

# Alarm Clock LCD

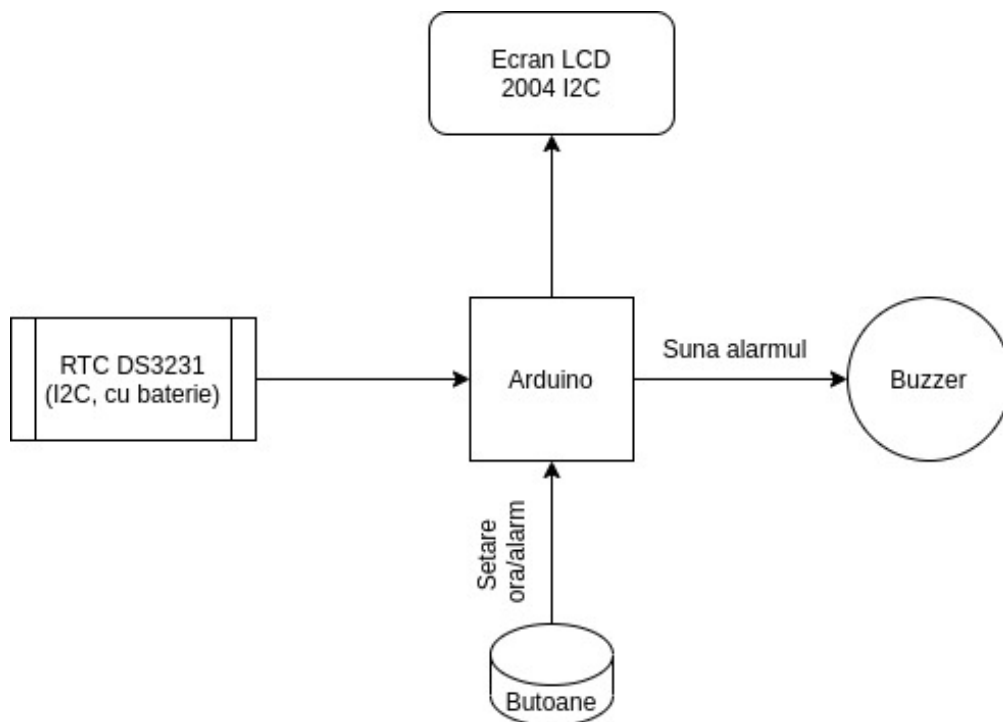
Autor: Dimovski Kirjan

Grupa: 332CB

## Introducere

Un ceas cu alarmă cu un LCD care utilizează un RTC DS3231 pentru a ține evidența timpului.

## Descriere generală



Arduino citește din modul, convertește datele și le trimite display-ului. Un buzzer porneste dacă timpul alarmei a ajuns la timpul curent. Pentru a opri alarma se apasă butonul din mijloc. Pentru că avem o baterie în modulul, acesta va ține evidența timpului, chiar dacă nu este alimentat de o conexiune USB.

## Hardware Design

- Arduino UNO R3

- LCD 2004 I2C (0x27)
- RTC DS3231 (0x68)
- Fire
- Breadboard
- Buzzer
- 3 x Rezistente
- 4 x Butoane



Am folosit si rezistore de pull up pentru ca am conectat 2 despozitive la I2C.

## Software Design

Dezvoltarea a fost facuta folosind mediul de dezvoltare [Arduino IDE](#)

## Bibliotecile externe folosite

- Wire.h: Pentru protocolul I2C.
- LiquidCrystal\_I2C.h: Pentru utilizarea mai usoara al ecranului.

## Variabile globale

```
bool alarm1_status, alarm2_status;
char Time[] = " : : ",
    calendar[] = " / /20 ",
    alarm1[] = "A1: : :00", alarm2[] = "A2: : :00",
    temperature[] = "T: . C";
byte i, second, minute, hour, day, date, month, year,
    alarm1_minute, alarm1_hour, alarm2_minute, alarm2_hour,
    status_reg;
```

## Functii

### setup()

Pinii de la 9 pana la 11 sunt setați ca intrări pentru butoane, utilizând rezistențele pull up încorporate

ale Arduino, iar pinul 12 este dezactivat (alarm\_pin) cu noTone (). Ecranul LCD este inițializat (lcd.begin (20, 4)) și pinul 2 este setat ca pinul de întrerupere, utilizând alarmă ca procedură de întrerupere.

## Alarm()

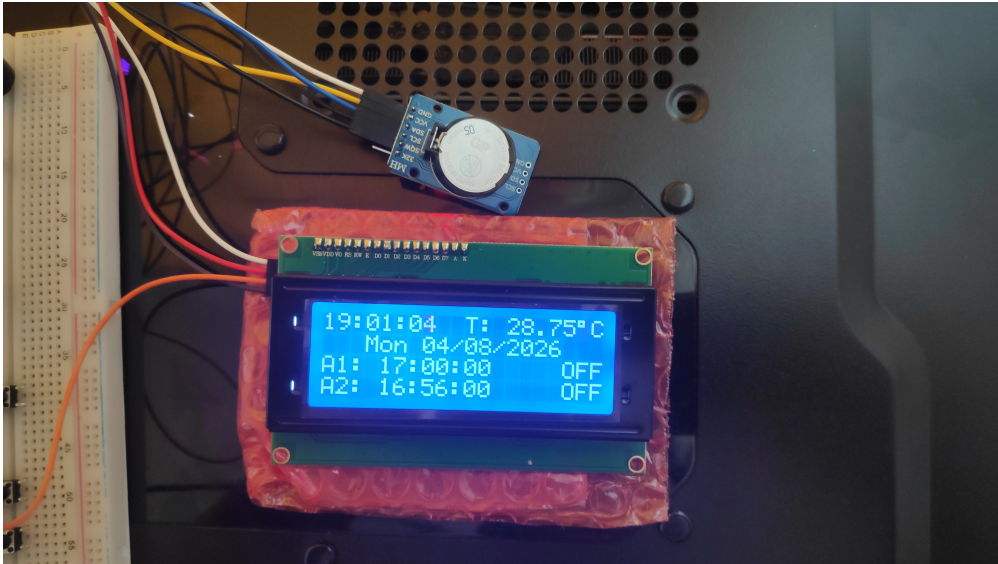
Porneste buzzerul la o frecvența de 1000. Folosit în setup() cu attachInterrupt().

## loop()

Verificăm dacă un buton a fost apăsat. Dacă e, edităm timpul sau alarmul și scriem pe modulul. Altfel, citim de la modulul și afișăm pe display-ul.

## Rezultate Obținute





[Link to YouTube](#) pentru demo-ul.

## Concluzii

Proiectul a fost foarte interesant de implementat. M-am familiarizat cu protocolul I2C, chiar dacă uneori a fost cam dificil. Cea mai mare parte a muncii a vizat formatarea de pe ecran, cu care vă puteți distra foarte mult dacă va puneti mintea la contribuție.

## Download

[arhiva\\_kd332cb.zip](#)

## Bibliografie/Resurse

- [RTC DS3231 Datasheet](#)
- [LiquidCrystal I2C GitHub](#)
- [How many Devices can you connect to the I2C Bus?](#)
- [Arduino UNO pinout](#)
- [Laboratorul 1](#)
- [Laboratorul 6](#)
- [YouTube 1](#)
- [YouTube 2](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/alazar/alarmclocklcd>



Last update: **2021/05/31 17:35**