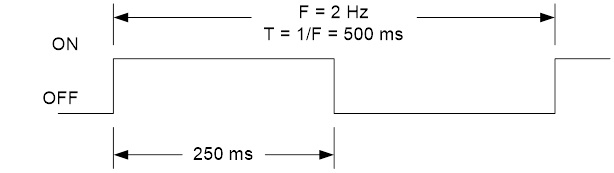
**TIMING TERMINOLOGY**

**Frequency**  
The number of times a particular event repeats within a 1-s period. The unit of frequency is Hertz, or cycles per second. For example, a sinusoidal signal with a 60-Hz frequency means that a full cycle of a sinusoid signal repeats itself 60 times each second, or every 16.67 ms. For the digital waveform shown, the frequency is 2 Hz.

**Period**  
The flip side of a frequency is a period. If an event occurs with a rate of 2 Hz, the period of that event is 500 ms. To find a period, given a frequency, or vice versa, we simply need to remember their inverse relationship, F = 1/T where F and T represent a frequency and the corresponding period, respectively.

**Duty Cycle**  
In many applications, periodic pulses are used as control signals. A good example is the use of a periodic pulse to control a servo motor. To control the direction and sometimes the speed of a motor, a periodic pulse signal with a changing duty cycle over time is used.

Duty cycle is defined as the percentage of one period a signal is ON. The periodic pulse signal shown in the Figure is ON for 50% of the signal period and off for the rest of the period. Therefore, we call the signal in a periodic pulse signal with a 50% duty cycle. This special case is also called a square wave.

*[](https://www.arxterra.com/wp-content/uploads/2018/08/6_DutyCycle50percent.jpg)*

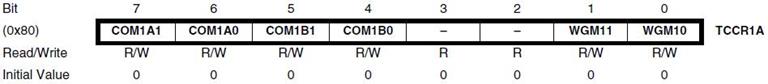
*Figure 4: 50% Duty Cycle*

**TIMER 1 MODES OF OPERATION**

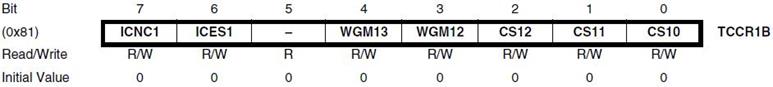
*Table 1: Waveform Generation Mode Bit Description*

**NORMAL MODE**

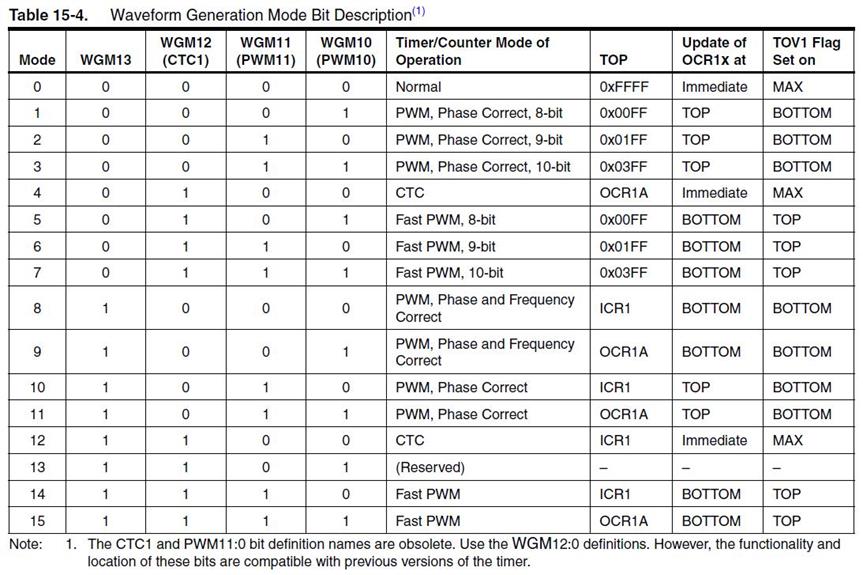
* The simplest AVR Timer mode of operation is the Normal mode. Waveform Generation Mode for Timer/Counter 1 (WGM1) bits 3:0 = 0. These bits are located in Timer/Counter Control Registers A/B (TCCR1A and TCCR1B).

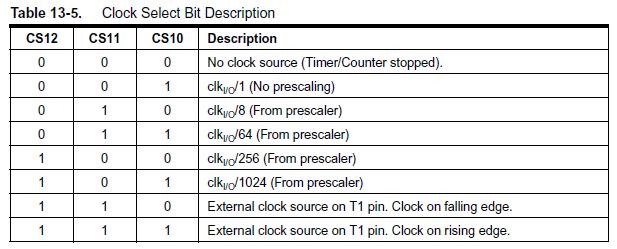
*[](https://www.arxterra.com/wp-content/uploads/2018/08/8a_TCCR1A.jpg)*

*Figure 5a: Timer/Counter Control Register A*

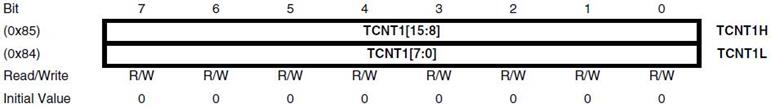
*[](https://www.arxterra.com/wp-content/uploads/2018/08/8b_TCCR1B.jpg)*

*Figure 5: Timer/Counter Control Register B*

*[](https://www.arxterra.com/wp-content/uploads/2018/08/7_WaveformGenerationModeBitDescription.jpg)*

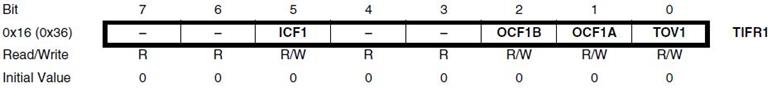
*[](https://www.arxterra.com/wp-content/uploads/2018/08/11_ClockSetBitDescription.jpg)*

* In this mode the Timer/Counter 1 Register (TCNT1H:TCNT1L) counts up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 16-bit value 0xFFFF and then restarts 0x0000.
* There are no special cases to consider in the Normal mode, a new counter value can be written anytime.

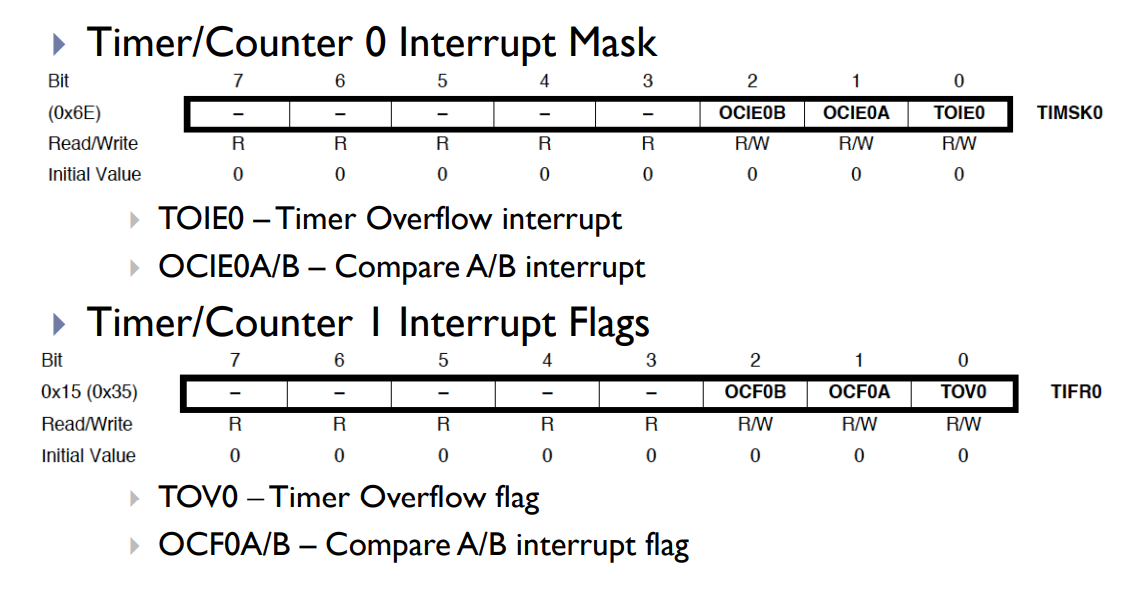
*[](https://www.arxterra.com/wp-content/uploads/2018/08/9_TCNT1.jpg)*

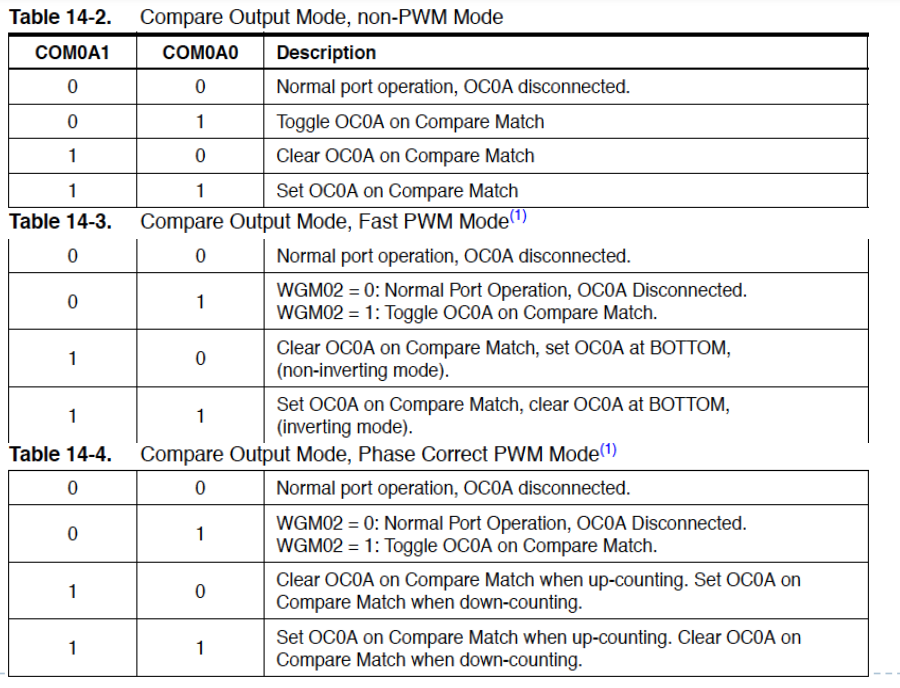
*Figure 6: Timer/Counter 1 Register*

* In normal operation the Timer/Counter Overflow Flag (TOV1) bit located in the Timer/Counter1 Interrupt Flag Register (T1FR1) will be set in the same timer clock cycle as the Timer/Counter 1 Register (TCNT1H:TCNT1L) becomes zero. The TOV1 Flag in this case behaves like a 17th bit, except that it is only set, not cleared.

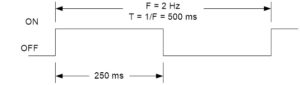
*[](https://www.arxterra.com/wp-content/uploads/2018/08/10_TIFR1.jpg)*

*Figure 7: Timer/Counter 1 Interrupt Flag Registe*





**TIMER/COUNTER 1 NORMAL MODE – DESIGN EXAMPLE**

[](https://www.arxterra.com/wp-content/uploads/2018/08/6_DutyCycle50percent.jpg)

* In this design example, we want to write a 250 msec delay routine assuming a system clock frequency of 16.000 MHz and a prescale divisor of 64.
* The first step is to discover if our 16-bit Timer/Counter 1 can generate a 250 ms delay.

**Variable Definitions**  
tclk\_T1  : period of clock input to Timer/Counter1  
fclk : AVR system clock frequency  
fTclk\_I/O : AVR Timer clock input frequency to Timer/Counter Waveform Generator

**How to Calculate Maximum Delay (Normal Mode)**

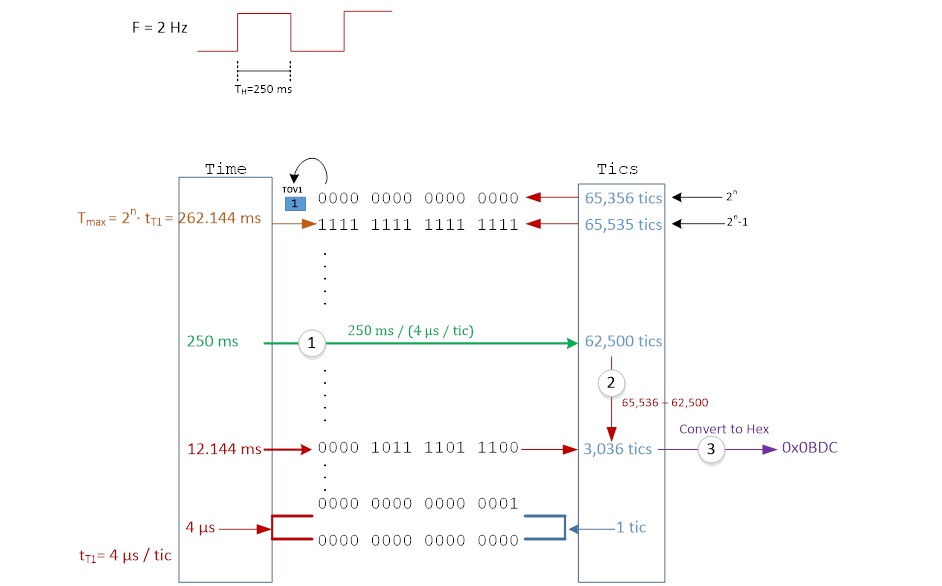
* The largest time delay possible is achieved by setting both TCNT1H and TCNT1L to zero, which results in the overflow flag TOV1 flag being set after 216 = 65,536 tics of the Timer/Counter1 clock.

https://latex.codecogs.com/gif.latex?f_%7bT1%7d&space;=&space;f_%7bTclk_%7bI/O%7d%7d/64, given https://latex.codecogs.com/gif.latex?f_%7bTclk_%7bI/O%7d%7d&space;=&space;f_%7bclk%7d then https://latex.codecogs.com/gif.latex?f_%7bT1%7d&space;=&space;16.000&space;MHz/64&space;=&space;250&space;KHz

and therefore https://latex.codecogs.com/gif.latex?T_%7b1max%7d&space;=&space;65536&space;tics/250&space;KHz&space;=&space;262.14&space;msec

* Clearly, Timer 1 can generate a delay of 250 msec
* Our next step is to calculate the TCNT1 load value needed to generate a 250 ms delay.

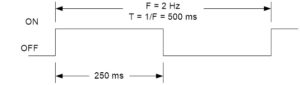
**HOW TO CALCULATE TIMER LOAD VALUE**

*[](https://www.arxterra.com/wp-content/uploads/2018/08/15_TimerLoadValueCalculationWhiteboard.jpg)*

*Figure 9: Process to Calculate Timer Load Value*

**STEPS TO CALCULATE TO TIMER LOAD VALUE (NORMAL MODE)**

**Problem**

[](https://www.arxterra.com/wp-content/uploads/2018/08/6_DutyCycle50percent.jpg)

Generate a 250 msec delay assuming a clock frequency of 16 MHz and a prescale divisor of 64.

**Solution**

1. Divide desired time delay by tclkT1 where tclkT1 = 64/fclkI/O = 64 / 16.000 MHz = 4 µsec/tic  
   250msec / 4 µs/tic = 62,500 tics  
   **short-cut:** TCNT1H = high(-62,500) and TCNT1L = low(-62,500)
2. Subtract 65,536 – step 1  
   65,536 – 62,500 = 3,036
3. Convert step 2 to hexadecimal.  
   3,036 = 0x0BDC  
   For our example TCNT1H = 0x0B and TCNT1L = 0xDC
4. Check Answer  
   3,036 tics x 4 µs/tic = 12.14 msec  
   262.14 msec – 250 msec = 12.14 msec √

**STEPS TO CALCULATE CLOCK DIVISOR (NORMAL MODE)**

In the previous example we assumed a divisor of 64, and then by calculating the maximum delay TMAX verified that this assumption was correct. After that we simply followed the steps defined in the previous slide to calculate the value to be loaded into 16-bit timer/counter TCNT1.

https://latex.codecogs.com/gif.latex?T_%7bMAX%7d&space;=&space;\frac%7b2%5enN%7d%7bf_%7bclk%7d%7d

Where:  
TMAX = maximum delay  
N = divisor  
n = number of flip-flops making-up the timer  
fclk = system clock frequency

But what if we are not given N and need to find TCNT1 for a given delay *tdelay*. In this case we know that *tdelay* ≤ TMAX and applying a little algebra can find an equation for N.

[https://www.arxterra.com/wp-content/uploads/2018/08/Tdelay-Eq..png](https://www.arxterra.com/wp-content/uploads/2018/08/Tdelay-Eq..png)

Let’s take a second look at our 250 msec delay problem. This time we will not assume a divisor of 64. Applying equation 2 we have:

https://latex.codecogs.com/gif.latex?N\geq&space;(250&space;msec&space;\times&space;16MHz)/2%5e%7b16%7d&space;=&space;61.03

From Table 13.5 “Clock Select Bit Description” on page 10, we see that the possible clock divisors are **1, 8, 64, 256, and 1024**. From this list we want to select the divisor that is the closest value, yet greater than or equal to **N**. For our example, not surprisingly the answer is again 64.