HY17P 系列

汇编 IDE 软件使用说明书

目 录

注意：

1、 本说明书中的内容，随着产品的改进，有可能不经过预告而更改。请客户及时到本公司网站下载更新 http://www.hycontek.com。

2、 本规格书中的图形、应用电路等，因第三方工业所有权引发的问题，本公司不承担其责任。

3、 本产品在单独应用的情况下，本公司保证它的性能、典型应用和功能符合说明书中的条件。当使用在客户的产品或设备中，以上条件我们不作保证，建议客户做充分的评估和测试。

4、 请注意输入电压、输出电压、负载电流的使用条件，使 IC 内的功耗不超过封装的容许功耗。对于客户在超出说明书中规定额定值使用产品，即使是瞬间的使用，由此所造成的损失，本公司不承担任何责任。

5、 本产品虽内置防静电保护电路，但请不要施加超过保护电路性能的过大静电。

6、 本规格书中的产品，未经书面许可，不可使用在要求高可靠性的电路中。例如健康医疗器械、防灾器械、车辆器械、车载器械及航空器械等对人体产生影响的器械或装置，不得作为其部件使用。

7、 本公司一直致力于提高产品的质量和可靠度，但所有的半导体产品都有一定的失效概率，这些失效概率可能会导致一些人身事故、火灾事故等。当设计产品时，请充分留意冗余设计并采用安全指标，这样可以避免事故的发生。

8、 本规格书中内容，未经本公司许可，严禁用于其他目的之转载或复制。

1. HY17P IDE 入门

1.1. 简介

为了方便客户使用紘康科技(HYCON)的 MCU 系列产品 ,开发出 Hycon-IDE 的发展环境，客户能在此开发平台上模拟仿真其终端产品，并将程序烧录到 HY17P 系列产品的 OTP 上，使客户的终端产品能快速的发展并销售到市面上。

1.2. HY17P IDE 安装及系统要求

运行 HY17P IDE 所需的最低系统配置：

- PC/NB Hardware requiremen:
  PC 兼容的奔腾（PENTIUM®）级系统
  512 MB 存储器（推荐 1GB）
  1 GB 硬盘空间

- Supported Products:
  -HY17P48
  -HY17P51
  -HY17P52
  -HY17P55
  -HY17P56
  -HY17P58
  -HY17P60 & HY17P60B
  -HY17P68

- Supported Hardware Model No:
  -HY17S58-DK02：HY17S58 IDE hardware (development kit)
  -HY17S68-DK02：HY17S68 IDE hardware (development kit)
  -HY17S68-DK03：HY17S68 IDE hardware (DMM 专用 development kit)

- Supported Software version:
  HY17P IDE V1.2 or above：HY17P 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. 安装及卸载

安装

对于某些 Windows 作业系统，要在计算机中安装软件，需要管理员访问权限。

- 在光盘或档案中寻找并执行 setup 执行档
- 按画面照指示一步一步向下执行安装步骤，如图 1

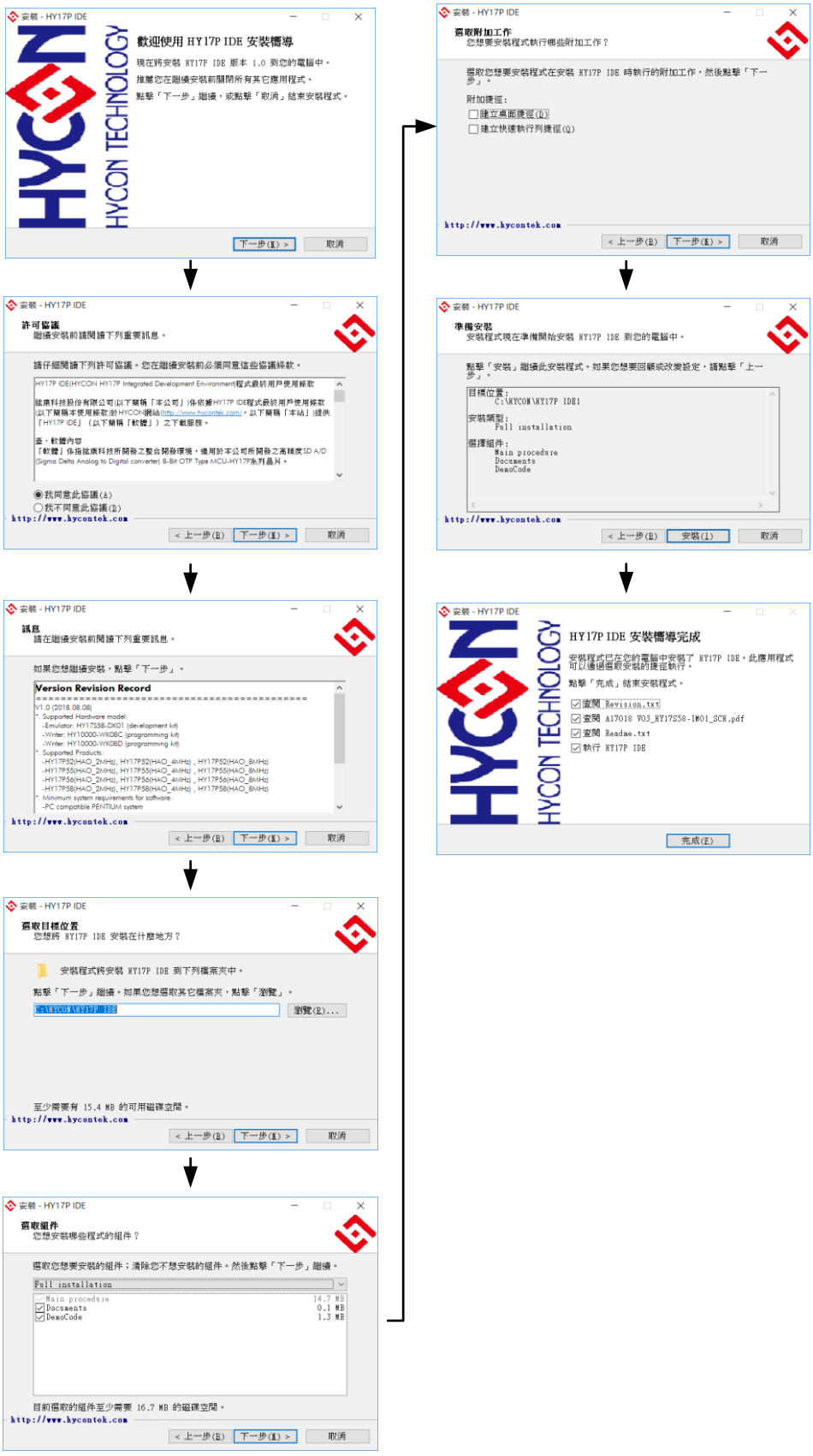

安裝 - HY17P IDE
歡迎使用 HY17P IDE 安裝精靈
現在將安裝 HY17P IDE 版本 1.0 到您的電腦中。
推薦您在繼續安裝前關閉所有其它應用程式。
點擊「下一步」繼續，或點擊「取消」結束安裝程式。
下一步(N) >
取消
選取附加工作
您想要安裝程式執行哪些附加工作？
選取您想要安裝程式在安裝 HY17P IDE 時執行的附加工作，然後點擊「下一步」。
附加捷徑：
建立桌面捷徑(D)
建立快速執行列捷徑(Q)
http://www.hycontek.com
< 上一步(B)
下一步(N) >
取消
許可協議
開始安裝前請閱讀下列重要訊息。
請仔細閱讀下列許可協議。您在繼續安裝前必須同意這些協議條款。
HY17P IDE(HYCON HY17P Integrated Development Environment)程式最終用戶使用條款
我同意此協議(A)
我不同意此協議(D)
http://www.hycontek.com
< 上一步(B)
下一步(N) >
取消
準備安裝
安裝程式現在準備開始安裝 HY17P IDE 到您的電腦中。
點擊「安裝」繼續此安裝程式。如果您想要回顧或改變設定，請點擊「上一步」。
目標位置：
C:\HYCON\HY17P IDE1
安裝類型：
Full installation
選擇組件：
Main procedure
Documents
DemoCode
< 上一步(B)
安裝(I)
取消
訊息
請在繼續安裝前閱讀下列重要訊息。
如果您想繼續安裝，點擊「下一步」。
Version Revision Record
V1.0 (2018.08.08)
* Supported Hardware model:
-Emulator: HY17S58-DK01 (development kit)
-Writer: HY10000-WK08C (programming kit)
-Writer: HY10000-WK08D (programming kit)
* Supported Products:
-HY17P52(HAO_2MHz), HY17P52(HAO_4MHz) , HY17P52(HAO_8MHz)
-HY17P55(HAO_2MHz), HY17P55(HAO_4MHz) , HY17P55(HAO_8MHz)
-HY17P56(HAO_2MHz), HY17P56(HAO_4MHz) , HY17P56(HAO_8MHz)
-HY17P58(HAO_2MHz), HY17P58(HAO_4MHz) , HY17P58(HAO_8MHz)
* Minimum system requirements for software:
-PC compatible PENTIUM system
HY17P IDE 安裝精靈完成
安裝程式已在您的電腦中安裝了 HY17P IDE。此應用程式可以通過選取安裝的捷徑執行。
點擊「完成」結束安裝程式。
查閱 Revision.txt
查閱 A17018 V03_HY17S58-IM01_SCH.pdf
查閱 Readme.txt
執行 HY17P IDE
完成(F)
選取目標位置
您想將 HY17P IDE 安裝在什麼地方？
安裝程式將安裝 HY17P IDE 到下列檔案夾中。
點擊「下一步」繼續。如果您想選取其它檔案夾，點擊「瀏覽」。
C:\HYCON\HY17P IDE
瀏覽(R)...
至少需要有 15.4 MB 的可用磁碟空間。
選取組件
您想安裝哪些程式的組件？
選取您想要安裝的組件；清除您不想安裝的組件。然後點擊「下一步」繼續。
Full installation
Main procedure 14.7 MB
Documents 0.1 MB
DemoCode 1.3 MB
目前選取的組件至少需要 16.7 MB 的磁碟空間。

图 1

卸载

请到控制面板的”卸载或更改程序”寻找 HY17P IDE 选择卸载程序，即可。

1.4. Demo Code 导入说明

- 开启 C:\HYCON\HY17P IDE\DemoCode 主程序
- 设定为组译主档
- 组译并进行除错

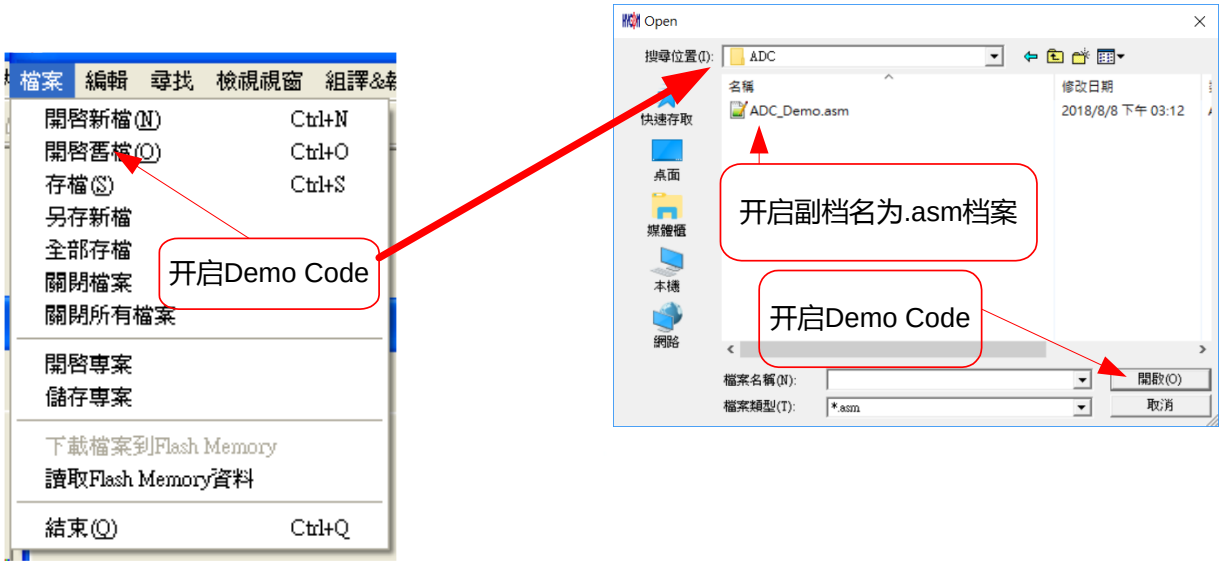


图 2

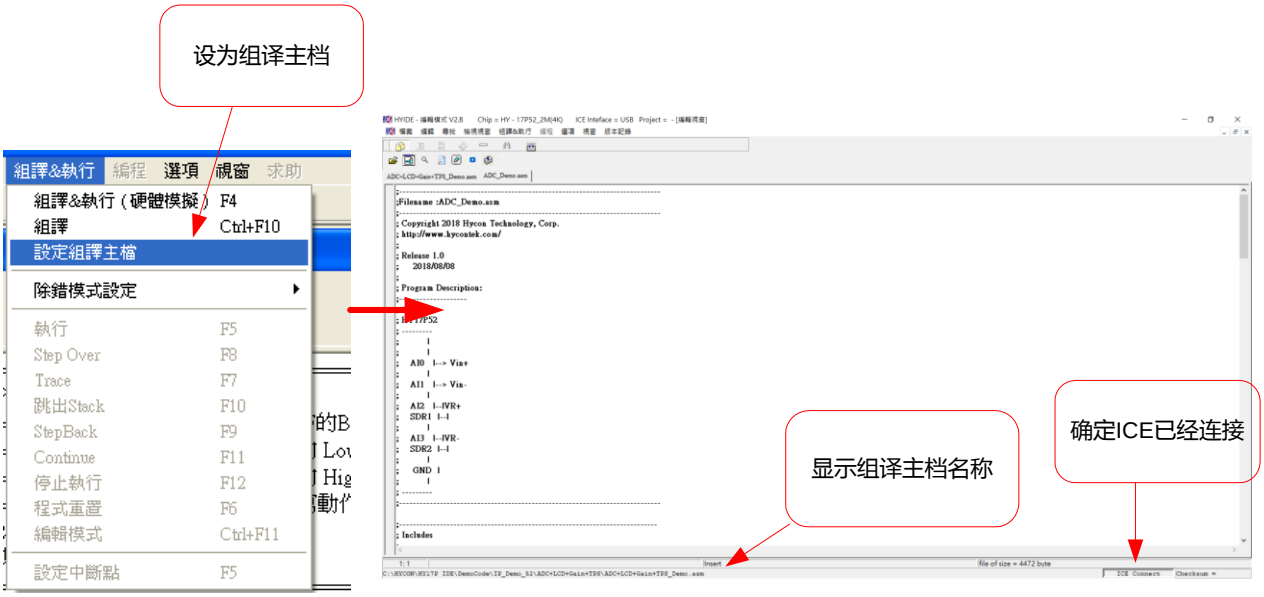


图 3

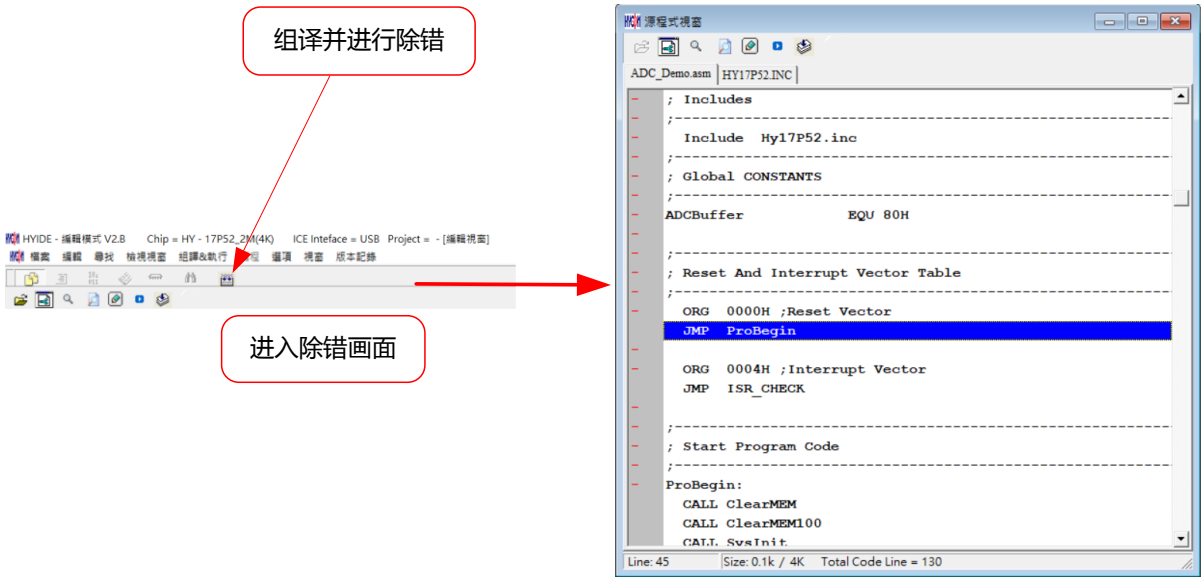


图 4

- 使用者可使用任何编辑器，来编辑 Source Code，只要能以 ASCII Code 的形式储存及可。程序组译时，会重新载入 Source Code 以确保程序正确性。下面章节将一一介绍除错与编辑的功能，组译并进行除错。

1.5. Demo Code 操作方式及使用

- 执行 HY17P IDE 软件安装后，于目录 C:\HYCON\HY17P IDE\DemoCode 主程序 下会有提供 Demo Code 供使用者参考。

2. HY17P IDE 介面说明

2.1. HY17P IDE 编辑介面

图 5

2.1.1. 编辑视窗

- 开启旧档

  开启存放在磁碟中已编辑好的档案。
- 设定标签

  设定标签，当开启档案很多时，可利用此项快速回到设定标签处。
- 跳至标签

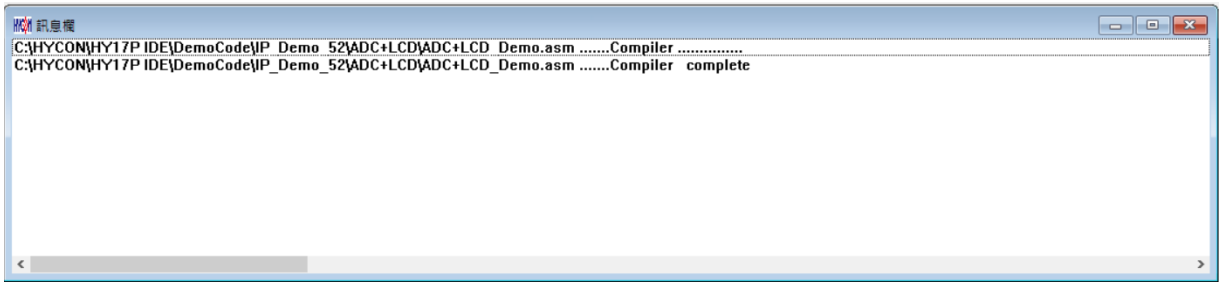
  跳到已设定的标签处。
- 寻找字符串

  寻找已输入过的字符串。
- 寻找下一个字符串
- 切换显示页面

  当开启档案很多时，可利用此项切换档案。
- 组译

  只有组译，不进入除错状态。

  当组译完成后会出现讯息栏



2.1.2. 档案

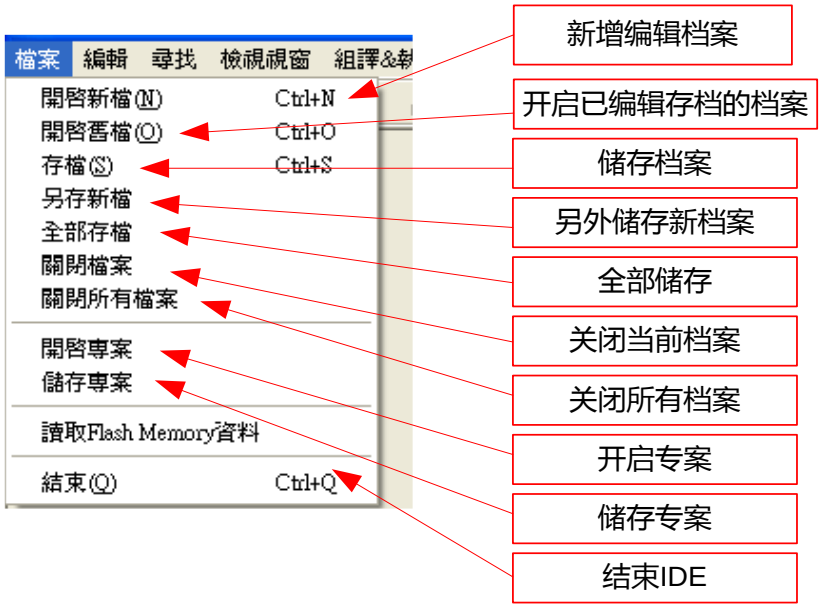


图 6

- 开启新档 → 新增编辑档案
- 开启旧档 → 开启已储存的编辑档案
- 存档 → 储存档案
- 另存新档 → 将档案用新的名称储存
- 全部储存 → 储存全部档案
- 开启项目 → 项目包括 (芯片型号、IDE 介面、组译主档名称、当前开启的状态、Checksum)，开启项目后会载入项目的状态。
- 储存项目 → 储存项目
- 结束 → 结束 Hycon-IDE 程序

2.1.3. 编辑

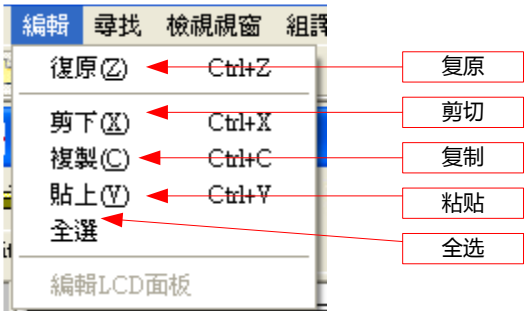


图 7

- 复原 → 回复上一次键入或删除
- 剪下 → 剪下选取的区域

- 复制 → 复制选取的区域
- 贴上 → 贴上复制的区域
- 全选 → 全部选择

2.1.4. 检视视窗

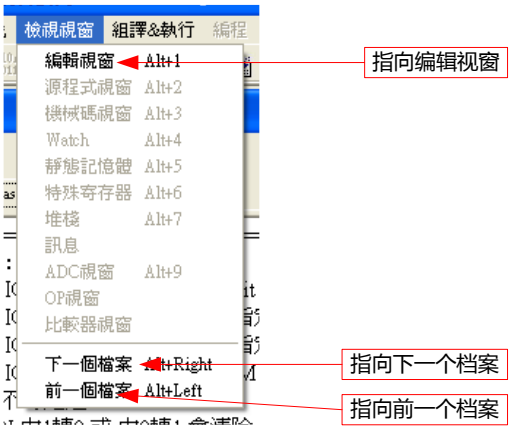


图 8

- 编辑视窗 → 将编辑视窗指定为当前的活动视窗
- 下一个档案 → 将下一个档案指定为当前的活动视窗
- 前一个档案 → 将前一个档案指定为当前的活动视窗

2.1.5. 组译&执行

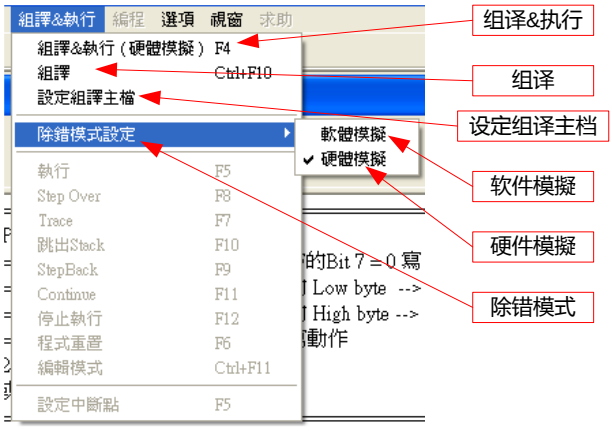


图 9

- 组译&执行 → 组译 Source Code 并执行除错模式
- 组译 → 只组译程序，不执行除错，此项组译并不会根据芯片型号产生错误讯息，只有当语句有误时才会显示错误讯息，一般用于制作 OBJ Code (Object)。
- 设定组译主档 → 设定为组译主档，Compiler 产生的文件名称如 Hex、MAP、ASC… 都将以此名称做为文件名称。

- 除错模式设定 → 选择使用软件除错或硬件除错。

2.1.6. 介面设定

图 10

- **介面设定(由选项中选取)**

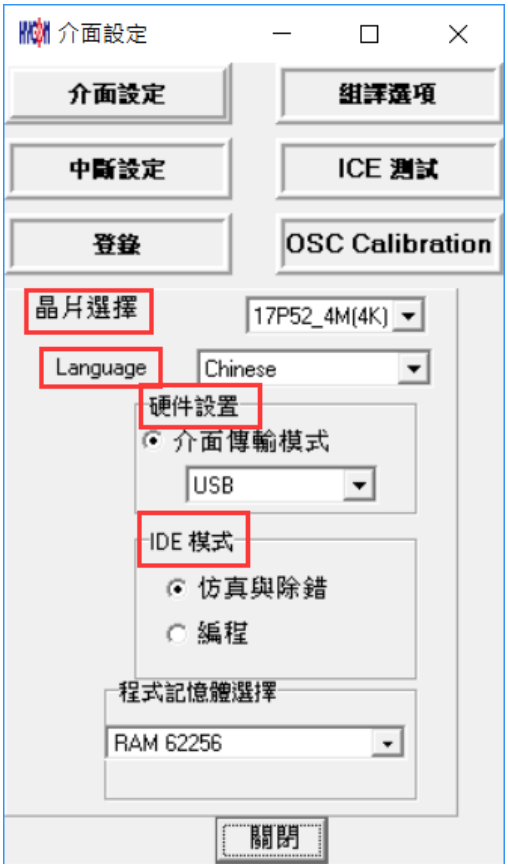


图 11

- 芯片选择 ： 选择芯片型号，Compiler 会根据选择的型号组译出烧录设定档案，并判断是否有误用到不存在的 Register 或 SRAM，或程序是否超出 ROM Size。
- Language： 可选择英文或中文介面。
- 软件设置 ： 选择 USB 传输介面。
- IDE 模式 ： 仿真与除错、烧录编程两种选择。

- **组译选项**

图 12

- 组译产生附档 ： 可选择输出以下档案
  - 二进制档 ：Hex
  - obj 档 ：obj
  - List 档 ：lst
  - ASCII 档 ：asc
- 堆栈操作 ： 依不同芯片型号选择 ,可选择堆栈满后重置 ,Stack over Stop 功能，当选择此项时，Compiler 会加入到 Hex 中，烧录时会将此设定烧入 OTP 的设定中。
- 烧录次数限制 ： 参考烧录视窗之介面设定章节。
- 编辑功能字型选择 ： 选择编辑器的字号。
- 未使用区域填满 ： 程序中未使用到的区域选择填满 0x0000 或 0xFFFF。
- 组译精简 ： 选择是否启动精简组译，当 JMP 或 CALL 小于 2K 时，会自动转换成 RJ 或 RCALL；但如果 CALL 后面的参数有设定时则不会转换成 RCALL。
- 烧录保护 ： 烧录视窗之介面设定章节。

- **中断设定**

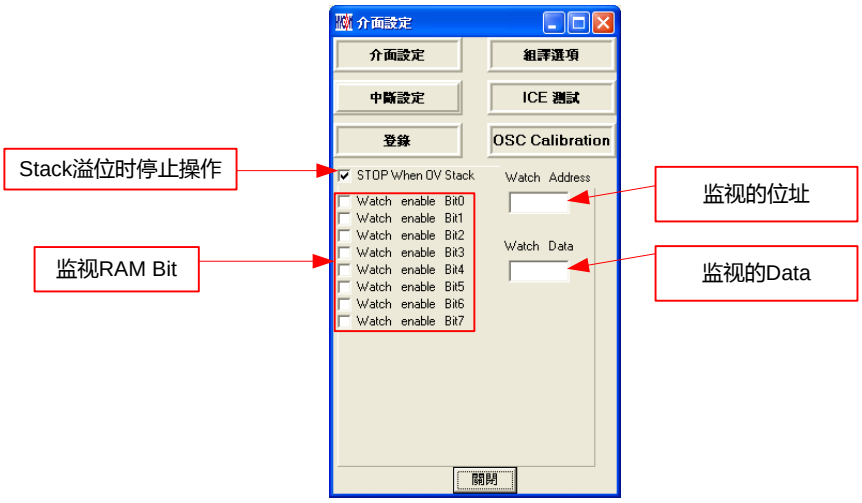


图 13

- Stack 溢位时停止操作 ： 当 Stack 溢位后 IDE 停止执行
- 监视的位址 ： 选择要监视 Register 或 RAM，当程序执行到 RAM 或 Register 的值与监视的 Data 相等时程序停止。
- 监视的 Data : 当监视的 Data 填上后，表示要监视的值
- 监视 RAM bit : 当监视的 bit 勾选后，表示要启动监视功能，并且当 Data 值的 bit 与勾选的 bit 相等时程序停止。

- **ICE 测试**

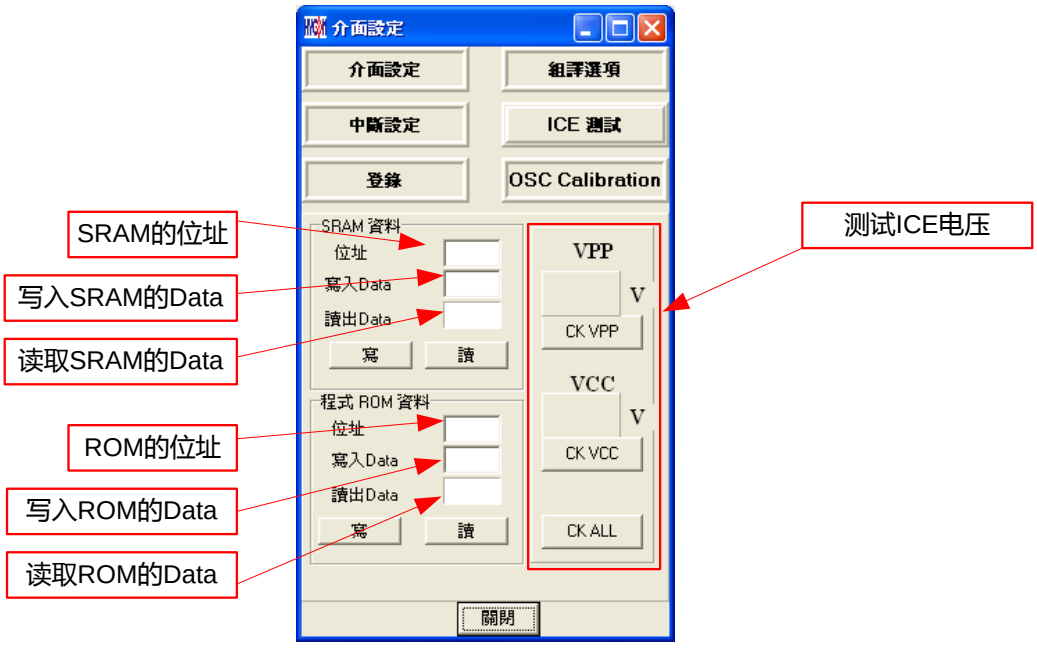


图 14

- **OSC calibration**

图 15

2.1.7. 视窗

可选择所有开启的视窗做垂直或水平的排列。

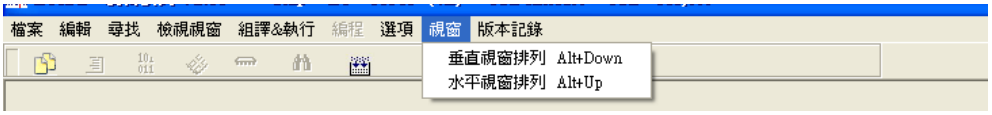


图 16

2.1.8. 程序架构

在开始编辑新的程序之前，须先由介面设定中设定芯片选择；

不同芯片搭配不同 Instruction Set，依芯片型号定义会区分有 H08A, H08B 指令集；

使用者一开始可以先参考软件所附的 demo code ,本文 1.6 章节有 demo code 导入说明，并搭配下列程序架构开始撰写程序，以下列出程序的基本架构说明：

- 程序名称定义为: ***.ASM
- 暂存器名称或 RAM Definition 定义为: ***.INC
- 如下，共有多个程序内容：
  - “Main.asm”、“Initial.asm”、“Interrupt.asm”、“Sub.asm”、“Mian.inc”、“H08.inc”

- “Main.asm” structure:

```
Include    17P.inc          ;程序名称可为任意名称
Include    Main.inc         ;HY17P 系列特殊暂存器名称、位址定义
ORG        00H              ;RAM 名称、位址定义
JMP        BEGIN            ;宣告程序开始
                            ;跳跃到主程序

ORG        04H              ;宣告中断旗标位置
Include    Interrupt.asm    ;引用”Interrupt.asm”中断子程序；
                            ;include 档案限制最多 100 个.

BEGIN:                      ;主程序开始. Label name 的定义可以为任意字
Include    Initial.asm      ;引用”Initial.asm”硬件初始化子程序
JMP        T1               ;跳跃到 T1 子程序
…
T1:
NOP

Include    Sub.asm          ;引用”Sub.asm”子程序
END                         ;程序结束
```

- 参考文件：

  IP 使用说明：

  Instruction Set 使用说明：H08A 指令集说明书

  Hycon-IDE Complier 使用说明：HY-MCU COMPILER

3. HY17P IDE 除错介面

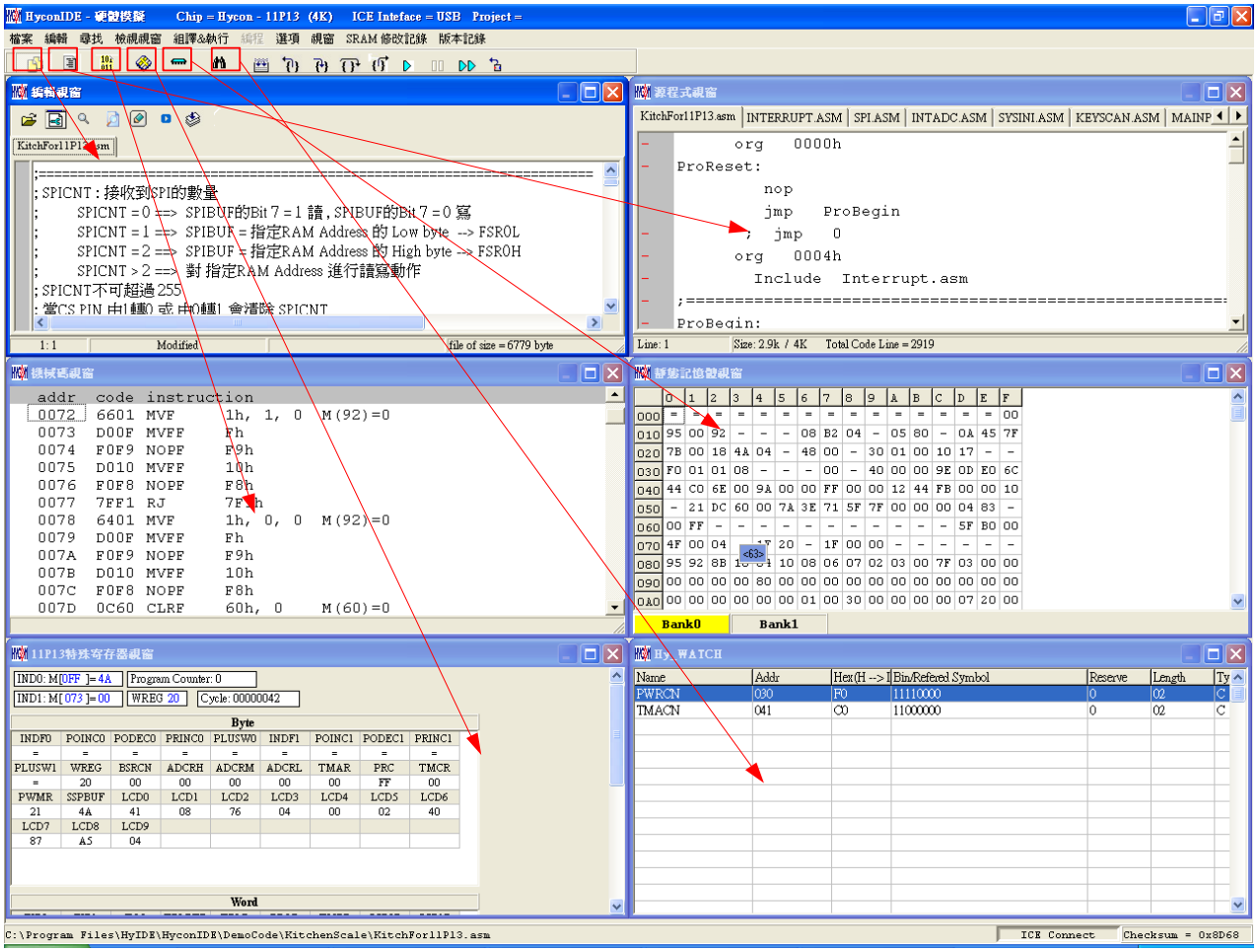
分为硬件除错与软件除错

- 硬件除错

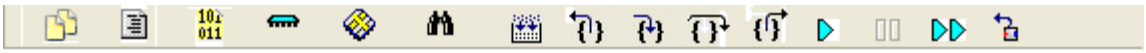
  指示栏棒为蓝色

- 软件除错

  指示栏棒为绿色

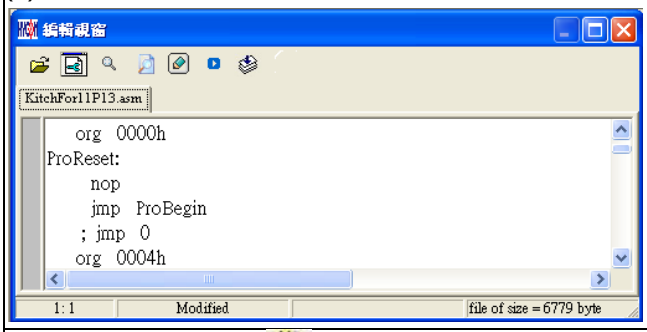


3.1. 快速执行

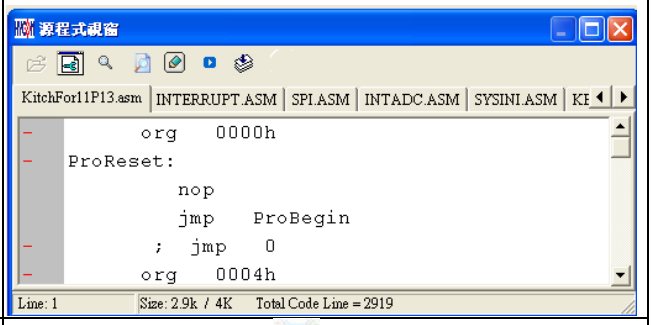


- **快速视窗切换**

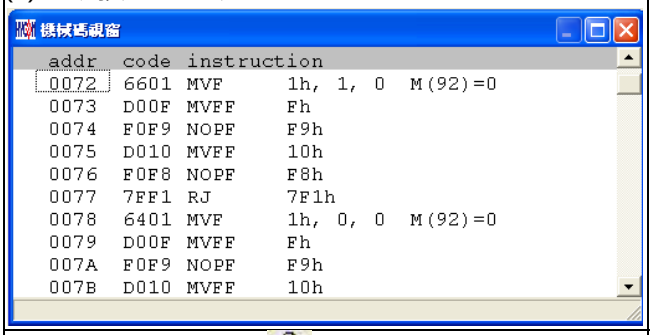
(1) 切换至 Edit 视窗



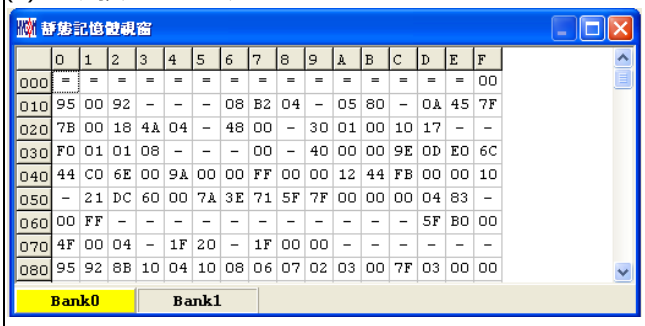
(2) 切换至 source 视窗



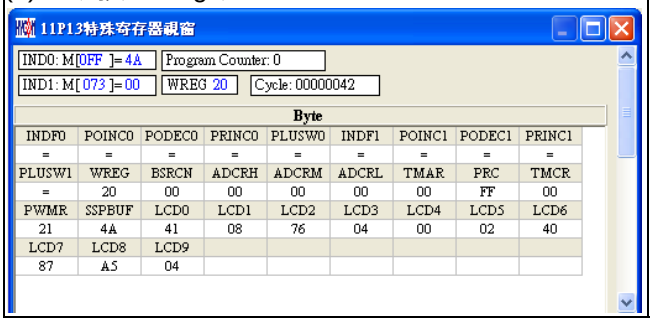
(3) 切换至 Hex 视窗



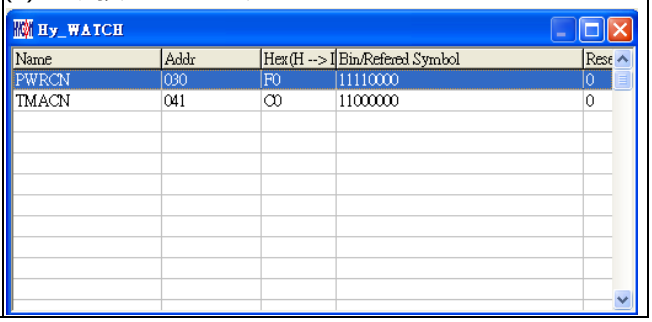
(4) 切换至 Ram 视窗



(5) 切换至 Reg 视窗



(6) 切换至 Watch 视窗



- **快速除错**

(1) 单步返回

(2) 单步执行(进入巨集/子程序)

(3) 单步执行(不进入巨集/子程序)

(4) 跳出 Call

(5) 执行(Free RUN)

(6) 暂停

(7) 连续执行

(8) 程序重置

(9) 返回编辑模式

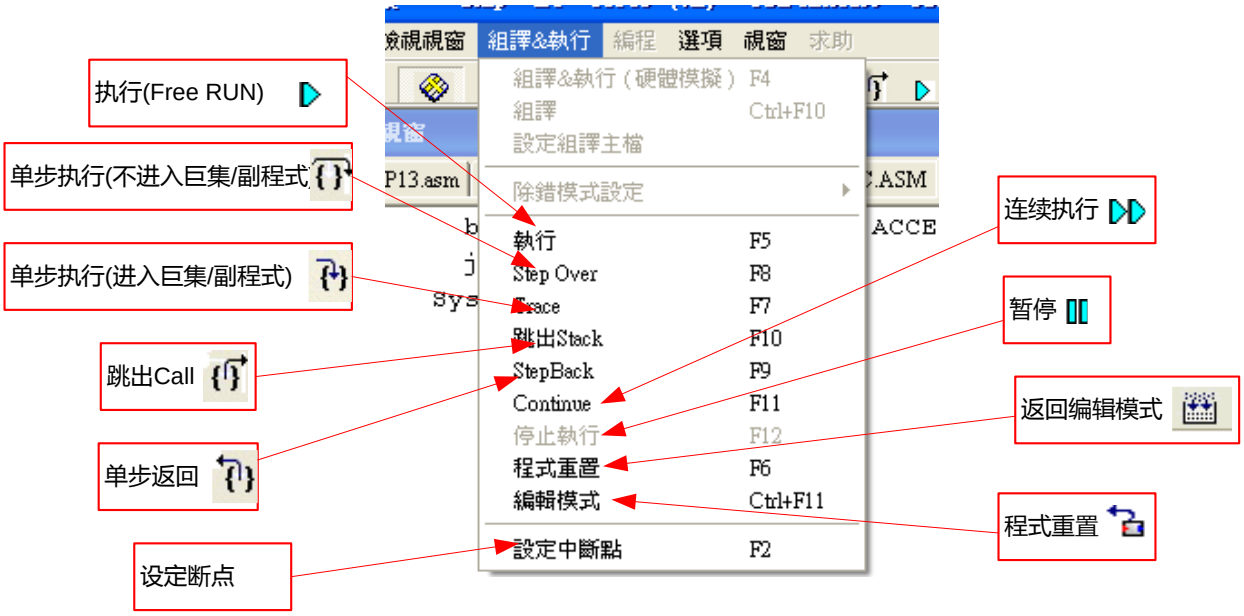


图 17

- **断点设定移除 2 种方式**

1. 在程序码视窗或是机械码视窗中将鼠标选择设置断点处，按键盘的”F2”键，即可设置或移除断点。
2. 在程序码视窗或是机械码视窗中将鼠标指向设置断点处，连续点击鼠标左键，即可设置或移除断点。

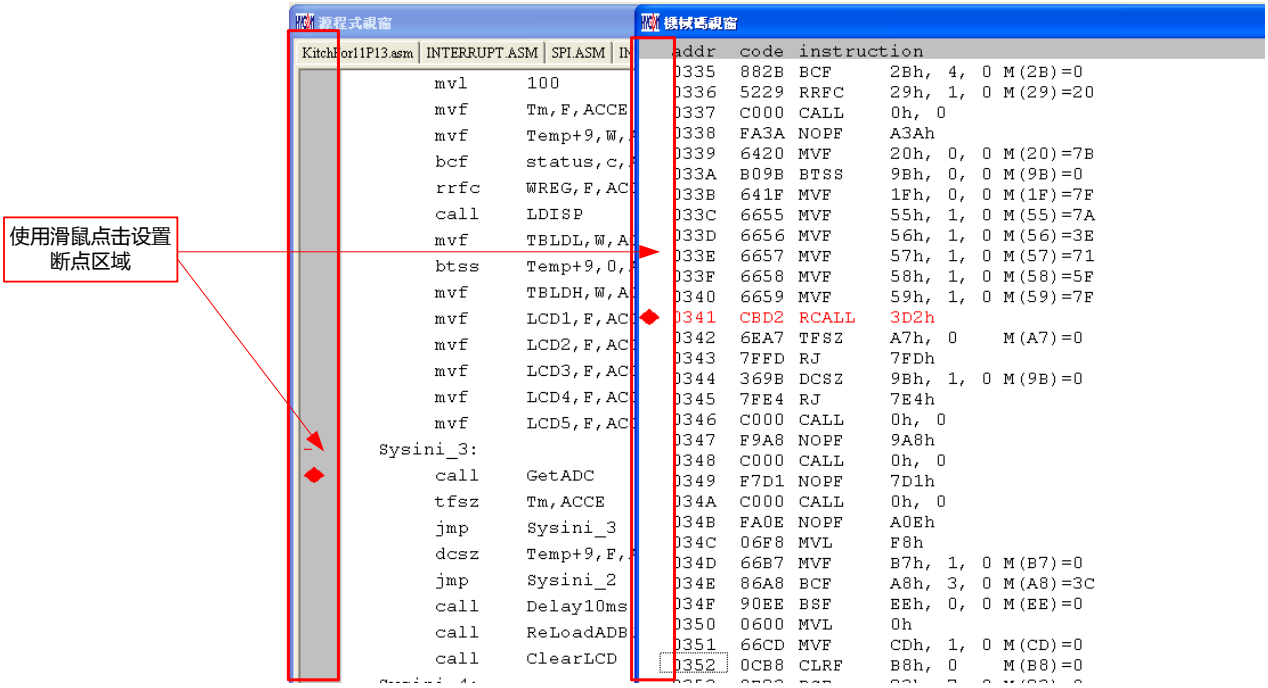

使用滑鼠点击设置
断点区域
源程式視窗
KitchFor11P13.asm
INTERRUPT.ASM
SPI.ASM
mvl 100
mvf Tm,F,ACCE
mvf Temp+9,W,
bcf status,c,
rrfc WREG,F,AC
call LDISP
mvf TBLDL,W,A
btss Temp+9,0,
mvf TBLDH,W,A
mvf LCD1,F,AC
mvf LCD2,F,AC
mvf LCD3,F,AC
mvf LCD4,F,AC
mvf LCD5,F,AC
Sysini_3:
call GetADC
tfsz Tm,ACCE
jmp Sysini_3
dcsz Temp+9,F,
jmp Sysini_2
call Delay10ms
call ReLoadADB
call ClearLCD
機械碼視窗
addr code instruction
0335 882B BCF 2Bh, 4, 0 M(2B)=0
0336 5229 RRFC 29h, 1, 0 M(29)=20
0337 C000 CALL 0h, 0
0338 FA3A NOPF A3Ah
0339 6420 MVF 20h, 0, 0 M(20)=7B
033A B09B BTSS 9Bh, 0, 0 M(9B)=0
033B 641F MVF 1Fh, 0, 0 M(1F)=7F
033C 6655 MVF 55h, 1, 0 M(55)=7A
033D 6656 MVF 56h, 1, 0 M(56)=3E
033E 6657 MVF 57h, 1, 0 M(57)=71
033F 6658 MVF 58h, 1, 0 M(58)=5F
0340 6659 MVF 59h, 1, 0 M(59)=7F
0341 CBD2 RCALL 3D2h
0342 6EA7 TFSZ A7h, 0 M(A7)=0
0343 7FFD RJ 7FDh
0344 369B DCSZ 9Bh, 1, 0 M(9B)=0
0345 7FE4 RJ 7E4h
0346 C000 CALL 0h, 0
0347 F9A8 NOPF 9A8h
0348 C000 CALL 0h, 0
0349 F7D1 NOPF 7D1h
034A C000 CALL 0h, 0
034B FA0E NOPF A0Eh
034C 06F8 MVL F8h
034D 66B7 MVF B7h, 1, 0 M(B7)=0
034E 86A8 BCF A8h, 3, 0 M(A8)=3C
034F 90EE BSF EEh, 0, 0 M(EE)=0
0350 0600 MVL 0h
0351 66CD MVF CDh, 1, 0 M(CD)=0
0352 0CB8 CLRF B8h, 0 M(B8)=0

图 18

3.2. RAM 视窗

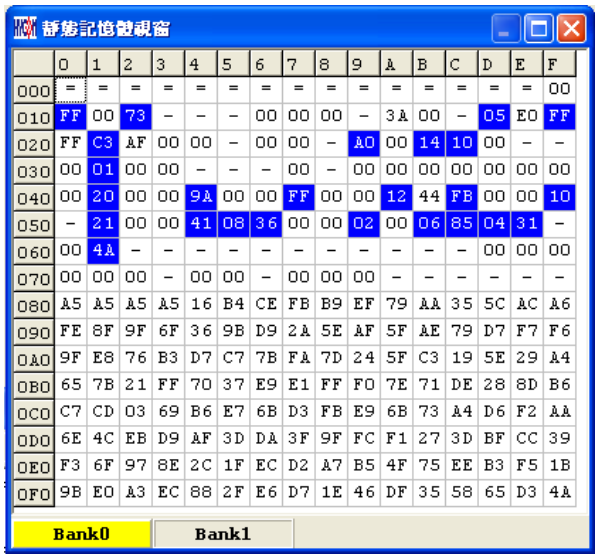


图 19

- 开启 RAM 视窗后 ,Bank 会根据所选择芯片显示其数量 ,每一个 Bank 有 256 byte。
- Bank0 由 0x00 ~ 0xFF，Bank1 由 0x100 ~ 0x1FF…。
- 如果该位址不存在，就会显示”-“。
- 如果要切换显示 Bank 可将鼠标指向欲显示的 Bank 区，按下鼠标确认(鼠标左键)。
- 如果该位址显示数字并有下底线，表示已设定 Hint。
- **注意 ：Bank0 的 Address 0x00 ~ 0x0E 为间接定址寄存器，无法直接更改，显示数值是不可参考的，如果要修改请参考 3.3 章节 :**修改间接定址 Data 或 Address

- **功能显示**

  按下鼠标选择键(鼠标右键)

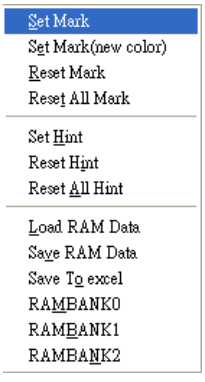


图 20

- 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 定义的 SRAM ,会在视窗中相对的位址自动产生 Hint ,当鼠标指标指向该位址，就会显示定义的字符串

例如 ： 程序定义 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
```

组译后进入除错状态，显示存储器视窗

当鼠标指向 80h 的位址，就会出现<80>:MD1

当鼠标指向 86h 的位址，就会出现<86>:MD4[0]

当鼠标指向 87h 的位址，就会出现<87>:MD4[1]

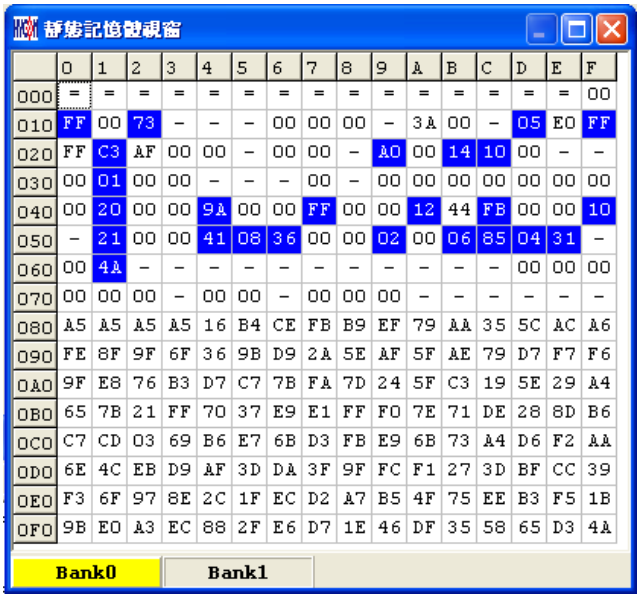


图 21

- **修改 SRAM 的值**有两种方式
  1. 将鼠标指向修改的地方，点击一下鼠标左键，用键盘直接 Key IN。
  2. 将鼠标指向修改的地方，连续点击两下鼠标左键出现图 22，使用键盘 Key In 或鼠标点击

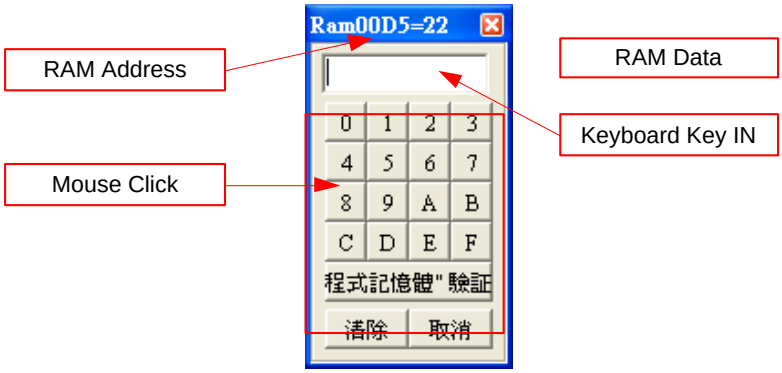


图 22

3.3. Register 视窗

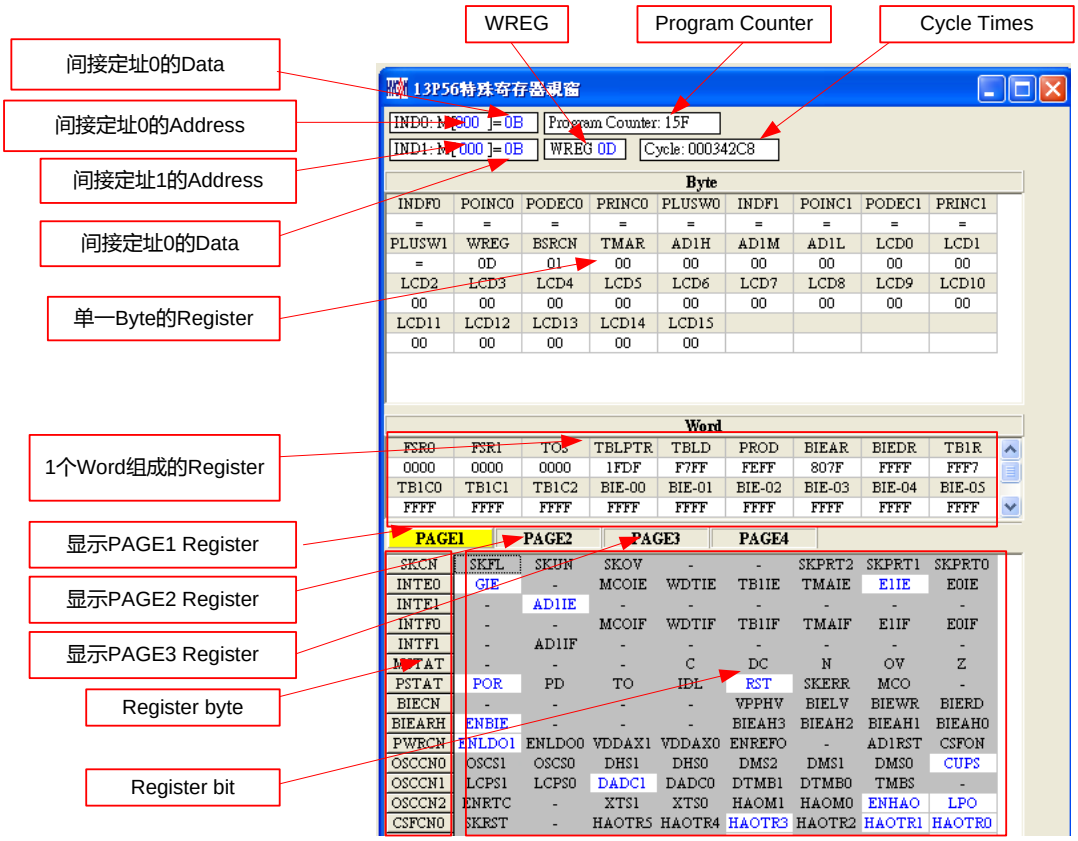


图 23

- 修改间接定址 Data 或 Address

  如图 24 设定后直接使用键盘 Key IN 或使用鼠标点选数值及可修改 Address

图 24

如图 25 设定后直接使用键盘 Key IN 或使用鼠标点选数值及可修改 Data

图 25

- 修改 WREG 的 Data

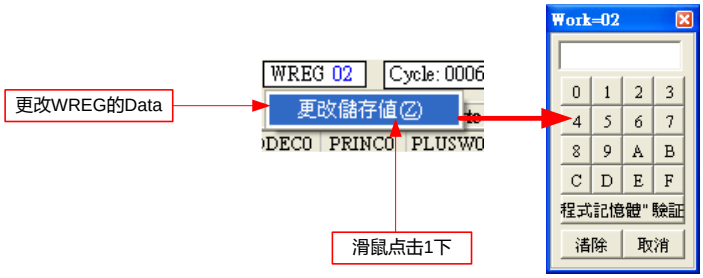


图 26

- 修改单 1byte 或 Word Register 的 Data

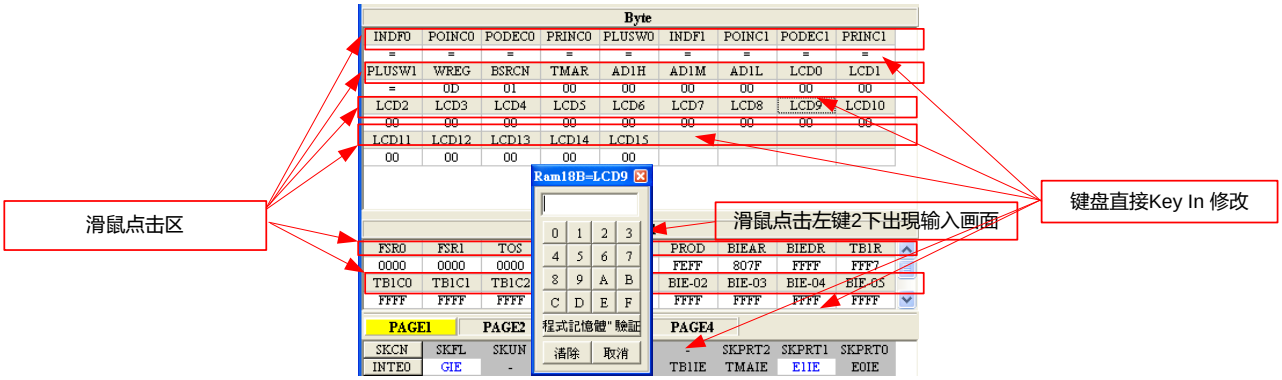


图 27

- 修改设置 Register 单 1 byte 或单 1 bit

  Bit 设置 1 后，该显示为反白，蓝色字

  Bit 设置 0 后，该显示为背景色，黑色字

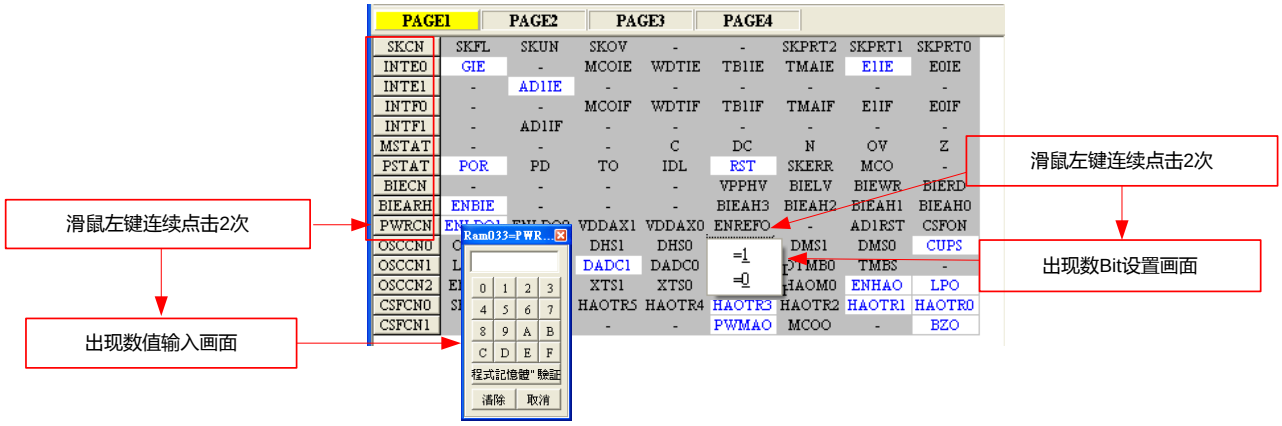


图 28

3.4. Watch 视窗

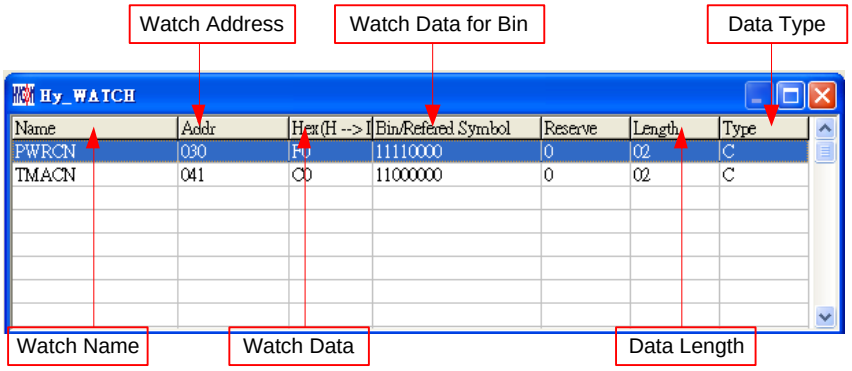


图 29

- Watch Name → 监看 Data 的名称，程序使用 EQU 或 DS 定义的名称
- Watch Address → 监看 Data 的 Address
- Watch Data → 显示数值，可以选择由右到左或是由左到右排列，也可显示十或十六进制显示

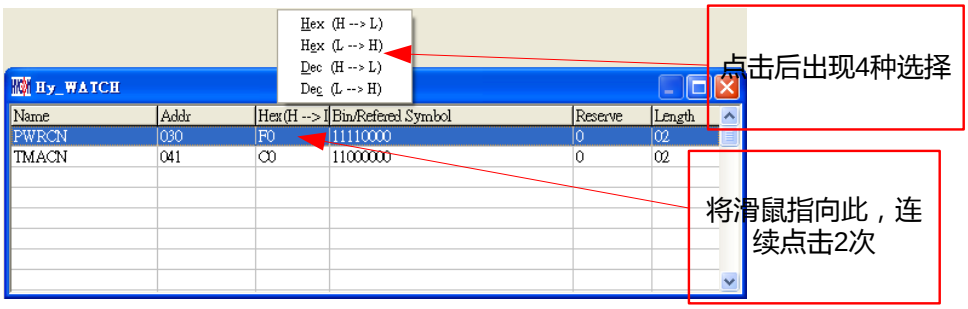


图 30

Hex (H → L)：十六进制显示，位址 H/L 由低至高

Hex (L → H)：十六进制显示，位址 L/H 由高至低

Dec (H → L)：十进制显示，位址 H/L 由低至高

Dec (L → H)：十进制显示，位址 L/H 由高至低

- Watch Data for Bin → Data 以二进制显示，只有用 EQU 定义的 Address 才有
- Data Length → Data 的长度，显示 DS 定义的长度；如果用 EQU 定义时，此数值显示 2
- Data Type → Data 的形式；D = DS 定义; C = EQU 定义

监看 EQU 所定义的 Register 或 RAM，按下鼠标右键选择要加入监看的 Register 或 RAM 如图 31

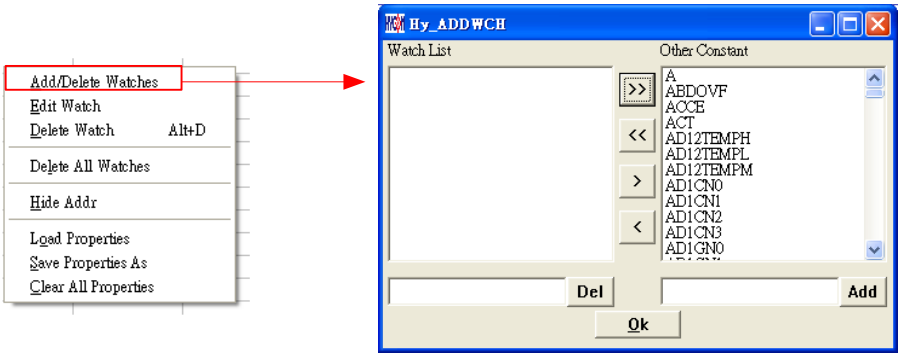


图 31

3.5. 堆栈视窗

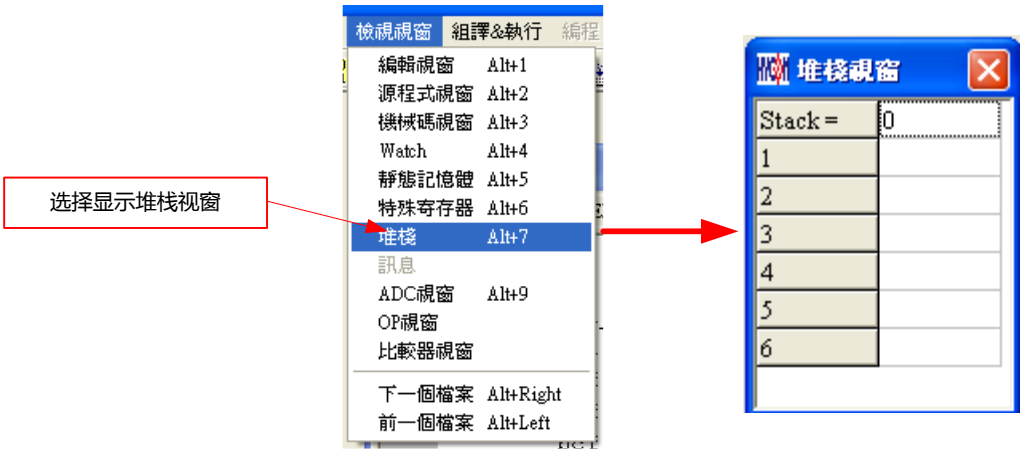


图 32

图 33

3.6. 暂存器修改记录

进入模拟视窗后(软件模拟或硬件模拟)，凡是暂存器或 SRAM 经过手动修改过(无论经由任何视窗修改 RAM、Register、ADC、OP 及 CMP)，就会被记录起来，当按下”SRAM 修改记录”后就会显示出来，此时视窗会停驻在此画面中直到将此画面关闭才能继续执行任何动作。

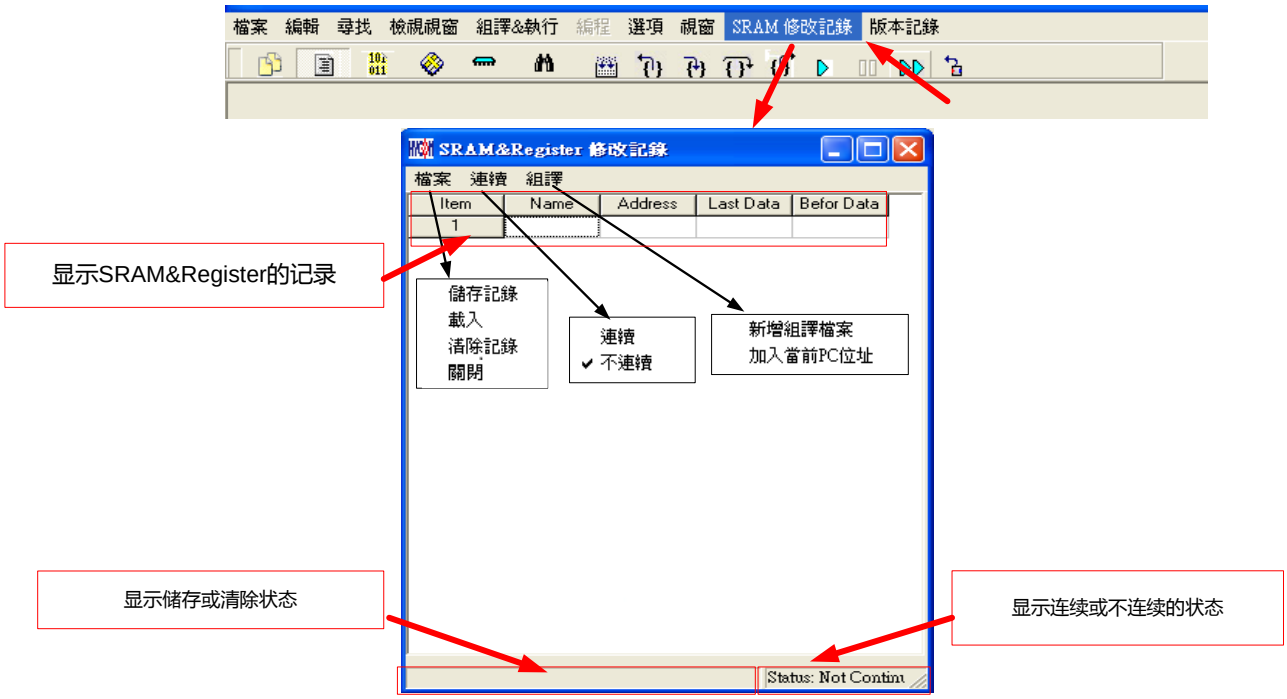


图 34

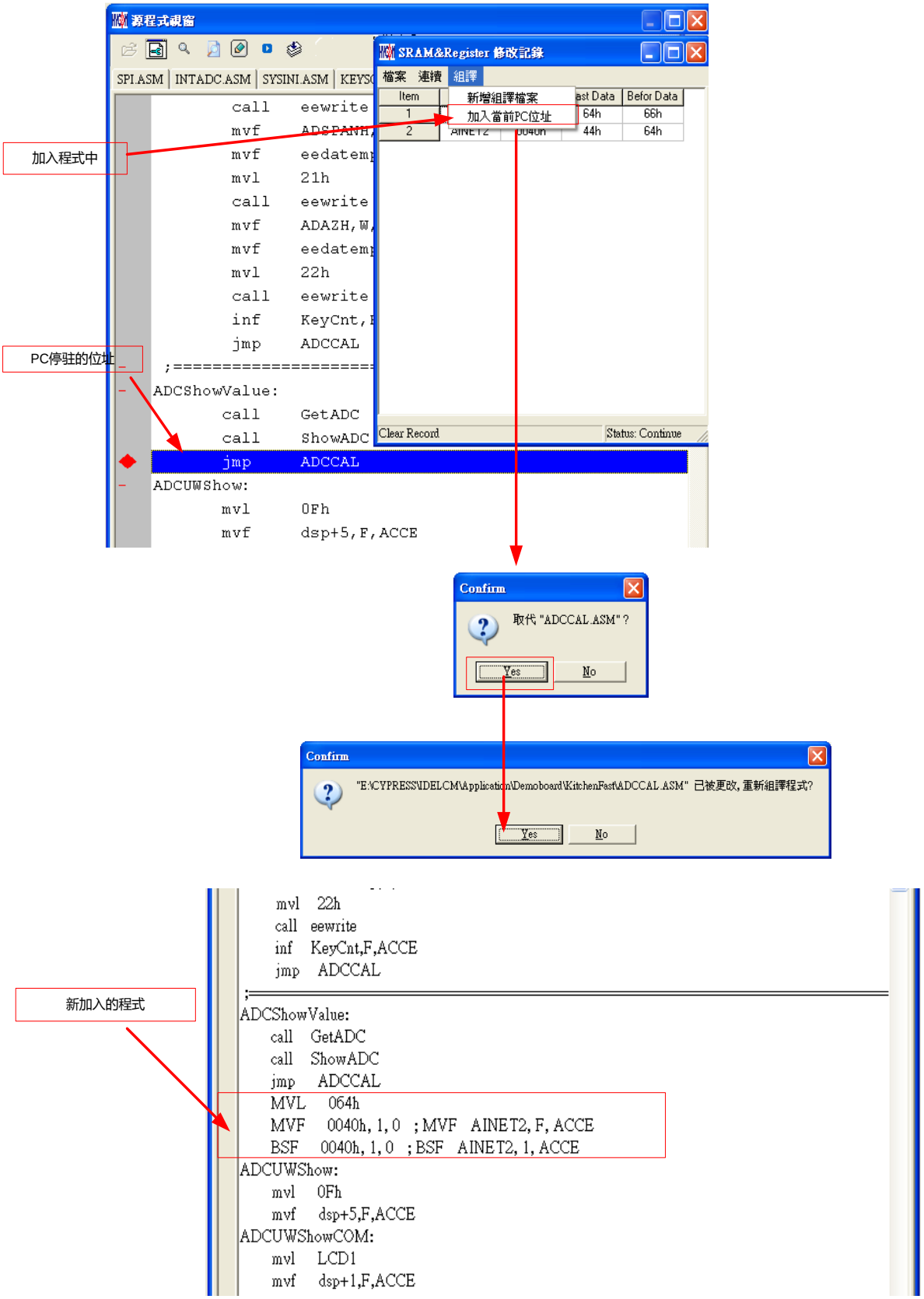

加入程式中
PC停驻的位址
新加入的程式
源程式視窗
SRAM&Register 修改記錄
檔案 連續 組譯
新增組譯檔案
加入當前PC位址
Item
Befor Data
64h
66h
44h
64h
SPI.ASM
INTADC.ASM
SYSINI.ASM
call eewrite
mvf ADSPANH,
mvf eedatem
mvl 21h
call eewrite
mvf ADAZH,W,
mvf eedatem
mvl 22h
call eewrite
inf KeyCnt,F
jmp ADCCAL
;==========
ADCShowValue:
call GetADC
call ShowADC
jmp ADCCAL
ADCUWShow:
mvl 0Fh
mvf dsp+5,F,ACCE
Clear Record
Status: Continue
Confirm
取代 "ADCCAL.ASM"?
Yes
No
Confirm
"E:\CYPRESS\IDELCM\Application\Demoboard\KitchenFast\ADCCAL.ASM" 已被更改, 重新組譯程式?
Yes
No
mvl 22h
call eewrite
inf KeyCnt,F,ACCE
jmp ADCCAL
;==========
ADCShowValue:
call GetADC
call ShowADC
jmp ADCCAL
MVL 064h
MVF 0040h, 1, 0 ; MVF AINET2, F, ACCE
BSF 0040h, 1, 0 ; BSF AINET2, 1, ACCE
ADCUWShow:
mvl 0Fh
mvf dsp+5,F,ACCE
ADCUWShowCOM:
mvl LCD1
mvf dsp+1,F,ACCE

图 35

3.7. 源程序视窗下的 Hint 功能

在源程序(Source Code)视窗下，要知道 Register 或 SRAM 的值及 Address，可以将鼠标指向此 Register 或 SRAM 的名称，就可显示名称、位址及 Data。

只有在以下指令后面所带的参数下才有此功能;

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)。

- 当指令为位元操作时只有第一个参数才有效，如图 36
- 当指令为 BCF、BSF、BTSS、BTSZ 及 BTGF 时，当指向第一个参数显示 Byte 数值，当指向第二个参数显示该 Bit 的值(1 or 0) ，如图 37
- 当指令为 MVFF 时(不是 Macro) ，当指向第一个参数显示第一个参数数值，当指向第二个参数显示第二个参数数值 ，如图 38
- 如果参数为 INDF0、POINC0、PODEC0、PRINC0、 INDF1、POINC1、PODEC1、PRINC1 时，此时 Data 为 FSR0 或 FSR1 内的位址的 Data，如图 39
- 如果参数为 PLUSW0 或 PLUSW1 时，此时 Data 为 FSR0+WREG 或 FSR1+WREG 内的位址的 Data，如图 40

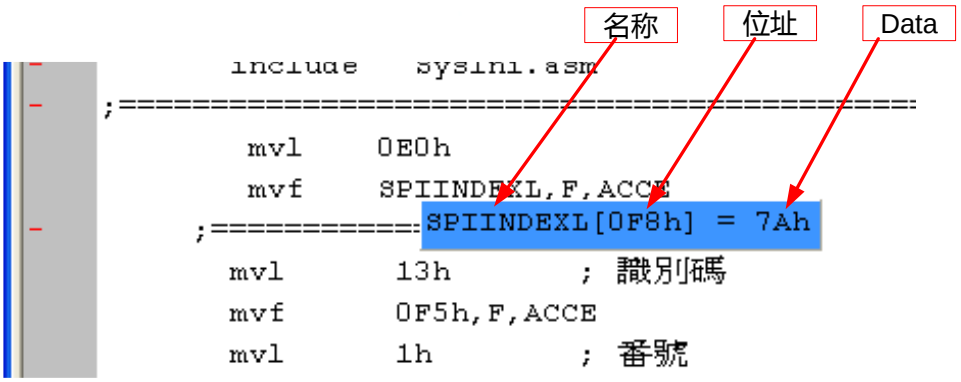


图 36

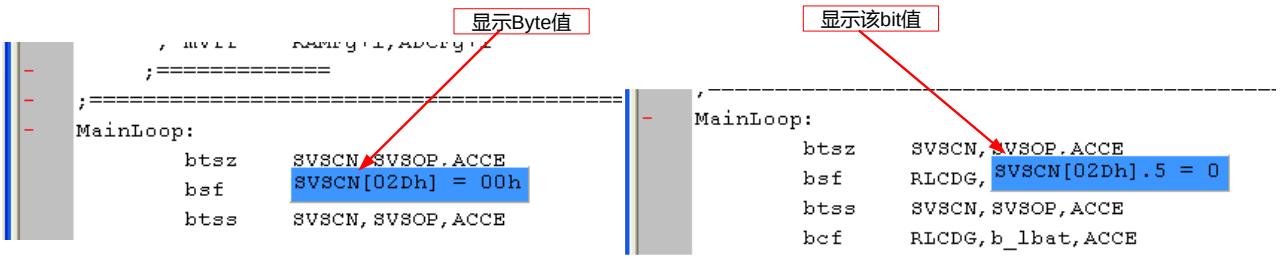


图 37

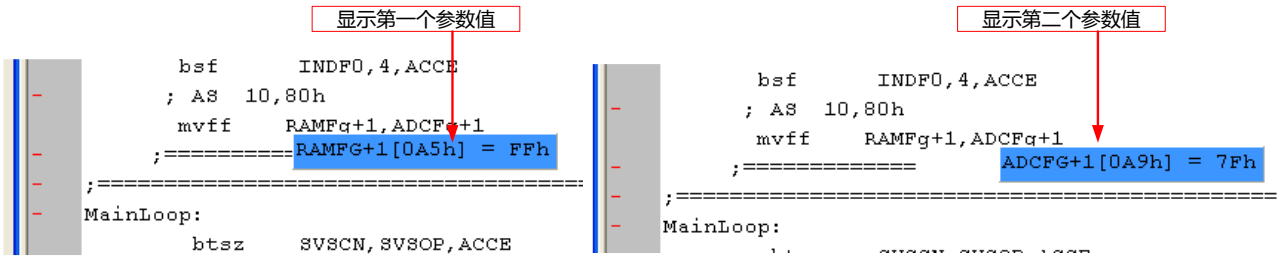

显示第一个参数值
bsf INDF0,4,ACCE
; AS 10,80h
mvff RAMFg+1,ADCFg+1
;==========
RAMFG+1[0A5h] = FFh
;==========
MainLoop:
btsz SVSCN,SVSOP,ACCE
显示第二个参数值
bsf INDF0,4,ACCE
; AS 10,80h
mvff RAMFg+1,ADCFg+1
;==========
ADCFG+1[0A9h] = 7Fh
;==========
MainLoop:

图 38

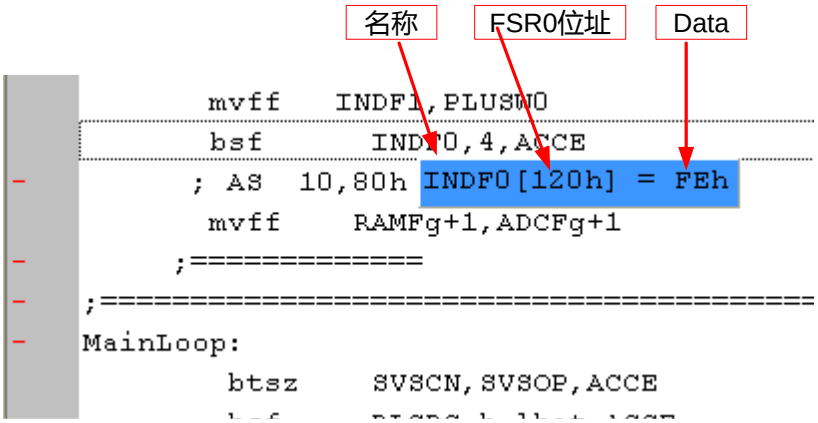

名称
FSR0位址
Data
mvff INDF1,PLUSW0
bsf INDF0,4,ACCE
; AS 10,80h INDF0[120h] = FEh
mvff RAMFg+1,ADCFg+1
;==========
;==========
MainLoop:
btsz SVSCN,SVSOP,ACCE

图 39

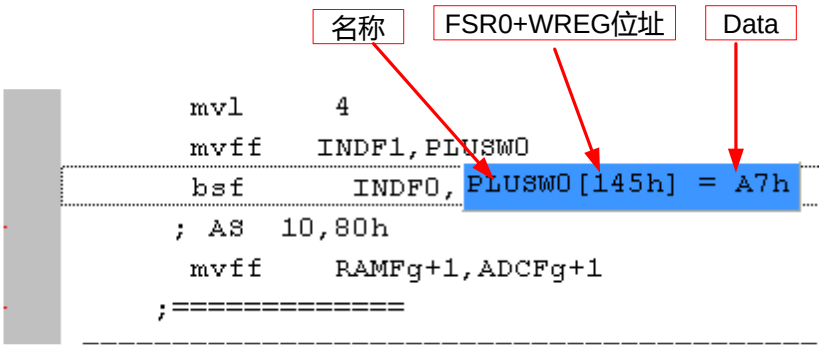

名称
FSR0+WREG位址
Data
mvl 4
mvff INDF1,PLUSW0
bsf INDF0, PLUSW0[145h] = A7h
; AS 10,80h
mvff RAMFg+1,ADCFg+1
;==========

图 40

4. 烧录视窗

4.1. 介面设定

要进入烧录视窗画面，点选”选项”，出现选择画面，点选”介面设定”，如图 41 所示。

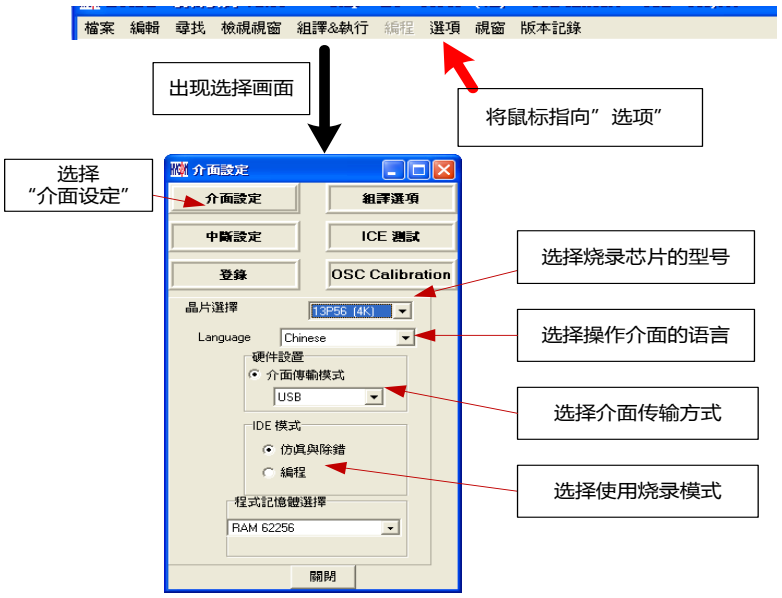


图 41

芯片选择 → 选择烧录芯片的型号 ,如果烧录芯片与选择的型号不同 ,则在 Blank Check、Program、Verify, 都会失败。

Language → 选择操作介面的语言，中文或英文。

硬件设置 → 可选择 USB 介面或 Parallel Port 介面。

IDE 模式 → 选择编程。

当介面设定完成后点选”组译选项”选择烧录的设定，如图 42。

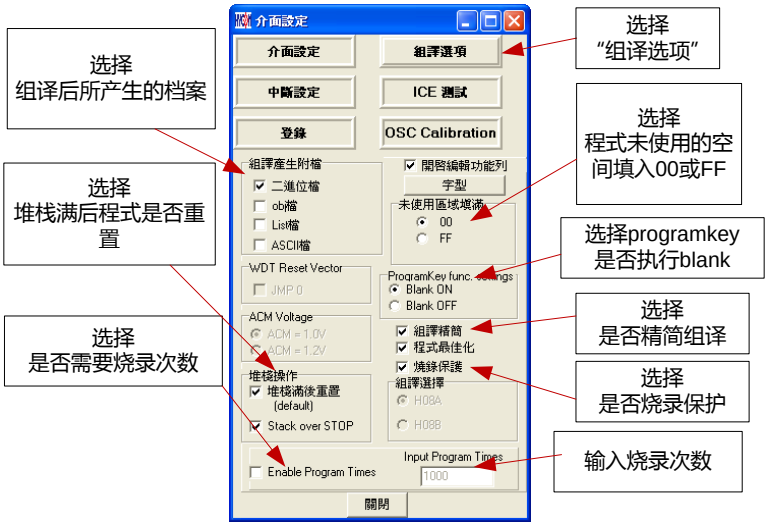


图 42

组译产生附档 → 选择组译程序后所产生的档案。
堆栈操作 → 选择当 OTP 程序运行后如果发生堆栈满或溢位是否要重置。
未使用区域填满 → 组译程序后，在未使用的程序空间选择填满 00 或 FF。
组译精简 → 选择是否要精简组译。
Enable Program Times → 选择是否启动 Download 的程序能被烧录的次数。
Input Program Times → 填入 Download 的程序能被烧录的次数(最大 2147483646，最小 1)。

当组译选项完成后点选”ICE 测试”进入测试电压是否正常，如图 43。

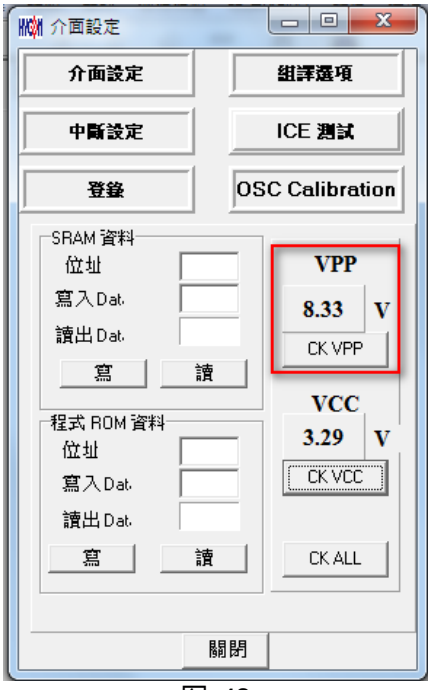


图 43

烧录时 VPP 的电压大约为 8.5V
烧录时 VDD 的电压大约为 4.5V

当测试电压测试完毕后，点选”OSC Calibration”进入软硬件频率校正

- **使用该功能前注意事项：**
  - 若启动软件 HAO/LPO 校正烧录 ,则芯片上电之后, RAM 0FEH/0FFH 位址资料有意义；
  - 单机烧录时间会增长约 500msec. (启动软件 LPO 校正烧录)；
  - 软件 HAO/LPO 校正功能并非校正实际频率，只是提供频率差异值供计算；
  - 在线烧录仅支援校正硬件 HAO，并不支援软件校正功能。

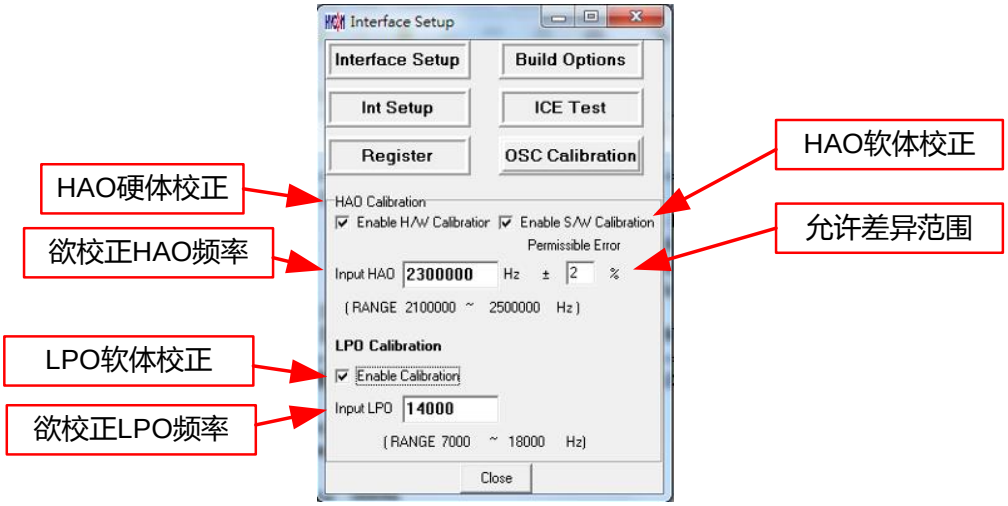


图 44

- HAO Calibration：
  - Enable H/W Calibration：启动硬件 HAO 校正功能，实际校正系统频率，此功能需在选择芯片型号后，再次确认是否可硬件校正。
  - Enable S/W Calibration：启动软件 HAO 差值校正功能，差值存放于 RAM **0FEH** 位址。
- LPO Calibration：
  - Enable S/W Calibration：启动软件 LPO 差值校正功能，差值存放于 RAM **0FFH** 位址。
- Input HAO or Input LPO：为欲校正频率数值。
- Permissible Error：校正后频率值与欲校正数值允许差异范围。

下面将说明软件校正：

- HAO Software Calibration：
  - 计算后频率差值存放于 RAM **0FEH** 位址 ;于芯片 Power on 时将差值写入 RAM 中，该动作并非实际校正频率源。
  - HAO Hardware Calibration, HAO Software Calibration 可同时存在，并以先执行 Hardware Calibration 之后再进行 Software Calibration 差值计算。
  - HAO 差值基频定义为 **4000HZ/LSB**。
  - **OFEH** 位址中资料格式为：
    - Bit7：0= +, 1= - ；Bit6~Bit0 代表差异频率值;
    - 01H 代表差异频率值为 +4000HZ；FFH 代表差异频率值为 -4000HZ;
  - Example：

HAO 欲校正 2000000HZ 频率，而实际芯片 HAO=1920000HZ，

则 (1920000-2000000)/4000 = -80000/4000= -20，因此该 RAM 0FEH 资料则为 **1110 1100b**

- Example1：

  HAO 欲校正 2000000HZ 频率，而实际芯片 HAO=2008000HZ，

  则(2008000-2000000)/4000 = 8000/4000=2，因此该 RAM 0FEH 资料则为 **0000 0010b**

- LPO Software Calibration：
  - 计算后频率差值存放于 RAM **0FFH** 位址 ;于芯片 Power on 时将差值写入 RAM 中，该动作并非实际校正频率源。
  - LPO 差值基频定义为 **64HZ/LSB**。
  - **OFFH** 位址中资料格式为：
    - Bit7: 0= +, 1= - ; Bit6~Bit0 代表差异频率值;
    - 01H 代表差异频率值为+64HZ；FFH 代表差异频率值为-64HZ;
  - Example:

    LPO 欲校正 28000HZ 频率，而实际芯片 LPO=28128HZ，

    则(28128-28000)/64 = 128/64 = 2，因此该 RAM 0FFH 资料则为 **0000 0010b**
  - Example1:

    LPO 欲校正 28000HZ 频率，而实际芯片 LPO=27872HZ，

    则(27872-28000)/64 = -128/64 = -2，因此该 RAM 0FFH 资料则为 **1111 1110b**

当介面设定完成后点选”关闭”，会将所设定的参数记录起来，下次开启此设定，会自动载入设定值，并在标题视窗显示设定烧录芯片型号，如图 45。

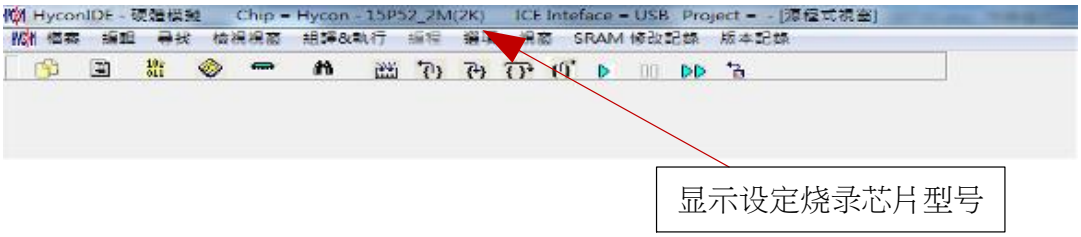


图 45

4.2. 操作步骤

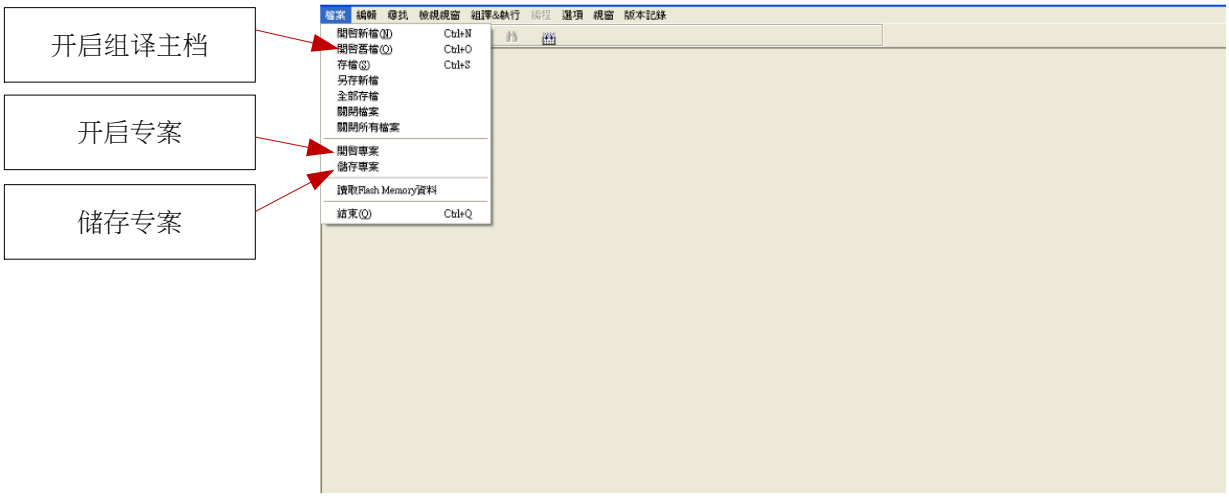


图 46

开启旧档 → 开启已经写好的源程序组译主档。

开启专案 → 开启储存的项目名称。

储存专案 → 储存已完成的项目。

4.2.1. 开启档案与组译

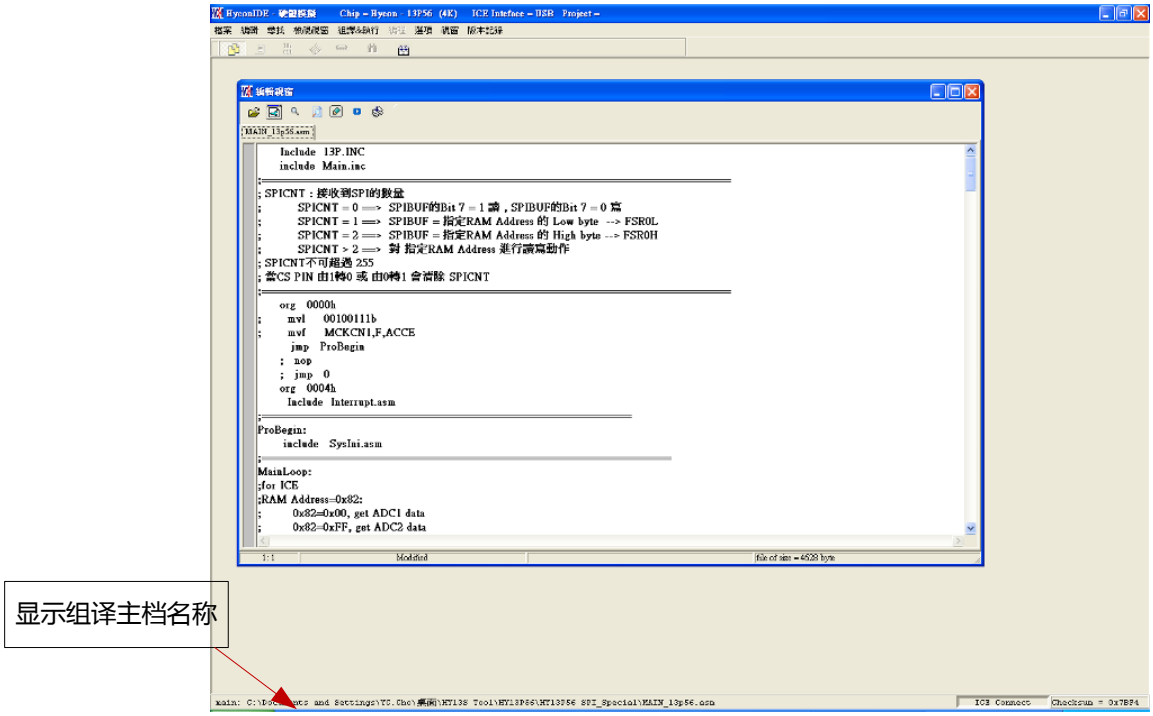


图 47

由开启档案将源程序的主档名称开启，并在显示组译主档名称下显示，如果显示名称与主档名称不同，将鼠标指向档案，按下鼠标右键，选择设为组译主档，如图 48。

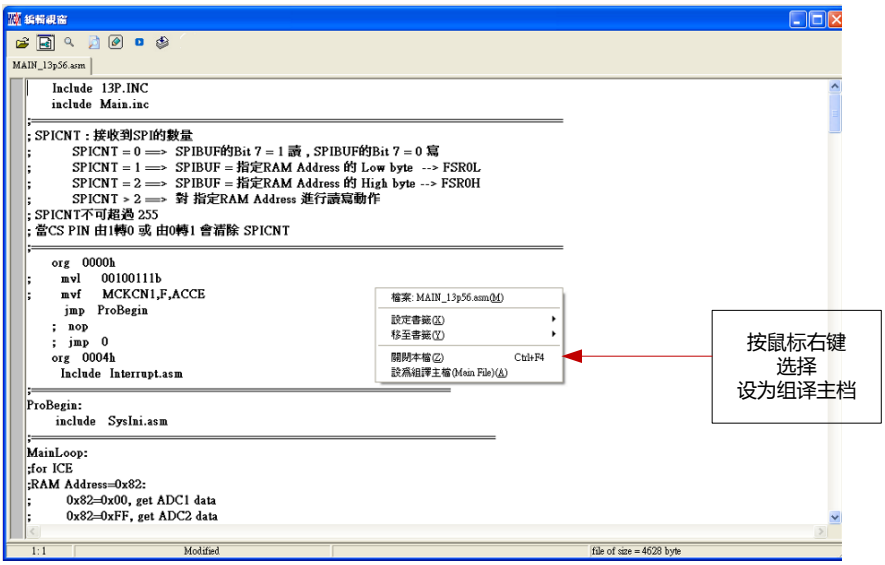


图 48

将 Source Code 组译并 Download 到烧录器或 IDE 的 Flash Memory，如图 49

图 49

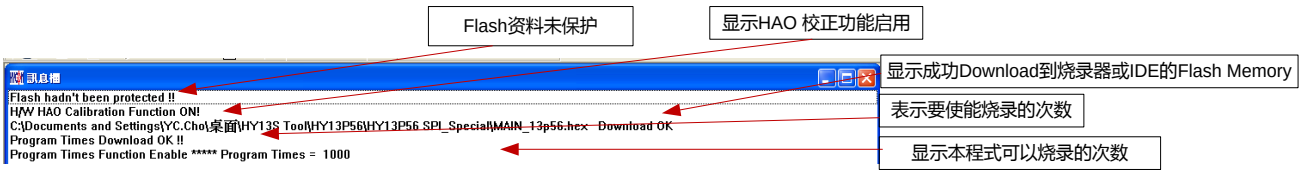


图 50

1. 当介面选择 USB，组译主程序完成后会将程序码，载入烧录器或 IDE 的 Flash Memory 内，作为生产线上量产烧录用。
2. 如果组译选项内有选择使能烧录次数，讯息栏位会显示程序可以烧录次数，如图 50。
3. 当组译完成后在下方显示组译完成后的 Hex 档名称与 Checksum，如图 51。

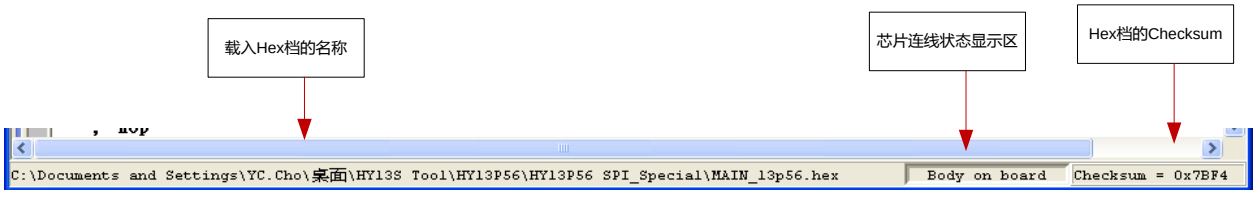


图 51

4.2.2. Download HEX File

如要 Download Hex File 请使用 HY17P-Hex Loader 软件并按照使用说明书操作。

4.3. PC 连线烧录 OTP

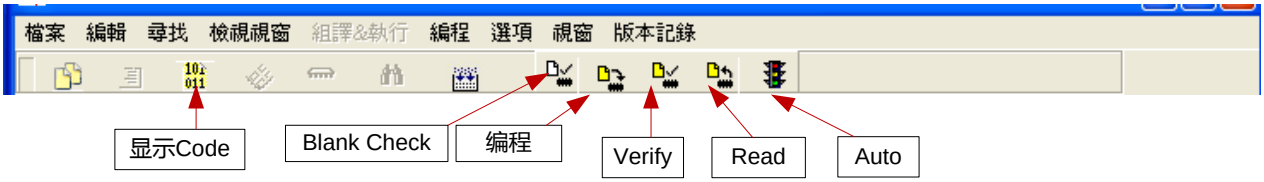


图 52

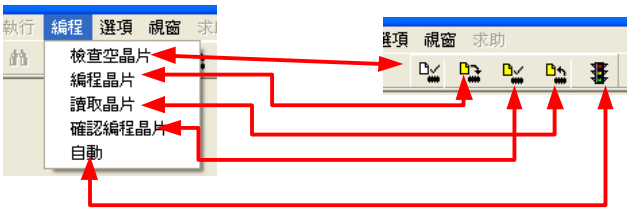


图 53

当烧录的档案成功的载入烧录器或 IDE 的 Flash Memory 内 ,将可以进行 Blank Check、烧录、Verify 及读取等动作，如果没有成功载入，则以上的动作将不会成功。

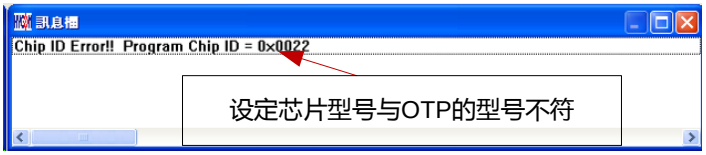


图 54

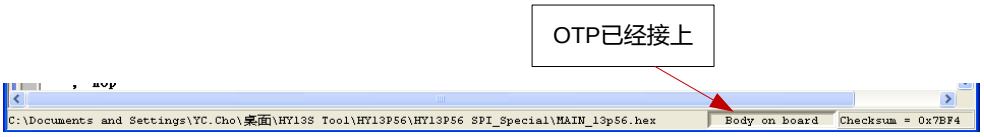


图 55

图 56

确定在标题视窗下所选择的烧录芯片型号 ,与 OTP 型号相同 ,当烧录器执行 Blank Check、烧录与 Verify，程序会比对设定选择芯片型号与烧录 OTP 型号是否相同，如果不同否则不会烧录到 OTP 内，在讯息栏内显示错误讯息如图 54。

在烧录之前如果想要确定型号是否正确，可以将鼠标指标指向”芯片连线状态显示区”上

按鼠标左键，如果芯片型号正确则显示如图 55；如果不正确则显示如图 56；如果有勾选 ”Enable Program Times” 则剩余烧录次数会显示于讯息栏内如图 57。

图 57

4.3.1. 芯片检查(Blank Check)

芯片检查(Blank Check) 图示为  ，在还没有烧录过的芯片，读取其内部的 Code 应该皆为 0xFFFF， 芯片检查的目的是确定此 OTP 所有位址的内容皆为 0xFFFF。检查芯片是否为空所指的是要烧录 OTP 位址的内容皆为 0xFFFF。如果选择芯片正确以及检查为空，讯息栏出现以下讯息(图 58)。

图 58

如果选择芯片不正确或是检查不为空，讯息栏出现以下讯息(图 59)。

图 59

4.3.2. 编程芯片(Program)

编程芯片(Program)图示为  ，编程的目的是将已经 Compiler 完成的程序烧录到 OTP 的芯片中，烧录完成后组装成品后，将可依照使用者所写的指令运行程序。

将已下载或组译完成的 Hex 档(显示于最下面的显示栏)，编程于选择芯片内，并确认编程芯片内容是否正确(步骤参考 4.2.1 或 4.2.2 一节)。

如果选择芯片正确以及编程成功，讯息栏出现以下讯息(图 60)，如果有勾选 ”Enable Program Times” 则允许烧录的次数会减 1，并将剩余烧录次数显示于讯息栏内。

图 60

如果选择芯片不正确以及编程不成功，讯息栏出现以下讯息(图 61)。

图 61

4.3.3. 确认编程芯片(Verify)

确认编程芯片(Verify)图示为 ，确认编程芯片的目的是在比对烧录到 OTP 芯片的程序是否与载入到烧录器的程序相同。

确认编程芯片内容是否与下载或组译完成的 Hex 档(显示于最下面的显示栏)一致，如果芯片已经编程保护，则此项无效或比对失败。

如果选择芯片正确以及确认编程成功，讯息栏出现以下讯息(图 62)。

图 62

如果选择芯片不正确以或确认编程不成功，讯息栏出现以下讯息(图 63)。

图 63

4.3.4. 读取芯片(Read)

读取芯片(Read)图示为 ，读取芯片的目的，是让使用者确认读取 OTP 的 Checksum 是否与烧录的Hex档相同。读取芯片内容(步骤如图64) ,并将内容显示于”显示Code”视窗内。

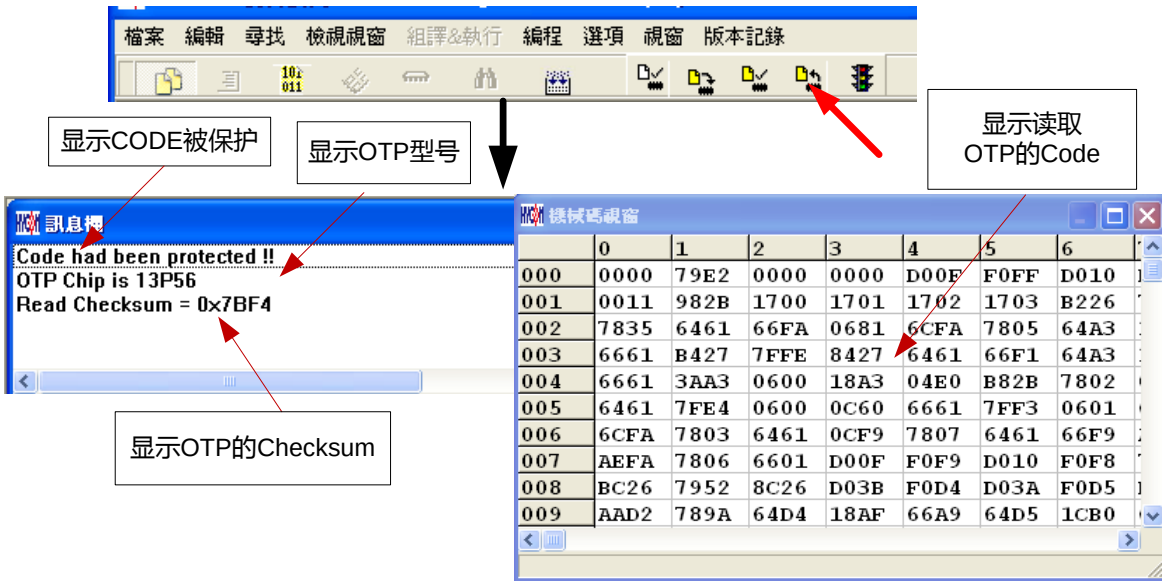

檔案 編輯 尋找 檢視視窗 組譯&執行 編程 選項 視窗 版本記錄
显示CODE被保护
显示OTP型号
显示读取
OTP的Code
訊息視窗
Code had been protected !!
OTP Chip is 13P56
Read Checksum = 0x7BF4
显示OTP的Checksum
機械碼視窗
0 1 2 3 4 5 6
000 0000 79E2 0000 0000 D00F F0FF D010
001 0011 982B 1700 1701 1702 1703 B226
002 7835 6461 66FA 0681 6CFA 7805 64A3
003 6661 B427 7FFE 8427 6461 66F1 64A3
004 6661 3AA3 0600 18A3 04E0 B82B 7802
005 6461 7FE4 0600 0C60 6661 7FF3 0601
006 6CFA 7803 6461 0CF9 7807 6461 66F9
007 AEFA 7806 6601 D00F F0F9 D010 F0F8
008 BC26 7952 8C26 D03B F0D4 D03A F0D5
009 AAD2 789A 64D4 18AF 66A9 64D5 1CB0

图 64

4.3.5. AUTO

AUTO 图示为 ，Auto 是综合 Blank Check、Program 及 Verify 三项功能，选择 Auto 会先检查芯片是否为空，然后编程，确认编程芯片。

当执行成功后，讯息栏出现以下讯息(图 65) ，如果有勾选 ”Enable Program Times” 则允许烧录的次数会减 1，并将剩余烧录次数显示于讯息栏内。

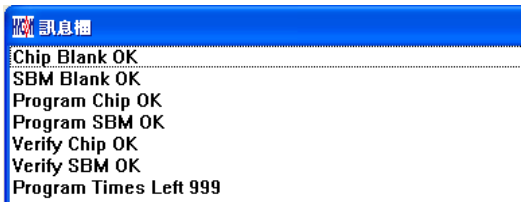


图 65

如果有一项失败，整个过程会立即停止，并在讯息栏显示错误讯息。

4.4. 离线烧录

4.4.1. 烧录说明

以下说明以 HY15P41 为范例

当用户程序由开发阶段进入工程试产阶段时，此时可以单独使用 HY10000-WK08D 烧录器(如下图 66)，无须连线 PC。

图 66

- USB: 与计算机连接载入 Hex 档用
  当离线烧入时，需要供给 5V 电源
- HY15P 系列烧录控制端口

PIN 1	VPP	连接芯片的 VPP
PIN 2	PSCK	连接芯片的 PSCK
PIN 3	PSDI	连接芯片的 PSDI
PIN 4	PSDO	连接芯片的 PSDO
PIN 5	VDD	连接芯片的 VDD
PIN 6	VSS	连接芯片的 VSS

- Program，芯片烧录按键
- Blank Check，芯片空白检查按键

- Information，烧入器内部 Hex 档信息查看
- L1 绿色 LED：USB 或 Adapter 上电、OTP 烧录、Blank Check…执行成功显示灯号

  L2 红色 LED：OTP 烧录、Blank Check、频率校正… 执行错误显示灯号

  L3 黄色 LED：烧录中

下图 67 为 PC 在线时 ,程序 Download ,在线烧录的芯片与控制板的烧录接脚连接方法：

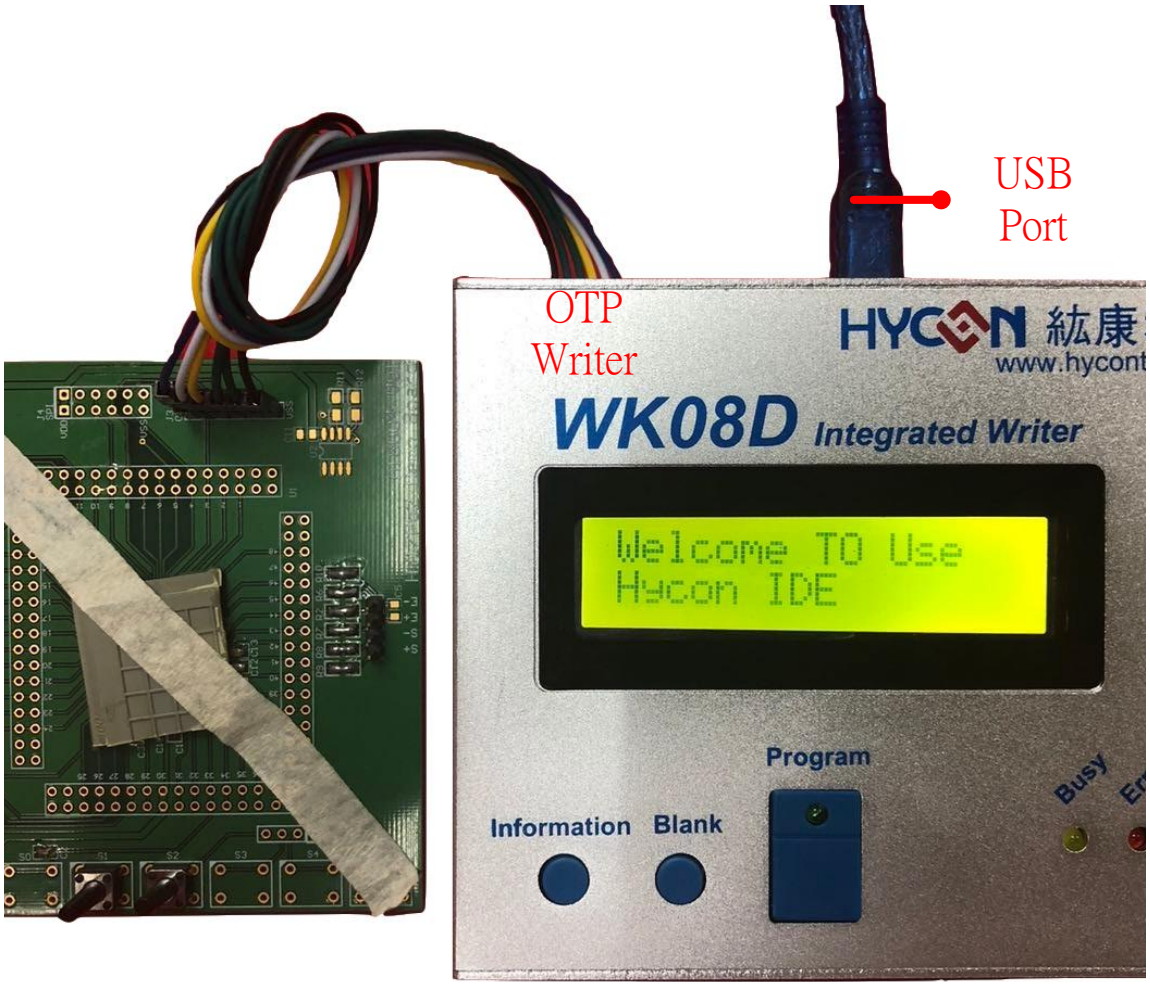


图 67

- 离线操作时需要先将 Hex 档 Download 到烧录器的 Flash Memory 内，步骤参考 4.2.1 或 4.2.2 一节。
- 离线烧录时 ,先按按键 Blank 可检查芯片是否为空 ,检查完后应为 L1 绿色 LED 亮。
- 按键 Program 为烧录按键，其步骤为 Blank Check → Program → Verify，如果在 Download 到 Flash Memory 之前有在”组译选项”中勾选”烧录保护”，则在 Verify 后将执行烧录保护；如果没有沟选择在 Verify 后停止，烧录完成后 L1 绿色 LED 亮。
- 烧录完成后按按键 Blank 再次检查芯片是否为空，此时应该亮 L2 红色 LED，表示

有烧录完成

- 如果在执行中有任何一项错误或失败 ,则 L2 红色 LED 亮 ,如果成功则 L1 绿色 LED 亮。

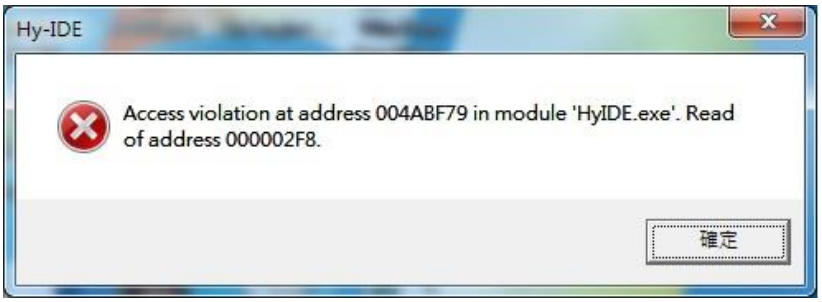
4.4.2. 限制烧录次数

在介面设定的”组译选项”内有一个栏位是点选”Enable Program Times” ,这个选项是允许限制 Download 程序的烧录次数。这是一个安全机制，在产在线限制烧录次数，防止烧录过量。当点选 ”Enable Program Times”之后，并在 ”Input Program Times”下方栏位上填写烧录的次数(最多为 99999999，最少 1)，当在 Compiler 程序后或下载档案到 Flash Memory 之后，会将此参数载入；当每一次执行烧录的动作时，会将此计数值自动减 1，当此计数值减到 0 时，如果继续烧录，则不会执行，并会亮 L2 错误讯息(红色灯号)，但 Blank Check 会正常动作。

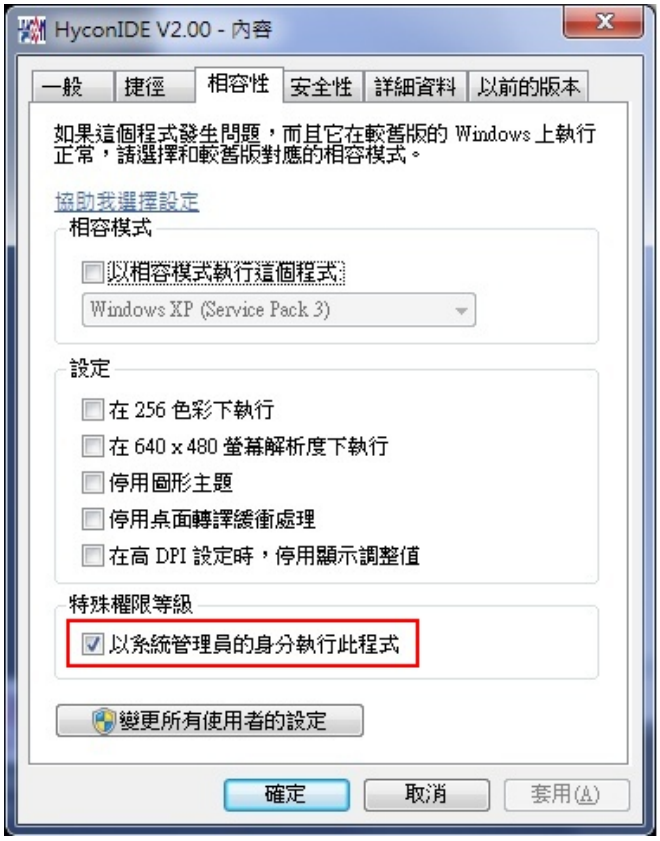
5. 故障排除

5.1. 无法使用 Hycon-IDE

如出现下图



通常在使用 windows 7 以上会出现这样的问题 ,则必须将 Hycon-IDE.exe 设定成如下图，以系统管理员的身分执行此程序，这样将可避免使用遇到相同问题。



6. 修订记录

以下描述本文件差异较大的地方，而标点符号与字形的改变不在此描述范围。

文件版次	页次	日期	摘要
V02	ALL	2021/11/08	初版发行