Anexo 10 – Contador

Este experimento demonstra como utilizar o Timer como um “Contador” para contar eventos externos, como por exemplo, pulsos de um sensor, número de vezes que um botão é acionado, etc.

## Objetivo

- Demonstrar o uso de Timers como Contadores de eventos externo;  
- Contar o número de vezes que um botão é pressionado e imprimir na porta serial;

## Código-fonte (main.c)

#include <stdio.h>

#include "pico/stdlib.h"

#include "hardware/gpio.h"

// Define o pino que receberá os eventos externos (pulso de descida EDGE\_FALL)

#define BOTAO 5

volatile uint32\_t contador = 0;

// Função de callback chamada na interrupção

void gpio\_callback(uint gpio, uint32\_t events) {

if (gpio == BOTAO && (events & GPIO\_IRQ\_EDGE\_FALL)) {

contador++;

}

}

int main() {

// Inicializa a comunicação serial para monitorar o contador

stdio\_init\_all();

// Configura o pino como entrada

gpio\_init(BOTAO);

gpio\_set\_dir(BOTAO, GPIO\_IN);

gpio\_pull\_up(BOTAO); // Resistor de PULL\_UP

// Configura a interrupção para detectar borda de descida

gpio\_set\_irq\_enabled\_with\_callback(BOTAO, GPIO\_IRQ\_EDGE\_FALL, true, &gpio\_callback);

while (true) {

printf ("Contador de eventos: %u\n", contador);

sleep\_ms(1000); // Atualiza a cada segundo

}

return 0;

}

# Observações Práticas

- Incluir a Biblioteca “hardware/gpio.h” no CMakeList.txt, como indicado abaixo e, em seguida, salvar o arquivo CMakeList.txt:

