Roleta RGB

Uma imagem com texto, captura de ecrã, livro, sinalizar

Descrição gerada automaticamente

|  |
| --- |
| ***Formando*** |
| Vítor Hugo Almeida Santos |
| ***Designação do Projeto*** |
| Roleta RGB |

|  |
| --- |
| ***Formando*** |
| Vítor Hugo Almeida Santos |
| ***Designação do Projeto*** |
| Roleta RGB |
| ***Enquadramento*** |
| No âmbito da realização de um pequeno projeto para o Curso: Programação com Arduíno T2, optei pela realização de uma roleta que servirá para jogar.  Este trabalho e as técnicas e processos utilizados, estão relacionados com algumas atividades realizadas ao longo deste ano letivo, nomeadamente: UFCDs da disciplina de Desenho Técnico do curso de Técnico de Mecatrónica, e disciplina de Eletricidade e Eletrónica do curso de Técnico de Eletrónica e Automação.  Será um bom teste aos reflexos dos jogadores. A pedido de formandos de um curso de mecatrónica, que irão participar num evento onde pretendem exibir um jogo para entreter os visitantes de outra escola às instalações das nossas oficinas no próximo dia 15 de abril. |
| ***Objetivos*** |
| Desenvolvimento de código para controlo de Leds RGB endereçáveis e em formato de circunferência;  Ajuste as configurações utilizando uma ferramenta online para criar código com as próprias animações;  Elaboração de esquema elétrico.  Criação de um modelo 3D que servirá de base para a Roleta e para o botão de pressão;  Configuração da impressão 3D;  Montagem e eletrificação;  Pretendo registar os resultados. |
| ***Recursos Necessários*** |
| **Software**:  Arduino IDE;  Tinkercad circuit;  Fusion 360;  Cura;  **Ardware**:  Arduíno (UNO ou NANO);  Computador;  Cabo USB;  Fonte de alimentação 12V;  Leds RGB endereçáveis (4 arcos com 15 leds cada);  Leds RGB endereçáveis (Fita com 5 Leds);  Besouro;  Botão de pressão;  Fios condutores;  PLA (Impressora 3D);  Ferro de soldar;  Solda 60/40; |

Conteúdo

[Imagens do trabalho: 4](#_Toc164003284)

[Funcionamento do jogo: 4](#_Toc164003285)

[Esquema: 5](#_Toc164003286)

[Material: 6](#_Toc164003287)

[NeoPixel Ring de 15 LED 6](#_Toc164003288)

[Arduino Uno R3 7](#_Toc164003289)

[Resistências 7](#_Toc164003290)

[Piezo (besouro) 7](#_Toc164003291)

[Fonte de energia 5V 8](#_Toc164003292)

[Caixa Transparente p/ Arduino 8](#_Toc164003293)

[BreadBoard 830 Contactos 8](#_Toc164003294)

[Orçamento: 9](#_Toc164003295)

[Lista de possíveis fornecedores: 9](#_Toc164003296)

[Software 13](#_Toc164003297)

[Ferramentas 15](#_Toc164003298)

[Datasheets 15](#_Toc164003299)

[**LED NeoPixel** 15](#_Toc164003300)

[**Buzzer** 16](#_Toc164003301)

[**Fonte de alimentação** 16](#_Toc164003302)

[Algumas notas 17](#_Toc164003303)

# Imagens do trabalho:

Uma imagem com texto, eletrónica, Dispositivo eletrónico, aparelho

Descrição gerada automaticamente Uma imagem com jarra, cerâmica, olaria, interior

Descrição gerada automaticamente

Montagem efetuada com recurso a madeira, Cartolina preta e amarela, cortes das letras efetuado com recurso a máquina Laser, roleta e punho para botão de pressão com recurso a impressão 3D.



# Funcionamento do jogo:

* O participante dispõe de um botão de pressão colocado num Joystick;
* No início a roleta apresenta os leds de acordo com a imagem anterior;
* De seguida pressionando o botão uma vez, a roleta coloca aleatoriamente um ou mais leds fixos, e uma sequência de movimento criado pelo acender e apagar um led de cada vez, no sentido dos ponteiros do relógio;
* É nesta fase que o jogador deverá tentar acertar no(s) led(s) aceso(s);
* Quando erra, os leds acendem a vermelho;
* Quando acerta, os leds acendem todos a verde;
* Errando, o jogo volta automaticamente ao inicio;
* Quando acerta, iniciará uma nova ronda mas com um acréscimo de dificuldade dado que o piscar dos leds (sequência) será mais rápida até ao nível máximo (nível 6).

# Esquema:

60 Leds Endereçáveis (a imagem mostra apenas 30)

Uma imagem com texto, eletrónica, Engenharia eletrónica, circuito

Descrição gerada automaticamente

Fonte de alimentação 5V

Botão de pressão

Besouro

Arduíno Uno

60 Leds Endereçáveis

# Material:

|  |  |
| --- | --- |
| Quantidade | Designação |
| 4 | NeoPixel Ring de 15 LED |
| 1 | Faixa de NeoPixel 6 LED |
| 1 | Arduino Uno R3 |
| 1 | Botão pressão |
| 2 | 220 Ω Resistência |
| 1 | 100 Ω Resistência |
| 1 | Piezo (besouro) |
| 1 | Fonte de energia 5V |
| 1 | Caixa Transparente p/ Arduino UNO REV3 - Transparente |
| 1 | Bread Board 830 Contactos |
|  |  |
| 1 | Solda 60/40 |
| 1 | Madeira |
|  | Fio condutor (preferencialmente flexível) |

# NeoPixel Ring de 15 LED

Uma imagem com metal, chão, arte

Descrição gerada automaticamente Uma imagem com círculo, file, Simetria, arte

Descrição gerada automaticamente

Faixa de NeoPixel 6 LED retirada de uma fita originalmente com 60 Leds.

Uma imagem com preto, arte, interior

Descrição gerada automaticamente

# Arduino Uno R3



Botão pressão com respetivos acessórios.



# Resistências



# Piezo (besouro)



# Fonte de energia 5V

uma das pontas foi cortada aproveitando os dois fios (vermelho e preto) que fornecem os 5 Volt.

Uma imagem com cabo, preto e branco

Descrição gerada automaticamente

# Caixa Transparente p/ Arduino

UNO REV3 – Transparente



# BreadBoard 830 Contactos



# Orçamento:

|  |  |
| --- | --- |
| NeoPixel - Arco (1/4 de circunferencia) c/ 15 LEDs RGB WS2812 5050 c/ Driver integrado | 11,75 € |
| WS2812B 30 RGB Led Strip 5050 5V Black - 1mt | 11,50 € |
| Arduino UNO REV3 | 26,03 € |
| Pulsador Redondo OFF-(ON) 250V/1A - Verde | 0,68 € |
| 220 Ω Resistência 0,5 watt | 0,02 € |
| 100 Ω Resistência 0,5 watt | 0,01 € |
| Buzzer Piezoeléctrico com Gerador Interno 3-16Vdc 80dB | 1,17 € |
| Carregador Fast Charging 15W (Preto) - SAMSUNG | 8,92 € |
| Caixa Transparente p/ Arduino UNO REV3 - Transparente | 1,99 € |
| Bread Board 830 Contactos | 2,99 € |
| Filamento PLA 1.75mm 1Kg - Cinzento | 25,89 € |
|  |  |
|  |  |
| total: | 90,95 € |

Valor que pode ser bastante inferior se entrarmos em consideração com o facto de podermos utilizar um Arduíno mais acessível, e que, de 1kg de PLA utilizamos apenas 120g (aproximadamente).

# Lista de possíveis fornecedores:

|  |  |
| --- | --- |
| NeoPixel - Arco (1/4) c/ 15 | [https://www.botnroll.com](https://www.botnroll.com/) |
| WS2812B 30 RGB Led Strip 5050 | [https://www.ptrobotics.com](https://www.ptrobotics.com/) |
| Arduino UNO REV3 | [https://www.castroelectronica.pt](https://www.castroelectronica.pt/) |
| Pulsador Redondo | [https://www.castroelectronica.pt](https://www.castroelectronica.pt/) |
| 220 Ω Resistência 0,5 watt | <https://www.castroelectronica.pt/pt/product/resistencia-12w-220k> |
| 100 Ω Resistência 0,5 watt | [https://www.castroelectronica.pt](https://www.castroelectronica.pt/) |
| Buzzer Piezoeléctrico | [https://www.ptrobotics.com](https://www.ptrobotics.com/) |
| Carregador Fast Charging | [https://www.castroelectronica.pt](https://www.castroelectronica.pt/) |
| Caixa Transparente | [https://www.castroelectronica.pt](https://www.castroelectronica.pt/) |
| Bread Board 830 Contactos | [https://www.castroelectronica.pt](https://www.castroelectronica.pt/) |
| Filamento PLA 1.75mm 1Kg | <https://www.ptrobotics.com/filamento-3d/8193-filamento-pla-175mm-1kg-cinzento.html> |

Código original (apresentado em duas colunas):

#include "FastLED.h"

#define NUM\_LEDS 60

#define DATA\_PIN A0

#define SCORE\_PIN 6

#define SCORE\_LEDS 6

#define BRIGHTNESS 60

CRGB leds[NUM\_LEDS];

CRGB sleds[NUM\_LEDS];

bool reachedEnd = false;

byte gameState = 0;

//byte ledSpeed = 0;

int period = 1000;

unsigned long time\_now = 0;

byte Position = 0;

byte level = 0;

const byte ledSpeed[6] = {50, 40, 30, 20, 14, 7};

//Debounce

bool findRandom = false;

byte spot = 0;

void setup() {

// put your setup code here, to run once:

FastLED.addLeds<WS2812B, DATA\_PIN, GRB>(leds, NUM\_LEDS);

FastLED.addLeds<WS2812B, SCORE\_PIN, GRB>(sleds, SCORE\_LEDS);

pinMode(A3, INPUT\_PULLUP);

Serial.begin(9600);

Serial.println("Reset");

}

void loop() {

// put your main code here, to run repeatedly:

FastLED.setBrightness(BRIGHTNESS );

if (gameState == 0) {

fill\_rainbow(leds, NUM\_LEDS, 0, 20); //2 = longer gradient strip

fill\_rainbow(sleds, SCORE\_LEDS, 0, 40); //2 = longer gradient strip

if (digitalRead(A3) == LOW) {

Position = 0;

findRandom = true;

delay(500);

for (byte i = 0; i < NUM\_LEDS; i++) {

leds[i].setRGB(0, 0, 0);

delay(40);

FastLED.show();

int thisPitch = map (i, 60, 0, 100, 1500);

tone(9, thisPitch,120);

}

for (byte i = 0; i < SCORE\_LEDS; i++) {

sleds[i].setRGB(0, 0, 0);

delay(100);

FastLED.show();

}

gameState = 1;

}

FastLED.show();

}

if (gameState == 1) {

period = ledSpeed[0];

if (millis() > time\_now + period) {

time\_now = millis();

if (findRandom) {

spot = random(56) + 3;

findRandom = false;

}

leds[spot - 1].setRGB(255, 140, 0);

leds[spot].setRGB(0, 255, 0);

leds[spot + 1].setRGB(255, 110, 0);

sleds[0].setRGB(0, 255, 0);

PlayGame(spot - 1, spot + 1);

}

if (digitalRead(A3) == LOW) {

delay(300);

findRandom = false;

if (Position > spot - 1 && Position < spot + 3) {

level = gameState;

gameState = 98;

} else {

gameState = 99;

}

}

}

if (gameState == 2) {

// period = 320;

period = ledSpeed[1];

if (millis() > time\_now + period) {

time\_now = millis();

if (findRandom) {

spot = random(56) + 3;

findRandom = false;

}

leds[spot - 1].setRGB(255, 190, 0);

leds[spot].setRGB(0, 255, 0);

leds[spot + 1].setRGB(255, 190, 0);

sleds[1].setRGB(255, 255, 0);

PlayGame(spot -1, spot +1);

}

if (digitalRead(A3) == LOW) {

delay(300);

findRandom = false;

if (Position > spot - 1 && Position < spot + 3) {

level = gameState;

gameState = 98;

} else {

gameState = 99;

}

}

}

if (gameState == 3) {

period = ledSpeed[2];

if (millis() > time\_now + period) {

time\_now = millis();

if (findRandom) {

spot = random(56) + 3;

findRandom = false;

}

leds[spot].setRGB(0, 255, 0);

sleds[2].setRGB(255, 50, 0);

PlayGame(spot, spot);

}

if (digitalRead(A3) == LOW) {

delay(300);

if (Position == spot+1) {

level = gameState;

gameState = 98;

} else {

gameState = 99;

}

}

}

if (gameState == 4) {

period = ledSpeed[3];

if (millis() > time\_now + period) {

time\_now = millis();

if (findRandom) {

spot = random(56) + 3;

findRandom = false;

}

leds[spot].setRGB(0, 255, 0);

sleds[3].setRGB(255, 0, 0);

PlayGame(spot, spot);

}

if (digitalRead(A3) == LOW) {

delay(300);

if (Position == spot+1) {

level = gameState;

gameState = 98;

} else {

gameState = 99;

}

}

}

if (gameState == 5) {

period = ledSpeed[4];

if (millis() > time\_now + period) {

time\_now = millis();

if (findRandom) {

spot = random(56) + 3;

findRandom = false;

}

leds[spot].setRGB(0, 255, 0);

sleds[4].setRGB(0, 50, 255);

PlayGame(spot , spot);

}

if (digitalRead(A3) == LOW) {

delay(300);

if (Position == spot+1) {

level = gameState;

gameState = 98;

} else {

gameState = 99;

}

}

}

if (gameState == 6) {

period = ledSpeed[5];

if (millis() > time\_now + period) {

time\_now = millis();

if (findRandom) {

spot = random(56) + 3;

findRandom = false;

}

leds[spot].setRGB(0, 255, 0);

sleds[5].setRGB(0, 150, 255);

PlayGame(spot , spot);

}

if (digitalRead(A3) == LOW) {

delay(300);

if (Position == spot+1) {

level = gameState;

gameState = 98;

} else {

gameState = 99;

}

}

}

if (gameState == 98) {

winner();

}

if (gameState == 99) {

loser();

}

}

void PlayGame(byte bound1, byte bound2) {

leds[Position].setRGB(255, 0, 0);

if (Position < bound1 + 1 || Position > bound2 + 1) {

leds[Position - 1].setRGB(0, 0, 0);

}

FastLED.show();

Position++;

if (Position >= NUM\_LEDS) {

leds[Position - 1].setRGB(0, 0, 0);

Position = 0;

}

}

void winner() {

for (byte i = 0; i < 3; i++) {

for (byte j = 0; j < NUM\_LEDS; j++) {

leds[j].setRGB(50, 255, 0);

tone(9, 1000, 250);

}

FastLED.show();

delay(500);

clearLEDS();

FastLED.show();

delay(500);

}

findRandom = true;

Position = 0;

gameState = level + 1;

if (gameState > 6) {

gameState = 0;

}

}

void loser() {

for (byte i = 0; i < 3; i++) {

for (byte j = 0; j < NUM\_LEDS; j++) {

leds[j].setRGB(255, 0, 0);

tone(9, 200, 250);

}

FastLED.show();

delay(500);

clearLEDS();

FastLED.show();

delay(500);

}

gameState = 0;

}

void clearLEDS() {

for (byte i = 0; i < NUM\_LEDS; i++) {

leds[i].setRGB(0, 0, 0);

}

}

void winAll(){

}

# Software

As dias peças foram realizadas com recurso a software de desenho (modelação), Fusion 360, com Ultimaker Cura, impressora 3D, PLA Cinza, um botão de pressão NA, fio condutor, e solda.

Joystick

Uma imagem com esboço, captura de ecrã, texto, desenho

Descrição gerada automaticamente Uma imagem com esboço, texto

Descrição gerada automaticamente

Fusion (Projeto) Fusion (Desenho)

Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente Uma imagem com texto, captura de ecrã, Tipo de letra, logótipo

Descrição gerada automaticamente Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente

Ferramentas do fusion utilizadas: (Forma; esboço; Extrusão, e Casca)

Uma imagem com arte, cilindro, design

Descrição gerada automaticamente Uma imagem com captura de ecrã, desenho, design, arte

Descrição gerada automaticamente Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

Cura (peça completa) (progresso) (tempo necessário)

Circunferência da roleta que serve de base para os Leds.

Uma imagem com texto, captura de ecrã, software, diagrama

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, file, diagrama

Descrição gerada automaticamente

Placa de madeira para suporte da roleta; Cartolina Preta para forrar a madeira; Cartolina amarela para as letras;

Uma imagem com mesa, interior, design, arte

Descrição gerada automaticamente

# Ferramentas

Ferro de soldar

Esponja ou palha de latão

Alicate de corte para eletrónica

Alicate de pontas para eletrónica

# Datasheets

Algumas notas relativas aos componentes

## **LED NeoPixel**

Cada LED NeoPixel é endereçável e cada LED consome ~18mA de corrente constante o que torna a cor muito estável mesmo que a tensão varie ligeiramente. Como o protocolo de comunicação é muito sensível à base de tempo, irá necessitar de microcontrolador para poder controlar estes LEDs RGB. Nota: É apenas 1/4 da circunferência, pelo que terá que adquirir 4 unidades para fazer um circulo completo

"Description:

- Work Voltage: 5V DC

- Width: 10mm

- Length: 1m

- Power: 18W

- Shell color: Black PCB

- LED resource: WS2812B led (5050 SMD RGB LED with built-in improved version of ws2811 IC)

- IC Type: improved version ws2812 IC (built inside the 5050 smd rgb led)

- Grey scale: 256

- Bits/color: 8-bits/color

- Protection rate: Waterproof

- Colors: Full color RGB, dream color changing

- Cuttable: every LED is cuttable

Features:

- Excellent, high quality intelligent lighting!

- Based on high quality SMD5050 RGB LEDs embed (Built-in) integrated WS2811 control ICs.

- Each LED is independently addressable (30 individually addressable LEDs in 1 meter), opening up entirely new lighting possibilities.

- One WS2811 IC control ONE LED Chip.

- With 8 bits of green, red and blue data shifted in for 24 bit color.

Note: These strips are sold per meter! If you buy 5 meters at a time, you'll might get full reels with two connectors. If you buy less than 5m, you'll get a single strip, but it will be a cut piece from a reel which may or may not have a connector on it. "

## **Buzzer**

"Tipo de transdutor de som: sinalizador piezoelétrico

Propriedades de transdutores de som: com gerador externo

Montagem: THT

Frequência de ressonância: 4kHz

Corrente de trabalho: 7mA

Temperatura de trabalho: -20...70°C

Diâmetro: 13.8mm

Altura: 7.5mm

Tensão de trabalho: 3...16V DC

Nível de som: 80dB (d=0.1 m)

Espécie de som: contínuo

Tolerância da frequência de ressonância: ±500Hz

Distância entre pinos: 7.6mm"

## **Fonte de alimentação**

"Carregamento rápido, flexível via USB Type-C. Com o adaptador de carregamento rápido de 15 Watt, pode fornecer energia de forma flexível e rápida novamente através de dispositivos Samsung compatíveis com USB Type-C. E através da ligação USB Type-C com PD 2.0, pode até carregar dispositivos compatíveis de outros fabricantes e sistemas operativos. O adaptador de carregamento rápido pode ser utilizado em várias regiões, apesar das diferentes tensões de rede. Basta ligá-lo a uma tomada com um adaptador de ficha opcional.

- Conector: USB-C

- Entrada: AC 100-240V

- Carregamento: 15W"

# Algumas notas

Relativamente aos Leds 1/4 de circunferência, estes devem ser soldados de acordo com os pinos de entrada DIN e saída DOUT. O pino de saída do primeiro arco deverá ser soldado ao pino de entrada do arco seguinte;

Em ambas as fitas de Led, deve ter-se em conta que o Arduíno ligará ao pino de entrada DIN dos Leds endereçáveis;

O Arduíno utilizado no projeto é original, mas em todo o caso poderá ser utilizado outro tipo de Arduíno, inclusive um NANO;

O botão de pressão deverá ser do tipo NA (Normalmente Aberto);

As resistências poderão ser de outros valos de potência, como por exemplo 1/4 Watt.