

Missão 1.0: Primeiro Circuito Eletrônico - Ascendendo um LED com Arduino

Rogério Moreira Almeida

Introdução:

Neste relatório, descrevemos o processo de montagem de um circuito simples utilizando um Arduino e uma protoboard para piscar um LED com diferentes períodos de tempo. O objetivo é entender os princípios básicos de eletrônica e programação Arduino, incluindo o dimensionamento do resistor para o LED e a programação do Arduino para controlar o LED com diferentes períodos de tempo.

Materiais Utilizados:

Arduino Uno

Protoboard

LED Vermelho (com corrente máxima e tensão direta conhecidas)

Resistores (para limitar a corrente do LED)

Cabos jumper

Computador com acesso à internet para utilizar o TinkerCAD

Arduino IDE (para programação)

Métodos:

Montagem do Circuito no TinkerCAD:

Utilizamos a plataforma TinkerCAD para simular o circuito. Criamos uma nova simulação na aba "Circuits" e montamos o circuito na protoboard, conectando o LED e o resistor ao Arduino conforme especificado.

Dimensionamento do Resistor:

Calculamos o valor do resistor necessário para limitar a corrente do LED. Usamos a lei de Ohm ($V = R * I$), onde V é a tensão do Arduino (5V), I é a corrente máxima suportada pelo LED e R é o valor do resistor.

Para o LED Vermelho sua faixa de tensão é 1,8V – 2,0V. E possuindo corrente máxima de 20mA

$$5 = (0,02 * R) + 2$$

$$3 = 0,02 * R$$

$$R = 150 \text{ ohms}$$

Programação do Arduino:

Utilizamos o Arduino IDE para escrever o código necessário para controlar o LED com diferentes períodos. O código consiste em uma série de instruções para ligar e desligar o LED com diferentes intervalos de tempo.

//Rogério Moreira Almeida

//Missão 1.0: Primeiro Circuito Eletrônico - Ascendendo um LED com Arduino

//Objetivo: Utilizando a placa de prototipagem Arduino para piscar um LED com diferentes //períodos de tempo.

// Definindo o pino do LED

```
const int ledPin = 13;
```

```
void setup() {
```

```
    // Inicializando o pino do LED como saída
```

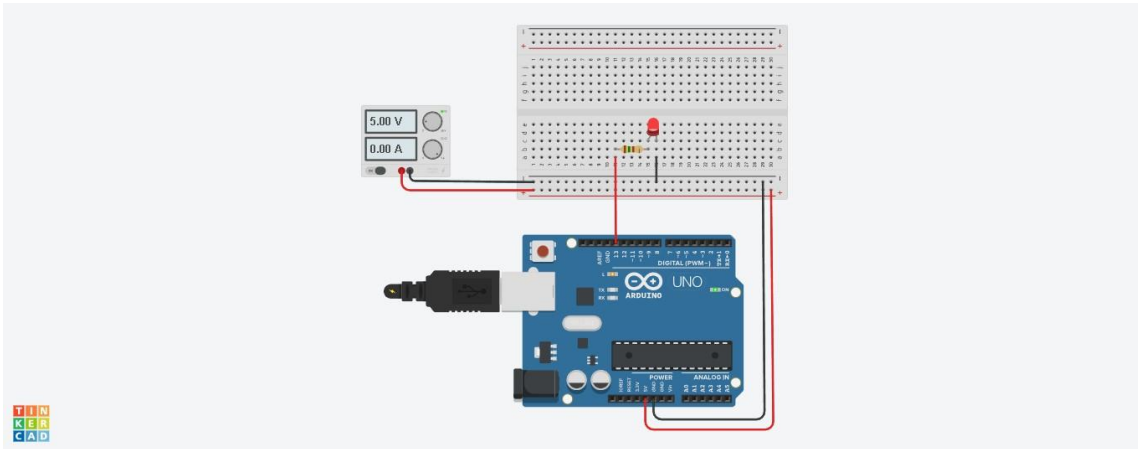
```
    pinMode(ledPin, OUTPUT);
```

```
}  
  
void loop() {  
  
    // Piscar o LED com um período de 500 milissegundos (0.5 segundos)  
  
    digitalWrite(ledPin, HIGH); // Liga o LED  
  
    delay(500); // Aguarda 500 milissegundos  
  
    digitalWrite(ledPin, LOW); // Desliga o LED  
  
    delay(500); // Aguarda mais 500 milissegundos  
  
    // Piscar o LED com um período de 1000 milissegundos (1 segundo)  
  
    digitalWrite(ledPin, HIGH); // Liga o LED  
  
    delay(1000); // Aguarda 1 segundo  
  
    digitalWrite(ledPin, LOW); // Desliga o LED  
  
    delay(1000); // Aguarda mais 1 segundo  
  
    // Piscar o LED com um período de 2000 milissegundos (2 segundos)  
  
    digitalWrite(ledPin, HIGH); // Liga o LED  
  
    delay(2000); // Aguarda 2 segundos  
  
    digitalWrite(ledPin, LOW); // Desliga o LED  
  
    delay(2000); // Aguarda mais 2 segundos  
  
}
```

Teste e Observações:

Após montar o circuito no TinkerCAD e programar o Arduino, testamos o circuito para garantir que o LED estava piscando conforme o esperado. Observamos o comportamento do LED com diferentes períodos e verificamos se o resistor estava dimensionado corretamente para limitar a corrente do LED.

Figura1: Circuito montado no TinKerCAD



Fonte: Autoria própria.

Resultados:

O circuito foi montado com sucesso no TinkerCAD.

O resistor foi dimensionado corretamente para limitar a corrente do LED.

O Arduino foi programado com sucesso para controlar o LED com diferentes períodos de tempo.

O LED piscou conforme o esperado, com períodos de tempo variados.

Conclusão:

Concluimos que é possível montar um circuito simples utilizando um Arduino e uma protoboard para controlar um LED. Através deste projeto, aprendemos sobre os princípios básicos de eletrônica e programação Arduino, incluindo o dimensionamento do resistor para limitar a corrente do LED e a programação do Arduino para controlar o LED com diferentes períodos de tempo. Este projeto serve como uma introdução valiosa ao mundo da eletrônica e da programação embarcada.

Anexo:

Link para a simulação do circuito no TinkerCAD:

https://www.tinkercad.com/things/gdATyC3Vd4Z-brilliant-habbi/editel?sharecode=OY_LB2Eu_yjOzf_J1ZJTU7rSpHj1J2Jv41ym-czPDlo