de senzori care măsoară diferite componente din aer și a unor plăcuțe de dezvoltare am reușit să realizăm o implementare eficientă la un cost minim, care va putea fi extinsă pe viitor prin folosirea de alți senzori.



Multumiri si credite lui Radu Nichita, cu care am colaborat in realizarea acestui proiect. El s-a ocupat de partea de prelucrare a datelor, realizand scriptul de Python.

Proiectul in functiune se poate vedea aici
Iar proiectul format pdf se poate descarca de aici

Bibliografie

https://electrogrup.ro/monitorizarea-calitatii-aerului/
https://aqicn.org/sensor/
https://buildmedia.readthedocs.org/media/pdf/kivy/latest/kivy.pdf
https://waqi.info/ro/

Proiectul integral se poate gasi aici:
https://github.com/daianaelena17/PM_Project

[1] Valorile normale ale CO2 pentru o camera bine aerisita se incadreaza intre 400-500 ppm