

George-Alexandru VLAD (78422) - Snake

Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

Imi propun implementarea unui joc de Snake care sa aiba urmatoarele functionalitati:

- buzzer care sa redea sunete in-game
- ecran lcd in care se va face afisarea
- nivele diferite de dificultate/accelerarea deplasarii sarpelui
- utilizare joystick pentru deplasarea sarpelui

Scopul jocului este sa deplasezi sarpele catre iteme generate pe harta care ii maresc dimensiunea evitand coliziunea cu el insusi sau cu eventuale alte obstacole depinzand de varianta de joc.

Ideea de la care am pornit este jocul Snake. [Istoric joc Snake](#)

Jocul Snake poate reprezenta o activitate recreativa.

Descriere generală

Schema bloc:



Hardware Design

Componente	Cantitate
ATMega324	1
Placa de baza	1
Placa de test	1
Modul LCD ST7735R	1
Joystick	1
Buzzer	1

Schema electrica:



Software Design

Mediu de dezvoltare Proiectul a fost dezvoltat în [Atmel Studio 7.0](#), bazat pe Visual Studio. Oferă opțiunea de “Build project”, care generează un fișier .hex ce poate fi încărcat direct pe placuta atât din linia de comandă cât și din interfața grafică utilizând [HIDBootFlash](#).

Structura proiect

LCD grafic

LCD-ul folosit are rezoluția 128×160 și este controlat prin controller-ul [ST7735R](#). Interfațarea cu microcontroller-ul se face prin SPI (MOSI/MISO/SCK). Comunicația între microcontroller și controller-ul LCD-ului grafic se face printr-un set de comenzi specificate în datasheet-ul controller-ului LCD. Scheletul laboratorului conține API-ul pentru lucrul cu acest controller în fișierul ST7735R.h. Mai jos sunt câteva dintre funcțiile oferite:

[ST7735R_TFT.h](#)

```
/* Initializeaza Display-ul grafic. */
void ST7735R_Begin();

/* Deseneaza o linie de la (x0, y0) la (x1, y1), avand culoarea data de
parametrii r, g, b. */
void ST7735R_Line(int x0, int y0, int x1, int y1, uint8_t r, uint8_t g,
uint8_t b);

/* Afiseaza un text, incepand de la pozitia (x, y).
* Textul are culoarea specificata de pereche (r, g, b).
* Background-ul pe care este afisat textul are culoarea (bgR, bgG,
bgB).
*/
void ST7735R_DrawText(int x, int y, const char *text, uint8_t r,
uint8_t g, uint8_t b,
uint8_t bgR, uint8_t bgG, uint8_t bgB);

/* Desenează un cerc cu centrul la coordonatele (x, y), de raza
'radius', cu culoarea specificată. */
void ST7735R_Circle(int x, int y, uint8_t radius, uint8_t red, uint8_t
green, uint8_t blue);

/* Desenează un cerc cu centrul la coordonatele (x, y) de raza 'radius'
și îl umple cu culoarea dată. */
void ST7735R_FilledCircle(int x, int y, uint8_t radius, uint8_t red,
uint8_t green, uint8_t blue);

/* Este un macro: deseneaza un dreptunghi cu colțul stânga sus dat de
(x0, y0)
```

```

* și colțul dreapta jos dat de (x1, y1).
* Dreptunghiul este umplut cu culoarea (r, g, b).
*/
ST7735R_FillRect(x0, y0, x1, y1, r, g, b);

```

spi.h

```

#ifndef SPI_H
#define SPI_H

#include <stdint.h>

/* SPI config */
#define SPI_PORT      PORTB
#define SPI_DDR       DDRB
#define SPI_MISO      PB6
#define SPI_MOSI      PB5
#define SPI_SCK       PB7

void SPI_init(void);
uint8_t SPI_exchange(uint8_t data);

#endif // SPI_H

```

Bibliotecile se pot găsi în cadrul laboratorului 5 [LCD ST7735R_TFT/SPI - Laborator 5](#).

Game API

Contine o serie de funcții publice sau private prin intermediul cărora se controlează fluxul jocului. De asemenea, contine structuri definite pentru a realiza o separare și grupare logică a componentelor/proprietăților anumitor elemente de joc.

În structura `game_t` se vor reține informații relevante despre starea jocului precum sarpele, statusul jocului (castigat, pierdut sau încă în desfășurare) și un vector cu itemele consumabile generate pe hartă. Sarpele este conceput ca un vector de segmente de tipul `segment_t`, fiecare segment păstrând informații privitoare la culoare, poziția pe hartă, dacă se identifică sau nu cu capul șarpelui, etc. Itemele consumabile sunt caracterizate de structura `food_t`, aceasta oferind informații despre poziția, culoarea itemului, cât și statusul acestuia, dacă este sau nu consumat. Pentru o mai ușoară manipulare a componentelor rgb ale culorilor le-am grupat într-o structură cu 3 câmpuri asociate acestora.

```

/*****
*
* Public Functions
*****/

void GAME_init(void);
void GAME_move_snake(int x, int y);
void GAME_loop(void);
int GAME_boost(void);

```

```
int GAME_buzz_delay(void);

/*****
 *
 * Private Functions Prototypes
 *****/

/* Functii implementate la momentul curent al dezvoltarii jocului */

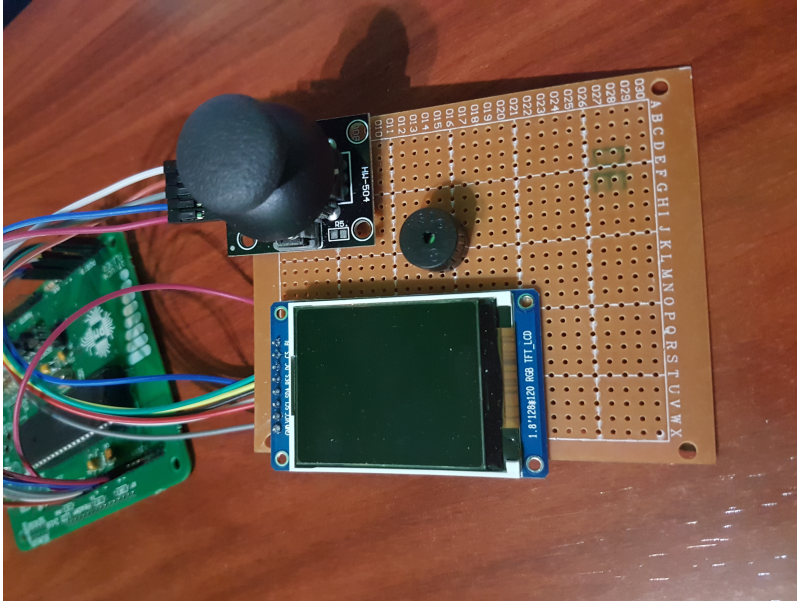
static void GAME_draw_snake(void);
static void GAME_draw_segment(int x, int y, uint8_t r, uint8_t g, uint8_t b);
static void GAME_generate_food(int index);
static void GAME_draw_food(int index);
static void GAME_draw_all_food(void);
static void GAME_init_food(void);
static void GAME_physics(void);
static void GAME_draw_score(void);
static void GAME_init_segment(uint8_t head, int pos_x, int pos_y, int index);
static int GAME_check_food_collision(void);
static int GAME_check_snake_collsion(void);
static void GAME_buzzer_morse(int tip_morse);
static void GAME_make_sound(int tip_morse, int type);
```

Joystick Nu a necesitat API specializat, controlul se face in cadrul fisierului sursa ||Game API||. Functia responsabila de miscarea directionata a sarpelui este ||void GAME_move_snake(int x, int y)||, cu parametrii x si y ce reprezinta valori ale inclinarii pe axele Ox/Oy, valori primite de la modulul joystick.

In functia **main()** a jocului se realizeaza initializarile necesare, apelul functiei de initializare a structurilor jocului, si rularea efectiva a jocului intr-o bucla *for*. Dupa termianarea jocului se vor astepta 4 secunde pana la reinceperea unei noi sesiuni de joc.

Rezultate Obținute

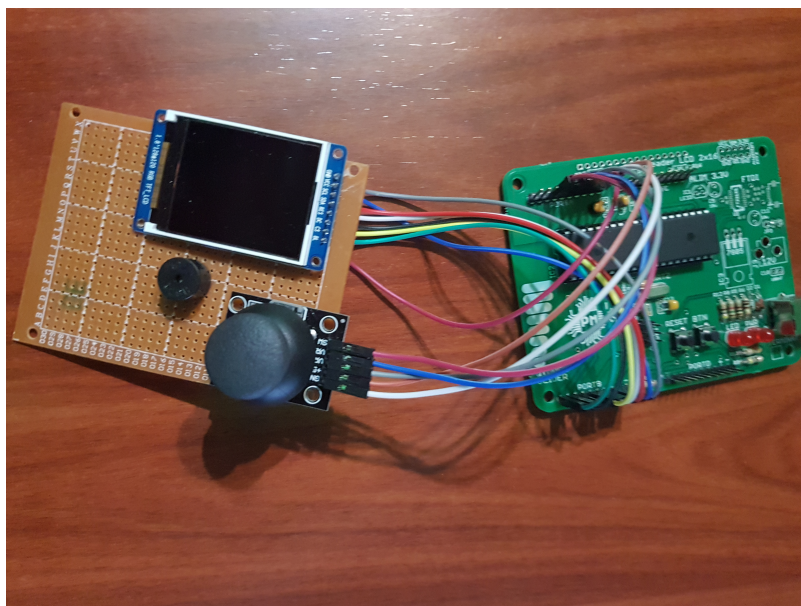
1) Conectare ecran LCD, Joystick, buzzer pe placuta de test



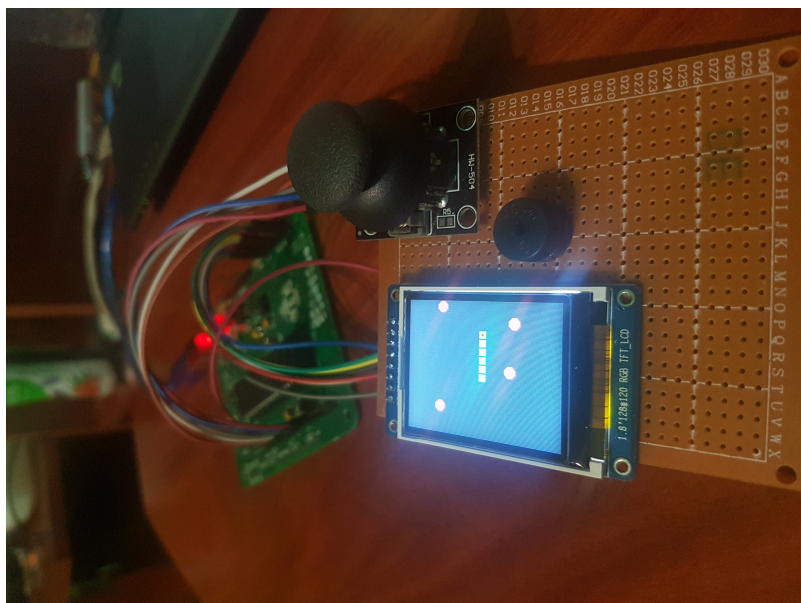
2) Placa de baza



3) Proiect hardware complet



4) Implementare software



Concluzii

Realizarea unui astfel de proiect, de o complexitate mai ridicata decat majoritatea proiectelor de pana acum, aduce o satisfactie personala foarte mare. Incepem practic de la 0 si construim pas cu pas produsul finit, atat hardware cat si software, vedem cum munca noastra se transpune intr-un element fizic. O experienta placuta ce a necesitat cercetare extensiva, rabdare si preocupare.

Download

Arhiva: [335cb_vlad_georgealexandru_snake.zip](#)

Jurnal

Saptamana 1: Stabilirea temei proiectului, schema bloc si formularea unei liste de piese

Saptamana 2: Finalizarea lipirii componentelor pe placa + incarcare bootloader

Saptamana 3: Achizitionarea componentelor specifice proiectului

Saptamana 4: Asamblarea componentelor finale

Saptamana 5: Realizarea software-ului

Bibliografie/Resurse

[Cablaj imprimat 2018](#)

[Datasheet ATmega324](#)

[Datasheet ST7735](#)

- Documentația în format [PDF](#)

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2018/astatulat/alexandru_vlad_snake_game



Last update: **2021/04/14 15:07**