

Asistent de parcare

Autor: Grigore Lucian-Florin 336CA

Introducere

Senzor de proximitate montat pe peretele garajului care ajuta soferul, in momentul parcarii, sa stie cat de aproape este de un anumit perete, prin schimbarea culorii afisate si a frecventei de afisare, precum si printr-o avertizare sonora.

Ideea a pornit de la dorinta de a ii ajuta pe cei care isi parcheaza masina in locuri stramte.

Descriere generala

Modulul Arduino primeste pe RX un vector de distante de la senzorul ultrasonic, calculeaza o distanta medie ca sa elimine putin din zgomotul transmis de acesta. Apoi, lumineaza banda LED, sau LED-urile individuale, si porneste buzzer-ul corespunzator distantei dintre masina si senzor. Cu cat masina se apropie mai mult de senzor, lumina devine mai rosie si buzzer-ul se audie din ce in ce mai des.

Schema bloc

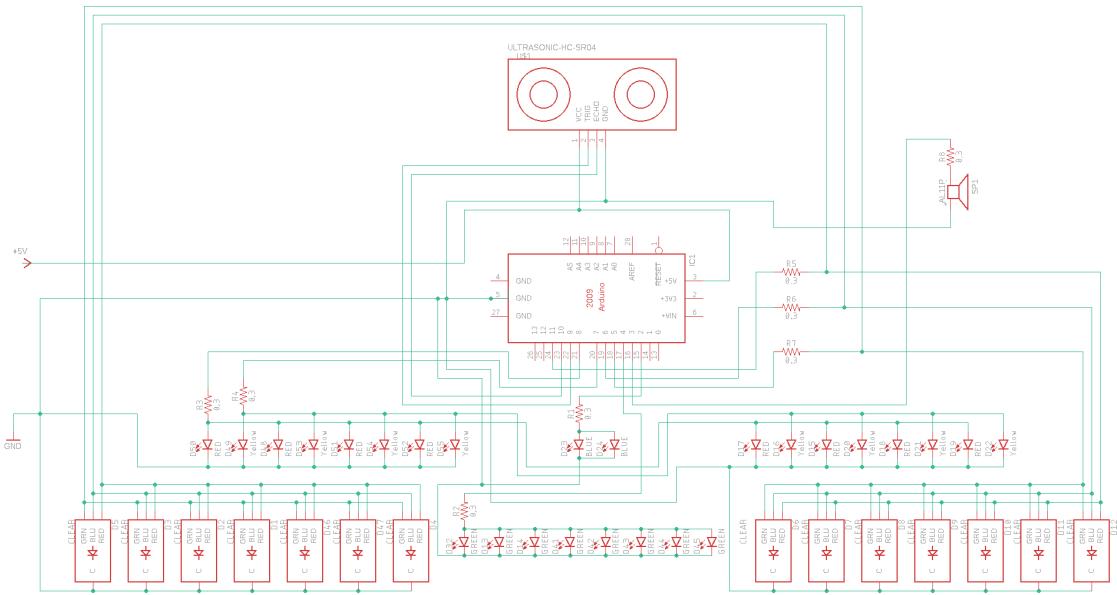


Hardware design

Piese necesare

- Arduino Uno/Nano
- HC-SR04 senzor ultrasonic
- Condensator 1000uF (se pot si mai mici, dar puse in paralel pentru a obtine aceeasi impedanta)
- Rezistor 470 ohm si 330 ohm (se poate folosi si doar un singur fel)
- WS2812B LED strip sau LED-uri individuale (RGB sau single color)
- Dioda 1N4148 (optionala, doar pentru alimentarea in acelasi timp si a placutei Arduino, si a benzii, de la priza)
- Buzzer/difuzor (am folosit buzzer de 3-24V)

Schema electrica cu LED-uri individuale



Schema electrica cu banda LED



Software design

Pentru unelte de analiza a datelor primite de la senzorul HC-SR04, am folosit libraria QuickStats. Aceasta nu este gasita in meniul "Manage libraries" din Arduino IDE, asa ca trebuie descarcata ca arhiva si trebuie folosita optiunea "Add a .ZIP library". Pentru controlul benzii LED am folosit libraria FastLED. Codul pentru Uno R3 a fost scris de mine in Arduino IDE. Codul de Nano este compatibil cu Uno R3, deoarece porturile folosite au aceleasi capabilitati (PWM).

LED strip

Daca se utilizeaza o banda WS2812B, atunci libraria FastLED contine functii precum show() si CRGB(). Cu cea din urma se seteaza culoarea intr-un vector de tipul CRGB, care reprezinta led-urile fizice, iar cu cea dintai se afiseaza noua configuratie a LED-urilor.

Individual LEDs

Folosind LED-uri individuale, se folosesc functiile analogWrite() (datorita necesitatii PWM pentru cele 3 canale de culoare) pentru RGB si digitalWrite() pentru LED-uri single color. Am creat functii care sa afiseze orice spectru de culori pe LED-urile RGB, iar pentru LED-urile single color am creat un numar fix de configuratii posibile de lumini, care sunt folosite in functie de cat de apropiata este masina de senzor.

Arduino Code

Codul Arduino folosit (pentru Uno R3) se poate gasi [aici](#).

Codul Arduino (pentru Nano) care se poate folosi impreuna cu o banda LED WS2812B se poate gasi [aici](#).

Rezultate obtinute

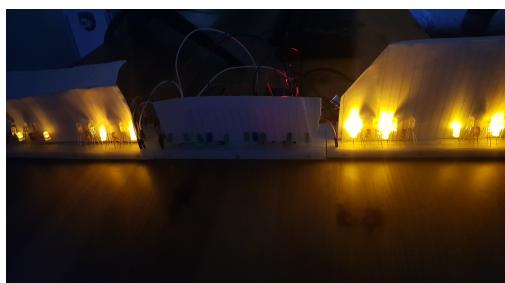
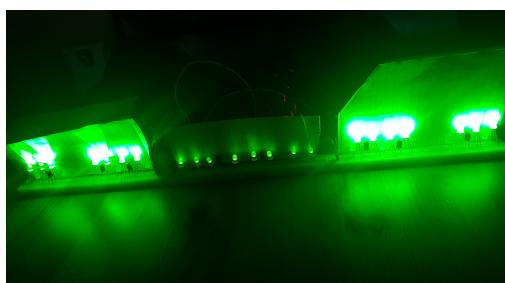
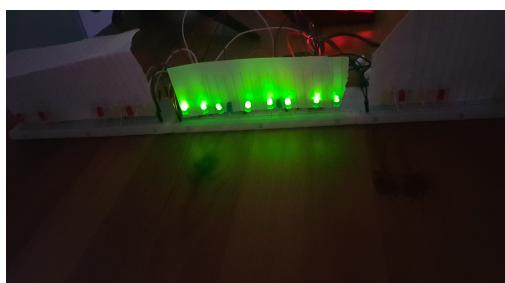
Functionare

Distantele configurate pentru demonstratie au fost destul de mari si la intervale mari, ceea ce poate fi destul de dificil pentru senzorul ultrasonic. Acesta se comporta mult mai bine la distante mai mici, unde precizia lui creste substantial. Astfel, intr-un scenariu real de utilizare, se recomanda configurarea pentru distante mici, unde se va obtine o performanta excelanta, dar si o consistenta mai solida.

Senzorul va avertiza (vizual si sonor) in urmatorul fel (in cadrul demo-ului):

- peste 300cm - lumina albastra si nicio avertizare sonora
- intre 250cm si 300cm - lumina verde slabă si usoara avertizare sonora
- intre 200cm si 250cm - lumina verde intensa si avertizare sonora mai alerta
- intre 150cm si 200cm - lumina galbena intensa si aceeasi avertizare sonora
- intre 0cm si 150cm - lumina rosie intensa si avertizare sonora foarte alarmanta

Poze cu starile descrise mai sus





Demo video

<https://www.youtube.com/watch?v=NoFG4ZSbEao>

Filmarea este realizata cu telefonul, asa ca am ales sa filmez seara, ca sa nu intre prea multa lumina naturala in cadrul, si cu farurile stinse la masina. De aceea, calitatea nu este iesita din comun, iar din cauza faptului ca este spatiu inchis, microfonul telefonului nu prinde asa bine sunetul buzzer-ului, desi si acesta poate fi auzit, in fundal. Cu toate acestea, functionalitatea asistentului de parcare este usor demonstrata prin acest demo.

Concluzii

Asistentul de parcare s-a dovedit a fi foarte eficient. Dat fiind faptul ca este vorba de detectarea unei masini, senzorul se descurca foarte bine cu obiectele mari, fiind convenabil de precis. Buzzer-ul folosit, chiar daca este mic, se aude foarte clar, chiar si din interiorul masinii, cu motorul pornit si geamurile inchise, iar luminile pot fi usor vazute, chiar si ziua, sau cu farurile pornite.

Download

[grigorelucianflorin_336ca_pm_project.zip](#)

Jurnal

- 24 mai 2021 - inceput proiect
- 29 mai 2021 - terminare design software si asamblare componente, cu exceptia benzii LED
- 31 mai 2021 - am ajuns la concluzia ca banda LED nu merge (probabil am ars-o), dupa ce am incercat sa o conectez in 100 de feluri
- 1 iunie 2021 - modificare proiect astfel incat sa folosesc LED-uri individuale
- 2 iunie 2021 - terminare montaj, filmare demo seara
- 4 iunie 2021 - terminare documentatie completa

Bibliografie/resurse

Resurse software

https://create.arduino.cc/projecthub/Bcjams/garage-parking-assistant-11446b?ref=search&ref_id=garage&offset=0

<http://fastled.io/>

<https://playground.arduino.cc/Main/QuickStats/>

<https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/>

<https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>

Resurse hardware

<https://howtomechatronics.com/tutorials/arduino/how-to-control-ws2812b-individually-addressable-leds-using-arduino/>

[Export to PDF](#)

Kudos to Bcjams for the idea of this type of project :)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/asistent_parcare

Last update: **2021/06/04 18:31**

