

Lista de Exercícios Unidade 4 - Capítulos 5 a 8

Tarefa 1:

Reutilize o circuito no Wokwi criado na sala de aula e escreva o código. Abaixo está o lembrete das GPIOs do kit BitDogLab.

	Componentes	GPIO
1	Led RGB	Vm - GPIO13 Az - GPIO12 Vd - GPIO11
2	Comunicação SPI	Tx - GP19 Rx - GPIO16 CSn - GPIO17 SCK - GP18
3	Display OLED	SDA - GPIO14 SCL - GPIO15
4	Botões	GPIO5 e GPIO6
5	Buzzer	GPIO10 e GPIO21
6	Matriz de Leds Coloridos	GPIO7, 5 linhas por 5 colunas
7	Microfone	GP28
8	Joystick	Vry - GPIO26 Vrx - GPIO27 Sw - GPIO22

 Link wokwi: <https://wokwi.com/projects/420360925119773697>

Tarefa 2:

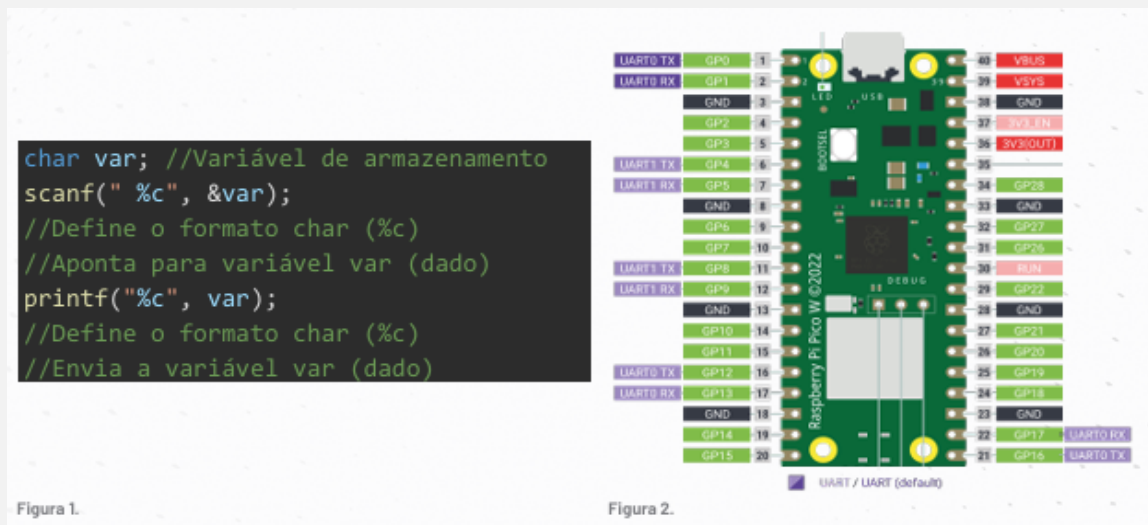
- 1) Elabore um programa para acionar um LED quando o botão A for pressionado 5 vezes, utilizando o temporizador como contador. Quando o valor da contagem atingir 5 vezes, um LED deve piscar por 10 segundos na frequência de 10 Hz.

 Link wokwi: <https://wokwi.com/projects/420992899799003137>

- 2) Na questão anterior, implemente o botão B, para mudar a frequência do LED para 1 Hz.

🔗 Link wokwi: <https://wokwi.com/projects/420992814076859393>

- 3) Elabore um código utilizando a interfaces UART0 e conecte os fios TX e RX atribuídos à essa interface. Essa estrutura envia dados e recebe os dados na mesma interface, apenas para verificar seu funcionamento. Utilize a função scanf da biblioteca stdio para enviar via console um dado à placa, em seguida, transmita da UART0 para a UART1, e por fim, transmita o dado recebido para o console utilizando o printf.



🔗 Link wokwi: <https://wokwi.com/projects/421995690218356737>

- 4) Já para a comunicação I2C, iremos utilizar o DS1307, que é um Real Time Clock - RTC disponível no simulador Wokwi. O endereço I2C do DS1307 é 0x68. Um RTC é um hardware que garante a contagem de tempo na unidade de segundos. Muitos microcontroladores possuem RTC internos, mas alguns fazem uso de hardware externos. Para ler os valores, é necessário inicialmente configurar um valor de data e hora que deve, por exemplo, ser configurado manualmente pelo usuário. Nessa questão você deverá configurar o RTC para 24/09/2024 - 13:27:00 e em seguida, realizar a leitura do mesmo a cada 5 segundos, e imprimir na tela do console (Serial USB) o valor lido. Na tabela a seguir são apresentados os principais endereços do RTC DS1307.

Endereço	Registro	Função	Formato	Acesso
0x00	Segundos	Armazena o valor dos segundos e o bit de Clock Halt	CH + BCD	Leitura/ Escrita
0x01	Minutos	Armazena o valor dos minutos	BCD	Leitura/ Escrita
0x02	Horas	Armazena o valor das horas e o formato 12/24 horas	12/24h + BCD	Leitura/ Escrita
0x03	Dia da Semana	Armazena o valor do dia da semana (1-7)	BCD	Leitura/ Escrita
0x04	Data	Armazena o valor do dia do mês	BCD	Leitura/ Escrita
0x05	Mês	Armazena o valor do mês	BCD	Leitura/ Escrita
0x06	Ano	Armazena o valor do ano (00-99)	BCD	Leitura/ Escrita
0x07	Controle	Controle da saída de frequência SQW/OUT	Bit Flags	Leitura/ Escrita
0x08 a 0x3F	RAM	56 bytes de RAM SRAM para armazenamento de dados adicionais	N/A	Leitura/ Escrita

🔗 Link wokwi: <https://wokwi.com/projects/422065902136821761>

- 5) Modifique o exemplo de código apresentado na videoaula (reproduzido abaixo) para controlar os três LEDs RGB da placa BitDogLab usando o módulo PWM e interrupções, seguindo as orientações a seguir:
- A. O LED vermelho deve ser acionado com um PWM de 1kHz.
 - B. O duty cycle deve ser iniciado em 5% e atualizado a cada 2 segundos em incrementos de 5%. Quando atingir o valor máximo, deve retornar a 5%.
 - O LED azul deve ser acionado com um PWM de 10kHz.

🔗 Link wokwi: <https://wokwi.com/projects/422178067379234817>

- 6) Refaça o programa prático 01 presente no Ebook do Capítulo de ADC, mude a unidade de medida da temperatura de celsius para fahrenheit.

Enunciado:

Exemplo 00: Esse programa realiza uma leitura contínua da temperatura interna do microcontrolador RP2040 e imprime o valor convertido em graus Celsius no terminal, uma vez por segundo. É um exemplo simples e eficaz de como utilizar o ADC para capturar dados analógicos e interpretá-los em um sistema embarcado.

Link wokwi: <https://wokwi.com/projects/422184440515522561>

7) Como o ADC converte sinais analógicos do joystick em valores digitais no exemplo 02?

Enunciado:

Exemplo 02: Este código realiza a leitura dos valores analógicos de um joystick conectado aos pinos GP26 e GP27 de um Raspberry Pi Pico, além de ler o estado de um botão conectado ao pino GP22. O código utiliza o ADC (Conversor Analógico-Digital) do microcontrolador para converter os sinais analógicos do joystick em valores digitais, que são então exibidos na interface serial.

O joystick gera uma tensão variável de saída, entre 0 e 3.3V, de acordo com a posição (eixo x e y). Em seguida o ADC converte essa tensão em um valor digital correspondente, pois possui uma resolução de 12 bits. Isso significa que ele divide a tensão em $2^{12} = 4096$ níveis discretos, onde cada nível do ADC representa aproximadamente 0.8mV, pois é a menor variação detectável pelo ADC ($3.3 \text{ V} / 4096 \approx 0.8 \text{ mV}$).

O microcontrolador RP2040 opera com uma tensão de referência padrão de 3.3V, por ser sua tensão de alimentação. Então todos os periféricos internos, incluindo o ADC, são projetados para trabalhar dentro desse limite.

Durante esse processo, ocorrem três etapas:

- Amostragem: onde o conversor digital-analógico lê a tensão no pino analógico do joystick;
- Quantização: onde divide a faixa de 0 a 3,3V em 4096 níveis discretos;
- Conversão: onde a tensão é transformada em um valor digital correspondente, segundo fórmula:

$$\text{Valor ADC} = \frac{V_{IN}}{V_{REF}} \times (2^{12} - 1)$$

Ao final disso, o código lê o valor digital, e não o analógico (voltagem), sendo:

0V → Valor digital = 0 (mínimo)

3.3V → Valor digital = 4095 (máximo)