

Sudoku 9x9

Introducere

Nume student: Dinu Marian-Gabriel

Grupa: 333CD

Fiind pasionat de jocuri care solicită creierul, dar și de programare și inginerie, m-am întrebat “Ce aș putea face cu informațiile acumulate până acum?” Răspunsul? Să le pun în practică realizând un proiect ce are la bază un joc pe care îl cunosc prea bine și căruia îi pot aduce îmbunătățiri, anume **SUDOKU**.

Sudoku este un joc de tip puzzle ce presupune completarea a 9 cadrane, fiecare cu câte 9 căsuțe, cu cifre de la 1 la 9, astfel încât pe linie, coloană și în cadran cifrele să nu se repete. Inițial cadranele nu sunt goale, fiind date câteva valori, iar în funcție de acestea trebuie completate restul de cifre lipsă.

În mod normal **Sudoku** se joacă pe hârtie, alături de un instrument de scris pentru completare, dar acest proiect vrea să ducă acest joc mai departe, anume în **lumea digitală**.

Îmbunătățiri față de jocul original:

- Afișaj pe ecran LCD
- Localizare poziție prin schimbarea culorii cifrei
- Modificare poziție cu ajutorul unui joystick, iar a valorii cu ajutorul unor butoane
- Timer pentru a vedea timpul de rezolvare
- Muzică ambientală relaxantă pentru a scăpa de stresul indus de joc

Consider că această nouă modalitate de a vedea acest joc îi va face pe cei care nu l-au încercat încă să îi dea o șansă, astfel ajungând să **petreacă timpul liber** într-un mod **constructiv**, stimulând **logica, memoria și răbdarea**. [External Link](#)

Descriere generală

Interacțiunea componentelor:

- La baza proiectului se găsește un **Arduino UNO**, care primește și transmite mai departe datele de la componentele adiacente

- Pentru afișarea jocului, plăcuța de Arduino transmite date către un **LCD** prin intermediul protocolului **SPI**
- Melodiile, precum și puzzle-le de Sudoku, sunt setate în interiorul codului, dar pot fi alese cu ajutorul butoanelor
- Semnalele sonore sunt transmise către un **amplificator**, ce permite schimbarea volumului prin intermediul unui **potențiomtru**
- Sunetul este redat cu ajutorul a două **spekere** conectate la **amplificator**
- **Butoanele** reprezintă acțiuni diverse, semnalele generate de acestea fiind considerate drept **întreruperi** de către Arduino și vor fi rezolvate ulterior
- Folosind un **timer intern**, Arduino va trimite către **LCD** valoarea încât să îți dai seama cât timp a trecut de la începerea puzzle-ului

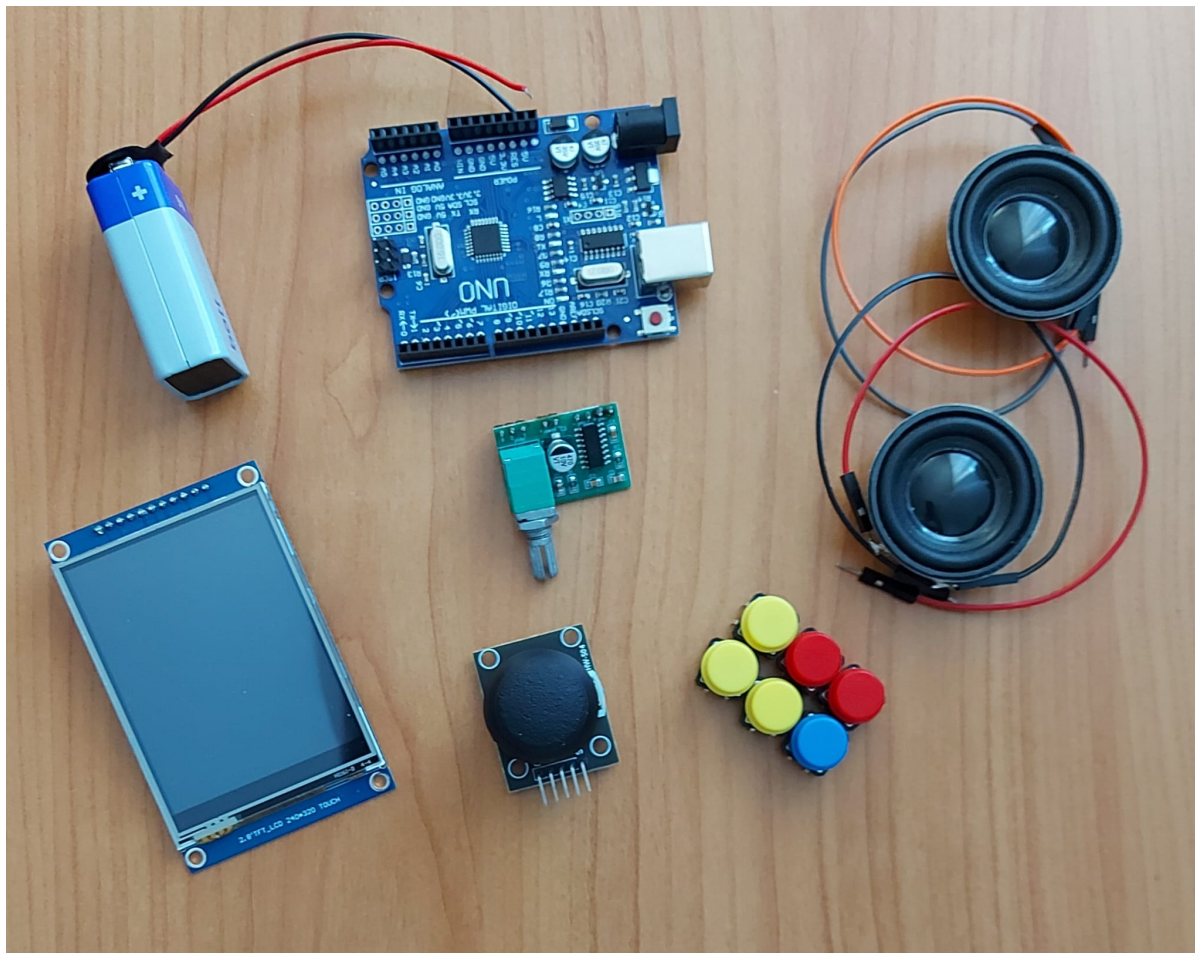
Schema bloc



Hardware Design

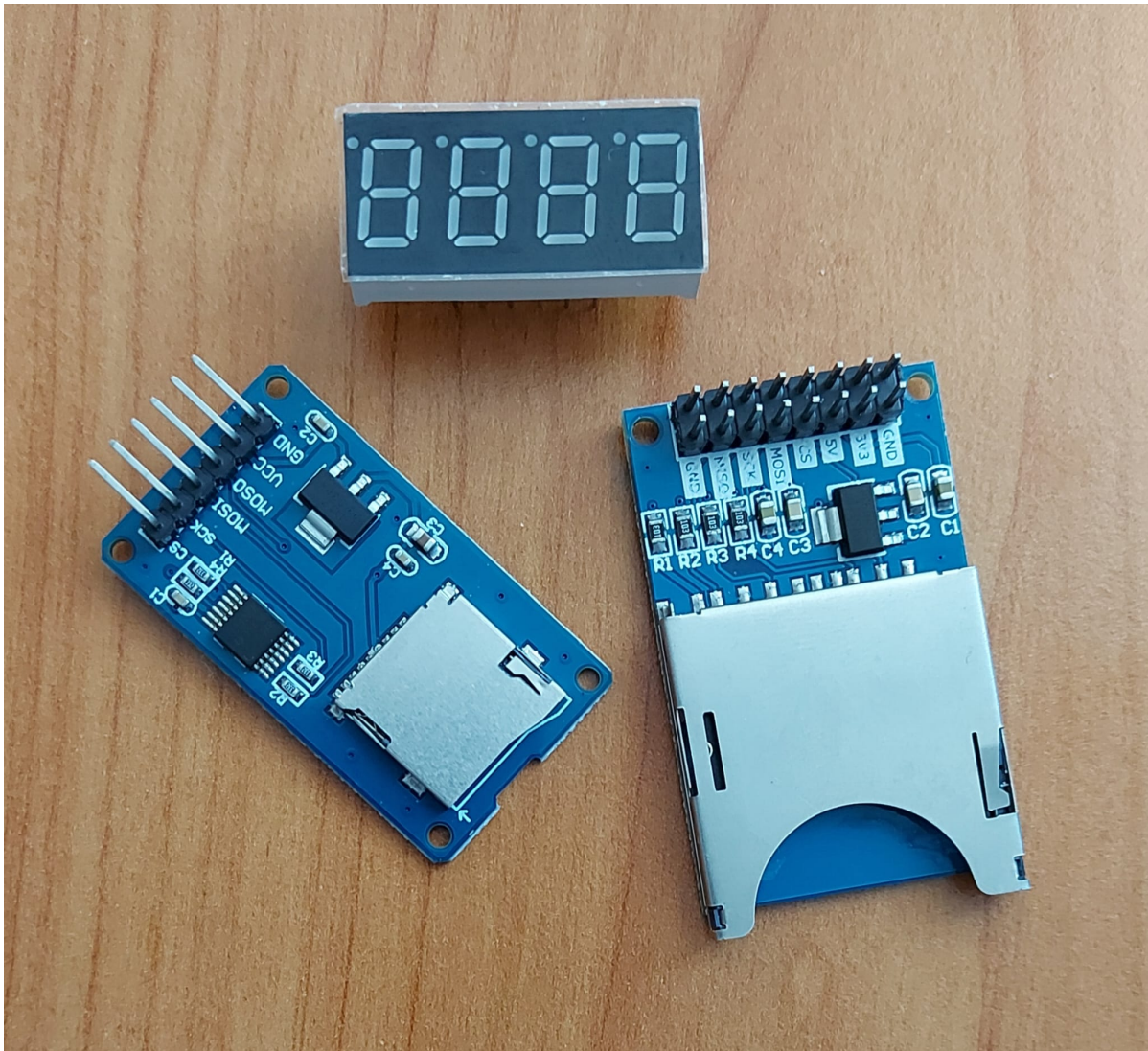
Listă componente:

- Arduino UNO cu ATmega328P
- Modul LCD SPI de 2.8" (240×320) cu driver ILI9341
- 2 speakers 4 ohm, 3 W
- Modul amplificator audio PAM8403, cu potențiomtru
- Modul joystick cu două axe XY HW-504
- butoane, baterii 9V, cabluri Dupont, rezistențe



Listă componente în plus:

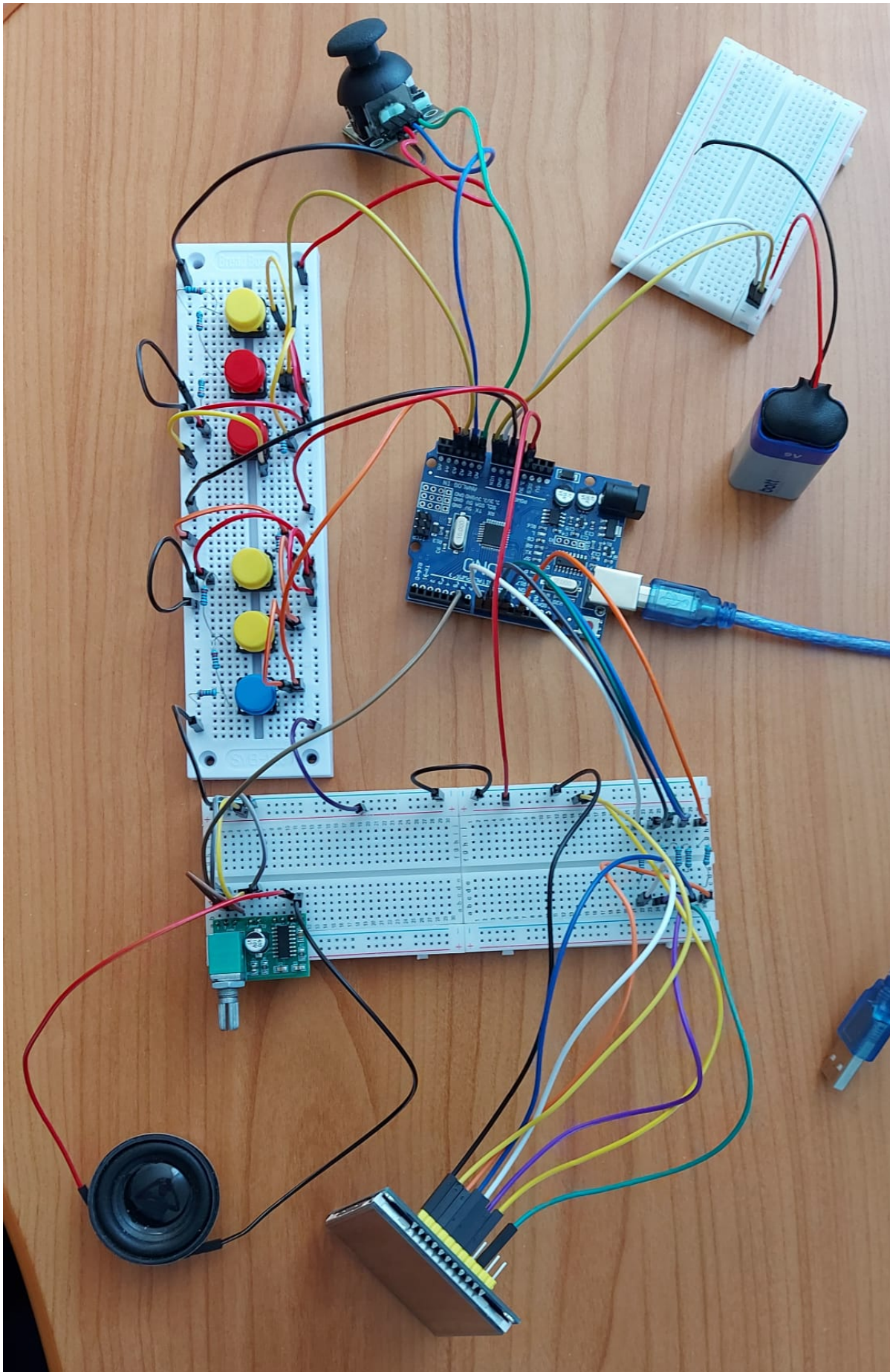
- Afișaj cu LED roșu cu 4 cifre cu anod comun de 0.36" F3641BS
- Modul cititor card MicroSD/modul cititor card SD



De ce nu sunt adăugate?

Datorită numărului mare de pini necesari pentru afișor și, respectiv, modulele de card SD care nu funcționau mereu sau incompatibilitatea bibliotecilor LCD-ului și cardului SD, am decis să nu mai adaug aceste componente în circuit. Probabil în viitor o să încerc să le adaug.

Circuit complet



Componentele vor fi legate în felul următor:

Arduino -> LCD

```
3.3V ----- VCC_LCD
3.3V ----- BLK_LCD (LED)
GND ----- GND_LCD
13 ---- 10k ---- CLK_LCD
11 ---- 10k ---- MOSI_LCD
10 ---- 10k ---- CS1_LCD
9 ---- 10k ---- DC_LCD
8 ---- 10k ---- RESET_LCD
```

CLK, MISO, MOSI și CS sunt necesare pentru a putea folosi protocolul SPI.

Intrările LCD-ului au nevoie de 3.3V, așa că am folosit rezistențe de 10k pentru a limita output-ul din Arduino.

Arduino -> Amplificator -> Speakere

```
5V ----- POWER +
GND ----- POWER -
GND ----- INPUT G
6 ----- INPUT R
6 ----- INPUT L
          OUT R+ ----- SPEAKER1+
          OUT R- ----- SPEAKER1-
          OUT L+ ----- SPEAKER2+
          OUT L- ----- SPEAKER2-
```

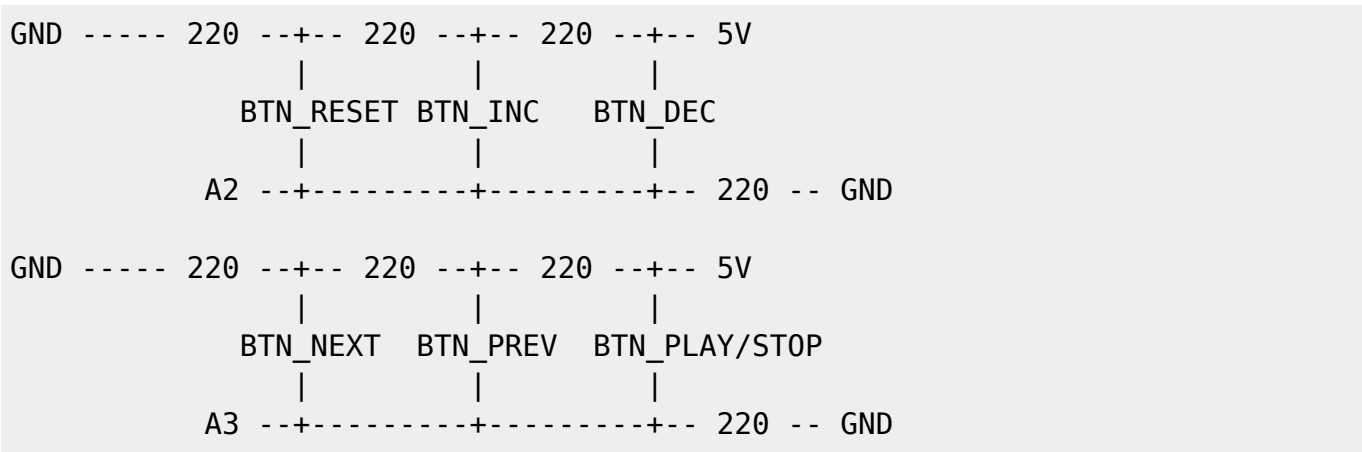
Arduino va trimite pe pinul 6 semnalul audio către amplificator. Cu ajutorul potențiometrului poate fi modificat volumul, iar apoi acesta va fi redat pe cele 2 canale, altfel spus cele 2 speakere. Pentru că cele 2 speakere au nevoie de o putere destul de mare (3W), o să folosesc doar un singur speaker.

Arduino -> Joystick

```
5V ----- 5V_Joystick
GND ----- GND_Joystick
A0 ----- VRx_Joystick
A1 ----- VRy_Joystick
```

Arduino va primi de la joystick poziția acestuia prin pinii A0 și A1 (ADC), apoi în funcție de aceasta ne deplasăm în SUDOKU.

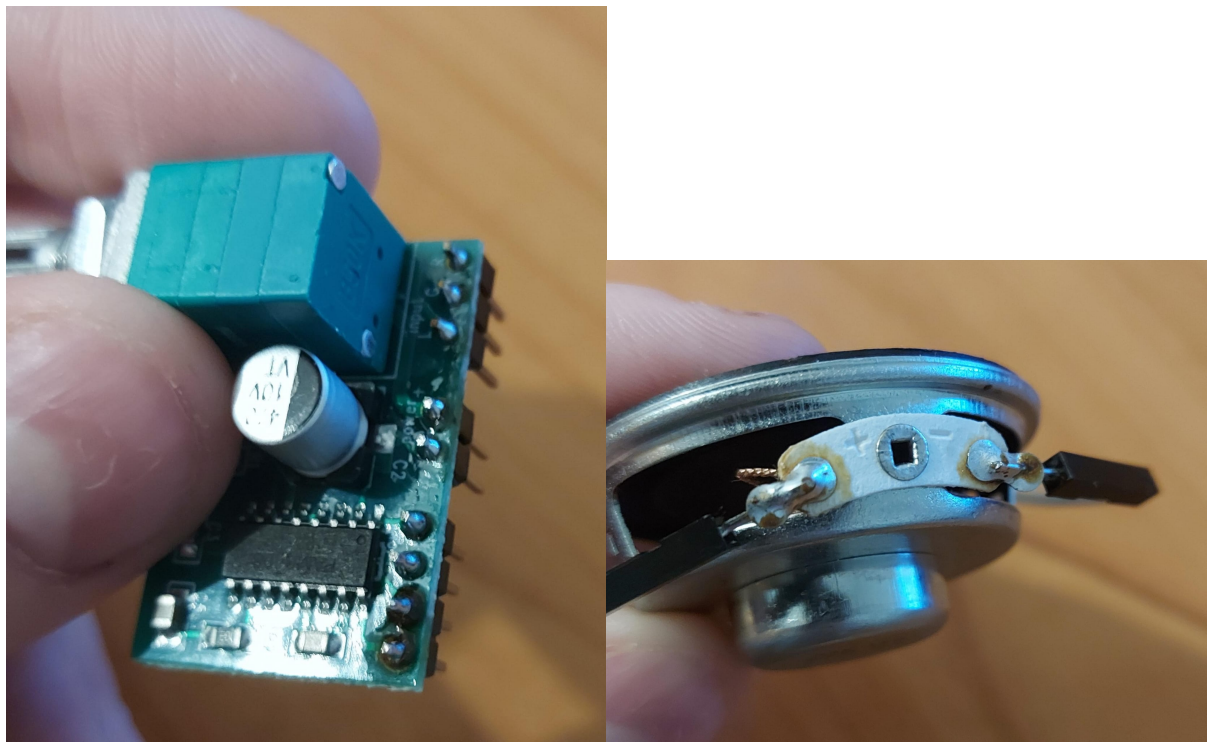
Arduino -> Butoane



Fiecare buton are o cădere de tensiune diferită (divizoare de tensiune). Semnalul ajuns la A2/A3 v-a fi descifrat (ADC), iar în funcție de valoarea primită se va realiza o acțiune (descrise prin denumirile lor). Rezistențele din partea de jos a BTN_DEC/BTN_PLAY/STOP trag la masă voltajul când butoanele nu sunt apăstate.

Soldering

Deoarece amplificatorul si speakerele nu aveau legaturi, in cadrul laboratoarelor practice am reusit impreuna cu laborantul sa lipesc pini si cabluri incat sa pot face legatura cu celelalte componente ale proiectului.



Software Design

Mediu de dezvoltare: **Arduino IDE 2.3.2**

Librării utilizate:

- SPI (Pentru comunicarea cu LCD-ul)
- Adafruit_GFX (Afisare pe LCD)
- Adafruit_ILI9341 (Afisare pe LCD)
- cod pentru melodii care merg pe arduino [1] (imperialmarch și zeldatheme)

Jocul de SUDOKU este destul de simplu: pui cifre în căsuțe până când toate sunt completate și nu se repetă atât pe linie, coloană, cât și în cadranele mari.

0 rulare a codului ar trebui să aibă următorul comportament:

- * Afișare meniu principal
- * Afișare meniu selecție sudoku
- * Afișare sudoku ce urmează a fi rezolvat
- * Mișcarea joystick-ului te mută în sudoku
- * Apăsarea butoanelor RESET (R), INCREMENT (I) sau DECREMENT (D) aduce modificări asupra cifrei pe care ne aflăm
- * Apăsarea butoanelor NEXT (N), PREV (P), PLAY/STOP (P/S) aduce modificări asupra melodiilor redade prin speaker

Funcțiile necesare pentru realizarea proiectului ar fi:

- * funcție ce verifică dacă sudoku este rezolvat complet
- * funcție care redă melodie
- * funcție care afișează sudoku
- * funcție care analizează mișcarea joystick-ului și te mișcă în sudoku
- * funcție care analizează apăsarea butoanelor care modifică sudoku și execută comanda
- * funcție care analizează apăsarea butoanelor care modifică melodia și execută comanda
- * funcție care afișează timer pe LCD

Din lipsa timpului, nu am reușit să integrez și partea de timer, dar celelalte funcționalități există.

Explicații legate de cod

Afișarea pe LCD

- Funcțiile oferite de librăriile Adafruit au ușurat partea de afișare text pe LCD, singura dificultate fiind calculul pozițiilor unde ar trebui să fie afișat textul
- Când afișez Sudoku pe LCD țin cont de poziția pe care mă aflu și de cifrele care erau de la bun început. Astfel, cifrele care erau deja vor fi afișate cu albastru, cifrele care pot fi modificate cu alb, cele pe care mă aflu cu roșu, iar dacă mă aflu pe o cifră care era deja de la început cu roz
- Mai mult, până să ajungi la Sudoku propriu zis se afișează un meniu de unde începi și altul în care alegi ce Sudoku să completezi

Butoanele și Joystick-ul

- În cazul joystick-ului, coordonatele le-am citit de pe porturi ADC, apoi în funcție de acestea m-am deplasat în sudoku. Când am ajuns la unul dintre capete, am rămas pe loc
- Butoanele le-am multiplexat (folosind conceptul de divizor de tensiune) câte 3 pe un pin. Apăsând fiecare buton pe rând se poate observa că arduino citește o cădere de tensiune diferită (ADC), fapt ce nu putea fi observat dacă foloseam pini digitali.
- Există 2 tipuri de butoane: butoane care modifică o valoare în Sudoku, altele care modifică melodia redată
- Butoanele care modifică valori în Sudoku sunt de 3 tipuri: R care resetează la 0 valoarea, I care incrementează de la 0 la 9, iar D decrementează de la 9 la 0
- Butoanele care modifică melodia sunt de 3 tipuri: N care redă de la început următoarea melodie, P care redă de la început melodia anterioară și P/S care oprește sau pornește melodia

Verificarea completării Sudoku-lui

- Verificarea se face pe linii, coloane și cadrane
- Toate cele 3 tipuri de verificări au la bază un vector de apariții, care apoi este verificat, iar dacă nu se găsește cel puțin o apariție pentru fiecare cifră, înseamnă că linia/coloana/cadrantul nu este completat cum trebuie

Validare funcționalități:

Dificultățile cele mai mari au fost datorate legăturilor hardware care ori mai aveau nevoie de ceva (LCD-ul lucrează pe 3v3, caz în care a trebuit să bag rezistențe) ori le-am pus unde nu trebuia (citire pin greșit).

În cazul software, la butoane verificam interval diferit de cel real, iar la LCD poziție greșită de afișare datorată calculului eronat.

Pentru a determina intervalele bune pentru butoane, dar și pozițiile joystick-ului, am folosit seriala, altfel spus am afișat mesaje cu valorile citite pe pini, pe care apoi le-am folosit în cod.

În cazul LCD-ului am stat și am numărat pixelii afișați încât să pot determina pozițiile bune.

Demo funcționalități: <https://youtu.be/k0xR6hXrWwk>

Rezultate Obținute

În urma realizării proiectului am reușit să mă mai obișnuiesc cu mediul de dezvoltare Arduino și cu microprocesorul, dar mai ales am reușit să obțin un joc care îmi face plăcere să îl joc când mă plictisesc.

Afișajul LCD-ului este unul viu colorat, cifrele se văd bine și îți dai seama unde te afli în Sudoku.

Muzica este o noutate, care te face să nu te simți singur, fiind un joc de o persoană.

Combi-nația dintre LCD, butoane și joystick au transformat jocul original într-o variantă asemănătoare cu cel de pe telefoanele noastre.

Concluzii

Proiectul a fost interesant din punctul de vedere că am putut să punem în practică ceea ce am făcut în cadrul laboratoarelor, dar mai ales că am putut și interacționa fizic cu componentele.

Împletirea dintre hardware și software a dus la crearea unui produs unicat care mi-a făcut plăcere să îl realizez, în ciuda timpului care a trecut foarte rapid.

Mi-am demonstrat că pot să fac ceva cu informațiile dobândite pe parcursul acestor ani de facultate.

Dificultățile cele mai mari au apărut datorită compatibilității pieselor, care are fiecare nevoie de anumiți pini, astfel reducând numărul acestora, dar și în software când librăriile nu voiau să lucreze împreună (cardul SD cu LCD-ul).

Totalizând, experiența a adus o nuanță de noutate, pe lângă majoritatea temelor care presupun doar să scrii cod.

Download

Codul poate fi descărcat aici: [sudoku_9x9.zip](#)

Bibliografie/Resurse

Hardware:

Componentele au fost achiziționate de la:

- Optimus Digital
- Sigmanortec
- Clește

Software:

- [1] <https://github.com/robsoncouto/arduino-songs/blob/master>

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2024/fstancu/marian_gabriel.dinu



Last update: **2024/05/26 23:26**