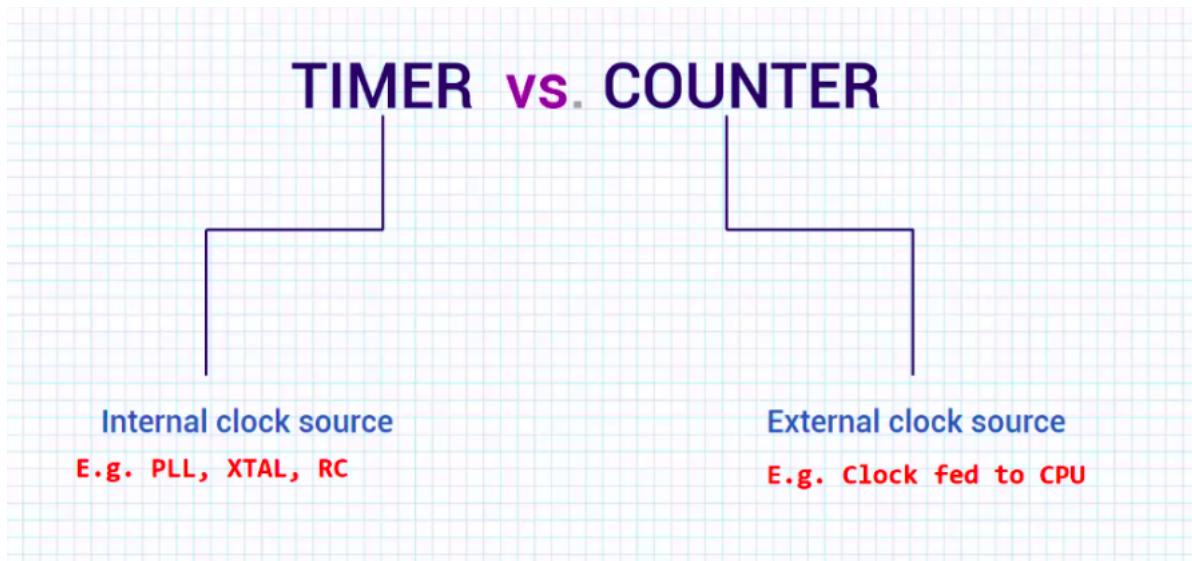


- Counting Events
- Creating Delays
- Measuring time between event



TIMER-STM32 Timers

- Can be used as time base generator
- Can be used to measure the frequency of an external event - *Input Capture Mode*
- Control an output waveform, or to indicate when a period of time has elapsed - *Output Compare Mode*
- *One pulse mode (OPM)*- allows the counter to be started in response to a stimulus and to generate a pulse with a programmable length after a programmable delay

TIMER-Registers

- **Timer Count Register (TIMx_CNT)**

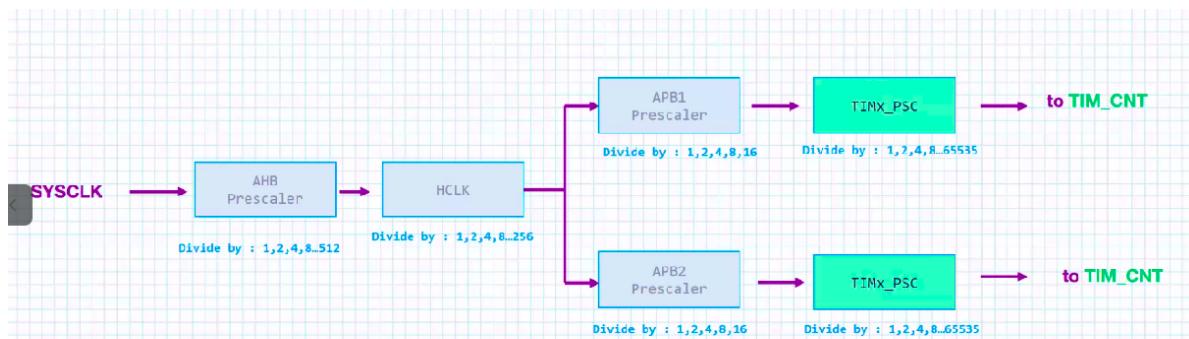
Shows the current counter value. Size could be 32-bit or 16-bit depending on timer module used.

- **Timer Auto-Reload Register (TIMx_ARR)**

Timer raises a flag and the counter restarts automatically when counter value reaches the value in the auto-reload register. The counter is an up counter by default but can also be configured to be a down counter.

- **Timer Prescaler Register (TIMx_PSC)**

The prescaler slows down the counting speed of the timer by dividing the input clock of the timer



- Timer prescaler (TIMx_PSC) determines how fast the timer counter(TIMx_CNT) increases/decreases.
- With each change in the counter(TIMx_CNT) value, the new value is compared to the value in the timer auto-reload register (TIM_ARR), when the values match, a flag is raised and an interrupt occurs.

TIMER- Some Terms

- **Update Event**

When timeout occurs or how long it takes for flag to be raised

- **Period**

Value loaded into auto-reload register(TIM_ARR)

- **Up counter**

Counts from zero to a set value.

- **Down counter**

Counts from a set value down to zero

$$\text{Update Event} = \frac{\text{Timer}_{\text{clock}}}{(\text{Prescaler}+1)(\text{Period}+1)}$$

Example

Let

Timer clock = APB1 clock = 48MHz
Prescaler = TIM_PSC value = 47999 + 1
Period = TIM_ARR value = 499 + 1

$$\text{Update Event} = \frac{48\,000\,000}{(47999+1)(499+1)} = 2\text{Hz} = \frac{1}{2}\text{s} = 0.5\text{s}$$

- **Prescaler (PSC)**

- *Prescaler value is put here*

EXAMPLE

```
TIM2->PSC = 1600 -1 ; // Set prescaler value to 1600
```

- **Auto-Rewload Register (ARR)**

- *Auto-reload value is put here*

EXAMPLE

```
TIM2->ARR = 10000 ; // Set prescaler value to 1600
```

TIM-Registers

- **Control Register 1 (CR1)**

- *Enabling and disabling timer*

EXAMPLE

```
TIM2->CR1 = 1 ; // Enable timer2
```

TIM-Registers

- **Status Register (SR)**

- *Checking, setting and clearing the flags of the timer*

EXAMPLE

```
TIM2->SR & 1 ; // Check update interrupt flag  
TIM2->SR &= ~1 ; // Clear update interrupt flag
```

- **Capture/Compare Register (CCR1, CCR2
CCR3, CCR4)**

- One capture/compare register for each of the 4 channels

EXAMPLE

```
timestamp =TIM2->CCR1 ; // read captured value
```

TIM-Registers

- **Capture Compare Mode Register 1 (CCMR1)**

- Configuring capture/compare functionality for CH1 and CH2

- **Capture Compare Mode Register 2 (CCMR2)**

- Configuring capture/compare functionality for CH3 and CH4

EXAMPLE

```
TIM2->CCMR1 = 0x41 ; // set CH1 to capture at every edge
```

- **Capture/Compare Enable Register (CCER)**

- *Used to enable any of the timer channels either as input capture or output compare*

EXAMPLE

```
TIM2->CCER = 1 ; // Enable channel 1
```