

Semáforo Didático com FreeRTOS: LED RGB Controlado por Tarefas

- Configure os pinos GPIO11 (verde), GPIO12 (azul) e GPIO13 (vermelho) como saídas digitais.
- Certifique-se de desligar os três canais no início da execução para evitar estados indesejados.
- Implemente uma tarefa no FreeRTOS que controle a alternância das cores do LED RGB com base nos tempos definidos.
- A sequência de execução deve ser: vermelho → verde → amarelo → repetir.

```
#include "FreeRTOS.h"
#include "task.h"
#include "semphr.h"
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/gpio.h"

// Pinos dos LEDs
#define LED_VERMELHO 13
#define LED_VERDE 11
#define LED_AZUL 12

// Semáforos
SemaphoreHandle_t sema_vermelho;
SemaphoreHandle_t sema_verde;
SemaphoreHandle_t sema_amarelo;

void setup_all() {
    gpio_init(LED_VERMELHO);
    gpio_set_dir(LED_VERMELHO, GPIO_OUT);
    gpio_put(LED_VERMELHO, 0);

    gpio_init(LED_VERDE);
    gpio_set_dir(LED_VERDE, GPIO_OUT);
    gpio_put(LED_VERDE, 0);

    gpio_init(LED_AZUL);
    gpio_set_dir(LED_AZUL, GPIO_OUT);
    gpio_put(LED_AZUL, 0);
}

// Tarefa: LED Vermelho
void tarefa_vermelha(void *pvParameters) {
    while (true) {
        xSemaphoreTake(sema_vermelho, portMAX_DELAY);
        printf("Sinal VERMELHO\n");
        gpio_put(LED_VERMELHO, 1);
        vTaskDelay(pdMS_TO_TICKS(5000));
        gpio_put(LED_VERMELHO, 0);
        xSemaphoreGive(sema_verde); // Libera próxima tarefa
    }
}
```

```

// Tarefa: LED Verde
void tarefa_verde(void *pvParameters) {
    while (true) {
        xSemaphoreTake(sema_verde, portMAX_DELAY);
        printf("Sinal VERDE\n");
        gpio_put(LED_VERDE, 1);
        vTaskDelay(pdMS_TO_TICKS(5000));
        gpio_put(LED_VERDE, 0);
        xSemaphoreGive(sema_amarelo); // Libera próxima tarefa
    }
}

// Tarefa: LED Amarelo
void tarefa_amarela(void *pvParameters) {
    while (true) {
        xSemaphoreTake(sema_amarelo, portMAX_DELAY);
        printf("Sinal AMARELO\n");
        gpio_put(LED_VERDE, 1);
        gpio_put(LED_VERMELHO, 1);
        vTaskDelay(pdMS_TO_TICKS(3000));
        gpio_put(LED_VERDE, 0);
        gpio_put(LED_VERMELHO, 0);
        xSemaphoreGive(sema_vermelho); // Volta para a primeira
    }
}

int main() {
    stdio_init_all();
    setup_all();
    sleep_ms(2000);
    printf("Iniciando tarefas com sincronização...\n");

    // Criação dos semáforos binários
    sema_vermelho = xSemaphoreCreateBinary();
    sema_verde = xSemaphoreCreateBinary();
    sema_amarelo = xSemaphoreCreateBinary();

    // Começa com vermelho liberado
    xSemaphoreGive(sema_vermelho);

    // Criação das tarefas
    xTaskCreate(tarefa_vermelha, "vermelha", 256, NULL, 1, NULL);
    xTaskCreate(tarefa_verde, "verde", 256, NULL, 1, NULL);
    xTaskCreate(tarefa_amarela, "amarela", 256, NULL, 1, NULL);

    vTaskStartScheduler();

    while (1) {
        tight_loop_contents();
    }
}

```