

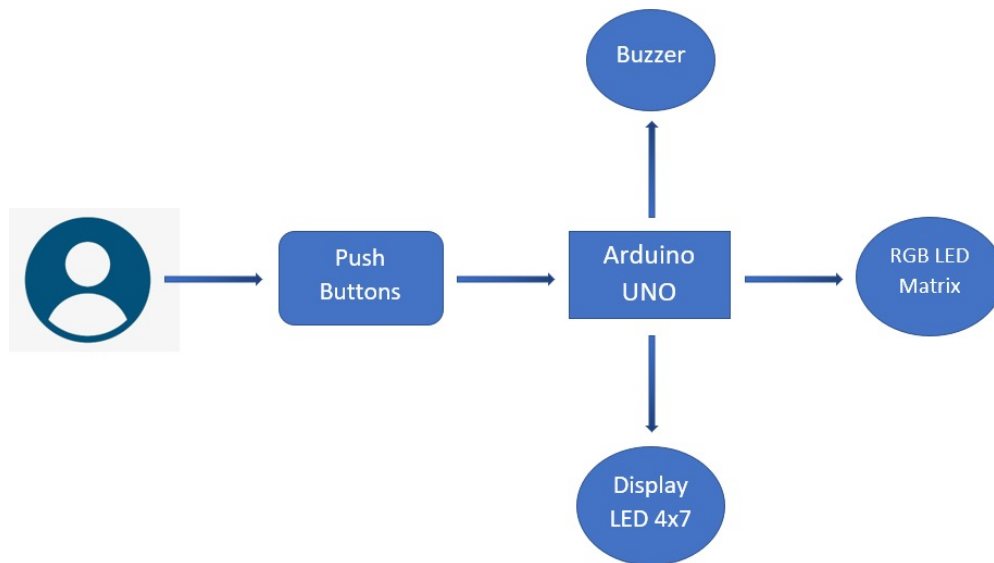
Joc snake

Autor: Pirlog Patricia 332CC

Introducere

- Proiectul consta in implementarea jocului Snake, pe o matrice RGB LED cu scor afisat pe un display LED 4x7 si sunet la atingerea marginilor sau atunci cand se prinde prada.
- Am ales acest proiect uitandu-ma pe proiectele deja existente din alti ani, iar scopul acestuia este pentru destindere si relaxare.

Descriere generala



Schema Bloc

- Jocul va fi controlat prin 4 butoane, iar de fiecare data cand este capturata prada, se va modifica scorul pe display LED si se va activa buzzer-ul (de asemenea si cand se ating marginile ecranului si este game over).

Hardware Design

Componentele necesare:

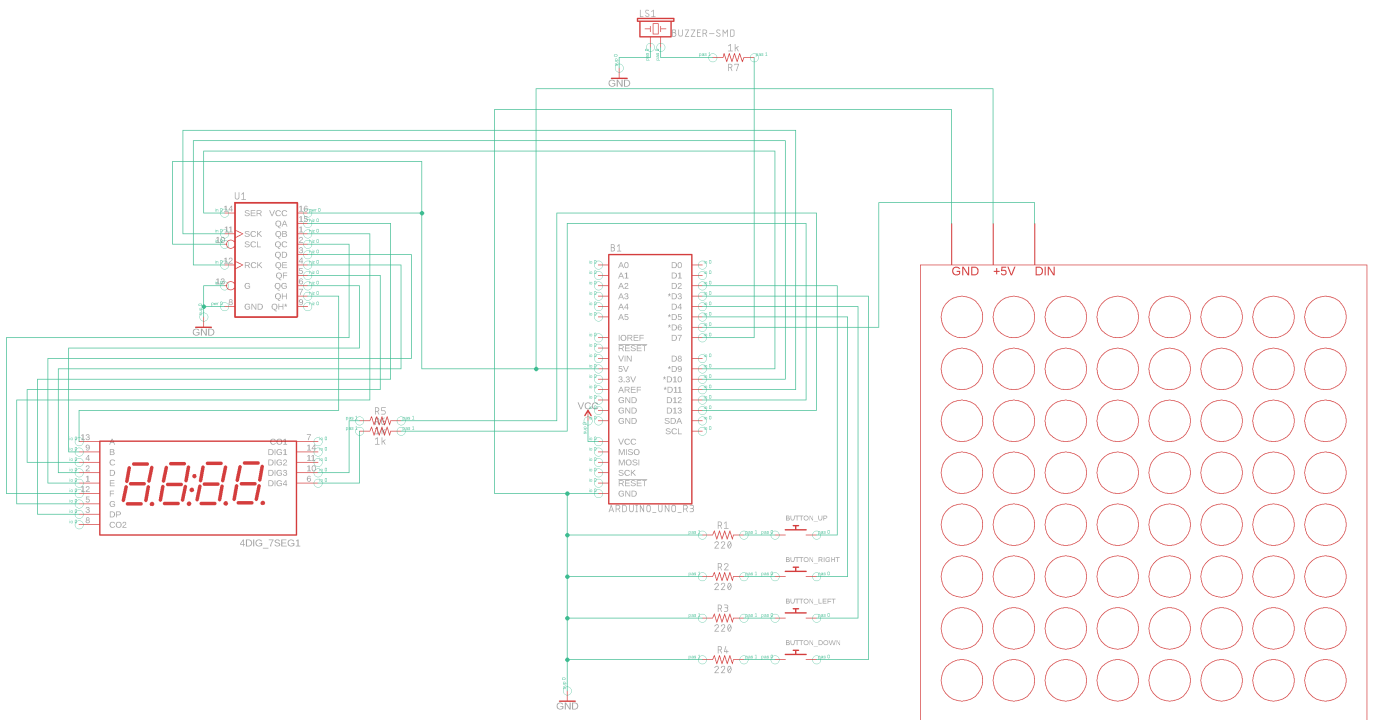
- Arduino UNO
- WS2812 8x8 RGB LED matrix
- Display LED 4x7 segmente

- Shifter 74HC595
- 4 butoane push
- Buzzer
- Breadboard
- Fire jumper si tata-tata
- 4 x rezistente 220 ohm
- 3 x rezistente 1k ohm

Pentru buzzer am folosit o rezistenta de 1K pentru a reduce intensitatea sunetului care ar fi fost mult prea deranjanta. Pentru afisarea scorului, am folosit doar ultimii 2 digits de pe display deoarece acesta nu poate fi mai mare de 64, deci nu aveam nevoie de mai mult.

In total am folosit de pe placuta arduino 11 pini digitali, un pin de 5V si un pin pentru GND.

Schema electrica



Software Design

In implementarea proiectului am folosit mediul de dezvoltare dedicat Arduino. Pentru a controla matricea RGB de LED-uri, am folosit biblioteca:

- [Adafruit_NeoPixel](#)

iar pentru buzzer fisierul cu notele suportate de acesta:

- [itches.h](#)

Matricea RGB este conectata printr-un singur pin (PIN 6) la placuta arduino si toate LED-urile pot fi controlate astfel prin indici. Folosesc mai multe functii pentru afisare in functie de elementul jocului: sarpele este colorat cu verde si este afisat treptat cu functia `displaySnakeLed()`, prada este de diferite culori generate random si este afisata cu functia `displayFood()` iar atunci cand sarpele atinge marginile sau propria coada, locul coliziunii este colorat in rosu (functia `displayDead`) si apare o animatie de Game Over (functia `displayGameOver()`).

Sarpele este controlat prin 4 butoane: UP, DOWN, LEFT, RIGHT, conectate la pinii 2, 3, 4, respectiv 5. Jocul incepe direct odata cu alimentarea placutei la sursa de curent, iar pentru a-l reincepte in urma unui Game Over, se apasa butonul RIGHT. Pentru a evita erori la tinerea apasata a butonului sau apasari multiple, am folosit un `DEBOUNCE_TIME` si un `DELAY_TIME`.

Pentru afisarea scorului, am folosit un 4 Digit 7 Segment Display si un shift register pentru ca pinii de pe placuta arduino nu erau suficiente si a trebuit sa reduc numarul lor. Scorul creste de fiecare data si este afisat atunci cand sarpele captureaza prada, iar pentru formarea cifrelor ma folosesc de ordinea Most Significant Bit si de un vector de bytes ce aprind segmentele de pe display.

Un alt lucru important de mentionat este faptul ca adresarea indecsilor matricei se face sub forma unui "snake pattern" iar pentru asta folosesc o functie `computeIndex()`. Pentru a retine coordonatele pozitiei fiecarui LED din sarpe sau prada, folosesc un struct ce contine 2 int-uri, x si y.

Jocul are 3 state-uri, care diferentiaza actiunea ce are loc la momentul respectiv:

GAME_INIT

Este initiat jocul prin stingerea LED-urilor matricii, setarea coordonatelor pentru sarpe initial in afara matricei si a prazii intr-un loc aleator. De asemenea, size-ul initial al sarpelui este de 2 (LED-uri). Se tine cont la fiecare moment de ultima apasare de buton pentru a nu avea probleme.

GAME_RUNNING

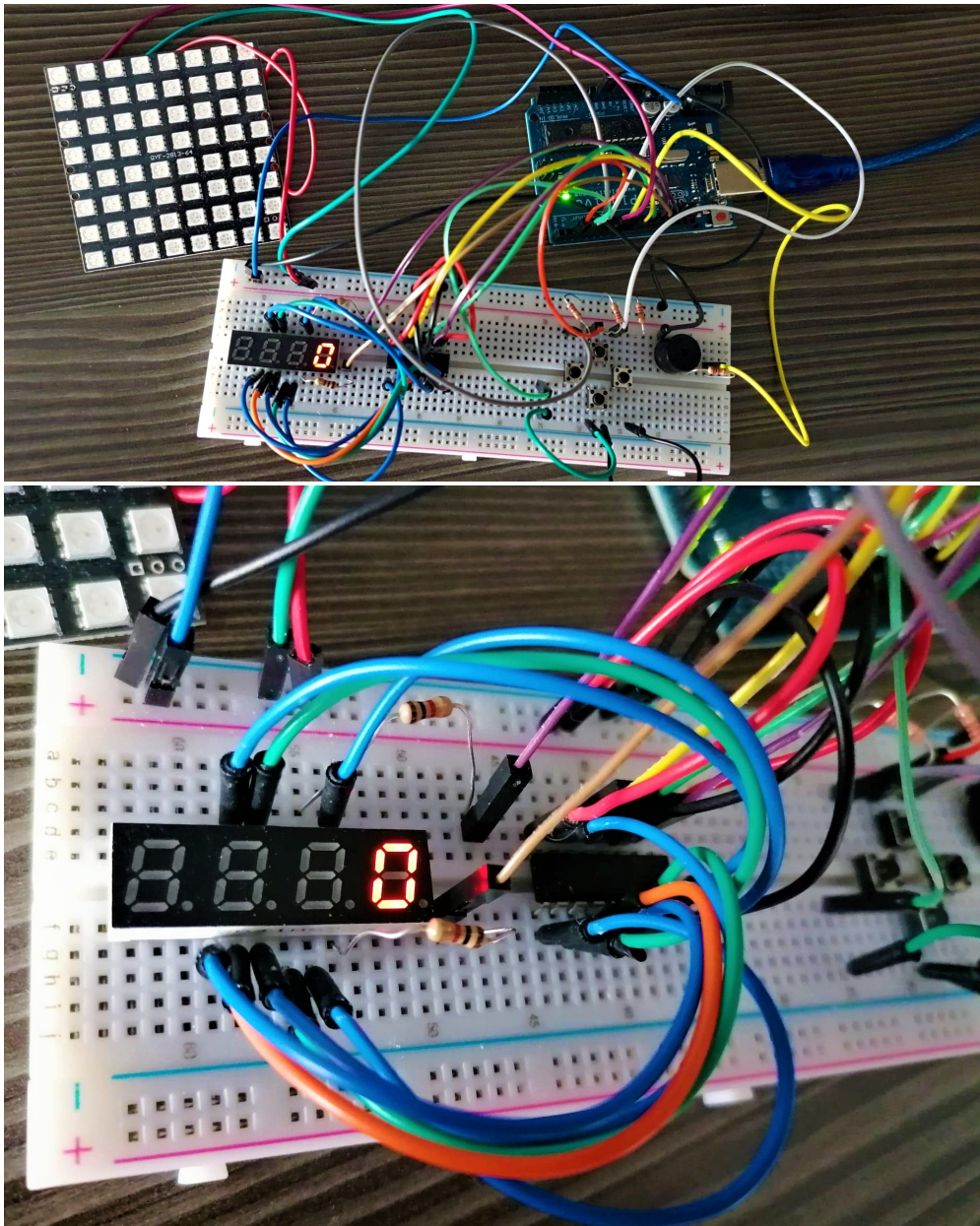
Aici e practic toata logica jocului implementata. La fiecare moment de timp (in functie de `DELAY_TIME`), se updateaza pozitia sarpelui LED cu LED, in functie de ultimul buton apasat. LED-urile se sting si aprind in functie de pozitia LED-ului din fata lor. De asemenea, se verifica de fiecare data coliziunile, caz in care jocul se termina dar si daca cumva este capturata prada (head-ul sarpelui intersecteaza LED-ul respectiv). In acest caz, size-ul creste iar mancarea este respawnata intr-un loc random dar care nu intersecteaza coada sarpelui. La prinderea hranei, buzzer-ul emite un sound de WIN.

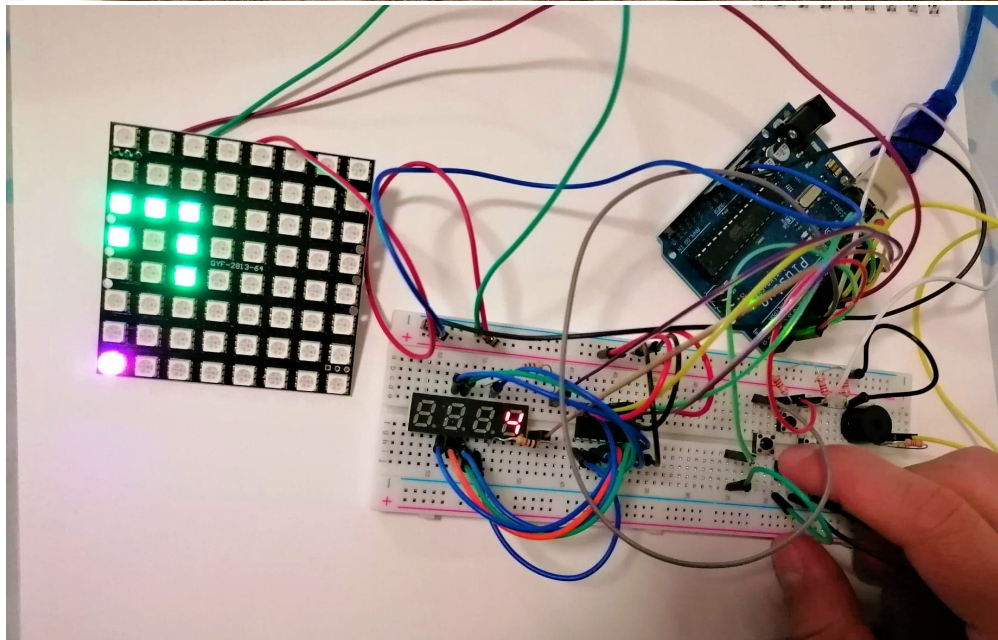
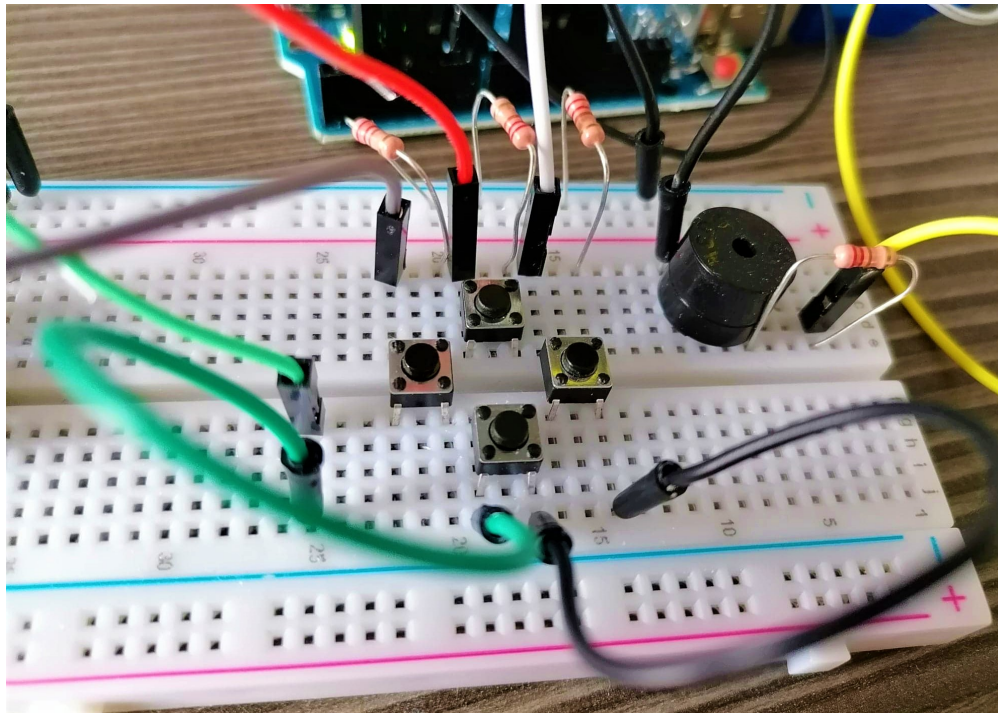
GAME_END

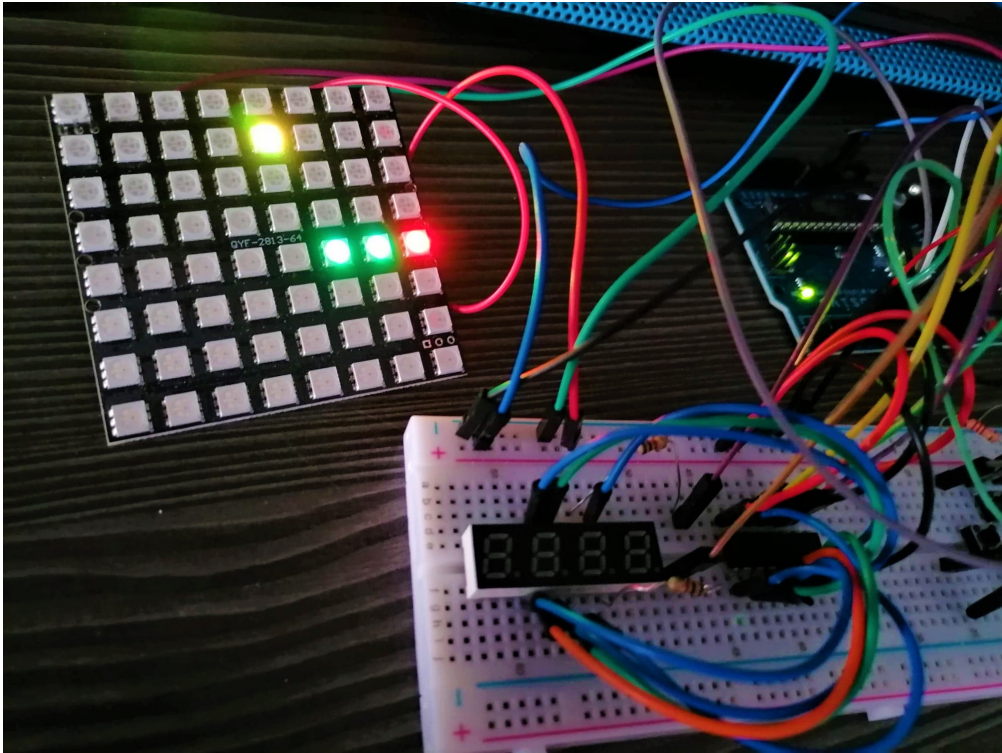
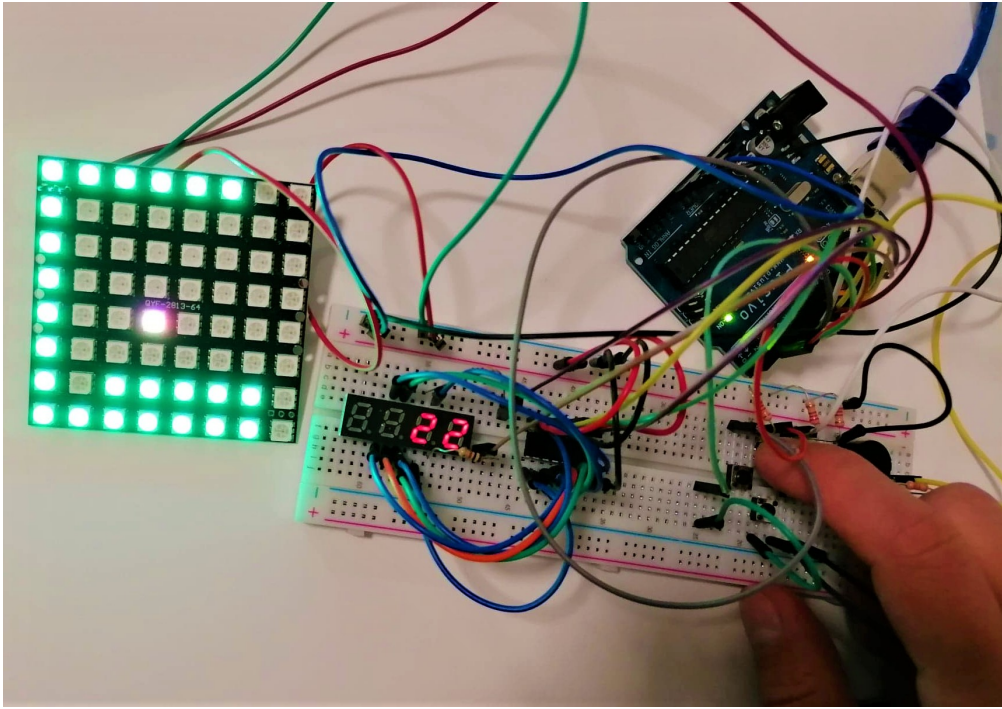
Se afiseaza cu rosu locul unde are loc coliziunea, se aude sunetul de GAME OVER la buzzer si se afiseaza o animatie pe matrice cu un sad face :(.

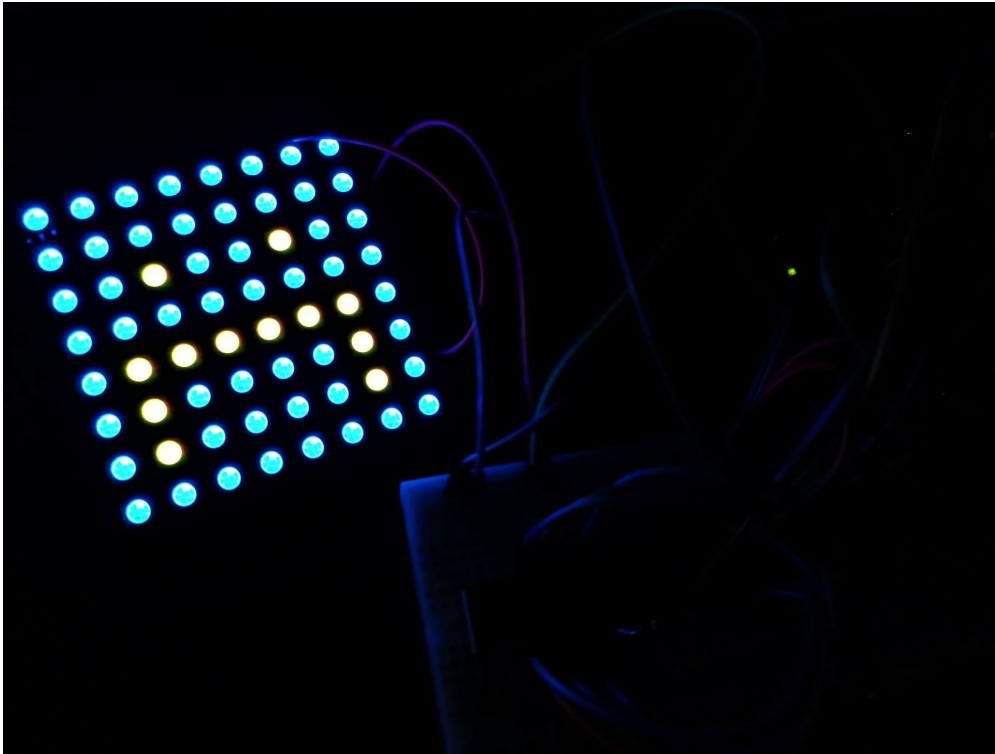
Rezultate Obtinute

Proiectul a iesit asa cum mi-am dorit, si chiar foarte bine din punct de vedere al culorilor de pe matrice, care sunt foarte luminoase si intense.









Download

Proiectul poate fi descarcat sub format pdf de [aici](#).

Concluzii

Proiectul mi-a placut destul de mult, am invatat cum sa folosesc o placuta arduino, cum sa conectez diverse componente si sa realizez logica din spatele functionarii lor. Mi s-a parut una dintre cele mai interesante si placute teme de pana acum.

Videoclipul cu demo-ul proiectului poate fi vazut [aici](#).
Link catre intregul proiect [aici](#).

Bibliografie/Resurse

<https://www.youtube.com/watch?v=RDWgww3LUwU>

<https://www.youtube.com/watch?v=gesrM2J8VnY>

<https://create.arduino.cc/projecthub/goarray/4-digit-7-segment-shift-register-counter-b7338d>

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Debounce>

<https://www.youtube.com/watch?v=2lByUJRN-ss>

<http://arduinolearning.com/code/ws2812-8x8-64-led-matrix-arduino-examples.php>

<https://create.arduino.cc/projecthub/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-un-o-89df45>

<https://www.arduino.cc/en/Tutorial/Foundations/ShiftOut>

<https://www.arduino.cc/reference/en/language/functions/time/millis/>

<https://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library-use>

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

<https://create.arduino.cc/projecthub/goarray/4-digit-7-segment-shift-register-counter-b7338d>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/apredescu/jocsnake>



Last update: **2021/06/03 12:49**