

Parking Sensor System - Banciu Horia-Adrian

Introducere

Proiectul constă în implementarea unui sistem de senzori de parcare. Șoferul va fi avertizat cu privire la eventualele obstacole din jurul mașinii atât vizual cât și acustic, evitând riscul unei coliziuni. Acesta este util în practică, deoarece sunt prevenite posibile coliziuni, este asigurată siguranța conducătorului auto și a altor persoane.

Descriere generală

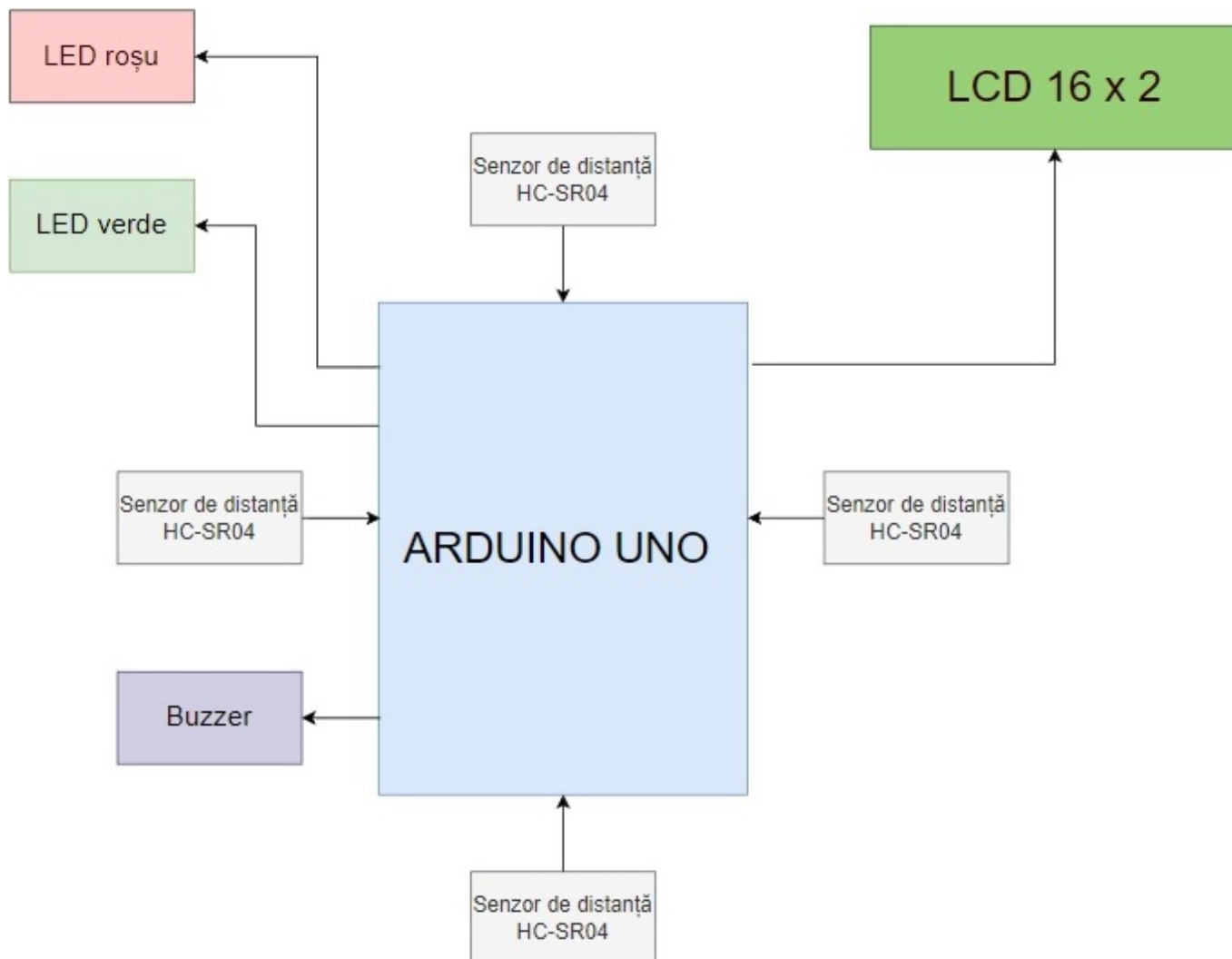
Senzorii de distanță se vor regăsi atât în fața și în spatele autoturismului, cât și în laterale. Sistemul va conține două LED-uri, unul verde și unul roșu: cel verde se va aprinde atunci când mașina se află la o distanță între 10 și 14 cm de obiect, iar cel roșu se va activa odată ce mașina ajunge la o distanță mai mică de 10 cm față de obiect. Pe măsură ce mașina se apropie de obstacol, LED-ul roșu își va modifica frecvența. De asemenea, conducătorul auto va fi avertizat acustic la apropierea de obiecte folosind un buzzer. Distanțele până la obstacole vor fi afișate pe un ecran LCD.

Componente

Pentru realizarea proiectului vor fi folosite următoarele piese:

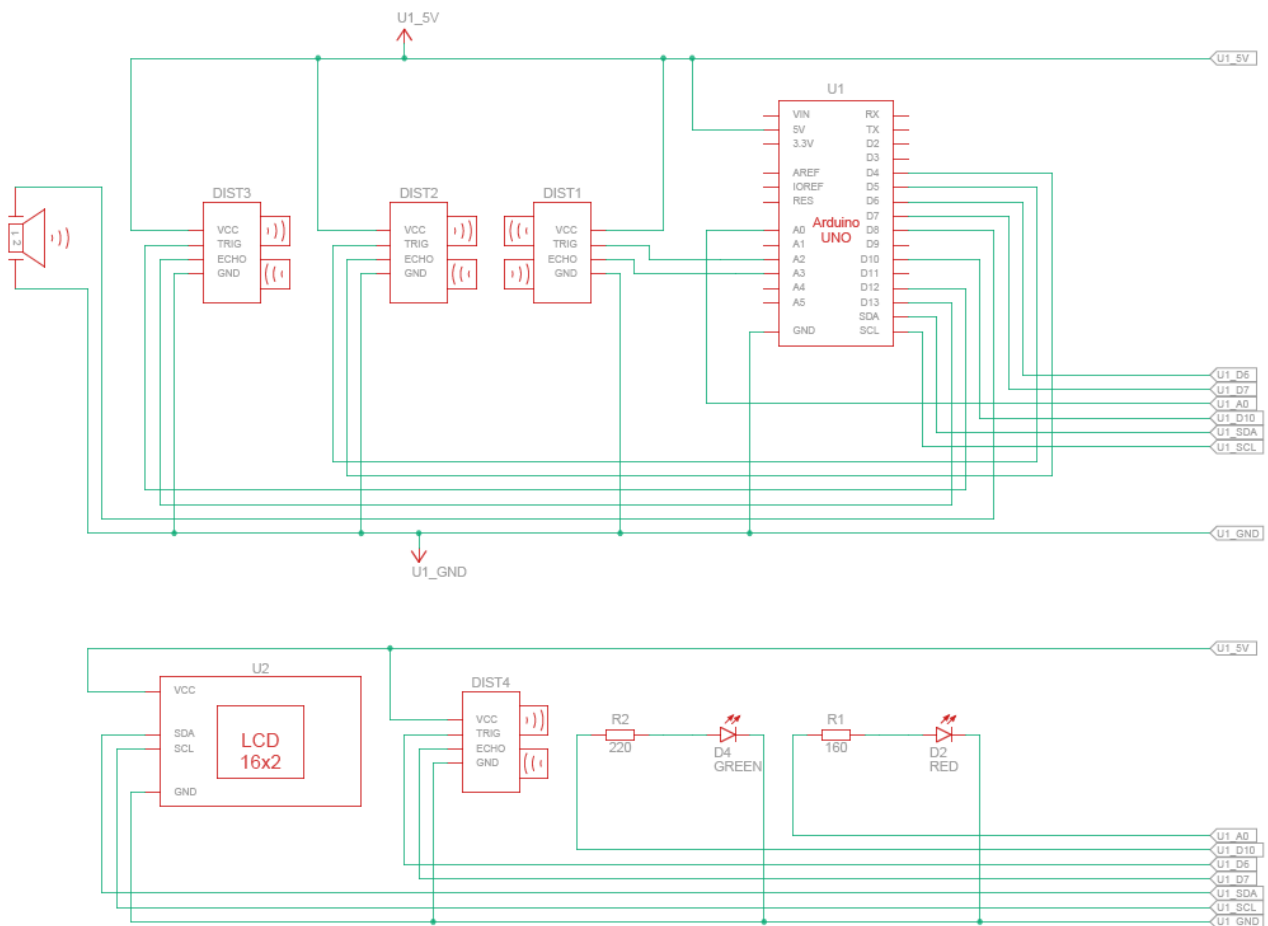
- Arduino UNO R3
- Breadboard
- LCD 16 x 2
- Senzori de distanță HC-SR04
- LED-uri
- Fire
- Rezistențe
- Buzzer

Schema bloc



[Export to PDF](#)

Hardware Design



Software Design

Mediul de dezvoltare folosit a fost Arduino IDE.

În implementarea proiectului am folosit următoarea bibliotecă:

- LiquidCrystal_I2C.h - pentru a afișa pe LCD distanțele de la mașină la obstacole.

În partea de inițializare au fost realizate următoarele operații:

- Setarea direcției pinilor pentru cele două LED-uri și pentru difuzor.
- Declararea unui vector conținând elemente de tip perechi (**distanță, valoare OCR1A**). Distanța calculată de senzori va fi comparată, pe rând, cu distanțele memorate în acest vector și se va actualiza valoarea registrului OCR1A. Pe măsură ce mașina se apropie de obstacole, valoarea din OCR1A va scădea, crescând astfel frecvența de aprindere a LED-urilor.
- Configurarea timer-ului Timer1: Odată ce este atinsă valoarea din OCR1A, se generează o **întrerupere** pentru modificarea stării LED-urilor.
- Pornirea ecranului LCD.

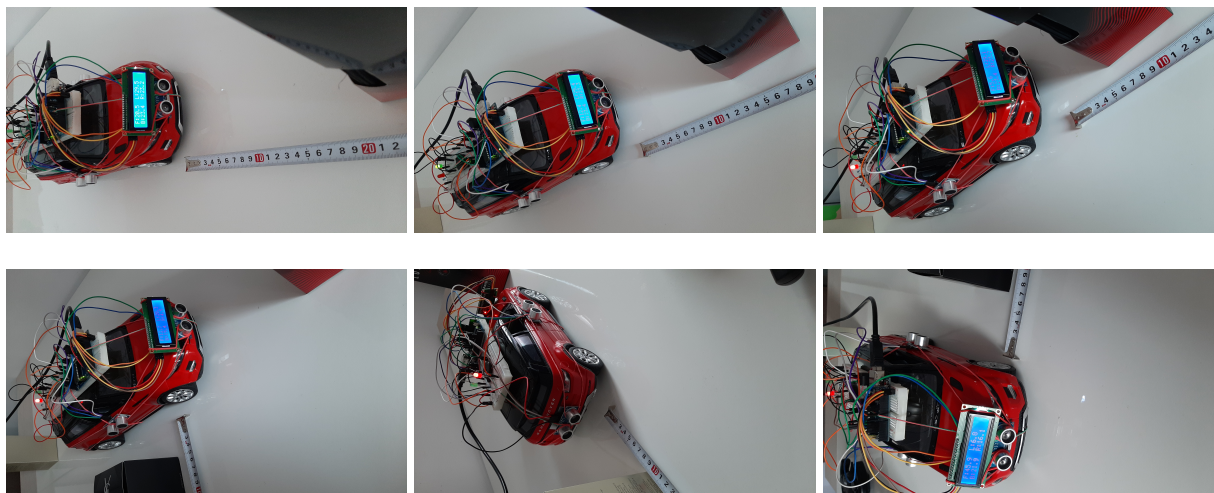
Implementarea conține următorii pași:

- Se calculează distanțele până la obstacolele din față, spatele și lateralul mașinii.
- Valorile calculate sunt afișate pe ecranul LCD.
- Dacă cel puțin una dintre distanțe este mai mică de 10 cm se va activa LED-ul roșu. Acesta își va modifica blink rate-ul pe măsură ce mașina se apropie și mai mult de obstacol prin actualizarea valorii registrului OCR1A. Aprinderea LED-ului verde corespunde unei distanțe cuprinse între 10 și 14 cm.
- Difuzorul va începe să funcționeze în timp ce LED-ul roșu este aprins, acesta producând sunete de intensitate mai mare odată cu apropierea autoturismului de obiecte.

Descrierea funcțiilor utilizate:

- **setup()** : Partea de inițializare a programului.
- **loop()** : Funcția în care citim valorile distanțelor, le afișăm pe ecran, activăm difuzorul.
- **initTimer()** : Configurează Timer1 în modul CTC, având prescaler-ul setat la 1024.
- **initStruct()** : Se adaugă într-un vector elemente de tip perechi de forma (distanță, valoare OCR1A). Am folosit acest vector în locul unor structuri alternative repetitive.
- **checkDistance(float dist)** : Returnează true dacă cel puțin una din cele 4 valori înregistrate de senzori este mai mică decât dist. În caz contrar, returnează false.
- **turnOnLed()** : Funcția este folosită **în contextul unei întreruperi**. Iterăm prin vector, iar la prima distanță de referință mai mare decât cel puțin una dintre valorile transmise de senzori modificăm registrul OCR1A și returnăm indexul poziției găsite. În caz contrar se returnează -1.
- **getDistance(int trigPin, int echoPin)** : Returnează distanța până la obiecte pentru senzorul care are pinii TRIG și ECHO egali cu cei trimiși ca parametru.

Rezultate Obținute



Demo: <https://www.youtube.com/watch?v=6EO87ohhu3I>

Concluzii

Proiectul are toate funcționalitățile pe care mi-am dorit să le implementez. Am lucrat cu plăcere la acesta și m-a ajutat să înțeleg mai bine lucrul cu timere și întreruperi.

Download

[parking_sensor_system_-_banciu_horia-adrian.zip](#)

Jurnal

02.05.2023 - Comandare piese

07.05.2023 - Realizare documentație inițială

14.05.2023 - Finalizare parte hardware

21.05.2023 - Finalizare parte software

Bibliografie/Resurse

https://ocw.cs.pub.ro/courses/_media/pm/atmel-7810-automotive-microcontrollers-atmega328p_datasheet.pdf - p.108 - 114

<https://www.electronicwings.com/sensors-modules/ultrasonic-module-hc-sr04>

<https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/>

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2023/fstancu/horia_adrian.banciu

Last update: **2023/05/29 15:55**



