

Megawin 8051 OCD ICE

使用說明書(中文版)

內容

1	介紹	3
	特色	3
	描述	3
2	硬體設定	4
3	軟體設定	5
3.1	安裝ICE轉接器的驅動程式	5
3.2	新增Megawin晶片的資料到Keil 8051 IDE	5
4	Keil IDE設定	6
4.1	Device選項	7
4.2	Target選項	7
4.3	Output選項	8
4.4	C51 選項	8
4.5	Debug選項	9
4.6	Utilities 選項	10
5	開始除錯	11
5.1	啟動dScope-Debugger功能	11
5.2	介紹除錯環境	12
5.2.1	重置(Reset)/執行(Run)/停止(Halt)/單步(Step)/執行到某行(Run-to-Cursor)	13
5.2.2	原始碼等級(Source-Level)的除錯	13
5.2.3	設定中斷點	14
5.2.4	顯示/編輯週邊暫存器的內容	15
5.2.5	檢視反組譯視窗	16
5.2.6	檢視查看視窗	17
5.2.7	檢視記憶體視窗	18
6	注意事項	19
6.1	暫存器定義檔	19
6.2	內建XRAM及外部資料記憶體	19
6.3	程式碼最佳化及原始碼除錯	20
6.4	for迴圈的原始碼除錯	21
6.5	使用除錯時的硬體選項要求	21
6.6	錯誤訊息	22
6.7	正確的連接ICE轉接器到電腦	23
	修訂記錄	24

1 介紹

特色

- 笙泉專利的 OCD(On-Chip-Debug) 技術
- MCU 內建即時除錯
- 獨立的兩雙接腳串列介面，不佔用系統的接腳
- 直接相容於 Keil 的 8051 IDE 除錯模擬介面
- 使用 USB 連接電腦於系統
- 強大的除錯動作：重置、全速執行、暫停、單步執行...等等
- 可程式化的中斷，可以同時插入四個中斷
- 多個有用的除錯視窗：暫存器/反組譯/監看變數/記憶體 視窗

描述

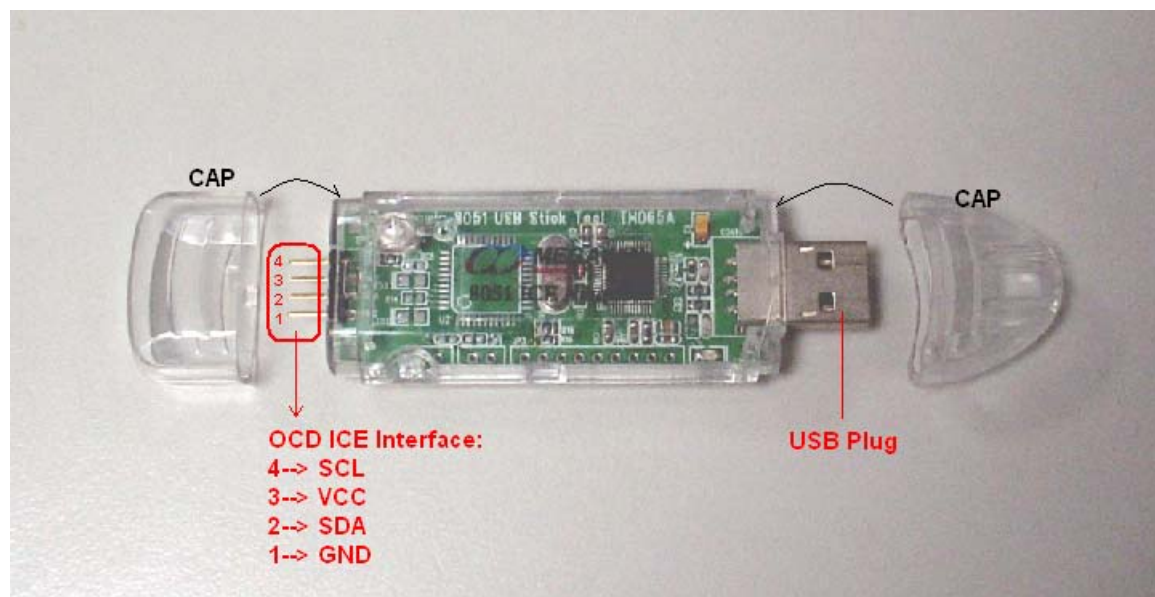
這個全新的“Megawin 8051 OCD ICE”對 8051 嵌入式系統來說是一個強而有力的開發工具。他是採用笙泉科技的 OCD(On-Chip-Debug)專利技術，這個 ICE 提供內建即時除錯的功能。在開發除錯時使用者不需要像傳統的 8051 ICE 一樣，準備任何的開發板或者是轉接腳座即可以開發除錯。使用者唯一需要做的就是預留一個 4 隻腳位的連接器給專屬的 OCD 介面即可：VCC,OCD_SDA,OCD_SCL 以及 GND。

另外，他最有用的特色是能夠直接連接使用者的系統到 Keil 8051 IDE software 的介面做除錯，而他是直接使用 Keil IDE 的 dScope-Debugger 功能來做除錯並且承襲了所有 Keil 的優點。

註：

“Keil”是“Keil Elektronik Gmbh and Keil Software, Inc.”的註冊商標，而“Keil 8051 IDE software”是8051 嵌入式系統的開發環境中最普通的一個。

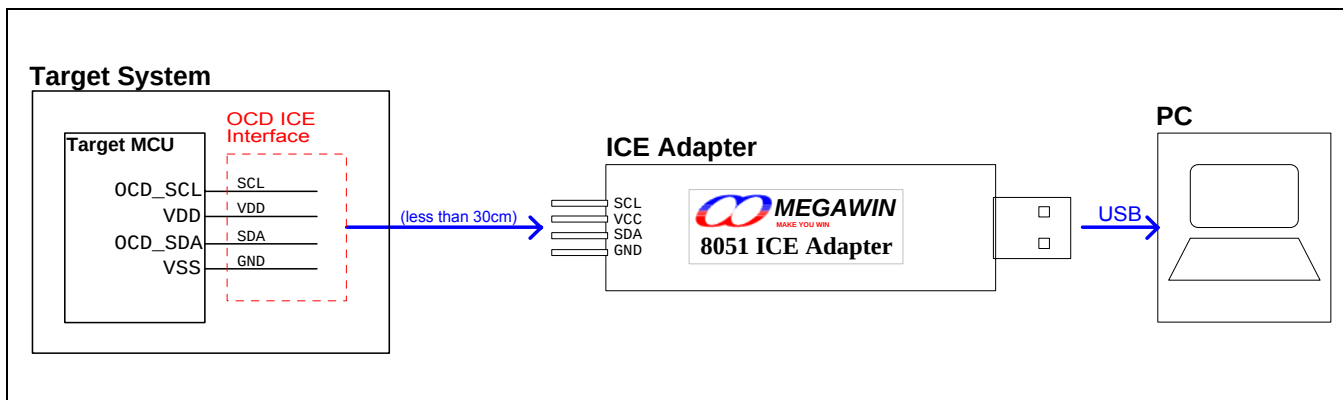
“Megawin 8051 OCD ICE”的ICE轉接器的照片



2 硬體設定

當要做除錯時，使用者必須使用 ICE 轉接器將電腦與系統連接起來，如下圖。ICE 轉接器是使用 USB 的電源，因此使用 ICE 時是不需要再接其他的電源的。

硬體連接圖



註：更多資訊請參考 [6.5 節](#)。

OCD ICE介面的腳位編號

Part No.	Package	OCD_SCL	OCD_SDA
MPC82G516	40-pin DIP	29	30
	44-pin PLCC	32	33
	44-pin QFP	26	27

3 軟體設定

這個章節會告訴你在使用 OCD ICE 之前要如何做軟體的設定。

3.1 安裝ICE轉接器的驅動程式

使用者只需要把這個 ICE 轉接器直接插上任何一個 USB 埠就可以了，不需要安裝任何驅動程式。

3.2 新增Megawin晶片的資料到Keil 8051 IDE

首先，將 ICE 轉接器插到電腦的 USB 埠，然後執行目錄[Database Installer]裡的"Setup.exe"將 Megawin 的晶片資訊新增至 Keil 8051 IDE 內。當然，您可以新增到 Keil 8051 IDE 的 μ Vision2 或是 μ Vision3 都可以。

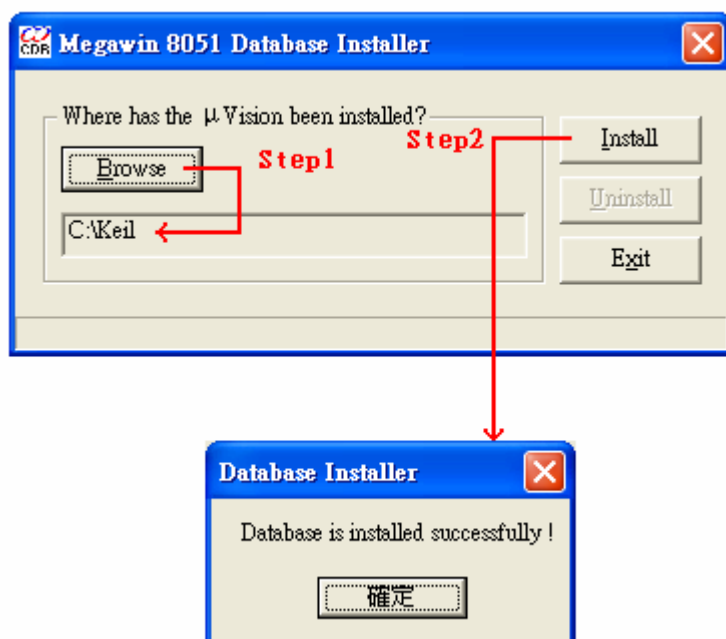
在打開 Database Installer 後，請依照下列順序完成新增動作，如圖示。

步驟一：按下 **Browse** 鈕指定 Keil 的安裝目錄。

(一般來說，安裝 Keil 8051 IDE 時預設的安裝路徑為"C:\KEIL".)

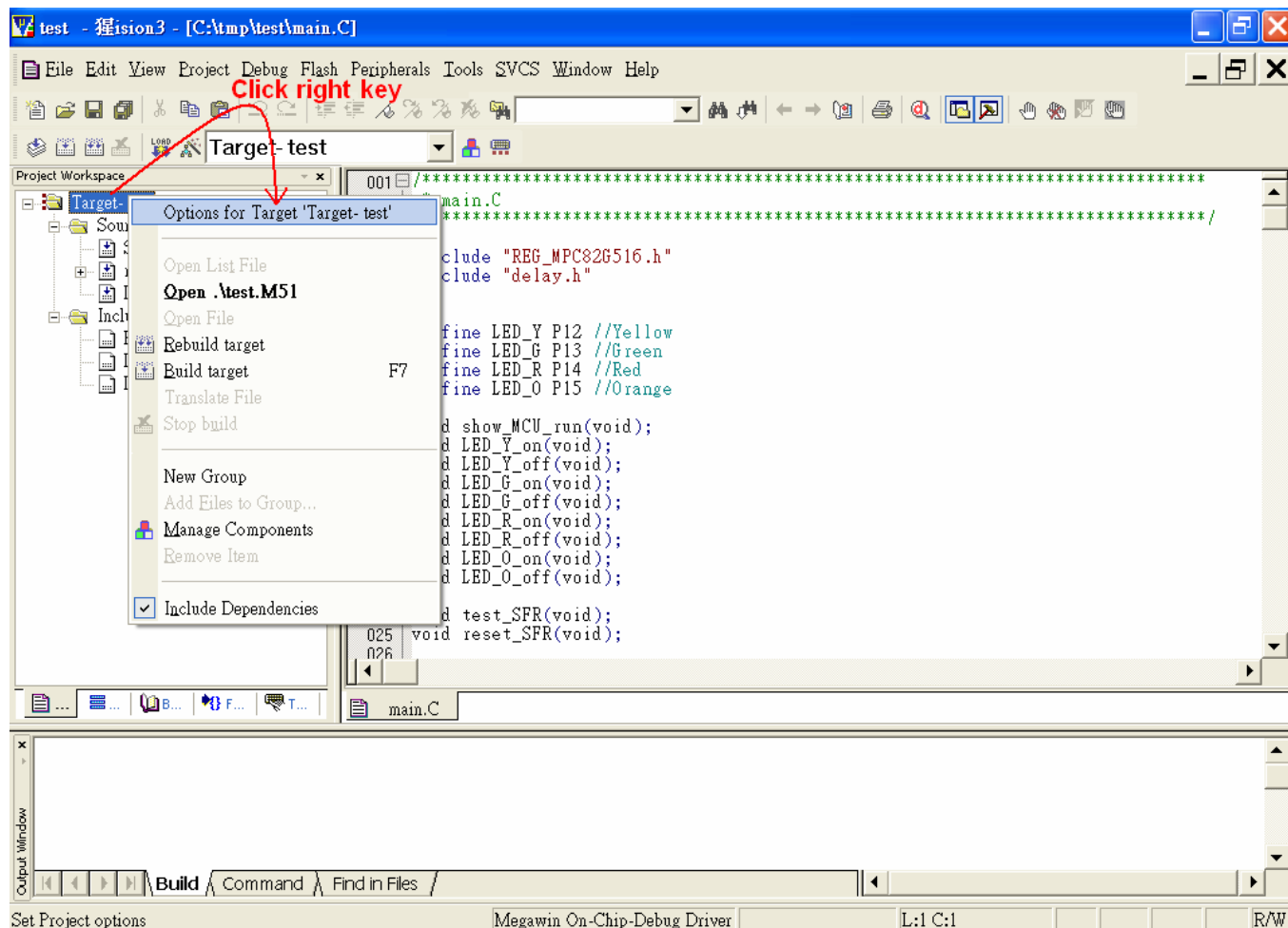
步驟二：按下 **Install** 鈕開始新增 Megawin 晶片的資料到 Keil 內。

安裝過程圖解



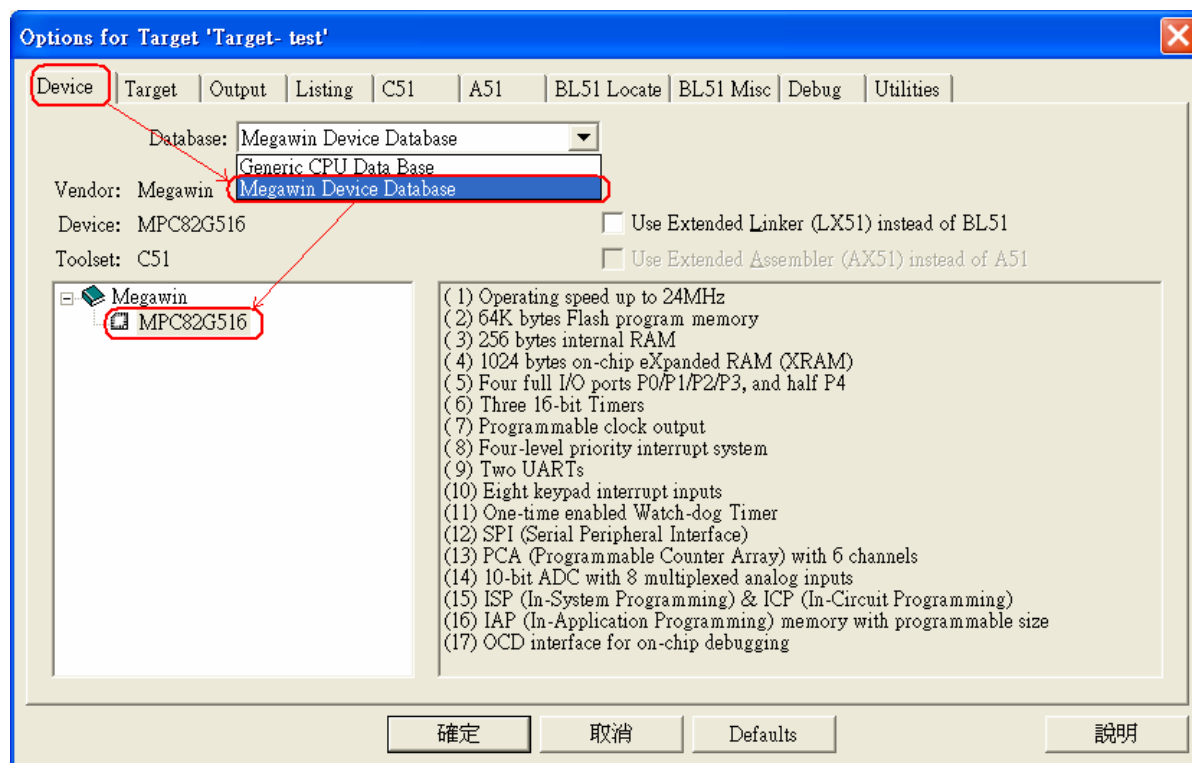
4 Keil IDE設定

在使用 Keil IDE 的 dScope-Debugger 功能之前使用者必須先對 Keil IDE 做一些設定。首先，先開啓您要除錯的 μ Vision 專案，然後在"Target-.."的地方按下右鍵並點選**"Options for Target"**如下圖所示：



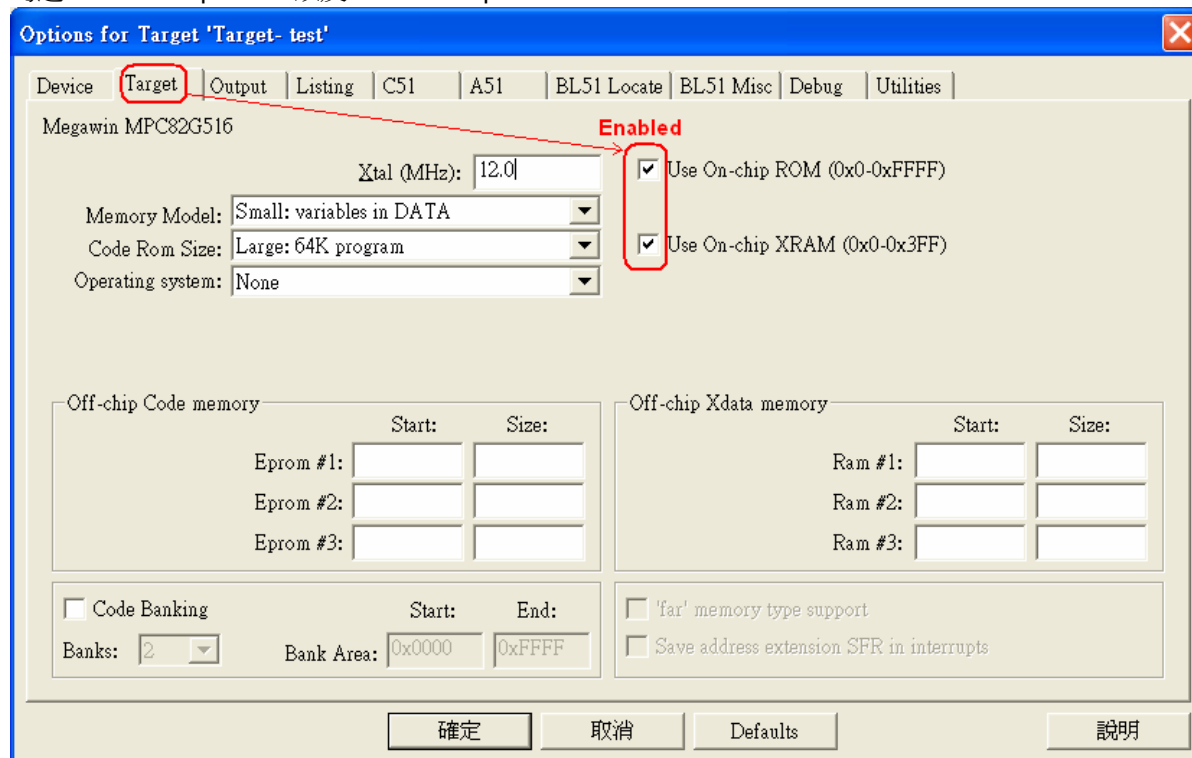
4.1 Device 選項

選擇“Megawin Device Database”以及型號。



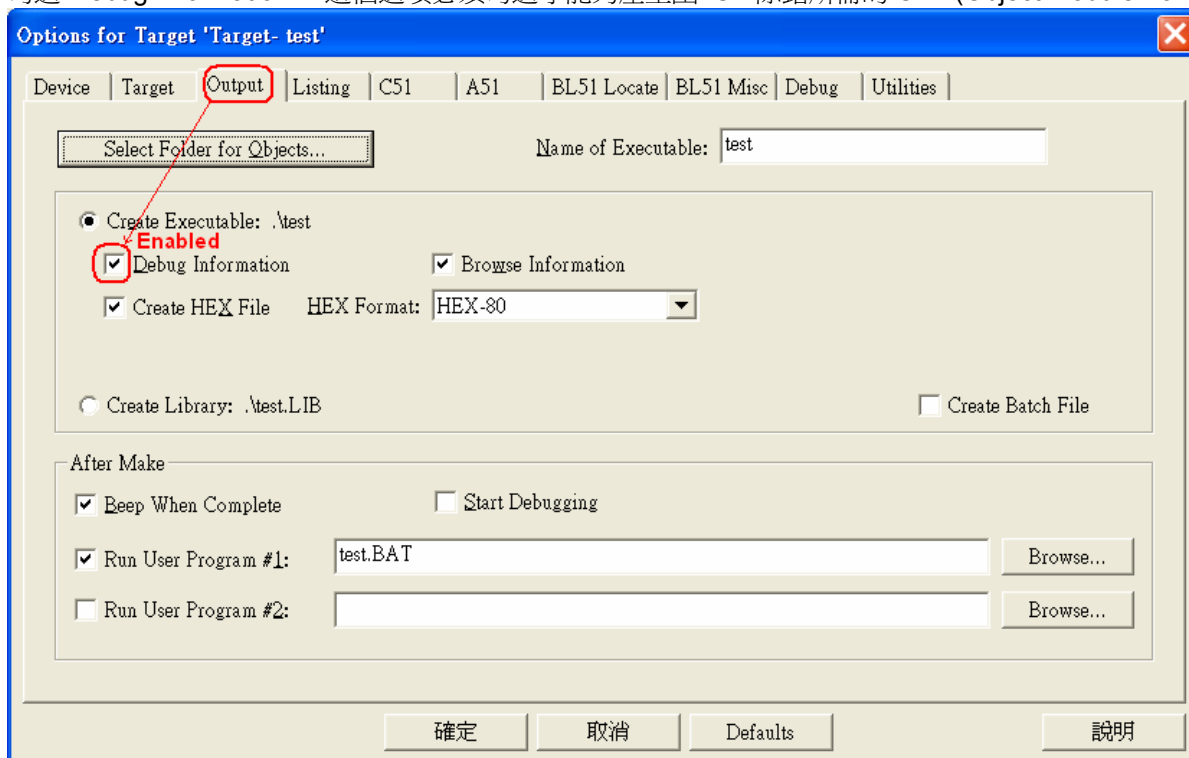
4.2 Target 選項

勾選“Use on-chip ROM”以及“Use on-chip XRAM”。



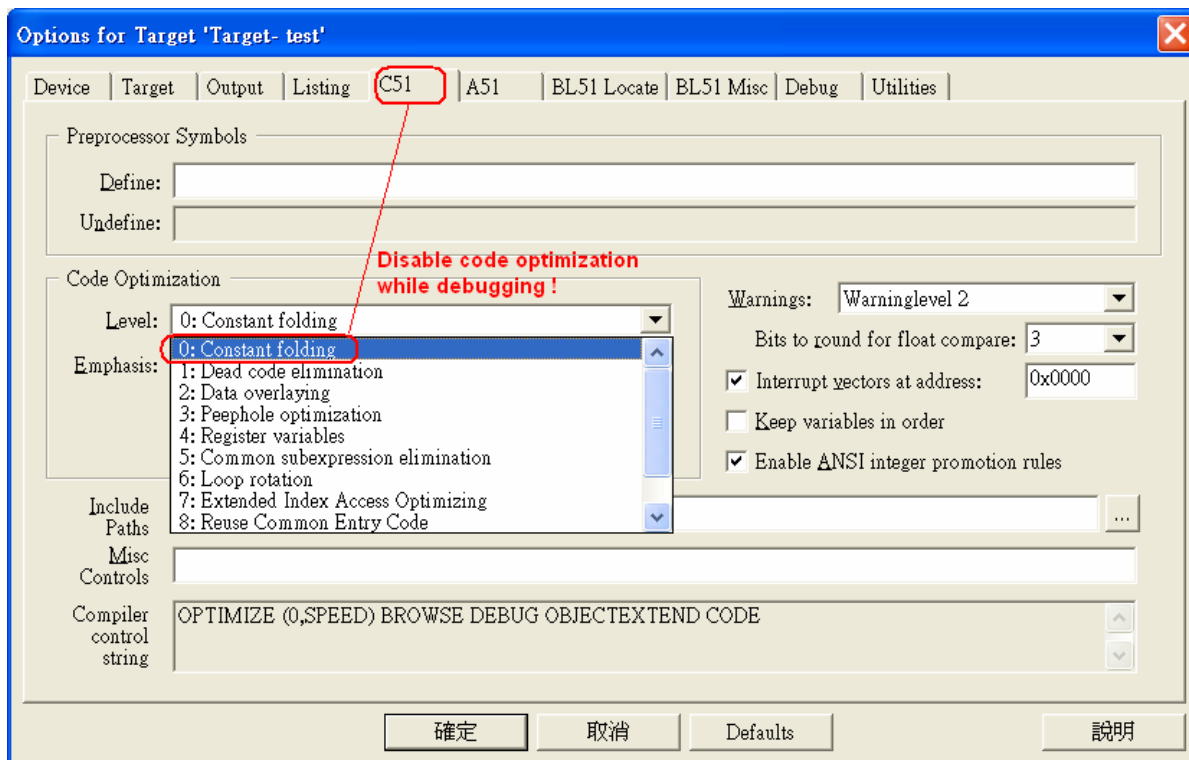
4.3 Output 選項

勾選“Debug Information”。這個選項必須勾選才能夠產生出 ICE 除錯所需的 OMF(Object Module Format)檔案。



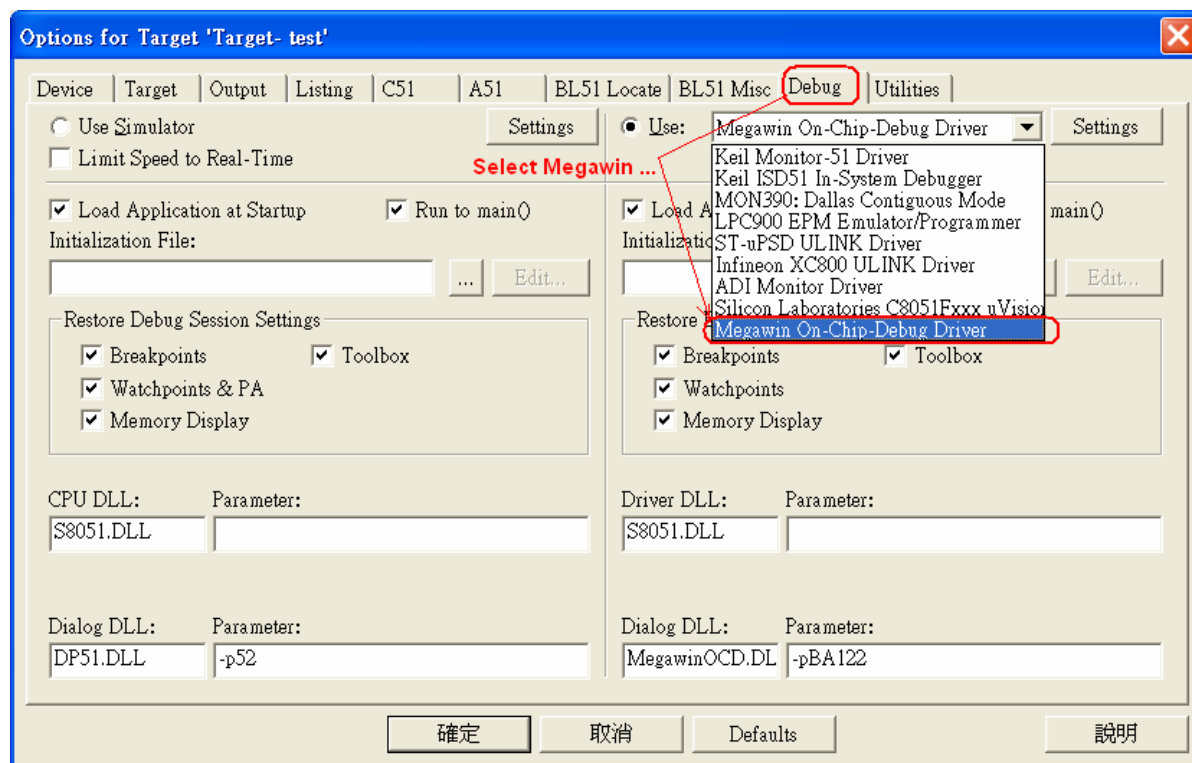
4.4 C51 選項

選擇“Level 0: Constant folding”用以關閉程式碼的最佳化。詳細資料請參考[6.3 節](#)。註：這個設定是非必要的。

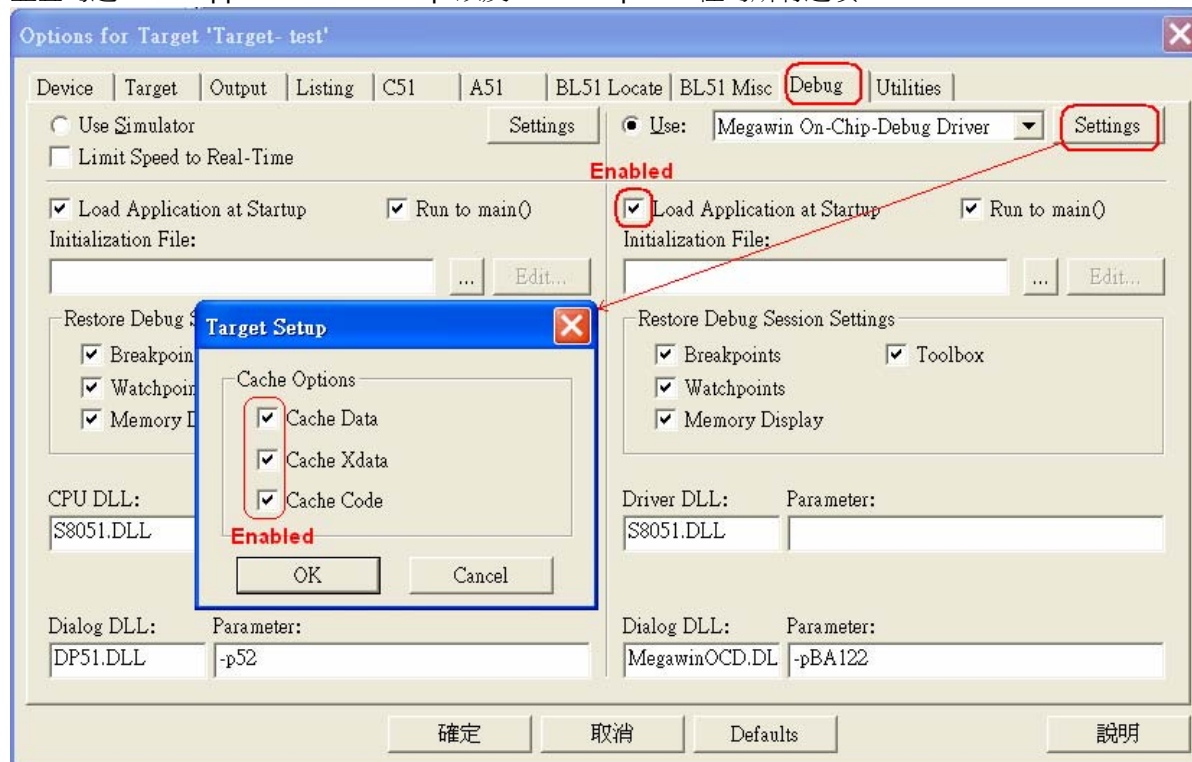


4.5 Debug 選項

選擇“Megawin On-Chip-Debug Driver”。



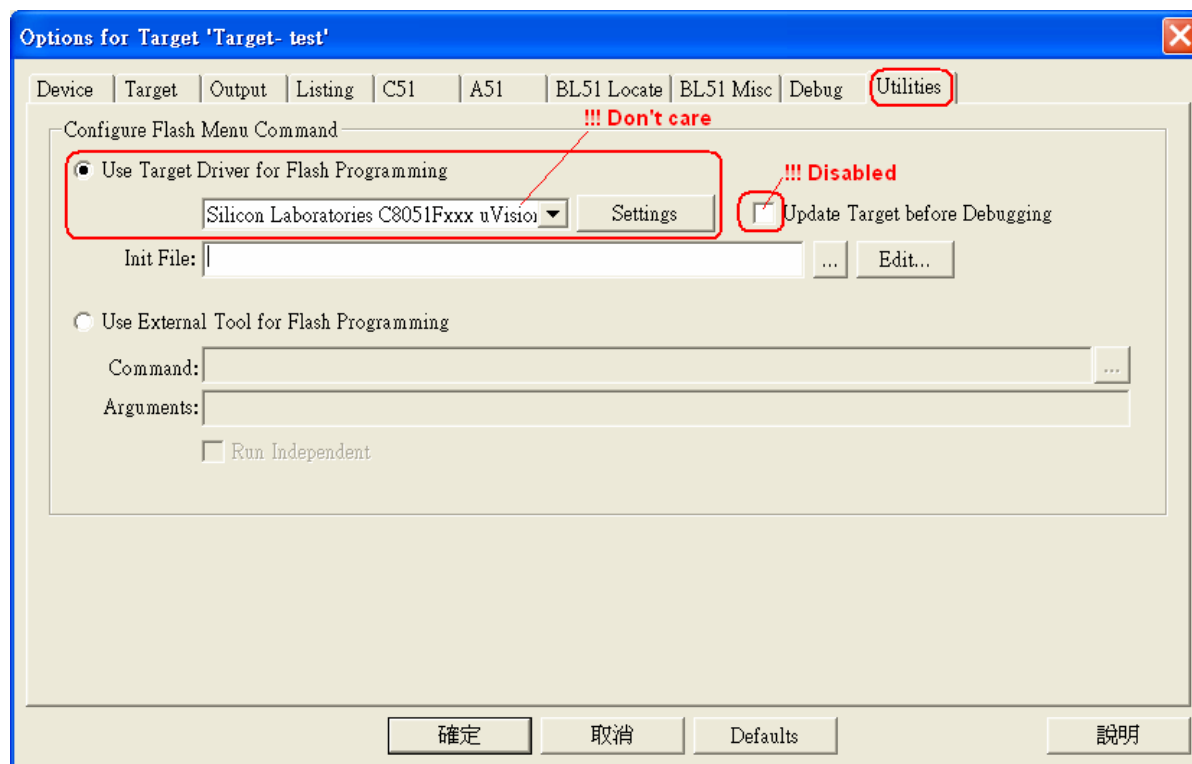
並且勾選“Load Application at Startup”以及 Cache Options 裡的所有選項。



4.6 Utilities 選項

“Update Target before Debugging”一定要關閉，因為我們已經勾選了“Load Application at Startup”參考 [4.5 節](#)。而“Use Target Driver for Flash Programming”可以乎略不用管他。

註：μVision2 沒有這個選項。

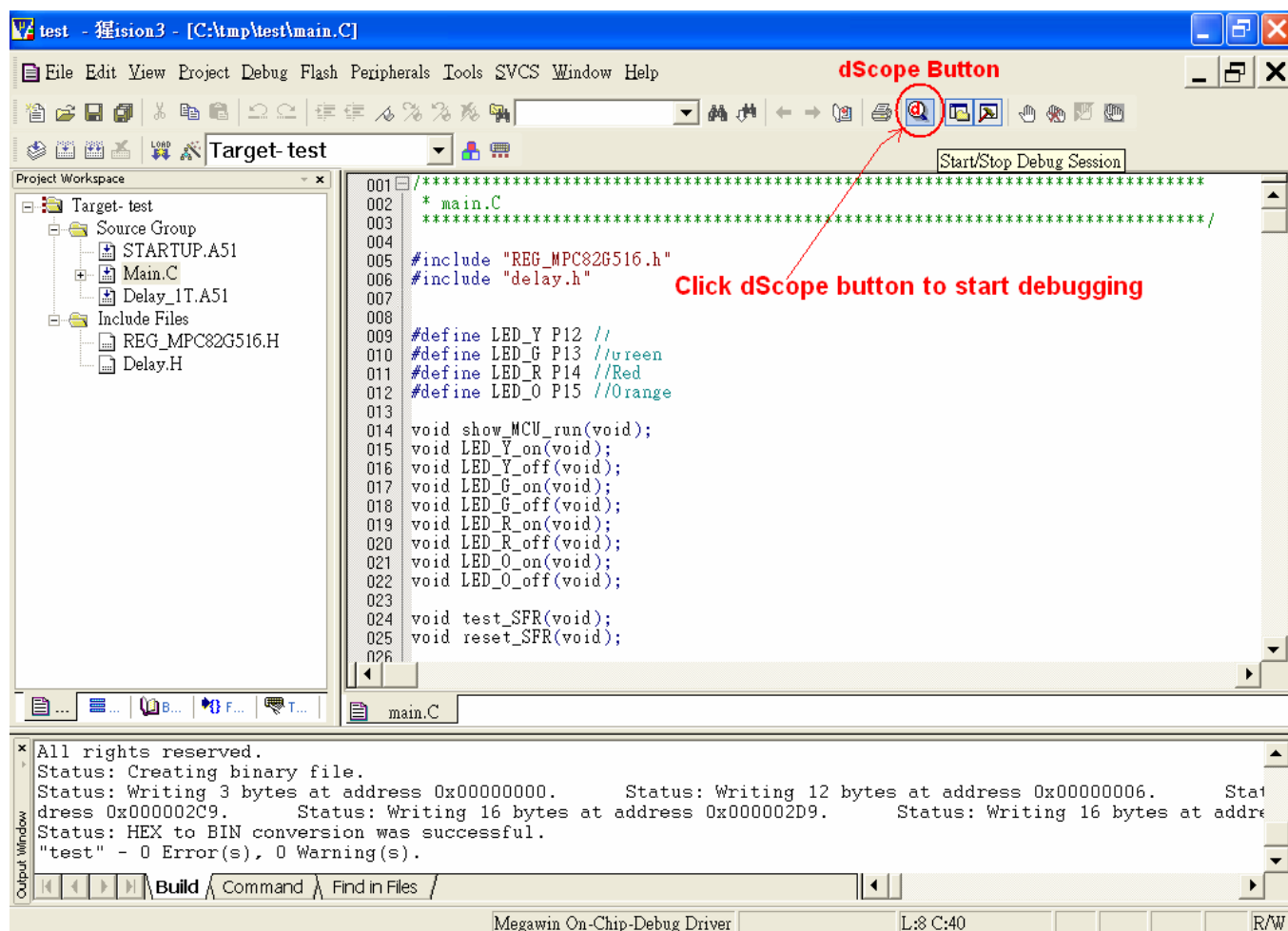


5 開始除錯

當您完成第二、三、四節的設定後，您就可以開始使用 μ Vision 來做除錯的功能了。

5.1 啟動 dScope-Debugger 功能

在做完專案的設定後(假設沒有錯誤的話)，您就可以按下 **dScope** 鈕進入 Keil IDE 的除錯模式了，按下後會自動將您的程式下載到 MCU 內部，而這個過程會花一點時間。



5.2 介紹除錯環境

在除錯的環境裡可以看到有四個基本的視窗，他們分別是暫存器(Register)視窗、反組譯(Disassembly)視窗、監看變數(Watch)視窗及記憶體(Memory)視窗，詳細說明如下：

暫存器視窗

這個視窗會顯示出目前的暫存器值(R0~R7)，還有系統暫存器(A,B,SP,DTPR 及 PC)還有程式狀態字元(PSW)。當暫存器顯示為藍底時代表他正被目前的指令改變他的數值。

反組譯視窗

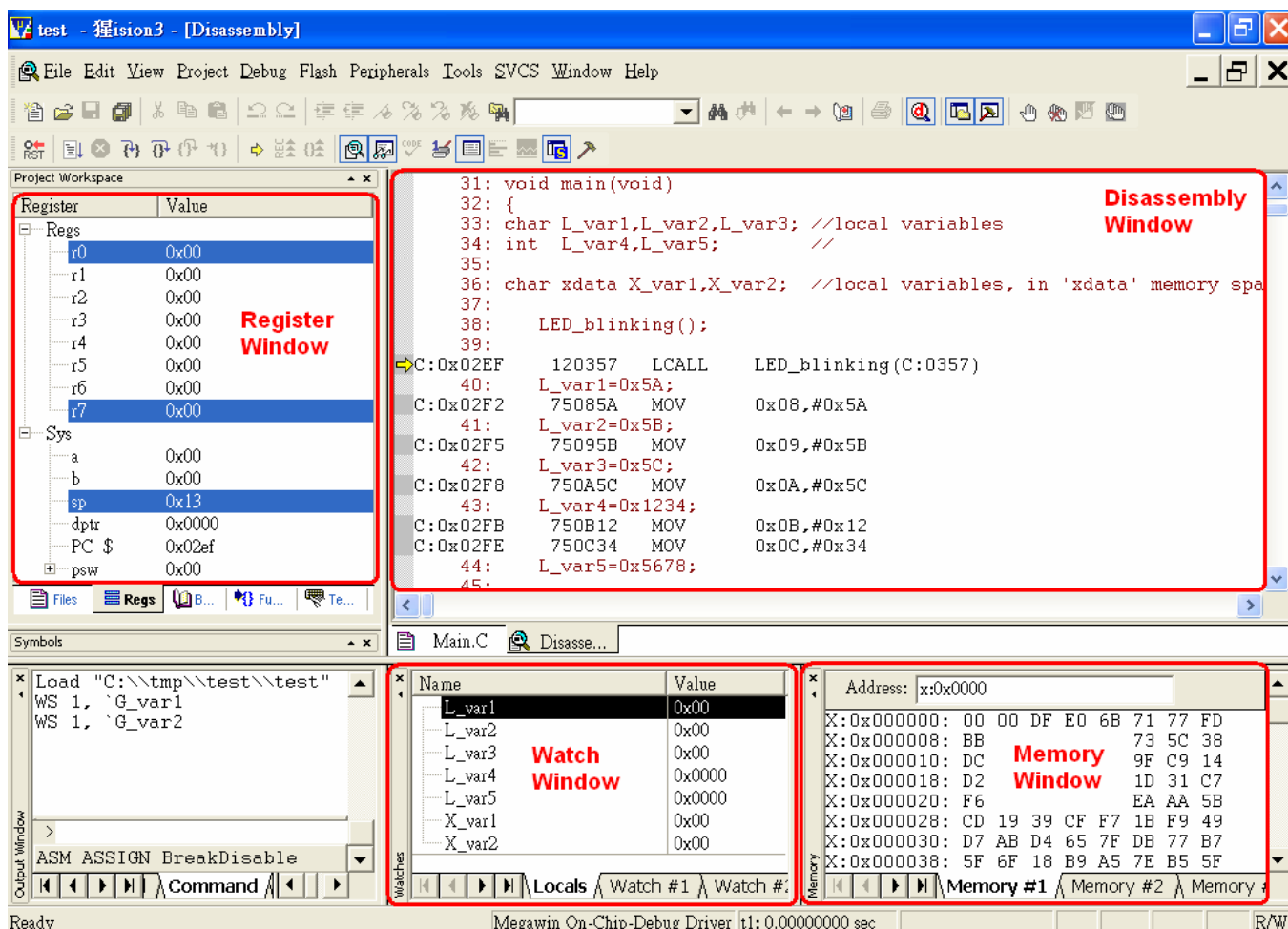
這個視窗在進入除錯模式後就會自動打開，他會將目前的程式碼以相對應的組合語言顯示出來。

查看視窗

當目前選在 **Locals** 分頁時，這個視窗會自動將區域變數顯示出來。這個區域變數的值包括在主迴圈 **main()** 的變數。如果要查看全域的變數則必須先將分頁選到 **Watch #1** 或 **Watch #2**，然後按下<F2>並輸入您要查看的變數名稱即可，同樣的，當變數為藍底時代表他被目前的指令改變數值。

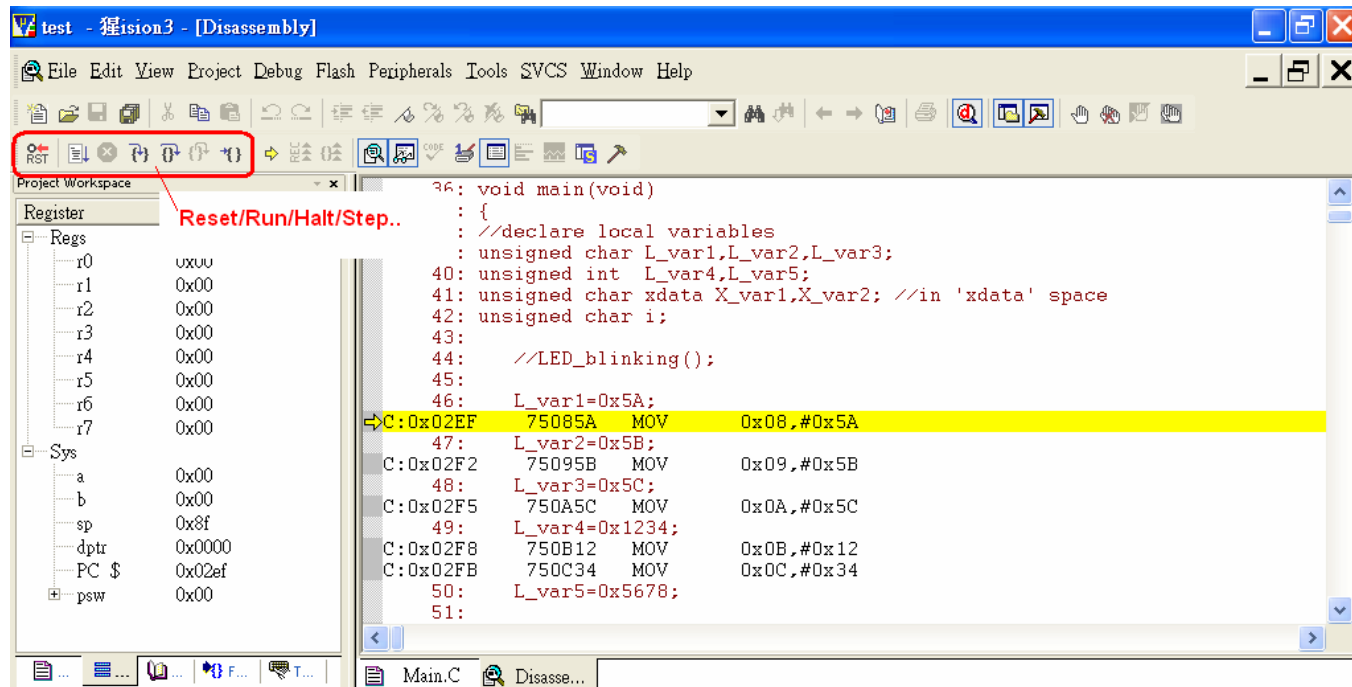
記憶體視窗

這個視窗可以顯示 **data/idata/xdata/code** 記憶體空間的內容，可以使用的命令為 **d:0x00~d:0xFF**, **i:0x00~i:0xFF**, **x:0x0000~x:0xFFFF** 以及 **c:0x0000~c:0xFFFF**，使用者可以用相對應該的命令查看這四種記憶體的內容。



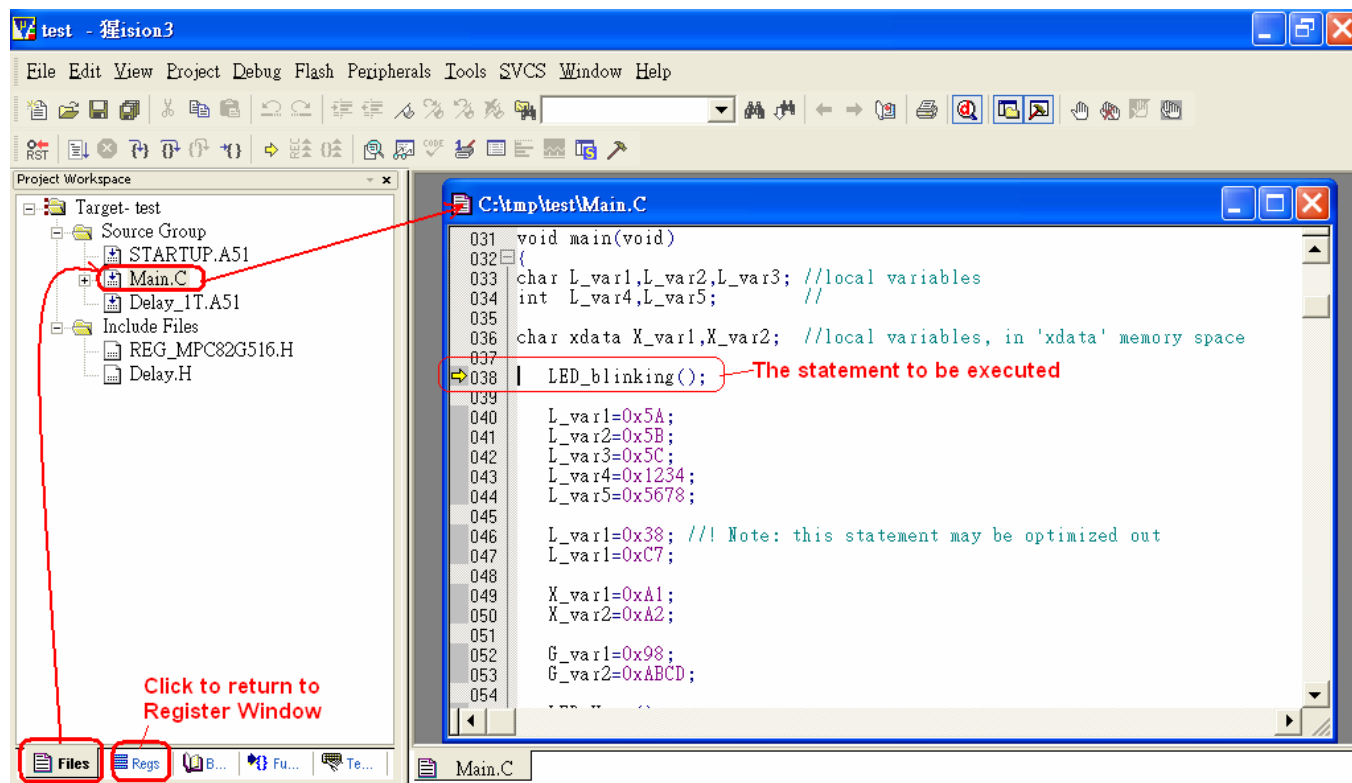
5.2.1 重置(Reset)/執行(Run)/停止(Halt)/單步(Step)/執行到某行(Run-to-Cursor)

重置、執行、停止、單步及執行到某行是基本的除錯動作，使用者可以輕易的在 GUI 的快捷列中執行這些功能，如下圖：



5.2.2 原始碼等級(Source-Level)的除錯

要做原始碼等級的除錯時，可以在 **Files** 分頁中打開預除錯的程式，再切回 **Regs** 分頁即可返回暫存器視窗，如下圖所示：

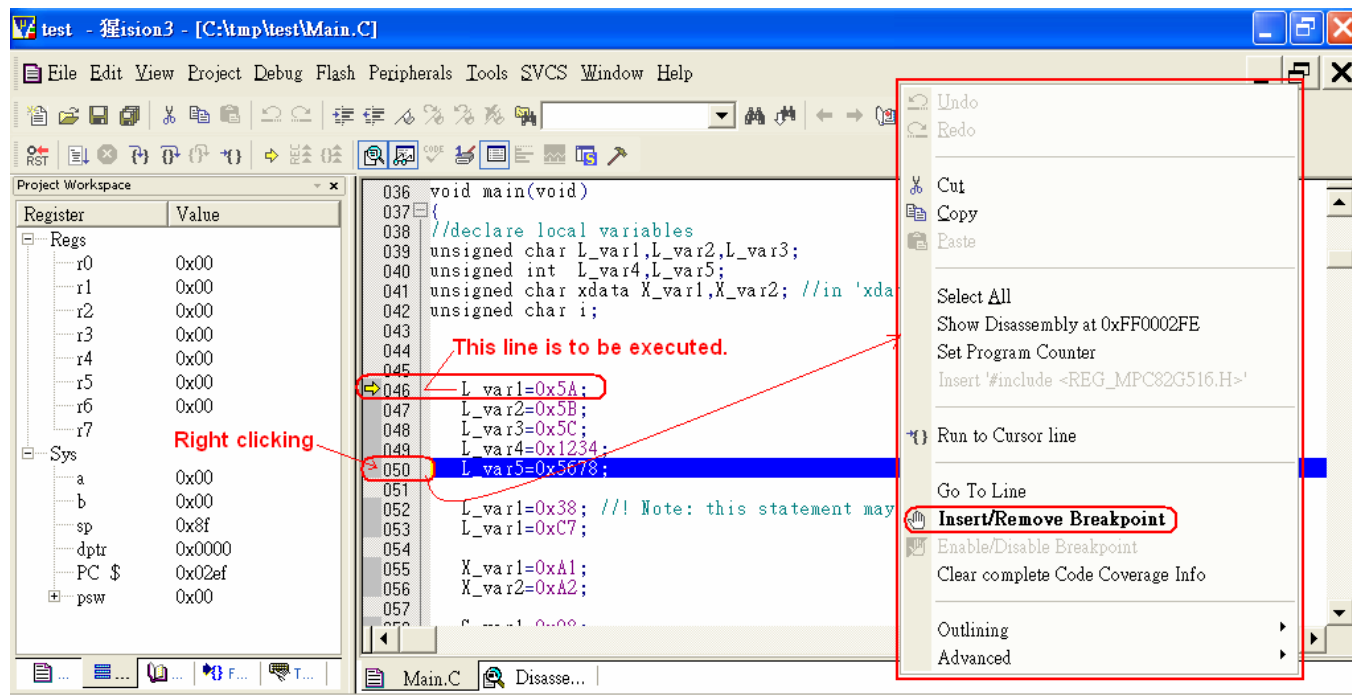


5.2.3 設定中斷點

除錯時最多可以同時設四個中斷來使用。

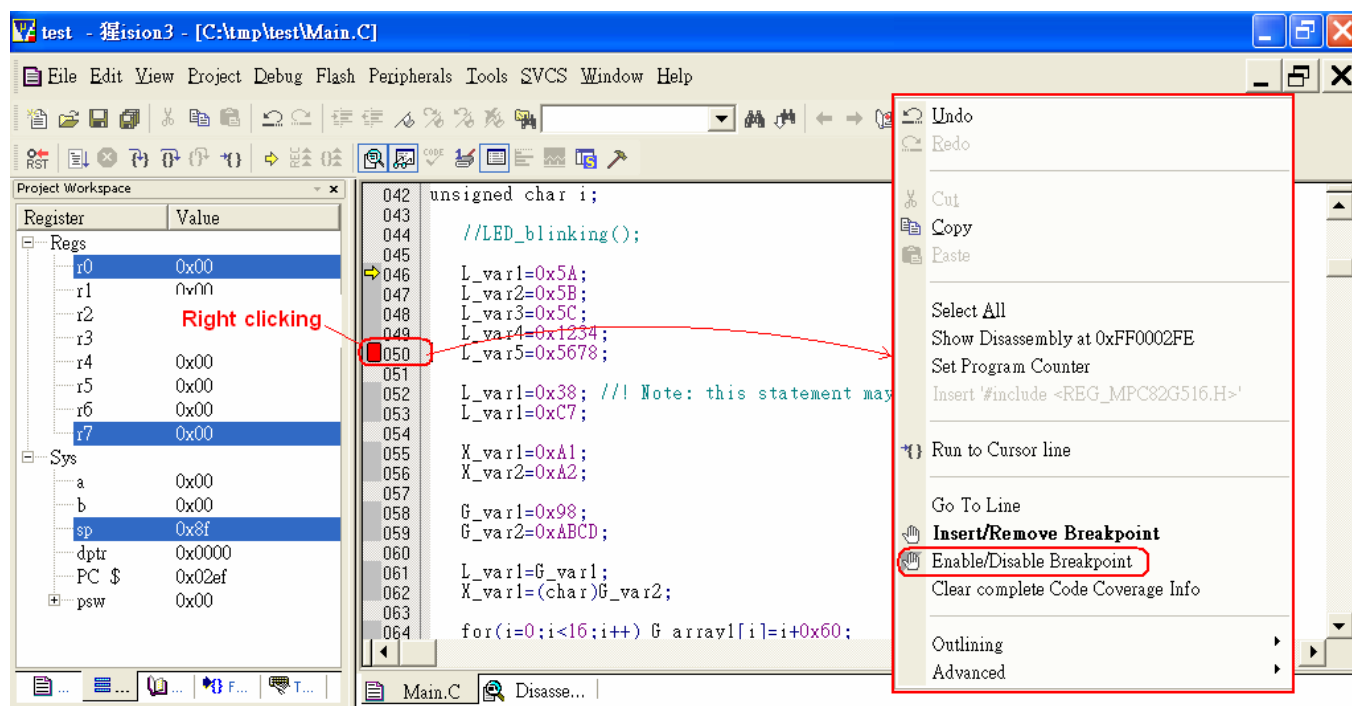
插入/移除中斷

將游標移至想要中斷的指令上並按下右鍵，然後選“Insert/Remove Breakpoint”可以在該行指令做插入或是移除中斷，如下所示：



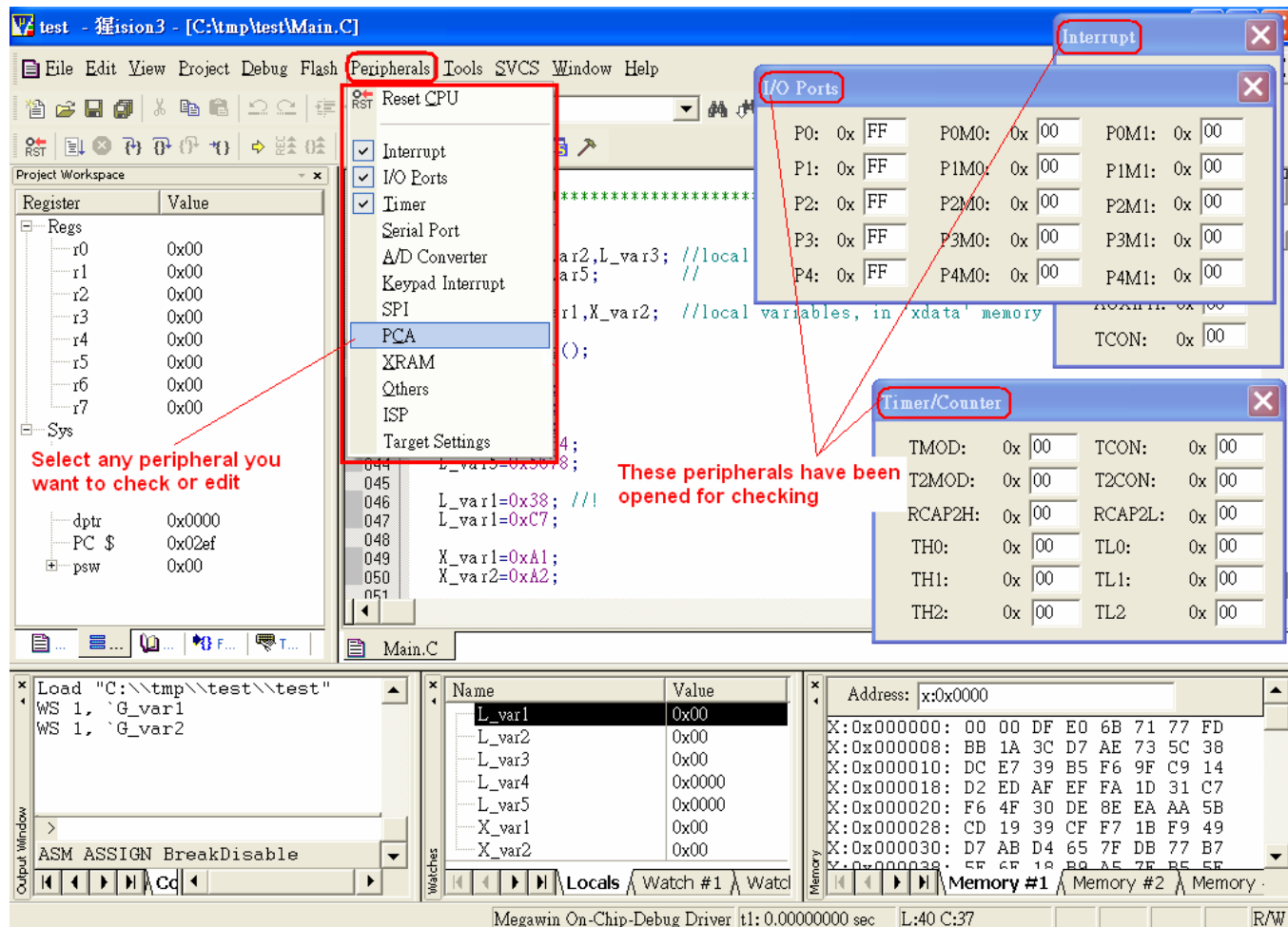
啟動/關閉中斷點

將游標移至想要設定的指令上並按下右鍵，然後選“Enable/Disable Breakpoint”可以在該中斷上設定是否啟動或是關閉該中斷功能，當然，該行指令必須先插入中斷。



5.2.4 顯示/編輯週邊暫存器的內容

有許多的週邊暫存器是不會顯示在暫存器視窗的，要查看或是編輯這些暫存器必須在主選單中選擇 **Peripherals**。之後會有顯示一個下拉視窗，使用者可以自行勾選預查看的暫存器，如下所示：



The screenshot shows the Megawin 8051 OCD ICE software interface. The **Peripherals** menu is open, listing various hardware components. The **I/O Ports** and **Timer/Counter** windows are also visible, displaying the status of these peripherals. Red annotations highlight the menu and the peripheral windows.

Peripherals Menu:

- Reset CPU
- ☒ Interrupt
- ☒ I/O Ports
- ☒ Timer
- Serial Port
- A/D Converter
- Keypad Interrupt
- SPI
- PCA
- XRAM
- Others
- ISP
- Target Settings

I/O Ports Window:

P0: 0x FF	P0M0: 0x 00	P0M1: 0x 00
P1: 0x FF	P1M0: 0x 00	P1M1: 0x 00
P2: 0x FF	P2M0: 0x 00	P2M1: 0x 00
P3: 0x FF	P3M0: 0x 00	P3M1: 0x 00
P4: 0x FF	P4M0: 0x 00	P4M1: 0x 00

Timer/Counter Window:

TMOD: 0x 00	TCON: 0x 00
T2MOD: 0x 00	T2CON: 0x 00
RCAP2H: 0x 00	RCAP2L: 0x 00
TH0: 0x 00	TL0: 0x 00
TH1: 0x 00	TL1: 0x 00
TH2: 0x 00	TL2: 0x 00

Register Window:

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00

Output Window:

```

Load "C:\\tmp\\test\\test"
WS 1, 'G_var1'
WS 1, 'G_var2'
  
```

Locals Window:

Name	Value
L_var1	0x00
L_var2	0x00
L_var3	0x00
L_var4	0x0000
L_var5	0x0000
X_var1	0x00
X_var2	0x00

Memory Window:

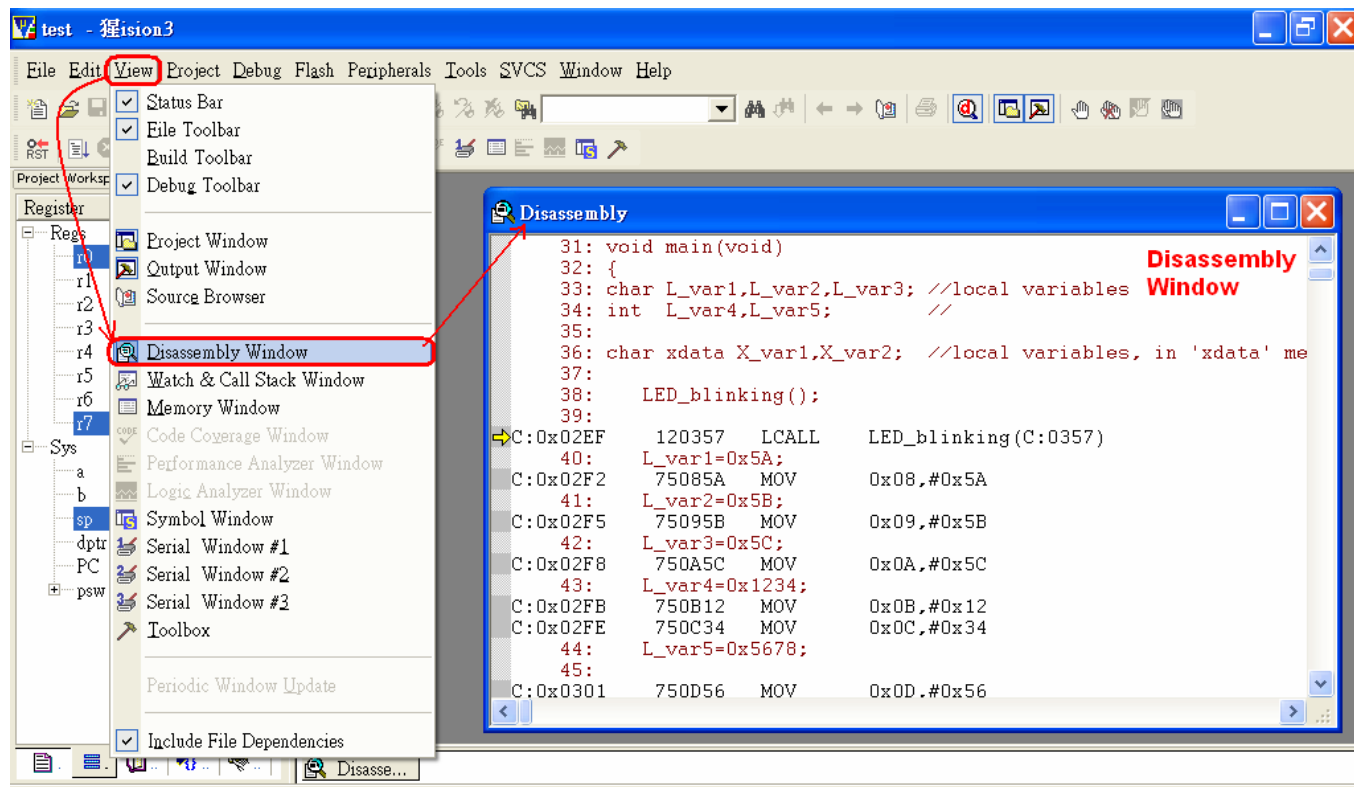
Address	Value
0x00000000	00 00 DF E0 6B 71 77 FD
0x00000008	BB 1A 3C D7 AE 73 5C 38
0x00000010	DC E7 39 B5 F6 9F C9 14
0x00000018	D2 ED AF EF FA 1D 31 C7
0x00000020	F6 4F 30 DE 8E EA AA 5B
0x00000028	CD 19 39 CF F7 1B F9 49
0x00000030	D7 AB D4 65 7F DB 77 B7
0x00000038	5F 6F 18 B9 A5 7E B5 5F

Annotations:

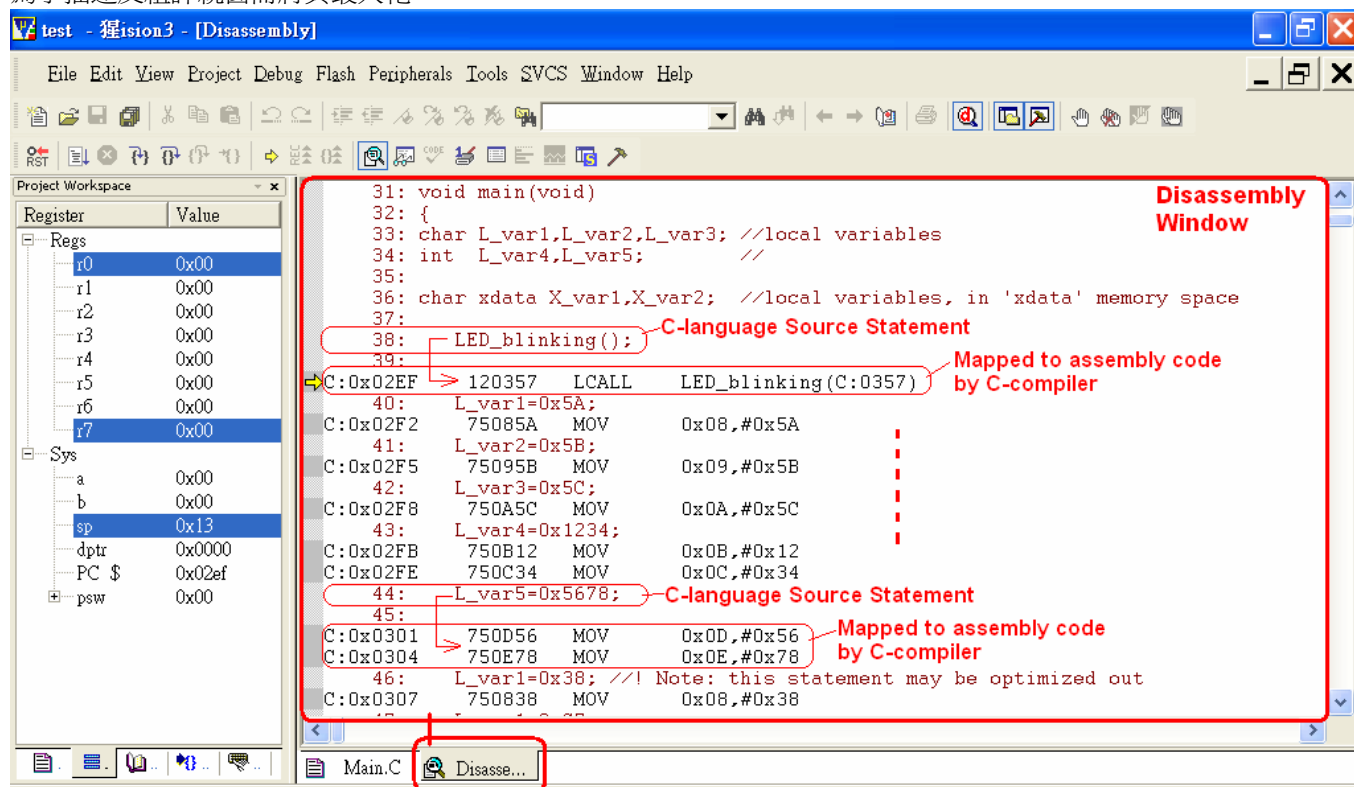
- Select any peripheral you want to check or edit** (points to the Peripherals menu)
- These peripherals have been opened for checking** (points to the I/O Ports and Timer/Counter windows)

5.2.5 檢視反組譯視窗

反組譯視窗顯示了原始碼相對應該組合語言，要開啓這個視窗可以先在主選單中點選 **View**，之後會有顯示一個下拉視窗，再點選 **Disassembly Window** 即可檢視反組譯視窗，如下圖所示：

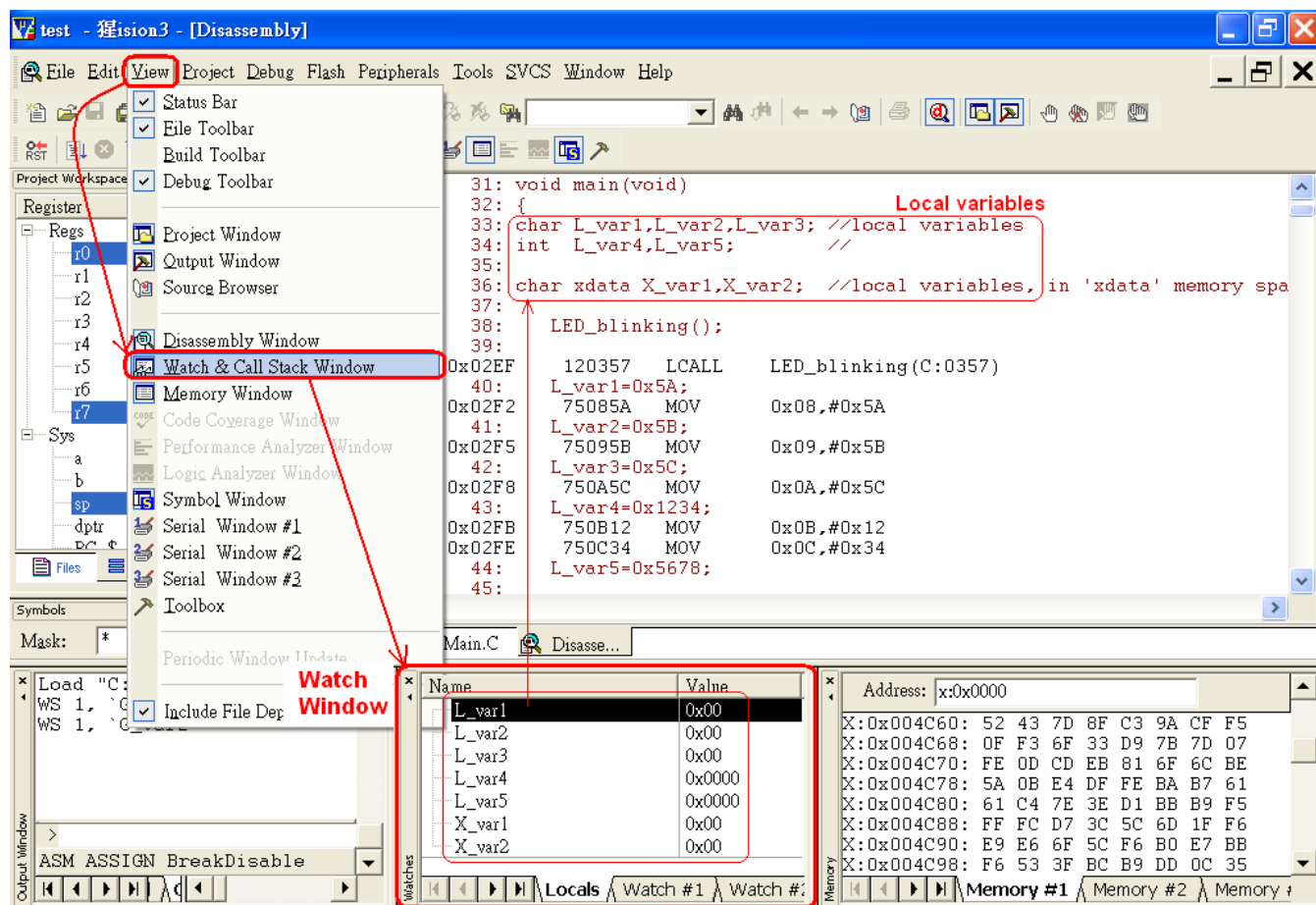


爲了描述反組譯視窗而將其最大化：

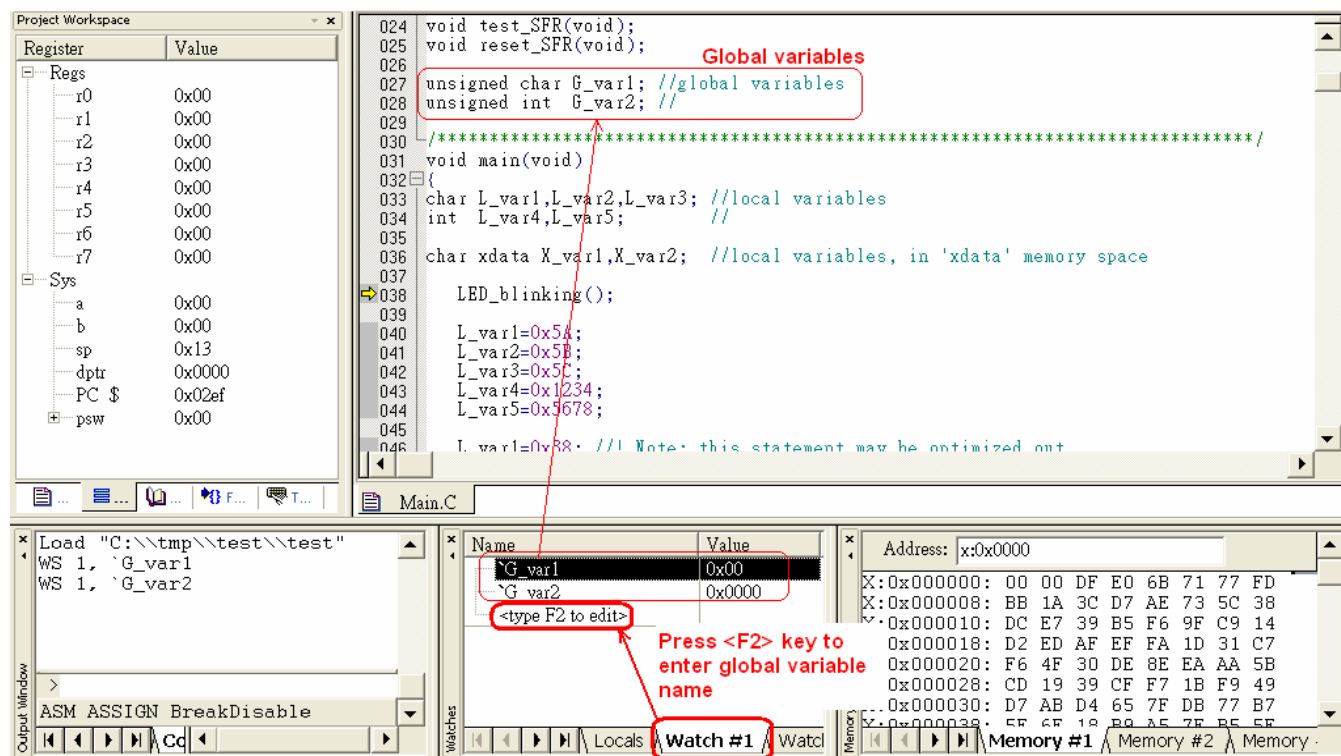


5.2.6 檢視查看視窗

檢視查看視窗可以協助使用者去查看區域變數以及全域變數，如下所示：



要查看全域變數時要先選到 **Watch #1** 或 **Watch #2** 分頁，然後按下 **<F2>** 鍵輸入變數名稱。

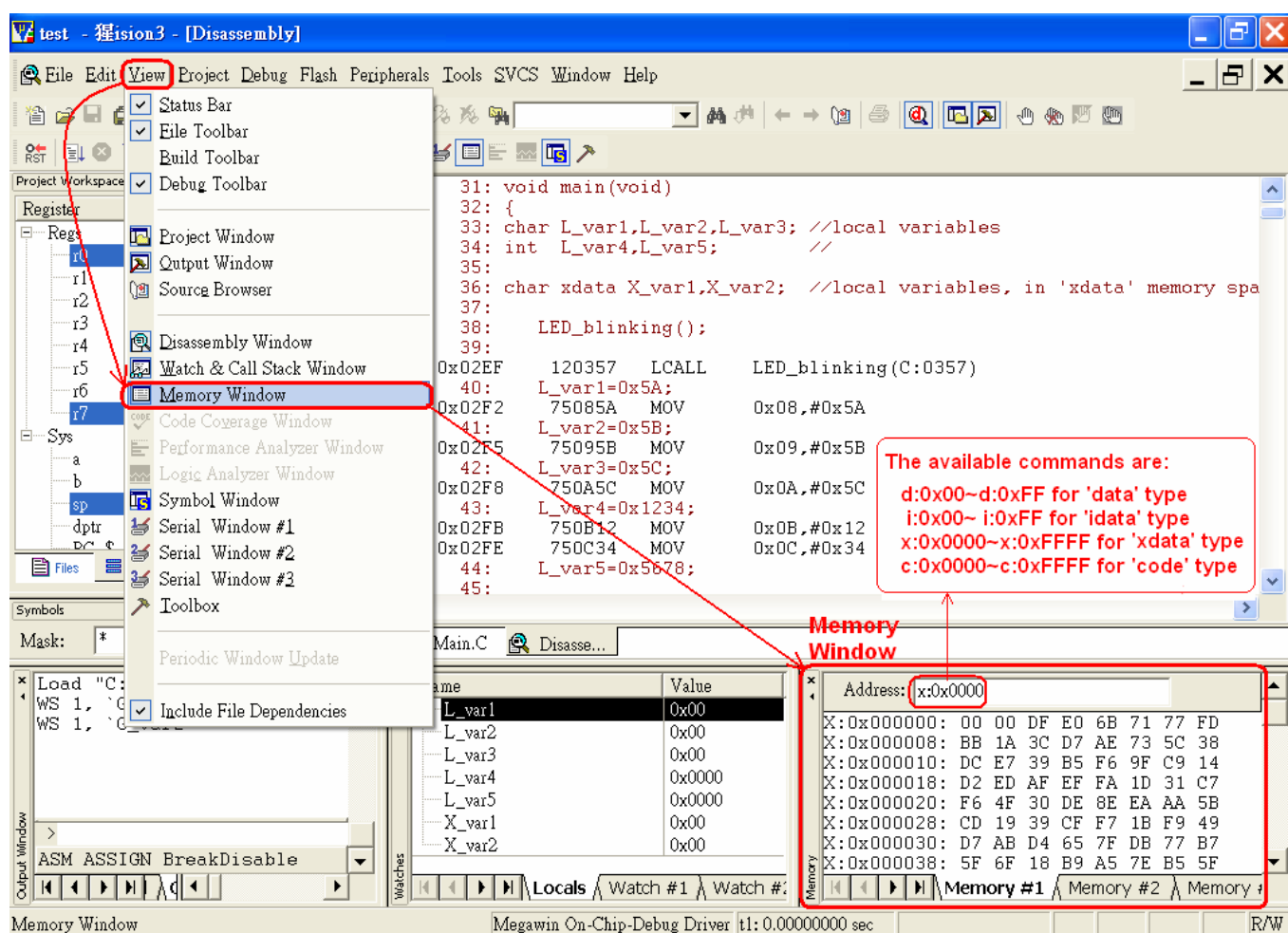


5.2.7 檢視記憶體視窗

要打開這個視窗，可以在主選單中點選 **View**，之後會有顯示一個下拉視窗，再點選 **Memory Window**，如下圖所示，而這個視窗支援的命令有四種：

- (1) 檢視'data'記憶體：d:0x00~d:0xFF
- (2) 檢視'idata'記憶體：i:0x00~i:0xFF
- (3) 檢視'xdata'記憶體：x:0x0000~x:0xFFFF
- (4) 檢視'code'記憶體：c:0x0000~c:0xFFFF

使用者可以用相對應該的命令查看這四種記憶體的內容，如要查看外部記憶'xdata'的內容可以參考第 [6.2 節](#)。



6 注意事項

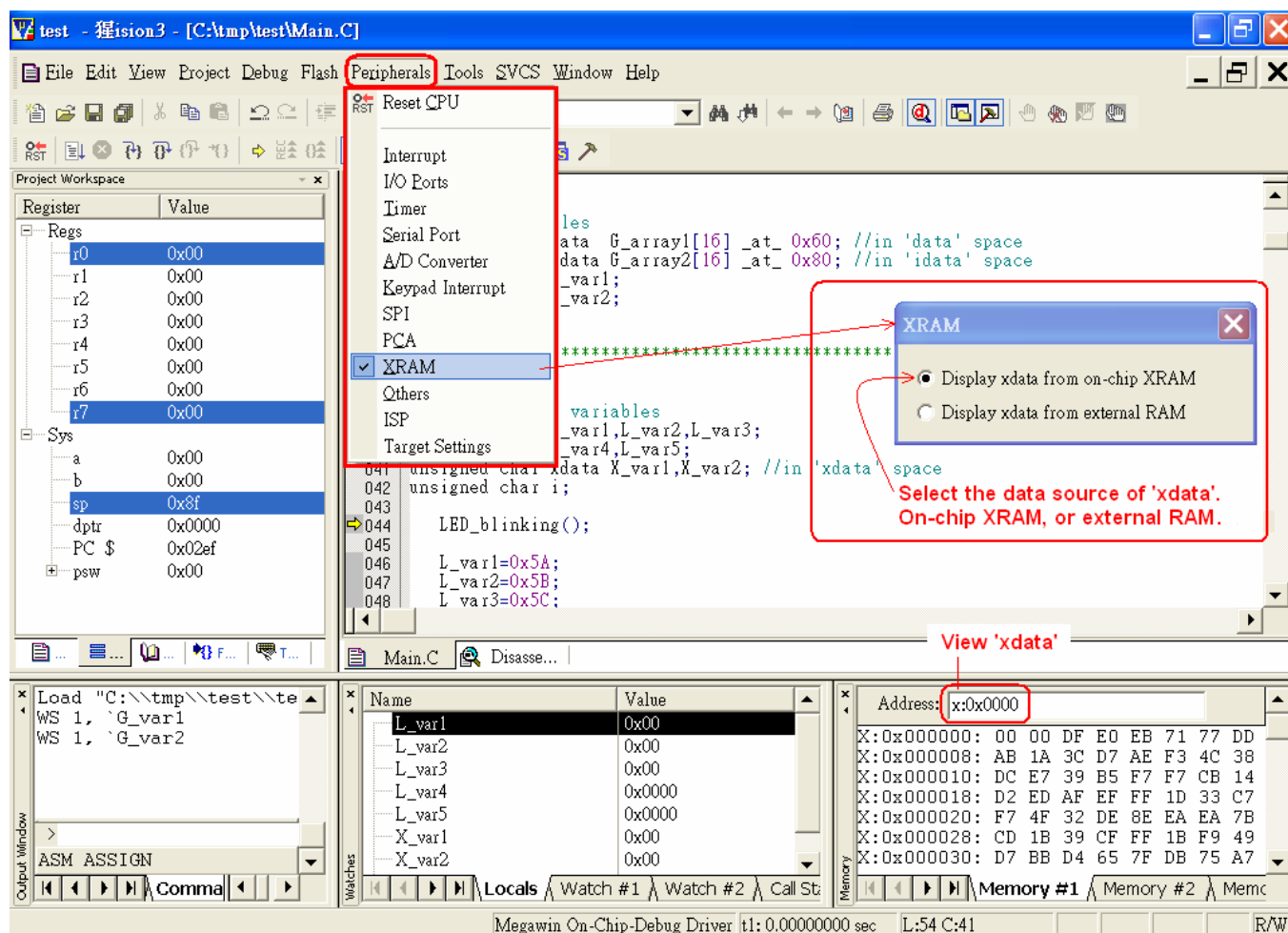
6.1 暫存器定義檔

暫存器定義檔 **REG_MPC82G516.INC** 及 **REG_MPC82G516.H** 定義了所有的特殊功能暫存器 (SFRs) 以及可位元定址的位元。在安裝 OCD ICE 時 (見第二節) 會同時將此暫存器定義檔安裝至 Keil 8051 IDE 的預設目錄內，因此，當使用 Keil 編寫程式時可以直接使用 **#INCLUDE (REG_MPC82G516.INC)** 或是 **#include <REG_MPC82G516.H>** 來將定義檔引入而不需要將定義檔拷貝到您的專案目錄。

6.2 內建 XRAM 及外部資料記憶體

Megawin 8051 提供了內建的記憶體 XRAM (eXpanded RAM)，他的存取方式就如同一般傳統的外接記憶體一樣，MPC82G516 的內建記憶體容量為 1024 個位元組，位址是從 0x0000 到 0x03FF，而由於這個內建記憶體的位址會跟外部記憶體的位址重疊到，因此必須要有一個控制位元來區分這兩個實體記憶體的位址。此時，這個 ERAM (暫存器 AUXR 的 bit-1) 位元扮演了這個角色。因為 C51 的組譯器不會自動去幫使用者切換這兩個記憶體，使用者要使用內建 XRAM 時必須手動去清除這個位元，要使用外接記憶體時必須手動去設定這個位元，而這個位元一開機或是 Reset 後的預設值為 0。

C51 的組譯器提供了兩種不同存取外部記憶體的方式：xdata 以及 pdata (xdata 可以定位到 64K 位元的外部記憶體資料，而 pdata 僅能定位到 256 位元的資料)，當使用者想要直接在記憶體視窗檢視 xdata 或 pdata 而不是在查看視窗時，則必須選取主選單的 **Peripherals- XRAM** 再選擇 **"Display xdata from on-chip XRAM"** 或是 **"Display xdata from external RAM"** 來選擇要檢視內建 XRAM 或是外接記憶體，操作方式如下圖所示：



下面的範例程式顯示出如何同時使用內建 XRAM 以及外接記憶體，我們用顯示內建記憶體“Display xdata from on-chip XRAM”來查看 G_array1[]陣列，用顯示外接記憶體“Display xdata from external RAM”來查看 G_array2[]陣列。

同時使用內建及外部記憶體的範例

```
unsigned char xdata G_array1[512] _at_ 0x0000; // in 'xdata' space, will use on-chip XRAM
unsigned char xdata G_array2[512] _at_ 0x0000; // in 'xdata' space, will use ext. RAM
unsigned int i;
```

```
AUXR&=0xFD; //clear AUXR.1 for on-chip XRAM
for (i=0; i<512; i++) G_array1[i]=0x5A; // fill XRAM with 0x5A
```

```
AUXR|=0x02; //set AUXR.1 for external RAM
for (i=0; i<512; i++) G_array2[i]=0xA5; // fill ext. RAM with 0xA5
```

在組譯時會出現以下的警告訊息，然而這是沒關係的，因為我們故意將 G_array1 及 G_array2 定義成相同的位址，但事實上，我們使用了 ERAM 這個位元切換不同的實體記憶體空間。

```
linking...
*** WARNING L6: XDATA SPACE MEMORY OVERLAP
FROM: 0000H
TO: 01FFH
```

6.3 程式碼最佳化及原始碼除錯

在以下的原始碼中，C51 的組譯器將不會產生“L_var1=0x38”的機械碼，因為下一道指令為“L_var1=0xC7”，因此這道指令將會變成沒有意義的，所以必須將程式碼最佳化關閉才不會將這道指令忽略掉，如[4.4 節](#)的動作

```
unsigned char L_var1;

L_var1=0x38; // ! Note: this statement may be optimized out by the C51 compiler
L_var1=0xC7;
```

所以，當執行原始碼除錯，執行到這道指令時 L_var1 將不會顯示 0x38 而可能顯示一個亂數，事實上，這道指令並沒有對應的機械碼，使用者必須注意到這一點！

有的時後，爲了要除錯，使用者會將程式碼最佳化關閉，此時可能會出現開啓時不會出現的連接錯誤。例如下圖的錯誤訊息，他的意思是你的變數超出了 MCU 記憶體的範圍，要讓這個錯誤消失，唯一的方式是開啓最佳化讓組譯程式更有效率的使用記憶體。

```
linking...
*** ERROR L107: ADDRESS SPACE OVERFLOW
SPACE: DATA
SEGMENT: ?DT?_VP_DISPLAYMODE?VP
LENGTH: 0001H
```

6.4 for 迴圈的原始碼除錯

以下兩組指令對於 8051 的 CPU 是完全一樣的，當用單步原始碼除錯時，在第一組指令是沒有問題的，然而，如果在第二組指令時將會花上許多的時間，我們認為那是因為在 Keil 的除錯功能裡這樣的程序是未知的，在我們尚未得到 Keil 的回覆之前，我們建議盡量使用第一組指令取代第二組指令，如果要使用單步執行來執行這類指令時。要除錯第二組指令的另一個方式是將游標移至第二行並執行 Run-to-Cursor 鍵來跳過第一行。

指令 1:

```
Line1: for (i=0; i<16; i++) {  
Line2:     G_array1[i]=i+0x60;  
Line3: }
```

指令 2:

```
Line1: for (i=0; i<16; i++) G_array1[i]=i+0x60;  
Line2: ...  
Line3: ...
```

6.5 使用除錯時的硬體選項要求

在 dScope-Debugger 模式下有兩個硬體選項的需求：

需求一：除錯的晶片必須在沒有上鎖的狀態

因為當要除錯的晶片上鎖的時後，在 dScope 模式下要下載使用者的應用程式到晶片之前會將晶片做完全刪除，因此所有的硬體選項將會被關閉，所以有可能原本設定的硬體選項會遺失而照成動作不正常。例如，有一顆有設定 IAP 的晶片上鎖了，在進入 dScope-Debugger 模式並下載程式之後，IAP 的設定將會消失，所以晶片執行就會出問題了。

需求二：除錯的晶片必須將 ISP 的功能關閉

由於當 ISP 功能開啓時，這個除錯的晶片每次重開機都會從 ISP 的位址開始執行並且執行 ISP 的程式，所以當進入 dScope-Debugger 模式時會給晶片一個 Reset 的命令，此時將會執行在 ISP 位址內的程式(例 ISP-code)，而不是執行 Keil 所開啓的專案，所以要做除錯功能時，必須將 HWBS 關閉以防止執行到 ISP 的功能。

註：

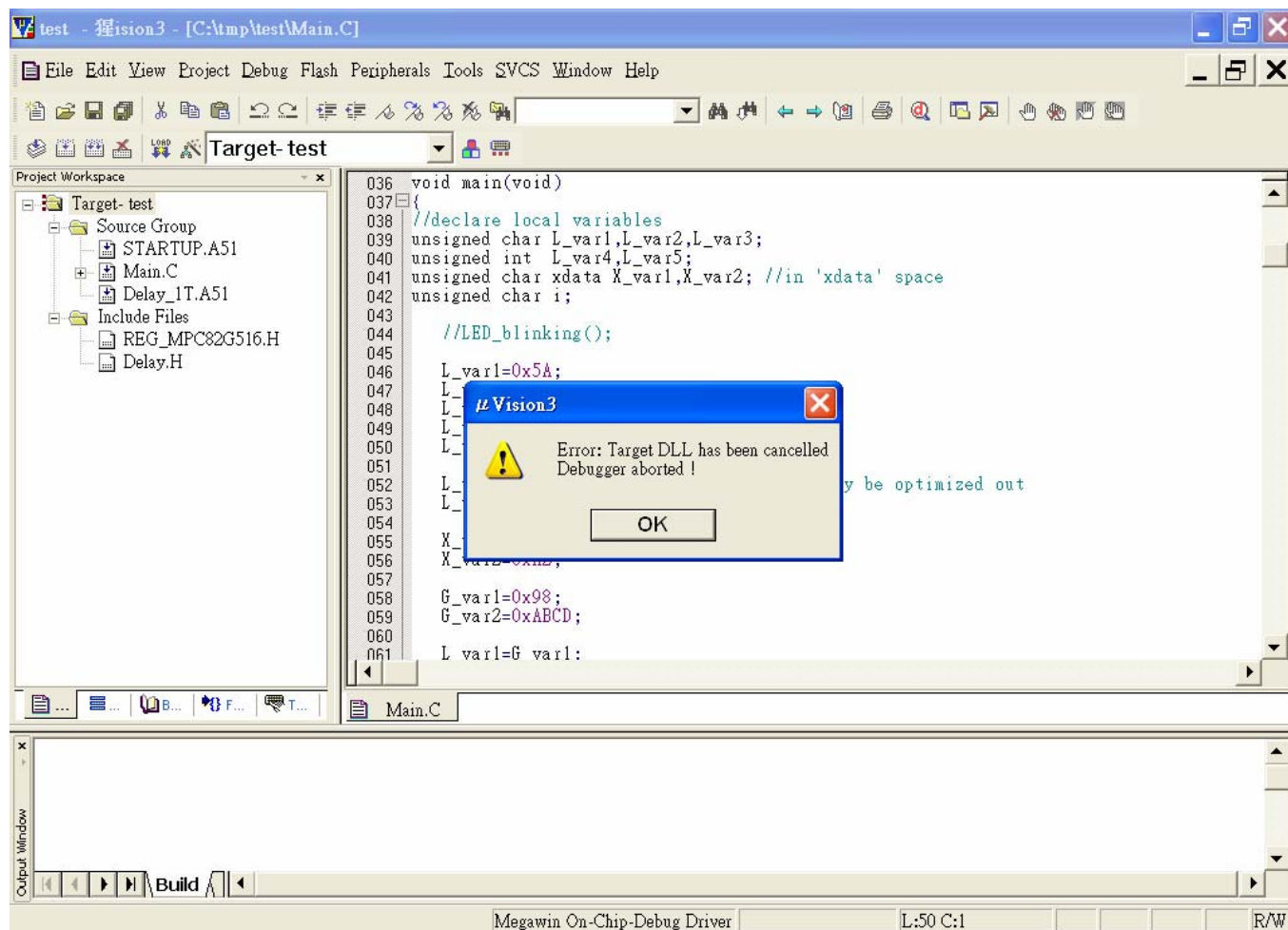
在應用程式除錯完成後，使用者可以用“Megawin 8051 ICP Programmer”來存回原來的硬體設定。

6.6 錯誤訊息

當有下列情況時會顯示錯誤訊息“**Error: Target DLL has been cancelled. Debugger aborted !**”如下圖所示：

- (1) ICE 轉接器固障
- (2) 主板晶片沒有動作(例如沒開電或損毀)
- (3) 連接 ICE 及 MCU 的線壞掉或是接線錯誤

當有此錯誤訊息時，按下“**OK**”然後看看是不是有以上情形以解決這個問題。



6.7 正確的連接 ICE 轉接器到電腦

如果電腦先經由一個 USB 分享器再接到 ICE 轉接器的話，資料傳輸速度將會被大大的減低，所以如果要用 dScope 的功能進行除錯的話，使用者必須直接將 ICE 轉接器直接接到電腦上才行，如圖 6.7.1，而不要經由分享器再到電腦，如圖 6.7.2

圖 6.7.1 直接接到電腦的 USB 埠

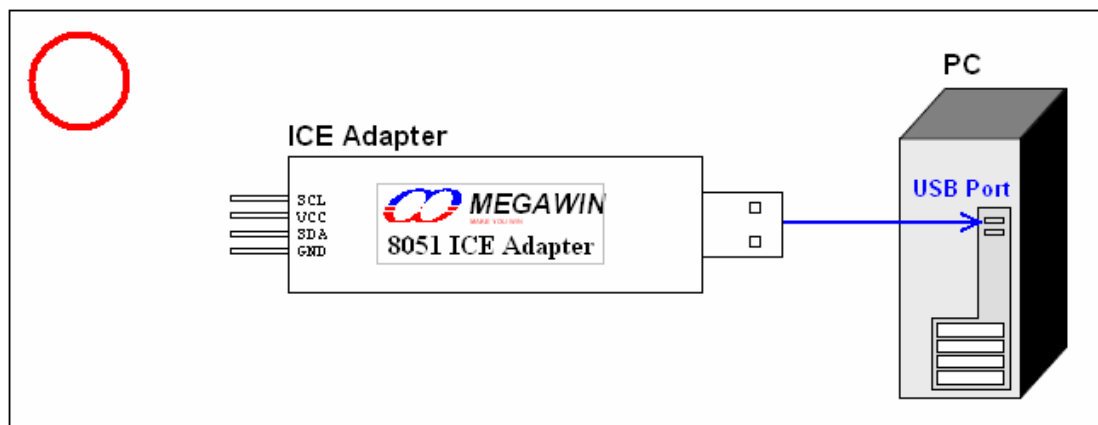
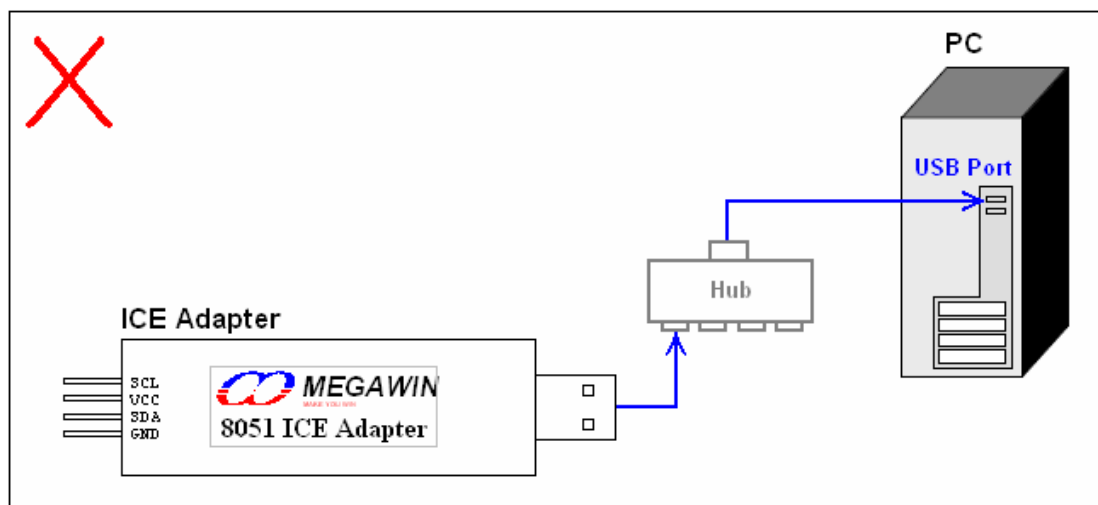


圖 6.7.2 不要經由分享器再接到電腦的 USB 埠



修訂記錄

Revision	Description	Date
v1.00	The first release for beta-site test.	2007/08/15
v1.01	Add notes when installation fails. (Section 3.2)	2007/08/24
v1.02	Change to manually specify the installation path of the Keil software. (Section 3.2)	2007/08/27
v2.00	Add the notification of default installation path of Keil 8051 IDE software. (Section 3.2)	2007/08/29
	Update the Keil IDE Setup. (Section 4.4)	2007/10/08
	Update the Special Notes. (Section 6)	2007/10/08
	The formal released version.	2007/10/08
v2.10	(1) Improve the defect of breakpoint setting. (2) Fix the bug of wrong erasing range when downloading the application code.	2007/12/26
V2.20	(1) Update the data base for all series of MCU in Driver Installer. (2) Removed the function of detecting the ICE adapter when install Driver.	2009/02/27
V2.21	Change the folder name of Driver Install to Database install	2009/04/01