

Componente: - senzor PULS - senzor TEMPERATURA - senzor UMIDITATE - ecran

Ecranul afiseaza concluzii in urma masuratorilor!

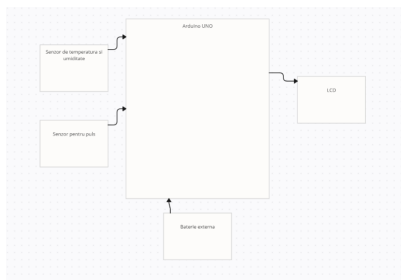
# Detector de emotii

## Introducere

Proiectul propus este un sistem de monitorizare al emoțiilor care utilizează senzori pentru puls, umiditate și temperatură, și afișează datele colectate pe un LCD. Codul sursă necesar pentru funcționarea acestui sistem este încărcat pe o placă Arduino.

Acest proiect nu numai că este capabil să monitorizeze emoțiile umane, ci poate fi, de asemenea, folosit ca un detector de minciuni, utilizând datele fiziologice și de mediu pentru a identifica discrepanțe între starea declarată și starea reală a utilizatorului. Prin integrarea mai multor senzori și algoritmilor de analiză a datelor, acest sistem poate oferi o evaluare mai precisă a stării emoționale și a sincerității individului, având potențialul de a fi folosit într-o varietate de aplicații, de la monitorizarea sănătății mentale până la interogatorii și evaluări de securitate.

## Descriere generală



### 1. Hardware:

- Arduino: Microcontrolerul Arduino este componenta centrală a proiectului și este responsabil de preluarea datelor de la senzori și controlul afișajului LCD
- Senzorul XD-58C: Acest senzor măsoară pulsul și ritmul cardiac și furnizează datele către Arduino
- Senzorul DHT11: Acest senzor măsoară temperatura și umiditatea și furnizează datele către Arduino
- Afișaj LCD: Afișajul LCD este utilizat pentru a afișa datele colectate de la senzori
- Baterie externă

### 2. Software:

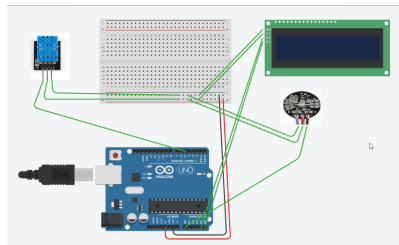
- Biblioteci: <LiquidCrystal\_I2C.h>, <dht.h>, <PulseSensorPlayground.h>
- Configurare: Software-ul include funcții de configurare a senzorilor și a afișajului LCD pentru a asigura funcționarea corectă a acestora.
- Prelevarea datelor: Arduino preia datele de la senzori prin intermediul interfeței hardware corespunzătoare și le procesează.
- Afișarea datelor: Arduino afișează datele colectate pe afișajul LCD într-un format ușor de citit.

Modul în care modulele hardware și software interacționează este următorul: Arduino inițializează și configurează senzorii și afișajul LCD. După aceasta, Arduino preia datele de la senzori în mod regulat și le afișează pe afișajul LCD. Astfel, utilizatorul poate vizualiza în timp real informațiile colectate de senzori, facilitând monitorizarea și interpretarea datelor.

## Hardware Design

Lista de piese:

- Senzor de puls XD-58C
- Senzor DHT11
- LCD I2C
- Arduino UNO
- Breadboard
- Fire de conexiune
- Baterie Externa



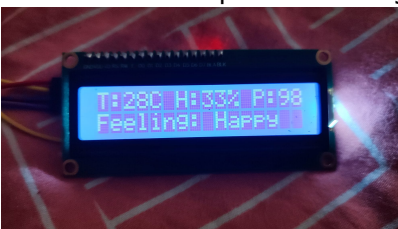
Schema electrica a circuitului:

Rezultatele Simularii:

- Simulare 1: Temperatura: 23 grade Celsius, Umiditate: 46%, Puls: 63 bpm, Sentiment: Relaxat



- Simulare 2: Temperatura: 23 grade Celsius, Umiditate: 46%, Puls: 63 bpm, Sentiment: Fericit



# Software Design

Mediul de dezvoltare:

Pentru acest proiect, am utilizat Arduino IDE pentru programarea pe Arduino UNO, deoarece este o aplicație gratuită și ușor de folosit, care oferă o serie de funcționalități utile pentru dezvoltarea proiectelor pe platforma Arduino.

Librării:

- **Wire:** Biblioteca Wire este utilizată pentru comunicația I2C (Inter-Integrated Circuit) între Arduino și alte dispozitive compatibile I2C, cum ar fi senzori, afișaje și alte microcontrolere. Aceasta permite conectarea a multiple dispozitive pe același bus de comunicație folosind doar două fire (SDA pentru date și SCL pentru ceas). Funcții principale: `begin()`, `requestFrom()`, `beginTransmission()`, `endTransmission()`, `write()`, `read()`.
- **LiquidCrystal\_I2C** simplifică controlul afișajelor LCD compatibile cu protocolul I2C. Folosirea acestei biblioteci reduce numărul de pini necesari pentru a conecta un afișaj LCD la Arduino, comparativ cu conexiunile standard (4 sau 8 pini). Funcții principale: `begin()`, `print()`, `setCursor()`, `clear()`.
- **DHT:** Biblioteca DHT este utilizată pentru a lucra cu senzorii de temperatură și umiditate DHT11, DHT22, etc. Aceste senzori sunt ieftini și ușor de utilizat, oferind măsurători precise ale temperaturii și umidității. Funcții principale: `begin()`, `readTemperature()`, `readHumidity()`.
- **PulseSensorPlayground:** Biblioteca PulseSensorPlayground este special creată pentru a lucra cu Pulse Sensor, un senzor pentru măsurarea ritmului cardiac. Aceasta simplifică procesul de citire a datelor și de interpretare a semnalelor provenite de la senzor. Funcții principale: `begin()`, `getBeatsPerMinute()`, `sawNewBeat()`.

Cod:

```
#include <Wire.h> #include <LiquidCrystal_I2C.h> #include <dht.h> #include  
<PulseSensorPlayground.h>
```

```
#define DHTPIN 5 #define PULSE_PIN A0
```

```
LiquidCrystal_I2C lcd(0x3F, 16, 2); PulseSensorPlayground pulseSensor; dht DHT;
```

```
const float happyTempThreshold = 25.0; const float happyHumidityThreshold = 50.0; const int  
happyPulseThreshold = 80;
```

```
const float sadTempThreshold = 20.0; const float sadHumidityThreshold = 60.0; const int  
sadPulseThreshold = 70;
```

```
void detectFeelings(float T, float H, int P) {
```

```
    if (T >= happyTempThreshold && H <= happyHumidityThreshold && P >=  
        happyPulseThreshold) {  
        lcd.print("Feeling: Happy");  
    } else if (T <= sadTempThreshold && H >= sadHumidityThreshold && P <=  
        sadPulseThreshold) {
```

```
    lcd.print("Feeling: Sad");
} else {
    lcd.print("Feeling: Chill");
}

}

void setup() {

    lcd.init();
    lcd.clear();
    lcd.backlight();

}

void loop() {

    DHT.read11(DHTPIN);

    float temperature = DHT.temperature;
    float humidity = DHT.humidity;
    float pressure = analogRead(PULSE_PIN);

    lcd.clear();

    lcd.setCursor(0, 0);
    lcd.print("T:");
    lcd.print((int)temperature);
    lcd.print("C");

    lcd.setCursor(6, 0);
    lcd.print("H:");
    lcd.print((int)humidity);
    lcd.print("%");

    lcd.setCursor(12, 0);
    if(pressure > 550){
        lcd.print("P:");
        lcd.print((int)pressure);
        //lcd.print("hPa");
    } else {
        lcd.print("P:0");
    }
    lcd.setCursor(0, 1);
    detectFeelings(temperature, humidity, pressure);

    delay(1000);

}
```

Detaliere a algoritmului:

- Inițializare și Configurare:

În secțiunea de inițializare, bibliotecile necesare sunt incluse: Wire.h pentru comunicarea I2C, LiquidCrystal\_I2C.h pentru controlul ecranului LCD, dht.h pentru senzorul DHT11 și PulseSensorPlayground.h pentru senzorul de puls. Sunt definite constante pentru pinii unde sunt conectați senzorii (DHTPIN și PULSE\_PIN). Un obiect LiquidCrystal\_I2C este creat pentru a controla ecranul LCD cu adresa I2C 0x3F și un obiect PulseSensorPlayground pentru senzorul de puls. De asemenea, este creat un obiect dht pentru senzorul de temperatură și umiditate.

- Praguri de Senzori:

Pragurile pentru detectarea stării de spirit sunt definite ca constante pentru temperatură, umiditate și puls. Aceste praguri sunt utilizate pentru a determina dacă valorile măsurate indică o stare de "Happy", "Sad" sau "Chill". De exemplu, dacă temperatura este peste 25°C, umiditatea sub 50%, și pulsul peste 80, programul va indica starea de "Happy".

- Funcția detectFeelings

Funcția detectFeelings primește ca parametri temperatura, umiditatea și pulsul, și folosește aceste valori pentru a determina starea de spirit. Dacă toate condițiile pentru starea de "Happy" sunt îndeplinite, afișează "Feeling: Happy" pe ecranul LCD. Dacă sunt îndeplinite condițiile pentru "Sad", afișează "Feeling: Sad". Dacă nici una dintre aceste condiții nu sunt îndeplinite, afișează "Feeling: Chill".

- Funcția setup și loop

În funcția setup, ecranul LCD este inițializat și curățat, iar iluminarea de fundal este activată. În bucla principală loop, senzorul DHT11 citește temperatura și umiditatea. Pulsul este citit de pe pinul analogic A0. Ecranul LCD este curățat și sunt afișate valorile măsurate pentru temperatură și umiditate. Dacă presiunea (care în acest context pare să fie greșit denumită, probabil se referă la puls) este mai mare de 550, este afișată valoarea acesteia, altfel se afișează "P:0". Funcția detectFeelings este apoi apelată pentru a afișa starea de spirit determinată. La sfârșit, codul așteaptă o secundă înainte de a relua citirile și afișările.

## Rezultate Obținute



Primul rând al ecranului LCD afișează trei valori principale:

- Temperatura (T): Este afișată în grade Celsius (C). De exemplu, dacă temperatura este 23 de grade, va apărea ca T:23C.

- Umiditatea (H): Este afișată în procente (%). De exemplu, dacă umiditatea este 45%, va apărea ca H:45%.
- Pulsul (P): În realitate, acest cod citește o valoare analogică de la senzorul de puls, iar dacă această valoare este mai mare de 550, va afișa P:<valoare>. Dacă valoarea este sub 550, va afișa P:0.

Al doilea rând al ecranului LCD afișează starea de spirit detectată în funcție de pragurile prestabilite pentru temperatură, umiditate și puls.

## Concluzii

Acest proiect Arduino integrează cu succes mai mulți senzori și un ecran LCD pentru a monitoriza și afișa date esențiale privind temperatura, umiditatea și pulsul utilizatorului în timp real. Folosind biblioteca Wire pentru comunicație I2C și biblioteci specifice pentru senzorii DHT11 și de puls, proiectul oferă o interfață intuitivă și ușor de înțeles. Prin analiza simplificată a datelor colectate, proiectul poate determina și afișa o stare de spirit generală, oferind astfel utilizatorului o perspectivă rapidă asupra condițiilor de mediu și a stării de sănătate. Acest proiect demonstrează modul în care tehnologia poate fi utilizată pentru a îmbunătăți confortul personal și pentru a monitoriza parametri importanți ai mediului și ai sănătății.

Posibile Îmbunătățiri: Implementarea unui algoritm de filtrare pentru datele de puls pentru a elimina zgomotul și a oferi citiri mai precise, Adăugarea unor LED-uri sau a unui buzzer pentru a emite semnale vizuale sau sonore atunci când sunt detectate condiții critice de mediu sau valori anormale ale pulsului, Implementarea unui sistem de alimentare cu baterii și a unor tehnici de economisire a energiei pentru a face dispozitivul portabil și autonom.

## Download

[arhiva.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/avaduva/cerasela.soare>



Last update: **2024/05/24 16:09**