

## **PROIECT 0 – INFORMATICĂ APLICATĂ**

### **TEMA 39 – Joc de spânzuratoarea cu lcd**

Titularul disciplinei:

- Mihai Antonescu
- Andrei-Alexandru Ulmămei
- Nicolae-Alexandru Guzu

Numele studenților:

- Neagu Alexandru
- Grigorescu Nicolae-Marius

Grupa: 412G

# Cuprins

## Capitole:

<b>1. Introducere .....</b>	<b>1</b>
<b>2. Resurse hardware: .....</b>	<b>2</b>
2.1 Breadboard.....	2
2.2 Placa ESP32.....	2
2.3 LCD I2C.....	2
2.4 Fire Dupont mamă – tată.....	3
2.5 LED – uri .....	3
2.6 Rezistoare 0.25 W, 10K Ohm.....	3
<b>3. Resurse software: .....</b>	<b>3</b>
3.1 Arduino IDE.....	3
3.2 Librăria externă LiquidCrystal_I2C.h .....	3
3.3 Simulatorul online WOKWI .....	4
<b>4. Implementare hardware.....</b>	<b>4</b>
<b>5. Implementare software.....</b>	<b>5</b>
<b>6. Concluzii.....</b>	<b>21</b>
<b>7. Bibliografie.....</b>	<b>22</b>

## **Introducere**

Proiectul simulează un joc de spânzurătoare, care în varianta clasică s-ar juca pe hârtie sau în alte moduri. În schimb, această variantă construită de noi, implementează jocul astfel încât să poată fi jucat de 2 persoane în același timp. Totul este afișat pe un ecran LCD, cuvintele și literele fiind introduse de către utilizatori de la tastatură. Totodată, în afară de implementarea efectivă a spânzurătoarei pe ecranul LCD, am atașat încă două LED – uri proiectului. Astfel, cel cu lumina verde se va aprinde când utilizatorul va introduce o literă corectă, iar cel cu lumină roșie în cazul contrar.

Din punct de vedere al obiectivelor propuse, putem spune că proiectul este destinat oricărei categorii de vârstă, fiind un joc interactiv, care poate îmbunătăți abilitățile atât cognitive, cât și intelectuale. De asemenea, proiectul este destinat entertainment-ului utilizatorilor. Implementarea noastră de a contoriza numărul de câștiguri sau de înfrângeri al fiecărui jucător, aduce în cadrul jocului ideea de competiție, ceea ce poate deveni, din nou, o sursă de amuzament dar și de învățare interactivă.

Tradiția acestui joc, cunoscut de o perioadă semnificativă de timp de către un public larg, este continuată, prin acest proiect, și în varianta mai modernă, prin intermediul mijloacelor hardware și software de care ne-am folosit.

## Resurse hardware

### 1. Breadboard

Un breadboard, sau protoboard, este o bază de construcție pentru prototiparea electronicelor. Inițial, cuvântul se referea la bucată de lemn lustruită folosită la felierea pâinii. În anii 1970 a devenit disponibilă placa fără sudură (plugboard, o placă cu matrice de terminale), iar în zilele noastre termenul „breadboard” este folosit în mod obișnuit pentru a se referi la acestea [1].

Deoarece placa nu necesită lipire, este reutilizabilă. Acest lucru o face ușor de utilizat pentru crearea de prototipuri temporare și experimentarea designului unor circuite. Din acest motiv, breadboard-urile sunt, de asemenea, populare în rândul studenților și în educația tehnologică.

### 2. Placa ESP32

ESP32 este o serie ieftină, de microcontroller cu WI-FI integrat și dual-mode Bluetooth. Unele dintre caracteristicile acestei plăci sunt [2]:

- CPU: Xtensa dual-core 32 -bit LX6, funcționând la 160 sau 240 MHz
- Memorie 320 KiB RAM, 448KiB ROM
- Wi-Fi: 802.11 b/g/n
- Bluetooth: v4.2 BR/EDR și BLE
- 34 de GPIOs programabili
- Etc

### 3. LCD I2C

- Acest ecran LCD 1602 cu modul IIC/I2C integrat poate fi folosit pentru afișarea informațiilor, primite de la diferiți senzori de temperatura, umiditate sau orice fel de mesaje programate.

Acest LCD este folosit de obicei pentru proiecte în care nu avem foarte mulți pini disponibili de la microcontroller, datorită adaptorului pentru interfață I2C, ce are nevoie de doar două conexiuni, SDA/SCL și conexiunea la masă.

LCD-ul are contrast ajustabil și vă poate ajuta să citiți clar ecranul într-un mediu întunecat. Culoarea caracterelor este alba, iar backlightul este albastru. [3]

- Caracteristici tehnice:

## **Facultatea de Electronică, Telecomunicații și Tehnologia Informației**

Material PCB + plastic

Tip ecran LCD

Dimensiune ecran 2.6 inch

Rezoluție 80 x 16

Tensiune de lucru 4.5 ~ 5.5V

Curent de lucru 80mA

Dimensiuni 3,15 in x 1,42 in x 0,71 in (8 cm x 3,6 cm x 1,8 cm)

Greutate 34 g

4. Fire Dupont mamă – tată (4 folosite in cadrul proiectului)
5. LED – uri(2 folosite in proiect)
6. Rezistoare 0.25 W, 10K Ohm (2 folosite in proiect)
  - Tip: CFR1 / 4
  - Valori de la 0.1 Ohm până la 22M Ohm
  - Toleranță:  $\pm 5\%$
  - Putere maximă la 70 C: 0.25 W
  - Dimensiuni: 2.5 x 6.8 mm [7]

## **Resurse software**

### **1. Arduino IDE**

Arduino IDE este un software care îți pune la dispoziție codul sursă, acesta poate fi îmbunătățit și modificat chiar de către utilizatori. Este utilizat pentru a scrie și încărca cod pe plăcile Arduino. Acest IDE este compatibil cu diferite sisteme de operare, precum: Windows, Mac OS X, și Linux. De asemenea, suportă limbaje de programare precum C and C++ .

### **2. Bibliotecă externă LiquidCrystal\_I2C.h**

Această bibliotecă permite comunicarea cu diferite LCD – uri, în special I2C, folosit în proiectul de față. Permite o placă Arduino/Genuino să controleze asemenea LCD – uri bazate pe chipsetul Hitachi HD44780. Această bibliotecă funcționează atât în 4, cât și în 8 biți. Această bibliotecă este compatibilă cu arhitectura AVR, așa că poate fi folosită pe următoarele plăci Arduino [4]:

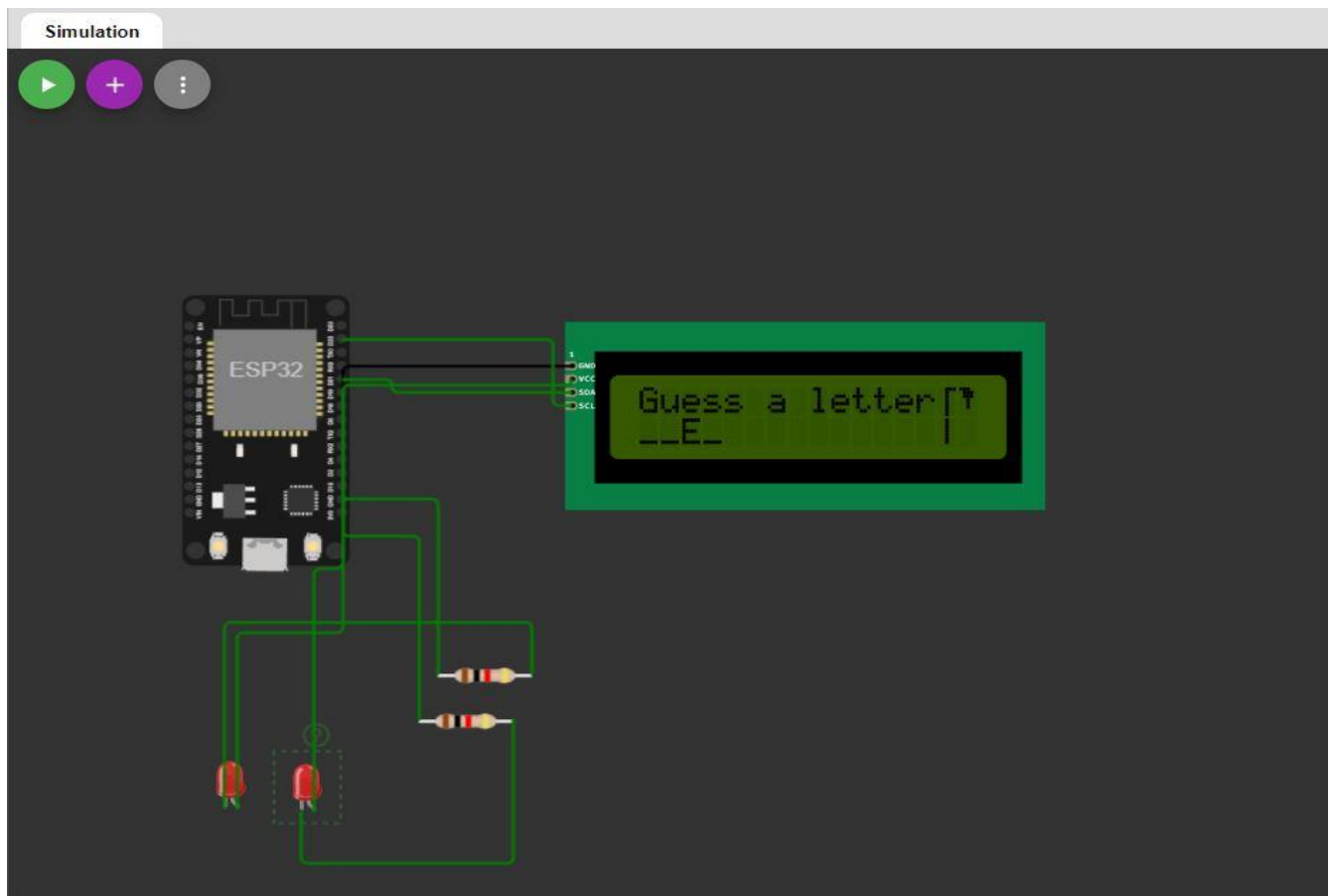
- Arduino Micro
- Arduino Leonardo
- Arduino Mega
- Arduino Nano
- Arduino Uno
- Arduino Yún

3. Simulatorul online WOKWI

Wokwi este un simulator online pentru proiecte legate de electronică. Principalul avantaj al acestui mediu de dezvoltare este siguranța că nu se poate avaria niciun dispozitiv hardware [5].

## Implementare hardware

Noi am realizat schema electronică a acestui proiect în cadrul simulatorului online wokwi.



### 4.1 - wokwi

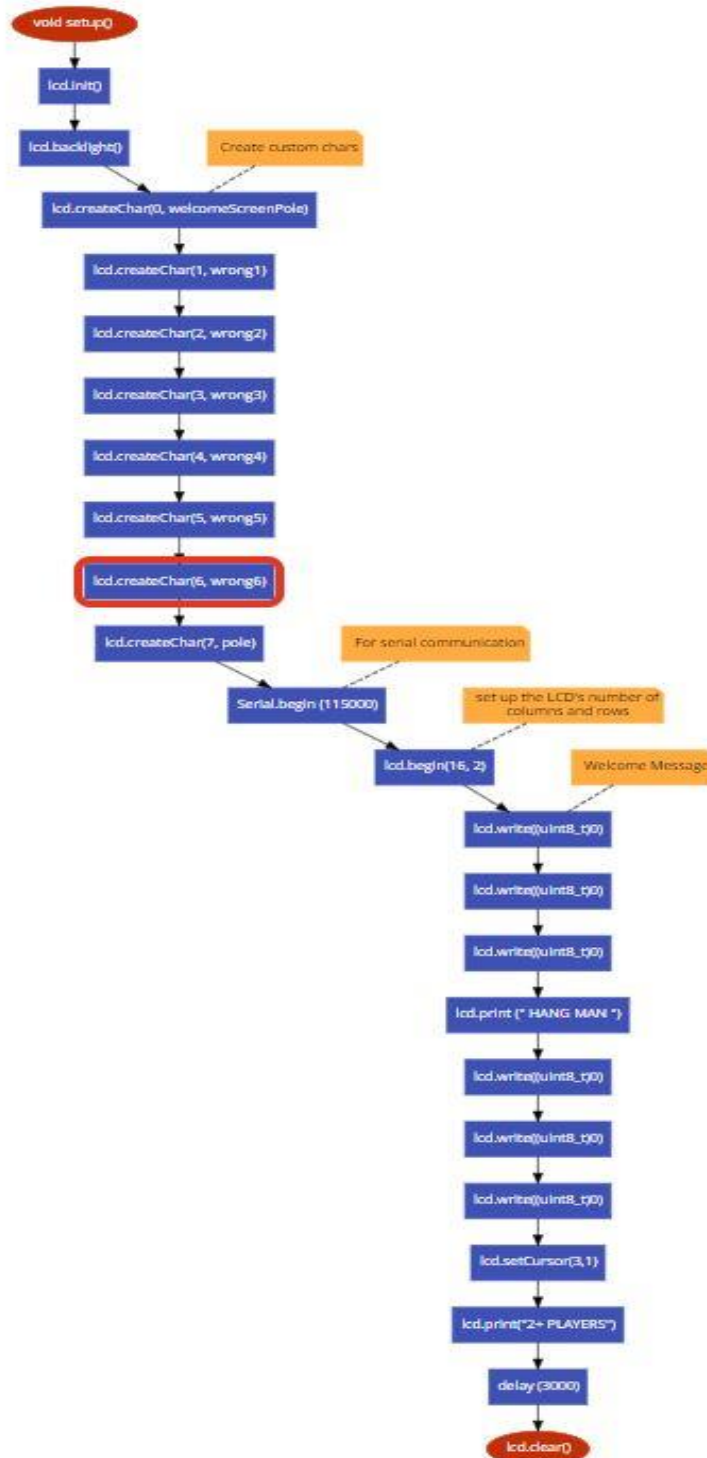
Conexiuni:

- GND (LCD) – GND(ESP32)
- VCC (LCD) – 3V3 (ESP32)
- SDA (LCD) – GPIO 21/GPIO 4
- SCL (LCD) – GPIO 22/GPIO 5

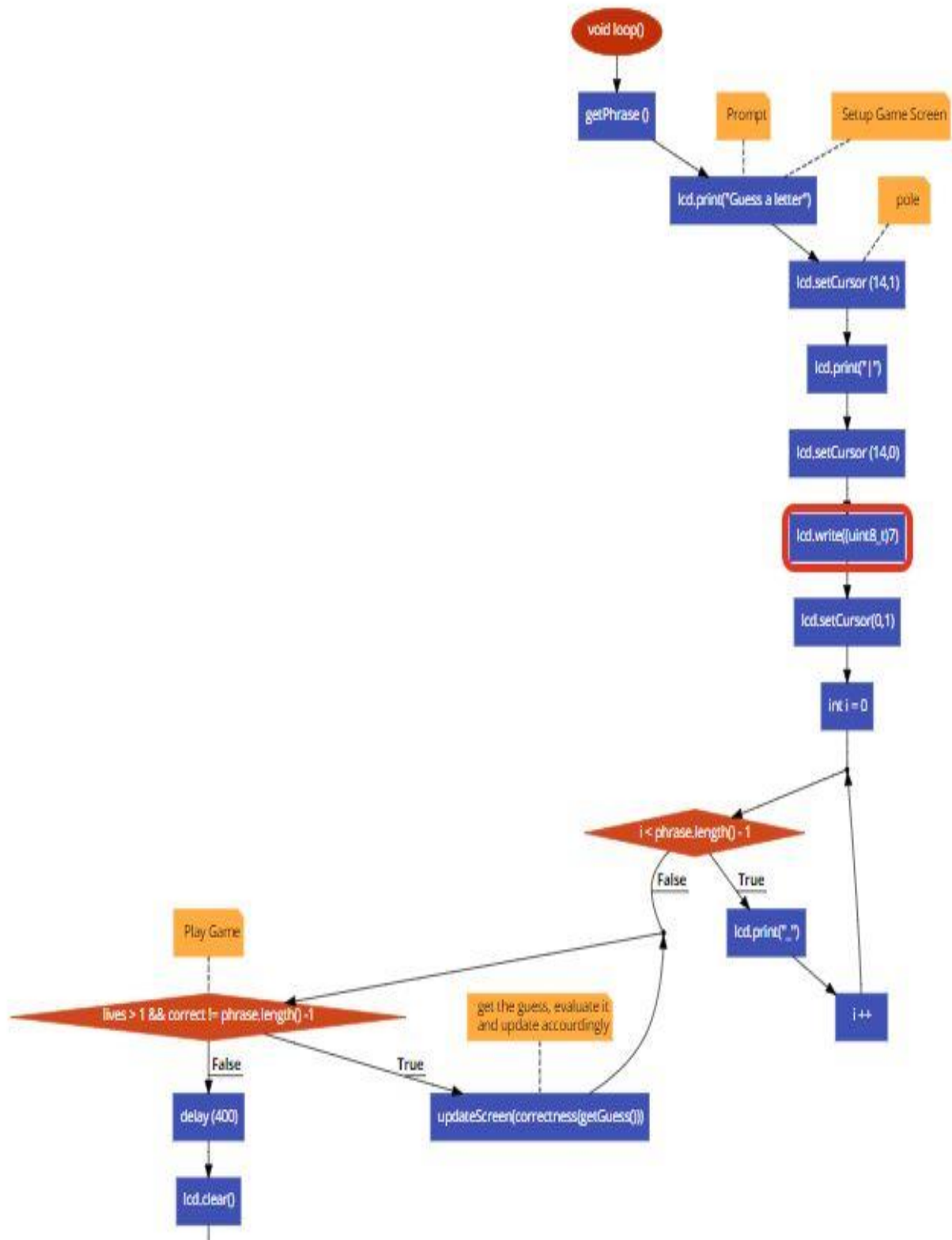
De asemenea, a fost nevoie de ajustarea potențiometrului integrat pe spatele ecranului LCD I2C, pentru a mări intensitatea caracterelor pe ecran, intensificând astfel contrastul dintre aceasta și lumina albastră de fundal [6].

## Implementare software

Organigrame pentru codul sursă:

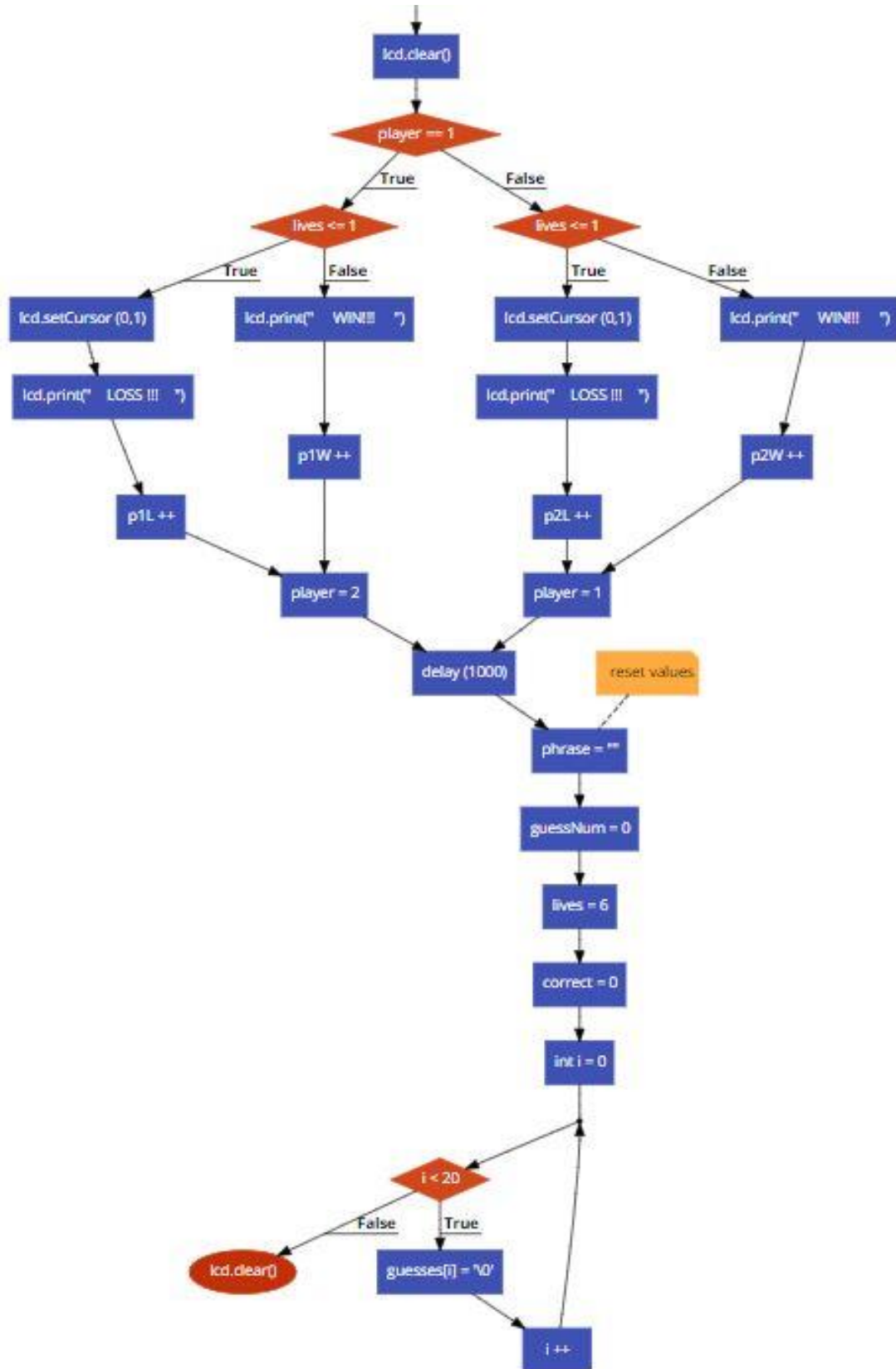


5.1 – void setup()

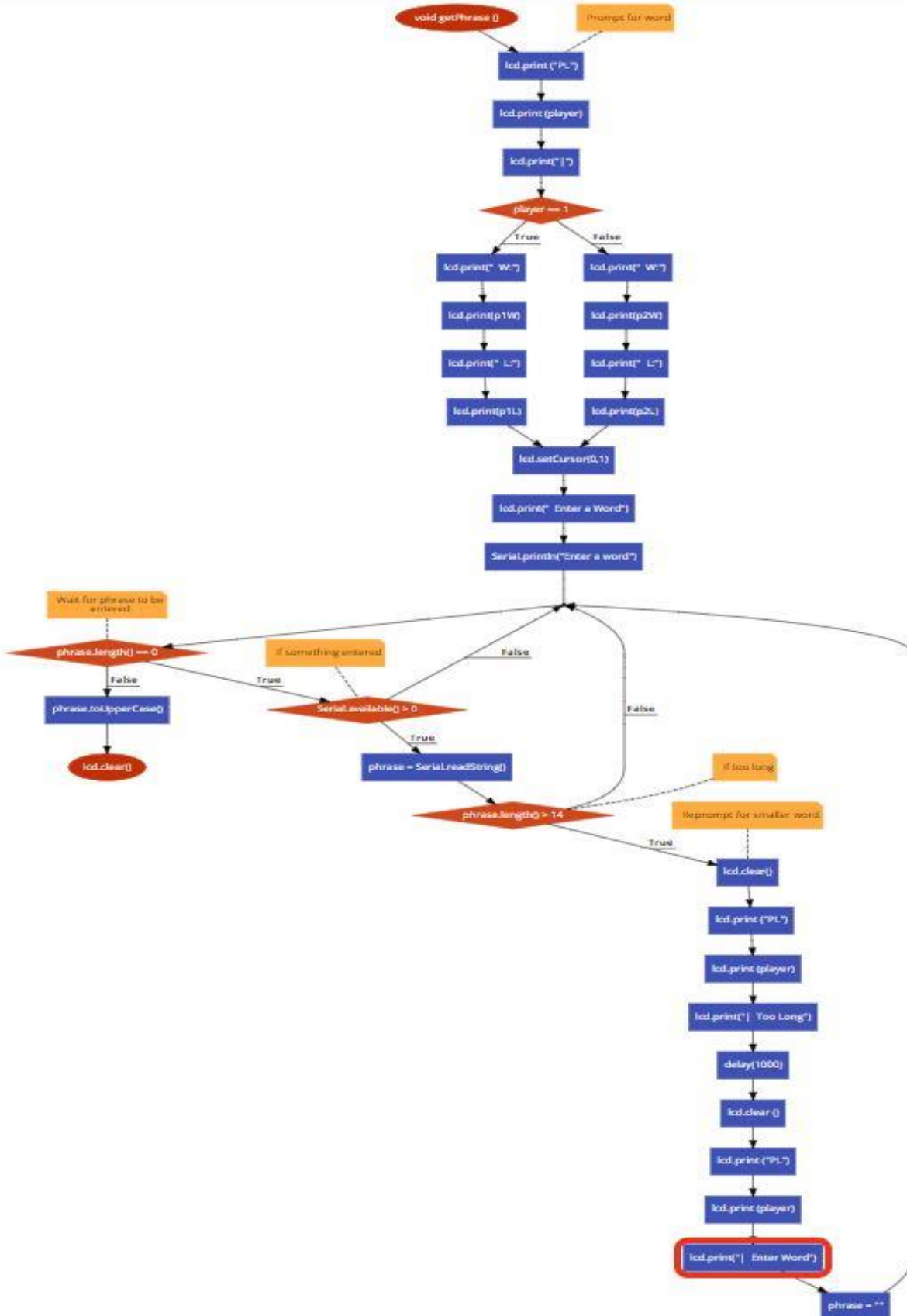




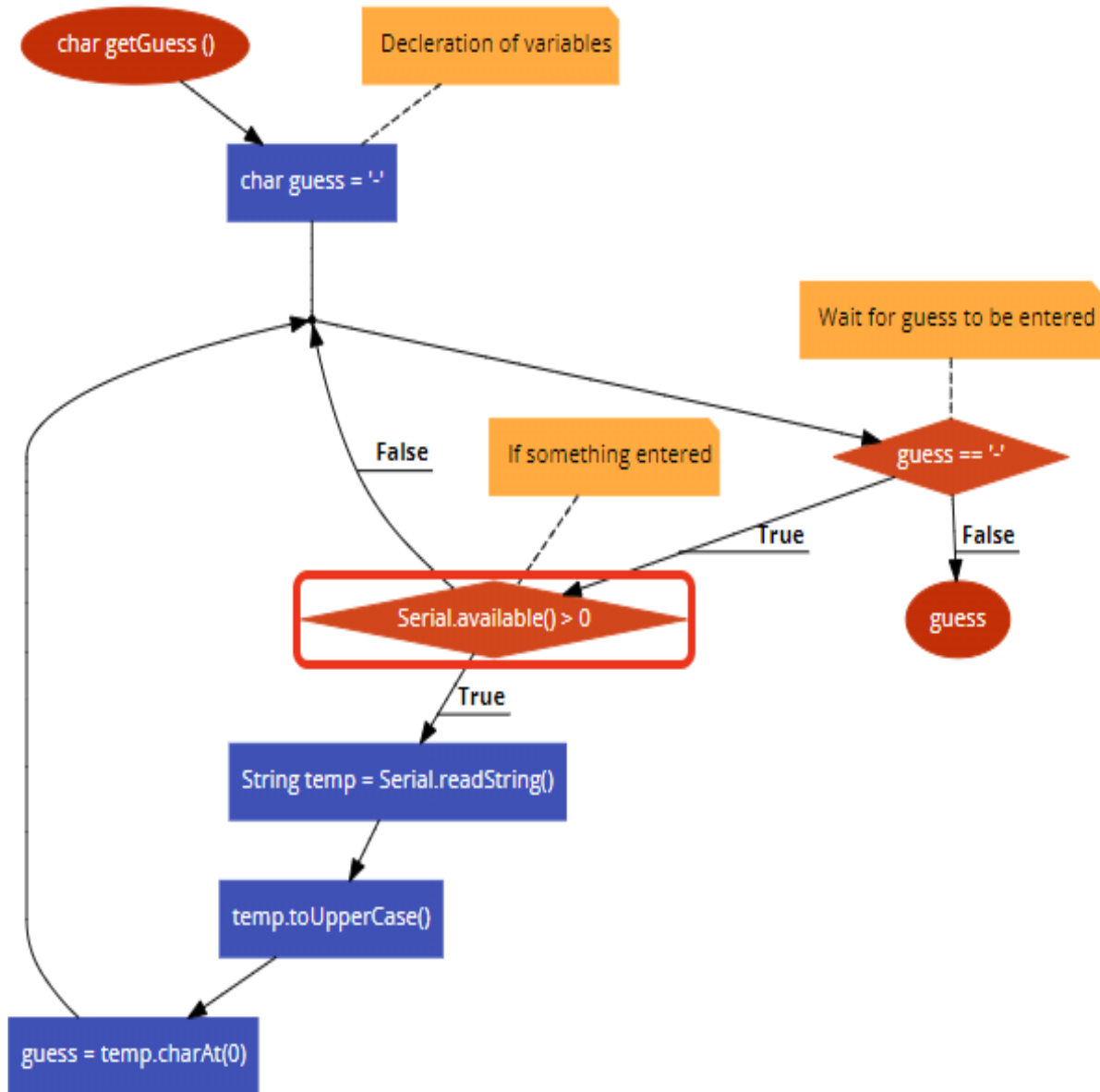
5.2 – void loop1()



5.3 – void loop2()



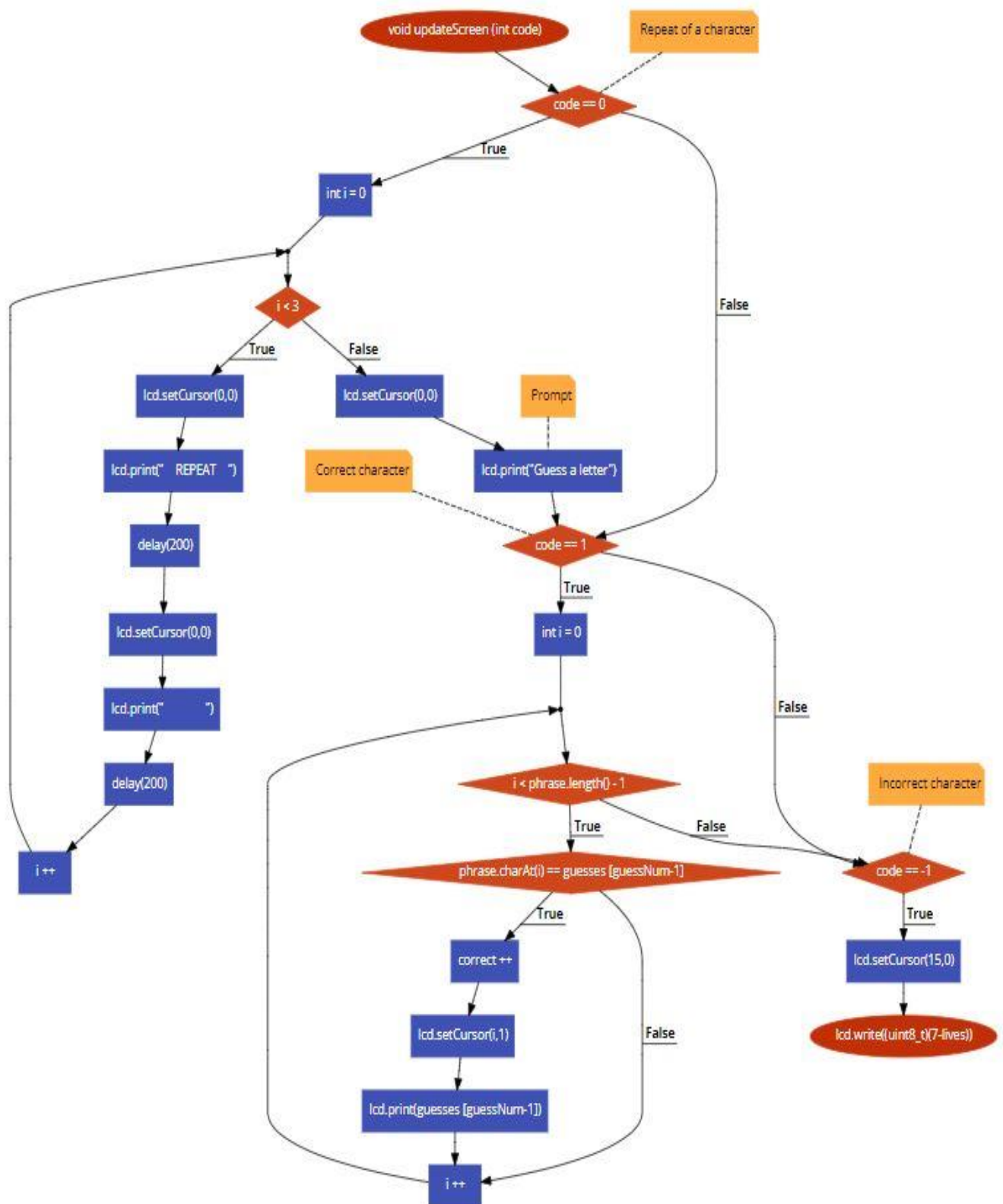
5.4 – void getPhrase()



5.5 – char getGuess()



## 5.6 – int correctness()



5.7 – void updateScreen (int code)

Codul sursă:

```
#include <LiquidCrystal_I2C.h>

int totalColumns = 16;
int totalRows = 2;

LiquidCrystal_I2C lcd(0x27, totalColumns, totalRows);

//Variabile
String phrase = "";
char guesses [20];
int guessNum = 0;
int lives = 6;
int player = 1;
int correct = 0;
int p1W = 0;
int p1L = 0;
int p2W = 0;
int p2L = 0;

// aici sunt create "imaginile" ce vor compune omul din spânzurătoare
byte welcomeScreenPole[8] = { 0b00111,
                               0b00101,
                               0b00100,
                               0b00100,
                               0b00100,
                               0b00100,
                               0b00100,
                               0b11111 };

byte wrong1[8] = { 0b11100,
                   0b01110,
                   0b01110,
                   0b00000,
                   0b00000,
                   0b00000,
                   0b00000,
                   0b00000 };

byte wrong2[8] = { 0b11100,
                   0b01110,
                   0b01110,
                   0b00100,
                   0b00100,
                   0b00100,
                   0b00100,
                   0b00000 };
```

```
        0b00000  };  
  
byte wrong3[8] = { 0b11100,  
                   0b01110,  
                   0b01110,  
                   0b00100,  
                   0b00111,  
                   0b00100,  
                   0b00000,  
                   0b00000  };  
  
byte wrong4[8] = { 0b11100,  
                   0b01110,  
                   0b01110,  
                   0b00100,  
                   0b11111,  
                   0b00100,  
                   0b00000,  
                   0b00000  };  
  
byte wrong5[8] = { 0b11100,  
                   0b01110,  
                   0b01110,  
                   0b00100,  
                   0b11111,  
                   0b00100,  
                   0b01000,  
                   0b01000  };  
  
byte wrong6[8] = { 0b11100,  
                   0b01110,  
                   0b01110,  
                   0b00100,  
                   0b11111,  
                   0b00100,  
                   0b01010,  
                   0b01010  };  
  
byte pole[8] = { 0b00111,  
                 0b00100,  
                 0b00100,  
                 0b00100,  
                 0b00100,  
                 0b00100,  
                 0b00100,  
                 0b00100  };
```

```
// ***** Setup
*****
void setup()
{

  pinMode(18,OUTPUT);
  pinMode(19, OUTPUT);
  lcd.init();
  lcd.backlight();

  //Create custom chars
  lcd.createChar(0, welcomeScreenPole);

  lcd.createChar(1, wrong1);
  lcd.createChar(2, wrong2);
  lcd.createChar(3, wrong3);
  lcd.createChar(4, wrong4);
  lcd.createChar(5, wrong5);
  lcd.createChar(6, wrong6);

  lcd.createChar(7, pole);

  //Pentru comunicarea cu seriala
  Serial.begin (115200);

  // setăm numărul de coloane și rânduri ale ecranului LCD - ului
  lcd.begin(16, 2);

  //Mesajul de început
  lcd.write((uint8_t)0);
  lcd.write((uint8_t)0);
  lcd.write((uint8_t)0);

  lcd.print (" HANG MAN ");

  lcd.write((uint8_t)0);
  lcd.write((uint8_t)0);
  lcd.write((uint8_t)0);

  lcd.setCursor(3,1);
  lcd.print ("2+ PLAYERS");
  delay (3000);
  lcd.clear();
```



```

}

// ***** Loop
*****
void loop()
{
    //funcție pentru a prelua un cuvânt de la tastatură pentru a fi ghicit
    getPhrase ();

    //începe jocul propriu - zis
    lcd.print("Guess a letter");//Prompt

    //pole
    lcd.setCursor (14,1);
    lcd.print("|");

    lcd.setCursor (14,0);
    lcd.write((uint8_t)7);

    //liniile ce înlocuiesc cuvântul la început
    lcd.setCursor(0,1);
    for (int i = 0; i < phrase.length() - 1; i ++)
    {
        lcd.print("_");
    }

    while (lives > 1 && correct != phrase.length() -1 )//Aici se joacă practic jocul
    {
        //primește încercarea user - ului, o verifică și arată pe ecran mesajul
        corespunzător

        updateScreen(correctness(getGuess()));
    }

    delay (400);
    lcd.clear();

    //se schimbă jucatorul și se înregistrează câștigul sau pierderea fiecăruia
    if (player == 1)
    {
        if (lives <= 1)
        {lcd.setCursor (0,1);
            lcd.print("    LOSS !!!    ");
            plL ++;

```

```
    }

    else
    {
        lcd.print("      WIN!!!      ");
        p1W ++;
    }

    player = 2;
}

else
{
    if (lives <= 1)
    {lcd.setCursor (0,1);
        lcd.print("      LOSS !!!      ");
        p2L ++;
    }
    else
    {
        lcd.print("      WIN!!!      ");
        p2W ++;
    }

    player = 1;
}

delay (1000);

//resetăm valorile
phrase = "";
guessNum = 0;
lives = 6;
correct = 0;

for (int i = 0; i < 20; i ++)
{
    guesses[i] = '\0';
}

lcd.clear();
}
```

```
// ***** getPhrase
*****
void getPhrase ()
{

  lcd.print ("PL");
  lcd.print (player);
  lcd.print("|");

  if (player == 1)
  {
    lcd.print("  W:");
    lcd.print(p1W);
    lcd.print("  L:");
    lcd.print(p1L);
  }
  else
  {
    lcd.print("  W:");
    lcd.print(p2W);
    lcd.print("  L:");
    lcd.print(p2L);
  }

  lcd.setCursor(0,1);
  lcd.print("  Enter a Word");
  Serial.println("Enter a word");

  while (phrase.length() == 0)//se așteaptă ca ceva să fie introdus
  {
    if (Serial.available() > 0)//dacă a fost ceva introdus
    {
      phrase = Serial.readString();

      if (phrase.length() > 14)// dacă e prea lung cuvântul
      {
        lcd.clear();
        lcd.print ("PL");
        lcd.print (player);
        lcd.print("|  Too Long");

        delay(1000);
        lcd.clear ();

        lcd.print ("PL");
        lcd.print (player);
        lcd.print("|  Enter Word");
      }
    }
  }
}
```

```

        phrase = "";
    }
}

phrase.toUpperCase();
lcd.clear();
}

// ***** getGuess
*****
char getGuess ()
{
    //declarăm variabil
    char guess = '-';

    while (guess == '-')//se așteaptă ca o literă sa fie introdusă
    {
        if (Serial.available() > 0)//dacă s-a introdus ceva
        {
            String temp = Serial.readString();
            temp.toUpperCase();
            guess = temp.charAt(0);
        }
    }

    return guess;
}

// ***** correctness
*****
int correctness (char guess)
{
    //verifică daca cuvântul conține caractere(char)
    for (int i = 0; i < phrase.length(); i++)
    {
        if (guess == guesses[i])//If previously entered
        {
            return 0;
        }

        if (phrase.charAt(i) == guess)//dacă este litera corectă
        { int j = 0;
          for(j=0; j<=2;j++){

```

```

        digitalWrite(18, HIGH);
        delay(200);
        digitalWrite(18, LOW);
        delay(200);
        digitalWrite(18, HIGH);
        delay(200);
        digitalWrite(18, LOW);
    }

    guessNum++;
    guesses [guessNum-1] = guess;
    return 1;
}
}

//dacă nu este litera corectă
int k = 0;
for(k=0;k<=2;k++){
    digitalWrite(19, HIGH);
    delay(200);
    digitalWrite(19, LOW);
    delay(200);
    digitalWrite(19, HIGH);
    delay(200);
    digitalWrite(19, LOW);
}

    lives--;
    guessNum++;
    guesses [guessNum-1] = guess;

    return -1;
}

// ***** updateScreen
*****
void updateScreen (int code)
{
    if (code == 0)//dacă se repetă litera introdusă
    {
        for (int i = 0; i < 3; i++)
        {
            lcd.setCursor(0,0);
            lcd.print("    REPEAT    ");

```

```
    delay(200);
    lcd.setCursor(0,0);
    lcd.print("          ");
    delay(200);
}

lcd.setCursor(0,0);
lcd.print("Guess a letter");
}

if (code == 1)//litera corectă
{
    for (int i = 0; i < phrase.length() - 1; i++)
    {
        if (phrase.charAt(i) == guesses [guessNum-1])
        {
            correct++;
            lcd.setCursor(i,1);
            lcd.print(guesses [guessNum-1]);
        }
    }
}

if (code == -1)//litera incorectă
{
    lcd.setCursor(15,0);
    lcd.write((uint8_t) (7-lives));
}
}
```

## Concluzii

În concluzie, partea de implementare a necesitat un efort destul de mic din punct de vedere hardware, în comparație cu partea software, unde programul a necesitat o atenție sporită datorată condițiilor necesare pentru funcționarea corectă a unui astfel de joc.

Probleme majore nu am întâmpinat pe parcursul dezvoltării acestui proiect, exceptând găsirea fiecărei componente. Acest lucru putem spune că a fost o mică piedică. Ușurința folosirii mediului Arduino, cât și simulatorului online Wokwi, a fost un avantaj major în compunerea proiectului.

Din punctul de vedere al posibilelor îmbunătățiri, ca în orice alt proiect, este întotdeauna loc de mai bine. Spre exemplu, un display LED cu 7 segmente sau chiar mai multe ar fi fost de ajutor pentru a contoriza numărul răspunsurilor greșite de la tastatură.

Un alt exemplu ar putea fi utilizarea mai multor LED – uri, care să fie dispuse în forma implementată și pe ecranul I2C. Astfel, la fiecare greșeală, un LED se va aprinde, rămânând aprins până la finalizarea ghicirii sau nu a cuvântului.

Așadar, noi suntem mulțumiți de alegerea făcută în privința temei acestui proiect, deoarece ne-a învățat diferite lucruri noi despre o mică parte a electronicii, stârnindu-ne curiozitatea de a afla mai mult, dar și mintea pentru a găsi diferite soluții pentru problemele întâmpinate.

## Bibliografie

- [1] <https://en.wikipedia.org/wiki/Breadboard>
- [2] <https://en.wikipedia.org/wiki/ESP32#ESP32-C6>
- [3] <https://cleste.ro/ecran-lcd-1602-iic-i2c.html>
- [4] <https://www.arduino.cc/reference/en/libraries/liquidcrystal/>
- [5] <https://docs.wokwi.com/>
- [6] <https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/>
- [7] <https://www.optimusdigital.ro/ro/componente-electronice-rezistoare/1088-rezistor-025w-100k.html>



