2048 game

Introducere

- Proiectul isi propune implementarea jocului clasic 2048
- Scopul jocului este sa atingi valoarea 2048
- Initial pornim cu un grid 4×4 cu o valoare de 2 intr-o casuta aleatoare
- Exista 4 tipuri de mutari: sus, jos, stanga, dreapta, si fiecare mutare consta intr-o shiftare a tuturor campurilor in directia aleasa. Campurile cu aceeasi valoare se aduna si se goleste unul din campuri. La fiecare miscare se adauga intr-o casuta goala aleatoare din grid o valoare de 2
- Jocul de termina cand nu mai exista miscari posibile (cand grid-ul s-a umplut) sau cand se atinge valoarea 2048

Exemplu joc clasic 2048

Descriere generală

Jocul va fi transpus sub forma unei matrici 4×4 pe un ecran LCD. Jucatorul va putea face mutari cu ajutorul unei telecomenzi cu infrarosu. Daca nici o casuta care nu este goala nu isi va schimba pozitia sau valoarea in urma unei mutari, nu se va mai genera o noua valoare de 2, mutarea fiind considerata invalida.

Schema bloc

Hardware Design

Componente folosite:

- Arduino UNO
- Ecran LCD color 1.8”
- receptor infrarosu
- telecomanda infrarosu
- breadboard

Pin wiring

Ecran LCD	**Arduino UNO**
BL	3.3V
CS	10
DC	9
RES	8
SDA	11
SCL	13
VCC	5V
GND	GND
IR receiver	**Arduino UNO**
SIG	7
GND	GND
VCC	5V
Arduino UNO	**Arduino UNO**
RES	2

Software Design

Codul a fost scris folosind Arduino IDE. Pentru comunicarea cu ecranul LCD, folosesc bibliotecile SPI.h si TFT.h, iar pentru utilizarea telecomenzii IR folosesc IRremote.h.

Pentru a primi semnale de la telecomanda IR, folosesc portul 7, iar pentru reset folosesc portul 2.

Jocul poate primi 4 inputuri din partea telecomenzii: sus, jos, stanga dreapta, reprezentate pe telecomanda prin 4 sageti sugestive. Fiecare din aceste comenzi va muta casutele in directia indicata. Pe langa aceste comenzi, jocul se poate reseta prin apasarea tastei 5. In cadrul unui joc 2048, aceste inputuri sunt suficiente. Totusi, din cauza regulilor, un joc dureaza destul de mult, de aceea in scopul prezentarii proiectului am adaugat 2 butoane ce cauzeaza castigarea respectiv pierderea unui joc. Un joc se considera castigat atunci cand se atinge valoarea 2048, si se considera pierdut cand nu mai exista mutari disponibile pe tabela. Aceste butoane sunt *4* si *6*. Primul buton insereaza valoare 2048 prima casuta din tabela, iar al doilea umple tabela cu valori ce nu mai pot fi adunate. Apasarea altor butoane decat cele mentionate nu va avea vreun efect, inputul fiind ignorat.

Codul este impartit in 3 parti:

- desenarea jocului pe ecran
- logica jocului si a mutarilor
- tratare input IR

Pentru desenarea jocului pe LCD am folosit biblioteca TFT.h, si functii proprii:

```
void drawBorders(void);    // Deseneaza liniile tabelei ce incadreaza
```

```
fiecare casuta
void drawGame(void);       // Deseneaza intreaga tabela
void colorPicker(int i);   // Alege culoarea unei casute in functie de
valoarea din ea
void displayScore1(void);  // Deseneaza pe ecran "Score" o singura data
void displayScore2(void);  // Deseneaza pe ecran valoarea scorului calculat
void gameEndMessage(void); // Afiseaza pe ecran un mesaj la finalul jocului
```

Pentru logica jocului si a mutarilor am ales o implementare proprie, care simuleaza regulile jocului clasic 2048:

```
/* Cele 4 mutari posiblie */
void moveDown(void);
void moveUp(void);
void moveRight(void);
void moveLeft(void);

void generateRandom(void);                // Genereaza o valoare de 2 intr-o
casuta aleatoare
int gameOver(void);                       // Verifica daca jocul s-a
terminat
int hasSpaces(int line, char direction);  // Functie ajutatoare pentru
mutari
```

Regulile jocului sunt simple: jucatorul poate face o mutare pe una din 4 directii posibile, si toate casutele din tabela se vor muta in directia aleasa, cu oricate pozitii este nevoie pentru a ajunge in capat. Casutele cu valori egale se aduna. Pentru a putea avansa in joc, este nevoie sa se genereze o noua valoare de 2 dupa fiecare mutare valida. O mutare este considerata valida daca cel putin o casuta si-a schimbat valoarea fie pozitia (daca in urma mutarii starea tabelei se schimba).

```
/*
 * Cod folosit pentru generatea unei valori de 2 pe o pozitie aleatoare
 * Va fi apelat dupa fiecare mutare valida si la inceputul jocului
 */
void generateRandom(void)
{
    while(1) {
        int x = random(4);
        int y = random(4);
        if (board[x][y] == 0) {
            board[x][y] = 2;
            break;
        }
    }
}
```

Tratarea inputurilor IR:

```
/*
 * Coduri pentru fiecare buton folosit de pe telecomanda
```

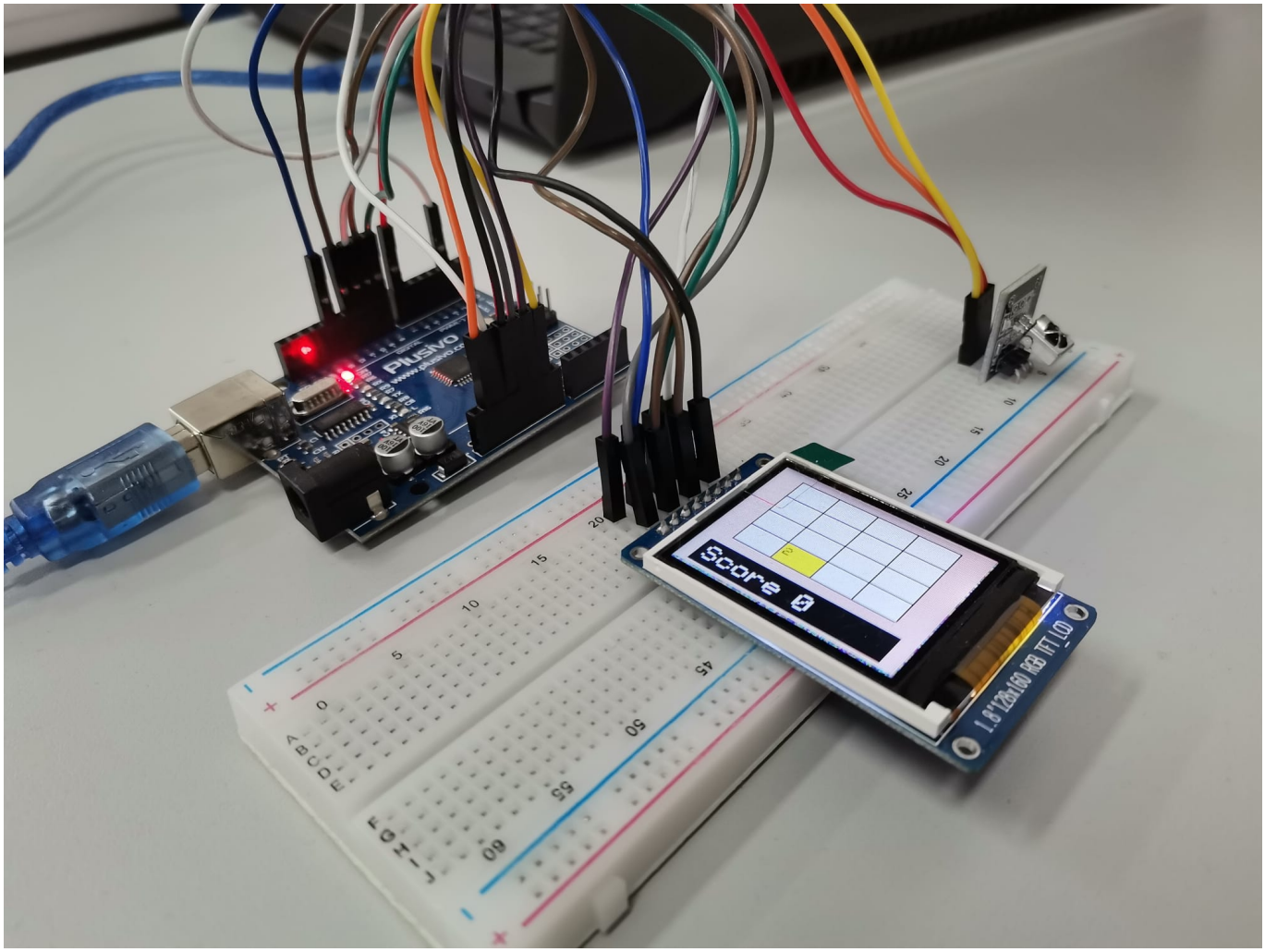
```
 * Primele 4 corespund mutarilor
 */
#define UP 0xFF18E7
#define DOWN 0xFF4AB5
#define LEFT 0xFF10EF
#define RIGHT 0xFF5AA5
#define RESET 0xFF02FD  // Tasta 5
/* Taste speciale folosite pentru prezentare */
#define WIN 0xFF22DD    // Tasta 4
#define LOSE 0xFFC23D   // Tasta 6

void IRinput(void);     // Trateaza inputul in functie de codul primit
void resetGame(void);
void winGame(void);
void loseGame(void);
```

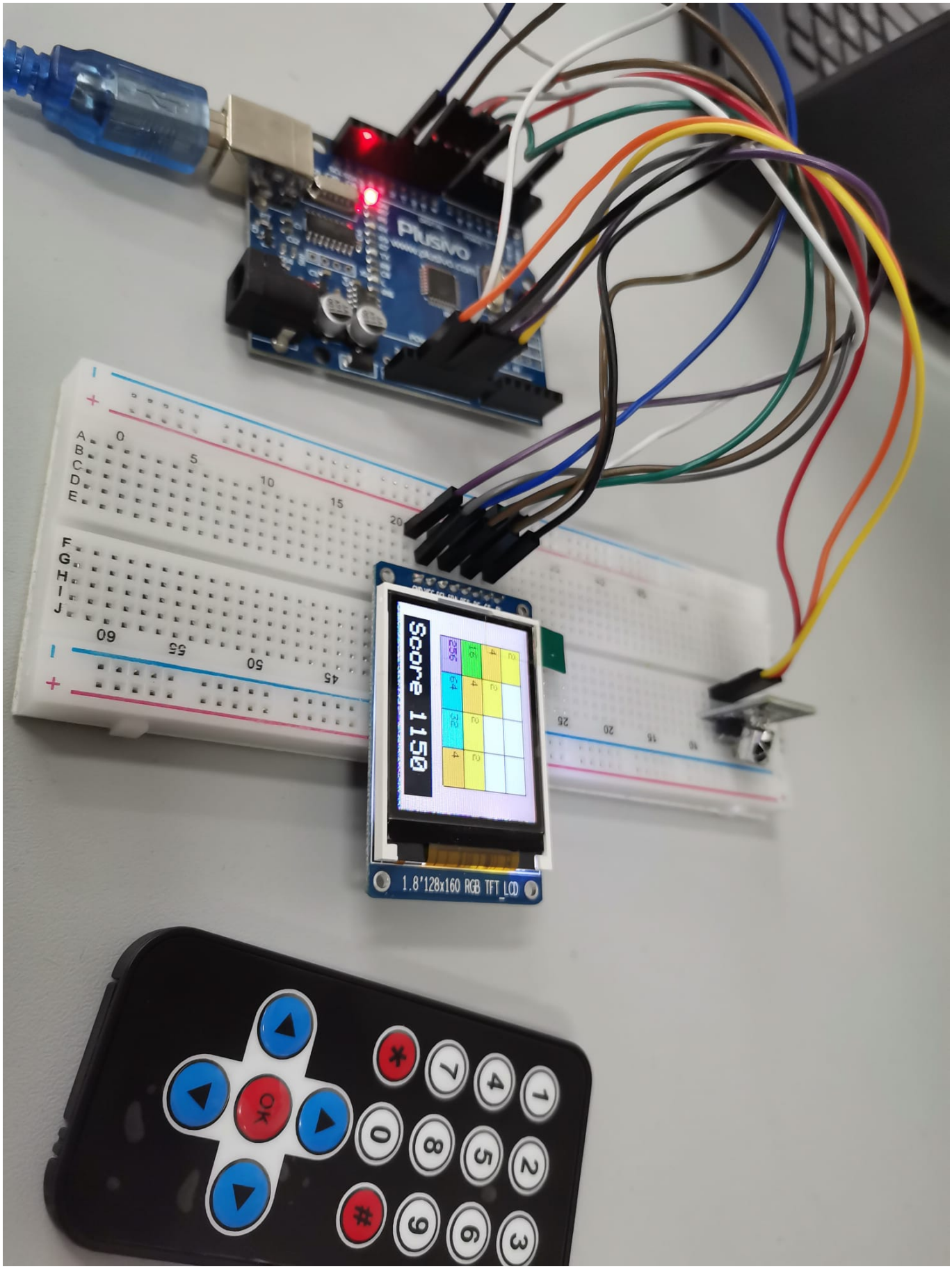
Rezultate Obţinute

Amsamblul final permite unui utilizator sa joace 2048 cu ajutorul unei telecomenzi IR. Am incercat simularea cat mai fidela a jocului clasic, prin respectarea regulilor, prin colorarea casutelor in functie de valori si prin contorizarea scorului.

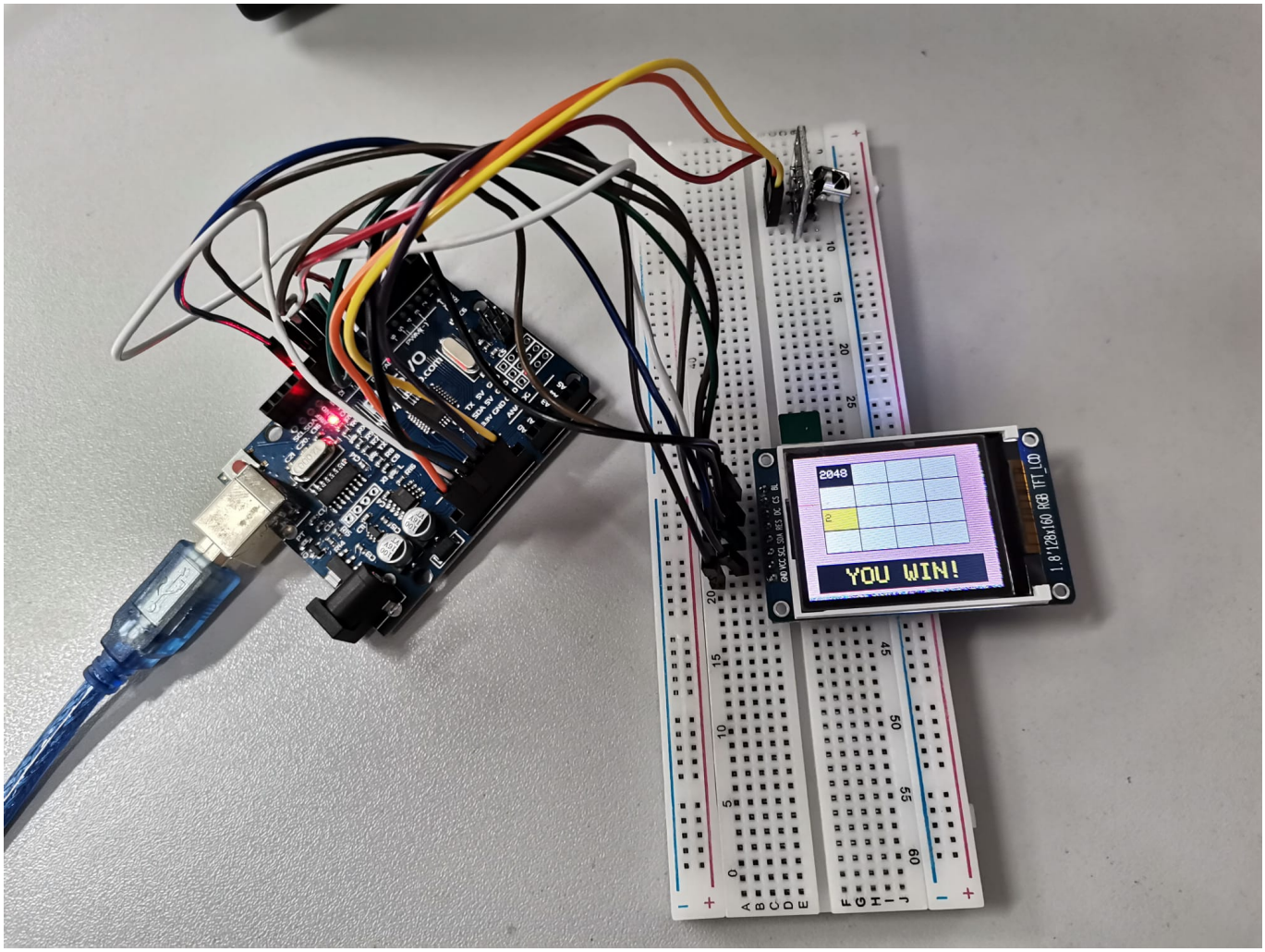
Stare initiala



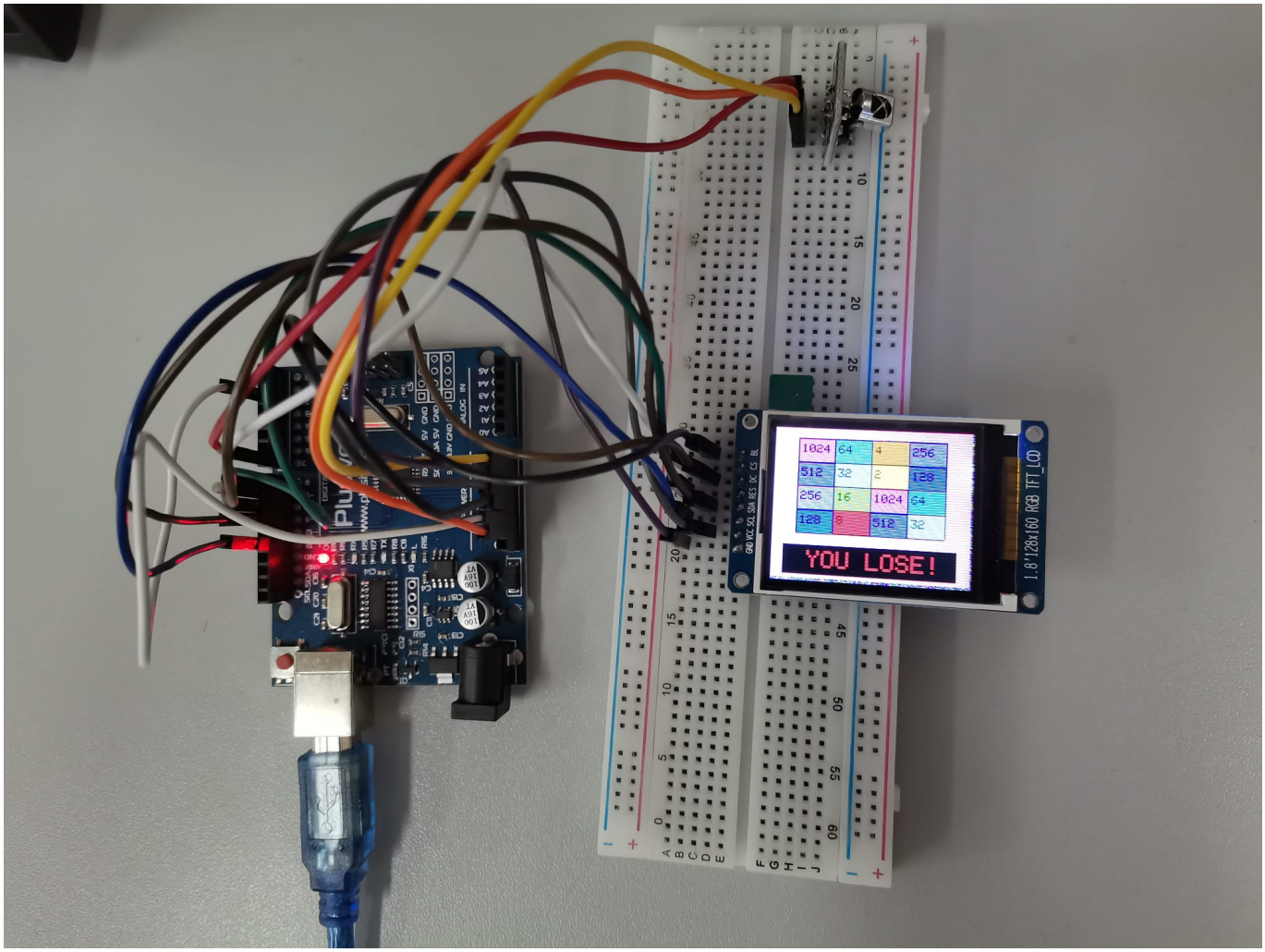
Stare intermediara



Joc castigat



Joc pierdut



Concluzii

Proiectul a fost destul de placut, a fost interesant sa lucrez cu anumite componente si cu diverslee biblioteci puse la dispozitie pentru LCD si IR remote.

Cea mai mare dificultate in realizarea proiectului a fost conectarea pieselor, nu neaparat din cauza complexitatii lor ci pentru ca piesele alese nu aveau documentatie atasata si a trebuit sa preiau informatii din mai multe locuri.

Download

Codul sursa se gaseste aici, sau se poate descarca direct arhiva

pm_2048_game.zip

Un demo cu varianta finala a proiectului se gaseste aici

Bibliografie/Resurse

2048 game
Milestone 1

TFT.h
IRremote.h

Export to PDF