Ştefania BUCUR (78536) - Memory Game

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

Introducere

Proiectul presupune implementarea unui Memory Game, ce se bazează pe redarea prin led-uri a unor secvențe continue și reproducerea ordinii acestora prin apăsarea butoanelor. Jocul va avea mai multe runde, iar dificultatea va crește după fiecare nivel depășit. Jucătorul trebuie să refacă în totalitate secvența și va trece la nivelul următor, altfel pierde și jocul se reia.

Am plecat de la ideea construirii unui joc care să fie și util în același timp. Scopul acestui joc este de a antrena memoria jucătorului într-un mod plăcut și interactiv, dezvoltând totodată capacitatea de concentrare a acestuia.

Descriere generală

Schemă bloc:

Descriere:

Jocul va conține 8 leduri, 8 butoane, un buzzer prin care fiecare led va avea asociat o notă la aprindere. De asemenea, după fiecare nivel trecut sau după pierderea jocului, se va reda o scurtă melodie de la buzzer. Jocul va avea și un ecran LCD pe care vor fi afișate informații despre starea în care se află jucătorul: mesaj de început, nivel curent, mesaj se felicitare pentru trecerea la nivelul următor, mesaj de final de joc.

Flow joc:

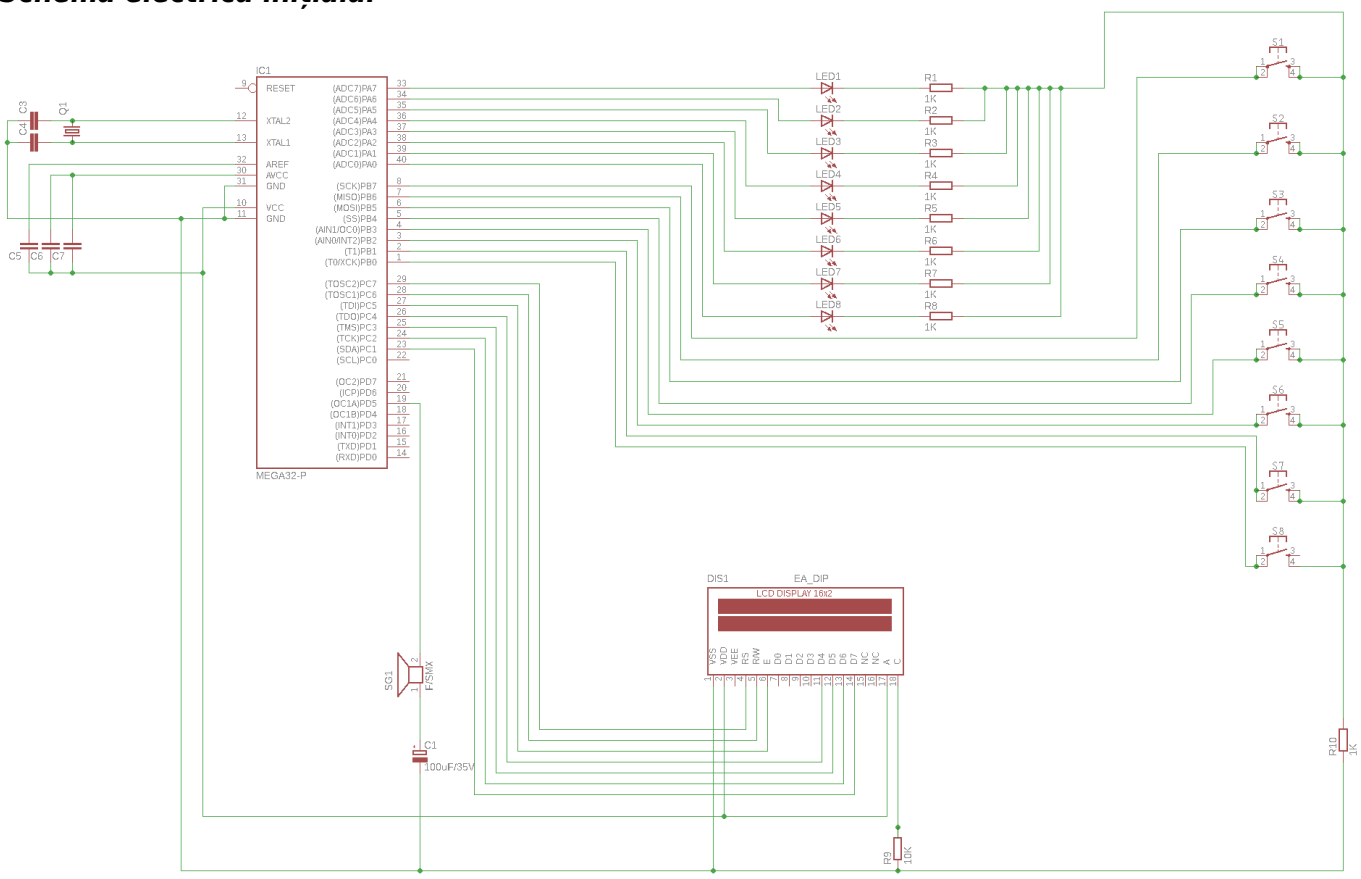
```
1. Jocul începe odată cu aprinderea plăcuței.
2. Inițial, se generează o secvență de un singur led, iar după fiecare rundă,
se vor aprinde +1 leduri față de runda anterioară.
3. Jocul va fi pe mai multe nivele de dificultate, secvența făcându-se din
ce în ce mai mare pe măsură ce jucătorul înaintează în joc.
4. Jucătorul va trece în runda următoare doar dacă apasă pe butoane în
ordinea în care au fost luminate ledurile. În caz contrar, jocul se termină.
```

Hardware Design

Listă piese:

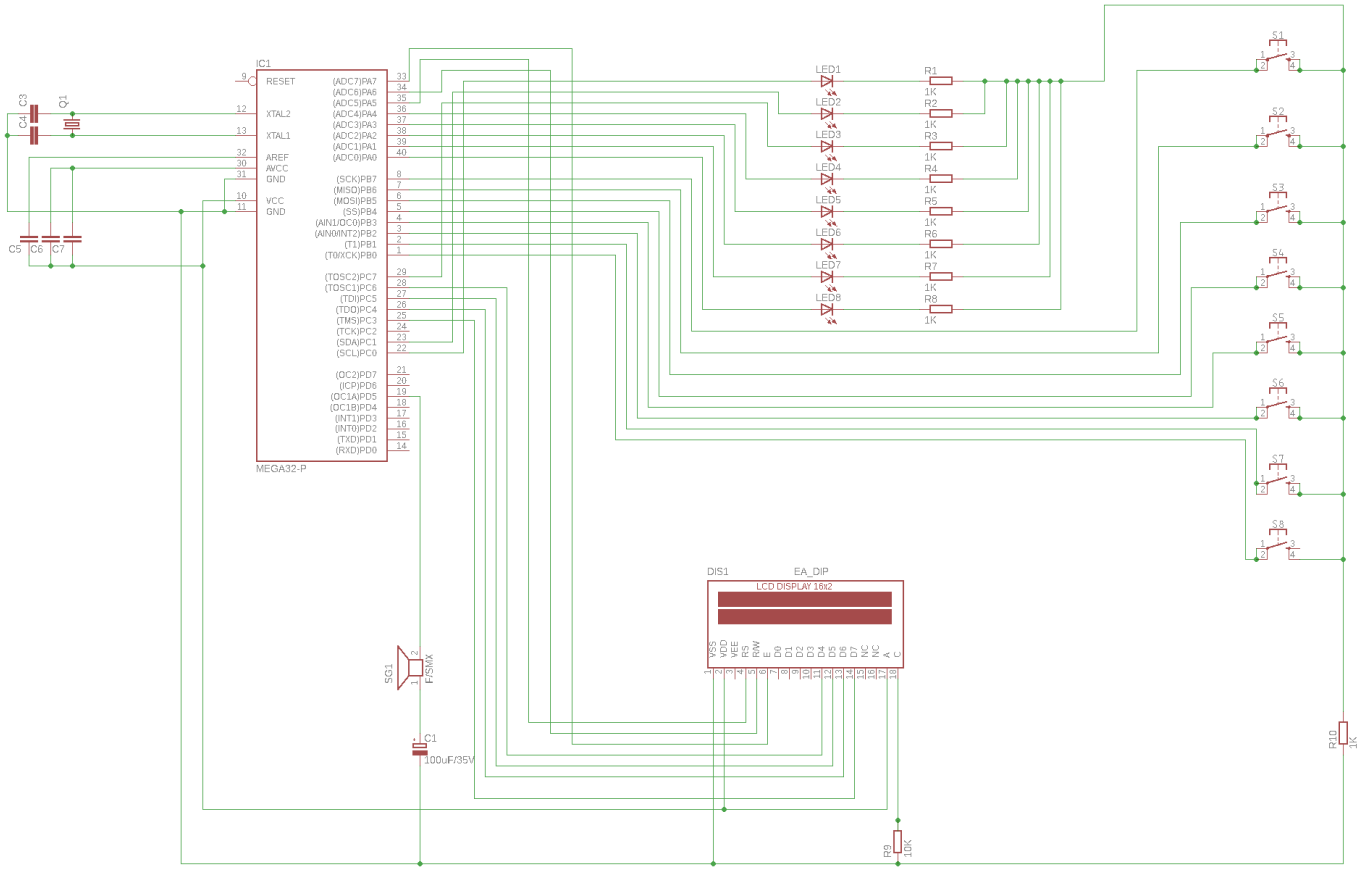
Denumire Componentă	Număr	Preț
Placă de bază	x1	8 RON
ATmega 324A + piese obligatorii	x1	24 RON
Modul LCD de 2×16 cu controller Hitachi	x1	10.49 RON
Leduri monocolor 5mm	x8	4 RON
Butoane 6x6x6	x8	8 RON
Buzzer activ de 5 V	x1	3.9 RON
Rezistente de 1 kΩ	x9	1 RON
Rezistente de 10 kΩ	x1	0.1 RON
Fire de legatura de tip mama-mama	x40	16 RON
Condensator 100uF	x1	0.5 RON
Headeri de pini	x40	2 RON
Cablaj de test	x1	5 RON

Schemă electrică inițială:



Nu am reușit să conectez cele 8 leduri doar la PORTA din cauza LCD-ului deoarece pinii de control sunt conectați la PA5, PA6 și PA7. Așadar, cele 3 leduri rămase au fost conectate la PC7, PC0 și PC1.

Schemă electrică finală:



Software Design

1. Mediu de dezvoltare:

1. Eagle 8.2.0 - circuitul electric
2. Programmer's Notepad
3. HID Boot Flash

2. Biblioteci incluse:

```
1. include <avr/io.h>
2. #include <stdint.h>
3. #include <stdlib.h>
4. #include <string.h>
5. #include <stdio.h>
6. #include <util/delay.h>
7. #include <avr/interrupt.h>
8. #include "lcd.h" - din laborator
```

3. Implementare:

Partea de **LCD** este preluată din laborator. Am folosit functiile LCD_Init pentru inițializarea LCD-ului, precum și LCD_printAt pentru afișarea statusului curent al jocului. O modificare pe care am adus-o în fișierul lcd.h este schimbarea pinilor de control ca în schema finală.

```
  #define LcdRS                   PA5
  #define LcdRW                   PA6
  #define LcdE                    PA7
```

În fișierul **memory_game.c**, am realizat implementarea efectivă a jocului.

Inițial, definesc numărul maxim de runde, numărul de leduri și un număr cu care va fi apelată funcția srand din main.

```
#define RANDOM_NO 2000
#define NO_LEDS 8
#define RANDOM_NO 2000
```

Constanta RANDOM_NO este utilă deoarece poate fi schimbată și se poate genera o altă secvență a jocului. Dacă această nu este schimbată, jocul va genera mereu aceeași secvență, srand() primind același seed după fiecare restart al jocului. Ideal, ar fi fost srand(time(NULL)), dar nu există funcția time în această versiune de avr-gcc.

Funcțiile cu care am realizat implementarea jocului sunt:

```
/*activare porturi*/
void ports_activ();

/*lucru cu întreruperi*/
void timer_init();
void tone();
void no_tone();

/*acțiune după câștigarea fiecărei runde*/
void win_round(char display_text[]);

/*acțiune după pierderea jocului*/
void lose_game(char display_text[], char convert_round[]);

/*afișare pe lcd nivelul curent și câte vieți mai sunt la dispoziție*/
void print_level(char display_text[], char convert_round[], char lives[]);

/*aprindere led însoțit de o notă dată de buzzer*/
void light_led_with_sound(int led_index);

/*așteptare introducere secvență de jucător*/
void wait_for_player(int seq[], char convert_round[], char display_text[],
int round);

/*funcție principală ce controlează flow-ul jocului*/
void play_game();
```

Detalii implementare:

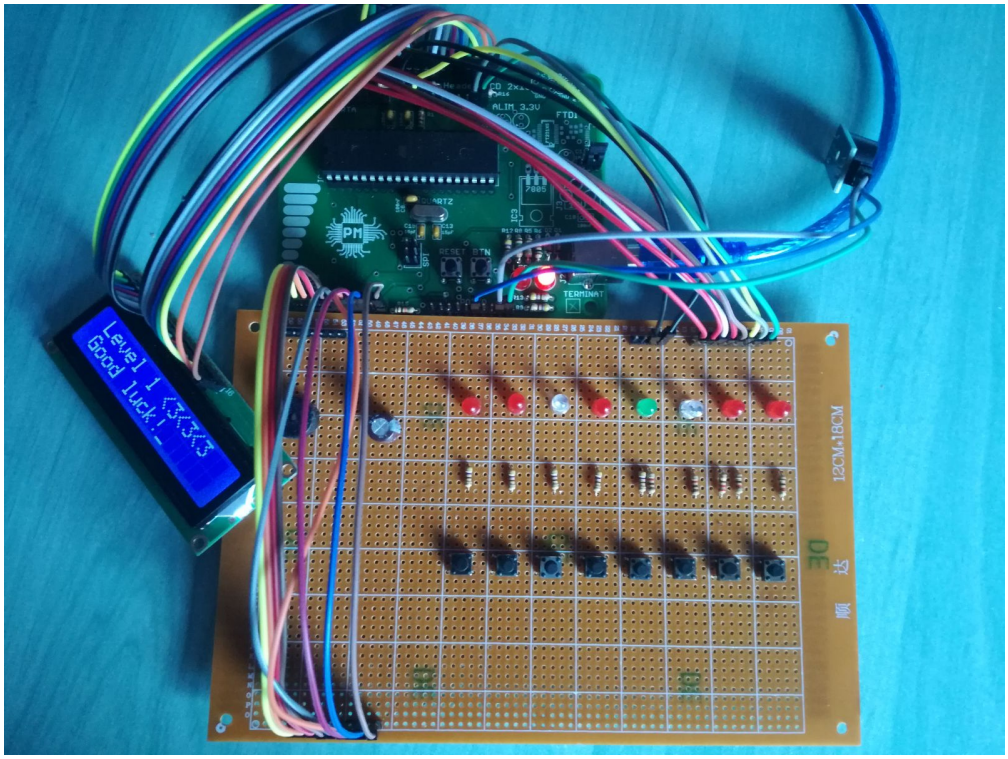
- Jocul începe imediat cu aprinderea plăcuței.
- Se generează o secvență de leduri, pornind de la un led și crescând numărul lor cu 1. Indexul ledului

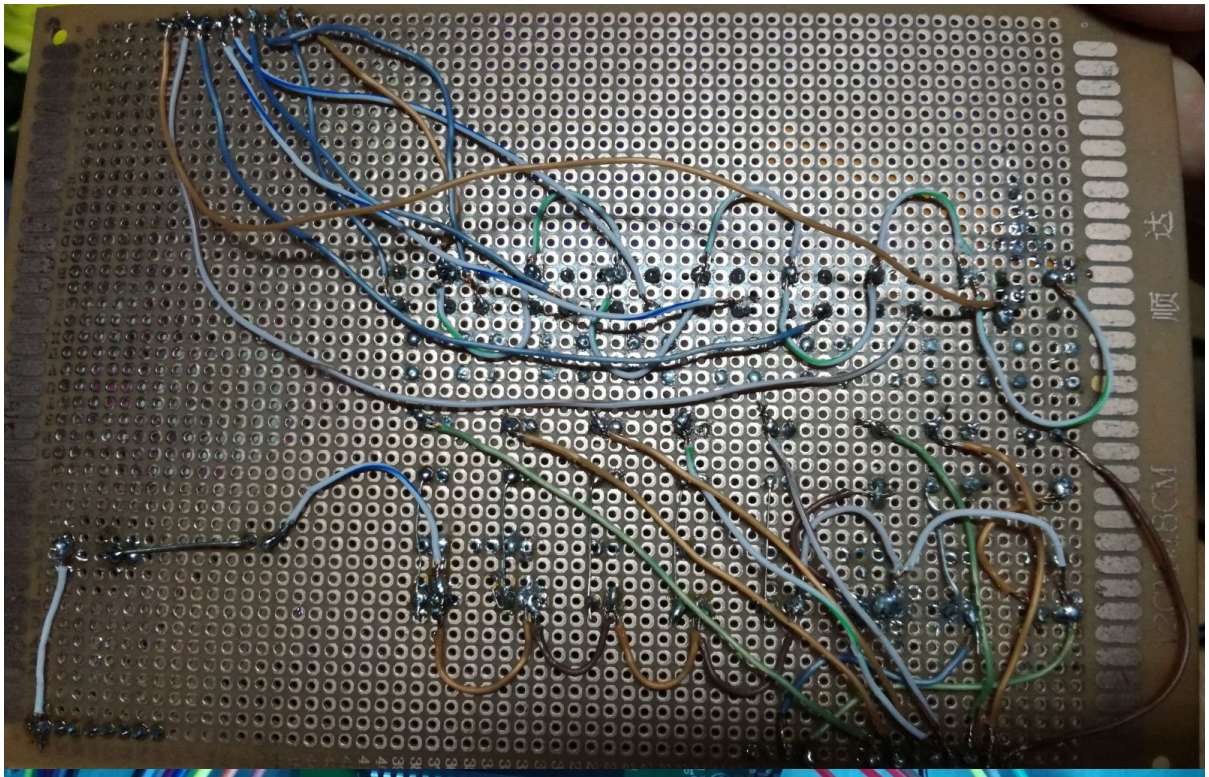
se afla prin **rand() % NO_LEDS;**.
- Odată generată secvența, se vor aprinde ledurile cu sunet în ordinea salvată.
- Se așteaptă inputul de la butoane al jucătorului și se verifică dacă a introdus butonul corespunzător ledului care a fost aprins.
- Dacă s-a apăsat toata secvența, se trece la secvența următoare, rezultând modifcarea nivelului de pe lcd și realizarea unui joculeț de lumini din leduri, acompaniate de buzzer.
- Inițial, se vor afișa pe LCD si viețile pe care le are jucatorul, și anume 3.
- Dacă jucatorul introduce un buton greșit de 3 ori, jocul de incheie.
- În cazul în care greșește o dată sau de două ori, dar dupa nimerește butonul corect, se va afișa pe LCD numărul de vieți rămase.
- La terminarea jocului, atunci când jucătorul greșește, va apărea pe LCD nivelul la care a ajuns și se va reda o melodie de terminare.

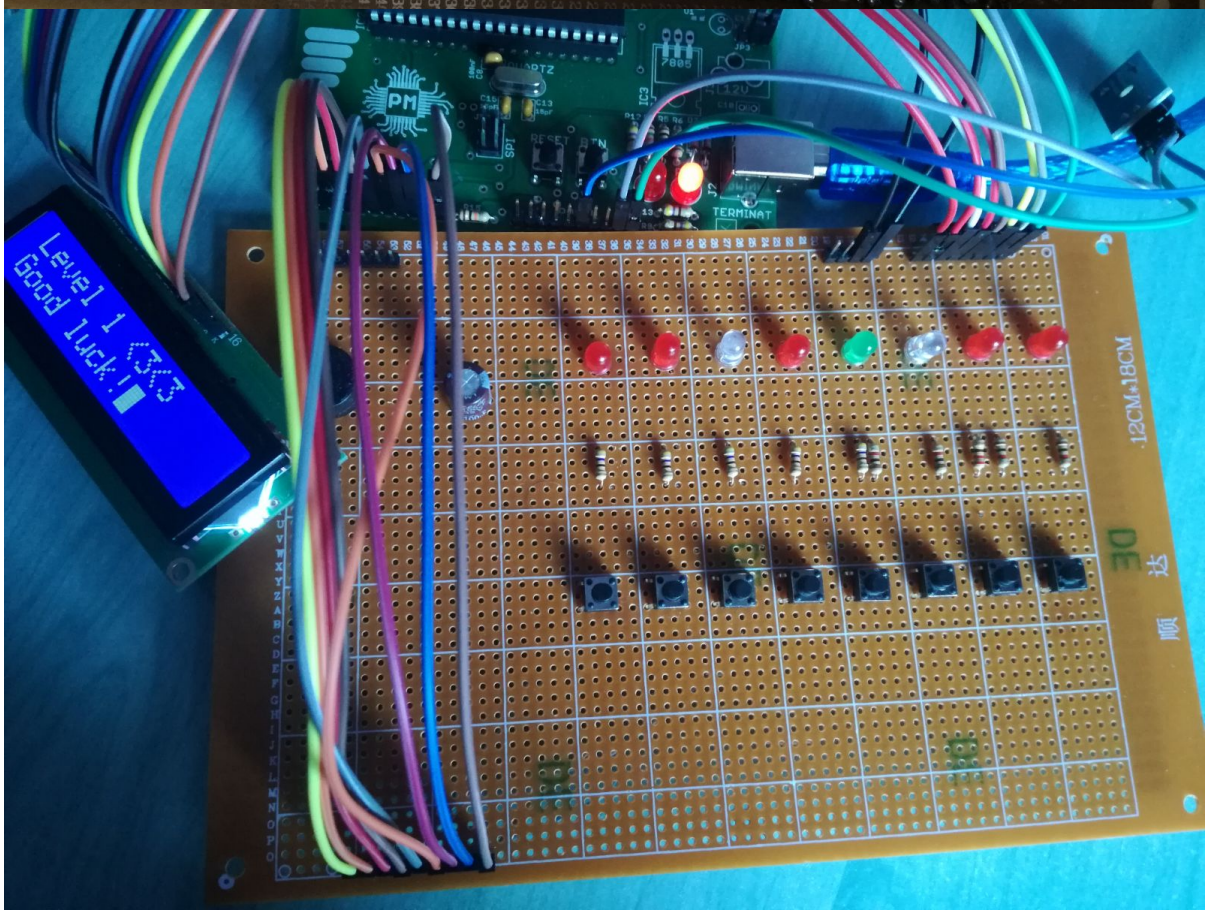
Probleme întâmpinate

- Am schimbat ledurile de două ori deoarece luminau foarte slab.
- Am adăugat noi rezistențe căci credeam că de la ele e problema de mai sus.
- Buzzer-ul nu făcea contact, așa că am cumpărat un modul de buzzer activ nou.

Rezultate Obţinute






Level 1
Good luck!
TERMINAT
RESET BTN
12CM*18CM

Un video demo este aici.

Concluzii

A fost un proiect interesant, de la care am învățat foarte multe atât pe partea de hardware, cât și de software. Pot spune că am avut ghinion cu fiecare componentă de pe placuță, nimic nu a funcționat bine din prima, dar până la urmă, toate s-au rezolvat.

Ca o concluzie personală, nimic nu este cu adevărat frumos dacă nu te chinui la el.

Download

Sursele proiectului

Jurnal

- **21.04.2018**: Alegere temă proiect și completare informații pentru Milestone1.
- **04.05.2018**: Finalizare placă de bază.

- **06.05.2018**: Milestone3: realizarea schemei electrice și descrierea pieselor hardware necesare.
- **11.05.2018 - 21.05.2018**: Lipire componente pas cu pas și realizarea de modificări hardware.
- **21.05.2018 - 22.05.2018**: Implementare software și modificare schemă electrică.
- **23.05.2018**: Completare pagină de Wiki cu descrierea implementării proiectului.

Bibliografie/Resurse

- Laboratorul 0 PM
- Laboratorul 1 PM
- Laboratorul 2 PM
- Atmega 324 Datasheet
- Proiectele din anii trecuţi

Schema electrică Eagle: memory_game.sch

- Documentația în format PDF