

SecureBox

Introducere

Proiectul "SecureBox" consta in realizarea unui seif electronic inteligent, care permite accesul doar utilizatorilor autorizati prin intermediul unui card RFID. Sistemul este controlat de un microcontroler Arduino Nano, care gestioneaza un modul RFID RC522 pentru identificare, un mecanism de inchidere electronica (yala) activat printr-un releu, si un afisaj LCD I2C pentru feedback vizual.

Descriere generală

Sistemul "SecureBox" este format din următoarele module principale:

- Modul de citire RFID (RC522) – citește un card/tag RFID de acces.
- Microcontroler (Arduino NANO) – gestionează logica de control și comunică cu toate celelalte componente.
- Yală electromagnetică – acționează mecanismul fizic al încuietorii.
- Afișaj LCD 16x2 (opțional) – afișează mesaje de stare pentru utilizator.
- Sursă de alimentare (9V sau baterie) – asigură tensiunea pentru aproape toate componentele.
- Cutie fizică (seif) – în care sunt montate toate componentele.



Descriere interacțiuni:

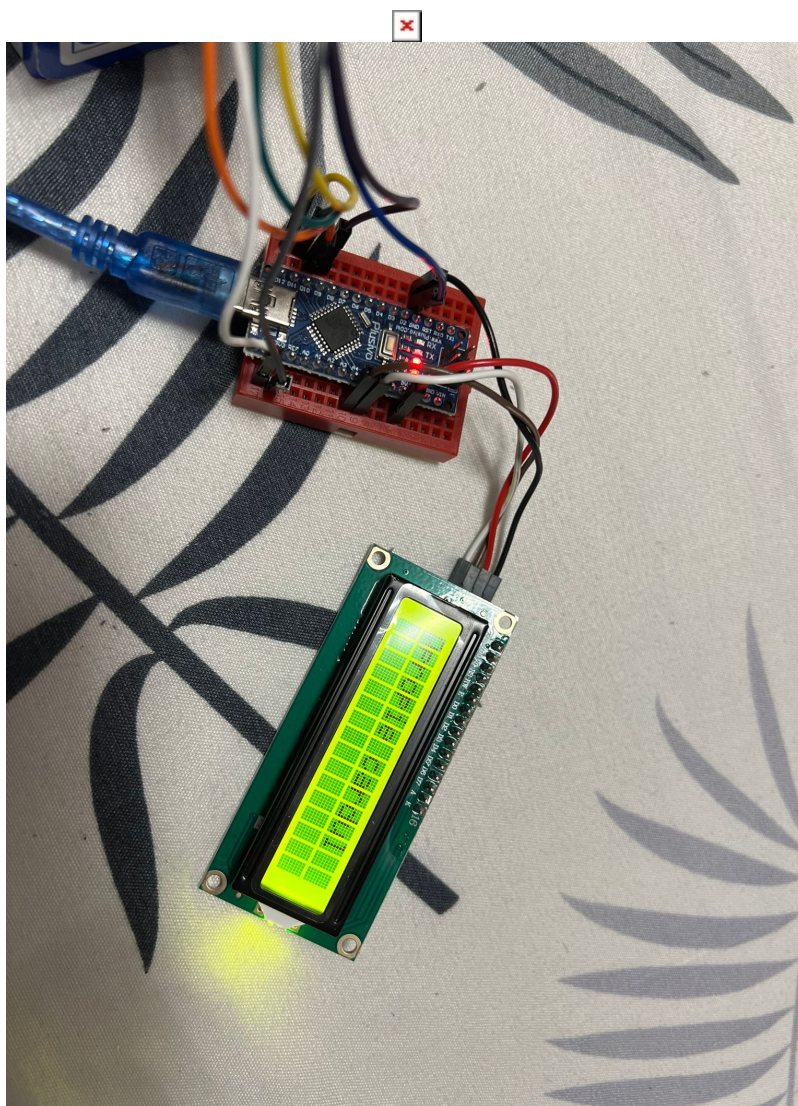
- Utilizatorul apropie un card RFID de cititorul RC522.
- RC522 trimite UID-ul cardului către Arduino prin SPI.
- Arduino verifică dacă UID-ul este valid.
- Dacă este, acționează yala pentru a debloca ușa.
- Afișează un mesaj de operațiune realizată cu succes.
- Dacă nu este valid, ușa rămâne blocată și se afișează un mesaj de eroare.

Hardware Design

Listă de piese:

- 1x Arduino Nano
- 1x Modul RFID RC522 + card și tag
- 1x Yală electromagnetică 12V

- 1x Breadboard + fire de conexiune
- 1x LCD 16x2
- 1x Sursă 9V (USB, baterie sau alimentator extern)
- 1x Ridicator de tensiune 9V → 12V pentru a actiona yala electrica
- 1x Releu pentru a actiona yala cu ajutorul Arduino Nano
- 2x Butoane
- 1x Cutie



Proiectul folosește un Arduino Nano pentru a controla o yala electromagnetă de 12V printr-un modul releu de 5V. Accesul este permis doar cardurilor RFID autorizate, citite de un modul MFRC522 conectat prin SPI. Un ecran LCD 16x2 I2C afișează mesaje de stare (ex: „Acces permis” / „Acces respins”).

Sistemul este alimentat cu 9V prin pinul Vin, iar yala primește 12V prin ridicator de tensiune și releu. Pentru interacțiune:

Un buton comută între modul de citire și scriere carduri.

Alt buton permite dezactivarea yalei manual.

Software Design

Biblioteci utilizate: <Wire.h>, <LiquidCrystal_I2C.h>, <SPI.h>, <MFRC522.h>

Github: github.com/robertenciu/SecureBox

1. Setup()

Această funcție este apelată o singură dată la pornirea microcontrolerului. În cadrul ei:

Se inițializează comunicațiile Serial, SPI și RFID.

Se configurează pinul pentru releu ca ieșire și este setat HIGH (yală închisă).

Se configurează butonul de mod ca intrare cu rezistență de pull-up.

Se pornește LCD-ul și se afișează starea curentă (Secure Box, Mod: ...).

Se inițializează vectorul care conține UID-urile autorizate (storedUIDs) cu valoarea 0xFF (memorie "goală").

2. loop()

Această funcție rulează continuu și:

Detectează apăsarea butonului de mod și comută între cele 3 moduri:

Mod 0: CITIRE

Mod 1: SCRIERE

Mod 2: ȘTERGERE

Evită afișarea repetitivă pe LCD dacă modurile nu se schimbă.

Detectează o nouă cartelă RFID și validează dacă are un UID de lungime corectă.

Afișează UID-ul detectat pe LCD.

În funcție de modul activ:

CITIRE: Verifică dacă UID-ul este autorizat și deschide temporar yala dacă da.

SCRIERE: Adaugă UID-ul în listă dacă nu este deja salvat și dacă mai este spațiu.

ȘTERGERE: Elimină UID-ul dacă există în listă.

După fiecare operație, se actualizează LCD-ul și se reafișează starea principală (Secure Box, Mod: ...).

3. **initUIDs()**

Inițializează toți vectorii de UID din memorie cu valoarea 0xFF, semnalând că sloturile sunt goale. De asemenea, resetează contorul UID-urilor salvate.

4. **findEmptySlot()**

Caută un slot liber în vectorul storedUIDs și returnează indexul unde poate fi salvat un nou UID. Dacă toate sloturile sunt ocupate, returnează MAX_UIDS.

5. **compareUIDs(byte* uid1, byte* uid2)**

Compară două UID-uri (de tip vector de byte) și returnează true dacă sunt identice, altfel false.

6. **isUIDAuthorized(byte* uid)**

Verifică dacă un UID scanat există deja în lista UID-urilor salvate (adică este autorizat).

7. **saveUID(byte* uid)**

Salvează un nou UID în vectorul storedUIDs la primul slot liber și crește contorul savedUIDCount.

8. deleteUID(byte* uid)

Caută și elimină un UID existent din lista de autorizări, adică setează slotul său din vector înapoi la 0xFF și decrementează savedUIDCount.

9. afiseazaModPeLCD()

Actualizează LCD-ul pentru a reflecta modul curent de funcționare: CITIRE, SCRIERE sau STERGERE, împreună cu titlul Secure Box.

Concepte folosite:

===== SPI (Serial Peripheral Interface) =====

Este folosit pentru comunicarea cu cititorul RFID MFRC522. Comunicarea SPI este inițiată cu SPI.begin() și folosită intern de biblioteca MFRC522 pentru a transmite comenzi și a primi date de la tagurile RFID.

===== I2C (Inter-Integrated Circuit) =====

Este utilizat pentru a controla afișajul LCD cu modul I2C. Acesta comunică prin pini SDA și SCL pentru a trimite text și comenzi către ecranul LCD. Biblioteca LiquidCrystal_I2C gestionează aceste comunicări.

===== UART (Universal Asynchronous Receiver/Transmitter) =====

Comunicarea serială prin Serial.begin(9600); este un exemplu de utilizare UART, unde microcontrolerul trimite datele la un terminal sau la un alt dispozitiv pentru debug sau afișare.

===== GPIO (General Purpose Input/Output) =====

Este folosit pentru interacțiunea cu butonul de schimbare mod și controlul releului conectat la yală.

- Pin A0 (RELAY_PIN) - Configurat ca ieșire pentru a activa/dezactiva yală.
- Pin D2 (BUTTON_MODE_PIN) - Configurat ca intrare cu pull-up pentru a detecta apăsarea butonului.

Rezultate Obținute

[Video testare](#)

Jurnal

Etape:

#1 - Testarea ecranului LCD

Am inceput cu testarea ecranului LCD 16x2 cu interfata I2C. Initial nu afisa nimic, dar dupa ce am verificat conexiunile (VCC, GND, SDA – A4, SCL – A5 pe Arduino Nano) si am folosit biblioteca LiquidCrystal_I2C, a functionat.

#2 - Modulul RFID MFRC522

Am conectat si testat modulul RFID. Am folosit biblioteci precum MFRC522 si l-am conectat prin SPI la Arduino (SS pe D10, RST pe D9, MOSI/MISO/SCK, etc). A citit cu succes cardurile venite in kit, dar cardul de la interfonul blocului nu a fost detectat(probabil foloseste un alt standard). Citirea UID-ului a fost afisata si pe LCD.

#3 - Releu si yala de 12V

Am testat modulul releu pentru a controla o yală electrică de 12V. Releul este de tip active LOW, deci este activat cand pinul este pus pe LOW. Initial am folosit un alimentator de 9V 0.6A cu un ridicator la 12V, dar mi-am dat seama ca este prea putin pentru ce am eu nevoie asa ca in final am folosit un alimentator de 12V 1.5A, care ofera suficient curent pentru yala ce consuma 0.4A. Yala a fost conectata pe terminalele NO și COM ale releului. Releul este comandat de un pin digital (A0) de pe Arduino.

#4 - Butoane

Am adaugat un buton pentru comutarea intre modul de citire și scriere tag RFID. Am vrut sa am si un buton pentru dezactivarea yalei dar am renuntat la el datorita faptului ca yala are un fel de suprafata oblica pe care poate sa alunece inauintrul cutiei si astfel sa se inchida fara a mai dezactiva-o. Yala se inchide la 3 secunde de functionare.

#5 - Cutie

In final am achizitionat o cutie in care sa plasez toate aceste componente.

Bibliografie/Resurse

Github – MFRC522 Library: <https://github.com/miguelbalboa/rfid>

Github – LiquidCrystal_I2C Library: https://github.com/johnrickman/LiquidCrystal_I2C

Arduino Forum: <https://forum.arduino.cc>

How to Use an RFID Module with Arduino for Beginners:

<https://www.youtube.com/watch?v=lg8HRY8q004>

I2C-LCD Tutorial: <https://lastminuteengineers.com/i2c-lcd-arduino-tutorial>

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2025/ccristi/robert_alex.enciu



Last update: **2025/05/26 13:26**