



Monitorizarea și măsurarea ritmului cardiac

Stancovici Marian

Preoteasa Rareș Marin

Neamțu Rareș Damian

Electronica

a avut un impact semnificativ asupra domeniului medical. Una dintre cele mai importante aplicații ale electronicii în medicină este măsurarea semnalelor electrice generate de inimă și interpretarea acestora într-un mod util în diagnosticarea și tratamentul bolilor cardiovasculare.

Introducere

- Proiectul nostru constă în realizarea unui sistem de monitorizare a ritmului cardiac utilizând un senzor de puls. Numărul de bătăi pe minut ale inimii va fi afișat pe un ecran LCD.
- Pentru valori firești ale ritmului cardiac se va aprinde un LED albastru. În cazul unor valori anormale ale ritmului cardiac, prea mici sau prea mari, se va aprinde un LED roșu.
- Sistemul este dotat cu un modul GSM, care permite trimiterea unui mesaj pe telefonul mobil în cazul în care LED-ul roșu se aprinde.

Resurse Hardware

Placa Intel Galileo Gen2



Senzor de Puls XD-58C



Modul LCD 1602 cu Backlight Albastru de 3.3 V



A6 GSM / GPRS Module

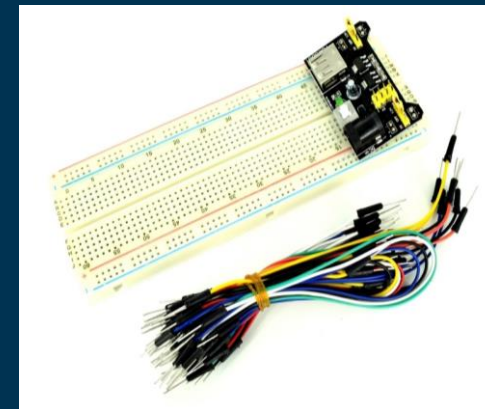


Resurse Hardware

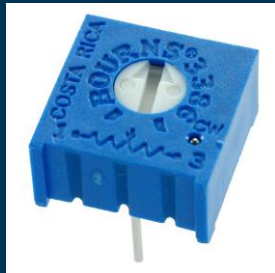
Arduino Uno



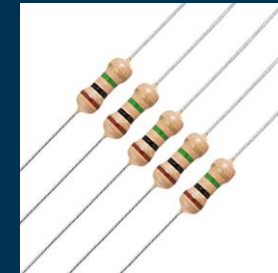
Kit Breadboard HQ830 cu Fire și Sursă



Potențiometru



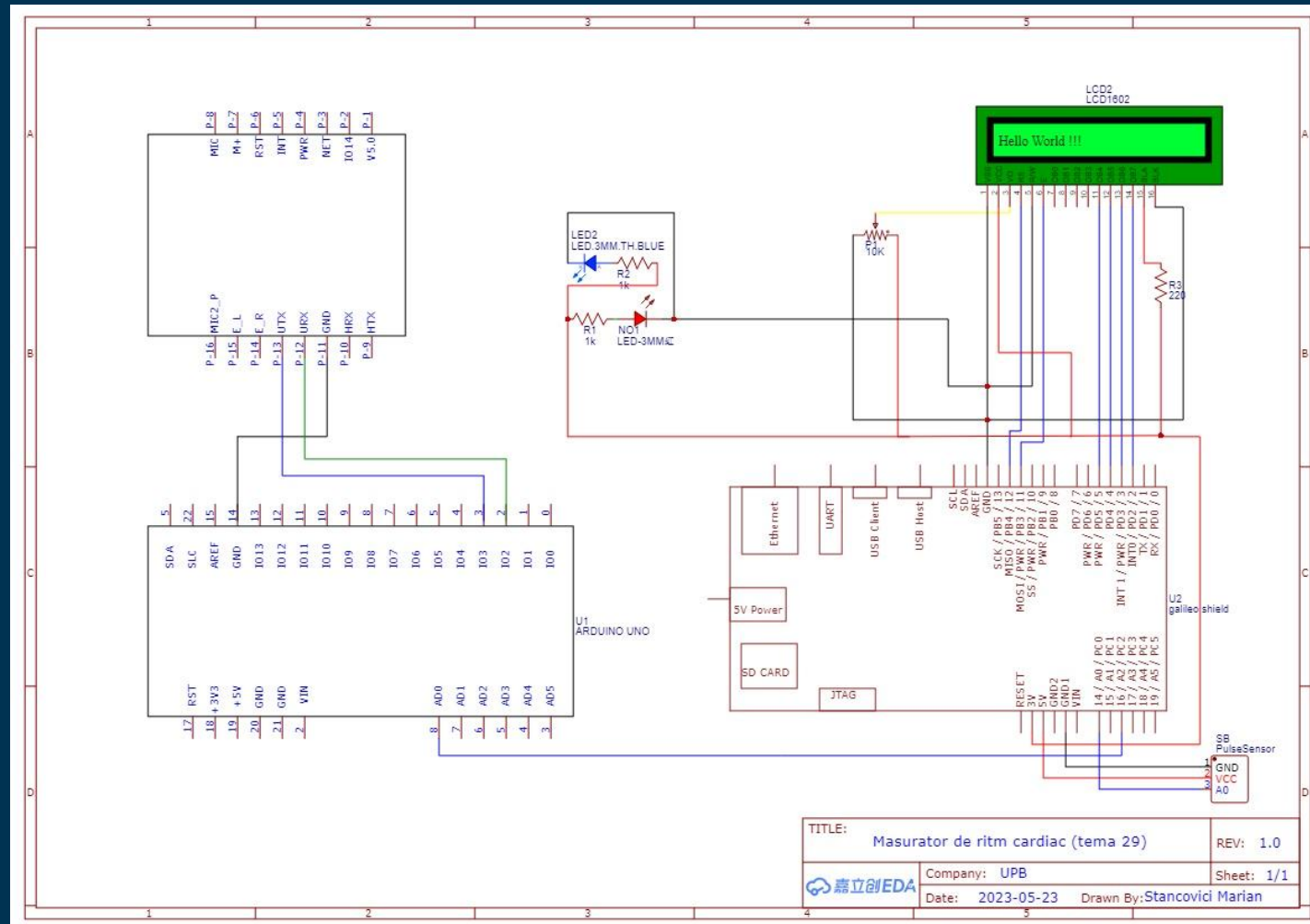
Rezistori



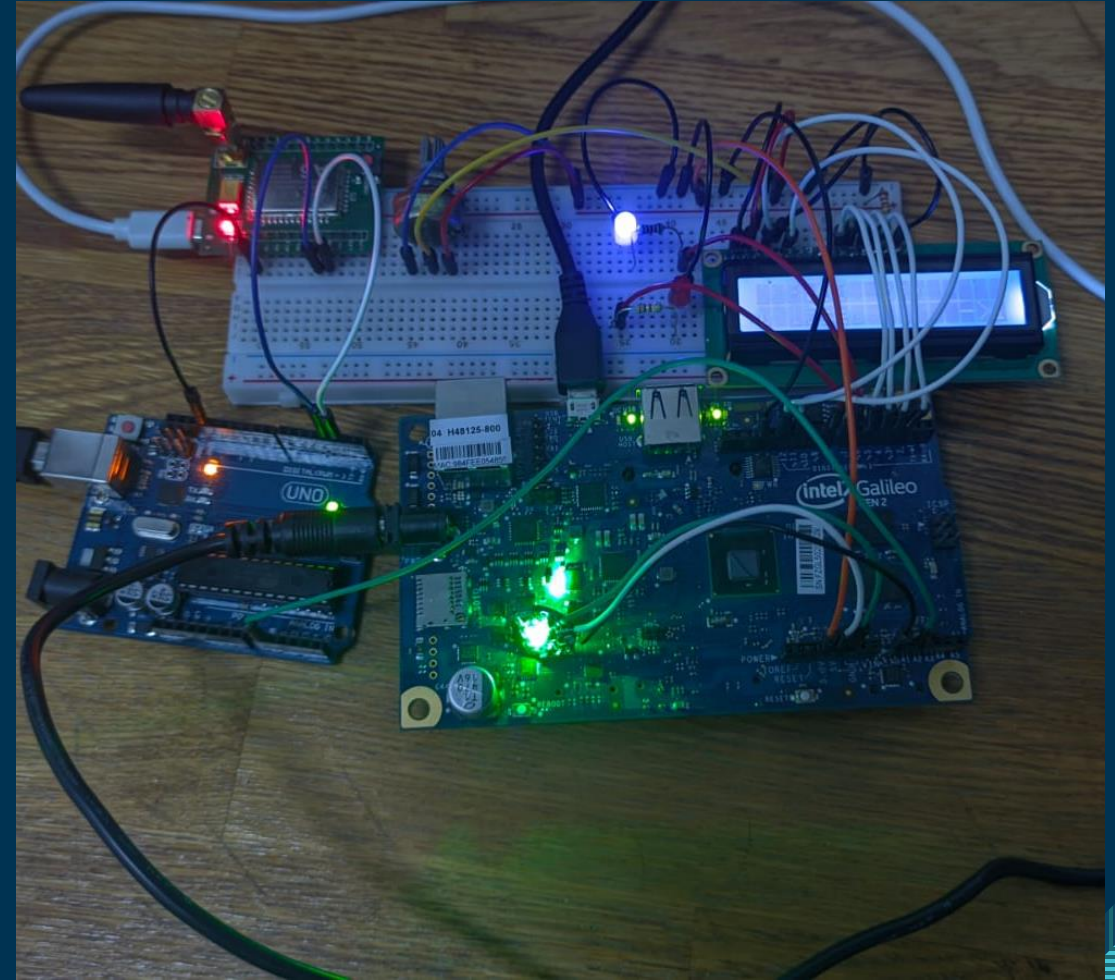
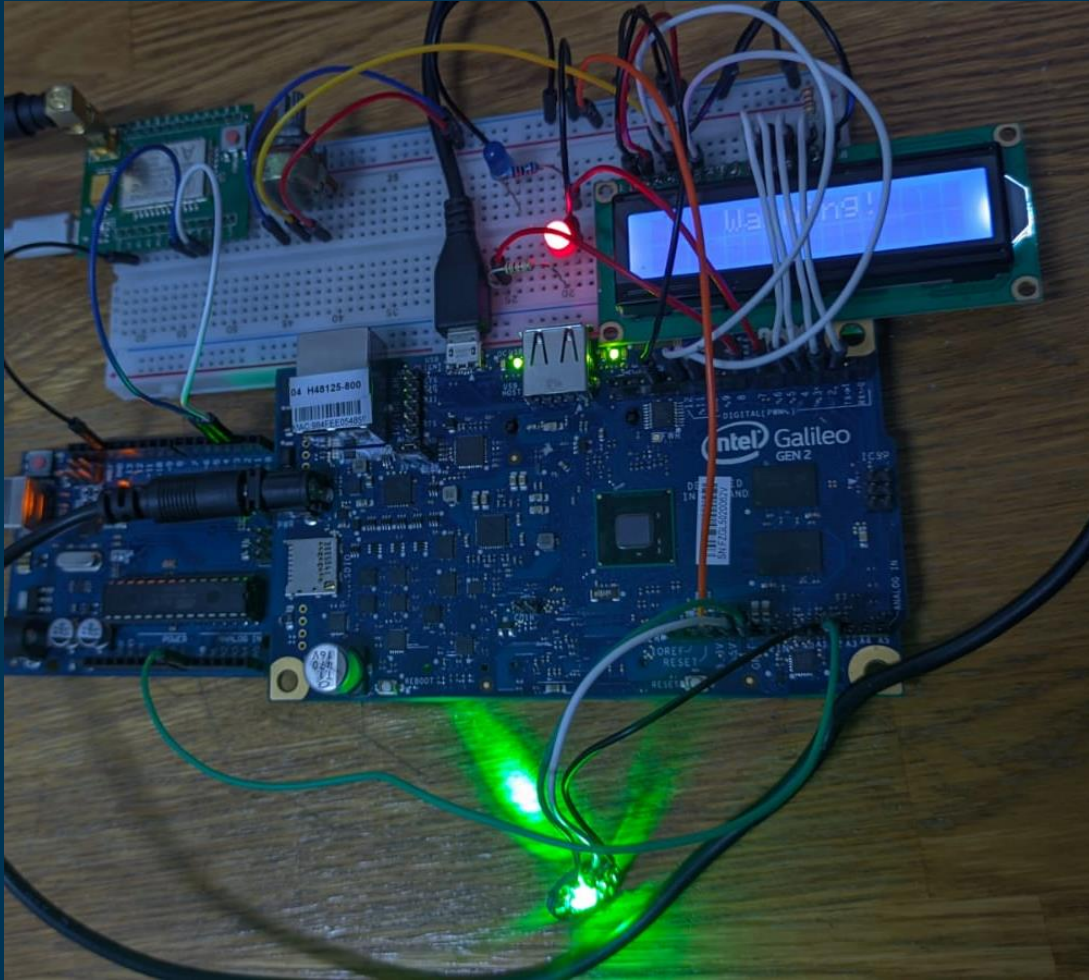
Resurse Software

- Plăcuța Galileo este compatibilă cu Arduino Software Development Environment (IDE).
- Arduino IDE suportă limbajele de programare C și C++ folosind reguli speciale de organizare a codului. 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.

Implementare Hardware - Schematic (EasyEda)



Implementare Hardware - Montaj

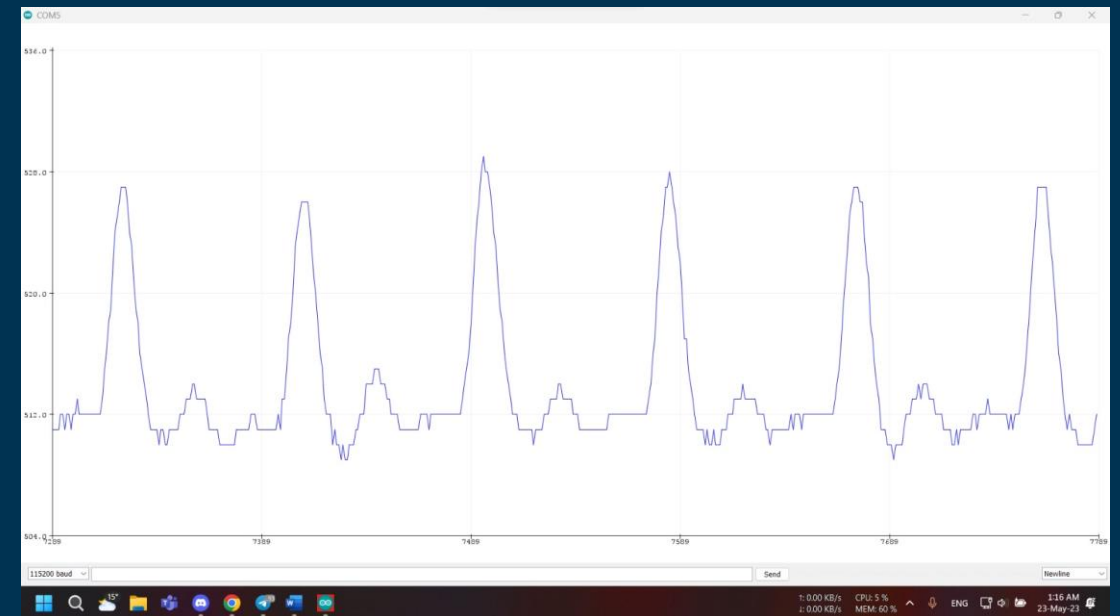


Implementare Software

Funcția care afișează pulsul:

```
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){  
        lcd.print((float)(nrbatai)*12);  
        timpAnt=nrtimp;  
        nrbatai=0;}  
}
```

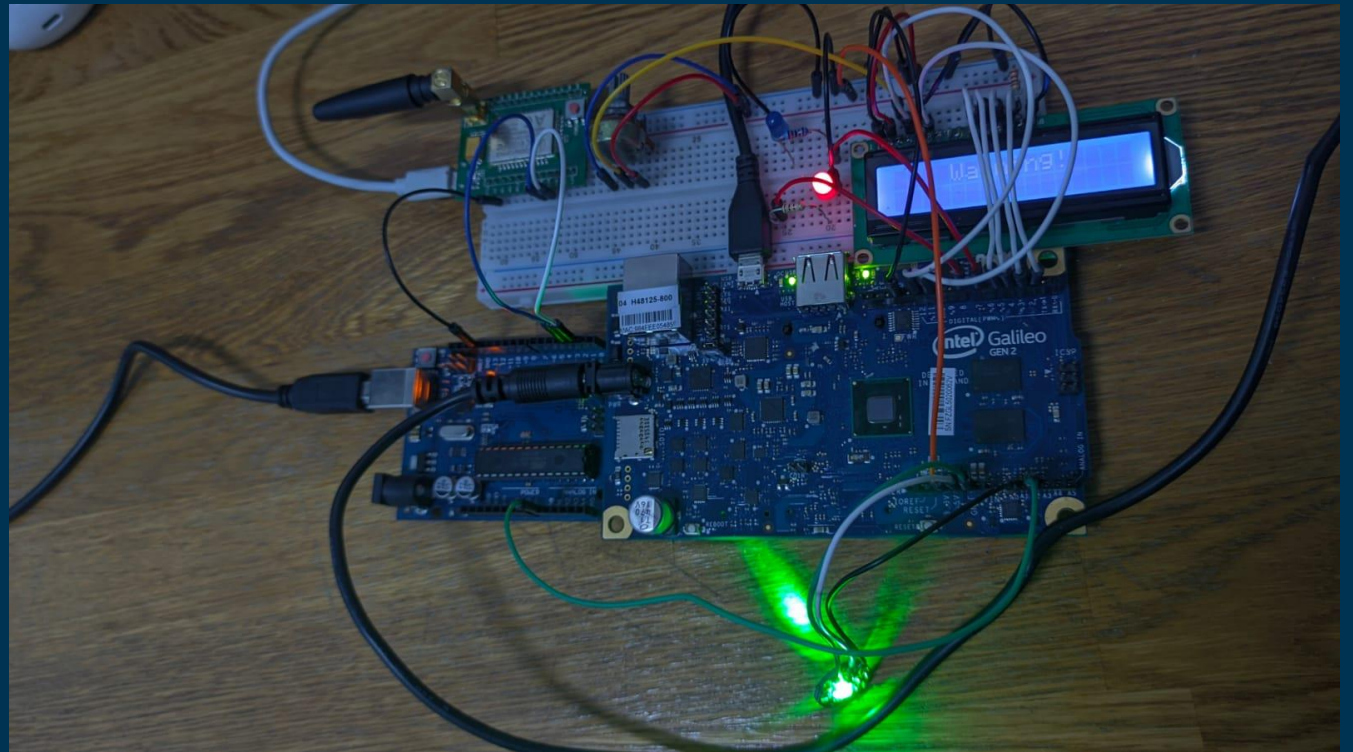
Ritmul cardiac normal sub forma de variații ale tensiunii de intrare date de senzorul de puls



Implementare Software

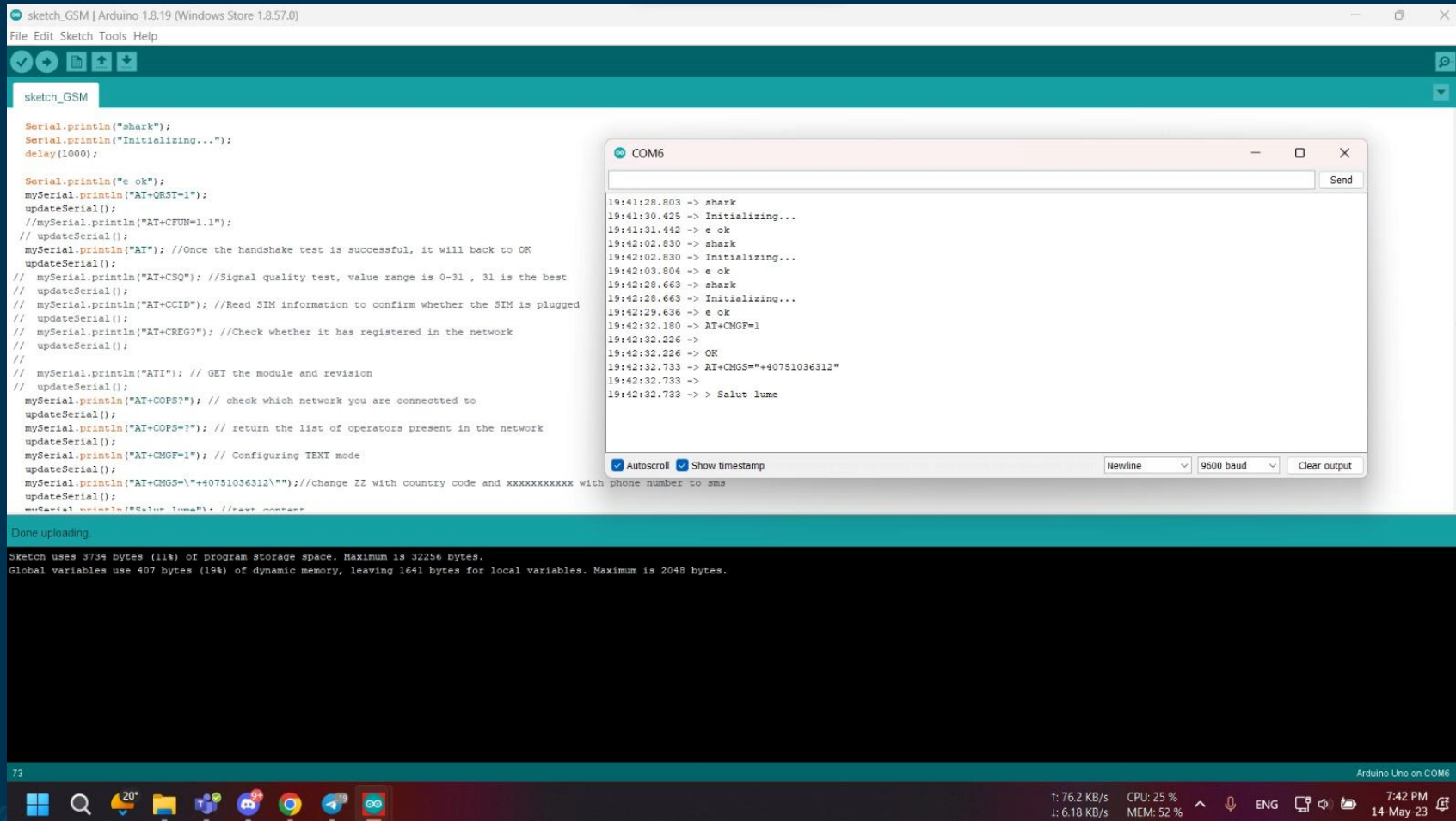
Verificare iregularități cardiace: „Warning!”

```
if((nr batai*12)>150 || (nr batai*12)<40){  
    possibleDanger++;  
    lcd.print(possibleDanger);  
}  
else{  
    possibleDanger = 0;  
}  
if(possibleDanger==3){  
    delay(1500);  
    possibleDanger = 0;  
    warning=1;  
    contorTimp = 0;  
}
```



Implementare Software

Funcționare Modul GSM:



The screenshot displays the Arduino IDE interface. The main window shows the 'sketch_GSM' code, which includes initialization and AT command sequences for a GSM module. A serial monitor window titled 'COM6' is open, showing the real-time output of the code. The output indicates successful initialization and the sending of an SMS.

```
sketch_GSM

Serial.println("shark");
Serial.println("Initializing...");
delay(1000);

Serial.println("e ok");
mySerial.println("AT+QBST=1");
updateSerial();
//mySerial.println("AT+CFUN=1.1");
// updateSerial();
mySerial.println("AT"); //Once the handshake test is successful, it will back to OK
updateSerial();
// mySerial.println("AT+CSQ"); //Signal quality test, value range is 0-31, 31 is the best
// updateSerial();
// mySerial.println("AT+CCID"); //Read SIM information to confirm whether the SIM is plugged
// updateSerial();
// mySerial.println("AT+CREG?"); //Check whether it has registered in the network
// updateSerial();
//
// mySerial.println("ATI"); // GET the module and revision
// updateSerial();
mySerial.println("AT+COPS?"); // check which network you are connected to
updateSerial();
mySerial.println("AT+COPS=?"); // return the list of operators present in the network
updateSerial();
mySerial.println("AT+CMGF=1"); // Configuring TEXT mode
updateSerial();
mySerial.println("AT+CMGS=\""+40751036312\""); //change ZZ with country code and XXXXXXXXXXXX with phone number to sms
updateSerial();
mySerial.println("Salut lume"); //send content

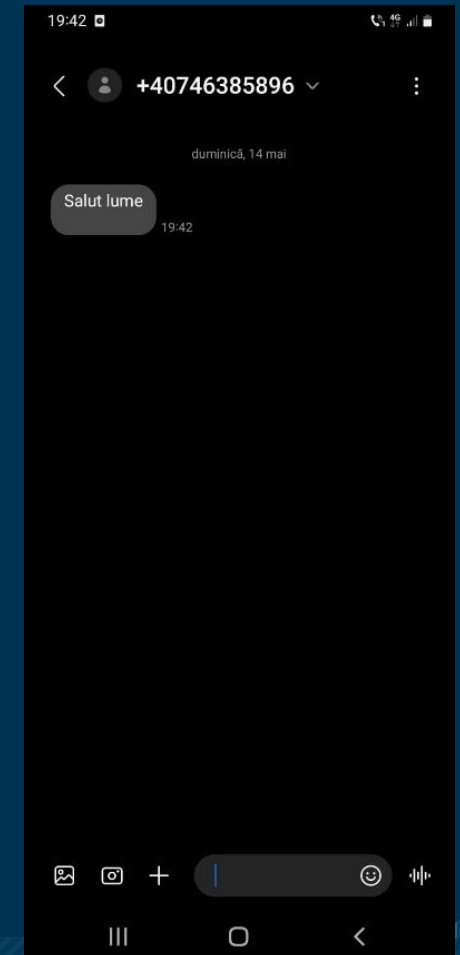
Done uploading.

Sketch uses 3734 bytes (11% of program storage space. Maximum is 32256 bytes.
Global variables use 407 bytes (19% of dynamic memory, leaving 1641 bytes for local variables. Maximum is 2048 bytes.
```

COM6

```
19:41:28.803 -> shark
19:41:30.425 -> Initializing...
19:41:31.442 -> e ok
19:42:02.830 -> shark
19:42:02.830 -> Initializing...
19:42:03.804 -> e ok
19:42:28.663 -> shark
19:42:28.663 -> Initializing...
19:42:29.636 -> e ok
19:42:32.180 -> AT+CMGF=1
19:42:32.226 ->
19:42:32.226 -> OK
19:42:32.733 -> AT+CMGS="+40751036312"
19:42:32.733 ->
19:42:32.733 -> > Salut lume
```

Autoscroll Show timestamp Newline 9600 baud Clear output



Concluzii

- Acest proiect reprezintă o soluție practică și eficientă pentru monitorizarea ritmului cardiac și pentru notificarea utilizatorilor în cazul unor valori anormale ale acestuia.
- Numărul de bătăi pe minut ale inimii este măsurat prin intermediul unui senzor de puls și afișat pe un ecran LCD.
- Integrarea LED-urilor, unul albastru, ce indică valori firești ale ritmului cardiac, și unul roșu, ce semnalează un ritm cardiac fie prea mare, fie prea mic, permite utilizatorului să primească un feedback vizual imediat.
- 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.

Muğumim! ☺