



MICROCHIP

Regional Training Centers

Section 2
IDE, Compiler, MCC &
Development Tools Introduction

MPLAB® X IDE

- ◆ New generation integrated Development Tools, Support PIC Series MCU, Provide Plug-in function to extend more advance function. Java Based, Cross platform, Current version is v5.10.
- ◆ The newest version (5.x) getting started to support AVR and SAM families. and newest tools, like ICD4, PICkit4 & Atmel ICE.



MPLAB XC16

- ◆ **New Generation Compiler, Support all 16 Bits MCU (PIC24, dsPIC).**
- ◆ **Base on GNU C, apply GPL License (GNU General Public License).**
- ◆ **Provide standard C libraries (printf, strlen, etc..) and Peripheral Libraries (old versions).**
- ◆ **All version you can download from www.microchip.com/xc16**



About XC Compiler Version

- ◆ There 4 different versions for XC Compiler:

- ◆ **Standard** (Standard Compiler Workstation License)

Charge. Provide Standard Optimizations function.
Save 20%~25% memory size (Max.).

- ◆ **Pro** (Pro Compiler Workstation License)

Charge. Provide Standard Optimizations function.
Save 50% memory size (Max.).

- ◆ **Lite**

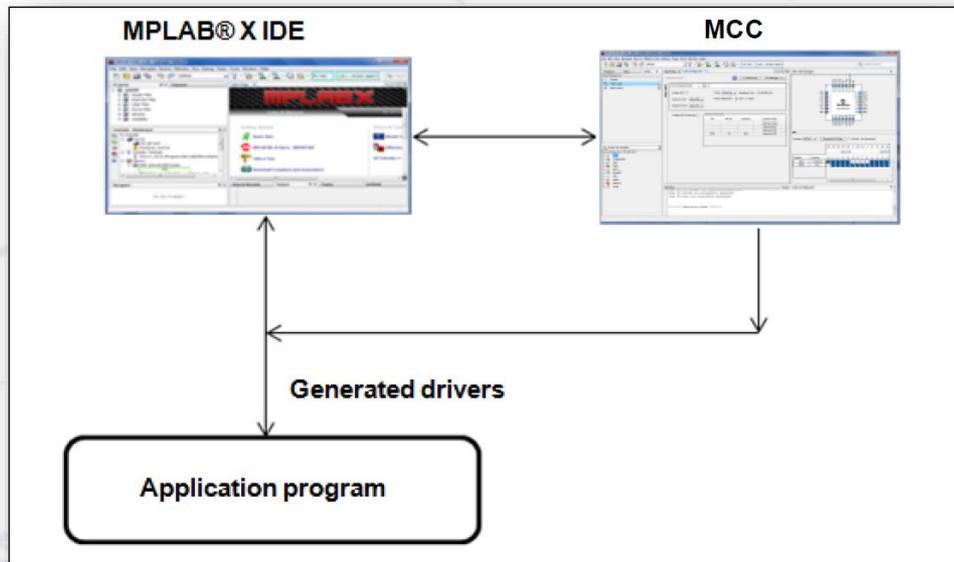
Free. Provide Basic Optimizations function (Level 1).

- ◆ **Evaluation**

Free. Provide Standard Optimizations function if you apply trial license.
Change to Lite version when license expires, automatically.

MPLAB® Code Configurator

- ◆ **MCC (MPLAB Code Configurator)** is a free, graphical programming environment that generates seamless, easy-to-understand C code to be inserted into your project.
- ◆ **Supports 8-bit, 16-bit and 32-bit PIC® microcontrollers.**



MCC Quick View

The screenshot displays the Microchip Configuration (MCC) software interface. The main window is titled "System Module" and shows the "Clock" configuration section. The clock source is set to "8000000 Hz" using the "FRC Oscillator" (8.0 MHz). The "FRC Postscaler" is checked, and the output is set to "4 MHz" with a "1:2" postscaler ratio. The "PLL Enable" checkbox is unchecked. The "Clock Output Pin Configuration" is set to "OSCO functions as CLKO (FOSC/2)". The "Use Secondary Oscillator" checkbox is checked, and the frequency is set to "(31 - 33) kHz". A red box highlights the "Clock" section, and a purple arrow points from a callout box to the "Use Secondary Oscillator" checkbox.

System clock Graphically. Configuration Word settings are more clear and simple.

MCC Quick View

The screenshot displays the MPLAB X IDE v5.10 interface. The central window is titled "Pin Manager: Package View" and shows a grid for assigning pins to various modules. The grid is organized by module and function, with columns representing pin numbers 0 through 14 for Port B. A red box highlights the "Pin Module" section, which includes GPIO input and output pins. A purple arrow points from a yellow callout box to the GPIO output row in the table.

Module	Function	Direction	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Clock	CLKI	input															
	CLKO	output															
	OSCI	input															
	OSCO	output															
	SOSCI	input															
ICD	SOSCO	output															
	PGCx	input															
Pin Module	PGDx	input															
	GPIO	input															
	GPIO	output															

**Pin assign Graphically.
PPS Code generate automatically.**



MCC Quick View

Interrupt Manager

Module	Interrupt	Description	IRQ Nu...	Enabled	Priority
Pin Module	CNI	CN - Change Notifica...	19	<input type="checkbox"/>	1
TMR2	TI	T2 - Timer2	7	<input checked="" type="checkbox"/>	1
TMR2	TNI	T3 - Timer3	8	<input type="checkbox"/>	1
TMR1	TI	T1 - Timer1	3	<input checked="" type="checkbox"/>	1
ADC1	ADI	ADC1 - A/D Converte...	13	<input checked="" type="checkbox"/>	1

Interrupt assign Graphically. ISR related code generate automatically.



MCC Quick View

More ?

www.microchip.com/mcc

MPLAB[®] PICKit[™] 4

Debugger/Programmer Probe

- ◆ Fast programming, increased functionality, at the same price as its predecessor.
- ◆ The MPLAB[®] PICKit[™] 4 In-Circuit Debugger/Programmer allows fast and easy debugging and programming of **PIC[®]**, **dsPIC[®]**, and **CEC** flash microcontrollers.
- ◆ An additional **micro SD** card slot and the ability to be self-powered from the target means you can take your code with you and **Program on To Go**.



Tools Download

- At this courses, you should install MPLAB X IDE, MCC & XC16 Compiler for all hands-on exercises.

✦ MPLAB X IDE v5.10, MCC v3.66, XC16 v1.35.

- You can download development tools from below link.

http://www.microchip.com.tw/Data_CD/

開發軟體, 編譯器

MPLAB® X IDE	v5.10 Windows(Local) Windows Version Linux Version Mac Version Detail Info.
MPLAB® IDE	v8.92(Local)
MPLAB® XC8	v1.45(Local) *Peripheral libraries not include
• Part Support Patch Files	v1.45b(Local)
• Peripheral Libraries	v2.00RC3(Local) v1.34(Local)
MPLAB® XC16	v1.35(Local) *Peripheral libraries not include
• Part Support Patch Files	v1.35(Local)
• Peripheral Libraries	v2.00(Local)
MPLAB® X IDE Plug-in	
MPLAB® XC32	v3.66(Local) v3.65.1(Local) v3.65(Local) v3.55.1(Local) v3.36(Local) v3.26.4(Local) v3.25(Local) v3.16(Local) v3.15(Local) v3.05(Local) v2.25.2(Local)
• Part Support Patch Files	
• Peripheral Libraries	v1.34(Local)

Tools Install

- ◆ **For MPLAB X IDE & XC16 Compiler installation, it's very easy. You just need next, next, next then finished.**
- ◆ **For MCC installation, it's a X IDE plug-in, you need install MPLAB X IDE before MCC.**
- ◆ **There are two way to install MCC.**
 - ◆ On Line Installation.
 - ◆ Off Line Installation.

MCC Installation



On Line Installation

a Menu

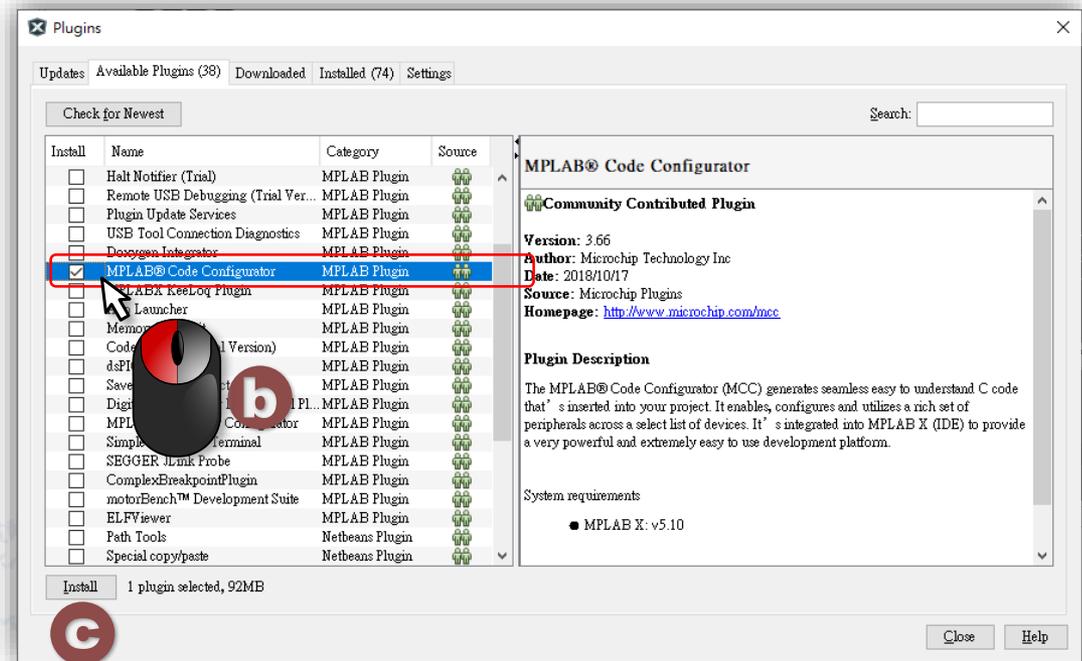
- ▶ Tools ▶ Plugins
- ▶ Available Plugins

b Select

- ▶ MPLAB Code Configurator

c Select ▶

Install



MCC Installation



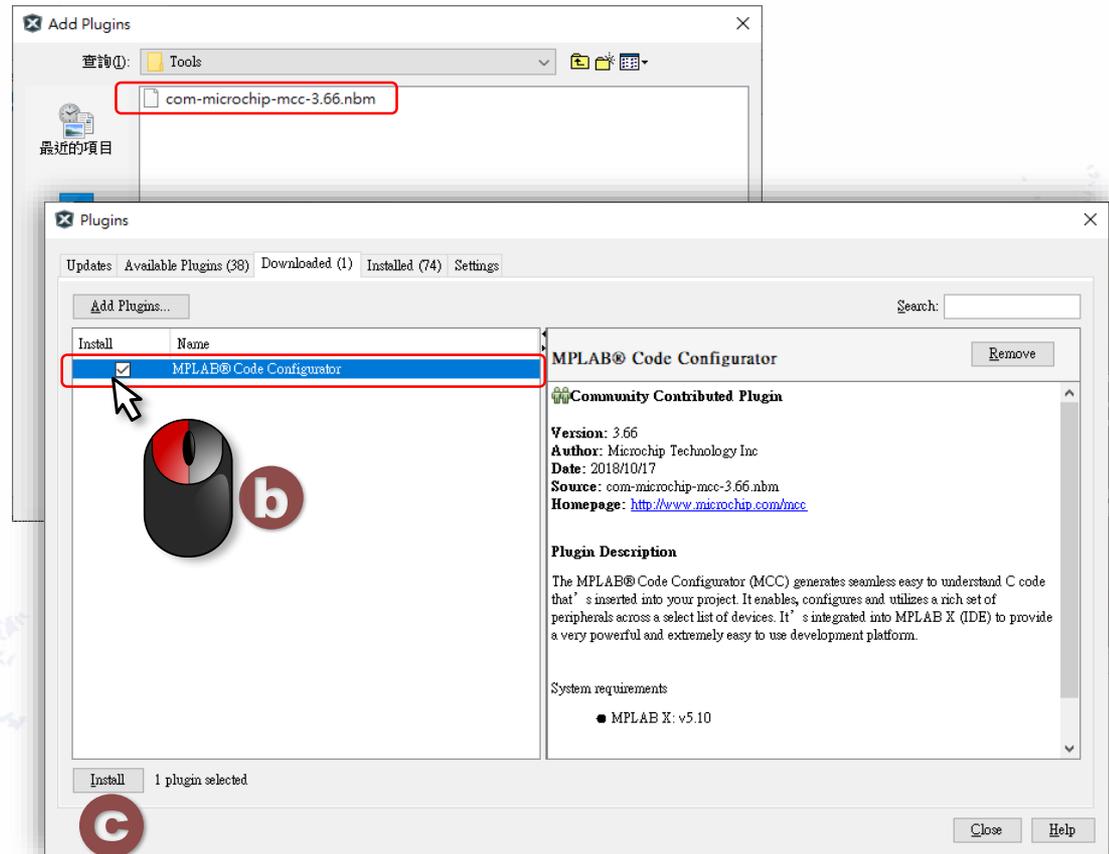
Off Line Installation

- a** Menu
- ▶ Tools ▶ Plugins
 - ▶ Download

- b** Select
- ▶ Add Plugins
 - ▶ Select mccxxx.nbm

- c** Select ▶

Install



MPLAB XC16's Data Type

- ◆ Data type is first thing you need know, about compiler.
- ◆ **The same data type has different definition at different platforms.**

Integer Data Type

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

Float Data Type

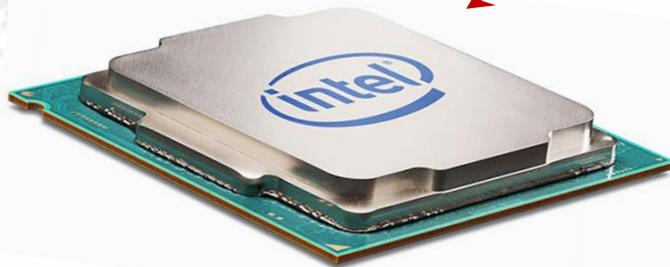
Type	Bits	E Min	E Max	N Min	N Max
float	32	-126	127	2^{-126}	2^{128}
double*	32	-126	127	2^{-126}	2^{128}
long double	64	-1022	1023	2^{-1022}	2^{1024}

E = Exponent
N = Normalized (approximate)
* double is equivalent to long double if -fno-short-double is used.

Thinking !!

- What kind differ when same code execute at different platform?

```
unsigned int i = 0;  
main()  
{  
    for( i= 0 ; i < 100000 ; i++ );  
    // Here ?  
}
```



<http://www.qmo.tw/security20180108027>

