TICE58

使用手冊

Tenx reserves the right to change or discontinue this product without notice.

tenx technology, inc.

目錄

產品簡介 2
簡介 2
系統配置 3
軟體安裝 4
快速開始 7
選單功能 12
主畫面功能選單 12
主畫面工具列 14
主畫面視窗 15
環境設定視窗 21
幫助視窗 24
使用技巧 25
中斷點的使用 25
INCLUDE的使用 30
虛擬電路的使用 31
注意事項 35

產品簡介

簡介

爲了縮短開發的時間和過程，十速科技特別開發出 IC 的仿真器 — TICE58。TICE58 是由十速科技自行開發完成的，可支援 TM58 全系列的 IC。

【TICE58 外觀】

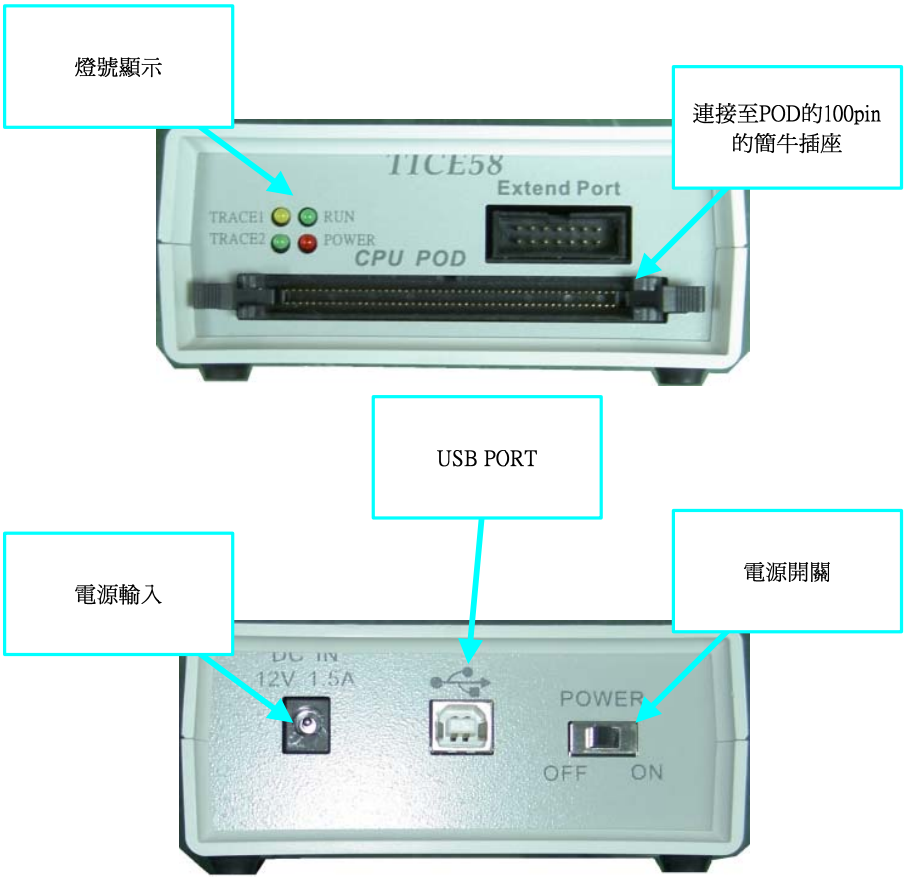
※系統要求：

A、IBM PC 486 或以上。
B、USB Port。
C、64M 以上記憶體。
D、WINDOWS 2000/XP 作業系統。

※TICE58 特點：

A、用 USB 和電腦作連線，仿真速度快。
B、使用內部或外部的振盪器。
C、可設置中斷點。
D、提供內部和外部的 RESET。
E、提供硬體仿真和軟體仿真。
F、編輯、除錯、組譯都可在 TICE58 的軟體完成。
G、除了提供 AUTO (GO)和單步執行(STEP)外，還提供了 STEP INTO 的功能。

系統配置



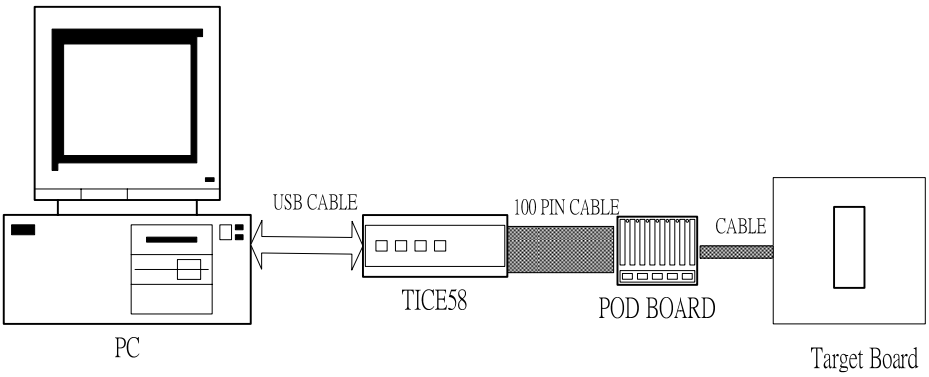
【主機外觀】

※用100pin的連接線連接TIC58主機和相應POD板。

※連接USB線以PC至TIC58主機。

※連接電源線。

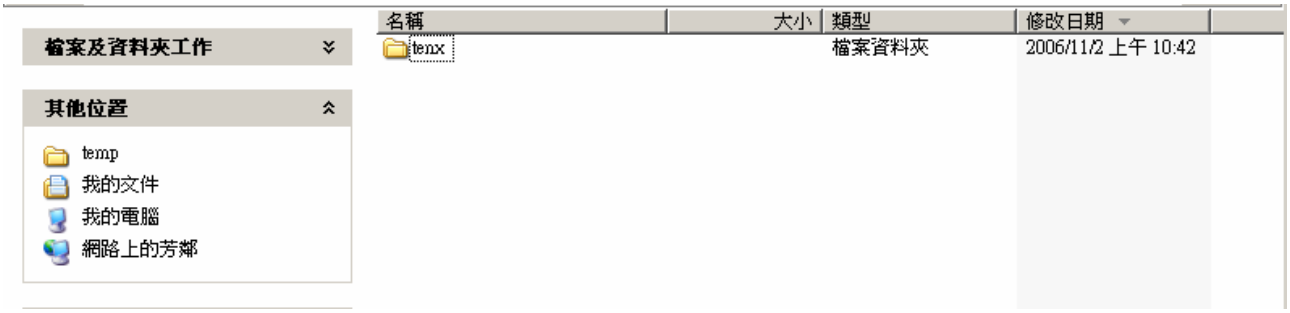
※打開電源及完成硬體安裝。



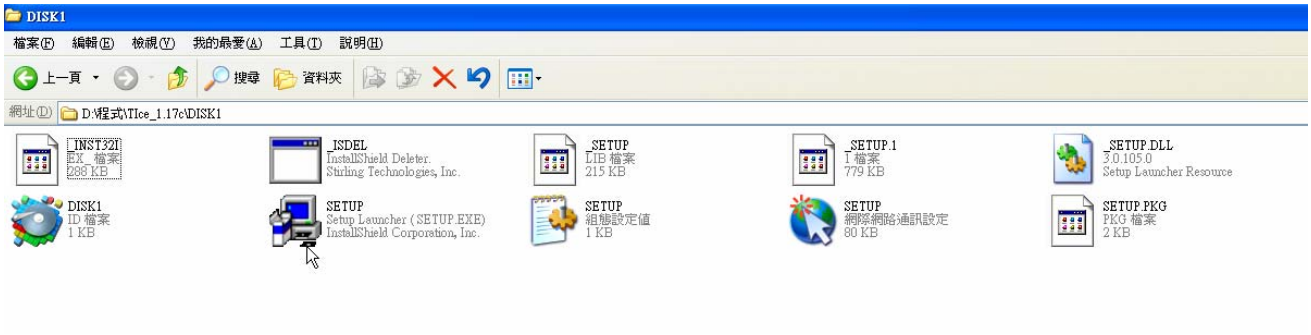
軟體安裝

【Step1】雙擊TICE58的軟體解壓縮，解壓縮完成後，會產生一個TICE58的目錄：

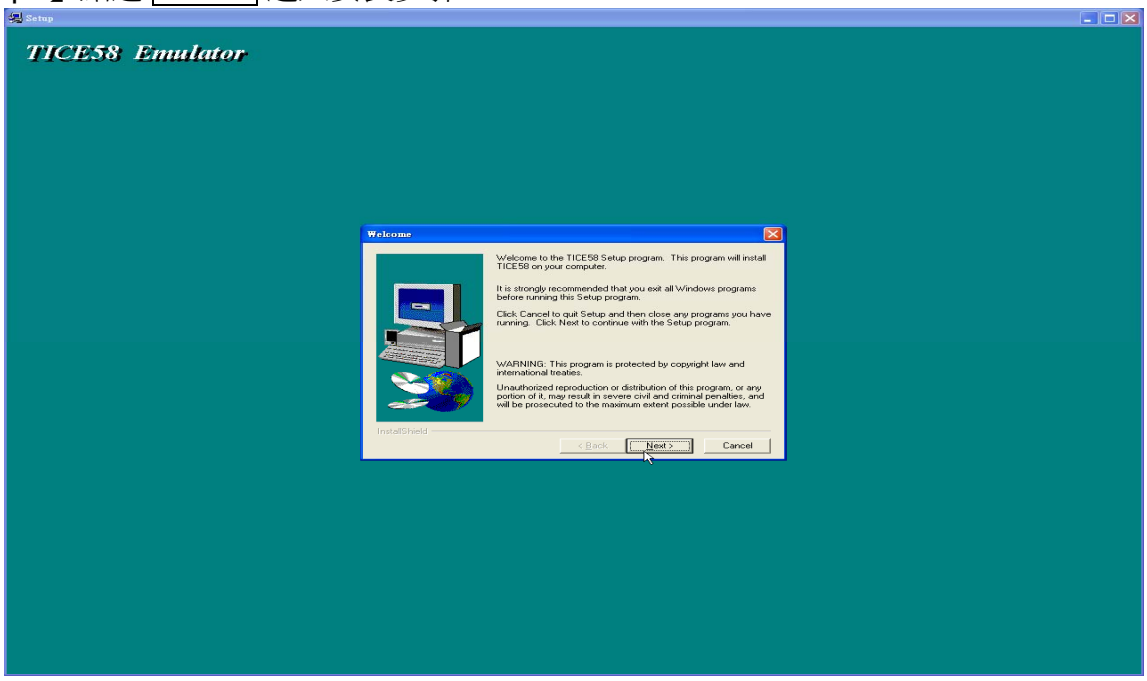
雙擊TICE58資料夾：



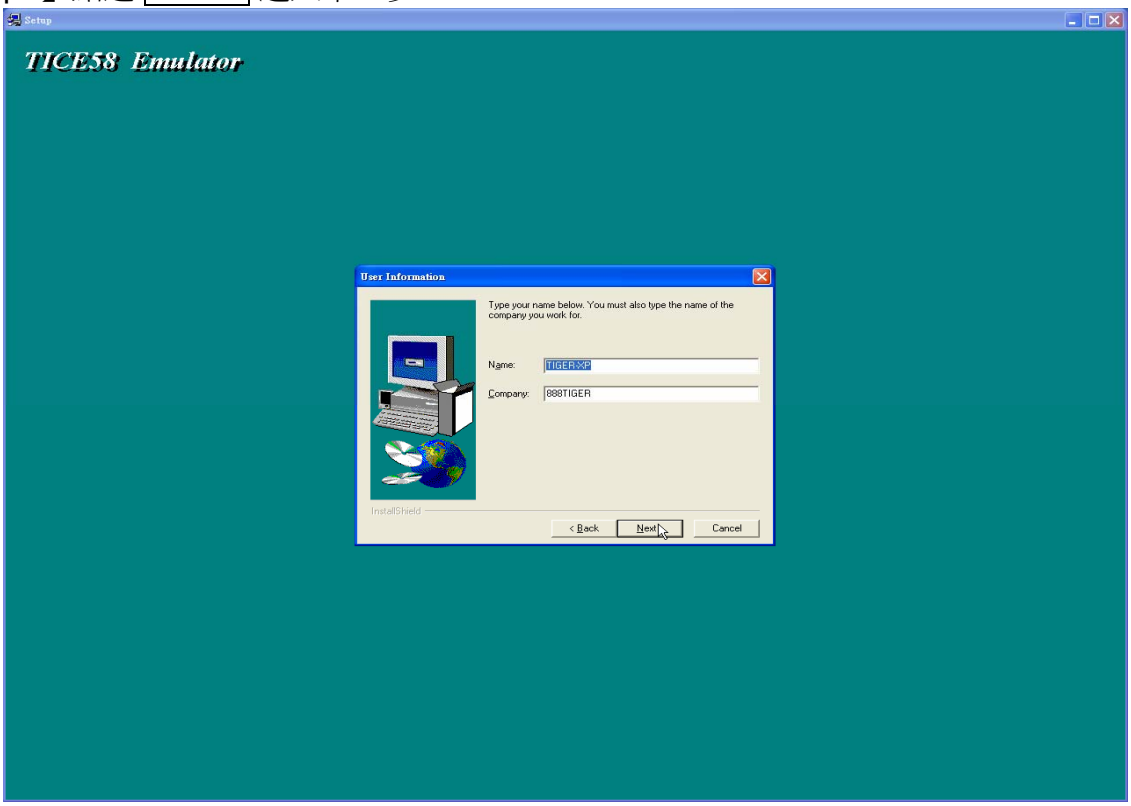
【Step2】點擊 SETUP Setup Launcher (SETUP.EXE) InstallShield Corporation, Inc. 進入下圖安裝軟體畫面：



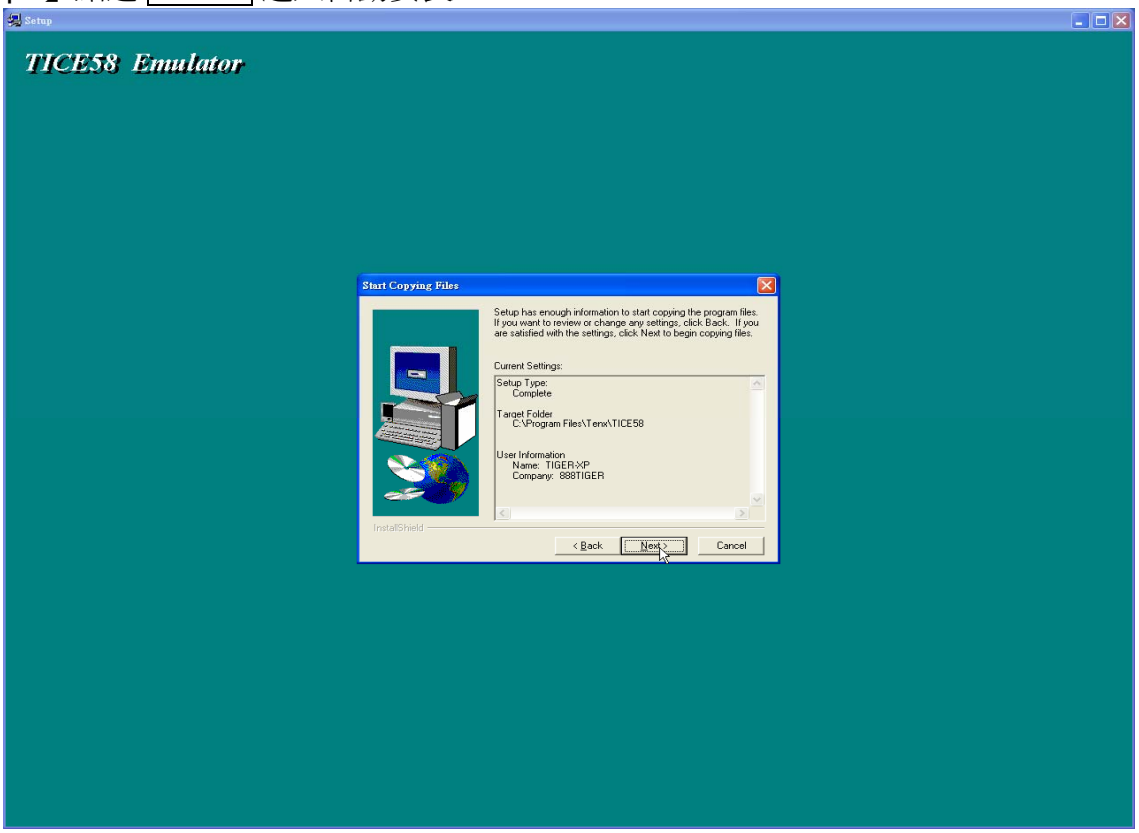
【Step3】點選 NEXT 進入安裝步驟：



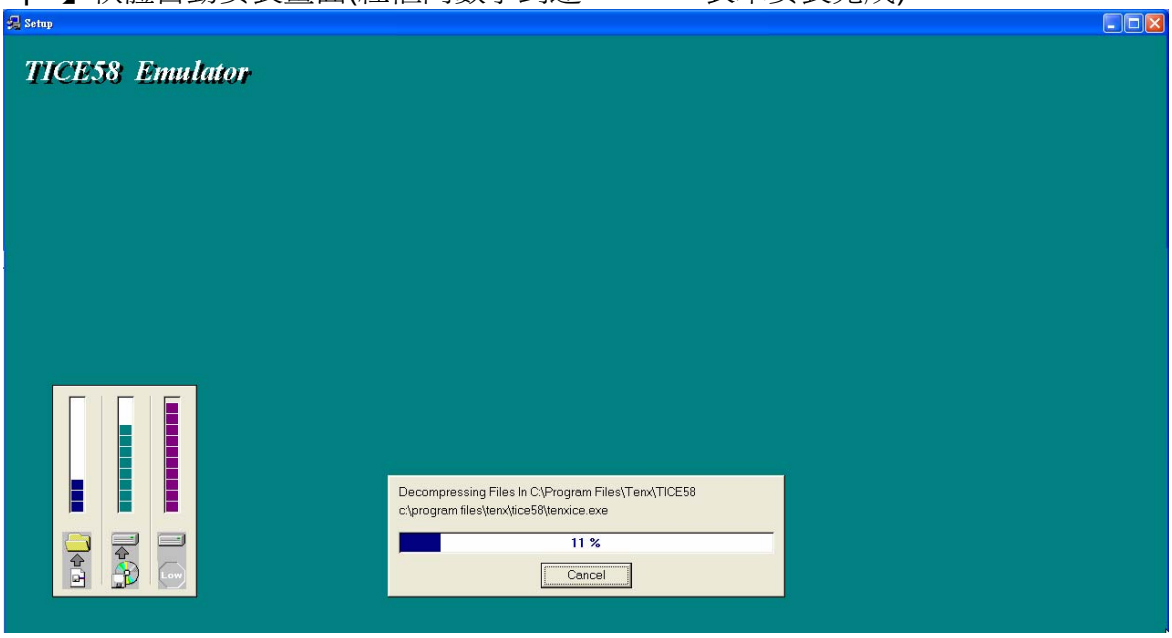
【Step4】點選 NEXT 進入下一步：



【Step5】點選 NEXT 進入自動安裝：

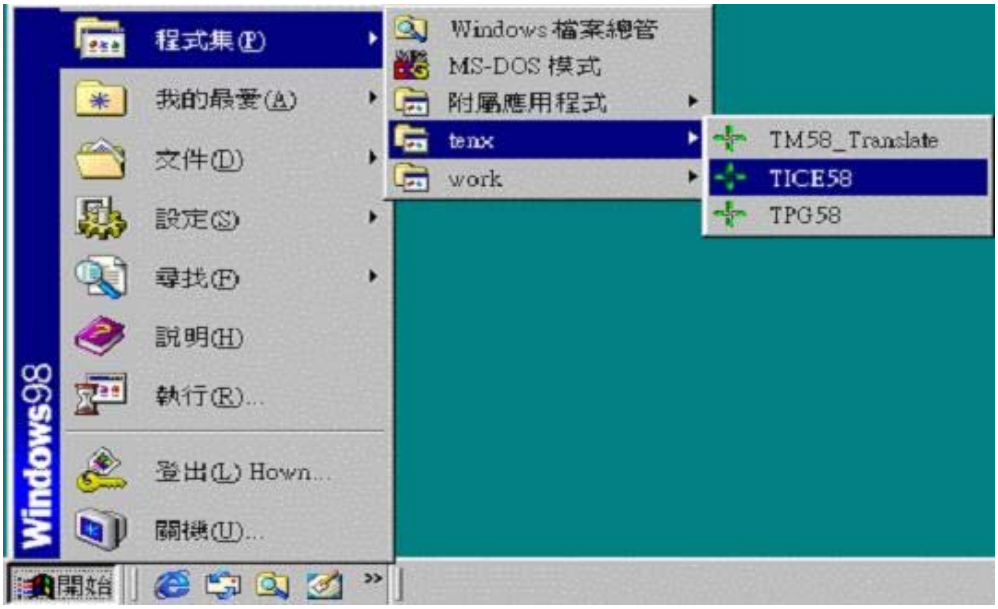


【Step6】軟體自動安裝畫面(紅框內數字到達 100%，表示安裝完成)：

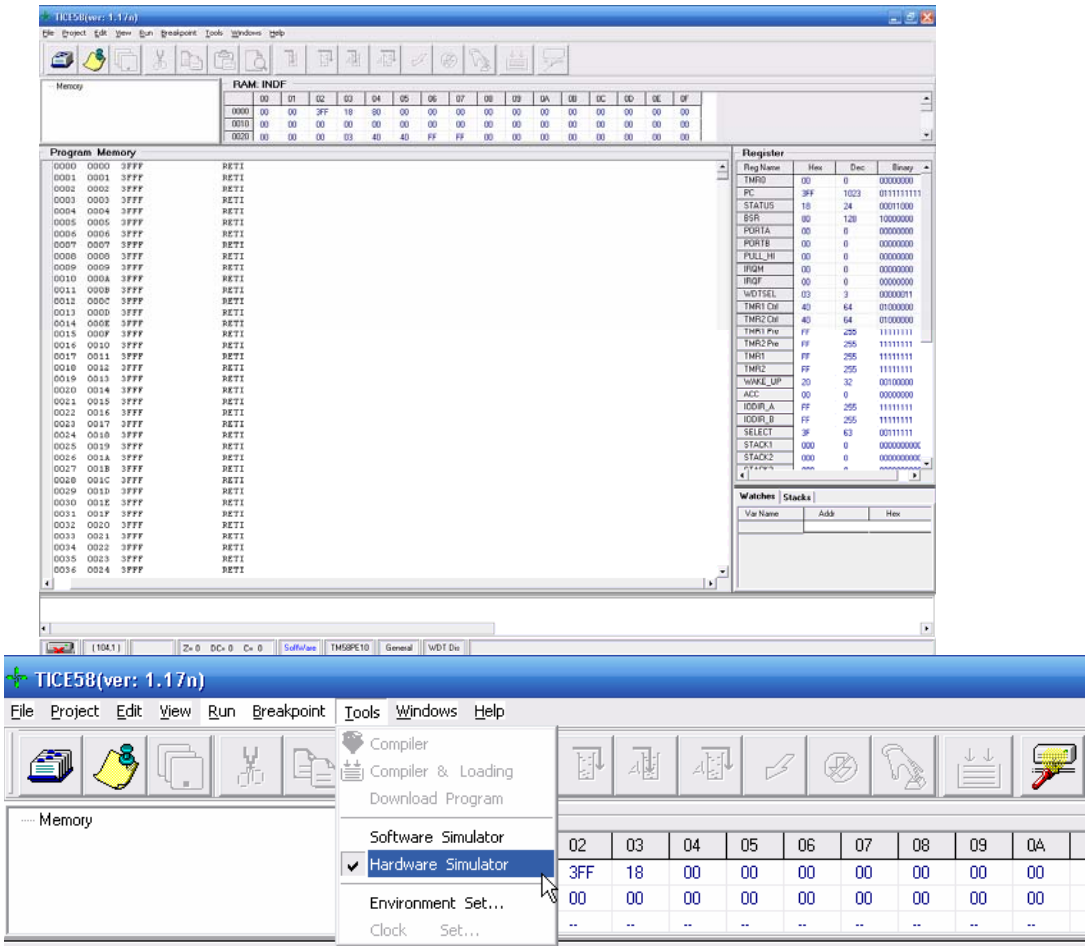


快速開始

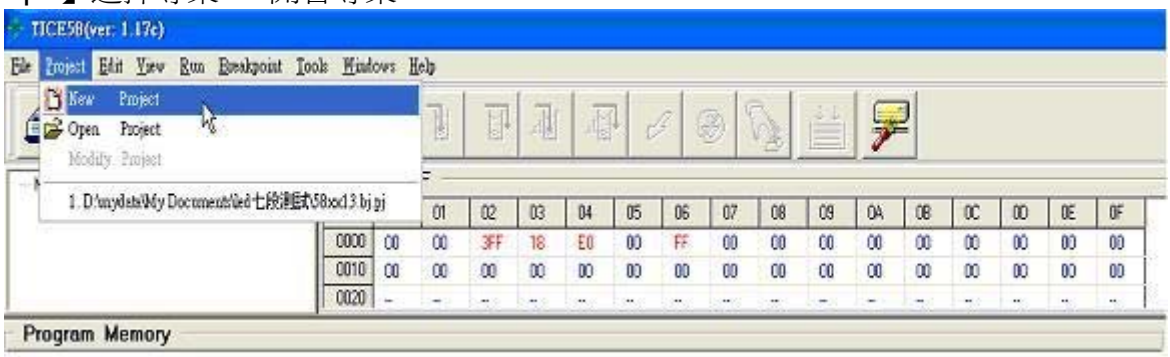
【Step1】執行 TICE58 程式



【Step2】 執行 TICE58 軟體後，出現下面視窗，設定爲硬體模擬模式：

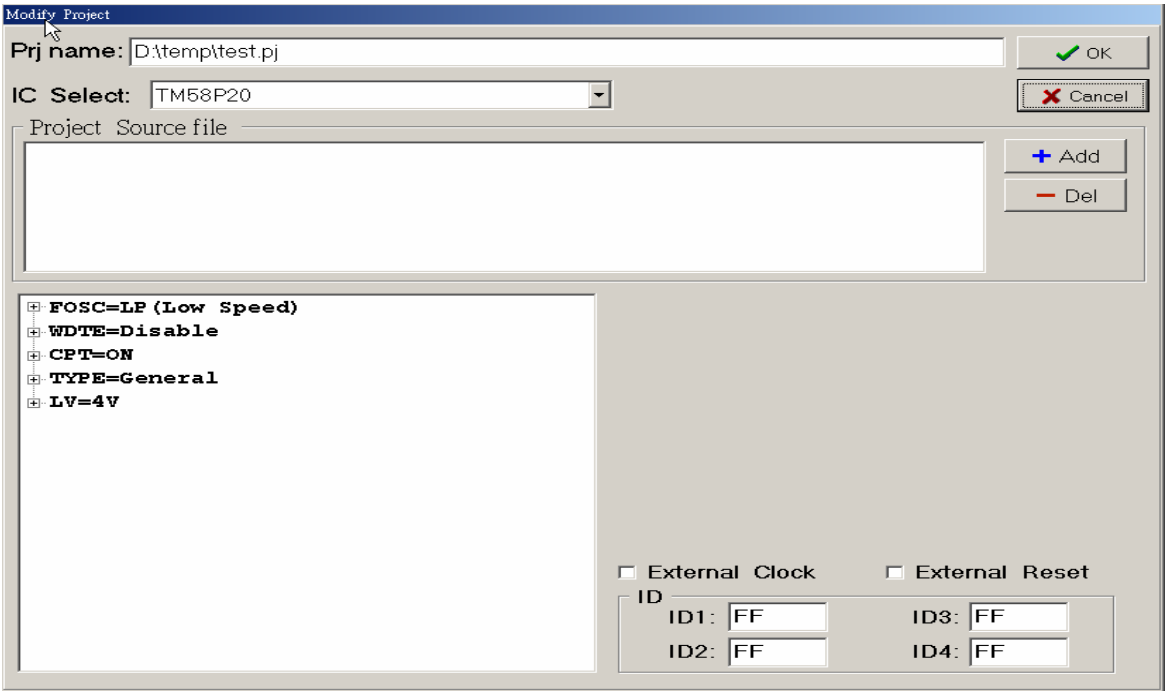


【Step3】選擇專案 → 開啓專案：

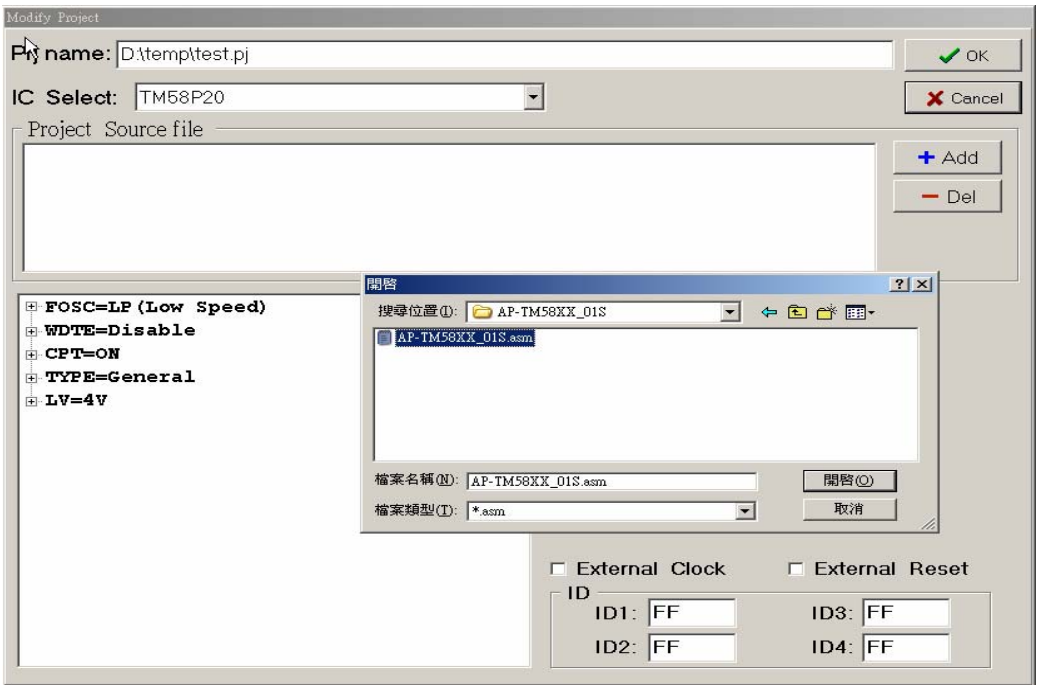


【Step4】選擇路徑，然後輸入專案的名稱(例如： “test.pj”)然後按下儲存鍵即可，如下圖所示：

【Step5】按下OK鍵後會出現選擇IC型號、CONFIGURATION設定和加入Source Code的畫面，如下圖所示，(IC型號需配合相應的POD板，有關pod板可參考注意事項說明3)：

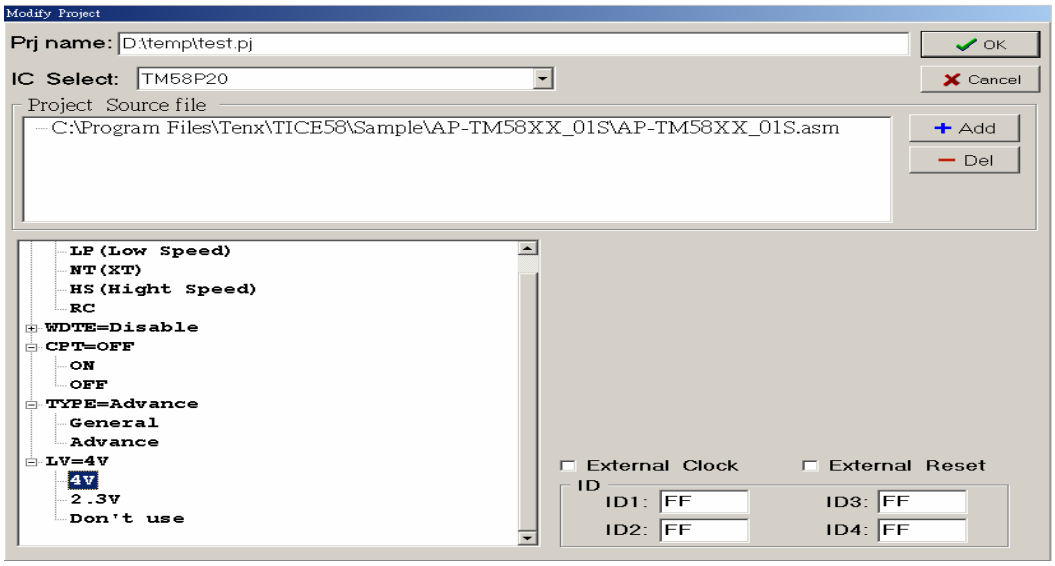


【Step6】首先請先選擇IC型號，再來點選 + Add ，加入Source Code。(在此我們加入TICE58的example的檔案，路徑爲C:\Program Files\Tenx\TICE58\example\AP-TM58XX_01S，<*C: 會因安裝不同而不同*>) 選擇AP-TM58XX_01S.asm後，按下開啓。



【Step7】結束加入Source Code後，會回到configuration 設定的畫面，如下圖。
根據example的程式，我們的設定爲：
Type: Advanced
FOSC: HS
WATCHDOG: DISABLE

※ 設定完成後，按下 OK 鍵。



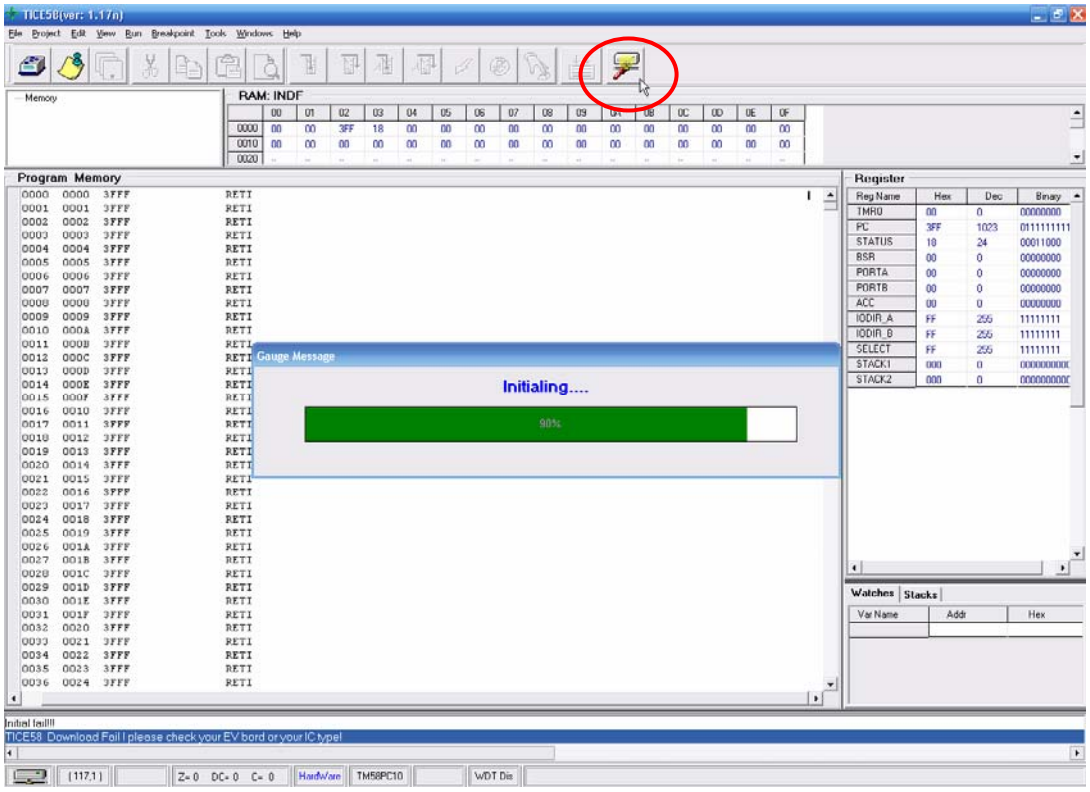
※ External Clock:
爲選擇使用內部／外部 CLOCK，沒打勾爲選擇內部 CLOCK(內部頻率固定爲 4MHz)
當選擇使用外部 CLOCK 時，需調整 TM58POD 板上的 Y1、R1、C1 與 C2 的數值，
對應表如下：

Y1	R1	C1 C2
32.768K	4.7K	4.7n
455K	4.7K	680p
1M	27K	20p
2M	20K~27K	20p
4M	20K~27K	20p
8M	1K	20p
10M	1K~470	20p
12M	470	20p
14.374M	100	20p
16M	3.3	20p

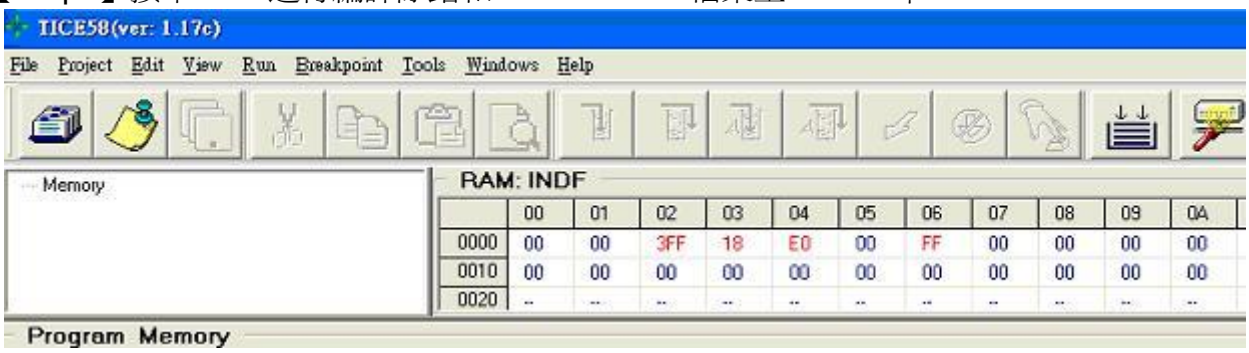
※ External Reset: 爲選擇使用 內部／外部 RESET，沒打勾爲選擇內部 RESET。

※ ID:此選項中有四個，是讓客戶自行設定 ID，預設值爲 FF。

【Step8】按下 透過 USB 與 TICE58 進行連結載入動作：



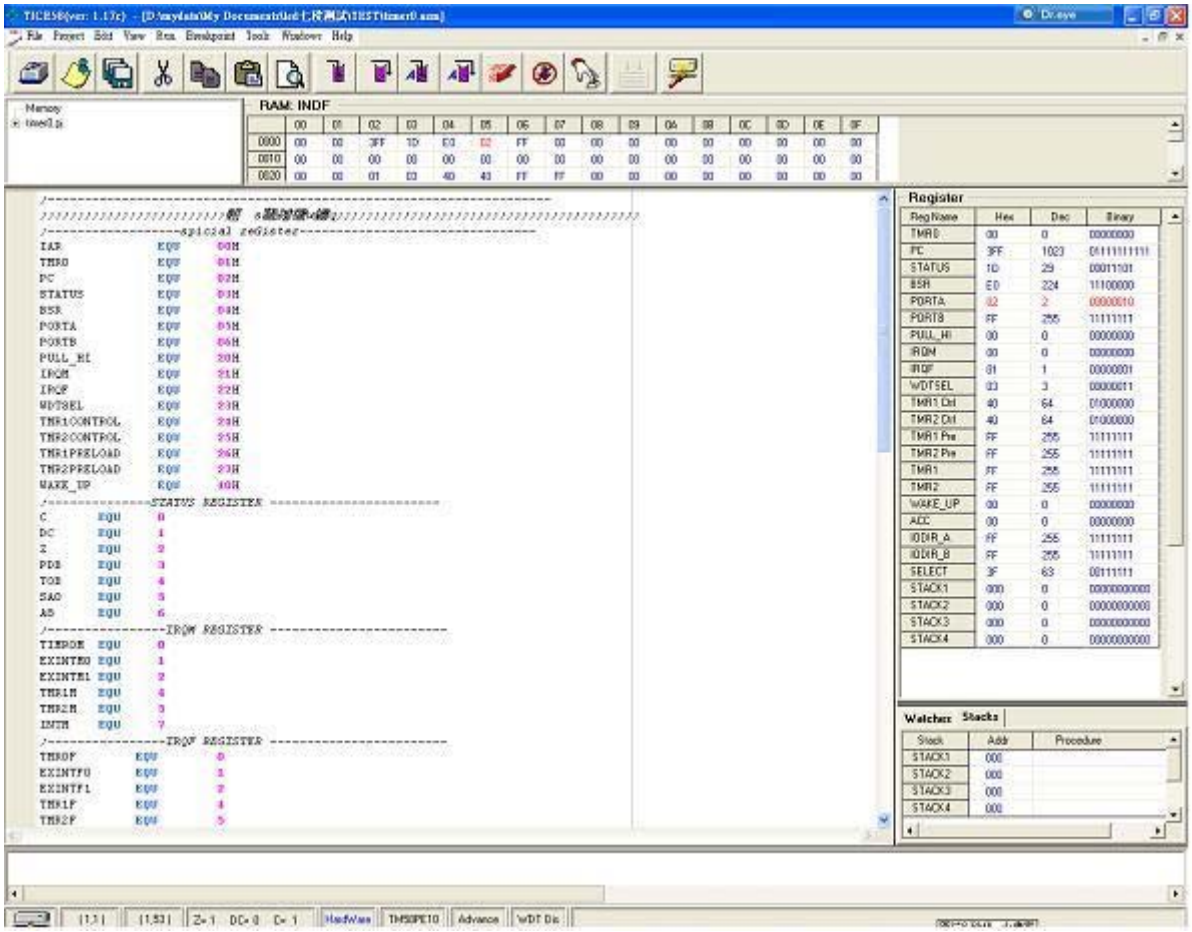
【Step9】按下進行編譯除錯和DOWNLOAD 檔案至TICE58中



※ 如果程式編譯沒有錯誤且完成程式載入的話，即可按全速執行開始執行，或是按下單步執行鍵行單步的功能。

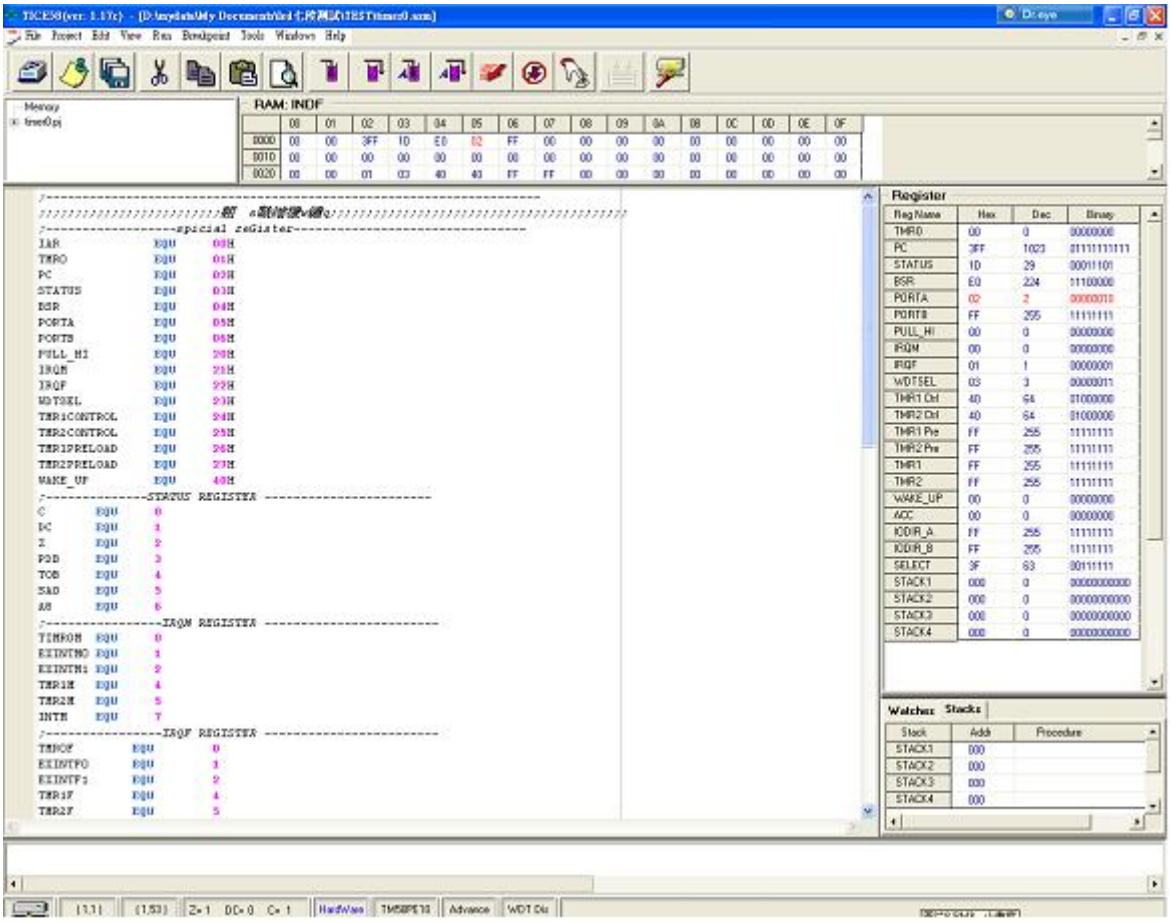
※ 如果在Download時發生錯誤，則需按下， 重新Download。

下圖是Download完成後的畫面：



選單功能

主畫面功能選單



File:

→**Create new file**：建立一個新的 ASM 檔案。
→**Open file**：開啓舊的檔案。
→**Save file** ：儲存目前的檔案。
→**Save as file**： 另存新檔。
→**Save all file** ： 儲存專案中所有的檔案。
→**Close all file** ：關閉專案中所有的檔案。
→**Printer** ： 列印檔。
→**Printer set** ：列印設定
→**Exit** ：關閉 TICE58 的程式。

Project:

→**New project**：建立新的專案。
→**Open project**：開啓舊的專案
→**Modify project**：修改專案中設定，包括 Configuration 的設定、ic 型號變更……。

Edit:

→**Copy**：複製。
→**Cut** ：剪下。
→**Paste** ：貼上。
→**Find**：尋找。
→**Replace**：取代。

View:

→**Register**：顯示 Register 的視窗。
→**Watches and Stacks** ：顯示 Watches 和 Stacks 的視窗。
→**Message**：顯示訊息的視窗。
→**RAM** ：顯示記憶體的視窗。
→**Circuit Output**(只有在軟體模擬時有用) ：請參考使用技巧 4.3
→**Remote Controlled**(只有在軟體模擬和時有用)

RUN:

→**Free run**：全速執行程式。
→**Auto step into** → 自動單步執行(所有程式)。
→**Auto step over**：自動單步執行(只執行程式)。
→**Step over** ：單步執行(所有程式)。
→**Setp into** ：單步執行(只執行程式)。
→**Run to cursor**：程式只執行到游標處。
→**Stop**：停止。
→**Warm reset**：重置。

BreakPoint:

→**Break Point List**：中斷點清單。

Tools:

→**Compiler**：編輯
→**Compiler & Loading** (只有在硬體模擬時有用) ：編輯完成後，
下載程式到 EV。
→**Download Program** (只有在硬體模擬時有用) ：下載程式到 EV。
→**Software Simulator** ：使用軟體模擬，不需 POD 板。(部份功能無法使用)
→**Hardware Simulator**：使用硬體模擬，必須連接 POD 板。
→**Environment Set** ：操作環境設定。

主畫面工具列

按鍵圖示	功能鍵	備註	
		開啓舊專案	開啓上一次或是之前的專案
		建立新的來源檔	建立新的來源檔(如 .asm)
		儲存專案中所有的檔案	儲存專案中所有的檔案
		剪下	剪下
		貼上	貼上
		尋找	尋找
	F7	單步執行 (所有程式)	單步執行所有的程式
	F8	單步執行 (只執行主程式)	單步執行, 不執行 CALL 指令 (只執行主程式)
	F10	自動單步執行 (所有程式)	自動單步執行所有的程式
	F11	自動單步執行 (只執行主程式)	自動單步執行所有的程式，但不執行 CALL 指令
	F9	執行程式	全速執行程式
	F3	停止	停止執行
	F4	重置	將程式回到最初的起始位置
		執行 Compiler 跟 Download	執行 Compiler,並在沒有任何錯誤的情況下 Download 檔案到 ICE。
		重新 Initial ICE	重新 Initial ICE。

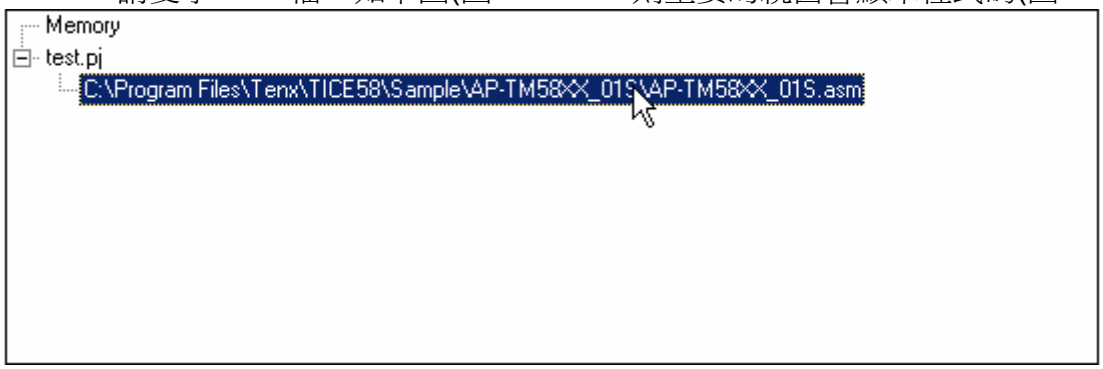
主畫面視窗

【專案檔案】

此視窗主要可分為來源檔的顯示或是 Memory 格式的顯示。使用者可依自已的習慣設定顯示的資訊。

a. 程式碼(*.asm)的顯示

請雙擊.asm 檔，如下圖(圖 3.3.1-1，則主要的視窗會顯示程式碼(圖 3.3.1-2)



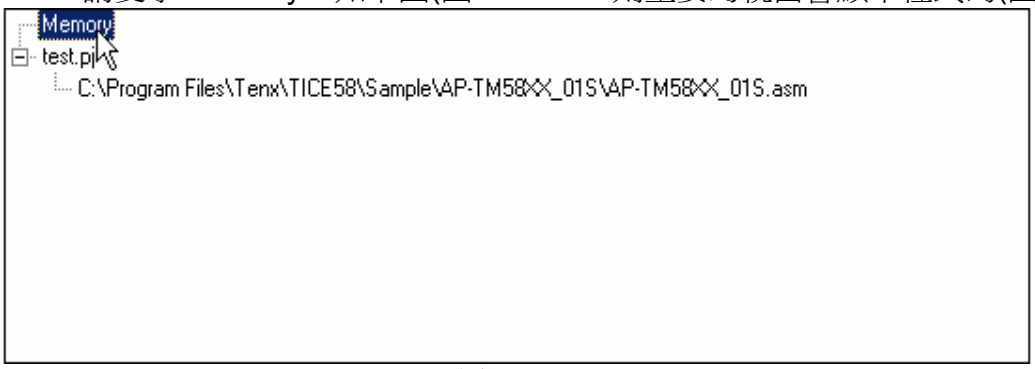
(圖 3.3.1-1)

```
;==========================================
;==========================================
org     7ffh     ;重定向量地址
goto            main
org     7feh     ;中斷副程式入口位址
goto    extint
org     000h     ;程式在ROM中的起始地址
;
;
;==========================================
;tmr0時間處理副程式
;==========================================
readtime:
movm    tmr0,w           ;讀取tmr0的值,結果放入累加器中
xoram   time_temp,w   ;tmr0的值與time_temp值相異或,結果放入累加器中
movam   time_buf         ;將tmr0與time_temp異或的結果送入time_buf中
xoram   time_temp,f     ;tmr0的值送入time_temp中
btmss   time_buf,7       ;判斷128*64us/1000=8.192ms是否已到
ret                      ;8.192ms未到，則退出副程式
btmsc   del_flag,0               ;8.192ms已到,判斷是否處於遮罩外中斷狀態
incm    del_cnt   ;處於遮罩外中斷狀態,則del_cnt值加1
incm    count1           ;count1值加1

movla   3dh                  ;判斷61*8.192ms=499.712ms約?0.5S是否已到
xoram   count1,w
btmss   status,z
ret     ;499.712ms未到，則退出副程式
clrm    count1    ;499.712ms已到，清count1
btmsc   led_flag,led1     ;判斷led1控制標誌位元是否?1？
goto            turn_on_led1     ;led1控制標誌位元?1,轉去開led1
btmsc   led_flag,led2     ;led1控制標誌位元不?1,判斷led2控制標誌位元是否?1？
```

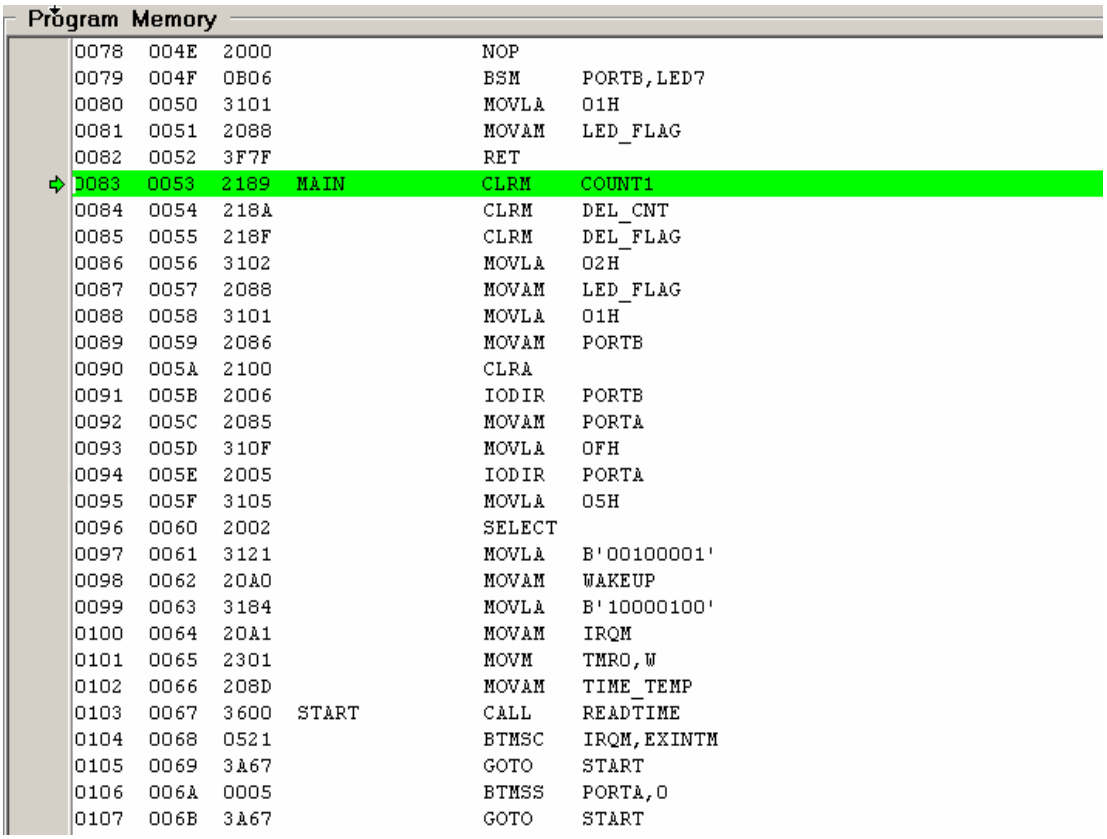
(圖 3.3.1-2)

b. 機器碼的顯示

請雙擊 Memory，如下圖(圖 3.3.1-3，則主要的視窗會顯示程式碼(圖 3.3.1-4)



(圖 3.3.1-3)



Program Memory

```
0078  004E  2000                NOP
0079  004F  0B06                BSM     PORTB,LED7
0080  0050  3101                MOVLA   01H
0081  0051  2088                MOVAM   LED_FLAG
0082  0052  3F7F                RET
0083  0053  2189  MAIN          CLRM    COUNT1
0084  0054  218A                CLRM    DEL_CNT
0085  0055  218F                CLRM    DEL_FLAG
0086  0056  3102                MOVLA   02H
0087  0057  2088                MOVAM   LED_FLAG
0088  0058  3101                MOVLA   01H
0089  0059  2086                MOVAM   PORTB
0090  005A  2100                CLRA
0091  005B  2006                IODIR   PORTB
0092  005C  2085                MOVAM   PORTA
0093  005D  310F                MOVLA   0FH
0094  005E  2005                IODIR   PORTA
0095  005F  3105                MOVLA   05H
0096  0060  2002                SELECT
0097  0061  3121                MOVLA   B'00100001'
0098  0062  20A0                MOVAM   WAKEUP
0099  0063  3184                MOVLA   B'10000100'
0100  0064  20A1                MOVAM   IRQM
0101  0065  2301                MOVM    TMR0,W
0102  0066  208D                MOVAM   TIME_TEMP
0103  0067  3600  START         CALL    READTIME
0104  0068  0521                BTMSC   IRQM,EXINTM
0105  0069  3A67                GOTO    START
0106  006A  0005                BTMSS   PORTA,0
0107  006B  3A67                GOTO    START
```

(圖 3.3.1-4)

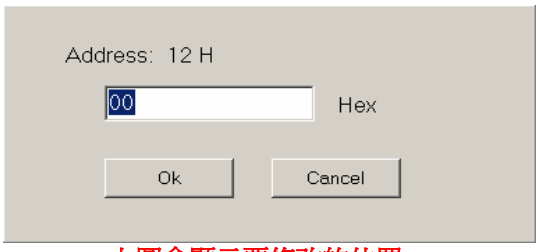
【RAM】

顯示 IC RAM 的資訊，對應位置的意義請參考每顆 IC 的 Data Sheet。

RAM: PC

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	00	78	53	1C	00	0F	01	FF	02	00	00	00	00	00	00	00
0010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0020	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0040	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
0050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0060	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
0070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

※ 在 Debug 中，如需要修改，只需在欲修改的位置，點兩下(如下圖)後，輸入數值，按下 OK 鍵即可。



上圖會顯示要修改的位置。

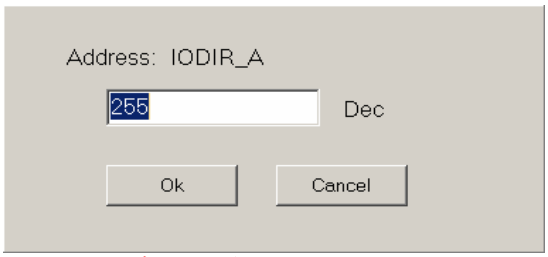
【Register】

顯示 IC Register 的資訊，每個 Register 的意義請參考每顆 IC 的 DataSheet。

Register

Reg Name	Hex	Dec	Binary
TMR0	78	120	01111000
PC	053	83	00001010011
STATUS	1C	28	00011100
BSR	00	0	00000000
PORTA	0F	15	00001111
PORTB	01	1	00000001
PORTC	FF	255	11111111
ACC	00	0	00000000
IODIR_A	FF	255	11111111
IODIR_B	FF	255	11111111
IODIRC	FF	255	11111111
SELECT	FF	255	11111111
STACK1	000	0	00000000000
STACK2	000	0	00000000000
STACK3	000	0	00000000000
STACK4	000	0	00000000000

※ 在 Debug 中，如需要修改，只需在欲修改的位置，點兩下(如下圖)後，輸入數值，按下 OK 鍵即可。

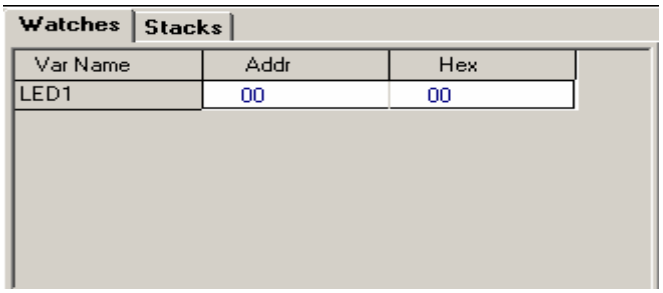


上圖會顯示要修改的 **Register**。

【Watches/Stacks】

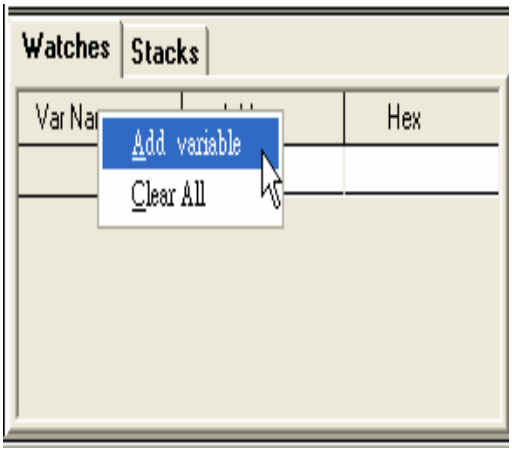
此視窗主要分成 Watches 和 Stacks。

1. Watches 是讓使用者方便追蹤程式中定義的變數數值的變化。

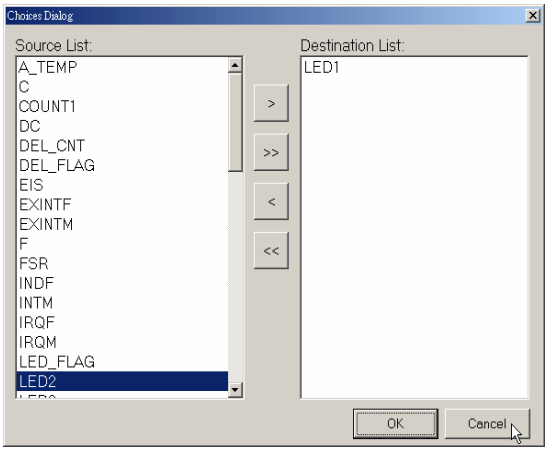


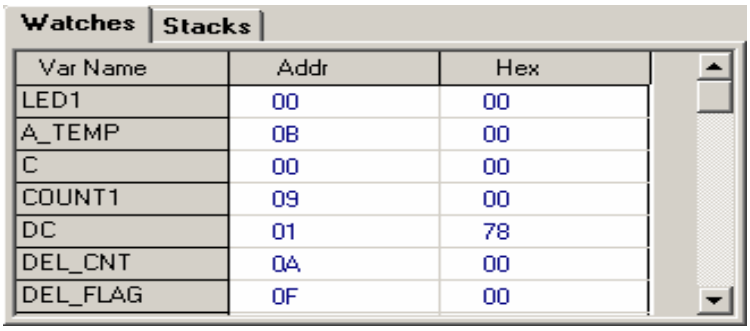
※ 加入方法如下：

(1). 在 Var Name 的位置按滑鼠的右鍵 (如下圖)。

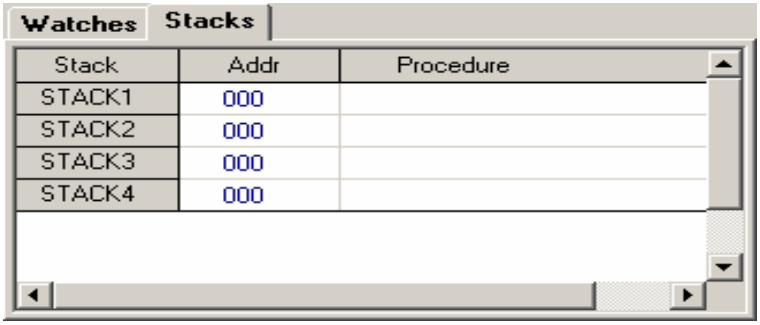


(2). 點選 Add variable 後，會出現下面的視窗，加入欲觀察的變數，按 ok 鍵。



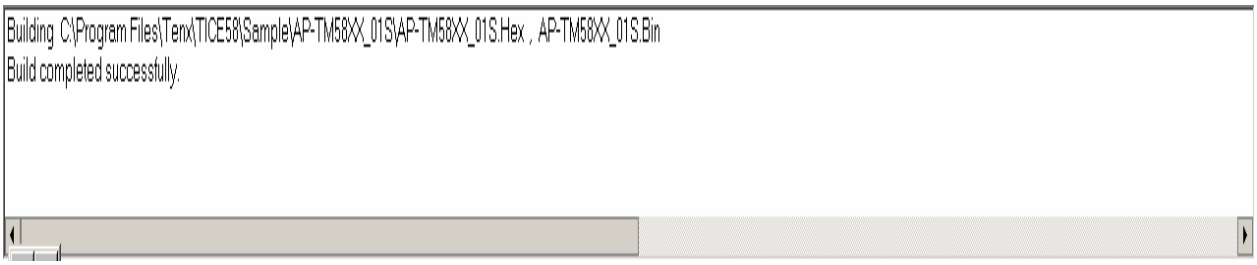


2. Stacks:是讓使用者觀察堆疊內數值。



【操作訊息】

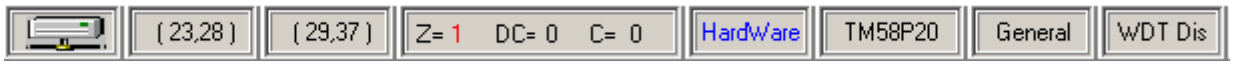
顯示操作、編輯.....等相關訊息。



【其它訊息】

如下圖所示：

A、硬體模擬



→ 代表和 TICE58 的主機連線。

(23,28) → 當主畫面是在機器碼的顯示下，游標的位置。

(29,37) → 當主畫面是在程式碼的顯示下，游標的位置。

Z= 1 DC= 0 C= 0 → 各個旗標暫存器的值。
(旗標暫存器的意義，請參考 IC 的 DataSheet)

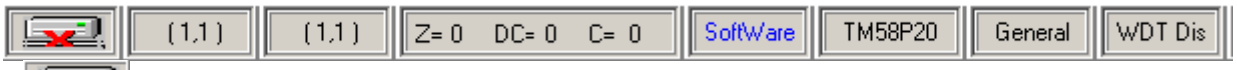
HardWare → 目前是使用硬體模擬。

TM58P20 → 目前所要模擬的 ic 型號。

General → TM58P20 Configuration 中 TYPE 所設定的值。
(每顆 ic 都提供不同的 Configuration，所以請參考 ic 的 DataDheet)

WDT Dis → WatchDog 的功能關閉。

B、軟體模擬

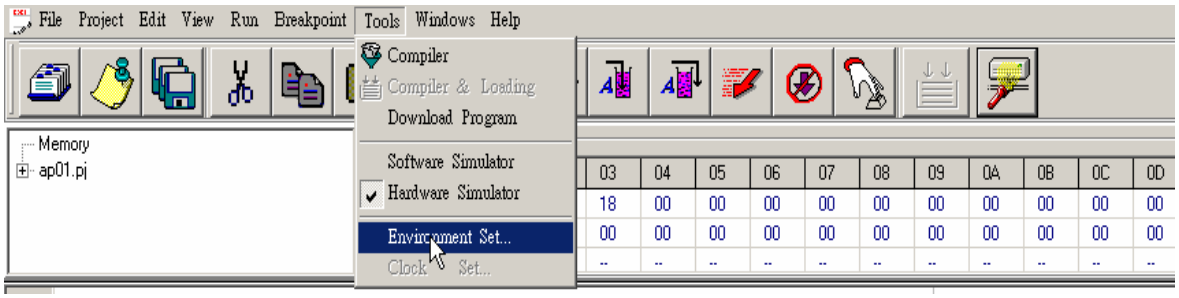


→ 無法和 TICE58 的主機連線。

SoftWare → 目前是使用軟體模擬。

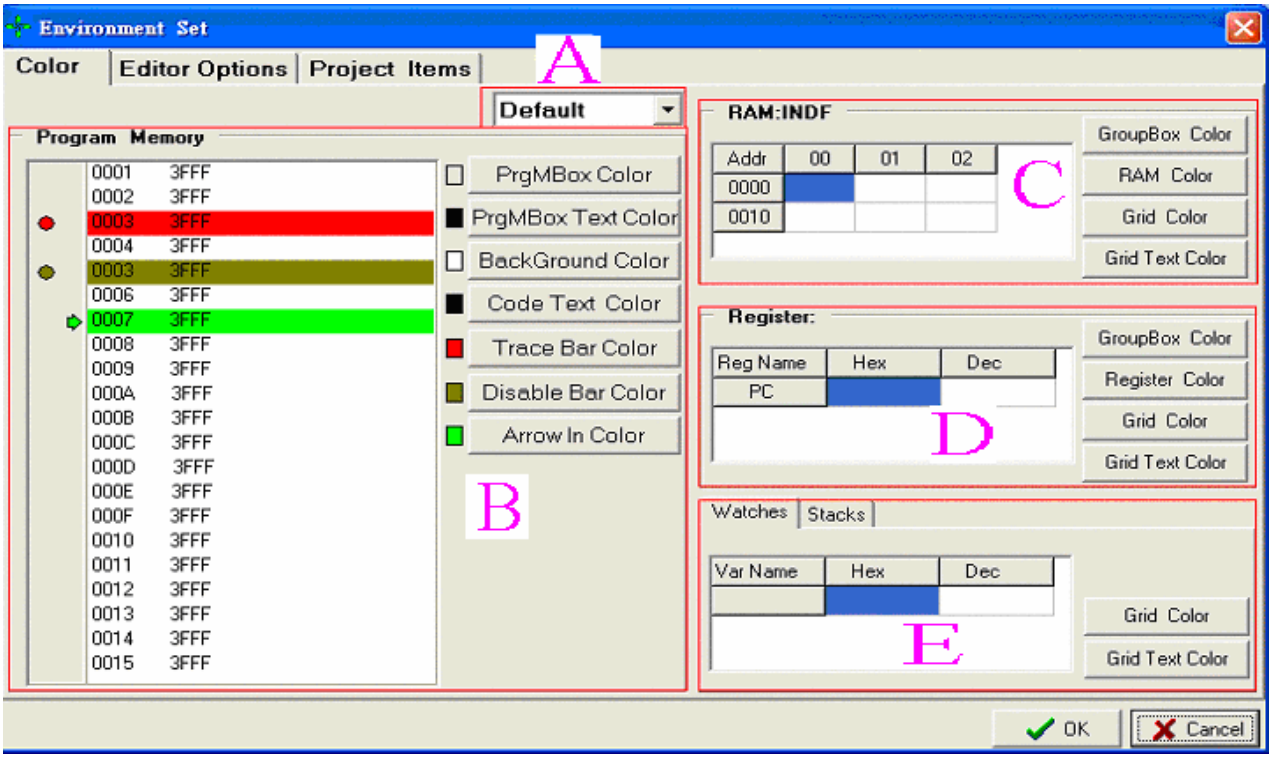
環境設定視窗

※ 在 Tools→ Environment set ……



A、Color

出現下面視窗，可以對於特性功能做選擇顏色的更改。



A: 設定主畫面背景顏色

B: PrgMBox Color 設定外框顏色

PrgMBox Text Color 設定上方文字顏色

BackGround Color 設定程式框背影顏色

Code Text Color 設定程式顏色

Trace Bar Color 設定 enable 時斷點的顏色

Disable Bar Color 設定 disable 時斷點的顏色

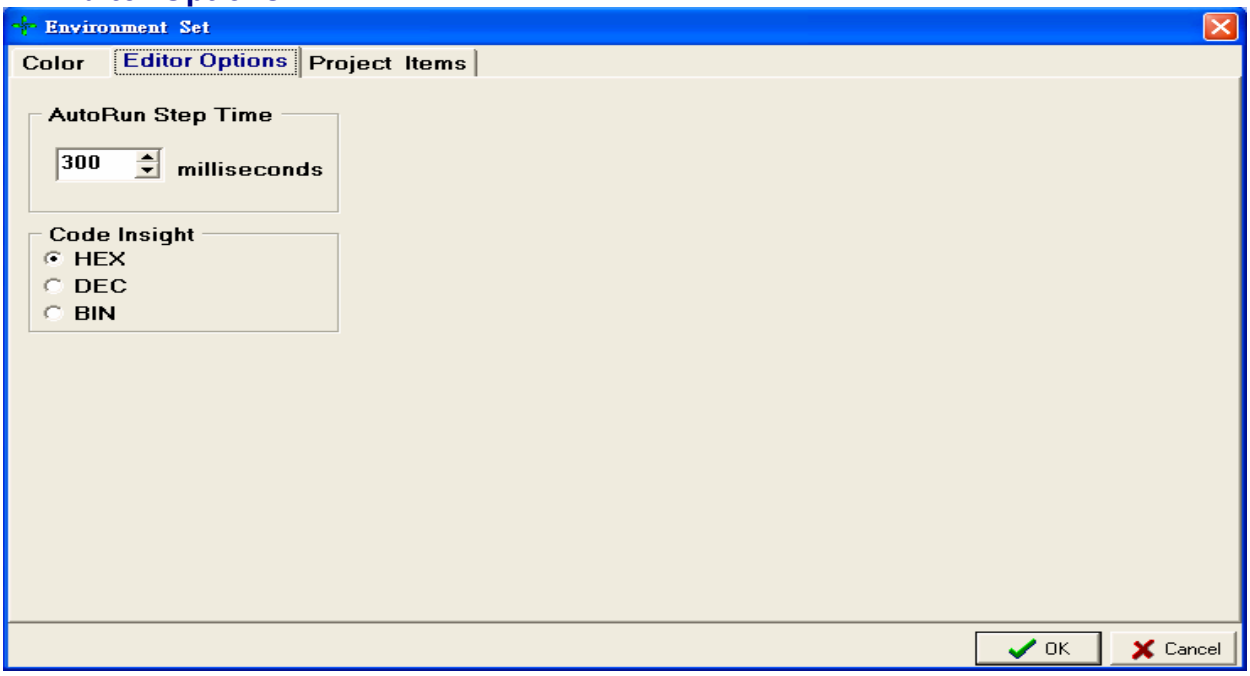
Arrow In Color 設定程式執行到某位址的顏色

C: 調整 RAM 視窗顏色

D: 調整 Register 視窗顏色

E: 調整 Watches 和 Stacks 視窗顏色

B、Editor Options



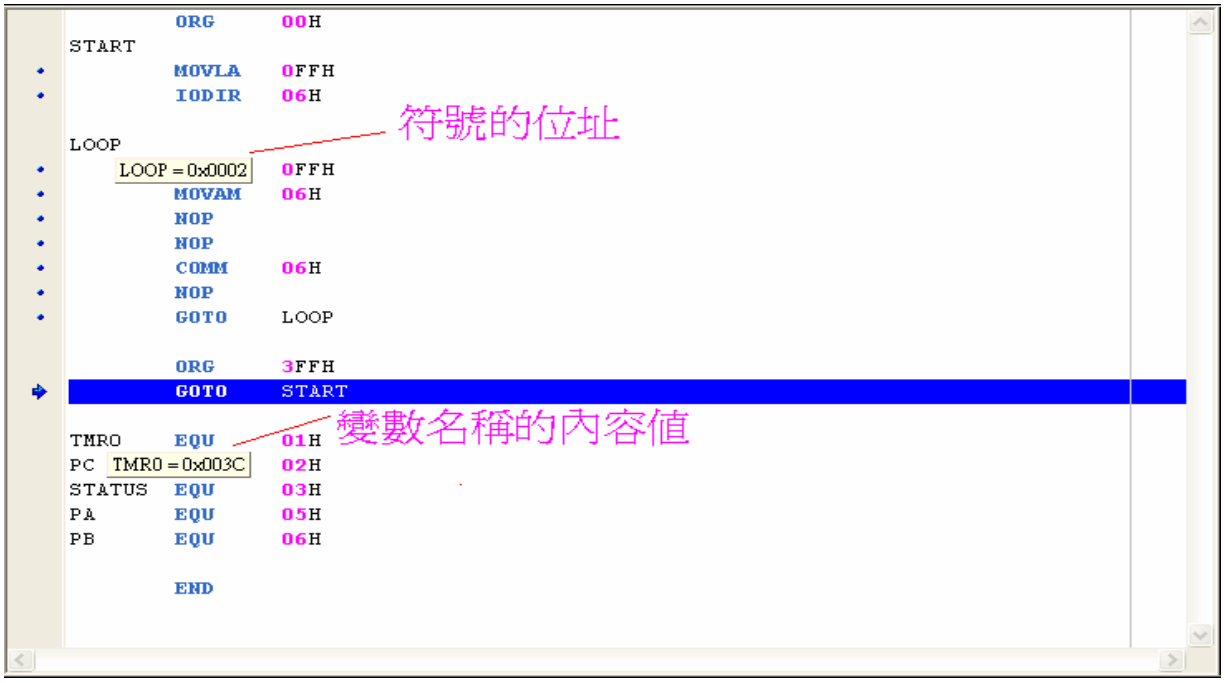
※AutoRun Setup Time:

設定當執行自動單步時，從一行指令跳至下一行指令的時間(單位:毫秒)。

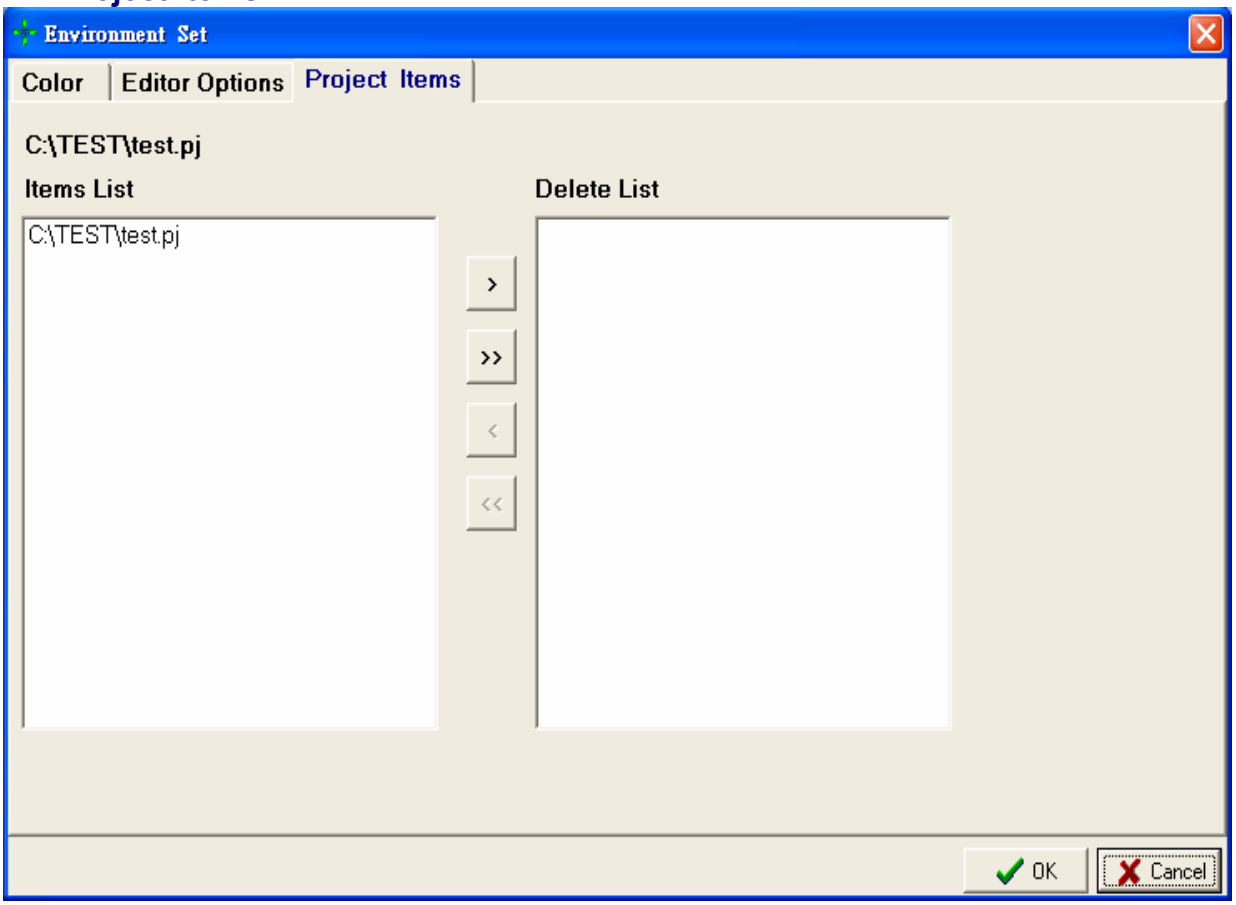
※Code Insight:

此選項只有在主畫面是在來源視窗下有用，用於顯示符號的位址和變數名稱的內容值。

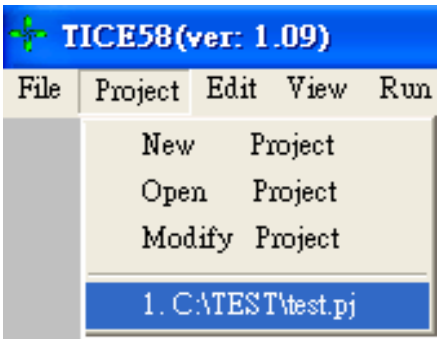
※顯示的方式有三種：16 進制、十進制、二進制。



C、Project items



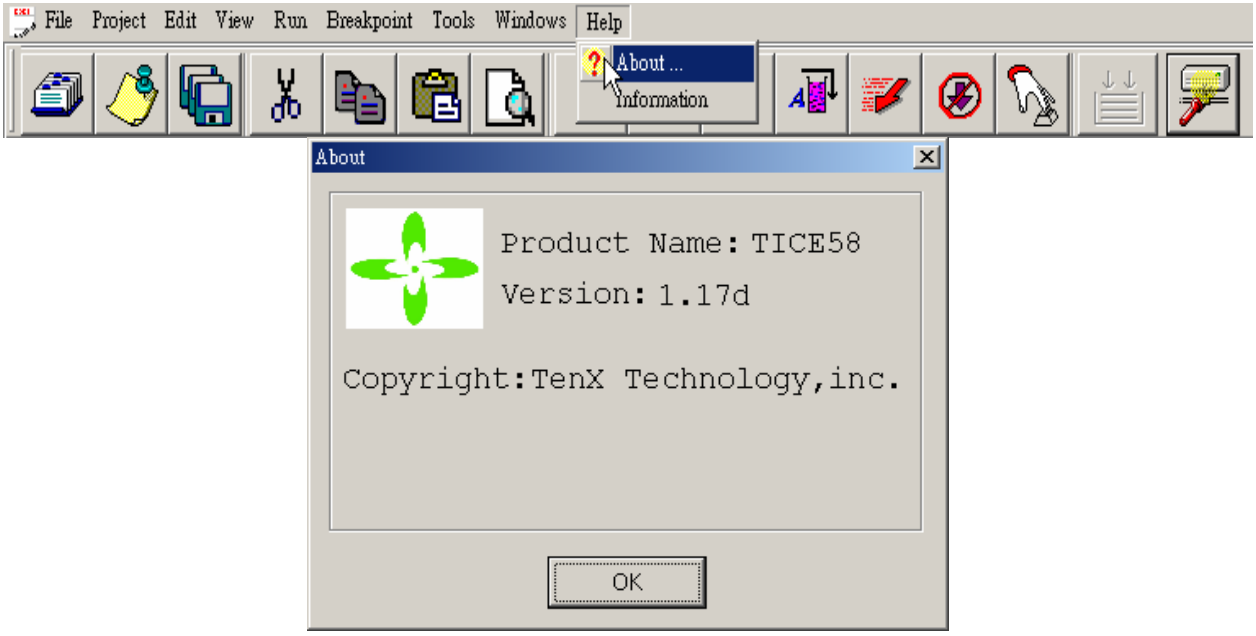
※ 此選項是在於當每次開啓不同的 project 時，會在"Project"(如下圖)中增加一快速開啓的選擇，當有些不想再用的*.pj 時，可由此處移除(在 Project 中最多能紀錄 5 個*.pj)



幫助視窗

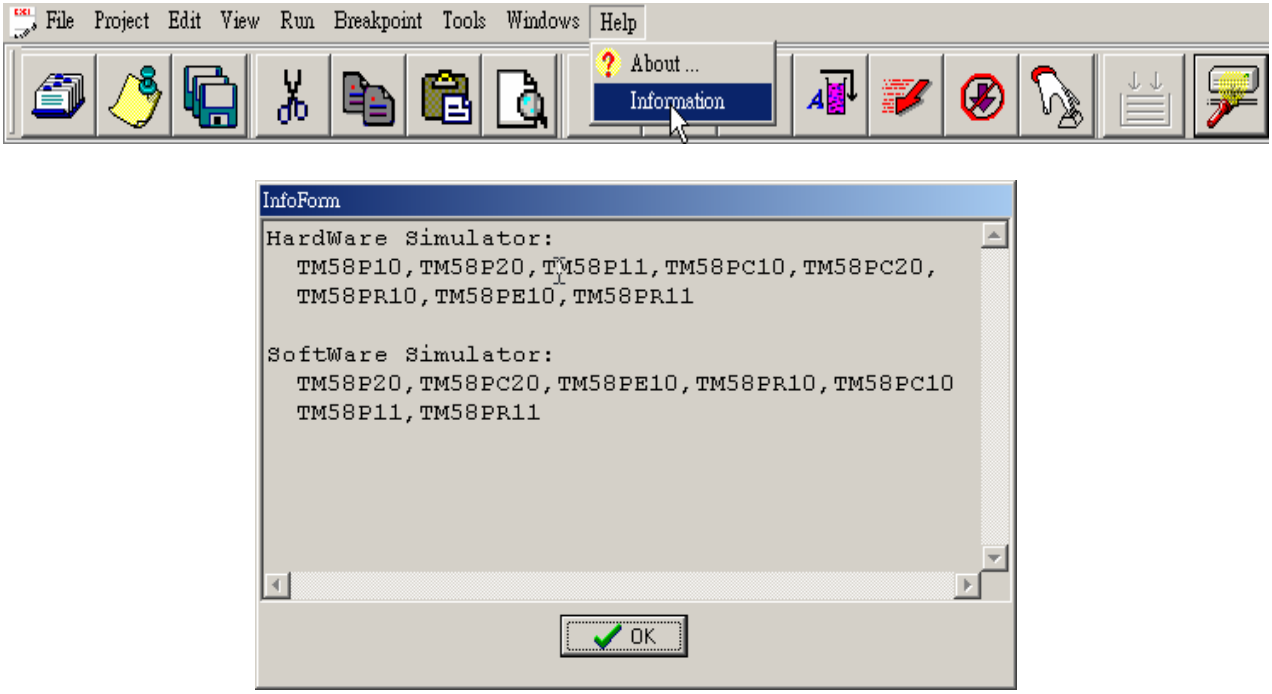
A、About

顯示 TOCE58 的版本。



B、Information

顯示硬體模擬和軟體模擬所支援的 IC 型號



使用技巧

中斷點的使用

爲了更方便使用者進行 Debug，TICE58 的軟體提供數種讓程式中斷的方法，分別如下：

- 使用 F2 鍵
- 使用中斷點
- 使用 Run to Cursor
- 使用 Break Point List
- 主畫面(中斷點發生時)

爲了更方便使用者進行 Debug，TICE58 的軟體提供數種讓程式中斷的方法，分別如下：

- 使用 F2 鍵
- 使用中斷點
- 使用 Run to Cursor
- 使用 Break Point List
- 主畫面(中斷點發生時)

```
clra                                    ;
iodir   portb   ;將portb口設?輸出態
movam   porta           ;porta口輸出?0
movla   0fh             ;
iodir   porta   ;將porta口設?輸入態
movla           05h                     ;
select                          ;tmr0採用內部時鐘計數,預分頻器分配給tmr0,
        ;分頻比?1:64
movla   b'00100001'     ;
movam   wakeup          ;設定porta,0口?外中斷輸入口
movla   b'10000100'     ;
movam   irqm                    ;開放外中斷功能
movm    tmr0,w          ;
movam   time_temp       ;讀tmr0的值結果送入time_temp中
;
;
start:;主迴圈
call    readtime        ;調tmr0時間處理副程式
btmsc   irqm,exintm     ;判斷外中斷功能是否關閉
goto    Start                   ;未關閉外中斷功能,返回start處執行
btmss   Porta,0             ;外中斷功能關閉,判斷按建是否?起。
goto    start                   ;按鍵未?起,返回start處執行
bsm     del_flag,0      ;按鍵?起,置控制延時標誌del_flag
movla   05h
xoram   del_cnt,w
btmss   status,z                ;判斷延時消抖時間5*8.192ms=40.96ms到否
goto    start                   ;延時消抖時間未到,返回start處執行
clrm    del_cnt             ;延時消抖時間到,清del_cnt
btmss   porta,0     ;再次確認按鍵是否真的?起
goto    start   ;按鍵未?起,返回start處執行
bcm     del_flag,0      ;按鍵確實?起,清控制延時標誌del_flag
bcm     irqf,exintf     ;清外中斷請求標誌位元
bsm     irqm,exintm     ;開放外中斷功能
goto    start           ;返回start處執行
;
;
;==========================================
;外中斷回應副程式
;==========================================
```

使用滑鼠右鍵設置，在要設斷點的位置按滑鼠右鍵，選擇“Set BreakPoint”設斷點，如要取消，將滑鼠移至要取消的地方按右鍵後選擇“Clear BreakPoint”取消

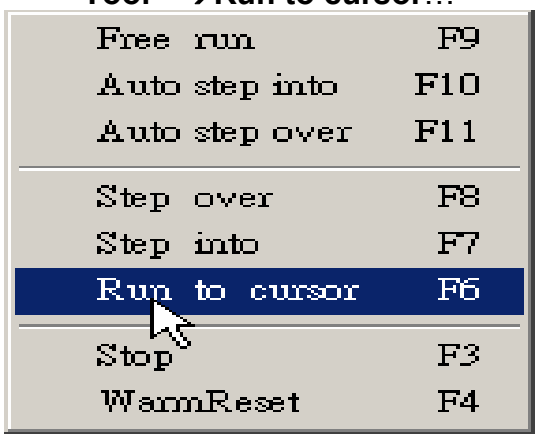
```
            ORG     00H
START
            MOVLA   0FFH
            IODIR   06H

LOOP
            MOVLA   0FFH
            MOVAM   06H
            NOP
            NOP
            COMM    06H
            Cut
            Copy    LOOP
            Paste
            Set BreakPoint  3FFH
            Clear BreakPoint  START
            Run to cursor   01H
PC          EQU     02H
STATUS      EQU     03H
PA          EQU     05H
PB          EQU     06H

            END
```

```
            ORG     00H
START
            MOVLA   0FFH
            IODIR   06H

LOOP
            MOVLA   0FFH
            MOVAM   06H
            NOP
            NOP
            COMM    06H
            Cut
            Copy    LOOP
            Paste
            Set BreakPoint  3FFH
            Clear BreakPoint  START
            Run to cursor   01H
PC          EQU     02H
STATUS      EQU     03H
PA          EQU     05H
PB          EQU     06H

            END
```

【使用 **Run to Cursor**】

Run to cursor 就是讓程式執行到游標所在的位置。

Tool---→Run to cursor…



※ 將游標移到要停止的程式碼上(紅色箭頭所指)，如下：

```
movla   b'00100001'   ;
movam   wakeup        ;設定porta,0口?外中斷輸入口
movla   b'10000100'   ;
movam   irqm                  ;開放外中斷功能
```

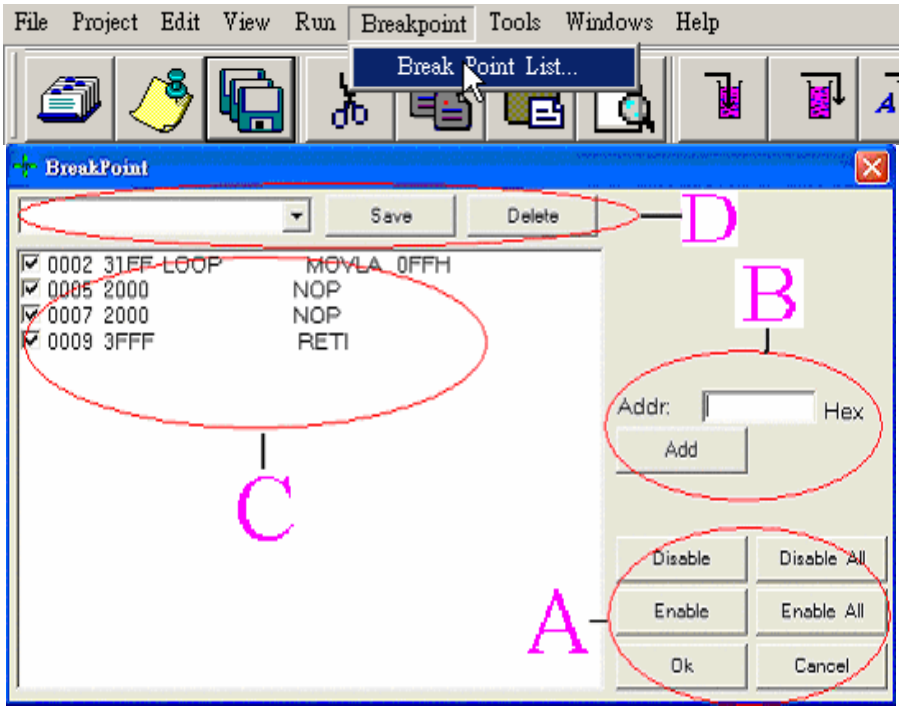
※ 執行 run to cursor 或是 F6，則程式執行到游標所在的位置就會停止，如下：

<紅色箭頭所指的程式將會變成藍色>

```
movla        05h                    ;
select                        ;tmr0採用內部時鐘計數,預分頻器分配給tmr0,
        ;分頻比?1:64
movla   b'00100001'   ;
movam   wakeup        ;設定porta,0口?外中斷輸入口
movla   b'10000100'   ;
movam   irqm                  ;開放外中斷功能
```

【使用 **Break Point List**】

點選 Breakpoint，選擇 Break Point List



A、控制用

B、輸入要設置斷點的位址數值後按 Add

C、所有設置斷點位址資料

D、將設置斷點的位置儲存起來，方便使用者設置多組不同的斷點，儲存的名稱可隨使用者自行定義，會將儲存的資料至於 *.pj 的檔案中。

【主畫面(中斷點發生時)】

A、在來源檔的顯示下

```
nop
nop
bsm     portb,led7      ;打開燈led7
movla   01h
movam   led_flag        ;給led_flag付值01h,使其指向燈led1
ret     ;退出副程式
;
;
;==========================================
;MAIN PROGRAM
;==========================================
main:;初始化
clrm    count1   ;清count1
clrm    del_cnt  ;清del_cnt
clrm    del_flag        ;清del_flag
movla   02h             ;
movam   led_flag        ;給led_flag付值02h,使其指向燈led2
movla   01h     ;
movam   portb           ;打開燈led1
clra                    ;
iodir   portb   ;將portb口設?輸出態
movam   porta           ;porta口輸出?0
movla   0fh             ;
iodir   porta   ;將porta口設?輸入態
movla           05h             ;
select                          ;tmr0採用內部時鐘計數,預分頻器分配給tmr0,
        ;分頻比?1:64
movla   b'00100001'     ;
movam   wakeup          ;設定porta,0口?外中斷輸入口
movla   b'10000100'     ;
movam   irqm                    ;開放外中斷功能
movm    tmr0,w          ;
movam   time_temp       ;讀tmr0的值結果送入time_temp中
;
;
start:;主迴圈
call    readtime        ;調tmr0時間處理副程式
btmsc   irqm,exintm     ;判斷外中斷功能是否關閉
goto    Start                   ;未關閉外中斷功能,返回start處執行
```

※ 紅色為設置的中斷點，灰色為設置失敗的中斷點，藍色為程式現在運行的位置

B、在機器碼的顯示下

```
0078  004E  2000                NOP
0079  004F  0B06                BSM     PORTB,LED7
0080  0050  3101                MOVLA   01H
0081  0051  2088                MOVAM   LED_FLAG
0082  0052  3F7F                RET
0083  0053  2189  MAIN          CLRM    COUNT1
0084  0054  218A                CLRM    DEL_CNT
0085  0055  218F                CLRM    DEL_FLAG
0086  0056  3102                MOVLA   02H
0087  0057  2088                MOVAM   LED_FLAG
0088  0058  3101                MOVLA   01H
0089  0059  2086                MOVAM   PORTB
0090  005A  2100                CLRA
0091  005B  2006                IODIR   PORTB
0092  005C  2085                MOVAM   PORTA
0093  005D  310F                MOVLA   0FH
0094  005E  2005                IODIR   PORTA
0095  005F  3105                MOVLA   05H
0096  0060  2002                SELECT
0097  0061  3121                MOVLA   B'00100001'
0098  0062  20A0                MOVAM   WAKEUP
0099  0063  3184                MOVLA   B'10000100'
0100  0064  20A1                MOVAM   IRQM
0101  0065  2301                MOVM    TMR0,W
0102  0066  208D                MOVAM   TIME_TEMP
0103  0067  3600  START         CALL    READTIME
0104  0068  0521                BTMSC   IRQM,EXINTM
0105  0069  3A67                GOTO    START
0106  006A  0005                BTMSS   PORTA,0
0107  006B  3A67                GOTO    START
0108  006C  080F                BSM     DEL_FLAG,0
0109  006D  3105                MOVLA   05H
0110  006E  2B0A                XORAM   DEL_CNT,W
0111  006F  0103                BTMSS   STATUS,Z
0112  0070  3A67                GOTO    START
0113  0071  218A                CLRM    DEL_CNT
0114  0072  0005                BTMSS   PORTA,0
0115  0073  3A67                GOTO    START
0116  0074  0C0F                BCM     DEL_FLAG,0
0117  0075  0D22                BCM     IRQF,EXINTF
```

※ 紅色為設置的中斷點，綠色為程式現在運行的位置

<註：顏色部份非固定，可由使用者另行定義>

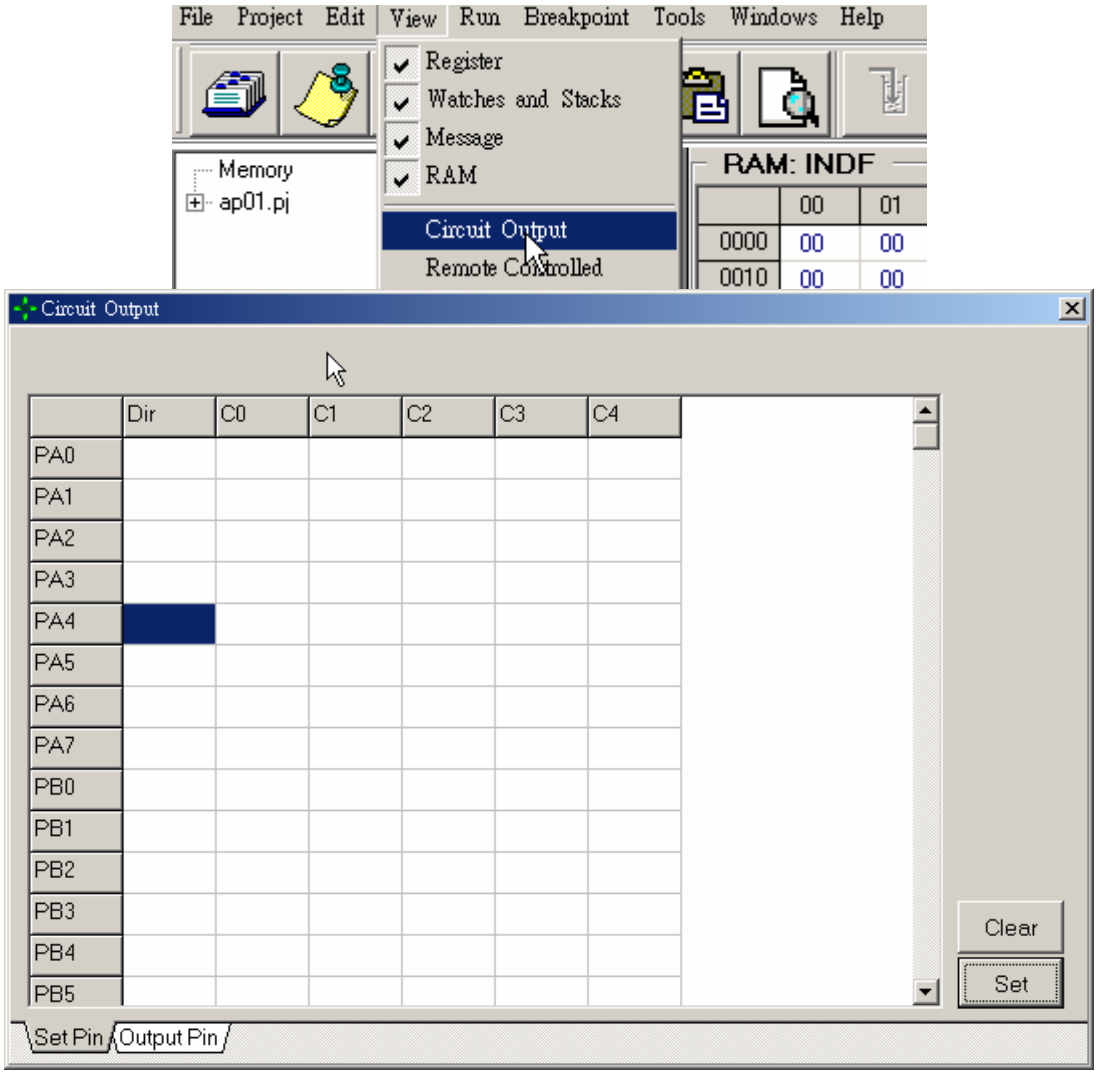
INCLUDE 的使用

使用include的指令，可縮短主程式對程式碼，方便閱讀。

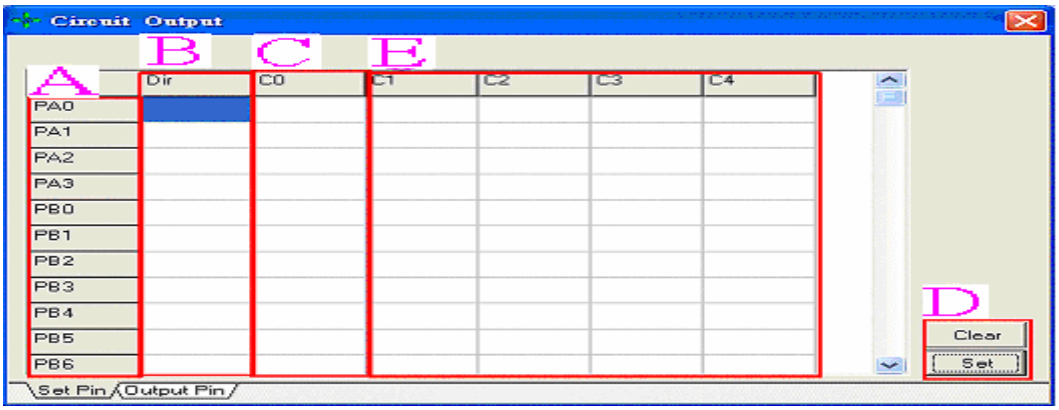
```
;osc1,osc2:clock pin(0)
;resetb0:pa0 be normal i/o pin(0)
;resetb1:resetb1(pa5) be system reset pin(1)
;fosc1~fosc0:nt(01)--8MHz
;lv1~lv0:don't use(11)
;----------------------------
sec_acc         equ     09h
second          equ     0ah
count_buf       equ     0bh
;----------------------------
#include "D:\tenx\1706\Tx58pe10.inc"
        org     3ffh
        movla   0x7f
        org     3feh
        lgoto   tmr1_int
        org     000h
        movam   0x41
        lgoto   main
        org     010h            ;pc---16,page:0
;---------------------------
led_table:
        addam   pc,f
        retla   00h     ;"0"
        retla   01h     ;"1"
        retla   02h     ;"2"
        retla   03h     ;"3"
        retla   04h     ;"4"
        retla   05h     ;"5"
        retla   06h     ;"6"
        retla   07h     ;"7"
        retla   08h     ;"8"
        retla   09h     ;"9"
        retla   0ah     ;"10"
        retla   0bh     ;"11"
        retla   0ch     ;"12"
        retla   0dh     ;"13"
        retla   0eh     ;"14"
        retla   0fh     ;"15"
        retla   10h     ;"16"
```

虛擬電路的使用

此功能只有在軟體模擬時才能夠使用。

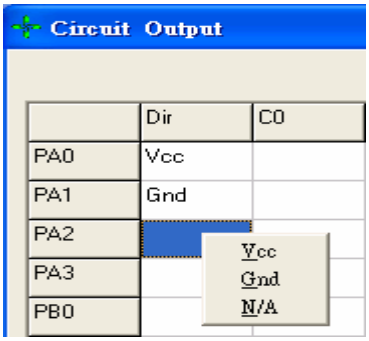


【設定 Pin】



A、IO PIN 腳：此處會根據所選不同型號的 IC 而不同。

B、IO 腳方向：設定模擬 LED 的接法，按滑鼠右鍵選擇，“N/A”選項表示不接，Vcc 與 Gnd 接法請參考圖一與圖二：



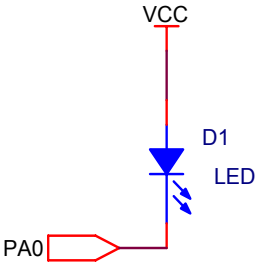


圖1

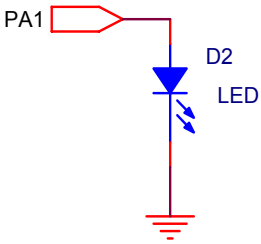
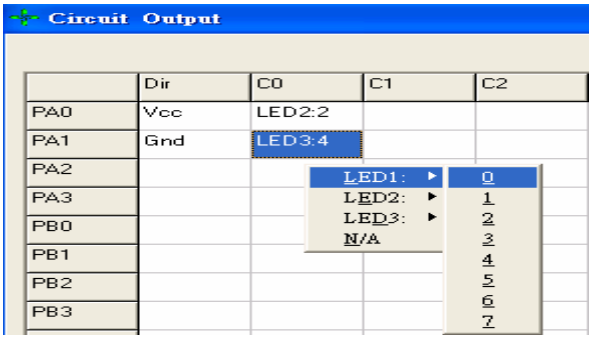


圖2

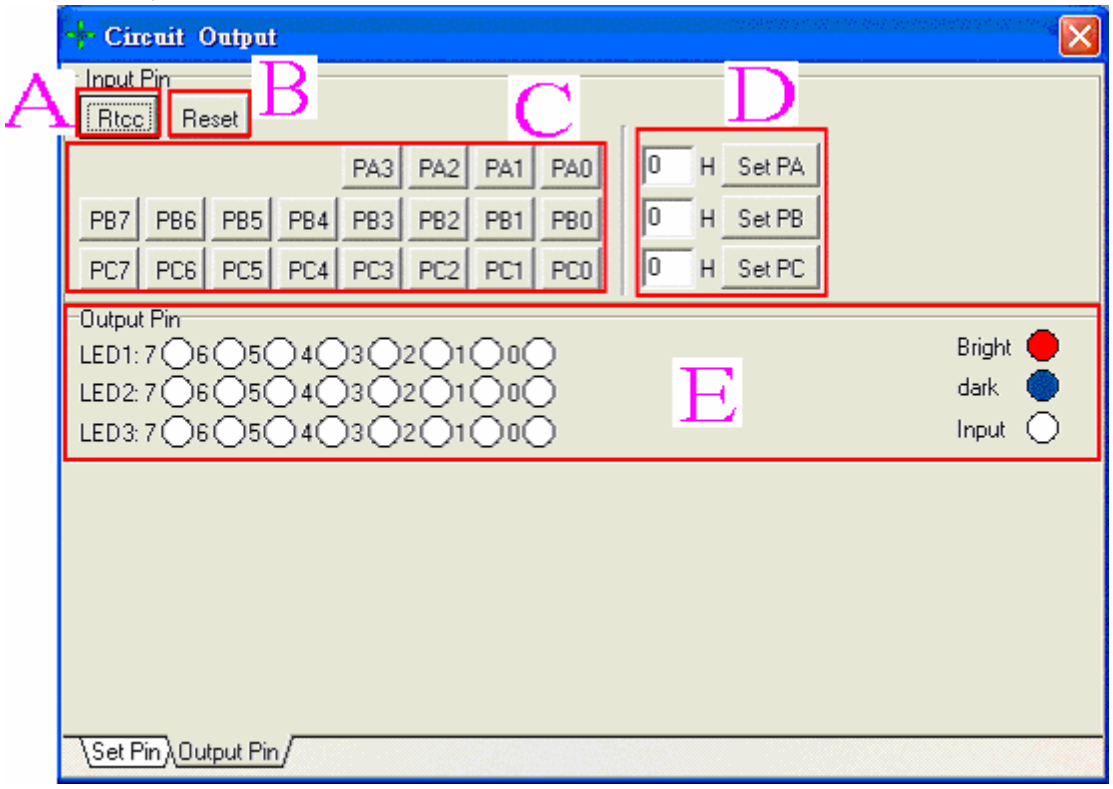
C、C0：在要設定的位置按滑鼠右鍵設定 LED 的位置，共有三組 LED 可以選擇，每組 LED 有 8 個可選，如下圖所示，LED2:2 表示是第二組第二個 LED；LED3:4 表示是第三組第四個 LED。



D、當設定完成後，按 Set 鍵儲存，如果設定完成後未按此鍵，則設定完成也不會有作用，要清除只需按 Clear 鍵就會將所有設定清除。

E、此四格暫時保留未用。

【觀察 PIN 輸出的狀態】



A、Rtcc：外部 clock 模擬信號，按此鍵一下 TMR0 加 1。

B、Reset 鍵，按下此鍵，程式會 Reset 回初始狀態。

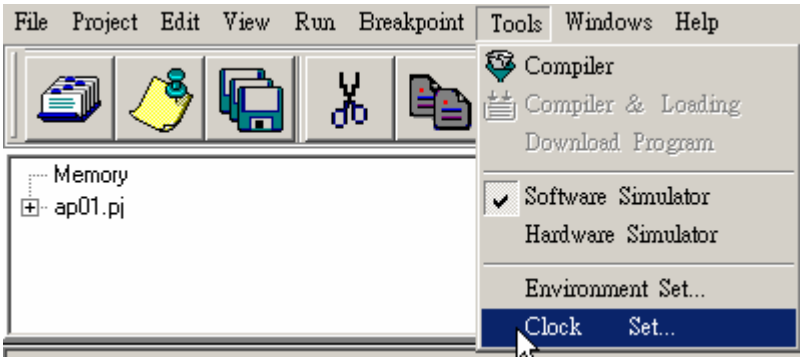
C、此框中的各鍵，是用於當程式有應用到 Wake-Up 或 Interrupt 時，當按下程式中所設定 Wake-Up 或 Interrupt 鍵時，就會產生相對應的動作，按鍵的有效與否是根據不同 IC 型號，而不是每個按鍵皆有效的，詳細資料請參考各 IC SPEC。

D、用於設定 I/O port “In/Out 數值”動作，I/O 的 In/Out 狀態由程式設定，在白框中輸入 16 進制值後按下 Set，將所輸入的數值由對應的 I/O port 輸出/輸入。

E、模擬 LED Output 動作，紅色表示點亮，藍色表示熄滅。

【頻率設定】

此功能只有在軟體模擬時才能夠使用。



※ 點選頻率設定會出現下圖。

Frequency：此選項為設定模擬時的 Clock 頻率。

RTCC：為模擬外加 TRM Clock 頻率，當沒打勾表示為未使用。

INRC Clock：當選擇的 IC 有使用到 Internal RC 時，會多一視窗設定 Internal RC 值，若選擇的 IC 無 Internal RC 的功能時，則不會出現。

注意事項

【說明1】

☐ External Clock 當選擇internal / external 時脈時，如選擇內部振盪時，不需將其位置“ ∨ “ 。

☐ External Reset 當選擇internal / external 時脈時，如選擇外部振盪時，需將其位置“ ∨ “ 。

【說明2】

在編譯除錯時，將會自動儲存＊.ASM檔案，所以＊.ASM檔案不可設定成”唯讀”檔，如果設定爲”唯讀”檔，將會在編譯除錯時，發生錯誤。

【說明3】

目前 TICE58 可以支援三種 POD。

1. TM58POD：支援 TM58P11、TM58PC05、TM58P10/PC10 和 TM58P20/PC20。

2. TM58POD V1.0：支援 TM58RC10、TM58PE10 和 TM58PR10。

3. TM58POD V3.0：支援 TM58PC10A 和 TM58PC20A。