



# **MICROCHIP**

---

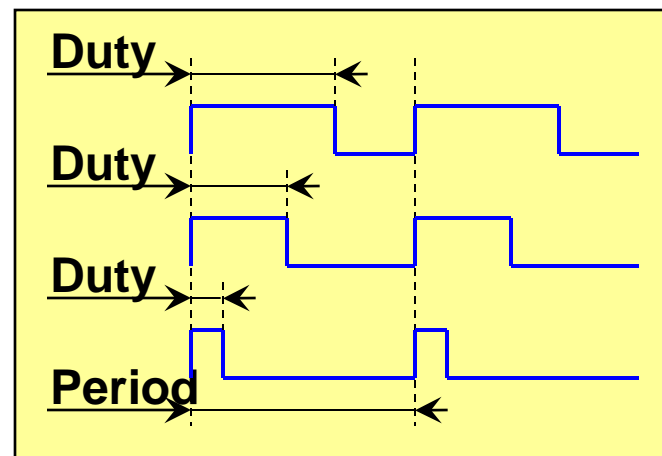
***Regional Training Centers***

## **Section 12**

### **Output Compare PWM**

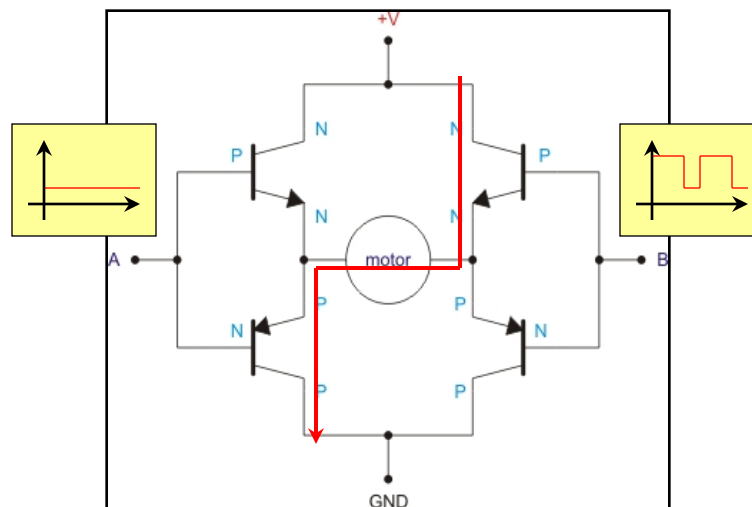
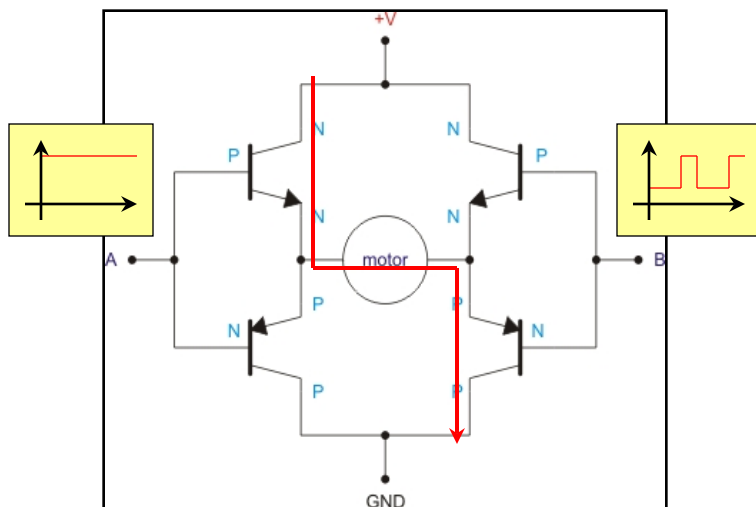
# What's PWM

- 脈波寬度調變(PWM, Pulse-width modulation)。一種調變工作週期的技術。透過高解析度的計數器,將方波的工作週期進行調變,用來對一個類比信號的電壓進行編碼。
- PWM是以數位控制來控制類比訊號一種非常有效的技術。廣泛的被應用在測量、通信及功率控制與變換等領域中。
- PWM控制兩個重要參數  
週期(Period)、比例(Duty)。



# Motor Control

- 以馬達控制為例,如圖所示的H Bridge結構馬達控制電路。可使用PWM控制正反轉與旋轉的速度。



# PIC32MX450 Output Compare

- PIC32MX450的Output Compare Module可以設定成以下幾種輸出模式(OCM<2:0>):

## Simple Compare Match Mode:

1. Normal Low, High on Match.
2. Normal High, Low on Match.
3. Toggle on Match.

## Dual Compare Mode:

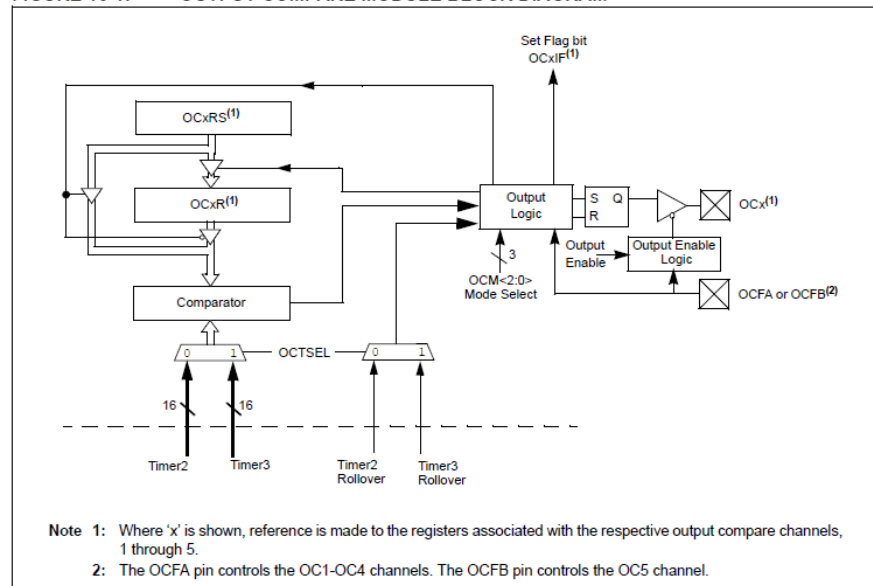
1. Single Pulse.
2. Continuous pulse.

## Simple PWM Mode:

1. PWM Mode, Fault Disable.
2. PWM Mode, Fault Enable.

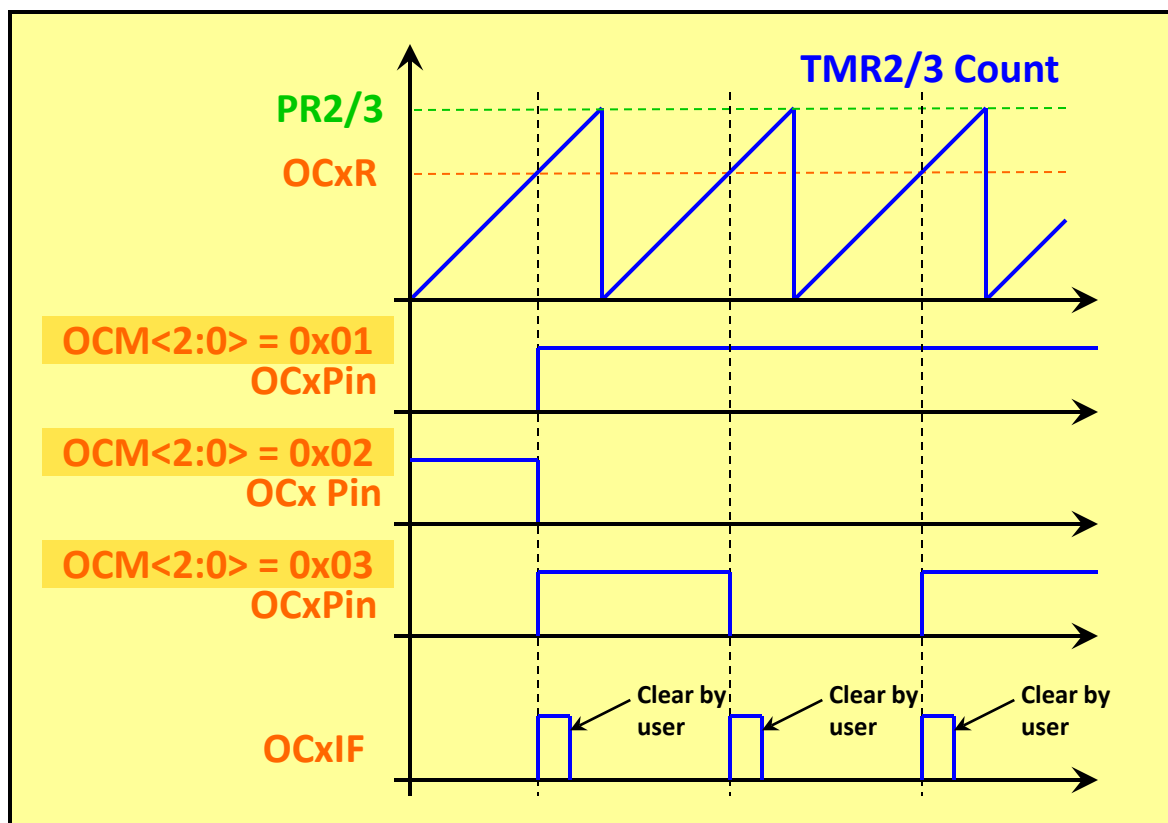
- 可使用Timer 2或Timer 3作為Time Base。

FIGURE 16-1: OUTPUT COMPARE MODULE BLOCK DIAGRAM



# Simple Compare Match Mode

- 在此模式下，TMR2/3遞增，在與OCxR時相同會改變OCx pin的狀態：
- OCM<2:0> = 0x01, Initialize OCx=0, “Set” on match。
- OCM<2:0> = 0x02, Initialize OCx=1, “Clear” on match。
- OCM<2:0> = 0x03, Toggle on match。

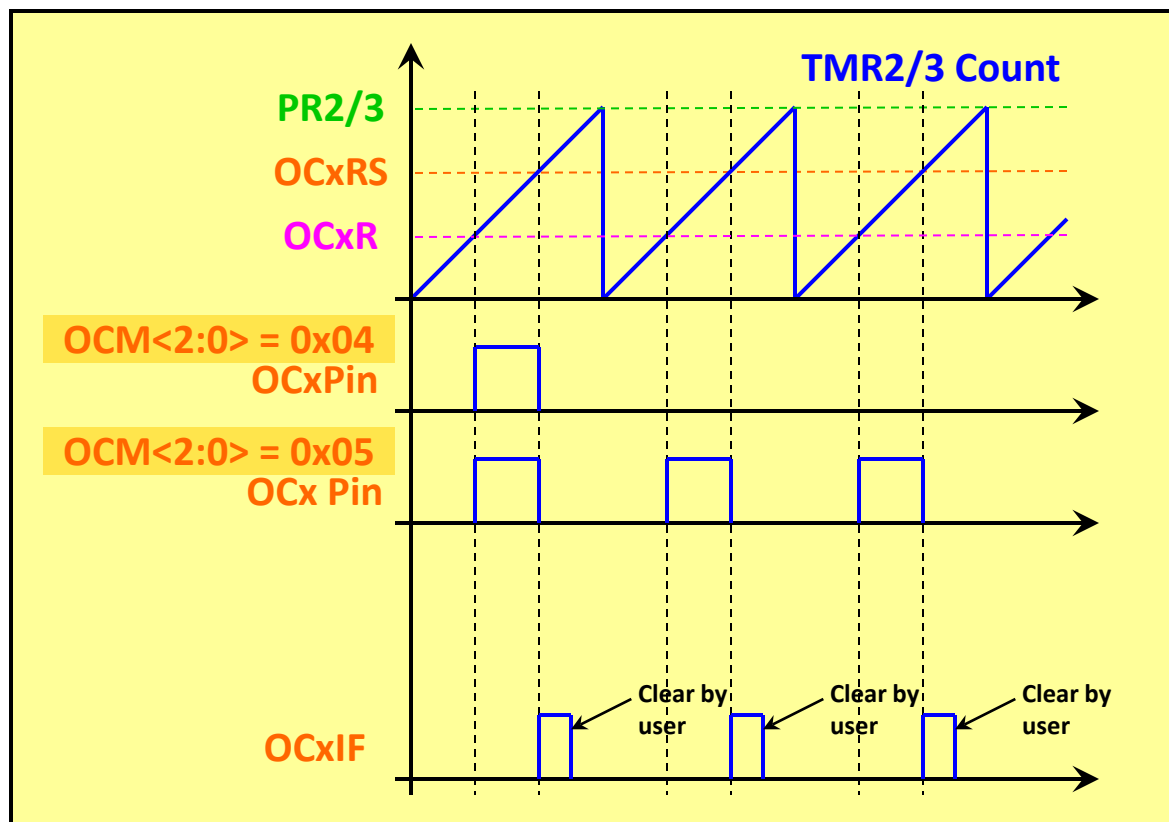


# Dual Compare Mode

- 在此模式下，TMR2/3遞增，在過程中會有兩次的比較動作。需要同時使用到 OCxR以及 OCxRS 暫存器。

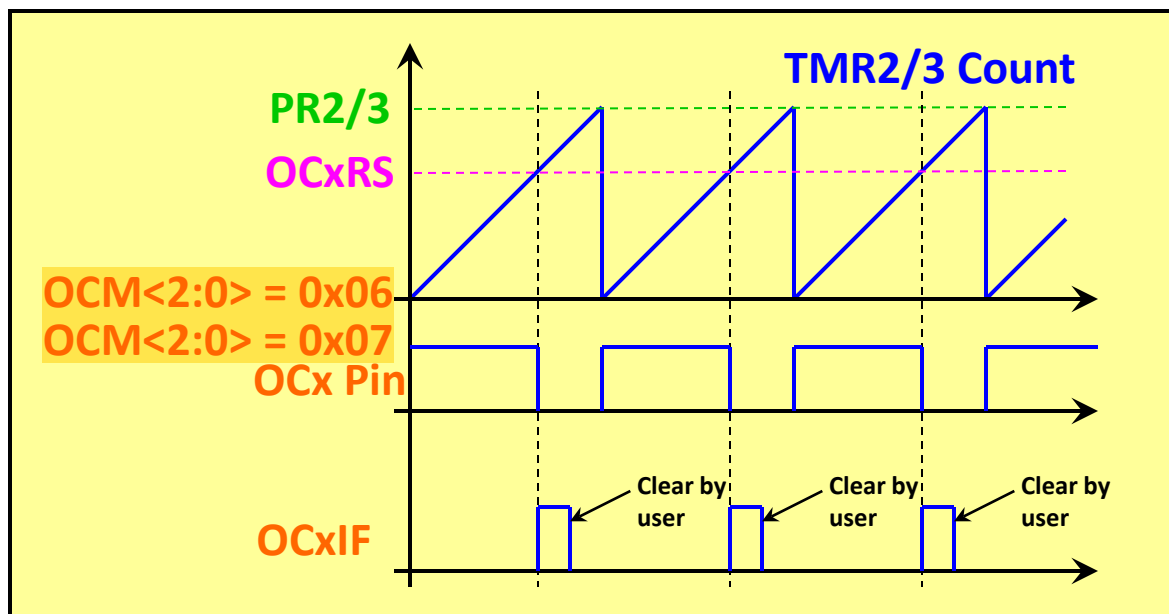
當OCxR與TMR2/3  
相等時OCx Pin  
變High,  
當OCxRS與TMR2/3  
相等時OCx Pin  
變Low。

- OCM<2:0> = 0x04,  
Signal Output Pulse
- OCM<2:0> = 0x05,  
Continuous Output Pulse。



# PWM Mode

- 在此模式下，TMR2/3遞增，在與OCxR相同會改變OCx pin的狀態：
- OCM<2:0> = 0x06 PWM, Fault Disable  
OCM<2:0> = 0x07 PWM, Fault Enable



# XC32 OC Function & Macro

- MPLAB XC32提供許多Timer Function可供使用。  
OpenOCx( );                   // 啟用OCx, 設定OCx工作模式。  
CloseOCx( );                 // 關閉OCx  
ConfigIntOCx( );            // 致能OCx中斷, 並設定中斷優先權。  
SetDCOCxPWM( );            // 設定OCxRS。  
ReadDCOCxPWM( );            // 讀取OCxRS。  
SetPulseOCx( );             // 設定OCxR, OCxRS。  
ReadRegOCx( );             // 讀取OCxR, OCxRS。  
IntClearFlag( INT\_OCx );    // 清除OCx中斷旗標。  
IntGetFlag( INT\_OCx );      // 取得OCx中斷旗標。  
...



# OC3 Initial Example

- **OC3 Initial Example**

```
OpenOC3( OC_ON | OC_IDLE_CON | OC_TIMER_MODE16 | OC_TIMER2_SRC |  
         OC_PWM_FAULT_PIN_DISABLE , 1000 , 1000 );
```

```
OpenTimer2( T2_ON | T2_IDLE_CON | T2_GATE_OFF | T2_PS_1_1 |  
            T2_SOURCE_INT, ( PBFrequency / 1 ) / 1000 );
```

```
ANSELBbits.ANSB10 = 0;
```

```
PPSUnlock;
```

```
PPSOutput(1, RPB10, OC3); //Assign OC3
```

```
PPSLock;
```

設定OC3為PWM模式, 時脈來源Timer2,  $OC3R/OC3RS = 1000$ 。

Timer2週期為1KHz, 所以PWM頻率 = 1K。

工作週期(Duty Cycle)為 $OC3RS/PR2 = 1000 / 10000 = 10\%$ 。

OC3的訊號由PPS指定到RPB10輸出。

# PWM 週期如何計算?

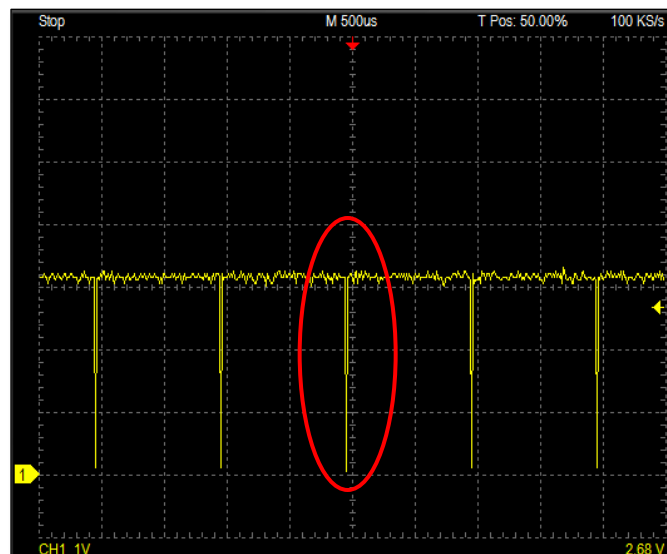
- PWM週期如何計算?

```
#define PBFrequency      10000000L
#define TimernTick      ( PBFrequency / i / TnTogglesPerSec )
#define TnTogglesPerSec 1000
OpenOCx( ... | OC_TIMERn_SRC | ... , 1000 , 1000 );
OpenTimern( ... | Tn_PS_1_1 | Tn_SOURCE_INT, TimernTick );
```

- OC PWMx的週期來自TMRn的週期, 所以可以根據Timer的設計來計算OCPWM的周期。  
OCPWMx週期=  $PBFrequency / n / TnTogglesPerSec$ 。

# PWM 工作週期控制

- 為何要100% Duty要單獨處理？
- PWM Mode下, 當
  1.  $TMRx = OCxRS$ 時輸出Low, 2.  $TMRx = PRx$ 時輸出High。
- 在這條件下, 如果因為想取得100% Duty輸出, 必須設定  $OCxRS = PRx$ 。但  $OCxRS = PRx$ 時, 上述兩個條件卻同時成立了。會造成下陷的突波(Glitch)出現。
- 為了避免此現象發生。必須避免  $OCxRS = PRx$ 。所以要輸出100% Duty時, 將  $OCxRS$ 設定的比  $PRx$ 大。 $TMRx = OCxRS$ 的條件就沒有機會成立。(  $TMRx = PRx$ 時,  $TMRx$ 會歸零, Timer的架構設定)。



# OC3 PWM DC Set Example

- **OC3 PWM Duty Cycle Set Example**

if( Duty  $\geq$  PR2 )

    SetDCOC3PWM( PR2 + 1);

else

    SetDCOC3PWM( Duty );

設定OC3的Duty Cycle。當Duty  $\geq$  PR2時(100% Duty Cycle),  
將OCxRS填入PR2+1。其他情形(0~99.999%Duty Cy)則填入  
Duty值。

# Lab12 Output Compare PWM ADC Adj.

- 利用Lab11的程式,將PWM的功能加入。初始化OC3, 設定為PWM Fault Disable Mode。並使用ADC的數值來改變PWM的Duty。PWM時脈來源設定為Timer2, 頻率為1KHz。
- 記住!OC3也必須透過PPS指定腳位, 回憶前面章節的說明來設定OC3的傳送接腳(OC3)為RPB10(D3)。
- 閱讀OC Function的說明文件,了解如何使用OpenOC3( ), SetDCOC3PWM( )等函數。
- 設定PC超級終端機軟體, 連接RS232 Cable。透過Bootloader將程式燒錄到MCU中, 調整VR並透過LED觀察亮度變化。也可透過超級終端機觀察數值變化。

# Lab12 Output Compare PWM ADC Adj.

## Step

- 可透過LED觀察PWM的變化。也可以透過Lab11的架構,在超級終端機會看到Duty跟Period的變化。

