

MPLAB[®] IDE v6.10 使用手冊

版權及商標說明

Information contained in this publication regarding device applications and the like is intended by way of suggestion only. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip.

© 2000 Microchip Technology Incorporated. All rights reserved.

The Microchip logo, name, MPLAB, PIC, PICSTART, PRO MATE, and PICmicro are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Microsoft, the Microsoft Internet Explorer, Windows, and MS-DOS are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Netscape is a registered trademark of Netscape Corporation in the United States and/or other countries.

Microwire is a registered trademark of National Semiconductor Corporation.

I 2 C is a trademark of Philips Corporation.

SPI is a registered trademark of Motorola Corporation

All product/company trademarks mentioned herein are the property of their respective companies.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro[®] 8-bit MCUs, KEELOQ[®] code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

第一章 MPLAB[®] IDE 介紹

1-1. MPLAB 介紹

MPLAB IDE 是美國 Microchip Technology Inc. 所發展的一套微處理控制器軟體研發作業平台，它架構於 Windows[®] 的作業環境；對工程師而言，MPLAB IDE 展現出優越的多視窗的除錯環境協助工程師迅速找出程式錯誤之所在。

MPLAB IDE 是一套整合式的發展平台，在 MPLAB 的架構下它提供多視窗文字編輯功能工程師可直接在 MPLAB 環境下直接撰寫、修改原始程式(Source Code)，同時工程師也可以直接啟動語言工具(Language Tools)進行 C 的編譯(Compiler)或組譯(Assembler) 工作，產生微處理機可執行的 HEX 檔(INTEL's HEX Format)。工程師可依實際的除錯環境來選擇使用何種硬體模擬或軟體模擬方式來除錯(如 圖 1-1 所示)，最後工程師可直接啟動 MPLAB 的燒錄功能來燒錄 PICmicro，完成此設計案；以上為研發微處理架構的產品所不可欠缺的步驟，透過 MPLAB 的整合從最初的程式撰寫到最終的燒錄都可一氣呵成無須跳出 MPLAB 的視窗。



圖 1-1 MPLAB 整合式的發展平台

1-2. MPALB 基本內建功能

一、程式編輯功能：

MPLAB 內建一功能強大的程式編輯功能，在撰寫程式時最多可分割成四個編輯視窗以方便設計師查尋各種定義、副程式名稱、函數 等；此外使用者也可以定義文件內各個字義的顏色，例如定出指令、常數、十進制數值、註解 等，讓設計師一目了然 (如圖 1-2)所示。

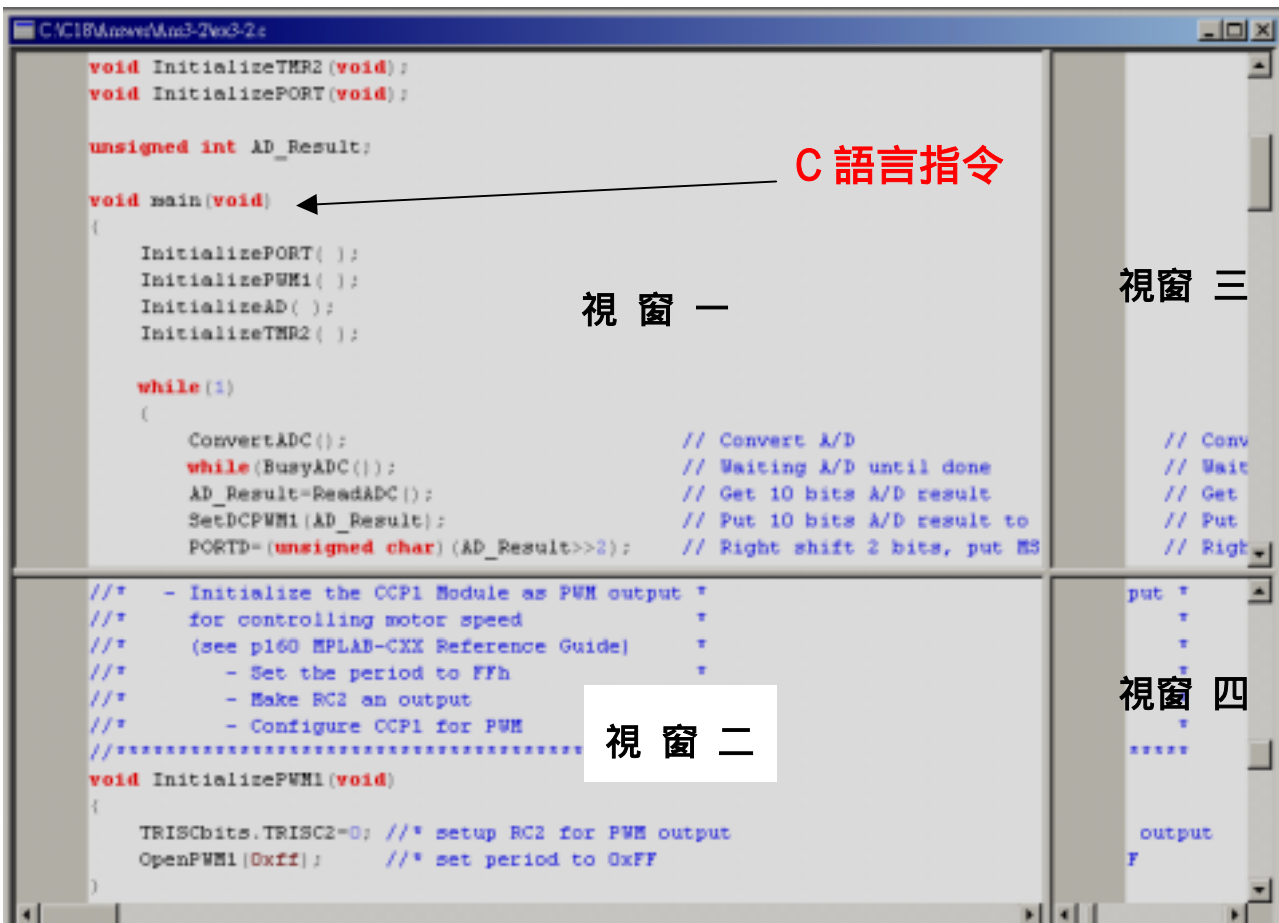


圖 1-2 MPLAB 程式編輯功能

二、專案管理模式：

MPLAB 每發展一個案子之前均須建立該案件的設定模式，此種模式的建立稱之為專案管理(Project Management)。一個 Project 的建立是很簡單的

但是一個 Project 卻記錄眾多的資訊 例如：

- 視窗個數、位置、大小、顏色
- 相關檔案的名稱、位置、工作路徑
- 偵錯訊息的設定值
- 組譯器、編譯器的選擇與設定

藉由專案管理的設定設計師可輕易設定其除錯模式，並在每次啟動 MPLAB 時均能回到上次的除錯環境，即使日後進程式修改也一樣簡易而不用擔心是否漏掉了那些設定。

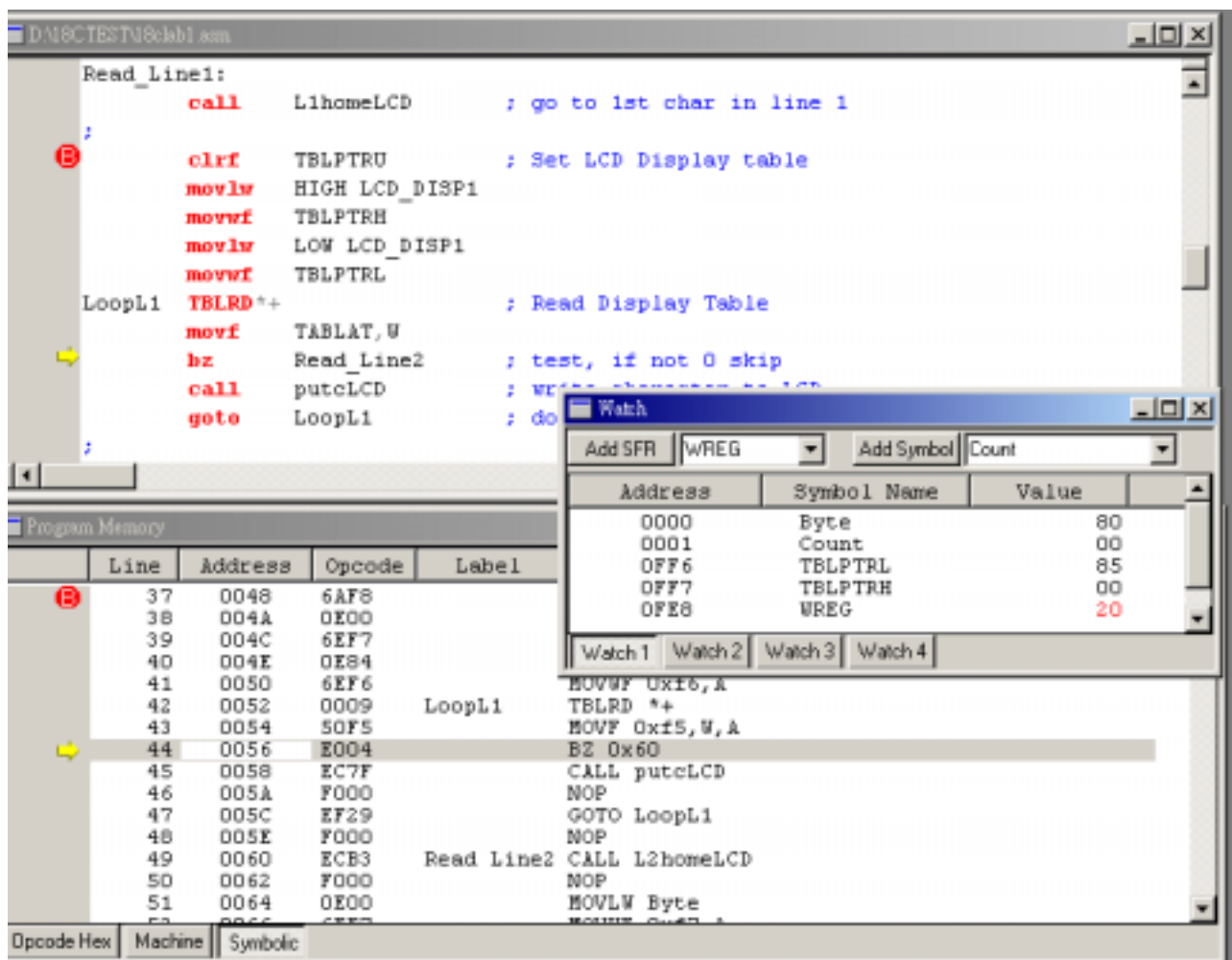


圖 1-3 原始程式除錯功能畫面

三、 原始程式除錯模式：

不管你是用 C 語言或是使用組合語言來撰寫程式，在除錯的過程中 MPLAB 都能以最佳的型態來呈現：原始程式除錯方式，讓您不必再辛苦的使用簡單的 Symbolic Debugger 來除錯。參考圖 1-3 你將會發現從此除錯將不再是看圖猜故事，你的程式是什麼就給你什麼樣的除錯方式。

1-3、 MPLAB 的基本外掛工具

MPLAB IDE 是一套 32 位元的視窗軟體，採開放式的架構可附加各種開發工具；換言之，MPLAB IDE 除了可使用 Microchip 自行研發的軟體工具外，它也可以外接由協力廠商所研發的工具。

一、 語言工具：

目前 MPLAB IDE 本身內建組合語言的組譯器(MPASM) 連結器(MPLINK) 以及資料庫管理(MPLIB)，對撰寫組合語言已足夠使用。對於高階語言的需求則需另外安裝 C 編譯器。有關詳細的語言工具如下所示，使用者可在安裝完 MPLAB v6.xx 後可在”Program Files\MPLAB IDE\MCHIP Tools\”的目錄下看到這些工具軟體：

組合語言組譯器	MPASMWIN.EXE
連結器	MPLINK.EXE
資料庫管理器	MPLIB.EXE
COFF 檔轉 COD	MP2COD.EXE
COFF 檔轉 HEX	MP2HEX.EXE
C 編譯器	MPLAB-C18 (需另行安裝)

二、 軟體模擬：

MPLAB IDE 本身已內建 PICmicro 的軟體模擬器稱為”MPLAB-SIM ”。MPLAB-SIM 為一功能強大的軟體模擬工具，它除了可追蹤程式的執行外

對 I/O 也可進行模擬輸出、入功能及暫存器檔案模擬輸入，除此之外它還提供了計時馬錶功能(Stop Watch)來協助工程師計算程式執行的時間。

三、硬體模擬：

MPLAB v6.10 支援 MPLAB-ICE2000 / ICE4000 及 MPLAB-ICD2，如果你使用的是舊有的 PICMASTER 或 MPLAB-ICD 則只有舊版的 MPLAB v5.xx 可供使用。

硬體設備	MPLAB v6.xx	MPLAB v5.xx
MPLAB-ICE4000	Yes	No
MPLAB-ICE2000	Yes	Yes
PICMASTER	No	Yes
ICEPIC	No	Yes
MPLAB-ICD	No	Yes
MPLAB-ICD2	Yes	Yes
PROMATE	Yes	Yes
PROMATE-II	Yes	Yes
PICSTART Plus	Yes	Yes

表 1-1 MPLAB 硬體支援一覽表

四、程式燒錄器：

MPLAB IDE 目前只支援兩種由 Microchip 自家研發的燒錄器，PROMATE-II 是屬於量產型燒錄器，可更換燒錄腳座以符合各種包裝之需求；PICSTART Plus 為一試用型燒錄器，主要是簡單、價格低適合用來作為研發測試時使用，Microchip 並不建議客戶使用 PICSTART Plus 做量產時的燒錄器使用。

1-4、何處可取得 MPLAB IDE 的軟體

在 Microchip 所出版的技術光碟裡都有 MPLAB 的軟體，使用者可直接從光碟

片的第二片載入。也許光碟片的版本可能是舊版本，最新版的 MPLAB 可在網站上取得。請瀏覽 www.microchip.com 你將發現它有很多知識與資料等您去發掘。

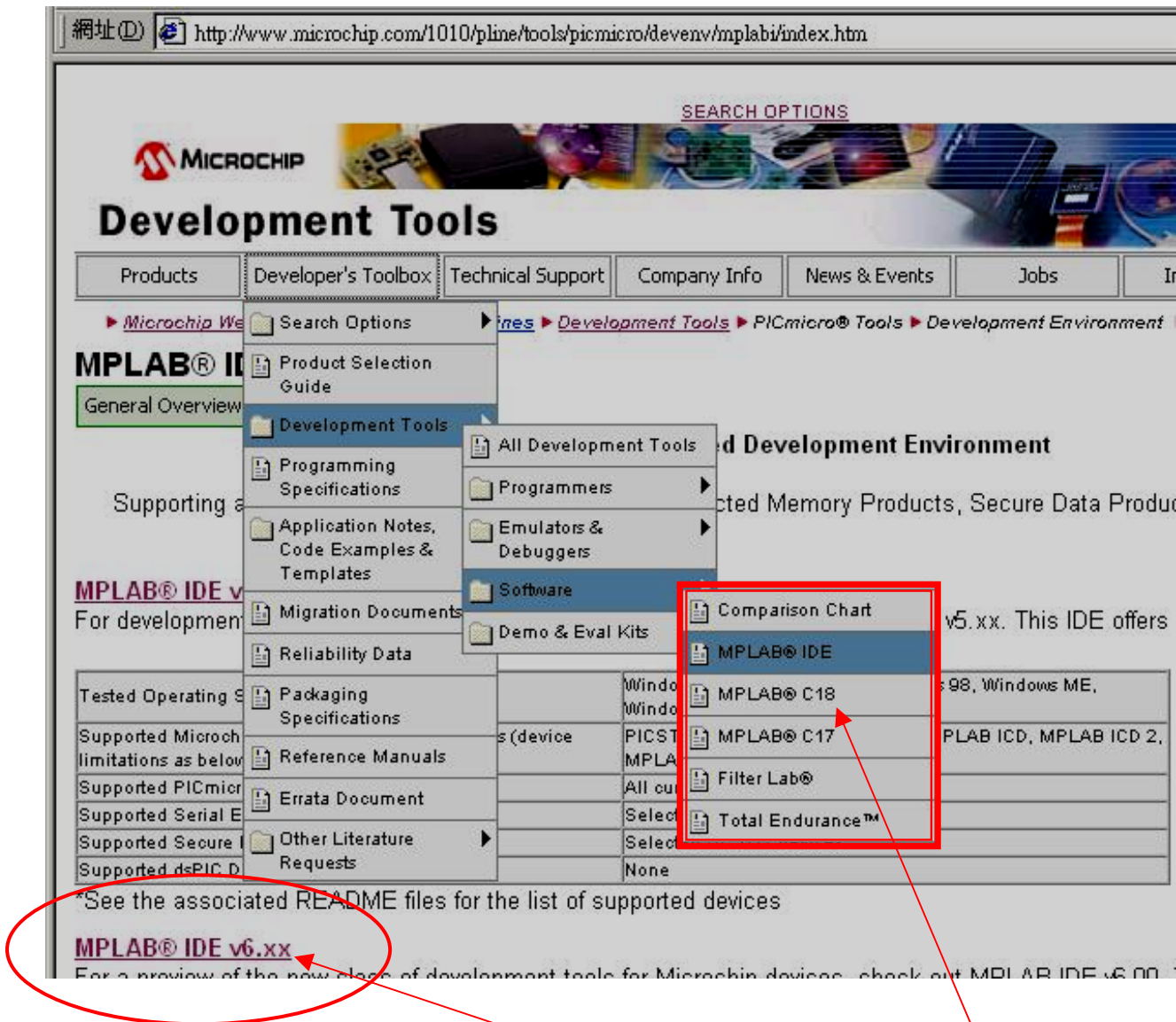


圖 1-4 MPLAB 軟體在網站上位置

由圖 1-4 所示 MPLAB 相關應用軟體，例如：MPLAB IDE 及 C 編譯器均可直接自 Microchip 的網站或 Technical Library CD-ROM 取得。讀者如需索取最新版的 CD-ROM 可與當地的代理商或 Microchip 的當地分公司聯絡。

2-1、系統的基本需求

安裝 MPLAB v6.xx 之前請檢查你的電腦系統，其最小需求如下所示：

- PC 相容電腦- Pentium[®] 以上之等級
- 微軟的 Windows[®] 98/2000 SP2/NT[®] SP6/ME/XP
- 45MB 的硬碟空間
- Internet 瀏覽器 V5.0 或更新版本
- 硬體介面：USB, Printer Port, RS-232

[Microchip WebSite](#) ▶ [Microchip Product Lines](#) ▶ [Development Tools](#) ▶ [PICmicro[®] Tools](#) ▶ [Development Environment](#)

MPLAB[®] IDE

General Overview

Microchip's Integrated Development Environment

Supporting all of Microchip's PICmicro[®] devices and selected Memory Products, Secure Data Products

MPLAB[®] IDE v5.xx
For development and production support of all supported devices, use MPLAB IDE v5.xx. This IDE offers

Tested Operating Systems	Windows [®] 3.1, Windows 95, Windows 98, Windows ME, Windows NT [®] 4 SP6, Windows 2000
Supported Microchip Development Systems Tools (device limitations as below)	PICSTART [®] Plus, PRO MATE [®] II, MPLAB ICD, MPLAB ICD 2, MPLAB ICE 2000
Supported PICmicro Devices	All currently released devices*
Supported Serial EEPROM's	Selected 24 and 93 series devices*
Supported Secure Data Products	Selected HCSxxx devices*
Supported dsPIC Devices	None

*See the associated README files for the list of supported devices

MPLAB[®] IDE v6.xx
For a preview of the new class of development tools for Microchip devices, check out MPLAB IDE v6.00. the following:

Tested Operating Systems	Windows 98, Windows 2000
Supported Microchip Development Systems Tools (device limitations as below)	PICSTART Plus, PRO MATE II, MPLAB ICD 2, MPLAB ICE 2000
Supported PICmicro Devices	PIC18CXXX, PIC18FXXX devices only
Supported Serial EEPROM's	None

圖 2-1 Download MPLAB v6.xx

2-2、開始安裝 MPLAB IDE v6.10

為確保安裝的是 MPLAB IDE 最新檔案，安裝之前請勿執行舊版的 MPLAB 以免產生常駐程式相互衝突現象。若先前已執行過 MPLAB，建議重新開機後再進行安裝，如此可避免版本的混淆。

新開機後再進行安裝，如此可避免版本的混淆。

一、 MPLAB v6.10 可從網站或 CD-ROM 載入壓縮檔案(如圖 2-1 所示)，原始畫面共有兩種 MPLAB 可使用，分別是 MPLAB v5.xx 及 MPLAB v6.xx，到底要使用何種版本來開發程式呢？底下提供一些選擇方向供工程師們參考：

■ 使用 MPLAB v6.10 的場合

1. 你想使用新的 MPLAB IDE Win-32 的環境
2. 使用 MPLAB-ICD2 + USB 介面為開發工具
3. 打算使用新一代 16 位元架構 dsPIC 的編譯工具及軟體模擬
4. 使用 MPLAB C18 為開發工具

■ 建議繼續使用 MPLAB v5.xx 的場合

1. 目前 MPLAB v6.10 只支援 PIC18，PIC16Cxxx 及 dsPIC 系列。如果你是開發 PIC16C5x 或 PIC17Cxxx 系列等則可繼續使用 MPLAB v5.xx。目前 Microchip 全系列的 PICmicro 會在新版 MPLAB v6.20 版本時支援。
2. 如果你是繼續使用 PICMASTER[®] 及 ICEPIC[™] 的發展系統
3. 目前使用 ICD

二、 如選用的是內定的檔案路徑則為：“C:\Program Files\MPLAB IDE\”
安裝完畢後其詳細的檔案內容如右圖所示

- MCHIP_Tools 存有組譯器、連結器及資料庫管理程式等及一些組合語言相關的定義檔(*.inc)
- Drivers2000 為 MPLAB IDE 外掛硬體所需的 Windows 2000 驅動

軟體，除 ICD2 會自動安裝外，其於的 ICE2000 及 ICD4000 的驅動程式均需自行安裝：

- Dbinst2k.htm 安裝 ICD2000 驅動程式說明
- Dbinst4k2k.htm 安裝 ICD4000 USB 驅動程式說明

- ICD2 則是一些有關 ICD2 要使用的除錯 BIOS
- Dlls 為 MPLAB 執行檔的位置，mplab.exe 就是在此目錄底下

名稱	大小	類型
Device		檔案資料夾
dlls		檔案資料夾
Drivers2000		檔案資料夾
dsPIC_Tools		檔案資料夾
HelpFiles		檔案資料夾
ICD2		檔案資料夾
MCHIP_Tools		檔案資料夾
READMEs		檔案資料夾
system		檔案資料夾
INSTALL.LOG	34 KB	小作家文件
UNWISE32.EXE	146 KB	應用程式

2-3、安裝 MPLAB-C18

Microchip 所提供的 C Compiler 為一標準架構的 ANSI C (ANSI-89 Specification)，同樣的它也可以自網站取得(如下圖所示)：

<http://www.microchip.com/1010/pline/tools/picmicro/code/mplab18/index.htm>

MPLAB -C18 C Compiler Development Additional Documentation				
Title	Date:	Size	D/L	View
MPLAB® C18 C Compiler User's Guide	12/2/02	(1.47 Mb)		
MPLAB® v2.09.24(Beta of version 2.10)	10/28/02	(7.36 Mb)		
MPLAB® C18 C Compiler Getting Started	11/14/02	(1.26 Mb)		
MPLAB® CXX User's and Reference Guide Addendum	6/14/02	(446 K)		
MPLAB® C18 C Compiler Development Tools Overview	6/18/01	(60 K)		
MPLAB® C18 ReadMe	1/16/02	(16 K)		
MPLAB® C18 Demo v2.00	6/14/02	(7.02 Mb)		
MPLAB® C18 Upgrade v1.10	7/10/02	(2.40 Mb)		
MPLAB® C18 Upgrade v2.10	11/25/02	(6.97 Mb)		
MPLAB® C18 Optimization Tips	12/6/00	(104 K)		
MPLAB® CXX Reference Guide	11/20/01	(1.47 Mb)		
MPLAB® CXX Quick Reference Card	3/7/00	(49 K)		

圖 2-2 MPLAB-C18 在網站的支援

圖中可看出各個版本的支援，同時也包含了 MPLAB C18 的使用手冊及函數庫使用參考手冊。在軟體的版本上有分 Demo 及 Upgrade 兩種，Demo 為全功能試用版對程式編譯的大小及選擇 PIC18 的 IC 都沒有限制，唯一的限制是它有 30 天的使用期限。Upgrade 版本是屬正式版的昇級套件，使用者必須擁有正式的版本不管是 v.1.00、v1.10、v1.10.02 或是 v2.00 的正式版都可升級為 v2.10 的版本。本使用手冊會以 MPLAB-C18 v2.10 為範例，使用者可直接下載 v2.10 的 Demo 版或 Beta v2.10 的 Demo 版使用。

MPLAB-C18 安裝完畢，如是選用內定的檔案路徑則為：C:\MCC18\

- bin 編譯器執行檔目錄
- doc 使用手冊目錄
- h 各種標準 include 檔
- lib 標準函數庫目錄
- lkr 檔案連結描述檔目錄
- src 標準函數庫原始程式的目錄區

名稱	大小	類型
bin		檔案資料夾
c++		檔案資料夾
doc		檔案資料夾
example		檔案資料夾
h		檔案資料夾
lib		檔案資料夾
lkr		檔案資料夾
popt-1.5		檔案資料夾
src		檔案資料夾
install.log	213 KB	小作家文件
Readme.c18	16 KB	C18 檔案
UNWISE.EXE	126 KB	應用程式

圖 2-3 MPLAB-C18 的目錄

3-1、啟動 MPLAB IDE

在第一章時我們曾說過 MPLAB 是採用專案管理 (Project Management) 的方式為設計案做最方便的服務。所以在使用 MPLAB 時我們就必須記住設定 Project 的方法與步驟。下圖為建立一個 Project 所需的步驟，希望能透過此圖讓使用者能快速且正確的建立一個 Project：

步驟	動作選項	功能
1	Configure→Select Device	選擇要使用的 IC 然後按 OK
2	Configure→Configuration Bits	設定 IC 的工作的模式
3	File→New & File→Save As	建立你的原始程式檔案後存檔
4	Project→New	建立 Project 名稱及其路徑
5	Project Window 加入 Source file & lkr	加入原始程式及連結描述檔
6	Project →Set Language Toolsuite	選擇語言工具(dsPIC 或 PICmicro)
7	Project→Set Language Tool Locations	設定語言工具的執行路徑
8	Project→Build Options→Project	設定資料庫、含入檔及連結描述檔的路徑
9	Project Save & Build All	儲存 Project 的設定與開始編譯程式

表 3-1 建立一個 Project 的步驟

由表 3-1 所示，大致可了解設定 Project 的步驟，乍看來似乎有點煩雜其實一個真正的案子在開發時不是三五天就完成了，這個 Project 的設定只要設一次其後你只要用開啟 Project 即可，除非你又有新案子要開發。接下來就要開始啟動 MPLAB IDE 了，啟動 MPLAB IDE 可在 Windows 目錄下執行”[開始>程式集>Microchip MPLAB IDE>MPLAB IDE](#)”

MPLAB IDE 的主畫面如下所示：

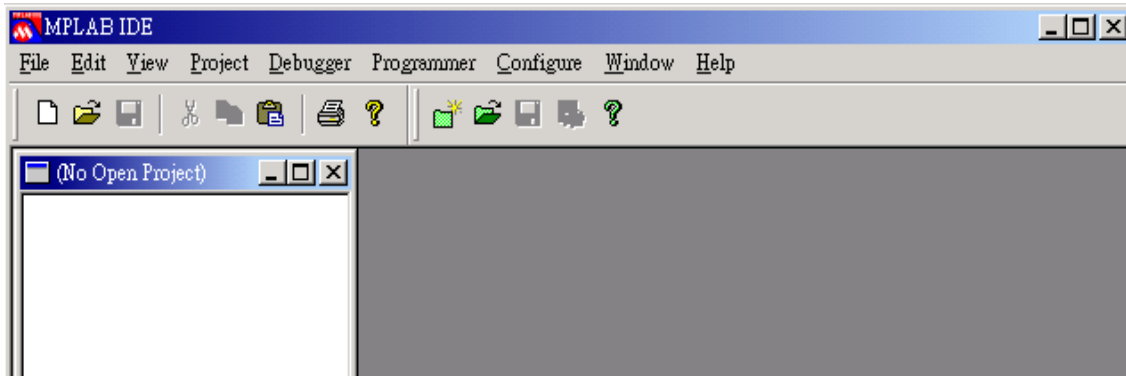


圖 3-1 MPLAB IDE 的主畫面

3-2、 第一步：選擇要使用的 PICmicro [Configure → Select Device]

在使用 MPLAB 的第一步驟就是要選擇所要的 PICmicro，在主目錄點選”Configure → Select Device”進入圖 3-2 畫面，目前 MPLAB IDE v6.xx 可支援 PIC18 所有系列但 PIC16 系列將會在版本 v6.10 後支援，在選定元件後同時也會顯示該元件目前發展工具支援情況。

例如：PIC18F452 目前有 MPLAB SIM、ICD2、ICD2000、PICSTART Plus 及 PROMATE-II 等開發工具可供使用(綠燈所示)，但則紅燈所示不支援。

選擇 OK 後注意主畫面底線元件的選擇應會改為 PIC18F452。



圖 3-2 步驟一、選擇元件

3-3. 第二步：設定 PICmicro 工作模式 [Configure → Configuration Bits]

PICmicro 選好了，接著就要設定這個 單晶片微控制器 的工作模式，一個 單晶片微控制器 要讓它正常工作不只是把程式燒進去就好了，你還要為它的硬體環境做適當的設定才行。如何了解這些設定值得意義是非常重要的，使用著應詳細閱讀相關的資料手冊以了解各項設定值的意義。

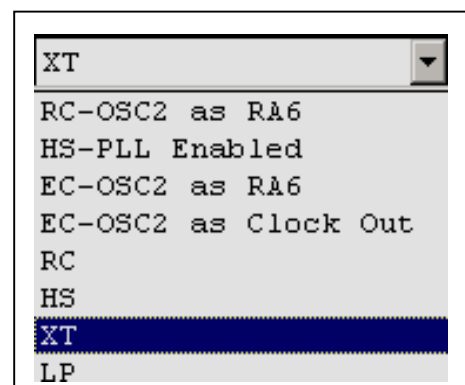
以 PIC18F452 為例，圖 3-3 為其設定視窗，最簡單的設定就是將所有選項設為“Disable”的選項包含 Code Protect 及 Table Read/Table Write，只選擇振盪器工作的模式(如果是使用 ICD2 為除錯工具時就必須將 Background Debug 模式 Enable)。

Address	Value	Category	Setting
300001	F9	Oscillator	XT
		Osc. Switch Enable	Disabled
300002	F5	Power Up Timer	Disabled
		Brown Out Detect	Disabled
		Brown Out Voltage	4.2V
300003	FE	Watchdog Timer	Disabled
		Watchdog Postscaler	1:128
300005	FF	CCP2 Mux	RC1
300006	7A	Low Voltage Program	Disabled
		Background Debug	Enabled

圖 3-3 步驟二、PIC18F452 的設定工作模式

一、 Oscillator：振盪器工作模式選擇

- RC-OSC as RA6：以外接 RC 為振盪器元件，RA6 為一標準的 I/O 腳功能
- HS-PLL Enable：外接振盪器或石英晶體為振盪元件，並啟動內部 4 倍頻電路為振盪來源，4 倍頻 PLL 電路只有在此模式下才會動作
- EC-OSC2 as RA6：採用外部振盪器從 OCS1 腳輸入振盪源，RA6 為一標準的 I/O 腳功能



- EC-OSC2 as Clock Out：採用外部振盪從 OCS1 腳輸入振盪源，RA6 變為 OCS2 的振盪頻率除 4 後的輸出腳($F_{osc}/4$)
- RC：RC 振盪模式(OSC2 輸出為 $F_{osc}/4$)
- HS：外接較高頻率石英振盪器(>4MHz~20MHz)或一般的陶瓷振盪器
- XT：外接石英振盪器(>200KHz~4MHz)
- LP：外接低頻率石英振盪器(32KHz~200KHz)

如果要讓 18F452 工作在 40MHz 的速度只有用 10MHz 的石英晶體及選用 HS-PLL Enable 模式方可達到，18F452 是無法直接振在 40MHz 的。

二、 Oscillator Switch Enable：允許切換主振盪器或 Timer1 OSC 為系統的主頻率的控制位元

- 如果設為 Enable 時，可經軟體切換 *OSCCON* 暫存器的 *SCS* bit 來選擇要工作在主振盪器頻率(高速模式)或 Timer1 OSC (低速模式)已達省電的目的。

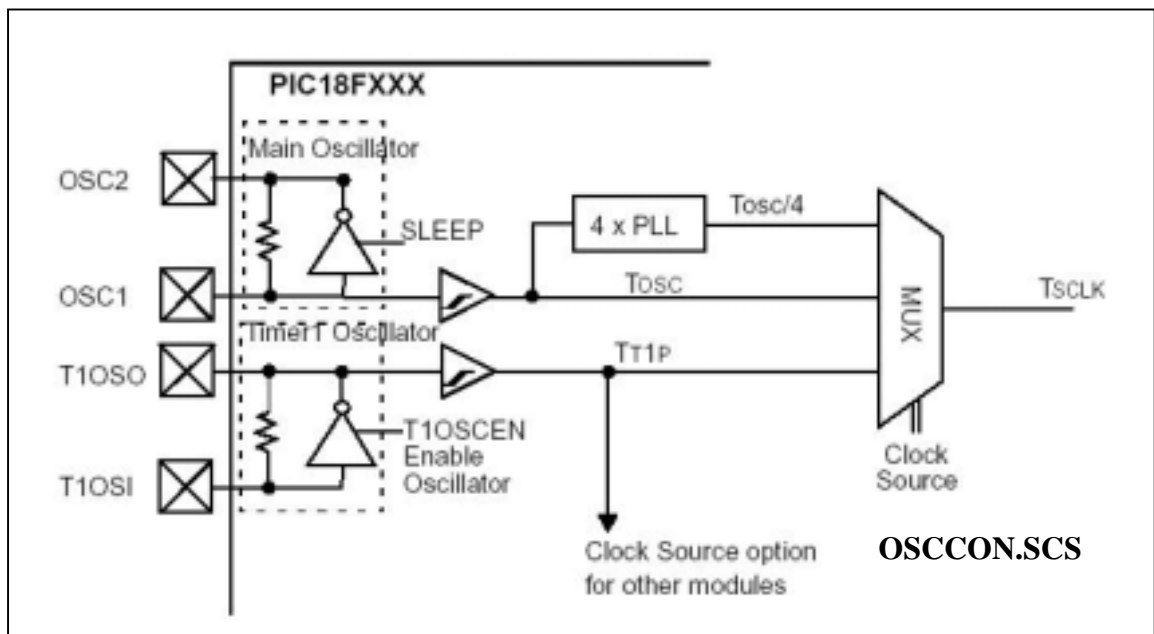


圖 3-4 PIC18F452 的工作頻率選擇模式

三、 Power Up Timer : 72 mS 開機延遲

- Enable : PWRTE=0 , 開機 Reset 後及 Brown-out 被偵測到時 , PICmicro 會自動延遲 72mS 後在啟動程式以避免電源的不穩定狀態或電源電壓開機時上升速度較慢所造成的當機現象。
- Disable : PWRTE=1 , 關閉 72mS 的自動延遲功能。

Oscillator Configuration	Power-up ⁽²⁾		Brown-out	Wake-up from SLEEP or Oscillator Switch
	$\overline{\text{PWRTE}} = 0$	$\overline{\text{PWRTE}} = 1$		
HS with PLL enabled ⁽¹⁾	72 ms + 1024Tosc + 2ms	1024Tosc + 2 ms	72 ms ⁽²⁾ + 1024Tosc + 2 ms	1024Tosc + 2 ms
HS, XT, LP	72 ms + 1024Tosc	1024Tosc	72 ms ⁽²⁾ + 1024Tosc	1024Tosc
EC	72 ms	—	72 ms ⁽²⁾	—
External RC	72 ms	—	72 ms ⁽²⁾	—

Note 1: 2 ms is the nominal time required for the 4x PLL to lock.
2: 72 ms is the nominal power-up timer delay, if implemented.

圖 3-5 PIC18F452 的開機延遲示意圖

四、 Brown Out Detect : 低工作電壓重置設定

正確的說應該是 Brown Out Reset(BOR) , 因為工作電壓一但低於此電壓 , PICmicro 即會進入 Reset 模式。

注意 : Enable 此 BOR 選項 , 必須同時 Enable “Power Up Timer (72 mS 開機延遲)” 的選項。此外 , PIC18Fxxxx 本身都有 “開機重置偵測裝置(Vcc Power-On Reset)” 其偵測電壓約為 0.7V , 記住這個 V_{POR} 與 BOR 是獨立分開的電路不要混淆在一起。

- Enable : 啟動工作電壓偵測重置模式並開啟 72mS 自動延遲功能 , 需注意所使用的 Vcc 及選擇 BOR 的 Reset 電壓值
- Disablel : 關閉工作電壓偵測重置模式

五、 Brown Out Voltage : Reset 電壓偵測點

基本上在 PIC18 的系列尚可提供四個電壓偵測點：2.0V，2.7V，4.2V，4.5V；使用者可依據所使用的 Vcc 來選擇。

注意：如果使用 PIC18LFxxx 的 IC 工作在 2V 的電壓時，就必須將 BOR 關掉使用內建的 Power-On Reset

六、 Watchdog Timer：溢時監測計時器(簡稱 WDT)

- Enable：啟動 WDT 功能
 1. 在一般工作模式下 WDT 發生溢時後會重新 Reset PICmicro
 2. 如在睡眠模式下 WDT 發生溢時後會執行 SLEEP 的下一個指令
- Disable：關閉 WDT 功能

七、 Watchdog Postscaler：溢時監測計時預除器

WDT 的基本時間是 18mS (最小為 7mS 到最大為 33mS)，以此時間可利用預除器來擴展時倍數，最大延遲時間為 $18\text{mS} * (1:128) = 2.3 \text{ 秒}$ ，但設計時不可用此值來估算，我們必須考慮最小的延遲時間 $7\text{mS} * (1:128) = 896\text{mS}$ ，以避免誤動作。

符號	說 明	Min	Typ	Max	時間	條 件
T _{WDT}	Watchdog Timer Time-out Period (No Postscaler)	7	18	33	mS	
T _{PWRT}	Power up Timer Period	28	72	132	mS	
T _{BOR}	Brown-Out Reset Pulse Width	200	--	--	mS	V _{cc} < V _{BOD}

圖 3-6 PIC18F452 的 WDT 延遲誤差表

八、 CCP2 Mux：設定 CCP2 功能腳位置

一般而言 CCP2 (Compare/Capture/PWM) 是設定在 RC1 的腳位上 (PIC16Cxx) 但如果要使用 CCP2 及外接 Timer1 石英振盪器時就會出現無法兼顧的問題，故在 PIC18 系列就可更改 CCP2 到 RB3 的腳位。

九、 Background Debug：啟動內部除錯功能(ICD2 專用功能)

ICD2 專屬功能，如使用 ICD2 作為除錯工具的話就必須將此功能 Enable。

十、 Stack Overflow Reset：32 層堆疊溢位重置

堆疊產生溢位情形重置 PICmicro，通常發生此情形以軟體錯誤居多也有可能受某種干擾程式進入了未知的程式段而造成堆疊錯誤，此時可重置 PICmicro 重新執行程式。

十一、 Code Protection：程式記憶體保護設定

十二、 EEPROM Protection：EEPROM 資料記憶體保護設定

基本上在開發的過程中不要將 IC 保護起來，等程式完成後再依實際需要來啟動保護設定。



3-5、第三步：建立組合語言原始檔案 [File→New & File→Save As]

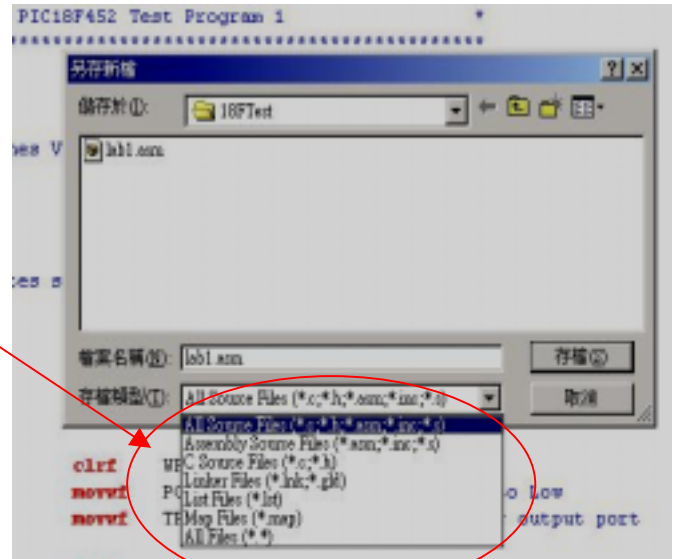
在 MPLAB IDE 組畫面下點選“file →New”以建立一個新的文字檔，並輸入下列組合語言程式：

```
.*****
;*          PIC18F452 Test Program 1          *
.*****
;
;          list p = 18f452
;          #include <p18f452.inc>
;
; Defines Variable
;
Del_uS      equ      0x00
Del_mS      equ      0x01
;
; Locates startup code @ the reset vector
;
;          org      0x00
;          goto     Start
;
;          org      0x20
Start
;          clrf     WREG
;          movwf    PORTC      ; Clear PORTC to Low
;          movwf    TRISC      ; Set PORTC for output port
;
Loop_Inc    call     Sub_Delay      ; Delay 100mS
;          incf     PORTC, F
;          goto     Loop_Inc
;
; Delay 100 mS Subroutine (4MHz Oscillator)
;
Sub_Delay
;          movlw   .200          ; Delay 100mS
;          movwf   Del_mS
;
D_Loop
;          movlw   .166          ; Delay 500uS
;          movwf   Del_uS
;          decfsz  Del_uS, F
;          Goto    D_1
;
;          decfsz  Del_mS, F
;          goto   D_Loop
;          return
;
;          end
```

輸入完畢請用”File → Save As”儲存這個檔案。

在此範例是存入在”18Ftest”的目錄下，檔案名稱為”Lab1.asm”。請注意在 MPLAB 儲存檔案的型態有很多種類，應正確使用之。

此例是以 .asm 組合語言型態儲存。



MPLAB 所提供的本文編輯器足以應付編寫程式的需求，底下就一些常用的編輯功能說明之：由下列兩圖表可大致看出此編輯器的基本使用方法和 Windows 的編輯器小作家類似使用上應很容易上手。

游標控制

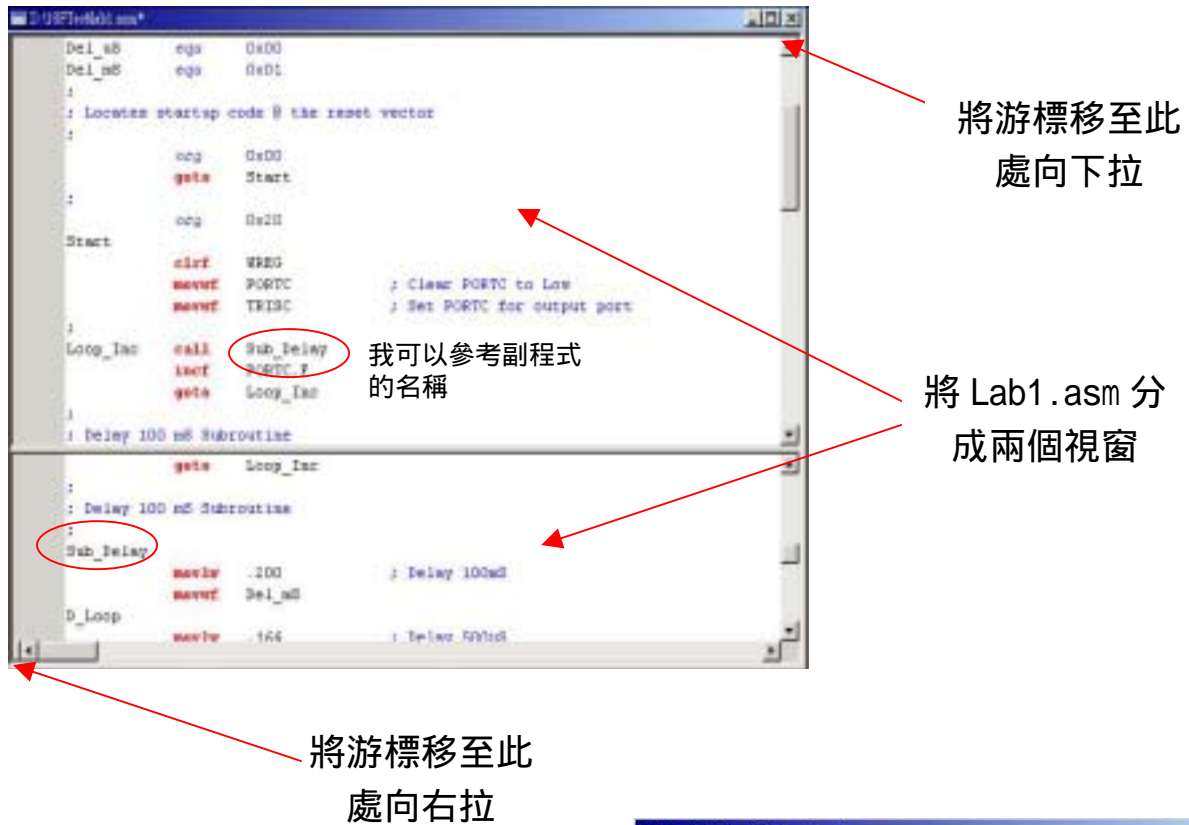
- ✍ Home - 游標移至本行起頭
- ✍ End - 游標移至本行尾
- ✍ Ctrl + Left - 向左移一個字
- ✍ Ctrl + Right - 向右移一個字
- ✍ Ctrl + Pg Up - 移至本頁起頭
- ✍ Ctrl + Pg Dn - 移至本頁尾
- ✍ Ctrl + Home - 移至本檔案起頭
- ✍ Ctrl + End - 移至本檔案尾

編輯控制

- ✍ Ctrl + Z - 回到上一次的輸入原狀
- ✍ Ctrl + Y - 恢復本次的輸入原狀
- ✍ Ctrl + C - 複製選擇
- ✍ Ctrl + V - 貼上複製
- ✍ Ctrl + X - 清除選擇
- ✍ Ctrl + A - 全選
- ✍ Del - 刪除
- ✍ Ctrl + G - 跳至第幾行
- ✍ Ctrl + F - 尋找字元或字串
- ✍ Ctrl + H - 替換所尋找字元或字串
- ✍ F3 - 向下重複尋找字元或字串
- ✍ Shift + F3 - 向上重複尋找字元或字串
- ✍ Mouse 右鍵按兩下 - 選擇該字

在編輯寫程式時如果程式大的時候有時要查詢副程式名稱或函數功能時，往往需不斷的捲動程式以記下其名稱非常不方便。但新版的 MPLAB 編譯器在編輯程式

時就可將程式劃分成四個畫面供設計師查詢其各程式段對寫程式而言可大大的提高書寫的方便性。如何使用此項功能呢？如下圖所示：



新版的 MPLAB v6.xx 在本文檔的編輯中它還會依據程式的型態來改變其字體的顏色，如何設定你所要的顏色呢？在”Edit → Customize Colors”中選出你喜歡的顏色。

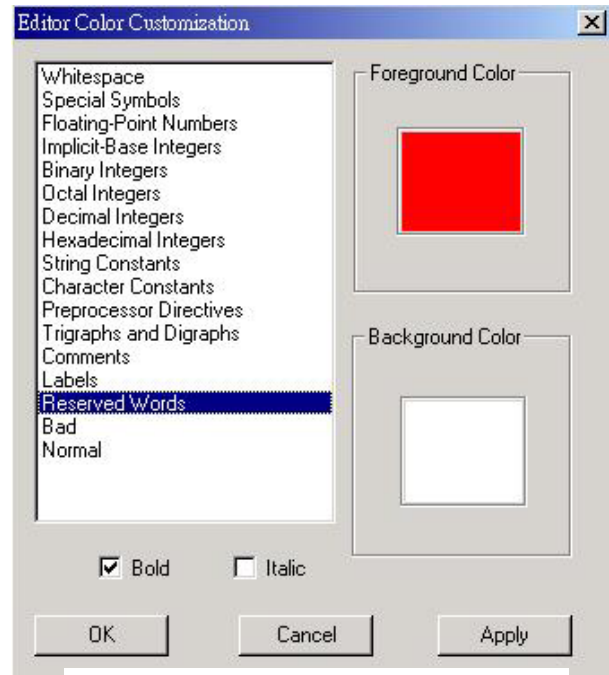
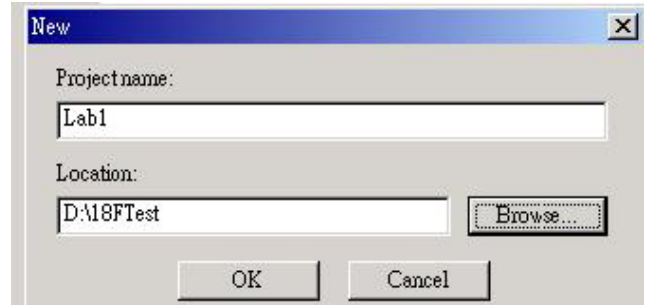


圖 3-8 改變程式編輯的顏色

3-5、 第四步：建立一個 Project [Project→New]

Project 的建立有助於日後設計案的管理，它可讓使用者設定新的 Project 後每次打開 MPLAB 時都能回到該 Project 的原設定，它並且能做多重檔案管理這對寫大程式就相當有助益。

在”Project→New”選項中開啟建立新 Project 的畫面，如右圖所示在



D:\18Ftest 的目錄下建立一個名稱為 Lab1 的 Project。此時會在目錄下建立一個新的檔案其名稱為 LAB1.mcp 這就是你的 Project 檔名。

3-6、 第五步：加入原始程式及連結描述檔

有了 Project 後你必須告訴它，你的檔案有那些？但是你必須考慮到你是採何種方式來寫程式：

- 用傳統單一檔案的組合語言且不可重新定址的方式，不使用 Linker
- 採用多重檔案且可重新定址的組合語言架構方式寫程式，需使用到 Linker 來安排其實際的位址
- 用 C 語言寫程式，那就更需要告訴 Project 有關 C Compiler 的各項相關檔案的目錄以及 C 語言專用的 Linker 描述檔(這點是不可與組合語言的 Linker 描述檔混淆在一起)

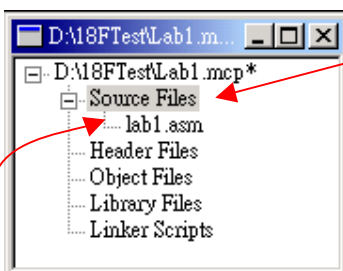
一、 採用單一檔案的組合語言

首先要說明的所謂”單一檔案的組合語言”是指採用直接指定程式執行位址及變數在 RAM 的位址的方式，即使你使用了 include 指令將其他的檔案加進來也是一樣

屬於單一檔案的組合語言的架構，因為在組譯的過程中你的組合語言不會經由 MPLINK.EXE 來產生 HEX 的執行檔而是由 MPASM.EXE 來產生。

在組合語言程式裡使用了”ORG”來定程式位址的及使用”CBLOCK”定 RAM 位址的就一定是”**單一檔案的組合語言**”架構語法。如先前所輸入的程式 Lab1.asm 就是用”**單一檔案的組合語言**”架構語法方式寫的程式。這種語法的優點是簡單、易懂非常適合初學者及較小程式的設計案。

底下就是以 *Lab1.asm* 為例建立一個”**單一檔案的組合語言**”架構的 Project 其名稱為 *Lab1.mcp*



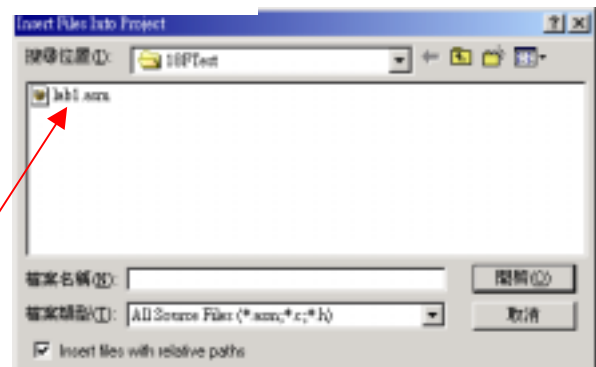
視窗 3-1

將游標移到 Source Files 上，再按老鼠的右鍵就會出現視窗 3-2



視窗 3-2

在視窗 3-2 處利用老鼠點選 Insert Files 則會出現視窗 3-3 以輸入原始組合語言檔案 (Lab1.asm)

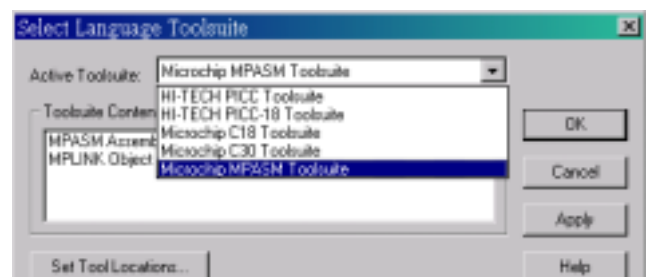


視窗 3-3

在視窗 3-3 加入 Lab1.asm 後，可在 Project 視窗下看到 Lab1.asm

經過以上之步驟，只要在確認語言工具的選擇，這 Project 就設定完成。

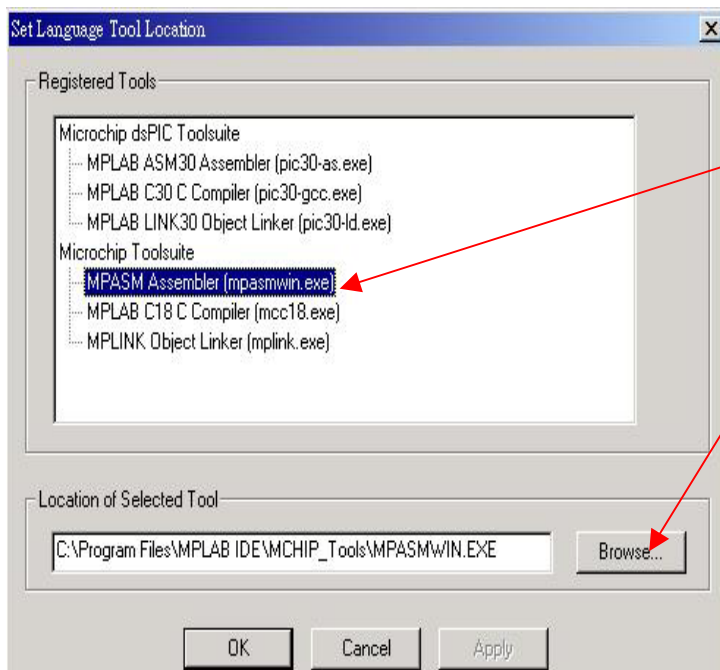
在 MPLABIDE 主目錄下點選”**Project → Set Language Toolsuite**”目前有數種選項 *Microchip C30*、*MPASM*、*C18 Toolsuite* 及 *Hi-Tech C* 等，日後會繼續加入、CCS、IAR 等協力廠商的 C Compiler。選擇 *Microchip*



MPASM Toolsuite 後再點選”Set Tool Locations”以加入各項語言工具的執行路徑。因為是 Lab1.asm 是單一檔案的組合語言架構，所以只要確定 MPASMWIN.EXE 的執行路徑如圖 3-9 所示。先點選 MPASM Assembler 然後將其路徑指到你所安裝 MPLAB IDE 的目錄下，此處是使用內設位置 (Default) *C:\Program Files\MPLAB IDE\MCHIP_Tools\MPASMWIN.EXE* 來設定組合語言的組譯器執行位置。

使用者可分別在設定其工具以方便日後使用，點選 MPLAB C18 Compiler 及 MPLINK Object Linker 分別設定其路徑如下所示：

- MPASM → C:\Program Files\MPLAB IDE\MCHIP Tools\MPASMWIN.EXE
- MPLINK → C:\Program Files\MPLAB IDE\MCHIP Tools\MPLINK.EXE
- MPLAB C18 → C:\MCC18\BIN\MCC18.EXE



利用老鼠點選後用
Browse 選擇其路徑

圖 3-9 設定語言工具執行路徑

至此，Project 的設定已完成，先 Save Project，接下來就可進行組譯 Lab1.asm。在 MPLAB IDE 主畫面下：

- 點選 Project 視窗內的 Lab1.asm 開啟原始程式視窗
- 在主目錄下，*Project* → *Build All* 或直接按 *CTRL+F10* 來執行組譯工作

- 若再組譯有錯誤發生可直接用 mouse 在錯誤的行號上快速按兩下則游標會出現在發生錯誤行的原始程式中，使用者可直接修改後再按 **CTRL+F10** 重新 Build All 直到所有錯誤及警告都消失，如圖 3-10 所示。

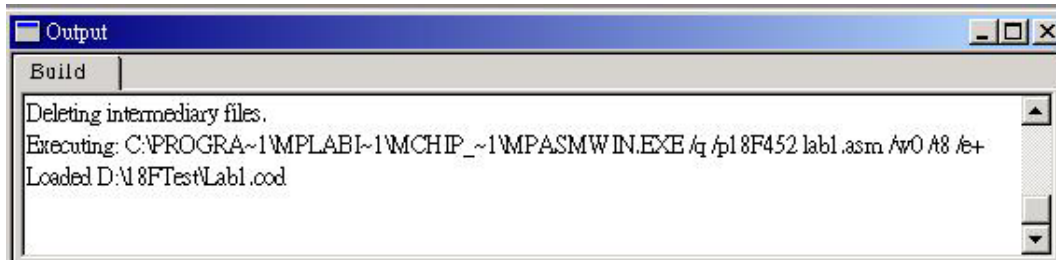


圖 3-10 組譯輸出視窗

二、 多檔案可重新定址的組合語言

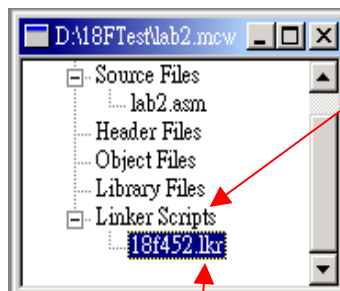
在發展大程式或是方便以模組化來管理程式這時就可能使用多檔案的方式來撰寫組合語言，每個組合語言檔案並不強制規定其位址而由 Linker 統籌決定。當然 Linker 也不是隨便就塞個位址給程式它還是要參考描述檔的指定去做設定的工作，而這個描述檔當然是程式設計師所指定的，所以回到頭來 Linker 所安排的位址實際就是程式設計師的意思。有關這個描述檔的內容 Microchip 已經寫好，程式設計師只要依實際需要修改即可，不過經驗告訴我們除非是有特殊的位址安排一般而言是不用去改 Microchip 所提供的描述檔的。

在可重新定位的架構下，在組合語言裡是看不到 `ORG`，`CBLOCK` 的指令，取而代之的是 `CODE`，`UDATA` 等指令，有興趣的讀者可參考“MPLAM User’s Guide with MPLINK and MPLIB”這本書。底下就是將 `Lab1.asm` 改為可重新定位的架構的組合語言語法。



```
*****
;
; * PIC18F452 Test Program 2 *
; *****
;
; list p = 18f452
; #include <p18f452.inc>
;
; Defines Variable
;
Del_uS equ 0x00
Del_mS equ 0x01
;
; Locates startup code @ the reset vector
;
; code 0x00
; goto Start
;
PROG1 code
Start
;
; clrf WREG
; movwf PORTC ; Clear PORTC to Low
; movwf TRISC ; Set PORTC for output port
;
Loop_Inc call Sub_Delay
; incf PORTC, F
; goto Loop_Inc
;
; Delay 100 mS Subroutine
;
Sub_Delay
; movlw .200 ; Delay 100mS
; movwf Del_mS
;
D_Loop
; movlw .166 ; Delay 500uS
; movwf Del_uS
D_1 decfsz Del_uS, F
; goto D_1
;
; decfsz Del_mS, F
; goto D_Loop
; return
;
end
```

讀者可將 *Lab1.asm* 修改後用 "Save As" 儲存名為 *Lab2.asm*，重新建立一新 Project。重複上述的步驟一到步驟四以建立一個可重新定位的 Project 名稱為 *Lab2.mcp*。底下就是以 *Lab2.asm* 為例子



視窗 3-4

將游標移到 LinkerScripts 上,再按老鼠的右鍵就會出現視窗 3-5



視窗 3-5



視窗 3-6

在視窗 3-5 處利用老鼠點選 Insert Files 則會出現視窗 3-6 以輸入 PIC18F452 的連結描述檔

需注意的是該處使用的是組合語言則必須使用 MPLAB IDE 所提供的連結描述檔，千萬不可撈過界去使用 MPLAB-C18 所提供的連結描述檔，切記。

此處是使用 PIC18F452 所以就要選擇 18F452.lkr，不同的 PICmicro 因記憶大小均不盡相同所以每個 PIC18Fxxxx 都有其連結描述檔，使用時不可混淆。連結描述檔所在的位置一定要有所了解說不定你還要去修改它，記住修改過的連結描述檔請用不同的檔名儲存以免干擾到你以前的程式。

■ 連結描述檔位置：*C:\Program Files\MPLAB IDE\MCHIP_Tools\LKR*

接著我們需設定連結描述檔的路徑，在 *Project → Build Options → Project* 開啟右

邊的設定畫

面，如右圖所示我們輸入了 Linker Script Path 以告訴 MPLINK 何處可找到相關的連結描述檔。

你也可以點選 MPASM 及 MPLINK 做更高階的設定，例如你想改變輸出的 Hex 格式或啟動 MAP，LST 等輸出檔案的設定。

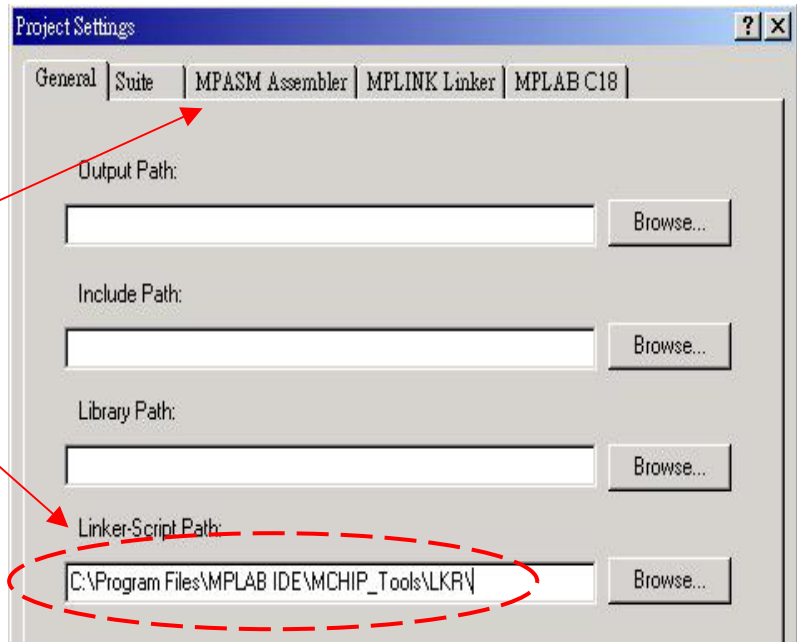


圖 3-11 連結描述檔位置

至此，重新定位的架構的組合語言語法的 Project 設定已完成，試試看 Build All (CTRL +F10)是否可組譯成功，如有錯誤發生請重新參照先前的步驟再設定一次，更詳細的錯誤說明請參考”MPASM User’s Guide (DS33014G)，讀者可在網站上或 CD-ROM 找到它

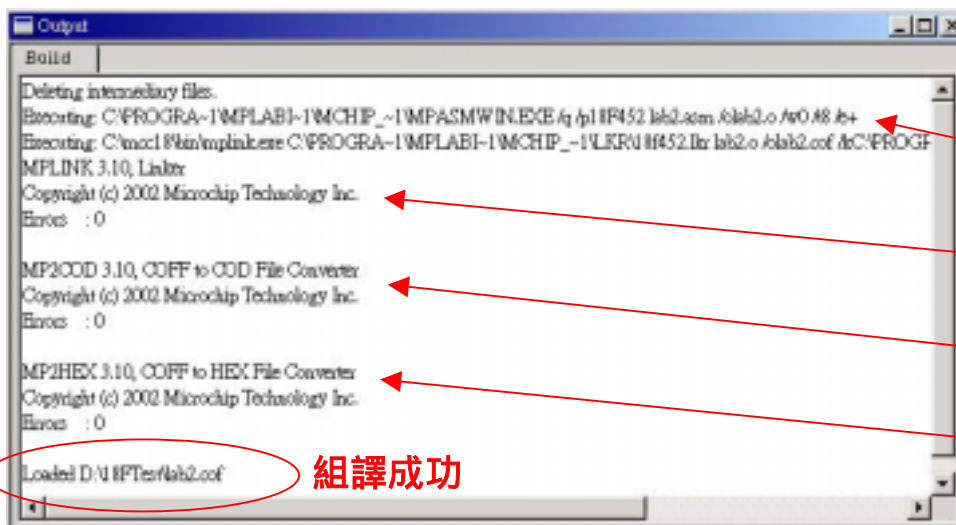


圖 3-12 組譯成功

組譯器

連結器

轉除錯檔

轉 Hex 檔

用 C 語言寫程式

用 C 語言所寫的程式，不僅攜帶性高、平台的相容性也高，最重要的是用 C 來寫大程式時較為容易除錯也較組合語言簡單多了。站在程式設計的效益的立場上我們也建議程式設計師：PIC18Fxxxx 系列的設計案，最好是使用 MPLAB-C18 v2.10 的版本(或更新版本)，由於 Microchip 當初在設計 PIC18 的架構時已經把 C 的語法架構考慮進去，對於指標、陣列、軟體堆疊及參數傳遞都能藉由內部的三個指位器來達到，同時 MPLAB-C18 也會對程式作最佳化的編譯以減少程式碼的大小、提高程式執行的效率。

安裝完 MPLAB-C18 後你會有一個 C:\MCC18 的目錄，如圖 2-3 所示各項子目錄分別存有各個不同的需求。

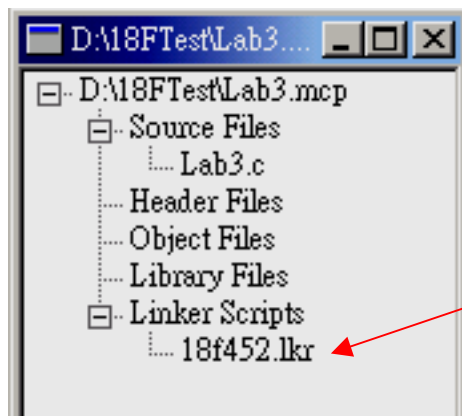
- \bin 編譯器執行檔目錄：內有 MPLAB-C18 的編輯器 MCC18.EXE
- \doc 使用手冊目錄：存有 MPLAB-C18 的資料庫參考手冊及使用手冊，檔案格式為 PDF，可列印參考
- \h 各種標準 include 檔：在這個目錄底下有眾多的 Header Files 再寫 C 語言時必須注意那些 H 檔是必須加入的，尤其是使用到 Microchip 所提供的函數庫
- \lib 標準函數庫目錄：包含了三個 C18 的啟動模組，標準函數庫(clib.lib) 以及所使用的 PICmicro 週邊函數庫
- \lkr 檔案連結描述檔目錄：MPLAB-C18 專用的連結描述檔，不可與組合語言的連結描述檔混淆
- \src 標準函數庫原始程目錄：這個目錄是所有 lib 目錄下資料庫的原始程式，有些是用組合語言寫的、有些是 C 寫的，基本上它是一個很好的程式參考及練功的區域。如果你認為 Microchip 所提供的資料庫需修改以符合你的需要時，可利用此原始程式修改之。

經由以上之說明，我們應已了解檔案相關的位置，同樣的再建立一個 C 的 Project 時，MPLAB IDE 也要知道相關的檔案位置及路徑。按照前面所提的步驟一到步驟四的程序來設定 MPLAB IDE 後在依以下之步驟建立一個 C 語言的 Project

底下是一個 C 語言的範例，試著將程式輸入並儲存檔名為 Lab3.c

```
//*****  
//*      Lab3.C Example C Program  
//*****  
#include <p18f452.h>  
#include <adc.h>  
  
void InitializeAD(void);  
  
void main(void)  
{  
    TRISD=0;                // Set PORTD for output  
    InitializeAD( );        // Initial A/D module  
  
    while(1)  
    {  
        ConvertADC();       // Convert A/D  
        while(BusyADC( ));  // Waiting A/D until done  
        PORTD=ADRESH;       // Put A/D result on PORTD  
    }  
}  
  
//*****  
//* Function: InitializeAD      *  
//*****  
void InitializeAD(void)  
{  
    OpenADC(  ADC_FOSC_8  
            & ADC_LEFT_JUST  
            & ADC_1ANA_OREF,  
            ADC_CHO  
            & ADC_INT_OFF );  
}
```

在 Lab3.c 的程式輸入並存檔完畢後開始建立一個屬於 C 語言的 Project, 重複之前建 Project 的方式在主目錄下點選 ”Project →New” 建立一個名為 Lab3.mcp 的 Project。



此 Project 名稱為 Lab3.mcp, 其 C 原始程式的檔名為 Lab3.C

使用 C 語言設計程式一定要用到 C 語言專用的連結描述檔, 在圖左 18f452.lkr 就是連結描述檔, 注意其所在的檔案目錄是在 C:\mcc18\lkr 裡

在建立了 Lab3.mcp 的 Project 後, 我們還是要檢查語言工具及設定 C Compiler 的各項路徑。在 ”Project →Set Language Tool Locations” 檢查各項語言工具是否如下所示：

- MPASM → C:\Program Files\MPLAB IDE\MCHIP Tools\MPASMWIN.EXE
- MPLINK → C:\Program Files\MPLAB IDE\MCHIP Tools\MPLINK.EXE
- MPLAB C18 → C:\MCC18\BIN\MCC18.EXE

在確認了語言工具後, 接下來請在主目錄下選擇 ”Project →Build Options →Project” 來設定 C 語言的各項路徑如圖 3-12 所示。

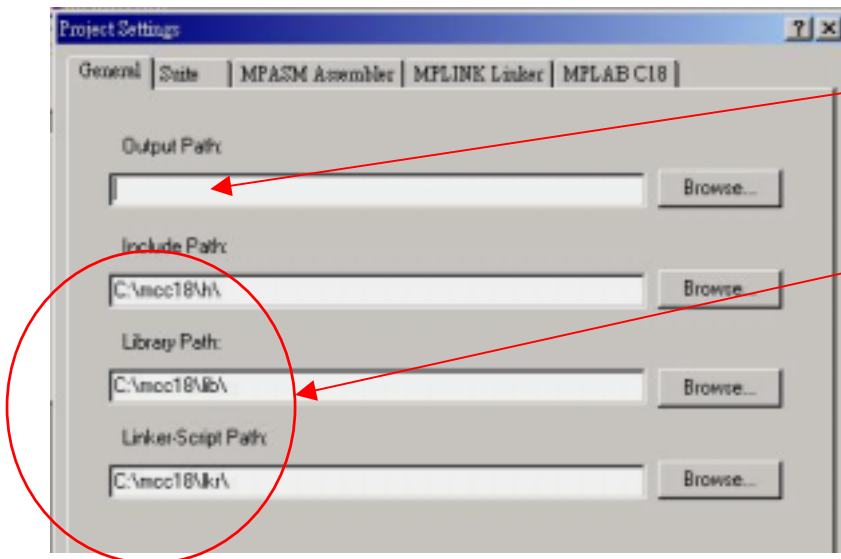


圖 3-12 C18 相關檔案路徑位置

Output Path 不設定路徑採用與 Lab3.mcp 相同之目錄

MPLAB-C18 相關檔案路徑設定，請一一分別加以指定不可以不設定或指錯其路徑，這樣在編譯的過程中會有錯誤產生。

此外使用者也可以單獨對語言工具作特別的指定，點選圖 3-12 上的書籤已進入單獨的設定選項：

- MPASM Assembler：設定 Hex 的輸出格式，大小寫辨識設定
- MPLINK Linker：設定是否要產生位址索引檔案(MAP)，建議使用時勾選此項目
- MPLAB C18：C Compiler 選項

現在你已經完成 C 語言的 Project 設定，試試看 **Build All (CTRL +F10)** 是否可組譯成功如圖 3-13 所示，至此你已經可以自行設定你所需要的 Project，不管是單一組合語言，可重新定位方式以及 C 語言的設定方式你都可以迎刃而解。

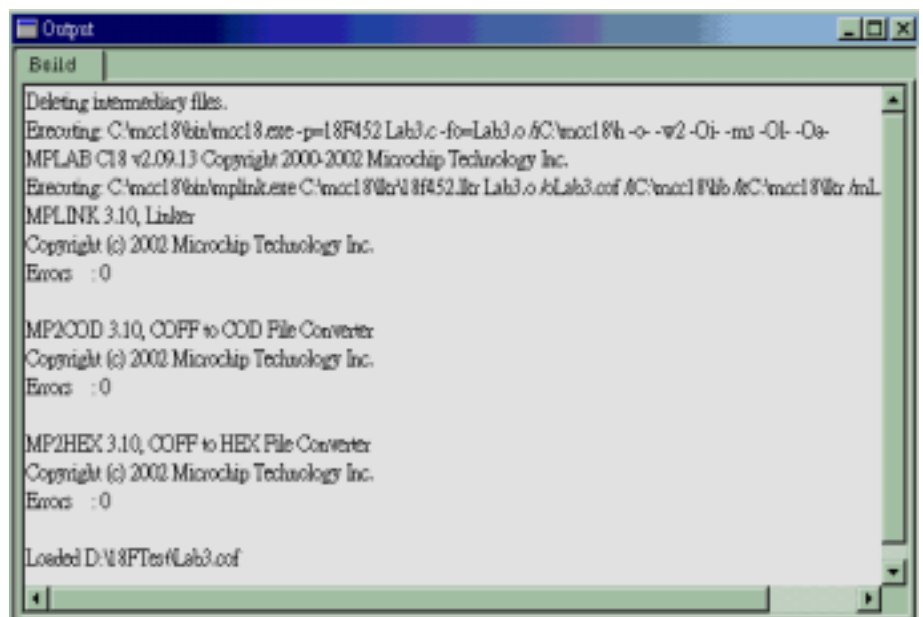


圖 3-13 C 語言編譯正確的結果

四、加入多個原始檔案

多數的軟體設計師常常使用一個以上的原始檔案來撰寫程式，這樣可將各個不同功能的模組寫在一起以方便程式的除錯與管理，同時對大程式的維護也較有層次分明的益處。MPLAB IDE 對這種多個原始檔案的管理提供了完整的方案，讓設計師檢視 Project 視窗就知道有多少原始檔案在使用。

關於多原始檔案的設定方式，只要在 Project 視窗的原始程式(Source Files) 下用老鼠右鍵加入你的原始檔案即可，如圖 3-14 所示。

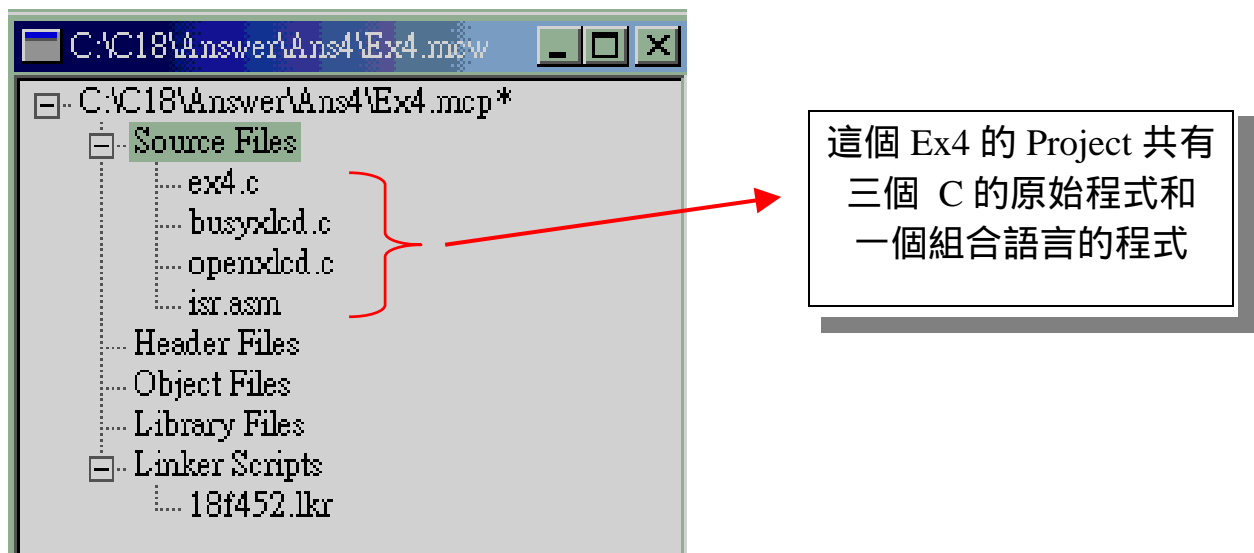


圖 3-14 多元使檔案設定例子

第四章 使用 MPLAB IDE

了解了 MPLAB IDE 對 Project 的管理、編輯、編譯及組譯各項功能後，你已經對 MPLAB IDE 使用上有個初步的認識，接下來我們要針對 MPLAB IDE 的各項功能做個較詳細的說明，基本上這些基本的操作方式都可以使用在 MPLAB SIM、ICD2、ICD2000 的工作環境，所以你只要會了所以只要會了 MPLAB IDE 的基本操作就能套用到各種除錯的工具環境。

4-1. “View”功能選項說明

要使用要使用 MPLAB IDE2 的工作環境首先就必須了解到底有那些重要的觀察視窗可以看到到底有那些重要的觀察視窗可以看到 PIC18F452 的各項內部資料。在主目錄下點選”View”開啟視窗選擇目錄：

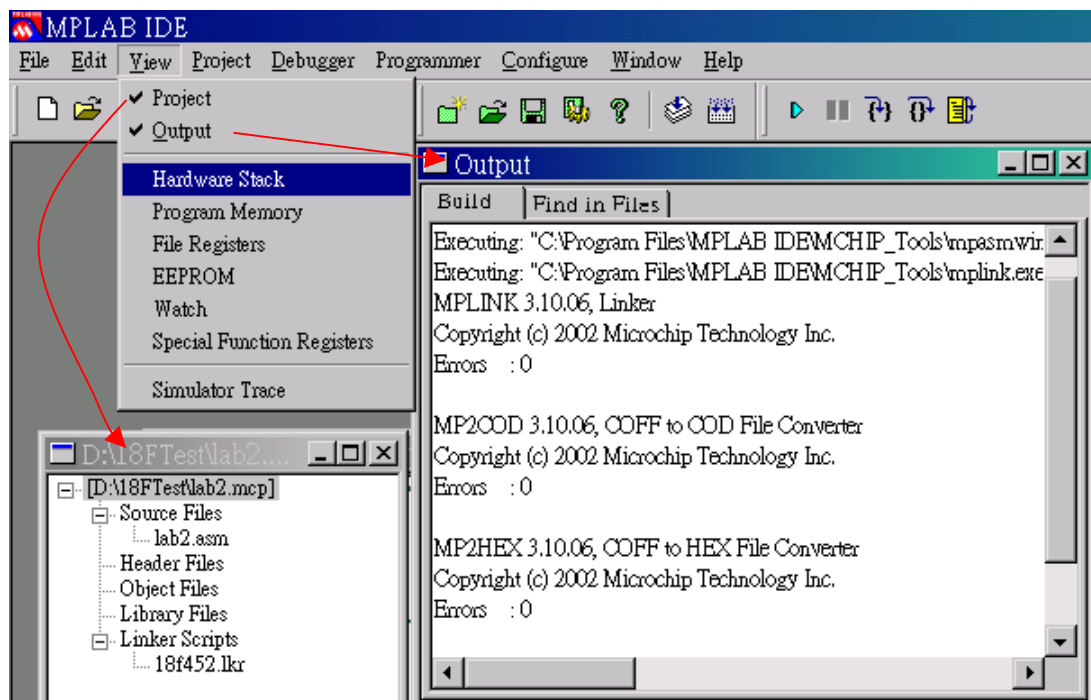


圖 4-1-1 MCU 觀察視窗選擇

勾選 Project 和 Output 顯示視窗，如圖 4-1-1 所示；Project 視窗顯示目前的工作平台裡含有那些檔案；Output 則顯示了程式編譯/組譯的結果，如有連接其他硬體(如 ICD2, ICE2000 等) 除錯裝置則會顯示出目前這些硬體的狀態。Output 視窗是一

個重要的事件觀察視窗，使用 MPLAB IDE 時應將其打開隨時檢查軟硬體的状态。

底下就常用之視窗逐一說明：

1. Hardware Stack：顯示硬體堆疊，以 PIC18F452 為例：此為 31 層深、21-bits 寬的硬體堆疊。本硬體堆疊指僅供程式計數器的存取，呼叫副程式(或函數)及中斷發生時才會使用到此硬體堆疊。
2. Program Memory：顯示程式記憶體的内容，該視窗可顯示三種不同的模式，使用者可依實際的需要進行切換：
 - a. Opcode Hex：顯示最基本的機械碼
 - b. Machine：顯示反組譯組合語言
 - c. Symbolic：顯示內含文字的反組譯組合語言

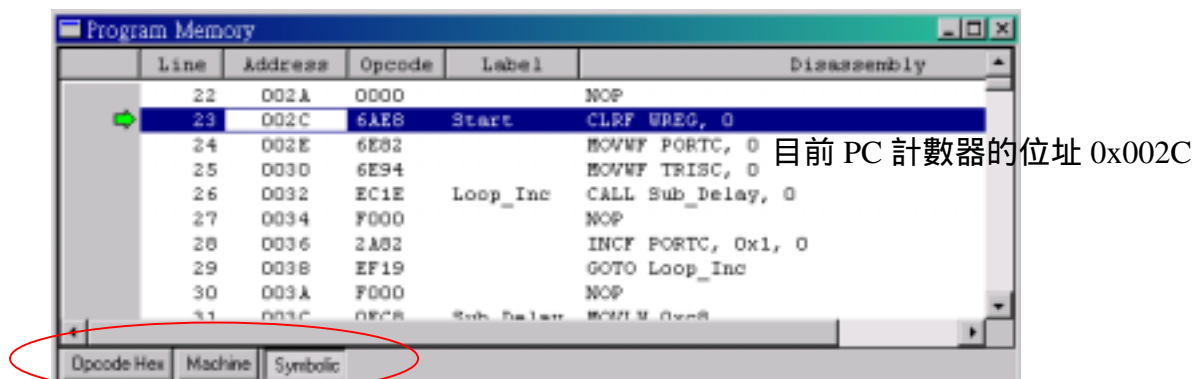


圖 4-1-2 顯示程式記憶體的内容

3. File Registers：顯示 RAM 記憶體的内容，使用者也可以直接修改 RAM 的内容。將游標移到欲修改的位置點選一下，直接輸入十六進制的值後按 Enter；使用者也可以用'Import Table / Export Table'的方式輸入一整個 RAM 的設定值(按老鼠的右鍵即可點選此功能)。
4. EEPROM：顯示 EEPROM 資料記憶體的内容。使用時應特別注意：如果你使用 ICD2 來除錯，此時所顯示的 EEPROM 内容是電腦(PC)內的舊值，而不是 18F452 內的 EEPROM 的内容值；如果你想觀察真正的 EEPROM 的内容值，就必須用"Read Target Device"的方式讀取 EEPROM 的内容值。

5. Watch：為一非常重要性的變數或暫存器觀察視窗。在 C 語言(組合語言也一樣)下你知道變數被 Linker 安排在哪裡嗎？這在 C 語言的除錯過程中能看到的變數的內容是非常重要的，Watch 視窗就是用來觀察變數的視窗。

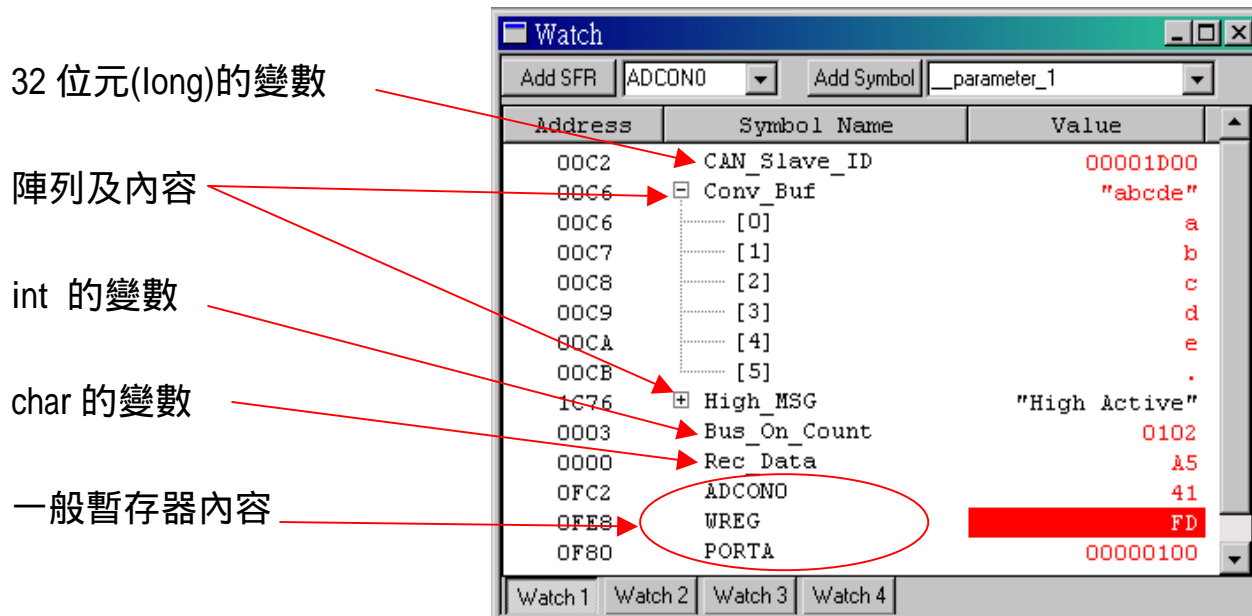


圖 4-1-3 Watch 視窗顯示

- Add SFR：增加暫存器顯示項目，選擇欲顯示的暫存器後按”Add FSR”加入此暫存器於觀察視窗行列。
- Add Symbol：增加變數顯示項目，選擇欲顯示的變數後按”Add Symbol”加入此變數於觀察視窗行列。

Watch 所觀測的變數會自動調整該變數的長度，使用者也可以改變觀測的顯示方式(二進制、十進制、ASCII ...等)，只要點選變數名稱再按 mouse 的右鍵選擇”內容(Properties..)”就可以修改顯示的模式。你也可以直接點選該變數(暫存器)的內容值採直接輸入方式修改此變數的內容。在 Watch 視窗下，欲觀測的變數也可以直接修改，mouse 選該變數(暫存器)的名稱後，直接輸入新的觀察測變數名稱即可。

總之，Watch 視窗是一個很重要的視窗，如果你要使用 MPLAB IDE 來除錯，你就必須了解它、熟悉它。透過 Watch 視窗的協助我們才能對變數掌握自如。

6. Special Function Registers : 顯示暫存器的內容，並可用 mouse 點選暫存器的內容後並修改之；一般在使用上有了 Watch 視窗該項顯示暫存器功能就不是那麼重要了。

4-2. “Project”功能選項說明

Project 的選項功能大致已在第三章說明過，讀者如對讀者如對 Project 選項不甚了解的話請參考第三章的說明。

此區功能為：建立、開啟、關閉 Project

Clean：清除編譯暫存檔案。

Build All：重新編譯所有的原始檔案以產生新的 OBJ 檔後再由 Linker 重新連結安排位址。

Make：只編譯修改過的原始檔案，產生新的 OBJ 檔後再由 Linker 與其它 OBJ 連結安排位址。

Build Options：編譯器的設定選項與 C18 相關檔案的路徑設定。

此區功能請參閱第 3-6 章的說明。

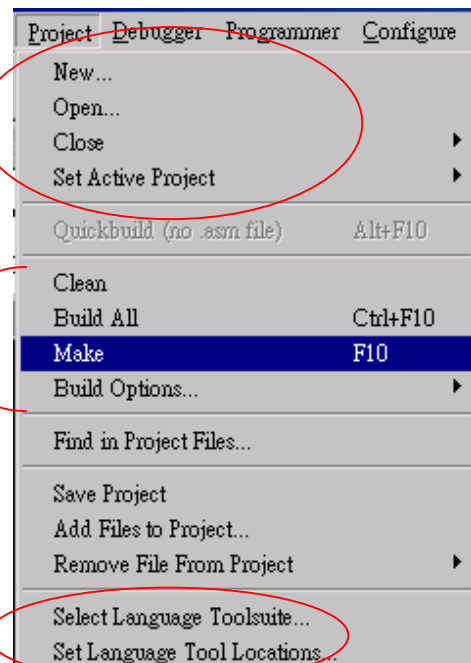


圖 4-2 Project 試窗顯示

4-3. “Debugger”功能選項說明

Debugger 的選項為 MPLAB IDE 中一向非常重要的選項，所有除錯的功能均在此目錄下進行，讀者必須詳細了解各項除錯的功能。MPLAB IDE 的除錯方式與步驟，基本上在在使用 MPLAB SIM，MPLAB ICE2000，MPLAB ICD2 ... 等均大同小異，為了能讓使用者能迅速了解除錯以下的說明是使用 MPLAB SIM 的軟體模擬方式說明。

啟動軟體模擬，只要在主目錄下點選”Debugger→Select Tool→MPLAB SIM”即

可啟動軟體模擬，此時在視窗的最下角左方使用工具狀態會出現”MPLAB SIM”，請注意模擬的元件是 PIC18F452，讀者可參考第二章的說明進行元件的切換及設定。

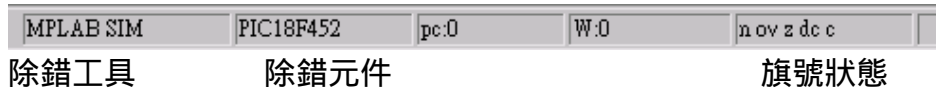


圖 4-3-1 MPLAB IDE 狀態顯示

1. Select Tool：選擇除錯工具，在安裝完成 MPLAB IDE 後軟體模擬會自動安裝在系統內，其它除錯元件需另外安裝，讀者可參考第五章 ICD2 的安裝說明及第六章的 ICE2000/ICD4000 的安裝說明。
2. Clear All Memory：將目前的 Program 清為 0xFFFF (NOP)；RAM 的內容清為零。
3. Run，Halt，Step Info，Step Over，Reset 為重要的除錯工具，使用時可直接使用功能鍵(例：Run 按 F9)或直接點選快速圖式功能按鈕(如圖 4-3-3 所示)。

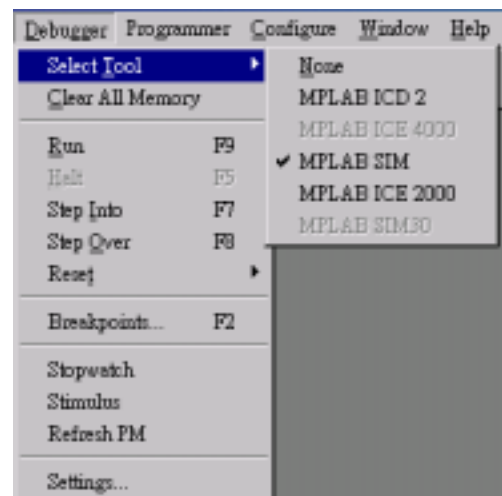


圖 4-3-2 Debugger 視窗顯示

圖 4-3-3 Debugger 視窗顯示



- Run：全速執行
- Halt：暫停
- Step Info：單步執行
- Step Over：進階單步執行；此項單步執行如遇到呼叫副程式或函數時，程式會直接以全速方式執行該副程式或函數後返回停在呼叫點下一個指令上。
- Reset：重置；將 MCU 做 Reset 的動作(記注是熱開機)，RAM 的值將會

除錯圖式功能按鈕，由左自右依序是 Run，Halt，Step，Step Over，Reset

被保留，暫存器的值恢復到 Reset 的內定值，PC 計數器歸零。

4. Breakpoints：中斷點的設定功能視窗；MPLAB IDE 設定中斷點的功能非常容易使用，使用者可利用 Breakpoints 選項加以設定或解除中斷點；更簡易的方式只要在原始程式中點選預設中斷點的指令或敘述，利用 mouse 的右鍵即可輕易設定中斷點。中斷點設定的多寡與使用的工具有關，MPLAB SIM 中斷點的設定不限制，ICD2 就只能設定一個中斷點。

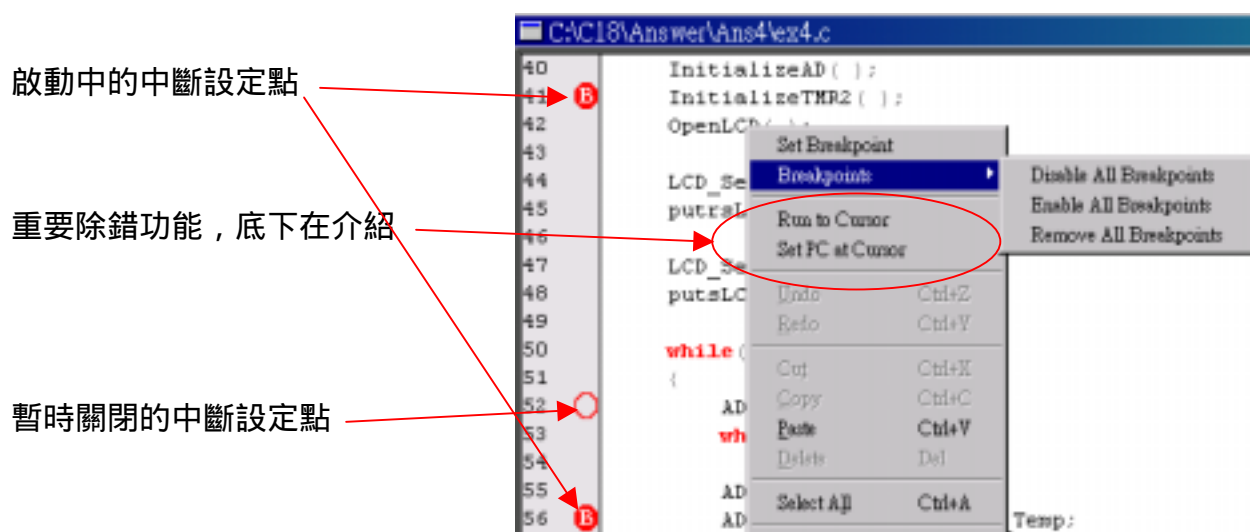


圖 4-3-4 中斷點設定功能

- **Run To Cursor**：程式將從目前的 PC 計數器位址開始執行，直到執行到目前游標所在的位址後停止執行。
- **Set PC at Cursor**：將 PC 計數器直接設定到目前游標所在的位址，使用此項功能時應特別小心，對一些副程式的呼叫及返回位址應特別注意。

5. Stopwatch：程式執行時間的計時馬錶功能；使用者可利用此項功能輕易計算出程式由某點到某點以目前的振盪頻率為基準所需的執行時間。尤其是用來算出延遲副程式或計算 I/O 輸出的脈波的時間時，為一非常好用的工具。目前該項功能只能用於 MPLAB SIM，無法在 ICD2 及 ICE2000 上執行。

使用 Stopwatch 時最好能配合中斷點的設定。使用時先將 PC 執行到欲計數的起點位置並在計數的終點設定一中斷點，在 Stopwatch 視窗下按”Zero”

按鈕先將計數器歸零，點選 RUN 圖示以執行程式直到程式停止或用 HALT 停住，此時 Stopwatch 所顯示的計數值就是程式所花的執行時間。

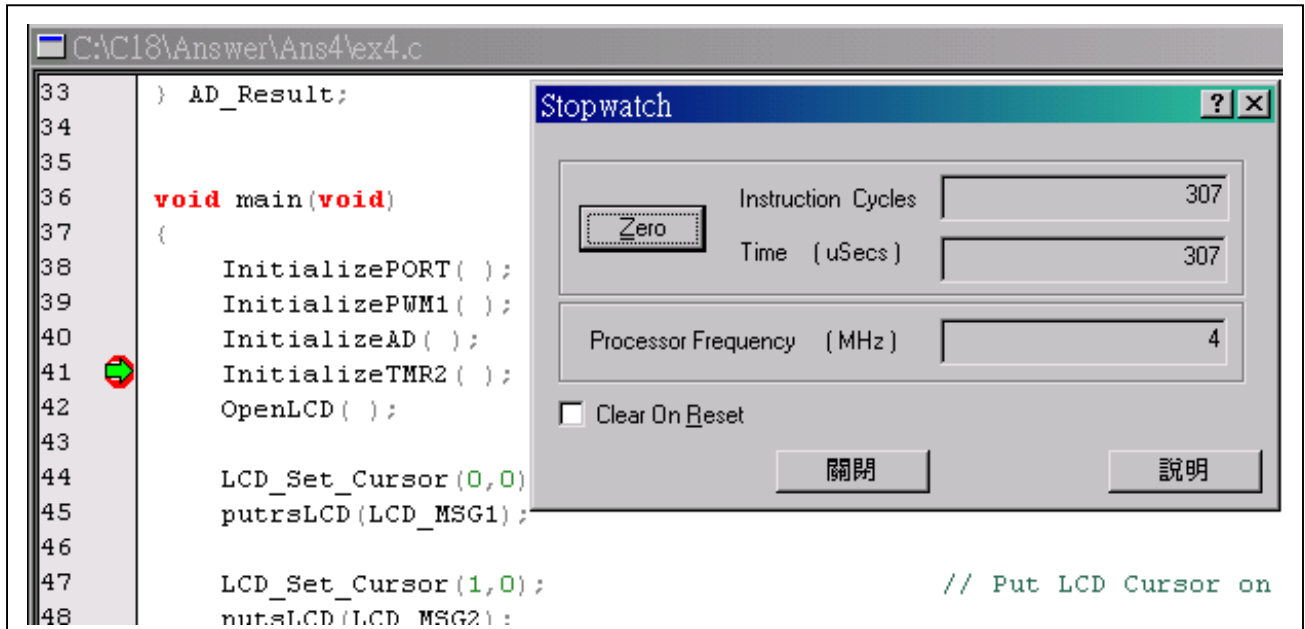


圖 4-3-5 Stopwatch 計時功能視窗

以圖 4-3-5 為例，程式從 Reset 後開始計時，從啟動模組”c018i.o” 開始執行到 `main()` 函數，再到中斷點(`InitializeTMR2()`)，共花了 307uS @4MHz。

6. Stimulus：軟體模擬激發輸入。在使用軟體模擬時，一般而言模擬的輸出比較容易觀察的到所以較無問題，但在想輸入一些 I/O 的狀態給軟體模擬時就會有些困難；Stimulus 的輸入激發功能就可以解決此輸入的困擾。由於此項功能是 MPLAB SIM 的專屬功能，故不在此章討論，讀者可參考第七章的 MPLAB SIM 章節。

4-4. “Edit”功能選項說明

Edit 的視窗選項大致與 Windows 的文書編輯程式相同，我們須知到的是文書編輯中的一些控制內容。在主目錄下點選”Edit → Properties”以開啟 Edit Options 對話視窗如圖 4-4-1 所示。

在 Edit Options 視窗下共有三種設定選項，第一為 Editor 設定，第二項為 Text 設定及第三 Tabs 設定。

在 Editor 設定選項下可設定

- 原始程式中是否要顯示行號。
- 在原始程式中，當 mouse 指到變數時要不要顯示其內容值。
- 簡易中斷點的設定或移除(mouse 連續按兩次)。

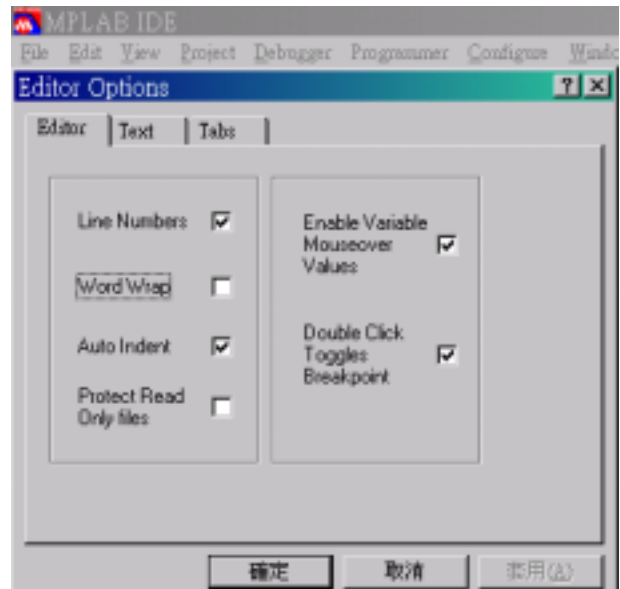


圖 4-4-1 Editor 的設定選項

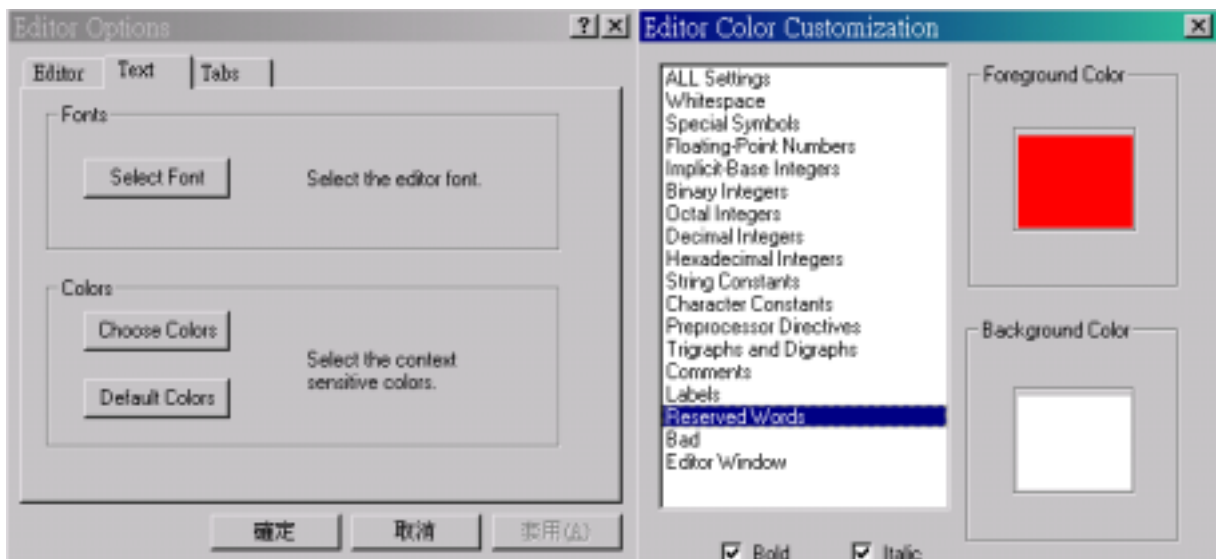


圖 4-4-2 Text 的設定選項

在 Text 設定選項下“Select Font”可以設定原始程式中顯示的字型及大小，“Choose Colors”則可設定原始程式中的各種分類顏色，讀者可依實際喜好去設定顏色，該項設定值會被儲存套用。最後第三項是設定 Tab 的間隔，一般設定為 4



MICROCHIP
The Embedded Control Solutions Company™

MPLAB[®] IDE v6.10 使用手冊

或 8 個間格。

5-1. 安裝 ICD2 的驅動程式

MPLAB ICD2 的驅動軟體屬於外掛式的，安裝 MPLAB IDE V6.10 後 ICD2 的 USB 驅動程式需自行安裝。ICD2 的驅動程式放在”Driver2000\ICD2_USB\”的目錄下(以 Windows 2000 作業系統為例)，使用者也可以依照 Ddcd22k.htm 所載之方式進行安裝。(讀者可參考第六章的安裝 ICE2000 的範例)

ICD2 對 PC 的通訊介面有兩種，一為 RS-232 另一為 USB，使用上 USB 因有較高的通訊速度所以本使用手冊範例是以 USB 為主。



圖 5-1 安裝 ICD2 驅動程式的目錄

5-1. 設定 ICD2 與連線

在使用 ICD2 作為發展工具之前我們還是要對 ICD2 有所了解，由於 ICD2 是會使用到 IC 內部的一些資源所以有些限制是必須知道的，使用者可參閱 ICD2 的使用手冊以了解更多的訊息。在此僅列出常用的 PIC18 系列在使用 ICD2 需注意的事項：

微處理器編號	ICD2 使用的暫存器	ICD2 使用的程式記憶體
PIC18F6720/8720	0xEF2 – 0xEFF	0x1FDC0 – 0x1FFFF
PIC18F6620/8620	0xEF2 – 0xEFF	0xFDC0 – 0xFFFF
PIC18F458/258/452/252	0x5F4 – 0x5FF	0x7DC0 – 0x7FFF
PIC18F448/248/442/242	0x2F4 – 0x2FF	0x3DC0 – 0x3FFF
PIC18C601/801	0x3F6 – 0x5FF	None

底下我們就使用 ICD2 來發展 PIC18F452 的程式為例來說明如何使用 ICD2

- 一、 設定以 PIC18F452 為開發案的 Project (參考第三章之說明)
- 二、 在主目錄下點選”*Debugger → Select Tool → MPLAB ICD2*”
啟動 ICD2 除錯工具，如圖 5-3 所示

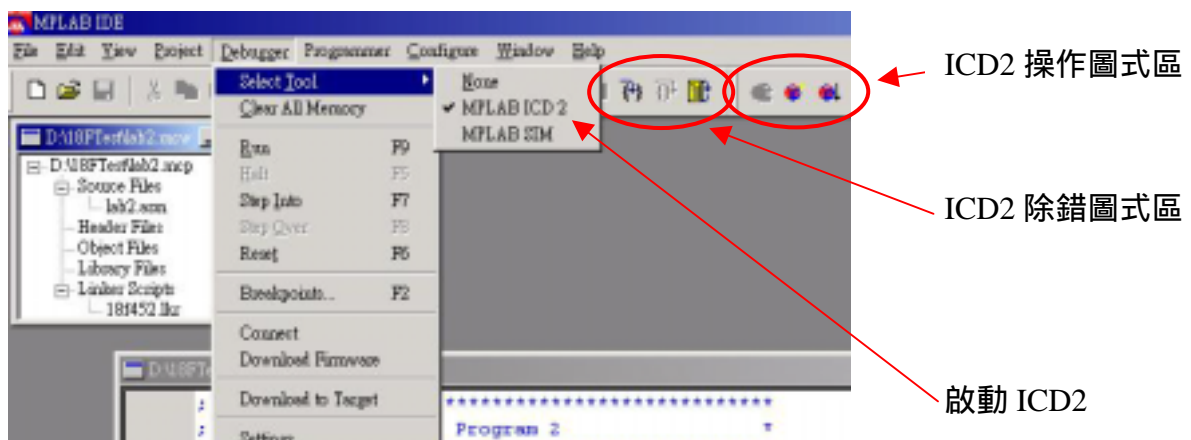


圖 5-3 啟動 ICD2

- 三、 點選”*Debugger → Setting*”進入 ICD2 的使用環境設定，此項的設定對使用 ICD2 是很重要的，依據圖 5-4 所示總共有六個選項需設定，我們將一一說明之



圖 5-4 ICD2 設定選項

Status 設定說明：

ICD2 允許一啟動時採自動連線方式運作，使用時可勾選此項功能。令外 ICD2 會將自我測試的結果顯示在視窗上。

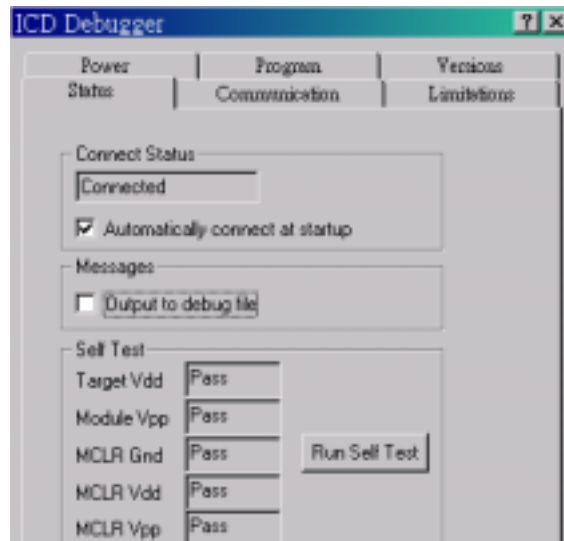


圖 5-4-1 ICD2 狀態顯示

Communication 設定說明：

ICD2 與 PC 的通訊方式有兩種選擇，第一為 RS-232 模式其最高速度為 57600bps 另一為 USB 介面採用 Full Speed 的速度，為讓 ICD2 能有較高的資料傳送速度建議使用 USB 介面。

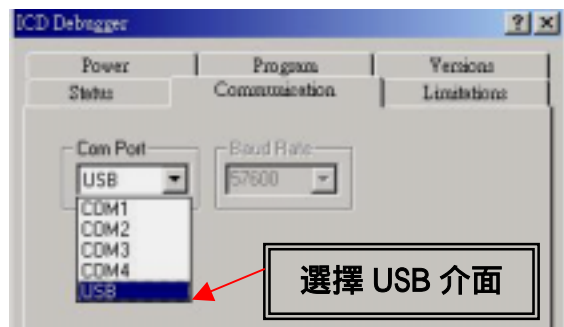


圖 5-4-2 ICD2 通訊設定

Power 設定說明：

在使用 ICD2 時，ICD2 的電源是可以從 Target 板或單獨給電的方式工作。在這裡我們建議不要勾選”以 ICD2 的電源作為 Target 板的電源來源”，這樣是做法主要是避免因 Target 板的負載過大而導致 ICD2 過電流而損壞。

圖中所顯式的 Target Vdd = 4.96V 是因為 ICD2 與 Target 板連線成功時所顯示出的 Target Vcc 電壓，如果沒有連線則此欄位會顯示空白。

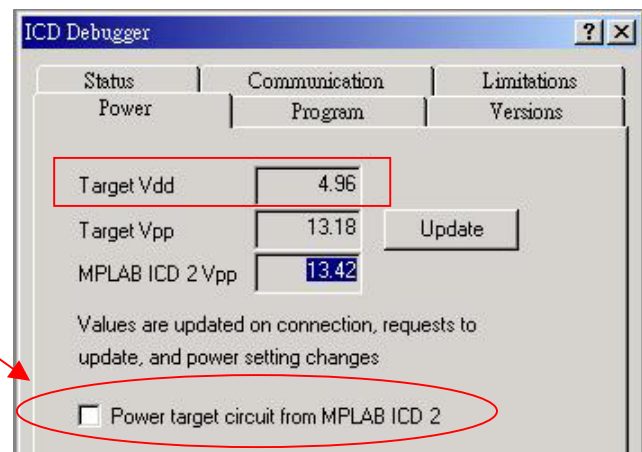


圖 5-4-3 ICD2 電源設定

USB 模式下，ICD2 的電源由 USB 供給不需額外的電源供應器，但 Target 板的電源必須單獨供應不可自 ICD2 取得電源。

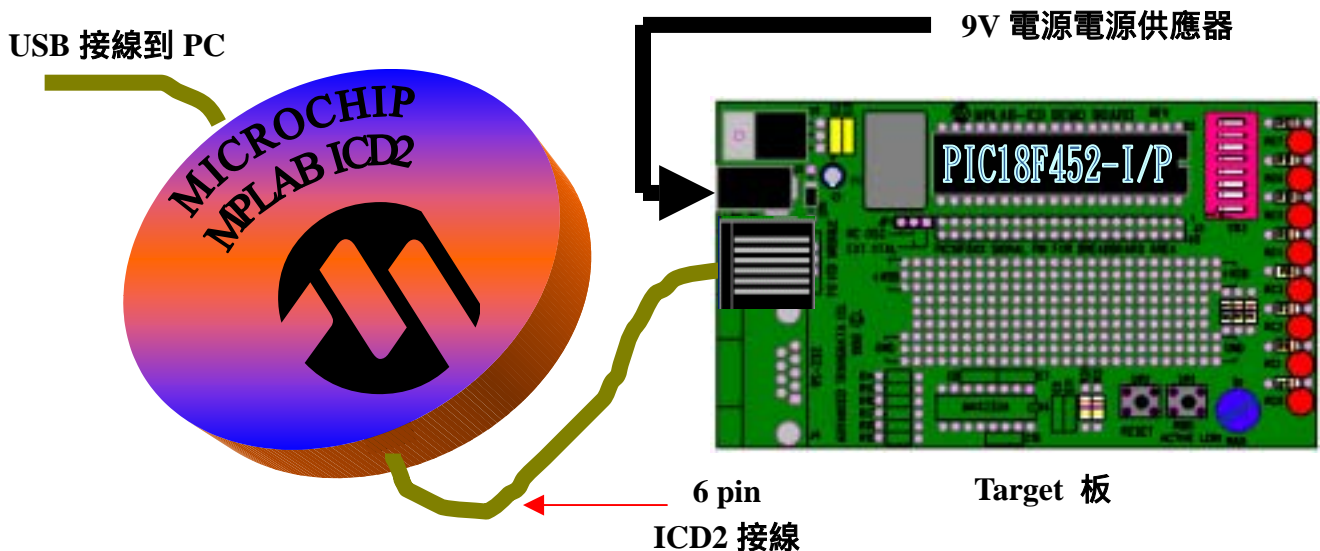
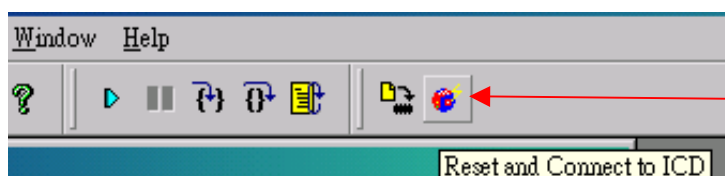


圖 5-4-4 ICD2 電源示意圖

RS-232 模式下，ICD2 可單獨供電或自 Target 板取得工作電源，但還是最好能避免將 ICD2 的電源供給 Target 板。

在此想說明一下實際使用 ICD2 的實際經驗與建議：使用 ICD2 時供電的方式有一定的順序，按底下的步驟給電，ICD2 的連接就會很順手。

4. 開始時確定實驗板的電源是關掉的。
5. 連接 USB cable 到 ICD2，此時 ICD2 的 Power 的綠色 LED 會亮。
6. 將 ICD2 的 6-pin cable (電話接頭線) 連接到實驗板的 ICSP 插座上。
7. 將電源連接至實驗板上。
8. 啟動 MPLAB IDE v6.10 後，再點選 Debugger→Select Tool→MPLAB ICD2 以啟動 ICD2，MPLAB 會自動開啟 Output 視窗讓使用檢查 ICD2 目前連線的狀態，點選 Reset ICD2 圖示開始連接上線。



按此圖示建立
ICD2 連線

Program 設定說明：

這一項的設定對 ICD2 而言是很重要的，因為你的程式是要被先被載入到 ICD2 內，再由 ICD2 以 ICSP 的方式將程式燒錄到 PIC18F452 的 Flash 中去執行。

Select Memories 是用來設定目前的燒錄將會動到那些記憶體，使用時必須特別加以注意。

Program Options 選擇”燒錄前清除所有的 Flash 記憶體”。

IDs and Checksum 選擇，如果要燒錄 ID 時要使用 Checksum 或 User's ID。要有此項功能就必須點選 ID 項目，至於 User's ID 的設定可在主目錄下點選”*Configure → ID Memory*”來輸入 ID。

Program Memory Address 是用來設定此次的記憶體燒錄範圍，選擇適當的燒錄範圍可節省燒錄時間，但一定要慎重不可設錯此燒錄範圍，這樣反而會讓程式只有前段，而後面的程式可能會因你將範圍設錯而沒有被燒錄進來。至於程式的大小，使用者可查看 LST 或 MAP 檔案(在 MPLINK 的選項設定)。

提到燒錄時間，在此順便提一下 Microchip 的新的 FLASH 製程”PEEC”，對一個 BLOCK 只需 1~2mS 的燒錄時間。以 PIC18F452 為例，它共有 32K-Bytes 的 Flash Memory，每一個 BLOCK 為 32 Bytes，所以所以 18F452 的燒錄時間只需 1~2 秒，當然這是純指燒錄時間，但一個個 MCU 再燒錄時還有傳送資料時間還有驗證時間，前後加起來以 ICD2 燒錄 8K-Bytes (0x0000~0x1FFF)的時間經實測燒錄約為 3 秒鐘驗證約為 2 秒，整個載入程式程序約為 5 秒；如果載入 32K 的話也只要 15 秒而已。

另外有兩個選項分別是”*External Memory Address*”及”*Bootloader 內的 User Memory Routine*”，目前我們是使用單晶片微控制器模式(Single Chip MCU)所以不會

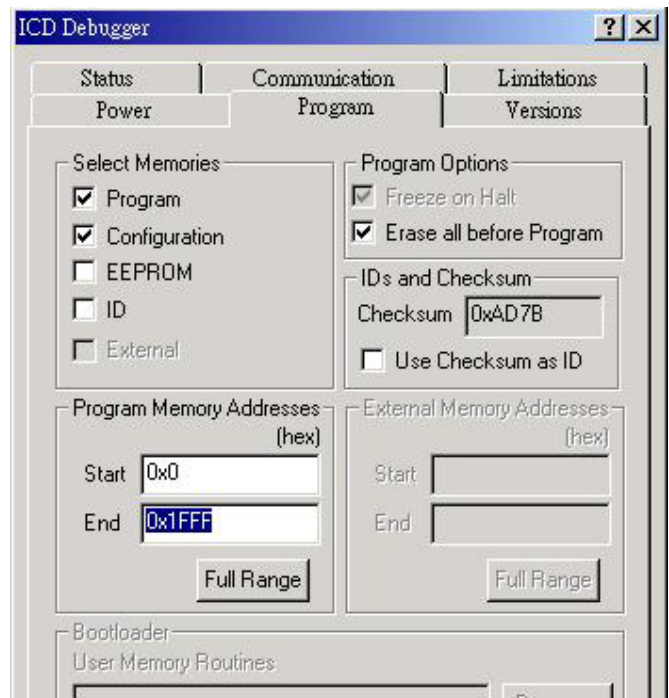


圖 5-4-5 ICD2 燒錄設定

用到，如果要使用微處理機模式(Micro-Processor Mode)例如使用 PIC18C601，PIC18C801 時就會使用到這兩個選項，容後在述。

Versions 版本顯示：

點選到這個項目時你將會看見目前 ICD2 所使用的驅動程式版本。這些驅動程式說明如下：

MPLAB ICD2 Version 是指目前你所使用的 MPLAB IDE 驅動軟體版本(圖右所示為使用 MPLAB IDE v6.10 時相對應的 ICD2 驅動版本)。

Firmware Version 是指 ICD2 本身的軟體版本，ICD2 本身也有一顆 Flash 的微控制器 PIC16F877 在裡面控制 ICD2 的工作，這個 MCU 的控制軟體是可以自 MPLAB 的主目錄下點選 ”Debugger→Download Firmware”將最新本的驅動程式載入。

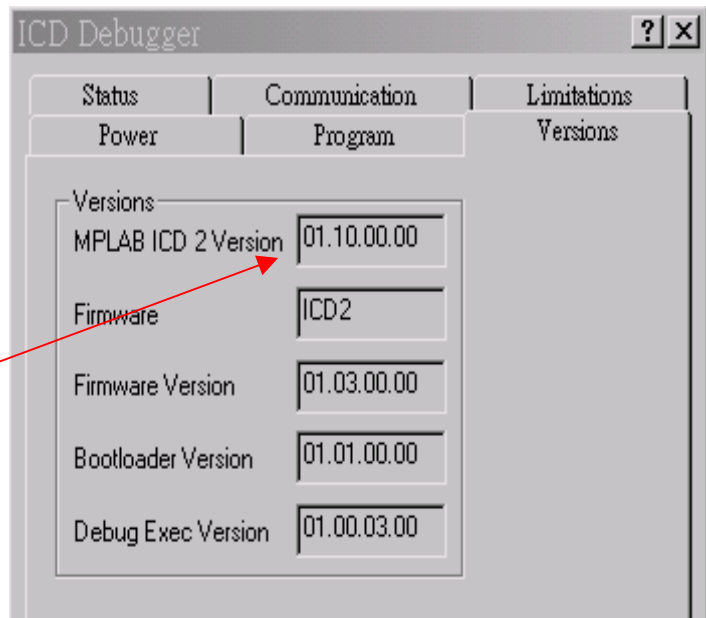


圖 5-4-5 使用 MPLAB IDE v6.10 時所顯示的版本訊息

注意！第一次使用 ICD2 時，務必重新載入目前 MPLAB IDE 版本的 ICD2 Firmware 以確保使用的版本能與 MPLAB 相容。

Bootloader Version 是指 ICD2 內部 PIC16F877 的 Bootloader 程式，目前這個驅動程式使用者無法自行升級更換，必須送回代理商或 Microchip 升級。

Debug Exec Version 載入到 PIC18F452 Target 板的除錯程式版本，這個程式放在 ICD2 的目錄下，在完成載入程式到 Target 板後就可以顯示這個版本訊息。

5-2. 開始使用 MPLAB ICD2

ICD2 在使用上只要兩條電纜線連接到 USB 插座和實驗板上，使用起來非常簡單方便，但先天上 ICD2 也是有一些限制的，例如：ICD2 在除錯時只能設定一個中斷點，ICD2 會使用到 RB6, RB7 兩隻 I/O 腳...等。但 ICD2 的低價位，配合 MPLAB 可做高階的原始程式除錯、程式燒錄，不失為一物美價廉的發展工具。

為了確定 ICD2 的連線成功，我們必須在 Output 的視窗下看到如圖 5-2-1 的連線成功的訊息，使用著可在主目錄下點選 Debugger→Connect 以起動 ICD2 的連線。

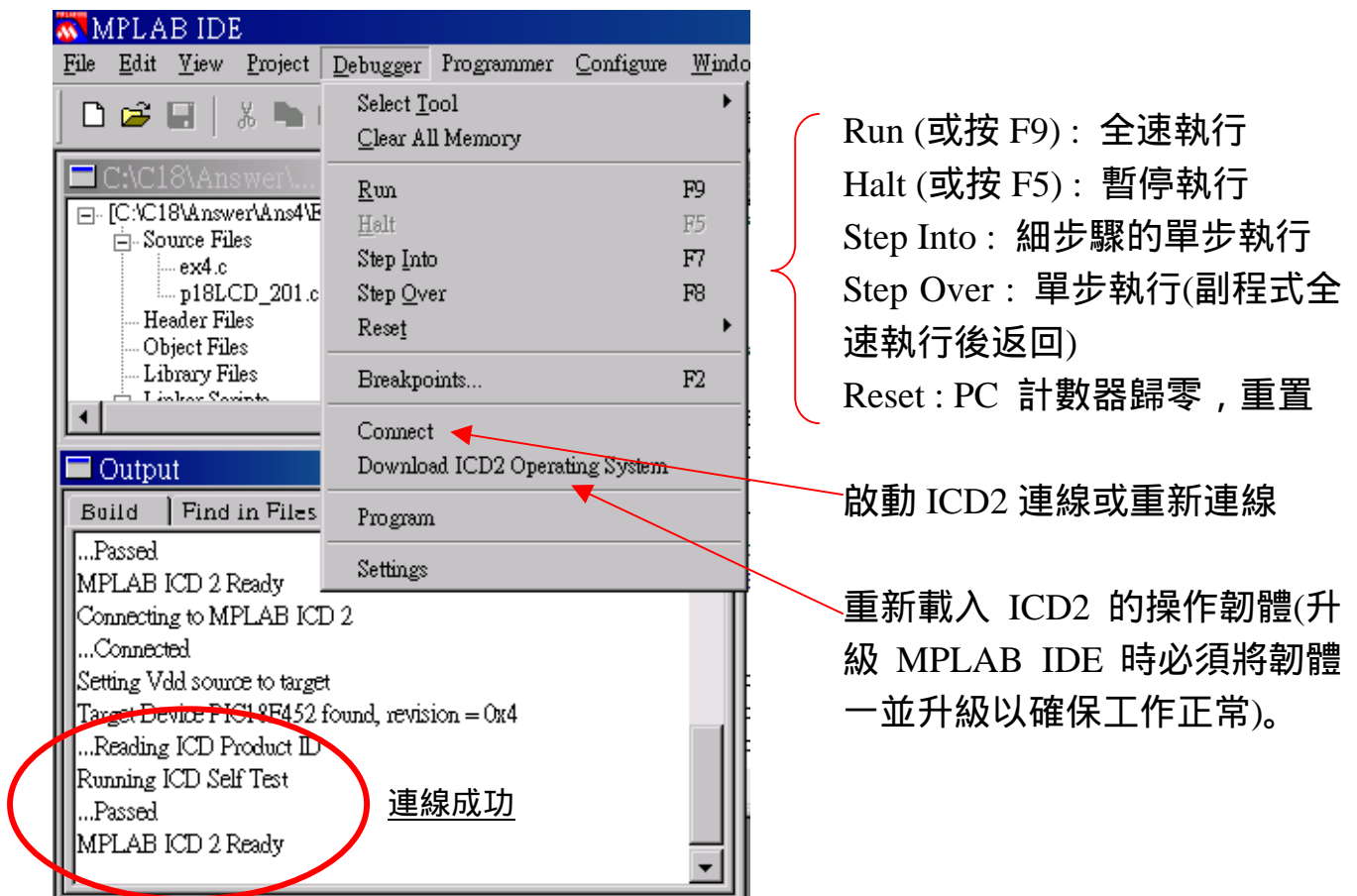
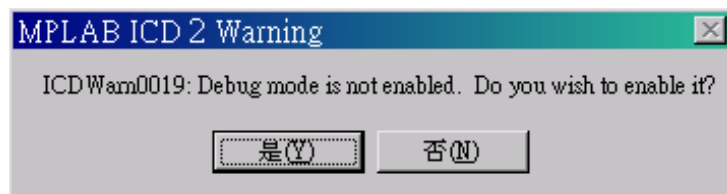


圖 5-2-1 的連線成功

Project 開啟後並啟動 ICD2 的連線，確定連線成功後，接下來就可以進行除錯的步驟。由於 ICD2 的除錯是必須將程式碼 (Hex Code) 燒錄到 PIC18F452 的程式記憶體器裡，然後對 PIC18F452 下達各項除錯的命令。底下就 ICD2 除錯的基本步驟條列下來以供參考：

6. 啟動 Project
7. 建立 ICD2 的連線
8. 設定 Background Debugger Mode (在 Configure → Config. Bits 選項中)
9. 確定編譯/組譯的過程中沒有錯誤產生，且有產生 Hex 檔。
10. 現按 Reset ICD 圖示確定連線正常，在點選”Program Target Device”圖示將程式碼燒錄到 PIC18F452 裡，必須確認燒錄與驗證過程均無任何錯誤發生。
11. 按 Reset 圖示(或按 F6)將 PIC18F452 重置，PC 計數器歸零。
12. 開始除錯 (重複步驟 4 到步驟 7 直到程式完成)
13. 關閉 Background Debugger Mode，重新燒錄 PIC18F452，MPLAB IDE 此時會詢問(如下圖所示)，只要回答 ”否” 則 PIC18F452 就可以不用 ICD2 來控制，直接就可以單獨執行程式(記得把 ICD 的電線拔除)。



第六章 使用 MPLAB ICE2000

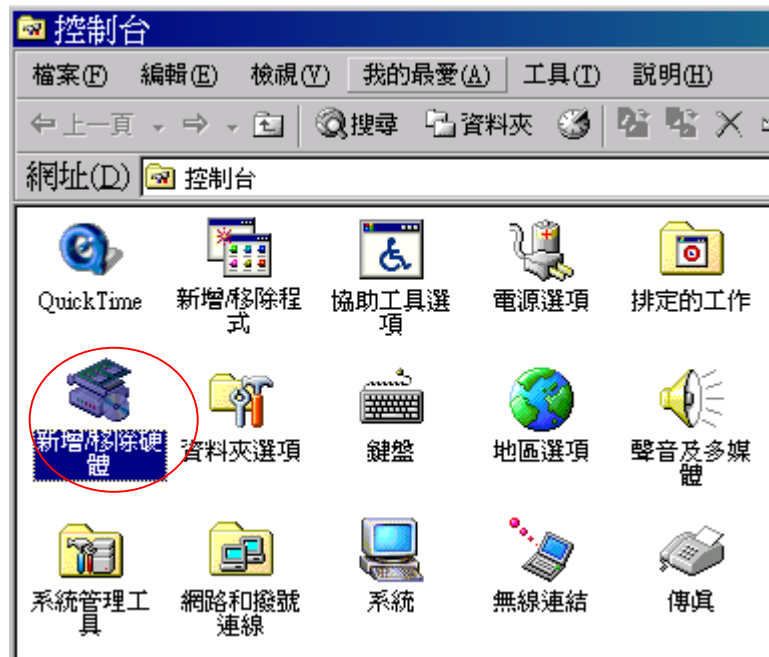
6-1. 安裝 ICE2000 的驅動程式

MPLAB ICE2000 的驅動程式必需另外安裝,使用者可參考底下的步驟進行安裝

步驟一

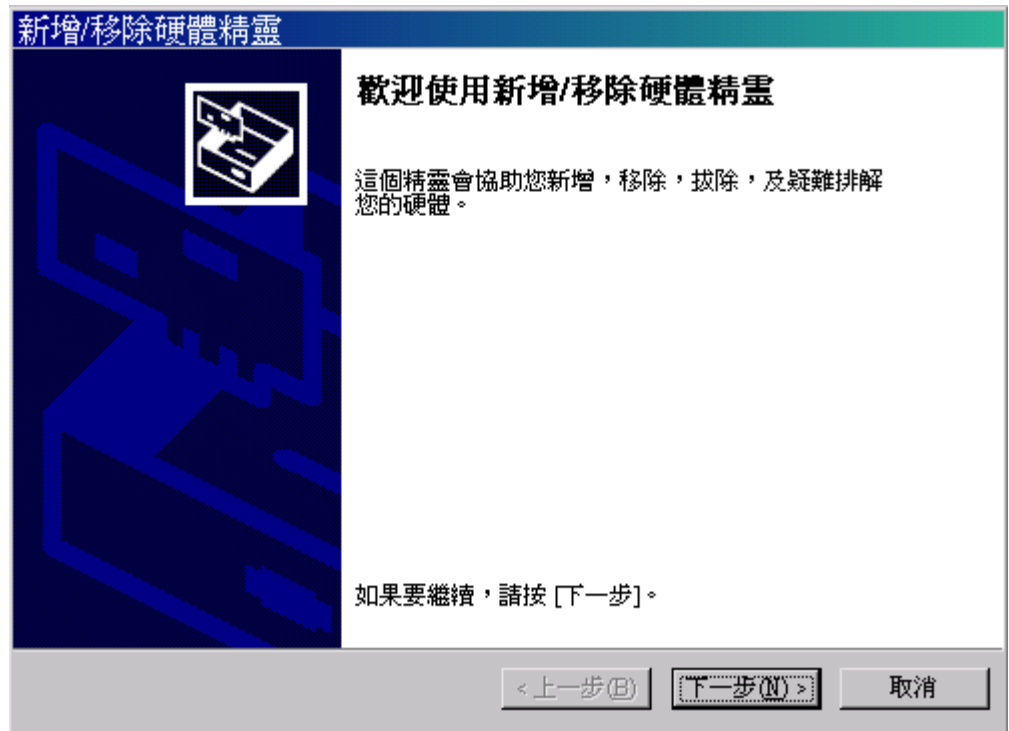
點選”開始→設定→控制台”打開控制台視窗後點選新增/移除程式的圖示以加入 ICE2000 的驅動程式。

(以 Windows 2000 為安裝範例)



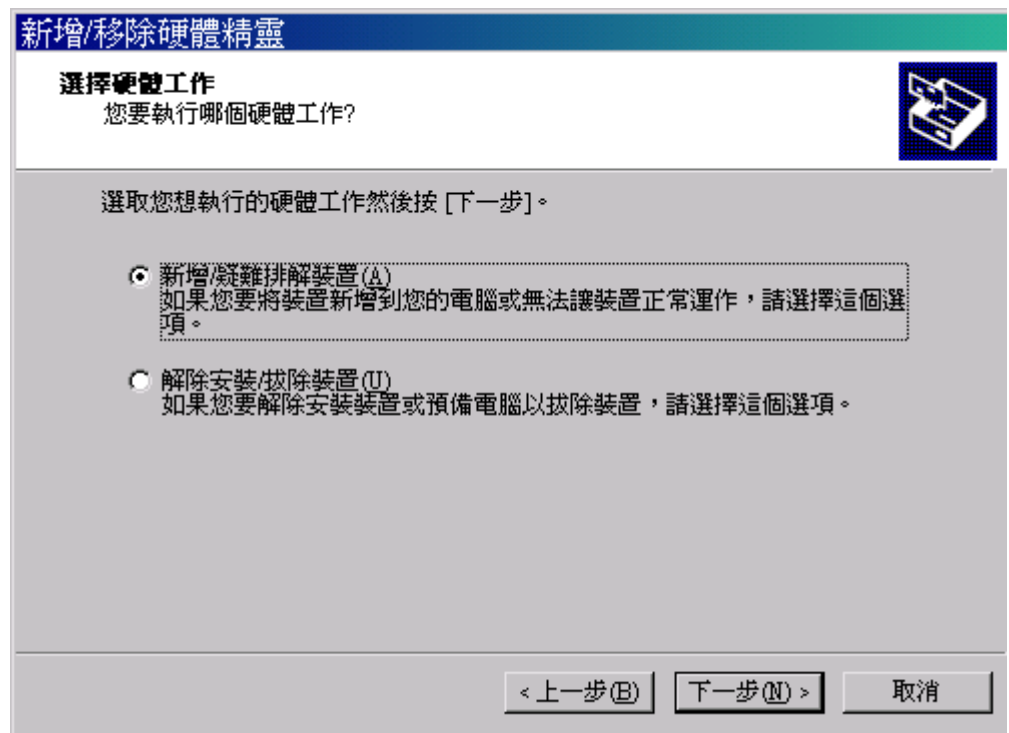
步驟二

啟動新增/移除精靈來加入新的硬體，按下下一步進入下一畫面



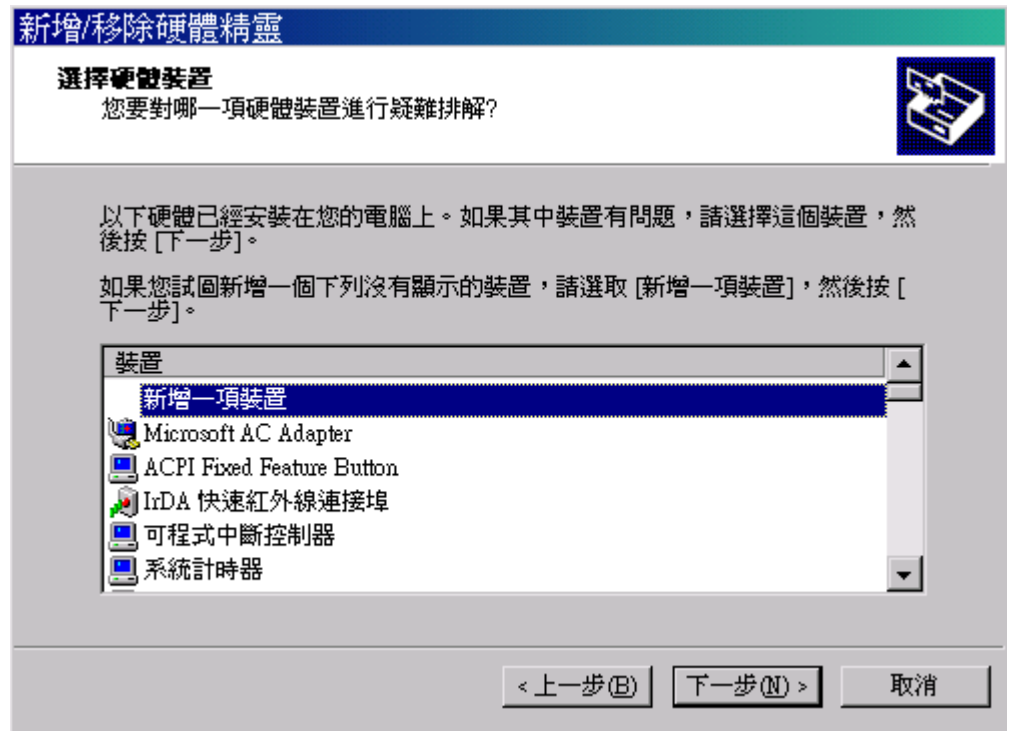
步驟三

點選新增/疑難排解裝置，按下下一步進入下一畫面



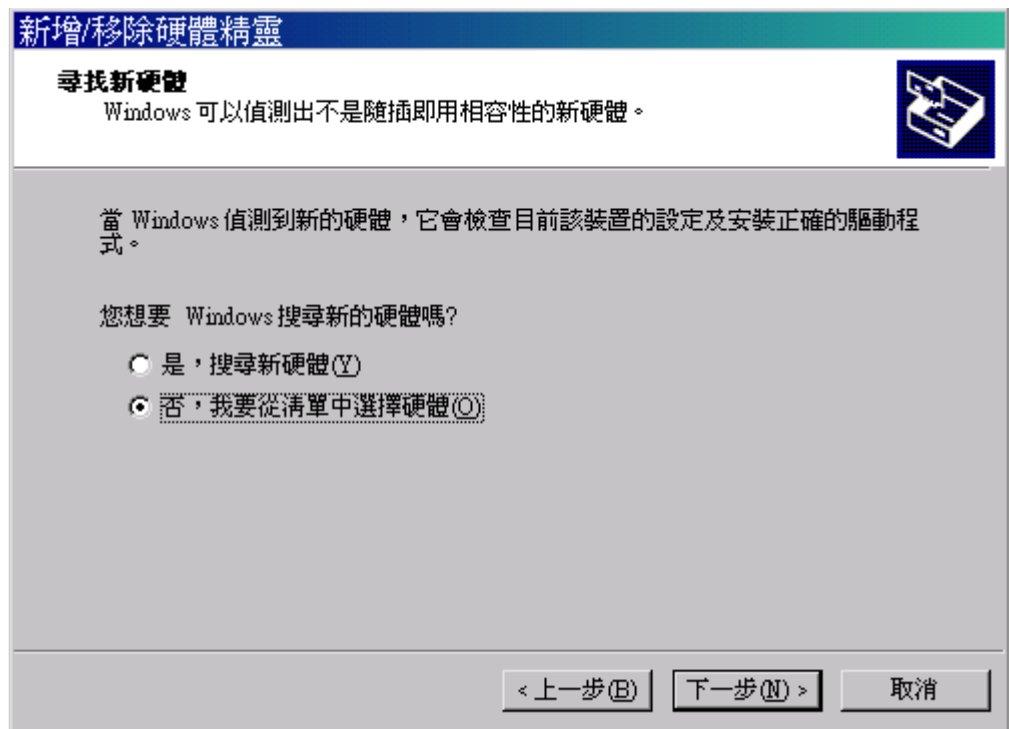
步驟四

點選新增一項裝置，按下一步進入下一畫面



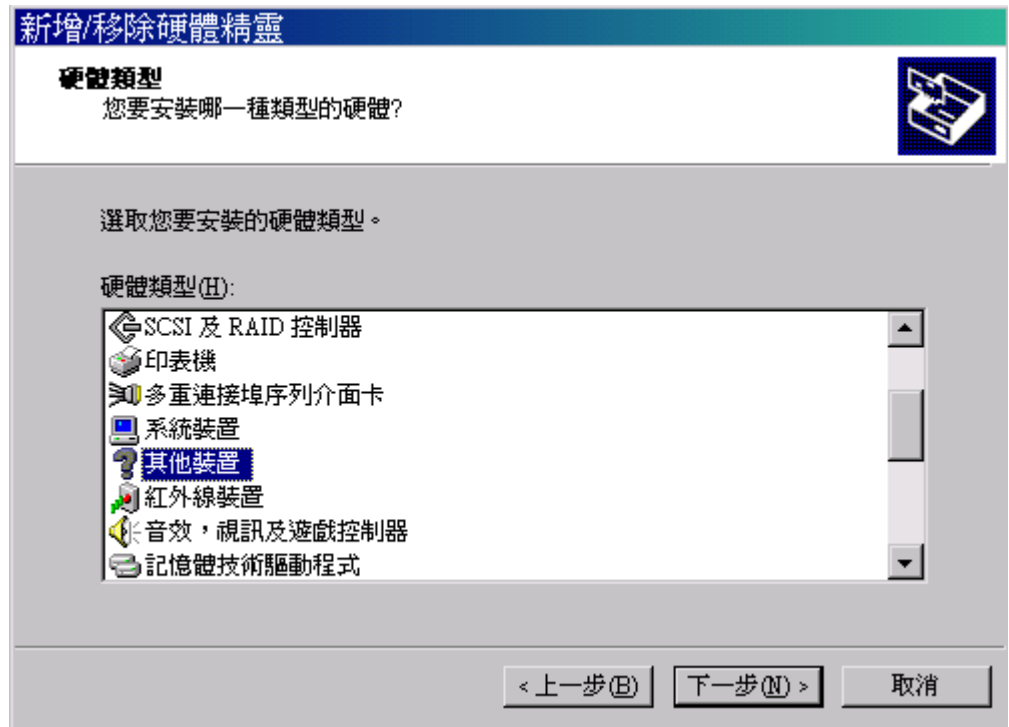
步驟五

點選從清單中選擇硬體，按下一步進入下一畫面



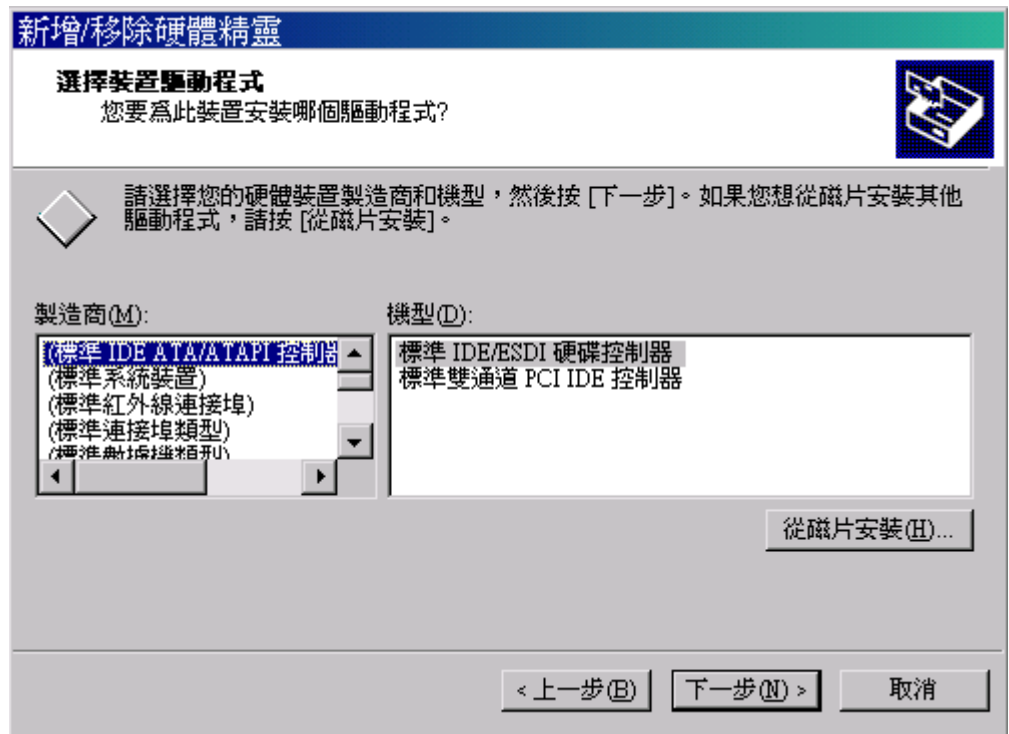
步驟六

在硬體類型中點選 **其他裝置**，按下一步進入下一畫面



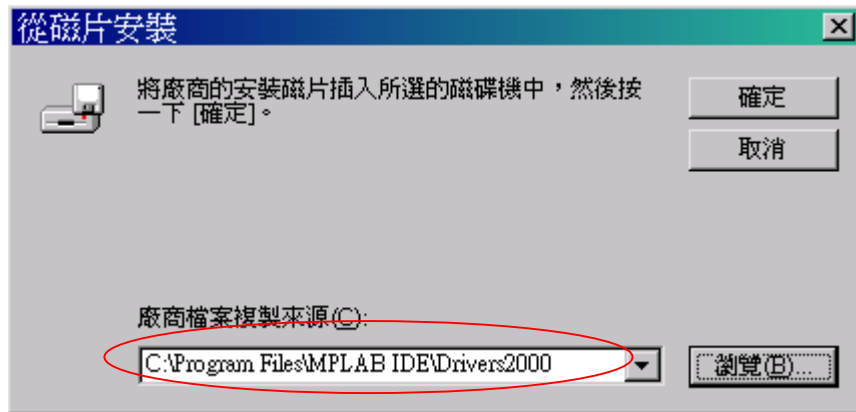
步驟七

直接點選從磁片安裝，按下一步進入下一畫面



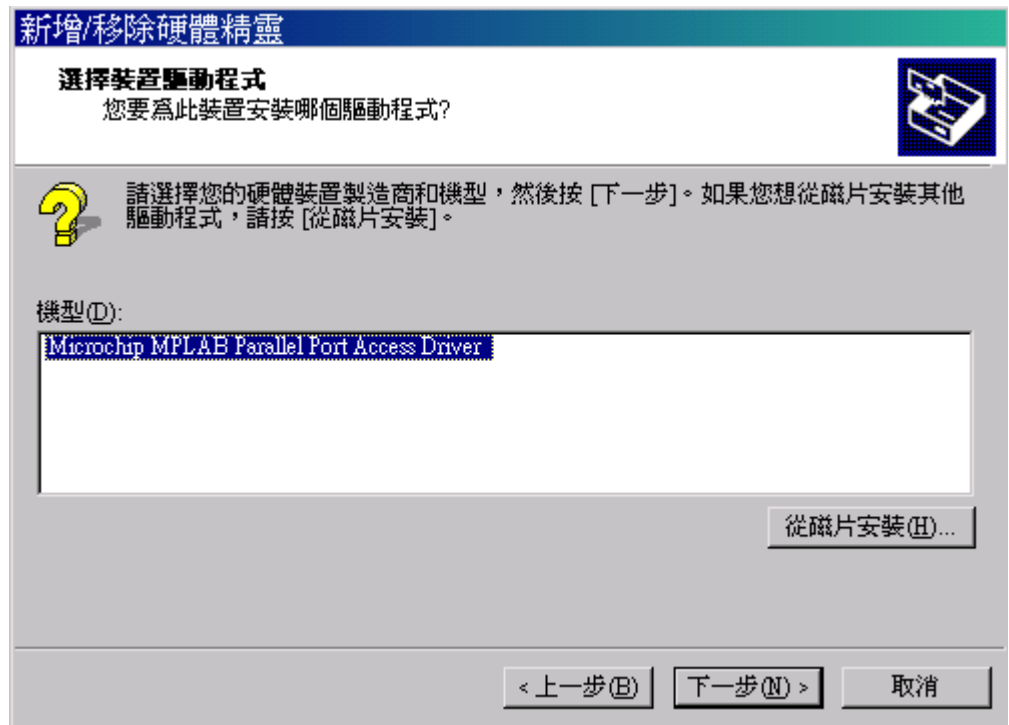
步驟八

利用瀏覽按鍵找出 ICE 2000 的驅動程式，路徑如右圖所示。按確定進入下一畫面



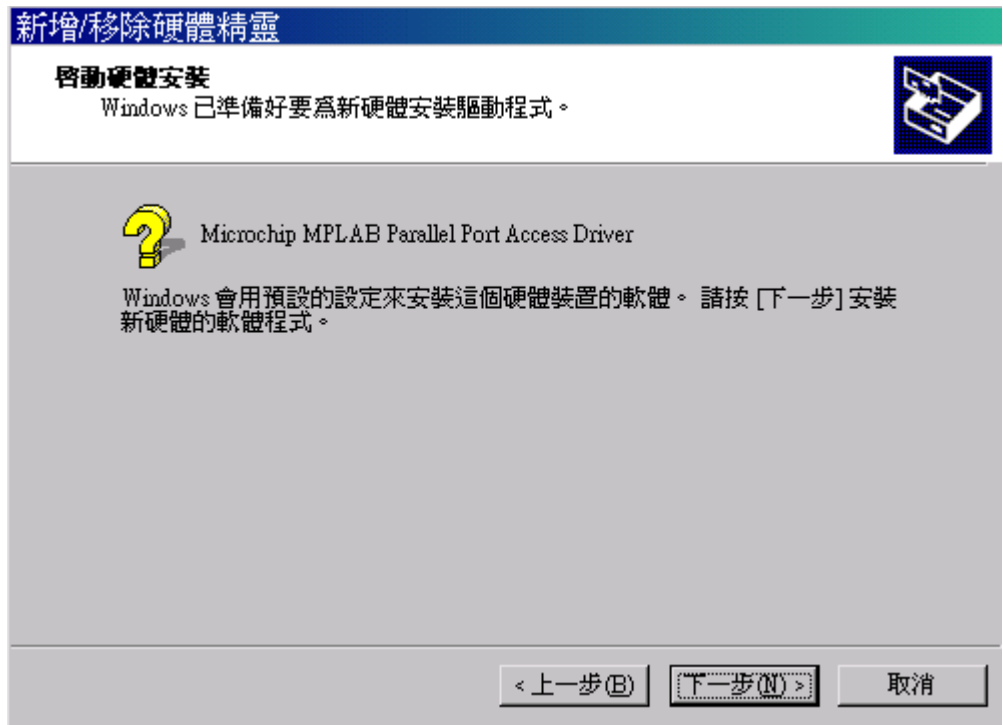
步驟九

點選 "Microchip MPLAB Parallel Port Access Driver"，按下一步進入下一畫面



步驟十

檢查驅動程式是如右所示，若有錯誤按取消鍵從來。如正確無誤點選下一步即可完成 ICE 2000 的驅動程式安裝。



步驟完成

ICE 2000 的驅動程式安裝完成，可以啟動 MPLAB IDE

