BME3321:Introduction to Microcontroller Programming

Topic 7: Usage of Timers in ARM Cortex MCUs

Assist. Prof. Dr. İsmail Cantürk

Mastering STM32 (ch11)

STM32F407 reference manual (ch 17,18,19,20)

Timers with external clock sources

Timers cannot work without a clock source, because it is used to increment the counter register.

The path highlighted in red is used to feed the timer when the clock of the bus is selected as source: the internal clock CK_INT feeds the Prescaler (TIMx_PSC), which in turn determines how fast the Counter Register (TIMx_CNT) is increased/decreased.

TIMx_CNT is compared with the content of the auto-reload register. When they match, the UEV is generated, and the corresponding IRQ is fired, if enabled.

A timer can have its clock frequency or starting signal from other sources rather than CK_INT. These can be internal or external sources

Timers with external clock sources

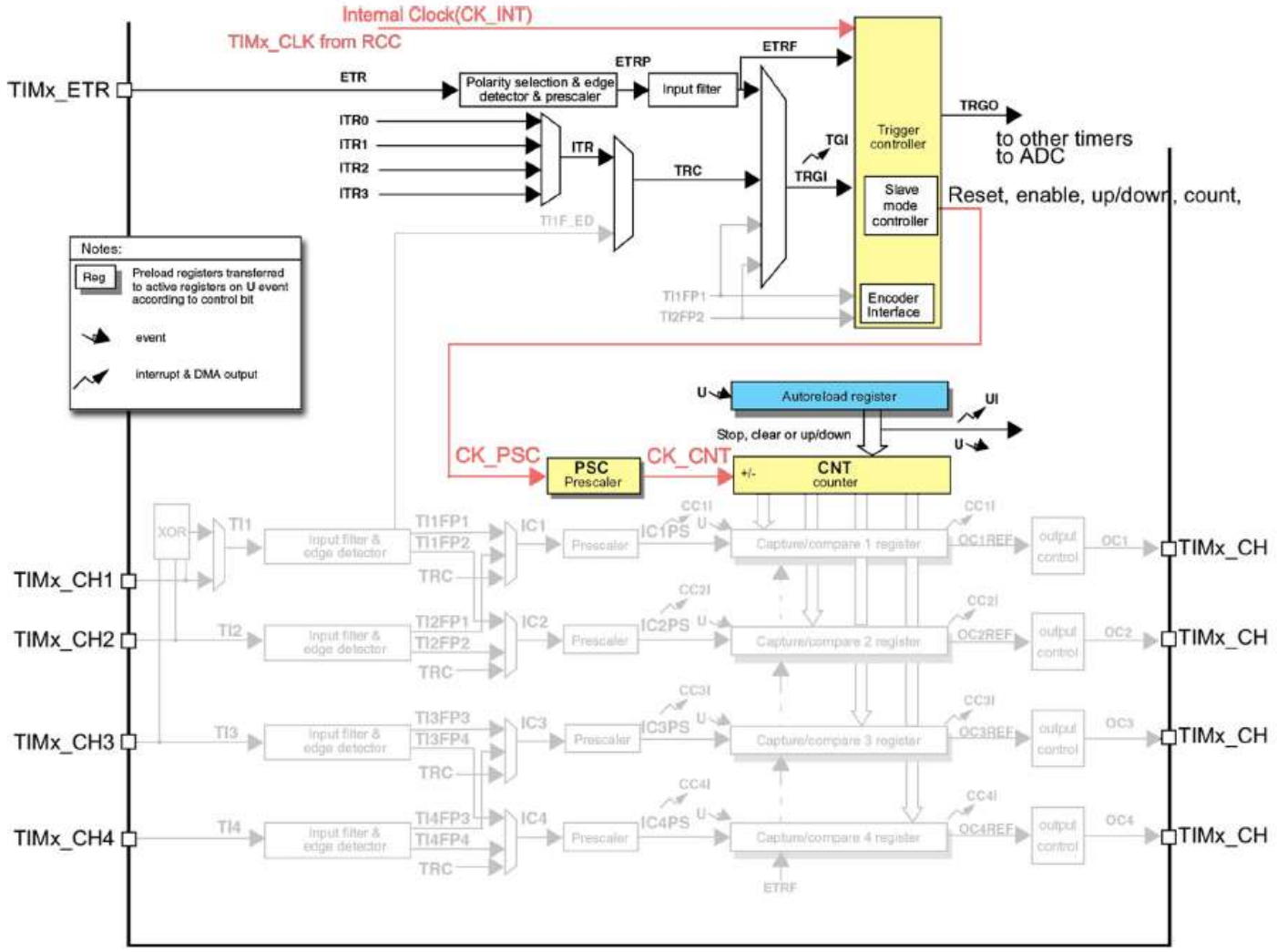


Figure 3: The structure of a *general purpose* timer

Timers with external clock sources

We can see that a timer can receive “stimuluses” from other sources. These can be divided in two main groups:

- Clock sources, which are used to clock the timer. They can come from external sources connected to the MCU pins or from other timers connected internally to the MCU. Keep in mind that a timer cannot work without a clock source, because this is used to increment the counter register.

- Trigger sources, which are used to synchronize the timer with external sources connected to the MCU pins or with other timers connected internally. For example, a timer can be configured to start counting when an external event triggers it.

  In this case the timer is clocked by another clock source and it is controlled by another device.

Timers with external clock sources

Once a timer operates in master mode it can feed another timer configured in slave mode through a dedicated output line, called Trigger Output (TRGO), connected to the internal dedicated lines called ITR0, ITR1, ITR2 and ITR3. The master timer can both provide the clock source or trigger the slave timer.

These Internal Trigger (ITR) lines (ITR0, ITR1, ITR2 and ITR3) are precisely internal to the chip, and each line is hardwired between two defined timers.

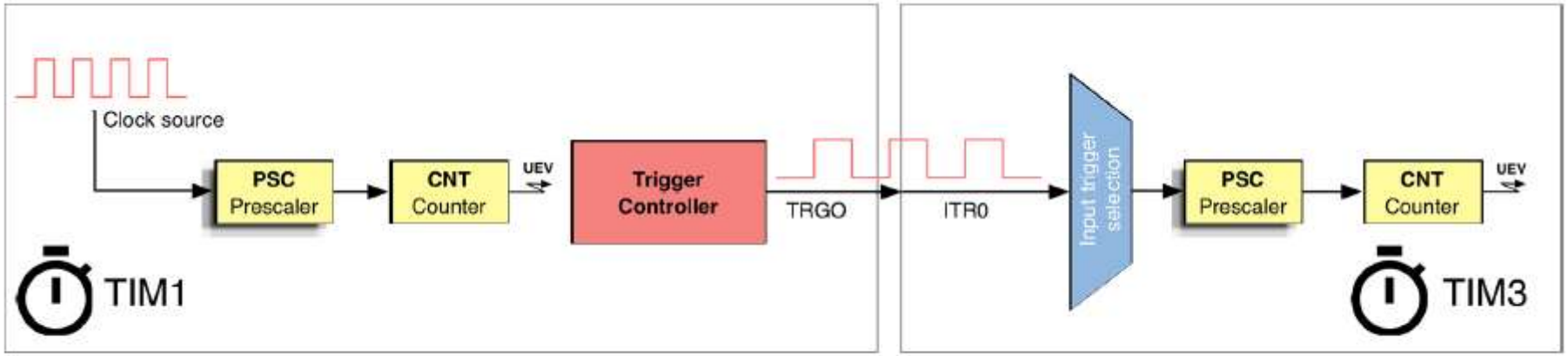


Figure 9: The TIM1 can fed the TIM2 timer through the ITR0 line

Timers with external clock sources

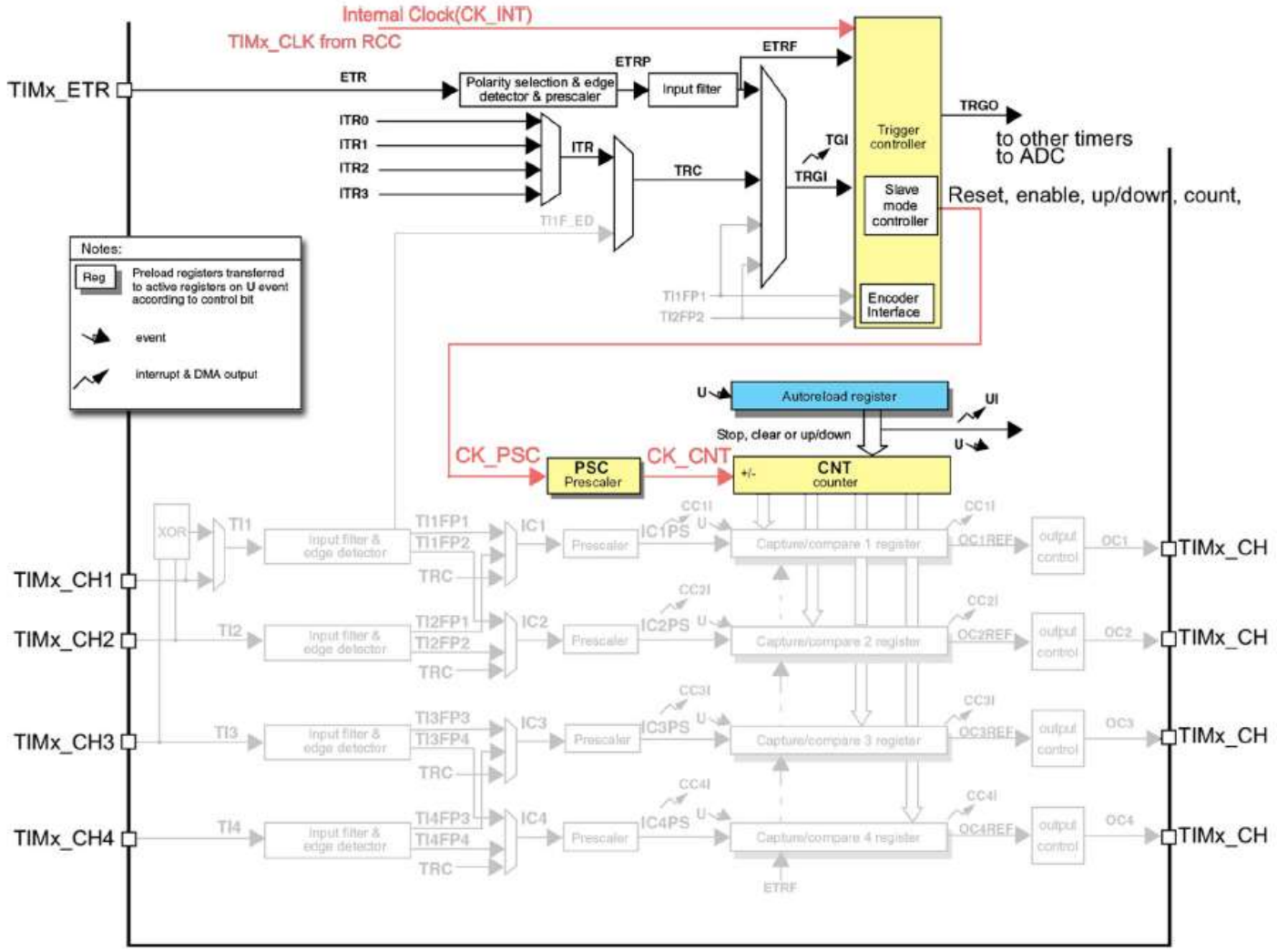


Figure 3: The structure of a *general purpose* timer

Timing for UEV of Timer3

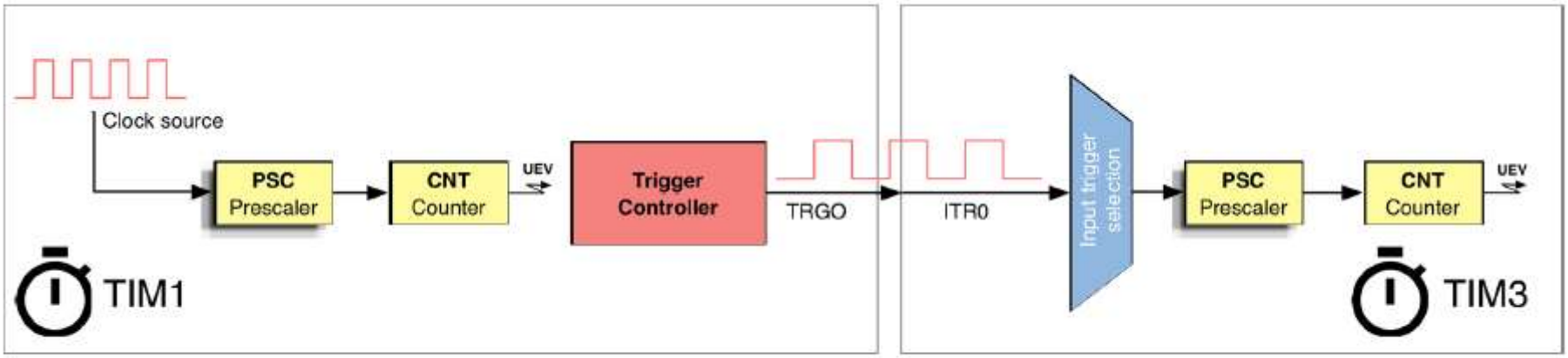
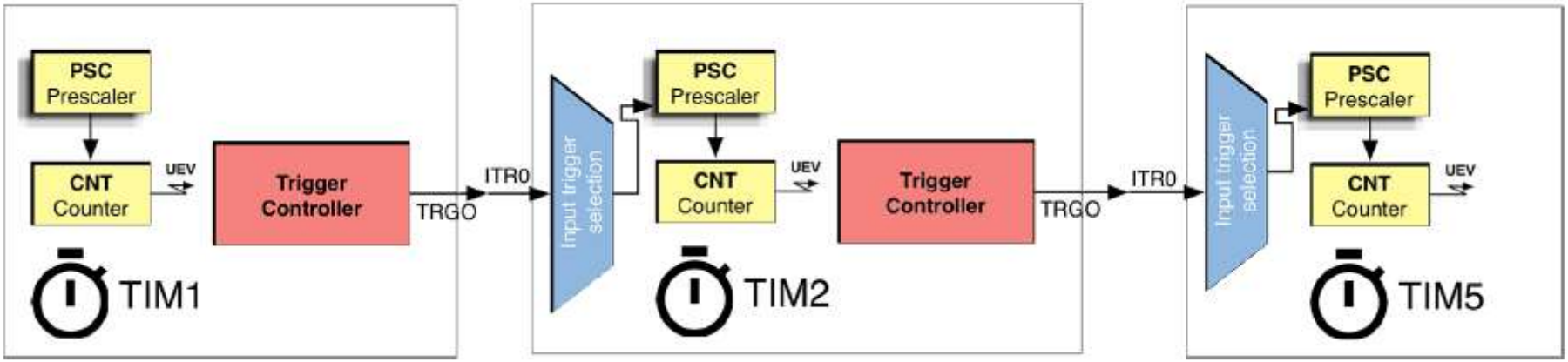


Figure 9: The TIM1 can fed the TIM2 timer through the ITR0 line

$$UpdateEvent\ for\ TIM3 = \frac{CK\_INT}{\underbrace{(Prescaler + 1)(Period + 1)}_{\text{for TIM1}}\underbrace{(Prescaler + 1)(Period + 1)}_{\text{for TIM3}}}$$

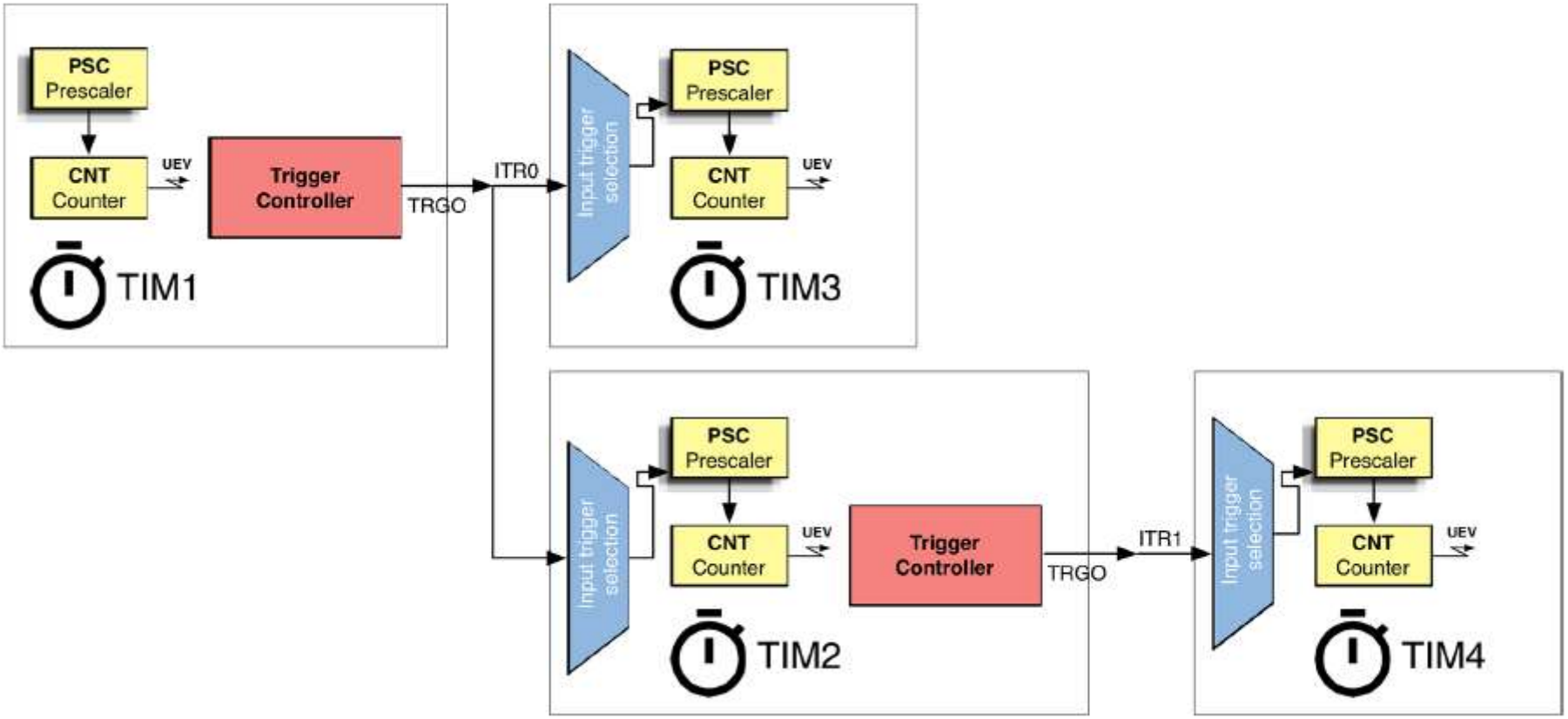
Timers with external clock sources

A timer configured as slave can also simultaneously act as master for another timer, allowing to create complex networks of timers.



Timers with external clock sources

Timers can form different structures using combinations of master/slave modes.



Timers applications

General purpose timers have not been designed to be used only as timebase generators.

General purpose timers offer much more advanced capabilities, which can be used to drive other important time-related activities.

These are:

Pulse width generation

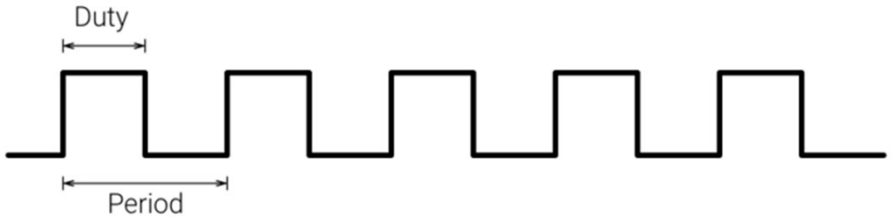
Input capture mode,

Output compare mode,

.
.
.

Pulse-Width Generation with Timers (Output compare mode)

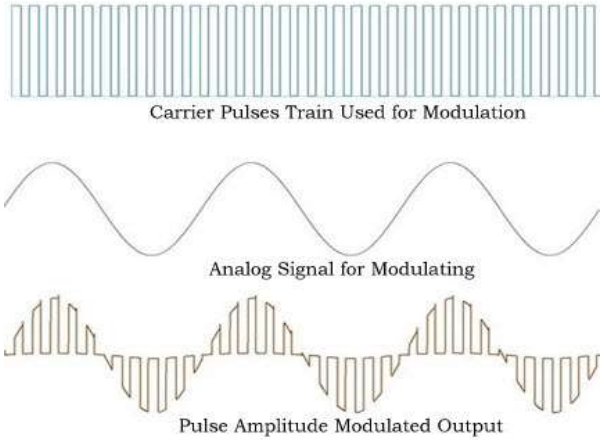
Pulse-width modulation (PWM) is a technique used to generate pulses with different duty cycles for a given period of time.



PWM has many applications, but all of them can be grouped in two main categories:

- control the output voltage (and hence the current),

- encoding a message (i.e., modulation).



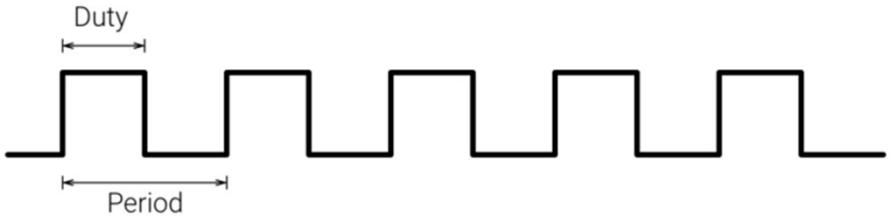
Pulse-Width Generation with Timers

The square waves until now (i.e., clock signals) have all one common characteristic: they have a T_{ON} period equal to the T_{OFF} one. For this reason they are also said to have a 50% duty cycle.

A duty cycle is the percentage of one period of time in which a signal is active.

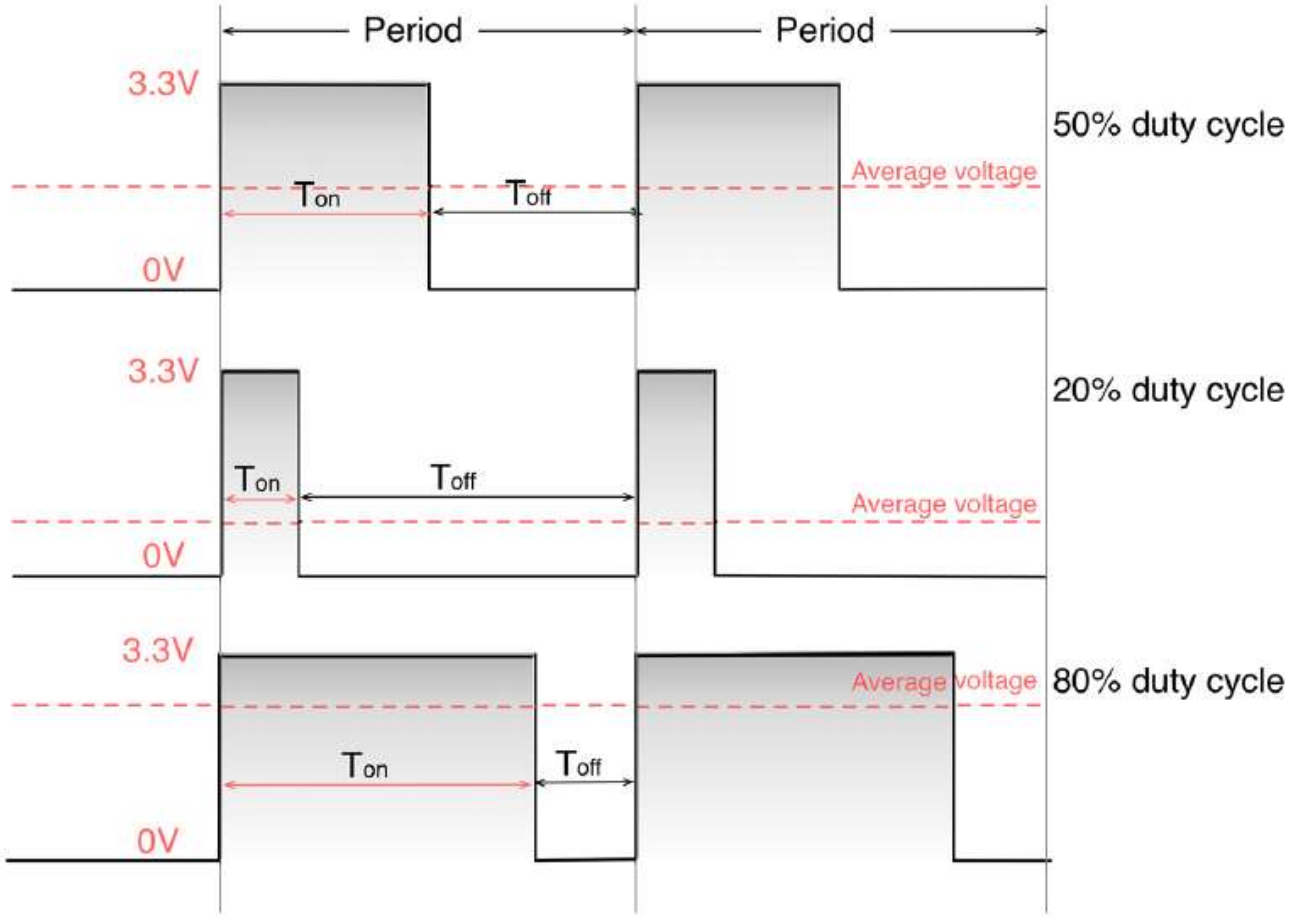
As a formula, a duty cycle is expressed as:

$$D = \frac{T_{ON}}{Period} x100$$



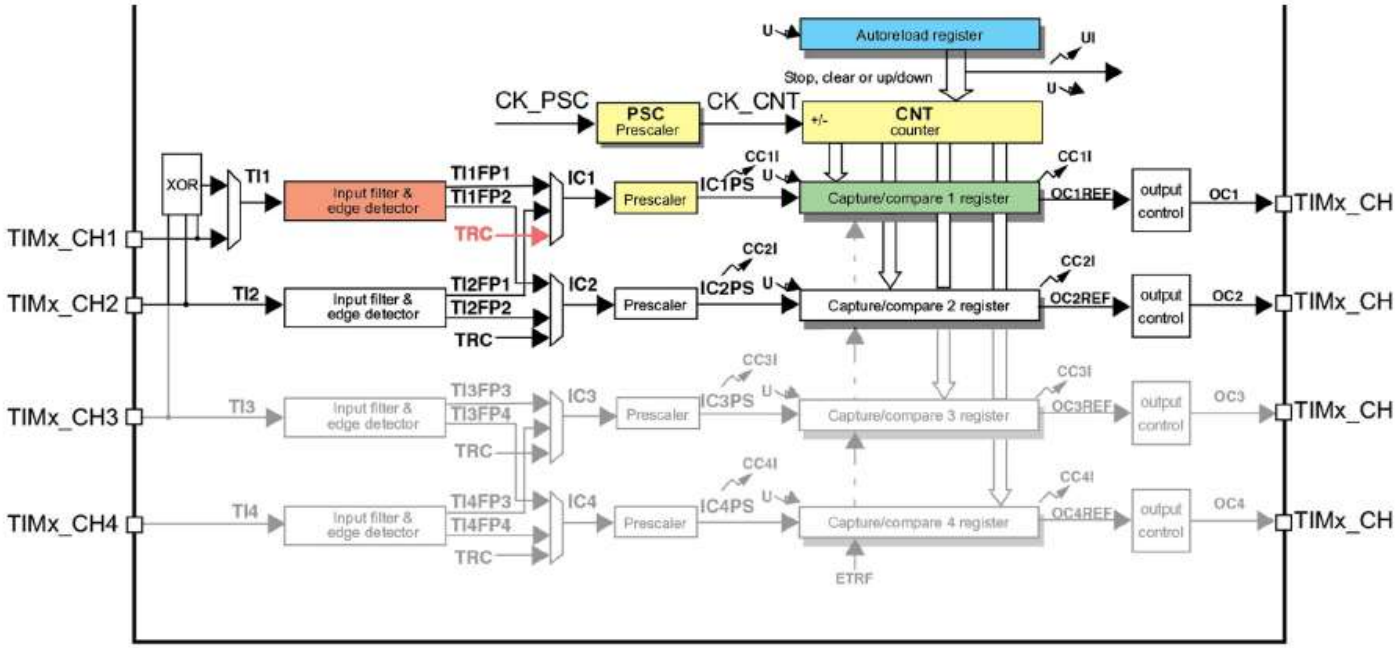
where D is the duty cycle, T_{ON} is the time the signal is active. Thus, a 50% duty cycle means the signal is on 50% of the time but off 50% of the time.

Pulse-Width Generation with Timers



Pulse-Width Generation with Timers

Capture/compare register is used to adjust duty cycle.



Capture/compare register

19.5.10 TIM10/11/13/14 capture/compare register 1 (TIMx_CCR1)

Address offset: 0x34

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw/ro	rw/ro	rw/ro	rw/ro	rw/ro	rw/ro	rw/ro	rw/ro	rw/ro	rw/ro	rw/ro	rw/ro	rw/ro	rw/ro	rw/ro	rw/ro

Bits 15:0 **CCR1[15:0]**: Capture/Compare 1 value

If channel CC1 is configured as output:
CCR1 is the value to be loaded in the actual capture/compare 1 register (preload value).

It is loaded permanently if the preload feature is not selected in the TIMx_CCMR1 register (bit OC1PE). Else the preload value is copied in the active capture/compare 1 register when an update event occurs.

The active capture/compare register contains the value to be compared to the counter TIMx_CNT and signaled on OC1 output.

If channel CC1is configured as input:
CCR1 is the counter value transferred by the last input capture 1 event (IC1). The TIMx_CCR1 register is read-only and cannot be programmed.

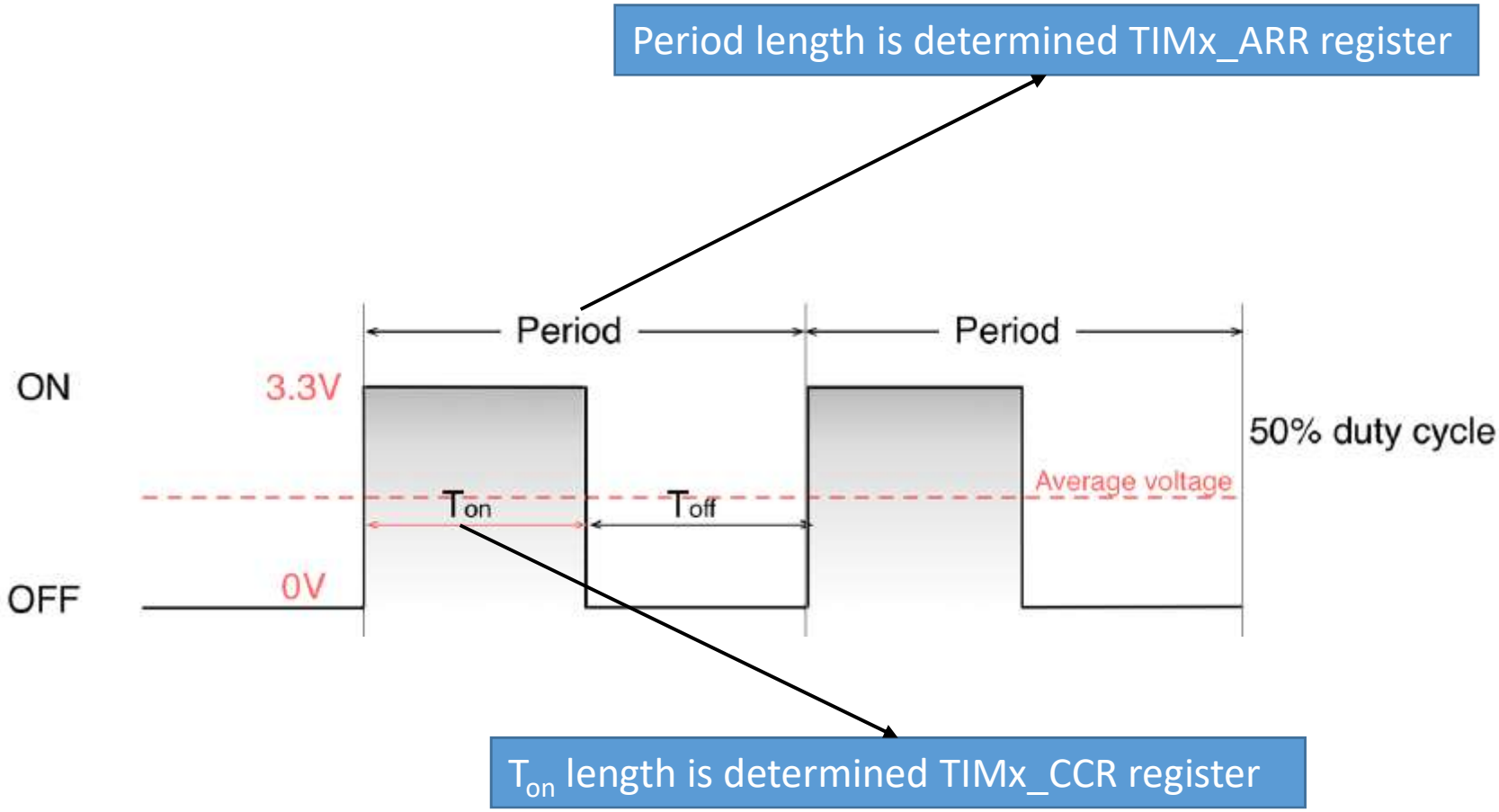
Pulse-Width Generation with Timers

```
.
.
.

HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);

.
.
.


while(1) {
.
.
.


TIM4->CCR1=100;
// __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_1, dutyCycle);

.
.
.
```

Timers can use predefined I/O pins as output channels

By setting value of CCR, we define T_{ON}.

Pulse-Width Generation with Timers



TIMx_CCR< TIMx_ARR

Generating a Sinusoidal Wave Using PWM

An output square wave generated with the PWM technique can be filtered with a Resistor-Capacitor low-pass filter

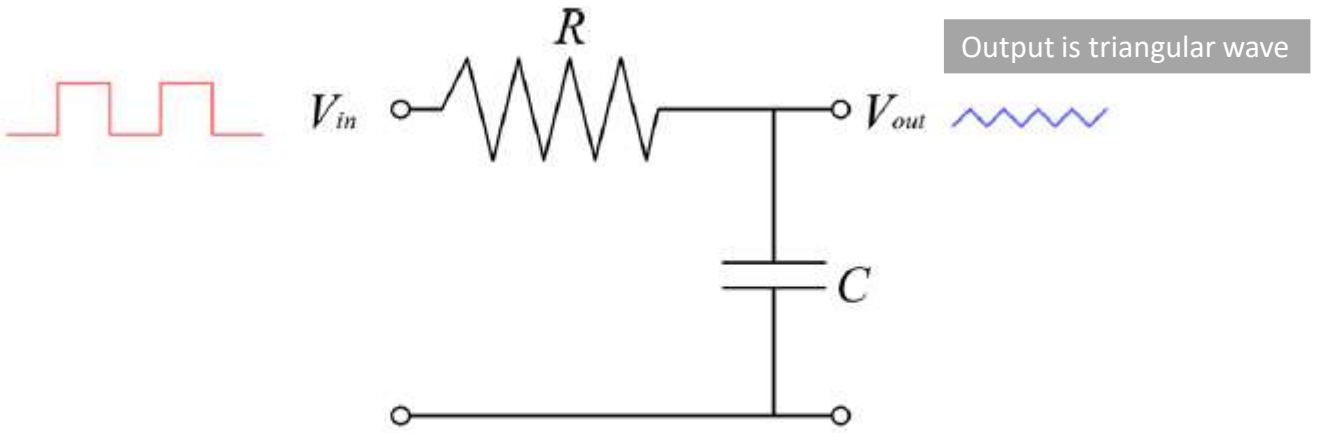


Figure 24: A typical low pass filter implemented with a resistor and a capscitor

Triangular wave → RC Low Pass filter → Sinusoidal wave

Input capture mode

The input capture mode offered by general purpose and advanced timers.

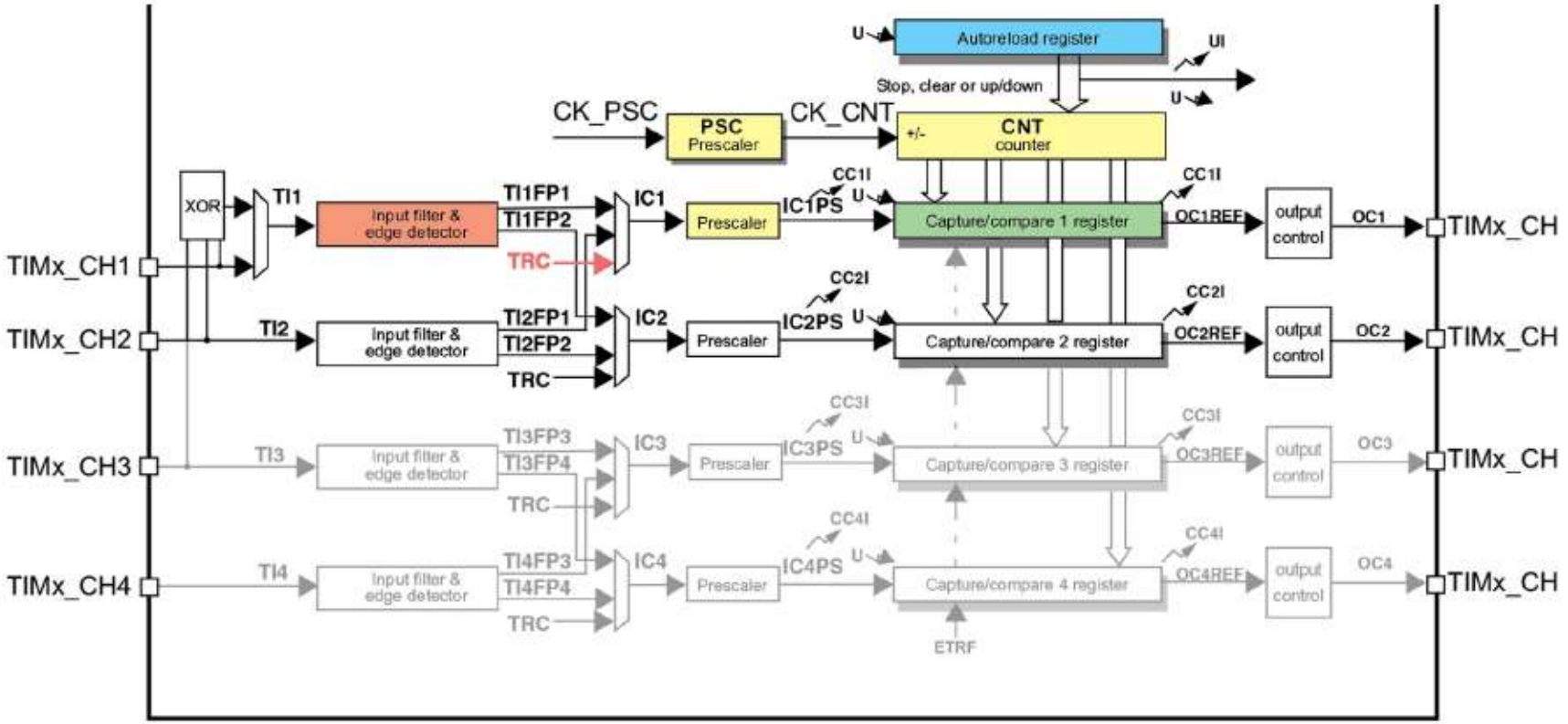
It allows to compute the frequency of external signals.

External signals can be applied to each one of the 4 channels of these timers.

And the capture is performed independently for each channel.

Input capture mode

As you can see, each input is connected to an edge detector, which is also equipped with a filter used to “debounce” the input signal. The output of the edge detector goes into a source multiplexer (IC1, IC2, etc.). Finally, a dedicated prescaler allows to “slow down” the frequency of the input signal.



Input capture mode

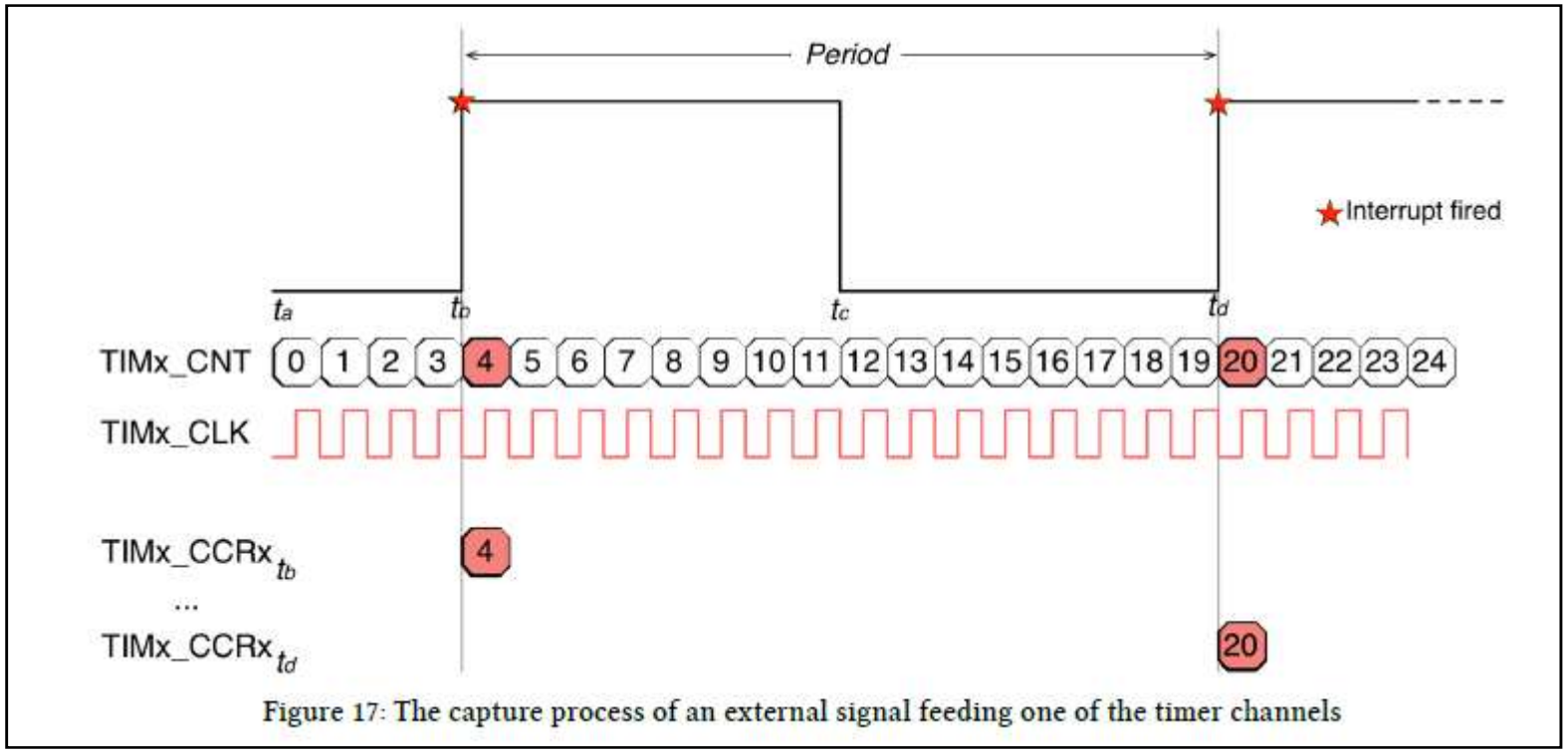


Figure 17: The capture process of an external signal feeding one of the timer channels

TIMx is a timer, configured to work at a given TIMx_CLK clock frequency. This means that it increments the TIMx_CNT register up to the Period value every 1/TIMx_CLK seconds.

Input capture mode

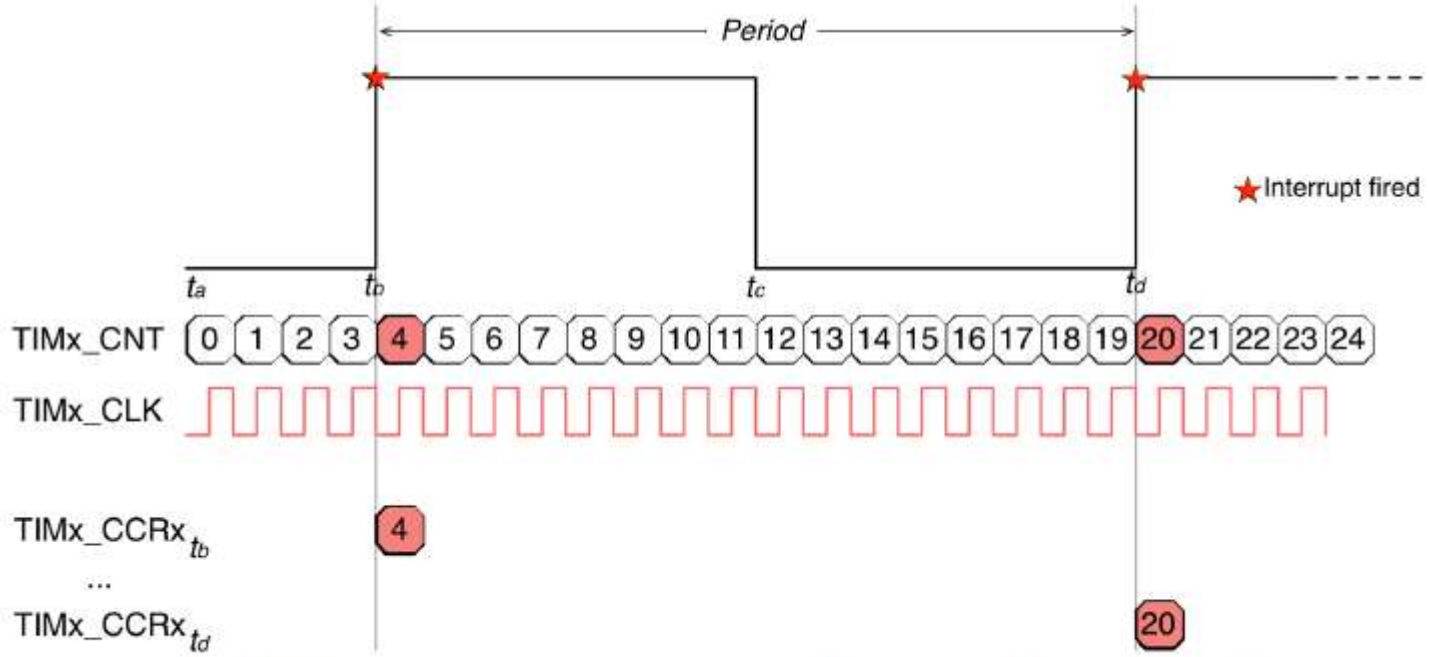


Figure 17: The capture process of an external signal feeding one of the timer channels

Supposing that we apply a square wave signal to one of the timer channels.
Supposing that we configure the timer to trigger at every rising edge of the input signal TIMx_CCRx register will be updated with the content of the TIMx_CNT register at every detected transition.
When this happens, the timer will generate a corresponding interrupt allowing to keep track of the counter value.

Input capture mode

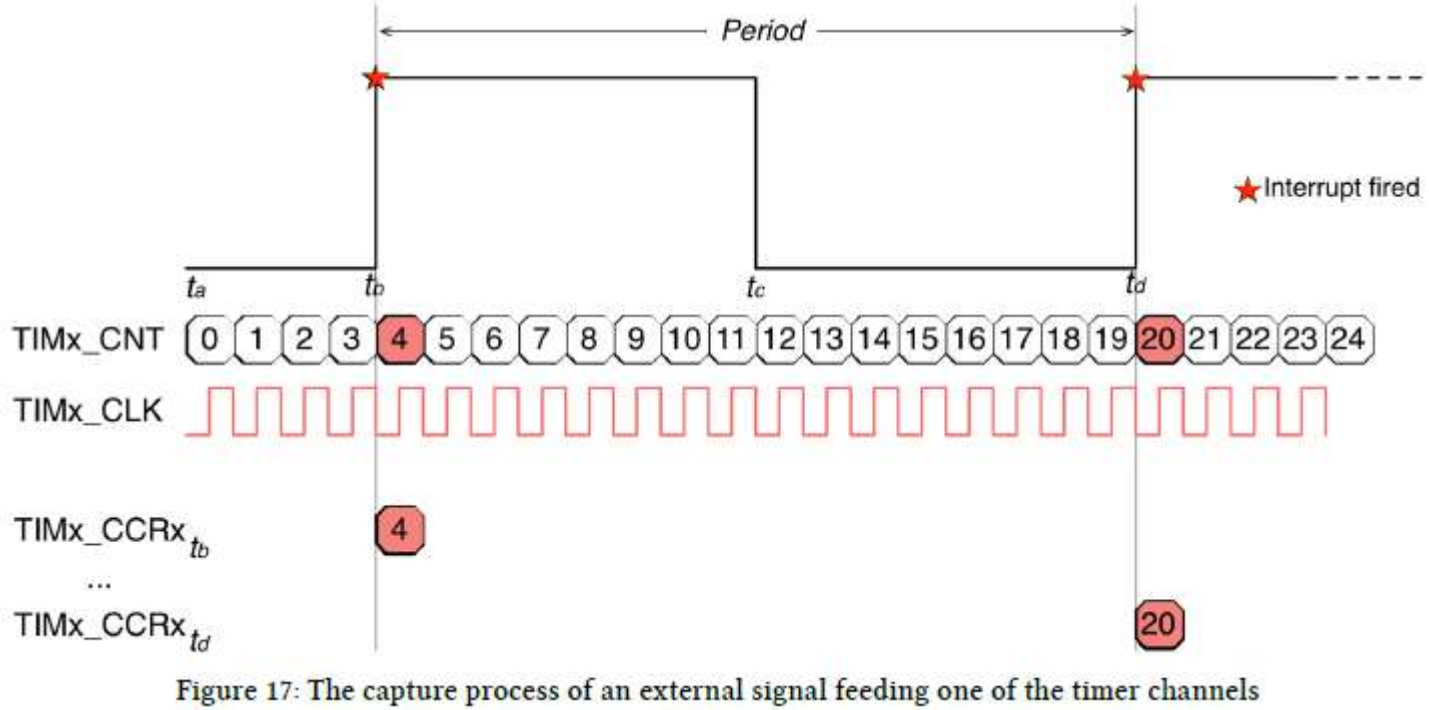


Figure 17: The capture process of an external signal feeding one of the timer channels

To get the external signal period, two consecutive captures are needed. The period is calculated by subtracting these two values, CNT0 and CNT1.

Input capture mode

$$Period = Capture.\left(\frac{TIMx\_CLK}{(Prescaler + 1)(CH_{Prescaler})(Polarity_{Index})}\right)^{-1}$$

$$Capture = CNT_1 - CNT_0 \; if \; CNT_1 > CNT_0$$

$$Capture = TIMx_{Period} - CNT_0 + CNT_1 \; if \; CNT_1 < CNT_0$$

$CH_{Prescaler}$ is a further prescaler that can be applied to the input channel and

$Polarity_{Index}$ is equal to 1 if the channel is configured to trigger on rising or falling edge of the input signal, or it is equal to 2 if both the edges are sampled.

Output compare mode (e.g., PWM)

The output compare is a mode offered by general purpose and advanced timers that allows to control the status of output channels when the channel compare register (TIMx_CCRx) matches with the timer counter register (TIMx_CNT).

There are six output compare modes available to programmers:

• Output compare timing: the comparison between the output compare register (CCRx) and the counter (CNT) has no effect on the output. This mode is used to generate a timing base.

• Output compare active: set the channel output to active level on match. The channel output is forced high when the counter (CNT) matches the capture/compare register (CCRx).

• Output compare inactive: set channel to inactive level on match. The channel output is forced low when the counter (CNT) matches the capture/compare register (CCRx).

• Output compare toggle: the channel output toggles when the counter (CNT) matches the capture/compare register (CCRx).

• Output compare forced active/inactive: the channel output is forced high (active mode) or low (inactive mode) independently from counter value.

Timer registers

Check timer registers from reference manual