

# **UNIVERSITATEA POLITEHNICA TIMIȘOARA**

## **FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

### **SISTEME INCORPORATE**

#### **Sistem de monitorizare a numărului de locuri libere într-o parcare**

Lunguleasa Eugen Alexandru  
Mitru Bogdan Alexandru  
CTI RO, Anul III

An universitar 2023-2024

## 2. Enunțul 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 (2 studenți)

Caracteristici:

- Se vor utiliza fotodiode sau senzori de detecție a obiectelor reflectorizante („reflective object sensors”) sau senzori cu ultrasunete pentru detectarea mașinilor care intră și ies. Se pot folosi alte tipuri de senzori conform cu decizia proiectantului.
- Modalitatea de legare a senzorilor va cuprinde cât mai puține fire (se vor alege interfețe seriale, precum I2C, CAN, LIN etc).
- Se va asigura afișarea numărului de locuri libere din parcare (afișaje cu segmente sau matrice de LED-uri sau afișaj LCD).
- Numărul inițial al locurilor libere de parcare va fi prestabilit.
- Aplicația 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 parării. 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 țină cont în redactare de constrângerile specifice care pot apărea în cadrul unui sistem încorporat.

Acest proiect constă în monitorizarea locurilor disponibile dintr-o parcare prevăzută cu 2 bariere, fiecare fiind comandată de un servomotor. Mașinile care intră în parcare sunt detectate de un senzor plasat astfel încât distanța detectată până la mașină să fie mai mică decât 5 cm, dar mai mare de 2. Dacă mașina se află în dreptul senzorului și respectă distanța, atunci bariera va fi ridicată cu ajutorul servomotorului. Bariera va rămâne ridicată atât timp cât mașina nu e detectată de senzorul de pe sensul opus de mers la o distanță de 7-9 cm. Când mașina e

detectată de senzor, bariera va fi coborâtă. La ieșirea mașinilor se aplică același principiu. LCD-ul folosit va afișa la început locurile de parcare disponibile și pe măsură ce mașinile intră în parcare, numărul de locuri va fi decrementat, iar când mașinile ies din parcare numărul de locuri va fi incrementat. De asemenea, mai există 2 LED-uri, unul verde care e aprins cât timp mai există locuri de parcare și unul roșu care e aprins cât timp nu mai sunt locuri disponibile.

### **3. Descrierea plăcii de dezvoltare utilizată în proiect**

#### **Placa de dezvoltare Arduino Uno:**

Arduino Uno este o placă de microcontroler bazată pe ATmega328P. Are 14 pini digitali de intrare/ieșire (dintre care 6 pot fi utilizați ca ieșiri PWM), 6 intrări analogice, un rezonator ceramic de 16 MHz (CSTCE16M0V53-R0), o conexiune USB, o priză de alimentare, un conector ICSP și un buton de resetare. Conține tot ceea ce este necesar pentru a susține microcontrolerul; trebuie doar să-l conectați la un computer cu un cablu USB sau să-l alimentați cu un adaptor de la AC la DC sau o baterie pentru a începe. Puteți experimenta cu Uno fără să vă faceți prea multe griji că veți face ceva greșit; în cel mai rău caz, puteți înlocui cipul pentru câțiva dolari și să începeți din nou.

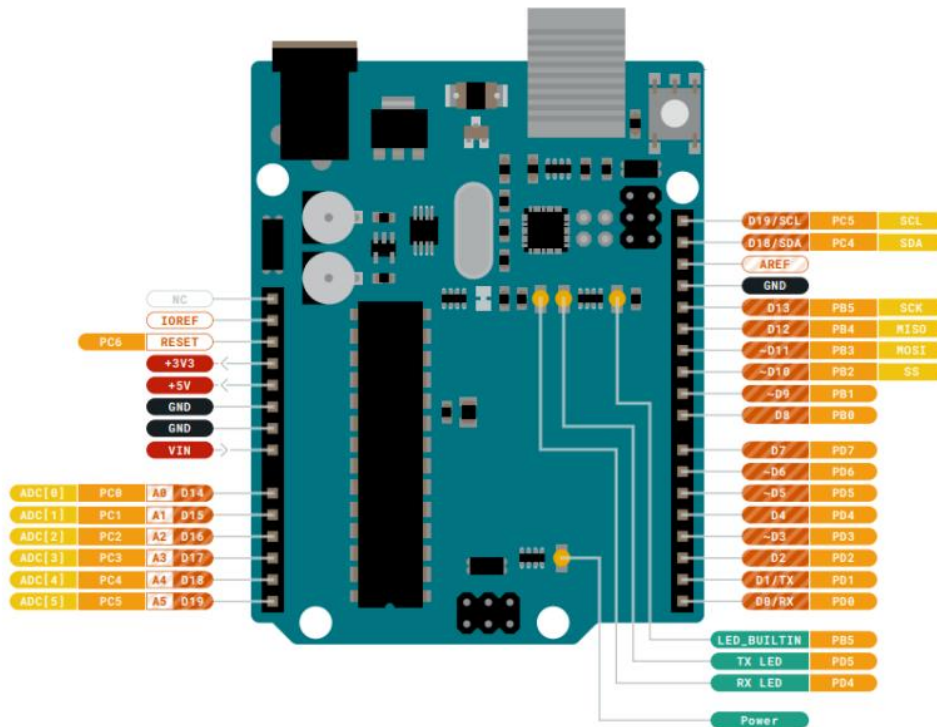
"Uno" înseamnă unu în italiană și a fost ales pentru a marca lansarea Arduino Software (IDE) 1.0. Placa Uno și versiunea 1.0 a Arduino Software (IDE) au fost versiunile de referință ale Arduino, acum evaluate către versiuni mai noi. Placa Uno este prima dintr-o serie de plăci Arduino cu USB și modelul de referință pentru platforma Arduino; pentru o listă extinsă a plăcilor actuale, trecute sau depășite, consultați indexul de plăci Arduino.

Caracteristici tehnice:

- Microcontroler: ATmega328P
- Tensiunea de lucru: 5V
- Pini digitali de intrare/ieșire: 14
- Pini PWM: 6

- Pini de intrare analogici: 6
- Curentul pe un pin de intrare/ieșire: 20mA
- Memoria Flash: 32KB
- SRAM: 2 KB
- EEPROM: 1 KB
- Viteza de Clock: 16 MHz

## 5 Connector Pinouts

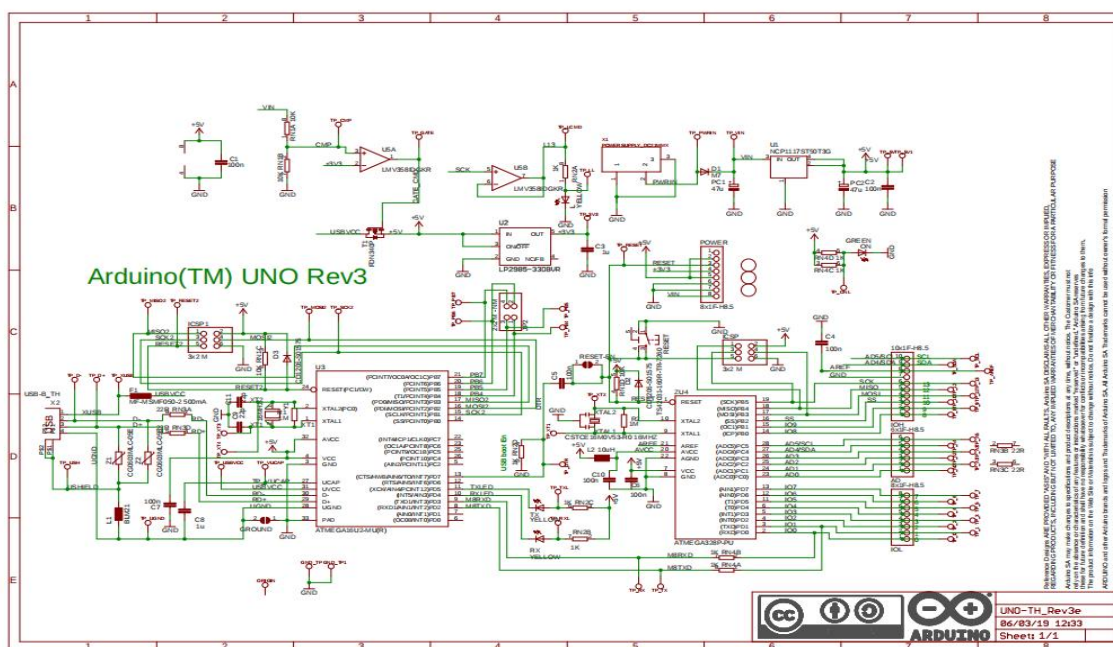


Pinout

## Microcontroler ATMEGA328P

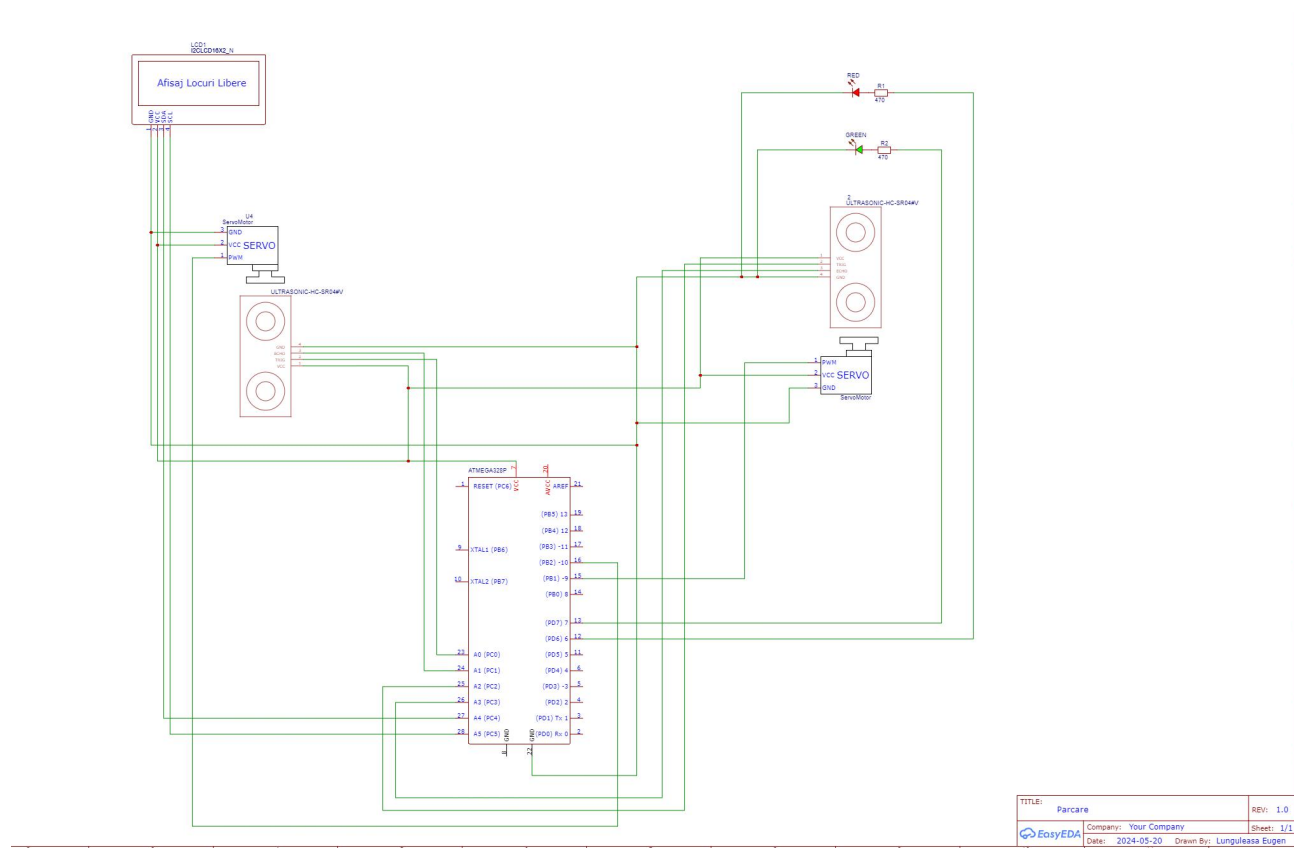
Microcontrolerul de înaltă performanță Microchip picoPower® pe 8 biți bazat pe AVR® RISC combină 32 KB memorie Flash ISP cu capacități de citire-în-timpul-scrierii, 1024B EEPROM, 2 KB SRAM, 23 de linii de I/O de uz general, 32 de registre de lucru de uz general, trei temporizatoare/numărătoare flexibile cu moduri de comparare, întreruperi interne și externe, USART programabil în serie, o interfață serială pe două fire orientată pe octeți, port serial SPI, un convertor A/D pe 10 biți cu 6 canale (8 canale în pachetele TQFP și QFN/MLF), temporizator watchdog programabil cu oscilator intern și cinci moduri de economisire a energiei selectabile prin software. Dispozitivul funcționează între 1,8 și 5,5 volți.

Executând instrucțiuni puternice într-un singur ciclu de ceas, dispozitivul atinge rate de transfer apropiate de un MIPS per MHz, echilibrând consumul de energie și viteza de procesare.



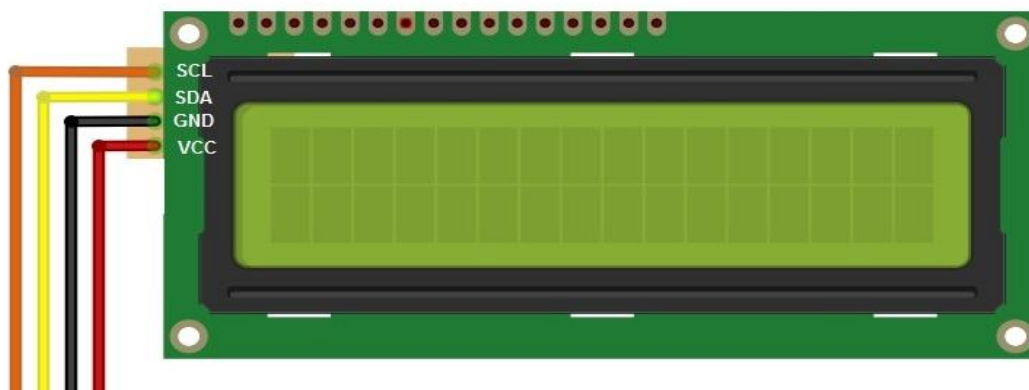
## 4. Arhitectura sistemului

În realizarea proiectului, pentru a lega modulele folosite, am utilizat un breadboard auxiliar. Am conectat servomotoarele astfel: GND la GND, VCC la 5V, PWM la pinul 9, respectiv pinul 10. Cei 2 senzori de distanță au pinii TRIG și ECHO conectați la A2 și A3, respectiv la A1 și A0. În vederea legării dispozitivului LCD, pinii GND, VCC, SDA, SCL au fost conectați la pinii cu aceeași denumire de pe placa Arduino Uno (SDA - A4 și SCL A5). Ledul roșu a fost conectat în serie cu o rezistență de  $470\ \Omega$  la pinul 6, în timp ce ledul verde a fost conectat în mod similar la pinul 7.



## 5. Funcționarea sistemului de afișaj

### Sistem de afișaj – LCD 16x2



Modulul LCD 16x02 afișează 2 linii a câte 16 caractere. Acesta poate fi folosit cu ușurință și în condiții de iluminare joasă, deoarece are și lumină de fundal. Pentru reducerea pinilor, la legarea LCD-ului la controler se poate folosi un modul I2C, astfel având nevoie de doar doi pini, unul pentru SDA și unul pentru SCL.

Caracteristici tehnice:

- Tensiune de alimentare: 4.7-5.3V;
- Curent: 2 mA;
- Tensiune alimentare backlight: 4.2V;
- Curent lumina de fundal: 250mA (MAX)

Principiul de funcționare al unui ecran LCD se bazează pe efectul curentului electric asupra moleculelor de cristal lichid. Sub incidența luminii, moleculele de sticlă polarizată vor bloca parțial trecerea acesteia în anumite porțiuni. Prin urmare, aceste porțiuni vor părea mai întunecate.

## 6. Descrierea tehnică a senzorilor

### Senzor ultrasonic HC-SR04

Caracteristici tehnice:

- Tensiune de alimentare: 5V;
- Curent consumat: 15mA;

- Distanță de funcționare: 2cm - 4m;
- Unghiul de măsurare: 30°;
- Eroare de doar 3mm;
- Durată semnal input: 10us.

Senzorul ultrasonic HC-SR04 este unul dintre cei mai populari și ușor de utilizat senzori de distanță. Acesta are o precizie de 3mm și măsoară la un unghi de 30°. Este ușor de folosit și consumă puțină energie, principalul dezavantaj fiind distanța relativ mică de măsurare: 2 cm - 4 m. Senzorul este foarte ușor de folosit, având doar 4 pini: VCC și GND, care sunt folosiți pentru alimentare și doi pini digitali, utilizați pentru unda emisă (trigger) și unda recepționată (echo).

Acești senzori i-am conectat la plăcuța Arduino după cum urmează:

Senzorul pentru intrare:

const int triggerPin = A2 (emițător)

const echoPin=A3 (receptor)

Senzorul pentru ieșire:

const int triggerPin = A0 (emițător)

const echoPin=A1 (receptor)

### ***Principiu de funcționare***

Senzorul ultrasonic utilizează sonarul pentru a determina distanța față de un obiect:

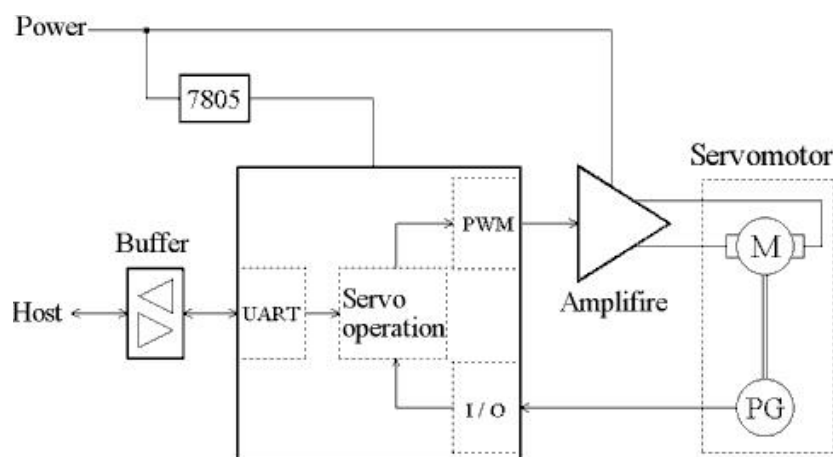
- Transmițătorul (pinul trig) trimite un semnal: un sunet de înaltă frecvență.
- Când semnalul găsește un obiect, acesta se reflectă
- Receptorul (pinul echo) îl primește

Senzorii funcționează în funcție de distanța citită până la obiectul pentru care dorim să acționeze. Măsurarea distanței se bazează pe diferența dintre momentul de timp la care s-a transmis trigger-ul și momentul la care acesta se detectează înapoi. Vom folosi o funcție dist() pentru fiecare senzor, care returnează distanța în centimetri.



## Servomotor

Schema bloc:



Caracteristici tehnice:

- Tensiune de alimentare: 4.8V - 6V;
- Consum redus de curent;
- Viteza de funcționare: 0.12 s/60o @ 4.8 V;
- Cuplu în blocare la 4.8V: 1.8 kgf\*cm;
- Frecvență PWM: 50Hz;
- Temperatura de funcționare: -30° C - +60° C.

### Principiu de funcționare

Un servomotor este un dispozitiv electromecanic care produce cuplu și viteză pe baza curentului și a tensiunii furnizate. Un servomotor funcționează ca parte a unui sistem cu buclă închisă, oferind cuplu și viteză, așa cum este comandat de la un microcontroler, utilizând un dispozitiv de feedback pentru a închide bucla. Servo este un termen general pentru un sistem de control cu buclă închisă.

Se poate controla servomotorul trimițând o serie de impulsuri către linia de semnal. Un servomotor analogic convențional se așteaptă să primească un impuls aproximativ la fiecare 20 de milisecunde (adică semnalul ar trebui să fie de 50Hz). Lungimea impulsului determină poziția servomotorului.

Cei trei pini ai celor două servomotoare folosite vor fi conectați la GND, VCC, iar input-ul va fi pinul 9, respectiv 10 al plăcuței Arduino, deoarece controlul

servomotorului se realizează cu ajutorul unui semnal de tip PWM. Pentru a putea folosi această componentă esențială proiectului nostru, în cod ne vom ajuta de funcțiile predefinite aflate în biblioteca aferentă servomotorului ales (<Servo.h>). Vom declara variabile de tip Servo. Când ridicăm/coborâm barierele, vom seta poziția la care dorim să se poziționeze servomotorul respectiv, folosindu-ne de variabilele declarate anterior.

## 7. Programul + comentarii

```
#include <HCSR04.h>
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//definire variabile asociate servomotoarelor
Servo barrierIn;
Servo barrierOut;

//definire variabile asociate senzorilor
HCSR04 sensorIn(A2,A3);
HCSR04 sensorOut(A0,A1);

//definire variabilă asociată LCD-ului
LiquidCrystal_I2C lcd(0x27, 16, 2);

int sInValue, sOutValue; //definire variabile pentru valorile citite de la senzori
int vacantPlaces = 1; //definire variabilă pentru numărul de locuri din parcare
```

```
int hasEntered = 0, hasExited = 0; //definire variabile flag
```

```
//definire variabile asociate LED-urilor
```

```
int redLed = 6;
```

```
int greenLed = 7;
```

```
void printOnLCD() {  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print(" Locuri libere");  
    lcd.setCursor(7,1);  
    lcd.print(vacantPlaces);  
}
```

```
void barrierIsOpening() {  
    lcd.setCursor(4,0);  
    lcd.print("Bariera");  
    lcd.setCursor(2,1);  
    lcd.print("se deschide");  
}
```

```
void barrierIsClosing() {  
    lcd.setCursor(4,0);  
    lcd.print("Bariera");  
    lcd.setCursor(2,1);  
    lcd.print("se inchide");  
}
```

```
void setup() {  
    Serial.begin(115200); //stabilirea comunicației seriale cu placa  
  
    //atașarea pinului 9/10 servomotorului care controlează bariera de la intrare/ieșire  
    barrierIn.attach(9);  
    barrierOut.attach(10);  
  
    //setarea poziției inițiale a barierei de la intrare/ieșire ca fiind de 80/95 de grade  
    barrierIn.write(80);  
    barrierOut.write(95);  
  
    lcd.begin(); //inițializare LCD  
    lcd.backlight(); //activare lumină de fundal  
  
    //se setează pinii LED-urilor ca OUTPUT  
    pinMode(redLed,OUTPUT);  
    pinMode(greenLed,OUTPUT);  
}  
  
void loop() {  
    //citire distanțe senzori  
    sInValue=sensorIn.dist();  
    sOutValue=sensorOut.dist();  
  
    lcd.begin();
```

```
//aprindere LED-uri în funcție de numărul de locuri libere
if(vacantPlaces > 0)
{
    digitalWrite(greenLed,HIGH);
    digitalWrite(redLed,LOW);
}
if(vacantPlaces == 0)
{
    digitalWrite(redLed,HIGH);
    digitalWrite(greenLed,LOW);
}

//intrarea în parcare
if(sInValue >=1 && sInValue <= 5 && vacantPlaces > 0 && hasEntered == 0)
{
    //ridicarea barierei
    for(int pos=80;pos>=0;pos-=2)
    {
        barrierIsOpening();
        barrierIn.write(pos);
    }
    hasEntered = 1;
    vacantPlaces--;
}
if(hasEntered == 1)
{
    delay(1000);
}
```

```
//coborârea barierei  
for(int pos=0;pos<=80;pos+=2)  
{  
    barrierIsClosing();  
    barrierIn.write(pos);  
}  
hasEntered = 0;  
}  
  
//ieșirea din parcare  
if(sOutValue >=1 && sOutValue <= 5 && hasExited == 0)  
{  
    //ridicarea barierei  
    for(int pos=95;pos>=10;pos-=2)  
    {  
        barrierIsOpening();  
        barrierOut.write(pos);  
    }  
    hasExited = 1;  
    vacantPlaces++;  
}  
if(hasExited == 1)  
{  
    delay(1000);  
    //coborârea barierei  
    for(int pos=10;pos<=95;pos+=2)  
    {
```

```
    barrierIsClosing();  
    barrierOut.write(pos);  
}  
hasExited = 0;  
}  
  
//asigurare că vacantPlaces nu depășește intervalul [0,10]  
if(vacantPlaces > 10)  
{  
    vacantPlaces = 10;  
}  
if(vacantPlaces < 0)  
{  
    vacantPlaces = 0;  
}  
  
//afișare pe LCD  
printOnLCD();  
delay(100);  
}
```

## 8. Bibliografie

1. <https://store.arduino.cc/products/arduino-uno-rev3#:~:text=Arduino%20Uno%20is%20a%20microcontroller,header%20and%20a%20reset%20button.>
2. <https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf>
3. [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)
4. <https://arduinogetstarted.com/tutorials/arduino-lcd-i2c>
5. <https://www.optimusdigital.ro/ro/motoare-servomotoare/26-micro-servomotor-sg90.html>