Mini Consola

Autor

Dragoș-Cristian Bute

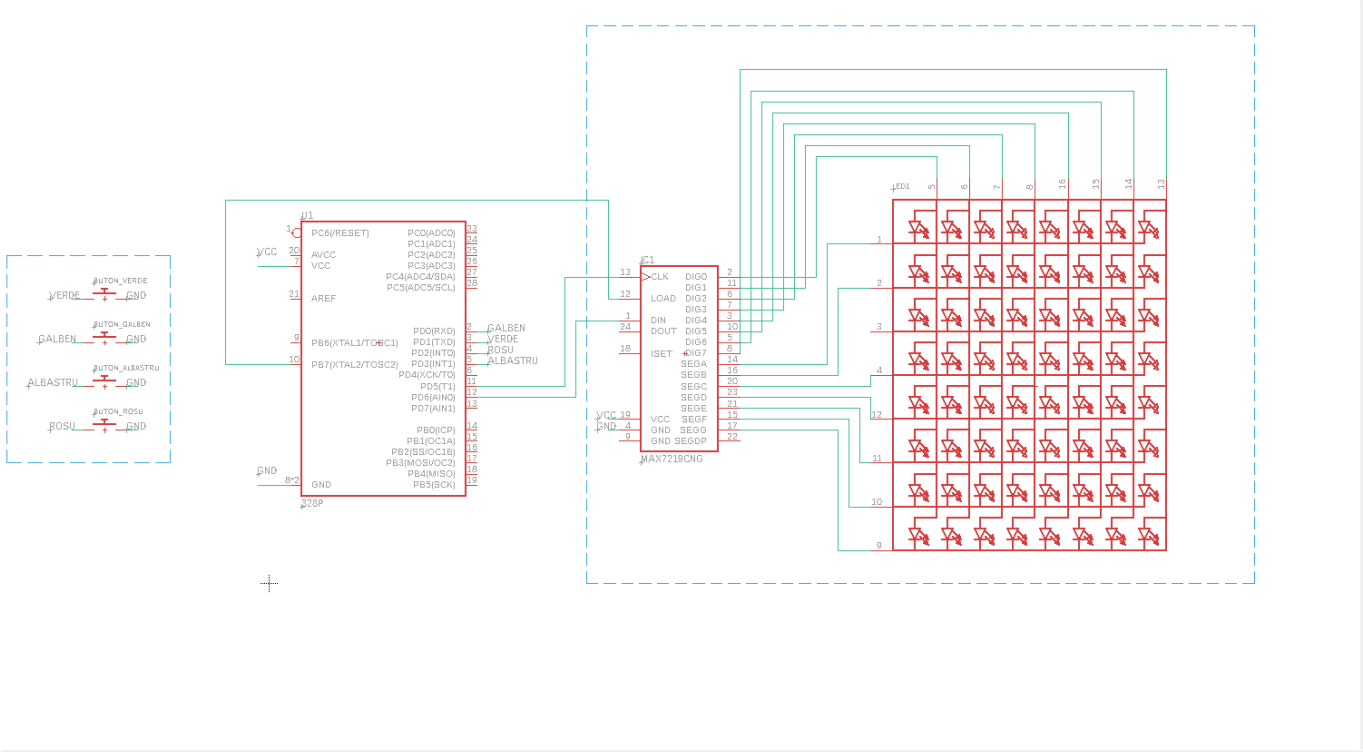
Introducere

Proiectul meu consta intr-o mini-consola capabila sa joace unul sau mai multe jocuri. Pentru demostrarea conceptului am implementat un joc de tip labirint(totul minimalist, display–ul fiind un ecran format din mai multe led-uri, 8×8 mai exact). Principala idee este aceea ce a realiza framework-ul necesar implementarii diferitor jocuri, adaugarea de noi jocuri fiind posibila dupa realizarea consolei propriu-zisa.

Descriere generală

Schema bloc:

Schema electrica:



Funcționalitate

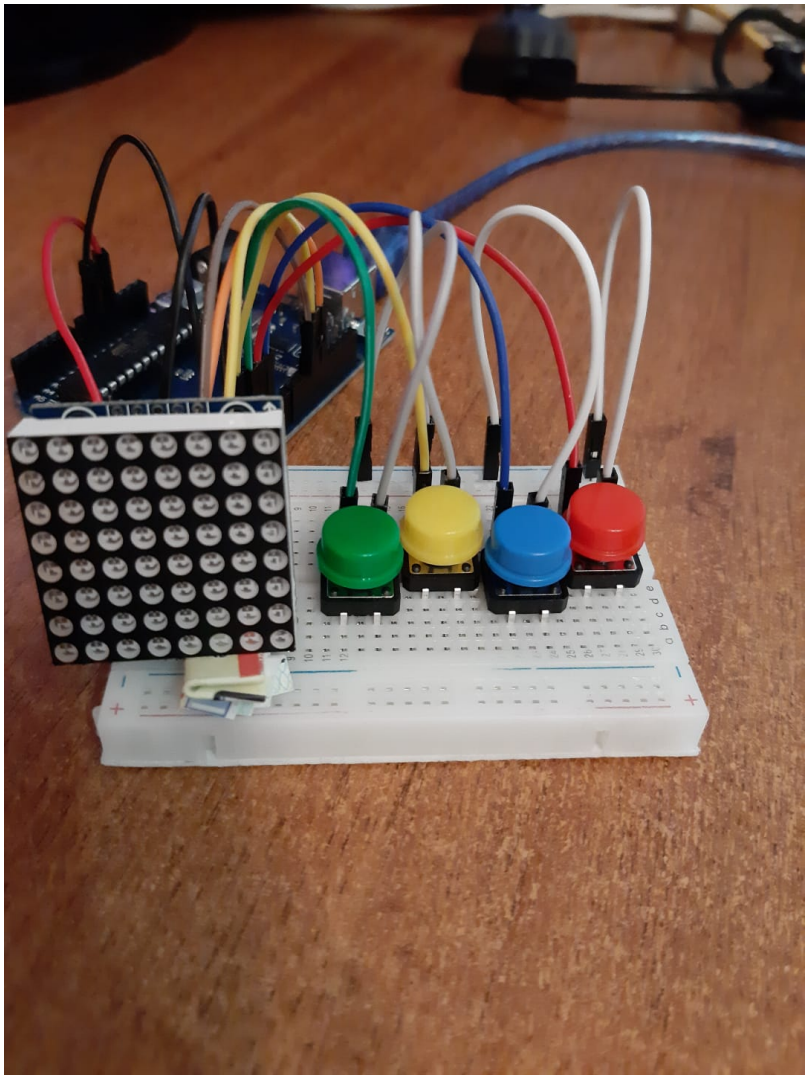
Consola va dispune de 4 butoane, acestea pot avea mai multe semnificatii insa principala folosire a acestora va fi data de directiile sus, jos, stanga, dreapta. Ecranul va fi o matrice de led-uri pentru a-mi usura viata cat si pentru a introduce o constrangere ce sper eu, va spori creativitatea. O data ce va fi pornita consola, jucatorul va intra direct in joc, pentru schimbarea jocului fiind necesara o schimbare a codului.

Hardware Design

Listă de piese

- Arduino Uno
- Ecran (Modul matrice LED MAX7219)
- Buton x 4
- Fire tata/tata
- Breadboard

Cum arată



Dupa cum se poate vedea, din cauza dimensiunii mici a breadboard-ului (cand am cumparat-o, in poza arata mai mare), totul e destul de cramped, dar are un șarm putin cyberpunk.

Modulul matrice LED are niste pini ce ar fi trebuit lipiti, eu nu stiu/nu am cu ce sa fac asta, asa ca e nevoie de o mica bucatica de bilet de tren pentru a merge :).

Software Design

Mediul de dezvoltare

Dat fiind faptul ca IDE-ul Arduino, dupa parerea mea e oribil, am folosit VScode cu extensia de Arduino.

Ideea principala

Ideea principala este de a avea o clasa abstracta/interfata Game pe care sa trebuiasca sa o extinzi

pentru a crea un nou joc.

Implementare

Clasa de baza Game

```
class Game {
    protected:
        LedControl *lc;

    public:
        void setLc(LedControl &lc) {
            this->lc = &lc;
        }
        virtual void upButtonAction() = 0;
        virtual void downButtonAction() = 0;
        virtual void leftButtonAction() = 0;
        virtual void rightButtonAction() = 0;
        virtual void displayVictory() = 0;
        virtual void setDisplay(byte image[]) = 0;
        virtual bool checkWon() = 0;
};
```

Pentru a implementa un nou joc este necesara implementarea metodelor virtuale declarate in clasa Game, astfel ne vom asigura ca fiecare joc face ceva cu butoanele pe care le avem (in cazul in care nu dorim asta, putem sa facem o functie goala), ca exista un mod in care sa vizualizam victoria in joc, ca se poate ajunge la victorie si ca putem sa setam display-ul astfel incat sa avem ceva pe ecran.

LedControl lc este obiectul prin care interactionam cu display-ul (matricea LED), motiv pentru care avem nevoie de el in clasa.

Exemplu implementare o functie

```
class MazeGame : public Game {
    public:
        void setDisplay(byte image[]) override {
            for(int i = 0; i < 8; i++) {
                lc->setColumn(0,i,image[i]);
            }

            lc->setLed(0, player.x, player.y, true);
        }
        void upButtonAction() override {
            static unsigned long last_interrupt_time = 0;
            unsigned long interrupt_time = millis();
```

```
            //Debounce
            if (interrupt_time - last_interrupt_time > 100) {
                lc->setLed(0, player.x, player.y, false);
                player.y--;
                if (player.y > -1) {
                    if (checkColliding()) //Implemented in the actual code
                        player.y++;
                    } else {
                        maps.current_map = maps.maps[(maps.current_map_y - 1)
* N_MAPS + maps.current_map_x];
                        maps.current_map_y--;
                        player.y = 7;
                    }
                lc->setLed(0, player.x, player.y, true);
            }
            last_interrupt_time = interrupt_time;
        }
    /*
     *
     * etc
     */
}
```

Dupa cum se poate observa, suprascrierea este simpla.

Un lucru de remarcat este debounce-uirea apasarii butoanelor, acest lucru regasindu-se in toate functiile *ButtonAction().

Sketch-ul efectiv

```
#include "LedControl.h"
#include "PinChangeInterrupt.h"
#include "game.h"

LedControl lc=LedControl(12,11,10,1);

MazeGame mazeGame;

void setup() {
  lc.shutdown(0,false);

  lc.setIntensity(0,3);

  lc.clearDisplay(0);

  mazeGame.setLc(lc);

  pinMode(UP_BUTTON, INPUT_PULLUP);
  pinMode(DOWN_BUTTON, INPUT_PULLUP);
```

```
  pinMode(LEFT_BUTTON, INPUT_PULLUP);
  pinMode(RIGHT_BUTTON, INPUT_PULLUP);

  attachInterrupt(digitalPinToInterrupt(UP_BUTTON),
[](){mazeGame.upButtonAction();}, FALLING);
  attachInterrupt(digitalPinToInterrupt(DOWN_BUTTON),
[](){mazeGame.downButtonAction();}, FALLING);
  attachPCINT(digitalPinToPCINT(LEFT_BUTTON),
[](){mazeGame.leftButtonAction();}, FALLING);
  attachPCINT(digitalPinToPCINT(RIGHT_BUTTON),
[](){mazeGame.rightButtonAction();}, FALLING);
}


void loop() {
  if (mazeGame.checkWon()) {
    mazeGame.displayVictory();
  } else {
    mazeGame.setDisplay(mazeGame.maps.current_map);
  }

}
```

Dupa cum se poate observa, tematica de simplitate se continua si in ceea ce putem numi “main”, in sketch doar instantiand atat jocul cat si LedControl-ul, setam pinii pentru input, atasam intreruperi pentru apasarea butoanelor (folosind functii lambda pentru ca sunt metode dintr-o clasa si nu merge altfel), iar in loop verificam daca s-a terminat sau nu jocul, caz in care afisam “grafica” corespunzatoare.

Schimbarea jocului se va face declarand jocul si schimband pe unde este necesar din mazeGame in numele jocului respectiv.

Codul din arhiva are si cateva comentarii, acestea sunt sterse aici pentru a ocupa mai putin loc.

Rezultate Obţinute

Rezultatele obtinute au fost pe masura asteptarilor, am reusit sa implementez ce imi doream, o mini-consola, invatand astfel sa lucrez atat cu elemente hardware noi, cat si cu cu concepte software precum intreruperi.

Demonstratie video a labirintului

Concluzii

In concluzie pot spune ca, privind din prezent in trecut, ar fi trebuit sa adaug si o componenta de

sunet, chiar si un buzzer, si ar fi fost frumos sa lipesc pinii de la display, cat si sa ii fac o carcasa, insa din lipsa timpului n-am facut asta.

Per total sunt multumit de proiect, a fost o experienta draguta care, cine stie, poate ma va ajuta si in viitorul indepartat.

Download

Arhiva cu codul

In arhiva putem vedea fisierul “arduino.ino” in care este codul efectiv pentru Arduino (Adica unde sunt setup si loop), si fisierul “game.h” cu clasa Game cat si cu implementarea jocului cu labirint, ideal ar fi fost sa fie un alt fisier .cpp cu implementarea si alte fisiere pentru jocul cu labirint, insa pentru usurinta mea si ca proof-of-concept am pus totul in acest fisier.

Jurnal

- 23 Aprilie → Alegerea temei proiectului
- 25 Aprilie → Creat pagina asta
- Candva in mai → Comandat piese
- Candva mai tarziu in mai → Terminat proiect (hardware/software)
- 02 Iunie → Terminat wiki

Bibliografie/Resurse

Biblioteca PinChangeInterrupt

Biblioteca LedControl

Download pagina asta