

pic_timer2.c – PIC TIMER2 pseudo code

module function prototypes:

```
// initialize timer module
void init_timer2(void);

// start timer
void startTimer2(long ms);

// reset timer
void resetTimer2(void);

// is timer expired
char isTimerExpired2(void);

// increment overflows
void incrementOverflows2(void);
```

private function prototypes:

```
// see how much time has elapsed since the timer started
static long getTimeElapsed2(void);
```

global variables:

```
unsigned int g_overflows2; // counter of timer overflows
```

module variables:

```
static long m_ms2; // timer length
static char m_started2; // whether the timer has started or not
```

pseudocode

```
void init_timer 2(void)
    set g_overflows2 = 0 // have not overflowed initially
    set m_started2 = 0 // timer hasn't not started yet
    set prescale on, prescaler to 4, postscaler to 10 (T2CON = 0x25)
```

```
clear interrupt flag (TMR2IF = 0)
enable Timer0 interrupts (TMR2IE = 1)
end init_timer2
```

```
-----
void startTimer2(long ms)
    if (timer not started yet)    // m_started2 == 0
        m_ms 2= ms;
        started = 1;            // m_started2 = 1
    end if
end startTimer2
```

```
-----
void resetTimer2(void)
    reset overflows                // g_overflows2 = 0
    timer unstarted                // m_started2 = 0
end resetTimer2
```

```
-----
char isTimerExpired2(void)
    if (timer has started)        // m_started2 == 1
        if (m_ms2 < time elapsed) // if ( m_ms2 < getTimeElapsed2() )
            return 1
        end if
    else
        return 0
    end else
end isTimerExpired2
```

```
-----
void incrementOverflows2(void)
    g_overflows2++
end incrementOverflows2
```

```
static long getTimeElapsed2(void)
    long timeElapsed = number of overflows * overflow period ; // g_overflows2 *
                                                                timer overflow period
                                                                (1ms)

    return timeElapsed;
end getTimeElapsed2
```
