

# Fluture Robert - Heart Rate Monitoring Device

## Introducere

Dispozitivul se bazeaza pe un senzor de monitorizare al batailor inimii, cu circuit inclus pentru a reduce noise.

- Monitorizeaza heart rate-ul utilizatorului si ofera feedback vizual in baza datelor intrepritate
- Ajuta la informarea despre si imbunatatirea conditiei de sanatate intr-o maniera interactiva si usor de interpretat
- O matrice de led-uri care interpreteaza si afiseaza pulsul (electrical baseline). Dar e nevoie de foarte multe led uri sau un mini monitor, asa ca am ales sa fac ceva similar dar cu feedback vizual diferit.

## Descriere generală



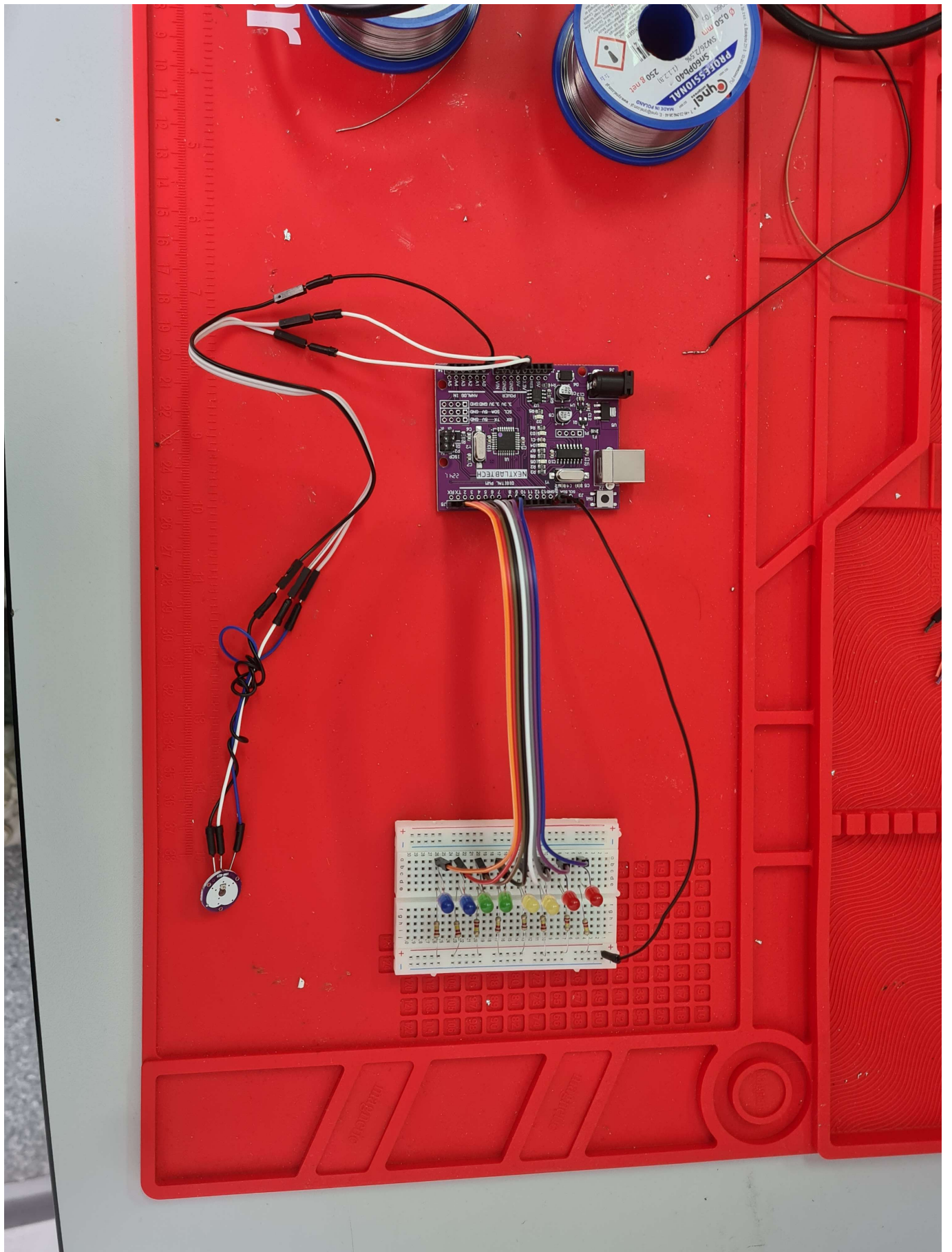
## Hardware Design

Lista Componente:

- Arduino
- Microcontroller to read sensor data and control LEDs.
- Heart Rate Sensor (HW-827)
- Sensor to measure the heart rate.
- 8 LEDs
- LEDs to display heart rate intensity in a loading bar format.
- 220-ohm Resistors (8)
- To limit current through the LEDs.
- Breadboard
- To make the circuit connections.
- Jumper Wires
- To connect components on the breadboard and to the Arduino.
- Power Source

- USB cable for Arduino power or a battery pack.





# Software Design

main.cpp

```
#include "I2Cdev.h"
#include <LiquidCrystal_I2C.h>

MPU6050 mpu;

const int heartRatePin = A0;
const int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9};
volatile int heartRate = 0;
int intensity = 0;
unsigned long previousMillis = 0;
const long interval = 500; // Interval for updating the LED display

void setup() {
  Serial.begin(9600);
  for (int i = 0; i < 8; i++) {
    pinMode(ledPins[i], OUTPUT);
  }

  // Configure timer interrupt
  Timer1.initialize(1000000); // Set timer to 1 second
  Timer1.attachInterrupt(timerISR); // Attach timer interrupt service
  routine

  // Enable ADC interrupt
  ADCSRA |= (1 << ADIE);
  sei(); // Enable global interrupts
}

ISR(ADC_vect) {
  heartRate = ADC; // Read the ADC value
  intensity = map(heartRate, 0, 1023, 0, 255); // Map heart rate to
  LED intensity
}

void timerISR() {
  // Trigger ADC conversion
  ADCSRA |= (1 << ADSC);
}

void loop() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    // Update LED display
```

```
int numLeds = map(heartRate, 0, 1023, 0, 8);
for (int i = 0; i < 8; i++) {
  if (i < numLeds) {
    analogWrite(ledPins[i], intensity);
  } else {
    digitalWrite(ledPins[i], LOW);
  }
}
}
```

Înteruperea ADC: înteruperea ADC (Convertor analog-digital) este activată pentru a gestiona citirile ritmului cardiac. Timer Interrupt: Timer1 este configurat să declanșeze o înterupere la fiecare secundă, pornind o conversie ADC. Manipularea intensității:

Intensitatea LED-urilor este mapată din citirile senzorului de ritm cardiac folosind funcția map(). Aceasta controlează luminozitatea LED-urilor folosind PWM. Temporizator de actualizare LED:

Funcția millis() este utilizată pentru a actualiza periodic afișajul LED la fiecare 500 de milisecunde.

## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

## Concluzii

## Download

[heartrate.zip](#)

## 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/prj2024/ddosaru/robert.fluture>



Last update: **2024/05/27 08:56**