# Roteiro de Experimento: Timer com Interrupção na BitDogLab

## 1. Título do Experimento

Disparo de um Alarme Temporizado via Timer e Interrupção no RP2040 (Projeto: isr\_timer\_bitdoglab)

## 2. Objetivo

- Compreender a configuração de temporizadores (Timers) no RP2040.  
- Agendar eventos futuros com precisão usando add\_alarm\_in\_ms().  
- Utilizar interrupções para execução de ações após um tempo programado.

## 3. Materiais Necessários

- Placa BitDogLab.  
- LED integrado à placa (GPIO12).  
- Computador com VSCode e Pico SDK configurado.

## 4. Diagrama de Montagem

• LED conectado ao GPIO12.  
• (A BitDogLab já possui esse LED integrado, sem necessidade de montagem.)

## 5. Procedimento

### 5.1. Inicialização do projeto

• Criar um novo projeto no VSCode com suporte ao Pico SDK.  
• Nome sugerido: isr\_timer\_bitdoglab.

### 5.2. Código Fonte Básico

#include <stdio.h>  
#include "pico/stdlib.h"  
#include "hardware/timer.h"  
  
#define LED 12  
  
int64\_t meu\_callback(alarm\_id\_t id, void \*user\_data) {  
 printf(" Alarme disparado!\n");  
 gpio\_put(LED, 1);  
 return 0;  
}  
  
int main() {  
 stdio\_init\_all();  
 sleep\_ms(10000);  
 printf("Aguardando alarme...\n");  
  
 gpio\_init(LED);  
 gpio\_set\_dir(LED, GPIO\_OUT);  
 gpio\_put(LED, 0);  
  
 add\_alarm\_in\_ms(3000, meu\_callback, NULL, true);  
  
 while (true) {  
 tight\_loop\_contents();  
 }  
}

## 6. Análises e Observações

- Após o início do programa, o sistema aguarda 10 segundos.  
- Em seguida, um alarme é agendado para 3 segundos depois.  
- Quando o alarme dispara, o LED acende e uma mensagem é exibida no terminal.  
- Discuta:  
 • Vantagem de usar Timer + callback versus sleep\_ms().  
 • Impacto de alterar o tempo programado.

## 7. Extensões e Desafios

- Agendar múltiplos alarmes.  
- Fazer o LED piscar após o alarme.  
- Criar uma sequência de eventos com múltiplos LEDs.  
- Implementar o cancelamento do alarme usando cancel\_alarm().

## 8. Importante

- A função de callback deve ser rápida e eficiente.  
- Programar corretamente o tempo no add\_alarm\_in\_ms().  
- O tight\_loop\_contents() mantém o sistema ativo aguardando eventos.