Monitor inteligent de puls cu memorie și display

Introducere

Proiectul pe care il voi realiza este un monitor inteligent de puls, care masoara ritmul cardiac la intervale regulate. Acesta afiseaza valoarea pulsului pe un ecran LCD, urmand ca dupa aceea sa semnaleze utilizatorul daca pulsul este in limite normale prin intermediul unor LED-uri. Ulterior, toate masuratorile vor fi salvate automat pe un SD card pentru analiza ulterioara.

Scopul proiectului este acela de a realiza un dispozitiv accesibil care sa permita monitorizarea ritmului cardiac in timp real, fara a fi nevoie de echipamente medicale costisitoare.

Ideea de la care am pornit o reprezinta dorinta de a construi un sistem simplu si portabil care sa poata oferi informatii esentiale despre starea de sanatate a unei persoane, in special in situatiile in care accesul la aparatura medicala este limitat.

Consider ca proiectul este util deoarece poate fi folosit de oricine doreste sa isi monitorizeze pulsul acasa, mai ales in cazul persoanelor cu afectiuni cardiovasculare sau sportivi.

Descriere generală

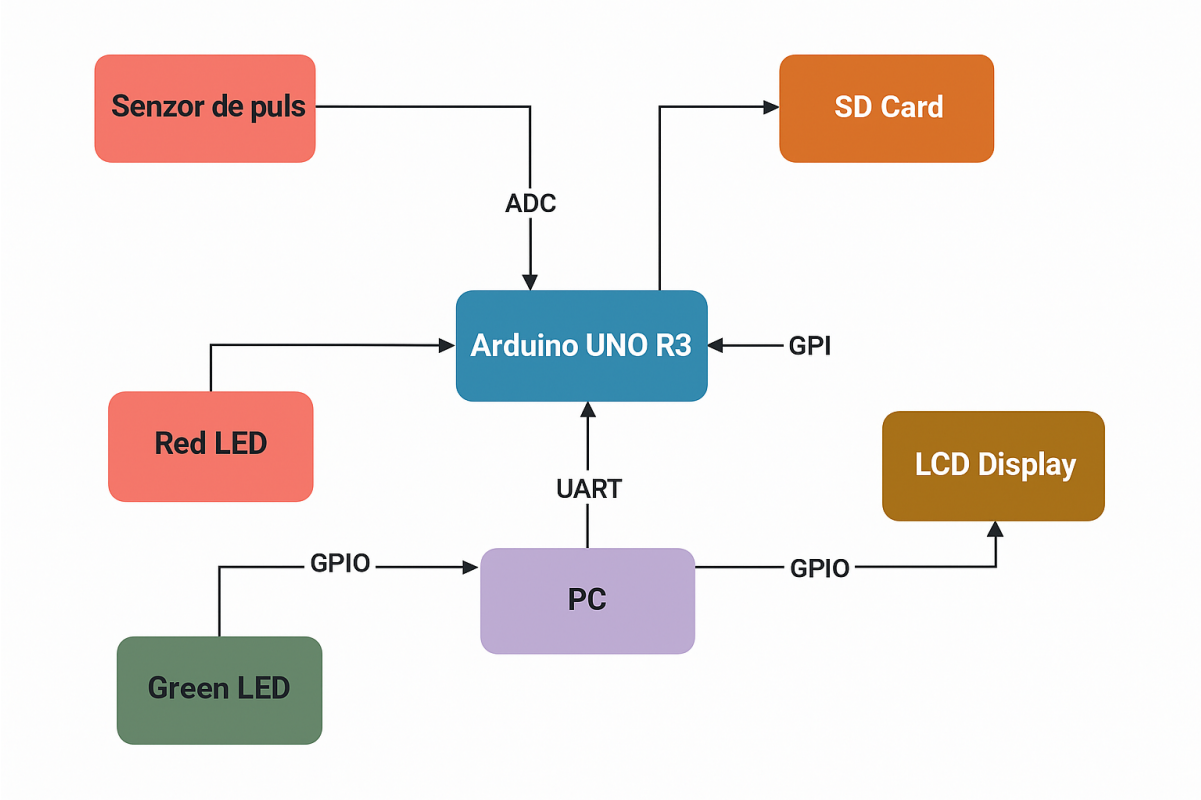
Proiectul consta in realizarea unui sistem inteligent de puls care:

- masoara pulsul utilizatorului la intervale regulate (5 secunde)
- semnaleaza vizual daca valoarea este in limite normale sau nu (prin intermediul unor LED-uri, daca valoarea este in limite normale, se aprinde LED-ul verde, iar daca valoarea nu este in limite normale, se aprinde LED-ul rosu)
- afiseaza in timp real valoarea pulsului pe un LCD Display
- stocheaza datele intr-un SD card, pentru analiza ulterioara

Sistemul foloseste un microcontroller Arduino UNO, un senzor de puls, un modul SD card, un ecran LCD cu interfata I2C si doua LED-uri pentru semnalizare vizuala. Masuratorile sunt realizate automat la fiecare 5 secunde cu ajutorul unui timer intern, fara a fi necesara interventia utilizatorului.

Pulsul este comparat cu un interval considerat optim (60-85 batai/minut), iar in functie de rezultat se aprinde LED-ul verde, in cazul unui puls normal, aflat intre aceste valori, sau LED-ul rosu, in cazul unui puls anormal, care nu se afla intre aceste valori. Toate valorile sunt inregistrate pe SD card. Aceste date vor fi trimite prin UART catre calculator.

Schema bloc:



Hardware Design

Listă de componente

#	Componentă	Datasheet	Rol în proiect	Modul	Sursă achiziție
1	Arduino UNO R3	ATmega328P	Unitate principală de control	Puls și afișare	https://hobbymarket.ro/platforme-dezvoltare/kit-arduino-starter-uno-r3-p-604.html
2	Pulse Sensor HW-827	-	Măsoară ritmul cardiac al utilizatorului	Puls	https://sigmanortec.ro/Senzor-puls-p126262523
3	LCD I2C 16×2	PCF8574 I2C	Afișează pulsul măsurat și eventuale mesaje	Afișare	https://www.optimusdigital.ro/ro/optoelectronice-lcd-uri/2894-lcd-cu-interfata-i2c-si-backlight-albastru.html
4	Card SD	microSD	Stochează datele despre puls în timp	Salvare date	-
5	Modul cititor SD (SPI)	SPI-based SD reader	Permite comunicarea între Arduino și card SD	Salvare date	https://sigmanortec.ro/Modul-card-SD-p137611367
6	LED verde	-	Se aprinde dacă pulsul este normal	Semnalizare vizuală	https://hobbymarket.ro/platforme-dezvoltare/kit-arduino-starter-uno-r3-p-604.html
7	LED roșu	-	Se aprinde dacă pulsul este prea mare	Semnalizare vizuală	https://hobbymarket.ro/platforme-dezvoltare/kit-arduino-starter-uno-r3-p-604.html
8	Rezistențe 220 Ω	-	Limitează curentul prin LED-uri	Protecție LED-uri	https://hobbymarket.ro/platforme-dezvoltare/kit-arduino-starter-uno-r3-p-604.html

9	Breadboard + fire	—	Permite conexiuni rapide între componente	Conectare / Prototipare	https://hobbymarket.ro/platforme-dezvoltare/kit-arduino-starter-uno-r3-p-604.html
10	Sursă alimentare USB	-	Oferă alimentare pentru întregul sistem	Alimentare	https://hobbymarket.ro/platforme-dezvoltare/kit-arduino-starter-uno-r3-p-604.html

Conexiuni si pini

- **Pulse Sensor HW-827** – măsoară ritmul cardiac al utilizatorului:

- Signal → A1
- VCC → 5V
- GND → GND

Justificare: A1 este un pin analog compatibil cu semnalul analogic generat de senzorul de puls, permițând citirea variațiilor corespunzătoare bătăilor inimii.

- **LCD 16×2 I2C (PCF8574)** – afișează pulsul și alte mesaje:

- GND → GND
- VCC → 5V
- SDA → A4
- SCL → A5

Justificare: A4 și A5 sunt pinii dedicați I2C pe Arduino UNO. Conexiunea I2C permite transmiterea eficientă a datelor către afișaj folosind doar două fire.

- **LED verde** – indică puls normal:

- Anod → pin 7 (prin rezistor de 220Ω)
- Catod → GND

Justificare: LED-ul este aprins sau stins de către Arduino în funcție de starea pulsului, pinul 7 fiind un pin digital de ieșire.

- **LED roșu** – indică puls ridicat sau periculos:

- Anod → pin 6 (prin rezistor de 220Ω)
- Catod → GND

Justificare: LED-ul roșu oferă o avertizare vizuală pentru ritm cardiac anormal, fiind controlat de Arduino prin pinul digital 6.

- **Modul cititor microSD (SPI)** – salvează datele despre puls:

- CS → pin 4

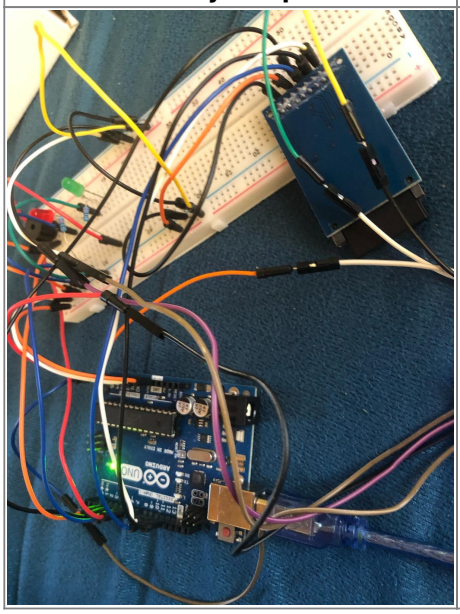
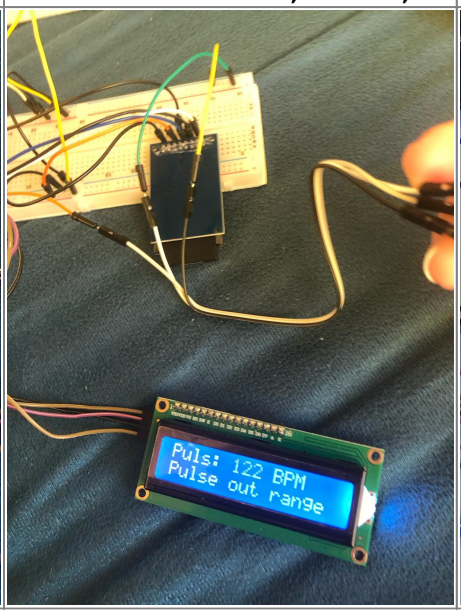
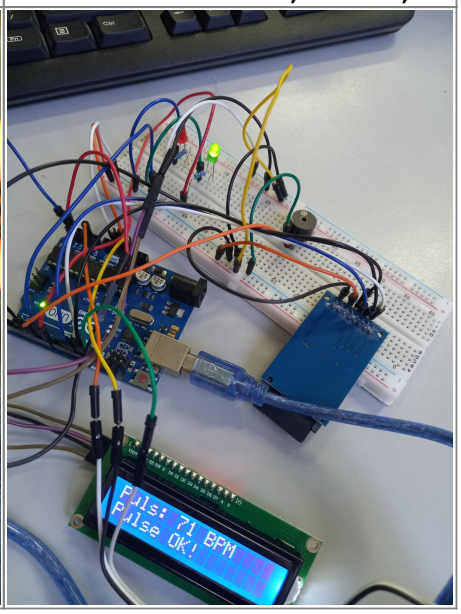
- MOSI → pin 11
- MISO → pin 12
- SCK → pin 13
- VCC → 5V
- GND → GND

Justificare: Modulul SD folosește comunicația SPI, iar pinii 11, 12 și 13 sunt dedicați acestei interfețe pe Arduino UNO. Pinul 4 este selectat ca CS (Chip Select) pentru activarea dispozitivului.

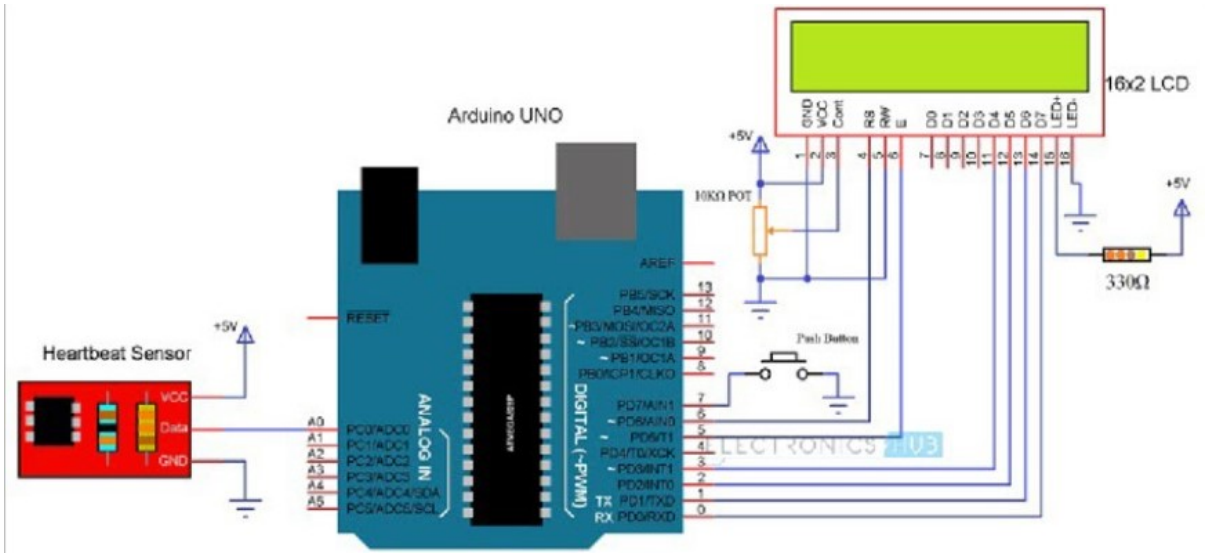
Stadiul actual al implementării hardware

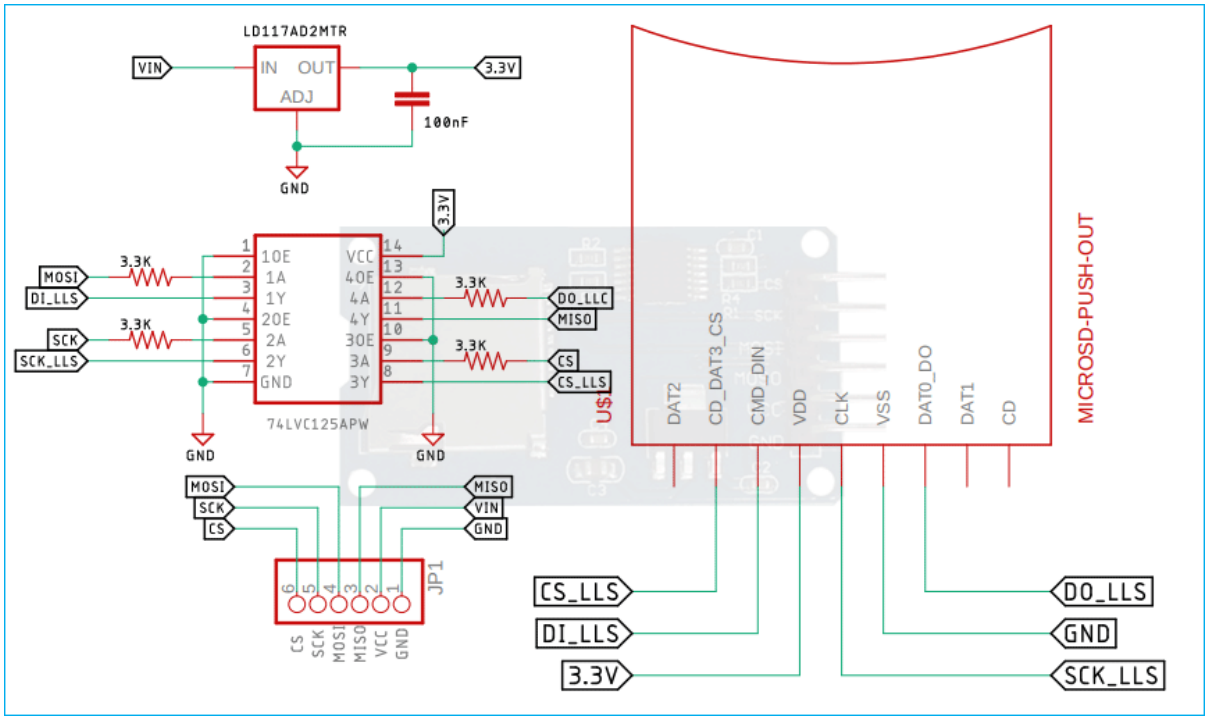
Structura hardware a proiectului prezentat în imagini este centrată pe măsurarea și afișarea ritmului cardiac, fiind împărțită într-un modul principal care integrează funcții de monitorizare și alertare. Sistemul este construit pe platforma Arduino UNO, care coordonează toate componentele conectate: un senzor de puls (senzor HW-827), un afișaj LCD 16×2 cu interfață I2C pentru afișarea valorilor BPM (bătăi pe minut), un LED roșu și unul verde pentru semnalizarea stării pulsului.

Circuitul este realizat pe o breadboard, cu fire de legătură ce conectează senzorul, ecranul și LED-urile la pinii digitali și analogici ai plăcii Arduino. Pe ecran este afișat pulsul măsurat, iar dacă acesta este în afara intervalului normal, apare mesajul „Puls anormal”, moment în care LED-ul roșu se aprinde.. În cazul unui puls normal, se aprinde LED-ul verde. Montajul este funcțional și a fost testat cu succes, așa cum se poate observa din valorile afișate pe ecran.

Montaj complet	Demonstrarea funcționalității	Demonstrarea funcționalității
		

Schema Electrica





Software Design

Stadiul actual al implementării software

La momentul actual, firmware-ul este complet funcțional, implementând toate funcțiile esențiale:

- Citirea valorilor de la **senzorul de puls** (analogic);
- Filtrarea semnalului pentru stabilitate;
- Detectarea bătăilor inimii și calcularea **BPM** (bătăi pe minut);
- Afișarea bătăilor pe minut (BPM) pe un **LCD I2C** și semnalizarea vizuală;

- Salvarea valorii BPM pe un **card SD**;
- Trecerea automată în mod standby dacă nu mai este detectat pulsul.

Toate funcțiile au fost testate individual și integrate într-o structură robustă.

Motivarea alegerii bibiotecilor utilizate

Bibliotecă	Rol în proiect	Motivație
LiquidCrystal_I2C.h	Controlul afișajului LCD prin interfață I2C	Economisește pini digitali, oferă o interfață intuitivă pentru a afișa mesaje pe LCD
Wire.h	Comunicație I2C între Arduino și componente (de exemplu LCD)	Este biblioteca standard pentru comunicația I2C, necesară pentru funcționarea LCD-ului I2C
SD.h	Acces și scriere pe cardul SD	Permite stocarea persistentă a valorilor BPM într-un fișier text pe cardul SD
SPI.h	Comunicație SPI între Arduino și modulul de card SD	Necesară pentru funcționarea bibliotecii **SD.h**, gestionează eficient protocoalele SPI

Elementul de noutate al proiectului

Proiectul propune două elemente distincte:

- Afișaj activ adaptiv: Sistemul detectează absența pulsului și trece automat într-un mod de „așteptare”, eliminând afișarea valorilor învechite (fals pozitive).
- Filtrare software low-cost: În locul hardware-ului complex, se folosește o medie mobilă pentru filtrare, obținând stabilitate mare fără componente externe.

Integrarea funcționalităților din laborator în cadrul proiectului

Proiectul realizat reflectă aplicarea practică a conceptelor și funcționalităților studiate în cadrul laboratoarelor, integrând într-un mod coerent cunoștințele teoretice în dezvoltarea unui sistem ce utilizează microcontrollere. Tabelul de mai jos prezintă corespondența dintre modulele hardware, tehnicile folosite în proiect și laboratoarele aferente în care acestea au fost studiate:

Laborator	Funcționalitate	Utilizare în proiect
GPIO (General Purpose I/O)	Controlul LED-urilor, citirea senzorului analogic	LED-urile (roșu și verde) sunt controlate pentru semnalizare vizuală; senzorul de puls este citit pe pin analogic A1
UART (Serial Communication)	Comunicare serială între PC și Arduino	Trimiterea valorilor BPM și a mesajelor de stare către monitorul serial prin Serial.print()

PWM (Pulse Width Modulation)	Generare semnal modulat în impulsuri	Utilizat pentru citirea semnalului trimis catre senzor
ADC (Analog to Digital Convertor)	Conversie analog-digital (citirea tensiunii analogice și transformarea în valoare digitală)	Citirea valorii de la senzorul de puls prin analogRead(PULSE_PIN)
SPI (Serial Peripheral Interface)	Comunicație serială rapidă între microcontroler și periferice (ex. card SD, senzori)	Utilizat de biblioteca SD.h pentru scrierea datelor pe cardul microSD
I2C (Inter-Integrated Circuit)	Comunicație serială pe două fire între microcontroler și periferice	Folosit împreună cu Wire.h și LiquidCrystal_I2C pentru afișarea datelor pe LCD

Scheletul proiectului

Proiectul este implementat pe o singură plăcuță Arduino UNO, care gestionează toate funcționalitățile într-un sketch unic. Arhitectura este modulară la nivel software, fiecare componentă având un rol clar definit.

Funcționalitate	Descriere
Măsurarea pulsului	Citirea semnalului analogic de la un senzor de puls (prin ADC) și detecția bătăilor inimii
Filtrarea semnalului	Aplicarea unui filtru mediu mobil pentru reducerea zgomotului din datele brute
Calcul BPM	Calcularea frecvenței cardiace în BPM, cu mediere pe 5 valori pentru stabilitate
Semnalizare vizuală	Afișarea pe LCD I2C și aprinderea LED-urilor (verde/roșu) în funcție de intervalul BPM
Scriere pe card SD	Logarea valorilor BPM în fișierul puls.txt pe cardul SD, folosind protocolul SPI
Gestionare stare de standby	Detectarea absenței pulsului și revenirea la ecranul de “Aștept puls…” după 3 secunde de inactivitate

Interacțiunea între funcționalități

Funcționalitate	Descriere	Interacțiuni principale
Citirea valorilor analogice	Se citește valoarea brută de la senzorul de puls folosind analogRead()	Trimite date către filtrare și detecția pulsului
Filtrarea semnalului	Aplică un filtru mediu mobil pe ultimii 5 valori pentru reducerea zgomotului	Transmite semnalul filtrat către detecția bătăilor
Detecția bătăilor	Verifică dacă semnalul depășește un prag (cu histerezis), marcând o bătaie nouă	Activează calculul BPM și actualizarea bufferului de valori
Calculul BPM	Calculează intervalul între bătăi și derivează BPM-ul mediu (dintr-un buffer de 5 valori)	Trimite BPM-ul către afișare și logare
Afișarea pe LCD	Afișează valorile BPM sau mesajul “Aștept puls…” pe ecranul LCD I2C	Primește date de la calcul BPM și gestiunea standby

Semnalizare LED	LED-urile indică dacă pulsul este normal sau anormal	Se activează după afișarea BPM-ului, în funcție de interval
Logare pe SD	Salvează valoarea BPM într-un fișier **.txt** de pe card SD	Primește BPM calculat și îl scrie în fișier
Mod standby (inactivitate)	Dacă nu e detectată nicio bătaie în 3 secunde, se revine la starea “Aștept puls...”	Se activează automat din loop() și oprește LED-urile și afișajul activ

Fluxul de date

1. analogRead() → filtru → detectie puls

2. Dacă e bătăie a inimii:

- calcul BPM
- update buffer

3. Dacă bufferul e complet:

- media BPM
- afișare LCD
- semnalizare LED
- scriere pe SD

4. Dacă nu este detectat puls timp de 3 secunde:

- standby display și oprire feedback

Modul de validare a funcționalității

Testarea sistemului a fost efectuată prin evaluare practică, monitorizare continuă în timp real și compararea comportamentului observat cu cel așteptat. Rezultatele obținute atestă funcționarea corectă atât a componentelor hardware, cât și a logicii implementate în cod.

- **Senzor de puls (analogic)** :
  - **Metodă de testare:** A fost plasat degetul pe senzor și semnalul analogic a fost monitorizat în Serial Monitor pentru variații corespunzătoare bătăilor inimii.
  - **Rezultat:** Sistemul detectează consistent bătăile, iar valorile BPM sunt afișate pe LCD și în consola serială, fără erori sau latență mare.

- **Filtrare semnal și detecție puls** :
  - **Metodă de testare:** Valorile analogice brute și cele filtrate au fost comparate în timp real. Detecția pulsului a fost verificată prin comparație cu ceasul.
  - **Rezultat:** Media mobilă filtrează zgomotul eficient. Detecția este stabilă, fără declanșări false, chiar în prezența variațiilor de tensiune ușoare.

- **LED-urile care determină starea** :
  - **Metodă de testare:** S-au simulat diverse intervale de BPM (sub 60, între 60–100, peste 100).
  - **Rezultat:** LED-ul verde se aprinde doar în intervalul normal, LED-ul roșu în afara acestuia. Comportamentul este corect și sincronizat cu afișajul LCD.
- **Afișaj LCD I2C (LiquidCrystal_I2C)** :
  - **Metodă de testare:** Afișarea textului a fost monitorizată pentru toate stările: puls detectat, standby, eroare SD.
  - **Rezultat:** Textul este clar, actualizat în timp real. Funcționare fără flicker, corect sincronizat cu logica principală.
- **Card SD** :
  - **Metodă de testare:** Fișierul puls.txt a fost verificat pe PC pentru a confirma salvarea valorilor BPM la fiecare actualizare.
  - **Rezultat:** Datele sunt salvate corect, cu timestamp relativ. SD-ul este recunoscut fiabil, iar mesajele de eroare sunt corect afișate dacă lipsește.
- **Comportament standby** :
  - **Metodă de testare:** A fost simulat scenariul fără puls timp de peste 3 secunde.
  - **Rezultat:** Sistemul intră automat în mod standby, oprește LED-urile, și afișează „Astept puls…”. Se reactivează imediat la detectarea unui puls nou.

Calibrarea senzorilor

Pentru a garanta precizia datelor, a fost necesară calibrarea **senzorului de puls**.

Procesul de calibrare:

- În timpul fazei inițiale a execuției, sistemul intră într-o perioadă de calibrare de 6 secunde (CAL_TIME_MS = 6000 ms). În această perioadă, sunt citite valorile maxime generate de senzor, pentru a determina un prag de detecție (threshold) optim. Acest prag este calculat ca: **threshold = baselineMax + SAFETY_MARGIN;** unde baselineMax este cea mai mare valoare înregistrată în timpul calibării, iar SAFETY_MARGIN este o valoare prestabilită (25 unități ADC) pentru a preveni declanșări false.

Validare:

- Detecția bătăilor inimii a fost testată pe utilizatori diferiți, confirmând că:
- sunt ignorate semnalele neregulate datorită aplicării unei filtrări medii mobile (cu 5 eșantioane),
- există o distanță minimă între bătăi (MIN_BEAT_SPACING = 600 ms) pentru a elimina erorile,
- o **isterezis** de 20 unități ajută la stabilizarea semnalului în jurul valorii prag.

Frecvența de eșantionare:

- Citirea senzorului se face la fiecare 10 ms (SAMPLE_RATE_MS), asigurând o rezoluție bună fără a supraîncărca procesorul.

Realizarea optimizărilor

- **Filtrare software** (medie mobilă): evită semnal fals și îmbunătățește detecția fără componente externe.
- **Standby inteligent**: oprirea automată a afisajului și LED-urilor dacă nu se mai simte puls – economie de energie și mai multă claritate.
- **Buffer circular BPM**: stocare optimă pentru mediere, evitând suprascrierea haotică.

Rezultate Obţinute

Proiectul a fost testat în condiții reale, folosind mai mulți utilizatori pentru a putea evalua acuratețea detecției pulsului, comportamentul interfeței vizuale și funcționarea generală a sistemului. Astfel, rezultatele obținute sunt următoarele:

1. **Detecția pulsului**
   - Sistemul a fost capabil să detecteze cuprins între 55 și 110 BPM, în funcție de utilizator și de efort fizic.
   - În stare de repaus, utilizatorii au avut un puls afișat stabil de 61-75 BPM, cu o variație de +-2 BPM între măsurători consecutive.
   - În stare activă, după realizarea a câteva exerciții fizice ușoare, valorile afișate au crescut progresiv până la 95-105 BPM, apoi au revenit la normal după câteva minute, confirmând răspunsul în timp real al senzorului.
2. **Fiabilitatea sistemului**
   - Senzorul de puls a fost capabil să detecteze lipsa semnalului după câteva secunde (aproximativ 3 secunde fără să fi detectat un puls), moment în care afișajul pe LCD a revenit la mesajul de așteptare, iar LED-urile au fost dezactivate.
   - Nu au fost detectate alarme false sau valori imposibile datorită filtrării și restricțiilor de interval impuse (45-160 BPM)
3. **Afișaj pe LCD I2C**
   - Toate mesajele au fost afișate corect pe ecranul I2C. Textul este clar și vizibil, fără erori vizuale.
4. **Logare SD Card**
   - La fiecare actualizare a pulsului, sistemul a scris în fișierul **puls.txt** valoarea corespunzătoare obținută.
   - Fișierul rezultat conține un istoric complet al sesiunii de testare
   - Verificarea pe PC a confirmat că toate înregistrările au fost realizate corect, fără pierderi de date.
5. **Semnalizare vizuală**
   - LED-ul verde s-a aprins automat când BPM-ul era în intervalul 60-85.
   - LED-ul roșu s-a activat automat când BPM-ul ieșea din acest interval.

```
Salvat in puls.txt - BPM: 68.0
70.2
67.8
86.8
Salvat in puls.txt - BPM: 70.3
Salvat in puls.txt - BPM: 70.3
```

Mai jos se găsește atașat un link către un video cu funcționalitatea proiectului și valorile obținute.
https://mega.nz/file/QUAylYjB#QLeDn22n2eUCR9RPRIzO98EqX4kl8R7zjHL49KlEdfI

Concluzii

Proiectul pe care l-am realizat, și anume un sistem de măsurare, afișare și semnalizare a pulsului uman, demonstrează cu succes aplicabilitatea practică a noțiunilor teoretice studiate în cadrul cursului și laboratorului. Funcționalitățile implementate au fost validate în condiții reale, obținându-se un sistem cu numeroase caracteristici:

1. **Fiabil**
   - Detecția pulsului s-a realizat cu acuratețe, filtrând variațiile anormale și ignorând semnalele false.
   - Sistemul răspunde rapid la modificările ritmului cardiac și semnalizează eficient depășirea limitelor normale.
2. **Ușor de utilizat**
   - Interfața cu utilizatorul (LCD și LED-uri) oferă feedback intuitiv și imediat.
   - Afișajul este clar, iar sistemul revine automat în modul de așteptare când nu este detectat un puls.
3. **Modular și extensibil**
   - Codul este structurat, facilitând adăugarea de funcții suplimentare
   - Poate fi adaptat cu ușurință pentru a integra și alți senzori biomedicali .
4. **Relevant din punct de vedere educațional**
   - Proiectul aplică direct concepte fundamentale precum: ADC, I2C, SPI, PWM, debounce software, filtrare semnal și SD logging.
   - Toate componentele și tehnologiile utilizate au fost abordate în laboratoarele aferente, consolidând învățarea practică.

Download

Arhiva proiectului: Turcitu Luiza-Alexandra, 333AA → proiect_pm_turcitu_luiza_3333aa.zip

Bibliografie/Resurse

Resurse HARDWARE

- Senzor de puls HW-827 : https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/, https://github.com/WorldFamousElectronics/PulseSensorPlayground
- Cititor SD Card și microSD : https://lastminuteengineers.com/arduino-micro-sd-card-module-tutorial/
- LED-uri : https://docs.arduino.cc/built-in-examples/basics/Blink/

Resurse SOFTWARE

- **LiquidCrystal_I2C** : https://github.com/johnrickman/LiquidCrystal_I2C, https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/, https://docs.arduino.cc/libraries/liquidcrystal/
- **SD.h** și **SPI.h** : https://docs.arduino.cc/libraries/sd/
- **Wire.h** : https://docs.arduino.cc/language-reference/en/functions/communication/wire/, https://playground.arduino.cc/Main/I2cScanner/

Export to PDF

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2025/avaduva/luiza.turcitu

Last update: **2025/05/28 10:13**