**Национална програма „Обучение за ИТ обучение и кариера“ на Министерство на образованието и науката**

**Модул 8**

**2023/2024**

**Проект Simon Says**

**Екип:**

Данчо Ковачев

Стефан Стефанов

Диана Шопова

Съдържание

[I. Описание на проекта 2](#_Toc133838436)

[II. Електрическа схема 3](#_Toc133838437)

[III. Съставни части 3](#_Toc133838438)

[IV. Блокова схема 4](#_Toc133838439)

[V. Сорс код и функционалност 5](#_Toc133838440)

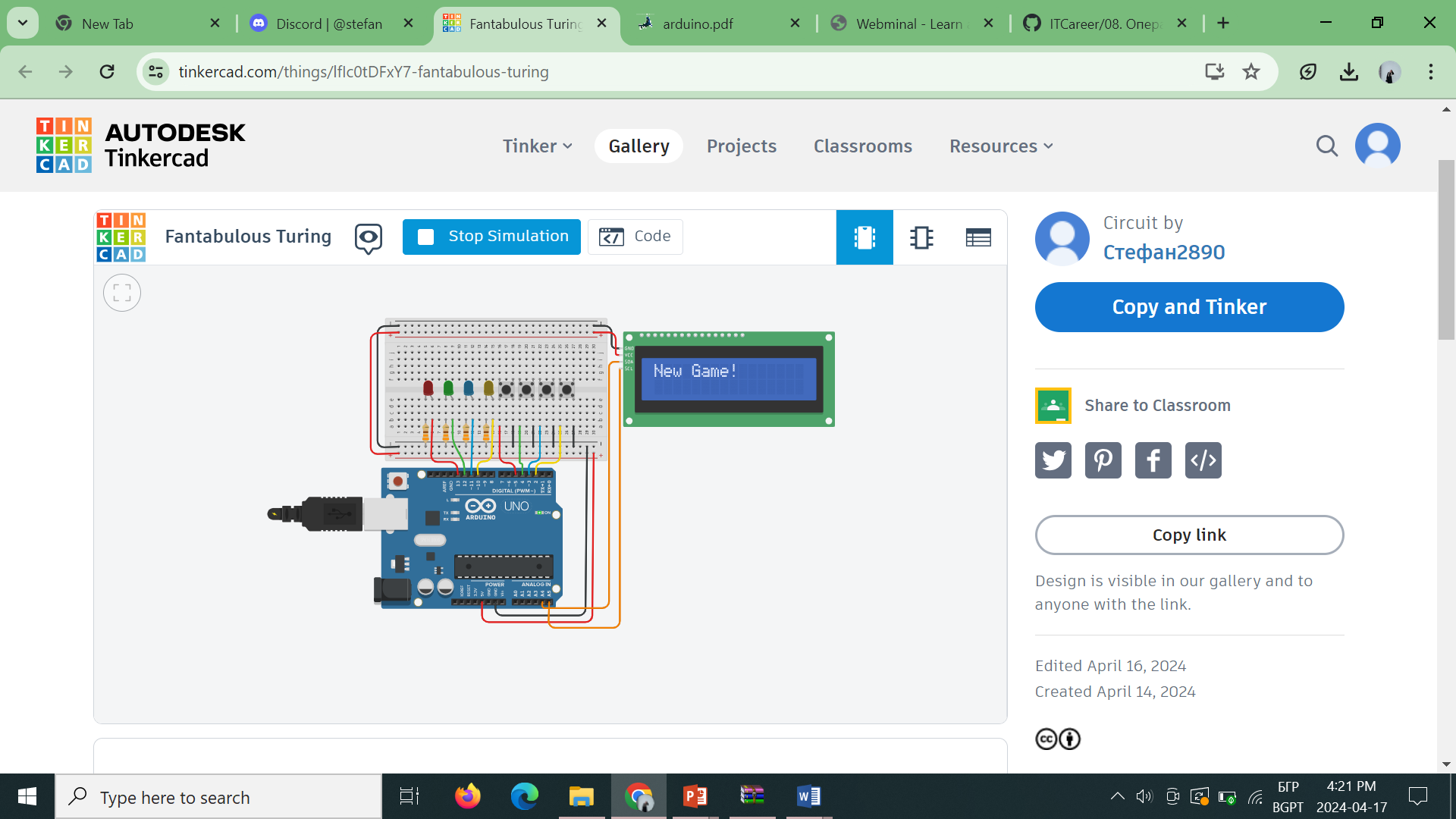
# Описание на проекта

Проектът Simon says е игра за краткосрочната памет. Създава поредица от светлини, посредством LED светодиоди, и се изисква потребителят да ги повтори в същата последователност, кликвайки върху съответстващите бутони.

Ако потребителят успее, поредицата става постепенно по-дълга и по-сложна. След като потребителят се провали или изтече лимита (10), играта свършва. Simon says разполага с монитор, който сигнализира за начало, загуба и победа, както и за кликнат бутон.



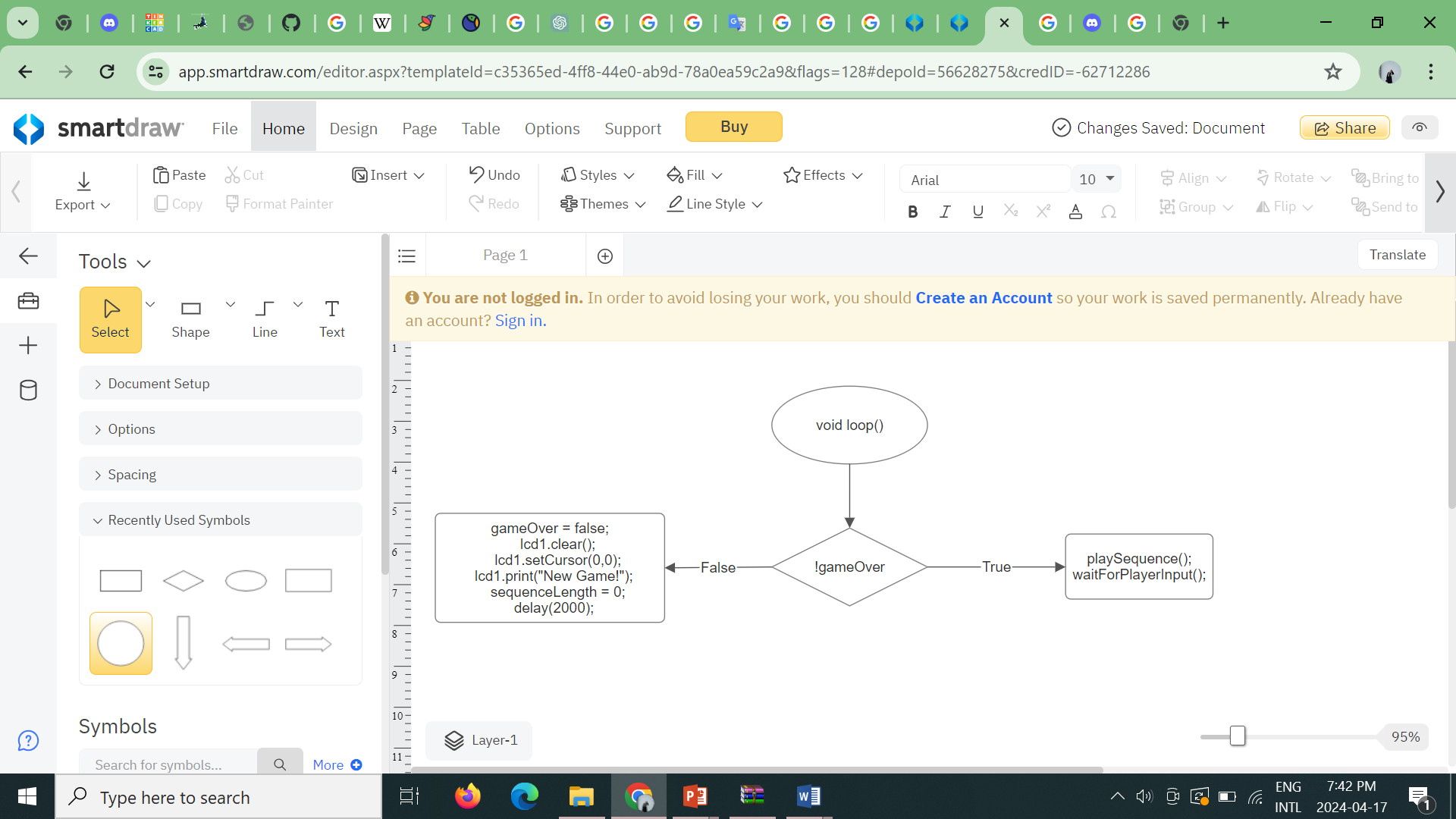
# Електрическа схема



# Съставни части

* Arduino UNO R3
* Red LED
* Yellow LED
* Green LED
* Blue LED
* Pushbutton \* 4
* 330 Ω Resistor \* 4
* PCF8574-based, 32 (0x20) LCD 16 x 2 (I2C)

# Блокова схема



# Сорс код и функционалност

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd1 (32,16,2);

Начална настройка:

Импортира библиотеката за LCD дисплеи. Създава дисплей с размери 32 – 16 колони и 2 реда.

Създаване на пиновете за управление на светодиодите. (червен, зелен, син, жълт)

Създаване на пиновете за управление на бутоните. (червен, зелен, син, жълт)

Създаване на променливите за дължината на поредицата, статуса на играта и резултата на потребителя.

const int redLED = 13;

const int greenLED = 12;

const int blueLED = 11;

const int yellowLED = 10;

const int buttonPinRED = 5;

const int buttonPinGREEN = 4;

const int buttonPinBLUE = 3;

const int buttonPinYELLOW = 2;

int sequence[10];

int sequenceLength = 0;

bool gameOver = false;

int score = 1;

void setup()

{

Инициализация на серийната комуникация.

Инициализация на LCD дисплея (lcd1). init() инициализира дисплея, backlight() включва задното осветление, а display() включва самият дисплей.

Задава позицията на курсора на (0,0) и изписва съобщение за начало на нова игра с изчакване 2 секунди.

Serial.begin(9600);

lcd1.init();

lcd1.backlight();

lcd1.display();

lcd1.setCursor(0,0);

lcd1.print("New Game!");

delay(2000);

pinMode(redLED, OUTPUT);

Конфигурация а пиновете, управляващи светодиодите като изходни и тези, управляващи бутоните като изходни с PULL-UP метода.

Случаен сийдинг, с аналогово четене, генериращ различна поредица при всяко ново стартиране на играта.

Извиква функцията nextSequence(), която генерира началната поредица от цветове за играта.

pinMode(greenLED, OUTPUT);

pinMode(blueLED, OUTPUT);

pinMode(yellowLED, OUTPUT);

pinMode(buttonPinRED, INPUT\_PULLUP);

pinMode(buttonPinGREEN, INPUT\_PULLUP);

pinMode(buttonPinBLUE, INPUT\_PULLUP);

pinMode(buttonPinYELLOW, INPUT\_PULLUP);

randomSeed(analogRead(0));

nextSequence();

}

void loop()

{

if (!gameOver)

Основен цикъл на играта:

Ако играта не е приключила, възпроизвежда поредицата от светлини с метода playSequence() и изчаква за потребителския input.

В противен случай, изчиства LCD дисплея и изписва (0,0) „New game“ за продължителност от 2 секунди.

{

playSequence();

waitForPlayerInput();

}

else

{

gameOver = false;

lcd1.clear();

lcd1.setCursor(0,0);

lcd1.print("New Game!");

sequenceLength = 0;

delay(2000);

}

}

void nextSequence()

{

Функцията nextSequence() генерира случайна последователност от цветове, като използва случайни числа за индексиране на светодиодите (червен, зелен, син, жълт).

Използва if/else statement за проверка дали дължината на поредицата е по-малка от 10 и увеличаването и/използването на метода displayGameWon().

if (sequenceLength < 10)

{

sequenceLength++;

}

else

{

displayGameWon();

}

for (int i = 0; i < sequenceLength; i++)

{

sequence[i] = random(10, 14);

}

}

void playSequence()

{

Функцията playSequence() използва for цикъл и оператора switch-case. Включва светодиодите според генерираната поредица за определеното време от 0,5 секунди., което се пренася с функцията delay(). След визуализирането на цялата поредица следва delay от 0,5 секунди.

for (int i = 0; i < sequenceLength; i++)

{

switch (sequence[i]) {

case 13:

digitalWrite(redLED, HIGH);

delay(500);

digitalWrite(redLED, LOW);

break;

case 12:

digitalWrite(greenLED, HIGH);

delay(500);

digitalWrite(greenLED, LOW);

break;

case 11:

digitalWrite(blueLED, HIGH);

delay(500);

digitalWrite(blueLED, LOW);

break;

case 10:

digitalWrite(yellowLED, HIGH);

delay(500);

digitalWrite(yellowLED, LOW);

break;

}

delay(500);

}

}

void waitForPlayerInput()

LCD дисплея се изчиства и на него се изписва съобщение, което инструктира потребителя да въведе поредицата.

Функцията използва for цикъл, за да обходи всички елементи от генерираната поредица

За всяко въведено число, функцията проверява дали бутонът, натиснат от играча, съответства на очаквания цвят в поредицата.

След като играчът успешно въведе цветовете в правилната последователност, функцията увеличава резултата на играча и продължава играта към следващото ниво.

{

lcd1.clear();

lcd1.print("Enter: ");

lcd1.print(score);

delay(2000);

lcd1.setCursor(0,1);

for(int i = 0; i<sequenceLength; i++)

{

bool state = true;

while(state)

{

int buttonPressed = buttonCheck();

if (buttonPressed > 9)

{

state =false;

int expectedLED = sequence[i];

lcd1.print("\*");

if (buttonPressed != expectedLED)

{

displayGameOver();

break;

}

}

}

delay(2000);

}

score++;

nextSequence();

}

int buttonCheck()

Проверява състоянието на бутоните, свързани към платката. Ако бутонът е натиснат връща стойността, която бутонът представлява (номерът на отговарящия светодиод).

Ако нито един бутон не е натискат, връща стойност 0, която може да се използва за обработка на случай, когато няма натиснати бутони.

{

if(digitalRead(buttonPinRED) == LOW)

{return 13;}

else if(digitalRead(buttonPinGREEN) == LOW)

{return 12;}

else if(digitalRead(buttonPinBLUE) == LOW)

{return 11;}

else if(digitalRead(buttonPinYELLOW) == LOW)

{return 10;}

else

{return 0; }

}

void displayGameWon()

{

score = 1;

Функцията displayGameWon() изчиства LCD дисплея и изписва "Game Won!" на първия и "Congratulations!" на втория ред, след което принтира резултата.

gameOver = true;

lcd1.clear();

lcd1.setCursor(0,0);

lcd1.print("Game Won!");

lcd1.setCursor(0,1);

lcd1.print("Congratulations!");

lcd1.print(score);

}

void displayGameOver()

{

Функцията displayGameOver() изчиства LCD дисплея и изписва "Game Over!" на първия и "Score: ", последвано от резултата на втория ред.

gameOver = true;

lcd1.clear();

lcd1.setCursor(0,0);

lcd1.print("Game Over!");

lcd1.setCursor(0,1);

lcd1.print("Score: ");

lcd1.print(score);

score = 1;

}