



HY17M Series

Assembly IDE 軟體使用說明書

目 錄

1. HY17M IDE 入門.....	5
1.1 簡介	5
1.2 HY17M IDE 安裝及系統要求.....	5
1.3 安裝及卸載.....	5
安裝	5
卸載	7
1.4 DEMO CODE 導入說明.....	7
1.5 DEMO CODE 操作方式及使用	8
2. HY17M IDE 介面說明.....	9
2.1 HY17M IDE 編輯介面	9
2.1.1 編輯視窗	10
2.1.2 檔案	11
2.1.3 編輯	11
2.1.4 檢視視窗	12
2.1.5 組譯&執行.....	12
2.1.6 介面設定	13
2.1.8 程式架構	17
3. HY17M IDE 除錯介面.....	18
3.1 快速執行	19
3.2 RAM 視窗	22
3.3 REGISTER 視窗.....	25
3.4 WATCH 視窗.....	27
3.5 堆棧視窗.....	29

HY17M Series

Assembly IDE 軟體使用說明書

3.6 暫存器修改記錄.....	30
3.7 源程式視窗下的 HINT 功能.....	32
4. 燒錄視窗	34
4.1 介面設定	34
4.2 操作步驟	36
4.2.1 開啟檔案與組譯	36
4.3 PC 連線燒錄 MTP.....	38
4.3.1 晶片檢查(Blank Check).....	39
4.3.2 編程晶片(Program).....	39
4.3.3 確認編程晶片(Verify).....	40
4.3.4 讀取晶片(Read).....	41
4.3.5 AUTO.....	41
5. 故障排除	43
5.1 無法使用 HYCON-IDE	43
6. 修訂記錄	44

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. HY17M 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 Hycon-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 HY17M 系列產品的 MTP 上，使客戶的終端產品能快速的發展並銷售到市面上。



1.2 HY17M IDE 安裝及系統要求

運行 HY17M IDE 所需的最低系統配置：

- PC/NB Hardware requirement:
PC 相容的奔騰 (PENTIUM®) 級系統
512 MB 記憶體 (推薦 1GB)
1 GB 硬碟空間
- Supported Products:
-HY17M Series chip
- Supported Hardware Model No.:
-HY17M24-DK01 : HY17M24 IDE Hardware (development kit)
-HY17M26-DK01 : HY17M26 IDE Hardware (development kit)
- Supported Software version:
HY17M IDE V2.0.2 above : HY17M Series Assembly IDE software
- Supported Operating System:
Windows XP, Windows Vista, Windows 7, Windows 8, Windows 10
- Apply the following Interface Modes:
USB Port with HID-compliant device

1.3 安裝及卸載

安裝

解壓縮 IDE 軟體包後運行  setup 就可以進行軟體的安裝(步驟如下圖 1)·安裝完畢後直接運行\HYCON\HY17M IDE 目錄下的  HY17M IDE.exe 就可以開啟 IDE 軟體進行操作。

HY17M Series


Assembly IDE 軟體使用說明書

Note : 在運行軟體時請以『系統管理員身份』來運行



圖 1

卸載

直接運行 IDE 軟體目錄下的  **unins000**，就能解除 IDE 軟體的安裝。

1.4 Demo Code 導入說明

- 開啟 C:\HYCON\HY17M IDE\DemoCode 主程式
- 設定為組譯主檔
- 組譯並進行除錯

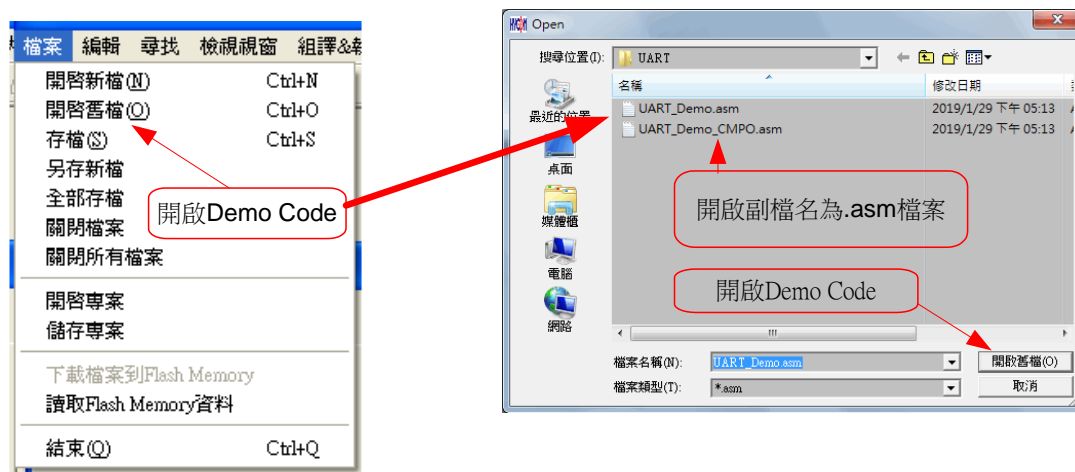


圖 2

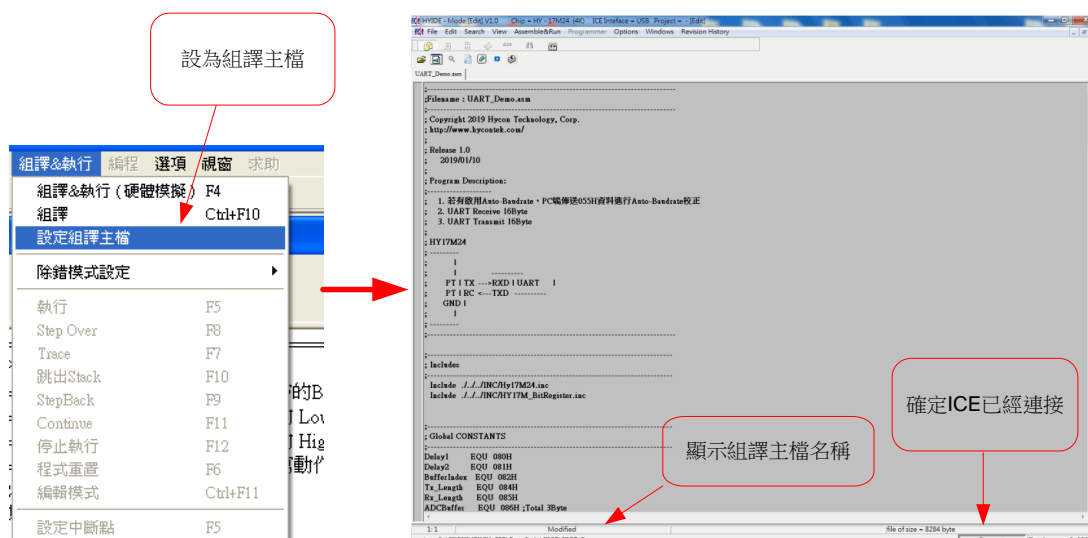


圖 3

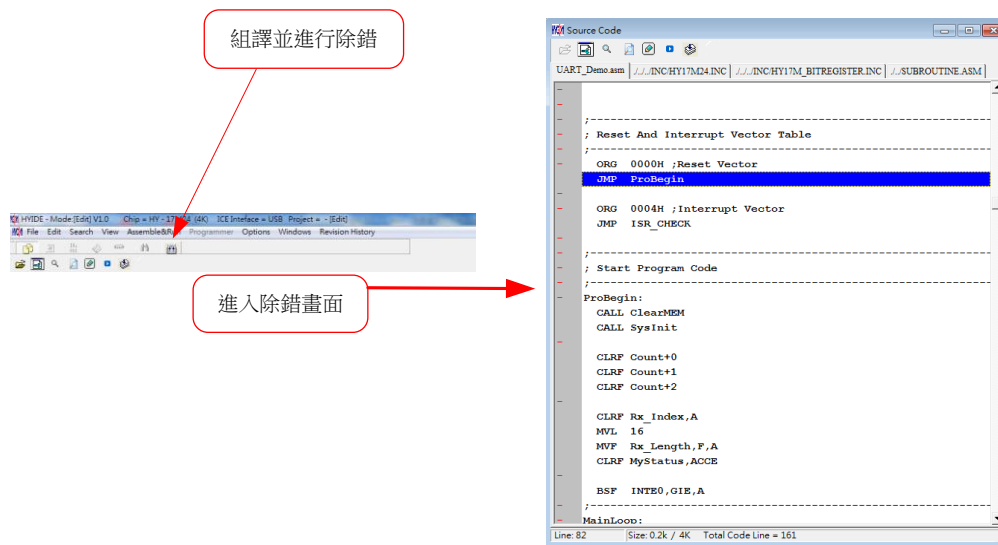


圖 4

- 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

1.5 Demo Code 操作方式及使用

- 執行 HY17M IDE 軟體安裝後，於目錄 C:\HYCON\HY17M IDE\DemoCode 主程式下會有提供 Demo Code 供使用者參考。

2. HY17M IDE 介面說明

2.1 HY17M IDE 編輯介面

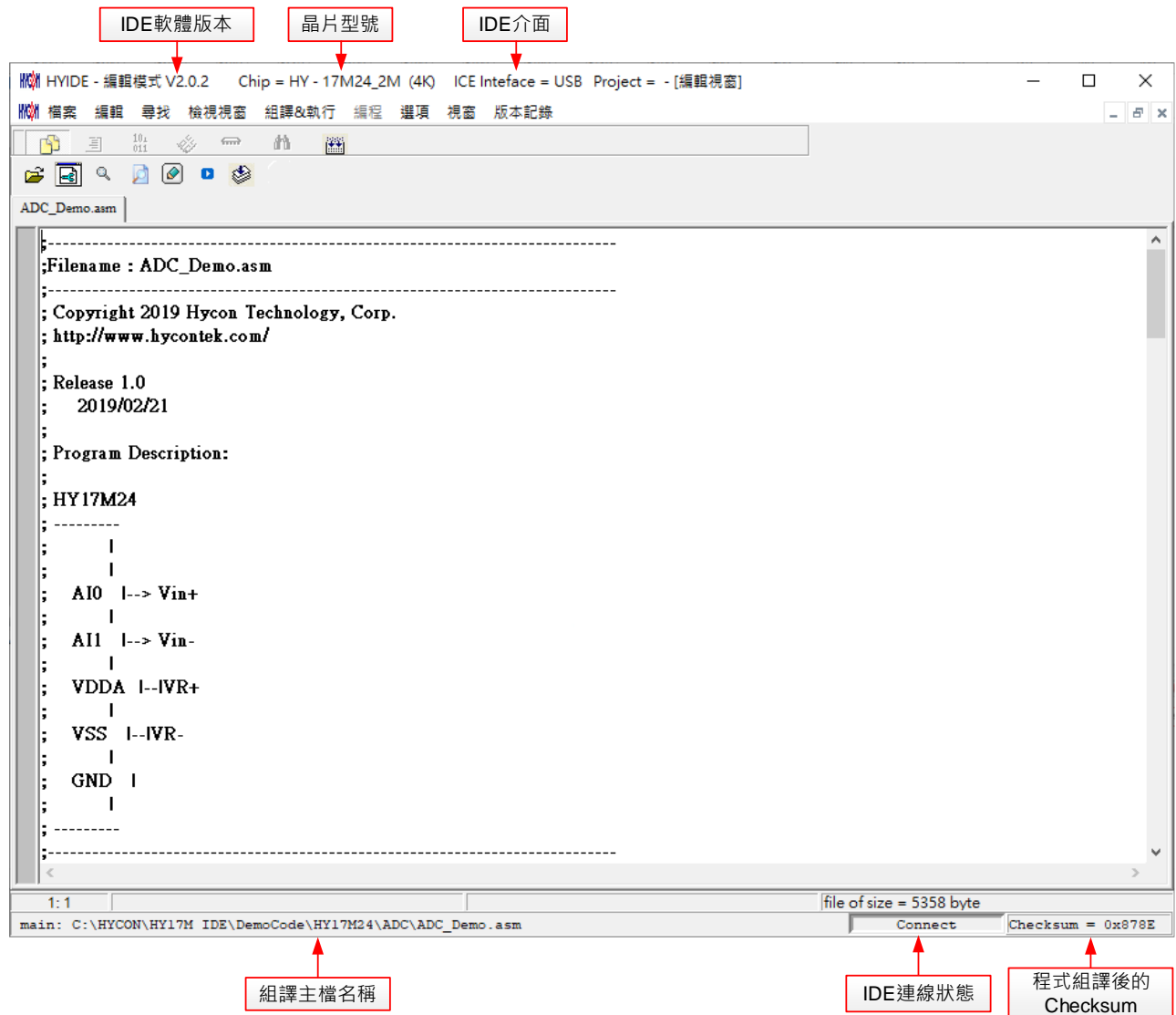









圖 5

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

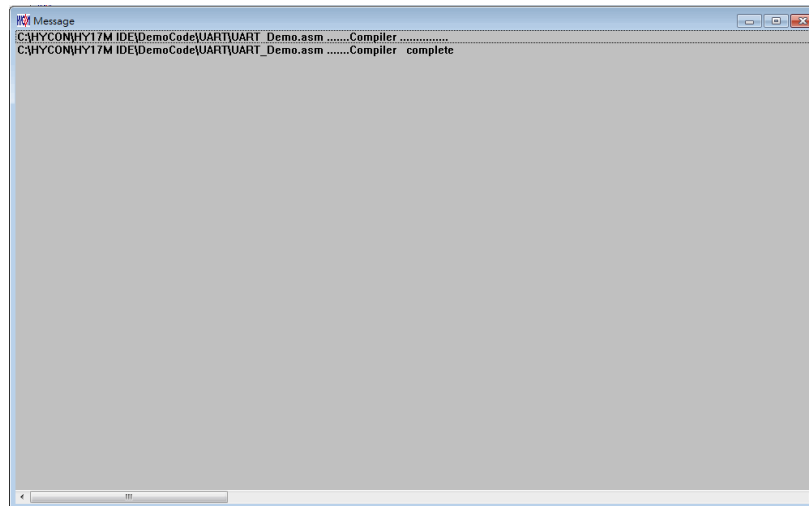


圖 6

2.1.2 檔案

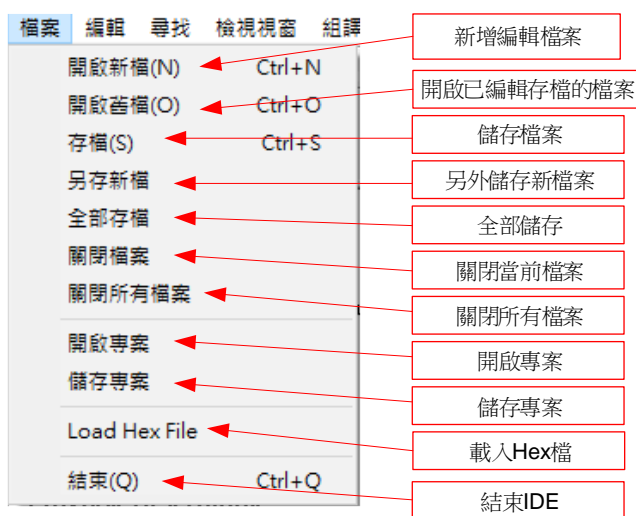


圖 7

- 開啟新檔 → 新增編輯檔案
- 開啟舊檔 → 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
- Load Hex File → 載入 Hex 檔案
- 結束 → 結束 Hycon-IDE 程式

2.1.3 編輯

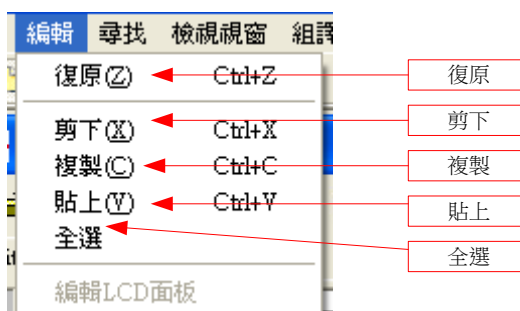


圖 8

- 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 → 貼上復制的區域
- 全選 → 全部選擇

2.1.4 檢視視窗



圖 9

- 編輯視窗 → 將編輯視窗指定為當前的活動視窗
- 訊息 → 將訊息欄指定為當前的活動視窗
- Read EEPROM → 將 EEPROM Memory 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

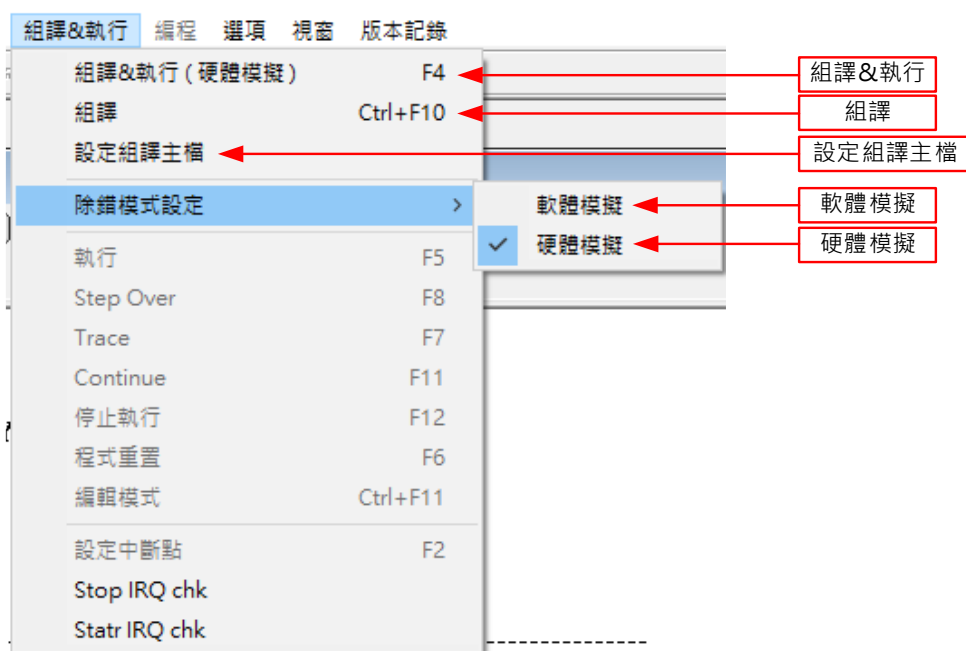


圖 10

- 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

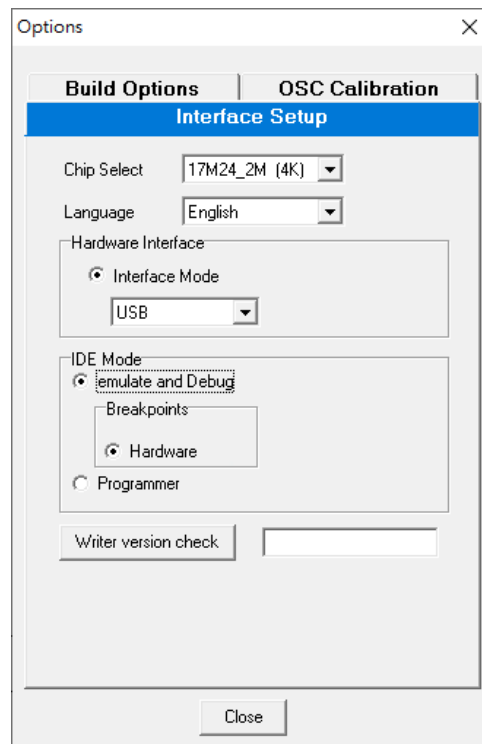


圖 11

- 介面設定(由選項中選取)



圖 12

- 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。
- Language：可選擇英文或中文介面。
- 軟件設置：選擇 USB 傳輸介面。
- IDE 模式：仿真與除錯、燒錄編程兩種選擇。

- 組譯選項



圖 13

- 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
- ASCII 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 MTP 的設定中。
- 燒錄次數限制：參考燒錄視窗之 2.1.6 介面設定章節。
- 編輯功能字型選擇：選擇編輯器的字型大小。
- 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。
- User Key：燒錄視窗之 2.1.6 介面設定章節。

- OSC Calibration

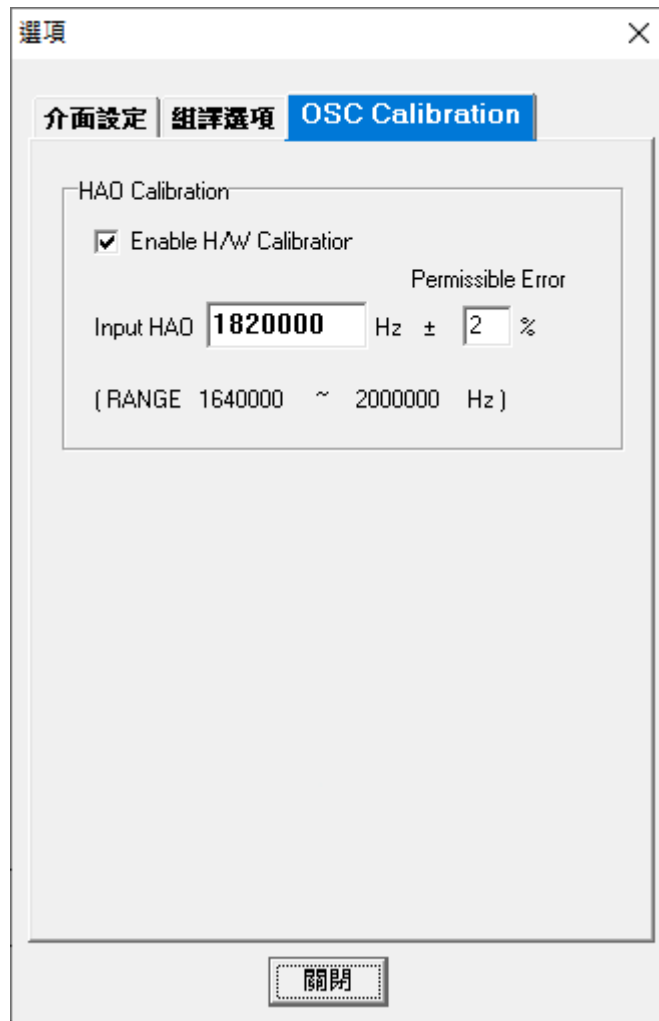


圖 14

- Enable H/W Calibration: 勾選是否進行硬體校正
- 輸入 HAO 校正值與誤差值，下方 RANGE 根據晶片型號改變

2.1.7 視窗

可選擇所有開啟的視窗做垂直或水平的排列。

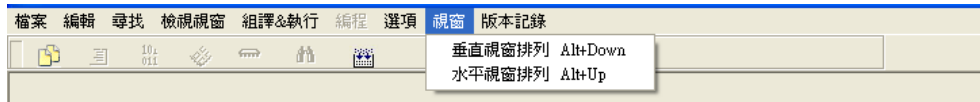


圖 15

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, H08D 指令集；

使用者一開始可以先參考軟體所附的 demo code，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:
 - `Include Hy17M24.inc` ;程式名稱可為任意名稱
 - `Include Main.inc` ;HY17M 系列特殊暫存器名稱、位址定義
 - `ORG 00H` ;RAM 名稱、位址定義
 - `JMP BEGIN` ;宣告程式開始
 - `ORG 04H` ;跳躍到主程式
 - `Include Interrupt.asm` ;宣告中斷旗標位置
 - `Include Interrupt.asm` ;引用“Interrupt.asm”中斷副程式；
 - `include` 檔案限制最多 100 個。
 - `BEGIN:` ;主程式開始. Label name 的定義可以為任意字
 - `Include Initial.asm` ;引用“Initial.asm”硬體初始化副程式
 - `JMP T1` ;跳躍到 T1 副程式
 - ...
 - `T1:`
 - `NOP`
 - `Include Sub.asm` ;引用“Sub.asm”副程式
 - `END` ;程式結束

3. HY17M 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

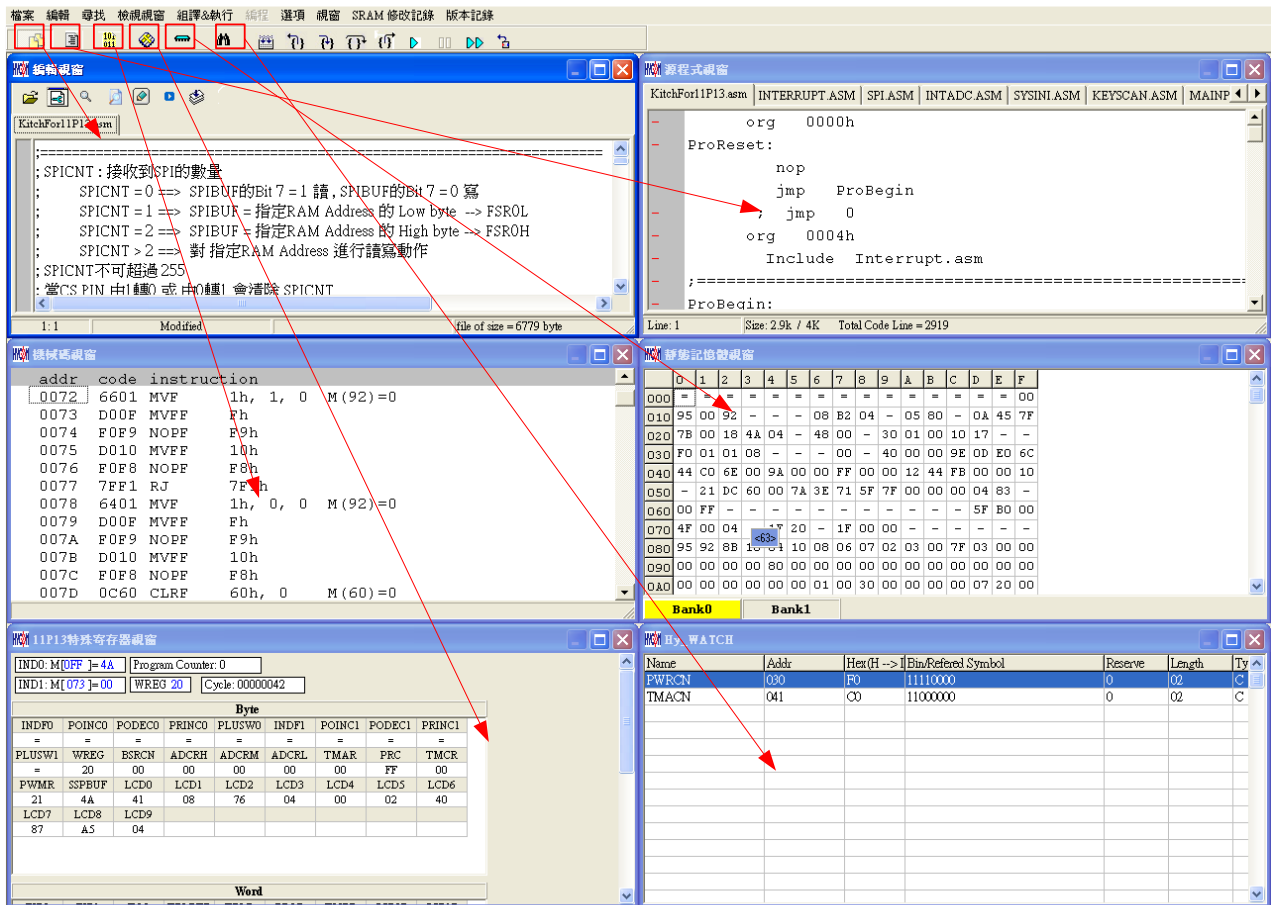
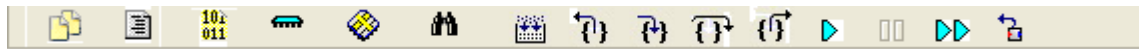



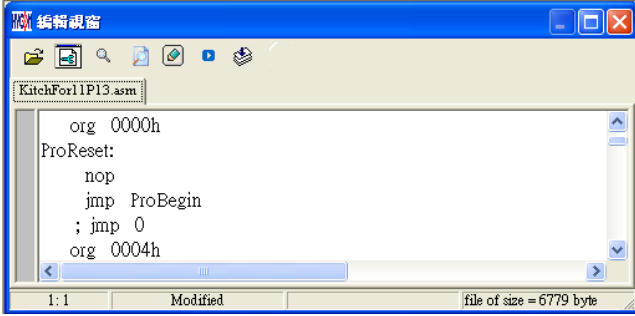
圖 16


3.1 快速執行

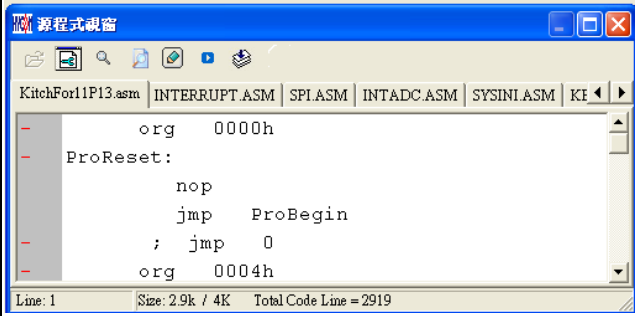



- 快速視窗切換

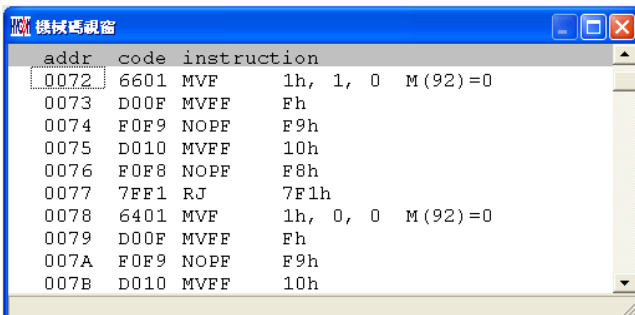
(1) 切換至 Edit 視窗 




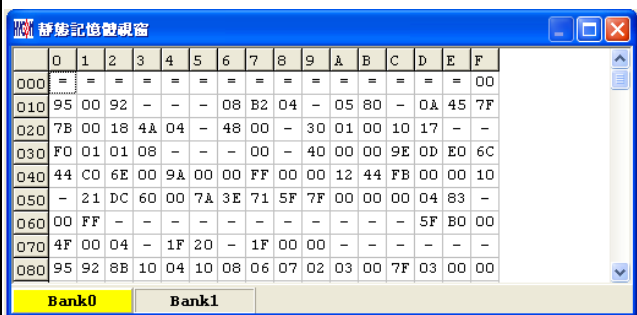
(2) 切換至 source 視窗 




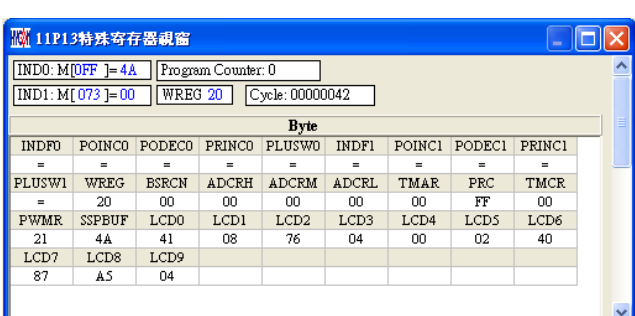
(3) 切換至 Hex 視窗 




(4) 切換至 Ram 視窗 



(5) 切換至 Reg 視窗 



(6) 切換至 Watch 視窗 

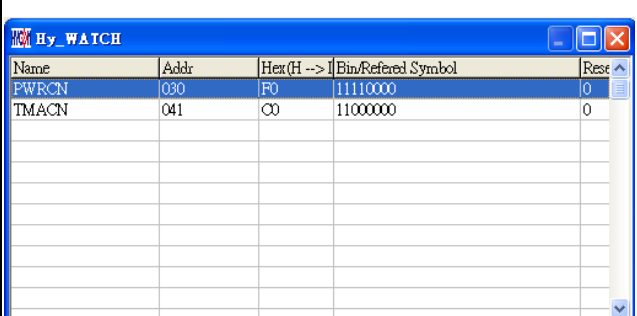


圖 17

- 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 

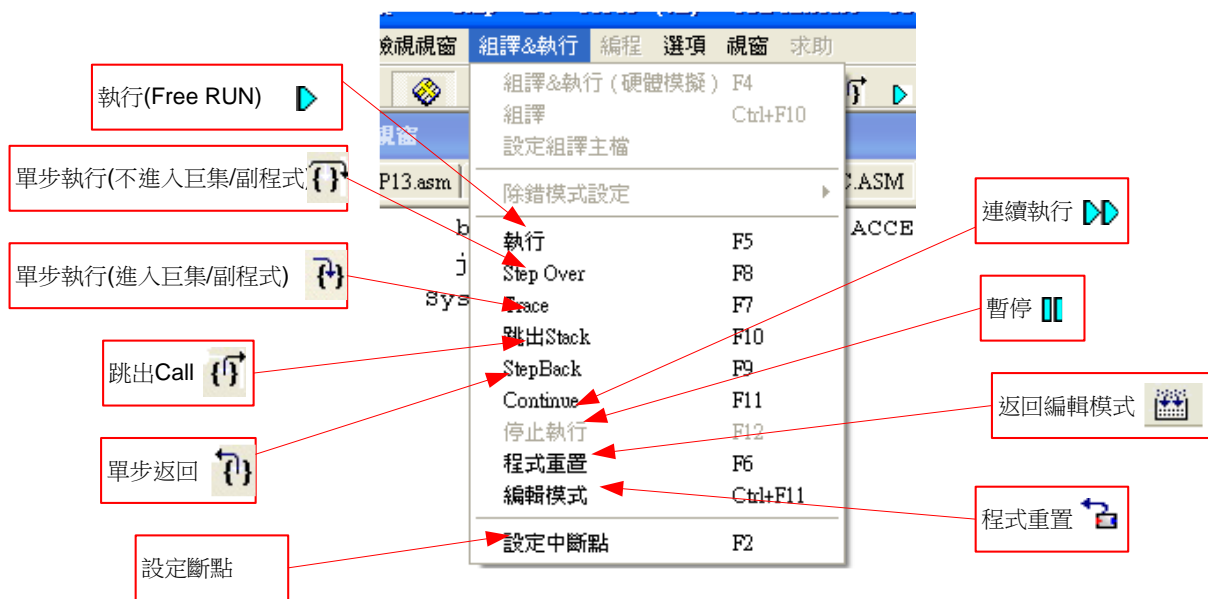


圖 18

- 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。

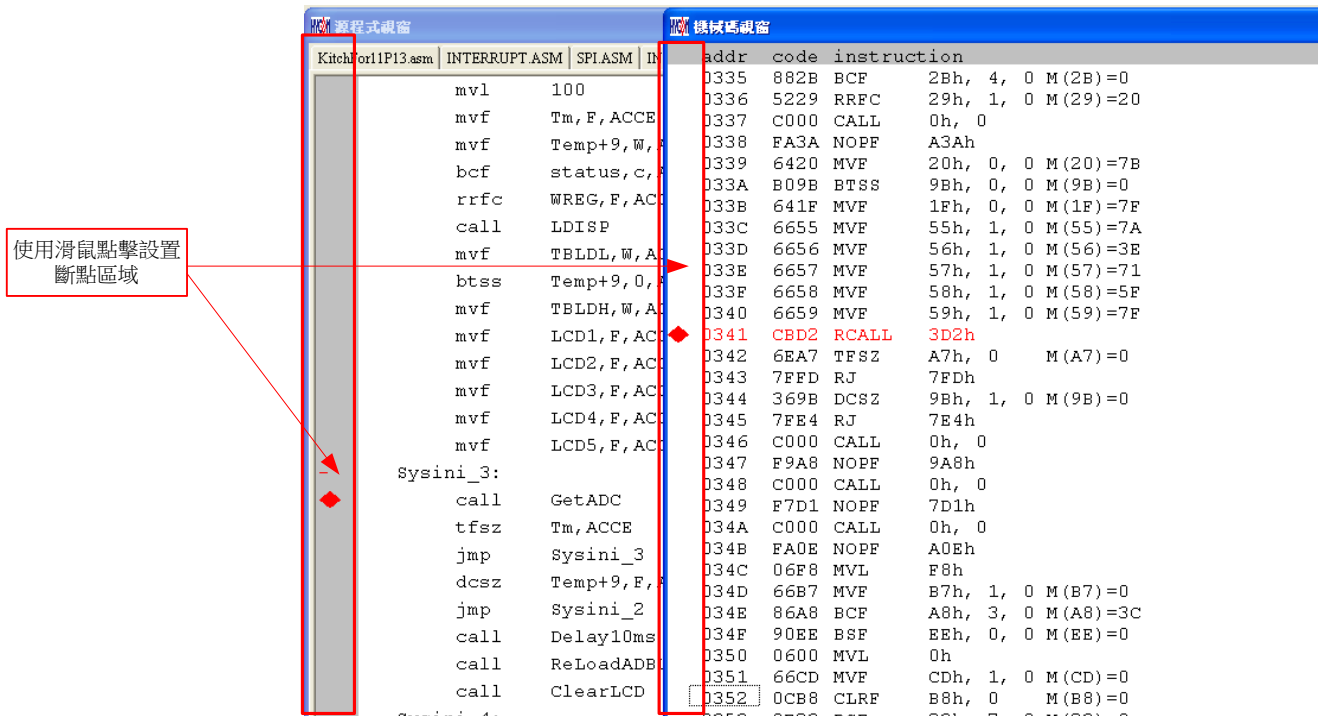


圖 19

3.2 RAM 視窗

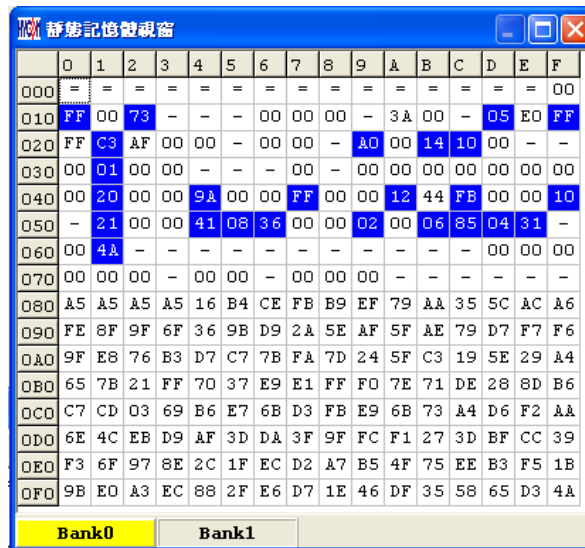


圖 20

- 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)

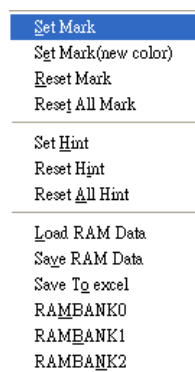


圖 21

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

HY17M Series

Assembly IDE 軟體使用說明書

- 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]

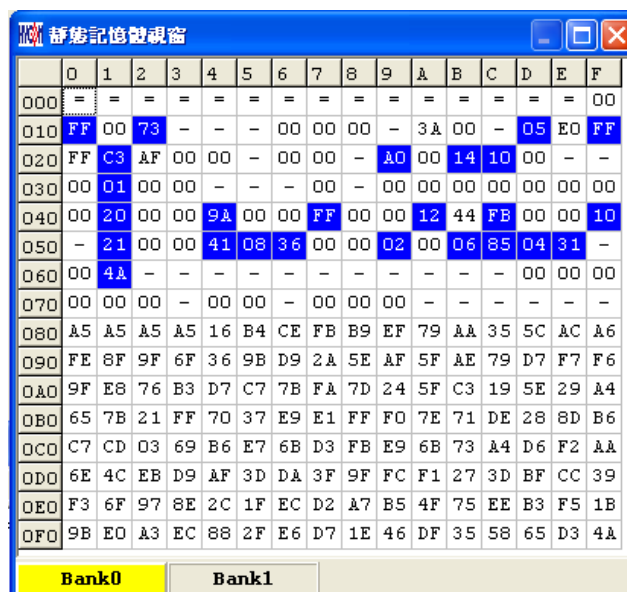


圖 22

HY17M Series

Assembly IDE 軟體使用說明書

- 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，使用鍵盤 Key In 或滑鼠點擊

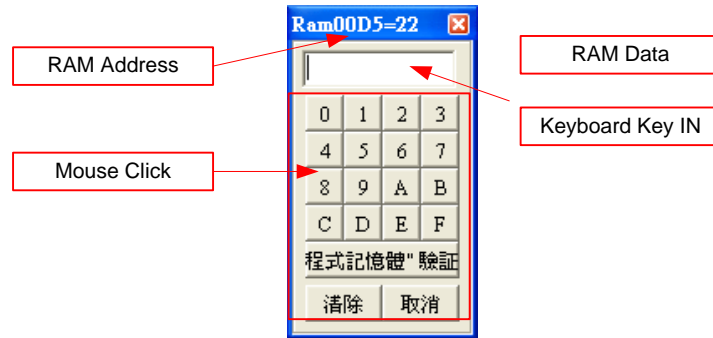


圖 23

3.3 Register 視窗

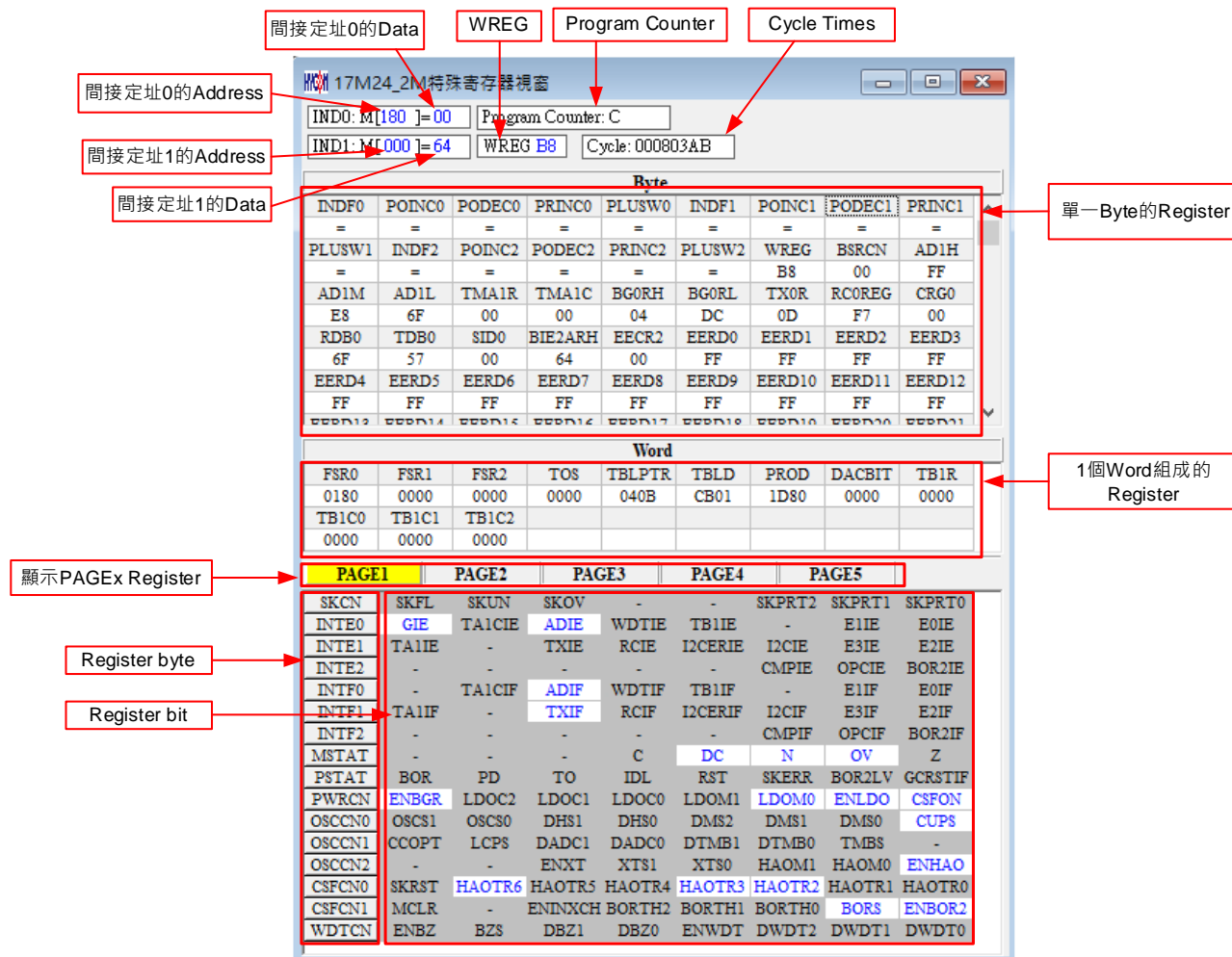


圖 24

- 修改間接定址 Data 或 Address

如圖 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 25

如圖 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 26

- 修改 WREG 的 Data

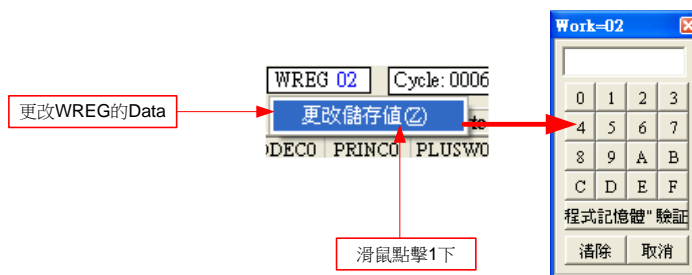


圖 27

- 修改單 1byte 或 Word Register 的 Data

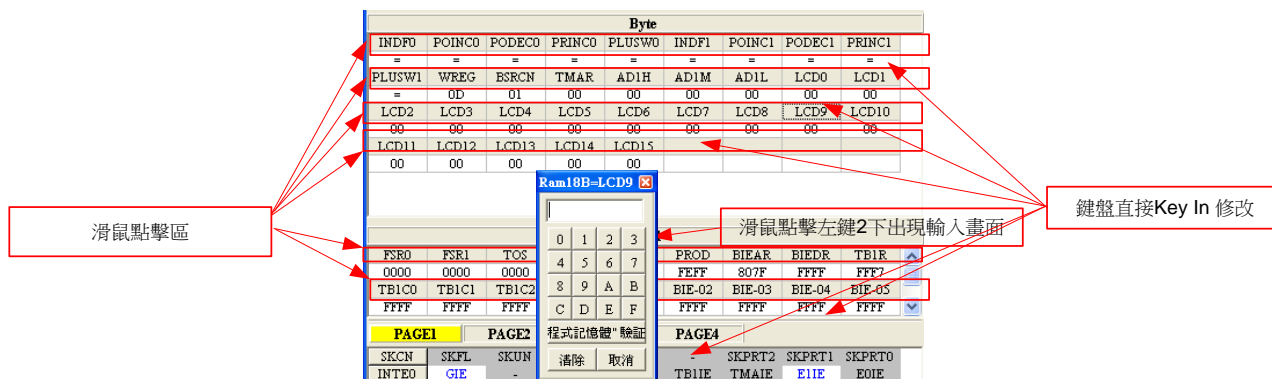


圖 28

- 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

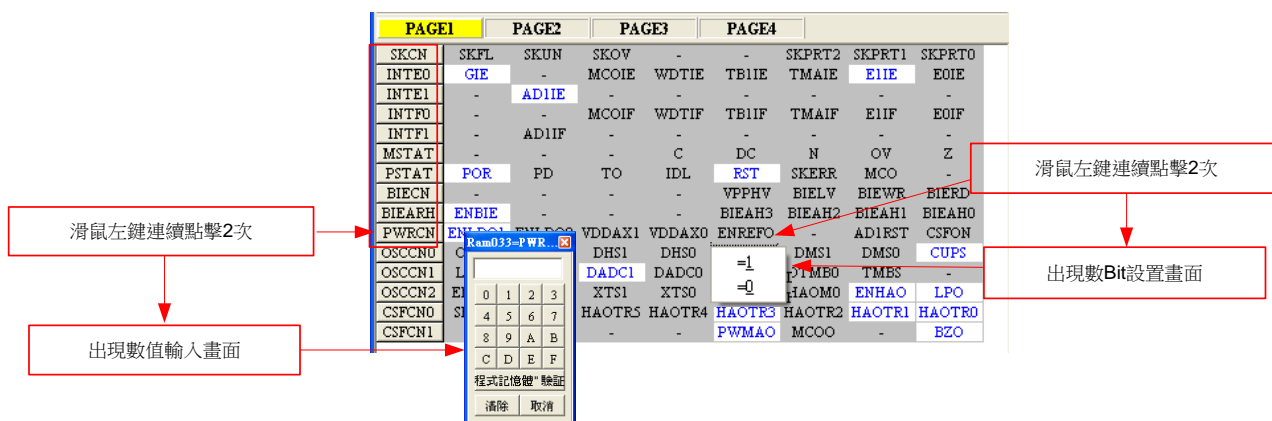


圖 29

3.4 Watch 視窗

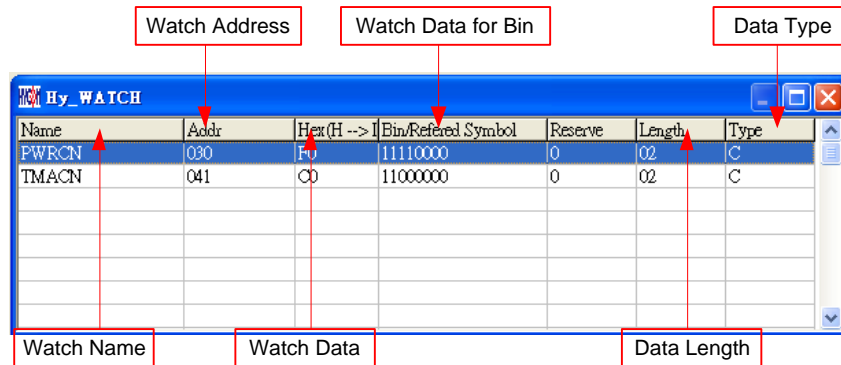


圖 30

- 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

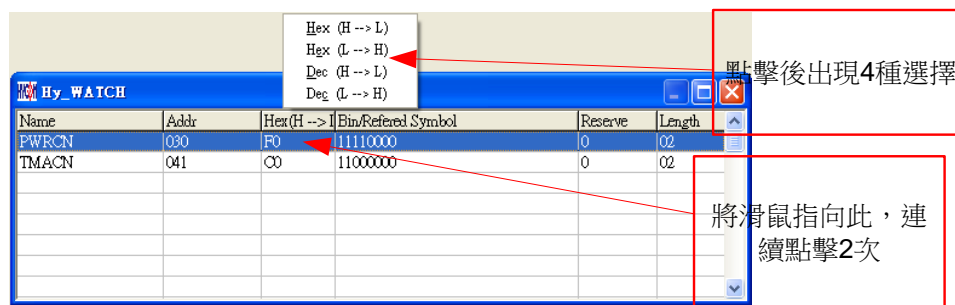


圖 31

- 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

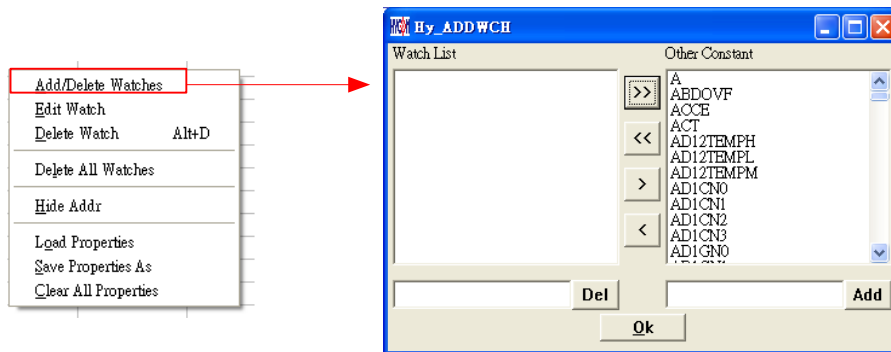


圖 32

3.5 堆棧視窗

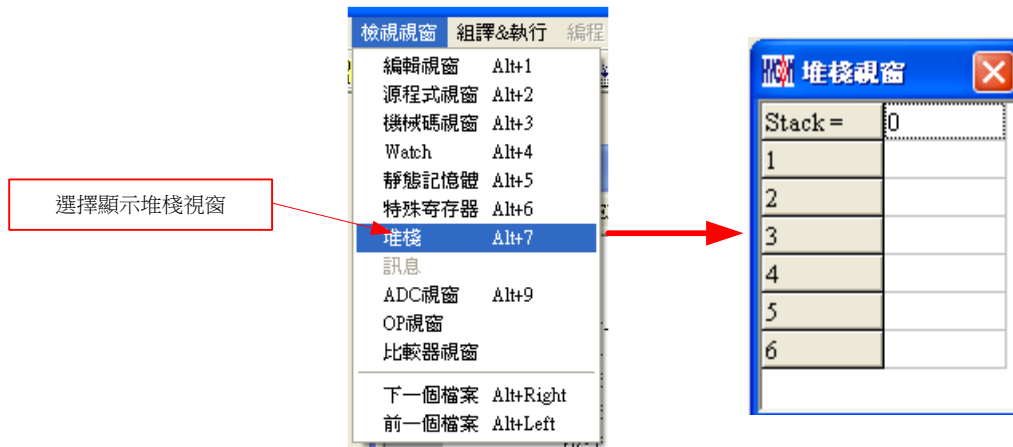


圖 33



圖 34

3.6 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。

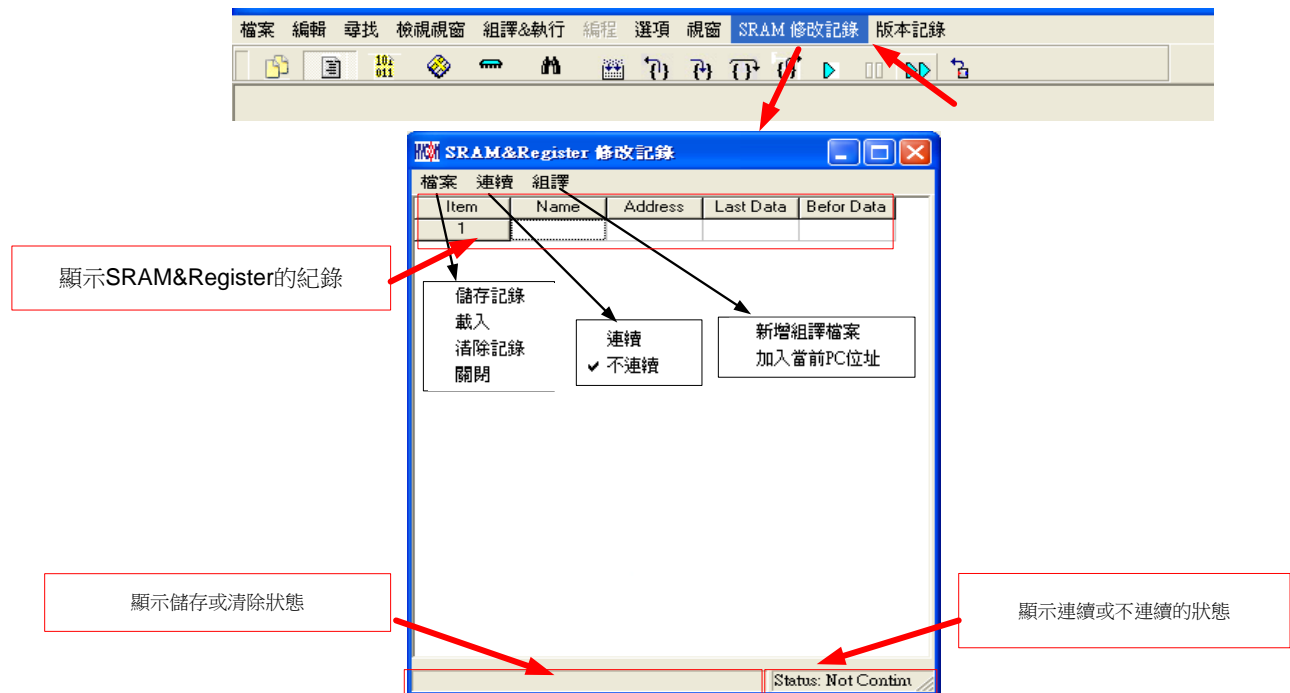


圖 35

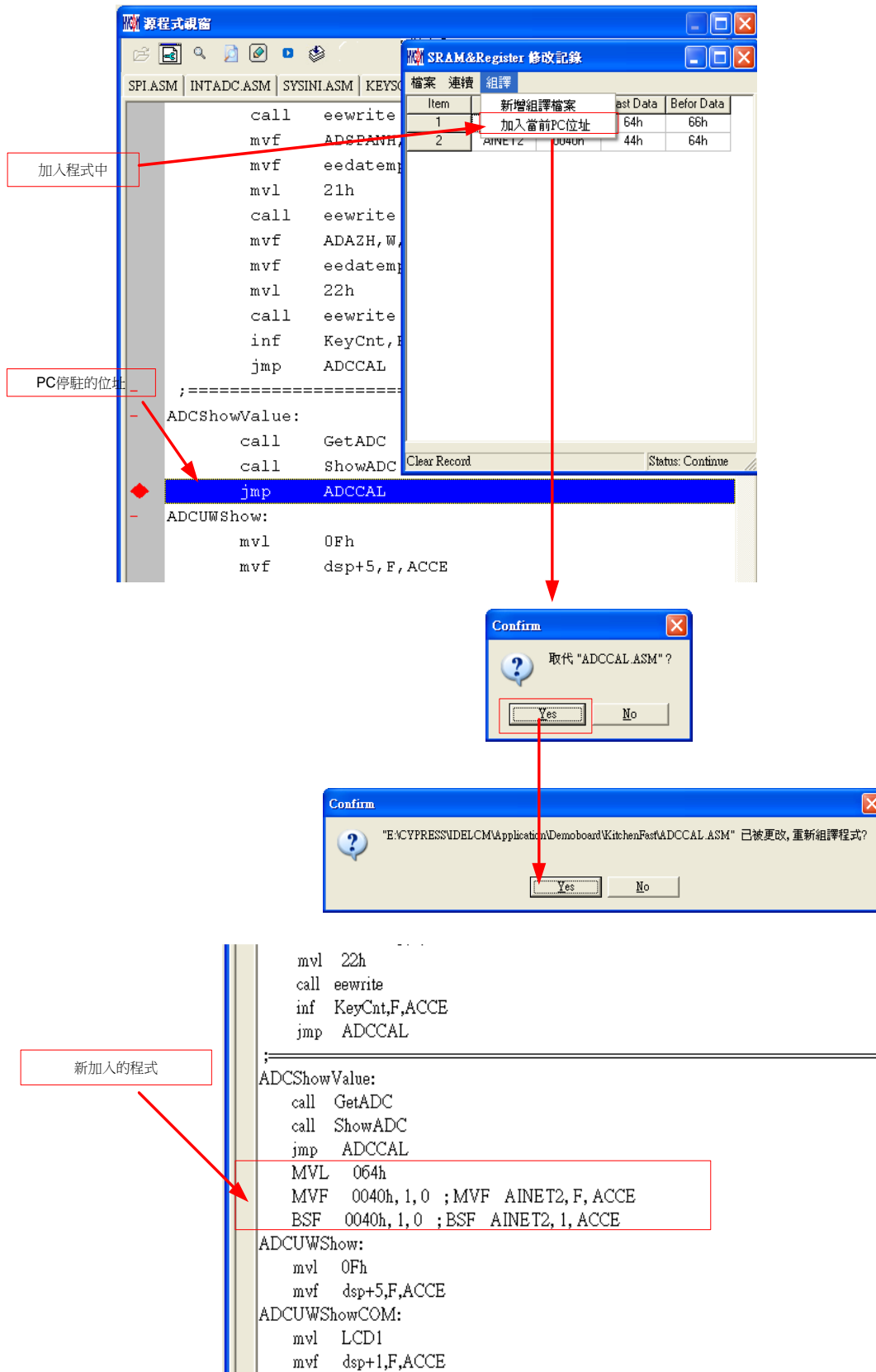


圖 36

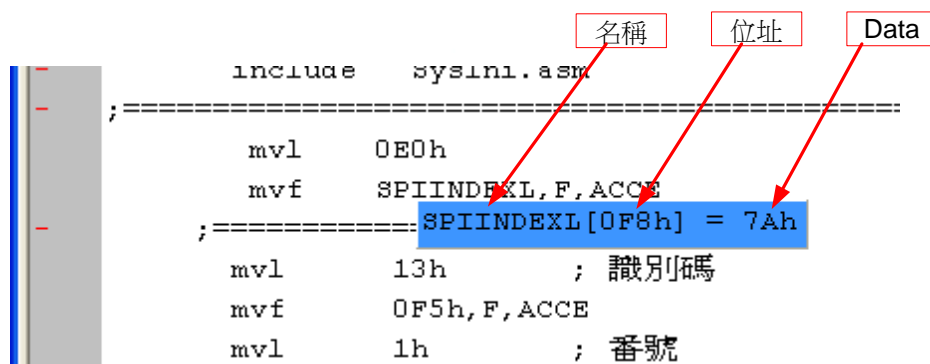
3.7 源程式視窗下的 Hint 功能

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)。

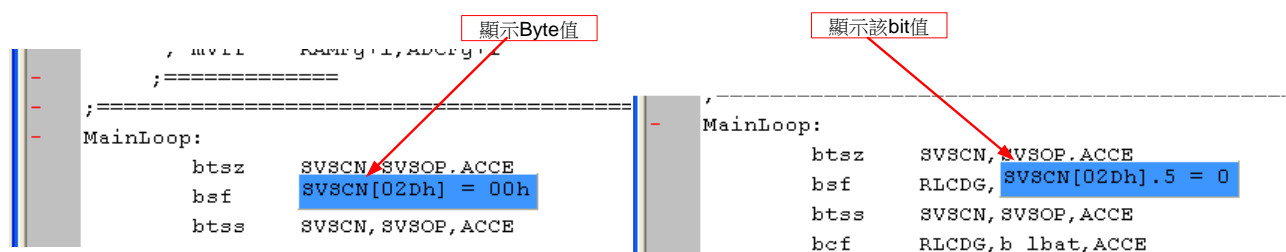
- 當指令為位元操作時只有第一個參數才有效
- 當指令為 BCF、BSF、BTSS、BTSZ 及 BTGF 時，當指向第一個參數顯示 Byte 數值，當指向第二個參數顯示該 Bit 的值(1 or 0)
- 當指令為 MVFF 時(不是 Macro)，當指向第一個參數顯示第一個參數數值，當指向第二個參數顯示第二個參數數值
- 如果參數為 INDF0、POINC0、PODEC0、PRINC0、INDF1、POINC1、PODEC1、PRINC1 時，此時 Data 為 FSR0 或 FSR1 內的位址的 Data
- 如果參數為 PLUSW0 或 PLUSW1 時，此時 Data 為 FSR0+WREG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



```

include sysin1.asm
;=====
mvl 0E0h
mvf SPIINDEXL, F, ACCE
;===== SPIINDEXL[0F8h] = 7Ah
mvl 13h ; 識別碼
mvf 0F5h, F, ACCE
mvl 1h ; 番號
    
```

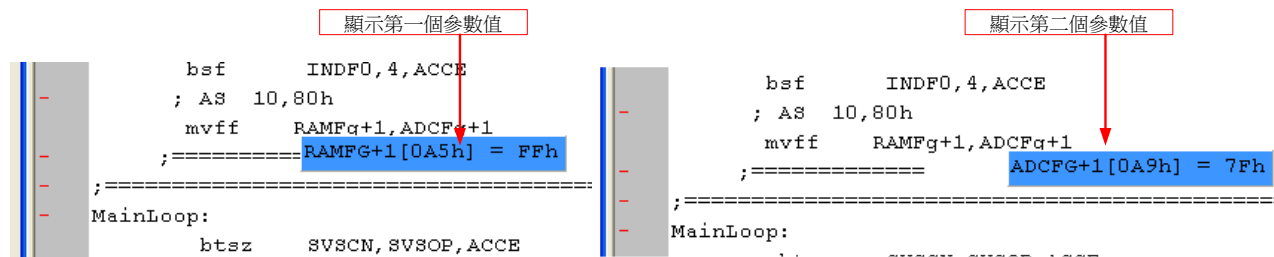
圖 37



```

; mvl 0E0h, ACCE
;=====
MainLoop:
btsz SVSCN, SVSOP, ACCE
bsf SVSCN[02Dh] = 00h
btss SVSCN, SVSOP, ACCE
;=====
MainLoop:
btsz SVSCN, SVSOP, ACCE
bsf RLC DG, SVSCN[02Dh].5 = 0
btss SVSCN, SVSOP, ACCE
bcf RLC DG, b_lbat, ACCE
    
```

圖 38



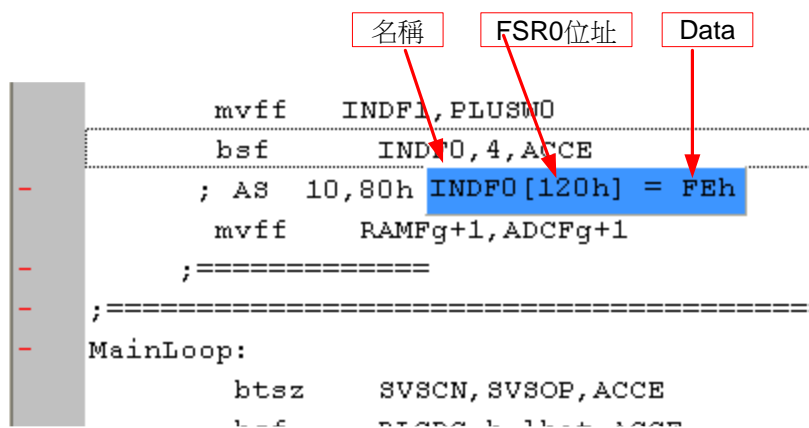
```

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:
    
```

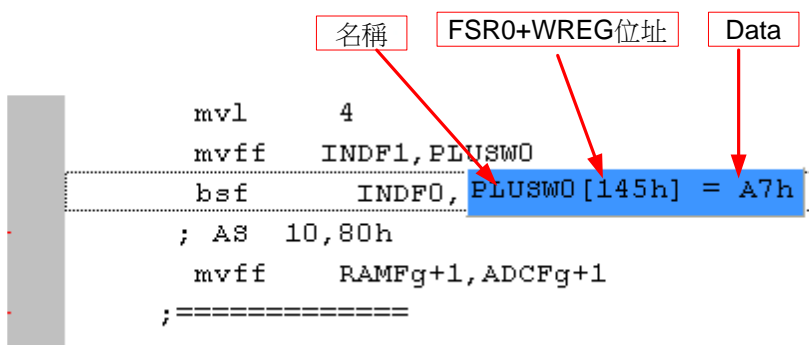
圖 39



```

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

圖 40



```

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

圖 41

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 42 所示。



圖 42

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 43。

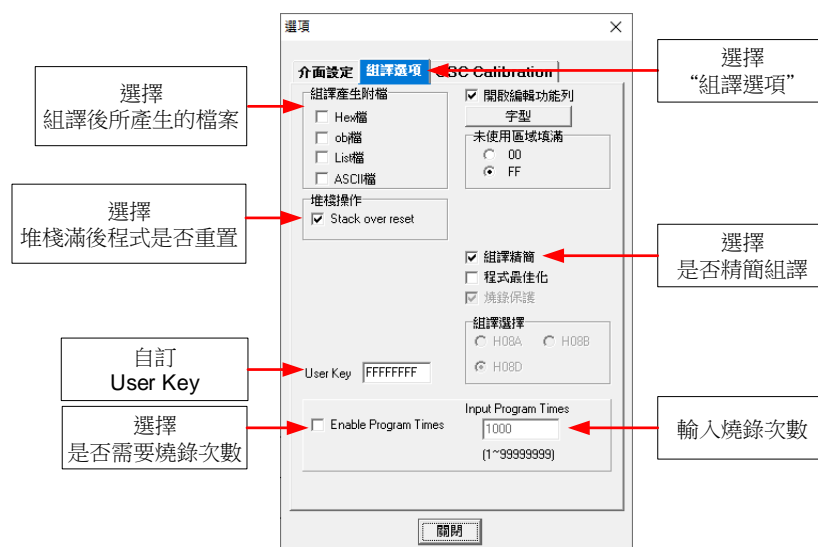


圖 43

- 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 MTP 程序運行後如果發生堆棧滿或溢位是否要重置。
- 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)。
- User Key → 填入讀取保護密碼，屆時回讀程式需輸入此密碼。

4.2 操作步驟

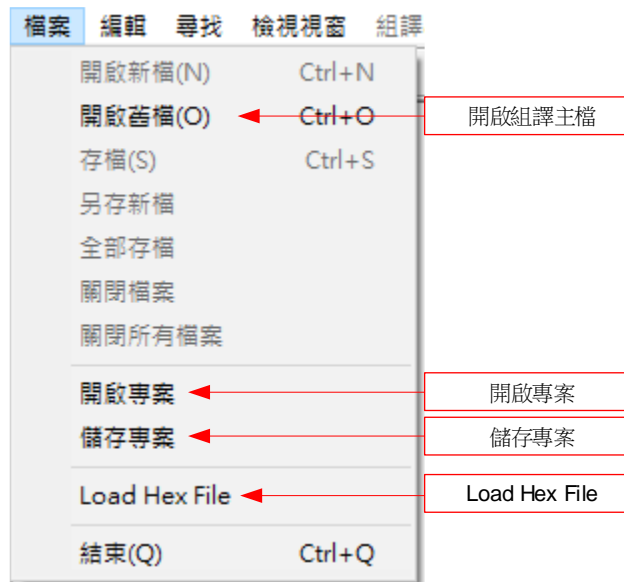


圖 44

- 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。
- Load Hex File → 載入 Hex 檔案

4.2.1 開啟檔案與組譯

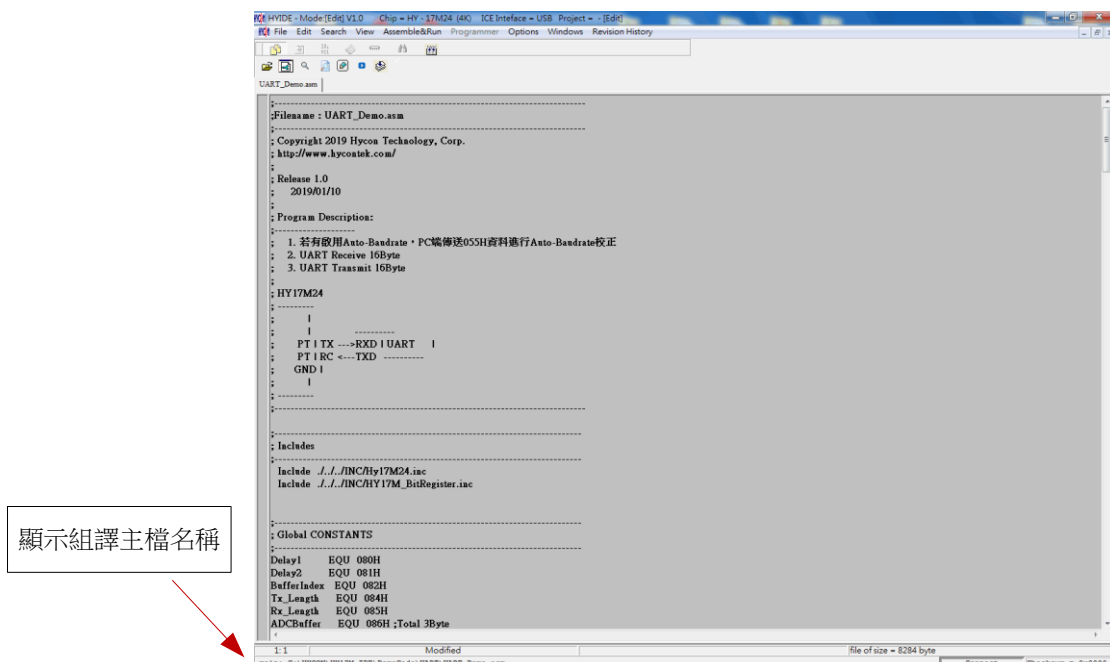


圖 45

HY17M Series

Assembly IDE 軟體使用說明書

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。

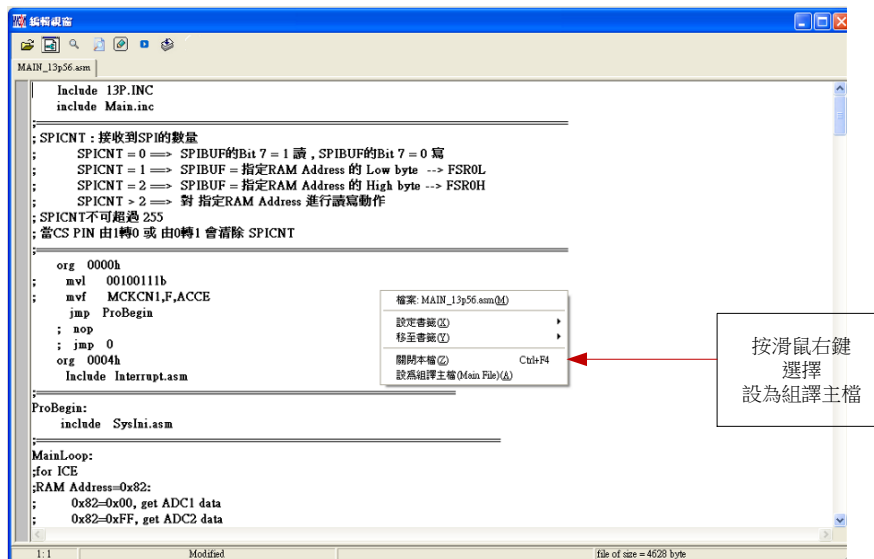


圖 46

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



圖 47

當組譯完成後在下方顯示組譯完成後的 Hex 檔名稱與 Checksum。

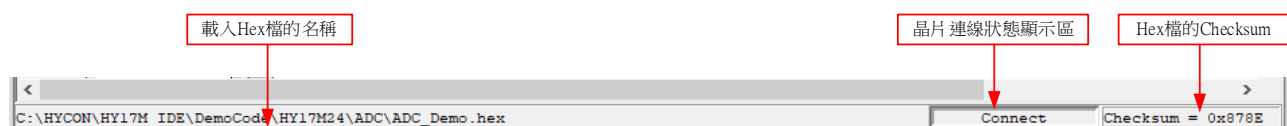


圖 49

4.3 PC 連線燒錄 MTP



圖 50

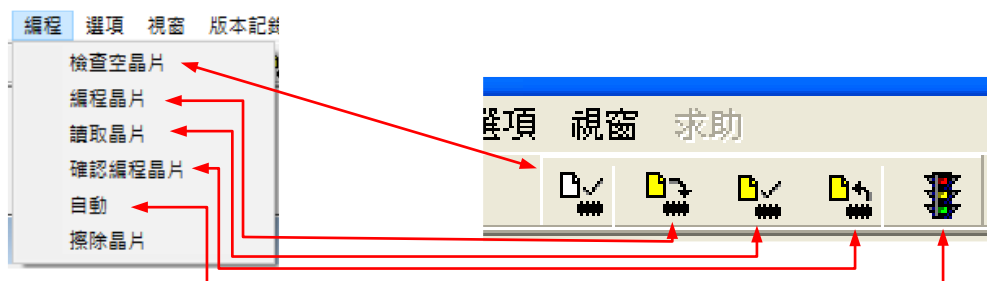


圖 51

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、擦除晶片等動作，如果沒有成功載入，則以上的動作將不會成功。如圖 52~圖 54。

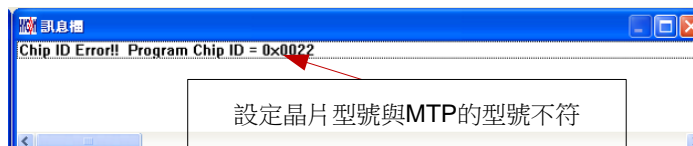


圖 52

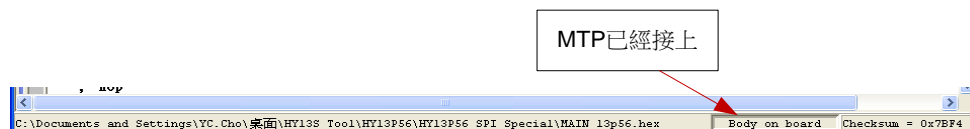


圖 53

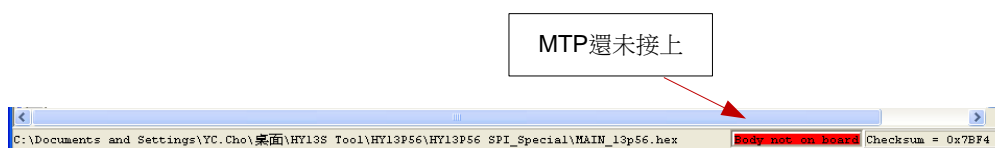


圖 54

確定在標題視窗下所選擇的燒錄晶片型號，與 MTP 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 MTP 型號是否相同，如果不同否則不會燒錄到 MTP 內。

如下圖所示，Chip Locked 意旨晶片已燒入成功且設定 User Key，故 IDE 無法順利連接，擦除晶片即可重新連接 IDE。此情況下仍可直接重新燒入(即擦除晶片後再燒入)或進入 Debug 模式。

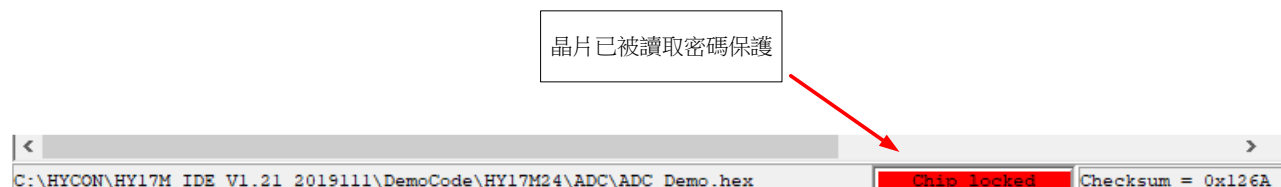


圖 55

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 MTP 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 MTP 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 56)。




圖 56

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 57)。



圖 57

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 MTP 的晶片或燒錄到控制盒中可進行離線燒錄功能，左擊圖示則跳出燒錄訊息視窗(圖 58)，選擇燒錄到晶片或控制盒，最後燒錄完成並組裝成品將可依照使用者所寫的指令運行程序。

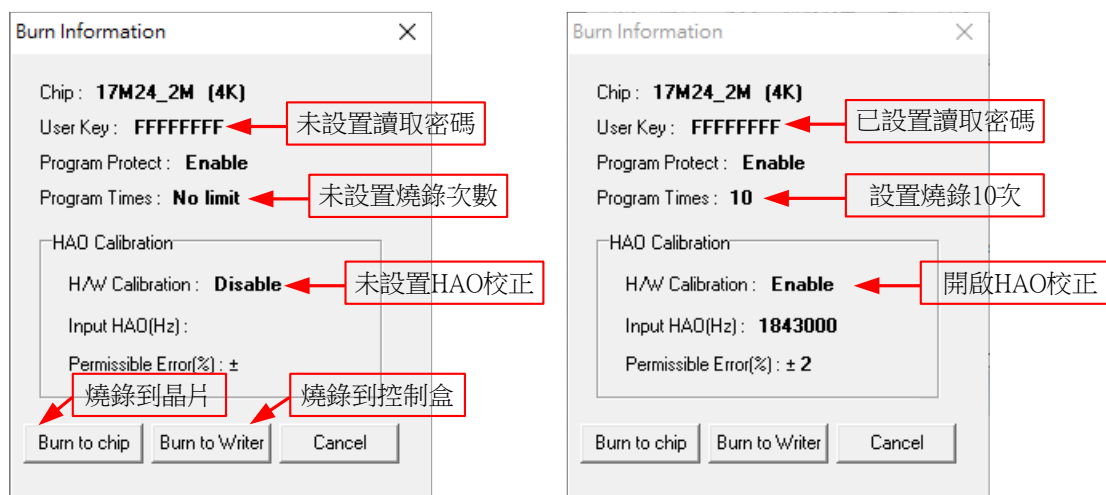


圖 58

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。

如果選擇晶片正確以及編程成功·訊息欄出現以下訊息(圖 59)·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。

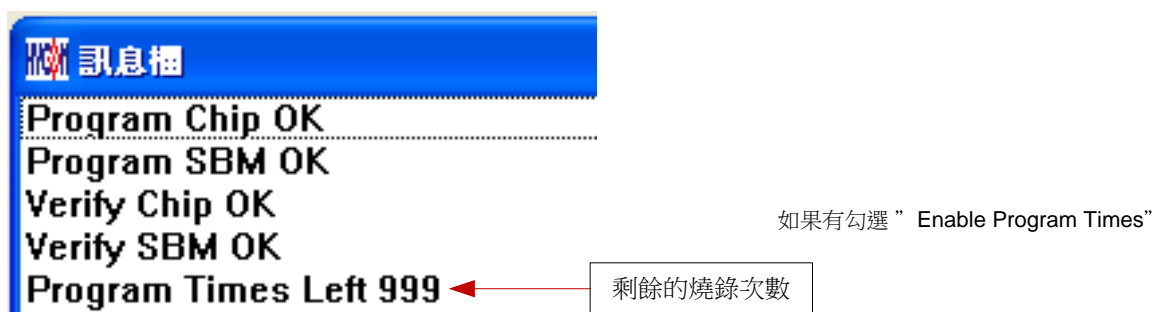



圖 59

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 60)。



圖 60

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 MTP 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 61)。



圖 61

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 62)。



圖 62

4.3.4 讀取晶片(Read)

讀取晶片(Read)圖示為 ，讀取晶片的目的是讓使用者確認讀取 MTP 的 Checksum 是否與燒錄的 Hex 檔相同。讀取晶片內容(步驟如圖 63)，並將內容顯示於”顯示 Code”視窗內。如有設置 User Key，操作此動作則需要輸入 User Key 即可讀取晶片。

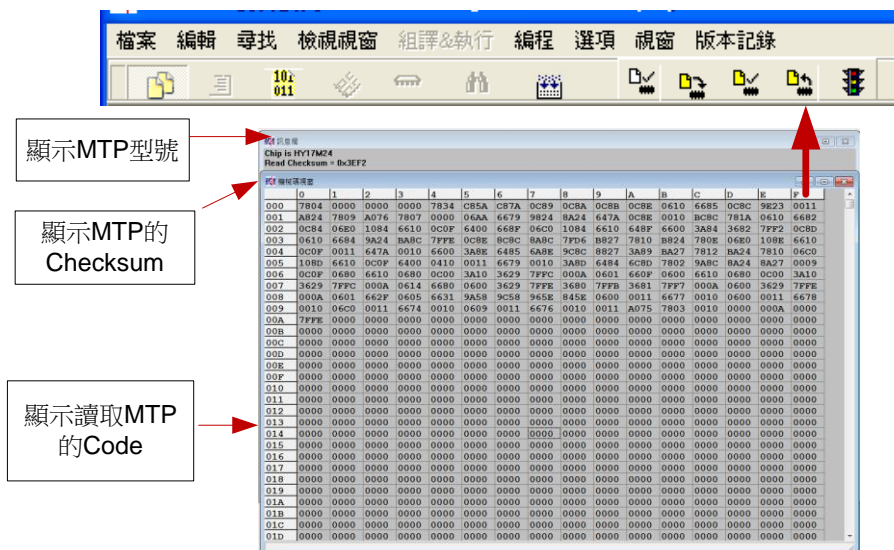


圖 63

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

HY17M Series

Assembly IDE 軟體使用說明書

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 64)，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 64

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。

5. 故障排除

5.1 無法使用 Hycon-IDE

如出現下圖 65

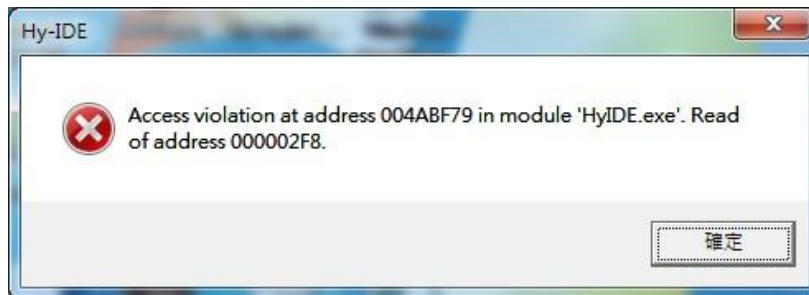


圖 65

通常在使用 windows 7 以上會出現這樣的問題，則必須將 Hycon-IDE.exe 設定成如下圖 66，以系統管理員的身分執行此程式，這樣將可避免使用遇到相同問題。

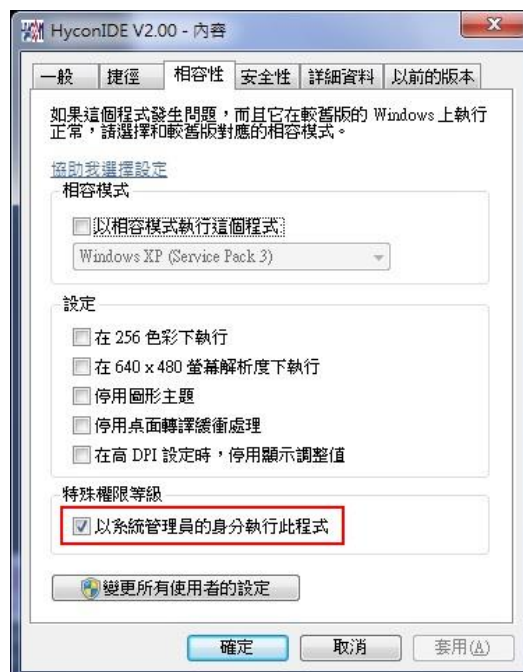


圖 66

HY17M Series

Assembly IDE 軟體使用說明書

6. 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。

文件版次	頁次	日期	摘要
V01	ALL	2019/02/14	初版發行
V02	ALL	2021/11/3	新增支援產品型號 修改部分圖示以符合新版本介面 修改部分功能說明