

# PIN Door Lock

Autor: Olaru Georgiana-Letitia 333CB

## Introducere

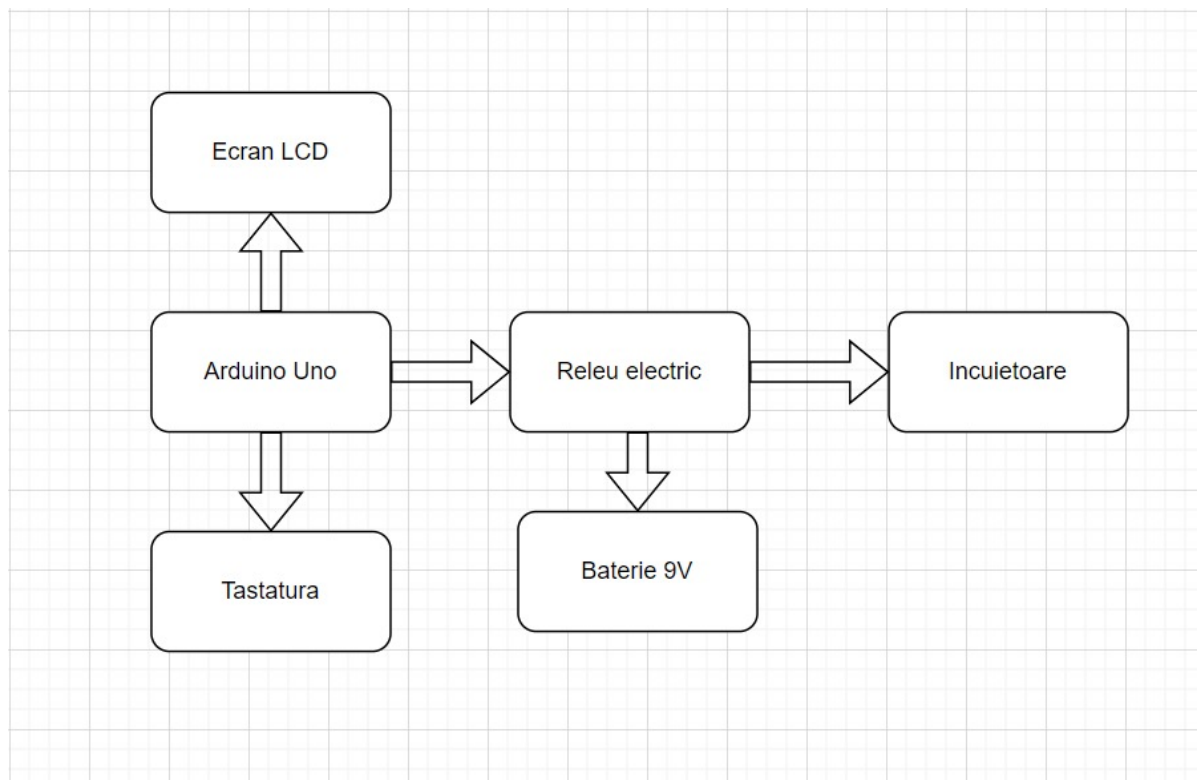
Proiectul consta in realizarea unui sistem care are ca scop deblocarea unei usi atunci cand este introdus codul pin corect, urmat de tasta '\*'. Daca codul introdus este corect se va auzi o melodie de succes sau de fail in cazul in care codul este gresit. De asemenea, codul pin se poate schimba cu unul nou daca este apasata tasta '#'.

## Descriere generală

Deschiderea usii se realizeaza prin introducerea codului pin de la tastatura, urmat de tasta '\*'. Daca codul introdus este corect, se va afisa un mesaj specific, iar daca este gresit, se va afisa alt mesaj si va putea fi introdus un alt cod, apasandu-se inainte tasta '#'. De asemenea, se poate schimba parola folosind tasta '#' cu una noua.

Vom folosi un timer pentru a nu trece foarte mult timp intre introducerea codului pin si apasarea tastei '\*', pentru ca altfel se va reseta si va trebui introdus din nou pinul.

## Schema Bloc

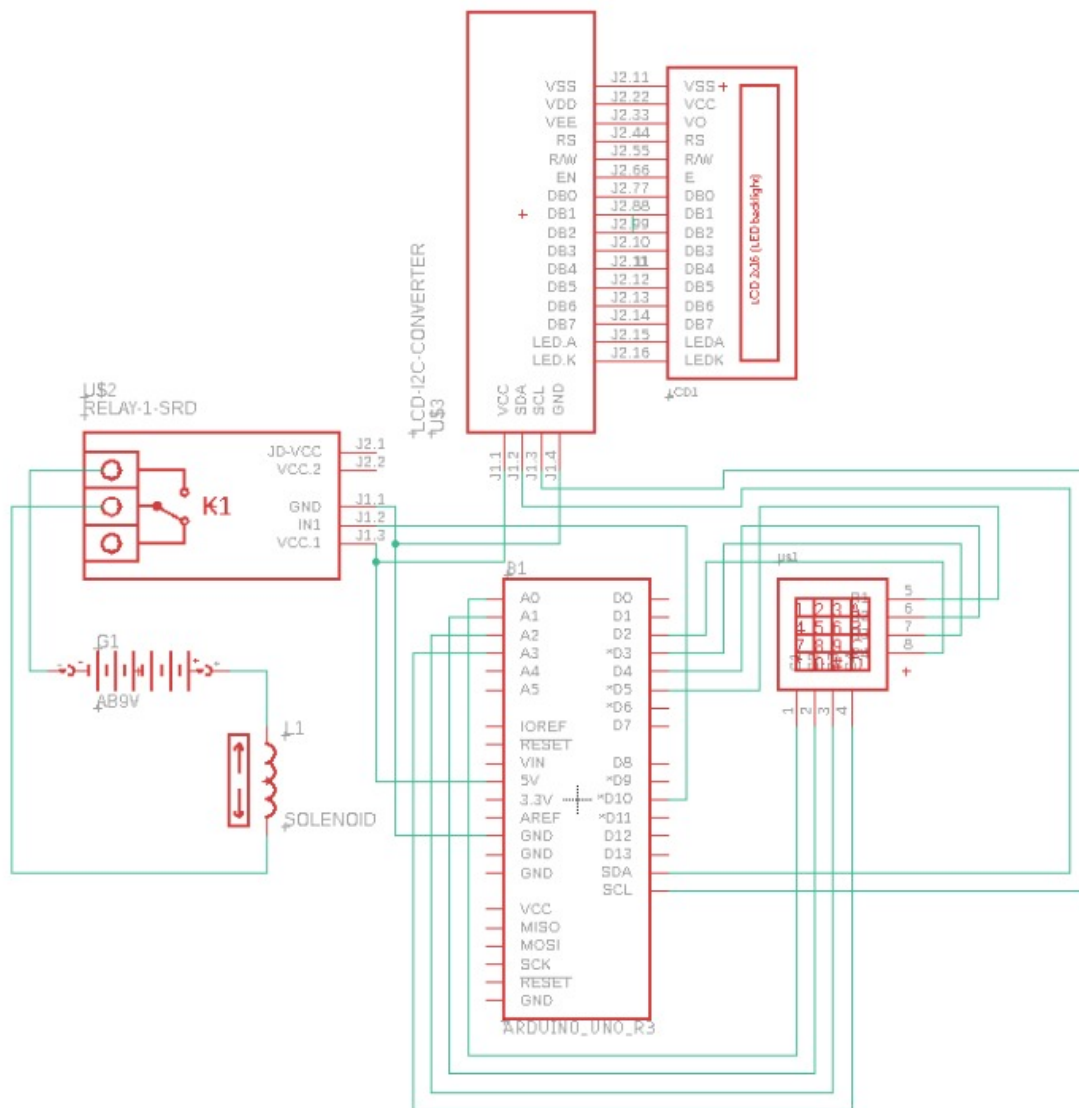


## Hardware Design

### Lista piese:

- Arduino Uno R3
- Breadboard
- Tastatura 4x4
- Ecran LCD
- Baterie 9v
- Releu electric
- Incuietoare
- Conector baterie 9v
- Buzzer
- Fire

### Schema electrica



## Software Design

Pentru a putea implementa partea software am folosit Arduino IDE si librariile Keypad.h de la Mark Stanley si Alexander Brevig, LiquidCrystal\_I2C.h de la Frank de Brabander si biblioteca oficiala EEPROM.h. (Toate link-urile sunt incluse in bibliografie).

La inceputul programului imi declar si imi initializez toate variabilele de care o sa am nevoie. Pentru a salva o parola am folosit memoria EEPROM. Initial am setat o parola demo (0000), dar apoi am comentat acele linii de initializare pentru a nu fi mereu acea parola. Apoi de fiecare data citesc ultima parola salvata pe Arduino si initializez parola in functia numita initialPassword, care este apelata mai tarziu in setup().

In functia setup() setez pe HIGH relay pin-ul (adica poarta este inchisa), este pornit ecranul, se citeste parola din memorie si setez registrele pentru timer.

Functia ISR este o functie de intrerupere care se declanseaza la fiecare 4 secunde si doar seteaza o variabila pe true pentru resetarea ecranului.

Functia `resetScreen()` reseteaza ecranul daca trec 4 secunde si reseteaza timer-ul.

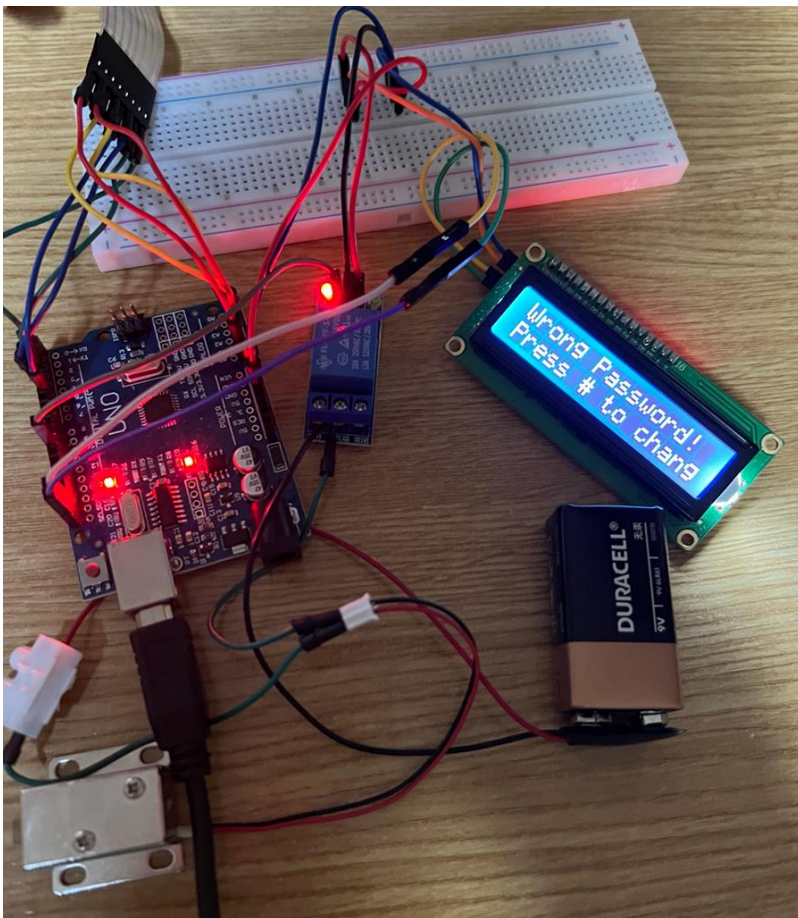
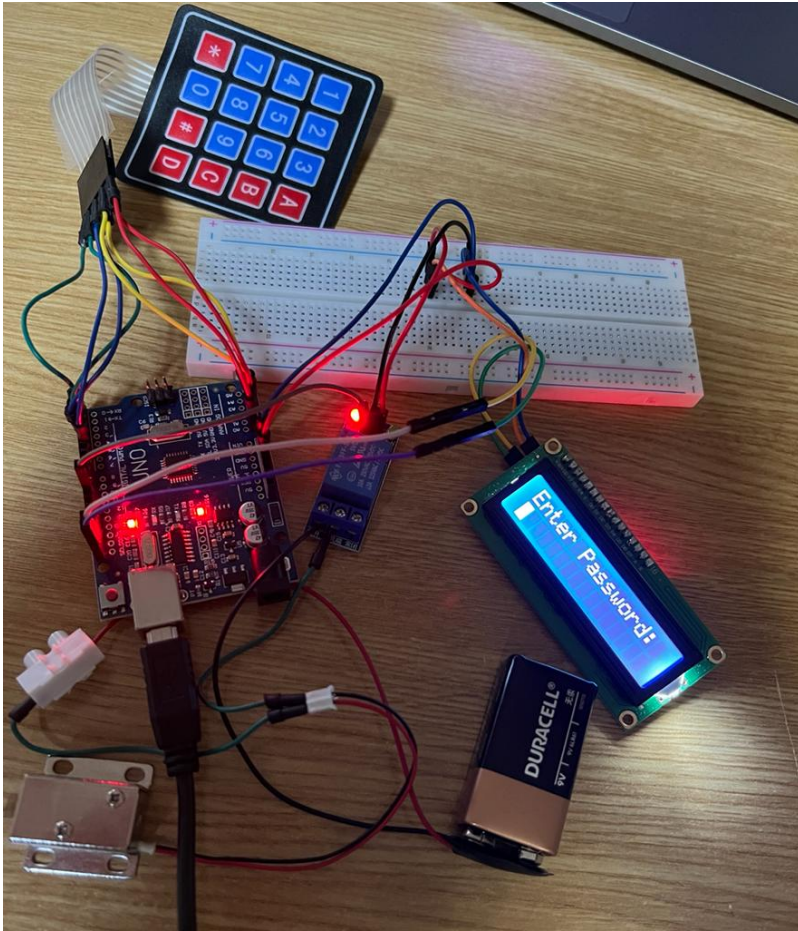
In functia `loop()` se verifica initial timer-ul, adica daca au trecut mai mult de 4 secunde de cand nu a fost apasat vreun buton si se reseteaza si codul pin trebuie introdus din nou. De asemenea, se apasa cate o cheie de pe tastatura si se verifica ce tasta reprezinta. Daca este '#', atunci se apeleaza functia `change()` care schimba parola cu una noua, daca este 'C', se sterge tot de pe ecran si se revine cu indexul la 0. Daca este altceva introdus, inseamna ca a fost apasata o tasta care reprezinta o bucata din parola, iar caracterul se adauga in string. Apoi, daca se apasa tasta '\*', incepe sa se verifice cu parola initiala. Se citeste iar parola, se verifica si daca este corecta, apare mesajul 'Pass Accepted!', se trece relay pin-ul in LOW si se deblocheaza poarta si ramane asa timp de 4 secunde. Tot aici iti afiseaza pe ecran daca vrei sa schimbi parola. Daca parola introdusa este gresita, relay pin-ul ramane in continuare pe HIGH, afiseaza 'Wrong Password!' si te intreaba din nou daca vrei sa schimbi parola.

Functia `change()` schimba parola cu una noua. Cand este apasata tasta '#', trebuie introdusa parola curenta si daca este cea corecta, poti sa introduci noua parola si se salveaza in EEPROM caracter cu caracter si afiseaza ca s-a schimbat parola cu succes.

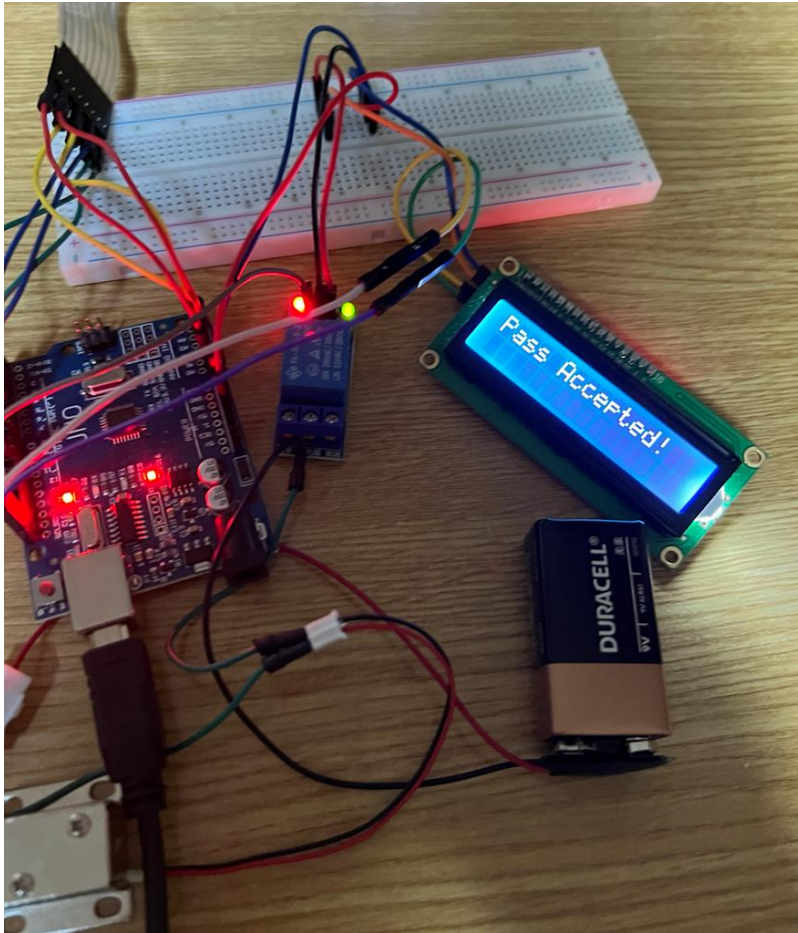
Folosindu-ma de functiile `playSuccessSong()` si `playWrongSong()` am implementat functionalitatea de a se auzi o melodie de succes pentru atunci cand parola este cea corecta si una de fail pentru atunci cand parola este gresita. Am 2 vectori ce contin notele muzicale pentru parola corecta, respectiv gresita. In cele doua functii, mai intai se calculeaza lungimea (numarul de note), apoi se trece prin fiecare nota, se genhereaza un sunet si apoi cate o pauza, care este mai mare decat durata notei.

## Rezultate Obținute

Demo: [https://www.youtube.com/watch?v=nUW5KafqmEY&ab\\_channel=LetitiaOlaru](https://www.youtube.com/watch?v=nUW5KafqmEY&ab_channel=LetitiaOlaru)







## Concluzii

Realizarea acestui proiect a fost o experienta interesanta pentru ca am putut expune cunostintele dobandite pe parcursul semestrului intr-un mod fun, putand sa ne alegem ce proiect vrem noi.

## Download

## Bibliografie/Resurse

[pm-pin\\_door\\_lock.pdf](#)

<https://ocw.cs.pub.ro/courses/pm>

<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>

<https://playground.arduino.cc/Code/Keypad/#Download>

<https://gist.githubusercontent.com/mikeputnam/2820675/raw/bbe995aa22826a8fbbb6b56ccd56513f9db6cb00/pitches.h>

<https://create.arduino.cc/projecthub/GeneralSpud/passive-buzzer-song-take-on-me-by-a-ha-0f04a8>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/imacovei/georgiana.olaru>

Last update: **2022/05/23 20:26**

