



# **UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA**

---

## **Sistem Intelligent de Control al Accesului in Parcare cu Bariera Automata**

*Proiectare cu microprocesoare*

---

Autor: Ilies Bianca Izabela

Grupa: 30235

FACULTATEA DE AUTOMATICA  
SI CALCULATOARE

08 Ianuarie 2026

## Cuprins

<b>1 Rezumat . . . . .</b>	<b>2</b>
<b>2 Introducere . . . . .</b>	<b>2</b>
<b>3 Proiectare si implementare . . . . .</b>	<b>3</b>
3.1 Metoda experimentală utilizată . . . . .	3
3.2 Alternative de proiectare analizate . . . . .	3
3.3 Arhitectura generală a sistemului . . . . .	4
3.4 Descriere componente . . . . .	5
3.4.1 Modul RFID RC522 . . . . .	5
3.4.2 Senzor Ultrasonic HC-SR04 . . . . .	7
3.4.3 Servomotor SG90 . . . . .	8
3.4.4 Display LCD 16×2 cu adaptor I2C . . . . .	10
3.4.5 Modul Bluetooth HC-05 . . . . .	11
3.4.6 Alimentare, interconectare si detalii de cablare globale . . . . .	12
3.5 Concluzii hardware . . . . .	12
3.6 Structura software, functii si interacțiunea cu hardware . . . . .	13
<b>4 Testarea sistemului de parcare intelligent . . . . .</b>	<b>13</b>
4.1 Concluzii finale de testare . . . . .	16

## 1 Rezumat

Sistemul intelligent de parcare reprezinta un pas necesar spre automatizarea controlului accesului in spatiile cu locuri limitate, eliminand erorile umane si timpul pierdut la barierele clasice. Proiectul a avut ca scop rezolvarea problemei gestionarii accesului intr-o parcare de mici dimensiuni prin implementarea unei bariere automate controlate de un microcontroller Arduino, avand ca obiective principale autentificarea utilizatorilor, detectarea vehiculelor la iesire, afisarea in timp real a locurilor disponibile si controlul barierei printr-un canal wireless.

Solutia propusa a utilizat un sistem embedded bazat pe Arduino impreuna cu un modul RFID pentru validarea accesului, un senzor ultrasonic pentru detectarea masinilor la iesire, un servomotor pentru actionarea barierei, un display LCD comunicand prin I2C pentru interactiunea vizuala cu utilizatorul si un modul Bluetooth pentru control extern al barierei. Metoda de rezolvare a fost implementata in C/C++ folosind Arduino IDE, impreuna cu biblioteci dedicate pentru comunicare cu module hardware, transmisie seriala software si protocoale precum SPI si I2C. Sistemul a fost testat in conditii reale, demonstrand functionarea corecta a barierei, actualizarea contorului de locuri libere la fiecare intrare sau iesire si raspunsul atat la citirea ID-urilor valide cat si la detectarea vehiculelor aflate in apropierea senzorului. Rezultatele au confirmat ca bariera s-a deschis doar la autentificari valide sau comenzi bluetooth permise, iar starea locurilor libere a fost actualizata pe afisorul LCD in timp real.

In concluzie, proiectul a demonstrat ca autentificarea combinata cu detectarea fizica a oferit o solutie eficienta, rapida si usor utilizabila pentru controlul accesului in parcare inteligente, atingand toate cerintele funktionale propuse.

## 2 Introducere

Automatizarea controlului in parcare a devenit o directie importanta in sistemele embedded moderne, pe fondul cresterii traficului in orase si al necesitatii gestionarii eficiente a spatiilor cu locuri limitate de parcare. Solutiile de tip "Smart parking" au evoluat rapid, integrand tehnologii precum:

- radiofrecventa (RFID)
- senzori de distanta pentru detectarea vehiculelor
- comunicatii wireless (Bluetooth)
- servomotoare pentru mecanisme fizice automate

In momentul acesta se pune accentul pe sisteme rapide, cu efecte vizuale in timp real si interactiuni wireless cat mai rapide, reducand timpii de asteptare la bariere clasice si eliminand erorile cauzate de gestionarea umana.

Proiectul s-a incardat in domeniul sistemelor embedded pentru control fizic automat, un sub-domeniu al telecomunicatiilor aplicante si al proiectarii cu microprocesoare in care microcontrollerul interactioneaza direct cu modulele hardware prin protocoale standard de comunicare (SPI, I2C, UART) si cu periferice dedicate (senzori, afisoare LCD).

Terminologia de baza a inclus:

- RFID, adica tehnologia de identificare prin radiofrecventa
- Senzor Ultrasonic HC-SR04, senzor de detectare a distantei pana la un obiect
- Servo Motor SG90 controlat PWM care permite pozitionarea barierei atat in modul deschis, cat si inchis
- LCD cu modul I2C, un afisor caruia i-am incorporat un modul I2C pentru a utiliza un numar redus de pini la legarea in circuit

- Modul Bluetooth HC-05 (transciever serial) ce ofera interfata de comunicare wireless configurata ca port serial software

Importanta domeniului a fost data de faptul ca astfel de sisteme pot fi implementate rapid pe hardware cu un pret redus, sunt modulare, extensibile si pot interactiona cu utilizatorul usor.

Problema de rezolvat a fost definita ca gestionarea automata a accesului intr-o parcare cu locuri limitate, cu validare sigura la intrare si detectarea vehiculelor la iesire, actualizand in timp real numarul de locuri disponibile pe afisor si actionand bariera doar in conditiile permise.

Obiectivele principale ale proiectului au fost:

1. Autentificarea utilizatorului la intrare folosind in RFID si compararea UID-ului cu cele valide
2. Detectarea prezentei vehiculelor in apropierea senzorului de distanta pentru parasirea parcarii fara autentificare suplimentara
3. Afisarea in timp real al numarului de locuri disponibile in parcare si recalcularea lui dupa intrarea sau iesirea unui autovehicul
4. Actionarea barierei atasate de servomotorul controlat PWM, oferind o miscare exacta in momentul deschiderii si inchiderii acestieia
5. Gestionarea corecta a controlului de locuri, incrementarea si decrementarea la fiecare iesire, respectiv intrare, fara depasirea limitelor impuse sistemului

Aceasta abordare fara implicare umana a fost superioara barierei clasice, deoarece a oferit accesul selectiv fara timpul de asteptare al unei bariere normale

Diferentele cheie fata de alte solutii au fost:

1. Decizia de acces selectiv fara server intermediar
2. Detectarea automata la iesire, fara validare suplimentara
3. Afisare imediata pe LCD
4. Control Bluetooth, fara modulul UART hardware dedicat

Raportul proiectului a fost structurat in ...DE CONTINUAT

### **3 Proiectare si implementare**

#### **3.1 Metoda experimentală utilizată**

Proiectul a fost realizat printr-o implementare experimentală hardware reală, controlată local de un microcontroller Arduino UNO, împreună cu un program embedded scris în C/C++ care gestionează accesul și iesirea vehiculelor într-o parcare cu capacitate limitată.

#### **3.2 Alternative de proiectare analizate**

În etapa de proiectare au fost analizate mai multe variante de implementare pentru fiecare modul al sistemului.

Pentru controlul accesului la intrare, s-a luat în considerare utilizarea unui buton fizic conectat la un pin digital, însă această soluție a fost respinsă, deoarece necesita interacțiune directă cu utilizatorul și elimină funcționalitatea contactless a sistemului. O altă variantă a fost folosirea unui senzor IR de proximitate pentru detectarea mașinii la intrare, fară autentificare, dar a fost abandonată ideea, pentru că se pierdea caracterul de securitate al proiectului. De asemenea s-a luat în calcul și implementarea unui keypad 4x4 pentru introducerea unui cod de acces, dar ocupa un număr prea mare de pini, mai mulți decât modulul RFID și era mai lent în utilizare.

Astfel am reusit sa pastrez atat caracterul de securitate, contactless, cat si timpul scurt la intrare si autentificare.

Pentru detectarea vehiculului la iesire, alternativa principală a fost utilizarea unui senzor IR, dar a fost respinsa, deoarece putea fi afectata de lumina ambientala si necesita calibrare mai sensibila, spre deosebire de senzorul ultrasonic care functioneaza independent de conditiile de iluminare. O alta varianta care mai era disponibila insa nu s-ar putea implementa ar fi folosirea unui senzor Hall cu magnet pe masina. Aceasta idee a fost abandonata, deoarece necesita modificarea masinii, iar acest aspect nu era unul dorit sau potrivit atat pentru proiectul meu, cat si pentru implementarea reala.

La actionarea barierei, s-a analizat utilizarea unui motor DC cu puncte H pentru deschidere si inchidere, dar s-a renuntat, fiind ca nu era la fel de precis si necesita encodere.

Pentru afisarea disponibilitatii parcurii, s-a luat in considerare utilizarea unui display SPI TFT, insa acesta nu era potrivit pentru ca necesita o grafica mai complexa si consuma mult mai multe resurse, in timp ce LCD-ul I2C era suficient pentru interactiunea ceruta.

In final, pentru comunicarea Bluetooth, s-a analizat portul UART hardware, insa acesta intra in conflict cu interfata seriala folosita pentru debbuging, motiv pentru care s-a ales varianta SoftwareSerial pe pinii D2/D3.

Solutia finala a fost selectata pe baza criteriilor:

- numarul redus de pini utilizati
- precizia in detectare
- securitate locala
- usurinta in testare si debugging
- lipsa dependentelor externe

### 3.3 Arhitectura generală a sistemului

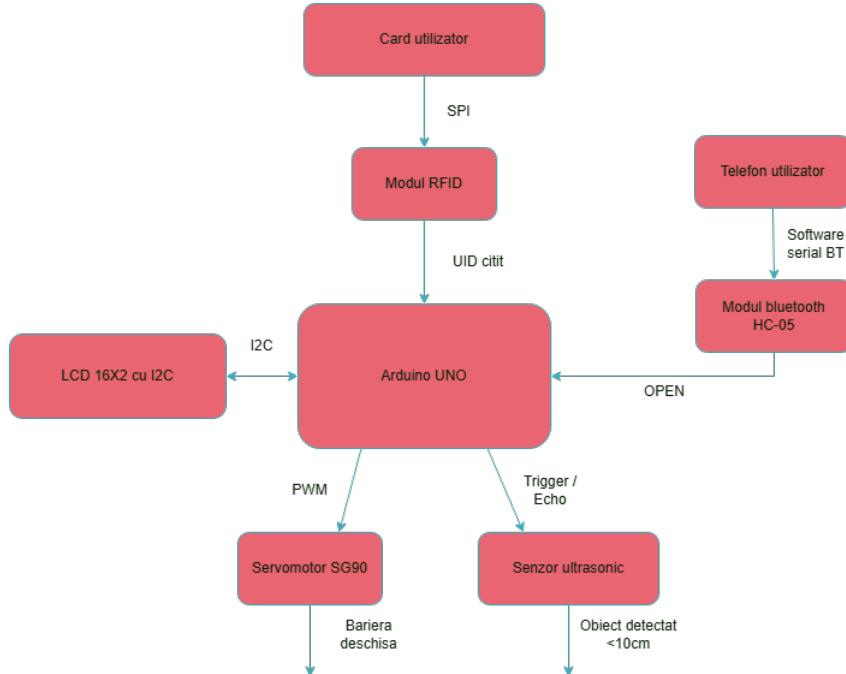


Figura 1: Schema bloc

Arhitectura generala a sistemului a fost proiectata modular avand microcontroller-ul Arduino UNO ca unitate centrala de procesare si decizie. Sistemul a integrat doua module de intrare: citirea cardului utilizatorului prin modului RFID RC522 cu comunicatie SPI si detectarea prezentei vehiculului la iesire prin senzorul ultrasonic HC-SR04.

Iesirile sistemului au constatat in actionarea mecanica a barierei prin servomotorul SG90, contolat PWM si afisarea statusului parcurii pe display-ul LCD 16x2 cu adaptor I2C, comunicand cu placa prin protocolul I2C, master-slave.

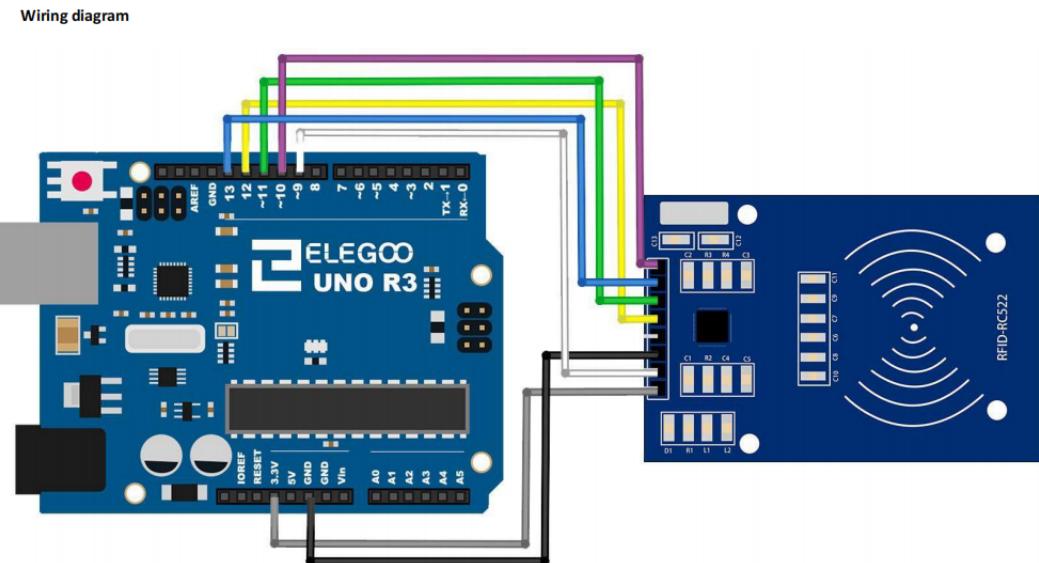
In plus, sistemul a inclus un canal de control wireless realizat prin modulul Bluetooth HC-05, utilizand interfata SoftwareSerial pe pinii digitali, permitand receptia comenziilor text. De exemplu, "OPEN" pentru deschiderea barierei de la telefon. Fluxul de control a fost gestionat de software-ul embedded, care a validat UID-urile la intrare, a comandat pozitia servo-ului pentru deschidere si inchidere, a actualizat contorul de locuri ramase si a preventit redeschiderile multiple ale barierei.

*Figura 1* a ilustrat aceasta arhitectura aratand legatura logica dintre module, sensul comunicatiilor si succesiunea deciziilor. Aceasta arhitectura a functionat singura, fara server intermediar si fara operator uman, demonstrand un sistem embedded cu resurse optimizate si interacțiune in timp real.

### 3.4 Descriere componente

#### 3.4.1 Modul RFID RC522

Modulul RFID RC522 este utilizat pentru autentificarea utilizatorilor la intrarea in parcare, prin citirea UID-ului cardurilor contactless in banda de 13.56 MHz. Acesta functioneaza ca periferic SPI, la tensiune de 3.3V, fiind compatibil direct cu placuta Arduino UNO. In cadrul proiectului, RC522 detecteaza si transmite UID-ul cardului catre microcontroller, care il compara local cu UID-ul autorizat pentru a decide deschiderea barierei. Modulul include antena integrata, regulator intern de tensiune si suport pentru carduri.



149 / 223

Figura 2: Conexiunile modulului RFID RC522 cu Arduino UNO

Modulul a fost conectat astfel:

- SDA/SS → D10
- RST → D9
- SCK → D13
- MOSI → D11
- MISO → D12
- VCC → 3.3V
- GND → GND

Fragmentul de cod embedded utilizat pentru initializarea și citirea cardului este:

```
1 #include <SPI.h>
2 #include <MFRC522.h>
3 #define SS_PIN 10
4 #define RST_PIN 9
5 MFRC522 rfid(SS_PIN, RST_PIN);
6
7 byte validUID[4] = {0x9D, 0xF1, 0x2C, 0x1F};
8
9 void setup() {
10     Serial.begin(9600);
11
12     SPI.begin();
13     rfid.PCD_Init();
14
15 }
16
17 void loop() {
18     if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
19
20         bool valid = true;
21         for (byte i = 0; i < 4; i++) {
22             if (rfid.uid.uidByte[i] != validUID[i]) {
23                 valid = false;
24             }
25         }
26
27         if (valid && locuriRamase > 0) {
28             lcd.setCursor(0, 1);
29             lcd.print("Acces permis");
30             deschideBariera();
31             locuriRamase--;
32         } else {
33             lcd.setCursor(0, 1);
34             lcd.print("Acces respins");
35             delay(1500);
36         }
37
38         rfid.PICC_HaltA();
39         rfid.PCD_StopCrypto1();
40
41     }
42 }
```

### 3.4.2 Senzor Ultrasonic HC-SR04

Senzorul ultrasonic HC-SR04 este utilizat in cadrul sistemului pentru detectarea vehiculelor la iesirea din parcare, fara a necesita autentificare suplimentara. Modulul utilizeaza unde acustice la 40 kHz pentru a determina distanta pana la un obiect, prin masurarea timpului dintre emisia si receptia echoului reflectat.

HC-SR04 opereaza la tensiunea de 5 V, este independent de conditiile de iluminare (spre deosebire de senzorii IR) si comunica cu Arduino printr-o interfata digitala bazata pe semnale de tip Trigger/Echo.

In cadrul proiectului, acesta are rolul de a semnaliza prezenta unui autovehicul in apropierea barierei de iesire. Daca distanta masurata este sub un prag definit (10 cm in implementarea curenta), microcontroller-ul comanda deschiderea barierei prin servomotor si actualizeaza local contorul de locuri disponibile, prevenind declansarile multiple prin utilizarea unui flag de detectie.

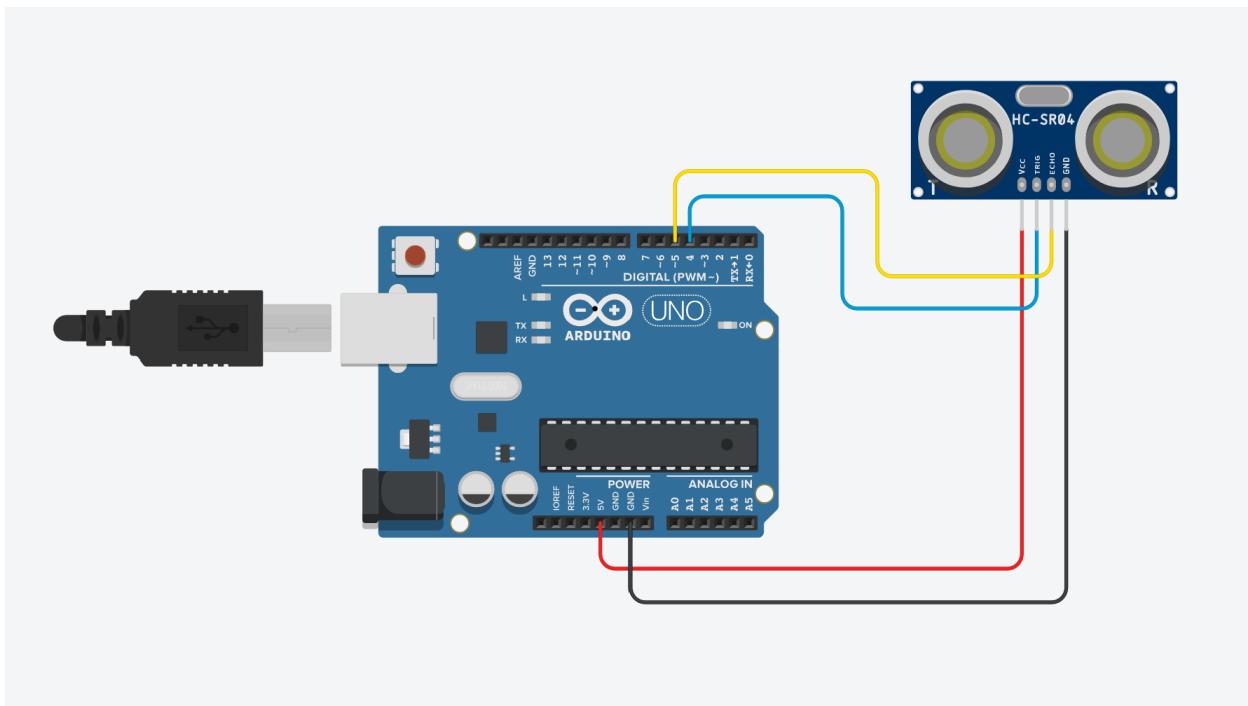


Figura 3: Conexiunile senzorului ultrasonic cu Arduino UNO

Senzorul a fost conectat astfel:

- VCC → 5V
- GND → GND
- TRIG → D4
- ECHO → D5

Fragmentul de cod embedded utilizat pentru generarea semnalului ultrasonic si calculul distantei este:

```
1 #include <SPI.h>
2 #include <MFRC522.h>
3
4 #define SS_PIN 10
5 #define RST_PIN 9
```

```

6  MFRC522 rfid(SS_PIN, RST_PIN);
7
8  const int trigPin = 4;
9  const int echoPin = 5;
10
11 long duration;
12 int distance;
13
14 void setup() {
15     Serial.begin(9600);
16
17     pinMode(trigPin, OUTPUT);
18     pinMode(echoPin, INPUT);
19
20 }
21
22 void loop() {
23     digitalWrite(trigPin, LOW);
24     delayMicroseconds(2);
25     digitalWrite(trigPin, HIGH);
26     delayMicroseconds(10);
27     digitalWrite(trigPin, LOW);
28
29     tmeduration = pulseIn(echoPin, HIGH, 30000);
30     distance = (0.034 * tmeduration) / 2;
31
32     if (distance > 0 && distance <= 10 && !masinaIesireDetectata) {
33         lcd.setCursor(0, 1);
34         lcd.print("Iesire...");
35         deschideBariera();
36
37         if (locuriRamase < totalLocuri) {
38             locuriRamase++;
39         }
40
41         masinaIesireDetectata = true;
42     }
43
44     if (distance > 20) {
45         masinaIesireDetectata = false;
46     }
47
48     delay(300);
49 }
50

```

### 3.4.3 Servomotor SG90

Servomotorul SG90 a avut rolul de a controla bariera, oferind o miscare precisa intre pozitiile inchis (0 grade) si deschis (90 grade), controlata prin semnal PWM generat de microcontroller-ul Arduino UNO.

Acest servomotor opereaza la tensiune de 5 V, consum redus, cu un unghi maxim de rotatie de 180 grade, insa in proiect a fost limitat la 90 grade pentru a actiona corect mecanismul barierei. Spre deosebire de un motor DC clasic, acesta nu a necesitat encodere, deoarece pozitionarea interna a fost gestionata direct de electronica servo-ului.

In implementarea sistemului, SG90 a fost comandat doar dupa validarea RFID sau detectarea ultrasonică la ieșire, iar un delay controlat a mentinut bariera deschisa suficient timp pentru trecerea masinii, inainte de revenirea automata in pozitia initiala.

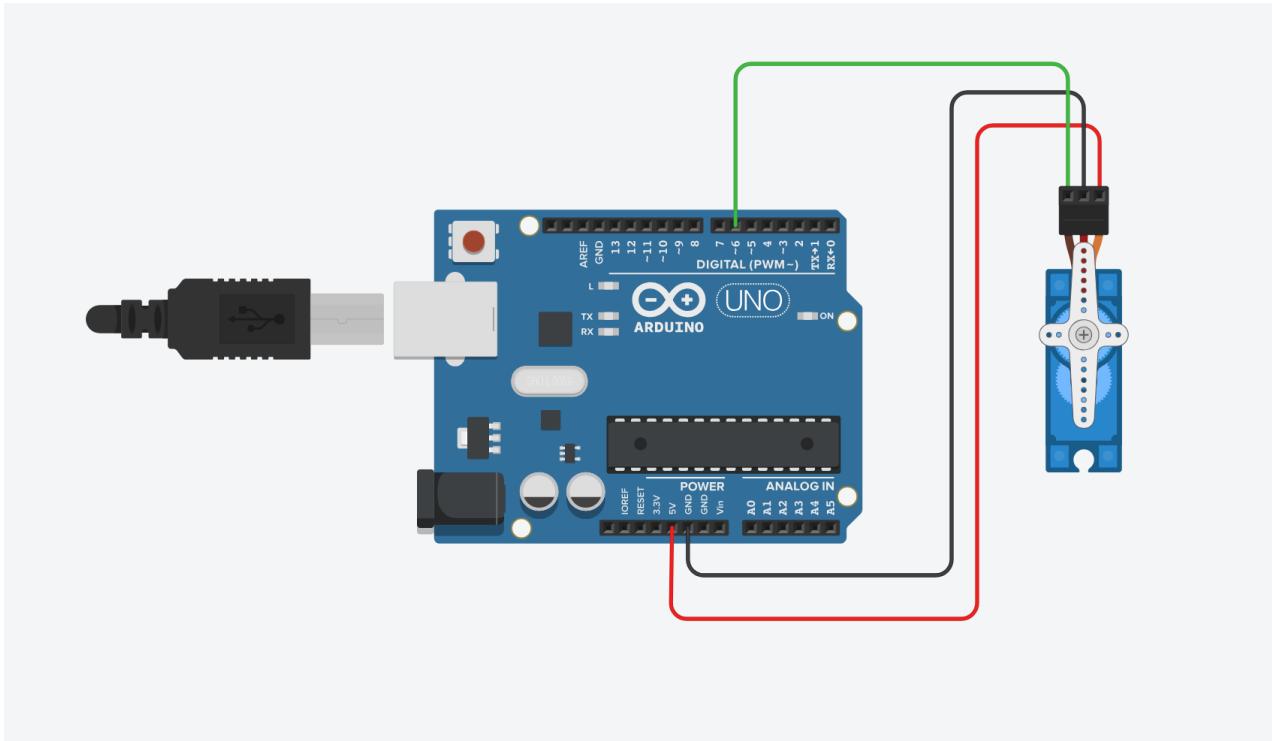


Figura 4: Conexiunile servomotorului SG90 cu Arduino UNO

Servo motorul a fost conectat astfel:

- VCC → 5 V
- GND → GND
- Semnal PWM → D6

Fragmentul de cod embedded utilizat pentru controlul barierei este:

```

1 #include <Servo.h>
2 Servo myservo;
3
4 void setup() {
5     myservo.attach(6);
6     myservo.write(0);
7 }
8
9
10 void deschideBariera() {
11     myservo.write(90);
12     delay(3000);
13     myservo.write(0);
14 }
```

### 3.4.4 Display LCD 16×2 cu adaptor I2C

Display-ul LCD 16×2 a fost utilizat ca interfata vizuala principală a sistemului, avand rolul de a afisa in timp real numarul de locuri ramase, mesajele de acces: permis sau respins si statusul barierei.

Pentru a optimiza numarul de pini utilizati pe microcontroller, display-ul a fost conectat printr-un modul adaptor I2C, care a functionat in arhitectura master-slave, Arduino UNO fiind master-ul magistralei.

Modulul a operat la tensiunea de 5 V, cu adresa standard 0x27, si a permis transmisia de date pe doar 2 pini: SDA si SCL, comparativ cu varianta paralela 4-bit care ar fi necesitat minimum 6 pini.

Avantajul utilizarii I2C a fost consumul redus de resurse hardware, simplitatea comunicatiei si stabilitatea in transmisie pentru distante scurte.

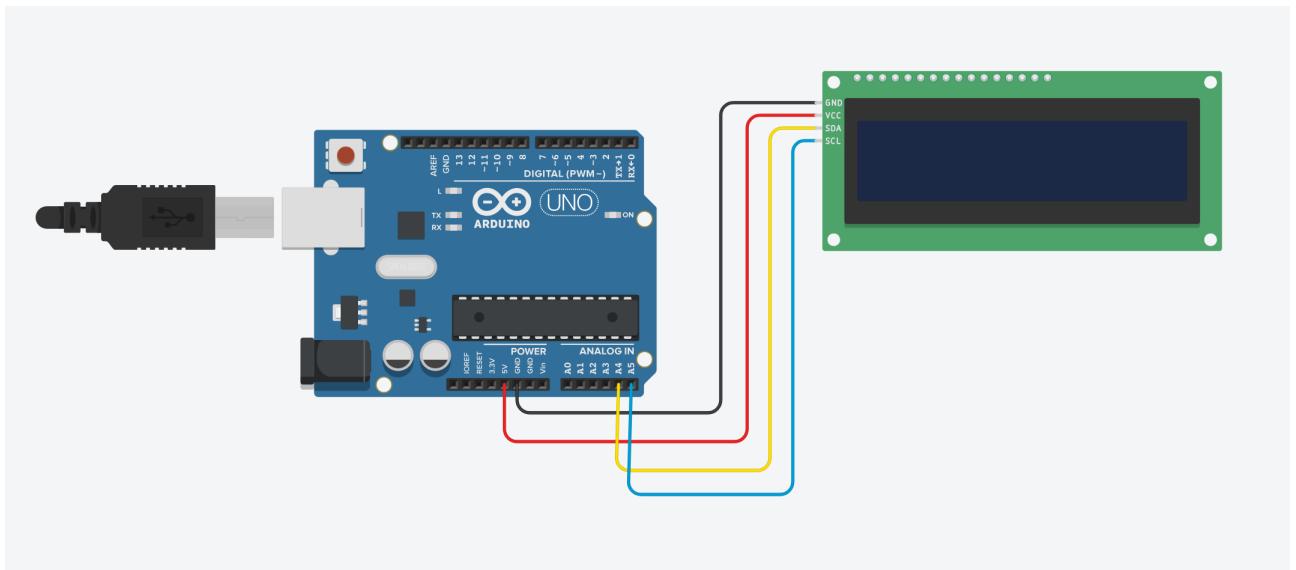


Figura 5: Conexiunile LCD-ului 16x2 cu I2C impreuna cu Arduino UNO

LCD-ul cu I2C a fost conectat astfel:

- VCC → 5V
- GND → GND
- SDA → A4
- SCL → A5

Fragmentul de cod embedded utilizat pentru controlul afisorului este:

```
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C lcd(0x27, 16, 2);
4
5 void setup() {
6     lcd.init();
7     lcd.backlight();
8     lcd.clear();
9     lcd.print("Parcare Ready");
10 }
```

```

12
13 void loop() {
14     lcd.clear();
15     lcd.setCursor(0, 0);
16     lcd.print("Locuri:");
17     lcd.print(locuriRamase);
18
19     if (valid && locuriRamase > 0) {
20         lcd.setCursor(0, 1);
21         lcd.print("Acces permis");
22         deschideBariera();
23         locuriRamase--;
24     } else {
25         lcd.setCursor(0, 1);
26         lcd.print("Acces respins");
27         delay(1500);
28     }
29 }
30

```

### 3.4.5 Modul Bluetooth HC-05

Modulul Bluetooth HC-05 a fost utilizat pentru a oferi un canal wireless de control al barierei, permitand transmiterea comenziilor de la un dispozitiv extern (telefon) direct catre microcontroller.

HC-05 functioneaza ca transceiver serial UART, insa in cadrul proiectului a fost conectat prin interfata SoftwareSerial, deoarece portul UART hardware al placutei Arduino UNO este deja utilizat pentru debugging prin cablul USB.

Modulul opereaza la 5V pe pinul VCC, dar liniile de comunicatie TX si RX sunt de 3.3 V logic-level, ceea ce il face compatibil cu Arduino. HC-05 permite configurarea modului Master sau Slave, iar in proiect a fost utilizat in mod Slave, Arduino initiind conexiunea si interpretand comenziile primite.

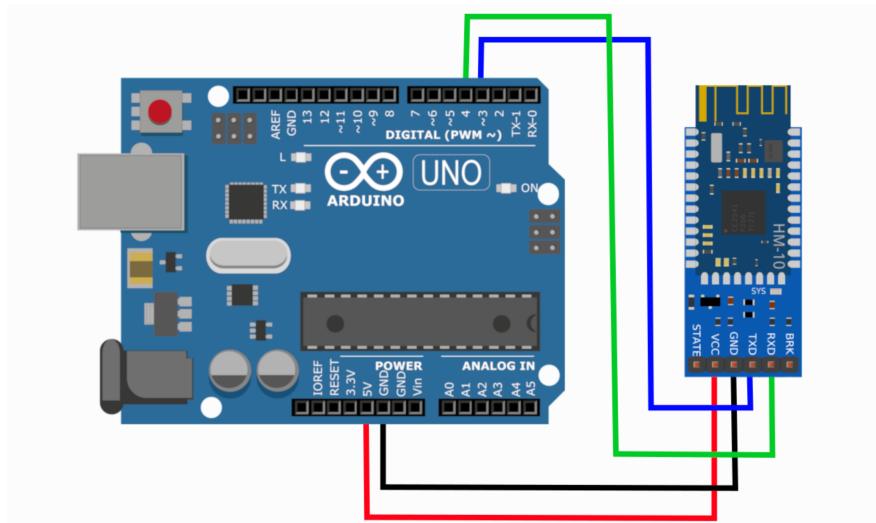


Figura 6: Conecarea modulului Bluetooth HC-05 la Arduino UNO prin SoftwareSerial

Modulul Bluetooth a fost conectat astfel:

- VCC → 5 V
- GND → GND
- TX → D2
- RX → D3

Fragmentul de cod embedded utilizat pentru initializarea interfetei Bluetooth si interpretarea comenziilor a fost:

```

1 #include <SoftwareSerial.h>
2 SoftwareSerial bt(2, 3);
3
4 void setup() {
5     Serial.begin(9600);
6     bt.begin(9600);
7 }
8
9 if (bt.available()) {
10     String cmd = bt.readStringUntil('\n');
11     cmd.trim();
12     cmd.toUpperCase();
13
14     Serial.print("BT: ");
15     Serial.println(cmd);
16
17     if (cmd == "OPEN") {
18         lcd.clear();
19         lcd.print("BT OPEN");
20         deschideBariera();
21     }
22 }
23

```

### 3.4.6 Alimentare, interconectare si detalii de cablare globale

Sistemul a utilizat o alimentare comună de 5V furnizată prin Arduino UNO, cu excepția modulului RFID care a fost alimentat la 3.3V, conform specificațiilor electrice ale modulului RC522.

Toate componentele au împărtit un GND comun, pentru a asigura referința corectă a semnalelor electrice între module.

Controlul barierei a fost realizat prin semnal PWM pe pinul D6, iar comunicația Bluetooth a utilizat pinii D2/D3 prin SoftwareSerial.

Magistrala I2C a fost utilizată pe pinii A4/A5 pentru LCD, iar interfața SPI a fost utilizată pe pinii D10–D13 pentru RFID RC522.

## 3.5 Concluzii hardware

Partea de hardware a proiectului a demonstrat integrarea corecta a modulelor embedded utilizand protocoale standard de comunicatie precum: SPI, I2C și UART, optimizand resursele microcontroller-ului.

Cablarea a fost realizata modular, utilizand un numar minim de pini, cu alimentare corespunzatoare pentru fiecare componenta si GND comun, asigurand functionare stabila in prototipare.

Modulele RFID, ultrasonic, servo și Bluetooth au cooperat corect cu unitatea centrala Arduino, oferind un sistem embedded complet functional pentru controlul accesului automat.

### 3.6 Structura software, functii si interactiunea cu hardware

Software-ul embedded al sistemului de parcare a fost dezvoltat în C/C++ utilizand Arduino IDE, avand ca rol principal procesarea datelor provenite de la senzori, validarea accesului contactless și controlul barierei în mod local, fără server intermediu. Programul este structurat în 3 module functionale:

1. control wireless Bluetooth
2. autentificare RFID prin SPI
3. detectia vehiculelor la ieșire folosind ultrasunete prin semnale digitale de tip Trigger/Echo

Fiecare dintre aceste module este sincronizat cu LCD-ul I2C, care afiseaza statusul parcarii în timp real.

La pornirea sistemului, funcția `setup()` inițializează toate interfețele hardware: UART (serial monitor), SPI (pentru RC522), I2C (pentru LCD) și PWM (pentru servo), asigurând configurația de funcționare master-slave acolo unde este cazul (I2C). Tot aici sunt setați pinii utilizați pentru comunicație și control: D2 și D3 pentru Bluetooth prin SoftwareSerial, D10 și D13 pentru magistrala SPI a modulului RFID RC522, D4 și D5 pentru senzorul ultrasonic HC-SR04, și D6 pentru controlul PWM al servomotorului SG90.

Functia `deschideBariera()` reprezinta elementul central de actionare mecanica. Aceasta comanda servomotorul la 90 grade (bariera deschisa), mentine pozitia 3 secunde pentru trecerea vehiculului, apoi revine la 0 grade (bariera inchisa). Aceasta functie este apelata atat in urma validarii unui UID autorizat la intrare, cat si in urma detectarii unei masini la ieșire sau a unei comenzi externe Bluetooth.

Autentificarea RFID este realizata în bucla principala `loop()`, unde microcontroller-ul verifica prezensa unui card nou prin `PICC_IsNewCardPresent()` și citeste UID-ul acestuia prin `PICC_ReadCardSerial()`. UID-ul este comparat octet cu octet cu un UID valid predefinit in array-ul `validUID[4]`. Daca UID-ul corespunde și exista locuri disponibile, sistemul afiseaza „Acces permis” pe LCD și decrementeaza contorul de locuri libere. In caz contrar, se afiseaza „Acces respins” și bariera ramane inchisa.

Pentru ieșire, senzorul ultrasonic genereaza un impuls TRIG de 10µs, apoi se masoara durata ecoului reflectat prin `pulseIn(echoPin, HIGH, 30000)`. Distanta este calculata conform relatiei fizice standard:  $d = \frac{0.034 \cdot t}{2}$ , unde 0.034 cm/µs este viteza sunetului. Daca o masina este detectata sub 10 cm și flagul de detectie permite, bariera este deschisa automat, iar contorul de locuri este incrementat, fara a depasi capacitatea maxima a parcarii.

In ansamblu, software-ul realizeaza legatura logica dintre toate componente hardware, implementand: decizii locale rapide, comunicatii modulare fără conflicte de interfete și protectii prin flag-uri care previn actionari multiple ale barierei sau actualizari eronate ale contorului de locuri. Sistemul a functionat în timp real, demonstrând o cooperare stabila intre SPI, I2C, UART software și PWM.

## 4 Testarea sistemului de parcare intelligent

Testarea hardware a confirmat funcționarea corecta a tuturor componentelor interconectate cu Arduino UNO. Modulul RFID RC522 a fost testat prin comunicatia SPI pe pinii SDA/SS=D10, RST=D9, SCK=D13, MOSI=D11, MISO=D12, alimentat la 3.3V, iar referinta electrica a fost asigurata prin GND comun. In demo, apropierea unui card a activat antena, iar LED-ul intern al modulului a confirmat alimentarea stabila. UID-ul citit a fost transmis imediat catre microcontroller, fara pierderi de date.

Senzorul ultrasonic HC-SR04 a fost testat ca modul principal pentru detectarea vehiculului la iesire. Impulsul TRIG generat de Arduino pe pinul D4 a avut o durata de 10  $\mu$ s, iar semnalul ECHO receptionat pe pinul D5 a fost masurat prin functia pulseIn(). Distanta a fost calculata conform relatiei fizice standard  $d = \frac{0.034 \cdot t}{2}$ , confirmata prin monitorul serial. Senzorul a detectat obiecte la 10 cm cu o eroare sub  $\pm 0.5$  cm, demonstrand precizie suficienta pentru demo.

Servomotorul SG90 a fost testat prin PWM pe pinul D6. Bariera a fost actionata mecanic intre 0 grade (inchis) si 90 grade (deschis). Miscarea servo-ului a fost fluida, fara vibratii, iar revenirea in pozitia initiala (inchis) a fost complet automata.

Interfata LCD 16x2 a fost testata prin adaptorul I2C la adresa 0x27, utilizand pinii SDA = A4 si SCL = A5. Magistrala a functionat in mod master-slave, Arduino fiind master-ul. Datele au fost transmise fara blocaje, iar afisajul a fost actualizat la fiecare 300 ms conform buclei demo, demonstrand functionare real-time.

Testarea software a verificat toate functiile embedded care interactioneaza cu hardware-ul. In demo, microcontroller-ul a:

1. detectat un card nou prin PICC\_IsNewCardPresent()
2. a citit UID-ul prin PICC\_ReadCardSerial()
3. l-a comparat local cu array-ul validUID[4]
4. a actualizat contorul locuriRamase fara a depasi limitele [0, totalLocuri]
5. a utilizat flag-ul masinaIesireDetectata pentru a preveni declansarile multiple ale servo-ului
6. a interpretat comenzi externe prin Bluetooth folosind SoftwareSerial bt(D2, D3)

La nivel functional, demo-ul a urmat fluxul:

1. Card detectat la intrare  $\rightarrow$  UID valid  $\rightarrow$  LCD afiseaza „Acces permis”  $\rightarrow$  Servo deschide bariera  $\rightarrow$  locuriRamase-
2. Obiect detectat la iesire  $\geq 10$  cm  $\rightarrow$  LCD afiseaza „Iesire...”  $\rightarrow$  Servo deschide bariera  $\rightarrow$  locuriRamase++
3. Comanda telefon Bluetooth „OPEN”  $\rightarrow$  LCD afiseaza „BT OPEN”  $\rightarrow$  Servo actioneaza bariera

Demo-ul a demonstrat astfel un sistem embedded complet functional, cu validare contactless sigura, detectare fizica, actualizare in timp real a locurilor pe LCD si control extern wireless fara conflicte de interfata. Testele functionale au confirmat ca bariera s-a deschis doar in conditiile autorizate, iar contorul de locuri a fost gestionat corect in toate scenariile demo.

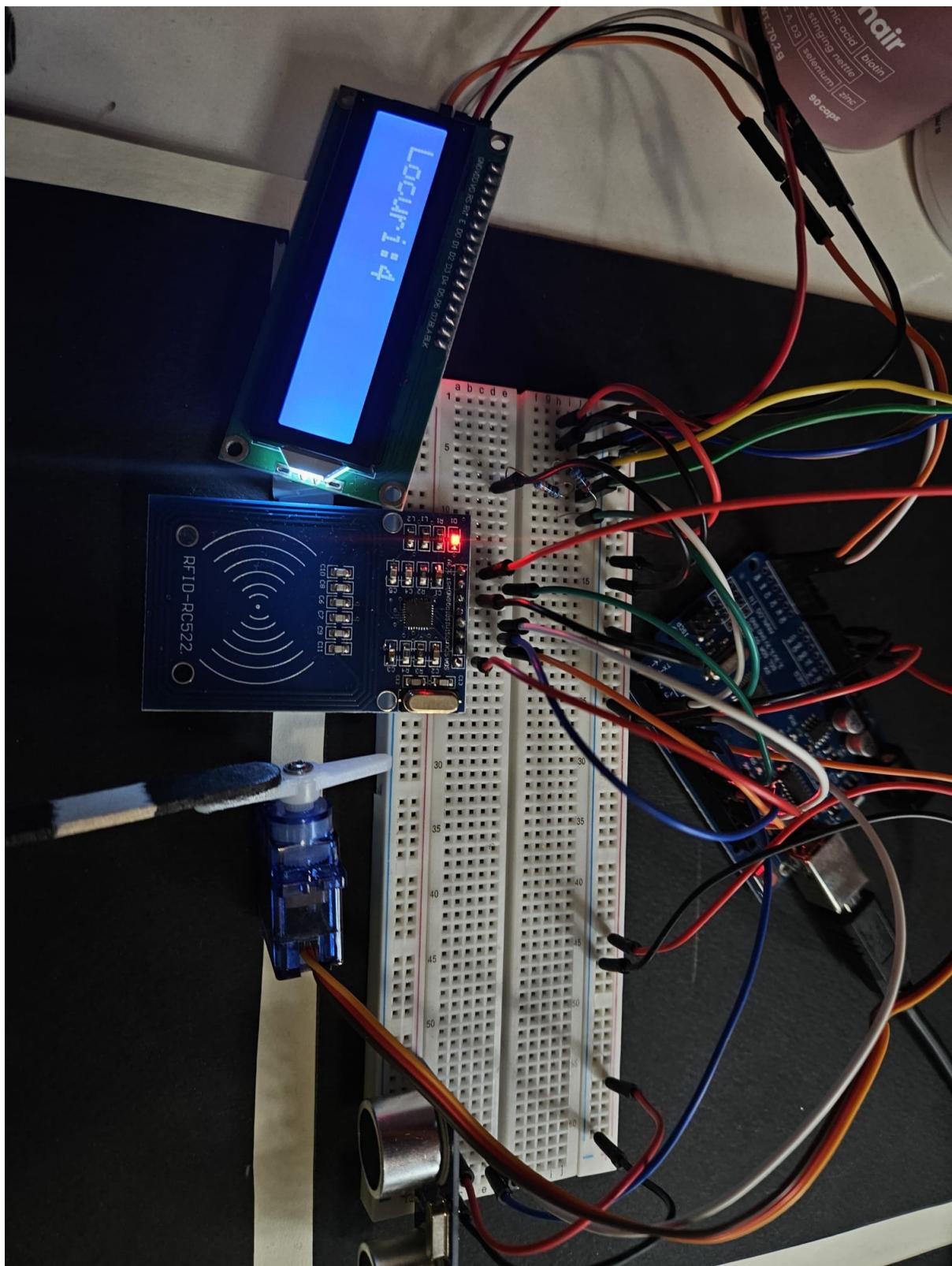


Figura 7: DEMO1

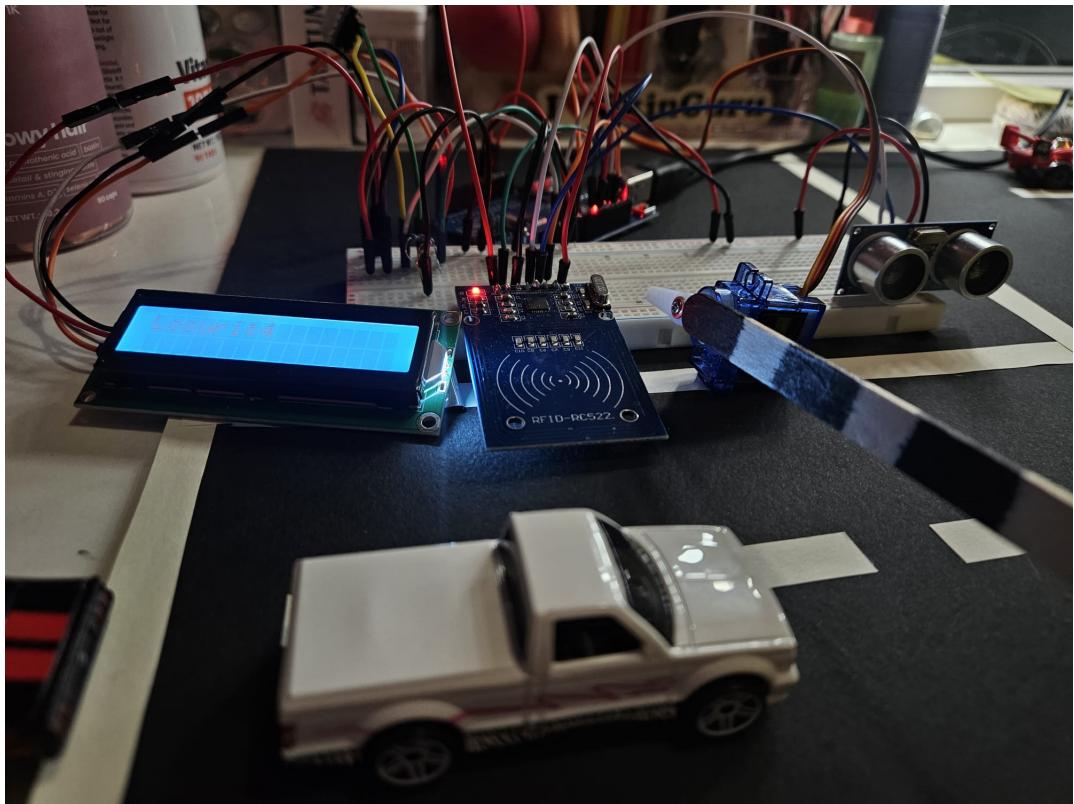


Figura 8: DEMO2

#### 4.1 Concluzii finale de testare

Testarea a validat integrarea completa hardware-software, demonstrand un prototip stabil, modular, precis și optimizat ca număr de pini, potrivit pentru extindere ulterioară într-un sistem real de smart parking.