PSI3441 – Arquitetura de Sistemas Embarcados

Interrupções

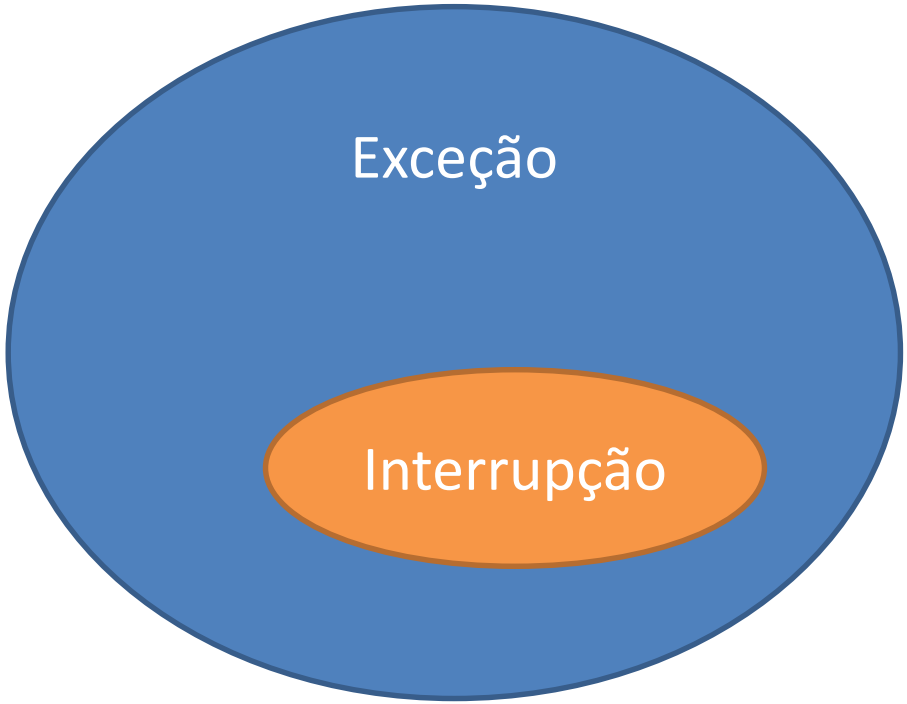
Escola Politécnica da Universidade de São Paulo

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

Primeiro Semestre de 2017

Exceção – Interrupção

- Exceção: Qualquer evento que altera o fluxo normal de execução de instruções.



Exceção: eventos anormais:
- Instruções inválidas
- Acesso ilegal

Interrupção: Sinal de hardware
- Sinais externos
- Flags de periféricos

Pooling vs. Interrupção

- Pooling

```
// Monitora continuamente o flag do registrador.

//No main
for(;;)
{
if(SysTick_PDD_ReadControlStatusReg(SysTick_DEVICE)&0x10000)
{
Bit1_NegVal();
}
}
```

- Interrupção

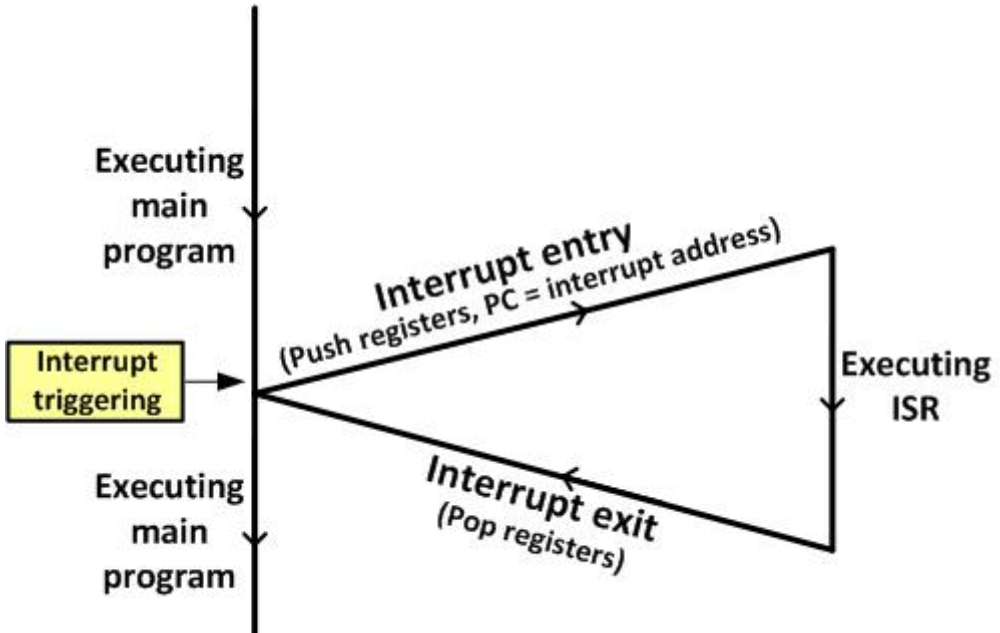
```
//Espera uma interrupção para executar a função.

//No main
for(;;)
{
}

//No Events (Interrupt service routine (ISR))
void TI1_OnInterrupt(LDD_TUserData *UserDataPtr)
{
Bit1_NegVal();
}
```

O que acontece quando ocorre uma exceção/interrupção

- CPU salva estado atual (contadores, registradores, etc);
- CPU busca o endereço das rotinas de tratamento de interrupção (Interrupt Service Routine - ISR) em uma tabela (Vector Table);
- CPU vai para o endereço do ISR e executa seu código (função);
- CPU retorna ao está anteriormente salvo.



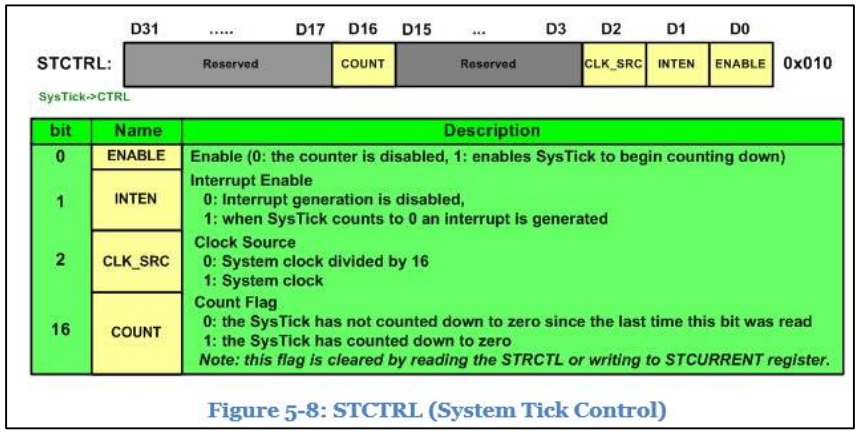
Vector Table

Core

SysTick

Address	Vector	IRQ[1]	NVIC IPR register number[2]	Source module	Source description
0x0000_0000	0	—	—	ARM core	Initial Stack Pointer
0x0000_0004	1	—	—	ARM core	Initial Program Counter
0x0000_0008	2	—	—	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	—	—	ARM core	Hard Fault
0x0000_0010	4	—	—	—	—
0x0000_0014	5	—	—	—	—
0x0000_0018	6	—	—	—	—
0x0000_001C	7	—	—	—	—
0x0000_0020	8	—	—	—	—
0x0000_0024	9	—	—	—	—
0x0000_0028	10	—	—	—	—
0x0000_002C	11	—	—	ARM core	Supervisor call (SVCall)
0x0000_0030	12	—	—	—	—
0x0000_0034	13	—	—	—	—
0x0000_0038	14	—	—	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	—	—	ARM core	System tick timer (SysTick)
Non-Core Vectors					
0x0000_0040	16	0	0	DMA	DMA channel 0 transfer complete and error
0x0000_0044	17	1	0	DMA	DMA channel 1 transfer complete and error
0x0000_0048	18	2	0	DMA	DMA channel 2 transfer complete and error
0x0000_004C	19	3	0	DMA	DMA channel 3 transfer complete and error
0x0000_0050	20	4	1	—	—
0x0000_0054	21	5	1	FTFA	Command complete and read collision
0x0000_0058	22	6	1	PMC	Low-voltage detect, low-voltage warning
0x0000_005C	23	7	1	LLWU	Low Leakage Wakeup
⋮					
0x0000_00A4	41	25	6	DAC0	
0x0000_00A8	42	26	6	TSI0	
0x0000_00AC	43	27	6	MCG	
0x0000_00B0	44	28	7	LPTMR0	
0x0000_00B4	45	29	7	—	
0x0000_00B8	46	30	7	Port control module	Pin detect (Port A)
0x0000_00BC	47	31	7	Port control module	Pin detect (Port D)

Timer



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)

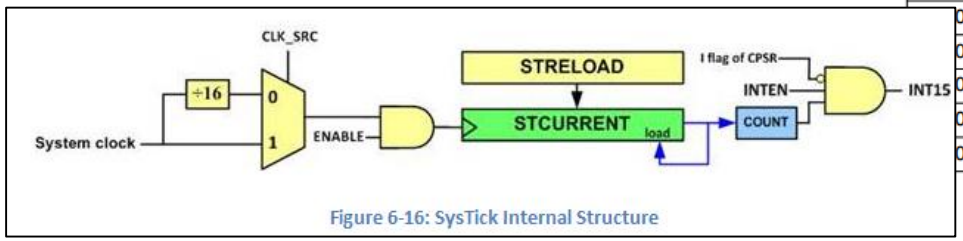
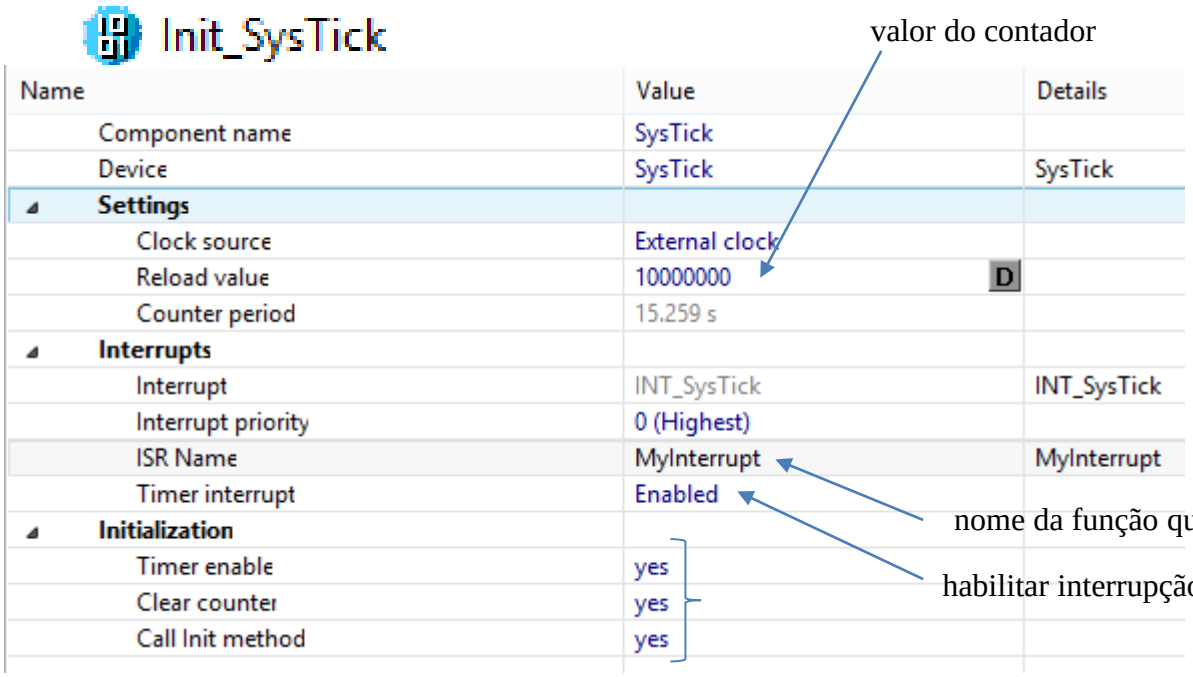


Figure 6-16: SysTick Internal Structure

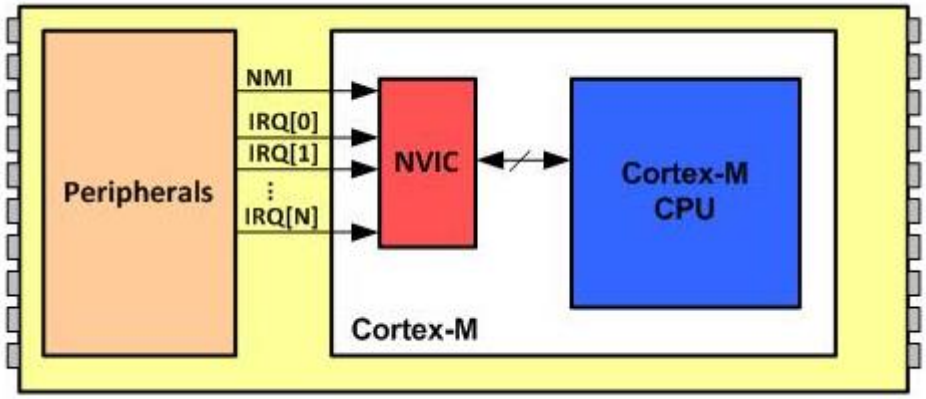
Interrupção SysTick no Code Warrior



- Habilitar interrupção na configuração do processador (Initialization Priority)
- Escrever o código da função em Events.c
- Não é preciso escrever nada no main.c

```
//Exemplo de função escrita no Events.c para mudar o estado de um LED
configurado com o componente BitIO
void MyInterrupt(){
 Bit3_NegVal();
}
```

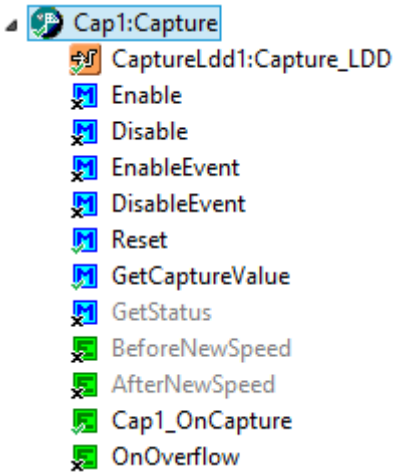
Interrupções de Periféricos



NVIC – Nested Vectored Interrupt Controller: Controla todos as interrupções externas

Interrupção - Input Capture (TPM)

Escolha o pino para monitorar – nem todos os pinos estão disponíveis. Escolhendo o pino, o timer é selecionado automaticamente



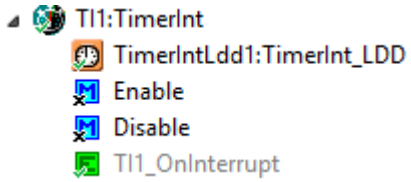
Name	Value	Details
Component name	Cap1	
Capture device	TPM0_C5V	TPM0_C5V
Counter	TPM0_CNT	TPM0_CNT
Capture input pin	CMP0_IN3/PTC9/I2C0_SDA/TPM0...	CMP0_IN3/PTC9/I2C0_SDA/TPM0_CH5
Capture input signal		
Edge	rising or falling edge	rising edge
Interrupt service/event	Enabled	
Capture interrupt	INT_TPM0	INT_TPM0
Capture priority	medium priority	2
Maximum time of event	400 ms	400 ms
Initialization		
Enabled in init. code	yes	
Events enabled in init.	yes	
CPU clock/speed selection		
High speed mode	This component enabled	This component is enabled
Low speed mode	This component disabled	This component is disabled
Slow speed mode	This component disabled	This component is disabled
Referenced components		
Capture_LDD	Capture_LDD	CaptureLdd1

Configure o pino para pull-up

Habilite a interrupção

- Habilitar interrupção na configuração do processador (Initialization Priority)
- Escrever o código da função em Events.c – na função gerada Cap1_onCapture
- Não é preciso escrever nada no main.c

Interrupção - Periódica



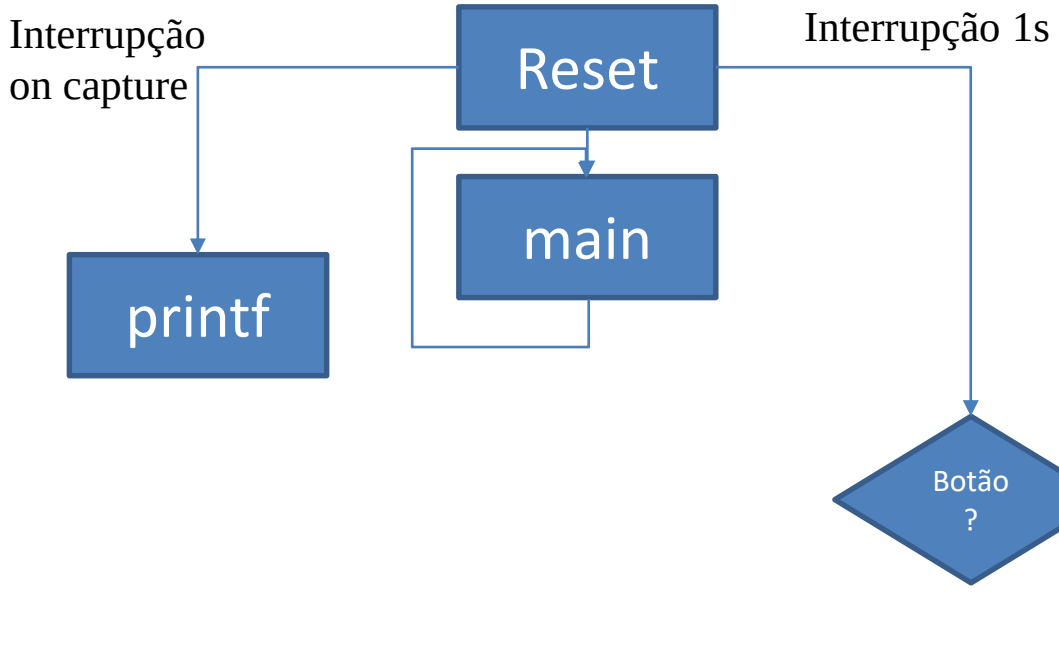
Name	Value	Details
Periodic interrupt source	PIT_LDVAL0	PIT_LDVAL0
Interrupt service/event	Enabled	
Interrupt priority	medium priority	2
Interrupt period	200 ms	200 ms
Initialization		
Enabled in init. code	yes	
Referenced components		

Escolha o tempo

- Habilitar interrupção na configuração do processador (Initialization Priority)
- Escrever o código da função em Events.c – na função gerada TI1_OnInterrupt
- Não é preciso escrever nada no main.c

Exercício

- Piscar LED Azul 1 x/seg. Ao pressionar um botão, piscar LED Verde e mandar para terminal do computador um aviso. Ao soltar o botão, voltar a piscar LED azul.



Configurações:

- Entrada digital com pull-up;
- Input Capture;
- Console (componente do processor Expert para comunicação serial
- Saídas – BitIO (Led Verde e Azul)

Printf no CodeWarrior/PE

- Usar o componente “ConsoIO”
- Configurar os pinos adequadamente
- Configurar o baudrate