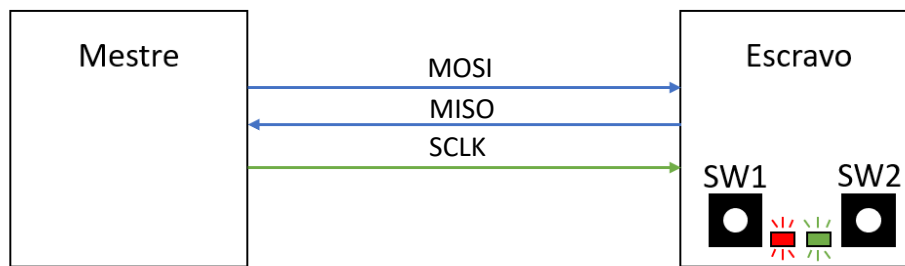


# Desafio 2

## Contexto

Vamos programar um escravo SPI que informa o estado dos botões para o mestre e ao mesmo tempo recebe do mestre comandos para controlar a luminosidade dos seus LEDs. Usando apenas uma LaunchPad, as conexões devem ser feitas entre duas interfaces dedicadas. Em caso de pino queimado ou não haver pinos disponíveis (caso da FR5994), o mestre pode ser implementado por bit-banging.



O escravo recebe comandos da seguinte forma:

Comando							
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0	g	g	g	0	r	r	r

O nibble mais significativo do comando é a intensidade do LED vermelho e o nibble menos significativo é a intensidade do LED verde. A intensidade só pode assumir 5 valores possíveis:

Valor do nibble	Intensidade do LED
0	LED totalmente apagado
1	25%
2	50%
3	75%
4	LED completamente aceso

Toda vez que o mestre quiser fazer uma leitura do escravo, deve enviar o byte 0xFF para receber o estado dos botões na seguinte forma:

Resposta do escravo							
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
-	-	-	SW2	-	-	-	SW1

Os valores dos bits SW1 e SW2 serão '1' quando o botão não estiver pressionado e '0' quando o botão estiver pressionado.

## Comportamento esperado

O mestre não tem botões nem LEDs. O mestre deve amostrar o valor dos botões pela linha SPI e deve reagir ao pressionar de cada botão. Quando o mestre perceber que o botão do escravo SW1 foi pressionado, deve incrementar a luminosidade do LED vermelho do escravo de um passo. Ao chegar na luminosidade máxima, deve voltar a luminosidade para zero e continuar incrementando ciclicamente. Exemplo de sequência: {0, 1, 2, 3, 4, 0, 1, 2, ...}. O mesmo comportamento é esperado do botão SW2 só que para o LED verde.

A pontuação será dividida da seguinte forma:

Escravo – 7 pontos

- E1) PWM para controlar a luminosidade dos LEDs (2 pontos)
- E2) Recepção de comandos do mestre (3 pontos)
- E3) Transmissão do estado do botão (2 pontos)

Mestre – 3 pontos

- M1) Amostrar o valor dos botões via SPI (1 ponto)
- M2) Reagir aos botões do escravo e enviar a luminosidade correta (1 ponto)
- M3) Remoção de rebotes (1 ponto)

A solução será avaliada pela clareza de código, forma e eficiência. Exemplo: Código sem comentário ou desorganizado receberão penalidades de 1 ponto. Optar por soluções sem uso de interrupções também receberão penalidades de 1 ponto.

*Dicas de implementação:*

- 1) Use apenas um timer para controlar a luminosidade dos dois LEDs.
- 2) Recomenda-se usar uma frequência de PWM superior a 60Hz.
- 3) Programe todas as ações do escravo por interrupções.
- 4) O mestre deve ser implementado usando pooling apenas.
- 5) O mestre deve tratar os rebotes das chaves do escravo.
- 6) É permitido e recomendado usar as bibliotecas desenvolvidas durante o curso.

## Entrega

O aluno deve entregar o código fonte e uma **foto da montagem** numa pasta compactada zip (não usar rar / 7z / gz / tar / etc.. ). O nome do arquivo zipado deve ser a matrícula do aluno.