MICROCHIP

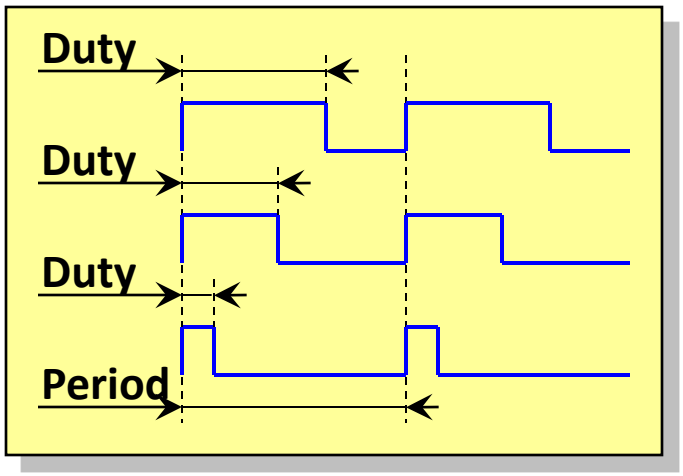
Regional Training Centers

Section 11
Output Compare PWM

MICROCHIP
Regional Training Centers

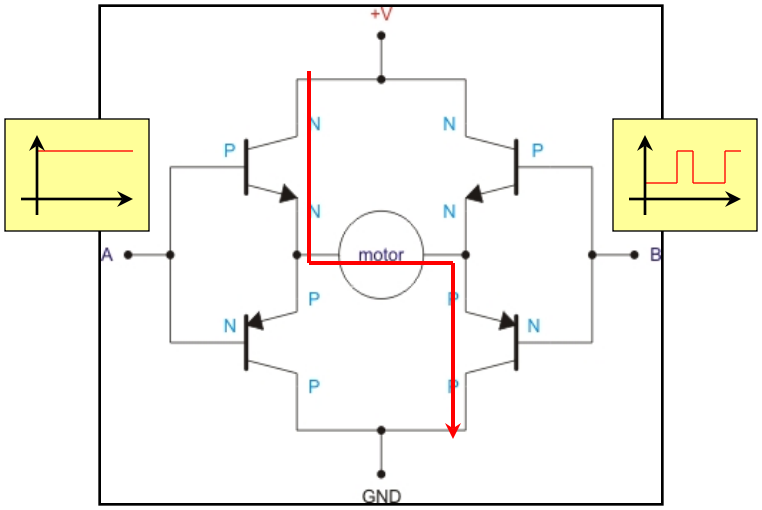
What's PWM

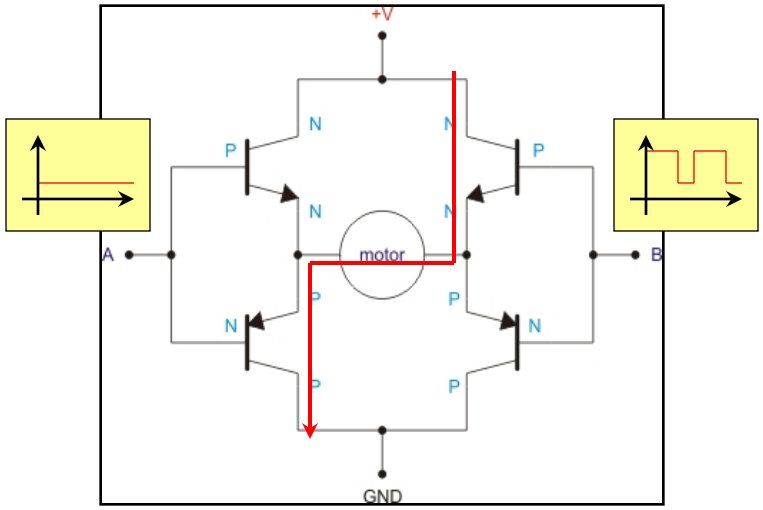
- 脈波寬度調變(PWM, Pulse-width modulation)。一種調變工作週期的技術。透過高解析度的計數器, 將脈波的工作週期進行調變,用來對一個類比信號的電壓進行編碼。
- PWM是以數位控制來控制類比訊號一種非常有效的技術。廣泛的被應用在測量、通信及功率控制與變換等領域中。
- PWM控制兩個重要參數
  週期(Period) 、比例(Duty) 。



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。





PIC24 Output Compare

- PIC24GB Serial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:

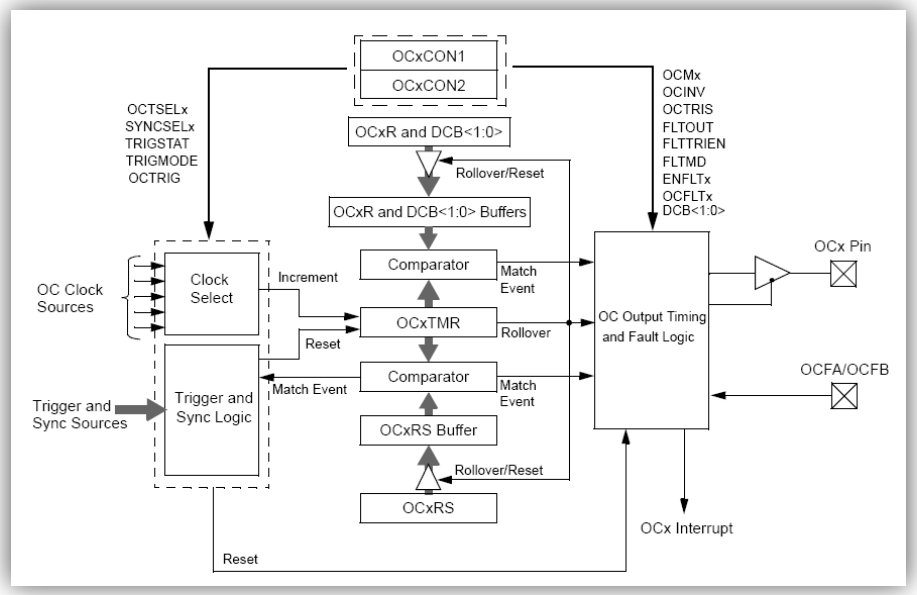
  1.Single Pulse

  2.Continuous pulse

  Simple PWM Mode:

  1.Edge-Aligned PWM Mode

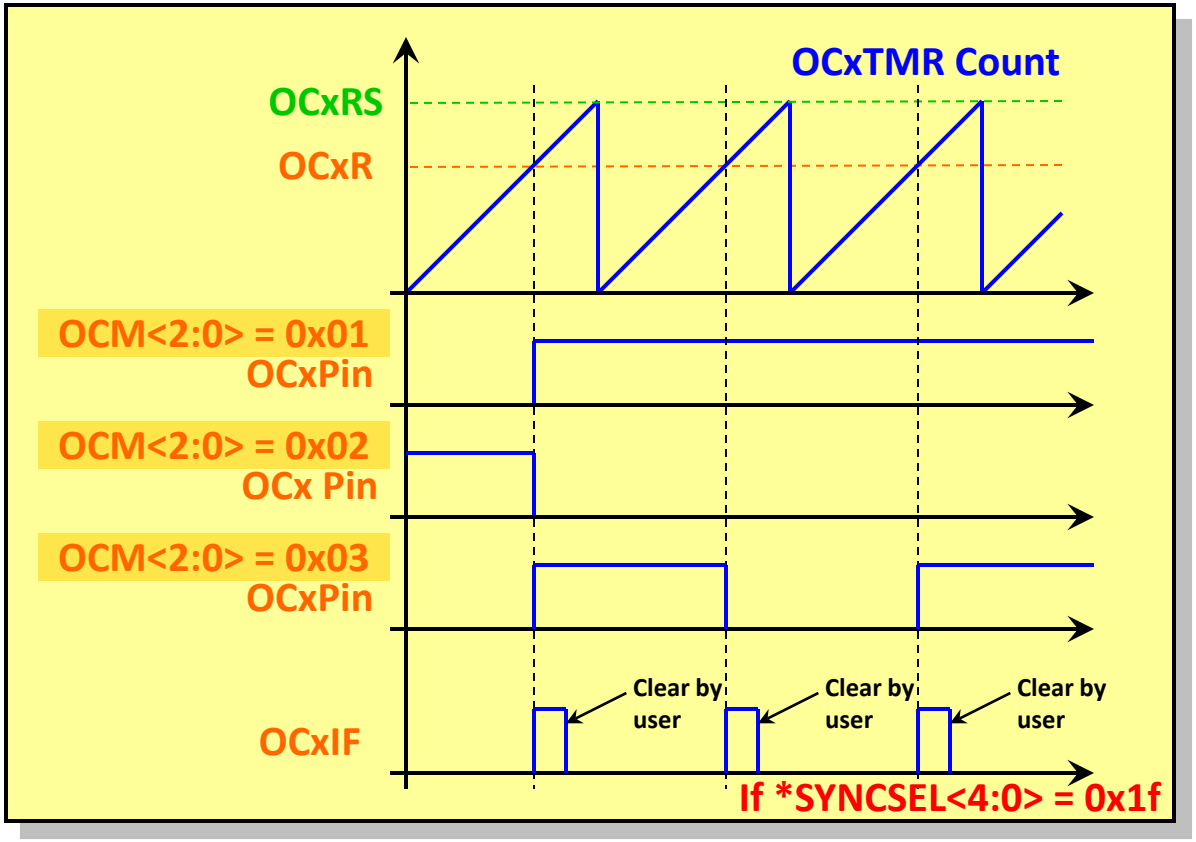
  2.Center-Aligned PWM Mode



- 可使用Timer 1~5或T_{CY}作時脈來源。

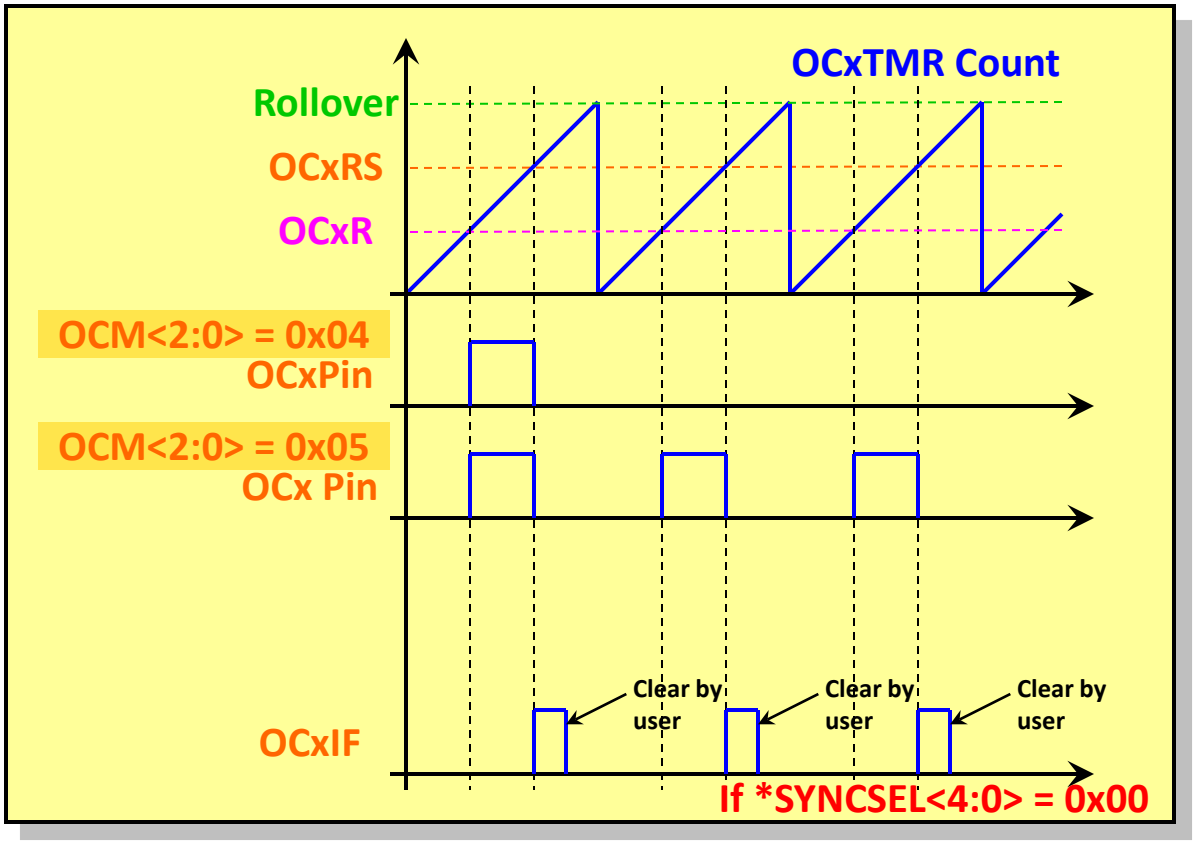
Simple Compare Match Mode

- 在此模式下, OCxTMR會遞增, 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
- OCxRS 必須大於 OCxR。



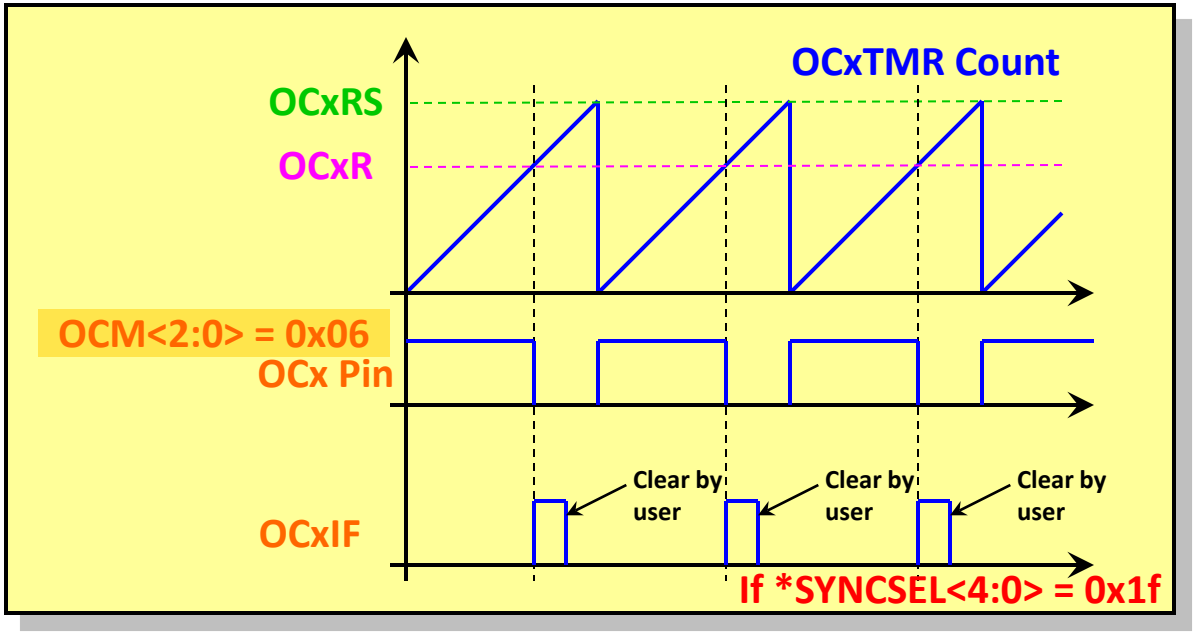
Dual Compare Mode

- 在此模式下, OCxTMR會遞增, 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OCxTMR相等時OCx Pin變High,
  當OCxR與OCxTMR相等時OCx Pin變Low。
- OCM<2:0> = 0x04,
  Signal Output Pulse
- OCM<2:0> = 0x05,
  Continuous Output Pulse



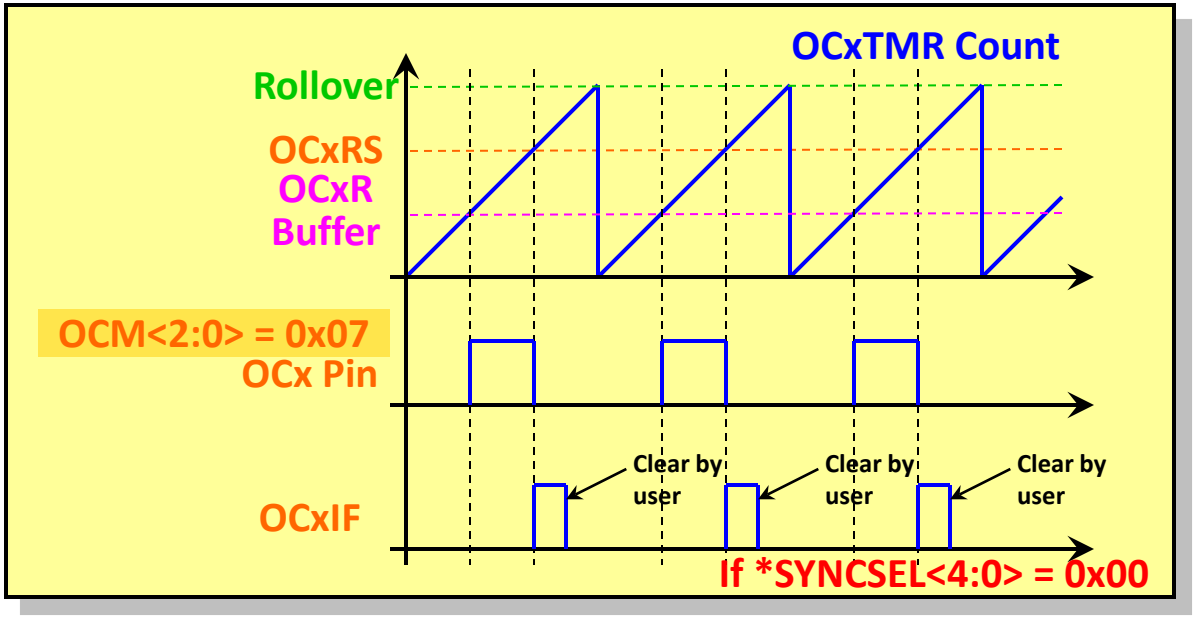
Edge-Aligned PWM Mode

- 在此模式下, OCxTMR會遞增, 在與OCxR相同會改變OCx pin的狀態:
- OCM<2:0> = 0x06, Edge-Aligned PWM



Center-Aligned PWM Mode

- 此模式與OCM<2:0> = 0x05 Continuous Output Pulse很像, 唯一的差別在於OCxR與OCxRS有Buffer, 意謂著可以隨時將新的寫入, 不怕影響目前訊號的運作, 新的OCxR與OCxRS會在OCxTMR Overflow時載入。
- OCM<2:0> = 0x07, Center-Aligned PWM



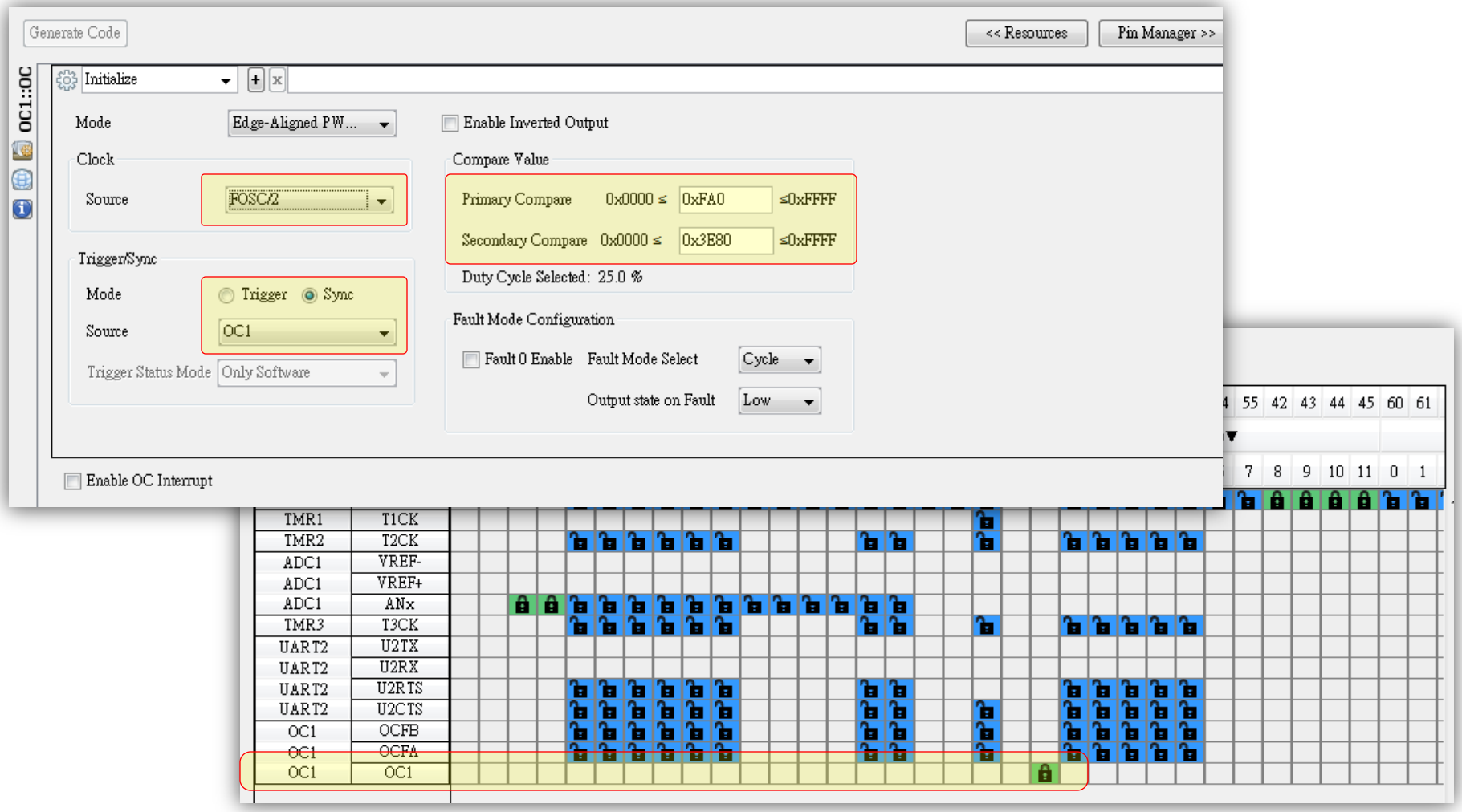
Lab12 Output Compare PWM

目 標

- 嘗試透過MCC的設定,在Lab11架構上加入**OC1**的設定。
  開啟OC1, **模式設定為**Centre Aligned PWM, Sync. itself(OC1), Clock Source F_{CY}。
- PWM頻率設定為1KHz, Duty 25%, 並將**OC1**的輸出指定至RD0(LED1)。使用杜邦線連接RD0與LED1後可以明確觀察到亮度差異。
- ***Primary Compare (OCxR, Duty)**
  ***Secondary Compare (OCxRS, Peroid)**

Lab12 Output Compare PWM

MCC's Setting



MICROCHIP
Regional Training Centers

MCC's O.C.PWM Function

- **MCC中的幾個OCx常用函數:**

void OC1_Initialize(void)

// 啟用OC1, 設定OC1工作模式。

void OC1_Start(void); void OC1_Stop(void);

// 開啟/關閉 OC1。

void OC1_SingleCompareValueSet(value)

// Single Compare設定 (OC1R <- value)

void OC1_DualCompareValueSet(priVal, secVal)

// Dual Compare設定 (OC1R <- priVal, OC1RS <- secVal)

void OC1_CentreAlignedPWMConfig(priVal, secVal)

// Centre Aligned PWM設定 (OC1R <- priVal, OC1RS <- secVal)

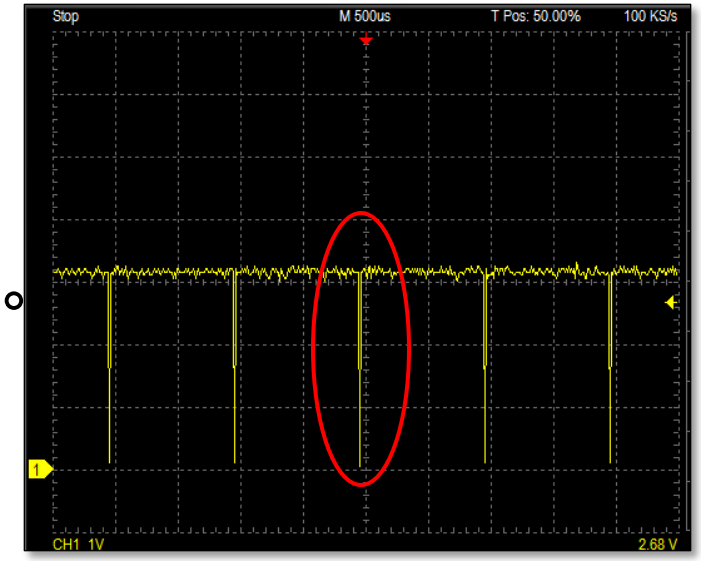
void OC1_EdgeAlignedPWMConfig(priVal, secVal)

// Edge Aligned PWM設定 (OC1R <- priVal, OC1RS <- secVal)

MICROCHIP
Regional Training Centers

100% Duty Issue

- 為何要100% Duty要單獨處理?
- PWM Mode下, Sync itself, 當
  1.OCxTMR = OCxR時輸出Low 2. OCxTMR = OCxRS時輸出High。
- 在這條件下, 如果因為想取得100% Duty輸出, 必須設定 OCxR=OCxRS 。但此時會導致上述兩個條件卻同時成立。因此有可能會造成下陷的突波(Glitch)出現。
- 為了避免此現象發生。當要輸出100% Duty時, 可以將OCxR設定的比OCxRS大。OCxTMR=OCxR 的條件就沒有機會成立。
  (OCxTMR = OCxRS時, OCxTMR會歸零)。



PWM Duty Cycle Setting Example

- **OC1 PWM Duty Cycle Set Example**

```
if( DutyValue >= OC1RS )
        OC1_EdgeAlignedPWMConfig(OC1RS + 1, OC1RS);
else
        OC1_EdgeAlignedPWMConfig(DutyValue, OC1RS);
```

設定OC1的Duty Cycle。
當DutyValue >= OC1RS時(100% Duty Cycle),
將OC1R填入OC1RS+1。
其他他情形(0~99.9%Duty Cycle)
則填入DutyValue值。

Lab13

Output Compare PWM ADC DutyAdj

目 標

- 嘗試使用Lab12的架構, 加入合適的程式片段。讓我們可以透過VR的值來改變PWM的Duty Cycle。
- VRValue(0 ~1023) <-> PWM Duty (0% ~ 100%)

MICROCHIP
Regional Training Centers

Lab13 O.C. PWM ADC DutyAdj Code Example

main.c

```
int main(void)
{
   while(1)
   {
      ...
      if(AD1Flag)
      {
         AD1Flag = 0;
         ...
         if (AD1Buffer[0] >= 1023)
            OC1_EdgeAlignedPWMConfig(OC1RS + 1, OC1RS);
         else
            OC1_EdgeAlignedPWMConfig(((long) AD1Buffer[0] * OC1RS) / 1023, OC1RS);
      }
   }
}
```

Lab14

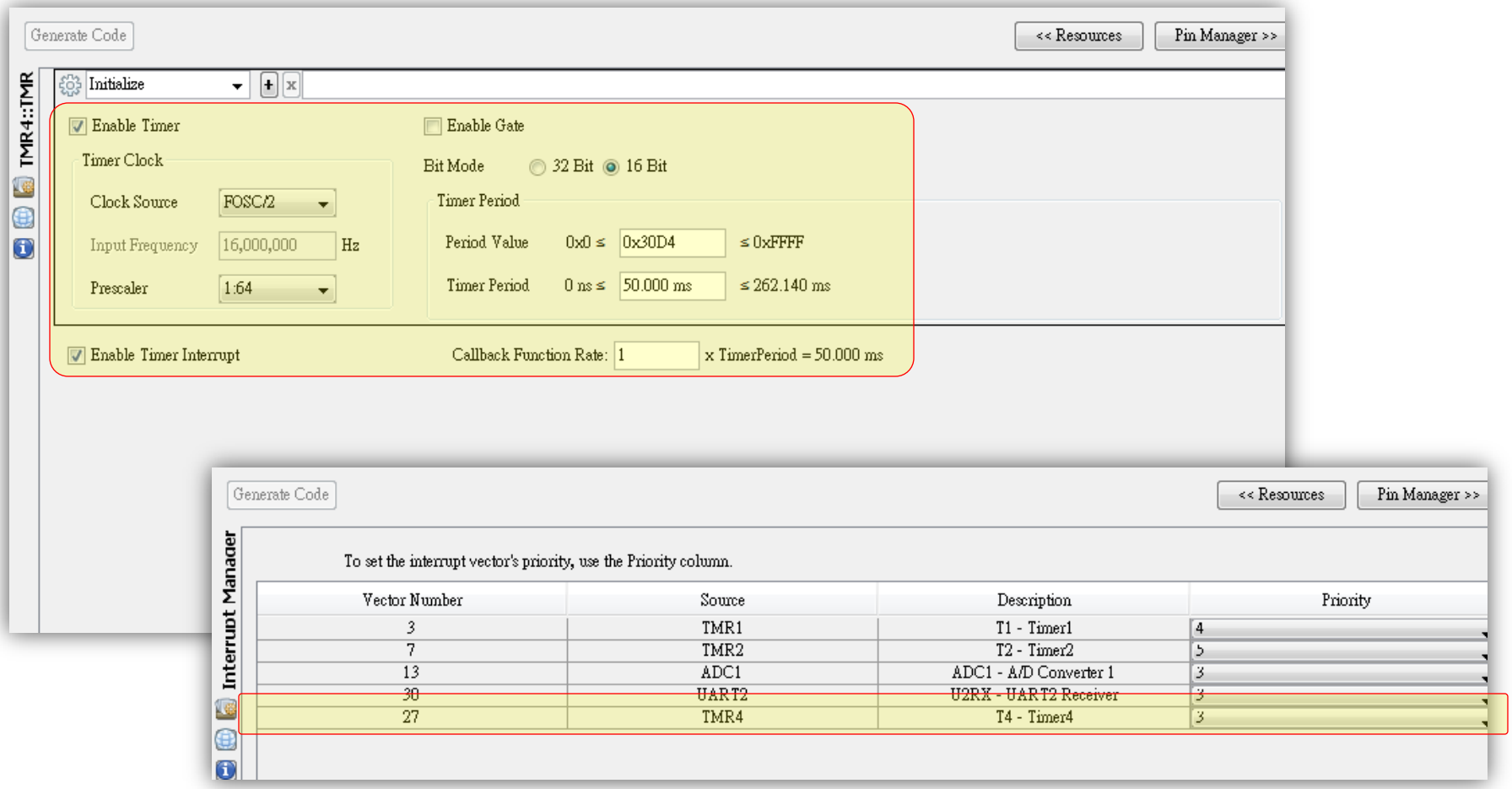
Output Compare PWM Auto DutyAdj

目 標

- 嘗試使用Lab12的架構, 加入合適的程式片段。讓我們可以透過Timer定時的, 自動的改變PWM的Duty Cycle。
- 新增Timer 4來定時增減Duty達成自動改變的機制, 請開啟Timer 4中斷, 每50mS調整一次Duty,中斷優先權設定為”3”。

Lab14 O.C. PWM Auto DutyAdj

MCC's Setting



MICROCHIP
Regional Training Centers

Lab14 O.C. PWM Auto DutyAdj Code Example

main.c

```
volatile unsigned char T4Flag = 0;
unsigned int Duty = 50;
char DutyDistance = 2;

 while (1)
{
   …
   if (T4Flag)
   {
      T4Flag = 0;
      if (Duty >= 100 || Duty <= 0)
         DutyDistance = -DutyDistance;

      if (Duty >= 100)
         OC1_EdgeAlignedPWMConfig(OC1RS + 1, OC1RS);
      else
         OC1_EdgeAlignedPWMConfig(((long) Duty * OC1RS) / 100, OC1RS);
   }
}
```

Tmr4.c

```
extern volatile unsigned char T4Flag;

void TMR4_CallBack(void)
{
   T4Flag = 1;
}
```