

# EnviroMind

## Introducere

**EnviroMind** este un sistem standalone de monitorizare a calității mediului ambiental, capabil să măsoare temperatura, umiditatea, presiunea atmosferică și calitatea aerului în timp real. Sistemul integrează un algoritm de **detectie a anomaliilor** bazat pe distanța Mahalanobis, care identifică abateri semnificative față de condițiile normale ale mediului și generează alerte vizuale și sonore. Controlul interfeței se realizează prin gesturi. Datele sunt accesibile de la distanță printr-un dashboard web, disponibil cu ajutorul unui modul ESP32.

**Ideea** a pornit de la o problemă din viața reală: **detectarea degradării calității aerului în spații închise**, cauzată de fumul de țigară electronică (vaping) sau alți poluanți. Într-un spațiu închis aglomerat, calitatea aerului poate scădea rapid fără ca oamenii să își dea seama. Un sistem autonom care monitorizează continuu și alertează când condițiile devin anormale poate preveni expunerea prelungită la aer viciat.

## Descriere generală

Sistemul este construit în jurul microcontrollerului **ATmega328P Xplained Mini**, care coordonează toți senzorii și perifericele. Un modul **ESP32** conectat prin UART găzduiește un server web accesibil prin WiFi, oferind un dashboard în timp real.



## Module

- **ATmega328P Xplained Mini** - microcontroller principal, coordonează toți senzorii și perifericele
- **ESP32 DevKit** - găzduiește un server web accesibil prin WiFi, primește date de la ATmega prin UART
- **AHT20 + BMP280** - măsoară temperatura, umiditatea și presiunea atmosferică prin I2C
- **APDS-9960** - detectează gesturi (UP/DOWN/LEFT/RIGHT) prin I2C, semnalizează prin întrerupere externă
- **MiCS-5524** - senzor analogic pentru calitatea aerului (CO, VOC), citit prin ADC
- **LCD 16x2 cu modul I2C** - afișează datele senzorilor și alertele
- **Modul MicroSD** - stochează profilul de calibrare și logurile CSV prin SPI
- **LED-uri (x3)** - indicatori vizuali de status (verde/galben/roșu)
- **Buzzer pasiv** - alertă sonoră prin semnal PWM

## Interacțiuni între module

- **I2C (A4/A5):** ATmega comunică cu LCD, APDS-9960, AHT20 și BMP280 pe același bus. Fiecare dispozitiv are adresă unică.
- **ADC (A0):** ATmega citește tensiunea analogică de la MiCS-5524 și o convertește digital.
- **SPI (D10-D13):** ATmega accesează modulul MicroSD pentru salvarea profilului de calibrare și logarea periodică a datelor în format CSV.
- **UART (D0/D1):** ATmega trimite datele senzorilor către ESP32. Deoarece ATmega operează la 5V logic și ESP32 la 3.3V, linia TX a ATmega trece printr-un voltage divider rezistiv înainte de a ajunge la RX-ul ESP32.
- **GPIO:** ATmega controlează cele 3 LED-uri de status în funcție de nivelul de anomalie detectat.
- **PWM (D3):** ATmega generează semnal PWM pentru buzzer la detectarea valorilor critice.
- **Întrerupere externă (D2):** Pinul INT al APDS-9960 este conectat la INT0 al ATmega pentru detecția gesturilor fără polling.

## Hardware Design

### Listă de componente

Componentă	Protocol	Tensiune
ATmega328P Xplained Mini	-	5V
ESP32 DevKit	UART	3.3V
Senzor AHT20 + BMP280 (modul combinat)	I2C	3.3V
Senzor gesturi APDS-9960	I2C	3.3V
Senzor calitate aer MiCS-5524	ADC	5V
Display LCD 16x2 cu modul I2C	I2C	5V
Modul MicroSD SPI	SPI	3.3V/5V
Buzzer pasiv	PWM	5V
LM386 (amplificator audio)	-	5V
LED	GPIO	5V

### Schema electrică



### Diagrame de semnal

#### I2C



Protocolul I2C foloseste doua linii: SDA (date) si SCL (clock). Datele sunt valide cand SCL este HIGH. Conditia START apare cand SDA coboara in timp ce SCL este HIGH, iar conditia STOP apare cand SDA urca in timp ce SCL este HIGH.

## SPI



Protocolul SPI foloseste patru linii: CS (chip select), SCK (clock), MOSI (master out) si MISO (master in). Comunicatia incepe cand CS coboara la LOW.

## UART

### [Diagrama de semnal UART](#)

Protocolul UART transmite date serial: un bit de start (LOW), 8 biti de date, optional un bit de paritate, si unul sau mai multi biti de stop (HIGH). Rata de transmisie este 57600 baud.

# Software Design

## 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/prj2026/bianca.popa1106/adela\\_andreea.marin](http://ocw.cs.pub.ro/courses/pm/prj2026/bianca.popa1106/adela_andreea.marin)



Last update: **2026/05/11 10:40**