Brick Breaker Game

Cristina-Andreea Szabo, 334CA

Introducere

Proiectul meu este un joc de tip Breakout, controlat cu Arduino Uno, care se joaca pe un display I2C OLED. Jucatorul controleaza paleta folosind un joystick, scopul fiind sa loveasca mingea si sa sparga toate blocurile fara sa piarda toate vietile.

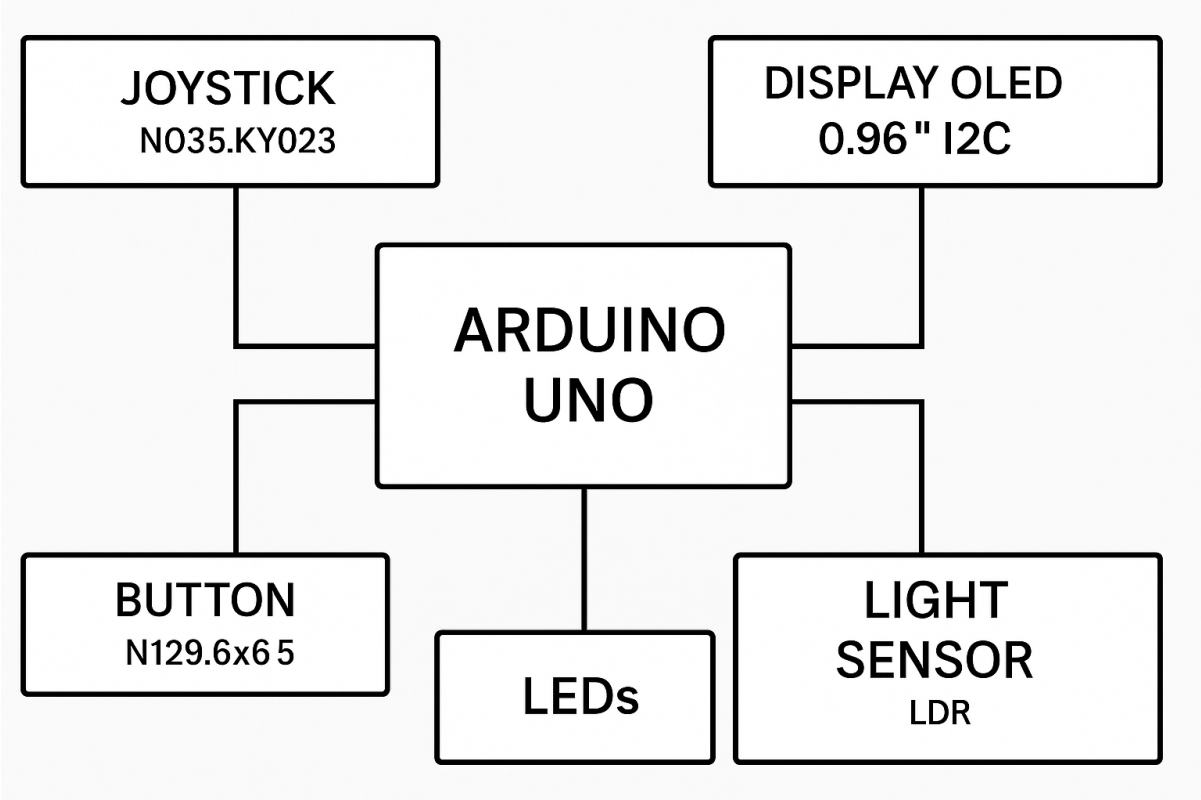
Am vrut sa recreez jocul copilariei mele si sa-l adaptez intr-o forma fizica, folosind componente simple. Am adaugat un buton pentru a pune jocul pe pauza, un modul semafor cu trei LED-uri care indica vietile ramase si un fotorezistor cu ajutorul caruia se schimba culoarea graficii de pe ecran in functie de cata lumina este in camera. Acest proiect imi ofera o baza pe care pot sa o extind pe viitor, adaugand noi niveluri si functionalitati.

Descriere generală

Pentru realizarea acestui proiect am folosit mai multe componente hardware care lucreaza impreuna pentru a face jocul sa functioneze corect. In centrul sistemului se afla placa Arduino Uno, care controleaza toate celelalte module. Pentru controlul paletei am folosit un joystick KY023, care trimite semnale analogice catre Arduino in functie de miscarea pe axa X. Aceste semnale sunt interpretate si paleta este mutata stanga-dreapta pe ecran.

Afisajul pe care se vede jocul este un display OLED de 0.96 inch, cu interfata I2C si rezolutie 128×64. El este conectat la Arduino prin pinii SDA si SCL si afiseaza grafic toate elementele jocului: mingea, paleta, caramizile si scorul. Am adaugat si un buton 6x6x5 care este conectat la un pin digital si are rolul de a pune jocul pe pauza atunci cand este apasat. Astfel, jucatorul poate intrerupe jocul si il poate relua oricand doreste.

Pentru afisarea vizuala a numarului de vieti ramase, am folosit un modul semafor cu trei LED-uri, conectate fiecare la cate un pin digital. La inceput toate trei sunt aprinse, iar pe masura ce se pierd vieti, ele se sting unul cate unul. Un alt element important este fotorezistorul, care este conectat la un pin analogic. Acesta citeste lumina ambientala si permite Arduino-ului sa adapteze culorile jocului.



Lista de componente:

- Arduino Uno - Placa de baza care controleaza intregul proiect

- Joystick KY023 pe doua axe XY - Folosit pentru a controla paleta din joc

- Display OLED 0.96” I2C - Afiseaza jocul

- Buton 6x6x5 - Permite punerea jocului pe pauza

- Modul LED semafor - Reprezinta vietile ramase ale jucatorului

- Fotorezistor - Detecteaza lumina ambientala

- Breadboard - Pentru montarea componentelor

- Fire de conexiune - Pentru legarea componentelor intre ele si la Arduino

- Rezistenta 10k - Folosita impreuna cu fotorezistorul

Hardware Design

BOM:

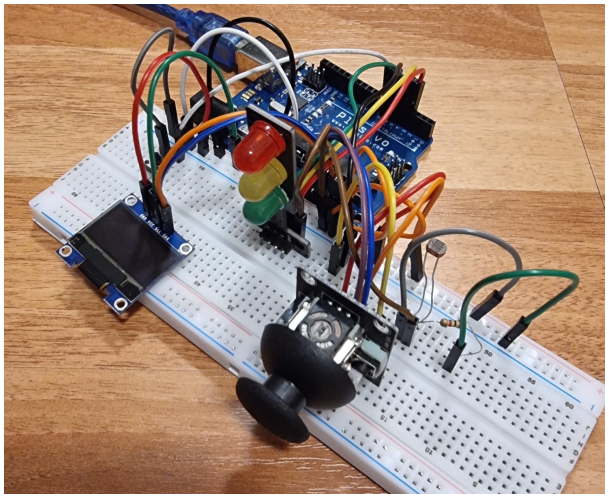
Componenta	**Link-uri**
Arduino Uno	Link
Breadboard 830 puncte MB-102	Link
Modul LED semafor, 56mm, 3.3-5V	Link

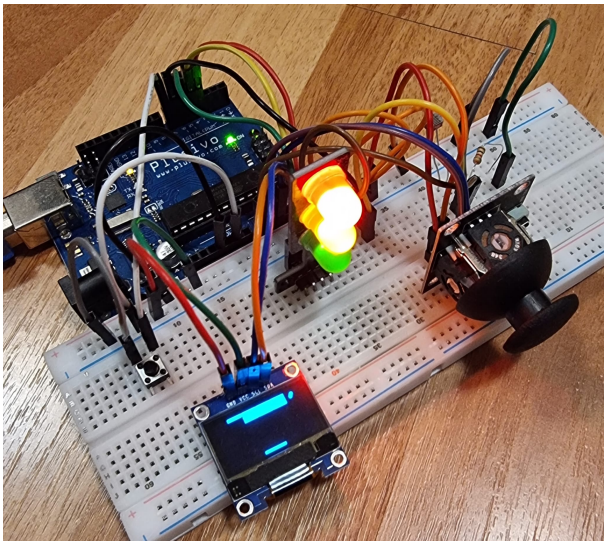
Buton Mini 6x6x5, 4 pini	Link
Fotorezistor (5537) 5mm	Link
Modul joystick doua axe XY	Link
Display OLED 0.96” I2C IIC Albastru	Link
2 x Fire de conexiune	Link

Descriere pini folositi:

Componenta	**Pini Arduino folositi**	**Functie**
Joystick KY-023	VRx → A1, VRy → A2 (nefolosit)	Axa X - control prin semnal analogic
	SW → D6	Buton apasare joystick
	VCC → 5V	Alimentare
	GND → GND	Masa
Display OLED I2C	SDA → A4	Comunicare I2C - date
	SCL → A5	Comunicare I2C - ceas
	VCC → 5V	Alimentare
	GND → GND	Masa
Buton 6x6x5 mm	Un pin → D7	Detecteaza intrerupere - pauza
	Celalalt pin → GND	Inchide circuitul
Modul semafor LED	Rosu → D3, Galben → D4, Verde → D5	Indica vietile ramase
	GND → GND	Masa comuna
Fotorezistor	Un capat → 5V	Alimentare
	Celalalt → A0	Citeste nivelul de lumina
	Rezistenta 10k → intre A0, GND	Formeaza divizor de tensiune

Implementare hardware:





Schema electrica:

Software Design

Surse si functii implementate

Functie	Descriere
setup()	Initializare pini, afisaj, intreruperi, lumina ambientala, stare joc
loop()	Executa logica jocului: miscare, coliziuni, afisare, verificare butoane, vieti si stare finala
reset()	Reseteaza complet jocul, afiseaza 3-2-1 cu delay hardware (prin startWaitTimer())
resetBall()	Pune mingea deasupra paletei si ii seteaza directia initiala
drawGame()	Deseneaza mingea, paleta si caramizile in functie de luminozitate
final()	Afiseaza pe ecran starea finala: “Winner” sau “Loser” si asteapta resetarea
updateLeds()	Aprinde LED-urile in functie de numarul de vieti
analogReadReg()	Citeste valorile analogice direct din registri ADC pentru viteza mai mare
startWaitTimer()	Configureaza Timer1 pentru o intarziere de ~1 secunda, neblocanta
ISR(TIMER1_COMPA_vect)	Semnalizeaza cand timpul setat in timer s-a scurs
ISR(PCINT2_vect)	Detecteaza apasarea butoanelor de pauza si reset, cu debounce minimal

Stadiul actual al implementarii software

Proiectul este complet functional. Jocul poate fi jucat cu minge si paleta, caramizile se distrug corect, afisajul OLED actualizeaza toate informatiile, iar luminile LED indica numarul de vieti ramase. Sunt gestionate corect pauza, resetarea jocului si detectarea starii finale (castig sau pierdere).

Alegerea bibliotecilor

Am folosit bibliotecile:

- `Wire.h` pentru comunicatia I2C cu ecranul OLED.

- `Adafruit_GFX.h` si `Adafruit_SSD1306.h` pentru a putea desena si scrie pe ecran.

Element de noutate

Proiectul include un mod automat de detectare a luminozitatii ambientale folosind un senzor LDR. In functie de lumina detectata, jocul isi inverseaza culorile pentru a fi vizibil si in intuneric, si in lumina puternica. Jocul foloseste un timer hardware (Timer1) pentru intarzieri precise fara a bloca logica programului. Jocul poate fi pus pe pauza si resetat prin intreruperi generate de butoane.

Justificarea utilizarii functionalitatilor din laborator

1. Intreruperi externe pentru butoanele de pauza si reset.

2. Timer hardware (Timer1) pentru realizarea unui delay neblocant inainte de startul jocului si la reseturi.

3. Acces direct la registrii ADC si I/O pentru eficienta sporita si control total.

4. Afisaj grafic OLED controlat prin I2C, ca aplicatie practica a interfetelor seriale studiate.

Structura si interactiunea dintre functionalitati

- La pornire, jocul verifica lumina ambientala si seteaza modul de afisare (light/dark).

- Se initializeaza paleta, mingea si caramizile.

- Joystick-ul controleaza paleta, iar mingea se misca automat, interactiunea fiind desenata in timp real pe ecran.

- Coliziunile cu caramizile sunt detectate si tratate, iar vietile sunt afisate cu LED-uri.

- Se foloseste un timer hardware pentru intarzieri fara a bloca restul programului.

- Jocul poate fi pus pe pauza sau resetat oricand prin intreruperi externe.

- Starea de castig sau pierdere este afisata pe ecran si permite reluarea jocului prin buton.

Calibrarea senzorilor

Senzorul LDR a fost calibrat folosind citiri brute din `analogReadReg()` la pornirea sistemului. Valoarea de prag (`threshold = 100`) a fost stabilita prin testare practica in mai multe conditii de lumina. Citirea se face cu acces direct la registrii ADC pentru eficienta si viteza.

Optimizari si mediu de dezvoltare

Folosirea ISR si a timerului hardware pentru delay-uri precise, fara blocaj si accesul direct la registri (`PORTD`, `ADMUX`, etc.) pentru viteza mai mare decat functiile standard Arduino. Mediul de dezvoltare este PlatformIO, codul fiind scris in C++, combinat cu cateva functii din bibliotecile folosite.

Rezultate Obţinute

Am atasat o arhiva cu codul meu si un filmulet de prezentare a proiectului.

brick_breaker_game.zip

Concluzii

Am realizat un joc de tip breakout pe un ecran OLED folosind Arduino. Am folosit un fotorezistor pentru a detecta luminozitatea ambientala, astfel incat afisajul sa se adapteze automat la conditiile de iluminare, fie in modul intunecat, fie in modul deschis. Controlul paletei se face prin joystick, iar butoanele imi permit sa pun jocul pe pauza sau sa il resetez. Logica jocului include detectarea coliziunilor mingii cu zidurile, paleta si caramizile, iar eu urmaresc reducerea vietilor si determinarea castigatorului sau invinsului. Am folosit intreruperi pentru butoane si timer pentru delay-uri, ceea ce imbunatateste functionalitatea si raspunsul sistemului. Prin acest proiect am vrut sa creez o experienta interactiva si distractiva.

Export to PDF