



# Lista de Exercícios #3

## Pado Labs - Microcontroladores

### Estruturas em C e Interrupções

*Tips and Tricks* : Utilizar o *User Manual UM2324* e o *User Manual UM2319* para resolver as questões.

*Requirements* : Resolva pelo menos **4 exercícios**. Exercícios com a tag **Challenge** valem por dois exercícios.

*Requirements* : Exercícios que requerem desenvolvimento de um código deve ser separado adicionado ao *Github*.

**1:** Qual a principal vantagem em utilizar uma interrupção ao invés de checar periodicamente o status de uma GPIO, por exemplo.

R: Por que a interrupção vai fazer com que o programa interrompa e te avise se estiver algo que não foi pré-definido pelo programa a ser executado. GPI não teria rapidez no envio de informações e eficiência ao usuário.

**2:** Por que não é recomendável executar operações custosas e longas dentro de uma rotina ISR?

R: Pois o programa processa as interrupções com prioridade acima do programa normal, não podendo ter mais do limite para que o processamento seja rápido.

**3:** Desenvolva um programa que altere o status do LD4 a cada toque no botão USER do kit, utilizando a interrupção.

**4:** Desenvolva um programa que conta quantas vezes o botão de USER foi pressionado, utilizando interrupção.

**5:** Aproveite o programa anterior, e desenvolva uma aplicação que além de contar a quantidade de vezes que foi pressionado, exiba em um display essa quantidade de 0 a 9 (ao chegar em 9 volte a zero). Utilize também um *enum* para indicar se o botão está pressionado e também uma *struct* para armazenar o contador e o status do botão.

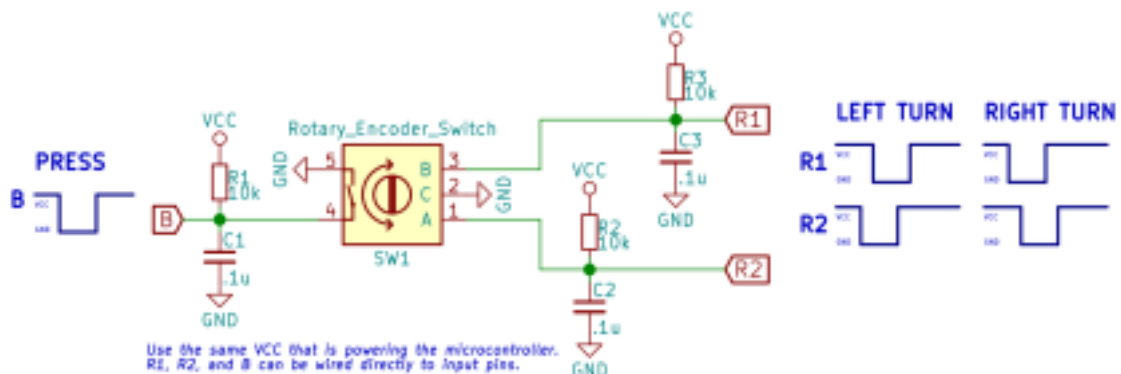
**6: Challenge:** Monte na protoboard 4 LEDs e 4 botões. Cada botão deve ser

associado a um LED, e cada toque no botão deve alterar a frequência de piscadas do LED associado. A frequência deve ser de 10Hz, 5Hz e 1Hz. Utilize uma *struct* e um *enumerate* para indicar o status do LED e quantas vezes o botão foi pressionado. Lembre-se que você pode utilizar arrays dentro de uma *struct*, e *structs* dentro de outra.

1

**7: Challenge:** O *Rotary Encoder*, conhecido também como potenciômetro infinito, é muito utilizado para controle em sons de carros e outros dispositivos. A grande vantagem desse componente é que podemos fazer uma alteração em parâmetros de diversas fontes sem causar algum impacto no potenciômetro. Por exemplo, um equipamento de som que possui um controle de volume rotativo e remotamente através de algum tipo de conexão USB. Com isto, elabore um programa que leia um *rotary encoder* e acenda um LED ao girar para o sentido horário e outro ao girar ao sentido anti-horário. O esquema de montagem para o dispositivo está ilustrado na figura 1.

Figura 1: Circuito para montagem de um *rotary encoder* com chave integrada.



Fonte: Google Imagens.

