Timer 2
The TImer2 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter
- Readable and writable
- Software programmable Prescaler/PostScaler up to 1:16
- Interrupt on overflow from FFh to 00h

Timer2 Registers

The below table shows the registers associated with the PIC16f877A Timer0 module.
Register
T2CON This registers is used to configure the TIMER2 Prescaler, Clock Source, etc
TMR2This register holds the timer count value which will be incremented depending on pre-scaler configuration
PIR1This register contains the Timer2 overflow flag(TMR2IF).
PIE1This register contains the Timer2 Interrupt Enable flag(TMR2IE).

T2CON Register 8 bits
7 -
6 TOUTPS3
5 TOUTPS2
4 TOUTPS1
3 TOUTPS0
2 TMR2ON
1 T2CKPS1
0 T2CKPS0

TOUTPS3:TOUTPS0: Timer2 Output Postscale Select bits
0000 = 1:1 postscale
0001 = 1:2 postscale
0010 = 1:3 postscale
•
•
1111 = 1:16 postscale
TMR2ON: Timer2 On bit
1-Timer2 is on
0-Timer2 is off
T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

Generating 1sec delay using Timer2:
As the timer2 is 8-bit and supports 1:16 prescalar, it is not possible to directly generate the delay of 1sec. The max delay with 1:16 prescaler will be:
Delay = 256 * (Prescaler*4)/Fosc = 256 * 16*4/20Mhz=819us
Now 500us can be generated using timers which will be used to increment a counter 2000 times

to get 1sec delay.
Delay Calculations for 500usec @20Mhz with Prescalar as 16:
RegValue = 256-(Delay * Fosc)/(Prescalar*4)) = 256-((500us* 20Mhz)/(16*4)) = 256-156=100

Below are the steps for configuring and using the Timer2 for delay generation:

1. Calculate the Timer Count for the required delay.
2. Set the Prescaler bits in **T2CON** as per the delay calculations.
3. Load the timer value into **TMR2** register.
4. Enable the Timer2 Interrupt by setting **TMR2IE** bit
5. Enable the Global and Peripheral interrupts by setting **GIE** and **PIE** bits
6. Finally start the Timer2 by setting **TMR2ON** bit