

# Smart Air Quality Device

**Autor:** [Epure Rareș - Ștefan](#)

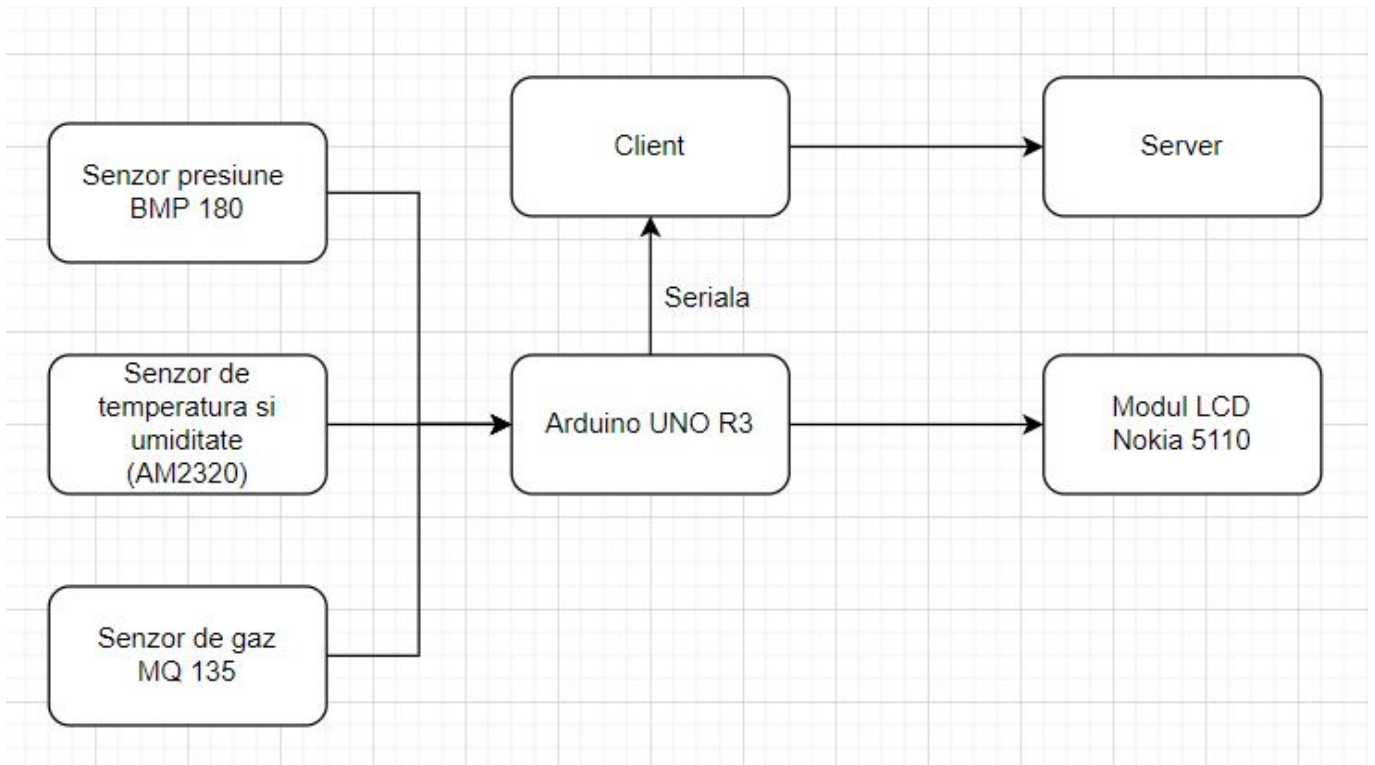
## Introducere

Dispozitiv conceput pentru a măsura calitatea aerului în funcție de anumiți parametri. Acești parametri sunt temperatura, umiditatea, presiunea, precum și măsurarea CO2 din aer, iar ca unitate de masura pentru cantitatea de CO2 se va folosi ppm (parti per milion).

## Descriere generală

Datele vor fi colectate de senzori, iar apoi prin intermediul plăcuței Arduino acestea vor fi prelucrate și trimise prin intermediul SPI-ului către modulul LCD, care le va afișa. Modulul LCD va avea un modul de baza de unde se poate selecta o pagina cu lista de date și o lista cu setari. Setarile tin de afisare (contrast și luminozitate). Pe partea de senzori exista 3 intrari in meniu de unde se poate selecta timpul datelor necesare. Pe langa afisarea datelor pe LCD exista și un client scris in Python care citește datele de pe seriala și le trimite către server prin intermediul unui REST-API. Datele sunt convertite în client și trimise pe server de unde pot fi accesate prin intermediul cererilor GET pe anumite rute (e.g. prin accesare cail /pressure se va afisa o lista cu presiuni). Datele care sunt afisate pe ecran vor fi afisate stocate și pe server.

## Schema bloc

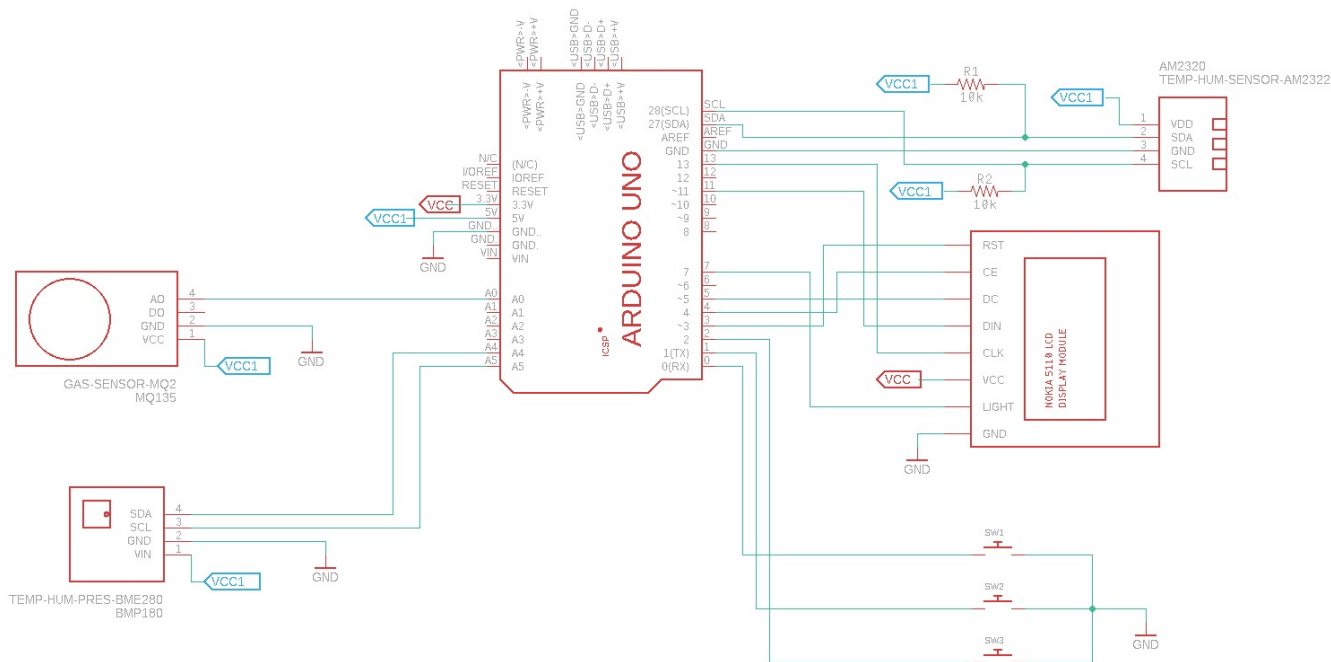


## Hardware design

### Lista Componente

Nr. Crt.	Componente
1.	Arduino UNO R3 (ATmega328p)
2.	Senzor digital de temperatura si umiditate AM2320
3.	Senzor de gaz MQ135
4.	Senzor de presiune BMP 180
5.	Modul LCD Nokia 5110
6.	Fire tata - tata si fire mama - tata
7.	Butoane

### Schema electrica



## Software design

Mediul de dezvoltare pentru implementarea proiectului este Arduino IDE, iar librariile si sursele third-party sunt: [Nokia 5110 Library](#), [Adafruit GFX Library](#), [Adafruit AM2320 Library](#), [MQ 135 Library](#), [BMP 180 Library](#).

Din punct de vedere al implementarii software, proiectul se imparte in 3 module:

- un modul implementat in Arduino IDE, ce reprezinta logica de colectare a datelor, de afisare a acestora pe modulul LCD si de utilizare a butoanelor
- doua module implementate in Python ce corespund unui client si unui server.

Logica definita in Arduino IDE este urmatoarea:

- datele sunt colectate de la toti senzorii prin interfetele specifice fiecaruia
- aceste date sunt afisate pe LCD
- Modulul LCD are o interfata grafica, ce se bazeaza pe un meniu. Exista un meniu pentru setari si un meniu din care se aleg anumite date. Pentru selectarea datelor si pentru navigarea prin meniu se folosesc 3 butoane (un buton pentru a parcurge lista de optiuni a unui meniu, altul prin care se selecteaza o intrare in meniu, iar cel de-al treilea are functionalitate de back).
- Totodata datele transmise LCD-ului sunt afisate si pe interfata seriala.

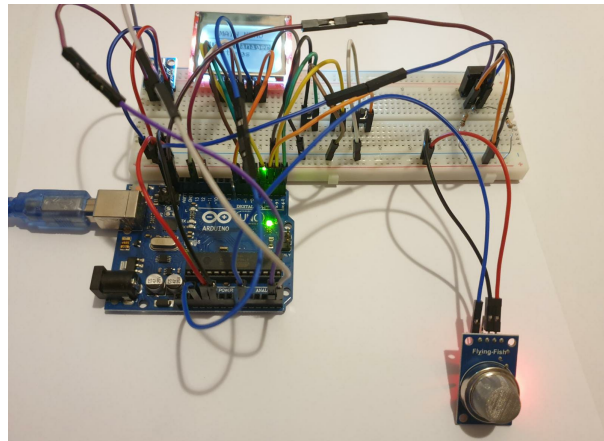
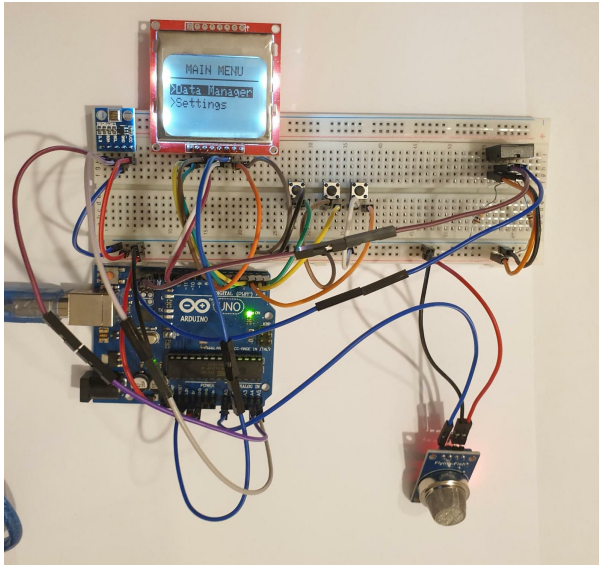
Logica client-server:

- Clientul preia datele de pe interfata seriala, prelucreaza datele si le transmite catre server printr-o cerere de tip POST.
- Serverul este implementat in Python pe framework-ul Flask. Primeste datele prin cererea de tip POST si le stocheaza in liste. Pe langa aceasta functionalitate, aceste date pot fi achizitionate prin cereri de tip GET pe anumite cai ("/pressure", "/temperature", "/gas", "/humidity").

Fluxul de dezvoltare a constat intai in testarea componentelor hardware, apoi s-a scris codul, pentru

achizitionarea datelor, aferent fiecarui senzor. Dupa ce toate componentele au fost testate si implementate separat, s-a trecut la grupare acestor functionalitati si testarea intregului sistem. Implementarea server-client a fost realizata ulterior, dupa ce functionalitatea de achizitionare de date a fost realizata cu succes.

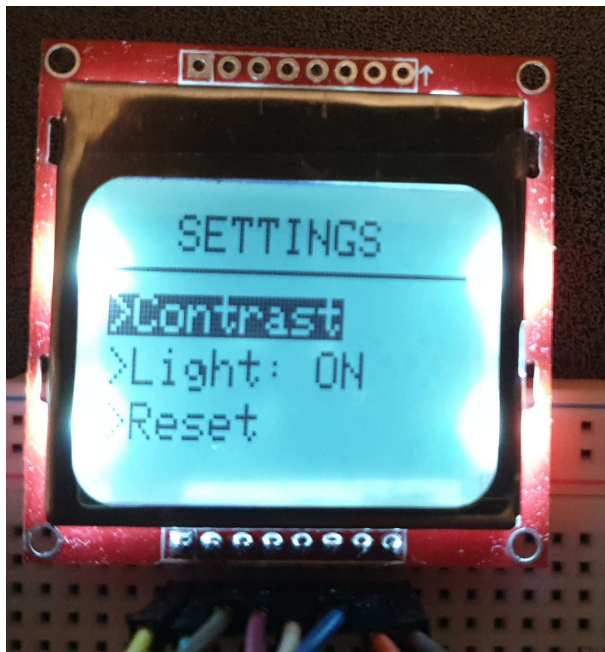
## Rezultate obținute



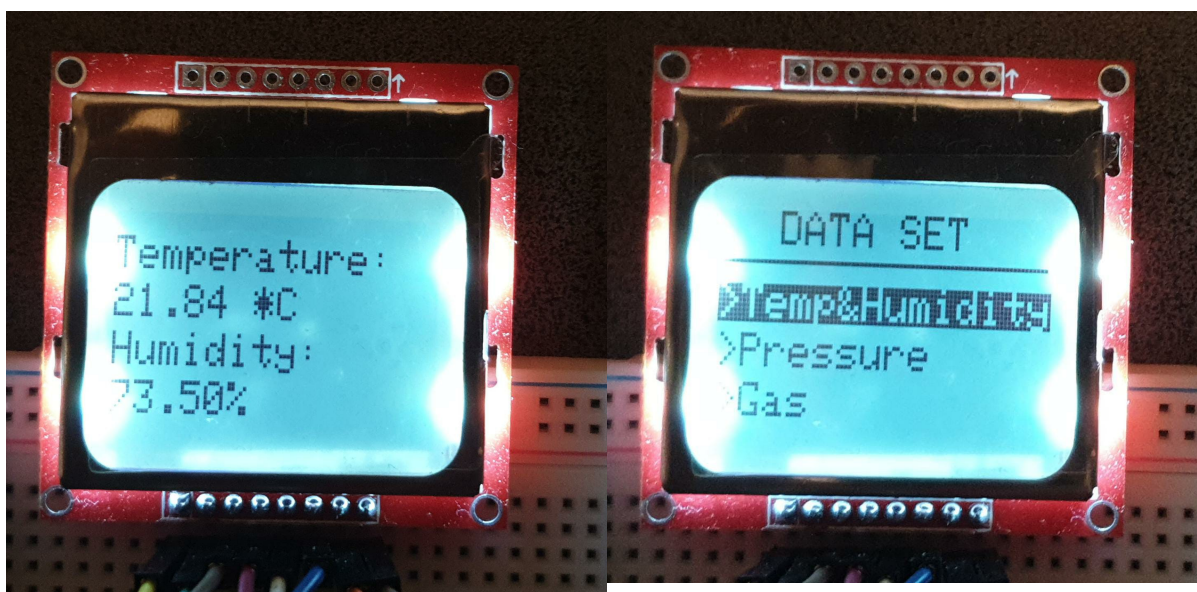
## Meniu principal



## Meniu setari



## Meniu date



## Concluzii

Proiectul mi-a oferti oportunitatea de a intelege mai bine partea de low level si de a programa un microcontroller. Pe langa partea de software, a fost interesanta conceperea partii hardware, prin creerea unui circuit simplu si prin legarea mai multor componente intre ele. Partea din cadrul proiectului care a necesitat mai multa munca a fost afisarea meniu-ului pe modului LCD, deoarece am incercat sa am mai multe pagini ale meniu-ului (o pagina de start, o pagina de setari si una de afisare a datelor). O alta parte interesanta a fost aceea ca experimentat faptul ca senzorii au anumite intarzieri, ceea ce poate produce anumite erori. Am intalnit o astfel de eroare la senzorul de gaz,

deoarece clientul nu putea citi date de pe interfata seriala. Dupa introducerea unei mici intarzieri prin functia delay, am putut colectat datele de la acest senzor cu ajutorul client-ului implementat in Python.

## Download

Arhiva contine urmatoarea structura de fisiere:

```
.  
|- resurse  
  |- meniu  
    |- meniu.ino (fisierul sursa care realizeaza achizitionarea si afisarea  
datelor pe ecran)  
  |- server-client  
    |- server.py  
    |- client.py
```

[resurse.zip](#)

[Demo](#)

## Jurnal

1. Alegere tema.
2. Stabilirea componentelor necesare.
3. Testarea individuala a componentelor.
4. Montarea tuturor componentelor pe breadboard.
5. Implementarea componentei software.
6. Testare.

## Bibliografie

[Documentatie modul LCD Nokia 5110](#)

[Documentatie modul BMP 180](#)

[AM2320 Sensor Datasheet](#)

[MQ135 Datasheet](#)

[Pagina in format PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/smart\\_air\\_quality\\_device](http://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/smart_air_quality_device)



Last update: **2021/06/04 12:28**