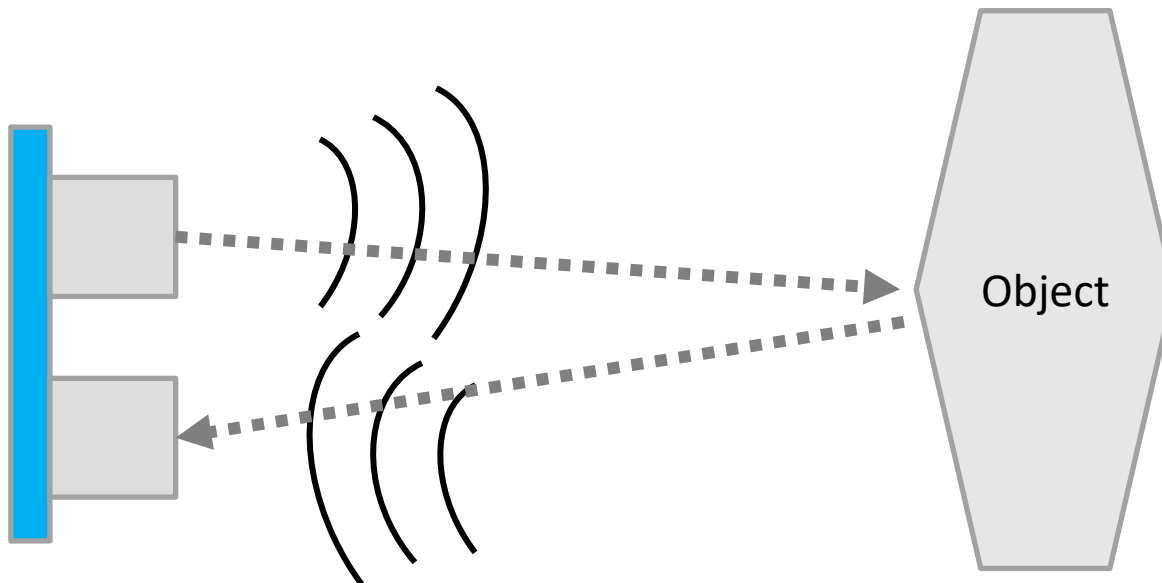


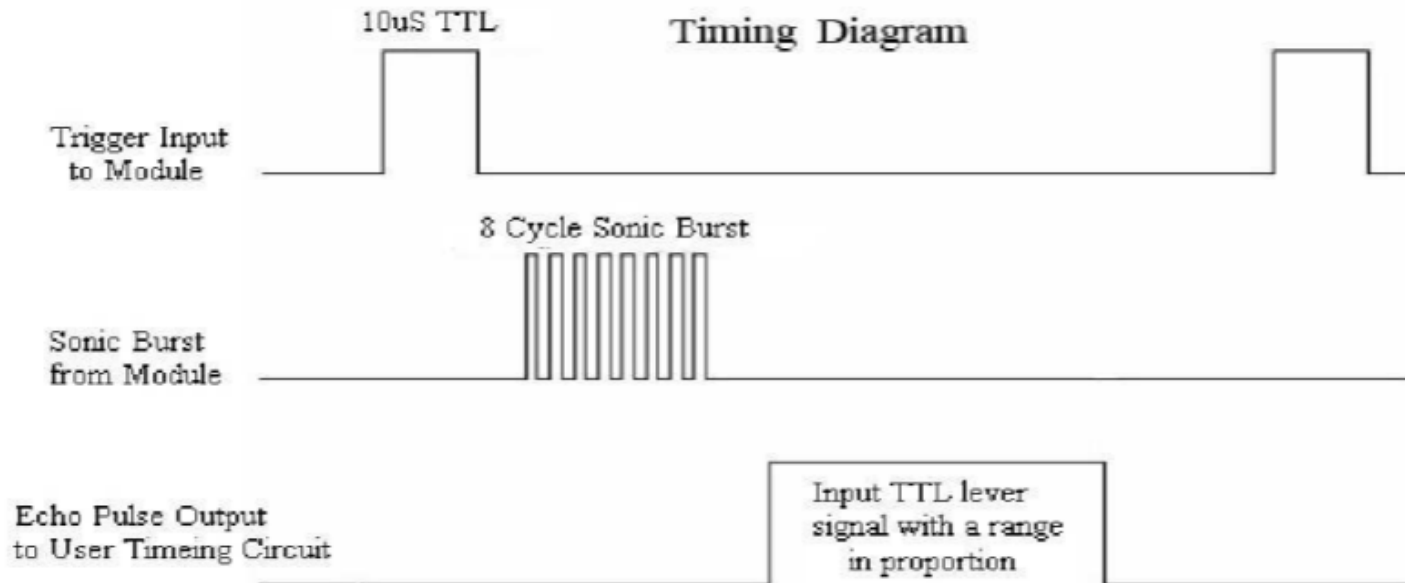
EE 109 Unit H – Timers & Fall 21 Project

Ultrasonic Sensor

- Generates an ultrasonic burst and measures the time for the echo to return



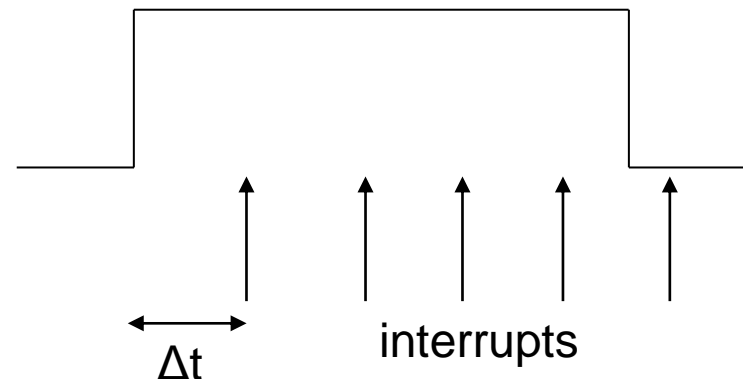
Ultrasonic Input/Output



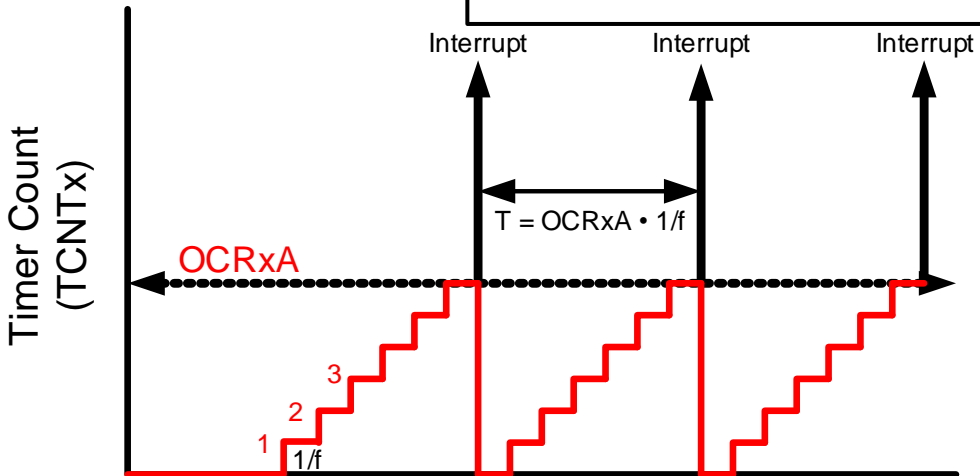
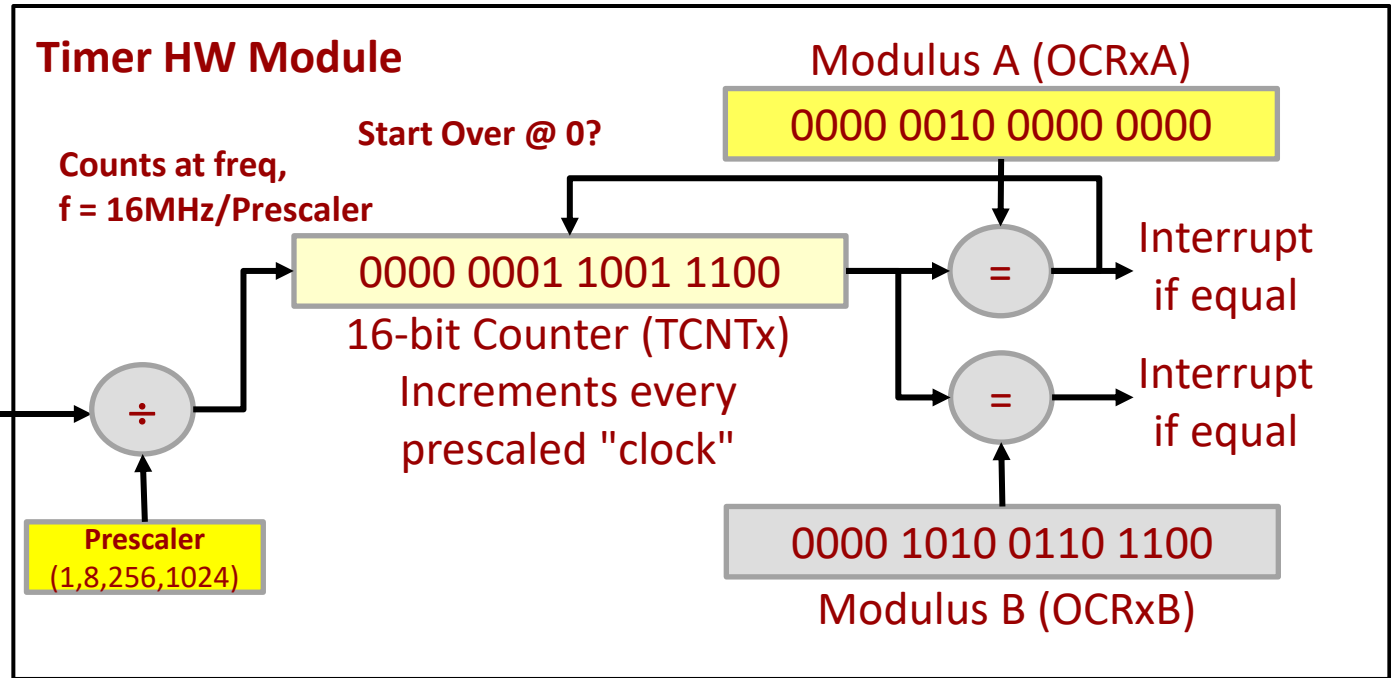
Timers

- In the past, you have used a timer to generate an interrupt at a regular interval (i.e. every 0.1 seconds) by having the timer count to some fixed number (OCR1A), generate an interrupt and then start over at 0 and repeat
- Using that approach your measurement would not be as accurate as it could be.

Return signal to Arduino
from Ultrasonic Ranger



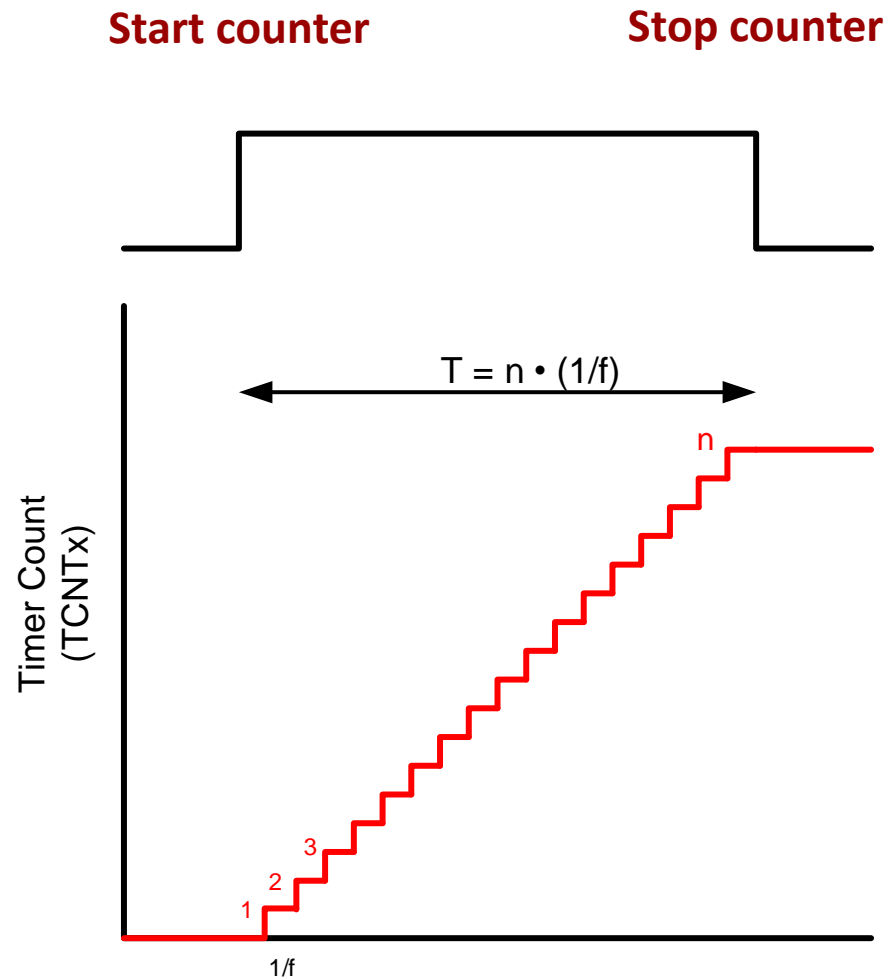
General Overview of Timer HW



We'll just use the modulus A register so you can ignore B for our class

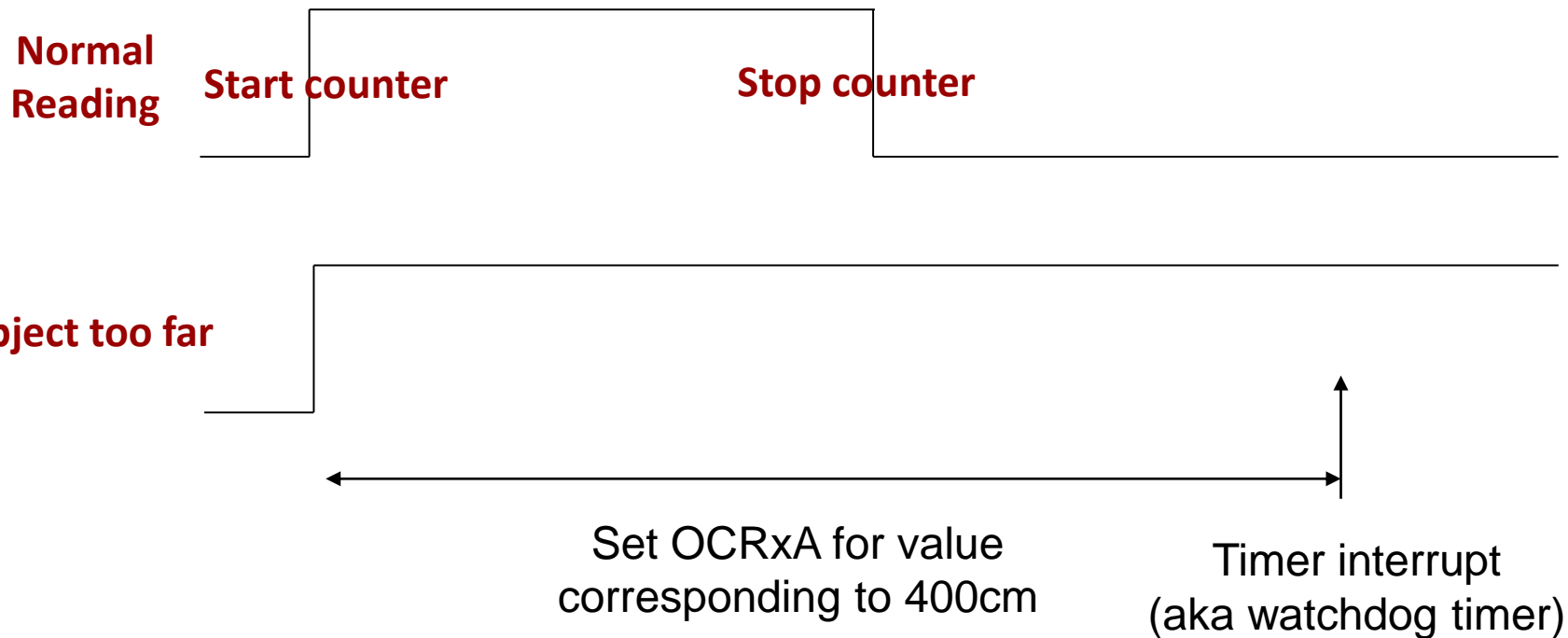
Duration Timer

- Timer can be configured to count at a certain rate (f)
- Start/stop the timer when the microcontroller senses the start/stop of an event
- The count value, n , can determine the duration of the event: $T = n \cdot (1/f)$
- Can use pin change interrupts to detect start and stop condition



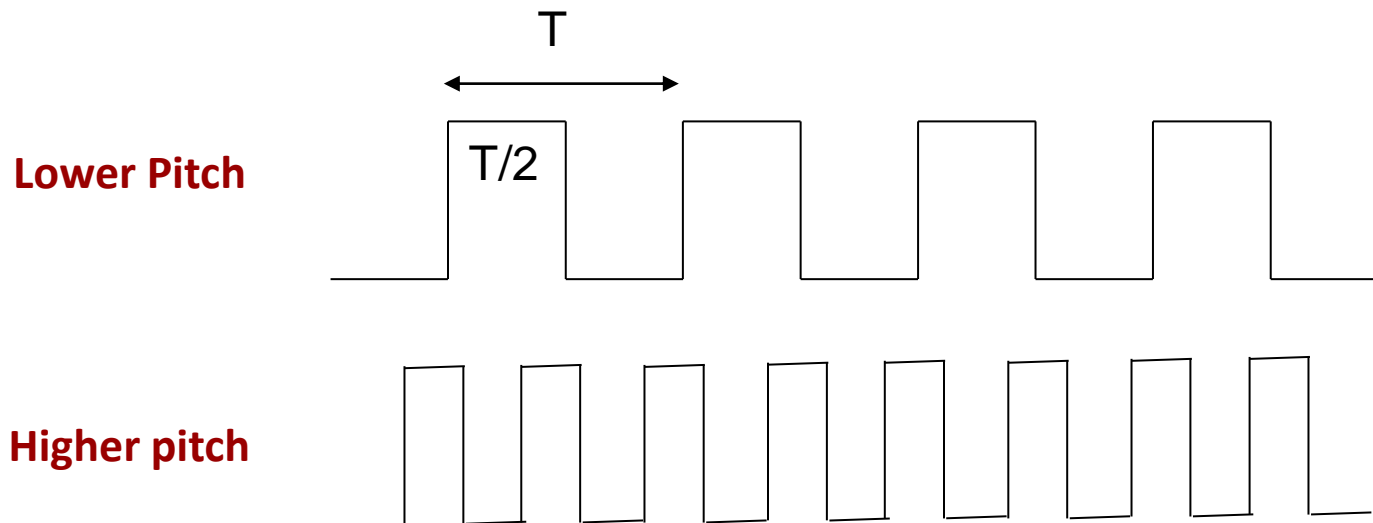
Watchdog Timer

- If no object is in range or an error occurs we may not get (or miss) the falling signal and just keep counting...let's set a maximum timeout value (aka watch dog timer) to interrupt us if we've waited the time equivalent to 400 cm of distance



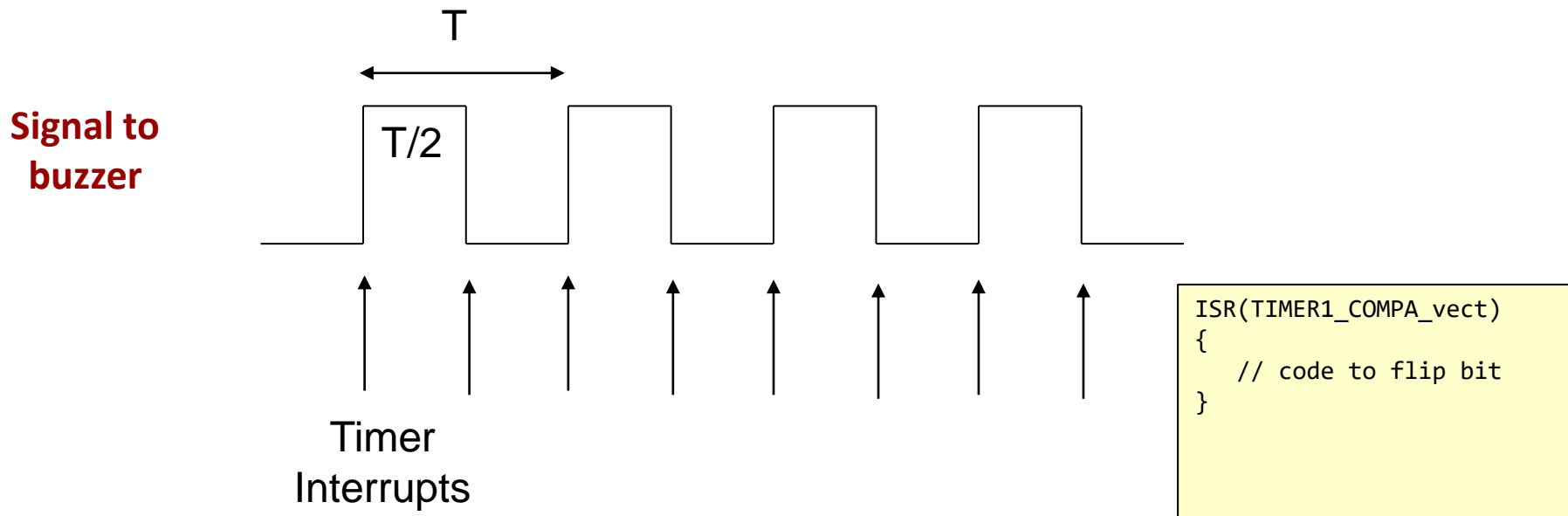
Play Notes

- Recall that to play a certain tone we generate a square wave to our buzzer at a particular frequency
 - The frequency (how rapidly we alternate the signal) determines the pitch of the tone



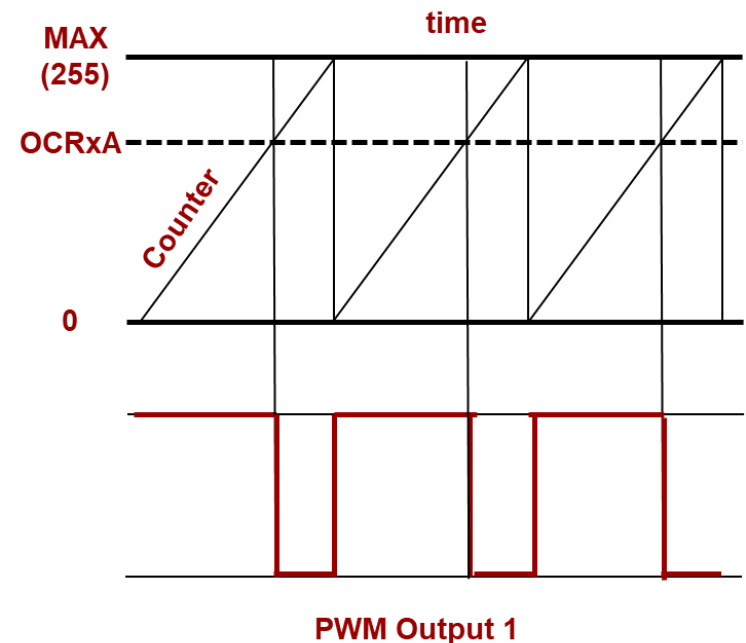
Using Timer to Generate Tones

- Rather than use delay statements to pass the time between alternating the signal high and low, we can use a timer interrupt so that the processor can do other work
- Use a timer interrupt set to a value for the desired $T/2$



PWM Signals

- The timers on the Arduino are built to be able to flip a specific output bit automatically based on the timer count and the OCRxA or OCRxB registers so that you don't have to write an ISR and can just let the HW do the work
- As an example usage, we can use the ability to generate different intensities for an RGB LED or for the rotation speed of a motor
- See separate unit on PWM



Timer Summary

- The Arduino has 3 timers:
 - Timer0 and Timer2 are 8-bit timers (can only count up to 255)
 - Timer1 is a 16-bit timer
- Project Fall 21
 - Timer0: Buzzer
 - Timer1: Echo signal length and watchdog timer
 - Timer2: PWM