Arduino UNO Timer and Interrupts

What is timer interrupts?

- timer interrupts allow a task to be performed at a very specific moment in time
- `loop()` cannot do that because it is difficult to tell how long a statement takes
- Arduino timer interrupts pause the normal sequence of events and execute the set of commands specified
- interrupts are useful for measuring an incoming signal at equally spaced intervals, sending out a signal periodically etc.

Clear Timer on Compare Match (CTC Mode)

- There are three timers in Uno called timer0, timer1, and timer2
- Each timer has a counter and is incremented on each clock tick
- interrupt is triggered when the counter reaches a specific value, stored in the compare match register
- the counter will be reset to 0 after reaching the value
- how often interrupts occur depend on the compare match value

Timer Interrupt Program Structure

- In `setup()`
  - setup up the right frequency of the timer by specifying the parameters appropriately
- define interrupt function
  - The interrupt function of Timer***X*** is `ISR(TIMERX_COMPA_vect)`
  - e.g. the interrupt function of Timer1 is `ISR(TIMER1_COMPA_vect)`
  - put the tasks you want to perform periodically inside the interrupt function
- In `loop()`
  - no statement is needed here if everything you want to do is in the interrupt function

Timer Parameters

- clock speed
    - Arduino clock runs at 16MHz
    - fastest speed the interrupts can occur (once every 1/16,000,000 second)
- maximum counter value (when to trigger interrupt)
    - timer0 and timer2 are 8 bit timers (maximum counter value = 255)
    - timer1 is 16-bit (at most 65535)
    - e.g. suppose we set the counter value of timer1 = 65535. Interrupts will occur every 65536/16,000,000 seconds (~4ms) in timer1 if we increment the timer in every tick
- prescaler (slow down counter increment)
    - timer speed (how often the counter increments) = Arduino clock speed / prescaler

Setting the right value

desired interrupt frequency

= Arduino clock speed/ (prescaler*(compare match register + 1))

compare match register

= Arduino clock speed/ (prescaler*desired interrupt frequency) – 1

e.g. we want an interrupt every second (1Hz). Let the prescaler = 1024

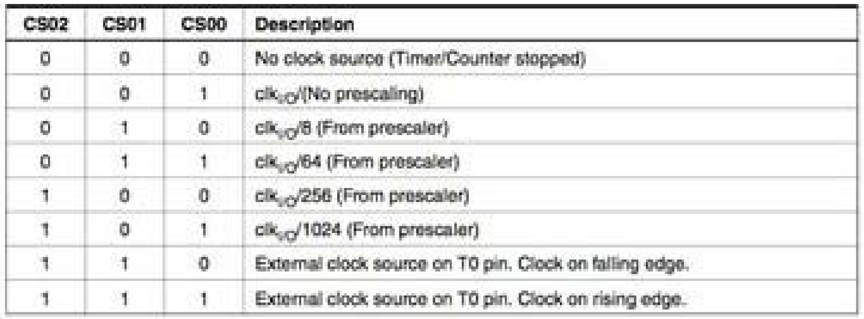
compare match register = [16,000,000 / (1024*1)] – 1 = 15624

Timer1 should be used

Prescaler

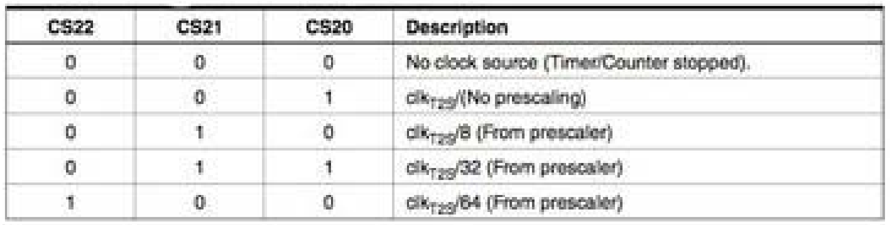
- prescaler set by using appropriate values
- Each timer has its own combinations

Timer 0



CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Timer 2



CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk_{T2S}/(No prescaling)
0	1	0	clk_{T2S}/8 (From prescaler)
0	1	1	clk_{T2S}/32 (From prescaler)
1	0	0	clk_{T2S}/64 (From prescaler)

Timer 1

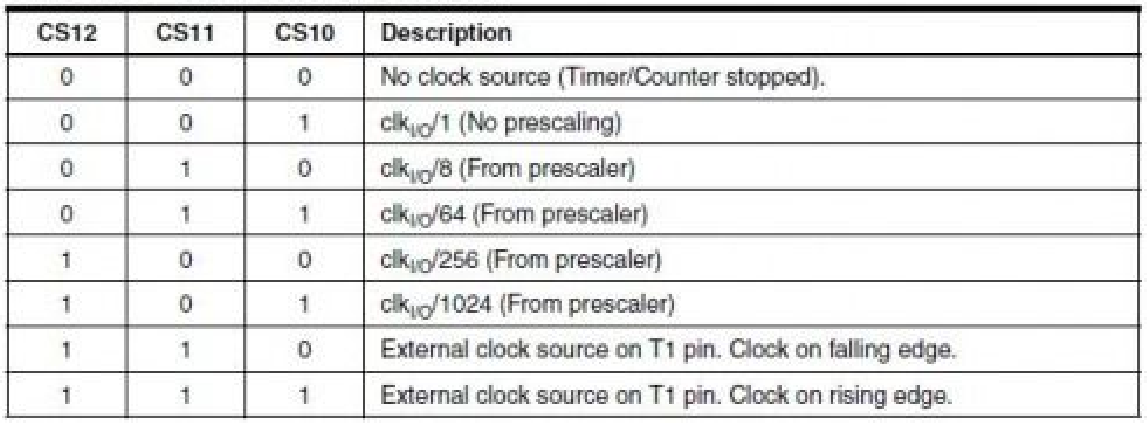


Table 16-5. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/1 (No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Reference: https://www.instructables.com/id/Arduino-Timer-Interrupts/

Timer setup code (1)

- Reset the timer (***X*** refers to the timer number)
  - `TCCRXA = 0;    // e.g. TCCR0A = 0; for timer0`
  - `TCCRXB = 0;`
  - `TCNTX = 0;`
- set the compare match register
  - `OCRXA = value;   // e.g. OCR0A = 124;`
- turn on CTC mode
  - `TCCR0A |= (1 << WGM01);    //for timer0`
  - `TCCR1B |= (1 << WGM12);    //for timer1`
  - `TCCR2A |= (1 << WGM21);    //for timer2`

Timer setup code (2)

- set prescaler
  - `TCCR2B |= (1 << CS22);`
    ```
           // Timer 2; prescaler = 64
    ```
  - `TCCR1B |= (1 << CS11);`
    ```
           // Timer 1; prescaler = 8
    ```
  - `TCCR0B |= (1 << CS02) | (1 << CS00);`
    ```
           // Timer 0; prescaler = 1024
    ```
- enable timer compare interrupt
  - `TIMSKX |= (1 << OCIEXA);`
    ```
                // e.g. TIMSK0 |= (1 << OCIE0A);
    ```