

# НП "Обучение за ИТ кариера" Курсов проект - “Тетрис”

Изготвил: Реджеб Реджеб

Група 08

Гр.Хасково - 2024г.

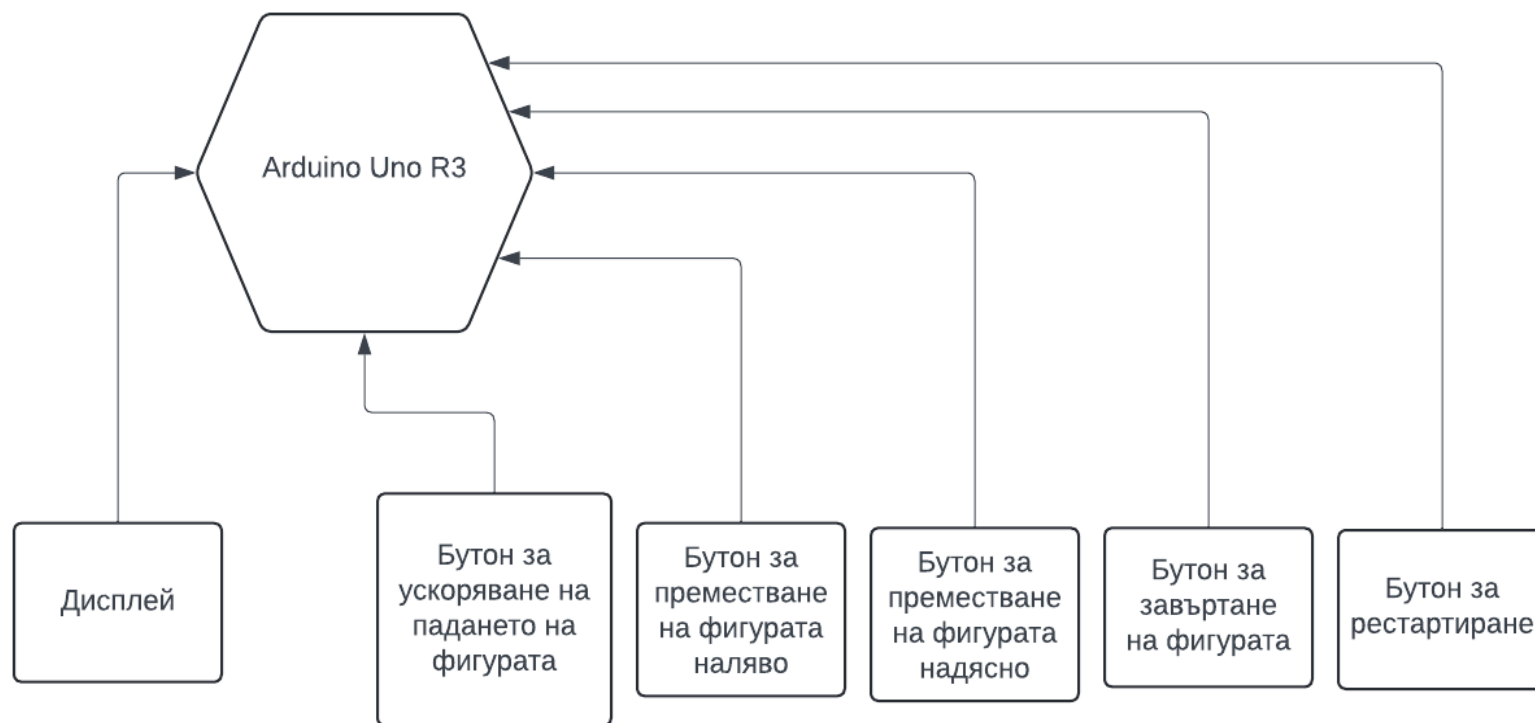
GitHub:<https://github.com/RedzhebRedzheb/tetrisItKariera.git>

Tinkercad: <https://www.tinkercad.com/things/6TPcSjbleHz-dazzling-turing>

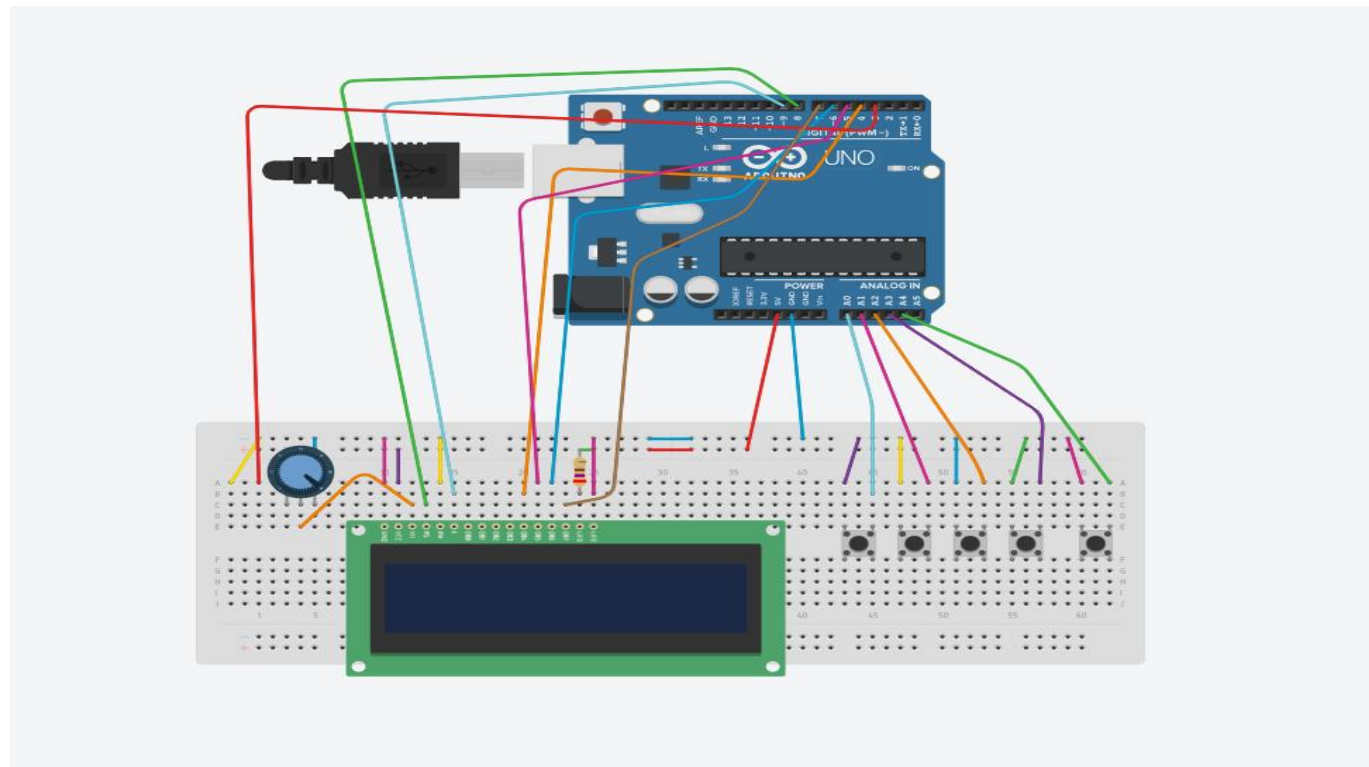
# Описание на проекта:

- Проектът представлява игра на тетрис, реализирана за микроконтролер с използване на Arduino платформата. Играта използва LCD дисплей за визуализация и бутони за управление на движението и ротацията на падащите фигури. Допълнително, проектът включва възможност за запазване на най-висок резултат чрез използване на EEPROM памет.

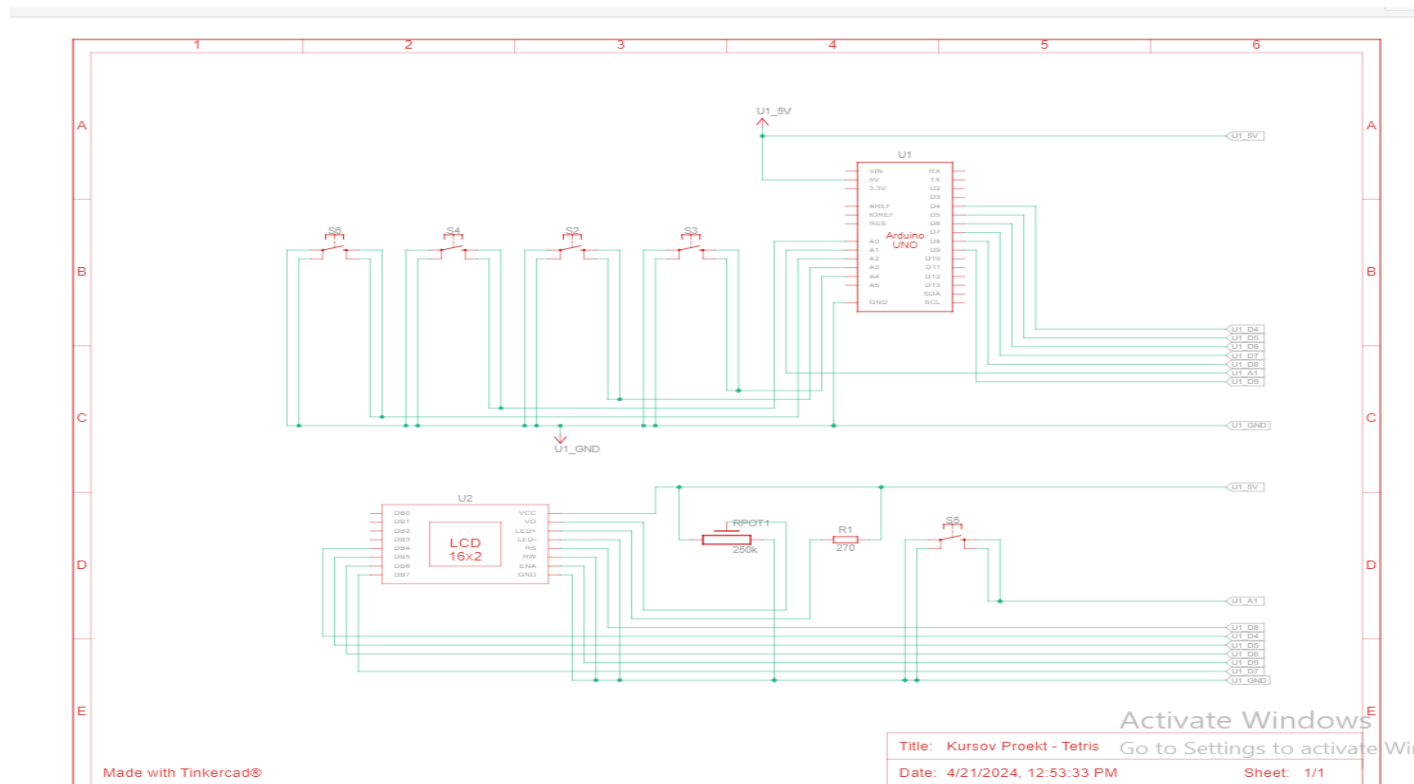
# Блокова схема:



# Блокова схема:



# Електрическа схема:



# Списък със съставните части:

- Микроконтролер Arduino Uno
- Breadboard
- LCD дисплей 16x2
- 5 бутона
- Резистор 270  $\Omega$
- Потенциометър

# Сорс код:

- `setup()`: Настройва пиновете за входове, инициализира LCD дисплея, създава символи за LCD, сетва начални стойности и показва началния екран с най-високия резултат.
- `initialize()`: Изчиства игралното поле и подготвя нова игра, като задава началните стойности за скоростта на игра, резултата и времето за следващото движение на блока.

```
void setup() {  
  pinMode(A0, INPUT);  
  digitalWrite(A0, INPUT_PULLUP);  
  pinMode(A1, INPUT);  
  digitalWrite(A1, INPUT_PULLUP);  
  pinMode(A2, INPUT);  
  digitalWrite(A2, INPUT_PULLUP);  
  pinMode(A3, INPUT);  
  digitalWrite(A3, INPUT_PULLUP);  
  pinMode(A4, INPUT);  
  digitalWrite(A4, INPUT_PULLUP);  
  lcd.begin(16, 2);  
  lcd.createChar(0, ctop);  
  lcd.createChar(1, cbottom);  
  lcd.createChar(2, cfull);  
  Serial.begin(9600);  
  randomSeed(A1);  
  initialize();  
  lcd.clear();  
  lcd.setCursor(1, 0);  
  lcd.clear();  
  lcd.setCursor(0, 0);  
  lcd.print("Highscore:");  
  lcd.setCursor(0, 1);  
  lcd.print(highScore);  
  delay(2000);  
}
```

```
void initialize() {  
  clearDisplay();  
  newShape();  
  gameSpeed = 600;  
  score = 0;  
  timeToMove = millis() + gameSpeed;  
  highScore = getHighScore();  
}
```

# Сорс код:

- loop(): Основният цикъл на играта, който проверява действията на потребителя чрез бутоните и обработва логиката за движение на блоковете, проверка за пълни редове и актуализация на резултата.
- getKey(): Чете статуса на бутоните и връща съответния код на натиснат бутон.

```
void loop() {
    bool gameOver = false;

    newShape();
    drawShape();
    drawScreen();

    while (!gameOver && moveDown()) {
        while (millis() < timeToMove) {
            int k = getKey();
            if (k != prevKey) {
                Serial.println(analogRead(A0));
                switch (k) {
                    case btn1:
                        while (moveDown());
                        break;
                    case btn2:
                        moveLeft();
                        break;
                    case btn3:
                        moveRight();
                        break;
                    case btn4:
                        rotate();
                        break;
                    case btn5:
                        if (currentY == 0) {
                            restart();
                            return;
                        }
                        break;
                }
                prevKey = k;
            }
        }
        timeToMove = millis() + gameSpeed;
    }
}
```

```
int getKey() {
    if (digitalRead(A0) == 0) return btn1;
    else if (digitalRead(A1) == 0) return btn2;
    else if (digitalRead(A2) == 0) return btn3;
    else if (digitalRead(A3) == 0) return btn4;
    else if (digitalRead(A4) == 0) return btn5;
    else return btnNone;
    delay(8);
}
```



# Сорс код:

- Функции за движение: moveDown(), moveLeft(), moveRight(), rotate(), които манипулират положението на текущата фигура въз основа на входа от бутоните.

```
boolean moveDown() {
    clearShape();
    if (isValid(currentX, currentY + 1)) {
        currentY++;
        drawShape();
        drawScreen();
        return true;
    }
    drawShape();
    return false;
}
```

```
boolean moveRight() {
    clearShape();
    if (isValid(currentX + 1, currentY)) {
        currentX++;
        drawShape();
        drawScreen();
        return true;
    }
    drawShape();
    return false;
}
```

```
boolean moveLeft() {
    clearShape();
    if (isValid(currentX - 1, currentY)) {
        currentX--;
        drawShape();
        drawScreen();
        return true;
    }
    drawShape();
    return false;
}
```

```
boolean rotate() {
    int r;
    clearShape();
    if (currentRot == maxRotations - 1) r = 0;
    else r = currentRot + 1;
    if (isValid(r, currentX, currentY)) {
        currentRot = r;
        drawShape();
        drawScreen();
        return true;
    }
    drawShape();
    return false;
}
```

# Сорс код:

- drawScreen(): Обновява дисплея, като показва текущото състояние на игралното поле.
- drawShape(), clearShape(): Показват или премахват текущата фигура от игралното поле.

```
void drawScreen() {  
    for (int i = 0; i < 16; i++) {  
        lcd.setCursor(i, 0);  
        if (matrix[i][0] && matrix[i][1]) lcd.write(byte(2));  
        else if (matrix[i][0]) lcd.write(byte(0));  
        else if (matrix[i][1]) lcd.write(byte(1));  
        else lcd.print(' ');  
        lcd.setCursor(i, 1);  
        if (matrix[i][2] && matrix[i][3]) lcd.write(byte(2));  
        else if (matrix[i][2]) lcd.write(byte(0));  
        else if (matrix[i][3]) lcd.write(byte(1));  
        else lcd.print(' ');  
    }  
}
```

```
void drawShape() {  
    putShape(true, currentShape, currentRot, currentX, currentY);  
}
```

```
void clearShape() {  
    putShape(false, currentShape, currentRot, currentX, currentY);  
}
```

# Сорс код:

- `clearLines()`: Проверява за пълни редове, изтрива ги и увеличава резултата.
- `getHighScore()`, `saveHighScore()`: Чете и записва най-високия резултат в EEPROM.

```
int getHighScore() {
    long two = EEPROM.read(0);
    long one = EEPROM.read(1);
    return ((two << 0) & 0xFF) + ((one << 8) & 0xFFFF);
}

void saveHighScore() {
    EEPROM.write(0, (score & 0xFF));
    EEPROM.write(1, ((score >> 8) & 0xFF));
    highScore = score;
}
```

# Сорс код:

- restart(): Рестартира играта, като изчиства дисплея и повторно инициализира стойностите.
- flashLine(): Визуализация на ефект при изчистване на ред.

```
void restart() {  
    lcd.clear();  
    initialize();  
    lcd.setCursor(0, 0);  
    lcd.print("Restarting...");  
    delay(1000);  
    setup();  
    lcd.clear();  
    drawScreen();  
}
```

```
void flashLine(int y) {  
    for (int t = 0; t < 3; t++) {  
        for (int i = 0; i < 4; i++) matrix[y][i] = 1;  
        drawScreen();  
        for (int i = 0; i < 4; i++) matrix[y][i] = 0;  
        drawScreen();  
    }  
}
```

# Заклучение

- Този проект е пример за приложение, което съчетава хардуерни и софтуерни компоненти за създаването на забавна игра. Играчите развиват своите стратегически и координационни умения, докато се състезават за подобряване на своите рекорди в класическата игра Тетрис.

# Благодаря за вниманието

Използвана литература: Arduino.CC; Reddit; SoftUni; Tinkercad, GitHub