

Iustin SÎRBU (78255) - Bingo

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

Introducere

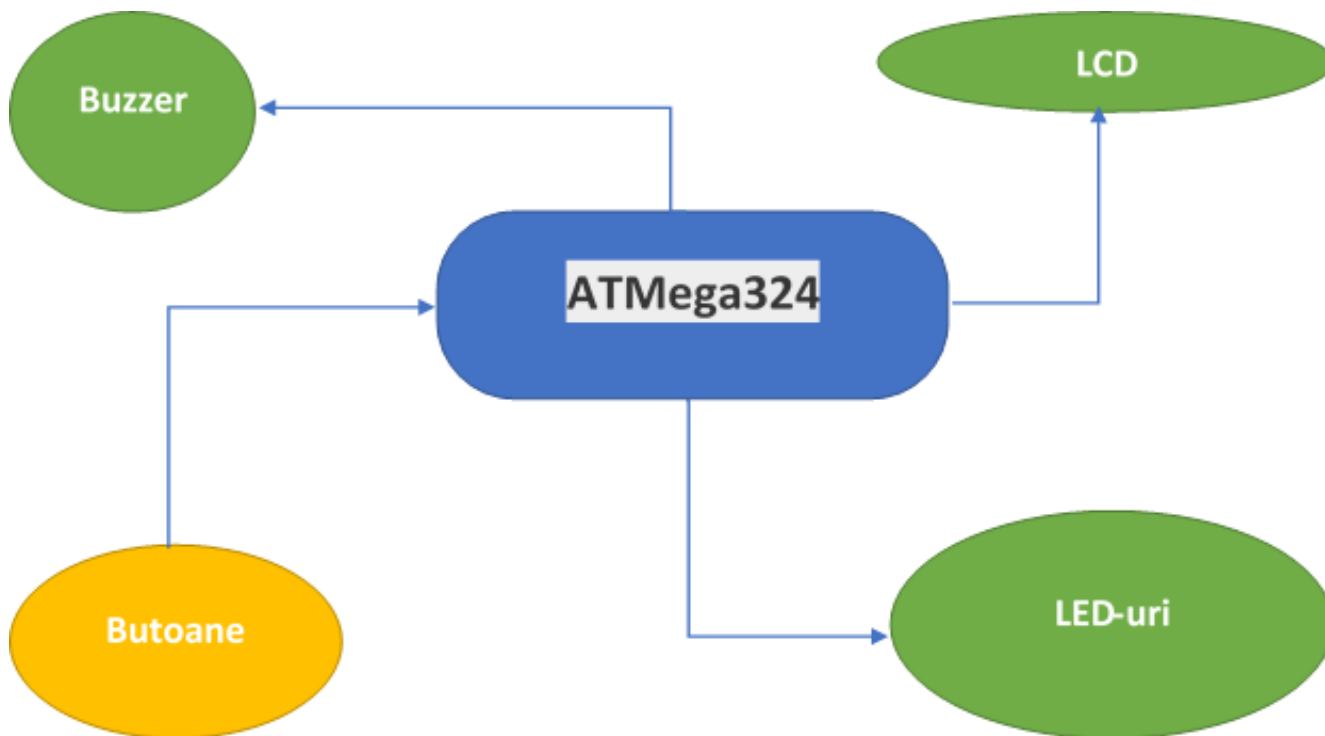
Bingo este un joc care decurge în felul următor: fiecare jucător are un tabel 5×5 în care ordonează aleator numerele de la 1 la 25. Pe rând, fiecare jucător alege un număr, iar acesta este colorat de toți participanții la joc. Castiga primul jucător care reușește să coloreze astfel 5 linii (acestea pot fi linii, coloane sau diagonale ale matricei), victoria fiind anunțată prin strigarea numelui jocului: "Bingo!".

În timp ce versiunea tradițională constă în desenarea tabelelor pe hârtie și hasurarea numerelor cu pixuri/creioane, versiunea curentă va folosi led-uri așezate sub forma unei matrice 5×5 în loc de tabelul clasic. Atunci când oponentul va selecta un număr, acesta se va colora automat și în tabelul propriu, prin aprinderea LED-ului corespunzător. Vor exista butoane care se vor folosi la selectarea numărului care se dorește a fi colorat, afișaj electronic (display) pentru a se menționa starea în care se află jocul (al cui este rândul, cine a câștigat), poate chiar și un buzzer care să sune atunci când cineva câștigă.

Această implementare susține Pamantul prin economisirea de hârtie folosită de numărul mare de studenți/elevi care joacă (în timpul pauzelor, desigur). Un scop mai particular ar fi acela de a mă ajuta pe mine să mă familiarizez cu lucrul la astfel de proiecte (care implică și hardware, care au legătură cu Proiectarea cu Microprocesoare), lucru care se va dovedi util în viitor, iar pe termen mai scurt obținerea punctajului maxim la PM. :D

Descriere generală

Schema bloc:



Utilizatorul va folosi butoanele pentru a selecta numere, iar ca raspuns se vor aprinde led-urile corespunzatoare de pe toate tebelele de joc, se vor scrie pe LCD informatii despre starea jocului, iar buzzer-ul va anunta victoria.

Hardware Design

Lista piese:

Numar de piese	Nume piese	Observatii
1	Placa de baza cu ATmega324	
2	Matrice LED (MAX7219)	una per jucator
6	Butoane	3 per jucator
1	LCD	
18	Fire	9 per jucator
1	Plexiglas	O singura bucata, indoita la cald
2	Placuta de test	una per jucator
12	Distantieri mici	4 per jucator, 4 placa de baza
4	Distantieri mari	
8	Suruburi	
12	Piulite	
1	Cablu de alimentare	
1	Baterie externa	ASUS ZenPOWER
2	Header pini	

Schema electrica:



Schema electrica a fost realizata in Eagle AutoDesk. Fiecare jucator are o telecomanda formata dintr-o matrice LED si 3 butoane. Unui jucator ii corespund matricea MAX7219(1) si butoanele S4, S5 si S6, iar celuilalt matricea MAX7219(2) si butoanele S1, S2 si S3.

Software Design

Mediu de dezvoltare: Atmel Studio 7.0

Bootloader: HID Boot Flash

Biblioteci: biblioteca pentru MAX7219, transcrisa pentru AVR (era pentru Arduino), fisierele pentru LCD din laborator, plus:

```
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
```

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

Modificări Header LCD

```
/*
 * Define-uri pentru parametrizarea bibliotecii LCD
 */
// Portul pe care conectam pinii de date ai LCD-ului
#define LcdDATA_DDR      DDRC
#define LcdDATA_PORT     PORTC
#define LcdDATA_PIN     PINC

// Pinii de date ai LCD-ului
#define LcdD4             PC6
#define LcdD5             PC5
```

```
#define LcdD6          PC4
#define LcdD7          PC3

// Portul pe care conectam pinii de control ai LCD-ului
#define LcdCTRL_DDR    DDRA
#define LcdCTRL_PORT   PORTA
#define LcdCTRL_PIN    PINA

// Pinii de control ai LCD-ului
#define LcdRS           PA5
#define LcdRW           PA6
#define LcdE            PA7
```

Funcții folosite:

```
Pentru controlul matricelor LED (pe langa cele de biblioteca):
void max7219_init(uint8_t player);
void update_display(uint8_t player);
void image(const __flash uint8_t im[8], uint8_t player);
void set_pixel (uint8_t r, uint8_t c, uint8_t value, uint8_t player);
```

Pentru desfasurarea jocului:

```
uint8_t find_value_of (uint8_t row, uint8_t col, uint8_t player);
uint8_t check_row (uint8_t row, uint8_t player);
uint8_t check_col (uint8_t col, uint8_t player);
uint8_t check_main_diag (uint8_t row, uint8_t col, uint8_t player);
uint8_t check_other_diag (uint8_t row, uint8_t col, uint8_t player);
void show_score(uint8_t player);
int check_all (uint8_t row, uint8_t col, uint8_t player);
uint8_t find_number(uint8_t row, uint8_t col, uint8_t player);
void find_position(uint8_t number, uint8_t *row, uint8_t *col, uint8_t
player);
void share_number (uint8_t row, uint8_t col, uint8_t player);
void initialize_game(void);
int main(void);
```

Fiecare jucator are pe ecranul dat de matricea led un chenar. La fiecare tura (specificata explicit pe display, iar pe matrice printr-un led care palpaie), un jucator se poate misca pe fiecare din axele oX si oY folosind doua butoane.

Atunci cand a ales patratelul pe care sa il hasureze el apasa butonul pentru OK. Se coloreaza patratelul sau si patratelul adversarului corespunzator, generat random la inceputul jocului, in etapa de initializare. (La inceput, fiecarui patratel al unui jucator i se asociaza in mod unic un numar de la 1 la 25, stiut doar de microprocesor, asa ca se cauta numarul respectiv in matricea adversa pentru a determina pozitia ledului corespunzator).

Scorul apare pe matrice sub forma unor beculete care se aprind la fiecare linie completata (fiecare jucator isi stie doar punctele sale), iar atunci cand cineva ajunge la 5 puncte, pe ecran ii apare o inimioara formata din led-uri aprinse, iar adversarului o inima franta :(

Rezultate Obținute

 [Click for video](#)

Chenarul de luminate care palpaie inca de la inceput reprezinta marginile chenarului 5x5. ACeستا pot fi setate sa stea aprinse sau stinse de catre fiecare jucator, individual, dupa preferinta.

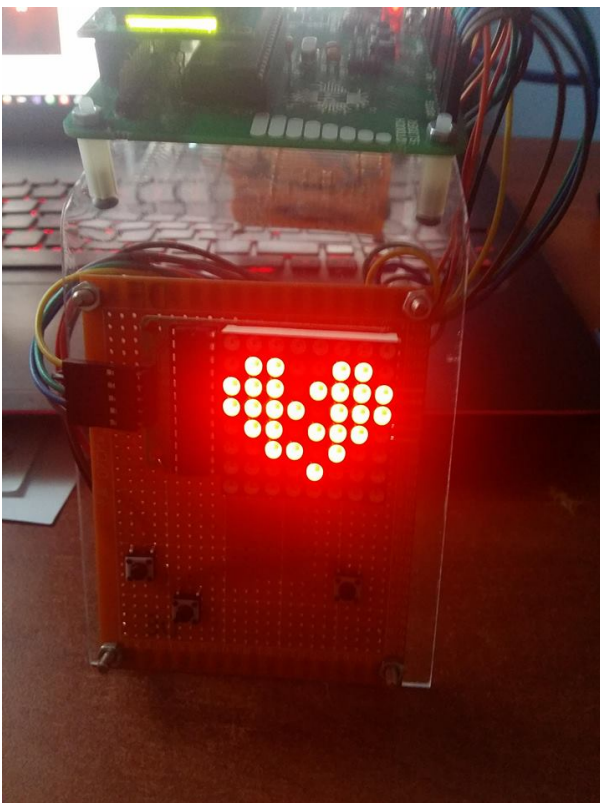
Cele doua LED-uri aprinse de pe margine reprezinta pozitiile curente pentru OX si OY, intersectia lor fiind cea colorata la apasarea butonului ok. Pozitia se poate modifica din butoanele de directie al fiecarui jucator. LED-ul din cltul dreapta-jos ii apare palpand doar jucatorului al carui rand este la mutare (palpaie diferit fata de chenarul suprafetei de joc, mai repede).



Pe LCD apare de asemenea al cui este randul, iar la final cine a castigat.



Chenarul de incadrare al tabelului are initial colturile lasate stinse. Odata cu fiecare linie completata de un jucator, acestuia i se va aprinde un colt al chenarului (deci pana la 4 puncte), urmand ca al 5-lea punct sa anunte victoria intr-un mod diferit. Astfel, jucatorii vor sti mereu cat de aproape sunt de victorie.



Inima din LED-uri apare la jucatoul victorios (amandoi in caz de egal), iar o inima franta ii apare pierzatorului. [Click to see how victory looks like](#)

Dispozitivul poate fi extins cu usurinta pentru a accepta jocuri noi, care pot fi implementate pe matrice de dimensiune maxima 8x8 (avioanele, X & O), iar prin apasarea butonului de pe placuta de baza sa se faca comutarea intre jocuri.

Concluzii

La inceput am fost enervat de faptul ca nu mi s-a echivalat proiectul, mai ales cand am aflat ca altii in aceeasi situatie au beneficiat de acest lucru. Totusi, pe parcurs a inceput sa imi placa ceea ce faceam, iar produsul final ma fascineaza. Nu credeam ca voi reusi sa fac totul asa cum mi-am propus (am renuntat totusi la buzzer, la sugestia asistentului, scopul lui fiind minor).

Problemele intampinate pe parcurs au fost legate de faptul ca matricile led nu functionau decat pe portul B si a durat pana sa imi dau seama de asta, iar unii pini de la portul D nu au functionat la conectarea butoanelor pe ei si din nou am avut de munca sa fac debug.

In ciuda acestor probleme consider ca m-am descurcat bine si am ajuns la un rezultat de apreciat.

Download

Jurnal

- **23 Aprilie** Tema proiectului, lista de piese și schema bloc
- **6 Mai** Schema electrică a proiectului
- **23 Mai** Versiune finală

Bibliografie/Resurse

Resurse Hardware

- [Datasheet Atmega 324](#)
- [Schemă electrică LCD](#)
- [Placa de baza pm 2018](#)

Resurse Software

- [Laborator 0 PM](#)
- [Laborator 1 PM](#)
- [Laborator 2 PM](#)
- [Laborator 3 PM](#)
- [Laborator 4 PM](#)
- [MAX7219 Library Arduino](#)

- [Atmel Studio 7](#)
- [HID BOOT Flash](#)

- Documentația în format [PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2018/abirlica/bingo13>



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