



# Proiectare cu micropresesoare

Proiect:  
Sistem de acces automatizat

**Student:** Mulesă Alexandru

**Grupa:** 30232

## Cuprins

1. Introducere .....	3
1.1 Context și propunere de proiect .....	3
1.2 Definirea problemei.....	3
2. Soluția problemei și descrierea proiectului.....	4
2.1 Descrierea soluției .....	4
2.2 Componente necesare .....	4
2.3 Codul sursă și logica programului.....	8
2.4 Ghid de utilizare .....	16
3. Schema circuitului .....	17
4. Poze ale proiectului .....	18
5. Bibliografie.....	19

# 1. Introducere

## 1.1 Context și propunere de proiect

În contextul actual al dezvoltării conceptului de “Smart Home” și al automatizării activităților cotidiene, securitatea și controlul accesului au devenit aspecte esențiale pentru eficientizarea sistemelor moderne. Astfel, proiectul își propune realizarea unui sistem de acces securizat, cu autentificare RFID și încuietoare mecanică automatizată, utilizând placă de dezvoltare Arduino UNO R3.

Sistemul este conceput pentru a înlocui sau a suplimenta mecanismele de închidere tradiționale, oferind o metodă de deblocare *contactless* (fără contact), bazată pe tehnologia identificării prin radiofrecvență (*RFID - Radio Frequency Identification*). Acesta integrează o componentă de execuție mecanică (un zăvor actionat de un servomotor), o interfață vizuală (display LCD, LED-uri) și una acustică (buzzer) pentru a oferi un feedback cât mai complet utilizatorului.

## 1.2 Definirea problemei

Sistemele de acces bazate pe chei fizice prezintă o serie de dezavantaje critice care justifică necesitatea implementării unei soluții automatizate, oferind un grad superior de securitate. O parte dintre aceste dezavantaje sunt:

- **Administrarea deficitară a cheilor:** Pierderea sau duplicarea neautorizată a cheilor mecanice compromite securitatea unei locații.
- **Lipsa monitorizării și a feedback-ului:** O încuietoare clasică nu poate comunica utilizatorului starea sa (blocat/deblocat) și nici nu poate avertiza în cazul în care s-a încercat accesul neautorizat.
- **Vulnerabilitatea mecanică:** Zăvoarele mecanice simple pot fi acționate manual de către oricine. Integrarea lor într-un sistem electronic permite blocarea activă și controlată.
- **Dificultatea în gestionarea utilizatorilor:** Într-un mediu cu mai mulți utilizatori, revocarea accesului unei singure persoane necesită schimbarea întregii încuietori, în timp ce într-un sistem bazat pe RFID, acest lucru se realizează printr-o simplă modificare la nivel de cod software.

## 2. Soluția problemei și descrierea proiectului

### 2.1 Descrierea soluției

Soluția propusă constă într-un sistem de securitate electronic capabil să controleze un zăvor mecanic prin intermediul unui servomotor. Sistemul utilizează un cititor RFID pentru a scană carduri/tag-uri, comparând codul unic (*UID*) citit cu o valoare stocată în memoria programului. În funcție de validitatea cardului, microcontrolerul Arduino coordonează deschiderea fizică a zăvorului și oferă feedback vizual (prin LCD și LED-uri) și sonor (prin buzzer).

### 2.2 Componente necesare

Pentru realizarea proiectului, au fost utilizate următoarele componente:

- **Arduino UNO R3:** placă de dezvoltare (unitatea centrală a sistemului)



- **Modul RFID RC522:** modul de scanare a cardurilor/tag-urilor



- **Display LCD 16x2, cu convertor I2C:** ecran pentru afișarea mesajelor de status, folosește convertorul I2C pentru a comunica cu placa de dezvoltare



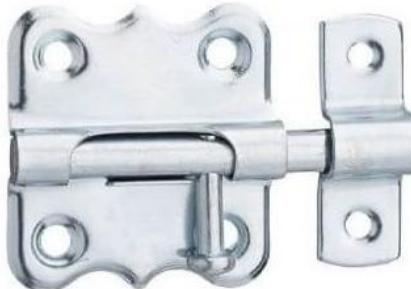
- **Buzzer activ:** folosit pentru generarea semnalelor acustice de avertizare



- **Servomotor SG90:** motor de precizie folosit pentru acționarea mecanică a încuietorii (tragerea/retragerea zăvorului mecanic, corespunzător stărilor deschis/inchis)



- **Zăvor mecanic:** zăvor metalic folosit pentru a simula mecanismul real de închidere/deschidere a unei uși



- **Sârmă de cupru:** folosită pentru a lega zăvorul de brațul servomotorului
- **LED roșu 3mm:** pentru indicarea respingerii accesului



- **LED verde 3mm:** pentru indicarea deblocării încuietorii și permiterea accesului



- **2 rezistoare de carbon, 220Ω, 0.25W:** folosite pentru limitarea intensității curentului electric, pentru evitarea arderii LED-urilor



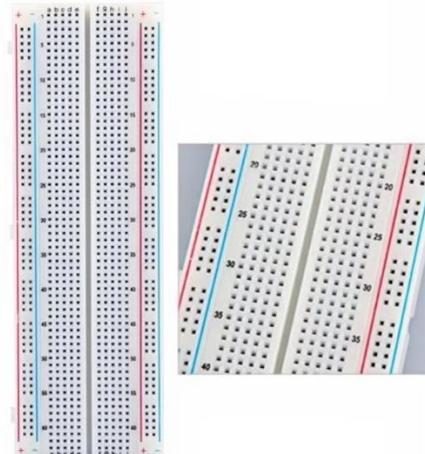
- **Condensator, 220  $\mu$ F, 10V:** folosit pentru stabilizarea tensiunii în momentul consumului brusc de curent al servomotorului (la pornire)



- **Fire Dupont tată-tată și mamă-tată:** folosite pentru interconectarea componentelor din circuit



- **Breadboard (placă de prototipizare):** folosită pentru interconectarea rapidă a componentelor fără lipire, permitând distribuirea alimentării (5V și GND) către multiple dispozitive simultan, precum și montarea rezistoarelor, a LED-urilor și a condensatorului în circuit



- **Cablu USB Type-A to Type-B:** asigură dubla funcționalitate de alimentare a circuitului cu o tensiune constantă de 5V de la calculator și de transfer al codului sursă de la mediul de dezvoltare (Arduino IDE) către memoria Flash a microcontrolerului



- **Supor din plexiglas (grosime 4mm):** folosit pentru fixarea rigidă a tuturor componentelor, asigurând stabilitatea structurală a sistemului



## 2.3 Codul sursă și logica programului

Implementarea software a fost împărțită în două etape:

- a) Identificarea codului UID al tag-ului folosit pentru acces (fișierul *ScanUID.ino*)
- b) Scrierea codului sursă pentru logica de acces (fișierul *project.ino*)

Codul sursă poate fi accesat la următoarea adresă:

<https://github.com/AlexMules/Arduino-RFID-Door-Lock>

a) *ScanUID.ino*

Acest prim program are rolul de a extrage “cheia digitală” (codul *UID – Unique Identifier*) a tag-ului RFID folosit pentru acces. UID-ul va fi folosit ulterior în codul sursă al programului principal.

Pentru a permite comunicarea dintre microcontroler și modulele periferice, au fost incluse următoarele biblioteci:

- **SPI.h** – Gestionă protocolul SPI, necesar pentru comunicarea între placa de dezvoltare și modulul RFID.
- **MFRC522.h** – Conține funcțiile specifice pentru controlul cititorului RFID, permitând inițializarea senzorului și decodificarea semnalelor primite de la carduri/tag-uri
- **LiquidCrystal\_I2C.h** - Permite controlul ecranului LCD prin protocolul I2C, utilizând doar doi pini (SDA și SCL) pentru afișarea informațiilor

În cadrul codului am definit 2 obiecte pentru a controla componente fizice:

- **mfrc522(SS\_PIN, RST\_PIN)** – instanțiază cititorul RFID utilizând pinii digitali 10 (Slave Select) și 9 (Reset)
- **Lcd(0x27, 16, 2)** – instanțiază ecranul LCD la adresa I2C 0x27, configurându-l pentru 16 coloane și 2 rânduri

În funcția *setup()* se folosește funcția *PCD\_Init()* pentru a activa modulul RFID și magistrala SPI. De asemenea, se execută *PCD\_DumpVersionToSerial()*, o funcție care confirmă inițializarea corectă a modulului prin afișarea versiunii de firmware în *Serial Monitor*.

Firmware Version: 0x92 = v2.0

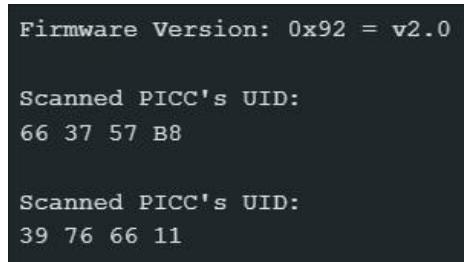
În funcția *loop()* se verifică constant prezența unui card prin *PICC\_IsNewCardPresent()*. Dacă un card este detectat, se încearcă citirea datelor

acestuia prin *PICC\_ReadCardSerial()*. Dacă citirea eşuează, programul revine la începutul buclei.

Cea mai importantă parte a programului este reprezentată de bucla de extragere a octetilor. Codul UID este stocat de către senzorul RFID într-un tablou de date numit *uidByte*. Se utilizează o structură repetitivă *for* care iterează de patru ori pentru a parcurge toți octetii cardului. În cadrul acestei bucle, fiecare octet este salvat în array-ul *readCard* și trimis simultan către două ieșiri: ecranul LCD și Serial Monitor.

În Serial Monitor, datele apar în format hexazecimal pentru a fi identificate ușor de către programator. Pe ecranul LCD, informația este afișată pe al doilea rând prin intermediul variabilei *columnPos*. Această variabilă crește cu trei unități la fiecare pas pentru a spația octetii și a-i face lizibili. Am introdus o pauză de tip *delay(500)* între citiri pentru a oferi un feedback vizual progresiv utilizatorului. La final, funcția *PICC\_HaltA()* oprește comunicarea cu cardul actual. Astfel, sistemul revine în starea de așteptare pentru a permite o nouă scanare.

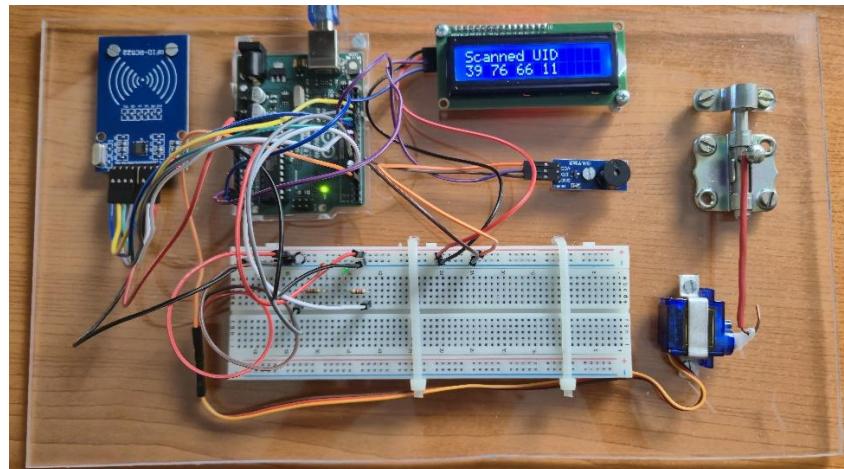
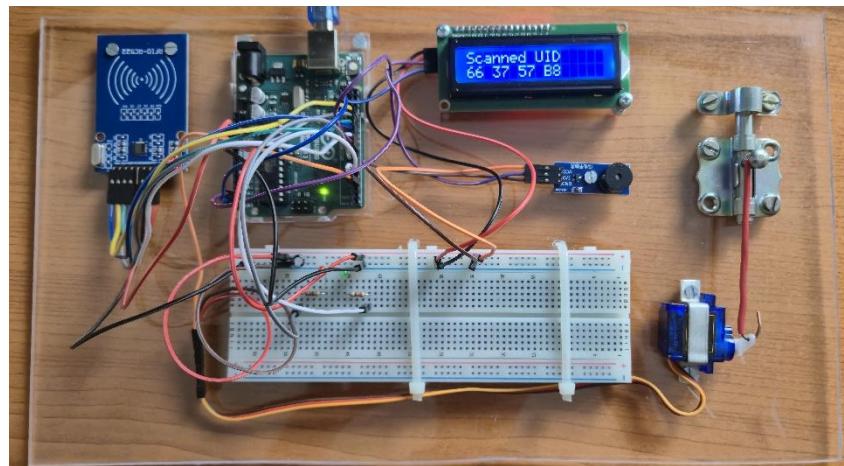
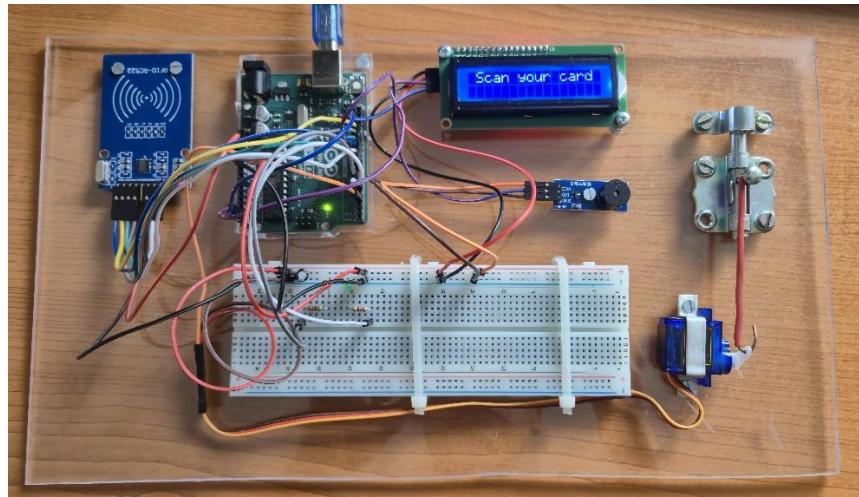
```
// bucla pentru citirea codului UID al cardului de acces
columnPos = 0;
Serial.println(F("Scanned PICC's UID:"));
for (uint8_t i = 0; i < 4; i++) { // 
    readCard[i] = mfrc522.uid.uidByte[i];
    Serial.print(readCard[i], HEX);
    Serial.print(" ");
    lcd.setCursor(columnPos, 1);
    lcd.print(readCard[i], HEX);
    lcd.print(" ");
    delay(500);
    columnPos += 3;
}
Serial.println("\n");
mfrc522.PICC_HaltA();
```



```
Firmware Version: 0x92 = v2.0

Scanned PICC's UID:
66 37 57 B8

Scanned PICC's UID:
39 76 66 11
```



### b) *project.ino*

Acest program gestionează logica funcționării sistemului, coordonând citirea datelor, validarea acestora și acționarea elementelor de execuție.

Pentru a facilita comunicarea dintre microcontroler și modulele periferice, au fost incluse următoarele biblioteci:

- **Servo.h** - Permite controlul precis al servomotorului SG90 pentru acționarea zăvorului.
- **SPI.h** – Gestionează protocolul SPI, necesar pentru comunicarea între placa de dezvoltare și modulul RFID.
- **MFRC522.h** – Conține funcțiile specifice pentru controlul cititorului RFID, permitând inițializarea senzorului și decodificarea semnalelor primite de la carduri/tag-uri.
- **LiquidCrystal\_I2C.h** - Permite controlul ecranului LCD prin protocolul I2C, utilizând doar doi pini (SDA și SCL) pentru afișarea informațiilor.

Pinii digitali sunt folosiți astfel:

- **5 și 6** → pentru LED-uri (5 – LED verde, 6 – LED roșu)
- **8** → pentru buzzer
- **9 și 10** → pinii de control pentru modulul RFID (9 – RST pin, 10 – SS pin)

```
#define SS_PIN 10
#define RST_PIN 9
#define GREEN_LED 5
#define RED_LED 6
#define BUZZER 8
```

Obiecte create:

- **servo** – instanțiază obiectul necesar pentru gestionarea mișcării și poziționării servomotorului

- **lcd(0x27, 16, 2)** - instanțiază ecranul LCD la adresa I2C *0x27*, configurându-l pentru 16 coloane și 2 rânduri
- **rfid(SS\_PIN, RST\_PIN)** - instanțiază cititorul RFID utilizând pinii digitali 10 (Slave Select) și 9 (Reset)

Constante folosite:

- **UID** – sir de caractere ce stochează codul de autorizare "**66 37 57 B8**", extras în etapa anterioară
- **lockedAngle** și **unlockedAngle** – sunt definite unghiurile servomotorului pentru controlul zăvorului ( $70^\circ$  pentru poziția închis și  $120^\circ$  pentru deschis)

Funcția *setup()* configurează starea inițială a sistemului, pornind comunicarea serială la 9600 *baud rate* și setând pinii LED-urilor și ai buzzer-ului ca ieșiri digitale. Pentru a asigura securitatea imediat după alimentare, LED-urile sunt stinse, buzzer-ul este dezactivat iar servomotorul este poziționat automat la  $70^\circ$  pentru a bloca zăvorul. În final, sunt inițializate ecranul LCD și modulul RFID prin protocoalele I2C și SPI, pregătind întregul hardware pentru etapa de monitorizare și scanare.

Funcția *loop()* implementează logica de monitorizare a accesului, asigurând comunicarea constantă între cititorul RFID, microcontroler și elementele de execuție. Procesul începe prin afișarea unui mesaj de întâmpinare pe ecranul LCD, sistemul rămânând într-o stare de așteptare până când funcția *PICC\_IsNewCardPresent()* detectează prezența unui obiect în proximitatea antenei. În momentul detectării, programul încearcă să citească datele seriale ale cardului. În caz de eșec, acesta trimită un mesaj de eroare către Serial Monitor și reia ciclul de monitorizare.

Dacă citirea este realizată cu succes, ecranul este curățat pentru a afișa textul „*Scanning*”, în timp ce o buclă *for* parcurge cei patru octeți ai codului unic (*UID*). În cadrul acestei bucle, fiecare octet este convertit într-un format hexazecimal și concatenat într-un sir de caractere numit *ID*. Procesul este raportat simultan către Serial Monitor pentru *debugging*, sub forma "*NUID tag is:* ". Pentru o experiență de utilizare interactivă, pe LCD apar puncte de progres, iar formatarea finală a ID-ului este standardizată prin transformarea literelor în majuscule. Astfel, se asigură compararea precisă cu valoarea de autorizare stocată în constanta *UID*.

```

NUID tag is: 66 37 57 B8
NUID tag is: 39 76 66 11
NUID tag is: 66 37 57 B8
NUID tag is: 39 76 66 11

```

Logica decizională se ramifică în funcție de validitatea cardului scanat. În cazul în care ID-ul corespunde valorii "66 37 57 B8", sistemul execută procedura de acces: se aprinde LED-ul verde, buzzer-ul emite un scurt semnal sonor de confirmare, iar servomotorul acționează zăvorul prin rotația la 120°. Mesajul „Access Granted” rămâne afișat pe LCD timp de 10 secunde, oferind utilizatorului timp pentru acces, după care sistemul revine automat la starea de blocare (70°).

```

// logica de acces
if (ID.substring(1) == UID) { // card corect
    digitalWrite(GREEN_LED, HIGH);

    // un scurt beep de confirmare acces
    digitalWrite(BUZZER, LOW);
    delay(200);
    digitalWrite(BUZZER, HIGH);

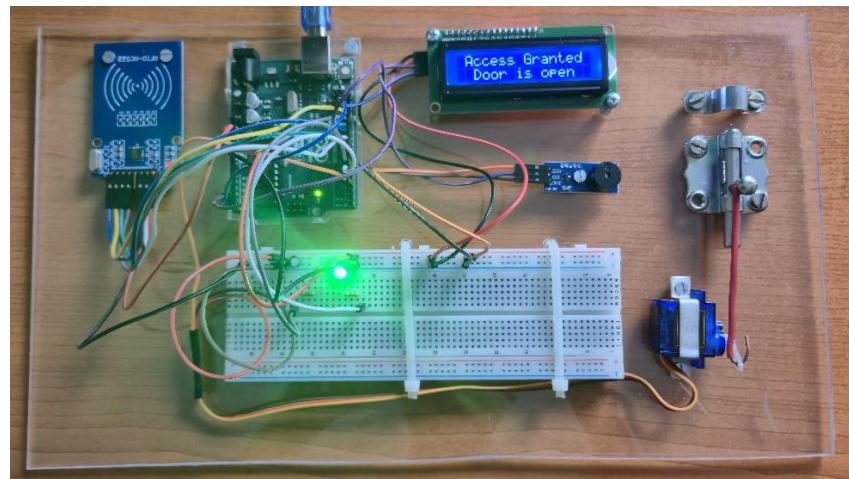
    // deschidere incuietoare
    servo.write(unlockedAngle);
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("Access Granted");
    lcd.setCursor(2, 1);
    lcd.print("Door is open");

    delay(10000); // incuietoarea ramane deschisa 10 secunde pentru a permite accesul

    servo.write(lockedAngle); // inchidere incuietoare dupa ce au trecut cele 10 secunde
    digitalWrite(GREEN_LED, LOW);

    lcd.clear();
    lcd.setCursor(2, 0);
    lcd.print("Door locked");
    delay(2000);
    lcd.clear();
}

```



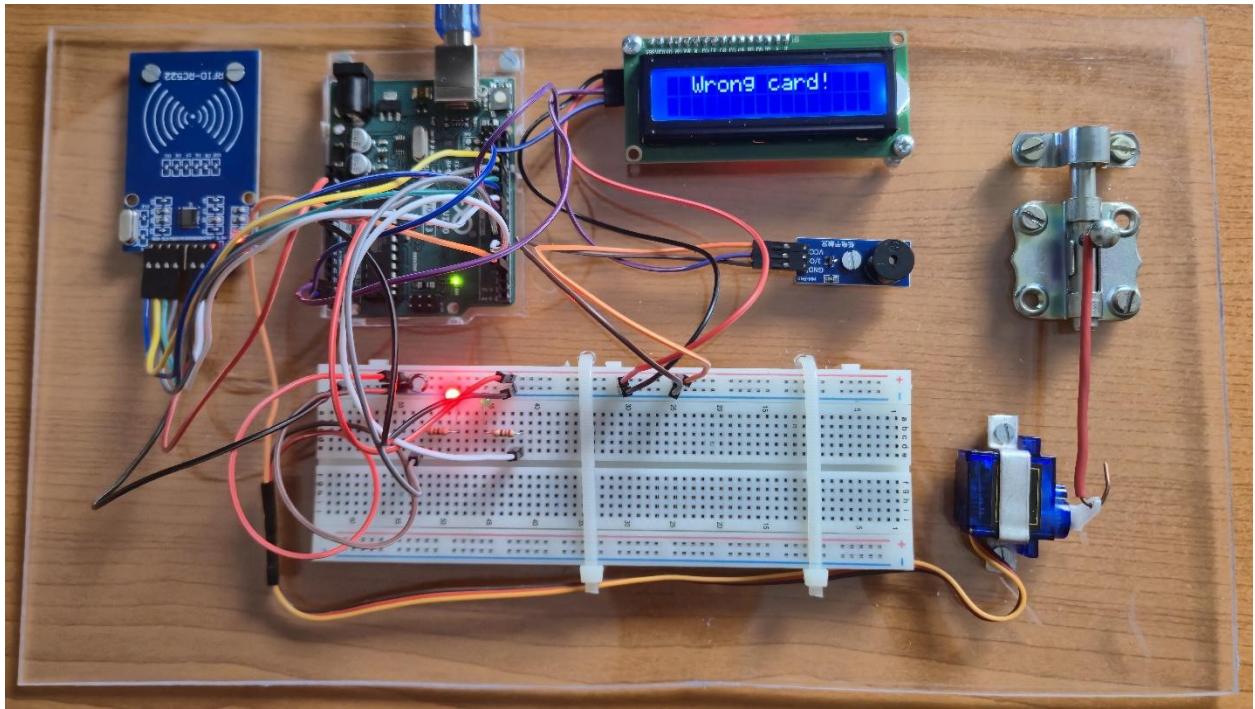
În schimb, dacă identificatorul nu este cel corect, programul declanșează o alertă prin aprinderea LED-ului roșu și generarea a trei semnale sonore rapide. Pe ecran apare avertizarea „*Wrong card!*”, iar servomotorul este menținut forțat în poziția de blocare, garantând securitatea sistemului.

```
else { // card gresit
    digitalWrite(RED_LED, HIGH);

    // alarma: 3 beep-uri rapide
    for (int i = 0; i < 3; i++) {
        digitalWrite(BUZZER, LOW); // alarma sună
        delay(100);
        digitalWrite(BUZZER, HIGH); // alarma se oprește
        delay(100);
    }

    lcd.clear();
    lcd.setCursor(2, 0);
    lcd.print("Wrong card!");
    delay(1000);

    servo.write(lockedAngle);
    delay(1000);
    digitalWrite(RED_LED, LOW);
    lcd.clear();
}
```



## 2.4 Ghid de utilizare

Această secțiune descrie pașii necesari pentru folosirea sistemului și modul în care utilizatorul trebuie să interpreteze semnalele oferite de dispozitiv.

### a) Pornirea și starea inițială (*standby*)

- **Alimentarea:** Sistemul se pornește prin conectarea cablului USB la o sursă de tensiune de 5V.
- **Initializarea:** La pornire, zăvorul se blochează automat (servomotorul se rotește la 70°), iar ecranul LCD afișează mesajul de întâmpinare: "**Welcome! Scan your card**".
- **Monitorizarea:** LED-urile sunt stinse iar buzzer-ul este în starea inactivă, confirmând că sistemul este gata pentru utilizare.

### b) Procedura de scanare

- **Interacțiunea:** Utilizatorul trebuie să apropie tag-ul sau cardul RFID la o distanță de 1-3 cm de antena cititorului MFRC522.
- **Feedback vizual:** În momentul detectării, ecranul LCD își schimbă mesajul în "**Scanning...**", afișând puncte de progres pe măsură ce octetii sunt citiți și procesați.

### c) Accesul permis (Card Autorizat)

Dacă utilizatorul folosește cardul cu UID-ul configurat în sistem, vor avea loc următoarele acțiuni:

- **Semnal vizual:** LED-ul verde se aprinde, iar pe LCD apare mesajul "**Access Granted / Door is open**".
- **Semnal sonor:** Buzzer-ul emite un singur „beep” scurt de confirmare.
- **Acțiune mecanică:** Servomotorul se rotește la 120°, retrăgând zăvorul și permitând deschiderea ușii.
- **Temporizarea:** Zăvorul rămâne în starea deschisă timp de **10 secunde**, după care sistemul revine automat în poziția blocată pentru a asigura securitatea perimetrului.

#### d) Accesul respins (Card Neautorizat)

Dacă se utilizează un card necunoscut, sistemul va reacționa astfel:

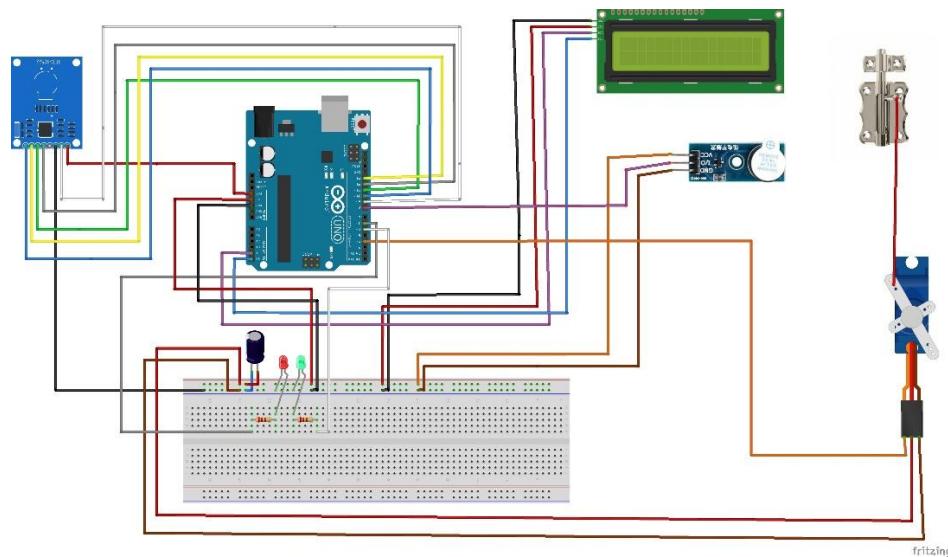
- **Semnal vizual:** LED-ul roșu se aprinde, iar pe LCD apare mesajul de avertizare "**Wrong card!**".
- **Alarma sonoră:** Buzzer-ul emite o succesiune rapidă de **3 semnale sonore**, avertizând asupra unei tentative de acces neautorizate.
- **Blocarea:** Zăvorul rămâne în poziția închis (70°), iar sistemul revine la starea inițială după o scurtă perioadă de timp (2 secunde).

#### e) Observații pentru depanare

- Dacă la scanare LCD-ul nu reacționează, verificați dacă tag-ul/cardul este plasat paralel cu cititorul.
- Monitorizarea în timp real a ID-urilor scanate se poate face prin deschiderea **Serial Monitor**, unde vor fi afișate codurile hexazecimale ale tuturor cardurilor detectate.

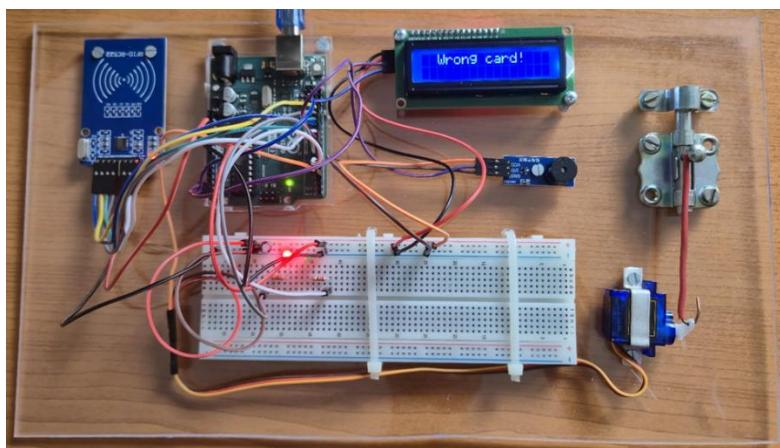
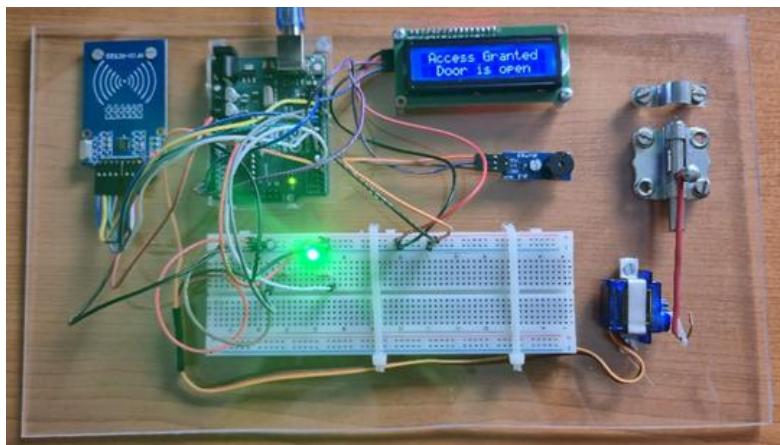
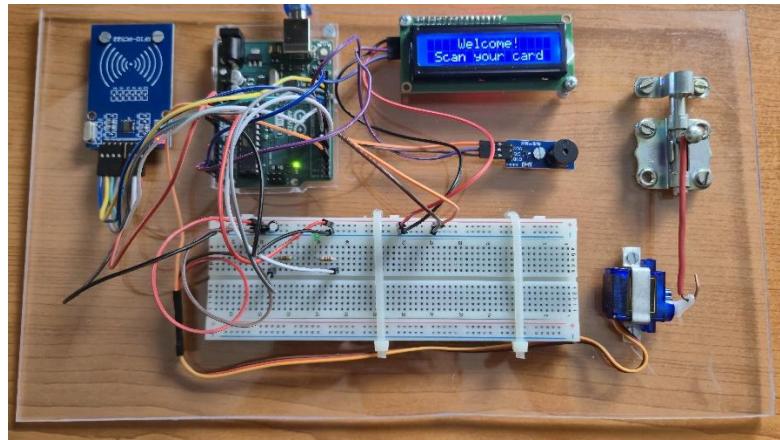
### 3. Schema circuitului

Schema circuitului a fost realizată în mediul de proiectare **Fritzing**, oferind o reprezentare vizuală clară a modului în care componentele sunt conectate cu placă de dezvoltare.



## 4. Poze ale proiectului

În această secțiune sunt prezentate câteva poze care arată funcționarea sistemului.



## 5. Bibliografie

- [1] Arduino UNO R3 docs: <https://docs.arduino.cc/hardware/uno-rev3/>
- [2] Arduino UNO R3 datasheet:  
<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>
- [3] Modul RFID: <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>
- [4] MFRC522 library: <https://docs.arduino.cc/libraries/mfrc522/>
- [5] SPI library: <https://docs.arduino.cc/language-reference/en/functions/communication/SPI/>
- [6] Display LCD cu convertor I2C:  
[https://projecthub.arduino.cc/arduino\\_uno\\_guy/i2c-liquid-crystal-displays-5eb615](https://projecthub.arduino.cc/arduino_uno_guy/i2c-liquid-crystal-displays-5eb615)
- [7] LiquidCrystal I2C library: <https://docs.arduino.cc/libraries/liquidcrystal-i2c/>
- [8] Buzzer: <https://www.ardumotive.com/how-to-use-a-passive-buzzer-module-en.html>
- [9] Servomotor: <https://docs.arduino.cc/learn/electronics/servo-motors/>
- [10] Servo library: <https://docs.arduino.cc/libraries/servo/>