Anexo A2 – Interrupção por Botão com Ação Imediata

# Descrição

Este exemplo implementa o modelo Foreground/Background, em que tarefas rotineiras são executadas no laço principal e eventos assíncronos são tratados por interrupções. O uso de interrupções permite respostas imediatas a entradas externas.

# Objetivo

- Piscar um LED principal (GPIO 12) no laço principal  
- Detectar botão pressionado (GPIO 5) via interrupção  
- Acionar outro LED (GPIO 11) de forma imediata como resposta ao evento

# Requisitos

- Placa BitDogLab (RP2040W)  
- SDK oficial do RP2040  
- LED principal no GPIO 12  
- Botão no GPIO 5 com pull-up  
- LED de alarme no GPIO 11

# Código-fonte (main.c)

#include <stdio.h>  
#include "pico/stdlib.h"  
  
#define LED\_PRINCIPAL 12  
#define LED\_ALARME 11  
#define BOTAO 5  
  
volatile bool alarme\_ativo = false;  
  
void gpio\_callback(uint gpio, uint32\_t events) {  
 if (gpio == BOTAO && (events & GPIO\_IRQ\_EDGE\_FALL)) {  
 alarme\_ativo = true;  
 }  
}  
  
int main() {  
 stdio\_init\_all();  
  
 gpio\_init(LED\_PRINCIPAL);  
 gpio\_set\_dir(LED\_PRINCIPAL, GPIO\_OUT);  
  
 gpio\_init(LED\_ALARME);  
 gpio\_set\_dir(LED\_ALARME, GPIO\_OUT);  
  
 gpio\_init(BOTAO);  
 gpio\_set\_dir(BOTAO, GPIO\_IN);  
 gpio\_pull\_up(BOTAO);  
  
 gpio\_set\_irq\_enabled\_with\_callback(BOTAO, GPIO\_IRQ\_EDGE\_FALL, true, &gpio\_callback);  
  
 while (true) {  
 gpio\_put(LED\_PRINCIPAL, 1);  
 sleep\_ms(500);  
 gpio\_put(LED\_PRINCIPAL, 0);  
 sleep\_ms(500);  
  
 if (alarme\_ativo) {  
 gpio\_put(LED\_ALARME, 1);  
 sleep\_ms(200);  
 gpio\_put(LED\_ALARME, 0);  
 alarme\_ativo = false;  
 }  
 }  
}

# CMakeLists.txt (mínimo funcional)

cmake\_minimum\_required(VERSION 3.13)  
include(pico\_sdk\_import.cmake)  
  
project(foreground\_background\_interrupt C CXX ASM)  
pico\_sdk\_init()  
  
add\_executable(fg\_bg  
 main.c  
)  
  
target\_link\_libraries(fg\_bg pico\_stdlib)  
pico\_enable\_stdio\_usb(fg\_bg 1)  
pico\_enable\_stdio\_uart(fg\_bg 0)  
  
pico\_add\_extra\_outputs(fg\_bg)

# Observações Práticas

- A interrupção permite detectar o botão mesmo quando o laço principal está ocupado com outra tarefa.  
- Ideal para tratamento de eventos assíncronos, como alarmes e sensores.  
- Pode ser expandido com debounce ou prioridade de interrupções.