MICROCHIP

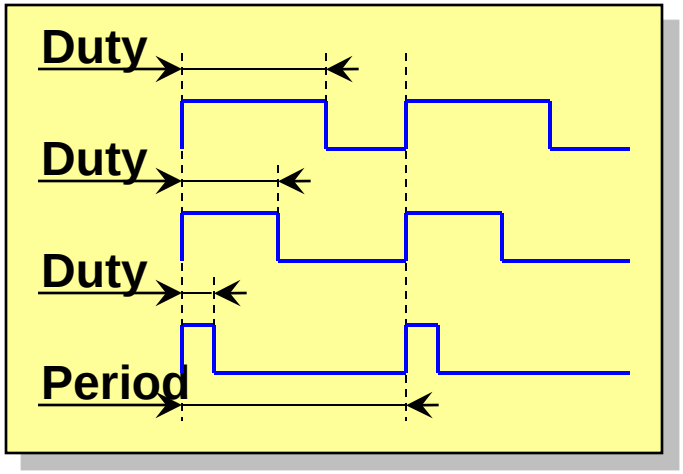
Regional Training Centers

Section 11
Output Compare PWM

MICROCHIP
Regional Training Centers

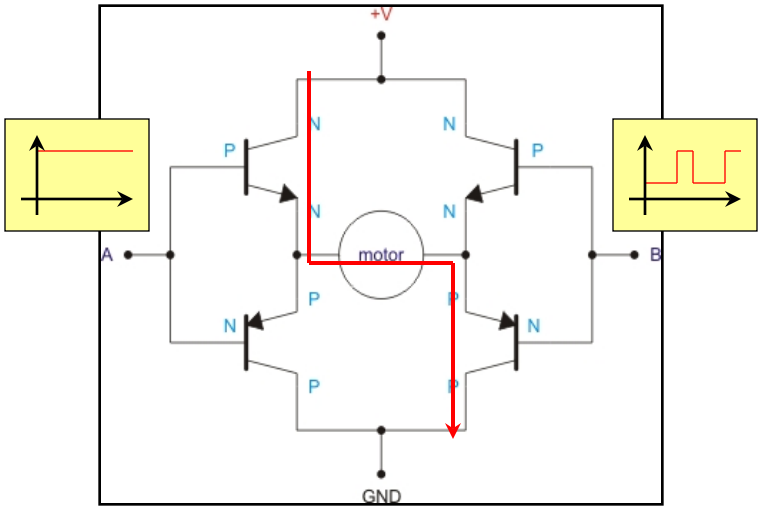
What's PWM

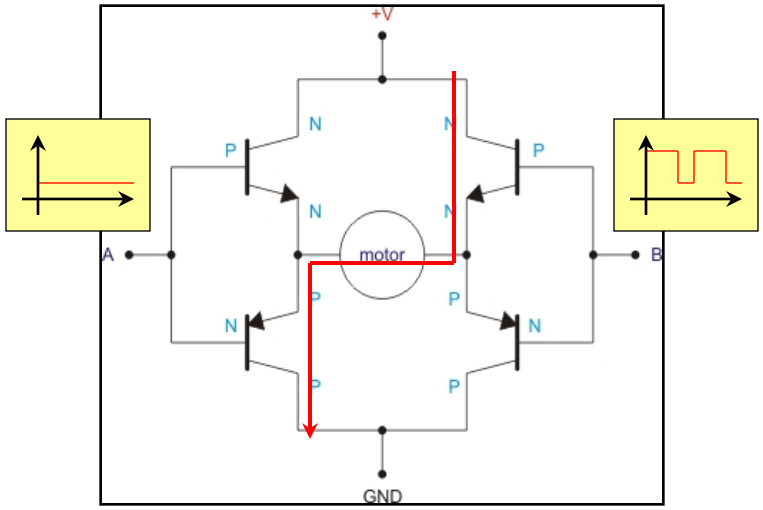
- 脈波寬度調變(PWM, Pulse-width modulation)。一種調變工作週期的技術。透過高解析度的計數器,將方波的工作週期進行調變,用來對一個類比信號的電壓進行編碼。
- PWM是以數位控制來控制類比訊號一種非常有效的技術。廣泛的被應用在測量、通信及功率控制與變換等領域中。
- PWM控制兩個重要參數
  週期(Period) 、比例(Duty) 。



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。





PIC32MX470 Output Compare

- PIC32MX470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。

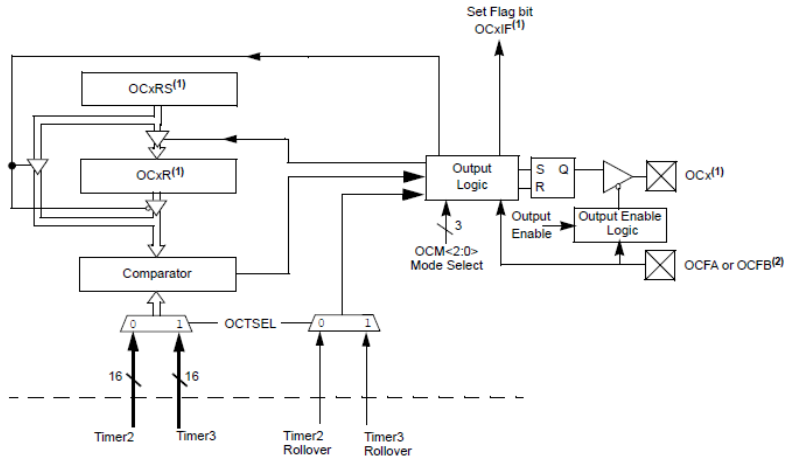
  2.Continuous pulse。

  Simple PWM Mode:

  1.PWM Mode, Fault Disable。

  2.PWM Mode, Fault Enable。

FIGURE 16-1: OUTPUT COMPARE MODULE BLOCK DIAGRAM

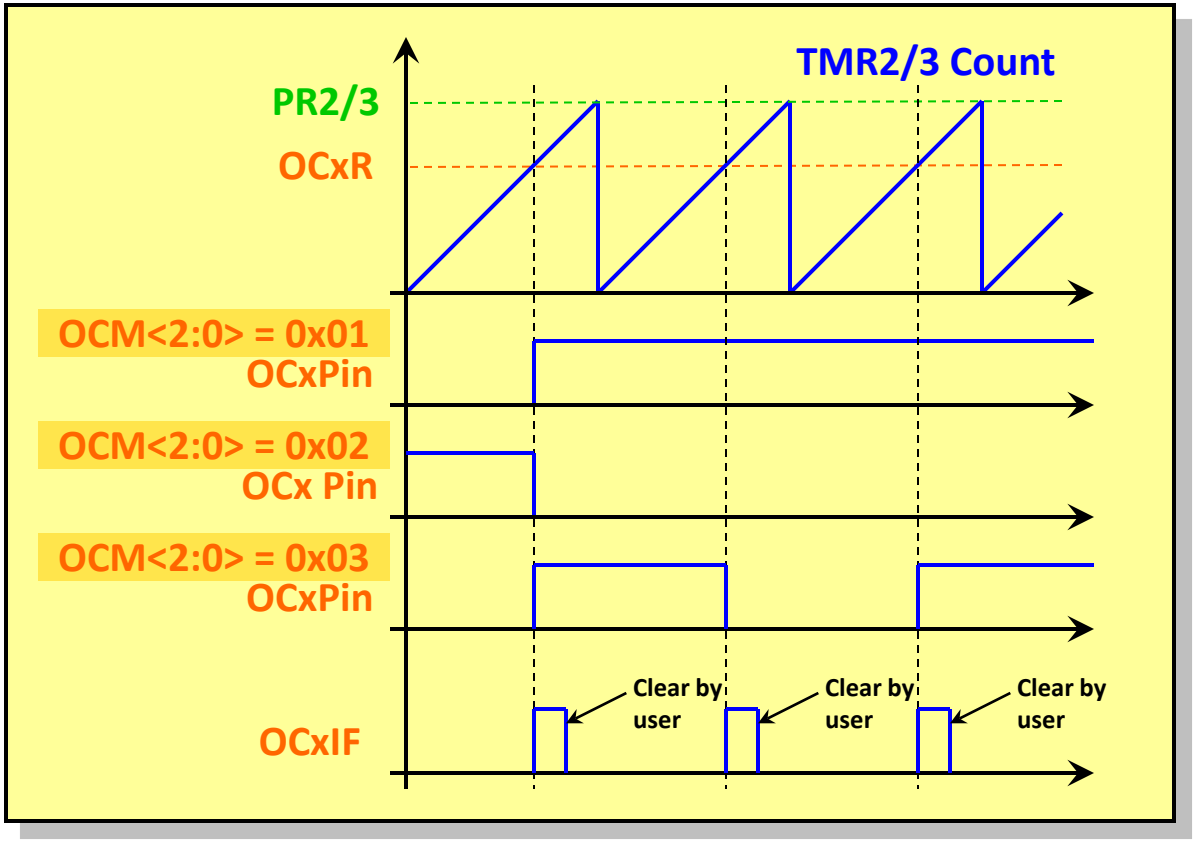


Note 1: Where 'x' is shown, reference is made to the registers associated with the respective output compare channels, 1 through 5.
2: The OCFA pin controls the OC1-OC4 channels. The OCFB pin controls the OC5 channel.

- 可使用Timer 2或Timer 3作為Time Base。

Simple Compare Match Mode

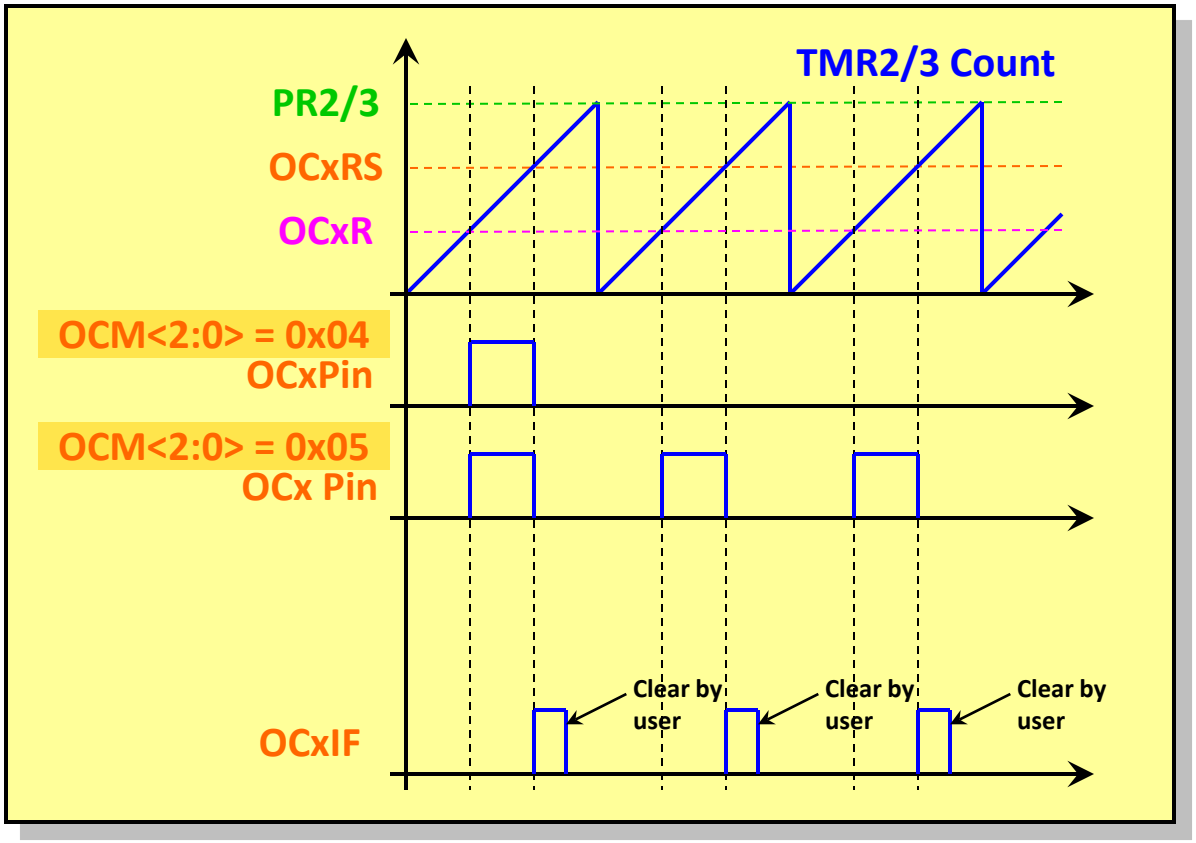
- 在此模式下，TMR2/3遞增, 在與OCxR時相同會改變OCx pin的狀態:
- OCM<2:0> = 0x01, Initialize OCx=0, “Set” on match 。
- OCM<2:0> = 0x02, Initialize OCx=1, “Clear” on match 。
- OCM<2:0> = 0x03, Toggle on match。



MICROCHIP
Regional Training Centers

Dual Compare Mode

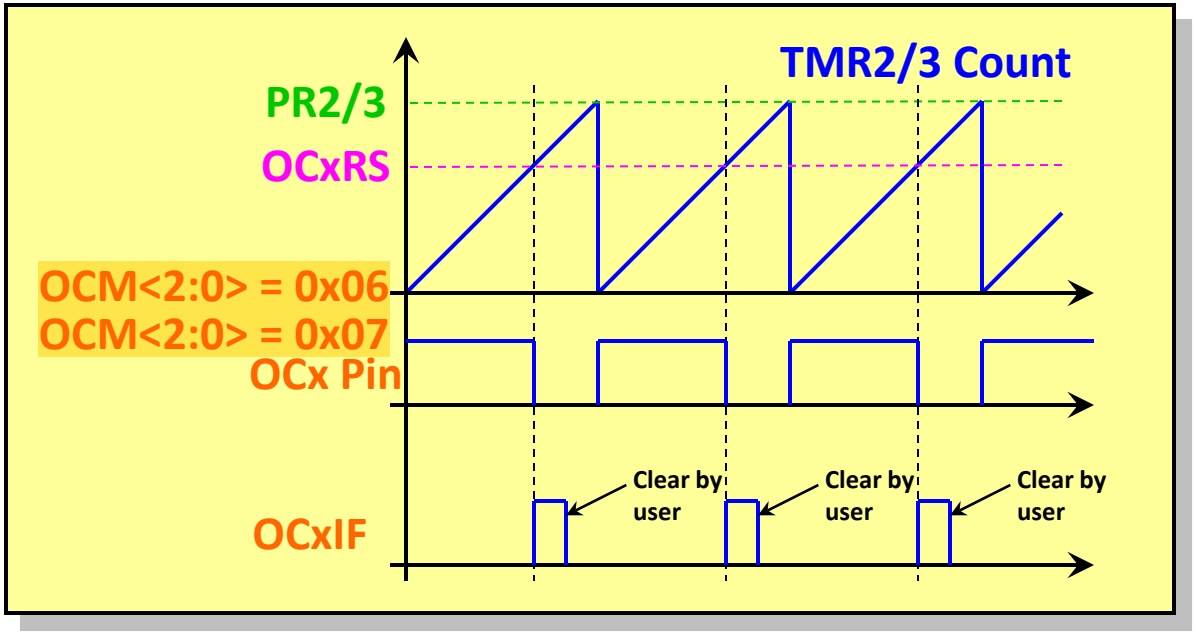
- 在此模式下，TMR2/3遞增, 在過程中會有兩次的比較動作。需要同時使用到 OCxR以及 OCxRS 暫存器。
  當OCxR與TMR2/3相等時OCx Pin變High,
  當OCxRS與TMR2/3相等時OCx Pin變Low。
- OCM<2:0> = 0x04,
  Signal Output Pulse
- OCM<2:0> = 0x05,
  Continuous Output Pulse。



MICROCHIP
Regional Training Centers

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。
...
```

MICROCHIP
Regional Training Centers

OC1 Initial Example

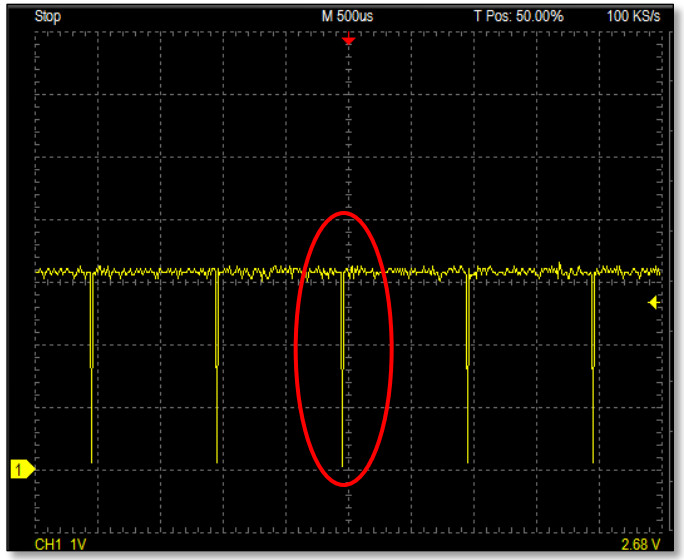
- **OC1 Initial Example**

```
OpenOC1( OC_ON | OC_TIMER_MODE16 | OC_TIMER2_SRC |
            OC_PWM_FAULT_PIN_DISABLE ,  1000 , 1000 );

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

PPSUnLock;
PPSOutput(4, RPD8, OC1); //Assign OC1
PPSLock;
```

設定OC1為PWM模式, 時脈來源Timer2, OC1R & OC1RS = 1000。Timer2週期為1KHz, 所以PWM頻率 = 1KHz。
工作週期(Duty Cycle)為OC1RS / PR2 = 1000 / 10000 = 10%。
OC1的訊號由PPS指定到RPD8輸出。

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)。



MICROCHIP
Regional Training Centers

OC1 PWM DC Set Example

- **OC1 PWM Duty Cycle Set Example**

```
if( Duty >= PR2 )
        SetDCOC1PWM( PR2 + 1);
else
        SetDCOC1PWM( Duty );
```

設定OC1的Duty Cycle。當Duty >= PR2時(100% Duty Cycle), 將OCxRS填入PR2+1。其他他情形(0~99.999%Duty Cy)則填入Duty值。

MICROCHIP
Regional Training Centers

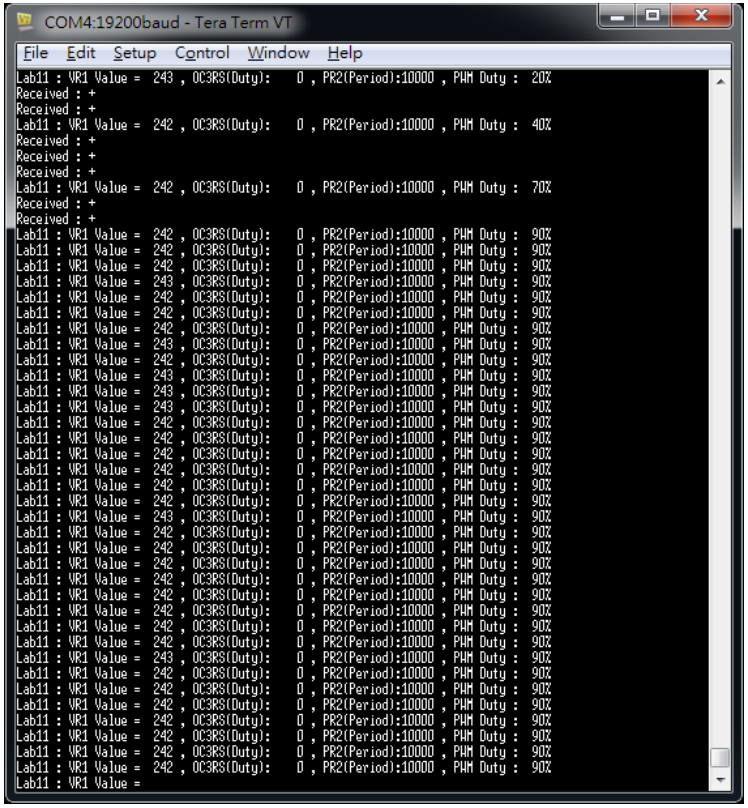
Lab11 Output Compare PWM User Adj.

- 利用Lab10的程式,將PWM的功能加入。初始化OC3, 設定為PWM Fault Disable Mode。並透過UART接收的命令來改變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RPD10(D5)。
- 閱讀OC Function的說明文件,了解如何使用OpenOC*n*( ), SetDCOC*n*PWM( )等函數。
- 使用Bootloader將程式燒錄進APP028-1。觀察程式執行的情況。驗證看看程式是否正常動作。

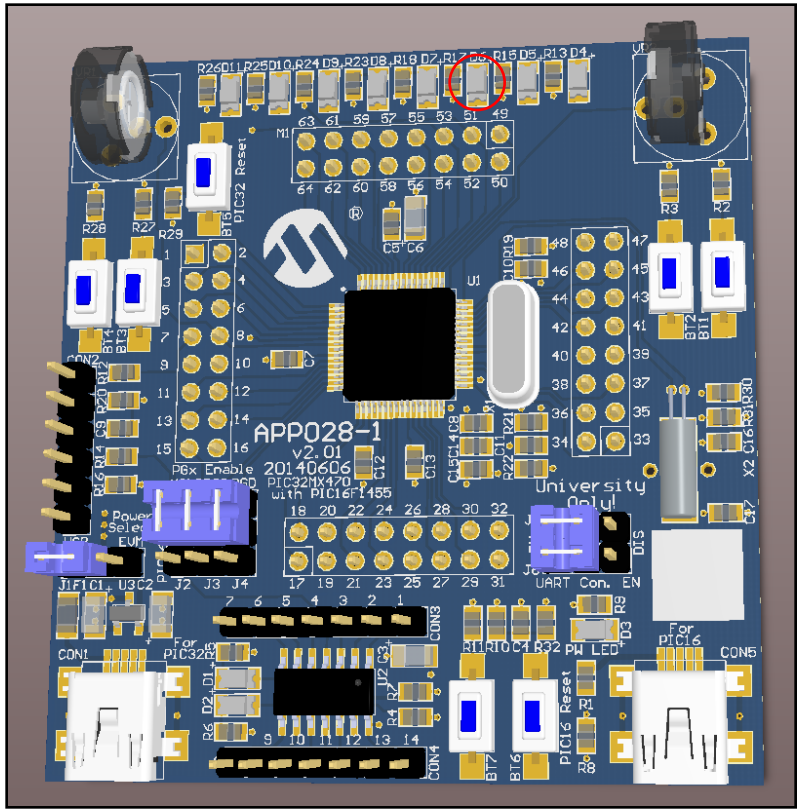
Lab11 Output Compare PWM User Adj.

Result!

- 可透過LED觀察PWM的變化。嘗試使用’+’ , ‘-’命令, 控制PWM的Duty, 也可透過超級終端機觀察Duty跟Period的變化。



```
Lab11 : VR1 Value =  243 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  20%
Received : +
Received : +
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  40%
Received : +
Received : +
Received : +
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  70%
Received : +
Received : +
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  243 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  243 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  243 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  243 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  243 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  243 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  243 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =  242 , OC3RS(Duty):    0 , PR2(Period):10000 , PWM Duty :  90%
Lab11 : VR1 Value =
```



MICROCHIP
Regional Training Centers

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。

MICROCHIP
Regional Training Centers

Lab12 Output Compare PWM ADC Adj.

- 嘗試利用Lab11的程式,將Duty調整的功能, 改由ADC的數值來改變PWM的Duty。
- 使用Bootloader將程式燒錄進APP028-1。觀察程式執行的情況。驗證看看程式是否正常動作。