

Joc Snake

Introducere

În cadrul acestui proiect, îmi propun să implementez jocul Snake. Jucătorul va putea interacționa prin intermediul unui joystick. Voi folosi o matrice LED de dimensiune 8x8, iar scorul va fi afișat pe un ecran LCD. Proiectul are ca scop însușirea și aplicarea cunoștințelor și a conceptelor învățate la această materie.

Descriere generală

După cum am menționat mai sus, prin intermediul unui joystick, jucătorul va putea controla direcția de deplasare. De asemenea, prin apăsarea butonului joystick-ului, se poate modifica dificultatea jocului (viteza de deplasare). În tot acest timp, pe ecranul LCD se va afișa scorul curent. Atunci când jucătorul pierde, jocul se termină și pe ecran se va afișa un mesaj corespunzător.

Comunicarea dintre ecranul LCD și plăcuța Arduino se va realiza printr-un adaptor I2C, reducând astfel numărul de pini care vor fi ocupați. În cazul joystick-ului, cei doi pini pentru direcție vor fi conectați la intrări analogice, iar butonul va fi conectat la unul dintre pinii digitali ai plăcuței.

Schema bloc a proiectului este următoarea:



Hardware Design

Componentele utilizate sunt:

- Atmega 328P
- Matrice cu LED-uri RGB
- Modul LCD 1602
- Adaptor I2C pentru LCD
- Modul Joystick 2 axe și buton

Schema electrica



Modul de conectare este urmatorul: matricea RGB este conectata la pinul PD6, pentru a realiza comunicarea cu lcd-ul prin intermediul adaptorului I2C, folosesc pinii A4 si A5 (SDA si SCL), cele 2 axe ale joystick-ului sunt conectate la pinii analogici A0 si A1, iar butonul acestuia este conectat la pinul

digital PD2.

Software Design

Pentru dezvoltare, am folosit mediul de dezvoltare Arduino IDE 1.8.19. Am folosit librariile "Adafruit_NeoPixel" (pentru matricea RGB) si "LiquidCrystal_I2C" (pentru a controla LCD-ul prin adaptorul I2C).

Jocul functioneaza in felul urmatoar: initial, sarpele are dimensiunea 1, iar capul acestuia este reprezentat printr-un LED galben. Prada este reprezentata de un LED verde. Atunci cand sarpele "mananca" prada, dimensiunea acestuia creste cu o unitate. Restul LED-urilor din care e format sarpele sunt de culoare rosie. Directia in care acesta se deplaseaza este controlata prin joystick. Astfel, daca sarpele se deplaseaza pe orizontala, atunci o schimbare de directie inseamna ca se va deplasa pe verticala (in sus sau in jos, in functie de valoarea citita pe pinul analogic). Prin butonul joystick-ului, jucatorul poate controla viteza, existand trei viteze/graduri de dificultate. Pentru a actualiza matricea, folosesc intreruperi cu timer. Astfel, cand se atinge valoarea OCR1A, se declanseaza intreruperea ISR(TIMER1_COMPA_vect), in care se sting led-urile de la pasul anterior si se reaprind in pozitia noua. Valoarea pusa in registrul OCR1A influenteaza dificultatea jocului. Cu cat aceasta valoare este mai mica, cu atat intreruperea se va declansa mai frecvent, deci sarpele se va misca mai repede. Astfel, valoarea din OCR1A se va modifica atunci cand se apasa pe butonul joystick-ului in timpul rularii jocului. Intre doua intreruperi succesive generate de timer, se incearca citirea valorii de pe axa joystick-ului corespunzatoare (daca sarpele se deplaseaza pe orizontala, se va incerca citirea unei valori pe pinul specific axei verticale, si invers). Atunci cand valoarea citita depaseste un anumit prag, se va calcula configuratia urmatoare a matricii RGB, urmand ca apoi sa fie declansata intreruperea si sa se afiseze led-urile calculate anterior. De asemenea, atunci cand se apasa butonul joystick-ului se va declansa intreruperea ISR(INT0_vect).

Variabile folosite:

- uint8_t snake_body[NUMPIXELS] → vector ce contine LED-urile din care e format sarpele (fiecare LED are o valoare intre 0 si 63 ce semnifica pozitia sa)
- game_state → starea jocului: poate fi RUNNING sau GAME_OVER. Cat timp jocul este in starea GAME_OVER, se asteapta o apasare a butonului pentru a reincepe, iar led-urile sarpelui se aprind si se sting intermitent
- was_read → o singura data trebuie citita o valoare a joystick-ului care sa fie peste pragul de toleranta, intre doua intreruperi ale timer-ului
- unsigned int speed[MAX_LEVEL]; → valori ale registrului OCR1A (viteza jocului)
- analog_pin → pinul de pe care se face citirea (A0 sau A1)

Funcțiile folosite:

- bool check_input_detected() → verifica daca s-a facut o citire valida
- void start_game() → initializeaza setarile de start ale jocului: pozitia sarpelui, pozitia prăzii, scorul, etc.
- void compute_next_position() → va actualiza vectorul snake_body; de asemenea, verifica daca sarpele a mancat prada si daca acesta s-a ciocnit de el insusi; se apeleaza in loop, daca s-a detectat o citire valida sau in intreruperea generate de timer, in caz ca nu s-a modificat directia de deplasare
- void print_score() si void print_game_over() → afiseaza scorul pe lcd

- void display_running() → aprinde led-urile sarpelui si al prazii
- void display_game_over() → apelata in intrerupere cat timp jocul este in starea GAME_OVER; va stinge si va reaprinde led-urile sarpelui

Rezultate Obținute

Toate functionalitatile mentionate in sectiunea "Descriere Generala" au fost implementate si functioneaza corect.

Concluzii

Pentru mine, acest proiect a fost unul foarte interesant si mi-a facut placere sa lucrez la el, fiind prima data cand programez intr-un mediu Arduino. De asemenea, consider ca a fost un proiect util, deoarece am avut ocazia sa aplic cat mai multe din conceptele utilizate pe parcursul semestrului la laborator, si astfel, am inteles unele dintre concepte mult mai bine.

Download

[comanrobert_333cb_proiect_pm.rar](#)

Jurnal

21.04 → alegerea temei

4.05 → achizitionare componente

16.05 → inceperea implementarii

23.05 → finalizarea implementarii si a documentatiei

Bibliografie/Resurse

Laboratoarele de PM (OCW)

https://github.com/adafruit/Adafruit_NeoPixel

<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/imacovei/robert.coman1205>



Last update: **2022/05/23 14:34**