

EA201 - EXERCÍCIO PRÁTICO 1

PREPARAÇÃO:

Para os exercícios práticos desta disciplina, vamos analisar sinais digitais encontrados em comunicações entre elementos de um sistema embarcado. Para isto, o professor realizou a captura destes sinais através de um módulo analisador lógico que trabalha junto com um *software*. A versão de demonstração deste *software* é gratuita e permite abrir os arquivos gerados durante a coleta dos dados, para análise. Assim, os alunos precisam instalar o software em seus computadores para poder visualizar e analisar os dados coletados.

O *software* pode ser encontrado no link abaixo (versões para vários sistemas operacionais):

<https://www.saleae.com/downloads/>

Obs: Deve-se usar a versão 2.x do *software*, pois o padrão de arquivo de captura “.sal” não é aceito na versão 1.x. O link acima é para a versão 2.x.

A instalação do Saleae Logic é simples. Ao abrir o programa, na parte principal da janela aparecerá a informação “Connect Saleae Device”, indicando que nenhum analisador lógico está conectado ao computador.

Após a confirmação da correta instalação, é hora de abrir o arquivo com a coleta de dados real realizada. Baixe o arquivo disponibilizado pelo professor (extensão “.sal”). Depois, no programa Logic, no menu padrão no topo da janela, escolha “File” – “Open capture” (Tecla de atalho Ctrl+O) e selecione o arquivo baixado.

Posicione o cursor na área dos sinais e use a roda de rolagem (“scroll”) do mouse para ajustar o zoom para uma leitura mais adequada. Alternativamente, pode-se usar as setas para cima e para baixo. A barra de rolagem horizontal embaixo permite deslocar o sinal para a esquerda e para a direita, podendo-se utilizar também as setas para os lados.

EXERCÍCIO DE TIMERS:

Os dois canais que aparecem no arquivo são saídas que refletem o estado de dois *timers* internos de uma MCU ST32F103C8.

O *Timer 1* tem o seu canal 1 no modo *output compare*. Neste modo, quando a contagem atinge um valor determinado, o pino de saída assume determinado estado lógico via *hardware*, ou seja, sem atrasos de *software*. Isto permite um controle preciso de temporizações. Neste caso, o pino de saída foi programado para fazer um *toggle*, ou seja, inverter seu estado lógico. Após a condição ser atingida (contagem chegar ao valor programado em um registrador), o contador zera, o pino de saída faz o *toggle* e um novo ciclo se inicia.

O *Timer 2* tem o seu canal 2 no chamado “modo PWM”. Este modo corresponde ao *timer* para PWM visto em aula. Como no modo anterior, um registrador determina o máximo valor de contagem, quando ocorre o reinício de contagem a partir de zero, definindo o período (e também a frequência) da onda PWM. Há ainda um registrador de “Pulso”, que determina o momento da

transição do PWM. Ou seja, para a saída do contador entre o valor zero e o correspondente ao gravado no registrador “pulso”, o pino correspondente permanece em nível Alto. Para valores entre o gravado no registrador “pulso” e o valor final de contagem definido no outro registrador, o pino correspondente permanece em nível Baixo. Cada *timer* possui um *prescaler* independente, que faz a divisão inicial da frequência do *clock* para seu *timer*. Neste exercício, a frequência do sistema (barramentos APB) para os dois *timers* (antes dos *prescalers*) foi configurada para 16MHz.

1. Para cada canal, ajuste um *zoom* adequado e posicione o mouse na área do canal correspondente, e veja que o programa apresenta várias leituras temporais. Ao colocar o cursor sobre uma área onde o sinal está em nível alto, o primeiro valor corresponde à duração deste nível alto, enquanto que ao colocar o cursor sobre a área com o sinal em nível baixo, o primeiro valor será a duração do nível baixo. Faça as medidas e apresente os valores de t_H , t_L (tempos em nível alto e baixo), período e frequência de cada um dos dois sinais.
2. O *Timer* 1 (Canal 0 do analisador) teve seu *clock* de contagem ajustado para 100kHz. Qual o valor que foi escrito no registrador de *prescaler* para se atingir esta frequência? **Obs:** os contadores dos *prescalers* possuem *reset* síncrono, ou seja, seu reinício não ocorre assim que atinge o valor de contagem, mas apenas após o próximo pulso de *clock*. Por isso, se o valor escrito for 35, a divisão será por 36, pois o timer vai contar de 0 a 35, inclusive. Assim, o valor a ser escrito no registrador é um a menos do que o fator de divisão.
3. Qual o valor que deve ser escrito no registrador de máximo valor de contagem do *Timer* 1 para que o sinal de saída seja o apresentado no arquivo? **Obs 1:** Como o pino de saída de sinal trabalha em *toggle*, o tempo de contagem define o tempo entre transições, ou seja, ocorre uma contagem completa para o t_H e outra contagem completa para o t_L . Por isso, o período de contagem do *timer* é metade do período da onda quadrada. **Obs 2:** o contador do *timer* também possui *reset* síncrono, então vale a mesma regra da observação do item 2.
4. O *Timer* 2 (Canal 1 do analisador) teve seu *clock* de contagem ajustado para 10kHz. Qual o valor que foi escrito no registrador de *prescaler*?
5. Para o período e frequência de PWM medido no canal 2, qual o valor do registrador de máximo valor de contagem? **Obs:** a mesma do item 2, mas não vale aqui a primeira observação do item 3, pois o ciclo completo de sinal ocorre durante uma única contagem até o valor máximo.
6. Para o t_H medido no canal 2, qual o valor do registrador de “pulso”, que define a transição de nível Alto para Baixo? **Obs:** neste caso não se aplica a observação do item 2, pois a transição no pino ocorre assim que o valor de contagem atinge o valor programado no registrador “pulso”.
7. Quando controlamos um PWM, o valor que interessa é o *duty cycle*, mas o que modificamos é o registrador “pulso”. Neste exercício, qual a relação entre o valor do registrador “pulso” e o *duty cycle* desejado? Em outras palavras, a quantas unidades no valor do registrador “pulso” corresponde cada 1% do *duty cycle*?
8. Suponha que precise configurar um *timer* no modo *input capture*. A frequência do *clock* do sistema continua sendo 16MHz. É necessário que o *timer* (de 16 bits) conte um tempo de até 3 segundos sem *overflow*. Sugira um valor de *prescaler* para que isso seja possível, e determine a resolução (intervalo de tempo entre 2 contagens sucessivas) do *timer* neste caso.