Lecture 16 CS3514 02/11/17

Timer prescaling and clear Timer on Compare (CTC)

Prescaling: Divides the clock by various powers of 2, thereby increasing the timer period.

The maximum prescaling value is 1024. That is achieved by setting CS10 and CS12

```
TCCR1B |= (1 << CS10) ;
TCCR1B |= (1 << CS12);
```

This gives a timer period of $1/16 \times 10^6 \times 1024 = 6.4 \times 10^{-5}$ sec (per count)

$\therefore$ the timer will overflow every $65,535 \times 6.4 \times 10^{-5}$ sec

$$= 4.194 \text{ Seconds}$$

CTC mode

If we want our timer to trigger an Interrupt after 1 Second, say. Then we need to use clear Timer on Compare match mode.

Instead of Counting until an overflow occurs, the timer Compares its Count to a value stored in a register. When they are equal, the timer can set a flag or trigger an Interrupt.

To use CTC to trigger an Interrupt after a Specific time t ($t <= 4.194$ sec). We need to figure out how many

Counts we need and which prescaler we should use.

To trigger an Interrupt after 1 second and assuming the 1024 prescaler:

$$\text{target time} = (\text{timer resolution}) * (\#\ \text{Timer Counts})$$

$$\therefore \#\ \text{Time Counts} = \text{target time} / \text{timer resolution}$$

$$= 1\ \text{sec} / 6.4 \times 10^{-5}\ \text{sec}$$

$$= 15625$$

In CTC mode, when the timer matches our desired count, it will reset itself to zero. This takes 1 clock cycle. If we are not using a prescaler, we should take this time into account by subtracting 1 from the number of timer counts in our initial equation.

```
Void Setup() {
  pinMode (ledpin, output);
  cli();
  TCCR1A = ∅;
  TCCR1B = ∅;     Timer-off
  OCR1A  = 15625;
  TCCR1B |= (1 << WGM12); //turn on CTC mode
  TCCR1B |= (1 << CS10);   //Prescale using 1024
  TCCR1B |= (1 << CS12);   //   "      "     "
  TIMSK1 |= (1 << OCIE1A); //enable timer Compare
                           | Interrupt
  sei();
}
```

```
ISR( Timer1_CompareA_vect) {
                                          ?
     digitalwrite( ledpin, !digitalRead(ledPin));
}
```

To trigger the call to a function at time Intervals ≥ 4 seconds we can use a counter variable. For example, to trigger an interrupt of 10 seconds:

```
ISR( Timer1_CompareA_vect) {
        Second ++;
        if (seconds == 10) {
              seconds = 0;                  ?
              digitalwrite( ledpin, !digitalRead(ledpin));
        }
}
```

Remember: Seconds must be declared as volatile:

```
Volatile int Seconds
```