

Joc de memorie

Realizator: Silisteanu Damian-Cristian
Email: damian.silisteanu@stud.acs.pub.ro

Introducere

Ideea de la care am pornit

Am pornit de la jocurile clasice care iti aratau pas cu pas o secventa de leduri aprinse si aveau ca scop retinerea secventei si repetarea acesteia de catre jucator pentru a demonstra ca a retinut-o. A fost un moment cand aceste jocuri au fost foarte populare si erau introduse ca minigame-uri sau sidequest in orice joc.

Descriere

Proiectul consta in realizarea unui joc clasic de memorie la care va trebui retinuta secventa de aprindere a unor led-uri. Jocul va avea doua poduri de a primi input de la utilizator.

Primul mod: Servo controlat cu potentiometru

Jucatorul va reusi sa selecteze led-ul care crede ca s-a aprins prin orientarea servo-ului catre acesta si aprinderea led-ului.

Al doilea mod: Senzor de distanta

Jucatorul va reusi sa selecteze led-ul care crede ca s-a aprins prin pozitionarea unui obiect sau a mainii sale in fata led-ului. Led-ul respectiv se va aprinde pentru a se vizualiza ca intr-adevar acela este dorit.

Scop

Principalul scop este realizare unui joc intuitiv, care sa fie simplu de utilizat, amuzant si competitiv.

Utilitate

Va stimula memoria jucatorului si ii va distra atentia de la problemele de zi cu zi. Va reprezenta un mod de divertisment si de testare a memoriei intr-o maniera cat mai placuta, acest lucru fiind util oricui care se afla in cautarea unui moment de relaxare.

Descriere Generala

Functionalitate

Jocul va incepe cu ecranul de start, adica prin afisarea pe LCD a mesajului "Press any key to START". Dupa apasarea oricarui dintre cele doua butoane(butonul rosu are functionalitatea de selectare mod de input, iar butonul albastru serveste la selectarea ledului dorit) jocul va incepe.

Se va indica secventa de leduri, la inceput formata dintr-un singur led, si se vor afisa pe ecran informatiile necesare desfasurarii jocului, adica numarul de leduri din secventa curenta si scorul actual. Dupa ce secventa este aratata jucatorului acesta poate incepe introducerea acesteia. Acum jucatorul are in orice moment optiunea de a comuta intre modurile de input(servo cu potentiometru sau ultrasonic) prin intermediul butonului rosu.

Pentru modul de input cu servo, jucatorul va regla din potentiometru astfel incat servomotorul sa indice ledul dorit care va sta si aprins. Pentru a confirma ledul dorit jucatorul trebuie ca in momentul in care ledul este aprins sa apese si pe butonul de selectare, adica cel albastru. Similar se va proceda si in cazul senzorului ultrasonic, prin asezarea unui obiect sau a mainii jucatorului in fata ledului dorit, ledul se va aprinde si va trebui selectat cu ajutorul butonului. In timpul jocului, jucatorul va sti pe ce mod de input se afla prin afisarea in coltul din stanga jos a unei iconite sugestive celor doua moduri de input.

In cazul in care ledul indicat este corect acesta va palpai pentru a-i confirma acest lucru jucatorului si se va trece mai departe in asteptarea introducerii celorlalte leduri din secventa sau daca a fost ultimul led din secventa curenta se va afisa secventa cu adaugarea unui nou led la final. Daca ledul indicat este gresit ledurile o sa faca un joc de lumini pentru a anunta jucatorul ca jocul s-a terminat, iar pe LCD se va afisa mesajul "GAME OVER !" urmat de scorul obtinut. In cazul in care jucatorul introduce fara nicio greseala toate secventele de leduri, inclusiv ultima secventa de 10 leduri atunci jocul se va opri, ledurile vor face un alt dans luminos care va anunta jucatorul ca a castigat si pe LCD se va afisa mesajul "CONGRATULATIONS!" si scorul obtinut. Din acest stadiu de joc se poate trece la apasarea oricarui buton in stadiul initial si jocul poate incepe din nou.

Schema bloc

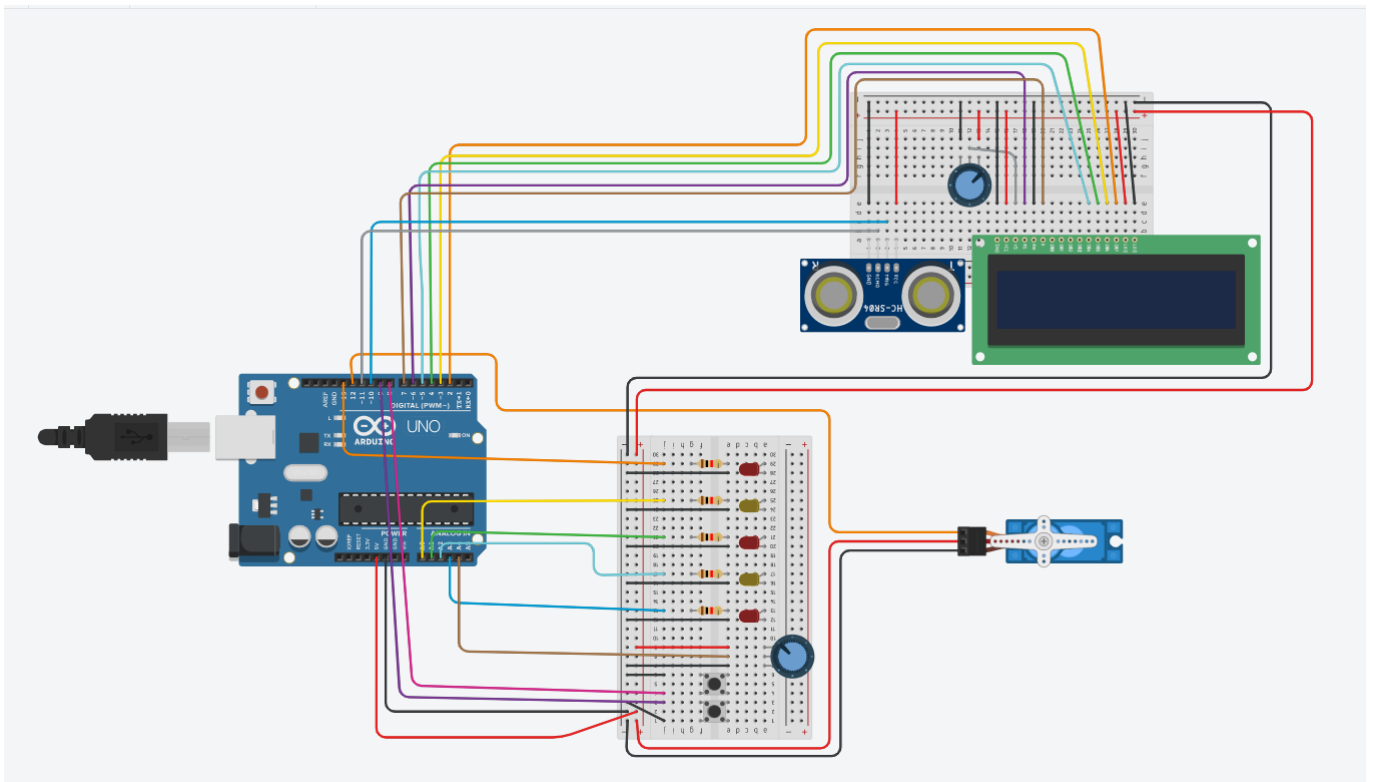


Hardware Design

Lista de componente:

Nume	Numar	Link
Arduino Uno	1	Chit Arduino
Breadboard	2	Chit Arduino
Potențiomtru 10K	2	Chit Arduino
Buton	2	Chit Arduino
Rezistenta 1K	5	Chit Arduino
Fir	52	Chit Arduino
Leduri roșii, galbene	5	Chit Arduino
LCD	1	Chit Arduino
Senzor ultrasonic	1	Senzor de distanta
Servomotor	1	Chit Arduino

Schema electrica:



Dupa cum se poate observa din schema electrica am utilizat toti pinii de pe placa. Intreaga lista a pinilor cu denumiri specifice fiecaruia poate fi gasita in codul sursa in antetul fisierului ca define-uri. Potentiometrul de langa LCD are rolul de a modifica luminozitatea acestuia.

Software Design

Mediul de dezvoltare

Arduino IDE

Biblioteci si surse 3rd-party

Din acest punct de vedere am utilizat doua biblioteci oficiale Arduino:

1. [Servo.h](#): folosita la controlarea mai simpla a servomotorului
 2. [LiquidCrystal.h](#): folosita la controlarea mai simpla a display-ului LCD
- Pentru ca am intampinat probleme la implementarea verificarii apasarii butoanelor atat prin metoda simpla, cat si prin metoda cu intreruperi (la o apasare se apela de 3-4 ori intreruperea) am cautat solutii online si am gasit implementarea unei clase [Button](#), ale carei metode gestionau exact problema mea.
 - Alte functii puse la dispozitie de Arduino care au fost utilizate in dezvoltarea proiectului sunt: [random](#), [map](#) si [pulseIn](#).

Algoritmi si structuri pe care planuiti să le implementati

Principalul algoritm de implementat este cel ce gestioneaza fluxul jocului, mai precis stările acestuia. Jocul va avea 5 stări:

1. **Starea de inceput** este starea in care se deschide jocul si se asteapta apasarea oricarui buton de catre jucator.
2. **Starea de initializare** este starea in care se mai adauga un led la finalul secventei si ii este prezentata jucatorului intreaga secventa.
3. **Starea de joc** este starea in care se asteapta introducerea secventei de leduri de catre jucator prin orice mod de input.
4. **Starea finala** este starea in care se ajunge pentru ca jucatorul greseste secventa si se desfasoara animatia luminoasa specifica.
5. **Starea castigatoare** este starea in care se ajunge dupa ce jucatorul a introdus, fara sa greseasca, toate secventele, inclusiv ultima secventa de 10 leduri.

Surse si functii implementate

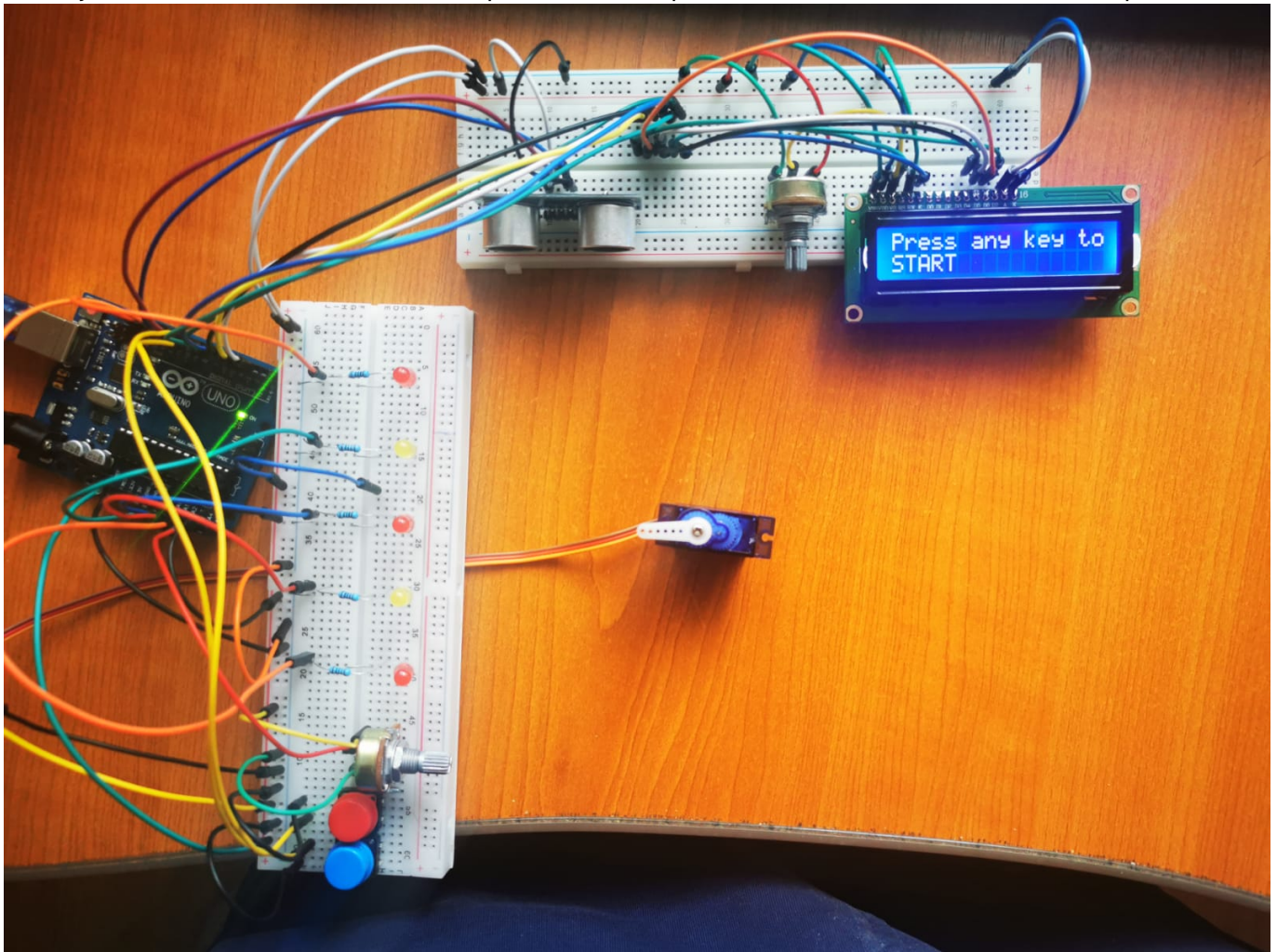
As dori sa prezint ca mod de implementare **starea de joc** pentru ca este cea mai complexa si, de fapt, epicentrul intregului joc. Aceasta incepe prin verificarea modului de input si daca se doreste

schimbarea sa. Apoi se asteapta input de la modul respectiv, fie el potentiometru sau senzor de distanta. Apoi se verifica acest input pentru a se identifica ledul pe care il doreste jucatorul in cazul in care doreste sa il selecteze. Se verifica daca jucatorul selecteaza ledul respectiv, in acest caz se verifica daca ledul selectat se potriveste in secventa la pozitia care era la rand de verificat. Daca se potriveste si mai sunt leduri in secventa curenta de verificat se ramane in aceeași stare doar ca se trece indexul de verificare mai departe. Daca nu mai sunt leduri in secventa curenta si secventa nu este de 10 leduri, atunci jocul intra in starea de initializare pentru a se mai adauga leduri. Daca secventa curenta este cea finala, de 10 leduri, jocul intra in starea castigatoare. Daca ledul selectat a fost gresit, jocul intra in starea finala de unde se poate reincepe jocul.

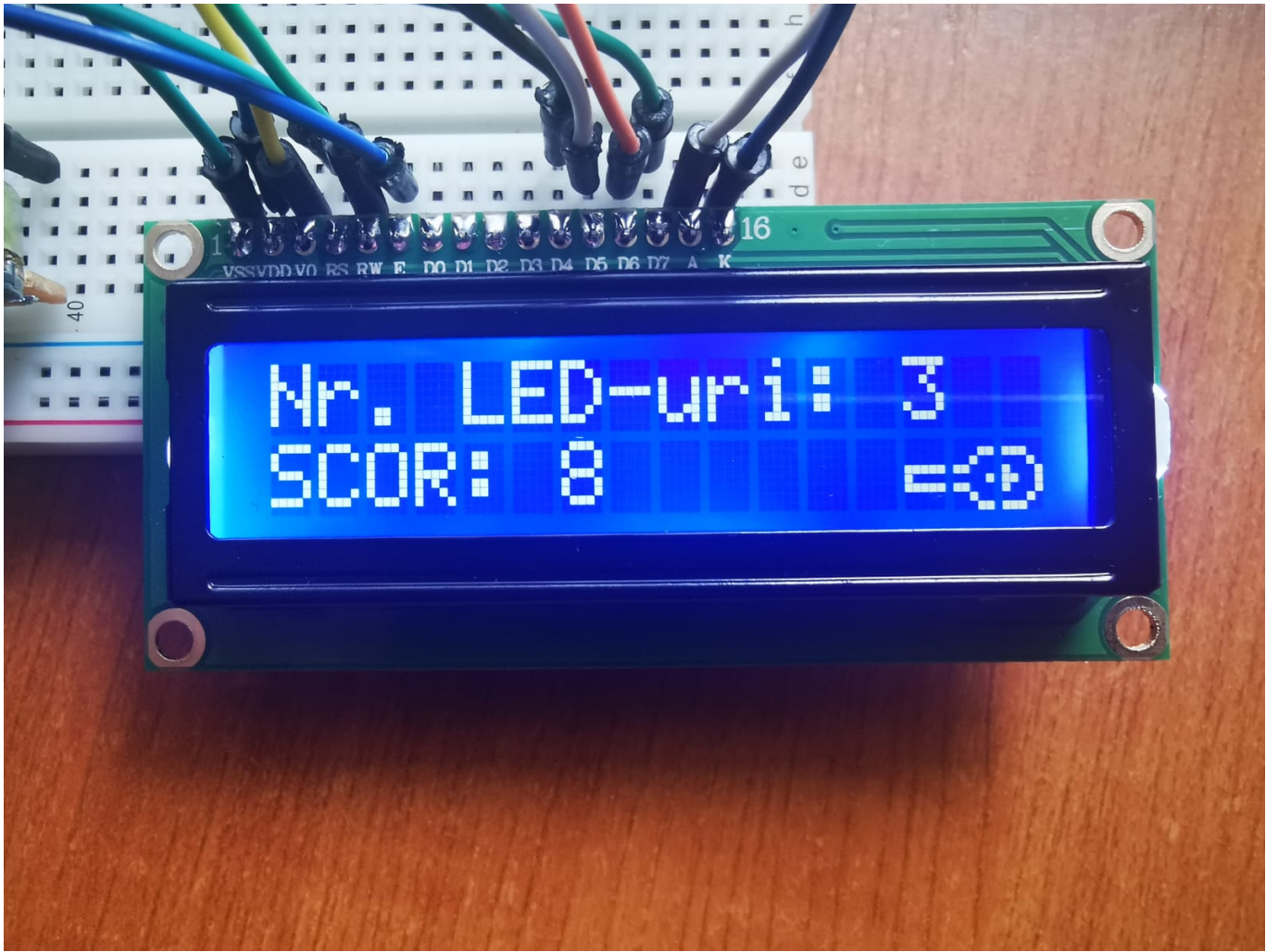
Alte functii implementate au fost: animatia de castig, animatia de resetare joc, animatia de schimbare mod input, functia care se ocupa cu afisarea pe display a unui simbol specific modului de input si functia care mai adauga un led la finalul secventei si afiseaza intreaga secventa.

Rezultate Obținute

Montajul final arata in felul urmator(se poate observa pe LCD ca ne aflam in starea de inceput):



In starea de joc se poate observa in coltul din dreapta jos simbolul pentru modul de input cu servomotor si potentiometru:



si modul de input cu senzorul de distanta:



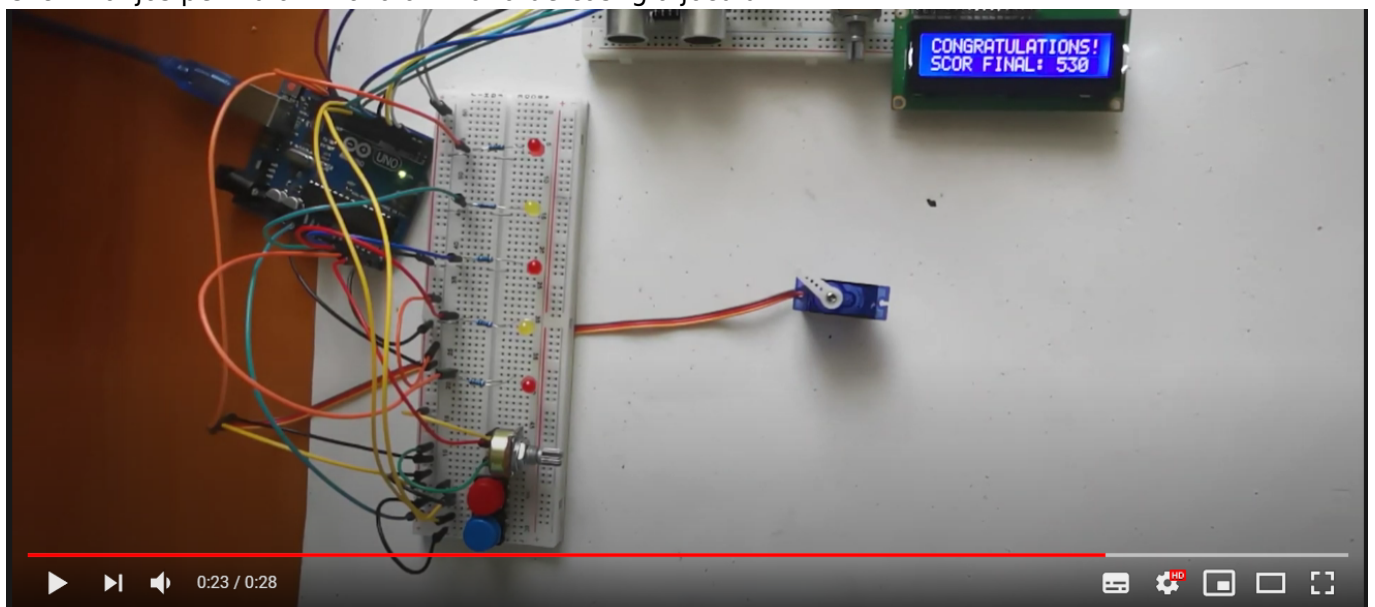
In poza de mai jos se poate observa ecranul de care este intampinat jucatorul in cazul in care greseste un led:



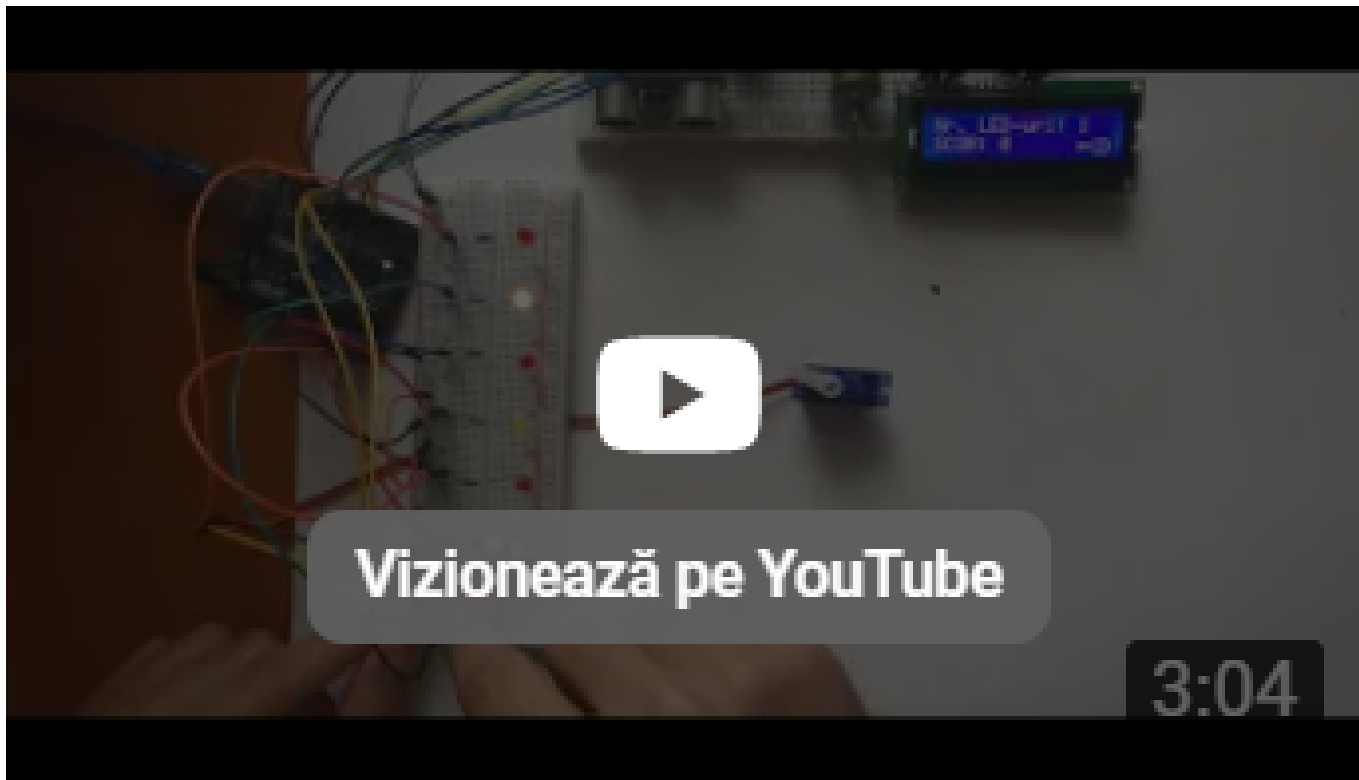
In poza de mai jos se poate observa ecranul de care este intampinat jucatorul in cazul in care termina cu succes toate secventele de leduri:



Click mai jos pentru a viziona animatia de castig a jocului:



Click mai jos pentru a viziona demo-ul:



Concluzii

- Am realizat proiectul exact așa cum mi-am propus să fie și sunt mandru că am reușit să îl aduc în acest stadiu.
- Ar mai merita adăugări viitoare cum ar fi: o carcasă a jocului care să ascundă de jucător întreaga structură de legături, adăugarea unui buzzer care să indice sonor anumite evenimente din joc, adăugarea unui mod de reținere a numelui persoanei respective la terminarea jocului și afișarea unui clasament.
- Proiectul m-a făcut să înțeleg cât de utile și ușor de folosit pot fi anumite module ca senzorul de distanță și display-ul LCD.
- Principalele învățături trase din acest proiect sunt:
 1. să nu exagerezi niciodată cu timpi de delay
 2. să rationalizezi cu foarte mare atenție pini de pe placă Arduino

Download

Fisierele sursă, schema bloc, schema electrică, precum și alte indicații se află în arhivă ce poate fi descărcată de [AICI](#).

Jurnal

- 23.04.2021: Ajunge comanda cu chitul de Arduino în care se află majoritatea pieselor.

- 28.04.2021: Ajunge si comanda data separat pentru senzorul de distanta.
- 22.05.2021: Incep realizarea proiectului prin implementarea miscarii servomotorului si de aici incep problemele:
 1. Placa din chit nu are destula putere cat sa alimenteze servomotorul(acesta scoate sunete dar nu se misca)
 2. Din cauza incercarilor repetate de a face servomotorul sa se miste Serialmonitor o ia razna la printarea valorii unghiului dat servomotorului comanda.
 3. Aceste lucruri duc la stricarea bootloader-ului placi Arduino.
 4. Obtinerea unei placi imprumut si realizarea calitatii scazute a placii din chit.
 5. Impelmentarea cu succes a miscarii servomotorului si indicarii ledurilor prin aprinderea lor.
- 25.05.2021: Realizarea partii de inceput si de introducere a secventei bazata doar pe servomotor si potentiometru.
- 26.05.2021: Incorporarea senzorului de distanata ca mod de input, realizarea unor masuratori si teste importante.
- 28.05.2021: Lipirea pinilor LCD-ului, obtinerea celei de-a doua placi breadboard, realizarea functionalitatii LCD-ului si implementarea starii de final si starii castigatoare.
- 29.05.2021: Inceperea scrierii documentatiei si modularizarii codului.
- 31.05.2021: Finalizarea documentatiei.

Bibliografie/Resurse

Documentația in format PDF poate fi descarcata de [AICI](#).

Bibliografie:

1. Laboratoarele facute anterior: [Lab3](#), [Lab2](#).
2. Diverse tutuoriale pentru: [LCD](#), [servomotor](#), [senzor de distanta](#).
3. Diversele functii si biblioteci folosite: [Servo.h](#), [LiquidCrystal.h](#), [Button.h](#)(se afla in arhiva cu sursele), [random](#), [map](#) si [pulseIn](#).
4. Site pentru piese de Arduino(pe care **NU** il recomand daca doresti o placa mai serioasa si sa nu ti se intample ce am patit eu«**vezi sectiunea de Jurnal**») [AICI](#).

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/dbrigalda/jocuri-de-memorie>



Last update: **2021/06/02 11:33**