



MICROCHIP

Regional Training Centers

Section 3

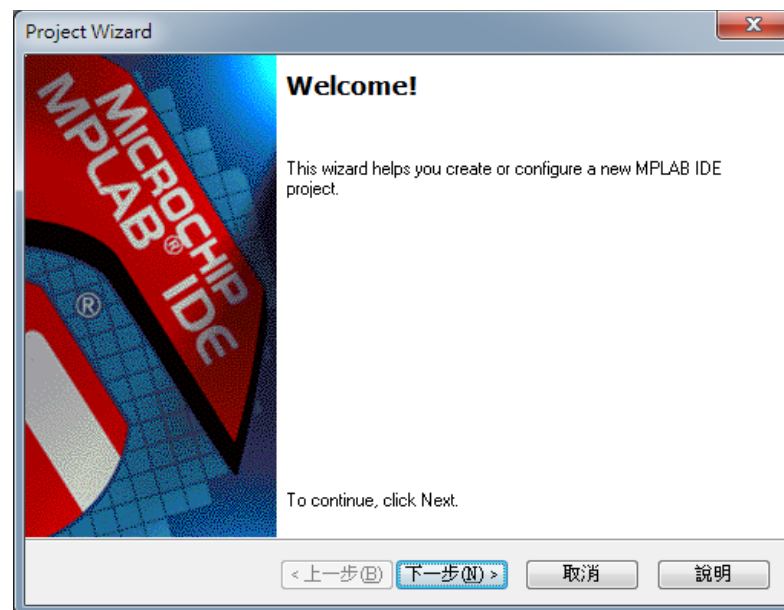
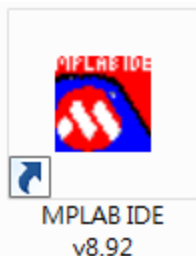
Getting Started First Project

Lab1 - First PIC32 Project

- 嘗試建立自己的第一個MPLAB IDE專案,並以MPLAB XC32做為預設的編譯器。
 - 在專案中建立自己的第一個程式FirstProject.c。
 - 透過建立專案,建立程式的過程了解MPLAB IDE的基本操作,語言工具的設定,以及軟體模擬的各項功能。
-
- 該如何開始?

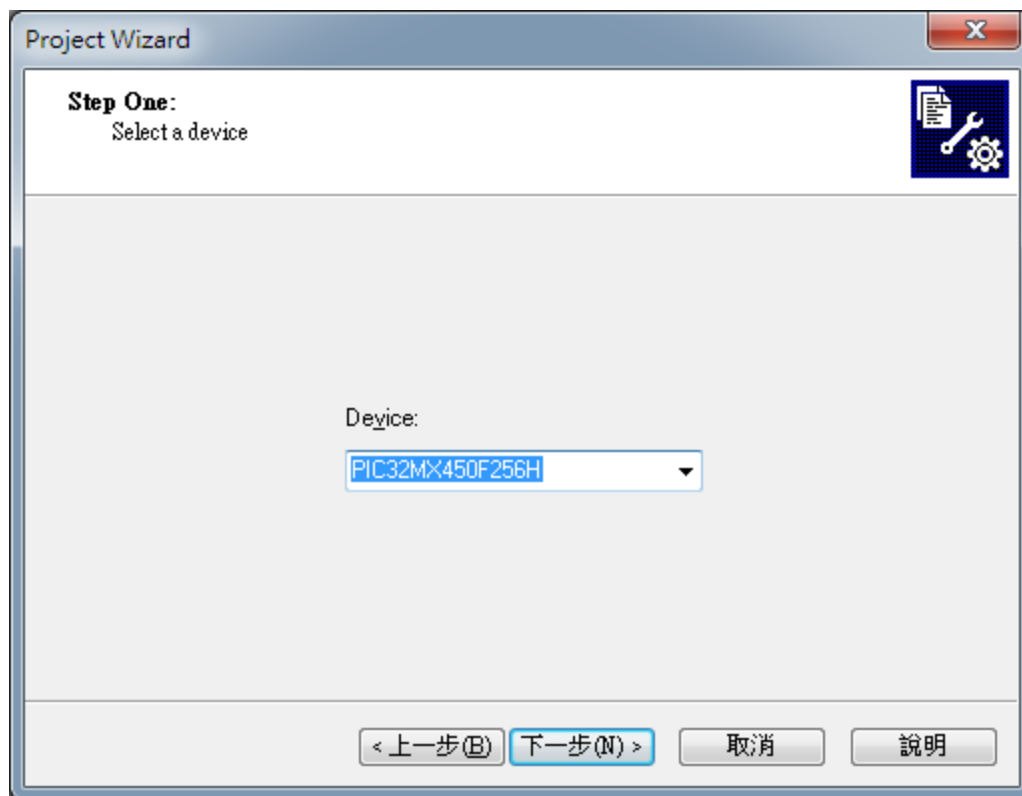
Lab1 - Create Project Step1

- 開始前, 請先安裝以下幾套軟體
 - 1.MPLAB IDE v8.92(整合式開發環境)
 - 2.MPLAB XC32 v1.30(Microchip 32-Bits MCU的C Compiler)
- 然後開啟MPLAB IDE, 選擇功能表\Project\Project Wizard。開啟專案精靈後, 直接點選下一步。



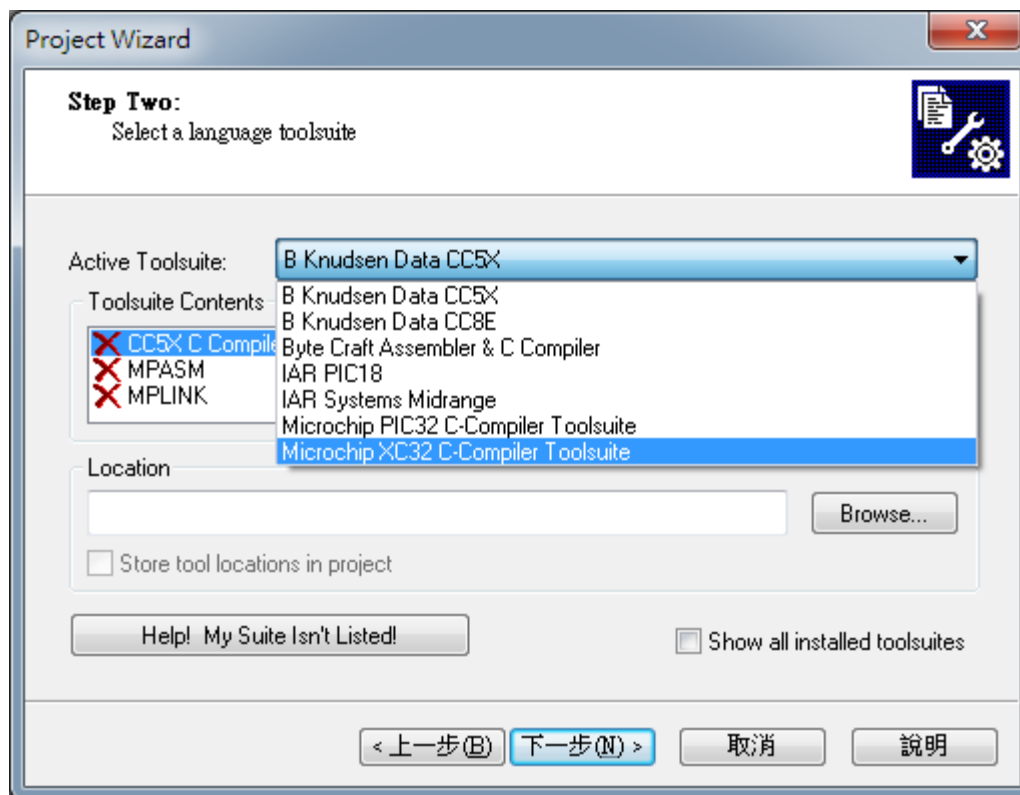
Lab1 - Create Project Step2

- 選擇Device編號 **PIC32MX450F256H**(此次課程使用的MCU編號)
確定後點選下一步。



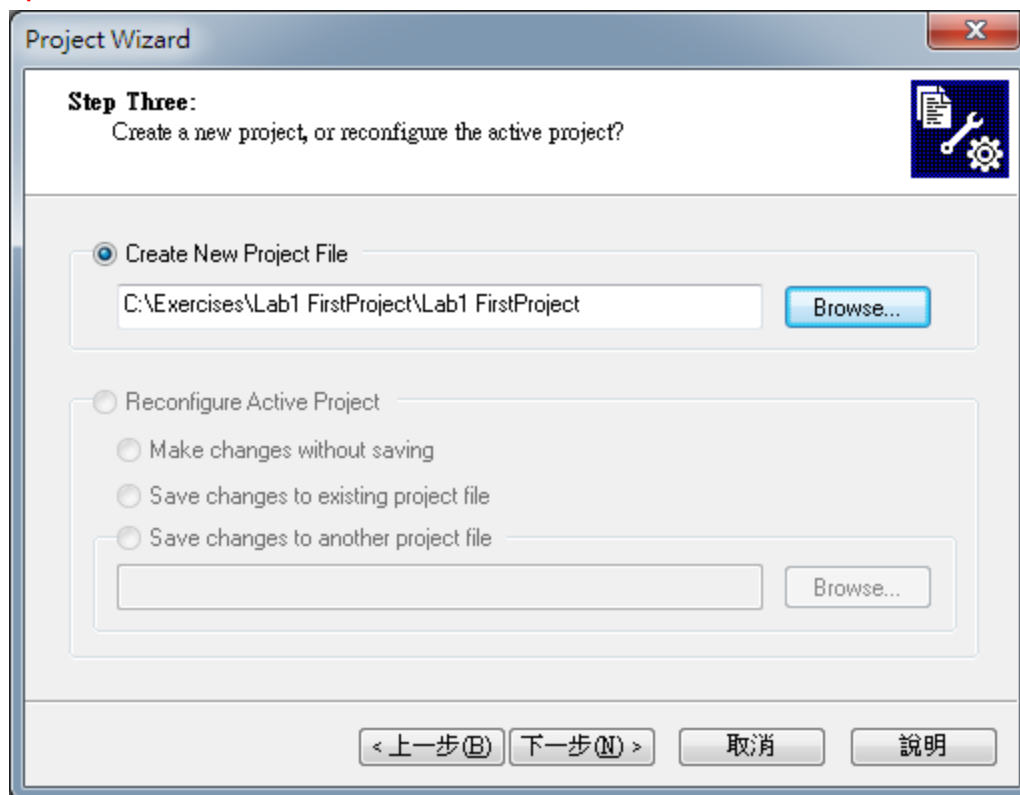
Lab1 - Create Project Step3

- 選擇使用的編譯器為Microchip XC32 C-Compiler Toolsuite。
確定後點選下一步。



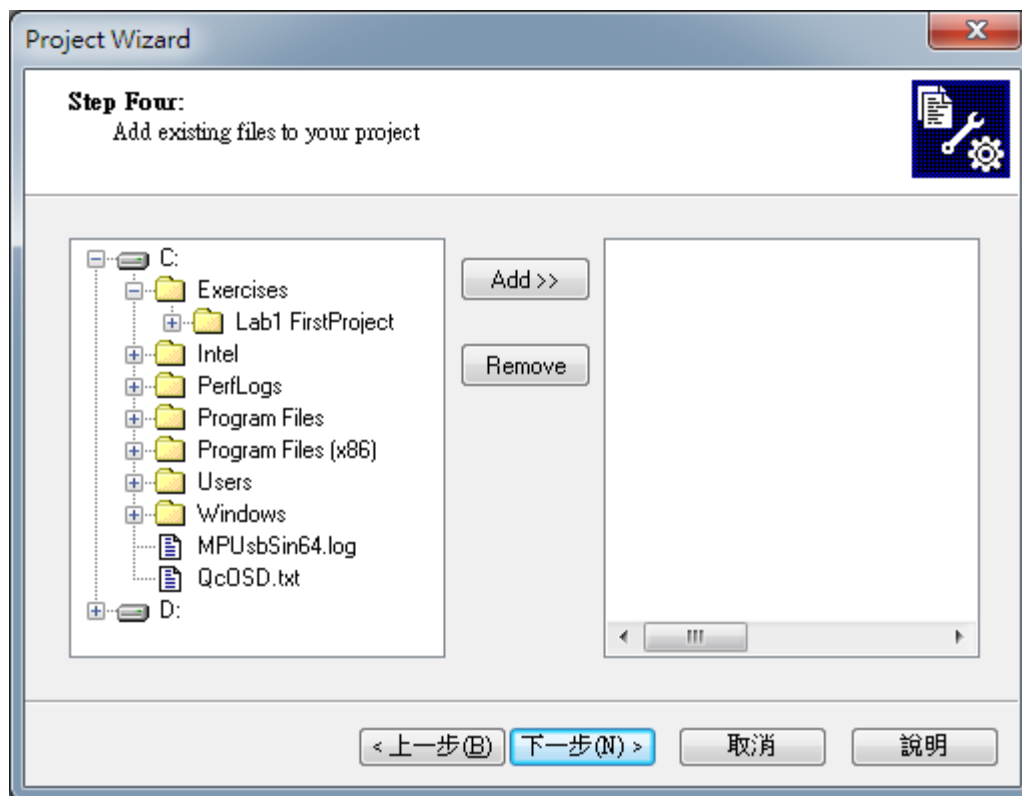
Lab1 - Create Project Step4

- 使用 Browse 按鈕或直接輸入專案檔名稱,及存放路徑。確定後點選下一步,專案檔的路徑及檔名,建議與Source code放於同一目錄下。(MPLAB IDE不支援 中文檔名/路徑, 桌面是在中文路徑下, 不支援)



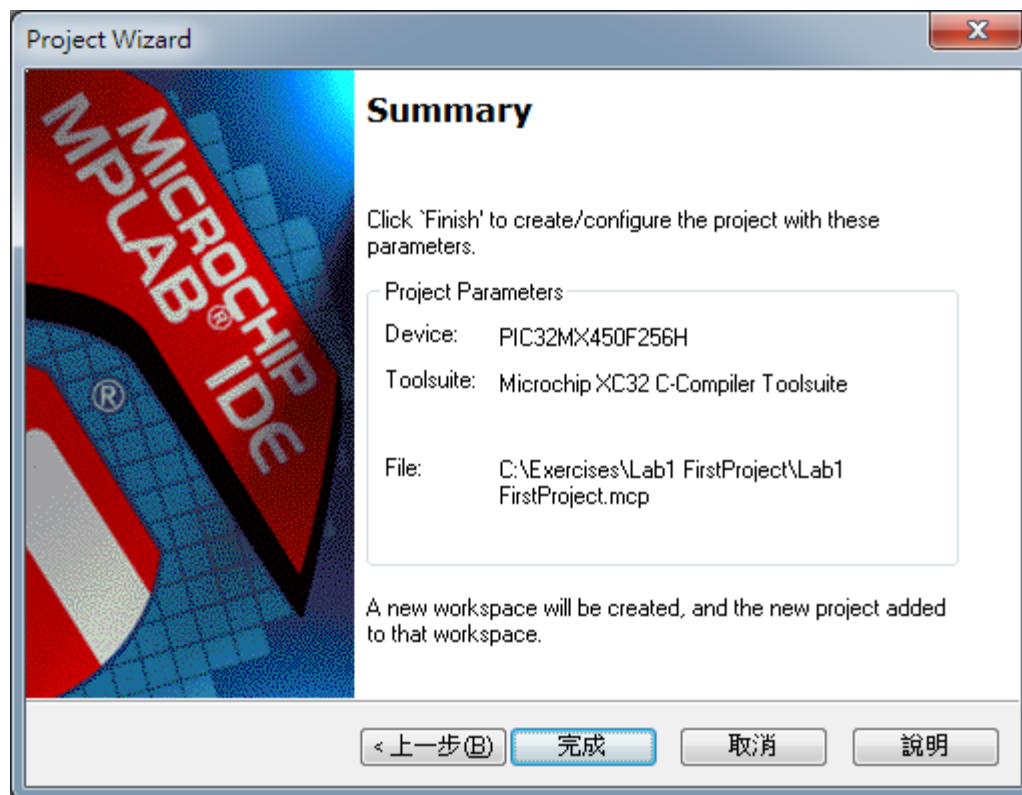
Lab1 - Create Project Step5

- 此步驟可以加入已經存在的檔案到專案中。但Lab1沒有已經存在的檔案, **直接點選下一步。**



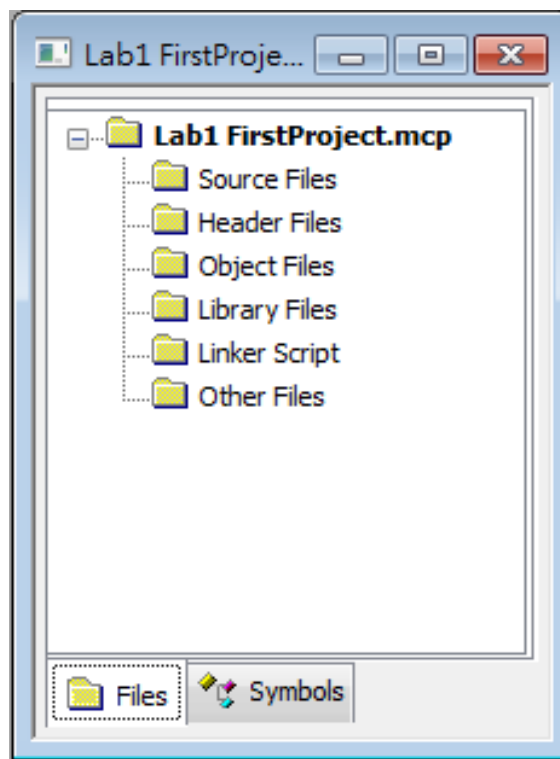
Lab1 - Create Project Step6

- 檢查Summary,若一切無誤則點選完成。



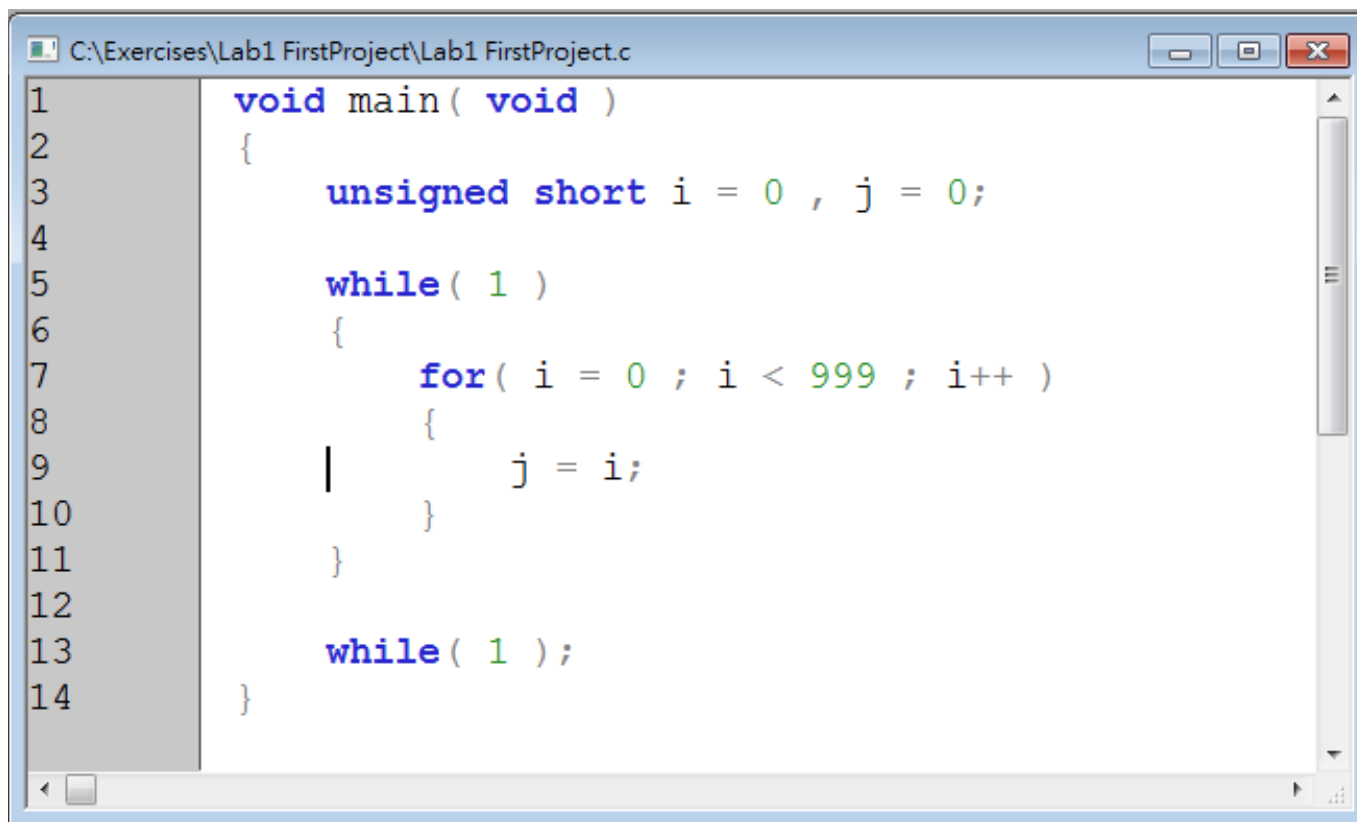
Lab1 - Create Project Step7

- 專案建立後, 可在專案視窗中看到專案的的相關檔案。目前是一個空的專案。



Lab1 - Create Project Step8

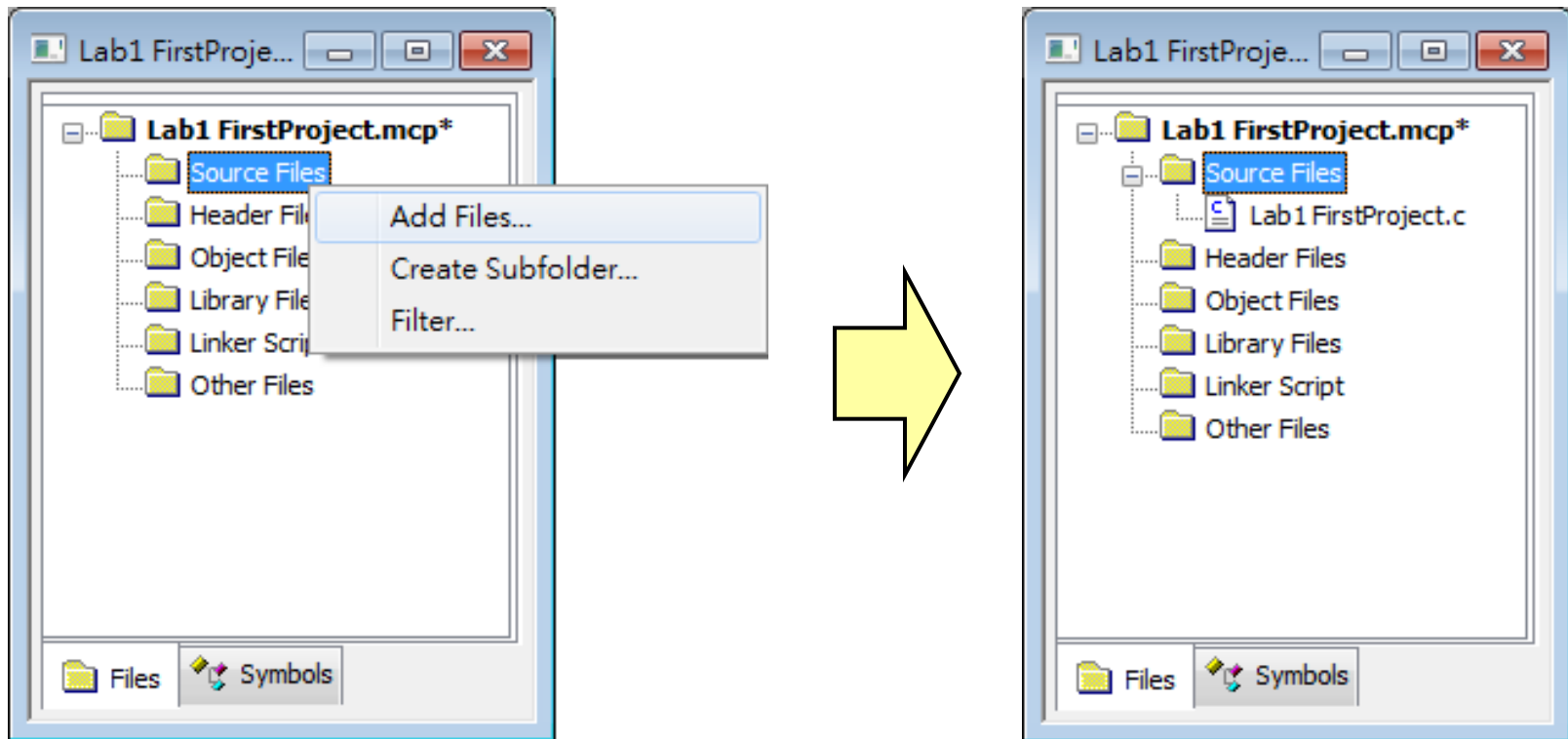
- 請選擇功能表 File\New,開啟一個新的檔案,並輸入以下程式碼,然後存檔為FirstProject.c,放置於Lab1目錄下。




```
C:\Exercises\Lab1 FirstProject\Lab1 FirstProject.c
1 void main( void )
2 {
3     unsigned short i = 0 , j = 0;
4
5     while( 1 )
6     {
7         for( i = 0 ; i < 999 ; i++ )
8         {
9             j = i;
10        }
11    }
12
13    while( 1 );
14 }
```

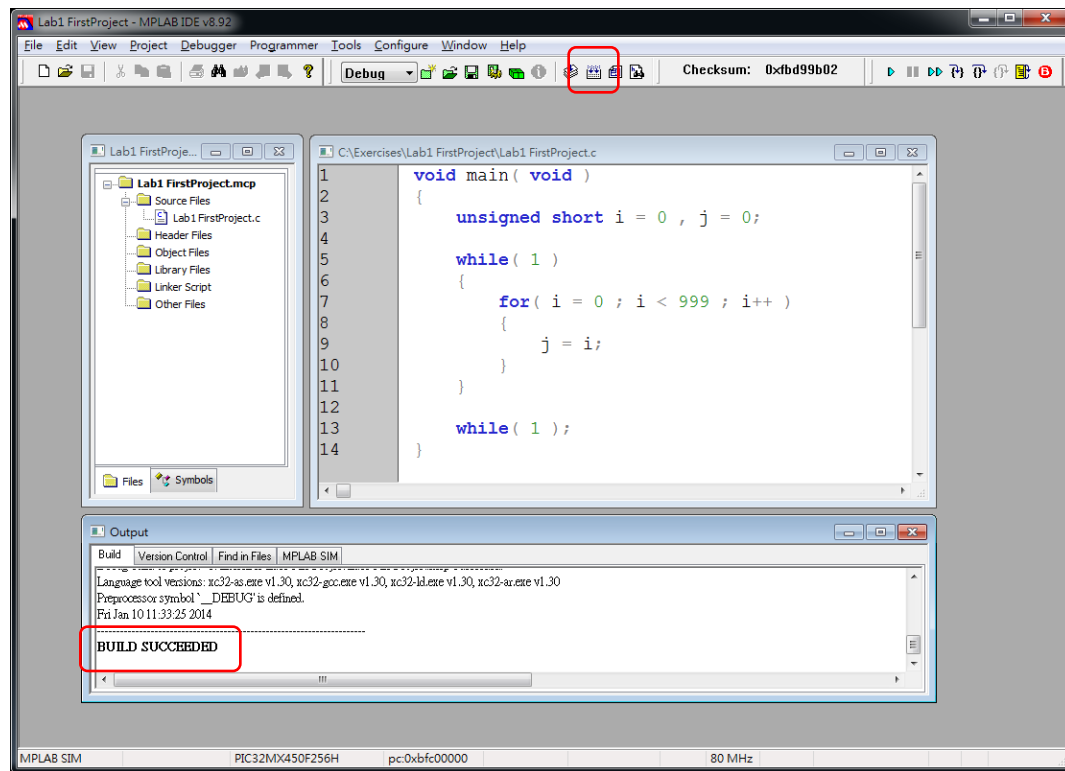
Lab1 - Create Project Step9

- 接著將FirstProject.c加到專案檔中。
專案視窗, Source Files資料夾上按滑鼠右鍵, Add Files。



Lab1 – Build & Debug Step1

- 到此,一個完整的專案建立完成,接著可以嘗試進行編譯程式,看看剛剛撰寫的程式是不是正確。
- 點選功能表Project\Build All或Build All圖示  進行編譯。
- 如果編譯成功,可以在Output視窗中看到“BUILD SUCCEEDED”。
- 如果失敗呢 ?

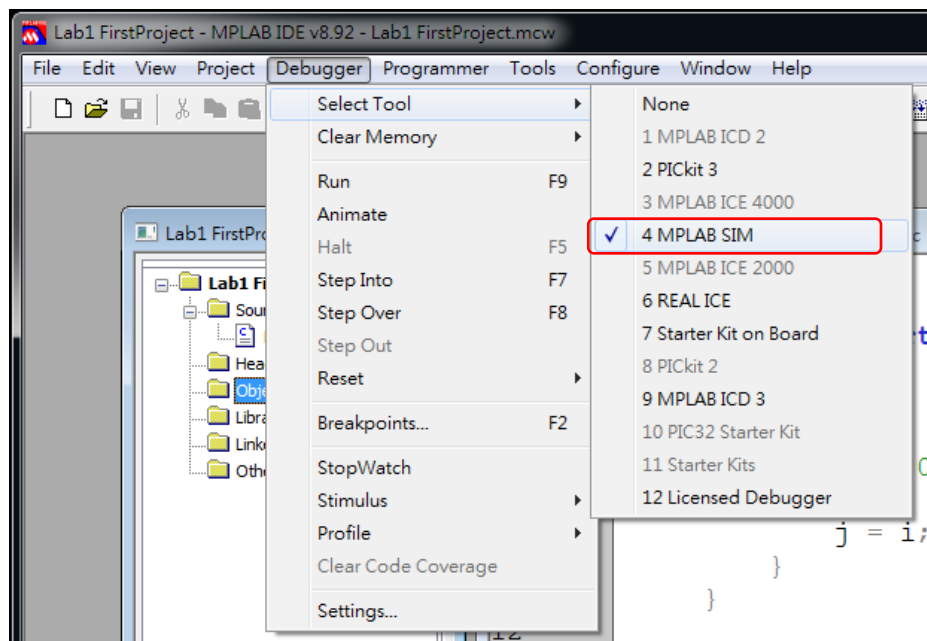


Build Error ?

- 若 Build 的結果有誤,可仔細閱讀Output視窗所提示錯誤訊息。通常可以用滑鼠雙擊錯誤訊息,就會跳到程式中有錯誤的地方。接著觀察,冷靜的判斷,找出錯誤。
- 常見的錯誤:
 - 大小寫不一致:
C 語言中大小寫不同就代表不一樣的名稱。
 - 變數未宣告,或重複宣告:
變數一定要先被宣告才能使用,且僅能宣告一次。若要使用在其他檔案所宣告的變數,則需加上`extern`修飾詞。
 - 敘述(Statement)區塊不完整:
少了敘述結尾的分號(;)或Statement Block的大括弧({ , })。
 - 引數或參數列不對稱:
呼叫函式時的參數數量或型別不正確。
Ex: `void MuFunc(int , int , int);` 呼叫時要給3個整數型態的參數。

Lab1 – Build & Debug Step2


- 編譯成功後,可用軟體模擬來測試程式的流程與變化是否正確。
- 先指定開發工具為MPLAB SIM。
功能表Debugger\Select Tool\MPLAB SIM。
- 接著就可以利用除錯工具列  執行Run, Halt, Step Into, Step Over等功能,來進程式流程的控制。



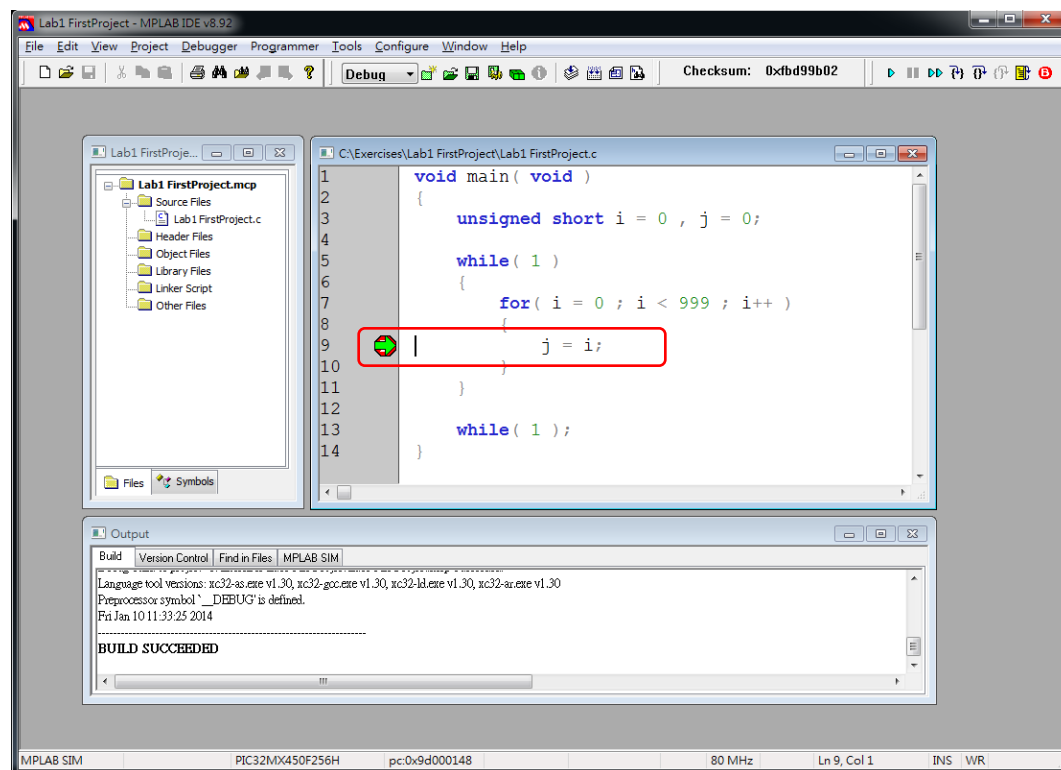
About Debug/Program Tools

- MPLAB SIM是屬於純軟體的模擬。可以在沒有實際開發板可以測試的時候,利用MPLAB SIM來模擬MCU的實際動作。
- PICKit3/ICD3/Real ICE則屬於硬體的除錯/燒錄器。可以搭配實際的硬體來進行除錯,也可做為單純的燒錄器,來進行MCU程式的燒錄。
- 如果要PICKit3/ICD3/Real ICE扮演除錯器(Debugger)的角色,可以在MPLAB IDE中,功能表Debugger\Select Tool選擇該工具。如果要作為燒錄器(Programmer)則在功能表Programmer\Select Tool該工具下選擇。

Lab1 – Build & Debug Step3

- 如果要設定斷點 , 可以利用滑鼠雙擊要設定斷點的地方。
- 只要執行到該行敘述時, 就會自動停下, 方便觀察程式的流程, 資料的變化, 是否符合預期。

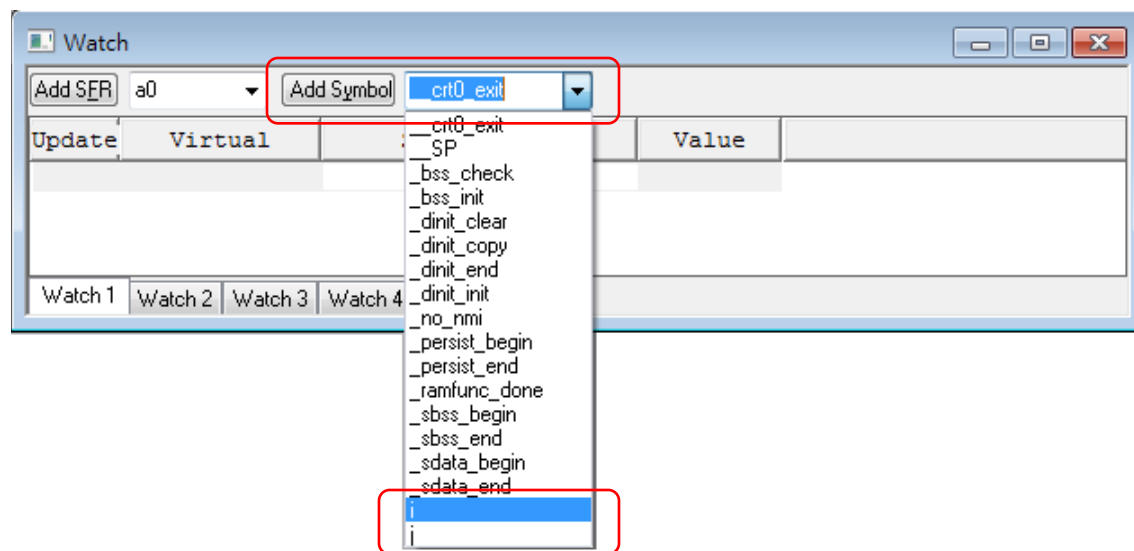
- 如何觀察變化呢 ?



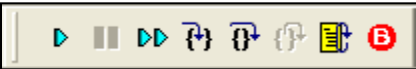
Watch Windows

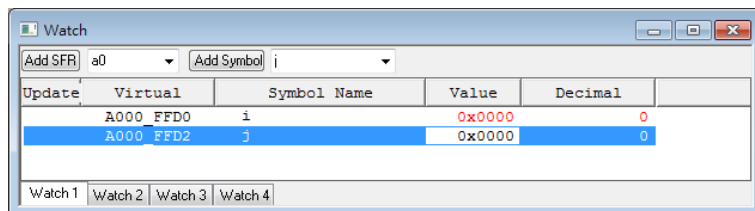
- Watch Windows可以用來觀看,想要監視的變數或暫存器。
- 程式在每次停下來時,MPLAB IDE會自動更新目前的數值,並且自動比對與前次數值的差異,所有不同,會以紅色標示,方便觀察。
- Watch Windows可以從功能表\View\Watch開啟。
- 點選Add SFR可以加入要監視的SFR。

點選Add Symbol
可以加入要監視的
變數或符號(Symbol)
等。



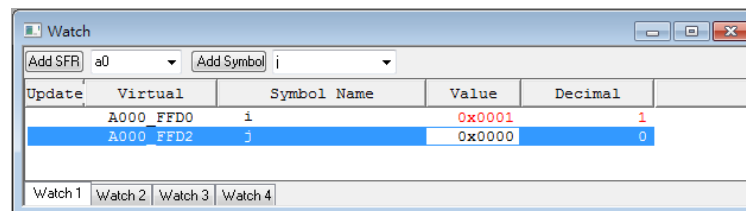
Lab1 – Build & Debug Step4

- 使用除錯工具列  來執行程式,並利用Watch Window來觀察每次停下來時的數值變化。是否符合預期。



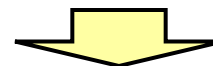
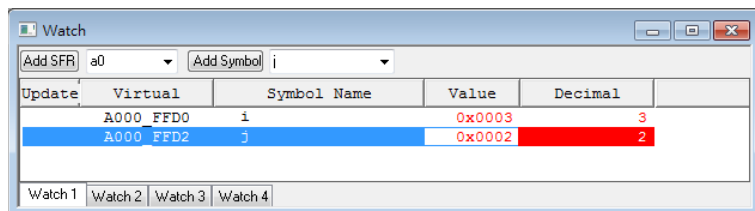
Watch Window Screenshot 1: Initial state. Watch 1 (A000_FFD0) has value 0x0000 (0). Watch 2 (A000_FFD2) has value 0x0000 (0).

Update	Virtual	Symbol Name	Value	Decimal
	A000_FFD0	1	0x0000	0
	A000_FFD2	3	0x0000	0

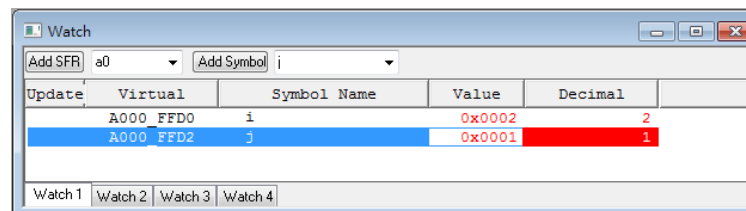
Watch Window Screenshot 2: After first instruction. Watch 1 (A000_FFD0) has value 0x0001 (1). Watch 2 (A000_FFD2) has value 0x0000 (0).

Update	Virtual	Symbol Name	Value	Decimal
	A000_FFD0	1	0x0001	1
	A000_FFD2	3	0x0000	0

Watch Window Screenshot 3: After second instruction. Watch 1 (A000_FFD0) has value 0x0003 (3). Watch 2 (A000_FFD2) has value 0x0002 (2).

Update	Virtual	Symbol Name	Value	Decimal
	A000_FFD0	1	0x0003	3
	A000_FFD2	3	0x0002	2

Watch Window Screenshot 4: After third instruction. Watch 1 (A000_FFD0) has value 0x0002 (2). Watch 2 (A000_FFD2) has value 0x0001 (1).

Update	Virtual	Symbol Name	Value	Decimal
	A000_FFD0	1	0x0002	2
	A000_FFD2	3	0x0001	1

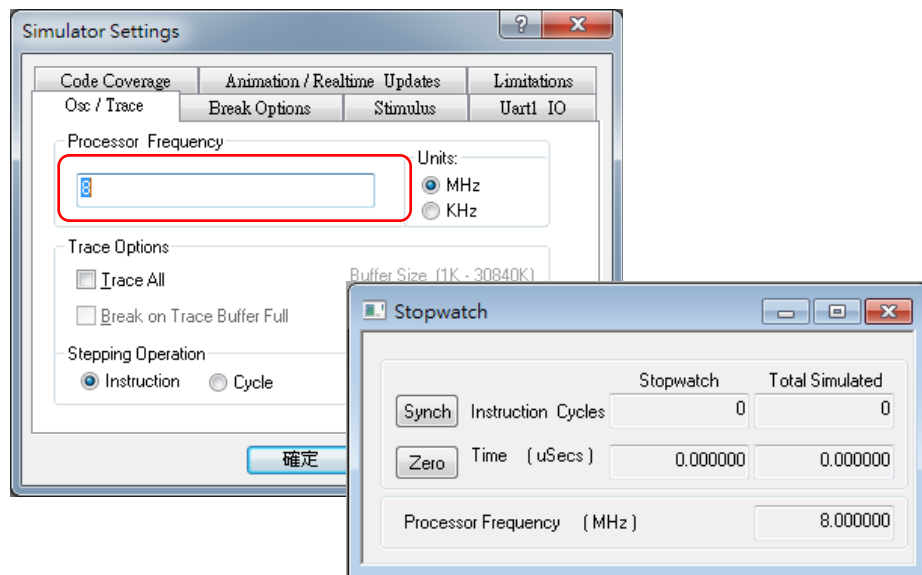
- 如何得知程式的執行時間 ?



Stopwatch

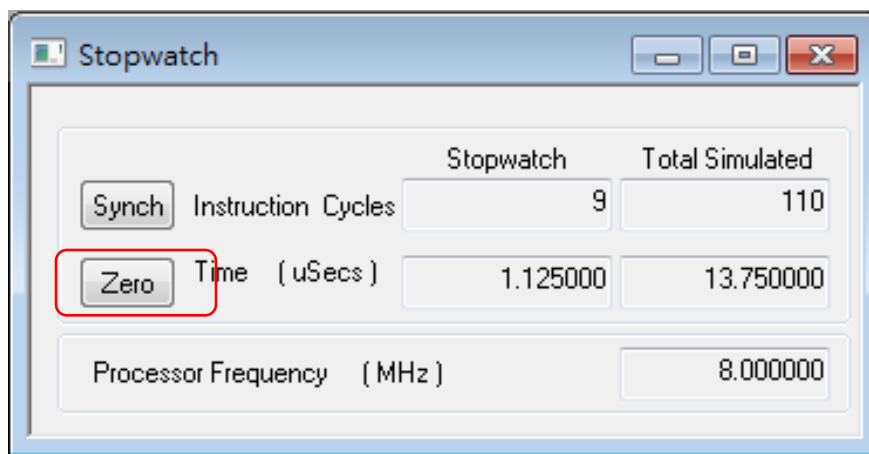
MPLAB SIM中的好用工具

- 選擇MPLAB SIM做為除錯工具時, 可以使用Stopwatch工具。Stopwatch只有在選擇MPLAB SIM做為除錯工具時才會出現。可以從功能表\Debugger\Stopwatch開啟。
- Stopwatch可以用來測量程式執行所消耗的時間, Stopwatch會計算程式在兩次停下來之間, 所花費的指令數及時間。是個相當好用的工具。
- Stopwatch是以花費的指令數搭配系統時脈, 來換算花費時間。因此必須先設定預定的工作時脈, 才能計算出正確的時間。請在功能表\Debugger\Settings下設定。



Lab1 – Build & Debug Step5

- 利用Stopwatch測測看,每跑一次for迴圈, 要花多少個指令數?
- 不同的C Compiler版本,對於同一個敘述所編譯出來的組合語言指令數量可能會有所不同,所以不同版本的Compiler所量測出來的指令數可能會不同。
- 可以使用Zero來歸零數值,重新計算。



Other Windows

- 功能表\View中還有許多Window可以用來觀看MCU內部的暫存器,程式記憶體,資料記憶體及特殊功能暫存器的數值。
- 使用硬體除錯工具時,建議不要持續開啟這些視窗,原因在於每次程式停下來的時候, MPLAB IDE會自MCU取得MCU內的資料,來更新所監視的數值,如果開太多視窗,就會有許多的資料更新的動作,會感覺軟體的反應變慢很多,感覺像當機一樣,但其時是還在進行資料更新。
- 強烈建議把要監視的數值,放入Watch Window中觀看。

