Tetra Pulse Arcade

Introducere

Prezentare pe scurt

Consola permite jucătorului să se bucure de experiența clasică Tetris, controlând mișcarea și rotirea pieselor. De asemenea acestuia ii este monitorizat pulsul, in functie de numarul de batai al ritmului cardiac pe care jucatorul il prezinta, dificultatea jocului creste sau scade.

Scopul proiectului

Scopul este de a recrea experiența autentică Tetris, oferind o modalitate distractivă și educațională de a învăța despre programarea pe un microcontroller.

Prezentarea pe scurt a proiectului vostru:

- ce face
- care este scopul lui
- care a fost ideea de la care aţi pornit
- de ce credeţi că este util pentru alţii şi pentru voi

Descriere generală

Implementarea unei console portabile pentru jocul Tetris, utilizeaza două matrice LED 8×8 conectate prin interfața SPI pentru afișarea pieselor și un display OLED 128×64 conectat prin I2C pentru afișarea scorului, pulsului și a altor informații suplimentare. Difuzorul redă muzica, un senzor de puls cardiac monitorizeaza ritmul cardiac al jucătorului iar cu ajutorul a șase butoane se manipuleaza pozitia pieselor de joc. Toate aceste componente sunt alimentate de o baterie.

Laboratoare utilizate:

- Laborator Intreruperi (un buton de reset al jocului)
- Laborator ADC (utilizam doar un pin analog pentru cele 6 butoane)
- Laborator SPI
- Laborator I2C

Schema Bloc

O schemă bloc cu toate modulele proiectului vostru, atât software cât şi hardware însoţită de o descriere a acestora precum şi a modului în care interacţionează.

Exemplu de schemă bloc: http://www.robs-projects.com/mp3proj/newplayer.html

Hardware Design

Componente utilizate

- 1 x Arduino UNO R3 ATmega328p
- 2 x Matrice cu led-uri 8×8
- 1 x Breadboard
- 1 x Difuzor
- 1 x Baterie 5V
- 6 x Buton switch tactil cu LED
- 1 x Senzor puls cardiac
- 1 x Ecran OLED 0.96 Inch SSD1306
- + Fire, Rezistente

Arduino UNO R3 ATmega328p

Rol: unitatea centrala a consolei ce contine microcontrolerul ATmega328p, aceasta preia inputul butoanelor apasate de utilizator si realizeaza comunicarea cu restul perifericelor.

Matrice cu led-uri 8×8

Rol: afiseaza jocul de Tetris

Pini:

- VCC - Pin 5V Arduino
- GND - Pin GND Arduino
- DIN - PD12
- CS - PD10
- CLK- PD11

Senzor puls cardiac

Rol: monitorizeaza pulsul cardiac al jucatorului

Pini:

- S - A0
- + - Pin 5V Arduino
- - - Pin GND Arduino

Ecran OLED 0.96 Inch SSD1306

Rol: afisarea scorului, pulsului jucatorului si a dificultatii

Pini:

- GND - Pin GND Arduino
- VDD - Pin 5V Arduino
- SCK - A5
- SDA - A4

Driver folosit: SSD1306, I2C

Difuzor

Rol: redarea muzicii ambientale

Pini:

- GND - Pin GND Arduino
- SIGNAL - PD13
- 3.3V - Pin 3.3V Arduino

Baterie

Rol: Generator de tensiune 5V pentru alimentarea Arduino de la USB

Breadboard + Butoane + Fire + Rezistente

Rol: Butoanele de tip push sunt conectate între ele prin rezistoare de 220Ohmi, respectiv 330Ohmi, astfel încât fiecare buton generează valori diferite pe pinul A1. Asfel, utilizand un singur pin si un ADC putem sa captam inputul utilizatorului de la cele 6 butoane.

Scheme Electrice

Calibrare Senzori

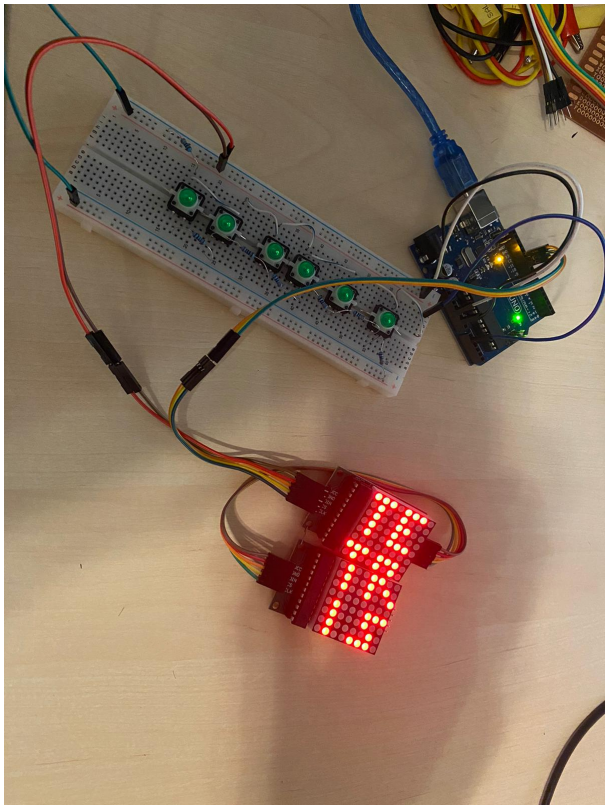
Senzor puls cardiac

Pentru a putea calibra intr-un mod cat mai corect senzorul am realizat urmatorii pasi: Am setat un prag al pulsului, am cosiderat ca acesta se situeaza intre 0 si 540, apoi am testat daca exista “noise”,

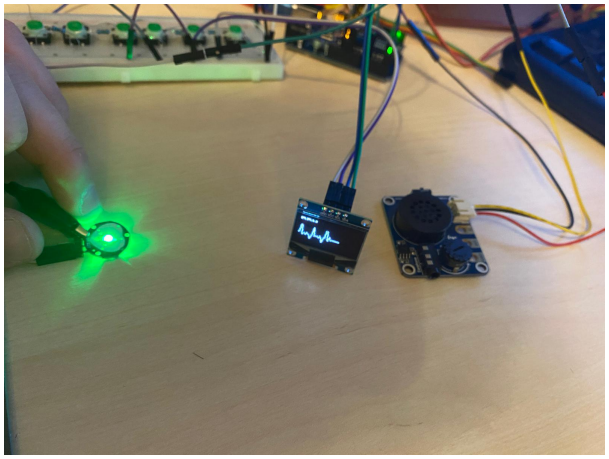
exista doar ca este foarte mic pentru a avea o influenta mare asupra rezultatului final. Am ales mai multe locuri de amplasare a senzorului si am inceput sa stabilesc niste medii de referinta, am comparat cu un alt pulsmetru(un pulsoximetru mai 😀) , valorile rezultate sunt cu +/- 10% mai mari sau mai mici, iar locul unde masori pulsul, sau presiunea pe care o aplici asupra senzorului au o influenta foarte mare asupra valorilor citite.

Etape de Dezvoltare

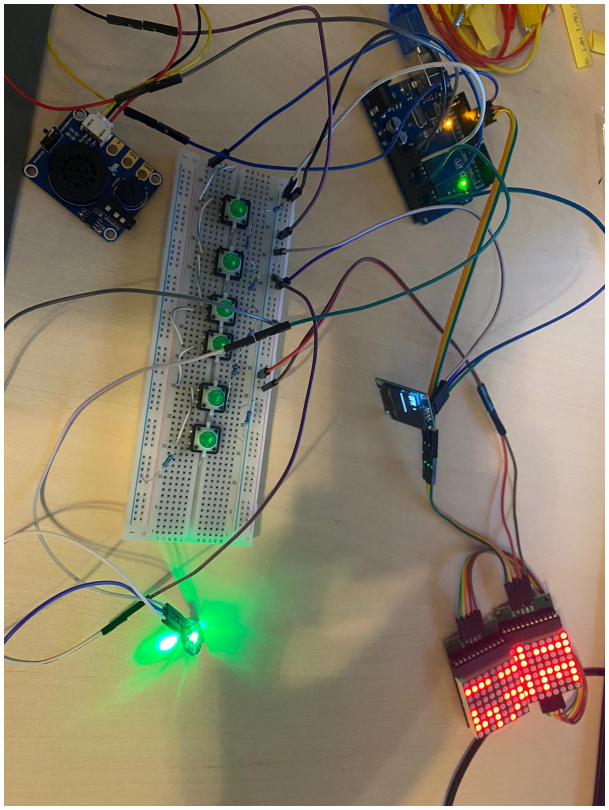
Testare Matrice Led 8*8 si butoane



Testare OLED Display, buzzer si pulsmetru



Prototip hardware



Aici puneţi tot ce ţine de hardware design:

- listă de piese
- scheme electrice (se pot lua şi de pe Internet şi din datasheet-uri, e.g. http://www.captain.at/electronic-atmega16-mmc-schematic.png)
- diagrame de semnal
- rezultatele simulării

Software Design

Stadiu actual implementare software

Codul este in proportie de 70% finalizat, jocul de tetris functioneaza bine, mai trebuie sa integrez melodia pentru buzzer si trebuie sa modific ca valorile butoanelor sa fie citite de pe un pin analog. Ma mai gandesc sa implementez si un buton de reset, care sa reseteze starea jocului (vad cat timp mai am si daca imi iese pana luni).

Mediu de dezvoltare

Am utilizat Arduino IDE

Librarii utilizate

- LedControl.h - Permite configurarea și controlul matricelor LED, oferă funcții pentru a seta intensitatea luminii, a porni/opri afișajele și a actualiza conținutul acestora.
- SPI.h - Aceasta librărie permite comunicarea cu dispozitivele care folosesc protocolul Serial Peripheral Interface (SPI). Este necesară pentru a comunica cu cipul MAX7219, care controlează matricea LED.
- Wire.h - Aceasta librărie este utilizată pentru comunicarea I2C, un alt protocol comun pentru interfațarea cu senzori și alte periferice. Este necesară pentru a comunica cu afișajul OLED SSD1306.
- Adafruit_GFX.h - Aceasta este o librărie de bază pentru grafică, dezvoltată de Adafruit. Oferă funcții pentru desenarea de forme, text și bitmap-uri.
- Adafruit_SSD1306.h - Aceasta librărie este specifică pentru controlul afișajelor OLED bazate pe controlerul SSD130

Utilitatea Laboratoarelor

Laboratorul de I2C → pentru a putea afisa pe diplay-ul OLED este utilizata biblioteca Adafruit_SSD1306 care foloseste protocolul I2C

Laboratorul de SPI → pentru a comunica cu Matricea de Led uri este utilizat protocolul SPI

Laboratorul ADC → pentru a nu mai utiliza cate un pin digital separat pentru fiecare buton, utilizam un singur pin pentru a citi valoarea a 6 butoane

Descrierea codului aplicaţiei (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării şi surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi şi structuri pe care plănuiţi să le implementaţi
- (etapa 3) surse şi funcţii implementate

Rezultate Obţinute

DEMO

Yt

Care au fost rezultatele obţinute în urma realizării proiectului vostru.

Concluzii

Download

Arhiva tema

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

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele 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** şi **Resurse Hardware**.

Export to PDF