

# Smart Vitals Monitor & Local Logger

## Introducere

**Ce face proiectul:** Proiectul meu este un dispozitiv portabil, numit Smart Vitals Monitor & Local Logger, care îți măsoară pulsul și temperatura pe loc. Valorile apar în timp real pe un ecran LCD, iar sistemul te avertizează imediat prin sunete (buzzer) și lumini (LED care pulsează) dacă datele arată că ceva nu este în regulă. În plus, aparatul ține minte ultimele măsurători importante într-un istoric local, pe care îl poți răsfoi oricând cu ajutorul unui buton rotativ.

**Care este scopul lui:** Scopul principal este să creez un aparat de monitorizare medicală simplu și accesibil, care să nu depindă de internet și care să funcționeze singur ca un mic asistent de sănătate, portabil.

**Ideea de la care am pornit:** Ideea mi-a venit inspirându-mă din studiu clinic propus pe Teams, unde se monitoriza ritmul cardiac și tensiunea “pacienților”. Am vrut să experimentez cu senzorul de puls și să transform conceptul de monitorizare medicală într-un proiect pe care să-l pot construi singură.

**Utilitatea:** Proiectul este util deoarece oferă o metodă rapidă de a verifica temperatura și pulsul. El salvează datele pentru a vedea evoluția stării persoanei chiar dacă aparatul este oprit.

## Descriere generală

- Arduino ATmega328P gestionează toate operațiunile și procesează datele primite, fiind fundamental pentru coordonarea activităților întregului sistem, de la colectarea datelor senzorilor până la afișarea informațiilor și alertarea utilizatorului.
- Senzorul optic de puls (Pulse Sensor) este utilizat pentru a măsura ritmul cardiac în timp real prin metoda optică de fotopletismografie. Semnalul analogic generat este citit continuu prin perifericul ADC al microcontrollerului, iar valorile BPM sunt calculate folosind un timer non-blocant, asigurând o monitorizare fluidă fără a bloca execuția celorlalte funcții ale sistemului.
- Senzorul de temperatură LM35 măsoară temperatura corporală sau ambientală, adăugând un strat suplimentar de date relevant pentru interpretarea stării de sănătate. Acesta este multiplexat pe același periferic ADC cu senzorul de puls, iar sistemul poate ajusta alertele în funcție de valorile înregistrate, oferind o monitorizare mai completă utilizatorului.
- Ecranul LCD 1602 cu modul adaptor I2C este folosit pentru a prezenta informațiile colectate într-un format accesibil, permițând utilizatorului să urmărească în timp real valorile vitale, cum ar fi ritmul cardiac și temperatura. Comunicarea prin protocolul I2C simplifică conexiunile fizice, reducând numărul de pini utilizați.
- Rotary Encoder-ul permite navigarea printr-un meniu cu stări multiple, oferind utilizatorului posibilitatea de a comuta între modul de Măsurare Live și modul Istoric. Acesta este gestionat curat

prin întreruperi hardware (INT/PCINT), asigurând o interacțiune rapidă și precisă fără pierderi de evenimente.

- Memoria EEPROM internă a microcontrollerului este utilizată pentru jurnalizarea locală a măsurătorilor importante. Datele sunt salvate persistent, rămânând disponibile chiar și după oprirea dispozitivului, permițând astfel urmărirea evoluției parametrilor vitali în timp.
- LED-ul RGB controlat prin PWM și Buzzer-ul sunt integrate pentru a furniza alerte vizuale și auditive atunci când valorile monitorizate depășesc pragurile predefinite. LED-ul își schimbă culoarea și comportamentul în funcție de tipul alertei: verde static pentru parametri normali, portocaliu pulsând pentru problemă de puls, portocaliu static pentru depășire de temperatură și roșu pulsând când ambii parametri sunt în afara limitelor.



## Hardware Design

Nume Componentă	Cantitate
Microcontroller Arduino Uno (ATmega328P)	1 set
Senzor de puls	1
Modul senzor de temperatură LM35	1
Display LCD 1602	1
Modul adaptor I2C pentru LCD	1
Modul Rotary Encoder KY-040	1
Modul Buzzer Activ 5V	1
LED-uri 5mm (2 culori)	1
Rezistențe 220 Ohm 0.25 W	1 set
Breadboard 840 puncte	1
Fire Dupont Tată-Tată 20cm	1 set
Fire Dupont Mamă-Tată 20cm	1 set

## 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/prj2026/cezar.zlatea/alara.hagi> 

Last update: **2026/05/09 21:20**