

Interactive speed radar

Introducere

Un dispozitiv care determina viteza unui obiect și o afișează pe un LCD, alături de alte mesaje sugestive, în funcție de setări. Utilizatorul poate selecta afișarea vitezei în km/h, mph sau m/s, poate seta limita de viteză și modul de afișare al mesajelor. În funcție de limita de viteză introdusă, se afișează diferite mesaje de informare însoțite de colorarea corespunzătoare a unei benzi LED, și chiar suma amenzii de viteză dacă lasăm gândurile intruzive să castige.

P.S: Este posibil ca valoarea amenzii să nu mai fie corectă pe viitor.

Descriere generală



Viteza obiectului este măsurată de doi senzori infraroșu, amplasați la o distanță cunoscută. Se măsoară intervalul de timp necesar activării ambilor senzori. Se împarte distanța la timp și se obține viteza v .

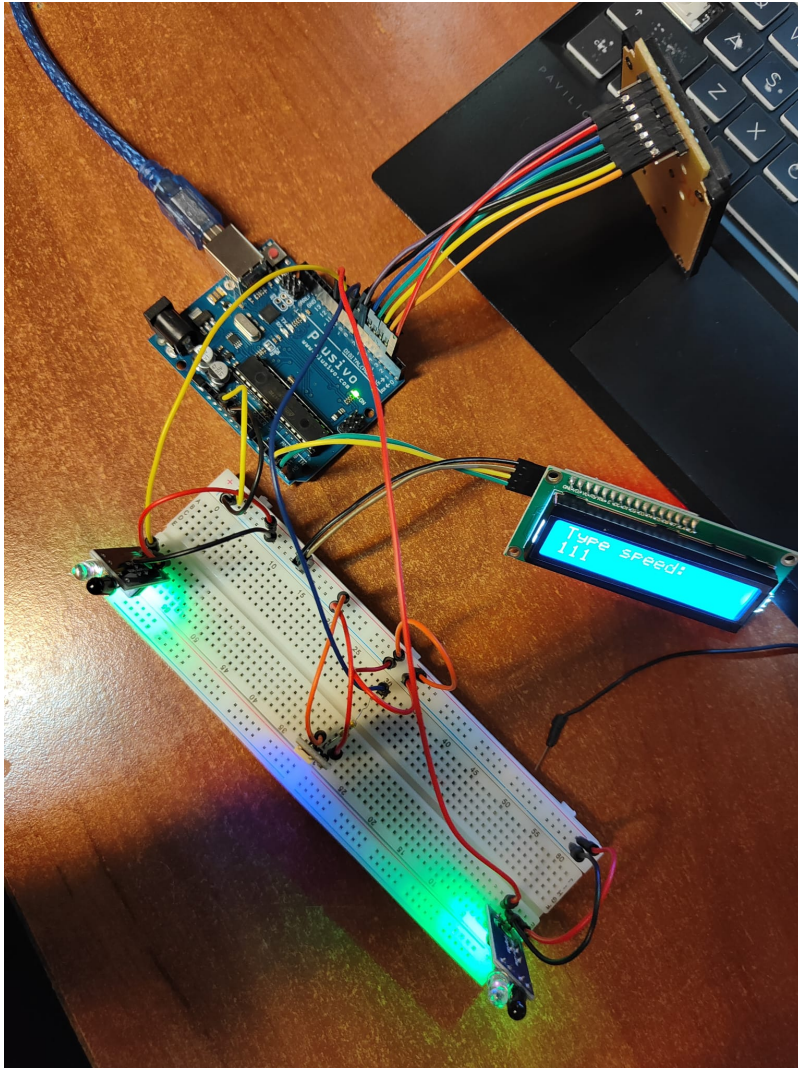
Prin intermediul unei tastaturi numerice, utilizatorul poate să modifice comportamentul radarului de viteză, astfel:

- poate modifica limita de viteză (pentru momentele când prinde Autobahn-ul liber)
- poate schimba modul în care se afișează mesajele informative (standard sau cu referințe de Formula 1)
- poate schimba unitatea de măsură a vitezei: km/h, m/s sau freedom units (mph)

Hardware Design

Componente folosite:

- Arduino Uno R3
- 2 x senzor infraroșu
- Ecran LCD
- Modul tastatură 4×3
- Banda LED RGB
- Breadboard
- Rezistență 470 ohm
- Fire de legătură



Modulul de keypad folosit in simulare nu este cel pe care il folosesc in proiectul meu, deoarece nu am reusit sa il gasesc.

Cu toate acestea, am incercat sa pastrez legaturile intre modulul din simulari si placuta cat mai aproape de realitate

Software Design

Pentru partea de software, am folosit bibliotecile Keypad pentru a citi caracterele de la keypad, LiquidCrystal_I2C pentru controlul display-ului LCD si Adafruit Neopixel pentru controlul benzii LED;

Partea centrala a codului este reprezentata de partea de calcul al vitezei, realizat prin calculul raportului intre distanta predefinita dintre senzorii infrarosii si timpul necesar declansarii ambilor senzori. Sunt facute verificari suplimentare pentru a oferi un calcul cat mai aproape de realitate al vitezei, precum verificarea printr-un flag ca ambii senzori au fost activati inainte de a trece la calculul vitezei.

In functie de viteza limita setata de utilizator si viteza calculata, se acorda o evaluare a vitezei si se afiseaza mesajele corespunzatoare, astfel:

- $viteza < 0.75 * limita_viteza \Rightarrow viteza \text{ prea mica}$
- $viteza \geq 0.75 * limita_viteza \ \&\& \ viteza < 1.05 * limita_viteza \Rightarrow viteza \text{ normala}$
- $viteza \geq 1.05 * limita_viteza \ \&\& \ viteza < 1.5 * limita_viteza \Rightarrow viteza \text{ mare; se va calcula valoarea amenzii in functie de viteza}$
- $viteza > 1.5 * limita_viteza \Rightarrow viteza \text{ excesiva; vine politia; se va suspenda permisul;}$

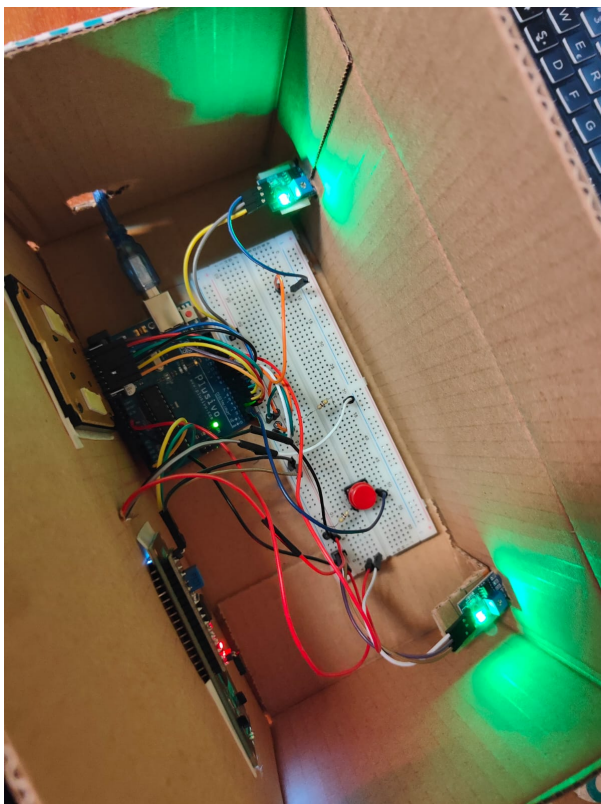
De asemenea, proiectul mai contine si un meniu de configurare, in care, prin apasarea unor taste, se pot schimba setarile dispozitivului:

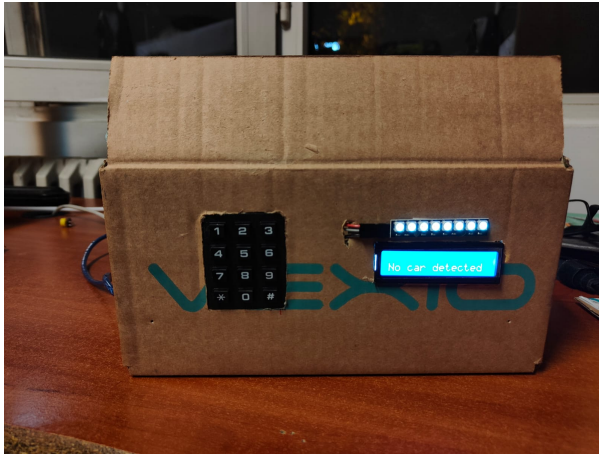
- limita de viteza
- unitatea de masura a vitezei
- modul de afisare al mesajelor informative

Comutarea intre modul de configurare si cel de detectare a vitezei se poate face prin apasarea unui buton care genereaza o intrerupere ce schimba un flag in cod.

Codul contine cateva functii de animare a benzii LED, in functie de gravitatea faptei.

Rezultate Obținute





Am obtinut un dispozitiv care masoara viteza si o afiseaza in diferite unitati de masura, insotita de mesaje de afisare diverse si diferite animatii de LED-uri. De asemenea, dispozitivul poate fi configurat dupa bunul plac.

Radarul foloseste tastatura pentru a citi diferite comenzi de la utilizator sau pentru a actualiza parametrii de rulare.

Concluzii

Per total, a fost un proiect engaging, care a combinat cunostintele software cu cele hardware si m-a pus in pozitia sa imi "murdaresc mainile", avand ocazia sa ma joc cu componentele hardware. Desi

exista multe locuri unde ar putea fi imbunatatit, sunt multumit de proiect si m-am simtit bine lucrând la el.

Download

[source_code_pana_sergiu_335ca.zip](#)

Jurnal

- 4 aprilie - prima comanda de componente
- 20 aprilie - a doua comanda de componente
- 28 aprilie - idee + documentatie
- 20 mai - hardware partial gata (componentele legate intre ele, fara cutie)
- 24 mai - software partial gata (parte din functionalitati implementate)
- 28 mai - proiect gata - hardware: aranjat componentele in cutie; software: implementat restul functiilor si facut legaturile

Bibliografie/Resurse

Biblioteci folosite:

- [Keypad](#)
- [Adafruit Neopixel](#)
- [LiquidCrystal_I2C](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2023/alucaci/interactive-speed-radar>



Last update: **2023/05/29 10:43**