

Remember the Glow

Student: Strîmbeanu Ionela Cristina
Grupa: 331CC

Introducere

- Joc de memorie ce testeaza atentia si reactia rapida.
- Scopul jocului: retinerea si reproducerea unei secvente de lumini, cu accent pe observarea schimbarilor dinamice din cadrul jocului.
- Ideea a pornit din dorinta de a crea un joc interactiv si captivant, mai complex decat varianta traditionala a jocului Simon Says.
- Util pentru evaluarea abilitatii de a mentine atentia intr-un mediu caracterizat de distrageri repetate.

Descriere generală



Etapele jocului:

1. Pe matricea principala de LED-uri se va aprinde un LED.
2. Pe cele doua matrici secundare de LED-uri se vor forma doua imagini. Matricile sunt conectate serial, miscorand astfel numarul de pini digitali necesari de pe microcontroler.
3. Prin intermediul a doua butoane, simbolizand DA sau NU, jucatorul trebuie sa raspunda corect la intrebarea: "Cele doua imagini sunt la fel?"
4. Se repeta pasii anteriori de 5 ori.
5. Prin intermediul unei matrice de butoane, corespunzatoare LED-urilor de pe matricea principala, jucatorul trebuie sa introduca corect secventa de LED-uri aprinse anterior.
6. Daca utilizatorul introduce corect secventa de lumini SI a raspuns corect la toate intrebarile referitoare la imagini, inseamna ca a castigat si se vor inalta steagurile victoriei, pe fundalul unei melodii triumfatoare.

Atentie la timp! Nu ai o vesnicie sa te gandesti.

Hardware Design

Schema hardware - realizata in wokwi.com:



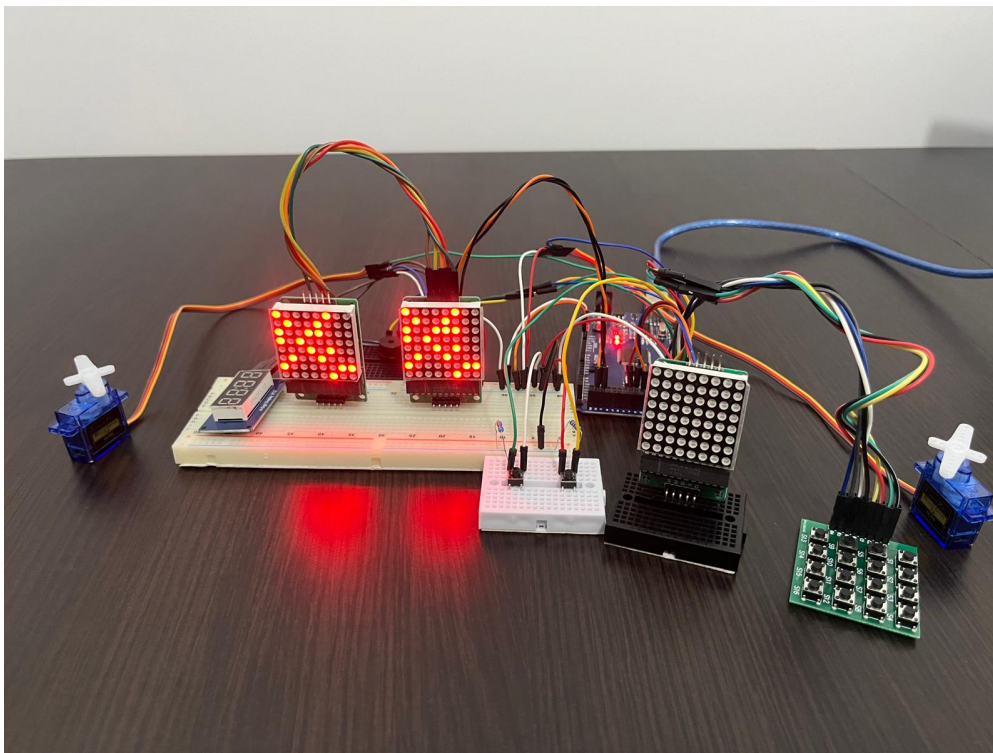
Lista piese:

- Arduino MEGA2560
- Matrice de LED-uri MAX7219 - folosite pentru reprezentarea efectiva a jocului: o matrice va reda secventa de lumini ce trebuie memorata, iar celelalte doua vor fi folosite pentru a arata jucatorului diferite combinatii de lumini (imagini)
- Matrice de butoane 4x4 - permite jucatorului sa introduca secventa de lumini
- Micro Servomotor SG90 - atunci cand jucatorul a castigat servomotoarele vor flutura mici stegulete
- 4-Digit Display - folosit ca si cronometru, pentru a aminti jucatorului cat timp mai are
- Butoane extra - doua la numar, folosite pentru ca jucatorul sa spuna daca cele doua matrici de LED-uri arata aceeasi imagine
- Buzzer - folosit pentru a reda diferite sunete si melodii, in functie de momentul jocului
- Breadboard

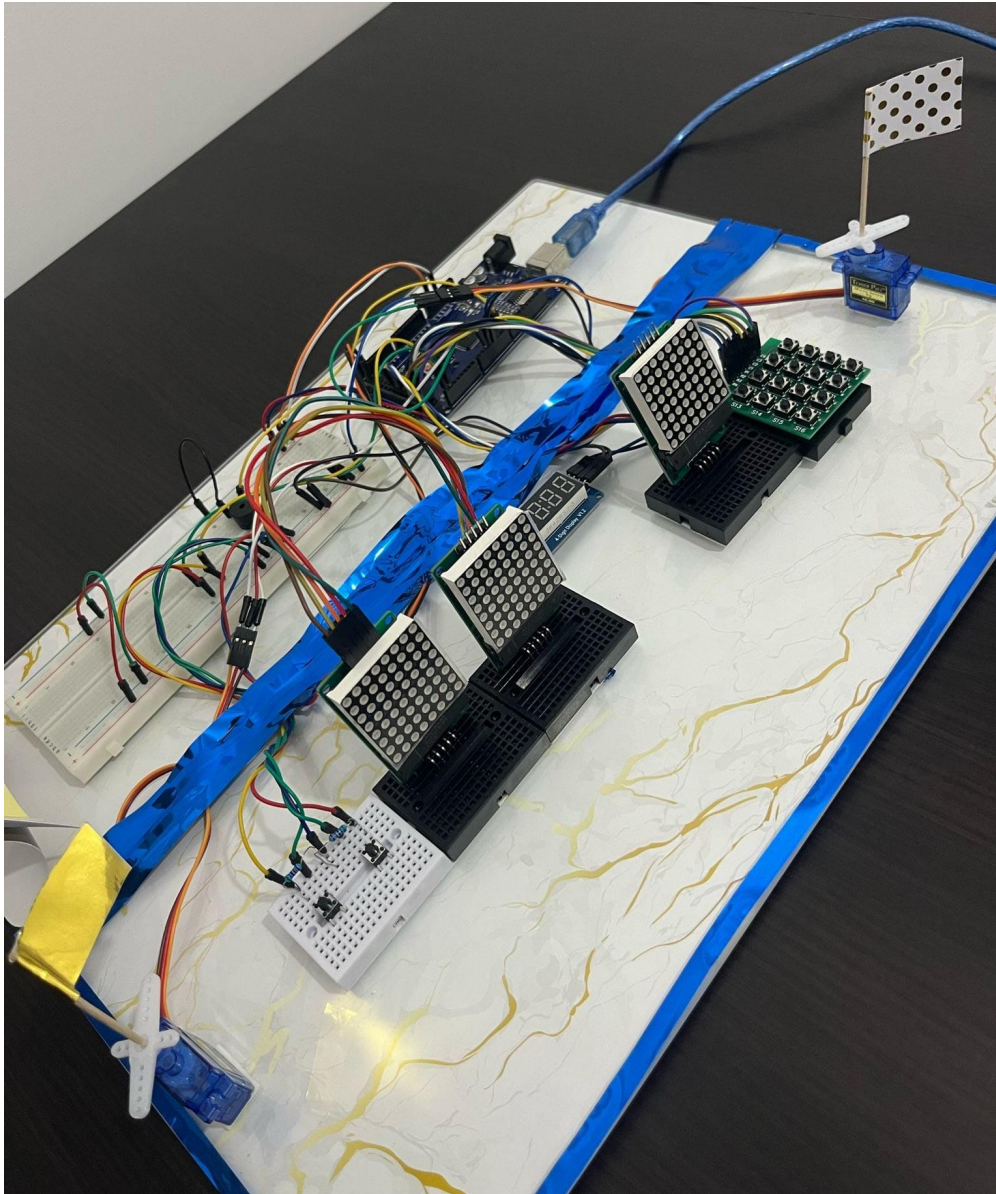
Alegerea pinilor:

- Cum matricile de LED-uri folosesc SPI, am folosit pinii aferenti de pe placa, respectiv D51, D52, D53.
- Dupa cum se observa pe schema hardware, am doua constructii diferite de matrici de LED-uri, motiv pentru care, pentru usurinta, matricea principala nu am conectat-o tot la pinii speciali de SPI de pe placa. In schimb, am conectat-o la pinii D36, D38, D40, biblioteca folosita realizand "bit banging", ceea ce imi permite sa folosesc orice pini digitali.
- Butoanele "DA" si "NU", folosite pentru a afirma daca imaginile de pe cele doua matrici sunt identice, folosesc intreruperi externe, deci sunt conectate la pinii D20 si D21.
- Cele doua Servo Motoare folosesc PWM si trebuie conectate la pini adecvati (care accepta PWM). Am ales random D2 si D3.
- Restul componentelor folosesc pini digitali, alesi random. Am incercat totusi sa ii aleg astfel incat sa minimizez firele intercalate si sa arate cat mai bine.

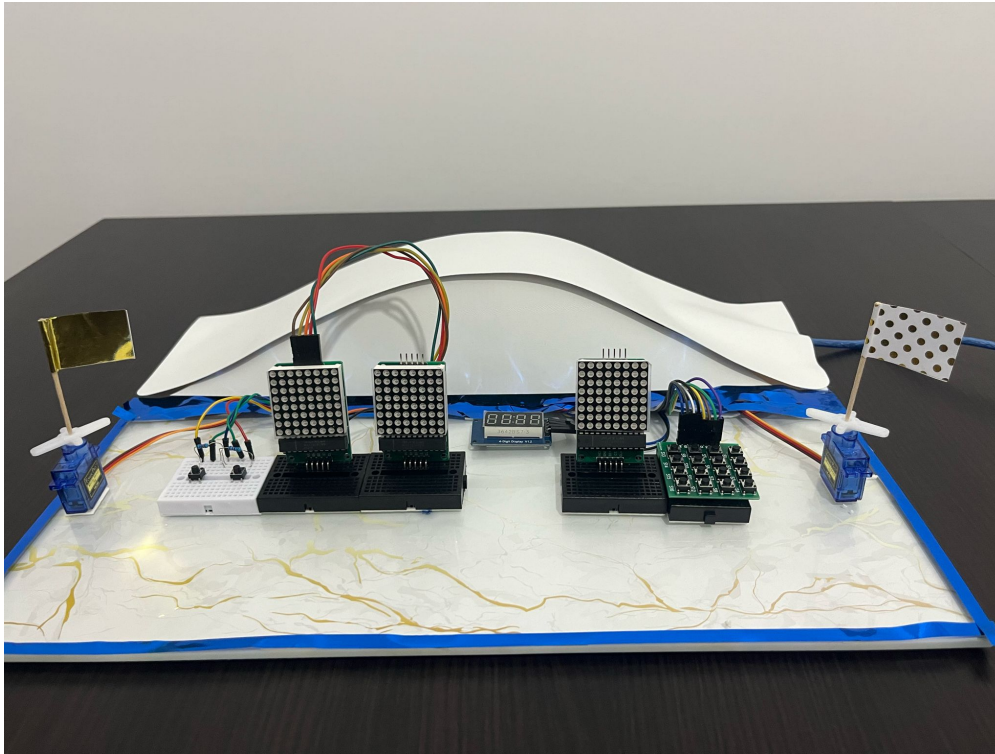
Forma initiala hardware:



Forma finala hardware:



Proiect final - prezentare:



Software Design

In dezvoltarea si testarea proiectului s-a folosit mediul de dezvoltare Arduino IDE.

Pentru simplitatea controlului pieselor s-au utilizat urmatoarele biblioteci:

- LedControl pentru Matricile de LED-uri MAX7219
- Adafruit_Keypad pentru Matricea de butoane
- TM1637Display pentru Cronometru (Temporizator)
- ezBuzzer pentru Buzzer

Biblioteca ezBuzzer a fost aleasa special pentru a permite redarea unei melodii in fundal, pe tot parcursul jocului. Toate bibliotecile alese ofera functii predefinite care functioneaza corect si sunt usor de utilizat si inteles.

Functionalitati din laboratoare implementate de mana (fara a avea in spate o biblioteca care se ocupa):

- INTRERUPERI pentru butoanele DA/NU: jucatorul poate apasa butonul care crede ca este corect in orice moment de timp. Totusi, pentru a exclude cazul in care jucatorul apasa pe butoane intamplator, pe tot parcursul jocului, se va considera o apasare valida de buton doar dupa ce au aparut imaginile pe cele doua matrice secundare.
- TIMER (Timer1 mai exact) cu un interval de o secunda, util in configurarea temporizatorului si in incrementarea unor variabile care calculeaza timpul de asteptare intre diferite evenimente din cadrul jocului
- FAST PWM folosit pentru controlul Servomotoarelor

Link proiect: [Remember_the_Glow](#)

Flow-ul programului:



Exista multiple fisiere .h care contin declarari si initializari variabilelor corespunzatoare fiecarei componente in parte. Toate functiile au fost implementate intr-un singur fisierul principal (proiect_pm.ino).

Descrierea functiilor implementate:

- Exista functii pentru stingerea tuturor LED-urilor de pe matricile de LED-uri (blank_matrix_uno, blank_matrix_duo), precum si functii de initializare a acestor matrici (initialize_solo, initialize_duo).
- Functia displayPatternSolo se ocupa de alegerea si aprinderea unui grup de 4 LED-uri pe matricea principala.
- Functia displayPatternDuo se ocupa de alegerea LED-urilor care vor fi aprinse pe matricile secundare. Aici se decide daca cele doua "imagini" reprezentate vor fi identice sau nu. Pentru dificultate sporita, doar 1 sau 2 LED-uri pot sa difere si doar printr-o pozitie.
- Functia manage_buttons_DANU asigura citirea butonului doar daca acesta trebuia apasat (pentru a preveni luarea in calcul a apasarii gresite) si verifica daca este cel corect sau nu.
- Functia check_keypad verifica daca butoanele apasate pe tastatura sunt cele corecte, in ordinea corecta.
- Urmeaza functiile care se ocupa de intreruperi. Intreruperea INTO este declansata de apasarea butonului NU (cel din stanga), iar intreruperea INT1 este declansata de apasarea butonului DA (cel din dreapta). Timer-ul 1 este configurat pe o secunda, ceea ce permite o configurare corecta si precisa a temporizatorului din joc, care scade la fiecare secunda. Timer-ul 3 este setat pe Fast PWM si este folosit pentru rotirea corecta a servomotoarelor.
- Temporizatorul, odata ce a ajuns la 0, va incepe sa licareasca: timp de o secunda se aprinde, timp de o secunda se stinge.
- Functia rotate_servos se ocupa de rotirea servomotoarelor de la 0 la 180 de grade, setand pulse width-ul la valoarea corespunzatoare.

Descriere loop():

- Se alege melodia care ruleaza in fundal, in functie de momentul de timp la care suntem in cadrul jocului. Daca s-a apasat un buton gresit, se ruleaza un sunet dezaprobat corespunzator. Astfel jucatorul stie ca a facut o greseala.
- Se realizeaza trecerea prin etapele jocului folosind variabile booleene pentru fiecare etapa: light_one_led = se aprinde un grup de 4 LED-uri pe matricea principala, light_matrix = se aprind cele doua "imagini" pe matricile secundare, wait_DANU = se asteapta apasarea unui buton din cele doua disponibile, wait_button_matrix = se asteapta introducerea secventei de LED-uri in ordinea corecta, game_final = jocul a luat sfarsit.
- La aceste variabile, se adauga multe altele care asigura buna functionare a jocului: unele numara cate butoane/LED-uri au fost apasate/aprinse pana la momentul curent de timp, altele se ocupa de retinerea corectitudinii sau a faptului ca s-a intamplat ceva (de ex. faptul ca a fost apasat un buton), iar altele sunt folosite pentru contorizarea timpului intre diferite actiuni sau momente ale jocului.
- Odata cu terminarea jocului, se decide daca jucatorul a castigat sau a pierdut pe baza numarului de greseli (nr_greseli) sau pe baza starii temporizatorului (daca a ajuns la 0, automat se considera ca jucatorul a pierdut).
- In functie de rezultat, timp de 30 de secunde, se va reda melodia corespunzatoare si, daca este cazul (jucatorul a castigat), se invart si servomotoarele. Se afiseaza scorul obtinut de jucator, cate actiuni corecte a facut din totalul de 10 actiuni posibile (5 apasari de butoane DA/NU, 5 apasari pe matricea de butoane, corespunzatoare secventei de LED-uri)

Rezultate Obținute

Video proiect: [Remember the Glow Video](#)

Concluzii

Mereu mi-a placut ideea de hardware, de lucrat cu piese si fire, si ma bucur ca am avut oportunitatea de a crea un proiect de la 0, atat din punct de vedere software, cat si hardware.

A fost o provocare. Fiind primul proiect de acest gen, a necesitat multa documentare. Chiar daca tema proiectului ales nu este una foarte complicata, consider ca am reusit sa fac un proiect interesant si, cel mai important, complet functional, intr-un timp limitat.

Inca de la inceput, stiam exact ce vreau sa fac, fiecare pas al proiectului. A necesitat putina cautare pentru a gasi piesele potrivite, astfel incat sa nu imi ingreunez inutil munca. Pe parcursul realizarii jocului am hotarat ca ar fi mai distractiv sa am muzica permanent in fundal. Ideea a evoluat apoi, astfel in proiectul final exista mai multe melodii, cate una pentru fiecare etapa a jocului.

Am pornit de la premisa ca voi folosi breadboard-uri, pentru usurinta, in detrimentul lipirii. Pentru un design mai estetic am hotarat sa folosesc mini breadboard-uri, negre in principal, si am optat pentru ascunderea firelor si a placutei Arduino. Astfel jucatorul este mai concentrat pe joc, si nu pe mediul inconjurator.

Singurul lucru care nu mi-a placut in dezvoltarea acestui proiect a fost faptul ca, lucrand in Arduino IDE, nu am reusit sa folosesc fisiere .c sau .cpp pentru modularizarea codului. Probabil se poate, dar nu am reusit sa fac sa mearga si nici nu mi-am dedicat mult timp pentru a gasi o solutie. Am considerat ca este mai important sa mearga intr-un mod corect si optim decat sa fie impartit in mai multe fisiere.

Download

Arhiva proiect: [Remember_the_Glow_arhiva.zip](#)

Jurnal

02.05 - Creare pagina si Introducere

03.04 - Comandare piese

04.05 - Descriere generala si Comandare piese

05.05 - Incepere Hardware si Software (pe parti)

10.05 - Unire piese Hardware intr-un singur proiect

13.05 - Incepere unire Software intr-un singur proiect

16.05 - Harware Design

18.05 - Comandare piese pentru Design Final Estetic Hardware

23.05 - Software Design

26.05 - Completare pagina ocw cu scheme, imagini noi si concluzii

Bibliografie/Resurse

- [Laboratoare 0-3](#)
- [Wokwi.com](#)
- [OptimusDigital](#)
- Exemple oferite de Arduino IDE, cu bibliotecile folosite:
 - [LED Matrix](#)
 - [Keypad](#)
 - [Timer](#)
 - [Buzzer](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/fgul/ionela.strimbeanu>



Last update: **2024/05/27 10:37**