

Fogueira com anel de LED endereçável

Objetivo

Simular a luminosidade e cintilar de uma fogueira usando um anel de LEDs endereçáveis. O efeito visual desse projeto é bastante interessante e bonito, pois é bem semelhante às chamas de uma lareira ou fogueira no chão, um projeto excelente para apresentações teatrais e para o conhecimento dos LEDs endereçáveis.

Montagem do circuito

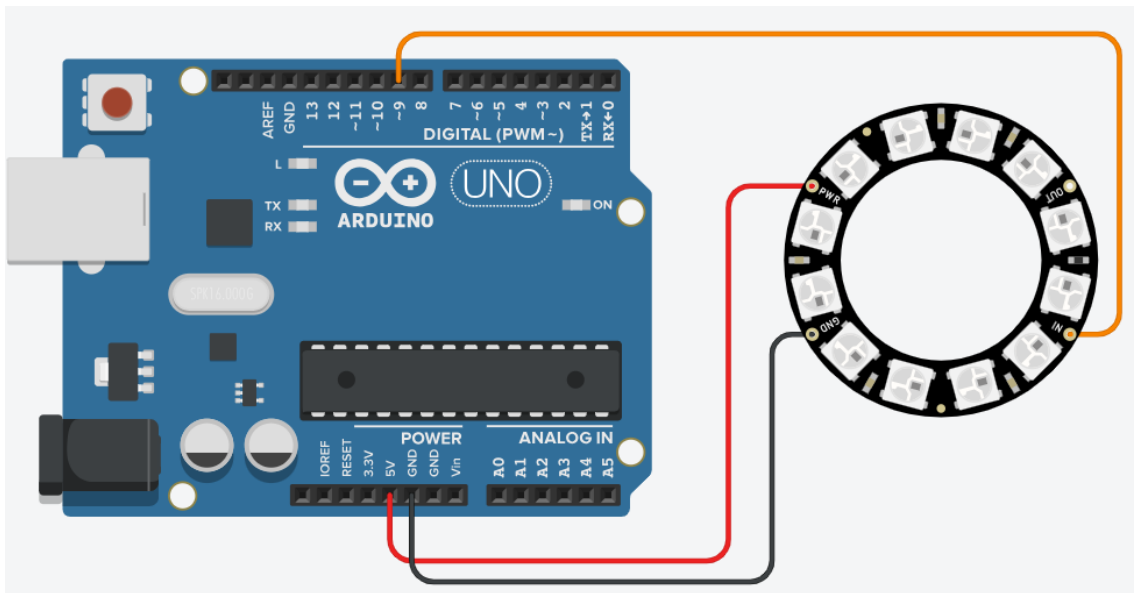


Figura 1 - Esquema de montagem do circuito.

Lista de materiais

1	Arduino UNO
1	Anel de LEDs endereçáveis NeoPixel 12 LEDs
	Cabinhos e fios para ligações

Teste e uso

Por ser um circuito bastante simples, basta ligar à porta USB do computador, ou em uma bateria externa e ver os LEDs começarem a cintilar, imitando o brilho alternado de chamas em uma fogueira.

O anel de LEDs NeoPixel

NeoPixel é o nome comercial da Adafruit para LEDs **endereçáveis individualmente**, baseados no chip **WS2812B**. Isso significa que você pode **controlar cor e brilho de cada LED separadamente** usando **apenas um pino de dados** do microcontrolador (como

Arduino, ESP32, etc.). Esse modelo usado em nosso projeto tem **12 LEDs RGB dispostos em um anel circular**. Cada LED pode exibir **16 milhões de cores (24 bits)**.

Pinout do anel de LEDs NeoPixel de 12 LEDs

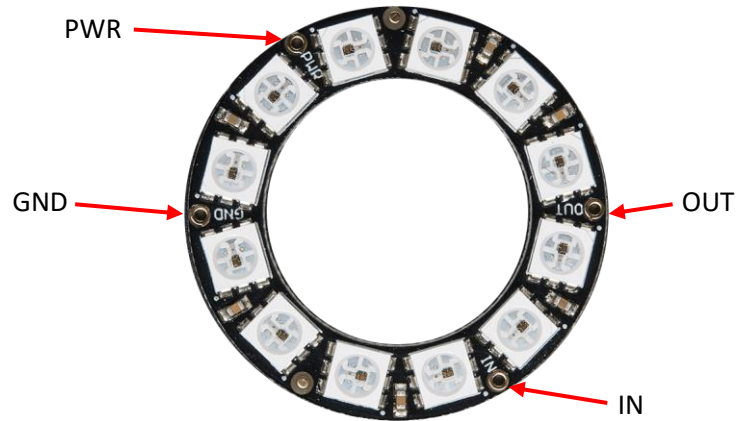


Figura 2 - Anel de LEDs endereçáveis NeoPixel da Adafruit.

Pino	Função
PWR	Alimentação (geralmente 5V)
GND	Terra
IN	Entrada de dados (pino de sinal)
OUT	Saída de dados para outro anel (encadeamento)

Código de programação

```
fogueira.ino
1  /*****
2  * Projeto: Fogueira com anel de LEDs endereçáveis *
3  * Desenvolvedor: Vanderlei Alves *
4  * Data: 14/05/2025 *
5  *****/
6
7  //Biblioteca de comando do anel
8  #include "FastLED.h"
9
10 //Definições do anel de LEDs
11 //Quantidade de LEDs no anel
12 #define NUM_LEDS 12
13 //Pino de comando do anel
14 #define DATA_PIN 9
15
16 //Variáveis globais
17 int Fr1,
18     Fg1,
19     Fr2,
20     Fg2;
21
22 //Instância do objeto LEDs
23 CRGB leds[NUM_LEDS];
24
25 void setup() {
26     //Escolha dos LEDs de modelo WS2812 GRB
27     FastLED.addLeds<WS2812, DATA_PIN, GRB>(leds, NUM_LEDS);
28
29     //Inicializa com valor aleatório usando o canal
30     //analógico A0 como gerador de aleatoriedade
31     randomSeed(analogRead(0));
32 }
33
34 void loop() {
35
36     //Capturando valores aleatórios
37     Fr1 = random(128, 255);
38     Fg1 = random(40, 80);
39     Fr2 = random(150, 255);
40     Fg2 = random(40, 60);
41
42     //Aplicando os valores aleatórios às
43     //cores vermelha e verde, formando diferentes
44     //tons de laranja
45     leds[0] = CRGB(Fr1,Fg1,0);
46     leds[1] = CRGB(Fr2,Fg2,0);
47     leds[2] = CRGB(Fr1,Fg2,0);
48     leds[3] = CRGB(Fr2,Fg1,0);
```

```
49
50   leds[4] = CRGB(Fr1,Fg1,0);
51   leds[5] = CRGB(Fr2,Fg2,0);
52   leds[6] = CRGB(Fr1,Fg2,0);
53   leds[7] = CRGB(Fr2,Fg1,0);
54
55   leds[8] = CRGB(Fr1,Fg1,0);
56   leds[9] = CRGB(Fr2,Fg2,0);
57   leds[10] = CRGB(Fr1,Fg2,0);
58   leds[11] = CRGB(Fr2,Fg1,0);
59
60   //Envia os valores para o anel de
61   //LEDs endereçáveis
62   FastLED.show();
63   delay(40);
64 }
```