Universitatea Politehnică Timișoara
Facultatea de Automatică si Calculatoare
An universitar 2023-2024

SISTEM DE MONITORIZARE A NUMĂRULUI DE LOCURI LIBERE ÎNTR-O PARCARE

SISTEME INCORPORATE

Bulharu Alina | Balint Patricia - anul III, CTI-Ro, gr. 1.1 și 1.2

Profesor coordonator: Banu-Taran Paul-Cristian

Caracteristicile proiectului:

(Sistem de monitorizare a numărului de locuri libere într-o parcare, implementat cu ajutorul unei machete demonstrative și incluzând o barieră comandată de un servomotor)

• Se vor utiliza fotodiode sau senzori de detecţie a obiectelor reflectorizante („reflective object sensors”) sau senzori cu ultrasunete pentru detectarea maşinilor care intră si ies. Se pot folosi alte tipuri de senzori conform cu decizia proiectantului.

• Modalitatea de legare a senzorilor va cuprinde cât mai puţine fire (se vor alege interfeţe seriale, precum I2C, CAN, LIN etc).

• Se va asigura afişarea numărului de locuri libere din parcare (afişaje cu segmente sau matrice de LED-uri sau afişaj LCD).

• Numărul iniţial al locurilor libere de parcare va fi prestabilit.

• Aplicaţia va dispune de 2 LED-uri: unul din ele va fi pornit atâta vreme cât mai există locuri libere, al doilea va fi pornit când nu vor mai fi locuri disponibile în cadrul parcării. Nu este posibilă pornirea concomitentă a celor două LED-uri.

• Demonstrarea practică a proiectului va include o machetă cu un anumit număr de locuri de parcare delimitate, iar la intrarea în parcare va fi montată o barieră comandată de un servomotor. Accesul unui automobil în parcare este permis prin ridicarea barierei doar dacă mai este cel puțin un loc liber disponibil.

La fiecare ridicare a barierei pentru intrare în parcare, numărul de locuri libere disponibile va fi decrementat, iar la fiecare ridicare a barierei pentru ieșire din parcare, numărul de locuri libere va fi incrementat.

• Codul sursă va trebui să ţină cont în redactare de constrângerile specifice care pot apărea în cadrul unui sistem încorporat.

Descrierea hardware-ului

Sistemul proiectat conține următoarele componente:

1. placă ArduinoUNO
2. 4 senzori ultrasonici pentru detectarea maşinilor
3. 2 servomotoare (bariere)
4. 2 LED-uri (rosu, verde)
5. afişaj LCD 16x2 I2C cu numărul de locuri de parcare libere

1.Placa Arduino1 (centrală)

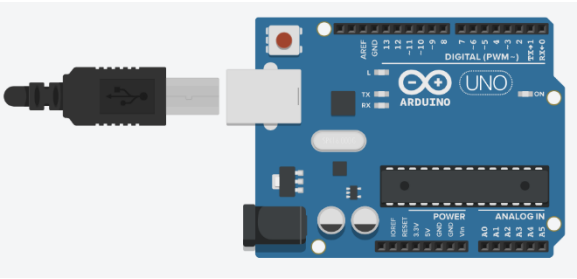
Funcționalitate: Placa Arduino UNO reprezintă centrul de comandă al întregului sistem, preluând datele de la senzori, controlând servomotoarele și gestionând afișajul LCD. Arduino UNO oferă o platformă flexibilă pentru programare și este ușor de integrat cu diverse componente electronice. Aceasta permite o comunicare eficientă și rapidă cu componentele externe datorită utilizării piniilor digitali și I2C.

Conexiuni:

Pini digitali: Utilizați pentru conectarea senzorilor ultrasonici, servomotoarelor, afișajului LCD și LED-urilor.

- **Senzori ultrasonici:** Se conectează la pinii D2-D7 și D10-D11 pentru trimiterea și primirea semnalelor de control (Trig și Echo).
- **Servomotoare:** SERVO1 se conectează la pinul D9, iar SERVO2 la pinul D3 pentru a primi semnale PWM și a fi alimentate.
- **Afișaj LCD:** Utilizarea pinii A4 (SDA) și A5 (SCL) pentru interfața I2C, plus alimentarea la 5V și GND.
- **LED-uri:** LED1 se conectează la pinul D8 și LED2 la pinul D2, ambele printr-o rezistență de 220 ohmi.

- **Pini analogici:** Utilizați pentru citirea valorilor analogice, dacă este necesar.
- **Pini de alimentare:** Asigură tensiune de alimentare (5V sau 3.3V) și masă (GND) pentru toate componentele.
- **Alți pini speciali:** De exemplu, pinul RESET pentru resetarea plăcii Arduino.



2.Senzorii ultrasonici pentru detectarea mașinilor

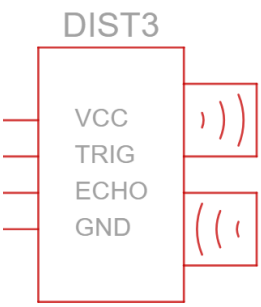
Funcționalitate: Acești senzori sunt folosiți pentru a măsura distanța până la mașinile care intră sau ies din parcare, asigurând o detecție precisă.

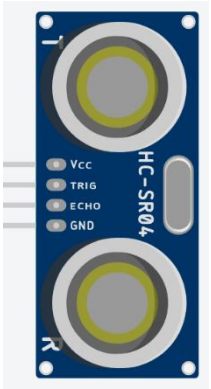
Principiul de funcționare: Senzorii emit unde ultrasonice și măsoară timpul necesar pentru ca aceste unde să se întoarcă de la un obiect. Distanța până la obiect este calculată pe baza acestui timp.

Conexiuni:

- **Trig (Trigger):** Pin de control care declanșează emisia de unde ultrasonice, conectat la un pin digital al plăcii Arduino.
- **Echo:** Pin care primește semnalul de întoarcere de la obiectul detectat, conectat la un alt pin digital al plăcii Arduino.
- DIST1 este conectat la pinii D13 și D12, DIST2 la pinii D11 și D10, DIST3 la pinii D7 și D6, și DIST4 la pinii D5 și D4.

Sunt utilizați patru senzori ultrasonici, câte doi pentru intrarea în parcare (DIST1 și DIST2) și doi pentru ieșirea din parcare (DIST3 și DIST4). Fiecare senzor are un pin pentru trimiterea semnalului (Trig) și unul pentru recepționarea ecoului (Echo), asigurând detecția precisă a vehiculelor.





3.Servomotoare (bariere)

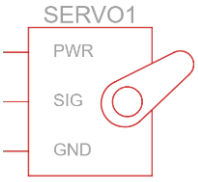
Funcționalitate: Servomotoarele controlează barierele care permit sau restricționează accesul în parcare.

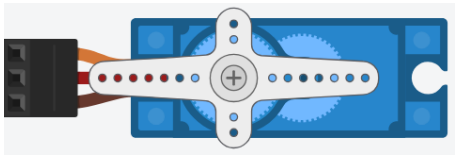
Principiul de funcționare: Servomotoarele se poziționează la un anumit unghi în funcție de semnalele PWM primite de la placa Arduino, astfel ridicând sau coborând barierele.

Conexiuni:

- **VCC (Alimentare):** Se conectează la sursa de alimentare (de obicei 5V).
- **GND (Masă):** Se conectează la masă (GND) a plăcii Arduino.
- **Control (PWM):** Pin de control care primește semnale PWM pentru a poziționa servomotorul, conectat la un pin digital al plăcii Arduino.
- SERVO1 este conectat la pinul D9, iar SERVO2 la pinul D3.

Sistemul folosește două servomotoare: unul pentru controlul barierei de intrare și celălalt pentru controlul barierei de ieșire. Servomotoarele sunt alimentate de la sursa de 5V a plăcii Arduino și primesc semnale PWM pentru a se poziționa corespunzător.





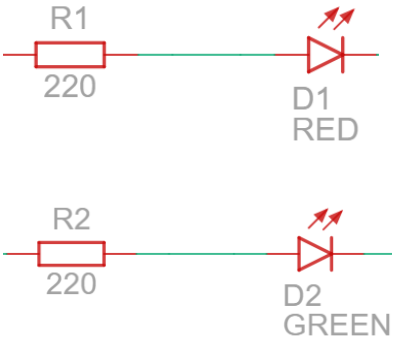
4.LED-uri

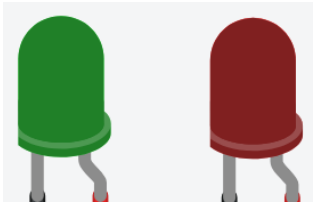
Funcționalitate: Sistemul include două LED-uri care indică statusul locurilor de parcare disponibile. LED-urile oferă utilizatorilor indicatoare vizuale clare și imediate despre disponibilitatea locurilor de parcare:

- LED1:Acesta va fi pornit atâta timp cât mai există locuri libere în cadrul parcării.
- LED2: Acesta va fi pornit când nu vor mai fi locuri disponibile în parcarea respectivă.
- Nu este posibilă pornirea concomitentă a celor două LED-uri
- LED-urile sunt conectate prin intermediul unor rezistențe pentru a limita curentul.

Conexiuni:

- **LED1:** este conectat la pinul D8 printr-o rezistență de 220 ohmi.
- **LED2:** este conectat la pinul D2 printr-o rezistență de 220 ohmi.





5.Afișajul LCD cu numărul de locuri de parcare libere

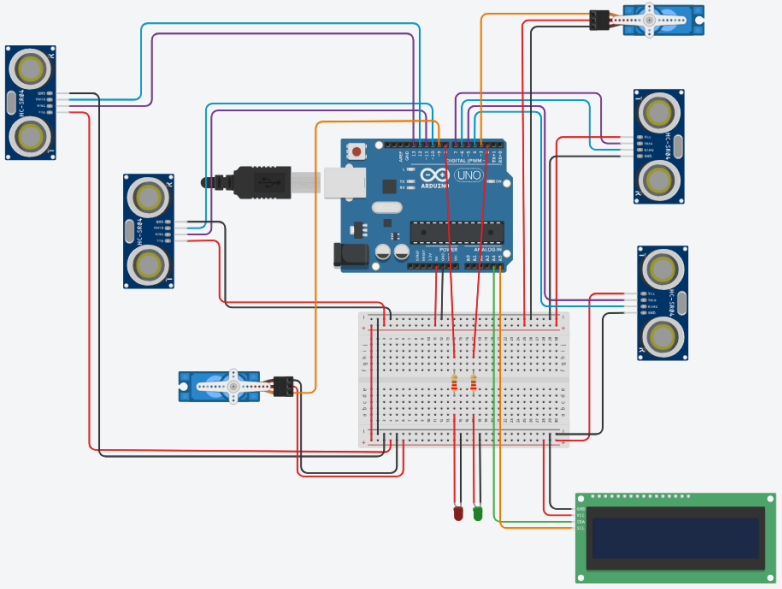
Funcționalitate: Afișajul LCD arată numărul de locuri de parcare disponibile, oferind utilizatorilor o vizualizare clară și actualizată a situației. Acesta afișează în timp real numărul de locuri de parcare disponibile.

Conexiuni:

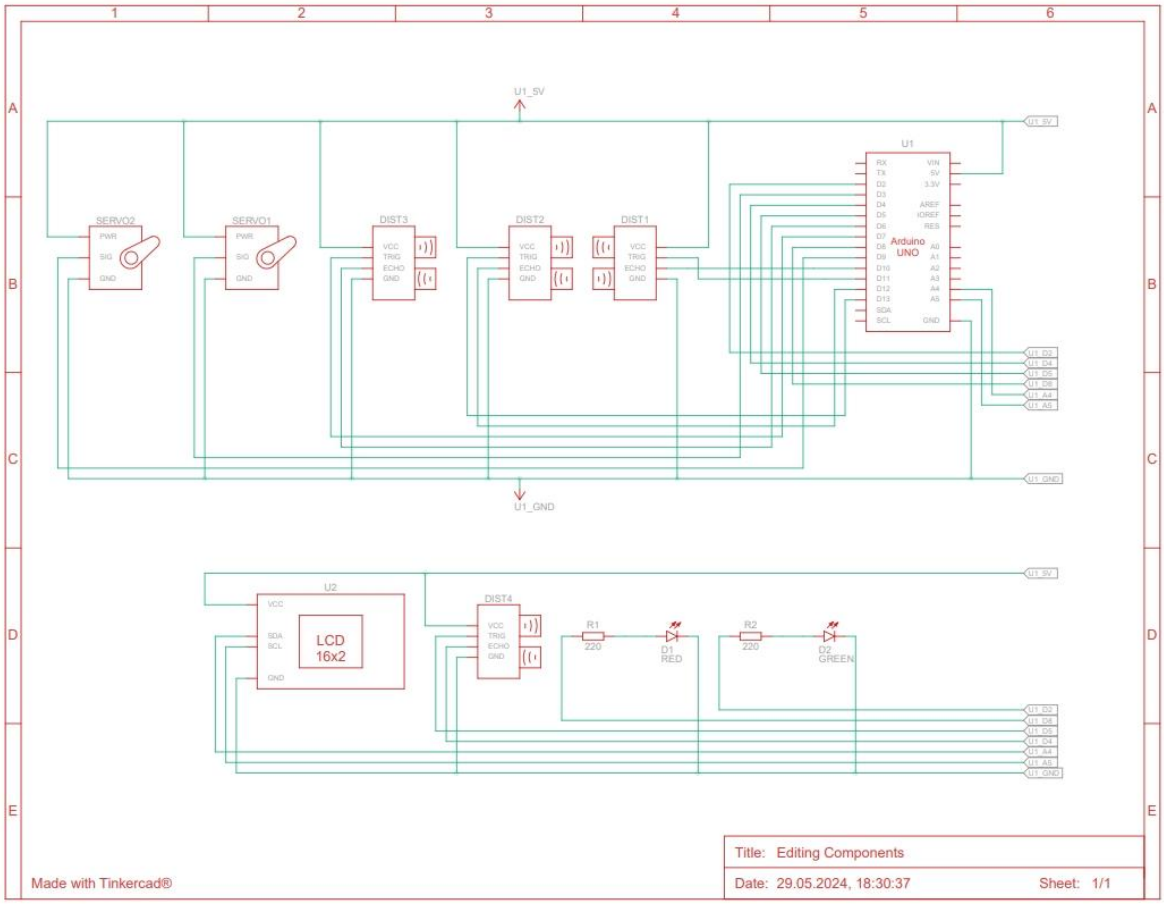
- **VCC (Alimentare):** Se conectează la sursa de alimentare (de obicei 5V).
- **GND (Masă):** Se conectează la masă (GND) a plăcii Arduino.
- **SCL (Serial Clock Line) și SDA (Serial Data Line):** Pini folosiți pentru comunicarea serială între Arduino și afișajul LCD, asigurând transferul datelor necesare afișării.
- **LCD 16x2** este conectat la interfața I2C a plăcii Arduino, utilizând pinii A4 (SDA) și A5 (SCL) pentru comunicare, plus alimentarea la 5V și GND.

Afișajul LCD utilizează interfața I2C pentru a comunica cu placa Arduino, ceea ce permite un transfer eficient de date. Comunicarea prin I2C asigură o conectare simplă și eficientă, reducând numărul de fire necesare pentru conectare.

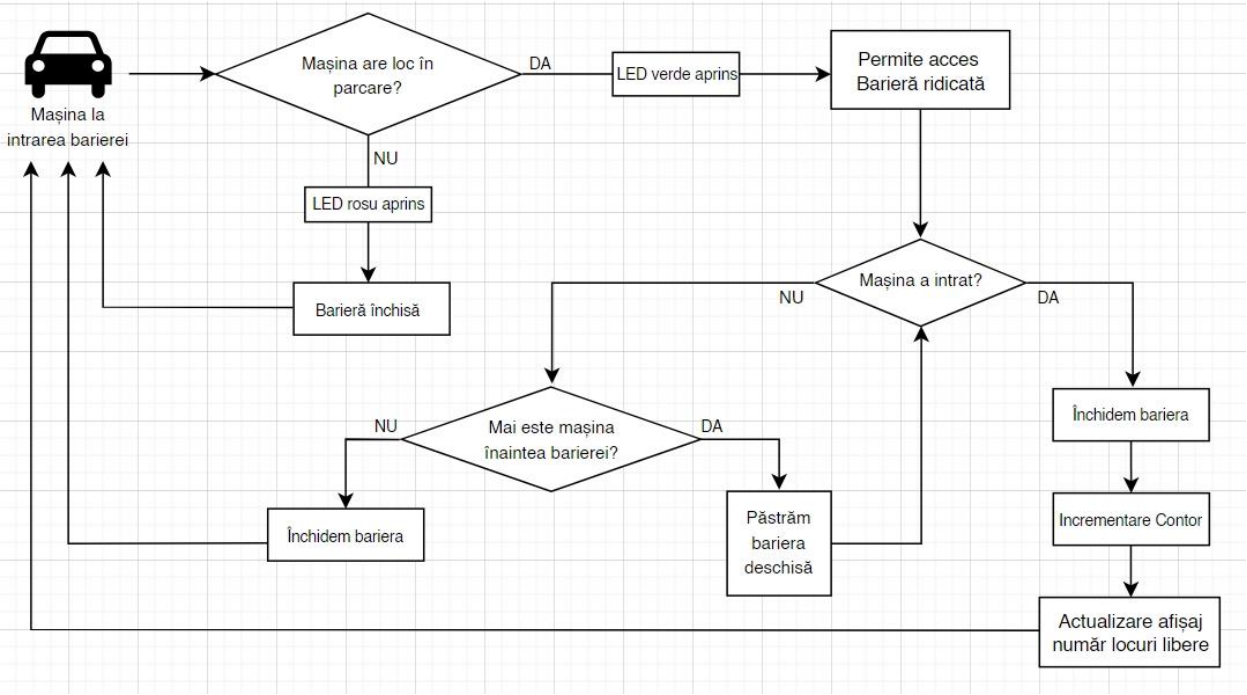
Arhitectura sistemului:



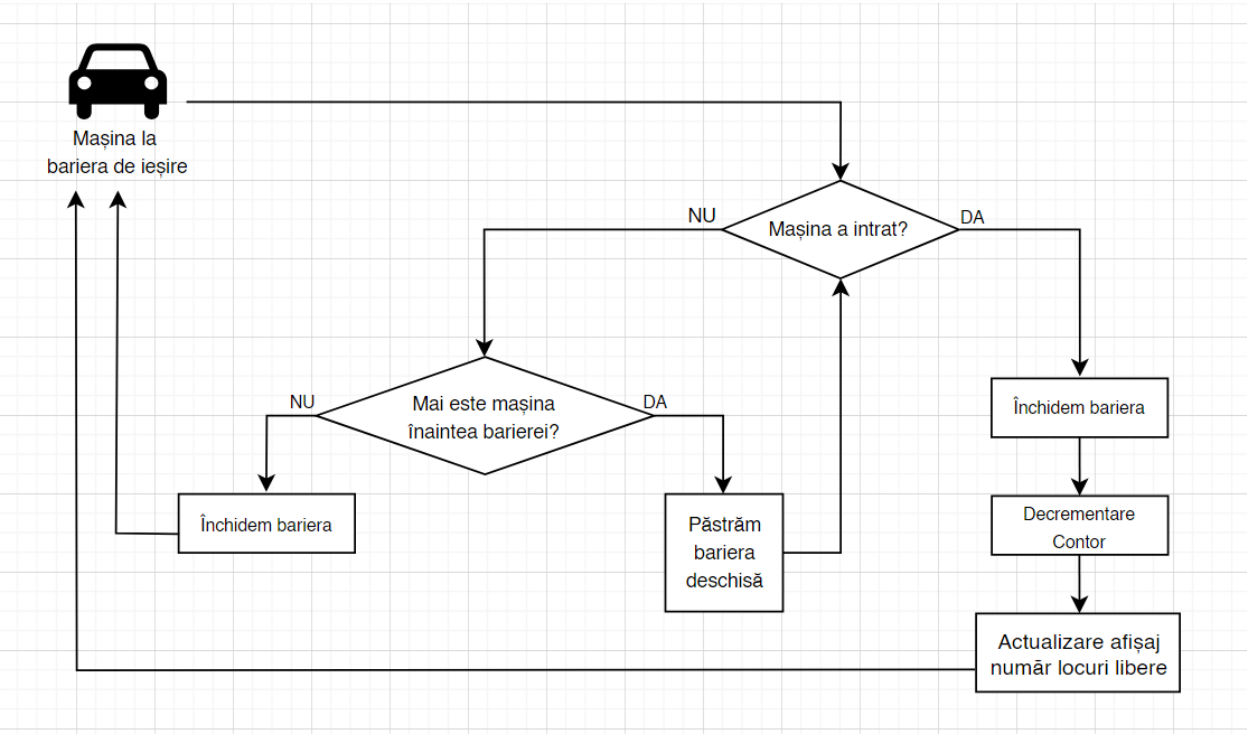
Schema Hardware:



Schma logică – intrarea în parcare:



Schma logică – ieșirea din parcare:



Cod:

```
#include <Servo.h> // Include biblioteca Servo
#include <Wire.h> // Include biblioteca Wire pentru comunicare I2C
#include <LiquidCrystal_I2C.h> // Include biblioteca LiquidCrystal_I2C pentru LCD

// Definire pinii pentru servomotoare și senzori ultrasonici
#define servoPin1 9
#define servoPin2 3

#define trigPin1 13
#define echoPin1 12

#define trigPin2 11
#define echoPin2 10

#define trigPin3 7
#define echoPin3 6

#define trigPin4 5
#define echoPin4 4

// Definire pinii pentru LED-uri
#define rosu 8
#define verde 2

// Definire distanțele minime și maxime
#define MIN_DISTANCE 3.0
#define MAX_DISTANCE 12.0
```

```
LiquidCrystal_I2C lcd_1(0x27, 16, 2); // Inițializare LCD cu adresa I2C 0x27, 16 coloane și 2 rânduri

Servo servo1, servo2; // Creare obiecte pentru servomotoare

int nr_locuri = 5; // Variabilă pentru a urmări locurile de parcare disponibile

// Funcție pentru a verifica dacă valoarea distanței este într-un interval valid
bool checkValue(long value) {
  return value >= 0.0 && value <= 500.0;
}

// Funcție pentru a obține distanța de la un senzor ultrasonic
long getSensorDistance(int trigPin, int echoPin) {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1; // Conversie a timpului în distanță
  return distance;
}
```

```
void setup() {
  Serial.begin(9600); // Pornire comunicare serială la 9600 baud

  // Inițializare senzori ultrasonici
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  pinMode(trigPin3, OUTPUT);
  pinMode(echoPin3, INPUT);
  pinMode(trigPin4, OUTPUT);
  pinMode(echoPin4, INPUT);

  // Inițializare LED-uri
  pinMode(rosu, OUTPUT);
  pinMode(verde, OUTPUT);

  // Inițializare LCD
  lcd_1.begin();
  lcd_1.clear();
  lcd_1.backlight();

  // Atașare și setare poziții inițiale pentru servomotoare
  servo1.attach(servoPin1);
  servo1.write(0); // Bariera intrare inchisa
  servo2.attach(servoPin2);
  servo2.write(0); // Bariera iesire inchisa
}
```

```
void loop() {
   // Obținerea distanțelor de la senzori
  long intrare1 = getSensorDistance(trigPin1, echoPin1);
  long intrare2 = getSensorDistance(trigPin2, echoPin2);
  long iesire1 = getSensorDistance(trigPin3, echoPin3);
  long iesire2 = getSensorDistance(trigPin4, echoPin4);

  // Afișare număr locuri de parcare disponibile
  print_locuri_libere(nr_locuri);

  // Verificare senzori de intrare
  if (checkValue(intrare1)) {
    bariera_intrare(intrare1, intrare2);
  }

  // Verificare senzori de ieșire
  if (checkValue(iesire1)) {
    bariera_iesire(iesire1, iesire2);
  }

  delay(100);
}

// Funcție pentru controlul barierei de intrare
void bariera_intrare(long intrare1, long intrare2) {
  if (nr_locuri > 0) { // Dacă sunt locuri disponibile
     digitalWrite(rosu,LOW); // Oprire LED roșu
    digitalWrite(verde,HIGH);  // Pornire LED verde

    if (intrare1 > MIN_DISTANCE && intrare1 <= MAX_DISTANCE) {
```

```
      Serial.println("Bariera ridicata pentru intare!");
      servo1.write(90); // Ridicare bariera
      delay(1000);
    }

    if (intrare2 > MIN_DISTANCE && intrare2 <= MAX_DISTANCE) {
      Serial.println("Bariera coborata dupa intrare!");
      servo1.write(0);  // Coborâre bariera
      nr_locuri--; //Decrementare număr locuri disponibile
      delay(1500);
    }
  } else {// Dacă nu sunt locuri disponibile
    digitalWrite(verde,LOW);  // Oprire LED verde
    digitalWrite(rosu,HIGH);  // Pornire LED roșu
    servo1.write(0);  // Asigurare că bariera este jos
  }
}

// Funcție pentru controlul barierei de ieșire
void bariera_iesire(long iesire1, long iesire2) {
  if (nr_locuri < 5) {  // Dacă parcarea nu este plină
    if (iesire1 > MIN_DISTANCE && iesire1 <= MAX_DISTANCE) {
      Serial.println("Bariera ridicata pentru iesire!");
      servo2.write(90);  // Ridicare bariera
      delay(1200);
    }


if (iesire2 > MIN_DISTANCE && iesire2 <= MAX_DISTANCE) {
      Serial.println("Bariera coborata dupa iesire!");
```

```
      servo2.write(0);  // Coborâre bariera
      nr_locuri++;  // Creștere număr locuri disponibile
      delay(1200);
    }
  }
}

// Funcție pentru afișarea numărului de locuri de parcare disponibile pe LCD
void print_locuri_libere(int nr_locuri) {
  lcd_1.clear();
  lcd_1.setCursor(0, 0);
  lcd_1.print("Bine ati venit!");
  if (nr_locuri > 0) {
    lcd_1.setCursor(0, 1);
    lcd_1.print(nr_locuri);
    lcd_1.print(" loc. disponibile");
  } else {
    lcd_1.setCursor(0, 1);
    lcd_1.print("Parcare plina!");
    delay(500);
  }
}
```

Bibliografie:

1. https://kompremos.com/ro/controlul-servomotorului-cu-un-arduino/
2. https://www.hwlibre.com/ro/hc-sr04/
3. https://docs.arduino.cc/learn/electronics/lcd-displays/