InteractiveBoard

Proiect realizat de **Ionescu Ioana**

Grupa **332CB**

Introducere

Acest proiect constă într-o consolă interactivă care integrează trei jocuri clasice: un joc de memorie bazat pe LED-uri colorate, Snake afișat pe un ecran și controlat cu butoane, precum și Piatra-Hârtie-Foarfecă, în care utilizatorul se confruntă cu un oponent virtual.

Scopul proiectului este de a crea un dispozitiv educativ și distractiv care combină componente de hardware și software într-un sistem embedded interactiv. Ideea a pornit de la dorința de a aduce jocurile clasice într-un format fizic, accesibil și portabil.

Consider că acest proiect este util deoarece dezvoltă abilități practice în domeniul microcontrolerelor, a interfeței hardware, afișajelor și programării logice. În plus, consola poate fi folosită ca instrument didactic în cadrul unor ateliere STEM pentru copii sau ca model demonstrativ pentru începători în domeniul electronicii.

Descriere generală

Proiectul este implementat pe un singur microcontroler (Arduino Uno), care gestionează toate componentele hardware și logica software pentru trei jocuri: Joc de Memorie, Snake și Piatra-Hârtie-Foarfecă. Componentele sunt partajate între module pentru optimizare hardware și software.

Pe ecranul LCD va fi o pagină principală de meniu, unde vei selecta unul dintre jocuri prin apăsarea butonului corespunzător jocului. După apăsare, intri în pagina de instrucțiuni a jocului, iar după câteva secunde începe jocul.

Joc de Memorie

Va avea un LED RGB și 3 butoane, fiecare pentru câte o culoare. Microcontrolerul se va ocupa de generarea și verificarea secvenței. Prima dată apare doar o culoare, iar pe măsură ce avansezi în joc, numărul de culori aprinse crește succesiv. Dacă ai greșit o culoare, ai pierdut și te întorci în pagina de meniu. În cazul în care te-ai plictisit de joc, poți apăsa pe un buton (butonul roșu), care te va trimite la pagina principală.

Modul Snake

Scopul jocului este să strângi 10 bucăți de mâncare ca să câștigi. Ai 4 butoane direcționale (sus, jos, stânga, dreapta). Microcontrolerul se ocupă de detectarea coliziunilor – dacă își atinge coada sau dacă iese din ecran (atinge marginea ecranului). Șarpele nu își poate modifica direcția de deplasare cu 180 de grade (ex: dacă merge în sus și apeși pe butonul pentru direcția jos, șarpele nu va merge în jos, ci va continua în sus). Poziția inițială a șarpelui este aleatorie, la fel și poziția mâncării. Dacă ai pierdut (ți-ai atins coada sau ai atins marginea ecranului), te întorci automat la pagina principală.

Modul Piatra-Hârtie-Foarfecă

Jucătorul va alege cu ce să joace apăsând pe unul din cele 3 butoane. După cum s-a mai menționat, jucătorul va concura cu calculatorul. Microcontrolerul se ocupă de alegerea aleatorie a opțiunii calculatorului, astfel încât nu vei ști niciodată dacă vei câștiga sau pierde. După ce jucătorul face o alegere, pe ecran va apărea alegerea jucătorului, alegerea calculatorului și apoi rezultatul: dacă ai câștigat, ai pierdut sau dacă este egalitate. Pentru a ieși din joc, jucătorul trebuie să apese pe butonul roșu.

Microcontroler Central – Arduino Uno

Se ocupă de logica fiecărui joc, afișaje, intrări de la butoane și controlul LED-urilor.

Schema bloc

Hardware Design

Listă de componente principale:

- Microcontroler Arduino UNO

- 8x Butoane

- LED RGB

- Ecran LCD

- Rezistențe pentru LED (1x 330 Ω și 2x 150 Ω)

- 2x Breadboard

- Fire tată-tată

- Fire U-shape

Schema circuit

Schemă electrică

Pini folosiți

Arduino are pini pentru alimentare (power), pini analogici și digitali. Am folosit pinii de alimentare și pinii digitali.

La pinii de alimentare am folosit: pinul de 5V pentru a alimenta circuitul, pinul de GND pentru a conecta masa circuitului.

La pinii digitali am folosit pinii de la D2 până la D13, astfel:

- **D2** → Buton roșu

- **D3** → Culoarea albastră a LED-ului

- **D4** → Buton 1

- **D5** → Culoarea verde a LED-ului

- **D6** → Culoarea roșie a LED-ului

- **D7** → Buton 2

- **D8** → Pinul RESET al ecranului

- **D9** → Pinul DC al ecranului

- **D10** → Pinul CS al ecranului

- **D11** → Pinul SDA al ecranului

- **D12** → Buton 3

- **D13** → Pinul SCK al ecranului

Am ales acești pini pentru culorile LED-ului deoarece permit utilizarea semnalului PWM.

Pentru ecranul LCD au mai rămas 3 pini, pe lângă cei conectați la Arduino:

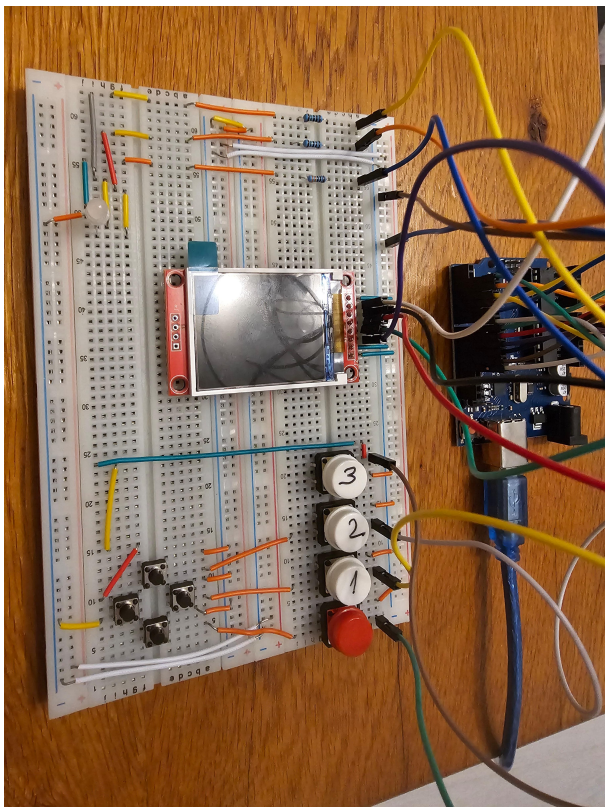
- **VCC** → conectat la +

- **LED** → conectat la +

- **GND** → conectat la -

a fost alimentat cu 5V de la microcontroler, iar – este legat la GND-ul microcontrolerului.

Poză circuit



Software Design

Mediu de Dezvoltare: **Arduino IDE**

Biblioteci:

- **Adafruit_ST7735** - control specific cu ecranul
- **Adafruit_GFX** - elemente grafice generale
- **SPI** - interacțiunea cu ecranul LCD

Descriere cod:

Am definit pinii pentru butoane, ecran și culorile LED-ului.

Am inițializat o variabilă booleană (isGame) care reține dacă te afli în jocul “Piatra, hârtie, foarfecă”.

În funcția **setup()** am inițializat generatorul de numere aleatoare, afișajul ecranului și am rotit ecranul cu 90 de grade pentru a se potrivi cu poziționarea lui pe breadboard. Am inițializat pinii pentru butoane ca fiind pini de intrare. Am inițializat și pinii pentru culorile LED-ului ca fiind pini de output și am apelat funcția care stinge LED-ul.

În funcția **loop()** am verificat starea variabilei isGame. Dacă nu mă aflam în joc, se afișa mesajul de pe meniul principal și verificam ce buton apăsam. Dacă apăsam un buton corespunzător celorlalte 2 jocuri, se apela funcția corespunzătoare jocului. Dacă apelam butonul cu “Piatra, hârtie, foarfecă”, schimbam valoarea variabilei, afișam mesajul din pagina de instrucțiuni și după se apela funcția

jocului.

Am avut și niște **funcții auxiliare**: afiseazaMesaj(const char* mesaj) care afișează pe ecran din stânga sus mesajul dat, afiseazaMesajMijloc(const char* mesaj) afișează pe mijlocul ecranului mesajul, stingeLED() setează pe 0 culorile LED-ului, aprindeCuloare(int culoare) setează pe 50 culoarea corespunzătoare, asteaptaCuloareUtilizator() care, în funcție de ce buton apăsăm, se aprinde culoarea corespunzătoare și se returnează valoarea pentru a fi verificată cu secvența, drawPixel(int x, int y, uint16_t color) colorează pixelul de la poziția x, y pentru jocul Snake, generateFood() generează aleator poziția mâncării pentru jocul Snake.

Pentru **jocul de memorie**, înainte de funcție am definit lungimea maximă a unei secvențe, un vector care reține secvența și nivelul curent. În funcție am afișat mesajele de instrucțiuni și, într-un while(true), am verificat dacă am apăsat butonul de întoarcere la meniu, am creat secvența cu culori random, am aprins culorile, de menționat că în timp ce afișez secvența mă pot întoarce în meniul principal, după am verificat dacă culorile introduse de jucător sunt la fel cu secvența, cresc nivelul și o iau de la capăt.

În funcția **jocPiatraHartieFoarfeca()** am văzut ce a ales jucătorul, reținând într-o variabilă alegerea, am afișat mesajul cu alegerea jucătorului, am făcut și alegerea calculatorului random și am afișat mesajul. Am verificat cele două variabile în ce caz se află și am afișat un mesaj corespunzător.

Pentru **Snake**, am definit înainte de funcție o lungime maximă a șarpelui, o structură de tip point, am definit șarpele ca fiind un vector de puncte, lungimea inițială a șarpelui, am definit și mâncarea ca un punct, direcția șarpelui și scorul. În funcție am afișat instrucțiunile, am generat poziția de start a șarpelui și am generat mâncarea. Apoi, într-un while(true), am citit direcția dată de la butoane, am blocat întoarcerea directă în sens opus, am mutat tot corpul șarpelui, am verificat coliziunile cu marginea ecranului sau cu propriul corp, verific dacă șarpele a mâncat mâncarea și desenez totul pe ecran.

Concluzii

Un proiect frumos și interesant de realizat. Datorită lui, mi-aș dori să mai realizez astfel de proiecte folosind microprocesoare.

Export to PDF