**Timer math exercises**

Represent any time solution as a rounded value to 3 significant figures.

When configured with a 1:1 prescaler, how much time goes by in 30,000 TMR1 counts?

1 second 1000 ms 1 clk

------------ \* -------- \* ------- \* 30,000 counts = 1.88ms

16\*10^6 clks 1 second 1 count

Starting at 30,000 how long will it take a 1:1 prescaled TMR1 to roll over?

1 second 1000 ms 1 clk

------------ \* -------- \* ------- \* (216 - 30,000) counts = 2.22ms

16\*10^6 clks 1 second 1 count

How many 1:1 prescaled TMR1 counts go by in 1.5ms?

16\*10^6 clks 1 second 1 count

------------ \* -------- \* ------- \* 1.5ms = 24,000 counts

1 second 1000 ms 1 clk

What starting value will cause a 1:1 prescaled TMR1 to roll over in 3ms?

16\*10^6 clks 1 second 1 count

------------ \* -------- \* ------- \* 3ms = 48,000 starting = (216 - 48,000) = 17,536

1 second 1000 ms 1 clk

How many times would a 1:1 prescaled TMR1 rollover in 100 seconds?

16\*10^6 clks 1 count 1 rollover

------------ \* -------- \* ---------- \* 100 seconds = 24,414 rollovers

1 second 1 clk 216 counts

If a 1:1 prescaled TMR1 starts at 0, what will its count value be after 10ms?

16\*10^6 clks 1 second 1 count rollover

------------ \* -------- \* ------- \* 10ms \* ---------- = 2.441 rollovers

1 second 1000 ms 1 clk 216 counts

216 counts

---------- \* 0.441 rollovers = 28,928 counts

rollover

When configured with a 1:8 prescaler, how much time goes by in 30,000 TMR0 counts?

1 second 1000 ms 8 clk

------------ \* -------- \* ------- \* 30,000 counts = 15 ms

16\*10^6 clks 1 second 1 count

Starting at 30,000 how long will it take a 1:4 prescaled TMR0 to roll over?

1 second 1000 ms 4 clk

------------ \* -------- \* ------- \* (216 - 30,000) counts = 8.88ms

16\*10^6 clks 1 second 1 count

How many counts and what prescaler should you use to measure 150ms on TMR0?

16\*10^6 clks 1 second 1 count

------------ \* -------- \* ------- \* 15ms = 7,500 counts

1 second 1000 ms 32 clk

How many times would a 1:64 prescaled TMR0 rollover in 100 seconds?

16\*10^6 clks 1 count 1 rollover

------------ \* -------- \* ---------- \* 100 seconds = 381 rollovers

1 second 64 clk 216 counts

You are using the following program to measure the duration of a pulse on RA2.

uint16\_t start, end, duration;

while(PORTAbits.RA2 == 1); // Wait for the start of pulse on RA0

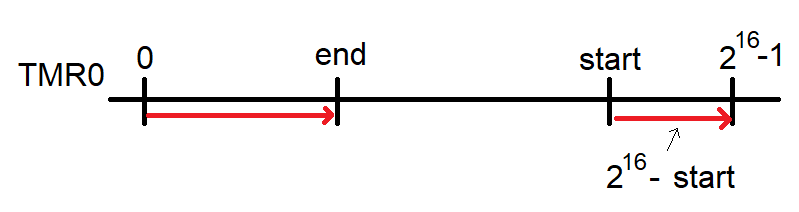
start = TMR0;

while(PORTAbits.RA2 == 0); // Wait for the pulse to end on PTT0

end = TMR0;

duration = end - start;

Normally you would expect start < end and in this case the calculation of duration should make sense. However, if TMR0 rolls over then end > start. Argue (using math and perhaps some graphics), why the calculation of duration is still valid.



If the timer rolls over while waiting for RA2 to equal 0, then end < start. Since subtraction of end - start is equivalent to addition of end with the 2's complement start we can re-write

duration = end - start  as  duration = end + (216 - start)

The second term in this statement (216 - start) is just the number of timer counts from the start of the delay until the timer rolls-over (the timer rolls over when it gets to 216). This is illustrated as the red line on the right side of the number line above. The term end in the rewritten duration expression is just the number of timer counts from 0.