RFID Door Lock Access

Introducere

Ce Face?

Este o descuietoare de usa cu un cititor de RFID tags, care daca a detectat un card care are acces atunci va actiona servo-motor-ul care va deschide zavorul.

Care este scopul lui?

Scopul lui este sa deschida/inchida usa numai oamenilor care au acces.

Care este ideea de la care ati pornit?

Pai la camin eu imi uit mereu cheia si ma gandeam prima data sa fac cu cod pin ca sa nu mai trebuiasca sa imi iau cheia cu mine dar dupa m-am gandit ca aceasta deschizatoare cu un card RFID ar fi mai interesanta si fancy.

De ce credeti ca este util pentru altii si pentru voi?

Este util pentru securitate, adica sa aiba acces sa deschida o camera sau ceva doar anumite persoane care au cardul respectiv.

Descriere generală

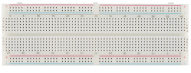
Cum o sa functioneze mecanismul:

1. O sa scanez tag-ul RFID, RFID reader-ul o sa trimita codul citit catre arduino si o sa verifice daca acesta este corect sau nu.
2. Daca codul citit este corect
   1. Se va aprinde un led rosu/verde in functie de raspuns
   2. va afisa un mesaj sugestiv pe ecranul lcd
   3. va actiona servo-motoru care va deschide zavorul in caz afirmativ
3. Dupa un anumit timp zavorul se va inchide

Schema Bloc

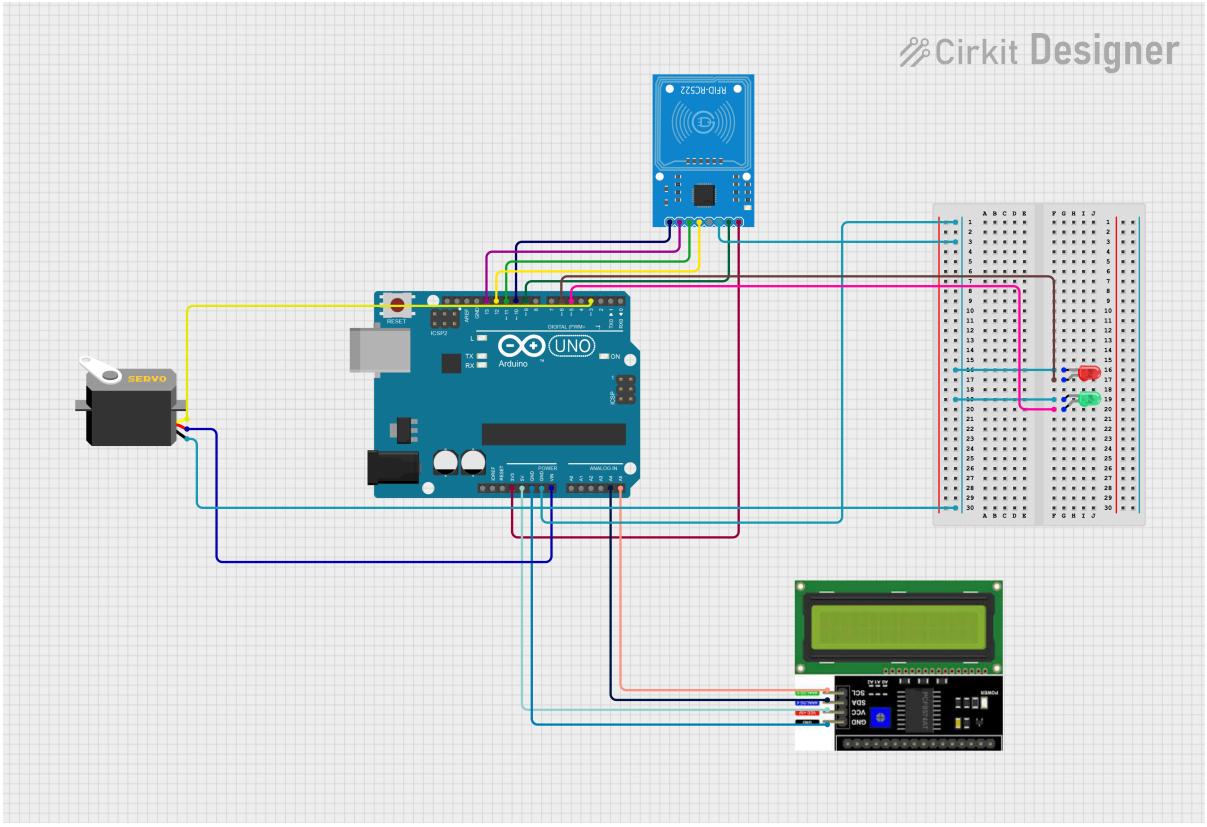
Hardware Design

Lista Componente

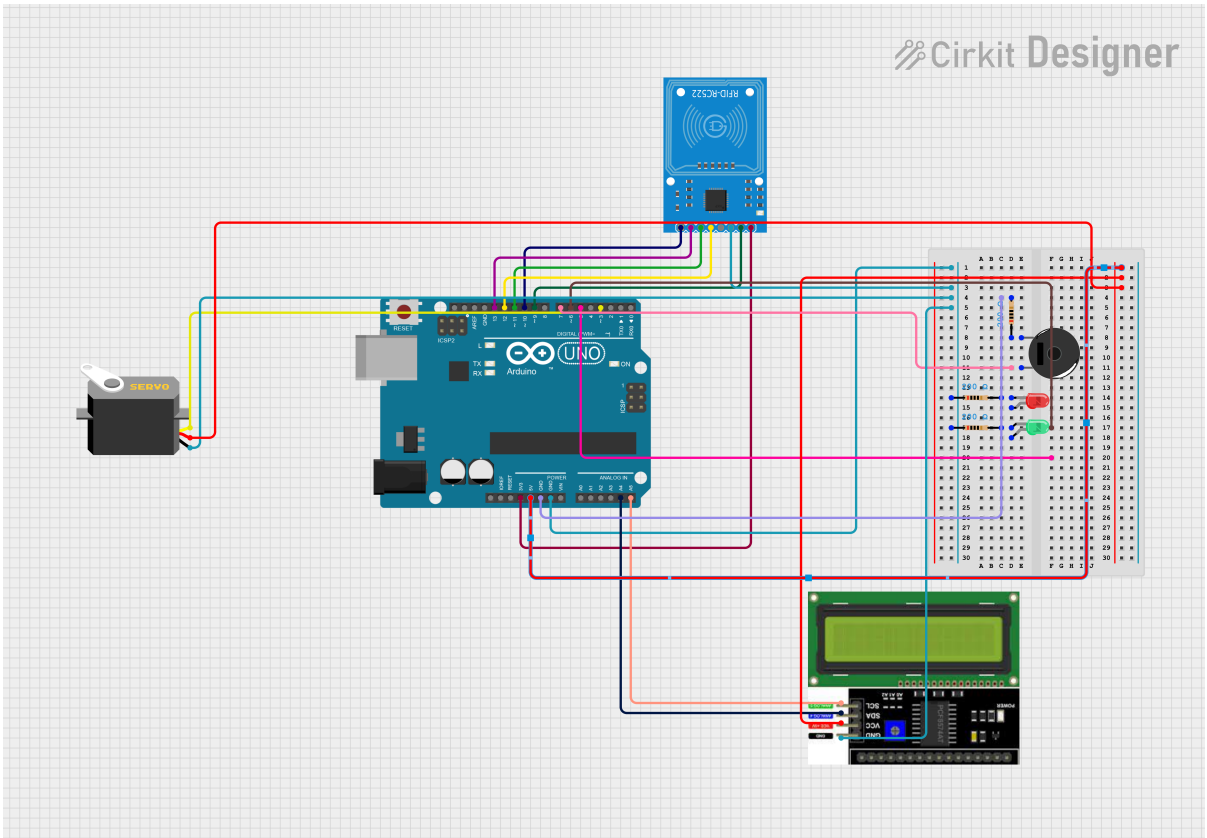
Componenta	Cantitate	Imagine	Descriere
ARDUINO UNO R3 ATmega328P	1		Placuta care controleaza toate actiunile
MOTOR SERVO SG90 9G	1		Pentru a trage de zavor si a deschide usa
MODUL RFID CU CARD SI TAG	1		Modulul RFID este baza de la care pleaca acest proiect pentru ca el detecteaza daca codul RFID este bun sau nu
LED ROSU	1		Se aprinde led-ul rosu daca codul de pe cardul RFID nu este unul care se afla in baza de date
LED VERDE	1		Se aprinde led-ul verde daca codul de pe cardul RFID este unul care se afla in baza de date
ECRAN LCD 1602 IIC/I2C	1		Va afisa un mesaj de succes sau eroare la fel ca ledurile
BREADBOARD 400 PUNCTE	1		Pentru a putea conecta ledurile la arduino
BUZZER ACTIV 3V	1		Cand scanez un RFID TAG gresit atunci va scoate un sunet

Scheme Electrice

Schema Fizica



Noua Schema Fizica



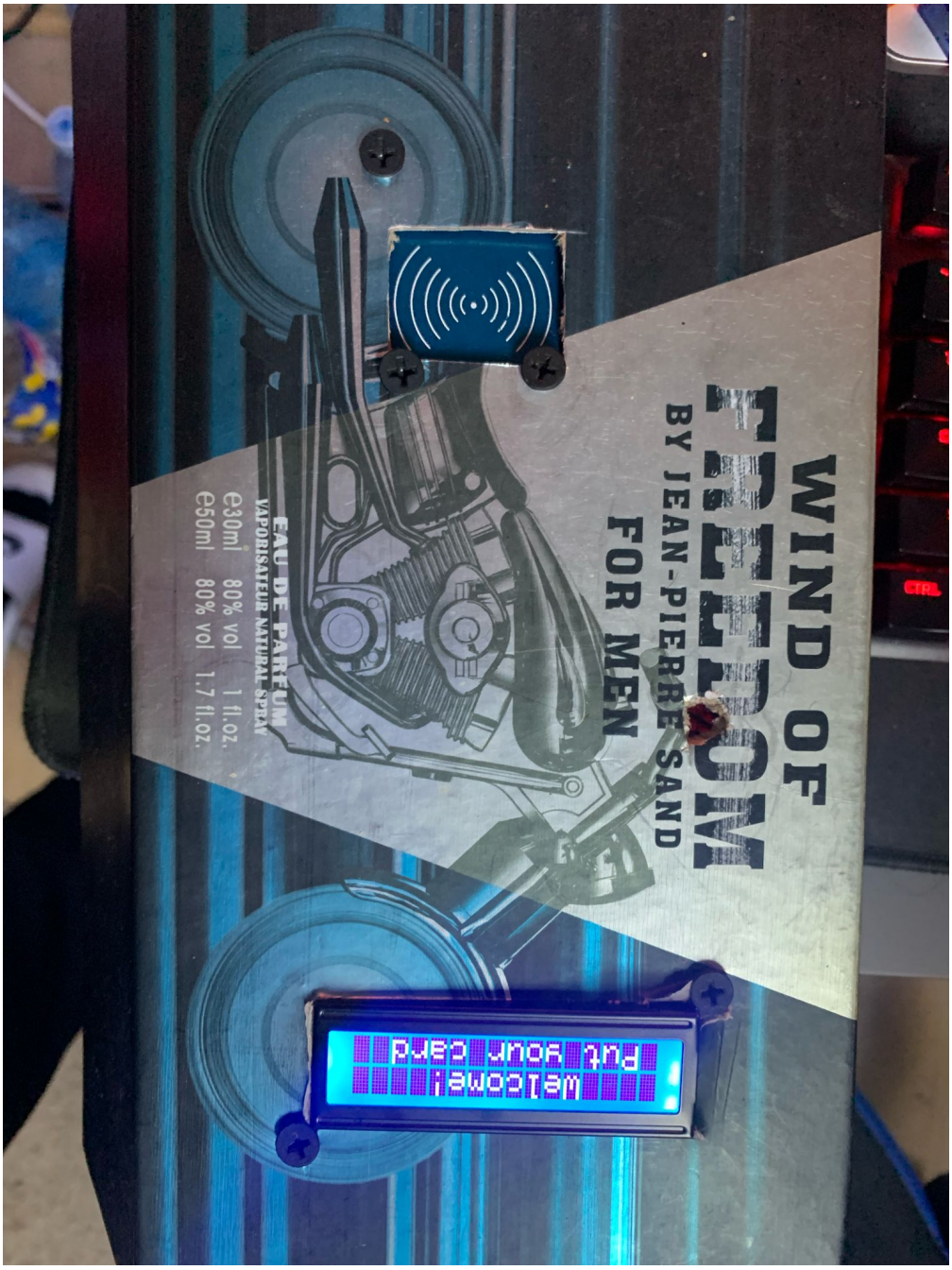
Alegerea Pinilor

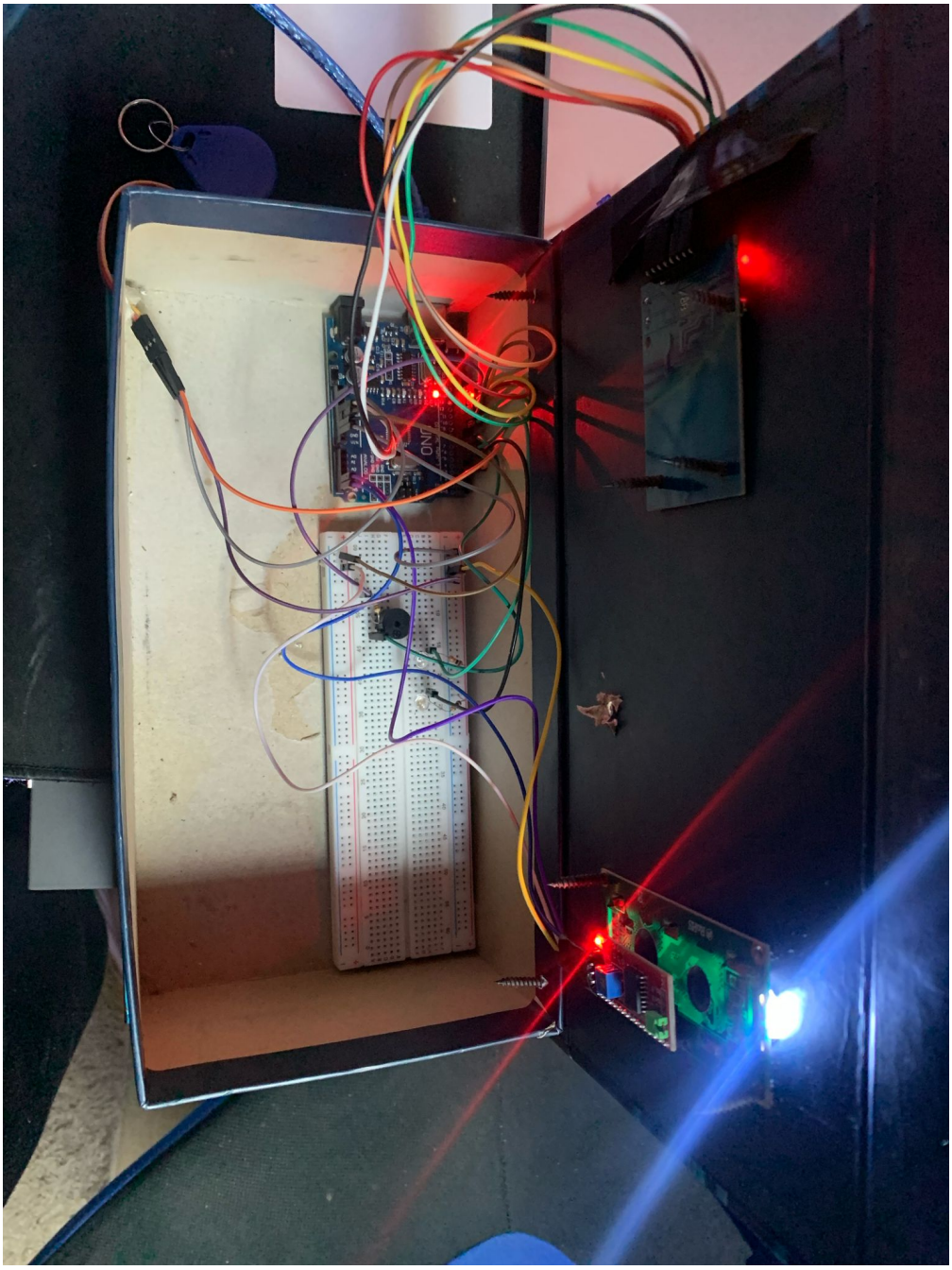
- Pinii 0-1 sunt rezervati pentru comunicare seriala si eu nu folosesc asa ceva in proiect.
- Pinii 5-6 sunt folositi pentru leduri. Pinul 5 este folosit pentru ledul Verde care inidca ca cardul scanat este unul bun si pinul 6 este folosit pentru ledul Rosu care indica ca cardul scanat este unul gresit.
- Pinul 3 este folosit pentru servo-motor pe care l-am si implementat deja in videoclipul de mai jos.
- Pinii 9-13 sunt folositi pentru piesa cea mai importanta din acest proiect, adica modulul RFID.
- Pinul 7 este folosit pentru buzzer care este folosit pentru a atentiona sonor, cardul care a fost scanat e gresit.
- Pinii A4-A5 sunt folositi pentru ecranul LCD.

Piesa Implementata

https://youtube.com/shorts/FqmM0xASqoQ?feature=share

Produsul Final Ambalat


WIND OF
FREEDOM
BY JEAN-PIERRE SAND
FOR MEN
EAU DE PARFUM
VAPORISATEUR NATURAL SPRAY
30ml 80% vol 1 fl.oz.
50ml 80% vol 1.7 fl.oz.
welcome!
put your card



Software Design

- Mediu de dezvoltare ARDUINO IDE

Detectare Card

Am inceput sa conectez doar LCD-ul si RFID la placuta Arduino UNO pentru a detecta ce cod are cardul RFID care va avea acces sa intre in cladire sau nu. Deci toate codurile pe care acesta le va scana le va afisa pe ecranul LCD si daca vreau le stochez in baza de date.

Include-urile si definirea pinilor

```
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>
```

LiquidCrystal_I2C.h pentru a interactiona cu un display LCD folosind protocolul I2C.

SPI.h pentru comunicarea SPI (Serial Peripheral Interface).

MFRC522.h pentru a lucra cu cititorul RFID RC522.

```
#define RST_PIN 9
#define SS_PIN  10
byte readCard[4];
byte a = 0;
```

RST_PIN și SS_PIN sunt pini pentru comunicarea cu cititorul RFID.

readCard este un array de 4 bytes pentru a stoca UID-ul cardului RFID citit.

a este o variabila pentru a controla pozitia cursorului pe LCD.

Initializarea obiectelor

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
MFRC522 mfrc522(SS_PIN, RST_PIN);
```

Obiectul lcd pentru a controla display-ul LCD (adresa I2C 0x27, 16×2 caractere).

Obiectul mfrc522 pentru a controla cititorul RFID.

Functia setup()

```
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  while (!Serial);
  SPI.begin();
  mfrc522.PCD_Init();
  delay(4);
  mfrc522.PCD_DumpVersionToSerial();
  lcd.setCursor(2, 0);
```

```
  lcd.print("Put your card");
}
```

Pornim comunicarea seriala.

Initializam LCD-ul si activam iluminarea de fundal.

Asteptam sa fie disponibil portul serial.

Incepem comunicarea SPI.

Initializam cititorul RFID.

Introducem un mic delay pentru stabilizare.

Afisam versiunea cititorului RFID in Serial Monitor.

Setam cursorul LCD-ului si afisam mesajul “Put your card”.

Functia loop()

```
void loop() {
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return 0;
  }
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return 0;
  }

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Scanned UID");
  a = 0;
  Serial.println(F("Scanned PICC's UID:")); // Proximity Integrated Circuit
Card
  for ( uint8_t i = 0; i < 4; i++) {  //
    readCard[i] = mfrc522.uid.uidByte[i];
    Serial.print(readCard[i], HEX);
    Serial.print(" ");
    lcd.setCursor(a, 1);
    lcd.print(readCard[i], HEX);
    lcd.print(" ");
    delay(500);
    a += 3;
  }
  Serial.println("");
  mfrc522.PICC_HaltA();
  return 1;
```

```
}
```

Daca nu este detectat un card nou, functia loop() se termina si reincepe (returneaza 0).

Daca este detectat un card, dar nu poate fi citit, functia loop() se termina si reincepe (returneaza 0).

Daca un card este citit cu succes:

Curatam LCD-ul.

Setam cursorul si afisam mesajul “Scanned UID”.

Resetam variabila a la 0 pentru pozitionarea cursorului pe LCD.

Afisam UID-ul cardului în Serial Monitor si pe LCD:

Iteram prin fiecare byte din UID-ul cardului citit.

Stocam fiecare byte in array-ul readCard.

Afisam byte-ul in hexazecimal in Serial Monitor si pe LCD.

Afisam fiecare byte la o pozitie diferita pe LCD, cu un spatiu intre ele, si adaugam un delay de 500 ms intre afiasri.

Incrementez variabila a pentru a schimba pozitia cursorului pe LCD.

După afisarea completă a UID-ului, opresc comunicatia cu cardul RFID folosind mfrc522.PICC_HaltA().

Functia loop() se termina returnând 1.

Cod Proiect MAIN

Acest cod permite deschiderea si inchiderea unei usi folosind un card RFID specific. LED-urile si buzzer-ul sunt folosite pentru a semnaliza starea corectă sau incorectă a cardului.

Include-urile și definirea pinilor

```
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>
```

Servo.h pentru a controla un servo motor.

LiquidCrystal_I2C.h pentru a interactiona cu un display LCD folosind protocolul I2C.

SPI.h pentru comunicarea SPI (Serial Peripheral Interface).

MFRC522.h pentru a lucra cu cititorul RFID RC522.

```
#define SS_PIN 10
#define RST_PIN 9
#define CORRECT_LED_PIN 5
#define WRONG_LED_PIN 6
#define BUZZER_PIN 7
```

SS_PIN și RST_PIN sunt pini pentru comunicarea cu cititorul RFID.

CORRECT_LED_PIN este pinul pentru LED-ul verde (corect).

WRONG_LED_PIN este pinul pentru LED-ul rosu (gresit).

BUZZER_PIN este pinul pentru buzzer (avertizare sonora).

```
String UID = "4C 35 8D 64";
byte lock = 0;
```

Aici definim un sir care contine UID-ul cardului RFID autorizat si o variabila lock care indica starea usii (0 pentru deschisa, 1 pentru inchisa).

Initializarea obiectelor

```
Servo servo;
LiquidCrystal_I2C lcd(0x27, 16, 2);
MFRC522 rfid(SS_PIN, RST_PIN);
```

Obiectul servo pentru a controla servo motorul.

Obiectul lcd pentru a controla display-ul LCD (adresă I2C 0x27, 16×2 caractere).

Obiectul rfid pentru a controla cititorul RFID.

Functia setup()

```
void setup() {
  Serial.begin(9600);
  servo.write(0);
  lcd.init();
  lcd.backlight();
  servo.attach(3);
  SPI.begin();
  rfid.PCD_Init();
```

```
  // Set pins 5, 6, and 7 as outputs using DDRD register
  DDRD |= (1 << DDD5) | (1 << DDD6) | (1 << DDD7);

  // Ensure LEDs are off initially using PORTD register
  PORTD &= ~((1 << PORTD5) | (1 << PORTD6) | (1 << PORTD7));
}
```

Pornim comunicarea seriala.

Setam servo la pozitia 0 grade.

Initializam LCD-ul si activam iluminarea de fundal.

Atasam servo la pinul 3.

Incepem comunicarea SPI.

Initializam cititorul RFID.

Setam pinii pentru LED-uri și buzzer ca output si ii inițializam in starea LOW (oprit).

Functia loop()

```
void loop() {
  lcd.setCursor(4, 0);
  lcd.print("Welcome!");
  lcd.setCursor(1, 1);
  lcd.print("Put your card");

  if (!rfid.PICC_IsNewCardPresent())
    return;
  if (!rfid.PICC_ReadCardSerial())
    return;

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Scanning");
  Serial.print("NUID tag is :");
  String ID = "";
  for (byte i = 0; i < rfid.uid.size; i++) {
    lcd.print(".");
    ID.concat(String(rfid.uid.uidByte[i] < 0x10 ? " 0" : " "));
    ID.concat(String(rfid.uid.uidByte[i], HEX));
    delay(300);
  }
  ID.toUpperCase();
```

Daca nu este detectat un card nou, functia loop() se termina si reincepe.

Daca este detectat un card, il citim.

Afisam un mesaj de “Scanning” pe LCD si construim string-ul ID din UID-ul cardului detectat.

Verificarea UID-ului si actionarea usii

```
if (ID.substring(1) == UID && lock == 0) {
    servo.write(0);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Door is locked");

    // Turn on correct LED and turn off wrong LED using PORTD register
    PORTD |= (1 << PORTD5);  // Turn on CORRECT_LED_PIN
    PORTD &= ~(1 << PORTD6); // Turn off WRONG_LED_PIN

    delay(1500);
    lcd.clear();
    lock = 1;
    PORTD &= ~(1 << PORTD5); // Turn off CORRECT_LED_PIN
  } else if (ID.substring(1) == UID && lock == 1) {
    servo.write(180);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Door is open");

    // Turn on correct LED and turn off wrong LED using PORTD register
    PORTD |= (1 << PORTD5);  // Turn on CORRECT_LED_PIN
    PORTD &= ~(1 << PORTD6); // Turn off WRONG_LED_PIN

    delay(1500);
    lcd.clear();
    lock = 0;
    PORTD &= ~(1 << PORTD5); // Turn off CORRECT_LED_PIN
  } else {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Wrong card!");

    // Turn off correct LED and turn on wrong LED using PORTD register
    PORTD &= ~(1 << PORTD5); // Turn off CORRECT_LED_PIN
    PORTD |= (1 << PORTD6);  // Turn on WRONG_LED_PIN

    tone(BUZZER_PIN, 1000);
    delay(1500);
    noTone(BUZZER_PIN);
    lcd.clear();
    PORTD &= ~(1 << PORTD6); // Turn off WRONG_LED_PIN
```

```
  }
}
```

Daca UID-ul cardului citit corespunde cu UID si usa este deschisa (lock == 0), inchidem usa (servo la 0 grade), aprindem LED-ul verde, asteptam 1.5 secunde, actualizam lock la 1 (usa inchisa) si stingem LED-ul verde.

Daca UID-ul cardului corespunde cu UID si usa este inchisa (lock == 1), deschidem usa (servo la 180 grade), aprindem LED-ul verde, asteptam 1.5 secunde, actualizam lock la 0 (usa deschisa) si stingem LED-ul verde.

Daca UID-ul cardului nu corespunde, afisam un mesaj de eroare pe LCD, aprindem LED-ul rosu si activam buzzer-ul pentru 1.5 secunde, dupa care stingem LED-ul rosu si buzzer-ul.

Rezultate Obţinute

https://youtube.com/shorts/UCl-P6DH4_A?feature=share

Concluzii

Mi-a placut tare mult sa lucrez la acest proiect singurul regret pe care il am este ca am un motor cam slabut si nu poate sa traga de zavor, puteam sa iau o broasca automata in loc sa fac cu servo si cu zavor si cel mai mare regret este ca nu am facut un proiect mai complex, de la ideea initiala am mai incercat sa aduc modificari dar nu foarte complexe. In rest a fost super fain 😎

Download

GITHUB REPO: https://github.com/DanAlin19/PM

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** şi **Resurse Hardware**.

Export to PDF