

# Manual simplificado para o usuário final

## Tecnologia de Camaleão Imitando Cores da Natureza

---

### 1. Introdução

Este projeto simula a capacidade de camuflagem dos camaleões, imitando as cores da natureza mediante um sistema embarcado. Utilizando um sensor de cor TCS34725, o sistema identifica a cor primária RGB de um objeto e reproduz esse cor em uma matriz de LEDs RGB WS2812B (Neopixel). Um display OLED exibe os valores exatos das cores (R, G, B) captada pelo sensor, proporcionando uma interface visual clara e informativa.

O projeto será desenvolvido utilizando a plataforma BitDogLab com o microcontrolador Raspberry Pi Pico, programado em Python no ambiente MicroPython usando o IDE Thonny.

### 2. Objetivo

O objetivo deste sistema é capturar cores RGB primários usando um sensor de cor e reproduzi-los em uma matriz de LEDs RGB. Ao mesmo tempo, os valores RGB exatos são exibidos em um display OLED, promovendo uma experiência interativa e educacional no aprendizado de combinação de cores.

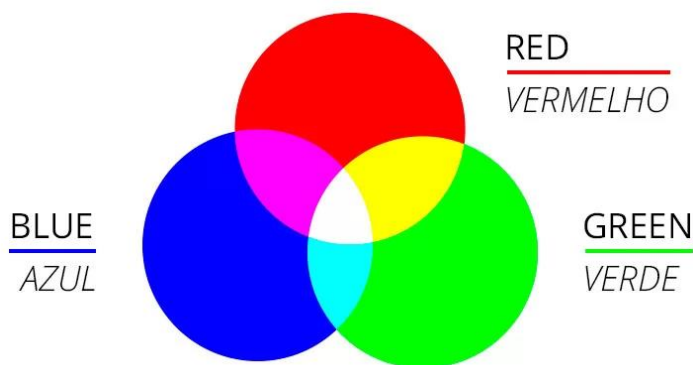


Figura 1: Cores Primárias RGB

---

### 3. Conexões e configurações de hardware

- **Matriz de LEDs WS2812B (Neopixel) :**
  - Pino de entrada (IN) conectado ao **GPIO7** do Raspberry Pi Pico.
- **Tela OLED (128x64) :**
  - Pino **SDA** conectado ao **GPIO14** e **SCL** ao **GPIO15** (I2C1).
- **Sensor de cor TCS34725 :**
  - Pino **SDA** conectado ao **GPIO00** e **SCL** ao **GPIO01** (I2C0).

---

## 4. Componentes Utilizados

- **Raspberry Pi Pico** com MicroPython instalado.
  - **Matriz de LEDs WS2812B (5x5)** para exibir os cores da combinação primária RGB.
  - **Display OLED 128x64** para exibir os valores RGB capturados.
  - **Sensor de cor TCS34725** para capturar cores.
- 

## 5. Instruções de como instalar o jogo

Passo 1: Instalar a IDE Thonny

Baixe e instale o IDE Thonny a partir do site oficial.

Passo 2: Configurar IDE thonny com o MicroPython ( Raspberry PI Pico)

Abra o IDE Thonny e clique no ícone de aonde tem configure interpreter (localizado no lado direito abaixo).

Selecione MicroPython ( Raspberry PI Pico)

Selecione a porta do microcontrolador

Copie o código do repositório pretendido e salve com na opção que vai aparecer Raspberry com nome main.py e faça o mesmo no código da biblioteca, mas como nome de TCS34725.py. Caso já tenha um código salvo, apenas substitua.

Passo 6: Enviar o Código para o plataforma Bitdoglab

No IDE Thonny, digitando o F5 para enviar e correr com programa no bitdoglab ou clique no botão run.

Passo 7: Ver o Código Funcionando

Após enviar o código, o Pico vai rodar o main.py automaticamente. Se tudo estiver certo, programa vai começar.

## 6. Explicação do Código

1. **Importação das Bibliotecas:** O código utiliza bibliotecas como `machine` para controle do hardware, `neopixel` para controlar a matriz de LEDs, `tcs34725` para o sensor de cor e `ssd1306` para o display OLED.

2. **Configuração do OLED e Sensor de Cor:** O OLED está configurado no canal I2C1, enquanto o sensor de cor utiliza o canal I2C0. O sensor de cor é inicializado com tempo de integração e ganho definido.
3. **Funções Principais:**
  - `ler_sensor_cor`: Lê os valores RGB e a clareza (intensidade de luz) capturados pelo sensor.
  - `captura_cor(r, g, b, c)`: Processa e identifica a cor
  - `ajustar_matriz_leds`: Ajusta a matriz de LEDs para reproduzir a cor detectada.
  - `exibir_cor_oled`: Exibe os valores RGB no display OLED.
  - `loop_principal`: Loop contínuo que realiza a leitura dos valores, normaliza os dados e os exibe tanto na matriz de LEDs quanto no OLED.
4. **Dicionário de cores predefinidas do RGB primários** : para garantir que a cor exibida seja precisa, os valores RGB são ajustadas com base na clareza capturada pelo sensor (analisando os valores de R, G e B pelo valor de C).
5. **Loop Infinito:** O programa funciona em um loop infinito, capturando e exibindo as cores continuamente.