# Projeto de Automação com Joystick, OLED e LEDs RGB – BitDogLab

## 1. Introdução

Este projeto tem como objetivo demonstrar o uso do **joystick analógico** e do **display OLED** para controlar dinamicamente os **LEDs RGB integrados na placa BitDogLab**, utilizando **MicroPython**. O sistema simula um **controle de iluminação automatizado**, onde o usuário ajusta cores e intensidades com o joystick, e visualiza em tempo real os valores no display OLED.

---

## 2. Componentes utilizados

| Componente | Função principal | Tipo de sinal |
|---|---|---|
| Joystick analógico | Entrada de controle (direcional) | Analógico e Digital |
| Display OLED 128x64 (I2C) | Exibição de informações | Digital (I2C) |
| LEDs RGB integrados | Saída de iluminação | PWM (saída analógica simulada) |
| Microcontrolador BitDogLab | Processamento e controle | — |

---

## 3. Organização dos pinos da BitDogLab

| Componente | Pino GPIO | Função |
|---|---|---|
| Joystick eixo X | GPIO26 | Entrada analógica (ADC) |
| Joystick eixo Y | GPIO27 | Entrada analógica (ADC) |
| Botão do joystick | GPIO22 | Entrada digital (com PULL-UP) |
| OLED SDA | GPIO14 | Linha de dados I2C |
| OLED SCL | GPIO15 | Linha de clock I2C |
| LED vermelho | GPIO13 | Saída PWM |
| LED verde | GPIO12 | Saída PWM |

| | | |
|---|---|---|
| LED azul | GPIO11 | Saída PWM |

---

## 4. Funcionamento do sistema

O sistema realiza três operações principais em loop contínuo:

1. **Leitura do joystick**
   O joystick é composto por dois potenciômetros (eixo X e Y) e um botão.
   - O **eixo X** gera um valor analógico entre **0 e 65535**, representando o deslocamento horizontal.
   - O **eixo Y** também gera um valor analógico, representando o deslocamento vertical.
   - O **botão** funciona como uma chave digital, retornando 1 (não pressionado) ou 0 (pressionado).
2. **Controle dos LEDs RGB**
   - O valor do eixo **X** é mapeado para o **vermelho (R)**.
   - O valor do eixo **Y** é mapeado para o **verde (G)**.
   - O **azul (B)** é calculado de forma inversa ao eixo X, criando um equilíbrio de cor dinâmico.
   - O botão do joystick alterna o estado geral dos LEDs (ligar/desligar).
3. **Exibição no display OLED**
   O OLED mostra, em tempo real:
   - Os valores dos eixos **X** e **Y** (convertidos para escala de 0–255);
   - O estado atual dos LEDs (**ON/OFF**).

## 5. Explicação do código

- A biblioteca `machine` é usada para acessar os pinos GPIO, ADC e PWM.
- `SoftI2C` permite usar pinos personalizados (GPIO14 e 15) para o display OLED.

- Os eixos X e Y do joystick são lidos via ADC e convertidos em valores PWM (0–65535) para controlar a cor dos LEDs.
- A função `map_value()` converte a faixa do ADC para a faixa de intensidade PWM.
- O botão do joystick alterna o estado dos LEDs com um pequeno atraso (`debounce`).
- O OLED é atualizado constantemente com os valores e o status do sistema.

```
from machine import Pin, ADC, PWM, SoftI2C
import ssd1306
import time

# --- Configuração do OLED ---
i2c = SoftI2C(scl=Pin(15), sda=Pin(14))
oled = ssd1306.SSD1306_I2C(128, 64, i2c)

# --- Joystick ---
adc_x = ADC(Pin(26))
adc_y = ADC(Pin(27))
button = Pin(22, Pin.IN, Pin.PULL_UP)

# --- LEDs RGB da BitDogLab ---
led_r = PWM(Pin(13))
led_g = PWM(Pin(12))
led_b = PWM(Pin(11))
for led in (led_r, led_g, led_b):
    led.freq(1000)

# --- Função utilitária ---
def map_value(val, in_min=0, in_max=65535, out_min=0, out_max=65535):
    return int((val - in_min) * (out_max - out_min) / (in_max - in_min) + out_min)

led_on = True

while True:
    x_val = adc_x.read_u16()
    y_val = adc_y.read_u16()

    if not button.value():
        led_on = not led_on
        time.sleep(0.3)

    if led_on:
        led_r.duty_u16(map_value(x_val))
```

```
    led_g.duty_u16(map_value(y_val))
    led_b.duty_u16(65535 - map_value(x_val))
else:
    led_r.duty_u16(0)
    led_g.duty_u16(0)
    led_b.duty_u16(0)

oled.fill(0)
oled.text(f'X:{map_value(x_val,0,65535,0,255)}', 0, 0)
oled.text(f'Y:{map_value(y_val,0,65535,0,255)}', 0, 10)
oled.text(f'LED: {"ON" if led_on else "OFF"}', 0, 20)
oled.show()

time.sleep(0.05)
```