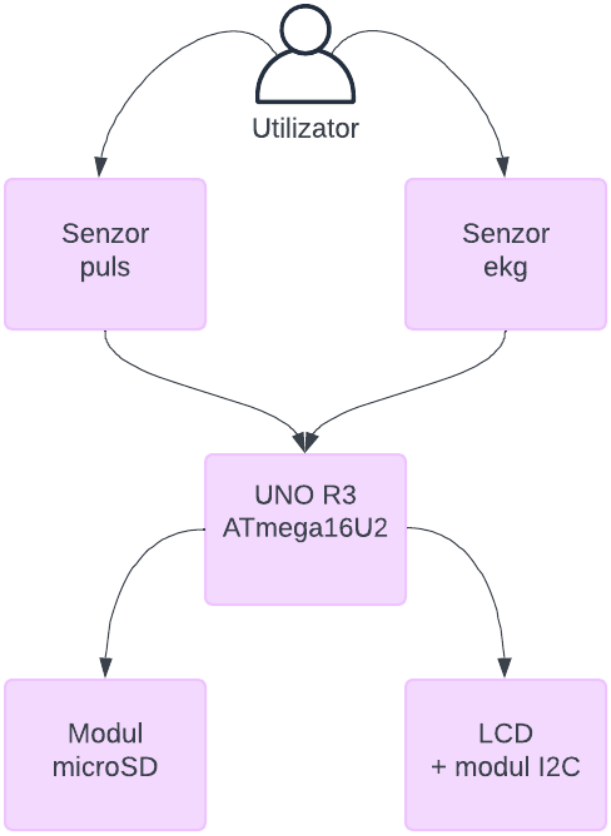
CardioLog

Introducere

CardioLog este un dispozitiv ce are ca scop principal oferirea unei modalități accesibile și exacte de monitorizarea a sănătății cardiace.
Am pornit de la nevoia de a dezvolta un dispozitiv de măsurare a sănătății cardiace ușor de utilizat care să vină în ajutorul persoanelor ce au nevoie de o monitorizare mai intensivă din motive medicale sau care pur și simplu doresc să-și monitorizeze regulat sănătatea cardiovasculară.
Consider CardioLog un dispozitiv util deoarece reprezintă o soluție pentru o nevoie importantă din domeniul sănătății.

Descriere generală

Proiectul constă într-un sistem de monitorizare a sănătății cardiace, care utilizează doi senzori specializați pentru măsurarea pulsului și a activității electro-cardiace. Valorile măsurate sunt afișate în timp real pe un ecran LCD și sunt înregistrate în același timp pe un card SD pentru analiza ulterioară.



Funcționalitate:
* Proiectul utilizează un senzor de puls, care detectează pulsul arterial al utilizatorului prin intermediul unor LED-uri cu infraroșu și a unui fotodetector. Valorile pulsului sunt măsurate în timp real și sunt afișate pe ecranul LCD pentru vizualizare.

* Pentru măsurarea activității electro-cardiace, proiectul utilizează un senzor EKG specializat, ce detectează semnalele electrice generate de inimă și le convertește în date interpretabile. Valorile EKG-ului sunt afișate, de asemenea, pe ecranul LCD în timp real.

* Valorile măsurate de puls și EKG sunt afișate pe ecranul LCD și sunt, în același timp, înregistrate pe un card SD. Această funcționalitate este asigurată de un modul micro SD conectat la plăcuța Arduino prin intermediul interfeței de comunicare SPI.

Interacțiunea cu utilizatorul:
Utilizatorul poate interacționa direct cu senzorii, unde poate citi pe ecranul LCD valorile măsurate și alte informații relevante. De asemenea, utilizatorul poate extrage cardul SD pentru a accesa datele înregistrate și pentru a le analiza ulterior.

Hardware Design

Piese utilizate

- Placa dezvoltare compatibila Arduino, UNO R3 ATmega16U2
- Modul LCD 1602
- Modul I2C pentru display LCD
- Senzor de Puls XD-58C
- Modul senzor ECG, EKG AD8232
- Modul MicroSD
- Breadboard
- Fire Dupont tata-tata, mama-tata

Schemă electrică

n realizat schema electrică în Autodesk Fusion.

Realizarea conexiunilor

Modulul microSD:
Modulul micro SD utilizează interfața SPI pentru comunicare cu placa Arduino. Această interfață

implică utilizarea mai multor pini pentru transmiterea și recepționarea datelor între placa Arduino și modulul micro SD Pinul 11 (MOSI) este utilizat pentru transmiterea datelor de la Arduino către modulul microSD. Pinul 12 (MISO) este utilizat pentru recepționarea datelor de la modulul microSD către Arduino. Pinul 13 (SCK) este utilizat pentru sincronizarea comunicației între Arduino și modulul micro SD Pinul 10 (CS) este utilizat pentru selectarea modulului microSD pentru comunicare cu Arduino.

Senzorul de puls:
Pentru conectarea senzorului de puls am folosit pinul A0, deoarece este unul dintre pinii analogici care pot fi folosiți pentru a citi semnale analogice provenite de la senzor.

Senzorul EKG AD8232:
Pinii analogici A1, si pinii digitali 4 și 3 ai plăcii Arduino au fost folosiți pentru conectarea cu senzorul EKG AD8232. Pinul analogic A1 a fost folosit pentru a citi semnalul de la senzor. Pinii digitali 4 și 3 au fost folosiți pentru a conecta pinii LO- și LO+ ai senzorului. Alegerea acestor pini a fost determinată de necesitatea de a citi semnale analogice și digitale și de a oferi o interfață corespunzătoare între placa Arduino și senzor.

LCD 1602 cu interfata I2C:
Interfața I2C a fost folosită pentru conectarea cu LCD-ul 1602, ceea ce a permis reducerea numărului de pini necesari pentru comunicare. Pinii SDA și SCL ai plăcii Arduino au fost folosiți pentru comunicația I2C cu LCD-ul. Acești pini au fost aleși pentru compatibilitatea cu comunicarea I2C și pentru a facilita interfața cu LCD-ul.

Software Design

Mediul de dezvoltare

Mediul de dezvoltare al programului este Arduino IDE.

Biblioteci utilizate

PulseSensorPlayground.h
Această bibliotecă este specializată pentru a lucra cu senzorul de puls. Am folosit funcții pentru a detecta bătăile inimii și pentru a calcula valoarea BPM-ului (numărul de bătăi pe minut ale inimii).

LiquidCrystal_I2C.h
Am folosit biblioteca pentru a afișa mesaje pe LCD-ul care folosește protocolul I2C.

SPI.h
Am folosit biblioteca SPI.h pentru a inițializa comunicația cu modulul SD. Funcțiile din aceasta bibliotecă includ configurarea pinului de selectare a cipului (chip select), trimiterea și primirea datelor pe magistrala SPI.

SD.h
Biblioteca SD folosește în spate funcții din biblioteca SPI.h pentru a interacționa cu cardul SD. Este folosită în principal la scrierea datelor pe card.

avr/wdt.h
Am folosit biblioteca Watchdog Timer (WDT) pentru a utiliza temporizatorul watchdog, care poate reseta automat microcontrolerul. O folosesc pentru a reseta programul în caz de eroare, dar și pentru a relua funcționalitatea programului.

Funcții implementate

setup()
Funcția setup() inițializează componentele și configurează mediul necesar pentru funcționarea corectă a programului. Inițial, comunicația serială este pornită la o viteză de 9600 baud pentru a permite transmiterea și afișarea mesajelor de debug. Watchdog timer-ul este dezactivat pentru a preveni resetările nedorite în timpul inițializării. LCD-ul este inițializat și se afișează un mesaj de confirmare. Apoi, se încearcă inițializarea cardului SD; dacă aceasta eșuează, se afișează un mesaj de eroare pe LCD și sistemul se resetează. Se configurează și pinul senzorului de puls ca pin de intrare analogic și pragul pentru detectarea bătăilor. De asemenea, pinii pentru detectarea deconectării electrozilor EKG sunt setați ca intrări.

loop()
Funcția loop() gestionează logica principală a programului. Mai întâi se pornește un countdown pentru două minute, oferind utilizatorului timp pentru a se pregăti să utilizeze senzorul de puls. Apoi se masoară pulsul timp de un minut și calculează media valoriilor măsurate pentru BPM. Apoi avem un alt countdown pentru un minut pentru ca utilizatorul să aibă timp să își atașeze electrozii, urmat de măsurarea valorilor EKG timp de 30 de secunde. În final, sistemul este resetat folosind watchdog timer-ul.

countdown(int minutes)
Funcția countdown() afișează un contor invers pe LCD pentru un număr specificat de minute, având ca scop oferirea utilizatorului o indicație vizuală a timpului rămas până la începerea măsurătorilor.

measurePulse(int minutes)
Funcția măsoară pulsul utilizatorului pentru un număr specificat de minute. În timpul acestui interval, senzorul de puls calculează BPM-ul (bătăile inimii pe minut). Dacă este detectată o bătaie, valoarea BPM este afișată în Serial Monitor. Valorile BPM sunt salvate pe cardul SD și se calculează suma și numărul bătăilor detectate. La final, se calculează media BPM, care este apoi afișată pe LCD și în Serial Monitor.

measureEKG()
Funcția măsoară valorile EKG ale utilizatorului timp de 30 de secunde. Aceasta verifică starea electrozilor pentru a detecta deconectările acestora și citește valorile de la senzorul EKG la intervale de timp prestabilite (20 milisecunde). Dacă electrozii sunt conectați corect, valorile EKG sunt citite și afișate în Serial Monitor. Astfel, variațiile semnalului EKG pot fi vizualizate în Serial Plotter-ul din Arduino IDE ca un grafic.

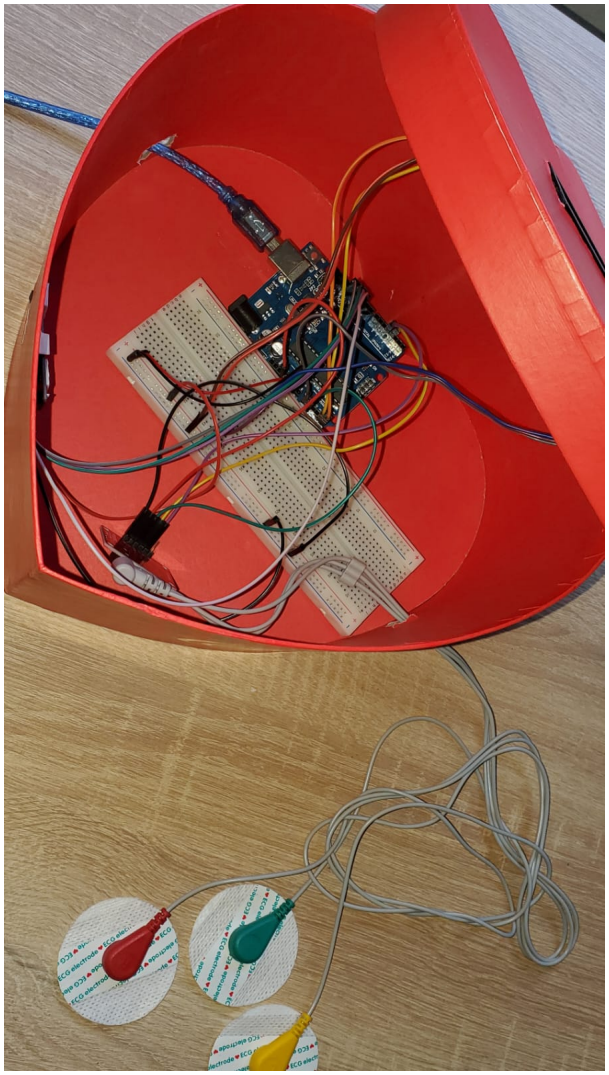
writeData(String dataType, int value, String file)
Scrie un tip de date specificat și o valoare într-un fișier pe cardul SD. Datele sunt scrise în fișier în

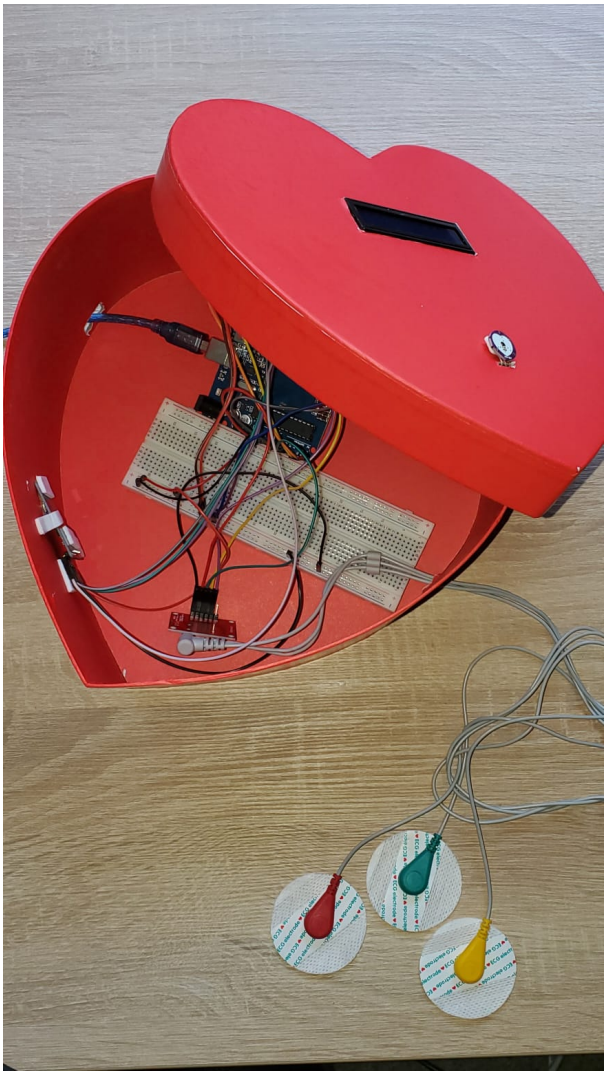
formatul “tip de date: valoare”. Dacă fișierul nu poate fi deschis, se afișează un mesaj de eroare în Serial Monitor.

resetSystem()
Funcția resetează sistemul folosind watchdog timer-ul. Aceasta afișează un mesaj de resetare pe LCD și apoi activează watchdog timer-ul cu un timeout de 1 secundă. Bucla infinită care urmează asigură că sistemul va fi resetat de watchdog timer, oferind astfel o metodă sigură de a reporni dispozitivul în caz de eroare sau la finalizarea măsurătorilor.

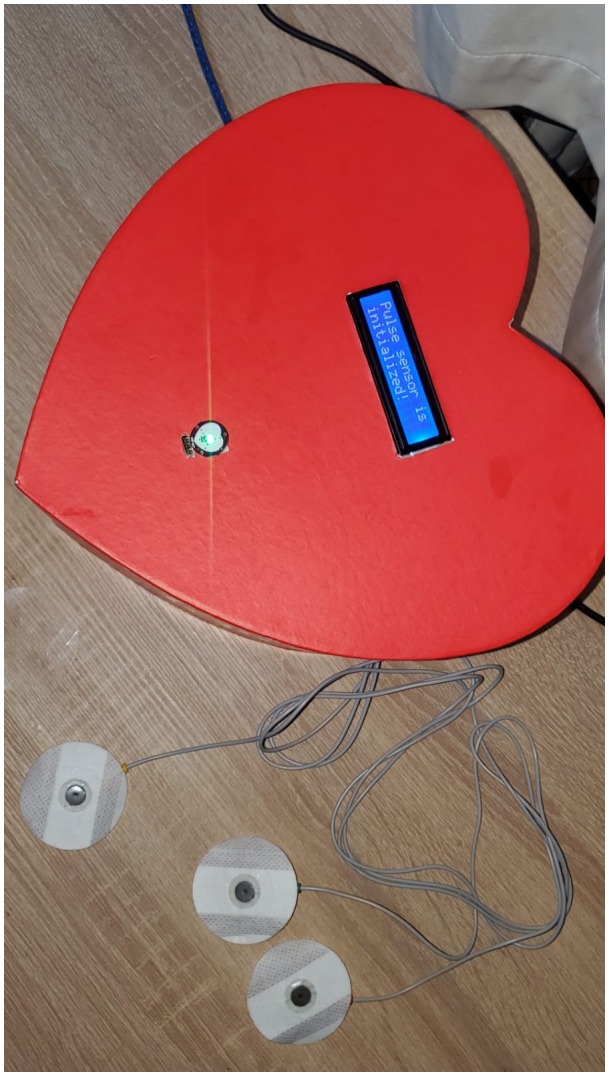
Rezultate Obţinute

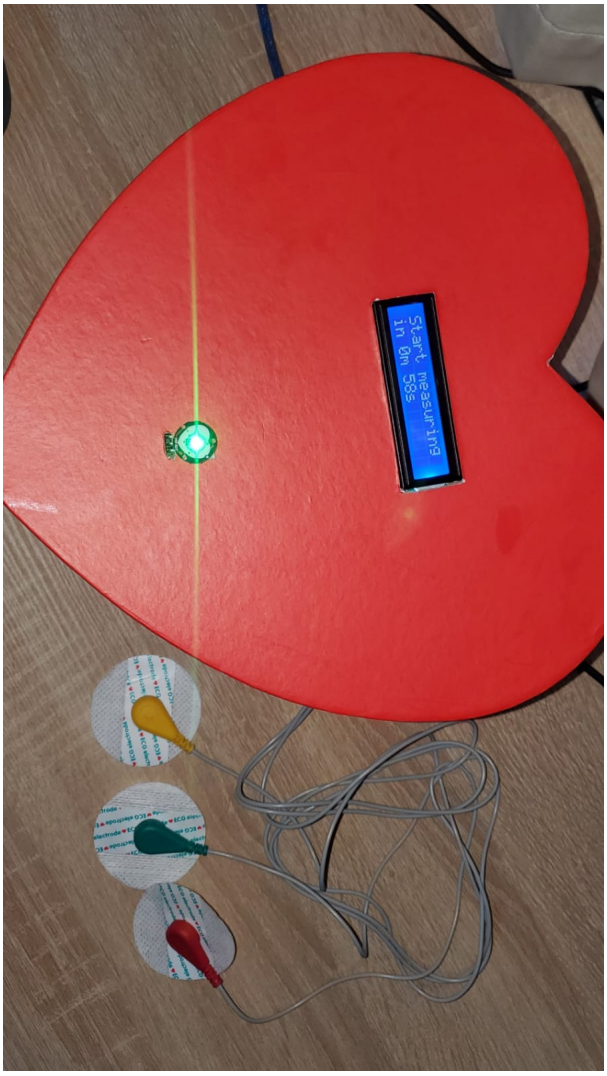
Conectarea componentelor

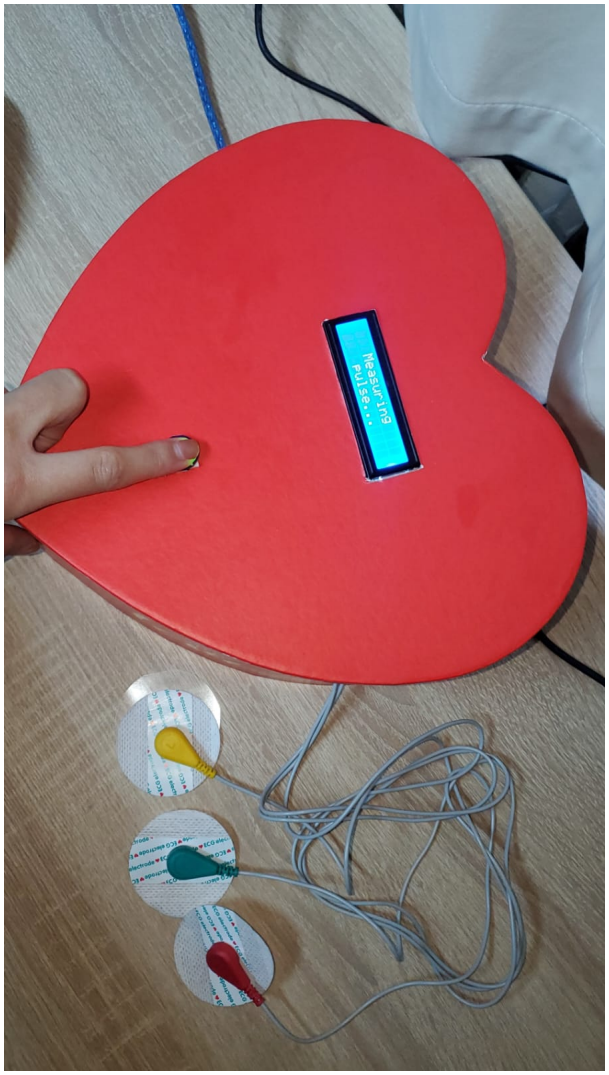


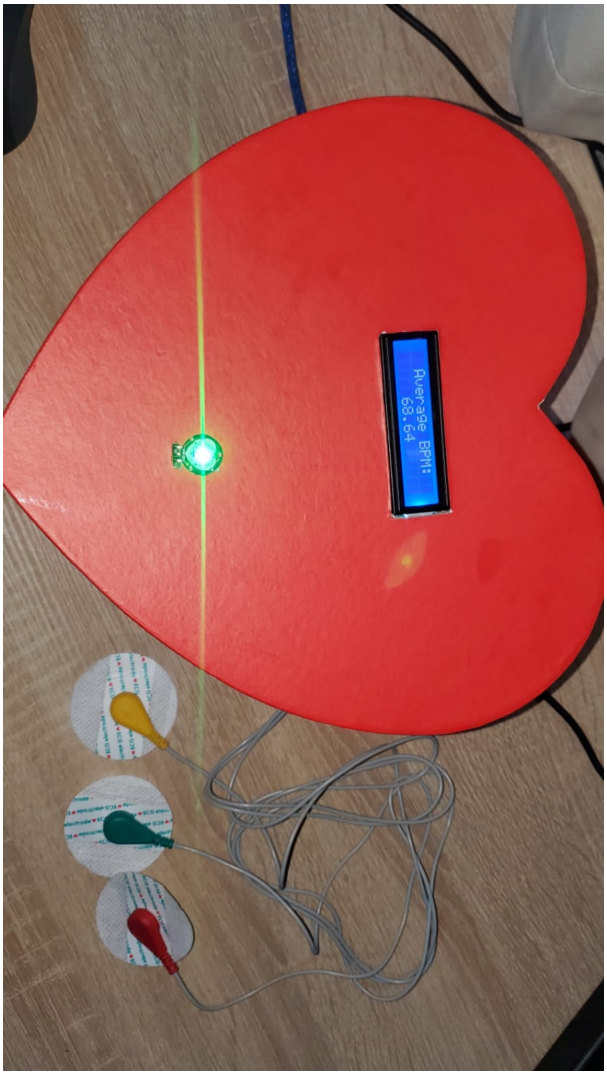


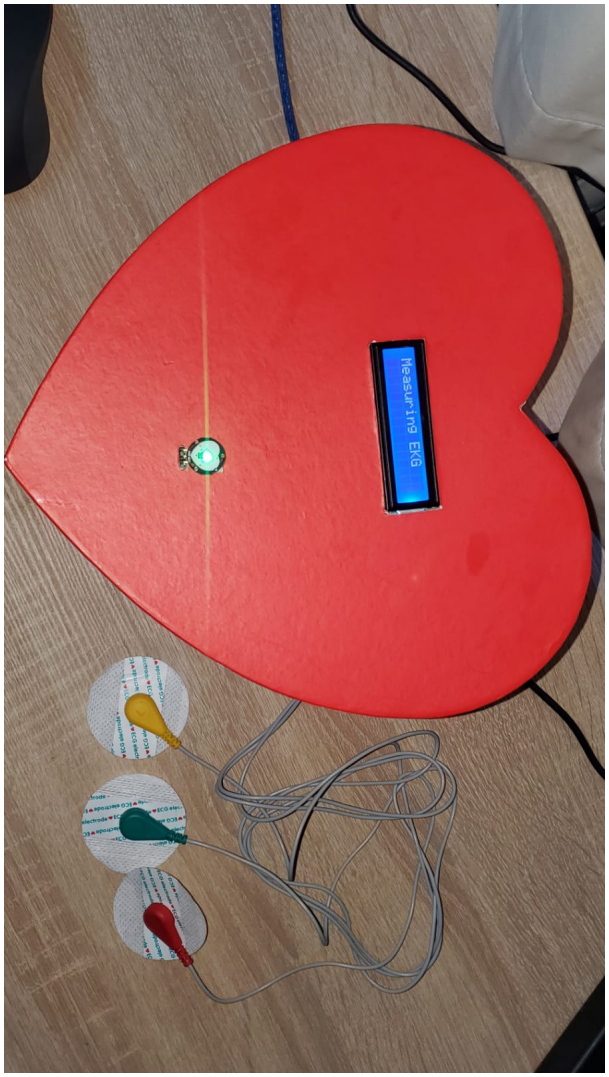
Funcționarea programului


Pulse sensor
initialized!

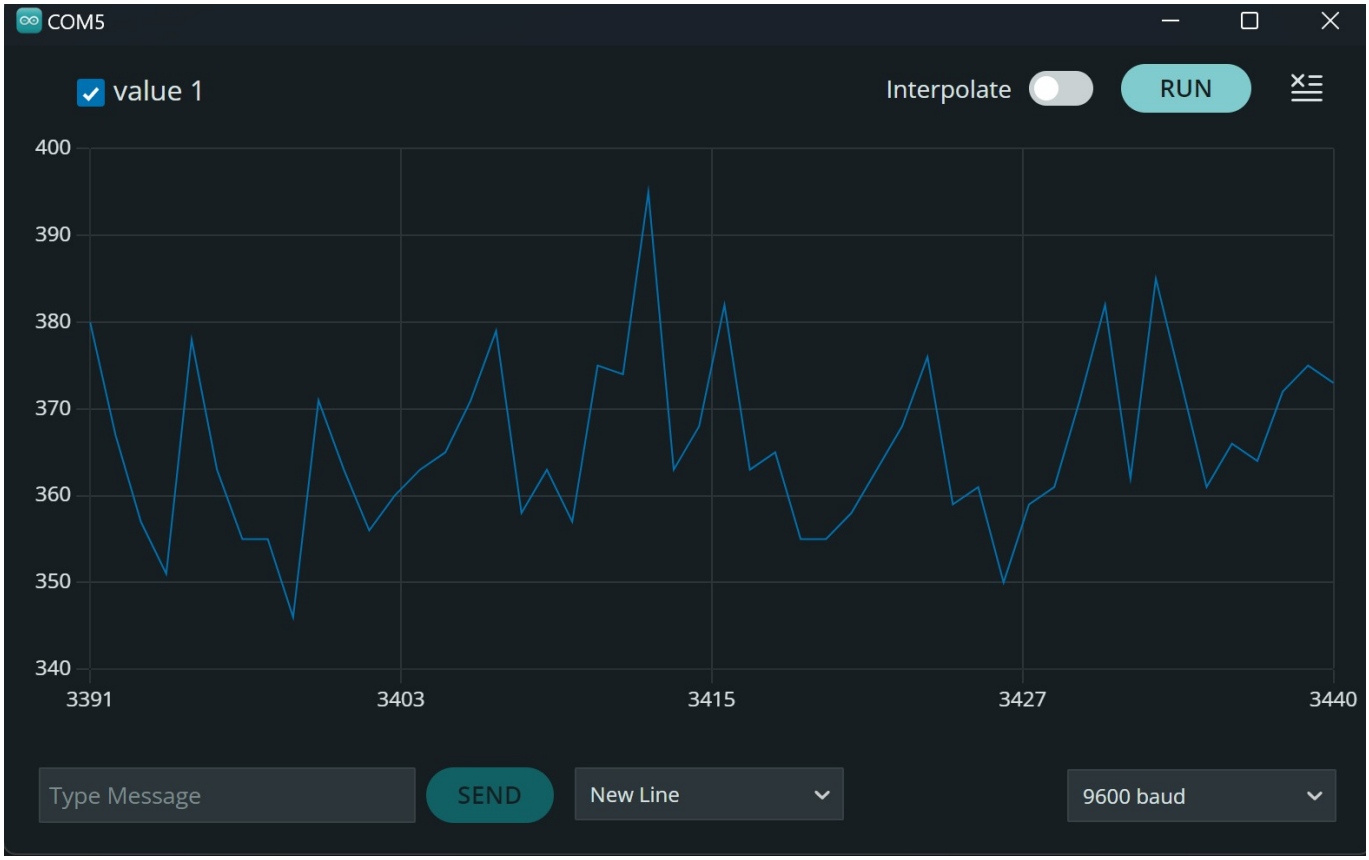

Start measuring
in 0m
50s


Measuring
Pulse...


Average BPM:
68.64



Eletrocardiogramă obținută



Date scrise pe cardul SD

Pulse: 61
Pulse: 63
Pulse: 62
Pulse: 61
Pulse: 62
Pulse: 63
Pulse: 63
Pulse: 64
Pulse: 64
Pulse: 65
Pulse: 65
Pulse: 64
Pulse: 66
Pulse: 66

Demonstrație a funcționării

Concluzii

A fost un proiect interesant deoarece a fost pentru prima data cand am realizat un proiect hardware. Deoarece intotdeauna m-a preocupat sanatatea mea, dar si subiectele medicale, alegerea acestei teme de proiect m-a facut sa-mi doresc sa descopar cat mai multe despre utilizarea acestor senzori de puls si EKG. Senzorul de puls mi-a cauzat cateva neplaceri, deoarece am incercat diferite implementari pana am constatat ca acesta nu masoara cu exactitate pulsul si ca la mentinerea nemiscata a degetului pe senzor, acesta nu detecteaza bataile reale ale inimii. Pentru a realiza demo-ul am simulat aceste batai prin atingeri succesive ale senzorului. In rest, functionalitatea obtinuta este cea dorita.

Download

dragusin-daniela-alexandra-proiect-pm-2024.zip

Bibliografie/Resurse

Resurse software:

- Simbol și footprint placă de dezvoltare
- Simbol și footprint modul EKG AD8232
- Simbol și footprint senzor de puls
- Simbol și footprint ecran LCD
- Tutorial senzor de puls
- Tutorial senzor EKG
- Biblioteca PulseSensorPlayground.h
- Biblioteca LiquidCrystal_I2C.h
- Biblioteca SD.h

Resurse hardware:

- Placă de dezvoltare
- Ecran LCD
- Modul I2C
- Senzor puls
- Modul senzor EKG
- Modul MicroSD

Export to PDF