

Sistem de Monitorizare a Mediului

Autor: Susai Florian

Grupa: 332

Seria: CA

Facultatea: Automatică și Calculatoare, Universitatea Politehnica din București

Introducere

Proiectul reprezintă un **dispozitiv hardware inteligent de achiziție a datelor** (*edge device*) destinat monitorizării continue a parametrilor de mediu dintr-un spațiu (interior sau exterior).

Ce face sistemul:

- Citește periodic parametrii ambientali: **luminozitate** (LDR), **temperatură și presiune atmosferică** (BMP280), respectiv **concentrația de gaze/fum** (MQ-2).
- Elimină zgomotul de citire prin filtrare digitală (medie mobilă pe buffer circular).
- Procesează datele matematic local, pe microcontroller, pentru a extrage informații de nivel înalt: **indice de disconfort termic și tendință barometrică** (predicție de furtună).
- Afișează un rezumat al stării pe un ecran **LCD 16x2** prin magistrala I2C.
- Transmite pachete serializate de date către un **smartphone** prin **Bluetooth** (modul HC-05) pentru monitorizare la distanță.

Scopul proiectului: Realizarea unei stații de mediu compacte, autonome și extensibile, care să demonstreze conceptul de *edge computing* pe un microcontroller cu resurse limitate (ATmega328P) — datele nu sunt trimise brute către cloud, ci procesate matematic local și transformate în informație utilă (alerte, stări calitative).

Ideea de pornire: Multe stații meteo de uz casnic afișează doar valori brute (temperatură, presiune), fără a oferi interpretări. Am pornit de la întrebarea: *“Cum aș putea face un sistem care, pe lângă numere, să spună utilizatorului dacă este Confort, Atenție sau Pericol, și care să prezică o furtună înainte ca aceasta să apară?”*. Răspunsul a fost combinarea senzorilor cu algoritmi clasici de procesare (Heat Index Rothfus, derivata presiunii în timp).

De ce este util:

- **Pentru utilizator:** primește informații deja interpretate (stare textuală, alertă), nu trebuie să cunoască semnificația valorilor brute.
- **Pentru dezvoltator (învățare):** integrează **trei periferice diferite** (USART, I2C, ADC) și **trei laboratoare** (Lab 1 – USART, Lab 2 – Timere/Întrepereri, Lab 4 – ADC, Lab 6 – I2C), oferind o imagine completă asupra arhitecturii unui sistem embedded real.
- **Extensibilitate:** structura modulară permite adăugarea ușoară a altor senzori (umiditate DHT22, calitate aer, PM2.5).

Descriere generală

Arhitectura sistemului este **modulară și ierarhică**, organizată pe trei niveluri funcționale:

1. **Nivelul de intrare (senzori):** patru module care convertesc mărimi fizice în semnale electrice citibile de microcontroller — fotorezistor (analogic, prin divizor de tensiune), MQ-2 (analogic, ieșire AO), BMP280 (digital, I2C) și HC-05 (digital, UART, dual-rol: intrare comenzi / ieșire date).
2. **Nivelul de procesare (unitate centrală):** placa Arduino Uno cu microcontrollerul ATmega328P, care orchestrează achiziția prin întreruperi de timer, aplică filtrarea digitală, calculează indici derivați și gestionează protocoalele de comunicație.
3. **Nivelul de ieșire (interfețe):** afișaj local (LCD 16×2 cu adaptor I2C) pentru utilizatorul fizic prezent și interfață externă (aplicație Bluetooth pe smartphone/PC) pentru monitorizare la distanță.

Schema bloc



Legendă:

- **Săgeți roșii** — alimentare 5V/GND (magistrală de putere)
- **Săgeți portocalii** — semnale de la senzori (analogice sau I2C)
- **Săgeți violete** — comunicație serială (UART/Bluetooth)
- **Săgeți verzi** — ieșire către afișaj (I2C)

Descrierea modulelor și a interacțiunii

Modul	Rol	Interfață cu Arduino	Tip semnal
Fotorezistor (LDR)	Măsoară luminozitatea ambientală	A1 (intrare analogică)	Tensiune divizor 0-5V
MQ-2	Detectează gaze inflamabile / fum	A0 (intrare analogică)	Tensiune analogică (AO)
BMP280	Măsoară presiune și temperatură	SDA→A4, SCL→A5 (I2C)	Date digitale I2C
HC-05 Bluetooth	Comunicație bidirecțională cu telefonul	TXD→D0(RX), RXD→D1(TX)	UART asincron
LCD 16×2 + adaptor I2C	Afișare locală a datelor procesate	SDA→A4, SCL→A5 (I2C)	Date digitale I2C
Arduino Uno (ATmega328P)	Unitate centrală de procesare	—	Coordonator
Alimentare USB 5V	Magistrală de putere	VIN/5V + GND	DC 5V

Fluxul de date și control

Fluxul logic al sistemului este ciclic și pornește de la întreruperea generată de Timer1 (configurat în

mod CTC la 1 Hz). La fiecare tact, ISR-ul setează un flag care semnalează loop-ului principal că este momentul pentru o nouă achiziție. Loop-ul citește pe rând cei doi senzori analogici (LDR pe A1 și MQ-2 pe A0), aplicând filtrul de medie mobilă pe câte un buffer circular de 16 eșantioane, apoi interoghează BMP280 pe magistrala I2C pentru a obține temperatura și presiunea. Datele brute sunt apoi procesate matematic: temperatura este transformată în indice de disconfort (stare textuală — Confort / Atenție / Pericol), iar presiunea este comparată cu istoricul ultimelor 30 de minute pentru a calcula tendința barometrică. În final, rezultatele sunt afișate pe LCD prin aceeași magistrală I2C și transmise serializat prin UART către modulul HC-05, care le retransmite via Bluetooth către aplicația de pe telefon.

Notă privind partajarea I2C: atât LCD-ul cât și BMP280 folosesc aceeași magistrală (pinii A4/A5), dar au **adrese diferite** (de obicei 0x27 pentru LCD și 0x76/0x77 pentru BMP280), deci nu există conflict. Magistrala este gestionată secvențial de biblioteca `Wire.h`.

Notă privind UART și HC-05: modulul Bluetooth este conectat pe pinii hardware D0 (RX) și D1 (TX). Acest lucru poate intra în conflict cu portul serial USB folosit pentru programare/debug. În varianta finală se folosește direct USART hardware; pentru debug în timpul dezvoltării se recomandă **SoftwareSerial** pe alți pini (de ex. D2/D3) sau deconectarea HC-05 pe perioada uploadului de cod.

Hardware Design

Listă de piese

- **Placă Arduino Uno R3** — microcontroller ATmega328P, 16 MHz, 32KB Flash, 2KB SRAM
- **Modul Bluetooth HC-05** (sau HC-06 ca alternativă) — Bluetooth 2.0 SPP, alimentare 3.3-6V
- **Display LCD 16x2 cu modul adaptor I2C** (PCF8574) — adresă I2C tipică 0x27
- **Senzor digital BMP280** — presiune (300-1100 hPa) și temperatură (-40...+85°C), I2C/SPI
- **Senzor MQ-2** — gaze inflamabile (LPG, propan, H₂, fum), ieșire analogică AO
- **Fotorezistor (LDR)** — GL5528 sau echivalent
- **Rezistor 10 kΩ** — pentru divizor de tensiune cu LDR (pull-down)
- **Breadboard 830 puncte**
- **Fire de conexiune** (jumper wires male-male, male-female)
- **Cablu USB tip B** pentru alimentare și programare

Conexiuni pin-cu-pin

Fotorezistor (LDR) — divizor de tensiune

LDR-ul formează împreună cu un rezistor de 10 kΩ un divizor de tensiune clasic: un capăt al LDR se leagă la 5V, celălalt capăt al LDR se conectează în nodul comun cu rezistorul de 10 kΩ (acesta merge la GND), iar tensiunea din nodul comun este preluată pe pinul A1 al Arduino. Când lumina ambientală este puternică, rezistența LDR scade și tensiunea pe A1 crește; pe întuneric, fenomenul este invers.

Capăt	Conectare
-------	-----------

Capăt 1 LDR	5V
Capăt 2 LDR (jonctiune cu R 10kΩ)	A1
Celălalt capăt al R 10kΩ	GND

MQ-2 (senzor gaz/fum)

Pin MQ-2	Arduino Uno
VCC	5V
GND	GND
AO (analog out)	A0
DO (digital out)	- (neutilizat)

BMP280 (I2C)

Pin BMP280	Arduino Uno
VCC	3.3V (sau 5V dacă modulul are regulator on-board)
GND	GND
SDA	A4
SCL	A5
CSB	VCC (forțează modul I2C)
SDO	GND (selectează adresa 0x76)

LCD 16×2 cu adaptor I2C

Pin adaptor I2C LCD	Arduino Uno
VCC	5V
GND	GND
SDA	A4 (partajat cu BMP280)
SCL	A5 (partajat cu BMP280)

HC-05 Bluetooth (UART)

Atenție la încrucișarea TX/RX: TX-ul unui dispozitiv se conectează la RX-ul celuilalt și invers!

Pin HC-05	Arduino Uno
VCC	5V
GND	GND
TXD	D0 (RX)
RXD	D1 (TX) — prin divizor 1kΩ/2kΩ (RXD HC-05 este la 3.3V logic)
EN	- (neutilizat în mod normal)
STATE	- (neutilizat)

Diagrame de semnal

Semnal LDR (ADC pe A1): Tensiunea variază între $\sim 0.1V$ (întuneric, LDR $\sim 1M\Omega$) și $\sim 4.5V$ (lumină puternică, LDR $\sim 1k\Omega$). ADC-ul pe 10 biți returnează valori între ~ 20 și ~ 920 .

Semnal MQ-2 (ADC pe A0): După perioada de *preheating* (20-60 secunde), AO returnează o tensiune proporțională cu concentrația de gaz. Valoare de bază (aer curat): $\sim 0.5-1V$ ($\sim 100-200$ ADC). Prag tipic de alarmă: $\sim 2.5V$ (~ 500 ADC).

Semnal I2C (SDA/SCL): Frecvență standard 100 kHz, formă de undă caracteristică start condition - address byte - ack - data - stop condition. Tranzacțiile către BMP280 (citire registre 0xF7-0xFC) și către LCD (comenzi 4-bit prin PCF8574) sunt secvențiale.

Semnal UART (HC-05): Baud rate implicit 9600 bps, 8N1. Pachet tipic de aproximativ 28 octeți (T:24.5|P:1012|Tr:Stable|L:85%), durată de transmisie ~ 30 ms.

Software Design

Mediu de dezvoltare

- **PlatformIO** cu VS Code — compilare și upload pentru ATmega328P
- **AVR-GCC toolchain**
- **Serial Bluetooth Terminal** (Android) — pentru testarea recepției datelor pe telefon
- **Git + GitHub** — versionare cod sursă

Librării și surse 3rd-party

Librărie	Rol	Sursă
Wire.h	Comunicație I2C (master)	inclusă în Arduino core
LiquidCrystal_I2C.h	Driver pentru LCD 16x2 cu adaptor PCF8574	Frank de Brabander (GitHub)
Adafruit_BMP280.h	Driver BMP280	Adafruit Industries
Adafruit_Sensor.h	Interfață unificată senzori (dependență BMP280)	Adafruit Industries
avr/io.h, avr/interrupt.h	Acces direct la registre AVR și ISR	AVR-libc (toolchain)
SoftwareSerial.h	UART pe pini software (dezvoltare/debug)	inclusă în Arduino core

Algoritmi și structuri implementate

1. Filtru digital — Medie mobilă pe buffer circular

Datele citite de la LDR și MQ-2 prin ADC prezintă fluctuații datorate zgomotului electric și variațiilor naturale. Pentru atenuare se folosește un buffer circular de **N = 16** eșantioane menținut în memoria RAM a microcontrollerului. La fiecare nouă conversie ADC, valoarea cea mai veche din buffer este suprascrisă cu cea nouă, iar media aritmetică a celor 16 eșantioane este recalculată. Doar această medie este folosită mai departe în procesarea aplicației, valorile brute fiind ignorate. Costul de memorie este minim (32 octeți pentru un buffer de 16 valori uint16_t), iar latența de răspuns la o schimbare reală a mediului este de ordinul a 16 secunde (la o frecvență de eșantionare de 1 Hz), ceea ce este perfect acceptabil pentru parametri de mediu cu variație lentă.

2. Indice de disconfort termic (Heat Index - Rothfus simplificat)

Formula completă a lui Rothfus folosește atât temperatură cât și umiditate. În absența senzorului de umiditate (BMP280 nu măsoară RH), se folosește o **versiune redusă** bazată doar pe temperatură, cu mapare pe stări calitative:

- $T < 27^{\circ}\text{C} \rightarrow$ "Confort"
- $27^{\circ}\text{C} \leq T < 32^{\circ}\text{C} \rightarrow$ "Atenție"
- $T \geq 32^{\circ}\text{C} \rightarrow$ "Pericol"

Rezultatul este transmis ca etichetă text, nu ca valoare numerică, către aplicația Bluetooth — utilizatorul primește direct o stare interpretabilă, nu trebuie să știe ce înseamnă "31.4°C" în termeni de confort.

3. Analiza tendinței barometrice (derivata presiunii în timp)

Presiunea atmosferică citită de BMP280 este memorată într-un buffer circular cu **rezoluție de 1 minut** (30 de eșantioane pentru a acoperi o fereastră de 30 de minute). La fiecare ciclu, sistemul calculează diferența ΔP între valoarea curentă și cea de acum 30 de minute. Interpretarea este următoarea:

- $\Delta P < -2 \text{ hPa} / 30 \text{ min} \rightarrow$ flag intern "**Furtună**" + alertă imediată pe LCD + push Bluetooth
- $-2 \leq \Delta P < -0.5 \rightarrow$ "Înrăutățire"
- $|\Delta P| \leq 0.5 \rightarrow$ "Stabil"
- $\Delta P > 0.5 \rightarrow$ "Îmbunătățire"

Această tehnică reproduce principiul barometrelor mecanice clasice, în care o scădere bruscă a presiunii este indicator clar de schimbare meteorologică în următoarele câteva ore.

4. Orchestrare temporală — Timer1 CTC

Pentru a evita funcția `delay()` (blocantă, care ar bloca complet microcontrollerul în timpul așteptării), se folosește **Timer1** hardware configurat în modul **CTC** (Clear Timer on Compare Match), cu prescaler 1024, pentru a genera o întrerupere la fiecare **1 secundă** exactă. Calculul valorii registrului OCR1A este: $16 \text{ MHz} / 1024 / 1 \text{ Hz} - 1 = 15624$. ISR-ul aferent (`TIMER1_COMPA_vect`) este

intenționat **foarte scurt** — doar setează un flag global volatil; procesarea propriu-zisă (citirea ADC, comunicația I2C, transmisia UART) se face în loop-ul principal pentru a păstra timpul petrecut în ISR sub câteva microsecunde. Această abordare garantează o eșantionare precisă în timp, esențială pentru calculul corect al derivatei presiunii și al mediei mobile.

5. Structura pachetelor Bluetooth

Datele transmise nu sunt trimise haotic, ci **serializate într-un format text delimitat**, ușor de parsat atât de aplicații generice (Serial Bluetooth Terminal pe Android) cât și de o aplicație custom. Formatul este: `T:24.5|P:1012.3|Tr:Stable|L:85%|G:120`, unde:

- T — temperatură (°C)
- P — presiune (hPa)
- Tr — tendință barometrică (Stable / Rain / Improving)
- L — luminozitate normalizată (%)
- G — valoare gaz MQ-2 (ADC raw)

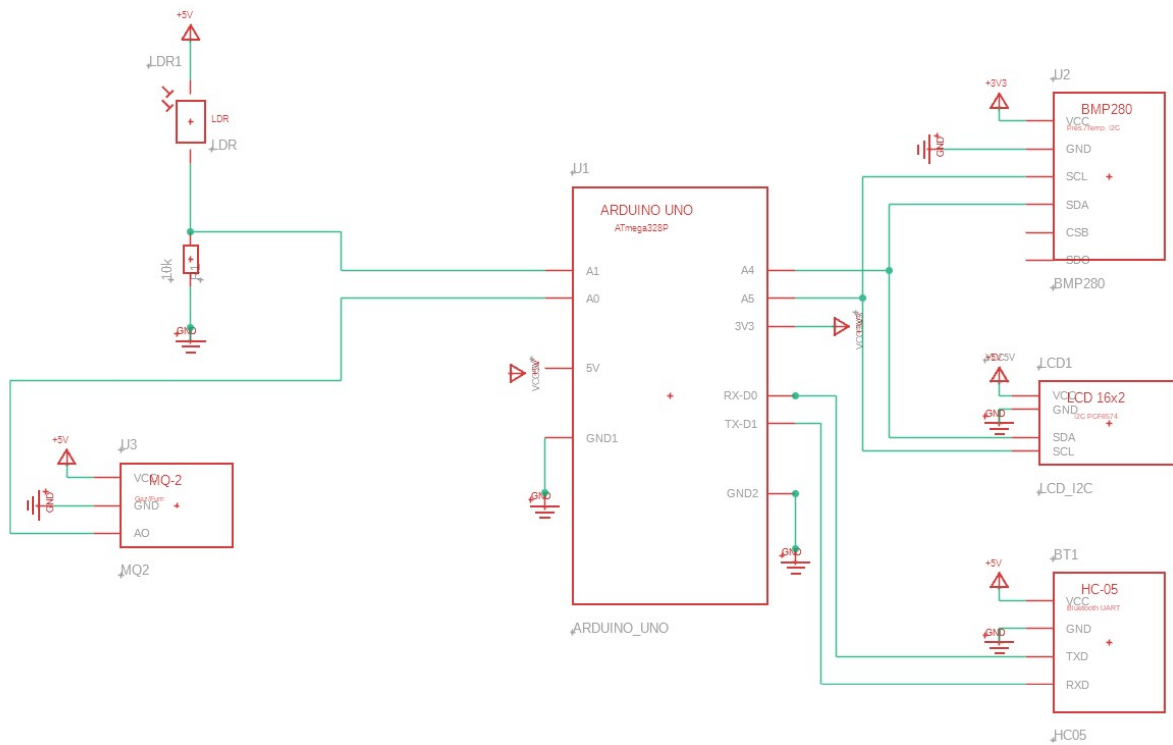
Fiecare pachet se termină cu caracterul newline (\n) pentru a permite parsarea linie-cu-linie pe partea de telefon.

Structura modulară a codului sursă

Codul este împărțit în module separate, fiecare cu responsabilitate unică, pentru a facilita testarea și mentenanța:

- `main.cpp` — bucla principală, gestionare flag-uri, dispatching
- `senzori_adc.cpp/.h` — citire LDR și MQ-2, filtru medie mobilă
- `comunicatie_usart.c/.h` — init UART, RX/TX, parser comenzi
- `display_i2c.cpp/.h` — wrapper LiquidCrystal_I2C, layout ecran
- `senzor_bmp280.cpp/.h` — wrapper Adafruit_BMP280
- `procesare_matematica.cpp/.h` — Heat Index, derivată presiune
- `timer_orchestrator.c/.h` — init Timer1 CTC, ISR

Schema electrică detaliată



Schema electrică prezintă toate conexiunile pin-cu-pin între Arduino Uno (ATmega328P) și cele patru module (LDR, MQ-2, BMP280, LCD I²C, HC-05), incluzând cele două divizoare de tensiune (LDR + 10kΩ pentru semnalul analogic pe A1, respectiv 1kΩ/2kΩ pentru protecția pinului RXD al HC-05 la nivel logic 3.3V).

Bibliografie și datasheet-uri

Datasheet-uri componente

- **Arduino Uno R3** (placa principală):
 - [Pagină produs oficială Arduino](#)
 - [Datasheet PDF \(A000066\)](#)
- **ATmega328P** (microcontroller, Microchip/Atmel):
 - [Pagină produs Microchip \(datasheet + errata\)](#)
 - [Datasheet complet PDF \(DS40002061\)](#)
- **BMP280** (senzor presiune/temperatură, Bosch Sensortec):
 - [Datasheet oficial Bosch \(BST-BMP280-DS001\)](#)
- **MQ-2** (senzor gaz/fum, Hanwei Electronics):
 - [Datasheet Hanwei PDF](#)
- **HC-05** (modul Bluetooth SPP, ITead Studio):
 - [Datasheet PDF \(ITead\)](#)
 - [Pinout și detalii Components101](#)
- **PCF8574** (I/O expander pentru adaptorul I²C al LCD-ului, NXP/TI):
 - [Datasheet NXP oficial](#)
 - [Datasheet Texas Instruments \(variantă\)](#)
- **HD44780U** (controller LCD-ul de 16x2, Hitachi):

- [Datasheet PDF \(mirror SparkFun\)](#)
- **GL5528 / LDR** (fotorezistor):
 - [Datasheet GL5528 \(Senba\)](#) — caracteristică rezistență-iluminare
- **Rezistori** (10kΩ, 1kΩ, 2kΩ — generici, 1/4W, ±5%):
 - Specificații standard E12/E24 — nu necesită datasheet specific

Resurse software și librării

- **Wire.h** (I²C master, Arduino core): [Documentație oficială](#)
- **LiquidCrystal_I2C** (Frank de Brabander): [Repository GitHub](#)
- **Adafruit_BMP280**: [Repository GitHub](#) · [Tutorial Adafruit](#)
- **AVR-libc** (avr/io.h, avr/interrupt.h): [Manual de referință](#)
- **PlatformIO**: [Documentație platformă atmelavr](#)

Referințe teoretice

- **Heat Index — Rothfusz Regression Equation** (NOAA / NWS): [US National Weather Service](#)
- **I²C Bus Specification** (NXP/Philips): [UM10204 - I2C-bus specification and user manual](#)
- **Tendință barometrică și predicție meteorologică** (Zambretti / King algorithm): articol Wikipedia [Zambretti Forecaster](#)

[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/florian.susai>



Last update: **2026/05/12 21:36**