

Mașină cu senzor de parcare și parcare automată - Dincă Alexandra-Cristina

Dincă Alexandra-Cristina, 334CD

Introducere

- Proiectul constă într-o mașină care, la comandă, începe să caute un loc de parcare. În funcție de spațiul găsit, decide dacă va realiza parcare paralelă sau cu spatele și, pe baza sistemului de parcare, va realiza manevrele fără coliziuni. Sistemul dispune de avertizări auditive pentru a informa șoferul asupra obstacolelor de la fiecare pas. În plus, mașina este dotată cu un LCD care afișează distanța până la coliziune în timpul parcării și temperatura când mașina este idle.
- Scopul proiectului este de a simula un sistem de parcare autonomă al unei mașini.
- Ideea de la care am pornit a fost senzorul de parcare, după care am decis să dezvolt proiectul astfel încât mașina să se folosească de informațiile primite pentru a realiza și parcare proprie-zisă.
- Proiectul este util deoarece are aplicabilitate și în viața reală, ușurând procesul de parcare pentru șoferi.

Descriere generală



Alimentarea tuturor componentelor se face dintr-un stabilizator de 5V al unui driver. Cele 2 drivere sunt alimentate din sursă cu 7.5V.

Cele 4 motoare sunt controlate de câte 2 drivere (punte H dublă). Acestea împart pini de input și enable (In1, In2 cu In3, In4 și EnA cu EnB pe fiecare driver) pentru a economisi pini de pe Arduino și pentru că fiecare laterală a masinii primește exact aceleași semnale.

Restul componentelor se conectează direct la plăcuță.

- 2 senzori ultrasunete se conectează la pini digitali, iar al 3-lea la pini analogi (inițial voiam să îi conectez pe toți 3 la pini analogi, dar A4 și A5 interferează cu I2C, iar A1 este necesar senzorului de temperatură)
- LCD I2C se conectează la SDA și SCL
- butonul se conectează la un pin digital
- senzorul de temperatură se conectează la pin analog (A1)

Hardware Design

Lista de piese:

Componentă	Cantitate	Link
Placă de plastic	4	https://www.optimusdigital.ro/ro/mecanica-accesorii-de-prindere/450-placa-din-plastic-cu-gaurigalbena.html?search_query=placa+de+plastic&results=39
Motor cu reductor si roată	4	https://www.optimusdigital.ro/ro/motoare-altele/139-motor-cu-reductor-si-roata.html?search_query=wheel&results=33
Punte H dublă L298N	2	https://www.optimusdigital.ro/ro/drivere-de-motoare-cu-perii/145-driver-de-motoare-dual-l298n.html?search_query=driver+modul+&results=151
Senzor ultrasunete HC SR-04P	3	https://sigmanortec.ro/Senzor-Ultrasunete-HC-SR-04P-3-5-5V-p148477760
Breadboard 400 puncte	2	https://sigmanortec.ro/Breadboard-400-puncte-p129872825
Placă de Dezvoltare Compatibilă cu Arduino UNO R3 (ATmega328p + ATmega16u2)	1	https://www.optimusdigital.ro/ro/placi-avr/4561-placa-de-dezvoltare-compatibila-cu-arduino-uno-r3-atmega328p-atmega16u2-cablu-50-cm.html?search_query=arduino+uno+r3&results=129
LCD 1602 cu interfață I2C	1	https://www.optimusdigital.ro/ro/optoelectronice-lcd-uri/2894-lcd-cu-interfata-i2c-si-backlight-albastru.html?search_query=lcd+i2c&results=17
Senzor de temperatură LM35D	1	https://www.optimusdigital.ro/ro/senzori/1469-senzor-de-temperatura-analogic-lm35d-to-92.html?search_query=lm35&results=2
Buzzer	1	https://sigmanortec.ro/Buzzer-activ-5v-p126421597
Butoane	2	https://sigmanortec.ro/buton-mini-6x6x5-4-pini
Suport de 6 baterii AA	1	https://www.optimusdigital.ro/ro/suporturi-de-baterii/941-suport-de-baterii-2-x-18650.html?search_query=suport&results=600

Scheme electrice:

Schemă circuit:



Schema conține driver-ul L293D, dar proiectul folosește punte dublă H L298N. Am adaptat (dpdv schematic) L293D la L298N astfel încât să simuleze alimentarea plăcuței prin stabilizator. Pini sunt similari (4 in, 4 out, 2 enable), dar diferențele de conectare se pot observa în schemele următoare:

- Alimentarea Arduino se face direct din L298N, folosindu-se de stabilizatorul de 5V:



- Conectarea pinilor (in + out + enables):



Asamblare hardware:

- Testare senzori ultrasunete, LCD, buzzer: <https://youtu.be/x5yYTv9ybwY?si=-RUvqDohS8IVTNfX>
- Testare drivere L298N și motoare: <https://youtube.com/shorts/KdDjAjdK4gs?si=r4qqvSlynApig93P>

Software Design

Mediu de dezvoltare: Arduino IDE

Descrierea flow-ului software:

Mașina stă default în modul idle, până la apăsarea pe buton, moment în care se generează o întrerupere și se începe rutina de căutare a locului de parcare.

- Starea 0 → Idle: la pornirea μC -ului sau la revenirea în starea 0, se măsoară temperatura o dată și se afișează pe LCD
- Starea 1 → Căutare loc de parcare: mașina se mișcă în față și se folosește de senzorul lateral pentru a găsi locul și a determina ce tip de loc e. Când acesta detectează o adâncime de peste 20 de cm, începe să măsoare locul. După ce adâncimea laterală se micșorează, algoritmul aproximează lungimea locului de parcare pe baza timpului petrecut scanând. În funcție de lungime, determină dacă parcare se va realiza lateral sau cu spatele. Senzorul din față are scopul de a opri mașina în cazul în care nu există loc și urmează să se producă o coliziune.
- Starea 2 → Realizare parcare laterală: mașina face o rotație de 45 de grade, dă cu spatele până senzorul din spate detectează impact, și apoi se rotește -45 grade
- Starea 3 → Realizare parcare cu spatele: mașina se poziționează corect (dând cu spatele puțin), se rotește 90 de grade, iar apoi dă cu spatele până când senzorul din spate detectează impact

GitHub:

<https://github.com/alexandradinca2908/Self-parking-car/>

Rezultate Obținute


Parcare laterala: https://youtube.com/shorts/XJ7GHUrpqy0?si=_H5usa8ieqtIjbZN

Parcare cu spatele: <https://youtube.com/shorts/IjI0gOafbaM?si=Wpsz6emT9Ds-nGsz>

Concluzii

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**.

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/fstancu/alexandra.dinca2908> 

Last update: **2025/05/27 23:49**