



Clock e Temporizadores

TAREFA INDIVIDUAL



Prof. Dr Ricardo Menezes Prates

Unidade 4 | Capítulo 5 – C50123A

29/01/2025

Executores:



INSTITUTO FEDERAL
Piauí



INSTITUTO FEDERAL
Rio Grande do Norte



INSTITUTO FEDERAL
Maranhão



INSTITUTO FEDERAL
Ceará



Coordenação:



Iniciativa:



Atividade 1: Temporizador periódico.

Com o emprego da função ***add_repeating_timer_ms()***, presente na ferramenta Pico SDK, projete um semáforo com temporização de 3 segundos para cada alteração de sinal. O fluxograma simplificado de atuação do semáforo pode ser visualizado na Figura 1. Nesta prática, será necessário simular os seguintes componentes (configuração sugerida está presente na Figura 2):

- 1) Microcontrolador Raspberry Pi Pico W.
- 2) 03 LEDs (vermelho, amarelo e verde).
- 3) 03 Resistores de 330 Ω .

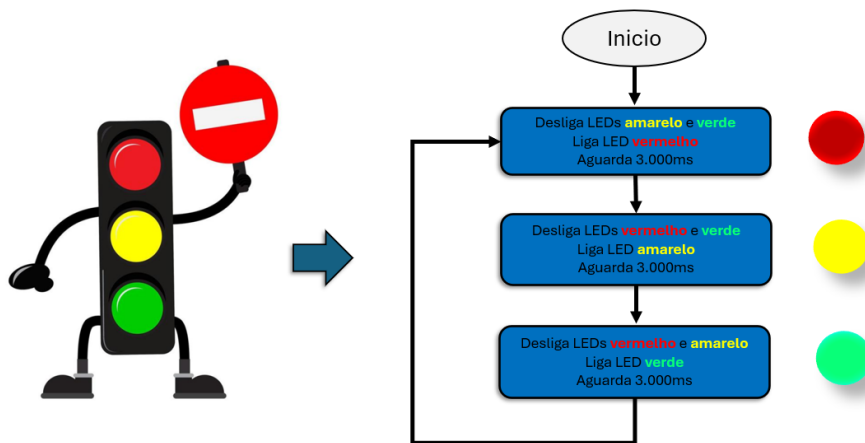


Figura 1 – Fluxograma simplificado de atuação do semáforo.

Fonte: autor

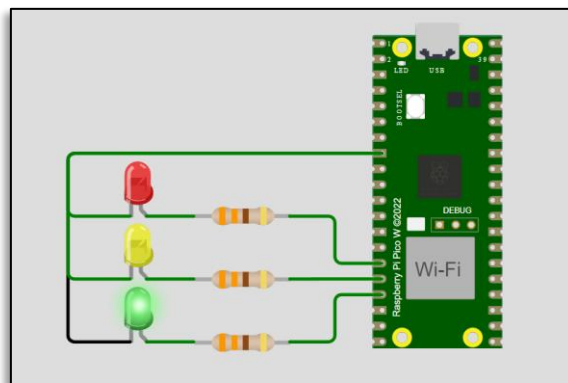


Figura 2 – Simulação sugerida no Wokwi para esta atividade. O vídeo associado a esta prática pode ser obtido no link a seguir:

<https://www.dropbox.com/scl/fi/6w37qxdq4ytljqvqzr6h/2025-01-25-17-48-11.mkv?rlkey=yjl1iqcfkx444xigglaxqw81e&dl=0>

Fonte: autor

Os requisitos para a realização desta atividade são:

- 1) O acionamento dos LEDs (sinais do semáforo) deve iniciar na cor vermelha, conforme orientação presente na Figura 1, alterando para amarela e, em seguida, verde.

- 2) O temporizador deve ser ajustado para um atraso de 3 segundos (3.000ms).
- 3) A mudança de estado dos LEDs deve ser implementada na função de *call-back* do temporizador, a exemplo da rotina trabalhada na aula síncrona – *repeating_timer_callback()*.
- 4) A rotina principal, presente no interior da estrutura de repetição *while*, deve imprimir algum tipo de informação a cada segundo (1.000 ms) – a mensagem enviada pela porta serial fica a critério do discente.
- 5) Com o emprego da Ferramenta Educacional BitDogLab, faça um experimento com o código deste exercício utilizando o LED RGB – GPIOs 11, 12 e 13.

Atividade 2: Temporizador de um disparo (One Shot).

Com o emprego da função *add_alarm_in_ms()*, presente na ferramenta Pico SDK, projete um sistema de temporização para o acionamento de LEDs, que atua a partir do clique em um botão (*pushbutton*). Nesta prática, será necessário simular os seguintes componentes (simulação sugerida presente na Figura 2):

- 1) Microcontrolador Raspberry Pi Pico W.
- 2) 03 LEDs (azul, vermelho e verde).
- 3) 03 Resistores de 330 Ω .
- 4) Botão (Pushbutton).

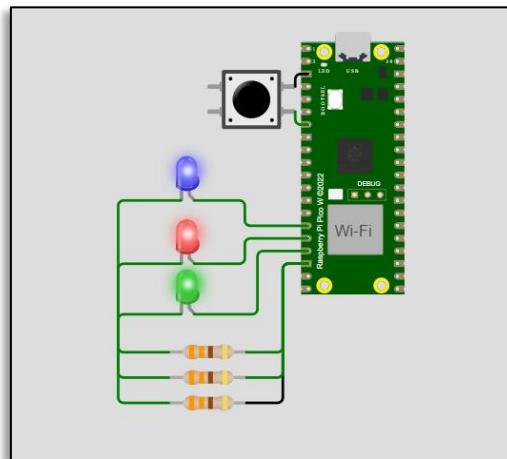


Figura 3 – Simulação sugerida no Wokwi para esta atividade. O vídeo associado a esta prática pode ser obtido no link a seguir:
<https://www.dropbox.com/scl/fi/7faa7ib6bjrxncr2dxnf7/2025-01-26-14-01-54.mkv?rlkey=wqvef7zpxpuwvz2joj6qucasd&dl=0>

Fonte: autor

Os requisitos para a realização desta atividade são:

- 1) Caso o usuário clique no botão (*pushbutton*), os três LEDs serão ligados (todos em nível alto). A partir da primeira rotina de

atraso, ocorrerá uma mudança de estado para dois LEDs ligados e, em seguida, apenas um. **Obs.: veja o vídeo associado a esta prática no link presente na Figura 3.**

- 2) O temporizador do alarme deve ser ajustado para um atraso de 3 segundos (3.000ms), entre os estados de acionamento dos LEDs.
- 3) A mudança de estado dos LEDs deve ser implementada em funções de *call-back* do temporizador, a exemplo da rotina trabalhada na aula síncrona - ***turn_off_callback()***.
- 4) O botão só pode alterar o estado dos LEDs quando o último LED for desligado. Deste modo, durante a execução das rotinas de temporização, o botão não pode iniciar uma nova chamada da função *call-back*.
- 5) Com o emprego da Ferramenta Educacional BitDogLab, faça um experimento com o código deste exercício utilizando o LED RGB – GPIOs 11, 12 e 13 e o Botão A, GPIO 05.
- 6) **Opcional**: Implementar uma rotina em *software* para atenuação do efeito *bouncing* no botão (*software debounce*).

Destacamos aqui alguns passos necessários para este projeto.

- Deve-se utilizar o ambiente de desenvolvimento VS Code em todas as duas atividades;
- Os códigos devem ser escritos na linguagem C, juntamente com os recursos do Kit de Desenvolvimento de *Software* Pico SDK;
- O simulador Wokwi deve estar integrado ao ambiente de desenvolvimento VS Code;
- Deve-se criar um repositório no github para o versionamento, registro e submissão dessa atividade.

Orientação para o desenvolvimento da tarefa:

- **Esta tarefa é individual.**
- Deve-se implementar testes básicos para garantir que cada recurso esteja funcionando corretamente.
- É necessário adicionar um arquivo README.md com instruções de uso do programa.



Prazo:

Você tem 5 dias para enviar essa atividade.