

Monitor ambiental pentru camere de hotel

Introducere

□ Ce face proiectul:

Dispozitivul monitorizează condițiile ambientale dintr-o cameră de hotel, măsurând:

- nivelul de zgomot
- temperatura și umiditatea aerului

Atunci când sunetul depășește un prag prestabilit:

- se aprinde un LED de avertizare
- se salvează datele (zgomot + climă) pe un card SD
- dacă se detectează 3 astfel de evenimente în mai puțin de 1 minut, se consideră o plângere și se:
 - salvează într-o bază de date separată
 - trimite automat un email către recepție (sau administrator)

Toate măsurătorile sunt transmise periodic prin **Bluetooth**, pentru a putea fi salvate într-o bază de date și analizate.

□ Scopul și ideea de la care am pornit:

Mi-am propus să ajut hotelurile să îmbunătățească confortul clienților și monitorizarea automată a comportamentului din camere, printr-un sistem accesibil care:

- avertizează discret asupra depășirilor de zgomot
- colectează date climatice
- automatizează notificarea personalului

□ De ce cred că este util:

- Pentru hoteluri, ajută la gestionarea plângerilor, păstrarea liniștii și prevenirea abuzurilor.
- Pentru personal, oferă acces la date și alerte în timp real fără intervenție manuală.
- Pentru clienți, contribuie la un mediu mai liniștit și mai controlat.

Descriere generală

Diagrama Bloc

Dispozitivul este format din următoarele componente:

- □ Modul microfon (analogic LM393) – detectează nivelul de sunet ambiental

- □ Senzor DHT22 – măsoară temperatura și umiditatea
- □ LED – se aprinde dacă zgomotul depășește pragul
- □ Modul microSD (SPI) – salvează datele local pentru verificare ulterioară
- □ Modul Bluetooth HC-05 – transmite datele către un laptop
- □ ATmega328P – citește toți senzorii, decide acțiunile și comunică cu cardul SD și Bluetooth
- □ Alimentare 5V (USB sau power bank) – pentru toate modulele



Interacțiuni între module

Microfonul (LM393) trimite un semnal analogic către pinul A0 al microcontrollerului. Acesta este citit prin ADC pentru a detecta dacă sunetul depășește un prag.

ATmega328P:

- compară valoarea semnalului de la microfon cu un prag prestabilit.
- aprinde LED-ul dacă pragul este depășit.
- citește temperatura și umiditatea prin protocol digital de la DHT22 (pe pinul D4).
- trimite periodic toate datele (zgomot, temperatură, umiditate) prin Bluetooth (HC-05).
- salvează aceleași date pe cardul SD prin interfața SPI (pinii D10–D13).

DHT22:

- măsoară temperatura și umiditatea aerului.
- trimite datele către ATmega328P o dată la câteva secunde, în format digital.

LED-ul:

- este aprins de către microcontroller atunci când sunetul este considerat excesiv.

Modulul microSD:

- primește comenzi prin SPI și salvează loguri locale cu valorile măsurate.

Modulul Bluetooth HC-05:

- primește date de la microcontroller prin SoftwareSerial (pinii D2 și D3).
- le transmite în timp real către un laptop, pentru afișare și salvare.

Laptopul:

- rulează un script Python care citește datele prin Bluetooth (COM4).
- salvează periodic datele într-o bază de date masuratori.db.
- dacă detectează 3 depășiri de zgomot într-un minut, salvează în complaints.db și trimite automat un email de alertă.

Alimentarea de 5V (de la power bank) este comună tuturor modulelor și oferă energie pentru întreg sistemul.

Hardware Design

Bill of Materials (Lista Pieselor)

Nr.	Componentă	Descriere	Link / Datasheet
1	Microcontroler ATmega328P	Microcontroller 8-bit AVR	Datasheet ATmega328P
2	Modul microfon LM393	Senzor zgomot cu comparator LM393	Link modul
3	Modul microSD SPI	Stocare loguri	Link modul
4	Card microSD	Stocare date	Link card
5	Modul DHT22	Senzor de temperatură și umiditate	Link modul
6	Modul HC-05 Bluetooth	Comunicare wireless cu laptopul	Link modul
7	LED Roșu	Indicator vizual	-
8	Breadboard 400 puncte	Placă de prototipare	-
9	Cabluri tată-tată	Conectare componente	-
10	Rezistor 220Ω	Protejare LED	-
11	Baterie externă 5V	Alimentare portabilă	-

Schema Electrică

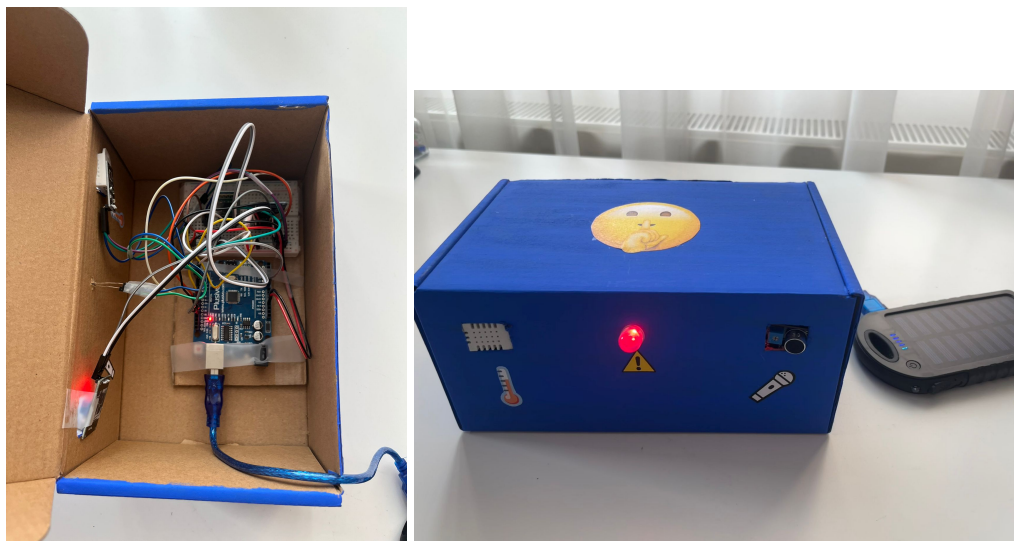


Tabel Legături Componente

Componentă	Pin Modul	Pin ATmega328P	Funcție
Modul Microfon LM393	OUT	A0	Citire semnal analogic (sunet)
	VCC	+5V	Alimentare
	GND	GND	Masă
LED	Anod (+)	D8	Control LED zgomot
	Catod (-) prin R 220Ω	GND	Masă (cu rezistor de protecție)
Modul microSD (SPI)	CS	D10	Chip Select SPI
	MOSI	D11	Date de la microcontroller către SD
	MISO	D12	Date de la SD către microcontroller
	SCK	D13	Ceas SPI
	VCC	+5V	Alimentare
	GND	GND	Masă
Modul DHT22	DATA	D4	Citire temperatură și umiditate
	VCC	+5V	Alimentare
	GND	GND	Masă
Modul Bluetooth HC-05	TXD	D3 (RX SoftwareSerial)	Date de la modul către microcontroller
	RXD	D2 (TX SoftwareSerial)	Date de la microcontroller către modul
	VCC	+5V	Alimentare
	GND	GND	Masă
	STATE / EN	-	(neutilizat în proiectul actual)

Alimentare 5V	VCC	+5V	Alimentare comună pentru toate modulele
	GND	GND	Masă comună

Componentele Hardware conectate



Software Design

Mediul de dezvoltare

- **Arduino IDE** - pentru programarea microcontrollerului ATmega328P.
- **Python 3** - pentru prelucrarea datelor primite prin Bluetooth și gestionarea bazelor de date + trimitere email.

Librării și surse 3rd-party

► Pe partea de Arduino:

- SPI.h - pentru comunicarea SPI cu modulul SD card
- SD.h - pentru logare fișiere text pe card microSD
- DHT.h - pentru citirea temperaturii și umidității de la senzorul DHT22
- SoftwareSerial.h - pentru comunicarea cu modulul Bluetooth HC-05 pe pini digitali

Toate aceste librării sunt standard sau pot fi adăugate ușor prin Library Manager în Arduino IDE.

► Pe partea de Python:

- serial (pyserial) - pentru citirea datelor prin Bluetooth
- sqlite3 - pentru salvarea datelor în fișiere .db locale
- smtplib și email.message - pentru trimiterea automată a emailurilor când se detectează zgomot excesiv

Algoritmi și structuri implementate

► În microcontroller (Arduino):

Se citesc:

- valoarea analogică a zgomotului (`analogRead(A0)`)
- temperatura și umiditatea prin DHT22

Dacă sunetul depășește un prag (> 100):

- se aprinde LED-ul
- se salvează valorile într-un fișier `log.txt` pe cardul SD

La interval de 5 secunde, se trimit prin Bluetooth valorile actuale

► În scriptul Python (PC/laptop):

Se deschid două baze de date SQLite:

- `masuratori.db` – pentru salvarea temperaturii și umidității
- `complaints.db` – pentru evenimentele de tip „noise complaint”

Se analizează fluxul de date din Bluetooth Dacă în decurs de 60 secunde se primesc 3 mesaje cu zgomot $>$ prag:

- se loghează evenimentul într-o bază separată
 - se trimite automat un email de avertizare
-

Surse și funcții implementate

► Arduino:

`setup()`:

- configurează pini I/O
- inițializează senzorii, Bluetooth-ul și cardul SD

`loop()`:

- citește continuu datele și trimite periodic valorile prin Bluetooth
- detectează evenimente de zgomot și le salvează pe SD

`logPeCard()`:

- creează/loghează într-un fișier text cu sunet, temperatură și umiditate

► Python:

`trimite_email()`:

- trimite notificare pe mail folosind contul Gmail configurat

while True:

- citește datele primite
- salvează în baza masuratori.db
- dacă se depășește pragul sonor de 3 ori în 60 secunde, salvează în complaints.db și trimite email

Rezultate Obținute

În urma implementării și testării proiectului, a fost realizat un sistem complet funcțional de monitorizare ambientală, capabil să:

- **Detecteze și semnalizeze zgomotul:**

Modulul de microfon analogic detectează nivelul de sunet din încăpere.

Când se depășește un prag predefinit, un LED de alertă se aprinde și evenimentul este salvat pe cardul microSD, împreună cu temperatura și umiditatea curente.

- **Salveze date local pe card microSD:**

Sistemul înregistrează valorile măsurate în fișierul log.txt.

Informațiile salvate includ: valoarea semnalului sonor, temperatura, umiditatea și momentul detectării (în milisecunde de la pornire).

- **Transmită datele în timp real prin Bluetooth:**

La fiecare 5 secunde, dispozitivul trimite date către un laptop sau alt dispozitiv conectat prin Bluetooth (HC-05).

Datele includ nivelul sunetului, temperatura și umiditatea.

- **Proceseze și analizeze datele în Python:**

Pe laptop, un script Python preia datele prin Bluetooth și le salvează într-o bază de date locală (masuratori.db).

La detectarea a 3 niveluri ridicate de zgomot într-un interval de 60 de secunde, sistemul:

- salvează evenimentul într-o bază separată (complaints.db);
- trimite automat un email de alertă către recepție (ex. ancutza@gmail.com, crinutza@gmail.com), notificând depășirile.

Validare

Sistemul a fost testat cu succes în condiții reale.

LED-ul se aprinde corect în funcție de zgomot, datele sunt salvate corect pe cardul SD, valorile sunt transmise prin Bluetooth, iar emailurile de alertă se trimit automat conform regulii definite.

Download

[Export to PDF](#)

Bibliografie/Resurse

Resurse Hardware

- [ATmega328P Datasheet](#)
- [Placă de dezvoltare Arduino UNO](#)
- [Modul microfon](#)
- [Modul microSD](#)
- [Card microSD](#)

Resurse Software

- [Arduino IDE](#)
- [Librărie SD.h - Documentație oficială](#)
- [Librărie SPI.h - Documentație oficială](#)
- [Documentație generală Arduino UNO + senzori audio](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/apredescu/anca.radu2411>



Last update: **2025/05/28 10:38**