



MICROCHIP *2010*
MASTERs Conference

C11H02 MPX

MPLAB[®] IDE v10 入门

- **MPLAB[®] IDE和组件:**
 - 在Microchip网站上查找工具和支持
 - 选择所需单片机的相关硬件开发工具
- **MPLAB X简介**
 - 概述
 - 主要特性
- **MPLAB X概览**
- **实验1——创建新项目**
- **如何编译应用程序**
- **如何使用编辑器**
- **如何使用主工具栏**

- **实验2——如何转换MPLAB 8项目**
- 如何使用调试器
- 如何进行项目导航
 - 搜索、编辑器功能和导航菜单
 - 导航器窗口
 - 收藏夹
 - 调用图
- **实验3——如何使用Watch（观察）窗口进行调试**
- 如何使用本地历史记录
- **MPLAB 8与MPLAB X**
 - 项目属性对比
- **实验4——如何切换MPLAB驱动程序**
- **在Linux操作系统中运行MPLAB X IDE**



MICROCHIP *2010*

MASTERs Conference

MPLAB[®] IDE和组件

PICkit™ 3 Debug Express

- 调试器/编程器
- PIC18F45K20
演示板
- 12课教程
- 调试教程



(DV164131)

MPLAB[®] ICD 3

- 高级调试
- 高速编程
- 低电压支持
- 测试接口模块
- **USB供电**



(DV164035)



MASTERS
CONFERENCE 2010

MPLAB[®] REAL ICE[™]

在线仿真器



- 高速USB接口
(480 Mb/s)
- 实时跟踪、观察和调试
- 数据捕捉/触发引脚
- 跑表
- 快速编程

(DV244005)

高性能工具包

(AC244002)



MPLAB[®] ICD/ICE资源需求

- 独占两个I/O引脚
- MCLR引脚的复用控制
- 一些文件寄存器（RAM）
- 一些程序存储单元（闪存）
- 一或两个硬件堆栈单元

一定需要



可能不需要

更多
信息...

- MPLAB IDE Help（帮助）⇒ Topics ...（主题...）⇒ Debuggers（调试器）⇒ PICkit™ / MPLAB[®] ICD / MPLAB ICE

仿真器选择

特性	PICKit™ 3	MPLAB® ICD 3	MPLAB® REAL ICE™ 在线仿真器
USB速度	全速	全速/高速	全速/高速
为目标板供电	✓	✓	
硬件断点	✓	✓	✓
软件断点和跑表		✓	✓
跟踪			✓
数据捕捉			✓
逻辑/探针触发			✓
适用于...	8位	16位	32位

PICDEM™ 2 Plus 演示板

- PIC18和PIC16 DIP
- 2 x 16 LCD
- 压电蜂鸣器
- 温度传感器
- MPLAB® ICD接口
- RS-232接口
- 演示程序



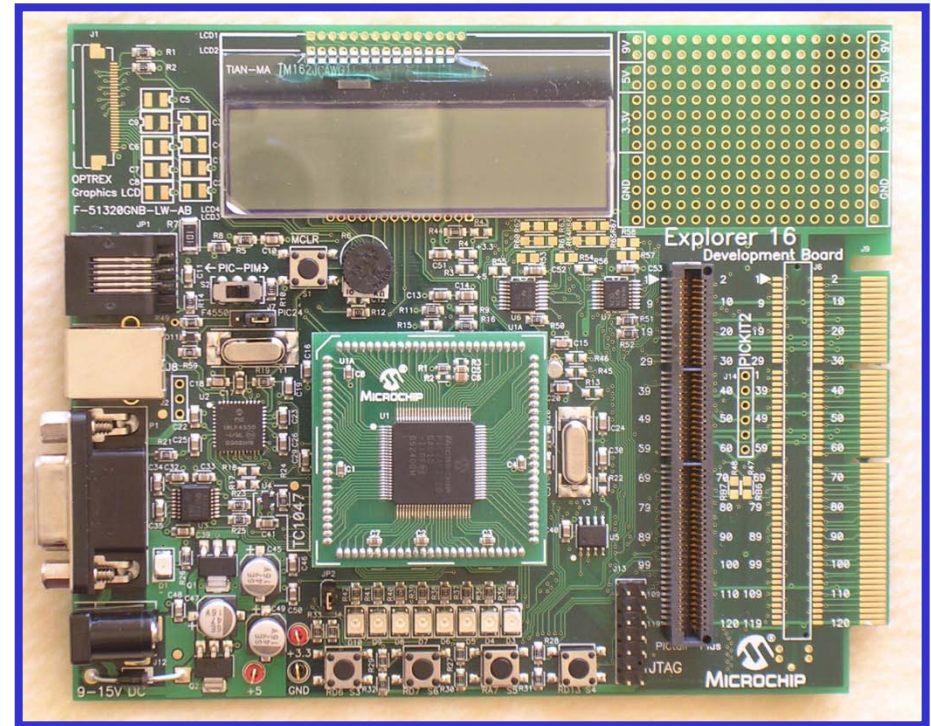
PICDEM™ 2 Plus
(DM163022)



**MPLAB ICD 2和
PICDEM 2 Plus**
(DV164006)

Explorer 16 演示板

- PIC24、dsPIC33和 PIC32MX PIM
- 2 x 16 LCD
- RS-232接口
- 温度传感器
- MPLAB® ICD接口
- PICtail™ Plus



Explorer 16 (DM240001)



MPLAB ICD 2
和 Explorer 16
(DV164033)

评估板和演示板

- PIC[®] MCU
- dsPIC[®] DSC
- PICtail[™]子卡
- 线性器件
- 接口器件
- 混合信号器件
- 存储器



Welcome to **microchip**DIRECT

We are the direct outlet of Microchip Technology Inc. Here you will find the largest available inventory of Microchip's Microcontrollers, Analog semiconductors, Serial EEPROMs, and Development Tools. We strive to provide low-risk product development, lower total system cost and faster time to market for thousands of diverse customer applications worldwide.

MPLAB® REAL ICE™
Now Supports PIC18F!



[Buy Now »](#)

Browse Our Complete Catalog

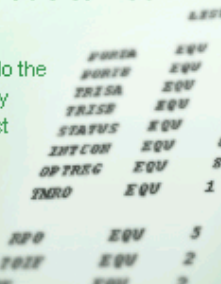
View Microchip's largest and most comprehensive inventory worldwide



[Start Shopping »](#)

Add Custom Code to Your Device

Let microchipDIRECT do the programming efficiently and with very low cost to you



[Start Here »](#)

Get Volume Pricing with a Business Account

Benefits include paying with a Purchase Order; credit lines with approval, scheduling orders, volume pricing and much more



[Apply Now »](#)

Ready to Buy?

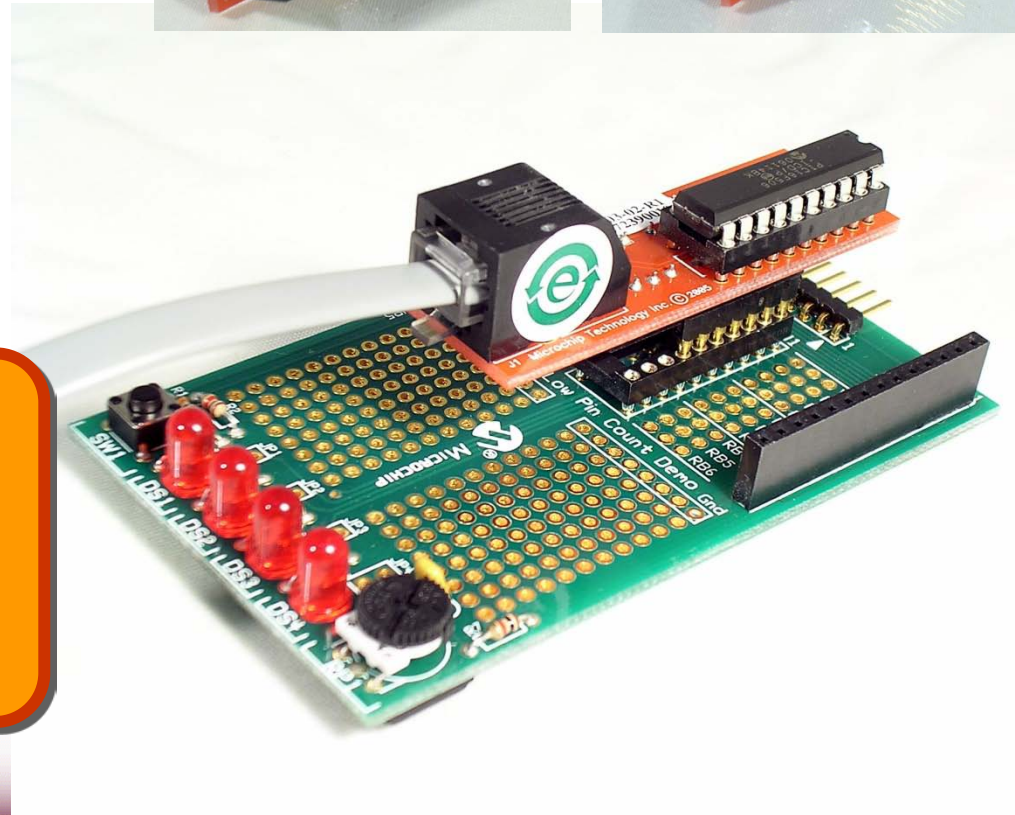
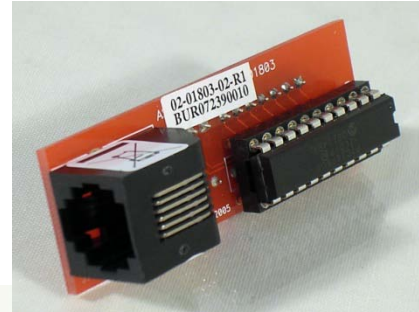


Don't have a **microchipDIRECT** Account?

[Click here to Register »](#)

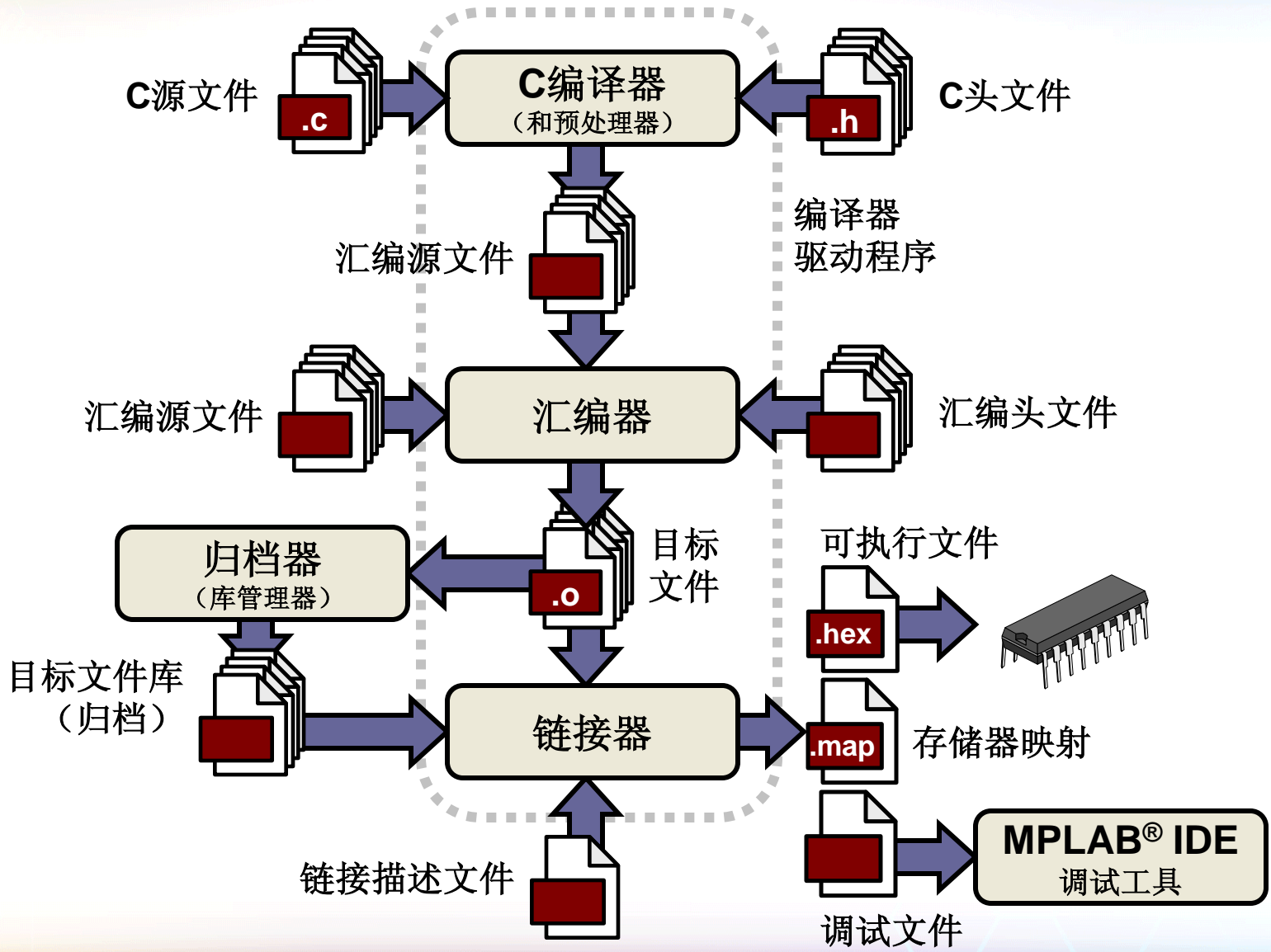
调试转接板

- 支持在以下情况下进行调试：
 - 没有足够引脚可用
 - 硅片不具备调试功能

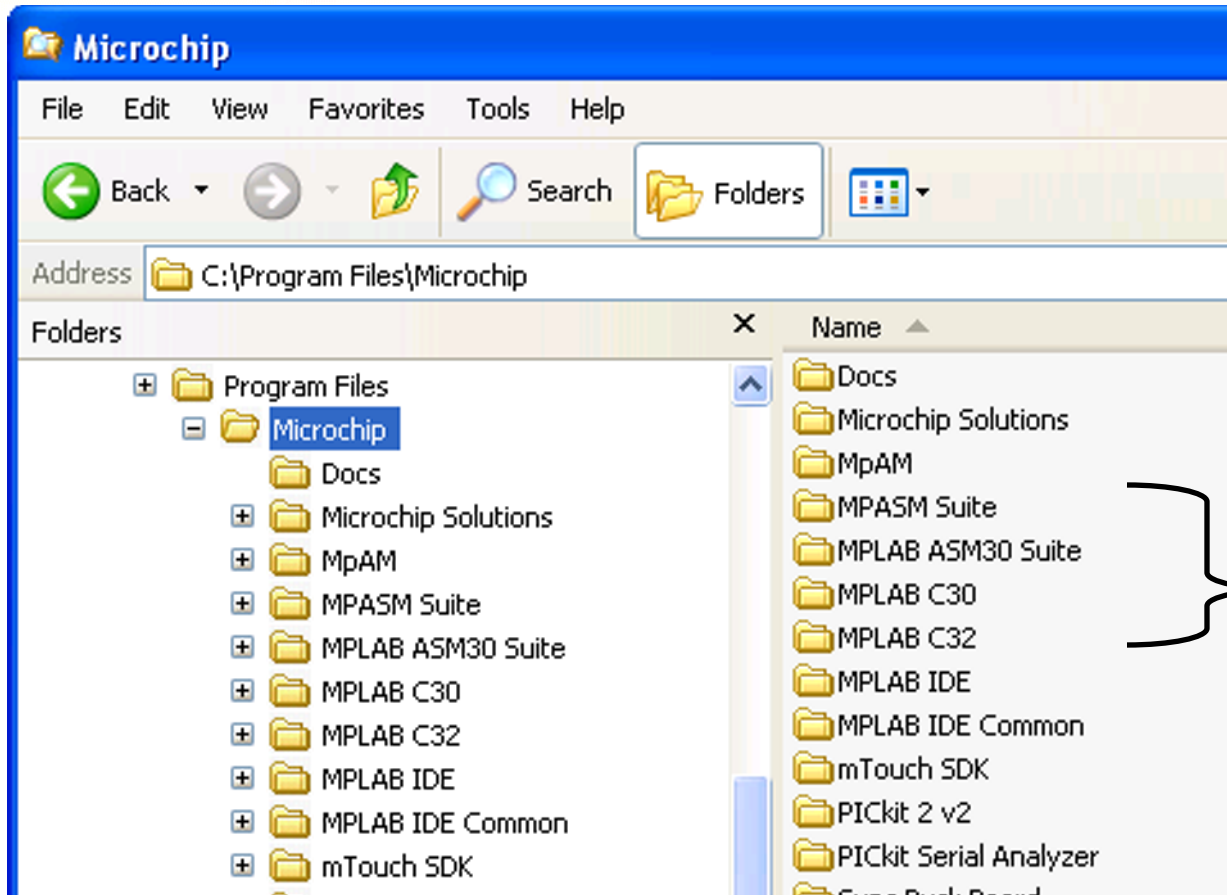


在MPLAB® IDE中
选择要使用的器件，
而不是转接板上的器件！

工具链



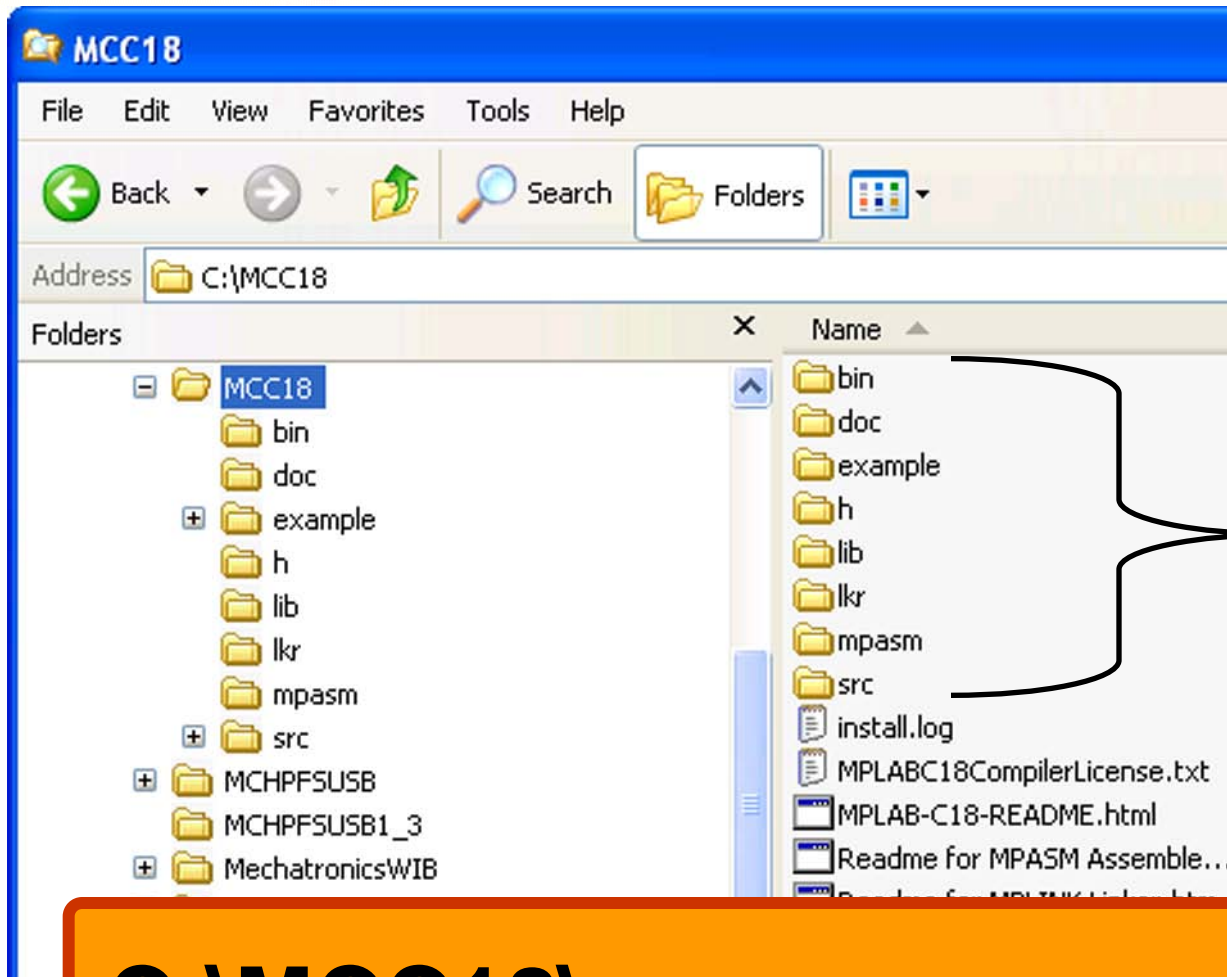
工具位置



- 汇编器
 - 8位
 - 16位
 - 32位
- C编译器
 - 16位
 - 32位

C:\Program Files\Microchip\...

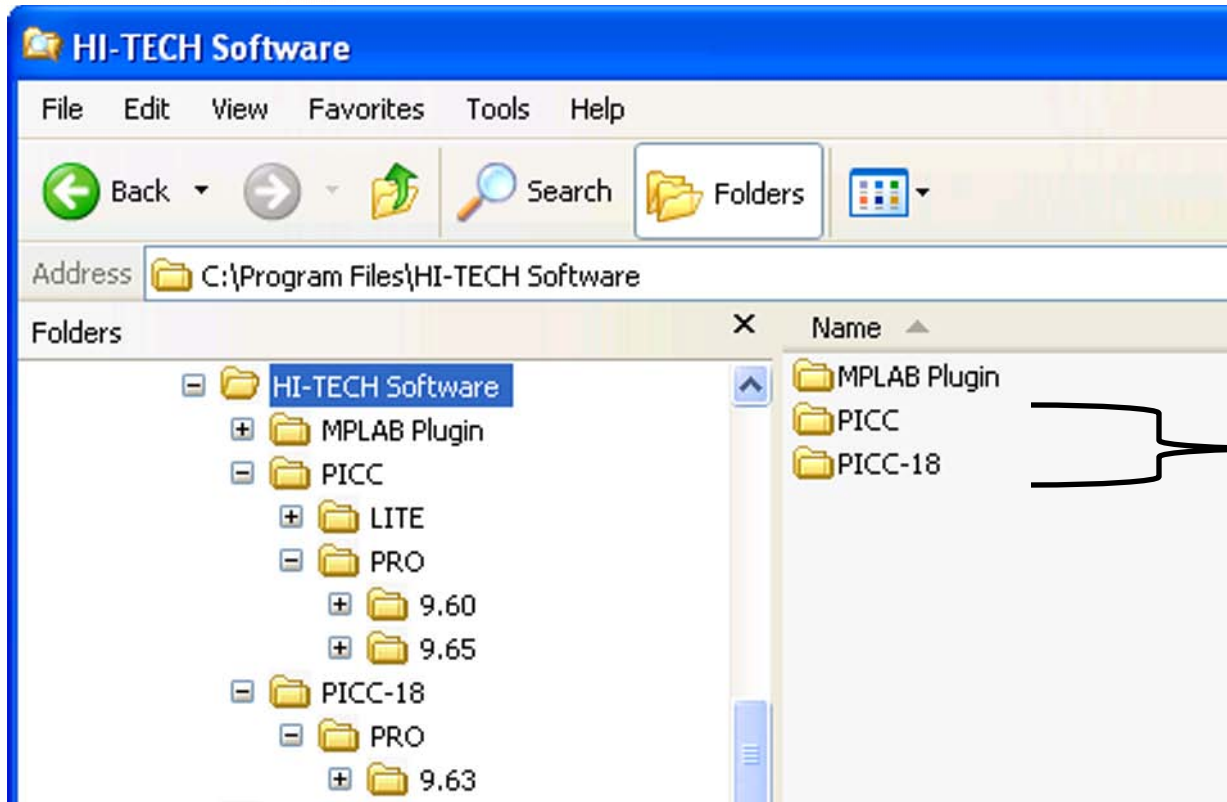
工具位置



•C编译器
•PIC18

C:\MCC18\...

工具位置

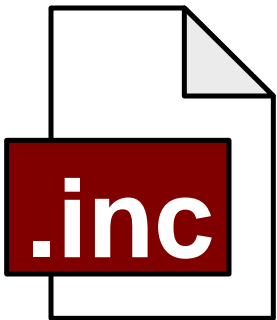
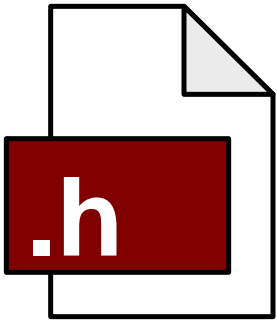


- 完整工具链
 - 8位
 - 16位
 - 32位

C:\Program Files\HI-TECH Software\...

重要文件

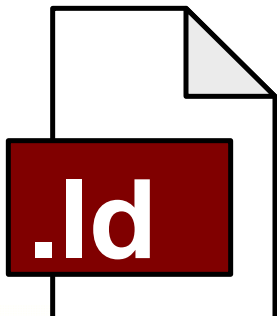
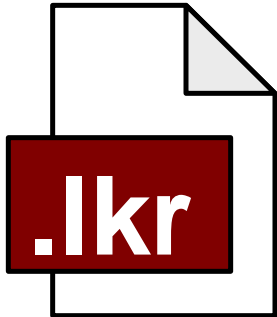
- 头文件
- 包含文件
- 内容：
 - 定义寄存器
 - 定义寄存器中的位
 - 定义配置位



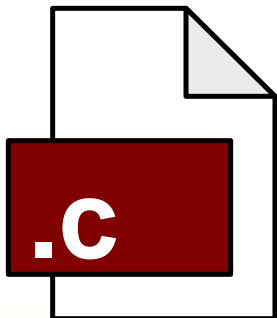
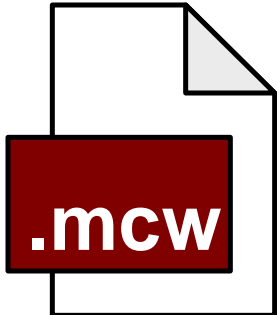
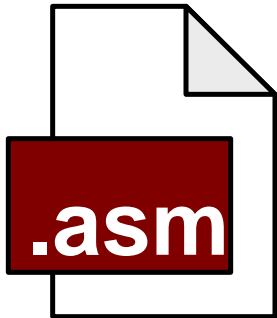
重要文件

■ 链接器文件

- 将源代码映射到**RAM**或**ROM**
- 存储类型、位置和大小
- 堆栈大小和位置
- 中断向量表
- 绝对**SFR**分配



模板文件和示例



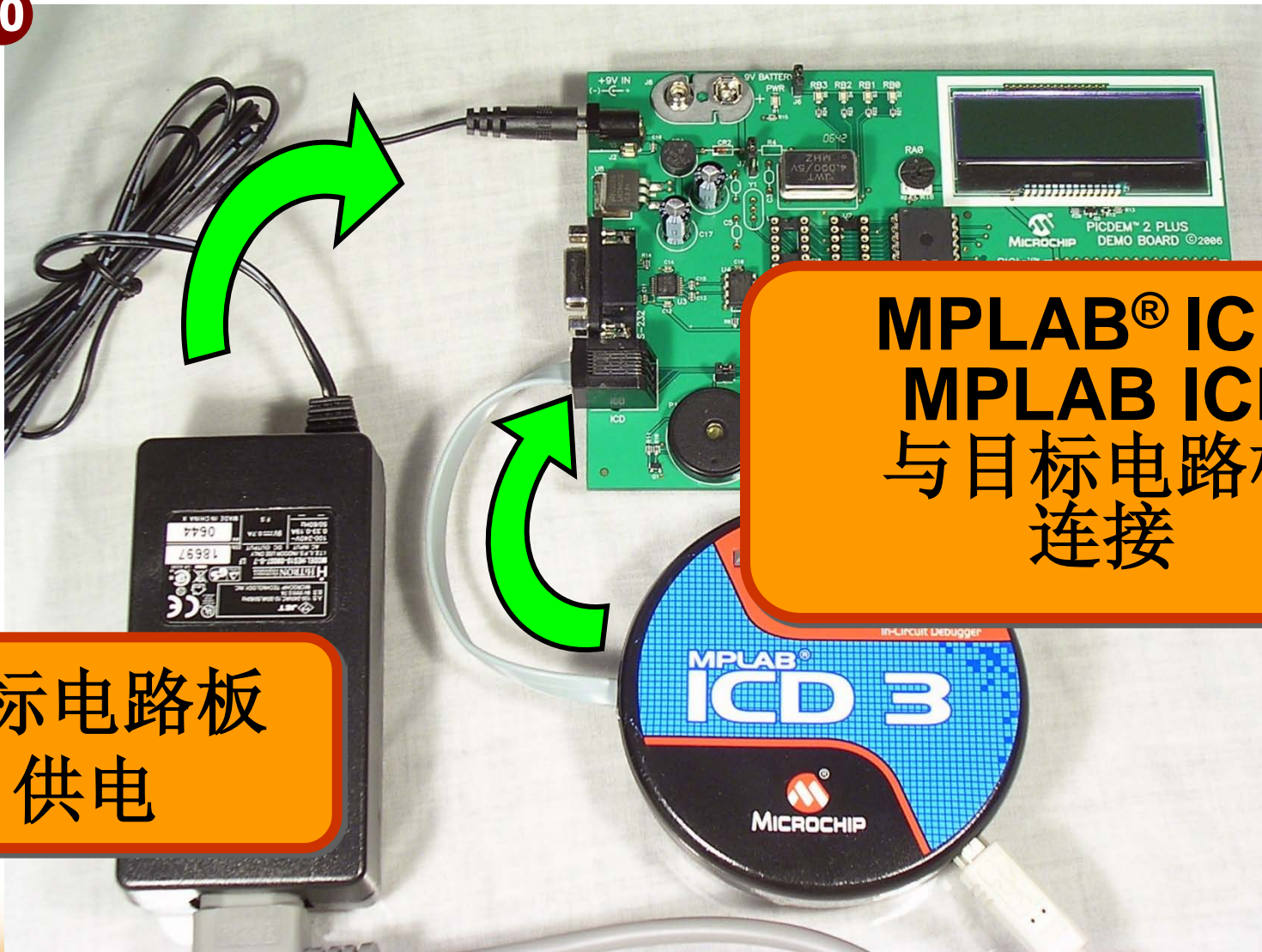
- 模板
 - 预先编写的汇编代码框架
- 示例
 - 完整的工作空间
- 范例
 - 具有一定功能的C代码示例

www.microchip.com/codeexamples



硬件设置

2 6 10



**MPLAB® ICD/
MPLAB ICE
与目标电路板
连接**

**目标电路板
供电**



MICROCHIP *2010*

MASTERs Conference

MPLAB[®] X IDE入门

迈向X之路

- **NetBeans 6.9开源IDE**
- **MASTERS 2009概念验证**
- **2009年11月Alpha测试**
- **2010年2月第一次Beta测试**
- **2010年8月公开发布RC1**

- **我们已杀入阵营！！**



范式转变

- 工作空间不再存在
- 全部对象都基于项目
 - 文件
 - 器件
 - 调试工具
 - 编译器
- 通用的可配置存储器窗口
 - 程序、数据、EEPROM和SFR...

一些新特性

- 入门指南
- 快速访问**Microchip**内容
- 访问开源功能
- 预分析器
- 调用图
- 本地历史记录

操作系统和工具

- **Windows: XP、Vista和Win7**
- **Mac 32/64位版本**
- **Linux 32位**
 - Debian、Fedora、Open Suse和Ubuntu...
- **已在Virtual Box和VMware上测试**

- **初始工具支持**
 - MPLAB[®] REAL ICE[™]在线仿真器、MPLAB ICD 3、PICKit[™] 3、MPLAB ICD 2和PM3

即将登场

- 当前**v8**中的所有功能都将在**v10**中提供
 - 模拟器外设
 - 更多**RTOS**支持
 - RoweBots
 - ThreadX
 - 完整的**DMCI**功能
 - **ICD2**中低档器件
- 独立的生产编程**GUI**

欢乐时光！！

你
希望看到什么？



MICROCHIP *2010*

MASTERs Conference

MPLAB[®] X IDE简介

简介

为什么改变一切？

- **MPLAB[®] 6-8.x**已完成使命
 - IDE功能集已显露不足
 - 难以添加许多要求的功能
 - 第三方无法方便地进行扩展
 - 缺少高级语言开发所需的基础结构

“一场小革命可以带来新气象，不是吗？”

Marko Ramius舰长
猎杀红色十月

简介

概述

- **成熟的IDE:** 经全球数以千计的开发人员多年使用验证
- **现代的IDE:** 具有所有最新功能
- **跨平台兼容性**
- **文档化、可扩展的架构**
 - 它是开源软件!
- 许多用户要求的功能已经存在
- 架构完全重新设计, 支持高级语言开发

简介

概述

- 功能丰富而不臃肿的**IDE**
- 由丰富插件组成的生态系统
- 可用于嵌入式和**PC**软件开发的**IDE**
- 可无需**IDE**的项目文件结构（适合外部工具使用）
- 可跟踪更改的**IDE**
- 功能强大的项目导航系统

简介

为什么不像其他人一样使用Eclipse?

- 因为这些其他人拖累了一个指标性的好产品 😊
- 有许多人认为**NetBeans**:
 - 更先进
 - 相对不臃肿
 - 更快速
 - 更直观 (**GUI**更易用)
- 我们是**NetBeans**在嵌入式开发领域的唯一合作伙伴

简介

为什么不像其他人一样使用**Eclipse**?

- 关于真实案例，请访问：

- <http://netbeans.org/switch/realstories.html>

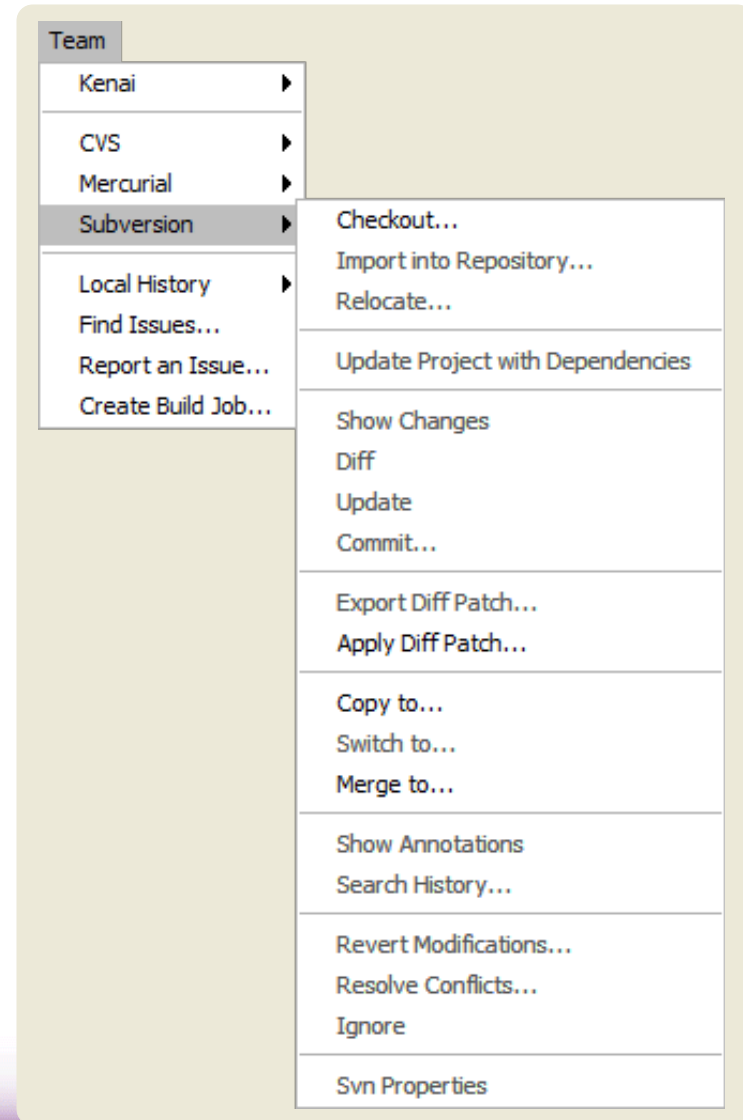
- 语法彩色高亮显示
- 即时分析功能，可在用户输入时标出错误
- 智能代码补全功能，可在用户输入时提供建议和提示
- 文件历史记录，无需版本控制系统也可查看最近更改和进行恢复
- 功能强大的导航工具：**Go to file**（转至文件）、**Go to type**（转至类型）、**Go to symbol**（转至符号）、**Go to header**（转至头文件）和**Go to declaration**（转至声明）...
- 重构工具，可智能地重构代码

MPLAB[®] X

主要特性

- 可基于用户定义的规则自动进行代码格式化
- 内置**Diff**功能（也可以使用外部功能）
- 任务导航功能，可查找标有特殊注释（由用户定义）的代码——例如，**//TODO**和**//FIXME**
- 可完全自定义的编辑器：工具栏、工具栏按钮、窗口和窗口位置...

- 内置支持版本控制系统：
 - Subversion、CVS和Mercurial——其他系统通过插件支持
- 内置支持Bugzilla问题跟踪系统
 - 与编辑器集成，方便进行问题导航

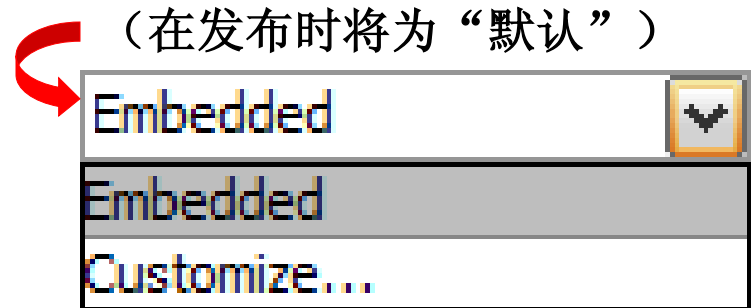
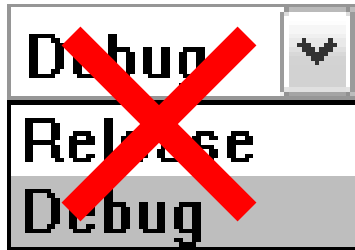


MPLAB[®] X

项目和工作空间

- 工作空间已不再存在
- 项目包含所有相关信息
- 在打开**MPLAB X**时，将会装入最近打开的项目
- 在打开一个项目时，大部分窗口/选项卡会恢复为它们原先的布局
- 多个项目可使用 **Project Groups**（项目组）处理

- 令人畏惧的**Debug/Release**（调试/发布）组合框已不再存在！

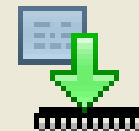


- 对于调试和发布，可使用不同工具栏按钮来完成编译和烧写

编译并烧写（调试）



编译并烧写（发布）



MPLAB[®] X

从MPLAB 8升级

- 内置了一个项目转换实用程序
 - MPLAB 8项目不能直接打开
 - 项目文件/结构完全不同
 - 手动转换很简单，但耗费时间
- **MPLAB X使用WinUSB/LibUSB驱动程序，而非MPLAB 8中的自定义驱动程序**
 - 8和X可以同时安装
 - 必须切换硬件工具驱动程序
 - 提供了切换实用程序
- 编译器/汇编器不受影响
 - MPASM[™]汇编器例外



MICROCHIP 2010

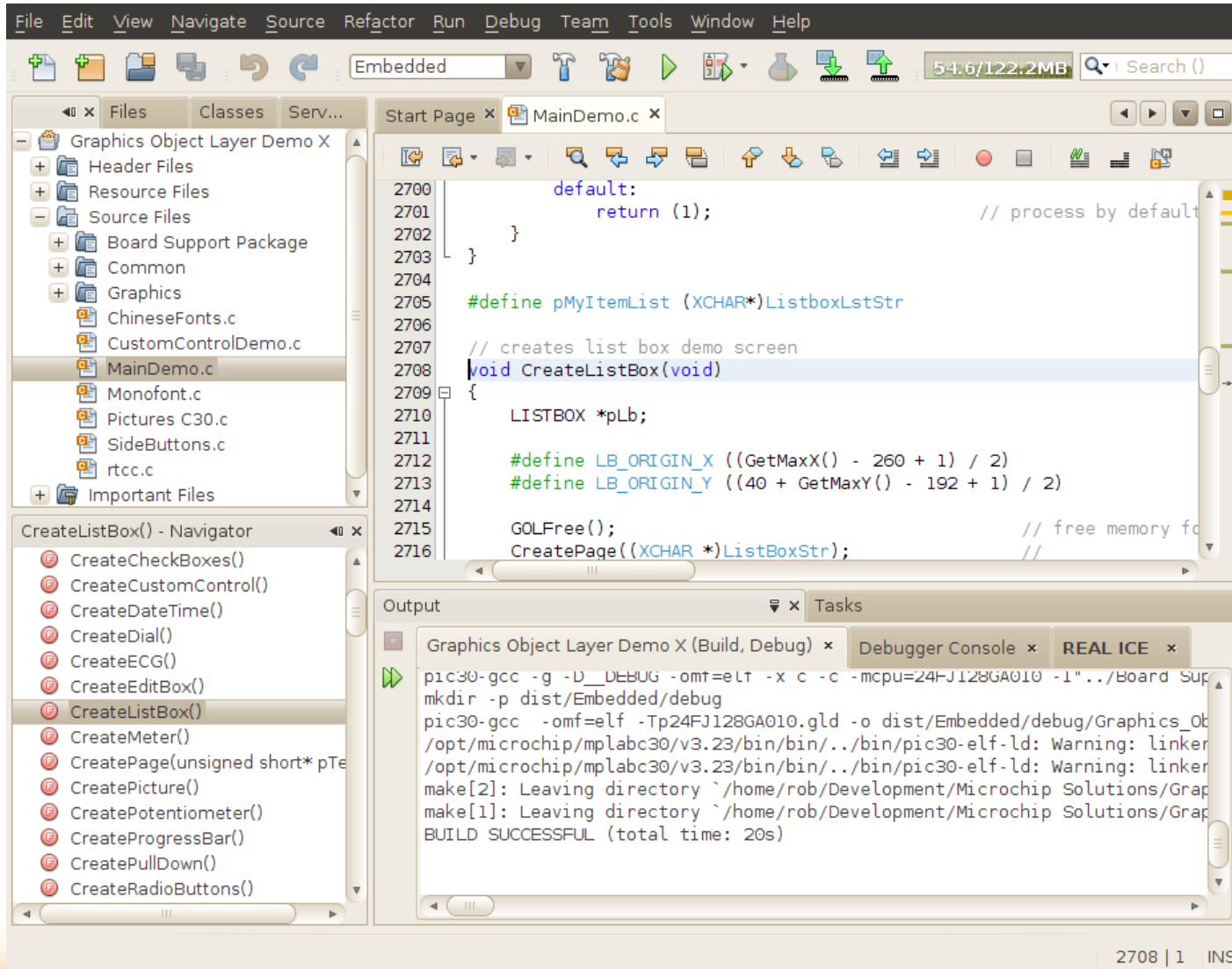
MASTERS Conference

MPLAB[®] X IDE概览

IDE的组成部分

MPLAB® X IDE概览

IDE的组成部分

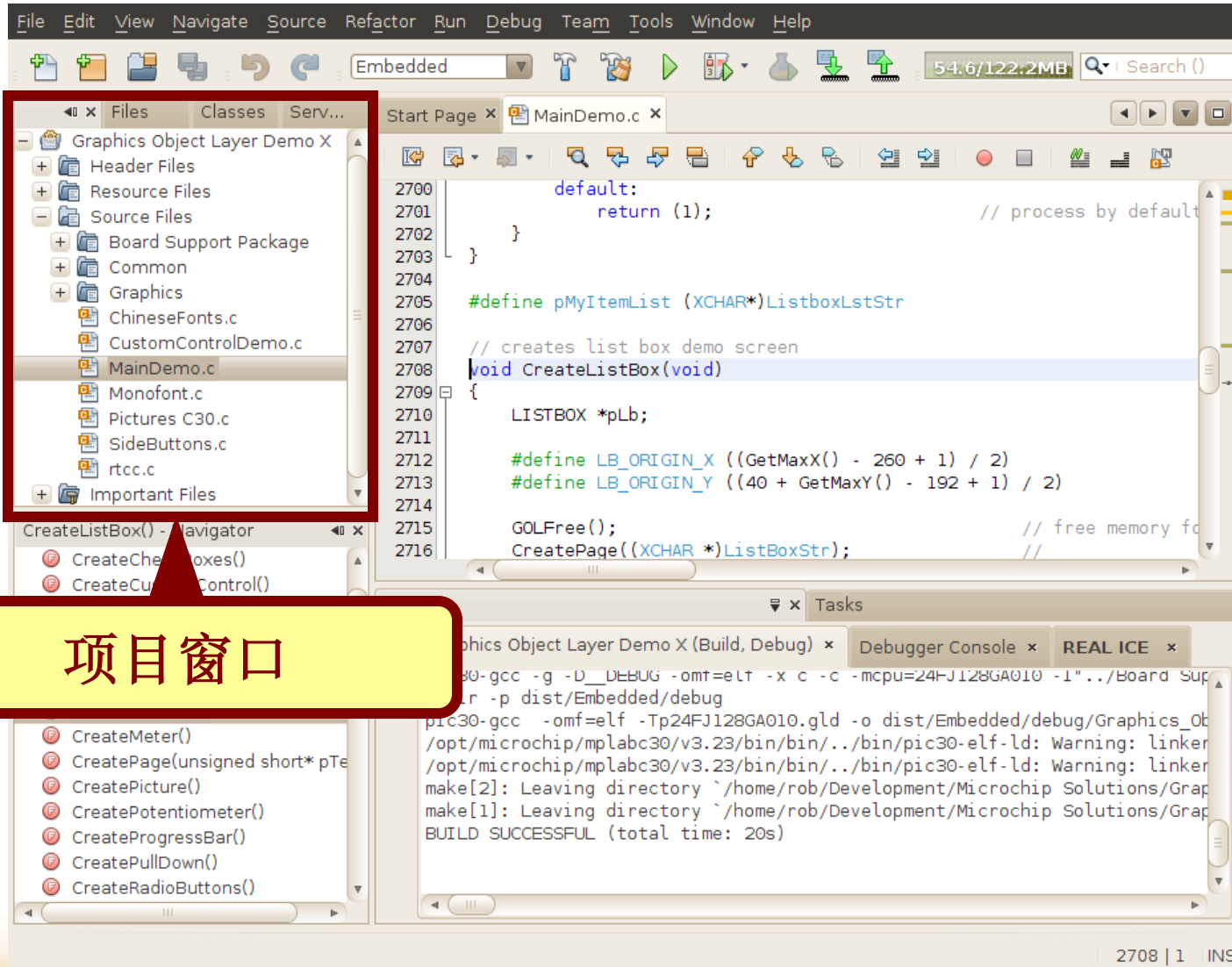


The screenshot displays the MPLAB X IDE interface with the following components:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Team, Tools, Window, Help.
- Toolbar:** Includes icons for file operations, search, and execution. The status bar shows "Embedded" and "54.6/122.2MB".
- Project Explorer (Left):** Shows a tree view of the project "Graphics Object Layer Demo X" with folders for Header Files, Resource Files, Source Files, Board Support Package, Common, Graphics, and Important Files. The file "MainDemo.c" is selected.
- Code Editor (Center):** Displays the source code for "MainDemo.c". The code includes a default return function, a macro definition for a list box, and a function "void CreateListBox(void)" that initializes a list box and calls "CreatePage".
- Navigator (Bottom Left):** Lists various GUI widget creation functions such as "CreateCheckBoxes()", "CreateCustomControl()", "CreateDateTime()", "CreateDial()", "CreateECG()", "CreateEditBox()", "CreateListBox()", "CreateMeter()", "CreatePage()", "CreatePicture()", "CreatePotentiometer()", "CreateProgressBar()", "CreatePullDown()", and "CreateRadioButtons()". "CreateListBox()" is currently selected.
- Output Window (Bottom Right):** Shows the build output for "Graphics Object Layer Demo X (Build, Debug)". The output includes compiler and linker commands, warnings, and a final "BUILD SUCCESSFUL (total time: 20s)" message.

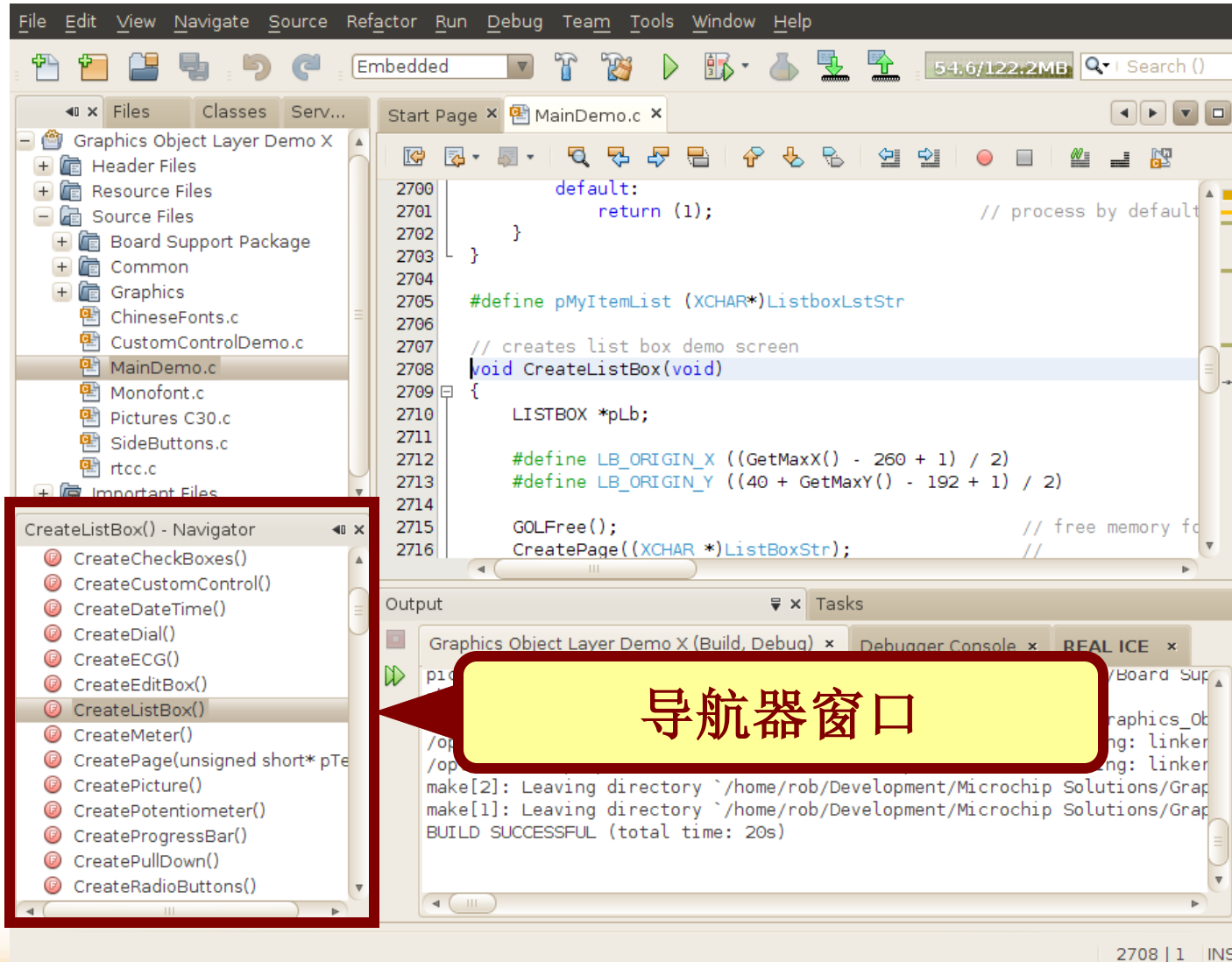
MPLAB[®] X IDE概览

IDE的组成部分



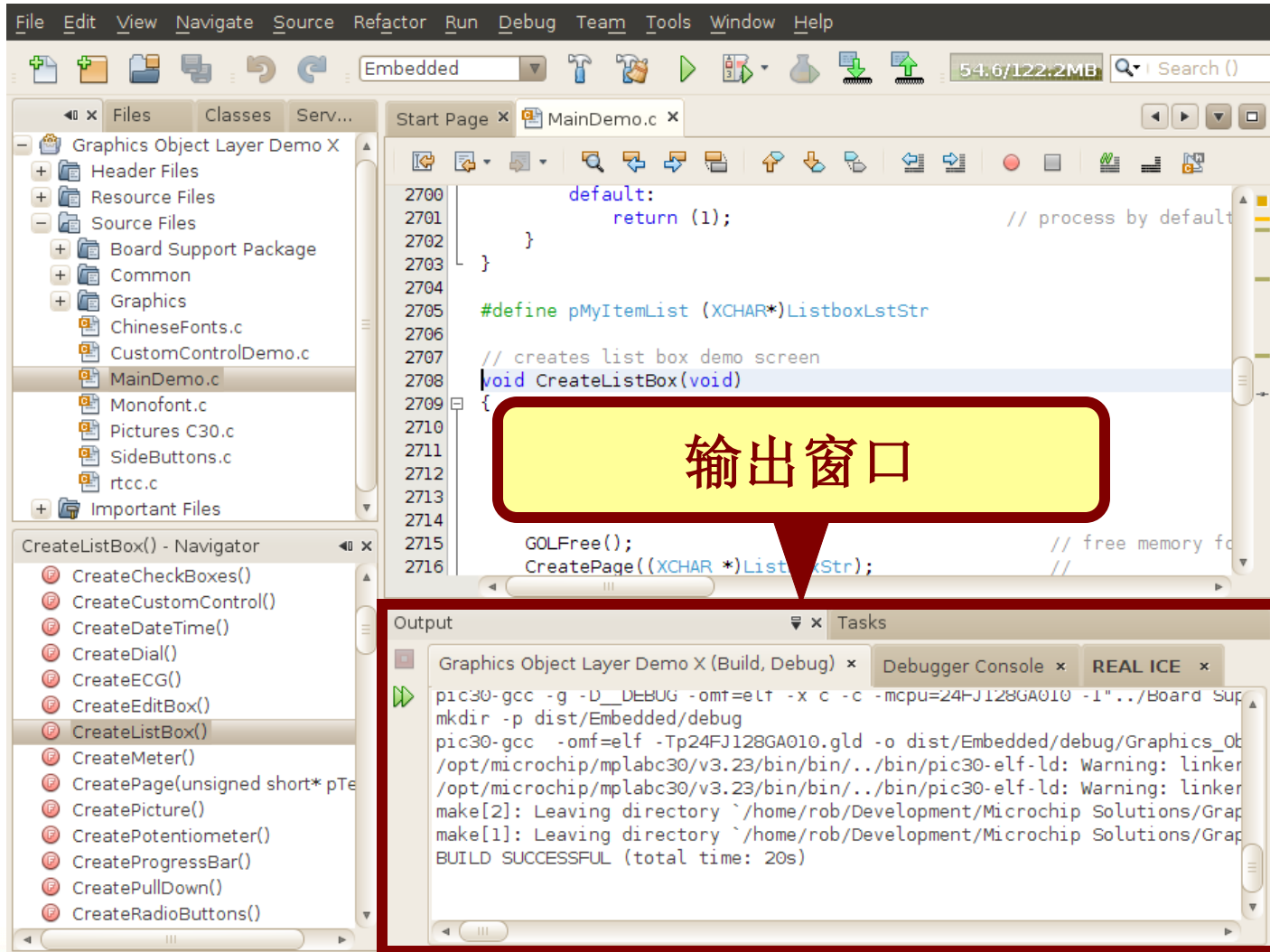
MPLAB[®] X IDE概览

IDE的组成部分



MPLAB® X IDE概览

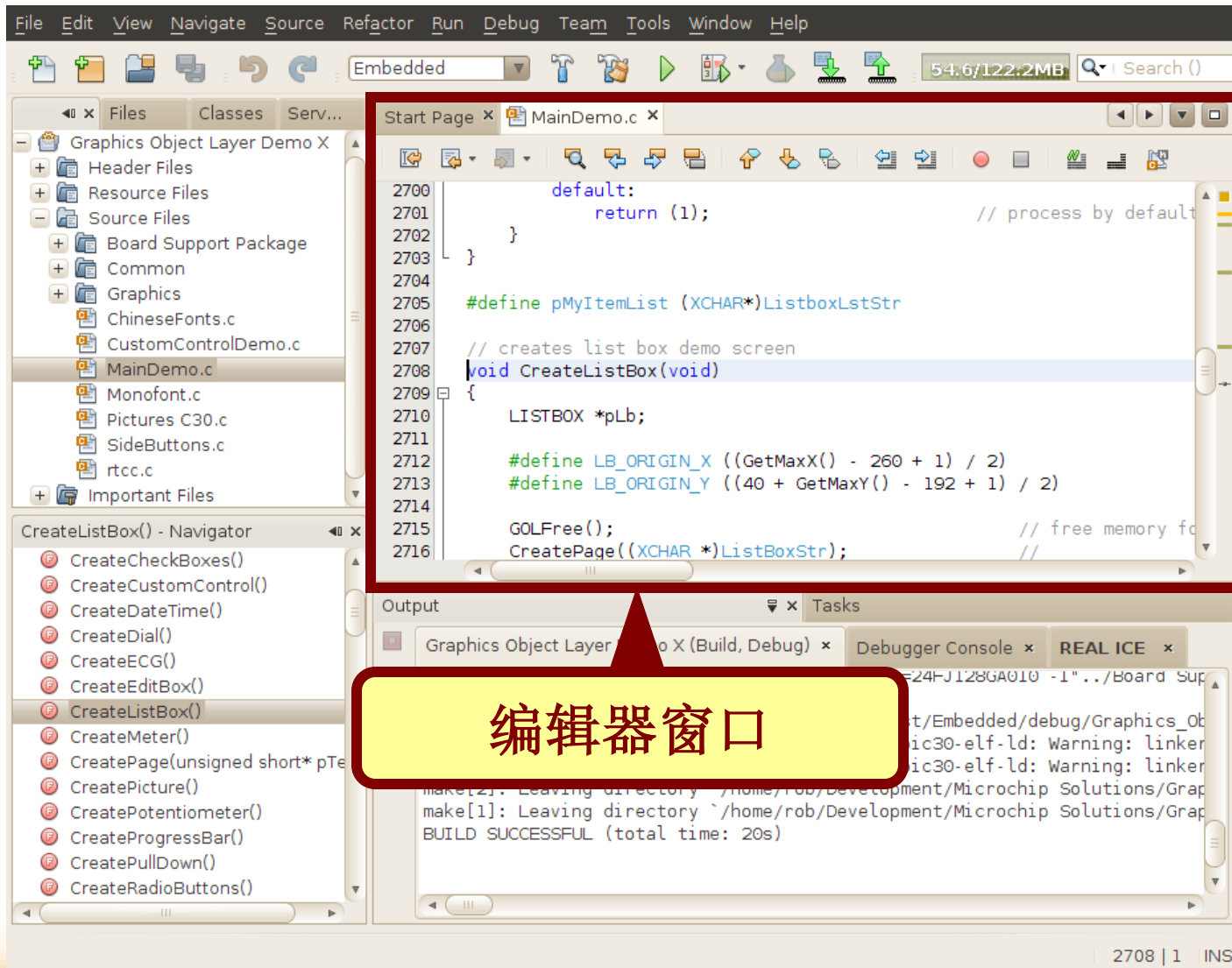
IDE的组成部分



The screenshot displays the MPLAB X IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and execution. The main editor window shows the source code for 'MainDemo.c', with a yellow callout box labeled '输出窗口' (Output Window) pointing to the 'void CreateListBox(void)' function. The left sidebar contains a 'Files' view showing the project structure, including 'Graphics Object Layer Demo X' and 'MainDemo.c'. Below the files view is a 'Navigator' window showing a list of functions, with 'CreateListBox()' selected. At the bottom, the 'Output' window displays the build output, including the command 'pic30-gcc -g -D __DEBUG -omt=elf -x c -c -mcpu=24-FJ128GA010 -I' and the message 'BUILD SUCCESSFUL (total time: 20s)'. The status bar at the bottom right shows '2708 | 1 | INS'.

MPLAB[®] X IDE概览

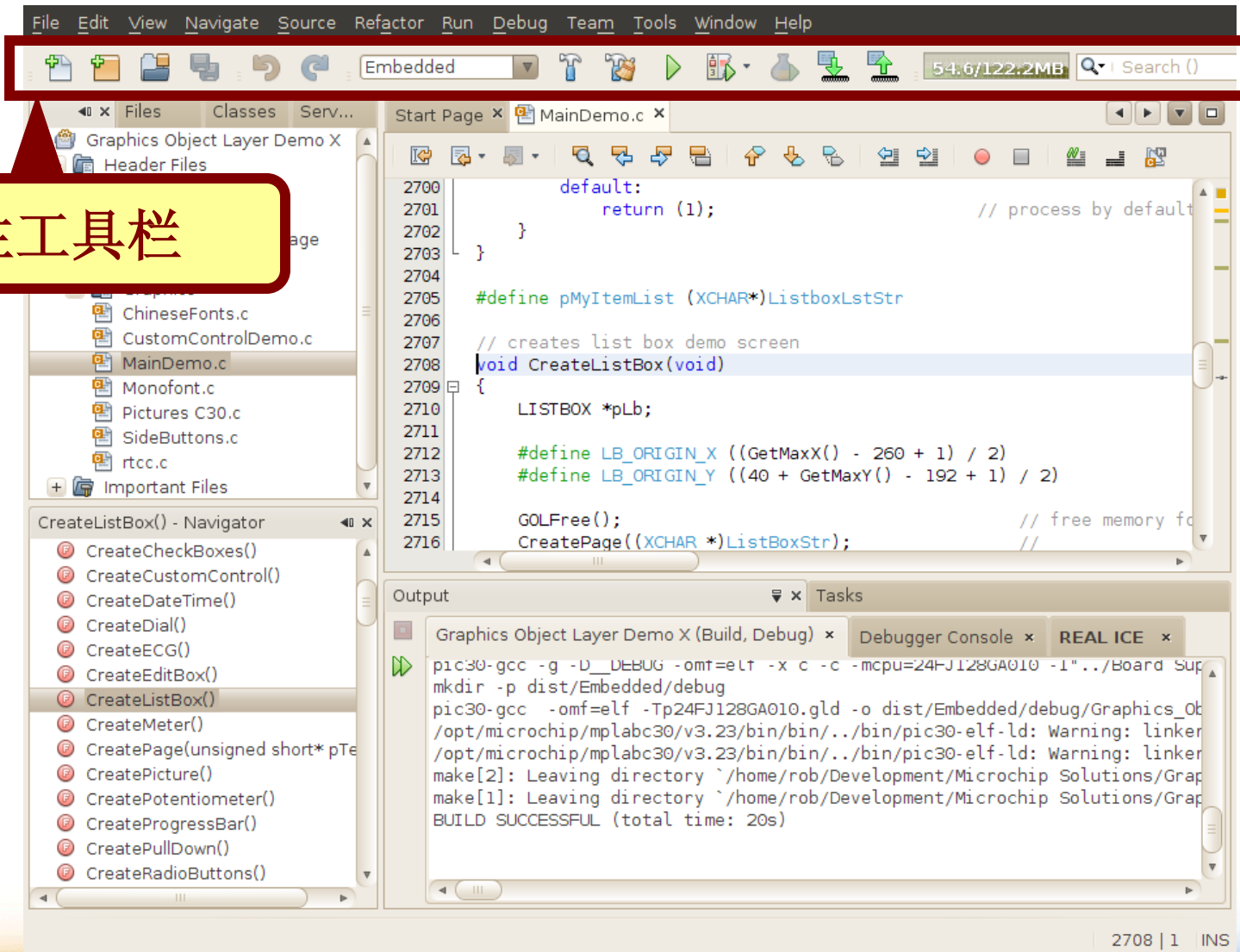
IDE的组成部分



MPLAB[®] X IDE概览

IDE的组成部分

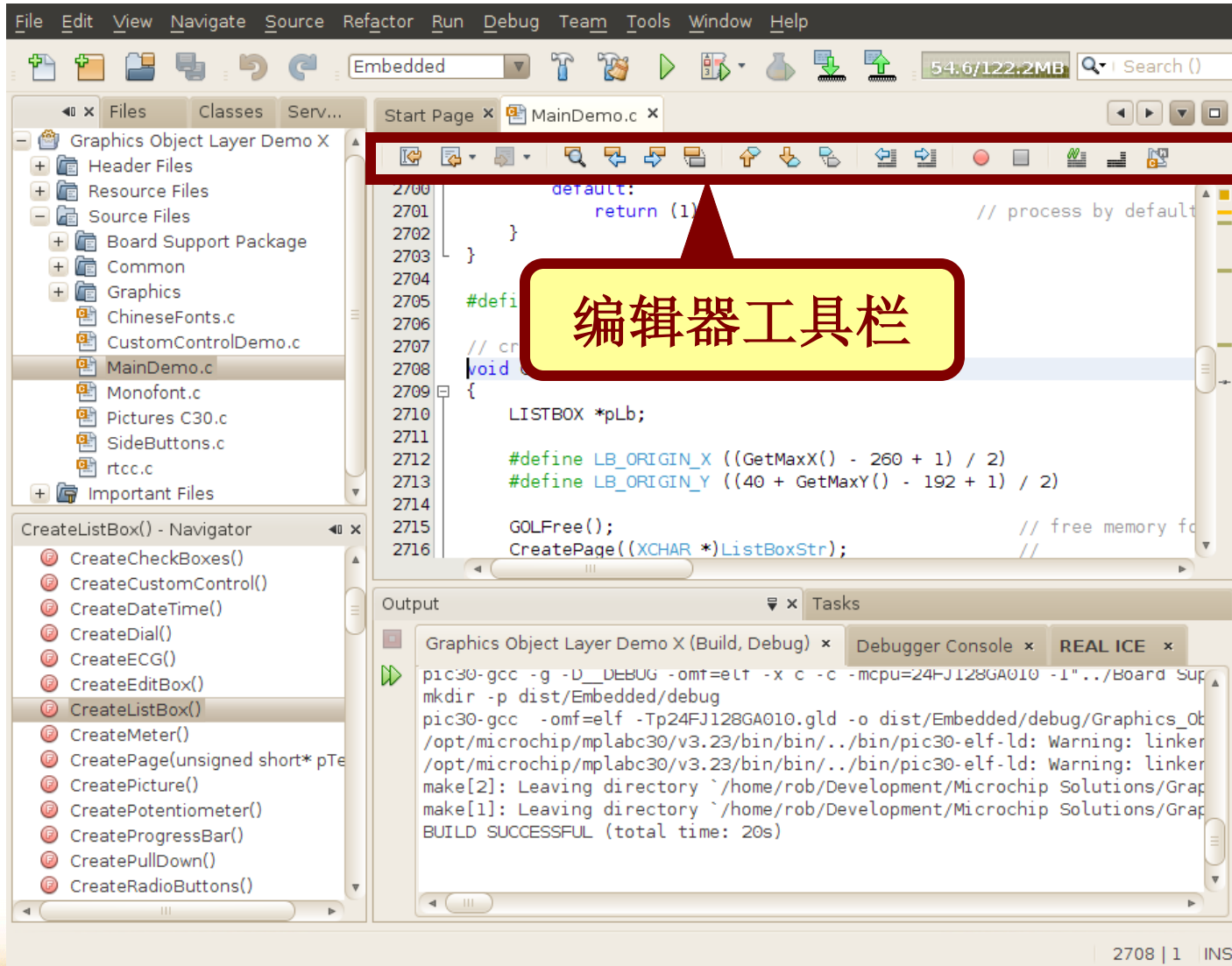
主工具栏



The screenshot displays the MPLAB X IDE interface. At the top, a menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Team, Tools, Window, and Help. Below the menu bar is a toolbar with various icons for file operations, navigation, and execution. A red box highlights this toolbar area, with a callout box containing the text '主工具栏' (Main Toolbar). The main workspace is divided into several panes: a Project Explorer on the left showing a file tree with 'MainDemo.c' selected; a central Editor window displaying C code for 'MainDemo.c' with line numbers 2700-2716; a Navigator pane at the bottom left listing various widget creation functions like 'CreateCheckBoxes()' and 'CreateListBox()'; and an Output pane at the bottom right showing the results of a build process, including compiler and linker commands and the message 'BUILD SUCCESSFUL (total time: 20s)'. The status bar at the bottom right shows '2708 | 1 | INS'.

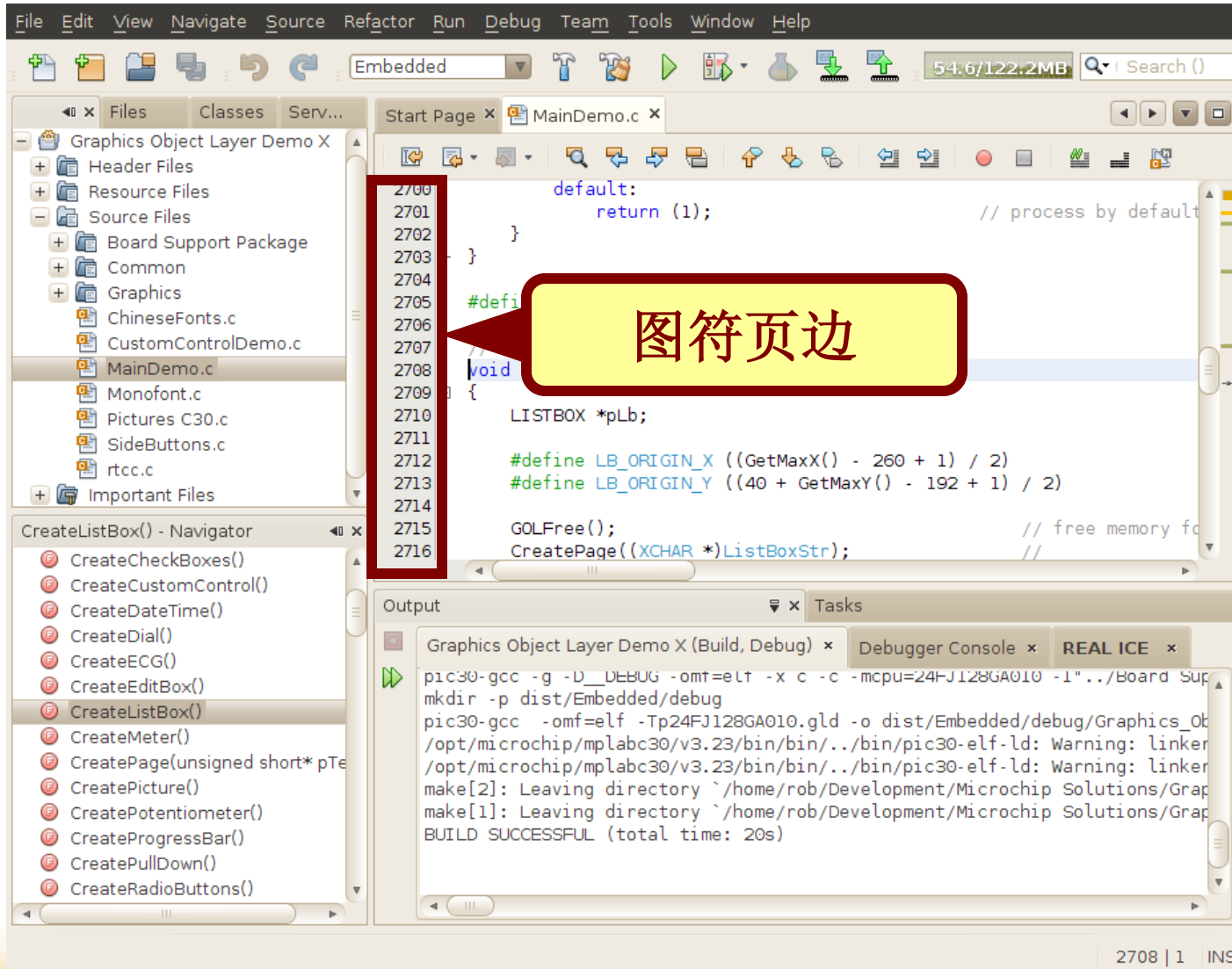
MPLAB[®] X IDE概览

IDE的组成部分



MPLAB[®] X IDE概览

IDE的组成部分



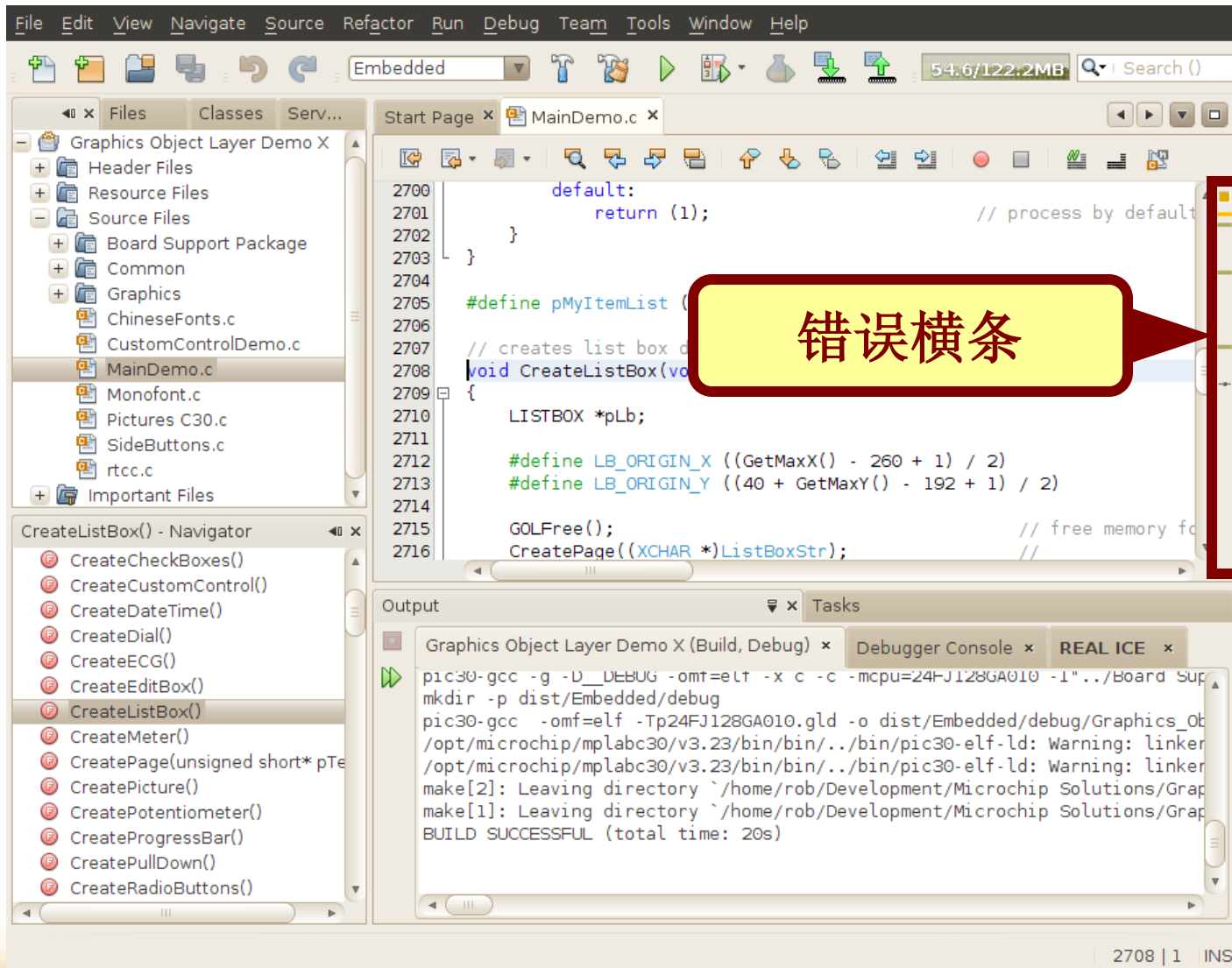
The screenshot displays the MPLAB X IDE interface. The main window shows the code editor for 'MainDemo.c' with the following code:

```
2700     default:
2701         return (1);           // process by default
2702     }
2703 }
2704
2705 #define
2706
2707 void
2708 {
2709     LISTBOX *pLb;
2710
2711     #define LB_ORIGIN_X ((GetMaxX() - 260 + 1) / 2)
2712     #define LB_ORIGIN_Y ((40 + GetMaxY() - 192 + 1) / 2)
2713
2714     GOLFree();                // free memory for
2715     CreatePage((XCHAR *)ListBoxStr); //
2716 }
```

A red callout box with the text '图符页边' (Symbol Margin) points to the left margin of the code editor, which is highlighted with a red box. The left margin contains line numbers 2700 through 2716. The IDE also shows a file explorer on the left, a navigator window with a list of functions, and an output window at the bottom showing the build process.

MPLAB[®] X IDE概览

IDE的组成部分



The screenshot displays the MPLAB X IDE interface. On the left, the 'Files' pane shows a project tree for 'Graphics Object Layer Demo X'. The central editor shows the source code for 'MainDemo.c', with a yellow callout box containing the text '错误横条' (Error bar) pointing to a horizontal error bar in the code editor. The bottom pane shows the 'Output' window with the following text:

```
Graphics Object Layer Demo X (Build, Debug) x Debugger Console x REAL ICE x
pic30-gcc -g -D__DEBUG -omt=elt -x c -c -mcpu=24-FJ128GA010 -I"../Board Supp
mkdir -p dist/Embedded/debug
pic30-gcc -omf=elf -Tp24FJ128GA010.gld -o dist/Embedded/debug/Graphics_Ob
/opt/microchip/mplabc30/v3.23/bin/bin/./bin/pic30-elf-ld: Warning: linker
/opt/microchip/mplabc30/v3.23/bin/bin/./bin/pic30-elf-ld: Warning: linker
make[2]: Leaving directory `/home/rob/Development/Microchip Solutions/Grap
make[1]: Leaving directory `/home/rob/Development/Microchip Solutions/Grap
BUILD SUCCESSFUL (total time: 20s)
```



MICROCHIP 2010

MASTERS Conference



实验1

如何创建新项目

讲师指导的动手练习

实验1

创建新项目



目标

- 使用项目向导创建新项目
- 新建C源文件
- 将用于控制**Explorer 16 LCD**的现有文件添加到项目中
- 配置C编译器工具链
- 编译代码并在目标电路板上进行调试



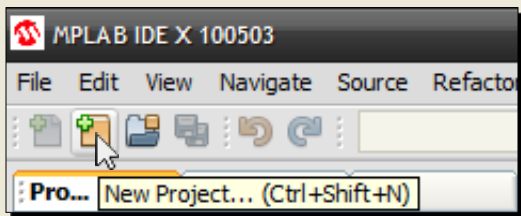
资源需求： 带有PIC24FJ128GA010的Explorer 16演示板

实验1

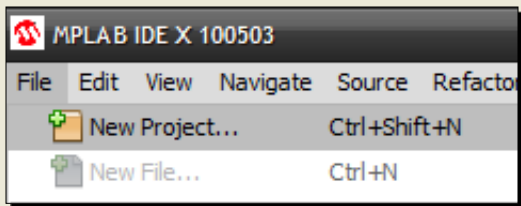
创建新项目



1 启动新建项目向导



在工具栏中单击**New Project**（新建项目）图标 ，或从菜单中选择**File**（文件）▶ **New Project**



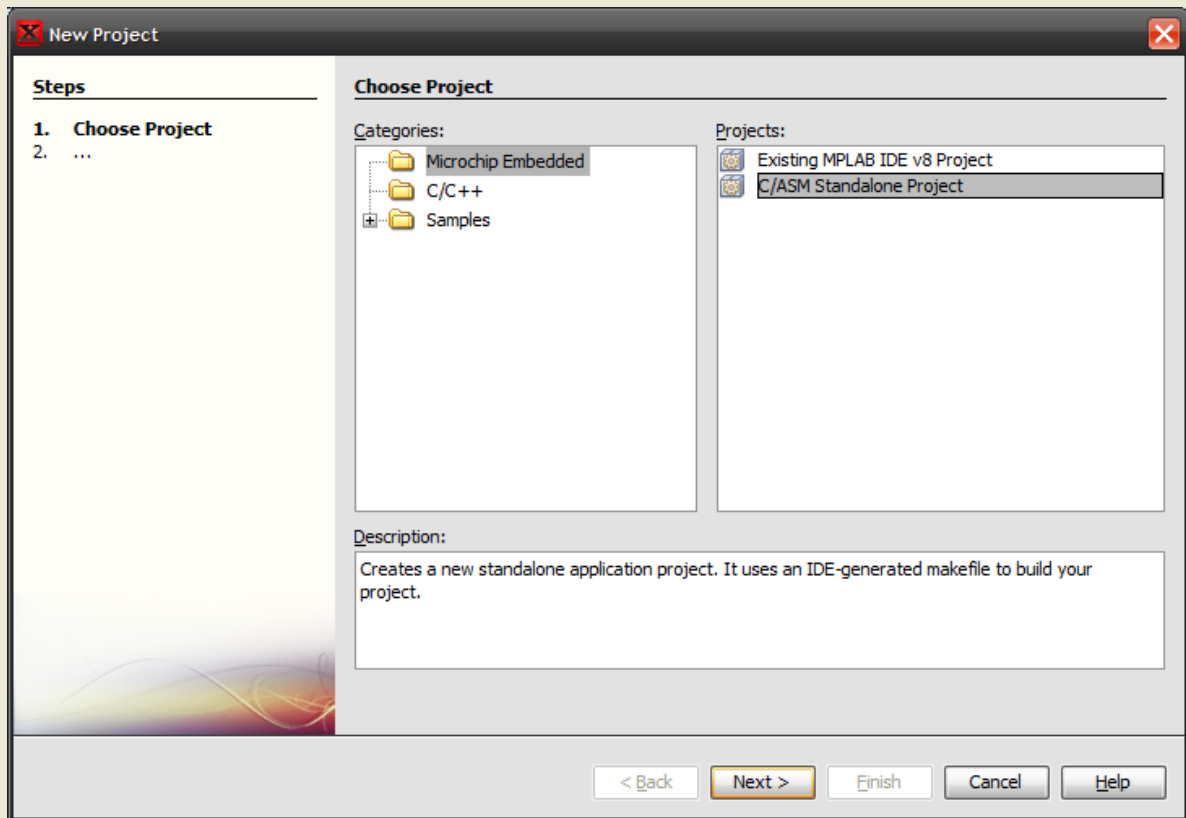
不同于MPLAB® 8，在MPLAB X中必须使用向导，因为它会同时创建项目目录本身和正确工作所需的完整子目录层级结构。手动创建项目是不可能的。

实验1

创建新项目



2 选择项目



a 在**Categories**（类别）下选择**Microchip Embedded**（Microchip嵌入式）

b 在**Projects**（项目）下选择**C/ASM Standalone Project**（C/ASM独立项目）

单击

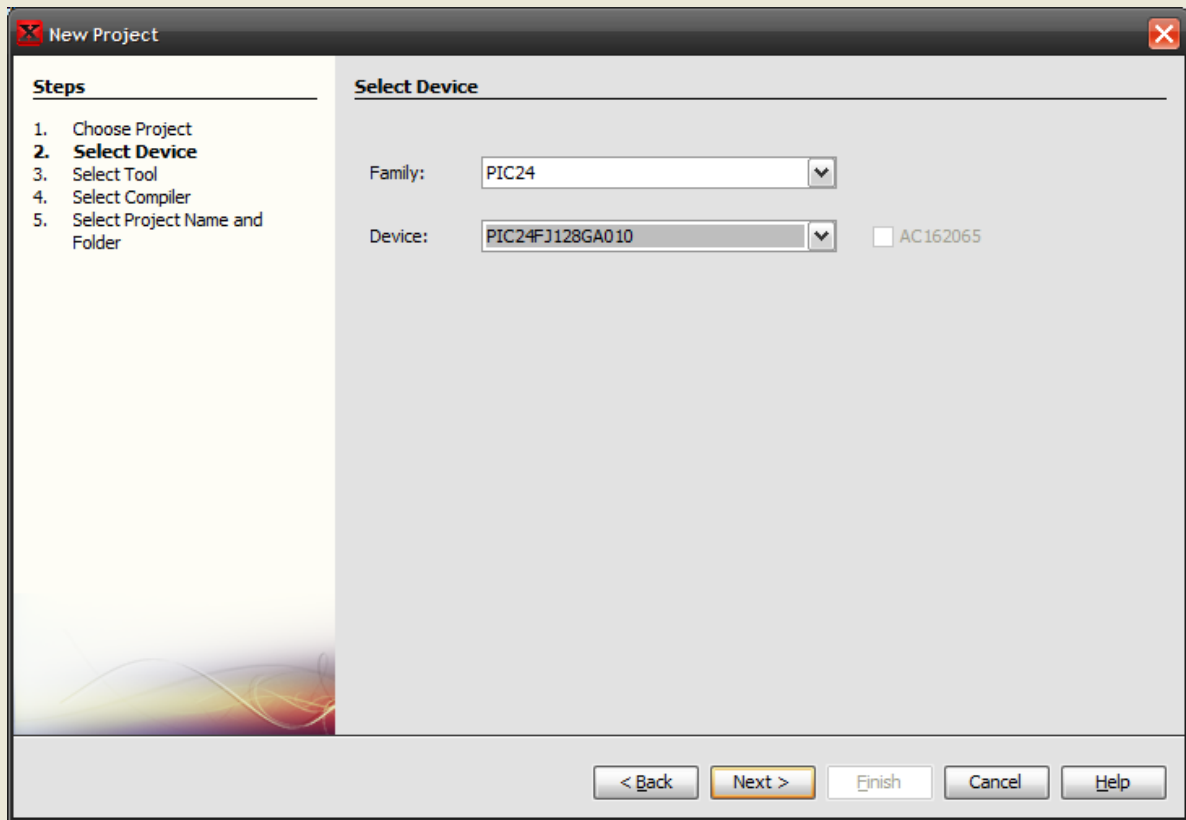
Next >

实验1

创建新项目



3 选择器件



选择

PIC24FJ128GA010

Family（系列）组合框会对**Device**（器件）组合框中的器件列表进行筛选，以方便查找特定器件。

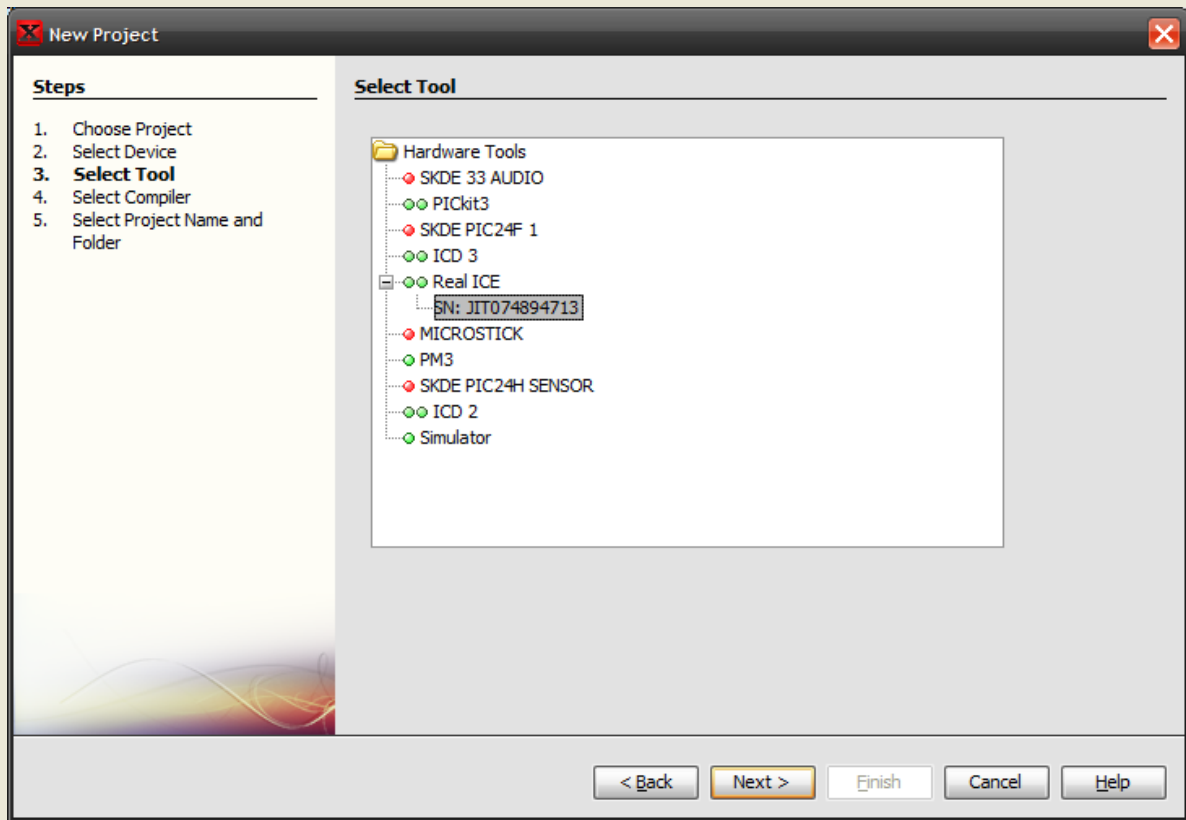
单击

Next >



4

选择工具



选择Real ICE

选择对应于希望使用的MPLAB® REAL ICE™在线仿真器的序列号。如果有多个REAL ICE设备与PC连接，则会看到多个序列号。

单击

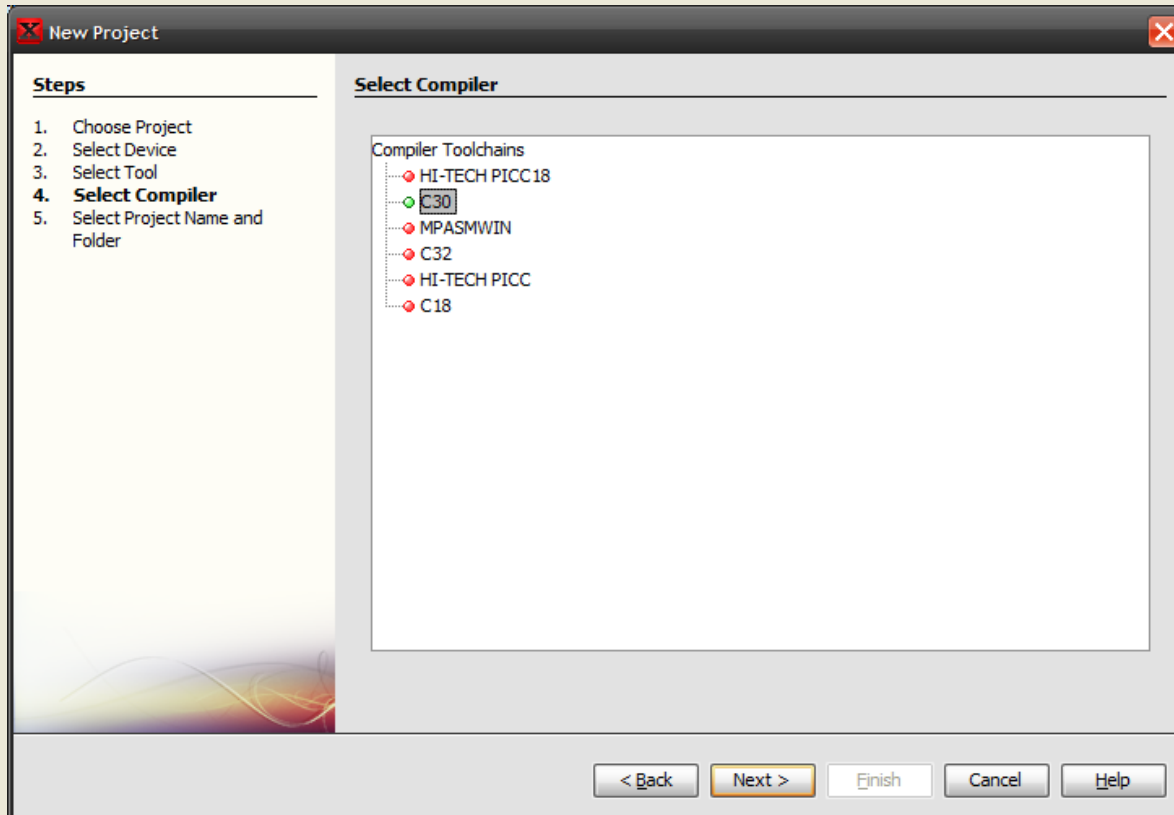
Next >

实验1

创建新项目



5 选择编译器



选择**C30**
编译器工具链

单击

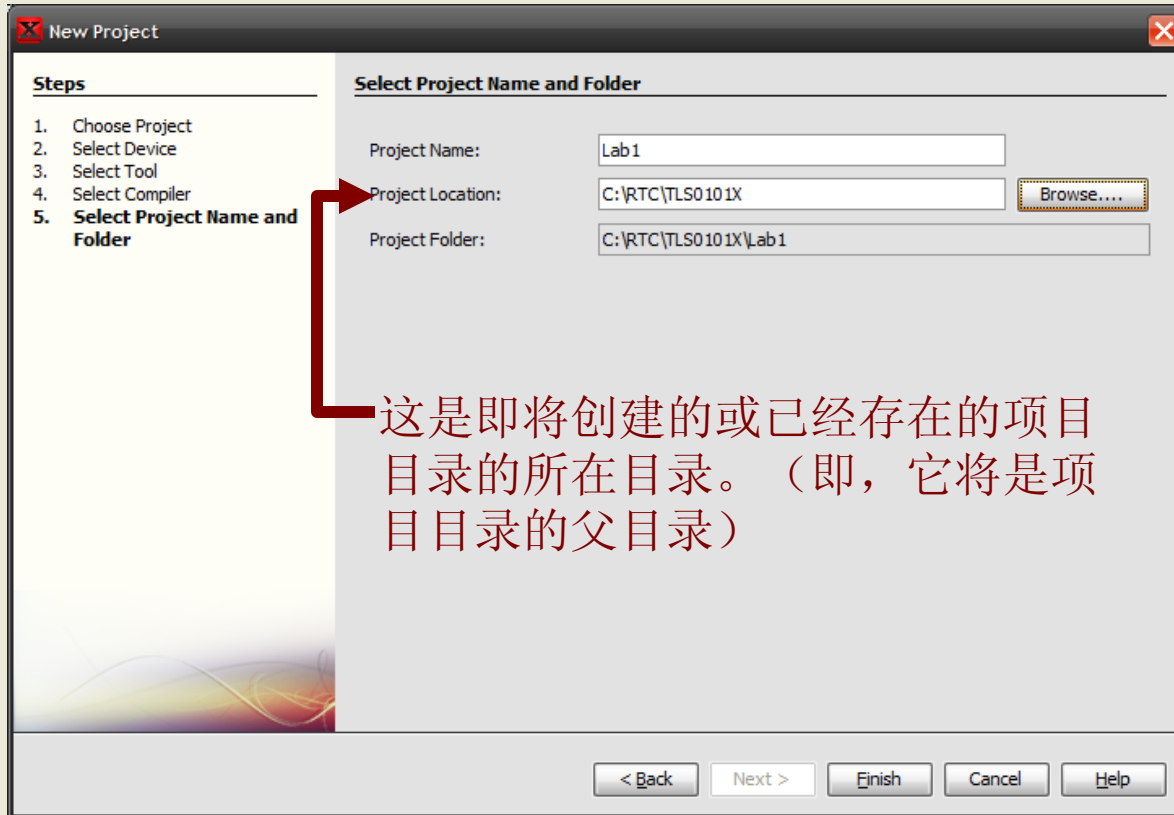
Next >

实验1

创建新项目



6 选择项目名称和文件夹



命名项目，并选择项目目录的创建位置

单击

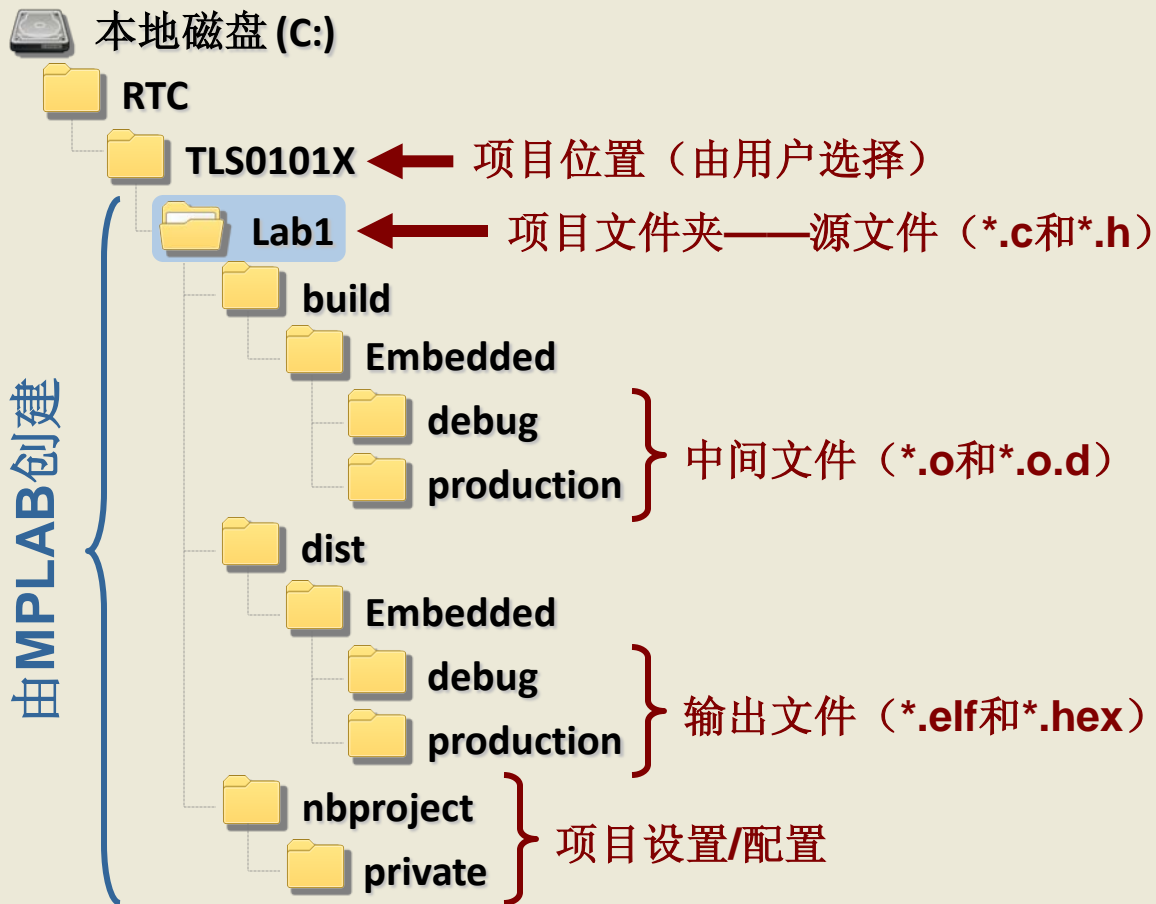
Finish

实验1

创建新项目



MPLAB® X自动生成的项目目录结构

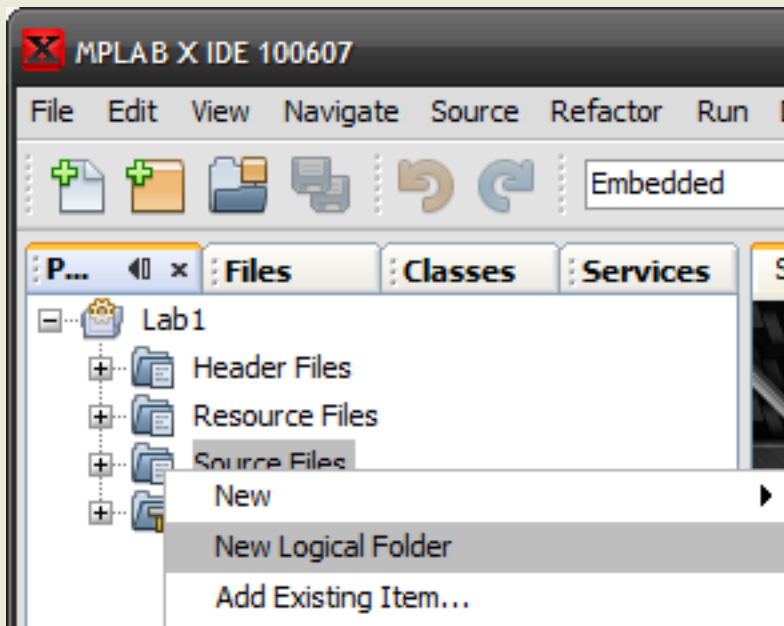


实验1

创建新项目



7 在Source Files下新建逻辑文件夹

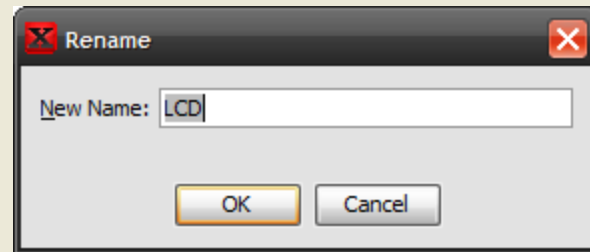
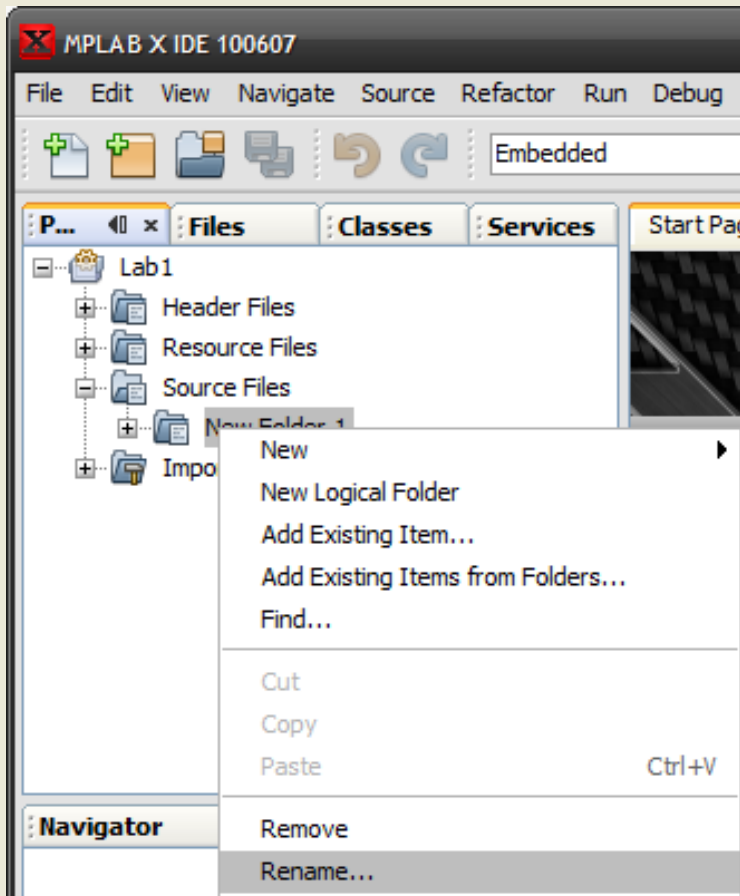


右键单击**Source Files**（源文件），并从弹出菜单中选择**New Logical Folder**（新建逻辑文件夹）



8 重命名新建的逻辑文件夹

展开**Source Files**，显示它的内容。
右键单击**New Folder 1**，并从弹出菜单中选择**Rename...**（重命名...）



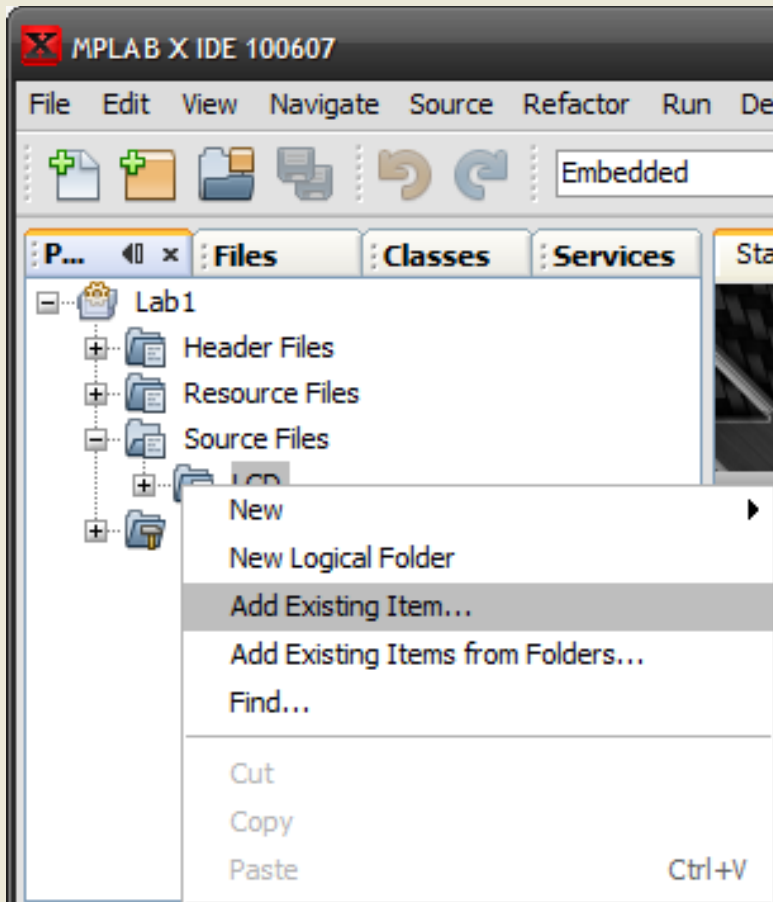
将文件夹命名为**LCD**，并单击**OK**

实验1

创建新项目



9 向逻辑文件夹中添加已有文件



右键单击**LCD**文件夹，并从弹出菜单中选择**Add Existing Item...**
(添加已有项...)

实验1

创建新项目

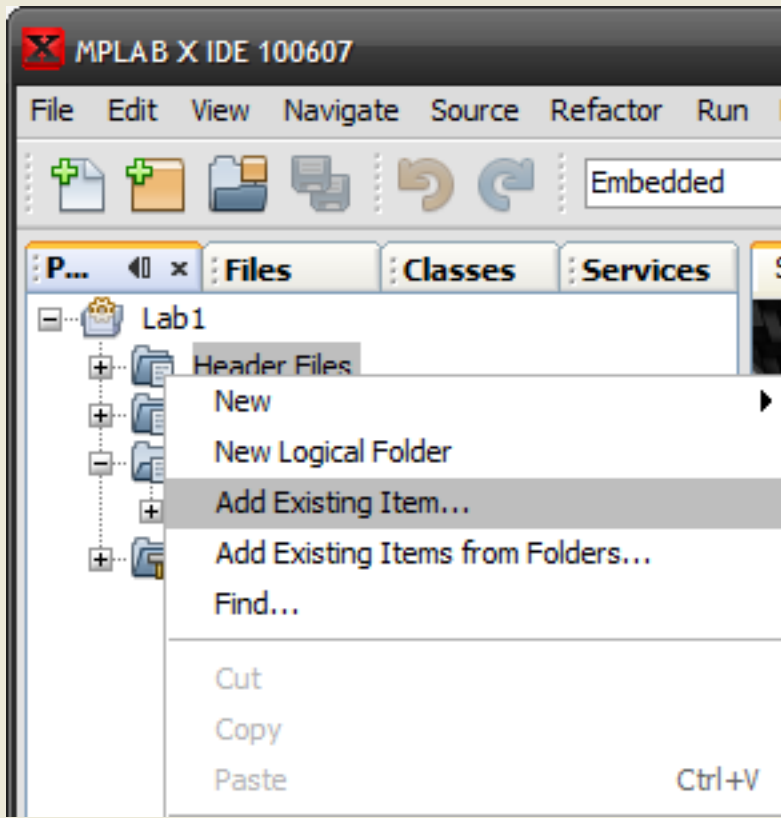


10 选择所有lcd*.c文件（使用ctrl+单击和/或shift+单击）





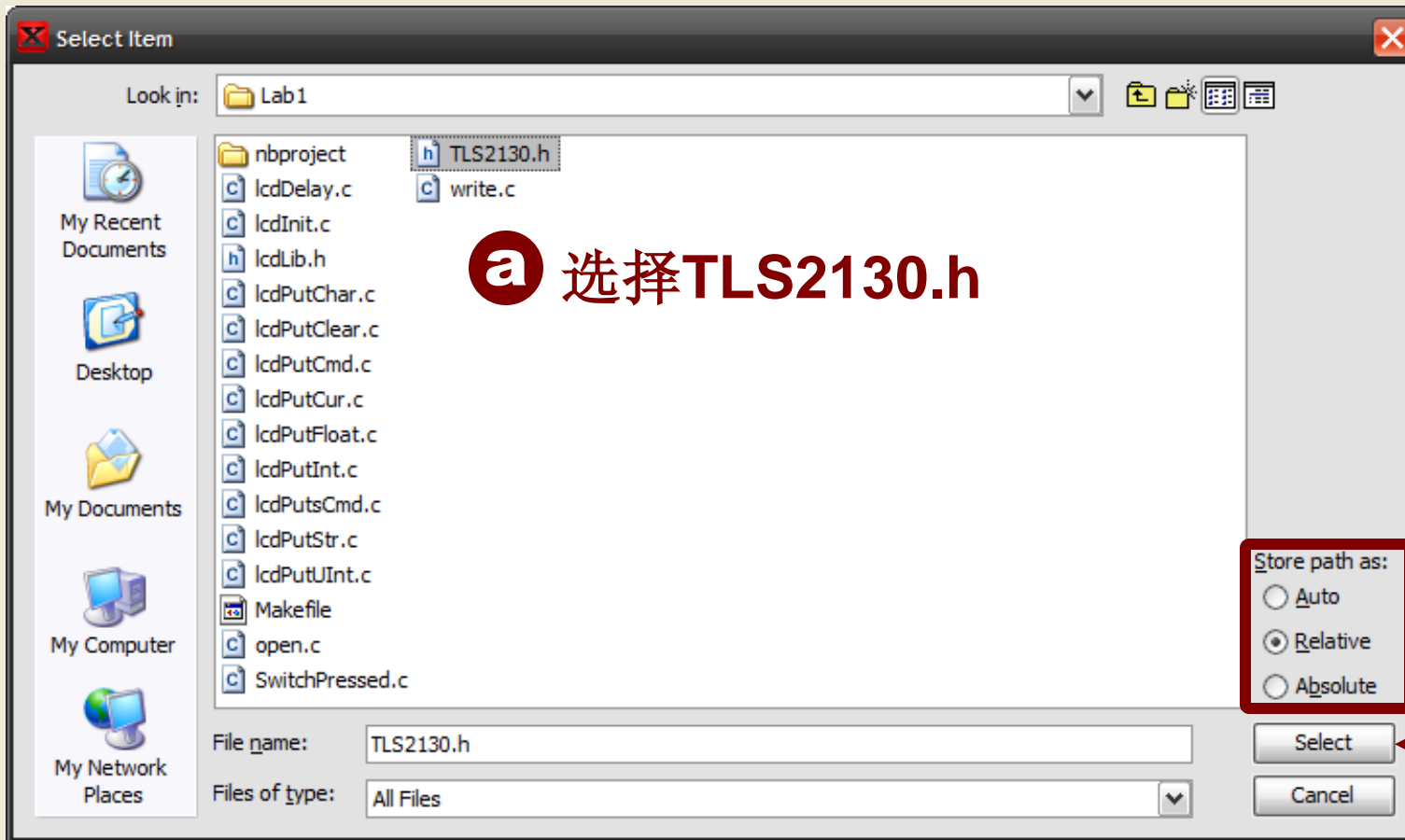
11 添加已有的头文件



右键单击**Header Files**（头文件），并从弹出菜单中选择**Add Existing Item...**



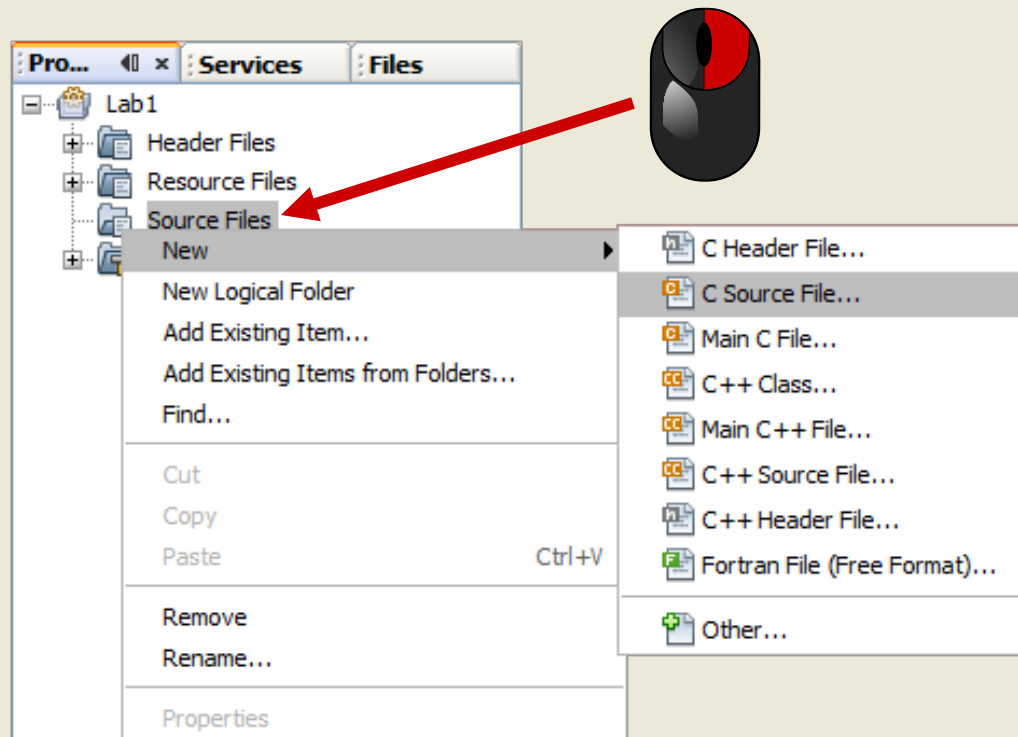
12 选择TLS2130.h





13 新建C源文件

右键单击**Projects**选项卡下的**Source Files**文件夹，并从弹出菜单中选择：**New（新建）** ▶ **C Source File...（C源文件...）**

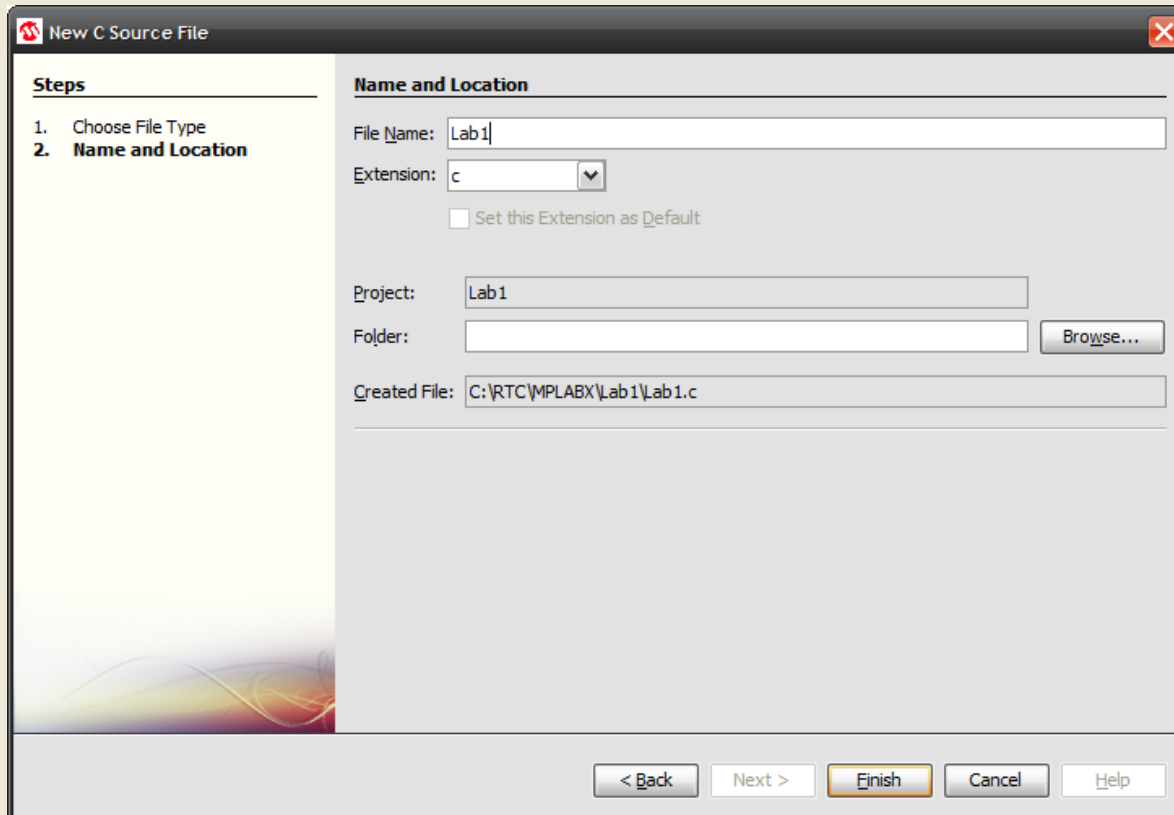


实验1

创建新项目



14 指定新建C源文件的名称和位置



New C Source File

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: Lab1

Extension: c

Set this Extension as Default

Project: Lab1

Folder: Browse...

Created File: C:\RTC\MPLABX\Lab1\Lab1.c

< Back Next > Finish Cancel Help

命名文件。
将**Folder**（文件夹）
框保留为空时，源文件
将创建在项目目录中。

单击

Finish



15 向新建源文件中添加代码

将以下代码添加到Lab1.c中

该内容可以从C:\RTC\TLS0101X\source.txt复制

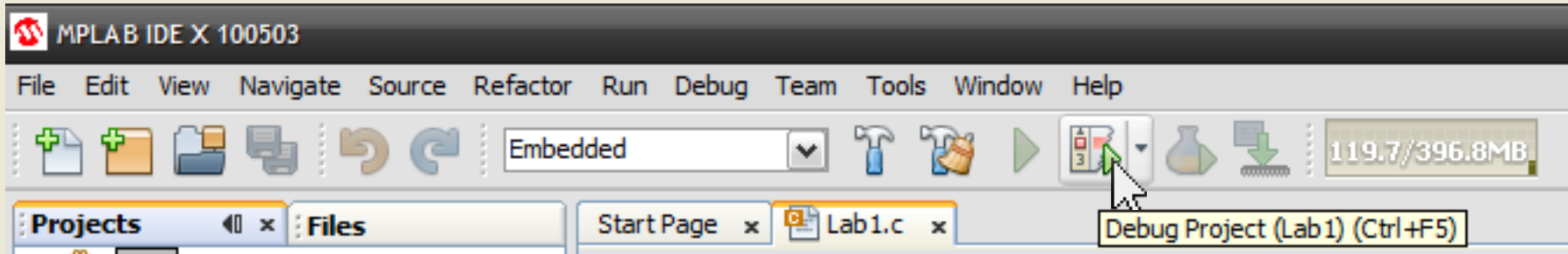
```
1  #include <p24FJ128GA010.h>
2  #include "TLS2130.h"
3
4  _CONFIG1(FWDTEN_OFF & JTAGEN_OFF)
5
6  int main(void)
7  {
8  lcdInit();
9  lcdPutStr("Hello, world!");
10 TRISA = 0;
11 LATA = 0x55;
12     while(1);
13 }
```


实验1

创建新项目





16 编译项目并在演示板上运行程序



单击**Debug Project**（调试项目） 工具栏按钮
这将会：

- 编译代码用于进行调试
- 烧写目标器件
- 运行程序

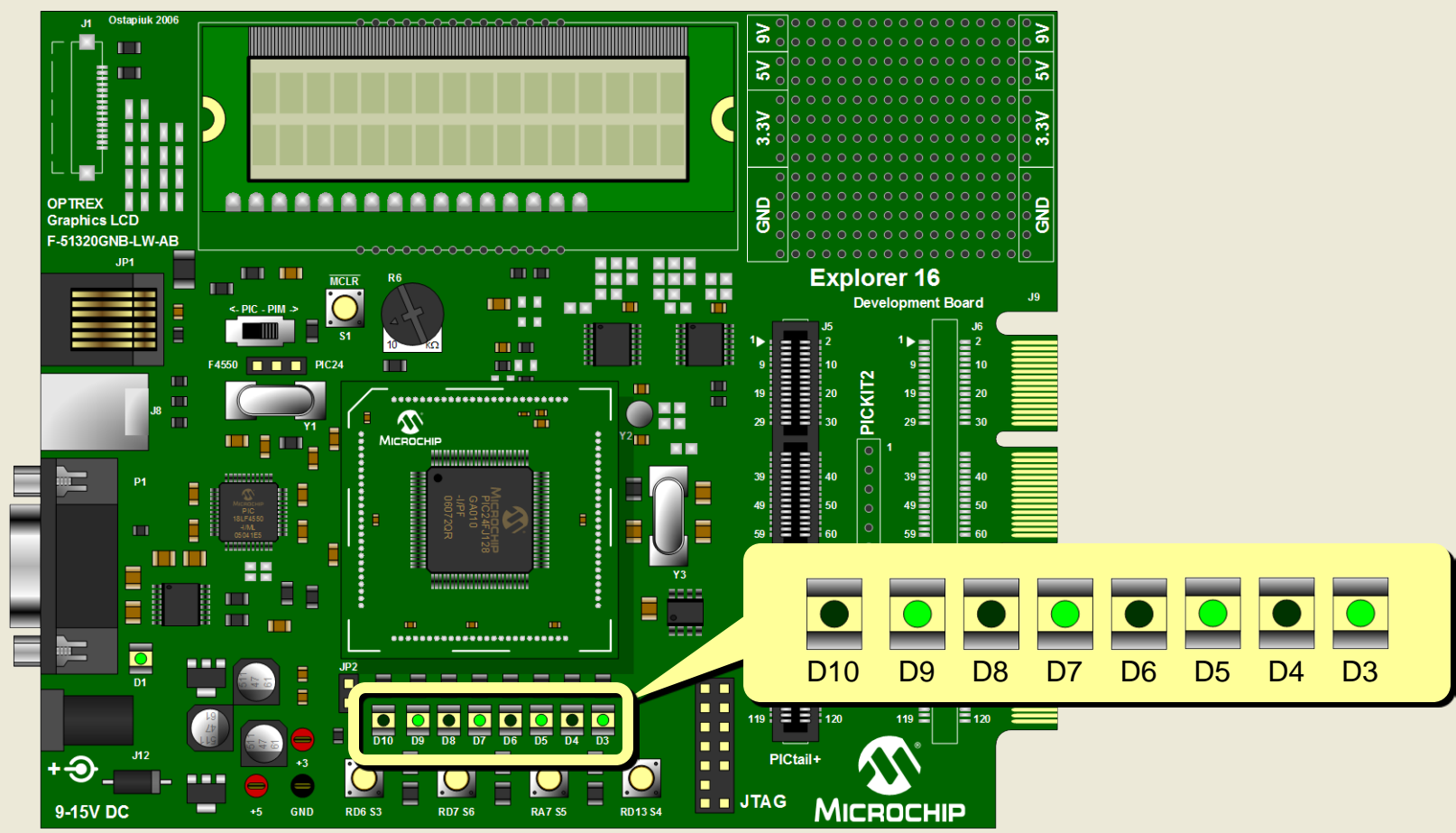
在完成时，单击**Pause**（暂停）, 然后单击
End Debug Session（结束调试会话）

实验1

创建新项目



 D9、D7、D5和D3将点亮，LCD上会显示“Hello, world!”





MICROCHIP 2010

MASTERS Conference

如何编译应用程序

代码编译或汇编

如何编译应用程序

MPLAB® 8



Make (编译)

(菜单: **Project** ▶ **Make**)

仅编译自上次编译之后发生更改的文件



Build All (全部编译)

(菜单: **Project** ▶ **Build All**)

无论自上次编译之后文件是否发生更改, 都编译所有文件

MPLAB X



Build Project

(编译项目)

仅编译自上次编译之后发生更改的文件



Clean and Build Project

(清除并编译项目)

无论自上次编译之后文件是否发生更改, 都编译所有文件

如何编译应用程序

使用调试器

MPLAB® 8

Debug  调试模式

 **Make**

 **Program Target Device**
(烧写目标器件)

 **Reset (复位)**

 **Run (运行)**

MPLAB X



Debug Project
(调试项目)

- Debug Mode (调试模式)
- Make
- Program Target Device
- Reset
- Run



大多数情况下，在单击该按钮之前，不需要单击“**Build**”或“**Clean and Build**”按钮。

如何编译应用程序

使用编程器

MPLAB® 8

Release  发布模式

 **Make**

 **Program Target Device**

MPLAB X



Program Target Project
(烧写目标项目)

- Release Mode (发布模式)
- Make
- Program Target Device



大多数情况下，在单击该按钮之前，不需要单击“**Build**”或“**Clean and Build**”按钮。

如何编译应用程序

使用调试器/编程器

MPLAB® 8

Release  发布模式

 **Make**

 **Program Target Device**

 **Release from Reset**
(从复位状态释放)

MPLAB X

 **Run Project (运行项目)**

- Release Mode
- Make
- Program Target Device
- Release from Reset (Run)
(从复位状态释放) (运行)



大多数情况下，在单击该按钮之前，不需要单击“**Build**”或“**Clean and Build**”按钮。

如何编译应用程序

陷阱——平台相关错误

- 相对路径应使用 “/” 而不是 “\” :

```
#include "Graphics/Graphics.h"
```

- “/” 在所有平台上都有效，包括Windows
- “\” 仅在Windows上有效
- 对于使用 “\” 的路径，Linux上的编译器将无法查找文件，该字符在Linux Bourne Again Shell (BASH) 中是一个转义符

- Linux区分大小写

- `#include <p24fj128ga010.h>` 将无法找到文件，因为文件名为 `p24FJ128GA010.h`



MICROCHIP 2010

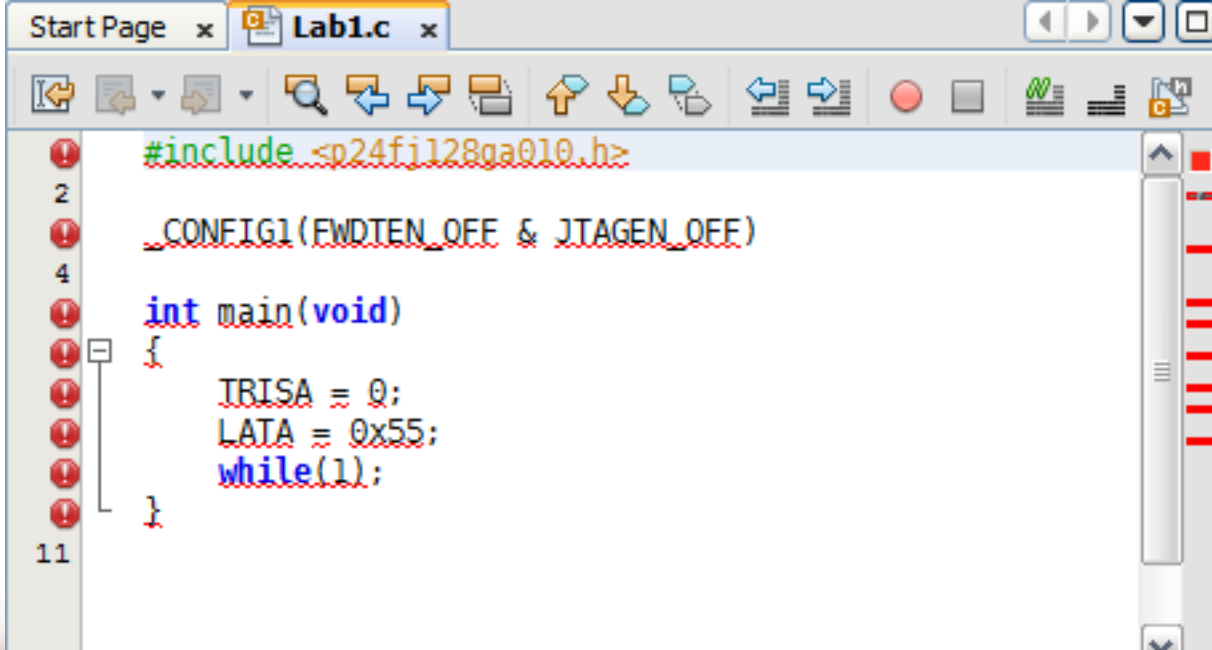
MASTERS Conference

如何使用编辑器

编辑器

即时解析

- IDE会在用户输入时解析代码
- 编辑器会立刻标出错误
- 工具提示和图符会显示编译器提示、警告、快速修复和语言文档



```
Start Page x Lab1.c x
#include <p24fj128ga010.h>
_CONFIG1(FWDTEN_OFF & JTAGEN_OFF)
int main(void)
{
    TRISA = 0;
    LATA = 0x55;
    while(1);
}
```

编辑器

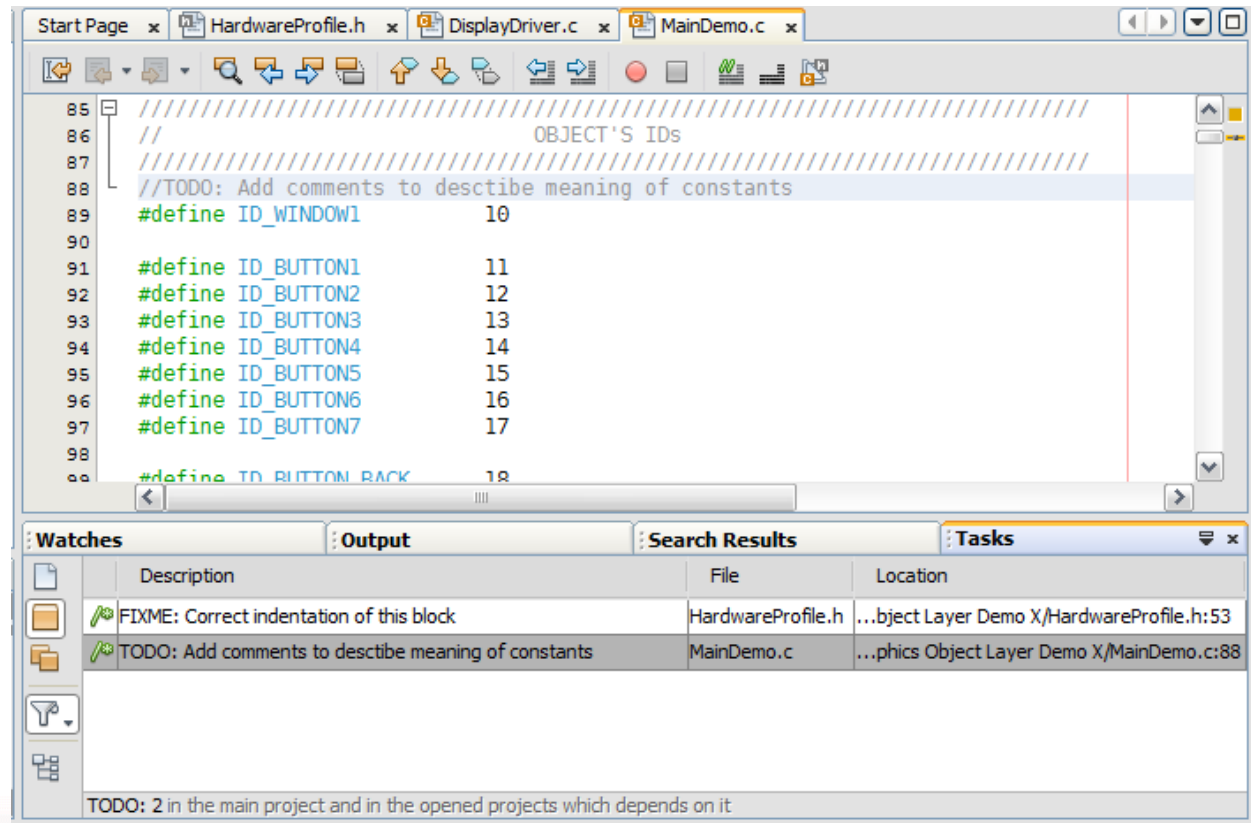
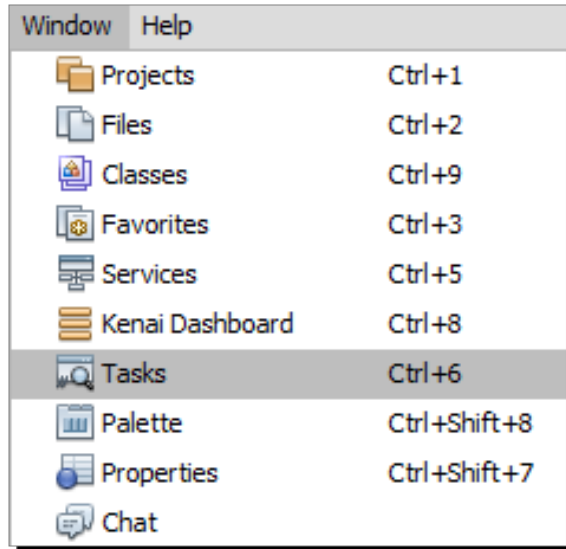
智能代码补全

- 目前的效果还不够好...
- 在输入一个函数时，将会出现一个弹出框，其中会显示参数列表和数据类型
- 输入函数或变量名的前几个字母之后，编辑器会提供标识符的补全建议

编辑器

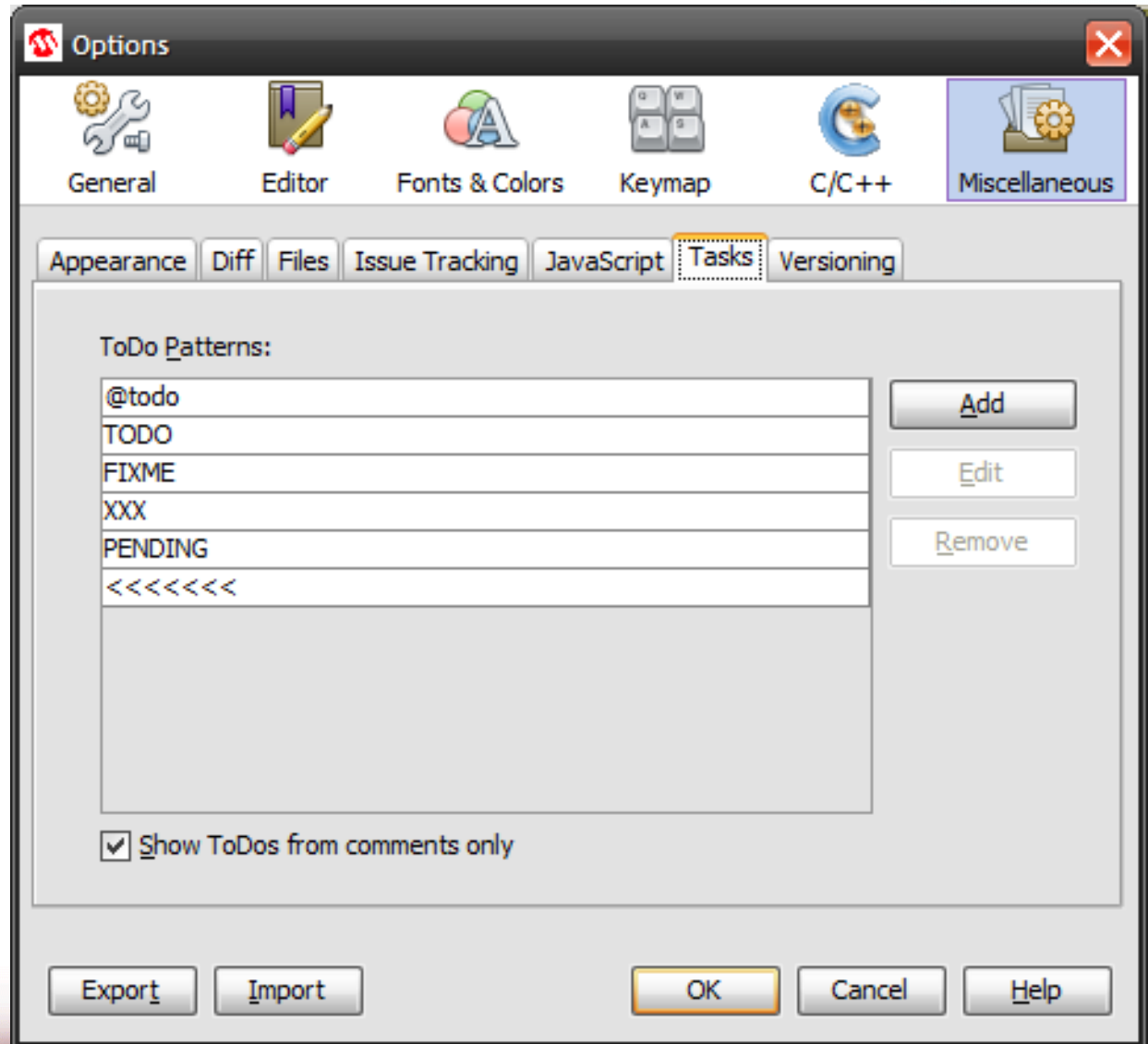
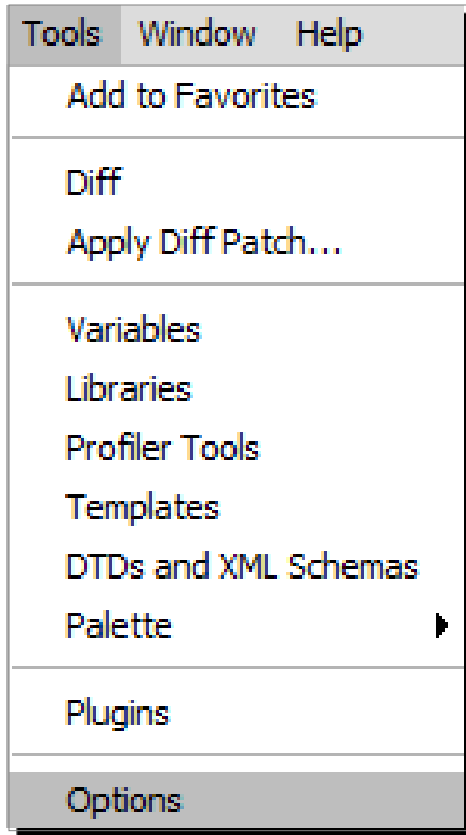
如何使用任务列表

- 自动扫描代码，查找注释中诸如**TODO**和**FIXME**之类的字词



编辑器

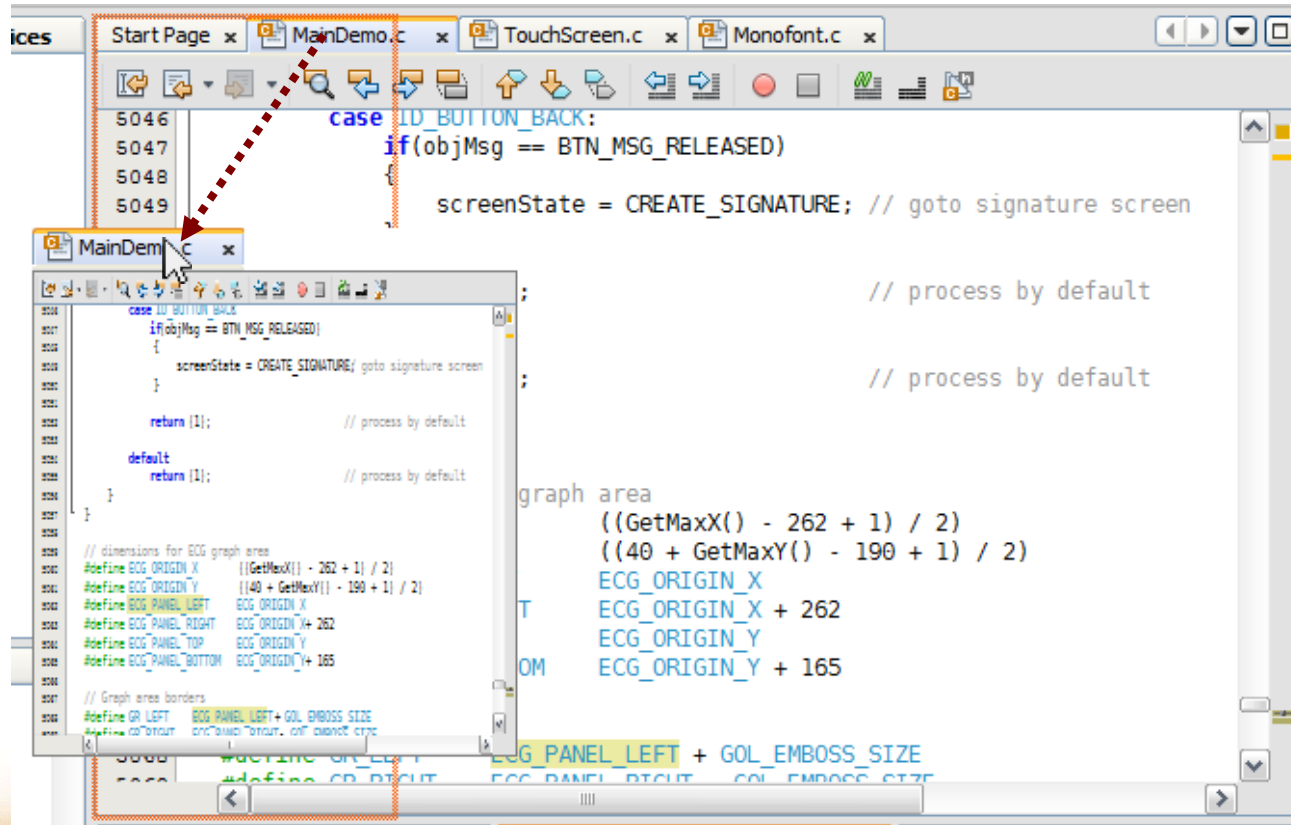
如何向任务列表解析器中添加关键字



编辑器

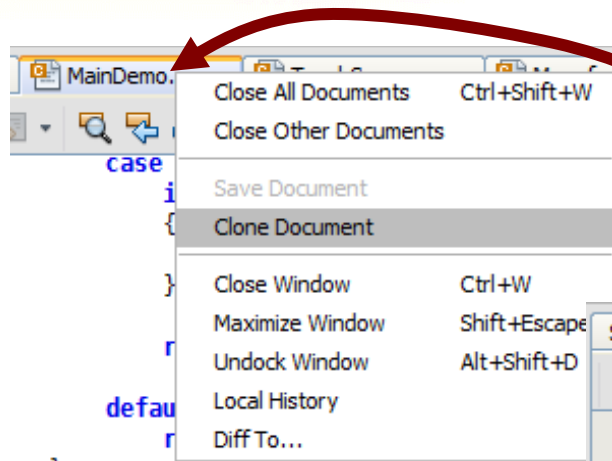
如何并排查看两个文件

单击某个选项卡，并将它拖离编辑器窗格顶部。红色轮廓框将指示在释放时它将位于何处。



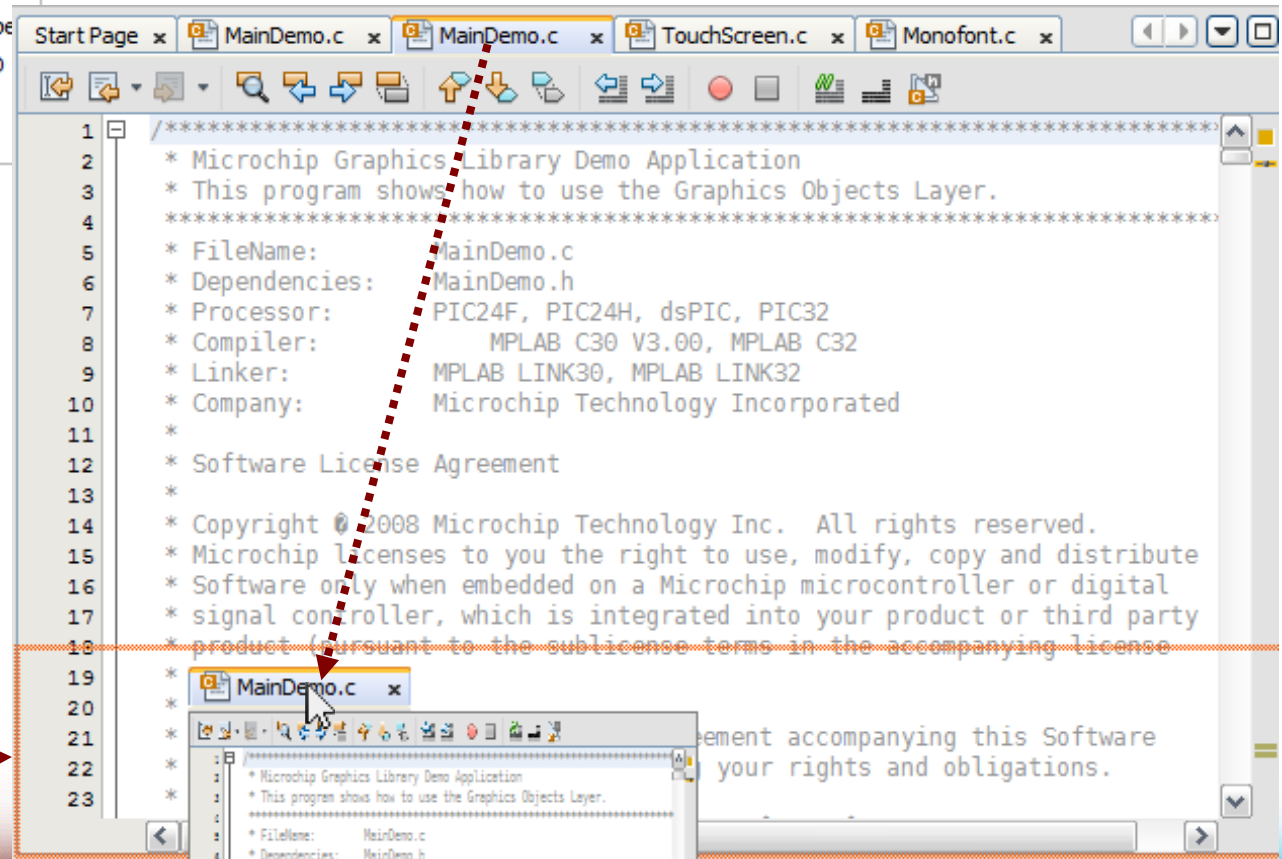
编辑器

如何拆分单个文件的视图



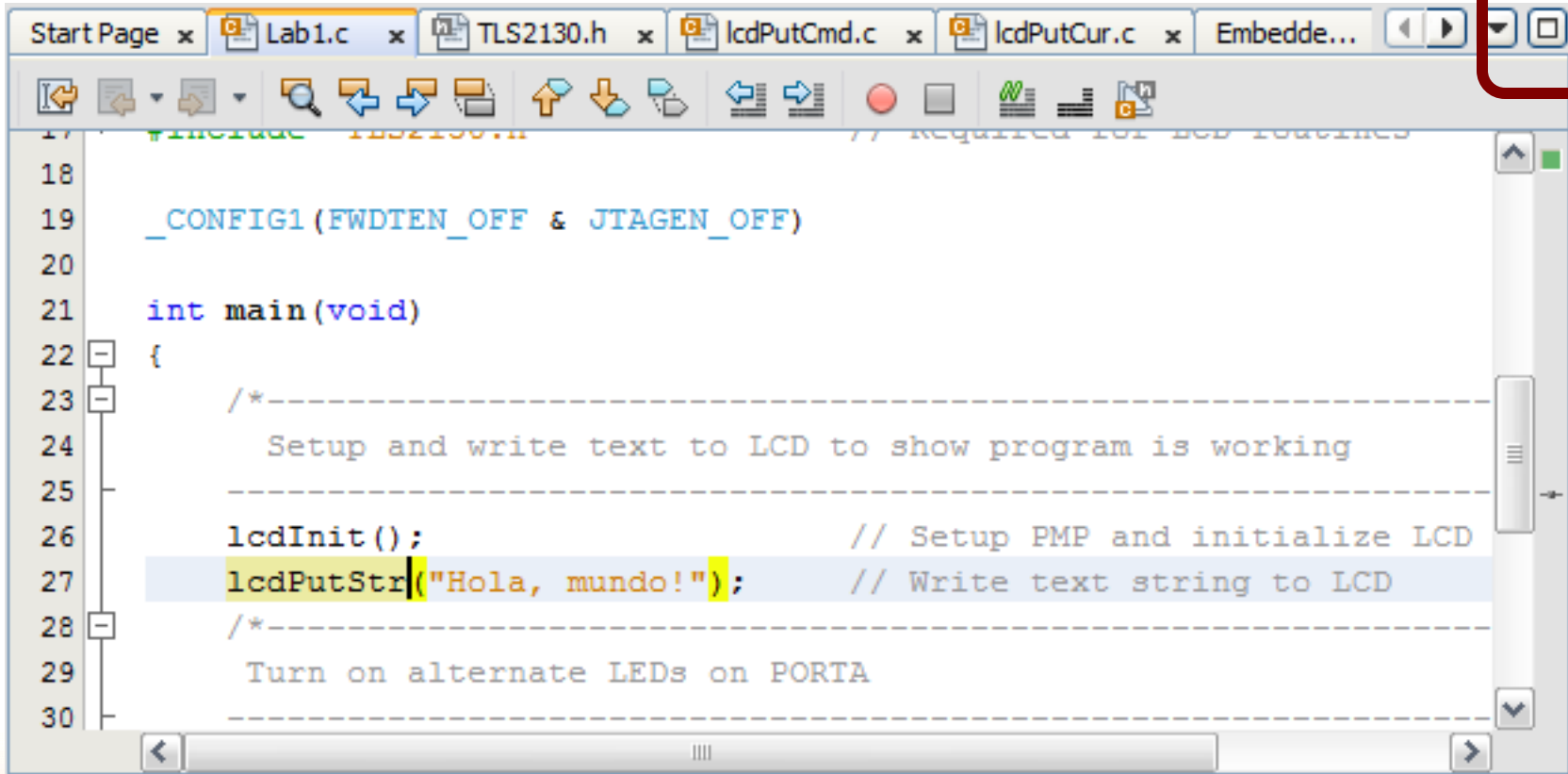
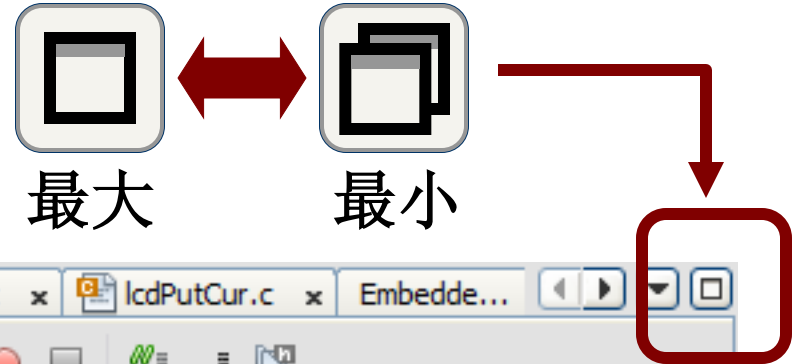
1 右键单击文件的选项卡，并从弹出菜单中选择**Clone Document**（克隆文档）

2 单击复制的选项卡并将它拖放到新位置



编辑器

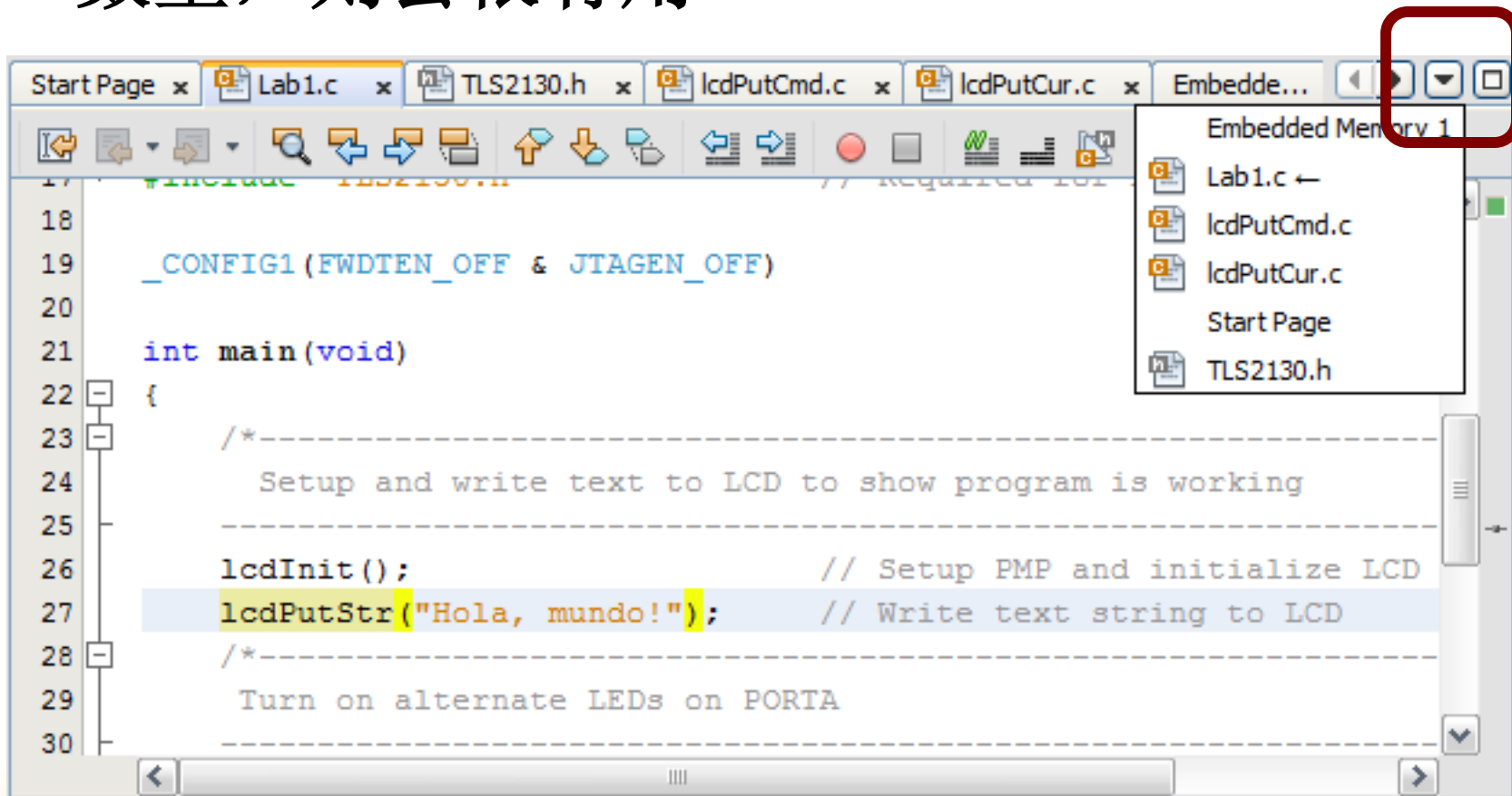
如何最大化/恢复编辑器窗口



编辑器

如何列出所有已打开选项卡，并在其间跳转

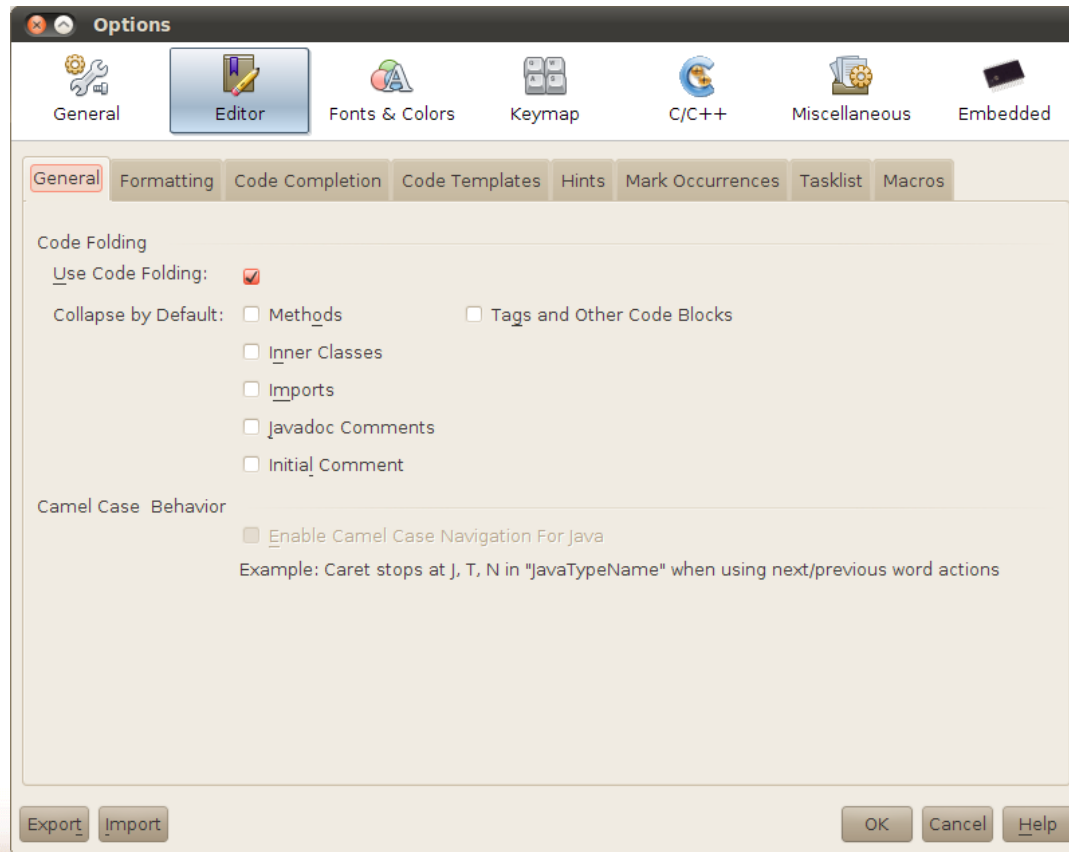
- 如果已打开选项卡的数量超出可在顶部显示的数量，则会很有用



编辑器

如何配置编辑器的设置

- 从菜单中选择 **Tools (工具)** ▶ **Options (选项)**，然后从顶部图标中选择  **Editor (编辑器)**





MICROCHIP 2010

MASTERS Conference

如何使用主工具栏

如何使用主工具栏

默认按钮



- | | |
|---|--|
|  New File (新建文件) |  Build (编译) |
|  New Project (新建项目) |  Rebuild (重新编译) (清除并编译) |
|  Open Project (打开项目) |  Run Project (运行项目) (编译、烧写和运行) |
|  Save All (全部保存) |  Debug (调试) (编译、烧写和运行) |
|  Undo (撤销) |  Profile Project (项目概要) |
|  Redo (重做) |  Program Target (烧写目标器件) |

Embedded

项目配置

41.6/61.9MB

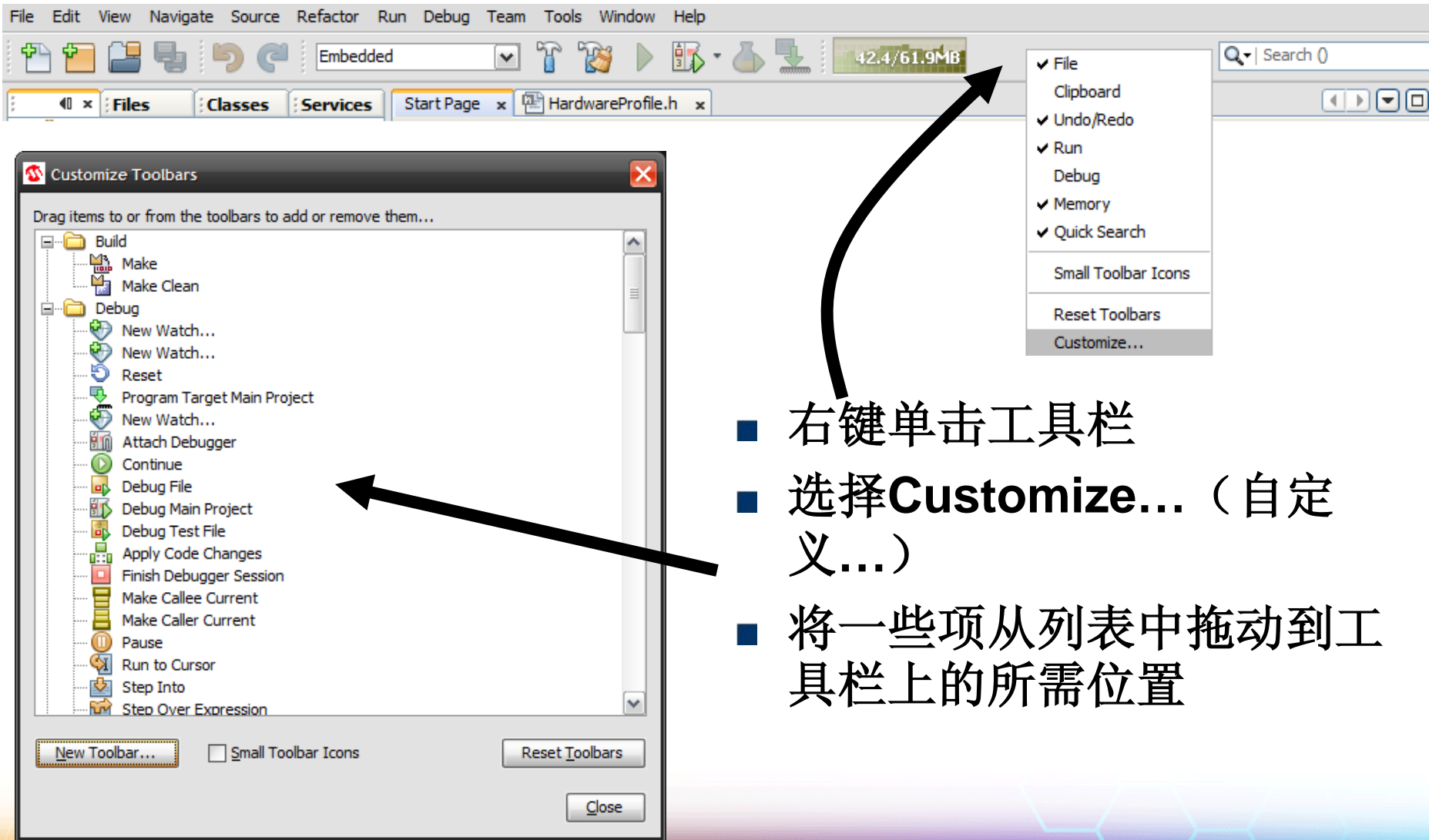
MPLAB® X
存储器使用情况

Search 0

快速搜索

如何使用主工具栏

如何自定义工具栏



File Edit View Navigate Source Refactor Run Debug Team Tools Window Help

Embedded 42.4/61.9MB

Files Classes Services Start Page x HardwareProfile.h x

Customize Toolbars

Drag items to or from the toolbars to add or remove them...

- Build
 - Make
 - Make Clean
- Debug
 - New Watch...
 - New Watch...
 - Reset
 - Program Target Main Project
 - New Watch...
 - Attach Debugger
 - Continue
 - Debug File
 - Debug Main Project
 - Debug Test File
 - Apply Code Changes
 - Finish Debugger Session
 - Make Callee Current
 - Make Caller Current
 - Pause
 - Run to Cursor
 - Step Into
 - Step Over Expression

New Toolbar... Small Toolbar Icons Reset Toolbars

Close

- File
- Clipboard
- Undo/Redo
- Run
- Debug
- Memory
- Quick Search
- Small Toolbar Icons
- Reset Toolbars
- Customize...

- 右键单击工具栏
- 选择**Customize...**（自定义...）
- 将一些项从列表中拖动到工具栏上的所需位置



MICROCHIP 2010

MASTERS Conference



实验2

如何转换已有项目

讲师指导的动手练习

实验2

转换已有项目



目标

- 使用项目向导转换来自MAL的**Graphics Object Layer Demo**项目
- 在**Explorer 16**上使用图形演示板（**V2或V3**）运行程序



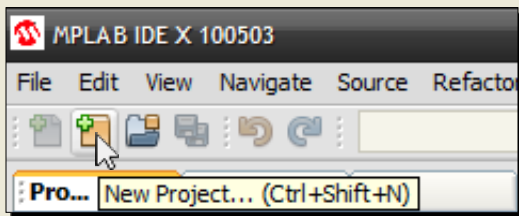
要求： 带有PIC24FJ128GA010的Explorer 16演示板

实验2

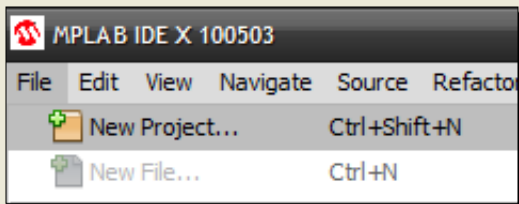
转换已有项目



1 启动新建项目向导



在工具栏中单击**New Project**图标, 或从菜单中选择**File ▶ New Project**



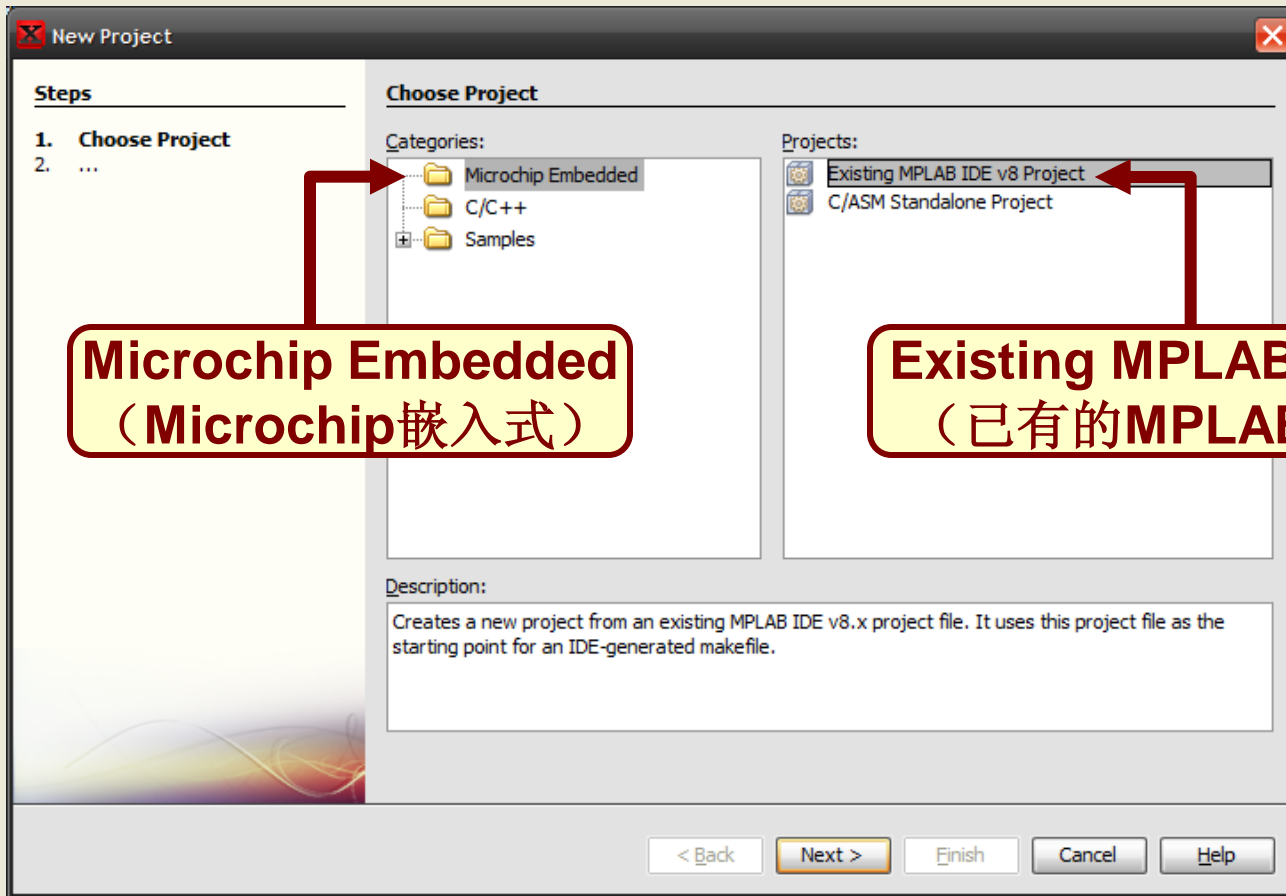
不同于MPLAB® 8, 在MPLAB X中必须使用向导, 因为它会同时创建项目目录本身和正确工作所需的完整子目录层级结构。手动创建项目是不可能的。

实验2

转换已有项目



2 选择项目



单击

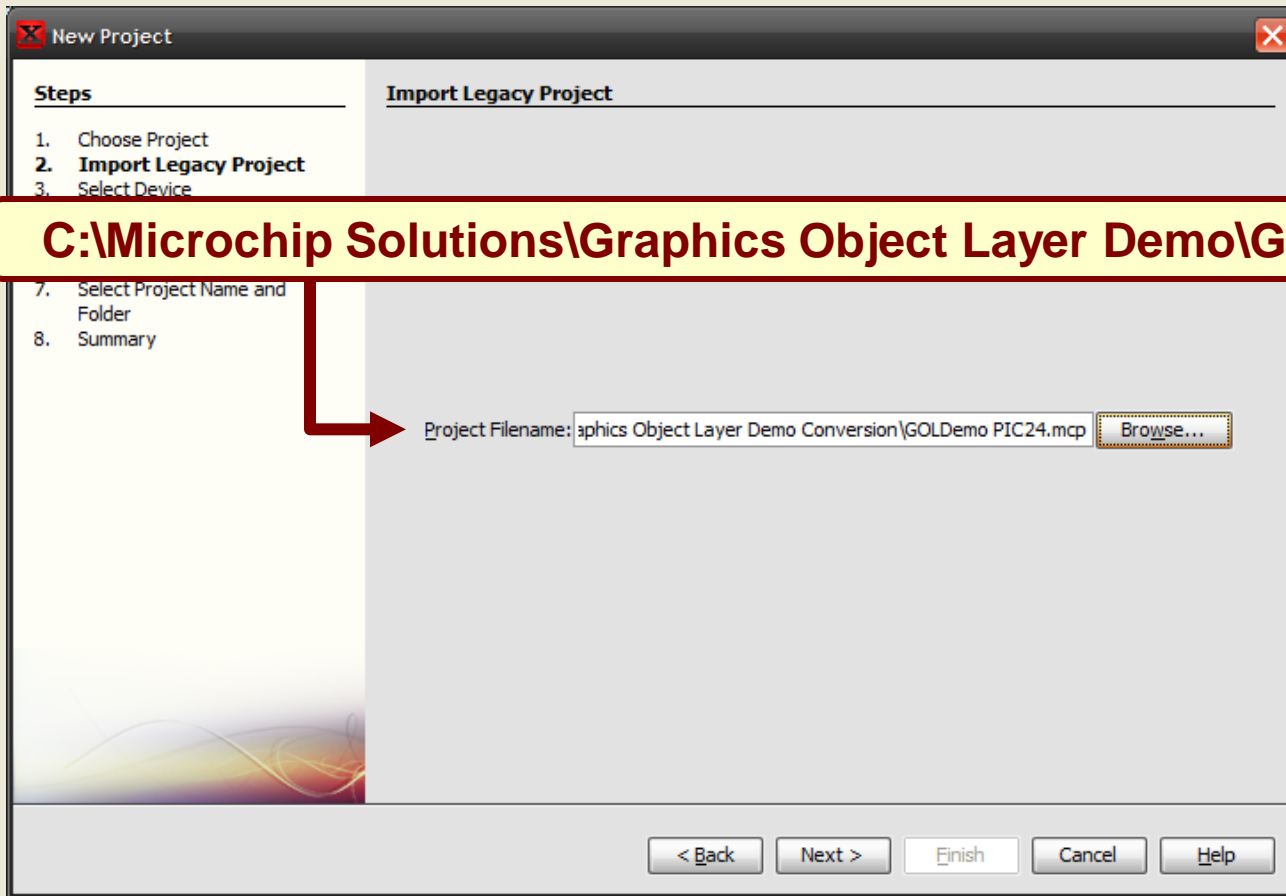
Next >

实验2

转换已有项目



3 导入旧有项目



单击

Next >

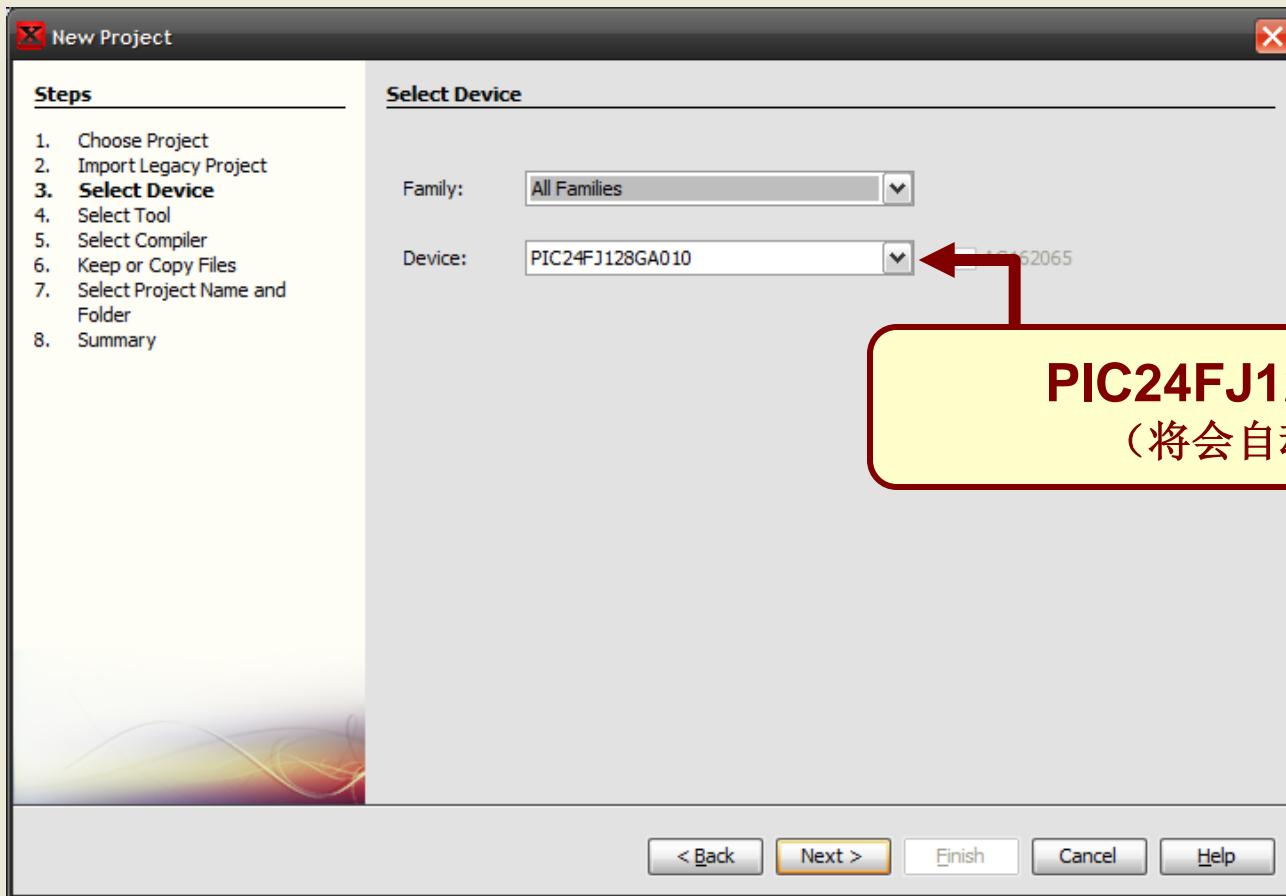
实验2

转换已有项目



4

选择器件



PIC24FJ128GA010
(将会自动选择)

单击

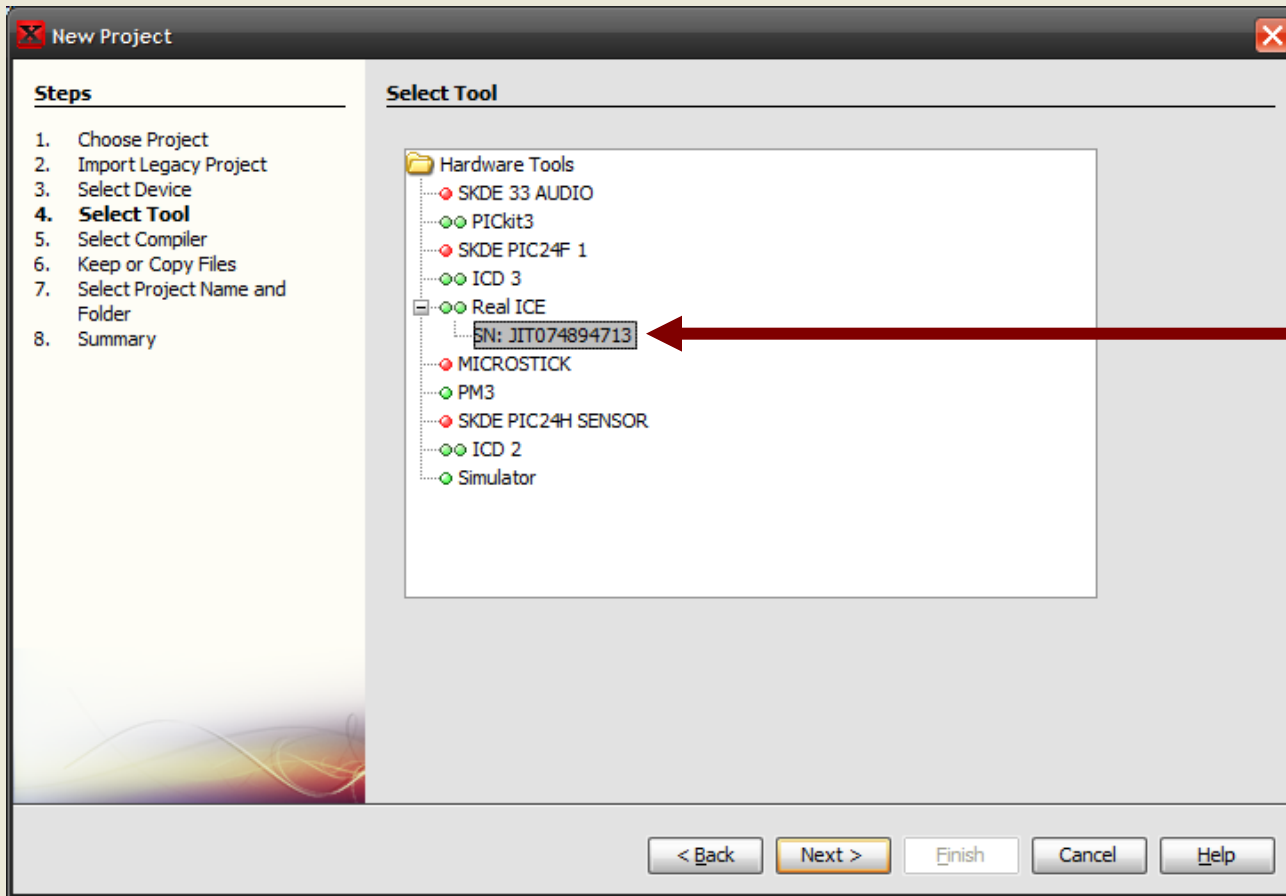
Next >

实验2

转换已有项目



5 选择工具



选择
Real ICE

单击

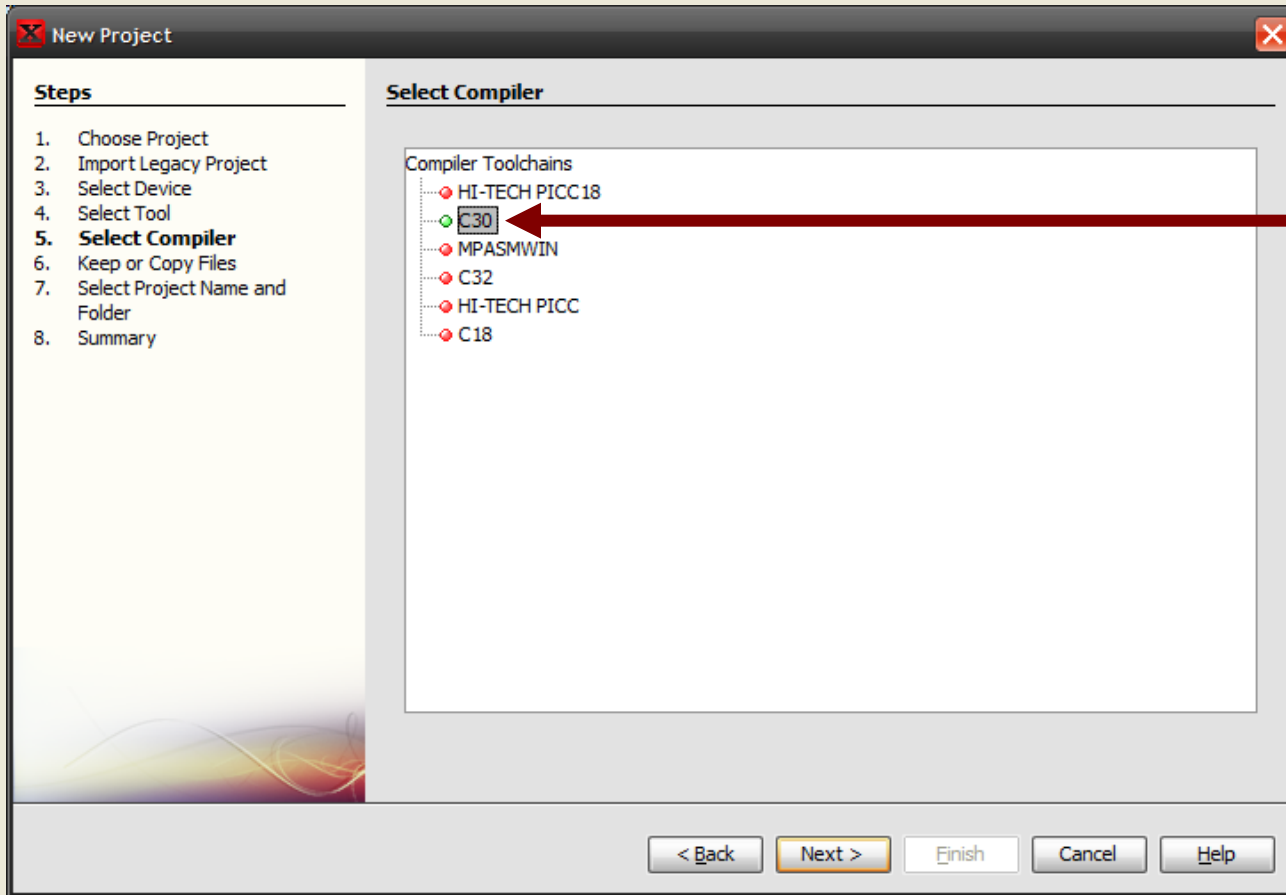
Next >

实验2

转换已有项目



6 选择编译器



选择
C30

单击

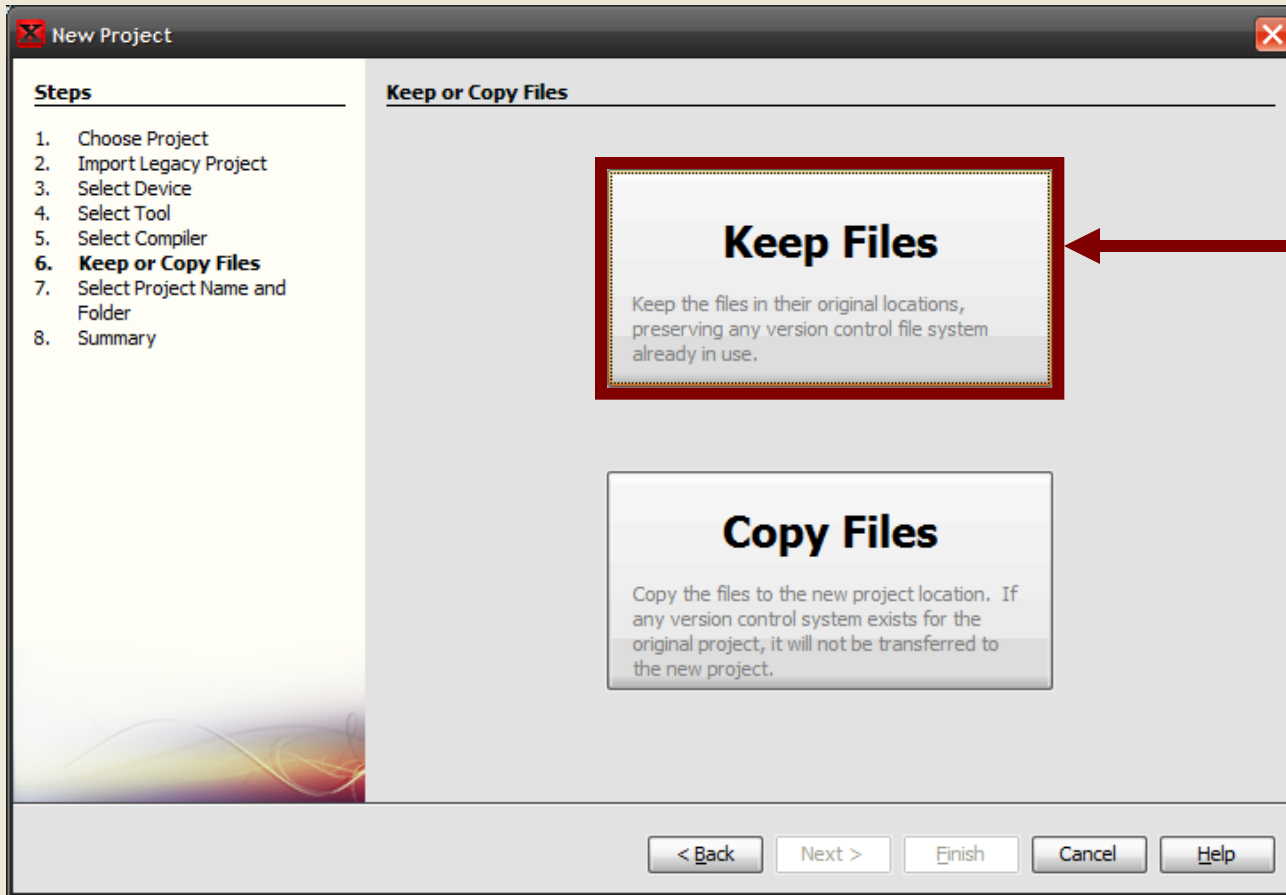
Next >

实验2

转换已有项目



7 保留或复制文件



单击
Keep Files
(保留文件)

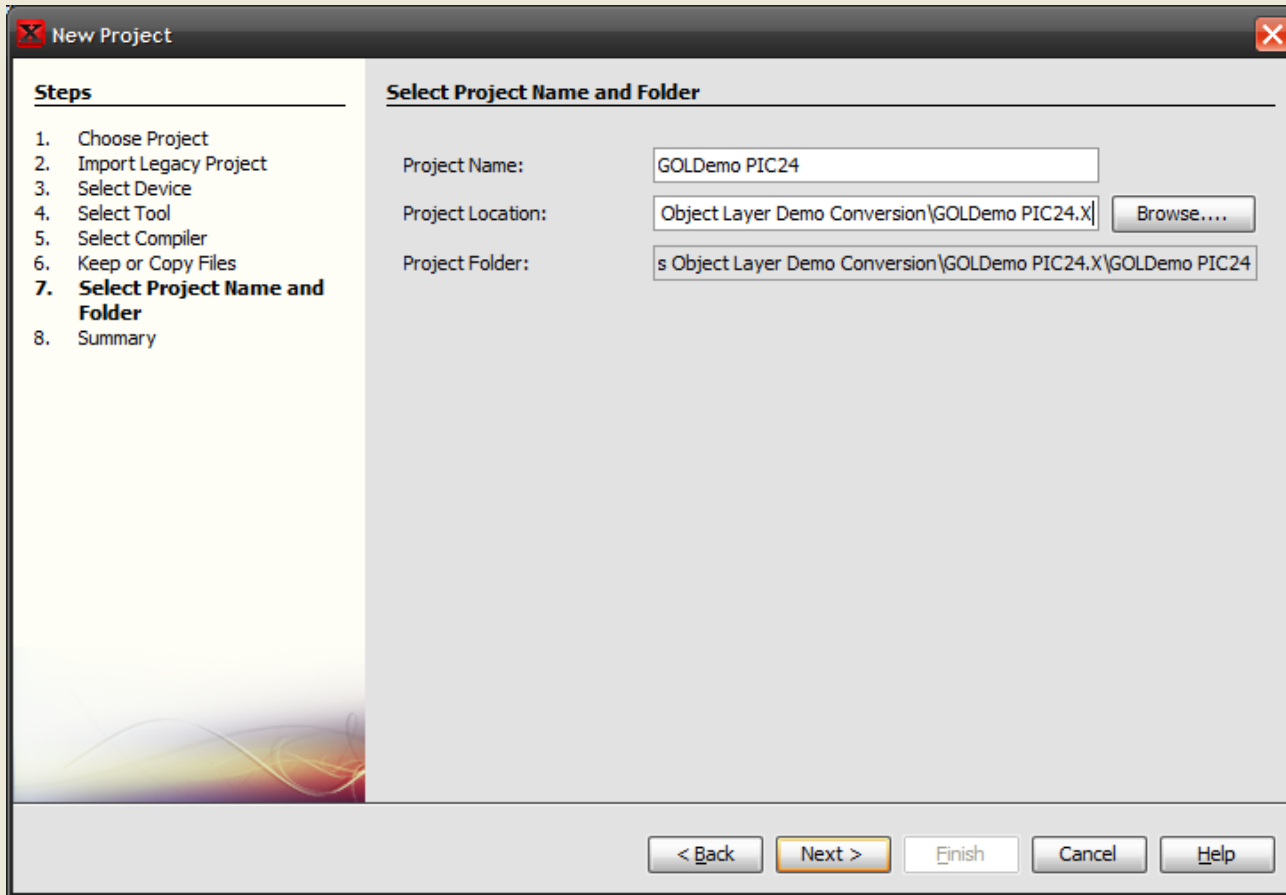
单击
Next >

实验2

转换已有项目



8 选择项目名称和文件夹



接受
默认设置

单击

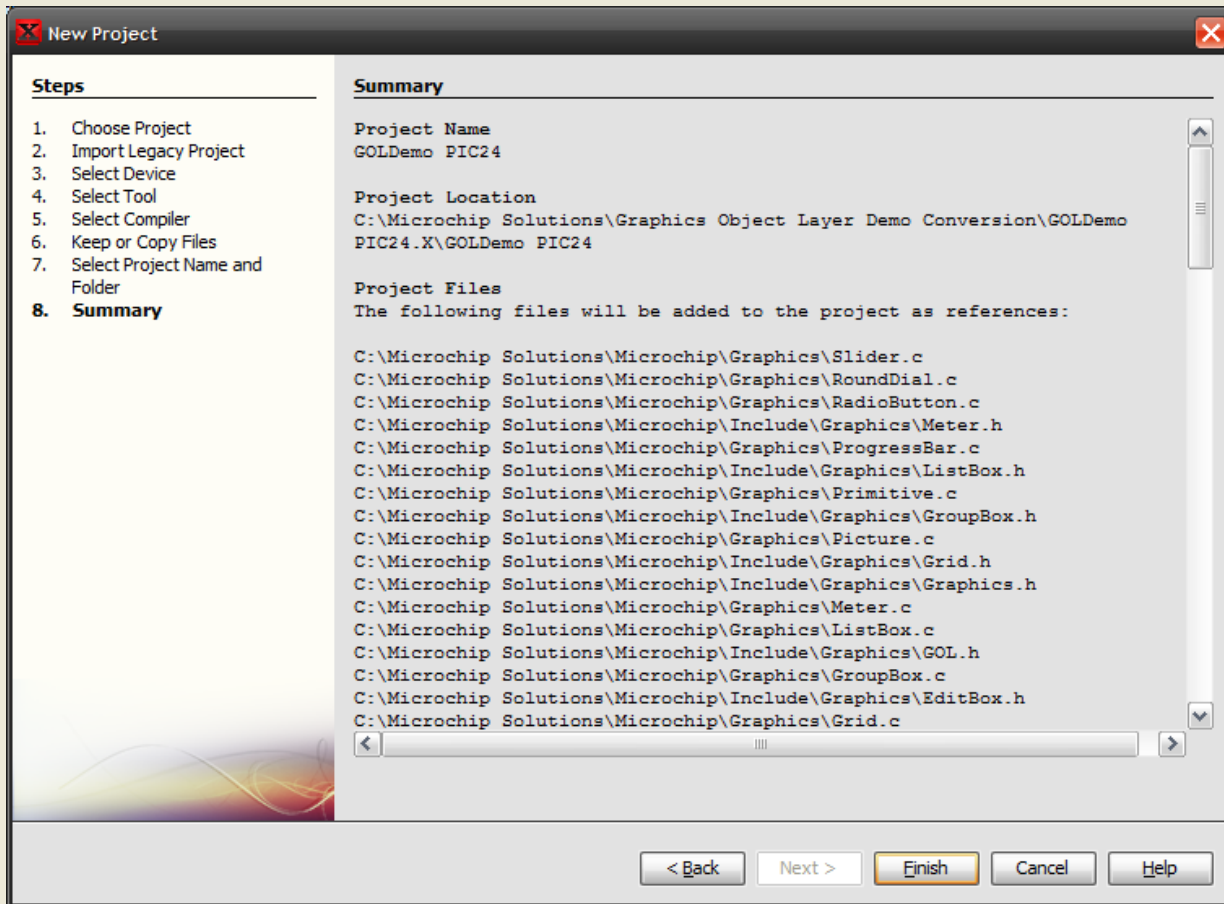
Next >

实验2

转换已有项目



9 汇总



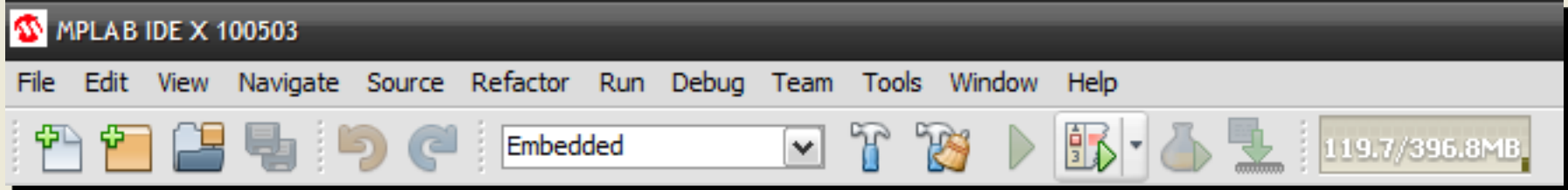
单击
Finish





实验2

转换已有项目



10 编译并测试项目



- a** 单击工具栏上的**Clean and Build**  按钮
- b** 如果编译成功，则说明项目已正确转换。为了绝对确保这一点，可通过单击工具栏上的**Debug**  按钮尝试在硬件上执行它
- c** 在完成时，单击**Pause** ，然后单击**End Debug Session** 



MICROCHIP 2010

MASTERS Conference

如何使用调试器

如何编译应用程序

使用调试器

MPLAB® 8

(无等效操作)

-  **Halt** (暂停)
-  **Run** (运行)
-  **Animate** (连续单步运行)
-  **Run to Cursor** (运行到光标)
(上下文菜单)
-  **Step Into** (单步运行)
-  **Step Over** (单步跳过)
-  **Step Out** (单步跳出)
-  **Reset** (复位)


MPLAB X

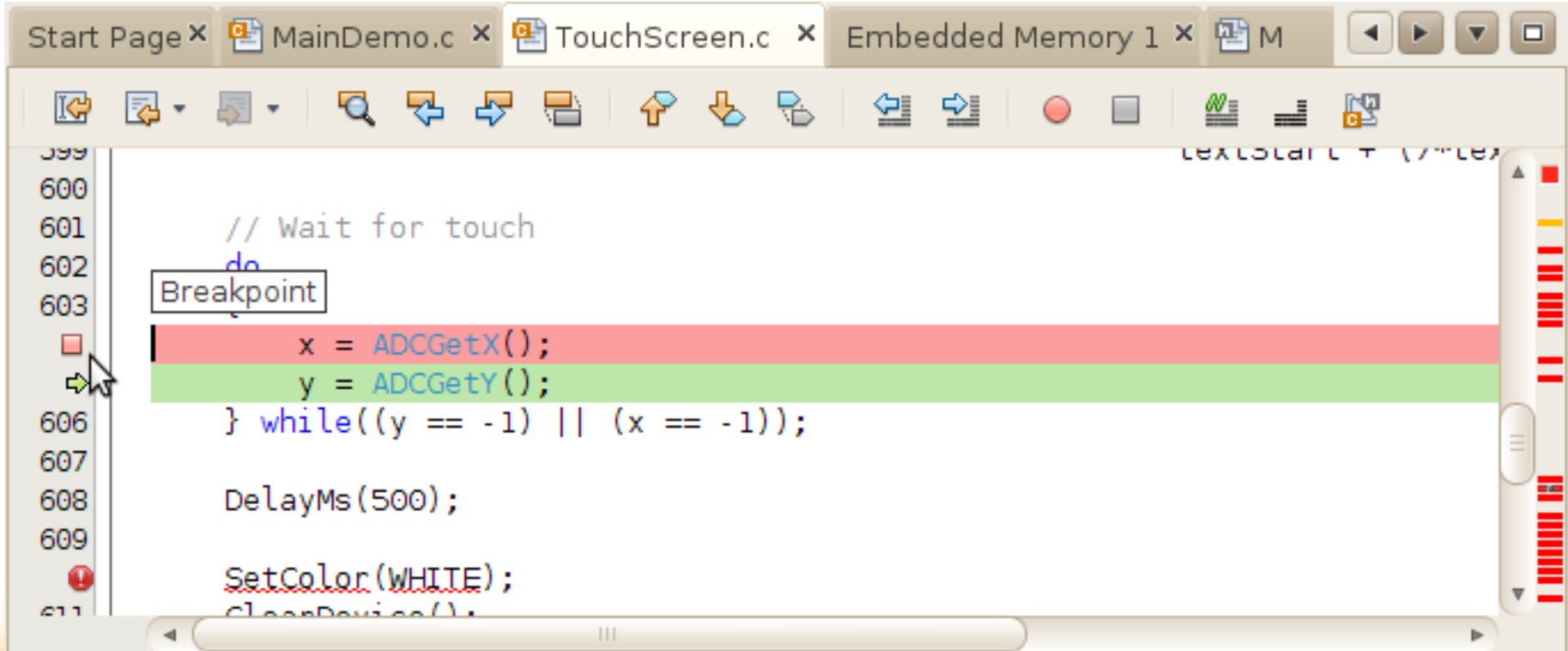
-  **Finish Debugger Session**
(完成调试器会话)
-  **Pause** (暂停)
-  **Continue** (继续)
(尚无等效操作...)
-  **Run To Cursor** (运行到光标)
-  **Step Into** (单步运行)
-  **Step Over** (单步跳过)
-  **Step Out** (单步跳出)
-  **Reset**
(复位)

图标已改变

断点

如何设置行断点

- 单击左页边（图符页边）中的行号，直到出现红色方形（）为止（已设置断点）。再次单击可清除断点。



The screenshot shows an IDE window with several tabs: Start Page, MainDemo.c, TouchScreen.c, Embedded Memory 1, and M. The main editor displays C code. A red square icon, representing a breakpoint, is set on line 603. A mouse cursor is pointing at this icon. The code on line 603 is highlighted in green, and the line number '603' in the left margin is highlighted in red. The code on line 603 is:

```
    x = ADCGetX();  
    y = ADCGetY();
```

The code on line 606 is:

```
} while((y == -1) || (x == -1));
```

The code on line 608 is:

```
DelayMs(500);
```

The code on line 609 is:

```
SetColor(WHITE);
```

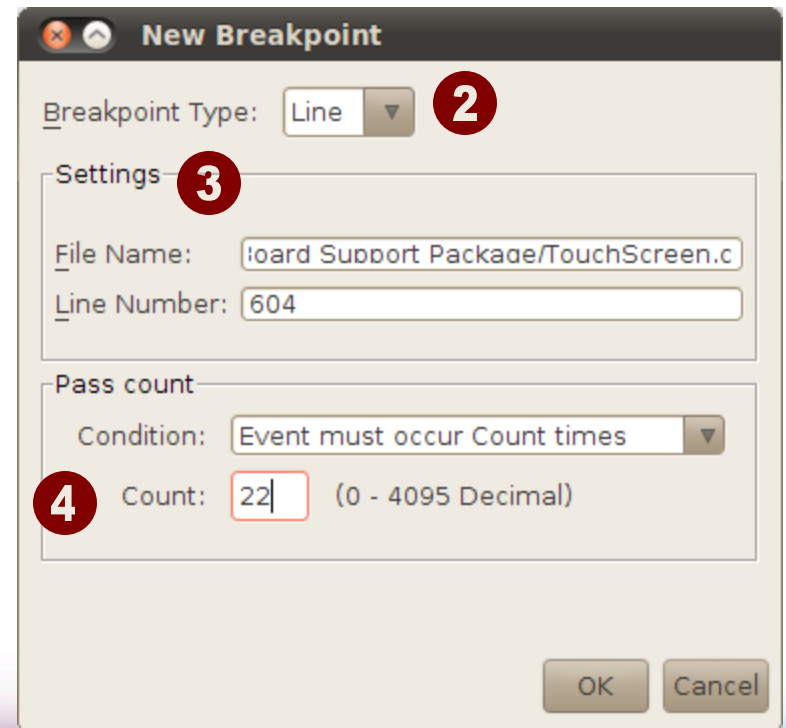
The code on line 611 is:

```
ClearDevice();
```

断点

如何设置条件性行断点

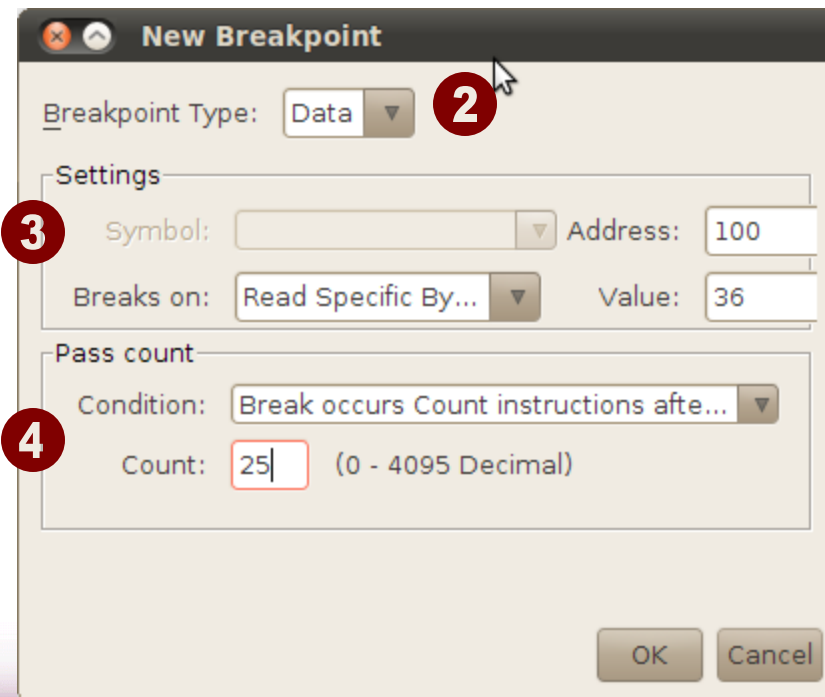
- 1 选择**Debug**（调试）▶ **New Breakpoint**（新断点）
- 2 选择**Line**（行）作为断点类型
- 3 指定数据**File**（文件）和**Line Number**（行号）设置
（根据光标位置自动填充）
- 4 指定**Pass count**（发生次数
计数） **Condition**（条件）



断点

如何设置数据断点

- 1 选择**Debug** ▶ **New Breakpoint**
- 2 选择**Data**（数据）作为断点类型
- 3 指定数据**Address**（地址）和**Breaks on**（中断条件）设置
- 4 指定**Pass count Condition**



断点

如何查看项目中所有断点的列表

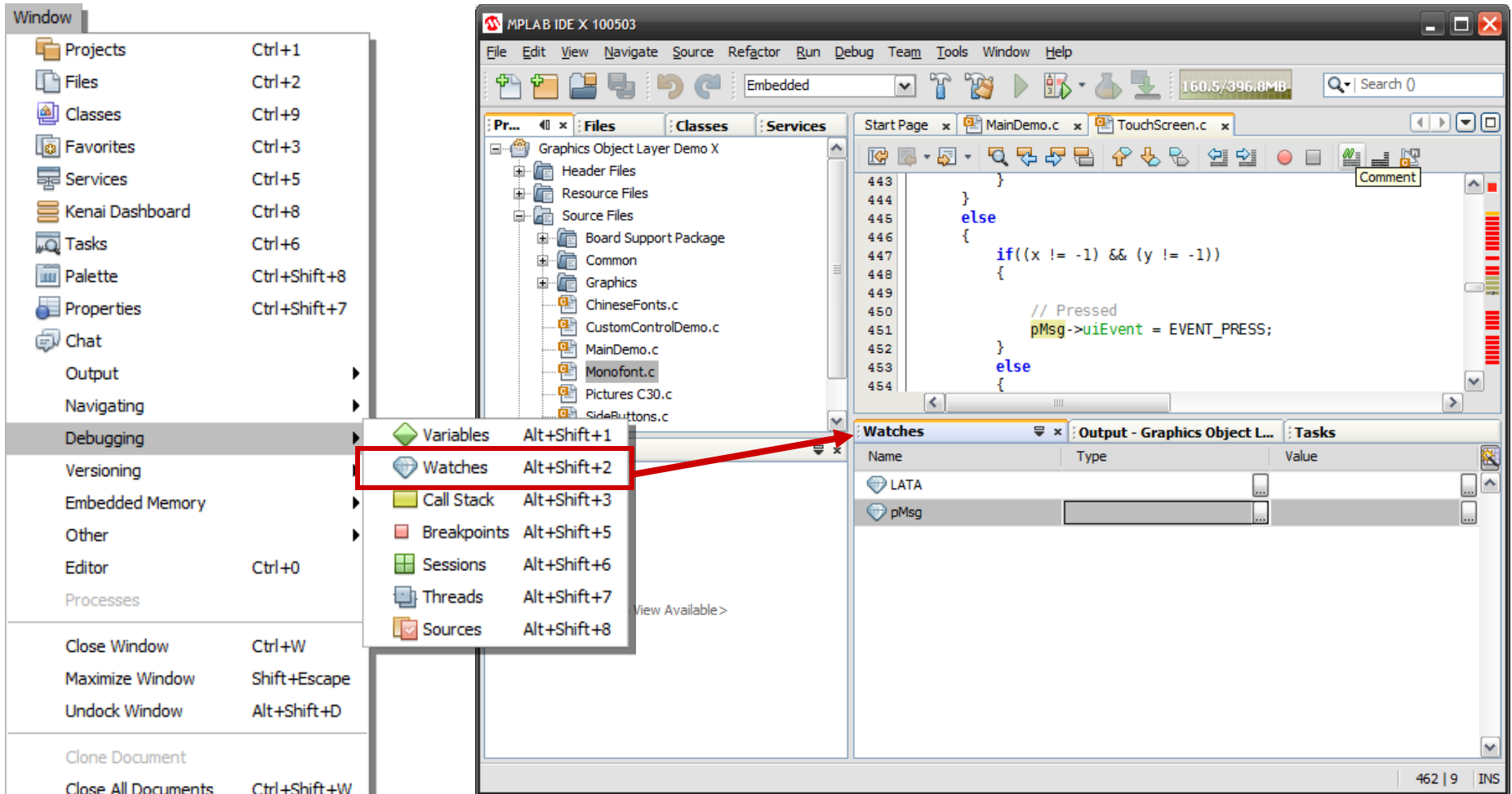
- 1 选择**Window**（窗口）▶ **Debugging**（调试）▶ **Breakpoints**（断点）



- 2 在**Breakpoints**窗口中单击右键可：
 - 创建新断点
 - 使能所有断点
 - 禁止所有断点
 - 删除所有断点
 - 组合断点（允许按组使能/禁止）
 - 更改断点设置

观察

如何显示Watches（观察）选项卡



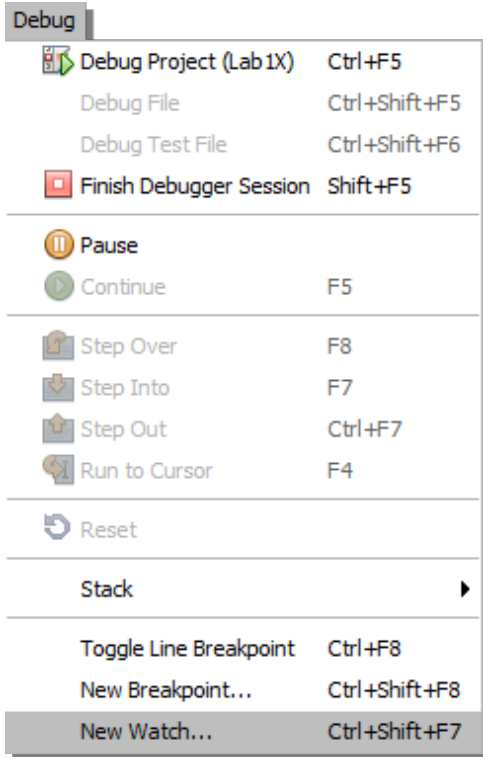
The screenshot shows the MPLAB IDE interface. The 'Window' menu is open, and the 'Debugging' option is selected. The 'Watches' option is highlighted with a red box and a red arrow pointing to the 'Watches' window in the IDE. The 'Watches' window displays a table with columns for Name, Type, and Value. The table contains two entries: LATA and pMsg.

Name	Type	Value
LATA		
pMsg		

从菜单中选择：
Window ▶ Debugging ▶ Watches（观察）

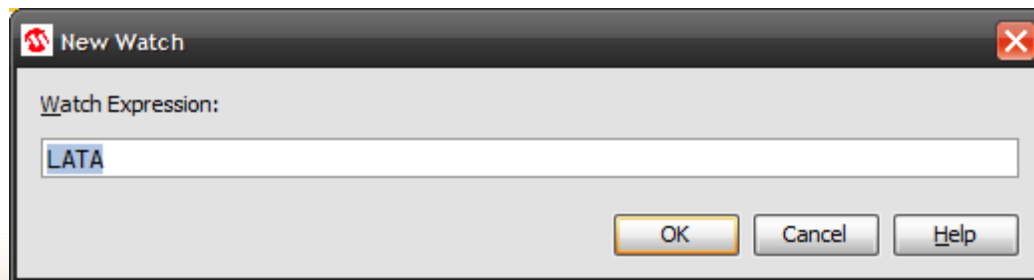
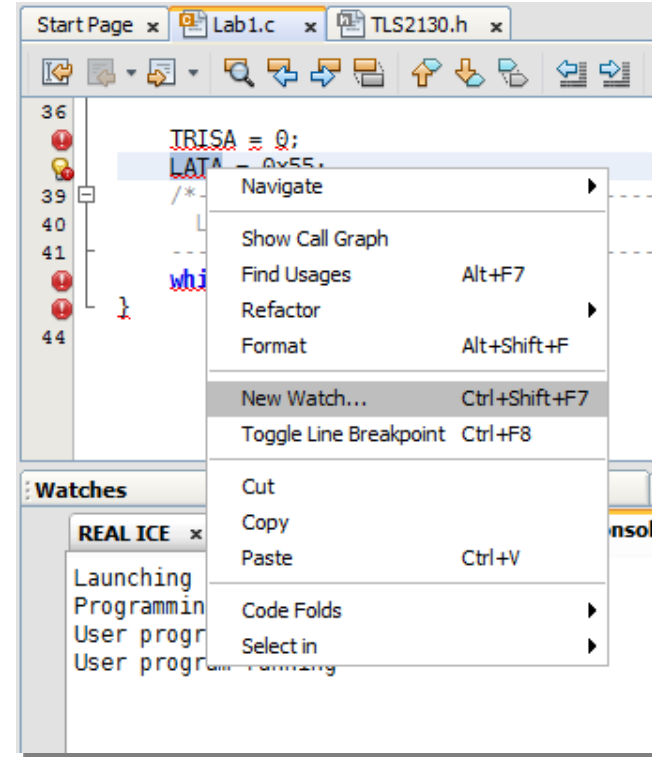
观察

如何向Watches选项卡中添加变量
















- 1 (可选) 在编辑器中高亮选中变量
- 2 在编辑器中单击右键或从菜单中选择:
Debug ▶ New Watch...
(新建观察变量...)
- 3 输入变量名或接受所显示的名称, 并单击

OK



观察

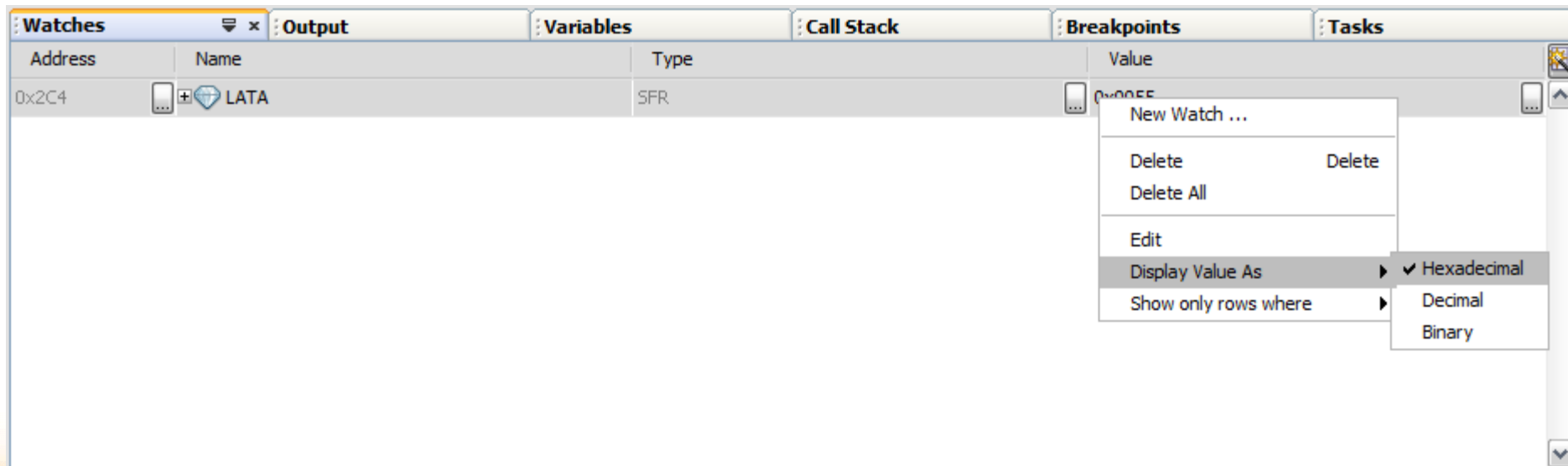
Watches: 结构体和联合体支持

Watches		Output	Variables
Addr...	Name	Type	Value
0x2C4	 LATA	SFR	0x0055
0x2C4	 LATA0	LATA<0>	0x01
0x2C4	 LATA1	LATA<1>	0x00
0x2C4	 LATA2	LATA<2>	0x01
0x2C4	 LATA3	LATA<3>	0x00
0x2C4	 LATA4	LATA<4>	0x01
0x2C4	 LATA5	LATA<5>	0x00
0x2C4	 LATA6	LATA<6>	0x01
0x2C4	 LATA7	LATA<7>	0x00
0x2C4	 LATA9	LATA<9>	0x00
0x2C4	 LATA10	LATA<10>	0x00
0x2C4	 LATA14	LATA<14>	0x00
0x2C4	 LATA15	LATA<15>	0x00

观察

如何更改观察变量的基数

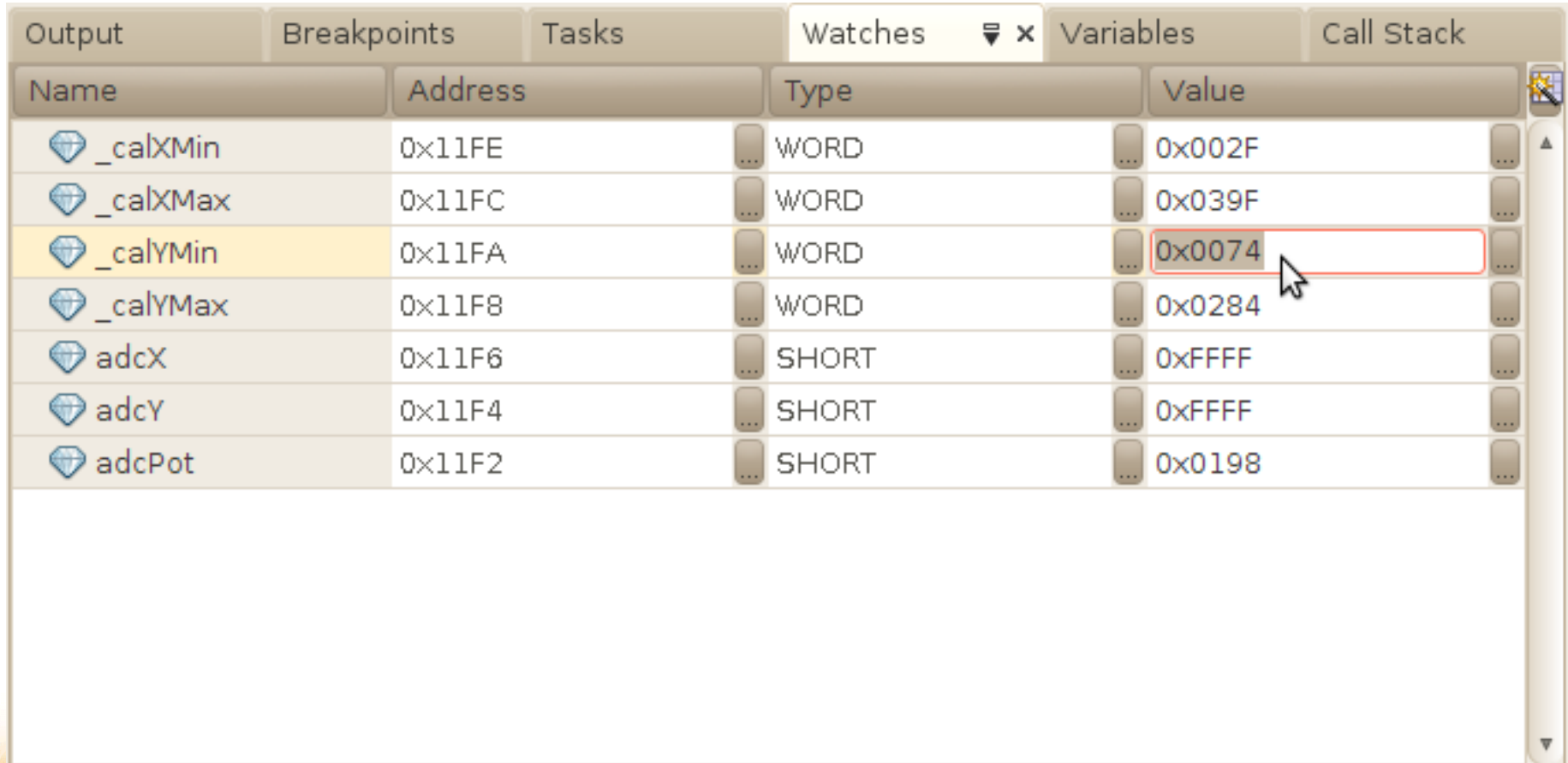
- 1 右键单击该值
- 2 从弹出菜单中选择：
Display Value As（值的显示形式）▶ *基数*










观察

如何更改观察变量的值

- 1 双击已有值
- 2 输入新值，在完成时按下**Enter**键

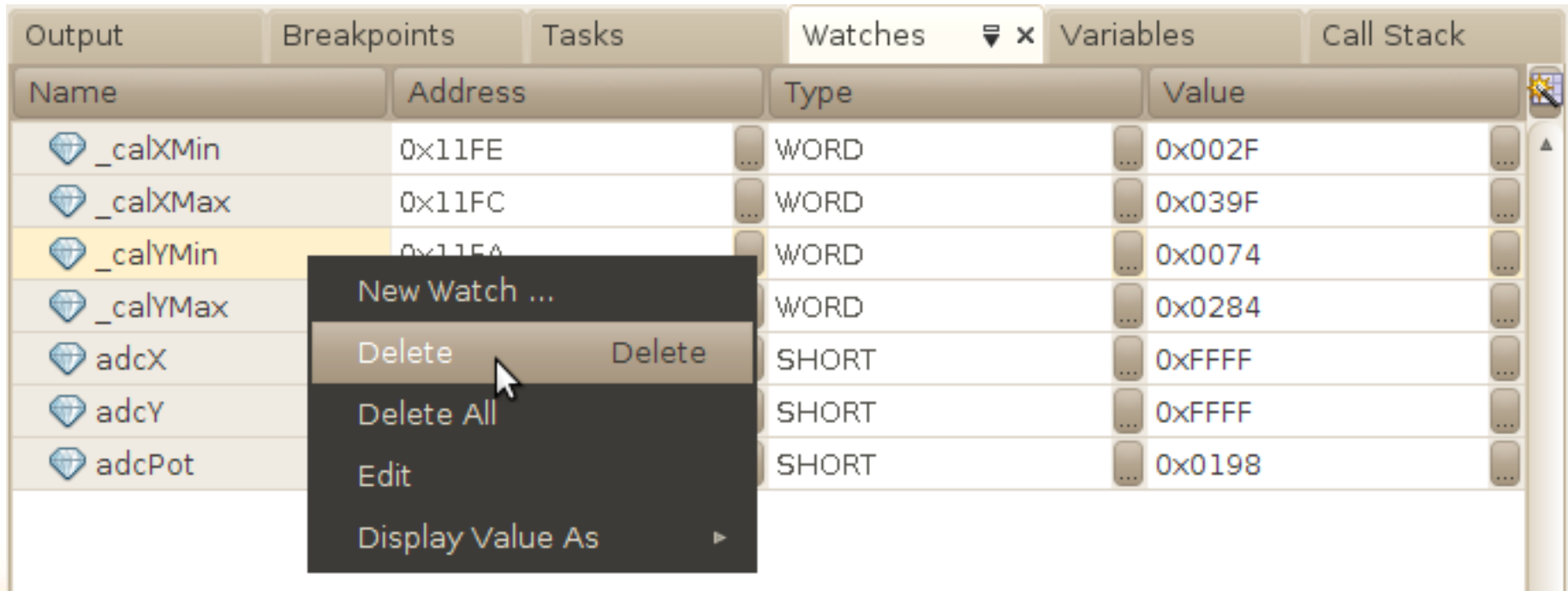


Output	Breakpoints	Tasks	Watches	Variables	Call Stack
Name	Address	Type	Value		
 _calXMin	0x11FE	WORD	0x002F		
 _calXMax	0x11FC	WORD	0x039F		
 _calYMin	0x11FA	WORD	0x0074		
 _calYMax	0x11F8	WORD	0x0284		
 adcX	0x11F6	SHORT	0xFFFF		
 adcY	0x11F4	SHORT	0xFFFF		
 adcPot	0x11F2	SHORT	0x0198		

观察

如何删除观察变量

- 选择观察变量并单击 ，或者右键单击变量并从弹出菜单中选择 **Delete**（删除）



The screenshot shows a debugger interface with the 'Watches' tab selected. A table lists several watch variables. The variable '_calYMin' is selected, and a context menu is open over it, showing options like 'New Watch ...', 'Delete', 'Delete All', 'Edit', and 'Display Value As'. The 'Delete' option is highlighted by the mouse cursor.

Name	Address	Type	Value
_calXMin	0x11FE	WORD	0x002F
_calXMax	0x11FC	WORD	0x039F
_calYMin	0x11FA	WORD	0x0074
_calYMax		WORD	0x0284
adcX		SHORT	0xFFFF
adcY		SHORT	0xFFFF
adcPot		SHORT	0x0198

观察

如何对观察变量进行排序

单击列标题...

1 默认
(字段按数字顺序排列)

Address	Name	T
0x2C4	LATA	SFF
0x2C4	LATA0	
0x2C4	LATA1	
0x2C4	LATA2	
0x2C4	LATA3	
0x2C4	LATA4	
0x2C4	LATA5	
0x2C4	LATA6	

2 第一次单击:
升序 (按字母顺序排列)

Address	Name	T
0x2C4	LATA	SFF
0x2C4	LATA0	
0x2C4	LATA1	
0x2C4	LATA10	
0x2C4	LATA14	
0x2C4	LATA15	
0x2C4	LATA2	
0x2C4	LATA3	

3 第二次单击:
降序 (按字母顺序排列)

Address	Name	T
0x2C4	LATA	SFF
0x2C4	LATA9	
0x2C4	LATA7	
0x2C4	LATA6	
0x2C4	LATA5	
0x2C4	LATA4	
0x2C4	LATA3	
0x2C4	LATA2	

4 第三次单击:
默认

Address	Name	T
0x2C4	LATA	SFF
0x2C4	LATA0	LAT
0x2C4	LATA1	LAT
0x2C4	LATA2	LAT
0x2C4	LATA3	LAT
0x2C4	LATA4	LAT
0x2C4	LATA5	LAT
0x2C4	LATA6	LAT

存储器窗口

如何查看嵌入式存储器



从菜单中选择:

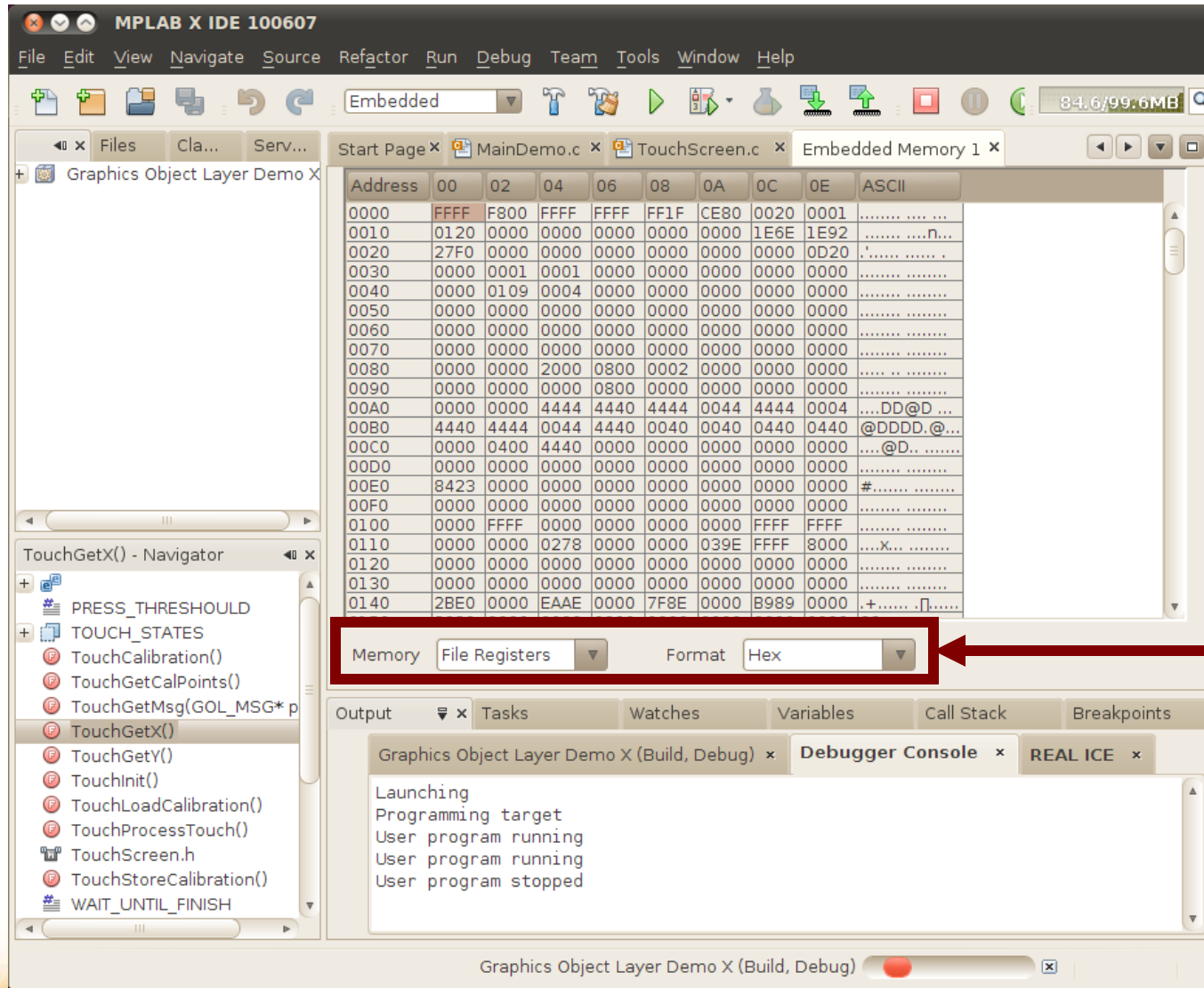
Window ▶ Embedded Memory (嵌入式存储器) ▶ Mdb Memory View *n* (Mdb存储器视图*n*)

最多可有4个窗口，是以下的任意组合:

- 文件寄存器
- SFR
- 程序
- 配置

存储器窗口

如何选择要显示的嵌入式存储器类型



从**Memory**（存储器）组合框中选择要显示的存储空间，然后从**Format**（格式）组合框中选择显示格式。

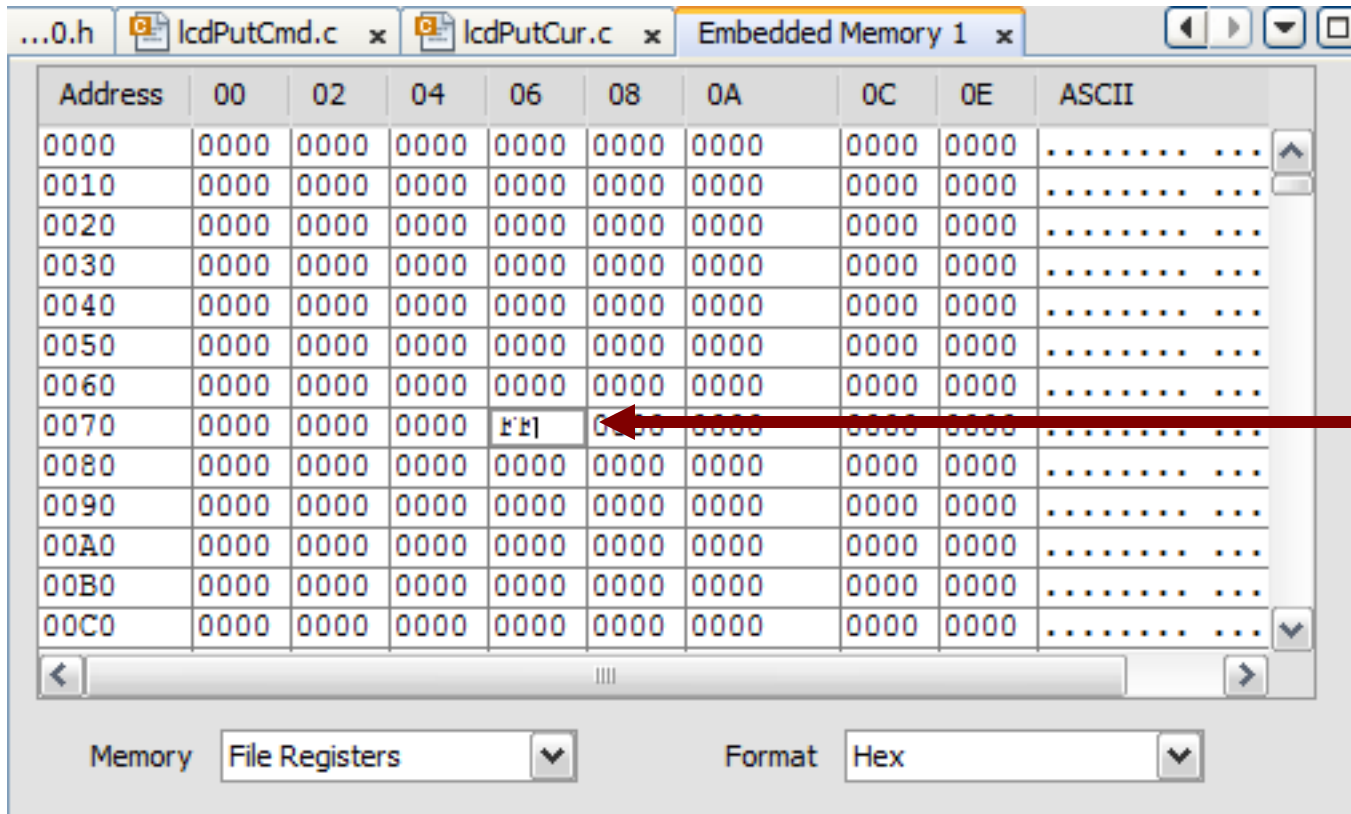
存储器窗口

显示选项

选项	值	说明
Format (格式)	Data (数据) Code (代码)	数据存储器 (RAM) 程序存储器 (ROM)
Memory (存储器)	RAM Physical (物理RAM) Flash Physical (物理闪存) Peripheral (外设) CPU Configuration Bits (配置位)	器件上的所有RAM存储器 器件上的所有闪存 所有SFR 所有CPU存储器 所有配置寄存器

存储器窗口

如何更改存储单元的值

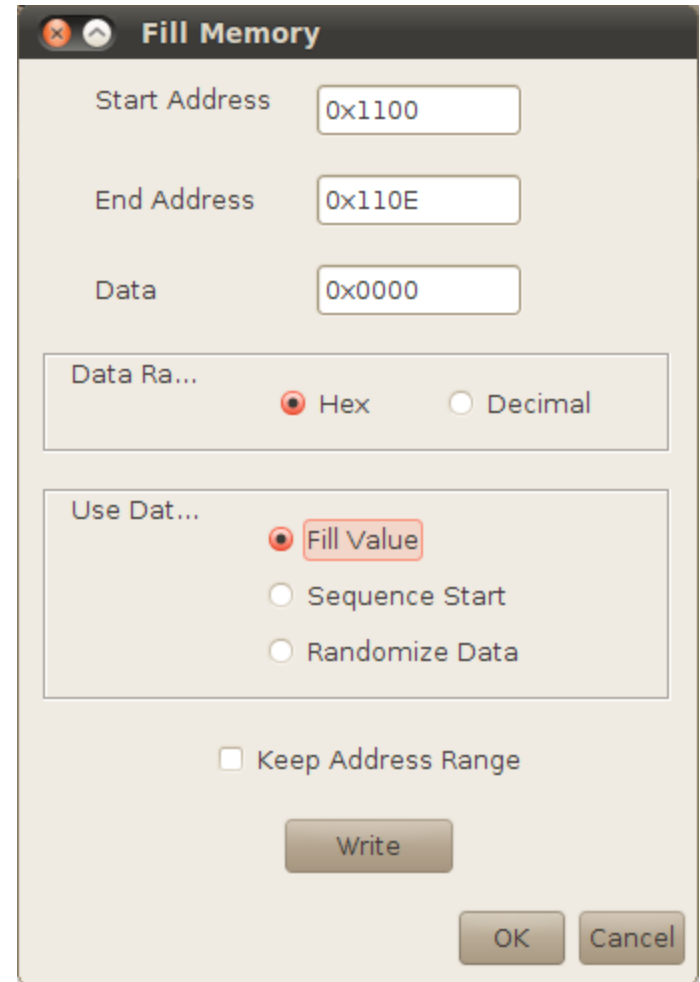


双击单元格并
输入新值——
在完成时按
Enter键

存储器窗口

如何填充存储器

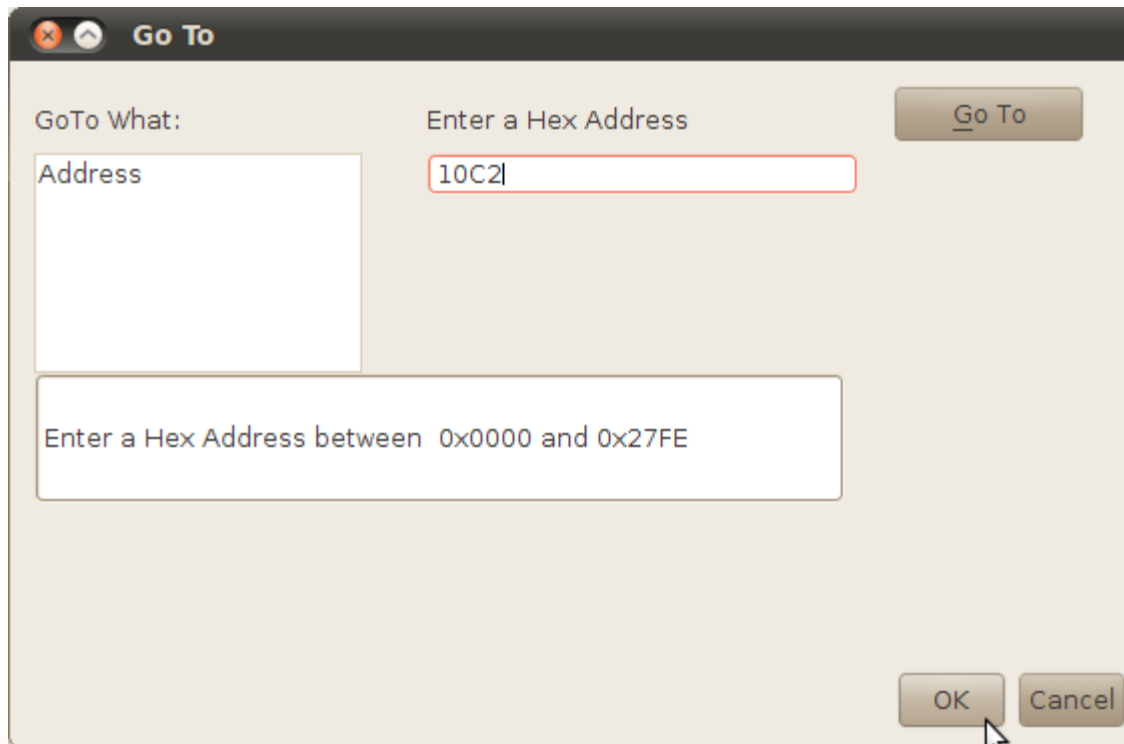
- 在存储器窗口中单击右键，并从弹出菜单中选择**Fill Memory...**（填充存储器...）



存储器窗口

如何快速跳转至特定地址

- 在存储器窗口中单击右键，并从弹出菜单中选择**Go To...**（转至...）



调用堆栈

如何查看调用堆栈

- 选择 **Window** ▶ **Debugging** ▶ **Call Stack**
(调用堆栈)





MICROCHIP 2010

MASTERS Conference

如何进行项目导航

搜索、编辑器功能和导航菜单

编辑器工具栏

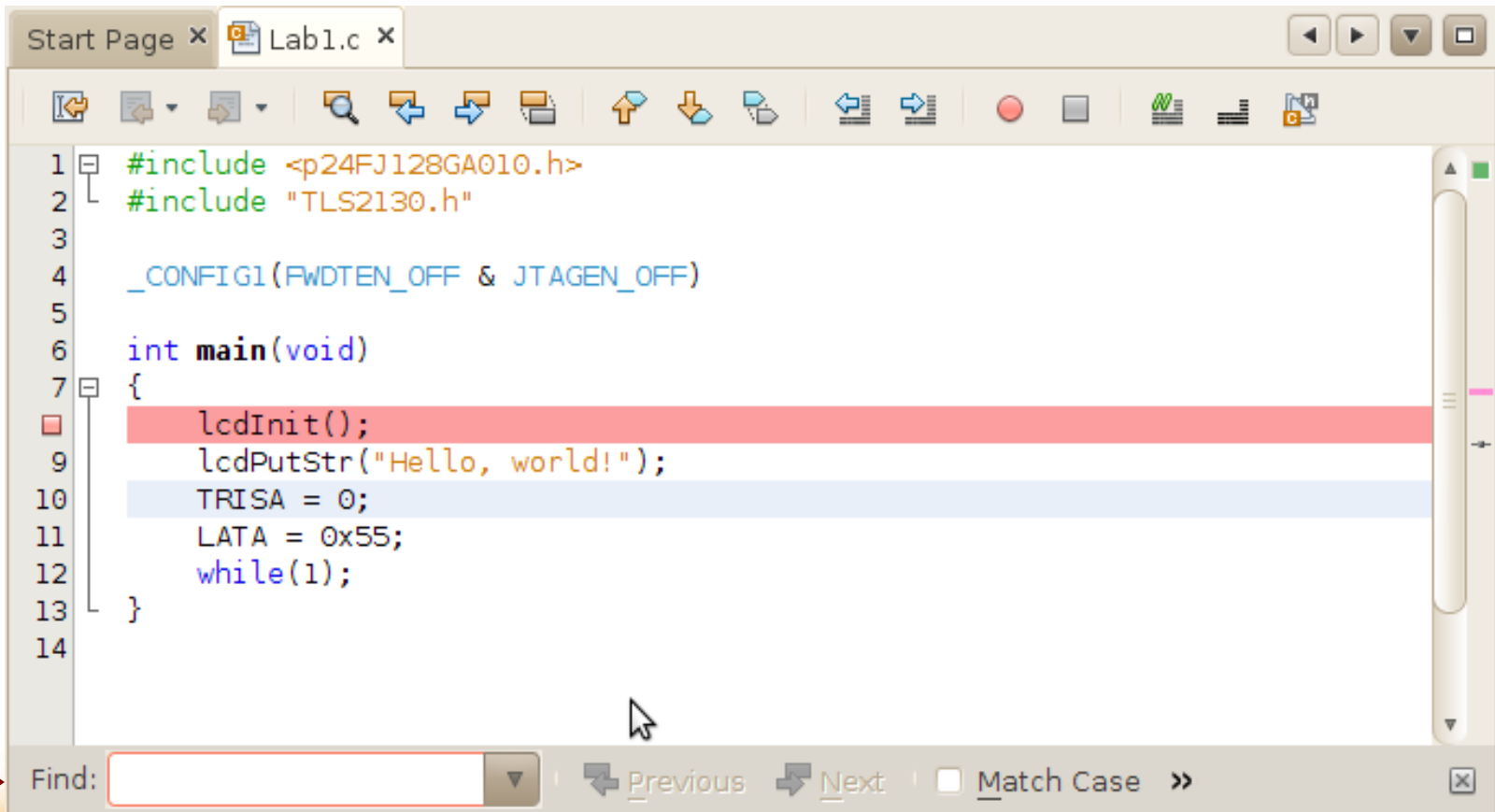


-  **Last Edit** (最近一处编辑)
——在编辑记录之间循环
-  **Previous Edit** (上一处编辑)
-  **Next Edit** (下一处编辑)
-  **Find Selection** (查找选择项)
-  **Find Previous Occurrence** (查找上一处)
-  **Find Next Occurrence** (查找下一处)
-  **Toggle Highlight Search** (切换高亮搜索项)
-  **Previous Bookmark** (上一个书签)
-  **Next Bookmark** (下一个书签)
-  **Toggle Bookmark** (切换书签)
-  **Shift Line Left** (左移代码行)
(减少缩进)
-  **Shift Line Right** (右移代码行)
(增加缩进)
-  **Start Macro Recording**
(启动宏录制)
-  **Stop Macro Recording**
(停止宏录制)
-  **Comment** (注释)
-  **Uncomment** (取消注释)
-  **Go to Header/Source**
(转至头/源文件)

* Next Edit和Previous Edit按钮在使用**Navigate** (导航) ▶ **Go to X...** (转至X...) 之后激活

如何在文件中进行搜索

- 对于在编辑器中打开的源文件，从菜单中选择 **Edit（编辑）** ▶ **Find（查找）**



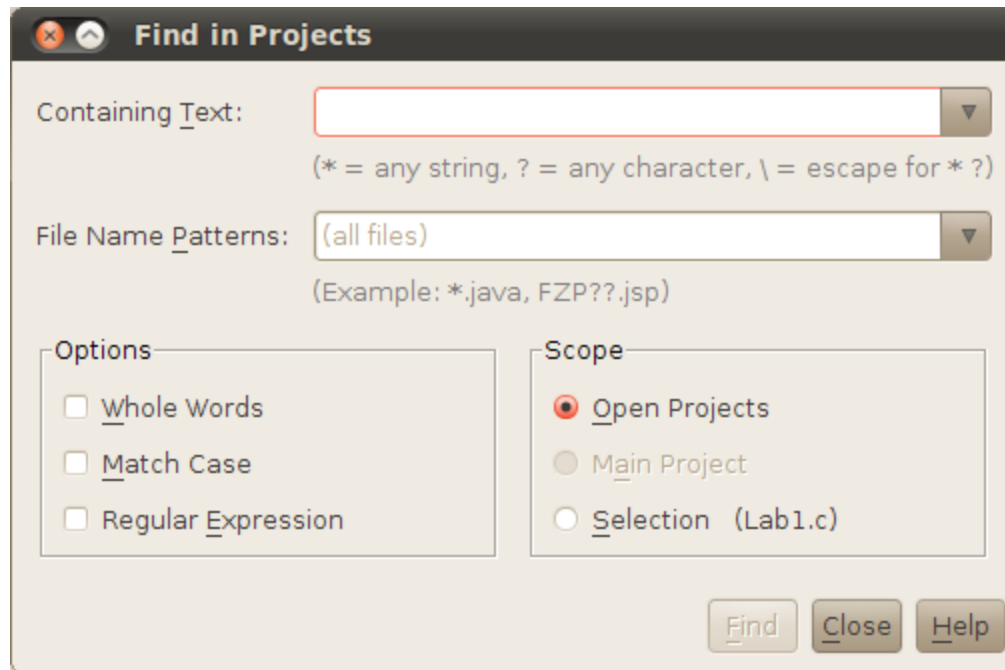
The screenshot shows an IDE window with a tab labeled 'Lab1.c'. The code editor contains the following C code:

```
1 #include <p24FJ128GA010.h>
2 #include "TLS2130.h"
3
4 _CONFIG1(FWDTEN_OFF & JTAGEN_OFF)
5
6 int main(void)
7 {
8     lcdInit();
9     lcdPutStr("Hello, world!");
10    TRISA = 0;
11    LATA = 0x55;
12    while(1);
13 }
14
```

At the bottom of the window, there is a search bar labeled 'Find:' with a text input field, a dropdown arrow, and buttons for 'Previous', 'Next', 'Match Case', and a close button. A red arrow points to the 'Find:' label.

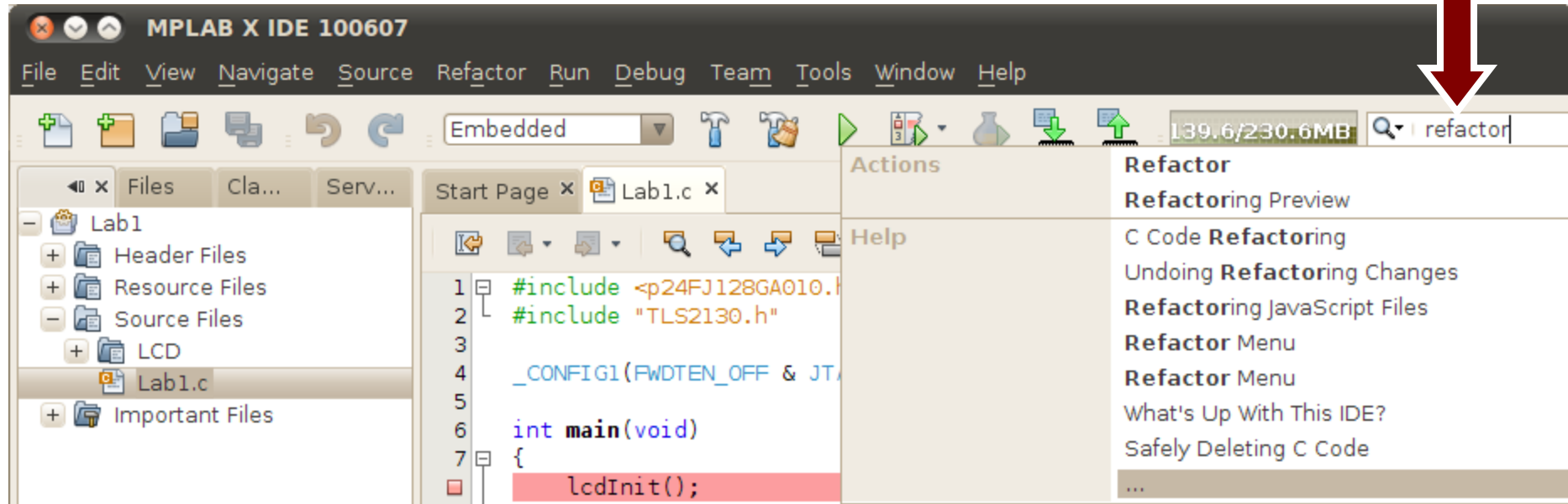
如何在项目中进行搜索

- 从菜单中选择 **Edit ► Find in Projects...** (在项目中查找...)



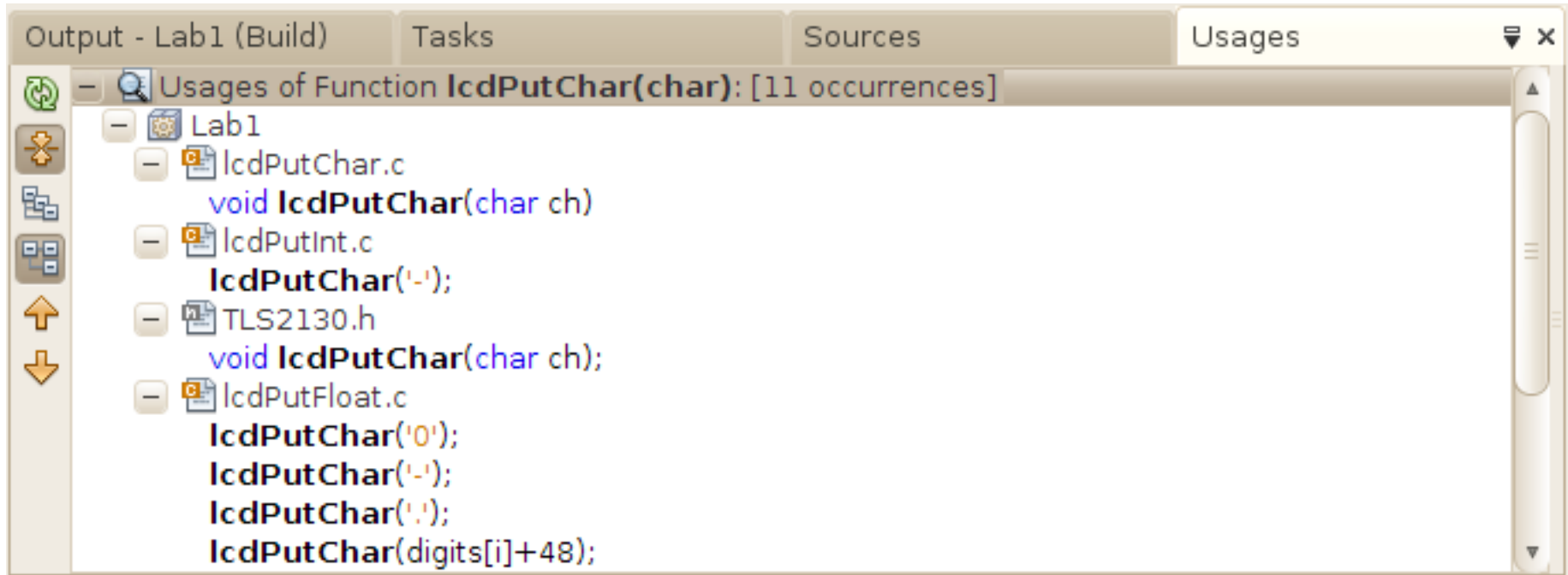
如何搜索文档

- 在Quick Search（快速搜索）框中输入文本



如何查找用法

- 高亮显示某个函数或变量
- 从菜单中选择 **Edit ▶ Find Usages**（查找用法）



Output - Lab1 (Build) | Tasks | Sources | Usages

Usages of Function `lcdPutChar(char)`: [11 occurrences]

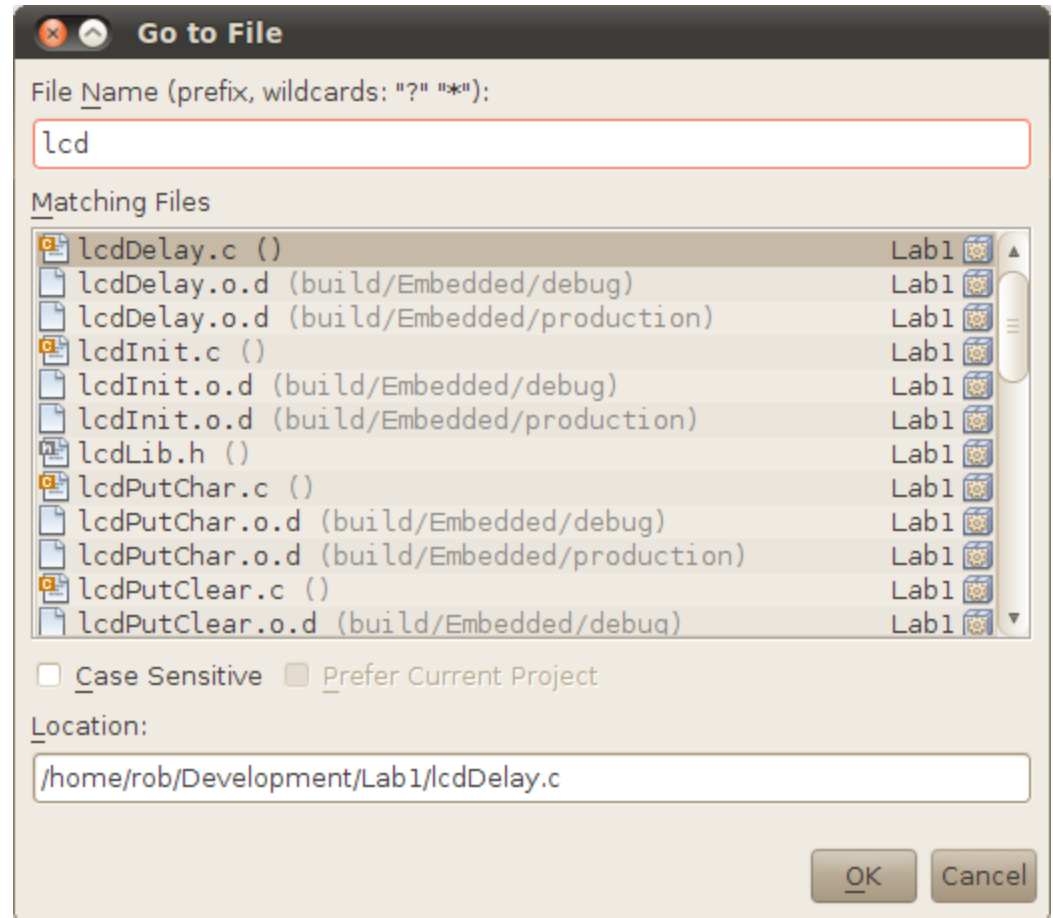
- Lab1
 - lcdPutChar.c
 - `void lcdPutChar(char ch)`
 - lcdPutInt.c
 - `lcdPutChar(' ');`
 - TLS2130.h
 - `void lcdPutChar(char ch);`
 - lcdPutFloat.c
 - `lcdPutChar('0');`
 - `lcdPutChar(' ');`
 - `lcdPutChar(' ');`
 - `lcdPutChar(digits[i]+48);`

编辑器

Go To File...

Navigate	Source	Refactor	Run	Debug	Team
Go to File...				Alt+Shift+O	
Go to Type...				Ctrl+O	
Go to Symbol...				Ctrl+Alt+Shift+O	
Go to Previous Document					

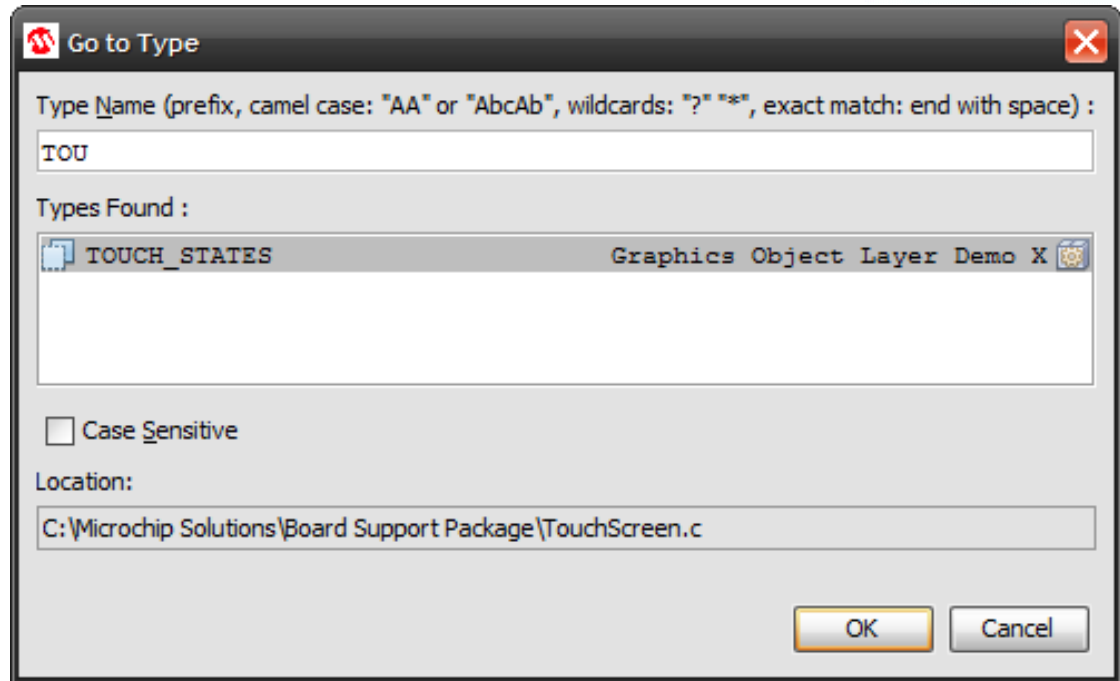
- 有时只会查找到目标文件（需要查看一下...）



编辑器

Go To Type...

Navigate	Source	Refactor	Run	Debug	Team
Go to File...			Alt+Shift+O		
Go to Type...			Ctrl+O		
Go to Symbol...			Ctrl+Alt+Shift+O		
Go to Previous Document					

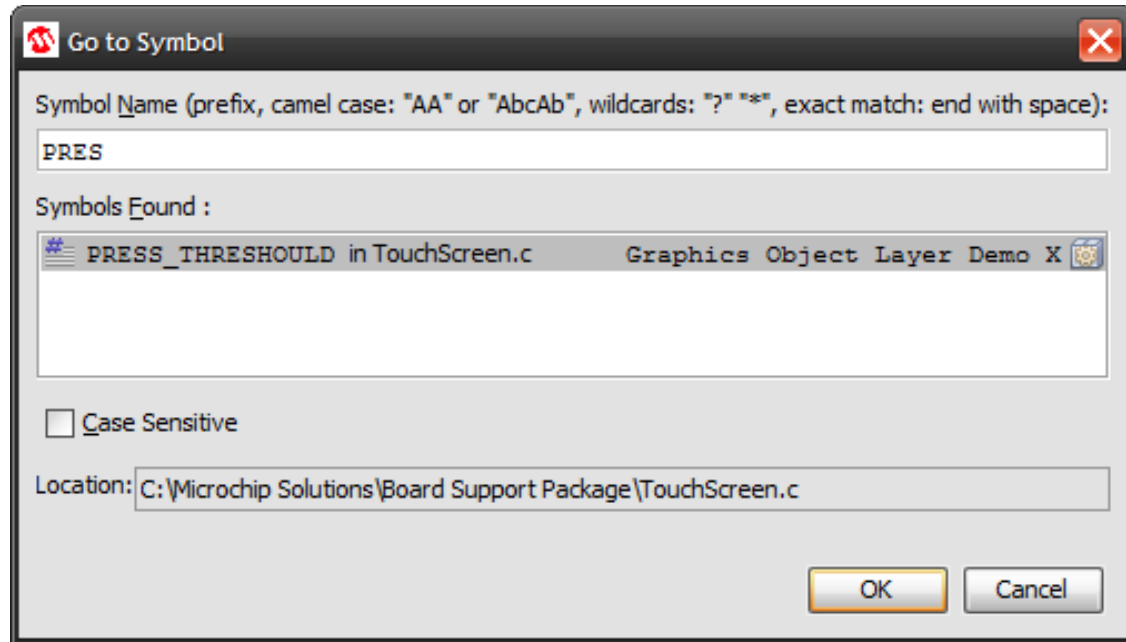


- 查找类型定义，并且在对话框列表中双击可跳转至这些定义
 - C语言中的enum、struct和union类型

编辑器

Go To Symbol...

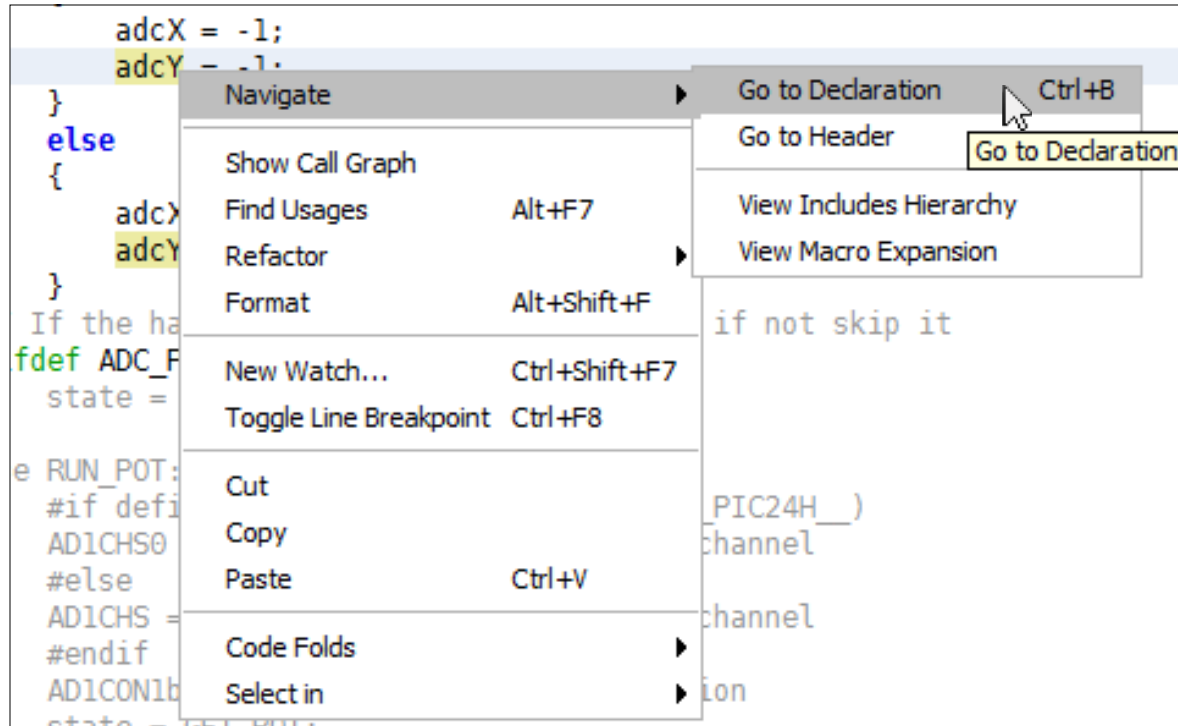
Navigate	Source	Refactor	Run	Debug	Team
Go to File...			Alt+Shift+O		
Go to Type...			Ctrl+O		
Go to Symbol...			Ctrl+Alt+Shift+O		
Go to Previous Document					



- 查找符号定义，并且在对话框列表中双击可跳转至这些定义

编辑器

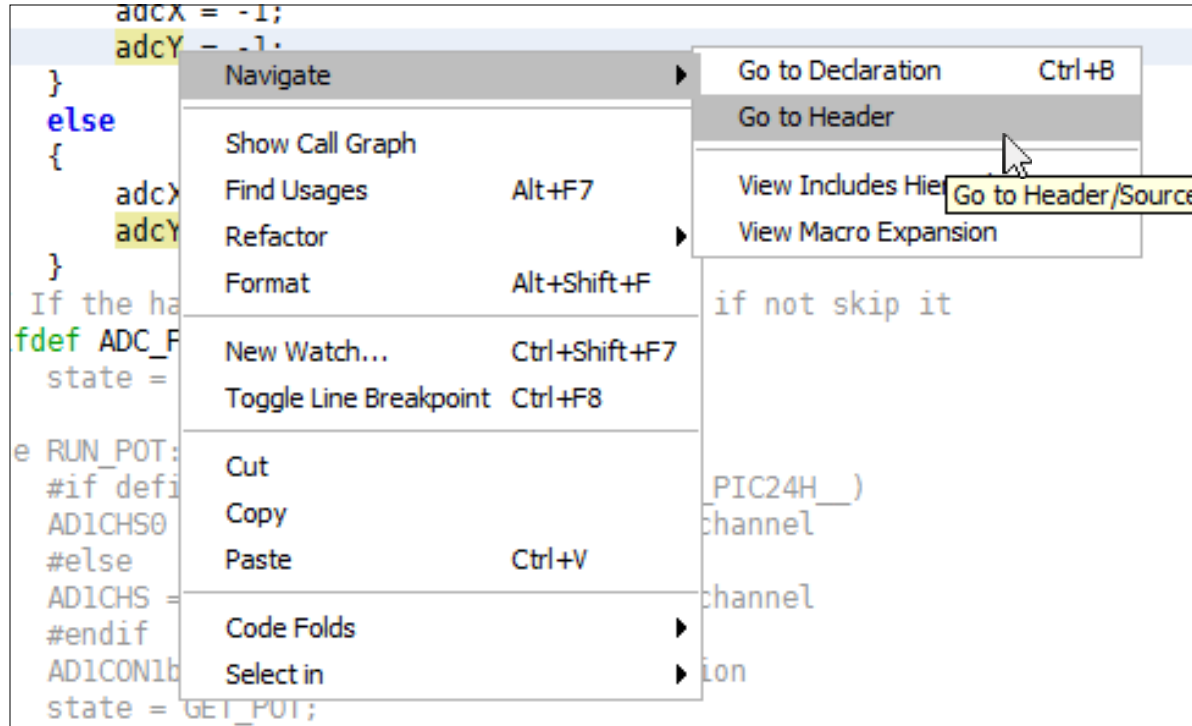
Go to Declaration...



- 高亮选中变量或函数名，单击右键，从弹出菜单中选择**Navigate ▶ Go to Declaration...**

编辑器

Go to Header...










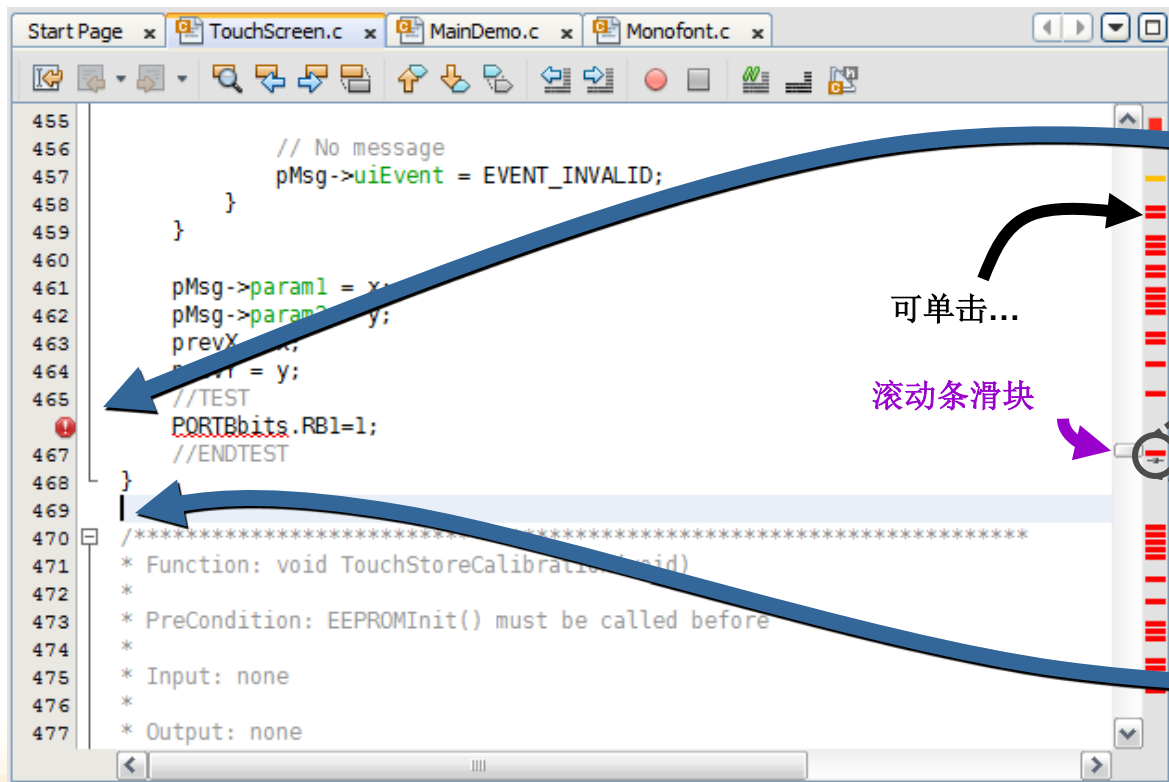
- 转至基本名称与当前源文件相同的头文件（如果存在）
例如，从 ***MyFile.c*** 转至 ***MyFile.h***

编辑器

导航：错误横条

- 代表文件的完整长度
- 显示错误、警告和出现当前高亮项的相对位置

颜色码	
	警告
	错误
	高亮项出现
	断点
	书签
	建议
	多项



可单击...

滚动条滑块

错误横条

错误

光标位置

编辑器

导航至错误和警告

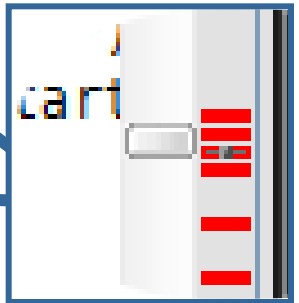
弹出框会显示对应于鼠标指针（箭头）所指图标的错误

```
585 WAIT_UNTIL_FINISH(OutTextXY((GetMaxX()-GetTextWidth((XCHAR *)
586 textStar
587 WAIT_UNTIL_FINISH(OutTextXY((GetMaxX()-GetTextWidth((XCHAR *)
588 textStar
589 WAIT_UNTIL_FINISH(OutTextXY((GetMaxX()-GetTextWidth((XCHAR *)
590 textStar
591 Unable to resolve identifier SetColor Y((GetMaxX()-GetTextWidth((XCHAR *)
592 textStar
593 Unable to resolve identifier BRIGHTRED Y((GetMaxX()-GetTextWidth((XCHAR *)
594 textStar
595 SetColor(BRIGHTRED);
596 WAIT_UNTIL_FINISH(OutTextXY((GetMaxX()-GetTextWidth((XCHAR *)
597 textStar
598 WAIT_UNTIL_FINISH(OutTextXY((GetMaxX()-GetTextWidth((XCHAR *)
599 textStar
600
601 // Wait for touch
602 do
603 {
604     x = ADCGetX();
605     y = ADCGetY();
606 } while((y == -1) || (x == -1));
```

错误横条

弹出框会显示对应于鼠标指针（手型）所指横条的错误

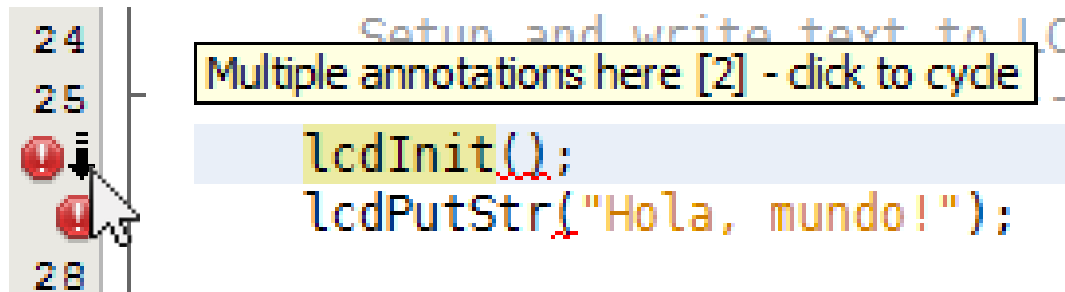
Unable to resolve identifier GetMaxY



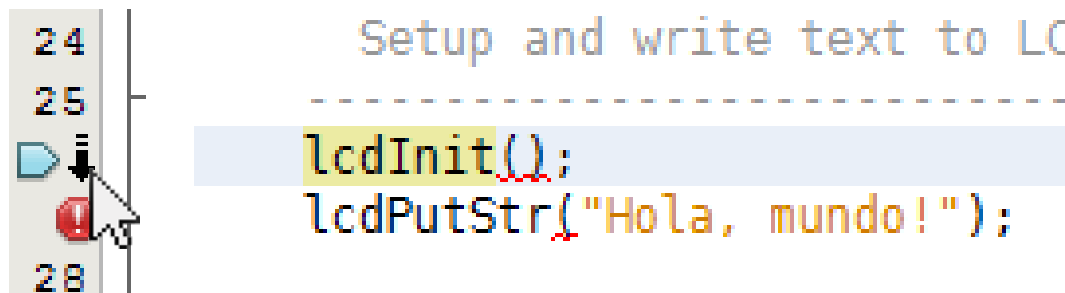
编辑器

导航：图符页边

错误和书签图符处于同一行



24 Setup and write text to LCD
25 Multiple annotations here [2] - click to cycle
26 lcdInit();
27 lcdPutStr("Hola, mundo!");
28



24 Setup and write text to LCD
25 -----
26 lcdInit();
27 lcdPutStr("Hola, mundo!");
28

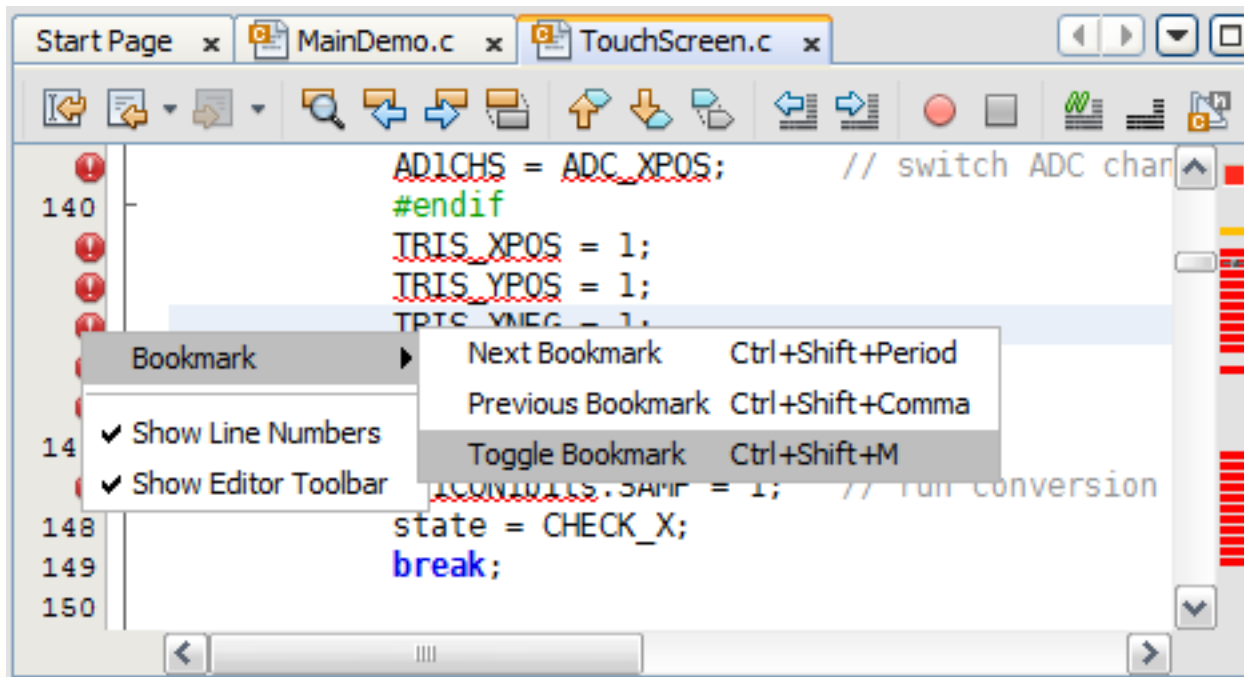
如果代码行对应多个图符，则会显示一个图符，并在图符右侧显示一个向下的黑色小箭头。

单击箭头将会循环显示这些图符。

编辑器

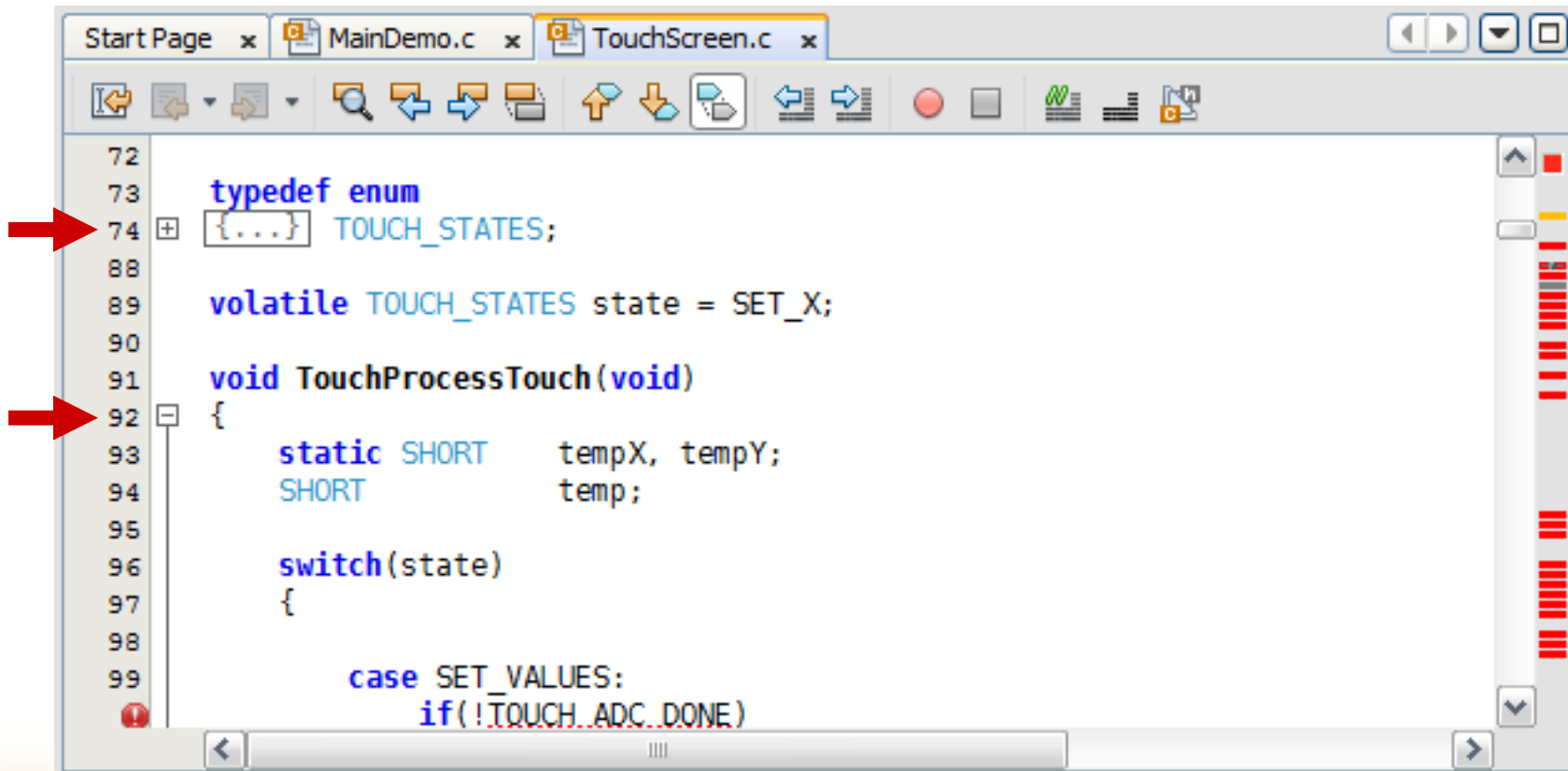
如何设置/清除书签

- 在页边中右键单击行号，并从弹出菜单中选择：
Bookmark (书签) ▶ Toggle Bookmark (切换书签)



编辑器 代码折叠

可通过**Tools ▶ Options ▶ Editor ▶ General**（常规）
使能或禁止

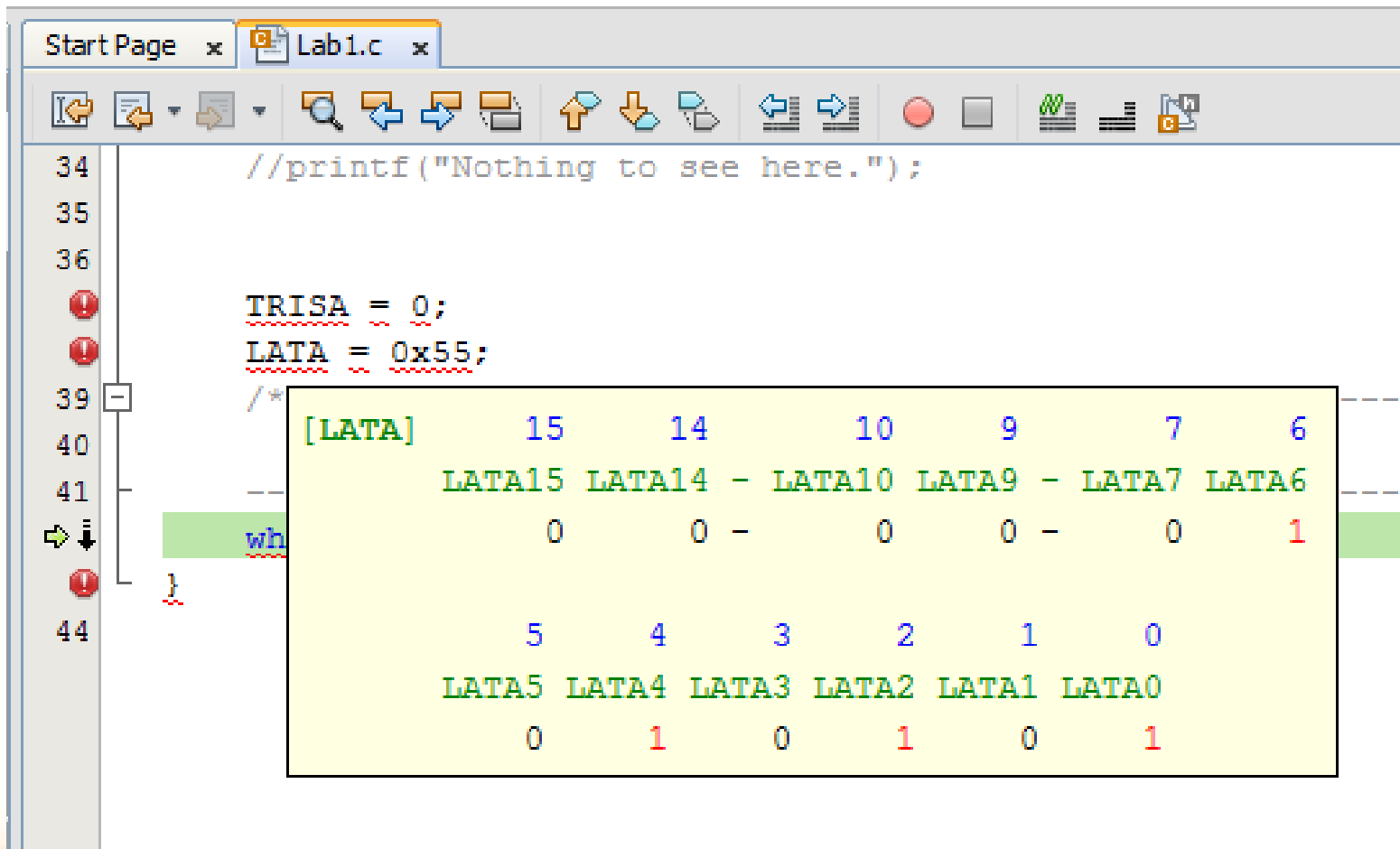


```
72
73 typedef enum
74 { ... } TOUCH_STATES;
88
89 volatile TOUCH_STATES state = SET_X;
90
91 void TouchProcessTouch(void)
92 {
93     static SHORT    tempX, tempY;
94     SHORT          temp;
95
96     switch(state)
97     {
98
99         case SET_VALUES:
100             if(!TOUCH_ADC_DONE)
```

编辑器

如何查看寄存器的位值

■ 将鼠标移到寄存器变量上...



The screenshot shows an IDE window titled 'Lab1.c' with the following code:

```

34 //printf("Nothing to see here.");
35
36
37 TRISA = 0;
38 LATA = 0x55;
39 /*
40 [LATA]
41 LATA15 LATA14 - LATA10 LATA9 - LATA7 LATA6
42 0 0 - 0 0 - 0 1
43
44 5 4 3 2 1 0
45 LATA5 LATA4 LATA3 LATA2 LATA1 LATA0
46 0 1 0 1 0 1

```

The tooltip displays the bit values for the LATA register:

Bit	Value
LATA15	0
LATA14	0
LATA13	-
LATA12	0
LATA11	0
LATA10	0
LATA9	0
LATA8	-
LATA7	0
LATA6	1
LATA5	0
LATA4	1
LATA3	0
LATA2	1
LATA1	0
LATA0	1



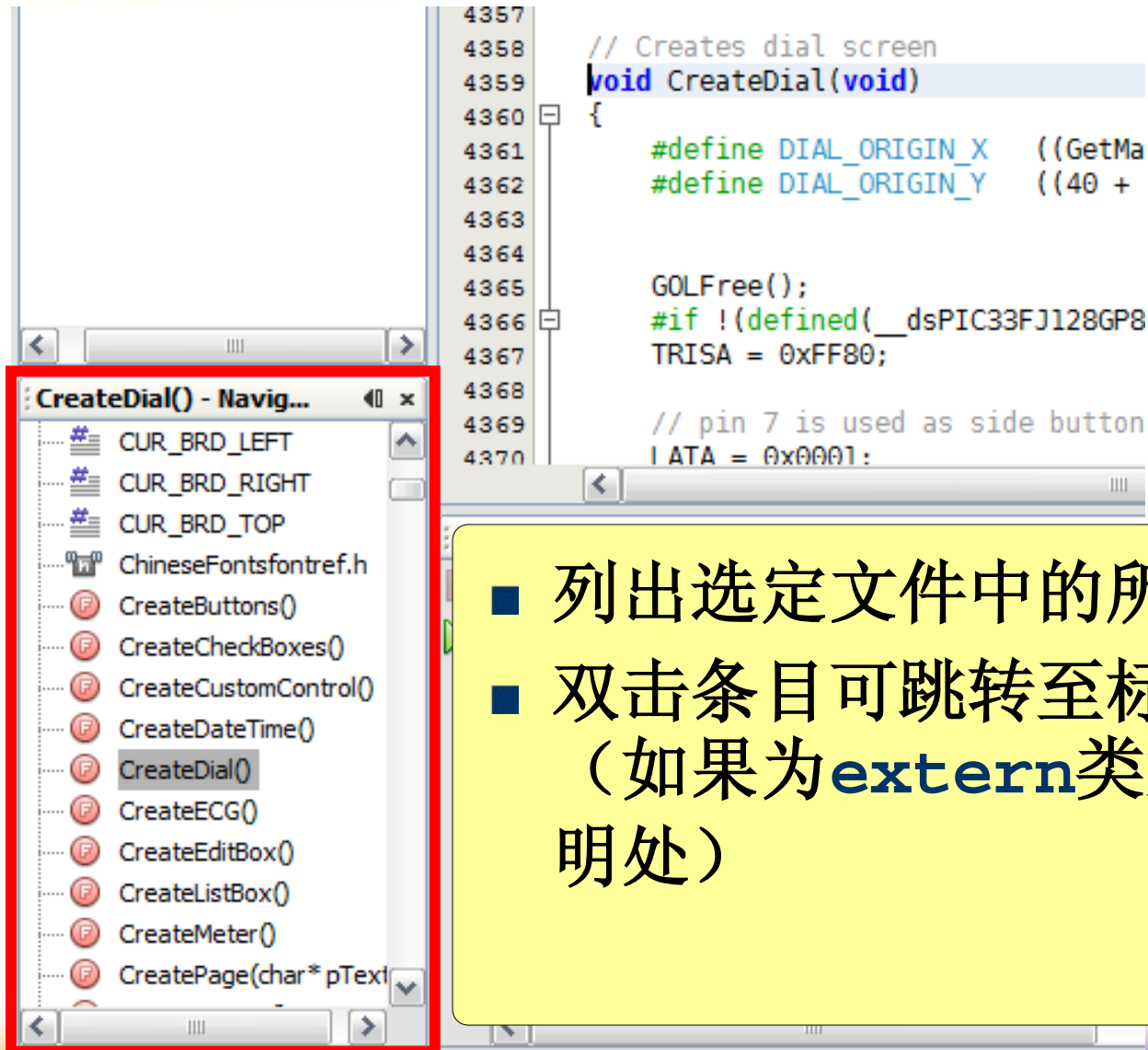
MICROCHIP 2010

MASTERS Conference

如何进行项目导航

如何使用导航器窗口

如何使用导航器

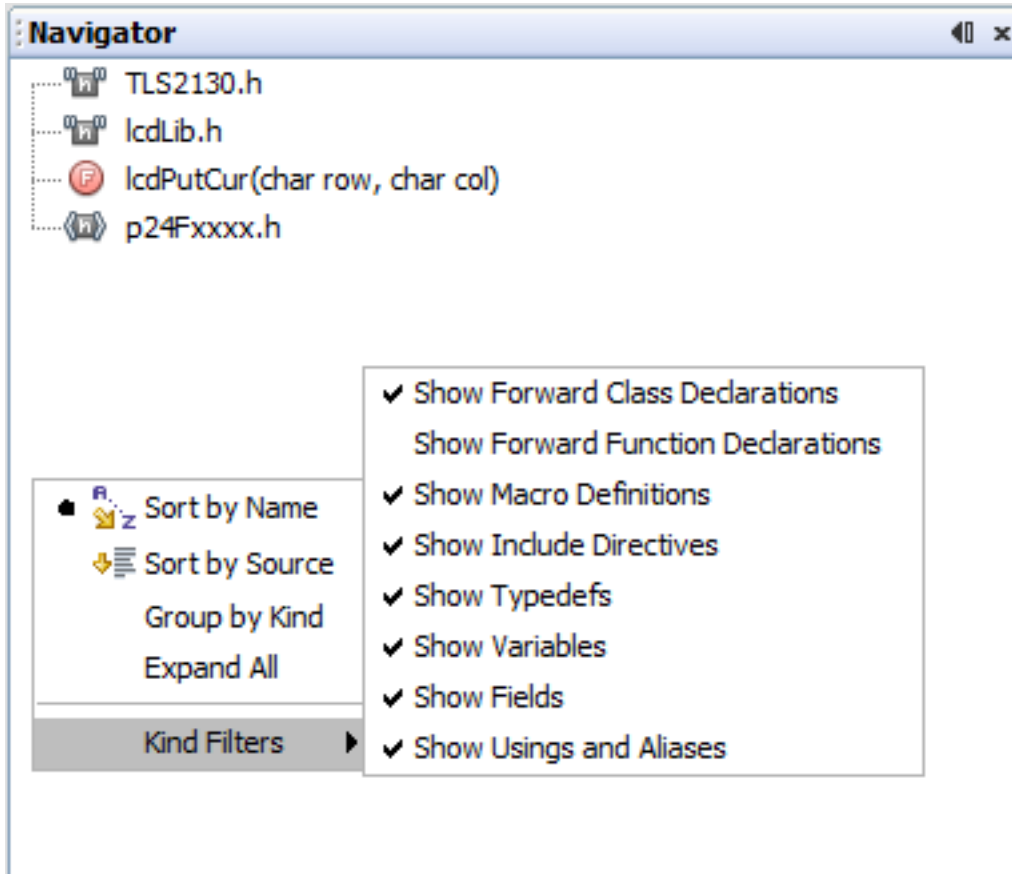


```
4357
4358 // Creates dial screen
4359 void CreateDial(void)
4360 {
4361     #define DIAL_ORIGIN_X ((GetMa
4362     #define DIAL_ORIGIN_Y ((40 +
4363
4364
4365     GOLFree();
4366     #if !(defined(__dsPIC33FJ128GP8
4367     TRISA = 0xFF80;
4368
4369     // pin 7 is used as side button
4370     IATA = 0x0001;
```

■ 列出选定文件中的所有标识符

■ 双击条目可跳转至标识符定义处
(如果为extern类型, 则跳转至声明处)

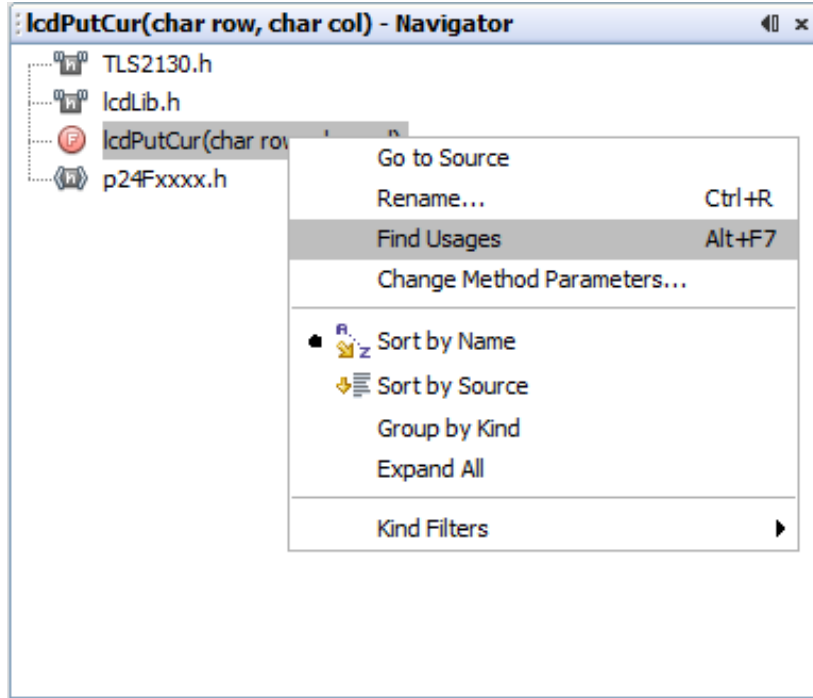
如何使用导航器 过滤器



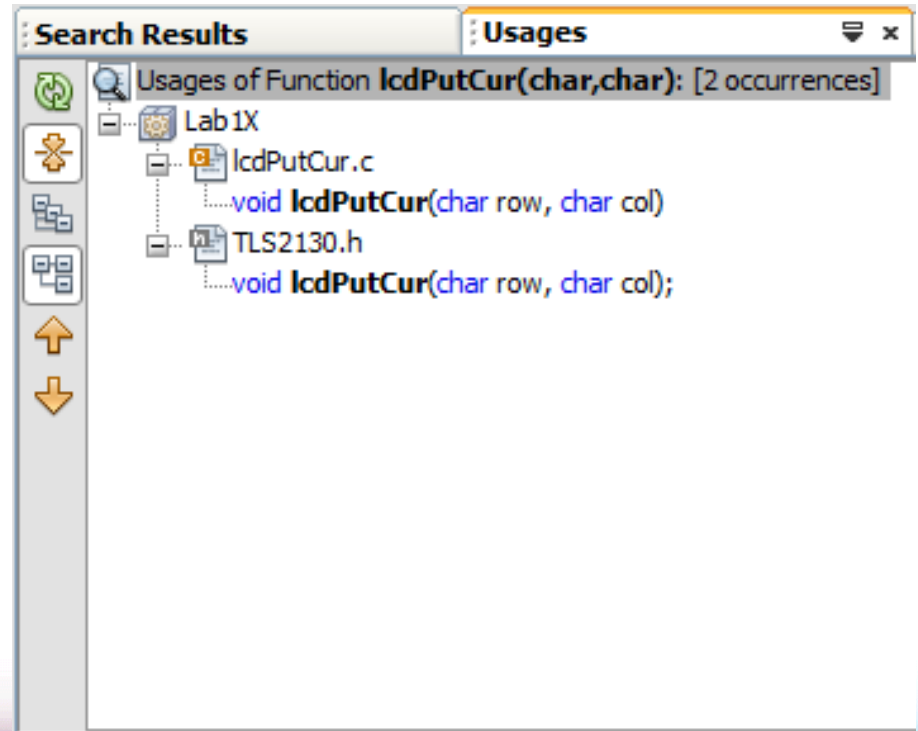
右键单击Navigator（导航器）窗口，并从弹出菜单中选择**Kind Filters（类型过滤器）**，这样可更改在窗口中显示的项

如何使用导航器

查找用法



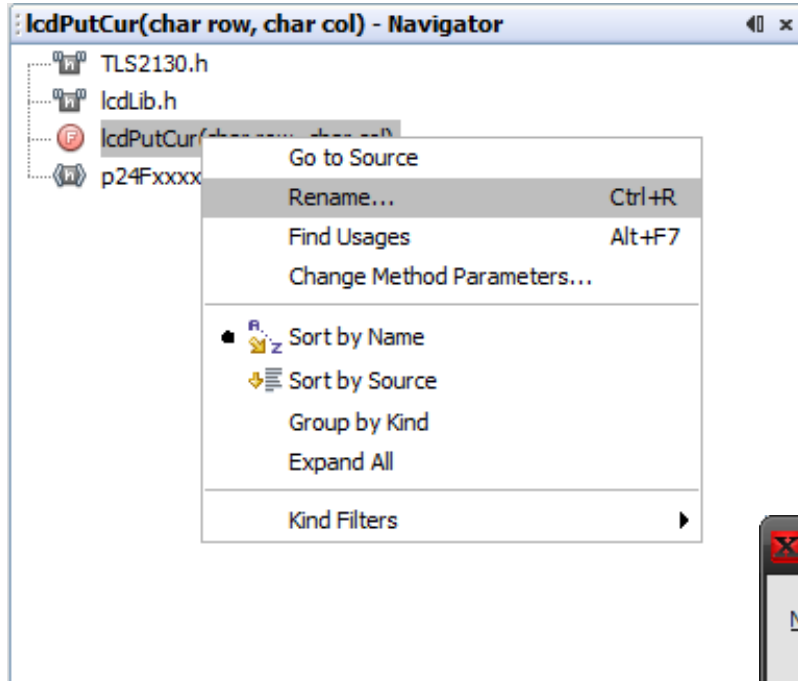
- a** 在导航器中右键单击符号名称，并从弹出菜单中选择 **Find Usages**



- b** 结果会显示在输出窗口的 **Usages (用法)** 选项卡下（双击它们可跳转至使用它们的源文件）

如何使用导航器

如何重构符号名称



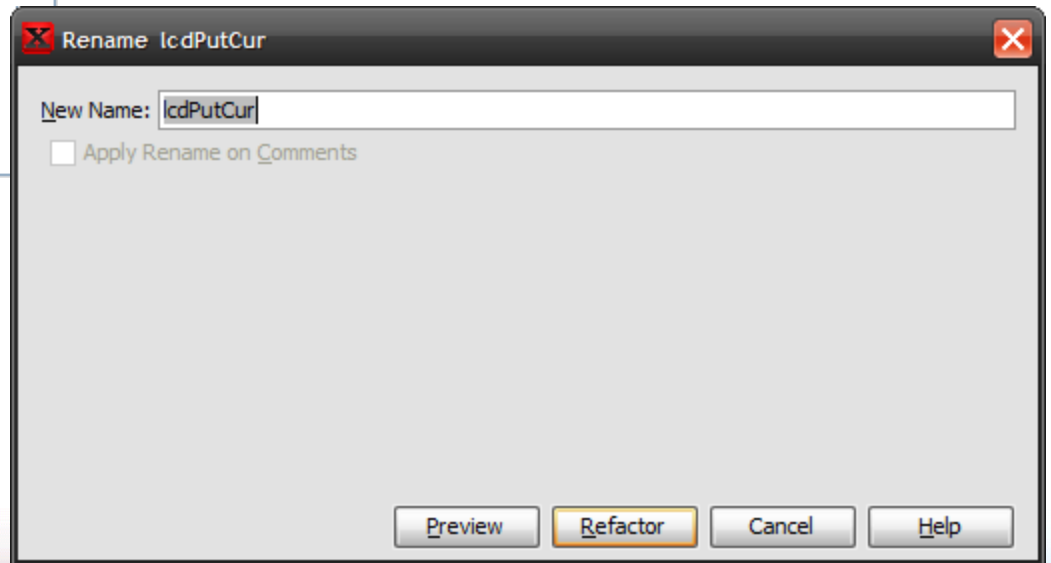
a 在导航器中右键单击符号名称，并从弹出菜单中选择 **Rename...**

b 输入新名称
并单击

Refactor

或

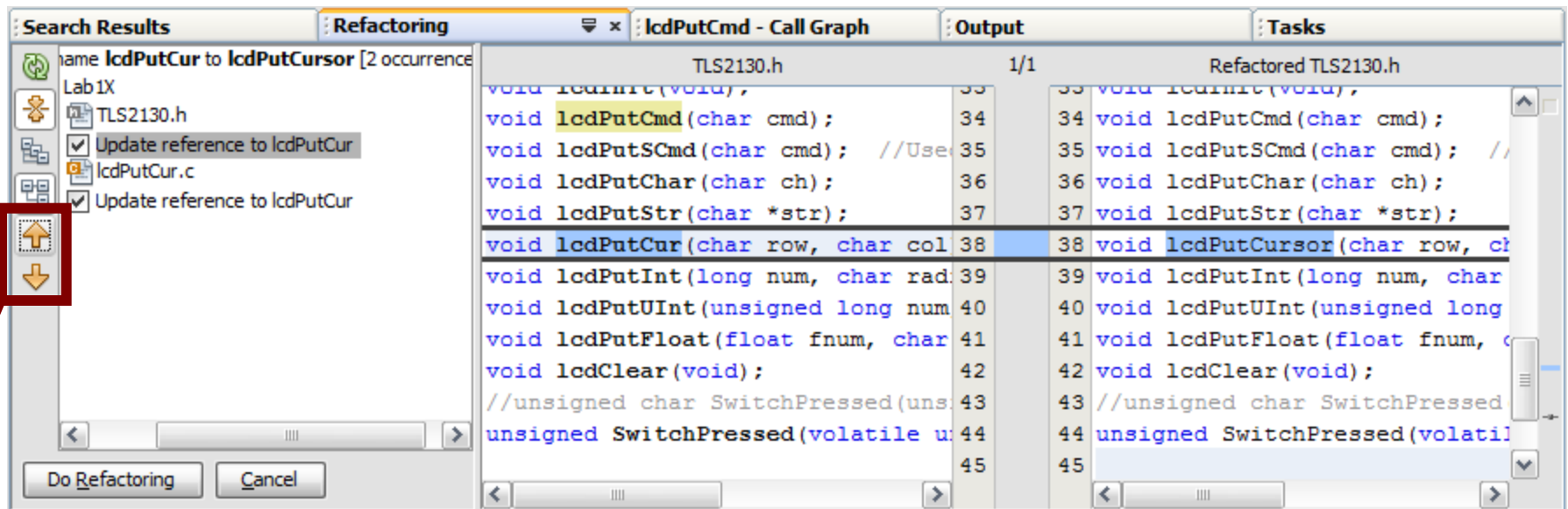
Preview



如何使用导航器

如何重构符号名称

- 通过 **Preview**，可以在实施重构之前预览将实现的更改



遍历右侧预览窗口中的每个实例



MICROCHIP 2010

MASTERS Conference

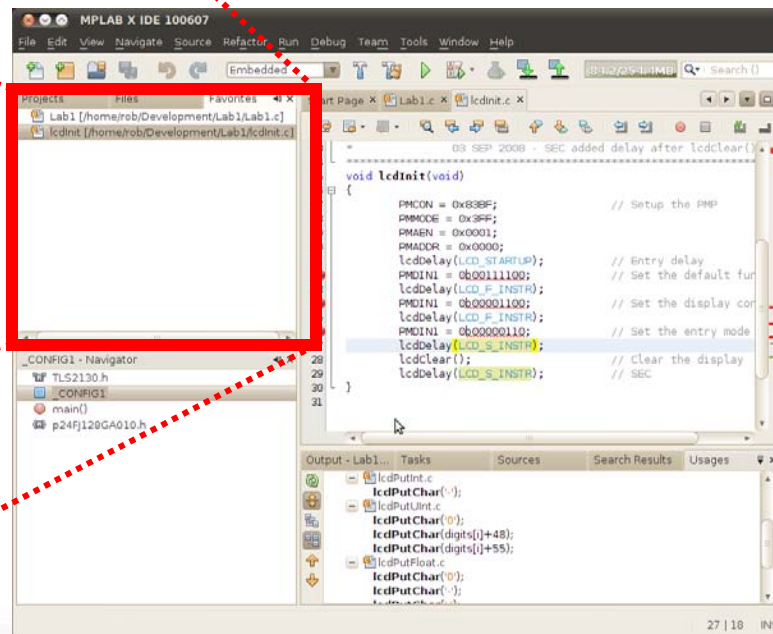
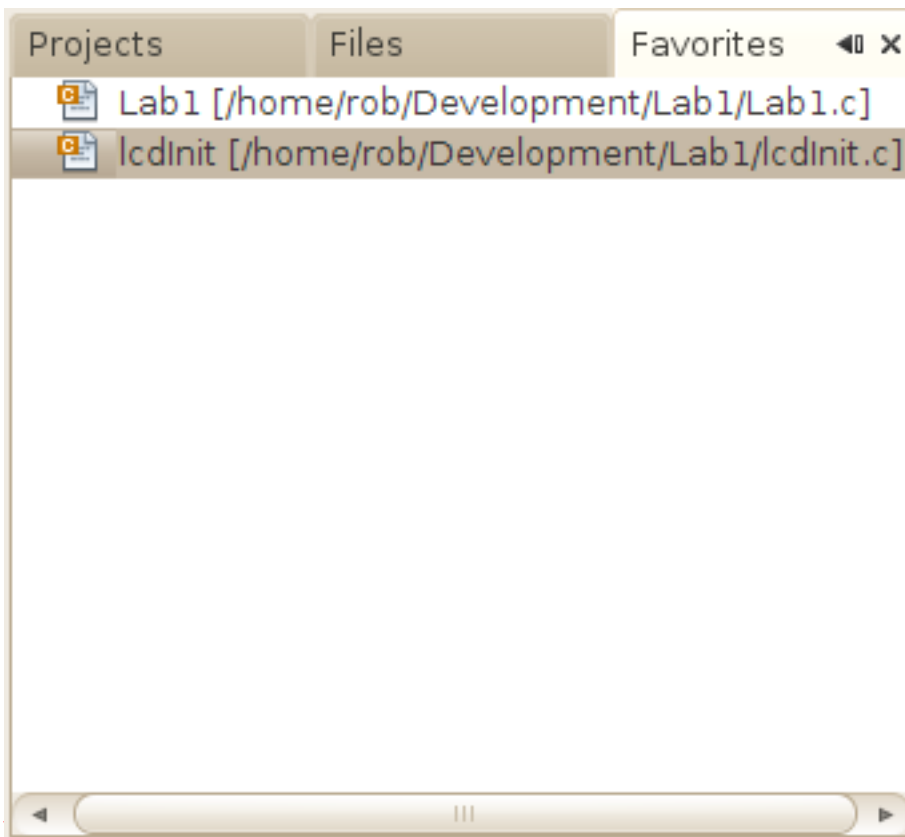
如何进行项目导航

收藏夹

收藏夹

如何查看收藏夹列表

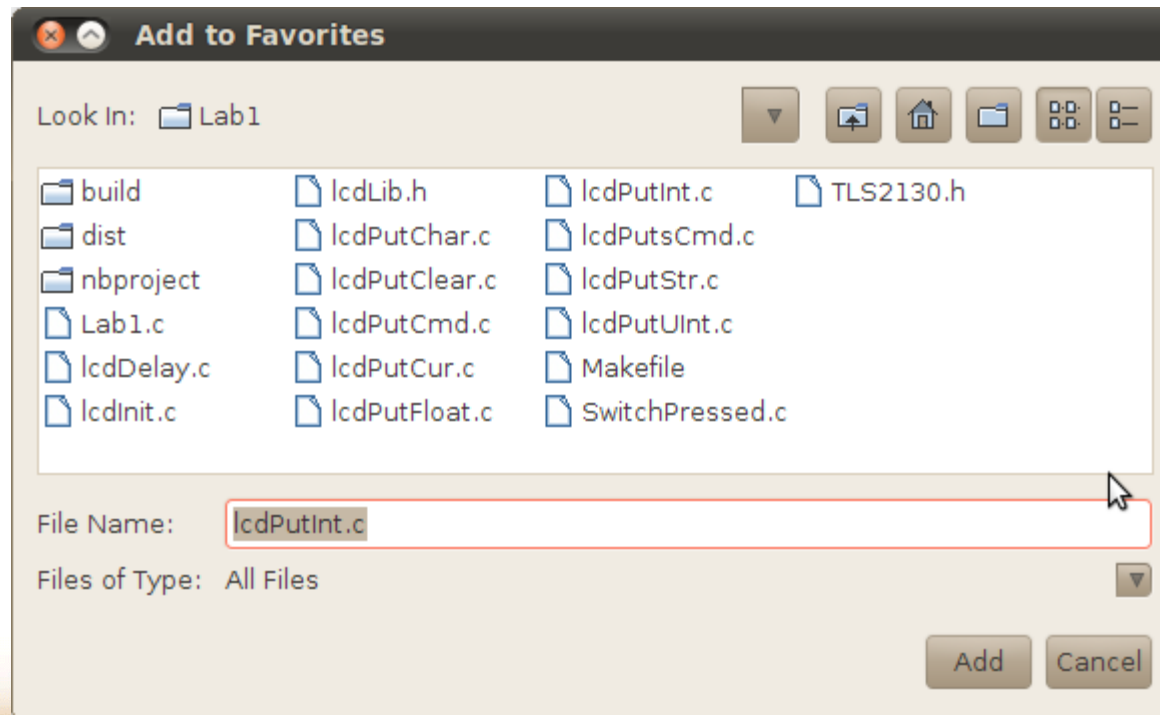
- 从菜单中选择 **Window ▶ Favorites** (收藏夹)



收藏夹

如何将文件添加到收藏夹列表中

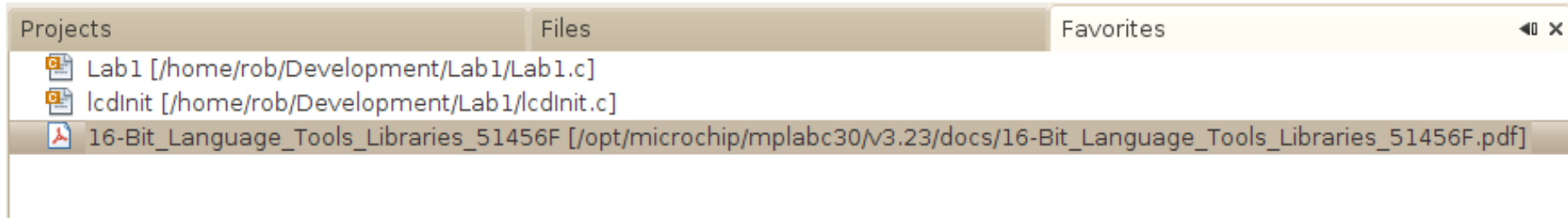
- 右键单击已打开的源文件，并从主菜单中选择 **Select in (选择) ▶ Favorites 或 Navigate ▶ Select in Favorites (在收藏夹中选择)**，可以打开 **Add to Favorites (添加至收藏夹)** 对话框（如果文件不在其中）



收藏夹

如何将文件添加到收藏夹列表中

- 在**Favorites**窗口中单击右键，并从弹出菜单中选择 **Add to Favorites...**（添加至收藏夹...）
- 不限于代码——可添加任意文件（例如，**PDF**参考手册）

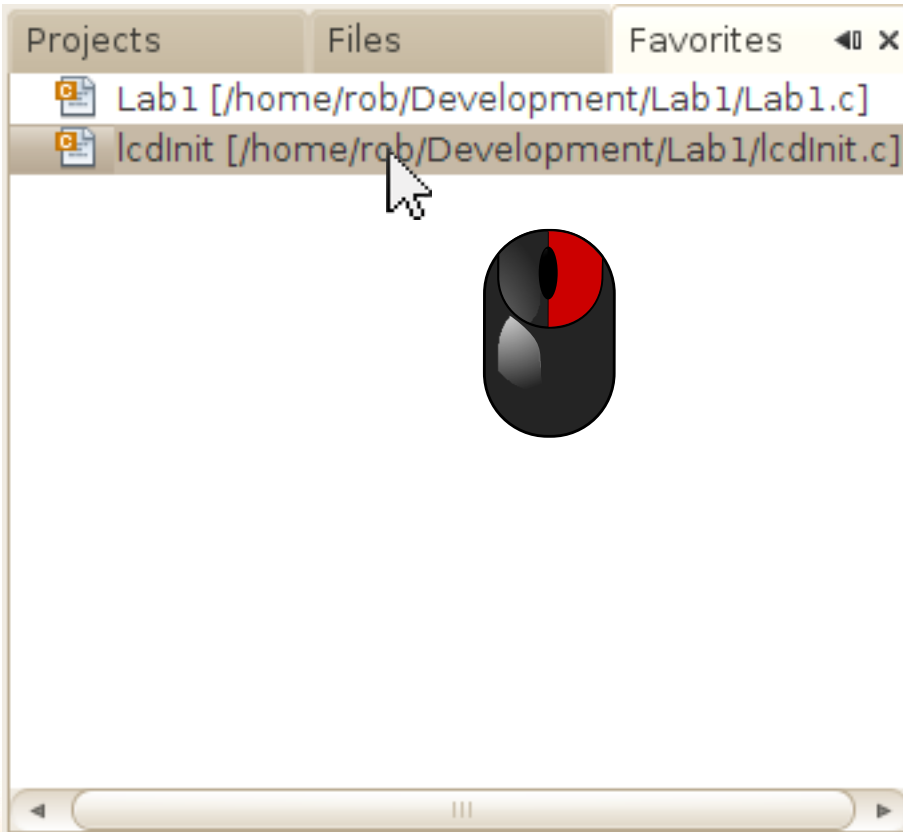


双击PDF时，将会在PDF阅读器（例如，Adobe Reader和Evince等）中打开文件

收藏夹

如何从收藏夹列表中移除文件

- 右键单击收藏夹窗口中的某一项，并选择 **Remove from Favorites**（从收藏夹中移除）





MICROCHIP 2010

MASTERS Conference

如何进行项目导航

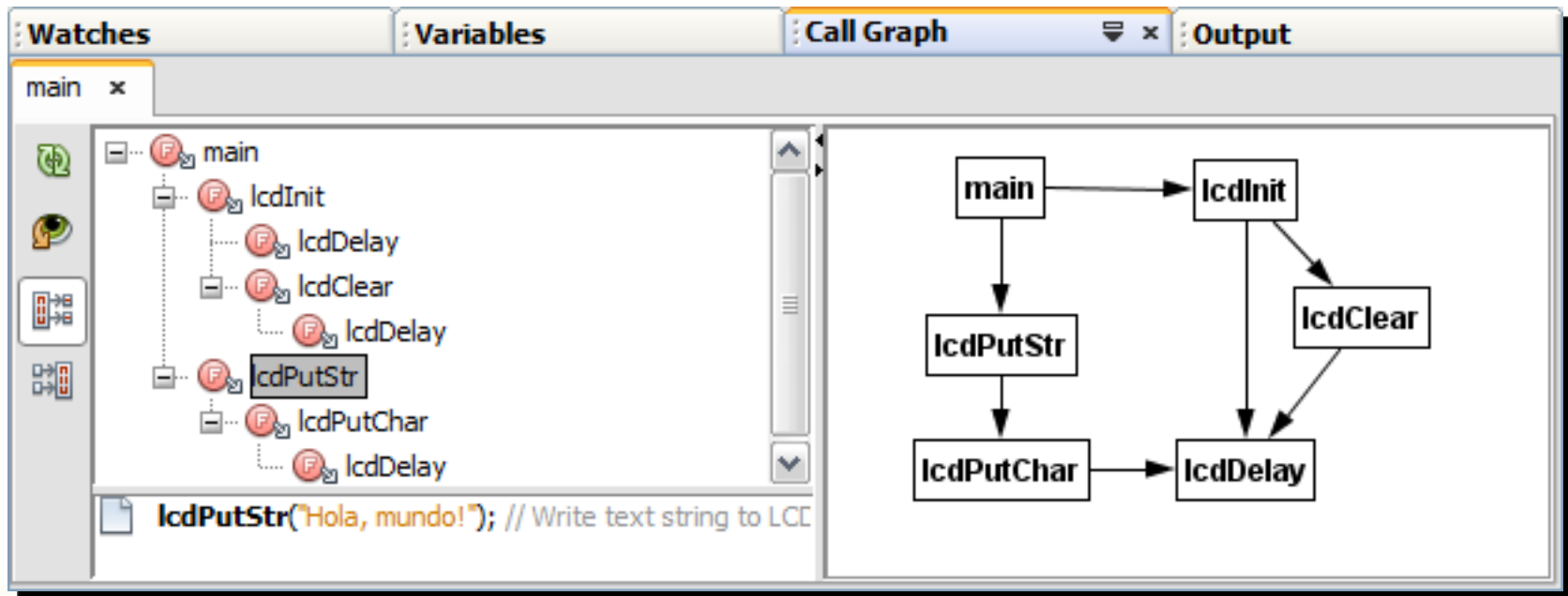
如何使用调用图

如何使用调用图

静态调用图

定义

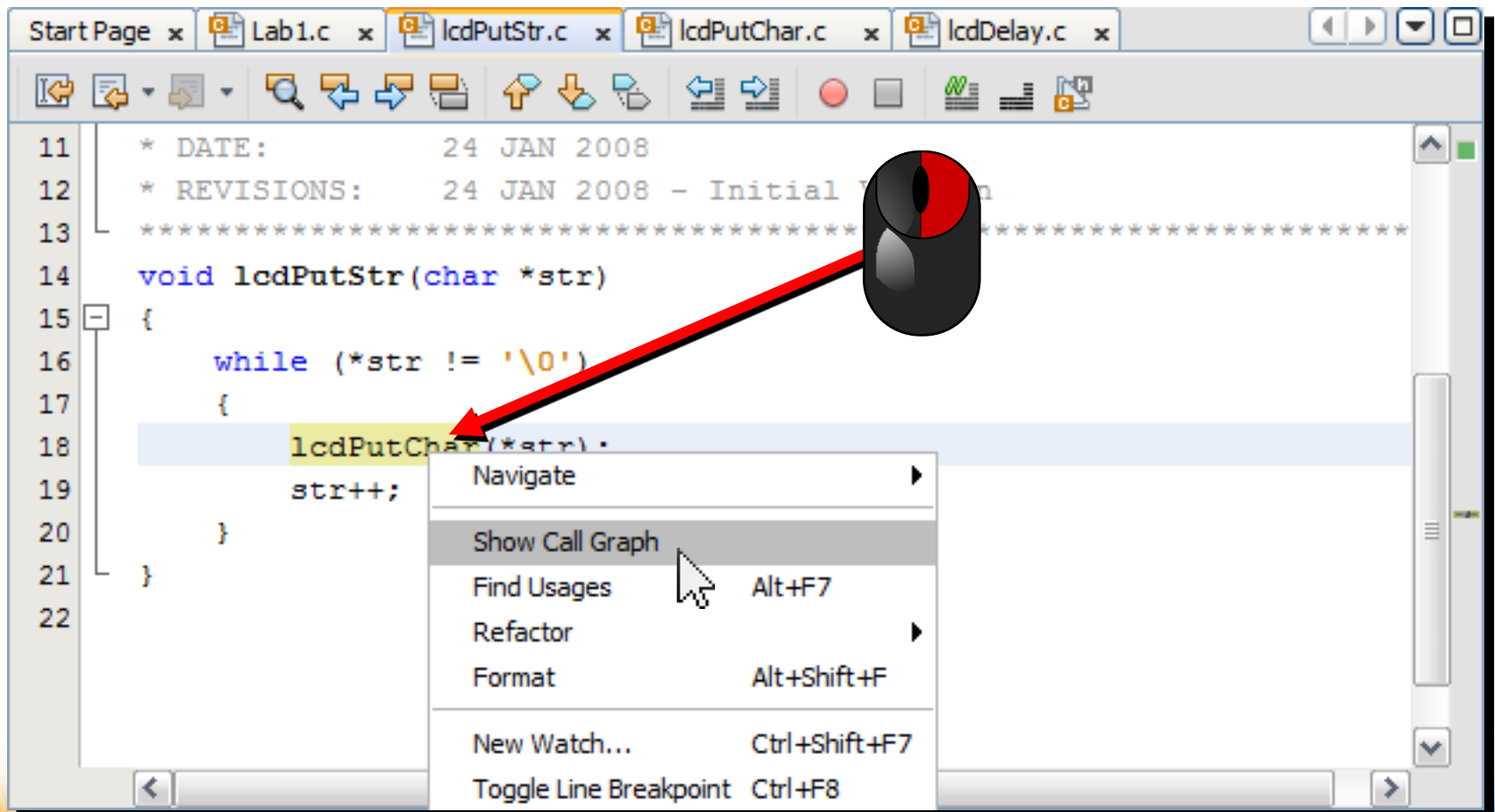
静态调用图——显示程序中各个子程序之间的调用关系的流程图。它代表函数可用的每个可能路径。



如何使用调用图

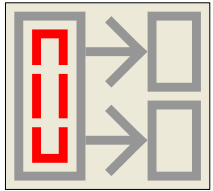
生成函数的调用图

- 右键单击函数，并从弹出菜单中选择 **Show Call Graph**（显示调用图）

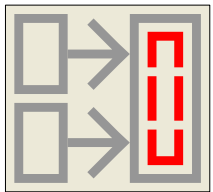
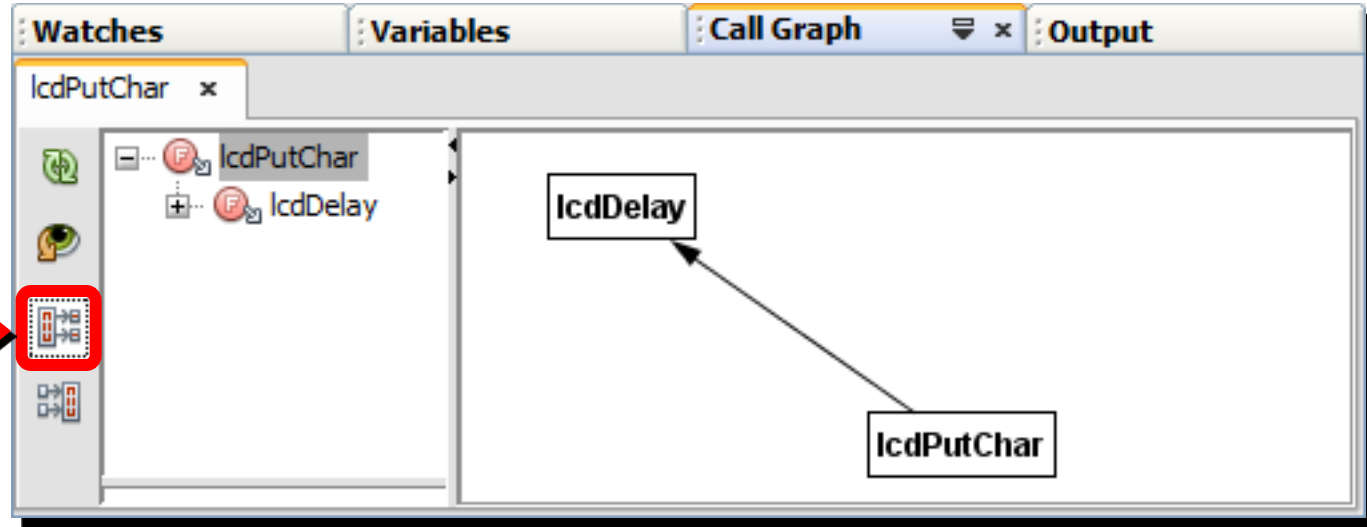
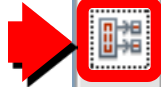


如何使用调用图

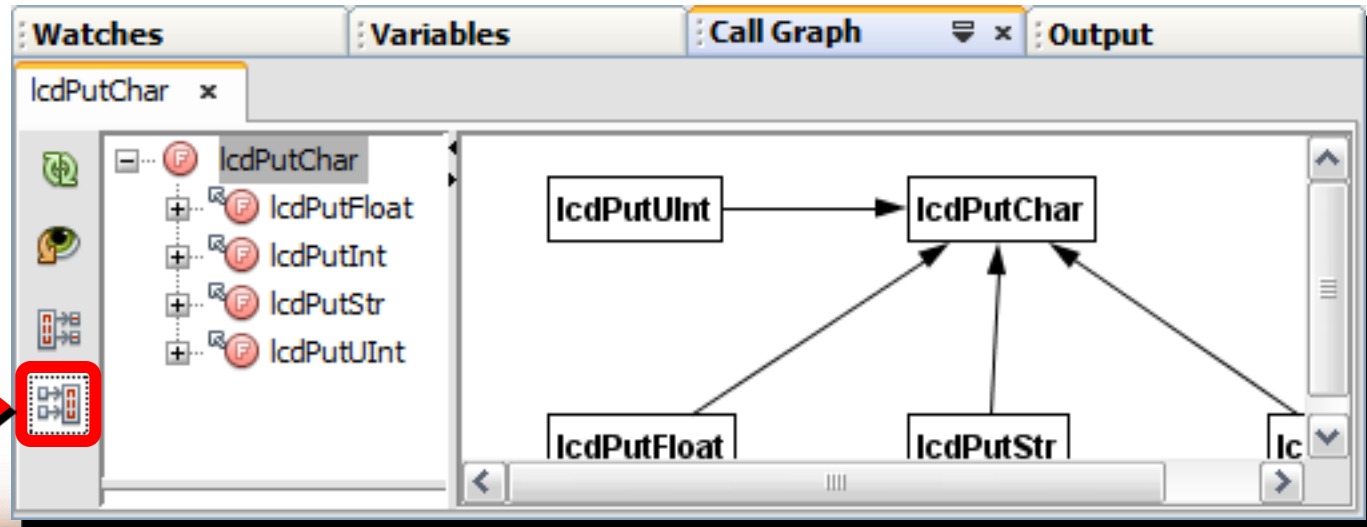
两种工作模式



该函数调用了
哪些函数？



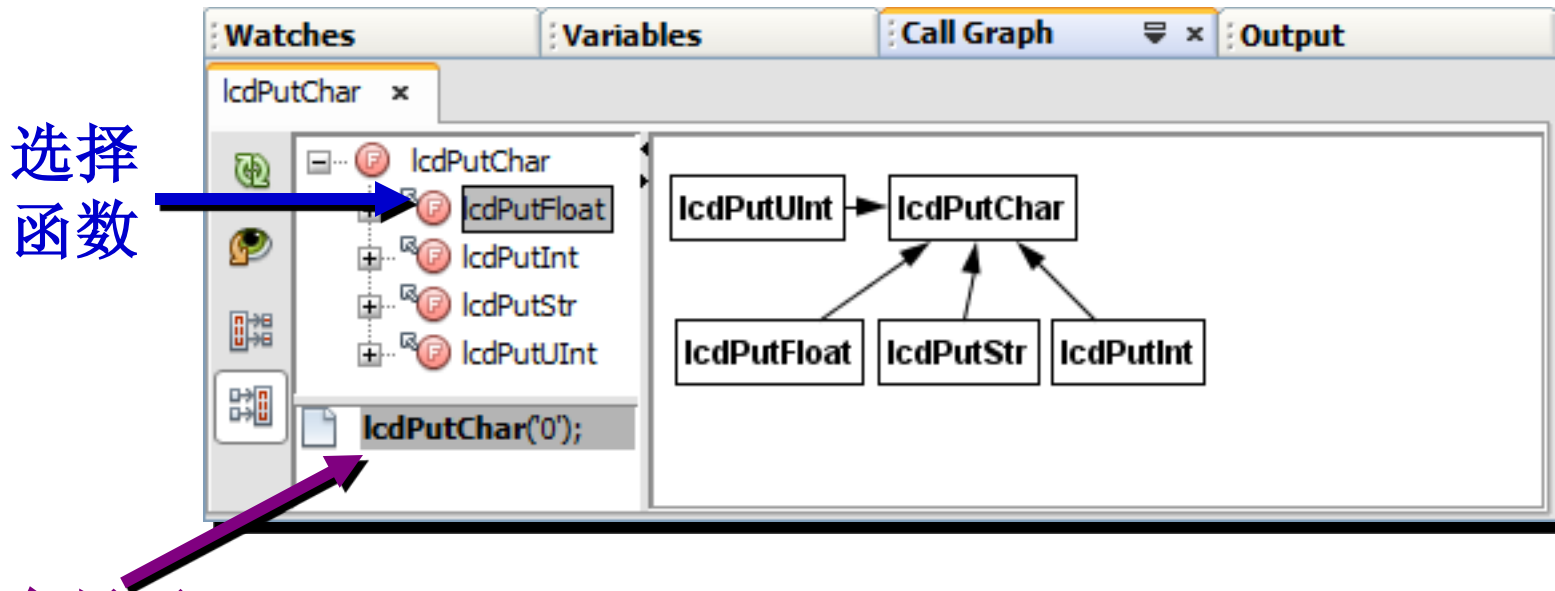
哪些函数调用
了该函数？



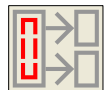
如何使用调用图

查看函数调用与参数

- 在左窗格中，选择子节点函数



下方窗格会显示:



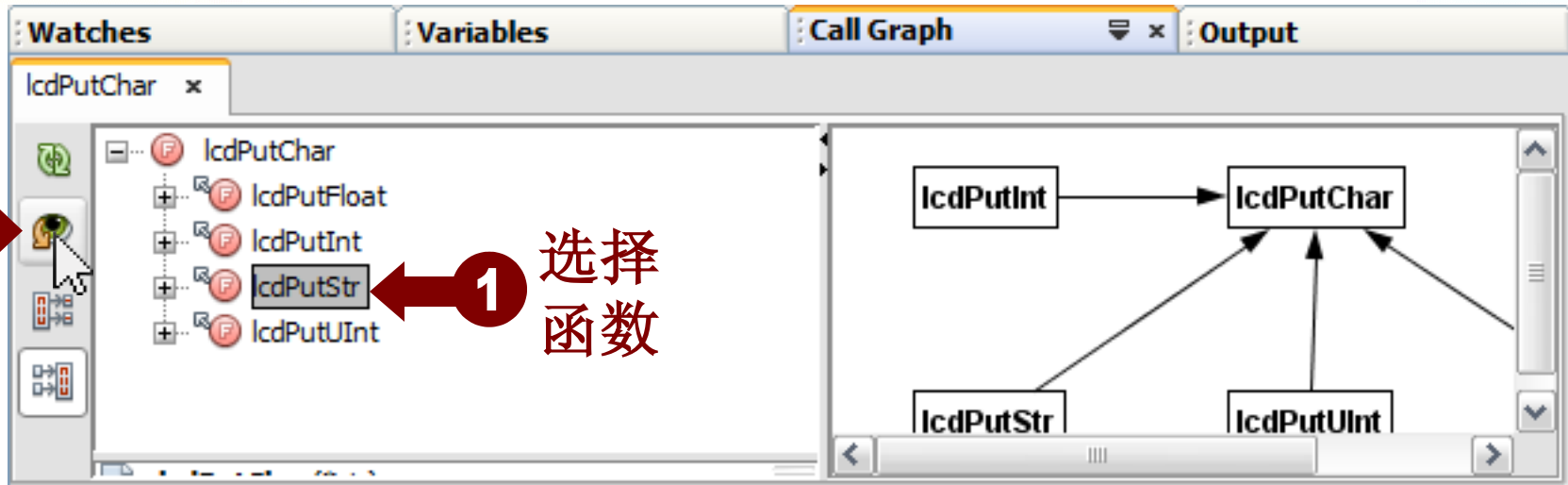
顶级节点函数如何调用子节点函数



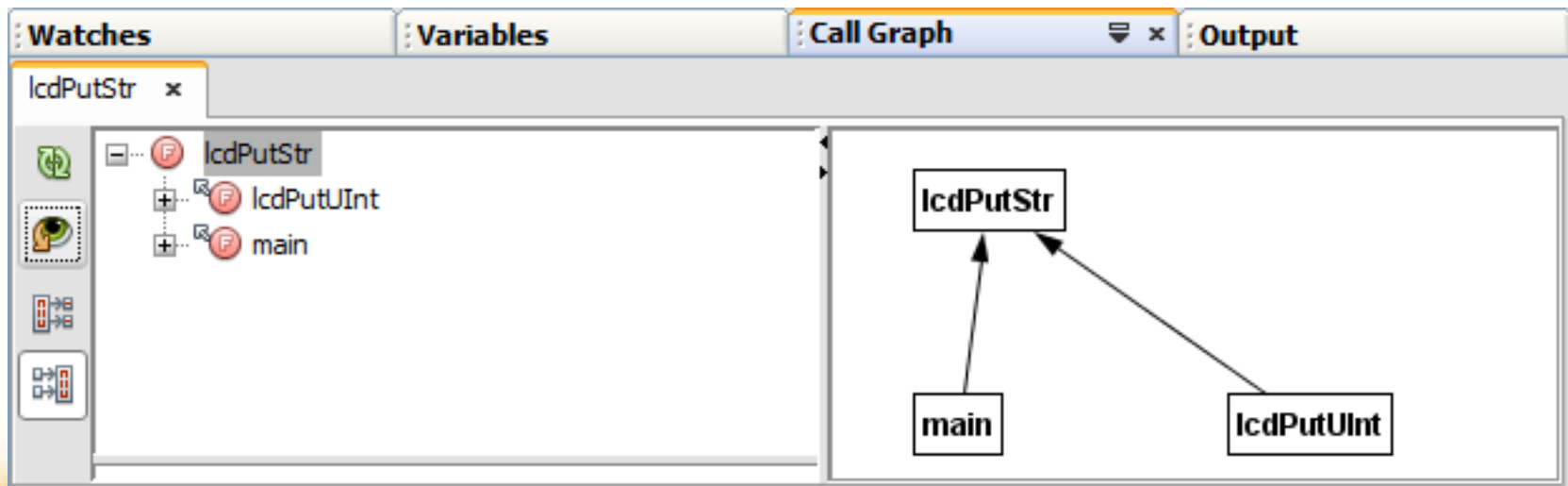
顶级节点函数如何被子节点函数调用

如何使用调用图

将焦点更改到子节点函数

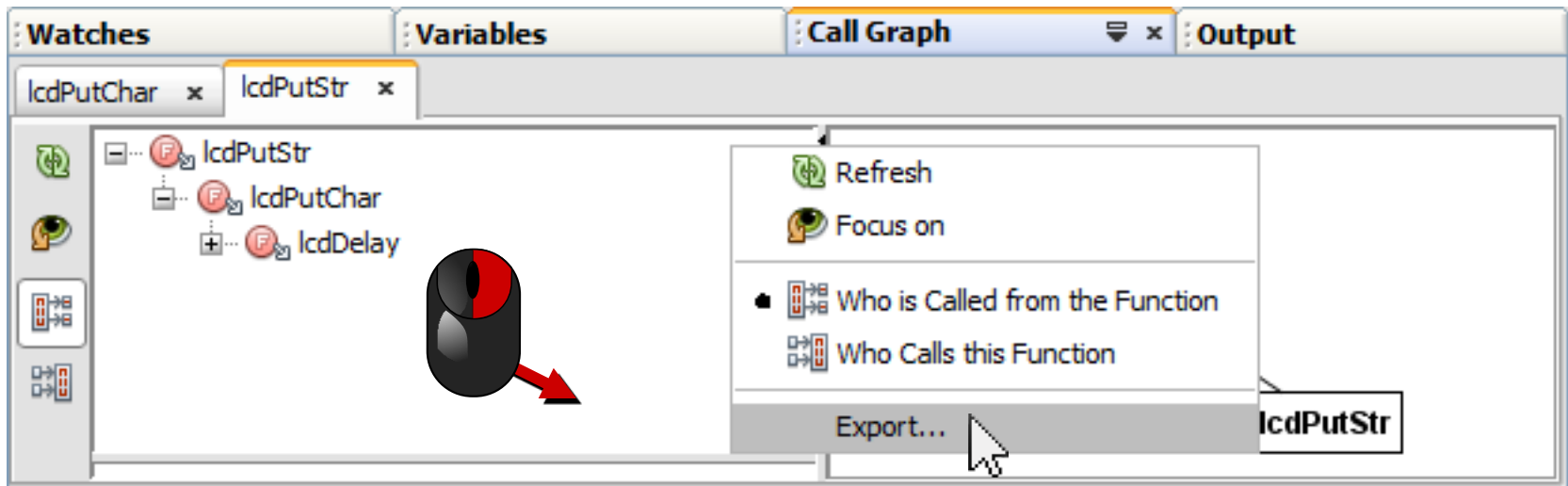


2 Focus On (定焦) 或者, 右键单击函数并选择 Focus On (定焦)



如何使用调用图 将图形导出为PNG文件

在调用图窗口中右键单击任意位置，并从弹出菜单中选择 **Export...**（导出...）





MICROCHIP 2010

MASTERS Conference



如何使用Watch窗口进行调试

动手练习

实验3

如何使用Watch窗口进行调试



目标

- 在调试器中设置断点。
- 观察变量。
- 学习如何观察存储器窗口。



要求： 带有PIC24FJ128GA010的Explorer 16演示板



MICROCHIP 2010

MASTERS Conference

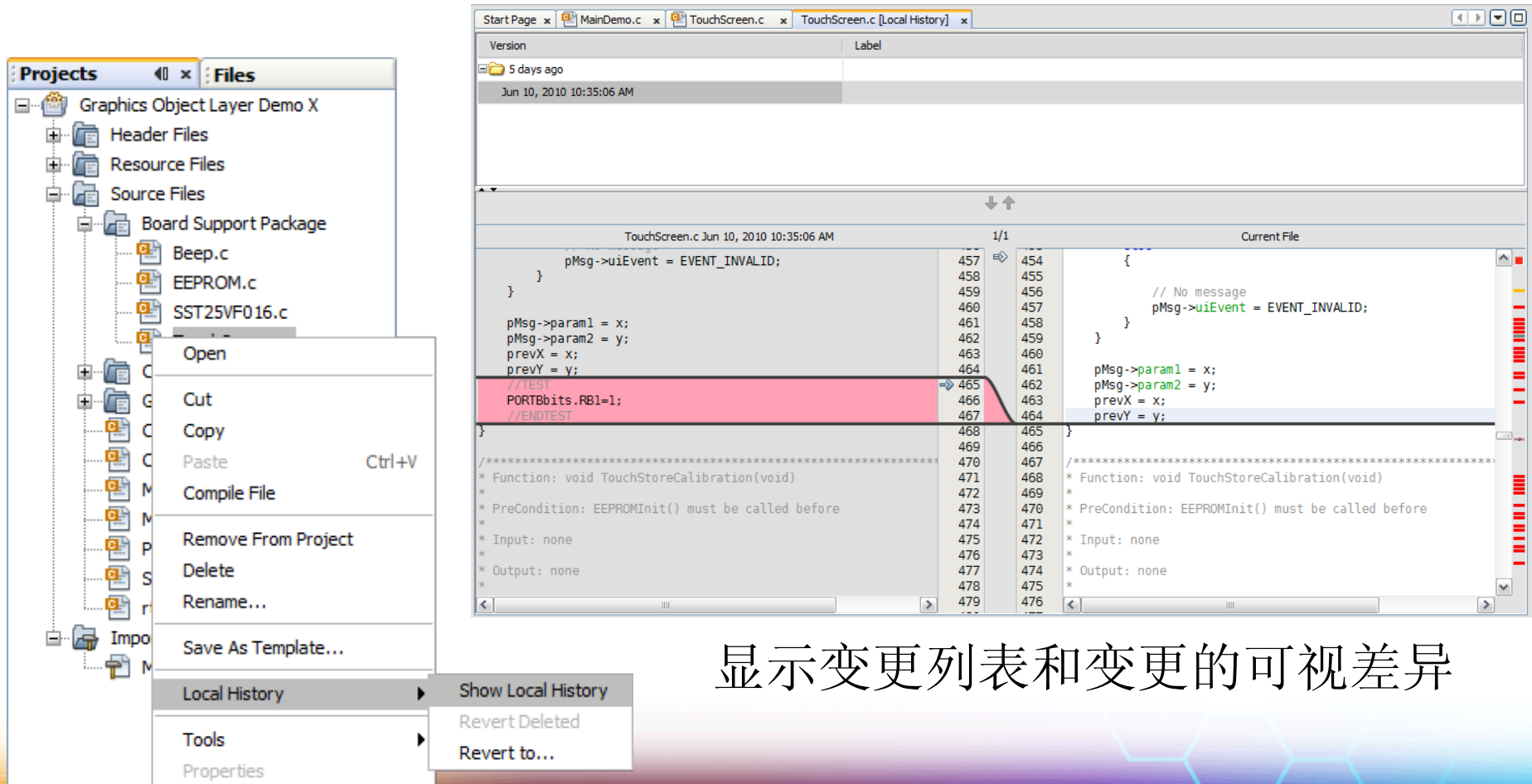
如何使用本地历史记录

本地历史记录

如何显示本地历史记录

- 右键单击文件，并从弹出菜单中选择：

Local History (本地历史记录) ▶ Show Local History (显示本地历史记录)



The screenshot illustrates the process of viewing local history in an IDE. On the left, a file explorer shows a project structure with a context menu open over a file. The 'Local History' option is selected, and a sub-menu shows 'Show Local History'. On the right, the 'Local History' window displays a list of versions for 'TouchScreen.c'. The selected version is 'Jun 10, 2010 10:35:06 AM'. Below this, a side-by-side comparison of code versions is shown, highlighting differences between the current file and a previous version.

Version	Label
5 days ago	
Jun 10, 2010 10:35:06 AM	

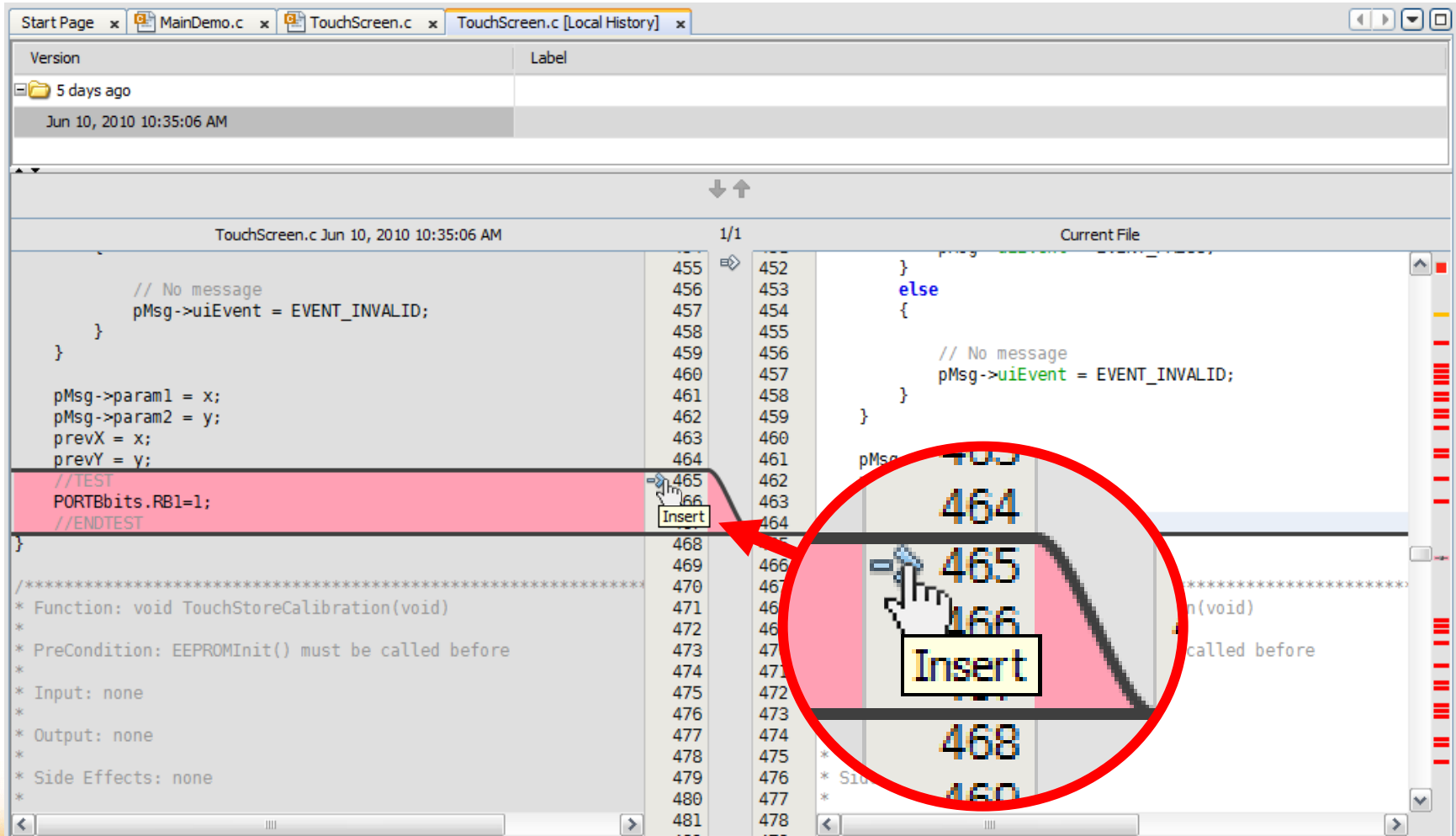
TouchScreen.c Jun 10, 2010 10:35:06 AM	1/1	Current File
<pre>pMsg->uiEvent = EVENT_INVALID; } } pMsg->param1 = x; pMsg->param2 = y; prevX = x; prevY = y; //TEST PORTBbits.RB1=1; //ENDTEST } /***** * Function: void TouchStoreCalibration(void) * PreCondition: EEPROMInit() must be called before * Input: none * Output: none *****/</pre>	<pre>457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479</pre>	<pre>{ // No message pMsg->uiEvent = EVENT_INVALID; } pMsg->param1 = x; pMsg->param2 = y; prevX = x; prevY = y; } /***** * Function: void TouchStoreCalibration(void) * PreCondition: EEPROMInit() must be called before * Input: none * Output: none *****/</pre>

显示变更列表和变更的可视差异

本地历史记录

如何插入来自先前版本的代码

■ 单击右侧页边中的蓝色箭头...



The screenshot shows an IDE window with the following tabs: Start Page, MainDemo.c, TouchScreen.c, and TouchScreen.c [Local History]. The main editor displays the current file's code, and the right-hand pane shows a previous version of the code. A red circle highlights the 'Insert' button in the right-hand pane, which is used to insert code from a previous version.

```
Version | Label
-----|-----
5 days ago
Jun 10, 2010 10:35:06 AM

TouchScreen.c Jun 10, 2010 10:35:06 AM | 1/1 | Current File

// No message
pMsg->uiEvent = EVENT_INVALID;
}
}

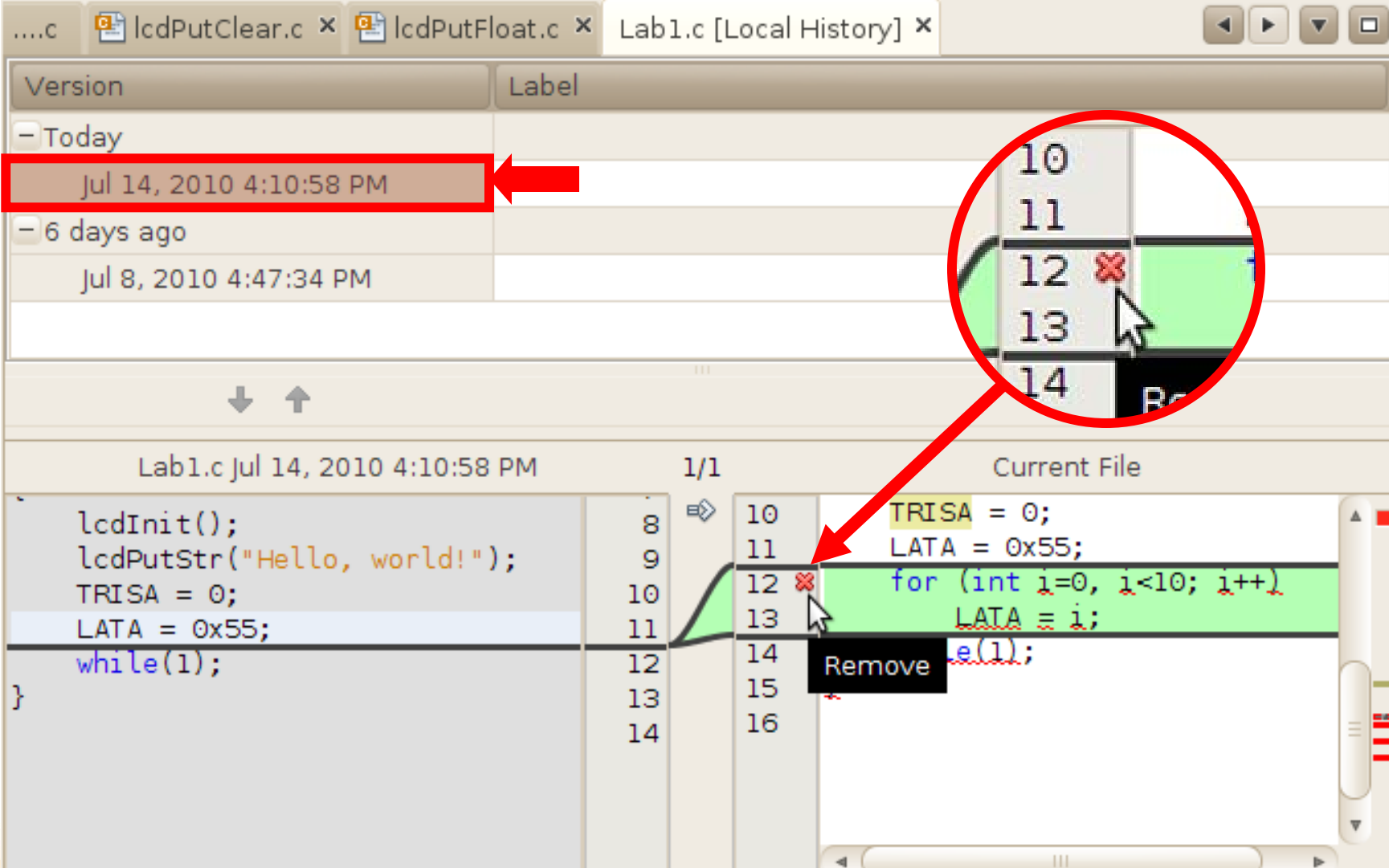
pMsg->param1 = x;
pMsg->param2 = y;
prevX = x;
prevY = y;
//TEST
PORTBbits.RB1=1;
//ENDTEST
}

/*****
* Function: void TouchStoreCalibration(void)
*
* PreCondition: EEPROMInit() must be called before
*
* Input: none
*
* Output: none
*
* Side Effects: none
*****/

455 | 452 | }
456 | 453 | else
457 | 454 | {
458 | 455 |
459 | 456 | // No message
460 | 457 | pMsg->uiEvent = EVENT_INVALID;
461 | 458 | }
462 | 459 | }
463 | 460 |
464 | 461 | pMsg->uiEvent = EVENT_INVALID;
465 | 462 | }
466 | 463 |
467 | 464 |
468 | 465 |
469 | 466 |
470 | 467 |
471 | 468 |
472 | 469 |
473 | 470 |
474 | 471 |
475 | 472 |
476 | 473 |
477 | 474 |
478 | 475 |
479 | 476 |
480 | 477 |
481 | 478 |
```

本地历史记录

如何撤销更改



The screenshot shows the local history window for 'Lab1.c [Local History]'. The history table has two columns: 'Version' and 'Label'. The 'Today' section contains one entry: 'jul 14, 2010 4:10:58 PM', which is highlighted with a red box and a red arrow. Below it is the '6 days ago' section with one entry: 'jul 8, 2010 4:47:34 PM'. The main editor shows the code for 'Lab1.c' as of 'jul 14, 2010 4:10:58 PM'. The code is:

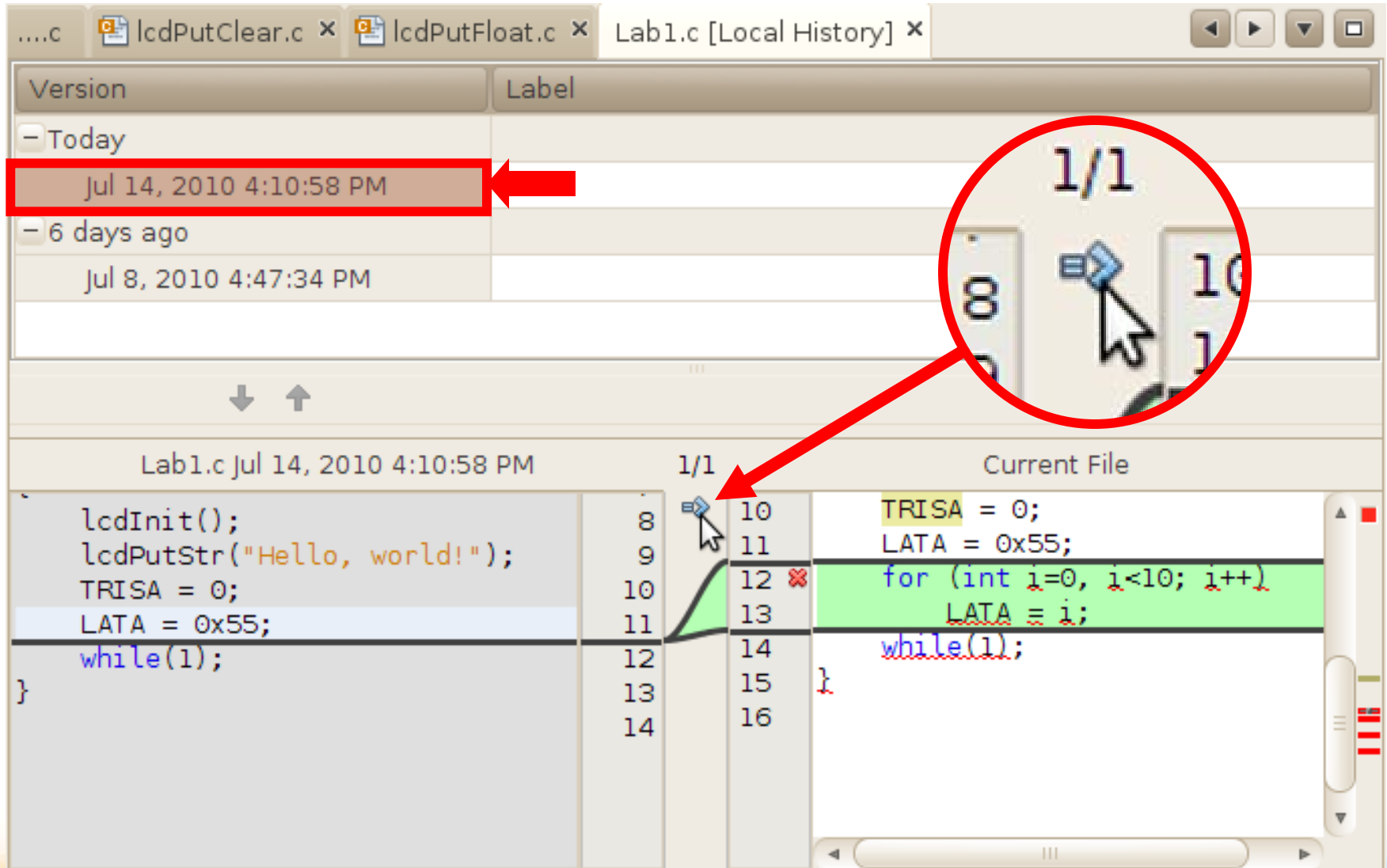
```
lcdInit();  
lcdPutStr("Hello, world!");  
TRISA = 0;  
LATA = 0x55;  
while(1);  
}
```

The code is displayed in two columns. The right column shows lines 10 through 16. Line 12 is highlighted with a green background and has a red 'X' icon next to it. A red circle highlights this 'X' icon, and a red arrow points from it to the 'Remove' tooltip that appears over the line. The code for lines 10-16 is:

```
10 TRISA = 0;  
11 LATA = 0x55;  
12 for (int i=0, i<10; i++)  
13     LATA = i;  
14  
15  
16
```

本地历史记录

如何恢复为文件的先前版本



...c lcdPutClear.c x lcdPutFloat.c x Lab1.c [Local History] x

Version	Label
- Today	
jul 14, 2010 4:10:58 PM	
- 6 days ago	
jul 8, 2010 4:47:34 PM	

Lab1.c jul 14, 2010 4:10:58 PM 1/1 Current File

```
lcdInit();  
lcdPutStr("Hello, world!");  
TRISA = 0;  
LATA = 0x55;  
while(1);  
}
```

```
TRISA = 0;  
LATA = 0x55;  
for (int i=0, i<10; i++)  
    LATA = i;  
while(1);  
}
```




MICROCHIP 2010

MASTERS Conference



实验4

如何切换MPLAB[®]驱动程序

讲师指导的动手练习

实验4

如何切换MPLAB®驱动程序



目标

- 在MPLAB 8和MPLAB X之间切换工具驱动程序



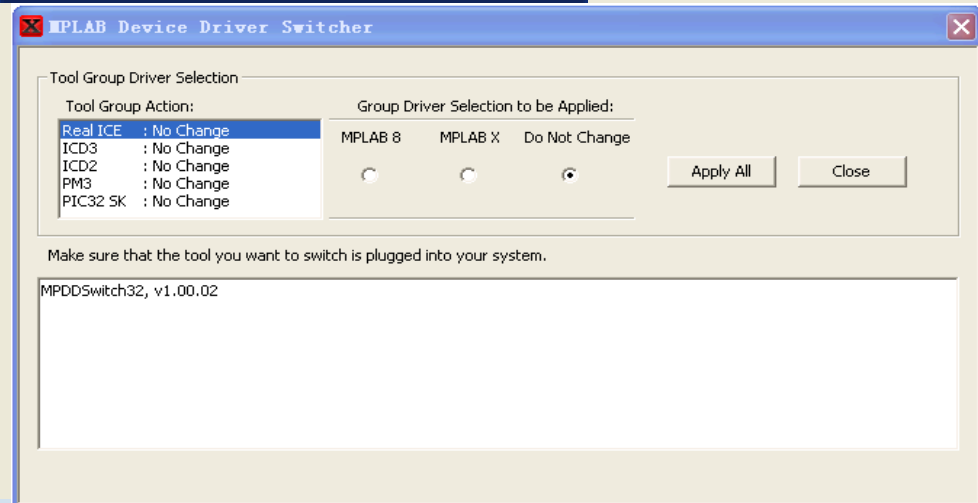
要求： ICD3



1 启动MPLAB驱动程序切换器



在菜单中单击**MPLAB driver switcher**（**MPLAB 驱动程序切换器**）来打开软件。

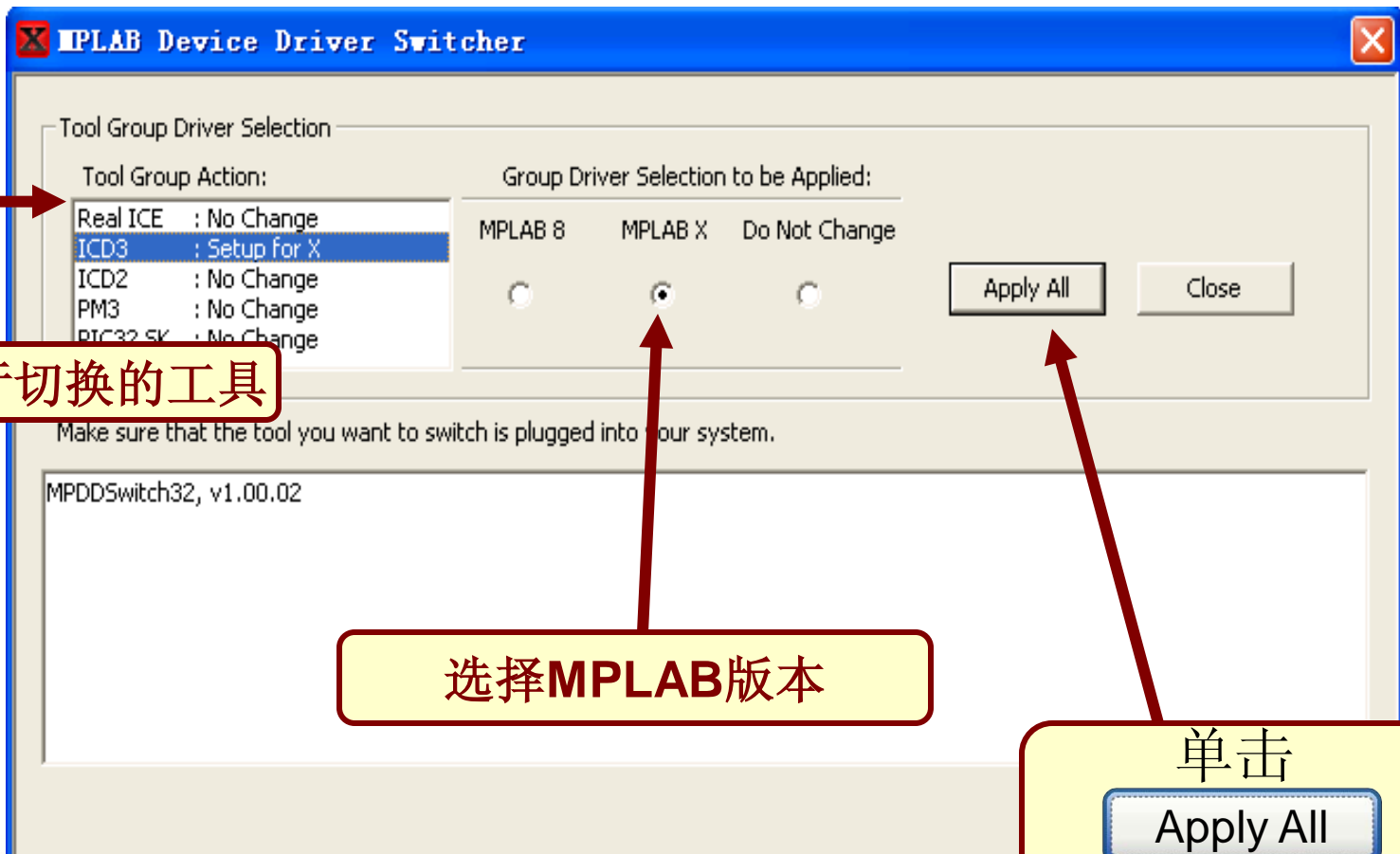


 MPLAB X为MPLAB工具提供了WinUSB驱动程序，所以需要将器件驱动程序从MPLAB 8切换为MPLAB X。



如何切换MPLAB®驱动程序

2 选择要进行切换的工具





MICROCHIP 2010

MASTERS Conference

在Linux操作系统中运行
MPLAB[®] X IDE

商标

Microchip的名称和徽标组合、Microchip徽标、dsPIC、KeeLoq、KeeLoq徽标、MPLAB、PIC、PICmicro、PICSTART、PIC³²徽标、rfPIC和UNI/O均为Microchip Technology Incorporated在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL和The Embedded Control Solutions Company均为Microchip Technology Incorporated在美国的注册商标。

Analog-for-the-Digital Age、Application Maestro、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified徽标、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICKit、PICTail、REAL ICE、rfLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock和ZENA均为Microchip Technology Incorporated在美国和其他国家或地区的商标。

SQTP是Microchip Technology Incorporated在美国的服务标记。

在此提及的所有其他商标均为各持有公司所有。

© 2010, Microchip Technology Incorporated, 版权所有。