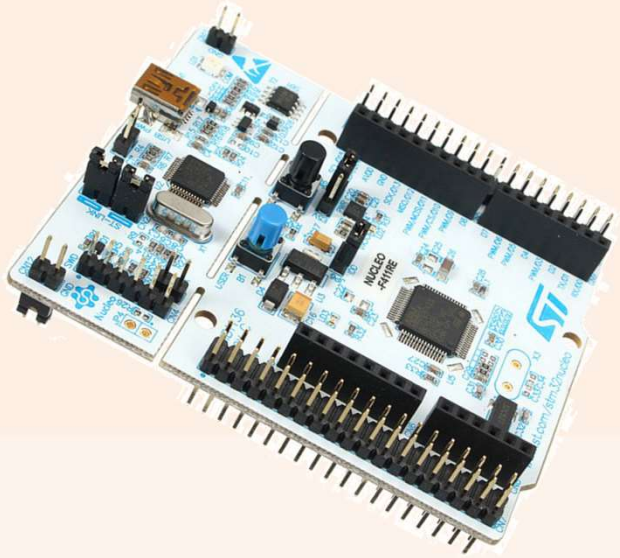
Competências Transferíveis

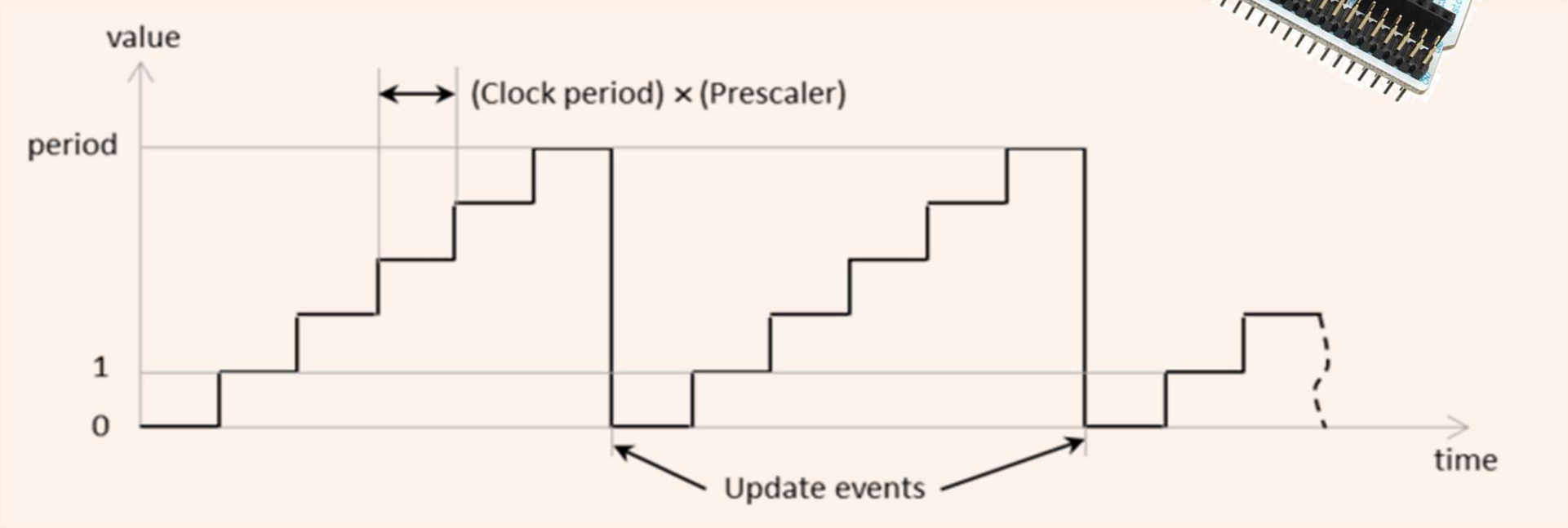
Microcontroladores e Interação com Sensores e Atuadores

V1_2

Rui Escadas Martins

Timers and Interrupts





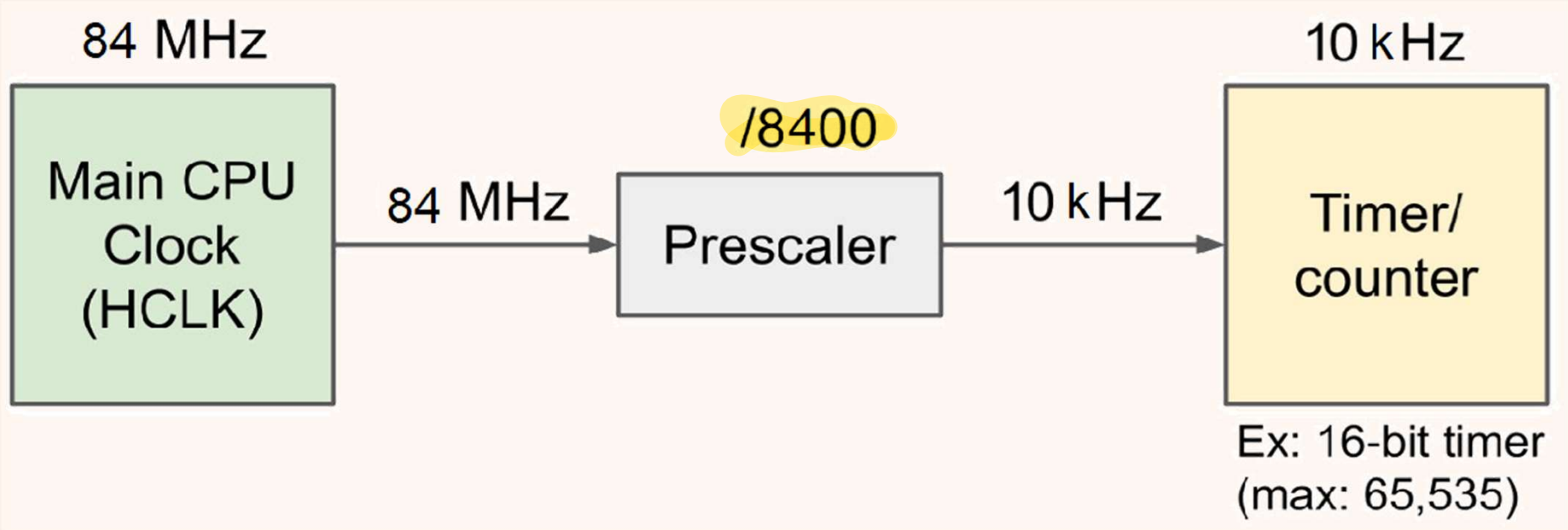
Timers

O que são:

São periféricos que manipulam contagem de eventos e/ou tempo com muita precisão.

Tipicamente, a referência é derivada do clock interno, mas pode originar de um sinal ligado a um determinado pino (externa) ou a outro oscilador (interno).

Como a frequência de relógio dos microcontroladores é muito elevadas é normal divider-se por um valor inteiro designado por "prescaler"!

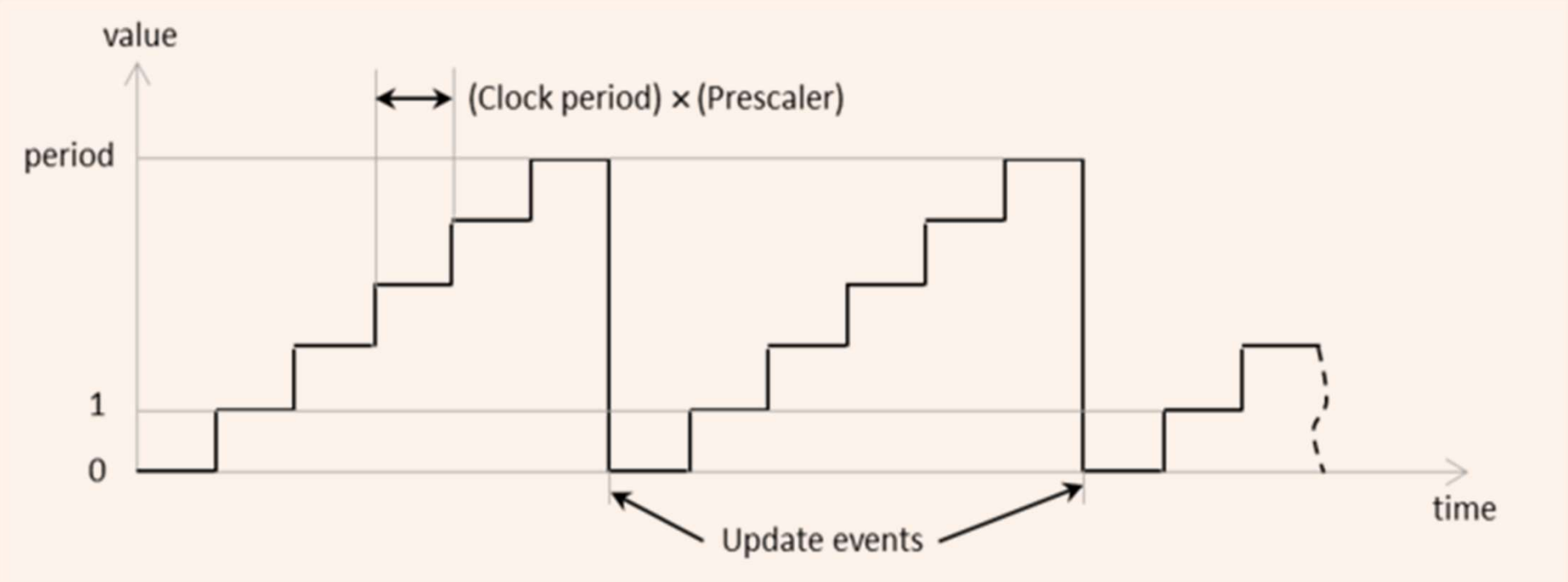


Timers

O que são:

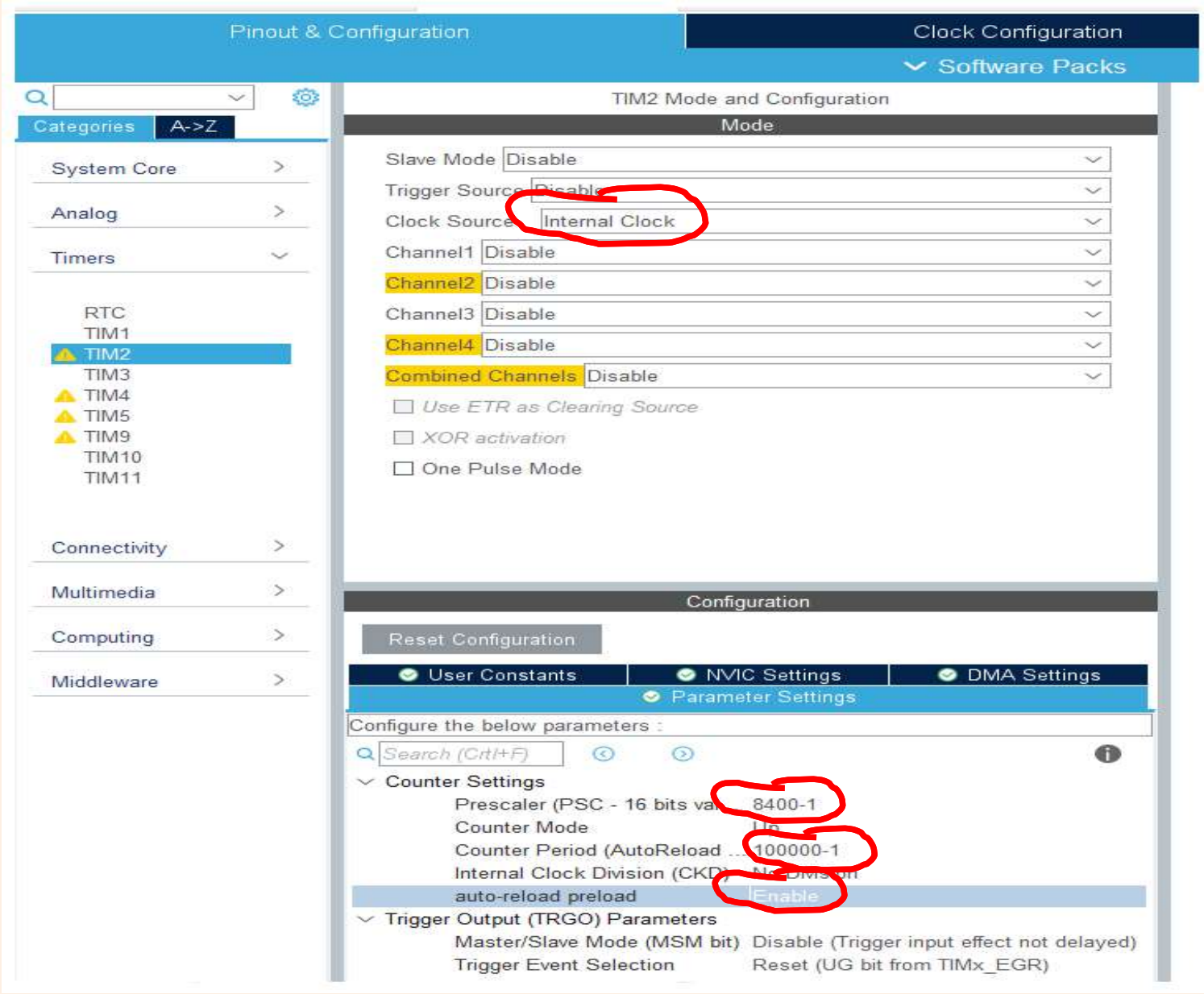
São periféricos que manipulam contagem de eventos e/ou tempo com muita precisão.

Tipicamente, a referência é derivada do clock interno, mas pode originar de um sinal ligado a um determinado pino (externo) ou a outro oscilador.



Timers

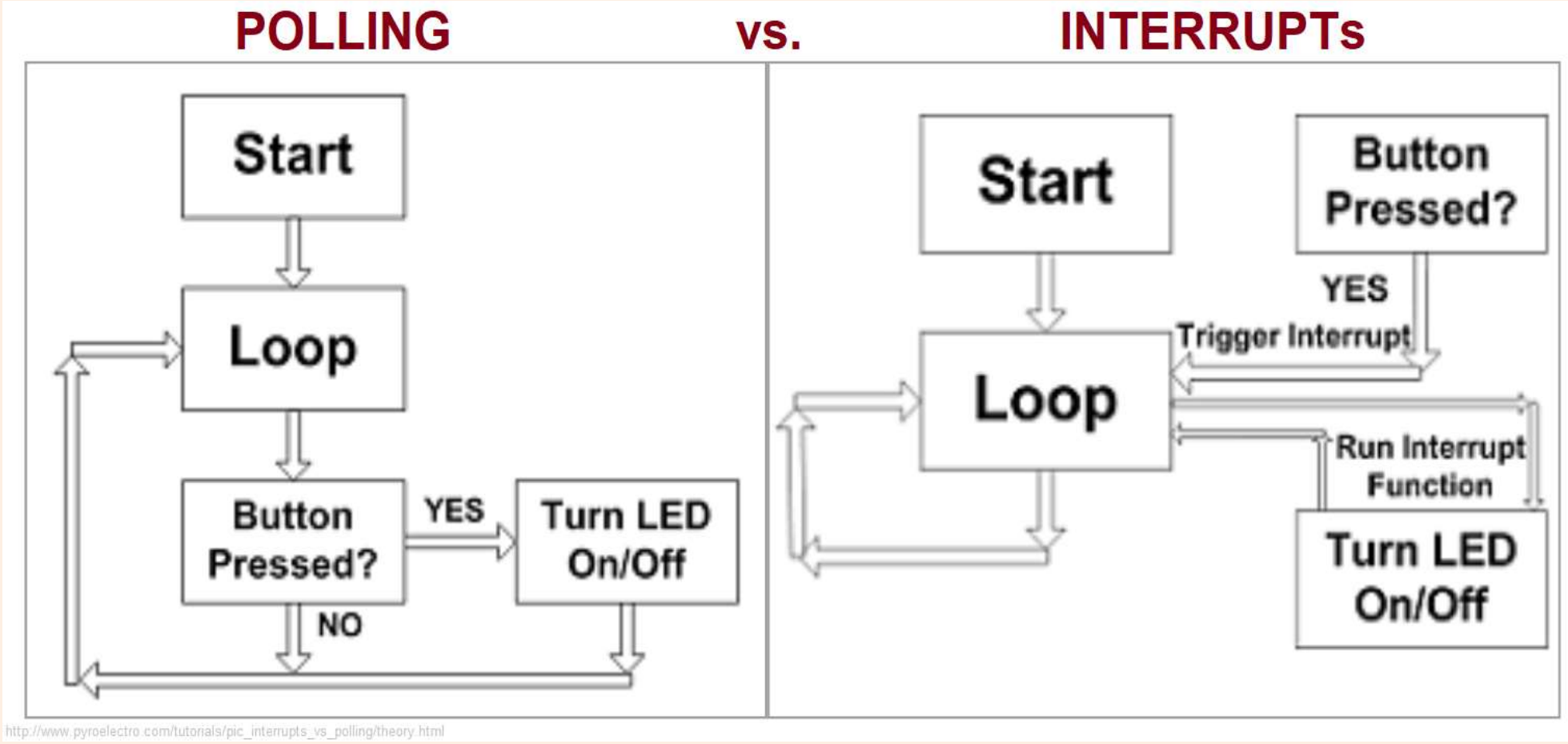
Configuração:



Interrupts

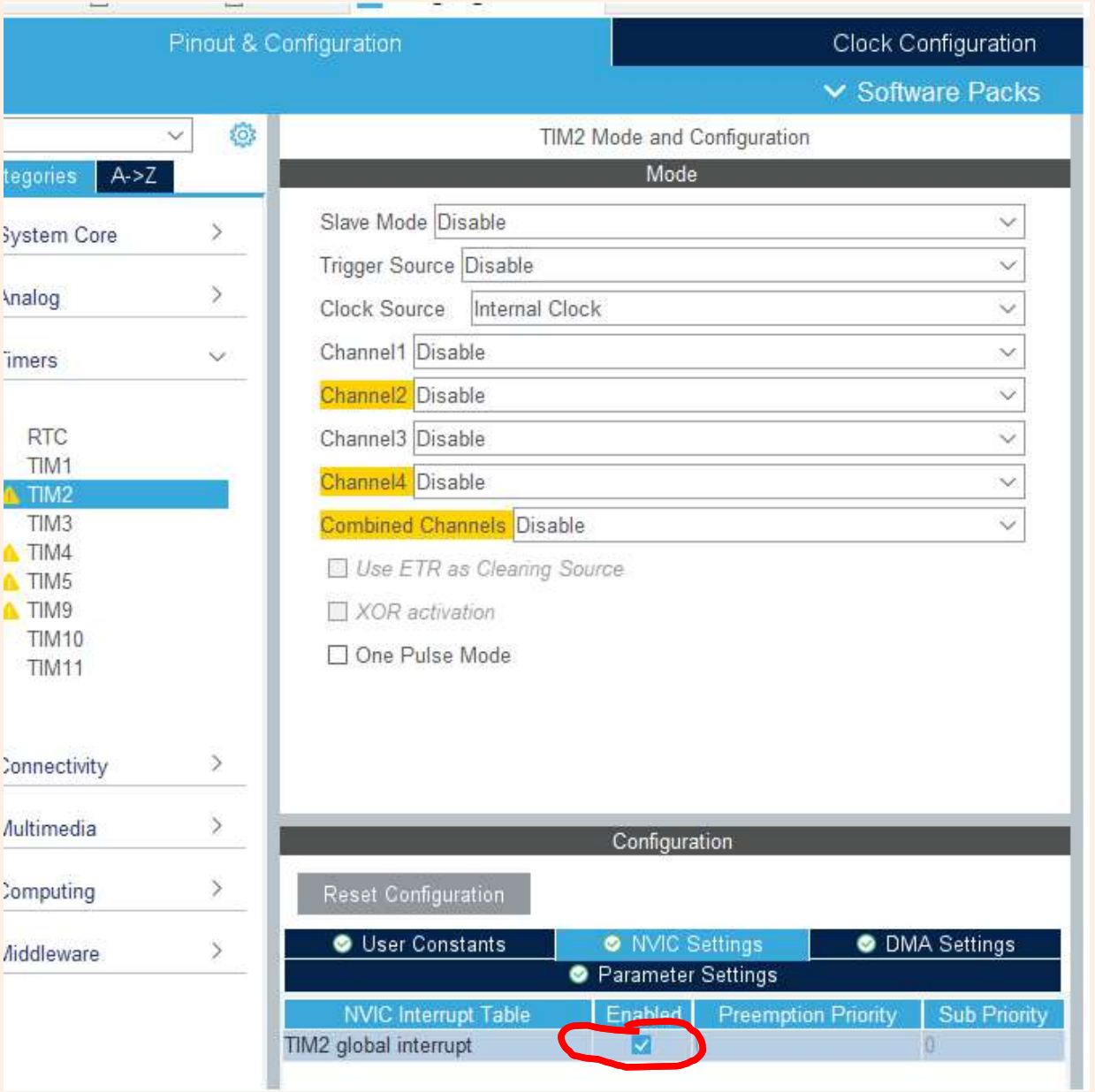
O que são:

São sinais causados por determinados eventos que interrompem o normal funcionamento de um programa e forçam a execução de uma rotina especial associada ao referido evento. São tipicamente mais eficientes do que polling!



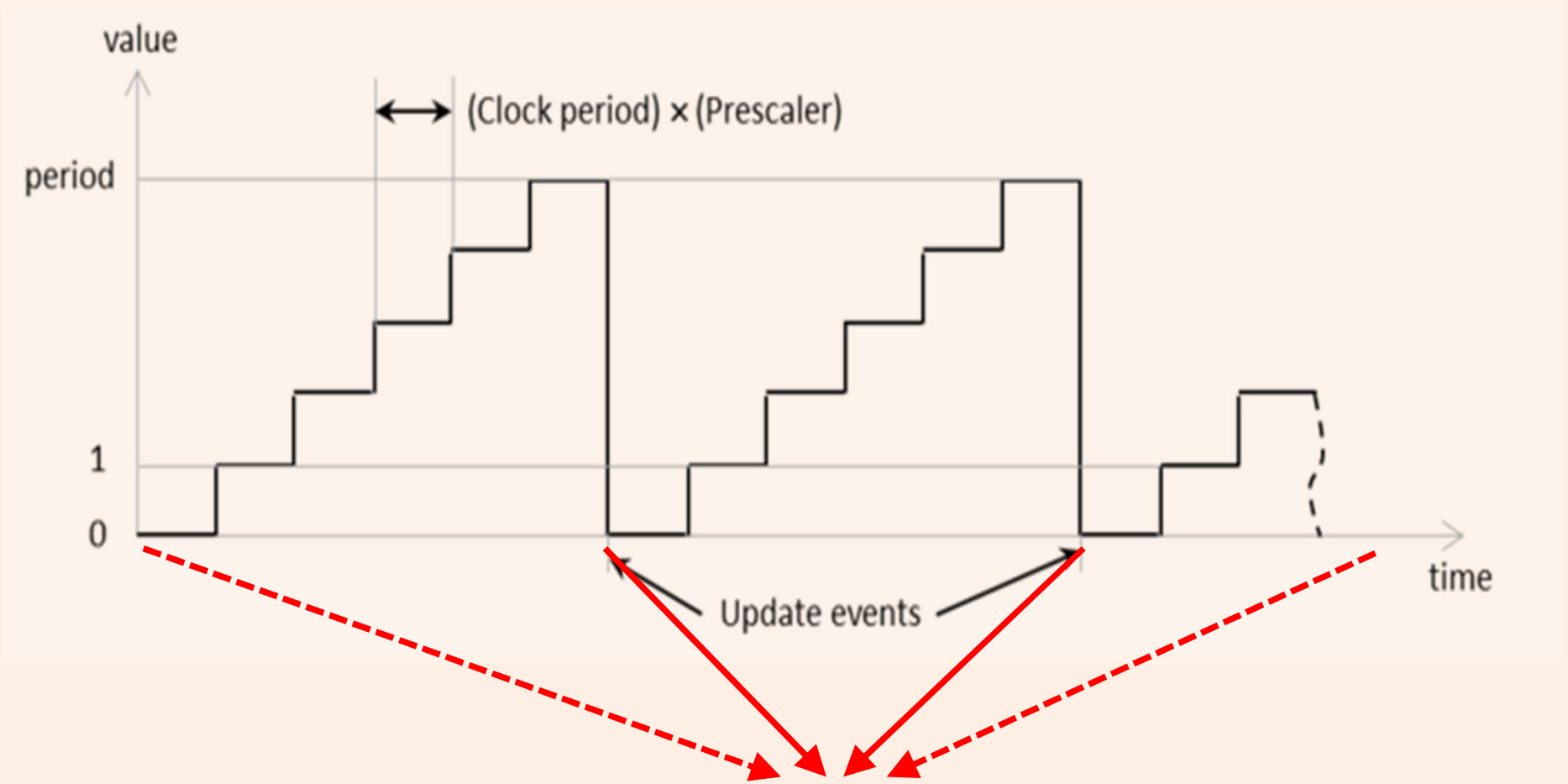
Interrupts

Configuração:



Interrupts

Configuração:



Nestes instantes são gerados interrupts!

Interrupts

Configuração: Adicionar este Código no ficheiro main.c. Recomenda-se que seja colocado entre: “/*USER CODE BEGIN 4 */” e “/*USER CODE END 4 */”

```
/* USER CODE BEGIN 4 */

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
   if (htim->Instance==TIM2)
   {
      HAL_UART_Transmit(&huart2, (uint8_t *)"Inside_ISR", strlen("Inside_ISR"), 0xFFFF);
      TIMER2_FLAG= 1;
   }
}

/* USER CODE END 4 */
```

Interrupts

Configuração: não esquecer de declarar a variável TIMER2_FLAG como variável global, tipicamente na zona das “Private variables”.

```
/* Private variables ---------------------------------------------------------*/

/* USER CODE BEGIN PV */

char TIMER2_FLAG= 0;

/* USER CODE END PV */
```

Configuração: não esquecer iniciar o timer e suas interrupções antes do loop infinito, por ex. entre “/* USER CODE BEGIN 2 */” e “/* USER CODE BEGIN 2 */”.

```
 MX_TIM2_Init();
 /* USER CODE BEGIN 2 */

 HAL_TIM_Base_MspInit(&htim2);
 HAL_TIM_Base_Start_IT(&htim2);

 /* USER CODE END 2 */
```