PART 6: Embedded Programming STM32

Peripherals: timers, PWM

Where to find information

Three fundamental documents

- ✓ The Datasheet
  - Electrical data
  - Hardware Overview
  - Register Map
- ✓ The Reference Manual
  - Details of peripheral registers
  - EEPROM programming
  - flash programming
- ✓ The Programming Manual
  - Instruction set
  - Addressing Mode and pipeline execution

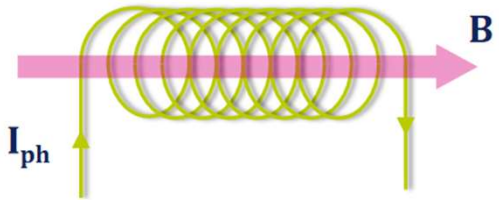
Timer Applications

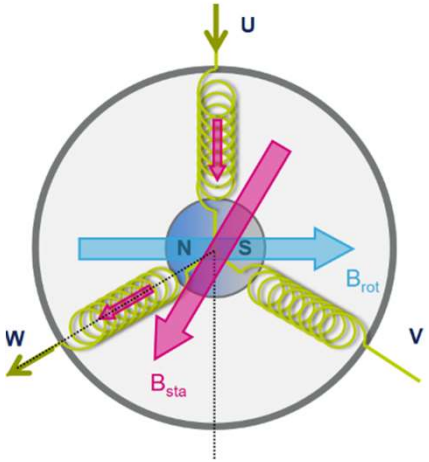
- ✓ Time based generation
- ✓ Measure the pulse lengths of input signal (input capture)
- ✓ Generate Output Wave Forms Signals (PWM, Output Compare and One-pulse mode)
- ✓ Interrupt capability on various events (capture, compare overflow)

Motor Control

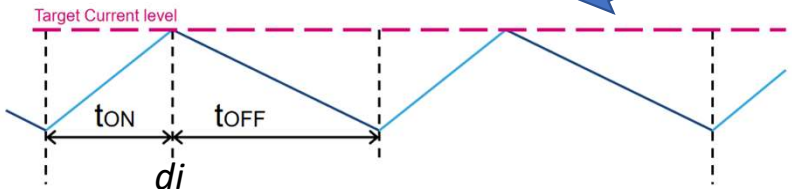
Magnet in a coil proportionnal to current

$$B = kI_{ph}$$





3-phase Moto: magnetic fields are created in coils to spin a magnet. Torque depends on magnetic field that depends on current



$$U=L\frac{di}{dt}$$

Timer as a PWM turns on and off some electronics to control current level

Timer thanks to PWM waveform controls speed of the motor

Timer Categories

Timer are split in 3 categories:

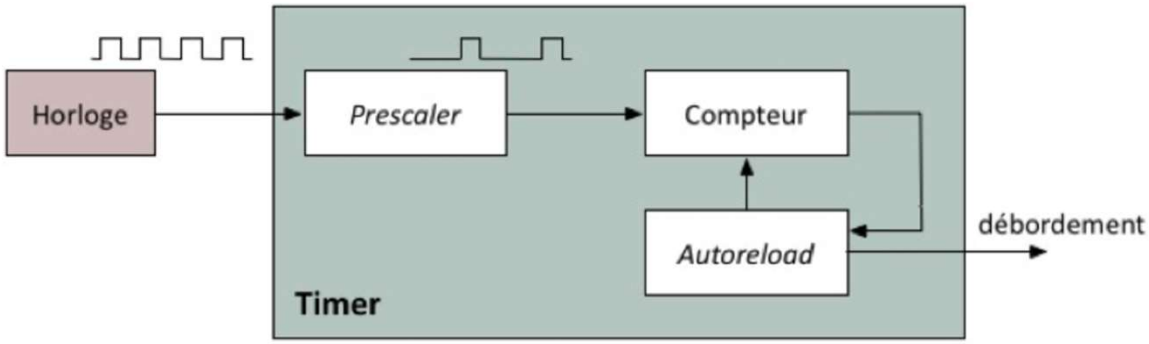
- ✓ Basic
- ✓ General Purpose
- ✓ Advanced

ISEN ALL IS DIGITAL! OUEST | yncréa

Timer Basics

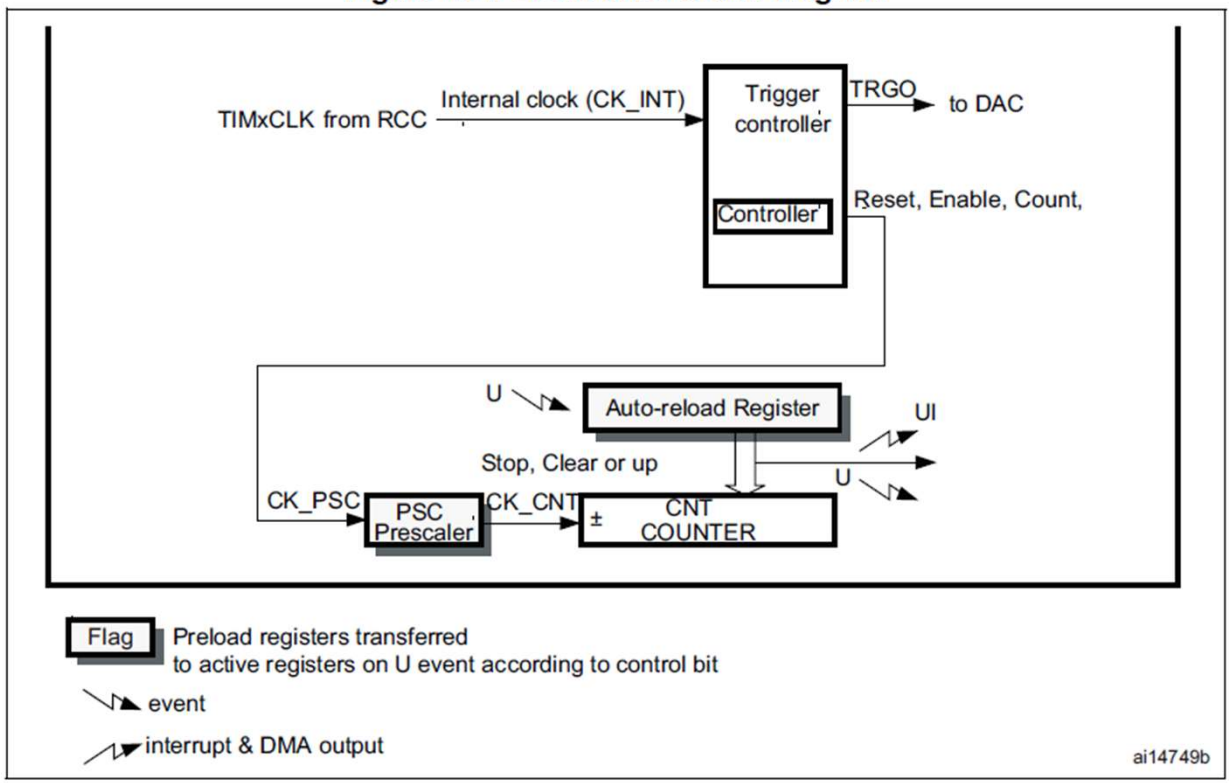
Any timer in any microcontroller has:

- ✓ Clock
- ✓ Prescaler to divide clock frequency
- ✓ Counter to store the number of clock tics
- ✓ Autoreload register to interact with the counter (compare timing, reload values)



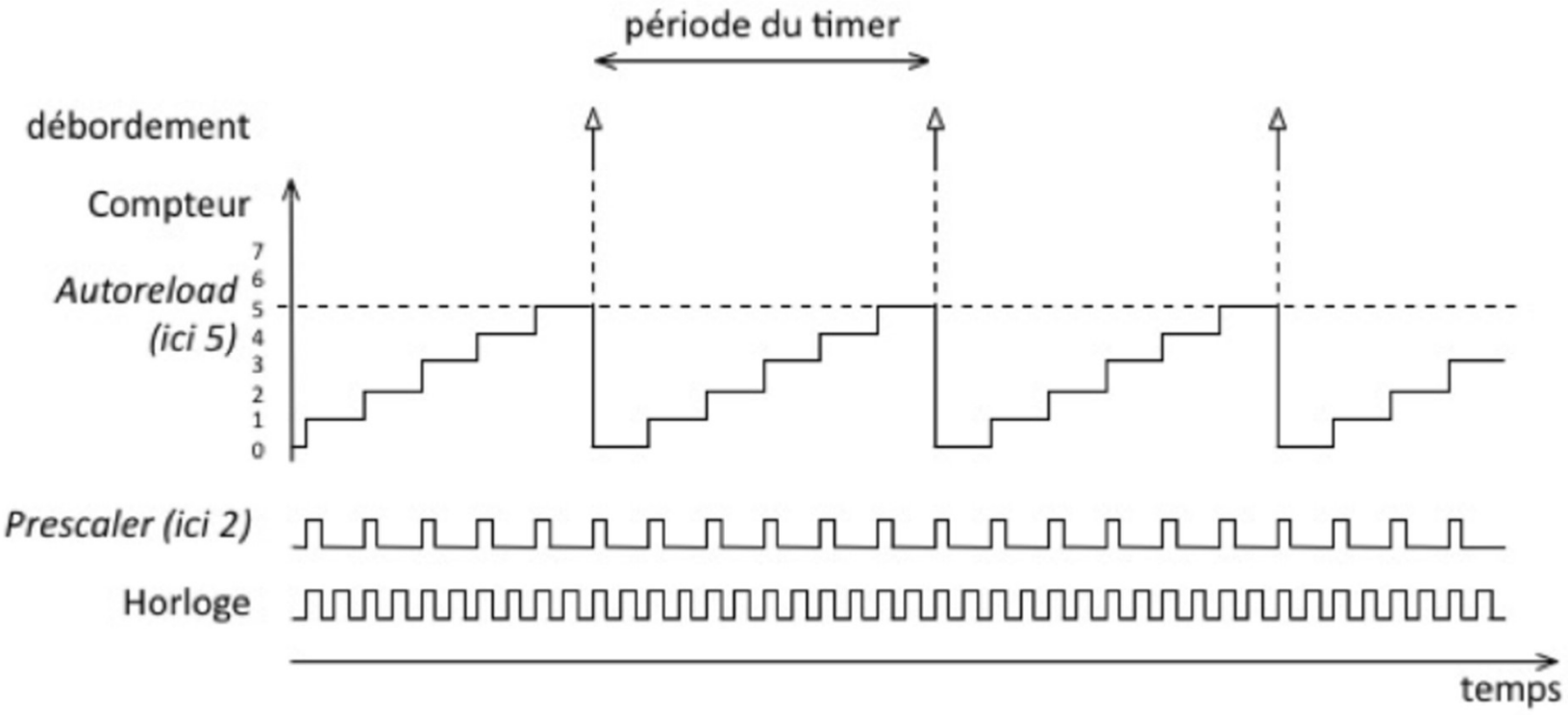
Timer6 Block Diagram

Figure 154. Basic timer block diagram



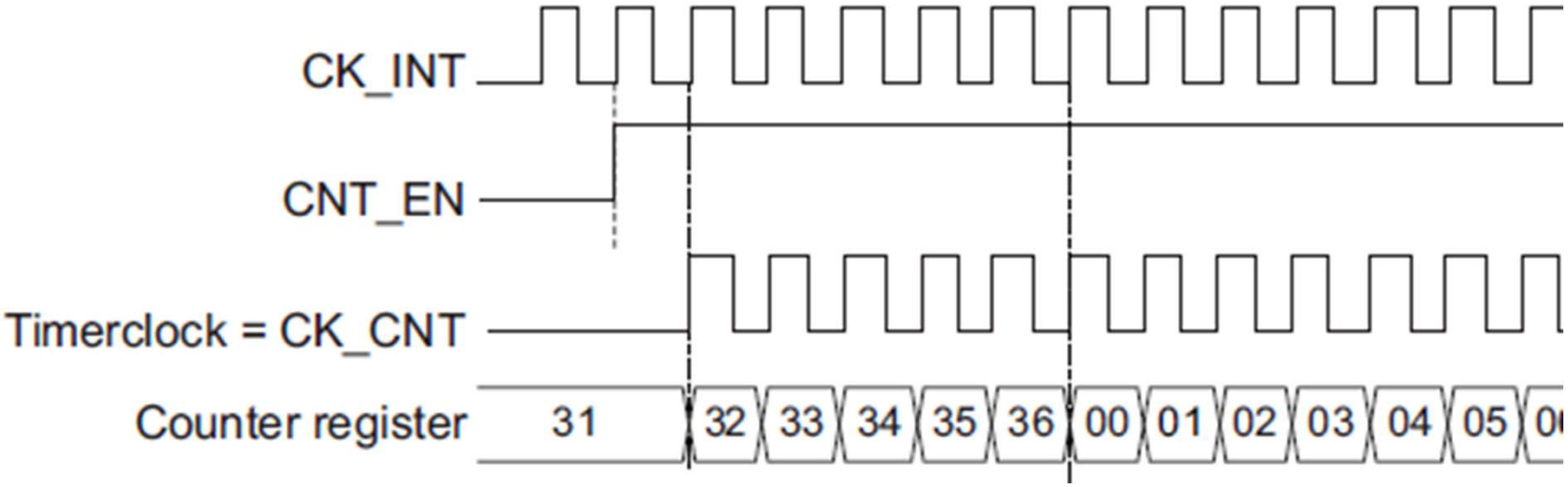
Timer6 in STM32L152 is a basic counter

Timer Basics



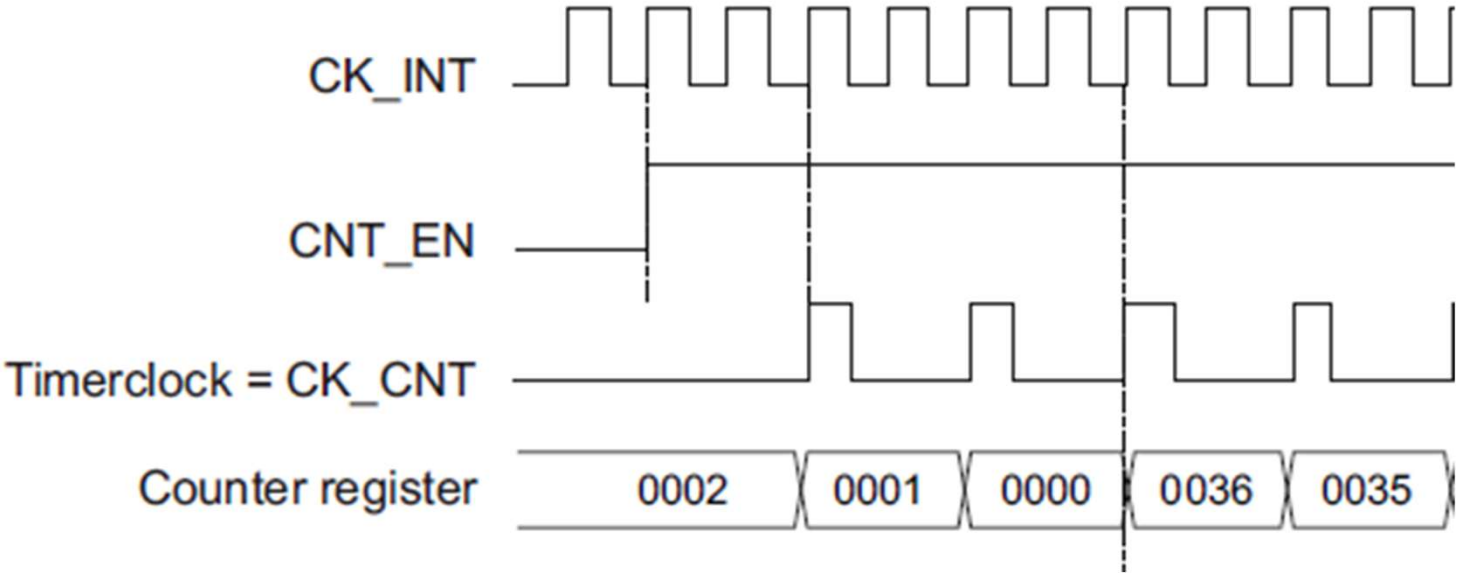
Counter (TIMx_CNT)

- ✓ Counter value is stored in the TIMx_CNT register
- ✓ Counter value is incremented (or decremented) at each timer clock rising edge



Prescaler Register (TIMx_PSC)

- ✓ Counter value is increment (or decremented) at each timer clock
- ✓ The prescaler value divides the CK_PSC clock frequency (by 2 in the example below).



Prescaler Register Implementation

✓ Prescaler implementation example in STM32M152

17.4.11 TIMx prescaler (TIMx_PSC)

Address offset: 0x28

Reset value: 0x0000

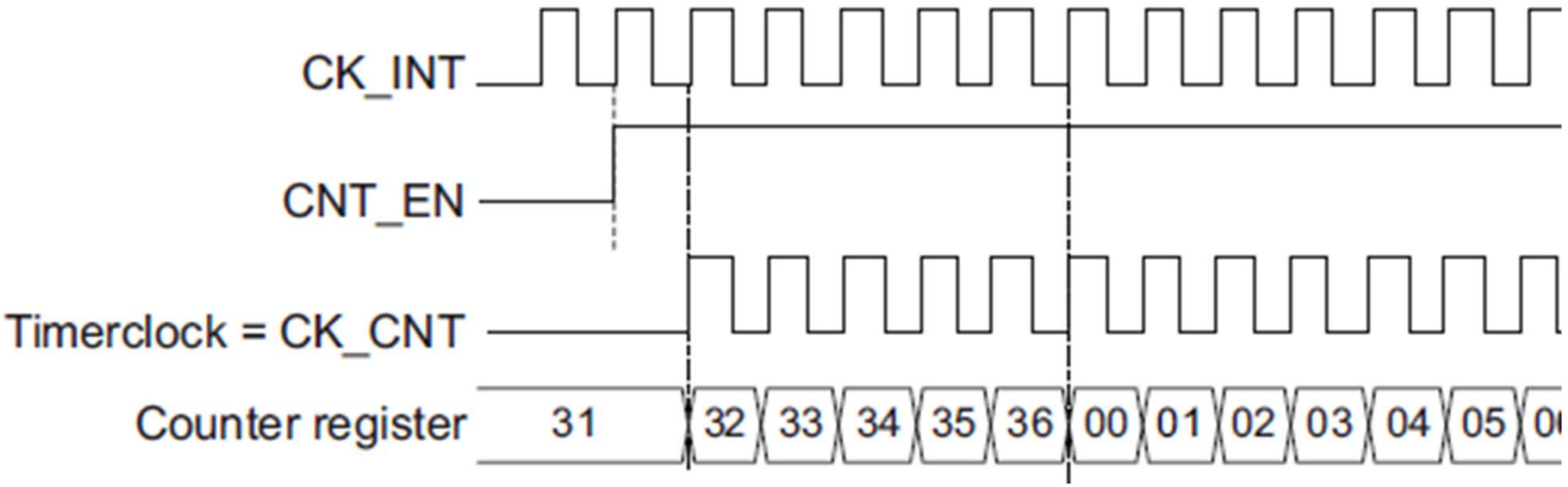
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 **PSC[15:0]**: Prescaler value

The counter clock frequency CK_CNT is equal to f_{CK_PSC} / (PSC[15:0] + 1).
PSC contains the value to be loaded in the active prescaler register at each update event (including when the counter is cleared through UG bit of TIMx_EGR register or through trigger controller when configured in “reset mode”).

Auto-reload Register (TIMx_ARR)

- ✓ Let's assume TIMx_ARR value is 0x36
- ✓ When Counter register is equal to 0x36, a counter overflow occurs and the counter value is reset



Update Event

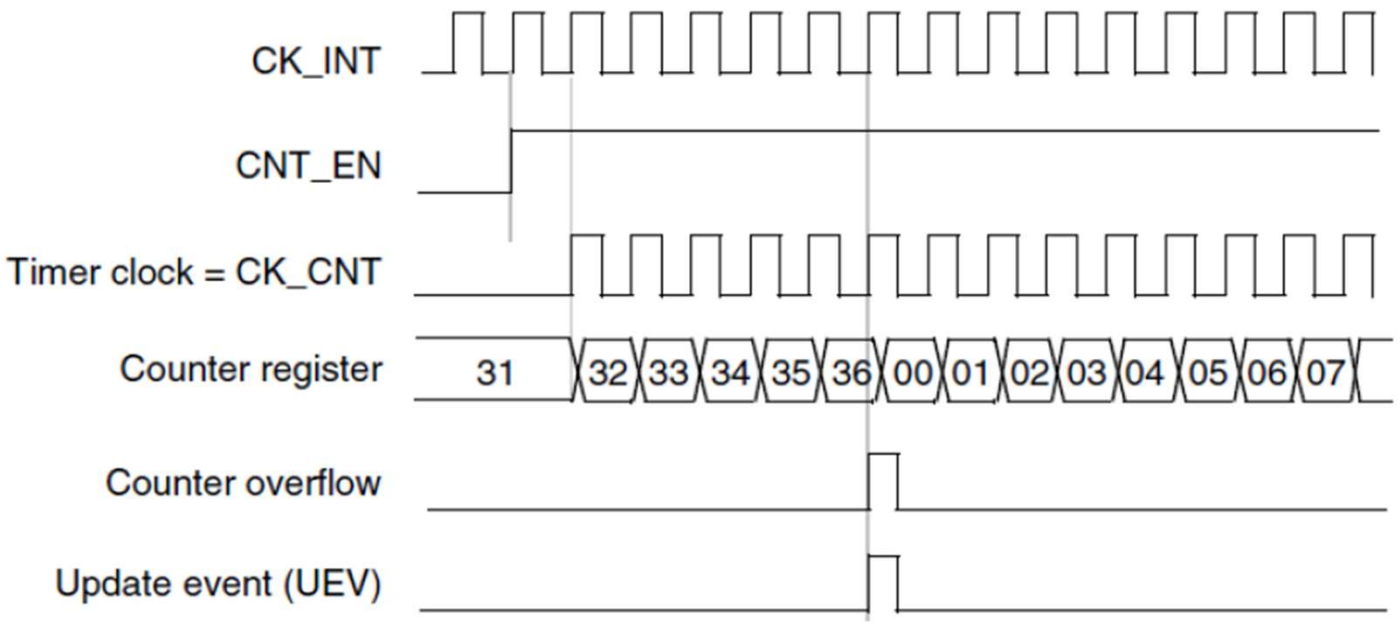
An event occurs when:

- ✓ The value in the counter register is equal to the value in the autoreload register
- ✓ Overflow (the counter reaches the maximum value (0xFF for an 8-bit timer, 0xFFFF for a 16-bit timer)
- ✓ Underflow (the counter reached 0x00)

- ✓ Generated by software by setting the UG bit in the TIMx_EGR register

Upcounting Mode Example

TIMx_ARR=0x36 , internal clock divided by 1

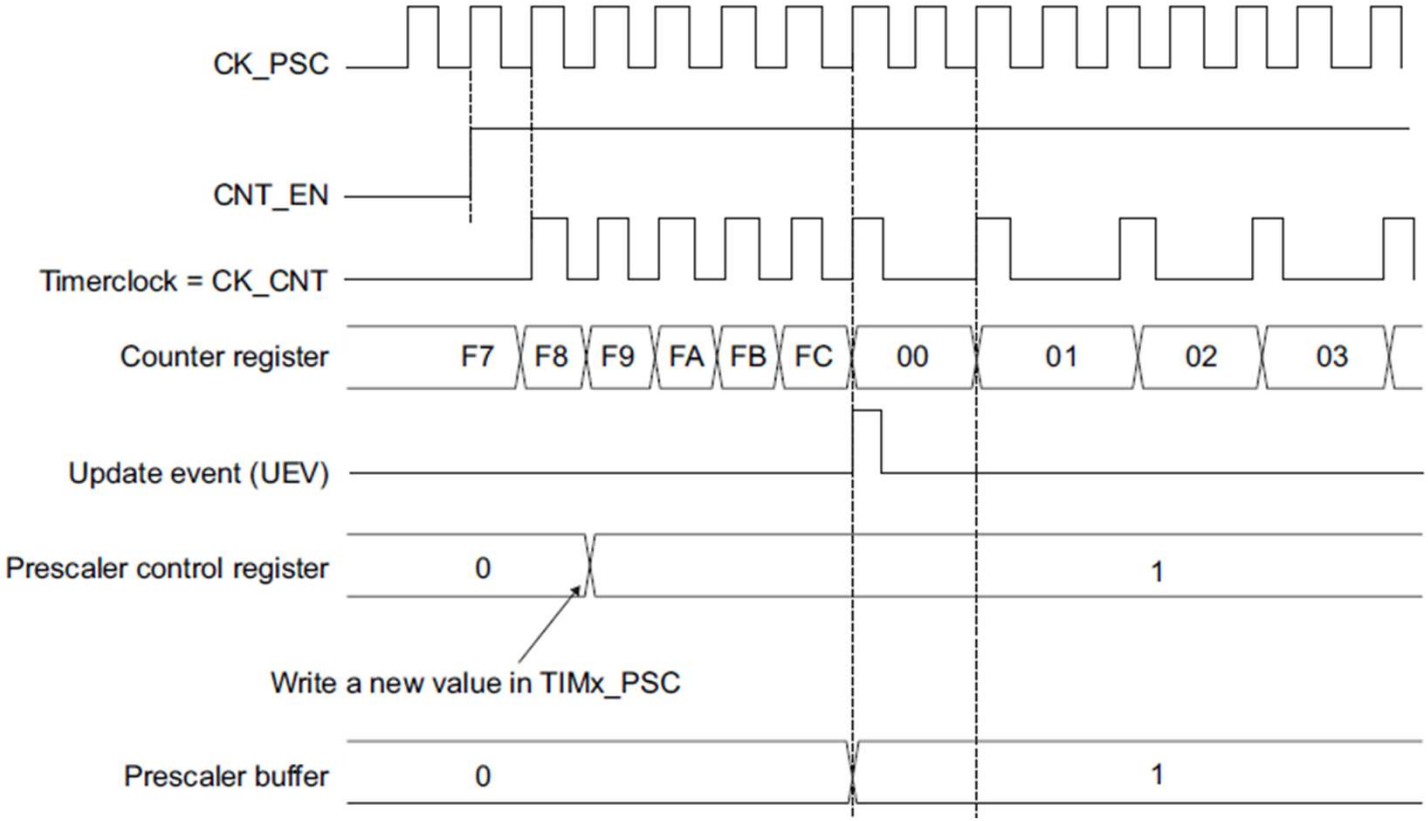


Actions at an Event

AT an event, registers are updated:

- ✓ Counter to a reset value
- ✓ Prescaler register update
- ✓ Auto-reload register update (from preload to shadow register)
- ✓ …more registers depending on timers

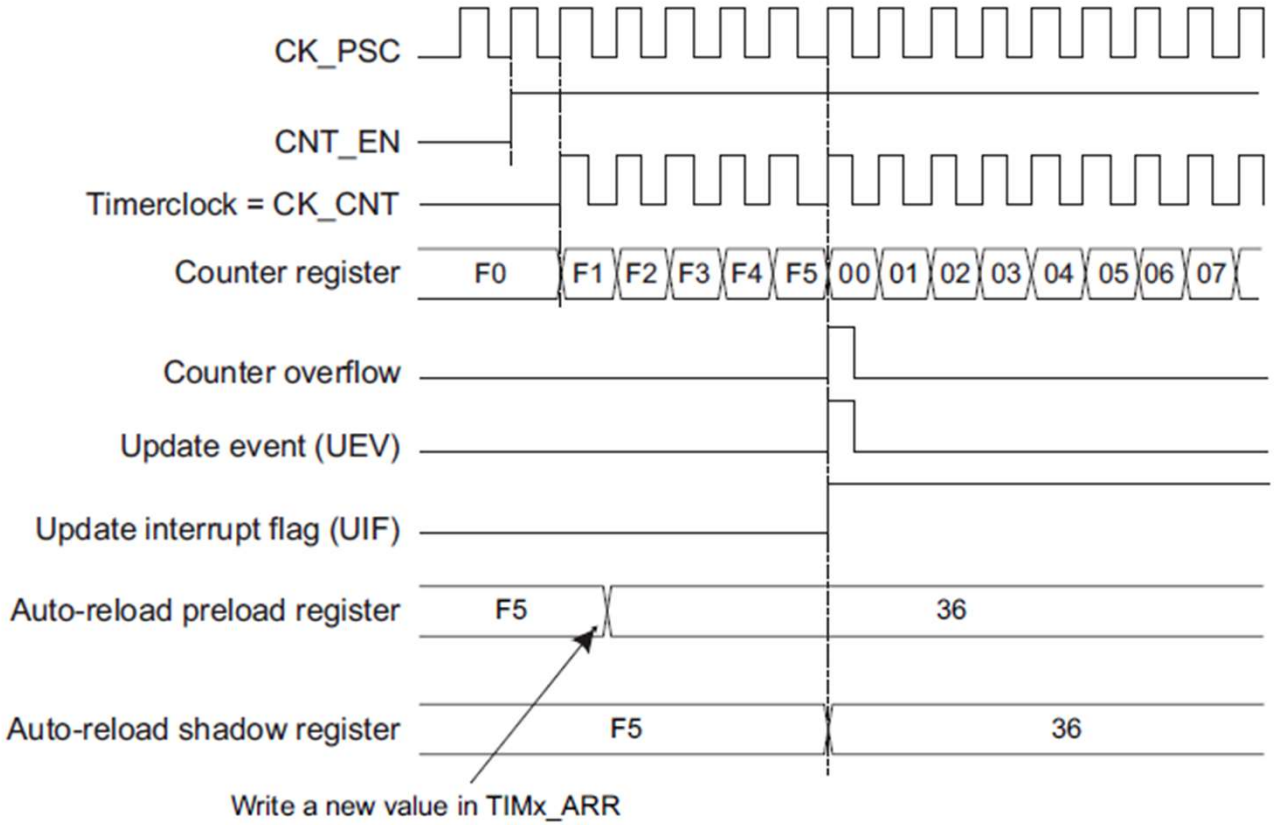
Prescaler Division Change from 1 to 2



ARPE bit

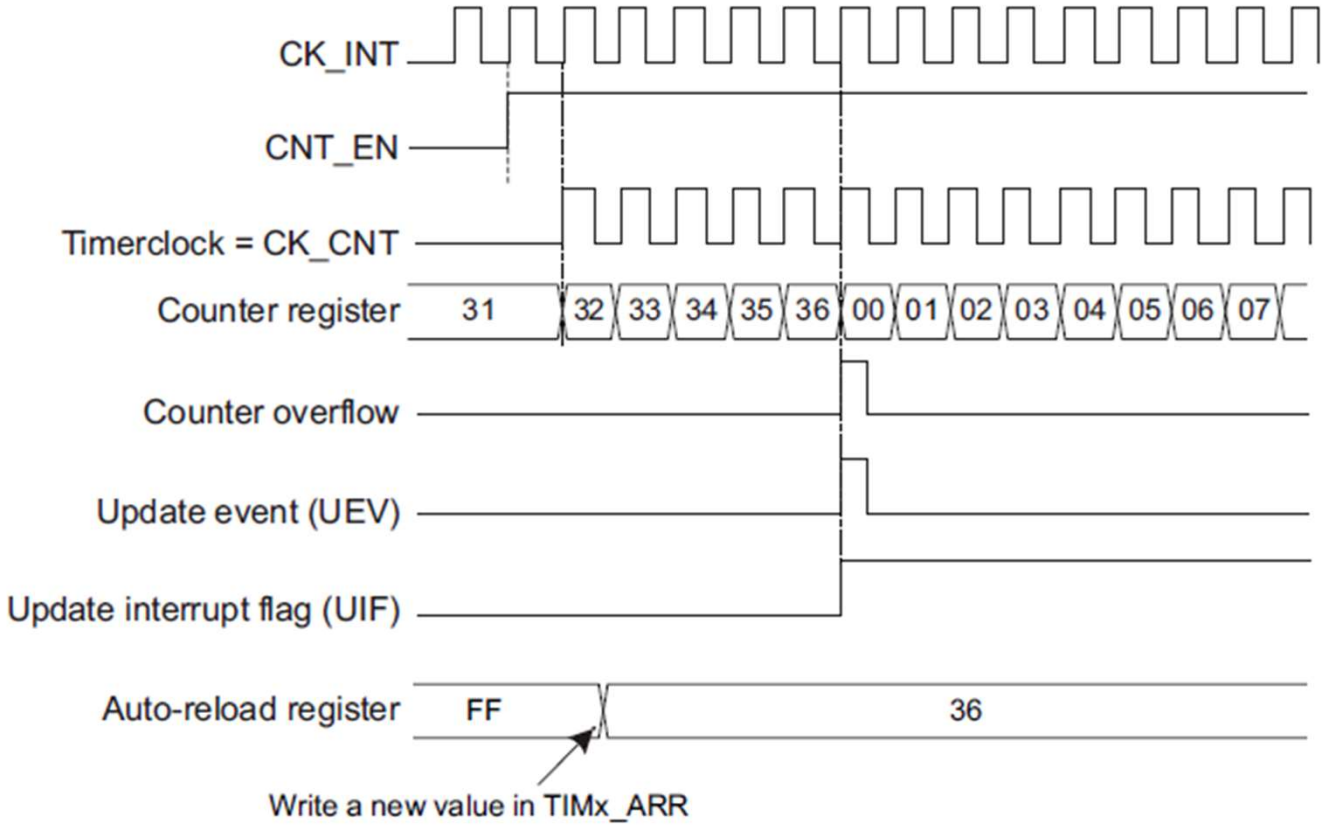
- ✓ ARPE is located in Control Register 1 (TIMx_CR1)
- ✓ ARPE stands for Auto-Reload Preload Enable.
- ✓ ARPE = 0 : TIMx_ARR register is not buffered through a preload register. It can be written directly
- ✓ ARPE = 1 : TIMx_ARR register is buffered through a preload register. An event is necessary to update the autoreload register

ARPE = 1



The overflow triggers an event and the auto reload register is updated at the event time

ARPE = 0



the auto reload register is updated immediately

Interrupts

19.4.3 TIM6 and TIM7 DMA/Interrupt enable register (TIMx_DIER)

Address offset: 0x0C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							UDE	Reserved							UIE
							rw								rw

Bit 0 **UIE**: Update interrupt enable
0: Update interrupt disabled
1: Update interrupt enabled

When an UEV occurs, an interrupt can be programmed at the same time.

43	50	settable	TIM6	TIM6 global interrupt	0x0000_00EC
44	51	settable	TIM7	TIM7 global interrupt	0x0000_00F0

The interrupts for TIM6 and 7 is at the address 0x0000 00EC and 0x0000 00F0

General Purpose Timer

- ✓ Basic Timers include: Clock, Prescaler to divide clock frequency, Counter to store the number clock tics, Autoreload register to interact with the counter (compare timing, reload values)

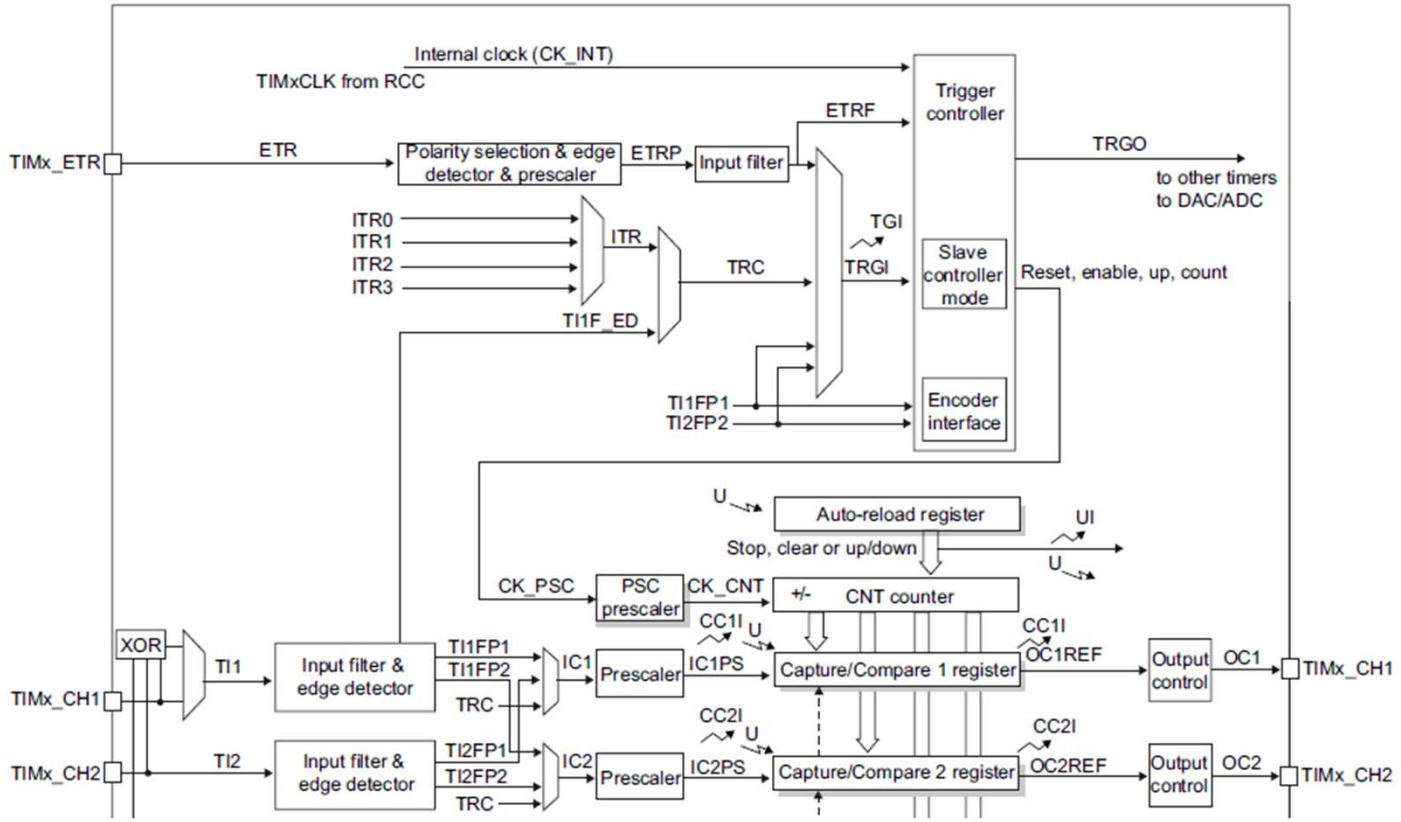
- ✓ Most General Purpose Timers include:
  - independent channels for:
    - Input capture
    - Output compare
  - PWM generation (edge-aligned mode)
  - Interrupt request generation on the following events:
    - Update: counter overflow, counter initialization (by software)
    - Input capture
    - Output compare

ISEN ALL IS DIGITAL! OUEST | yncréa

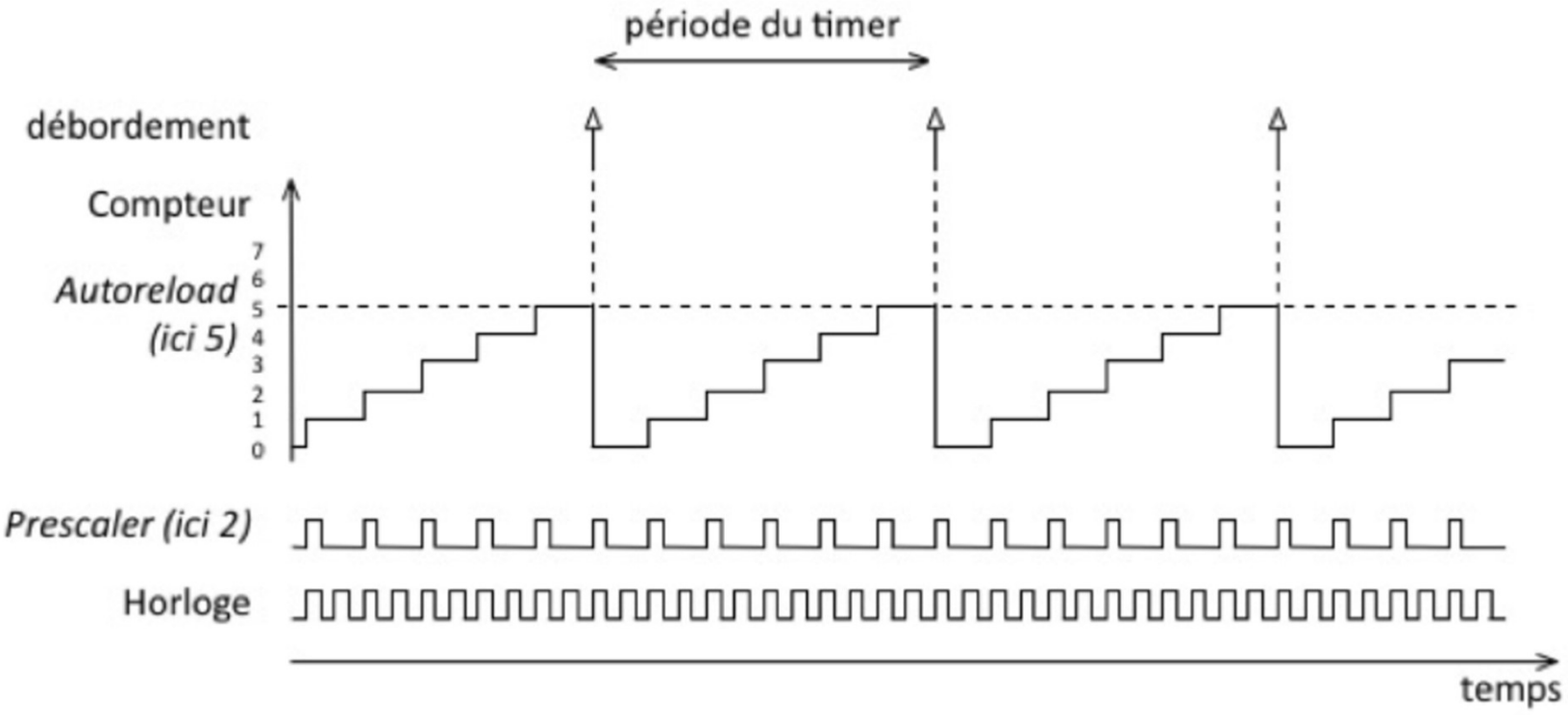
Timer2 Block Diagram

Basic Timer

General Purpose Timer

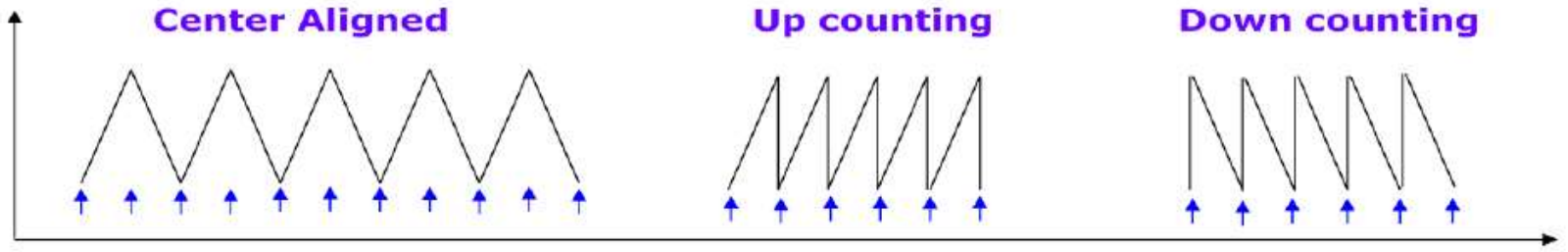


Timer Basics

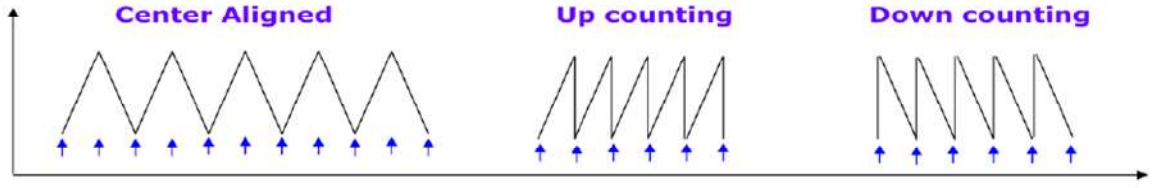


Counting Mode

- ✓ Up-counting Mode: the counter counts from 0 to a user-defined compare value (content of the ARR (AutoReload Register) register. It then restarts from 0 and generates a counter overflow
- ✓ Down-counting Mode: the counter counts from the auto-reload value down to 0. it then restarts from the auto-reload value and generates a counter underflow
- ✓ Center-aligned (up/down counting): the counter counts from 0 to the auto-reload value of -1 (content of the ARR register). This generates a counter overflow event. The counter then counts down to 0 and generates a counter underflow event. After this, the counter restarts counting from 0.



Counting Mode Configuration



7	6	5	4	3	2	1	0
ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
rw	rw	rw	rw	rw	rw	rw	rw

Counting Mode is configured thanks to CMS (Center-aligned mode selection) and DIR (Direction) bits in the Control Register 1 (TIMx_CR1)

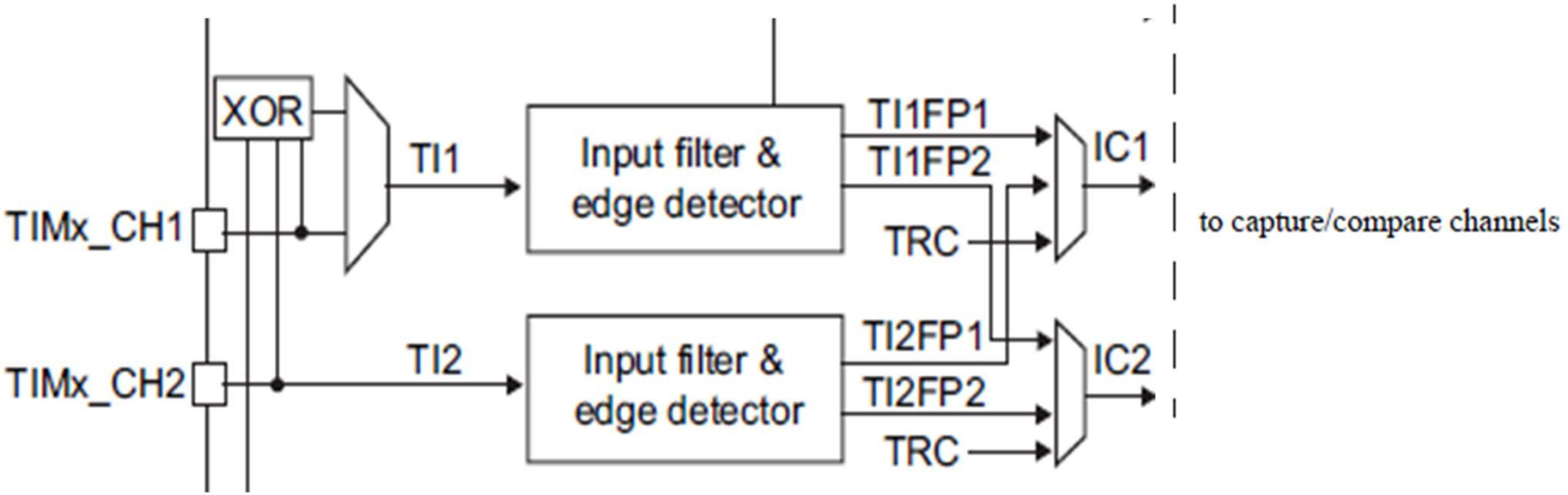
Channel of Timer

- ✓ A channel connects pins to a timer
- ✓ A channel can be configured in input (capture mode) or in output (compare mode)
- ✓ Direction of the channel is defined by configuring the CC1S bits in the Capture/compare mode register 1 (TIMx_CCMRy)
- ✓ A timer can have various channels and each channel must be configured

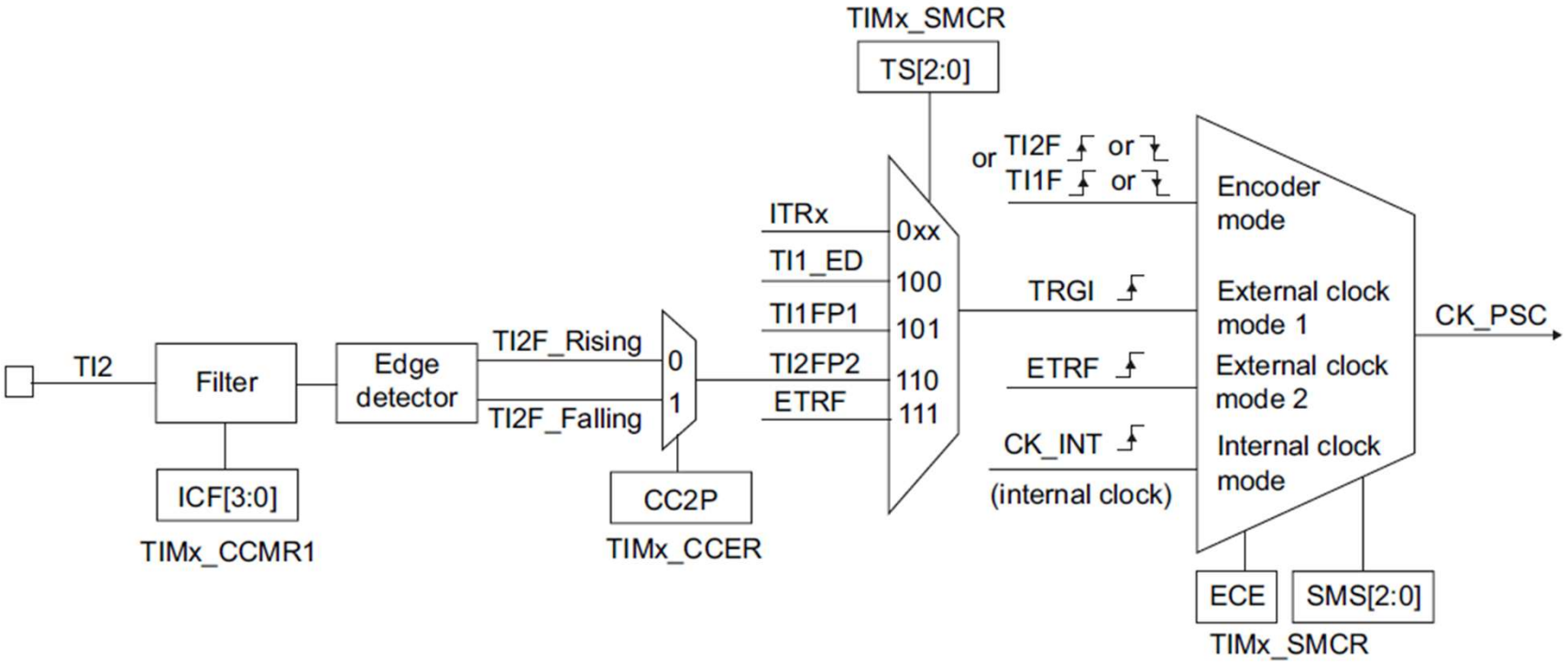
Input Capture

- ✓ Input Capture helps to measure the width of input signals.
- ✓ Records a timestamp in memory when an input signal is received
- ✓ Set a flag indicating that an input has been captured and can be read
- ✓ Value is stored in the Capture/compare Register. The counter value (CNTR register) is transferred to the TIMx_CCRxx register on an ICx event (transition detected on the corresponding Ici signal)

Channel Input Stage Schematic

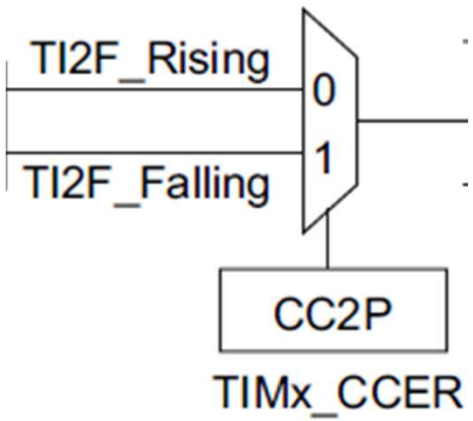


Input Stage STM32L152



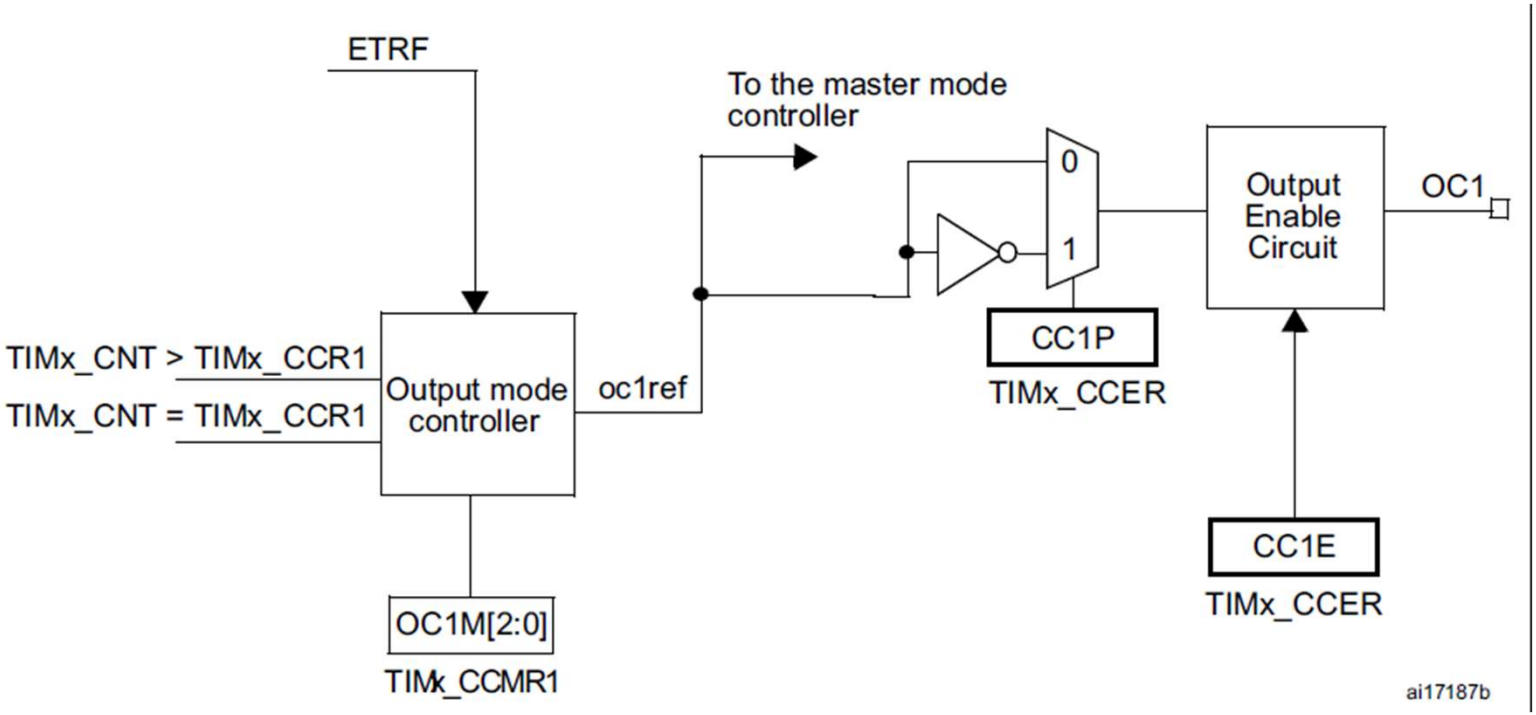
Polarity

After the Timer Input is filtered, the type of signal to detect must be configured.



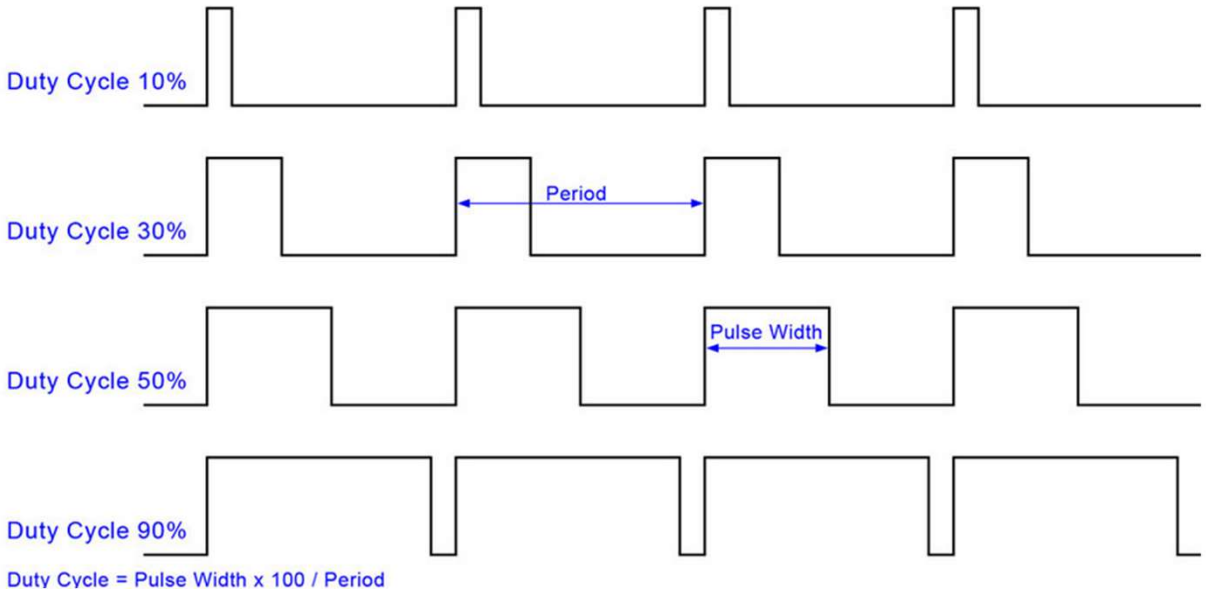
- ✓ A rising or falling edge is called the polarity of the signal
- ✓ It is configured by the bit CCxP (Capture/Compare Polarity) in the TIMx_CCER1 register (Capture/Compare Enable Register 1),

Output Stage

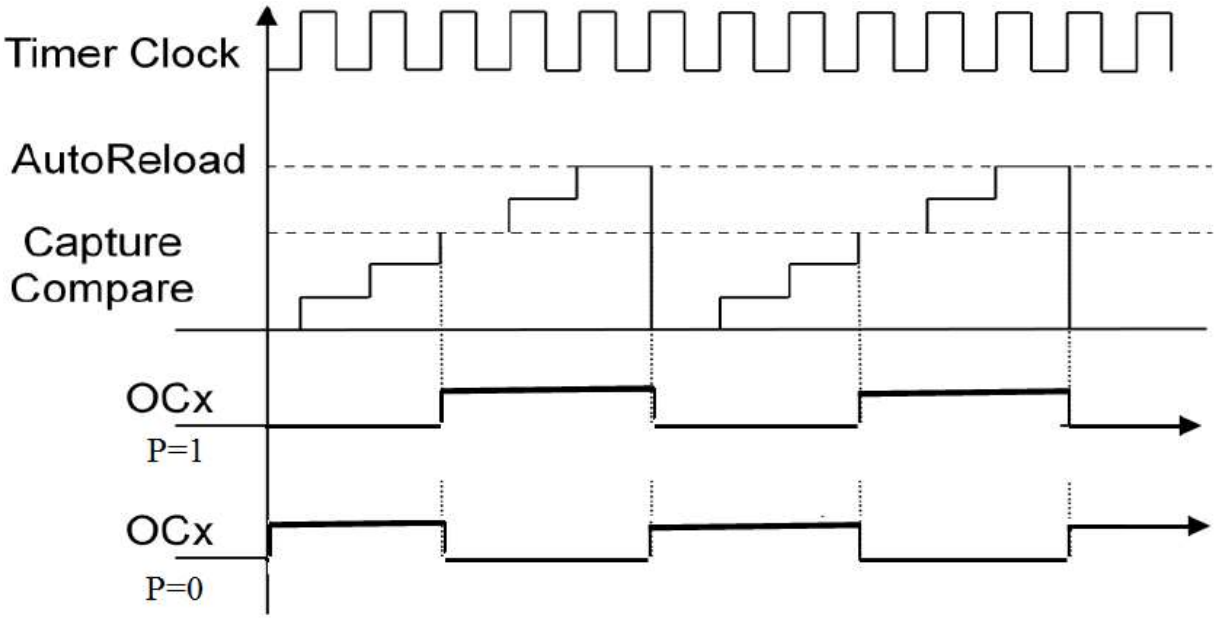


PWM

✓ PWM stands for Pulse Width Modulation

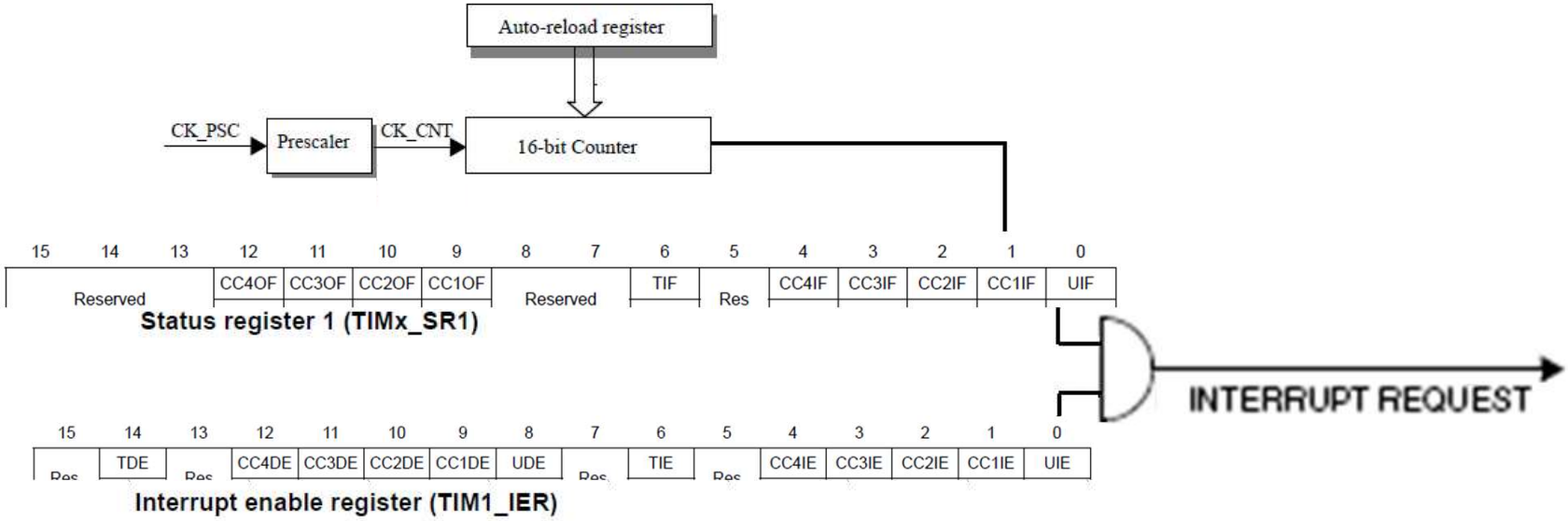


Output Stage Polarity



- ✓ Frequency depends on Autoreload Register
- ✓ Duty cycle depends on Capture Compare Register

Interrupt Timer



- ✓ UIF (Update Interrupt Flag) is set by hardware when registers are updated
- ✓ UIE is Update Interrupt Enable and is set by software

Timer Events Connected to Interrupts

Position	Priority	Type of priority	Acronym	Description	Address
25	32	settable	TIM9	TIM9 global interrupt	0x0000_00A4
26	33	settable	TIM10	TIM10 global interrupt	0x0000_00A8
27	34	settable	TIM11	TIM11 global interrupt	0x0000_00AC
28	35	settable	TIM2	TIM2 global interrupt	0x0000_00B0
29	36	settable	TIM3	TIM3 global interrupt	0x0000_00B4
30	37	settable	TIM4	TIM4 global interrupt	0x0000_00B8

Each timer has dedicated interrupts

ISEN ALL IS DIGITAL! OUEST | yncréa

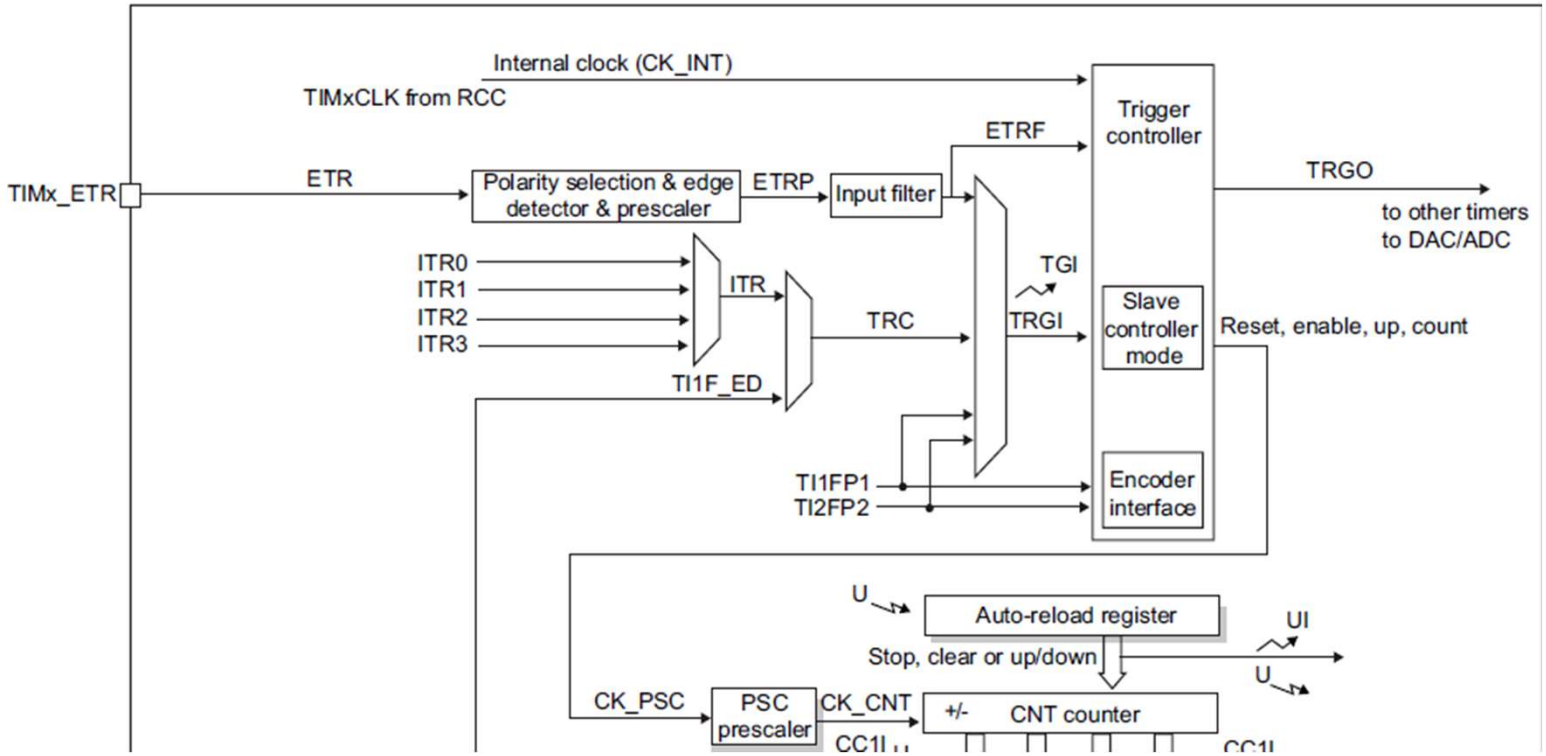
Timer1 Registers Acronyms

To get an overview of the timer, try to remember the acronyms of the register

Acronym	Meaning
ARR	Auto-reload Register
CCMR	Capture Compare Mode register
CCR	Capture/compare Register
CNTR	Counter Register
CR	Control Register
EGR	Event Generation Register

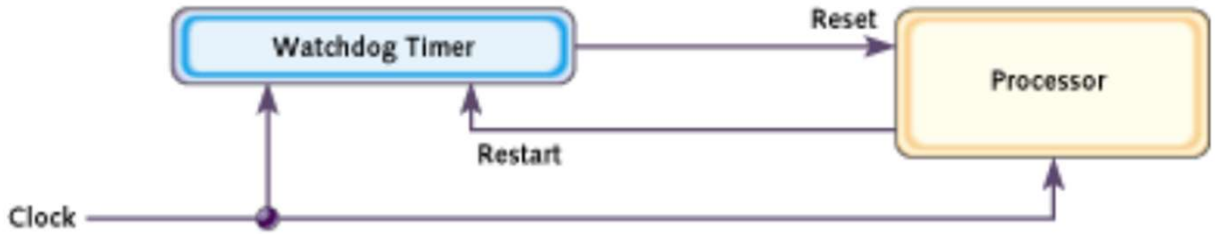
Acronym	Meaning
ETR	External Trigger Register
IER	Interrupt Enable Register
PSCR	Prescaler Register
RCR	Repetition Counter Register
SMCR	Slave Mode Control Register
SR	Status Register

Timer1 General Block Diagram



A timer can have many options for the Input Clock

Watchdog Timer



- ✓ A watchdog timer is a piece of hardware that can be used to automatically detect software anomalies and reset the processor if any occur. Generally speaking, a watchdog timer is based on a counter that counts down from some initial value to zero.
- ✓ The embedded software selects the counter's initial value and periodically restarts it. If the counter ever reaches zero before the software restarts it, the software is presumed to be malfunctioning and the processor's reset signal is asserted. The processor (and the embedded software it's running) will be restarted as if a human operator had cycled the power.