

# **Proiect MT - Sistem de acces și gestionare a parcării cu Arduino:** **Integrare RFID și barieră automată .**

Nume : Andronache Robert Stefan

## **->Descriere Tehnica.**

Proiectul reprezinta un **sistem de parcare automatizat**, dezvoltat folosind platforma Arduino Uno R3, ce integreaza mai multe componente hardware Si module electronice pentru a oferi o soluție eficientă de gestionare a accesului vehiculelor într-un spatiu de parcare. Sistemul utilizeaza un **modul RFID** pentru autentificare, **senzori IR** pentru detectarea vehiculelor, Și un **servo motor** pentru controlul barierei automate.

Interactiunea cu utilizatorul este asigurata printr-un **display LCD**, care afiseaza mesaje privind starea sistemului, cum ar fi „Acces Permis” sau „Acces Refuzat”. Indicatoarele LED ofera feedback vizual pentru accesul aprobat sau refuzat. Alimentarea sistemului este asigurata printr-un adaptor de 5V, iar conexiunile sunt realizate pe o placa de dezvoltare (breadboard) utilizând fire jumper.

---

## **->Functionalitati Tehnice Principale.**

1. **Autentificare RFID:** Identificarea vehiculelor utilizând un modul RFID care citește carduri/tag-uri de proximitate.
  2. **Controlul barierei:** Servo motorul ridică Și coboară bratul barierei în funcție de accesul permis.
  3. **Detectarea vehiculelor:** Senzorii IR detectează prezența vehiculelor pentru a activa bariera.
  4. **Afisare status:** Mesajele sistemului sunt afișate pe un LCD 16x2 pentru a informa utilizatorii despre starea accesului.
  5. **Indicatie LED:** LED-urile roșu Și verde indică starea accesului (refuzat sau permis).
-

->Descriere Practica.

Acest sistem poate fi implementat în diverse scenarii reale, cum ar fi parcarile rezidentiale, spatiile de birouri sau parcarile comerciale, oferind un mod sigur, eficient Și automatizat de gestionare a accesului vehiculelor. Sistemul elimina nevoia interventiei manuale Și reduce erorile umane, contribuind astfel la un flux de trafic optimizat.

-> Tabelul Componentelor Electronice și Specificațiilor Tehnice.

Hardware/Module Name	Description	Specifications	Usage in Project
1. Arduino Uno R3	-> Microcontroller board based on the ATmega328P.	-> 14 digital I/O pins, 6 analog inputs, 16 MHz quartz crystal.	-> Core component for controlling sensors, RFID reader, and servo.
2. RFID Module	-> Reads RFID tags and sends data to the Arduino	-> Frequency: 13.56 MHz, Range: up to 5 cm	-> Used for scanning RFID tags for access control.
3. IR Sensors	-> Detects the presence of objects or vehicles.	-> Operating Voltage: 3-5V, Range: 2-30 cm	-> Detects vehicle entry and exit.
4. Servo Motor	-> Controls the parking barrier arm.	-> Operating Voltage: 4.8-6V, Torque: 1.8 kg-cm	-> Used for raising and lowering the barrier.
5. LCD Display	-> Displays system status messages.	-> 16x2 characters	-> Provides feedback to the user, e.g., 'Access Granted' or 'Denied.'
6. LEDs	-> Indicates system states.	-> Red, Green LEDs	-> Red for denied access, Green for granted access.
7. Breadboard	-> Prototyping board for wiring connections	-> Dimensions: 830 tie-points	-> Holds and organizes the circuit connections.
8. Jumper Wires	-> Connect components on the breadboard	-> Various lengths and male-to-male or female-to-male	-> Connects all the components electrically.
9. Power Supply	-> Provides power to the system	-> 5V DC Adapter	-> Supplies consistent power to the Arduino and other components.

->Cod Sursă: Sistem de Parcare Automatizat cu Arduino.

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

#define SS_PIN 10 // Pinul SS pentru modulul RFID
#define RST_PIN 7 // Pinul de reset pentru modulul RFID
#define servoPin 9 // Pinul pentru servomotor
#define IR 3 // Pinul pentru senzorul IR de intrare
#define GREEN 2 // Pinul pentru LED-ul verde
#define RED 6 // Pinul pentru LED-ul roșu
#define YELLOW 5 // Pinul pentru LED-ul galben
#define IR1 8 // Pinul pentru senzorul IR de ieșire
#define buzzer A3 // Pinul pentru buzzer
bool exitRequested = false;
int flag1 = 0;
int flag2 = 0;
int unghi = 0;
Servo servo;
MFRC522 mfrc522(SS_PIN, RST_PIN); // Instance of MFRC522
LiquidCrystal_I2C lcd(0x27, 16, 2);
MFRC522::MIFARE_Key key;

void setup() {
    checkExitSensor();
    Serial.begin(9600);
    SPI.begin(); // Initialize SPI
    mfrc522.PCD_Init(); // Initialize RC522
    lcd.init();
    servo.attach(servoPin);
    pinMode(IR, INPUT);
    pinMode(IR1, INPUT);
    pinMode(GREEN, OUTPUT);
    pinMode(YELLOW, OUTPUT);
    pinMode(buzzer, OUTPUT);
    pinMode(RED, OUTPUT);
}

void loop() {
    checkExitSensor() ;
```

```

lcd.setCursor(0, 0);
lcd.print("Put card on rdr ");
lcd.backlight();

// Turn yellow LED on to indicate waiting for a card
digitalWrite(YELLOW, HIGH);

// Look for new cards
if (!mfrc522.PICC_IsNewCardPresent()) {
    return;
}

// Select one of the cards
if (!mfrc522.PICC_ReadCardSerial()) {
    return;
}

// Display UID on serial monitor
Serial.print("UID tag :");
String content = "";
for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
Serial.println();

// Verify UID for authorized access
content.toUpperCase();
if (content.substring(1) == "83 D2 7F 35") { // Change to your authorized
UID
    lcd.clear();
    Serial.println("Authorized access");
    tone(buzzer, 4000);
    lcd.setCursor(0, 0);
    lcd.print("Access Granted");

    // Turn yellow LED off, green LED on for access granted
    digitalWrite(YELLOW, LOW);
    delay(100); // Short delay to ensure LED state change
    digitalWrite(GREEN, HIGH);

    // Move servo to open position
    for (unghi = 90; unghi >= 0; unghi--) {

```

```

        servo.write(unghi);
        delay(5);
    }

    // Wait until IR sensor detects object passing through
    while(digitalRead(IR) != 0) {
        // Waiting
    }

    delay(5000); // Additional delay for security
    noTone(buzzer);
    // Close the servo
    for (unghi = 0; unghi <= 90; unghi++) {
        servo.write(unghi);
        delay(5);
    }

    // Reset LED states
    digitalWrite(GREEN, LOW);
    digitalWrite(YELLOW, HIGH);
} else {
    lcd.clear();
    Serial.println("Access denied");

    // Turn yellow LED off, red LED on for access denied
    digitalWrite(YELLOW, LOW);
    delay(100); // Short delay to ensure LED state change
    digitalWrite(RED, HIGH);

    lcd.setCursor(0, 0);
    lcd.print("Access denied.");
    delay(3000);

    // Reset LED states
    digitalWrite(RED, LOW);
    digitalWrite(YELLOW, HIGH);
}
lcd.clear();
}

void checkExitSensor() {
    if (digitalRead(IR1) == LOW && !exitRequested) {
        exitRequested = true; // Set flag to avoid repeated triggering
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Exit Open");
        tone(buzzer, 4000);
    }
}

```

```

digitalWrite(YELLOW, LOW);
delay(100); // Short delay to ensure LED state change
digitalWrite(GREEN, HIGH);

for (unghi = 90; unghi >= 0; unghi--) {
    servo.write(unghi);
    delay(5);
}

while(digitalRead(IR) != 0) { /* Waiting */ }

delay(3000);
noTone(buzzer);

for (unghi = 0; unghi <= 90; unghi++) {
    servo.write(unghi);
    delay(5);
}

    digitalWrite(GREEN, LOW);
    delay(100); // Short delay to ensure LED state change
    digitalWrite(YELLOW, HIGH);
    lcd.clear();
    exitRequested = false; // Reset flag for next exit detection
}
}

```

-> Link către videoclipul demonstrativ al proiectului pe YouTube:

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