Resumo do Projeto: Piscar LED com FreeRTOS no Raspberry Pi Pico

# 1) Objetivos

Este projeto tem como finalidade demonstrar o uso básico de FreeRTOS no Raspberry Pi Pico para controle de hardware em tempo real. O foco é:  
- Criar uma única tarefa (`led\_task\_function`) usando o FreeRTOS.  
- Alternar o estado de um LED conectado ao GPIO 12.  
- Utilizar `vTaskDelay()` para temporização precisa sem bloqueio da CPU.  
- Mostrar mensagens no terminal (via `printf`) indicando o estado do LED (ON/OFF).

# 2) Implementação na BitDogLab

O código pode ser facilmente implementado e carregado na placa BitDogLab, que usa o Raspberry Pi Pico como núcleo e já fornece acesso prático aos GPIOs. O GPIO 12 pode ser usado diretamente para ligar/desligar um LED com resistor em série.  
  
Compilação e upload:  
- Utiliza-se o VSCode com SDK do Pico + FreeRTOS  
- O terminal USB da BitDogLab permite observar o `printf` com `stdio\_init\_all()`.

# 3) Onde ver os resultados

Com o código gravado na placa:  
- O LED no GPIO 12 irá piscar a cada 500ms (meio segundo).  
- No terminal serial (ex: PuTTY ou Minicom), o usuário verá:  
  
 LED ON  
 LED OFF  
 LED ON  
 LED OFF  
 ...  
  
- O `sleep\_ms(2000)` no início garante tempo de inicialização da USB antes de imprimir.

# 4) Alterar a prioridade da tarefa

No trecho da função `xTaskCreate`, a prioridade está definida como `1`:  
  
xTaskCreate(  
 led\_task\_function,  
 "LED\_Task",  
 256,  
 NULL,  
 1, // <- Prioridade da tarefa  
 &led\_task\_handle  
);  
  
Para alterar a prioridade:  
- Basta trocar o valor `1` por outro inteiro.  
- Exemplo: `5` indica prioridade maior do que tarefas com prioridade `1`.  
  
No caso deste projeto com uma única tarefa, a prioridade não impacta, mas é fundamental em projetos com múltiplas tarefas para determinar qual deve ser executada primeiro.