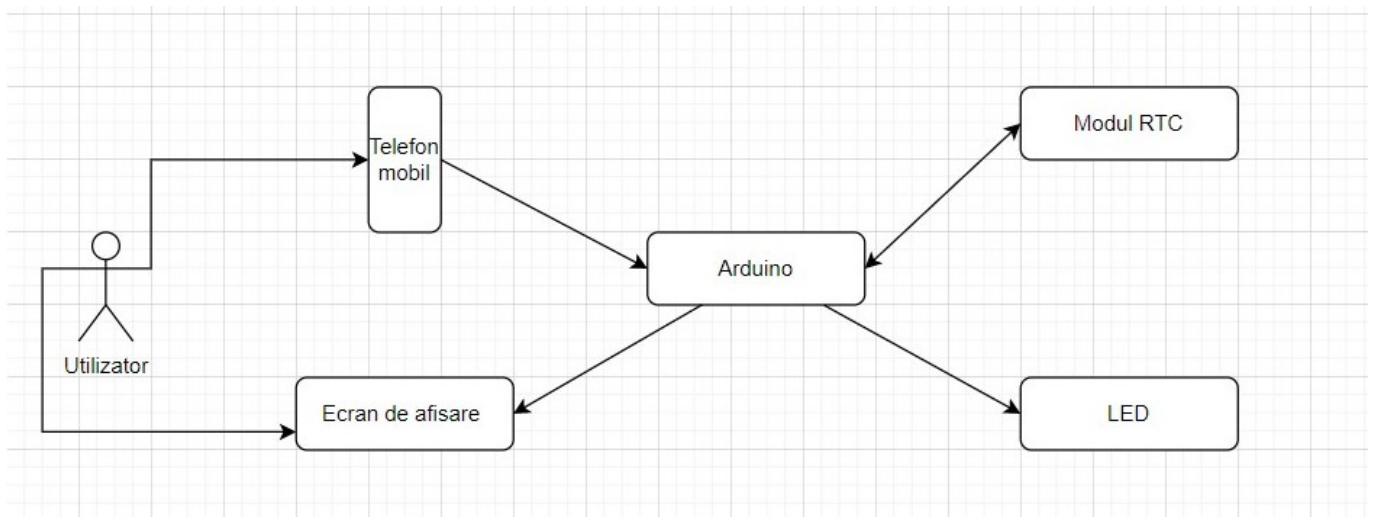


Ceas luminos

Introducere

- In cadrul acestui proiect imi propun sa fac un ceas digital cu alarma de tip rasait. Acesta incepe sa lumineze treptat cu o lumina calda pentru a simula un rasarit de soare, oferind o alternativa mai naturala in locul unui ceas obisnuit. In plus, acesta poate fi conectat prin Bluetooth la o aplicatie mobila pentru a fi controlat. In utilizarea acestui ceas, utilizatorii pot îmbunătăți calitatea somnului și se pot simți mai odihniți și mai energici pe tot parcursul zilei.

Descriere generală

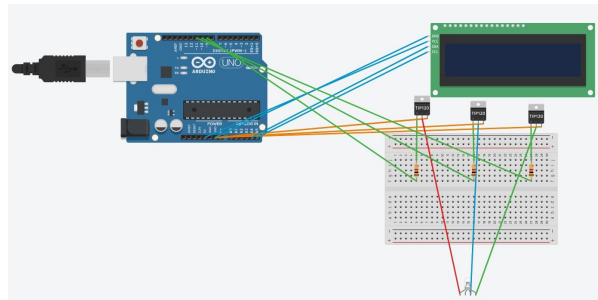


- Utilizatorul poate seta ora pentru alarma din aplicatia mobila sau cu ajutorul butoanelor de pe ceas.
- Ceasul va incepe sa lumineze treptat cu o jumătate de ora înainte de ora selectata, pana ajunge la intensitatea maxima.
- Acesta poate fi configurat si sa inceapa sa bazeie pe la finalul celor 30 de minute, daca doar lumina nu este suficienta.

Hardware Design

- Arduino Uno
- Ecran LCD
- Breadboard
- Modul RTC
- 3 Tranzistoare TIP122

- Banda LED
- Glob de plastic
- Rezistente 1k
- Butoane
- Modul Bluetooth HC-05
- Alimentator 12V
- Potentiometru



Software Design

```
#include <Wire.h> #include <LiquidCrystal_I2C.h> #include <DS3231.h> #define red 9 #define  
green 10 #define blue 11
```

```
int brightness = 50; int fadeAmount = 5; LiquidCrystal_I2C lcd(0x27,16,2);
```

```
Init the DS3231 using the hardware interface DS3231 rtc(SDA, SCL); alarma int h_alarm= 22; int  
m_alarm = 3;
```

```
void setup() {
```

```
  lcd.init();// initialize the lcd  
  lcd.backlight();//open the backlight  
  rtc.begin();  
  //rtc.setTime(10,53,0);  
  //rtc.setDate(29,05,2023);
```

```
}
```

```
void loop() {
```

```
  lcd.setCursor(0, 0);  
  lcd.print("TIME ~ " );  
  lcd.print(rtc.getTimeStr());  
  lcd.setCursor(0, 1);  
  lcd.print("DATE ~ " );  
  lcd.print(rtc.getDateStr());  
  delay (1000);  
  lcd.clear();  
  if(h == h_alarm && m == m_alarm)  
  {
```

```
    analogWrite(red, brightness);
    analogWrite(green, brightness);
    analogWrite(blue, brightness);
    brightness = brightness + fadeAmount;
    if(brightness <=0 || brightness >= 255)
    {
        fadeAmount = -fadeAmount;
    }
    delay(500);
}
}
```

Descrierea codului aplicației (firmware):


- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

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/prj2023/daniel/ceas-luminos>



Last update: **2023/05/29 21:53**