

十速

# TM52系列-软件开发环境

使用手册

**Rev V0.90**

**tenx** reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **tenx** does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses **tenx** products for any such unintended or unauthorized application, Buyer shall indemnify and hold **tenx** and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that **tenx** was negligent regarding the design or manufacture of the part.

---

## 修改記錄

版本	日期	描述
V0.90	Dec, 2013	新頒

## 目錄

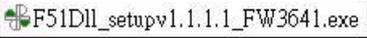
修改記錄.....	2
1. 安装 Keil C 软件 .....	4
2. 安装 tenx F51&L51 Keil 插件 .....	4
3. 连接 USB .....	7
4. 开启 Keil C 设定编写程序 .....	8
5. 编写程序代码 .....	17
6. 模拟调试 (Debug) .....	21
7. 调试命令 .....	35
8. C 语言文件中嵌入汇编 .....	40
常见问题 Q & A .....	43

## TM52 系列-软件开发环境使用手册

tenx（十速科技）推出的 tenx F8051 系列单芯片是兼容于 8051，因此可以使用成熟的 Keil uVision 系列软件作为开发环境。本文将对 tenx F51&L51 系列在 Keil 软件中的配置和使用做简单介绍。

### 1. 安装 Keil C 软件

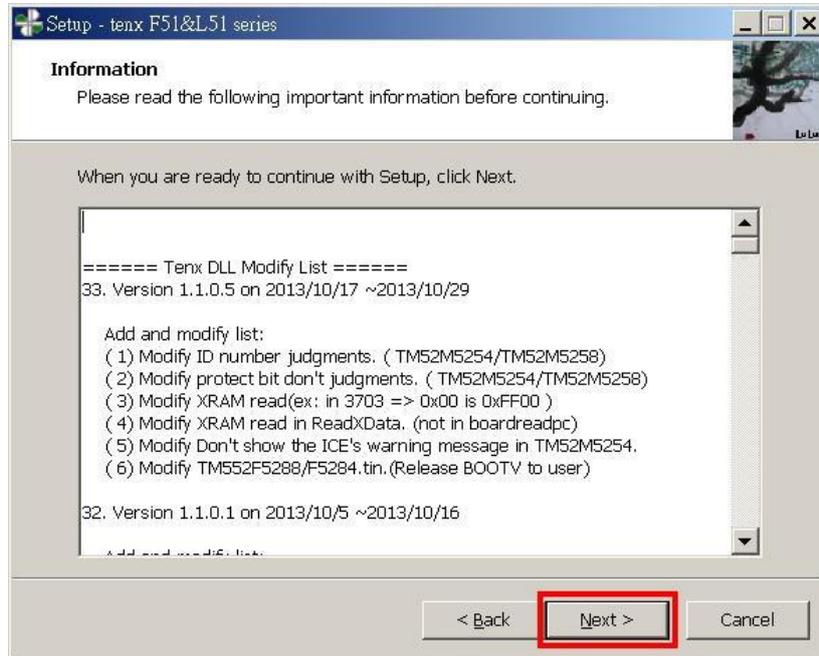
### 2. 安装 tenx F51&L51 Keil 插件

安装 tenx F51&L51 Keil 插件 ，按下面步骤完成安装，安装默认路径为 C:\Keil:

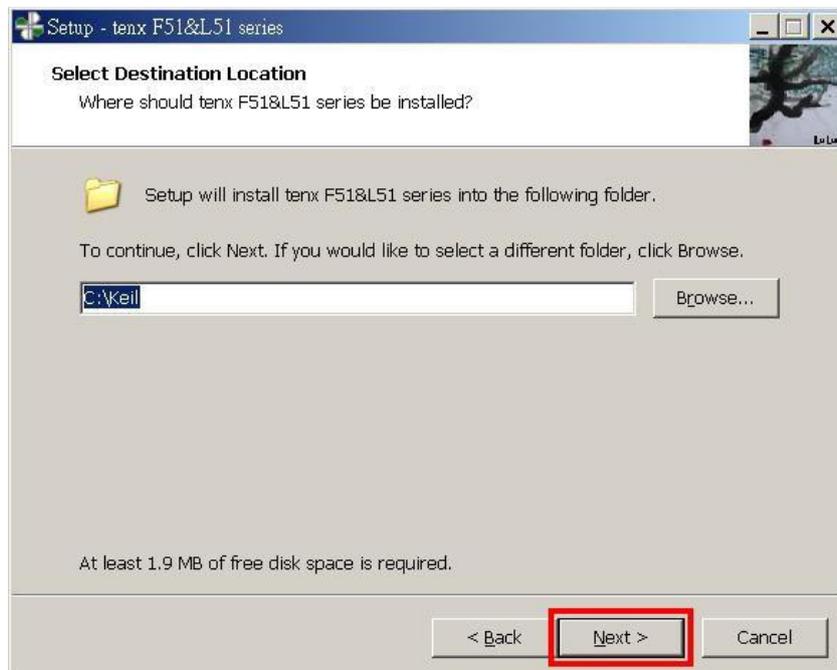
#### 2.1 选择 Keil C 版本后点选 Next>



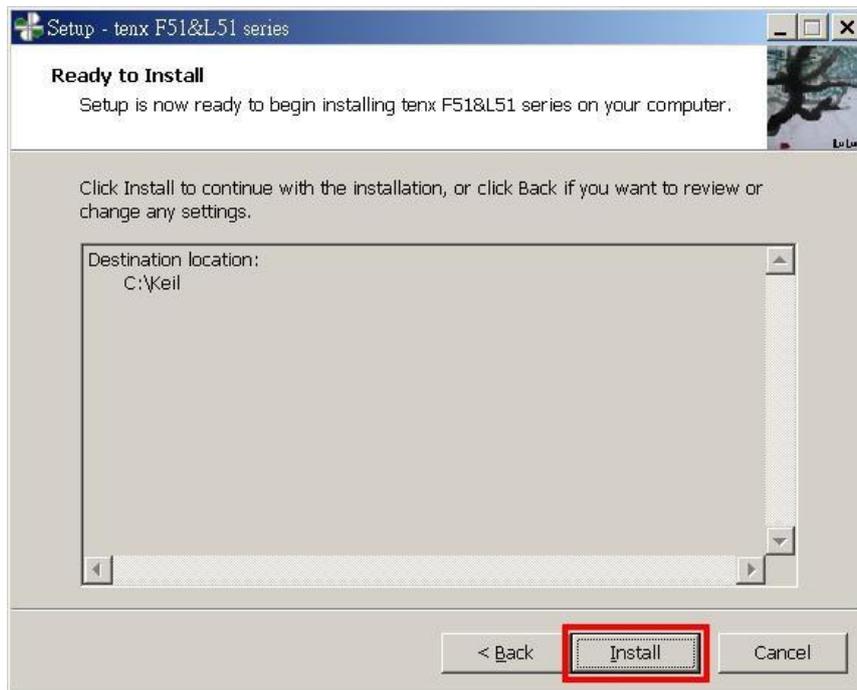
## 2.2 再点选 Next>



## 2.3 默认路径为 C:\Keil, 点选 Next>



## 2.4 点选 Install

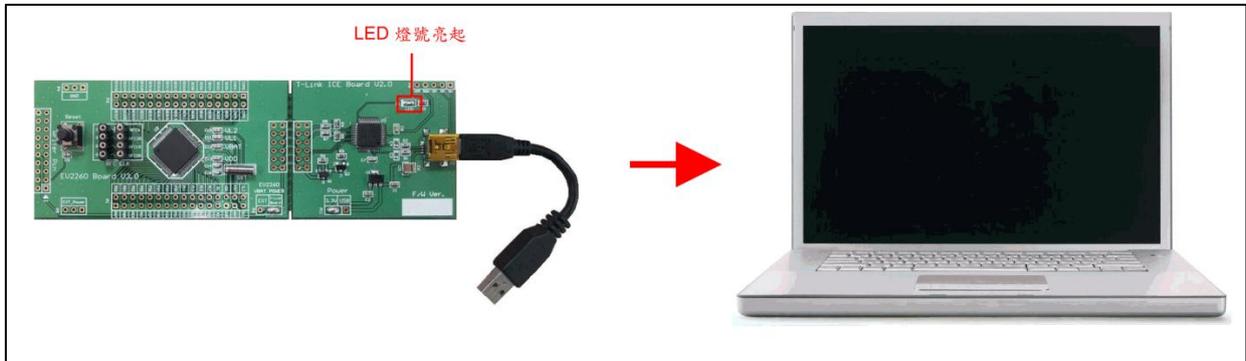


## 2.5 点选 Finish 完成安装

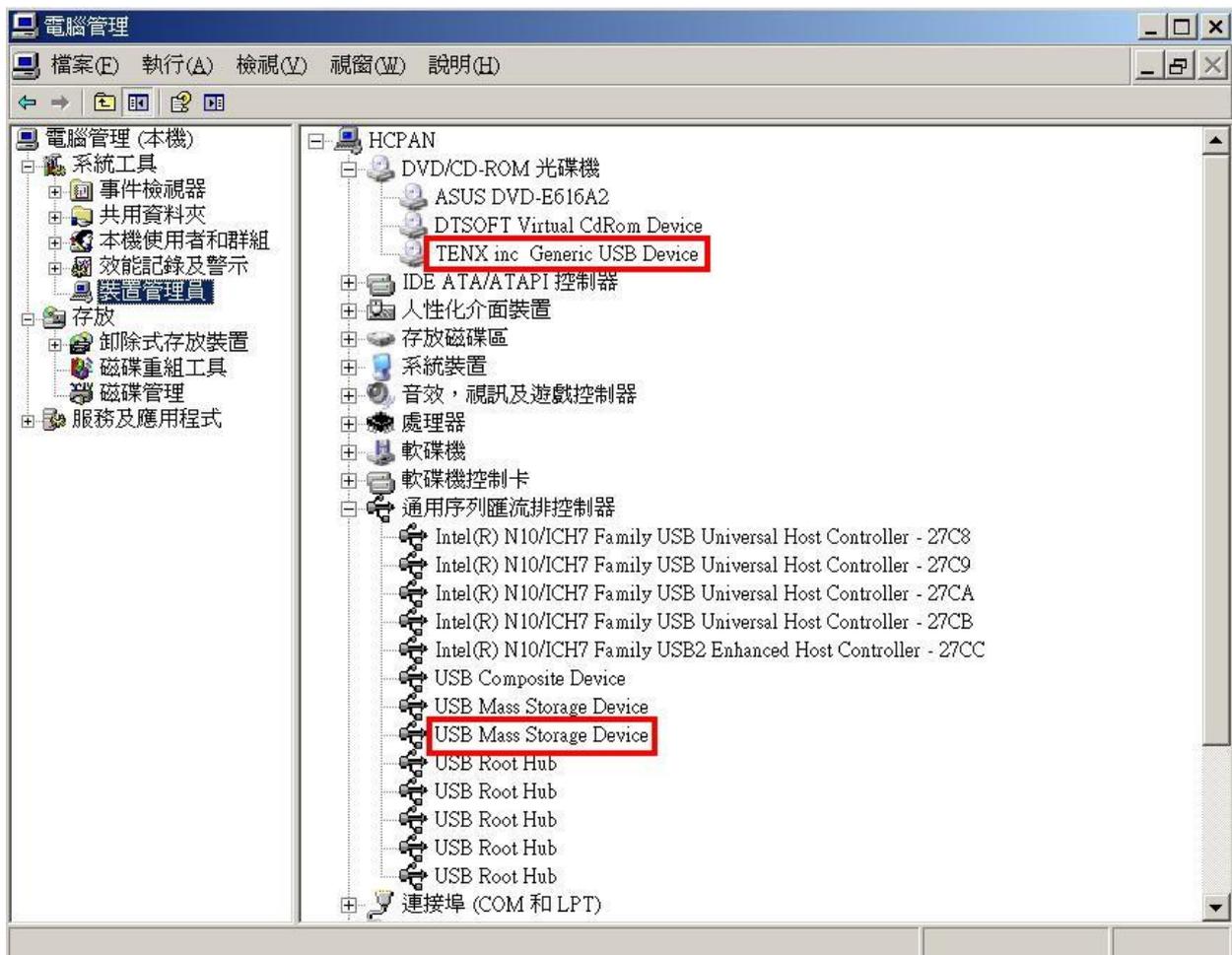


### 3. 连接 USB

#### 3.1 连接 EV Board 的 USB (MINI B) 接头至 PC(此时 LED 灯号亮起)

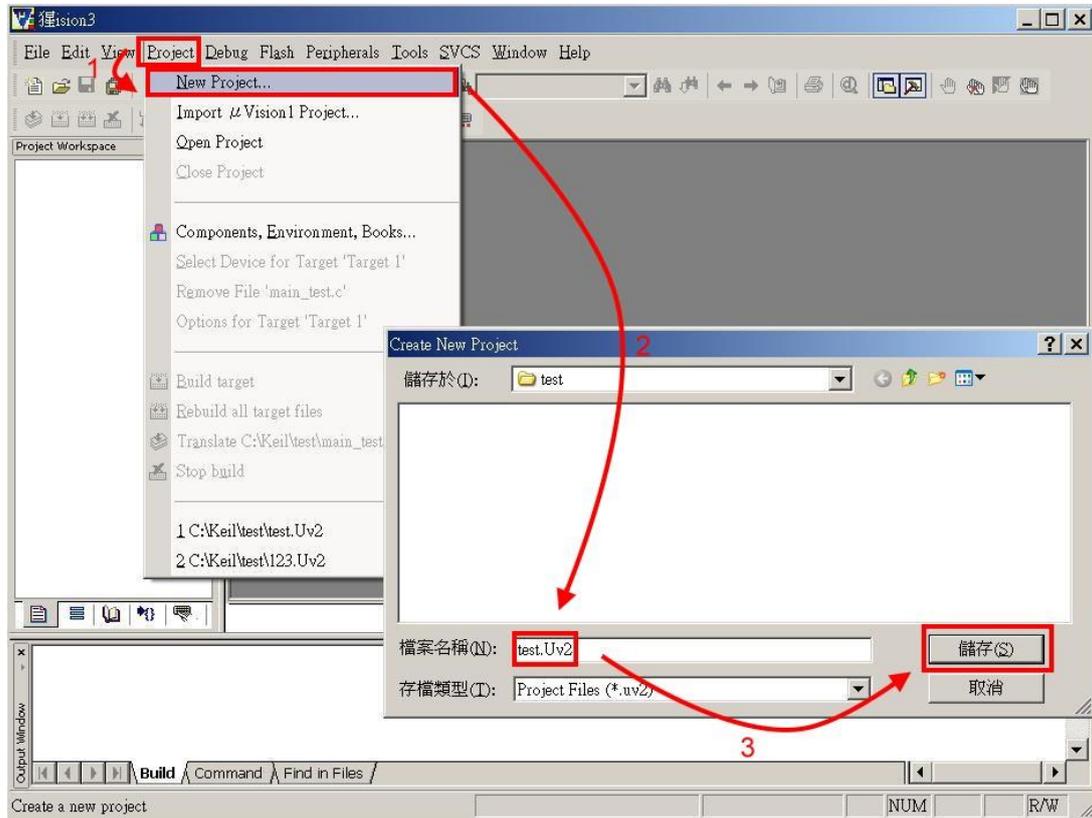


#### 3.2 确认设备管理器能识别到装置

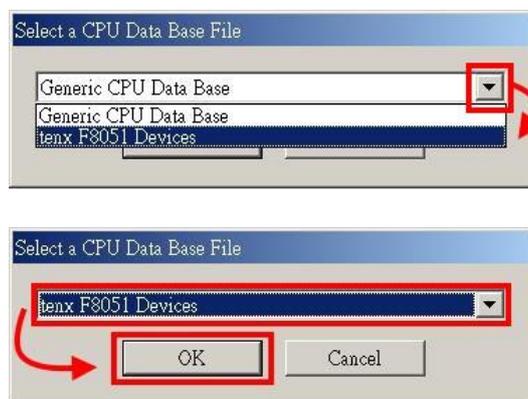


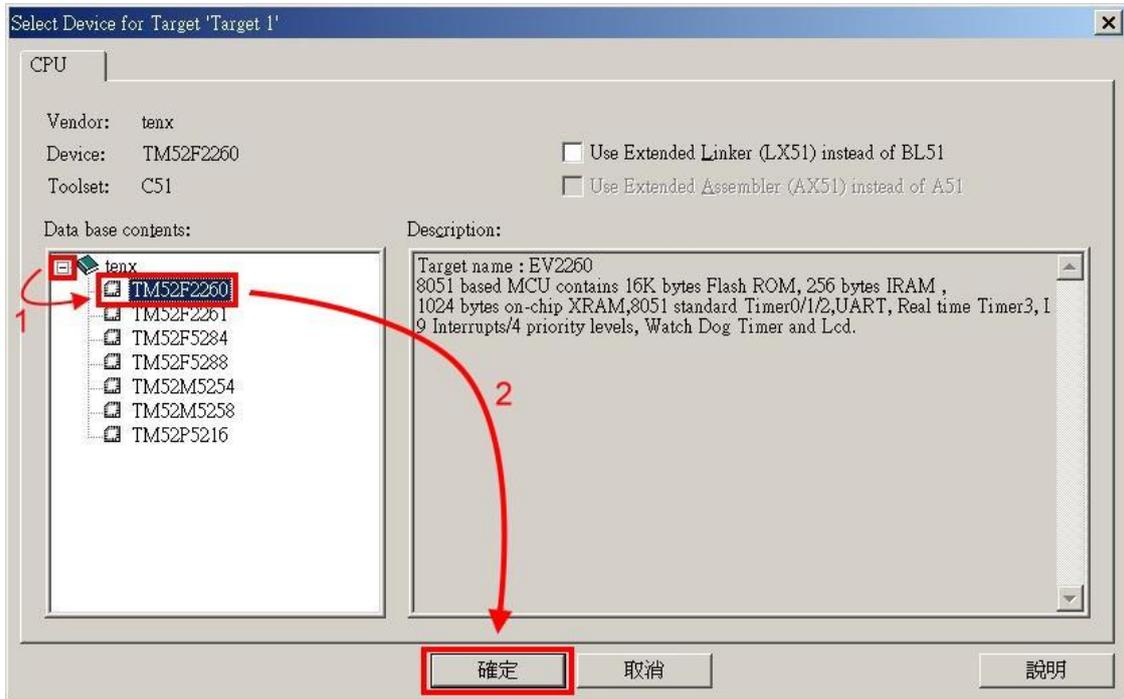
## 4. 开启 Keil C 设定编写程序

4.1 开启项目项目，点击菜单 Project 下 New Project 命令出现新建项目窗口，键入项目程名称点击保存即可。



4.2 选择芯片型号，点击储存按钮后，弹出 CPU Data Base 选择对话框，在下拉选择框中选择 tenx F8051 Devices 后点击 OK 按钮，弹出芯片型号选择接口，选择所连接的 EV Board 型号，例：TM52F2260 (EV2260) 窗口右边显示了此型号单芯片的一些参数，点击确定按钮。（注意：附注为芯片型号对应的 EV Board 型号）

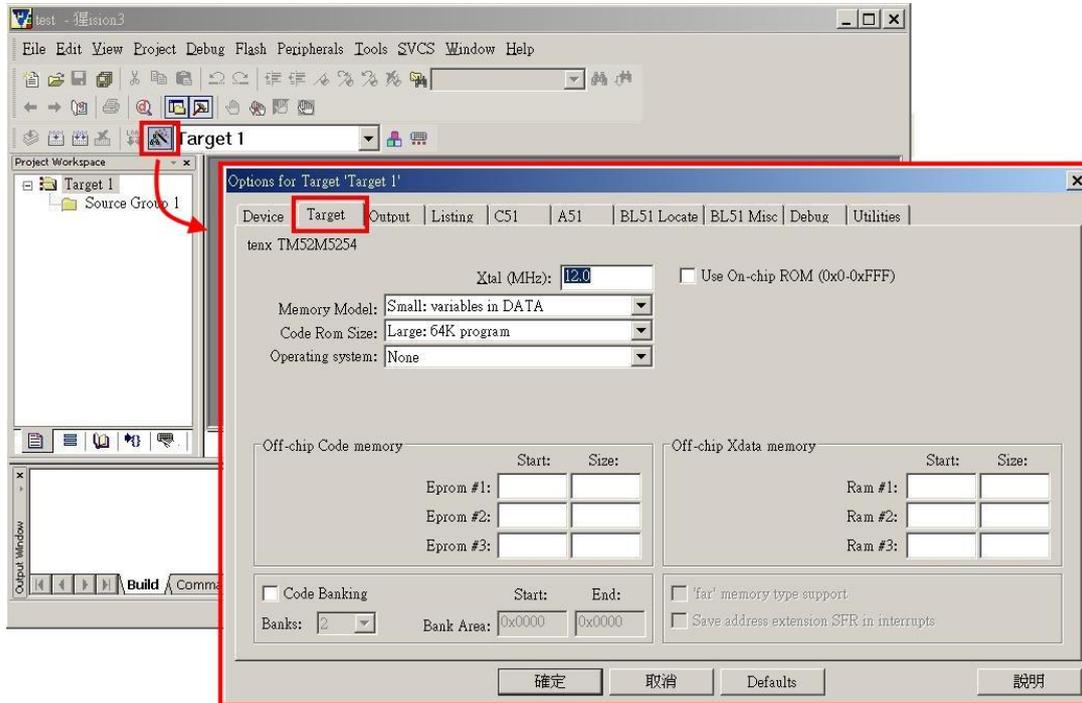




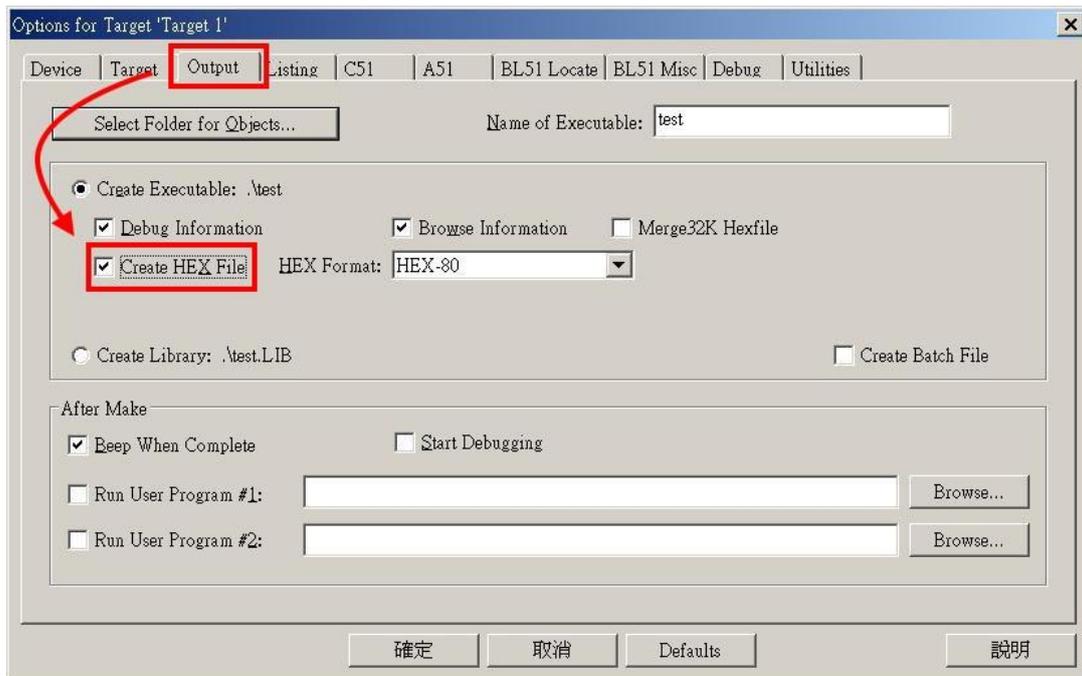
**附注(5.2) :芯片型号对应的 EV Board 型号**

芯片型号	EV Board 型号
TM52M5254 / TM52M5258	EV5254 / EV5258
TM52F5288 / TM52F5284	EV5288 / EV5284
TM52F2260	EV2260
TM52F2261 / TM52F2264	EV2261 / EV2264

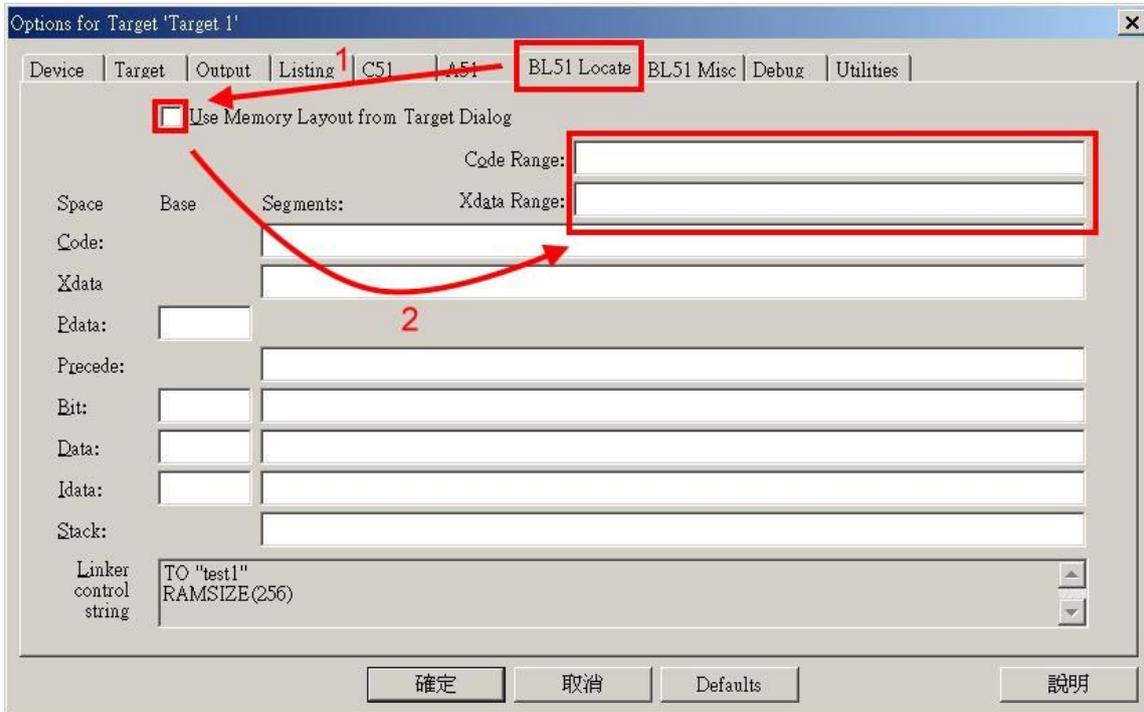
4.3 项目配置，项目文件创建后还必须对项目文件进行有关的设置，在主菜单中点击 Options for Target 项目设置方块，如下图所示，将弹出属性设置窗口。



4.4 在 Output 选项勾选 Create HEX File 选项，如下图所示：



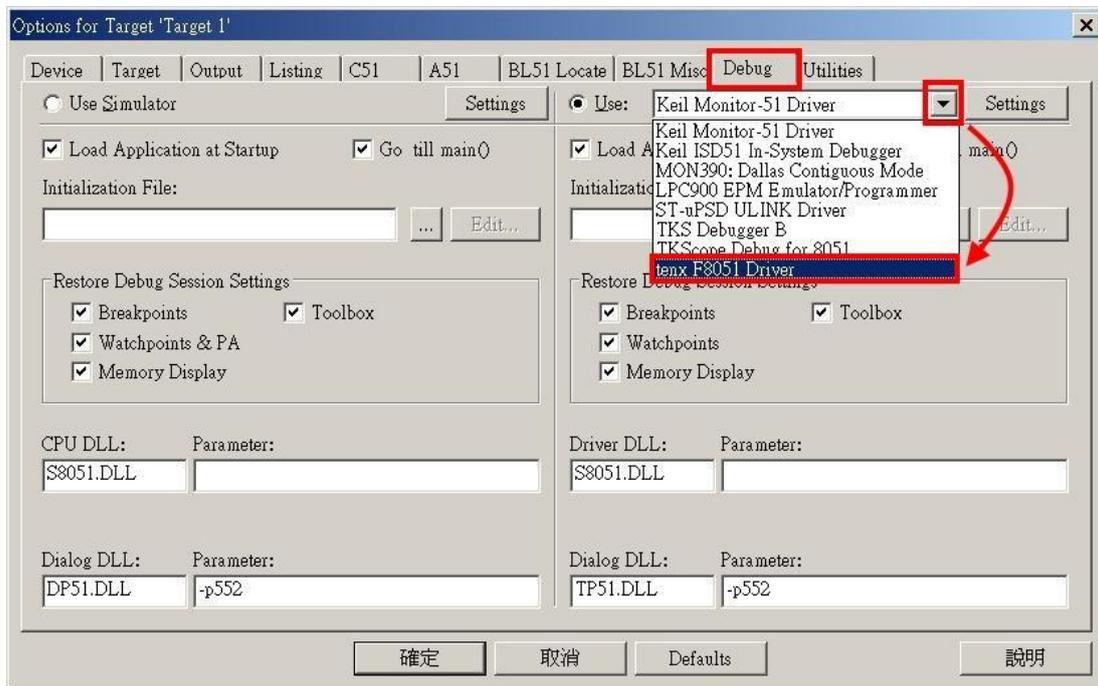
4.5 在 BL51 Locate 选项中取消勾选 Use Memory Layout from Target Dialog，填写 Code Range（也就是程序内存 ROM 区）及 Xdata Range(XRAM 区）。（注意：芯片的 Code Range，有分为不需调试模式  及调试模式  的 ROM 区可用范围，若需不同模式请记得至此区修改 Code Range，请参考附注(2)所示）



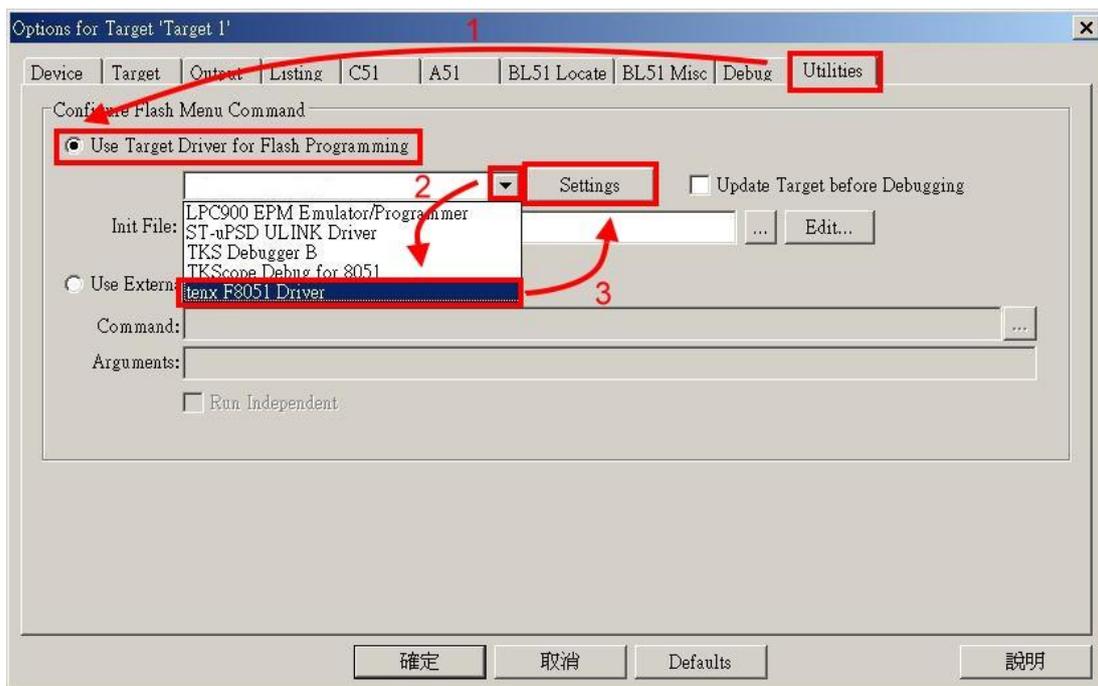
附注(4.5): Code Range 及 Xdata Range 可用范围

芯片型号	ROM 区可用范围		XRAM 区范围
	不需调试模式 	调试模式 	
TM52M5254 / TM52M5258	0x00-0x32,0x36-0xffff	0x00-0x32,0x36-0xffd	
TM52F5288 / TM52F5284	0x00-0x32,0x36-0x3ffe	0x00-0x32,0x36-0x3d00	0xff00-0xffff
TM52F2260	0x00-0x32,0x36-0x3ffe	0x00-0x32,0x36-0x3d00	0xfc00-0xffff
TM52F2261 / TM52F2264	0x00-0x32,0x36-0x3ffe	0x00-0x32,0x36-0x3d00	0xfe00-0xffff

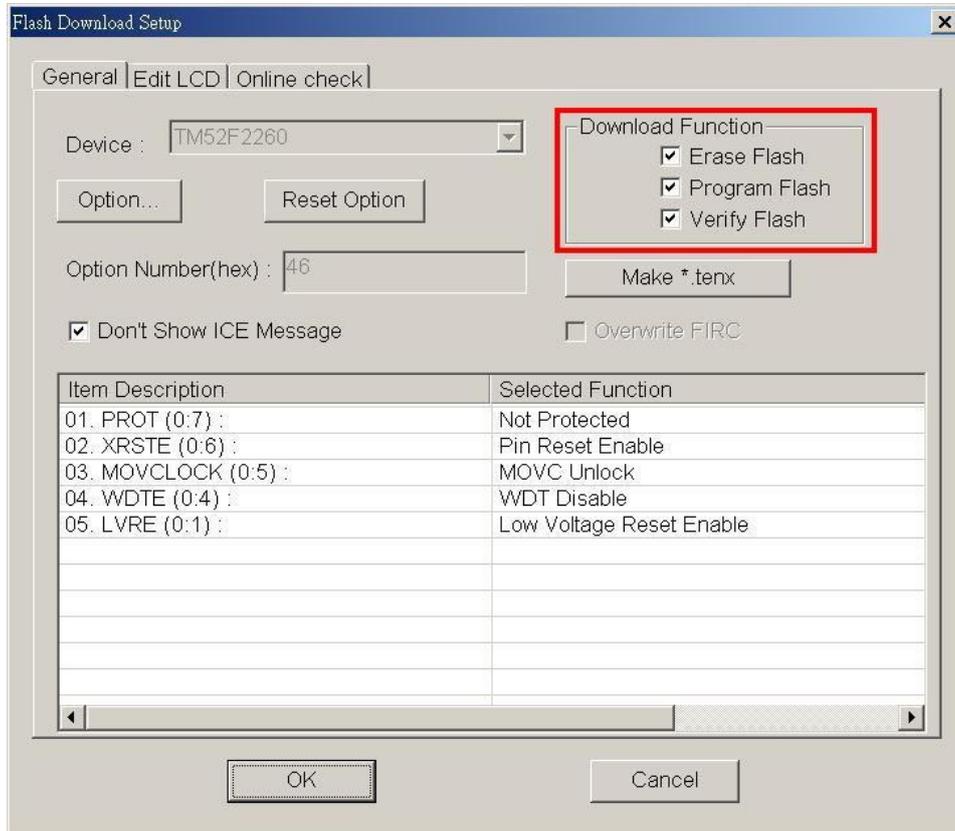
4.6 在 Debug 选项中选择使用仿真驱动 tenx F8051 Driver，如下图：



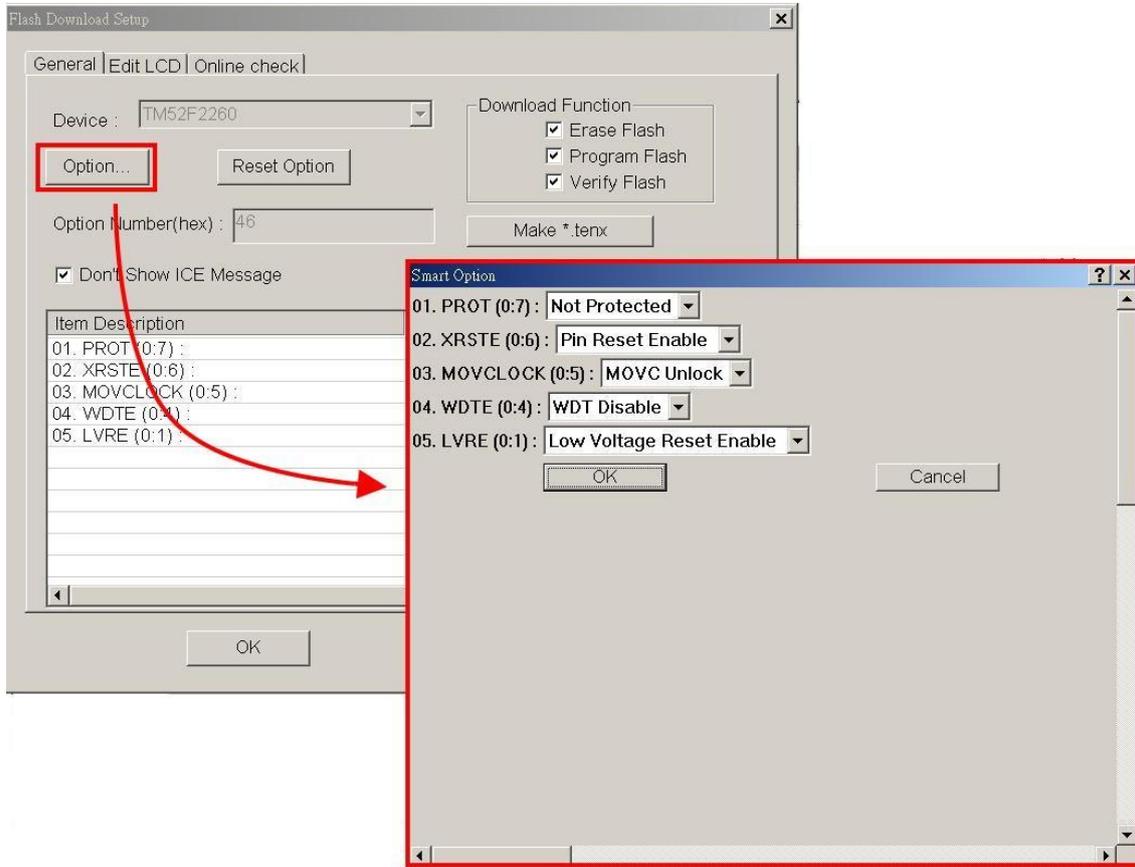
4.7 在 Utilities 选项中的点选 Use Target Driver for Flash Programming 并选择 tenx F8051 Driver，如下图所示：



4.8 点击上图中的 setting 按钮，弹出 Flash Download Setup 窗口，勾选 Download Function 选项，如下图所示(以 TM52F2260 为例)。

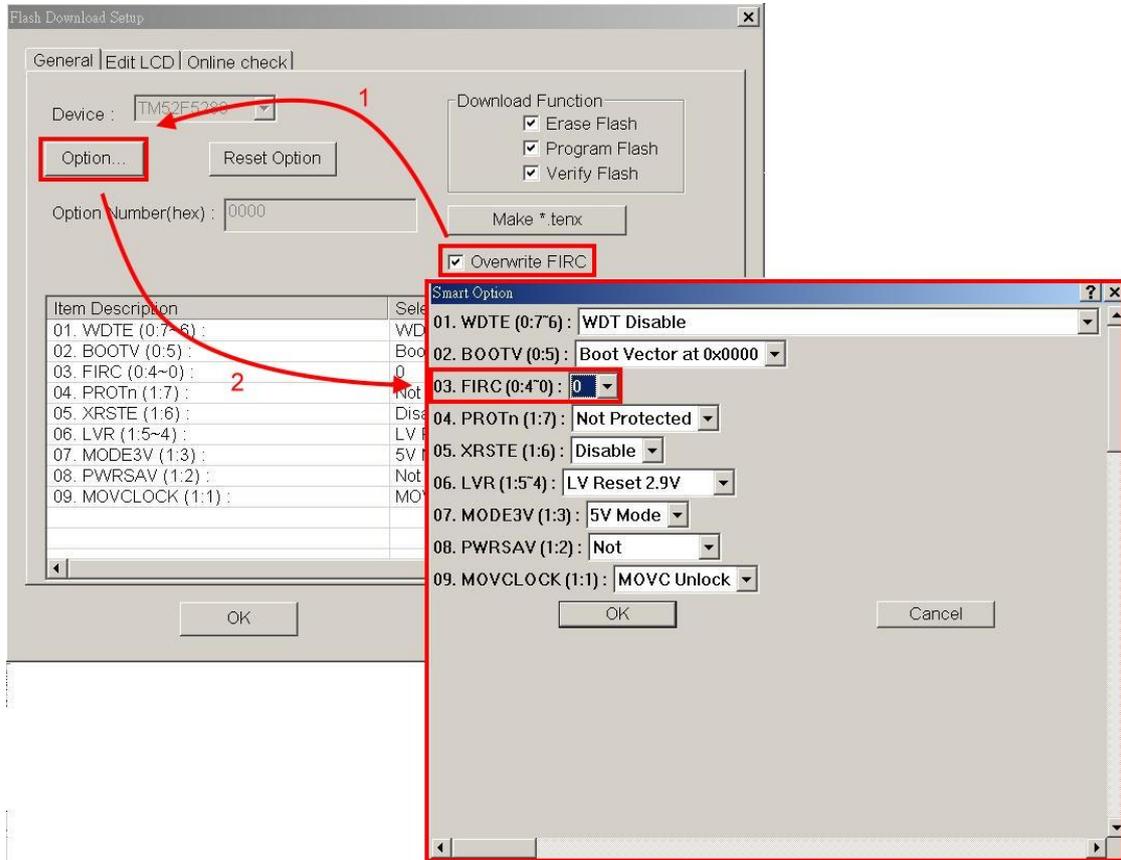


4.9 点选 option..按钮将弹出 CFGW 设置窗口，以 TM52F2260 为例：可以在此窗口修改 CFGW 的值，如下图所示，CFGW 各选项含义配置方法请参考芯片规格书相应章节，此处只作简单说明。



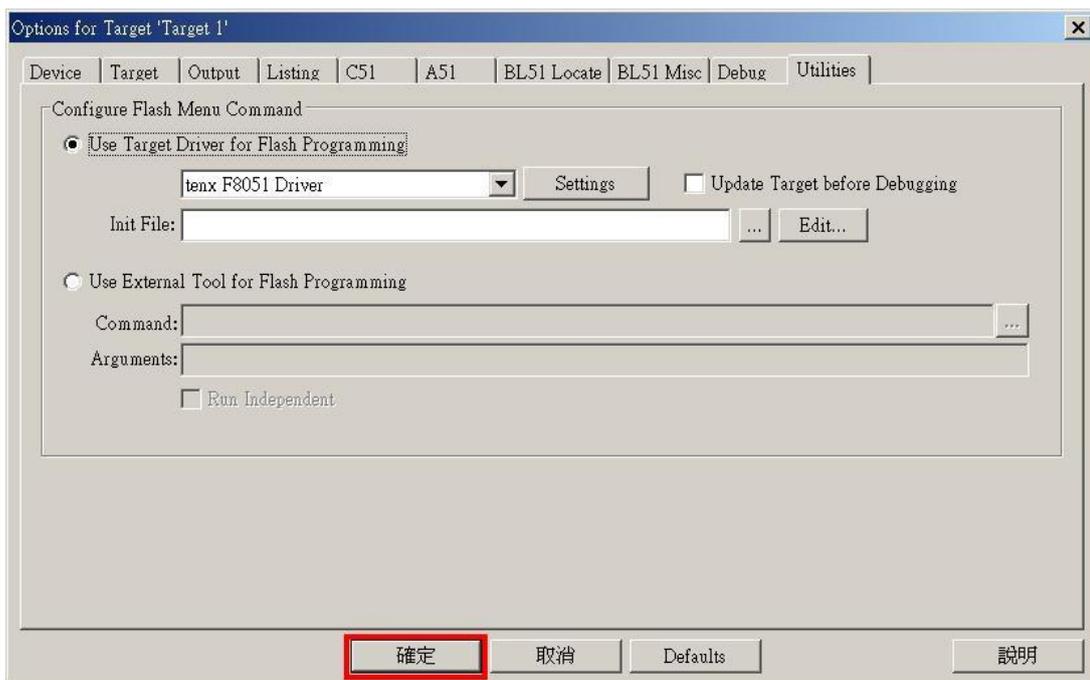
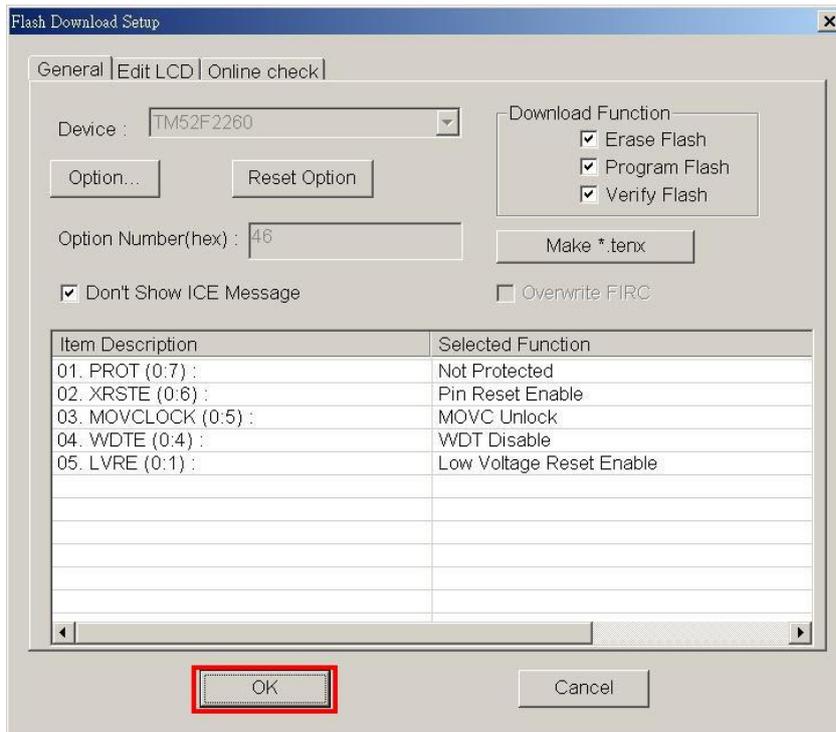
- PROT[0:7]:芯片刻录内容保护功能配置，Protect=开启芯片内容保护；Not Protected=不启用芯片内容保护。
- XRSTE[0:6]:芯片外部脚位重置功能配置，Pin Reset Enable=开启外部引脚复位；Pin Reset Disable=禁止外部引脚复位。
- MOVLOCK[0:5]: MOVX、MOVC 指令控制，MOVC Lock=禁止 MOVX、MOVC 指令，无法通过 MOVX、MOVC 指令寻址相应地址；MOVC Unlock=MOVC 指令可以访问其相应程序地址，MOVX 指令与 IAPLL 位状态配合访问其相应目标地址。
- WDTE[0:4]:看门狗功能配置，WDT Enable=开启看门狗复位；WDT Disable=禁止看门狗复位。
- LVRE[0:1]:低电压重置功能配置，Low Voltage Reset Enable=开启低电压重置；Low Voltage Reset Disable=禁止外部引脚复位和低电压重置。

4.10 CFGW FIRC (Fast Internal RC)，若有需要将 Overwrite FIRC 勾选后可进行 32 阶的频率调整。（TM52F2260/TM52F2261 无此功能）



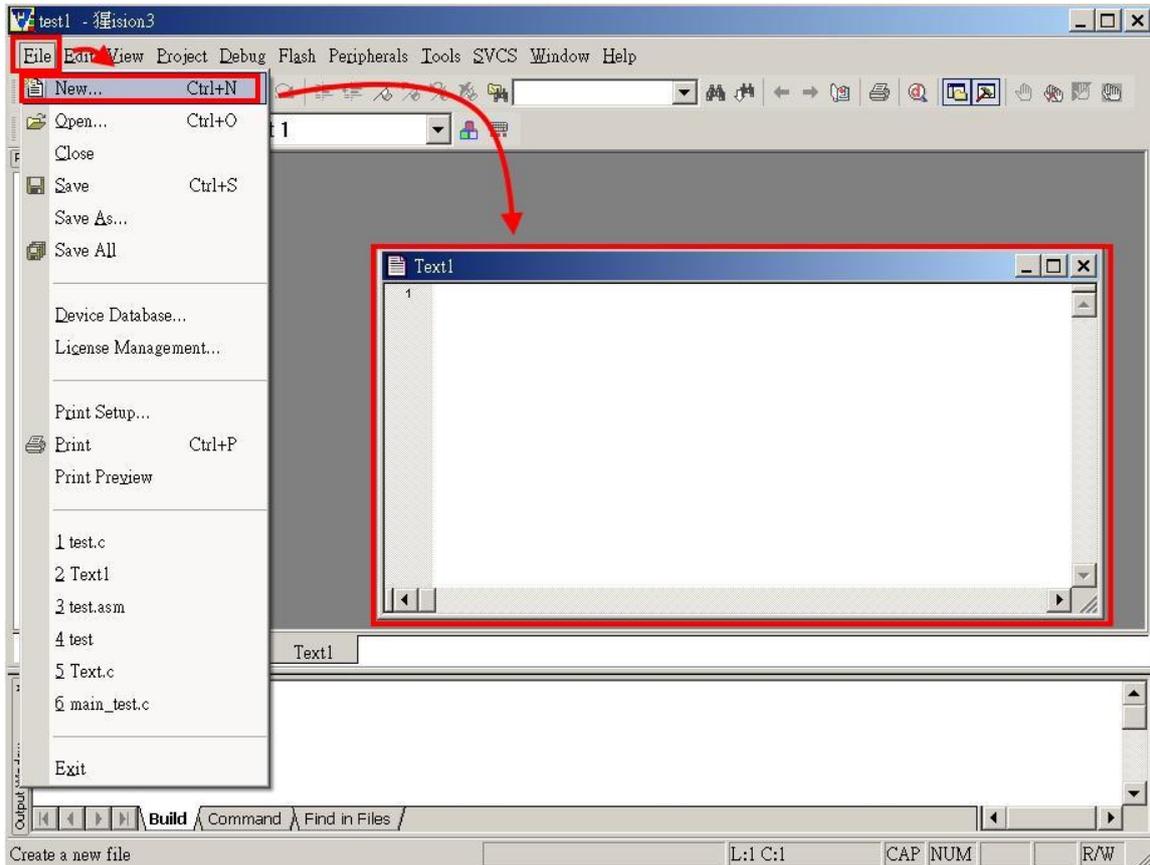
4.11 设定完成后，点击 OK 按钮返回 Flash Download Setup 窗口后点 OK 按钮返回 Utilities 窗口，到此，所有新建项目配置完成，点击确定按钮退出项目属性设置窗口，即可开始程序编写。

（注意：建立好的项目在使用中如需变更芯片型号，则需重新配置各型号相对应的 Device 及 Code range 及 Flash Download Setup 窗口，否则会造成无法正常使用的情况，如果是旧项目(Open Project)则需在 Flash Download Setup 窗口中按下 Reset Option，新项目(New Project)则会预设为 Reset 状态）

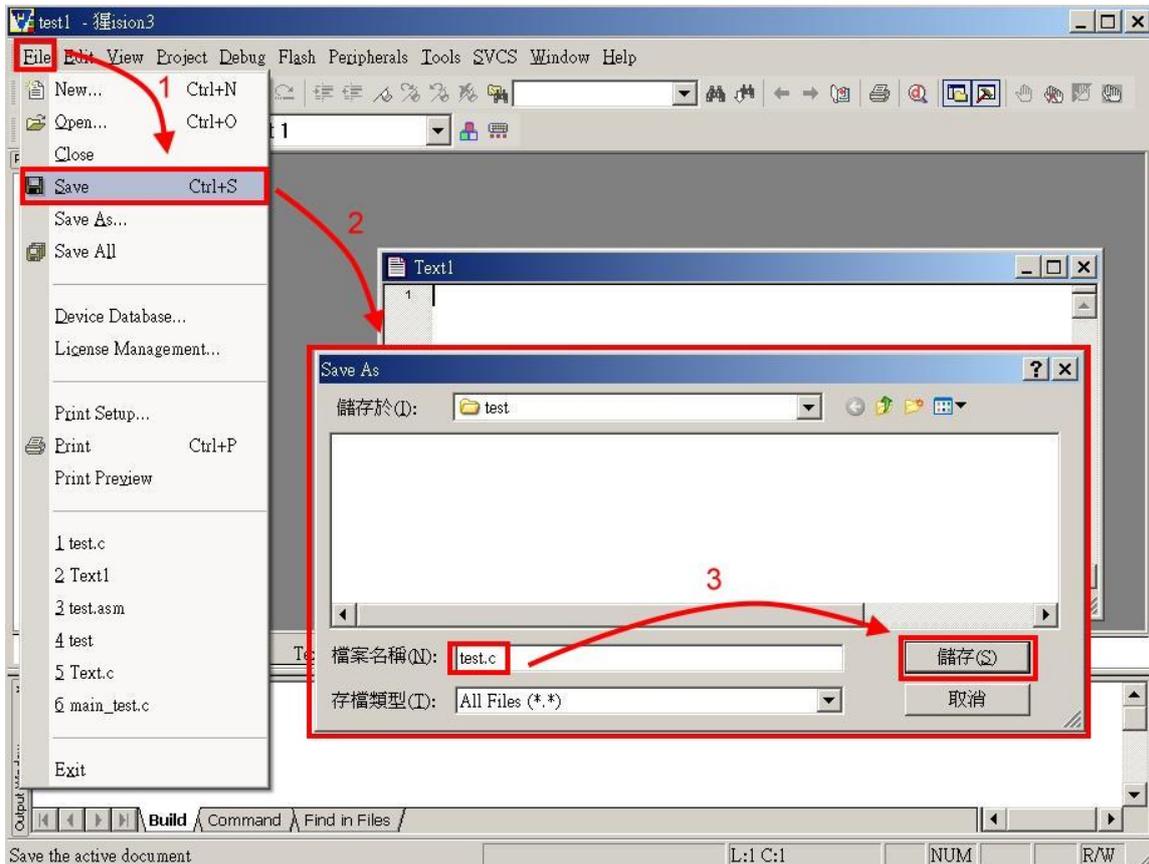


## 5. 编写程序代码

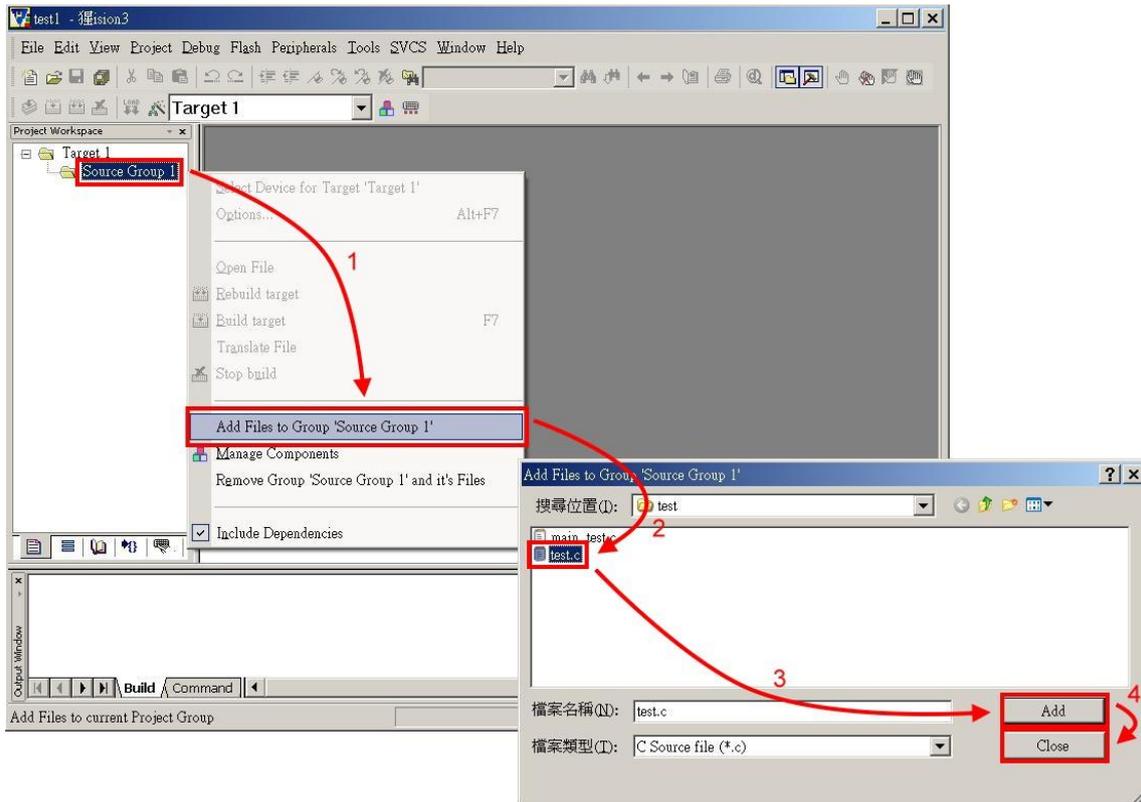
5.1 新建项目后在 Keil 主窗口点击新建文件图标创建一个空白文本框，或者主菜单 File 下点击 New 命令创建一个空白文本框，在此文本框输入编写的源代码，可以是 C 语言代码，也可以是汇编语言代码。



5.2 在 Keil 主窗口点击保存文件图标创建一个空白文本框，或者主菜单 File 下点击 Save 命令或 Save As...命令，在弹出的文件保存对话框中选择要保存的路径，在“文件名(N)”框中输入文件名，注意一定要输入档扩展名，如果是 C 程序文件，扩展名为 .c；如果是汇编程序档，扩展名为 .asm；如果是 ini 档，扩展名为 .ini。本例中档保存路径与项目文件一样，文件名为 TEST.c。单击保存按钮，完成源代码文件的新建。



5.3 上面虽然新建了一个项目文件，也新建了源代码档，但这项目文件盒源代码档还没有什么关系。下面需要将源代码档加入到项目文件中。点击 Target1 前面的“+”号，展开里面的内容 Source Group1，用右键点击 Source Group1，弹出 Source Group1 的右键快捷菜单，选择 Add Files to Group‘Source Group1’选项弹出 Add Files to Group‘Source Group1’对话框，选择所需的档案后单击 Add 按钮将其添加到项目中。通过对“文件类型”下拉菜单，可以选择不同的文件类型，如 .c 文件、.asm 文件等。如果要添加多个档，可以不断添加。添加完毕单击 Close 按钮，关闭该对话框。本例中将 test.c 档添加到专案中。

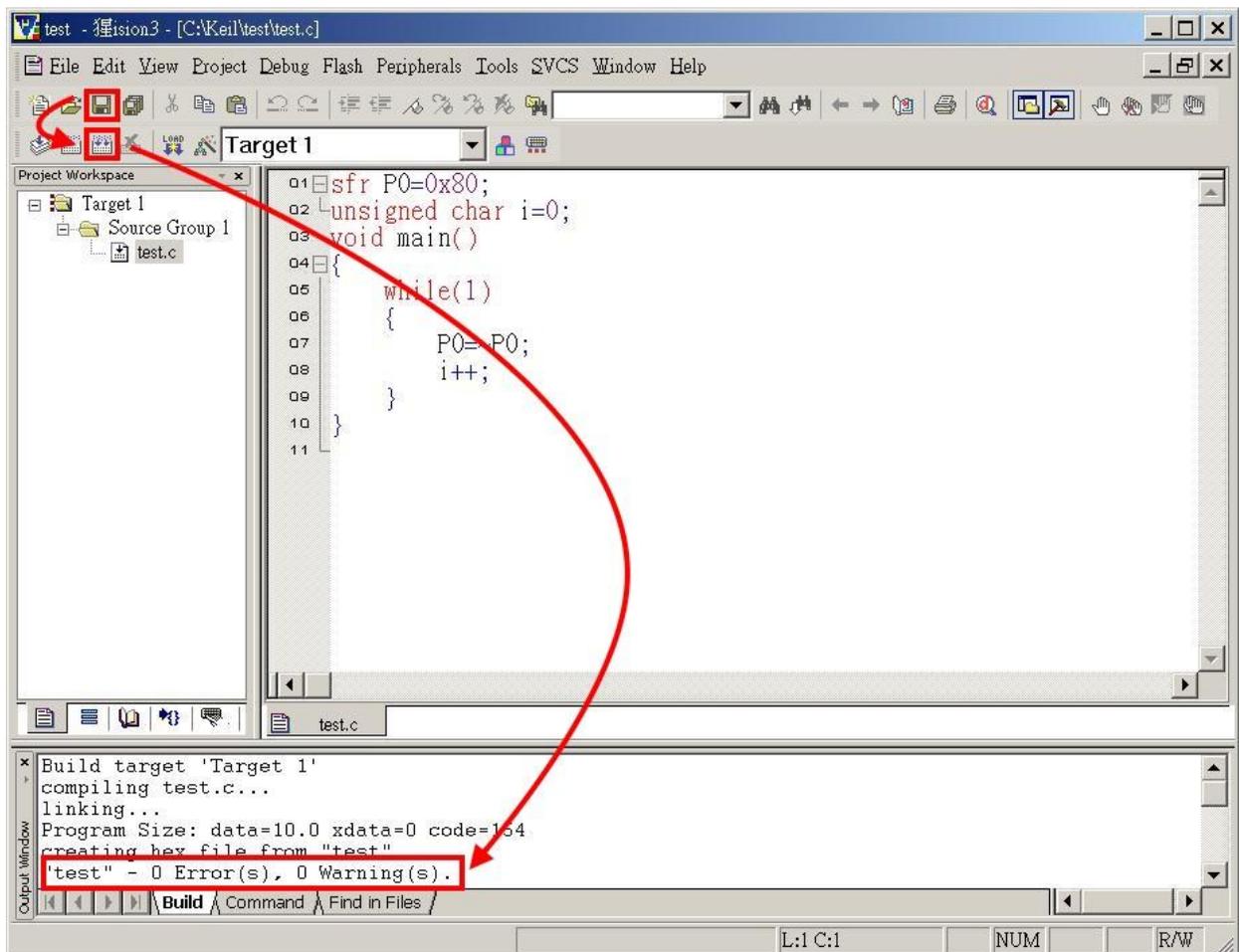


5.4 下面开始在代码编辑窗口编写源代码，本示例程序将实现在 P1 口输出方波，变量 i 自加 1，在 TEST.c 的编辑窗口键入以下代码：

```
sfr P0=0x80;
unsigned char i=0;
void main()
{
    while(1)
    {
        P0=~P0;
        i++;
    }
}
```

点击保存按钮保存后，点击编译按钮编译，结果如下图提示通过编译且无错误、无警告。

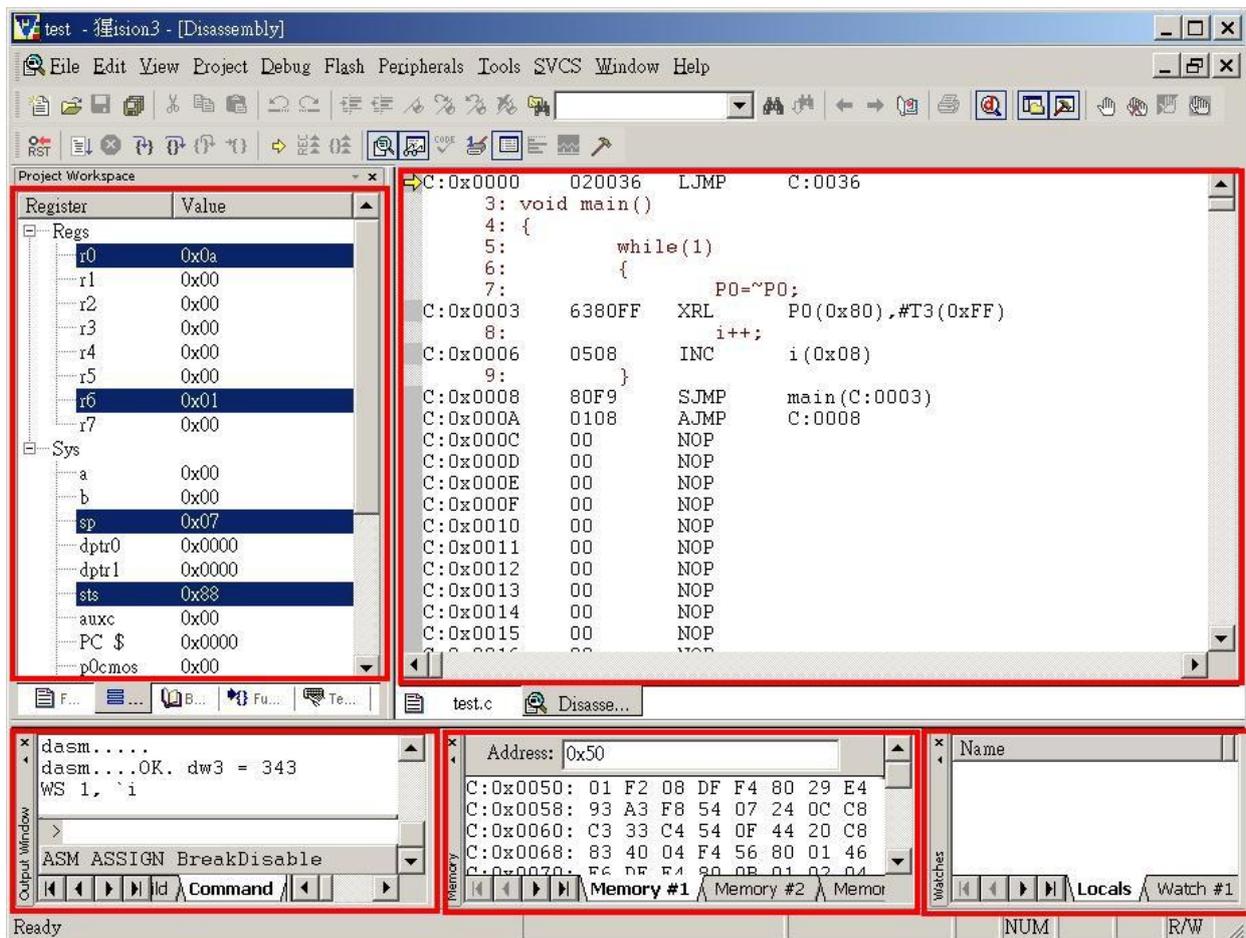
通过编译后，如果 EV Board 连接正确（关于硬件说明请参考各 EV Board 使用手册），即可利用 Keil 的仿真功能进行硬件在线仿真和刻录。



## 6. 模拟调试 (Debug)

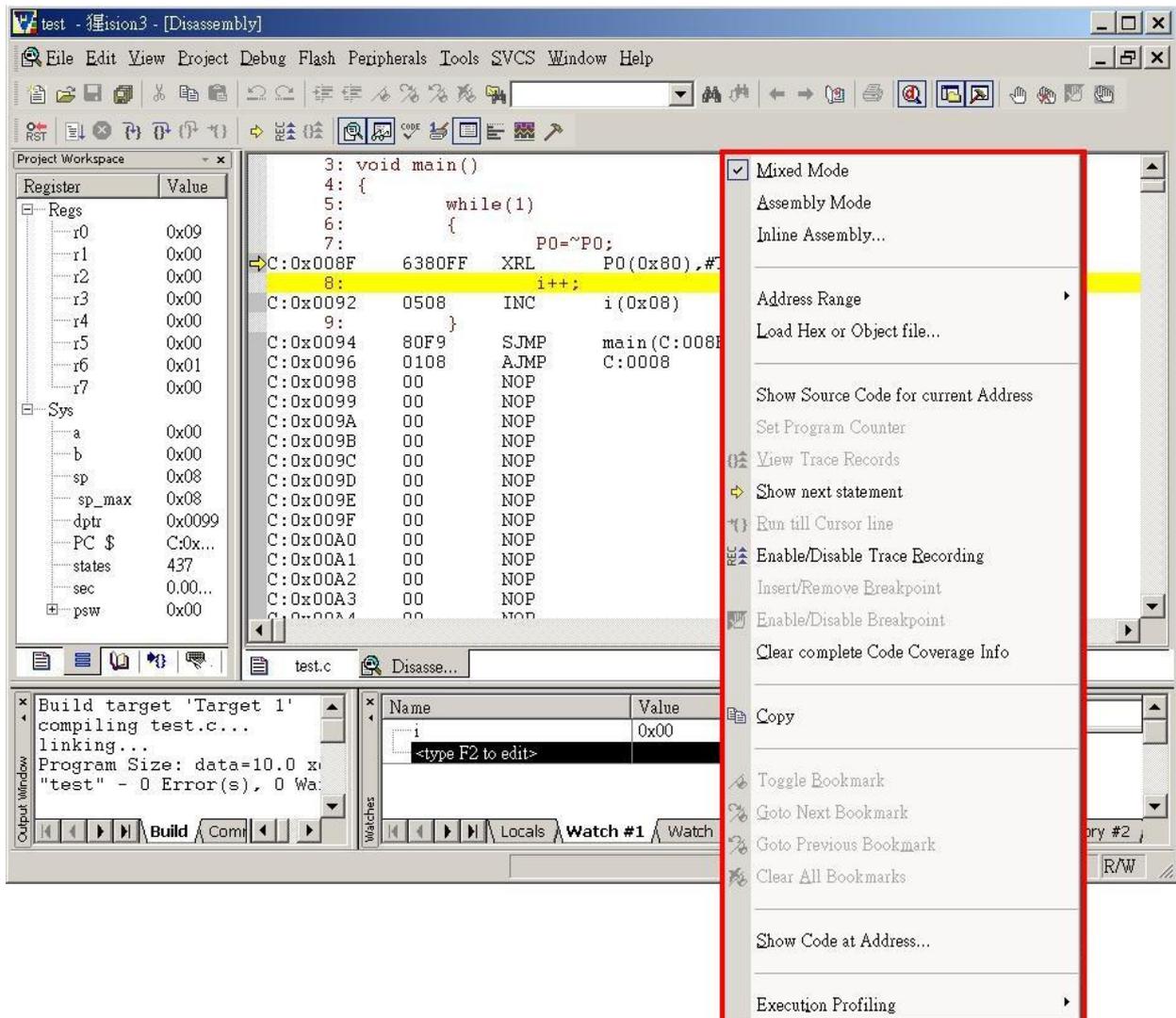
### 6.1 调试窗口

点击 Debug 菜单中的 Start/Stop Debug Session 或者按图示 按钮，程序将通过 USB 接口下载到单芯片 Flash，下载完成后进入如下图的 debug 接口，Keil 软件的调试程序提供多个窗口，主要包括反汇编窗口 (Disassembly Window)、观察窗口 (Watch&Call Stack Window)、内存窗口 (Memory Window)、代码报告窗口 (Code Coverage Window)、性能分析窗口 (Performance Analyzer) 等。通过 View 菜单下的相应命令打开或者关闭这些窗口。另外，项目窗口的 Register 选项和通过菜单命令 Peripherals 打开的一些窗口也是重要的调试窗口，将在下面将一起介绍。



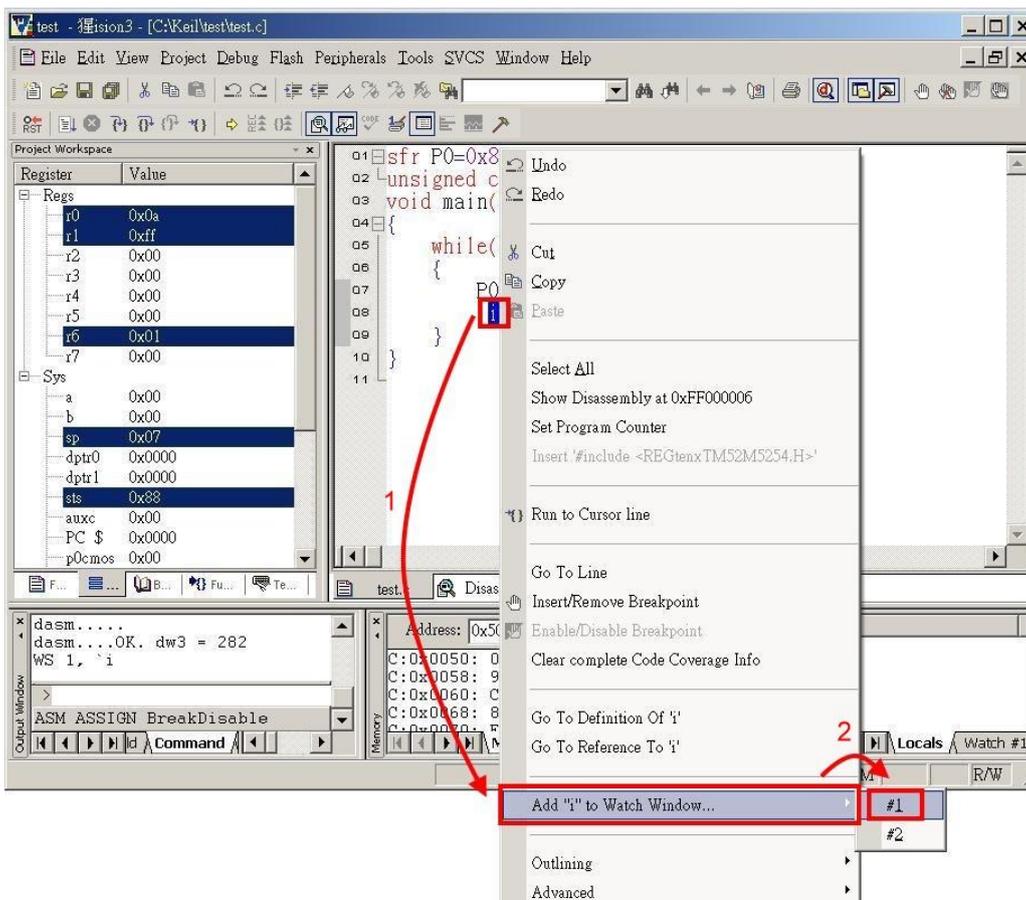
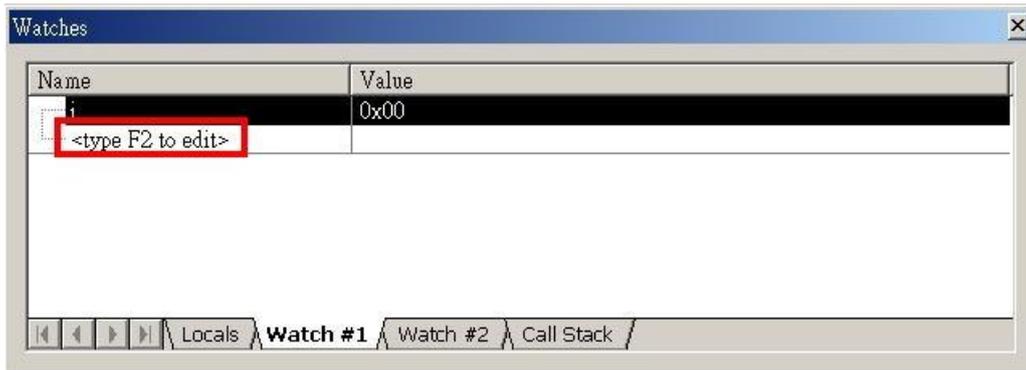
## 6.2 反汇编窗口

该窗口可以显示反汇编后的代码、源程序代码和相应反汇编代码的混合代码，在反汇编窗口点击右键会弹出下图菜单，选择 **Mixed Mode** 选项将采用高级语言与汇编语言混合显示方式；选择 **Assembly Mode** 选项将采用汇编语言方式显示；选择 **Inline Assembly** 选项将用在线汇编方式显示。右键菜单第二栏的 **Address Range** 选项用于显示用户程序的地址范围；**Load Hex or Object file** 选项用于重载 **Hex** 或 **Object** 档到集成工作环境中进行调试。第三栏是调试命令，将在后面给的调试功能项给予说明。第四栏和第五栏中的命令含义与 **Edit** 菜单相关命令相同。最后一栏 **Show Code at Address** 选项用于显示指定地址处的用户程序代码。



### 6.3 观察窗口

观察窗口又分四个选项，分别是 Locals、Watch #1、Watch #2、Call Stack。Locals 用于显示局部变量的值，但要注意的是，局部变量只有在它的有效区间内才会自动出现。Watch #1 和 Watch #2 用于观察用户自定义需要观察的变量，在此选项下按 F2 键可手动添加需要观察的变量，也可在 C 源程序处选中变量名后点击右键点击 Add “x” to Watch Window -> #1/#2(“x”表示变量名)。Call Stack 项用于显示程序执行过程中对子程序的调用情况。

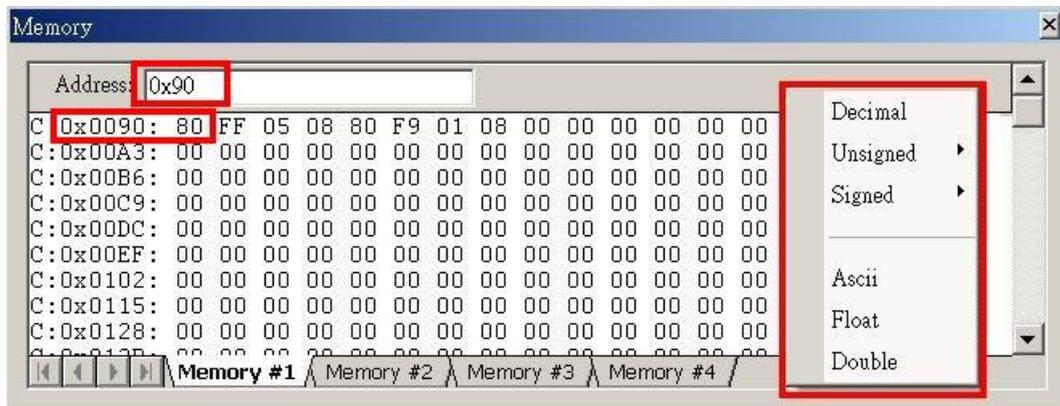


## 6.4 内存窗口

内存窗口同样包括四个选项卡，分别是 Memory #1、Memory #2、Memory #3、Memory #4。通过这些窗口可以观察不同存储区不同存储单元的值，方法是在 Address 栏输入相应的命令。当需要注意的是，在输入地址时要指定内存的类型。内存包括以下类型。

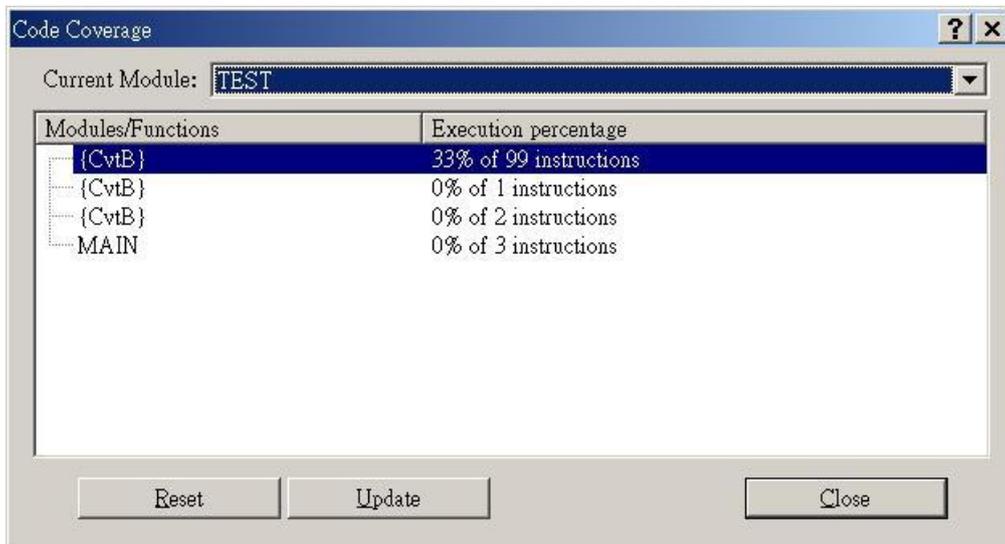
- D: DATA，可直接寻址的片内数据存储区。
- X: XDATA，外部数据存储区。
- I: IDATA，间接寻址的片内资料存储区。
- B: BDATA，可位寻址的片内数据存储区。
- C: CODE，程序存储区。

例如要观察可直接寻址的片内数据存储区 90H 的内容，只要在 Address 栏输入“C:0x90”，系统会在窗口给出从 90H 单元开始的，可立即寻址的内部数据存储器及相应的值，第一个就是 0x90 也就是 P1 埠的值，此时为 0x80。在内存窗口中单击鼠标右键，可以通过不同的菜单命令选择内存内容的显示方式。Keil 支持的显示方式有：十进制（Decimal）、无符号数（Unsigned）、有符号数（Signed）、ASCII 码、浮点型（Float）、双精度型（Double）等。通过鼠标右键菜单命令可以很方便地修改内存的内容。方法是先将鼠标指向希望修改的内存地址，再点击右键，选择 Modify Memory，下弹出的对话框中输入新的数值，单击 OK 按钮后完成某个内存单元内容的修改。



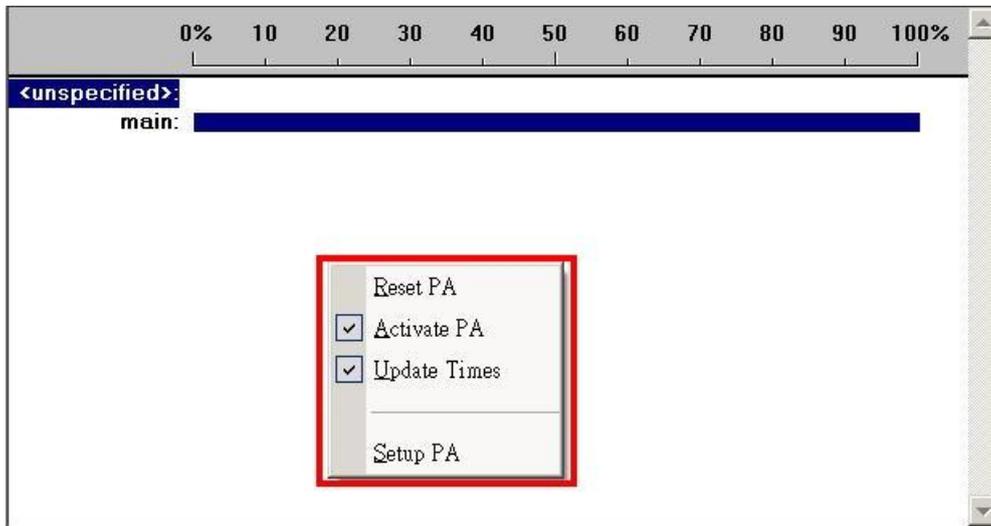
## 6.5 代码报告窗口

代码窗口用于显示指定用户程序模块的代码执行情况。在 **Current** 栏内键入指定的用户程序模块名称，改模块中各函数包含的指令条数及其已经执行指令的百分数将显示出来。按钮 **Update** 用于显示值进行更新，按钮 **Reset** 用于复位被执行指令的百分数。



## 6.6 性能分析窗口

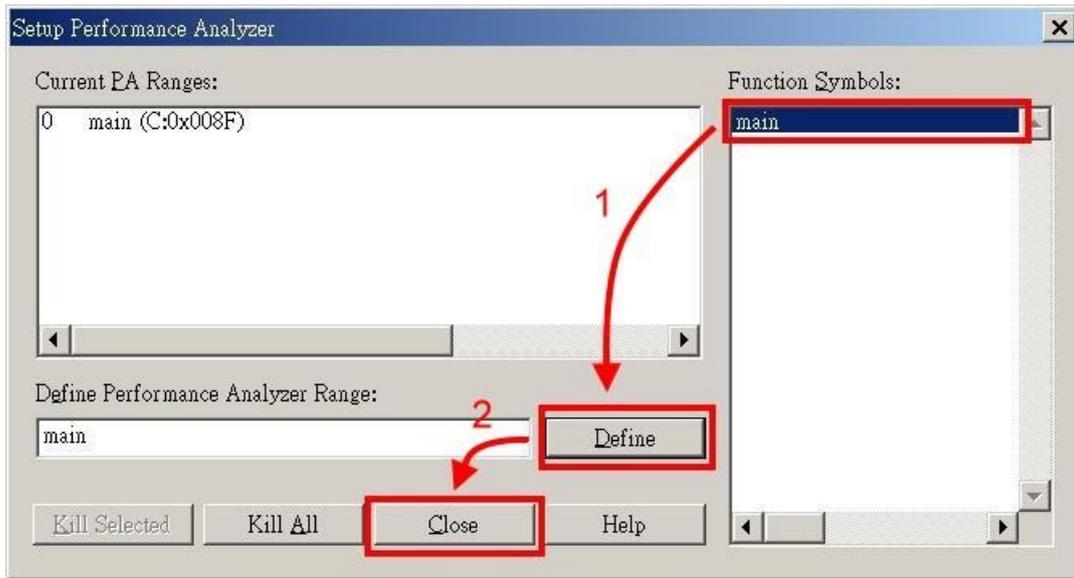
性能分析窗口用于显示指定程序模块被调用的次数及运行时间，分析结果以棒状图形显示在窗口中。通过窗口内的一个运行性能统计标尺很容易了解某个程序模块的运行性能。窗口下边显示有如下信息。



- min: 指定程序模块所消耗的最小时间。
- max: 指定程序模块所消耗的最大时间。
- avg: 指定程序模块所消耗的平均时间。
- total: 指定程序模块所消耗的总时间。
- %: 指定程序模块所消耗的时间占全部或运行时间的百分数。
- Count: 指定程序模块被调用的次数。

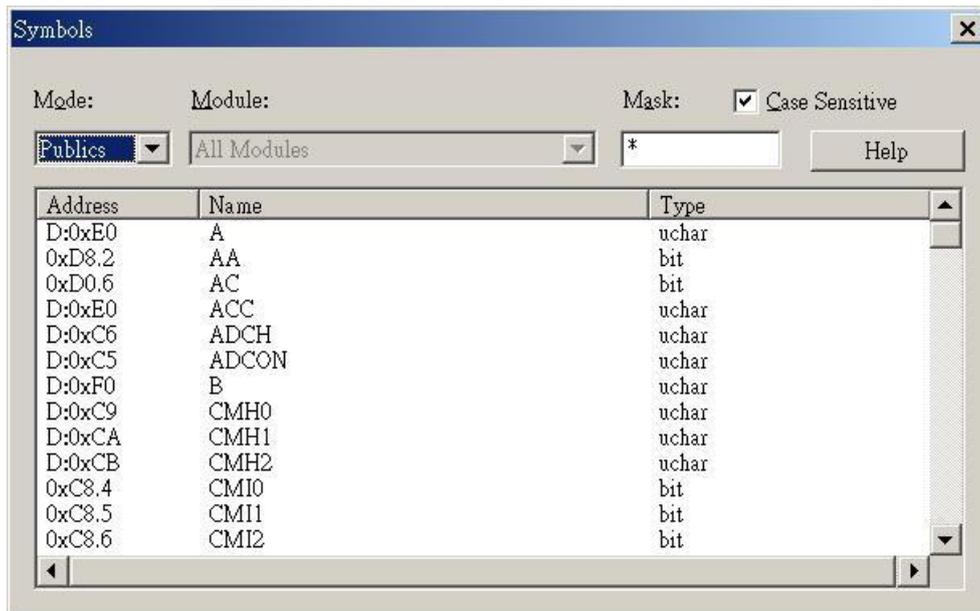
在性能分析窗口中点击鼠标右键，会弹出其右键快捷菜单，通过此菜单可以定制性能分析窗口的显示内容。选项 **Reset PA** 用于重置后程序模块的性能分析值；选项 **Activate PA** 用于启动/停止性能分析；选项 **Update Times** 用于更新程序运行时间；选项 **Setup PA** 用于设置性能分析窗口显示内容。

单击 **Setup PA**，会弹出下图窗口，其中 **Current PA Ranges** 栏显示的是当前已经设定的性能分析程序模块，单击 **Kill All** 按钮将删除全部已设定的模块，也可以用鼠标选择其中某一模块，再单击 **Kill Selected** 按钮删除指定模块。在 **Define Performance Analyzer** 栏中，再点击 **Define** 按钮即可将该其设定为一个性能分析模块。



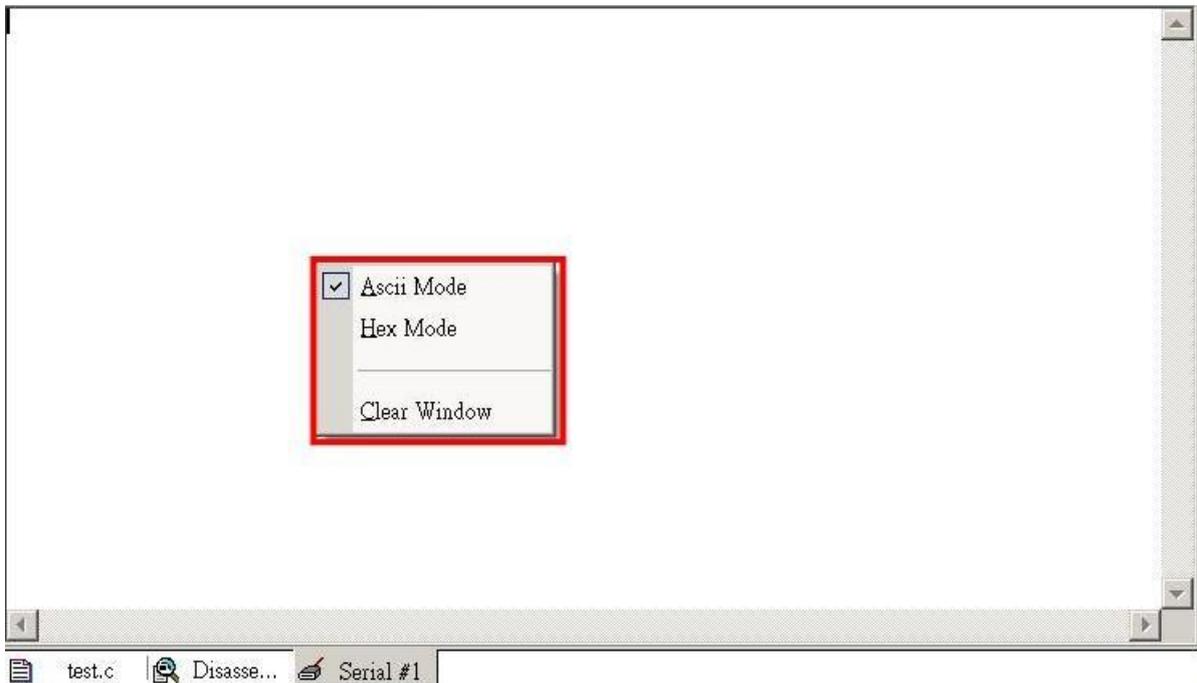
## 6.7 符号窗口

符号窗口用于显示当前程序中的各种符号。通过窗口左上角的 **Mode** 栏可选择显示公共符号（**Publics**）、局部符号（**Locals**）还是行号（**Lines**）。**Current** 栏用于选择当前的程序模块。在 **Mask** 栏中输入希望显示的符号名，可以使用通配符“\*”，点击 **Apply** 按钮确认，对应的符号信息将立即显示在窗口中。



## 6.8 串行窗口

串行窗口又两个：Serial Window #1 和 Serial Window #2。该窗口用作单片机串口的输入输出窗口，其中 Serial Window #1 用作输出窗口，Serial Window #2 用作输入窗口。如果仿真 CPU 通过串口发送字符，该字符虽然不会被显示出来，但却能传递到模拟 CPU 中。该窗口在进行用户程序调试时十分有用。如果用户程序中调用了 C51 的库函数 `scanf()` 和 `printf()`，则必须利用该窗口来完成输入输出的操作。在串行窗口中点击鼠标右键，可以选择窗口内容以 Hex 或 Ascii 格式显示。串行窗口中可以保持最近 8 KB 的资料，并可以进行翻滚显示。可以通过调用库函数 `printf()` 来完成，执行程序后，Serial Window #1 接口如下图。



### 6.9 缓存器窗口

在项目中点击 Regs 选项即可打开缓存器窗口。缓存器窗口包括两组缓存器：通用缓存器组 Regs 和系统特殊缓存器组 Sys。通用缓存器组报告 R0-R7 共 8 个缓存器。系统缓存器组除包括 A、B、SP、PC、DPTR、PSW 等传统 8051 系统特殊缓存器外，还增加 tenx F8051 系列单芯片扩展的系统特殊缓存器。这些缓存器是程序中经常使用的，并且对控制程序运行至关重要。通过观察这些缓存器值的变化将更加有利于用户分析程序。

The screenshot shows the 'Project Workspace' window with a 'Register' table. The table is divided into two main sections: 'Regs' and 'Sys'. Red boxes and arrows highlight specific parts of the table:

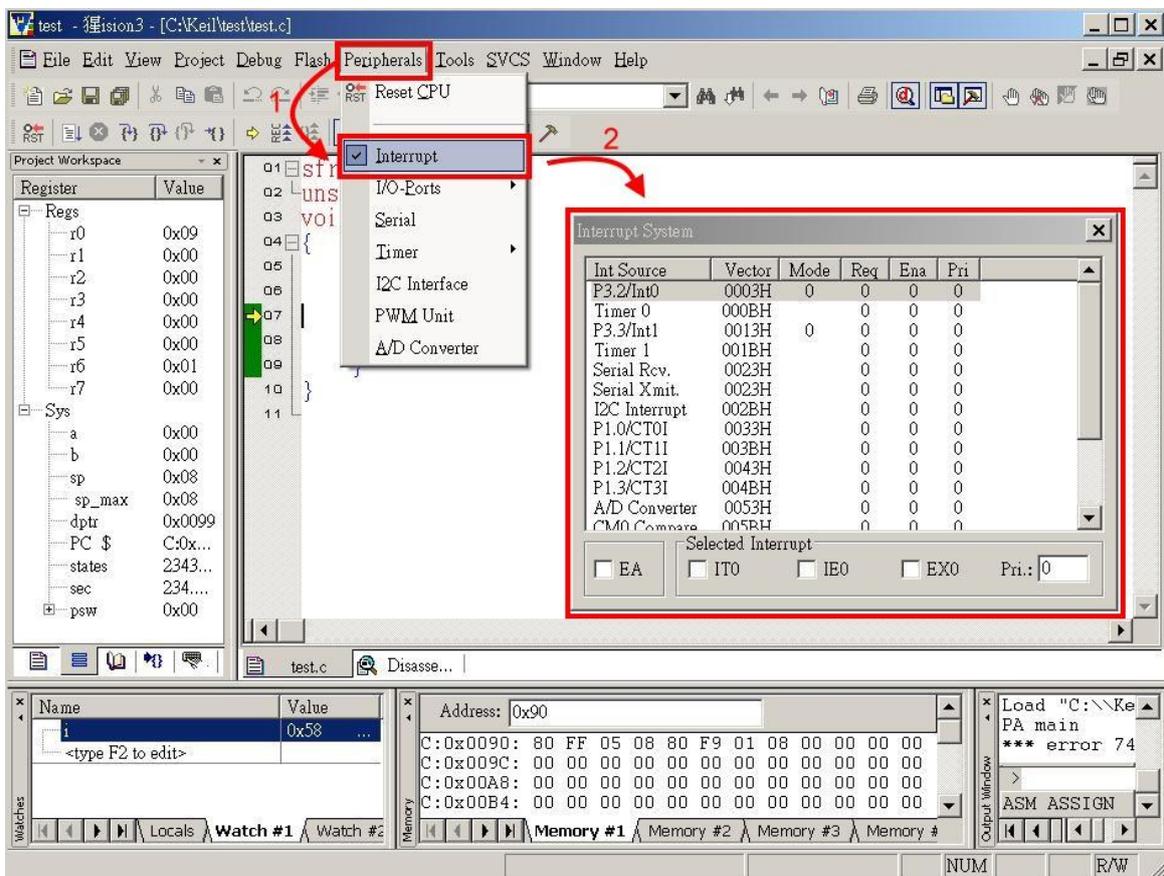
- A red box around the 'Regs' section (r0-r7) is labeled '通用暂存器' (General Registers).
- A red box around the 'Sys' section (a, b, sp, dptr0, dptr1, sts, auxc, PC \$, p0cmos, p0pun, clkctrl1, clkctrl2, intflg, plinte, pwm0duty, pwm0prd, pwm0ctrl, psw) is labeled '系统特殊暂存器' (System Special Registers).
- A red box around the 'auxc' through 'pwm0ctrl' entries in the 'Sys' section is labeled 'tenx F8051 扩展特殊暂存器' (tenx F8051 Extension Special Registers).

Register	Value
<b>Regs</b>	
r0	0x09
r1	0xff
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x01
r7	0x00
<b>Sys</b>	
a	0x00
b	0x00
sp	0x07
dptr0	0x0000
dptr1	0x0000
sts	0x88
auxc	0x00
PC \$	0x0000
p0cmos	0x00
p0pun	0x00
clkctrl1	0x00
clkctrl2	0x00
intflg	0x00
plinte	0x00
pwm0duty	0xff
pwm0prd	0x80
pwm0ctrl	0xff
psw	0x00

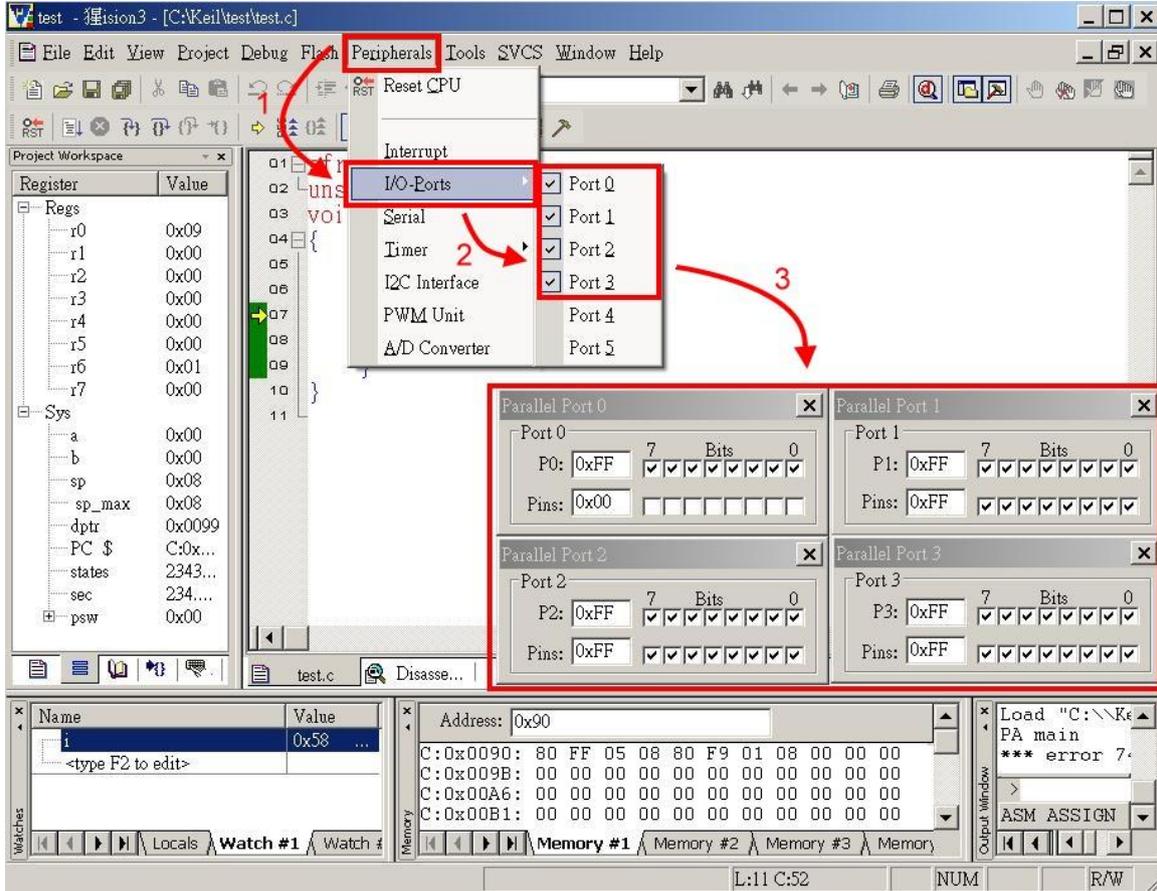
### 6.10 外围集成器件窗口

tenx F8051 系列单芯片有多种型号，不同型号的单芯片外围集成器件稍有不同，此处针对兼容传统 8051 的外围集成器件进行说明。在调试状态下，所有外围集成器件状态都可以通过菜单 Peripherals 下相应的命令来选择。下面介绍常用的外围集成器件状态观察窗口，如中断系统（Interrupt System）、并行 I/O 口（P0~P3）、串口（Serial Channel）和定时器/计数器（Timer/Counter）等。

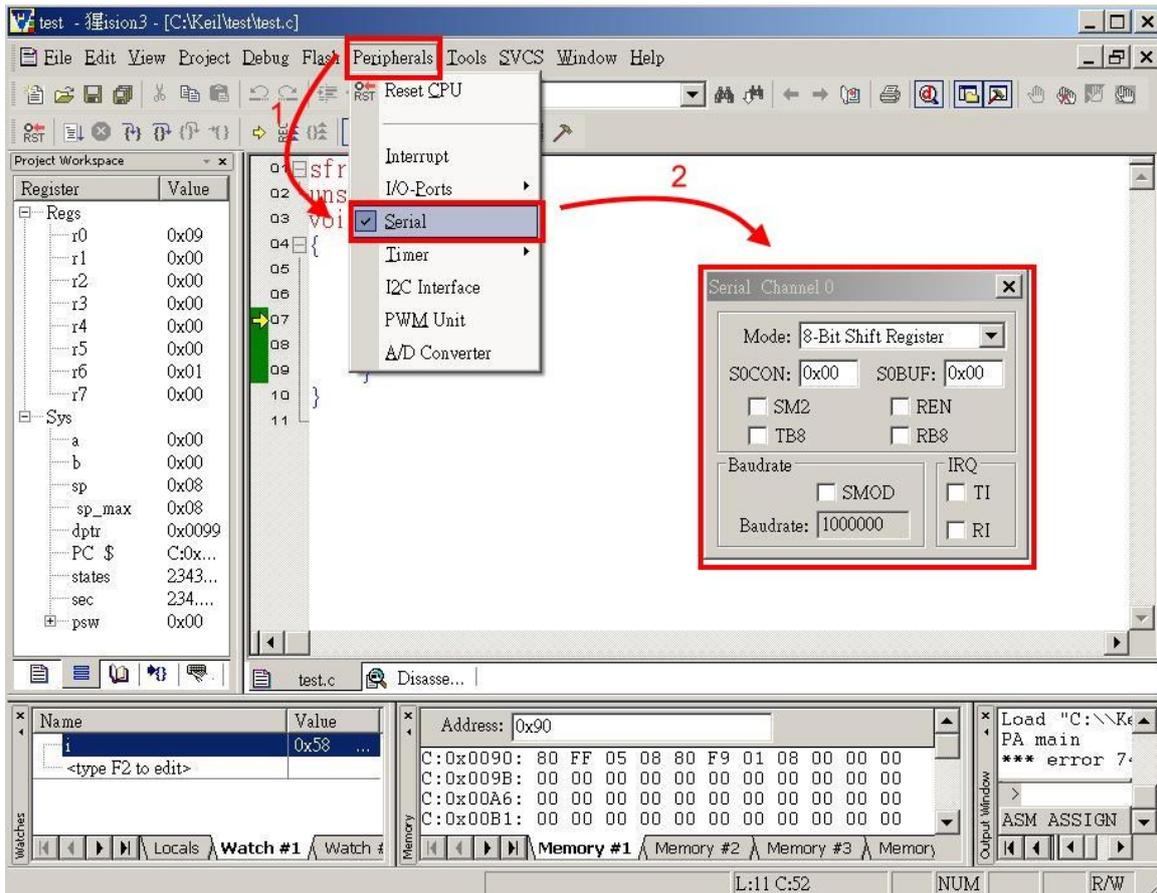
- 中断系统：中断系统状态观察窗口如下图所示。选中不同的中断源，窗口中 Interrupt 栏将出现与之相对应的中断允许和中断标志位的复选框，通过对这些状态位的置位和复位操作，很容易实现单芯片中断系统的仿真。



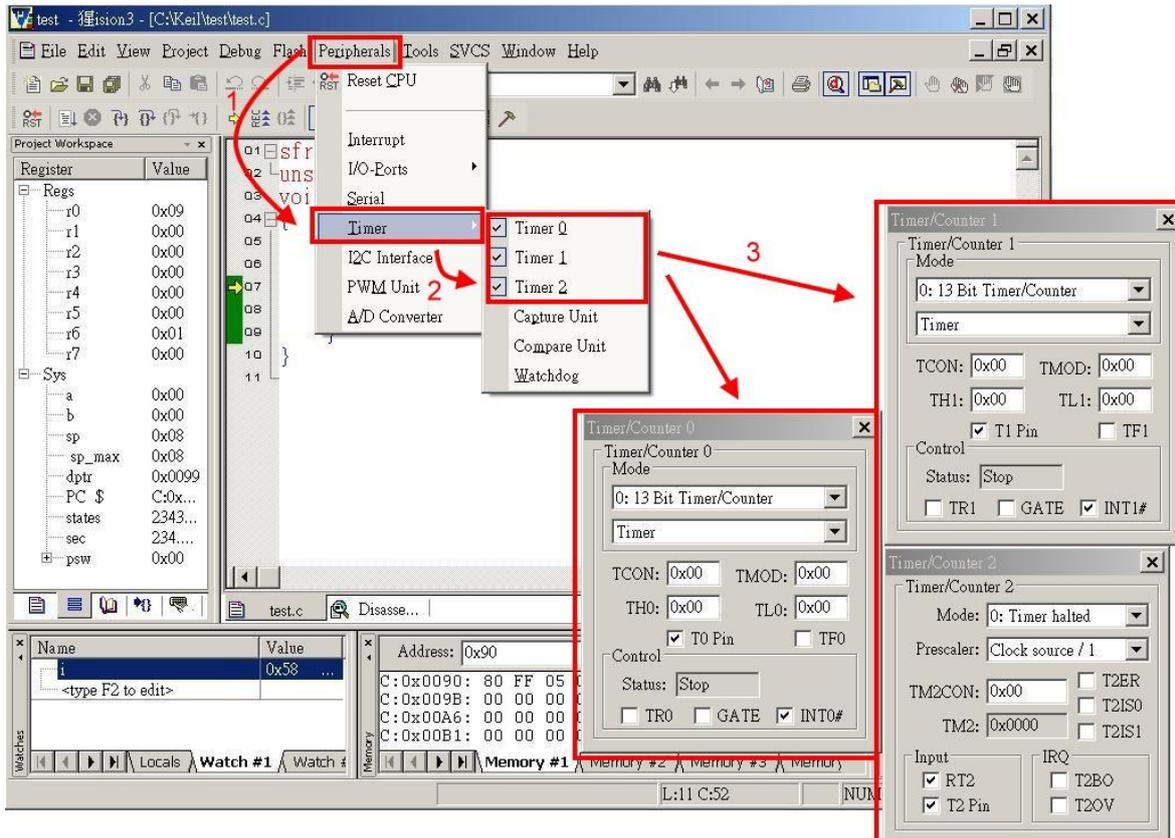
- 并行 I/O 口：并行 I/O 口观察窗口如下图所示。四个并行 I/O 口都是一样的（I/O 口应用及配置方法请参考芯片规格书章节），以 P0 为例，图中 P0 栏用于显示 P0 口锁存器状态，Bits 栏用于显示 P0 口各个引脚的状态。模拟时各位的状态都根据需要进行修改。



- 串口：串口状态观察窗口如下图所示。用户在程序中配置好串口工作方式后（串口工作方式配置方法请参考芯片规格书相应章节），仿真时在仿真窗口中 **Mode** 栏会显示配置好的串口口工作方式。相应的特殊工作缓存器 **SCON** 和 **SBUF** 的控制字也显示在窗口中。通过对特殊控制位 **SM2**、**REN**、**TB8**、**RB8**、**TI** 和 **RI** 复选框的置位和复位操作，很容易实现对单芯片内部串行口的仿真。**Baudrate** 栏用于显示串口的工作波特率，**SMOD** 位置位时将使波特率加倍，**IRQ** 栏用于显示串口的发送和接收中断标志。



- 定时器/计数器：**定时器/计数器观察窗口如下图。定时器/计数器 0 和 1 的窗口完全一致，以定时器/计数器 0 为例，用户在程序中配置好定时器/计数器 0 工作方式后（定时器/计数器工作方式配置方法请参考芯片规格书相应章节），仿真时在仿真窗口中 Mode 栏会显示配置好的定时器/计数器 0 工作方式。相应的特殊工作寄存器 TCON 和 TMOD 的字也显示在窗口中。TH0 和 TL0 用于显示计数初值，T0 Pin 和 TF0 复选框用于显示 T0 引脚和定时器/计数器 0 的溢出状态，。Control 栏用于显示很控制定时器/计数器 0 的工作状态。通过对这些状态位的置位和复位操作，很容易实现对定时器/计数器的模拟。



## 7. 调试命令

7.1 在 Keil 软件中，调试命令主要有两大类：一是程序运行控制命令，二是断点管理命令。另外还有其他一些的调试命令。这些命令的基本含义如下表格，后面介绍这些命令的使用场合和方法。

图 6 调试命令行表

Debug 菜单	工具栏	快捷键	描述
Start/Stop Debugging		Ctrl+F5	开始/停止调试模式
Go		F5	运行程序，直到遇到一个中断
Step		F11	单步执行程序，遇到子程序则进入
Step over		F10	单步执行程序，跳过子程序
Step out of Current function		Ctrl+F11	执行到当前函数结束
Run to Cursor line		Ctrl+F10	执行到光标所在的行
Stop Running		ESC	停止程序运行
Breakpoints			打开断点对话框
Insert/Remove Breakpoint			设置/取消当前行的断点
Enable/Disable Breakpoint			使能/禁止当前行的断点
Disable All Breakpoint			禁止所有的断点
Kill All Breakpoint			取消所有的断点
Show Next Statement			显示下一条指令
Enable/Disable Trace Recording			使能/禁止程序运行的轨迹
View Trace Records			显示程序运行过的指令
Memory Map			打开内存空间配置对话框
Performance Analyzer			打开设置性能分析的窗口
Inline Assembly			对某一行重新汇编，可修改汇编代码
Function Editor			编辑调试函数和调试配置文件

## 7.2 程序运行控制命令

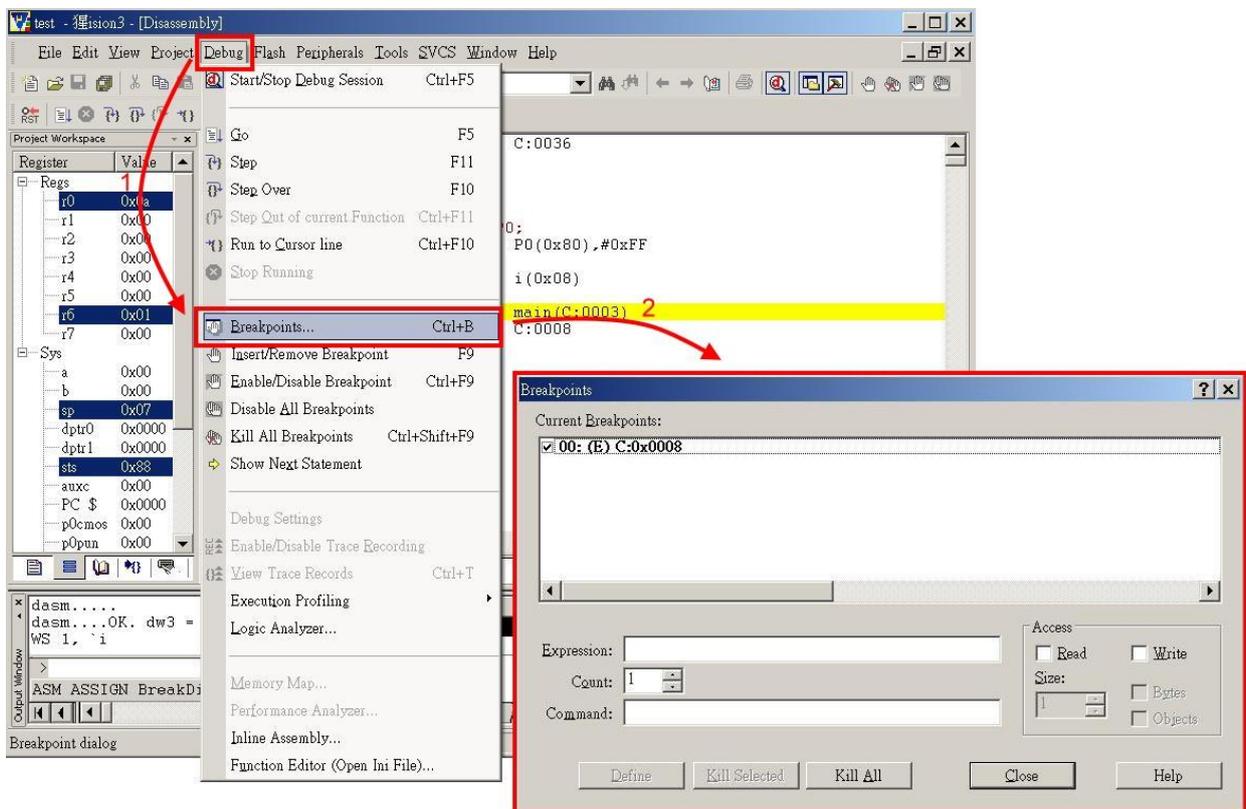
程序运行控制命令为用户提供了一种运行程序的方法，用户可以通过相应的命令来用全速、单步等多种方法运行程序。常见的程序运行调试命令有以下几种。

- **Go (F5)**：执行此命令将全速运行用户的应用程序。一般来说，此命令在软件模拟仿真时单独使用是没有意义的，但它和断点一起使用则能达到较好的效果。若在程序的关键处已经设置了断点，执行此命令后程序将运行到该断点处，且 PC 指针指向该程序并等待其他命令。
- **Step (F11)**：此命令可精确地控制程序的执行，此命令将执行当前光标指向的命令，根据当前的显示模式，它可以是一个 C 命令行或是一个单独的汇编行。如果这个命令执行的是函数调用则会跳到 C 函数或者子程序里面，使用户看到这个子程序里面包含的代码。
- **Step Over (F10)**：此命令将执行当前光标指向的命令语句。根据当前的显示模式，它可以是一个 C 命令行或是一个单独的汇编行，如果这个命令执行的是函数调用，该命令将执行完这个函数，而不进入函数内部。
- **Step out of Current function (Ctrl+F11)**：此命令用于跳出当前的子程序。当发现处于一个不感兴趣的函数中，并且希望快速返回至原来进入的程序时，这个跳出命令非常有用。
- **Run to Cursor line (Ctrl+F10)**：执行此命令可使程序执行到代码窗口中的当前光标位置处，这相当于把光标所在行作为一个零时断点。
- **Stop Running (Esc)**：执行此命令可以在一个不确定的位置中断，或停止正在运行的程序。

### 7.3 断点管理命令

断点功能对于用户的仿真调试十分重要，它可以在某个特定地址或是满足某种特定条件下暂停用户程序的运行，以便观察了解程序的运行状态、查找或排除错误。常用的断点管理命令有：插入/删除断点、启动/禁止当前断点、禁止所有断点、删除所有断点。对这些命令的基本描述见图 6 调试命令行表。

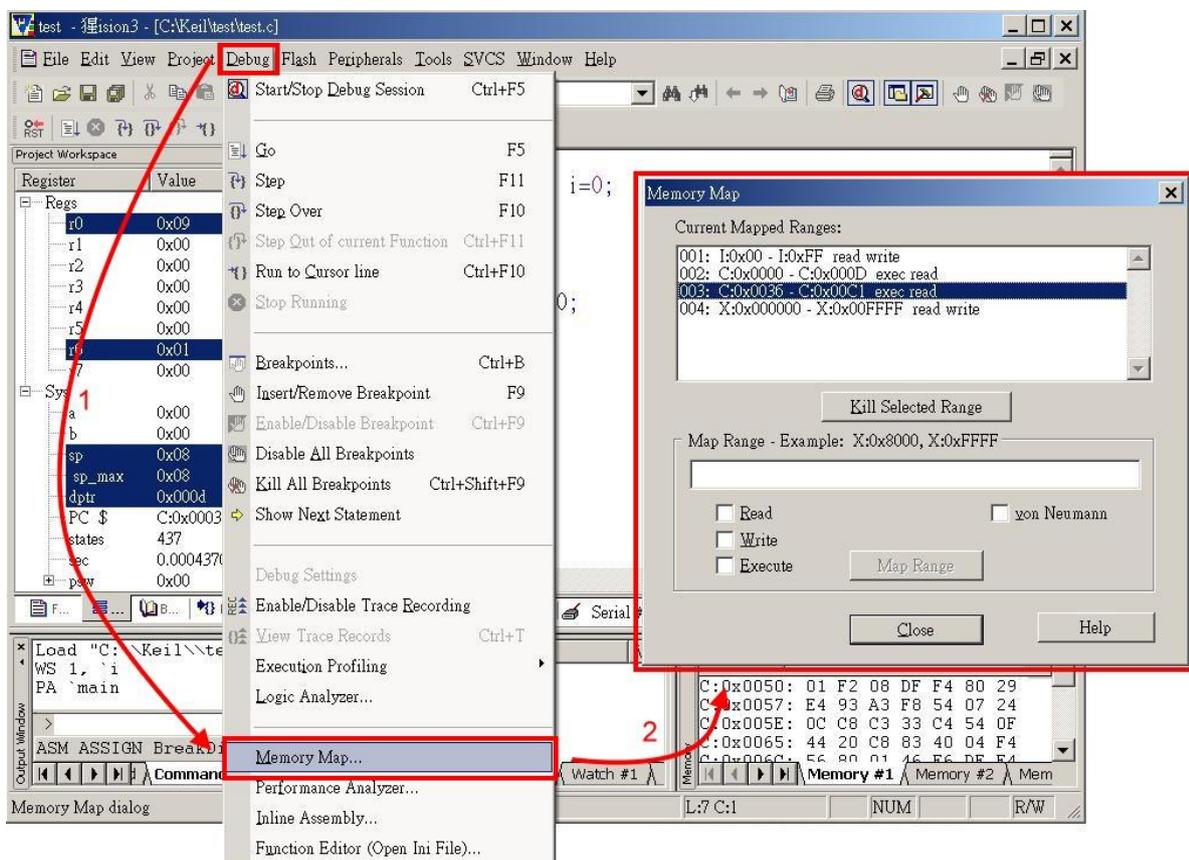
为了方便调试，Keil 软件还提供了一个断点管理窗口。执行菜单 Debug 下 Breakpoints 命令，将弹出下图所示的断点管理窗口。Current Breakpoints 栏中显示了所有已经设置的断点。在 Current Breakpoints 栏中选中某个断点，该断点有关的信息将在窗口下班的相关栏中显示出来。选中某个断点后单击 Kill Selected 按钮，可立即删除该断点，单击 Kill All 按钮将删除全部断点。



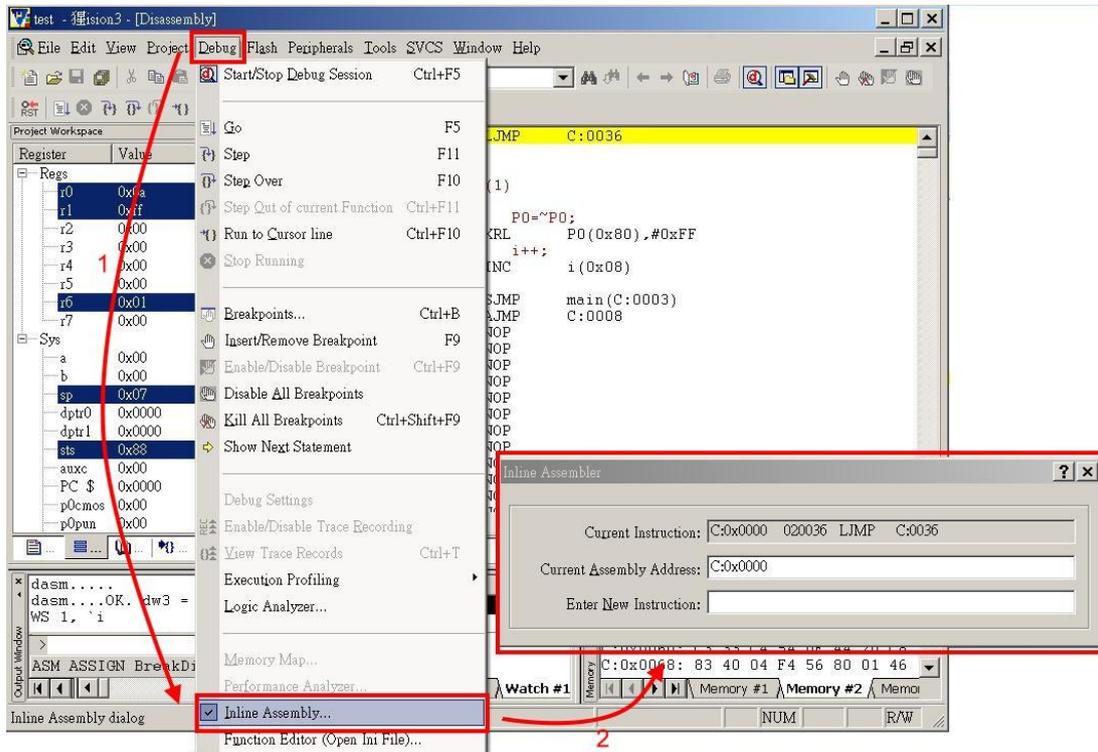
### 7.4 其他调试命令

除程序运行控制盒断点管理命令外，Keil 软件还提供了其他的一些调试命令，主要有：显示下一条指令、使能/禁止程序运行轨迹的标识、显示程序运行过的指令、设置内存空间镜像、设置性能分析的窗口、在线汇编、编辑调试函数很调试配置文件等。这些命令的基本描述见图 6 调试命令行表。设置性能分析窗口的的基本方法在之前已经述说过，下面逐一介绍其中比较重要的另外两条命令。

(8-4-1)设置内存空间镜像：选择菜单 Debug 下 Memory Map 命令弹出下图所示的内存镜像设置窗口。窗口中的 Current Mapped 栏显示所有已被设置的内存空间镜像，选择其中某一项并单击 Kill Selected Range 按钮，将删除选定的内存空间镜像，删除后用户程序不能对该部分内存进行操作，否则将会报 access violation 错误，如使用 tenx F8051 仿真器在线仿真此项不必理会。



**7.4.1 在线汇编：**所谓的在线汇编是指，修改程序后不必退出调试环境重新编译而使修改生效的技术。在线汇编大大方便了程序的调试，选择菜单 **Debug** 下 **Inline Assembly** 命令，启动在线汇编命令，会弹出如下图所示的对话框，在 **Enter New** 栏中输入希望的新指令，完成后回车，新键入的指令将取代指定行上原来的指令，并且自动指向下一条语句，可以继续修改，如果不再需要修改，单击右上角的关闭按钮关闭窗口即可。

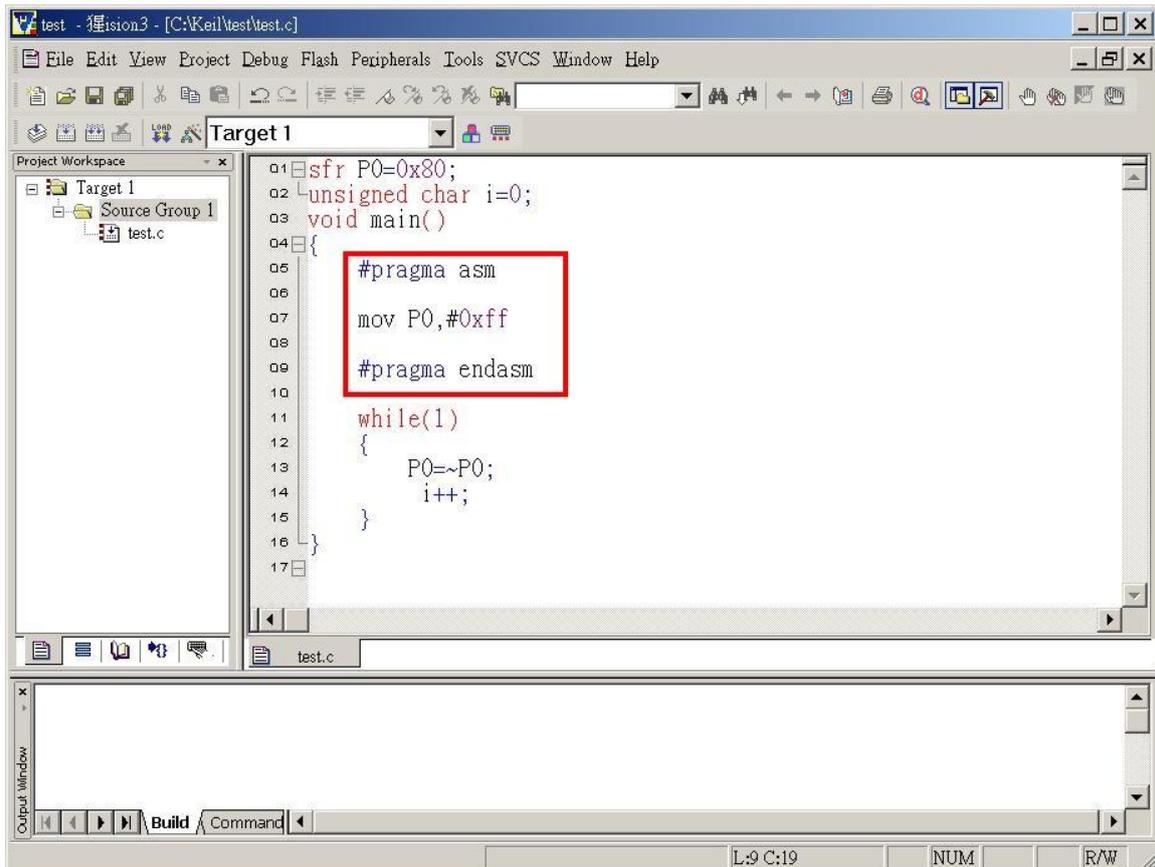


**7.4.2** 上面简单介绍了软件调试的基本方法，在实际使用过程中，可以灵活应用，或多种方法综合应用，以便更好地分析程序。项目文件经过在线调试之后，说明功能已经达到设计要求。下一步是将经过在线仿真后的程序加载到硬件目标板上加电运行，进入 **Debug** 接口时，程序已经刻录到单片机 **Flash**，因此退出 **Debug** 后，可以断开 **USB** 连接，直接给单芯片上电单独运行，用示波器观察实际运行结果。

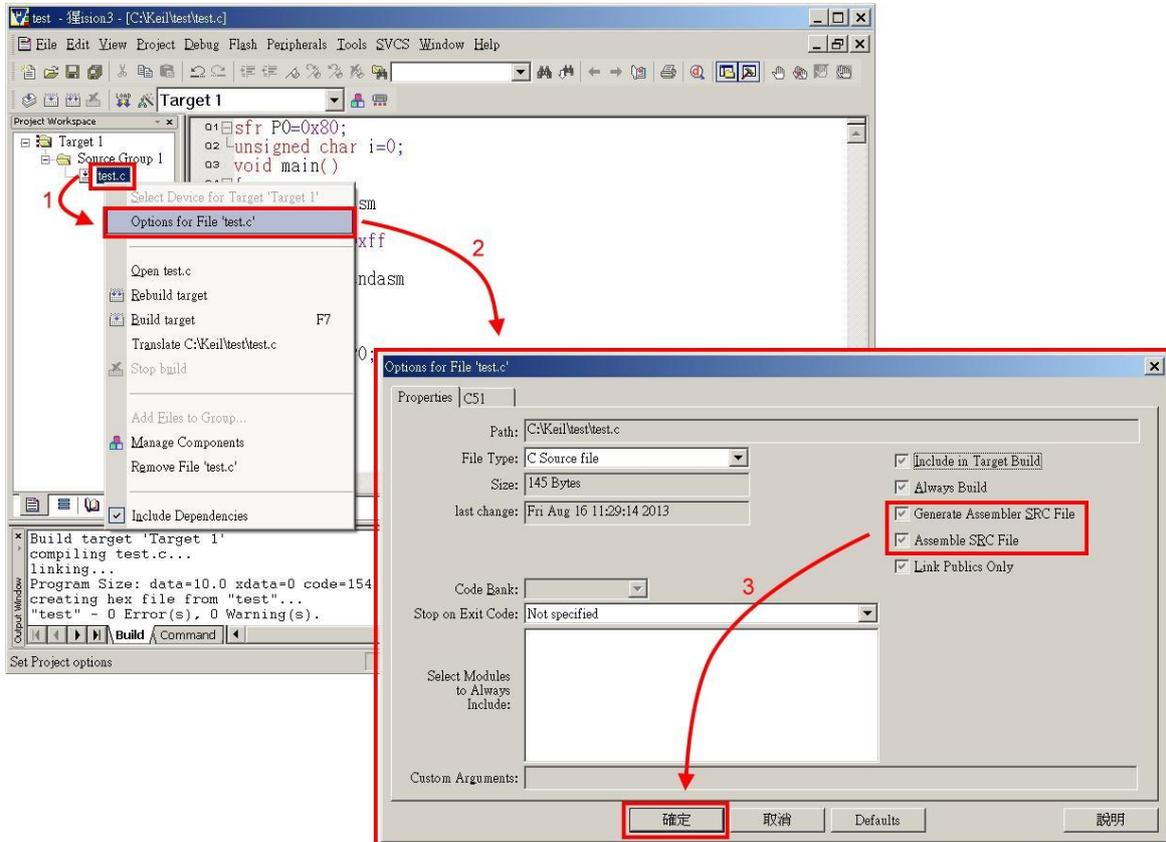
## 8. C 语言文件中嵌入汇编

8.1 Keil 库文件中包含了一些内联的函数，可以完成一些汇编语言的功能，因此一般情况不建议使用嵌入汇编的方式，如有特殊情况需要使用嵌入汇编，可按以下方法操作。

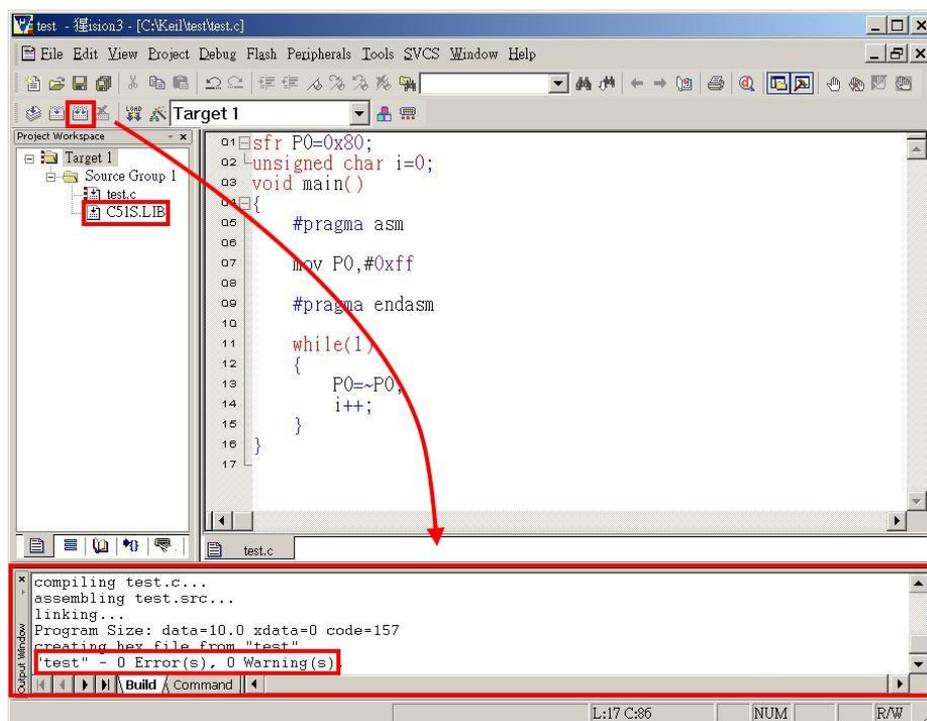
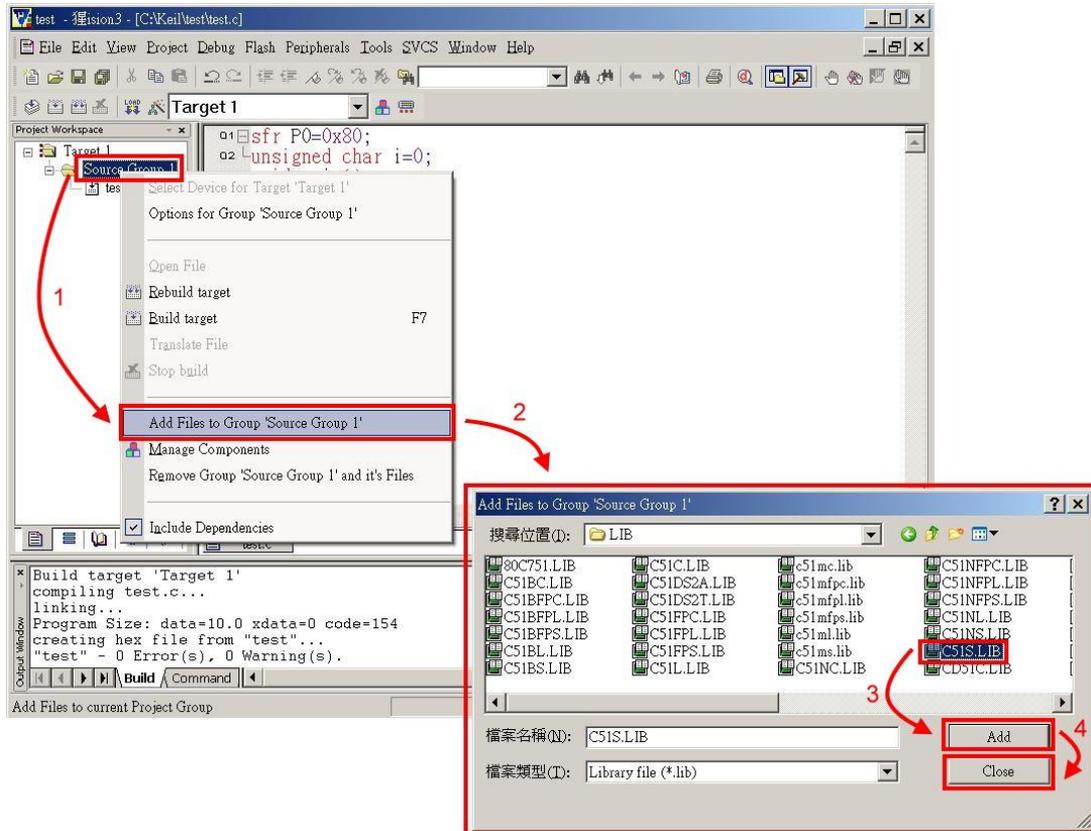
如下图，在 C 档中要嵌入汇编的地方用 `#pragma asm` 和 `#pragma endasm` 分隔开来，这样编译时 Keil 就知道这中间的一段是汇编了。



8.2 在有加入汇编的文件中，还要设置编译该文件时的选项:右键点击 test.c 弹出菜单选择 Options for File 'test.c' 弹出'test.c'文件配置接口，勾选 Generate Assembler SRC File 生成汇编 SRC 文件及 Assemble SRC File 封装汇编文件，选上这两项就可以在 c 中嵌入汇编了。



8.3 为了能对汇编进行封装还要在项目中加入相应的封装数据库档案，这些档案是在 Keil 安装目录下的 LIB 目录中。其中 C51s 是最常用的。加好后就可以顺利编译了。



## 常见问题 Q & A

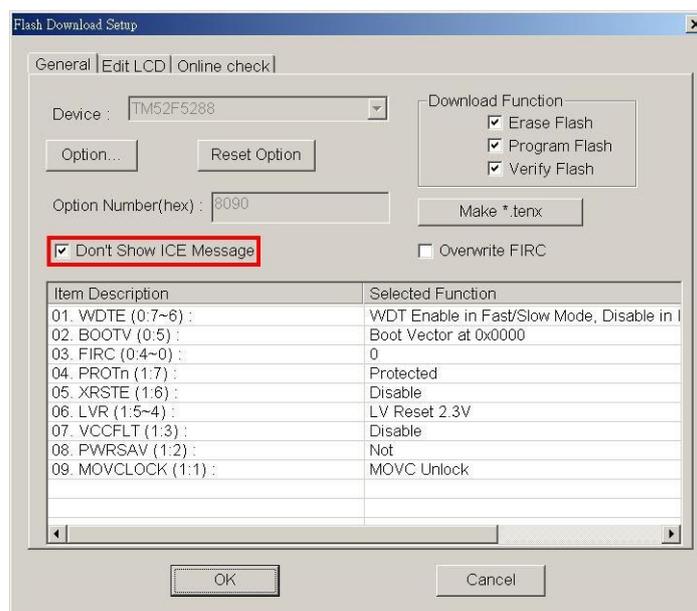
**Q1: 如何同时使用 Keil C 的 UV3 及 UV4 版本?**

**A1:** 若 UV3、UV4 安装在同一路径(UV3、UV4 都在 C:\Keil) , 需先将先安装的 UV3 或 UV4 的 C51 文件夹变更名称(因会有 2 个 C51 文件夹), 再安装 tenx F51&L51 插件, 例如现在若要使用 UV3 则需将 UV4 的 C51 文件夹档名做变更, 反之, 若要使用 UV4 则需将 UV3 的 C51 文件夹档名做变更(因 UV3、UV4 的数据默认都是指定到 C51); 假使安装在不同路径时(UV3 在 C:\ Keil、UV4 在 D:\ Keil), 安装 tenx F51&L51 插件时点选各自 Keil C 安装路径即可。

**Q2: 为何点击调试 (Debug) 按钮后会出现下图警告窗口?**



**A2:** 用户在撰写程序时必须避开控制到调试控制脚位 P12、P13(例如: P1 Mode 组态改变), 若下次调试时若不想出现此窗口, 请直接勾选 Do not show next time 或到 Flash Download Setup 窗口勾选 Do not show ICE Message。



**Q3:** 为何点击调试 (Debug) 按钮后会出现下图警告窗口后跳离调试模式?



**A3:** 因设置时芯片型号与对应的 EV Board 型号不符, 请检查设置型号。

**Q4:** 为何点击调试 (Debug) 或不调试 (Download)按钮后会出现下图警告窗口后跳离模式?

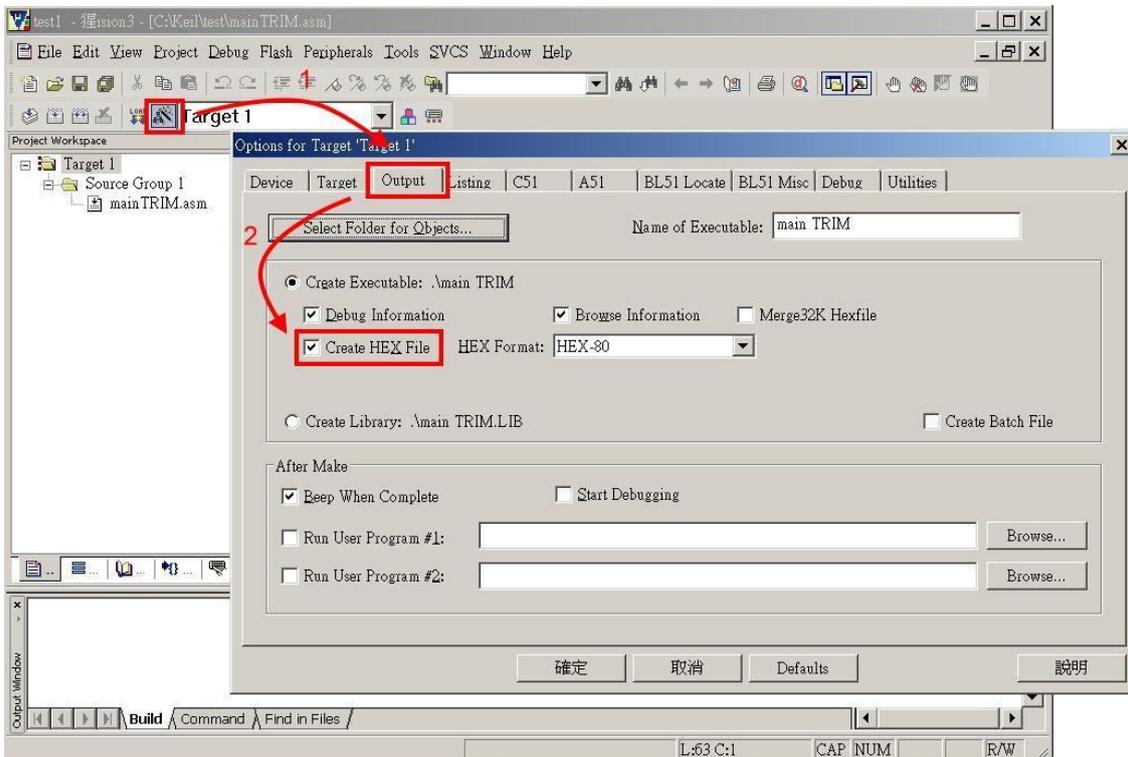


**A4:** 因撰写程序时超出 ROM code 的可用范围, 请参考第 11 页附注(4.5)。

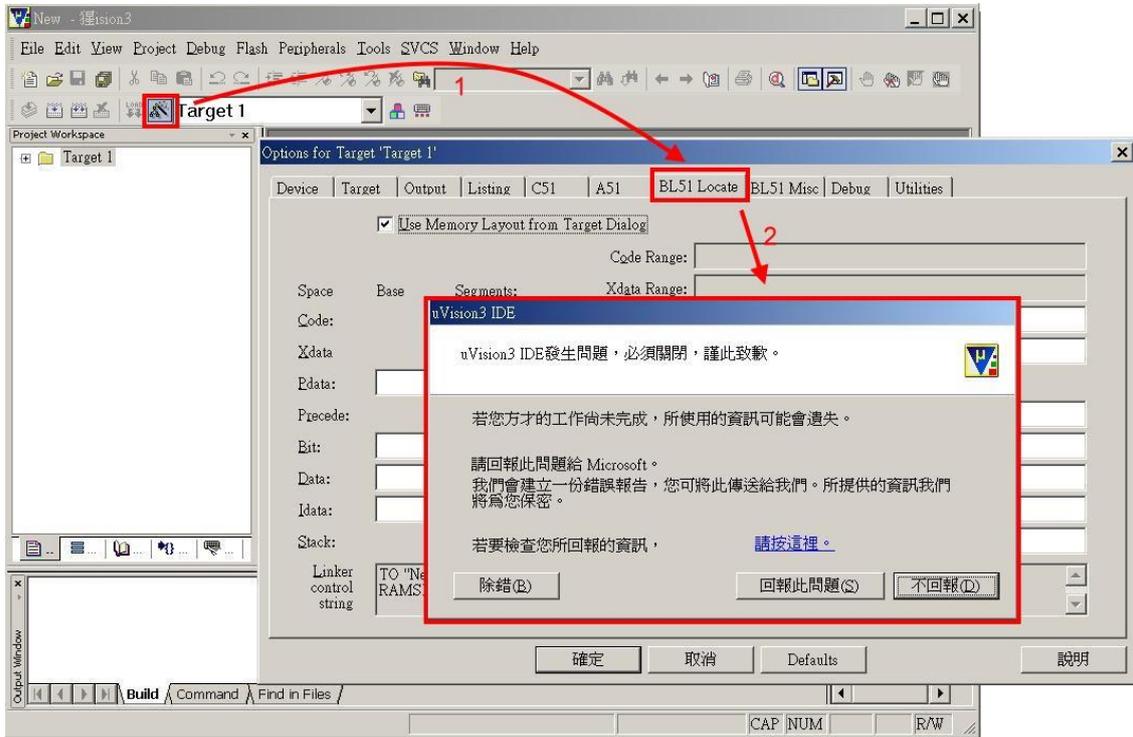
**Q5:** 为何点击调试 (Debug) 按钮后会出现下图警告窗口后跳离调试模式?



**A5:** 因上未设定勾选产生 HEX File, 请参考下图设定。



**Q6:** 为何使用 Keil C 的 UV3 版本，开启新项目后有时在设定选项直接点选 BL51 Locate 会发生 Keil C 当机问题？



**A6:** 此问题为 UV3 版本的 Bug（UV4 版本不会发生），请先点选其他设定再点选 BL51 Locate 即可。

**Q7:** 为何点击不需调试模式  (Download to Flash Memory) 时没有保护 ROM code 的功能？

**A7:** 若需要有保护 ROM code 的功能，需设定 Flash download setup→Option→Protected 后点击产生 .tenx 檔，再使用 Writer 刻录才会有保护 ROM code 功能。