

Sistem de protecție prin parolă

Introducere

Proiectul presupune implementarea unui sistem de protecție prin parolă ce poate fi aplicată pe diverse dispozitive pentru a ale proteja. Utilizatorul interacționează cu sistemul de protecție prin intermediul unei tastaturi pentru a:

- seta o nouă parolă (trebuie să cunoască parola veche),
- introduce parola curentă pentru a accesa dispozitivul protejat.

Descriere generală

În cazul în care utilizatorul introduce parola greșit în număr de 3 ori pe parcursul a 5 minute, alarma declanșează un sunet prelung. Dacă utilizatorul introduce parola greșit dar nu atinge limita de 3 încercări în 5 minute, pentru fiecare încercare eșuată, alarma declanșează un sunet scurt. În cazul în care parola a fost introdusă corect, utilizatorul va fi întâmpinat cu o melodie interpretată de buzzer.

Alarma va aștepta input doar dacă utilizatorul se află la un 1 metru față de dispozitivul protejat.



Hardware Design

Lista pieselor folosite:

- [Arduino Uno](#)
- [Buzzer pasiv 5V](#)
- [LCD 16x2 I2C](#)
- [Keypad 4x4](#)
- [Fir pentru Baterie de 9V cu Mufa DC](#) (nice to have)
- Senzor de mișcare
- Breadboard
- Fire

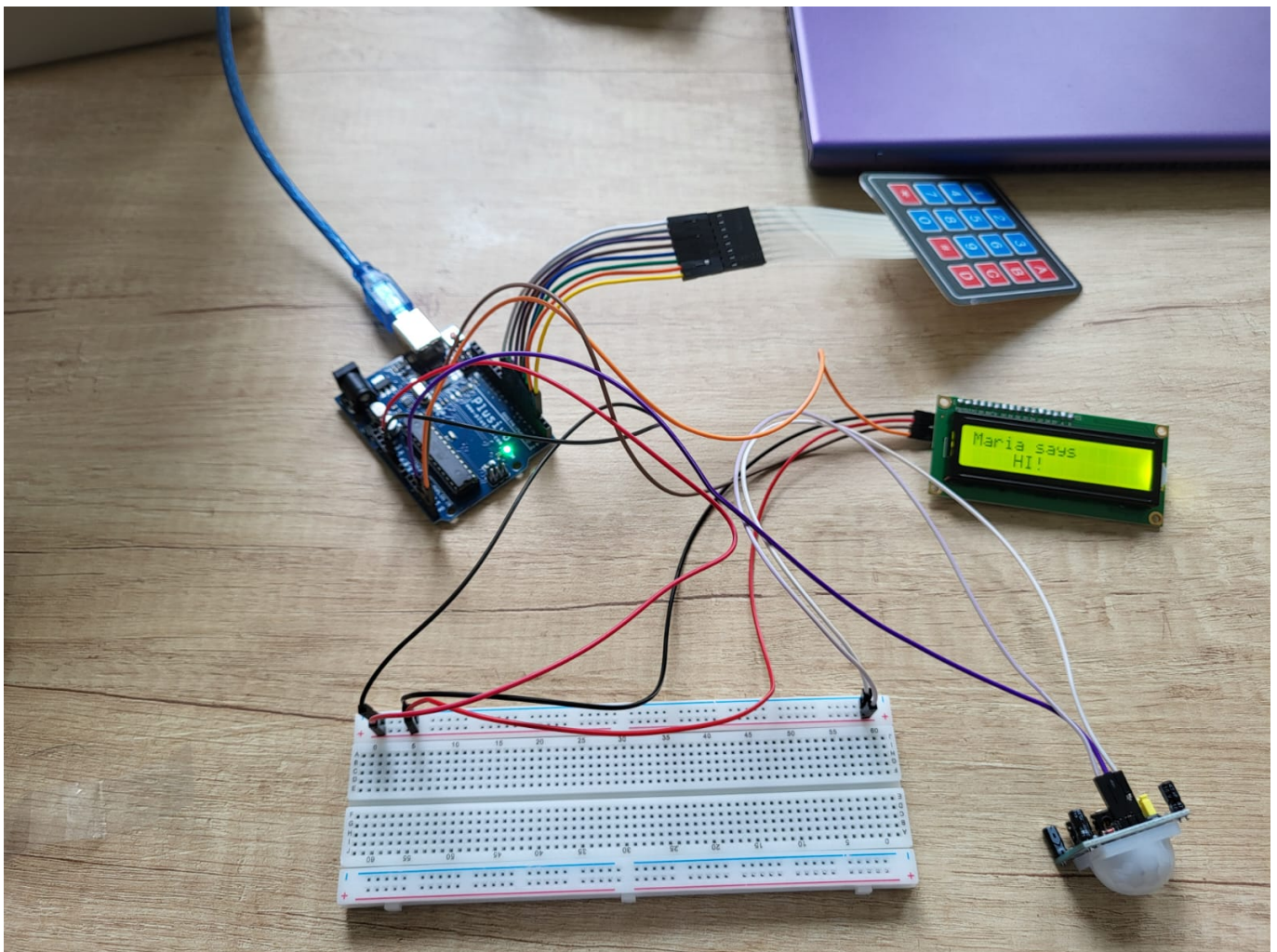
Schema folosită pentru asamblarea diferitelor module hardware este:



După cum este evidențiat în schema electrică, rolul componentelor și elementele lor de conectare sunt:

- **Arduino Uno**: folosit ca putere de calcul și memorie în cazul buzzerului; conectat prin cablu USB AM la sursa de alimentare.
- **Buzzer pasiv 5V**: folosit pentru a reda o melodie (salvată în memoria microcontrolerului) în cazul introducerii corecte a parolei sau atenționarea utilizatorului altfel; conectat la pinul 10 PWM (ales aleator dintre pinii de PWM) și la GND.
- **LCD 16x2 I2C**: folosit pentru a reda pe prima linie mesajul "Type passwd", iar pe a doua linie parola introdusă în clar; conectat la GND și 5V pentru a fi alimentat și la pinii analogici A4 și A5 prin SDA și SCL [după cum specifică datasheet-ul](#).
- **Keypad 4x4**: folosit pentru a introduce caracterele parolei; conectat prin 4 linii la pinii digitali D9, D8, D7, D6 și prin 4 coloane la pinii digitali D5, D4, D3, D2, [după cum specifică datasheet-ul](#), astfel un buton al tastaturii este identificat printr-un pin de coloană și unul de linie.
- **Senzor de mișcare PIR**: folosit pentru a determina dacă utilizatorul se regăsește la cel puțin 1m de alarmă, este setat în modul single trigger pentru a anunța o singură dată intrarea utilizatorului în aria de acțiune; conectat la GND și 5V pentru a fi alimentat și la pinul analogic A3 (ales aleator) pentru a trimite semnalul de pornire a alarmei în cazul apropierii utilizatorului.

Proof of work:



Software Design

Mediu de dezvoltare: VSCode PlatformIO

Biblioteci folosite:

- marcoschwartz/LiquidCrystal_I2C@1.1.4 - folosită pentru display-ul LCD care afișează parola
- chris--a/Keypad@3.1.1 - folosită pentru keypad-ul care introduce parola
- arduinogetstarted/ezBuzzer@1.0.0 - folosită pentru activarea buzzer-ului asincron (adică fără funcția `delay()`)

Funcții și structuri folosite:


- `activate_alarm()` - activează alarma în urma input-ului primit de la senzorul de mișcare
- `introduce_passwd()` - primește input-ul de la keypad pe care îl afișează pe LCD; setează un flag legat de ce acțiune s-a ales: introducere parolă curentă sau update parolă
- `check_passwd()` - verifică parola în cazul în care s-a ales introducerea ei; în cazul în care se face update la parolă, se verifică parola veche și se face update la cea nouă
- `play_melody()` - emite melodia care anunță introducerea corectă a parolei
- `play_alarm()` - emite alarmă care anunță introducerea greșită a parolei

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/prj2024/rpascale/maria.sfiraiala>



Last update: **2024/05/26 09:24**