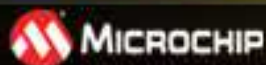


Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



使用 MPLAB X IDE

**MPLAB IDE 轉移至
MPLAB[®] X IDE**

課程目標

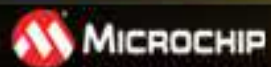
- 介紹與 **MPLAB® IDE v8.x** 的區別
- 如何設定開發工具和編譯器
- 建立和轉換專案
- 使用新的專案功能
- 編譯、燒錄和除錯
- 如何使用編輯器
- 使用專案導航工具
- 自訂介面
- 探索一些全新功能

課程安排

- 理論基礎
- 主要區別
- **MPLAB X IDE** 概述
- 新方法和新功能導覽
 - 專案、編輯器、除錯及其他
- 實驗一：建立獨立專案
- 實驗二：建立專案設定
- 實驗三：專案除錯
- 實驗四：將 **IDE v8** 專案轉換為 **X IDE** 專案
- 多個專案開發時注意事項
- 資源

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC MCU PLATFORM



理論基礎

為何做如此大的改變？

理論基礎

為何要開發這款全新的 **MPLAB® X IDE** ？

- **MPLAB v8.x** 程式架構已經過時：
 - IDE 缺少一些時尚的功能
 - 加入一些所需功能較困難
 - 協力廠商程式擴展比較困難
 - 缺少一些高階語言開發所需的基礎架構
 - 僅限用於 **Windows®** 平臺

理論基礎

為什麼採用 **NetBeans IDE** ?

- 這種新的平臺提供：
 - 經多年使用驗證成熟的 **IDE**
 - 具備所有最新功能的現代 **IDE**
 - 程式組織完善且可擴展的架構
 - 已具備所需的許多功能
 - 其架構支援高階語言
 - 可在 **Windows[®]**、**Mac** 和 **Linux** 平臺上執行

理論基礎

為什麼採用 **NetBeans IDE** ?

- 這種新的平臺提供：
 - 一個擁有豐富外掛程式的系統環境
 - 在同一 **IDE** 中進行 **PC** 和嵌入式程式開發
 - 不依賴於 **IDE** 的舊的檔案結構
 - 提供了許多之前版本的 **MPLAB® IDE** 中所沒有的新功能

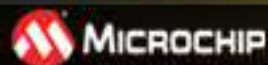
理論基礎

為什麼不採用 **Eclipse IDE** ？

- 使用 **Eclipse IDE** 程式的人實在太多了
- 我們對 **NetBeans** 有一定的經驗（**Hi-Tide**）
- 我們是 **NetBeans** 惟一的嵌入式合作夥伴
- **NetBeans** 因為以下特性獲得普遍認可：
 - 更為先進
 - 程式長度較小/速度更快
 - 更為直觀
- 欲知相關資料，請瀏覽：
<http://netbeans.org/switch/realstories.html>

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



MPLAB[®] X 中的 XC 編譯器

簡化的新型編譯器

MPLAB® XC 編譯器

基於 Hi-Tech PICC 架構

MPLAB® XC8

8-bit 編譯器

PIC10, PIC12,
PIC16, PIC18

基於 MPLAB C30 / GCC 架構

MPLAB® XC16

16-bit 編譯器

PIC24,
dsPIC30, dsPIC33

基於 MPLAB C32 / GCC 架構

MPLAB® XC32

32-bit 編譯器

PIC32

CC1 — 公共編譯器介面

免費版

免費下載使用，無使用日期限制，最基本的最佳化功能，支援所有元件

標準版

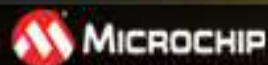
入門級價格，支援最佳化功能，獲得優先技術支援

專業版

全價格，全方位的最佳化功能，獲得優先技術支援

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



主要差異

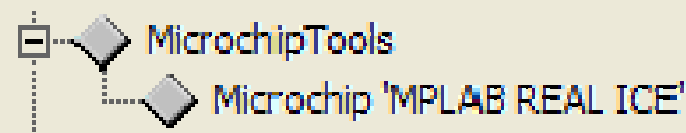
概述與 **MPLAB[®] IDE v8.x** 的功能比較

主要差異

USB 驅動程式

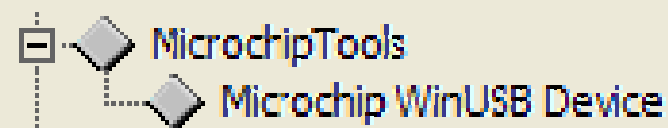
MPLAB® 8

- **ICD 3 和 REAL ICE™** 採用專屬的驅動程式
- **PICKIT™ 3** 則採用標準 **HID** 驅動程式



MPLAB® X

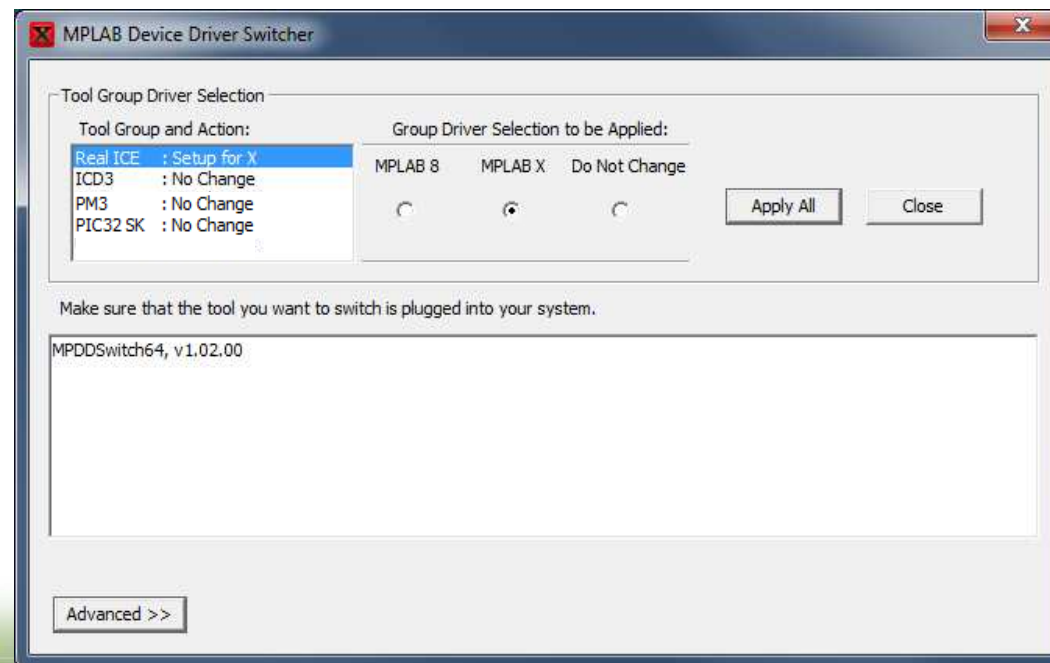
- **ICD 3 和 REAL ICE™** 採用開放式的驅動程式
 - Windows® 平臺上的 WinUSB
 - Mac 和 Linux 平臺上的 LibUSB
- **PICKIT™ 3** 採用標準 **HID** 驅動程式



主要差異

USB 驅動程式的切換

- **PC 同時安裝 MPLAB IDE 及 X IDE 者才要切換**
- **USB 驅動程式切換實用程式**
 - 允許用戶在同一台 PC 上使用 MPLAB® IDE v8.x 和 MPLAB X IDE
 - 切換驅動程式時，要確定對應的**工具必須已連接**





主要差異

與硬體工具之間的 **USB** 通信連結

MPLAB® IDE v8

- 透過選擇工具或打開已選定工具的工作區，均可建立連結
- 直到關閉 MPLAB® IDE 時**USB** 連結才會斷線
- 工具致能時， **USB** 連結 始終接通

MPLAB® X IDE

- 啟動除錯圖示時建立連結 
- 終止除錯圖示時連結斷線 
- 除非正在進行除錯，否則連結 始終斷開

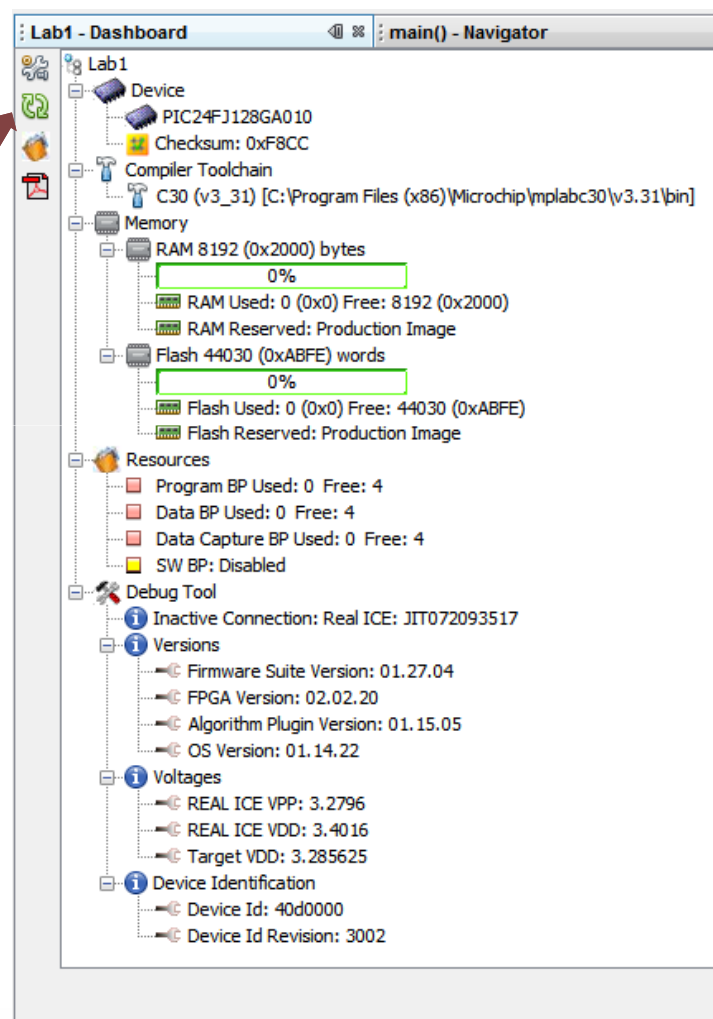
主要差異

與硬體工具之間的 **USB** 通信鏈路

不在除錯對話中
時可以刷新工具
狀態

1 從主選單打開專案儀錶板：
Windows ► Dashboard
(視窗 ► 儀錶板)

2 按一下 **Refresh
Debug Tool
Status** (刷新除錯
工具狀態) 圖示



主要差異

專案和工作區

MPLAB® v8.x

- 專案包含：
 - 基本專案設置、專案檔案和位置以及目標元件
- 工作區包含：
 - 打開的專案、視窗位置、編譯和除錯工具

MPLAB® X

- 專案專案包含：
 - 無需工作區
- 專案專案組：
 - 專案列表，便於一次打開經常要一起打開的多個相關專案

主要區別

專案檔案和結構

MPLAB® v8.x

- 由單獨的檔表示專案和工作區，必須按兩下才能打開專案或工作區：
 - *.mcp（專案）
 - *.mcw（工作區）

MPLAB® X

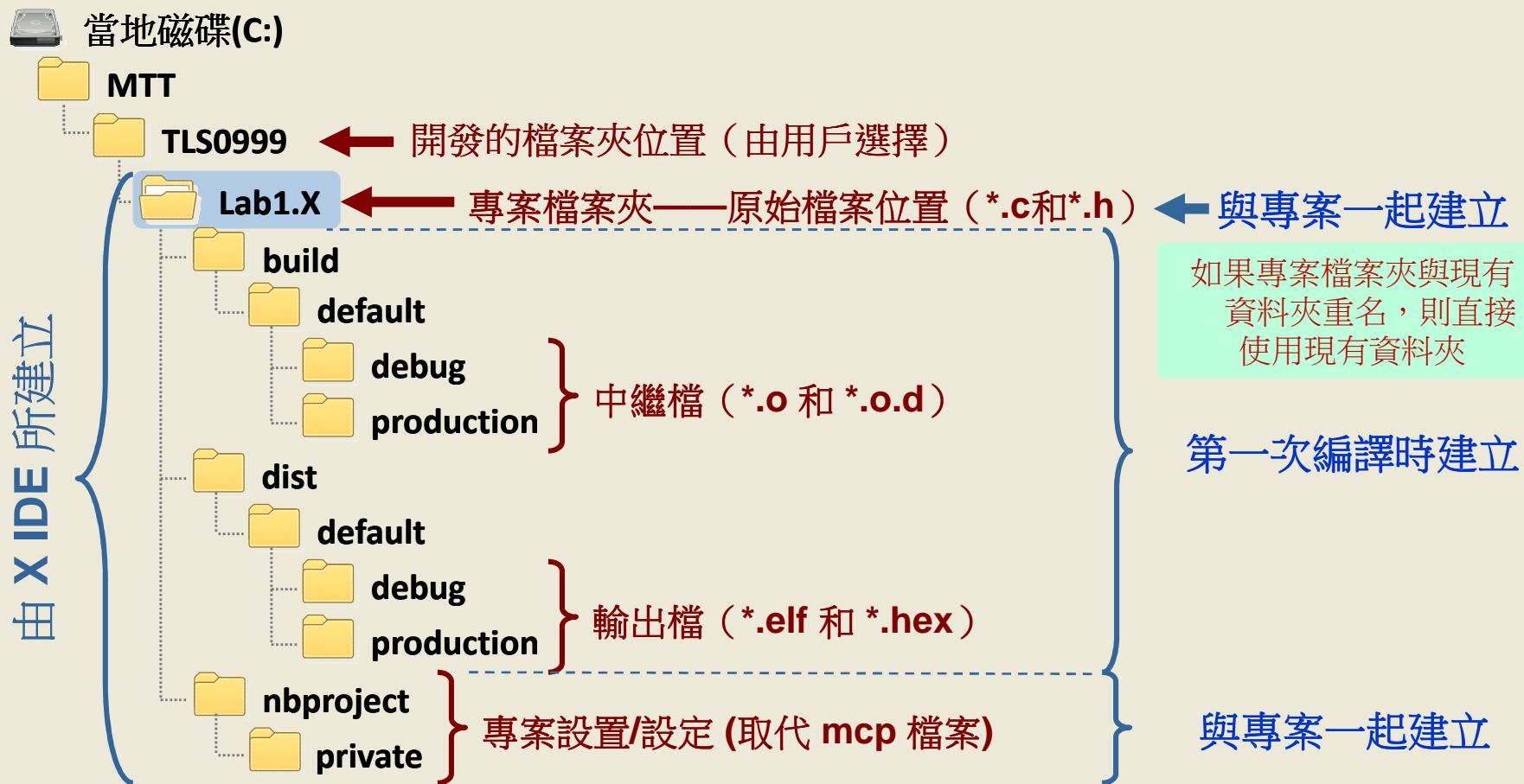
- 由一個包含 **XML** 檔的特定目錄結構來代表專案
- 沒有可按兩下的專案檔案
- 專案必須在 **IDE** 中打開

主要差異

專案檔案和結構



自動建立的專案目錄結構



主要差異

基於專案的 IDE

MPLAB® IDE v8.x

- 不是所有任務都需要專案：
 - 導入 Hex 文件
 - 快速編譯

MPLAB® X IDE

- 所有任務均需要專案：
 - **Import Hex File**（導入Hex檔） ► **Create “Prebuilt” project**（建立預編譯專案）
 - **Quick Build**（快速編譯）——在 **X IDE** 中無法實現該功能

主要差異

除錯/釋放模式 編譯設置

MPLAB® IDE v8.x

- 必須在 **Debug/Release**（除錯/釋放）下拉式列示方塊中明確地選擇所需的除錯類型



Make（編譯）

或



BuildAll（編譯所有）

MPLAB® X IDE

- 由特定的編譯命令指示 **除錯模式** 或 **釋放模式**

釋放模式



Build (Make)



Program Target



Clean and Build (Build All)

除錯模式



Debug Run Project

主要差異

新、舊開發工具支援

MPLAB® X IDE 已丕支援舊有工具

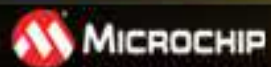
- **ICD 2 (2002)** --不支持2010年9月以後推出的新元件
- **ICE 2000 (1998)** 和 **ICE 4000 (2003)**
- **PRO MATE II (1996)**
- **PICSTART® Plus (1997)**

MPLAB® X IDE — 目前所支援的工具

- **ICD 3 (2008)** 、 **PICkit™ 3 (2008)** 和 **PICkit 2 (2005)**
- **REAL ICE™ (2006)**
- **MPLAB PM3 (2004)**
- 入門工具套件.....

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



概述

MPLAB[®] X IDE 組件概覽

安裝 MPLAB X IDE & XC8



Microchip 產品、教育訓練及實驗板資料

Nov. 30, 2012 Ver.2.0.0.0

[Microchip Focus Product Selector Guide \(February 2012\)](#)

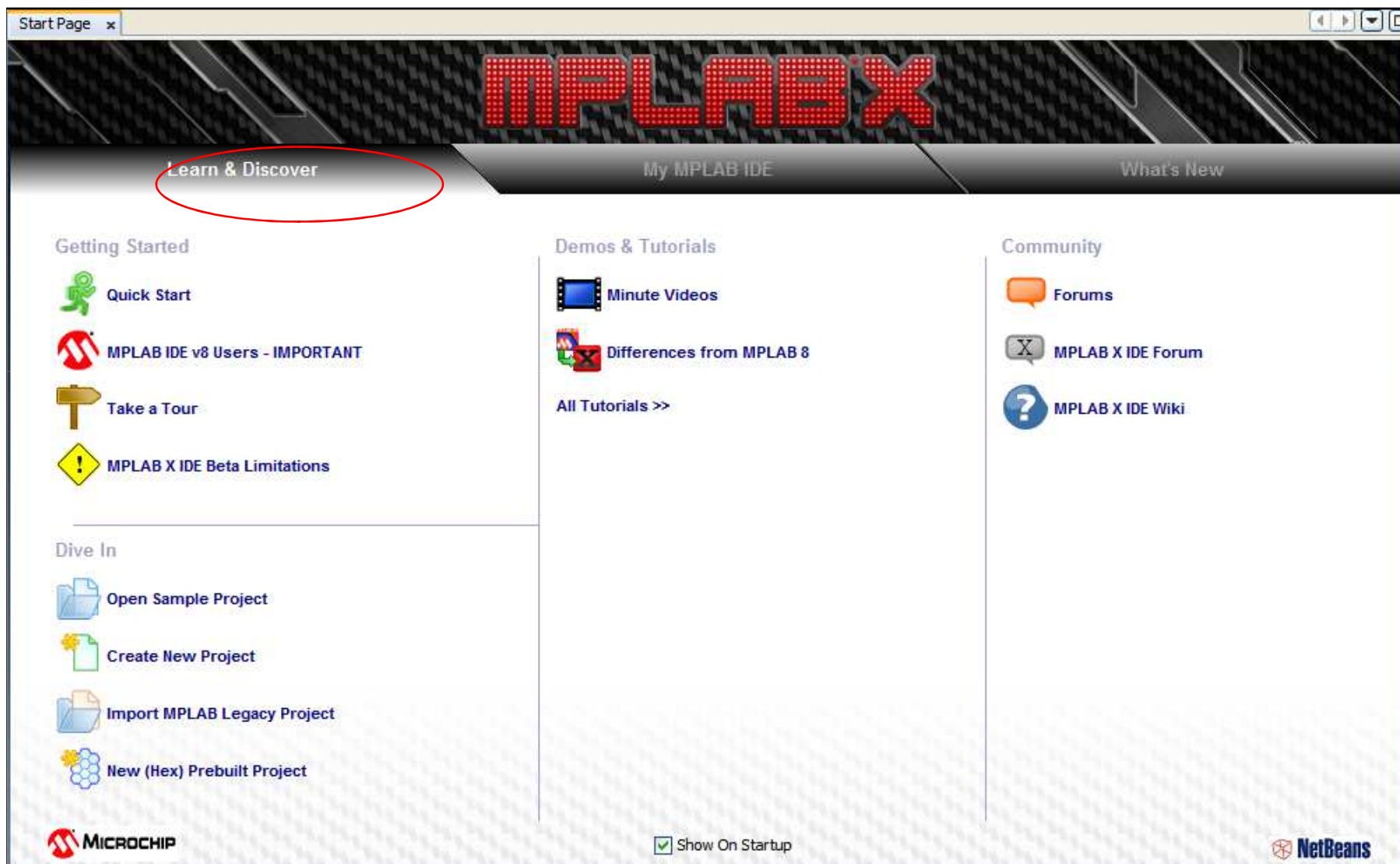
開發軟體, 編譯器 (www.microchip.com/developmenttools)

MPLAB X IDE	v1.50Windows(Local) v1.41Windows(Local) Windows Version Linux Version Mac Version Detail Info.
MPLAB IDE	v8.88(Local)
MPLAB XC8	v1.11(Local) v1.10(Local)
MPLAB XC16	v1.10(Local)
MPLAB XC32	v1.11a(local) v1.10(Local)
MPLAB C18 Lite	v3.44(Local) v3.43(Local)
MPLAB C30 Lite	v3.31(Local)
MPLAB C32 Lite	v2.02a(Local)
HLTECH C for PIC10/12/16	v9.83(Local)

- 連上 Microchip 台灣網站後點選右邊“教育訓練光碟”即可開啟上面的畫面。
- 開啟 RTC 上課光碟也可以或直接到 Microchip 美國網站下載安裝。

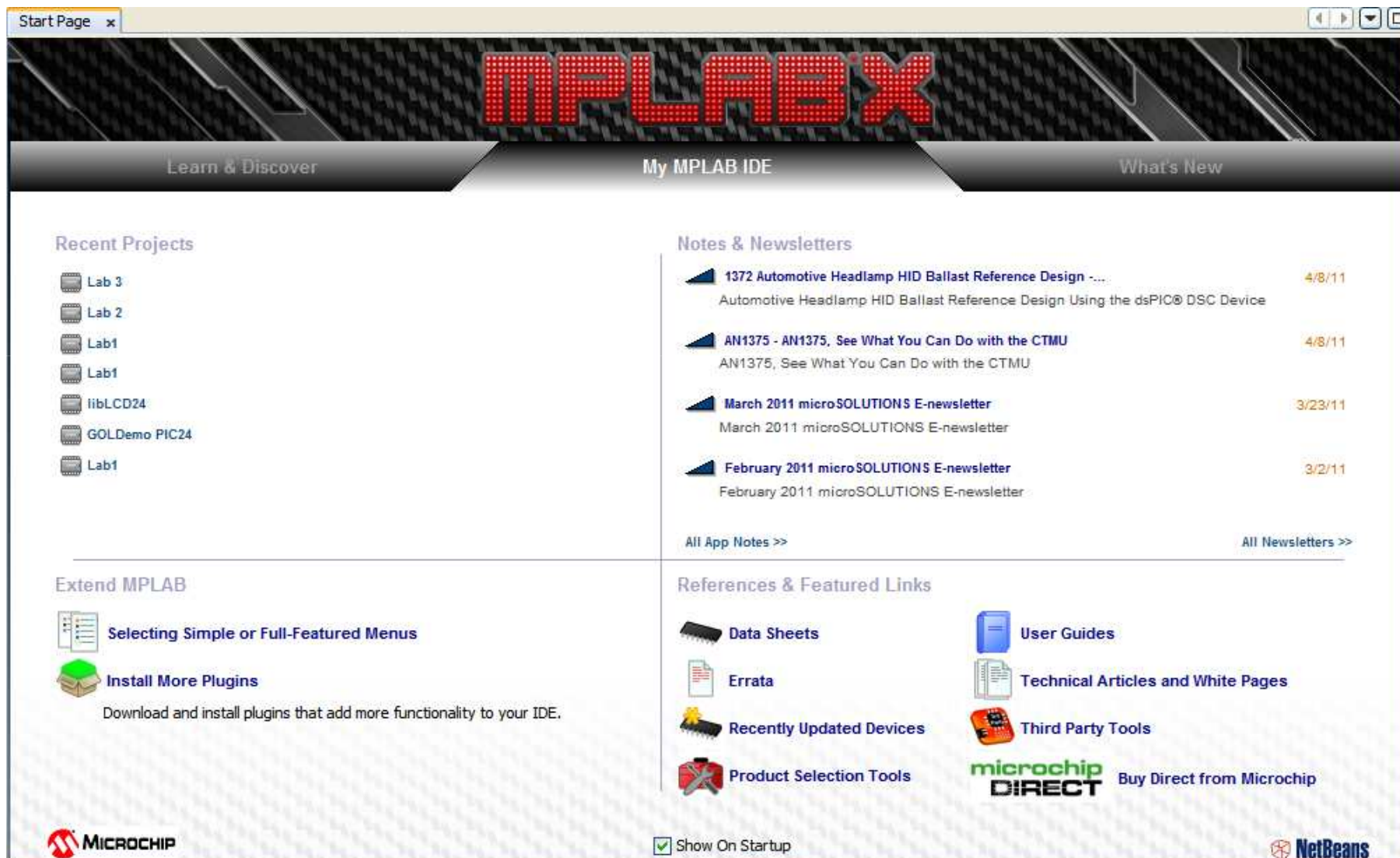
啟動頁面

Learn & Discover (學習和探索) 選項卡



啟動頁面

MyMPLAB® IDE 選項卡 (需上網連接)



The screenshot shows the MyMPLAB IDE Start Page. At the top, there is a navigation bar with three tabs: "Learn & Discover", "My MPLAB IDE" (which is selected), and "What's New". Below the navigation bar, the page is divided into four main sections:

- Recent Projects:** A list of recent projects including "Lab 3", "Lab 2", "Lab1", "LibLCD24", "GOLDemo PIC24", and "Lab1".
- Notes & Newsletters:** A list of notes and newsletters, including "1372 Automotive Headlamp HID Ballast Reference Design", "AN1375 - AN1375, See What You Can Do with the CTMU", "March 2011 microSOLUTIONS E-newsletter", and "February 2011 microSOLUTIONS E-newsletter".
- Extend MPLAB:** A section for extending the IDE with plugins, including "Selecting Simple or Full-Featured Menus" and "Install More Plugins".
- References & Featured Links:** A section for references and featured links, including "Data Sheets", "Errata", "Recently Updated Devices", "Product Selection Tools", "User Guides", "Technical Articles and White Pages", "Third Party Tools", and "microchip DIRECT Buy Direct from Microchip".

At the bottom of the page, there is a "Show On Startup" checkbox and the NetBeans logo.

啟動頁面

What's New (新功能) 選項卡 (需上網連接)



Start Page x

MPLAB X

Learn & Discover My MPLAB IDE What's New

Data Sheets & Errata

-  [PIC24FV32KA304 Family Data Sheet](#) 4/8/11
PIC24FV32KA304 Family Data Sheet
-  [PIC18F87J72 Single-Phase Energy Meter Reference Design ...](#) 4/8/11
PIC18F87J72 Single-Phase Energy Meter Reference Design User's Guide
-  [PIC18F87K22 Family Silicon Errata and Data Sheet Clarif...](#) 4/7/11
PIC18F87K22 Family Silicon Errata and Data Sheet Clarification
-  [PIC18F87K90 Family Silicon/Data Sheet Errata](#) 4/7/11
PIC18F87K90 Family Silicon/Data Sheet Errata

[All Data Sheets >>](#) [All Errata >>](#)

Reference Manuals & Programming Specs

-  [PIC24FXXKAXXX Programming Specification](#) 4/8/11
PIC24FXXKAXXX Programming Specification
-  [PIC24FJXXXGA1/GB1 Families Flash Programming Specificat...](#) 4/5/11
PIC24FJXXXGA1/GB1 Families Flash Programming Specification
-  [PIC24F Family Ref. Manual, Sect. 51 12-Bit A/D Converte...](#) 3/22/11
PIC24F Family Ref. Manual, Sect. 51 12-Bit A/D Converter with Threshold Detect
-  [PIC24F Family Reference Manual, Sect. 41 32-Bit Program...](#) 3/10/11
PIC24F Family Reference Manual, Sect. 41 32-Bit Programmable CRC

Recently Released Software

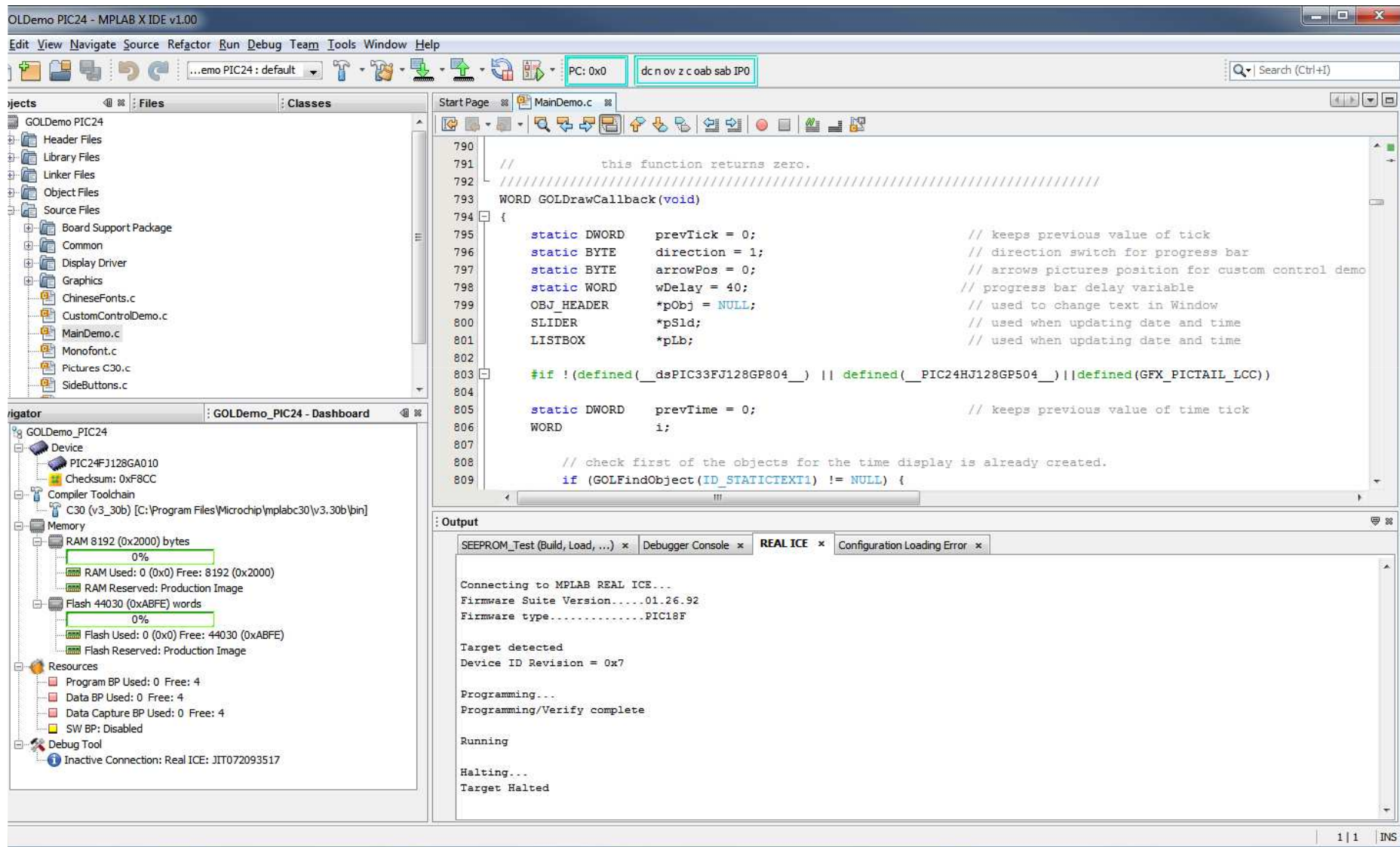
-  [PIC24H \(dsPIC33F\) Capacitive Touch Evaluation Kit Softw...](#) 4/5/11
PIC24H (dsPIC33F) Capacitive Touch Evaluation Kit Software
-  [LIN Serial Analyzer Source Code](#) 4/4/11
LIN Serial Analyzer Source Code
-  [PIC32 ESK TCP/IP Demo](#) 3/30/11
PIC32 ESK TCP/IP Demo
-  [MPLAB IDE Current Release Notes](#) 3/25/11
MPLAB IDE Current Release Notes

[All Recently Released Software >>](#)

Product & Corporate News

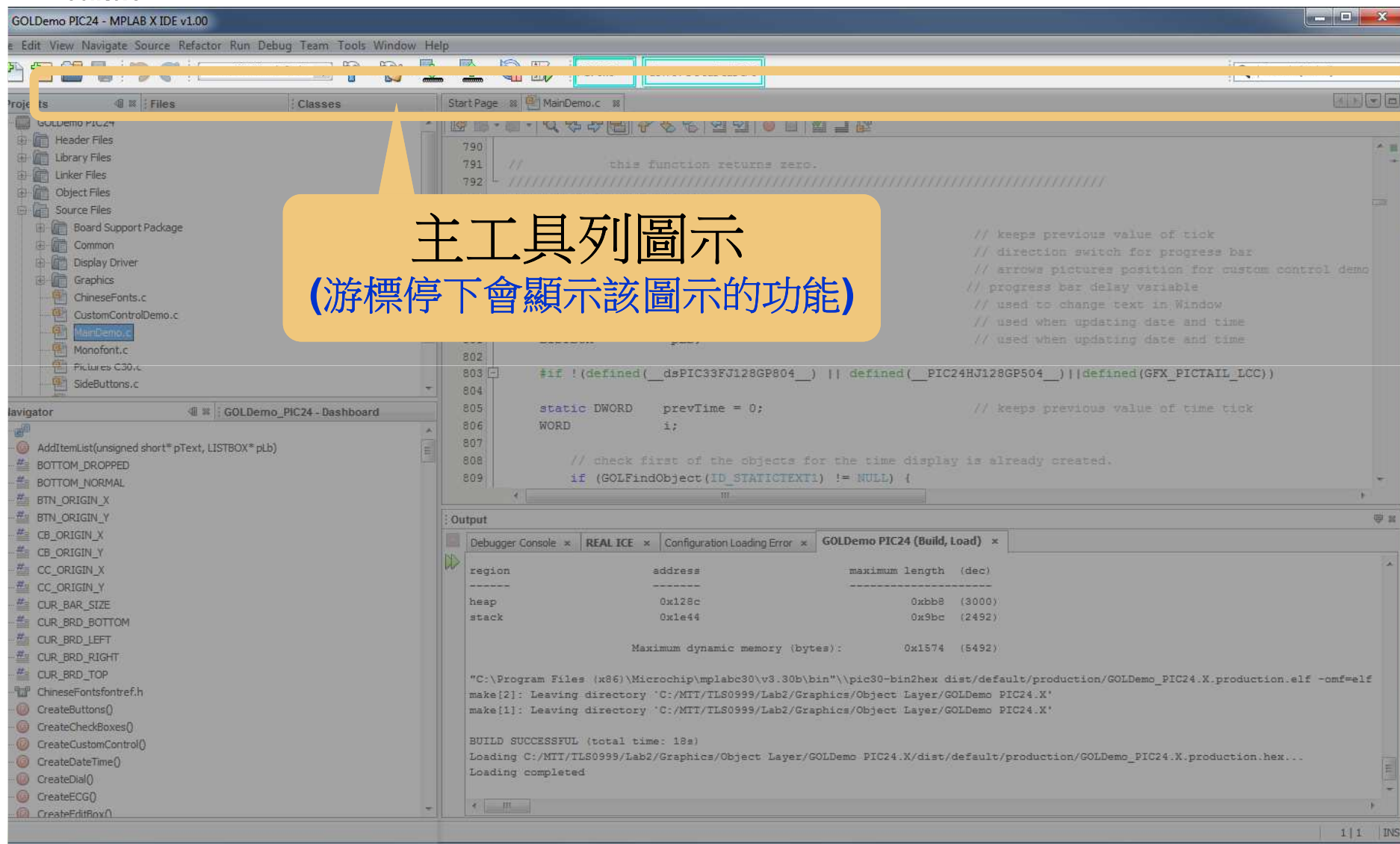
-  [Microchip Expands Stand-Alone Real-Time Clock/Calendar ...](#) 4/11/11
Microchip Expands Stand-Alone Real-Time Clock/Calendar Family
-  [Microchip Expands MOSFET Driver Family](#) 4/5/11
Microchip Expands MOSFET Driver Family
-  [Media Invited to Attend the FIRST Robotics Competition ...](#) 3/15/11
Media Invited to Attend the FIRST Robotics Competition Arizona Regional
-  [Microchip Receives Top Honors for Business Excellence](#) 1/25/11
Microchip Receives Top Honors for Business Excellence

MPLAB X IDE 的版面配置



X IDE 版面配置

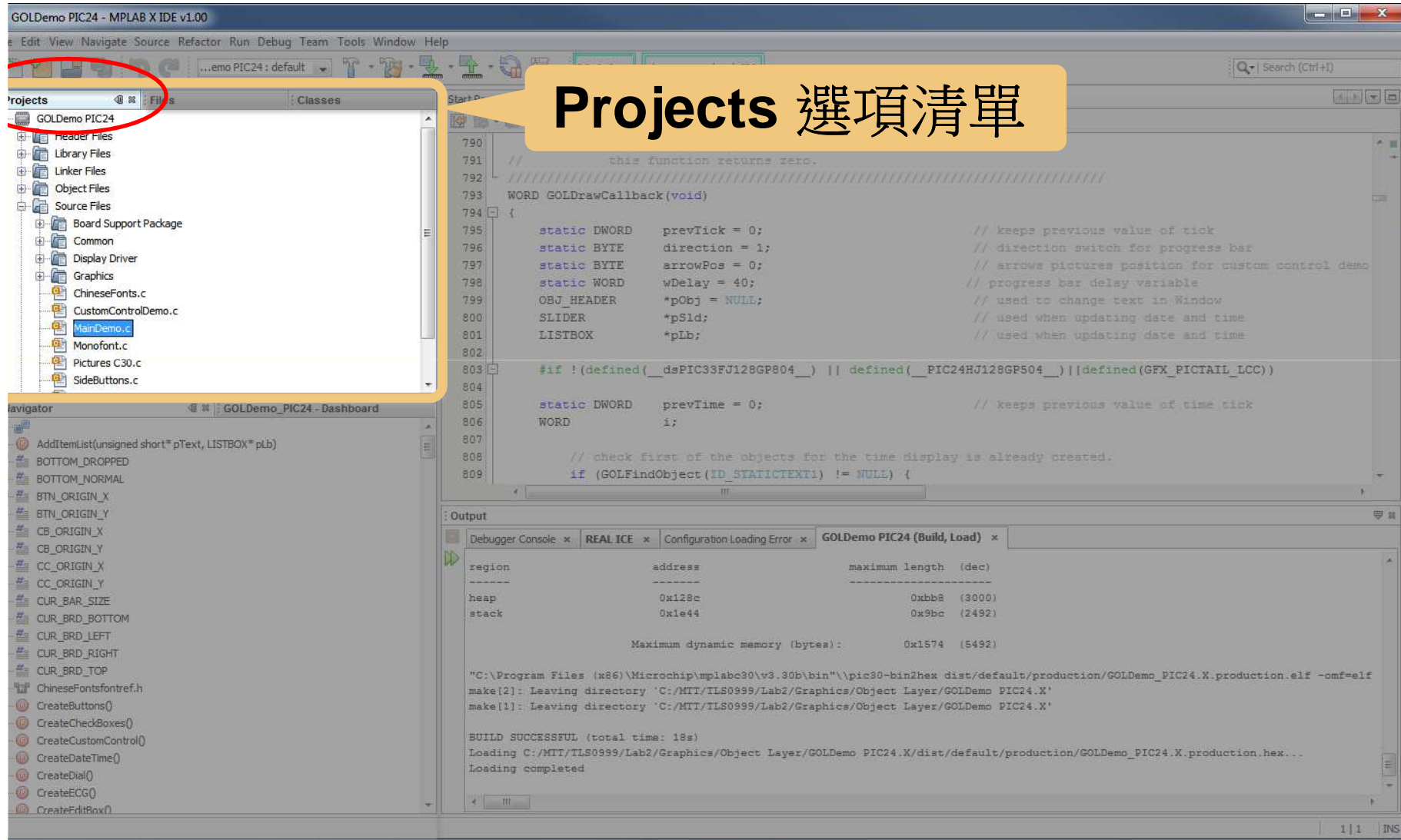
主工具列



X IDE 佈局

Projects (專案) 清單 (Project Tree)

Projects 選項清單



The screenshot displays the MPLAB X IDE interface. The 'Projects' window on the left shows a tree view of the project 'GOLDemo PIC24'. The 'Main' window shows the source code for 'MainDemo.c'. The 'Output' window at the bottom shows the build process for 'GOLDemo PIC24'.

Output:

region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)

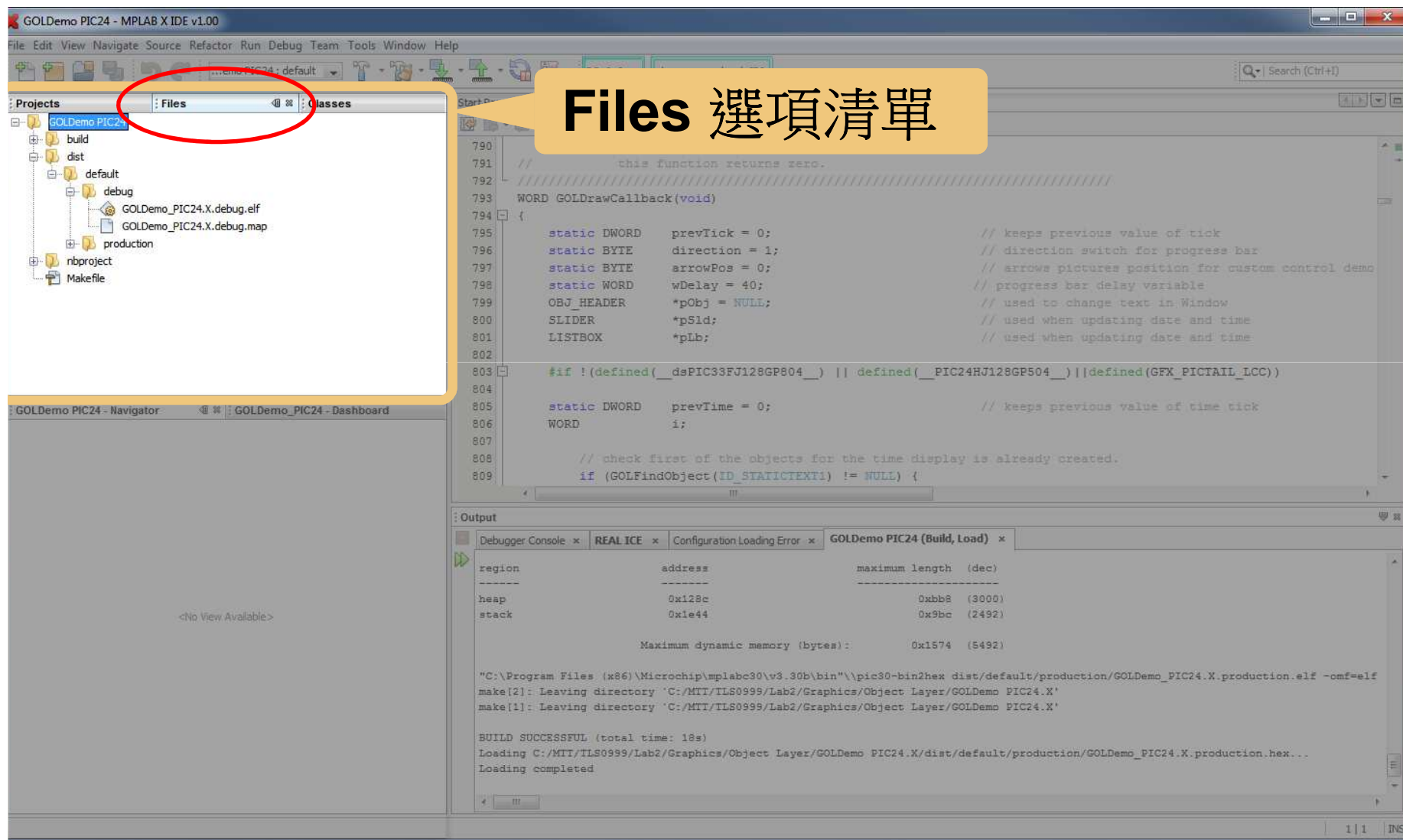
Maximum dynamic memory (bytes): 0x1574 (5492)

"C:\Program Files (x86)\Microchip\mplabx30\bin\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -cmf=elf
make[2]: Leaving directory 'C:/MIT/TL0999/Lab2/Graphics/Object Layer/GOLDemo_PIC24.X'
make[1]: Leaving directory 'C:/MIT/TL0999/Lab2/Graphics/Object Layer/GOLDemo_PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MIT/TL0999/Lab2/Graphics/Object Layer/GOLDemo_PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed

X IDE 版面配置

Files (檔案) 清單



Files 選項清單

```
790
791 // this function returns zero.
792 ///////////////////////////////////////////////////////////////////
793 WORD GOLDrawCallback(void)
794 {
795     static DWORD    prevTick = 0;           // keeps previous value of tick
796     static BYTE     direction = 1;         // direction switch for progress bar
797     static BYTE     arrowPos = 0;          // arrows pictures position for custom control demo
798     static WORD     wDelay = 40;           // progress bar delay variable
799     OBJ_HEADER      *pObj = NULL;         // used to change text in Window
800     SLIDER           *pSlid;              // used when updating date and time
801     LISTBOX          *pLb;                // used when updating date and time
802
803     #if !(defined(_dsPIC33FJ128GP804_) || defined(_PIC24HJ128GP504_) || defined(GFX_PICTAIL_LCC))
804
805     static DWORD    prevTime = 0;           // keeps previous value of time tick
806     WORD            i;
807
808     // check first of the objects for the time display is already created.
809     if (GOLFindObject(ID_STATICTEXT1) != NULL) {
```

Output

region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)

Maximum dynamic memory (bytes): 0x1574 (5492)

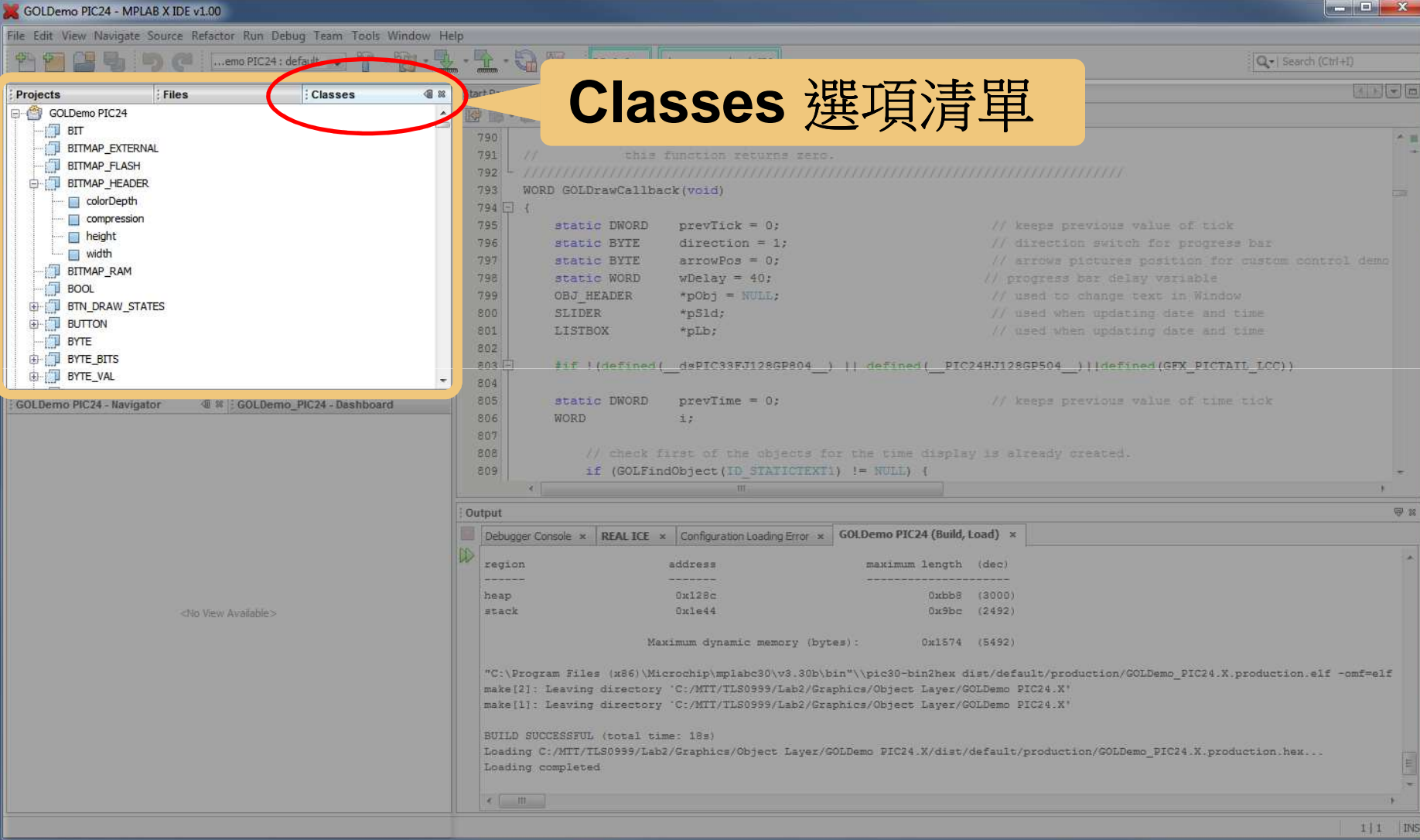
"C:\Program Files (x86)\Microchip\mplabc30\v3.30b\bin"\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -cmf=elf
make[2]: Leaving directory 'C:/MIT/TL80999/Lab2/Graphics/Object Layer/GOLDemo_PIC24.X'
make[1]: Leaving directory 'C:/MIT/TL80999/Lab2/Graphics/Object Layer/GOLDemo_PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MIT/TL80999/Lab2/Graphics/Object Layer/GOLDemo_PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed

X IDE 版面配置

Classes (類別) 清單

Classes 選項清單



The screenshot shows the MPLAB X IDE interface. The 'Classes' tab in the 'Projects' window is highlighted with a red circle. The 'Classes' list includes:

- BIT
- BITMAP_EXTERNAL
- BITMAP_FLASH
- BITMAP_HEADER
- colorDepth
- compression
- height
- width
- BITMAP_RAM
- BOOL
- BTN_DRAW_STATES
- BUTTON
- BYTE
- BYTE_BITS
- BYTE_VAL

The main editor displays C code for a function named `GOLDDrawCallback`. The code includes static variables for `prevTick`, `direction`, `arrowPos`, `wDelay`, `OBJ_HEADER`, `pObj`, `pSld`, and `pLb`. It also includes a check for the existence of the `ID_STATICTEXT1` object.

The Output window shows the build process for `GOLDemo PIC24`. The output includes the following table:

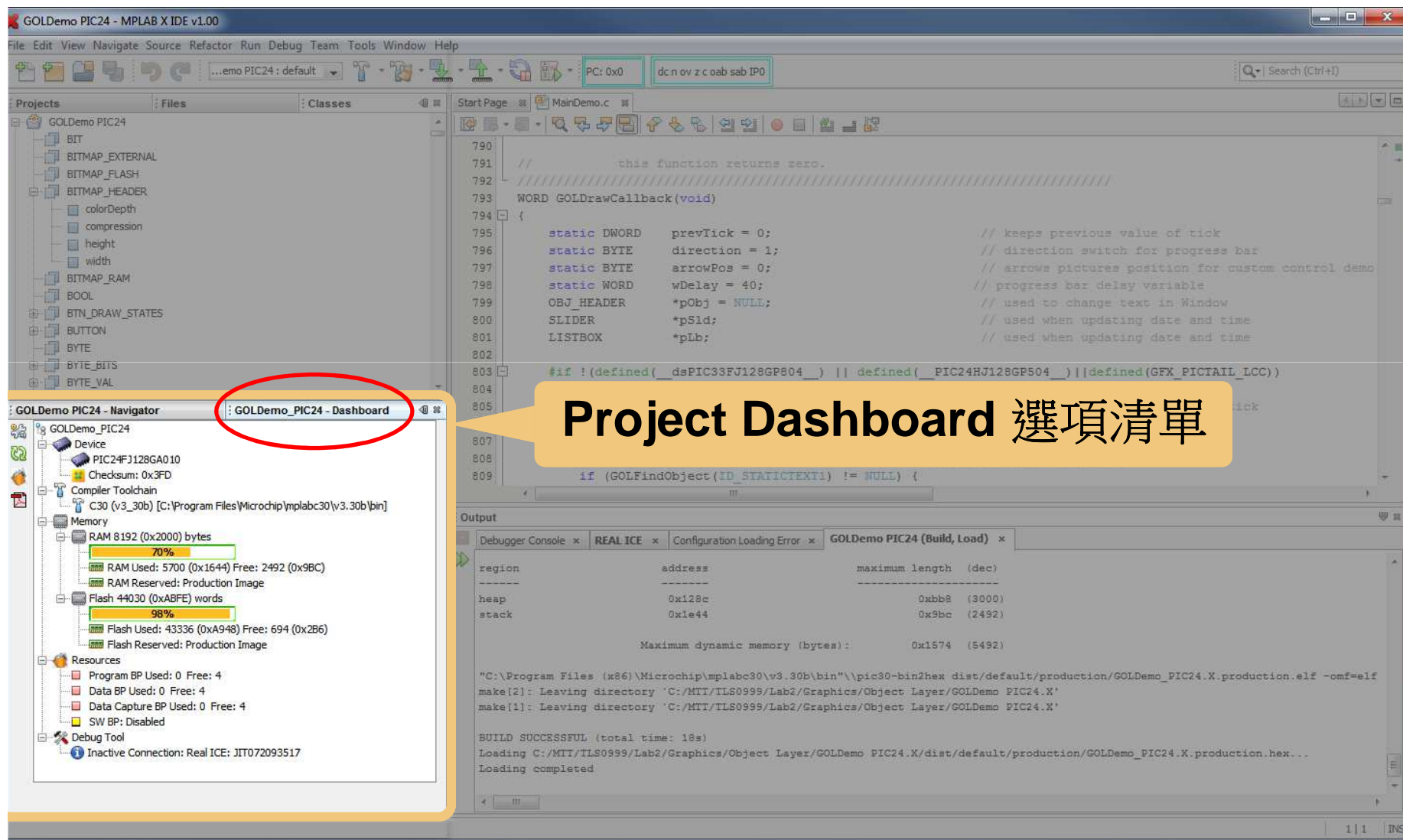
region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)

Maximum dynamic memory (bytes): 0x1574 (5492)

The build process is successful, and the output file is `GOLDemo_PIC24.X.production.hex`.

X IDE 版面配置

Project Dashboard (專案儀錶板) 清單



The screenshot shows the MPLAB X IDE interface with the Project Dashboard window open. The dashboard displays various project settings and resource usage. A red circle highlights the 'GOLDemo_PIC24 - Dashboard' tab, and a yellow callout box points to it with the text 'Project Dashboard 選項清單'.

Project Dashboard 選項清單

- Device: PIC24FJ128GA010
- Checksum: 0x3FD
- Compiler Toolchain: C30 (v3_30b) [C:\Program Files\Microchip\mplabc30\v3.30b\bin]
- Memory:
 - RAM 8192 (0x2000) bytes: 70%
 - RAM Used: 5700 (0x1644) Free: 2492 (0x9BC)
 - RAM Reserved: Production Image
 - Flash 44030 (0xABFE) words: 98%
 - Flash Used: 43336 (0xA948) Free: 694 (0x2B6)
 - Flash Reserved: Production Image
- Resources:
 - Program BP Used: 0 Free: 4
 - Data BP Used: 0 Free: 4
 - Data Capture BP Used: 0 Free: 4
 - SW BP: Disabled
- Debug Tool: Inactive Connection: Real ICE: J1T072093517

The main editor window shows the source code for 'MainDemo.c' with the following code snippet:

```

790 // this function returns zero.
791 ///////////////////////////////////////////////////////////////////
792 WORD GOLDrawCallback(void)
793 {
794     static DWORD    prevTick = 0;           // keeps previous value of tick
795     static BYTE     direction = 1;         // direction switch for progress bar
796     static BYTE     arrowPos = 0;          // arrows pictures position for custom control demo
797     static WORD     wDelay = 40;           // progress bar delay variable
798     OBJ_HEADER      *pObj = NULL;         // used to change text in Window
799     SLIDER           *pSlid;               // used when updating date and time
800     LISTBOX          *pLb;                // used when updating date and time
801
802     #if !(defined(_dsPIC33FJ128GP804_) || defined(_PIC24HJ128GP504_) || defined(GFX_PICTAIL_LCC))
803
804     if (GOLFindObject(ID_STATICTEXT1) != NULL) {
805
806     }
807
808     if (GOLFindObject(ID_STATICTEXT1) != NULL) {
809
810     }
811
812     }
813
814     }
815
816     }
817
818     }
819
820     }
821
822     }
823
824     }
825
826     }
827
828     }
829
830     }
831
832     }
833
834     }
835
836     }
837
838     }
839
840     }
841
842     }
843
844     }
845
846     }
847
848     }
849
850     }
851
852     }
853
854     }
855
856     }
857
858     }
859
860     }
861
862     }
863
864     }
865
866     }
867
868     }
869
870     }
871
872     }
873
874     }
875
876     }
877
878     }
879
880     }
881
882     }
883
884     }
885
886     }
887
888     }
889
890     }
891
892     }
893
894     }
895
896     }
897
898     }
899
900     }
901
902     }
903
904     }
905
906     }
907
908     }
909
910     }
911
912     }
913
914     }
915
916     }
917
918     }
919
920     }
921
922     }
923
924     }
925
926     }
927
928     }
929
930     }
931
932     }
933
934     }
935
936     }
937
938     }
939
940     }
941
942     }
943
944     }
945
946     }
947
948     }
949
950     }
951
952     }
953
954     }
955
956     }
957
958     }
959
960     }
961
962     }
963
964     }
965
966     }
967
968     }
969
970     }
971
972     }
973
974     }
975
976     }
977
978     }
979
980     }
981
982     }
983
984     }
985
986     }
987
988     }
989
990     }
991
992     }
993
994     }
995
996     }
997
998     }
999
1000    }
  
```

The Output window shows the following text:

```

region          address          maximum length (dec)
-----
heap            0x128c             0xbb8 (3000)
stack           0x1e44             0x9bc (2492)

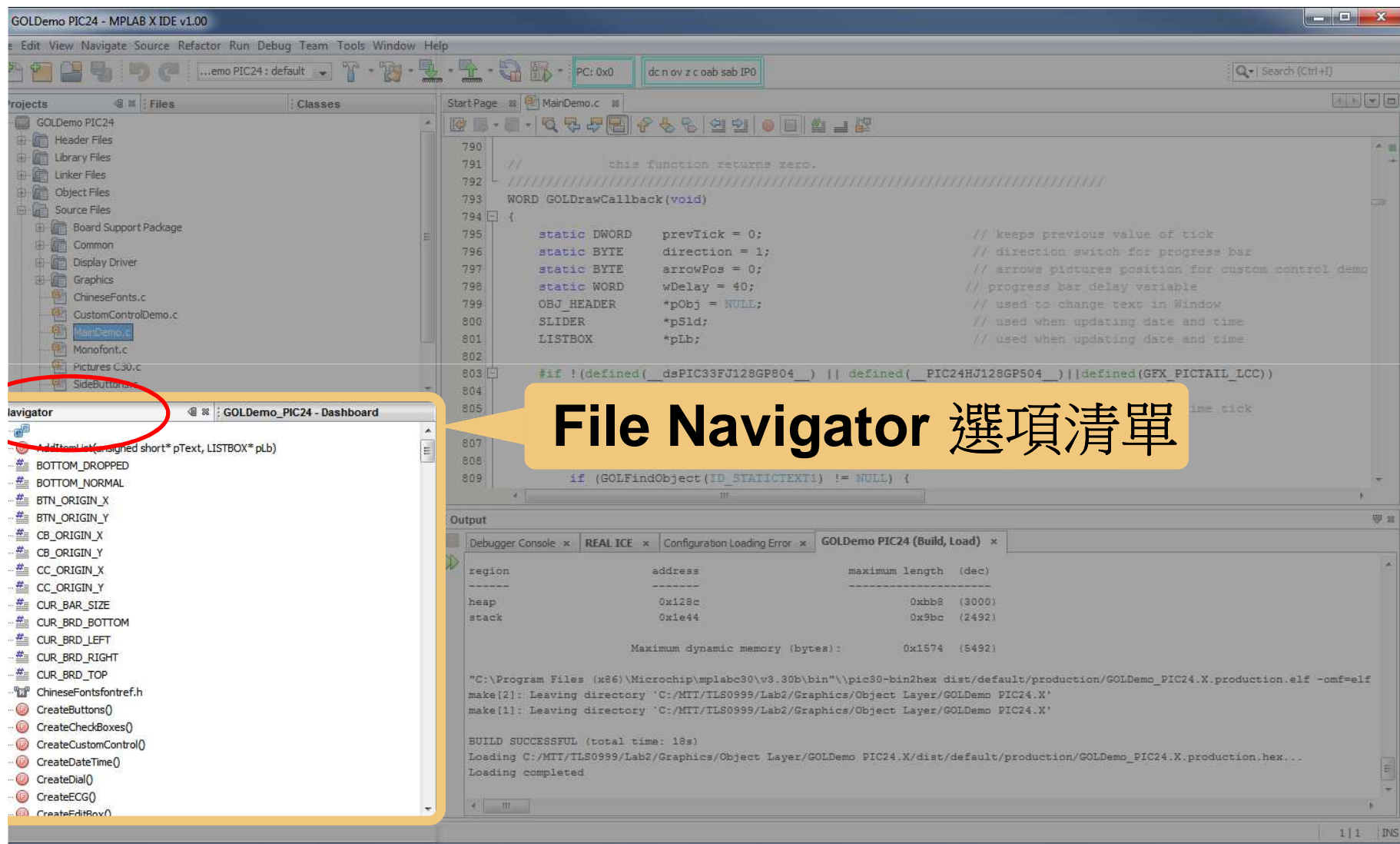
Maximum dynamic memory (bytes): 0x1574 (5492)

"C:\Program Files (x86)\Microchip\mplabc30\v3.30b\bin"\pic30-bin2hex dist/default/production/GOLDemo_PIC24.X.production.elf -cmf=elf
make[2]: Leaving directory 'C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo_PIC24.X'
make[1]: Leaving directory 'C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo_PIC24.X'

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MIT/ILS0999/Lab2/Graphics/Object Layer/GOLDemo_PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed
  
```

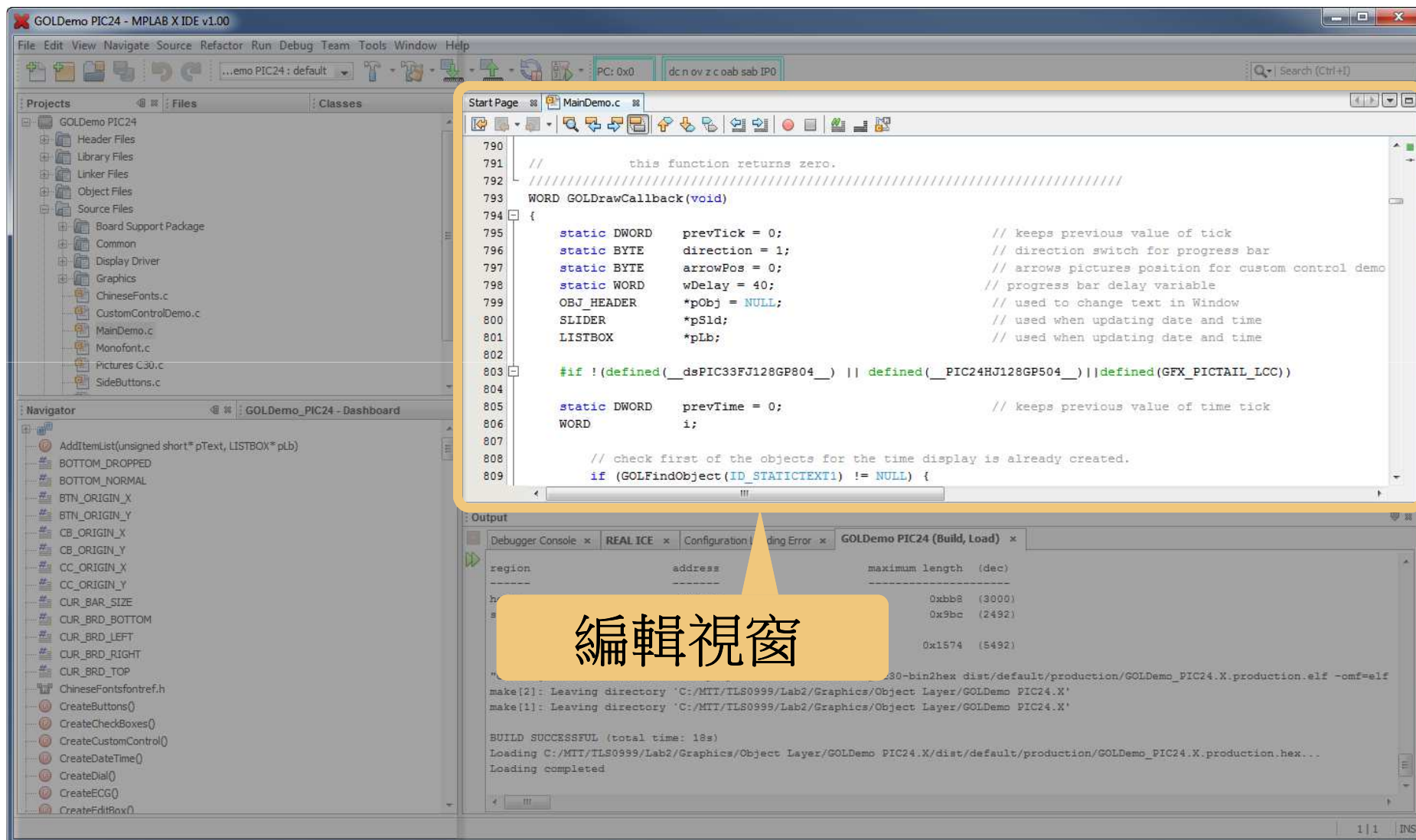

X IDE 版面配置

Navigator (導航器)



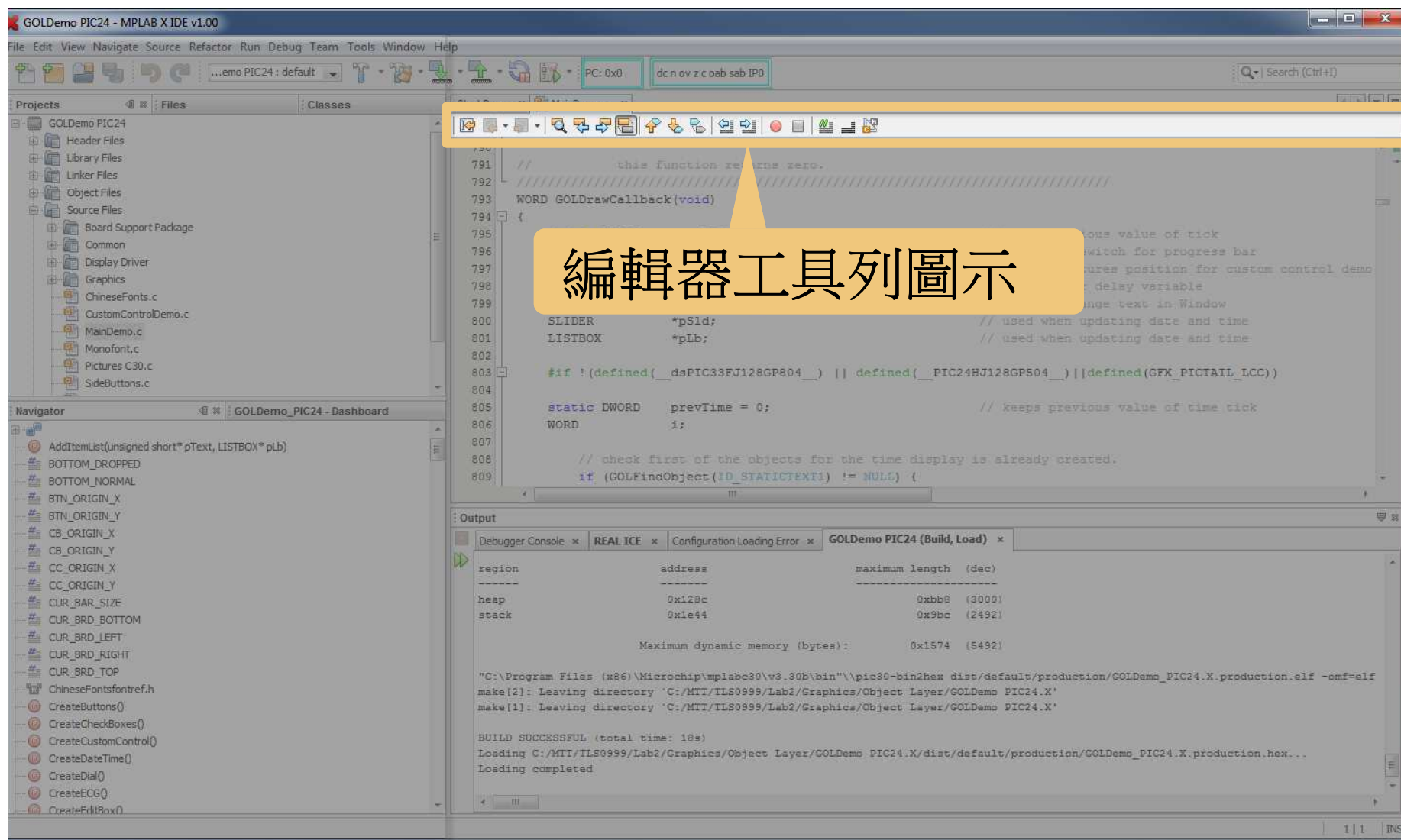
X IDE 版面配置

程式編輯器

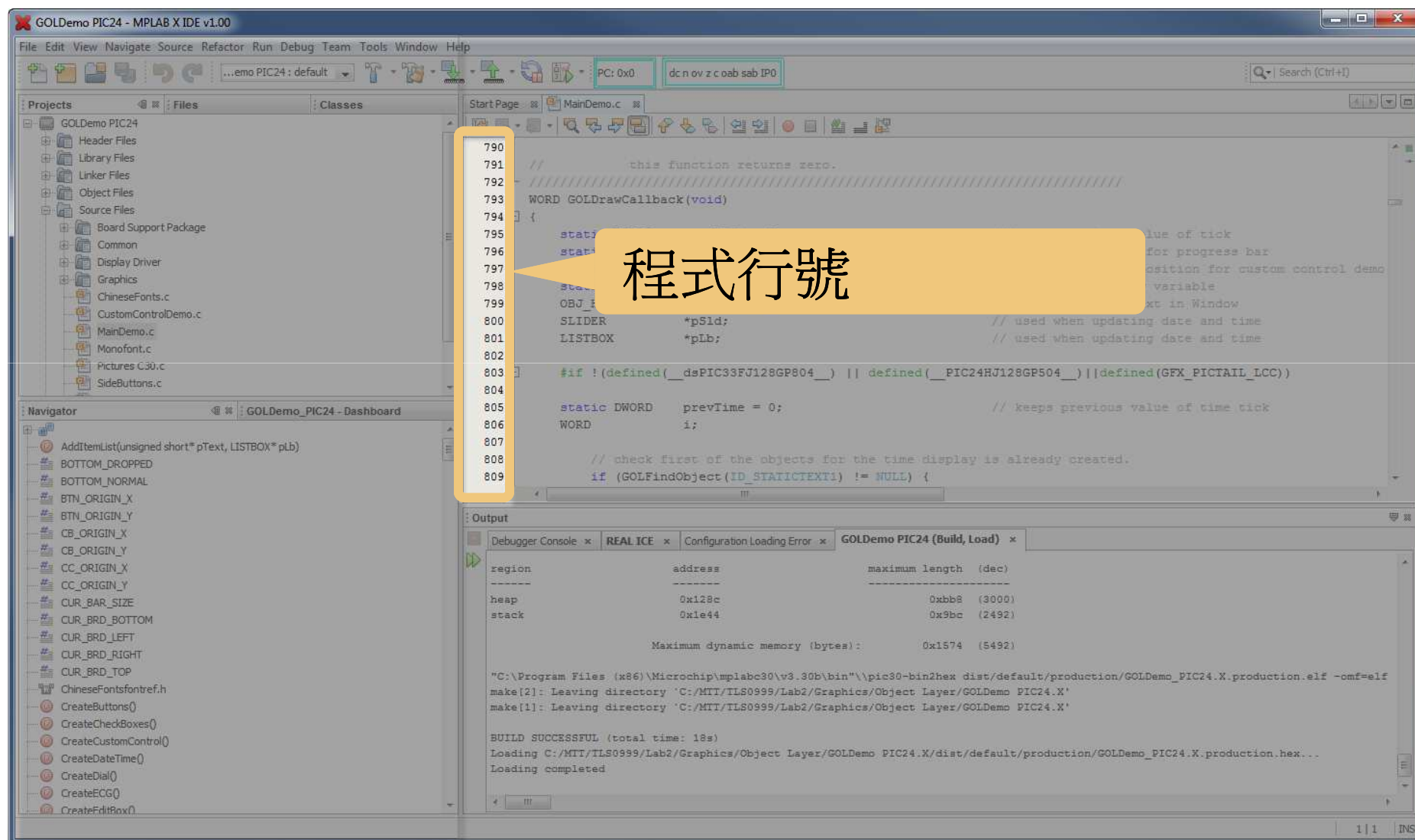


X IDE 版面配置

編輯器工具列



X IDE 版面配置 行號



程式行號

```

790
791 // this function returns zero.
792 ///////////////////////////////////////////////////////////////////
793 WORD GOLDrawCallback(void)
794 {
795     static DWORD prevTime = 0; // keeps previous value of time tick
796     static WORD i;
797     // check first of the objects for the time display is already created.
798     if (GOLFindObject(ID_STATICTEXT1) != NULL) {
799         OBJ_
800         SLIDER *pSlid; // used when updating date and time
801         LISTBOX *pLb; // used when updating date and time
802
803         #if !(defined(_dsPIC33FJ128GP804_) || defined(_PIC24HJ128GP504_) || defined(GFX_PICTAIL_LCC))
804
805         static DWORD prevTime = 0; // keeps previous value of time tick
806         WORD i;
807
808         // check first of the objects for the time display is already created.
809         if (GOLFindObject(ID_STATICTEXT1) != NULL) {

```

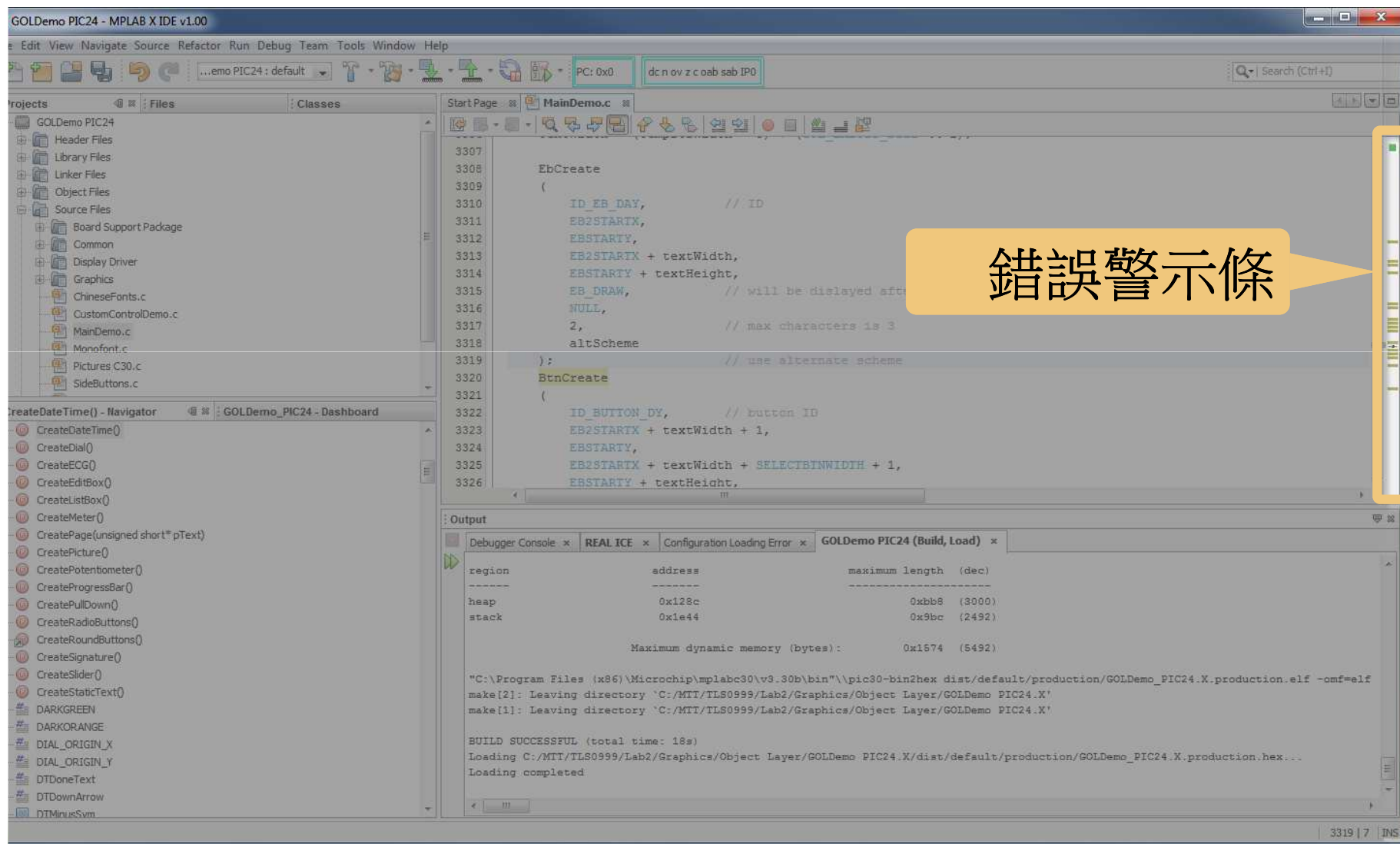
Output

region	address	maximum length (dec)
heap	0x128c	0xbb8 (3000)
stack	0x1e44	0x9bc (2492)
Maximum dynamic memory (bytes):		0x1574 (5492)

BUILD SUCCESSFUL (total time: 18s)
Loading C:/MIT/TL80999/Lab2/Graphics/Object Layer/GOLDemo PIC24.X/dist/default/production/GOLDemo_PIC24.X.production.hex...
Loading completed

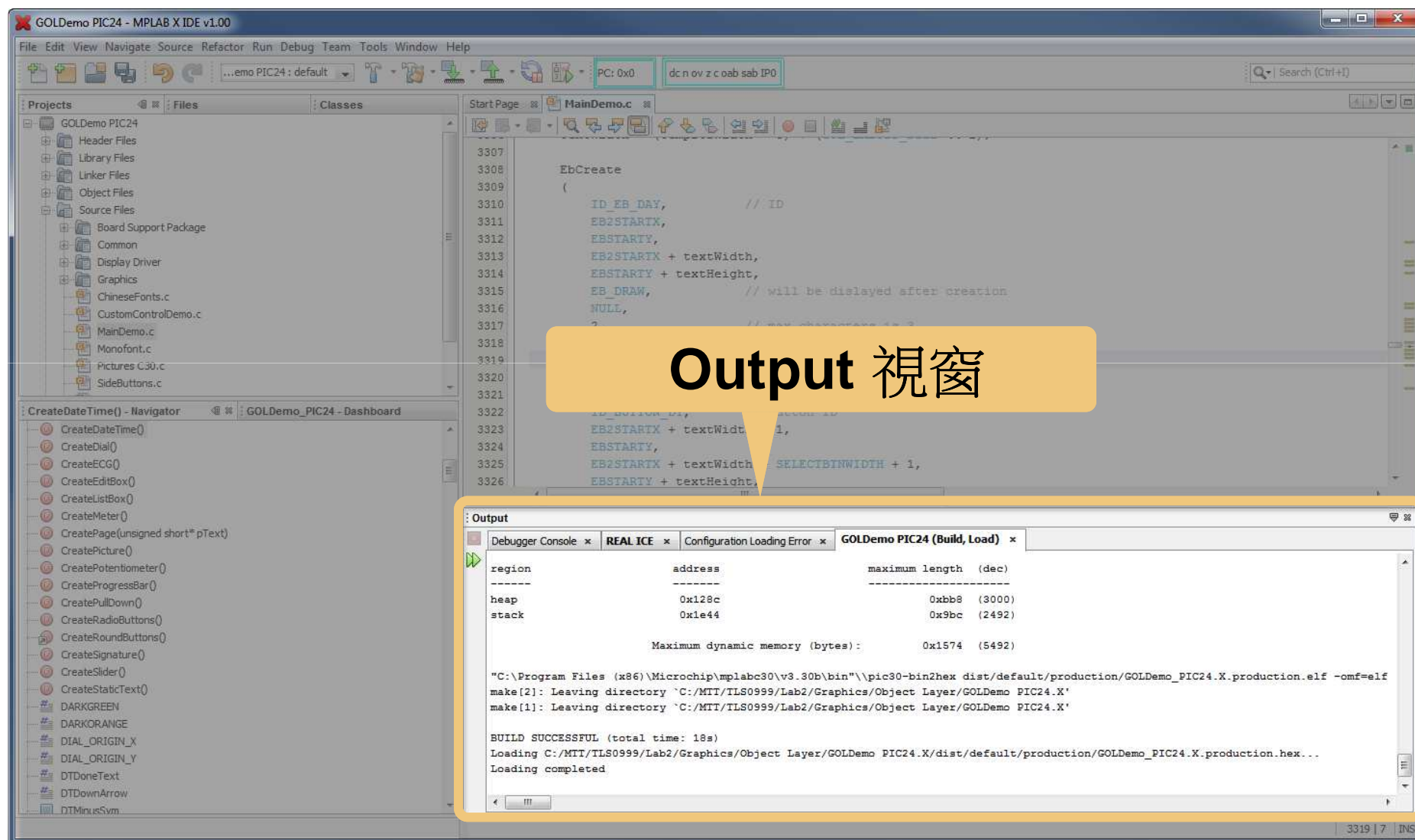
X IDE 版面配置

錯誤警示條



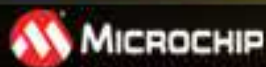
X IDE 版面配置

Output (輸出) 視窗

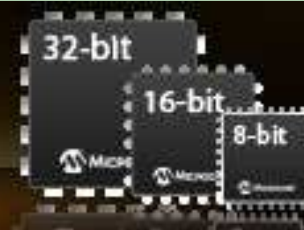


Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



如何切換 **USB** 驅動程式

在 **MPLAB[®] IDE v8.x** 和 **MPLAB X IDE** 間

切換驅動程式

(僅針對 **Windows[®]** 平臺)

先了解切換步驟

開發工具 ICD 3 的切換

- **PICKit 3 使用 HID 驅動程式**
 - **MPLAB IDE v8.x 和 MPLAB X IDE 一樣不用做切換**
- **切換 ICD 3 (確定 ICD3 已經連接 USB)**
 - 確定 **MPLAB IDE 及 X IDE 都已經關閉**
 - 執行 **MPLAB Device Driver Switcher**
 - 選擇 **ICD 3 給 MPLAB 後，按下 Apply Changes 選項**
 - 於對話視窗確定一下 **ICD 3 切換成供訊息**
- 此切換工具很笨，它不會提示目前工具的設定是給 **MPLAB IDE 或 MPLAB X IDE 來使用**



如何切換 USB 驅動程式

1 關閉 MPLAB，插入硬體工具

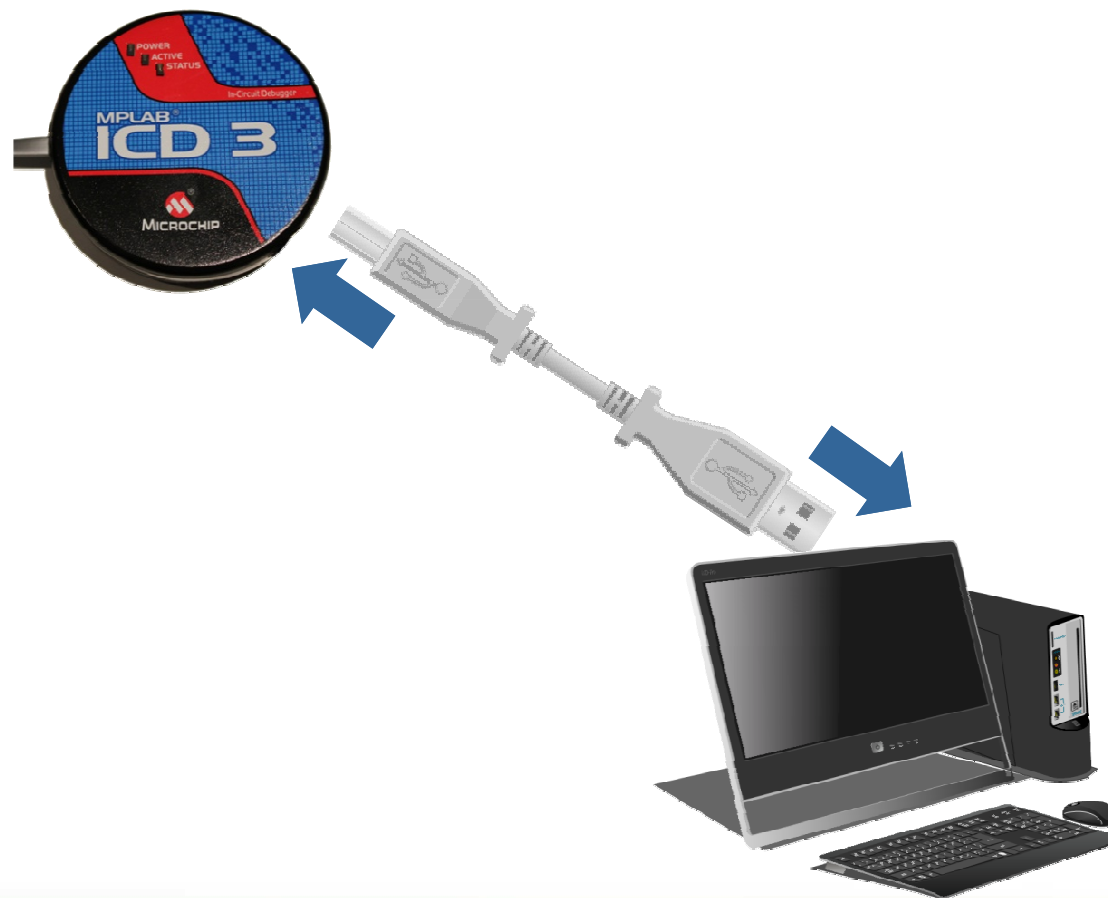


在進行其他操作之前：

1. 確定 **MPLAB IDE v8.x** 及 **MPLAB X IDE** 程式已關閉。

2. 將 **REAL ICE™** 或 **MPLAB® ICD 3** 連接到 PC 的 **USB**。

除非工具已插入並且現有驅動程式已載入，否則無法切換驅動程式。



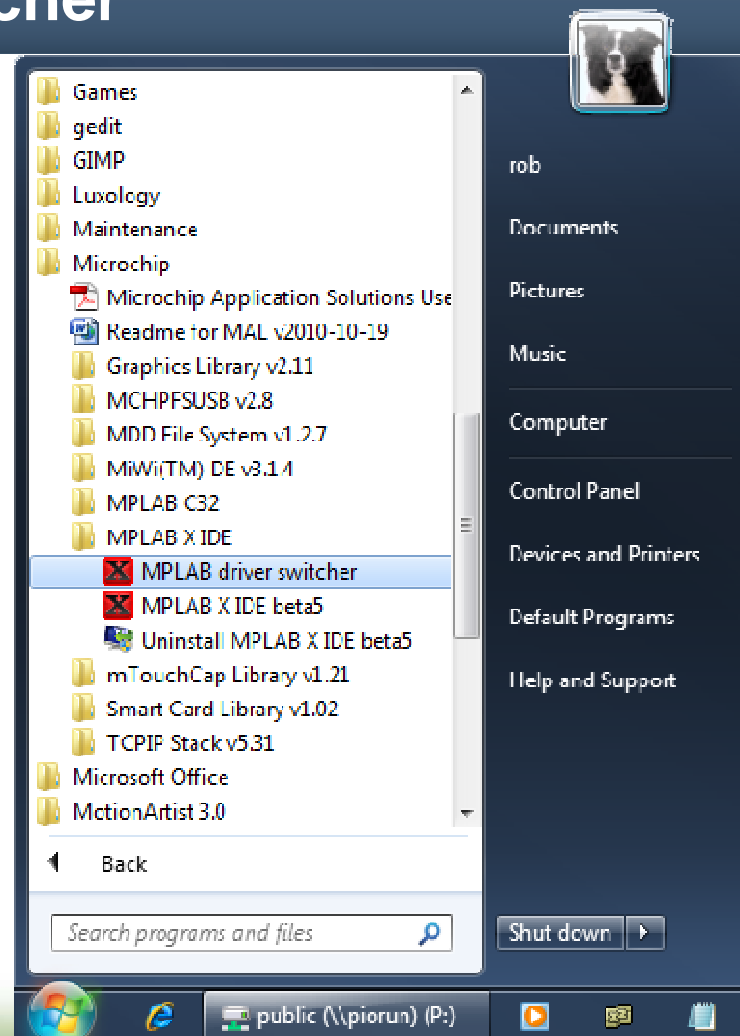
如何切換USB驅動程式

2 啟動 MPLAB driver switcher

在Windows®“開始”選單中選擇：

- ▶ 所有程式/程式
- ▶ Microchip
- ▶ MPLAB® X IDE
- ▶ MPLAB Driver Switcher
(MPLAB 驅動程式切換工具)

直接在桌面開啟圖示：

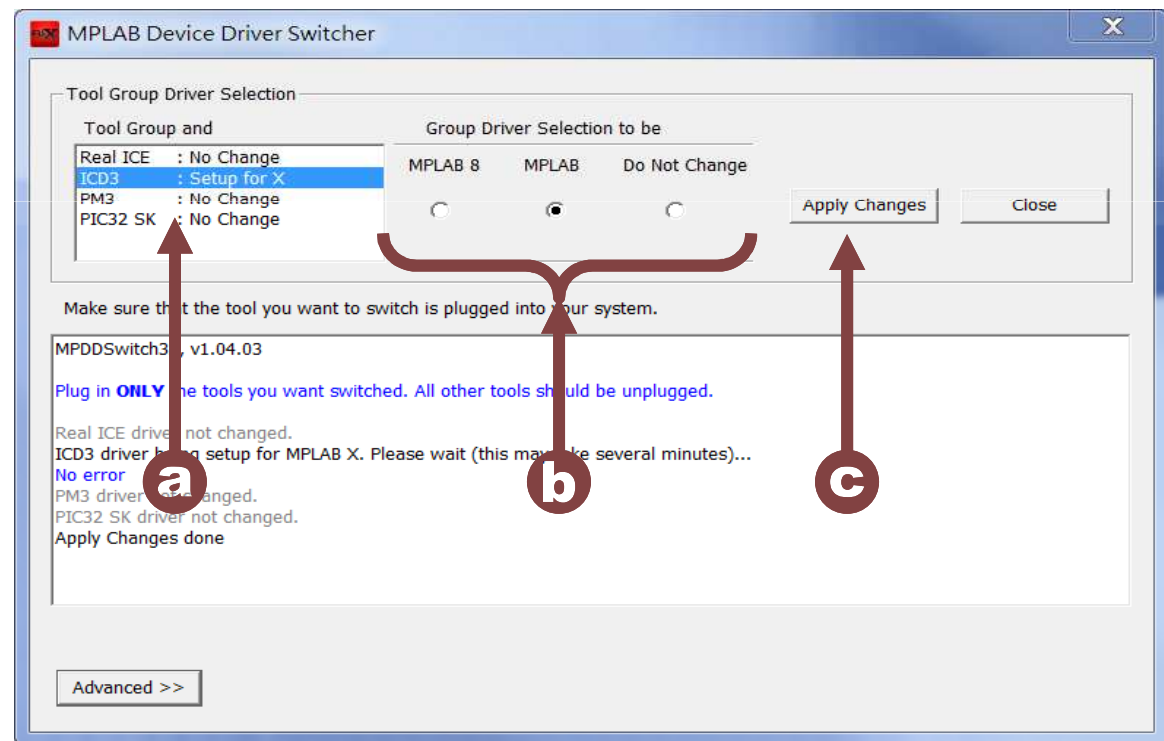


如何切換USB驅動程式

3 選擇工具和驅動程式

- a** 選擇要切換驅動程式的工具
- b** 選擇所需的驅動程式
- c** 按一下

Apply Changes



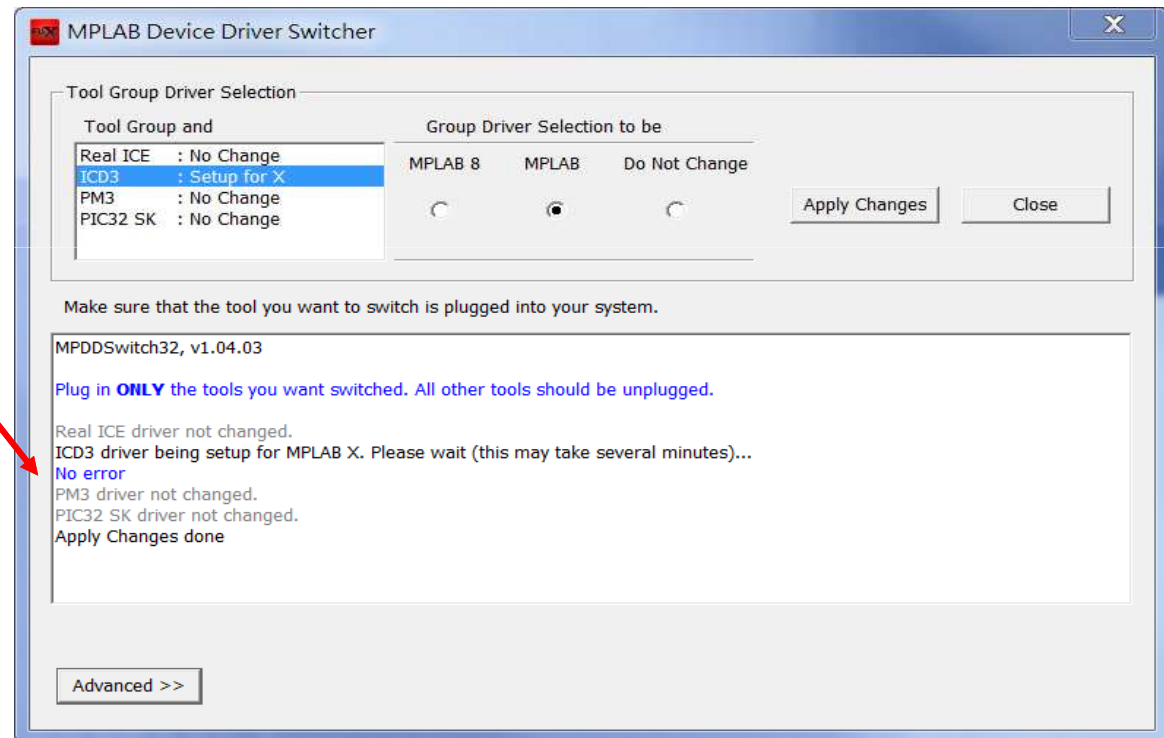
如何切換 USB 驅動程式

4 切換完成

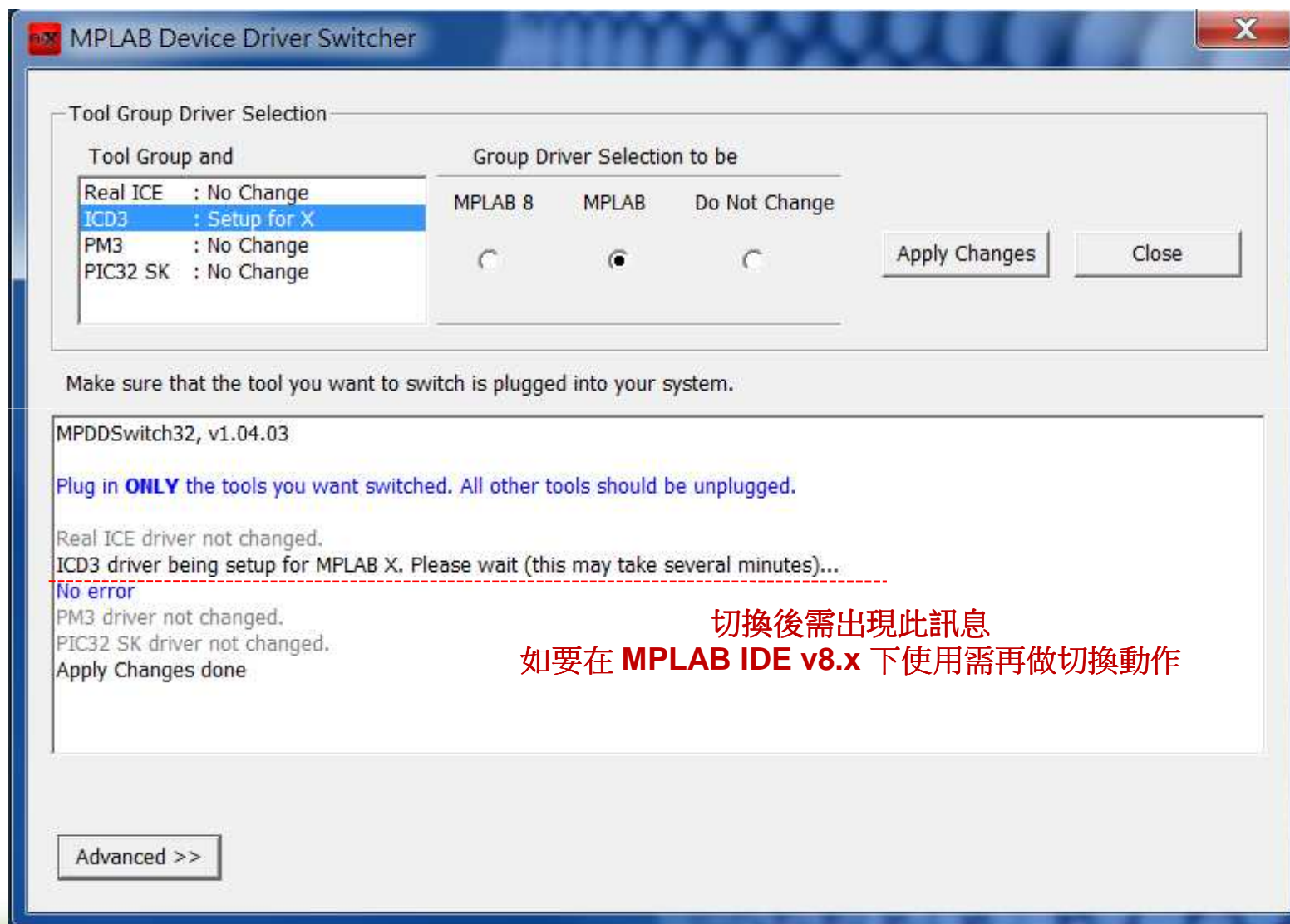
完成更改可能需要等一下讓
驅動程式完成切換。

USB 驅動程式成功切換後，
Output 視窗將顯示 **No error**
(沒有錯誤)。

圖中顯示 ICD 3 切換成功的
對話訊息。



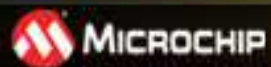
ICD 3 成功切換至 MPLAB X IDE



切換後需出現此訊息
如要在 **MPLAB IDE v8.x** 下使用需再做切換動作

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



如何設定 **XC** 編譯器

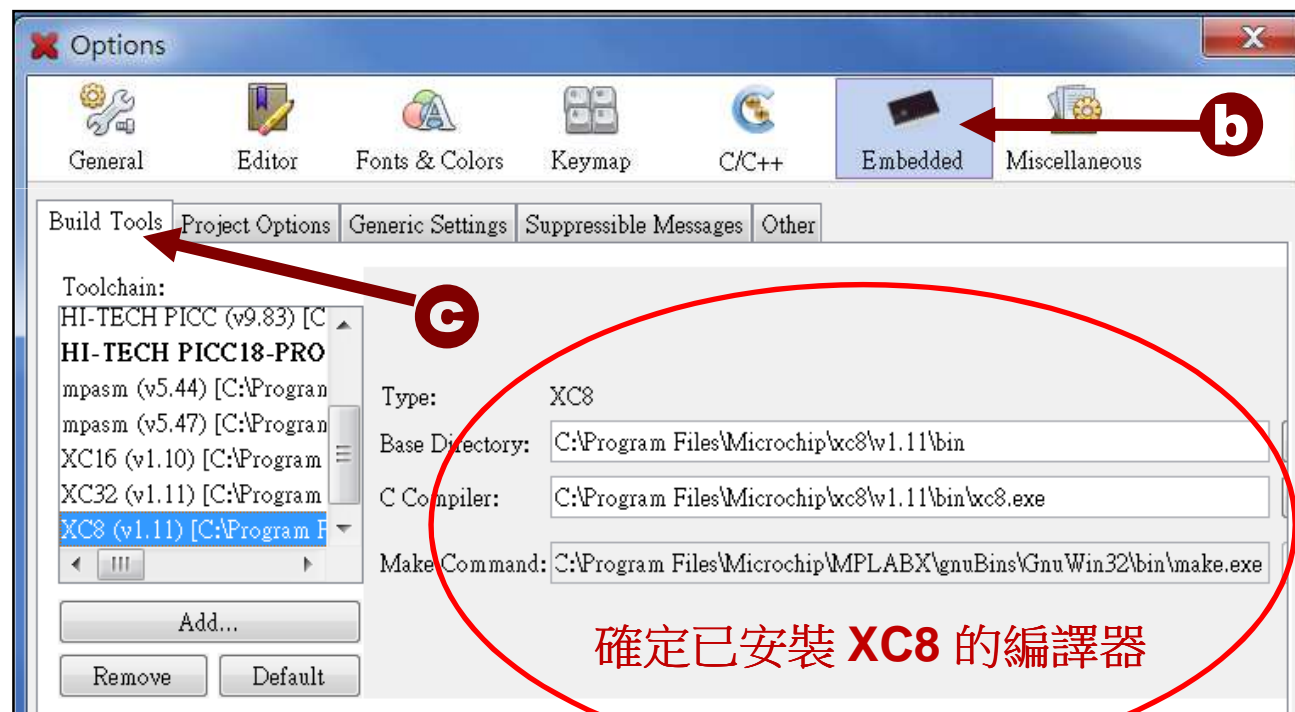
如何設定 XC 編譯器

1 打開 **Embedded** 選項視窗

a 從 主功能表 ► **Tools** (工具) ► **Options** (選項)

b 在打開的視窗中，
選擇 **Embedded**
分類

c 選擇 **Build Tools**
(編譯工具) 選項
卡



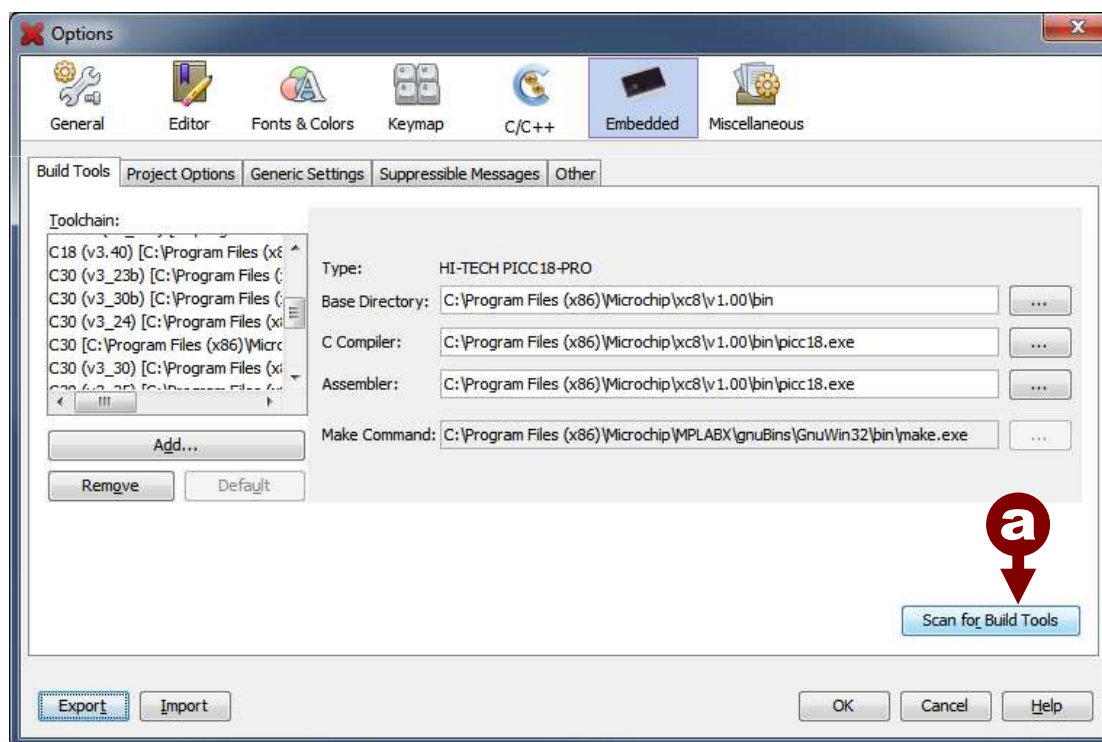
如何設定 XC 編譯器

2 恢復預設值

a 按一下 **Scan for Build Tools** 按鈕

若視窗中顯示了您已經安裝的編譯器，則完成操作。

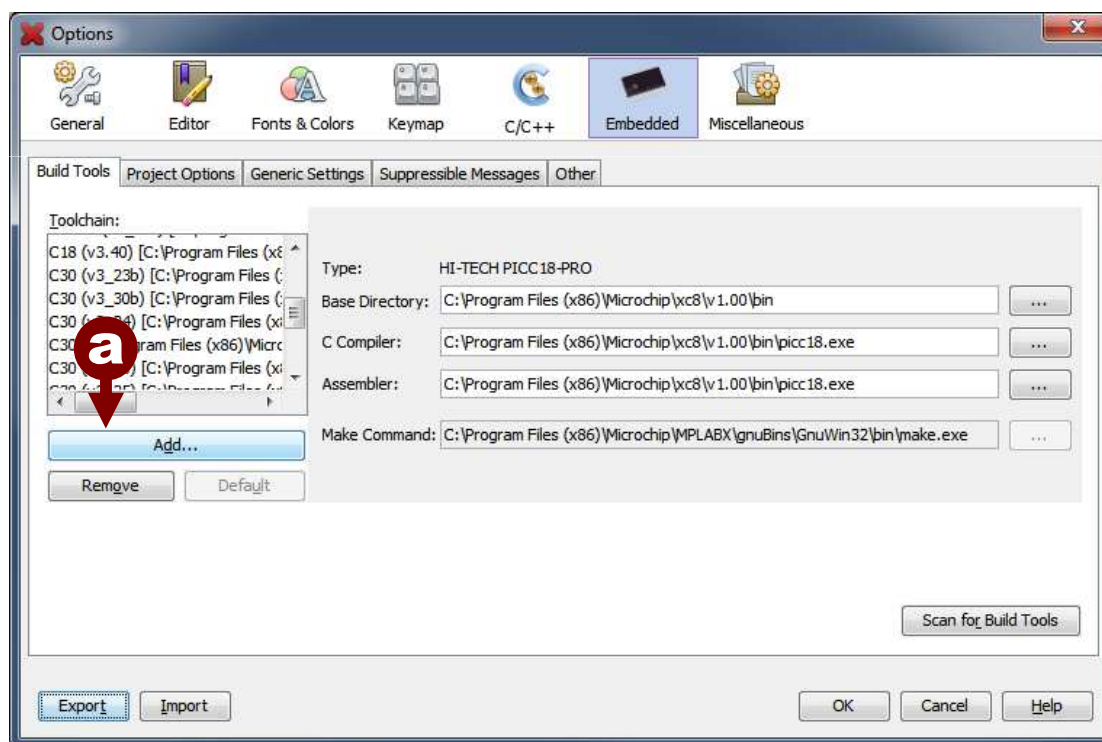
若找不到編譯器，則繼續下一步驟 3。



如何設定 XC 編譯器

3 加入一個新的編譯器

a 按一  按鈕

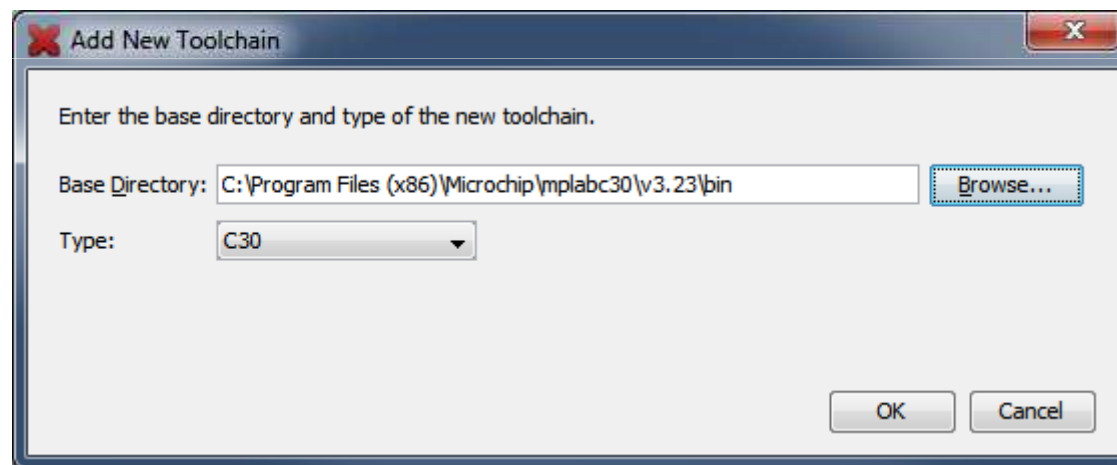


如何設定 XC 編譯器

4 指定編譯器的根目錄

a 按一下 **Browse...** 按鈕並移至編譯器根目錄（通常稱為 **bin**）

Type（類型）框應自動填充正確的類型。如果未檢測到正確的類型，可以手動進行更改。

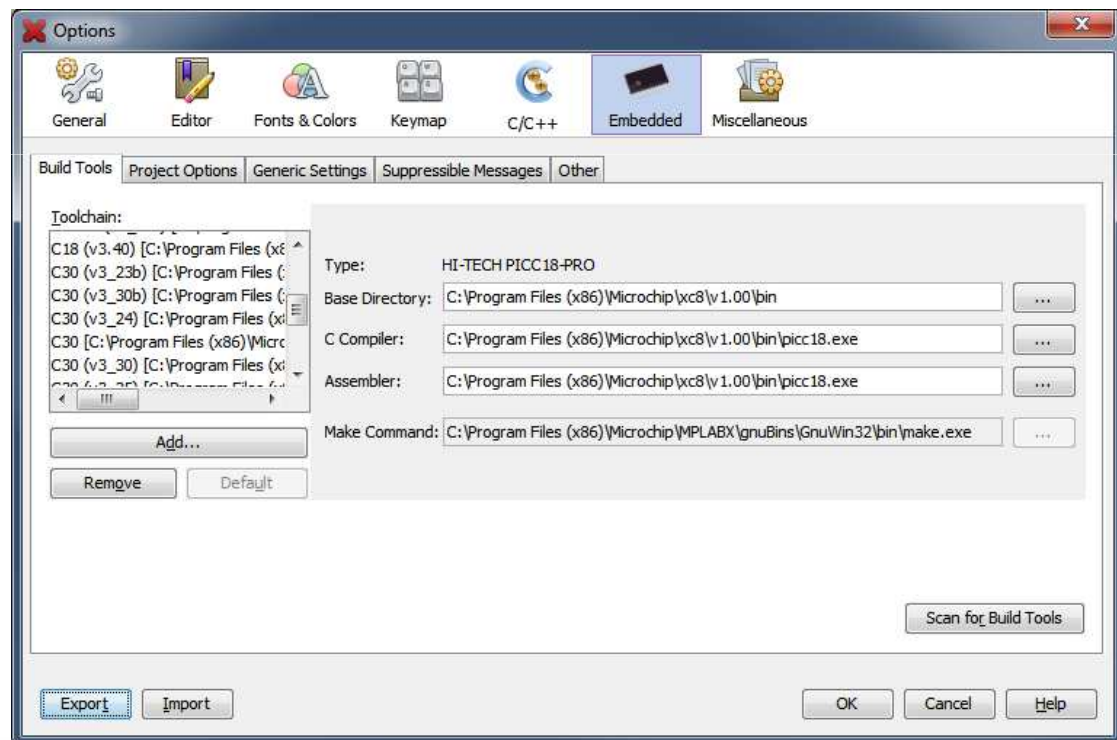


b 按一下 **OK** 按鈕

如何設定 XC 編譯器

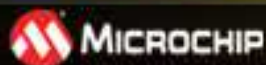
5 完成

所有路徑將自動填入



Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



Configuration Bits

使 **_config** 的程式設定
在 **MPLAB X IDE** 下手動修改
Config. Source Code Generator

設定 Configuration Bits

定義

Configuration Bits - 暫存器中的特殊位元，只能在程式設計時進行修改。這些設定位元主控制著 MCU “永久” 功能的設置，通常在燒錄程式時設置一次。

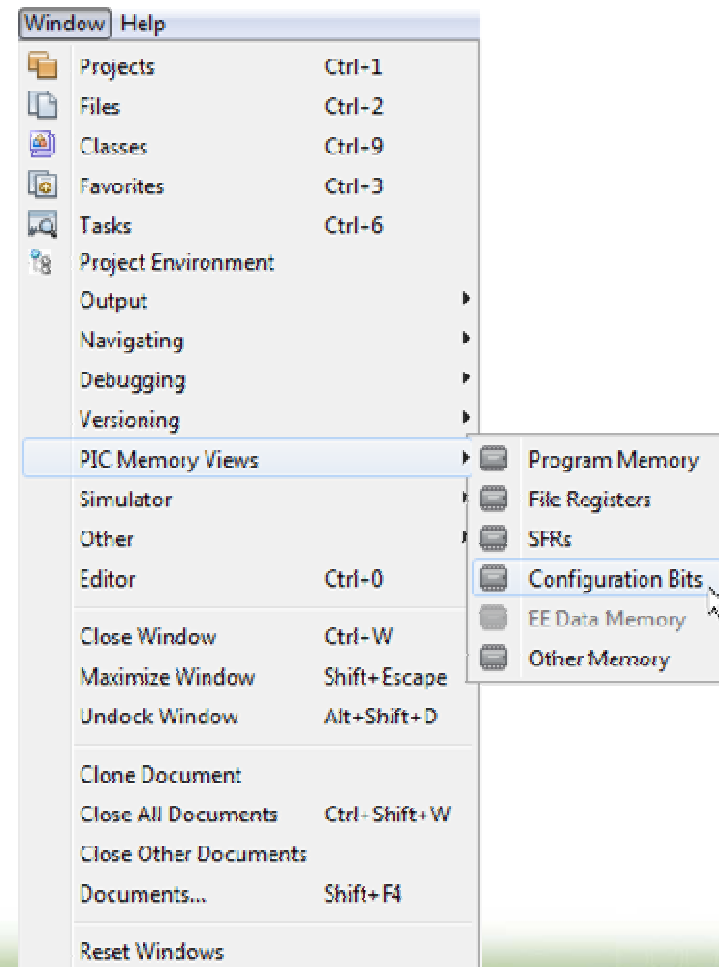
- 設定如下功能：
 - 振盪器類型
 - 看門狗計時器
 - 程式保護 ...
- 更多詳細資訊，請參見資料手冊中的“**PIC** 的特殊功能暫存器”

如何查看 Configuration Bits

1 打開 Configuration Bits（設定位元）視窗

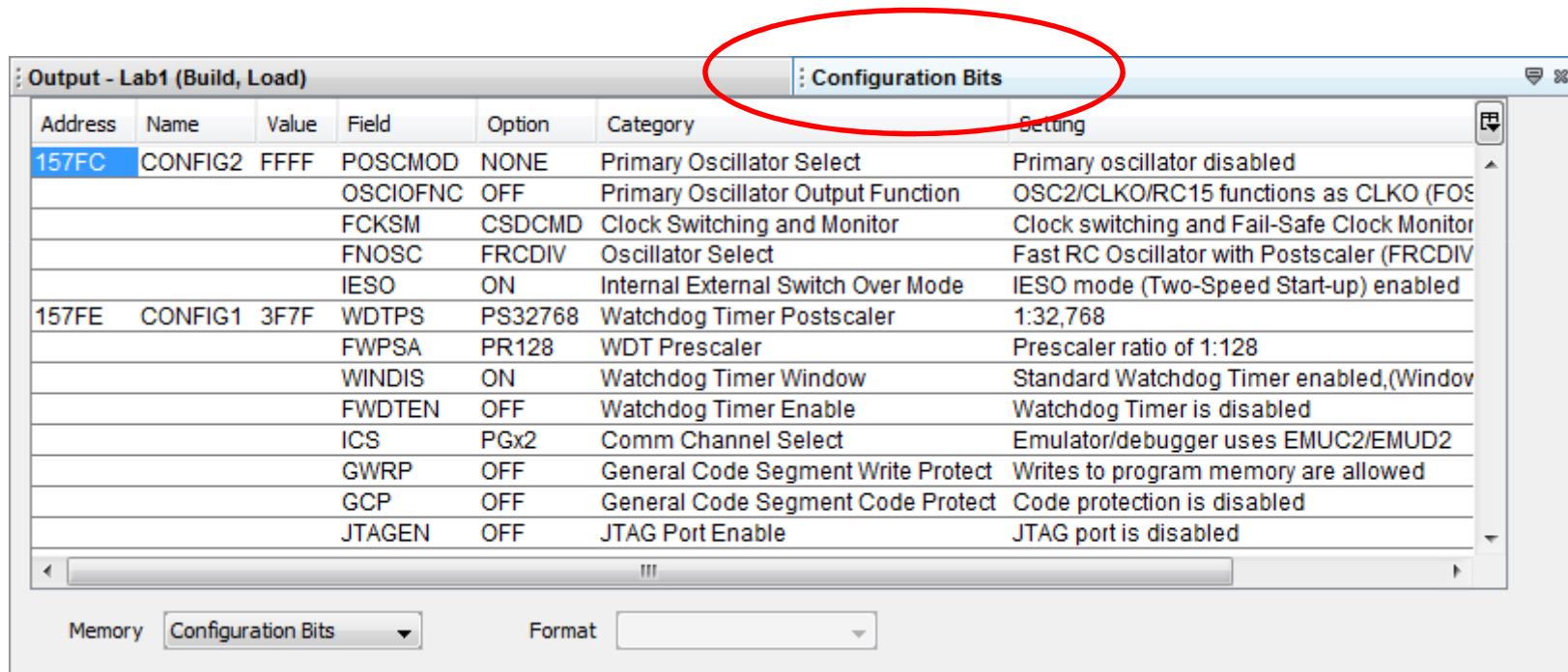
從主功能表中選擇：

- ▶ **Window**
- ▶ **PIC Memory Views**
（PIC 記憶體視圖）
- ▶ **Configuration Bits**



如何查看 Configuration Bits

2 該視窗以選項卡的形式在 **Output** 視窗旁邊打開。



在程式裡設定 **Configuration Bits**

- 建議在程式中設定完成
 - 當然你也可以手動方式去修改設定值

PIC24 和 dsPIC

範例:

```
#include <p24FJ128GA010.h>
__CONFIG1(FWDTEN_OFF & JTAGEN_OFF)
```

PIC18F

```
#include <p18f4520.h>
#pragma config OSC=INTIO7, WDT=OFF, BOREN = ON
```

PIC16F

```
#include <pic.h>
__CONFIG ( FOSC_INTOSC & WDTE_OFF & PWRTE_OFF
```



CONFIG 的設定必須在
#include <device> 行
之後的某個位置

如何修改 Configuration Bits

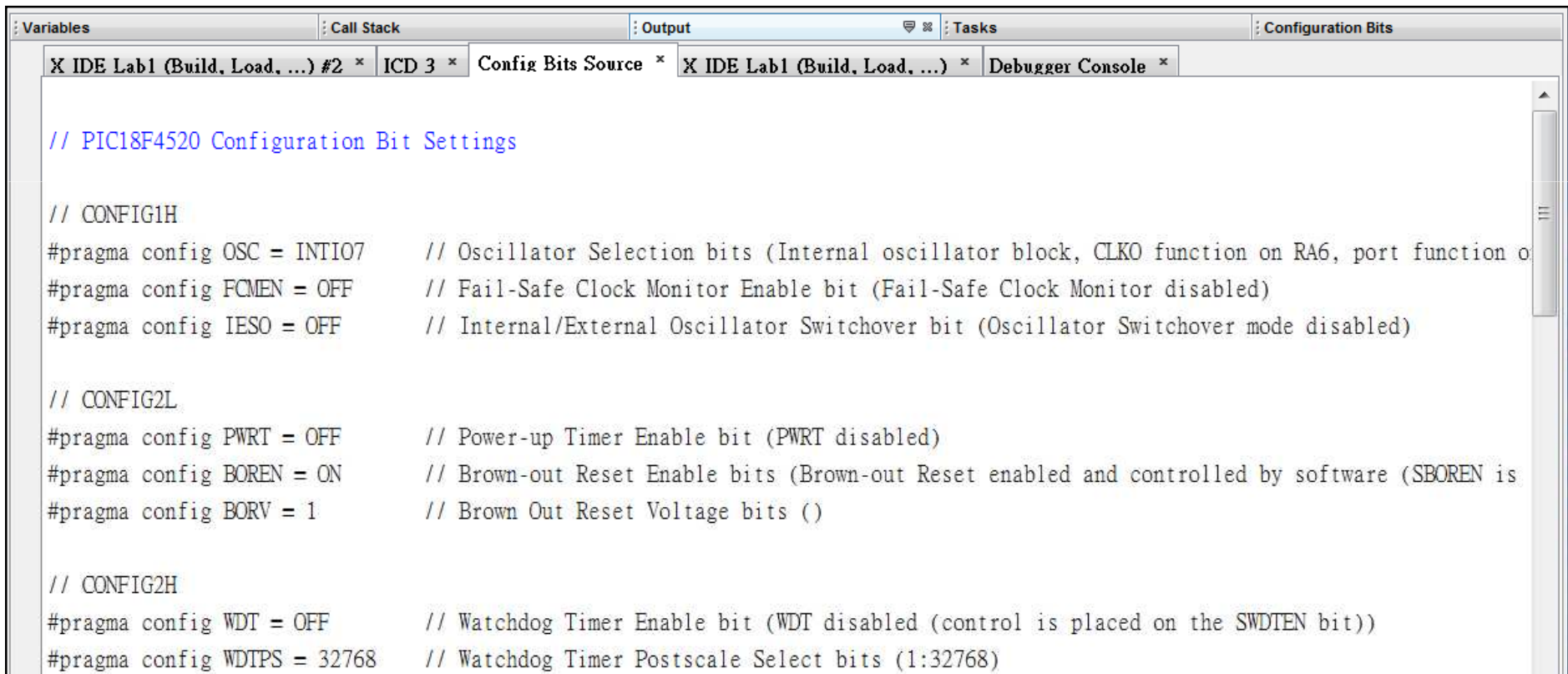
- 在 MPLAB® X IDE 下手動修改設定
 - 在 Option 下，點選要更改的專案
 - 在下拉式功能表中選擇值
 - 按一下按鈕 “Generate Source Code to Output”
(產生設定 Config. 的 C 原始程式)

Address	Name	Value	Field	Option	Category	Setting
157FC	CONFIG2	FFFF	POSCMOD	NONE	Primary Oscillator Select	Primary oscillator disabled
			OSCIOFNC	OFF	Primary Oscillator Output Function	OSC2/CLKO/RC15 functions as CLKO (FOSC/2)
			FCKSM	CSDCMD	Clock Switching and Monitor	Clock switching and Fail-Safe Clock Monitor are disabled
			FNOSC	FRCDIV	Oscillator Select	Fast RC Oscillator with Postscaler (FRCDIV)
			IESO	ON	Internal External Switch Over Mode	IESO mode (Two-Speed Start-Up) enabled
157FE	CONFIG1	7FFF	WDTPS	PS32768	Watchdog Timer Postscaler	1:32,768
			FWPSA	PR128	WDT Prescaler	Prescaler ratio of 1:128
			WINDIS	ON	Watchdog Timer Window	Standard Watchdog Timer enabled,(Windowed-mode is disabled)
			FWDTEN	ON	Watchdog Timer Enable	Watchdog Timer is enabled
			ICS	PGx2	Comm Channel Select	Emulator/debugger uses EMUC2/EMUD2
			GWPR	OFF	General Code Segment Write Protect	Writes to program memory are allowed
			GCP	OFF	General Code Segment Code Protect	Code protection is disabled
			JTAGEN	ON	JTAG Port Enable	JTAG port is enabled

Memory Configuration Bits
Format Read/Write
Generate Source Code to Output

使用 Configuration Bits

- 開啟 “Config Bits Source” 的選項
- 選擇 “Save As” (另存為) config.c
- 將此 C 檔加到專案中並編譯。



```
// PIC18F4520 Configuration Bit Settings

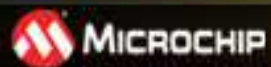
// CONFIG1H
#pragma config OSC = INTIO7       // Oscillator Selection bits (Internal oscillator block, CLKO function on RA6, port function on RA4)
#pragma config FCMEN = OFF        // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF         // Internal/External Oscillator Switchover bit (Oscillator Switchover mode disabled)

// CONFIG2L
#pragma config PWRT = OFF         // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = ON         // Brown-out Reset Enable bits (Brown-out Reset enabled and controlled by software (SBOREN is controlled by hardware))
#pragma config BORV = 1           // Brown Out Reset Voltage bits (1:2.0V to 2.7V)

// CONFIG2H
#pragma config WDT = OFF          // Watchdog Timer Enable bit (WDT disabled (control is placed on the SWDTEN bit))
#pragma config WDTPS = 32768      // Watchdog Timer Postscale Select bits (1:32768)
```

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC MCU PLATFORM



實驗一

如何新建獨立專案

實驗一

如何建立獨立專案



目的

- 使用專案精靈從頭開始建立專案
- 描述在專案精靈中選擇的各個選項
- 提供可編譯自己專案的平臺

實驗一

如何建立獨立專案

目標

- 練習一所提供的原始檔案從頭開始建立新的 **MPLAB® X** 專案
 - **PIC18F_LCD_APP001.c**
 - **Lab1.c**
- 使 **LED1** 閃爍，並在 **APP001** 板的 **LCD** 模組上顯示：

“Welcome to X IDE”

實驗一

如何建立獨立專案



步驟

1. 選擇專案類型
2. 指定開發的元件編號
3. 指定除錯器 / 程式燒錄器
4. 指定編譯器
5. 指定專案名稱和位置
6. 將檔加入到專案中
7. 執行專案

實驗一

如何建立獨立的專案

1 啟動新建專案精靈

工具列



選單

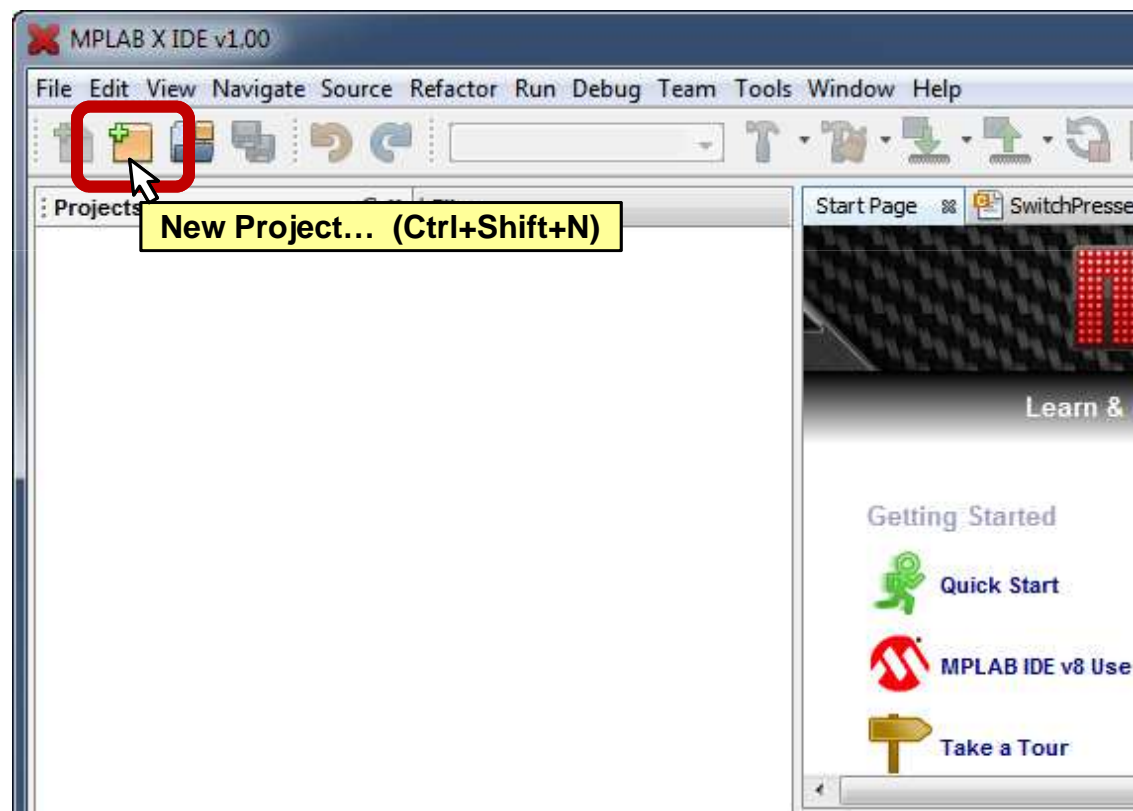
File (文件) ►
New Project...
(新建專案...)

鍵盤

Ctrl

↑ Shift

N



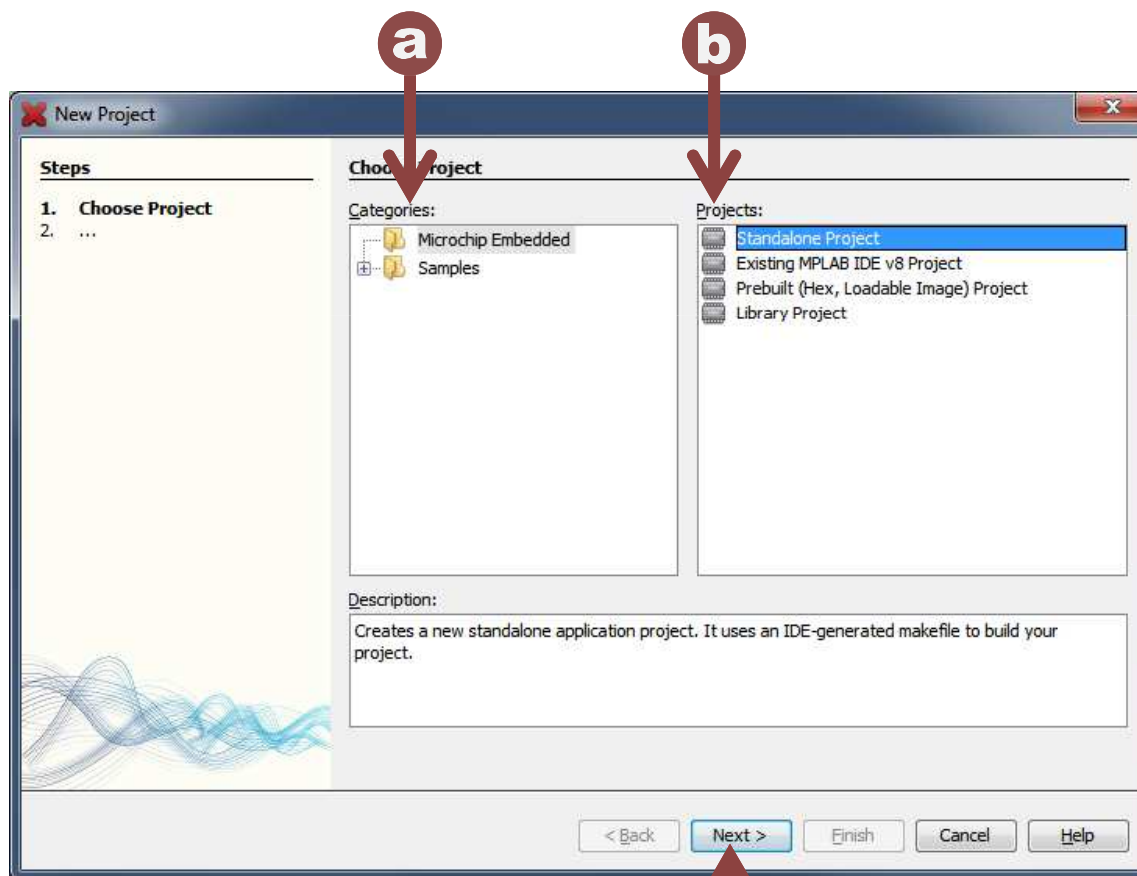
實驗一

如何建立獨立專案

2 選擇專案

- a** 在“**Categories**”（類別）下選擇：
Microchip Embedded
（Microchip 嵌入式）
- b** 在“**Projects**”（專案）下選擇
Standalone Project
（獨立專案）

- c** 按一下 **Next >**



實驗一

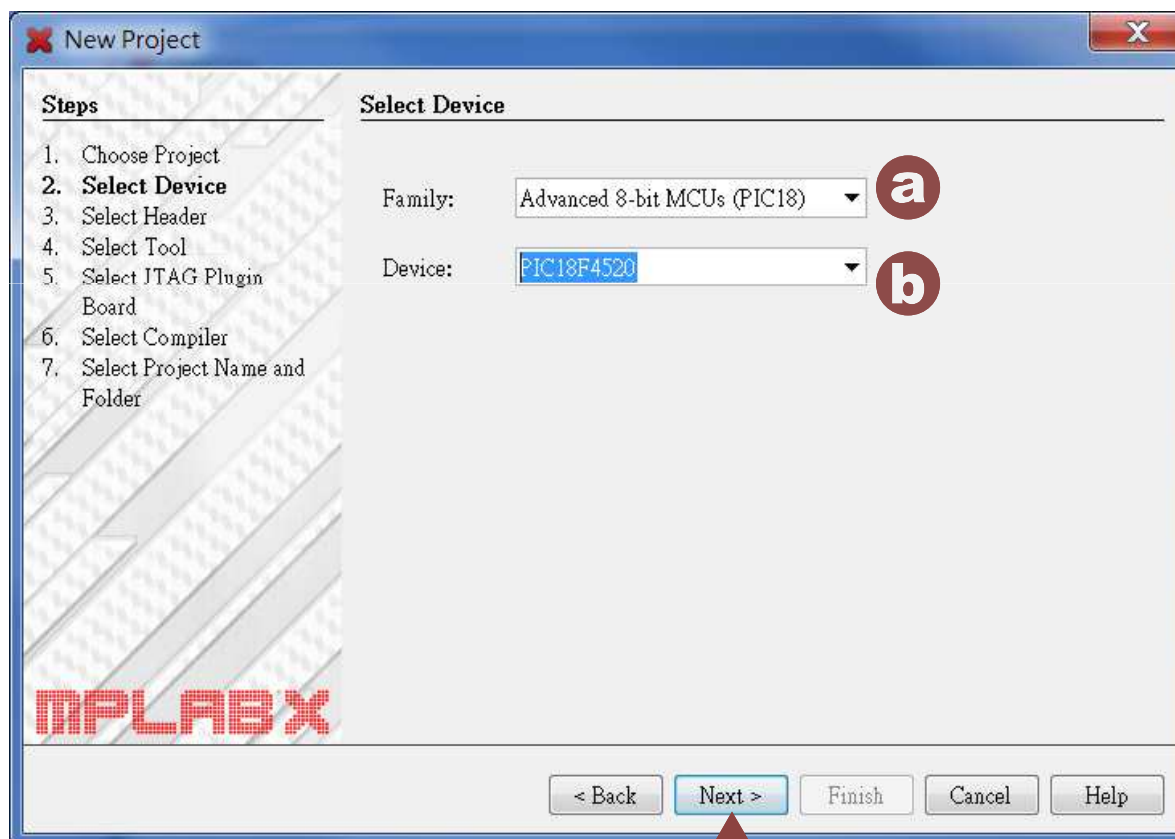
如何建立獨立專案

3 選擇元件

a 在“**Family**”（系列）中選擇：
Advanced 8-bit MCUs (PIC18)

b 在“**Device**”（元件）中選擇：
PIC18F4520

c 按一下 **Next >**



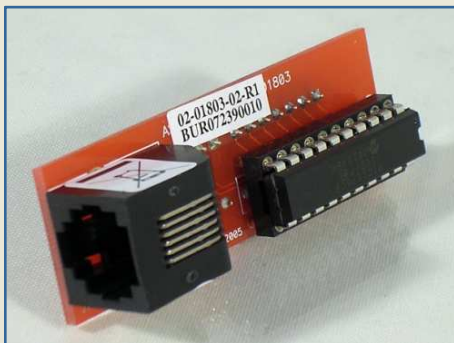
實驗一

如何建立獨立專案

4 選擇元件轉接座 (Device Header)

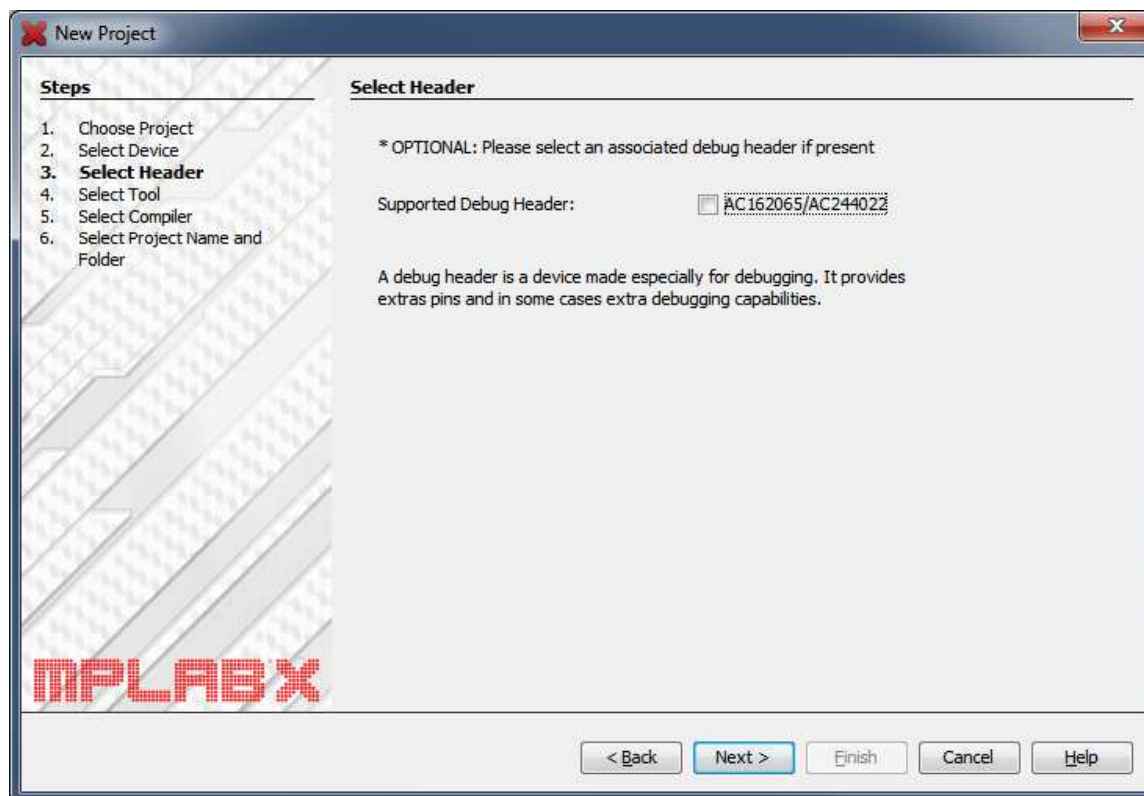
本練習中使用的元件無需模擬頭。

自動略過此項目，但有些較少腳位的元件會有此需求。



按一下

Next >



實驗一

如何建立獨立專案

5 選擇工具

a 選擇 點選 **ICD3** 下的序號



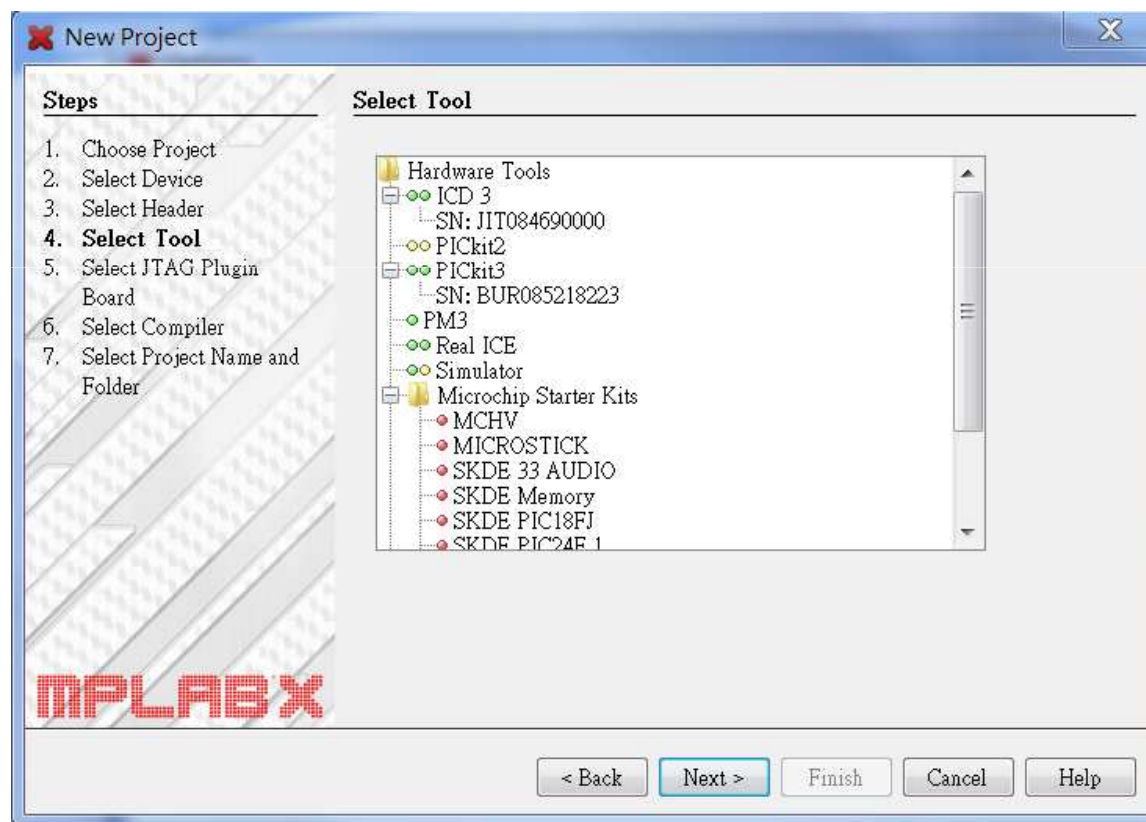
硬體工具必須插上 **USB** 槽，
先讓 **X IDE** 辨識。

如果使用硬體除錯工具，則
選擇其**序號**，如右側 **ICD 3**
下方所示。

在工具選用項裡可以看到有
兩種工具可以選用：

- 一為 **ICD 3**
- 二為 **PICKit 3**

b 按一下 **Next >**



實驗一

如何建立獨立專案

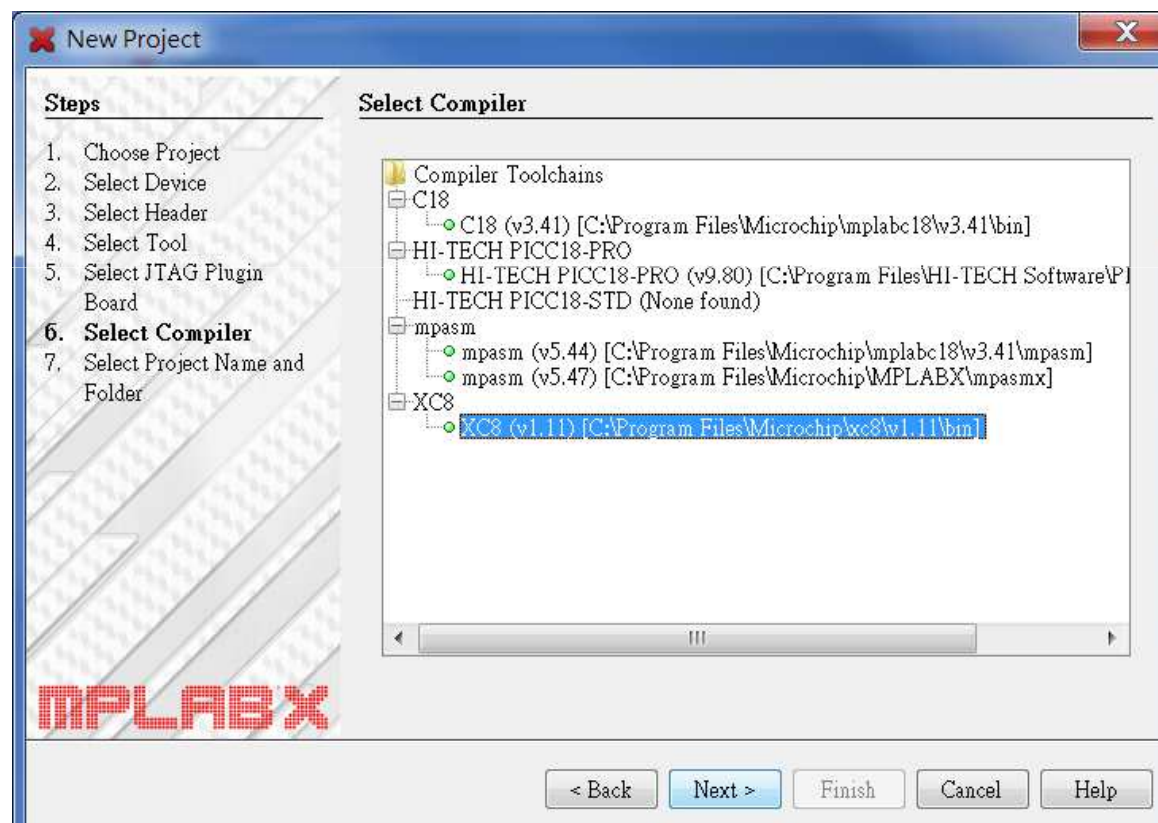
6 選擇 C 編譯器

- a** 按一下要使用的編譯器名稱下的編譯器版本。
實驗中請選擇 XC8 底下 XC8 (v1.11)



如果在編譯器名稱下看不到版本號，則可能是編譯器未安裝或 IDE 無法查找到該編譯器。

- b** 按一下 **Next >**



實驗一

如何建立獨立專案

7 選擇專案名稱和資料夾及中文編碼格式

a 輸入專案名稱：

X IDE Lab1

b 輸入專案位置：

D:\..\PIC18F Labs\Lab1

會在專案文件夾下建立一個具有專案名稱的資料夾。



磁碟機 (D:)



MPLAB X IDE RTC



PIC18F Labs



Lab1

← 專案資料夾



X IDE Lab1.X

← 專案名稱.X

c

按一下

Finish

Select Project Name and Folder

Project Name: X IDE Lab1

Project Location: D:\MPLAB X IDE RTC\PIC18F Labs\Lab1

Project Folder: D:\MPLAB X IDE RTC\PIC18F Labs\Lab1\X IDE Lab1.X

☐ Overwrite existing project.

☐ Also delete sources.

☐ Set as main project

Encoding: UTF-8

有關中文註解及顯示問題
請參考下一張投影片的說明

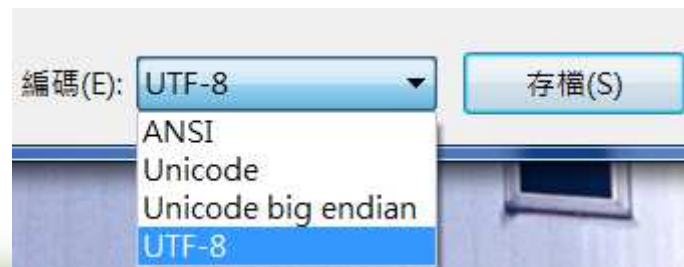
編碼方式

實驗一

關於中文編輯與顯示

- 建議 **MPLAB X IDE** 使用 **UTF-8** 的中文編碼格式
 - 所以直接在 **X IDE** 下，直接撰寫程式是沒有中文顯示的問題
- 但 **MPLAB IDE v8.x** 的中文使用 **Big-5** 編碼方式，除非在 **X IDE** 也一樣選用 **Big-5** 編碼來顯示，如果 **X IDE** 設成 **UTF-8** 會造成中文變成亂碼...
- 解決方法：
 - 將原先在 **MPLAB v8.x** 的原始程式檔 (*.C) 先用”記事本”開啟後再用 **Save as** 方式選擇 **UTF-8** 編碼後回存。

記事本選擇編碼
後再回存



實驗一

如何建立獨立專案

8 將檔加入到新專案中

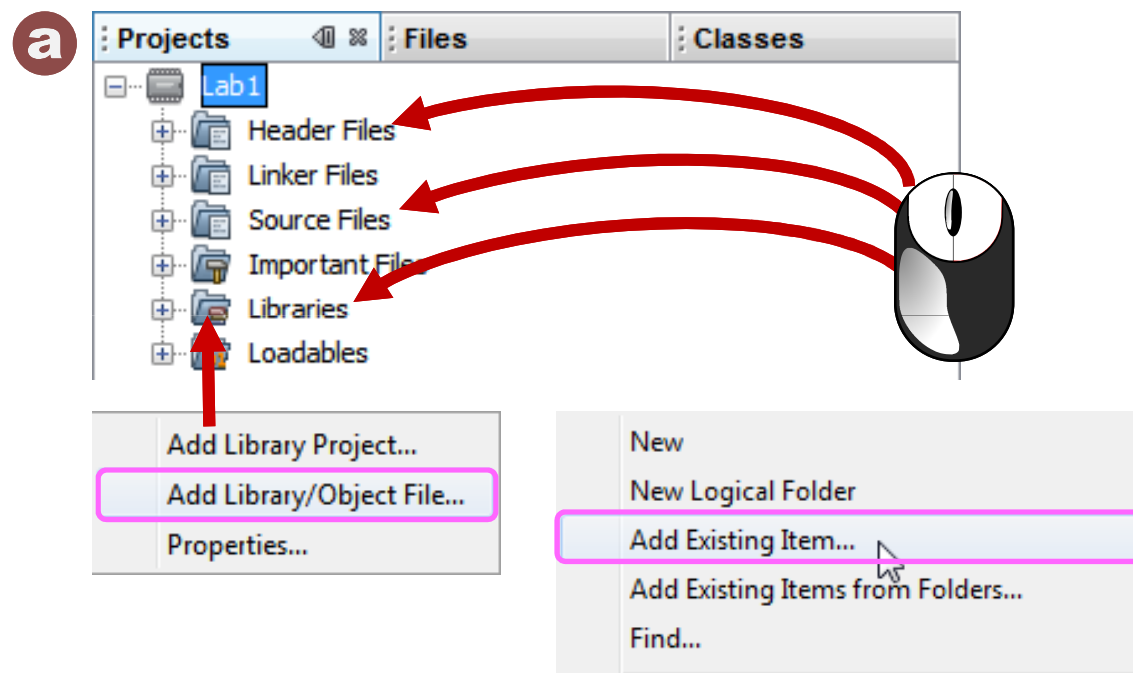
- a** 在專案夾中，利用老鼠右鍵後，從彈出功能表中選擇：**Add Existing Item...**
(加入現存有的檔案...)

對於函式庫，從彈出選單中選擇：

Add Library/Object File
(加入函式庫/目的檔...)

將相關檔案加到以下資料夾中：

- b** • **Header Files**
- c** • **Source Files**
- d** • **Libraries**
(詳見下一頁)








實驗一

如何建立獨立專案

8 將相關檔案加入到新專案中（續.....）

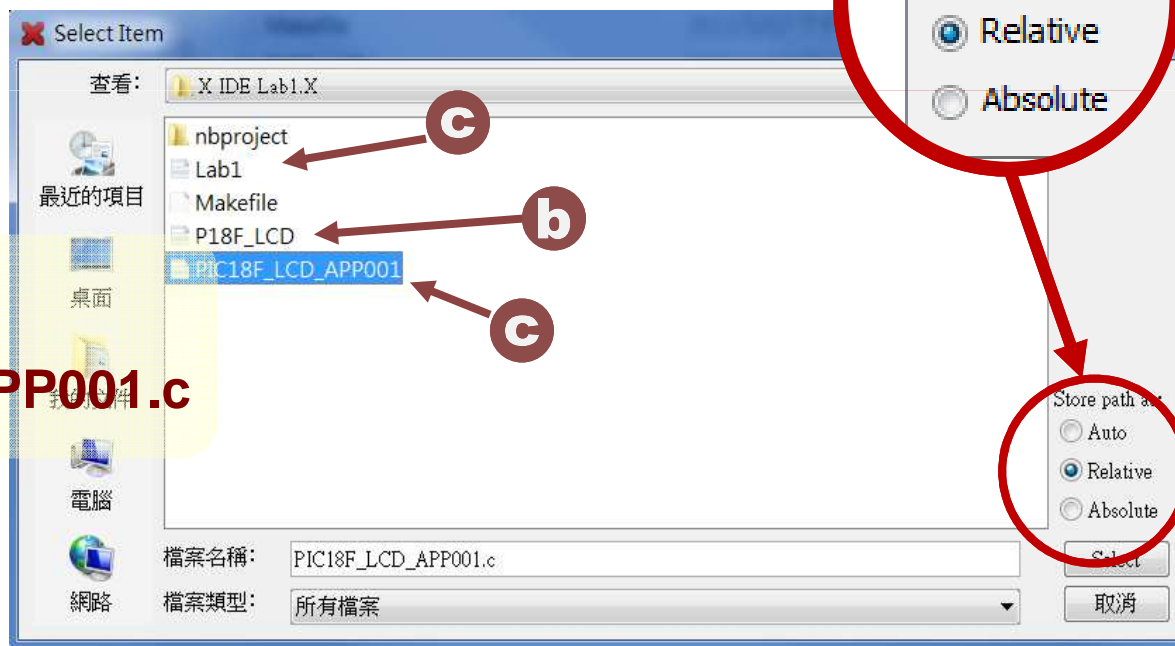
從 **X IDE Lab1.X** 目錄中加入下列文件

b  **Header Files**
 **P18F_LCD.h**

c  **Source Files**
 **Lab1.c**
 **PIC18F_LCD_APP001.c**

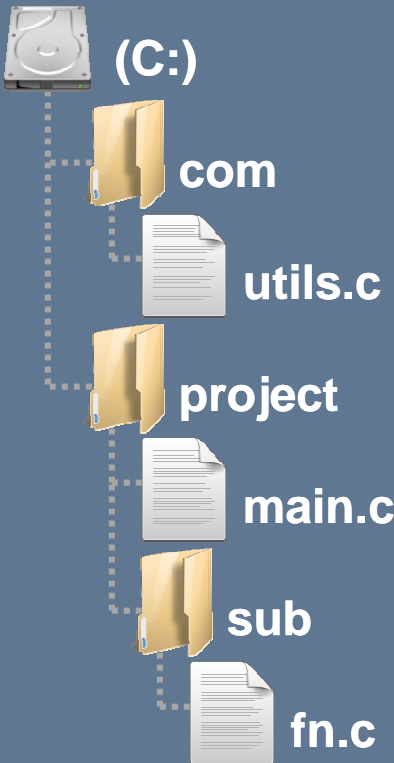
 **Libraries**
 **未使用**

選擇相對檔案儲存方式以便
日後專案的複製使用



實驗一

如何建立獨立專案

儲存路徑 範例	絕對路徑 Absolute	相對路徑 Relative	自動 Auto
	<div> <div>..\ = 上移一層</div> <div>.\ = 專案目錄</div> </div>		
	專案目錄以外		
	C:\com\utils.c	..\com\utils.c	C:\com\utils.c
	使用絕對路徑		
	專案目錄內		
	C:\project\main.c	.\main.c	.\main.c
	使用相對路徑		
	專案目錄的子目錄內		
	C:\project\sub\fn.c	.\sub\fn.c	.\sub\fn.c
	使用相對路徑		

實驗一

如何建立獨立專案

9 編譯和執行專案

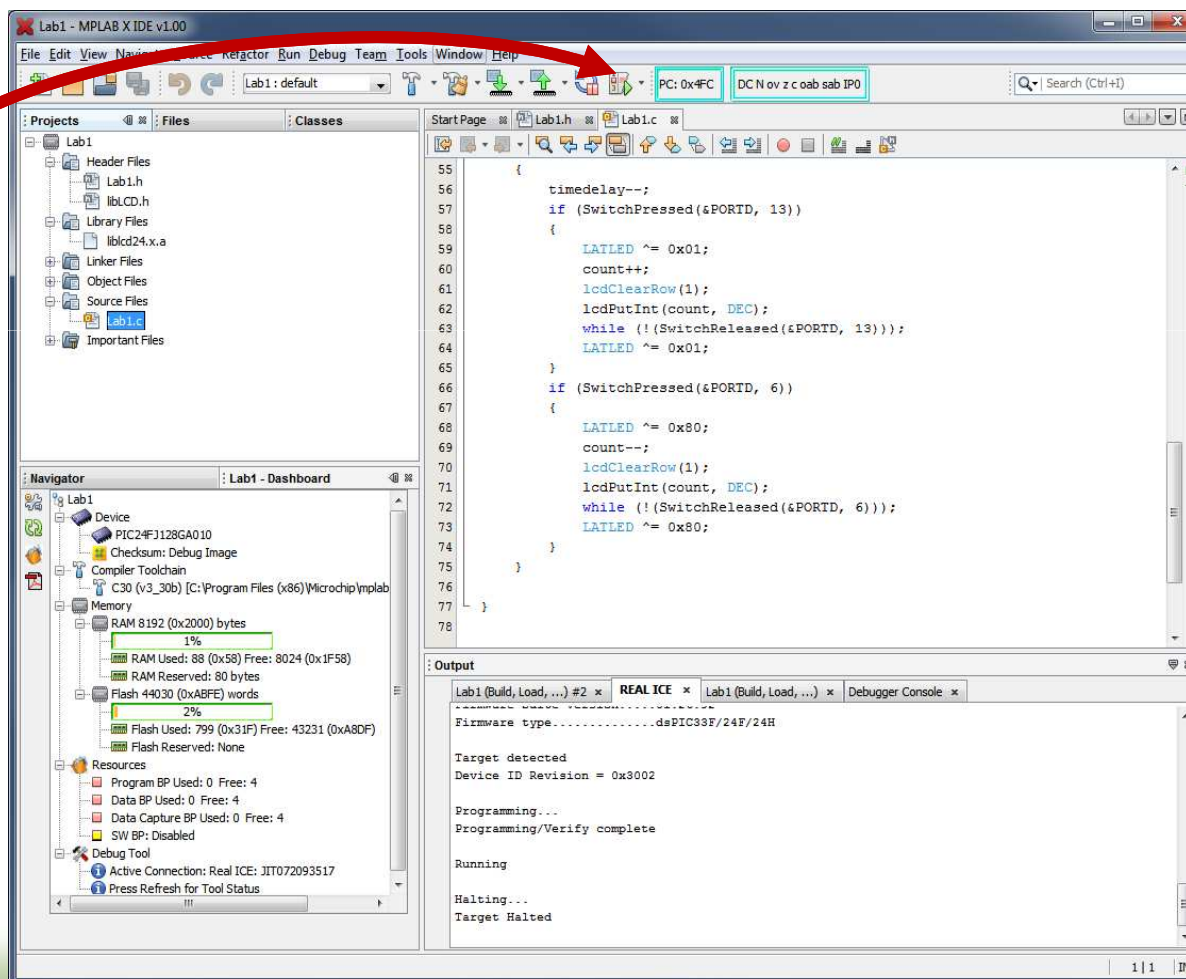
編譯專案，確保每一項都正確完成。

按一下 **Debug Project** (除錯專案) 圖示。



該按鈕的功能：

1. 在 **除錯** 模式下編譯 (**make**) 專案
2. 重新與 ICD3 連線
3. 將程式燒錄至目標板上的 PIC18F4520 元件
4. 執行 C 啟動程式 (游標停在 **main()** 函式)
5. 按繼續執行圖示開始執行程式



實驗一

如何建立獨立專案



結果

如果所有操作均正確完成，則
LCD 螢幕上會顯示：

“ **Welcomr to X IDE** ”

且 **LED1** 在閃爍。

完成後：

按一下 **Pause**（暫停）按鈕：



按 **Finish Debug Session**
（終止除錯對話）按鈕：



（下文將給出更多資訊……）

閃爍的 **LED1**



實驗一

如何建立獨立專案

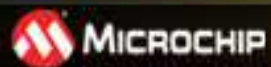


結論

- 您現在應已瞭解建立獨立專案要執行的操作包括：
 - 選擇目標元件編號
 - 選擇編譯和除錯工具
 - 選擇專案目錄
 - 編譯並在目標板上或除錯器中執行程式

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC MCU PLATFORM



如何編譯 (Build)專案

如何編譯應用程式

MPLAB® v8.x



Make

(選單：Project ► Make)
僅編譯目前有修改過的
原始檔案



Build All

(選單：Project ► Build All)
編譯所有檔案。
無論原始檔案是否有修改過

MPLAB X



Build Project

(編譯專案)
僅編譯目前有修改過的
原始檔案



Clean and Build Project

(清除並編譯專案)
編譯所有檔案。
無論原始檔案是否有修改過

如何編譯應用程式

使用除錯器

MPLAB® v8.x

Debug ▼

除錯模式



Make



Program Target Device
(燒錄目標元件)



Reset (重置)



Run (執行)

MPLAB X



Debug Project
(除錯專案)

1. 使用除錯模式運作
2. 編譯程式 (有錯會停止)
3. 啟動除錯工具的連接
4. 燒錄目標元件
5. 重置
6. 執行 (可選)

以上動作一氣呵成，無須單獨點選



在大多數情況下，在點擊該按鈕前，無需點擊“**Build**”或“**Clean and Build**”按鈕。

如何編譯應用程式

使用除錯器

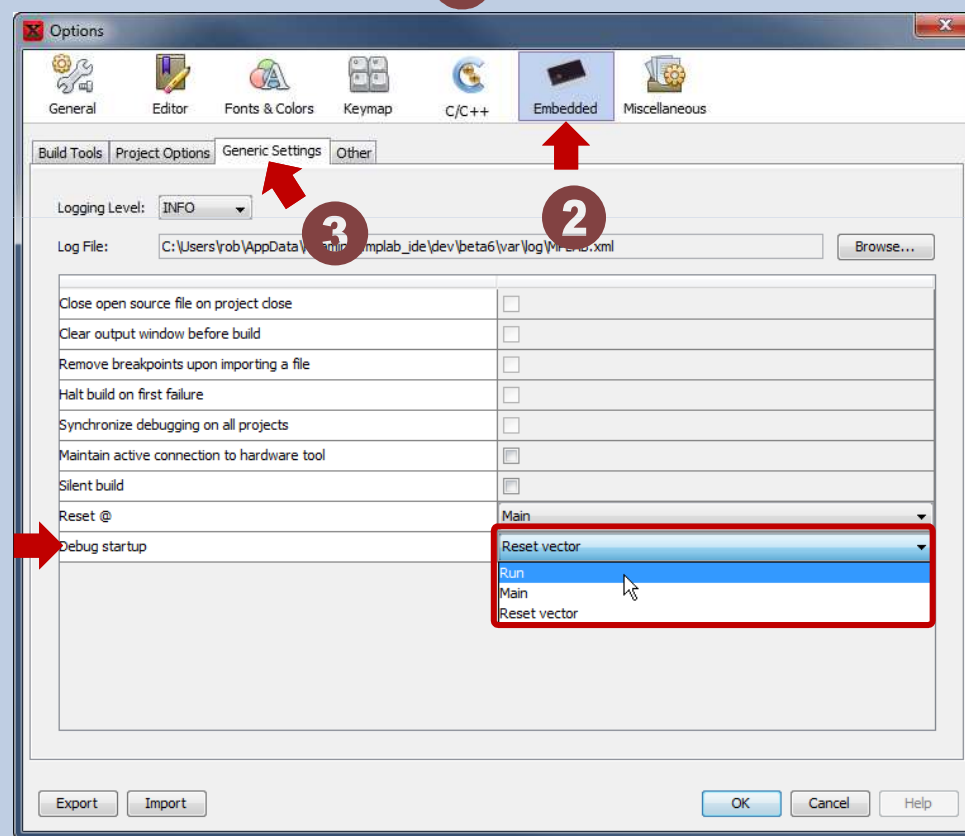


如果您不想除錯器自動開始執行：

- 1 從主功能表中選擇：**Tools ▶ Options**
- 2 選擇 **Embedded** 圖示
- 3 選擇 **Generic Settings** (普通設定) 選項卡
- 4 對於 **Debug startup** (除錯啟動) 設置，選擇：

1. **Main (Reset 後到 Main)**
2. **Reset vector (停在 Reset 位址)**
3. **Run (Reset 後直接執行程式)**

4



如何編譯應用程式

使用程式設計器

MPLAB® 8

Release ▼ 釋放模式



Make



燒錄目標元件

MPLAB X



Program Target Project
(燒錄目標專案)

- 進入釋放模式
- 編譯專案內程式
- 燒錄目標元件




圖示按鈕可以切換 **Hold in Reset** 或 **Release from Reset** 兩種模式。這將允許目標板在不拔除燒錄器的情況下保持在重置狀態或執行。

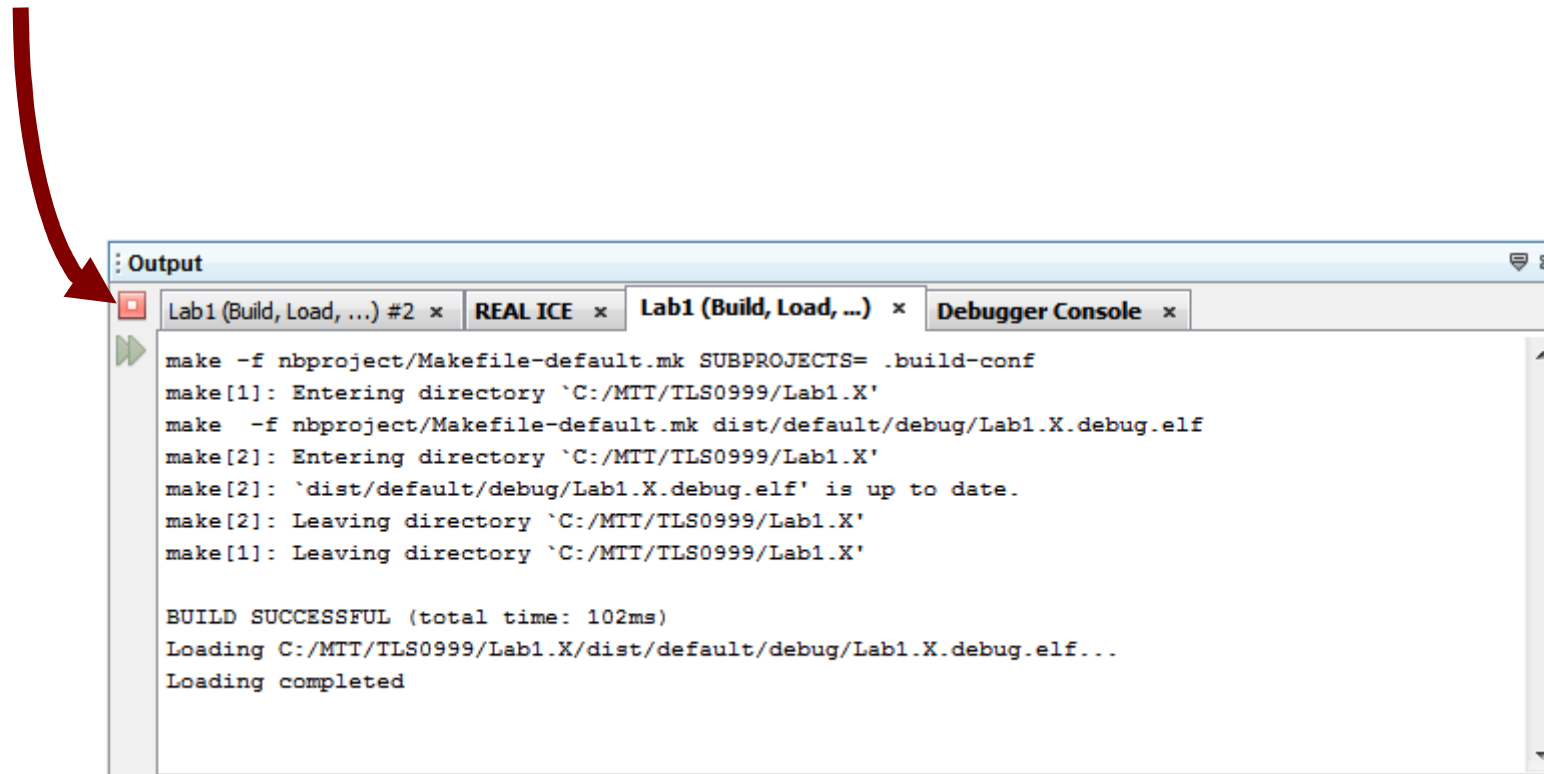


在大多數情況下，在按一下該按鈕之前，無需按一下 **“Build”** 或 **“Clean and Build”** 按鈕。

如何編譯應用程式

在編譯過程中停止編譯

- 按一下 **Output** 視窗左邊緣區域中的  圖示



如何編譯應用程式

陷阱——目前.....

- 若程式或標頭檔是通過 **#include** 語句在程式中引用，而未出現在專案樹中，則 **make** 目前無法檢測到對這些檔的修改，因此在進行 **Debug** 或 **Program** 之前，必須先執行 **Clean and Build**（清除並編譯）
- 也就是編譯器不知道檔案是否有被改過！

如何編譯應用程式

陷阱——與平臺相關的錯誤

- 相對路徑應使用 “/” 而不是 “\”：

#include "Graphics/Graphics.h"

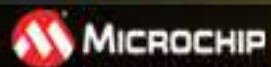
- “/” 在所有平臺中均可識別，包括 Windows®
- “\” 僅能被 Windows 識別
- Linux 平臺中的編譯器無法查找到路徑中採用 “\” 的檔，在 Linux 中該符號是轉義字元

- **Linux 區分大小寫**

- 採用 **# include <p24fj128ga010.h>** 無法找到文件，因為檔案名為 **p24FJ128GA010.h**

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



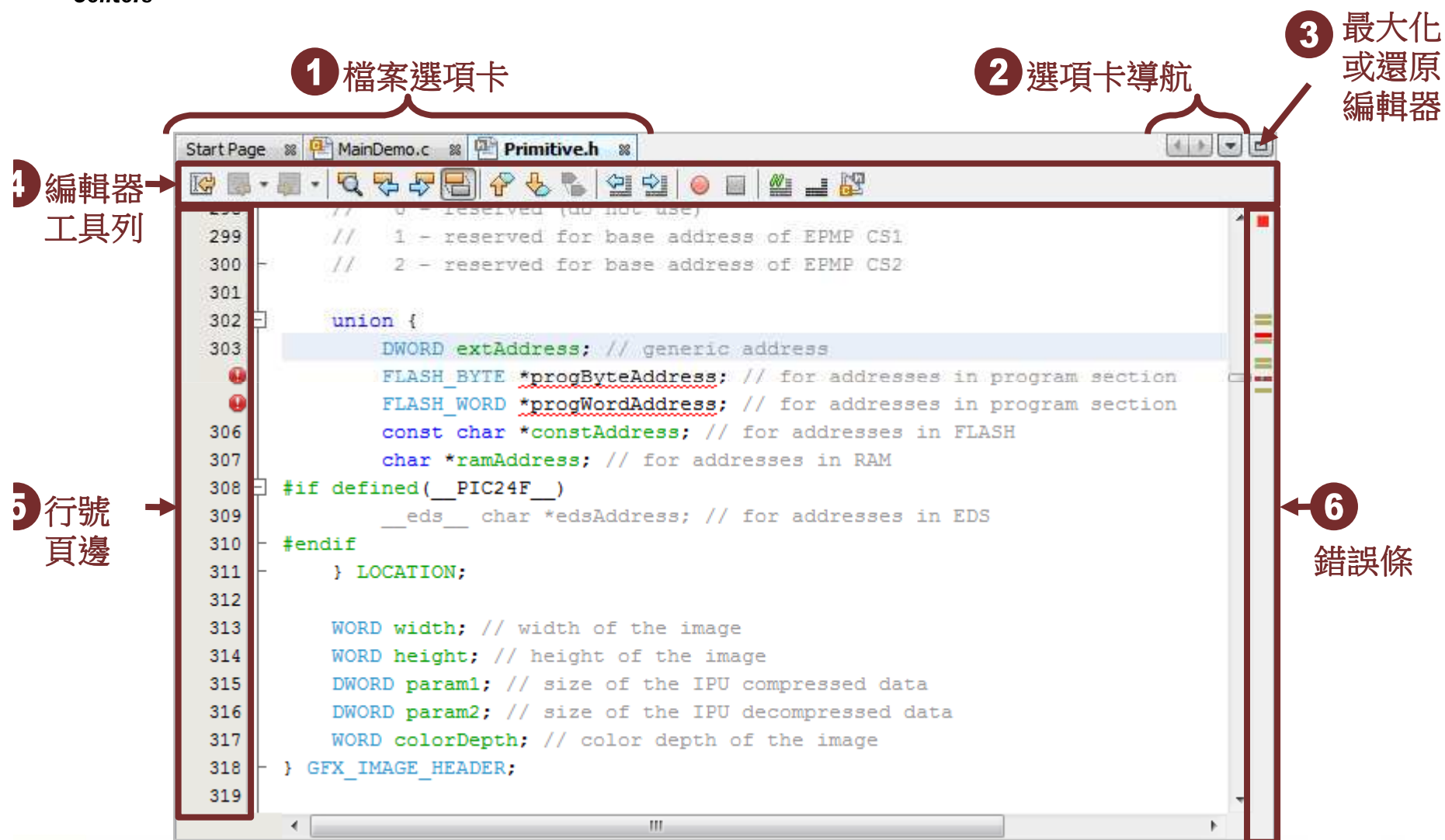
ONE PIC[®] MCU PLATFORM



如何使用編輯器

如何使用編輯器

編輯器的各部分



如何使用編輯器

特性：編輯器工具列圖示



- | | |
|--|---|
|  Last Edit (最近一處編輯)
——在編輯記錄之間迴圈 |  Shift Line Left (Indent Less)
(左移程式行 (減少縮進)) |
|  Previous Edit (上一處編輯) |  Shift Line Right (Indent More)
(右移程式行 (增加縮進)) |
|  Next Edit (下一處編輯) |  Start Macro Recording
(啟動巨集錄製) |
|  Find Selection (查找選擇項) |  Stop Macro Recording
(停止巨集錄製) |
|  Find Previous Occurrence
(查找上一處) |  Comment (註解) |
|  Find Next Occurrence (查找下一處) |  Uncomment (取消註解) |
|  Toggle Highlight Search
(切換高亮搜索項) |  Go to Header/Source
(轉至頭/原始檔案) |
|  Previous Bookmark (上一個書籤) | |
|  Next Bookmark (下一個書籤) | |
|  Toggle Bookmark (切換書籤) | |

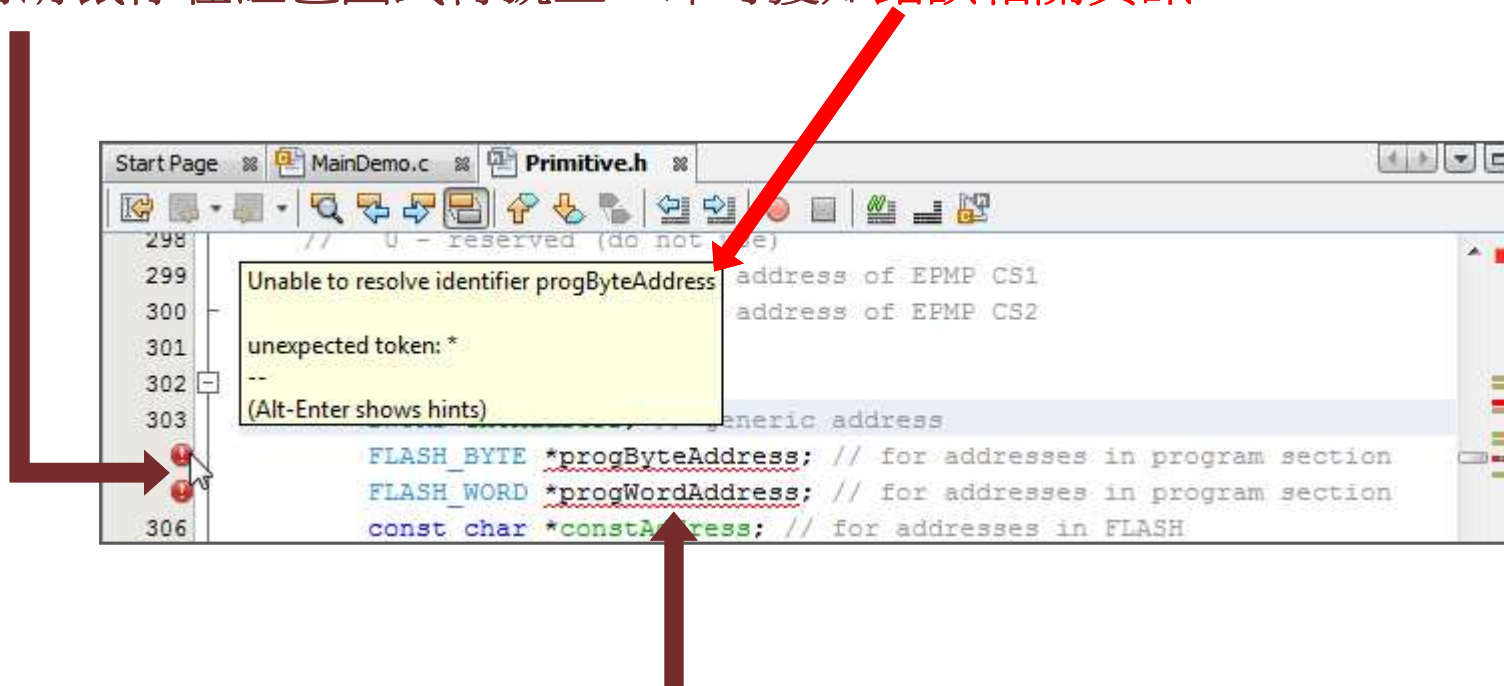
* Next Edit和Previous Edit按钮在使用**Navigate** (導航) ▶ **Go to X...** (轉至X.....) 之後啟動

如何使用編輯器

特性：即時解析

- 編輯器解析輸入的程式並立即標示出錯誤

將滑鼠停在紅色圖式符號上，即可獲知錯誤相關資訊

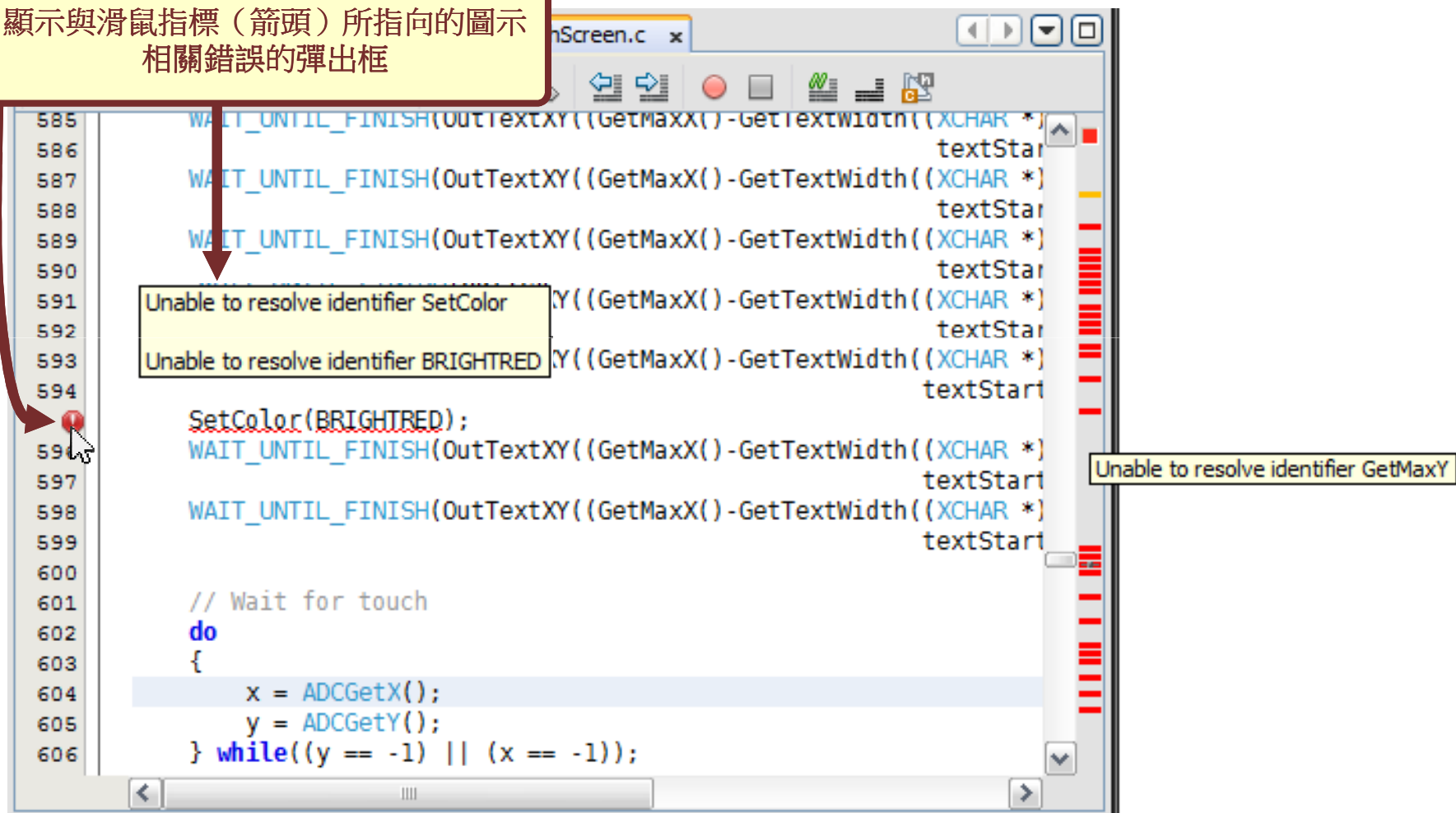


錯誤由紅底線標注（就像文字處理工具標注拼寫錯誤的文字段落一樣）

如何使用編輯器

特性：圖式符號頁邊行號

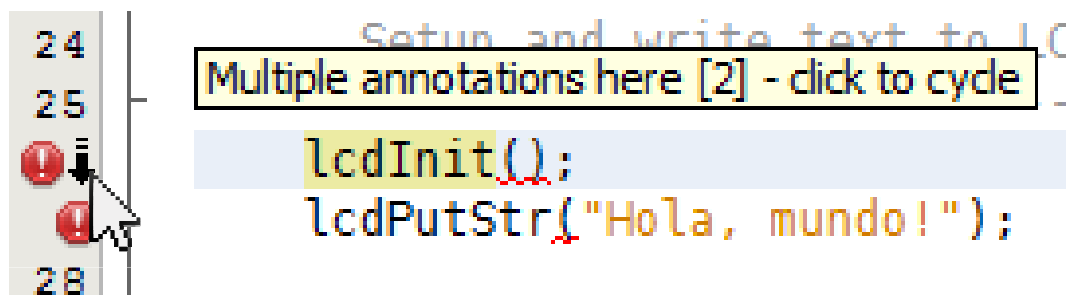
顯示與滑鼠指標（箭頭）所指向的圖式
相關錯誤的彈出框



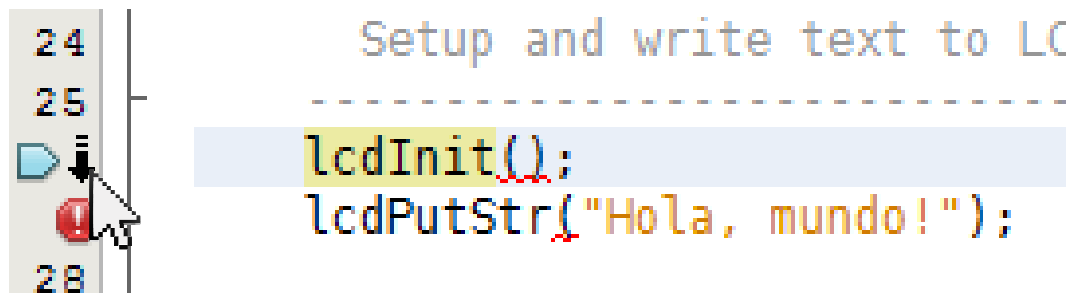
如何使用編輯器

特性：圖式符號頁邊

錯誤和書籤在同一行



當一行需要多個圖符時，會在一個圖符的右邊顯示一個小的黑色向下箭頭。



按一下這個箭頭即可迴圈顯示每個圖符。

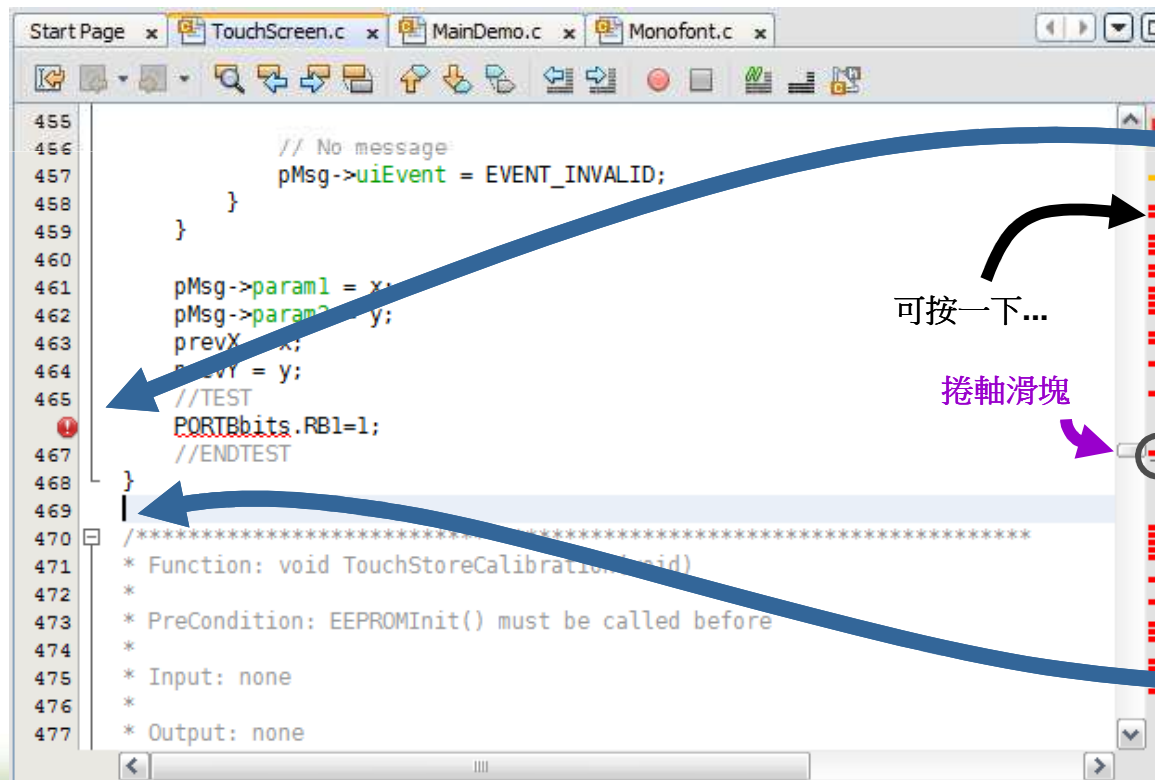
如何使用編輯器

導航：錯誤條

- 代表檔的完整長度
- 顯示錯誤、警告和當前出現高亮項的相對位置

顏色程式

	警告
	錯誤
	高亮項
	中斷點
	書籤
	建議
	多項



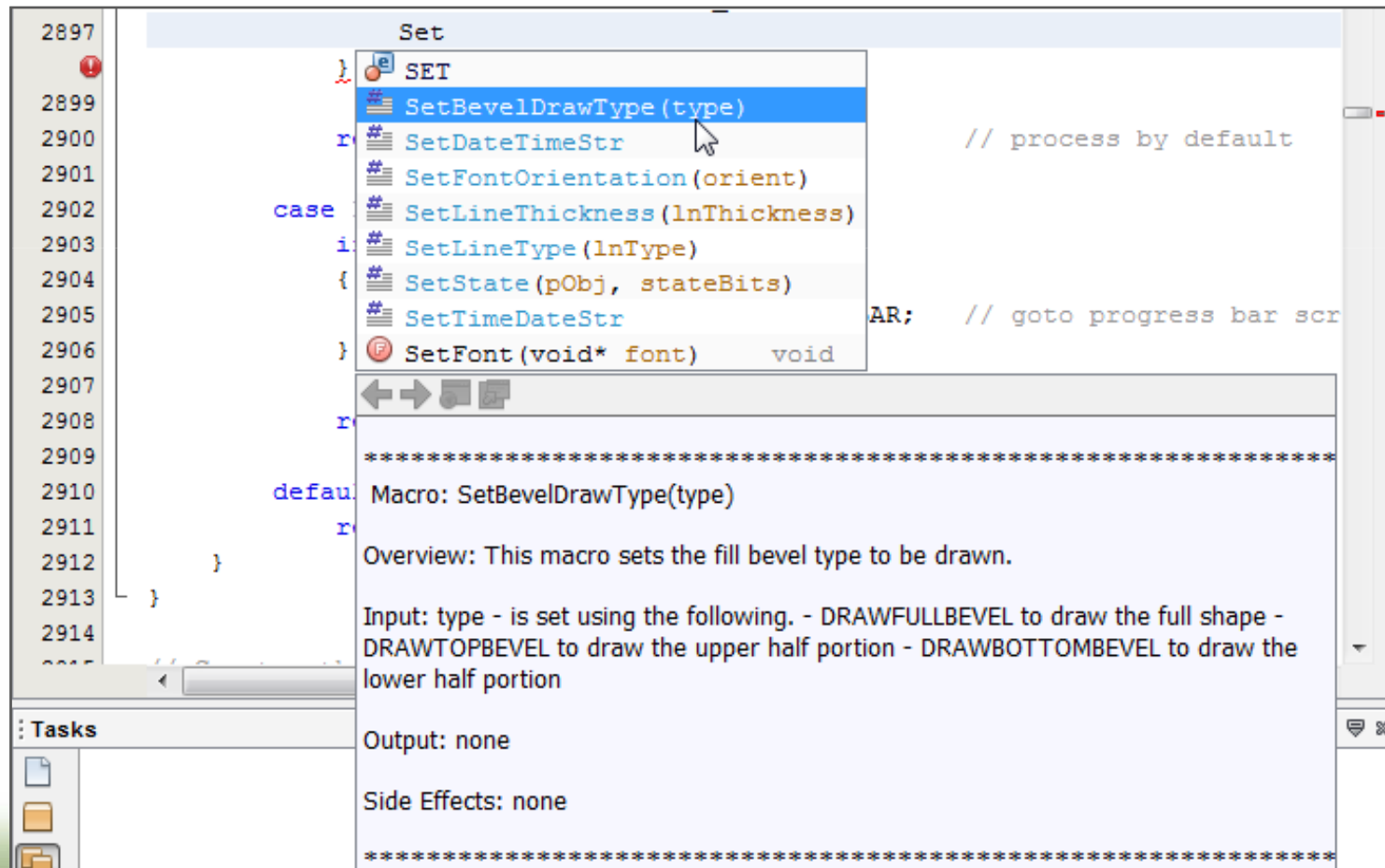
錯誤

目前游標
所在位置

如何使用編輯器

特性：智慧型函式或變數關聯字顯示

- 在輸入了識別字的一部分後按下 **Ctrl + 'I'** 後，相關的函式或變數即可智慧型顯示以供選用 (**Ctrl + Alt + Space**)



如何使用編輯器

特性：函式參數的宣告顯示

- 編輯器會在工具提示中顯示您輸入的函式的參數清單（可通過按下 **Ctrl** + **P** 顯示）

2 工具提示自動彈出，但需要您按下 **Ctrl+P** 手動顯示

```
38 //Type casting required because library is built with  
39 //MPLAB defaults unsigned char, unsigned char  
40 LCD_Set_Cursor(0,0);  
41 putrsLCD((const char char*)" X IDE Lab 2 ");
```

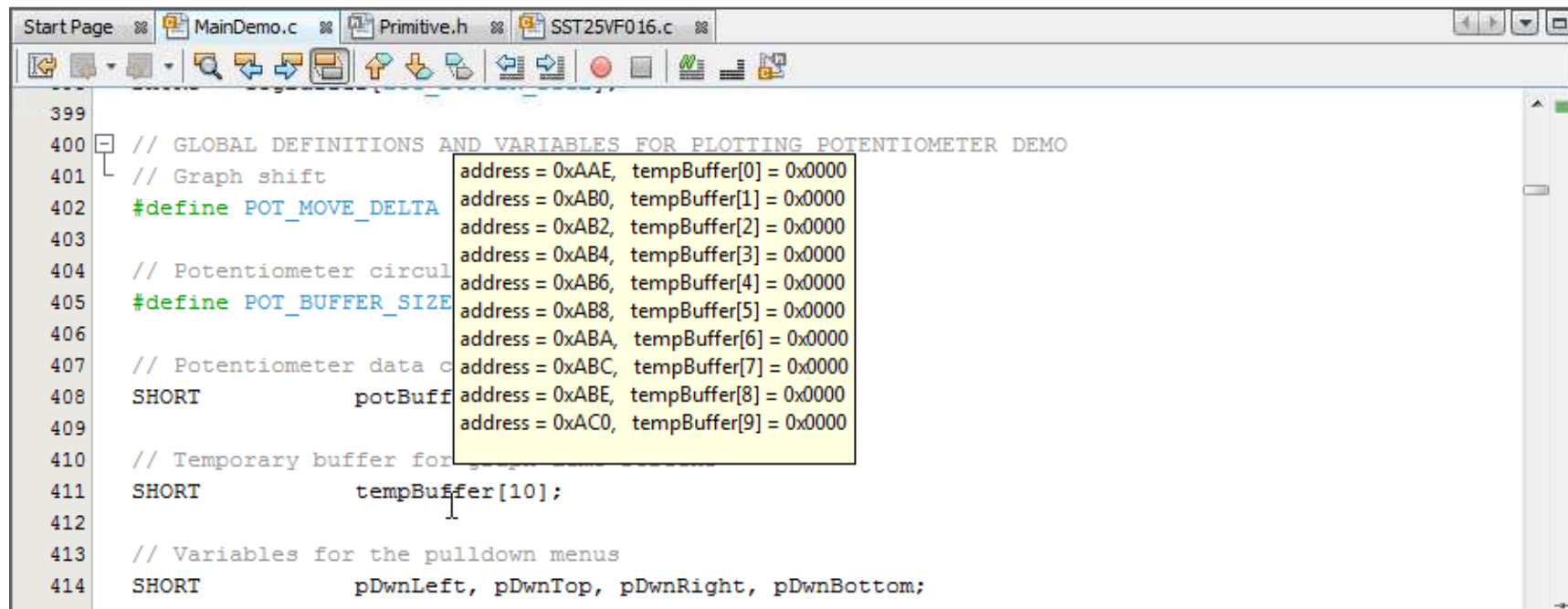
1

輸入函式名稱和開始括弧“(”

如何使用編輯器

特性：查看變數內容

- 需在除錯模式下
- 將滑鼠停留在暫存器或變數上.....



The screenshot shows the MPLAB X IDE interface. The main window displays a C source file named 'MainDemo.c'. The code is at line 400, showing global definitions and variables for a potentiometer demo. A tooltip is visible over the variable 'tempBuffer[0]', displaying its memory address (0xAAE) and value (0x0000). The tooltip also lists other variables in the array, such as 'tempBuffer[1]' through 'tempBuffer[9]', all with values of 0x0000. The code in the background includes comments like '// Graph shift', '#define POT_MOVE_DELTA', and '// Potentiometer circula'. The IDE's toolbar and project explorer are also visible.

```
399
400 // GLOBAL DEFINITIONS AND VARIABLES FOR PLOTTING POTENTIOMETER DEMO
401 // Graph shift
402 #define POT_MOVE_DELTA
403
404 // Potentiometer circula
405 #define POT_BUFFER_SIZE
406
407 // Potentiometer data c
408 SHORT      potBuff
409
410 // Temporary buffer for
411 SHORT      tempBuffer[10];
412
413 // Variables for the pulldown menus
414 SHORT      pDwnLeft, pDwnTop, pDwnRight, pDwnBottom;
```

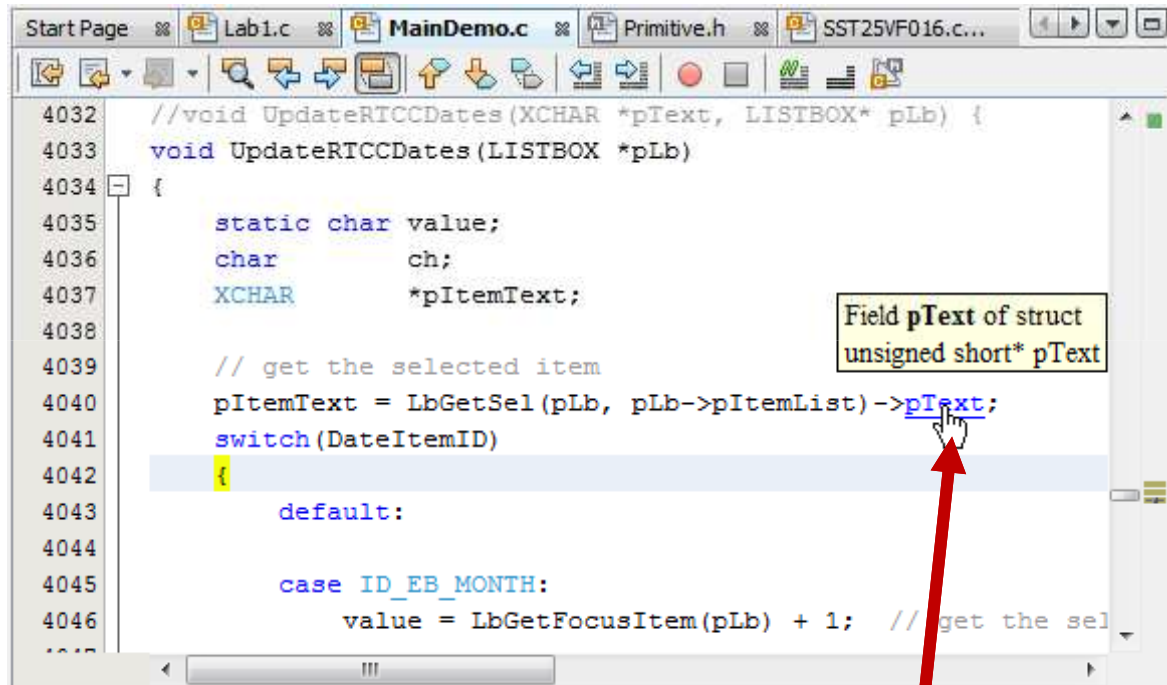
address = 0xAAE, tempBuffer[0] = 0x0000
address = 0xAB0, tempBuffer[1] = 0x0000
address = 0xAB2, tempBuffer[2] = 0x0000
address = 0xAB4, tempBuffer[3] = 0x0000
address = 0xAB6, tempBuffer[4] = 0x0000
address = 0xAB8, tempBuffer[5] = 0x0000
address = 0xABA, tempBuffer[6] = 0x0000
address = 0xABC, tempBuffer[7] = 0x0000
address = 0xABE, tempBuffer[8] = 0x0000
address = 0xAC0, tempBuffer[9] = 0x0000

如何使用編輯器

導航：帶超連結的識別字

要顯示連結：
將滑鼠停留在一個識別字
上，然後按下 **Ctrl** 鍵

按一下 連結 跳轉到編輯器
中聲明該識別字的地方

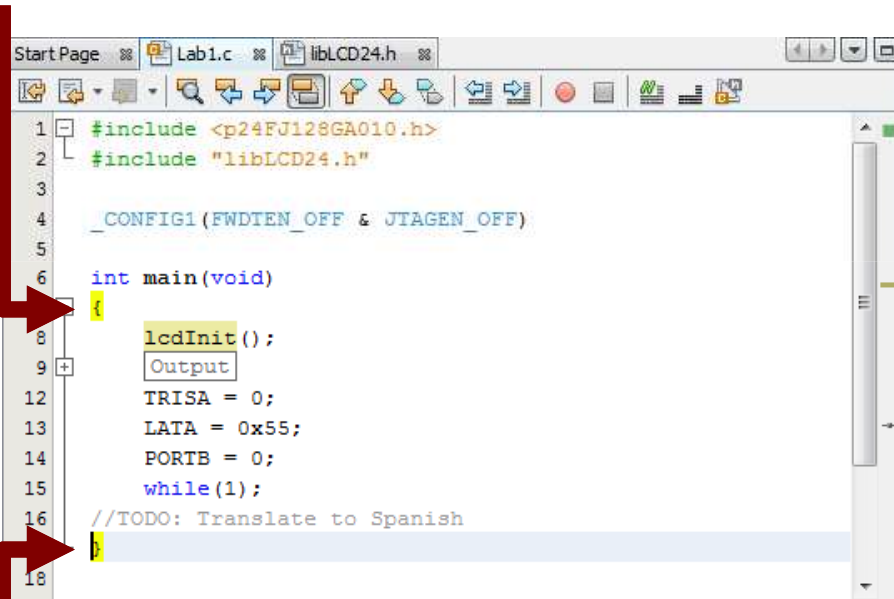


如何使用編輯器

導航：自動大括弧配對

選擇任何一個大括弧“{”或“}”，
然後這個括弧和與它配對的
括弧都會高亮顯示。

配對的大括弧



```
1 #include <p24FJ128GA010.h>
2 #include "libLCD24.h"
3
4 _CONFIG1(FWDTEN_OFF & JTAGEN_OFF)
5
6 int main(void)
7 {
8     lcdInit();
9     Output
10
11     TRISA = 0;
12     LATA = 0x55;
13     PORTB = 0;
14     while(1);
15
16     //TODO: Translate to Spanish
17 }
18
```

選定的大括弧

如何使用編輯器

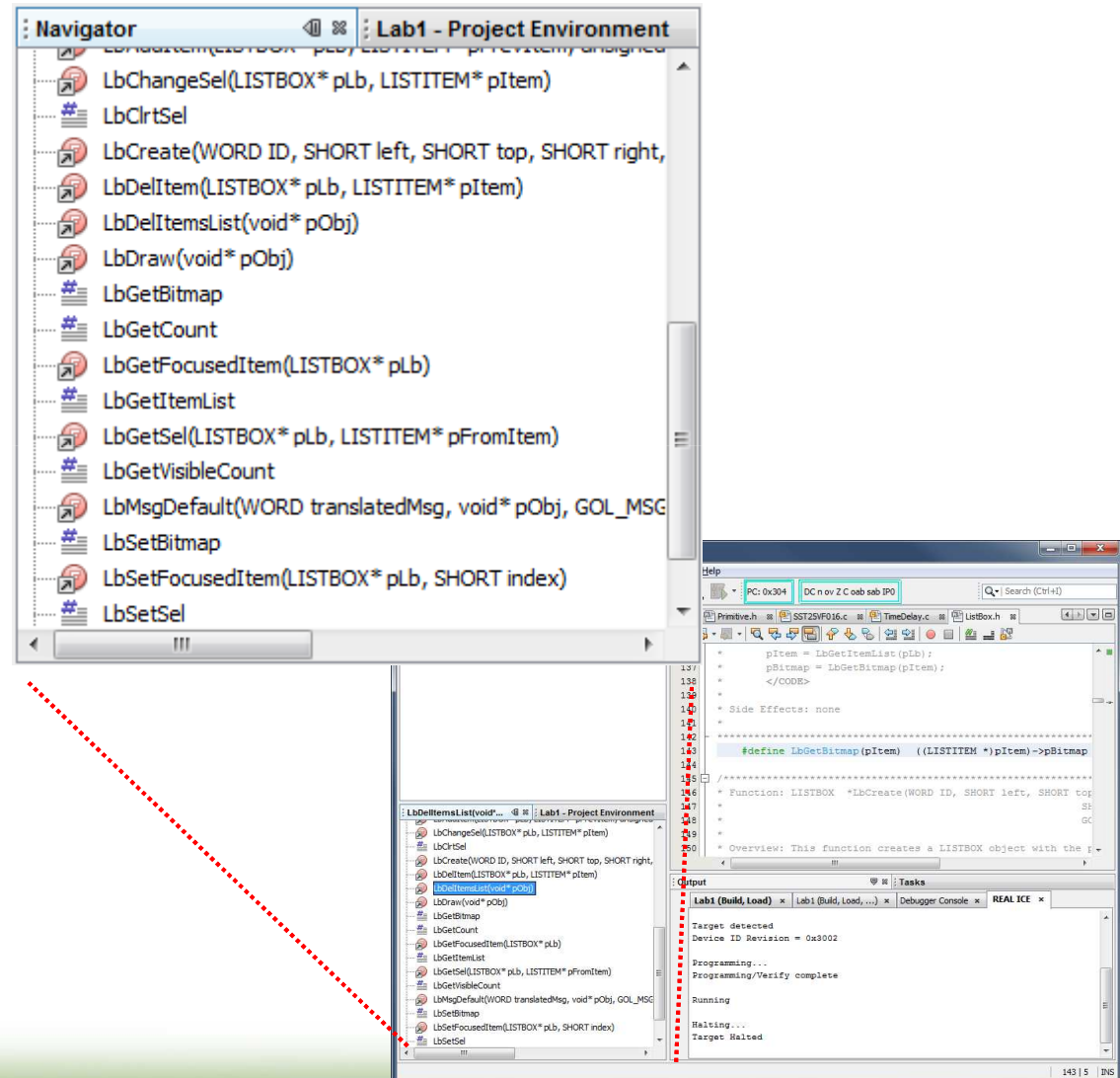
導航：Go To...（轉至）

- 轉至聲明
- 轉至原始程式
- 轉至類型
- 轉至文件
- 轉至符號
- 轉至行號

如何使用編輯器

導航：Navigator視窗

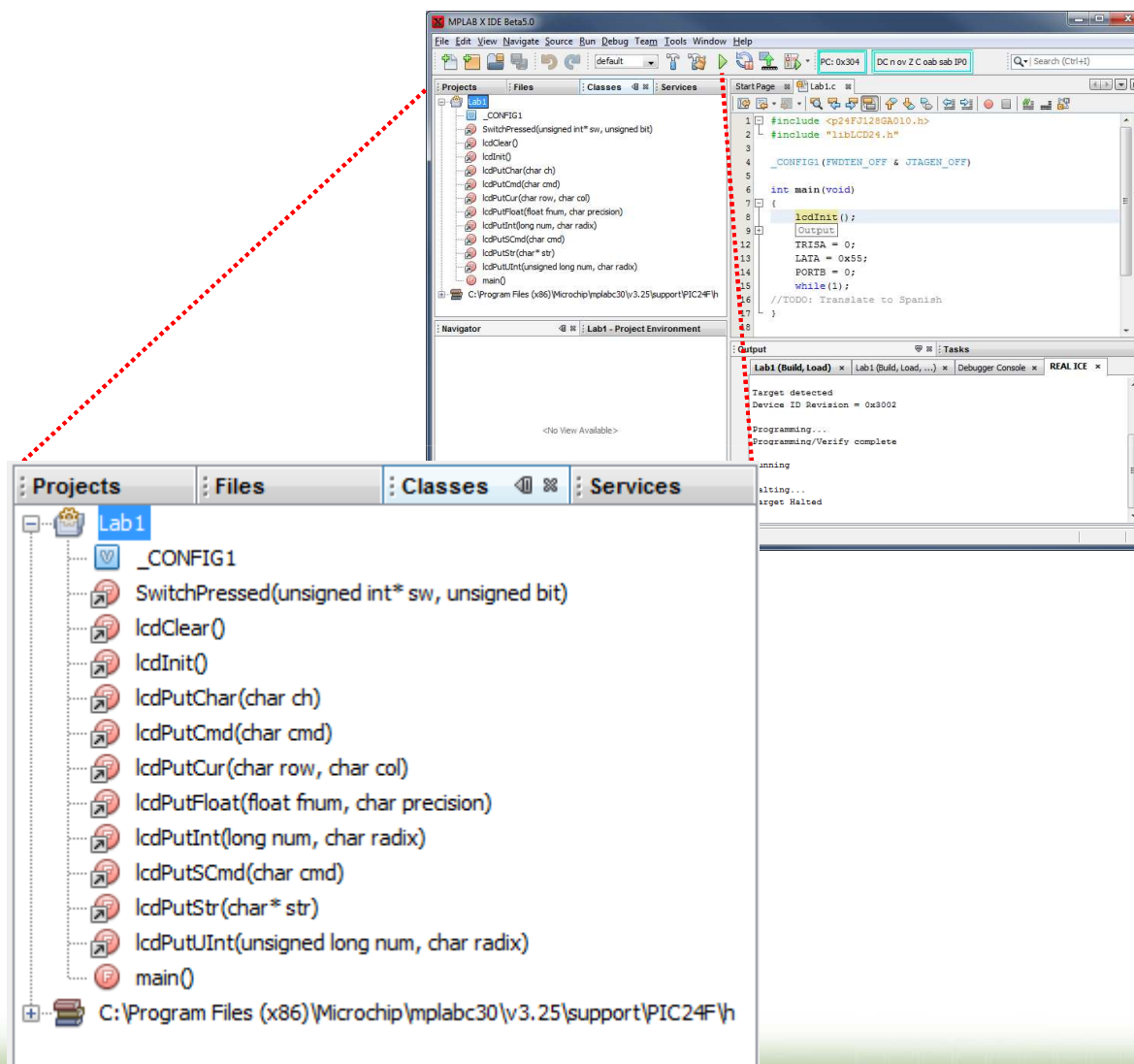
- 列出專案裡所用到的
的識別字、函式名稱
和符號
- 滑鼠移置選項後按兩
下，即可跳轉到其聲
明所在位置
- 按右鍵查看更多選項
 - **Go to source**
 - **Rename**
 - **Find usages**
 - 更多 ...



如何使用編輯器

導航：Classes視窗

- 列出專案的識別字
- 按兩下一項即可跳轉到其聲明所在位置
- 按右鍵查看更多選項



如何使用編輯器

特性：指定程式的隱藏/展開

- 不使用{ }來縮減程式的顯示：

語法

```
//<editor-fold defaultstate="collapsed" desc="Descriptive Text">  
Your Code Here  
//</editor-fold>
```

未隱藏/展開

```
6  int main(void)  
7  {  
8      // <editor-fold defaultstate=  
9      lcdInit();  
10     // </editor-fold>  
11     lcdPutStr("Hello, world!");  
12     TRISA = 0;  
13     LATA = 0x55;  
14     while(1);
```

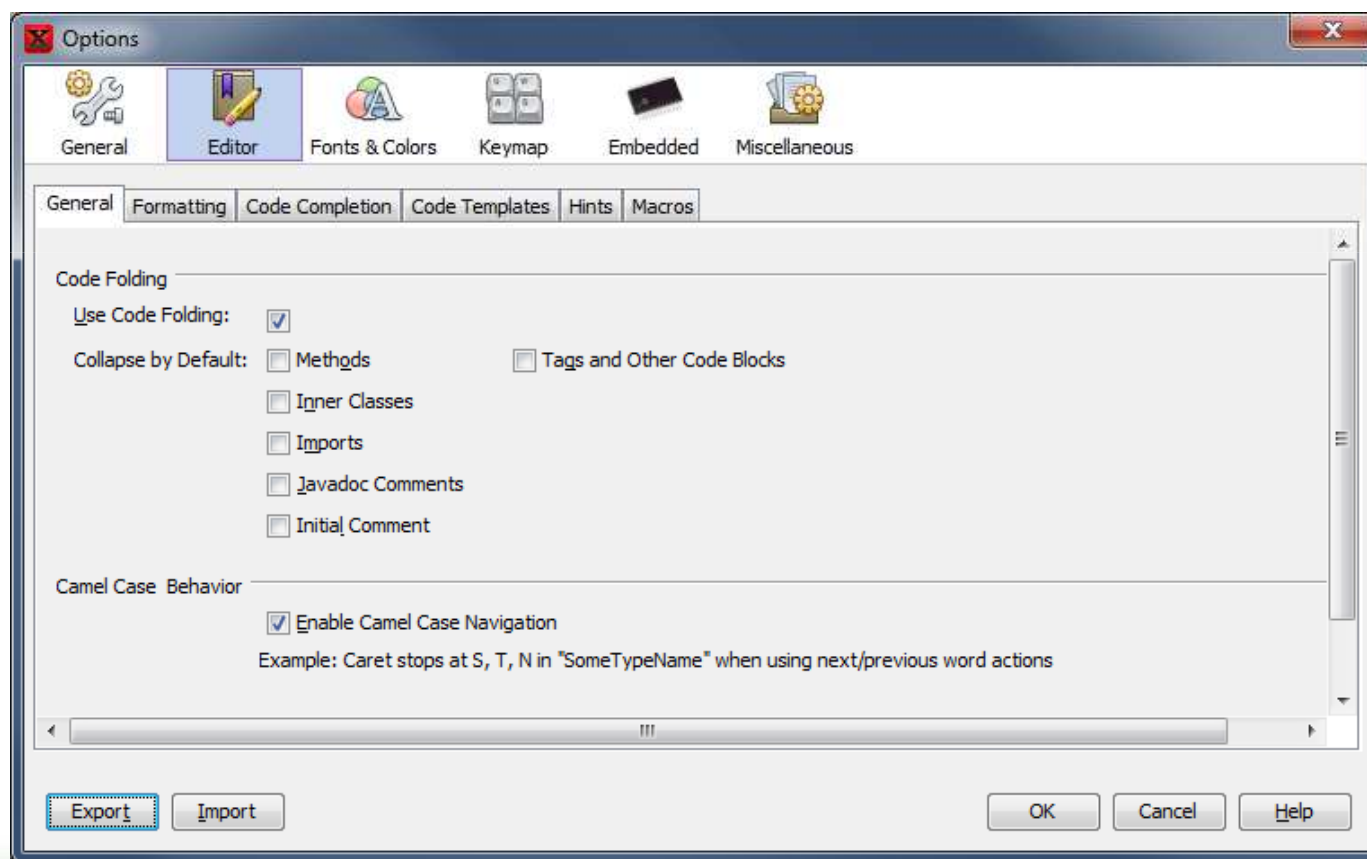
隱藏後

```
6  int main(void)  
7  {  
8      Initialization Code  
11     lcdPutStr("Hello, world!");  
12     TRISA = 0;  
13     LATA = 0x55;  
14     while(1);  
15 }  
16
```

如何使用編輯器

如何設定編輯器設置

- 從功能表中選擇 **Tools**（工具）▶ **Options**（選項），然後選擇頂部的 **Editor**（編輯器）圖示 

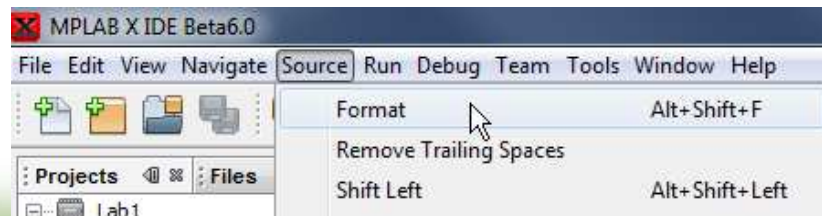
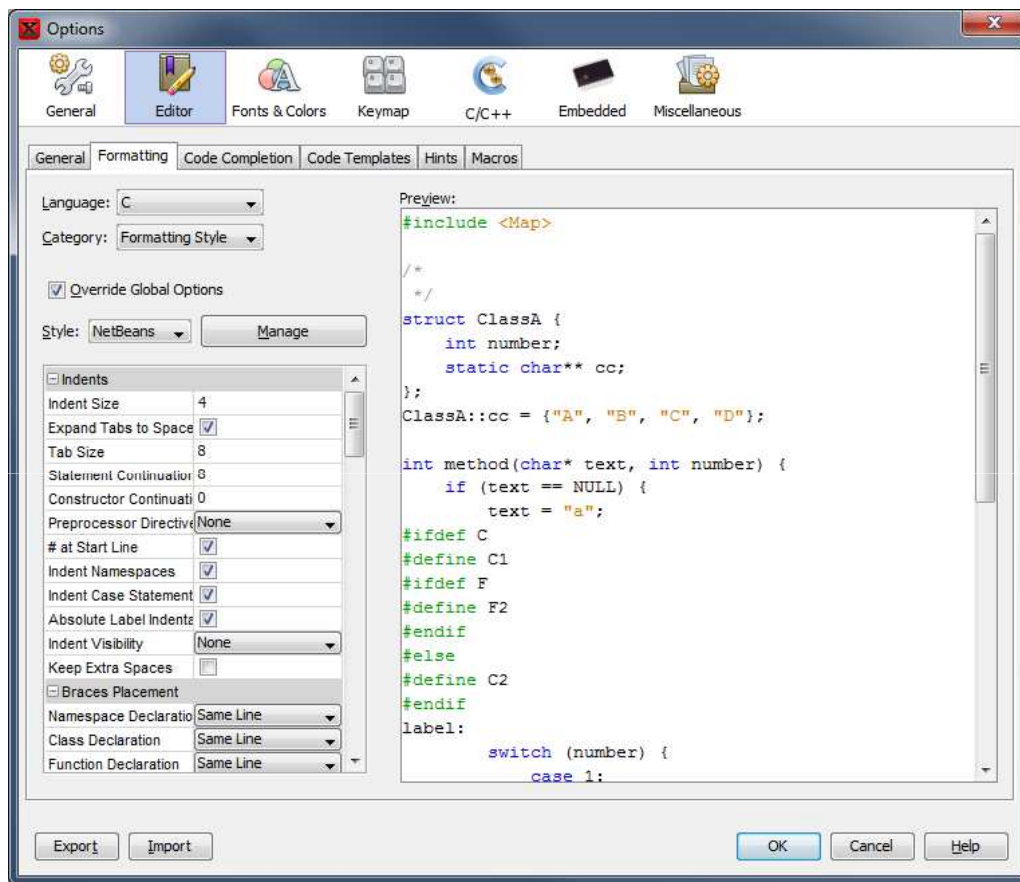


如何使用編輯器

自動程式格式

C 程式格式的規則可在 **Tools** (工具) ▶ **Options** (選項) ▶ **Editor** (編輯器) ▶ **Formatting** (格式) 中定義。從語言下拉式列示方塊中選擇 **C**。

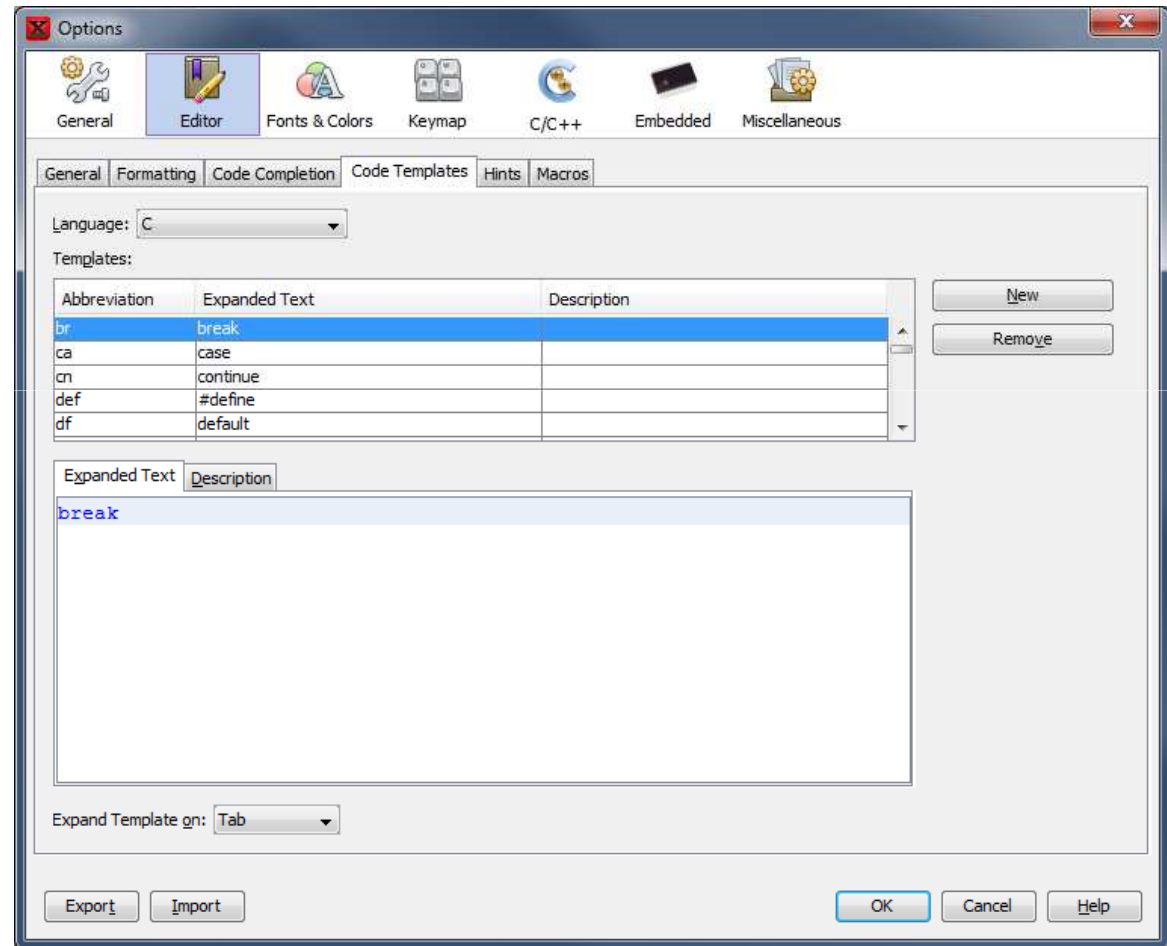
要將規則應用到在編輯器中當前選定的檔，從主功能表中選擇 **Source** (原始文件) ▶ **Format** (格式)。



如何使用編輯器 程式範本

程式範本提供了程式片段的簡略符號，程式片段將由編輯器展開。

程式範本可以接受參數，因此，無需移動游標即可方便地在程式中輸入參數的值。



如何使用編輯器

程式範本：範例

While 語句的範本定義

```
while (${EXP default="exp"}) {  
    ${selection line}${cursor}  
}
```

在編輯器中輸入以下資訊，然後按下**Tab**鍵：

```
wh
```

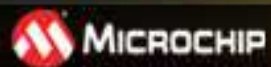
編輯器會將其展開為：

```
while (exp) {  
  
}
```

鍵入**exp**的值，然後按下**Enter**鍵。游標將自動被放置到大括弧內。

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



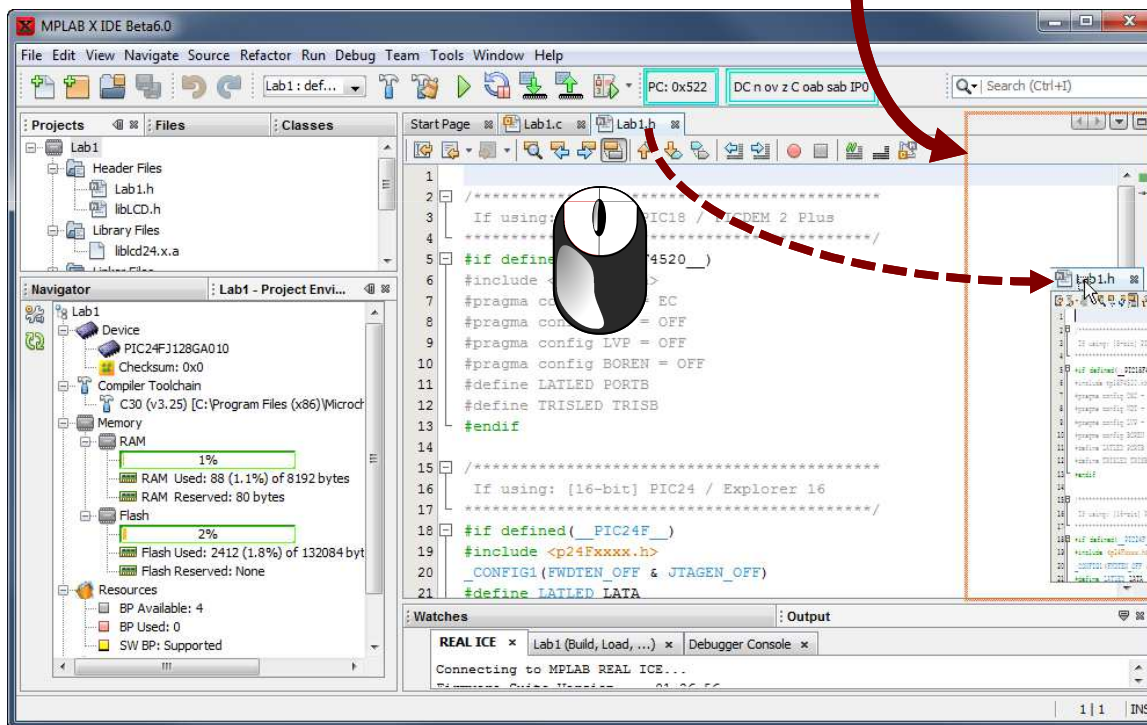
編輯視窗的檢視

併排檢視兩個文件

1 按一下選項卡的檔案並拖動至所需位置

在編輯器頂端有選項卡，將要併排檢視的原始程式檔案拖離至編輯視窗區域。

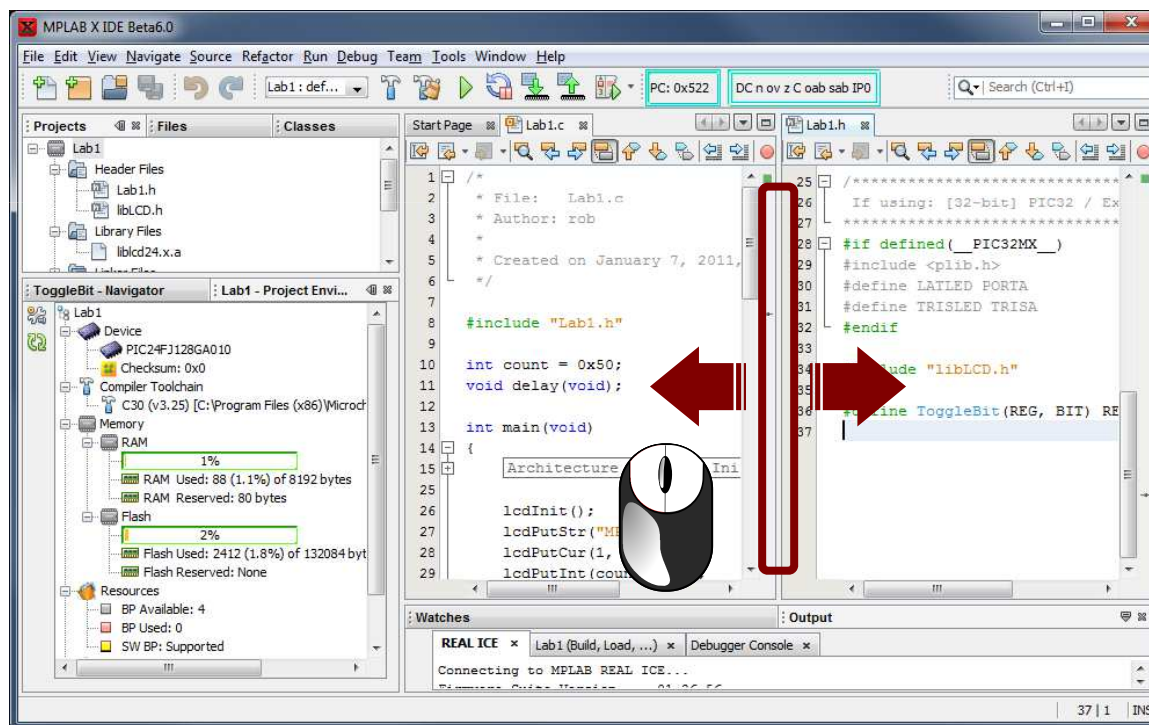
請留意其後的 **橘紅色邊框**。該邊框指示新選編及視窗將顯示的位置。



併排檢視兩個文件

2 釋放滑鼠停靠視窗

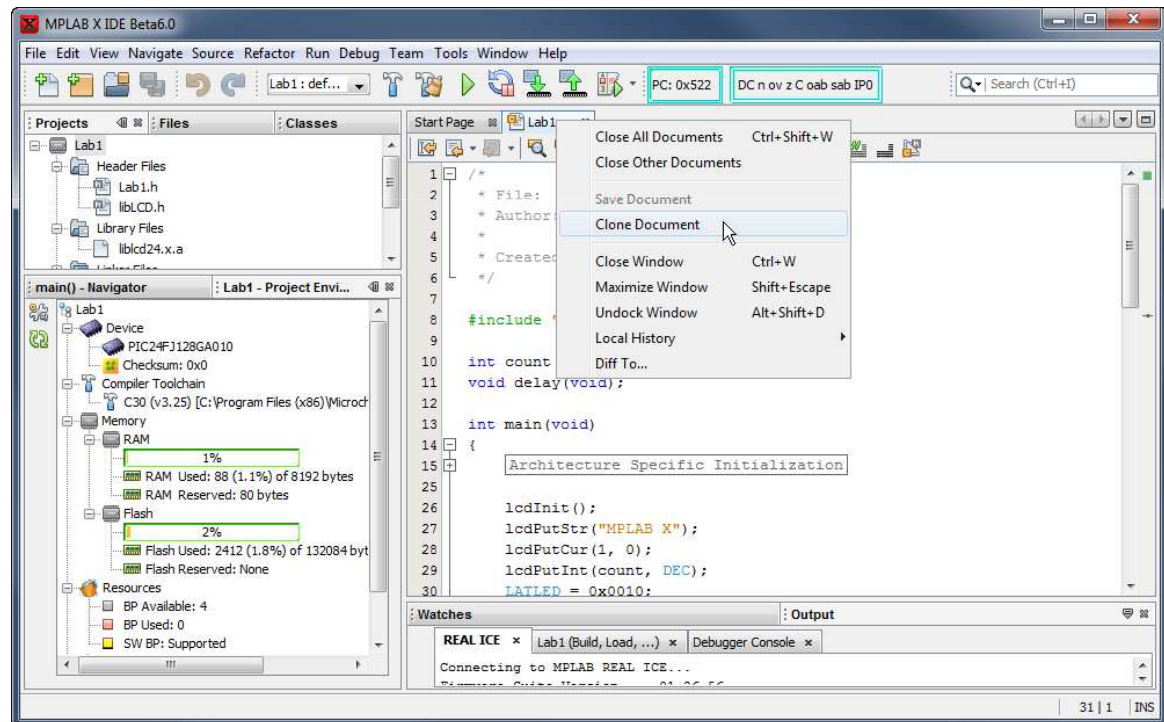
按一下並拖動兩個視窗間的豎條即可調節視窗的寬度。



單個文件使用雙視窗檢視

1 按右鍵選項卡並選擇 **Clone Document**（孿生文件）

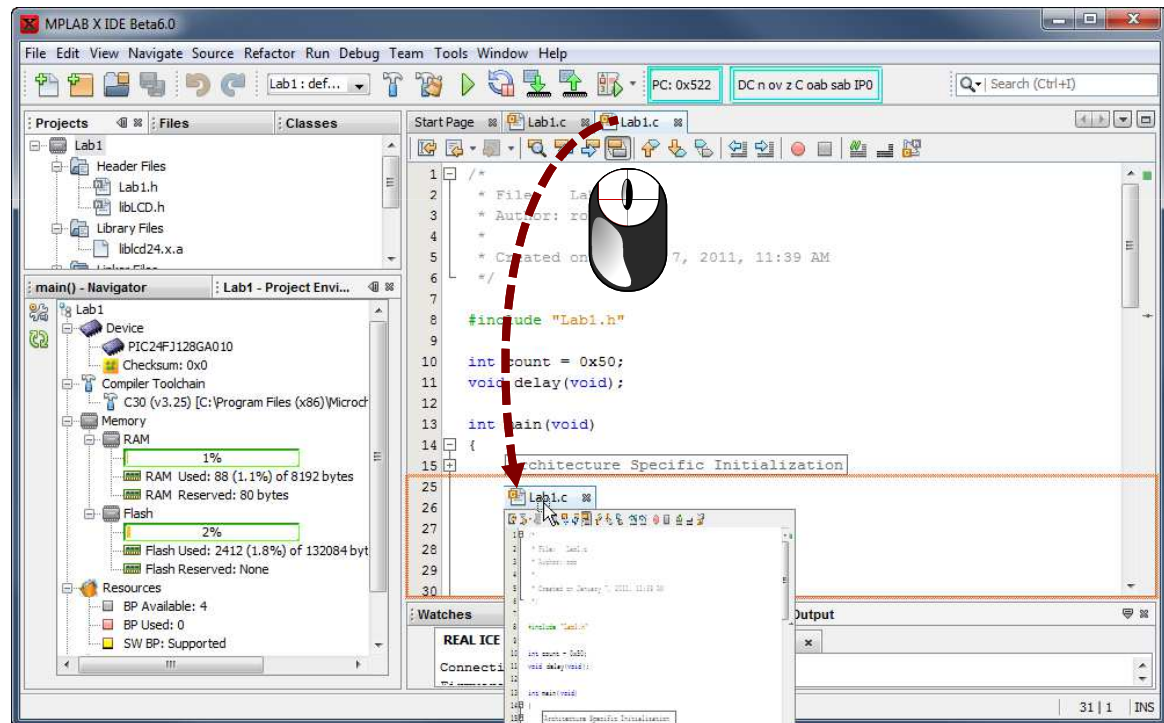
一般情況下一個原始程式檔案只能開起一個編輯視窗。文件檔 **Clone** 功能允許建立同一檔案的另一個編輯視窗，從而允許您在同一時間在不同位置查看同一檔案以利程式的編輯及檢查。



單個文件使用雙視窗檢視

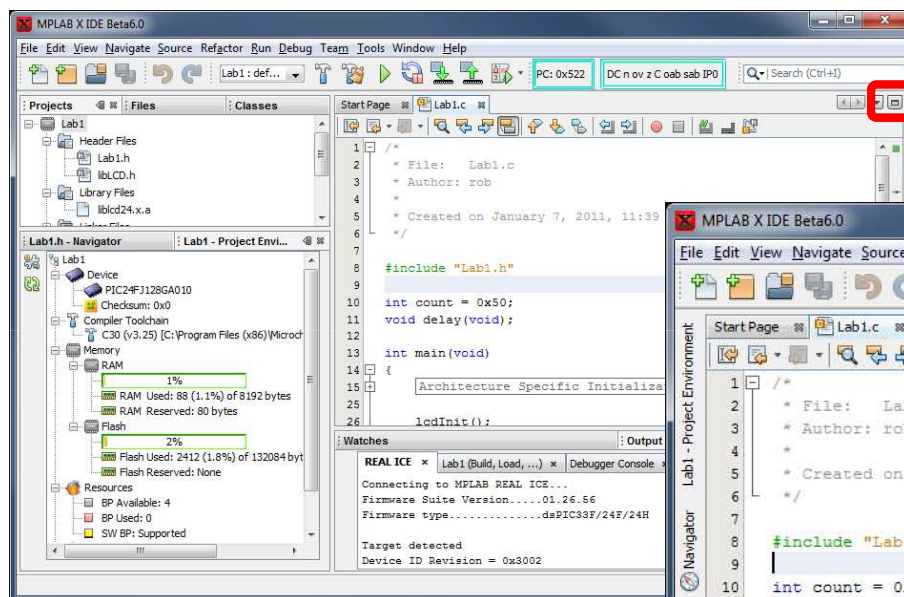
2 按一下選項卡並拖動至所需位置

一旦打開了 **Clone** 功能的文件檔視窗，就可以像其它選項卡一樣移動和停靠該文件檔的選項卡。



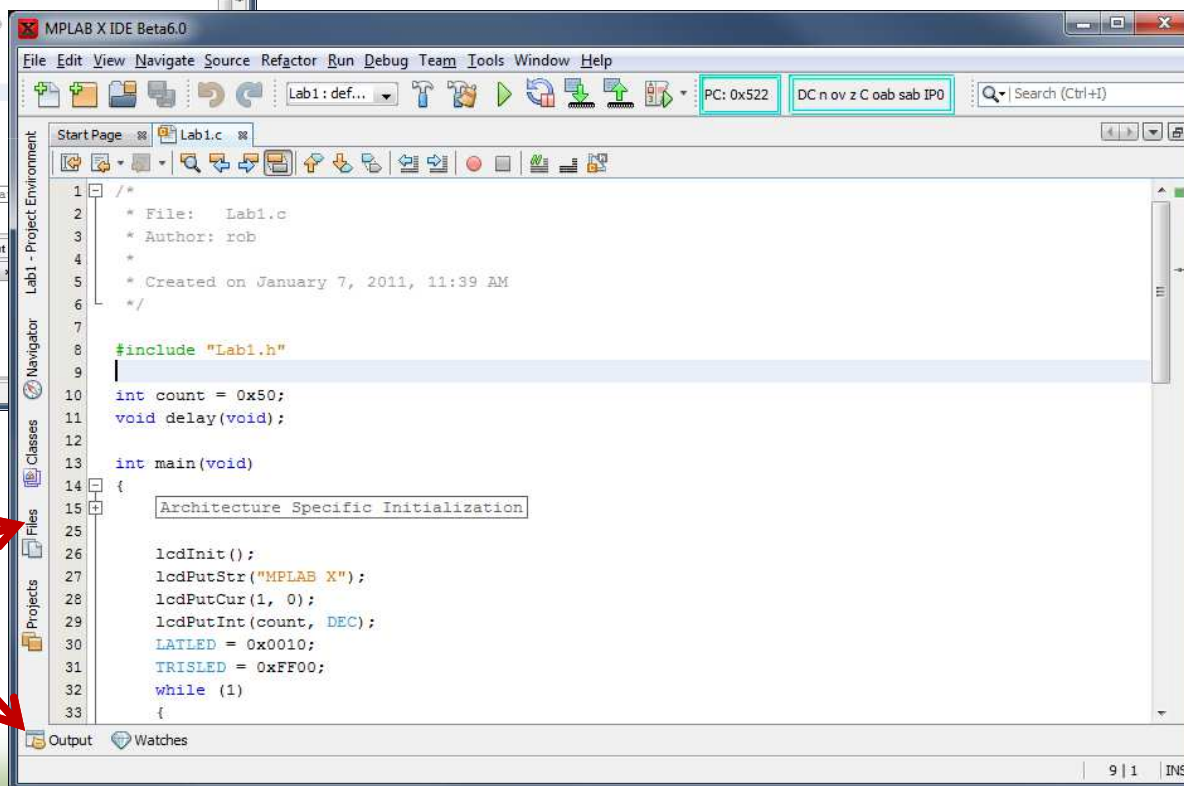
如何將編輯器視窗最大化

1 按一下編輯器的最大化視窗按鈕



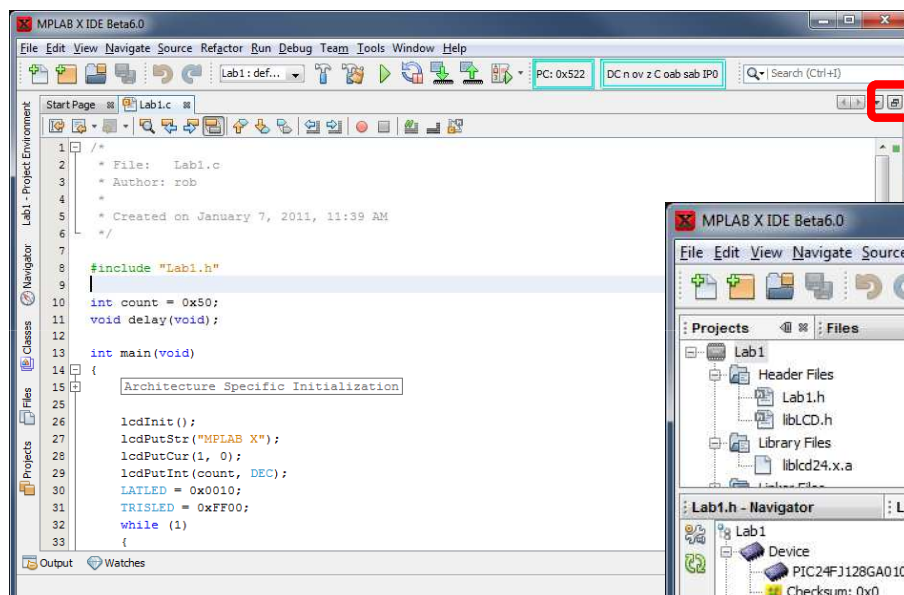
(或按一下想要最大化的選項)

當前視窗最大化後，選項卡會放置在新視窗側面和底部的條中。將滑鼠暫停其上即可臨時彈出相應的視窗/選項卡。

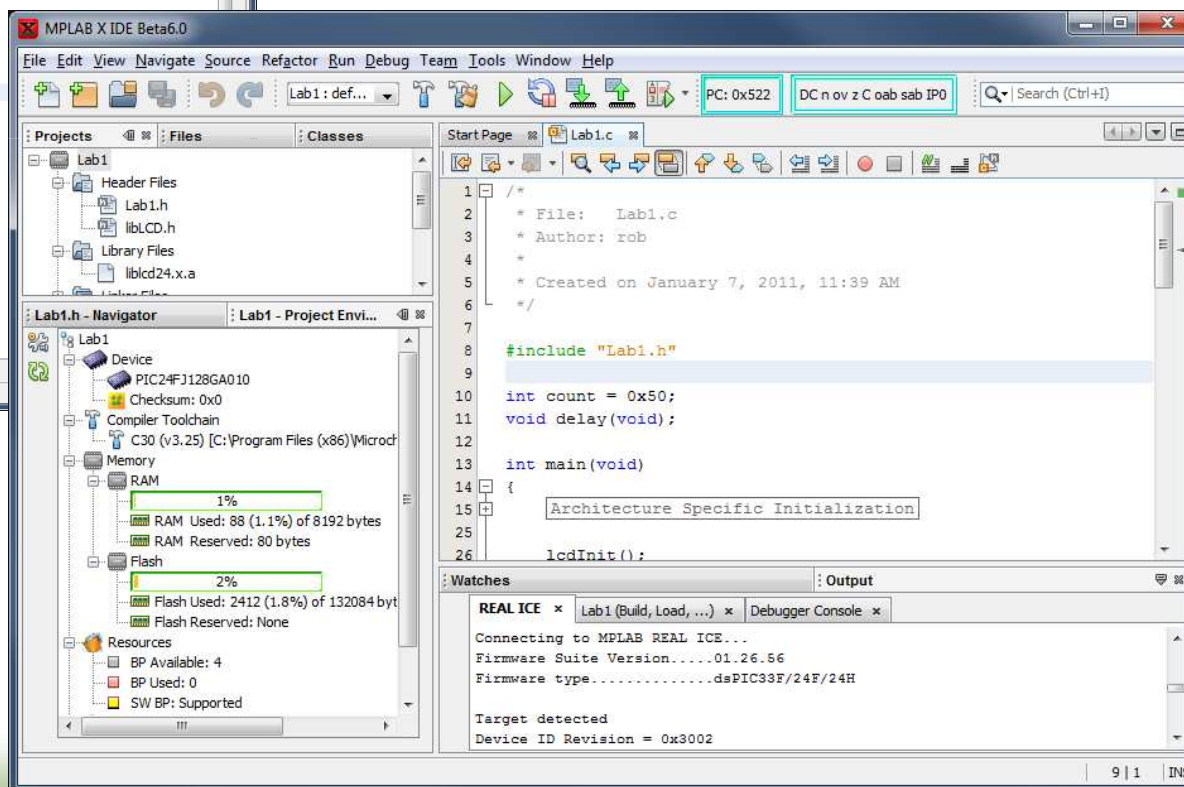


如何將編輯器視窗最小化

1 按一下編輯器的最小化視窗按鈕



(或按兩下選項卡，恢復最初的佈局)

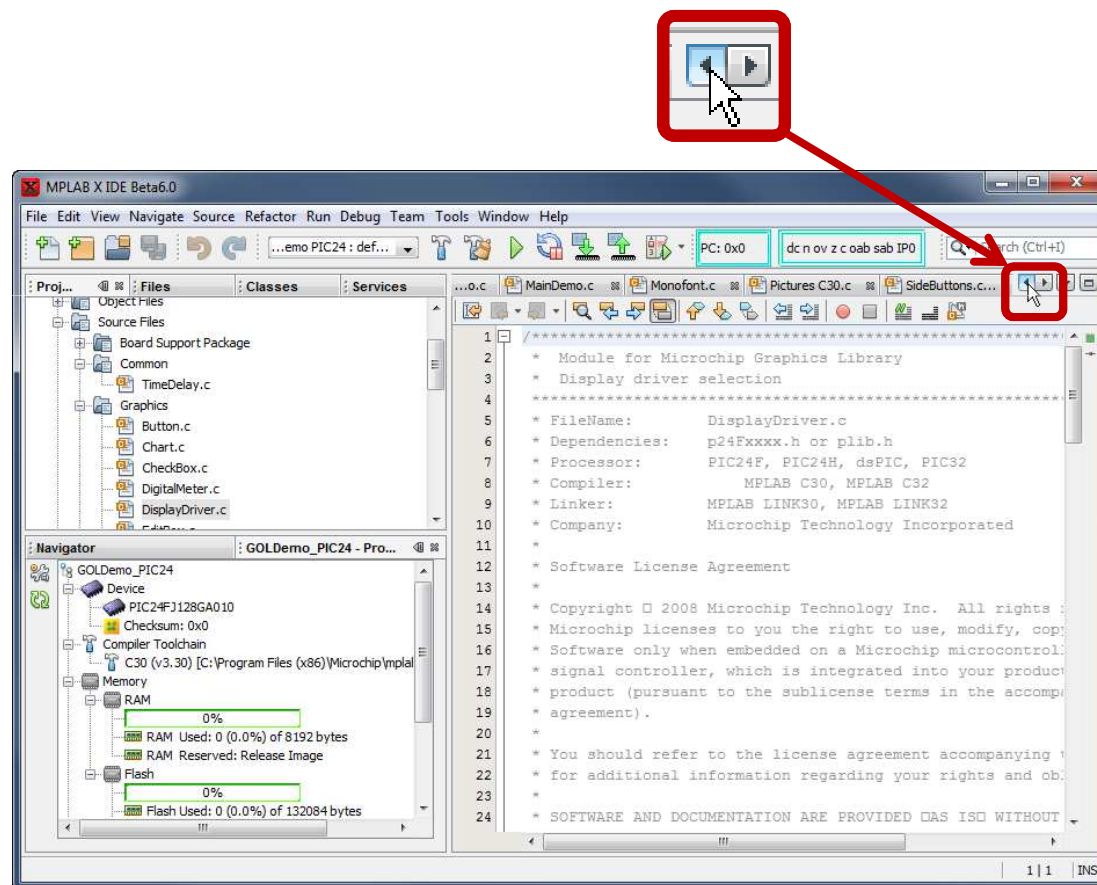


如何在選項卡之間檢索編輯檔案

方法一

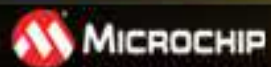
1 採用頂部的箭頭按鈕在選項卡間滾動

若打開的文件檔案數量過多，使得沒有足夠的空間顯示所有的編輯檔案，就可使用箭頭按鈕左右滾動選項卡，使隱藏的選項卡顯示出來。



Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



如何使用主工具列圖示

使用主工具列圖示

預設按鈕



New File (建立新文件)



New Project (建立新專案)



Open Project (打開專案)



Save All (全部儲存)



Undo (恢復)



Redo (重做)



Build (編譯)



Rebuild (Clean and Build)
重新編譯 (清除並編譯)



Program Target (燒錄目標元件)



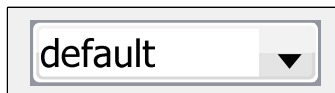
Read Target (讀取目標元件)



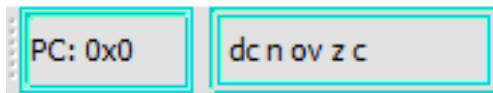
Hold in Reset (保持重置)



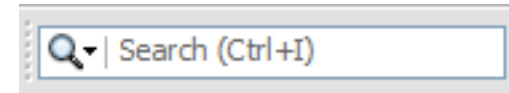
Debug (Build, Program, Run)
除錯模式 (編譯、燒錄並執行)



專案設定



程式計數器和
狀態位元



快速搜索

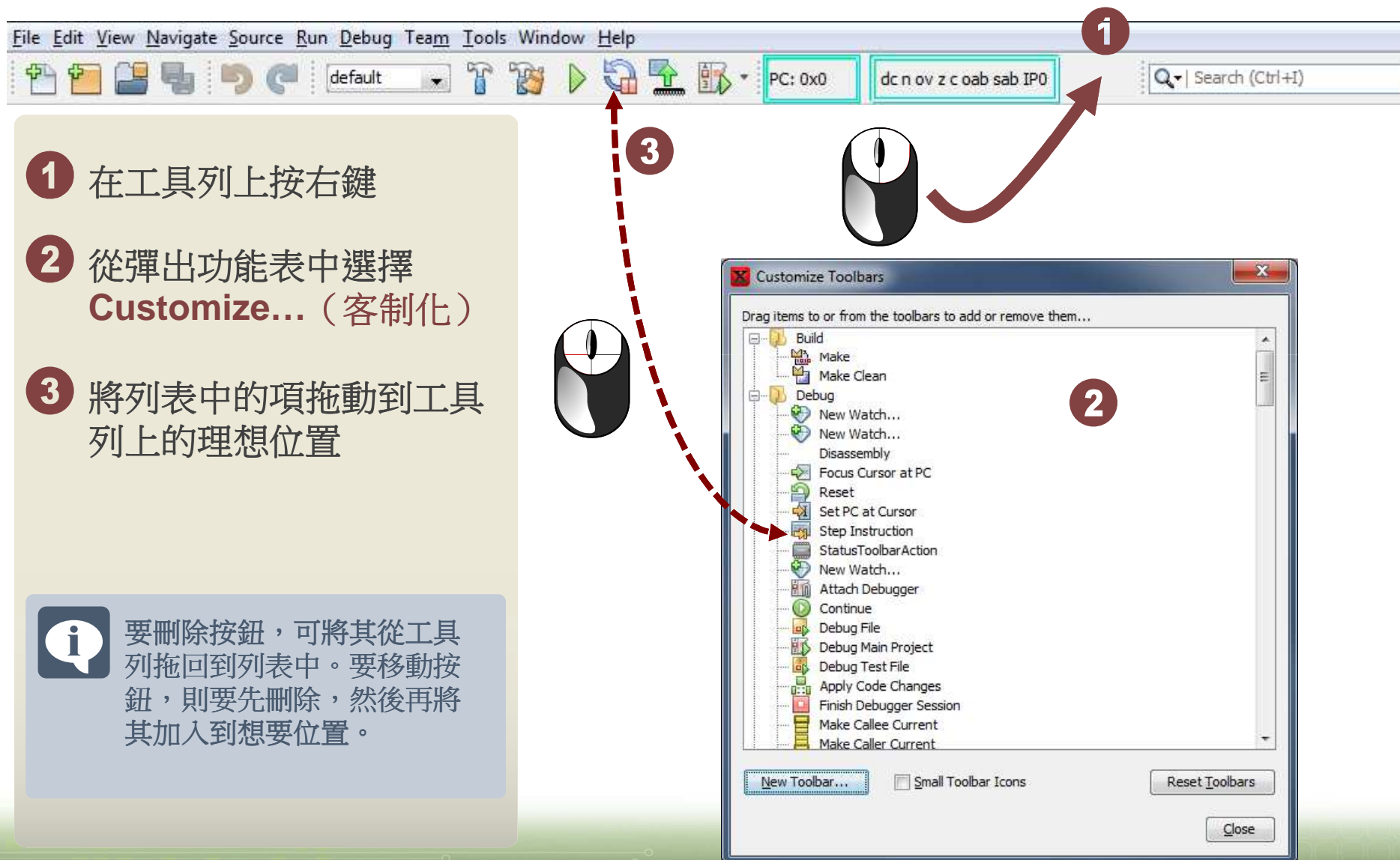
如何使用主工具列圖示

如何自訂客制化工具列

- 1 在工具列上按右鍵
- 2 從彈出功能表中選擇 **Customize...** (客制化)
- 3 將列表中的項拖動到工具列上的理想位置

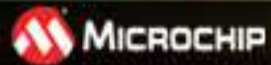


要刪除按鈕，可將其從工具列拖回到列表中。要移動按鈕，則要先刪除，然後再將其加入到想要位置。



Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



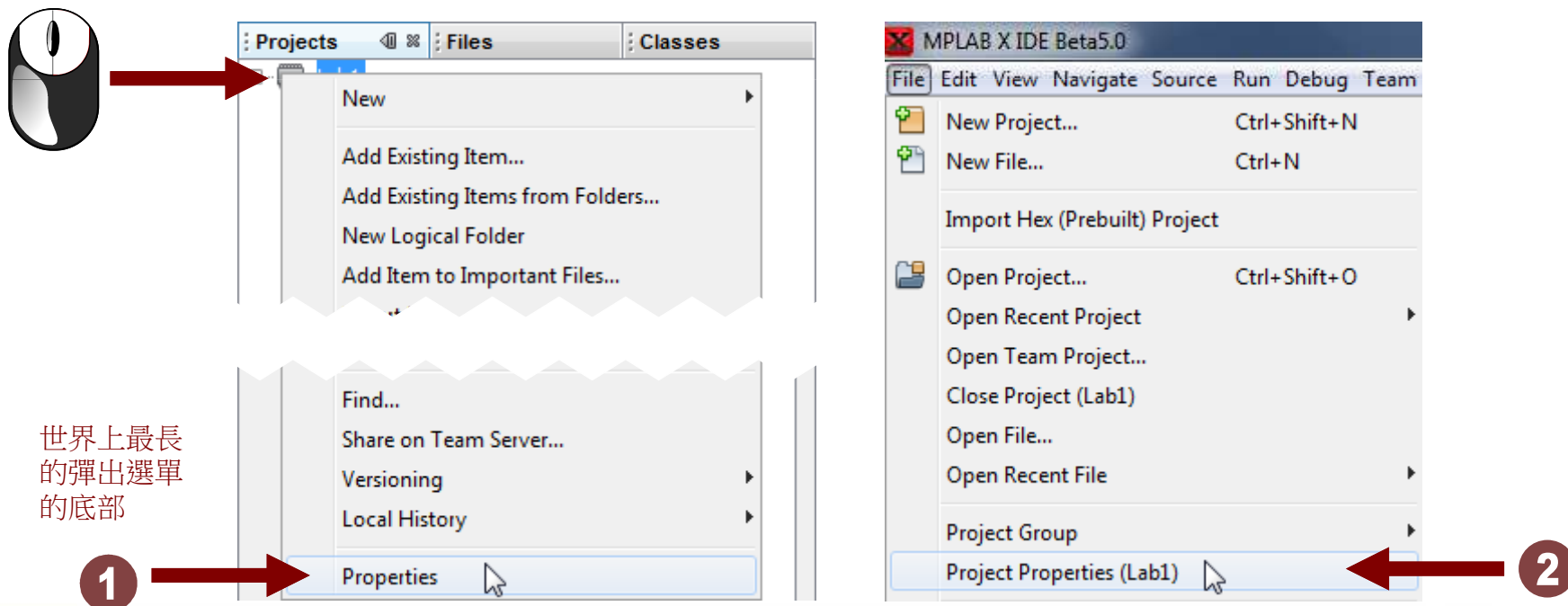
ONE PIC[®] MCU PLATFORM



如何使用專案

如何更改專案的內容

- 1 按右鍵專案樹中專案的頂部結點，然後從彈出功能表中選擇 **Properties**（內容） – 或者 –
- 2 或在 **File ► Project Properties**（專案內容）



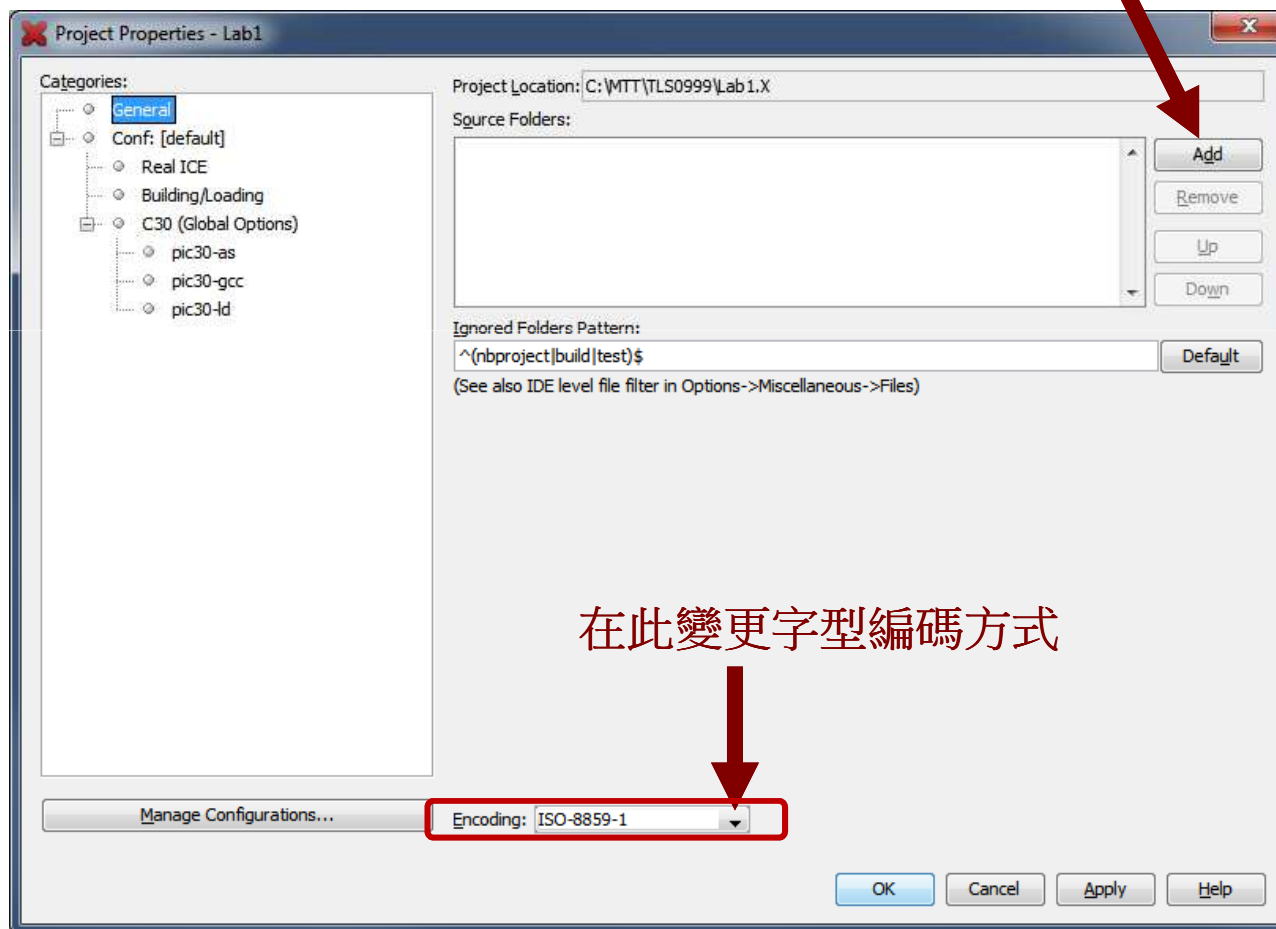
如何更改專案內容

General (一般)

在此加入除專案檔案
夾外的原始檔案夾

File ► Project Properties 下的 General

若使用專案目錄外的原始
檔案，應在此加入這些檔
件所在目錄的路徑。



如何更改專案內容 元件、編譯工具和除錯器

File ► Project Properties 下的 Conf: [default]

適用於目前專案設定的選項包括：

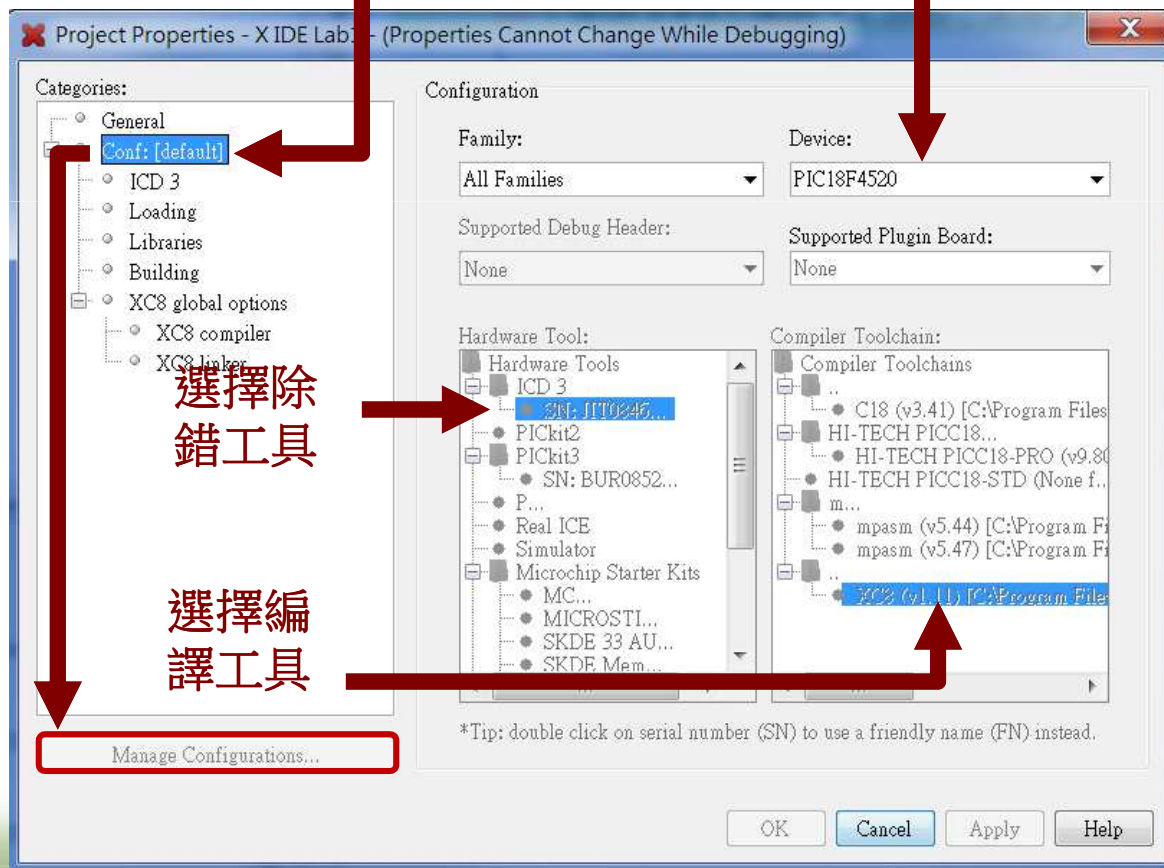
- **Device**（元件）
- **Hardware Tool**（硬體工具）
- **Compiler Toolchain**（編譯器工具鏈）

還可以建立其他設定，以在經常變更的選項之間快速切換



設置選定設定的選項

選擇元件



選擇除錯工具

選擇編譯工具

如何更改專案內容

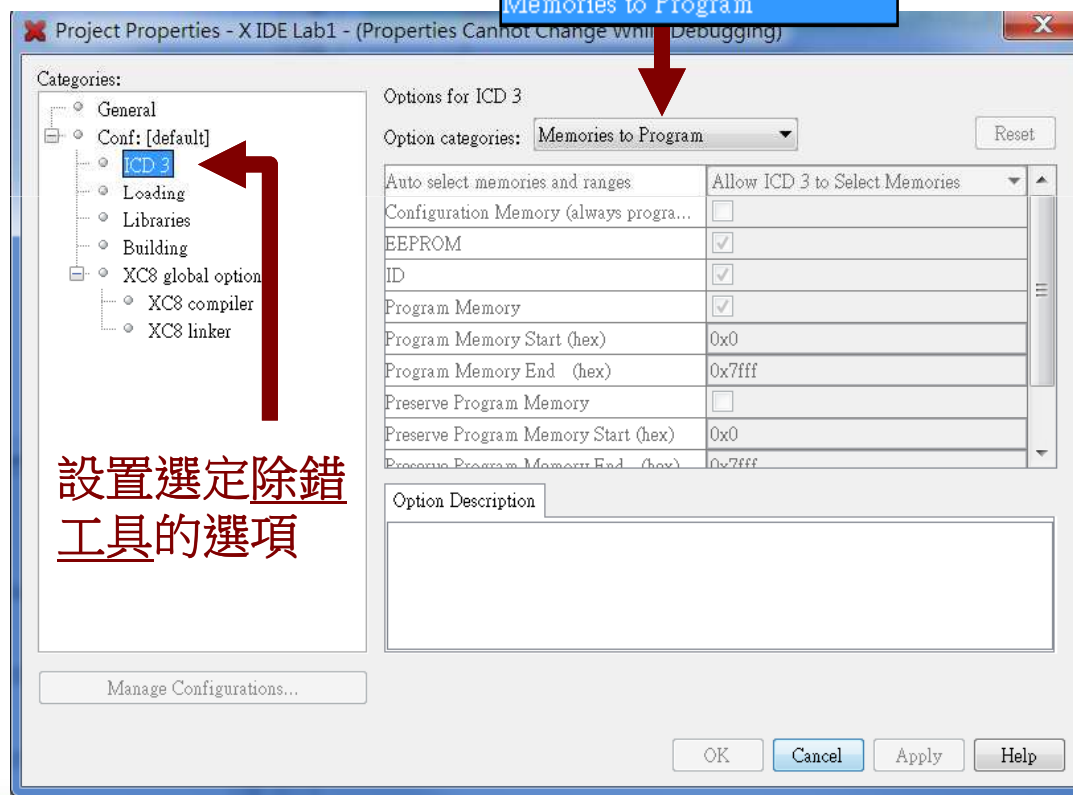
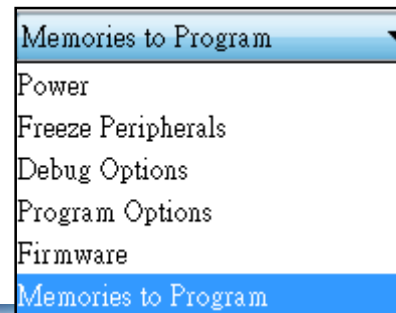
除錯工具屬性

File ► Project Properties
Conf: [default]
► ICD 3

除錯和程式燒錄選項:

- 燒錄範圍
- 燒錄電源設定
- 凍結周邊
- 除錯選項
- 燒錄選項
- 韌體

Categories
(與 MPLAB® 8 相同)



設置選定除錯
工具的選項

如何更改專案內容

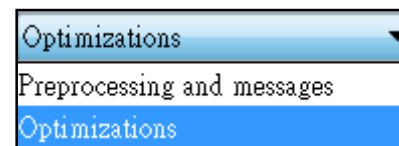
XC8 編譯器內容

File ► Project Properties
XC8 compiler

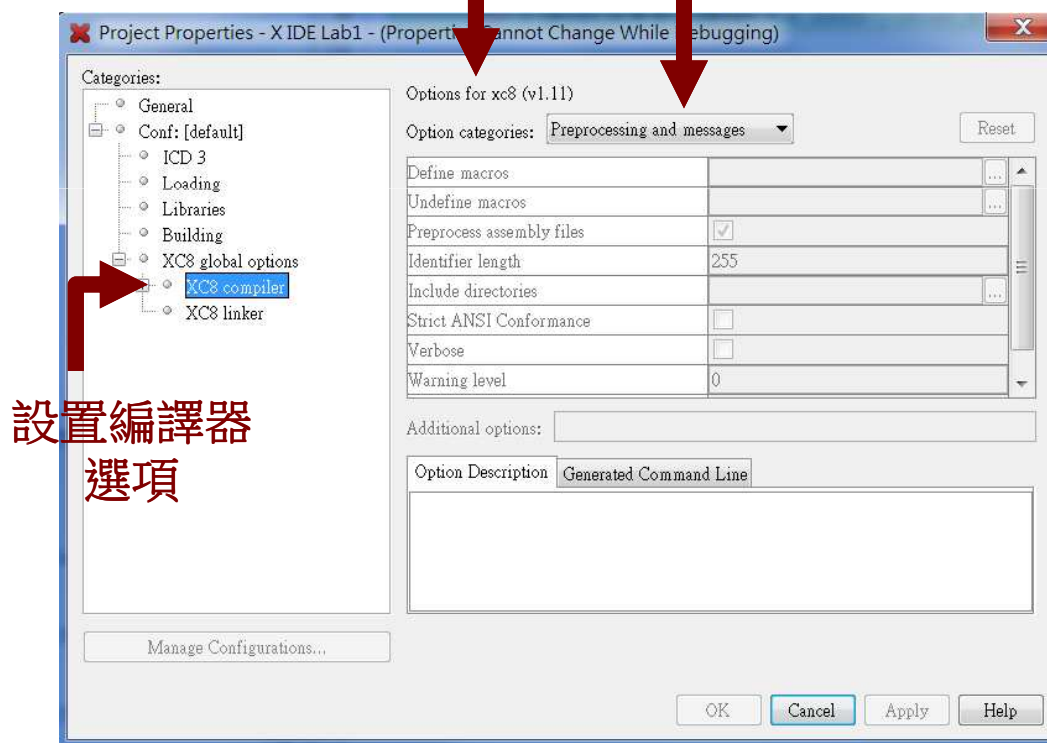
編譯器選項包括：

- 前置處理與訊息
- 最佳化選項

Categories
(與MPLAB® 8相同)



版本



設置編譯器
選項

如何更改專案內容

XC8 連結器內容

File ► Project Properties
XC8 linker

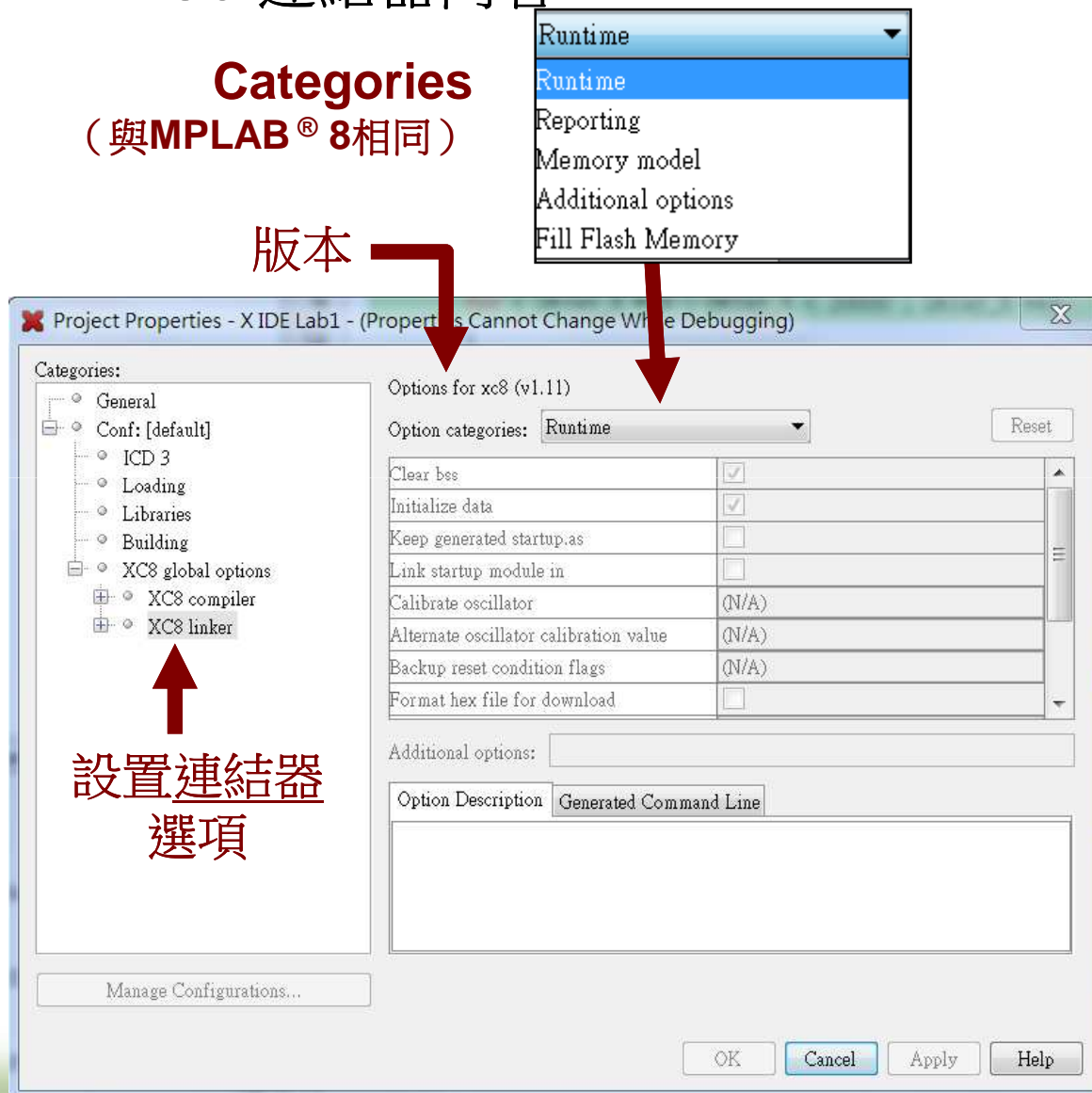
連結器選項包括：

- 執行時間
- 回報
- 記憶體模式
- 加入其它選項
- 填入程式記憶體

Categories
(與MPLAB® 8相同)

版本

設置連結器
選項



如何查看程式的統計資料

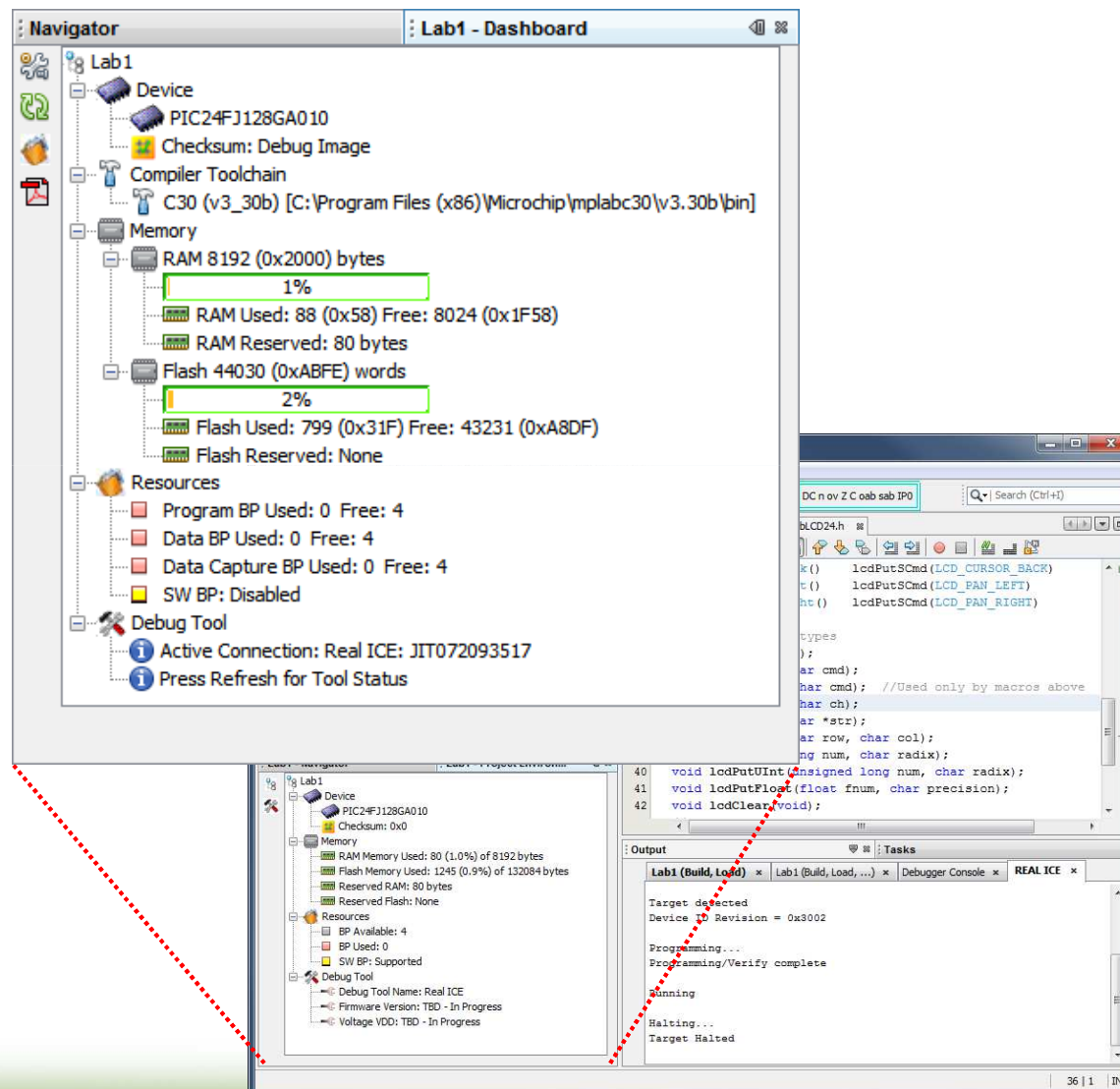
Dashboard 視窗

- 1 從功能表中選擇
Window ► Dashboard
- 2 在專案樹中選擇專案
- 3 出現 **Dashboard** 選項卡

- 元件
- 編譯器
- 記憶體使用狀況
- 可用資源
- 除錯工具

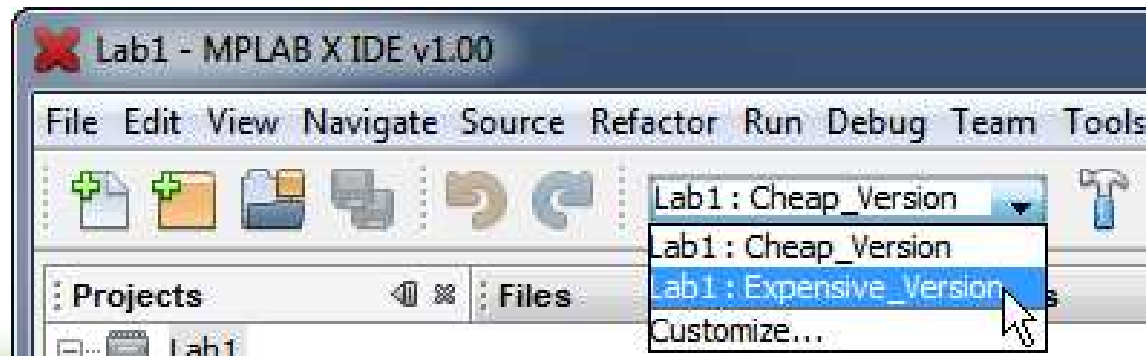


打開
project properties
(專案內容) 視窗



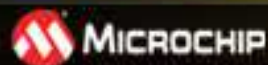
專案下的劇本設定

- 一個專案可以為同一程式設置多種編譯劇本：
 - 不同目標元件
 - 不同除錯工具
 - 不同編譯器設置
 - **project settings**（專案設定）視窗中的任何項均可不相同
- 可從工具列下拉式顯示方塊中選擇所需的劇本來編譯（實驗二）：



Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC MCU PLATFORM



實驗二

如何建立專案下的編譯劇本

實現條件式編譯功能

為何要有不同的編譯劇本

- 暫時變更 **MCU**元件的選擇
- **C compiler** 的選擇使用 (**XC8 or C18**)
- 開發工具的選用 (**ICD3 or PICKit3 ...**)
- 使用不同的連結描述檔 **LKR** 及 **GLD** 做記憶體變更設定
- 軟體版本的控制
- 軟體因客戶不同需求的修改
- 除錯模式與一般模式的選擇
- 編譯過程中的控制

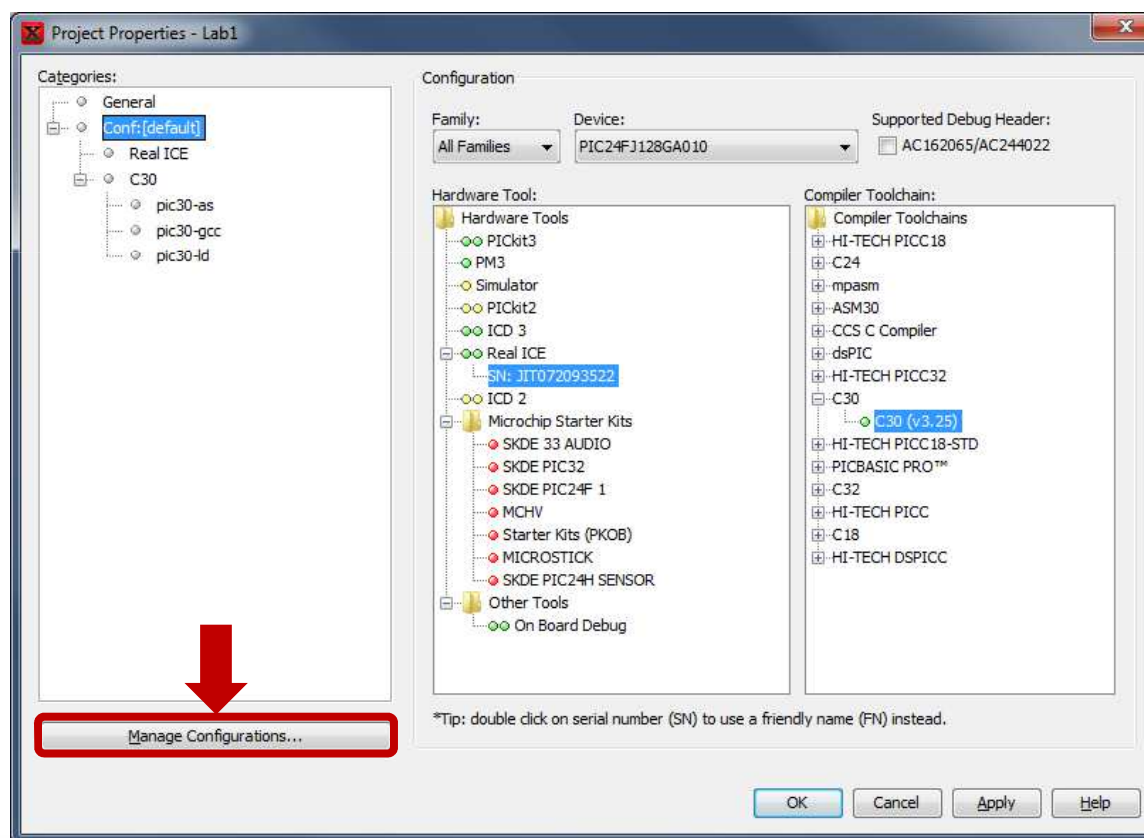
如何建立編譯劇本

1 專案下的編譯劇本管理

開啟開實驗二 (X IDE Lab2)

在 **Project Properties** 視窗中，按一下 **Manage Configurations...** (管理編譯劇本...) 選項。

在這裡 **Configurations** 把它翻譯成劇本，也就是說在同一個專案下可以有好幾套的編譯劇本來做開發，可以讓使用者選用不同的元件、連結描述檔、除錯模式、條件式編譯...等需求。

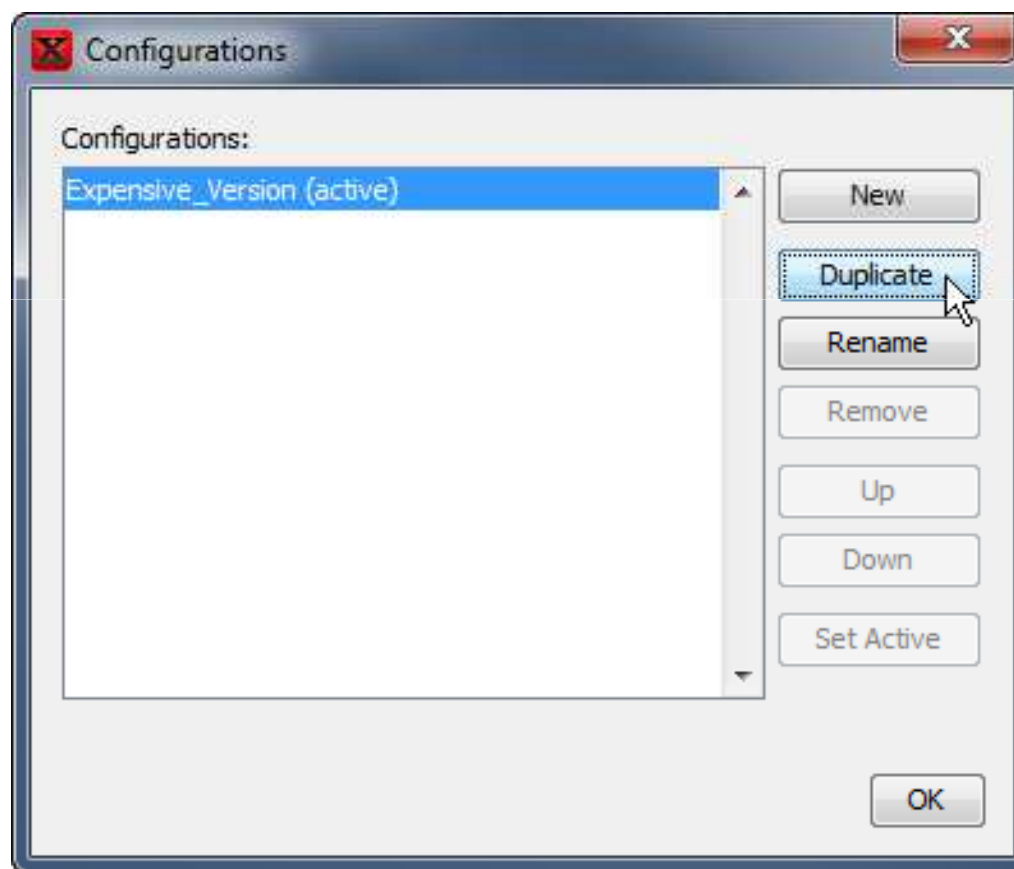


如何建立編譯劇本

2 新建或複製現有劇本

- 新建編譯劇本所需的步驟和建立一個新專案一樣多。
- 複製劇本方式將會建立一個所選定劇本的副本，您可根據需要在其中任意進行修改。

選擇 **Duplicate** 來複製一個新的劇本二



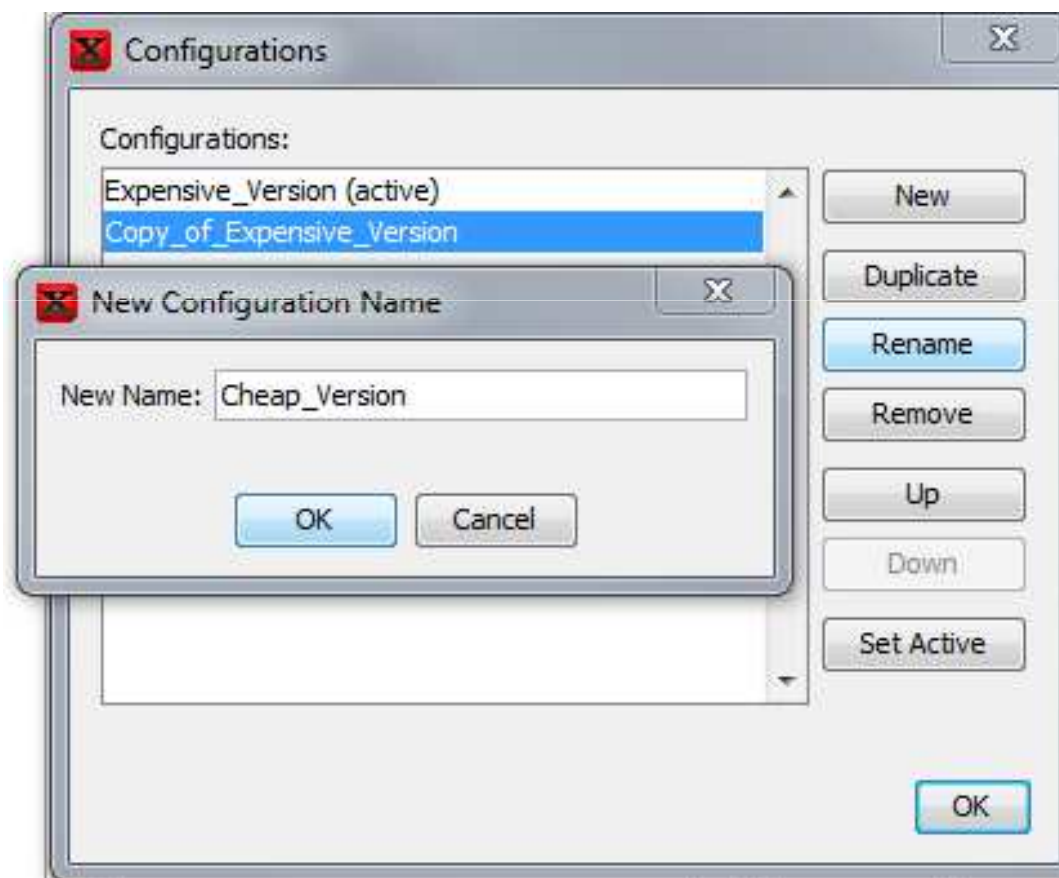
如何建立劇本

3 給設定賦一個有意義的名稱

設定新劇本名稱。按一下
Rename（重命名）按鈕分
別輸入兩種劇本名稱：

1. Expensive_Version
2. Cheap_Version

在 **Configurations** 視窗中
按一下 **OK**，即可在
Project Properties 視窗中
加入新劇本。



如何建立劇本

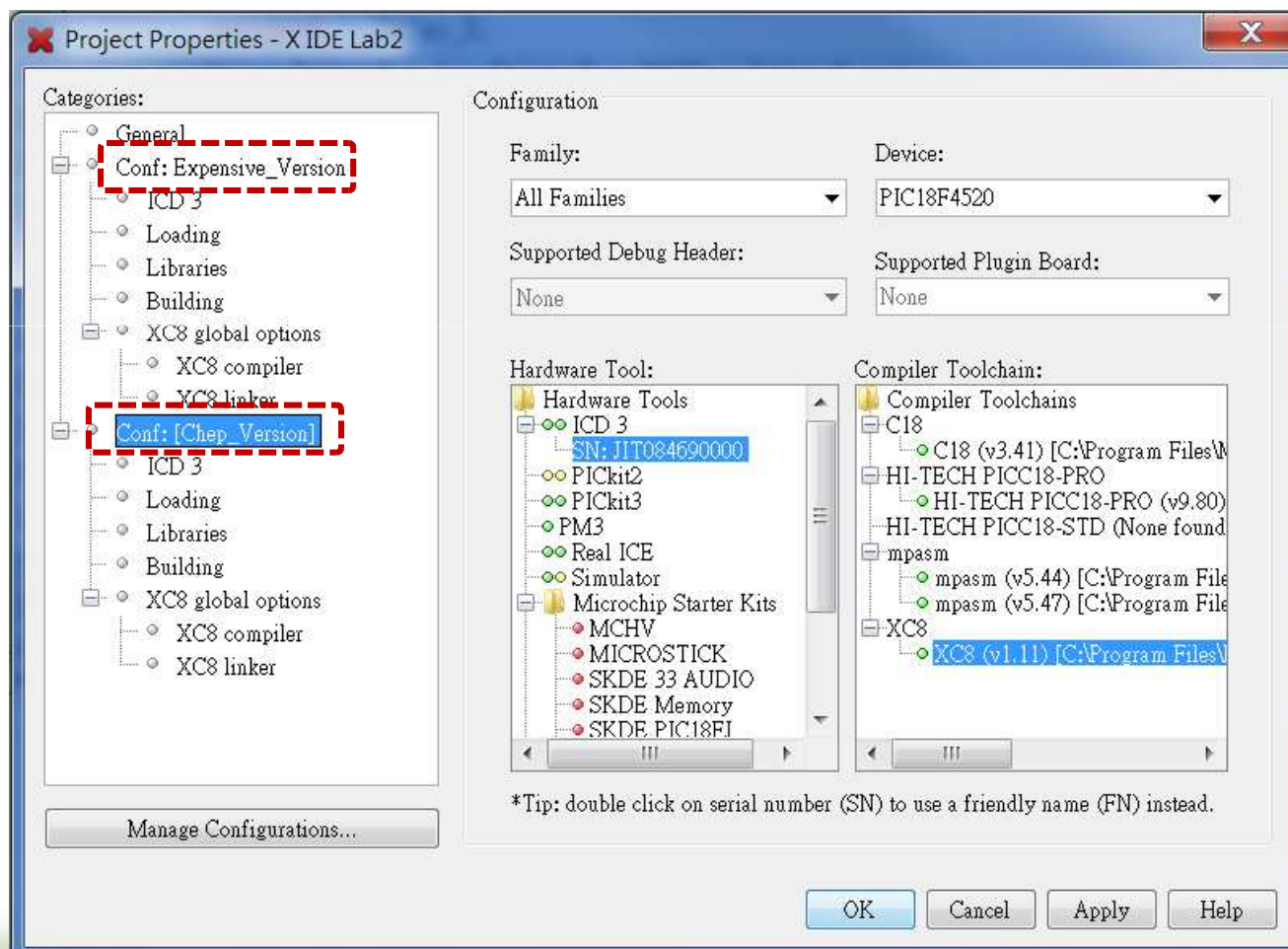
4 更改新劇本以滿足您的需求

每個劇本的設定所包含的選項項目是一樣的。您可以單獨設置各個選項裡的設定。

右圖有兩個劇本：

- Expensive_Version
- Cheap_Version

可供選擇來編譯，目前兩個劇本編譯的條件都一樣，必須要在 **XC8 Global options** 加入編譯的巨集指令或連結用的檔案等控制因素



如何建立劇本

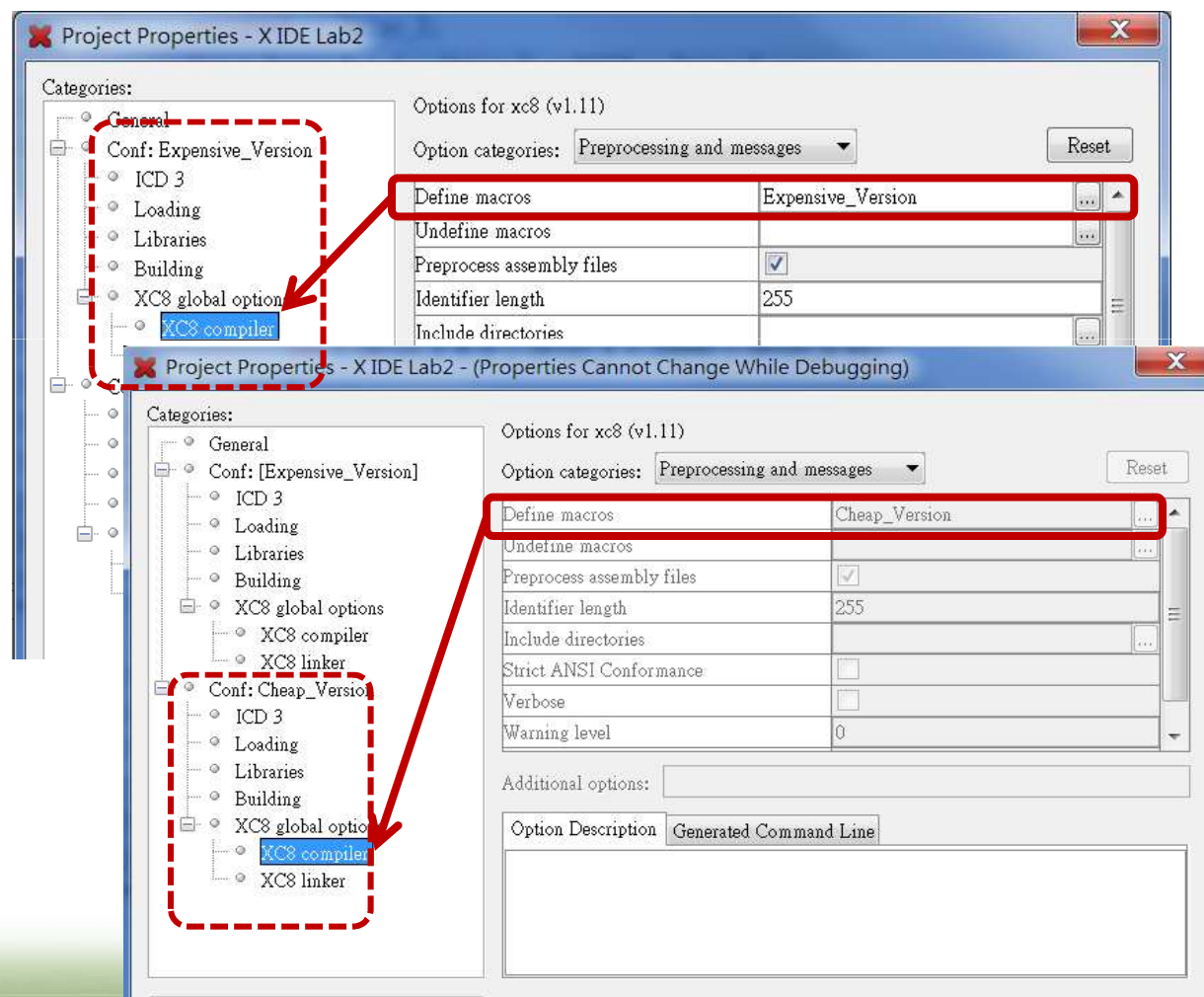
5 建立預處理巨集來識別設定

在劇本一點選 XC8 compiler
後會出現對話視窗，在
Define macros 輸入：
Expensive_Version

在劇本二點選 XC8 compiler
一樣在 Define macros 輸入：
Cheap_Version

也可以點選 XC8 linker 輸入
連結控制方式

您也可以通過在組譯器設定中加入預
處理巨集，在組合語言中實現完全相
同的功能。



如何建立設定

6 在程式中使用預處理巨集

用剛剛定義的預處理巨集
程式裡建立特定的設定。

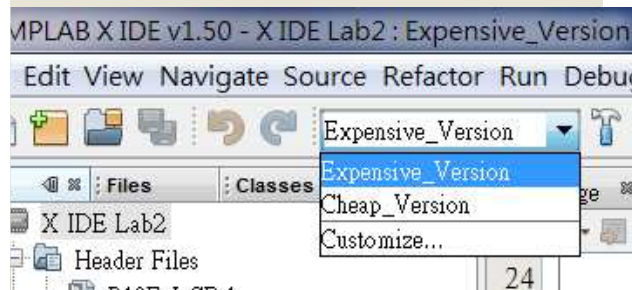
式裡: 透過 **#ifdef** /
#endif 的虛擬指令使用預
理巨集，做為程式內的條
式的編譯設定。

uilt 之前，使用工具列中的
拉式選擇視窗，在不同劇
間選擇不同的劇本來編譯。

在你的程式中需加入此巨集的判斷

```
#ifdef Expensive_Version
// 建立條件式編譯劇本一 (Expensive Version)
putsLCD((const far char*)"I am Expensive..");
#endif
```

```
#ifdef Cheap_Version
// 建立條件式編譯劇本二 (Chep Version)
putsLCD((const far char*)"I am Chep.....");
#endif
```



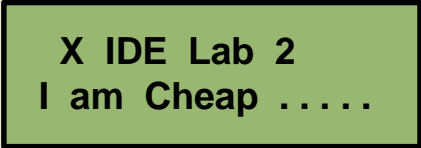
實驗二

如何建立專案的劇本

結果

選擇 “Cheap_Version”，編譯並執行.....

LCD顯示幕：



X IDE Lab 2
I am Cheap

選擇 “Expensive_Version”，編譯並執行.....

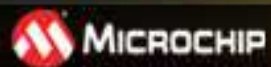
LCD顯示幕：



X IDE Lab 2
I am Expensive ...

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



如何除錯應用程式

如何啟動除錯程序



Debug Project (除錯專案)

啟動除錯程序。

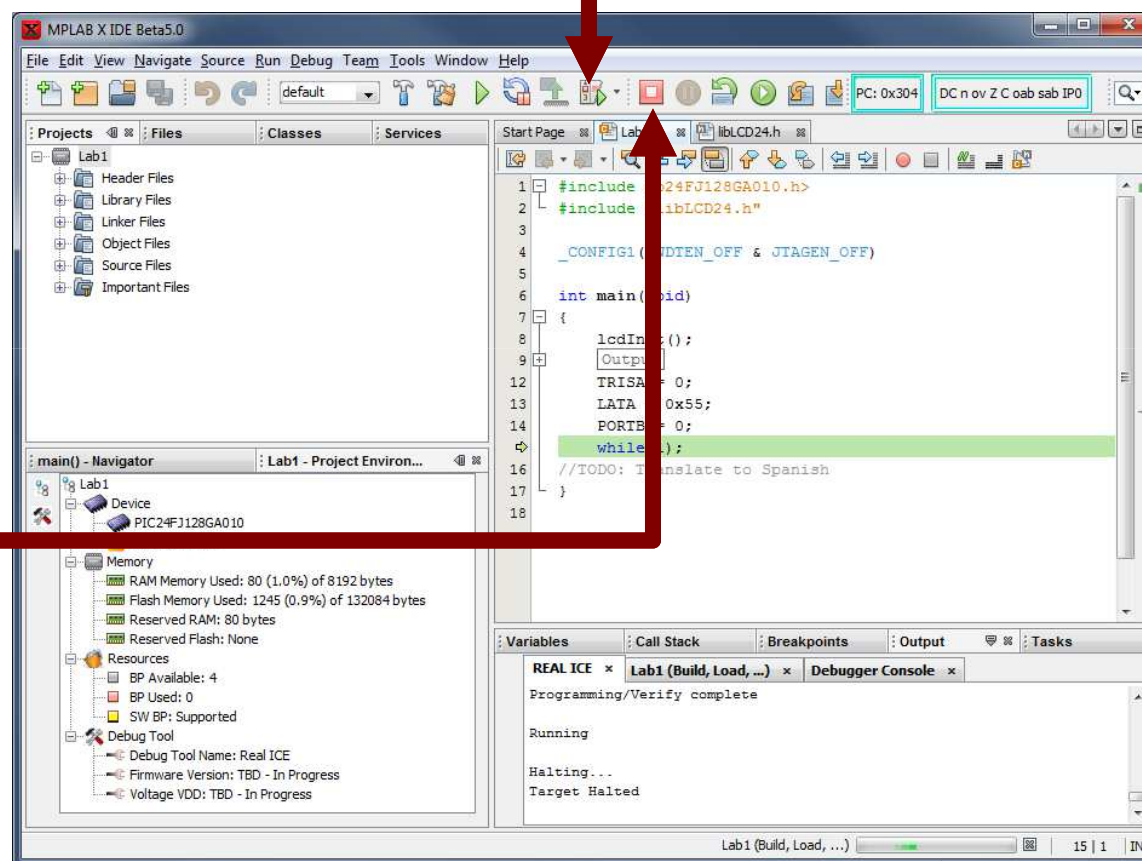
1. 在除錯模式下編譯專案。
2. 連接有硬體除錯器，該操作將建立USB連接
3. 燒錄目標元件
4. 並執行程式。



End DebugSession (中止除錯程序)















中止除錯對話。

1. 斷開硬體除錯器
2. 關閉 USB 的連接。
3. 該操作必須在修改程式並繼續進行除錯前下達中止除錯程序。



對話啟動時，
進度條會有顯示

如何使用除錯工具列


功能	MPLAB® 8	MPLAB® X
中止除錯步驟	無	
中止/暫停		
執行/繼續		
連續單步執行		無
執行程式至游標位置處	在上下文選單中	
單步執行		
單步跳過		
重置		
聚焦游標到 PC	無	

除錯斷點

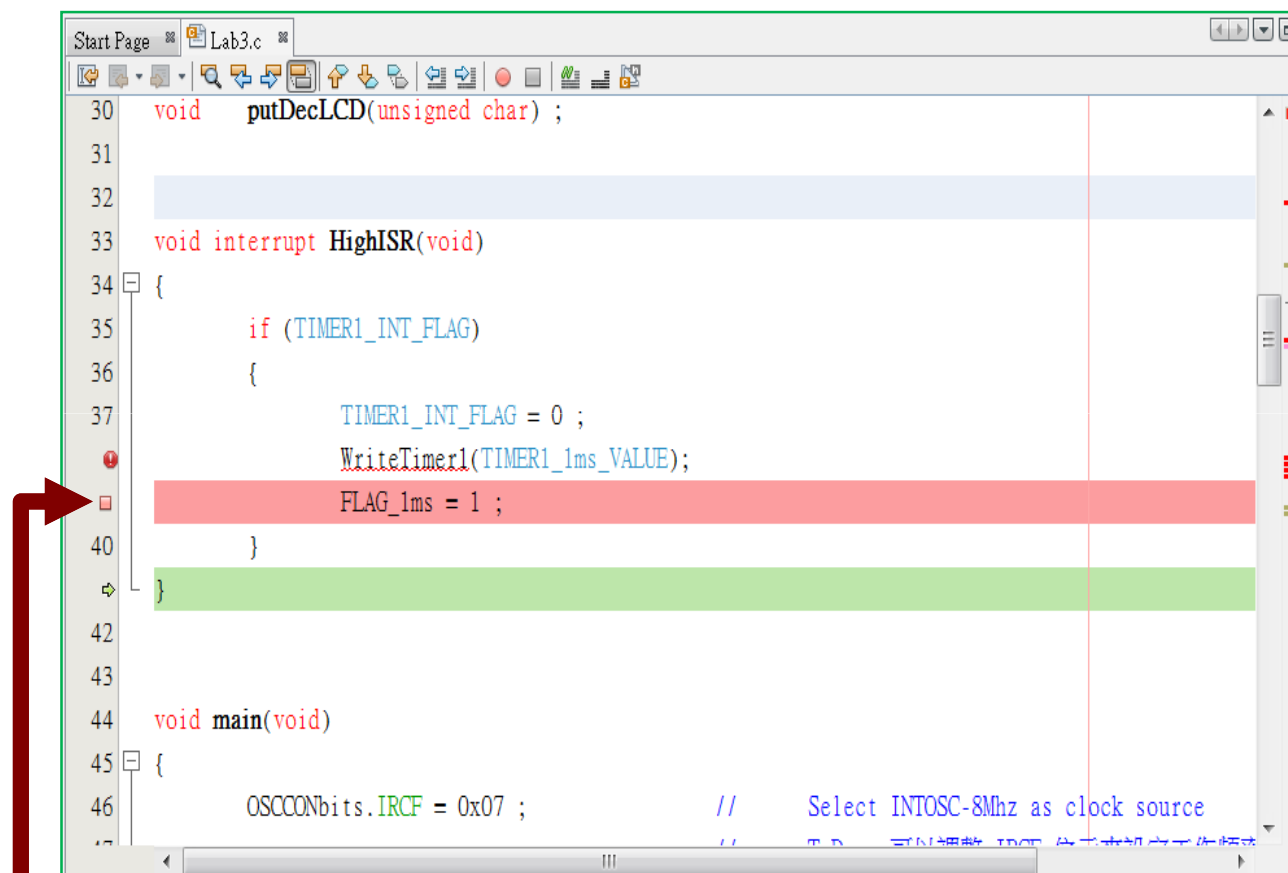
如何設置或清除斷點

斷點設定

一下圖符頁邊上的行號，設置（當前無斷點）和余（當前有斷點）此行勾斷點。

固紅色正方形（）示斷點已設置。對應的式行也將以紅色背景高顯示。

色為目前 PC 暫停的位



```
30 void putDecLCD(unsigned char) ;
31
32
33 void interrupt HighISR(void)
34 {
35     if (TIMER1_INT_FLAG)
36     {
37         TIMER1_INT_FLAG = 0 ;
38         WriteTimer1(TIMER1_1ms_VALUE);
39         FLAG_1ms = 1 ;
40     }
41 }
42
43
44 void main(void)
45 {
46     OSCCONbits.IRCF = 0x07 ; // Select INTOSC-8Mhz as clock source
47     // M.D. 可以調整 IRCF 位元來設定工作頻率
```

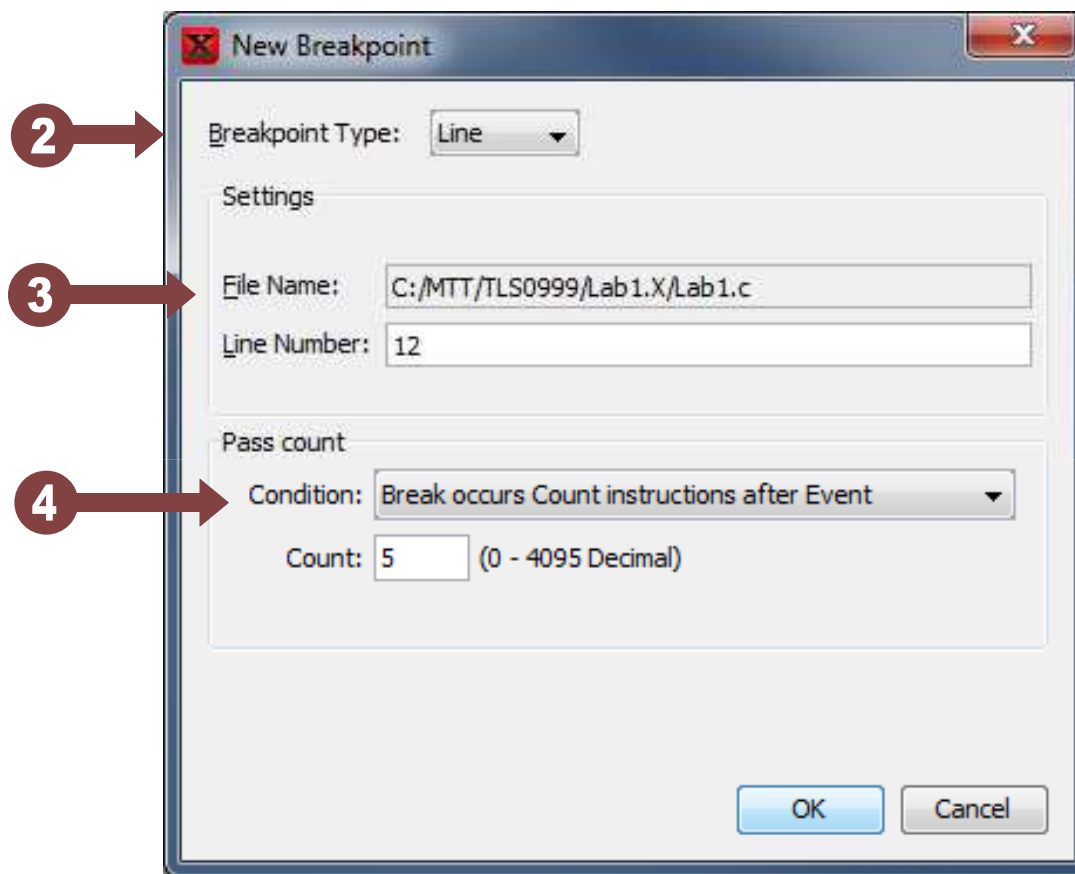
在行號上按一下以切換斷點
有 / 無 的交互設定

斷點

如何設置條件式的斷點

條件式斷點的設定

- 1 從功能表中選擇 **Debug** (除錯) ▶ **New Breakpoint** (新斷點)
- 2 選擇 **Line** (行) 作為斷點類型
- 3 指定數據 **File** (文件) 和 **Line Number** (行號) 設置
(根據游標位置自動填充)
- 4 指定 **Condition** (條件) 及 **Passcount** (發生次數計數)



此項功能一所選用的元件而有所不同。
PIC18F4520 不具有此功能

中斷點

如何設置條件性資料中斷點

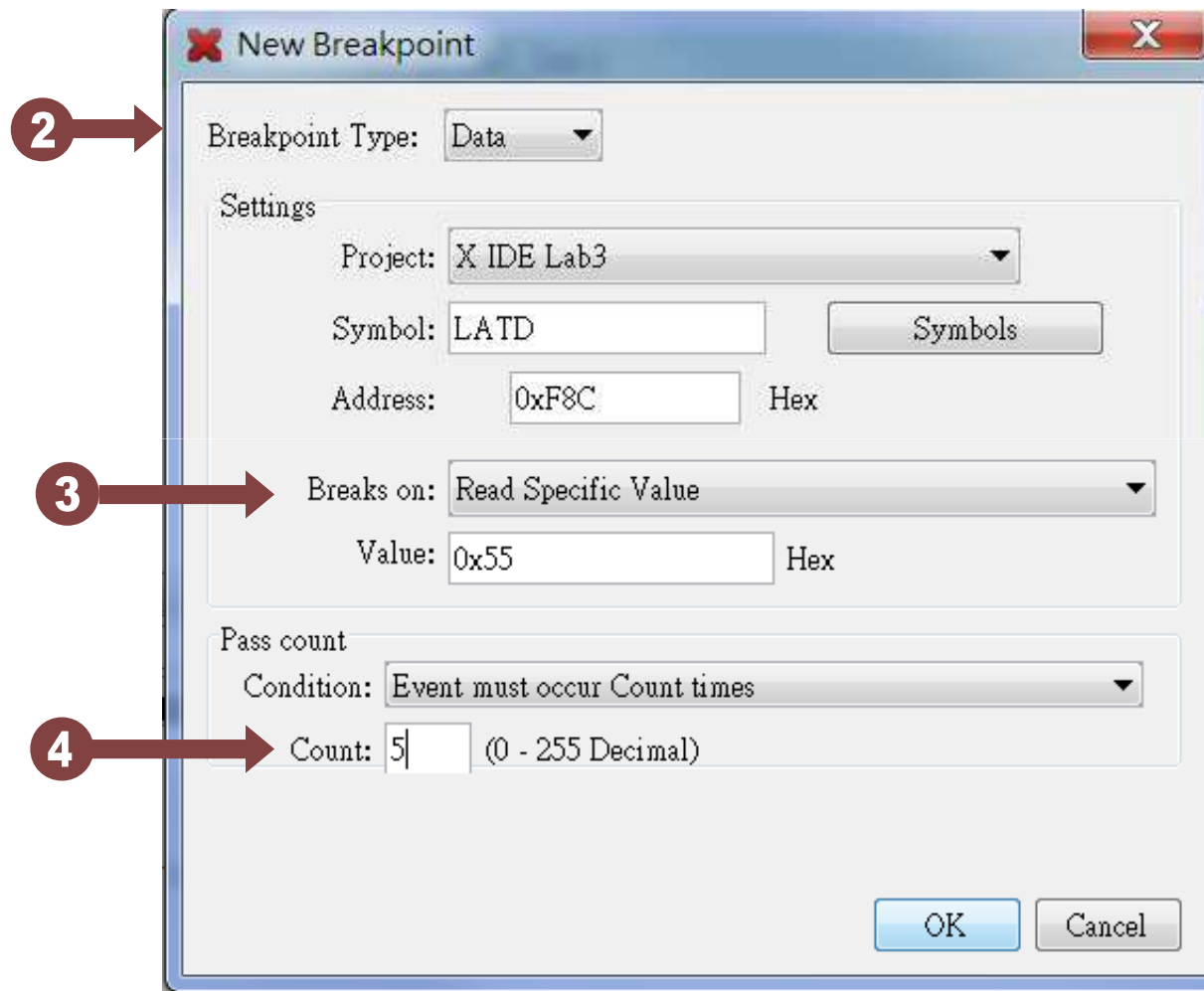
條件性 數據中斷點

從功能表中選擇
Debug►New Breakpoint

選擇**Data**（數據）作為中
斷點類型

指定數據**Address**（地址）
和**Breaks on**（中斷條件）
設置

指定**Pass count**
Condition



斷點

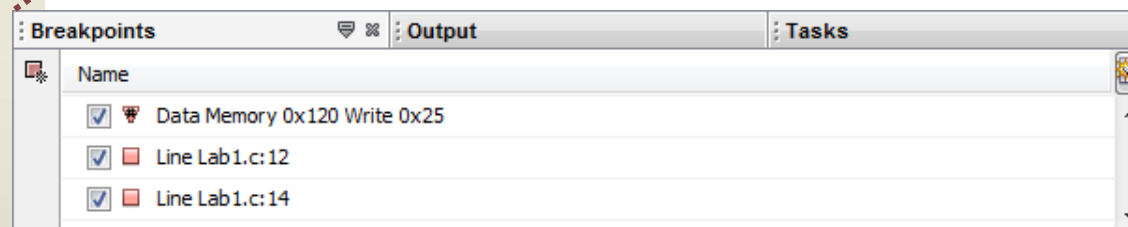
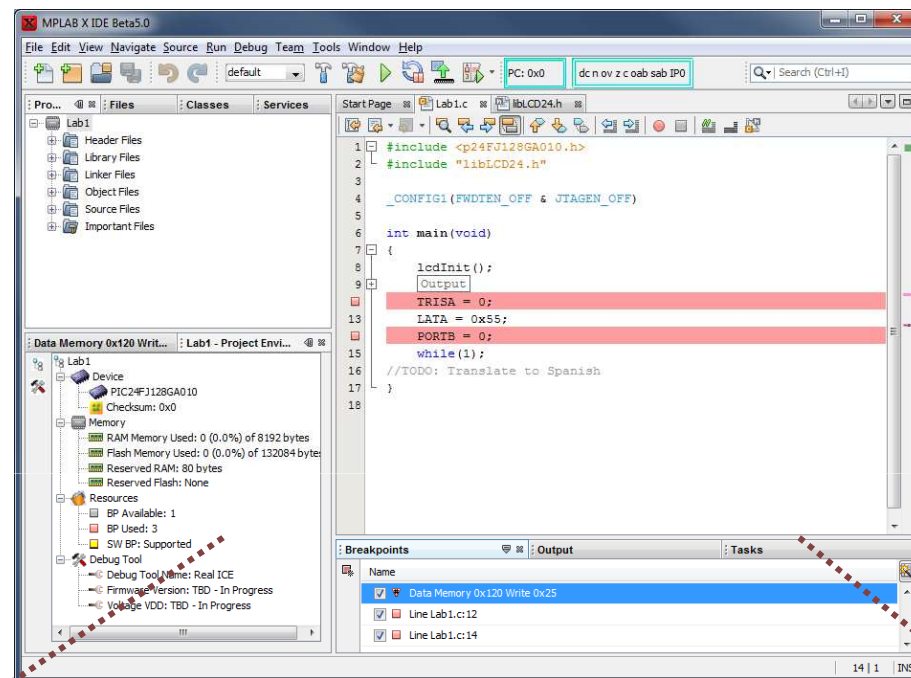
如何查看專案中的所有斷點

顯示Breakpoint (斷點) 視窗

1 選擇 **Windows**
► **Debugging** ► **Breakpoints**

2 按右鍵以：

- 建立新斷點
- 使能所有斷點
- 禁止所有斷點
- 刪除所有斷點
- 組合斷點（允許按組使能/禁止）
- 更改斷點設置

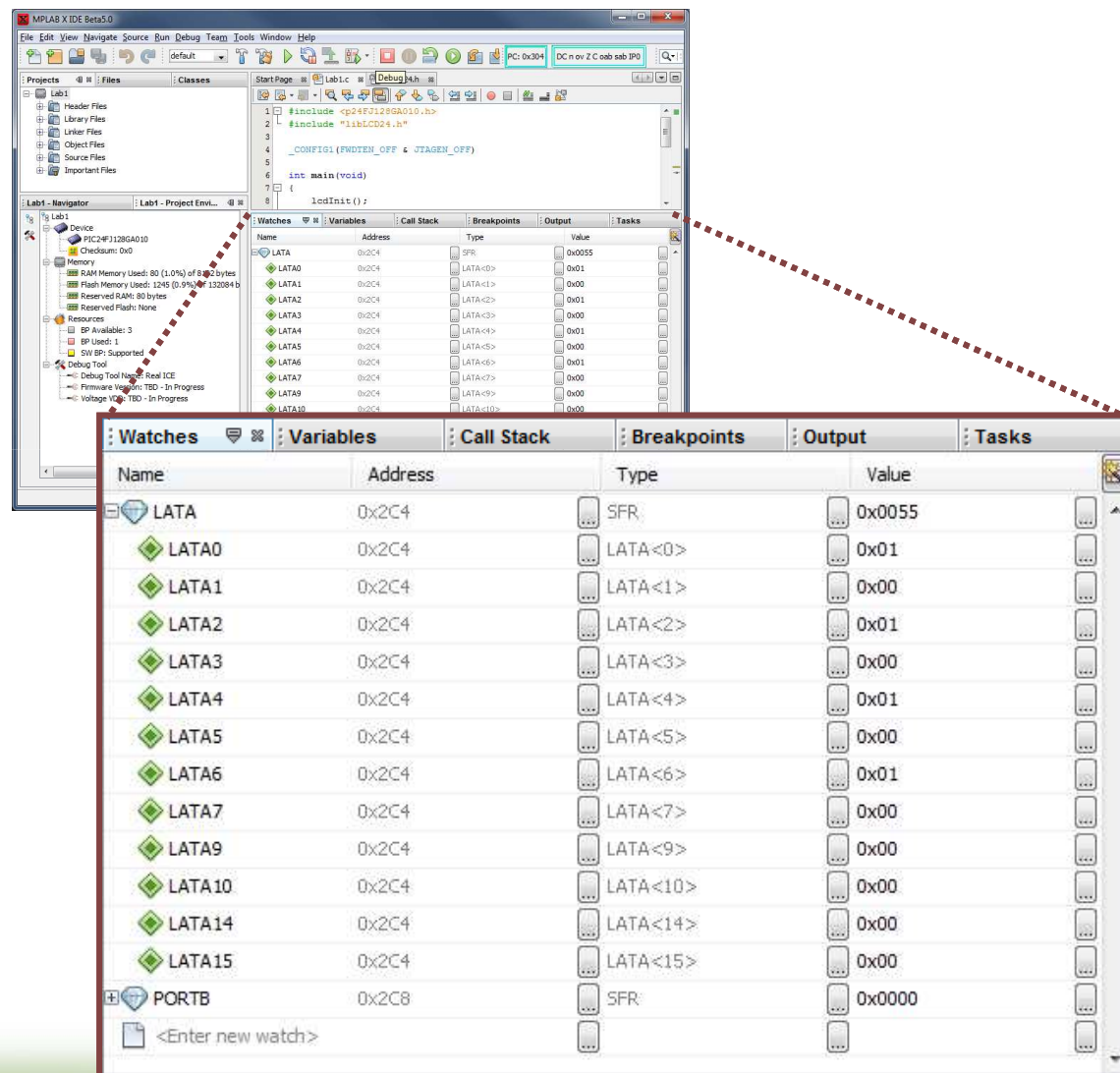
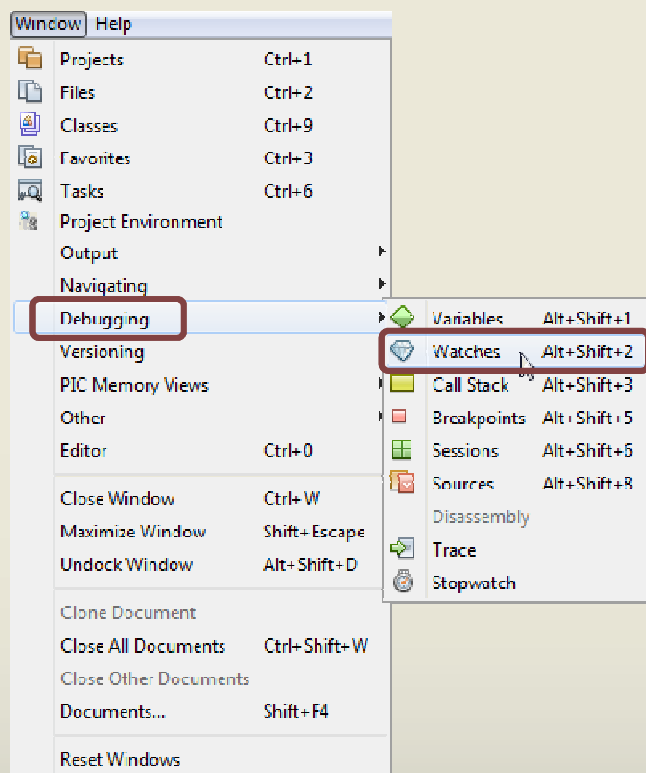


按兩下斷點可轉至原始檔案

觀察變數

如何顯示 Watches (觀察) 選項卡

從主功能表中選擇：
Window ▶ Debugging ▶ Watches



觀察變數

如何向 **Watches** 選項卡中加入變數

1 (可選) 在編輯器中高亮選中變數

2 執行以下一種操作：

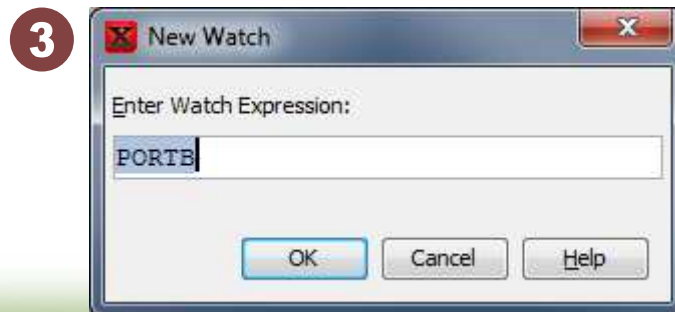
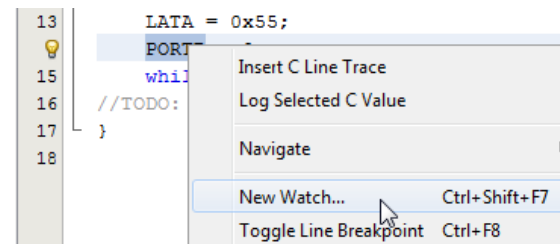
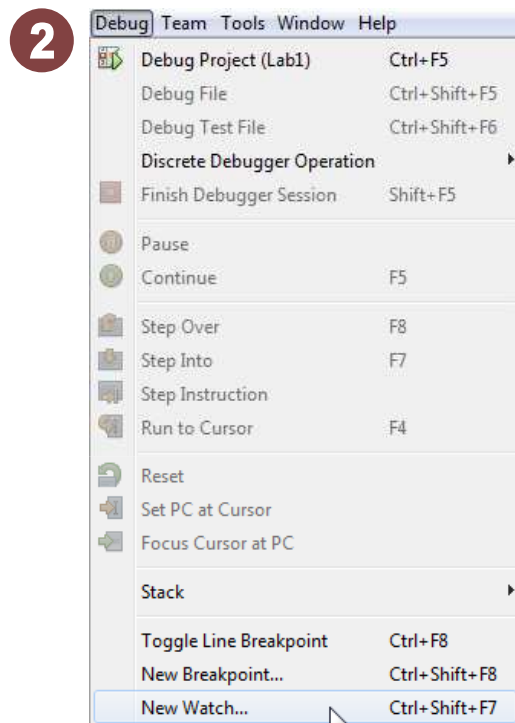
a 在編輯器中按右鍵並從彈出功能表中選擇 **New Watch...** (新建觀察變數.....)

b 從主功能表中選擇：
Debug ▶ New Watch...

c 在編輯器中高亮顯示一個變數，然後點擊並將它拖動到 **Watches** 視窗

d   

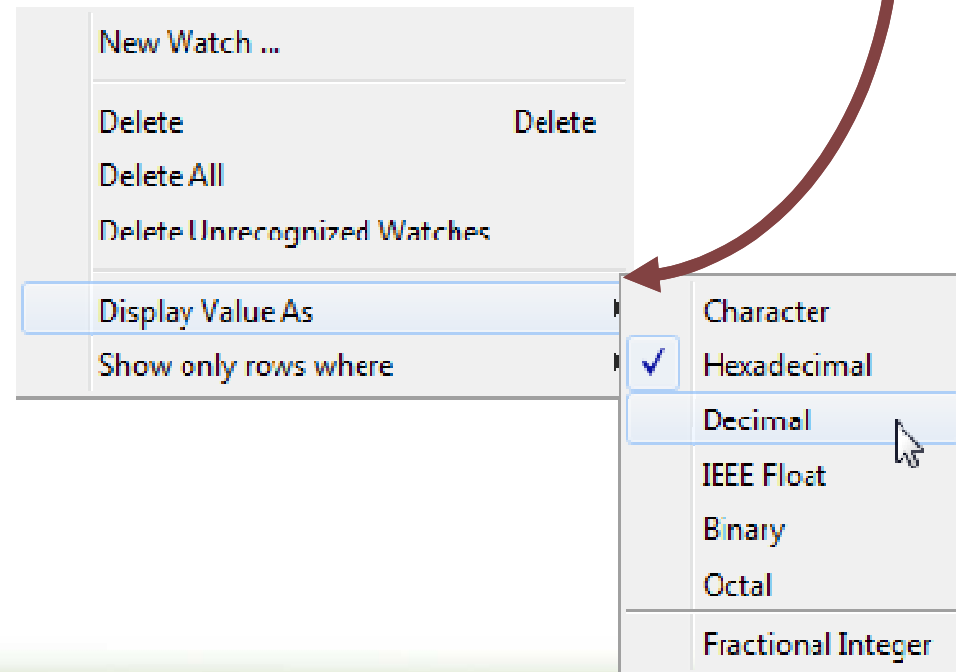
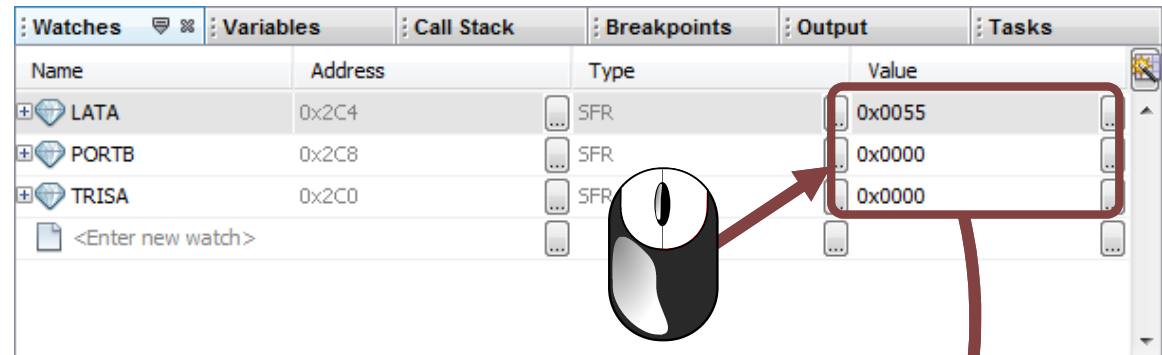
3 輸入變數名或僅接受所顯示的名稱並按一下 **OK**



觀察變數

如何更改觀察變量的基數

- 1 按右鍵 **Value** (值) 列中的一個值
- 2 從彈出功能表中選擇：
Display Value As (值的顯示形式)
- 3 選擇所需的基數/格式：
 - **Character** (字元)
 - **Hexadecimal** (十六進位)
 - **Decimal** (十進位)
 - **IEEE Float** (IEEE 浮點)
 - **Binary** (二進位)
 - **Octal** (八進制)
 - **Fractional Integer** (小數整數)



觀察變數

如何更改觀察變數的值

- 1 按兩下**Value**列中的一個值
- 2 輸入新值，完成後按**Enter**鍵



無法動態更改值。要進行更改，您必須首先暫停程式執行。

Watches	Variables	Call Stack	Breakpoints	Output	Tasks
Name	Address	Type	Value		
+ LATA	0x2C4	SFR	0x0055		
+ PORTB	0x2C8	SFR	0x0000		
+ TRISA	0x2C0	SFR	0x0000		
<Enter new watch>					

觀察變數

如何對觀察變量進行排序

1 Default
欄位按數位順序排列)

Address	Name	Type
0x2C4	LATA	SFR
0x2C4	LATA0	
0x2C4	LATA1	
0x2C4	LATA2	
0x2C4	LATA3	
0x2C4	LATA4	
0x2C4	LATA5	
0x2C4	LATA6	

2 第一次按一下：
昇冪（按字母順序排列）

Address	Name	Type
0x2C4	LATA	SFR
0x2C4	LATA0	
0x2C4	LATA1	
0x2C4	LATA10	
0x2C4	LATA14	
0x2C4	LATA15	
0x2C4	LATA2	
0x2C4	LATA3	

按一下下列標題.....

3 第二次按一下
降冪（按字母順序排列）

Address	Name	Type
0x2C4	LATA	SFR
0x2C4	LATA9	
0x2C4	LATA7	
0x2C4	LATA6	
0x2C4	LATA5	
0x2C4	LATA4	
0x2C4	LATA3	
0x2C4	LATA2	

4 第三次按一下：
回 Default

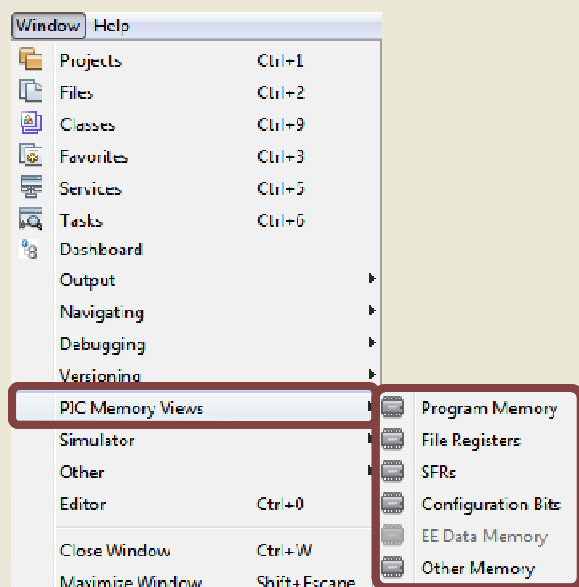
Address	Name	Type
0x2C4	LATA	SFR
0x2C4	LATA0	LAT
0x2C4	LATA1	LAT
0x2C4	LATA2	LAT
0x2C4	LATA3	LAT
0x2C4	LATA4	LAT
0x2C4	LATA5	LAT
0x2C4	LATA6	LAT

此顯示視窗以 **16-bit** 的元件為範例

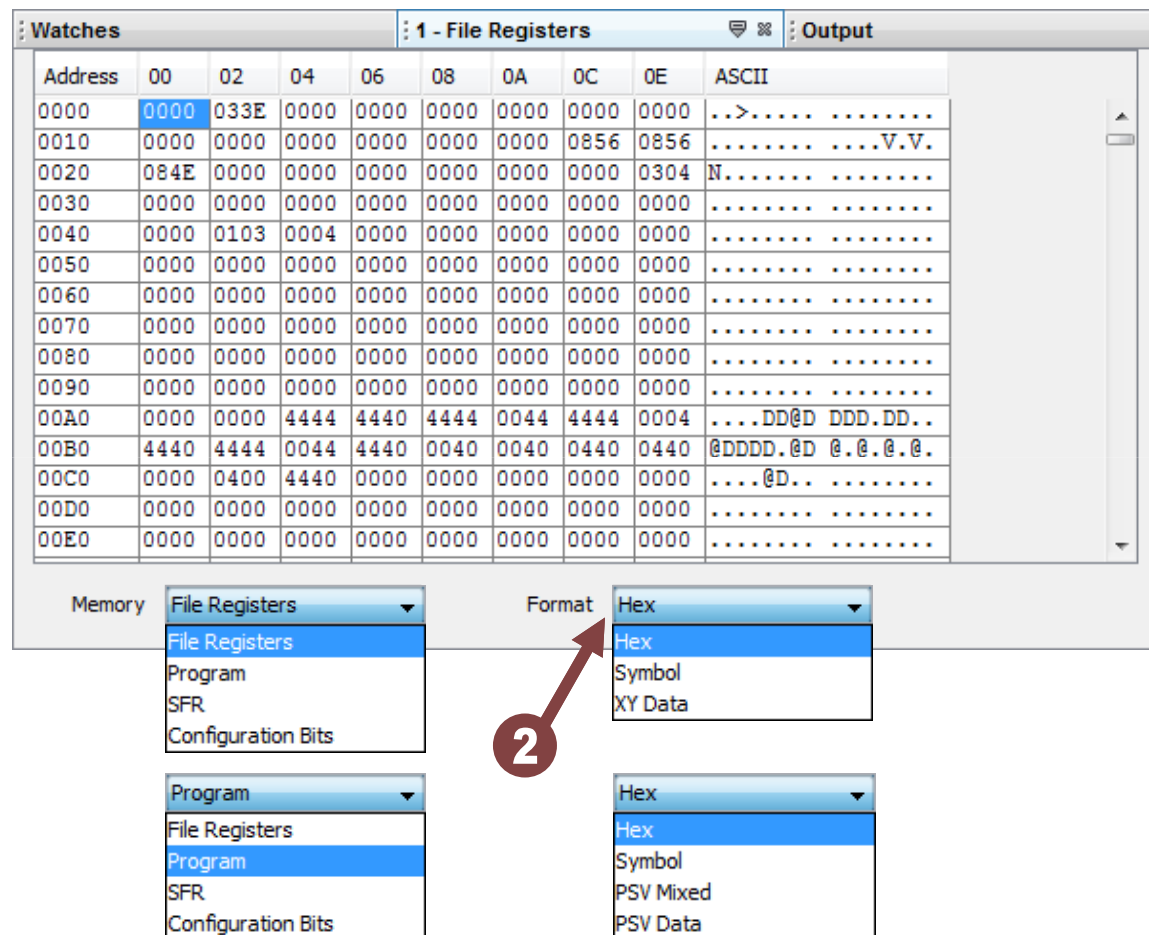
記憶體視窗

如何查看嵌入式記憶體

- 1** 從主功能表中選擇：
Window▶
PIC Memory Views▶



- 2** 選擇顯示格式



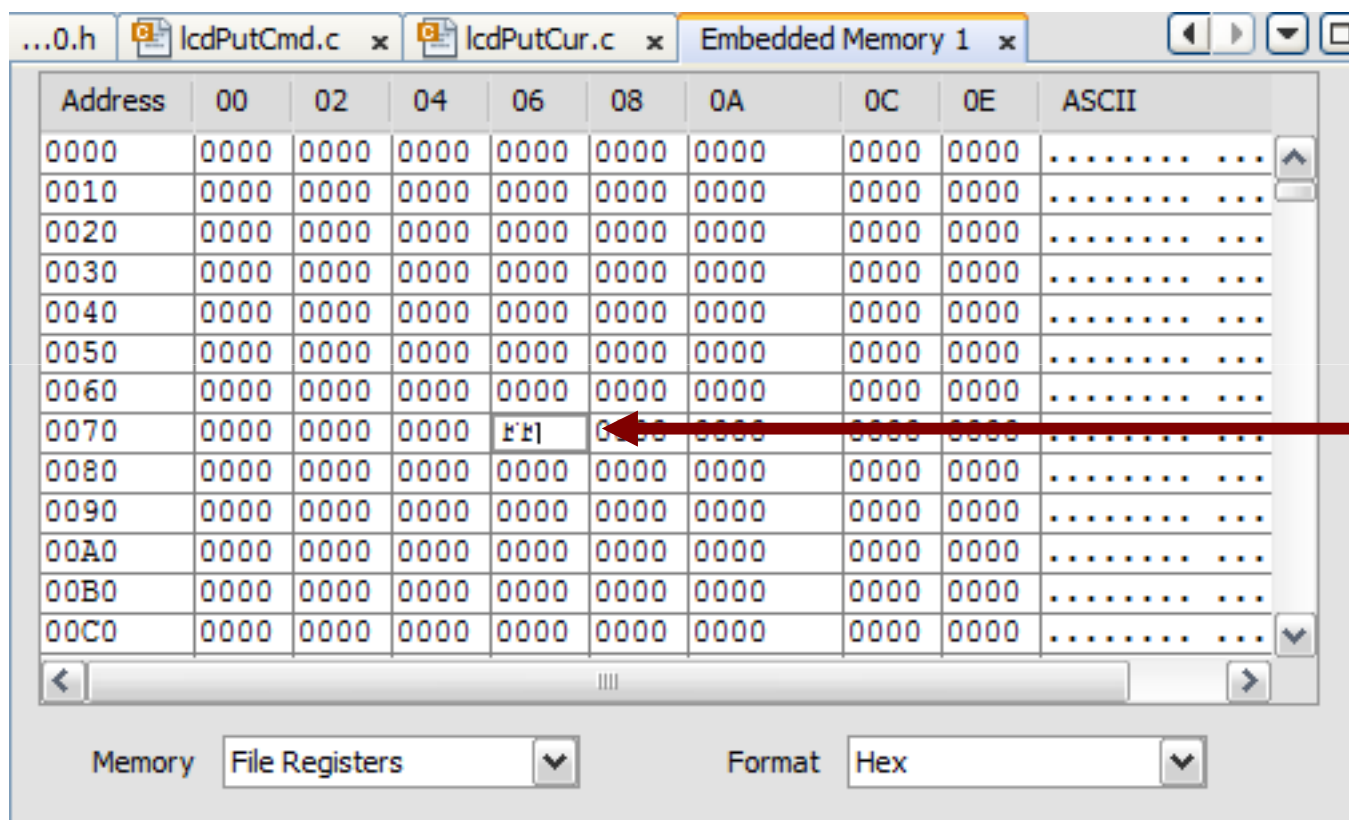
此顯示視窗以 **16-bit** 的元件為範例



可對每個記憶體視圖視窗進行設定，以顯示任何支援的記憶體類型。

記憶體視窗

如何更改儲存單元的值



按兩下儲存格
並輸入新值——
在完成時按
Enter鍵

此顯示視窗以 **16-bit** 的元件為範例

記憶體視窗

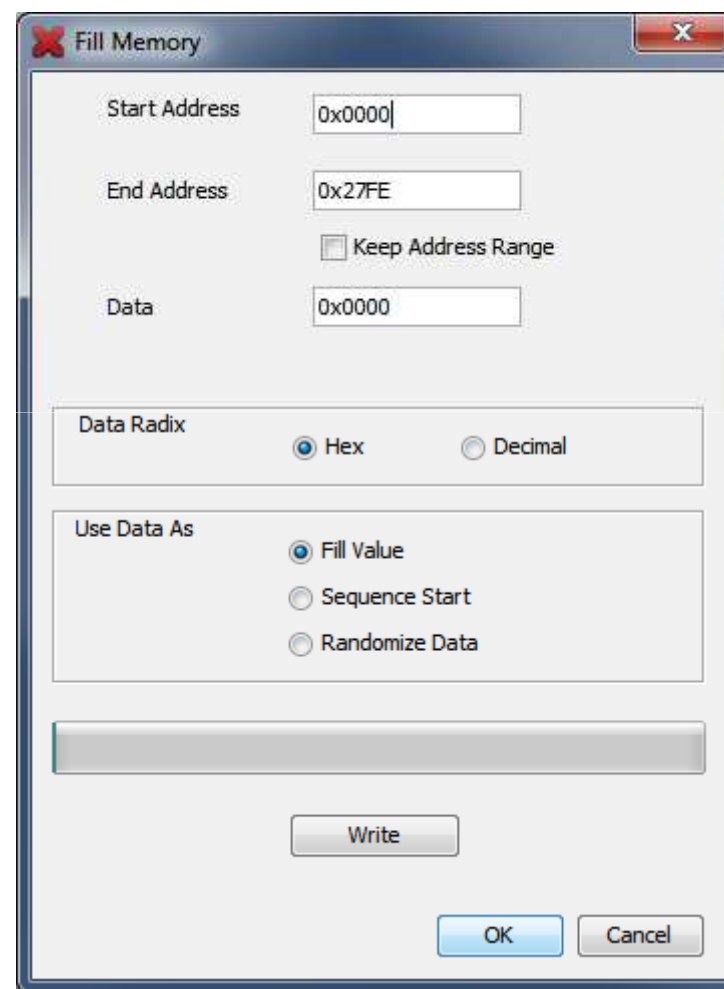
如何填充記憶體

- 在記憶體視窗中按一下右鍵，並從彈出選單中選擇：

Fill Memory...

(填入記憶體.....)

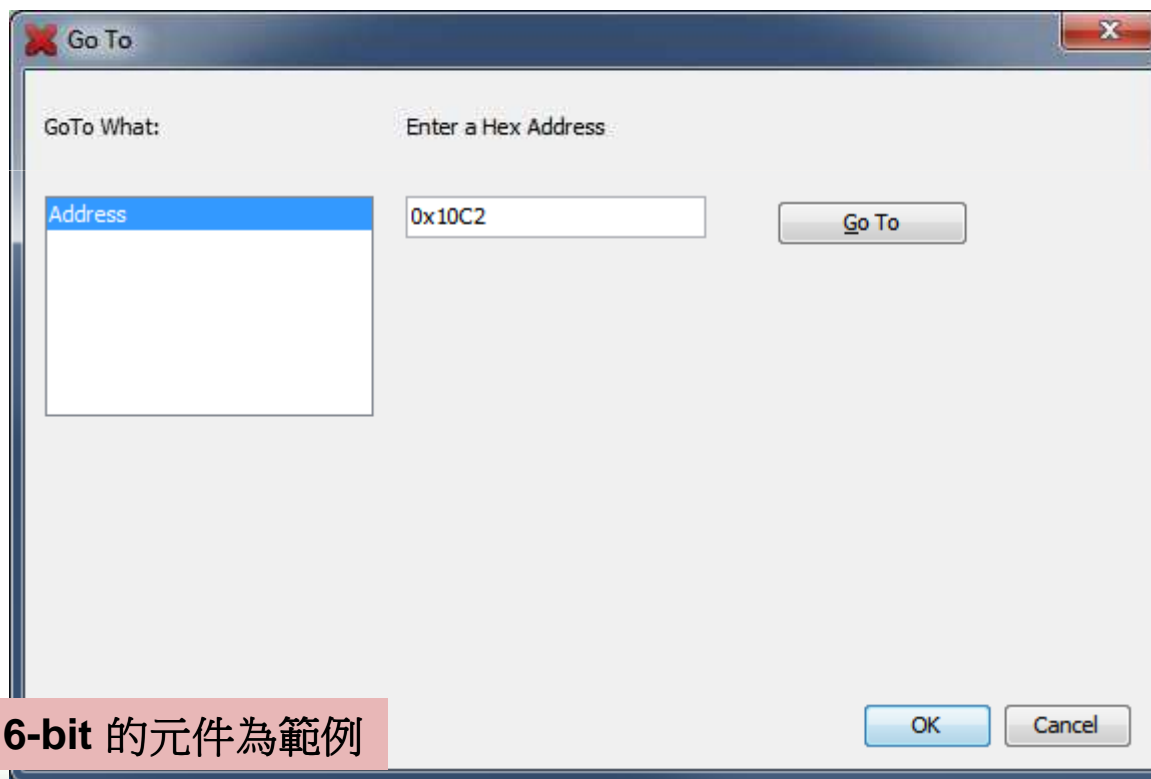
此顯示視窗以 **16-bit** 的元件為範例



記憶體視窗

如何快速跳轉至特定位址

- 在記憶體視窗中按一下右鍵，並從彈出選單中選擇 **Go To...**（轉至.....）

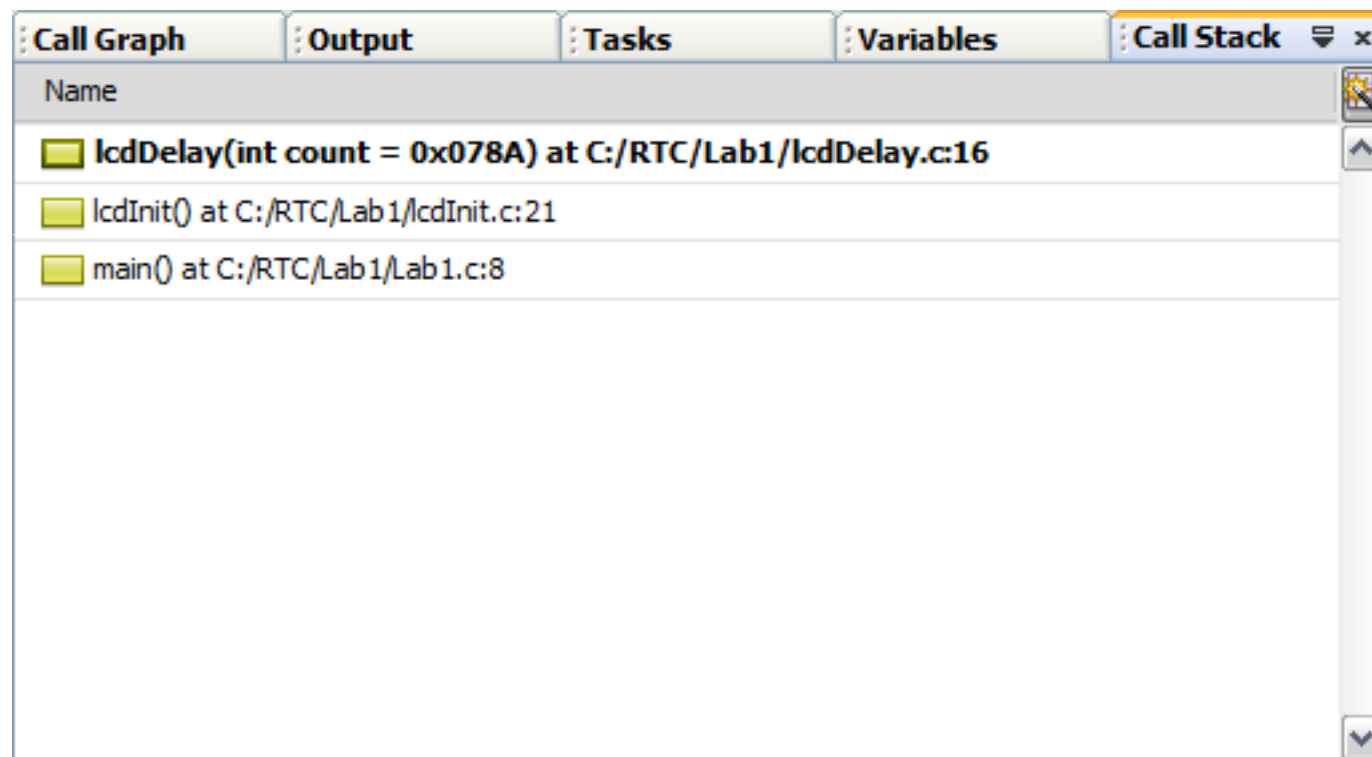


此顯示視窗以 **16-bit** 的元件為範例

叫用堆疊

如何查看叫用堆疊

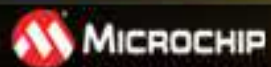
- 選擇 **Window► Debugging► Call Stack** (叫用堆疊)



對某個堆疊級按兩下或按右鍵，在編輯器中顯示執行叫用的程式行。

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



Lab 3

在硬體上除錯專案

Lab 3

使用 XC8 的中斷設定

- 在 **PIC18F** 設定高優先權

```
void interrupt ISR_High (void)
{
    if (PIR1bits.TMR1IF)
    {
        PIR1bits.TMR1IF= 0 ;
        :
    }
}
```

- 在 **PIC18F** 設定低優先權

```
void interrupt low_priority ISR_Low(void)
{
    if (PIR1bits.TMR1IF)
    {
        PIR1bits.TMR1IF= 0 ;
        :
    }
}
```

Lab 3 的動作

- **Lab 3** 的動作採用高優先全中斷方式運作
- 設定基礎中斷時間為 **1mS**
- 主程式檢查是否已到 **1 Second** 的計時了，將時間的顯示在 **LCD** 上
- 開啟專案位置：
 - C:\MPLAB X IDE RTC\PIC18F Labs\Lab3\X IDE Lab3.x

Lab 3

在硬體上除錯專案



目標

- 選擇一個除錯器/程式設計器作為硬體除錯工具
 - 將除錯功能關閉後，可以在 **Dashboard** 下更換除錯工具。
- 使用硬體工具編譯並燒錄程式的除錯版本
- 控制硬體平臺上程式的執行
- 檢查如何在除錯對話期間直接寫入變數和 SFR

Lab 3

在硬體上除錯專案

步驟

打開 Lab 3

- 按右鍵專案名稱並選擇 **Close**（關閉）來關閉任何打開的專案
- 按一下“Open Project”（打開專案）圖示
- 瀏覽至並打開專案 **C:\MPLAB X IDE RTC\PIC18F Labs\Lab3\X IDE Lab3.X**
- 確保除錯對話將起始於main()開頭
- Tools -> Options -> Embedded -> “Generic Settings”（工具 -> 選項 -> 嵌入式 -> “通用設置”）
- 選擇**Debug startup**（除錯啟動）作為Main



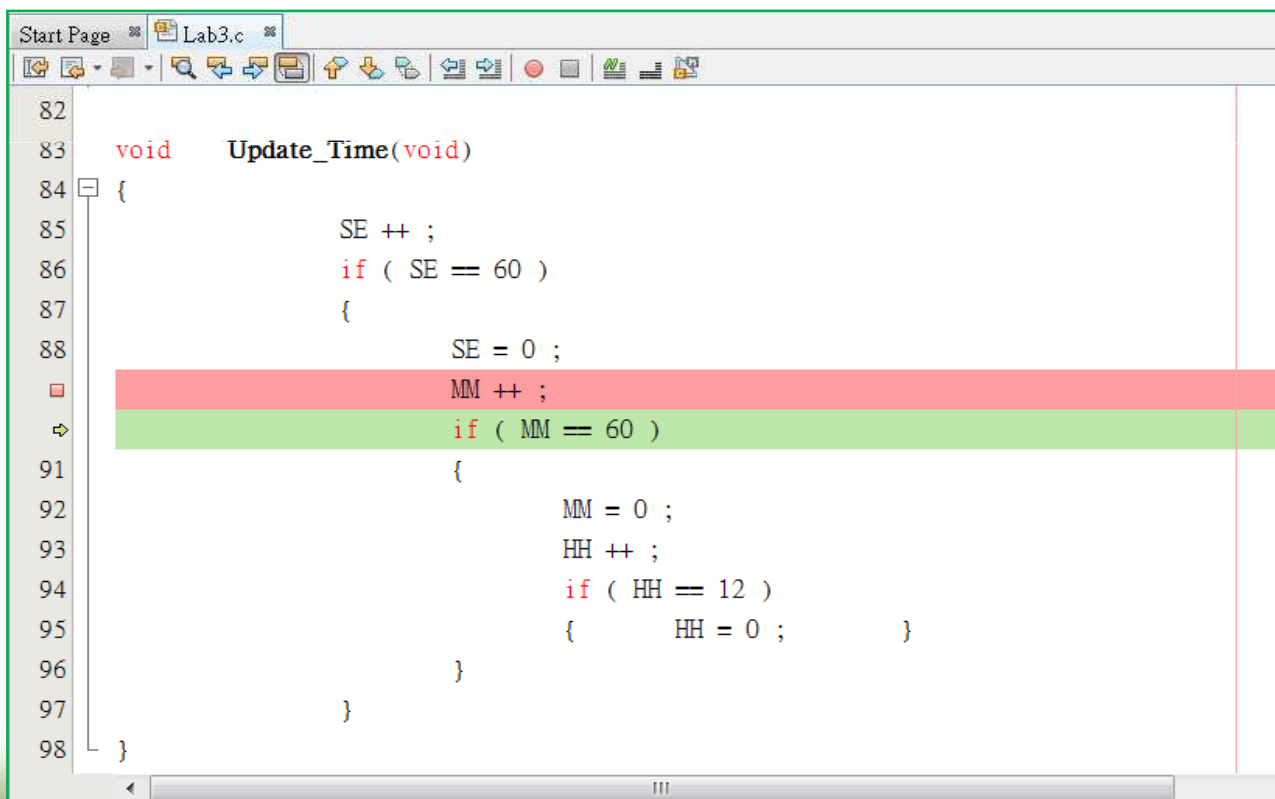
- 按一下“Debug”（除錯）圖示編譯並除錯

Lab 3

在硬體上除錯專案

步驟

編譯完成後進入除錯模式，在行號處按一下老鼠又鍵設定斷點，這時整行敘述變成紅色。先按 **Reset** 圖示後，再按執行圖示。程式開始執行，LCD 的秒數開始增加，直到計數到 **59** 秒後停下。綠色為程式暫停時的 PC 位置。



```
82
83 void    Update_Time(void)
84 {
85     SE ++ ;
86     if ( SE == 60 )
87     {
88         SE = 0 ;
89         MM ++ ;
90         if ( MM == 60 )
91         {
92             MM = 0 ;
93             HH ++ ;
94             if ( HH == 12 )
95             {
96                 HH = 0 ;
97             }
98         }
99     }
100 }
```

Lab 3

在硬體上除錯專案


步驟





















從程式中刪除中斷點

- 在Windows®功能表中選擇 Debugging，然後選擇Breakpoints
- 在Breakpoint 視窗中按右鍵，然後選擇 Remove All Breakpoints（全部刪除）

向 **Watch** 視窗中加入變數

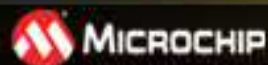
- 在 Windows 功能表中選擇 Debugging，然後選擇 Watches
- 在 Watch 中按右鍵，然後選擇“**New Watch...**”
- 將變數 **SE**，**MM**、**HH** 加入到 Watch 視窗中，修改它們的值，然後按一下

繼續按鈕  並觀察時間會發生什麼。

tput		Tasks	Configuration Bits	Variables	Call Stack
Name	Type	Address	Value		
 SE	unsigned char	 0x33	 0x00		
 MM	unsigned char	 0x31	 0x01		
 HH	unsigned char	 0x30	 0x02		
 <Enter new watch>					

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC MCU PLATFORM



使用當地歷史紀錄

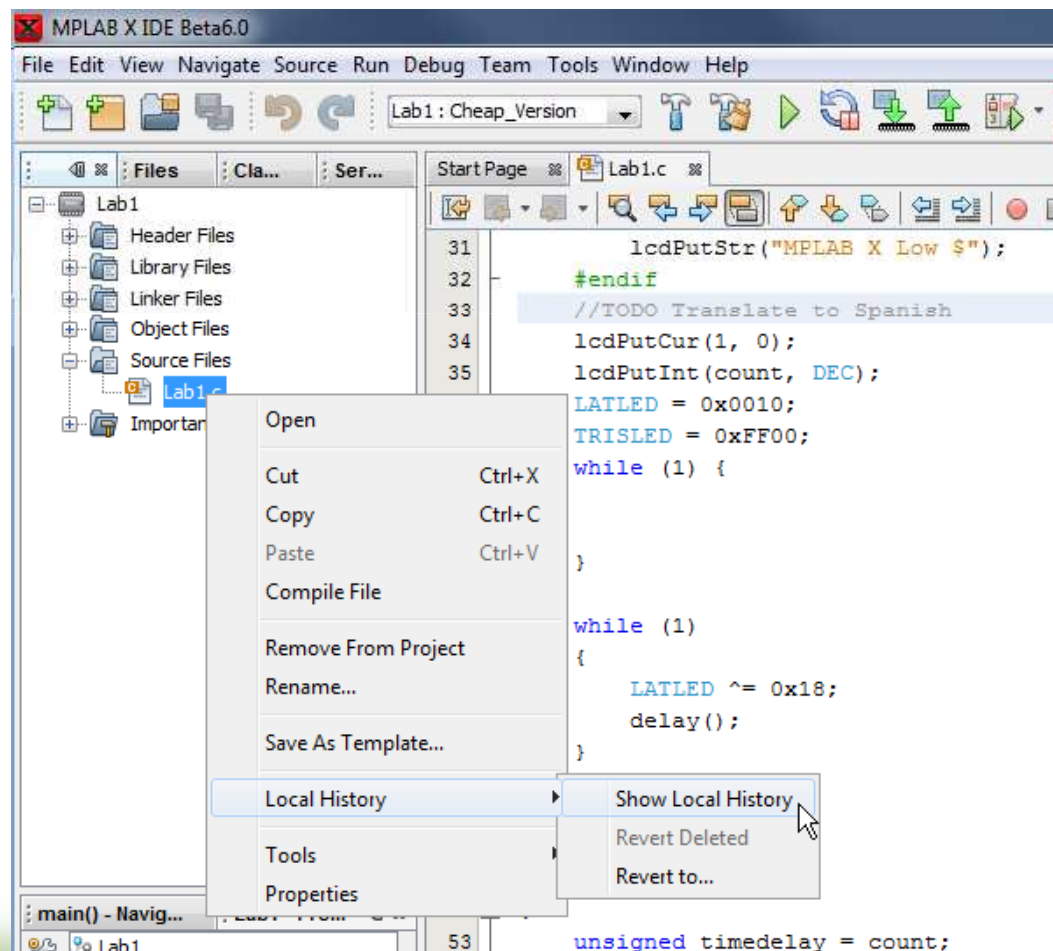
原始檔改錯了怎樣恢復

找出當地歷史紀錄

1 選擇原始檔案並顯示其當地歷史紀錄

按右鍵專案樹中的原始檔案，
並從彈出功能表中選擇：

Local History（當地歷史紀錄）
► **Show Local History**
（顯示當地歷史紀錄）



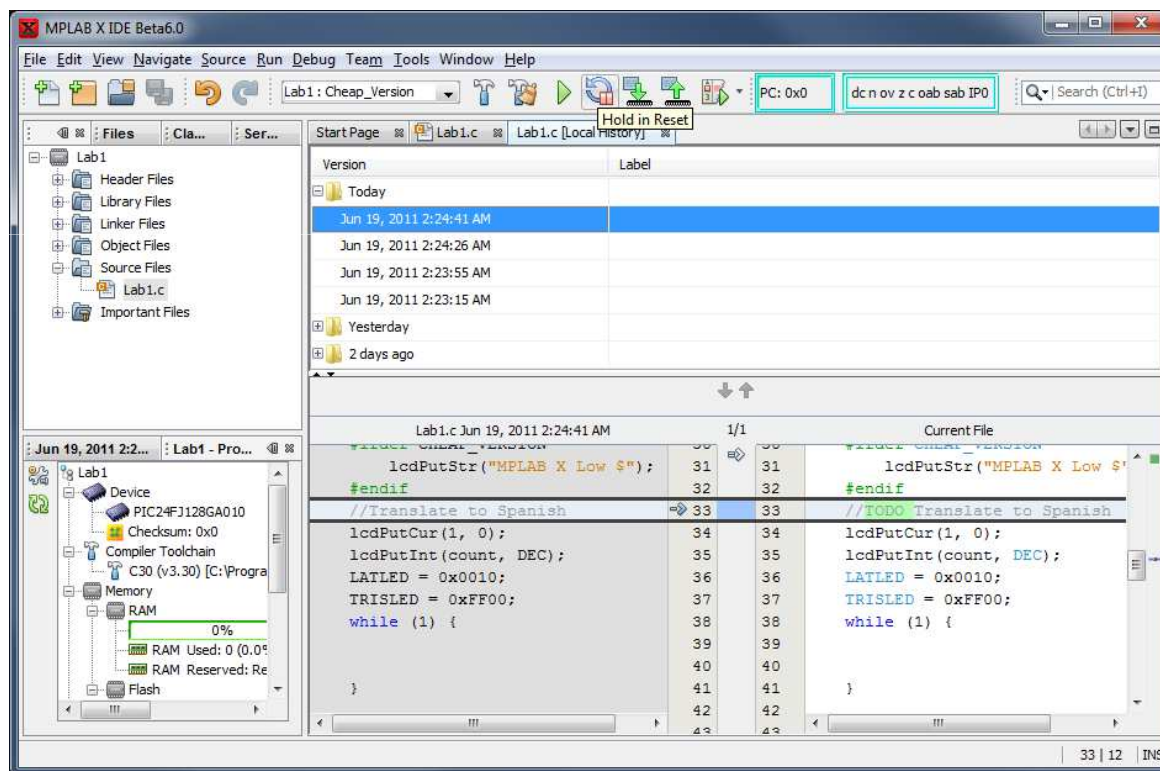
顯示當地歷史紀錄

2 在上部的窗格中選擇一個之前的版本

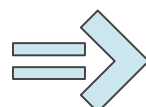
將在編輯器區域打開一個新的選項卡。

上部的窗格顯示選定檔之前版本的列表。

下部的窗格顯示目前版本（右邊）和上面所選擇的先前版本（左邊）的不同點比較。

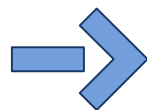


使用當地歷史紀錄

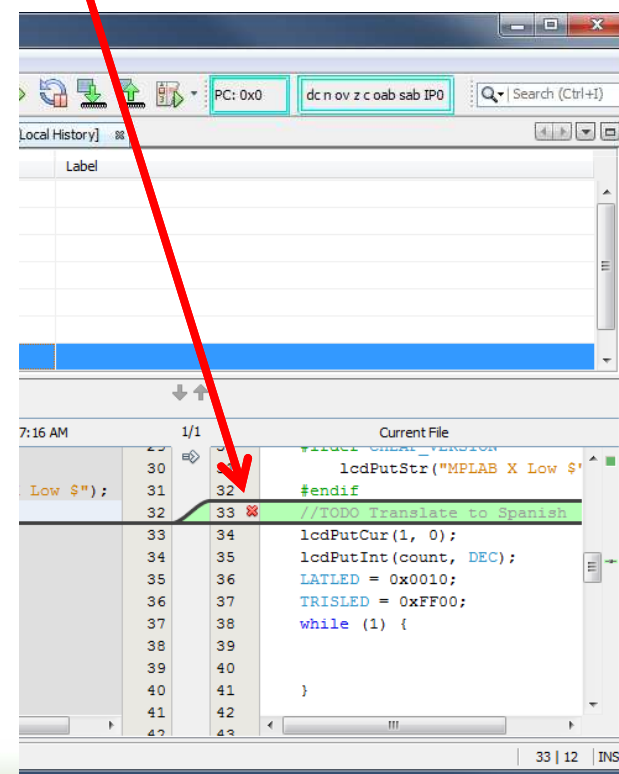
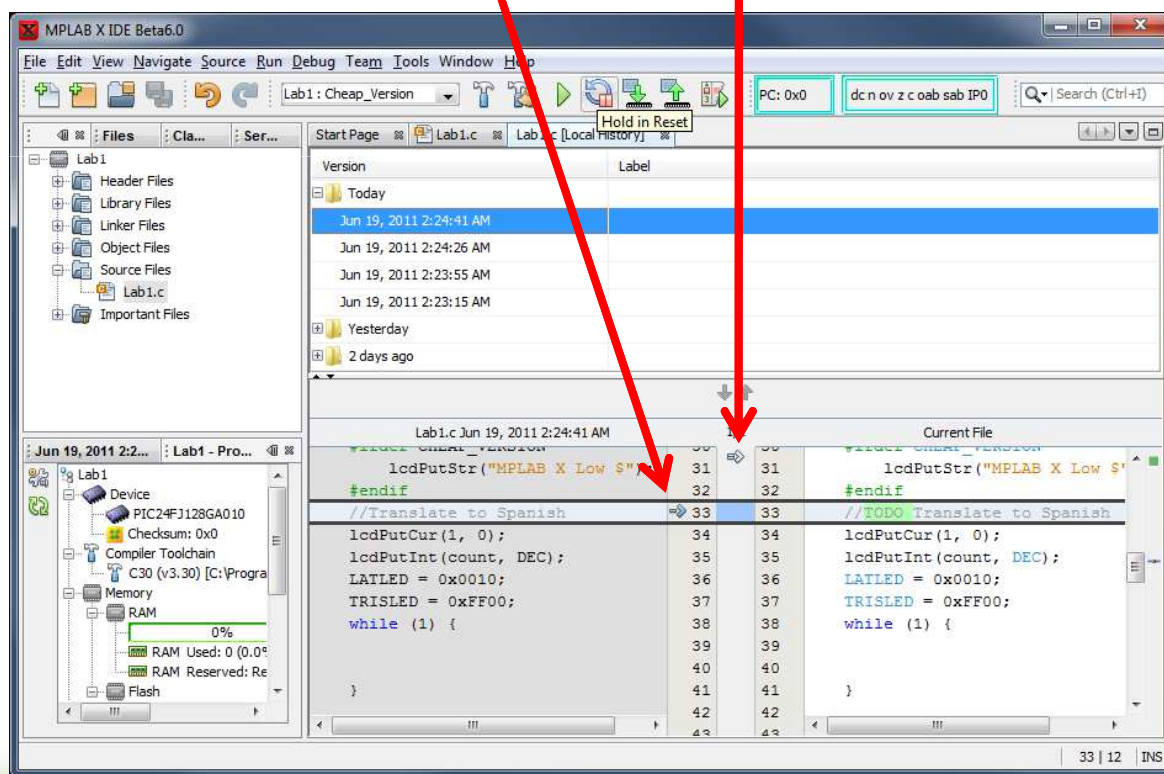


將整個檔案修復至先前版本（用先前版本的檔替換目前的版本）。

將先前版本的選項加入至目前版本。



從目前版本中刪除項，以與先前版本匹配。



設定當地歷史紀錄的天數

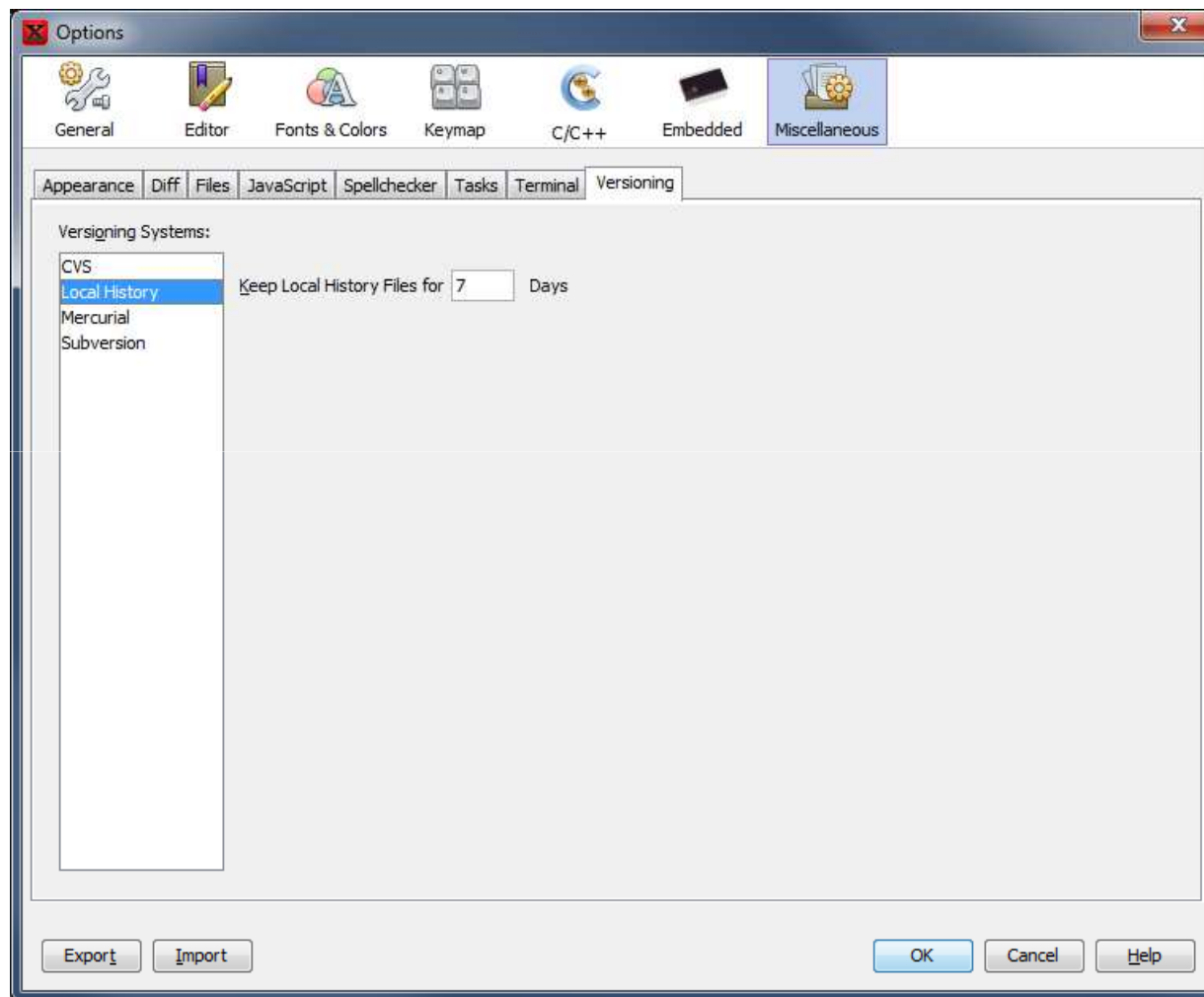
要改變當地歷史紀錄持續跟蹤檔的時間長度，應從主功能表中選擇：

Tools ▶ Options

選擇 **Miscellaneous**（其他方面）類別

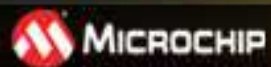
選擇 **Versioning**（版本）選項卡

在左邊的 **Versioning Systems**（版本系統）中選擇 **Local History**



Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



舊的專案轉換

MPLAB IDE to MPLAB X IDE

轉換的準備事項 (一)

- 確保專案在**MPLAB® IDE v8**中編譯
- 確保 **IDE** 的版本為 **8.14.03A** 或之後的版本
- 確保所有標頭檔都存在於專案樹中，不僅僅是透過 **#include** 包含在程式中
- 若轉換是為了適用於 **Linux** 或 **Mac** 平臺：
 - 確保**#include** 檔案名採用正確的大小寫
例如，“p24FJ128GA010.h”——不能寫作“p24fj128ga010.h”
 - 在相對路徑中使用 “/” 而不是 “\
例如，`#include "Graphics/graphics.h"`

轉換的準備事項 (二)

中文顯示的問題 (參考練習一)

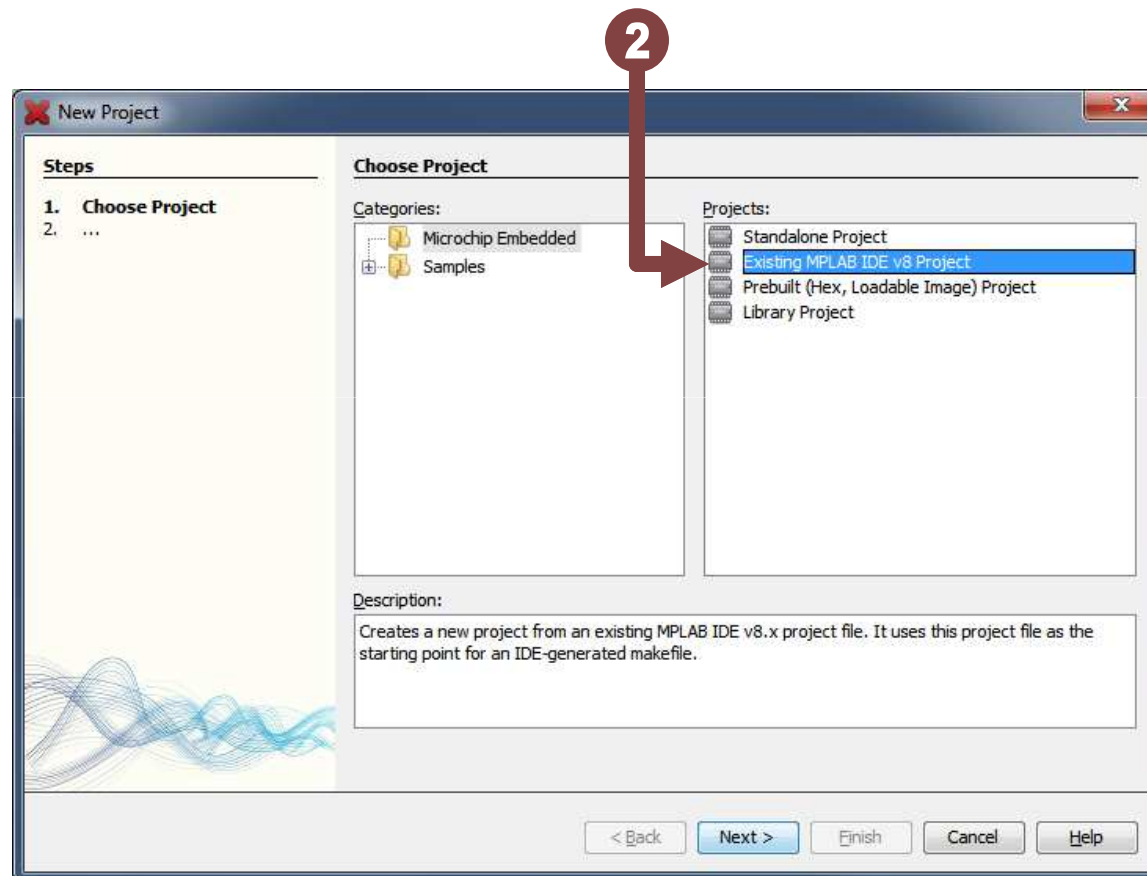
- **MPLAB IDE 使用 Big 5 編碼**
 - 直接設定 **X IDE 使用 Big 5 編碼**
- **MPLAB X IDE 使用 UTF-8 編碼**
 - 將原先在 **MPLAB v8.x** 的原始程式檔 (*.C) 先用”記事本”開啟後再用 **Save as** 方式選擇 **UTF-8 編碼**後回存。

轉換後

- 可以在 **MPLAB® 8** 和 **MPLAB X** 專案中共用原始檔案
- 在兩種 **IDE** 下均可對專案進行操作
 - 對原始檔案的修改在兩種 **IDE** 中均有反映
 - 修改專案設置僅對當前 **IDE** 有效
- **MPLAB X** 專案檔案位於 **MPLAB 8** 專案目錄的子目錄中
- 注意一下除錯工具的切換

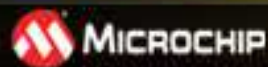
如何轉換舊有專案

- 1 按一下新建專案圖示開始新專案 
- 2 在 **Projects** (專案) 下選擇 **Existing MPLAB® IDE v8 Project** (現有MPLAB® IDE v8 專案)
- 3 按照嚮導中的其餘步驟進行操作，這些步驟與新建一個獨立項目非常類似。



Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



Lab 4

將 **MPLAB[®] 8** 專案
轉換為 **MPLAB X** 格式

Lab 4

如何將MPLAB®8專案轉換為MPLAB X格式



目的

- 完成本實驗後，您將能夠：
 - 將一個有用的 MPLAB® 8 專案轉換為新的MPLAB X 格式
 - 在 MPLAB IDE 8 和 MPLAB X IDE 中使用同一個專案

Lab 4

如何將 **MPLAB®8** 專案轉換為 **MPLAB X** 格式



目標

- 本專案將一個簡單的 **ADC** 應用從 **MPLAB® 8** 格式轉換為 **MPLAB X** 格式
- 練習四動作為：每 **100mS** 讀取 **ADC 10-bit** 值後，顯示在 **LCD** 上
 - ◆ 本練習有用高優先權中斷，中斷時間為 **1mS**

Lab 4

如何將 MPLAB®8 專案轉換為 MPLAB X 格式




注意事項

- **Lab 4 在 IDE 下的專案**
 - 使用 **XC8** 編譯器
 - 使用 **Big-5** 中文編碼格式
- 注意一下，**ICD3** 否有切換到 **X IDE**
- **Lab 4 的路徑:**
 - C:\MPLAB X IDE RTC\PIC18F Labs\Lab4

Lab 4

如何將MPLAB®8專案轉換為MPLAB X格式

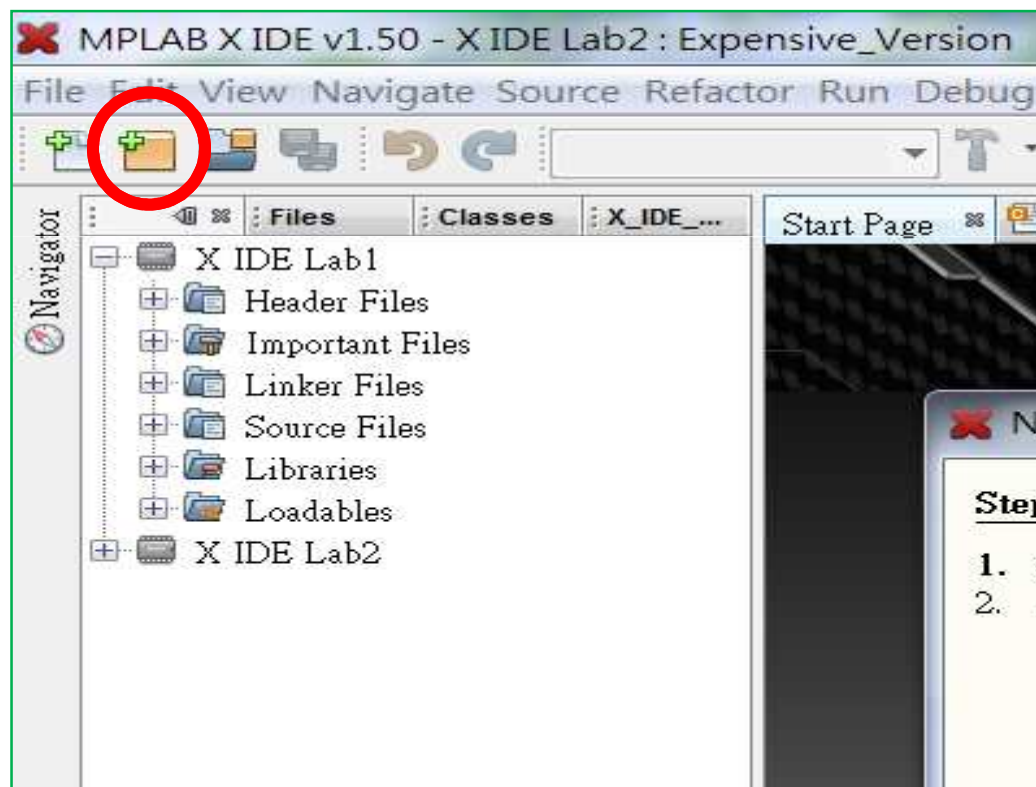
1 啟動新建專案精靈

按一下工具列上的 

或

從功能表中選擇：

File ► New Project



Lab 4

如何將MPLAB®8專案轉換為MPLAB X格式

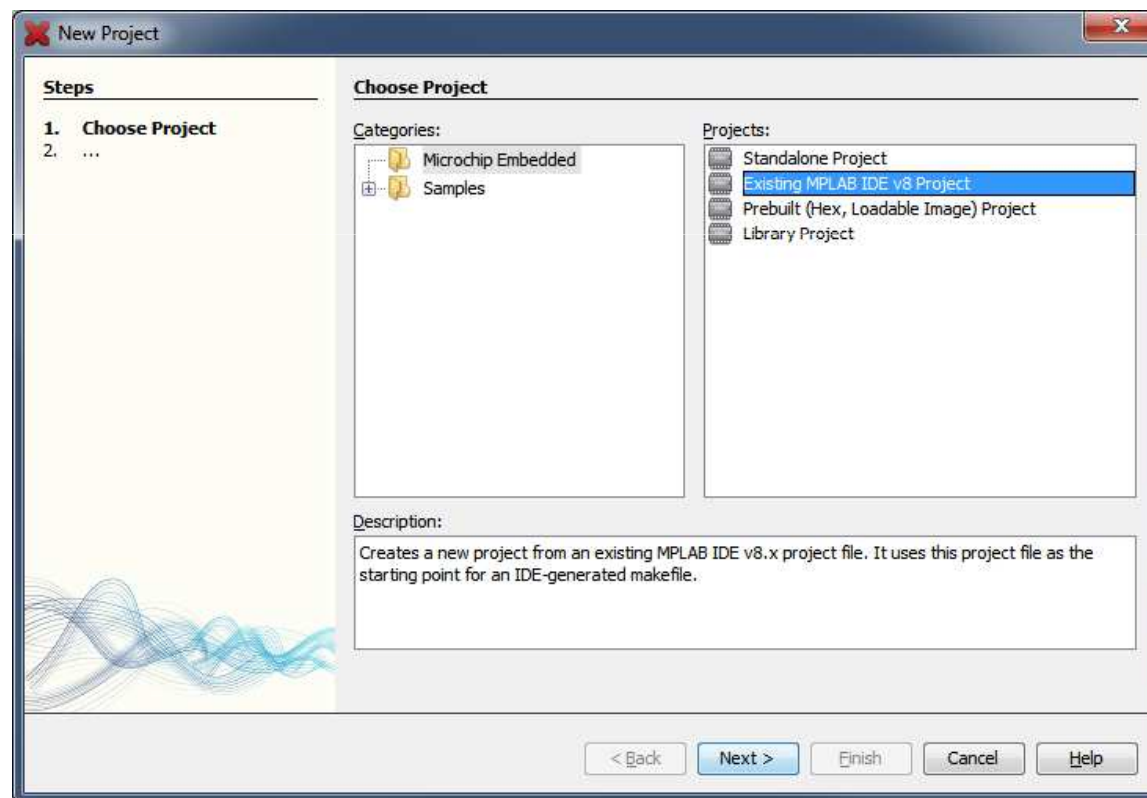
2 選擇專案類型

在“Categories”下，選擇：
Microchip Embedded

在“Projects”下，選擇：
Existing MPLAB IDE v8 Project

按一下

Next >



Lab 4

如何將MPLAB®8專案轉換為MPLAB X格式

3 選擇 MPLAB 8 專案檔案

使用Lab4.mcp



我的電腦



當地磁片 (C:)



MPLAB X IDE RTC



PIC18F Labs



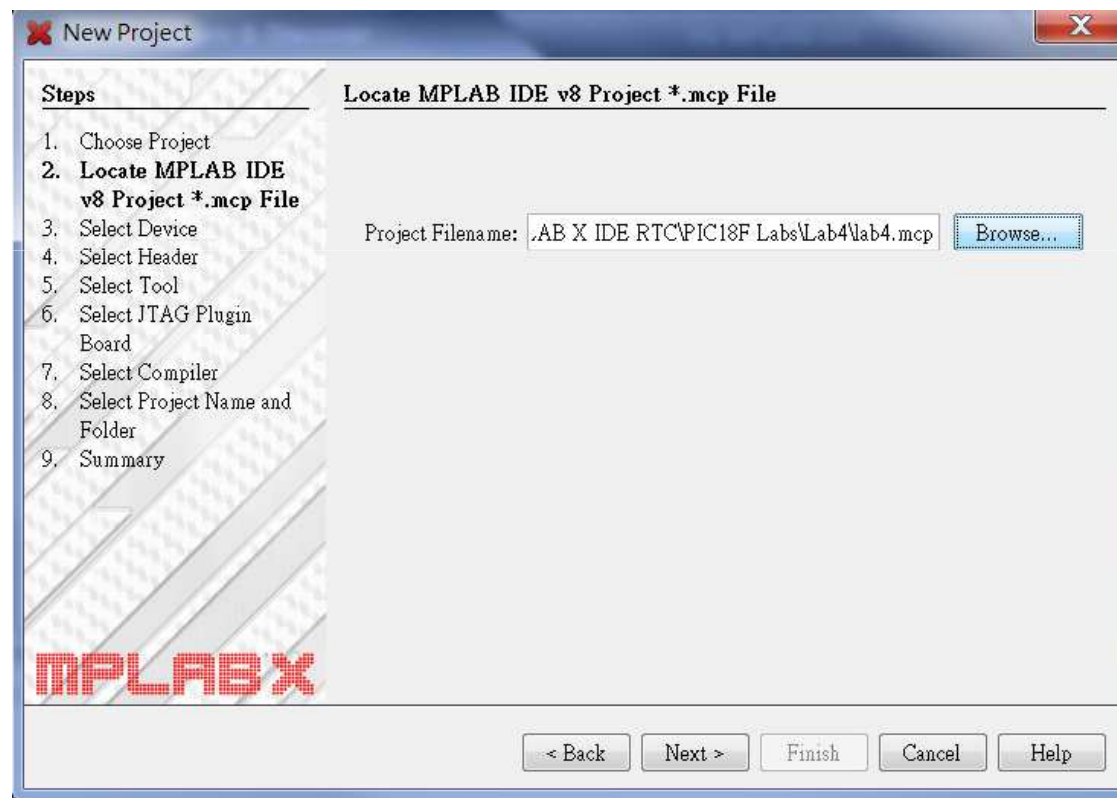
Lab4



lab4

按一下

Next >



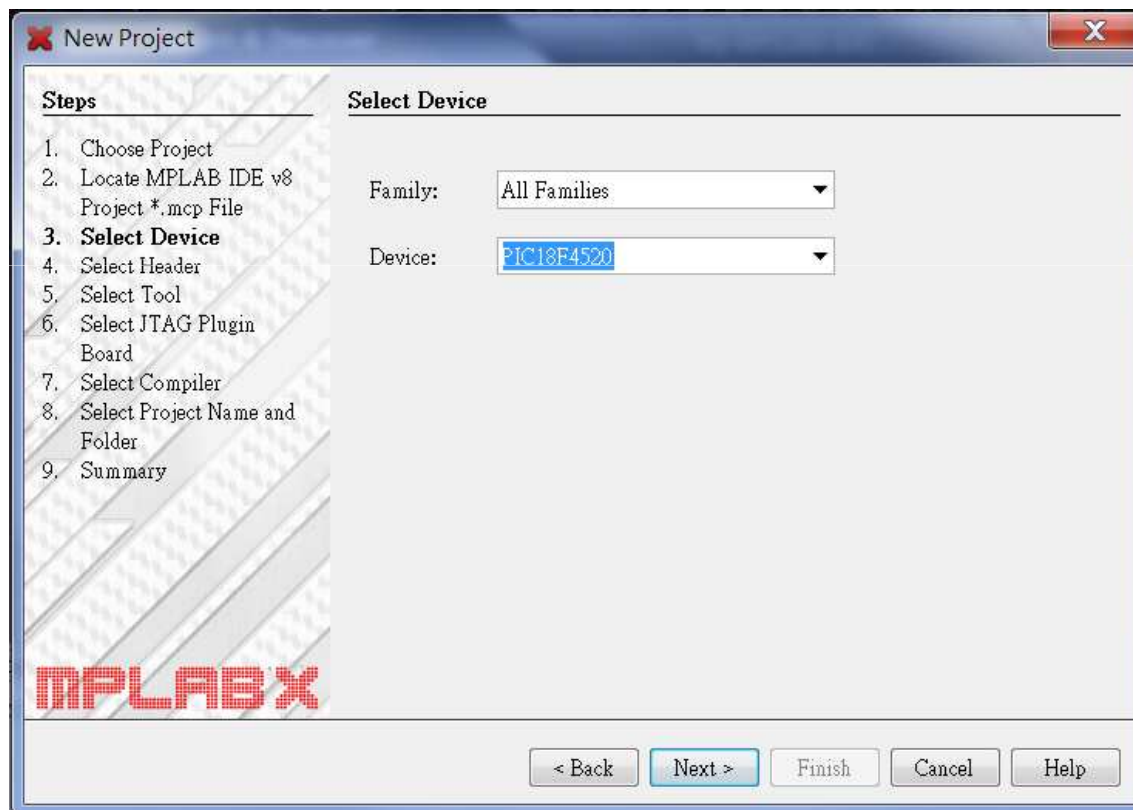
Lab 4

如何將MPLAB®8專案轉換為MPLAB X格式

4 選擇元件

將元件改為
PIC18F4520

按一下 **Next >**



Lab 4

如何將MPLAB®8專案轉換為MPLAB X格式

5 選擇模擬頭

不使用任何模擬轉接座。
保留核取方塊不選，並按一

Next >

PIC18F4520 無需使用
轉接座就可以直接使用
內建的除錯模組除錯。



Lab 4

如何將MPLAB®8專案轉換為MPLAB X格式

5 選擇工具

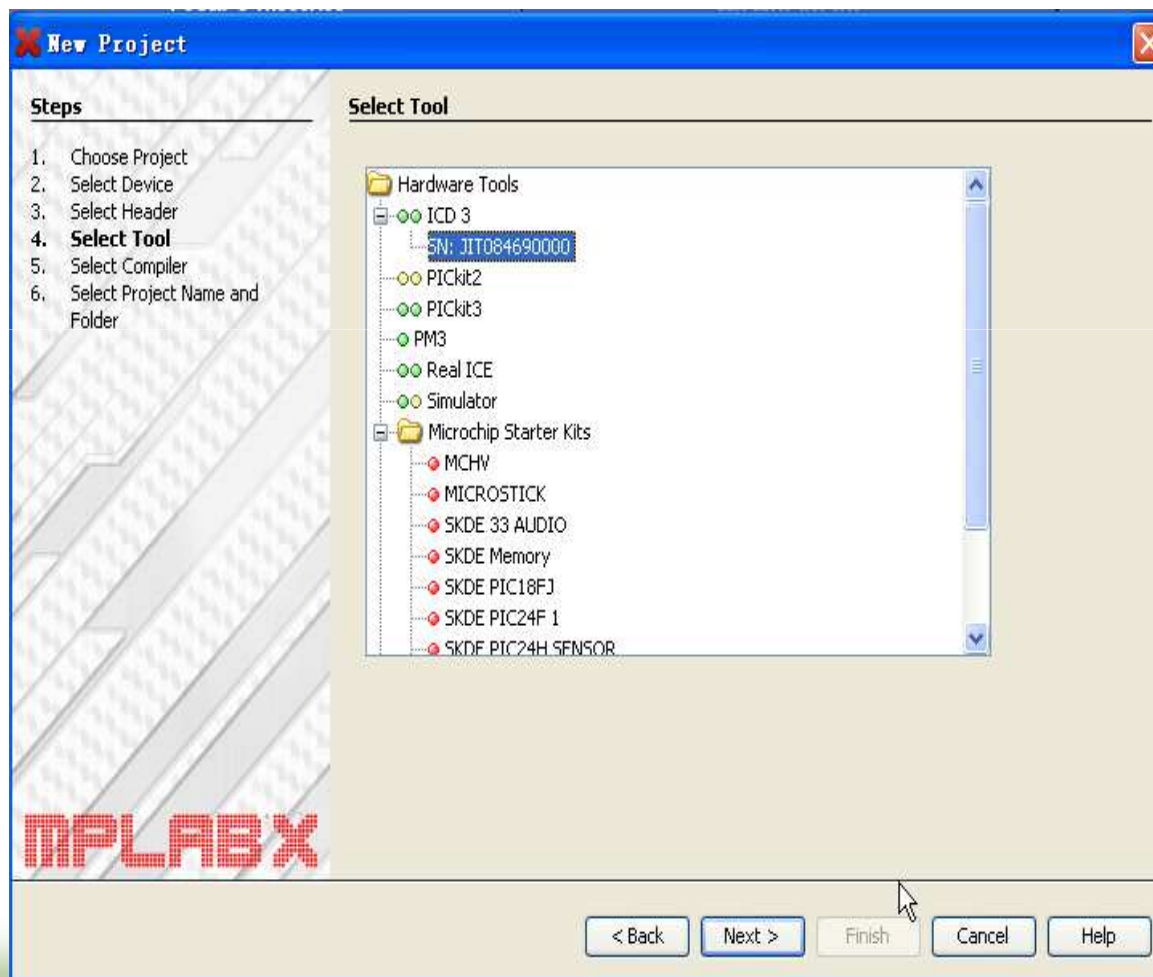
a 選擇 ICD3 SN



如果使用硬體除錯工具，則選擇其序號，如右側ICD3下方所示。

確認 ICD3 已成功切換成 X IDE 的驅動模式

b 按一下 **Next >**



Lab 4

如何將MPLAB®8專案轉換為MPLAB X格式

7 選擇編譯器

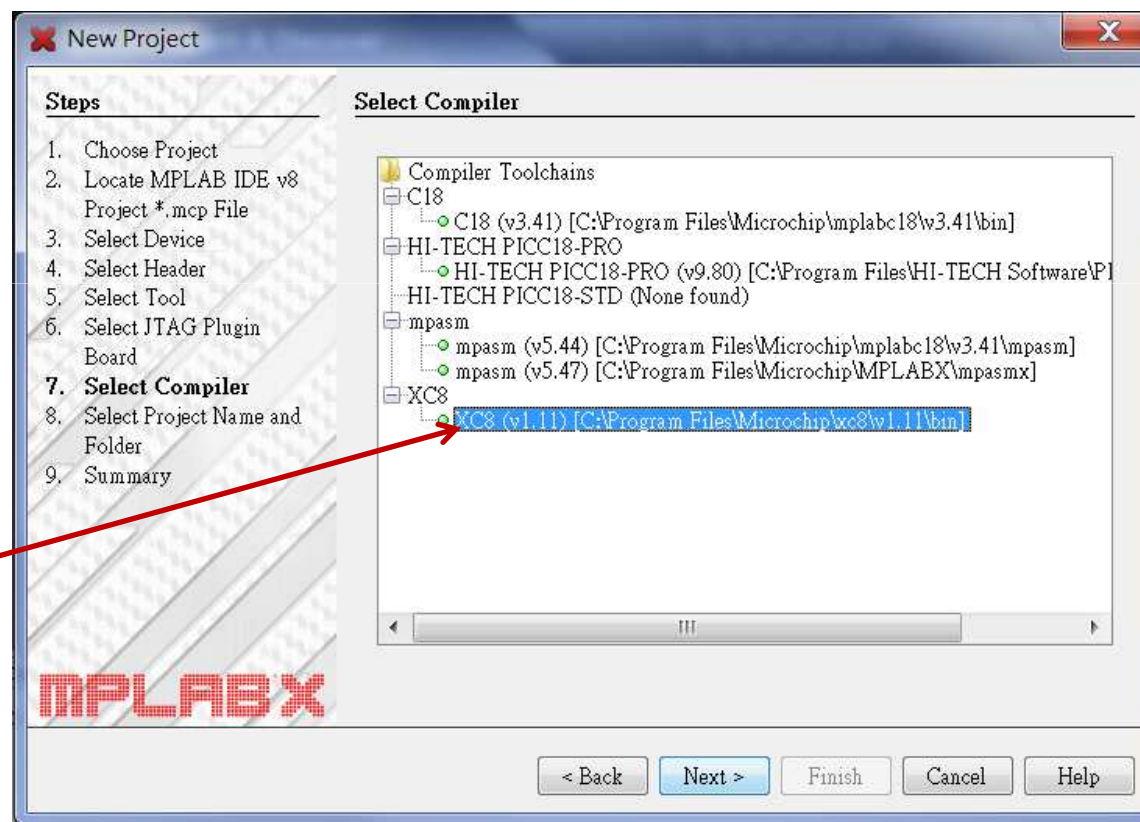
選擇編譯器：

■ **MPLAB XC8 v1.11**（或更高版本）



確保按一下版本號，而不是工具的標題。

按一下 **Next >**



Lab 4

如何將MPLAB®8專案轉換為MPLAB X格式

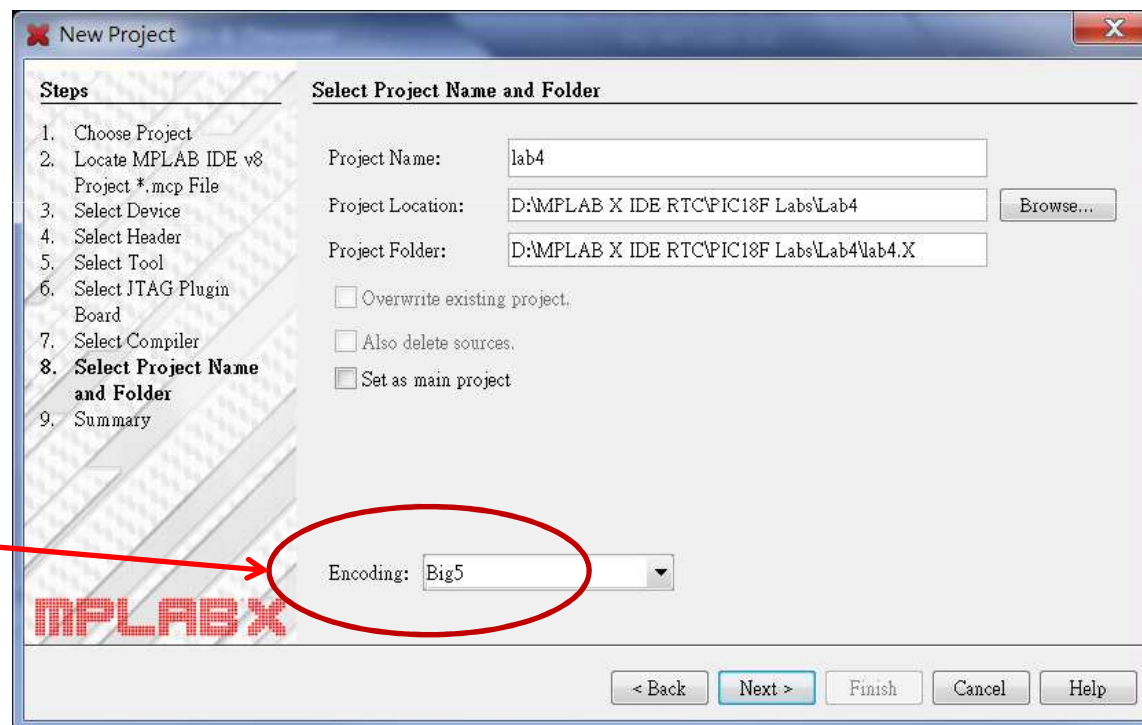
8 選擇專案名稱和資料夾

使用內定的專案名稱和資料夾

這些設置可以更改，但對於本實驗，請保留其預設值

注意：因 Lab4.mcp 使用 Big-5 編碼格式，為方便轉換故仍使用 Big-5 編碼。

按一下 **Next >**



Lab 4

如何將MPLAB®8專案轉換為MPLAB X格式

9 總結

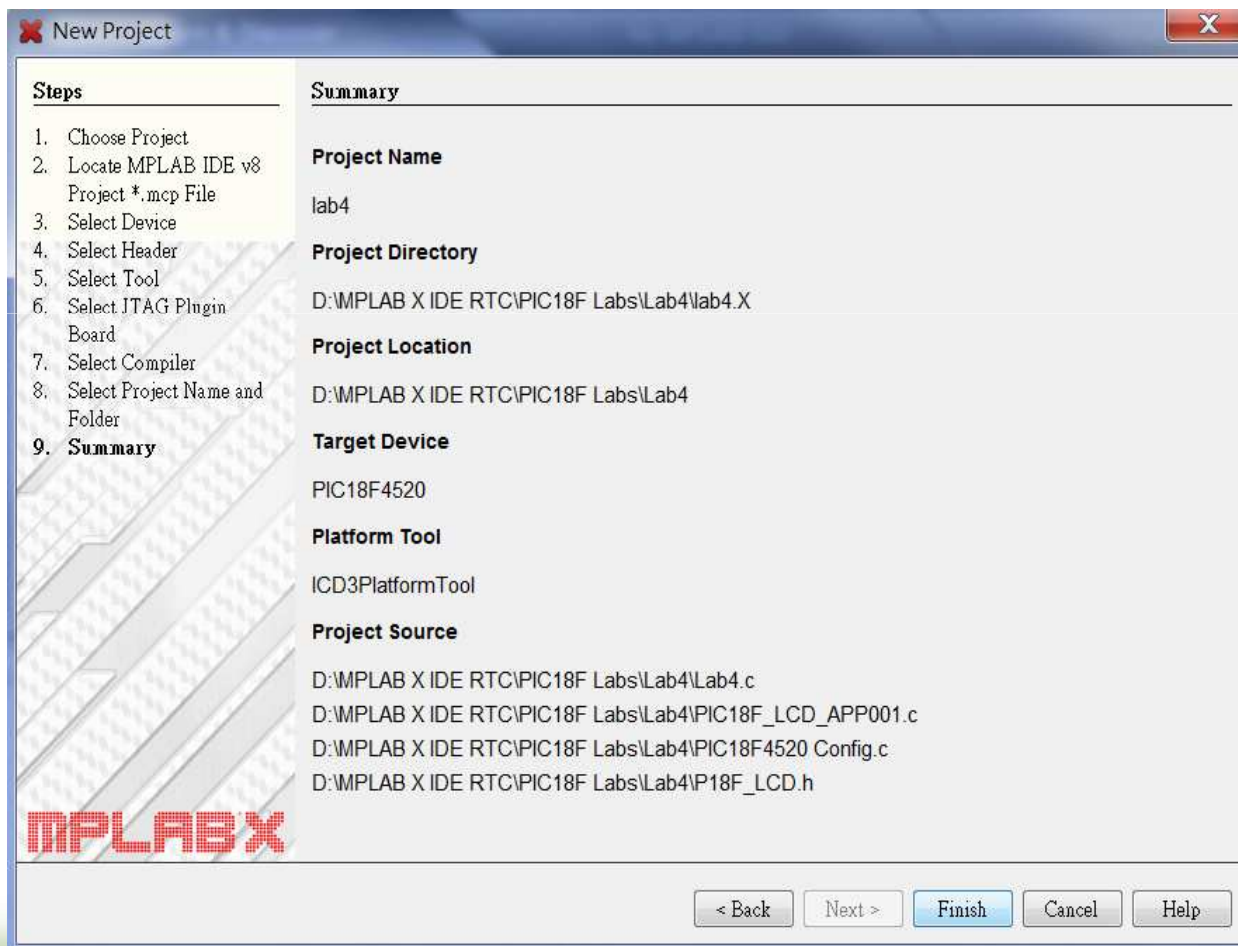
查此次的轉換檔案來源與
換後的儲存路徑。

認使用的元件及開發發工
。

認所使用的原始程式黨是
正確。

間檔的輸出路徑不會保留
設置，因為MPLAB® XIDE
用預定義的目錄保存中間
。

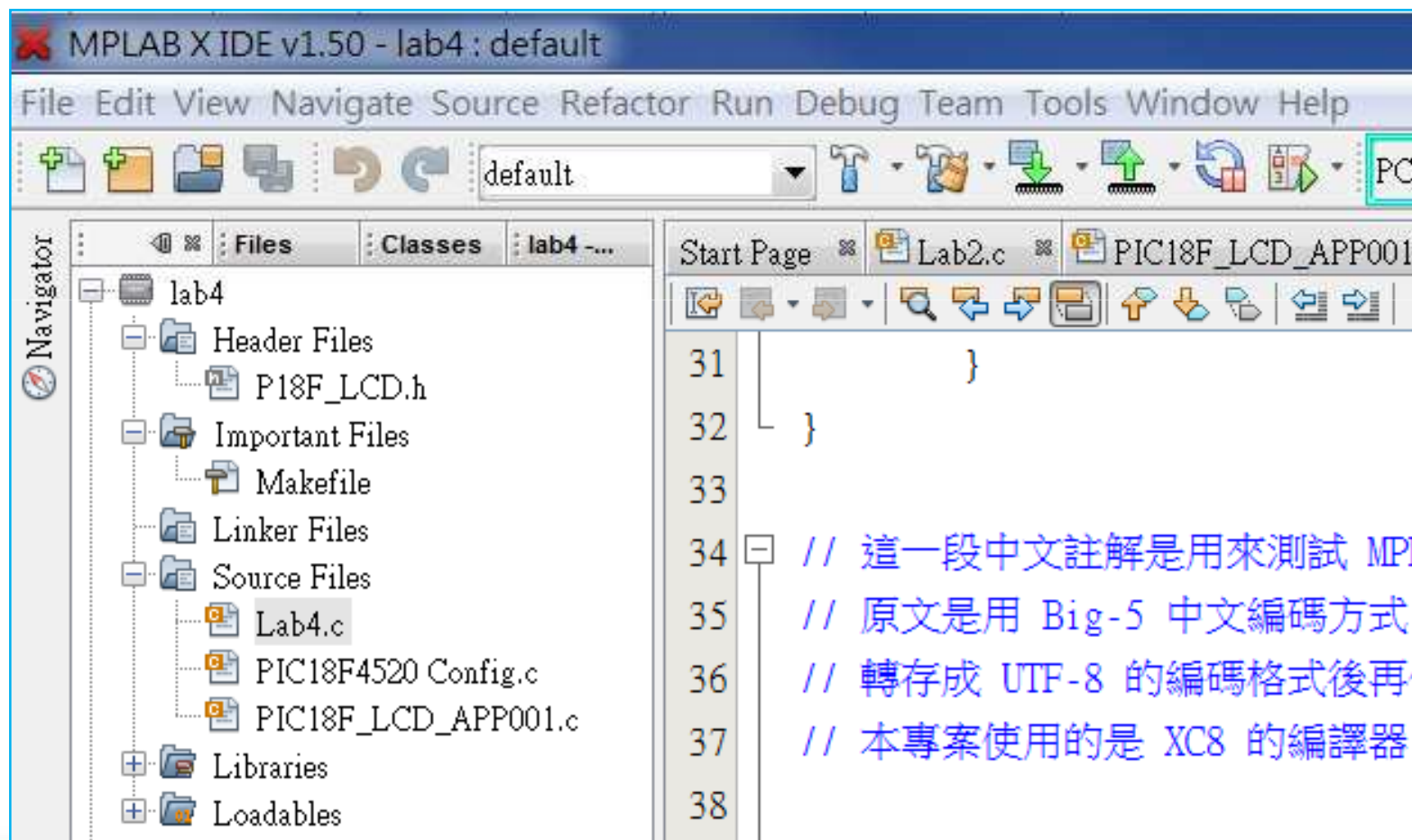
一下 **Finish**



Lab 4

如何將MPLAB®8專案轉換為MPLAB X格式

轉換完後可以在 X IDE 看到專案裡的檔案程式中的中文註解部分也可以顯示



Lab 4

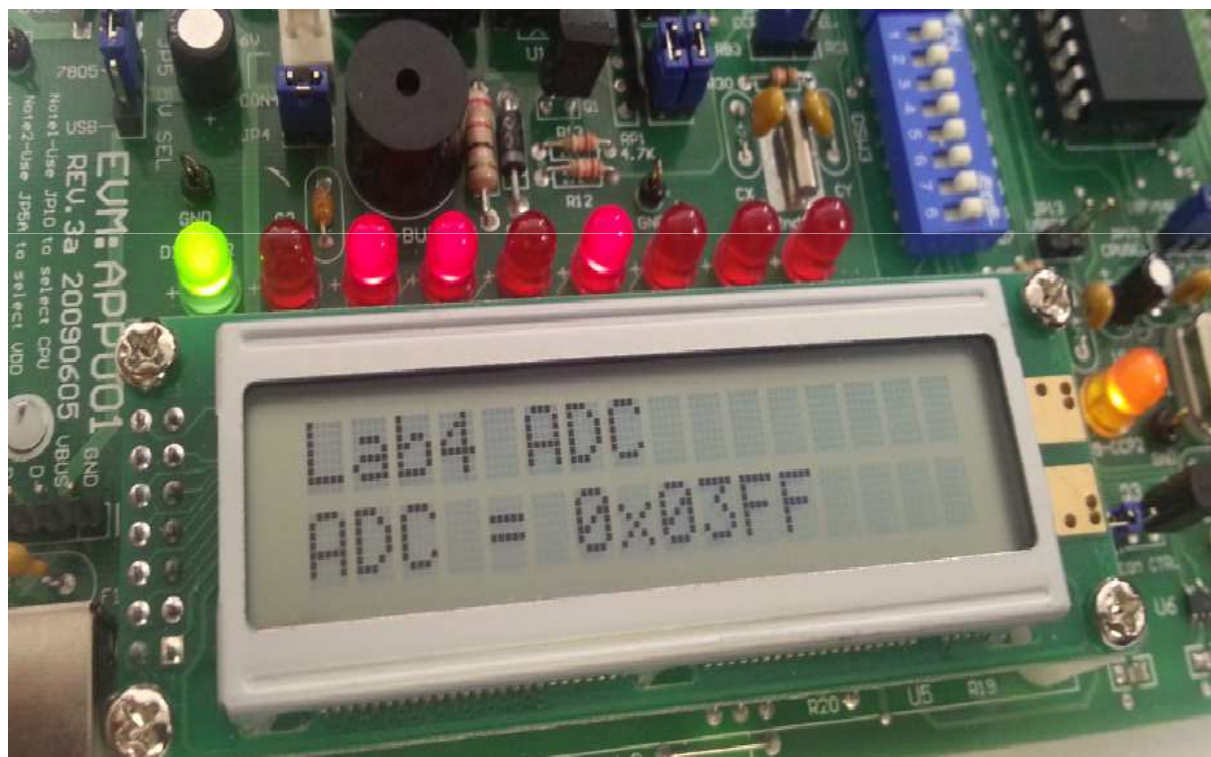
轉換為MPLAB X 格式後的執行

10 編譯並執行專案

按一下工具列上的Debug
Project 圖示：



轉動一下 VR3 觀察 LCD 模
組上的 ADC 轉換顯示



Lab 4

如何將 **MPLAB®8** 專案轉換為 **MPLABX** 格式

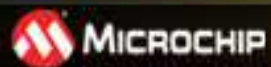


結論

- 該專案轉換實用程式可以方便地將舊有 **MPLAB® 8** 的專案轉換為新的 **MPLAB X** 的格式
- 兩個版本的 **IDE** 之間可以共用程式，從而簡化了轉換。

Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



如何使用多專案

如何使用多專案

活動專案與主專案

活動專案

活動專案 即當前在專案樹中選定的專案。在選擇不同的專案之前，所有與專案有關的操作都將應用至活動專案。

主專案

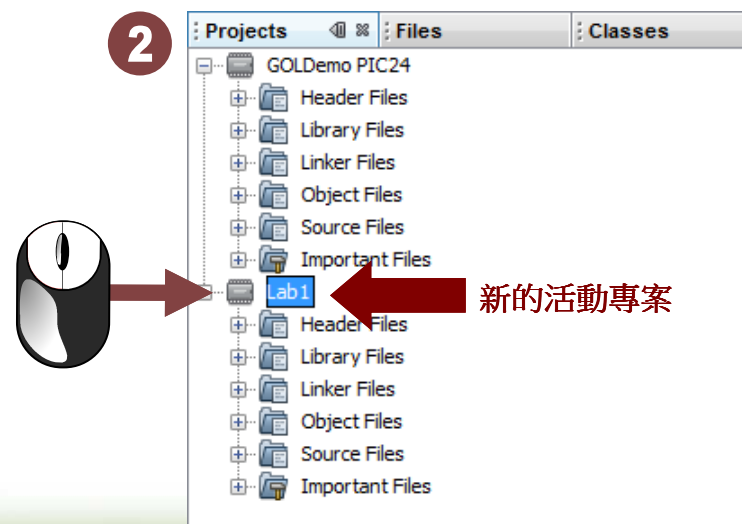
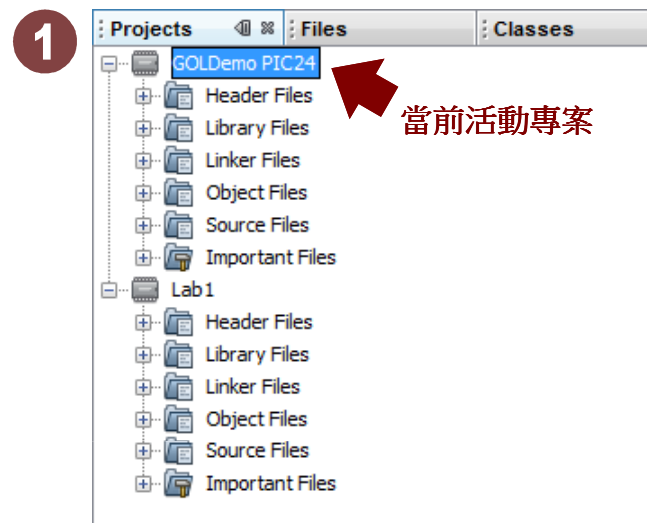
主專案 是通過滑鼠右鍵功能表明確選定的專案。一旦選定後，無論在專案樹中選擇了哪個專案，所有與專案相關的操作均將應用至主專案。

如何使用多專案

如何選擇活動專案

在專案樹中，左鍵按一下要作為活動專案的專案。

選定專案後，所有顯示視窗均將給出新選定專案的相關資料，並且所有操作均將應用至該專案。

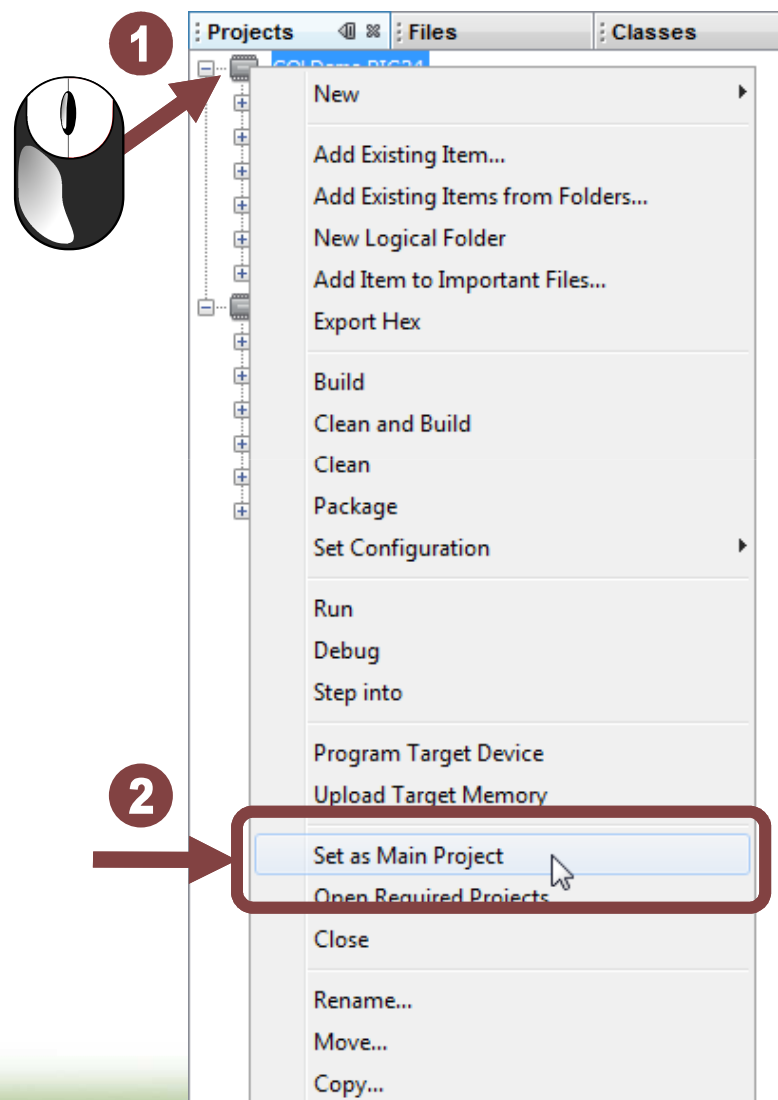


如何使用多專案

如何設置主專案

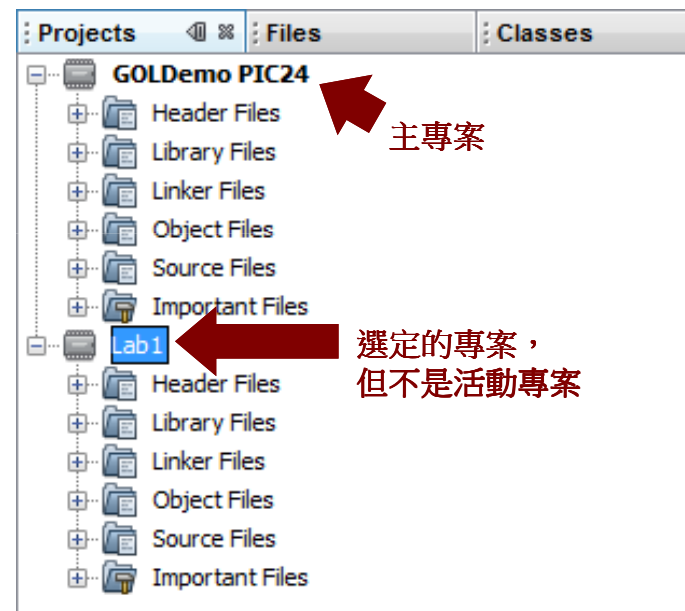
在專案樹中按右鍵想要設置為主專案的專案，然後從彈出選單中選擇 **Set as Main Project**（設置為主專案）。

所有顯示視窗均將給出與主專案相關的資料，並且所有操作均將只應用至主專案。



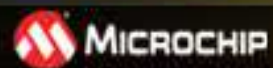
如何使用多專案 主專案

主專案將始終以粗體顯示，指示無論在專案樹中選擇了什麼專案，主專案都將是活動專案。



Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC[®] MCU PLATFORM



總結

資源

MPLAB® X IDE下載

<http://www.microchip.com/mplabx>

MPLAB® X IDE開發人員幫助中心

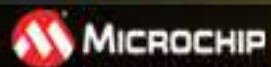
http://microchip.wikidot.com/mplab:_start

問題？



Reduce your development time • Reuse your code • Scale up or down

ONE DEVELOPMENT ENVIRONMENT



ONE PIC MCU PLATFORM



謝謝！

商標

Microchip的名稱和徽標組合、**Microchip**徽標、**dsPIC**、**FlashFlex**、**KeeLoq**、**KeeLoq**徽標、**MPLAB**、**PIC**、**PICmicro**、**PICSTART**、**PIC³²**徽標、**rfPIC**、**SST**、**SST**徽標、**SuperFlash**及**UNI/O**均為**Microchip Technology Inc.**在美國和其他國家或地區的註冊商標。

FilterLab、**Hampshire**、**HI-TECH C**、**Linear Active Thermistor**、**MTP**、**SEEVAL**和**The Embedded Control Solutions Company**均為**Microchip Technology Inc.**在美國的註冊商標。**Silicon Storage Technology**為**Microchip Technology Inc.**在除美國外的國家或地區的註冊商標。

Analog-for-the-Digital Age、**Application Maestro**、**BodyCom**、**chipKIT**、**chipKIT**徽標、**CodeGuard**、**dsPICDEM**、**dsPICDEM.net**、**dsPICworks**、**dsSPEAK**、**ECAN**、**ECONOMONITOR**、**FanSense**、**HI-TIDE**、**In-Circuit Serial Programming**、**ICSP**、**Mindi**、**MiWi**、**MPASM**、**MPF**、**MPLABCertified**徽標、**MPLIB**、**MPLINK**、**mTouch**、**Omniscient Code Generation**、**PICC**、**PICC-18**、**PICDEM**、**PICDEM.net**、**PICKit**、**PICtail**、**REAL ICE**、**rfLAB**、**Select Mode**、**SQL**、**Serial Quad I/O**、**Total Endurance**、**TSHARC**、**UniWinDriver**、**WiperLock**、**ZENA**和**Z-Scale**均為**Microchip Technology Inc.**在美國和其他國家或地區的商標。**SQTP**是**Microchip Technology Inc.**在美國的服務標記。

GestIC和**ULPP**為**Microchip Technology Inc.**的子公司**Microchip Technology Germany II GmbH & Co. &KG**在除美國外的國家或地區的註冊商標。

在此提及的所有其他商標均為各持有公司所有。

© 2012, **Microchip Technology Inc.**版權所有。