

Sistem de parcare inteligent

Autor: Toia Emilia Antonia

Grupa: 333AA

Introducere

Prezentarea pe scurt a proiectului vostru:

- Ce face proiectul?

Acest proiect își propune realizarea unei parări inteligente automatizate, controlată cu ajutorul unui microcontroler Arduino. Sistemul este compus din două bariere acționate de motoare, una pentru intrare și una pentru ieșire, controlate automat prin senzori cu infraroșu (IR) care detectează apropierea unei mașini.

În plus, parcare este echipată cu senzori de detecție pe fiecare loc de parcare, astfel încât un ecran LCD să afișeze în timp real numărul locurilor disponibile, precum și starea fiecărui loc (ocupat sau liber).

- Care este scopul lui?

Scopul proiectului este de a **reduce intervenția umană**, de a **automatiza gestionarea locurilor de parcare** și de a **crește eficiența într-un spațiu limitat**.

- Care a fost ideea de la care am pornit?

Ideea a pornit de la observația că, în multe parări mici sau nesupravegheate, se pierde timp prețios căutând loc liber. Sistemul meu vine în întâmpinarea acestei probleme cu o soluție simplă și accesibilă.

- De ce cred că este util?

Utilitatea lui este evidentă:

- Pentru utilizatori: economie de timp și confort;
- Pentru administratori: control mai bun și monitorizare automată;

Descriere generală

Schema bloc de mai jos ilustrează arhitectura generală a proiectului „Parcare Inteligentă cu Arduino UNO”, evidențiind componentele hardware și software implicate și modul în care acestea interacționează între ele.



1. Componente hardware:

- **Arduino UNO**: rol central în sistem, acționează ca unitate de control. Primește informații de la senzori și controlează componentele de ieșire.
- **Senzor IR Intrare & Senzor IR Ieșire**: detectează prezența unei mașini la intrarea și ieșirea din parcare. Transmit un semnal digital către Arduino pentru a activa barierele.
- **4 Senzori de parcare**: monitorizează fiecare loc de parcare. Trimit informații către Arduino despre disponibilitatea fiecărui loc.
- **Motoare Servo - Intrare și Ieșire**: acționează barierele. Arduino trimite comenzi pentru a le ridica/coboară în funcție de prezența mașinii detectată de senzorii IR.
- **Display LCD 16×2**: afișează în timp real câte locuri de parcare sunt disponibile și care sunt ocupate. Primește date de la Arduino.

2. Componente software:

- **Cod Arduino C++**
 - Programează logica de decizie: dacă un senzor IR detectează o mașină și există locuri libere, se ridică bariera, se actualizează numărul de locuri și se afișează pe LCD.
 - Verifică constant toți cei 4 senzori de parcare și actualizează starea acestora.
 - Controlează sincronizarea între senzori, motoare și afișaj.

Hardware Design

Componente hardware:

Componentă	Funcție	Detalii tehnice
Arduino UNO	Unitate de control	14 pini digitali, 6 pini analogici, 5V
Senzor IR Intrare	Detectează mașinile care intră	Emite semnal digital la detecție
Senzor IR Ieșire	Detectează mașinile care ies	Emite semnal digital la detecție
Senzori de parcare	Verifică dacă un loc este ocupat	4 senzori HC-SR04 cu rază ~2-400 cm
Motor Servo Intrare	Ridică/Coboară bariera de intrare	Servo SG90, control PWM
Motor Servo Ieșire	Ridică/Coboară bariera de ieșire	Servo SG90, control PWM
LCD 16×2	Afișează numărul de locuri libere	I2C

Schema electrică:



Diagrama de semnal intrare:



Explicație:

Componentă	Pin	Descriere comportament
IR_SENSOR_1	D3	Detectează mașina la intrare. Semnalul trece de la HIGH la LOW.
SERVO_PIN_1	D9	Primește semnal PWM pentru a deschide bariera (110°), apoi revine la 0°(3 secunde așteaptă).
TRIG_PIN_1	D4	Trimite un impuls de 10 μs către senzorul ultrasonic pentru a începe măsurarea.
ECHO_PIN_1	D5	Primește semnal HIGH proporțional cu distanța până la obstacol (mașina).
LCD	-	Afișează actualizat numărul de locuri libere după intrarea mașinii.

Diagrama de semnal ieșire:



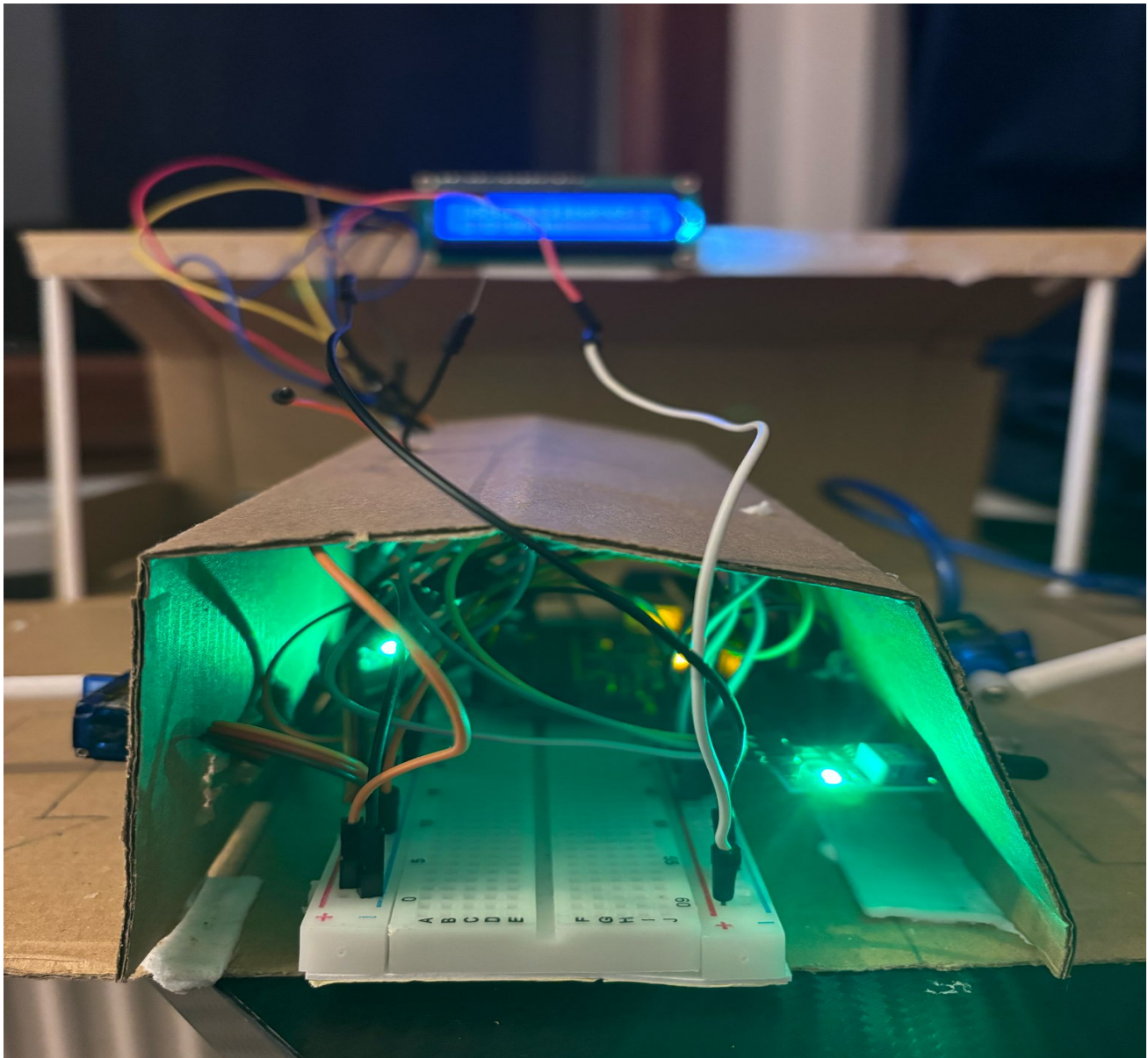
Explicație:

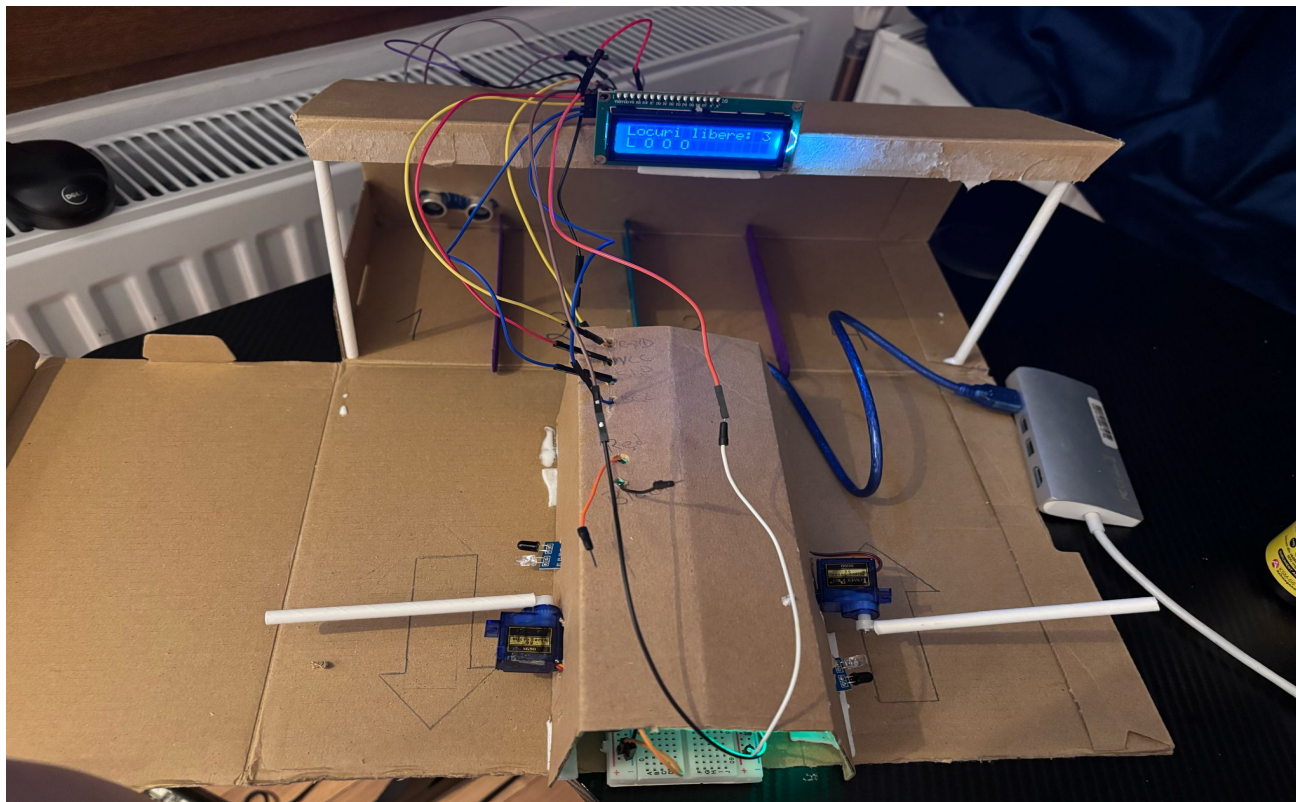
Componentă	Pin	Descriere comportament
IR_SENSOR_2	D2	Detectează mașina care iese din parcare. Semnalul trece de la HIGH la LOW la detectare.
SERVO_PIN_2	D10	Primește semnal pentru a deschide ușa/poarta (semnal HIGH), apoi o închide (semnal LOW).
LCD	-	Afișează numărul de locuri disponibile, actualizat când mașina iese din parcare.

Simulare:



Stadiul actual:





Software Design

1. Mediu de dezvoltare:

- Platformă utilizată: Arduino IDE
- Placă utilizată: Arduino UNO R3
- Limbaj de programare: C/C++ specific Arduino

2. Librării și surse third-party:

- Wire.h - pentru comunicarea I2C.
- LiquidCrystal_I2C.h - pentru controlul unui afișaj LCD 16x2 cu interfață I2C.
- Servo.h - pentru controlul servomotoarelor.

3. Surse și funcții implementate

Funcție	Descriere tehnică	Observații
setup()	Inițializează pinii, LCD-ul și servo-urile	Se apelează o singură dată
loop()	Rulează continuu și gestionează starea sistemului	
updateLCD()	Afișează pe LCD locurile disponibile și starea lor	
checkSlot()	Verifică dacă un loc este ocupat, cu senzor HC-SR04	

handleEntry()	Gestionează intrarea unei mașini	Afișează mesaj dacă parcare e plină
handleExit()	Gestionează ieșirea unei mașini	

4. Integrarea funcționalităților din laborator:

Laborator	Concept folosit în proiect	Unde apare în cod / ce face
GPIO	Configurare pini digitali ca IN/OUT	setup(): pinMode pentru butoane, trig/echo
UART	Comunicare serială pentru debug	Serial.begin() și Serial.print()
PWM	Controlul servomotoarelor prin PWM	Servo library → myservo.write()
I2C	Comunicare cu LCD 16×2 prin I2C	LiquidCrystal_I2C lcd(0x27, 16, 2);

5. Scheletul proiectului & interacțiunea dintre componente:

1. *Intrare/ieșire autoturism:*

Senzorii IR detectează prezența.

Dacă există locuri disponibile, servomotorul se deschide și availableSlots se actualizează.

2. *Monitorizare locuri:*

Fiecare loc de parcare are un senzor HC-SR04 care verifică dacă este ocupat.

Afișajul LCD indică:

- Numărul total de locuri libere (availableSlots – actualizat doar de IR).
- Starea fiecărui loc: L = liber, 0 = ocupat.

3. *Afișaj LCD:*

- Primul rând: numărul de locuri disponibile.
- Al doilea rând: codificare pentru starea locurilor.

6. Demo video și explicații:

https://youtube.com/shorts/tx_SWk4TcRA?feature=share

7. Calibrarea senzorilor:

1. *IR (detecție):*

Testat cu mână/obiect negru pentru a determina distanța la care senzorul IR detectează.

Asigurat ca starea LOW este returnată doar când e cu adevărat blocat.

2. *HC-SR04:*

Testat manual la diferite distanțe.

DIST_THRESHOLD = 10 cm ales ca limită de ocupare – validat cu obiecte reprezentând mașini

8. Optimizări realizate:

Separare logică a sloturilor și IR:

- Elimină fluctuații false în numărul de locuri.
- Asigură stabilitate.

Flag-uri entryFlag și exitFlag:

- Evită modificări multiple de locuri dintr-o singură trecere.
- Elimină necesitatea unor componente debounce hardware.
- Delay redus în loop():
- A crescut timpul de reacție și responsivitatea sistemului.

LCD curățat dinamic:

- Se evită texte fantomă sau caractere rămase de la actualizări.

Rezultate Obținute

În urma implementării proiectului sistemului de parcare automată controlat cu Arduino UNO, s-au obținut următoarele rezultate:

- **Detectarea precisă a vehiculelor la intrare și ieșire:** senzorii IR amplasați la punctele de acces au reușit să detecteze în mod eficient prezența unei mașini, trimițând semnalul corespunzător către Arduino pentru a acționa barierele.
- **Control automat al barierele:** motoarele servo conectate la bariere au răspuns corect comenzilor primite de la Arduino, ridicând sau coborând barierele în funcție de semnalul primit de la senzorii IR. Astfel, s-a asigurat un flux controlat al vehiculelor în parcare.
- **Monitorizarea în timp real a locurilor de parcare:** cei patru senzori de parcare au reușit să identifice starea fiecărui loc (ocupat sau liber), trimițând date precise către unitatea de control.
- **Afișaj informativ pentru utilizatori:** display-ul LCD 16x2 a afișat în timp real numărul de locuri disponibile și cele ocupate, oferind utilizatorilor informații clare și utile la momentul potrivit.
- **Funcționare autonomă și sincronizată a componentelor:** întregul sistem a funcționat sincronizat, fără intervenție umană, demonstrând o bună integrare între senzori, actuatori și unitatea de control Arduino.

Concluzii

Proiectul realizat demonstrează eficiența și viabilitatea utilizării unui sistem automatizat pentru gestionarea locurilor de parcare, bazat pe microcontrolerul Arduino UNO. Implementarea senzorilor IR pentru detecția vehiculelor și a senzorilor de parcare pentru monitorizarea în timp real a locurilor disponibile a dus la crearea unui sistem funcțional, precis și ușor de utilizat.

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

1. Resurse software:

<https://forum.arduino.cc/t/project-on-automatic-parking-system/1083991/15>

<https://marobotic.com/2023/10/18/how-to-make-car-parking-system-using-arduino-and-i2c-lcd-display-automatic-car-parking-system/>

2. Resurse hardware:

http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

<https://www.geeksforgeeks.org/smart-parking-system-using-arduino/>

3. Pentru schema electrică și simulare:

<https://www.tinkercad.com/>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2025/avaduva/emilia_antonia.toia



Last update: **2025/05/25 11:54**