## Timer initialize ( simple delay)

Conter register: 16 bit

Prescaler register: 16 bit

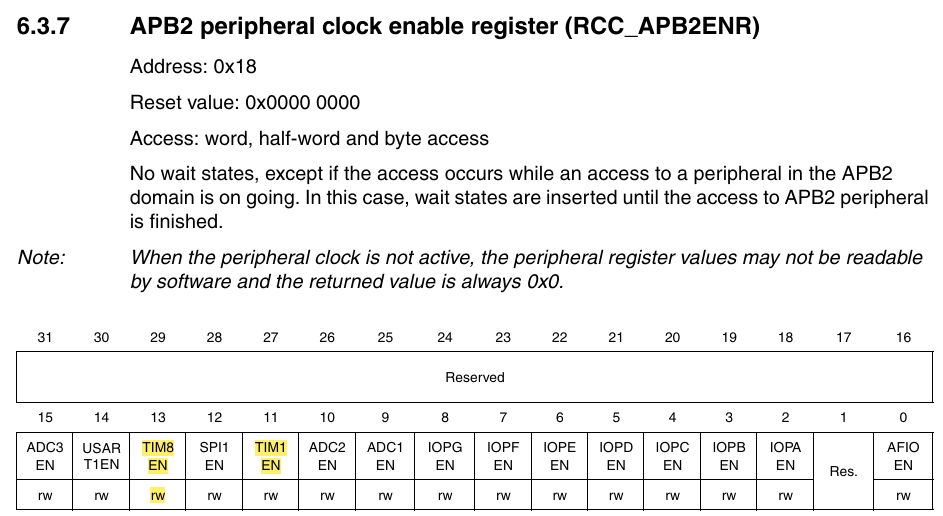
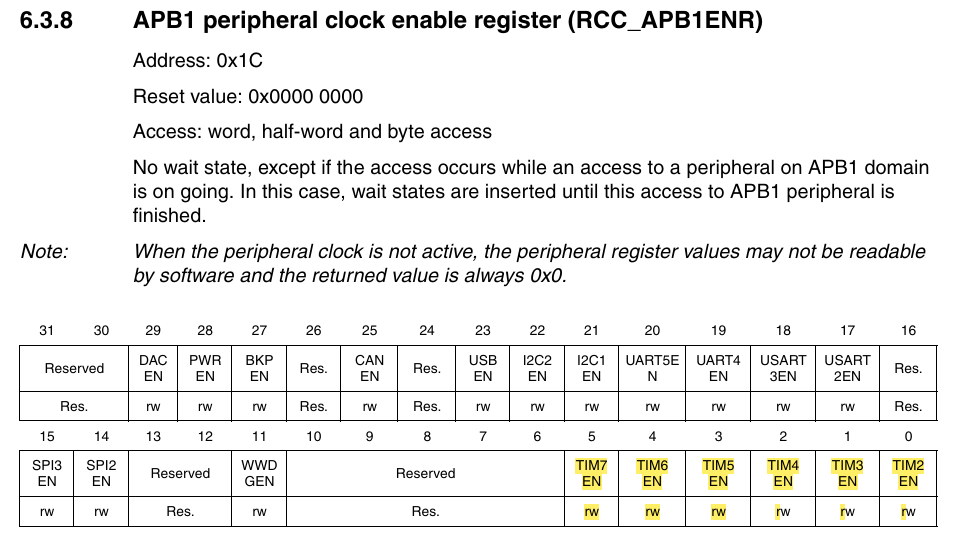
Auto reload register: 16 bit

Tim->1 is in APB2

Tim->2,3,4 in APB1

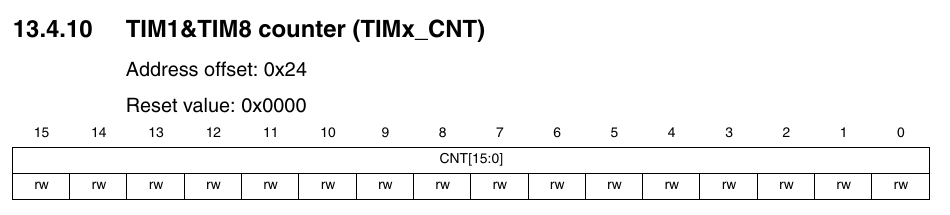
## Steps to enable:

### Enable clock:

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

### 

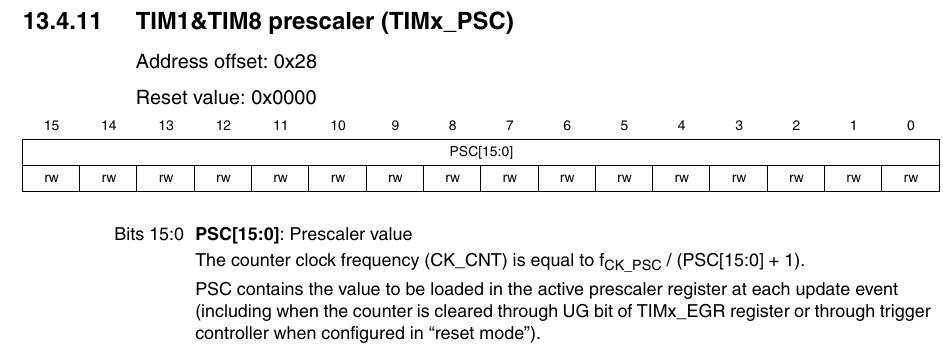
### Set counter value zero



### Set prescaler

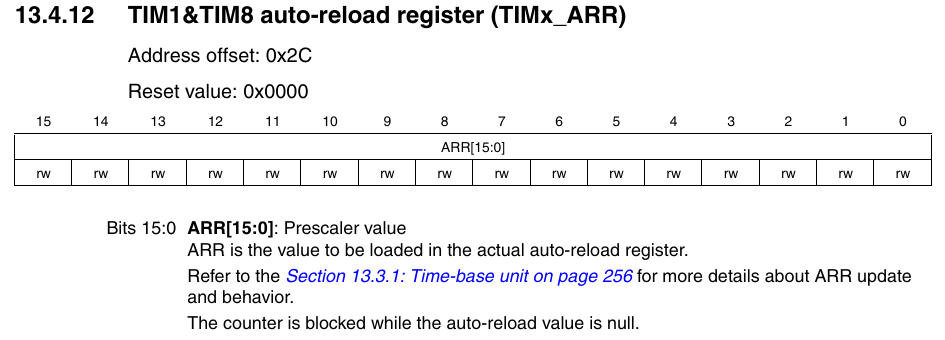
Time delay = (1+psc)/7200 ms

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.

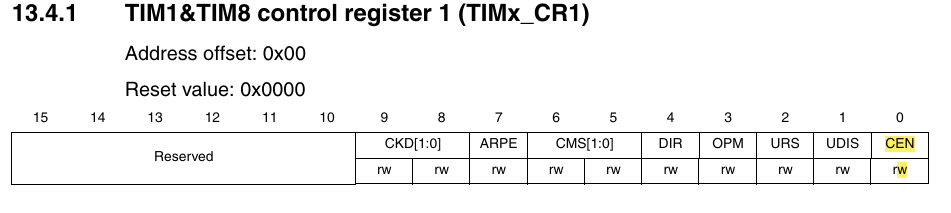


### Select up or down counter

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

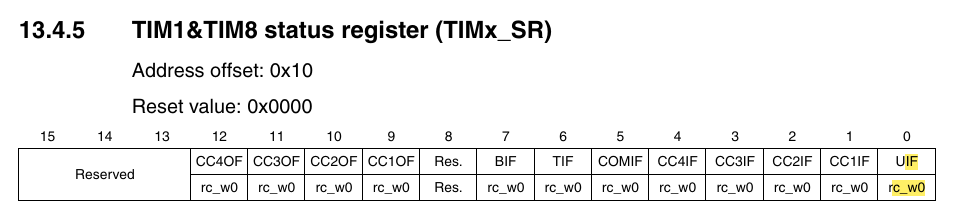
### Enable counter

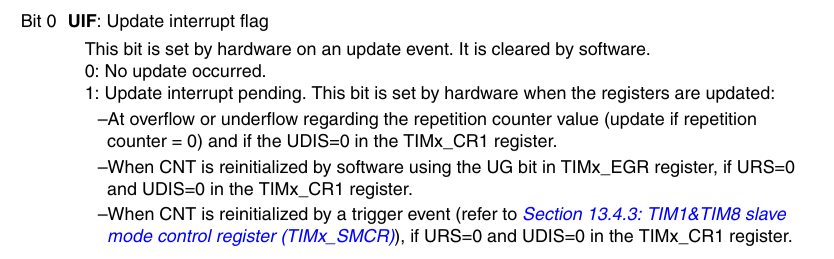
Enable the counter in timer control register.



### Chacking status register

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



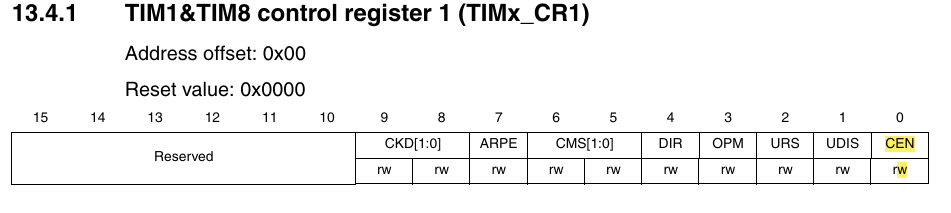


This bit can be used to check time count…

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
* 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.