

## CCS Pic-C: TIMERO Module

### Timer( ) Module Operation:

#### Counter Mode:

In this counter mode operation pulses applied RA4/TOCKI pin are counted in TMRO. These pulses can be input manually from a push-button, from a signal source such as a sensor. TMRO can also be loaded with an initial value.

#### Timer Mode:

In this mode, timing measurements and hardware-generated delays can be obtained by calculating the time between start of the counting from TMRO's preloaded initial value until timer overflow flag is set that causes interrupt, which is given by

$$(4 / F_{xt}) * A * (256 - B)$$

Where;

$F_{xt}$  is oscillator frequency,

A is prescaler value,

B is initial value loaded to TMRO.

Example: If  $F_{xt} = 4\text{MHz}$ ,  $A=32$  and  $B=156$  then total-time delay until TMRO overflow interrupt occurs is 3.2 msec.

## CCS Pic-C: TIMERO Module

In the counter mode operation pulses applied RA4/TOCKI pin are counted in **TMRO**.

In timer mode, timer is incremented at time intervals of  $(4/F_{xt}) * A$ , where  $F_{xt}$  is **oscillator frequency** and **A is prescaler value** ( $A=RTCC\_DIV\_XX$ ).

As an example, if  $F_{xt} = 4\text{MHz}$ , and prescaler is  $RTCC\_DIV\_8$ , then timer is incremented at each  $(4/4000000)*8=8\text{microseconds}$ .

Total time it takes for the timer to overflow is calculated using  $(4/F_{xt}) * A * (256-B)$ ,

Where;

$F_{xt}$  is oscillator frequency,

A is prescaler value ( $A=RTCC\_DIV\_XX$ ),

B is initial value loaded to TMRO.

Example: If  $F_{xt} = 4\text{MHz}$ ,  $A=32$  and  $B=156$  then total-time delay until TMRO overflow interrupt occurs is 3.2 msec.

Functions to set a value to TMRO and get its value are, **set\_timerO()** and **get\_timerO()**, respectively.