

# Alexandru DINU (66860) - DX-Ball

Autorul poate fi contactat la adresa: [alex.dinu07@gmail.com](mailto:alex.dinu07@gmail.com)

## Introducere

<imgcaption exemplu\_gameplay|Original DX-Ball gameplay>



</imgcaption>

Proiectul constă în implementarea clasicului joc de tip arcade [DX-Ball](#) (1996), în care user-ul controlează un paddle și o minge, cu scopul de a elimina blocurile colorate și de a avansa în nivele. Vor exista multiple tipuri de blocuri, acestea respectând un anumit layout în funcție de nivel și fiecare incrementând scorul în mod diferit:

- blocuri care vor da diverse power-up-uri (atât pozitive, cât și negative);
- blocuri care vor exploda, eliminând blocurile adiacente;
- blocuri care necesită mai mult decât o lovitură pentru a fi distruse.

Pentru a păstra cât mai mult din gameplay-ul original și pentru a obține o experiență vizuală mai bogată, voi folosi un modul LCD color de 2.2" (controller HX8340-B). De asemenea, controlul plăcuței (paddle-ului) se va face cu ajutorul accelerometrului.

Am ales DX-Ball pentru că a fost printre primele jocuri pe care le-am jucat și am considerat că implementarea acestuia în cadrul proiectului mă va ajuta să aprofundez cunoștințele din acest domeniu într-un mod plăcut.

Proiectul este, în principal, util pentru a învăța cum să realizez interfațarea cu un modul LCD și un accelerometru atât hardware, cât și prin folosirea unor API-uri specifice.

## Descriere generală

<imgcaption schema\_bloc\_proiect|Schema bloc a proiectului>



</imgcaption>

Controlul efectiv al paddle-ului va fi realizat prin interpretarea datelor de la accelerometru: rotația acestuia în jurul unei axe va fi mapată (software) pe mișcarea paddle-ului. Începerea jocului (prin lansarea mingii de pe paddle) va fi marcată prin apăsarea unui pushbutton de pe placa de bază. Desenarea pe LCD[1] va fi implementată adaptând biblioteca acestuia pentru utilizarea cu ATmega324. În biblioteca sunt definite funcții de generare de primitive (linii, cercuri, dreptunghiuri), precum și funcții care permit scrierea de text.

În joc vor exista blocuri de diferite tipuri, a căror distrugere va incrementa scorul jucătorului cu o anumită valoare și, eventual, va genera un power-up (pozitiv sau negativ) pe care jucătorul trebuie să-l preia sau să-l ignore. Scopul este de a avansa în nivele și de a nu lăsa mingea să cada în afara limitei inferioare a ecranului (caz în care numărul de vieți scade). Jocul se termină fie când numărul de vieți ajunge la 0, fie când se termină ultimul nivel. Power-up-urile pozitive sunt: extra life, shooting paddle, multiply exploding, iar cele negative: kill paddle, mini ball, fast ball.

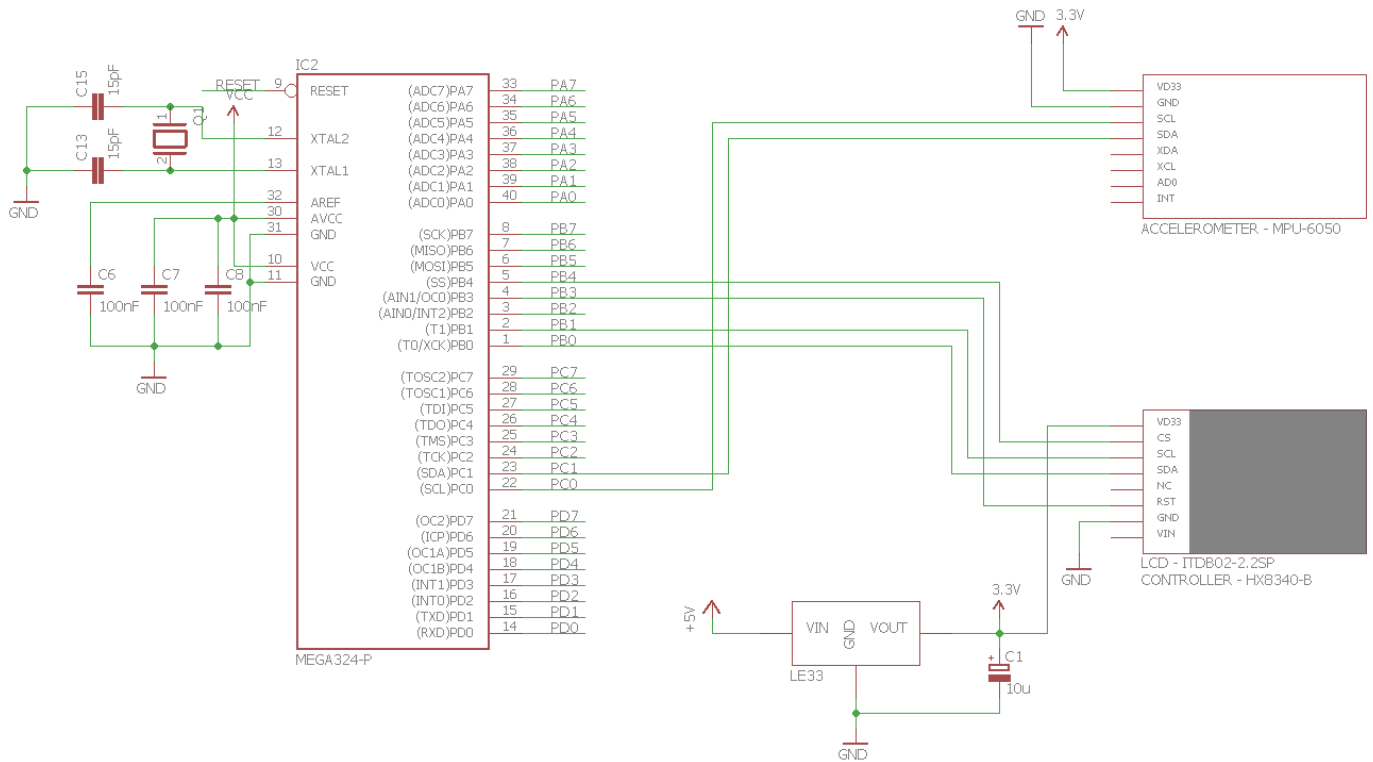
[1]Alimentarea modului LCD se face la 3.3V utilizând un regulator de tensiune LE33 (plăcuța în sine va funcționa la 3.3V).

## Hardware Design

### Lista de piese

Nume	Detalii	Cantitate
Placă de bază	PM 2017	1
Modul LCD	<a href="#">ITDB02-2.2SP (controller HX8340-B)</a>	1
Regulator tensiune 5V-3.3V	LE33	1
Condensator electrolitic	Capacitate 10 $\mu$ F	1
Accelerometru	<a href="#">MPU-6050</a>	1
Pushbutton	TS-06E (TS-3)	1

### Schema electrică



## Software Design

## Mediu de dezvoltare

WinAVR + Programmer's Notepad

## Biblioteci / surse 3rd-party

- [UTFT lib](#), adaptată pentru controller HX8340-B, portată în C (eliminare cod inutil, .hex mai mic)
- [MPU6050 lib](#)

## Algoritmi, tipuri și structuri de date

- Obiecte specifice jocului (paddle, ball, blocks) modelate ca structuri
- Starea curentă a jocului (nivel, scor, power-up - dacă există, număr de vieți rămase etc)
- Block bitmaps (folosite cu scopul de a defini un nivel prin aranjarea blocurilor pe ecran)
- AABB 2D collision detection (ball - blocks, ball - paddle)

## Surse implementate

- API lightweight pentru interfațarea cu modulul LCD

```
// lcd.h / lcd.c
void init_lcd(uint8_t orientation);
void fill_screen(uint8_t r, uint8_t g, uint8_t b);
void clear_screen();
void set_pixel(uint8_t r, uint8_t g, uint8_t b);
void draw_pixel(int x, int y);
void set_foreground_color(uint8_t r, uint8_t g, uint8_t b);
void set_background_color(uint8_t r, uint8_t g, uint8_t b);
```

- API pentru printare text și desenare primitive

```
// lcd.h / lcd.c
void set_font(uint8_t *font);
void print_char(uint8_t c, int x, int y);
void print_string(char *st, int x, int y, int deg);

void draw_straight_line(int x, int y, int l, int type);
void draw_empty_rectangle(int x1, int y1, int x2, int y2);
void draw_filled_rectangle(int x1, int y1, int x2, int y2);
void draw_empty_circle(int x, int y, int radius);
void draw_filled_circle(int x, int y, int radius);
```

- API pentru interfațarea cu accelerometrul MPU-6050

```
// mpu6050.h / mpu6050.c
// initialise the accelerometer and gyroscope
void mpu6050_init();
//get raw data converted to g and deg/sec values
void mpu6050_getConvData(double* axg, double* ayg, double* azg, double* gxds,
double* gyds, double* gzds);
```

## Rezultate obținute

În implementarea inițială am vrut să interfațez cu QTouch slider-ul, dar după integrarea bibliotecii și câteva ore de debug am observat că datele primite de la acesta sunt lipsite de sens. Astfel, pentru a nu pierde prea mult timp, am decis să înlocuiesc QTouch slider-ul cu un accelerometru (MPU-6050), pentru îndeplinirea aceluiași scop: mișcarea paddle-ului pe ecran. Am reușit foarte repede să interfațez cu accelerometrul și să interpretez datele de la acesta pentru a mișca paddle-ul. Mai mult, o rotație cu un unghi mai mare rezultă în accelerarea mișcării paddle-ului pe ecran.

Din acest punct, am început să schițez logica jocului în sine.

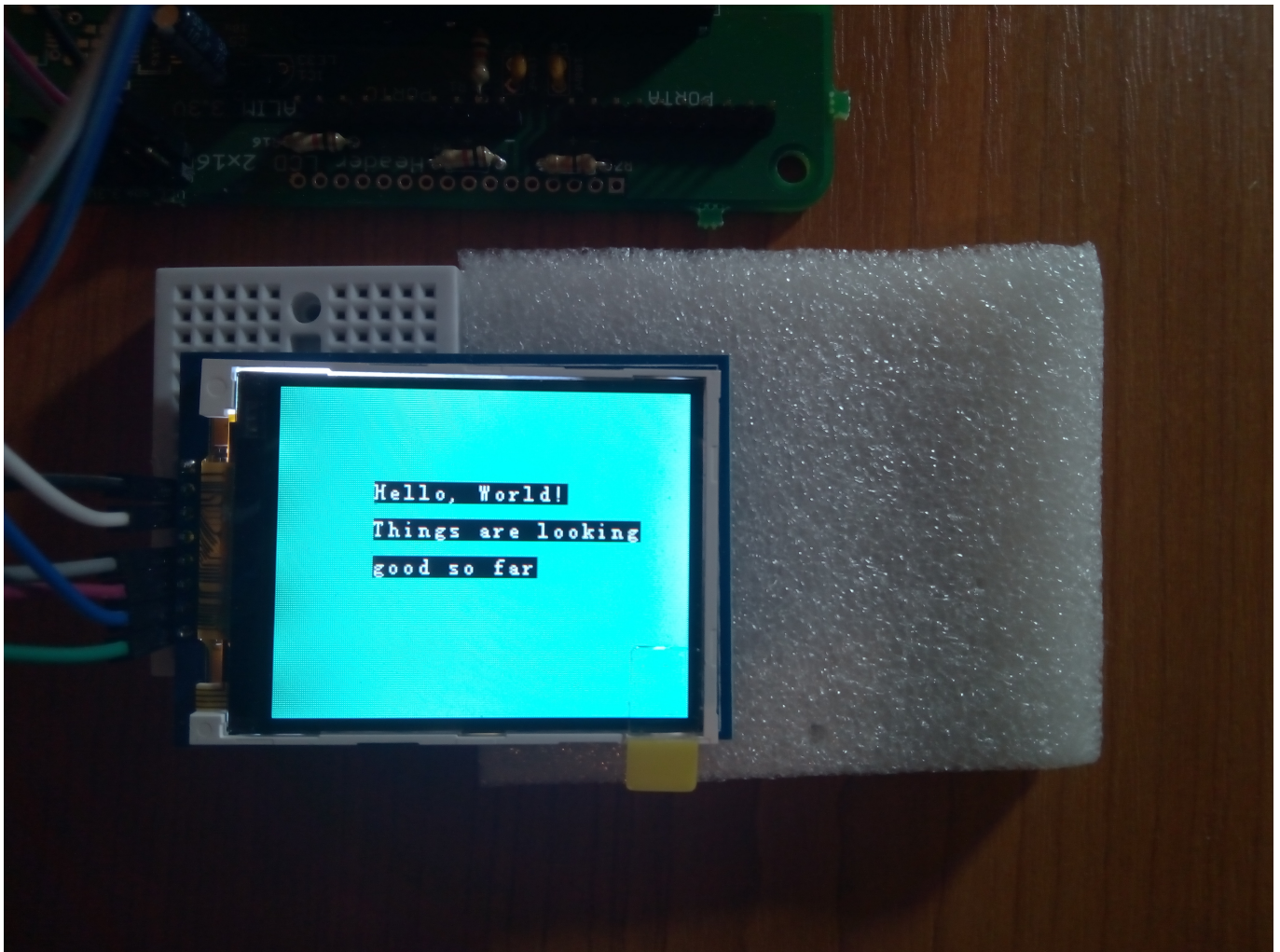
## Concluzii

## Download

## Jurnal

- 23/04 - schema bloc a proiectului + lista de piese
- 28/04 - lipirea componentelor pe placa de bază
- 07/05 - schema electrică a proiectului
- 14/05 - stadiu incipient de funcționare
- 21/05 - portare bibliotecă LCD în C

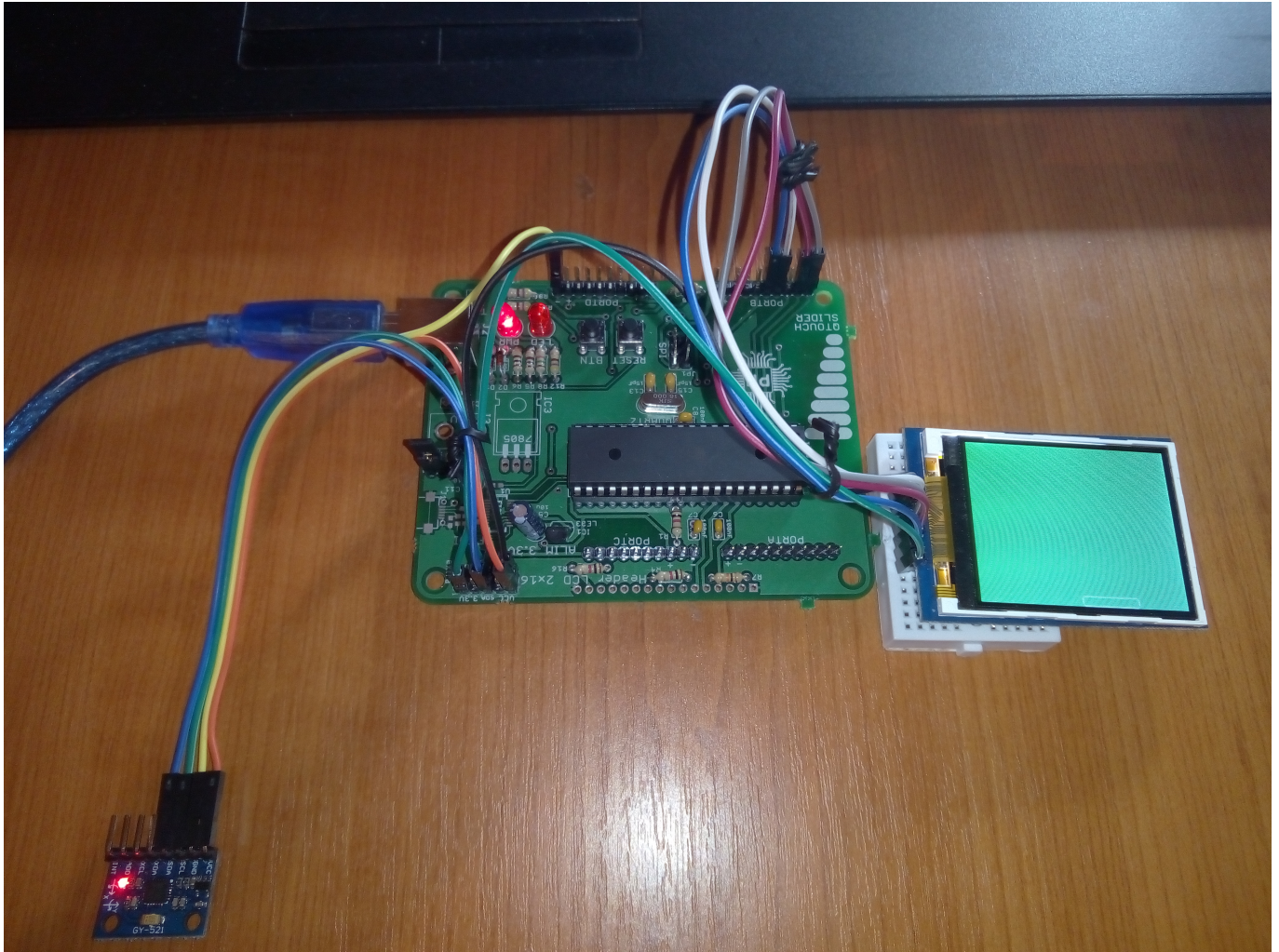
<imgcaption hello\_world|Hello, World!>



</imgcaption>

- 22/05 - debugging inițial QTouch
- 23/05 - renunțare la integrarea QTouch slider-ului în favoarea accelerometrului, pentru îndeplinirea aceluiași scop din cauza unor probleme de achiziție de date

<imgcaption setup\_actual|Setup actual: (accelerometru -  $\mu$ C - LCD)>



</imgcaption>

- 24/05 - schițarea logicii jocului în sine

## Bibliografie/Resurse

- [Datasheet controller LCD - HX8340-B](#)
- [Datasheet modul LCD - ITDB02-2.2SP](#)
- [Schematic LCD](#)
- [Dataseel accelerometru - MPU-6050](#)
- Documentația în format [PDF](#)

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

Permanent link:  
[http://ocw.cs.pub.ro/courses/pm/prj2017/ddragomir/dx\\_ball](http://ocw.cs.pub.ro/courses/pm/prj2017/ddragomir/dx_ball)



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

