

Pulsometru

Student: Serban Emilia-Bianca, 333CC

Introducere

Proiectul presupune proiectarea unui dispozitiv care detectează și măsoară pulsul cardiac al utilizatorului. Acest tip de dispozitiv se numește pulsometru și are scop medical, acesta verificând cât de bine pompează inima sângele în corp și fiind folosit pentru monitorizarea stării de sănătate a utilizatorilor.

Pentru realizarea pulsometrului, voi utiliza o placă de dezvoltare Arduino UNO R3, un senzor de puls, un buton, un buzzer care va reda bătăile inimii, un ecran OLED care va afișa valoarea și graficul ritmului cardiac, și LED-uri care, după evaluarea pulsului, vor indica dacă acesta se află în parametrii normali.

Descriere generală

Mod de funcționare

- Se apasă butonul pentru a porni dispozitivul
- După apăsarea butonului, se înregistrează pulsul cardiac al utilizatorului prin intermediul senzorului de puls
- Se afișează valoarea și graficul ritmului cardiac pe ecranul OLED și/sau în interfața serială a calculatorului
- În timpul preluării pulsului, buzzer-ul va reda sunetul bătăilor inimii
- Se evaluează valoarea pulsului și, în funcție de intervalul în care se încadrează, LED-urile vor avea culoarea verde (puls normal), galben (puls ușor ridicat) sau roșu (puls ridicat)

Schema bloc



Hardware Design

Listă de piese:

- Arduino UNO R3 ATMEGA328P

- Senzor de Puls XD-58C
- Buzzer activ 5V
- LED-uri RGB 4mm
- Display OLED 0.96" I2C IIC Albastru



Mentionez ca in aceasta schema, senzorul de temperatura, de fapt, este un senzor de puls, iar buzzerul nu este acest model.

Software Design

Mediu de dezvoltare: Arduino IDE

Biblioteci:

- Pentru ecranul OLED:
 - Adafruit_GFX.h
 - Adafruit_SSD1306.h
 - Wire.h
 - SPI.h
- Pentru senzorul de puls:
 - PulseSensorPlayground.h

Descrierea codului

ISR si setup_interrup_INT0(): Se implementează o întrerupere externă INT0, care va fi declanșată de un buton extern conectat la pinul PD2. În interiorul întreruperii, se actualizează variabila de stare pentru a indica că se dorește urmărirea pulsului.

setup(): Se inițializează comunicarea serială și afișajul OLED. Apoi, se configurează LED-urile, buzzerul și managerul senzorului de puls pentru a utiliza pinul A0 și se setează pragul pentru detecția pulsului. Dacă inițializarea senzorului de puls nu reușește, programul va rămâne într-o buclă infinită cu LED-ul pulsând pentru a indica eroarea.

loop(): Se efectuează următoarele acțiuni, în funcție de starea curentă:

- Starea 0: Așteaptă declanșarea butonului pentru a începe monitorizarea pulsului. Se activează întreruperea externă INT0.
- Starea 1: Se inițializează măsurătorile pulsului și se efectuează citiri periodice ale valorilor senzorului de puls timp de 20 secunde. Se calculează suma valorilor și numărul de valori citite (valori detectate care au trecut Threshold-ul). De asemenea, buzzerul va reda la fiecare citire un sunet. Acesta este menit să redea acest sunet la fel de frecvent ca bataile inimii.

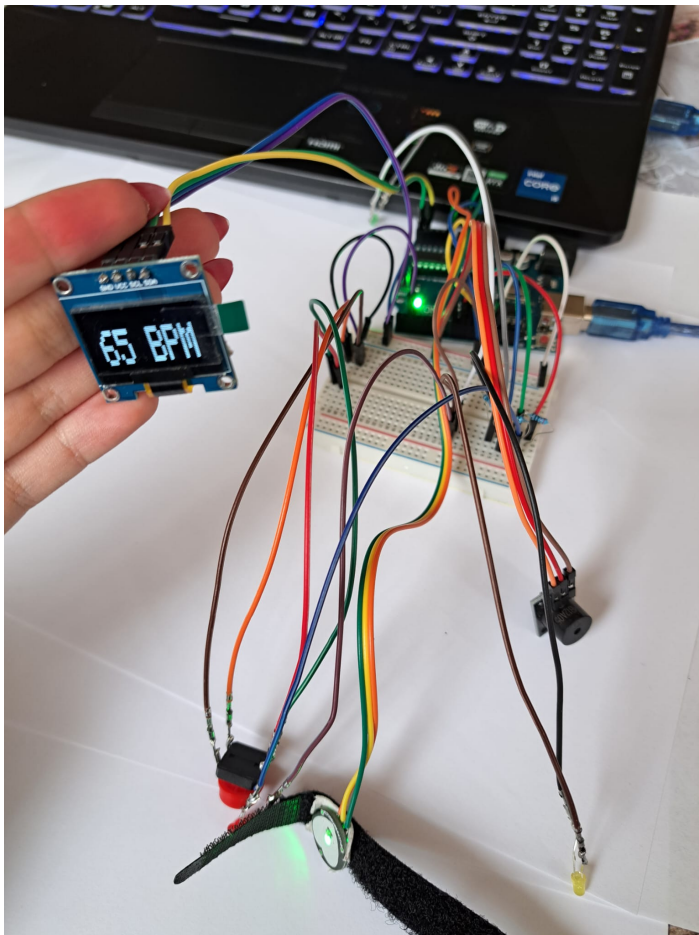
- Starea 2: După monitorizarea de 20 de secunde, se calculează rezultatele finale ca raportul dintre suma și numărul de valori citite, se afișează pe ecran și se evaluează valoarea medie a pulsului. În funcție de limitele prestabilite, LED-urile corespunzătoare se aprind în funcție de rezultat. După o pauză de 10 secunde, starea revine la 0, iar variabilele pentru suma și numărul de măsurători se resetează.

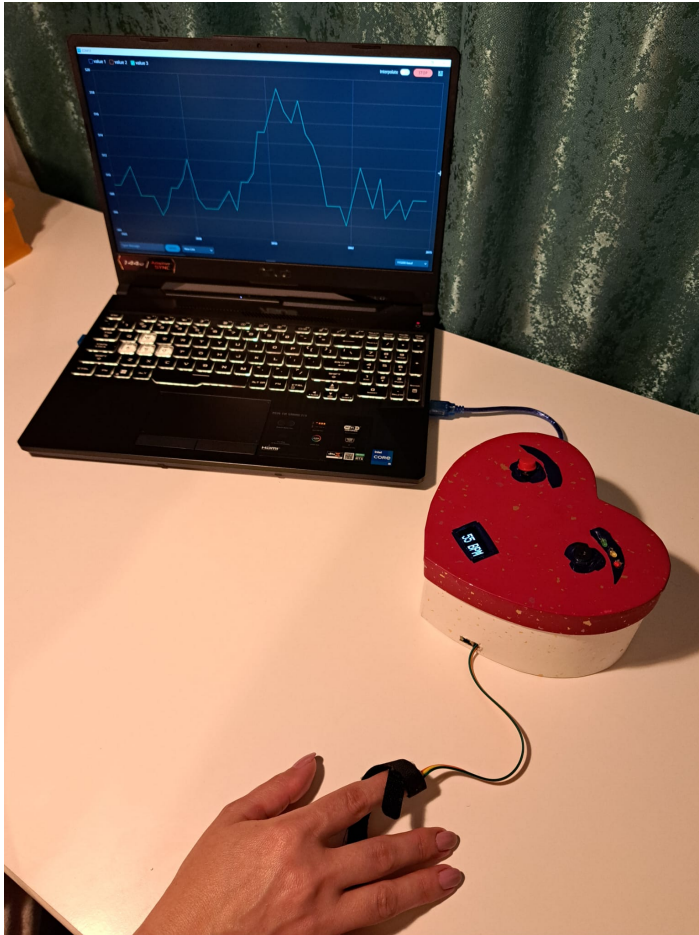
Valori prestabilite și corespondența cu LED-urile

- 60 BPM - 100 BPM : LED Verde
- +/- 10 BPM față de cele normale: LED Galben
- În afara valorilor menționate mai sus: LED Roșu

Pulsul normal sau frecvența cardiacă normală variază la adulți între 60 și 100 de bătăi pe minut atunci când se află într-o stare relaxată.

Rezultate Obținute





- [Aici](#) se gaseste un scurt demo.

Concluzii

Codul prezentat implementează o funcționalitate de bază pentru monitorizarea ritmului cardiac și evaluarea acestuia în funcție de anumite limite prestabilite. Utilizarea sa în scop medical ar necesita îmbunătățiri și validări suplimentare pentru a asigura acuratețea și siguranța datelor colectate și a rezultatelor furnizate. Înainte de a utiliza acest cod într-un scop medical, este important să se efectueze teste și validări riguroase pentru a verifica faptul că valorile măsurate sunt precise și că rezultatele obținute sunt în concordanță cu standardele medicale.

Trebuie menționat faptul că senzorul nu este de cea mai înaltă calitate, acest fapt influențând datele colectate și corectitudinea lor. Totuși, este de apreciat faptul că, dacă persoana careia i se preiau datele se află într-o poziție corectă, nemiscată și relaxată, datele sunt foarte apropiate de cele reale.

Arhiva cu programul: [serbanbianca_pulsometru2023.zip](#)

Jurnal

- 5 Mai: Creare pagină Wiki + documentație
- 19 Mai: Prezentare parte Hardware
- 26 Mai: Prezentare Software + Prezentare finala
- 28 Mai Definitivare Wiki

Bibliografie/Resurse

Resurse Software:

- <https://www.medlife.ro/articole-medicale/valorile-normale-ale-pulsului>
- <https://pulsesensor.com/pages/code-and-guide>
- <https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/>

Resurse Hardware:

- <https://www.instructables.com/Arduino-YL-44-Buzzer-module/>
- <https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/daniel/pulsometru>



Last update: **2023/05/28 21:12**