



MICROCHIP

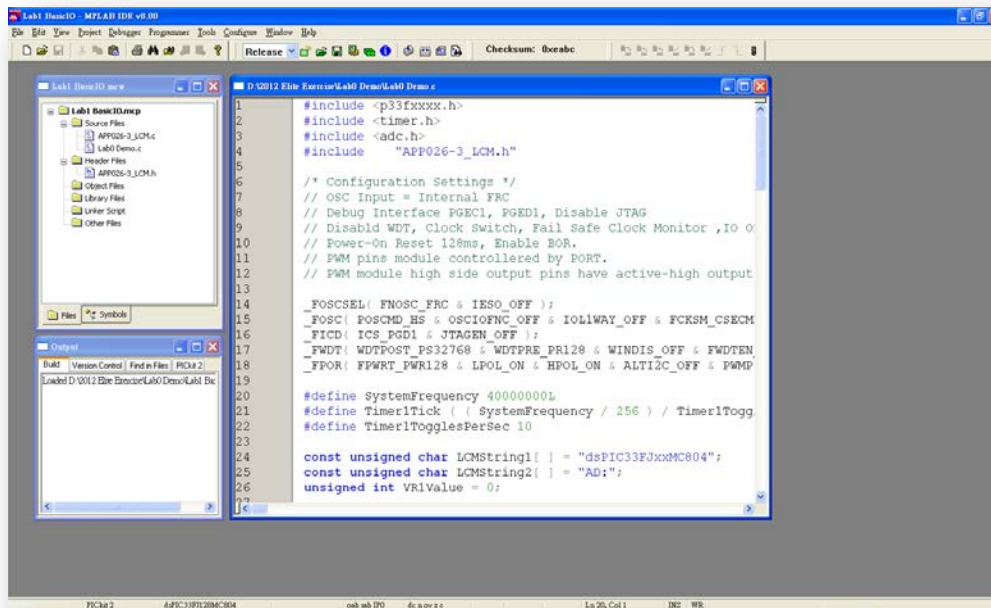
Regional Training Centers

Section 2

IDE, Compiler, MCC & Development Tools Introduction

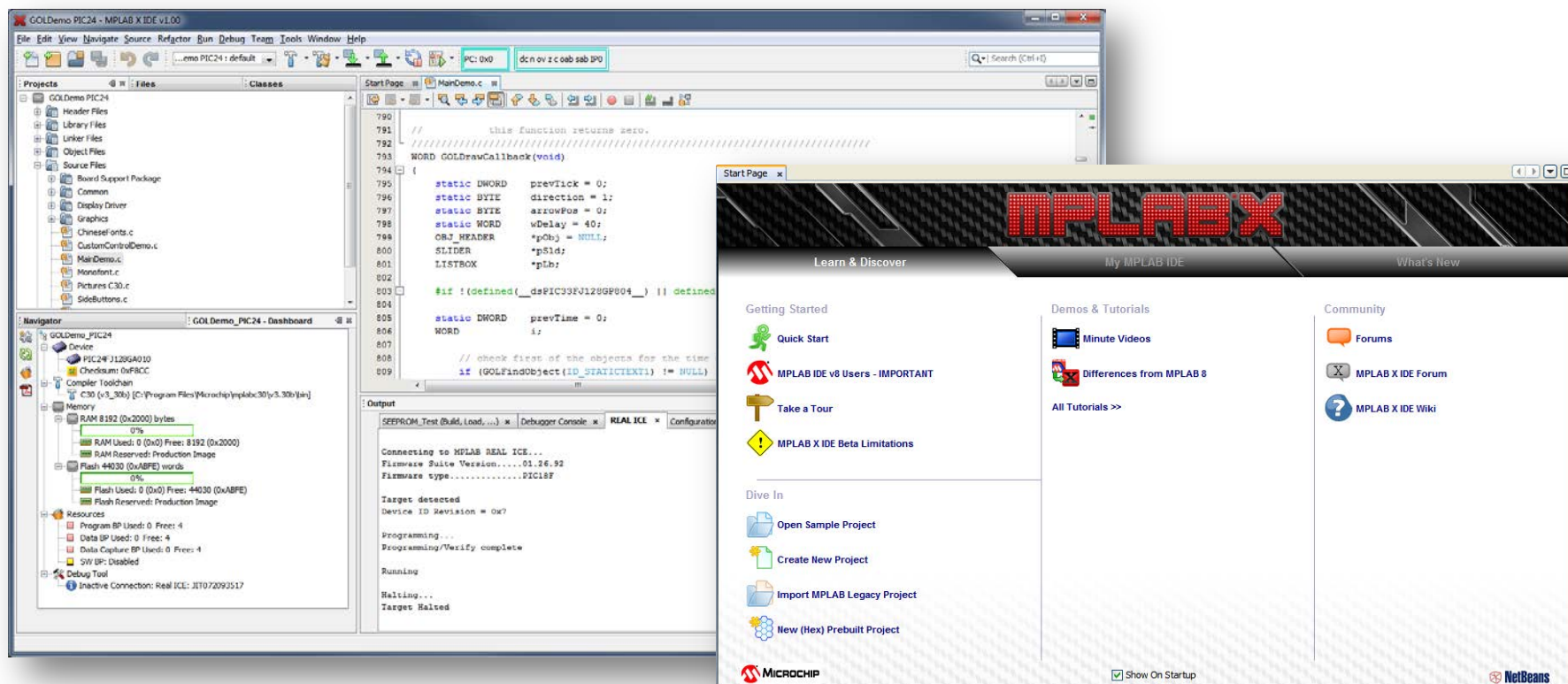
MPLAB® IDE

- Microchip提供的整合式開發環境, 支援全系列8-Bits, 16-Bits及32-Bits的MCU。所有的MCU都可透過相同的環境開發。最終版本是v8.92(不再更新, 以MPLAB X IDE取代)。
- 可整合各式的組譯器/編譯器(MPLAB C, Hi-TECH C, etc.), 開發工具(PICkit3, ICD3, Real ICE, etc..)。

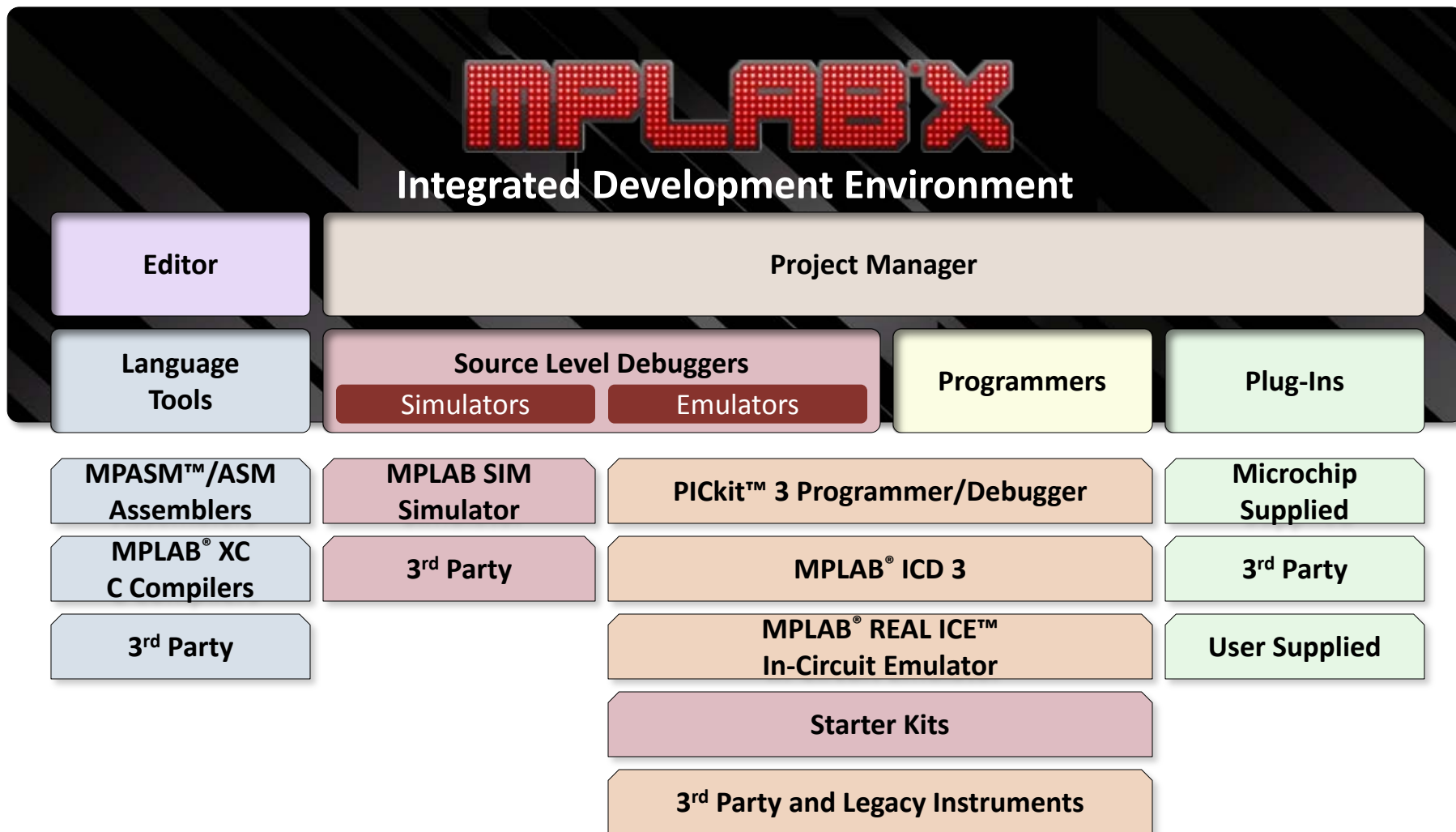


MPLAB® X IDE

- 新一代的整合式開發環境, 可支援全系列MCU, 擁有許多便利的功能與特徵。Java Base, 可以跨平台, 目前最新的版本是v3.20。
- 不支援ICD2, ICE2000/4000, ProMATE II, PICStart Plus.



MPLAB® X IDE Overview



MPLAB XC16

- 新一代的C Compiler, 支援Microchip 16-Bits MCU。架構於GCC Compiler, 採用GNU General Public License (GPL)。
 - * GPL:使用者可以自由執行/複製/修改/改進再公開
- 支援PIC24/dsPIC30/dsPIC33全系列MCU, MPLAB XC16可整合於MPLAB IDE / MPLAB X IDE中。
- 提供標準C Functions(sprintf, scanf, etc..), 週邊函式庫, 數學運算函數等。
- 可至Microchip網站, 取得免費版本。
<http://www.microchip.com/xc16>

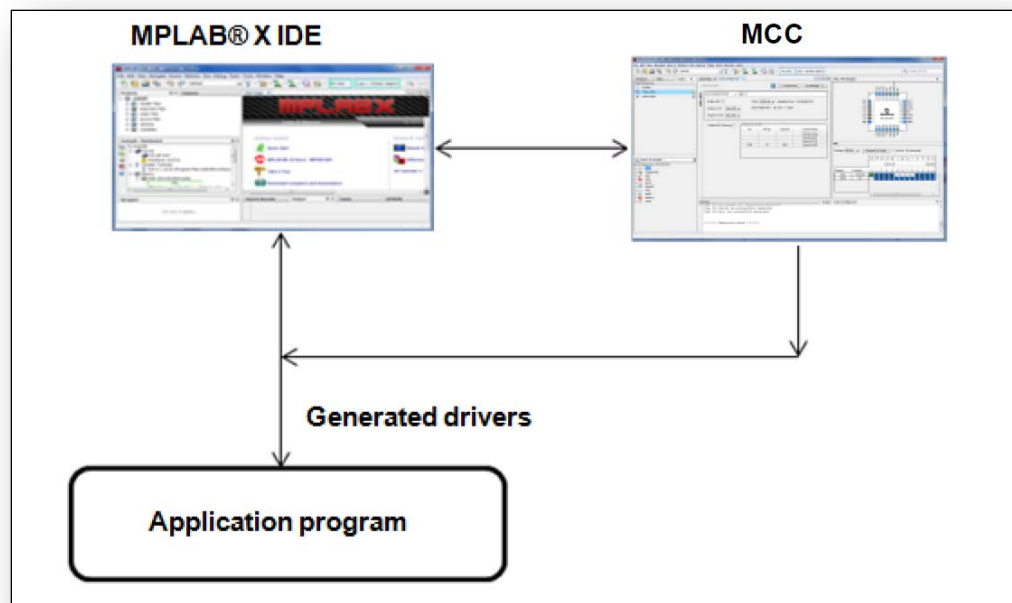


About XC Compiler Version

- MPLAB XC Compiler提供四種不同版本的C Compiler:
 - 標準版(Standard) (Standard Compiler Workstation License, 495 USD)
付費版本,擁有完整編譯功能及最佳化(Optimizations)的功能,可跟精簡版相比,最高可以減少20%~25%的記憶體空間。
 - 專業版(Pro) (Pro Compiler Workstation License, 995 USD)
付費版本,擁有完整編譯功能及最佳化(Optimizations)的功能,可跟精簡版相比,最高可以減少50%的記憶體空間。
 - 精簡版(Lite)
免費版本,擁有完整編譯功能,與正式版唯一的差異,就是沒有完整的最佳化功能,只擁有Level 1最佳化功能。
 - 評估版(Evaluation)
免費版本,安裝後60天內,具有標準版的所有功能,60天後自動變成精簡版。
- 需要付費的標準版(Standard) 跟專業版(Pro), 其授權區分成 Workstation License跟Network Server License兩種。

MPLAB® Code Configurator

- MPLAB Code Configurator, MCC。是一個圖形化的程式碼設定工具。透過MCC可以圖形化的方式,快速的產生周邊模組的初始化。除此之外,被選用周邊所需的控制函數也會一併產生。透過MCC可以大幅加快程式的設計流程,並減少重複性的程式碼撰寫工作。
- MCC套件以 Plug-in 的方式,整合到MPLAB X IDE 中。
- MCC目前支援大部分的 Microchip 8 Bits與16 Bits MCU。



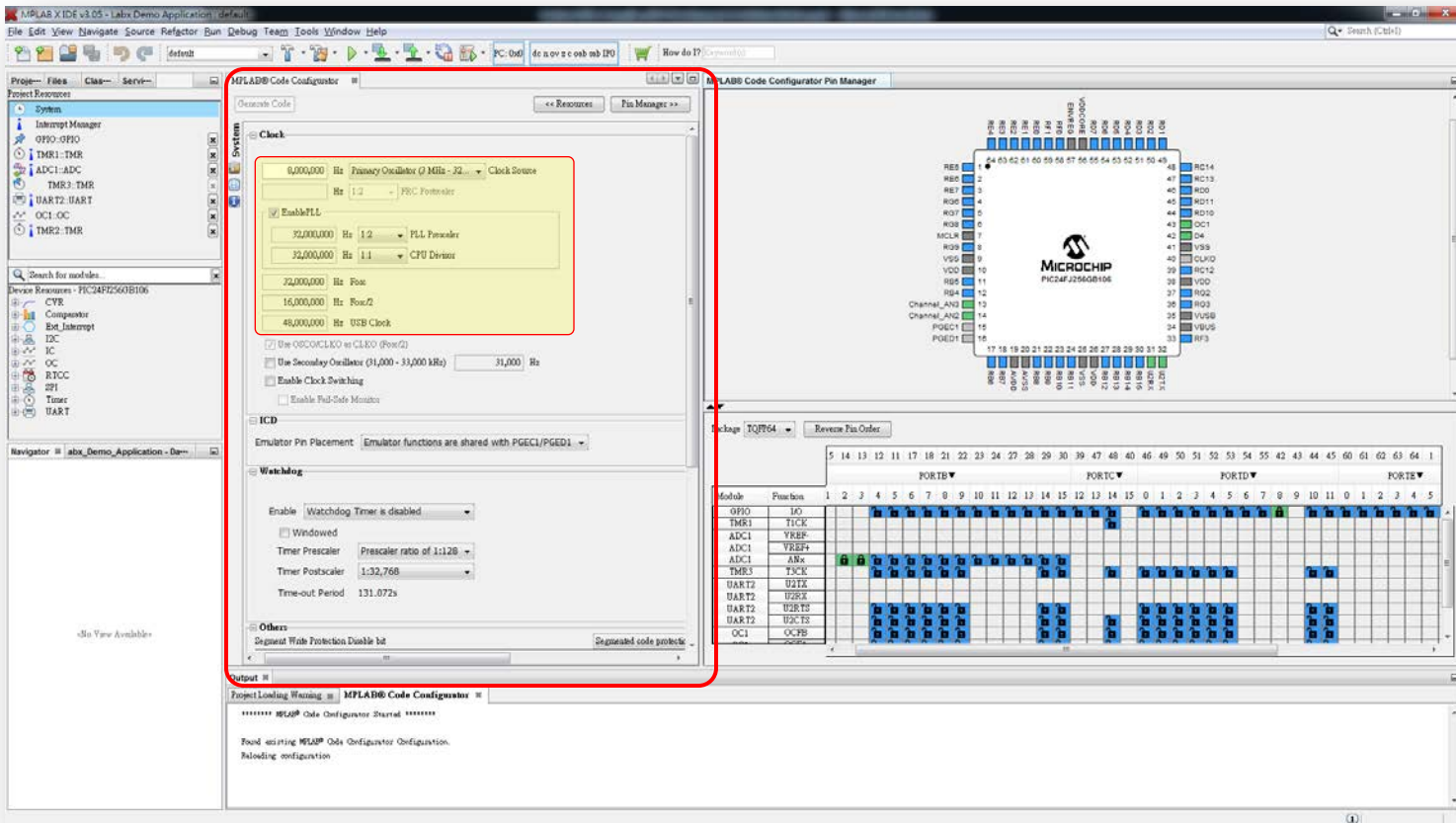
- 接腳管理區





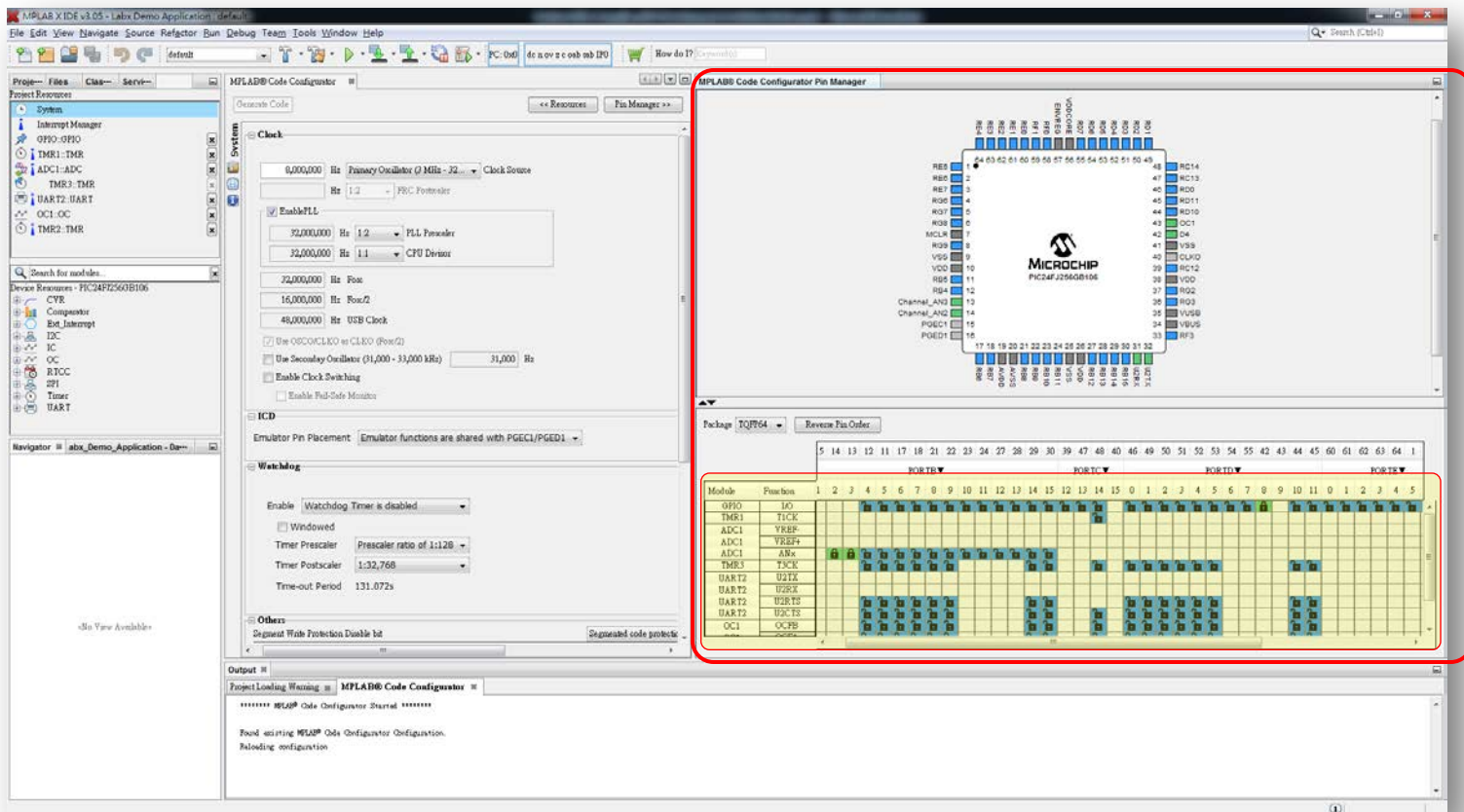
MCC Quick View

- **圖形化設定系統時脈, Configuration Word設定更直覺。**



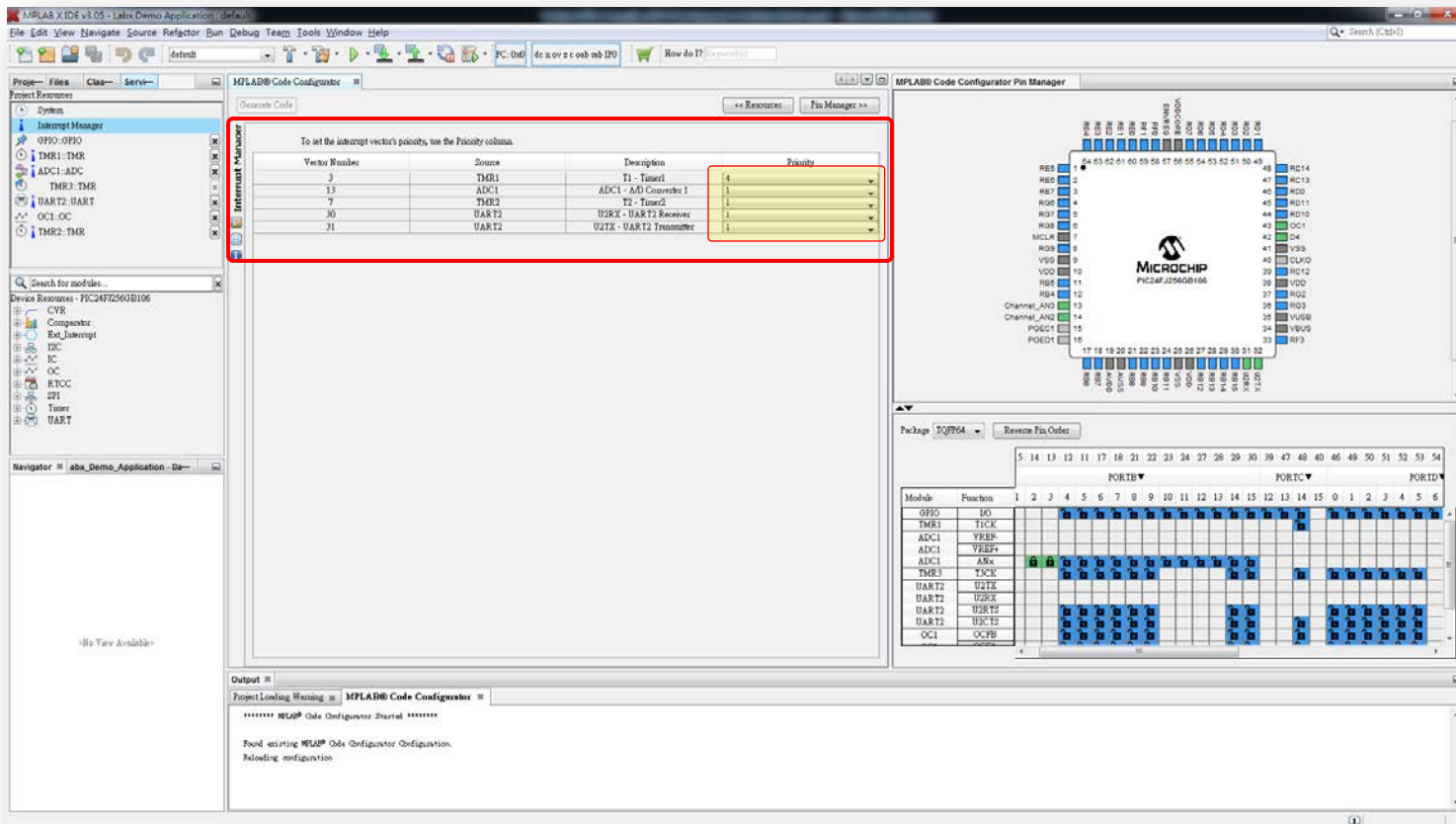
MCC Quick View

- 圖形化接腳指定, 衝突一目了然, 自動產生對應的PPS程式碼。



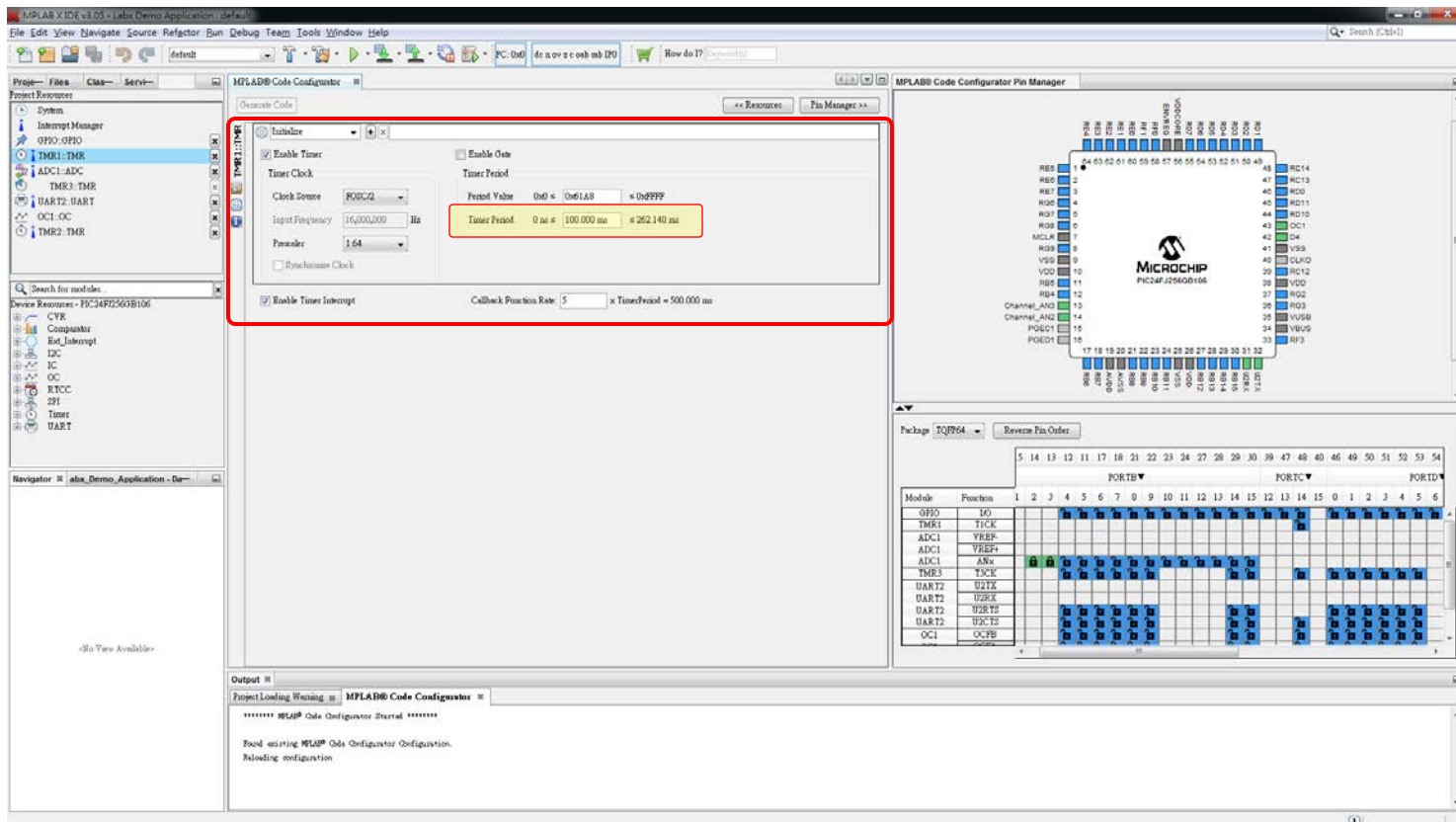
MCC Quick View

- 圖形化的中斷設定介面, 更易於管理, 中斷相關函式會自動產生, 不需要擔心語法錯誤與遺漏問題。



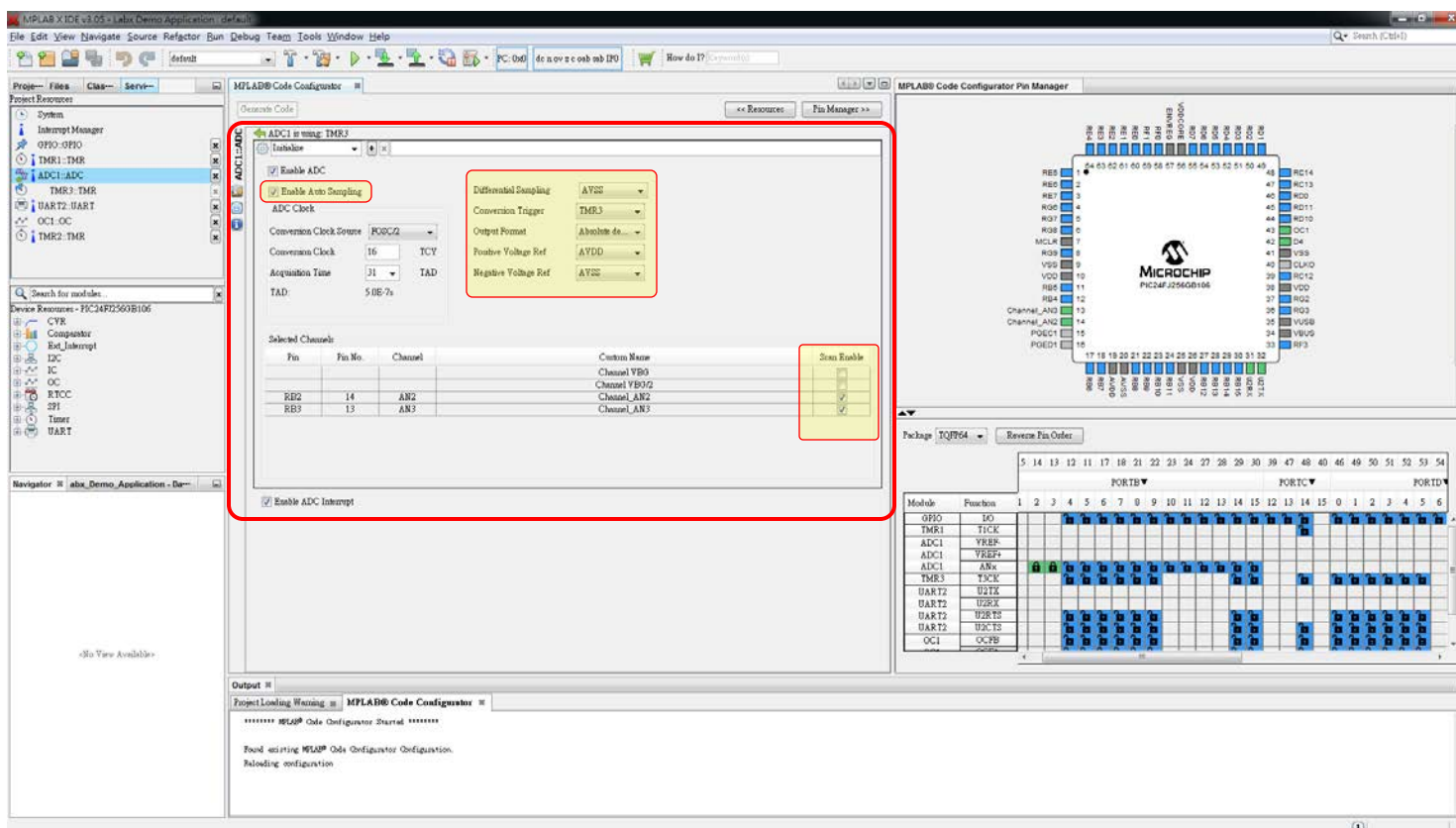
MCC Quick View

- 周邊模組得設定更加直覺, Timer設定不再需要繁複的計算。



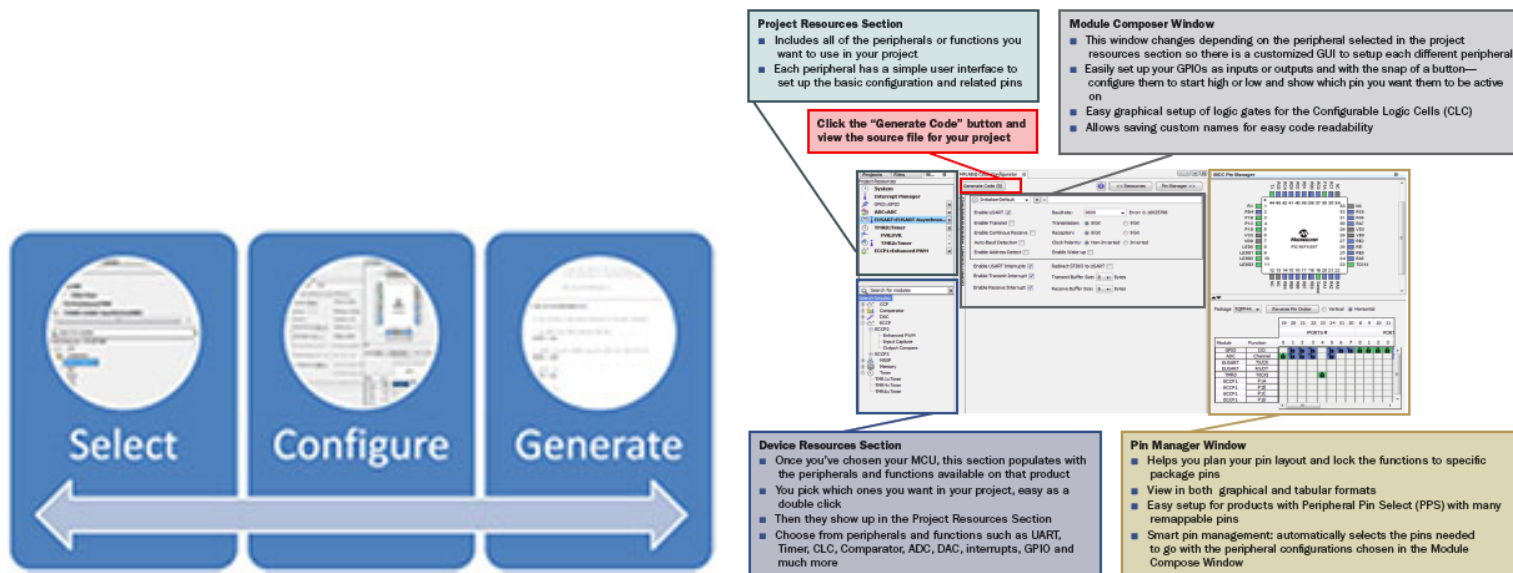
MCC Quick View

- ADC設定只需透過滑鼠點選, 接腳類比模式自動設定, 觸發與掃描模式皆可設定。



MCC Quick View

- 更多的功能, 在後續的內容中可以看到更多...



Tools Installation

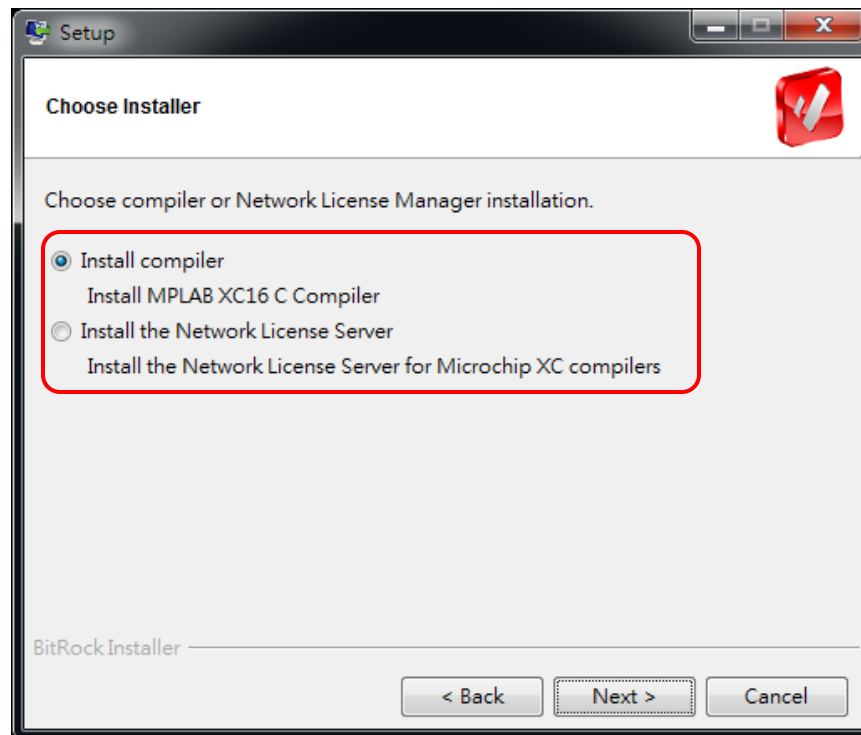
- 開發工具可以使用 Microchip RTC 教育訓練光碟或至以下連結取得 http://www.microchip.com.tw/Data_CD/
- MPLAB X IDE v3.20**
MPLAB XC16 v1.24

開發軟體 編譯器

MPLAB® X IDE	v3.20 Windows(Local)
	Windows Version Linux Version Mac Version Detail Info.
MPLAB® IDE	v8.92(Local)
MPLAB® XC8	v1.35(Local) *Peripheral libraries not include
• Legacy Peripheral Libraries	v2.00 RC3(Local) v1.34(Local)
MPLAB® XC16	v1.25(Local) *Peripheral libraries not include
• Legacy Peripheral Libraries	v2.0(Local) v1.24(Local)
MPLAB® XC32	v1.40(local) *Peripheral libraries not include
• Legacy Peripheral Libraries	v20150617(Local) v1.34(Local)

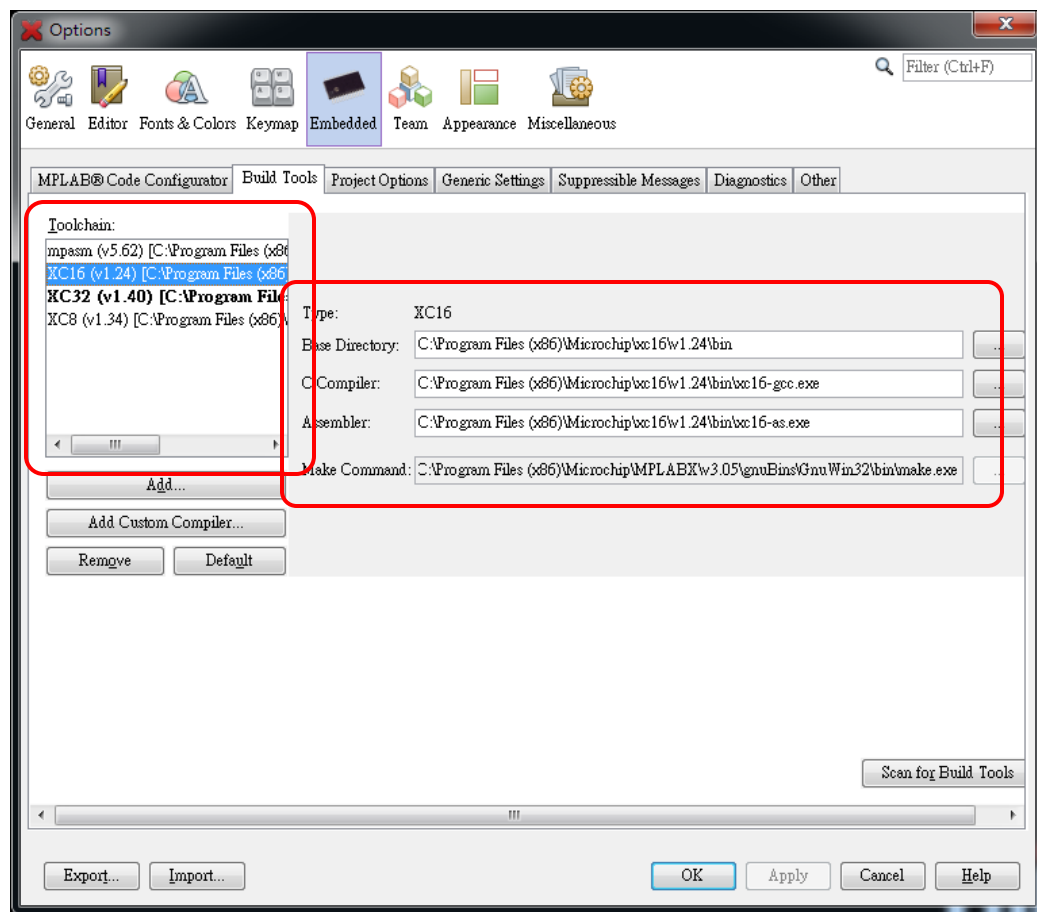
MPLAB XC16 Install

- MPLAB IDE安裝時, 並不會安裝MPLAB XC16。必須自行安裝。
- MPLAB XC16安裝時, 預設都是精簡版。如果要使用其他版本, 必須透過後續的線上註冊來改變。
- 本次課程會使用v1.24的精簡版, 預設安裝路徑為
C:\Program Files\Microchip\XC16\v1.24\
- 評估版必須透過線上申請。專業版, 標準版則必須付費取得授權後, 再透過線上啟用。



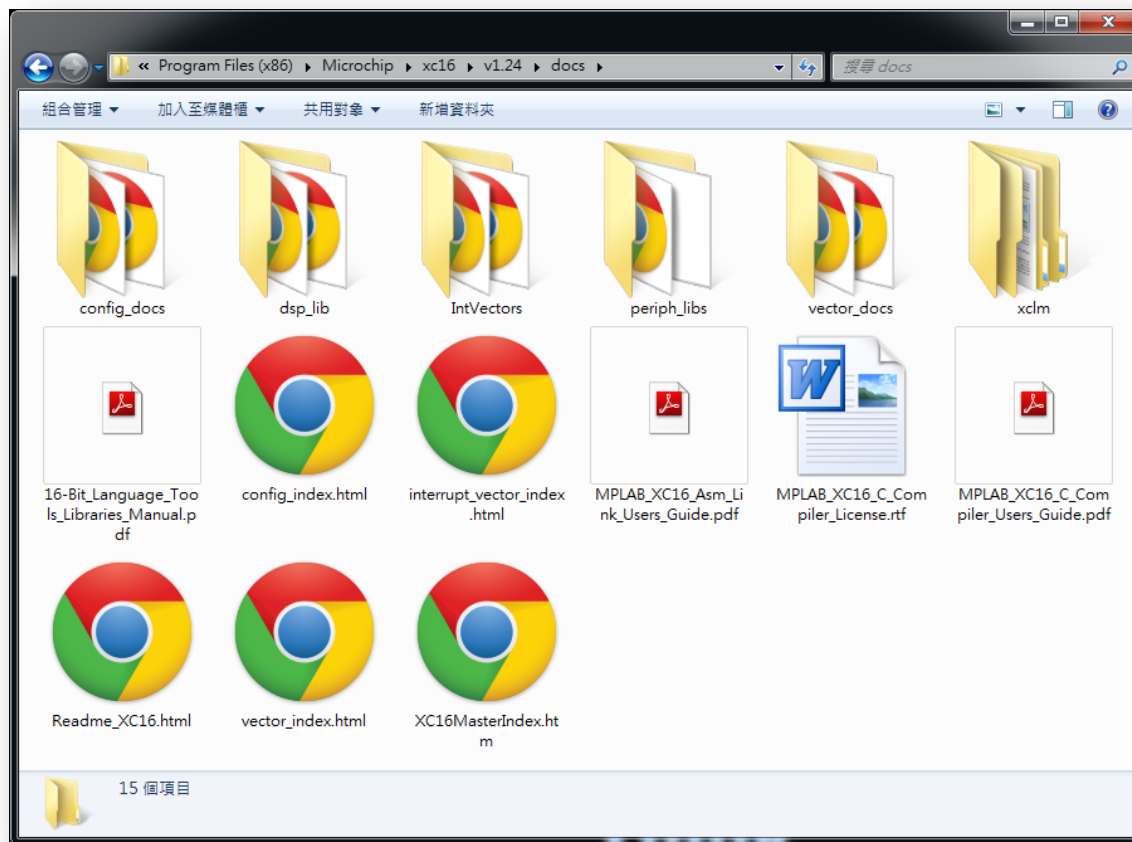
Install Status Check

- 安裝完成後, 可以使用MPLAB X IDE功能表中的Tools\Options來確認。
- 先選擇Tools\Options, 然後點選“Embedded”, 再選擇“Build Tools”, 確認 XC16 已正確安裝。



MPLAB XC16's Documents

- MPLAB XC16的相關文件,可以
C:\Program Files\Microchip\XC16\vx.xx\docs\裡找到。



MCC Installation

- MCC的安裝，可以使用線上安裝或離線安裝方式。



線上安裝

a 功能表 選擇

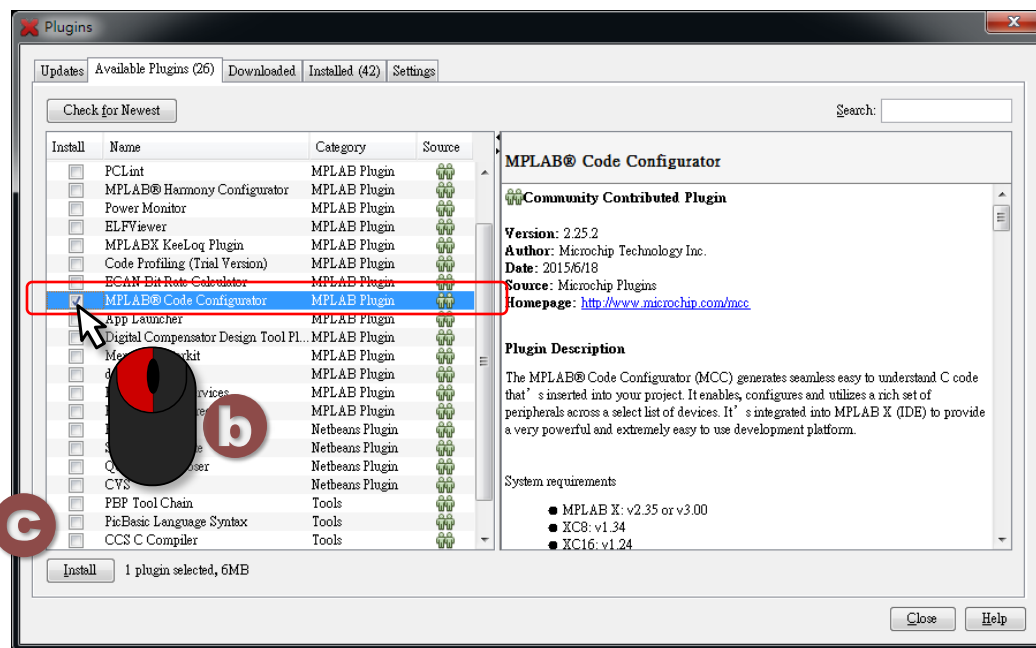
- Tools ► Plugins
- Available Plugins

b 選擇

- MPLAB Code Configurator

c 選擇

- **Install**



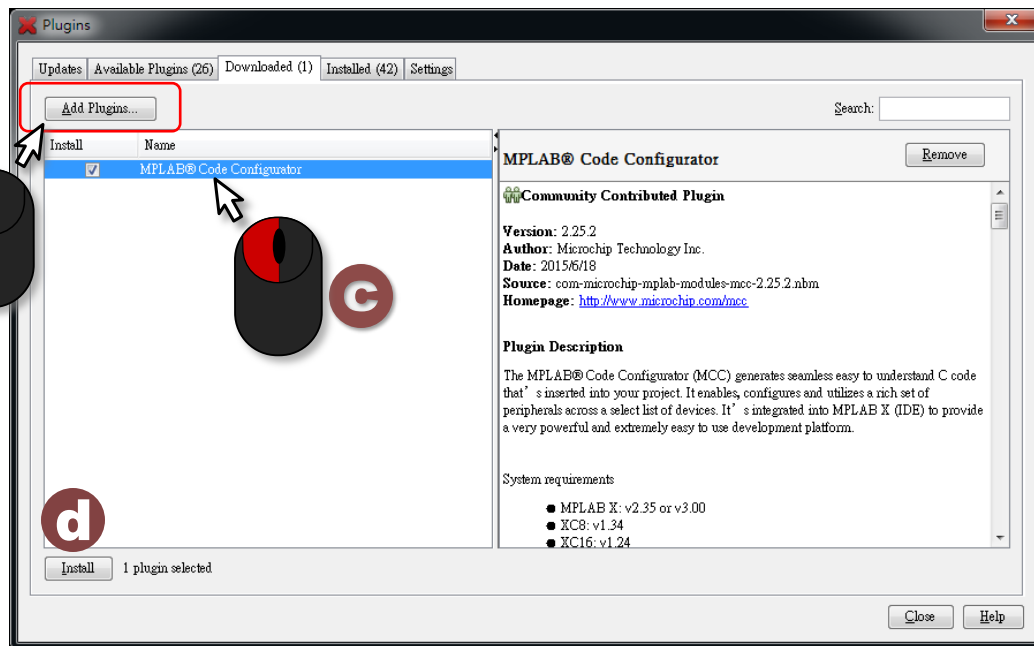
MCC Installation

- MCC的安裝，可以使用線上安裝或離線安裝方式。



離線安裝

- a** 功能表 選擇
 - ▶ **Tools ▶ Plugins**
 - ▶ **Downloaded**
- b** 選擇
 - ▶ **Add Plugins**, 並指定檔案
- c** 選擇
 - ▶ **MPLAB Code Configurator**
- d** 選擇
 - ▶ **Install**



MCC Plug-in Download

- MCC離線安裝的檔案可以使用 Microchip RTC 教育訓練光碟或至以下連結取得 http://www.microchip.com.tw/Data_CD/
- **MPLAB Code Configurator v2.25.2**

MPLAB® XC32	v1.40(Local) *Peripheral libraries not include
• Legacy Peripheral Libraries	v20150617(Local)
	v1.34(Local)

MPLAB® C18 Lite	v3.47(Local)
MPLAB® C30 Lite	v3.31(Local)
MPLAB® C32 Lite	v2.02a(Local)

HI-TECH C for PIC10/12/16	v9.83(Local)
HI-TECH C for PIC18	v9.80(Local)

MPLAB® X IDE Plug-in

MPLAB® Code Configurator	v2.25.2(Local) v2.10.3(Local) v2.1.0(Local) v2.0.1(Local)
--------------------------	---

MPLAB XC16's Data Type

- 資料型別(Data Type)是用來定義資料存放時所佔記憶體的大小及其被處理的方式,MPLAB XC16支援以下幾種資料型別(Data Type):

整數(Integer)資料型別

Type	Bits	Min	Max
char, signed char	8	-128	127
unsigned char	8	0	255
short, signed short	16	-32768	32767
unsigned short	16	0	65535
int, signed int	16	-32768	32767
unsigned int	16	0	65535
long, signed long	32	-2^{31}	$2^{31} - 1$
unsigned long	32	0	$2^{32} - 1$
long long**, signed long long**	64	-2^{63}	$2^{63} - 1$
unsigned long long**	64	0	$2^{64} - 1$
** ANSI-89 extension			

浮點數(Float)資料型別

Type	Bits	E Min	E Max	N Min	N Max
float	32	-126	127	2^{-126}	2^{128}
double*	32	-126	127	2^{-126}	2^{128}
long double	64	-1022	1023	2^{-1022}	2^{1024}
E = Exponent N = Normalized (approximate) * double is equivalent to long double if -fno-short-double is used.					

Access SFRs

- 每個特殊功能暫存器(Special Function Register)都有特定的名稱與位址。MPLAB XC16為了使用SFRs, 必需使用連結檔(Linker Script File, *.gld)來定義其名稱與位址, 並透過MCU的標頭檔(Header File)來定義其屬性並引用, 才可以正確存取SFRs。
- E.g.:
`extern volatile unsigned int LATB __attribute__((__sfr__));`
- 因此使用SFRs時,必須先含入(include)對應MCU的標頭檔才能正確使用。
E.g.:
`#include <p33fj128mc804.h>`
`#include <p24f16ka102.h>`
- 標頭檔(Header File)的命名方式並非所有使用者都很熟悉,因此建議可直接含入(include)通用標頭檔(Generic Header File),由MPLAB XC16自己來尋找正確的標頭檔。
E.g.:
`#include <p33fxxxx.h> // for dsPIC33`
`#include <p24fxxxx.h> // for PIC24`

Access SFRs

- 透過連結檔(Linker Script File, *.gld)的定義與MCU標頭檔(Header File)引用, 才可以正確的存取SFRs。例如:
TRISB = 0xFFFE;
LATB = 0x0001;
- MCU標頭檔(Header File)中, 也增加了各個SFRs結構變數的定義, 因此也可以透過存取對應的結構變數, 存取SFRs內的特定成員 (*SFRNAMEbits.BITNAME*)。如:
TRISBbits.TRISB0 = 0;
LATBbits.LATB0 = 1;
- TRISB及TRISBbits都參考到同一SFRs的位址(透過Linker Script指定)。因此操作TRISB或TRISBbits會得到相同的結果。
- 每個SFRs的結構型態定義都不同,建議直接打開標頭檔(Header File)來觀察看看。

MCU's Header File

- p24fj16ka102.h MCU標頭檔(Head File)的內容片段

C:\Program Files (x86)\Microchip\xc16\v1.24\support\PIC24F\h\

```
extern volatile unsigned int LATB __attribute__((__sfr__));
```

```
typedef struct tagLATBBITS
```

```
{
```

```
    unsigned LATB0:1;
```

```
    unsigned LATB1:1;
```

```
    unsigned LATB2:1;
```

```
    unsigned LATB3:1;
```

```
    unsigned LATB4:1;
```

```
    unsigned LATB5:1;
```

```
    ....;
```

```
    ....;
```

```
    ....;
```

```
    unsigned LATB13:1;
```

```
    unsigned LATB14:1;
```

```
    unsigned LATB15:1;
```

```
} LATBBITS;
```

```
extern volatile LATBBITS LATBbits __attribute__((__sfr__));
```

TABLE 4-13: PORTB REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11 ⁽³⁾	TRISB10 ⁽³⁾	TRISB9	TRISB8	TRISB7	TRISB6 ⁽³⁾	TRISB5 ⁽³⁾	TRISB4	TRISB3 ⁽³⁾	TRISB2	TRISB1	TRISB0	FFFF
PORTB	02CA	RB15	RB14	RB13	RB12	RB11 ⁽³⁾	RB10 ⁽³⁾	RB9	RB8	RB7	RB6 ⁽³⁾	RB5 ⁽³⁾	RB4 ⁽³⁾	RB3 ⁽³⁾	RB2	RB1 ⁽¹⁾	RB0 ⁽¹⁾	000000
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11⁽³⁾	LATB10⁽³⁾	LATB9	LATB8	LATB7	LATB6⁽³⁾	LATB5⁽³⁾	LATB4	LATB3⁽³⁾	LATB2	LATB1	LATB0	000000
ODCB	02CE	ODB15	ODB14	ODB13	ODB12	ODB11	ODB10	ODB9	ODB8	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0	0000

連結檔 (Linker Script File)

- p24fj16ka102.gld 連結檔(Linker Script File)的內容片段:

```
TRISB      = 0x2C8;  
_TRISB     = 0x2C8;  
_TRISBbits = 0x2C8;  
PORTB      = 0x2CA;  
_PORTB     = 0x2CA;  
_PORTBbits = 0x2CA;  
LATB       = 0x2CC;  
_LATB      = 0x2CC;  
_LATBbits = 0x2CC;
```

- 連結檔(Linker Script File)中除了定義SFR的位址外,也定義了資料與程式記憶體的大小,位置,重置向量(Reset Vector),中斷向量表的位址等。
(中斷架構章節會詳細說明)

```
MEMORY  
{  
    data(a!xr):ORIGIN = 0x800,LENGTH = 0x0600  
    reset:ORIGIN = 0x0,LENGTH = 0x4  
    ivt:ORIGIN = 0x4,LENGTH = 0xFC  
    _reserved:ORIGIN = 0x100,LENGTH = 0x04  
    aivt:ORIGIN = 0x104,LENGTH = 0xFC  
    program(xr):ORIGIN = 0x200,LENGTH = 0x2A00  
    ...  
}  
  
.ivt __IVT_BASE :  
{  
    LONG( DEFINED(__ReservedTrap0) ?  
    ABSOLUTE(__ReservedTrap0):  
    ABSOLUTE(__DefaultInterrupt));  
    ....  
}
```