

Air Quality Monitor

Introducere

Ce face proiectul?

Proiectul este un monitor de calitate a aerului, bazat pe Arduino, care masoara temperature dintr-o incapere cu ajutorul unui sensor de temperature si de asemenea poate detecta nivelul de gaze. Valorile sunt afisate pe un display: temperature curenta este mereu vizibila, iar nivelul gazelor apare atunci cand se detecteaza un anumit prag. De asemenea, un LED RGB ofera un semnal visual: isi schimba culoarea in functie de temperature camerei (albastru pentru temperaturi mai scazute si rosu pentru temperaturi mai ridicate).

Care este scopul lui?

Scopul proiectului este de a monitoriza atat temperature cat si calitatea aerului intr-un mod simplu si accesibil, oferind utilizatorului informatii esentiale despre mediul in care se afla. Este util mai ales in spatii precum bucatariile, unde pot aparea scurgeri de gaze sau acumulari periculoase, dar si in orice alta camera in care vrei sa stii cat de curat este aerul si cat de confortabila este temperature.

Care a fost ideea de la care am pornit?

Ideea a pornit de la nevoia de a avea un dispozitiv compact si usor de folosit, care sa monitorizeze permanent doua aspect importante din orice locuinta: temperature si nivelul de gaze. M-am gandit ca un astfel de dispozitiv poate preveni probleme de sanatate sau accidente cauzate de acumularea gazelor si, in acelasi timp, poate oferi un control mai bun asupra confortului termic.

De ce credem ca este util?

Cred ca proiectul este util pentru ca ofera o solutie simpla si vizuala pentru monitorizarea calitatii aerului si a temperaturii dintr-o camera, ceea ce poate contribui la siguranta si confortul locuintelor. Poate atrage interesul celor care vor sa-si imbunatateasca siguranta casei fara a investi in sisteme costisitoare.

Descriere generală

Sistemul este compus din urmatoarele module:

- Senzor de temperature si umiditate (DHT11) - masoara temperature ambientala si trimite date digitale catre microcontroller.
- Senzor de calitate a aerului (MQ-8) - detecteaza nivelul gazelor din aer si emite un semnal analogic proportional cu concentratia acestora
- LED RGB - ofera feedback visual: isi schimba culoarea de la albastru (pentru temperaturi scazute) la rosu aprins (pentru temperaturi ridicate)
- Display LCD 1602 cu I2C - afiseaza temperature curenta si nivelul gazelor masurate
- Microcontroler (Arduino Uno) - colecteaza datele de la senzori, proceseaza informatiile si controleaza LED-ul RGB si afisajul LCD

Interactiunea dintre module:

- senzorii transmit date catre microcontroller.
- microcontrollerul proceseaza valorile primite si evalueaza temperature si nivelul gazelor.
- pe baza acestor date, microcontrollerul actualizeaza informatiile afisate pe LCD si modifica culoarea LED-ului RGB.

Diagrama bloc:



Hardware Design

BILL OF MATERIALS

Nr.	Componentă	Model / Tip	Protocol	Link de achiziție
1	Microcontroller	Arduino Uno (inclus în kit Plusivo)	-	[Plusivo Kit - Optimus](https://shorturl.at/AEv17)
2	Senzor de gaze	MQ-8	ADC (analogic)	[MQ-8 Sensor - Optimus](https://shorturl.at/0iMty)
3	Senzor temperatură și umiditate	DHT11 (modul)	Digital(1-Wire)	[DHT11 Sensor - Optimus](https://shorturl.at/iSJwZ)
4	Display OLED	0.96" I2C	I2C	[OLED Display - eMAG](https://shorturl.at/XEDay)
5	LED RGB	Catod Comun	PWM	Inclus în kit (Plusivo)
6	Breadboard + jumper wires	-	-	Inclus în kit (Plusivo)

Schema Electrica



Descrierea funcționalității hardware

- **1. Microcontroller - Arduino Uno**
 - Citirea valorilor de la senzori (gaz și temperatură)
 - Procesarea acestora
 - Afișarea valorilor pe un ecran OLED
 - Activarea LED-ului RGB
- **2. Senzor de gaze - MQ-8**
 - Conectat la pinul analogic A0
 - Funcționează pe principiul unui rezistor variabil
 - Semnalul este preluat printr-un divizor de tensiune cu o rezistență de 10kΩ la GND
 - Protocol: ADC (analog)
- **3. Senzor de temperatură - DHT11**
 - Conectat la pinul digital D2
 - Alimentat la 5V, cu GND comun
 - Protocol: Digital (1-Wire)
- **4. Display OLED 0.96" I2C**
 - Conectat la pinii A4 (SDA) și A5 (SCL)
 - Afișează în timp real valorile de la senzori
 - Protocol: I2C

Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuieți să le implementați
- (etapa 3) surse și funcții implementate

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2025/vstoica/dennis_stefan.dobra

Last update: **2025/05/19 10:42**

