Anexo 7 – Temporizador de Hardware 1

# Descrição

Este experimento usa um temporizador de hardware repetitivo (periódico) para acionar uma função (repeating\_timer\_callback()) a cada segundo. O temporizador é configurado para acionar automaticamente a função de callback, que imprime a mensagem.

# Objetivo

- Implementar um Timer de Hardware (com o uso da biblioteca “hardware/timer.h”;  
- Mostrar com o funciona um Time repetitivo;  
- Realizar uma tarefa, através da função de Callback, quando o tempo é alcançado.

# Código-fonte (main.c)

#include <stdio.h> // Inclui a biblioteca padrão de entrada e saída para uso de funções como printf.

#include "pico/stdlib.h" // Inclui a biblioteca padrão para funcionalidades básicas como GPIO, temporização e comunicação serial.

#include "hardware/timer.h" // Inclui a biblioteca para gerenciamento de temporizadores de hardware.

// Função de callback que será chamada repetidamente pelo temporizador

// O tipo bool indica que a função deve retornar verdadeiro ou falso para continuar ou parar o temporizador.

bool repeating\_timer\_callback(struct repeating\_timer \*t) {

// Imprime uma mensagem na saída serial indicando que 1 segundo se passou.

printf("1 segundo passou\n");

// Retorna true para manter o temporizador repetindo. Se retornar false, o temporizador para.

return true;

}

int main() {

// Inicializa a comunicação serial, permitindo o uso de funções como printf.

// Necessário para enviar mensagens através da interface USB ou UART.

stdio\_init\_all();

// Declaração de uma estrutura de temporizador de repetição.

// Esta estrutura armazenará informações sobre o temporizador configurado.

struct repeating\_timer timer;

// Configura o temporizador para chamar a função de callback a cada 1 segundo.

// Parâmetros:

// 1000: Intervalo de tempo em milissegundos (1 segundo).

// repeating\_timer\_callback: Função de callback que será chamada a cada intervalo.

// NULL: Dados adicionais passados para a função de callback (não utilizados neste caso).

// &timer: Ponteiro para a estrutura que armazenará informações do temporizador.

add\_repeating\_timer\_ms(1000, repeating\_timer\_callback, NULL, &timer);

// Loop infinito que mantém o programa em execução.

// Aqui você pode adicionar outras tarefas que deseja que o microcontrolador execute.

while (true) {

// Pausa de 1 segundo para reduzir o uso da CPU.

// Embora o temporizador opere independentemente, esta pausa evita que o loop

// seja executado continuamente sem realizar nada útil.

sleep\_ms(1000);

}

// Código nunca chegará aqui devido ao loop infinito, mas é uma boa prática incluir return.

return 0;

}

# Observações Práticas

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

