

Student: Stefan Alexia

Grupa: 332CB

Garage Parking Sensor

Introducere

- **Tema** consta in proiectarea unui senzor de parcare, ce va determina aprinderea unor leduri si emiterea unui sunet, in functie de distanta fata de un obstacol, dar si afisarea in permanenta a distantei pe un display.
- **Scopul** proiectului este de a ajuta la aprecierea distantei de catre sofer cand parcheaza intr-un perimetru mic.
- **Ideea** care a condus la alegerea acestui proiect a fost gandul ca multe masini nu sunt echipate cu senzori/sisteme de parcare, dezavantaj major, mai ales pentru soferii incepatori, care nu au multa experienta.
- **Parerea mea** este ca un astfel de dispozitiv este **folositor** pentru evitarea micilor atingeri/zgarieturi in parcare a masinilor, ajutand, astfel, la pastrarea intr-o stare buna a caroseriei, dar si pentru orientarea in spatiu a soferului.

Descriere generală



Senzorul va transmite date la microcontroller. Acesta, apoi, va determina aprindere ledurilor corespunzatoare si pornirea buzzer-ului, in functie de distanta fata de obstacol. Cu cat distanta este mai mica, cu atat sunt aprinse mai multe leduri. Buzzer-ul este activat incepand cu o anumita distanta, pentru o avertizare suplimentara. Distanta este afisata in permanenta pe display.

Hardware Design

Lista piese:

- Arduino UNO
- Senzor Ultrasonic de distanta HC-SR04+
- LCD 1602 cu Interfata I2C
- Rezistori de 220 ohm

- Leduri
- Breadboard
- Jumper wires
- Buzzer

Kit:

- <https://www.aliexpress.com/item/1005005970564532.html?spm=a2g0o>
- <https://www.conexelectronic.ro/senzori-si-module-pentru-platforme-de-dezvoltare/144-MODUL-SENZORI-ULTRASUNETE-HC-SR04.html>

Schema circuit:**1) LCD**

Am optat sa afisez distanta de la senzor pe un ecran LCD folosind protocolul I2C.
Concept: I2C

- Pinul SCA → conectat la pinul analogic A4 de pe Arduino
- Pinul SCL → conectat la pinul analogic A5 de pe Arduino

2) Senzor ultrasonic de distanta HC-SR04+

Distanta o voi masura cu ajutorul unui senzor ultrasonic de distanta HC-SR04+.
Concept: GPIO

- Pinul GND → conectat pe breadboard la (-) si de acolo pe placuta Arduino la un GND
- Pinul Echo → conectat pe placuta arduino la pinul digital D4
- Pinul Trig → conectat pe placuta arduino la pinul digital D3
- Pinul VCC → conectat pe breadboard la (+) si de acolo pe placuta Arduino la un pic 5V

Observatii:

- Echo este pin de input, folosit pentru receptionarea semnalului reflectat
- Trig este pin de output, folosit pentru a transmite un puls ultrasonic

3) Buzzer

Cand o anumita distanta este intrecuta, se declanseaza buzzer-ul, ca un avertisment suplimentar pentru sofer.

Buzzer-ul este conectat astfel:

- (+) → conectat la pinul D2 de pe placuta Arduino
- (-) → conectat pe breadboard la (-) si de acolo pe placuta Arduino la un GND

4) Leduri

- 3 x Leduri Verzi: conectate la pinii digitali D13, D12, D11

- 3 x Leduri Galbene: conectate la pinii digitali D10, D9, D8
- 3 x Leduri Rosii: conectate la pinii digitali D7, D6, D5

Software Design

Mediu de dezvoltare:

- Arduino IDE: dezvoltarea codului si incarcarea lui pe Arduino
- Draw.io: realizarea schemei bloc <https://app.diagrams.net/>
- Thinkercad: realizarea schemei circuit <https://www.tinkercad.com/>

Cod:

- https://github.com/alexiastfn/ParkingAlertSystem_Arduino
- Biblioteci: Wire.h (pentru comunicarea I2C), LiquidCrystal_I2C.h (pentru display)

Implementare:

1. Inainte de setup():

- initializez lcd-ul cu adresa I2C a ecranului (0x27) si dimensiunea de 16x2
- definirea pinilor
- definirea variabilelor pentru afisarea pe display (duration, distance)
- definirea variabilelor pentru controlul buzzer-ului si al blink-ului de ledurilor rosii.

2. setup():

- setare pini pentru senzorul de distanta, buzzer si leduri.
- initializare lcd

3. loop():

- senzorul ultrasonic trimite un puls si masoara durata pentru a calcula distanta
- distanta este afisata pe display
- buzzer-ul este activat atunci cand distanta incepe sa fie mai mica de 14 cm: verific daca a trecut un interval "interval" pentru a da urmatorul beep & in functie de distanta buzzer-ul este setat la o frecventa (distanta cea mai mica - frecventa cea mai mare, distanta medie - frecventa medie, distanta cea mai mare - frecventa cea mai mica)
- cand distanta este cea mai mica, pe langa sunetul de buzzer voi face ca cele 3 leduri rosii sa blink-uiasca.
- astfel: led-urile rosii se aprind si blink-uiesc daca distanta este mai mică sau egală cu 4 cm, led-urile galbene se aprind pentru distante intre 15-21 cm, led-urile verzi se aprind pentru distante mai mari de 22 cm.

Concepte:

1. GPIO

- pinMode, digitalWrite, si digitalRead sunt functii folosite pentru a configura si a controla pinii GPIO
- Pinii trig, echo, buzzer, si LED-urile sunt controlati prin GPIO

2. I2C

- Biblioteca Wire si LiquidCrystal_I2C sunt folosite pentru a comunica cu display-ul LCD pe interfata I2C

3. Timere

- millis() este folosit pentru a masura intervale de timp fara a bloca executia codului (la controlul intervalelor de timp pentru buzzer si LED-uri)

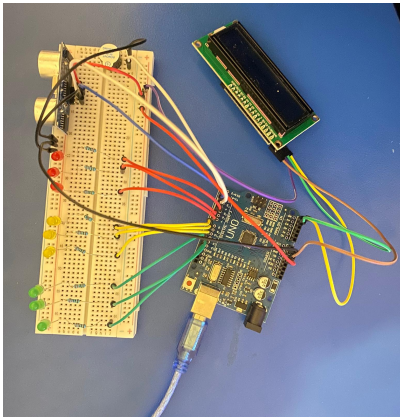
4. PWM

- Functiile tone(buzzer, frequency) si noTone(buzzer) sunt utilizate pentru a genera semnale PWM pe pinul buzzer pentru a produce sunete la frecvente diferite

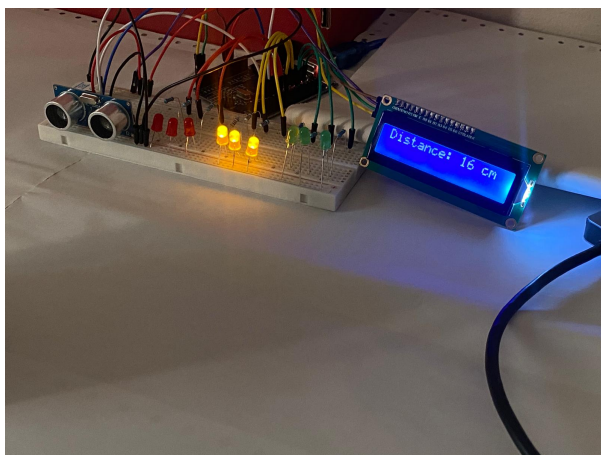
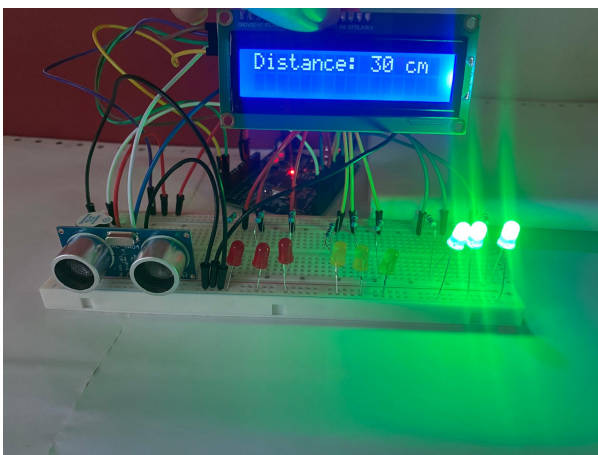
Rezultate Obținute

Demo youtube: <https://youtu.be/UWKqFVtKsrl?si=T8ebAcAZYVC9DqkY>

Faza de asamblare (15.05.2024):



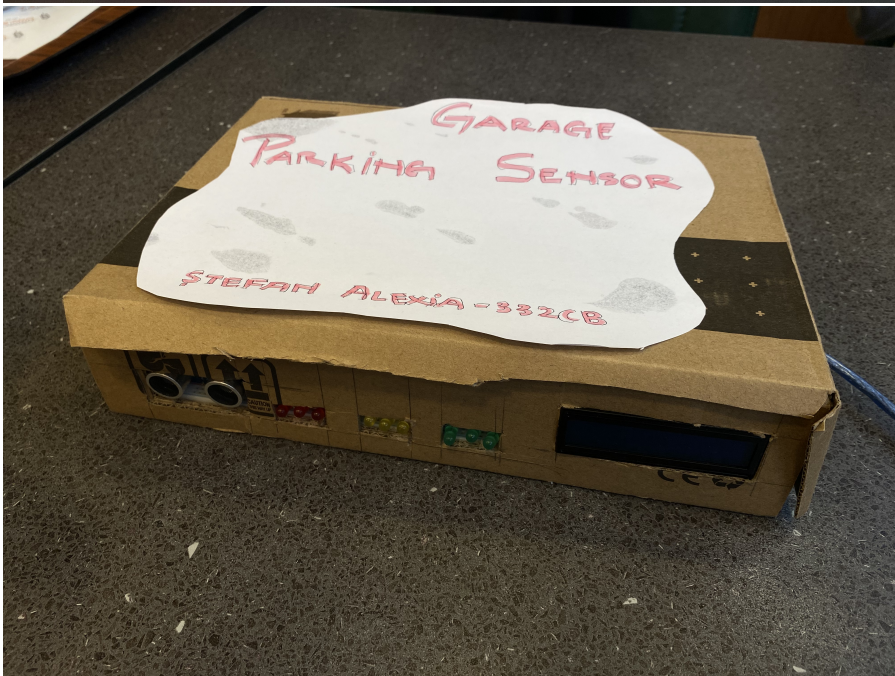
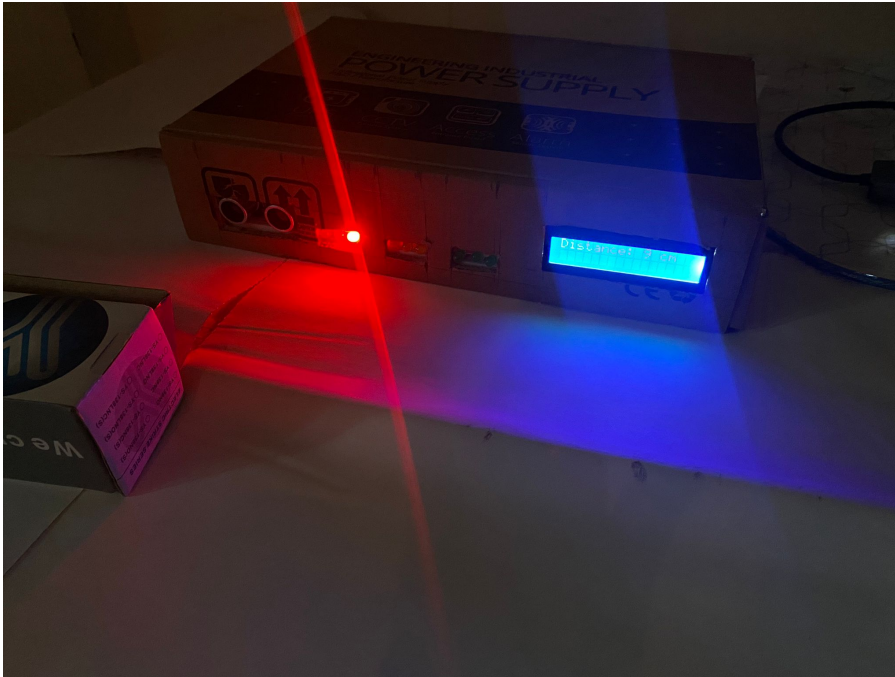
Finalizare cod:



Pentru cazul de apropiere mare:

- 1 x Led Rosu aprins: <https://youtu.be/3I7xgFHnDPQ>
- 2 x Led Rosu aprins: <https://youtu.be/yk7963xlwjE>
- 3 x Led Rosu aprins: <https://youtu.be/rZqBc7jjlic> (se poate observa comportamentul de blinking ale celor 3 led-uri)

Asamblare finala:



Concluzii

- Realizarea acestui proiect a fost o experienta distractiva
- A fost foarte util sa proiectez mai intai in Tinkercad, iar apoi sa asamblez
- Daca as putea sa iau proiectul de la capat, as opta pentru un senzor normal de distanta, nu cu ultrasunete
- Cea mai enervanta parte a fost cea de decupare a cutiei

Jurnal

Gantt Chart:



Tinte

Tinte de performanta:

1. Precizia masuratorilor de distanta: eroare maxim admisa de sub 1 cm
2. Viteza de raspuns: sub 1 sec
3. Stabilitatea afisarii distantei pe display: afisajul nu trebuie sa intampine nicio eroare

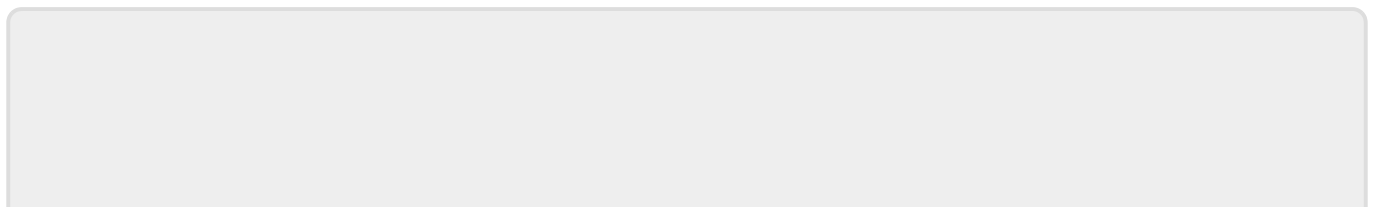
Download

Codul se poate descarca de pe: https://github.com/alexiastrfn/ParkingAlertSystem_Arduino

Bibliografie/Resurse

- <https://www.youtube.com/playlist?list=PLGs0Vkk2DiYw-L-RibttcvK-WBZm8WLEP>
- <https://www.youtube.com/watch?v=nL34zDTPkcs>
- <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>
- <https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/>

[Export to PDF](#)



From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/ccontasel/alexia.stefan>



Last update: **2024/05/27 10:13**