

EMBARCATECH - Instituto Hardware BR

Tarefa 1 - Unidade 4 - Cap. 5

Estudante: Victor Orfeu Merlo

- 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 ser piscar por 10 segundos na frequência de 10 Hz.**

A função `main` inicializa o GPIO do LED e do botão, e inicializa um timer periódico de 100ms. Esse timer periódico chama uma função que verifica se o botão foi pressionado, realizando debounce, e aumenta um contador caso seja.

Ao atingir 5 pressionamentos do botão, o contador é zerado, o código indica que o LED pode ser ativado e inicia outro timer de 10 segundos. Ao finalizar esse timer de 10 segundos, o código indica que o LED deve ser desativado.

Enquanto o LED pode ser ativado, a função do timer de 100ms alterna o LED entre aceso e apagado. Como o período de execução dessa função é 100ms, ela tem uma frequência de execução de 10Hz, portanto o LED já pisca a 10Hz.

Link do projeto no Wokwi:

<https://wokwi.com/projects/419696126502032385>

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

Como não estamos usando interrupções, no timer periódico se verifica os dois botões. As informações entre eles não se misturam pois cada botão é designado a um struct dedicado.

Como esse timer periódico está executando a 10Hz, se ele apenas trocar o estado do LED a cada 10 vezes que executa, tem-se que o LED pisca a 1Hz. Para isso, foi implementado um contador que faz o LED trocar de estado quando ele chega a um limite.

Esse limite é controlado pelo botão B. Ao pressionar esse botão, o código alterna entre o limite ser 1 (que representa a frequência de 10Hz) e 10 (que representa a frequência de 1Hz).

Link do projeto no Wokwi:

<https://wokwi.com/projects/419803681837565953>

- 3 **Elabore um código utilizando a interfaces UART0 e conecte os fios TX e RX atribuídos à essa interface entre. 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`.**

Para a transmissão UART, os pinos devem estar conectados. O pino TX de um deve se conectar ao pino RX do outro e vice-versa.

Com a parte física pronta, foi configurado os pinos para as respectivas interfaces UART0 e UART1, assim como a velocidade de transmissão (*baudrate*) que deve ser a mesma nos dois.

Como o enunciado pede, um caracter é lido através de `scanf`, transmitido da UART0 para a UART1 e mostrado no terminal através de `printf`.

Link do projeto no Wokwi:

<https://wokwi.com/projects/419807034311476225>

- 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.**

Para comunicar com o RTC, foram criadas 2 funções para leitura e escrita dos registradores de forma simplificada, onde só é necessário indicar o endereço do registrador e o dado a ser escrito ou lido.

Na função `main`, primeiro é inicializado o I2C do RPi Pico, em seguida os valores dos registradores do RTC, de acordo com o enunciado. Após isso, inicia-se um timer que se repete a cada 5 segundos.

Na função desse timer, é feita a leitura dos registradores de data e hora do RTC, que são mostrados no terminal.

Foi observado um problema na simulação em que a contagem de tempo do RTC não é consistente. Enquanto a função do timer é chamada a cada 5 segundos de simulação, o valor lido do RTC avança mais tempo que isso.

Link do projeto no Wokwi:

<https://wokwi.com/projects/420160020961746945>

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:

- O LED vermelho deve ser acionado com um PWM de 1kHz.
- 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.

Para configurar o PWM com as frequências desejadas, precisamos configurar o divisor do clock e o contador do PWM.

O divisor do clock é um valor fracionário, com 8 bits de parte inteira e 4 bits de parte fracionária, podendo ter um valor máximo de 255,9375. O contador é de 16 bits, podendo ter um valor máximo de 65535.

Para atingirmos a frequência de 1kHz no PWM do LED vermelho, podemos usar:

$$f = \frac{f_c}{\text{divisor} \cdot \text{counter}} = \frac{125.000.000}{125 \cdot 1000} = 1.000 = 1 \text{ kHz}$$

De forma análoga, para chegarmos na frequência de 10kHz no LED azul:

$$f = \frac{f_c}{\text{divisor} \cdot \text{counter}} = \frac{125.000.000}{125 \cdot 100} = 10.000 = 10 \text{ kHz}$$

Esses valores são utilizados na função `main` para configurar o `slice` de cada PWM.

Por fim, no loop `while`, o duty cycle do LED vermelho é incrementado em 5% a cada 2 segundos, retornando a 5% quando chega em 100%.

No entanto, não é possível ver muito claramente a mudança de intensidade do LED vermelho na simulação.

Link do projeto no Wokwi:

<https://wokwi.com/projects/420171949079344129>

6 Refaça o programa pratico 01 presente no Ebook do Capítulo de ADC, mude a unidade de medida da temperatura de celsius para Fahrenheit.

A conversão de Celsius para Fahrenheit se dá pela fórmula:

$$F = C \cdot \frac{9}{5} + 32$$

Como o datasheet do Raspberry Pi Pico já nos mostra a conversão do valor do ADC em bits para graus Celsius, basta aplicar essa fórmula após a conversão.

Link do projeto no Wokwi:

<https://wokwi.com/projects/420179904224796673>