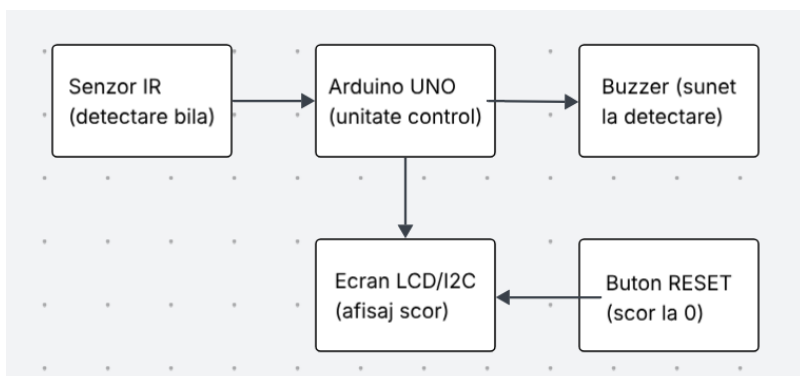


Penalty Shootout

Introducere

Proiectul constă în realizarea unui sistem interactiv care detectează când o bilă trece printr-o poartă de lemn și reacționează prin activarea unui buzzer și actualizarea unui scor afișat pe un ecran, dar și luminarea unor LED-uri cand cineva marcheaza/rateaza (verde/rosu). Scopul este de a simula un mecanism simplu de marcare, asemănător celor întâlnite în jocurile de tip mini-fotbal sau pinball. Ideea a pornit de la dorința de a combina elemente hardware și software într-un mod creativ și vizual. Este util atât din punct de vedere educațional (înțelegerea senzorilor, a afișajelor și a controlului cu Arduino), cât și recreativ, putând fi folosit în proiecte de tip DIY sau în jocuri interactive.

Descriere generală



Interacțiuni:

Cineva arunca bila de la o distanta anume spre poarta mica din lemn

Bila trece prin poartă → senzorul trimite semnal către Arduino

Arduino pornește buzzerul, actualizează scorul și luminează LED-urile verzi

Ecranul afișează GOAL! și scorul actualizat

Bila ratează poarta → senzorul trimite semnal către Arduino

Arduino pornește buzzerul cu un sunet trist, scorul nu se actualizează și luminează LED-urile roșii

Ecranul afișează MISS! și scorul rămas

Hardware Design



1) Module folosite:

Senzor ultrasonic (HC-SR04): Detectează trecerea bilei prin poartă

Arduino Uno R2 (A000066): Unitatea centrală de control

Buzzer (KY-006): Emite un sunet/melodie când bila este detectată

Ecran LCD/I2C (LCM1602 IIC V1): Afișează scorul (counter incrementat)

Fire clasice (Jumper wires) M-M si F-M: Asamblez cu ele proiectul

Led-uri (5mm): Lumineaza cand se da gol

2) Lista cu pinii folositi:

Microcontroller: Arduino UNO (A1)

Pin Arduino	Funcție	Legat la
-----	-----	-----
D2	Trigger senzor ultrasonic	TRIG pin (senzor ultrasonic)
D3	Echo senzor ultrasonic	ECHO pin (senzor ultrasonic)
D4	Control LED1	Anod D1 (LED1)
D5	Control LED2	Anod D2 (LED2)
D6	Control buzzer	BZ1 (buzzer activ)
A4 (SDA)	I2C data	SDA (pin 3 de la ecranul LCD I2C - U2)
A5 (SCL)	I2C clock	SCL (pin 4 de la ecranul LCD I2C - U2)
5V	Alimentare 5V pentru componente	VCC senzor ultrasonic, VDD LCD, buzzer, LED-uri
GND	Masă comună	GND senzor, LCD, buzzer, LED-uri

Senzor Ultrasonic (HC-SR04)

Pin Senzor	Conectat la
-----	-----
VCC	5V Arduino
GND	GND Arduino
TRIG	D2 Arduino
ECHO	D3 Arduino

LED-uri

LED	Anod	Catod
---	-----	---
D1	D4 Arduino	GND
D2	D5 Arduino	GND

Buzzer (BZ1)

Pin Buzzer	Conectat la
-----	-----
1	D6 Arduino
2	GND

LCD (U2: NHD-C0220BiZ - LCD cu interfață I2C)

Pin LCD	Funcție	Conectat la Arduino
-----	-----	-----
1 (VOUT)	Tensiune referință	-
2 (VSS)	GND	GND
3 (SDA)	I2C data	A4
4 (SCL)	I2C clock	A5
5 (VDD)	Alimentare	5V
6 (RST)	Reset	-
7, 8 (C1+, C1-)	Condensatori	-
9 (A)	Anod LED backlight	5V
10 (K)	Catod LED backlight	GND

Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare: Arduino
- librării și surse 3rd-party: Wire.h, LiquidCrystal_I2C.h
- algoritmi și structuri pe care plănuieți să le implementați:

Deteție bilă (senzor ultrasonic):

Măsoară distanța și detectează trecerea bilei prin poartă.

Praguri ajustabile (distance < 15cm = gol, distance > 30cm = ratare).

Gestionare scor:

Variabilă score + buton RESET pentru resetare.

Afișare pe LCD ("GOAL!", "MISS!", scor actual).

Feedback:

LED-uri: verde (gol), roșu (ratare).

Buzzer: melodii diferite pentru gol/ratare.

Structuri simple:

Array-uri pentru sunete (goalSound[], missSound[]).

Boolean (ballDetected) pentru evitarea trigger-urilor multiple.

* laboratoare utilizate:

GPIO - folosit in cam tot proiectul

PWM - leduri luminozitate

I2C - ecran LCD/I2C utilizat

* pseudocod:

1) Inițializare

Configurează pinii pentru:
senzor ultrasonic (TRIG, ECHO)
LED-uri (verde și roșu)
buzzer
buton de resetare

2) Inițializează ecranul LCD și afișează mesajul inițial:
„Penalty Shootout” și „Score: 0”

3) Loop:

3.1) Verifică dacă s-a apăsător butonul de reset:
Dacă DA → resetează scorul la 0
Afișează pe LCD: „Score Reset!” și „New Score: 0”
Apoi revine la ecranul principal

3.2) Măsoară distanța cu senzorul ultrasonic:
Trimite un semnal TRIG
Măsoară timpul de întoarcere a semnalului ECHO
Calculează distanța în centimetri

3.3) Verifică dacă mingea este detectată (GOAL):
Dacă distanța este între 5 cm și 15 cm și mingea nu a fost deja detectată:
Crește scorul
Afișează pe LCD: „GOAL!” și noul scor
Aprinde LED-ul verde
Redă un sunet de succes cu buzzerul

3.4) Verifică dacă mingea a trecut și e afară (MISS):
Dacă distanța e mai mare de 30 cm și mingea nu a fost detectată:
Afișează pe LCD: „MISS!” și scorul curent
Aprinde LED-ul roșu
Redă un sunet de eșec cu buzzerul

3.5) Așteaptă puțin și revine la pasul 1

- cod sursa: https://github.com/BanicaDragos/pm_project


Rezultate Obținute

În urma realizării proiectului “Penalty Shootout”, s-a reușit implementarea unui sistem interactiv de detectare a golurilor folosind un senzor ultrasonic. Dispozitivul a reacționat corect la trecerea mingii prin zona de detecție, afișând scorul pe un ecran LCD, oferind feedback vizual prin LED-uri și auditiv prin buzzer. Sistemul s-a dovedit funcțional și eficient în simularea unei experiențe de joc simplă, dar captivantă.

Concluzii

Proiectul a demonstrat cu succes integrarea senzorilor și a componentelor electronice într-o aplicație practică și distractivă. Prin combinarea hardware-ului cu programarea Arduino, s-a realizat un sistem capabil să ofere feedback în timp real, evidențiind potențialul tehnologiei în dezvoltarea jocurilor interactive.

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/prj2025/eradu/dragos.banica0711> 

Last update: **2025/05/25 13:03**

