Nim pe LCD

Introducere

Proiectul meu consta in realizarea jocului Nim pe o placuta ATMega328P, folosind ca display un LCD ST7735S. Nim reprezinta un joc matematic in care doi jucatori iau cu randul obiecte din multimi diferite. Pentru acest proiect am ales varianta Misere a jocului, in care pierzator este cel care ia ultimul obiect din joc, indiferent de multime.
Reguli joc: pe rand, fiecare jucator poate lua orice numar de elemente dintr-o singura multime. Nu se pot lua in aceeasi tura elemente din doua sau mai multe multimi. Jocul se termina doar atunci cand nu mai exista elemente de luat, caz in care jucatorul care a facut ultima mutare pierde.
Exemplu de joc: 2 multimi A si B cu 2 elemente fiecare, 2 jucatori, Alice si Bob.

1. stare joc: A 2 elemente (A-2), B 2 elemente (B-2)
2. Alice ia din A un element, stare joc: A-1 B-2
3. Bob ia din B toate elementele, stare joc: A-1 B-0
4. Alice ia din A un element, stare joc: A-0 B-0

Pentru ca Alice a luat ultimul element, ea a pierdut.

Scopul proiectului este unul pur recreational, realizat pentru amuzamentul utilizatorului.

Descriere generală

Pentru acest proiect, jocul are 4 multimi a cate 7 elemente fiecare. Voi utiliza 4 butoane, unul pentru fiecare heap, si un buton care sa marcheze confirmarea actiunii si trecerea la jucatorul urmator. Fiind doar un joc pe display, nu sunt necesare multe module pentru a asigura buna functionare a acestuia, importanta consta in inregistrarea inputului jucatorului si afisarea pe ecran a actiunii.

Hardware Design

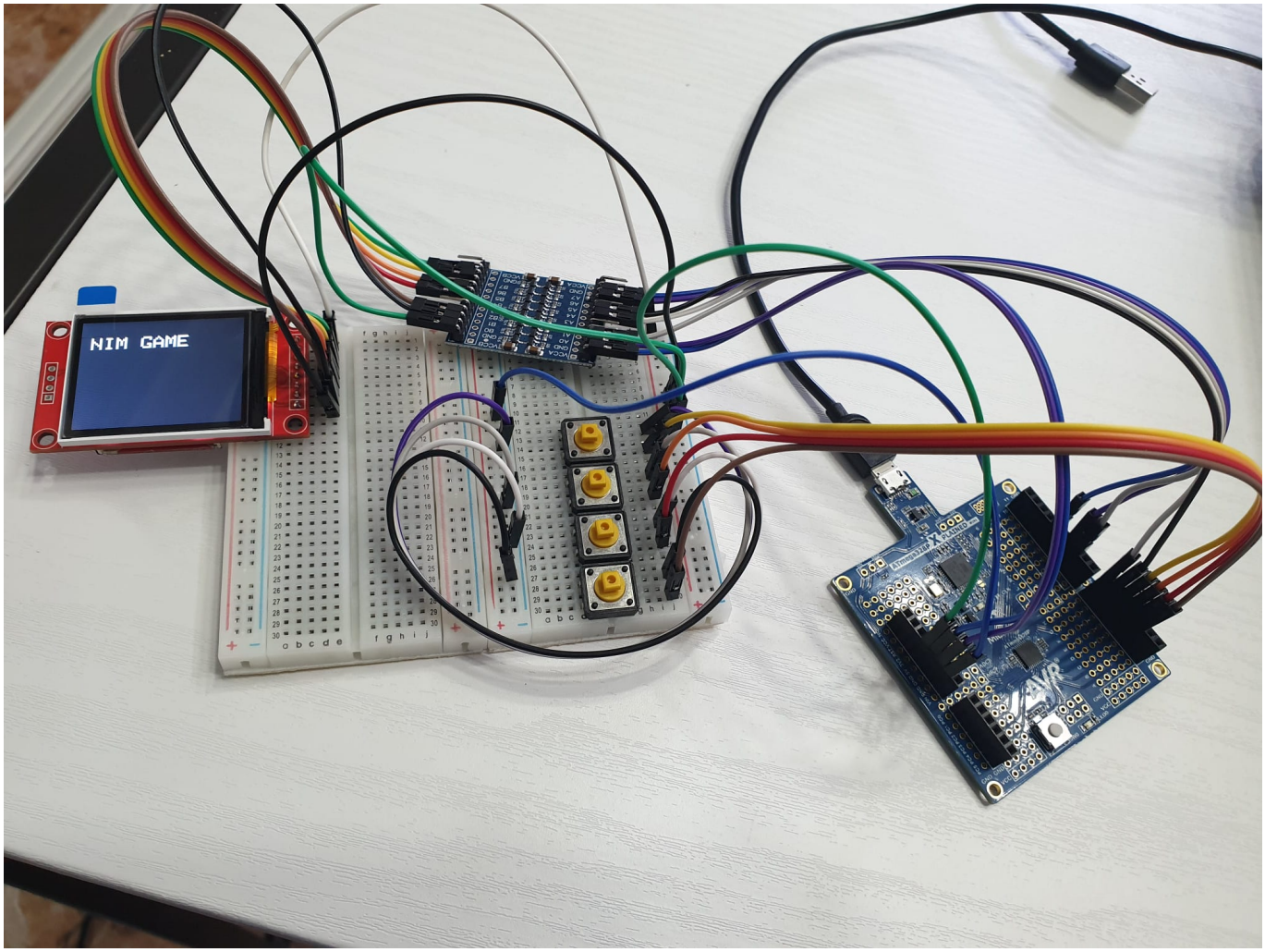
Lista componente

- Microcontroller ATMega328P Mini Xplained
- 4 + 1 butoane pentru input
- Display LCD 1.8 TFT ST7735S cu SPI
- cablu USB pentru alimentare 5V

- breadboard
- level shifter bidirectional 3V3 ↔ 5V

Pini utilizati:

Pin	Rol	Componenta
PB7	GPIO	Buton master
PD2	GPIO	Buton heap 1
PD3	GPIO	Buton heap 2
PD4	GPIO	Buton heap 3
PD5	GPIO	Buton heap 4
PB2	SPI	CS
PD7	GPIO	Reset
PD6	GPIO	DC
PB3	SPI	SDA
PB5	SPI	SCK
GND	Ground	
GND	Ground	
5V	Power	
3V3	Power	



Software Design

Proiectul va fi dezvoltat cu ajutorul **PlatformIO** care este integrat in MS Visual Studio Code, folosind avr-gcc.
Modulele software ale proiectului sunt:

- Modulul pentru timere: utilizat pentru a numara milisecundele trecute de la pornire
- Modulul pentru logica jocului: va avea logica botului si tratarea inputului jucatorului. De asemenea, verifica regulile jocului la fiecare mutare
- Modulul SPI: esential, permite comunicarea cu display-ul. Este modulul principal al proiectului
- Modulul pentru intreruperi: utilizat pentru setarea flag-urilor privind apasatul butoanelor
- Modulul de USART, folosit doar pentru debugging.

Functia main are rolul de initializare a modulelor si rularea unui loop infinit care sa inglobeze toate rundele care sunt jucate. Asta inseamna ca exista un loop care numara rundele castigate sau pierdute, fara a se ocupa de logica jocului. in acest loop exista game loop-ul propriu zis, care verifica constant starea jocului si determina castigatorul.

Logica jocului este urmatoarea:

1. cpu-ul mereu face prima miscare. Acesta este singurul mod in care jucatorul are un set de miscari cu care poate castiga
2. se verifica daca au mai ramas obiecte de luat
3. jucatorului ii este permisa mutarea
4. se verifica daca au mai ramas obiecte de luat sau daca a expirat timpul
5. se reia de la primul punct

Jocul este reprezentat in cod de un struct care contine numarul total obiecte si numarul de obiecte din fiecare heap.

LCD-ul este modul prin care jocul comunica informatie jucatorului. Exista trei ecrane principale: cel de start joc, jocul propriu zis, si ecranul de win/lose, alaturi de numarul de runde castigate/pierdute. Dintre acestea, cel mai complex este ecranul in timpul jocului, deoarece sunt randate multimile, numarul lor, timer-ul jucatorului si mesajul de play/wait. Pentru ca nu sunt modificate parti ale ecranului foarte des, nu exista un framebuffer care este trimis LCD-ul la fiecare frame, ci portiuni sunt updatate direct. Functiile principale sunt

- render_win()
- render_lost()
- draw_heap()
- draw_turn_mark()
- render_time()

Toate aceste functii fac apel la bibliotecile pentru SPI Adafruit GFX library si Adafruit ST7735 and ST7789 library

Intreruperile sunt in principal folosite pentru debouncing-ul butoanelor. Acestea se impart in doua

categorii: butoanele pentru heap-uri si butonul principal. Fiecare din aceste categorii au routine diferite si registre de flaguri diferite. Pentru butonul principal este reprezentat de 2 valori boolene, iar restul butoanelor sunt reprezentate toate pe 2 bytes. Starile lor sunt citite si modificate cu ajutorul unor masti.

Rezultate Obţinute

Aici este un scurt demo al proiectului: demo_proiect.zip

Concluzii

Pentru partea de hardware, singurul neajuns a fost necesitatea level shifter-ului pentru trecerea de la 5V (pinii mcu) si 3V3 (pinii LCD), desi rezolvarea acesteia nu a prezentat o problema.

Pentru software, partea cea mai voluminoasa a proiectului, comunicarea SPI, este rezolvata de bibiliotecile Adafruit. De asemenea, codul pentru miscarea cpu-ului a necesitat putin mai multa atentie pentru a ma asigura ca sunt acoperite toate cazurile de exceptie, reusind astfel sa il fac destul de greu de invins.

Pentru viitor, imi doresc sa imi fac propriile fire de conexiune pentru un aspect ingrijit, iar pe partea de software as vrea sa adaug mai multe nivele de dificultate jocului, prin adaugarea unor “greseli” cu o anumita probabilitate.

Sunt multumit ca am reusit sa fac un proiect hardware si software de la 0 cu ajutorul cunostiintelor dobandite in cadrul cursului si laboratorului de PM.

Download

nimlcd.zip
https://github.com/GionutN/ATmega328P-Nim

Jurnal

- 05/04/2026 Initializarea paginii proiectului, completare “Introducere” si “Descriere generela”
- 05/06/2026 Adaugare diagrame block si schita electrica
- 05/16/2026 Finalizare hardware
- 05/22/2026 Finalizare software

Bibliografie/Resurse

Resurse Hardware
Pentru realizarea diagramei am utilizat EasyEDA. Lista footprinturi piese:

- Microcontroller - TQFP-32_L7.0-W7.0-P0.80-LS9.0-BL
- Butoane - SW-SMD_4P-L6.0-W6.0-P4.50-LS9.2
- Level shifter - MPN-TH-20_L28.0-W14.0-P2.54-BL
- LCD - LCD-TH_8P-P2.54-L56.0-W34.0_1.8-TFT

Resurse software

- Pentru realizarea diagramei bloc am utilizat PlantUML.
- Adafruit GFX library - https://github.com/adafruit/Adafruit-GFX-Library
- Adafruit ST7735 and ST7789 library - https://github.com/adafruit/Adafruit-ST7735-Library

Export to PDF

From:
https://ocw.cs.pub.ro/courses/ - **CS Open CourseWare**

Permanent link:
https://ocw.cs.pub.ro/courses/pm/prj2026/ciprian.popescu0411/ionut.nicolae2009

Last update: **2026/05/22 14:22**