



Microcontroladores – GPIO

ATIVIDADE 2



Prof. Dr Ricardo Menezes Prates

Unidade 4 | Capítulo 2 – 201234A

14/01/2025

Executores:



INSTITUTO FEDERAL
Rio de Janeiro



INSTITUTO FEDERAL
Rio Grande do Norte



INSTITUTO FEDERAL
Maranhão



INSTITUTO FEDERAL
Ceará



Coordenação:



Iniciativa:



Atividade 2 (Participativa): Controle de pinos GPIO da ferramenta educacional BitDogLab.

Com o emprego da ferramenta educacional BitDogLab, use a porta serial UART para controlar pinos GPIO do microcontrolador RP2040. Para a utilização dos periféricos ilustrados na *Figura 1* (itens 1 e 2), faça o acionamento do LED RGB (GPIOs 11, 12 e 13), juntamente com o controle de sinal sonoro de um buzzer, conectado à GPIO 21 – Botão A (item 1), ilustrado na *Figura 1*. Como forma de comunicação entre o usuário e o sistema embarcado, deve-se escrever comandos (*conjuntos de caracteres*) para serem interpretados pelo microcontrolador de modo a realizar o controle dos periféricos trabalhados. Para esta prática, será necessária a utilização dos seguintes componentes e acessórios.

- 1) Ferramenta educacional BitDogLab (versão 6.3).
- 2) Cabo tipo micro-usb para usb-a.
- 3) Computador pessoal.

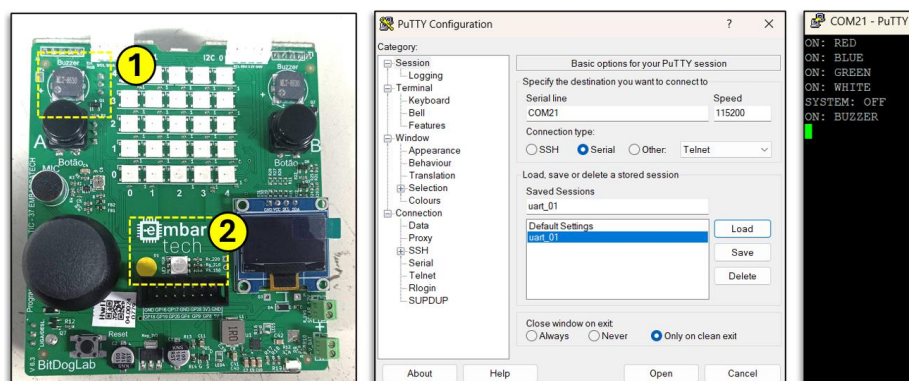


Figura 1 – Uso de pinos GPIOs para o controle de periféricos da ferramenta educacional BitDogLab.

Figura 1 #detalhamento: A imagem à esquerda se refere a ferramenta educacional BitdogLab, destacando os periféricos: 1) componente buzzer, conectado ao pino GPIO21, e 2) LED RGB, conectado aos pinos GPIOs 11, 12 e 13. As duas imagens à direita tratam do software de emulação de terminal PuTTY, que será útil para estabelecer uma conexão com o microcontrolador RP2040 através da interface serial UART. O vídeo associado a esta prática pode ser acessado no link:

<https://www.dropbox.com/scl/fi/jevku9wjif9zy1iechuxd/VID-20250109-WA0044.mp4?rlkey=4cb9qag7ziis20u8ivjgtvmdk&dl=0>

Fonte: autor

Vale mencionar que as estratégias de acionamento dos LEDs e de geração do sinal elétrico do buzzer ficarão a critério da equipe de desenvolvimento. Contudo, em caráter sugestivo, é apresentado um vídeo com uma [prática associada a esta tarefa](#). Recomenda-se que a equipe de desenvolvimento utilize o *software* de emulação de terminal PuTTY de modo a estabelecer a comunicação entre o computador e o microcontrolador. As rotinas necessárias para contemplar os requisitos desta tarefa são:

- 1) Ligar LED verde (GPIO 11) – desligar as demais GPIOs;
- 2) Ligar LED azul (GPIO 12) – desligar as demais GPIOs;
- 3) Ligar LED vermelho (GPIO 13) – desligar as demais GPIOs;
- 4) Ligar os três LEDs (luz branca);
- 5) Desligar todos os LEDs;
- 6) Acionar o *buzzer* por 2 segundos – emissão de sinal sonoro;
- 7) Sair do modo de execução e habilitar o modo de gravação via software (reboot) – **rotina opcional**.

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 do github para o versionamento, registro e submissão dessa atividade.

Formação das Equipe:

- A equipe deve ser constituída de, no máximo, 10 integrantes. Ela pode ser dividida em 1 Líder (faz a gestão do repositório) e 9 desenvolvedores – cada um com uma tarefa específica.
- Todos os participantes do grupo devem enviar esta atividade (fazer parte do repositório do github) e devem possuir um cadastro/perfil no github que viabilize a identificação do autor por parte dos mentores do Embarcotech.

Orientação para o desenvolvimento da tarefa:

- Cada membro do projeto deve criar um branch na ferramenta git.
- Os desenvolvedores devem trabalhar em suas tarefas.
- Deve-se realizar commits regulares com mensagens descritivas.
- Desenvolvedores devem realizar pull requests.
- O Líder deve revisar o código e aprovar os pull requests.
- 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.
- **Como etapa final, a equipe irá produzir um vídeo ensaio, com duração máxima de 3 minutos, apresentando a solução trabalhada e os resultados obtidos nos experimentos. O link deste vídeo deve estar presente no arquivo README.md do repositório do github.**



Prazo:

Você tem 5 dias para enviar essa atividade.