



Universitatea POLITEHNICA Bucuresti
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

PROIECT 2 – Electronică Programabilă

Monitorizarea și măsurarea ritmului cardiac

Titularii disciplinei

Prof. Dr. Ing. Corneliu Burileanu

Studenți

Stancovici Marian

Preoteasa Rareș Marin

Neamțu Rareș Damian

Coordonatorii proiectului

Ing. Ana-Antonia Neacșu

București

2022-2023

CUPRINS

I. INTRODUCERE.....	3
II.RESURSE HARDWARE.....	4
III. RESURSE SOFTWARE.....	10
IV. IMPLEMENTARE HARDWARE.....	12
V. IMPLEMENTARE SOFTWARE.....	17
VI. CONCLUZII.....	24
VII. BIBLIOGRAFIE	25

I. INTRODUCERE

Electronica a avut un impact semnificativ asupra domeniului medical, aducând inovații și îmbunătățiri importante în diagnosticare, tratament și monitorizarea pacienților. Una dintre cele mai importante aplicații ale electronicii în medicină este utilizarea dispozitivelor electronice și a sistemelor informatice pentru a măsura semnalele electrice generate de inimă și pentru a interpreta aceste semnale într-un mod util în diagnosticarea și tratamentul bolilor cardiovasculare.

Proiectul nostru constă în realizarea unui sistem de monitorizare a ritmului cardiac utilizând un senzor de puls. Obiectivul principal al sistemului măsurarea numărului de bătăi pe minut ale inimii și afișarea acestui rezultat pe un ecran LCD, oferind utilizatorului informații în timp real despre ritmul său cardiac.

Monitorizarea va fi realizată prin intermediul a două LED-uri, unul roșu care se va aprinde în momentul în care numărul bătăilor este prea mare sau prea mic, în rest va fi aprins un LED albastru.

Mai mult, sistemul este dotat cu un modul GSM, care permite trimiterea unui mesaj pe telefonul mobil în cazul în care LED-ul roșu se aprinde, pentru a trage încă un semnal de alarmă în cazul în care numărul bătăilor este prea mare sau prea mic.

II. RESURSE HARDWARE

1. Placa de dezvoltare Intel Galileo

Galileo este o plăcuță de dezvoltare cu microcontroler, bazată pe procesorul Intel® Quark SoC X1000, un chip pe 32-bit din clasa Intel Pentium. Este prima plăcuță bazată pe arhitectura Intel, proiectată să fie compatibilă cu pinii shield-urilor Arduino pentru Uno R3, atât din punct de vedere hardware cât și din punct de vedere software. Pinii digitali de la 0 la 13 (și pinii adiacenți AREF și GND), intrările analogice de la 0 la 5, conectorul de alimentare, conectorul ICSP și pinii portului UART (0 și 1), toate acestea se află în aceleași poziții ca și la Arduino Uno R3. Acest lucru se mai numește și modul Arduino 1.0 de așezare a pinilor.

Galileo este proiectat să suporte shield-uri care operează la 3.3V sau la 5V. Tensiunea de funcționare de bază pentru Galileo este de 3.3V. Totuși, un jumper de pe plăcuță activează conversia la 5V pe pinii I/O. Acest lucru oferă suport pentru shield-urile Uno de 5V și este comportamentul implicit. Prin schimbarea poziției jumperului, conversia tensiunii poate fi dezactivată pentru a oferi funcționarea la 3.3V a pinilor de I/O.

Desigur, plăcuța Galileo este compatibilă, de asemenea, și din punct de vedere software cu Mediul de Dezvoltare Software Arduino (IDE), lucru care face posibilă utilizarea și introducerea foarte rapide. În plus față de compatibilitatea hardware și software a lui Arduino, plăcuța Galileo are mai multe porturi I/O și caracteristici standard pentru PC pentru a extinde utilizarea și capacitățile native dincolo de ecosistemul shield-ului Arduino. Un slot mini-PCI Express de dimensiuni complete, port Ethernet de 100 Mb, slot Micro-SD, port serial RS-232, port USB Host, port USB Client, și flash NOR de 8MByte sunt în mod standard pe plăcuță.

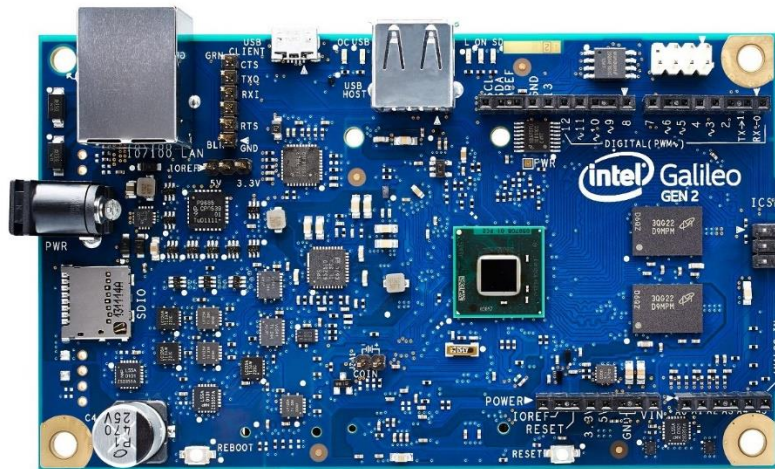


Figura 2.1 Placa Intel Galileo Gen 2

Galileo este proiectat să suporte shield-uri care operează fie la 3.3V, fie la 5V. Un jumper de pe plăcuță permite conversia tensiunii de la 3.3V la 5V pe pinii I/O

- Există 14 pini digitali de intrare/ieșire, dintre care 6 pot fi folosiți ca ieșiri PWM(Pulse Width Modulation). Toți pinii digitali pot fi utilizați atât ca intrări, cât și ca ieșiri, folosind funcțiile `pinMode()`, `digitalWrite()` and `digitalRead()`. Aceștia operează la 3.3V sau la 5V. Fiecare pin poate oferi un maxim de 10mA sau poate primi un maxim de 25mA și are un rezistor de tracțiune intern (deconectat în mod implicit) de 5.6k Ω până la 10k Ω .
- Există 6 intrări analogice, prin intermediul unui convertor A/D AD7298 (A0 –A5). Fiecare dintre cele 6 intrări analogice, marcate de la A0 până la A5, oferă o rezoluție de 12 biți. (4096 valori diferite). În mod implicit, acestea măsoară de la masă la 5V.
- Magistrală I2C, TWI: pinii SDA și SCL care sunt localizați lângă pin-ul AREF. TWI: pinul A4 sau SDA și pinul A5 sau SCL. Suportă comunicație TWI folosind biblioteca Wire.
- SPI: Implicit la 4 MHz pentru a suporta shield-uri Arduino Uno. Programabilă până la 25Mhz.
- UART (port serial): Port de viteză UART programabil (pinii digitali 0 (RX) și 1 (TX))
- ICSP (SPI): Un conector serial de programare „in-circuit” cu 6 pini, localizat în mod corespunzător pentru conectarea la shield-urile existente. Acești pini suportă comunicarea SPI folosind biblioteca SPI.
- VIN: Tensiunea de intrare în plăcuța Galileo atunci când se folosește o sursă externă de putere (spre deosebire de 5V de la sursa de alimentare constantă conectată la mufa de alimentare). Se poate alimenta cu tensiune prin acest pin, sau, dacă se alimentează cu tensiune prin mufa de alimentare, poate fi accesată prin acest pin. Atenție: Tensiunea aplicată acestui pin trebuie să fie de la o sursă de 5V, altfel ar putea deteriora placa Galileo sau ar cauza o funcționare incorectă.
- Pin de ieșire de 5V: Acest pin scoate 5V de la sursa externă sau de la conectorul USB. Debitul maxim al curentului la shield este: 800mA.
- Pin de ieșire de 3.3V: O alimentare de 3.3 volți generată de regulatorul încorporat. Debitul maxim al curentului la ecran este: 800mA.
- GND: Pini de masă
- IOREF: Pin-ul IOREF de pe Galileo permite unui shield atașat cu o configurație corectă să se adapteze la tensiunea furnizată de placă. Tensiunea pin-ului IOREF este controlată de un jumper de pe placă, adică, un jumper de selecție de pe placă este folosit pentru a selecta între funcționarea shield-ului la 3.3V și la 5V.
- Buton/pin de RESET: Aduce linia pe LOW pentru a reseta sketch-ul. Folosit, în mod obișnuit, pentru a adăuga un buton de resetare shield-urilor care îl blochează pe cel de pe placă.
- AREF nu este folosit pe Galileo. Furnizarea unei tensiuni de referință externe pentru intrările analogice nu este suportată. Pentru Galileo, nu este posibil să se modifice capătul superior al domeniului de intrare analogic utilizând pinul AREF și funcția `analogReference()`.

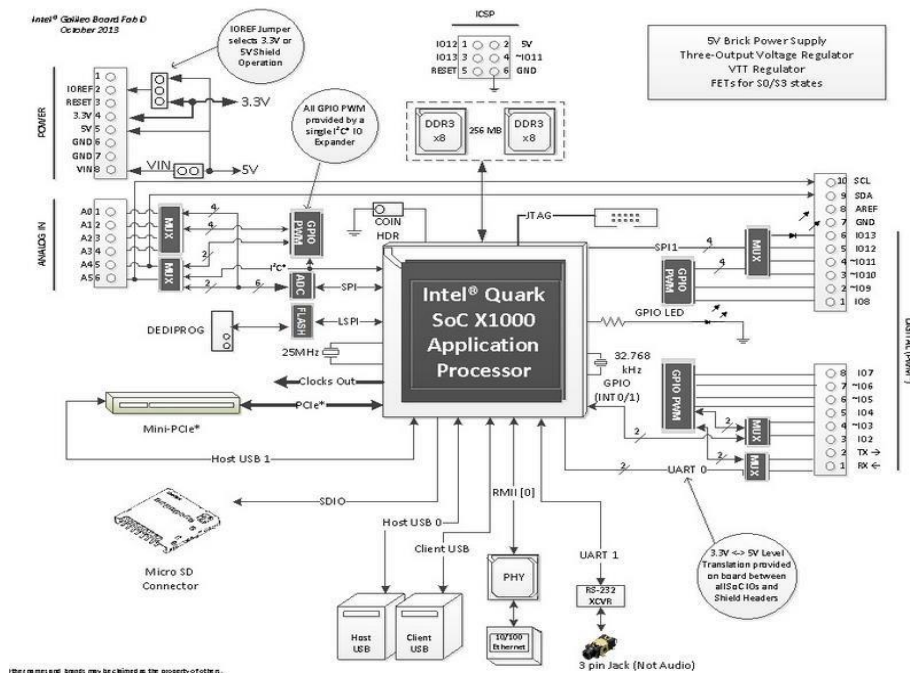


Figura 2.2 Schema logica Intel Galileo

2. Modul LCD 1602 cu Backlight Albastru de 3.3 V

- Tensiune de alimentare: 3.3 V;
- Dimensiuni: 80 mm x 36 mm x 12 mm.

LCD-ul are contrast ajustabil, culoarea caracterelor este alba, iar backlight-ul albastru.



Figura 2.3 Modul LCD 1602 cu Backlight Albastru de 3.3 V

3. Senzor de Puls XD-58C

- Tensiunea de alimentare: 3.3 V și 5 V;
- Curent consumat: ~ 4 mA @ 5 V;
- Diametru: ~16 mm;
- Grosime medie: ~3 mm;

Senzorul de puls XD-58C este un senzor plug-and-play este proiectat să funcționeze împreună cu o placă de dezvoltare echipată cu cel puțin un pin analog. Această componentă este utilă pentru a colecta date despre bătăile inimii în diferite situații, cum ar fi în timpul exercițiilor fizice.

Modulul are consum redus, doar 4mA alimentat la 5V. Poate fi conectat foarte ușor la Arduino.



Figura 2.4 Senzor de Puls XD-58C

4. A6 GSM / GPRS Module

Modulul A6 GSM/GPRS este un modem GSM în miniatură, care poate fi integrat într-un număr mare de proiecte. În centrul modulului se află cipul celular A6 GSM ce comunică cu un microcontroler prin UART și acceptă rata de transmisie de la 1200 bps la 115200 bps cu detectare automată.

Modulul are nevoie de o antenă externă pentru orice fel de comunicații de voce sau date, precum și unele comenzi SIM. Deci, de obicei vine cu o antenă mică de rață cu câștig de 2 dB și impedanță de 50Ω, care oferă o acoperire excelentă chiar dacă proiectul este în interior.

Există un buton de pornire furnizat pentru a porni/opri modulul manual. Starea modulului este indicată de un LED în partea dreaptă sus a modulului.

Deși modulul poate funcționa la 5V, tensiunea de funcționare a cipului este de la 3,3V la 4,2V. Pentru a menține tensiunea de alimentare în siguranță la 4,1 V, modulul vine cu un regulator de comutare de înaltă frecvență MP1584 de la Monolithic Power Systems - capabil să gestioneze curenți de sarcină de până la 3A. Modulul poate fi alimentat și printr-un conector micro USB.



Figura 2.5 A6 GSM / GPRS Module

5. Kit Breadboard HQ830 cu Fire și Sursă

- 1 Breadboard HQ (830 Puncte);
- 1 Sursă de alimentare pentru breadboard cu ieșire de 3.3V sau 5V;
- 1 Set fire pentru breadboard.

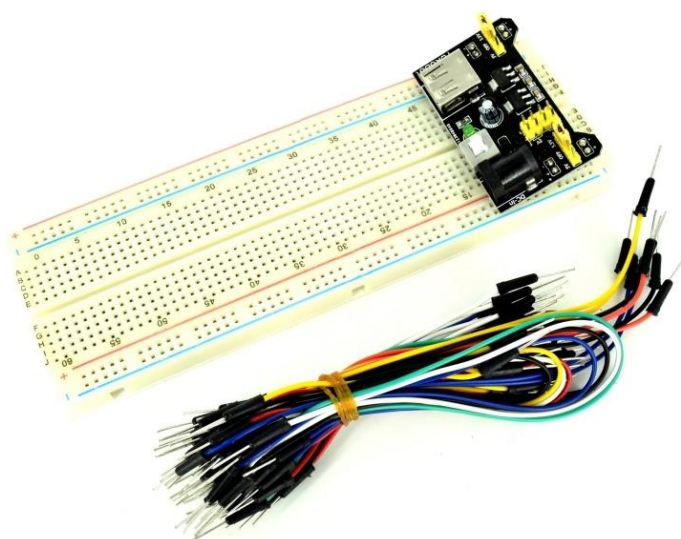


Figura 2.6 Breadboard HQ830 cu Fire și Sursă

6. Arduino Uno

Arduino Uno este o placă de dezvoltare bazată pe microcontrollerul ATmega328. Are 14 pini de intrare/ieșire (dintre care 6 pot fi folosiți ca ieșiri PWM), 6 intrări analog, un oscilator de 16MHz, o conexiune USB, mufă de alimentare, și un buton de reset.

Poate fi alimentat direct de la calculator, de la portul USB, prin intermediul unei baterii de 9V sau a unui alimentator de 9V.



Figura 2.7 Arduino Uno

7. LED-uri

- Roșu
- Albastru

8. Potențiometru de 10 kOhm

9. Rezistori

- 1 x 220 Ohm
- 1 x 1 kOhm
- 1 x 1.1 kOhm

III. RESURSE SOFTWARE

Arduino Software

Plăcuța Galileo este compatibilă cu Arduino Software Development Environment (IDE), lucru care face posibilă utilizarea foarte ușoară și rapidă.

Arduino Integrated Development Environment sau Arduino Software (IDE) – conține un editor de text pentru a scrie codul, o zonă de mesaje, o consolă text, o bară de instrumente cu butoane pentru funcții obișnuite și o serie de meniuri. Acesta se conectează la hardware-ul Arduino și Genuino pentru a încărca programe și pentru a comunica cu acestea.

Programele scrise folosind Arduino Software (IDE) sunt numite sketch-uri (schite). Acestea sunt scrise în editorul de text și sunt salvate cu extensia .ino. Zona de mesaje oferă feedback în timp ce se salvează și se exportă și, de asemenea, afișează și erori. Consola afișează textul de ieșire dat de Arduino Software (IDE), inclusiv mesaje de eroare. Colțul din dreapta jos al ferestrei afișează placa și portul serial configurate. Butoanele barei de instrumente permit verificarea și încărcarea în program, crearea, deschiderea și salvarea sketch-uri și deschiderea monitorul serial.

Arduino IDE suportă limbajele de programare C și C++ folosind reguli speciale de organizare a codului. O primă deosebire cu un program propriu-zis C/C++ este ca un sketch tipic Arduino este compus din două funcții:

- `setup()`: o funcție care este rulată o singură dată la începutul programului, când se inițializează setările.
- `loop()`: o funcție apelată în mod repetat până la oprirea alimentării cu energie a plăcuței

De asemenea, sunt puse la dispoziția utilizatorului o serie de funcții care fac gestionarea resurselor hardware foarte simplă și intuitivă:

- `pinMode()` – stabilește rolul unui pin (intrare / ieșire);
- `digitalWrite()` – stabilește starea logice a ieșirii digitale;
- `digitalRead()` – preia stării logice a intrării digitale;
- `analogWrite()` – stabilește valoarea ieșirii analogice;
- `analogRead()` – preia valoarea intrării analogice;
- `delay()` – oprește programul pentru timpul specificat în milisecunde dat ca parametru

Și constantele:

1. HIGH / LOW – semnificația depinde dacă pinul este setat ca INPUT sau OUTPUT. Astfel dacă pinul este configurat ca INPUT cu `pinMode()` și interogată cu `digitalRead()` semnificația este:

▪ HIGH

- tensiune mai mare de 3.0V la pin (5V)
- tensiune mai mare de 2.0V la pin (3.3V)

▪ LOW

- o tensiune mai mică de 1.5V la pin (5V)
- o tensiune mai mică de 1V la pin (3.3V)

Dacă pinul este configurat ca OUTPUT cu pinMode() și setat cu digitalWrite() semnificația este:

- HIGH

- o tensiune de 5.0V la pin (5V)

- o tensiune de 3.3V la pin (3.3V)

- LOW

- o tensiune de 0V la pin (3.3V)

2. INPUT / OUTPUT – pin de intrare (switch, senzor) / pin de ieșire (LED, motor)

IV. IMPLEMENTARE HARDWARE

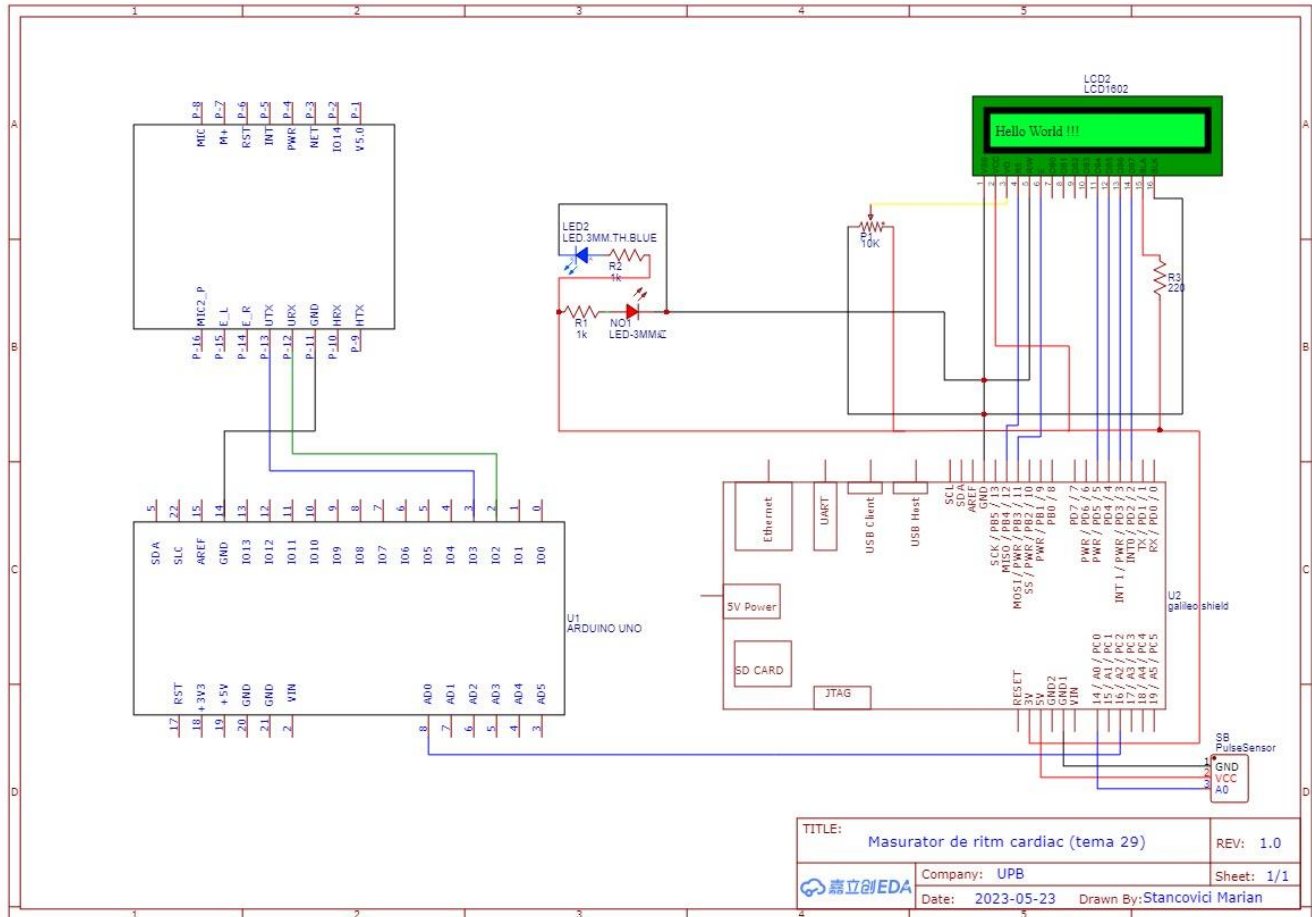


Figura 4.1 Schematic_EasyEda

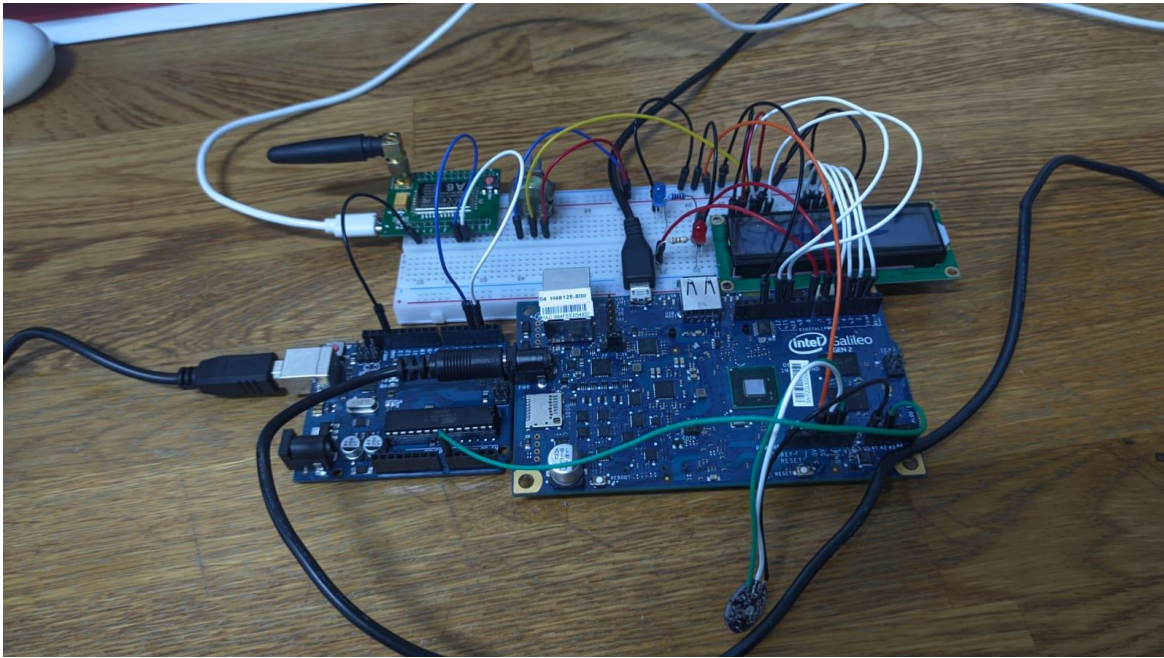


Figura 4.2 Proiect_Totul deconectat

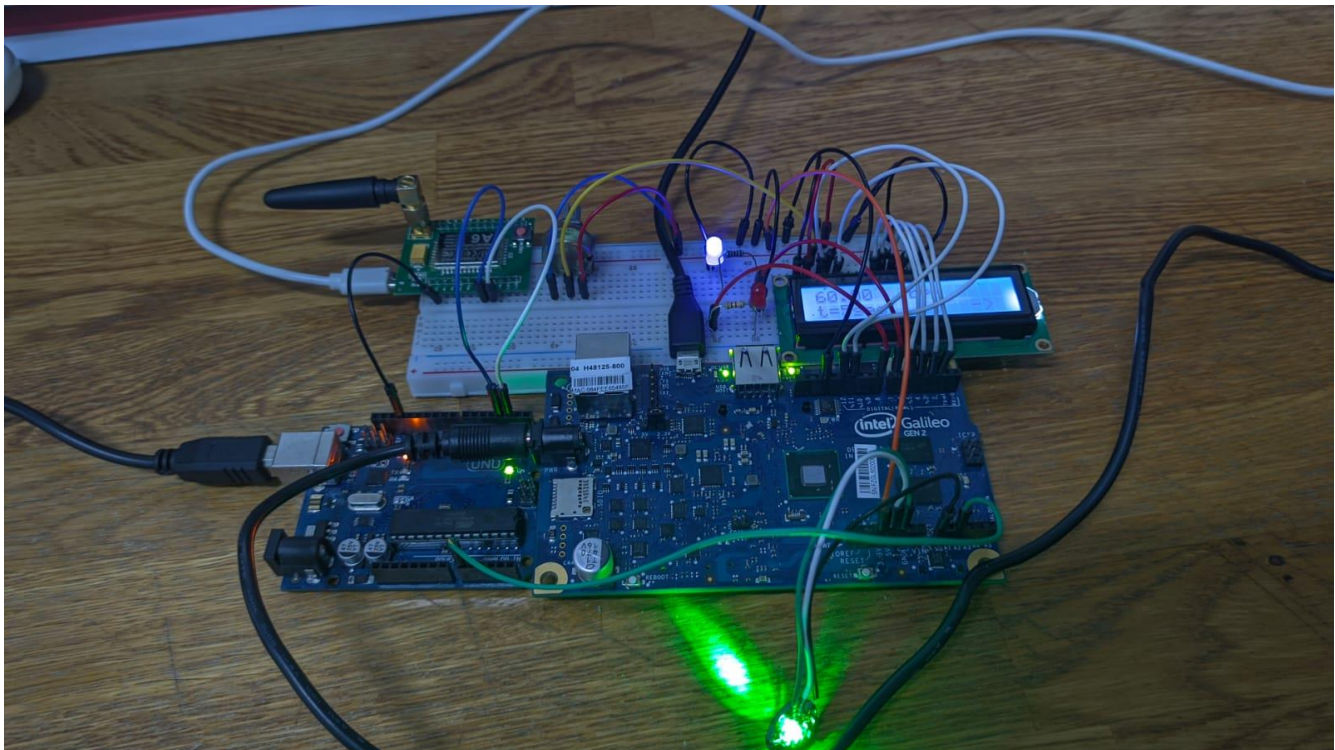


Figura 4.3 Proiect_Pornit

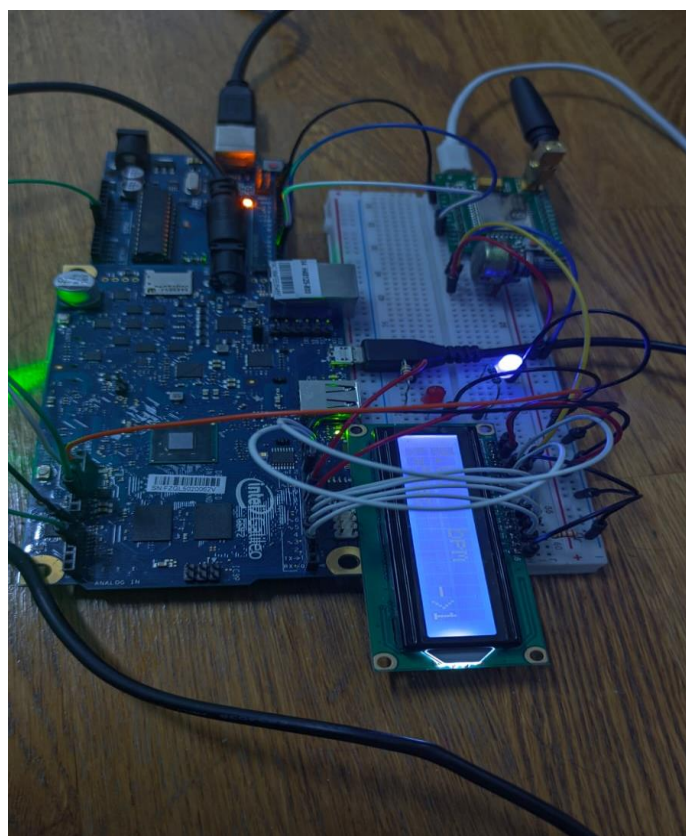


Figura 4.4. Proiect din alt unghi

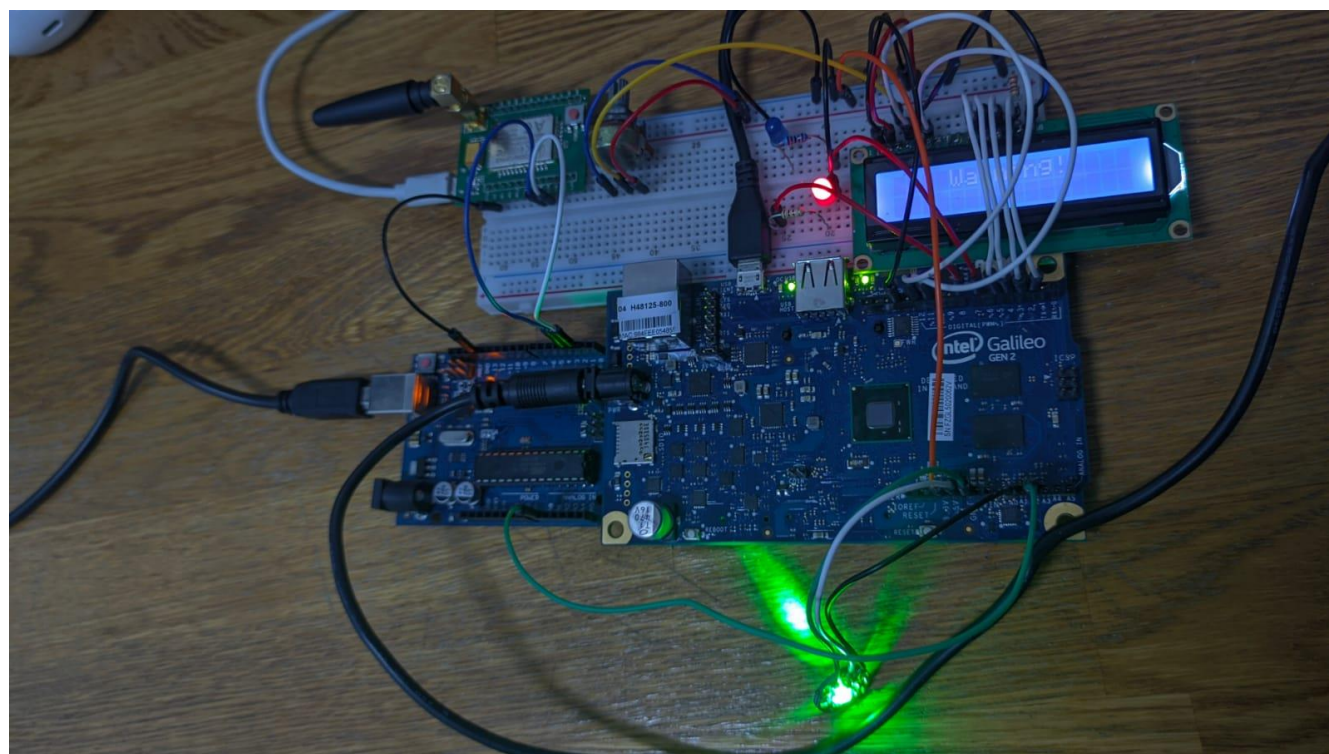


Figura 4.5. Proiect_LED rosu aprins (warning)

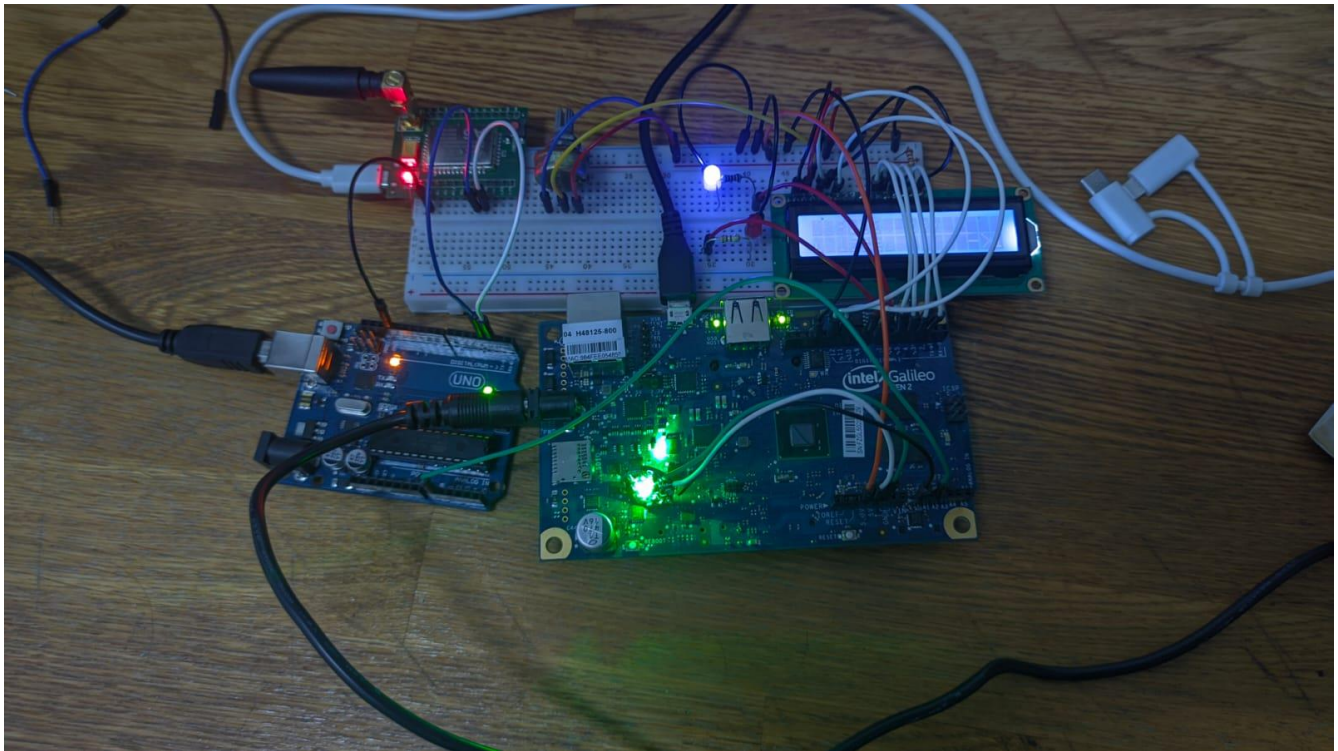


Figura 4.6. Proiect (GSM+LCD+Senzor puls)



Figura 4.7 Grafic pentru ritmul cardiac

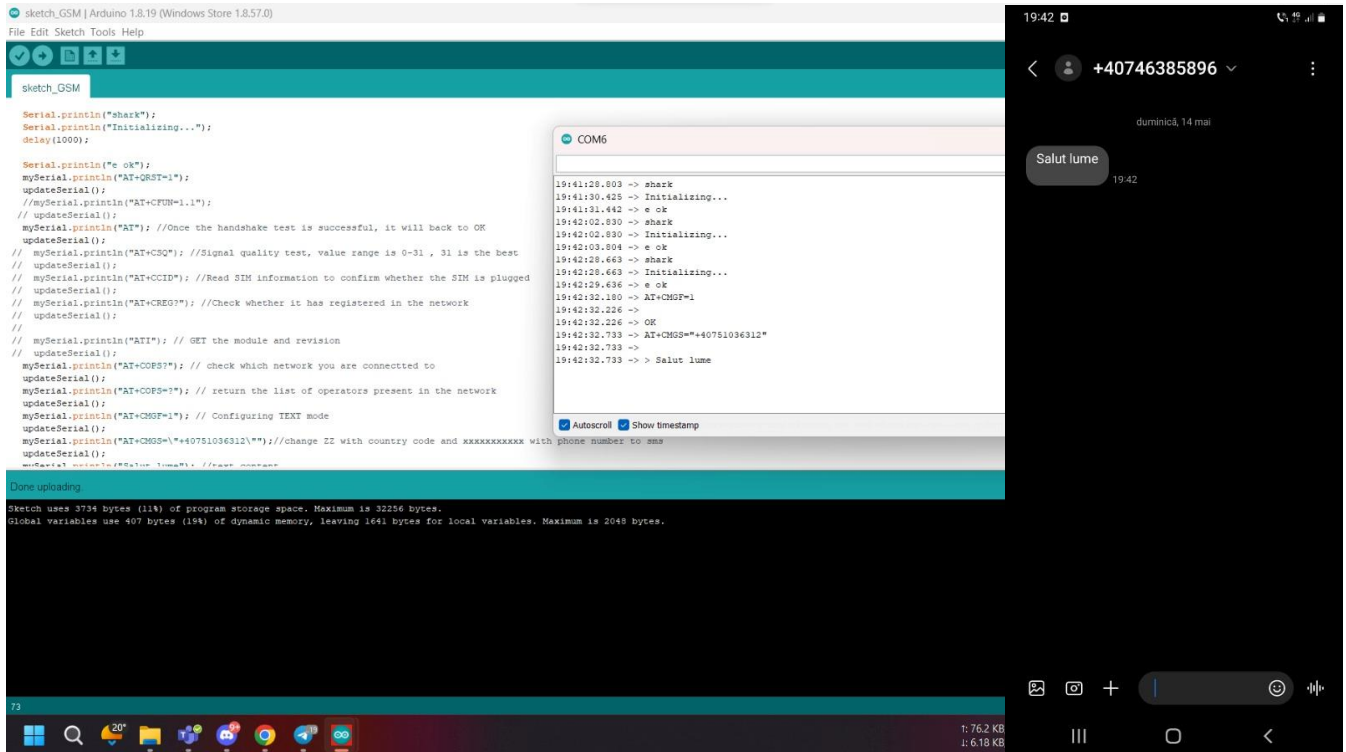


Figura 4.8 Mesaj trimis de modulul GSM + consola de comenzi

V. IMPLEMENTARE SOFTWARE

Cod LCD+Senzor de puls (Galileo)

```
##include <Arduino.h>

#include <LiquidCrystal.h>

////////////////////////Variables zone

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

static int nrtimp=0;
const int pinInput = A0;
const int pinOut= A2;
const int tension=520;//Jumatate din intervalul de valori posibile de la senzor
[290,550]
int nrbatai=0;
bool val=false;
int data=0;
int timpAnt=0;
static int possibleDanger=0;
bool warning=0;
static int contorTimp = 0;

////////////////////////Setup

void setup() {
    lcd.begin(16, 2);
    pinMode(7,OUTPUT);//led albastru
    pinMode(8,OUTPUT);//led rosu
    pinMode(A0,INPUT);
```

```

    digitalWrite(7,HIGH); //led albastru
    digitalWrite(8,LOW); //led rosu
    Serial.begin(115200); //baud rate este de 115200
    pinMode(pinOut,OUTPUT); //Folosim pinul A2 pentru a transmite la Uno inceperea
transmiterii mesajului prin GSM
}

//////////Loop
void loop() {
    if(!warning){
        digitalWrite(7,HIGH); //led albastru
        digitalWrite(8,LOW); //led rosu
        lcd.setCursor(0, 0);
        data=analogRead(pinInput); // data ia valoarea(intre 280 si 560) de la A0
        Serial.println(data); //Asa aflam valorile venite de la senzor
        SendPulse(data);
        lcd.setCursor(8, 0);
        lcd.print("bpm");
    }
    else{
        analogWrite(pinOut,255); //8 biti
        digitalWrite(7,LOW); //led albastru
        digitalWrite(8,HIGH); //led rosu
        lcd.clear();
        lcd.setCursor(4,0);
        lcd.print("Warning!");
        delay(3000);
        lcd.clear();
        delay(1000);
        warning=0;
        contorTimp = 0;
    }
}

```

```

//////////Functions
//Functia care printeaza pulsul(va trebui sa punem valoarea la LCD)
void SendPulse(int data){
    nrtimp=millis();//nrtimp ia valoarea din millis
    if(tension>data && val==true && data>514){
        val=false;
        nrbatai++;
    }
    if((tension<=data && val==false)){
        val=true;
    }
    if(nrtimp-timpAnt>=5000){//se verifica daca a trecut 1 secunda(senzorul
aproximeaza automat 60 batai intr-o secunda)
        //calc nr batai=(nrbatai/2.9) am calibrat folosind un ceas inteligent
        // eroare maxima de 11%

        lcd.print((float)(nrbatai)*12);
        //Serial.println((float)(nrbatai)*12);

        lcd.setCursor(0,1);
        contorTimp++;
        lcd.print("t=");
        lcd.print(contorTimp*5);
        lcd.print("sec");
        lcd.setCursor(13,1);
        lcd.print("->");
        lcd.setCursor(15,1);

        if((nrbatai*12)>150 || (nrbatai*12)<40){
            possibleDanger++;
            lcd.print(possibleDanger);
        }
    }
}

```

```
else{  
    possibleDanger = 0;  
}  
  
if(possibleDanger==3){  
    delay(1500);  
    possibleDanger = 0;  
    warning=1;  
    contorTimp = 0;  
}  
  
timpAnt=nrtimp;  
nrbatai=0;  
}  
}
```

Cod Modul GSM

```
#include <SoftwareSerial.h>

//Create software serial object to communicate with A6
SoftwareSerial mySerial(3, 2); //A6 Tx & Rx is connected to Arduino #3 & #2

//////////Variables

int pinIn=A0;
bool stopcode=0;

void setup()
{
    //Begin serial communication with Arduino and Arduino IDE (Serial Monitor)
    pinMode(pinIn, INPUT);
    Serial.begin(9600);

    //Begin serial communication with Arduino and A6
    mySerial.begin(9600);
    mySerial.println("AT+CFUN=1.1");
    updateSerial();

    ////////////Testing
    //  Serial.println("Merge pana aici");
    //  mySerial.println("AT");//Once the handshake test is successful, it will back to
    OK
    //  updateSerial();
    //  delay(1000);
    //  mySerial.println("AT+QBAND=7");
    //  updateSerial();
    //  mySerial.println("AT+CSQ"); //Signal quality test, value range is 0-31 , 31 is
    the best
    //  updateSerial();
    //  delay(1000);
```

```

// mySerial.println("AT+CCID"); //Read SIM information to confirm whether the SIM
is plugged
// updateSerial();
// delay(1000);
// mySerial.println("AT+CREG?"); //Check whether it has registered in the network
// updateSerial();
// delay(1000);
// mySerial.println("ATI"); // GET the module and revision
// updateSerial();
// mySerial.println("AT+COPS?"); // check which network you are connecttd to
// updateSerial();
// delay(1000);
// mySerial.println("AT+COPS=?"); // return the list of operators present in the
network
// updateSerial();

}

void loop()
{
    if(pinIn && !stopcode){
        ////////////////Messaging
        Serial.println("Incepe codul");
        delay(1000);
        mySerial.println("AT");//Once the handshake test is successful, it will back to
OK
        updateSerial();
        delay(1000);
        mySerial.println("AT+CMGF=1"); // Configuring TEXT mode
        updateSerial();
        delay(1000);
        mySerial.println("AT+CMGS=\""+40729364858\"");//change ZZ with country code and
xxxxxxxxxxxx with phone number to sms
    }
}

```

```

    updateSerial();
    delay(1000);
    mySerial.println("Puls grav, vezi ca mori"); //text content
    updateSerial();
    delay(1000);
    mySerial.write(26);
    ///////////////////////////////////Calling
    // mySerial.println("ATD+40729364858"); // change ZZ with country code and
    xxxxxxxxxxxx with phone number to dial
    // updateSerial();
    // delay(10000); // wait for 20 seconds...
    // mySerial.println("ATH"); //hang up
    // updateSerial();
    stopcode=~stopcode;
    }
    updateSerial();

}

void updateSerial()
{
    delay(500);
    while (Serial.available())
    {
        mySerial.write(Serial.read()); //Forward what Serial received to Software Serial
Port
    }
    while(mySerial.available())
    {
        Serial.write(mySerial.read()); //Forward what Software Serial received to Serial
Port
    }
}

```

VI. CONCLUZII

În ansamblu, acest proiect reprezintă o soluție practică și eficientă pentru monitorizarea ritmului cardiac și pentru notificarea utilizatorilor în cazul unor valori anormale ale ritmului cardiac.

Numărul de bătăi pe minut ale inimii este măsurat prin intermediul unui senzor de puls și afișat pe un ecran LCD, oferind utilizatorului informații în timp real despre ritmul său cardiac.

Integrarea LED-urilor, unul albastru și unul roșu, permite utilizatorului să primească feedback vizual imediat în ceea ce privește ritmul cardiac. LED-ul albastru indică un ritm cardiac normal, în timp ce aprinderea LED-ului roșu avertizează asupra unui ritm cardiac anormal, fie prea mare, fie prea mic.

Adăugarea unui modul GSM pentru a trimite un mesaj pe telefonul mobil în cazul aprinderii LED-ului roșu adaugă un nivel suplimentar de notificare și siguranță. Utilizatorii pot fi alertați în timp util și pot solicita asistență medicală de urgență sau pot notifica persoanele de contact în cazul unei situații critice.

VII. BIBLIOGRAFIE

[1] Intel Galileo Gen2

- <https://www.intel.com/content/dam/www/public/us/en/documents/datasheets/galileo-g2-datasheet.pdf>

[2] Modul LCD 1602 cu Backlight Albastru de 3.3 V:

- <https://www.optimusdigital.ro/ro/optoelectronice-lcd-uri/868-modul-lcd-1602-cu-backlight-galben-verde-de-33-v.html>
- <https://docs.arduino.cc/learn/electronics/lcd-displays>

[3] Senzor de Puls XD-58C:

- <https://www.optimusdigital.ro/ro/senzori-altele/1273-senzor-de-puls-xd-58c.html>
- <https://github.com/WorldFamousElectronics/PulseSensorPlayground>

[4] A6 GSM / GPRS Module

- <https://lastminuteengineers.com/a6-gsm-gprs-module-arduino-tutorial/>

[5] Arduino Uno

- <https://cleste.ro/arduino-uno-r3-atmega328p.html>

[6] Kit Breadboard HQ830 cu Fire și Sursă

- https://www.optimusdigital.ro/ro/kituri/2222-kit-breadboard-hq-830-p.html?search_query=kit+breadboard&results=37

[7] Arduino IDE

- https://docshare.tips/marian_58200141b6d87f6f998b4827.html