

**HYCON 紘康科技**

---

**HY11S14**

**HY-IDE 軟體使用說明書**

目 錄

1. HY-IDE 入門 .....	4
1.1 簡介 .....	4
1.2 HYIDE 安裝及系統要求 .....	4
1.3 安裝及卸載 .....	4
1.3.1. 安裝 .....	4
1.3.2. 卸載 .....	6
1.4 登錄 .....	7
1.5 DEMO CODE 導入說明 .....	9
1.6 DEMO CODE 操作方式及使用 .....	10
2. HY-IDE 介面說明 .....	11
2.1 HY-IDE 編輯介面 .....	11
2.1.1 編輯視窗 .....	12
2.1.2 檔案 .....	13
2.1.3 編輯 .....	13
2.1.4 檢視視窗 .....	14
2.1.5 組譯&執行 .....	14
2.1.6 介面設定 .....	15
2.1.7 視窗 .....	18
2.1.8 程式架構 .....	18
2.1.9 自訂指令 .....	19
3. HY-IDE 除錯介面 .....	20
3.1 快速執行 .....	21
3.2 RAM 視窗 .....	24
3.3 REGISTER 視窗 .....	27
3.4 WATCH 視窗 .....	29
3.5 堆棧視窗 .....	31
3.6 ADC 視窗 .....	32
3.7 OP 視窗 .....	37
3.8 比較器視窗 .....	39
3.9 暫存器修改記錄 .....	42
3.10 源程式視窗下的 HINT 功能 .....	44
4. 燒錄視窗 .....	46
4.1 介面設定 .....	46
4.2 操作步驟 .....	48
4.2.1 開啟檔案與組譯 .....	48

4.2.2 Download HEX File.....	50
<b>4.3 PC連線燒錄OTP.....</b>	<b>51</b>
4.3.1 晶片檢查(Blank Check).....	52
4.3.2 編程晶片(Program).....	52
4.3.3 確認編程晶片(Verify).....	53
4.3.4 讀取晶片(Read).....	53
4.3.5 AUTO.....	54
<b>4.4 離線燒錄.....</b>	<b>55</b>
4.4.1 燒錄說明.....	55
4.4.2 限制燒錄次數.....	57
<b>5.修訂記錄.....</b>	<b>58</b>

注意：

- 1、本說明書中的內容，隨著產品的改進，有可能不經過預告而更改。請客戶及時到本公司網站下載更新 <http://www.hycontek.com>。
- 2、本規格書中的圖形、應用電路等，因第三方工業所有權引發的問題，本公司不承擔其責任。
- 3、本產品在單獨應用的情況下，本公司保證它的性能、典型應用和功能符合說明書中的條件。當使用在客戶的產品或設備中，以上條件我們不作保證，建議客戶做充分的評估和測試。
- 4、請注意輸入電壓、輸出電壓、負載電流的使用條件，使 IC 內的功耗不超過封裝的容許功耗。對於客戶在超出說明書中規定額定值使用產品，即使是瞬間的使用，由此所造成的損失，本公司不承擔任何責任。
- 5、本產品雖內置防靜電保護電路，但請不要施加超過保護電路性能的過大靜電。
- 6、本規格書中的產品，未經書面許可，不可使用在要求高可靠性的電路中。例如健康醫療器械、防災器械、車輛器械、車載器械及航空器械等對人體產生影響的器械或裝置，不得作為其部件使用。
- 7、本公司一直致力於提高產品的品質和可靠度，但所有的半導體產品都有一定的失效概率，這些失效概率可能會導致一些人身事故、火災事故等。當設計產品時，請充分留意冗餘設計並採用安全指標，這樣可以避免事故的發生。
- 8、本規格書中內容，未經本公司許可，嚴禁用於其他目的之轉載或複製。

## 1. HY-IDE入門

### 1.1 簡介

為了方便客戶使用紘康科技(HYCON)的 MCU 系列產品，開發出 HY-IDE 的發展環境，客戶能在此開發平台上模擬仿真其終端產品，並將程序燒錄到 HY 系列產品的 OTP 上，使客戶的終端產品能快速的發展並銷售到市面上。

### 1.2 HyIDE安裝及系統要求

運行 Hycon IDE 所需的最低系統配置：

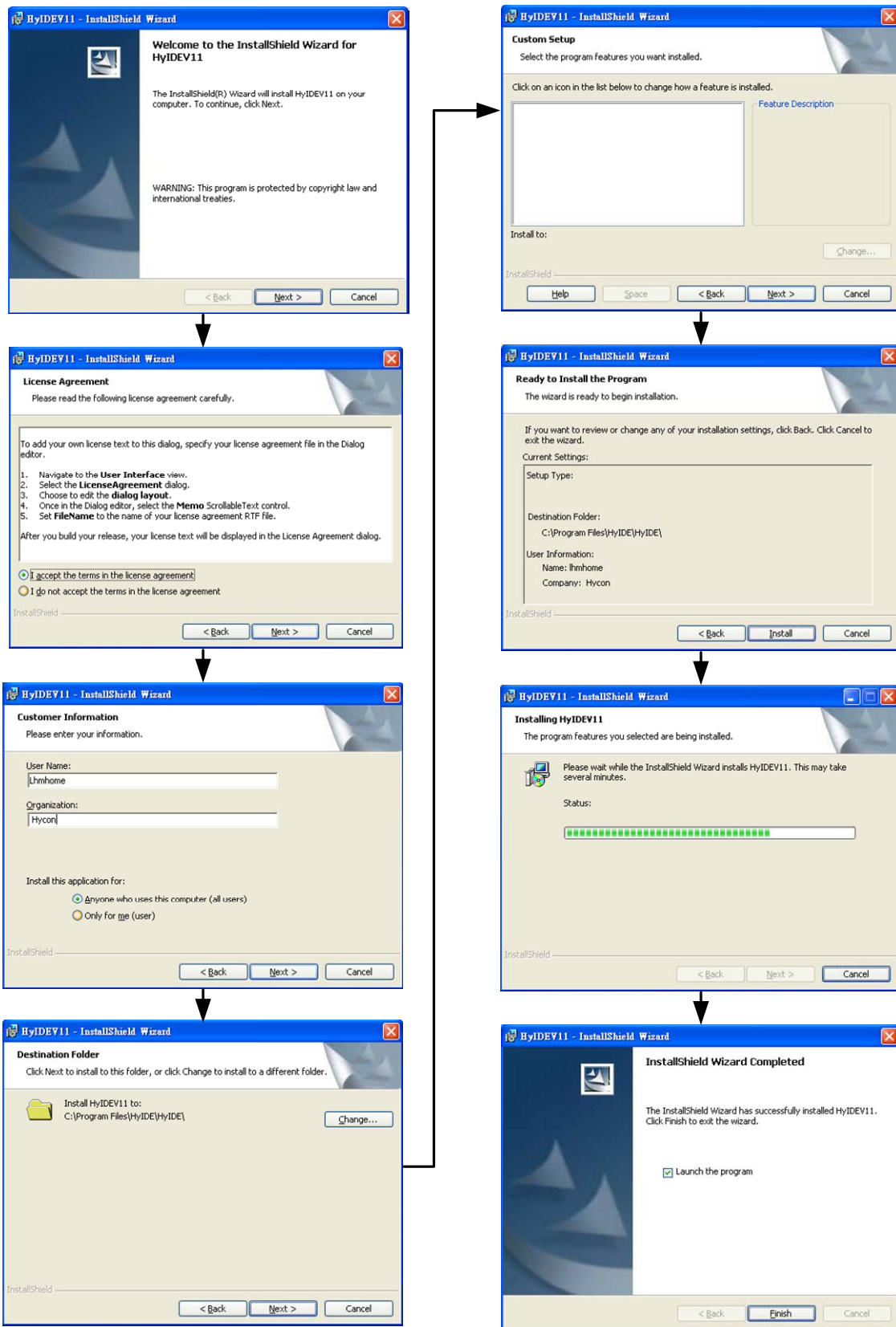
- PC 硬體需求
  - PC 相容的奔騰 (PENTIUM®) 級系統
  - 128 MB 記憶體 (推薦 256MB)
  - 10 GB 硬碟空間
  
- OS
  - Windows 98SE
  - Windows 2000
  - Windows XP
  - Windows Vista
  - Windows 7
  
- 適用下列介面模式
  - USB Port

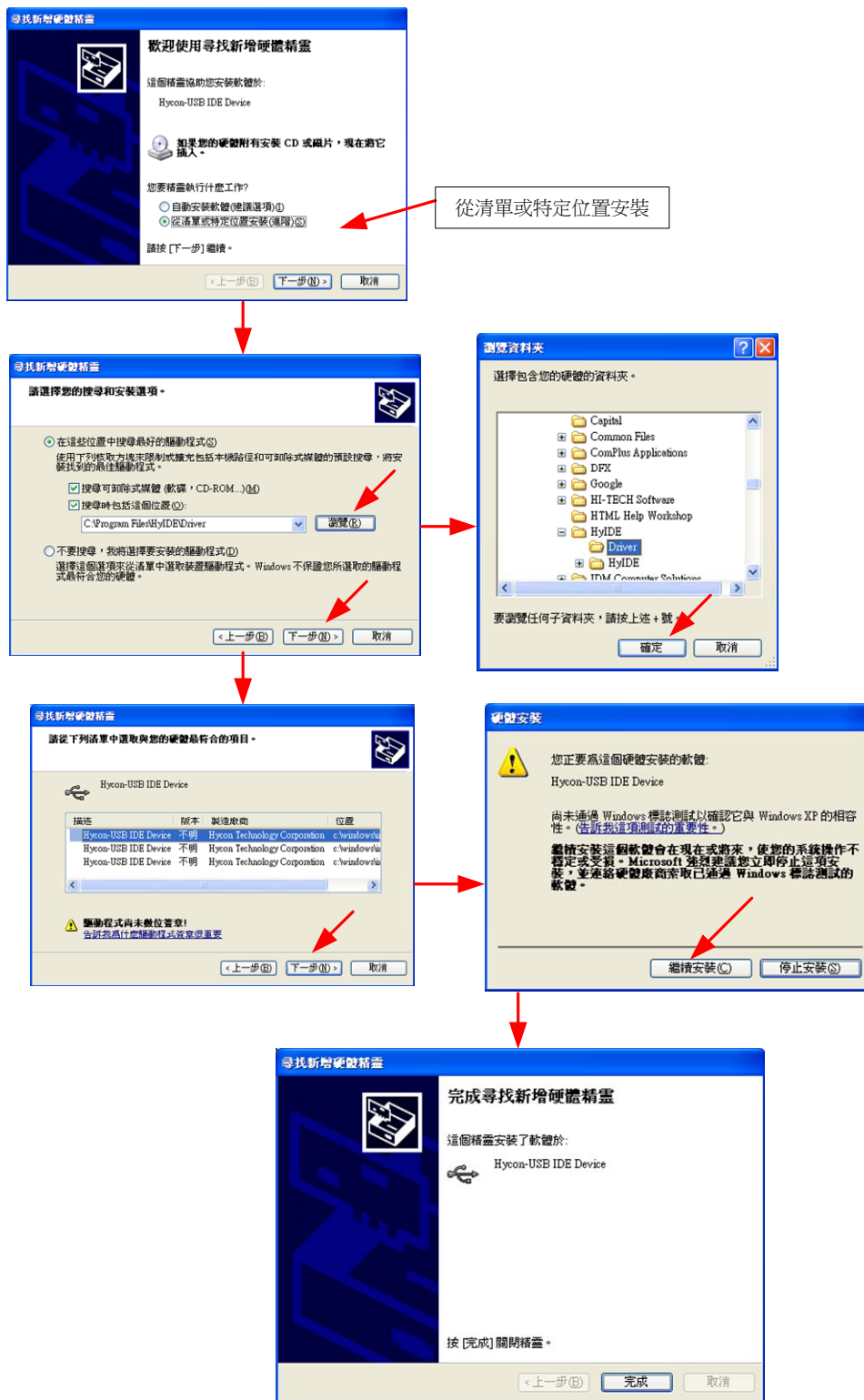
### 1.3 安裝及卸載

#### 1.3.1. 安裝

對於某些 Windows 作業系統，要在電腦中安裝軟體，需要管理員訪問許可權。

- 在光碟或檔案中尋找並執行 Setup.exe 執行檔
- 按畫面照指示一步一步向下執行安裝步驟，如 圖 1
- 第一次安裝需要啟動USB驅動程式，安裝步驟如 圖 2





### 1.3.2. 卸載

請到控制台的“新增或移除程式”尋找 HyIDEV11 選擇移除程式，即可。

## 1.4 登錄

客戶首次使用 ICE 硬體模擬，或是燒錄 OTP 晶片時。如果發生以下訊息或是 IDE 使用發生不正常當機現象，需要做登錄動作。

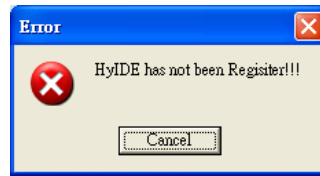


圖 3

### 登錄步驟

1. 請在包裝盒上查詢 HyIDE Machine Number(HyIDE 編碼)，並將此編碼透過 Email 或上網登錄到紘康科技後，會回傳另外一組客戶登錄碼。
2. 透過 PC 連接 USB 到 HyIDE 控制板(Control Board)。
3. 執行 HyIDE 軟體(HyIDE.exe)，並到 “選項”內執行 “登錄”。
4. 將客戶登錄碼填入 “Register Number”框內，然後點擊 “Write”。

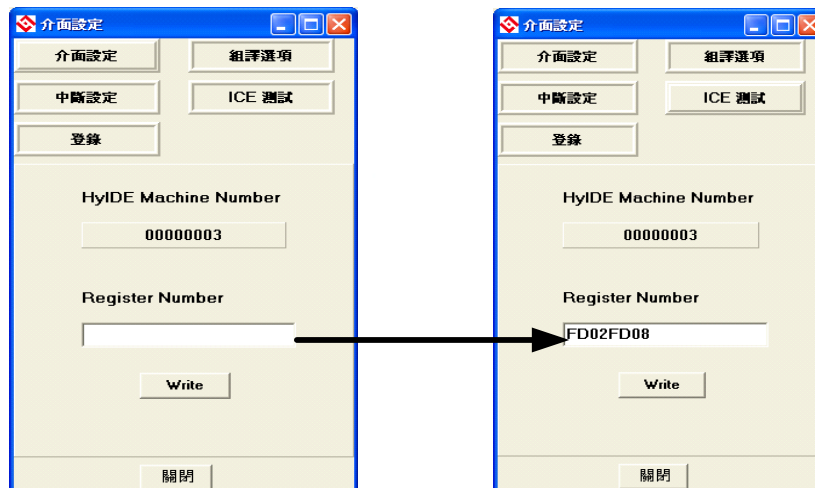


圖 4

5. 如果出現以下訊息表示登錄成功



圖 5

6. 如果出現以下訊息表示登錄失敗

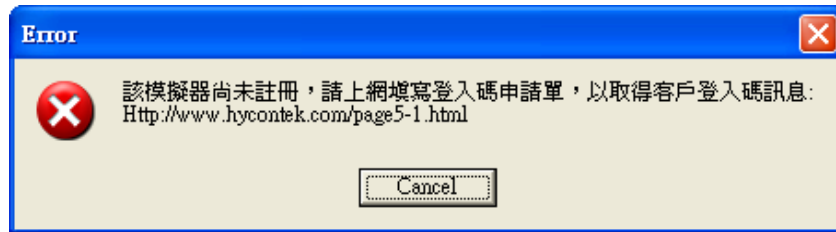


圖 6

7. 一旦登錄成功後，無論在“Register Number”輸入任何值，將不再被寫入，客戶可以安心使用。



## 1.5 Demo Code導入說明

- 開啟 C:\Program Files\HyIDE\HyIDE\DemoCode\KitchenScale\KitchFor11P13.asm 主程式
- 設定為組譯主檔
- 組譯並進行除錯



圖 7

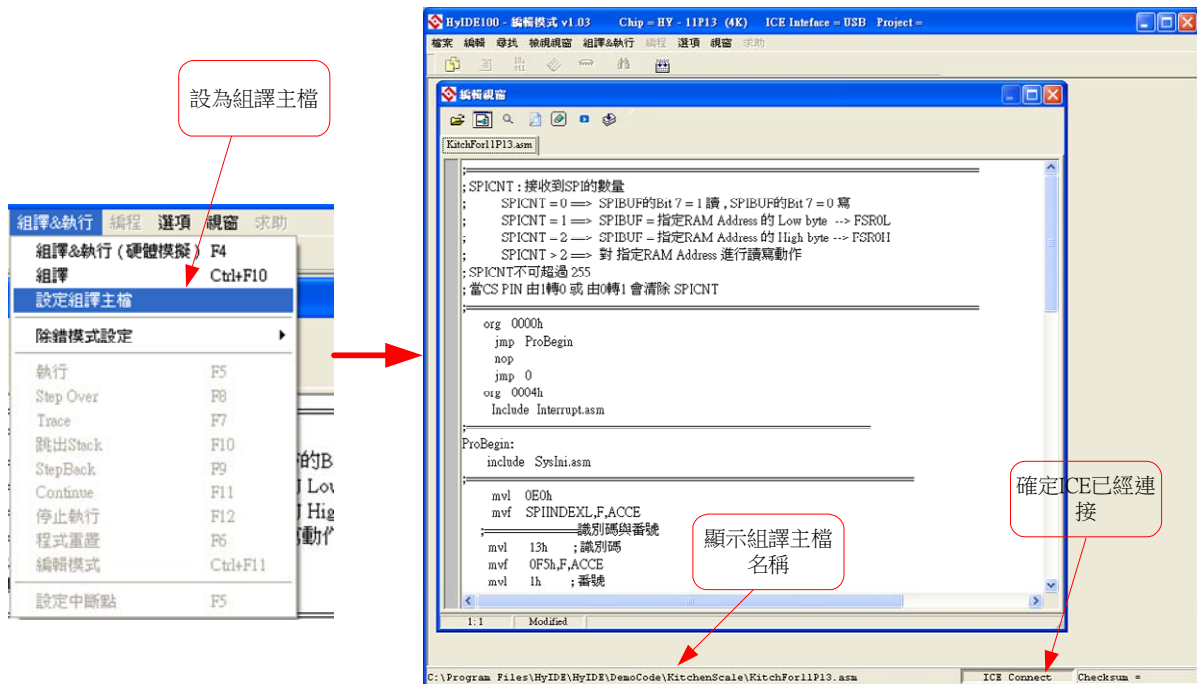


圖 8

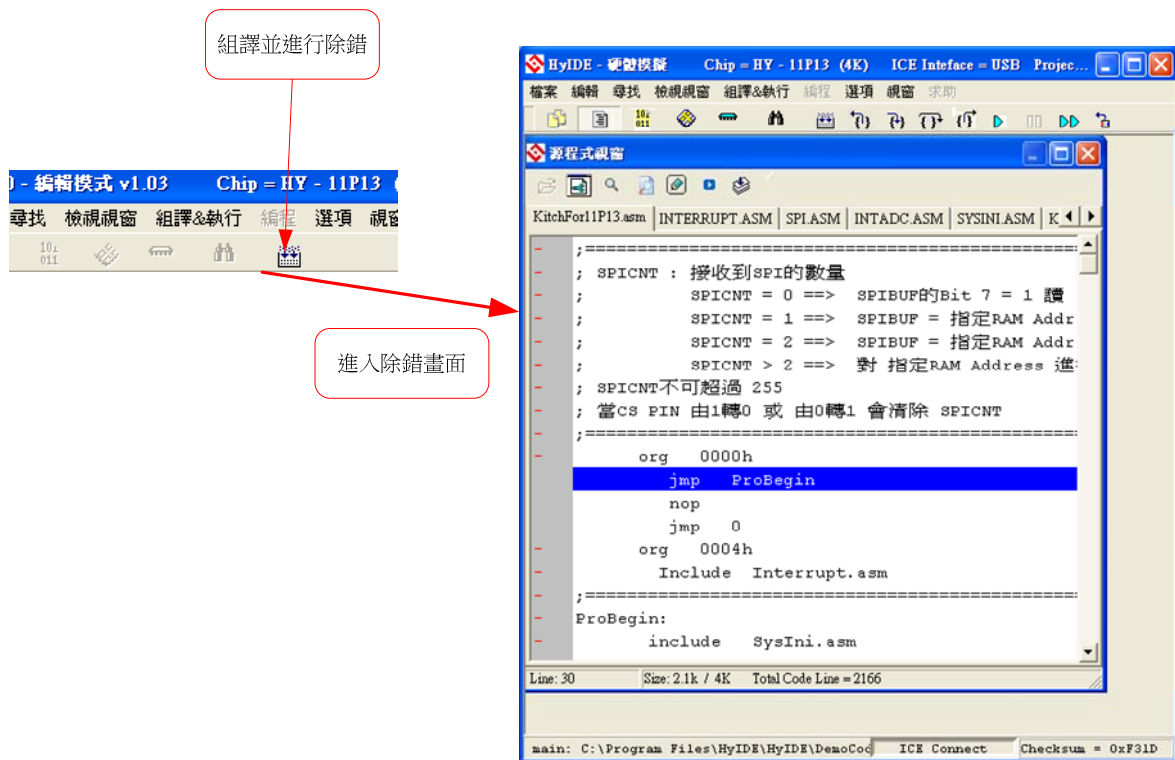


圖 9

- 使用者可使用任何編輯器，來編輯 Source Code，只要能以 ASCII Code 的形式儲存及可。程式組譯時，會重新載入 Source Code 以確保程式正確性。下面章節將一一介紹除錯與編輯的功能。組譯並進行除錯

### 1.6 Demo Code操作方式及使用

- 執行 HYIDE 軟體安裝後，於目錄 C:\Program Files\HyIDE\HyIDE\DemoCode 下會有提供 Demo Code 供使用者參考。
- Demo Code參考文件放置於 <http://www.hycontek.com/page2.html> 網頁之產品應用說明書中。以下列出相對照文件參考使用說明：
  - KitchenScale 參考 APD-SD18007 (廚房秤應用說明)
  - KitchenScale-SINC3 參考 APD-SD18003 (50/60Hz Rejection 的解決方案)
  - HY11P32Democode 參考 APD-SD18004 (無開關自動測量應用)

## 2. HY-IDE 介面說明

### 2.1 HY-IDE 編輯介面

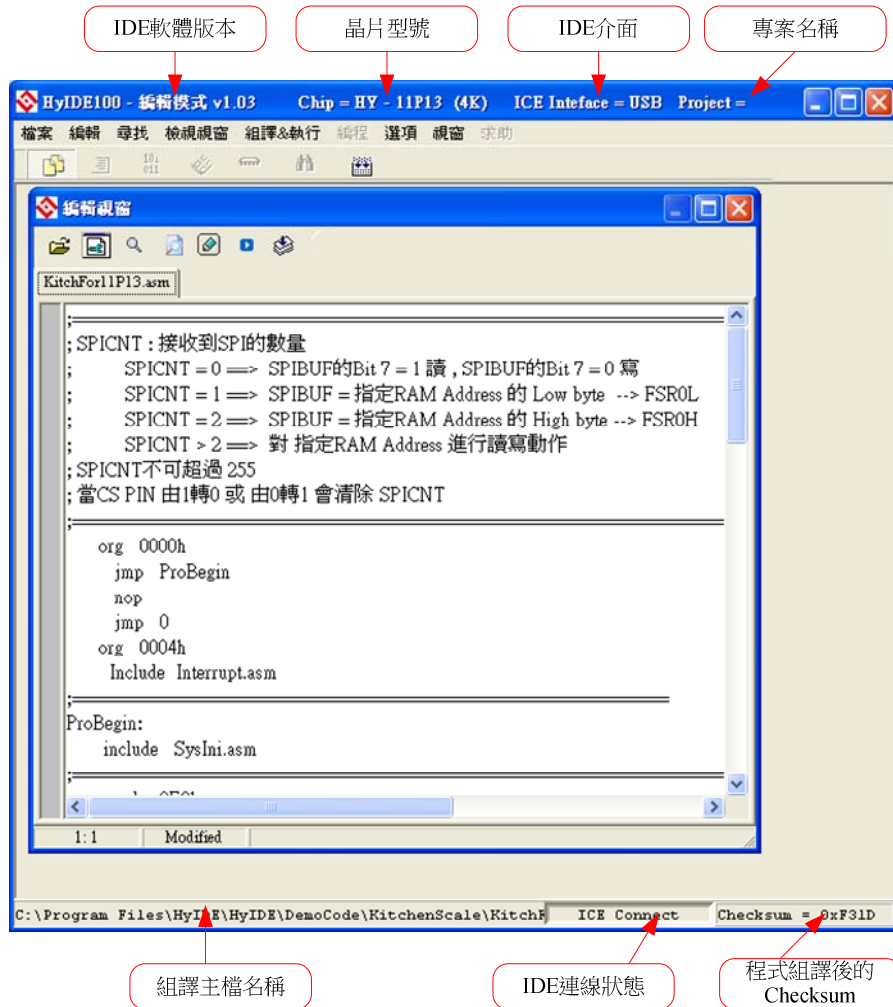









圖 10

### 2.1.1 編輯視窗

- 開啟舊檔   
開啟存放在磁碟中已編輯好的檔案。
- 設定標籤   
設定標籤，當開啟檔案很多時，可利用此項快速回到設定標籤處。
- 跳至標籤   
跳到以設定的標籤處。
- 尋找字串   
尋找已輸入過的字串。
- 尋找下一個字串   
尋找下一個字串。
- 切換顯示頁面   
當開啟檔案很多時，可利用此項切換檔案。
- 組譯   
只有組譯，不進入除錯狀態。  
當組譯完成後會出現訊息欄



### 2.1.2 檔案

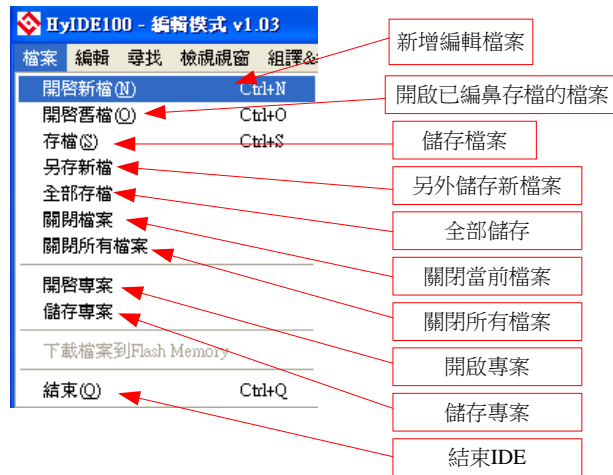


圖 11

- 開啓新檔 → 新增編輯檔案
- 開啓舊檔 → 開啟已儲存的編輯檔案
- 存檔 → 儲存檔案
- 另存新檔 → 將檔案用新的名稱儲存
- 全部儲存 → 儲存全部檔案
- 開啟專案 → 專案包括 (晶片型號、IDE 介面、組譯主檔名稱、當前開啟的狀態、Checksum)，開啟專案後會載入專案的狀態。
- 儲存專案 → 儲存專案
- 結束 → 結束 HyIDE 程式

### 2.1.3 編輯

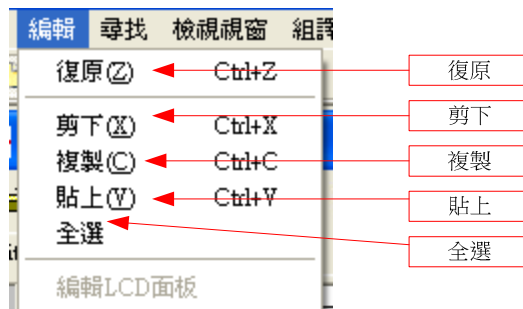


圖 12

- 復原 → 回復上一次鍵入或刪除
- 剪下 → 剪下選取的區域
- 複製 → 複製選取的區域
- 貼上 → 貼上負制的區域
- 全選 → 全部選擇

### 2.1.4 檢視視窗

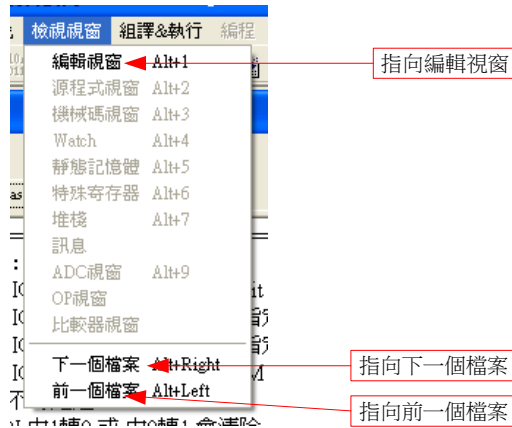


圖 13

- 編輯視窗 → 將編輯視窗指定為當前的活動視窗
- 下一個檔案 → 將下一個檔案指定為當前的活動視窗
- 前一個檔案 → 將前一個檔案指定為當前的活動視窗

### 2.1.5 組譯&執行

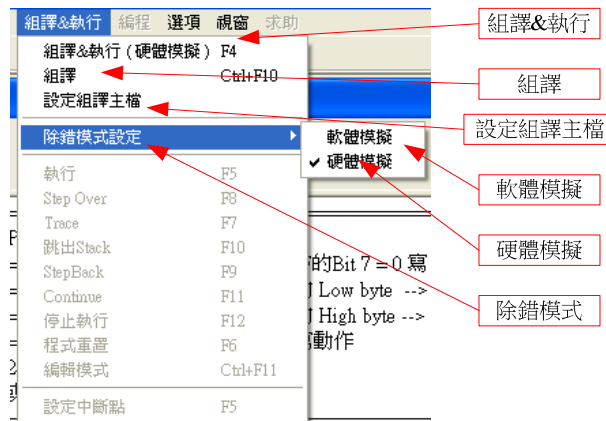


圖 14

- 組譯&執行 → 組譯 Source Code 並執行除錯模式
- 組譯 → 只組譯程序，不執行除錯，此項組譯並不會根據晶片型號產生錯誤訊息，只有當語句有誤時才會顯示錯誤訊息，一般用於製作 OBJ Code (Object)。
- 設定組譯主檔 → 設定為組譯主檔，Compiler 產生的檔案名稱如 Hex、MAP、ASC... 都將以此名稱做為檔案名稱。
- 除錯模式設定 → 選擇使用軟體除錯或硬體除錯。

### 2.1.6 介面設定

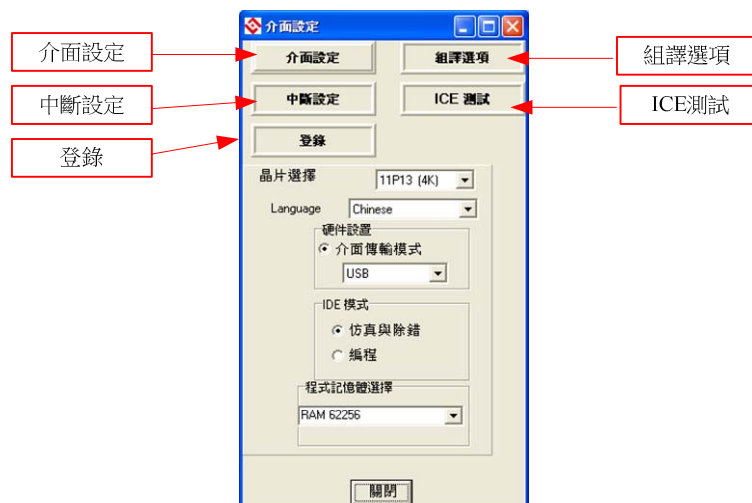


圖 15

- 介面設定(由選項中選取)



圖 16

- 晶片選擇：選擇晶片型號，Compiler 會根據選擇的型號組譯出燒錄設定檔案，並判斷是否有誤用到不存在的 Register 或 SRAM，或程式是否超出 ROM Size。
- 語言選擇：可選擇英文或中文介面。
- 傳輸介面選擇：選擇 IDE 傳輸介面。
- 模式選擇：仿真與除錯、燒錄編程兩種選擇。

- 組譯選項



圖 17

- 組譯產生附檔：可選擇輸出以下檔案
- 二進位檔：Hex
- obj 檔：obj
- List 檔：lst
- ADCII 檔：asc
- 堆疊操作：依不同晶片型號選擇，可選擇堆棧滿後重置，Stack over Stop 功能，當選擇此項時，Compiler 會加入入 Hex 中，燒錄時會將此設定燒入 OTP 的設定中。
- 燒錄次數限制：參考燒錄視窗之 [介面設定](#) 章節。
- 編輯功能字型選擇：選擇編輯器的字型大小。
- 未使用區域填滿：程式中未使用到的區域選擇填滿 0x0000 或 0xFFFF。
- 組譯精簡：選擇是否啟動精簡組譯，當 JMP 或 CALL 小於 2K 時，會自動轉換成 RJ 或 RCALL；但如果 CALL 後面的參數有設定時則不會轉換成 RCALL。
- 燒錄保護：燒錄視窗之 [介面設定](#) 章節。



- 中斷設定



圖 18

- Stack 溢位時停止操作：當 Stack 溢位後 IDE 停止執行
- 監視的位址：選擇要監視 Register 或 RAM，當程式執行到 RAM 或 Register 的值與監視的 Data 相等時程式停止。
- 監視的 Data：當監視的 Data 填上後，表示要監視的值
- 監視 RAM bit：當監視的 bit 勾選後，表示要啟動監視功能，並且當 Data 值的 bit 與勾選的 bit 相等時程式停止。

- ICE 測試

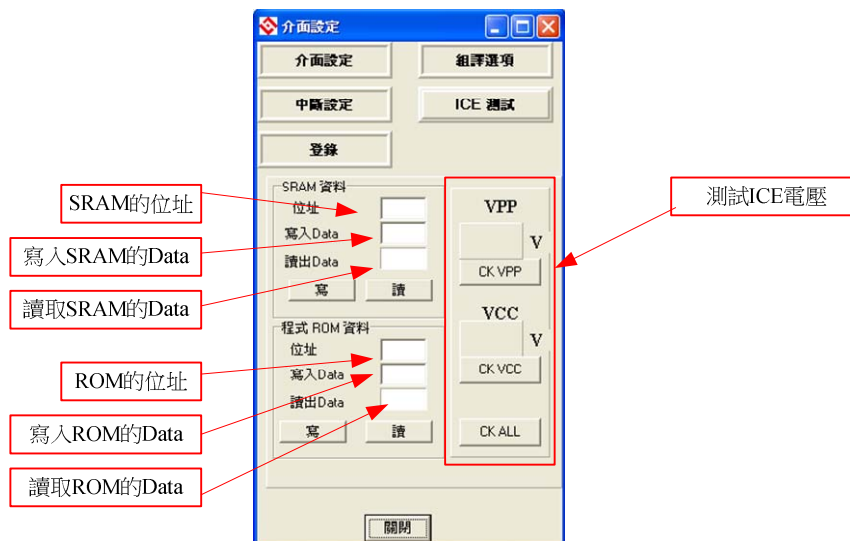


圖 19

### 2.1.7 視窗

可選擇所有開啟的視窗做垂直或水平的排列。

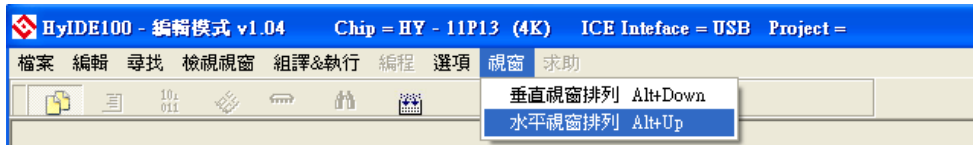


圖 20

### 2.1.8 程式架構

在開始編輯新的程式之前，須先由介面設定中設定晶片選擇；

不同晶片搭配不同 Instruction Set，依晶片型號定義會區分有 H08A, H08B 指令集；

使用者一開始可以先參考軟體所附的demo code，本文 1.6 章節有demo code導入說明，並搭配下列程式架構開始撰寫程式，以下列出程式的基本架構說明：

- 程式名稱定義為: **\*\*\*.ASM**
- 暫存器名稱或 RAM Definition 定義為: **\*\*\*.INC**
- 如下，共有多個程式內容：
  - “Main.asm”、“Initial.asm”、“Interrupt.asm”、“Sub.asm”、“Mian.inc”、“H08.inc”
- “Main.asm” structure:

<b>ORG</b>	<b>00H</b>	;程式名稱可為任意名稱
		;宣告程式開始
<b>JMP</b>	<b>BEGIN</b>	;跳躍到主程式
<b>ORG</b>	<b>04H</b>	;宣告中斷旗標位置
<b>Include</b>	<b>Interrupt.asm</b>	;引用”Interrupt.asm”中斷副程式；
		;include 檔案限制最多 100 個.
<b>BEGIN:</b>		;主程式開始. Label name 的定義可以為任意字
<b>Include</b>	<b>Initial.asm</b>	;引用”Initial.asm”硬體初始化副程式
<b>JMP</b>	<b>T1</b>	;跳躍到 T1 副程式
...		
<b>T1:</b>		
<b>NOP</b>		
<b>Include</b>	<b>Sub.asm</b>	;引用”Sub.asm”副程式
<b>Include</b>	<b>H08.inc</b>	;HY11P 系列特殊暫存器名稱、位址定義
<b>Include</b>	<b>Main.inc</b>	;RAM 名稱、位址定義
<b>END</b>		;程式結束
- 參考文件：
  - IP使用說明：[User's Guide](#)
  - Instruction Set使用說明：[H08A指令集說明書](#) or [H08B指令集說明書](#)
  - HYIDE Compiler使用說明：[HY-MCU COMPILER](#)

### 2.1.9 自訂指令

- HYIDE 從 V1.6 版開始增加使用者自訂指令功能，此功能提供使用者將所熟悉的 MCU 指令，自行定義使其與 HY11P 系列指令相同
- 使用說明：
  1. 所有自訂指令功能設定於安裝目錄下 `Inst.txt` 檔案，共分兩列，每行第一個指令(第一列)為 HYCON 原始指令名稱，使用者不可變更；第二個(第二列)指令為"使用者"自行定義指令名稱。
  2. 第一個及第二個指令中間只可以空格、多格或 Tab 區分開。
  3. 第二個指令後面可以使用分號(;)做註解。
  4. 第二個指令名稱定義可以和第一個指令名稱相同。
  5. 第二個指令名稱不能定義成除了同一行指令名稱外的任一 HYCON 原始指令名稱，否則該行定義無效，仍採用原始 HYCON 指令名稱組譯程式。
  6. 當使用者自行定義第二個指令名稱後，則程式組譯時，第一個或是第二個指令名稱都可以使用。
  7. 每一行僅能自定義一個指令名稱，重覆則該行指令無效。
- 例如：

```
JMP JUMP JMM JPP JU ;×錯誤的定義方式
```
- 重覆定義指令或定義自訂義的指令將視為無效定義。  
例如：

```
JMP JUMP  
JMP JPP ;×指令名稱再度定義，在此 JUMP 將為無效指令，只有 JPP 有效。  
JPP JU ;×無法使用自定名稱再度定義  
JMP JN ;×無法定義成 HYCON 原本存在指令名稱
```

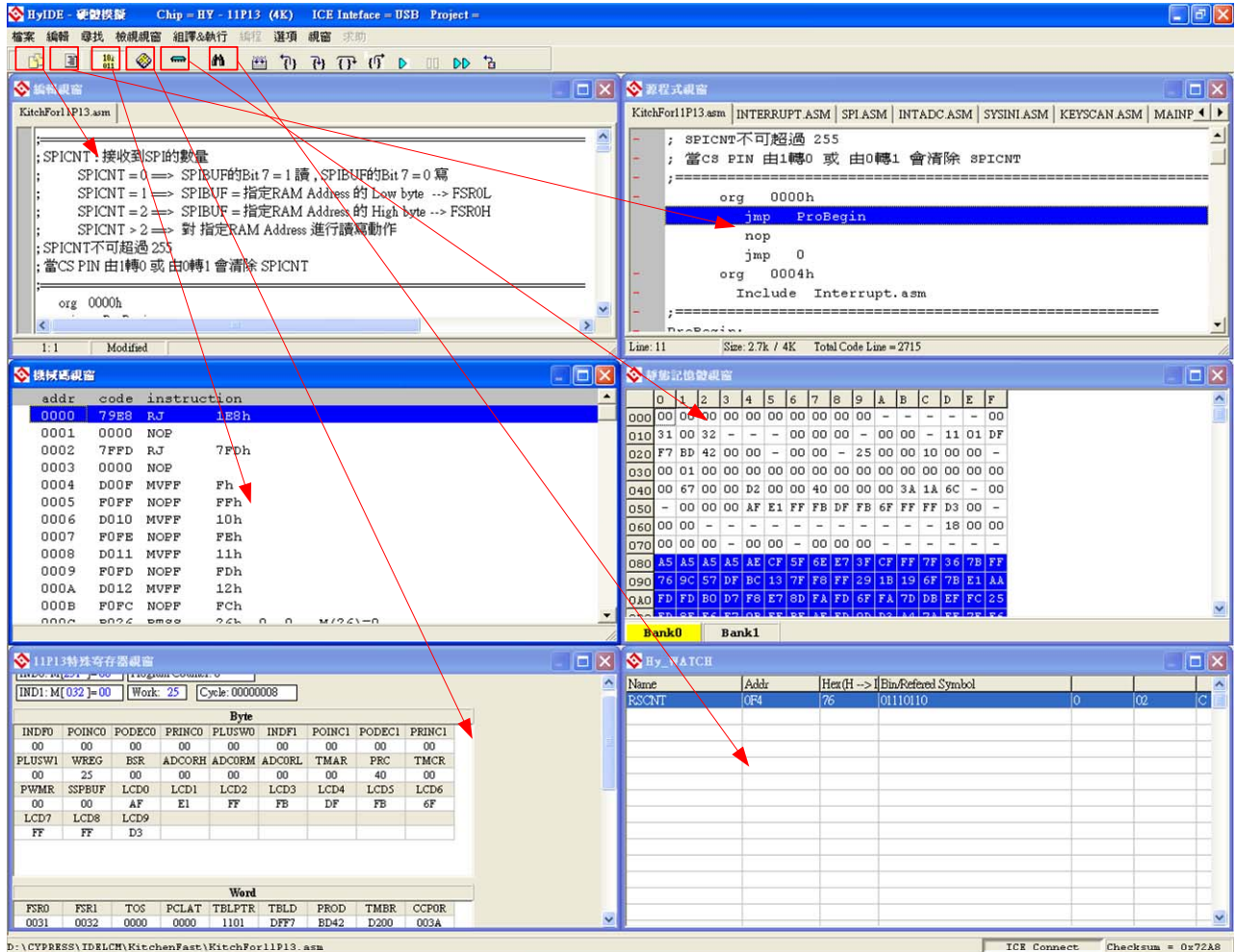
正確的用法是：

```
JMP JUMP
```

### 3. HY-IDE 除錯介面

分為硬體除錯與軟體除錯

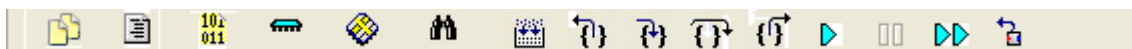
- 硬體除錯  
指示欄棒為藍色
- 軟體除錯  
指示欄棒為綠色



The screenshot displays the HY-IDE IDE interface with several windows open:

- Source Code Window:** Shows assembly code for 'KitchFor11P13.asm'. Comments describe SPICNT values and SPI operations. A blue highlight is visible on the 'jmp ProBegin' instruction.
- Instruction Window:** Lists assembly instructions with their addresses and codes. The instruction at address 0000 is highlighted in blue.
- Memory Window:** Shows a memory dump with addresses and hex values. A blue highlight is visible on the memory at address 0000.
- Registers Window:** Displays the state of various registers, including INDF0, PLUSW1, PWSR, LCD0, LCD1, LCD2, LCD3, LCD4, LCD5, LCD6, LCD7, LCD8, LCD9, FSR0, FSR1, TOS, PCLAT, TBLPTR, TBLD, PROD, TMRB, CCPOR, and CCPOR.
- Watch Window:** Shows a table of watched variables. The variable 'RSCNT' is listed with address 0F4, hex value 76, and bin value 01110110.




### 3.1 快速執行



- 快速視窗切換

<p>(1) 切換至 Edit 視窗 </p>	<p>(2) 切換至 source 視窗 </p>																																																																																																		
<p>(3) 切換至 Hex 視窗 </p> <table border="1"> <thead> <tr> <th>addr</th> <th>code</th> <th>instruction</th> </tr> </thead> <tbody> <tr><td>0000</td><td>78DD</td><td>RJ DDh</td></tr> <tr><td>0001</td><td>0000</td><td>NOP</td></tr> <tr><td>0002</td><td>7FFD</td><td>RJ 7FDh</td></tr> <tr><td>0003</td><td>0000</td><td>NOP</td></tr> <tr><td>0004</td><td>DOOF</td><td>MVFF Fh</td></tr> </tbody> </table>	addr	code	instruction	0000	78DD	RJ DDh	0001	0000	NOP	0002	7FFD	RJ 7FDh	0003	0000	NOP	0004	DOOF	MVFF Fh	<p>(4) 切換至 Ram 視窗 </p> <table border="1"> <thead> <tr> <th></th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> <th>8</th> <th>9</th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> </tr> </thead> <tbody> <tr><td>000</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>010</td><td>DB</td><td>00</td><td>00</td><td>-</td><td>-</td><td>-</td><td>00</td><td>00</td><td>00</td><td>-</td><td>00</td><td>00</td><td>-</td><td>00</td><td>00</td></tr> <tr><td>020</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>-</td><td>00</td><td>80</td><td>-</td><td>15</td><td>01</td><td>00</td><td>10</td><td>00</td><td>00</td></tr> <tr><td>030</td><td>00</td><td>01</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>-</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td></tr> </tbody> </table>		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	000	00	00	00	00	00	00	00	00	00	00	-	-	-	-	-	010	DB	00	00	-	-	-	00	00	00	-	00	00	-	00	00	020	00	00	00	00	00	-	00	80	-	15	01	00	10	00	00	030	00	01	00	00	00	00	00	00	-	00	00	00	00	00	00
addr	code	instruction																																																																																																	
0000	78DD	RJ DDh																																																																																																	
0001	0000	NOP																																																																																																	
0002	7FFD	RJ 7FDh																																																																																																	
0003	0000	NOP																																																																																																	
0004	DOOF	MVFF Fh																																																																																																	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E																																																																																				
000	00	00	00	00	00	00	00	00	00	00	-	-	-	-	-																																																																																				
010	DB	00	00	-	-	-	00	00	00	-	00	00	-	00	00																																																																																				
020	00	00	00	00	00	-	00	80	-	15	01	00	10	00	00																																																																																				
030	00	01	00	00	00	00	00	00	-	00	00	00	00	00	00																																																																																				
<p>(5) 切換至 Reg 視窗 </p> <table border="1"> <thead> <tr> <th colspan="8">Byte</th> </tr> <tr> <th>INDF0</th> <th>POINCO</th> <th>PODECO</th> <th>PRINCO</th> <th>PLUSW0</th> <th>INDF1</th> <th>POINC1</th> <th>POI</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>00</td> <td>00</td> <td>00</td> <td>00</td> <td>00</td> <td>00</td> <td>00</td> </tr> <tr> <td>PLUSW1</td> <td>WREG</td> <td>BSR</td> <td>ADCORH</td> <td>ADCORM</td> <td>ADCORL</td> <td>TMAR</td> <td>P</td> </tr> <tr> <td>00</td> <td>15</td> <td>01</td> <td>00</td> <td>00</td> <td>00</td> <td>00</td> <td>00</td> </tr> </tbody> </table>	Byte								INDF0	POINCO	PODECO	PRINCO	PLUSW0	INDF1	POINC1	POI	00	00	00	00	00	00	00	00	PLUSW1	WREG	BSR	ADCORH	ADCORM	ADCORL	TMAR	P	00	15	01	00	00	00	00	00	<p>(6) 切換至 Watch 視窗 </p> <table border="1"> <thead> <tr> <th>Name</th> <th>Addr</th> <th>Hex (L-&gt;H)</th> <th>Bin/Refered Symbol</th> </tr> </thead> <tbody> <tr> <td>CHADDR1</td> <td>0A3</td> <td>000129</td> <td></td> </tr> <tr> <td>CHADDR2</td> <td>0A6</td> <td>000129</td> <td></td> </tr> </tbody> </table>	Name	Addr	Hex (L->H)	Bin/Refered Symbol	CHADDR1	0A3	000129		CHADDR2	0A6	000129																																															
Byte																																																																																																			
INDF0	POINCO	PODECO	PRINCO	PLUSW0	INDF1	POINC1	POI																																																																																												
00	00	00	00	00	00	00	00																																																																																												
PLUSW1	WREG	BSR	ADCORH	ADCORM	ADCORL	TMAR	P																																																																																												
00	15	01	00	00	00	00	00																																																																																												
Name	Addr	Hex (L->H)	Bin/Refered Symbol																																																																																																
CHADDR1	0A3	000129																																																																																																	
CHADDR2	0A6	000129																																																																																																	

- 快速除錯

- (1) 單步返回 
- (2) 單步執行(進入巨集/副程式) 
- (3) 單步執行(不進入巨集/副程式) 
- (4) 跳出 Call 
- (5) 執行(Free RUN) 
- (6) 暫停 
- (7) 連續執行 
- (8) 程式重置 
- (9) 返回編輯模式 

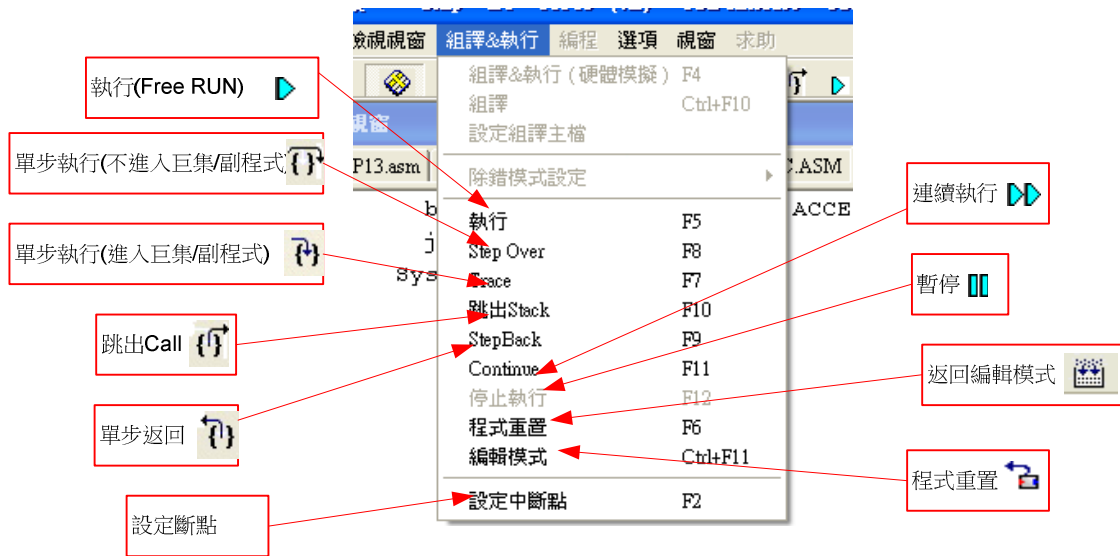


圖 21

- 斷點設定移除 2 種方式

1. 在程式碼視窗或是機械碼視窗中將滑鼠選擇設置斷點處，按鍵盤的“F2”鍵，即可設置或移除斷點
2. 在程式碼視窗或是機械碼視窗中將滑鼠指向設置斷點處，連續點擊滑鼠左鍵，即可設置或移除斷點

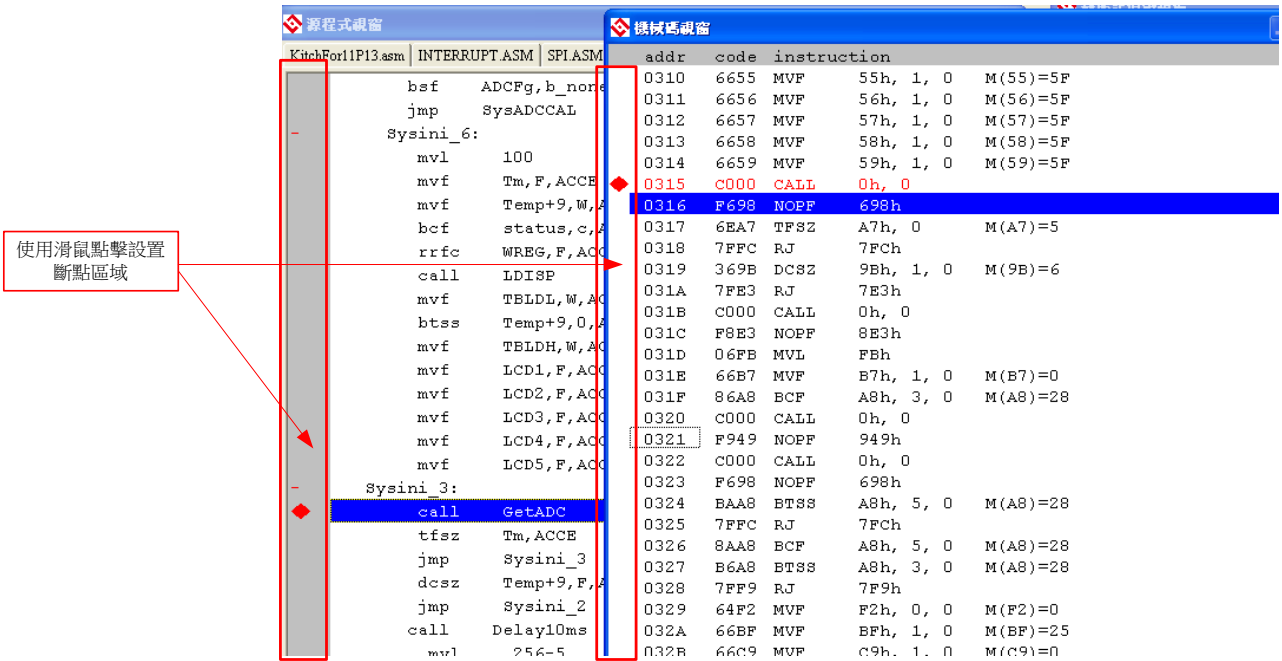
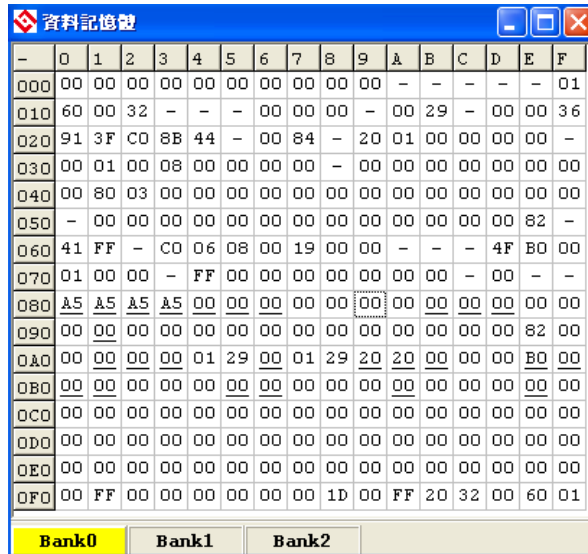


圖 22

### 3.2 RAM視窗



-	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	00	00	00	00	00	00	00	00	00	00	-	-	-	-	-	01
010	60	00	32	-	-	-	00	00	00	-	00	29	-	00	00	36
020	91	3F	C0	8B	44	-	00	84	-	20	01	00	00	00	00	-
030	00	01	00	08	00	00	00	00	-	00	00	00	00	00	00	00
040	00	80	03	00	00	00	00	00	00	00	00	00	00	00	00	00
050	-	00	00	00	00	00	00	00	00	00	00	00	00	00	00	82
060	41	FF	-	C0	06	08	00	19	00	00	-	-	-	4F	B0	00
070	01	00	00	-	FF	00	00	00	00	00	00	00	-	00	-	-
080	<u>A5</u>	<u>A5</u>	<u>A5</u>	<u>A5</u>	00	00	00	00	00	00	00	00	00	00	00	00
090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	82
0A0	00	00	00	00	01	29	00	01	29	20	20	00	00	00	B0	00
0B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0F0	00	FF	00	00	00	00	00	00	1D	00	FF	20	32	00	60	01

圖 23

- 開啟 RAM 視窗後，Bank 會根據所選擇晶片顯示其數量，每一個 Bank 有 256 byte。
  - Bank0 由 0x00 ~ 0xFF，Bank1 由 0x100 ~ 0x1FF...
  - 如果該位址不存在，就會顯示“-”。
  - 如果要切換顯示 Bank 可將滑鼠指向欲顯示的 Bank 區，按下滑鼠確認(滑鼠左鍵)。
  - 如果該位址顯示數字並有下底線，表示已設定 Hint。
  - **注意：Bank0 的 Address 0x00 ~ 0x0E 為間接定址寄存器，無法直接更改，顯示數值是不可參考的，如果要修改請參考 3.3 章節：修改間接定址 Data 或 Address**
- 功能顯示
    - 按下滑鼠選擇鍵(滑鼠右鍵)

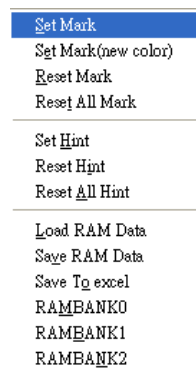


圖 24

- Set Mark
- Set Mark(new color)
- Reset Mark
- Reset All Mark
- Set Hint
- Reset Hint



- Reset All Hint
- Load RAM Data
- Save RAM Data
- Save To excel
- RAMBANK0
- Hint

使用 DS 定義的 SRAM，會在視窗中相對的位址自動產生 Hint，當滑鼠指標指向該位址，就會顯示定義的字串

例如：程式定義 SRAM

```
MEMAR      080h
MD1        DS      1
MD2        DS      1
MD3        DS      1
MDL1       DS      1
MDL2       DS      1
MDL3       DS      1
MD4        DS      5
S_REG      DS      1
r_Len      DS      1
SQRTmp     DS      4
Temp       DS     16
```

組譯後進入除錯狀態，顯示記憶體視窗

當滑鼠指向 80h 的位址，就會出現<80>:MD1

當滑鼠指向 86h 的位址，就會出現<86>:MD4[0]

當滑鼠指向 87h 的位址，就會出現<87>:MD4[1]

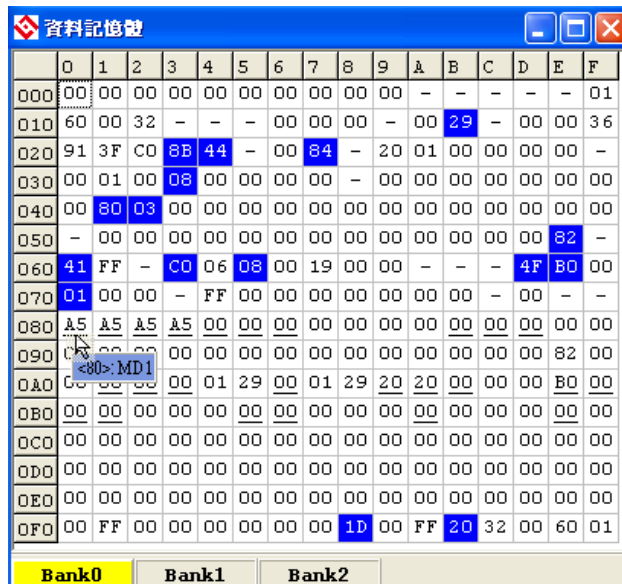


圖 25

- 修改 SRAM 的值有兩種方式
  1. 將滑鼠指向修改的地方，點擊一下滑鼠左鍵，用鍵盤直接 Key IN 。
  2. 將滑鼠指向修改的地方，連續點擊兩下滑鼠左鍵出現 圖 26，使用鍵盤Key In或滑鼠點擊

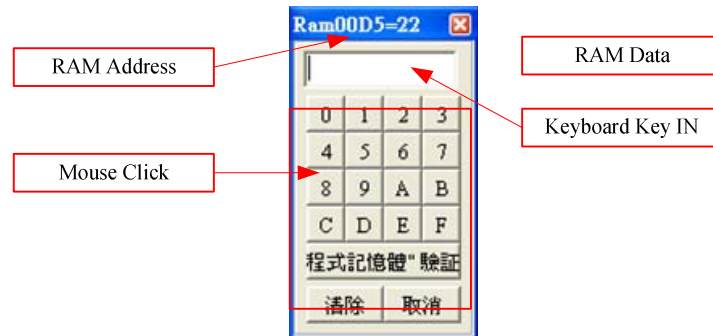


圖 26

**3.3 Register 視窗**

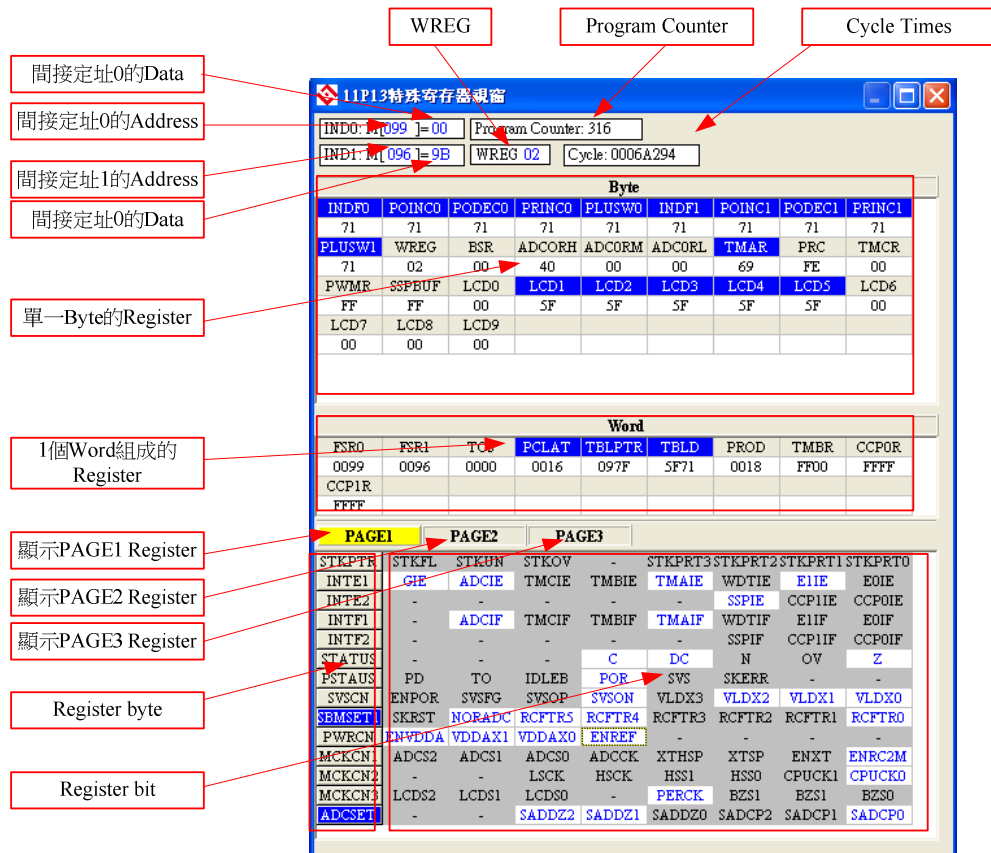


圖 27

- 修改間接定址 Data 或 Address

如圖 28 設定後直接使用鍵盤Key IN或使用滑鼠點選數值及可修改Address

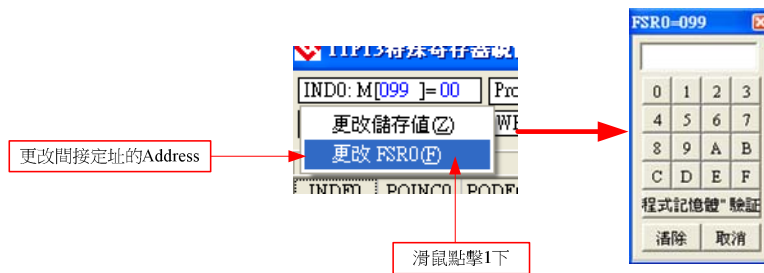


圖 28

如圖 29 設定後直接使用鍵盤Key IN或使用滑鼠點選數值及可修改Data

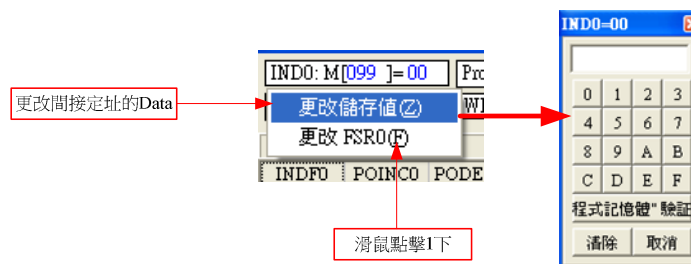


圖 29

- 修改 WREG 的 Data

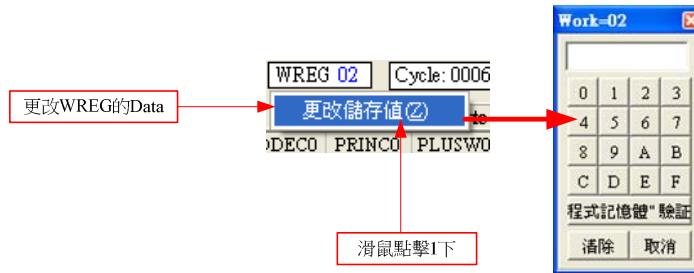


圖 30

- 修改單 1byte 或 Word Register 的 Data

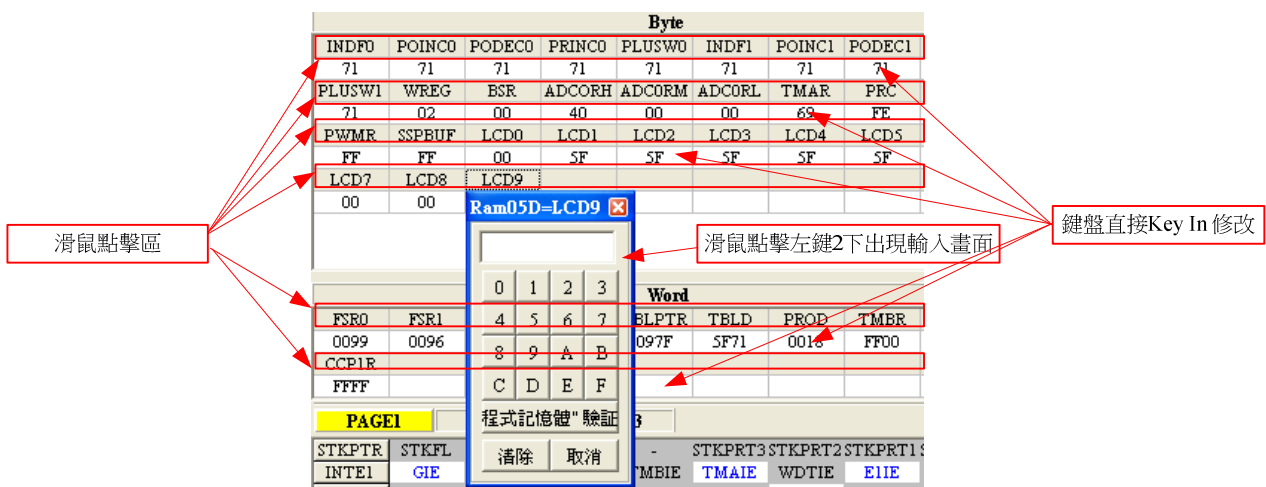


圖 31

- 修改設置 Register 單 1 byte 或單 1 bit
- Bit 設置 1 後，該顯示為反白，藍色字
- Bit 設置 0 後，該顯示為背景色，黑色字

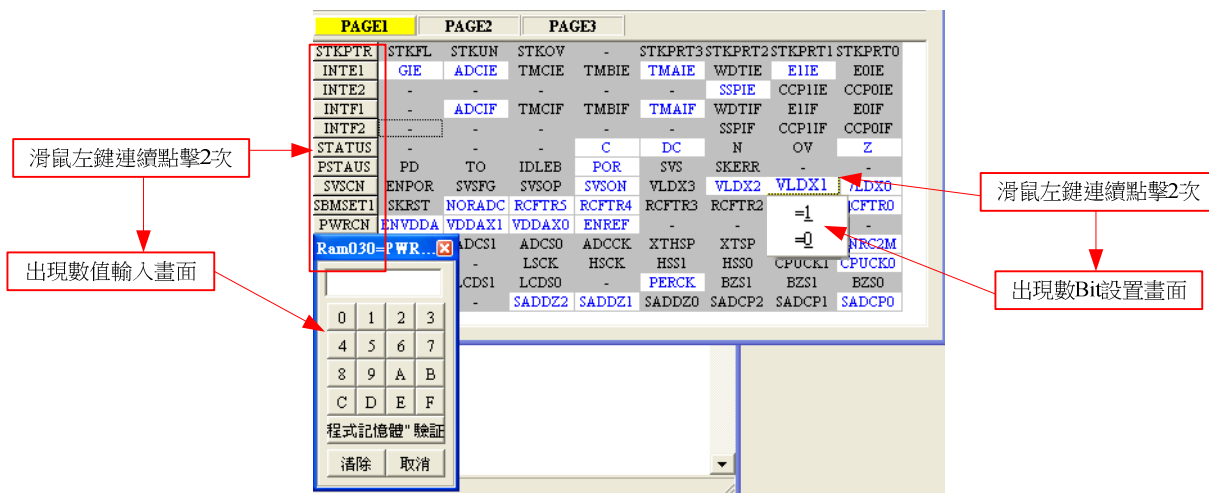


圖 32

### 3.4 Watch 視窗

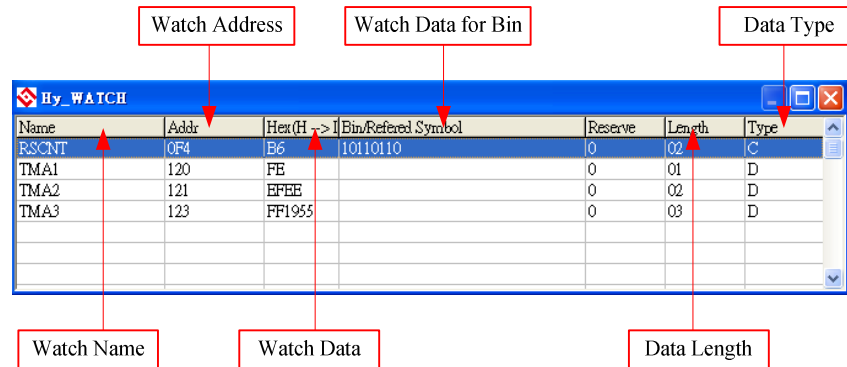


圖 33

- Watch Name → 監看 Data 的名稱，程式使用 EQU 或 DS 定義的名稱
- Watch Address → 監看 Data 的 Address
- Watch Data → 顯示數值，可以選擇由右到左或是由左到右排列，也可顯示十或十六進制顯示

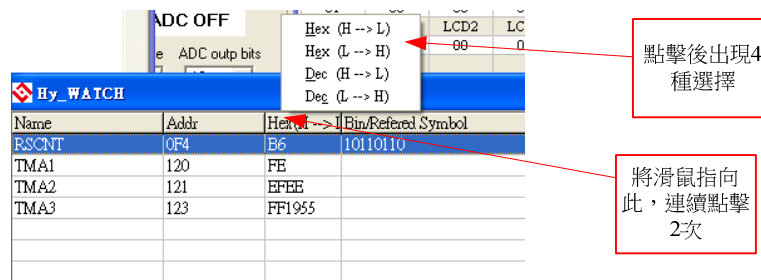


圖 34

Hex (H → L) : 十六進制顯示，位址 H/L 由低至高  
 Hex (L → H) : 十六進制顯示，位址 L/H 由高至低  
 Dec (H → L) : 十進制顯示，位址 H/L 由低至高  
 Dec (L → H) : 十進制顯示，位址 L/H 由高至低

- Watch Data for Bin → Data 以二進制顯示，只有用 EQU 定義的 Address 才有
- Data Length → Data 的長度，顯示 DS 定義的長度；如果用 EQU 定義時，此數值顯示 2
- Data Type → Data 的形式；D = DS 定義; C = EQU 定義



- 監看EQU所定義的Register 或 RAM，，按下滑鼠右鍵選擇要加入監看的Register或RAM如 圖 35

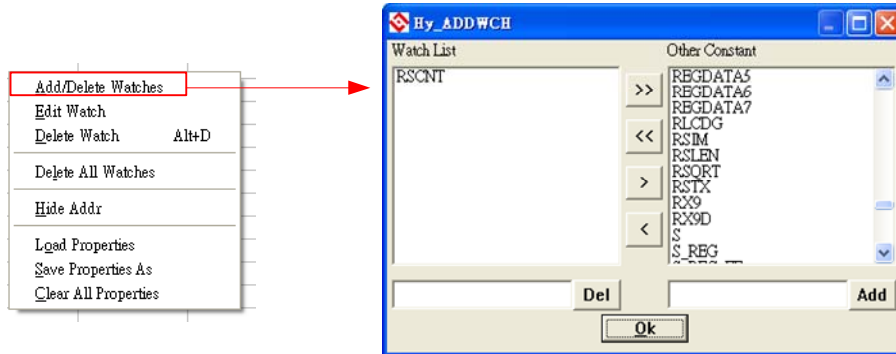


圖 35

### 3.5 堆棧視窗



圖 36



圖 37

### 3.6 ADC視窗

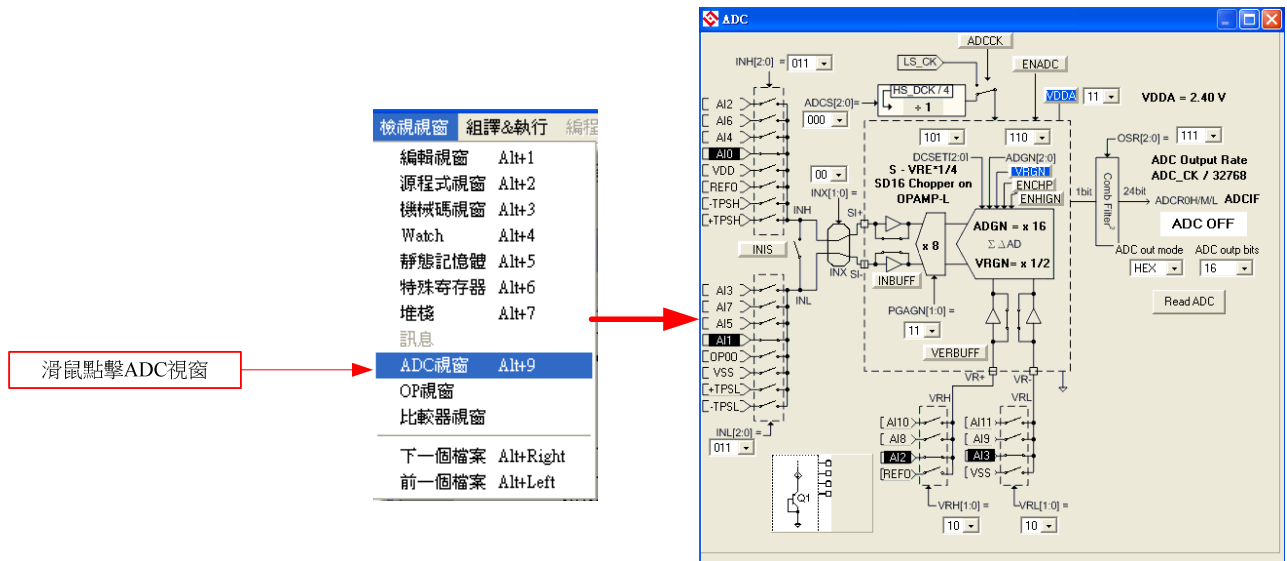


圖 38

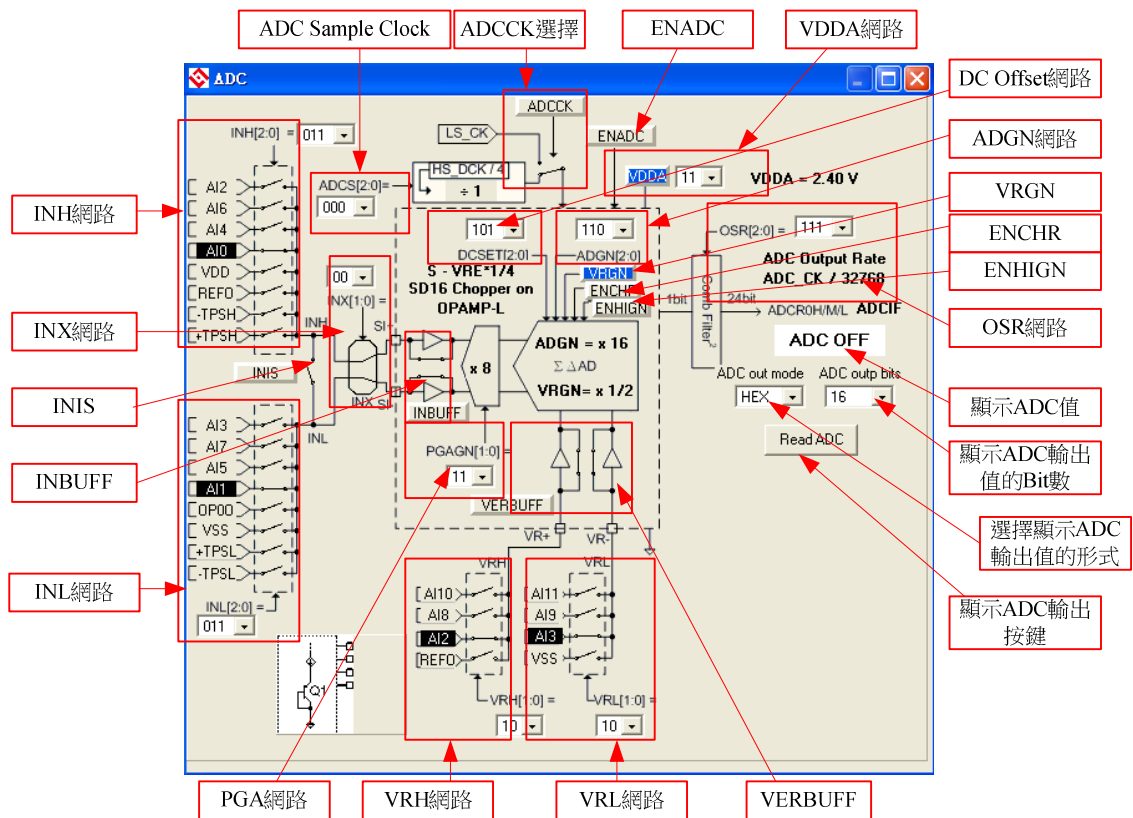


圖 39

- INH 網路

- (1) 以滑鼠左鍵點擊該網路名稱, INH 就可選擇到該網路
- (2) 以滑鼠左鍵點擊該網路開關, INH 就可選擇到該網路
- (3) 滑鼠左鍵點擊, 就可出現 圖 40 畫面, 然後可選擇到開關網路



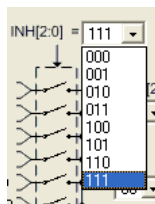


圖 40

- INL 網路

- (1) 以滑鼠左鍵點擊該網路名稱，INL 就可選擇到該網路
- (2) 以滑鼠左鍵點擊該網路開關，INL 就可選擇到該網路
- (3) 滑鼠左鍵點擊，就可出現 圖 41 畫面，然後可選擇到開關網路

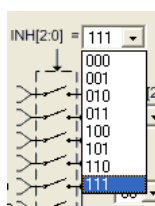


圖 41

- INIS 開關

- (1) 以滑鼠左鍵點擊該網路名稱，INIS 開關就會 ON/OFF
- (2) 以滑鼠左鍵點擊該網路開關，INIS 開關就會 ON/OFF

- INX 網路開關

- (1) 以滑鼠左鍵點擊該網路可切換如 圖 42 ，4 種切換開關

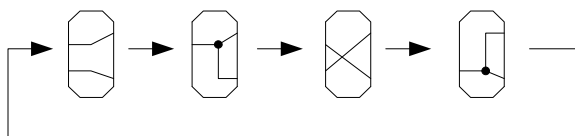


圖 42

- (2) 滑鼠左鍵點擊，就可出現 圖 43 畫面，然後可選擇到開關網路

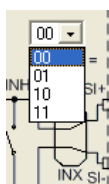


圖 43

- INBUFF 開關

- (1) 以滑鼠左鍵點擊該網路名稱，INBUFF 開關就會 ON/OFF
- (2) 以滑鼠左鍵點擊該網路開關，INBUFF 開關就會 ON/OFF

- INL 網路

- (1) 以滑鼠左鍵點擊該網路名稱，INL 就可選擇到該網路
- (2) 以滑鼠左鍵點擊該網路開關，INL 就可選擇到該網路

(3) 滑鼠左鍵點擊，就可出現 圖 44 畫面，然後可選擇到開關網路

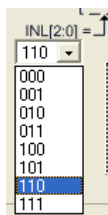


圖 44

- ADC Sample Clock

滑鼠左鍵點擊，就可出現 圖 45 畫面，然後可選擇到開關網路

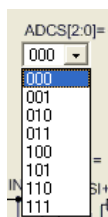


圖 45

- ADCCK 選擇

(1) 以滑鼠左鍵點擊該網路名稱，ADCCK 就會切換選擇

(2) 以滑鼠左鍵點擊該開關，ADCCK 開關就會切換選擇

- ENADC

以滑鼠左鍵點擊該網路名稱，ENADC 就會 ON/OFF，當 ENADC = ON 時顯示 ADC 區才會輸出值

- VDDA 網路

ENVDDA 制能控制

(1) 選擇 ENVDDA 開關 ON/OFF

(2) 選擇 VDDA 電壓

滑鼠左鍵點擊，就可出現 圖 46 畫面，然後可選擇到該模式

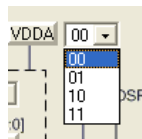


圖 46

(3) 顯示 VDDA 電壓

當 ENVDDA = 0 時，此區域顯示 VDDA = External

當 ENVDDA = 1 時，此區域顯示 VDDX 選擇的電壓

- PGA 網路

滑鼠左鍵點擊，就可出現 圖 47 畫面，然後可選擇到網路

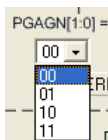


圖 47

- VRH 網路
  - (1) 以滑鼠左鍵點擊該網路名稱，VRH 就可選擇到該網路
  - (2) 以滑鼠左鍵點擊該網路開關，VRH 就可選擇到該網路
  - (3) 滑鼠左鍵點擊，就可出現 圖 48 畫面，然後可選擇到開關網路

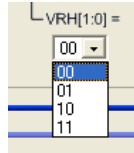


圖 48

- VRL 網路
  - (1) 以滑鼠左鍵點擊該網路名稱，VRL 就可選擇到該網路
  - (2) 以滑鼠左鍵點擊該網路開關，VRL 就可選擇到該網路
  - (3) 滑鼠左鍵點擊，就可出現 圖 49 畫面，然後可選擇到開關網路

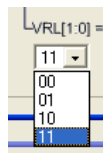


圖 49

- VERBUFF
  - (1) 以滑鼠左鍵點擊該網路名稱，VERBUFF 開關就會 ON/OFF
  - (2) 以滑鼠左鍵點擊該網路開關，VERBUFF 開關就會 ON/OFF

- DC Offset 網路

滑鼠左鍵點擊，就可出現 圖 50 畫面，然後可選擇到網路

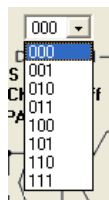


圖 50

- ADGN 網路

滑鼠左鍵點擊，就可出現 圖 51 畫面，然後可選擇到網路

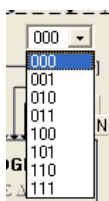


圖 51

- VRGN  
以滑鼠左鍵點擊該網路名稱，VRGN 可選擇到該網路
- OSR 網路  
滑鼠左鍵點擊，就可出現 圖 52 畫面，然後可選擇到網路

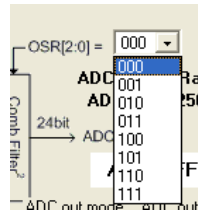


圖 52

- ADC 顯示區
  - (1) 選擇 ADC 值輸出形式 → 可選擇 Hex 或 Dec 輸出
  - (2) 選擇 ADC 值輸出 Bit 數 → 可選擇 8 ~ 24 Bit 輸出
  - (3) 顯示輸出按鍵 → 點擊此鍵可立即顯示 ADC 輸出值

## 3.7 OP視窗

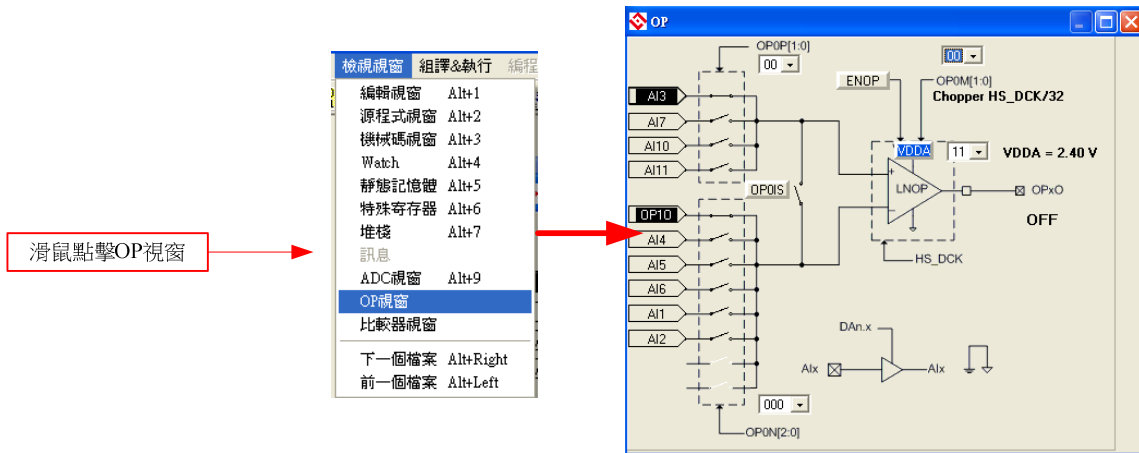


圖 53

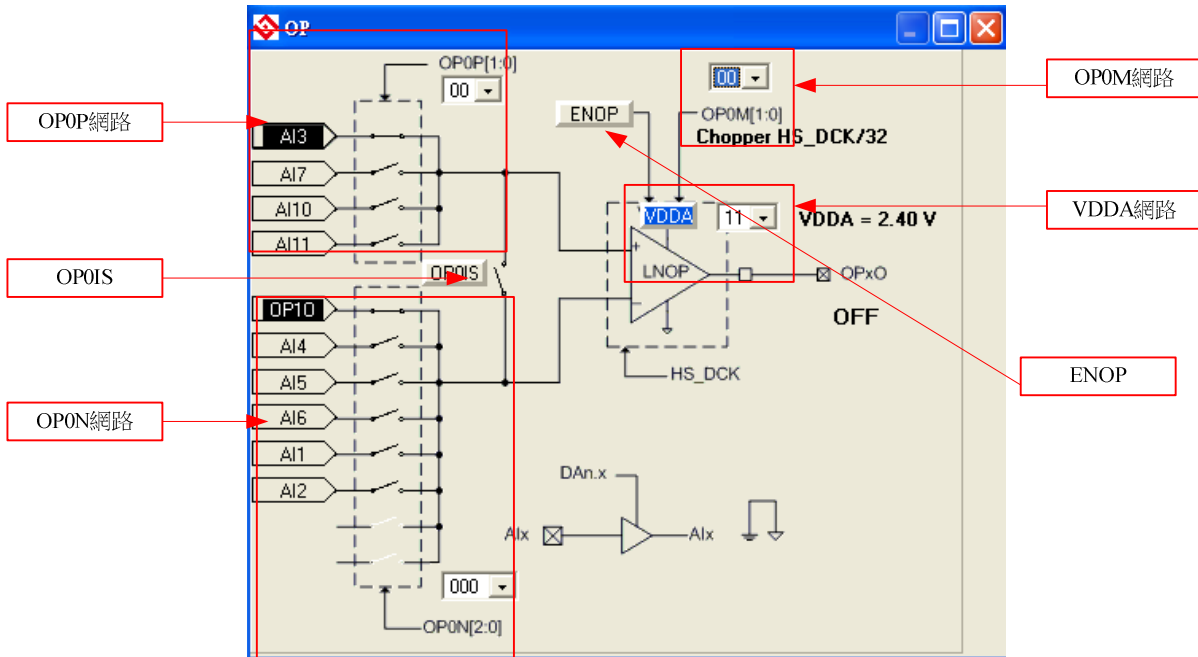


圖 54

- OP0P 網路
  - (1) 以滑鼠左鍵點擊該網路名稱，OP0P 就可選擇到該網路
  - (2) 以滑鼠左鍵點擊該網路開關，OP0P 就可選擇到該網路
  - (3) 滑鼠左鍵點擊，就可出現 圖 55 畫面，然後可選擇到開關網路

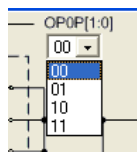


圖 55

- OP0N 網路
  - (1) 以滑鼠左鍵點擊該網路名稱，OP0N 就可選擇到該網路
  - (2) 以滑鼠左鍵點擊該網路開關，OP0N 就可選擇到該網路
  - (3) 滑鼠左鍵點擊，就可出 圖 56 現畫面，然後可選擇到開關網路

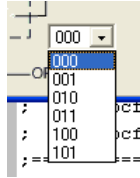


圖 56

- ENOP
  - 以滑鼠左鍵點擊該網路名稱，ENOP 就會 ON/OFF, ENOP 狀態顯示
  - 當 ENOP = 1 時，顯示 ON
  - 當 ENOP = 0 時，顯示 OFF
- VDDA 網路 (參考 ADC 視窗的 VDDA 網路)
- OP0M 網路

滑鼠左鍵點擊，就可出 圖 57 現畫面，然後可選擇到開關網路

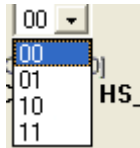


圖 57

### 3.8 比較器視窗

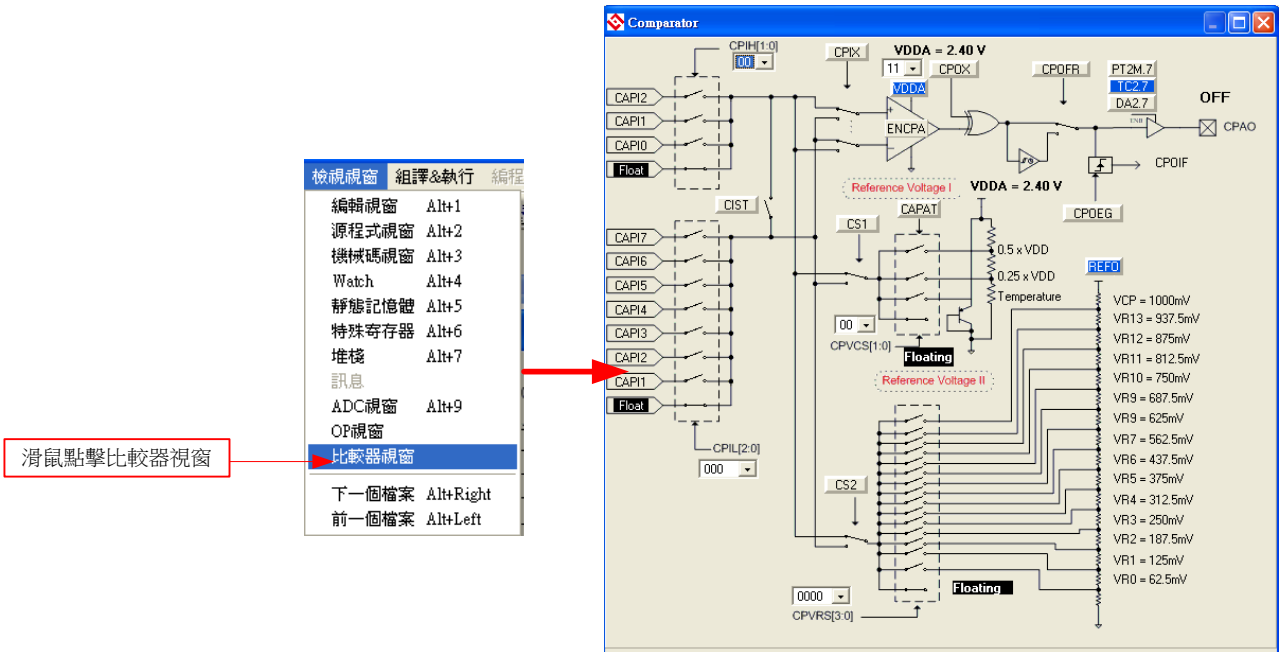


圖 58

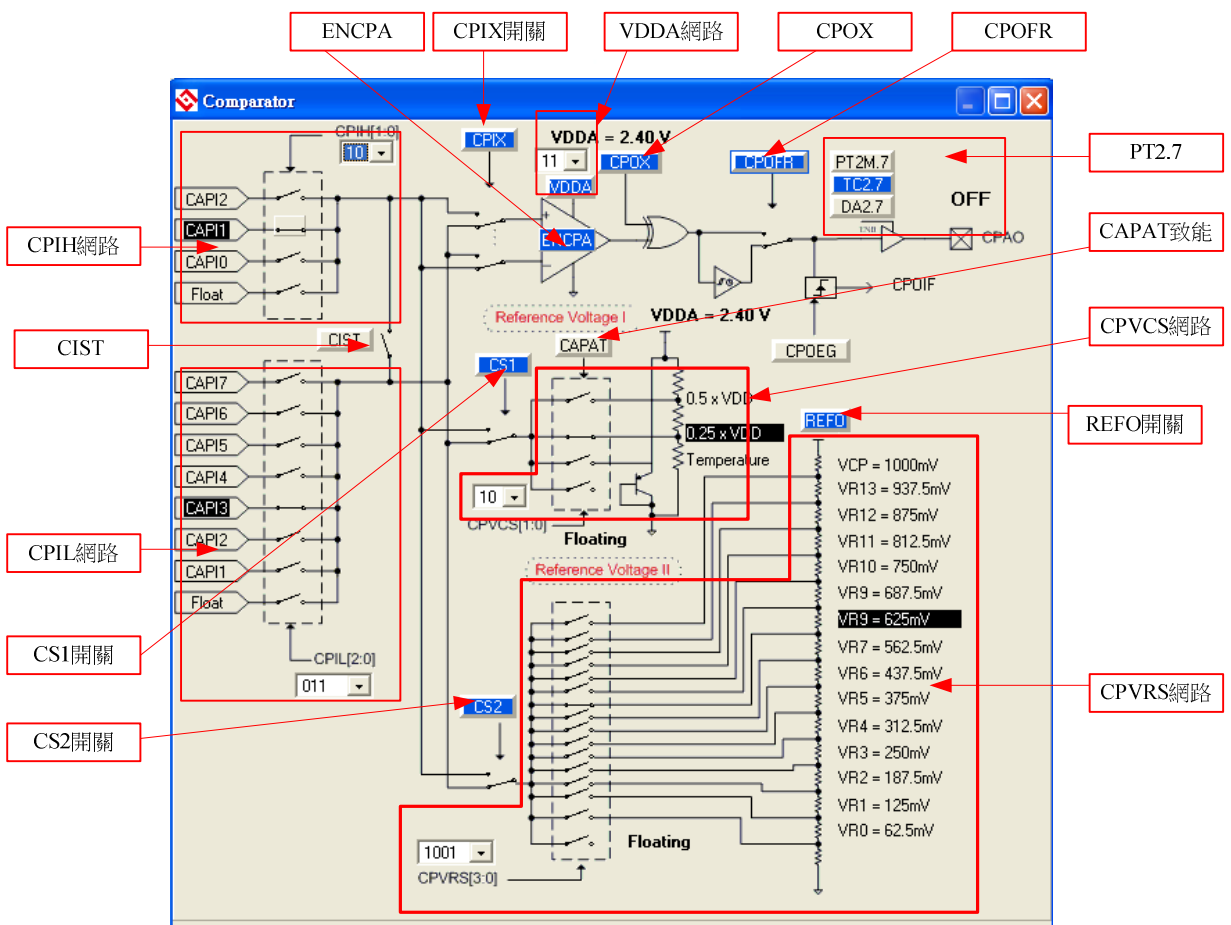


圖 59

- CPIH 網路

- (1) 以滑鼠左鍵點擊該網路名稱，CPIH 就可選擇到該網路
- (2) 以滑鼠左鍵點擊該網路開關，CPIH 就可選擇到該網路
- (3) 滑鼠左鍵點擊，就可出現 圖 60 畫面，然後可選擇到開關網路

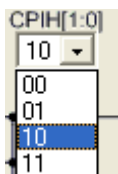


圖 60

- CPIL 網路

- (1) 以滑鼠左鍵點擊該網路名稱，CPIL 就可選擇到該網路
- (2) 以滑鼠左鍵點擊該網路開關，CPIL 就可選擇到該網路
- (3) 滑鼠左鍵點擊，就可出現 圖 61 畫面，然後可選擇到開關網路

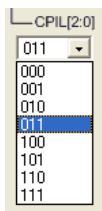


圖 61

- CPVCS 網路

- (1) 以滑鼠左鍵點擊該網路名稱，CPVCS 就可選擇到該網路
- (2) 以滑鼠左鍵點擊該網路開關，CPVCS 就可選擇到該網路
- (3) 滑鼠左鍵點擊，就可出現 圖 62 畫面，然後可選擇到開關網路

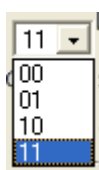


圖 62

- CPVRS 網路

- (1) 以滑鼠左鍵點擊該網路名稱，CPVRS 就可選擇到該網路
- (2) 以滑鼠左鍵點擊該網路開關，CPVRS 就可選擇到該網路
- (3) 滑鼠左鍵點擊，就可出現 圖 63 畫面，然後可選擇到開關網路

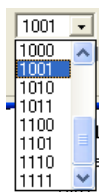


圖 63



- VDDA 網路 (參考 ADC 視窗的 VDDA 網路)
- CPIX 開關  
以滑鼠左鍵點擊該網路名稱，CPIX 就會切換
- CIST  
以滑鼠左鍵點擊該網路名稱，CIST 就會 ON/OFF  
CIST 狀態顯示  
當 CIST = 1 時，顯示 ON  
當 CIST = 0 時，顯示 OFF
- CS1 開關  
以滑鼠左鍵點擊該網路名稱，CS1 就會上下切換
- CS2 開關  
以滑鼠左鍵點擊該網路名稱，CS2 就會上下切換
- ENCPA  
以滑鼠左鍵點擊該網路名稱，ENCPA 啟動  
ENCPA 狀態顯示  
當 ENCPA = 1 時，比較器啟動  
當 ENCPA = 0 時，比較器關閉
- CPOX  
以滑鼠左鍵點擊該網路名稱，CPOX 就會 ON/OFF
- CPOFR  
以滑鼠左鍵點擊該網路名稱，CPOFR 就會上下切換
- CAPAT 置能  
以滑鼠左鍵點擊該網路名稱，CAPAT 模組就會啟動或關閉  
CAPAT 狀態顯示  
當 CAPAT = 1 時，CAPAT 模組啟動  
當 CAPAT = 0 時，CAPAT 模組關閉
- REFO 開關  
以滑鼠左鍵點擊該網路名稱，REFO 就會 ON/OFF  
REFO 狀態顯示  
當 REFO = 1 時，REFO 啟動  
當 REFO = 0 時，REFO 關閉
- PT2.7  
當 PT2M.7 = 1 且 TC2.7 = 1 且 DA2.7 = 1 時 CAPO 才會有輸出。

### 3.9 暫存器修改記錄

進入模擬視窗後(軟體模擬或硬體模擬)，凡是暫存器或 SRAM 經過手動修改過(無論經由任何視窗修改 RAM、Register、ADC、OP 及 CMP)，就會被記錄起來，當按下”SRAM 修改記錄”後就會顯示出來，此時視窗會停駐在此畫面中直到將此畫面關閉才能繼續執行任何動作。

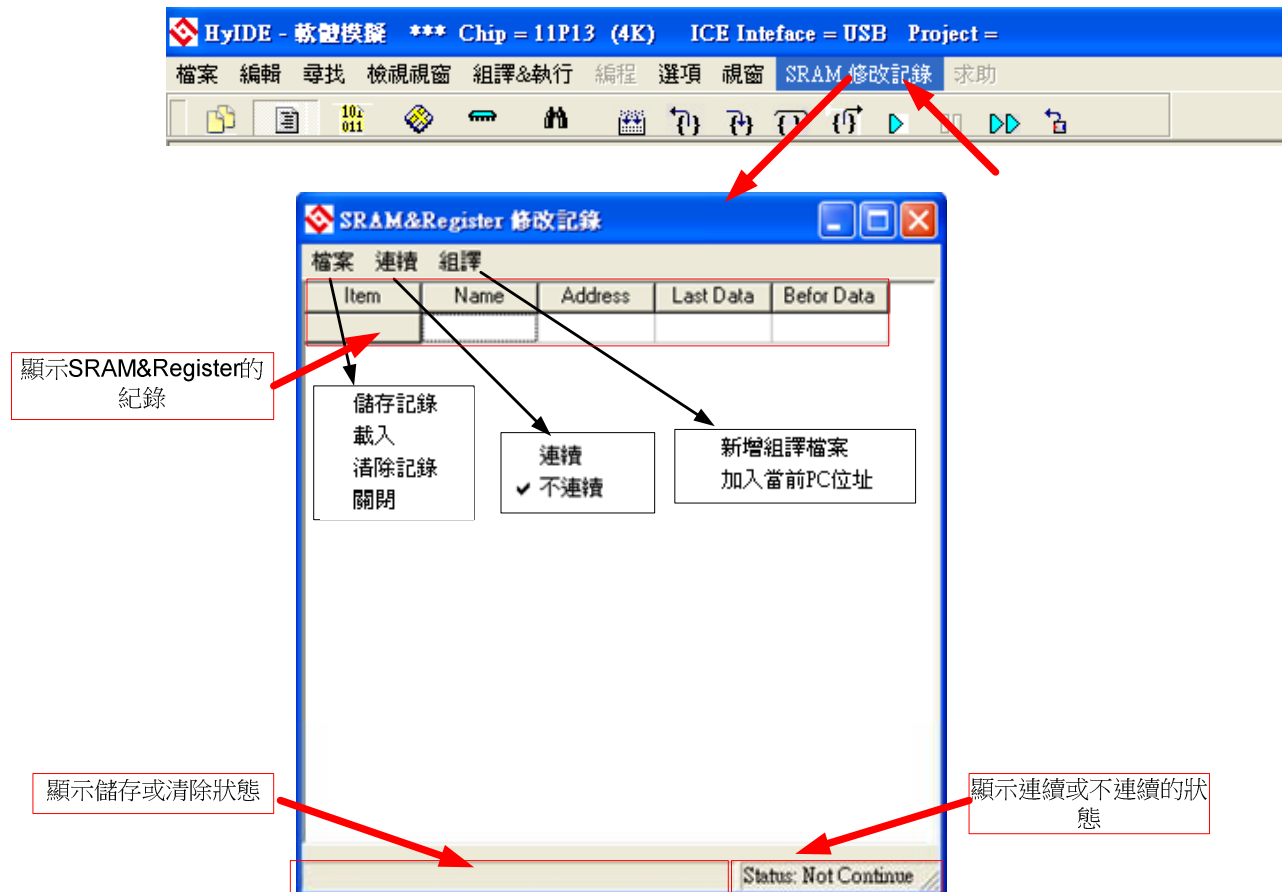


圖 64

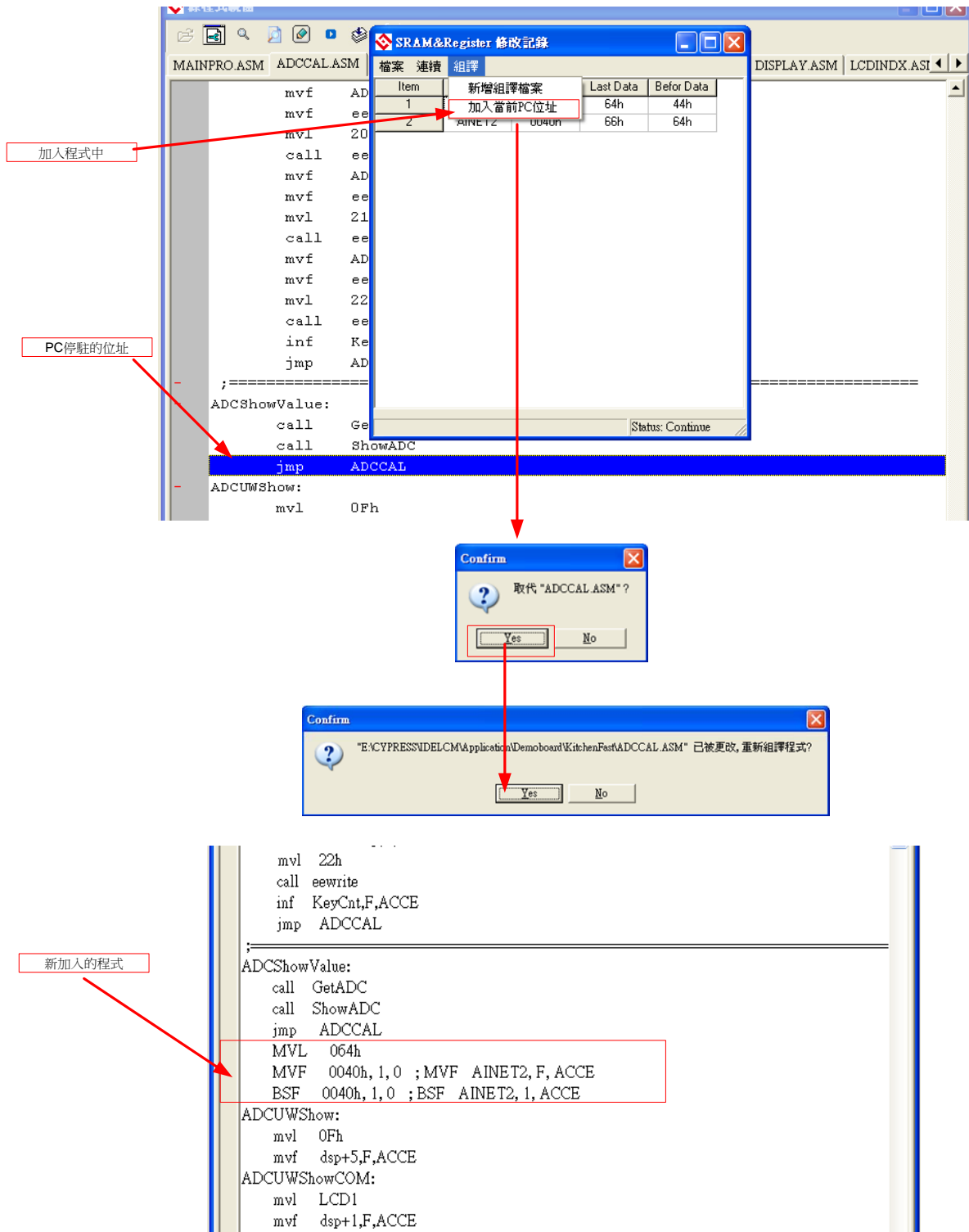


圖 65

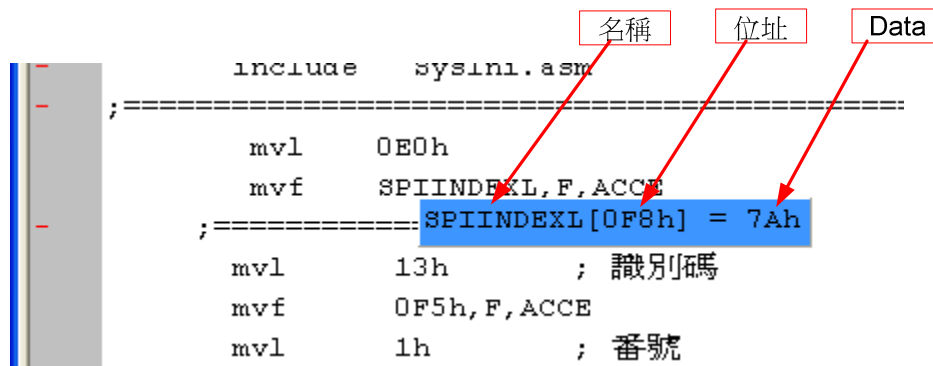
### 3.10 源程式視窗下的Hint功能

在源程式(Source Code)視窗下，要知道 Register 或 SRAM 的值及 Address，可以將滑鼠指向此 Register 或 SRAM 的名稱，就可顯示名稱、位址及 Data。

只有在以下指令後面所帶的參數下才有此功能：

CLRF, ADDF, INF, INSZ, DCF, DCSZ, SUBF, COMF, ADDC, ANDF, IORF, XORF, SUBC, RRF, SETF, MULF, RLF, JZ, RRFC, RLFC, SWPF, DAW, INSUZ, DCSUZ, ARLC, ARRC, CPSG, CPSL, CPSE, TFSZ, BTFG BSF, BCF, BTSS, BTSZ, MVFF(不是 Macro)。

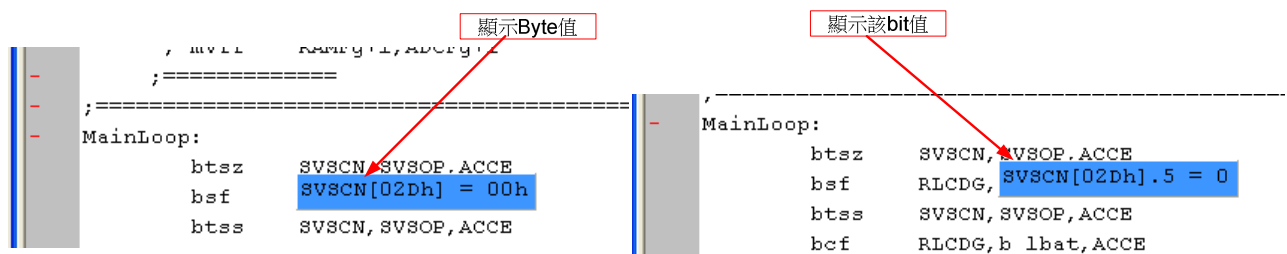
- 當指令為位元操作時只有第一個參數才有效，如 圖 66
- 當指令為BCF、BSF、BTSS、BTSZ及BTGF時，當指向第一個參數顯示Byte數值，當指向第二個參數顯示該Bit的值(1 or 0)，如 圖 67
- 當指令為MVFF時(不是Macro)，當指向第一個參數顯示第一個參數數值，當指向第二個參數顯示第二個參數數值，如 圖 68
- 如果參數為INDF0、POINC0、PODEC0、PRINC0、INDF1、POINC1、PODEC1、PRINC1 時，此時Data為FSR0 或FSR1 內的位址的Data，如 圖 69
- 如果參數為PLUSW0 或PLUSW1 時，此時Data為FSR0+WREG或FSR1+WREG內的位址的Data，如 圖 70



```

include sysini.asm
;=====
mvl    0E0h
mvf    SPIINDEXL, F, ACCE
;===== SPIINDEXL[0F8h] = 7Ah
mvl    13h      ; 識別碼
mvf    0F5h, F, ACCE
mvl    1h      ; 番號
    
```

圖 66

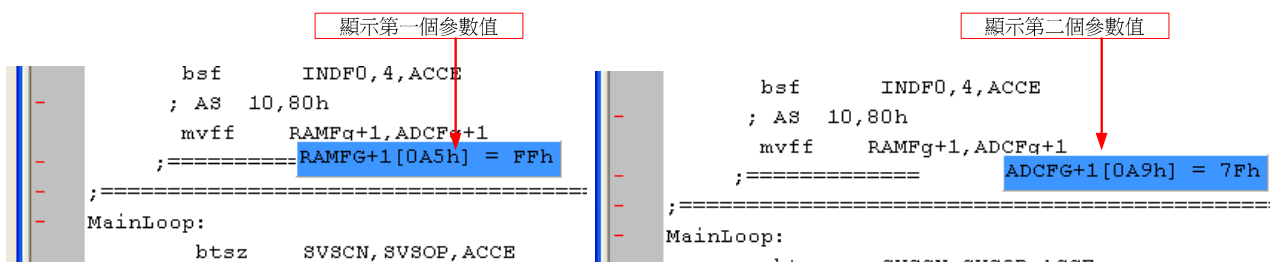


```

;=====
MainLoop:
btsz   SVSCN, SVSOP, ACCE
bsf    SVSCN[02Dh] = 00h
btss   SVSCN, SVSOP, ACCE

MainLoop:
btsz   SVSCN, SVSOP, ACCE
bsf    RLCDG, SVSCN[02Dh].5 = 0
btss   SVSCN, SVSOP, ACCE
bcf    RLCDG, b_lbat, ACCE
    
```

圖 67



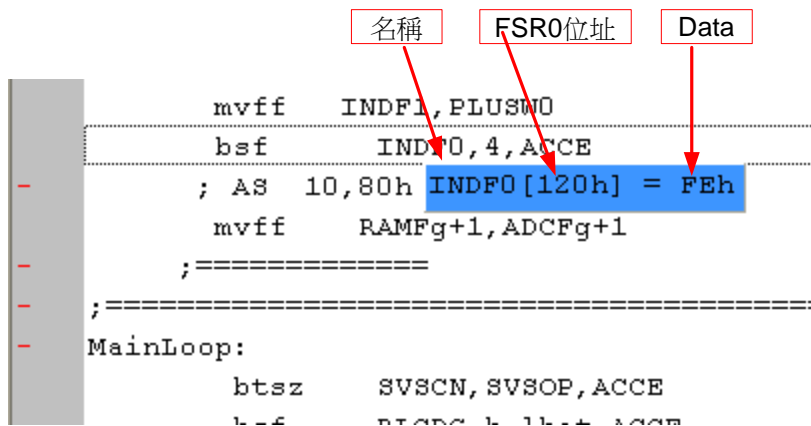
```

    bsf     INDF0,4,ACCE
    ; AS 10,80h
    mvff   RAMFG+1,ADCFG+1
    ;===== RAMFG+1[0A5h] = FFh
    ;=====
MainLoop:
    btsz   SVSCN,SVSOP,ACCE
  
```

```

    bsf     INDF0,4,ACCE
    ; AS 10,80h
    mvff   RAMFG+1,ADCFG+1
    ;===== ADCFG+1[0A9h] = 7Fh
    ;=====
MainLoop:
  
```

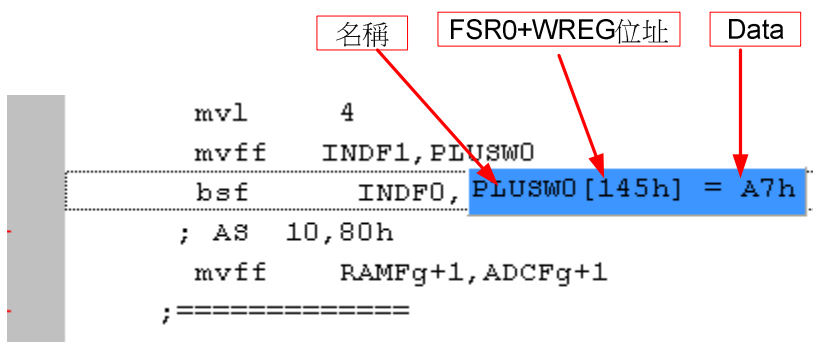
圖 68



```

    mvff   INDF1,PLUSW0
    bsf     INDF0,4,ACCE
    ; AS 10,80h INDF0 [120h] = FEh
    mvff   RAMFG+1,ADCFG+1
    ;=====
    ;=====
MainLoop:
    btsz   SVSCN,SVSOP,ACCE
    bsf     INDF0,4,ACCE
  
```

圖 69



```

    mvl     4
    mvff   INDF1,PLUSW0
    bsf     INDF0,PLUSW0 [145h] = A7h
    ; AS 10,80h
    mvff   RAMFG+1,ADCFG+1
    ;=====
  
```

圖 70

## 4. 燒錄視窗

### 4.1 介面設定

要進入燒錄式窗畫面，點選”選項”，出現選擇畫面，點選”介面設定”，如圖 71 所示。

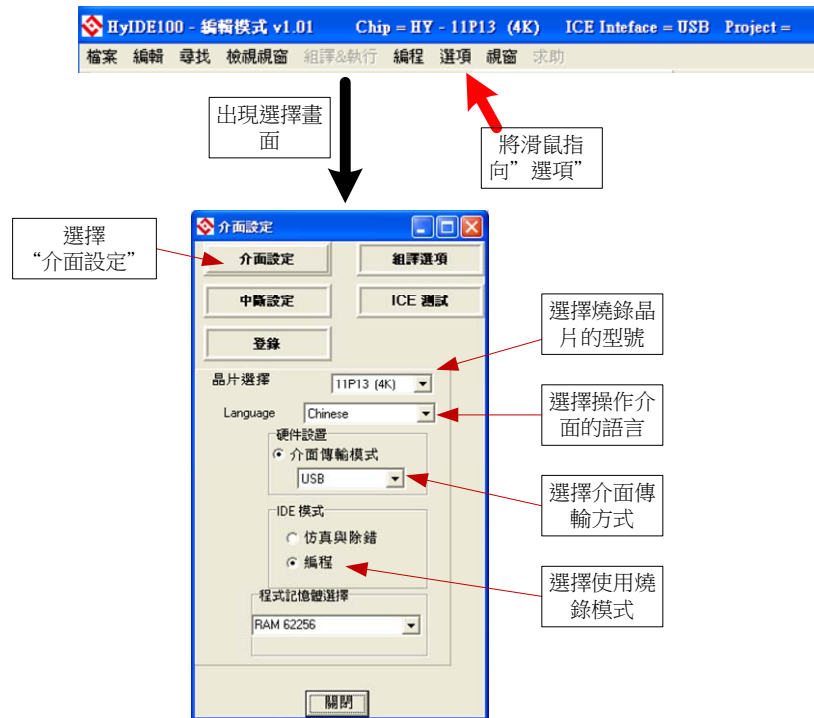


圖 71

晶片選擇 → 選擇燒錄晶片的型號

如果燒錄晶片與選擇的型號不同，則在 Blank Check、Program、Verify，都會失敗。

Language → 選擇操作介面的語言，中文或英文。硬體設置 → 可選擇 USB 介面或 Parallel Port 介面。

IDE 模式 → 選擇編程。

當介面設定完成後點選”組譯選項”選擇燒錄的設定，如圖 72。



圖 72

- 組譯產生附檔 → 選擇組譯程式後所產生的檔案。
- 堆棧操作 → 選擇當 OTP 程序運行後如果發生堆棧滿或溢位是否要重置。
- 未使用區域填滿 → 組譯程式後，在未使用的程式空間選擇填滿 00 或 FF。
- 組譯精簡 → 選擇是否要精簡組譯。
- Enable Program Times → 選擇是否啟動 Download 的程式能被燒錄的次數。
- Input Program Times → 填入 Download 的程式能被燒錄的次數(最大 2147483646，最小 1)。

當組譯選項完成後點選”ICE測試”進入測試電壓是否正常(在進入”選項”之前須先連接IDE，並插入 9V 電源)，如圖 73。

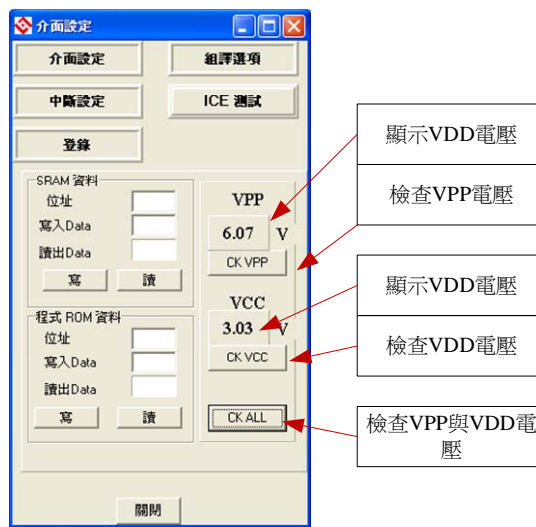


圖 73

當介面設定完成後點選”關閉”，會將所設定的參數記錄起來，下次開啟此設定，會自動載入設定值，並在標題視窗顯示設定燒錄晶片型號，如圖 74。

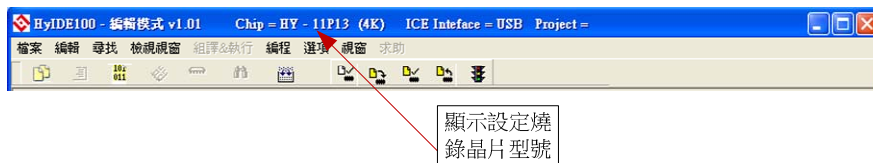


圖 74

- 燒錄時 VPP 的電壓  $5.6 < VPP < 6.6$
- 燒錄時 VDD 的電壓  $2.7 < VDD < 3.6$

## 4.2 操作步驟

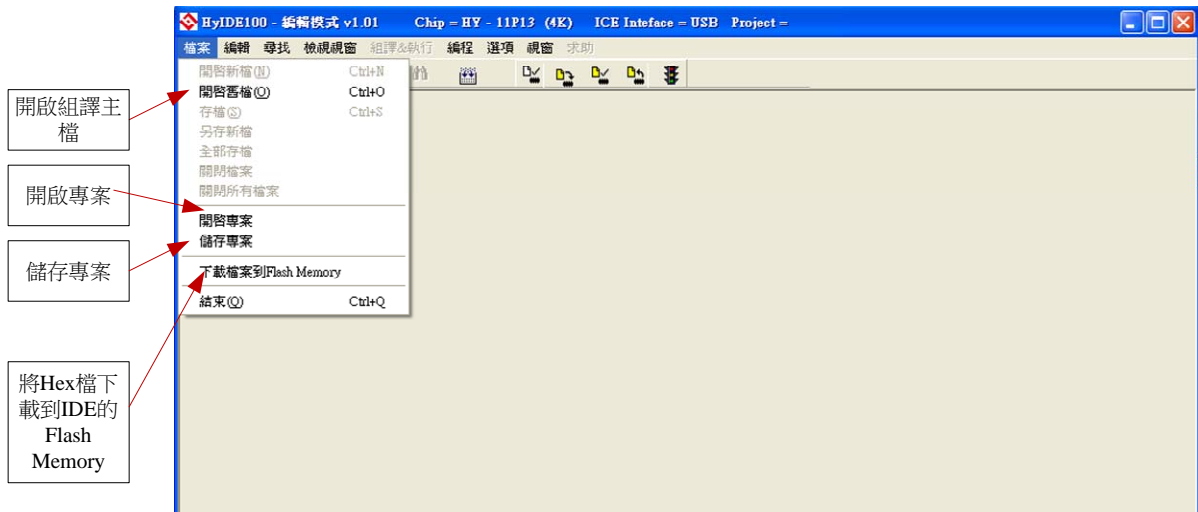


圖 75

開啟舊檔 → 開啟已經寫好的源程式組譯主檔。

開啟專案 → 開啟儲存的專案名稱。

儲存專案 → 儲存已完成的專案。

下載檔案到 Flash Memory → 將組譯完成後所產生的 Hex 檔案，載入燒錄器或 IDE 的 Flash Memory。

### 4.2.1 開啟檔案與組譯

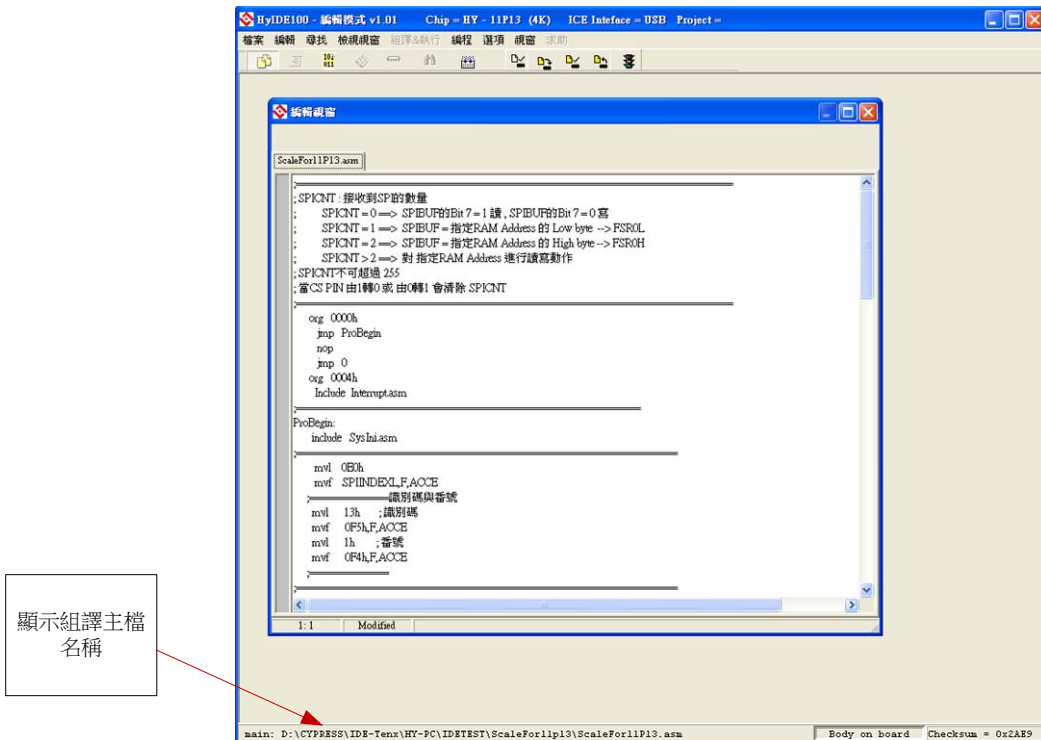


圖 76

由開啟檔案將源程式的主檔名稱開啟，並在顯示組譯主檔名稱下顯示，如果顯示名稱與主檔名稱不同，將鼠標指向檔案，按下滑鼠右鍵，選擇設為組譯主檔，如圖 77。



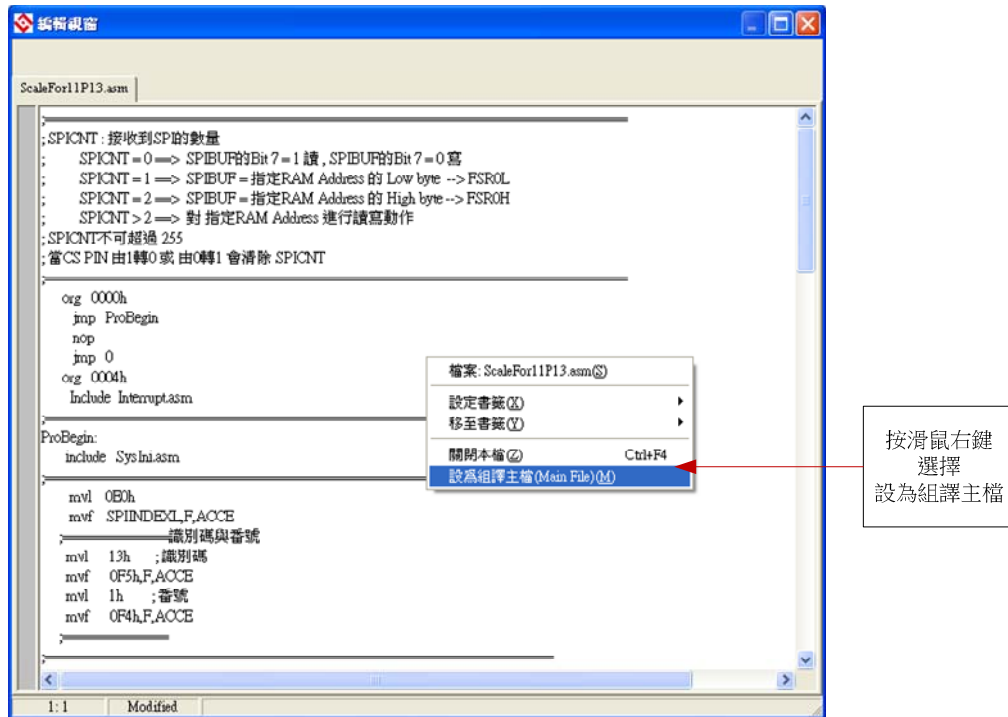


圖 77

將Source Code 組譯並Download到燒錄器或IDE的Flash Memory，如圖 78

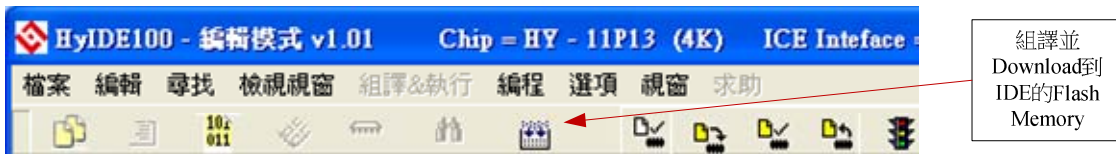


圖 78

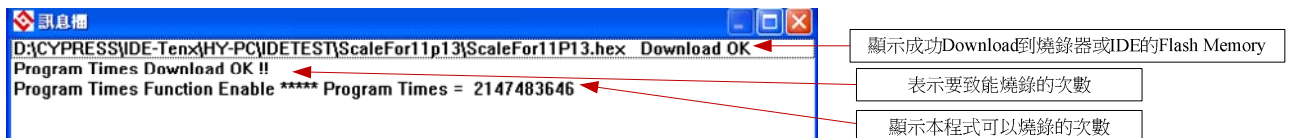


圖 79

1. 當介面選擇 USB，組譯主程式完成後會將程式碼，載入燒錄器或 IDE 的 Flash Memory 內，作為生產線上量產燒錄用。
2. 如果組譯選項內有選擇致能燒錄次數，訊息欄位會顯示程式可以燒錄次數，如圖 79。
3. 當組譯完成後在下方顯示組譯完成後的Hex檔名稱與Checksum，如圖 80。



圖 80

### 4.2.2 Download HEX File

要燒錄一個已經組譯完成的Hex檔案，“檔案”→“下載檔案到Flash Memory”，選擇要燒錄的Hex 檔案，最後一步的選項會依照當初組譯選擇是否保護而有所改變，如果當初選擇保護，就不會詢問是否要保護，若未選擇保護就會詢問，如圖 81。



圖 81

當載入成功後訊息欄顯示成功訊息，如圖 79，並在顯示欄位上顯示Download的Hex檔案及Checksum，如圖 80。

## 4.3 PC連線燒錄OTP

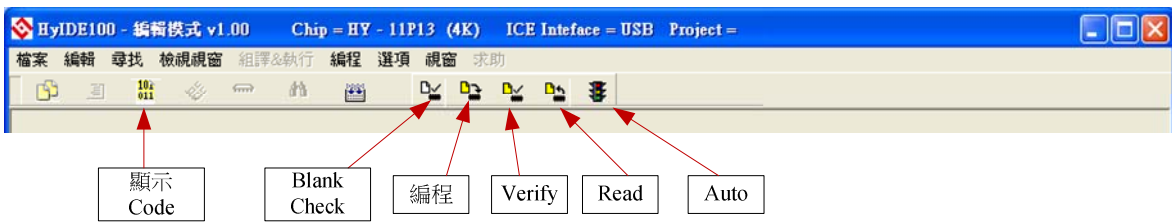


圖 82

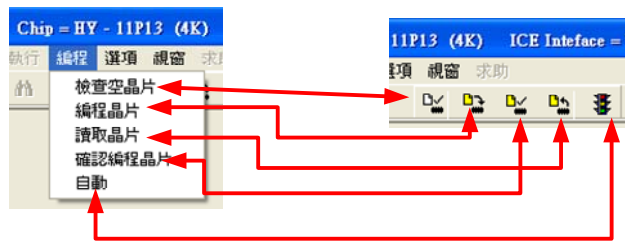


圖 83

當燒錄的檔案成功的載入燒錄器或 IDE 的 Flash Memory 內，將可以進行 Blank Check、燒錄、Verify 及讀取等動作，如果沒有成功載入，則以上的動作將不會成功。

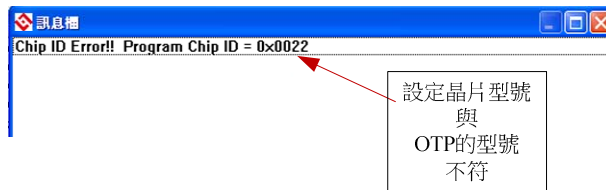


圖 84



圖 85



圖 86

確定在標題視窗下所選擇的燒錄晶片型號(圖 74)，與OTP型號相同，當燒錄器執行Blank Check、燒錄與Verify，程序會比對設定選擇晶片型號與燒錄OTP型號是否相同，如果不同否則不會燒錄到OTP內，在訊息欄內顯示錯誤訊息如圖 84。

在燒錄之前如果想要確定型號是否正確，可以將滑鼠指標指向”晶片連線狀態顯示區”上按滑鼠左鍵，如果晶片型號正確則顯示如圖 85；如果不正確則顯示如圖 86；如果有勾選”Enable Program Times”則剩餘燒錄次數會顯示於訊息欄內如圖 87。



圖 87

### 4.3.1 晶片檢查(Blank Check)


晶片檢查(Blank Check) 圖示為 ，在還沒有燒錄過的晶片，讀取其內部的Code應該皆為 0xFFFF，晶片檢查的目的是確定此OTP所有位址的內容皆為 0xFFFF。檢查晶片是否為空所指的是要燒錄OTP 位址的內容皆為 0xFFFF。如果選擇晶片正確以及檢查為空，訊息欄出現以下訊息(圖 88)。




圖 88

如果選擇晶片不正確或是檢查不為空，訊息欄出現以下訊息(圖 89)。



圖 89

### 4.3.2 編程晶片(Program)

編程晶片(Program)圖示為 ，編程的目的是將已經 Compiler 完成的程序燒錄到 OTP 的晶片中，燒錄完成後組裝成品後，將可依照使用者所寫的指令運行程序。

將已下載或組譯完成的Hex檔(顯示於最下面的顯示欄)，編程於選擇晶片內，並確認編程晶片內容是否正確(步驟參考 4.2.1 或 4.2.2 一節)。

如果選擇晶片正確以及編程成功，訊息欄出現以下訊息(圖 90)，如果有勾選 "Enable Program Times" 則允許燒錄的次數會減 1，並將剩餘燒錄次數顯示於訊息欄內。

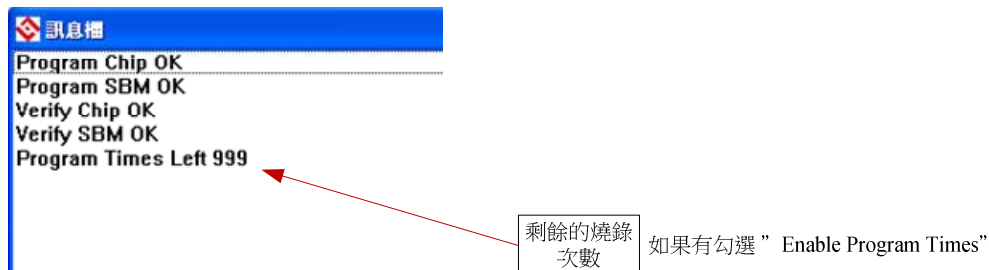



圖 90

如果選擇晶片不正確以及編程不成功，訊息欄出現以下訊息(圖 91)。



圖 91

### 4.3.3 確認編程晶片 (Verify)

確認編程晶片 (Verify) 圖示為 ，確認編程晶片的目的是在比對燒錄到 OTP 晶片的程式是否與載入到燒錄器的程式相同。

確認編程晶片內容是否與下載或組譯完成的 Hex 檔(顯示於最下面的顯示欄)一致，如果晶片已經編程保護，則此項無效或比對失敗。

如果選擇晶片正確以及確認編程成功，訊息欄出現以下訊息(圖 92)。




圖 92

如果選擇晶片不正確以或確認編程不成功，訊息欄出現以下訊息(圖 93)。



圖 93

### 4.3.4 讀取晶片 (Read)

讀取晶片 (Read) 圖示為 ，讀取晶片的目的是，是讓使用者確認讀取 OTP 的 Checksum 是否與燒錄的 Hex 檔相同。讀取晶片內容(步驟如 圖 94)，並將內容顯示於“顯示Code”視窗內。

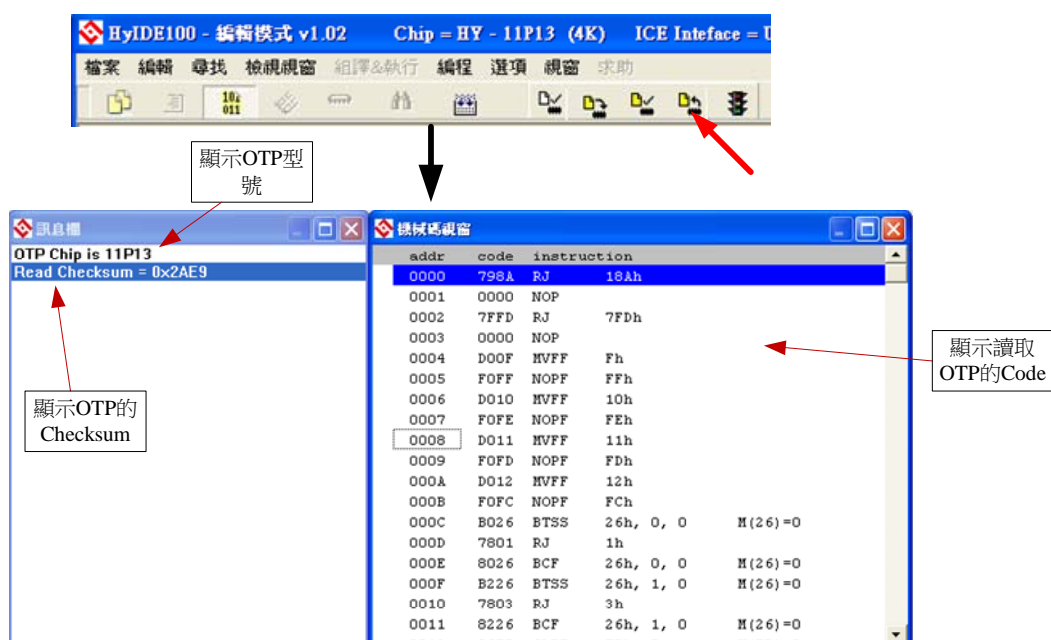



圖 94

### 4.3.5 AUTO

AUTO 圖示為 ，Auto 是綜合 Blank Check、Program 及 Verify 三項功能，選擇 Auto 會先檢查晶片是否為空，然後編程，確認編程晶片。

當執行成功後，訊息欄出現以下訊息(圖 95)，如果有勾選 "Enable Program Times" 則允許燒錄的次數會減 1，並將剩餘燒錄次數顯示於訊息欄內。



圖 95

如果有一項失敗，整個過程會立即停止，並在訊息欄顯示錯誤訊息。

### 4.4 離線燒錄

#### 4.4.1 燒錄說明

當用戶程序由開發階段進入工程試產階段時，此時可以單獨使用燒錄器，無須連線 PC。

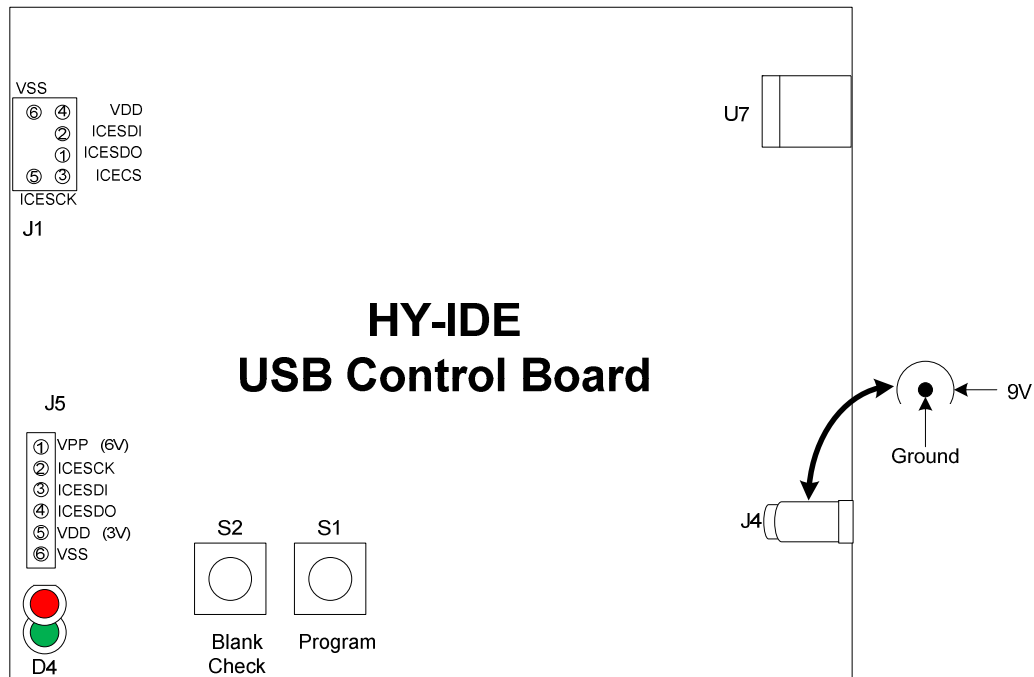


圖 96

- ◆ J4 : Adapter 9V 輸入  
內負外正，供應燒錄電壓源(燒錄 OTP 時需要接上)
- ◆ U7 : USB 連接座與 PC 端連接  
下載程序供仿真除錯使用  
下載燒錄程序供 HY11P 系列產品別燒錄使用
- ◆ J5 : HY11P 系列燒錄控制端口  
PIN 1 VPP(6V) 連接晶片的 VPP  
PIN 2 ICECK 連接晶片的 PSCK  
PIN 3 ICESDI 連接晶片的 PSDI  
PIN 4 ICESDO 連接晶片的 PSDO  
PIN 5 VDD(3V) 連接晶片的 VDD  
PIN 6 VSS 連接晶片的 VSS
- ◆ S1 : Program, 晶片燒錄按鍵
- ◆ S2 : Blank Check, 晶片空白檢查按鍵
- ◆ D4 : 雙色 LED ;  
紅色 LED : OTP 燒錄、Blank Check... 執行錯誤顯示燈號  
綠色 LED : OTP 燒錄、Blank Check... 執行成功顯示燈號  
綠色 LED : USB 或 Adapter 上電時顯示



下圖 97 為PC在線時，程序Download，在線燒錄的晶片與控制板的燒錄接腳連接方法：



圖 97

下圖 98 為PC離線時已經將程序Download完成，離線燒錄晶片與控制板的燒錄接腳連接方法：



圖 98

- 離線操作時需要先將Hex檔Download到燒錄器的Flash Memory內，步驟參考 4.2.1 或 4.2.2 一節。
- 離線燒錄時，先按按鍵 S2 可檢查晶片是否為空，檢查完後應為 D4 綠色 LED 亮。
- 按鍵 S1 為燒錄按鍵，其步驟為 Blank Check → Program → Verify，如果在 Download 到 Flash Memory 之前有在“組譯選項”中勾選“燒錄保護”，則在 Verify 後將執行燒錄保護；如果沒有溝選擇在 Verify 後停止，燒錄完成後 D4 綠色 LED 亮。
- 燒錄完成後按按鍵 S2 再次檢查晶片是否為空，此時應該亮 D4 紅色 LED，表示有燒錄完成
- 如果在執行中有任何一項錯誤或失敗，則 D4 紅色 LED 亮，如果成功則 D4 綠色 LED 亮。



### 4.4.2 限制燒錄次數

在介面設定的“組譯選項”內有一個欄位是點選“Enable Program Times”如圖 72，這個選項是允許限制 Download 程式的燒錄次數。

這是一個安全機制，在產線上限制燒錄次數，防止燒錄過量。

當點選 “Enable Program Times”之後，並在 “Input Program Times”下方欄位上填寫燒錄的次數(最多為 99999999，最少 1)，當在 Compiler 程序後或下載檔案到 Flash Memory 之後，會將此參數載入；當每一次執行燒錄的動作時，會將此計數值自動減 1，當此計數值減到 0 時，如果繼續燒錄，則不會執行，並會亮錯誤訊息(紅色燈號)，但 Blank Check 會正常動作。

### 5.修訂記錄

以下描述本文件差異較大的地方，而標點符號與字形的改變不在此描述範圍。

---

版本	頁次	變更摘要
V01	ALL	初版發行
V02	10	增加 Demo Code 操作方式及使用
	18	增加程式架構說明及程式參考文件
	56	補充線上燒錄與離線燒錄使用方法
V03	4	增加軟體支援 Vista & Win7
	19	增加自訂指令功能
	50	修改燒錄保護說明