**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?

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

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

------------ \* -------- \* ------- \* (216 – ) counts =

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

------------ \* -------- \* ------- \* 1.5ms =

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

------------ \* -------- \* ------- \* 3ms = starting = (216 -

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

------------ \* -------- \* ---------- \* 100 seconds =

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

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

216 counts

---------- \* rollovers =

rollover

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

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

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

------------ \* -------- \* ------- \* (216 - ) counts = 2

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

------------ \* -------- \* ------- \* 15ms = counts

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

------------ \* -------- \* ---------- \* 100 seconds =

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.