

# Ceas cu afisaj LCD si 'silent alarm'

## Introducere

Proiectul consta in realizarea unui ceas cu alarma 'siletnioasa'. In loc de o alarma clasica care foloseste un difuzor, la declansarea alarmei ceasul va porni diverse jocuri de lumini pentru a trezi pe utilizatorul. Am pornit de la un ceas pe care l-am primit cand eram mica asemanator cu cel din linkul [1], un ceas desteptator cu ecran LCD pe care afisa mai multe informatii si atunci cand se declansa alarma se ilumina carcasa si se pornea cate o melodie, putand sa selectezi melodia preferata si culoarea pentru alarma.

Asa ca am hotarat sa adaptez ideea ceasului pentru o problema a studentilor care locuiesc in camine si anume folositul unei alarme dimineata fara a trezi si alte persoane, de unde a aparut si numele proiectului 'silent alarm'. Proiectul meu implementeaza functionalitatile de baza ale ceasului dat exemplul anterior, cu o posibilitate mai mare de personalizare a jocului de lumini.

Link: [FlashingBlinkyLights](#)

## Descriere generală

Din punct de vedere fizic ceasul va fi un cub din plexiglas care va avea pe partea din fata un ecran LCD pentru a afisa ora actuala si ora la care este setata alarma, precum si temperatura camerei in care se afla printr-un senzor de temperatura. In interiorul cubului vor fi doua nivele: cel de jos va contine arduino-ul si sezorul de temperatura si cel de sus ledurile pentru jocurile de lumini. Jocurile de lumini si alarma se vor putea seta din laptop care va comunica cu arduino-ul prin USART. Din laptop utilizatorul va putea porni ceasul, seta alarma si jocul de lumini.

### Schema bloc:



## Hardware Design

### Lista componente hardware:

- Arduino Uno
- Led-uri RGB
- Cabluri
- Ecran LCD cu I2C
- Senzor de temperatura

- 8-bit shift register x3
- Breadboard
- Rezistoare
- Cablu UART
- Plexiglas

- Pentru a aprinde LED-uri vor fi folosite timere și întreruperi.
- Pentru a afișa ora actuala, ora alarmei și temperatura camerei pe LCD se va folosi protocolul I2C.
- Alimentarea se va realiza de la laptop
- LED-urile se vor fixa pe PCB sau vor fi lipite direct în carcasa (TBD)
- Laptop-ul va comunica cu Arduino-ul prin cablu folosind UART

### Schema electrica a circuitului:



### Schema:

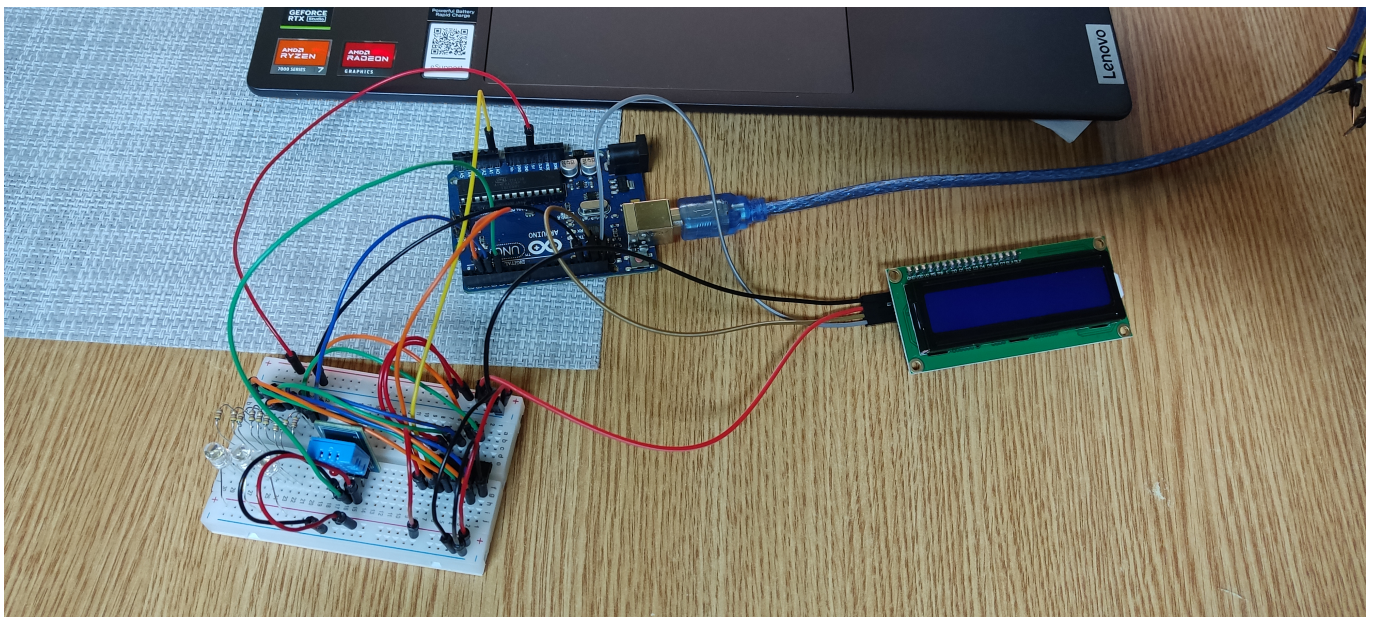


Link către schema în Tinkercad: [Schema proiect](#)

### Montaj fizic:

#### Partea 1:

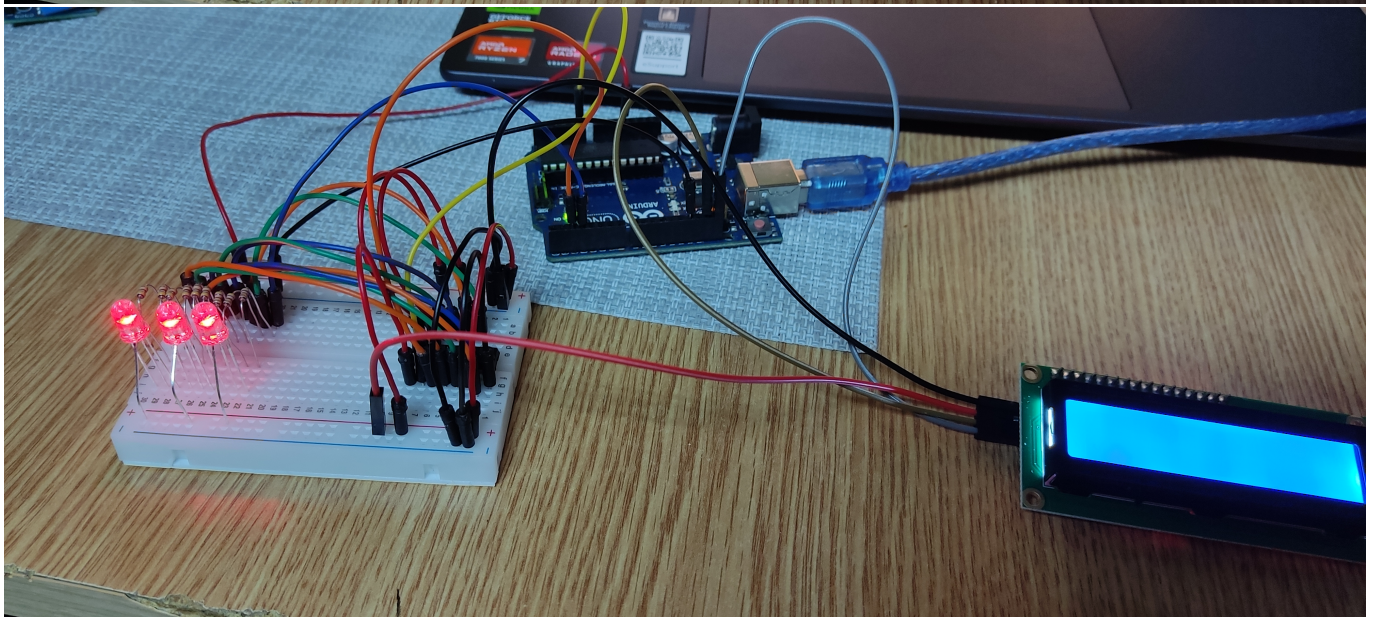
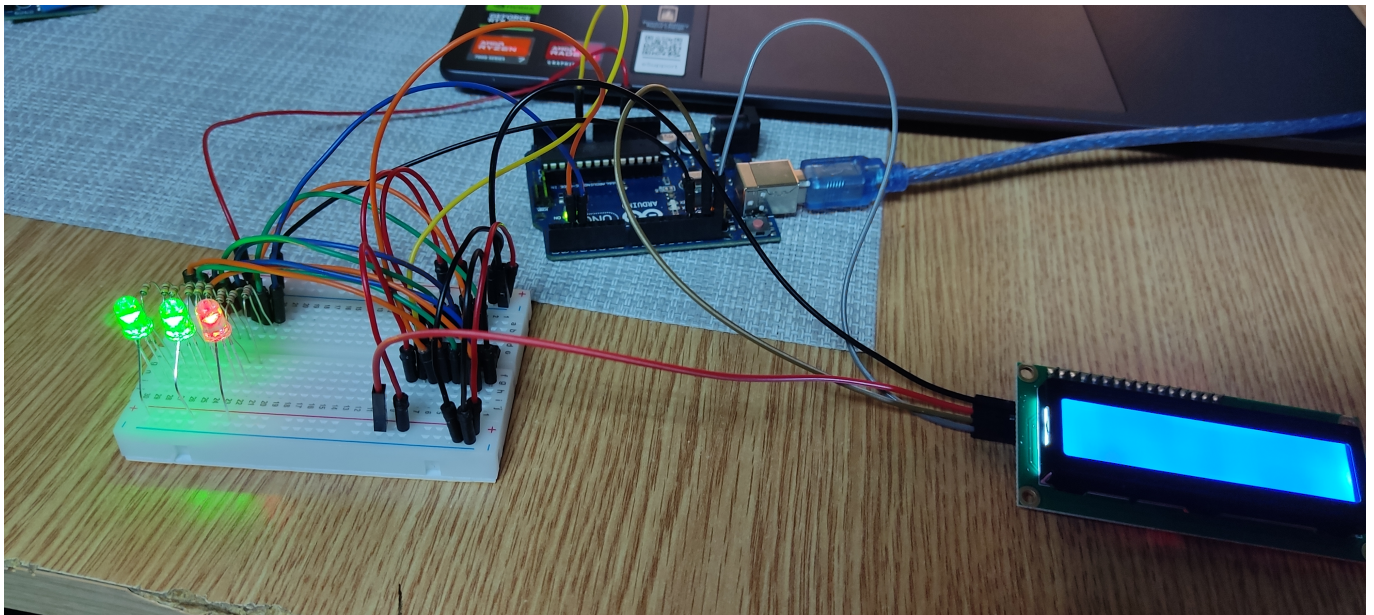
Am conectat ecranul LCD, senzorul de temperatura și cele 3 LED-uri RGB la Arduino.

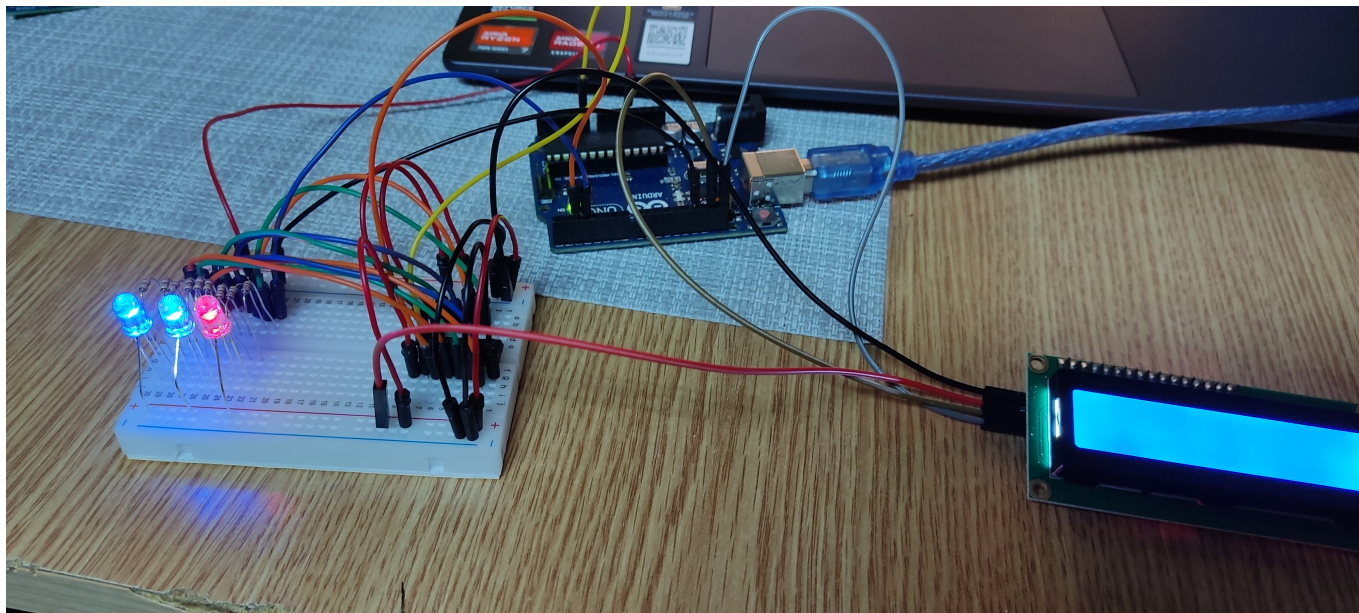


Piniile de pe LCD de GND și VCC i-am conectat la breadboard, cei de SCL și SDA direct la Arduino.

Pentru LED-urile RGB am conectat câte o rezistență de 480Ω la fiecare pin: red, green și blue, după care am conectat piniile RGB ale fiecărui LED la piniile de output ale 8-bit shift register. Am conectat restul

pinilor de pe 8-bit register shift astfel: STROBE la linia de putere de pe breadboard, DATA la pinul A0 de pe arduino, CLOCK la pinul 2 de pe arduino, VSS la ground-ul de pe breadboard, VDD si Output enable la linia de putere. Ultimul pin la celui de-al treilea led (blue) l-am conectat direct la arduino la pinul 3 pentru ca nu am mai avut pini de output pentru registrii pe 8-bit shift register si nu avea rost sa mai folosesc inca unul.





## Software Design

Link pentru repo-ul de GitHub: [GitHub](#)

Mediu de dezvoltare: Arduino IDE

**Biblioteci utilizate:** LiquidCrystal\_I2C.h (pentru LCD-ul cu I2C), DHT.h (pentru senzorul de temperatura DHT11)

### Software design:

Codul se afla in fisierul proiect.ino si contine implementarea pentru jocurile de lumini si afisarea datelor pe LCD de la senzorul de temperatura si a datelor suplimentare pentru ceas si alarma.

Funcția de **setup** se ocupă cu inițializarea componentelor (Led-uri, ecran LCD si senzor).

Pentru **jocurile de lumini** am cate o functie pentru fiecare. Primele doua jocuri de lumini sunt prestabilite (primul un 'trenulet' de lumini si al doilea consta in schimbarea led-urilor simultan intr-o culoare), al treilea este in functie de temperatura camerei: **albastru** pentru temperaturi de mai putin de 20 de grade Celsius, **galben** pentru temperaturi intre 20 si 30 de grade Celsius si **rosu** pentru temperaturi mai mari de 30 de grade Celsius. Am folosit millis() in loc de intreruperi pentru a evita eventualele blocaje cand comunic cu LCD-ul si senzorul. Mai exista functii ajutatoare pentru jocurile de lumini: **setLEDColor** pentru a facilita selectarea led-ului pe care dorim sa il aprindem si culoarea acestuia si o **matrice** in care am definit cateva culori.

Pentru **afisarea datelor pe ecranul LCD** am o functie ajutatoare care preia inputul de la utilizator prin seriala. Am initializat ceasul sa porneasca de la 0:0:0, iar utilizatorul poate sa dea urmatoarele comenzi:

- Co:m:s → C urmat de ora:minut:secunda, pentru a seta ora actuala a ceasului.
- Ao:m:s → A urmat de ora:minut:secunda, pentru a seta alarma la ora dorita. Aceasta se va afisa pe ecran pe al doilea rand.

- Mnr\_mod → M urmart de numarul modului pentru jocul de lumini.
- S → pentru a opri alarma (daca nu este oprita de utilizator, alarma se va opri dupa 60 de secunde).
- R → pentru a reseta alarma (o sa dispara si de pe ecran alarma).

Codul pentru ceas l-am facut in loop si tot aici citesc temoeratura de la senzor folosind biblioteca DHT.h

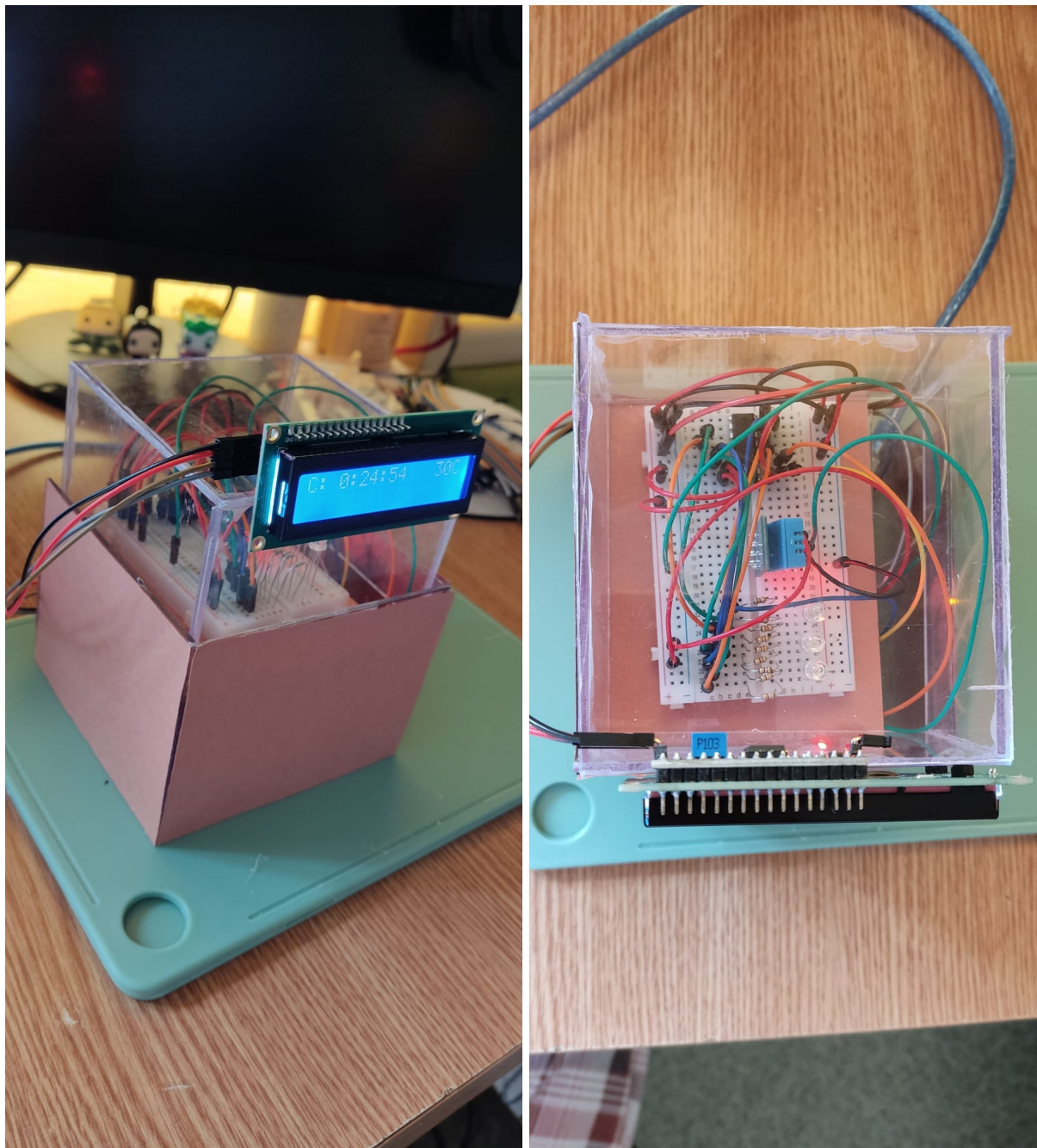
Pe ecran se afiseaza ora curenta pe primul rand in stanga, temperatura pe primul rand in dreapta si alarma pe al doilea rand.



## Rezultate Obținute

Videoclip prezentare proiect: [Video Proiect PM](#)

Ceasul respecta planul inintial pentru hardware. Carcasa de plexiglas este formata din doua bucati, in partea inferioara se afla arduino-ul, in partea superioara breadboard-ul cu senzorul de temepratura si LED-urile, iar pe carcasa este ecranul LCD.



Senzorul de temperatura, desi scria ca este DHT11 nu functiona utilizand biblioteca pentru el, dar a mers folosindu-l ca fiind DHT22. Temperatura are o marja de eroare destul de mare, posibil un defect din fabrica, dar tinand cont ca nu trebuia sa fie exacta nu a reprezentat un impediment.

Se pot adauga din cod noi jocuri de lumini sau se poate extinde codul pentru a lasa utilizatorul sa isi creeze propriile jocuri de lumini.

## Concluzii

Am folosit laboratoarele: 1.UART (pentru comunicarea cu laptop-ul), 2.Intreruperi (am folosit intreruperi pentru a realiza ceasul), 6.I2C (pentru LCD-ul cu I2C).

A fost interesant de implementat atat partea software, cat si cea hardware. A durat putin pana am

facut schita carcasei ca sa pot sa trag toate firele necesare de la arduino la breadboard si la lcd, montajul electric nu a durat foarte mult pentru ca am urmat schema electrica. Codul ar putea fi imbunatatit cu o modalitate prin care utilizatorii sa isi poata crea singuri propeile jocuri de lumini fara sa fie necesar sa se modifice codul.

## Download

Arhiva proiect: [proiect\\_pm\\_ms.zip](#)

## Bibliografie/Resurse

### Resurse folosite:

- [Datasheet 8-bit register shifter](#)
- [Tutorial shift-and-store bus register](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/mdinica/mihaela.stefan1411>



Last update: **2024/05/27 14:23**