

Mecanism de verificare a parolelor

Autor

Autor: [Marin Ioana-Valentina](#)

Introducere

- Scopul proiectului este de a implementa un mecanism simplist de verificare a parolelor, un seif. Parola va fi stocata in mod dinamic folosindu-se un modul RTC DS3231 si in functie de ora data de acesta folosindu-se o functie de hash se va obtine o parola.
-
- In momentul in care utilizatorul va introduce un sir de caractere folosind un keypad, se va verifica daca textul introdus corespunde cu parola memorata de catre seif. In caz afirmativ, LED-ul verde al seifului se va aprinde, va fi afisat un mesaj corespunzator pe ecranul LCD, buzzer-ul va genera o melodie si se va putea deschide usita seifului actionand manerul acesteia. In caz contrar se va aprinde LED-ul rosu al seifului, va fi afisat un mesaj de eroare pe ecranul LCD, buzzer-ul va genera un ton simplist de eroare si incuietoarea seifului va ramane inchisa.

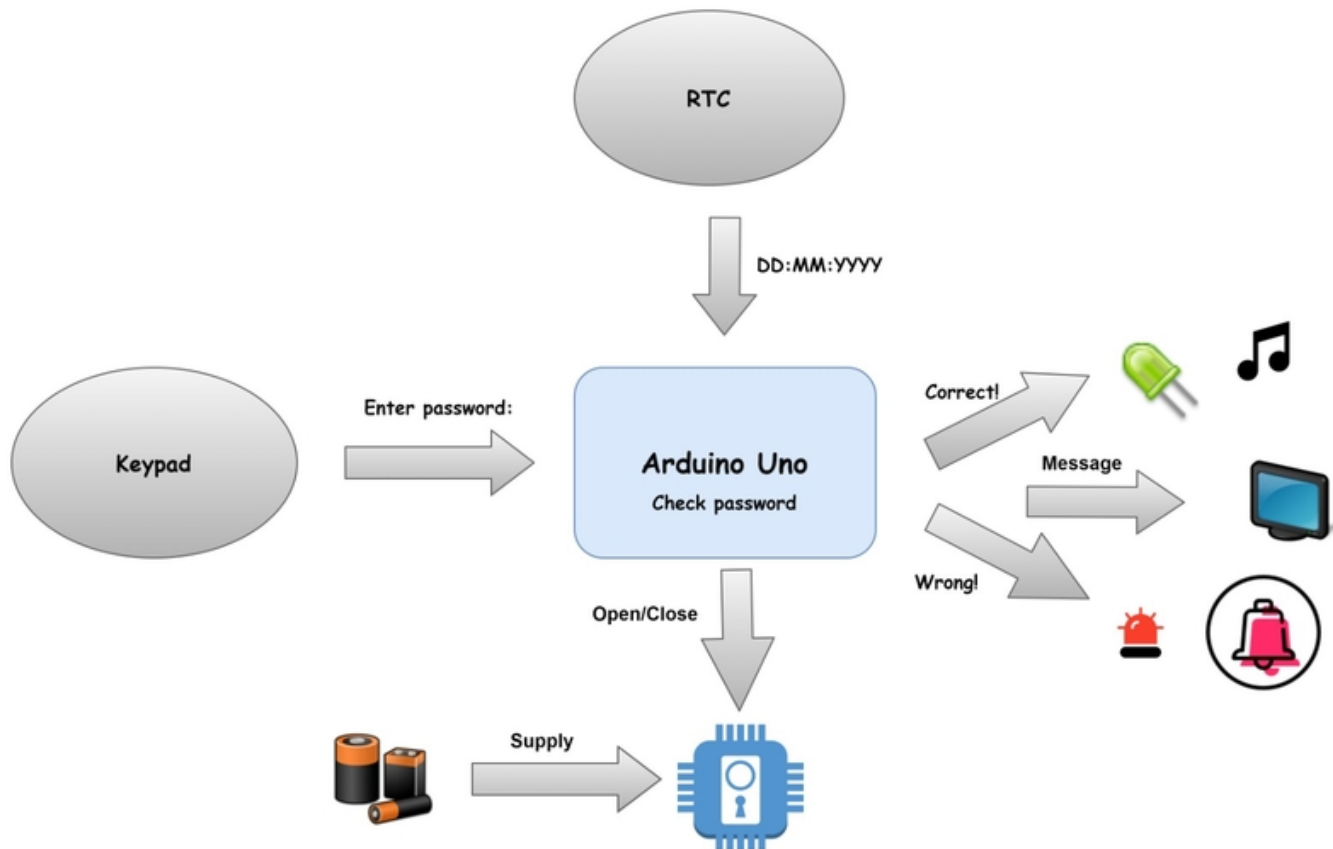


Descriere generală

Schema bloc

Proiectul este format din doua parti conectate:

1. Arduino Uno la care vom conecta breadboard-ul pe care vom atasa LED-urile (rosu si verde), buzzer-ul, LCD-ul, keypad-ul si modulul RTC.
2. Incuietoarea alimentata de modulul releu impreuna cu cele 4 baterii de 3.7V Li-Ion in suportul corespunzator.



1. Arduino:
 - Nucleu al seifului.
 - Componenta ce mediaza fluxul de date dintre Keypad, modul RTC, LCD 16×2, solenoid.
 - Are rolul de a compara parola primita prin intermediul Keypad-ului (de la utilizator) si parola generata pe baza datelor emise de modului RTC.
 - Actioneaza LED-urile si buzzer-ul in functie de corectitudinea sirului de caractere primit de la utilizator.
2. Keypad:
 - Are rolul de a trimite datele introduse de utilizator (sirul de caractere) catre Arduino pentru a putea fi verificate.
3. Modul RTC DS3231:
 - Are rolul de a "dinamiza" aplicatia.
 - Generarea parolelor stocate pe Arduino in mod dinamic in functie de momentul zilei.
4. Modul LCD 16×2 I2C:
 - Are rolul de a afisa mesaje catre utilizator: atat instructiuni de utilizare a seifului cat si de a instiinta utilizatorul daca parola introdusa de acesta este corecta sau nu.
5. Solenoid:

- Incuietoarea seifului.
 - Aceasta se deschide in momentul in care este introdusa parola corecta si se inchide in momentul in care este apasata tasta 'C' a keypad-ului.
 - Daca parola introdusa de utilizator este gresita, aceasta nu este actionata in niciun fel.
6. Releu:
- Are rolul de a genera un camp electromagnetic care va actiona incuietoarea seifului.
 - Este conectat la un pin analogic cu rol de output de la Arduino.
7. Baterii:
- 4 baterii de 3.7V Li-Ion
 - Folosite pentru a alimenta cu 12V solenoidul.
8. LED-uri:
- In cazul in care parola este corecta se va aprinde LED-ul verde, iar in caz contrar se va aprinde cel rosu.
9. Buzzer:
- In cazul in care parola este corecta va genera o melodie simpla, iar in caz contrar va genera un ton de eroare.

Hardware Design

Lista de piese

Nume	Numar piese
Arduino Uno R3	1
Keypad	1
Modul RTC	1
Buzzer	1
Breadbord	1
Ecran LCD	1
Solenoid	1
Modul Releu	1
Suport 4 baterii	1
Baterie 3.7V Li-Ion	4
LED	2
Rezistor	2
Fire tata-tata	
Fire mama-tata	
Fire mama-mama	

Schema electrica

[Safe deposit schematic](#)

Conectarea Keypad-ului

Conectarea Keypad-ului a presupus conectarea celor 8 pini ai Keypad-ului la pinii de la 2 la 9 de pe Arduino. Cei 8 pini ai Keypad-ului provin de la faptul ca fiind o matrice tastatura, aceasta are 4 linii si 4 coloane, fiecare linie avand conectate 4 diode si fiecare coloana avand conectate 4 butoane.

Conectarea LCD 16x2 I2C

Acest tip simplist de LCD, destul de micut, se conecteaza la Arduino folosind interfata seriala I2C. Mai exact, acesta vine cu 4 pini: VCC, GND, SDA si SCL. Toti cei 4 pini vor fi conectati la pinii corespunzatori de pe Arduino. LCD-ul a venit si cu modulul I2C inclus, astfel ca nu a fost nevoie de a face legaturile dintre pinii de pe LCD manual.

Inainte de a incepe programul, utilizatorul LCD-ului este nevoit sa afle adresa LCD-ului pentru a se putea comunica cu acesta. Dupa ce am aflat adresa acestuia, o vom seta folosind functia "LiquidCrystal_I2C Lcd(address, 16, 2)".

Conectarea LED-urilor & Buzzer

Pentru conectarea LED-urilor s-au folosit 2 rezistente de PULLUP si pinii 11 si 12 de pe Arduino. Pentru conectarea buzzer-ului a fost folosit pinul 10.

Conectarea modulului RTC DS3231

Pentru conectarea modulului RTC, modelul DS3231, am avut in primul rand nevoie de o baterie CR2032. Acesta va comunica cu Arduino prin interfata seriala I2C si asemanator cu modulul LCD, a fost nevoie de conectarea celor 4 pini de care dispune (VCC, GND, SDA si SCL) la pinii corespunzatori de pe Arduino.

Pentru a asigura conexiunea cu acesta, am folosit biblioteca "RTCLib.h" impreuna cu apelul functiei "now()" ce returneaza o variabila de tipul "DateTime" de unde putem extrage ora, minutul, numarul de secunde, data curenta.

Conectarea modulului releu

Releul este prevazut cu doua tipuri de pini:

- Pini de joasa tensiune
 - VCC
 - GND
 - IN

- Pini de inalta tensiune
 - NC: normally closed pin
 - COM: common pin
 - NO: normally open pin

Pentru conectarea releului la Arduino va trebui sa se realizeze conectarea pinilor de tensiune joasa (VCC, GND si IN) la pinii corespunzatori de pe Arduino. Pin-ul IN va fi conectat la pinul analogic A3 care va fi de OUTPUT.

Pinii de tensiune inalta vor fi conectati astfel:

- Pin-ul COMM va fi conectat la borna pozitiva a solenoidului folosindu-se un fir intermediar tata-tata.
- Pin-ul NO va fi conectat la borna pozitiva a celor 4 baterii de 3.7V Li-Ion.
- Pin-ul NC va ramane neconectat deoarece dorim doar deschiderea solenoidului in mod normal, acest pin fiind de inchidere.

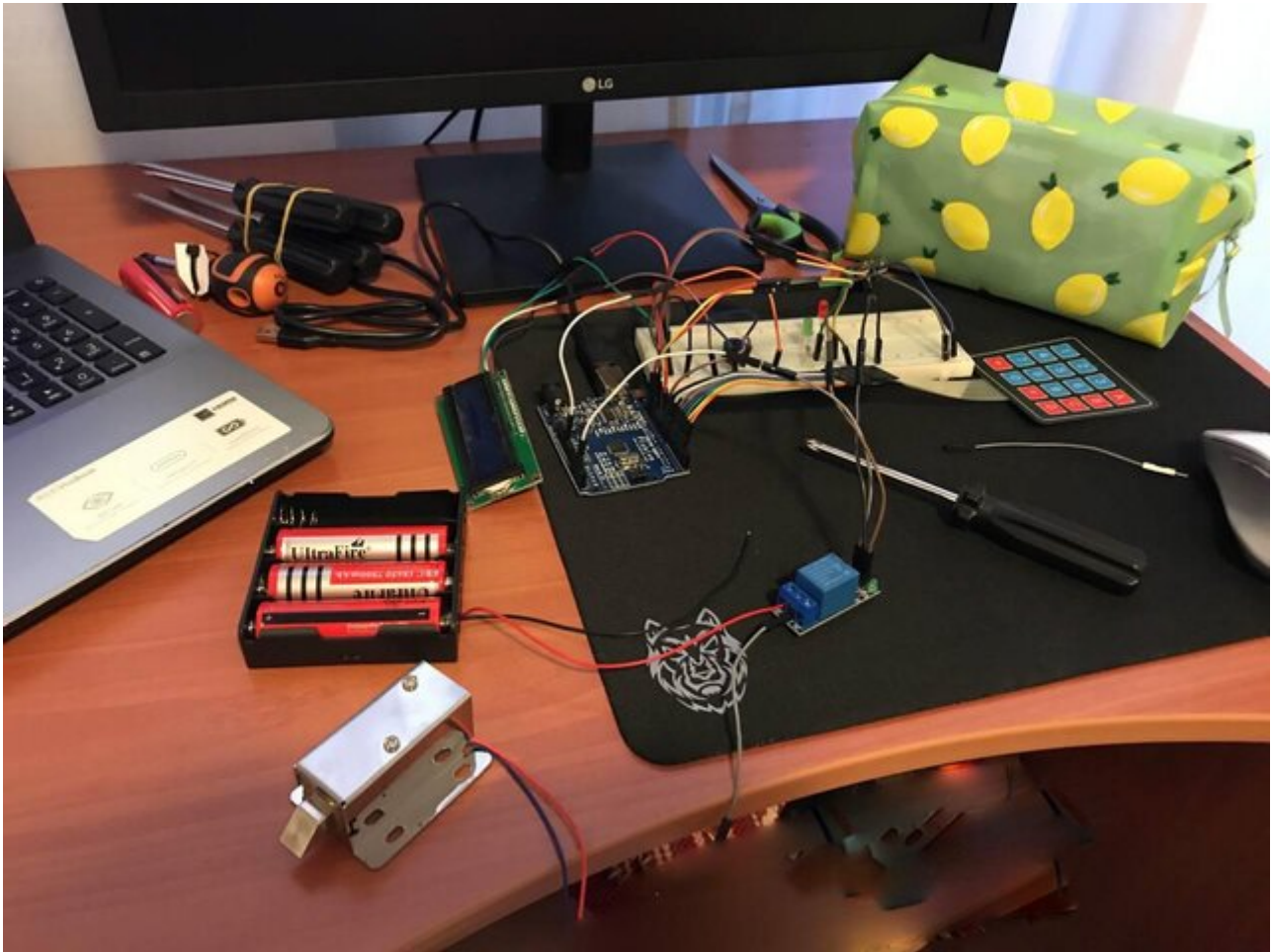
Conectarea incuietorii (solenoid)

Pentru actionarea solenoidului am folosit un releu care alimentat de cele 4 baterii va genera un camp electromagnetic care va deschide/inchide incuietoarea. Cele doua borne ale solenoidului sunt conectate astfel:

- Borna negativa este conectata la borna negativa a celor 4 baterii pozitionate in serie.
- Borna pozitiva este conectata la NO pin al releului.

Rezultatul initial

Rezultatul initial este format doar din conectarea tuturor componentelor si functionarea corecta a programului incarcat pe Arduino, dar fara niciun fel de package.



Software Design

- Limbaj de programare: C
- Mediu de dezvoltare si compilare: Arduino IDE

Implementare

Biblioteci folosite

- [Keypad.h](#): Folosita pentru conectarea Keypad-ului
- [Wire.h](#): Folosita pentru a se putea comunica prin interfata seriala I2C cu anumite componente
- [LiquidCrystal_I2C.h](#): Folosita pentru a se face comunicarea cu modulul LCD 16x2
- [RTClib.h](#): Folosita pentru a se putea realiza comunicarea cu modulul RTC DS3231.

Program

Ideea implementarii programului este urmatoarea:

1. Functia "setup()":

- Se realizeaza initializarea modulului RTC folosind functia "RTC_module_handler()" care va verifica daca exista conectat acest modul si il va initializa cu un anumit format de data (la alegerea fiecarui utilizator).
- Se va seta frecventa interfetei "Serial" de pe Arduino folosita pentru afisarea parolei dinamice a seifului.
- Se vor seta pini corespunzatori LED-urilor, buzzer-ului si solenoidului de tip pini de OUTPUT.
- Se va extrage momentul actual al zilei folosindu-se modulul RTC si se va obtine o parola folosindu-se o functie de hash aplicata asupra momentului din zi curent, lucru realizat cu ajutorul functiei "set_password(moment, pass_code)".

2. Functia "loop()":

- Vor fi afisate pe ecranul LCD regulile seifului in legatura cu lungimea parolei si tasta care trebuie apasata pentru a se efectua verificarea parolei (tasta'D') prin intermediul functiei "lcd_setup()".
- Se va face verificarea parolei propriu-zis:
 - Se va verifica fiecare caracter introdus de la Keypad existand trei cazuri:
 - In cazul in care caracterul introdus de catre utilizator este corect se va trece la urmatorul caracter fara a se sta o variabila boolean ce contorizeaza faptul ca un caracter introdus este gresit sau nu.
 - In cazul in care caracterul introdus este gresit se va seta variabila de tip boolean cu valoarea "true" (deci parola introdusa de utilizator este gresita).
 - Exista cazul in care utilizatorul, din neatenție, poate tasta gresit un caracter al parolei, asa ca exista optiunea de stergere a caracterului anterior folosindu-se tasta 'C'. Astfel se ofera o a doua sansa la a putea deschide seiful 😊.
- In final, se verifica daca parola introdusa este corecta prin verificarea variabilei de tip boolean:
 - Daca parola este corecta:
 - LED-ul verde se va aprinde
 - Buzzerul va genera o melodie
 - Incuietoarea pozitionata pe usita seifului va fi actionata astfel ca usita seifului va putea fi deschisa
 - In cazul in care dorim sa resetam seiful si usita seifului sa fie incuiata din nou, se va apasa tasta 'C'.
 - Daca parola este gresita:
 - LED-ul rosu se va aprinde
 - Buzzerul va genera un ton de eroare simplist
 - Usita seifului va ramane inchisa, neputand fi deschisa, iar utilizatorul va trebuie sa astepte 5 secunde pentru a incerca iar sa introduca parola. La trei incercari consecutive esuate de introducere a parolei, seiful se va bloca timp de 15 secunde.

3. Functii aditionale:

- Functia "set_password(moment, password)":
 - Este functia ce realizeaza aplicarea unui hash asupra momentului zilei in ora, minute si secunde dat de modulul RTC.
 - Este generat cate un caracter din cele 6 ale parolei folosindu-se operatii matematice aplicate asupra numerelor ce reprezinta ora, minutul si secundele din momentul zilei.
 - Sunt folosite atat ridicari la putere, cat si inmultiri, impartiri, adunari si verificari daca de exemplu ora data de modulul RTC este divizibila cu anumite numere prime, in fiecare caz existand un anumit cod hash, astfel asigurandu-ne ca un anumit moment al zilei va avea un hash unic.

Rezultate Obținute

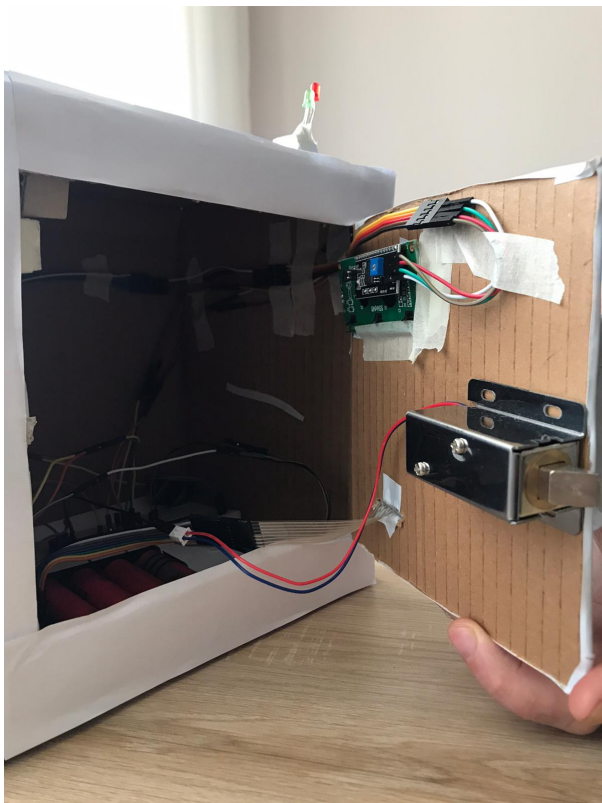
Design-ul exterior

Acesta este design-ul seifului privit din fata si din spate.





Am folosit o cutie de carton inainturul careia am reusit sa incorporez toate componentele care nu trebuia sa fie neaparat vizibile. Am scos cele doua LED-uri in afara cutiei iar Keypad-ul si ecranul LCD au fost plasate pe usita seifului impreuna cu un maner cu care putem deschide usor usita.





Rezultatul final

Am obtinut in final un mini-seif functional construit treptat, initial fiind un proiect mult mai simplist, incuietoarea si releul impreuna cu bateriile fiind achizionate ulterior, dorindu-se a iesi ceva mai complex decat un simplu mecanism de verificare a parolelor.





Functionare

[Video Demo Safe-Deposit Box](#)

Concluzii

- Proiectul este realizat in intregime si functional.
- Initial s-a dorit a fi un mecanism simplit de verificare a parolelor, dar ulterior au fost adaugate solenoidul, releul si cele 4 baterii 3.7V Li-Ion.
- A fost foarte interesant tot procesul de realizare a sa (din punct de vedere hardware, conectarea componentelor si research-ul facut despre toate acestea), chiar daca in final de la prea multe teste aplicatie asupra sa am reusit aproape sa stric cablul de alimentare dintre Arduino si laptop (inca este functional, dar cu greu).
- Procesul de confectionare al cutiei a fost costisitor din punct de vedere al timpului insa cred ca a meritat.

Download

Sursele pot fi gasite [aici](#).

Jurnal

- 23/04: alegere tema proiect.
- 25/04: creare pagina Wiki, descriere, diagrama bloc, lista componente hardware (sumar).
- 02/05: achizitionarea primelor componente (ecran LCD, Keypad, modul RTC, fire, baterie pentru modulul RTC).
- 08/05: inceperea lucrului la proiect
- 12/05: achizitionarea solenoidului, releului, suportului pentru cele 4 baterii impreuna cu acestea.
- 14/05: adaugarea noilor componente la proiectul initial.
- 26/05: realizarea cutiei in care vor fi incorporate toate componentele proiectului.
- 22/05: revizuirea codului, filmarea videoclipului demo.
- 23/05: realizarea paginii de wiki finale pentru proiect.

Bibliografie/Resurse

- <https://www.arduino.cc/reference/en/libraries/rtclib/>
- <https://www.arduino.cc/reference/en/language/functions/communication/serial/>
- <https://create.arduino.cc/projecthub/akshayjoseph666/interface-i2c-16x2-lcd-with-arduino-uno-just-4-wires-273b24>
- <https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/>
- <https://arduinogetstarted.com/tutorials/arduino-relay>
- https://www.youtube.com/watch?v=0wYp7rd4llo&ab_channel=cytrontech
- <https://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/>
- <https://create.arduino.cc/projecthub/akshayjoseph666/interface-i2c-16x2-lcd-with-arduino-uno-just-4-wires-273b24>
- <https://learn.adafruit.com/adafruit-ds3231-precision-rtc-breakout/arduino-usage>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab0-2021>
- <https://www.arduino.cc/en/Tutorial/BuiltInExamples/toneMelody/>
- <https://github.com/robsoncouto/arduino-songs>

PDF

Mecanism de verificare a parolelor

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/mecanism_verificare_parola

Last update: **2021/05/25 14:28**

