

Pac Man

- **Autor:** Dinu Andreea Sabina
- **Grupa:** 332CB

Introducere

Pac-Man (パッパマン Pakkuman?) este un joc arcade dezvoltat de compania Namco. A fost lansat în 1980, și a devenit imediat foarte popular.

Proiectul meu reprezintă o versiune mai simplă a jocului clasic și poate fi controlat prin intermediul unui joystick și a unui ecran LCD. Scopul jocului este de a controla personajul principal pentru a evita fantomele și a colecta punctele de pe harta.

Consider că acest proiect este util pentru alții deoarece oferă oportunitatea de a construi un joc interactiv și educativ utilizând componente electronice și o placa Arduino. De asemenea, poate fi util pentru cei care doresc să învețe programarea Arduino sau să își îmbunătățească abilitățile de programare.

Chiar dacă jocul are 256 de stagii, ultimul nivel nu poate fi terminat din cauza unei probleme apărute la crearea jocului.

Descriere generală

Arduino Uno este placa de dezvoltare care controlează toate aspectele jocului și comunică cu celelalte componente pentru a afișa jocul, pentru a citi valorile de la joystick și pentru a produce efecte de sunet. Joystick-ul este conectat la pinii analogici ai plăcii Arduino pentru a citi poziția jucătorului (valorile de pe axele X și Y), iar ecranul LCD este conectat la pinii digitali ai plăcii Arduino pentru a afișa jocul, scorul și nivelul.

Buzzer-ul este conectat la un pin digital pentru a crea efecte de sunet în momente importante ale jocului, cum ar fi atunci când jucătorul reușește să colecteze toate campurile de pe harta. Butonul de reset este conectat la pinul digital pentru a reporni jocul.

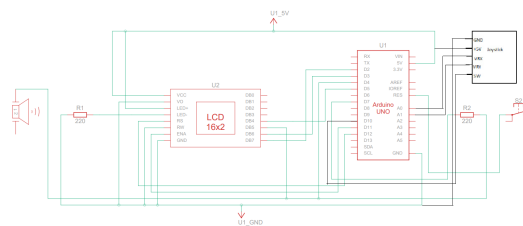
Schema bloc



Hardware Design

Componente necesare

- Placa Arduino Uno
- Joystick
- Ecran LCD
- Breadboard (pentru a face prototipul)
- Cabluri jumper (pentru a conecta componentele)
- Rezistențe (pentru a proteja componentele)
- Buzzer (pentru efecte de sunet)
- Buton de reset (pentru a reporni jocul)



Software Design

Jocul folosește următoarele tehnologii și componente:

1. Biblioteca LiquidCrystal_I2C pentru a controla afișajul LCD cu interfață I2C.
2. Biblioteca EEPROM pentru a salva și citi scorul maxim în memoria permanentă a microcontrollerului.
3. Un buzzer conectat la pinul 9 al Arduino pentru a reda sunete și melodii.
4. Un joystick conectat la pinii A0 și A1 pentru a controla mișcarea lui Pac-Man.
5. Un senzor ADC (conversor analog-digital) pentru a citi valorile analogice ale joystickului.

La începutul jocului, este redată o melodie de start, urmata de un mesaj de Welcome si instructiuni pentru joc. Scopul jocului este să colectezi inimioarele reprezentate pe ecran, evitând fantomele care sunt afisate pe exran. Fantomele pot fi evitate de către Pac-Man, iar când este atins de o fantomă, jocul se încheie cu un semnal sonor si cu afisarea scorului curent, dar si a highscore-ului. Jucătorul poate controla mișcarea lui Pac-Man(sus/jos) folosind joystickul.



În implementarea acestui joc, se folosesc întreruperi pentru a detecta mișcarea joystickului. În fiecare ciclu de joc, se citește valoarea ADC a joystickului pentru a determina poziția acestuia. Dacă joystickul este mișcat, se declanșează o întrerupere care semnalează că joystickul a fost mutat. Apoi, în funcția loop(), se verifică starea jocului și se execută acțiunile corespunzătoare.

Jocul are trei stări principale:

-STATE_INTRO: Starea de introducere a jocului. În această stare, se afișează un mesaj de bun venit și instrucțiuni pentru jucător. De asemenea, se redă o melodie de început a jocului. -STATE_PLAY: Starea

de joc. În această stare, jucătorul controlează mișcarea lui Pac-Man pentru a colecta inimioarele și a evita fantomele. Scorul este actualizat în timp real pe afișajul LCD. -STATE_GAMEOVER: Starea de sfârșit a jocului. În această stare, se afișează un mesaj de game over și se redă o melodie de sfârșit a jocului.



Jocul utilizează o serie de structuri și variabile pentru a gestiona sprite-urile (personaje) și starea jocului. De exemplu, există o structură pentru fiecare sprite care conține informații despre poziția și tipul sprite-ului.



În codul furnizat, se folosesc întreruperile pentru a trata mișcarea joystick-ului. Prin intermediul întreruperii PCINT1 (Pin Change Interrupt 1), se detectează schimbările de stare pe pinul A1 (PC1). Atunci când joystick-ul este mutat în sus sau în jos, se declanșează o întrerupere și se setează variabila joystickMoved pe true. Această variabilă este ulterior verificată în funcția loop() pentru a determina acțiunea corespunzătoare.



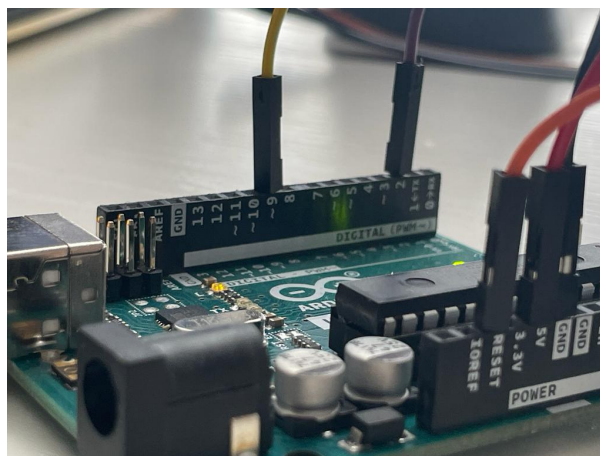
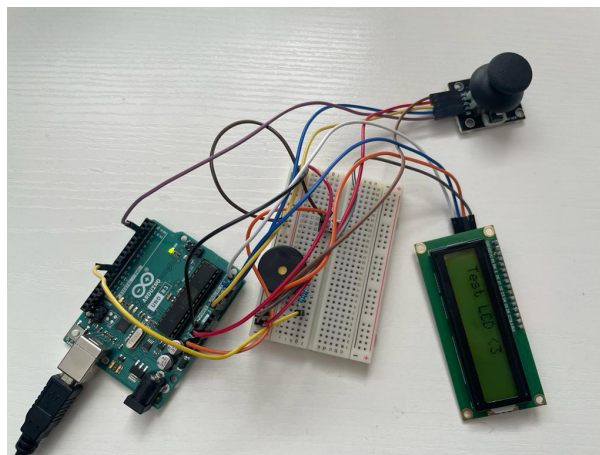
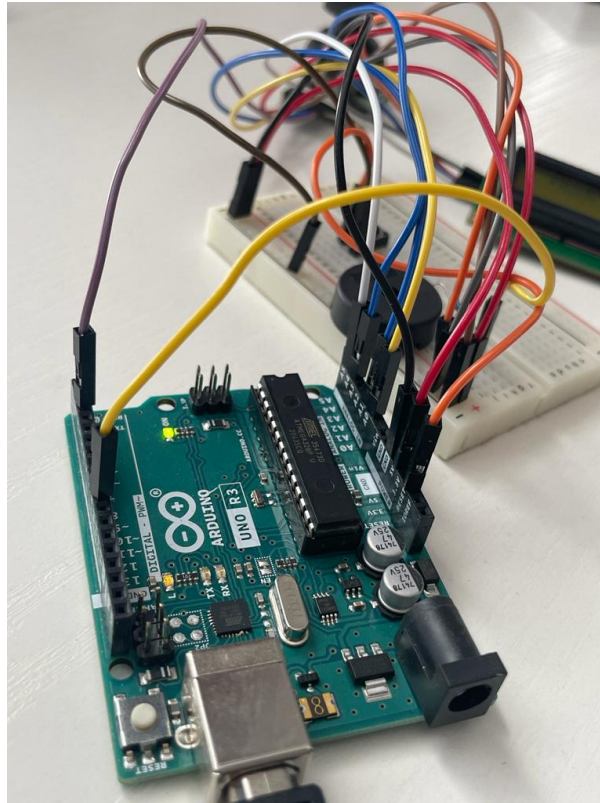
În funcția setup(), se configurează întreruperea și se activează ADC (Analog-to-Digital Converter). Pinul A1 (PC1) este setat ca pin de intrare, iar ADMUX (ADC Multiplexer Selection Register) și ADCSRA (ADC Control and Status Register A) sunt configurați corespunzător. În funcția loop(), se realizează citirea valorii analogice de pe pinul A1 folosind ADC. Această valoare reprezintă poziția joystick-ului pe axa Y și este stocată în variabila yValue. Apoi, întreruperile sunt dezactivate temporar (cli()) pentru a evita modificări ale valorii în timpul verificărilor.

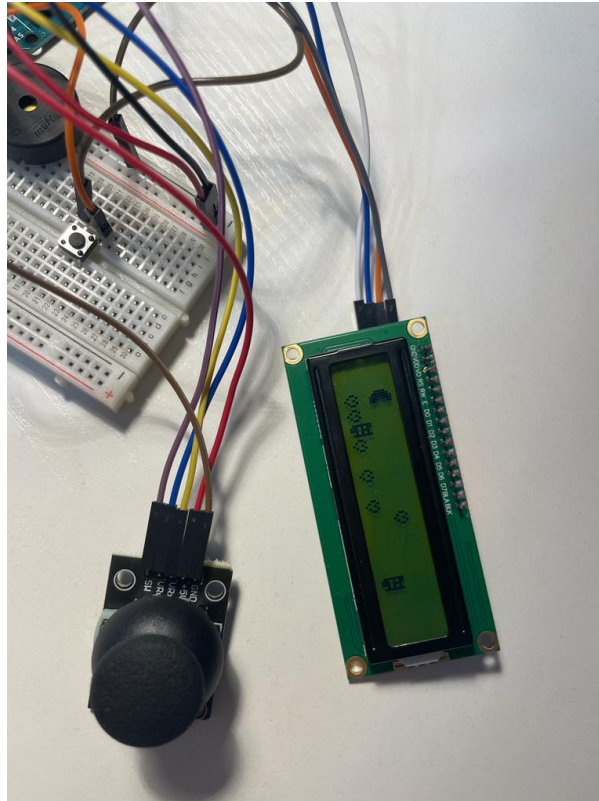
În funcția play(), valoarea yValue este utilizată pentru a determina poziția curentă a joystick-ului și pentru a controla mișcarea lui Pac-Man în sus și în jos în funcție de aceasta.

ADC este folosit pentru a citi valorile analogice de la joystick și a le transforma în valori digitale pe care Arduino le poate procesa. Această conversie analog-digitală se realizează prin setarea corespunzătoare a registrelor ADMUX și ADCSRA și prin apelarea funcției analogRead() în care se specifică pinul analog de citire. Valoarea digitală rezultată este stocată în variabila yValue și este utilizată ulterior în funcția play() pentru a controla mișcarea Pac-Man.



Concluzii





Pentru functionalitatea completa:

<https://www.youtube.com/watch?v=6VUz6QU2y5k&list=PLvBYedcS0rmK3VVGU3XVkyZdAiXhIE7lu>

Download

Am atasat o arhiva cu fisierele obtinute in urma realizarii proiectului: surse, scheme, poze si un video

[pm.zip](#)

Jurnal

- 1-MAI-2023]- Finalizare Documentatie
- [10-MAI-2023]- Finalizare Hardware
- [23-MAI-2023]- Finalizare Software
- [27-MAI-2023]- Adaugare video, arhiva, explicatii pentru software

Resurse Software

<https://www.arduinolibraries.info/libraries/liquid-crystal-i2-c>

<https://docs.arduino.cc/learn/built-in-libraries/eeprom> <https://github.com/robsoncouto/arduino-songs>

<https://reference.arduino.cc/reference/en/language/functions/analog-io/analogread/>

<https://www.arduino.cc/reference/en/language/functions/digital-io/pinmode/>

Resurse Hardware

<https://arduinogetstarted.com/tutorials/arduino-joystick> <https://docs.arduino.cc/hardware/uno-rev3>

<https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/alexau/pac-man>



Last update: **2023/05/27 17:18**