


# Corina RADU - LCD Multifunctional Clock

Autorul poate fi contactat la adresa: **Login pentru adresa**

## Introducere

Scopul acestui proiect este realizarea unui **ceas digital**, care, pe langa afisarea orei si datei curente, sa aiba functiile de **alarma** si de **afisare a temperaturii** ambientale. Am ajuns la aceasta idee cautand o modalitate de a integra ceea ce am invatat pana acum la PM intr-o aplicatie, care, desi poate fi considerata simpla ca idee, imi va oferi satisfactia de a fi reusit sa creez un sistem embedded pe cont propriu. Cat despre utilitate, o astfel de aplicatie este perfecta pentru situatiile in care telefonul ramane fara baterie. 

## Descriere generală



Dispozitivul va prelua data si ora exacta de la un **PC** printr-o *interfata seriala*, nemaifiind astfel nevoie de setarea manuala a acestora.

**Butoanele** vor avea urmatoarele roluri:

- activare/dezactivare alarma
- setarea orei pentru alarma
- setarea minutului pentru alarma
- oprirea alarmei dupa ce porneste buzzer-ul (butonul de STOP)

**Senzorul de temperatura** va transmite semnale analogice catre convertorul analog-digital integrat in microcontroller.

**Buzzer-ul** va reda o melodie atunci cand ora pre-setata va fi aceeasi cu ora curenta. Melodia va fi redata timp de 1 minut, dupa care se va opri singura (optiunea de SNOOZE). Dupa alte 2 minute, melodia va "porni" din nou, din acest moment putand fi oprita doar prin apasarea butonului STOP.

**LCD-ul** are rol de afisare al orei exacte, datei curente, temperaturii din camera. La apasarea unui buton, se va afisa starea alarmei (ON/OFF) si ora setata (00:00 daca alarma este in starea OFF, ora setata de utilizator daca alarma este in starea ON).

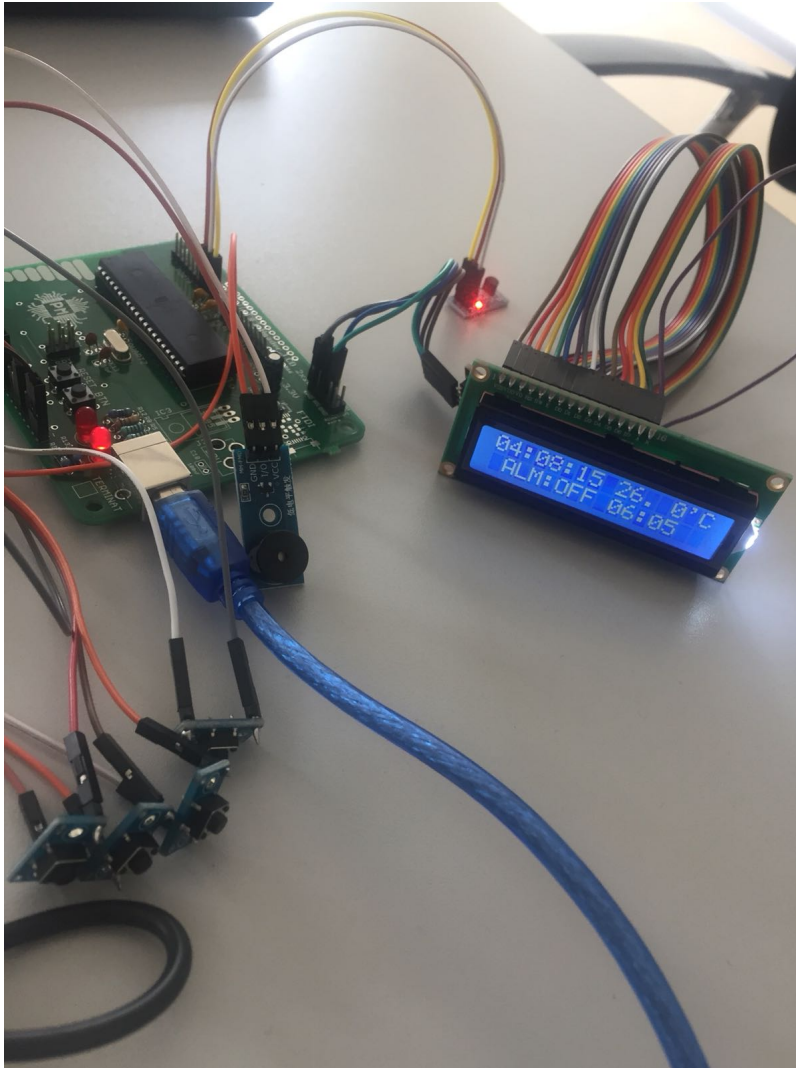


- Configurarea butoanelor:
  - prin butoanele PB0 si PB1 se seteaza ora, respectiv minutul pentru ceas
  - prin butoanele PB2 si PB3 se seteaza ora, respectiv minutul pentru alarma
  - la apasarea simultana a butoanelor PB0 si PB1, se va schimba modul de afisare a temperaturii (Celsius → Fahrenheit sau invers)
  - la apasarea simultana a butoanelor PB2 si PB3, se va schimba starea alarmei (ON → OFF sau invers)
- Configurarea buzzer-ului, care va reda un sunet pana cand va fi oprit prin buton
- Citirea temperaturii, cu ajutorul functiilor din biblioteca `ds18b20.h`
- Afisarea orei exacte, a orei si starii alarmei, si a temperaturii pe ecranul LCD. Deoarece am utilizat un adaptor I2C pentru LCD, am avut nevoie de 3 biblioteci: `i2cmaster.h`, `pcf8574.h` si `lcdpcf8574`.

## Rezultate Obținute

Produsul final este functional, inasa, din cauza achizitonarii unor componente nepotrivite, am fost nevoita sa renunt la cateva dintre specificatiile initiale ale proiectului.

- In loc de a folosi interfata seriala pentru a seta ora exacta, am facut acest lucru din butoane, din cauza proastei functionari a cablului convertor USB-UART. Nemaifiind nevoie de o legatura directa cu laptopul, am folosit pentru alimentare o baterie externa de telefon.
- Deoarece buzzerul achizitionat de mine nu a fost pasiv, ci activ, nu am putut reda o melodie, dupa cum mi-am propus initial, ci un singur sunet, care porneste si se opreste pana cand utilizatorul apasa un buton.



## Concluzii

Pentru mine proiectul a fost challenging, chiar daca aplicatia in sine ar fi putut fi usoara. Primele dificultati au aparut la alegerea componentelor - nu am avut o idee clara de la inceput referitoare la ce trebuie sa cumpar, cum sa conectez componentele intre ele, asa ca am cumparat multe componente "orbeste", dandu-mi seama dupa aceea ca mi-ar fi fost mai util altceva.

Un alt obstacol important a fost calitatea scazuta a cablului convertor USB-UART, despre care am aflat abia pe ultima suta de metri ca nu este original, ci o copie chinezeasca. Din acest motiv am intampinat in prima faza probleme cu conectarea USB-ului la PC (dispozitivul USB nu era recunoscut), iar dupa ce am rezolvat aceasta problema am remarcat faptul ca datele transmise prin seriala erau corupte, cu toate ca am respectat regulile si configuratiile studiate la laborator. In consecinta, am fost nevoita sa modific aceasta configuratie si sa optez pentru un ceas normal, controlat din butoane, alimentat de o baterie externa.

Acest proiect, in concluzie, m-a invatat doua lucruri importante: cum sa imbin si sa integrez hardware-ul si software-ul asa incat o aplicatie sa fie functionala, si faptul ca *time management* este un concept esential.

## Download

[Codul sursa](#)

## Jurnal

12-19 aprilie 2018: Alegerea proiectului

18 aprilie 2018: Primirea placutei de baza

21 aprilie 2018: Descrierea generala a proiectului (**milestone 1**)

3 mai 2018: Lipirea pieselor pe placa de baza (**milestone 2**)

6 mai 2018: Schema electrica (**milestone 3**)

7-16 mai 2018: Research si achizitionarea pieselor aditionale necesare (senzor de temperatura, LCD, adaptor I2C, buzzer, modul RTC, module cu buton, fire, convertorul USB-UART etc)

21-23 mai 2018: Conectarea componentelor hardware

23 mai 2018: Documentatia provizorie (**milestone 4**)

23-24 mai 2018: Implementarea software-ului

24 mai 2018: Obtinerea produsului final

25 mai 2018: Documentatia finala

## Bibliografie/Resurse

- Documentația în format [PDF](#)
- [Laboratoare PM](#)
- [Biblioteca i2cmaster.h](#), creata de Peter Fleury
- [Bibliotecile pentru LCD conectat la I2C](#), create de Davide Girone

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2018/cpetrisor/alarm-clock>



Last update: **2021/04/14 15:07**