

# Parking Sensors

## Introducere

Proiectul reprezintă un sistem de senzori de parcare, realizat cu ajutorul unei plăcuțe Arduino. Scopul proiectului este de a ajuta șoferii să parcheze mașinile într-un mod mai precis și sigur.

Ideea a pornit de la observarea faptului că mulți șoferi întâmpină dificultăți în a parca mașina în spații înguste sau aglomerate. De multe ori, aceștia depind doar de propria experiență și de propriile abilități, ceea ce poate duce la accidente sau zgârieturi la caroseria mașinii.

Sistemul de parcare constă într-un set de senzori cu ultrasunete, care detectează obstacolele din jurul mașinii și le afișează pe un ecran LCD. În plus, sistemul are un buzzer care emite semnale sonore în funcție de distanța față de obstacolele detectate.

Consider că acest proiect este util atât pentru ceilalți, cât și pentru mine, deoarece poate fi implementat cu ușurință în orice mașină și poate oferi un plus de siguranță și precizie în timpul manevrelor de parcare. De asemenea, este o soluție economică și ușor de instalat, ceea ce face ca proiectul să fie atractiv și pentru utilizatorii cu mai puțină experiență în electronice și programare.

## Descriere generală

## Simulare Tinkercad



## Schemă electrică



## Hardware Design

Piese utilizate:

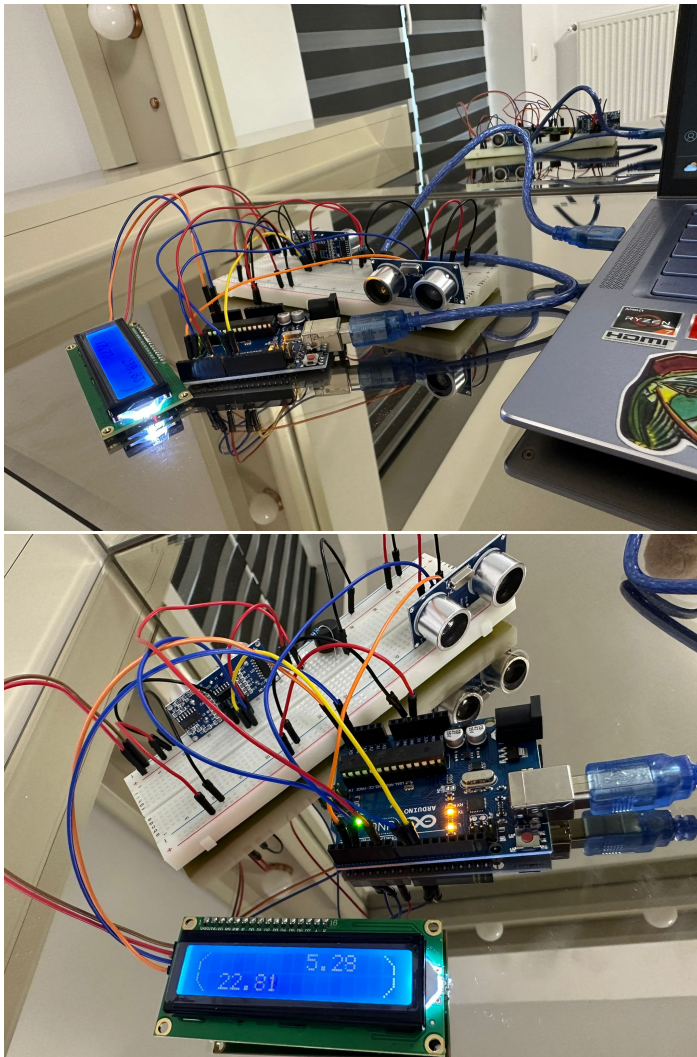
- Arduino UNO R3 (ATmega328P)
- Modul afișaj LCD cu I2C
- 2 x Senzor Ultrasonic HC-SR04
- Breadboard
- Buzzer
- Buton

## Software Design

1. Mediu de dezvoltare: Arduino IDE
2. Biblioteci și surse 3rd-party:
  - "usart.h": Gestionează comunicarea serială.
  - "Ultrasonic.h": Oferă funcționalitatea pentru măsurarea distanței utilizând senzorii ultrasonici.
  - "LiquidCrystal\_I2C.h": Permite controlul unui afișaj LCD utilizând protocolul I2C.
3. Algoritmi, structuri și funcții implementate:
  - standBy(): Afișează un mesaj de stare pe ecranul LCD în modul de așteptare.
  - showDistances(): Afișează distanțele măsurate pe afișajul LCD utilizând caractere personalizate.
  - usartHandler(): Gestionează primirea comenzilor prin comunicarea serială și interpretează comenzile primite. Funcția primește caractere de la comunicarea serială până la primirea caracterului newline (\n).
  - setup(): Inițializează și configurează componentele sistemului.
  - loop(): Este responsabilă pentru efectuarea măsurătorilor și actualizarea afișajului LCD în mod repetitiv. Funcția citește distanțele de la senzorii ultrasonici și aplică algoritmi de control pentru a determina frecvența și durata sunetului buzzerului în funcție de distanțele măsurate.
  - Funcția de întrerupere (ISR) pentru INT0: Gestionează întreruperea generată de apăsarea butonului. Atunci când butonul este apăsat, se determină pornirea sau oprirea măsurătorilor.

## Rezultate Obținute

<https://youtu.be/P5qeTL85x6k>



## Concluzii

Proiectul a fost relativ ușor din punct de vedere hardware, deoarece a implicat utilizarea unor componente comune. Configurarea și conectarea acestora nu a fost dificilă, iar schema hardware a proiectului nu a necesitat componente complexe sau personalizate.

Pe de altă parte, proiectul a fost considerabil mai dificil din punct de vedere software. Aceasta se datorează restricției impuse de utilizarea registrelor și de necesitatea de a implementa algoritmi de control manual. De exemplu, gestionarea comunicării seriale prin intermediul registrelor și interpretarea comenzilor primite au necesitat o abordare detaliată și atentă.

Totuși, dacă afișarea distanței este o sarcină aparent simplă și facilă, de ce mașina mea nu o face?

## Download

[Cod sursă](#)

## Jurnal

07.05.2023 - Documentația inițială

18.05.2023 - Schema electrică & Hardware design

28.05.2023 - Software Design

## Bibliografie/Resurse

### Resurse Hardware:

- <https://www.robofun.ro/>
- <https://www.optimusdigital.ro/en/>

### Resurse Software:

- <https://www.arduino.cc/reference/en/>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/avaduva/parkingsensors>



Last update: **2023/05/29 20:20**