

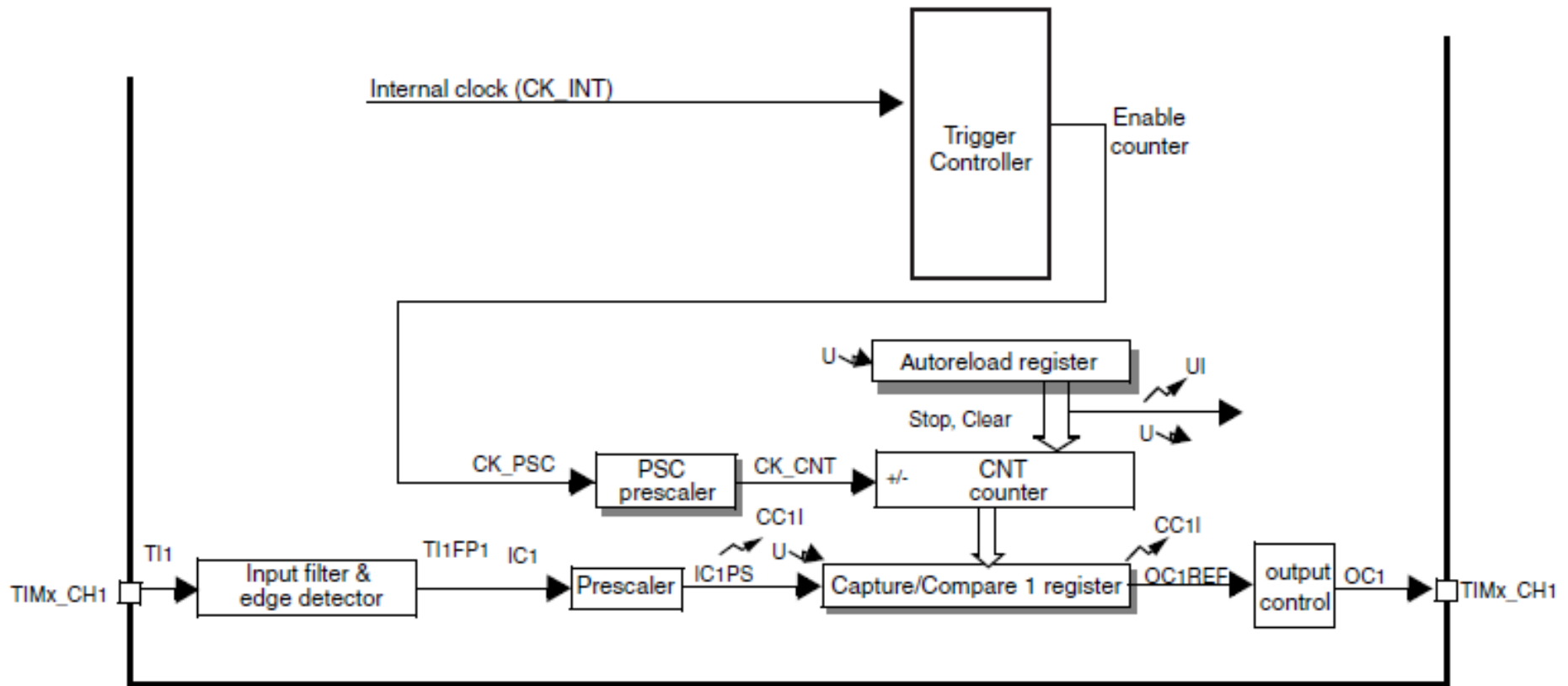
# STM32F4xx Timers

# Temporizadores / Contadores

- Temporizadores / contadores são periféricos destinados a realizar operações de contagem.
- Se a fonte de clock do contador é uma base de tempo conhecida, dizemos que opera no modo temporizador, ou seja, mede tempo.
- Além de contar eventos, medir passagem de tempo, temporizar eventos, os módulos temporizadores também podem ser utilizados para gerar sinais, periódicos ou não.
- O STM32F411 possui diferentes conjuntos de periféricos temporizadores, relacionados na tabela a seguir.

Timer	Tipo	Características	Resolução	Canais captura compara	Transf Via DMA	Máx clock int	A P B
TIM 1	Avançado	Saidas complement. tempo morto Sincronização Sens. Quadrat e Hall	16 bits Up / down updown	4	Sim	100	2
TIM2 TIM5	Uso geral	Sincronização Sens. Quadrat e Hall	32 bits Up / down updown	4	Sim	50	1
TIM3 TIM4	Uso geral	Sincronização Sens. Quadrat e Hall	16 bits Up / down updown	4	Sim	50	1
TIM9	Uso geral	Sincronização Sens. Quadrat e Hall	16 bits up	2	Não	100	2
TIM10 TIM11	Uso geral	Temporizador Auto-recarga com prescaler	16 bits up	1	Não	100	2

# TIM10 / 11 – Arquitetura temporizador básico



# Estrutura registradores periféricos TIMx

**typedef struct**

```
{
    __IO uint32_t CR1;          /*!< TIM control register 1,          */
    __IO uint32_t CR2;          /*!< TIM control register 2,          */
    __IO uint32_t SMCR;         /*!< TIM slave mode control register,  */
    __IO uint32_t DIER;         /*!< TIM DMA/interrupt enable register,*/
    __IO uint32_t SR;           /*!< TIM status register,             */
    __IO uint32_t EGR;          /*!< TIM event generation register,    */
    __IO uint32_t CCMR1;        /*!< TIM capture/compare mode register 1,*/
    __IO uint32_t CCMR2;        /*!< TIM capture/compare mode register 2,*/
    __IO uint32_t CCER;         /*!< TIM capture/compare enable register,*/
    __IO uint32_t CNT;          /*!< TIM counter register,            */
    __IO uint32_t PSC;          /*!< TIM prescaler,                  */
    __IO uint32_t ARR;          /*!< TIM auto-reload register,         */
    __IO uint32_t RCR;          /*!< TIM repetition counter register,  */
    __IO uint32_t CCR1;         /*!< TIM capture/compare register 1,    */
    __IO uint32_t CCR2;         /*!< TIM capture/compare register 2,    */
    __IO uint32_t CCR3;         /*!< TIM capture/compare register 3,    */
    __IO uint32_t CCR4;         /*!< TIM capture/compare register 4,    */
    __IO uint32_t BDTR;         /*!< TIM break and dead-time register,  */
    __IO uint32_t DCR;          /*!< TIM DMA control register,         */
    __IO uint32_t DMAR;         /*!< TIM DMA address for full transfer, */
    __IO uint32_t OR;           /*!< TIM option register,             */
} TIM_TypeDef;
```

# TIM10 e 11 – Operação

Opera somente com o clock interno

CK\_INT padrão = 16 MHZ (pode ser configurado)

Prescaler (**TIMx→PSC**) divide a frequência de CK\_INT por um valor inteiro entre 1 e 65536

A contagem é realizada pelo registrador **TIMx→CNT** até o valor definido pelo registrador de auto-reload (**TIMx→ARR**), em operação no modo UP.

Se operar no modo DOWN, a contagem inicia no valor **TIMx→ARR** e o estouro ocorre em zero.

O reinício do TIMx é sinalizado pelo bit UIF do registrador status (**TIM→SR**)

$$T_{\text{reinício}} = t_{\text{CK\_INT}} * (\text{TIMx} \rightarrow \text{PSC} + 1) * (\text{TIMx} \rightarrow \text{ARR} + 1)$$

# TIMx->CR1 – Control Register 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD[1:0]		ARPE	Reserved				URS	UDIS	CEN
						rw	rw	rw					rw	rw	rw

**Valor no RESET : 0x00**

**CEN** (Counter Enable)– Habilita a operação do contador (0 – desliga 1- liga)

**UDIS** (Update Disable) - Desabilita evento de update (reset contador + atualização dos registradores sombra do PSC e ARR). Contador pode ser reiniciado pelo bit UG.

**URS** (Update Request Source ) - Seleciona geração do evento de update

0 – Update gerado pelo estouro do contador e pelo bit UG

1 – Update gerado somente pelo estouro do contador

**ARPE** (Auto-reload preload enable) – Quando em 1, TIMx\_ARR utiliza registrador sombra, que é atualizado no evento de update.

**CKD[1:0]** – Clock dos filtros digitais dos canais captura

00 – CK\_INT | 00 – 2\*CK\_INT | 10 – 4\*CK\_INT | 11 – reservado

# TIMx->SR – Status Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CC1OF	Reserved						CC1IF	UIF	
						rc_w0							rc_w0		

**Valor no RESET : 0x00**

Os bits deste registrador são **ligados pelo hardware** e **devem ser desligados pelo programa**, para sinalizar reconhecimento do evento.

**UIF** (Update interrupt flag) – sinaliza que ocorreu estouro do contador ou reinício através do bit UG

**CC1IF** (Capture/compare 1 interrupt flag) – sinaliza evento de captura ou comparação no canal 1

**CC1OF** (Capture/compare 1 overcapture flag) – ocorreu sobreescrita no registrador CCR1



# TIMx->EGR – Event Generator Register

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CC1G	UG
														W	W

**Valor no RESET : 0x00**

Estes bits são ligados por software e automaticamente desligados por hardware

**UG** (Update generate) – zera PSC e CNT. Atualiza registradores sombra.

**CC1G** (Capture/compare generate) – gera evento no canal CC1

# RCC->APB2ENR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											SPI5EN	Reser- ved	TIM11 EN	TIM10 EN	TIM9 EN
											rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reser- ved	SYSCF G EN	SPI4EN	SPI1 EN	SDIO EN	Reserved		ADC1 EN	Reserved		USART6 EN	USART1 EN	Reserved		TIM1 EN	
	rw	rw	rw	rw			rw			rw	rw				

**Valor no RESET : 0x00**

Habilita o clock para os seguintes periféricos :

TIM 1, 9, 10, 11

USART 1 e 6

ADC1

SPI1, 4 e 5

Além disso permite a configuração do sistema de clock por software

# TIM9 – Arquitetura temporizador trigger

