MICROCHIP

Regional Training Centers

Section 5
Timer

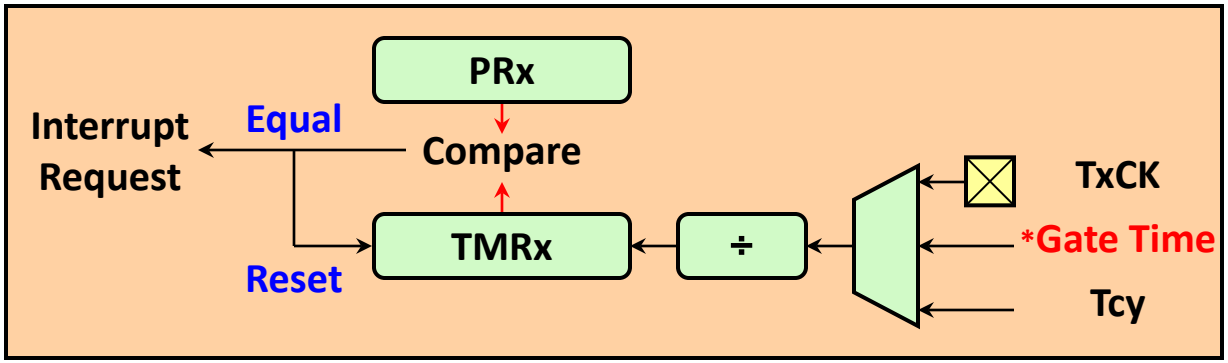
MICROCHIP
Regional Training Centers

What's Timer ?

- Timer簡單的說就是一個會持續不斷的計算進入Timer中Clock數量的模組。
- Timer一般會有兩種稱呼方式Timer或Counter。當Clock的頻率未知時,僅能得知計數到多少個Clock,所以稱為Counter。如果知道Clock的頻率,就可以換算出時間,所以稱為Timer。其實都是一樣的東西。
- 計數值可能是遞增或遞減。Microchip 16-bit MCU的Timer僅支援遞增。也就是會從"0"開始計數。模組可以預先設定當數到多少時要通知CPU。例如可以設定數到1,000個Clock後,通知CPU。
- 此處指的通知,就是中斷旗標(Interrupt Flag) 。當數到設定值時,中斷旗標會被設定,對CPU發出中斷請求 (Interrupt Request)。如果此時中斷是致能的,就會進入中斷服務常式。

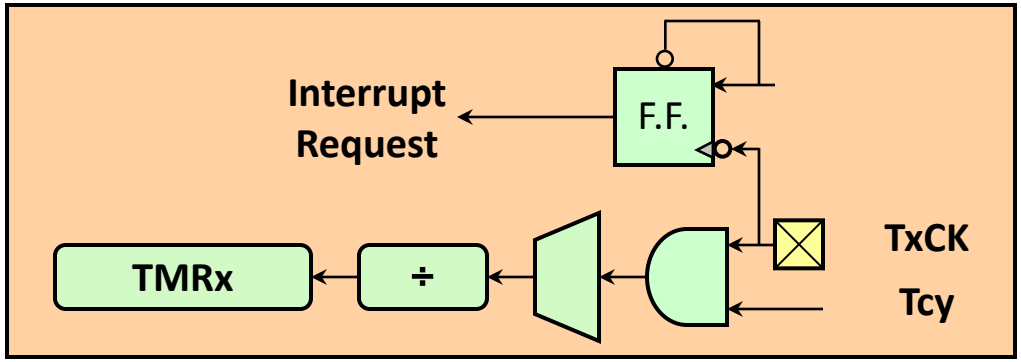
16-Bits Timers

- 16-Bits MCU的Timers方塊圖, 如圖所示。
- Clock可以選擇內部,或者由外部接腳輸入。經過預除器後,送入TMRx。TMRx從"0"開始遞增,Compare會不斷比較PRx與TMRx的值,相同時發出中斷需求,並且將TMRx的值歸零。
- 舉例來說,假設Tcy是1mS,想要計算1S的時間。則可將Clock設為Tcy,預除器設為除1, PRx設為1,000。如此一來,每次數到1,000 Clock時(1S),Timer就會清除TMRx,設定中斷旗標,發出中斷需求。
- 如果中斷有致能，就會進入中斷服務常式。或者透過輪詢(Polling)中斷旗標得知。

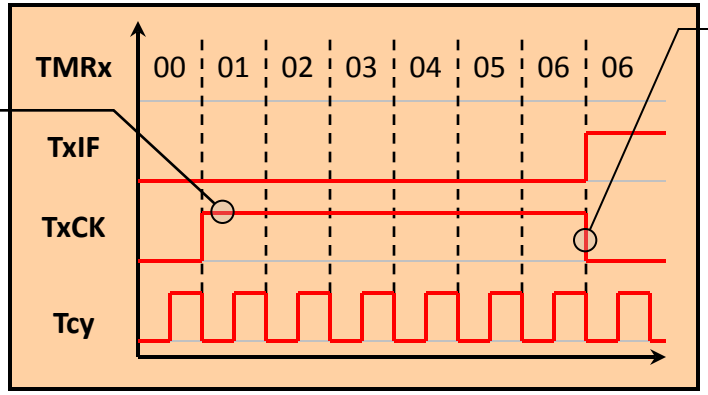


Gate Time Function

- 16-Bits MCU的Timers還有一個叫閘控計時(Gate Time)的特殊功能。可以用來計數外部輸入訊號的寬度。
- TxCK的輸入為 High時,Tcy可以被Timer計數,一直持續到TxCK的輸入由High變Low時停止。同時設定中斷旗標,發出中斷需求。



TxCK為High,AND Gate動作,Tcy可進入TMRx。



負緣觸發正反器,設定中斷旗標,發出中斷需求。

MICROCHIP
Regional Training Centers

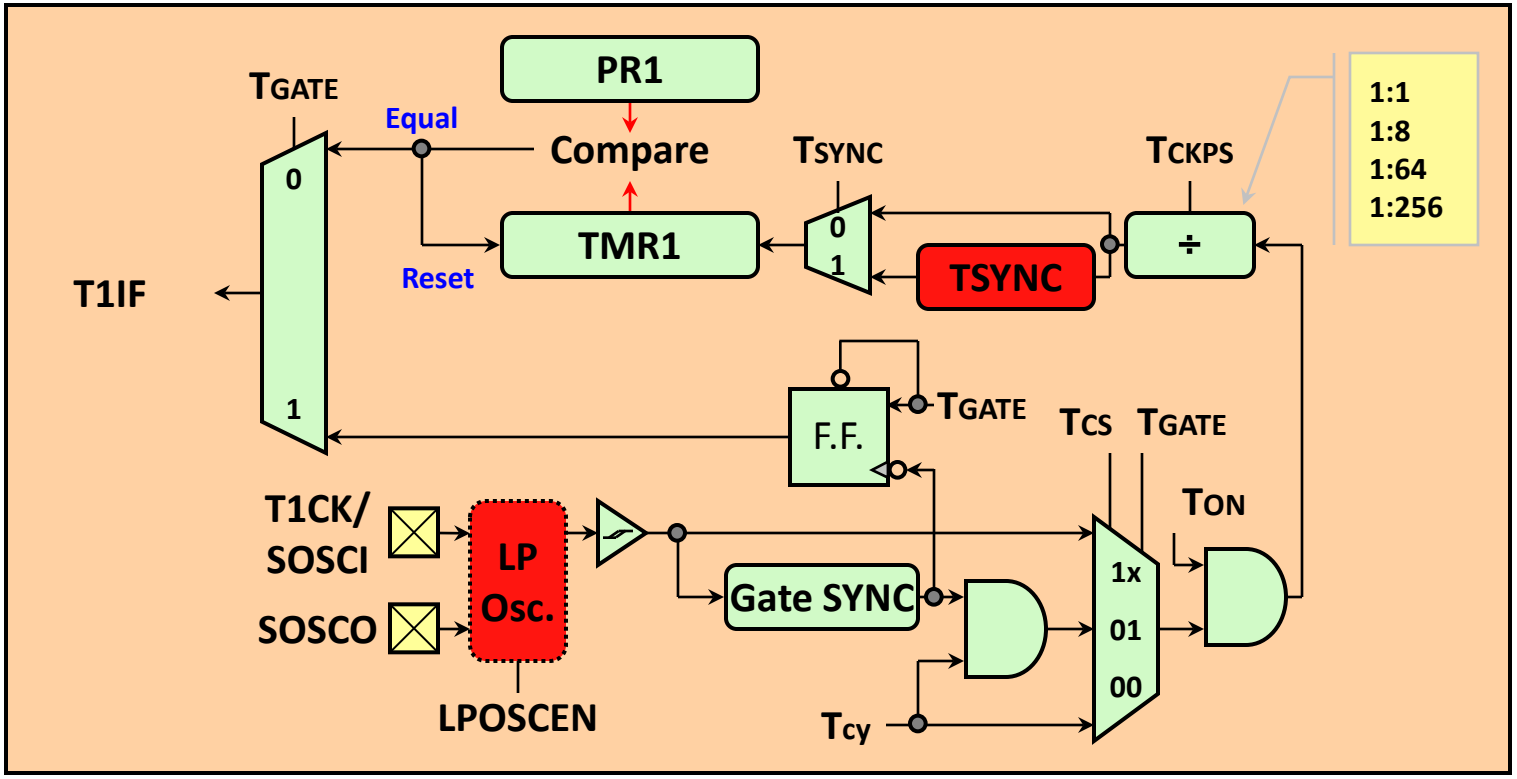
16-Bits Timers Introduction

- 16-Bits MCU具有5個General Purpose Timers/Counters。這5個Timers/Counters的功能與操作方式,幾乎相同,只有少許差異。
- Timer是16-Bits,所以計數範圍0~65,535。TMRx, PRx都不可超過。
- 預除器可以用來擴大計數範圍。舉例來說,如果預計TMRx要數1,000。代表必須有1,000個Clock進入TMRx。以一個Clock 1mS來看,則是1S。此時如果把預除器設為除8(1:8),則變成Clock每8個才會有一個進入TMRx,意即TMRx數到1,000時,實際的Clock已經產生了8,000個,經過了8S。

- 5個Timer的實際差異在哪？

Type A Timer (Timer1)

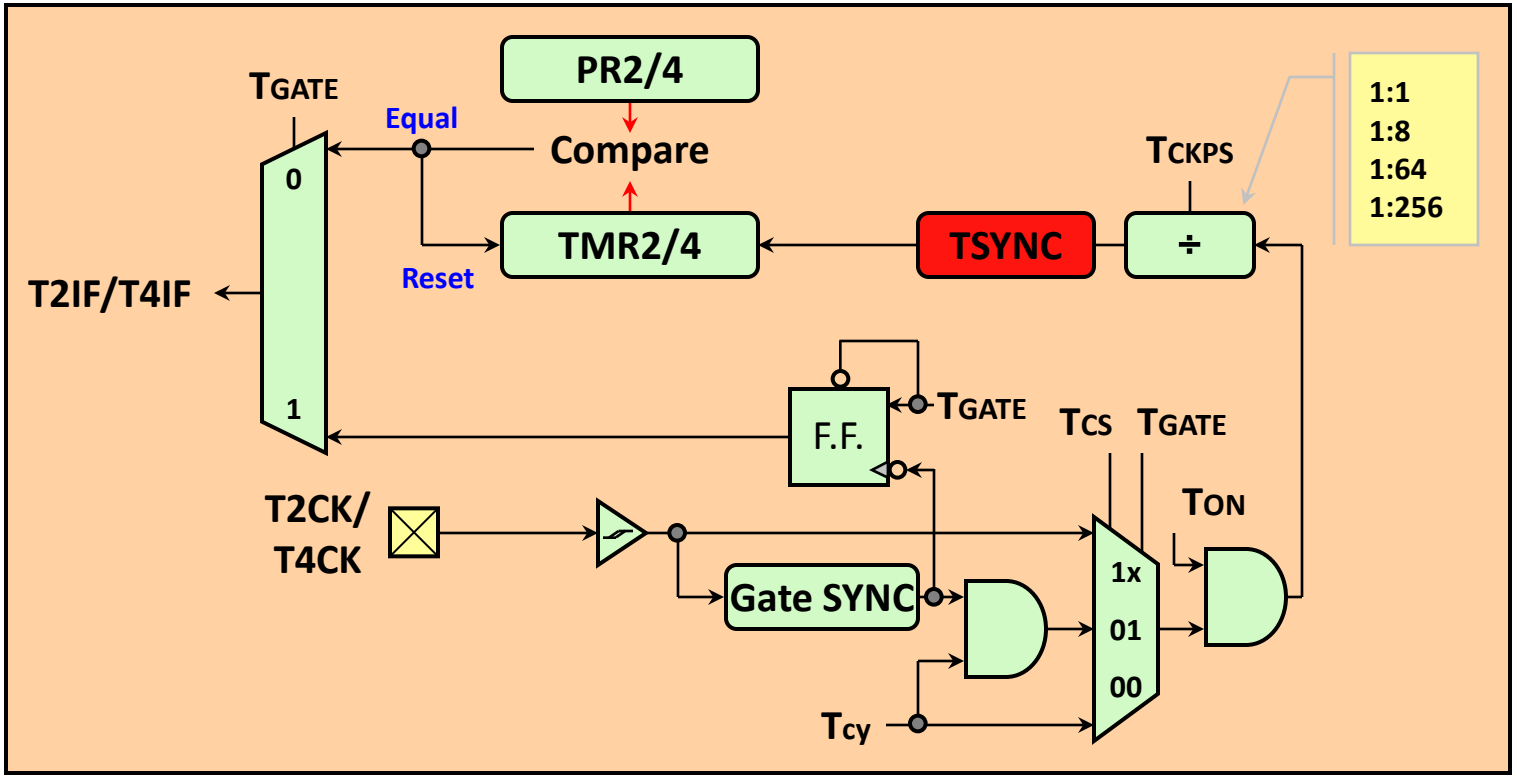
- Timer1方塊圖如下。Timer1最大的特點,是可選擇同步與否,與內建晶體振盪電路,可以直接連接外部的Low Power Crystal, 如:32.768K Hz,做RTCC。



MICROCHIP
Regional Training Centers

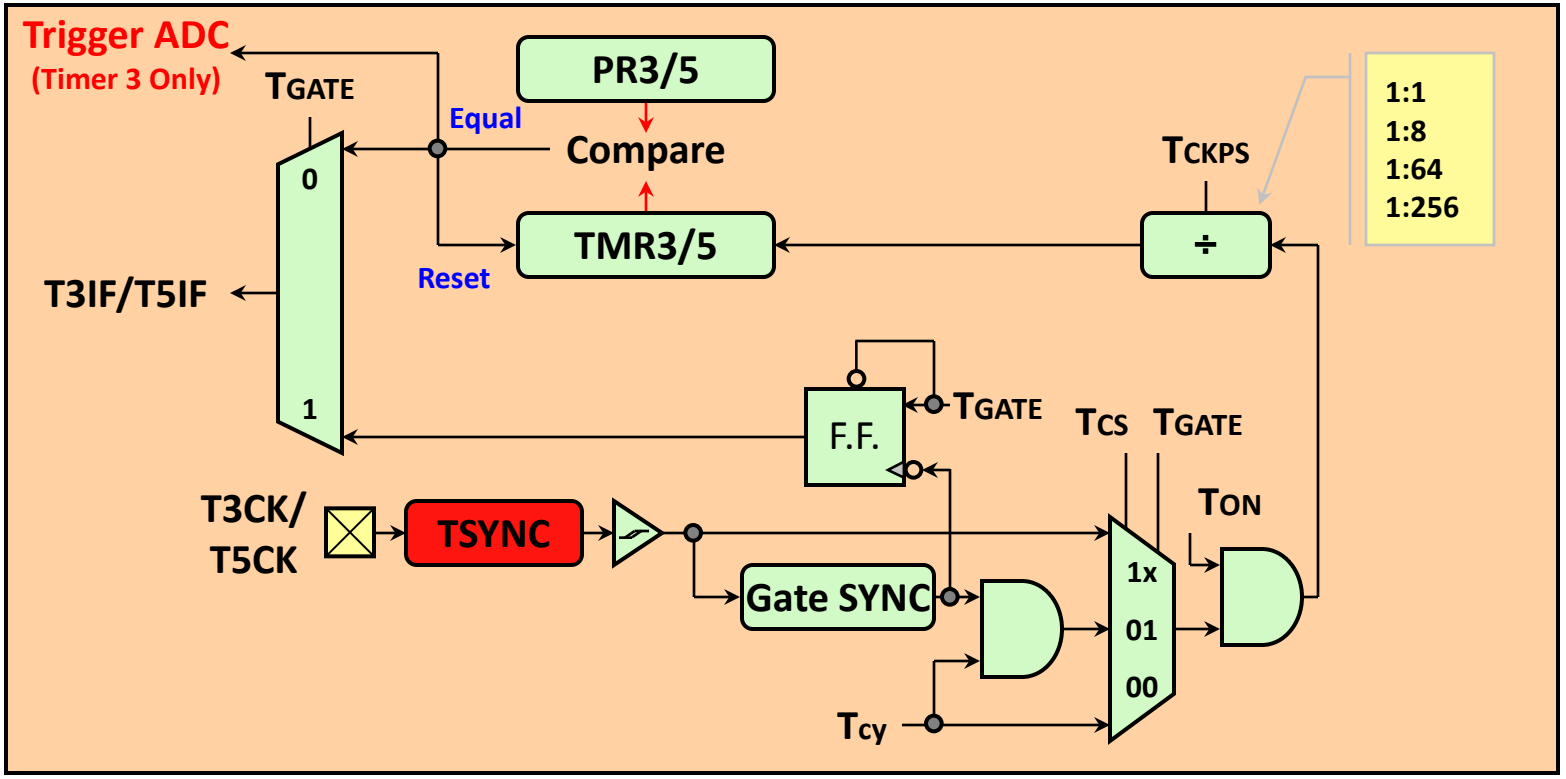
Type B Timer (Timer2,4)

- Timer2,4方塊圖如下。Timer2,4與Timer1的差異,就是強制與系統時脈同步,如果要計數外部訊號時,必須連接有明確正負緣狀態的方波,不可以直接連接Crystal。



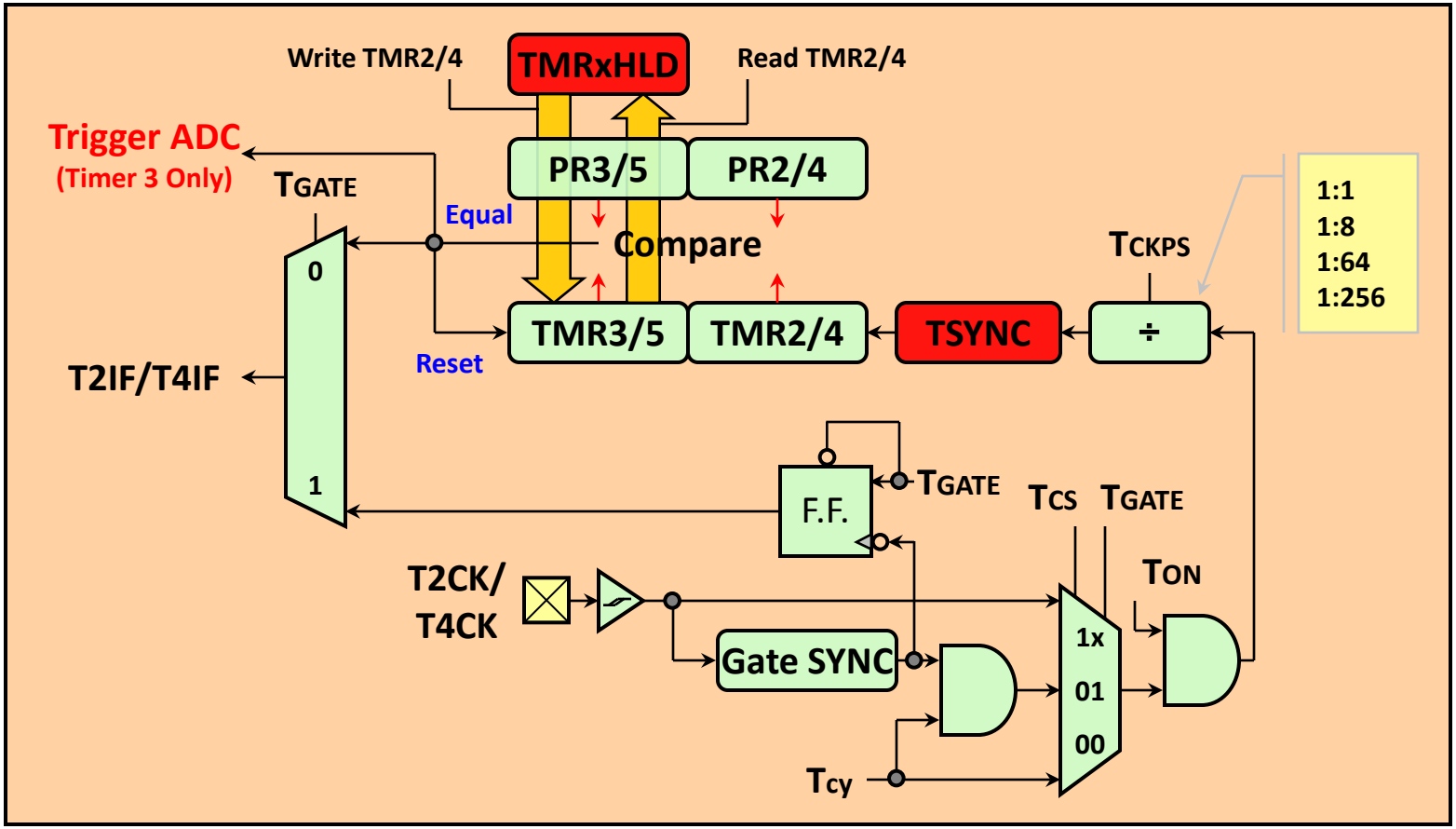
Type C Timer (Timer3,5)

- Timer3,5方塊圖如下。 Timer3,5與Timer2,4的差異,在於同步的點不同。同步點的不同會影響可輸入訊號的最高頻率。
- Timer3/5還可以作為AD轉換的觸發來源。



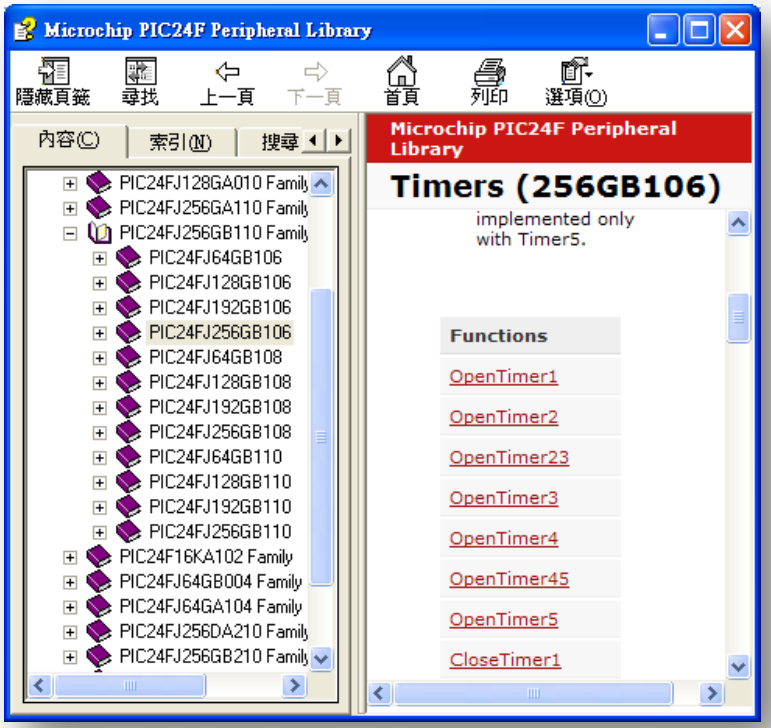
32-Bits Timer

- 兩組16-Bits Timer(Type B+C)可以組合成一組32-Bits的Timer。



MPLAB C30對Timer的支援

- MPLAB C30提供許多Timer Function可供使用。要使用Timer時先含入(include)Timer的標頭檔。
  Ex:#include <timer.h>
- 在C:\ProgramFiles\Microchip\mplabc30\vx.x\docs\periph_lib\Microchip PIC24F Peripheral Library.chm 檔案中可以找到詳細的使用說明。
- 使用Timer Function前,最好先閱讀下Function的說明。



MICROCHIP
Regional Training Centers

MPLAB C30 Timer Function

- MPLAB C30中常用的Timer Function
  OpenTimerx( ); // 啟用Timerx,設定Timerx工作模式。
  ReadTimerx( ); // 讀取TMRx。
  WriteTimerx( ); // 寫入TMRx。
  CloseTimerx( ); // 關閉Timerx。
  ConfigIntTimerx( ); // 致能Timerx的中斷,並設定中斷優先權。
- 32-Bits Timer模式的Function
  OpenTimerxy( ); // 啟用32-Bits Timerxy,設定Timer工作模式。
  ReadTimerxy( ); // 讀取32-Bits Timerxy的TMRx。
  WriteTimerxy( ); // 寫入32-Bits Timerxy的TMRx。
  CloseTimerxy( ); // 關閉32-Bits Timerxy。
  …

MICROCHIP
Regional Training Centers

Timer1 Example

- Timer1的初始化範例:

  void OpenTimer1(unsigned int *config* , unsigned int *period*);

  config:Timer的工作模式, *period*:PRx的值。

  Ex:

```
OpenTimer1( T1_ON & T1_IDLE_CON & T1_GATE_OFF & T1_PS_1_256 &
            T1_SYNC_EXT_OFF & T1_SOURCE_INT , 1000 );
```

- Timer1的初始化後,可以利用Polling T1IF(Timer的中斷旗標)的方式,來得知Timer是否計數到預期值或者開啟中斷來得知。

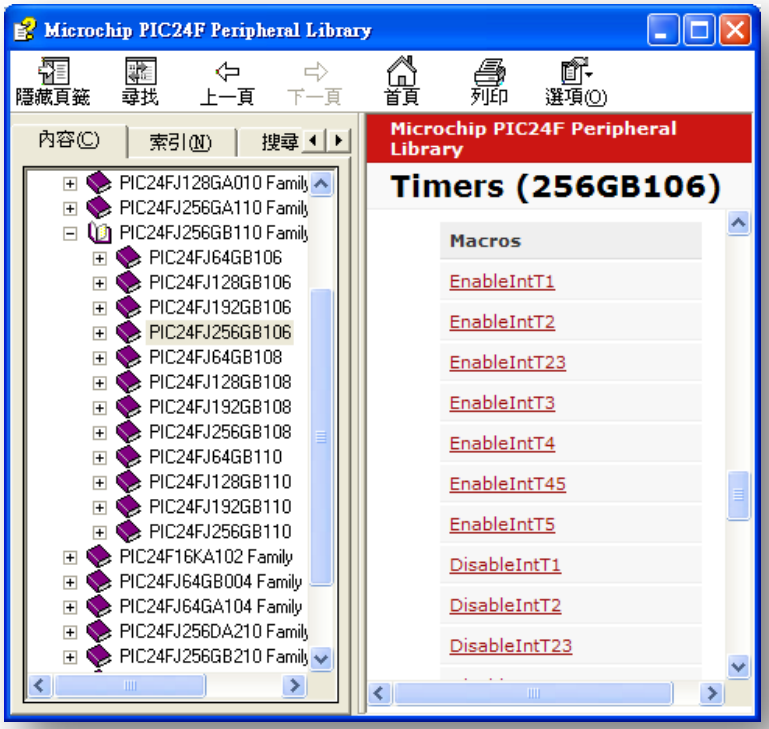
  Ex:

```
if( IFS0bits.T1IF == 1 )
{
  IFS0bits.T1IF = 0;
  ....
}
```

- 相關的定義必須參考 “timer.h”及Function的說明文件。

MPLAB C30 Timer Macro

- MPLAB C30針對Timer也設計了幾個方便使用的巨集(Macro):
  EnableIntTx; // 開啟Timerx的中斷。
  Ex:EnableIntT1;
  DisableIntTx; // 關閉Timerx的中斷。
  Ex:DisableIntT1;
  SetPriorityIntTx( );
  // 設定Timerx中斷優先權。
  Ex:SetPriorityIntT1(7);



MICROCHIP
Regional Training Centers

Lab3 - Timer Polling

- 在Lab2的程式基礎上,嘗試改用Timer1來取代原先的軟體Delay。控制RD0可以可以不斷的轉態(Toggle, 1->0->1->…) 。 每次Toggle的時間間隔設一樣設為500 mS 。
- 記住前面所提過的,要使用使用Timer時必須先含入(include) Timer的標頭檔。
  Ex:#include <timer.h>
- 以軟體模擬(MPLAB SIM)為工具,系統時脈(Fosc)設為8MHz。
- 先閱讀Timer Function的說明文件,了解Timer Function的始用方法。至少必須了解OpenTimerX( )的用法。

MICROCHIP
Regional Training Centers

Lab3 - Timer Polling Step1

- 程式要如何改才能用Timer取代軟體Delay的功能？

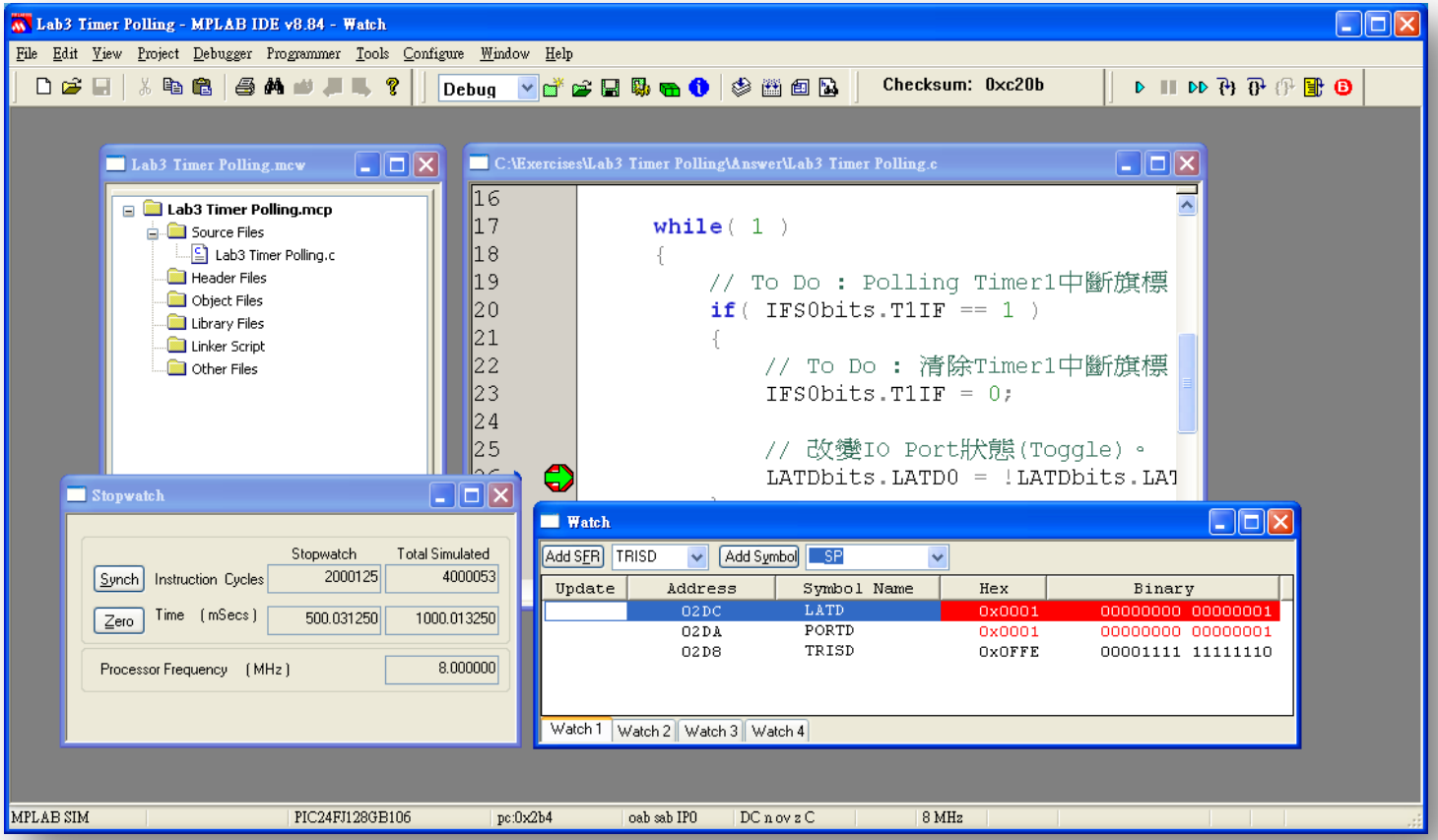
  刪除掉原先的Delay副程式，並將Timer1透過OpenTimer1 Function設定完成。
  在主程式中,Polling T1IF來得知Timer是否計數到預期值。T1IF必須手動清除,因此Polling=1後,必須利用程式將其清除為”0”。

- Timer要如何設定？

  設定時脈來源為內部(Tcy = (1/8MHz) * 2)。然後試著計算PR1及預除器該填入多少,才能達到500mS。

Lab3 - Timer Polling Step2

- 跟Lab2一樣,使用Watch Window跟Stopwatch來觀察程式的變化以及Timer的計數狀態。



MICROCHIP
Regional Training Centers

PRx 計算?

- PRx有沒有更容易了解計算的方式?
  自己算太累了,請Compiler幫我們算。

```
#define SystemFrequency 8000000L / 2
#define Timer1Tick ( ( SystemFrequency / 256 ) / Timer1TogglesPerSec )
#define Timer1TogglesPerSec 10
OpenTimer1( … & T1_PS_1_256 & … , Timer1Tick );
```

SystemFrequency:系統頻率(Fcy), Fcy = Fosc / 2 (dsPIC33/PIC24F/24H/24E)。
SystemFrequency / 256(預除器的設定):取得進入Timer1頻率。
換個方式想, 就是計算出一秒鐘進入Timer1的Clock個數。
(SystemFrequency / 256) / Timer1TogglesPerSec:一秒鐘有這麼多個Clock,那0.5秒就是/2, 0.1秒就是/10。