

---

**HYCON 紘康科技**

**HY17M Series**

**Assembly IDE 软件使用说明书**

### 目 录

<b>1. HY17M IDE 入门</b> .....	<b>5</b>
1.1. 简介.....	5
1.2. HY17M IDE 安装及系统要求.....	5
1.3. 安装及卸载.....	5
1.4. Demo Code 导入说明.....	7
1.5. Demo Code 操作方式及使用.....	8
<b>2. HY17M IDE 介面说明</b> .....	<b>9</b>
2.1. HY17M IDE 编辑介面.....	9
<b>3. HY17M IDE 除错介面</b> .....	<b>18</b>
3.1. 快速执行.....	19
3.2. RAM 视窗.....	22
3.3. Register 视窗.....	25
3.4. Watch 视窗.....	27
3.5. 堆栈视窗.....	29
3.6. 暂存器修改记录.....	30
3.7. 源程序视窗下的 Hint 功能.....	32
<b>4. 烧录视窗</b> .....	<b>34</b>
4.1. 介面设定.....	34
4.2. 操作步骤.....	36
4.3. PC 连线烧录 MTP.....	38
<b>5. 故障排除</b> .....	<b>43</b>

---

5.1. 无法使用 Hycon-IDE.....	43
6. 修订记录.....	44

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。

# HY17M Series

## Assembly IDE 软件使用说明书

## 1. HY17M 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 HY17M 系列产品的 MTP 上，使客户的终端产品能快速的开发并销售到市面上。

### 1.2. HY17M IDE 安装及系统要求

#### 运行 HY17M IDE 所需的最低系统配置：

- PC/NB Hardware requirement:  
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:  
-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. 安装及卸载

#### 安装

解压缩 IDE 软件包后运行  setup 就可以进行软件的安装(步骤如下图 1),安装完毕后直接运行\HYCON\HY17M IDE 目录下的  HY17M IDE.exe 就可以开启 IDE 软件进行操作。

**Note** : 在运行软件时请以『系统管理员身份』来运行

# HY17M Series Assembly IDE 软件使用说明书

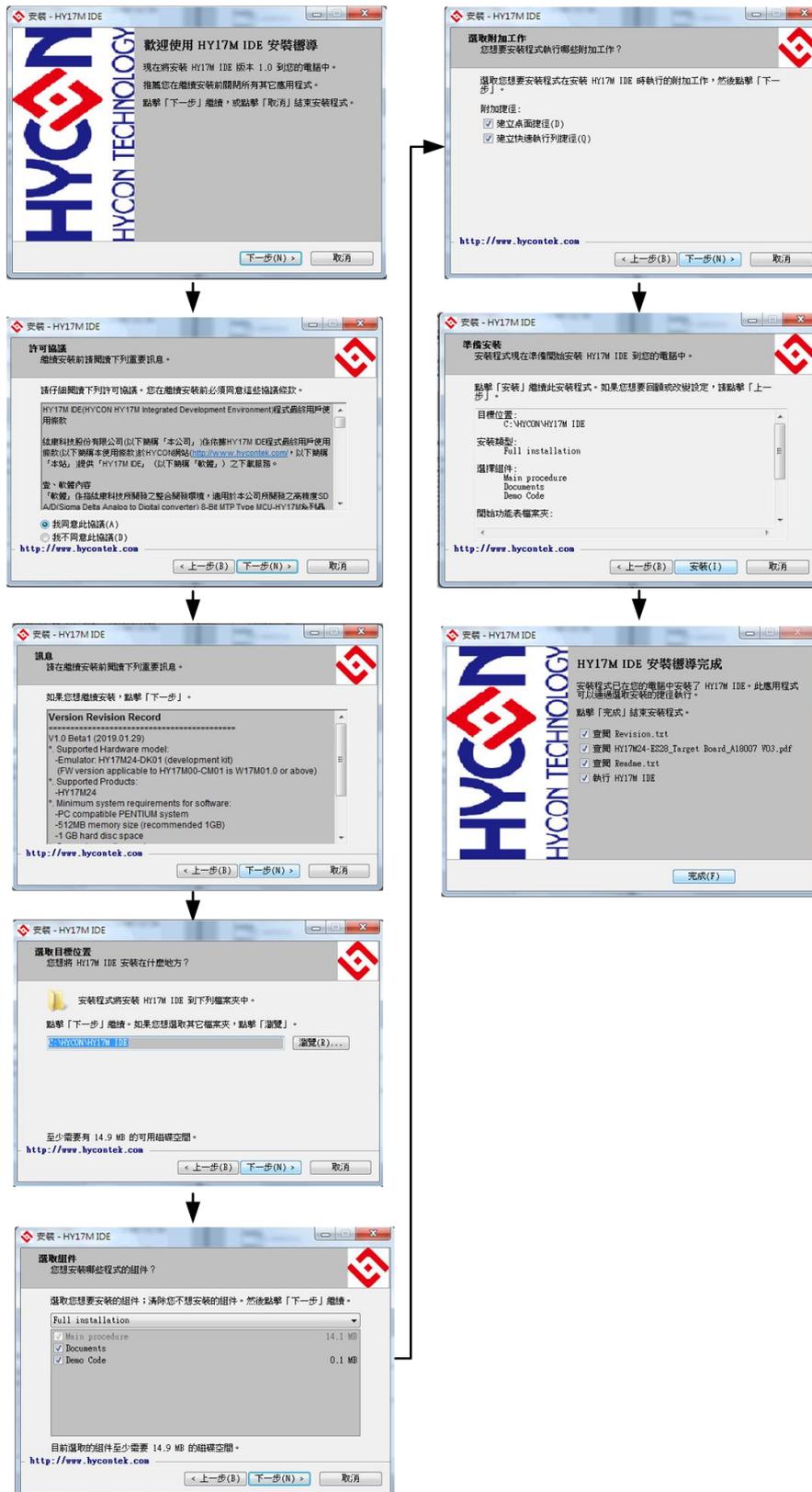


图 1

## 卸载

直接运行 IDE 软件目录下的 unins000，就能解除 IDE 软件的安装。

### 1.4. Demo Code 导入说明

- 开启 C:\HYCON\HY17M IDE\DemoCode 主程序
- 设定为组译主档
- 组译并进行除错

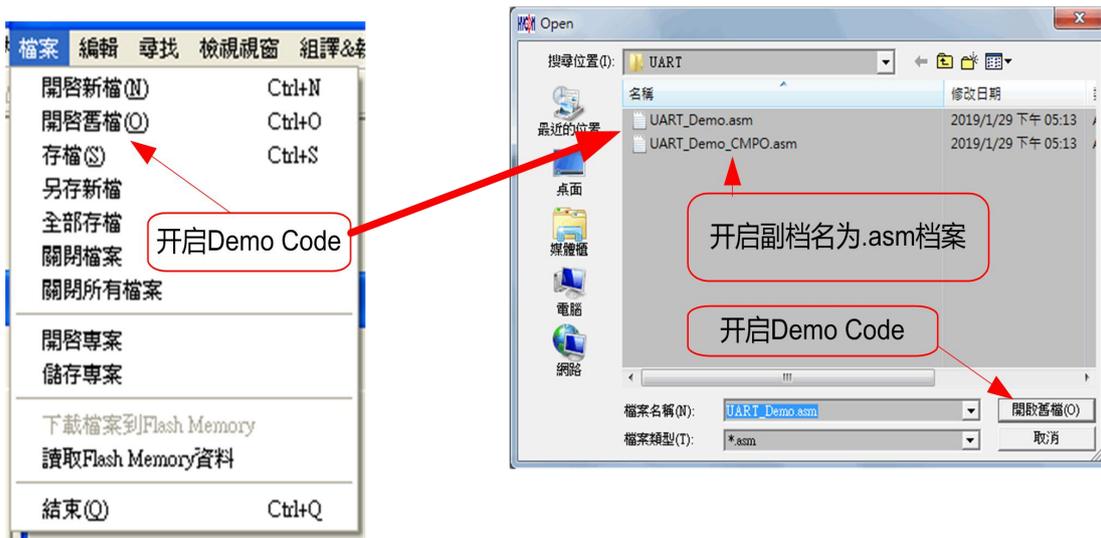


图 2

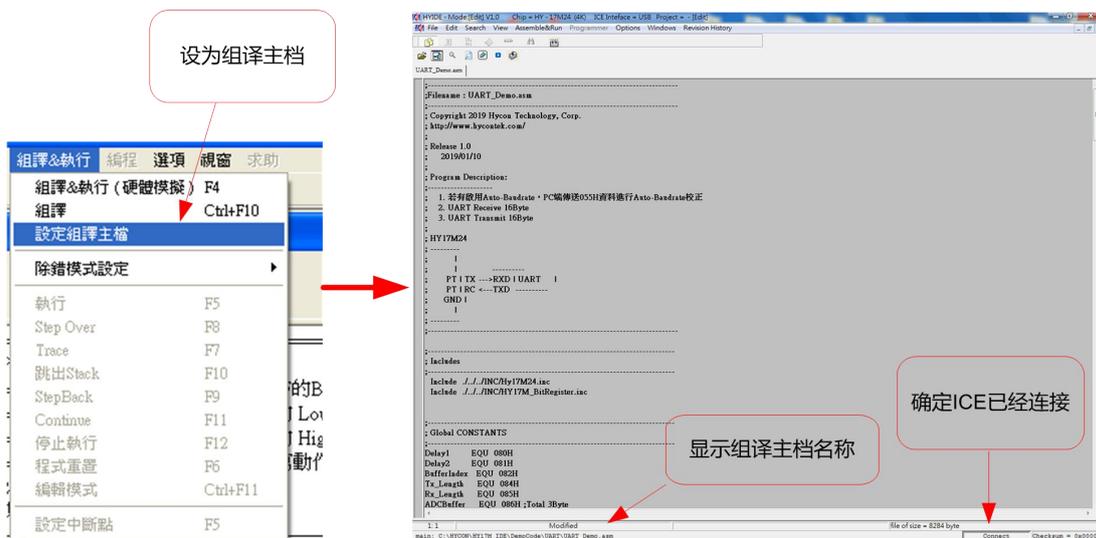


图 3

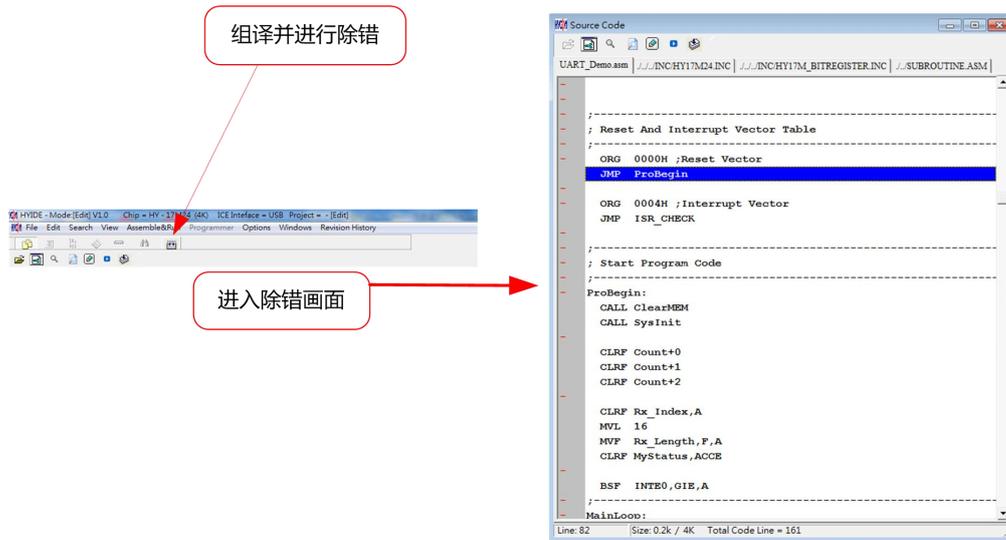


图 4

- 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 HY17M IDE 软件安装后，于目录 C:\HYCON\HY17M IDE\DemoCode 主程序下会有提供 Demo Code 供使用者参考。

## 2. HY17M IDE 介面说明

### 2.1. HY17M IDE 编辑介面

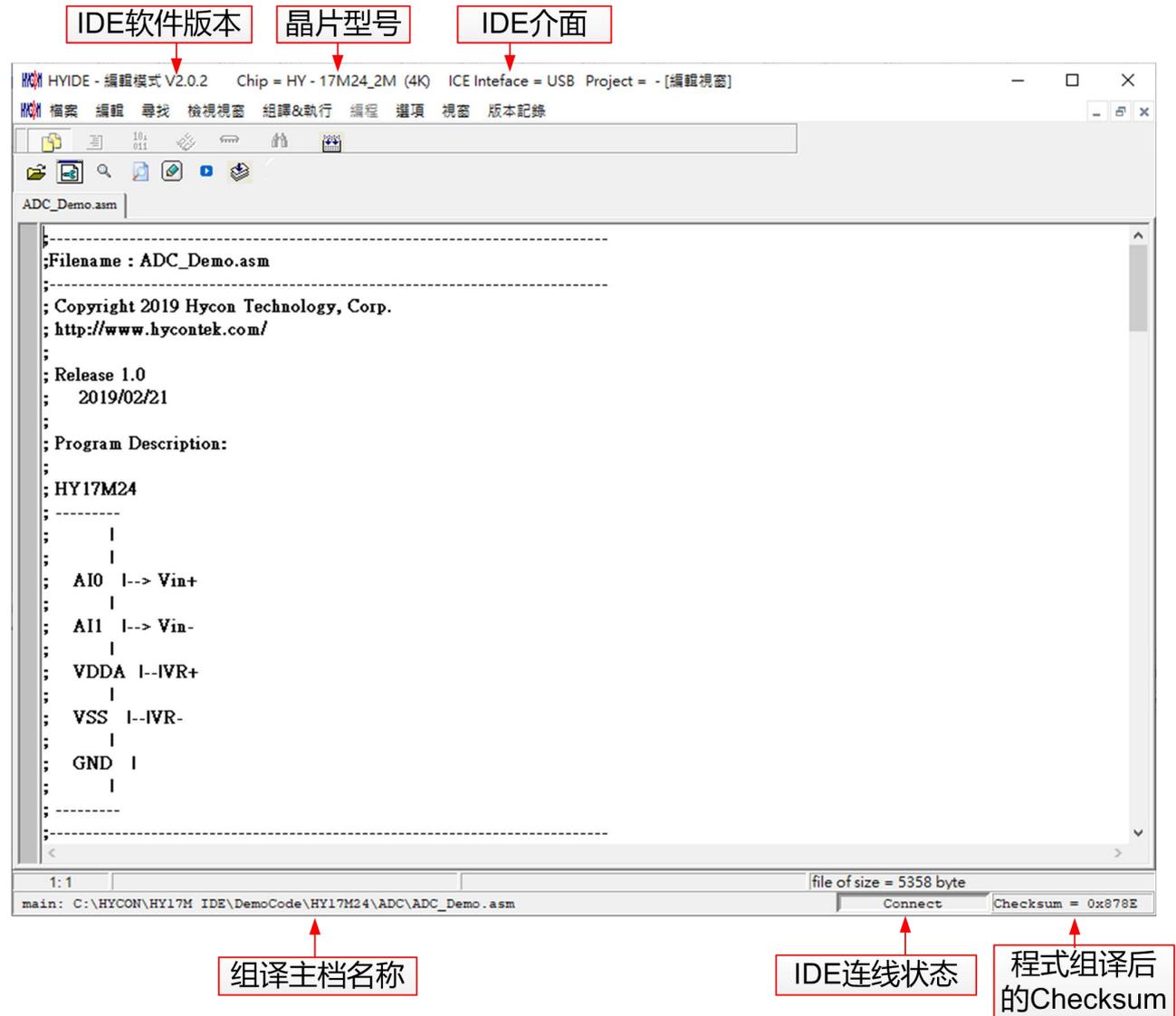


图 5

### 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   
跳到以设定的标签处。
- 寻找字符串   
寻找已输入过的字符串。
- 寻找下一个字符串   
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

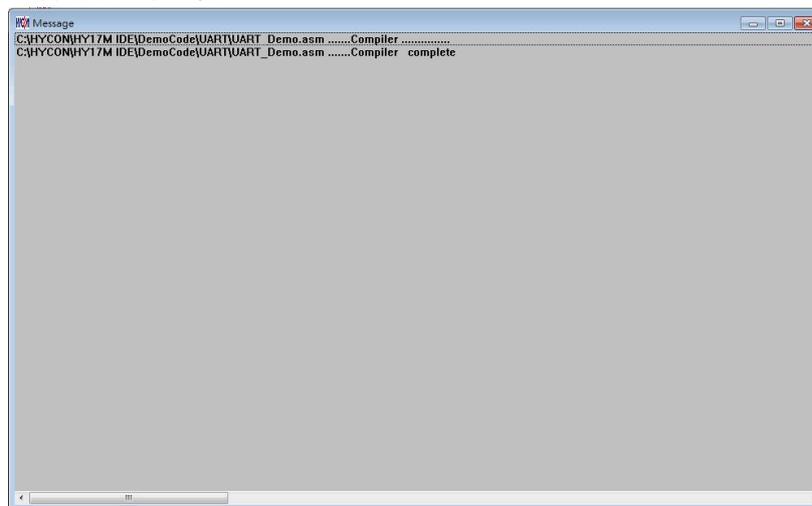


图 6

### 2.1.2. 档案



图 7

- 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
- Load Hex File → 载入 Hex 档案
- 结束 → 结束 Hycon-IDE 程序

### 2.1.3. 编辑

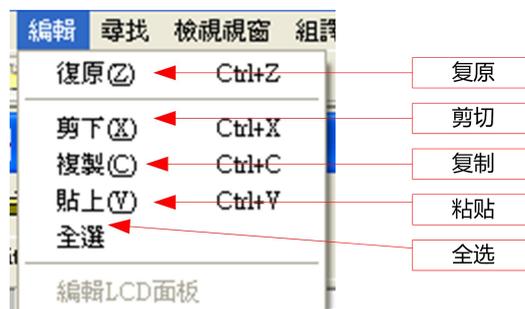


图 8

- 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 9

- 编辑视窗 → 将编辑视窗指定为当前的活动视窗
- 讯息 → 将讯息栏指定为当前的活动视窗
- Read EEPROM → 将 EEPROM Memory 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

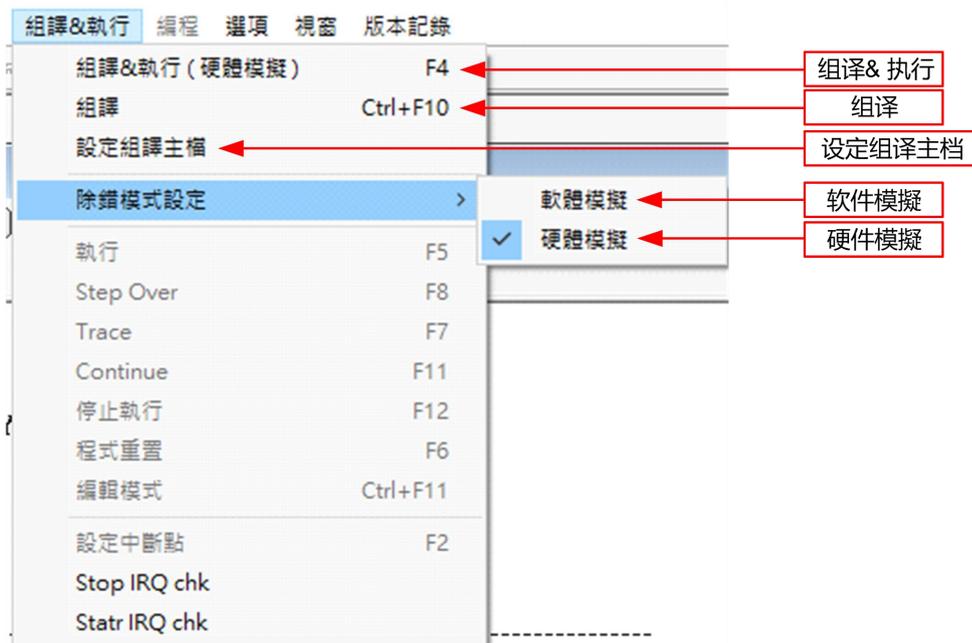


图 10

- 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

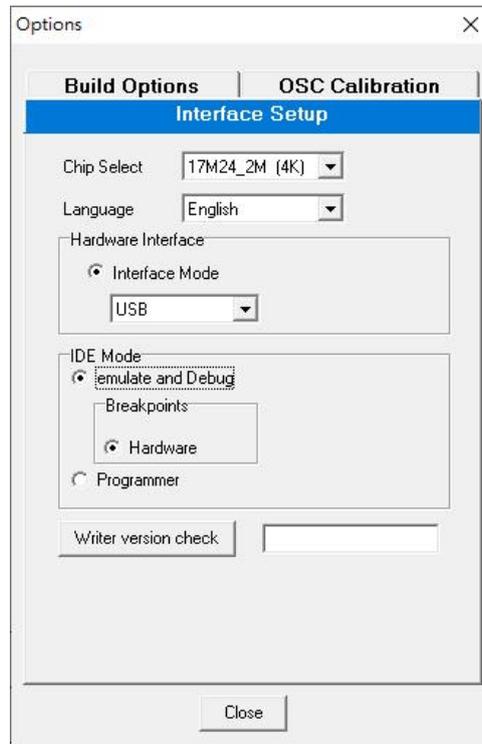


图 11

- 介面设定(由选项中选取)



图 12

- 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 13

- 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 MTP 的设定中。
- 烧录次数限制：参考烧录视窗之 2.1.6 介面设定章节。
- 编辑功能字型选择：选择编辑器的字号。
- 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。
- User Key：烧录视窗之 2.1.6 介面设定章节。

- OSC Calibration

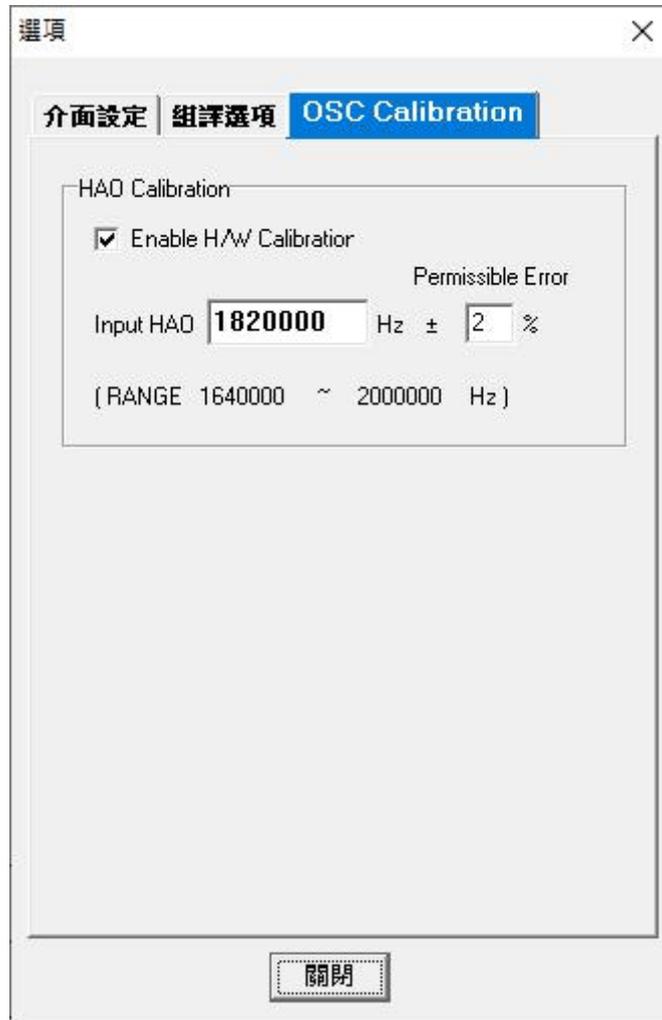


图 14

- Enable H/W Calibration:勾选是否进行硬件校正
- 输入 HAO 校正值与误差值，下方 RANGE 根据芯片型号改变

### 2.1.7. 视窗

可选择所有开启的视窗做垂直或水平的排列。

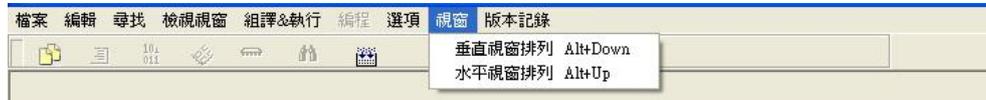


图 15

### 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, H08D 指令集；

使用者一开始可以先参考软件所附的 demo code，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:

<code>;程序名称可为任意名称</code>	
<code>Include Hy17M24.inc</code>	<code>;HY17M 系列特殊暂存器名称、位址定义</code>
<code>Include Main.inc</code>	<code>;RAM 名称、位址定义</code>
<code>ORG 00H</code>	<code>;宣告程序开始</code>
<code>JMP BEGIN</code>	<code>;跳跃到主程序</code>
<code>ORG 04H</code>	<code>;宣告中断旗标位置</code>
<code>Include Interrupt.asm</code>	<code>;引用“Interrupt.asm”中断子程序；</code>
	<code>;include 档案限制最多 100 个。</code>
<code>BEGIN:</code>	<code>;主程序开始. Label name 的定义可以为任意字</code>
<code>Include Initial.asm</code>	<code>;引用“Initial.asm”硬件初始化子程序</code>
<code>JMP T1</code>	<code>;跳跃到 T1 子程序</code>
<code>...</code>	
<code>T1:</code>	
<code>NOP</code>	
<code>Include Sub.asm</code>	<code>;引用“Sub.asm”子程序</code>
<code>END</code>	<code>;程序结束</code>

# HY17M Series

## Assembly IDE 软件使用说明书

### 3. HY17M 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

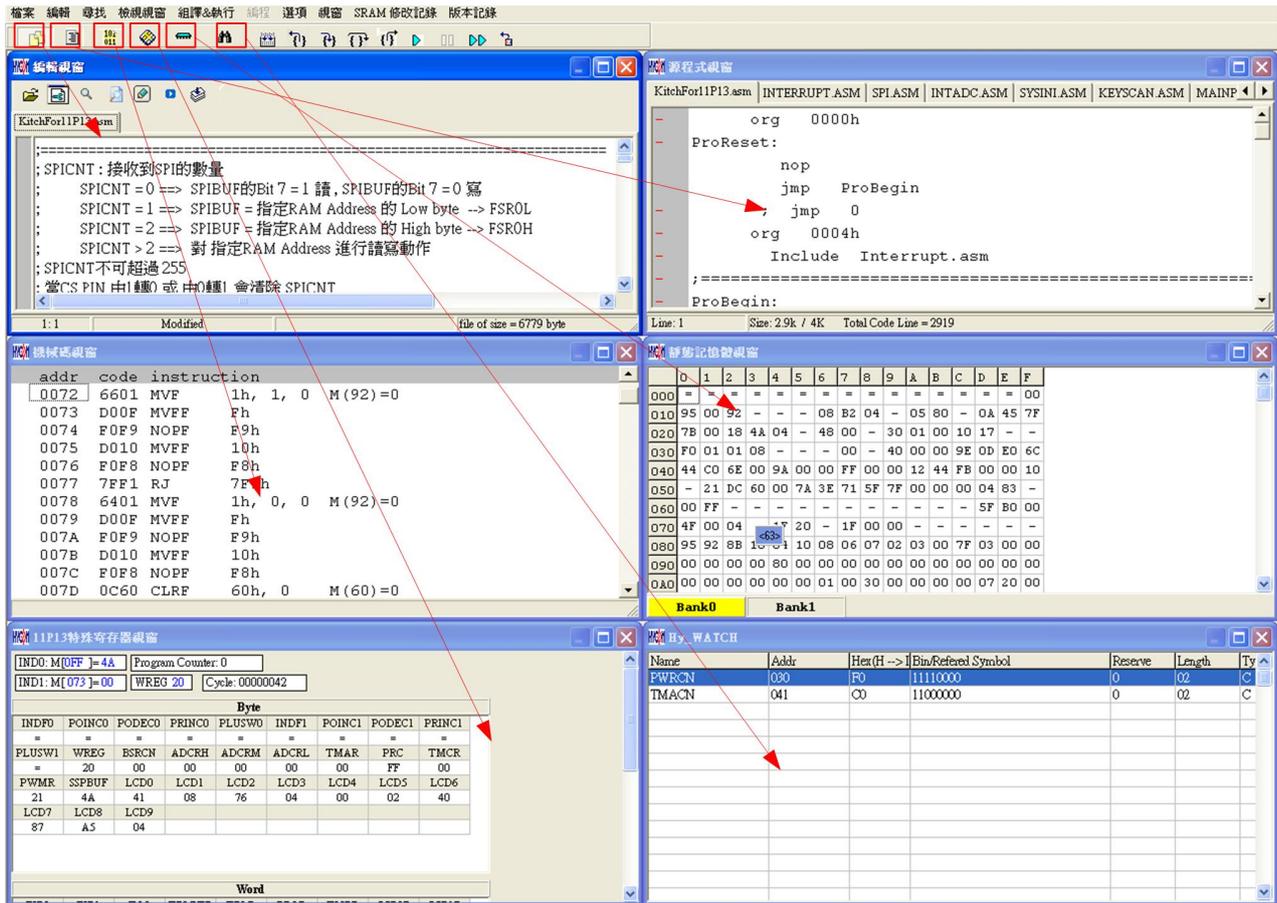
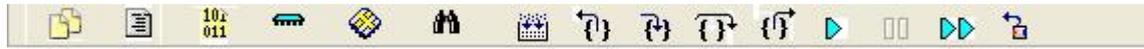


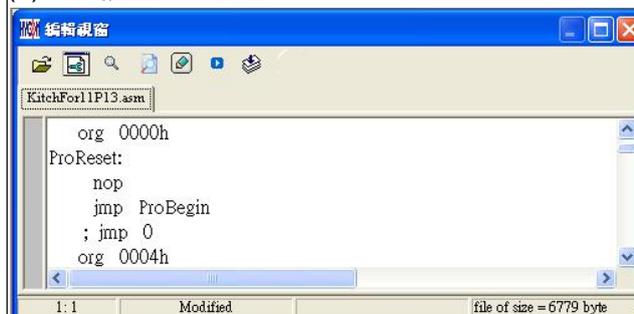
图 16

### 3.1. 快速执行

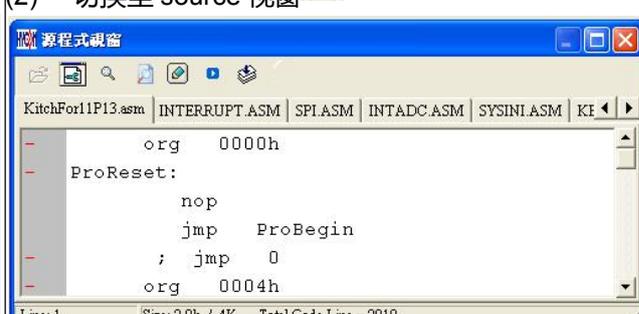


- 快速视窗切换

(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



图 17

- 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 18

- 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。

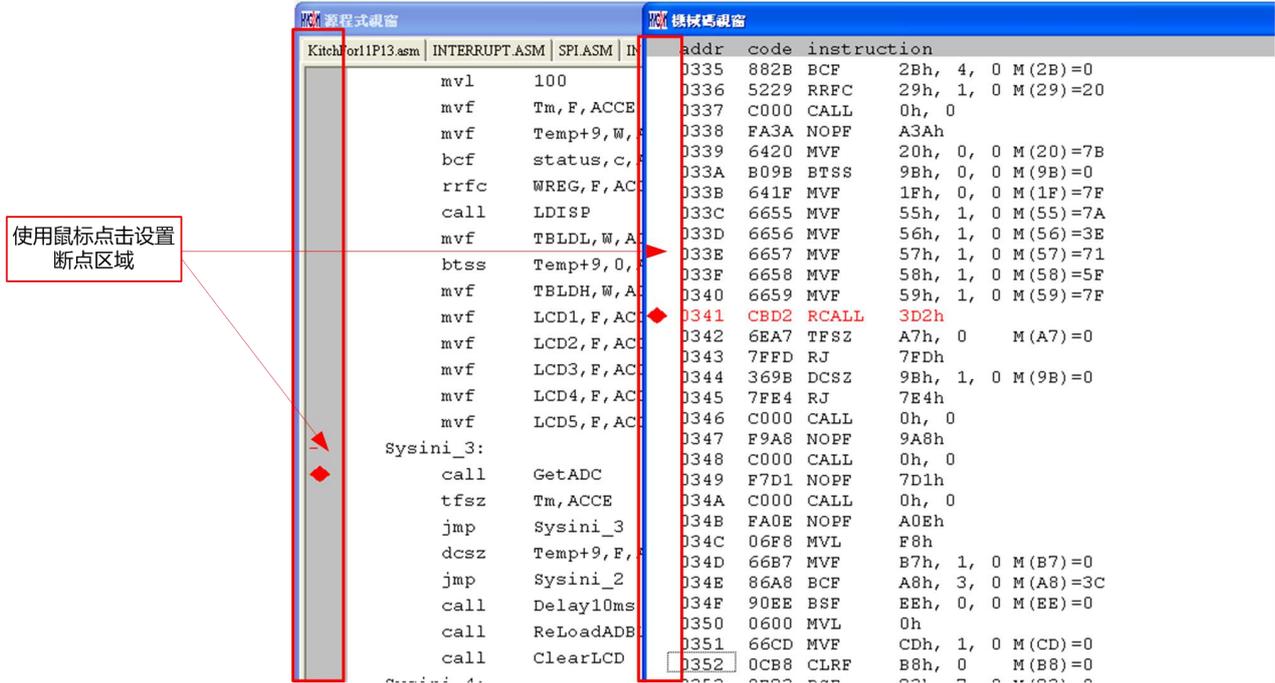


图 19

### 3.2. RAM 视窗

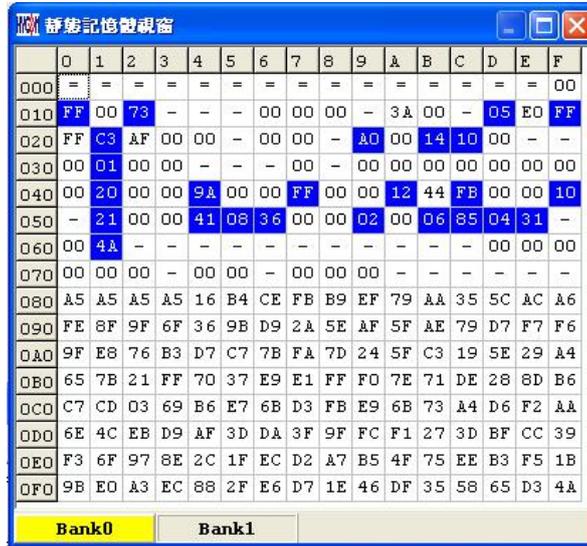


图 20

- 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 21

- 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 22

- 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，使用键盘 Key In 或鼠标点击

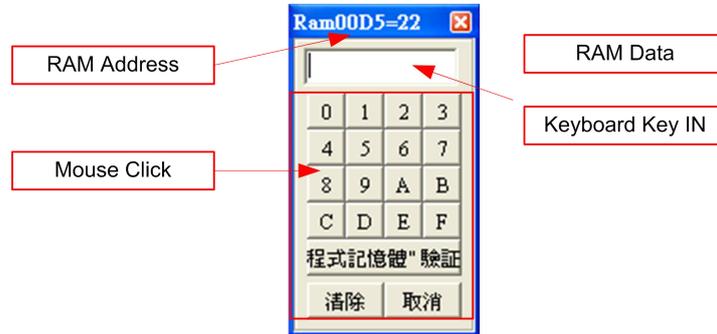


图 23

### 3.3. Register 视窗

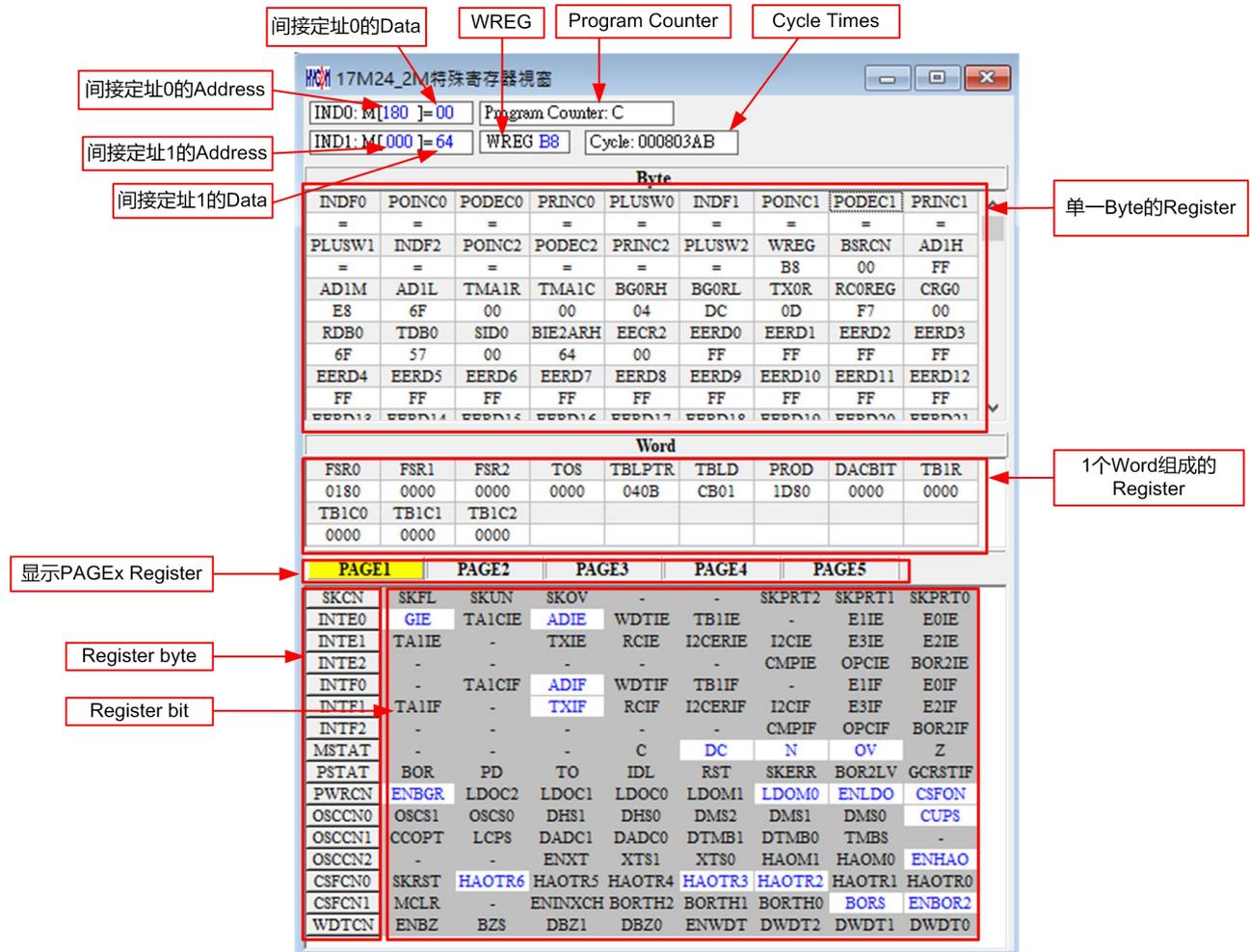


图 24

- 修改间接定址 Data 或 Address

如图 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 25

如图 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



### 3.4. Watch 视窗

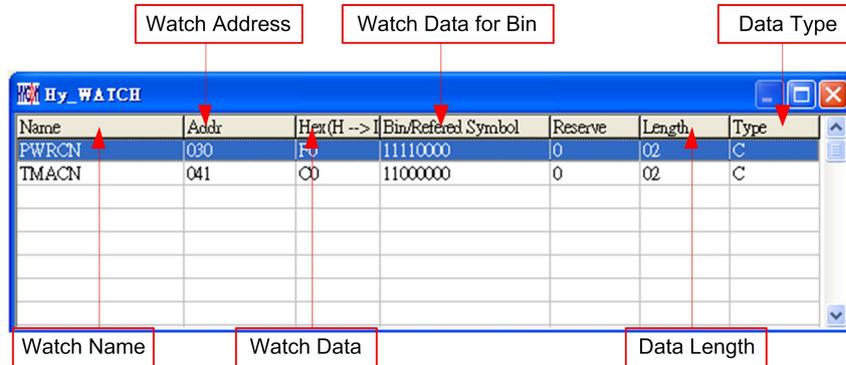


图 30

- 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

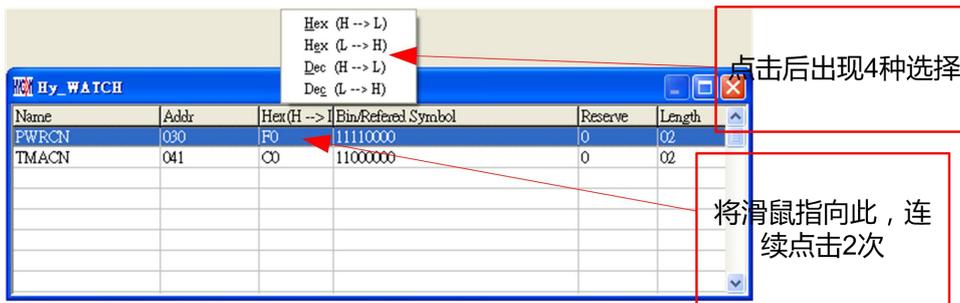


图 31

- 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

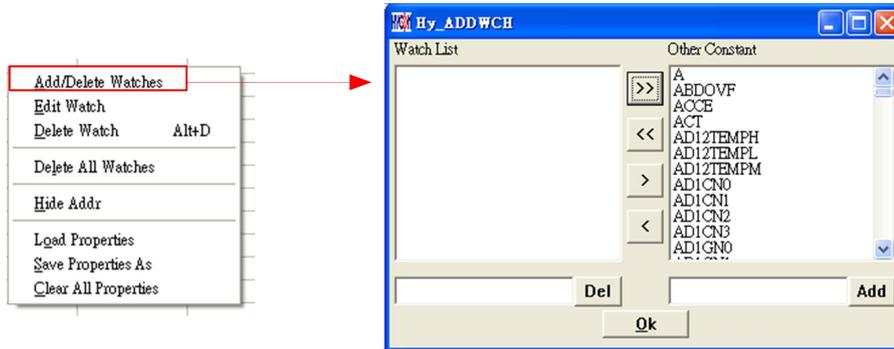


图 32

### 3.5. 堆栈视窗

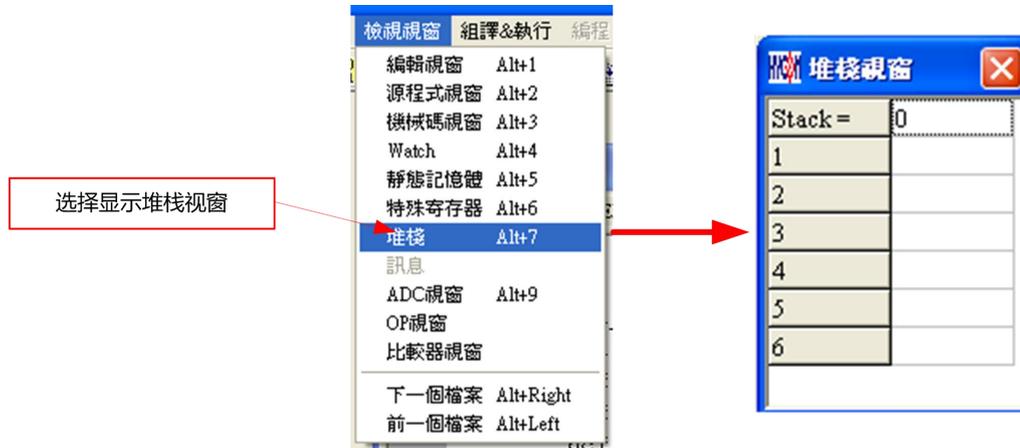


图 33



图 34

### 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。

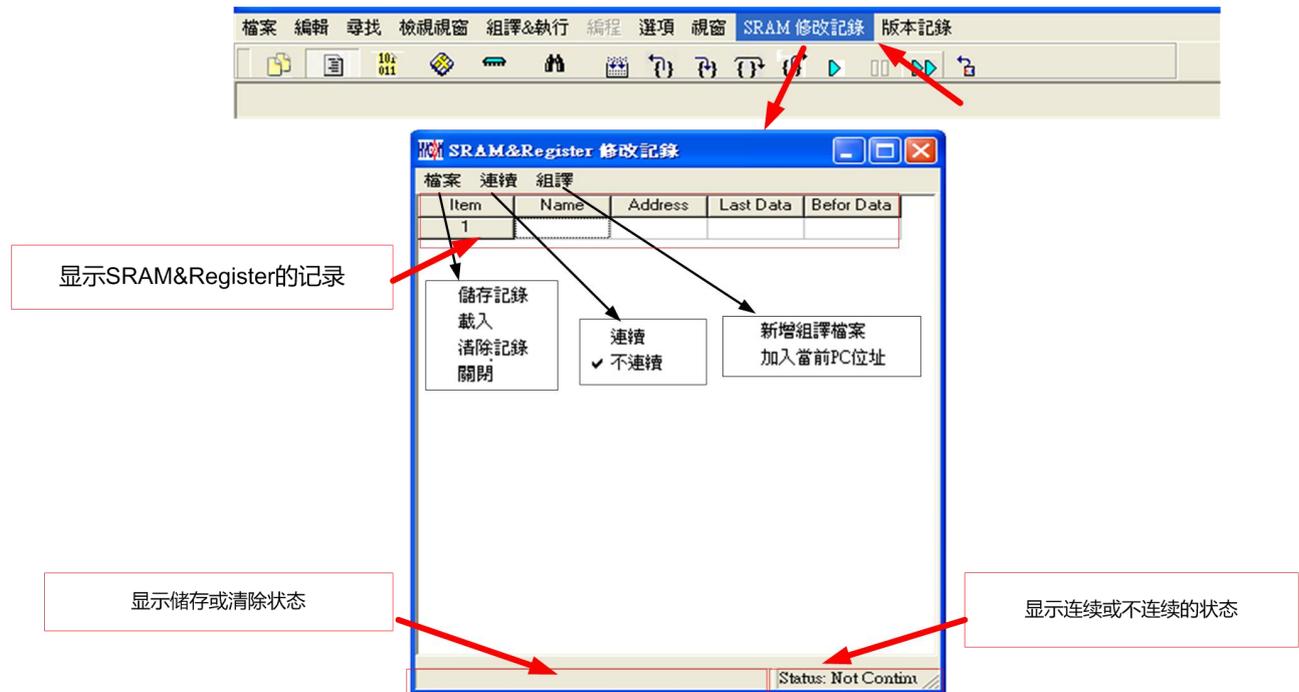


图 35

# HY17M Series Assembly IDE 软件使用说明书

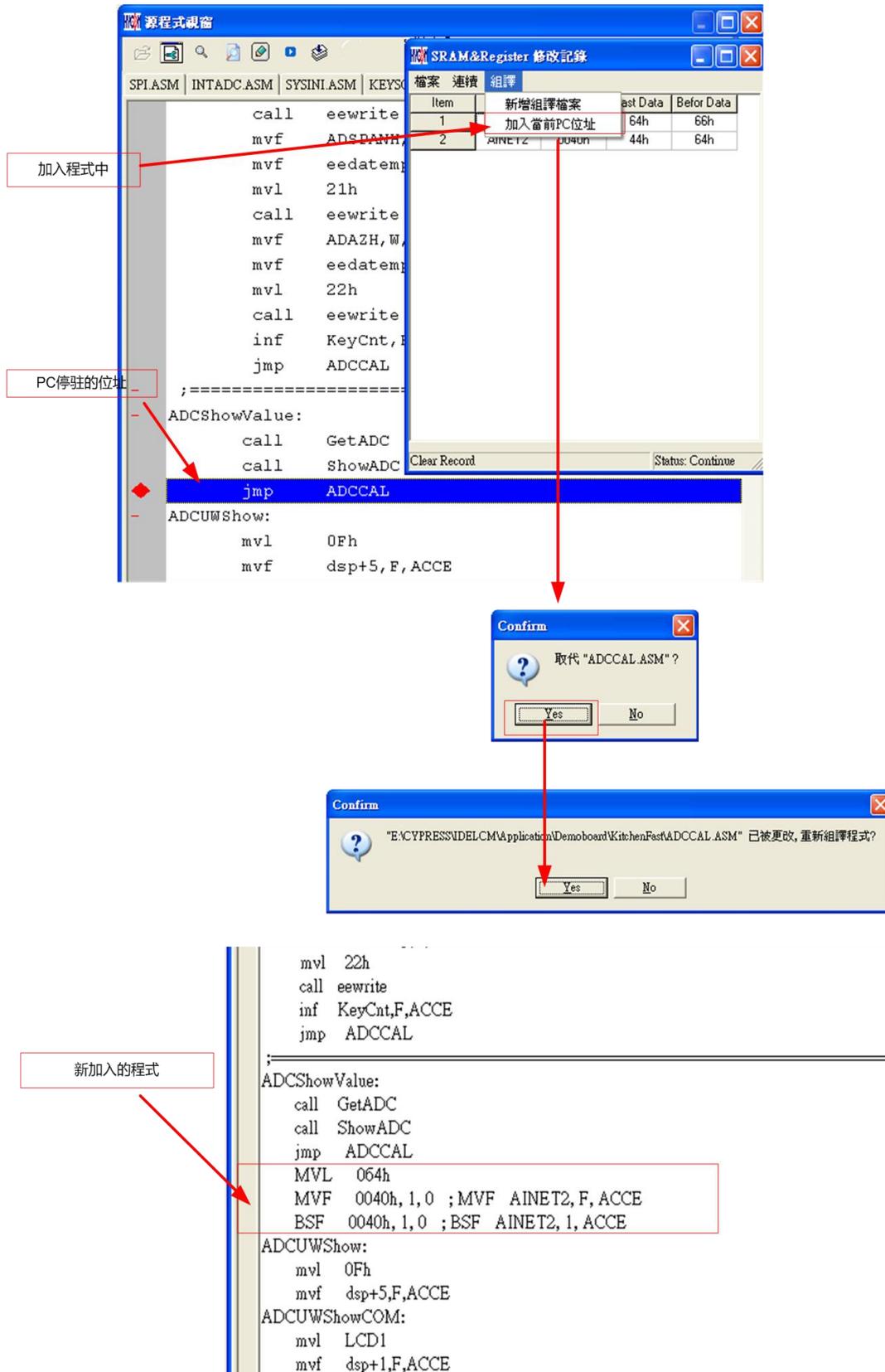


图 36

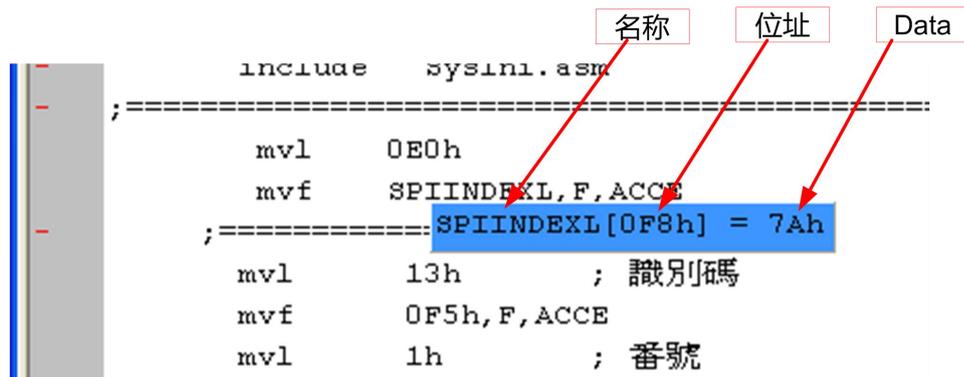
### 3.7. 源程序视窗下的 Hint 功能

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, RRF, RLFC, SWPF, DAW, INSUZ, DCSUZ, ARLC, ARRC, CPSG, CPSL, CPSE, TFSZ, BTFF, BSF, BCF, BTSS, BTSZ, MVFF(不是 Macro)。

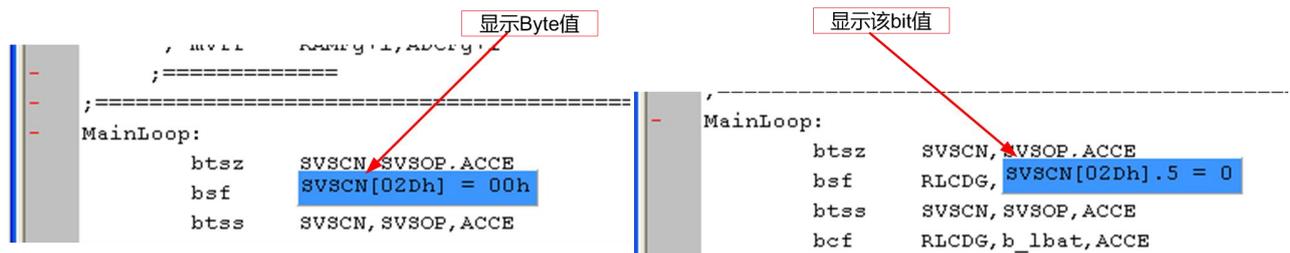
- 当指令为位元操作时只有第一个参数才有效
- 当指令为 BCF、BSF、BTSS、BTSZ 及 BTGF 时,当指向第一个参数显示 Byte 数值,当指向第二个参数显示该 Bit 的值(1 or 0)
- 当指令为 MVFF 时(不是 Macro) ,当指向第一个参数显示第一个参数数值,当指向第二个参数显示第二个参数数值
- 如果参数为 INDF0、POINC0、PODEC0、PRINC0、INDF1、POINC1、PODEC1、PRINC1 时,此时 Data 为 FSR0 或 FSR1 内的位址的 Data
- 如果参数为 PLUSW0 或 PLUSW1 时,此时 Data 为 FSR0+WREG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



```

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
    
```

图 37



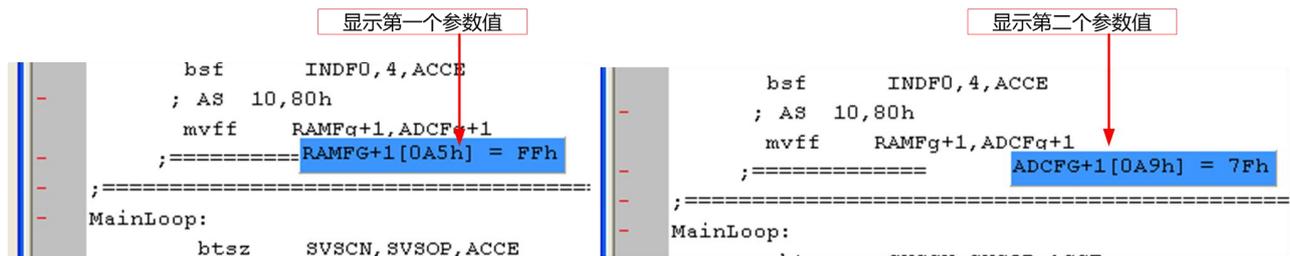
```

;=====
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
    
```

图 38



```

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

```

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

图 39

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
    btf  ...

```

图 40

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 41

### 4. 烧录视窗

#### 4.1. 介面设定

要进入烧录式窗画面，点选“选项”，出现选择画面，点选“介面设定”，如图 42 所示。



图 42

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 43。

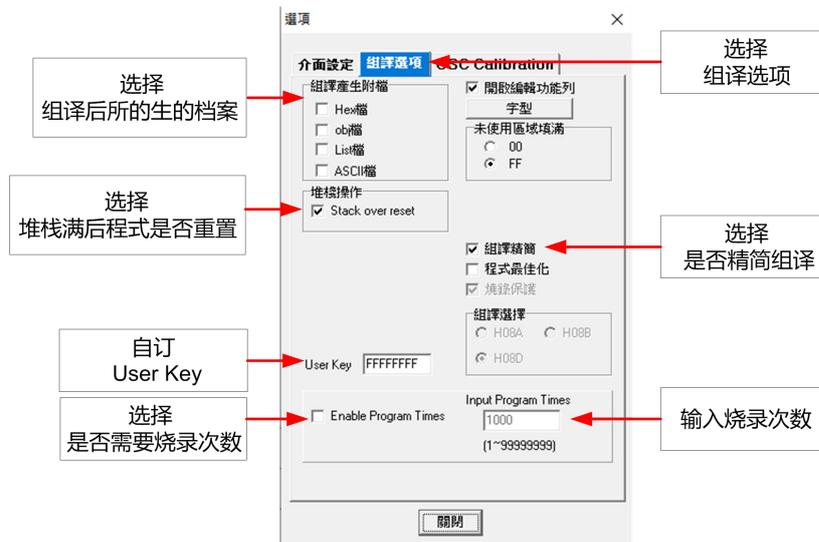


图 43

- 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 MTP 程序运行后如果发生堆栈满或溢位是否要重置。
- 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)。
- User Key → 填入读取保护密码, 届时回读程序需输入此密码。

## 4.2. 操作步骤



图 44

开启旧档 → 开启已经写好的源程序组译主档。

开启项目 → 开启储存的项目名称。

储存项目 → 储存已完成的项目。

Load Hex File → 载入 Hex 档案

### 4.2.1. 开启档案与组译

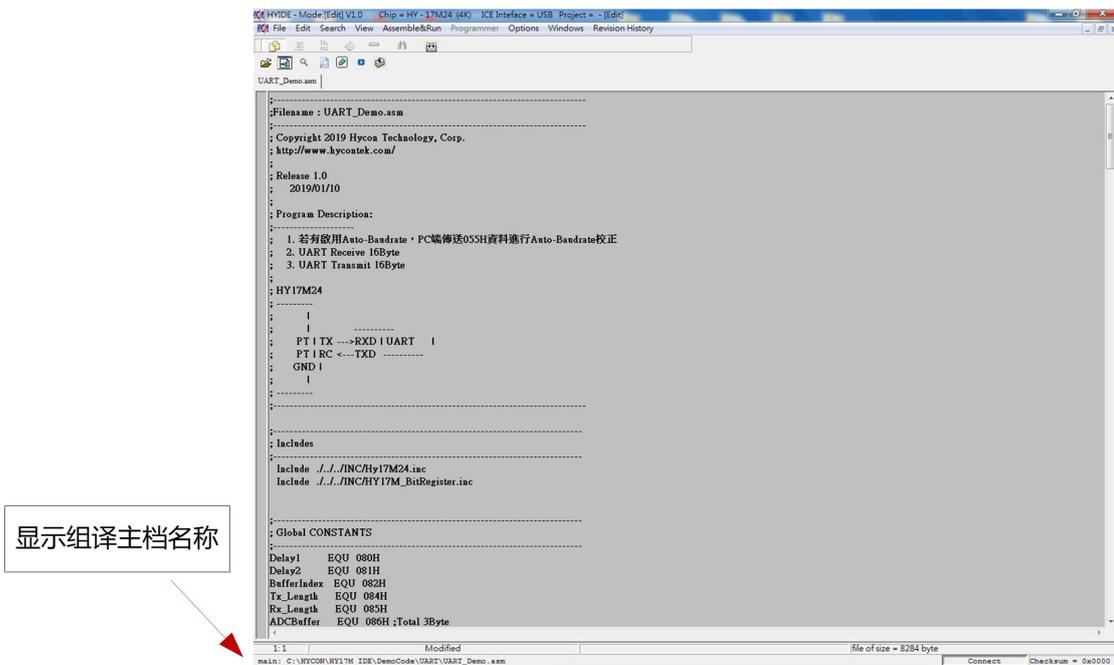


图 45

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。

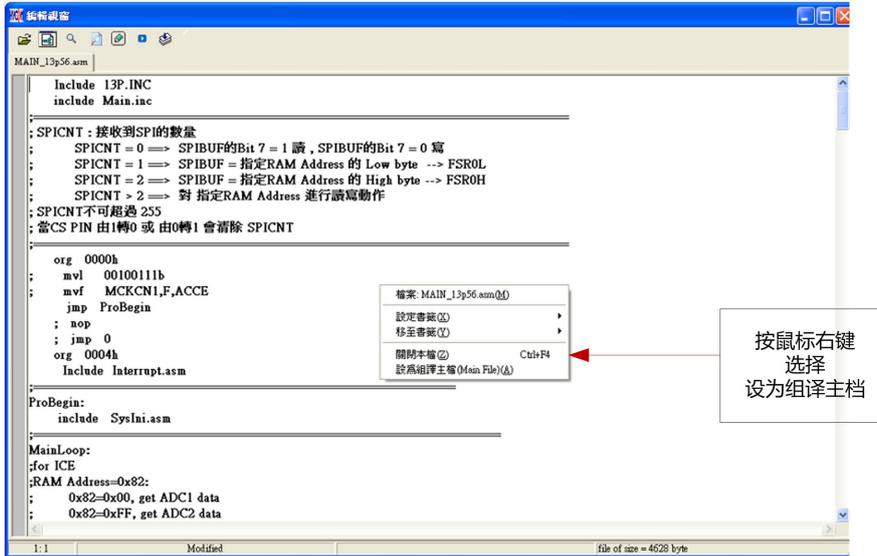


图 46

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



图 47

1. 当组译完成后在下方显示组译完成后的 Hex 档名称与 Checksum。

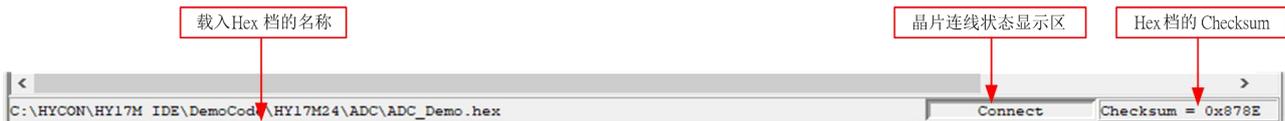


图 49

## 4.3. PC 连线烧录 MTP



图 50

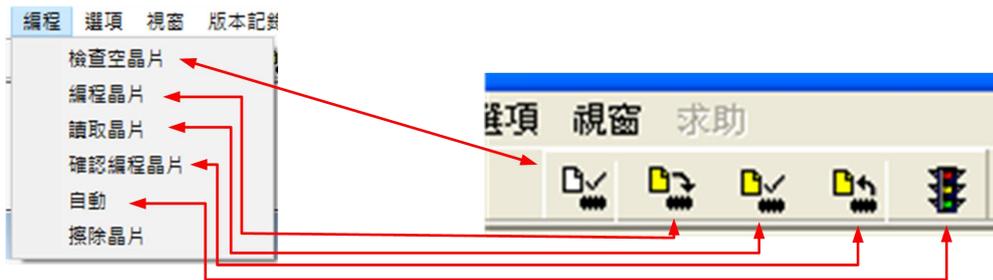


图 51

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、擦除芯片等动作，如果没有成功载入，则以上的动作将不会成功。如图 52~图 54。

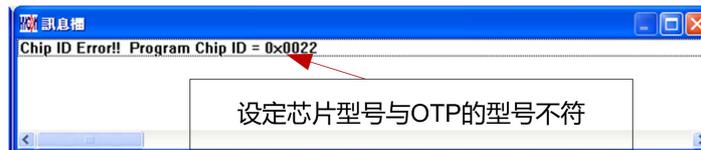


图 52



图 53



图 54

确定在标题视窗下所选择的烧录芯片型号，与 MTP 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 MTP 型号是否相同，如果不同否则不会烧录到 MTP 内。

如下图所示，Chip Locked 意旨芯片已烧入成功且设定 User Key，故 IDE 无法顺利连接，擦除芯片即可重新连接 IDE。此情况下仍可直接重新烧入(即擦除芯片后再烧入)或进入 Debug 模式。

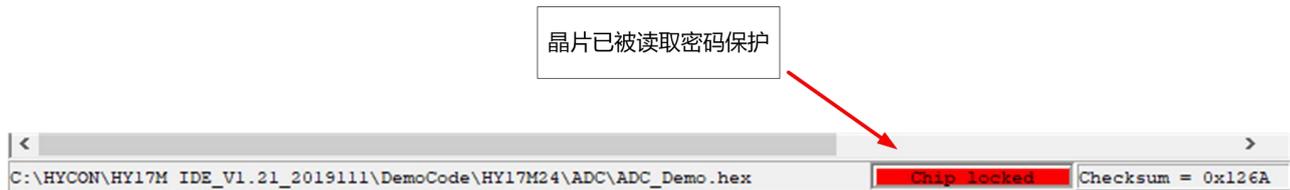


图 55

### 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 MTP 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 MTP 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 56)。



图 56

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 57)。



图 57

### 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 MTP 的芯片中或烧录到控制盒中可进行离线烧录功能，左击图示则跳出烧录讯息视窗(图 58)，选择烧录到芯片或控制盒，最后烧录完成并组装成品将可依照使用者所写的指令运行程序。

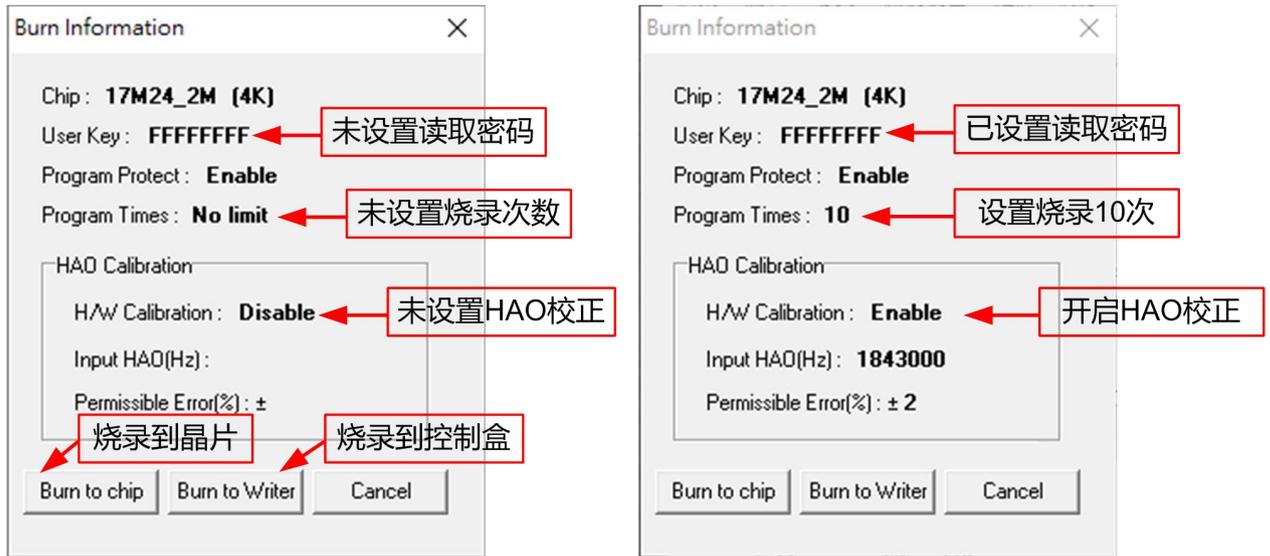


图 58

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。

如果选择芯片正确以及编程成功, 讯息栏出现以下讯息(图 59), 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 59

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 60)。



图 60

### 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 MTP 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 61)。



图 61

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 62)。



图 62

### 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 MTP 的 Checksum 是否与烧录的 Hex 档相同。读取芯片内容(步骤如图 63)，并将内容显示于”显示 Code”视窗内。如有设置 User Key，操作此动作则需要输入 User Key 即可读取芯片。

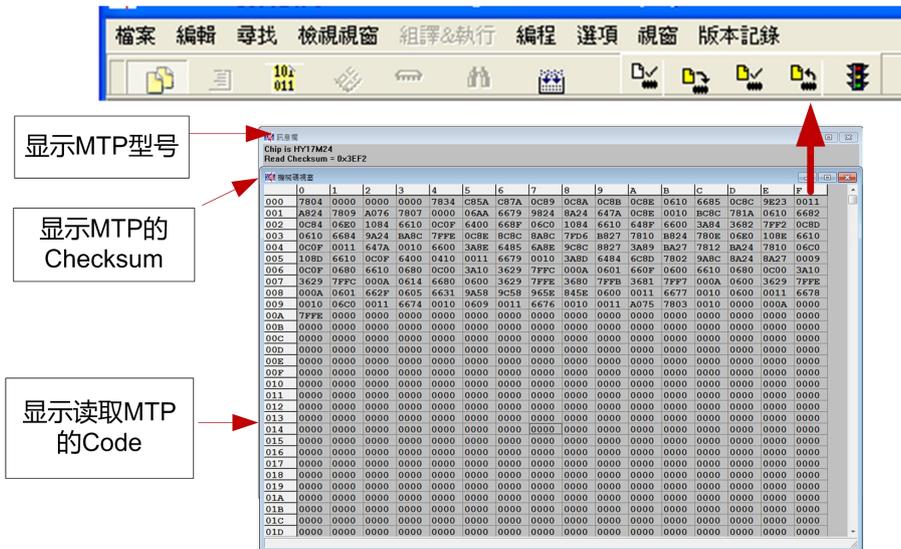


图 63

### 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 64)，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 64

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。

### 5. 故障排除

#### 5.1. 无法使用 Hycon-IDE

如出现下图 65

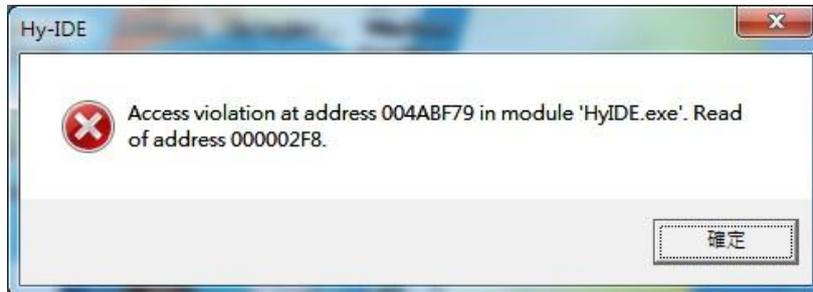


图 65

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 66，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。

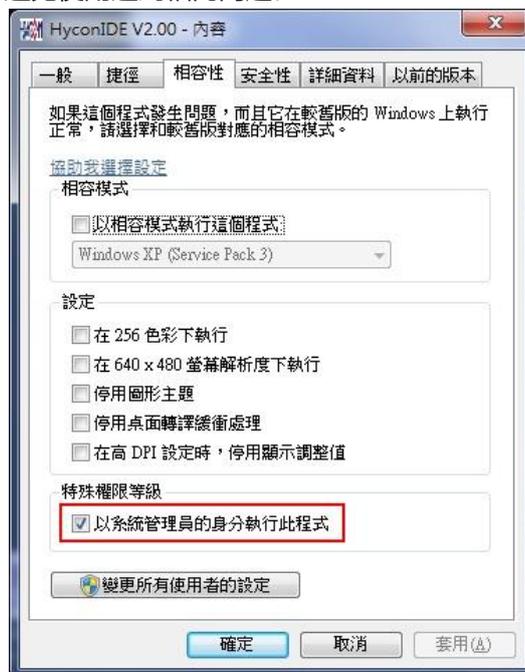


图 66

### 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	2022/04/11	初版发行