

Exercícios Desenvolvidos no Wokwi

Tarefa 1 - Unidade 4 - Cap. 5



Autor:

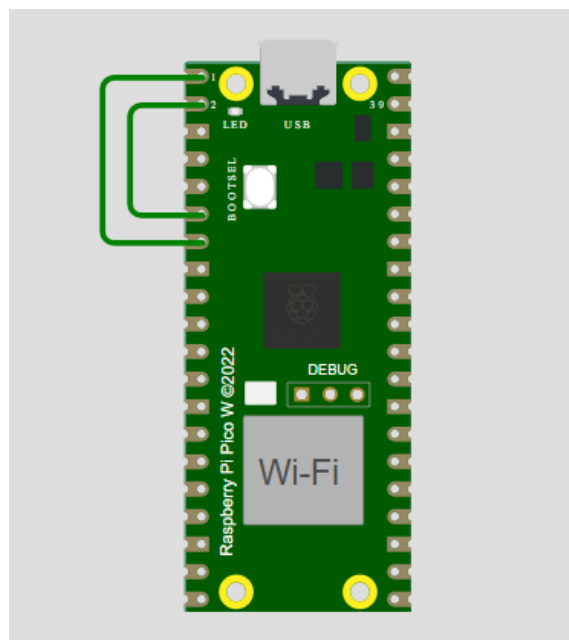
Vinícius Esperança Mantovani

- Link para o projeto <https://wokwi.com/projects/420062902018386945>.

- Link para o projeto <https://wokwi.com/projects/420638928141062145>.

Este exercício foi feito utilizando partes do código exposto na “Figura 4 – Exemplo 2 de código em C/C++”, no e-book do capítulo 6 da unidade 4 em “4.2. Configuração Básica da UART no RP2040” (<https://ead.hardware.org.br/mod/resource/view.php?id=307>). Isso se deu de maneira que o código foi alterado para que a placa recebesse por “scanf” um dado (pelo UART0) e, em seguida, o

enviasse pelo TX do UART0 para o RX do UART1, o qual, após alterações no código base passou a gerar interrupções para recepção de dados em RX (ao invés do UART0 que era para o qual havia interrupção tratada por “irq” do código de exemplo). Finalmente, após a recepção no UART1, na rotina de tratamento da interrupção, o dado é salvo em uma variável global “c” do tipo “char” que é, por fim, impressa no terminal por um “printf”. Vale destacar que a interrupção para recepção de dados no RX do UART1 teve de ser desabilitada antes das impressões “printf” do programa e reabilitada depois delas, uma vez que elas estão atreladas ao TX do UART0, o que ocasionaria a reimpressão subsequente, pois o UART1 receberia a mensagem logo que fosse impressa e já a reenviaria para reimpressão, gerando um “loop” infinito de recepção e escrita da mensagem. Além disso, é importante notar que mais de um caractere pode ser escrito, mas esses serão lidos e passados de volta para o terminal individualmente em diferentes iterações do “while(true)” do programa.

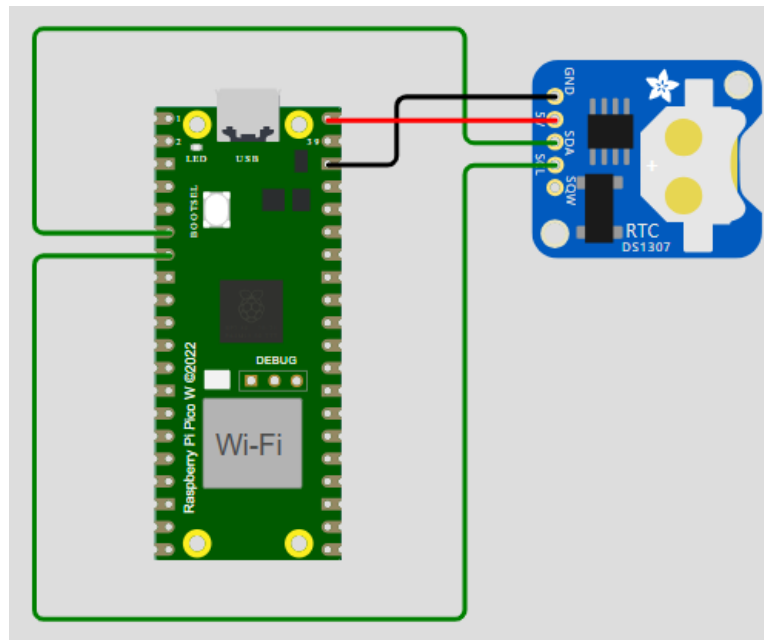


4. Já para a comunicação I2C, utilize o DS1307, que é um Real Time Clock – RTC – disponível no simulador Wokwi. O endereço I2C do DS1307 é 0x68. Um RTC é um hardware que garante a contagem de tempo na unidade de segundos. Muitos microcontroladores possuem RTC internos, mas alguns fazem uso de hardwares externos. Para ler os valores, é necessário inicialmente configurar um valor de data e hora que deve, por exemplo, ser inserido 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, imprimindo na tela do console (Serial USB) o valor lido. Na tabela a seguir, são apresentados os principais endereços do RTC DS1307.

Link para o projeto <https://wokwi.com/projects/420179370888537089>.

Neste exercício, por sua natureza mais afastada dos exemplos expostos, embora tenha sido inspirado em tais exemplos (do e-book do Capítulo 6 - Unidade 4), o código foi escrito do zero. Para tanto, não foi utilizada interrupção alguma, mas sim foi implementada uma lógica iterativa no “while(true)” do programa, de modo que a placa lê de cinco em cinco segundos o tempo presente nos registradores do RTC externo. Desse modo, passando por uma função que organiza o formato dos dados de tempo de cada registrador, esses dados recebidos por I²C do RTC são, finalmente, impressos

no terminal com a função “printf” no formato adequado (por exemplo, inicialmente: Terça-feira, 24/09/2024 - 13:27:00).



5. Modifique o exemplo de código apresentado na videoaula (reproduzido abaixo) para controlar os LEDs RGB da placa BitDogLab usando o módulo PWM e interrupções, seguindo as orientações a seguir:

link para projeto: <https://wokwi.com/projects/420347014753015809>.

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

O LED em questão está usando um PWM de “wrap” 2000, um divisor de 62,5 e um “duty cycle” inicial de 100 → $125\text{Mhz} / 2001 * 62,5 = 999,500 \cong 1\text{Khz}$.

2. O LED verde deve ser acionado com um PWM de 10kHz.

O LED em questão está usando um PWM de “wrap” 2000, um divisor de 6,25 e um “duty cycle” inicial de 100 → $125\text{Mhz} / 2001 * 6,25 = 9995,00 \cong 10\text{Khz}$.

3. 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% e continuar a incrementando. - Fazer isso para ambos os LEDs: azul e vermelho.

Conforme requisitado, o “duty cycle” inicial é de 5% do “wrap” = 100 para os três LEDs e são incrementados em 5% a cada 2 segundos. Além disso, a lógica feita dita que ao chegar a 100%, o “duty cycle” volta a ser 5% e continua sendo incrementado gradativamente.

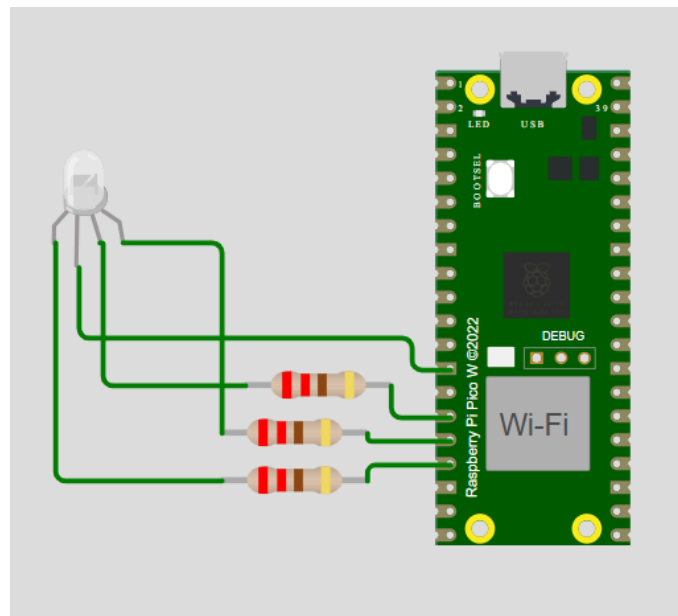
4. Tente controlar frequência e o Duty Cycle do LED azul de forma independente do que fez nos LEDs vermelho e verde. Você consegue? Por que não?

Não é possível fazer isso, uma vez que os LEDs azul e vermelho estão ligados a saídas de PWM contidas em um mesmo “slice”, o que implica que as frequências para ambas essas saídas será invariavelmente a mesma. Ainda assim, é possível controlar esses parâmetros do PWM de maneira independente para o LED verde e o “duty cycle” de cada um dos três LEDs pode ser alterado

individualmente de qualquer maneira, uma vez que há um registrador de nível de saída para cada um dos dois canais PWM de um mesmo “slice”.

O CÓDIGO:

Neste último exercício, foi definida a “irq” para tratamento dos três PWMs que é chamada a cada “wrap” das saídas PWM, de modo que o “duty cycle” dos três LEDs é alterado nessa “irq”. Isso se dá, por meio de uma variável global, “time_count”, do tipo “absolute_time_t”, usada para armazenar o tempo absoluto, inicialmente, do momento de início do programa. Assim, tal variável é usada como comparação com o tempo atual para determinar se já se passaram dois segundos e, conseqüentemente, garantir que após este tempo os LEDs sejam atualizados (seus “duty cycles” sejam alterados).



6. Refaça o Programa Prático 00 presente no Ebook do Capítulo de ADC (Capítulo 8), mas implementando no código a conversão da unidade de medida da temperatura de Celsius para Fahrenheit.

link para projeto: <https://wokwi.com/projects/421063321667959809>.

Neste exercício, foi feito diretamente o que foi pedido. Para além da impressão da temperatura em Celsius, foi calculada a temperatura em Fahrenheit somando 32 ao produto do valor em Celsius por 1,8. Por fim, foi usado “printf” para imprimir ambos os valores no terminal.

