

# Guia Rapido - Matriz de LED 5x5

Chip: WS2812B (WorldSemi) | NeoPixel RGB Inteligente 5050

## 1. O que e esse hardware?

A matriz 5x5 da BitDogLab e composta por 25 LEDs WS2812B. Cada LED e um modulo completo com chip de controle integrado, LEDs R/G/B e driver de corrente tudo num pacote SMD 5050. Todos os 25 LEDs sao controlados por um UNICO pino digital usando protocolo serie NZR (1 fio).

```
Cascata: GPIO7 -> LED[0] -> LED[1] -> ... -> LED[24]
```

Cada LED recebe 24 bits de dados: 8 bits G + 8 bits R + 8 bits B

Ordem dos dados: GRB (atencao: NAO e RGB!)

Resolucao: 256 niveis por canal = 16.777.216 cores possiveis

## 2. Conexao na BitDogLab

Sinal	GPIO	Descricao
DIN (dados)	GPIO7	Sinal de controle NZR para o primeiro LED
VDD	5V	Alimentacao dos LEDs (via placa)
VSS	GND	Terra

O protocolo NZR opera a 800kbps. O RP2040 usa o bloco PIO (I/O programavel) para gerar os pulsos com timing preciso - tudo gerenciado automaticamente pela biblioteca neopixel.

## 3. Mapeamento da matriz 5x5

Os 25 LEDs sao numerados de 0 a 24 em ordem serpentina (snake) ou linear, dependendo do layout fisico da placa. Na BitDogLab o mapeamento e:

```
Linha 0: LED  0  1  2  3  4    (topo)
Linha 1: LED  5  6  7  8  9
Linha 2: LED 10 11 12 13 14
Linha 3: LED 15 16 17 18 19
Linha 4: LED 20 21 22 23 24    (base)
```

Formula: indice = linha \* 5 + coluna

Exemplo: linha=2, coluna=3 -> indice = 2\*5+3 = 13

## 4. Codigo base em MicroPython

```
import neopixel
from machine import Pin
import time

NUM_LEDS = 25
np = neopixel.NeoPixel(Pin(7), NUM_LEDS)

# Acender LED individual (linha, coluna, R, G, B)
```

```
def pixel(linha, col, r, g, b):
    np[linha * 5 + col] = (r, g, b)
    np.write()

# Apagar todos
def limpar():
    np.fill((0, 0, 0))
    np.write()

pixel(0, 0, 255, 0, 0)    # LED [0,0] vermelho
pixel(2, 2, 0, 255, 0)    # LED central verde
np.fill((0, 0, 50))       # todos azul fraco
np.write()
```

## 5. Controle de brilho

---

O brilho é controlado diretamente pelos valores R, G, B (0-255). Para brilho geral, escala os valores. Evite usar brilho 255 em muitos LEDs ao mesmo tempo: cada WS2812B consome até 60mA em brilho máximo (todos brancos =  $25 \times 60\text{mA} = 1.5\text{A}$ !).

```
BRILHO = 30    # 0 a 255 (30 = ~12%, seguro para uso escolar)

def pixel_brilho(linha, col, r, g, b):
    escala = BRILHO / 255
    np[linha*5+col] = (int(r*escala), int(g*escala), int(b*escala))
    np.write()
```

## 6. Características WS2812B

---

Parametro	Valor
Tensão de alimentação	3.5V a 5.3V (VDD)
Corrente por LED (max)	60mA (branco total)
Protocolo	NZP serial, 800 kbps
Dados por pixel	24 bits (GRB, 8 bits cada)
Frequência de varredura	mínimo 400 Hz
Temperatura de operação	-25 a +80 graus C

## 7. Aplicações práticas

---

- Emojis e ícones: definir matriz de pixels com cores.
- Animações: atualizar np e chamar np.write() em cada frame.
- Indicador de direção: acender linha ou coluna conforme joystick.
- Jogo da vida ou Snake: lógica celular com refresh a cada 100-200ms.