

# Heart Beat Care

Autor: [Antonia NICOLAESCU](#)

## Introducere

Proiectul urmareste implementarea unui aparat de masurare a ritmului cardiac care sa afiseze pe un ecran OLED graficul pulsului si valoarea acestuia. Aparatul va detecta si semnaliza tahicardia.

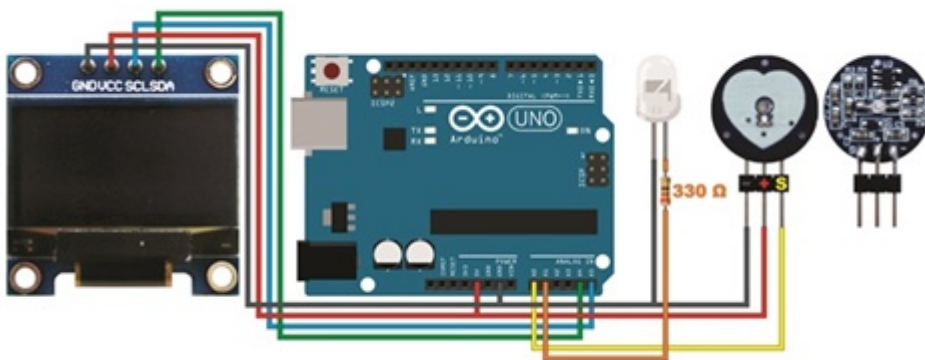
## Descriere generală

## Schema Bloc



## Hardware Design

## Schema Electrica



## Lista de piese necesare

- Arduino UNO
- Breadboard
- Senzor puls
- Led
- Ecran OLED
- Rezistori
- Fire

## Software Design

Pentru implementarea software a proiectului am folosit Arduino IDE.

Librarii folosite:

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

**Adafruit GFX Library** - este necesară pentru ca librăria pentru displayul OLED să funcționeze.

**Adafruit SSD1306** - librărie pentru displayul OLED

Variabilele:

```
//-----Declarare variabile
unsigned long previousMillisGetHR = 0; //--> stocheaza cand Millis a fost
updatata (pentru a calcula Heartbeat)
unsigned long previousMillisResultHR = 0; //--> stocheaza cand Millis a fost
updatata (pentru a calcula BPM)

const long intervalGetHR = 20; //--> Interval pentru a citi ritmul cardiac =
10 ms
const long intervalResultHR = 10000; //Intervalul de citire a rezultatului
final = 10 secunde

int PulseSensorSignal;
const int PulseSensorHRWire = 0; //--> Senzorul de puls este conectat la
pinul analog 0.
const int LED_A1 = A1; //--> LEDul este conectat la pinul A1.
int UpperThreshold = 330; //--> Determina ce semnal sa fie contorizat ca
bataie si stabileste sensibilitatea senzorului.
int LowerThreshold = 300;

int cntHB = 0; //--> Numara bataile inimii.
```

```

boolean ThresholdStat = true; //--> True daca e contorizat ca bataie, false
daca nu.
int BPMval = 0; //--> Stocheaza rezultatul.

int x=0; //--> Axa x pe OLED
int y=0; //--> Axa y pe OLED
int lastx=0;
int lasty=0;

```

Functia de citire a batailor inimii:

```

//Citeste bataile inimii si calculeaza valoarea BPM-ului
void GetHeartRate() {
    //----- Procesul de citire a batailor
    inimii
    unsigned long currentMillisGetHR = millis();

    if (currentMillisGetHR - previousMillisGetHR >= intervalGetHR) {
        previousMillisGetHR = currentMillisGetHR;

        PulseSensorSignal = analogRead(PulseSensorHRWire);

        if (PulseSensorSignal > UpperThreshold && ThresholdStat == true) {
            cntHB++;
            ThresholdStat = false;
            digitalWrite(LED_A1,HIGH);
        }

        if (PulseSensorSignal < LowerThreshold) {
            ThresholdStat = true;
            digitalWrite(LED_A1,LOW);
        }

        DrawGraph();
    }
}

```

Functia care afiseaza graficul ritmului cardiac:

```

//-----
-----Afisarea graficului.
void DrawGraph() {
    //-----Conditie ca graficul sa fie
    resetat atunci cand umple ecranul OLED.
    if (x > 127) {
        display.fillRect(0, 0, 128, 42, BLACK);
        x = 0;
        lastx = 0;
    }
}

```

```
}  
//-----  
int ySignal = PulseSensorSignal;  
  
if (ySignal > 850) ySignal = 850;  
if (ySignal < 350) ySignal = 350;  
  
int ySignalMap = map(ySignal, 350, 850, 0, 40);  
  
y = 40 - ySignalMap;  
//-----  
  
if ( BPMval < 120 ) {  
  display.writeLine(lastx,lasty,x,y,WHITE);  
  display.display();  
}  
//-----  
  
lastx = x;  
lasty = y;  
  
x++;  
}
```

Am folosit Serial Plotter din Arduino IDE pentru a testa senzorul de puls si pentru a stabili senzibilitatea acestuia. Acest pas a fost foarte util in setarea valorilor pentru UpperThreshold si LowerThreshold.



De asemenea, am folosit acest site pentru a transforma imaginile in secvente: <https://diyusthad.com>. Am adaugat imagini reprezentative pentru logo-ul proiectului, care sa apara de fiecare data cand este rulat, si imagine pentru semnalizarea tahicardiei.

**Logo-ul:**



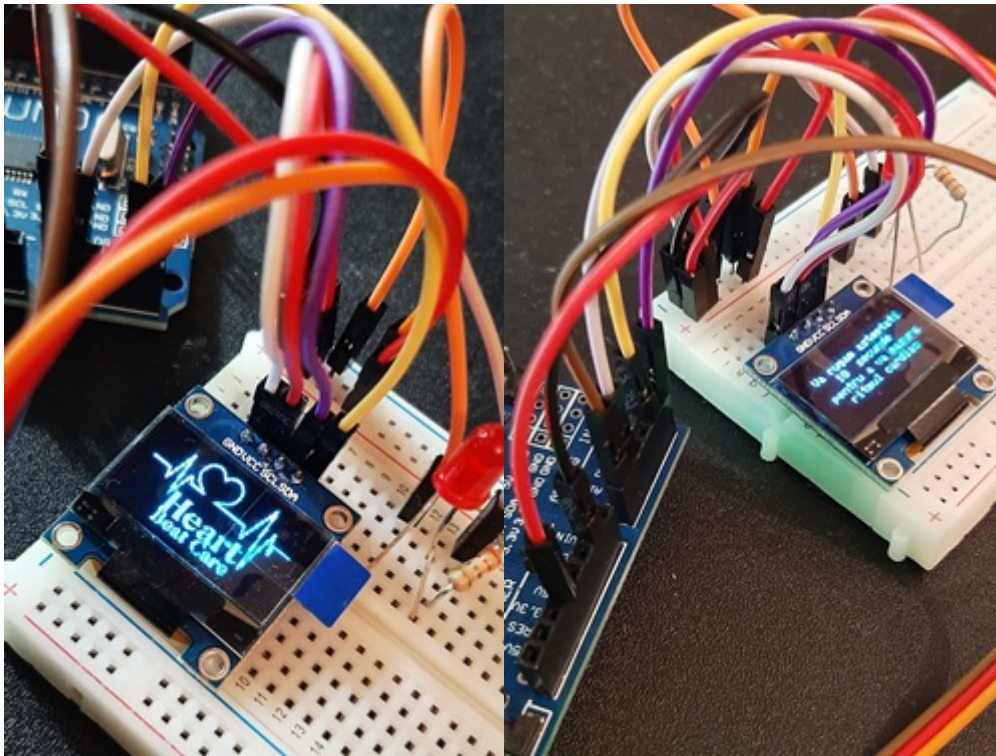
**Imaginea care apare atunci cand este detectata tahicardia:**



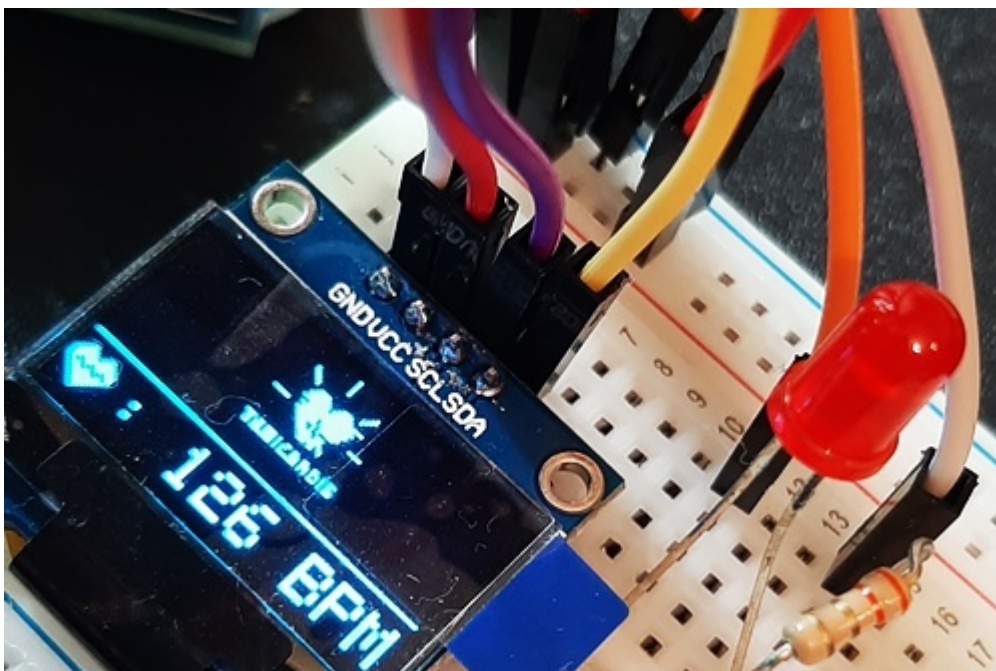
## Rezultate Obținute

**Video de prezentare:** <https://www.youtube.com/watch?v=R59K-Fxla98>

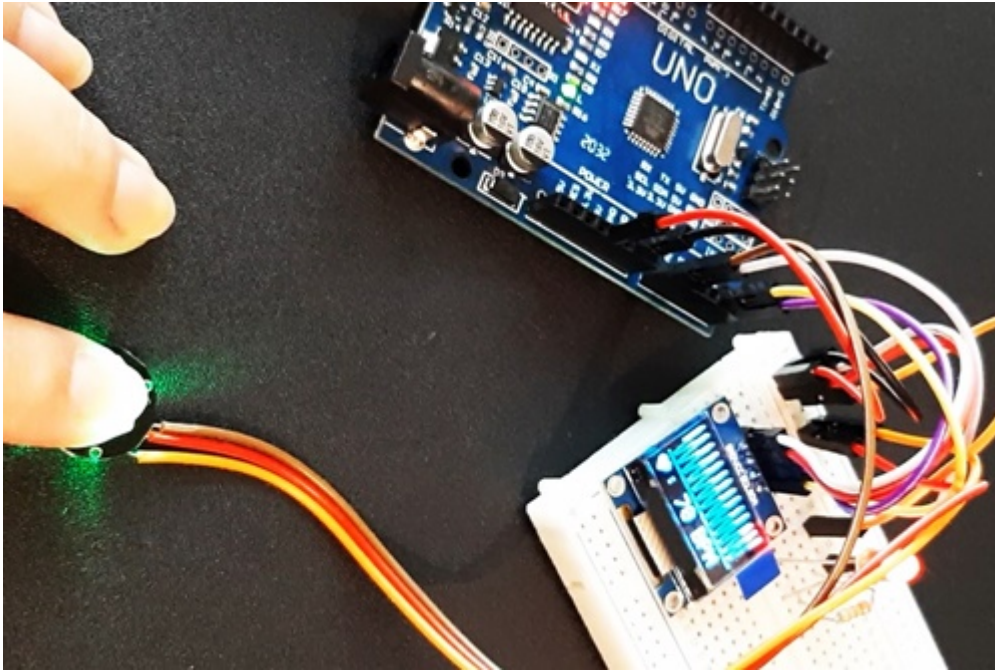
**Afisajul de start: logo-ul si mesajul cu indicatiile de utilizare**



**Afisajul in caz de tahicardie**



**Functionalitate si afisaj:**



## Concluzii

Rezultatul proiectului este asemanator cu cel propus initial (singura modificare este inlocuirea buzzer-ului cu un led ce semnalizeaza bataile inimii). Cel mai dificil de realizat au fost stabilirea sensibilitatii senzoului si generarea graficului (tip EKG) pe ecranul OLED (probleme pe care le-am depasit cu ajutorul link-urilor din Bibliografie).

## Jurnal

- 27 aprilie → Alegere tema proiect
- 28 aprilie - 10 mai → Achiziționare componente
- 11 mai - 15 mai → Testare componente
- 15 mai - 20 mai → Realizare parte software
- 21 mai - 25 mai → Imbunatatire si optimizare
- 25 mai - 30 mai → Finalizare proiect

## Download

Documentatia proiectului: [Heart-Beat-Care](#)

Arhiva cod: [heartbeatcare.zip](#)

## Bibliografie

**Realizare schema electrica:** <https://www.circuito.io>

**Testare senzor puls:** <https://pulsesensor.com/pages/code-and-guide>

**Afisare semnale pe ecran OLED:**

<https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/heartbeatcare>



Last update: **2021/06/03 10:09**