

BitDogLab – Perguntas Frequentes (FAQ)

Informações Gerais

P: O que é a BitDogLab?

R: A BitDogLab é uma placa eletrônica open source criada no âmbito do Projeto Escola 4.0 da Unicamp. Baseada na placa Raspberry Pi Pico H ou W, ela foi desenvolvida para tornar o ensino de programação, robótica e eletrônica acessível e prático para estudantes do ensino fundamental II, médio, pode ser aplicada em aulas de sistemas embarcados em disciplinas de graduação ou pós – graduação como foi o caso da disciplina IE323 – Tópicos em eletrônica com abordagem Steam e também o projeto Embarcatech.

P: Qual é o objetivo do Projeto Escola 4.0?

R: Levar de forma prática, divertida e acessível os conceitos de programação, pensamento computacional e robótica para as escolas brasileiras, promovendo a educação tecnológica desde cedo.

P: A BitDogLab é um projeto aberto?

R: Sim. O projeto da BitDogLab é 100% aberto. Os arquivos de hardware e software estão disponíveis no GitHub no endereço <https://github.com/BitDogLab/BitDogLab> e qualquer pessoa pode fabricar, montar e modificar a placa.

P: Onde encontro mais informações sobre a BitDogLab?

R: Todas as informações estão disponíveis no repositório oficial: <https://github.com/BitDogLab/BitDogLab>

Componentes e Conexões da Placa

P: Quais componentes estão integrados na BitDogLab?

R: A placa conta com: LED RGB, dois botões (A e B), buzzer passivo, matriz WS2812B com 25 LEDs, joystick analógico, display OLED I2C, microfone de eletreto, barra de terminais, conectores I2C/UART e um conector IDC para expansão.

P: Como estão conectados os LEDs RGB?

R:

- Vermelho: GPIO13 com resistor de 220 ohms
- Verde: GPIO11 com resistor de 220 ohms
- Azul: GPIO12 com resistor de 150 ohms

P: Como os botões A e B funcionam?

R: O Botão A está no GPIO5 e o Botão B no GPIO6. Ambos usam resistores de pull-up internos. Quando pressionados, os GPIOs mudam de HIGH para LOW.

Exemplo :

<https://github.com/BitDogLab/BitDogLab/blob/main/software/Bot%C3%B5es/teste%20com%20os%20botoes.py>

P: Qual é a conexão do buzzer?

R:

- Buzzer A: GPIO21 (via transistor)
- Buzzer B: GPIO10

Exemplo:

https://github.com/BitDogLab/BitDogLab/blob/main/software/Buzzer/marcha_imperial%20GPIO%2021%20e%208.py

P: Como uso a matriz de LEDs (Neopixel)?

R: A entrada (DIN) da matriz WS2812B está conectada ao GPIO7. A matriz possui 5 linhas por 5 colunas (25 LEDs).

Exemplo:

<https://github.com/BitDogLab/BitDogLab/blob/main/software/Matriz%20de%20LEDs%20RGB%205%20x%205/Matriz%20de%20LEDs%205%20x%205%20efeito%20rostinho%20feliz%20com%20piscada.py>

P: Como funciona o joystick analógico?

R:

- VRx: GPIO27
- VRy: GPIO26
- Botão: GPIO22 (com pull-up interno)

Exemplo

<https://github.com/BitDogLab/BitDogLab/blob/main/software/Joystick/joystick%20V2A%20com%20fun%C3%A7%C3%A3o%20bot%C3%A3o.py>

P: E o display OLED?

R: O OLED I2C (128x64) usa:

- SDA: GPIO14
- SCL: GPIO15

Pode-se usar SoftI2C para comunicação robusta. Endereço padrão: 0x3C.

P: Como uso o microfone analógico?

R: O microfone está conectado ao GPIO28. Ele gera um sinal analógico com nível médio de 1,65V.

P: O que é o conector IDC?

R: Um conector de 14 pinos que permite expansão com módulos como BitMoveI Motor Driver ou LoRa. Contém GPIOs, GND, 3V3, 5V e comunicação SPI.

P: Quais são os conectores I2C/UART?

R:

- Conector direito (I2C0): GPIO0 (SDA), GPIO1 (SCL), 3V3 e GND
- Conector esquerdo (I2C1): GPIO2 (SDA), GPIO3 (SCL), 3V3 e GND
Ambos podem ser usados para UART.
- Para conferir exemplos de código acesse:
- <https://github.com/BitDogLab/BitDogLab/blob/main/software/I2C/teste%20e%20Scam%20de%20I2C%20V2C%20mostrando%20no%20OLED%20%20funcionando.py>

P: A BitDog Lab possui algum tipo de comunicação sem fio ?

R: Depende, se a placa utilizada for a Raspberry pico W ou pico 2 W, ela possui comunicação Wifi.

Se ela tiver o Shield LoRa, além da comunicação Wifi, irá possuir a comunicação LoRa- Comunicação de Longo Alcance.

Em nosso repositório temos um exemplo de código com comunicação Wifi e LoRa.

Wifi:

<https://github.com/BitDogLab/BitDogLab/tree/main/software/http>

LoRa:

https://github.com/BitDogLab/BitDogLab/tree/main/software/LoRa_Shield

Programação com MicroPython

P: Quais bibliotecas devo importar?

R:

```
from machine import Pin, PWM, SoftI2C, ADC
from ssd1306 import SSD1306_I2C
import neopixel, utime, math, random
```

P: Como inicializar o OLED com SoftI2C?

R:

```
i2c = SoftI2C(scl=Pin(15), sda=Pin(14))
oled = SSD1306_I2C(128, 64, i2c)
```

P: Como configurar a matriz de LEDs?

R:

```
NUM_LEDS = 25
np = neopixel.NeoPixel(Pin(7), NUM_LEDS)
LED_MATRIX = [
    [24, 23, 22, 21, 20],
    [15, 16, 17, 18, 19],
    [14, 13, 12, 11, 10],
    [5, 6, 7, 8, 9],
    [4, 3, 2, 1, 0]
]
```

P: Qual é a diferença entre time e utime?

R: A biblioteca utime é otimizada para dispositivos embarcados como a Raspberry Pi Pico. Recomenda-se usá-la no lugar de time.

Aplicabilidade Educacional

P: Quem pode usar a BitDogLab?

R: Estudantes do ensino fundamental II, médio, e também universitários iniciantes. Ideal para escolas, clubes de ciência e espaços maker.

P: Que tipo de projetos posso desenvolver?

R: Desde projetos simples com LEDs e botões até sistemas com sensores, motores, comunicação sem fio e automação.

P: Posso usar IA para me ajudar a programar a BitDogLab?

R: Sim! A BitDogLab foi pensada para ser usada com assistentes de IA como o ChatGPT. Isso facilita o aprendizado e a resolução de dúvidas de forma interativa.

P: O que é a BNCC ?

R: A Base Nacional Comum Curricular (BNCC) é um documento de

caráter normativo que define o conjunto orgânico e progressivo de

aprendizagens essenciais que todos os alunos devem desenvolver

ao longo das etapas e modalidades da Educação Básica, de modo

a que tenham assegurados seus direitos de aprendizagem e desenvolvimento, em conformidade com o que preceitua o Plano Nacional de Educação (PNE). Este documento normativo aplica-se exclusivamente à educação escolar, tal como a define o § 1o do Artigo 1o da Lei de Diretrizes e Bases da Educação Nacional (LDB, Lei no 9.394/1996)¹ e está orientado pelos princípios éticos, políticos e estéticos que

visam à formação humana integral e à construção de uma sociedade justa, democrática e inclusiva, como fundamentado nas Diretrizes Curriculares Nacionais da Educação Básica (DCN)2.

P: Quais as principais competências de pensamento computacional da BNCC para a educação infantil?

R: EI03CO01: Reconhecer padrão de repetição em sequência de sons, movimentos, desenhos.

- EI03CO02: Expressar as etapas para a realização de uma tarefa de forma clara e ordenada.
- EI03CO03: Experimentar a execução de algoritmos brincando com objetos (des)plugados.
- EI03CO04: Criar e representar algoritmos para resolver problemas.
- EI03CO05: Comparar soluções algorítmicas para resolver um mesmo problema.
- EI03CO06: Compreender decisões em dois estados (verdadeiro ou falso).

Exemplos de Projetos:

- Montar sequências com blocos coloridos para reconhecer padrões.
- Criar rotinas com imagens (ex: escovar os dentes, vestir o pijama).
- Simular percursos no chão com comandos simples (algoritmos desplugados).

P: Quais as principais competências de pensamento computacional da BNCC para a 1º ano do ensino fundamental?

R: EF01CO01: Organizar objetos considerando padrões.

- EF01CO02: Identificar e seguir passos no dia a dia para resolver problemas.
- EF01CO03: Reorganizar e criar sequências de passos (algoritmos).

Exemplos de Projetos:

- Agrupar figuras por formas ou cores (organização por padrão).
- Criar e seguir instruções para dobradura de papel (algoritmos).
- Descrever um jogo usando desenhos em sequência.

P: Quais as principais competências de pensamento computacional da BNCC para a 2º ano do ensino fundamental?

R: EF02CO01: Criar e comparar modelos (representações) de objetos.

- EF02CO02: Criar e simular algoritmos com repetições simples.

Exemplos de Projetos:

- Classificar meios de transporte por atributos (modelagem).
- Criar instruções com repetições simples para um boneco andar em um tabuleiro.
- Identificar comandos específicos usados por brinquedos eletrônicos.

P: Quais as principais competências de pensamento computacional da BNCC para a 3ª ano do ensino fundamental?

R: EF03CO01: Associar valores 'verdadeiro' e 'falso' a sentenças lógicas.

- EF03CO02: Criar algoritmos com repetições condicionais simples.
- EF03CO03: Aplicar decomposição para resolver problemas.

Exemplos de Projetos:

- Criar histórias com perguntas “verdadeiro ou falso”.
- Simular jogos de labirinto com condições do tipo “enquanto não encontrar a saída”.
- Dividir tarefas como “fazer um lanche” em partes e montar os passos.

P: Quais as principais competências de pensamento computacional da BNCC para a 4ª ano do ensino fundamental?

R: EF04CO01: Representar objetos por matrizes.

- EF04CO02: Representar objetos por registros.
- EF04CO03: Criar algoritmos com repetições simples e aninhadas.

Exemplos de Projetos:

- Criar um tabuleiro de batalha naval para trabalhar com matrizes e coordenadas.
- Desenvolver um formulário de cadastro de personagens com campos variados (registro).
- Simular lavagem de janelas em prédios com repetições aninhadas (algoritmo com loops dentro de loops).

Exemplos de Projetos:

- Criar um tabuleiro de batalha naval para trabalhar com matrizes e coordenadas.
- Desenvolver um formulário de cadastro de personagens com campos variados (registro).
- Simular lavagem de janelas em prédios com repetições aninhadas.

P: Quais as principais competências de pensamento computacional da BNCC para a 5ª ano do ensino fundamental?

R: • EF15CO01: Identificar formas de organizar e representar a informação.

- EF15CO02: Construir algoritmos com sequências, seleções e repetições.
- EF15CO03: Operar com lógica computacional (not, and, or).
- EF15CO04: Aplicar decomposição em problemas.

Exemplos de Projetos:

- Organizar fila de alunos como lista dinâmica com inserções e remoções.
- Criar grafos representando amigos na turma ou trajetos no bairro.
- Escrever algoritmos com decisões condicionais (ex: “se chover, leve guarda-chuva”).

P: Quais as principais competências de pensamento computacional da BNCC para a 6º ano do ensino fundamental?

R: Continua os objetivos do 5º ano, com foco no uso mais elaborado de algoritmos, lógica e abstração.

Exemplos de Projetos:

- Desenvolver algoritmos com variáveis e estruturas mais elaboradas.
- Criar jogos simples com uso de lógica e estruturas como listas e matrizes.
- Dividir um problema em várias partes e resolver colaborativamente.

P: Quais as principais competências de pensamento computacional da BNCC para a 7º ano do ensino fundamental?

R: Ênfase em programação com registros e matrizes, desenvolvimento de projetos com soluções computacionais.

Exemplos de Projetos:

- Simular um sistema de votação usando programação com registros.
- Criar uma animação ou jogo com variáveis e estruturas de repetição.
- Planejar e construir um algoritmo para resolver um desafio matemático. Ex Projeto Quiz matemático da disciplina IE323 – Tópicos em eletrônica com abordagem Steam

P: Quais as principais competências de pensamento computacional da BNCC para a 8º ano do ensino fundamental?

R: Desenvolvimento de soluções computacionais com programação, lógica e estruturas de dados.

Exemplos de Projetos:

- Desenvolver um chatbot simples com condições e estruturas de dados.
- Criar um sistema de agendamento usando listas ou grafos.
- Simular uma rede de computadores em formato visual.

P: Quais as principais competências de pensamento computacional da BNCC para a 9º ano do ensino fundamental?

R: Consolidação da habilidade em lógica computacional, estruturas de dados e soluções com algoritmos.

Exemplos de Projetos:

- Criar um sistema de recomendação simples com estrutura condicional.
- Projetar uma rede social fictícia usando conceitos de grafos.
- Resolver um problema real da escola com algoritmo colaborativo.

P: Quais as principais competências de pensamento computacional da BNCC para o ensino médio ?

R: Projetos baseados em problemas com uso de computação.

- Automação de processos com responsabilidade e ética. Ex horta inteligente
- Desenvolvimento de soluções computacionais em contextos diversos.

Exemplos de Projetos:

- Criar um app ou sistema que automatize alguma tarefa cotidiana.

Exemplo: Projeto para monitorar os componentes do ar atmosférico

Para mais informações consulte informações no repositório da BitDog Lab:

https://github.com/danielvieira95/Projetos_Disciplina_IE323/tree/main/Projeto3/Grupo2-Projeto3-EfeitoEstufa

- Desenvolver um jogo educativo envolvendo lógica e programação.
- Simular um processo social com uso de algoritmos e estruturas computacionais.

Exemplo robô seguidor de linha.

Para mais informações consulte informações no repositório da BitDog Lab:

https://github.com/danielvieira95/Projetos_Disciplina_IE323/tree/main/Projeto3/Grupo3-BitMovel-Seguidor-de-Linha

Quais são alguns exemplos de projetos que dá pra fazer com a placa BitDog Lab?

Você pode fazer vários projetos legais com a BitDog Lab! Aqui vão alguns exemplos:

- **Quiz Matemático**

Um jogo de perguntas de matemática com botões e display.

- Para mais informações consulte informações no repositório da BitDog Lab:

https://github.com/danielvieira95/Projetos_Disciplina_IE323/tree/main/PROJETO_1/vinicius-edson-matematica

- **Aprendendo elementos químicos**

Mostra os símbolos químicos no display da BitDog Lab.

- Para mais informações consulte informações no repositório da BitDog Lab:


https://github.com/danielvieira95/Projetos_Disciplina_IE323/tree/main/PROJETO_1/Display_OLED

- **Batalha Naval com LEDs**

Jogo interativo usando matriz de LED e botões.

Para mais informações consulte informações no repositório da BitDog Lab:

https://github.com/danielvieira95/Projetos_Disciplina_IE323/tree/main/PROJETO_1/GameDogLab

-  **Robô Explorador**
Robô que se movimenta e evita obstáculos.

https://github.com/danielvieira95/Projetos_Disciplina_IE323/tree/main/Projeto3/BitMovel-Robo-Explorador

P : Quero criar um robô seguidor de linha ?

R: Para criar um robô seguidor de linha com a BitDog Lab temos um projeto de referência no repositório

https://github.com/danielvieira95/Projetos_Disciplina_IE323/tree/main/Projeto3/Grupo3-BitMovel-Seguidor-de-Linha

P: Quero criar um código em micropython para piscar o LED RGB da BitDog Lab?

R: from machine import Pin

import time

Configuração do LED RGB

led_r = Pin(13, Pin.OUT)

led_g = Pin(12, Pin.OUT)

led_b = Pin(11, Pin.OUT)

def set_rgb_color(r, g, b):

 led_r.value(r)

 led_g.value(g)

 led_b.value(b)

Função para testar o LED RGB

def test_rgb():

 print("Red ON")

 set_rgb_color(1, 0, 0)

 time.sleep(2)

```
print("Green ON")
```

```
set_rgb_color(0, 1, 0)
```

```
time.sleep(2)
```

```
print("Blue ON")
```

```
set_rgb_color(0, 0, 1)
```

```
time.sleep(2)
```

```
print("All OFF")
```

```
set_rgb_color(0, 0, 0)
```

```
time.sleep(2)
```

```
# Chama a função de teste
```

```
test_rgb()
```

P: Quero criar um código em micropython para quando o botão A for pressionado o Led azul acende e quando o botão estiver solto o led apaga BitDog Lab?

P: Onde posso encontrar exemplos de código para a BitDog Lab ?

R: É possível encontrar exemplos de códigos para a BitDog Lab em Micropython

<https://github.com/BitDogLab/BitDogLab/tree/main/software>

E em linguagem C

<https://github.com/BitDogLab/BitDogLab-C>

Suporte e Comunidade

P: Onde posso tirar dúvidas ou buscar ajuda?

R: Pelo GitHub oficial <https://github.com/BitDogLab/BitDogLab>, pela comunidade do Projeto Escola 4.0 ou diretamente com o professor Dr Fabiano Fruett projetista da BitDog Lab

P: Posso contribuir com o projeto?

R: Sim! A BitDogLab é um projeto de código aberto. Todos são bem-vindos para sugerir melhorias, desenvolver extensões ou compartilhar experiências de uso.

Glossário BitDogLab

ADC (Conversor Analógico-Digital)

Componente da Raspberry Pi Pico que converte sinais analógicos (como do joystick ou microfone) em valores digitais que podem ser processados pelo programa.

BitDogLab

Placa educacional desenvolvida pela Unicamp no Projeto Escola 4.0, projetada para ensino de programação e eletrônica com hardware aberto baseado na Raspberry Pi Pico.

Botão A / Botão B

Botões físicos conectados aos GPIOs 5 e 6 da Pico. Usam configuração de pull-up, sendo lidos como LOW quando pressionados.

Buzzer

Componente que emite sons ao ser excitado com sinais PWM. A BitDogLab possui dois buzzers conectados aos GPIOs 10 e 21.

GPIO (General Purpose Input/Output)

Pinos da Raspberry Pi Pico usados para ler sinais de entrada ou enviar comandos para componentes externos.

I2C (Inter-Integrated Circuit)

Protocolo de comunicação serial usado para conectar sensores e displays. A BitDogLab utiliza I2C para o display OLED e para expansões externas.

Joystick KY-023

Módulo analógico com dois eixos (VRx e VRy) e botão embutido. Usado para controlar interfaces ou jogos.

LED RGB (Cátodo Comum)

LED com três cores (vermelho, verde e azul) controlado via GPIOs 11, 12 e 13. Pode gerar diversas cores com combinações de PWM.

MicroPython

Versão da linguagem Python otimizada para microcontroladores como a Raspberry Pi Pico.

Microfone de Eletreto

Módulo analógico conectado ao GPIO28, que capta sons do ambiente e converte em sinais analógicos.

Neopixel (WS2812B)

Matriz de LEDs RGB endereçáveis conectada ao GPIO7. Permite controle individual de cada LED em termos de cor e intensidade.

OLED (Display)

Display gráfico de 128x64 pixels que se comunica via I2C (padrão no endereço 0x3C). Usado para exibir textos e gráficos simples.

PWM (Pulse Width Modulation)

Técnica usada para controlar intensidade de luz, som ou velocidade de motores variando a largura dos pulsos digitais.

Raspberry Pi Pico (H ou W)

Microcontrolador baseado no chip RP2040, que serve como “cérebro” da BitDogLab.

SoftI2C

Implementação em software do protocolo I2C. Usada quando a comunicação com dispositivos I2C precisa ser mais robusta.

SPI (Serial Peripheral Interface)

Outro protocolo de comunicação, usado principalmente com módulos de expansões conectados via conector IDC.

utime

Biblioteca otimizada para microcontroladores, usada em substituição à time em MicroPython.