

## Documentação do Código para Raspberry Pi Pico W

**Objetivo:** O código desenvolvido para a placa **BitDogLab** tem como objetivo controlar dois LEDs (vermelho e azul) e responder ao aperto de botões para acionar os LEDs. O código implementa uma funcionalidade de piscar os LEDs de acordo com a velocidade configurada e desliga os LEDs quando nenhum botão está sendo pressionado.

### Pinos

- **LED\_RED\_PIN (Pino 12):** Este pino controla o LED vermelho.
- **LED\_BLUE\_PIN (Pino 13):** Este pino controla o LED azul.
- **BUTTON\_PIN\_5 (Pino 5):** Este pino é utilizado para o primeiro botão, que altera a velocidade de piscar dos LEDs.
- **BUTTON\_PIN\_6 (Pino 6):** Este pino é utilizado para o segundo botão, que também altera a velocidade dos LEDs.

### Função `blink_led(int speed)`

A função é responsável por alternar os LEDs de forma intercalada.

- **Como funciona:**
  - A variável `delay_time` é calculada dividindo 500 por `speed`, o que determina o tempo entre cada mudança de estado dos LEDs.
  - O LED vermelho é aceso e o LED azul é apagado e vice versa.
- **Parâmetro:**
  - `speed`: Define a velocidade do piscar dos LEDs. Valores maiores resultam em uma troca mais rápida entre os LEDs.

### Estrutura do Main

O código entra em um loop infinito que observa o estado dos botões e alterna o comportamento dos LEDs de acordo com os botões pressionados.

- **Comportamento:**
  - Se o **botão 5** for pressionado, a função `blink_led(1)` é chamada fazendo os LEDs piscarem com uma velocidade baixa.
  - Se o **botão 6** for pressionado, a função `blink_led(2)` é chamada fazendo os LEDs piscarem com uma velocidade maior.
  - Quando nenhum botão é pressionado os LEDs são apagados.

### Considerações Finais

Este código é simples e demonstra como utilizar entradas e saídas digitais na Placa BitDogLab para controlar LEDs e responder a botões. Ele pode ser expandido para outras funcionalidades, como controlar mais LEDs, display ou implementar comportamentos diferentes baseados nas entradas.

### Requisitos de Hardware

- **Placa BitDogLab**

# CÓDIGO

```
=====
#include "pico/stdlib.h"

#define LED_RED_PIN 12
#define LED_BLUE_PIN 13

#define BUTTON_PIN_5 5
#define BUTTON_PIN_6 6

void blink_led(int speed) {
    int delay_time = 500 / speed;

    gpio_put(LED_RED_PIN, 1);
    gpio_put(LED_BLUE_PIN, 0);
    sleep_ms(delay_time);

    gpio_put(LED_RED_PIN, 0);
    gpio_put(LED_BLUE_PIN, 1);
    sleep_ms(delay_time);
}

int main() {
    gpio_init(BUTTON_PIN_5);
    gpio_set_dir(BUTTON_PIN_5, GPIO_IN);
    gpio_pull_up(BUTTON_PIN_5);

    gpio_init(BUTTON_PIN_6);
    gpio_set_dir(BUTTON_PIN_6, GPIO_IN);
    gpio_pull_up(BUTTON_PIN_6);

    gpio_init(LED_RED_PIN);
    gpio_set_dir(LED_RED_PIN, GPIO_OUT);

    gpio_init(LED_BLUE_PIN);
    gpio_set_dir(LED_BLUE_PIN, GPIO_OUT);
}
```

```
while (1) {
    while (gpio_get(BUTTON_PIN_5) == 0) {
        blink_led(1);
        sleep_ms(50);
    }

    while (gpio_get(BUTTON_PIN_6) == 0) {
        blink_led(2);
        sleep_ms(50);
    }
    gpio_put(LED_BLUE_PIN, 0);
    gpio_put(LED_RED_PIN, 0);
}

return 0;
}
```