

# Sistem parcare

## Introducere

**Nume student:** Costea Maria-Ionela

**Grupă:** 333CD

**Îndrumător:** Florin Stancu

## Descriere

În cadrul acestui proiect, îmi propun să realizez un sistem de parcare cu 4 senzori.

Senzorii vor detecta dacă un autovehicul se află la o distanță prea mică față de un obstacol aflat în față, spatele sau părțile sale laterale. Sistemul va avea încorporat un buzzer care va emite diferite sunete, în funcție de direcția în care există riscul de ciocnire. De asemenea, sistemul de parcare va conține și un LCD pe care va fi afișată distanța până la diverse obstacole.

## Scop

Utilitatea acestui proiect este relativ evidentă. Un sistem de parcare atenționează un conducător auto de diversele obstacole aflate în proximitatea mașinii sale, ducând astfel la evitarea coliziunilor.

## Descriere generală



- Senzorii vor calcula distanța față de obstacolele din jur.
- În cazul în care unul dintre senzori ajunge prea aproape de un obstacol, buzzer-ul va începe să scoată un anumit sunet, diferit pentru fiecare senzor.
- De asemenea, în cazul apropierii prea mari dintre un senzor și un obstacol, LCD-ul va începe să afișeze distanța până la impact.

# Hardware Design

## Lista de piese:

- 1 x Arduino UNO R3
- 4 x Senzor Ultrasonic Distanță HC-SR04
- 1 x Buzzer Pasiv
- 1 x Breadbord
- 1 x LCD 1602 I2C
- Fire



- 1. Conectare senzor ultrasonic de distanta HC-SR04 1 (Left):**
  - VCC la 5V pe Arduino
  - Trig la pinul digital D3
  - Echo la pinul digital D7
  - GND la GND pe Arduino
- 2. Conectare senzor ultrasonic de distanta HC-SR04 2 (Front):**
  - VCC la 5V pe Arduino
  - Trig la pinul digital D2
  - Echo la pinul digital D6
  - GND la GND pe Arduino
- 3. Conectare senzor ultrasonic de distanta HC-SR04 3 (Back):**
  - VCC la 5V pe Arduino
  - Trig la pinul digital D5
  - Echo la pinul digital D9
  - GND la GND pe Arduino
- 4. Conectare senzor ultrasonic de distanta HC-SR04 4 (Right):**
  - VCC la 5V pe Arduino
  - Trig la pinul digital D4
  - Echo la pinul digital D8
  - GND la GND pe Arduino
- 5. Conectare LCD 1602 I2C:**
  - VCC la 5V pe Arduino
  - SDA la pinul analogic A4
  - SCL pinul analogic A5
  - GND la GND pe Arduino
- 6. Conectare Buzzer Pasiv:**
  - + la pinul digital D12
  - - la GND pe Arduino

# Software Design

**Mediul de dezvoltare:** Arduino IDE

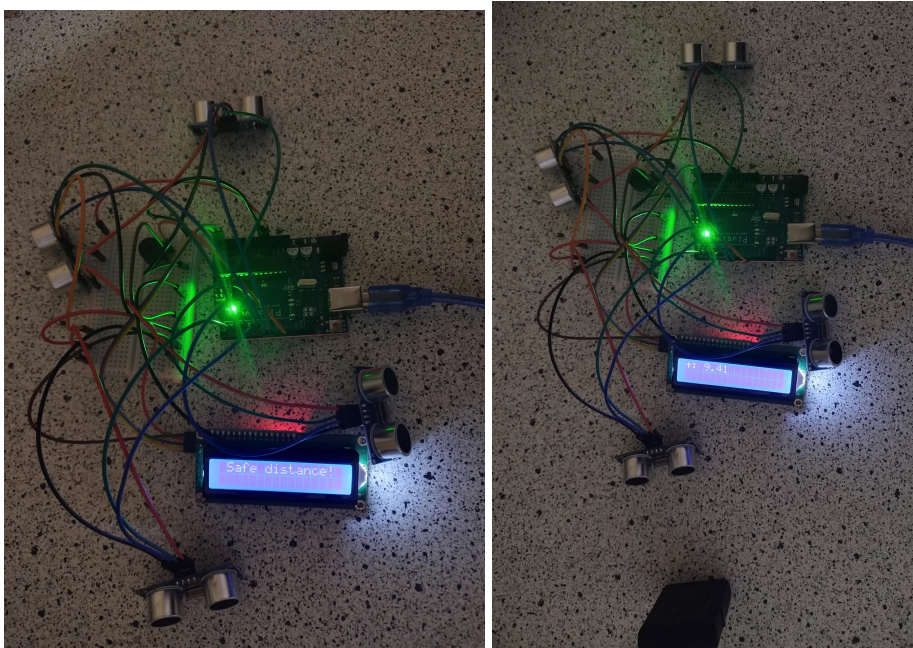
**Biblioteci folosite:** Wire.h, LiquidCrystal\_I2C.h

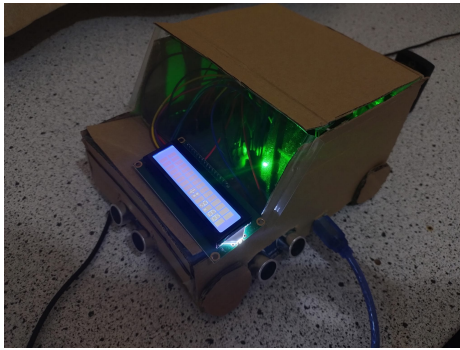
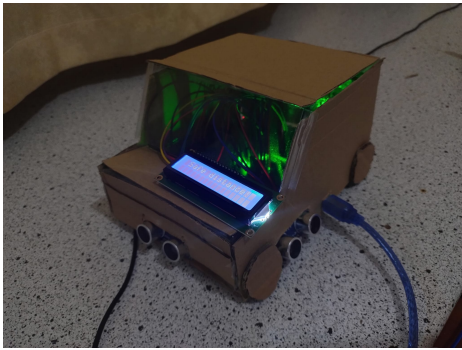
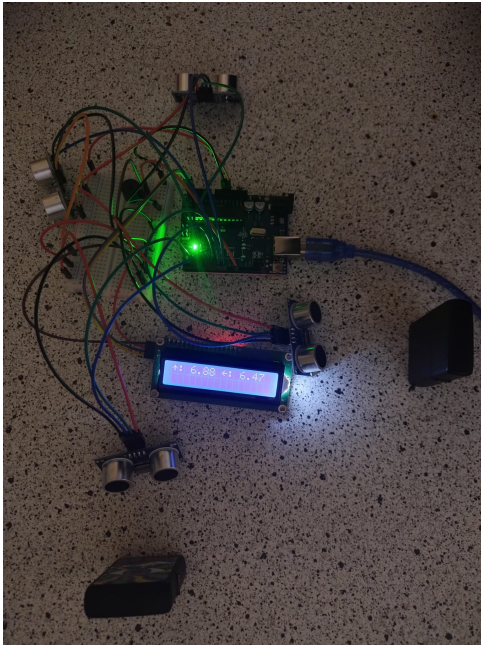
## Funcții:

- **setupSensor:** setează pinii de input și output ai senzorilor HC-SR04;
- **setupLCD:** creează simbolurile de săgeți, folosite la afișarea distanței, și inițializează LCD-ul;
- **setupBuzzer:** setează pinul de output al buzzer-ului;
- **init\_timer1:** inițializează timer-ul 1, în modul CTC, cu prescaler de 1024;
- **setup:** apelează toate funcțiile de setup și inițializare;
- **microsecondsToCentimeters:** transformă microsecundele citite de senzorii HC-SR04 în centimetri;
- **calculateDistance:** calculează distanța de la un senzor la cel mai apropiat obstacol;
- **ISR:** în timpul întreruperii, se calculează distanțele față de fiecare senzor, se rețin distanțele care nu corespund cu limita minimă de apropiere (dacă există) și se activează buzzer-ul, pentru a atenționa șoferul mașinii (în cazul în care mașina se află prea aproape de un obstacol);
- **loop:** face afișarea pe LCD, odată la 500ms.

## Rezultate Obținute

Proiectul funcționează conform așteptărilor.





## Concluzii

Proiectul a fost interesant. Mi-a făcut plăcere să lucrez la el și m-a ajutat să aprofundez și să înțeleg mai bine anumite noțiuni învățate în timpul laboratoarelor de PM.

## Download

[sistem\\_parcare.zip](#)

## Jurnal

- 06.05: Am comandat piesele pentru proiect.
- 08.05: Am primit piesele pentru proiect.
- 11.05: Am implementat partea hardware a proiectului.
- 19.05: Am implementat varianta inițială a software-ului.
- 21.05: Am implementat varianta actuală a software-ului.
- 26.05: Am construit mașinuța din carton.

## Bibliografie/Resurse

### Resurse Hardware:

- [Datasheet Arduino R3](#)

### Resurse Software:

- [Laboratorul 0: GPIO](#)
- [Laboratorul 2: Întreruperi](#)
- [Laboratorul 3: Timere, PWM](#)
- [Laboratorul 6: I2C](#)
- [Millis](#)
- [Timere](#)
- [Caractere LCD custom](#)

### Resurse Hardware+Software (de ajutor atât la asamblarea componentelor, cât și la implementarea software):

- [LCD I2C](#)
- [Senzori HC-SR04](#)
- [Buzzer 1](#)
- [Buzzer 2](#)

[Export to PDF](#)

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

Permanent link:  
[http://ocw.cs.pub.ro/courses/pm/prj2024/fstancu/maria\\_ionela.costea](http://ocw.cs.pub.ro/courses/pm/prj2024/fstancu/maria_ionela.costea)



Last update: **2024/05/27 00:55**