

Heart Rate Monitor

Introducere

Proiectul implică dezvoltarea unui dispozitiv portabil de monitorizare a ritmului cardiac. Scopul principal al dispozitivului este de a înregistra diverse statistici legate de ritmul cardiac pe durata unor activități fizice, cum ar fi sesiuni de workout sau fitness.

Ideea acestui proiect a pornit de la dorința de a oferi sportivilor amatori o modalitate accesibilă și ușor de utilizat pentru a monitoriza și înțelege mai bine impactul activităților fizice asupra corpului lor.

Acest dispozitiv este util pentru oricine dorește să obțină feedback instantaneu despre starea sa de sănătate cardiovasculară în timpul exercițiilor, permițând ajustări ale intensității pentru maximizarea beneficiilor antrenamentului.

Descriere generală

Dispozitivul de monitorizare a ritmului cardiac funcționează astfel:

- Inițierea monitorizării:** Utilizatorul pornește sesiunea de monitorizare a pulsului apăsând butonul conectat la Arduino. Acest moment marchează începutul perioadei de activitate fizică.
- Colectarea datelor:** Îndată ce sesiunea este inițiată, senzorul de puls începe să înregistreze ritmul cardiac al utilizatorului prin intermediul unor LED-uri cu infraroșu și a unui fotodetector, și transmite aceste date la placa Arduino în timp real. De asemenea, un LED RGB va semnala fiecare bătaie a inimii înregistrată.
- Procesarea și stocarea datelor:** Datele de la senzor sunt colectate și procesate continuu de Arduino pe durata activității.
- Încheierea monitorizării:** Utilizatorul finalizează sesiunea de monitorizare apăsând din nou butonul. Acest lucru indică sfârșitul perioadei de activitate și inițiază procesul de afișare a datelor.
- Afișarea statisticilor:** În momentul încheierii sesiunii, Arduino procesează datele finale și le afișează pe display-ul LCD prin intermediul adaptorului I2C. Statisticile afișate includ ritmul cardiac minim/mediu/maxim sau durata totală a sesiunii. Aceste informații oferă utilizatorului o perspectivă detaliată asupra performanței sale cardiace pe durata activității.



Hardware Design

Lista pieselor și modul de conectare:

1. Arduino UNO

2. Display LCD 20×4 + Adaptor I2C

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

3. Senzor Pulsoximetru MAX30102

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

4. LED RGB

- *Red* → *Digital Pin 7*
- *Anod comun* → *5V*
- *Blue* → *Digital Pin 6*
- *Green* → *Digital Pin 5*

5. Push button

- *GND*
- *Digital Pin 2*

6. Powerbank

7. Breadboard

8. Jumper wires

9. Rezistente

În circuit și schematic, senzorul MAX30102 este reprezentat printr-un LED roșu și un fototranzistor (gruparea de pe partea stângă a breadboard-ului).

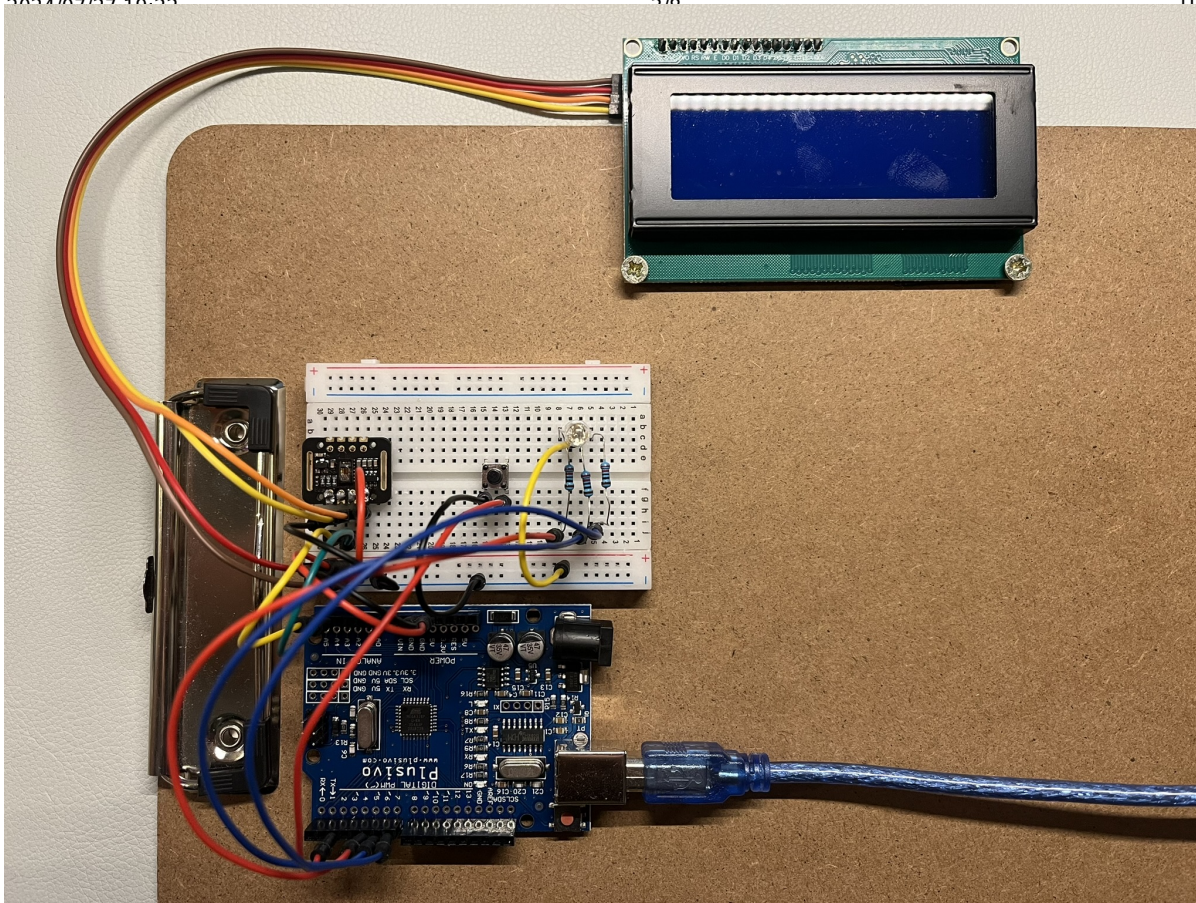
Circuit:



Schematic:



Stadiu curent:



Software Design

Biblioteci folosite

- **Wire.h** - bibliotecă standard care facilitează comunicarea I2C între Arduino și alte dispozitive I2C.
- **SparkFun_MAX3010x_Sensor_Library** - conține toate metodele de configurare/utilizare a senzorilor din gama MAX3010x a celor de la SparkFun, dintre care am avut nevoie de următoarele:
 - **MAX30105.h** - conține metode de configurare a senzorului (control on/off, setarea parametrilor de funcționare, etc.).
 - **heartRate.h** - conține metode specifice colectării de date referitoare la puls.
- **LiquidCrystal_I2C.h** - facilitează afișarea datelor pe ecranul LCD.
- **displayMethods.h** - bibliotecă creată de mine, pentru separarea logicii programului de definirea metodelor de afișare pe ecranul LCD.

Laboratoare folosite

- Laborator 1: **UART** → folosit pentru a afișa informații de debug.
- Laborator 2: **Înteruperi** → apăsarea butonului acționează o întrerupere pe INT0.
- Laborator 3: **PWM** → folosit pentru a controla intensitatea led-ului RGB.

• Laborator 6: I2C → senzorul de puls MAX30102 și LCD-ul folosesc I2C pentru comunicarea cu Arduino.

Descrierea logicii și metodelor folosite

- **setup()**: folosită pentru initializarea următoarelor componente:
 - Serial Monitor cu Baud Rate 115200 - pentru debug, în special pentru senzorul de puls
 - ecranul LCD
 - butonul cu rezistență internă de pull-up și întrerupere atașată: `attachInterrupt(digitalPinToInterrupt(2), buttonISR, CHANGE);`
`pinMode(2, INPUT_PULLUP);`
 - Led RGB cu anod comun, setat pe 255 (OFF) cu ***analogWrite()***.
 - senzorul de puls cu led IR (necesar colectării datelor despre puls) și Red (pentru a simboliza statusul on/off) active.
- **buttonISR()**: rutina de tratat a întreruperii, care setează valoarea flag-ului *buttonPressed* pe *TRUE* în cazul unei apăsări complete de buton (press & release). Înglobează și o tehnică de **debouncing** pentru filtrarea noise-ului.
- **loop()**:
 - verifică dacă butonul a fost apăsător (*buttonPressed*), caz în care re setează valoarea flag-ului pe *FALSE* și inversează valoarea flag-ului *isMonitoring* (pentru a putea face switch între starea de monitorizare a pulsului și cea de afișare a rezultatelor). Resetarea flag-ului *buttonPressed* pe *FALSE* permite executarea instrucțiunilor interioare o singură dată, la apăsarea butonului.
 - verifică dacă ne aflăm în starea de monitorizare, caz în care apelează metoda ***monitorHeartbeat()***. În acest caz, monitorizarea se va executa la fiecare apel al funcției ***loop()***, cât timp flag-ul *isMonitoring* este *TRUE*.
- **monitorHeartbeat()**:
 - obține valoarea returnată de fotodetector apelând funcția ***getIR()*** din biblioteca ***MAX30105.h***.
 - o valoare sub 50 000 indică absența degetului pe senzor, caz în care utilizatorul este îndrumat să pună degetul pe el.
 - altfel, folosesc metoda ***checkForBeat()*** din biblioteca ***heartRate.h*** pentru a detecta o bătaie a inimii. BPM-ul este determinat pe baza duratei de timp dintre bătăi, formula fiind aplicată de funcția auxiliară ***getBPM()***. Pentru fiecare bătaie a inimii de pe timpul monitorizării, led-ul RGB va clipici, cu efect de fade-out.

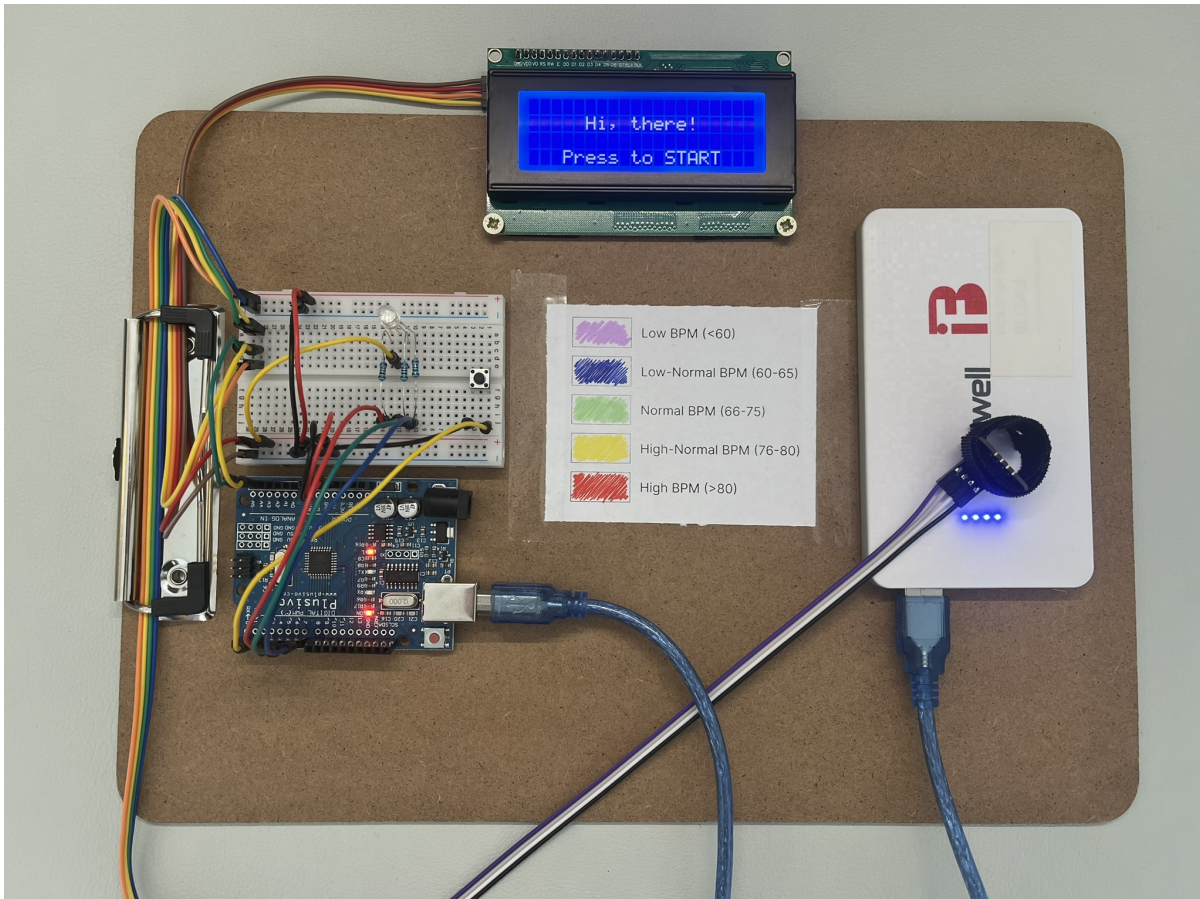
Pentru a **calibra** valorile obținute, utilizatorului îi vor fi afișate informațiile folosind următorul algoritm: BPM-ul afișat (calibrat) va reprezenta **media** ultimelor 6 valori raw obținute. Prin urmare, până la obținerea primelor 6 valori, *Heart Rate Monitor* se va afla într-o stare de **warm up**.

- **computeStats()**: apelată în ***loop()*** în momentul în care este apăsător butonul și flag-ul *isMonitoring* e setat pe *FALSE*, metoda calculează statisticile afișate utilizatorului la finalul perioadei de monitorizare.

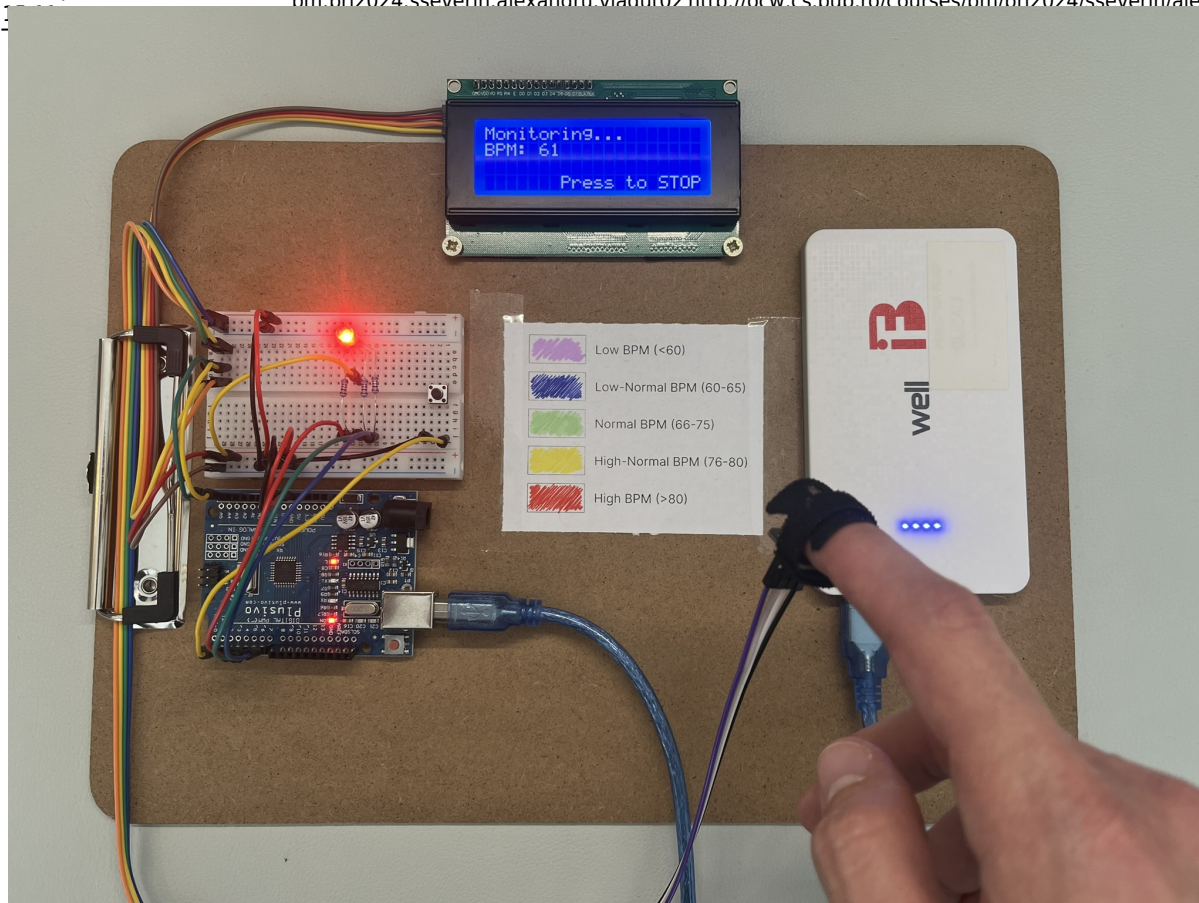
Codul și alte resurse folosite se pot găsi pe [GitHub](#).

Rezultate Obținute

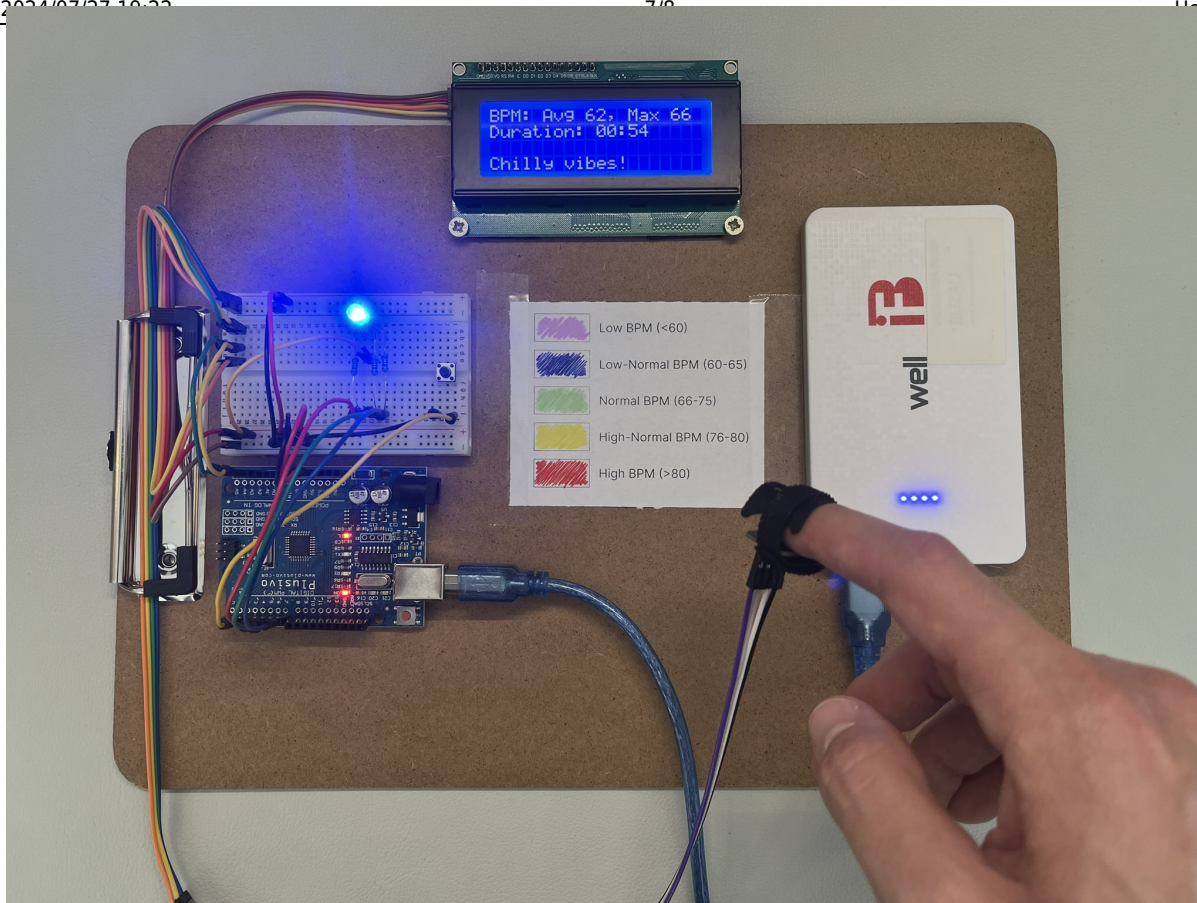
Pornirea *Heart Rate Monitor*



Starea de monitorizare, în care LED-ul clipește în culoarea roșie la fiecare bătaie a inimii.



Afișarea rezultatelor, moment în care LED-ul este aprins static în diferite culori, în funcție de BPM.



Concluzii

Senzorii de puls sunt foarte pretențioși, din mai multe puncte de vedere:

- În primul rând, cei de tip **HW-827** nu s-au dovedit a fi cei mai funcționali, în cazul meu cel puțin. După folosirea eșuată a 2 senzori de acest tip, am trecut la **MAX30102** (o replică a pulsoximetrului făcut de SparkFun).
- În al doilea rând, senzorul de puls are nevoie de o presiune constantă a degetului pentru a oferi valori accurate continuu. Oricât am crede noi, ca oameni, că aplicăm aceasta presiune constant, cu siguranță nu reușim (informație găsită și în documentația senzorului). Prin urmare, am atașat senzorul unei benzi cu scai reglabile, simulând un inel, însă nici acest lucru nu a adus o funcționare perfectă a senzorului (ocasional omite unele bătăi ale inimii).

Download

[vladut_alexandru-nicolae_332ca.zip](#)

Bibliografie/Resurse

Resurse Software:

- https://github.com/johnrickman/LiquidCrystal_I2C
- https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library
- <https://docs.arduino.cc/learn/microcontrollers/digital-pins/>
- <https://docs.arduino.cc/learn/microcontrollers/analog-input/>
- <https://docs.arduino.cc/learn/microcontrollers/analog-output/>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab1-2023>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab2-2023>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023-2024>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2023-2024>

Resurse Hardware:

- https://www.handsontec.com/dataspecs/I2C_2004_LCD.pdf
- <https://cdn.sparkfun.com/assets/f/e/9/4/1/MAX30101.pdf>
- <https://learn.sparkfun.com/tutorials/max30105-particle-and-pulse-ox-sensor-hookup-guide/all>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/sseverin/alexandru.vladut02>



Last update: **2024/05/27 15:11**