

Universitatea Politehnica Timisoara

# Camera de Interogatoriu

Proiect realizat de:  
Bacer Raul  
Bar Bianca

# Ideea proiectului

Proiectul consta in **detectarea momentelor in care o persoana minte.**

Daca se detecteaza astfel de momente, **un ventilator va porni** pentru a reduce temperatura din camera.

Ulterior, detectorul de minciuni isi **va relua activitatea** si va continua pana cand interogarea se termina.

# Cum functioneaza

Se intampla o singura data

Calculeaza **pulsul mediu** astfel:

- Se defineste un **numar de batai de inima** (ex.: 3), cu acest numar de batai se va calcula media pulsului
- Se face o **suma** din acel numar de batai (ex.:  $64+65+63=\text{suma}$ )
- Suma se imparte la numarul de batai si se face **media pulsului** (ex.: 64)

# Cum functioneaza

Se intampla pana se opreste programul:

Verifica daca persoana **minte**:

- Se citește **pulsul actual**
- Se afiseaza pe **LCD** pulsul actual si cel mediu
- Daca pulsul actual **e mai mare** cu un numar de unitati (ex.: 15) decat cel mediu, persoana **minte**; in caz contrar, spune **adevarul**

# Minciuna:

- se opreste becul verde
- porneste becul rosu
- se afiseaza pe LCD textul  
“MINCIUNA”
- se afiseaza in consola textul  
“MINCIUNA”
- porneste ventilatorul, se va invarti  
pentru un numar specificat de secunde
- se opreste becul rosu
- se sterge textul de pe LCD
- porneste becul verde
- se trece la urmatoare iteratie a  
programului

# Adevar:

- se trece la urmatoarea iteratie a  
programului (nu se cauta adevarul)

# Componente

- **placa** Arduino Uno
- **breadboard**
- **bec** verde
- **bec** rosu
- **senzor** de puls
- **ecran** LCD 16x2
- **potentiometru**
- **servo-motor**



# **Variabile si Valori folosite**

```
//SENZOR PULS  
#include <PulseSensorPlayground.h>  
PulseSensorPlayground senzorPuls;
```

Aici se initializeaza senzorul de puls  
(biblioteca este creata pentru astfel de  
senzori).

```
//MOTOR  
#include <Servo.h>  
Servo motor;
```

Aici se initializeaza servo-motorul.

```
//LCD  
#include <LiquidCrystal.h>  
LiquidCrystal LCD(12,11,10,9,8,7);
```

Aici se initializeaza LCD-ul si se  
conecteaza la pinii digitali specificati.



```
//PINI  
const int pinBecV=2;  
const int pinBecR=3;  
  
const int pinMotor=4;  
  
const int pinPuls=A0;
```

Acestia sunt pinii folositi si legati la placa Arduino.

```
//VALORI  
const int secundeMotor=2;  
const int bataiPentruMedie=15;  
const int pragMinciuna=15;
```

Acestea sunt valorile folosite in program.  
Ele pot fi schimbate pentru o acuratete mai mare a  
codului.

```
//PULS
```

```
int puls;
```

```
int numarPuls;
```

```
int sumaPuls;
```

```
int pulsMediu;
```

```
//MINCIUNA
```

```
int minciuna;
```

Acestea sunt variabilele declarate global ce vor fi folosite pe parcursul programului.

# Funcțiile implementate

Unele funcții au la final un delay mic pentru ca ceea ce se întâmplă pe placă să funcționeze “smooth”.

# citestePuls

```
void citestePuls()  
{  
    puls=senzorPuls.getBeatsPerMinute();  
}
```

Aceasta functie are rolul de a citi pulsul curent de la incheietura mainii.

Implementeaza o alta functie din biblioteca introdusa anterior.

**pornesteBec**

**opresteBec**

```
void pornesteBec(const int pinBec)
{
    digitalWrite(pinBec,HIGH);
    delay(10);
}
```

```
void opresteBec(const int pinBec)
{
    digitalWrite(pinBec,LOW);
    delay(10);
}
```

Aceste functii au rolul de a porni becul, respectiv de a il opri.

# calculeazaPulsMediu

```
void calculeazaPulsMediu()  
{  
    sumaPuls=sumaPuls+puls;  
    numarPuls=numarPuls+1;  
  
    if(numarPuls==bataiPentruMedie)  
    {  
        pulsMediu=sumaPuls/numarPuls;  
    }  
}
```

Aceasta functie are rolul de a calcula pulsul mediu, proces explicat in slide-ul 3.

Abia dupa ce acesta a fost determinat, programul va incepe sa detecteze minciuni.

# verificaMinciuna

```
void verificaMinciuna()  
{  
    if(puls>=pulsMediu+pragMinciuna)  
    {  
        minciuna=1;  
    }  
    else  
    {  
        minciuna=0;  
    }  
}
```

Aceasta functie are rolul de a observa daca persoana minte, procesul de functionare a fost explicat in slide-ul 4.

Daca pulsul indica minciuna, variabila “minciuna” va fi setata pe 1, in caz contrar pe 0.



```
void existaMinciuna()
{
    opresteBec(pinBecV);

    pornesteBec(pinBecR);

    scrieLCD(0,1,"MINCIUNA!",-1);
    scrieConsola("MINCIUNA!",-1);

    pornesteMotor();

    opresteBec(pinBecR);

    scrieLCD(0,1,"          ",-1);

    pornesteBec(pinBecV);
}
```

## existaMinciuna

Aceasta functie are rolul de a face actiunile ce trebuiesc facute cand exista o minciuna, proces explicat in slide-ul 5.

# pornesteMotor

```
void pornesteMotor()  
{  
    motor.write(0);  
    delay(secundeMotor*1000);  
    motor.write(90);  
    delay(300);  
}
```

Aceasta functie are rolul de a porni motorul pentru un numar specificat de secunde.

```
void scrieConsola(char* sir, int valoare)
{
    if(valoare!=-1)
    {
        Serial.print(sir);
        Serial.print(valoare);
        Serial.println();
    }
    else
    {
        Serial.print(sir);
        Serial.println();
    }
}
```

**scrieConsola**

Aceasta functie are rolul de a scrie in consola.

Daca variabila “valoare” are valoarea -1, se va scrie in consola doar variabila “sir”.

Daca variabila “valoare” are valoarea diferita de -1, se vor scrie in consola variabilele “sir” si “valoare”.

```
void scrieLCD(int coloana, int linie, char* sir, int valoare)
{
    LCD.setCursor(coloana,linie);
    int i;
    for(i=0;i<16;i++)
    {
        LCD.print(" ");
    }
    LCD.setCursor(coloana,linie);
    if(valoare!=-1)
    {
        LCD.print(sir);
        LCD.print(valoare);
    }
    else
    {
        LCD.print(sir);
    }
    delay(10);
}
```

Aceasta functie are rolul de a  
scrie pe LCD.

Daca variabila “valoare” are  
valoarea -1, se va scrie pe LCD  
doar variabila “sir”.

Daca variabila “valoare” are  
valoarea diferita de -1, se vor  
scrie pe LCD variabilele “sir” si  
“valoare”.

Variabilele “coloana” si “linie”  
indica pozitia cursorului pe LCD.

**scrieLCD**

```
void setup()
```

```
{
```

```
pinMode(pinBecV,OUTPUT);.....Initializez pinii cu valori de input/output.
```

```
pinMode(pinBecR,OUTPUT);
```

```
pinMode(pinMotor,OUTPUT);
```

```
pinMode(pinPuls,INPUT);
```

```
senzorPuls.analogInput(pinPuls);.....Initializez senzorul de puls si il pornesc.
```

```
senzorPuls.begin();
```

```
motor.attach(pinMotor); .....Initializez servo-motorul.
```

```
LCD.begin(16,2); .....Pornesc LCD-ul.
```

```
minciuna=0; .....Initializez variabilele globale:
```

```
numarPuls=0;
```

```
sumaPuls=0;
```

```
pulsMediu=0;
```

```
pornesteBec(pinBecV);
```

```
Serial.begin(9600);
```

```
}
```

**setup**

- minciuna, este 1 daca exista minciuna
- numarPuls, reprezinta cate batai de inima au fost citite
- sumaPuls, suma acelor batai citite
- pulsMediu este sumaPuls/numarPuls

```

void loop()
{
    citestePuls(); ..... Citesc pulsul si il afisez in consola si
    scrieConsola("Puls: ",puls);                          pe LCD.
    scrieLCD(0,0,"Puls: ",puls);
    if(pulsMediu==0) ..... Daca nu exista puls mediu (nu s-
    {                               au facut destule iteratii pentru a il
        calculeazaPulsMediu();                      calcula), se va calcula.
    }
    else ..... Daca exista puls mediu:
    {
        scrieConsola("Puls mediu: ",pulsMediu);
        scrieLCD(0,0,"Puls: ",puls);
        scrieLCD(9,0,"Med: ",pulsMediu);
        verificaMinciuna();
        if(minciuna==1)
        {
            existaMinciuna();
        }
    }
    scrieConsola("",-1);
    delay(500);
}

```

**loop**

# **Conectarea pinilor**

# Servo-motor

Semnal -> 4 (Arduino Digital)

Power -> Breadboard +

Ground -> Breadboard -



# Ecran LCD

GND -> Breadboard -

VDD -> Breadboard +

VO -> Mijloc (Potentiometru)

RS -> 12 (Arduino Digital)

RM -> Breadboard -

E -> 11 (Arduino Digital)

D4 -> 10 (Arduino Digital)

D5 -> 9 (Arduino Digital)

D6 -> 8 (Arduino Digital)

D7 -> 7 (Arduino Digital)

BLA -> Breadboard + (si rezistenta de 220 OHM)

BLK -> Breadboard -

# Potentiometru

Stanga -> Breadboard -

Mijloc -> VO (LCD)

Dreapta -> Breadboard +

# Senzor de puls

Semnal -> A0 (Arduino Analog)

Power -> Breadboard +

Ground -> Breadboard -

# Arduino

## Power:

5V -> Breadboard +  
GND -> Breadboard -

## Digital:

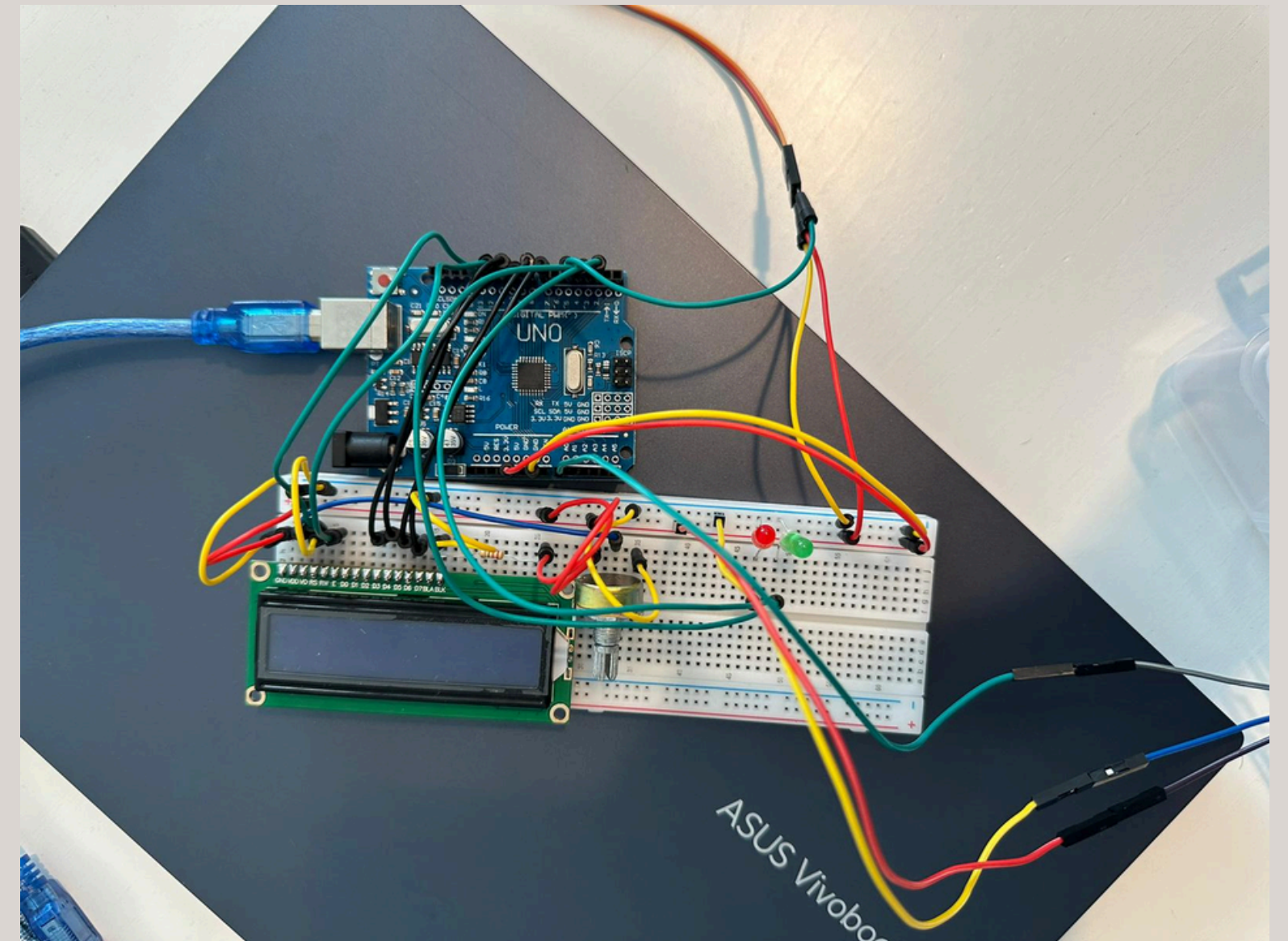
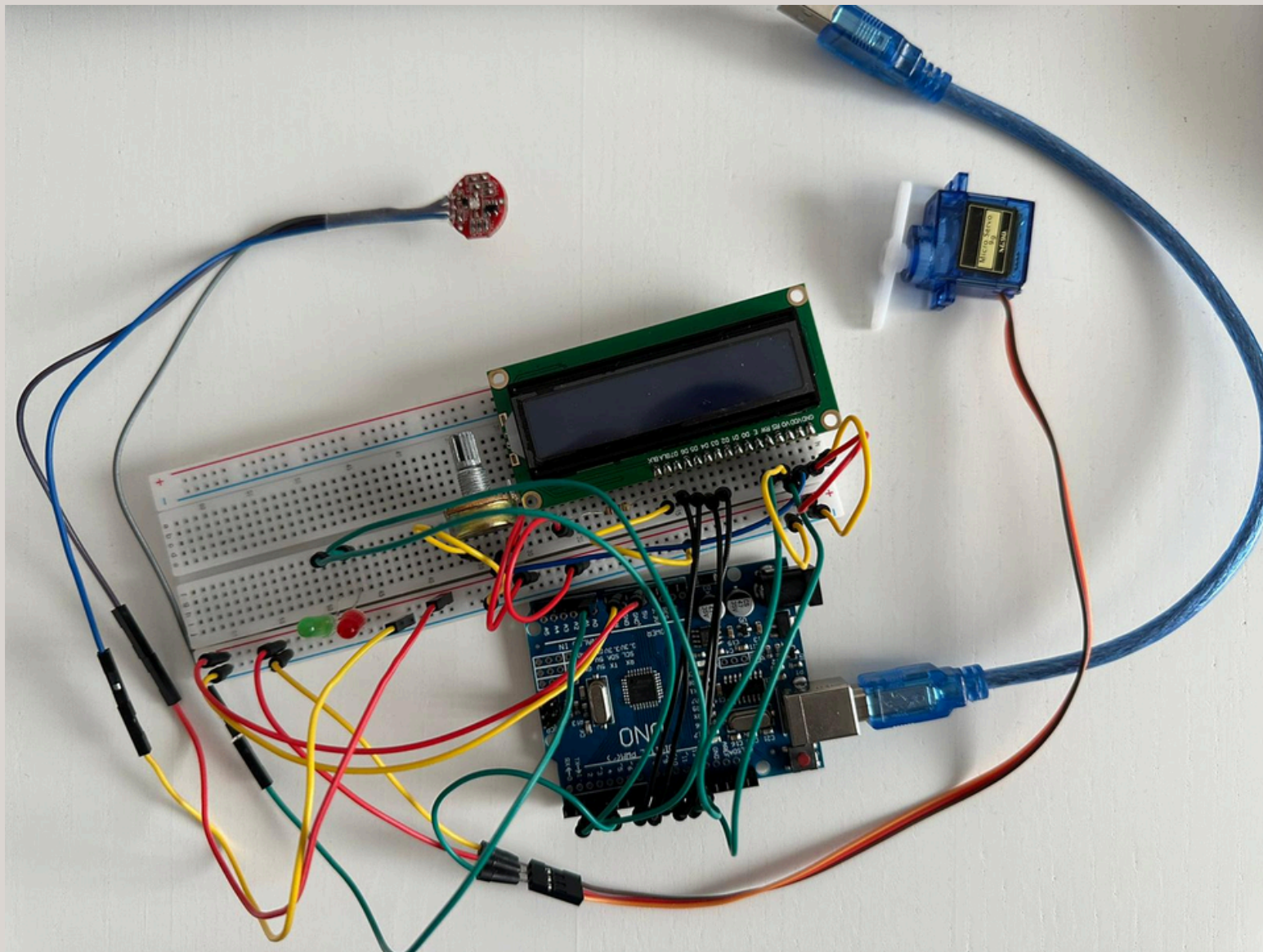
2 -> Bec Verde  
3 -> Bec Rosu  
4 -> Servo-motor  
7 -> D7 (LCD)  
8 -> D6 (LCD)  
9 -> D5 (LCD)  
10 -> D4 (LCD)  
11 -> E (LCD)  
12 -> RS (LCD)

## Analog:

A0 -> Senzor de puls



# Schema proiectului



*Thank  
you!*