

Whack-A-Mole

Introducere

Proiectul este un joc interactiv de tip arcade, cu scop de entertainment, în care 4 servomotoare simulează apariția unor ținte ("cârțițe") la intervale aleatoare. Jucătorul trebuie să apese rapid butonul corespunzător fiecărei ținte pentru a înregistra o lovitură. Un câștig se obține la lovirea unui număr fixat de cârțițe în timpul alocat.

Ideea principală a fost de a implementa o logică de tip automat cu stări finite, cu diverse animații și sunete care să atragă potențiali jucători.

Proiectul integrează un afișaj cu interfață I2C pentru monitorizarea timpului și a scorului, un potențiomtru pentru ajustarea dificultății, un buzzer pasiv pentru sunete la acțiuni și câștig și un set de LED-uri pentru generarea unor animații la finalizarea jocului.

Poate fi util persoanelor de toate vârstele, inclusiv copiilor, care doresc să joace runde rapide cu feedback vizual plăcut.

Descriere generală

O rundă se începe prin apăsarea butonului de start.

Microcontrolerul contorizează timpul rămas prin intermediul unui timer intern (pentru durata totală a rundeii).

Un singur servomotor va fi activat la un moment dat, după un număr generat aleator.

Potențiometrul se citește de pe un pin ADC și descrie durata de activare a servomotorului, adică dificultatea aleasă.

Dacă se detectează o apăsare a butonului corespunzător în acest timp, scorul crește, și este afișat pe ecran, alături de timpul rămas și de dificultate.

Simultan, se emite un semnal PWM pe pinul buzzer-ului pentru confirmare sonoră. La câștig, se activează și buzzer-ul, și ledurile cu o animație.

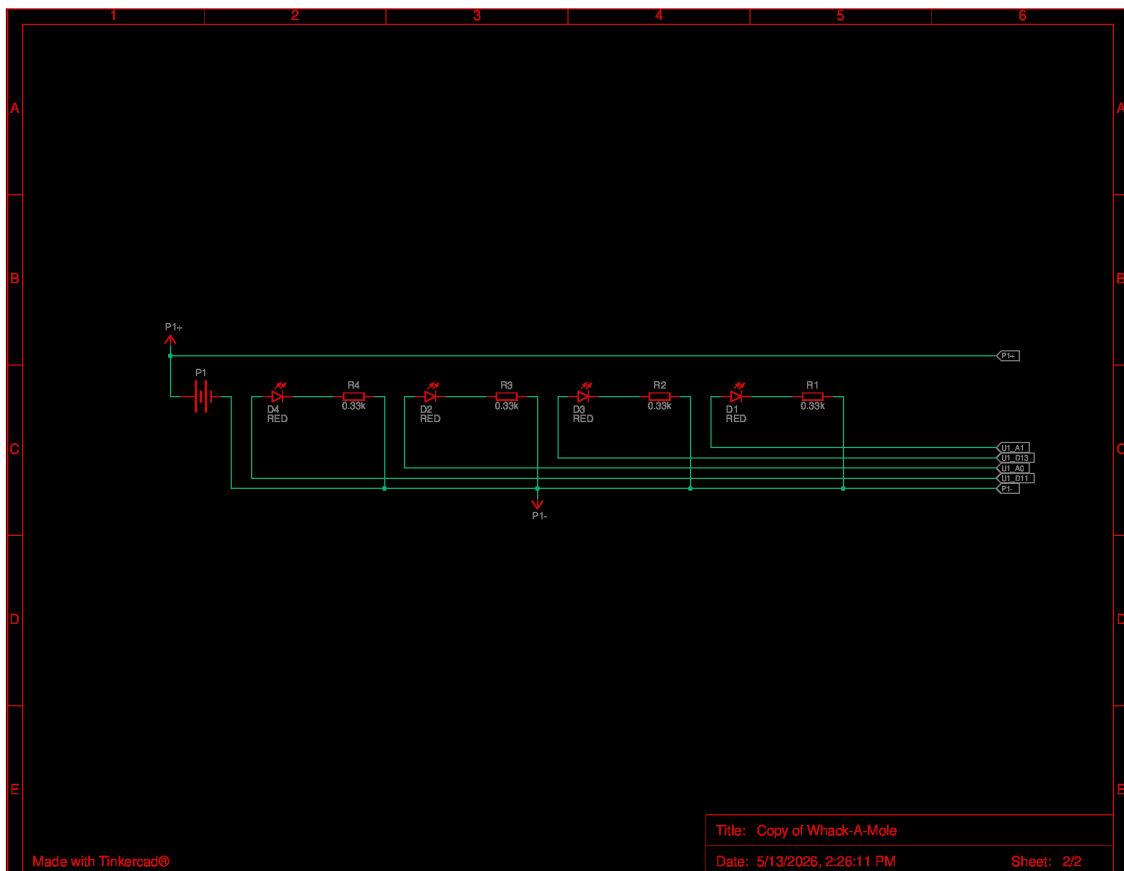


Hardware Design

Listă de componente:

- 1 kit rezistente (din care se folosesc 4 de 330 ohmi pentru led-uri)
- 1 buzzer pasiv pentru sunete la câștig
- 1 breadboard 830 puncte pentru montaj
- 1 potentiometru 10k pentru reglarea dificultății
- 5 butoane pcb 12x12x7.3mm, 4 de joc și 1 de start joc
- 4 capace roșii buton pcb 12x12x7.3mm
- 1 display oled 0.96 inch
- 4 servomotoare SG90
- 4 leduri roșii
- 1 placă compatibilă Arduino (cu uC ATmega328P)
- 1 sursă alimentare 9V 2A + 1 sursă alimentare breadboard MB102 pentru servomotoare
- 1 set cabluri jumper breadboard

Schema electrică



Servomotoarele sunt alimentate separat prin sursa de alimentare MB102 conectată la încărcătorul de

9V/2A pentru a nu suprasolicita magistrala de 5V de la uC. Ambele surse de alimentare au masă comună (punte între magistralele - de pe breadboard) pentru a trimite corect comenzile PWM.

Se propune funcționarea unui singur servo la un moment dat, prin urmare sursa MB102 care scoate maxim 5V/700mA este suficientă.

Pinout

Componentă	Pin
SERVO1	D3
SERVO2	D5
SERVO3	D6
SERVO4	D9
BUZZER	D10
Display_SDA	A4
Display_SCL	A5
POT1	A2
BTN_START	D2
BTN1	D4
BTN2	D7
BTN3	D8
BTN4	D12
LED1	D11
LED2	D13
LED3	A0
LED4	A1

Aranjamentele pe pini au luat în considerare și poziționarea fizică a pieselor (să nu fie prea îndepărtate sloturile de pe breadboard de uC etc.). Anumite neconcordanțe între distanțele fizice și pinii aleși pot fi explicate de mutarea poziționării pieselor pe breadboard pe parcursul proiectului (inițial LED-urile se aflau toate pe partea dreaptă), și a adăugării lor între timp.

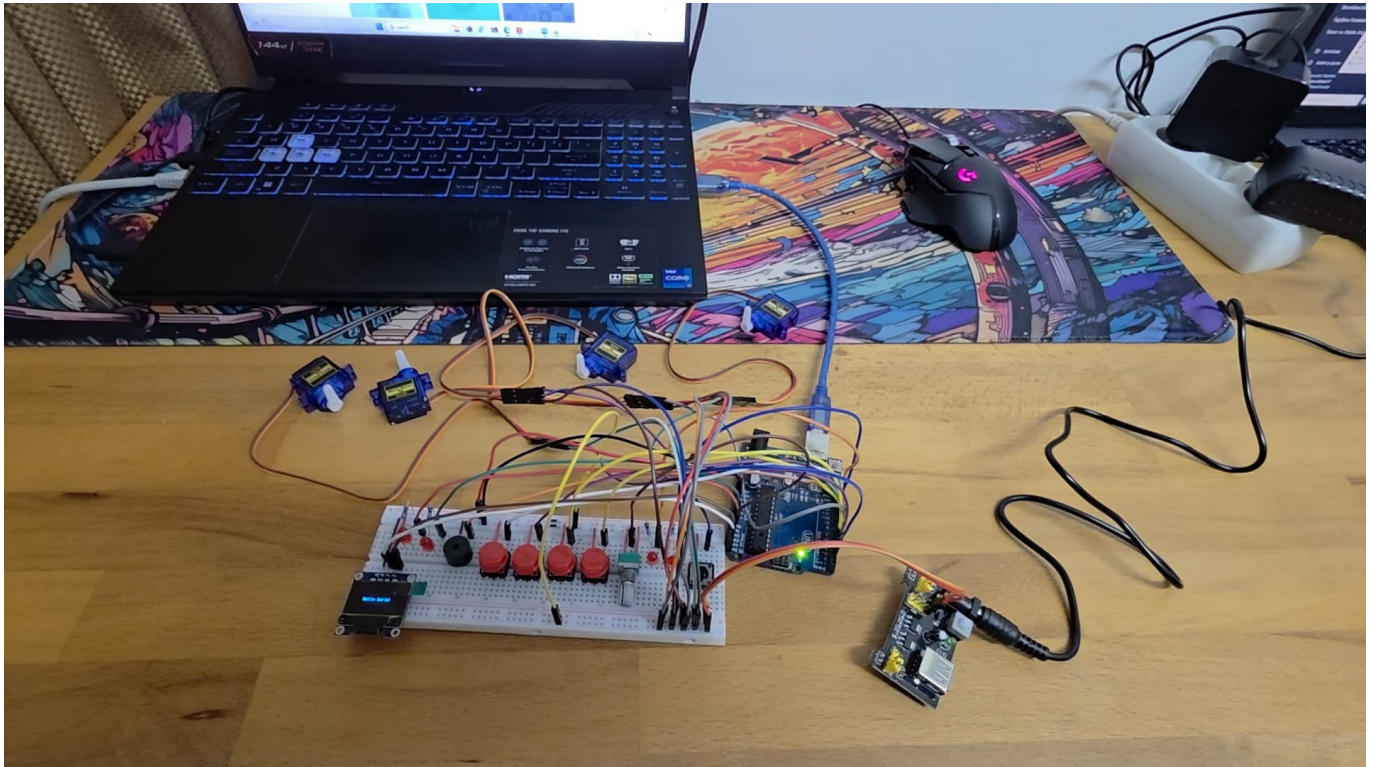
Două LED-uri și pinii I2C ai display-ului s-au pus și pe pini analogici din lipsă de pini disponibili, însă pot funcționa și cu output digital la fel de bine.

Potențiometrul necesită conversie ADC, deci pinul I-am ales dintre cei pentru input analogic.

Comenzile pentru servomotoare și buzzer se fac în regim PWM, am ales PWM hardware prin pinii digitali ce suportă.

Rezultate inițiale

Poza cu proiectul:



Demo-ul pentru etapa inițială hardware poate fi vizualizată în acest video: [Demo YouTube](#)

Alternativ, video-ul disponibil aici: [demo_razvan_miron.zip](#)

Se observă funcționarea display-ului prin mesaj Hello World împreună cu ansamblul de servomotoare care nu afectează intensitatea LED-urilor când acționează.

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

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/razvan.miron> 

Last update: **2026/05/13 15:30**