PSI2662 – Projeto em Sistemas Eletrônicos Embarcados: Sensores e Atuadores

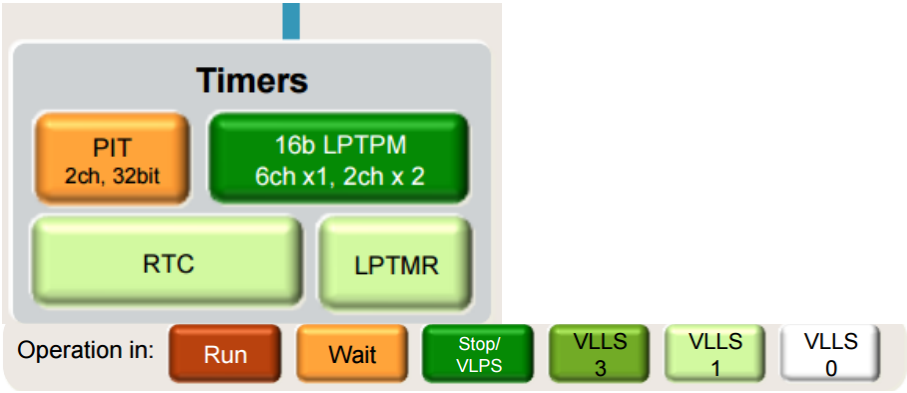
Contadores e Timers

Escola Politécnica da Universidade de São Paulo

Prof. Gustavo Rehder – grehder@lme.usp.br

Segundo Semestre de 2015

Timers no KL25Z



- PIT - Periodic Interrupt Timer
  - Gera interrupções periódicas.
- TPM - Timer/PWM Module
  - Conectado a portas de I/O; possui input capture, output compare, pode gerar sinais de PWM; pode gerar interrupções.
- LPTMR - Low-Power Timer
  - Pode operar como timer ou contador in todos os modos de potência; pode "acordar" o sistema com interrupções; pode sincronizar o hardware.
- Real-Time Clock
  - Alimentado por um cristal externo de 32.768 kHz; rastreia tempo em segudos utilizando um registrador de 32 bits; pode gerar um alarme; pode gerar um sinal de 1 Hz e/ou uma interrupção; pode "acordar o sistema com interrupção.
- SYSTICK
  - Parte do Cortex M0+ Core; contador que pode gerar interrupções

System Tick Timer

- Ação executada periodicamente
- 24-bit down counter
- Clock ou Clock/16

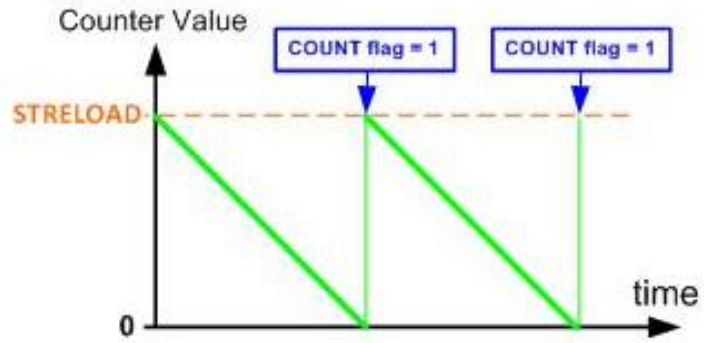


Figure 5-9: System Tick Counting

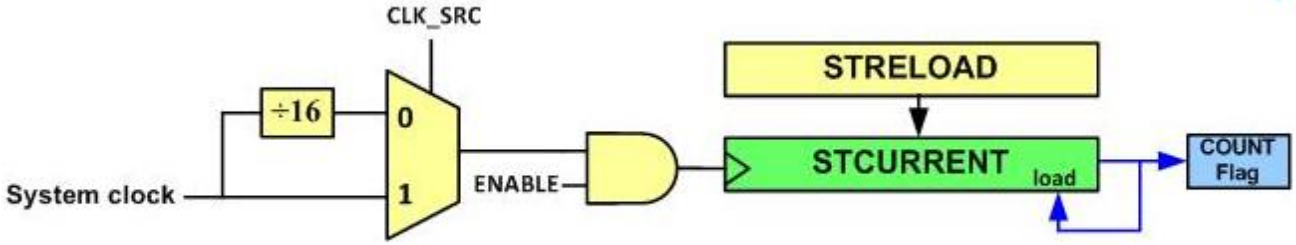


Figure 5-7: System Tick Timer Internal Structure

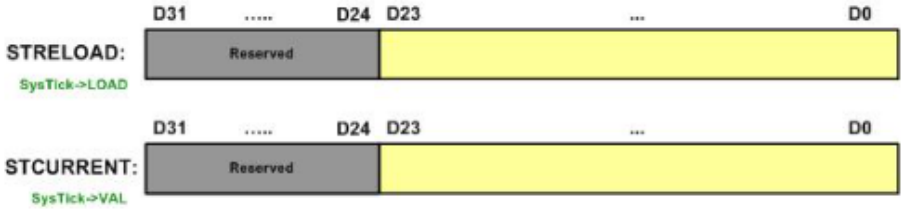


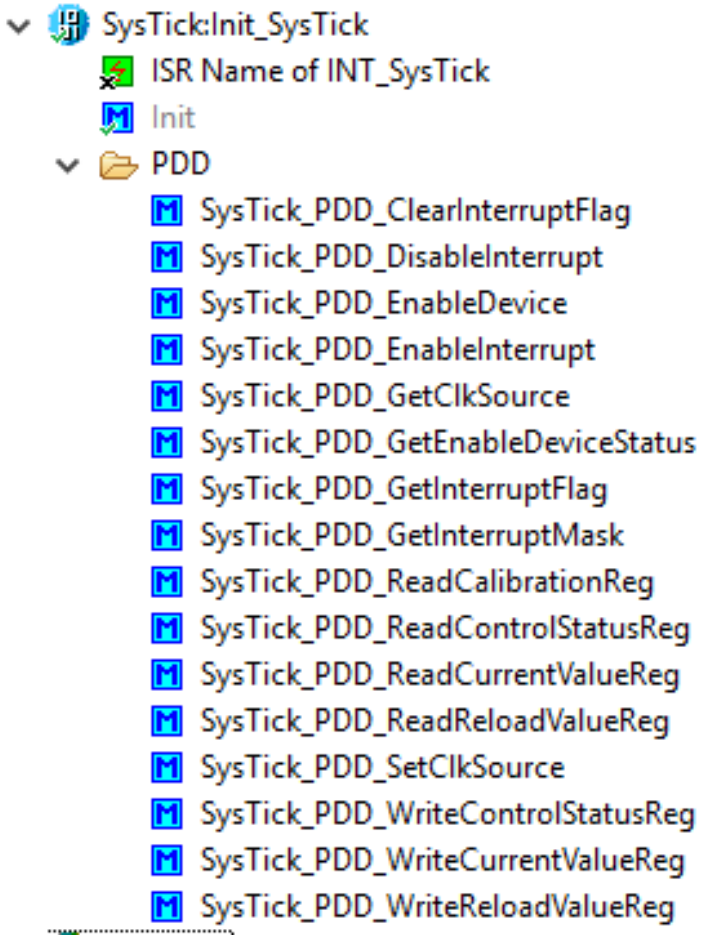
Figure 5-10: STRELOAD vs. STCURRENT

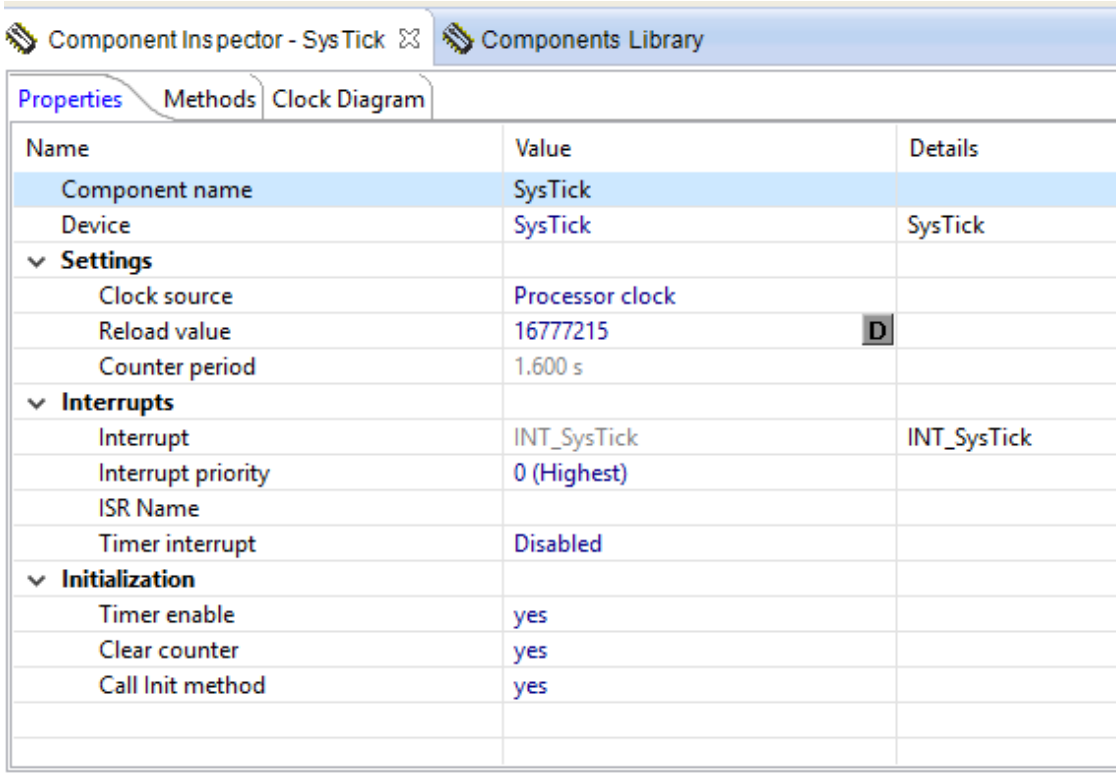
bit	Name	Description
0	ENABLE	Enable (0: the counter is disabled, 1: enables SysTick to begin counting down)
1	INTEN	Interrupt Enable 0: Interrupt generation is disabled, 1: when SysTick counts to 0 an interrupt is generated
2	CLK_SRC	Clock Source 0: System clock divided by 16 1: System clock
16	COUNT	Count Flag 0: the SysTick has not counted down to zero since the last time this bit was read 1: the SysTick has counted down to zero *Note: this flag is cleared by reading the STRCTL or writing to STCURRENT register.*

Figure 5-8: STCTRL (System Tick Control)

SysTick Contador CodeWarrior (PE)

Incluir (add file...) PDD header em <CodeWarrior Installation Path>\MCU\ProcessorExpert\lib\Kinetis\pdd\inc\






```
for(;;)
{
if(SysTick_PDD_ReadControlStatusReg(SysTick_DEVICE)&0x10000)
{
Bit1_NegVal();
}
}
```

Timer/PWM Module

- 3 módulos (TPMx = TPM0, TPM1 e TPM2)
  - 1 com 6 canais e 2 com 2 canais
  - Contador de 16 bits (up ou down)
  - Modos: Output Compare, Input Capture e PWM

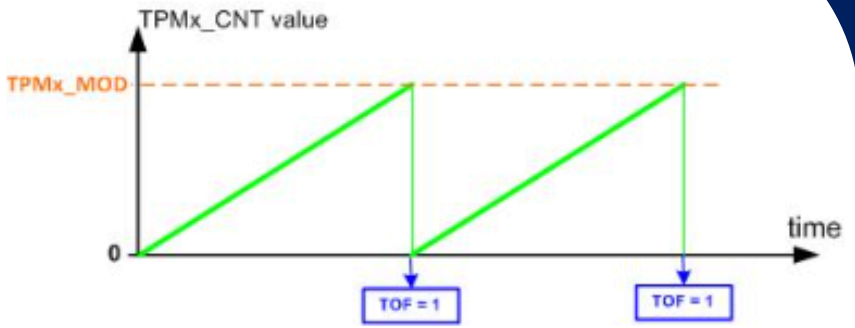


Figure 5-16: The role of TPMx_MOD

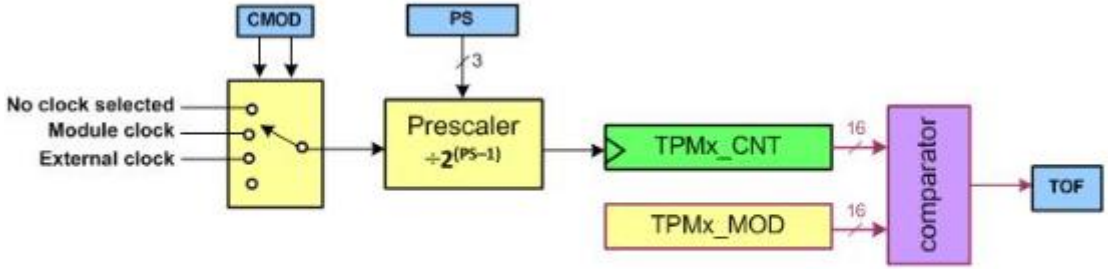
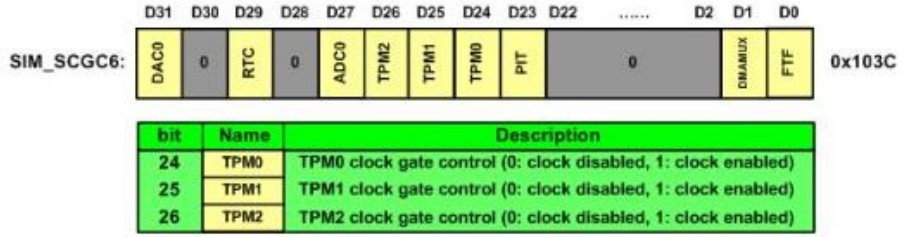


Figure 5-17: CMOD and PS (Prescaler) bits



bit	Name	Description
24	TPM0	TPM0 clock gate control (0: clock disabled, 1: clock enabled)
25	TPM1	TPM1 clock gate control (0: clock disabled, 1: clock enabled)
26	TPM2	TPM2 clock gate control (0: clock disabled, 1: clock enabled)

Figure 5-11: SIM_SCGC6 (SIM Clock Gating Control Register 6)

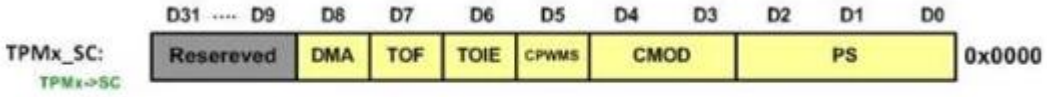
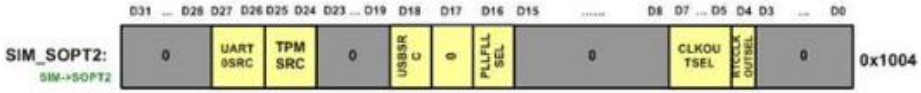


Figure 5-18: Timer Status and Control (TPMx_SC) Register



TPM clock source select selects the clock source for the TPM counter clock

TPMSRC	Selected Clock
00	Clock disabled
01	MCGFLLCLK clock or MCGFLLCLK/2
10	OSCERCLK clock
11	MCGIRCLK clock

Figure 5-12: SIM_SOPT2 (System Options 2)

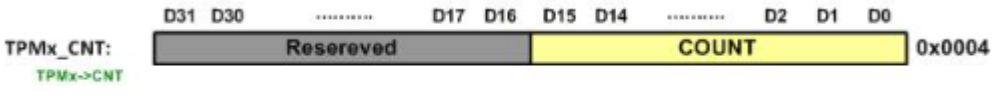


Figure 5-13: TPMx_CNT Register

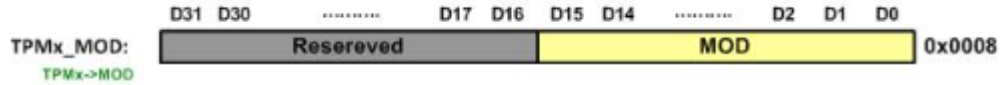
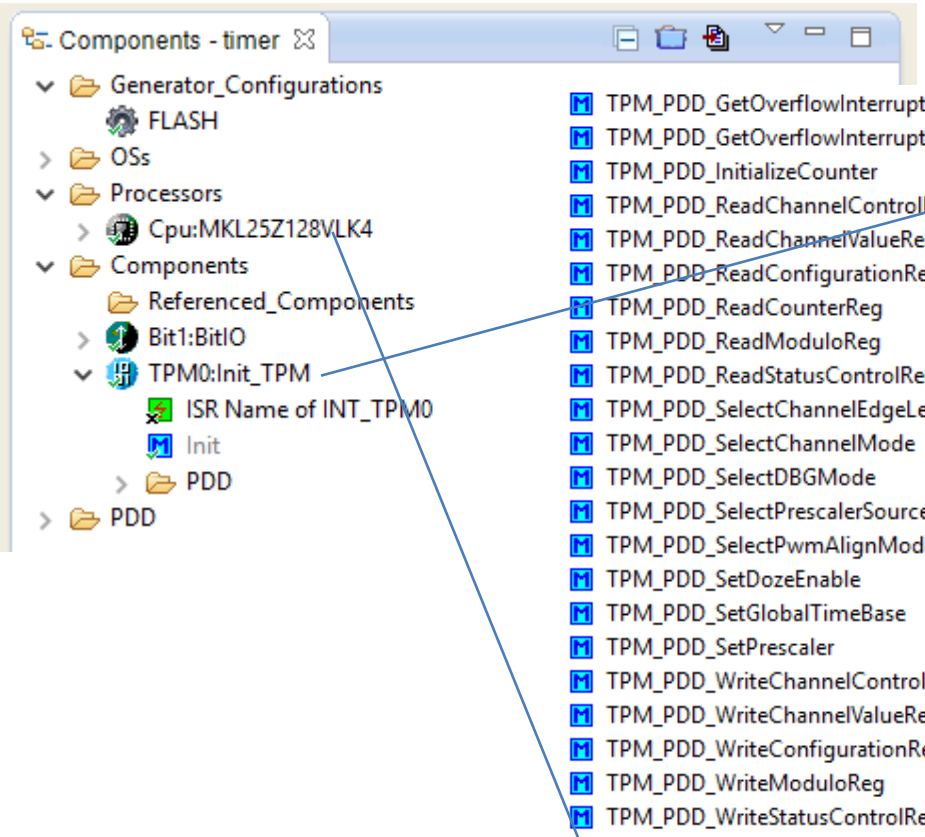


Figure 5-14: TPMx_CNT and TPMX_MOD registers

TPM Contador CodeWarrior (PE)



Name	Value
Component name	TPM0
Device	TPM0
Settings	
Clock gate	Enabled
Clock settings	
Clock source	TPM counter clock
Prescaler	divide by 128
Counter frequency	156.250 kHz
Modulo counter	65535 D
Period	419.430 ms
DBG mode	TPM counter stopped; output pins...
Global time base	Disabled
Counter reload on trigger	Disabled
Counter start on trigger	Disabled
Counter stop on overflow	Disabled
Counter in Doze mode	Enabled
Channels	
Pins	
Interrupts	
Initialization	
Call Init method	yes

```
for(;;)
{
if(TPM_PDD_ReadStatusControlReg(TPM0_DEVICE)&0x80)
{
TPM_PDD_WriteStatusControlReg(TPM0_DEVICE,TPM_PDD_ReadSta
tusControlReg(TPM0_DEVICE)|0x80);
Bit1_NegVal();
}
}
```

TPM clock selection	Internal reference	
Clock frequency [MHz]	0.032768	0.032768 MHz

TPM Output Compare

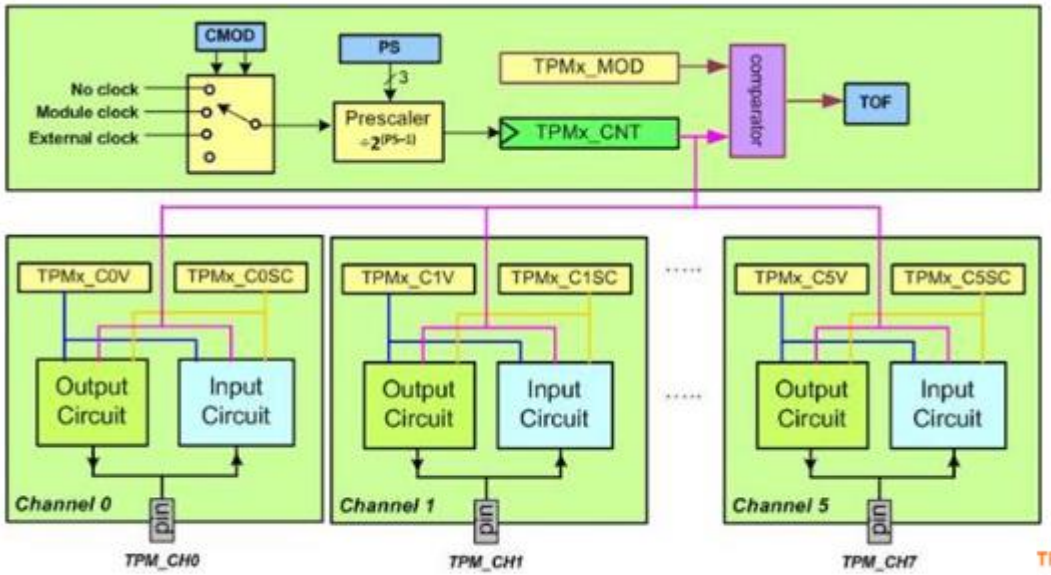


Figure 5-19: The Channels of TPMx

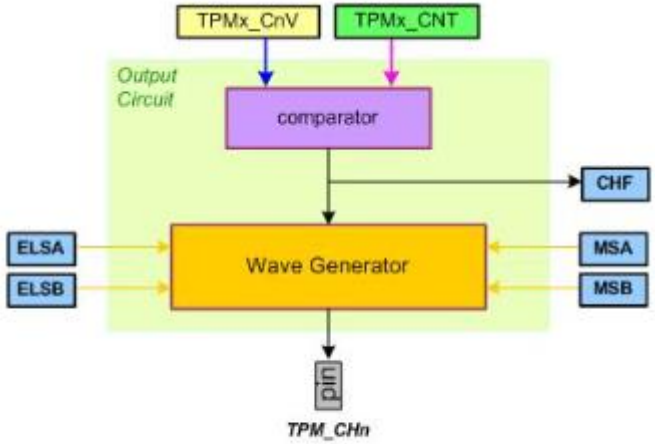


Figure 5-21: Output Circuit

Figure 5-20: TPMx_CnV (TPMx Channel Value) Register

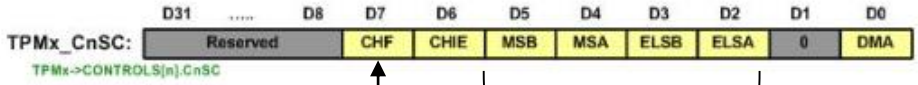


Figure 5-22: TPMxCnSC (TPMx Channel Status and Control)

Flag

Configuração da Saída

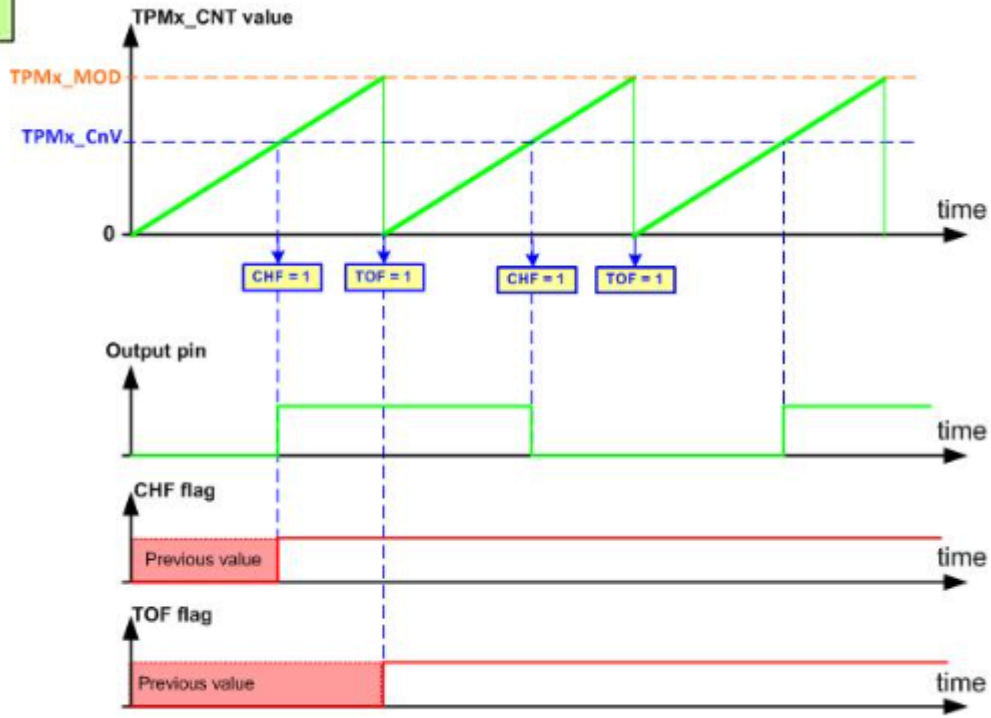
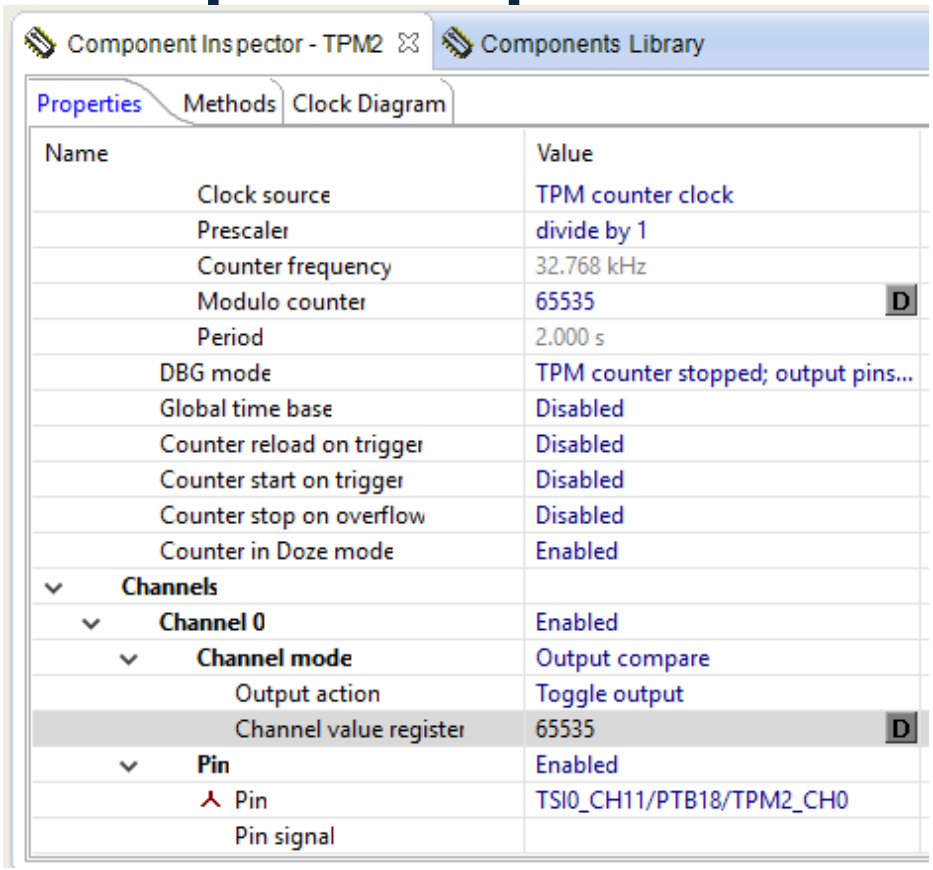


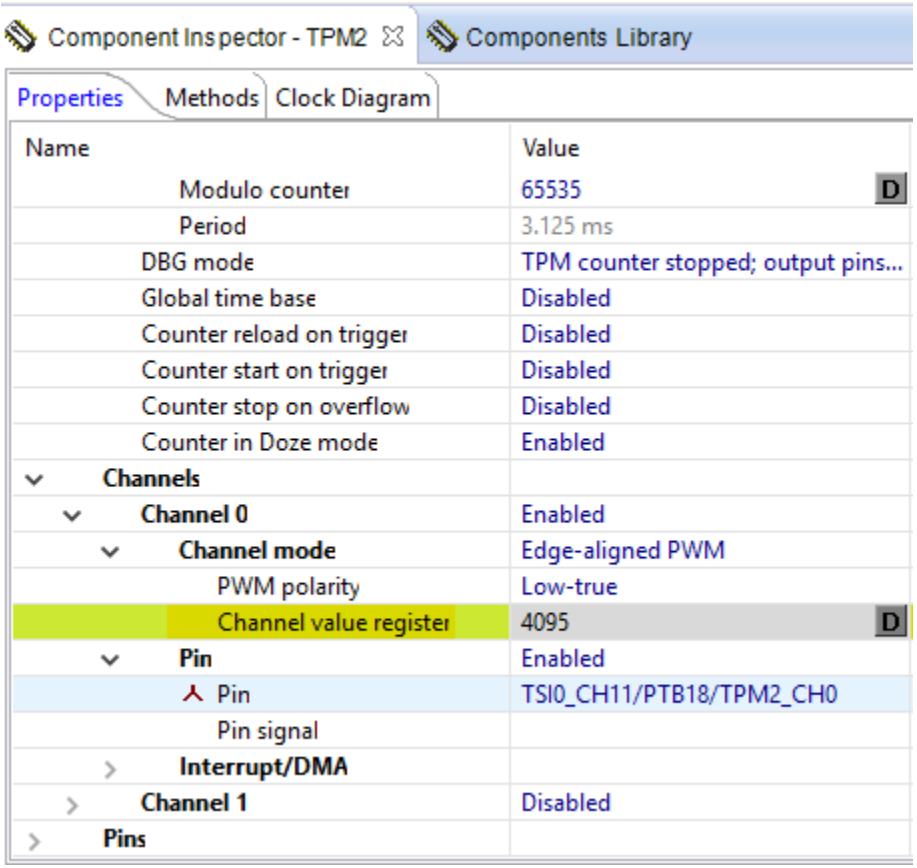
Figure 5-23: In Toggle Mode

TPM Contador CodeWarrior (PE):

Output Compare



PWM




```
for(;;) { }
```

Input Capture

Frequência e Largura de Pulso

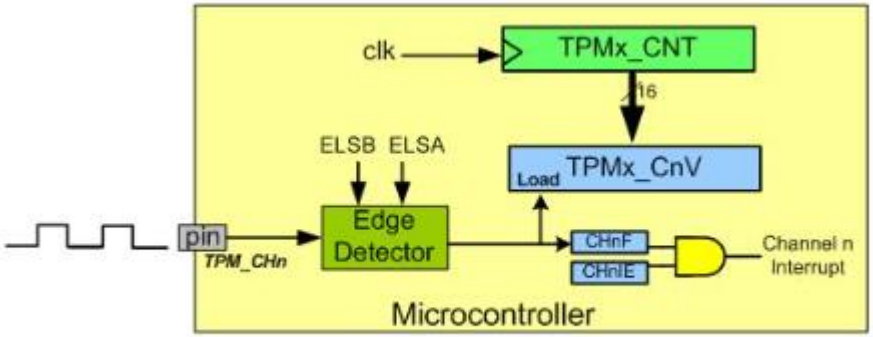


Figure 5-27: Input Edge Time Capturing

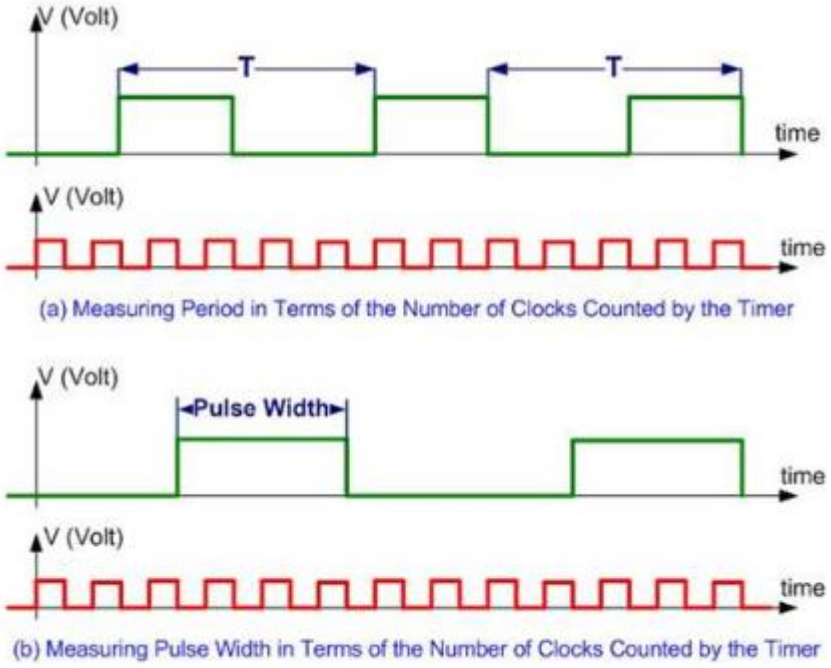


Figure 5-28: Measuring Period and Pulse Width

Eventos

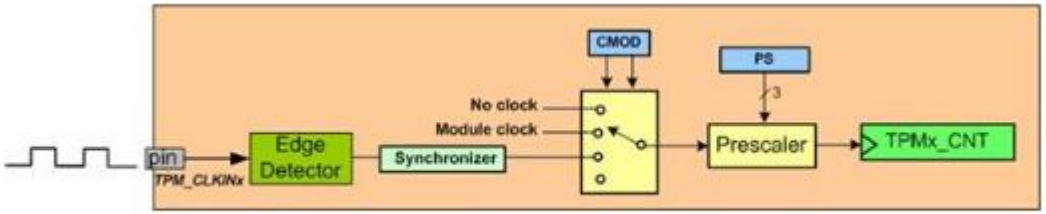
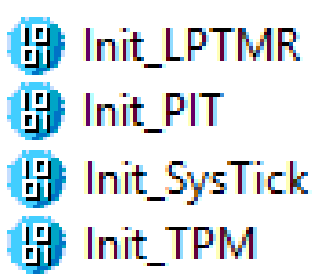


Figure 5-29: Counter Diagram

→ Interrupção

Code Warrior Processor Expert -Timers



Utiliza PDD – configuração direta com registradores

Auxilia na configuração de qualquer tipo de timer

Interrupção periódica com qualquer tipo de timer

Configuração do PWM utilizando um timer TPM.

Code Warrior Processor Expert -Outros

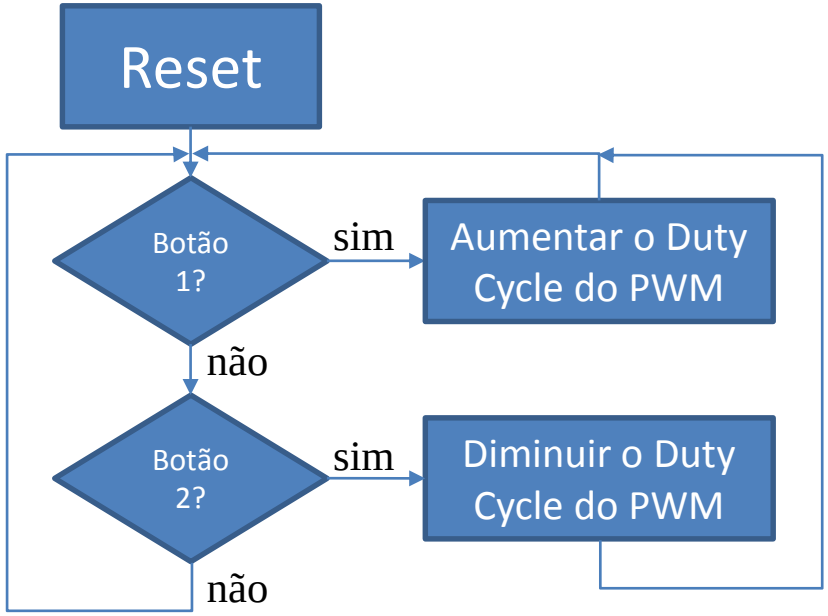
Inicialização das portas GPIO – Configurações (pull-up drive strength, filtro etc.)

Configuração de Pinos individuais – Entrada/Saída

Componentes para importar (Wait): http://www.steinerberg.com/EmbeddedComponents/

Exercício

- Monitorar entradas para aumentar e diminuir o *duty-cycle* de um PWM que alimenta um LED.



ou usar componente PWM do Processor Expert

olhar pinagem

Configurações:

- Timer TPM – PWM
  - Qual timer usar? TPM0, TPM1 ou TPM2?
  - Qual canal?
  - Qual frequência do timer?

  Init_TPM
- Entradas digitais
  - Quais pinos?
  - Pull-up, pull-down?

Qual frequência da CPU?

Desafio: Utilizar o botão de rest para controlar a intensidade do LED.