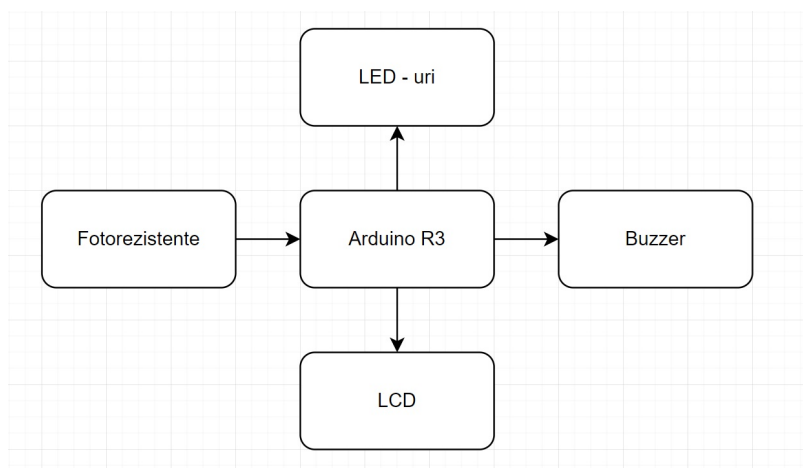


Memory Game

Introducere

Proiectul acesta presupune realizarea unui joc de memorie, in care jucatorul trebuie sa reproduca in mod exact secventa aleatoare pe care jocul o prezinta. De fiecare data cand jucatorul raspunde corect, jocul trece la urmatorul nivel, adaugand inca un nivel de dificultate; in cazul in care jucatorul nu a reusit sa reproduca intocmai secventa, jocul se termina. Acest proiect poate fi folositor pentru testarea memoriei si antrenarea ei.

Descriere generală

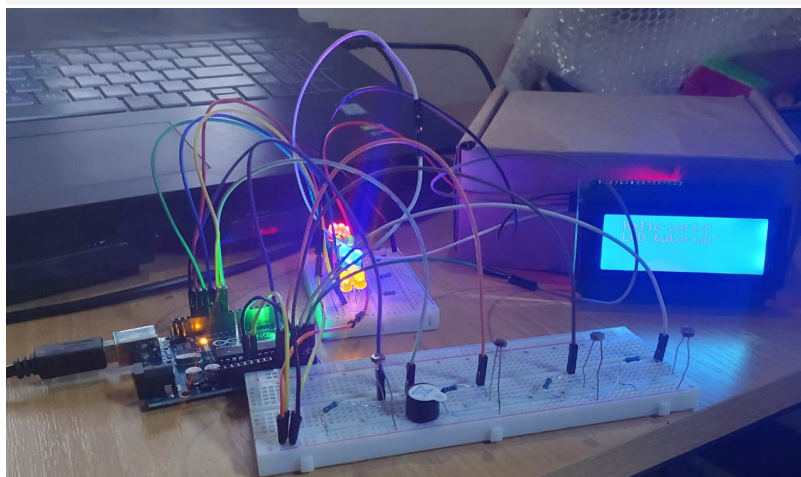
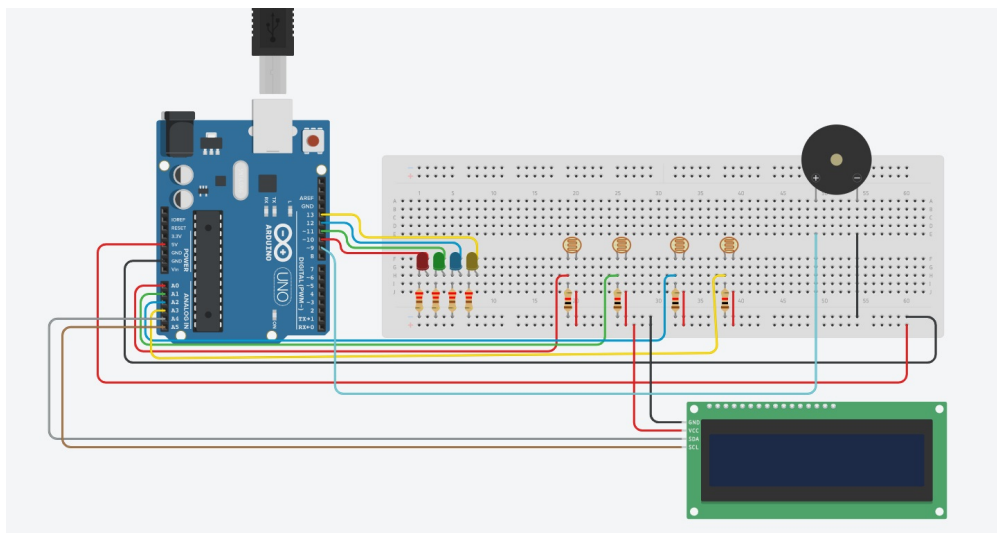


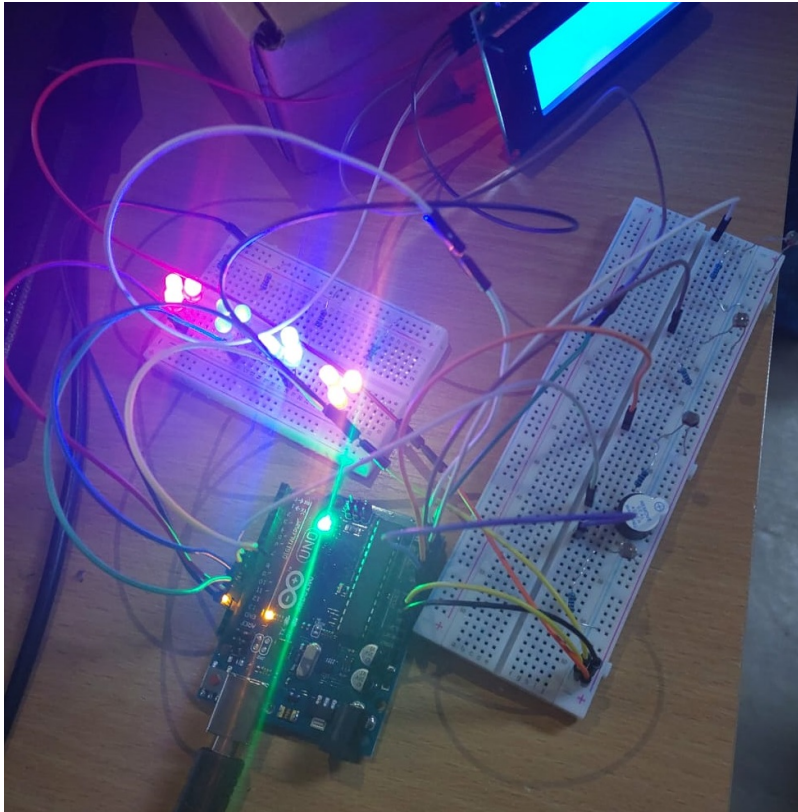
Jocul se bazează pe reproducerea unei secvențe aleatorii care este reprodusă de joc. Dacă utilizatorul reușește să își amintească secvența întocmai, trece la nivelul următor și îi crește scorul. În caz contrar, pierde și jocul se termină. Secvențele vor fi reprezentate de niște LED-uri care se aprind și se sting. Butoanele cu ajutorul cărora jucatorul va introduce secvența vor avea la bază câte o fotorezistență. Jocul are și un buzzer pentru a scoate niște sunete specifice unor diferite acțiuni: secvența a fost introdusă corect, respectiv secvența introdusă nu este corectă și jocul se termină. Utilizatorul poate să își vada scorul pe ecranul LCD, dar și timpul pe care îl mai are la dispoziție ca să insereze secvența.

Hardware Design

Lista de componente folosite::

- Arduino R3
- ecran LCD
- buzzer
- fotorezistente
- LED-uri
- rezistente (220 Ω , respectiv 1k Ω)
- modul I2C





Software Design

Pentru partea de software am folosit bibliotecile `Wire.h` si `LiquidCrystal_I2C.h` pentru a afisa output-ul pe ecranul LCD.

In functia **setup** am initializat LED-urile, nivelul de dificultate al jocului de la care se porneste, ecranul LCD, buzzerul si intreruperile pentru timer.

In functia **loop** verific constant daca jocul s-a terminat sau nu (daca player-ul a gresit secventa sau nu). In cazul in care s-a terminat, afisez un mesaj care sa instiinteze jucatorul de acest lucru, urmand ca dupa aceea jocul sa o ia de la capat, incepand de la nivelul cel mai usor. In caz contrar, se genereaza o secventa utilizand functia **generateSequence**, mai apoi secventa este aratata prin aprinderea LED-urilor corespunzatoare prin functia **showLEDSequence**, urmand ca in final, playerul sa vina cu input-ul pentru a reproduce secventa in exactitate, lucru care se observa in functia **getPlayerInput**.

Functia **getPlayerInput** se foloseste de ADC pentru a citi inputul de la fotorezistori (daca playerul tine degetul deasupra unui fotorezistor e ca si cum el ar apasa pe un "buton"). Daca input-ul este corect, jocul merge mai departe, cu un nivel de dificultate mai greu, adica o secventa mai mare de LED-uri care o sa se aprinda. In caz contrar, jocul se termina, iar jucatorul este instiintat de acest lucru cu un mesaj sugestiv pe ecranul LCD si cu o atentionare audio din partea buzzer-ului.

Rezultate Obținute

Rezultatele obtinute au fost cele la care ma asteptam. Jocul are rolul de a stimula memoria si de a aduce in prim plan o nevoie de baza, anume nevoia unei memorii antrenate, a unei "baze de date" bine pusa la punct care ne poate scoate din situatii dificile si ne poate imbunatati stilul de vaita.

Concluzii

In concluzie, consider ca proiectul a fost foarte fun de facut, foarte interesant sa lucrezi "hands on" cu componente hardware si ulterior sa lucrez si la software.

Download

Codul necesar rularii aplicatiei:

[cod_proiect_pm_volintiru.txt](#)

Jurnal

07.05.2023 - Creat pagina wiki.

15.05.2023 - Adaugat schema circuit Thinkercad.

18.05.2023 - Adaugat poza cu schema fizica a proiectului.

28.05.2024 - Adaugat informatii despre software si codul sursa al aplicatiei.

Bibliografie/Resurse

Pentru implementarea timer-ului: **Laboratorul 3: Timere. PWM**

Pentru convertirea inputului provenit de la fotorezistori: **Laboratorul 4: ADC**

Pentru folosirea LCD-ului: **Laboratorul 6: I2C**

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/memorygame>



Last update: **2023/05/28 21:05**