

Documentação do Projeto: Semáforo com Buzzer e Arduino

Participantes: Bruno, Nicolas, Lucas e Vinicius

1. Objetivo

O projeto consiste em simular o funcionamento de um semáforo utilizando **LEDs** (vermelho, amarelo e verde) e um **buzzer ativo** para emitir sinais sonoros.

- **LEDs** → indicam as cores do semáforo.
- **Buzzer** → reforça a sinalização, emitindo sons diferentes para cada estado (bipes no verde e som contínuo no vermelho).

2. Equipamentos Utilizados

1. Placa Arduino (Uno, Nano ou compatível)

- Atua como “cérebro” do projeto, controlando os LEDs e o buzzer.

2. Protoboard (placa de ensaio)

- Usada para montar o circuito de forma simples, sem necessidade de solda.

3. LEDs (3 unidades)

- 1x LED Vermelho
- 1x LED Amarelo (ou Laranja)
- 1x LED Verde

4. Resistores 220 Ω (3 unidades)

- Um para cada LED, garantindo que não queimem devido à corrente do Arduino.

5. Buzzer Ativo (1 unidade)

- Responsável por emitir som.
- O tipo ativo foi escolhido porque consegue gerar o sinal sonoro diretamente, controlado pelos pinos do Arduino.

6. Fios jumper (macho-macho)

- Usados para realizar todas as conexões entre o Arduino, a protoboard e os componentes.

7. Cabo USB

- Para carregar o código no Arduino e fornecer energia ao circuito.

3. Montagem do Circuito

- **LED Verde** → pino digital **8** (via resistor 220Ω) → GND.
- **LED Amarelo** → pino digital **9** (via resistor 220Ω) → GND.
- **LED Vermelho** → pino digital **10** (via resistor 220Ω) → GND.
- **Buzzer Ativo:**
 - Pino **12** → terminal positivo do buzzer.
 - Pino **11** → terminal negativo do buzzer (modo push-pull).
- **Alimentação:** Arduino ligado via cabo USB.

Observação: O buzzer está sendo usado no **modo push-pull**, onde dois pinos digitais alternam o nível lógico para criar o som.

4. Lógica de Funcionamento

1. Sinal Verde

- LED Verde acende por **5 segundos**.
- O buzzer emite **bipes rápidos** (100 ms de som + 150 ms de silêncio repetidamente).

2. Sinal Amarelo

- LED Amarelo acende por **2 segundos**.
- O buzzer **permanece desligado**.

3. Sinal Vermelho

- LED Vermelho acende por **5 segundos**.
- O buzzer toca um **som contínuo** durante esse tempo.

5. Código usado

*

Código para um Semáforo com Buzzer

- Utiliza a técnica Push-Pull nos pinos 11 e 12 para o volume máximo do buzzer.

- Bipes rápidos no sinal verde.
- Som contínuo no sinal vermelho.

```
*/  
  
// --- PINOS DOS LEDs ---  
  
const int pinoLedVermelho = 10;  
  
const int pinoLedAmarelo = 9;  
  
const int pinoLedVerde = 8;  
  
// --- PINOS DO BUZZER (MODO PUSH-PULL) ---  
  
const int buzzerPin1 = 12;  
  
const int buzzerPin2 = 11;  
  
// Frequência do som em Hertz (Hz).  
  
const int frequenciaSom = 3000; // Frequência alta para mais volume  
  
/*  
  
* playPushPull: Esta é a nossa função "Loud Tone".  
  
* Ela já é a versão mais otimizada para gerar um som alto  
  
* porque usa dois pinos para dobrar a voltagem.  
  
*/  
  
void playPushPull(int pin1, int pin2, int frequency, long duration) {  
    if (frequency <= 0) {  
        delay(duration);  
        return;  
    }  
  
    long halfPeriod = 1000000L / frequency / 2;  
  
    long cycles = (long)frequency * duration / 1000;  
  
    for (long i = 0; i < cycles; i++) {  
        digitalWrite(pin1, HIGH);  
        digitalWrite(pin2, LOW);  
        delayMicroseconds(halfPeriod);
```

```
digitalWrite(pin1, LOW);
digitalWrite(pin2, HIGH);
delayMicroseconds(halfPeriod);
}

/*void setup() {
// Configura os pinos dos LEDs como SAÍDA.
pinMode(pinoLedVermelho, OUTPUT);
pinMode(pinoLedAmarelo, OUTPUT);
pinMode(pinoLedVerde, OUTPUT);
// Configura os pinos do buzzer como SAÍDA.
pinMode(buzzerPin1, OUTPUT);
pinMode(buzzerPin2, OUTPUT);
}

void loop() {
// --- SINAL VERDE (AGORA COM BIPES RÁPIDOS) ---
digitalWrite(pinoLedVerde, HIGH);
digitalWrite(pinoLedAmarelo, LOW);
digitalWrite(pinoLedVermelho, LOW);
for (int i = 0; i < 20; i++) {
playPushPull(buzzerPin1, buzzerPin2, frequenciaSom, 100);
delay(150);
}

// --- SINAL AMARELO (SEM SOM) ---
digitalWrite(pinoLedVerde, LOW);
digitalWrite(pinoLedAmarelo, HIGH);
digitalWrite(pinoLedVermelho, LOW);
delay(2000);
```

```
// --- SINAL VERMELHO (AGORA COM SOM CONTÍNUO) ---  
digitalWrite(pinoLedVerde, LOW);  
digitalWrite(pinoLedAmarelo, LOW);  
digitalWrite(pinoLedVermelho, HIGH);  
playPushPull(buzzerPin1, buzzerPin2, frequenciaSom, 5000);  
}
```

6. Principais Pontos de Atenção

- **Resistores obrigatórios nos LEDs:** sem eles, os LEDs podem queimar.
- **Alimentação via USB** é suficiente para esse projeto.
- O **buzzer ativo** simplifica a implementação, pois não exige cálculos complexos de frequência.
- **Modo push-pull** dá mais potência ao som, pois alterna a polaridade em dois pinos do Arduino.
- **Organização da protoboard:** manter fios curtos e bem dispostos facilita a demonstração e evita erros.

7. Possíveis Extensões do Projeto

- Adicionar um **botão** para ativar o semáforo somente quando pressionado.
- Usar **display LCD** para mostrar contagem regressiva dos tempos.
- Controlar os tempos via **potenciômetro**.
- Substituir o buzzer por **módulo MP3** para sons reais de trânsito.

8. Conclusão

Esse projeto integra conceitos básicos de **eletrônica** e **programação com Arduino**. Ele permite entender:

- Controle de saídas digitais (LEDs e buzzer).
- Uso de temporizações (delay).
- Geração de sinais sonoros via código.

O semáforo com buzzer é uma aplicação prática para **aprendizado inicial** em Arduino, podendo ser expandida para sistemas de **trânsito inteligente, sinalização acessível e projetos educacionais**.