

# EZpark

## Introducere

Proiectul constă în crearea unui ansamblu de senzori de proximitate ce pot ghida o mașină în locuri fără vizibilitate. De aceea, voi pune un senzor în față și unul în spatele mașinii. În momentul când detectează un obiect în apropiere, un led RGB va semnaliza situația, un buzzer va reda cât de aproape e pericolul ajustând frecvența, iar pe un ecran LCD va fi reprezentată zona și distanța (un fel de HUD). Pentru a face proiectul mai interesant voi atașa senzorii și plăcuța pe o macheta de mașina.

## Descriere generală

Cu ajutorul a doi senzori de proximitate, o mașină va putea “vedea” obiectele din calea sa. Cei doi senzori vor returna distanța față de obstacole, iar aceasta va fi prelucrată astfel încât:

- un buzzer va suna, dacă un obiect este periculos de aproape, iar intensitatea sunetului va fi în funcție de gradul de apropiere
- un led RGB va fi verde dacă mașina are cale liberă, galben dacă a sesizat un eventual pericol și roșu dacă accidentul e iminent
- un ecran LCD măricel sau două ecrane pentru fiecare senzor de proximitate, unde pot reprezenta mașina și obiectele în funcție de distanță

### Schema bloc:

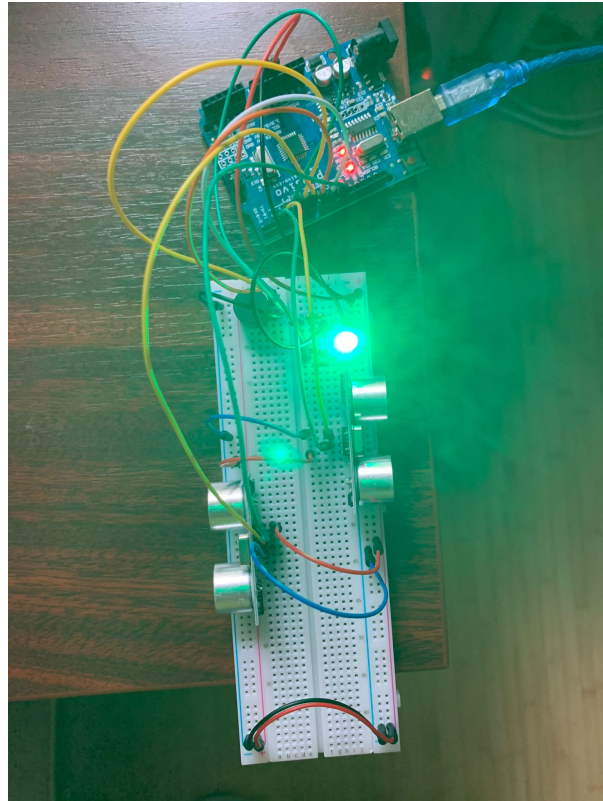
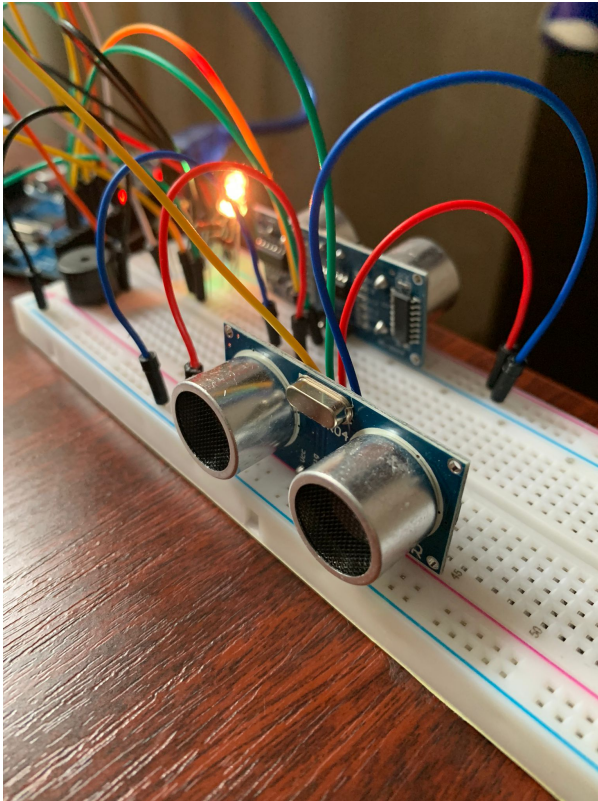


## Hardware Design

### Listă piese:

- Arduino Uno x 1
- Breadboard x 1/2?
- Led RGB x 1
- Buzzer x 1
- Senzor de proximitate HC-SR04 x 2
- Ecran LCD mare x 1 sau mici x 2
- Rezistente
- Fire

### Schema electrica:



## Software Design

### Mediu dezvoltare:

- Arduino IDE

### Librării și surse 3rd-party:

- Ambele librării sunt folosite pentru ecranele lcd
- <LiquidCrystal.h>
- <Adafruit\_LiquidCrystal.h>

### Descriere sursa și funcții implementate:

- La început am definit pentru buzzer, led\_rgb și ecranele lcd
- In setup() am setat pinii de output și am initializat interfata LCD.
- In loop() apelez funcția distSensor(). De aici codul meu se împarte în mai multe funcții.
- In distSensor(), pentru început calculez distanța detectată de fiecare din cei doi senzori de proximitate cu ajutorul funcției readUltrasonicDistance(), apoi urmează 4 condiții în funcție de aceste distanțe realizate cu ajutorul if else. Condițiile implică nivelul de amenințare detectat de senzorul de distanță. Astfel, dacă una din distanțe e mai mică de 7 cm programul va aprinde ledul rgb pe culoarea roșie, buzzerul va suna foarte ascuțit și des, iar pe ecran se va afișa DANGER IMINENT! și o reprezentare a distanței cu linii pentru o înțelegere mai ușoară. Analog, dacă distanțele sunt între 7 și 15 cm ledul va fi portocaliu, buzzerul va suna mai rar și cu o tonalitate mai joasă, iar pe ecran va scrie MEDIUM DANGER! și vor fi afișate mai multe linii decât în cazul precedent.

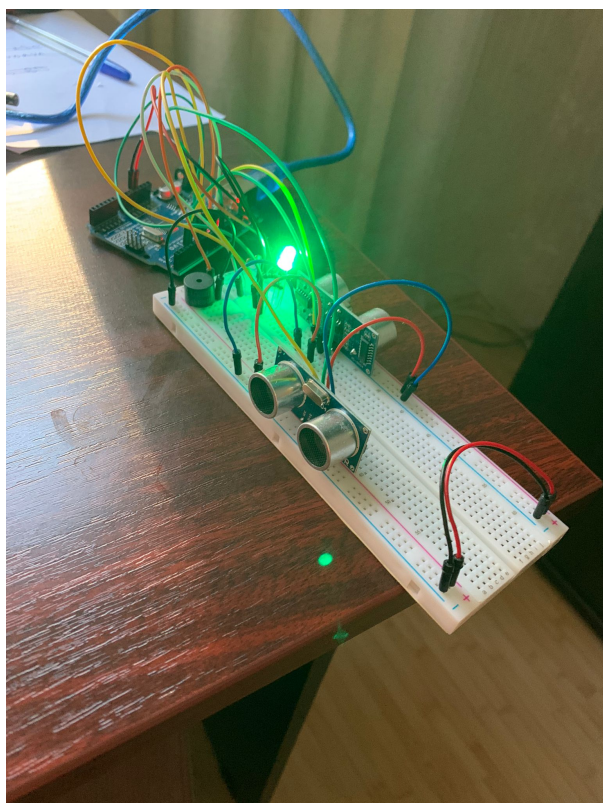
Daca distantele sunt intre 15 si 25 de cm ledul va fi galben, buzzerul va suna rar si jos, iar lcd-ul va afisa CAUTION DANGER!. Altfel, daca distantele sunt mai mari de 30 de cm va ledul va fi verde, buzzerul nu va suna, iar pe ecran nu se va afisa nimic. Daca distantele nu intra pe niciuna din conditiile de mai sus ledul va fi mov. Pentru a aprinde ledul o anume culoare vom folosi functia RGB\_color(). Pentru a porni buzzerul si a modifica frecventa si amplitudinea sunetului vom folosi functia tone() inclusa in biblioteca Arduino. Pentru a afisa pe ecranele lcd vom apela functiile displayBack() si displayFront().

- Functia readUltrasonicDistance() primeste ca parametri triggerPin si echoPin de la senzorul ultrasonic, pune pinul trigger pe HIGH timp de 10 ms, iar apoi citeste de pe pinul echo si returneaza timpul in ms al undei de sunet pana cand intalneste un obiect.
- Functia RGB\_color() primeste 3 parametrii R, G, B si scrie aceste valori pe pinii ledului RGB aprinzandu-l in diferite culori.
- Functiile displayBack() si displayFront() afiseaza diferite mesaje pe ecranele corespunzatoare senzorului de proximitate din fata, respectiv cel din spate.

}

## Rezultate Obținute

- Cand nu exista obstacol ledul ramane verde:



- Atunci cand detecteaza un obstacol la o distanta relativ sigura ledul va fi galben:

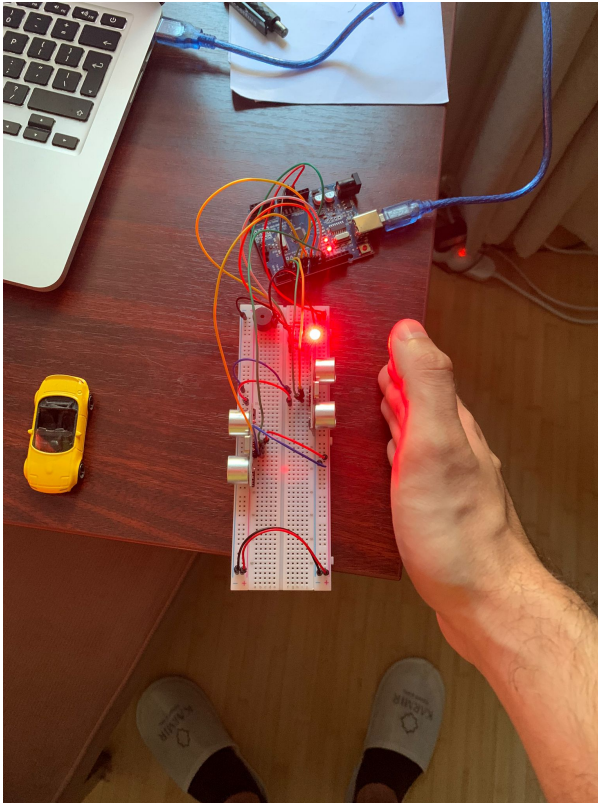


- La o distanta medie, ledul devine portocaliu:

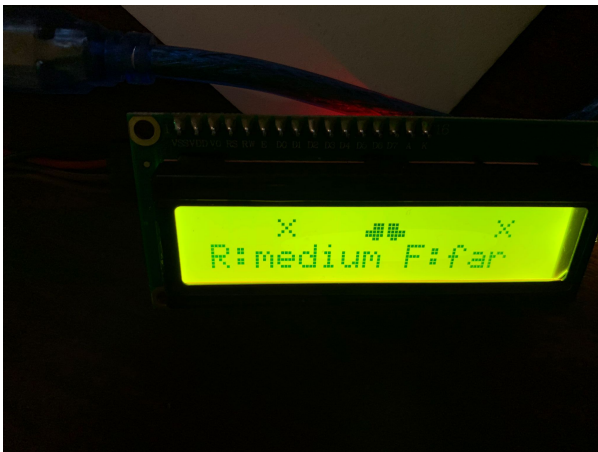


- Cand un obstacol e foarte aproape ledul devine rosu:

De mentionat faptul ca programul prioritizeaza severitatea apropierii, astfel incat daca un obiect e la o distanta medie, iar altul foarte aproape, ledul va semnala obiectul cel mai aproape.



- LCD-ul afiseaza un hud cu apropierea obstacolelor fata de masina:



- Video: [https://www.youtube.com/watch?v=\\_KWYNKqpjdQ](https://www.youtube.com/watch?v=_KWYNKqpjdQ)

## Concluzii

- A fost printre cele mai interesante proiecte pe care le-am facut
- Am cumparat 3 ecrane in total deoarece 2 ecrane lcd 20x4 nu functionau cand le conectam, asa ca am luat inca unul I2C
- A fost foarte util sa fac proiectul in Tinkercad, iar apoi doar sa-l asamblez
- Am ars un led RGB
- Am invatat ca la astfel de proiecte e foarte util sa ai piese de rezerva
- Nu exista niciun magazin fizic de unde sa iei componentele in Bucuresti, asa ca de fiecare data cand iti trebuie ceva e nevoie sa dai o comanda pe internet

## Download

- [archive\\_ezpark.zip](#)

[Export to PDF](#)

## Jurnal

- 21.04.2022 - Alegere tema proiect, descriere proiect
- 05.05.2022 - Comparare componente
- 20.05.2022 - Finalizare schita proiect in Tinkercad
- 22.05.2022 - Asamblare proiect, e nevoie de un nou ecran lcd
- 27.05.2022 - Documentatie finala
- 28.05.2022 - Nou ecran lcd

## Bibliografie/Resurse

- <https://www.allaboutcircuits.com/tools/resistor-color-code-calculator/>
- <https://www.tinkercad.com/things/kvo7nINsMf5-copy-of-park-rgb/editel?tenant=circuits>
- <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
- <https://melihkeskinn.wordpress.com/2017/07/18/connecting-a-1602a-lcd-display-and-a-light-sensor-to-arduino-uno/#:~:text=First%20you%20need%20to%20power%20the%20display%20and,backlight%20power%20to%20the%20plus-%20and%20min%20rows>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/imacovei/ezpark>



Last update: **2022/05/29 19:46**