

Flappy Bird

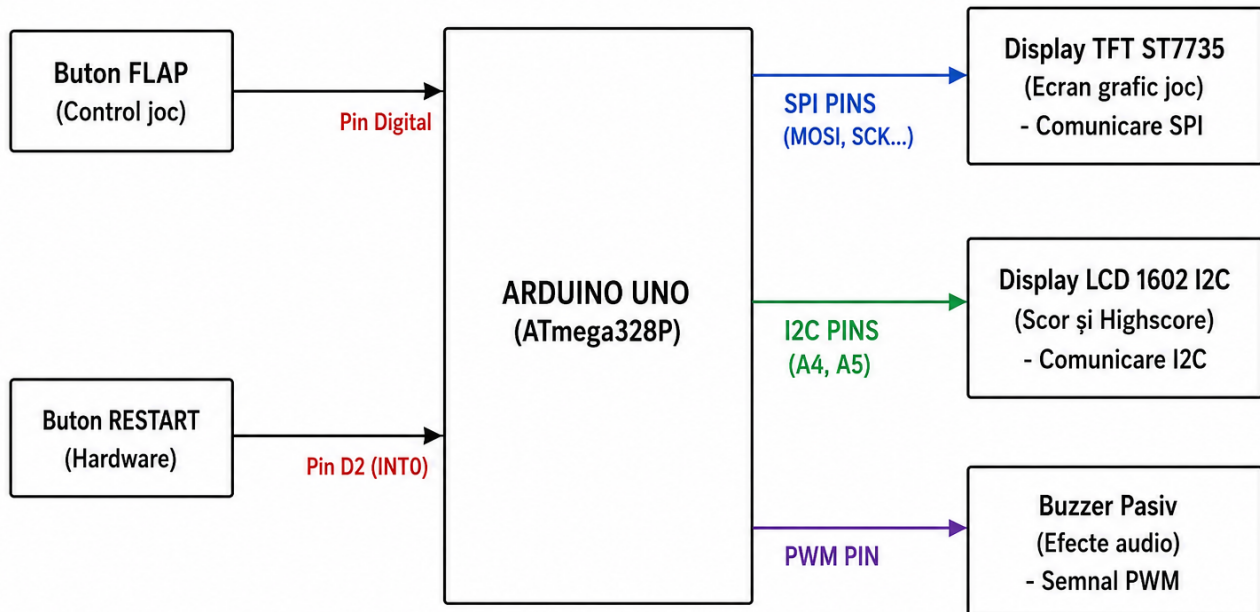
Introducere

Scopul acestui proiect este implementarea unei replici a celebrului joc "Flappy Bird", folosind un microcontroler compatibil ATmega328P. Jucătorul controlează o pasăre care trebuie să navigheze printre obstacole (țevi) generate aleator, apăsând un singur buton pentru a zbura mai sus. Ideea de la care a pornit proiectul este dorința de a îmbina logica software a unui joc video (mecanici de coliziune, generare de cadre, fizică simulată) cu utilizarea unor protocoale variate de comunicație hardware (SPI și I2C). Proiectul demonstrează cum un microcontroler simplu poate randa grafică pe un ecran TFT și gestiona simultan date secundare pe un ecran LCD, având feedback audio și reacții instantanee prin întreruperi.

Descriere generală

Sistemul este centrat pe placa Arduino Uno care procesează logica jocului și comunică cu patru module periferice:

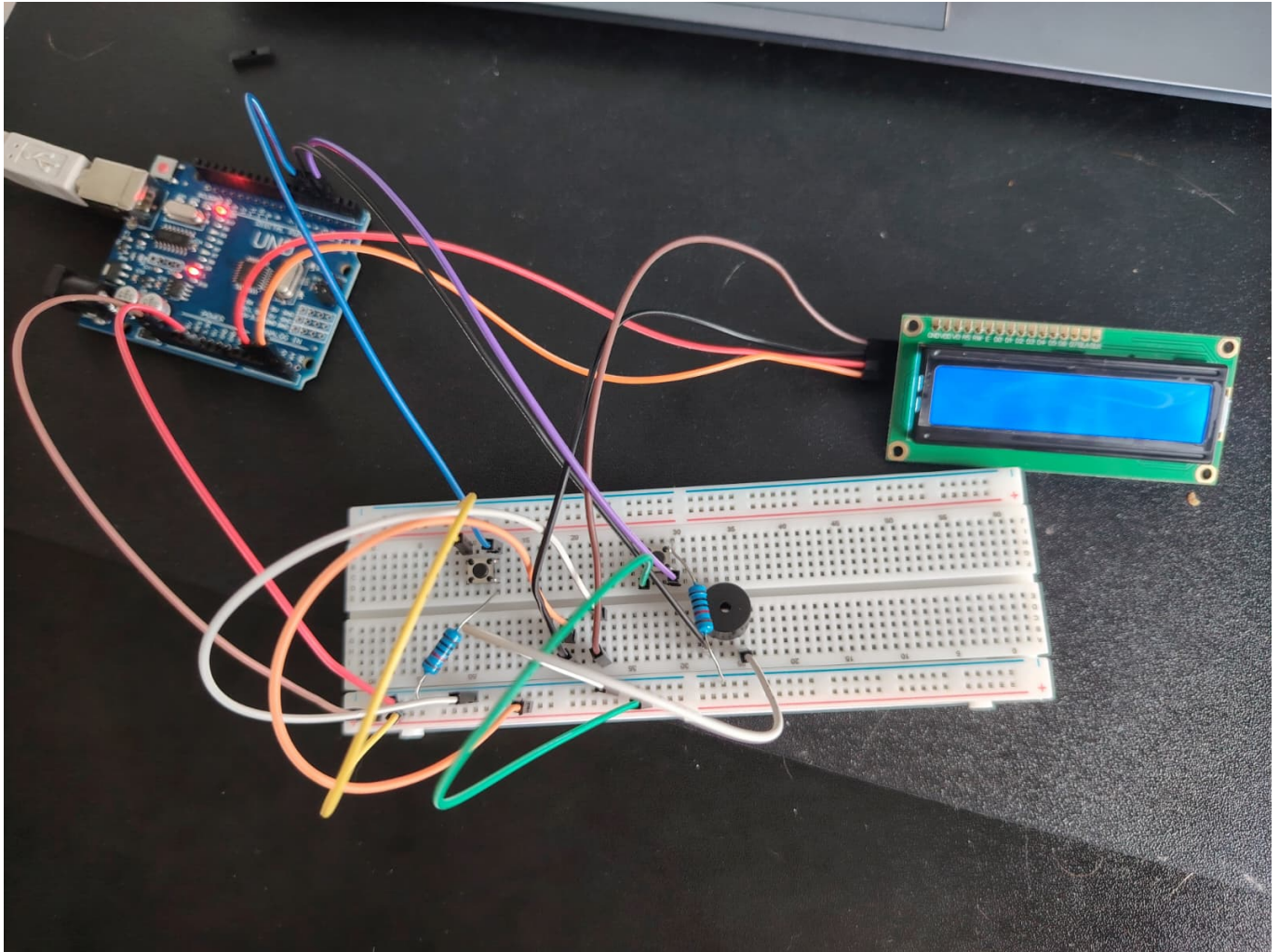
- Modulul de Afișare Grafic (SPI): Un display TFT ST7735 pe care se randează interfața principală a jocului (pasărea, fundalul, obstacolele).
- Modulul de Afișare Text (I2C): Un display LCD 1602 utilizat exclusiv pentru afișarea în timp real a scorului curent și a celui maxim (Highscore).
- Modulul de Intrare: Două butoane (Flap și Restart) citite prin rutine și întreruperi externe pentru un răspuns rapid la comenzile utilizatorului.
- Modulul Audio: Un buzzer pasiv acționat prin semnale PWM pentru a genera sunete la săritură și la terminarea jocului.



Hardware Design

Lista de piese (Bill of Materials):

1. Placă de dezvoltare Arduino Uno: Microcontrolerul principal care rulează logica jocului.
2. Display TFT ST7735 1.8": Ecran color comunicare SPI pentru randarea elementelor grafice.
3. Display LCD 1602 cu I2C: Ecran text pentru afișarea scorului, economisind pini datorită modului I2C.
4. Butoane (Push-button): 2 bucăți pentru controlul acțiunilor ("Sari" și "Restart").
5. Buzzer pasiv: Difuzor pentru feedback sonor (Flap, Game Over).
6. Rezistențe 10k ohmi: Folosite în configurație pull-down pentru butoane.
7. Breadboard și Fire Jumper: Pentru realizarea conexiunilor fizice.



Software Design

Descrierea codului aplicației (firmware):


- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

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/prj2026/ciprian.popescu0411/tudor_mihai.ostin 

Last update: **2026/05/15 08:05**