

# Security Smart Alarm

## Introducere

Proiectul este un **sistem de securitate smart** care utilizează senzori ultrasonici și PIR pentru a detecta mișcarea într-o zonă monitorizată. Odată ce mișcarea este detectată, un microcontroller activează un ecran LCD pentru a afișa un mesaj de avertizare și pornește un semnal sonor deranjant. Alarma poate fi activată/dezactivată din telefon prin intermediul unui modul bluetooth.

**Scopul proiectului** este de a crea o alarmă care să ofere proprietarului posibilitatea de a monitoriza și proteja zona în timp real, oferind informații relevante despre evenimentele detectate.

## Descriere generală

O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.



**Senzorul Ultrasonic** detectează obiectele sau persoanele prin emisia de unde sonore și măsurarea timpului de întoarcere a acestora. **Senzorul PIR HC-SR501** detectează mișcarea prin schimbările de radiații infraroșii emise de obiectele în mișcare. **Microcontrollerul** preia datele de la senzori, controlează afișajul LCD și gestionarea comunicațiilor Bluetooth/Wi-Fi. **Ecranul LCD** afișează informații despre starea sistemului și evenimentele detectate. **Modul Bluetooth** permite proprietarului să activeze sau să dezactiveze oricând alarma prin intermediul telefonului.

### Laboratoare folosite:

1. USART: <https://ocw.cs.pub.ro/courses/pm/lab/lab1-2023> (Folosesc USART pentru debug și pentru analizarea real-time a datelor înregistrate de senzori)
2. I2C: <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2023-2024> (Folosesc I2C în comunicarea cu ecranul LCD)
3. ADC: <https://ocw.cs.pub.ro/courses/pm/lab/lab4-2023-2024> (Folosesc ADC pentru senzorul de proximitate, în calcularea valorilor și normalizarea lor <partea de preprocesare a valorilor generate>)

### Elemente de noutate ale proiectului

Fata de alte proiecte asemănătoare, am folosit mai mulți senzori de același tip, sau diferiți și am creat o macheta personalizată (o situație reală) pentru o casă pe un singur nivel, cu două camere și un hol. Am împartit senzorii astfel încât fiecare dintre ei să detecteze mișcarea pentru o anumită arie.

# Hardware Design

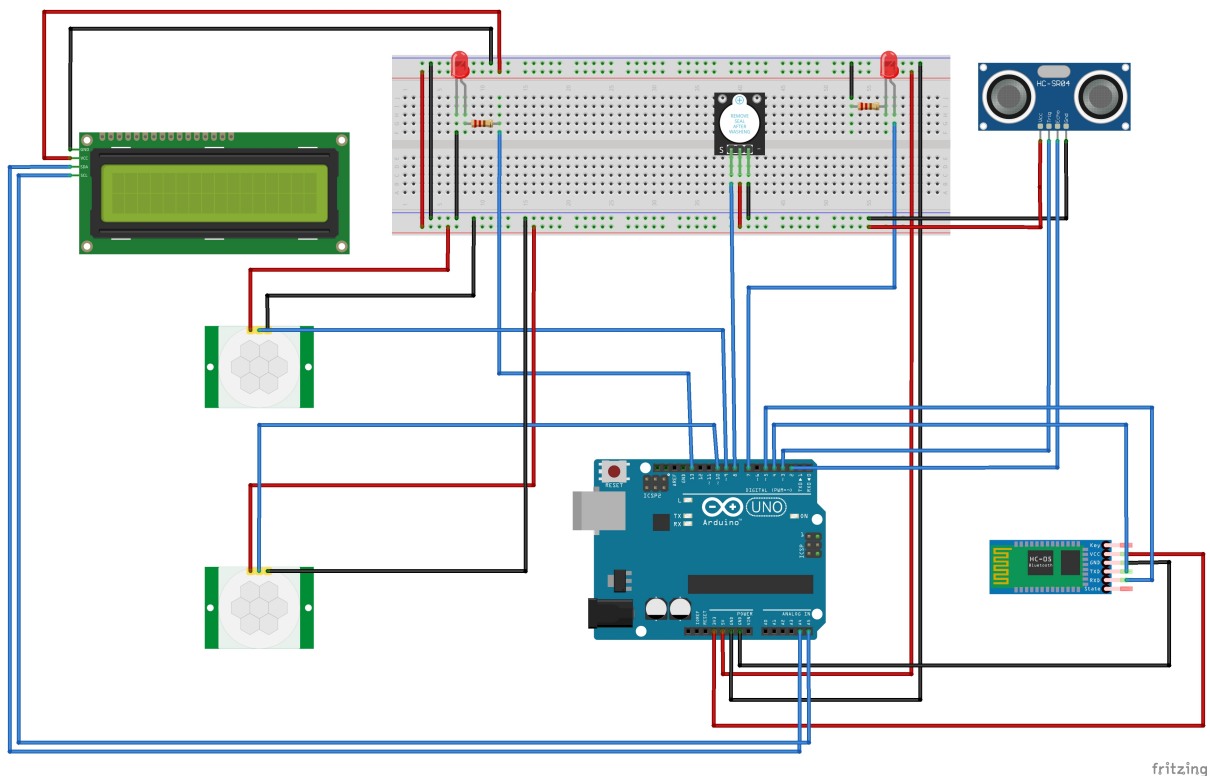
Aici puneți tot ce ține de hardware design:

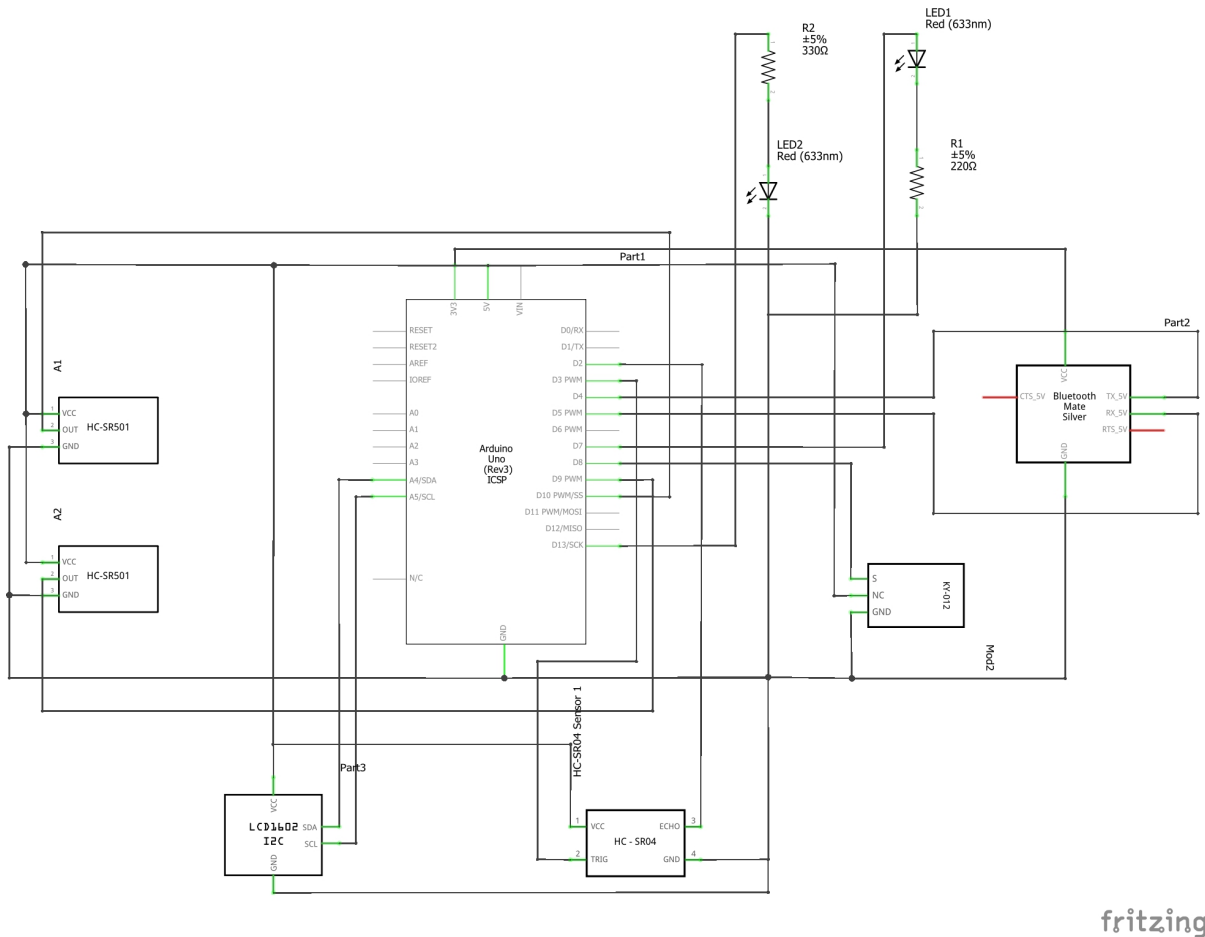
- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)
- diagrame de semnal
- rezultatele simulării

## Listă de piese:

- Senzor Ultrasonic HC-SR04
- 2x Senzor PIR HC-SR501
- Microcontroller
- Ecran LCD
- Modul Bluetooth
- Breadboard
- Buzzer activ

## Schema electrică





## Calibrarea senzorilor și a modulelor

În cadrul proiectului am folosit 3 senzori, doi de mișcare și unul de proximitate. Pentru a folosi cum trebuie senzorii a trebuit să îi reglez cu ajutorul potenciometrelor existente pe ei. În plus la senzorul de proximitate am făcut câteva modificări asupra valorilor citite direct în cod.

- **Senzorii PIR HC-SR501:** Senzorul poate înregistra mișcarea obiectelor într-o rază de până la 6 metrii, cu un unghi de până la 110 grade. Astfel că amplasat undeva deasupra, spre exemplu pe tavanul încăperii, senzorul poate să înregistreze mișcarea din întreaga încăpere (fie ea în formă dreptunghiulară, pătratică sau circulară). Pe partea din spate a senzorului se află două potenciometre, unul pentru reglarea sensibilității acestuia la mișcare și altul pentru timpul de așteptare până la următoarea depistare a mișcărilor. Am modificat puțin valoarea de delay întrucât era setată pe la aproximativ 3 minute, iar senzorul depista foarte greu mișcărilor. Acum senzorul este setat să citească o nouă valoare a mișcării la fiecare 0.3 secunde. (Intervalul de timp de hold este 0.3 secunde, 5 minute)

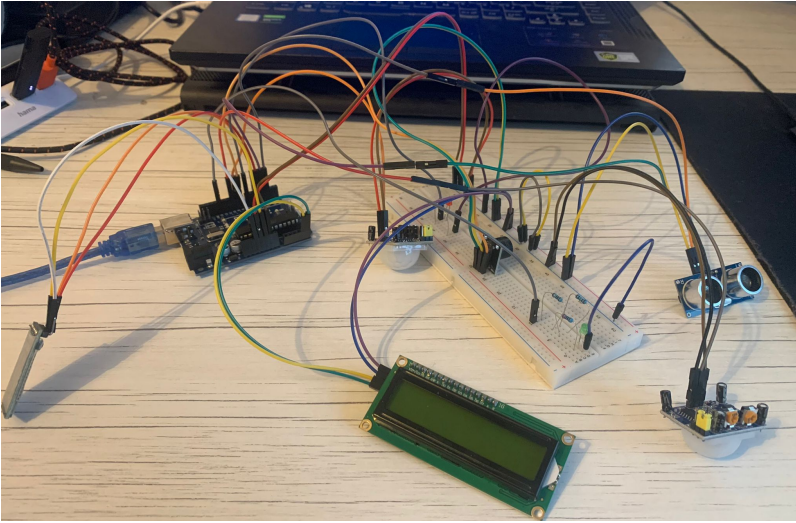
- **Senzorul de proximitate:** Acesta citește încontinuu valori până la cel mai apropiat obstacol din fața lui, desigur dacă acesta există pe o distanță cuprinsă între 2 centimetrii și 4 metrii. Modificările aduse asupra acestui senzor sunt doar pentru normalizarea valorilor citite, pentru a putea face verificările mai ușor.

- **Ecranul LCD:** Pentru acest ecran a fost nevoie să introduc o nouă bibliotecă în cod și să depistez cu ajutorul unui cod găsit pe internet adresa modului (codul poate fi găsit aici: <https://playground.arduino.cc/Main/I2cScanner>). Apoi a trebuit să ajustez potenciometrul de pe partea anterioară a ecranului pentru a se afișa mesajele transmise către acesta.

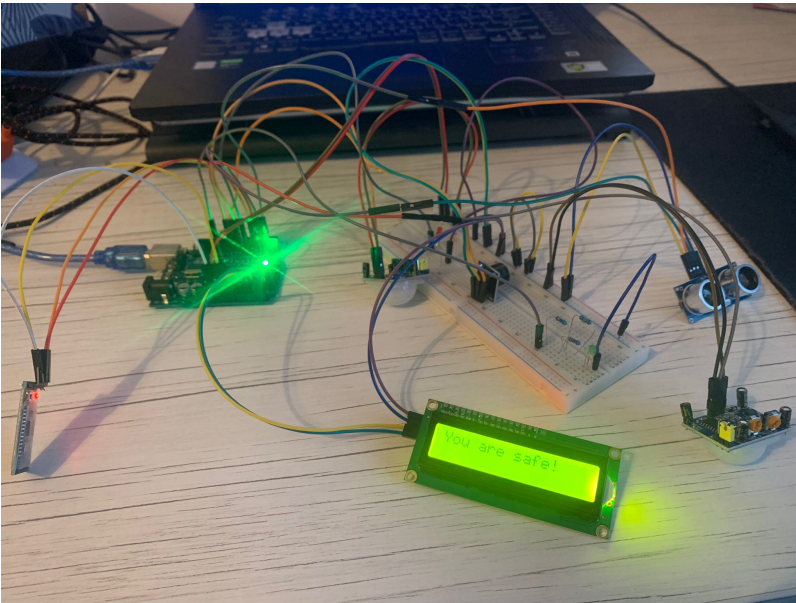
- **Modulul Bluetooth:** Pentru acest modul am schimbat sursa de alimentare, pentru a nu îl arde, de

la 5v la 3.3v și am instalat pe telefon o aplicație (Bluetooth Serial Terminal) pentru a mă putea conecta la el. A fost nevoie și de adăugarea în cod a unei noi biblioteci specifică modulului.

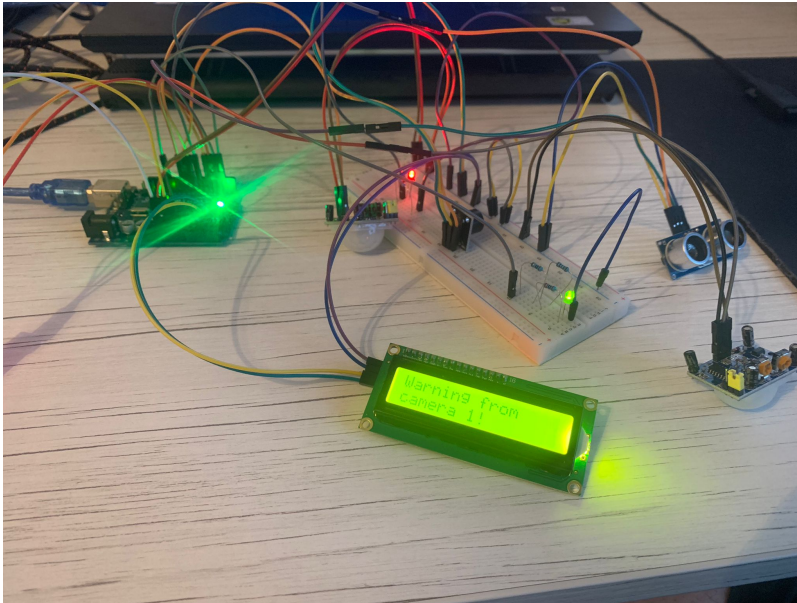
### Poze cu circuitul



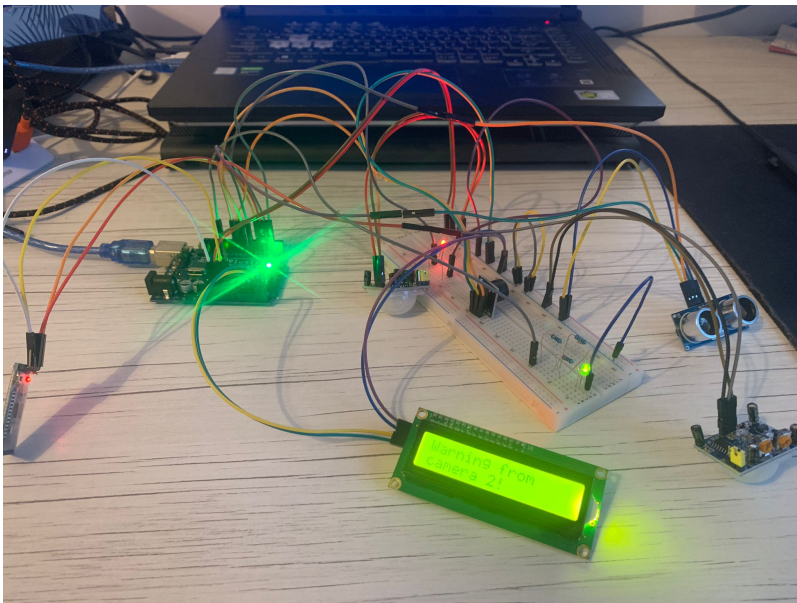
Circuitul conectat la alimentare, dar fără să fie pornită alarma (se poate observa că ledurile sunt stinse, iar senzorii nu vor detecta niciun fel de mișcare).



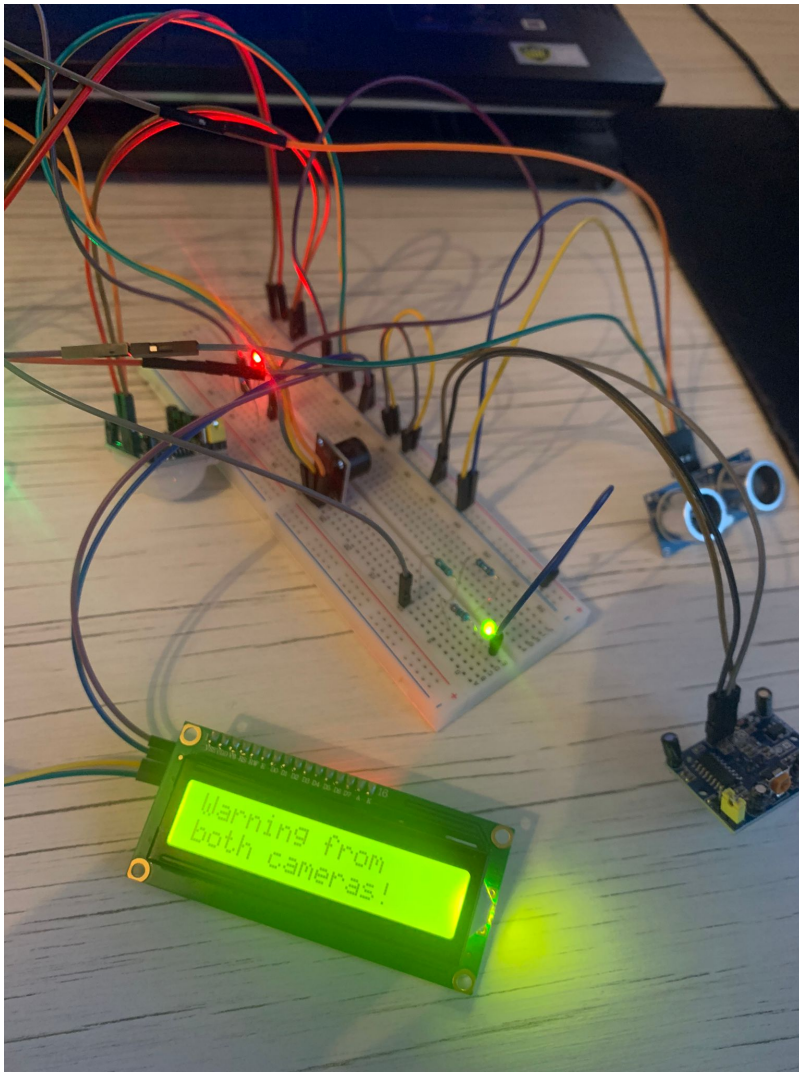
Senzorul 1, amplasat în prima cameră detectează mișcare:



Senzorul 2, amplasat în a doua cameră detectează mișcare:



Ambii senzori, din ambele camere detectează mișcare:



Proiect în varianta finală:



**Demo produs final** [Proiect PM](#)

## Software Design

## Mediu de dezvoltare

Pentru proiectarea placutei de Arduino Uno R3 am folosit Arduino IDE, integrand in cod diverse librarii si functii ale acestora.

### Librarii

- `Wire.h` : Aceasta este o librerie standard pentru comunicatia I2C pe Arduino, folosita pentru a interactiona cu dispozitivele conectate pe magistrala I2C.
- `LiquidCrystal_I2C.h` : O librerie pentru controlul afisajelor LCD prin intermediul protocolului I2C, oferind functionalitati pentru setarea cursorului, stergerea ecranului si afisarea de text. Am folosit-o pentru a afisa diverse mesaje pe ecranul LCD.
- `SoftwareSerial.h` : Permite utilizarea unor pini digitali pentru comunicatie seriala, oferind posibilitatea de a crea porturi seriale suplimentare pe Arduino. Am folosit-o pentru interactiunea cu modulul Bluetooth, intrucat unul dintre pinii mei de pe placuta (RX sau TX) era stricat si nu aveam cum sa ii folosesc pe aceia.

### Functii implementate

- `setup()` : Functia de initializare a componentelor hardware si software. Configureaza pini, initializează comunicatia serială și modulul LCD, și setează starea inițială a sistemului.

*Initializarea comunicatiei seriale:* `Serial.begin(9600)`

*Initializarea ecranului LCD:* `lcd.init() | lcd.backlight()`

*Initializarea comunicatiei cu modulul Bluetooth:* `HM10.begin(9600)`

- `loop()` : Funcția principală care se execută continuu, monitorizând starea senzorilor și gestionând afișajul și alarmarea.

### Ascultarea portului serial HM10:

```
HM10.listen();
while (HM10.available() > 0)
  appData = HM10.read();
  inData = String(appData);
  Serial.write(appData);
if (Serial.available())
  delay(10);
  HM10.write(Serial.read());
```

### Controlul stării alarmei:

```
if (inData == "0") {
  Serial.println("Alarm OFF");
  digitalWrite(13, LOW);
  if (nr == 0) {
    refresh();
    digitalWrite(ledPin, LOW);
    nr = 1;
  }
}
```

```
}  
while (inData == "1") {  
  nr = 0;  
  Serial.println("Alarm ON");  
  digitalWrite(13, HIGH);  
  val = digitalRead(pirPin);  
  val_cam2 = digitalRead(pirPinCam2);  
  ...  
}
```

### Măsurarea distanței folosind senzorul ultrasonic

```
digitalWrite(TRIGPIN, LOW);  
delayMicroseconds(2);  
digitalWrite(TRIGPIN, HIGH);  
delayMicroseconds(10);  
digitalWrite(TRIGPIN, LOW);  
float distance = pulseIn(ECHOPIN, HIGH) / 58;  
Serial.print("Door distance: ");  
Serial.println(distance);
```

-**refresh()** : Funcție care resetează afișajul LCD și afișează mesajul "You are safe!".

```
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("You are safe!");
```

-**pericol()** : Funcție care afișează mesajul de avertizare de la camera 1 pe LCD.

```
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("Warning from ");  
lcd.setCursor(0, 1);  
lcd.print("camera 1!");
```

-**pericol\_cam2()** : Funcție care afișează mesajul de avertizare de la camera 2 pe LCD.

```
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("Warning from ");  
lcd.setCursor(0, 1);  
lcd.print("camera 2!");
```

-**pericol\_both()** : Funcție care afișează mesajul de avertizare de la ambele camere pe LCD.

```
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("Warning from ");  
lcd.setCursor(0, 1);  
lcd.print("both cameras!");
```

**-alarm(long duration, int freq)** : Funcție care activează buzzerul pentru un anumit timp și frecvență.

```
tone(buzzerPin, freq);  
delay(duration);  
noTone(buzzerPin);
```

## Funcționare

*Starea inițială și actualizările ecranului LCD:*

Afișajul LCD este inițializat și setat să afișeze "You are safe!" în starea inițială. Funcțiile refresh(), pericol(), pericol\_cam2(), și pericol\_both() sunt folosite pentru a actualiza mesajele afișate pe LCD în funcție de starea sistemului și de senzorii PIR.

*Gestionarea alarmei și senzorilor PIR:*

Când inData este "1", alarma este pornită și sunt verificați senzorii PIR pentru a detecta mișcarea. În funcție de valoarea senzorilor, se afișează mesaje de avertizare corespunzătoare și se activează buzzerul. Când inData este "0", alarma este oprită, LED-ul verde este stins, iar afișajul LCD este resetat la mesajul "You are safe!".

*Măsurarea distanței cu senzorul ultrasonic:*

În timpul buclei principale, se măsoară constant distanța folosind senzorul ultrasonic și se afișează pe consola serială.

Aceste funcționalități și structura codului permit monitorizarea și afișarea stării de siguranță sau pericol în funcție de senzorii PIR, activând buzzerul și afișând mesajele corespunzătoare pe LCD.

## Reprezentare logică prin diagrama UML



## Rezultate Obținute

**În urma realizării acestui proiect, am obținut următoarele rezultate:**

- Monitorizarea eficientă a mișcării:

Sistemul a reușit să detecteze cu succes mișcările folosind senzorii PIR plasați în două locații diferite. Atunci când a fost detectată mișcare în oricare dintre cele două camere, sistemul a afișat mesajul corespunzător pe ecranul LCD și a activat buzzerul pentru a emite un semnal sonor de avertizare.

- Afișaj LCD intuitiv și informativ:

Ecranul LCD a fost utilizat eficient pentru a afișa starea de siguranță ("You are safe!") și pentru a emite mesaje de avertizare clare în cazul detectării mișcării, specificând camera de unde provine

alerta. Mesajele au fost ușor de citit și au oferit informații în timp real.

- Interfață de comunicare prin Bluetooth:

Comunicarea bidirecțională prin modulul Bluetooth HM10 a funcționat corespunzător. Sistemul a putut primi comenzi de la o aplicație de pe telefon pentru a activa sau dezactiva alarma. De asemenea, sistemul a trimis date relevante către aplicație, asigurând o monitorizare și control de la distanță eficiente.

- Măsurarea distanței cu senzorul ultrasonic:

Senzorul ultrasonic a măsurat constant distanța până la un obiect aflat în fața sa, afișând această informație pe consola serială. Aceasta poate fi utilizată pentru monitorizarea deschiderii ușii sau a oamenilor care .

- Răspuns prompt la situații de pericol:

Sistemul a fost capabil să reacționeze rapid la detectarea mișcării prin activarea imediată a buzzerului și a LED-ului, oferind o alertă vizuală și auditivă. Această promptitudine este esențială pentru sistemele de securitate și monitorizare.

- Stabilitatea și Fiabilitatea Sistemului:

Pe durata testelor, sistemul a demonstrat stabilitate și fiabilitate în funcționare. Toate componentele hardware și software au funcționat armonios, fără întârzieri semnificative sau erori.

## Optimizări

În urma realizării proiectului, am implementat câteva optimizări pentru a îmbunătăți performanța și eficiența sistemului.

- Gestionarea Eficientă a Senzorilor PIR

Am redus frecvența citirilor de la senzorii PIR prin utilizarea lor doar atunci când alarma este activată, deoarece citirea constantă a senzorilor PIR poate genera alarme false și poate consuma resurse inutile. În funcția `loop()`, citirea senzorilor PIR este realizată doar în timpul activării alarmei, reducând astfel încărcarea procesorului.

- Optimizarea Afișajului LCD

Am optimizat utilizarea funcțiilor LCD pentru a reduce numărul de comenzi trimise la afișaj. În loc să folosim `lcd.clear()` de fiecare dată, am utilizat funcții specifice pentru actualizarea doar a secțiunilor necesare ale afișajului. Afișajele LCD I2C pot fi lente și consumatoare de resurse dacă sunt utilizate ineficient. Am utilizat funcțiile LCD doar în blocurile `refresh()`, `pericol()`, `pericol_cam2()`, și `pericol_both()`, astfel am minimizat utilizarea funcției `lcd.clear()` pentru a evita timpii de răspuns mari.

## Concluzii

Proiectul a atins obiectivele propuse, oferind un sistem de monitorizare și alertare eficient, ușor de

utilizat și de înțeles. Implementarea senzorilor PIR, a senzorului de proximitate, a modulului Bluetooth și a afișajului LCD a permis realizarea unui sistem integrat care poate fi utilizat pentru a asigura securitatea într-o varietate de scenarii. Fiabilitatea și reacția promptă a sistemului demonstrează potențialul acestuia pentru aplicații practice în domeniul securității și monitorizării.

## Download

[sketch\\_may15a.zip](#)

## 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/prj2024/avaduva/matei.costescu2301>



Last update: **2024/05/26 20:57**