

pic_timer.c – PIC TIMER0 pseudo code

module function prototypes:

```
// initialize timer module
void init_timer(void);

// start timer
void startTimer(long ms);

// reset timer
void resetTimer(void);

// is timer expired
char isTimerExpired(void);

// increment overflows
void incrementOverflows(void);
```

private function prototypes:

```
// see how much time has elapsed since the timer started
static long getTimeElapsed(void);
```

global variables:

```
unsigned int g_overflows; // counter of timer overflows
```

module variables:

```
static long m_ms; // timer length
static char m_started; // whether the timer has started or not
```

pseudocode

```
void init_timer(void)
    set g_overflows = 0    // have not overflowed initially
    set m_started = 0     // timer hasn't not started yet
    set prescale to 64 (OPTION_REG = 0x05)
```

```
clear interrupt flag (TOIF = 0)
enable Timer0 interrupts (TOIE = 1)
end init_timer
```

```
-----
void startTimer(long ms)
    if (timer not started yet)    // m_started == 0
        m_ms = ms;
        started = 1;             // m_started = 1
    end if
end startTimer
```

```
-----
void resetTimer(void)
    reset overflows                // g_overflows = 0
    timer unstarted                // m_started = 0
end resetTimer
```

```
-----
char isTimerExpired(void)
    if (timer has started)        // m_started == 1
        if (m_ms < time elapsed) // if ( m_ms < getTimeElapsed() )
            return 1
        end if
        else
            return 0
        end else
    end if
    else
        return 0
    end else
end isTimerExpired
```

```
-----
void incrementOverflows(void)
    g_overflows++
end incrementOverflows
```

```
-----  
static long getTimeElapsed(void)  
    long timeElapsed = number of overflows * overflow period ; // g_overflows * timer  
                                                                overflow period (3.2  
                                                                ms)  
    return timeElapsed;  
end getTimeElapsed  
-----
```