# Roteiro de Experimento: Recepção UART com Controle de LED na BitDogLab

## 1. Título do Experimento

Recepção de Dados via UART USB e Controle de LED (Projeto: isr\_uart\_bitdoglab)

## 2. Objetivo

- Compreender o funcionamento da recepção de dados via UART no RP2040.  
- Aprender a reagir a eventos de comunicação serial recebendo caracteres.  
- Controlar um LED de forma interativa com base na entrada serial.

## 3. Materiais Necessários

- Placa BitDogLab.  
- LED integrado no GPIO12.  
- Computador com acesso a terminal serial USB (recomendado Putty, Tera Term, CoolTerm, etc.).

## 4. Observação Importante

• O terminal serial nativo do VSCode não funciona bem para recepção de dados USB CDC do Raspberry Pi Pico.  
• Recomenda-se o uso de terminais dedicados como Putty, Tera Term, CoolTerm ou similares para comunicação estável.

## 5. Diagrama de Montagem

• LED conectado ao GPIO12.  
• (A BitDogLab já possui o LED integrado, sem necessidade de montagem adicional.)

## 6. Procedimento

### 6.1. Inicialização do projeto

• Criar um novo projeto no VSCode com suporte ao Pico SDK.  
• Nome sugerido: isr\_uart\_bitdoglab.

### 6.2. Código Fonte Básico

#include <stdio.h>  
#include "pico/stdlib.h"  
  
#define LED\_VERMELHO 13  
  
int main() {  
 stdio\_init\_all();  
 sleep\_ms(2000);  
 printf(" Pico pronto. Digite algo no terminal USB...\n");  
  
 gpio\_init(LED\_VERMELHO);  
 gpio\_set\_dir(LED\_VERMELHO, GPIO\_OUT);  
 gpio\_put(LED\_VERMELHO, 0);  
  
 while (true) {  
 int ch = getchar();  
 char c = (char)ch;  
  
 printf(" Recebido: '%c' (ASCII: %d)\n", c, ch);  
  
 gpio\_put(LED\_VERMELHO, 0);  
 sleep\_ms(200);  
 gpio\_put(LED\_VERMELHO, 1);  
 }  
}

## 7. Análises e Observações

- O sistema aguarda o recebimento de um caractere pela porta USB Serial.  
- Quando recebe:  
 • Mostra o caractere e seu valor ASCII.  
 • Pisca o LED Vermelho por 200 ms.  
- Discuta:  
 • O que acontece se pressionar várias teclas rapidamente?  
 • Como modificar o comportamento dependendo da tecla recebida?

## 8. Extensões e Desafios

- Fazer o LED piscar em diferentes padrões dependendo da letra recebida.  
- Contar quantos caracteres já foram recebidos.  
- Implementar comandos específicos como "ON" para acender e "OFF" para apagar o LED.  
- Implementar recepção não bloqueante usando uart\_is\_readable().

## 9. Importante

- getchar() é uma função bloqueante.  
- sleep\_ms(2000) é necessário para estabilizar a conexão USB.  
- Sempre iniciar o terminal serial após o Pico ser detectado no computador.  
- Recomenda-se utilizar terminais externos (Putty, Tera Term) para melhor funcionamento.