


# Alarma securitate

## Introducere

**Alarma securitate** este o alarma cu display care se declanseaza daca detecteaza miscare, putand fi dezactivata doar printr-o telecomanda.

## Descriere generală

La inceput alarma e pe off si poate fi activata prin telecomanda prin plimbarea printr-un meniu  afisat pe display-ul LCD 12C. In plus, tot din acest meniu se poate seta timer-ul la 5 sec sau 10 sec. Acest timer e folosit pt. a sti dupa cate secunde se porneste senzorul de miscare dupa alegerea activarii alarmei. Daca senzorul de miscare detecteaza ceva, o sa se auda un sunet ca de alarma care poate fi oprit doar din telecomanda iar alarma se va dezactiva, urmand ca apoi sa fie reactivata tot din telecomanda daca se doreste. In plus mai sunt 3 LED-uri, dintre care unul rosu care indica daca alarma e pe off, unul verde care indica daca e pe on si unul galben care indica cand alarma poate primi comenzi de la telecomanda intervalul fiind de minim 1 sec intre 2 comenzi succesive.

## Hardware Design

 *Lista piese:*

- Arduino Uno
- LCD 12C
- Senzor miscare
- Buzzer
- 3xLED
- Telecomanda
- Senzor
- Breadboard

## Software Design

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 plănuieți să le implementați
- (etapa 3) surse și funcții implementate

- Mediu de dezvoltare: Arduino IDE
- Librării și surse 3rd-party: IRremote.h, LiquidCrystal\_I2C.h

## Detalii de implementare

În `setup()` am configurat toți pinii pe de care aveam nevoie și am configurat întreruperile de tip timer. Pentru configurările întreruperilor de tip timer m-am folosit de [laboratorul 3 de PM](#). În codul întreruperii, în funcția **ISR()** am o var. **volatile long count** pe care o cresc mereu.

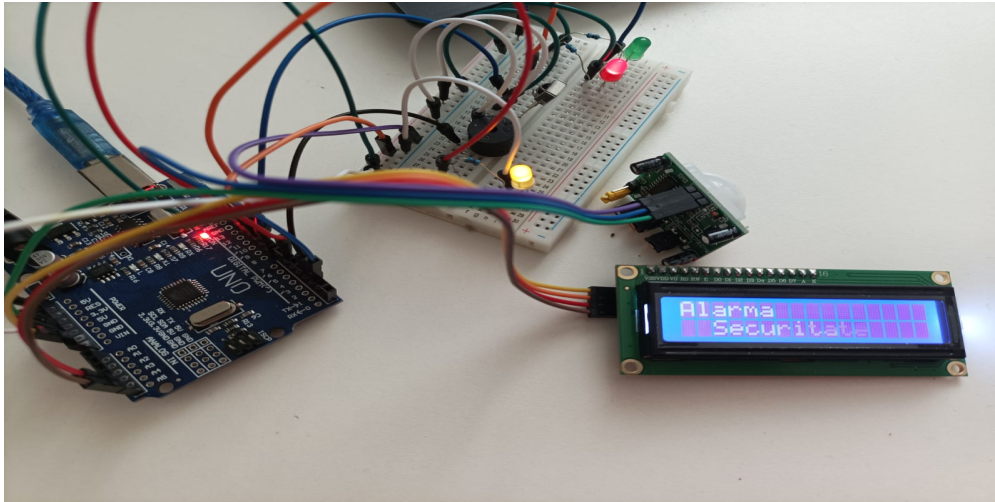
Pentru a ști ce tastă apasă pe telecomandă m-am folosit de **IrReceiver.decodedIRData.command** care îmi întoarce un număr mai mare ca zero pentru fiecare tastă în parte, după care m-am definit cu define toate tastele cu numărul comenzii dat de senzorul IR cu define (`#define TASTA_OK 28`). Când apăsăm o dată pe o tastă de cele mai multe ori era interpretată de mai multe ori lucru ce ducea la alegerea unor opțiuni nedorite pe meniul afișat pe ecranul LCD, lucru ce a trebuit rezolvat prin folosirea unui timer, astfel trebuie să așteptăm 1 sec între 2 comenzi succesive.

Pentru afișarea meniului pe display m-am folosit de un switch case pe mai multe stări, iar în funcție de ce tastă e apăsată de pe telecomandă, se schimbă var **int STATE = 0** cu o altă stare. Pentru a afișa text pe ecran m-am folosit de funcțiile **lcd.clear()**, **lcd.setCursor()** și **lcd.print()**.

Când se apasă tastă OK STATE ia valoarea lui STATE\_4, iar var **int flag\_alarma = 0** devine 1, astfel se intră într-un if statement unde se verifică cu **digitalRead(PIN\_TO\_SENSOR)** dacă valoarea pe pin e LOW sau HIGH. Dacă e HIGH înseamnă că a fost detectată mișcare iar var **int alarma = 0** devine 1, care duce la apelarea funcției **sound()**. Pe pin-ul PIN\_TO\_SENSOR e conectat senzorul de mișcare HC-SR501. După ce se pornește alarma prin selectarea opțiunii de ON, nu se verifică imediat dacă senzorul a detectat mișcare și abia după **alarm\_timer** sec, care inițial e setat pe 5 doar poate fi crescut la 10 prin selectarea unei opțiuni pe display.

Funcția **sound()** scrie valori de HIGH, urmate de valori de LOW prin **digitalWrite(buzzer)** la intervale de 1 sec pentru crearea sunetului de alarmă, **const int buzzer = 3** e pin-ul pt buzzer.

## Rezultate Obținute



Care au fost rezultatele obținute în urma realizării proiectului vostru.

Video cu functionalitatile: [Alarma securitate](#)

## Concluzii

Nu poti folosi functia tone() impreuna cu biblioteca IRremote deoarece, ambele lucreaza pe acelasi timer lucru care duce la incapacitatea senzorului IR de a mai procesa comenzi. Eu avand nevoie doar de un sunet care sa sune a alarma am putut rezolva acest lucru prin scrierea pe pin-ul pe care e conectat buzze-ru semnale de HIGH sau LOW, evitand astfel utilizarea functiei tone(). O alta metoda era schimbarea timer-ului pe care functioneaza biblioteca IRremote.

## Download

Codul sursa → [Arhiva](#)

## Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

## Resurse Software

- [Motion sensor HC-SR501](#)

- [LCD 12C](#)
- [IR sensor](#)
- [Timer](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/daniel/alarma-securitate>



Last update: **2023/05/28 18:01**