

Security Alarm

Autor

Tomescu Petru-Alin

Grupa: 331CC

Introducere

Proiectul descrie crearea unui sistem de securitate ce poate să detecteze intrușii prin intermediul unui senzor PIR și care comunică cu telefonul mobil.

Scopul proiectului este de a obține un sistem de alarmă, de dimensiuni reduse, capabil să atenționeze utilizatorii, atât prin intermediul unor indicatori fizici ai săi precum un ecran LCD și un buzzer, cât și prin transmiterea unor alerte la distanță, pe telefon. Oprirea alarmei se poate face fie fizic prin introducerea unei parole, fie de la distanță, prin intermediul aplicației.

Consider că proiectul devine util pentru protejarea unei zone împotriva accesului nedorit, oferind protecție continuă prin capacitatea sa de alertare la distanță.

Descriere generală

La pornirea sistemului, acesta va citi parola de securitate, salvată pe un card SD. Totodată, sistemul oferă utilizatorului posibilitatea de a o reseta. Pentru a avea acces la conținutul cardului SD și a putea realiza operații (citire/scriere), va fi utilizat un modul de card SD. Urmează apoi inițializarea componentelor, durând aproximativ un minut pentru acomodarea senzorului PIR cu camera în care se află.

Atunci când senzorul PIR detectează mișcare, buzzer-ul sistemului de securitate va începe să scoată un zgomot de atenționare. Sistemul va folosi modulul bluetooth pentru a transmite că alarma a fost activată și pe telefonul mobil. Oprirea alarmei se realizează direct din aplicație, sau prin apăsarea unui buton pentru a activa o întrerupere a sistemului, ce permite apoi utilizatorului să introducă parola de securitate (anterior setată sau resetată).

În timpul funcționării, un LCD va fi utilizat cu scopul de a afișa diverse mesaje despre starea în care se găsește sistemul de securitate. Sistemul dispune și de un keypad, utilizat pentru introducerea parolei sau acceptarea, neacceptarea diferitelor operații. Aceste două componente vor fi conectate nu direct la placa Arduino, ci prin intermediul unor module I2C. Această interfațare are drept scop reducerea numărului de pini necesari pentru conectarea celor două componente.

Schema Bloc



Hardware Design

Componente Utilizate	Cantitate
Arduino UNO R3 ATmega328p	X1
Senzor PIR HC-SR501	X1
Buzzer	X1
LCD 1602	X1
I2C adaptor pentru LCD	X1
Button	X1
Tastatura Matriciala 4x4 (Keypad)	X1
Modul expansiune IO I2C	X1
Modul SD card	X1
Card Micro SDHC de 8 GB	X1
Modul bluetooth HC-05	X1
Breadboard	X5
Fire legatura	-

Schemă Electrică



Software Design

In cadrul proiectului, am folosit ca mediu de dezvoltare Arduino IDE.

Arduino Uno

Biblioteci utilizate

In cadrul proiectului am utilizat urmatoarele biblioteci:

- I2CKeyPad.h, precum si Keypad.h pentru a utiliza keypad-ul 4x4 impreuna cu modulul I2C expander;

- SD.h si SPI.h pentru a face posibil accesul la cardul micro-sd si a putea citi parola din fisierul de configurare stocat pe acesta;
- SoftwareSerial.h pentru a putea comunica cu aplicatia de telefon prin modulul de bluetooth;
- LiquidCrystal_I2C.h si Wire.h pentru a folosi LCD 1602 impreuna cu modulul I2C;

Functionare sistem

Setup

La pornirea sistemului, au loc urmatoarele actiuni:

- Setarea pinilor pentru PIR (INPUT), Buzzer(OUTPUT), precum si a celui pentru Buton (INPUT PULLUP);
- Activarea intreruperilor externe pentru buton;
- Initializarea LCD-ului;
- Pornirea modulului bluetooth, moment din care aplicatia poate fi conectata la acesta;
- Citirea parolei, setata in cadrul functionarii anterioare, de pe cardul micro-sd, precum si salvarea ei in memorie (prin functia getPassword());
- Initializarea keypad-ului;
- Apelarea functiei de reset, pentru a permite utilizatorului sa reseteze parola, inainte de inceperea sistemului;
- Calibrarea modulului PIR, prin asteptarea unui minut pentru a se obsinui cu temperatura camerei;

Bucula principala

Pe durata functionarii sistemului de alarma, exista 4 stari(moduri) in care acesta se afla:

- Modul 0 (Activ): Aici are loc citirea senzorului PIR. In cazul in care acesta a detectat miscare, se afiseaza pe LCD mesajul corespunzator, se trimite o alerta pe aplicatie. Apoi se face trecerea sistemului in modul 1. Tot aici, in cazul in care butonul este apasat, se trece in modul 3 unde se realizeaza resetarea parolei sistemului si salvarea ei atat in memoria programului, cat si pe cardul micro-sd.
- Modul 1 (Miscare detecata): In acest mod, miscare a fost detectata, deci este pornit si sunetul buzzerului. In cazul in care se primeste un mesaj de oprire de la aplicatie, alarma se opreste si are loc recalibrarea senzorului PIR (asteptarea 1 minut), precum si trecerea in modul 0. In cazul in care se apasa be buton, se realizeza trecerea in modul 2, pentru introducerea parolei.
- Modul 2 (Introducere parola): In acest mod, poate fi introdusa parola. In cazul in care aceasta este incorecta, se trece inapoi in modul 1. Daca parola e corecta, se realizeaza recalibrarea senzorului PIR, precum si trecerea inapoi in modul 0.
- Modul 3 (Resetare parola): In acest mod se poate ajunge doar prin apasare butonului, din modul 0. Are loc resetarea parolei, precum si intoarcerea in modul 0.

Senzor Bluetooth HC-05 si aplicatia de telefon

Comunicarea cu senzorul HC-05 se realizeaza printr-o aplicatie de android. Pentru implementarea functionalitatilor aplicatiei, am utilizat MIT app inventor. Aplicatia permite conectarea la senzorul HC-05, precum si deconectarea de la acesta in orice moment in timpul functionarii aplicatiei.

La primirea mesajului "Motion Detected!", aplicatia afiseaza pe ecran mesajul. Totodata, se activeaza butonul de stop alarm. Apasarea sa determina trimiterea unui mesaj de stop inapoi catre Arduino, ce opreste alarma.

La primirea mesajului "WAIT", aplicatia nu mai permite apasarea butonului de stop alarm, in cazul in care acesta era activ. Scopul este impiedicarea trimiterii unui mesaj de stop eronat catre Arduino, in cazul in care alarma oricum este dezactivata de la sistemul principal, in mod fizic. Desi primirea mesajului nu afecteaza alarma, atunci cand aceasta este in alt mod de functionare, mesajul ramane salvat in buffer-ul lui HC-05. Astfel, la o viitoare declansare a alarmei, mesajul de stop trimis eronat va primit, si alarma oprita in mod gresit.

Funcții implementate

- `calibratePir()`:

Funcția realizează așteptarea unui minut, ceea ce permite resetarea senzorului PIR pentru a putea detecta din nou.

- `setPassword()`:

Funcția așteaptă introducerea unei parole de 4 digiti de la keypad, și o salvează în memoria programului.

- `getPassword()`:

Funcția citește din memoria cardului micro sd parola, și o încarcă în memoria programului.

- `setButton()`:

Funcția realizează setarea pinului butonului, în mod INPUT PULLUP.

- `setPins()`:

Funcția realizează setarea pinilor pentru PIR și Buzzer, precum și pentru buton prin apelarea lui `setButton()`.

- `setInterrupts()`:

Funcția setează întreruperile externe, declanșate la apăsarea butonului.

- `initLCD()`:

Funcția inițializează LCD-ul.

- `writeToCard()`:

Funcția scrie pe cardul micro-sd parola curentă salvată în memorie.

- `reset()`:

Funcția permite resetarea parolei curente, utilizatorul putând accepta (apasa tasta A) sau nu (apasa tasta D). Dacă decide resetarea, parola este setată prin funcția `setPassword()` și apoi scrisă pe cardul micro-sd cu un apel al funcției `writeToCard()`.

- `introducePassword()`:

Funcția așteaptă introducerea unei parole de 4 cifre de la tastatură. Dacă aceasta este corectă, are loc recalibrarea senzorului PIR (un apel `calibratePir()`), precum și trecerea în modul 0 de funcționare. Dacă parola este greșită, sistemul va rămâne în modul 1, iar alarma va rămâne activă.

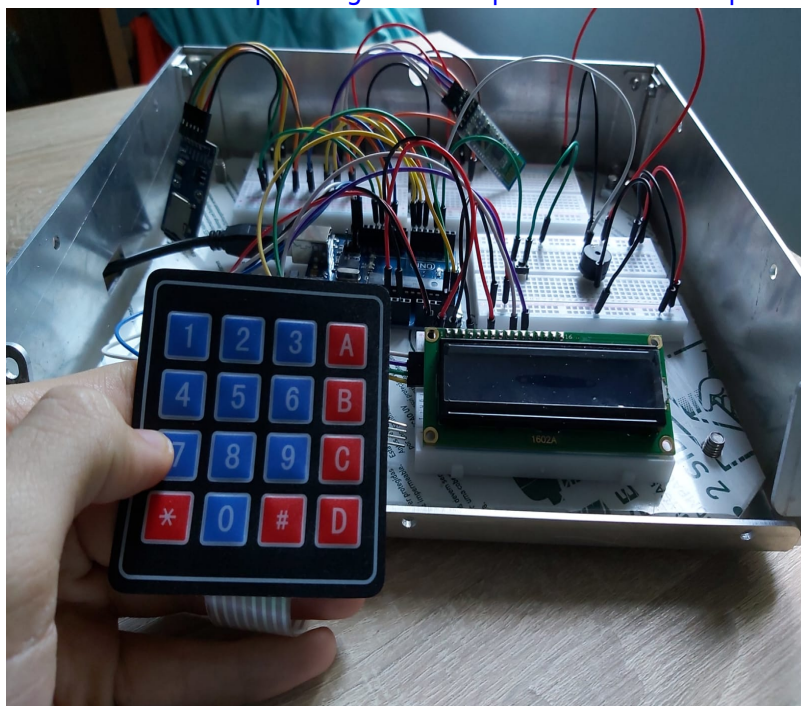
Rezultate Obținute

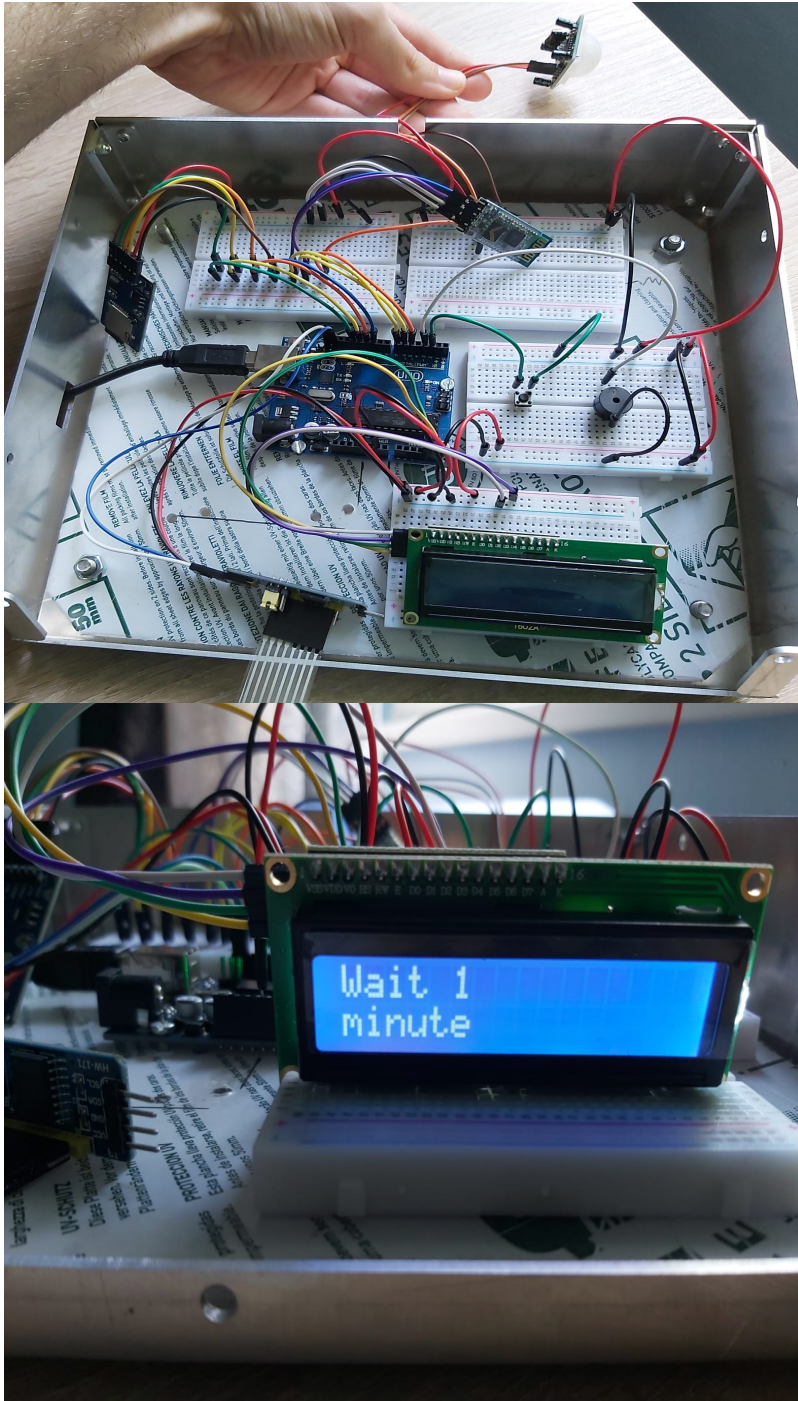
[Demo activare alarma si oprire prin aplicatie](#)

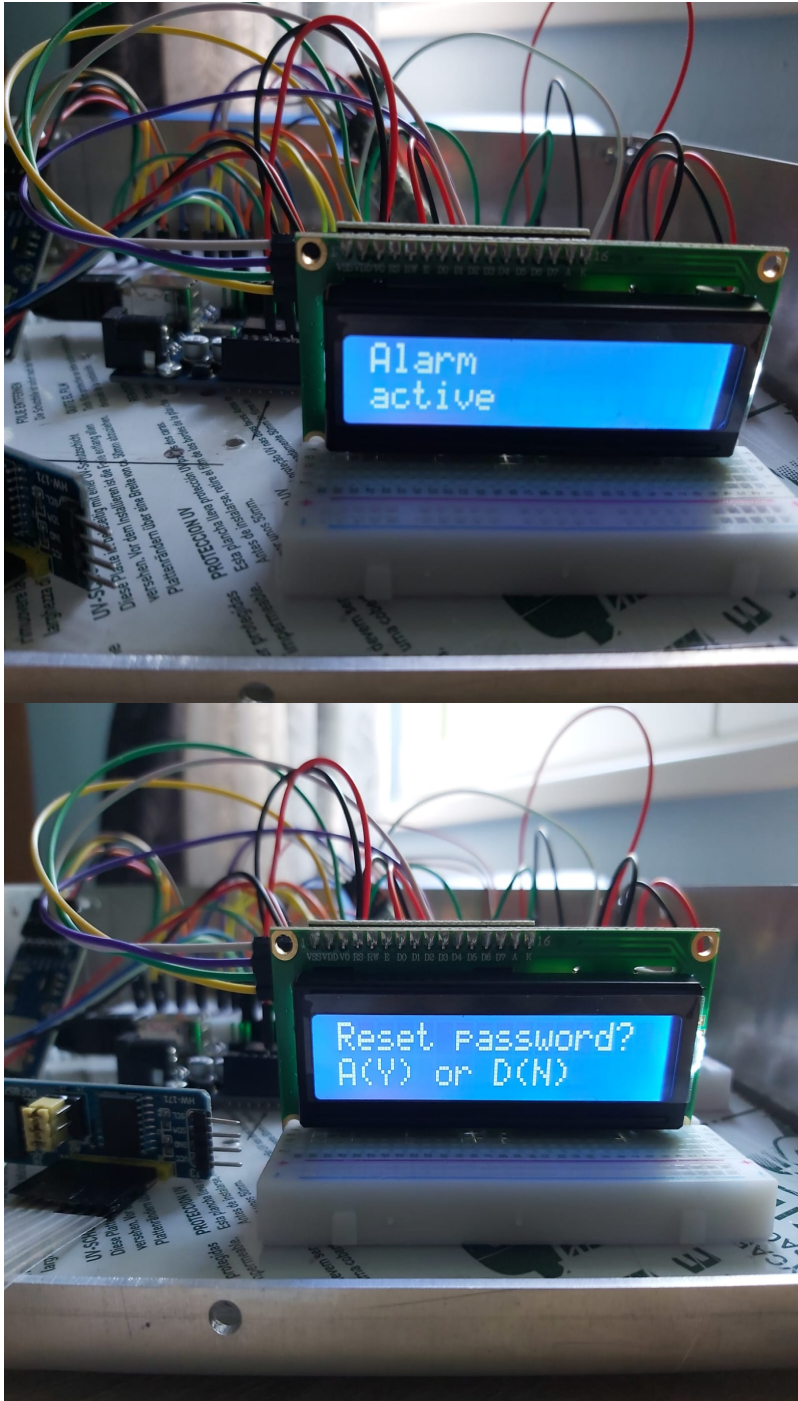
[Demo introducere parola incorecta si corecta](#)

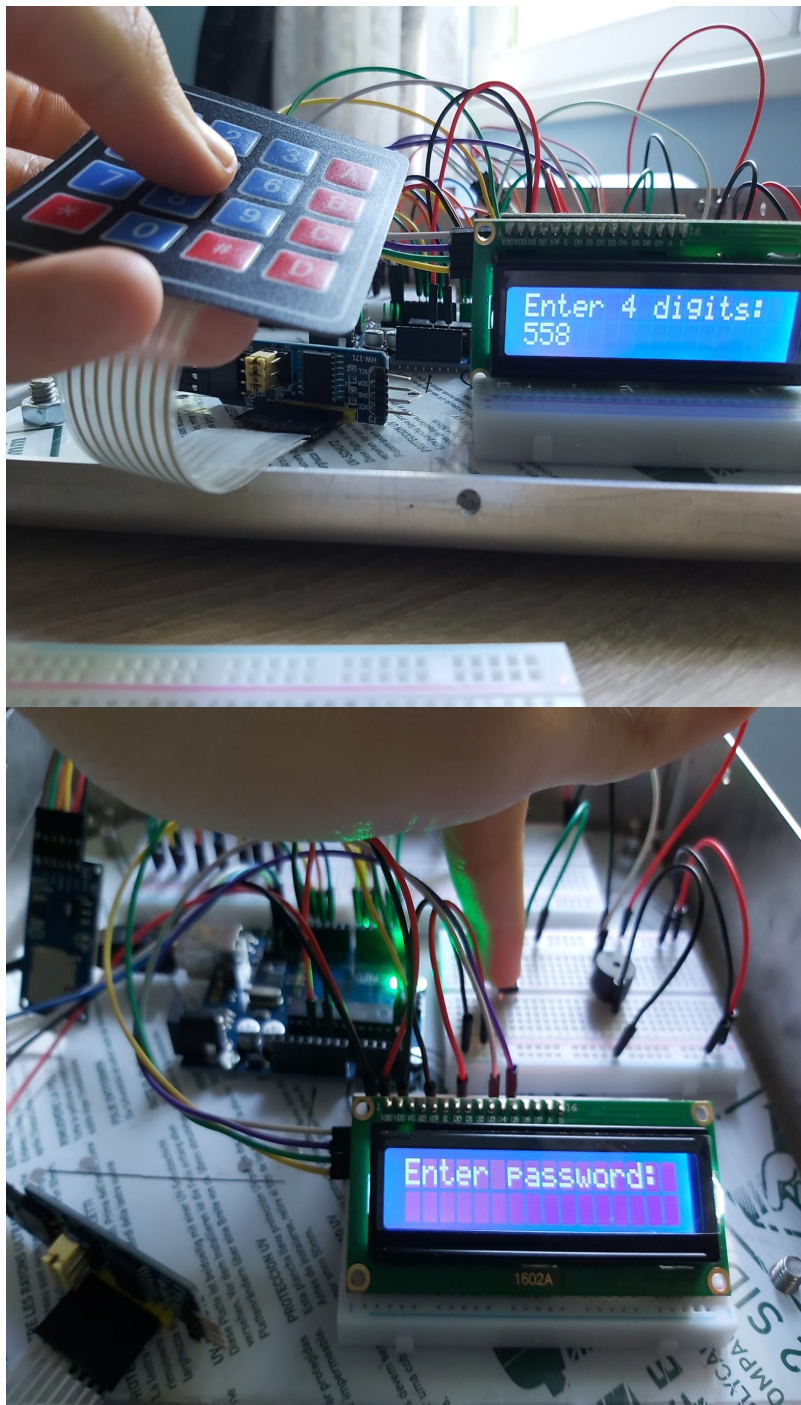
[Demo password reset](#)

[Demo introducere parola gresita si oprire alarma din aplicatie](#)





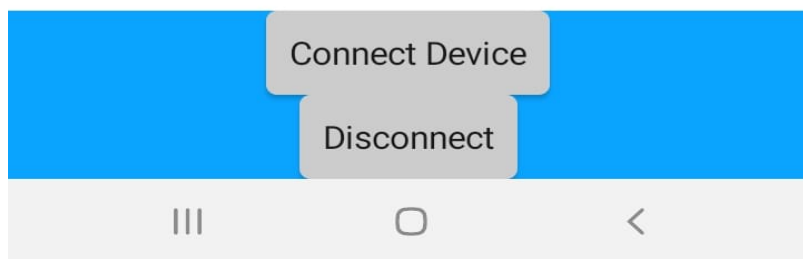
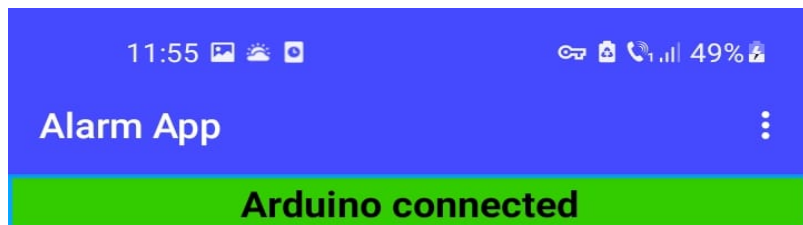




Aplicatie

Aplicatie neconectata \ conectata





Aplicatie conectata, cu notificare primita. Butonul de oprirea a sistemului nu e activat. Se asteapta oprirea prin introducerea parolei.



Aplicatie conectata, cu notificare primita si butonul de oprire a sistemului activ.



Concluzii

De-a lungul proiectului, am intampinat mici obstacole, atat software, cat si hardware. Acestea m-au determinat sa lucrez cu mai multe metode de debugging, de la printarea pe seriala, la testarea separate a componentelor, atat a functionalitatii, cat si a alimentarii lor corecte, utilizand leduri. Am

invatat, de asemenea, sa lucrez mai bine cu Arduino, proiectul oferindu-mi posibilitatea de a aprofunda conceptele studiate la laborator si curs.

In concluzie, experienta a fost una placuta si o modalitate excelenta de a pune in practica ceea ce am invatat de-a lungul semestrului despre lucrul cu Arduino, cu datasheet-uri pentru diverse componente electronice si de intra in lumea micro-controller-elor.

Download

Link catre arhiva zip, unde se pot gasi atat codul sursa, cat si codul aplicatie de android si schema electrica:

[security_alarm.zip](#)

Github: [SecurityAlarm](#)

Jurnal

- 21.4.2022: Testat senzorul PIR, modulul pentru micro-sd card si keypadul;
- 23.4.2022: Imbinarat senzorii in cadrul proiectului;
- 2.5.2022: Testarea Modulului Bluetooth si crearea unei aplicatii pentru primirea

alertelor;

- 6.5.2022: Lipit LCD si modulul I2C, precum si testarea lui;
- 12.5.2022: Adaugare LCD si modul Bluetooth in cadrul proiectului;
- 13.5.2022: Terminat sectiuniile pentru M1;
- 18.5.2022: Creare schema electrica;
- 20.5.2022: Adaugare posibilitatea de reset a parolei in timpul functionarii sistemului;
- 22.5.2022: Extinderea aplicatiei de android pentru a putea si trimite un semnal de oprire a alarmei;
- 23.5.2022: Update la schema electrica de pe ocw;
- 24.5.2022: Fixat componentele si Arduino pe o placa de sustinere;
- 25.5.2022: Adaugarea sectiunii despre Software pe ocw;
- 26.5.2022: Adaugat poze, demouri + Bibliografie si resurse;

Bibliografie/Resurse

[Security Alarm](#)

Linkuri utile:

Laboratoare

- <https://ocw.cs.pub.ro/courses/pm/lab/lab0-2022>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab1-2022>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2022>

- <https://ocw.cs.pub.ro/courses/pm/lab/lab5-2022>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2022>

Software

- <https://www.arduino.cc/reference/en/libraries/sd/>
- <https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>
- <https://www.arduino.cc/reference/en/libraries/i2c keypad/>
- <https://www.arduino.cc/reference/en/language/functions/communication/spi/>
- <https://create.arduino.cc/projecthub/electropeak/getting-started-with-hc-05-bluetooth-module-arduino-e0ca81>
- https://create.arduino.cc/projecthub/Arnov_Sharma_makes/lcd-i2c-tutorial-664e5a
- <https://appinventor.mit.edu/> (Creare aplicatie android)

Hardware

- https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- https://components101.com/sites/default/files/component_datasheet/Hc-05%20Datasheet.pdf
- <https://www.homemade-circuits.com/pir-sensor-datasheet-pinout-specification-working/>
- <https://components101.com/modules/micro-sd-card-module-pinout-features-datasheet-alternatives>
- <https://electronics.stackexchange.com/questions/220641/pir-sensor-always-gives-high>
- https://www.youtube.com/watch?v=-B6_RVCNMoe (Lipire LCD cu modulul I2C)
- https://www.mouser.com/datasheet/2/400/ef532_ps-13444.pdf

Biblioteci utilizate:

- <https://github.com/RobTillaart/I2CKeyPad/blob/master/I2CKeyPad.h>
- https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library/blob/master/LiquidCrystal_I2C.h
- <https://github.com/PaulStoffregen/SoftwareSerial>
- <https://github.com/esp8266/Arduino/blob/master/libraries/Wire/Wire.h>

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2022/sgherman/security_alarm



Last update: **2022/05/31 08:28**