## Timer initialize ( simple delay)

Conter register: 16 bit

Prescaler register: 16 bit

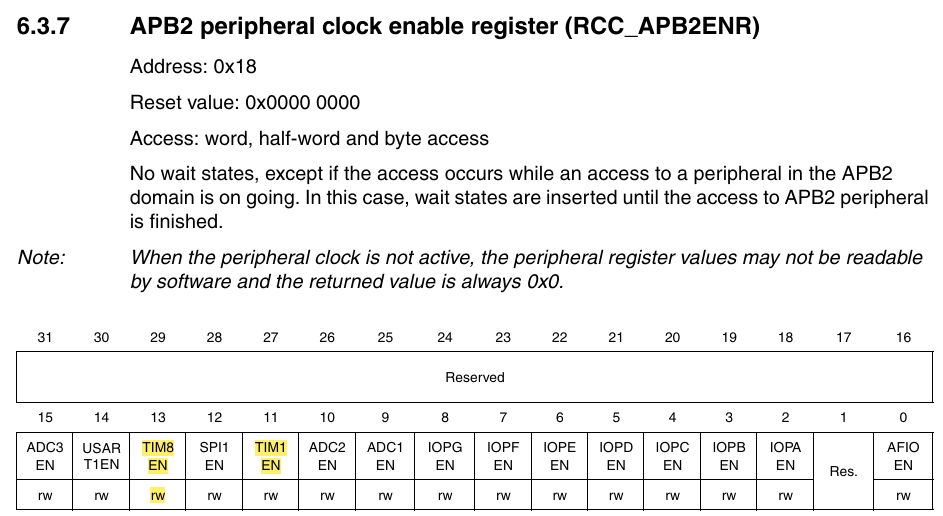
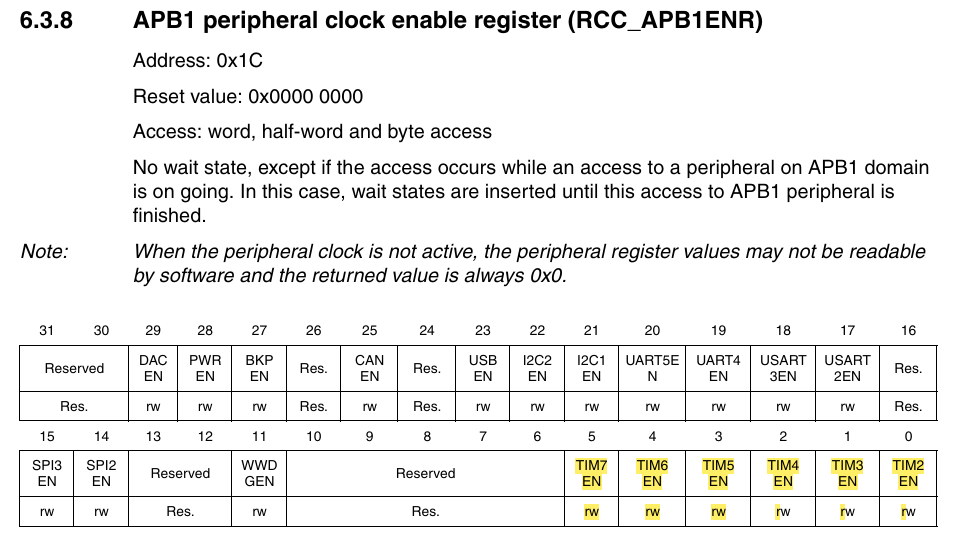
Auto reload register: 16 bit

Tim->1 is in APB2

Tim->2,3,4 in APB1

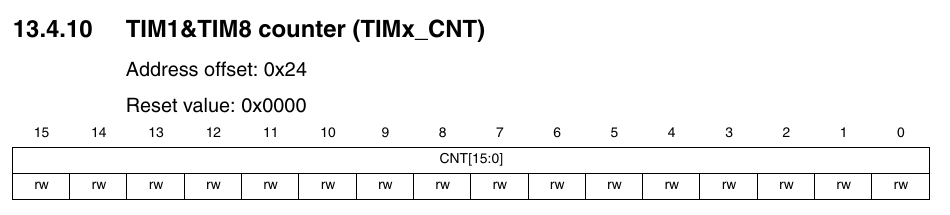
## Steps to enable:

### Enable clock:

* 1. Tim->1 is in APB2-> bit 11
  2. Tim->2,3,4 in APB1

### 

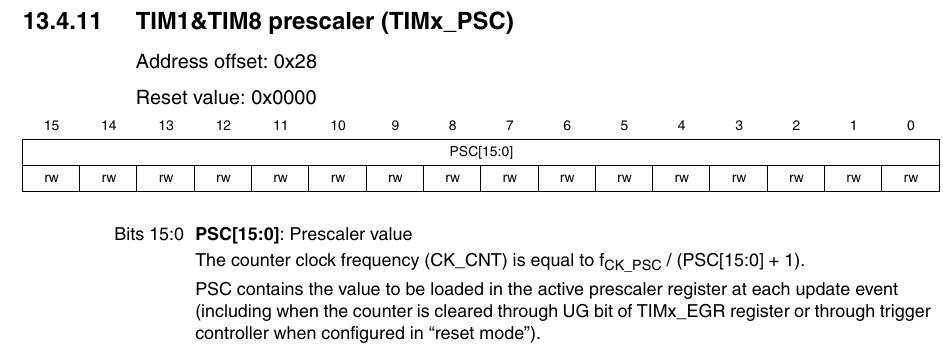
### Set counter value zero



### Set prescaler

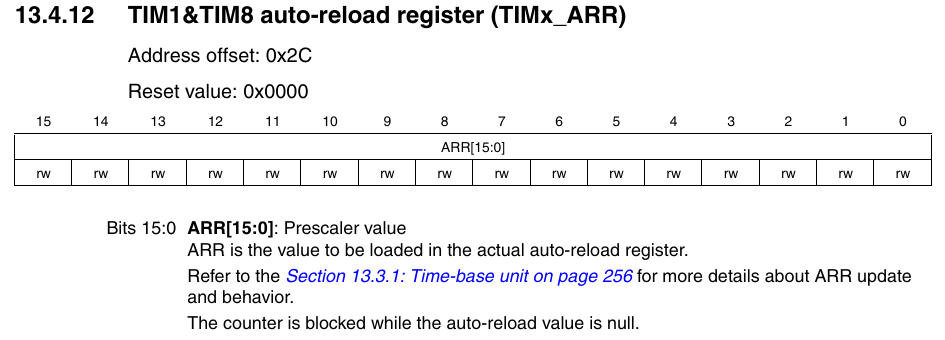
Time delay = (1+psc)/7200 ms

For 1 ms delay, psc = 7200-1



### Set reload value

TIM->ARR (auto reload register), the value of this register is loaded to the counter cnt autometic when counter goes to zero.

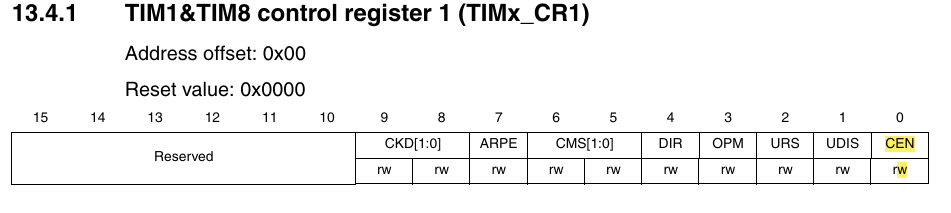


### Select up or down counter

In Tim cr1, bit 4 is DIR (direction) is 0, then up counter, if 1 then down counter.

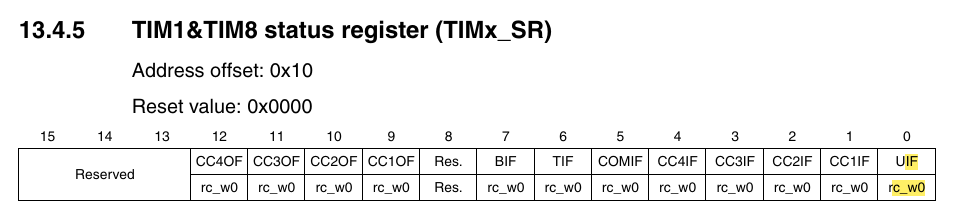
### Enable counter

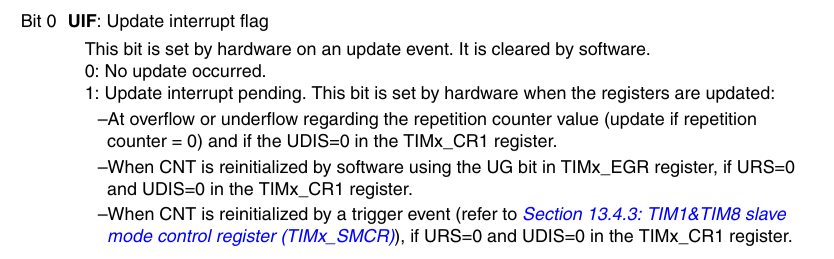
Enable the counter in timer control register.



### Checking status register

TIM1->SR contain a bit which is set to 1 while reloading.





This bit can be used to check time count…

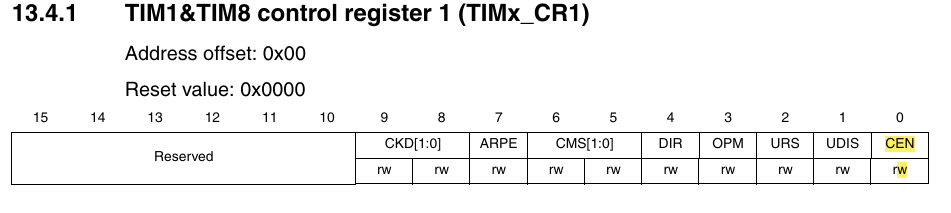
### For global interrupt (NVIC)

* set the the UIE (Update interrupt Enable) in TIMx->DIER
* Enable NVIC interrupt for this Interrupt request number (NVIC\_EnableIRQ(TIM2\_IRQn)
* Write interrupt handler in TIM2\_IRQHandler(void)

On the other hand one pulse in cr1 can also be used to detect the count.

## One pulse

In CR1 bit 3 (OPM) is one pulse mode. If it is enableed, the counterwill stop afte single pulse… then it need to be enable again by setting CEN in CR1 bit 0;



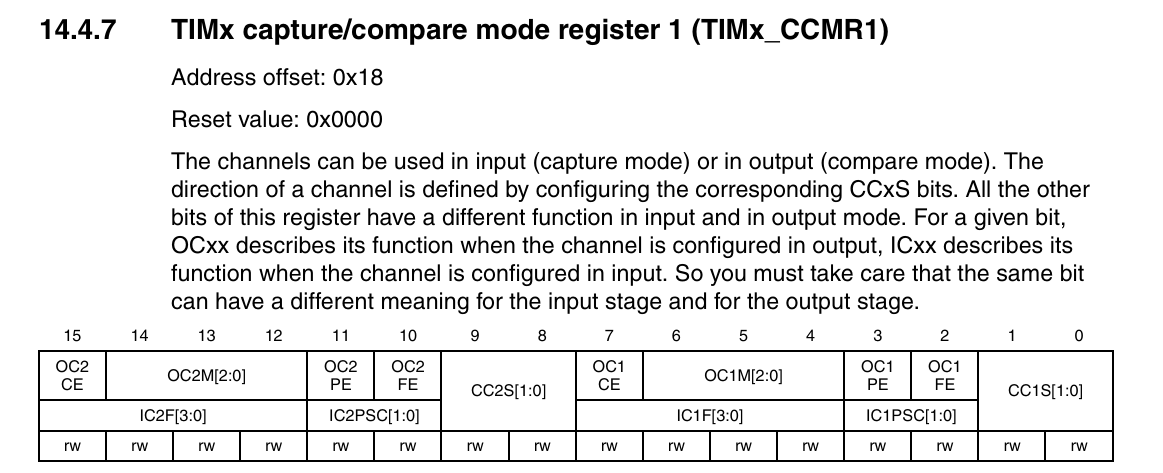
This feature canbe used to make delay function.

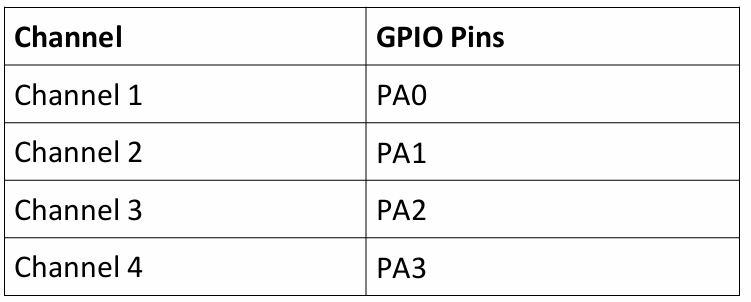
## Output Compare Mode

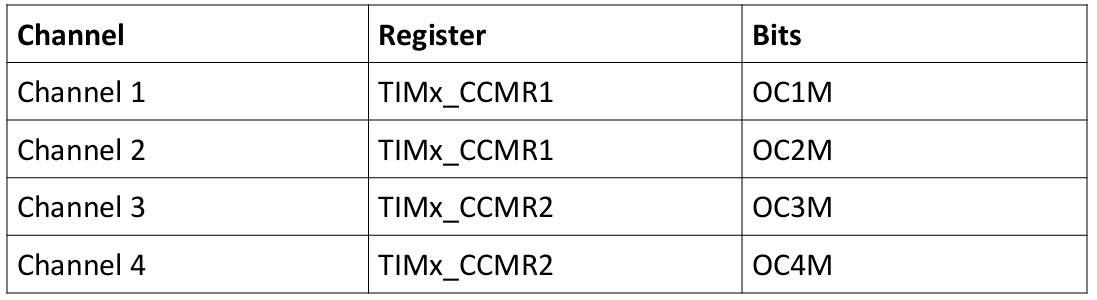
* Step-1: Enable the timer and configure the timer registers- PSC, ARR and CNT

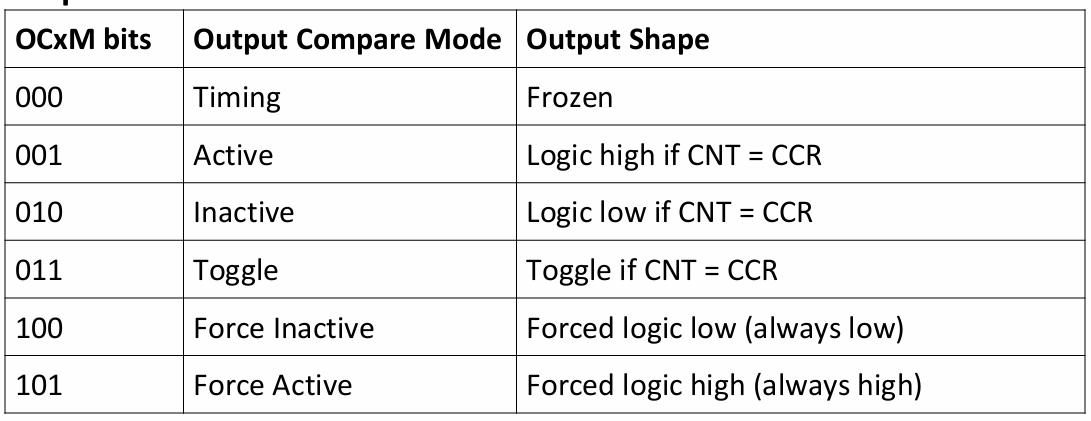
Basic timer setup

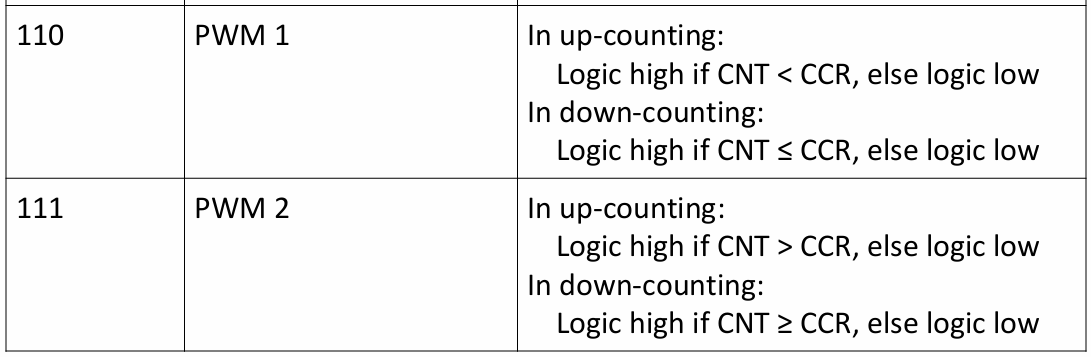
* Step-2: Configure output mode using OCxM in TIMx\_CCMR1/ TIMx\_CCMR2 register. (The shape of output signal after event occured)



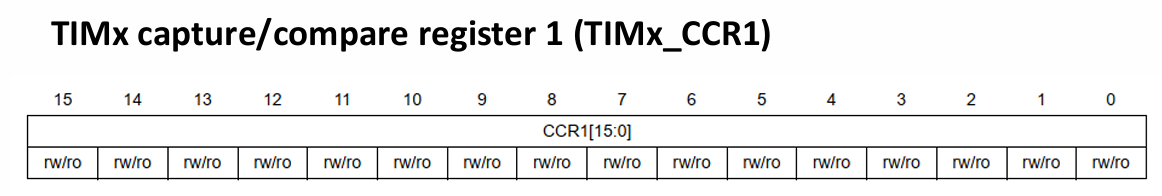




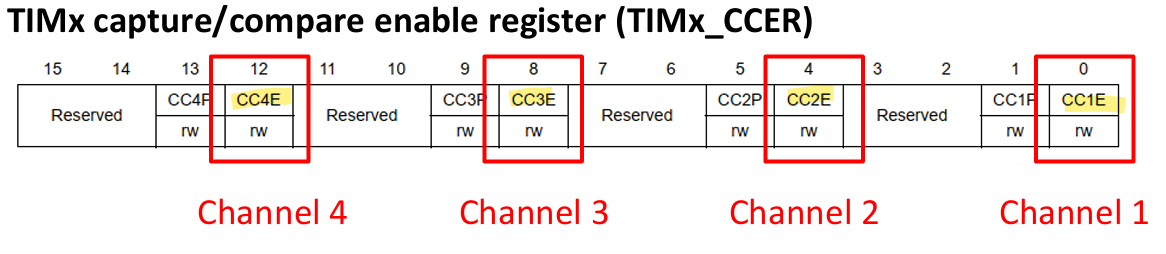




* Step-3: Set compare value in CCRx register. (You can set more than one CCRx. Each channel will generate separate output. You need to configure.)

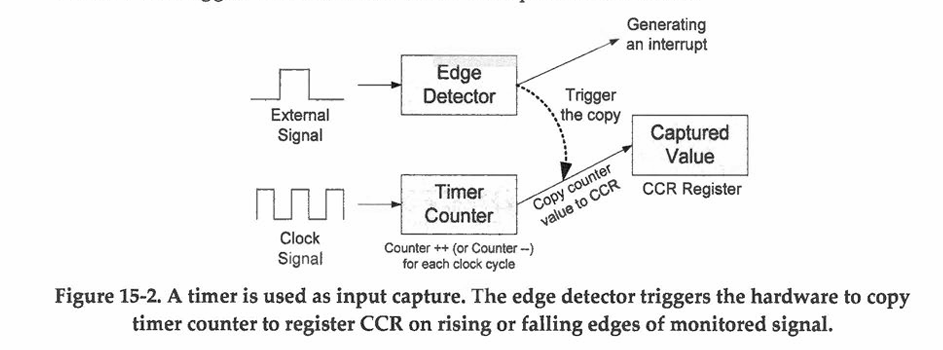


* Step-4: Enable the channels using TIMx\_CCER. (If needed)



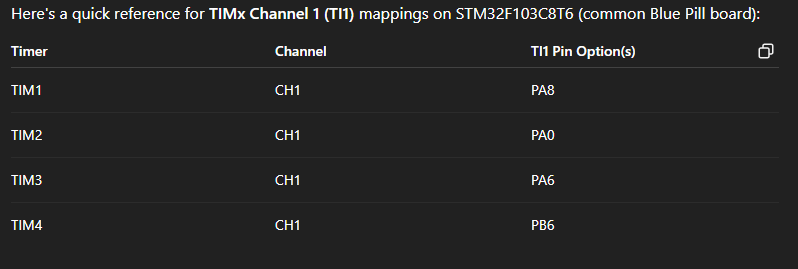
* Step-5: Write the interrupt routine and enable it in NVIC. (If needed)
* Step-6: Start the timer by setting CEN bit in CR1 register.

## Input capture

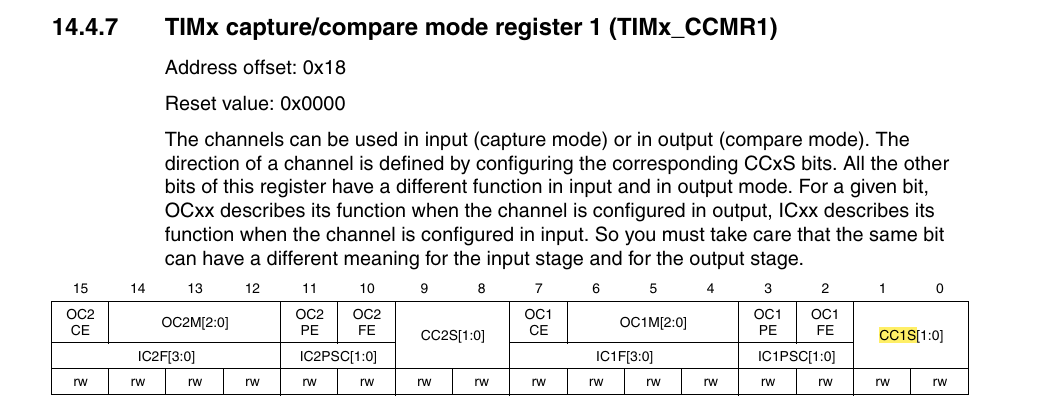


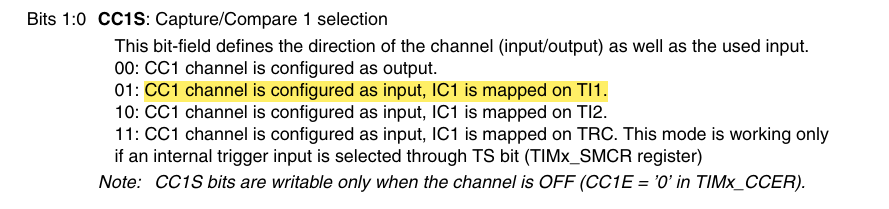
Steps:

1. Setup the timer, cnt, psc and arr
2. Pin and TI1 number

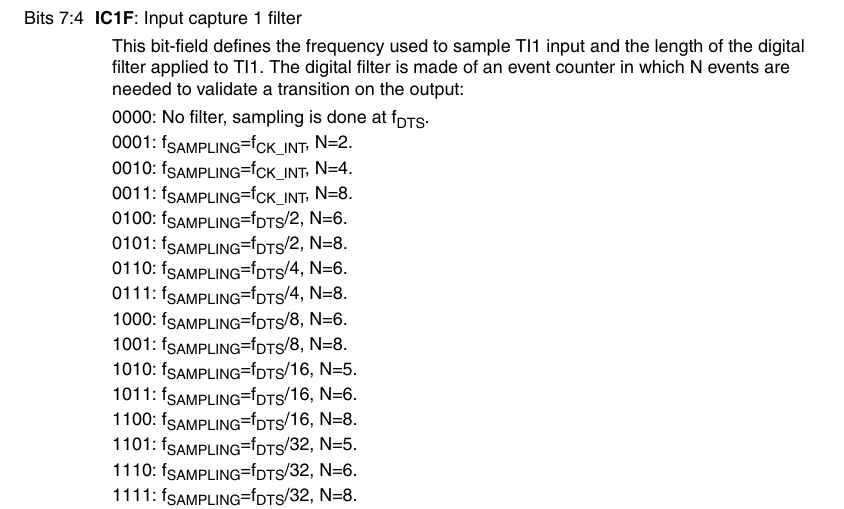


1. 1st of all set bit (CC1S) of TIMx->CCMR1 to convert the channels as input. As soon as CC1S becomes different from 00, the channel is configured in input and the TIMx\_CCR1 register becomes read-only.





1. The IC1F bits (bits 7:4 of the TIMx\_CCMR1 register) configure a digital filter for the TI1 input in Input Capture mode. This filter helps suppress spurious edges, bounces, or noise on the input signal, especially useful for mechanical switches or noisy signals. A **transition** (e.g., low to high) is only **accepted** if the sampled value is **stable for N consecutive samples**. 1111 is suitable for bouncing switch. 0000 is good for no bouncing switch.



1. Select the edge of the active transition on the TI1 channel by writing CC1P bit to 0 in the TIMx\_CCER register (rising edge in this case). And set the CC1E to enable capture

