Traffic Racer Game

Autor: Craciun Andrada-Sinziana 334CA

Introducere

Despre proiect

Proiectul consta in realizarea unui joc in care o masina trebuie sa evite obstacolele aparute in drum, folosind un ecran LCD I2C. Initial masina are un numar de vieti pe care le poate pierde pe parcursul jocului.

Scopul proiectului

Scopul proiectului este familiarizarea cu Arduino și perifericele necesare pentru implementarea jocului, cat si scrierea unui cod ce controlează direct componente hardware. Consider că reprezintă o metodă bună de a pune în aplicare toate noțiunile dobândite, atât software cât și hardware, pentru a realiza un joc de la 0. Ideea de la care am pornit este realizarea unui joc de obstacole in genul Flappy Bird pe care l-am adaptat dupa propriile idei.

Exista mai multe tipuri de obstacole, iar in functie de tipul lor, unele pot decrementa numarul de vieti ale masinii, iar altele il pot incrementa.

In coltul drept al ecranului se vor afisa numarul de vieti ale jucatorului si scorul adunat pe parcursul jocului.

Descriere generală

Funcționalitate

Plăcuța Arduino va prelua datele de intrare cu ajutorul unui joystick pe care jucatorul il va folosi pentru mișcarea piesei. In momentul in care masina loveste un obstacol care ii decrementeaza numarul de vieti, buzzer-ul va scoate un sunet, iar ledul se va lumina rosu. Daca masina loveste un obstacol special, ce ii incrementeaza numarul de vieti, ledul se va aprinde verde, iar buzzer-ul va scoate un sunet diferit. Întregul joc va fi afișat pe un ecran LCD I2C.

Jocul incepe prin afisarea pe ecran a mesajului: **Move joystick up**.

Cand masina nu mai are nicio viata, jocul se termina iar pe ecran se va afisa mesajul: **Game over**, iar buzzer-ul va scoate un sunet diferit de cele de dinainte.

Obstacolele speciale apar mult mai rar decat obstacolele obisnuite si sunt reprezentate sub forma H (HealthPack) pe ecran, iar atunci cand apare o coliziune cu acestea, led-ul se va colora verde, iar buzzer-ul va scoate un sunet diferit.

Laptopul personal va asigura prin cablul USB tensiunea necesară funcționării plăcuței Arduino Uno (ATMega328p). Aceasta va alimenta la rândul său ecranul LCD (la 5V), cât și joystickul(la 5V). Plăcuța Arduino va comunica cu ecranul, iar joystickul va transmite acțiunile utilizatorului prin intermediul a 3 pini (VRx si VRy trimit catre pinii analogici A0 si A1, iar SW catre pinul digital D2). Buzzerul va acționa prin intermediul unui pin digital de pe plăcuța Arduino (D8).

Utilizatorul va folosi joystickul pentru a se mișca sus-jos. În diverse situații (exp. coliziunea cu un obstacol) buzzerul va emite sunete specifice. Jucătorul trebuie să încerce să evite obstacolele pentru a ramane în viață (să nu piardă toate viețile pe care le are la dispoziție).

Schemă bloc

Schema bloc a fost realizata in Draw.io.

Functionalitatea Modulelor

- Arduino Uno: logica jocului este implementata in microprocesorul ATmega328p
- Input: un joystick conectat la Arduino prin 3 pini
- Output: un ecran LCD I2C, un buzzer conectat la Arduino printr-un pin digital, un led RGB cu catod comun conectat la 3 pini digitali(R:D11, G:D10, B:D9)

Hardware Design

Lista de piese

Piesele folosite pentru implementarea jocului:

- Arduino UNO (ATMega328p)
- Joystick
- Buzzer pasiv
- Ecran LCD I2C
- Led RGB
- Breadboard
- 3 rezistori de 470 ohm
- Fire de lagatura mama-tata, tata-tata

Schema Hardware

Schema electrica

Conexiuni propriu-zise:

Ecranul LCD 1602 a fost conectat prin I2C la pinii:

- SCL si SDA conectati la pinii analogici A4 si A5
- VCC conectat la VCC-ul de la arduino
- GND conectat GND-ul Arduino

Joystick-ul:

- VRx si VRy conectati la pinii analogici A0 si A1
- SW la pinul digital D2
- VCC conectat la VCC-ul de la arduino
- GND conectat GND-ul Arduino

LED-ul RGB:

- conectat cu 3 rezistențe de 470 ohm pentru a asigura o functionare corecta
- componenta R a fost conectata la pinul digital D11
- componenta G a fost conectata la pinul digital D10
- componenta B a fost conectata la pinul digital D9
- GND conectat GND-ul Arduino

Buzzer-ul:

- GND conectat GND-ul Arduino
- partea pozitiva conectat la pinul digital D8

Software Design

Mediul de dezvoltare folosit:

- *Draw.io:* pentru realizarea schemei bloc
- *Fritzing:* pentru realivarea design-ului fizic
- *Fritzing:* pentru proiectarea schemei electrice
- *Arduino Integrated Development Environment (IDE):* pentru incarcarea codului pe placuta si debugging

Biblioteci folosite:

- Pentru interacțiunea cu LCD-ul, am descărcat libraria: *LiquidCrystal_I2C.* https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library

Funcții folosite:

Fisierul Car Game:

- *generateTerrain()* - functie genereaza obstacolele
- *moveCar()* - functie care misca masina pe ecran si deseneaza scena
- *shiftTerrain()* - functie care shifteaza atat planul superior cat si cel inferior (de pe ecran) cu o pozitie la stanga
- *RGB_LED()* - functie care aprinde LED ul cu culorile date ca parametru
- *beep()* - functie care activeaza buzzerul
- *setup()* - functie care este apelata la inceputul programului pentru a initializa pinii si ecranul LCD
- *loop()* - functie care se apeleaza constant pe parcursul programului

Fisierul Generate_Ghraphics:

- *generateGraphics()* - functie care genereaza animatiile si grafica
- *drawScene ()* - functie care deseneaza pe ecranul LCD scena
- *generateNewBlock()* - functie care genereaza un bloc nou pe ecran

Alte observații:

Odata inceput jocul, se genereaza scena si masina incepe sa se miste pe ecran, scorul crescand constant.

Miscarea masinii este simulata cu ajutorul functiei shiftTerrain(), care muta cu un spatiu la stanga toata scena. Scena este compusa din 2 planuri: superior si inferior, care reprezinta practic 2 vectori de caractere.

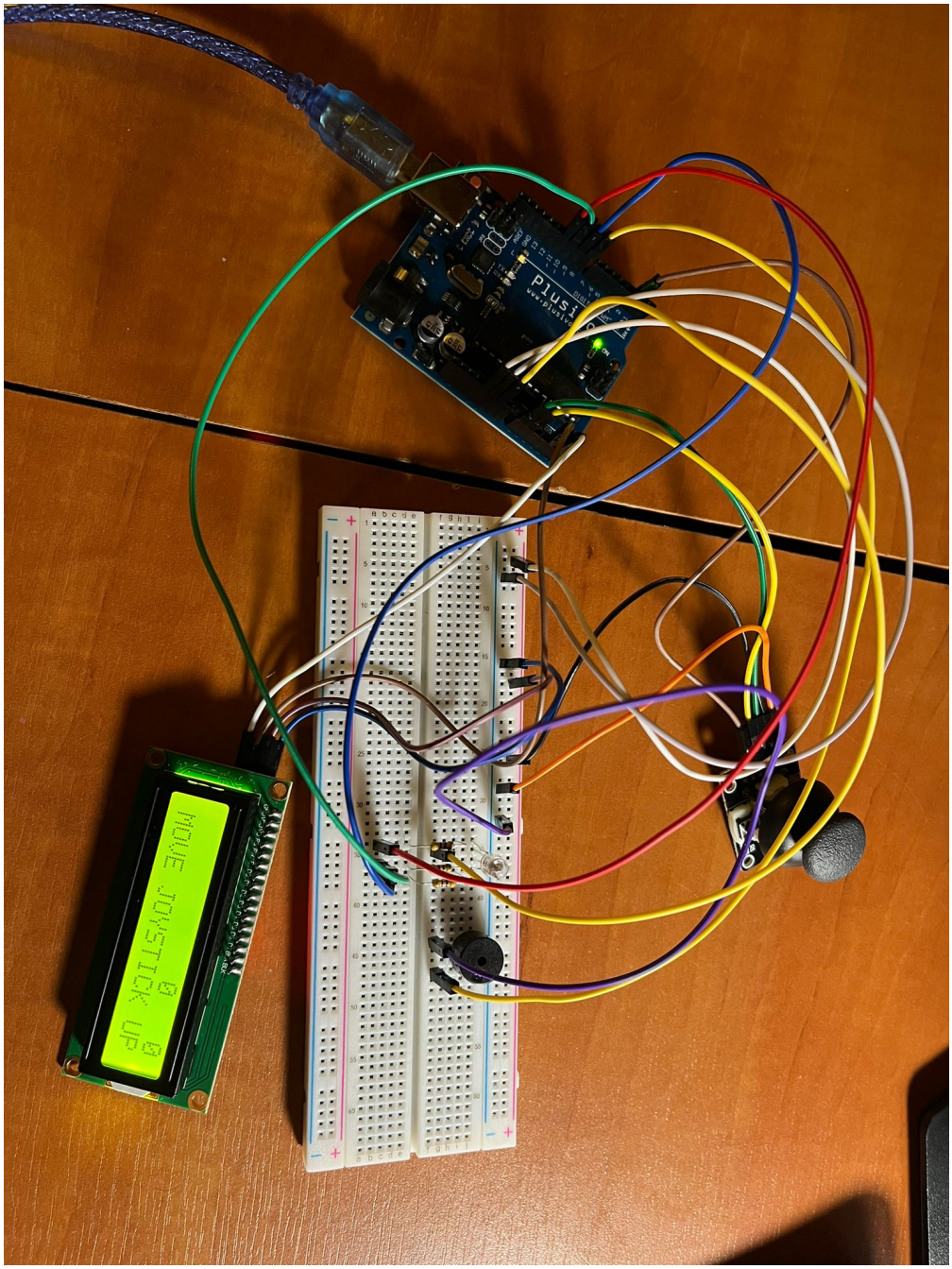
Masina se misca intre planuri prin miscarea joystickului.

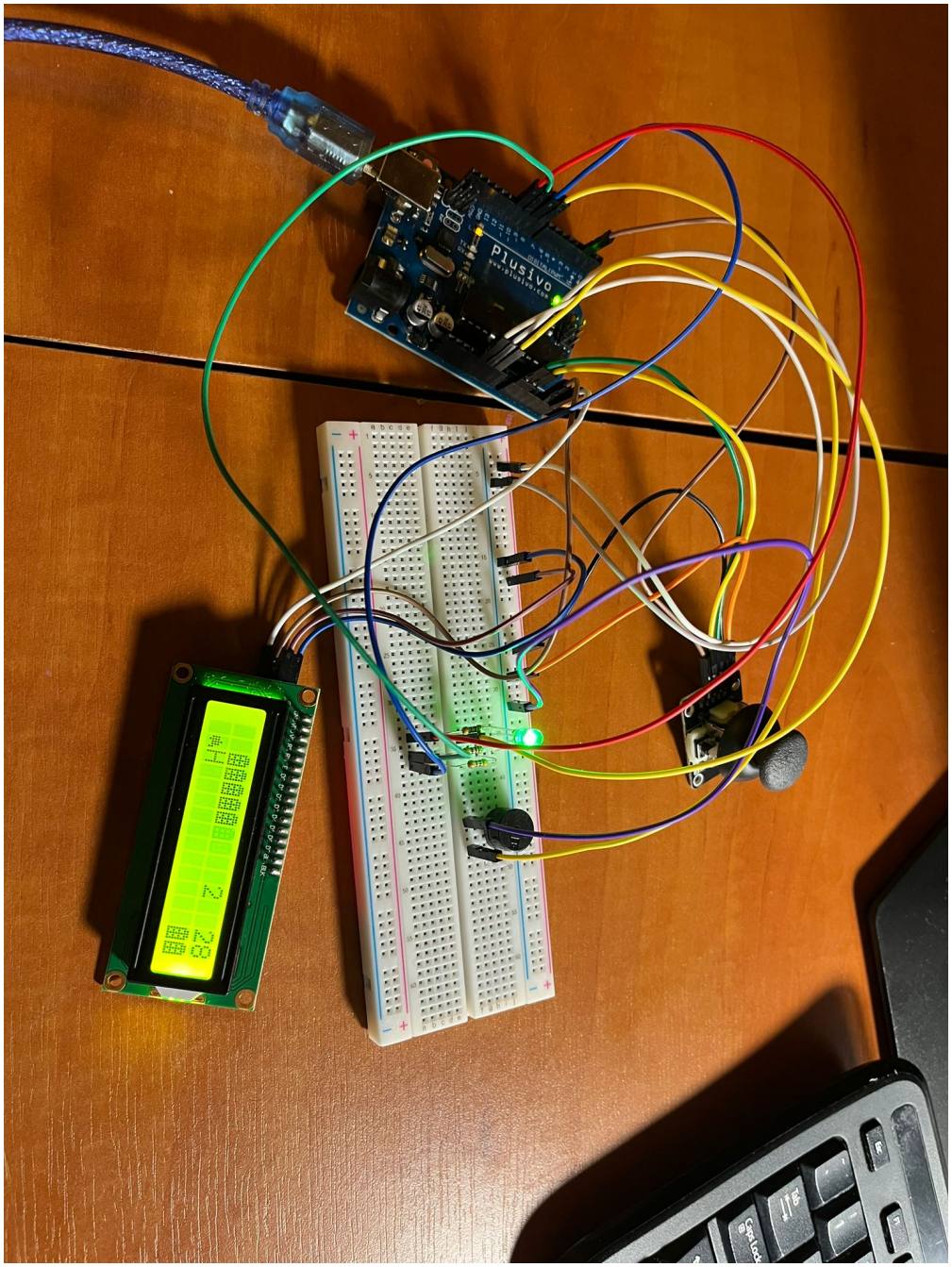
LED ul RGB se lumineaza rosu atunci cand se identifica o coliziune cu un obstacol si verde atunci cand jocul incepe sau se obtine o viata aditionala.

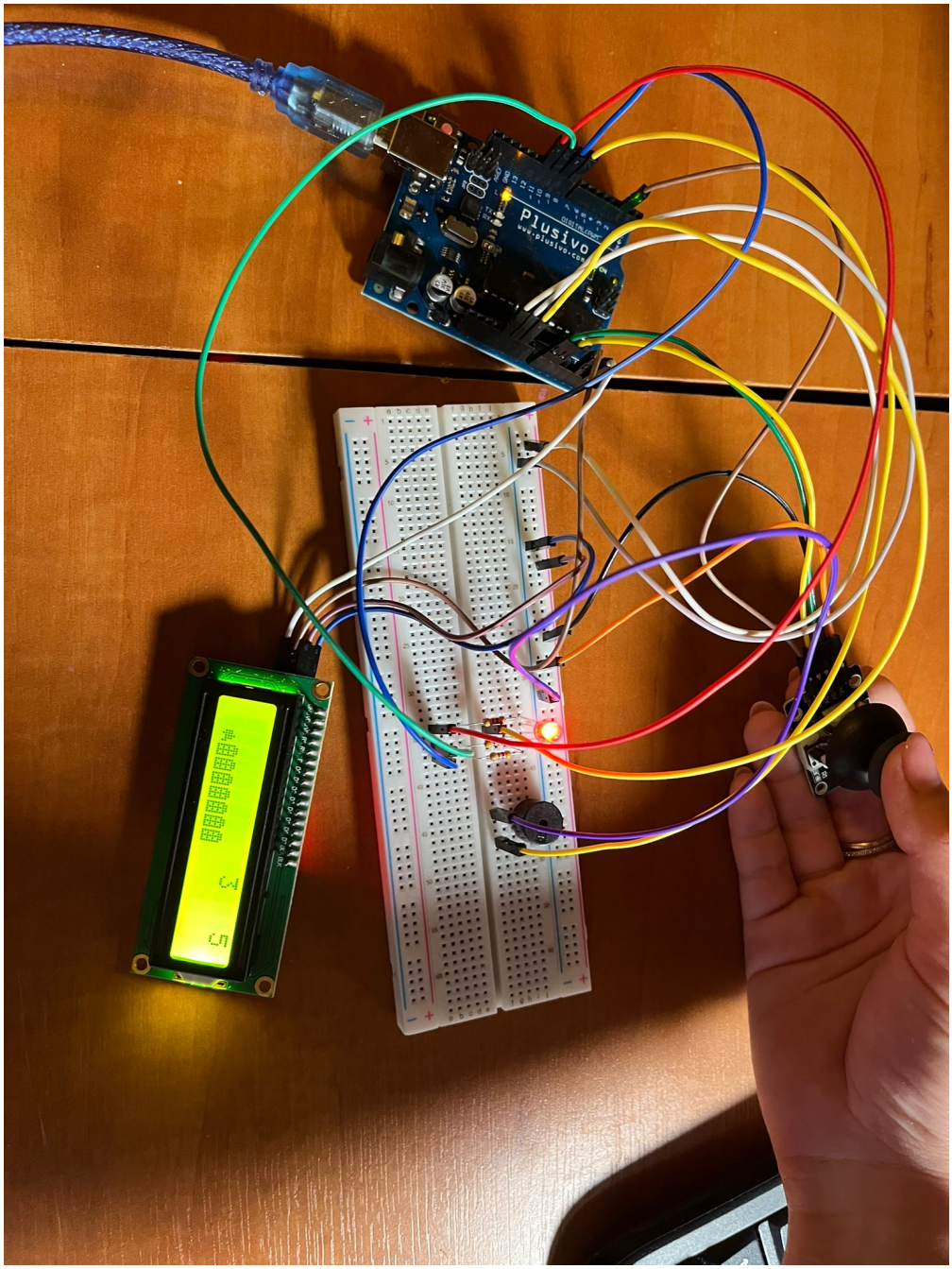
Buzzerul genereaza un sunet specific atunci cand se pierde o viata, cand se castiga o viata sau cand se pierde jocul.

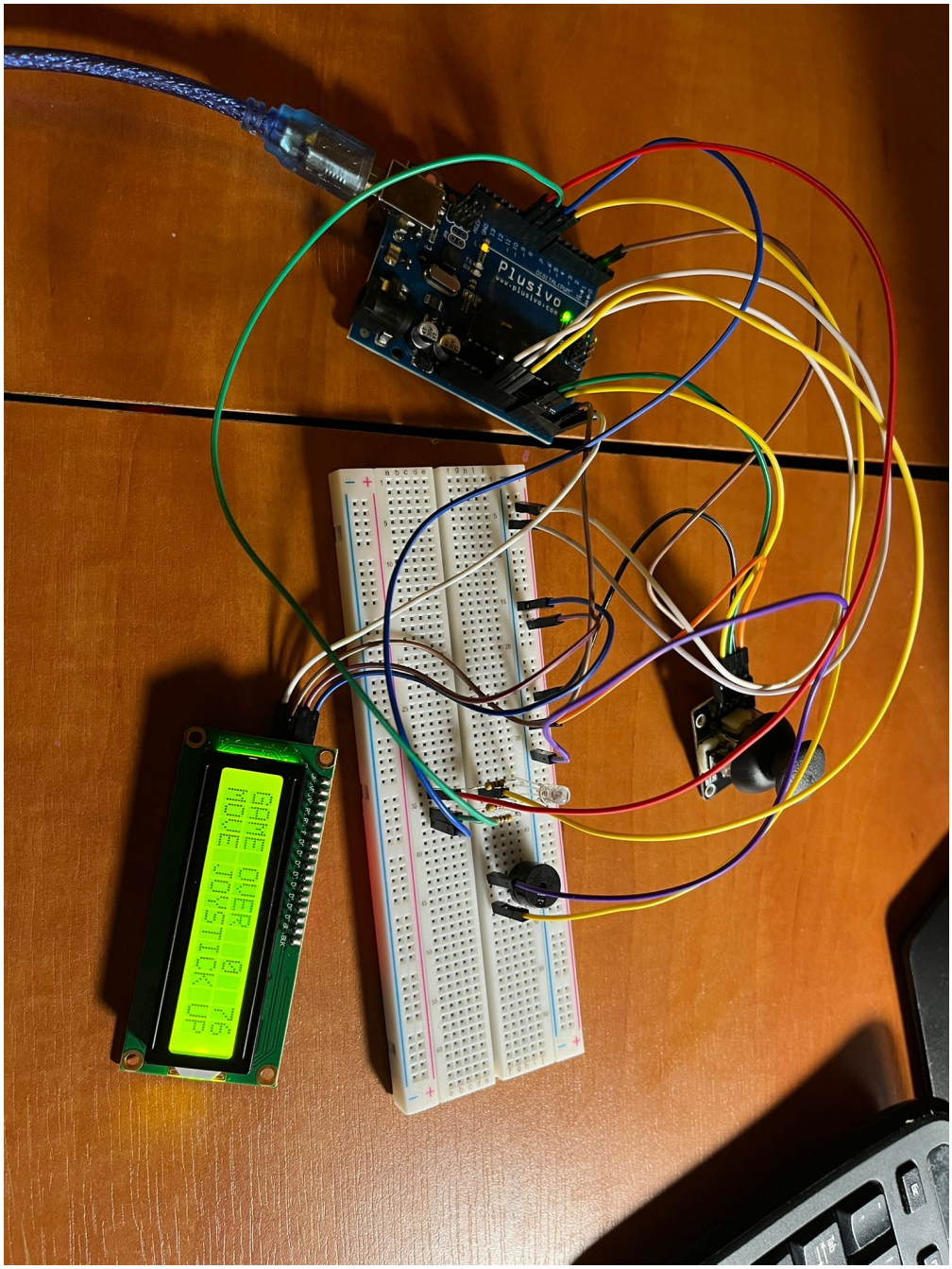
Rezultate Obţinute

Galerie foto:






Plusivo



Prin toate cele prezentate am reușit implementarea unui joc minimal, plăcut și ușor de jucat. Mai jos se poate găsi un link către un scurt demo al jocului.

Demo: https://youtu.be/GTjpPZyiX9o

Concluzii

Consider ca proiectul a fost unul interesant si ma bucur ca am reusit sa-l implementez pana la capat, in ciuda dificultatilor intampinate pe parcursul implementarii. A fost o experiență nouă în cadrul căreia am reușit să iau contact cu programarea embedded și să realizez un joc funcțional pornind doar de la câteva componente hardware si multe idei.

Download

- Pagina pentru OCW pdf: proiect_traffic_racer.pdf
- Arhiva cod: craciunandradasinziana_trafficracergame.zip

Jurnal

18 Aprilie: alegere tema proiect

2 Mai: realizarea schemei bloc

14-15 Mai: punerea primelor componente de placuta si modelarea unui cod minimalist

21-22 Mai: finalizarea circuitului si a codului

26-27 Mai: realizarea paginii de wiki

Bibliografie/Resurse

Link-uri folosite:

- Magazin online: https://www.optimusdigital.ro/ro/
- Utilizare buzzer: https://create.arduino.cc/projecthub/akshayjoseph666/interface-buzzer-with-arduino-uno-694059?ref=user&ref_id=600499&offset=3
- Utilizare led: https://create.arduino.cc/projecthub/muhammad-aqib/arduino-rgb-led-tutorial-fc003e
- Utilizare joystick: https://arduinogetstarted.com/tutorials/arduino-joystick
- https://www.youtube.com/watch?v=vGZiePqgrnY
- https://www.circuitar.com/projects/controlling-the-buzzer/index.html

Export to PDF

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2022/ncaroi/temperature-and-humidity-alarm

Last update: **2022/05/27 21:12**