

Dispositivo CDC (Communication Device Class) responsivo e com identificação visual:

Criar um Dispositivo CDC (Communication Device Class), que responda aos comandos realizados no computador através de eco e indique visualmente o comando executado. O projeto deve ser implementado com a biblioteca TinyUSB.

Arquivo principal.c (main)

```
#include <stdio.h>
#include <string.h>
#include "pico/stdlib.h"
#include "tusb.h"

// GPIOs para os LEDs
#define LED_VERDE 11
#define LED_AZUL 12
#define LED_VERMELHO 13

void Inicializa_GPIO()
{
    gpio_init(LED_VERDE);
    gpio_set_dir(LED_VERDE, GPIO_OUT);
    gpio_put(LED_VERDE, 0);

    gpio_init(LED_AZUL);
    gpio_set_dir(LED_AZUL, GPIO_OUT);
    gpio_put(LED_AZUL, 0);

    gpio_init(LED_VERMELHO);
    gpio_set_dir(LED_VERMELHO, GPIO_OUT);
    gpio_put(LED_VERMELHO, 0);
}

void acender_led(int gpio)
{
    gpio_put(gpio, 1);
    sleep_ms(2000);
    gpio_put(gpio, 0);
}

int main()
{
    // Inicializa o USB
    stdio_init_all();
    Inicializa_GPIO();

    // Aguarda o USB ser montado e Verifica se o host (PC) conectou-se ao dispositivo CDC
    while (!tud_cdc_connected())
    {
        sleep_ms(2000);
    }
    printf("USB conectado!\n"); // Informa via terminal serial que a conexão foi detectada

    // Loop principal: ecoa o que receber
    while (true)
    {
        if (tud_cdc_available())
        { // Verifica se há dados disponíveis vindos do host (PC)

            uint8_t buf[12]; // Declara um buffer de 64 bytes
            int count = tud_cdc_read(buf, sizeof(buf));
            buf[count] = '\0'; // Finaliza a string recebida

            if (strcmp(buf, "verde") == 0)
            {
```

```

        acender_led(LED_VERDE); // Acende o LED verde
    }
    else if (strcmp(buf, "azul") == 0)
    {
        acender_led(LED_AZUL); // Acende o LED azul
    }
    else if (strcmp(buf, "vermelho") == 0)
    {
        acender_led(LED_VERMELHO); // Acende o LED vermelho
    }
    else
    {
        printf("Comando nao reconhecido\n");
    }
    printf("Palavra recebido: %s\n", buf);
    printf("\n");
    tud_cdc_write(buf, count); // Escreve os mesmos dados de volta ao host, efetivamente
fazendo um eco
    tud_cdc_write_flush();

}

tud_task(); // Executa tarefas USB
}

return 0;
}

```