

**Patrick de Carvalho T. R. Ferreira**

1. Time Overflow Interruption (TOI) é uma interrupção do KL25 que pode ser habilitada para chamar uma função de tratamento setando-se o bit 6 do registrador TPMx\_SC (onde x indica qual o módulo TPM sendo utilizado). É chamada sempre que o contador de um TPM atinge valor igual ao do respectivo registrador MOD e incrementa.
2. Deve-se setar o bit 6 (Timer Overflow Interrupt Enable - TOIE) do registrador TPMx\_SC (onde x indica qual o módulo TPM sendo utilizado). Além disso, deve-se setar o bit equivalente ao respectivo número de IRQ da dada interrupção no registrador NVIC\_ISER. Para habilitar esta interrupção no TPM0, por exemplo, deve-se setar o bit 17 de NVIC\_ISER.
3. **void FTMx\_IRQHandler(void)** é o nome da rotina de tratamento desta interrupção (onde x é o módulo TPM sendo tratado). Para obter esta informação, foi necessário consultar a tabela Table 3-7. Interrupt vector assignments no manual do kl25 e obter o índice da rotina desejada no vetor de interrupção dentro de kinetis\_sysinit.c, no CodeWarrior. Consultando agora kinetis\_sysinit.c e contando os elementos do vetor, identificou-se que esta rotina era **void FTMx\_IRQHandler(void)**.
4. No registrador TMPx\_SC (onde x é o dado módulo TPM), o bit 7, Timer Overflow Flag, estará setado em ALTO caso esta interrupção seja devido ao evento de Time Overflow.
5. É uma interrupção invocada por um canal do TPM atuando no modo captura de entrada. Ela será chamada sempre que o respectivo canal ler uma borda de pulso (de subida ou descida, de acordo com as configurações).
6. No registrador TPMx\_CnSC, CPWMS, MSnB e MSnA devem estar resetados em zero e CHIE, ELSnA e ELSnB devem estar setados em um (ALTO) quando se deseja detectar bordas de subida e/ou descida, respectivamente. Além disso, deve-se setar o bit equivalente ao respectivo número de IRQ da dada interrupção no registrador NVIC\_ISER. Para habilitar esta interrupção no TPM0 CH0, por exemplo, deve-se setar o bit 17 de NVIC\_ISER.
7. **void FTMx\_IRQHandler(void)** é o nome da rotina de tratamento desta interrupção (onde x é o módulo TPM que hospeda o canal sendo tratado). Para obter esta informação, foi necessário consultar a tabela Table 3-7. Interrupt vector assignments no manual do kl25 e obter o índice da rotina desejada no vetor de interrupção dentro de kinetis\_sysinit.c, no CodeWarrior. Consultando agora kinetis\_sysinit.c e contando os elementos do vetor, identificou-se que esta rotina era **void FTMx\_IRQHandler(void)**.
8. Em TPMx\_CnSC, devemos olhar se CHF foi setada (para dizer se é interrupção de canal) e qual flag foi setada para identificar qual canal causou a interrupção.
9. Sempre que houver uma borda de pulso sendo detectada no canal que executada leitura, podemos verificar qual o ponto de contagem do LPTPM counter. Se usarmos esta informação (sabendo-se o período de contagem) para detectar o intervalo entre

duas bordas iguais, mediremos o período do ciclo de pwm, que é o inverso da frequência. Se usarmos esta informação (sabendo-se o período de contagem) para detectar o intervalo entre duas bordas (uma de subida e a outra de descida) estaremos obtendo a largura do pulso de pwm.

10. Podemos fazer isto verificando-se quantas interrupções de Time Overflow ocorreram e inserindo este dado nos nossos cálculos de modo a incluir um período maior do que o do contador atingir o valor de MOD. A partir de então, contamos o tempo até o final do primeiro período mensurado e o tempo decorrido da última contagem incompleta.