Joc X SI 0

Introducere

Ce face proiectul:

Am realizat un joc de X și 0 (Tic-Tac-Toe) cu ajutorul unei plăci Arduino Uno, care poate fi jucat în două moduri:

Jucător vs. Jucător Jucător vs. Bot (automat) Jocul este afișat pe un ecran LCD TFT de 1.44'', iar controlul jocului se face prin intermediul a două butoane. Interfața este simplă, iar rezultatul partidei este afișat în timp real pe ecran.

Scopul proiectului:

Scopul a fost să îmbinăm cunoștințele de electronică și programare pentru a crea un joc interactiv, care să fie atât educativ, cât și distractiv. Am vrut să învățăm mai multe despre afișajele grafice, interacțiunea cu utilizatorul prin butoane și logica unui joc automatizat.

Ideea de la care am pornit:

Totul a început cu ideea de a crea un joc simplu, dar atractiv, folosind componente de bază Arduino. Am vrut să ne testăm creativitatea și să adăugăm și un mod cu inteligență artificială simplă (bot), pentru a face proiectul mai interesant și mai complex.

De ce credem că este util:

Pentru alții: este un proiect ideal pentru începători care doresc să învețe programare Arduino, folosirea unui ecran LCD, lucrul cu butoane și aplicarea logicii în jocuri. Pentru noi: ne-a ajutat să înțelegem mai bine cum interacționează componentele hardware cu codul, cum se gestionează interfețele grafice și cum se scrie o logică de joc, inclusiv pentru un bot.

Descriere generală

O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.

Descrierea componentelor:

1. Plăcuța Arduino Uno

Rol: Este creierul proiectului. Este un microcontroler programabil care citește intrări (ex: de la butoane) și controlează ieșiri (ex: LED-uri, motoare).

2. Fire mama-mama

Rol: Sunt folosite pentru a face conexiunile între componente și placa de dezvoltare (Arduino ↔ Breadboard ↔ Componente).

3. 2 rezistențe de 10 ohm

Rol: Limitează curentul care trece printr-un circuit. La 10 ohmi sunt destul de mici și pot fi folosite în testări simple sau pentru protecție minimă.

4. 2 pushbutton-uri

Rol: Sunt folosite ca dispozitive de intrare. Când apeși pe ele, trimit un semnal digital (HIGH/LOW) către Arduino.

5. Breadboard (placă de test)

Rol: Permite realizarea de conexiuni temporare între componente fără a fi nevoie de lipire.

Exemplu de schemă bloc: 🗵

Hardware Design

Componentele pentru proiect:

- 1. placuta Arduino Uno
- 2. fire mama-mama
- 3. 2 rezistente de 10 ohm
- 4. 2 pushbutton
- 5. un breadboard

Arduino UNO

```
    5V: Alimentare pentru componente externe, cum ar fi Display-ul TFT.
    GND: Comun pentru toate componentele, asigurând un potențial de
referință comun.
```

3) D2: Utilizat pentru a citi semnalul de la primul buton (Pushbutton).

4) D7: Utilizat pentru a citi semnalul de la al doilea buton (Pushbutton). 5) D8: Conectat la pinul A0/DC al Display-ului TFT pentru a controla modul de date/comandă. 6) D9: Conectat la pinul RESET al Display-ului TFT pentru a reseta display-ul. 7) D10: Conectat la pinul CS al Display-ului TFT pentru a selecta dispozitivul în comunicația SPI. 8) D11: Utilizat pentru a transmite date (MOSI) către Display-ul TFT. 9) D13: Utilizat pentru a transmite semnale de ceas (SCK) către Display-ul TFT. Display TFT LCD 1.44" 128×128 1) VCC: Alimentare de la Arduino (5V). 2) GND: Conectat la GND-ul Arduino. 3) CS: Conectat la D10 pe Arduino pentru a selecta display-ul în comunicatia SPI. 4) RESET: Conectat la D9 pe Arduino pentru a permite resetarea display-ului. 5) A0/DC: Conectat la D8 pe Arduino pentru a comuta între modul de date și modul de comandă. 6) SDA/MOSI: Conectat la D11 pe Arduino pentru a primi date. 7) SCK/CLK: Conectat la D13 pe Arduino pentru a primi semnale de ceas.

Pushbutton

 Pin 3 (out): Conectat la GND și la D2 pe Arduino pentru a citi starea butonului.
 Pin 1 (in): Conectat la D2 pe Arduino pentru a citi starea butonului.

Pushbutton

 Pin 3 (out): Conectat la GND și la D7 pe Arduino pentru a citi starea butonului.
 Pin 1 (in): Conectat la D7 pe Arduino pentru a citi starea butonului.

Schema electrica:

×

Software Design

Bibliotecile folosite:

1. <SPI.h> Motivație: Biblioteca SPI.h este necesară deoarece ecranul TFT utilizează protocolul de comunicare SPI (Serial Peripheral Interface) pentru transmiterea datelor.

Rol: Asigură comunicarea rapidă și eficientă între microcontroller (de exemplu, Arduino) și display.

2. <Adafruit_GFX.h> Motivație: Biblioteca Adafruit_GFX oferă o interfață grafică generală pentru desene, texte, forme și linii, compatibilă cu o gamă largă de display-uri.

Rol: Oferă funcții pentru desenarea de linii, dreptunghiuri, texte etc. Abstractizează partea grafică, permițând o codare simplificată și portabilă.

Exemplu de utilizare: drawRect, drawLine, setCursor, print, setTextColor, etc.

3. <Adafruit_ST7735.h> Motivație: Această bibliotecă este specifică pentru controlul display-urilor TFT care folosesc driverul ST7735, foarte popular în proiectele cu Arduino.

Rol: Inițializează ecranul TFT. Controlează afișajul color și dimensiunile acestuia. Se integrează perfect cu Adafruit_GFX pentru redarea conținutului grafic.

void drawStartScreen() Ce face: Afișează pe ecran meniul de start, unde utilizatorul poate alege între "Vs Bot" și "Vs Player".

Detalii:

Evidențiază opțiunea selectată cu un chenar roșu. Afișează instrucțiuni pentru butoane: Select = schimbă opțiunea, OK = start joc.

void drawWinnerScreen(int winner) Ce face: Afișează un ecran de final cu rezultatul jocului: "X câștigă", "O câștigă" sau "Remiză".

Detalii: Schimbă culoarea textului în verde pentru câștigător. Afișează instrucțiuni pentru a relua jocul.

void drawBoard() Ce face: Desenează tabla de joc (3×3) și plasează X sau O în funcție de mutările făcute.

Detalii: Desenează liniile care împart ecranul în 9 pătrate. Evidențiază pătratul selectat cu un chenar galben. Desenează "X" cu roșu și "O" cu albastru în pătratele ocupate.

int checkWinner() Ce face: Verifică dacă există un câștigător sau remiză.

Returnează: 1 \rightarrow dacă X a câștigat 2 \rightarrow dacă O a câștigat 3 \rightarrow dacă e remiză (tabla e plină) 0 \rightarrow jocul continuă

void botMove() Ce face: Realizează o mutare automată pentru bot (jucătorul O).

Strategie: Verifică dacă botul poate câștiga \rightarrow joacă acolo. Dacă nu, verifică dacă trebuie să blocheze X \rightarrow blochează. Dacă nu, joacă pe poziții preferate (mijloc, colțuri etc). Include: feedback sonor și redesenarea tablei.

void resetGame() Ce face: Resetează toate variabilele pentru a începe un nou joc.

Detalii: Șterge tabla (board[]), resetează jucătorul și indicatorii (gameOver, winner). Apelează drawBoard() pentru a afișa o tablă goală.

void setup() Ce face: Este apelată o singură dată la pornirea plăcii Arduino.

Inițializează: Pinurile pentru butoane și buzzere. Ecranul TFT (rotire, modul de inițializare). Afișează

ecranul de start (drawStartScreen()).

void loop() Ce face: Este funcția care rulează în buclă continuu și gestionează logica principală a jocului.

Structură logică: Dacă jocul nu a început: permite selectarea modului de joc. Dacă jocul s-a terminat: permite revenirea la ecranul de start. Dacă jocul e activ: Citire buton SELECT \rightarrow mută cursorul pe tablă. Citire buton OK \rightarrow plasează X sau O. Verifică dacă există câștigător sau remiză. Dacă e modul Vs Bot și e rândul lui \rightarrow apelează botMove().

În proiectul Tic-Tac-Toe cu ecran TFT și butoane, optimizările au fost realizate pentru a îmbunătăți: -Viteza de reacție și claritatea interfeței -Ușurința în utilizare -Citirea clară a codului -Reducerea resurselor folosite (memorie, sunete inutile etc.

Rezultate Obținute

Rezultate obținute in proiect. https://youtube.com/shorts/mDymTbDc0fk?si=P2IqIXAzB6K4TqPN

Schelet de cod pm1.zip

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

Export to PDF

×

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

Permanent link: http://ocw.cs.pub.ro/courses/pm/prj2025/czlatea/andrei.ignat

Last update: 2025/05/25 17:24