

💡 Roteiro 1: Acender e Apagar um LED com Arduino 💡

Objetivo: Compreender o funcionamento básico do Arduino controlando um LED (Light Emitting Diode) através de código simples.

Materiais Necessários:

- 1 Placa Arduino (Uno, Nano, etc.)
- 1 Cabo USB para Arduino
- 1 Protoboard (placa de ensaio)
- 1 LED (qualquer cor)
- 1 Resistor de 220 ohms (ou valor próximo)
- 2 Fios jumper
- Computador com o software Arduino IDE instalado

Passos:

1. Montagem do Circuito:

- Identifique o LED: Um LED tem dois terminais: o mais longo é o ânodo (positivo) e o mais curto é o cátodo (negativo).
- Conecte o Resistor: Conecte uma ponta do resistor (220 ohms) ao terminal mais longo (ânodo) do LED. O resistor é crucial para proteger o LED de queimar devido ao excesso de corrente.
- Insira o LED na Protoboard: Coloque o LED na protoboard, de forma que seus terminais (com o resistor conectado ao ânodo) fiquem em trilhas diferentes.
- Conecte ao Arduino (Positivo): Use um fio jumper para conectar a outra ponta do resistor (que está ligada ao ânodo do LED) ao pino digital 13 do Arduino.
- Conecte ao Arduino (Negativo): Use outro fio jumper para conectar o terminal mais curto (cátodo) do LED à trilha GND (Ground/Terra) do Arduino.

2. Preparação do Código (Arduino IDE):

- Abra o software Arduino IDE no seu computador.
- Vá em Arquivo > Exemplos > 01.Basics > Blink. Este é um exemplo pronto para piscar um LED.

3. Entendendo o Código (Blink):

```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT); : Esta linha configura o  
    pino (neste caso, LED_BUILTIN que geralmente é o pino 13)  
    como uma SAÍDA, ou seja, o Arduino vai enviar energia para  
    ele.  
  
    void loop() {  
        digitalWrite(LED_BUILTIN, HIGH); : Liga o LED (manda 5V  
        para o pino).  
  
        delay(1000); : Espera 1000 milissegundos (1 segundo).  
  
        digitalWrite(LED_BUILTIN, LOW); : Desliga o LED (manda  
        0V para o pino).  
  
        delay(1000); : Espera 1000 milissegundos (1 segundo).
```

- **Observação:** No código Blink, LED_BUILTIN é uma constante que representa o pino 13. Se você usou outro pino, troque LED_BUILTIN pelo número do pino (ex: 13).

4. Carregando o Código para o Arduino:

- Conecte o Arduino ao computador usando o cabo USB.
- No Arduino IDE, vá em Ferramentas > Placa e selecione sua placa Arduino (ex: "Arduino Uno").
- Vá em Ferramentas > Porta e selecione a porta serial correspondente ao seu Arduino (geralmente uma porta COM no Windows ou /dev/ttyUSB ou /dev/tty.usbmodem no Linux/Mac).
- Clique no botão "Carregar" (a seta para a direita) na barra de ferramentas.

5. Observação:

O LED conectado ao pino 13 do Arduino deverá começar a piscar em intervalos de 1 segundo. Você pode alterar os valores delay() para mudar a velocidade do piscar.

Roteiro 2: Criar um Semáforo com Arduino

Objetivo: Aprender a controlar múltiplos LEDs em sequência, simulando um semáforo de trânsito.

Materiais Necessários:

- 1 Placa Arduino (Uno, Nano, etc.)
- 1 Cabo USB para Arduino
- 1 Protoboard
- 3 LEDs: 1 vermelho, 1 amarelo, 1 verde
- 3 Resistores de 220 ohms (ou 330 ohms)
- 4 Fios jumper (macho-macho)
- Computador com o software Arduino IDE instalado

Passos:

1. Montagem do Circuito:

- Insira os LEDs na Protoboard: Coloque os três LEDs (vermelho, amarelo, verde) na protoboard, um ao lado do outro, deixando um espaço entre eles. Lembre-se: o terminal mais longo é o positivo (ânodo).
- Conecte os Resistores: Conecte um resistor (220 ohms) ao terminal mais longo (ânodo) de cada LED.
- Conecte os LEDs ao Arduino (Positivo):
 - Use um fio jumper para conectar a outra ponta do resistor do LED VERMELHO ao pino digital 8 do Arduino.
 - Use um fio jumper para conectar a outra ponta do resistor do LED AMARELO ao pino digital 9 do Arduino.
 - Use um fio jumper para conectar a outra ponta do resistor do LED VERDE ao pino digital 10 do Arduino.
- Conecte os LEDs ao Arduino (Negativo - GND Comum): Use um fio jumper para conectar o terminal mais curto (cátodo) de TODOS os LEDs a uma mesma trilha horizontal na protoboard (linha de aterramento). Em seguida, conecte essa trilha horizontal ao pino GND do Arduino.

2. Preparação do Código (Arduino IDE):

Abra o software Arduino IDE.

Copie e cole o seguinte código:

C++

```
// Definindo os pinos para cada LED
```

```
const int ledVermelho = 8;
```

```
const int ledAmarelo = 9;
```

```
const int ledVerde = 10;
```

```
void setup() {
```

```
// Configura os pinos como SAÍDA
```

```
pinMode(ledVermelho, OUTPUT);
```

```
pinMode(ledAmarelo, OUTPUT);
```

```
pinMode(ledVerde, OUTPUT);
```

```
}
```

```
void loop() {
```

```
// Estado VERMELHO (trânsito parado)
```

```
digitalWrite(ledVermelho, HIGH); // Liga o LED vermelho
```

```
digitalWrite(ledAmarelo, LOW); // Desliga o amarelo
```

```
digitalWrite(ledVerde, LOW); // Desliga o verde
```

```
delay(5000); // Permanece vermelho por 5 segundos
```

```
// Estado AMARELO (atenção)
```

```
digitalWrite(ledVermelho, LOW); // Desliga o vermelho
```

```
digitalWrite(ledAmarelo, HIGH); // Liga o LED amarelo
```

```
digitalWrite(ledVerde, LOW); // Desliga o verde
```

```
delay(2000); // Permanece amarelo por 2 segundos
```

SEMANA QUÂNTICA – 4ª Edição / Oficina - PIBID

```
        // Estado VERDE (trânsito liberado)
digitalWrite(ledVermelho, LOW); // Desliga o vermelho
digitalWrite(ledAmarelo, LOW); // Desliga o amarelo
digitalWrite(ledVerde, HIGH); // Liga o LED verde
delay(5000); // Permanece verde por 5 segundos

// Volta para AMARELO antes de fechar para vermelho novamente (opcional,
// mas comum em semáforos)
digitalWrite(ledVermelho, LOW);
digitalWrite(ledAmarelo, HIGH);
digitalWrite(ledVerde, LOW);
delay(2000); // Permanece amarelo por 2 segundos
    }
```

3. Carregando o Código para o Arduino:

- Conecte o Arduino ao computador usando o cabo USB.
 - No Arduino IDE, vá em Ferramentas > Placa e selecione sua placa Arduino.
 - Vá em Ferramentas > Porta e selecione a porta serial correspondente.
 - Clique no botão "Carregar" (a seta para a direita).

4. Observação:

Observe a sequência de luzes do semáforo. Experimente alterar os valores de delay() para mudar a duração de cada cor e crie diferentes padrões de semáforo!