



十速科技股份有限公司
tenx technology inc.

TICE58

User Manual

www.tenx.com.tw

Contents

Product Introduction	2
1. Introduction	2
2. System Configuration	3
3. Software Installation	4
Quick Start	8
Menu Functions	14
1. Main Screen-Functions in Menu	14
2. Main Screen-Tool Bar	16
3. Main Screen Window	17
4. Environment Set Window	23
5. Help Window	26
Tips on using the software	27
1. Use of BreakPoint	27
2. Use of INCLUDE	32
3. Use of Virtual Circuit	33
Points to note	37

Product Introduction

1. Introduction

To minimise the development time and process, tenx technology exclusively designed the IC emulator-TICE58. TICE58 was developed by tenx technology; it supports all IC models in the TM58 series.



【TICE58 exterior】

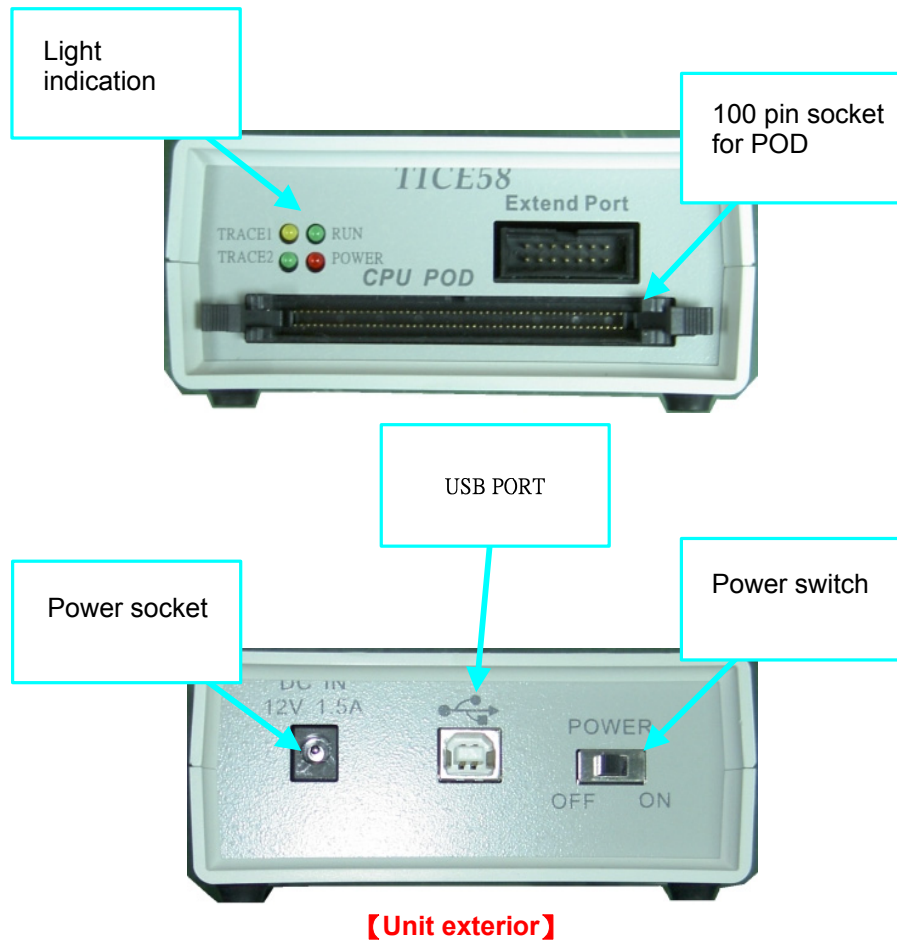
※System Requirements:

1. IBM PC 486 compatible or above.
2. USB Port.
3. 64M memory space or above.
4. WINDOWS 2000/XP operating system.

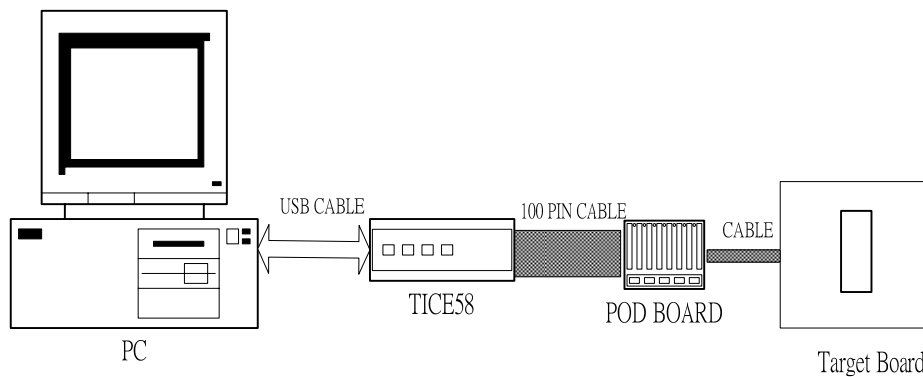
※Features of TICE58:

1. USB port connection to computers, high-speed emulation.
2. With internal or external oscillator.
3. Ability to set up BreakPoint.
4. Internal and external RESET.
5. Hardware and software emulations.
6. Compiling, debugging and assembly can all be done with the software.
7. In addition to AUTO (GO) and (STEP), also provides STEP INTO function.

2. System Configuration

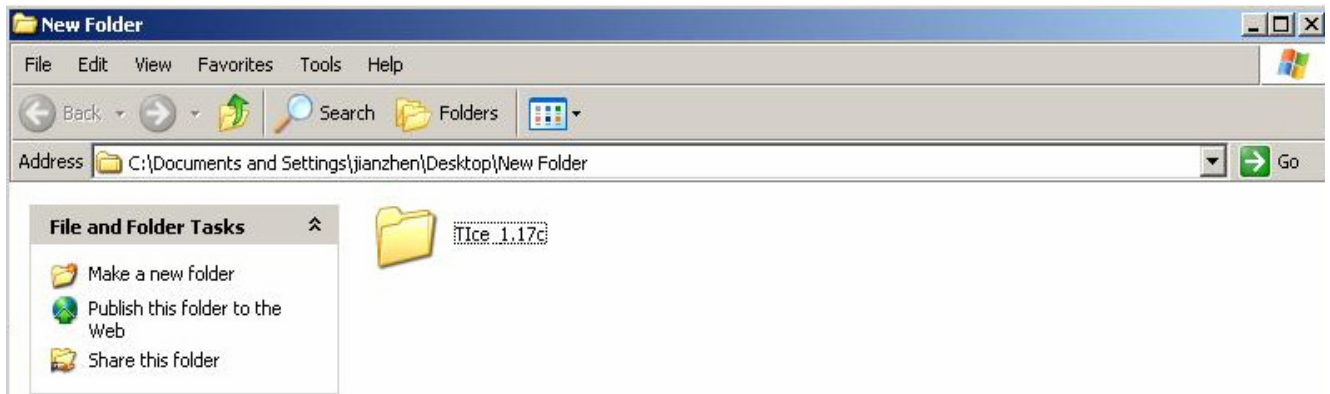


- ※ Connect the 100 pin cable to TIC 58 unit and corresponding POD board.
- ※ Connect the USB cable from PC to TICE58 unit.
- ※ Connect power cord.
- ※ Turn power on and finish hardware installation.

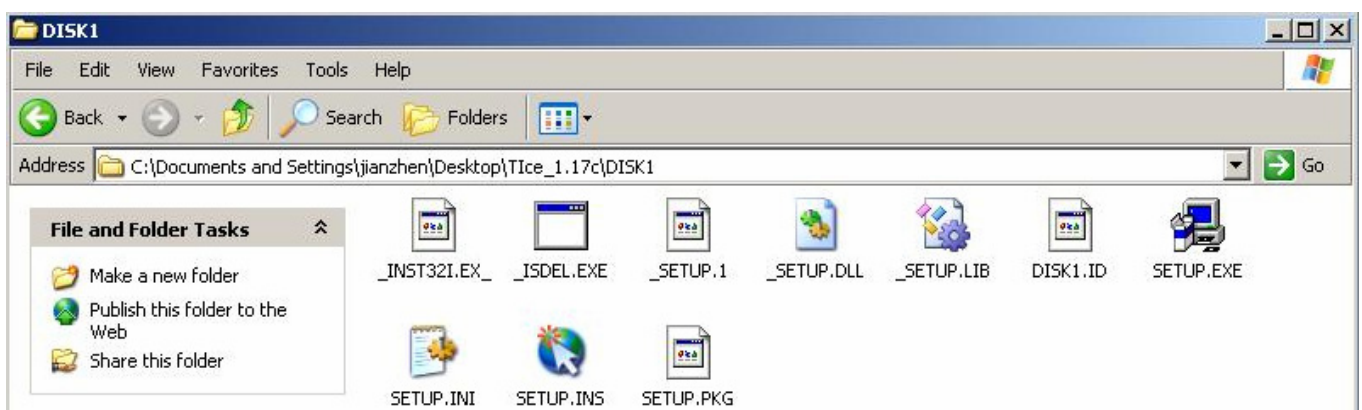


3. Software Installation

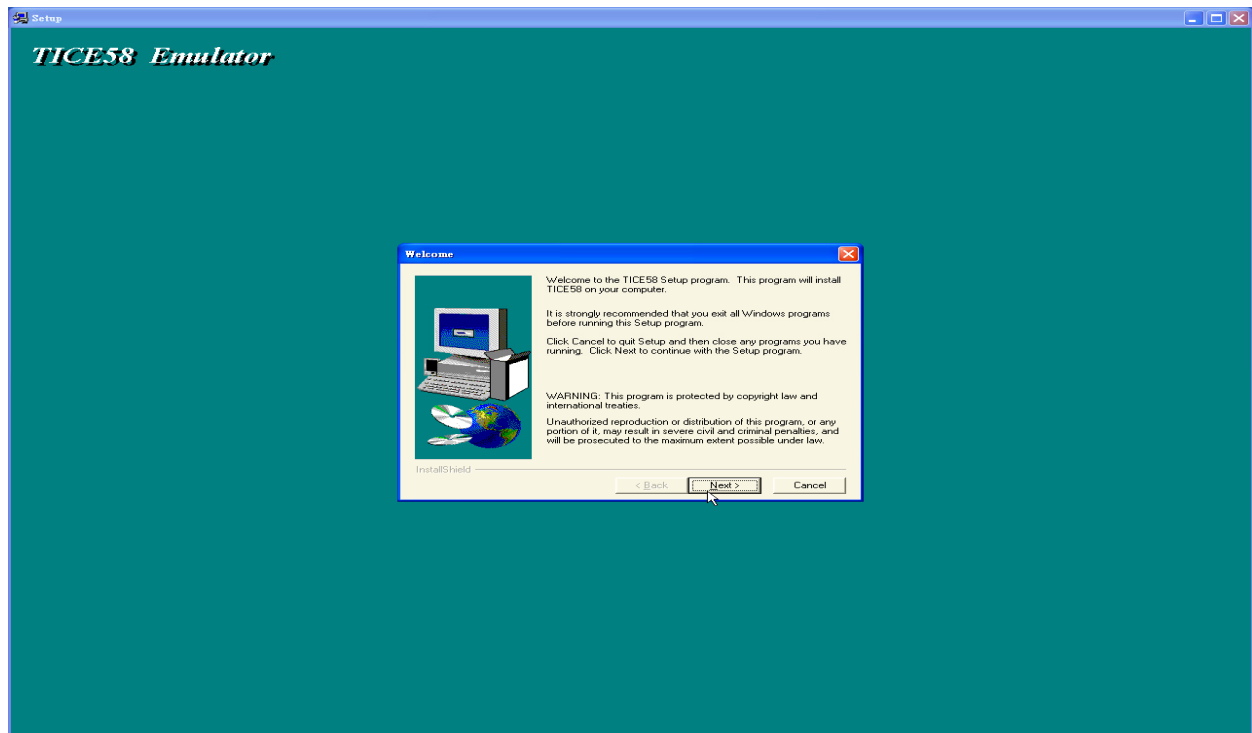
- 【Step 1】** (1). Double click on TICE58 software and unzip the file. When it's done,
a TICE58 index will be generated:
(2). Double click on TICE58 folder:



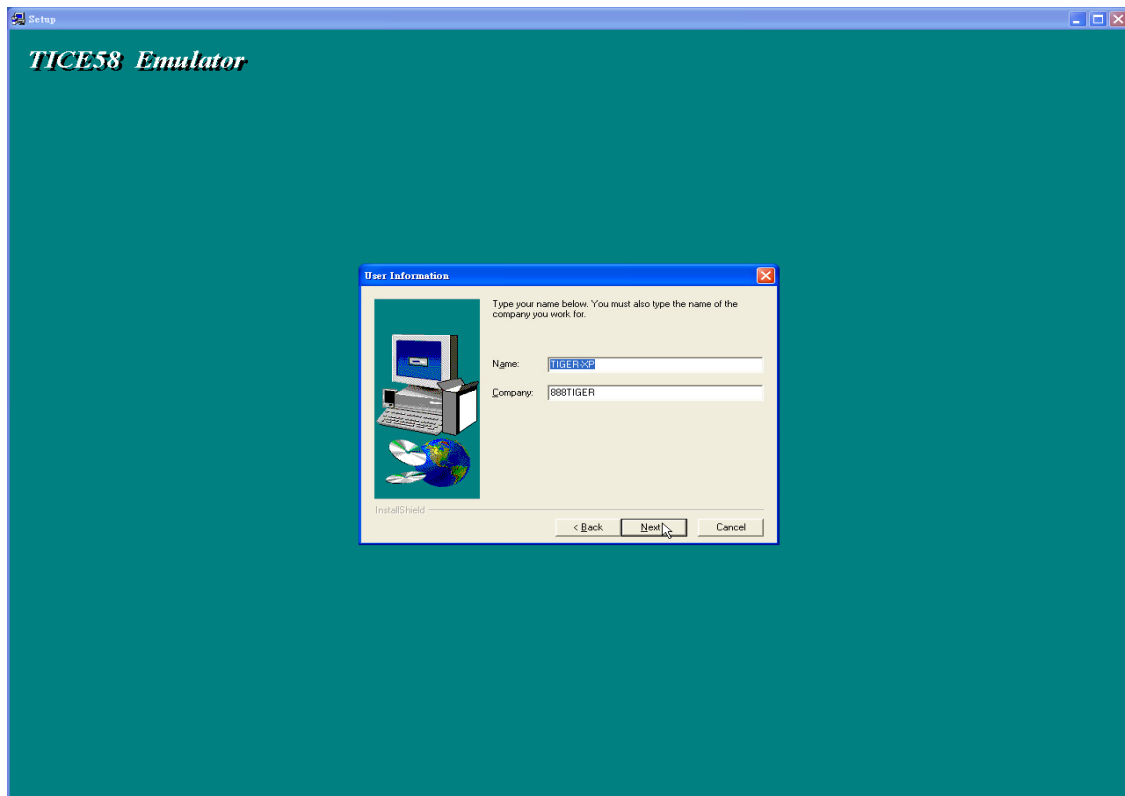
- 【Step 2】** Click  **SETUP**
Setup Launcher (SETUP.EXE)
InstallShield Corporation, Inc. to enter the following software installation
screen:



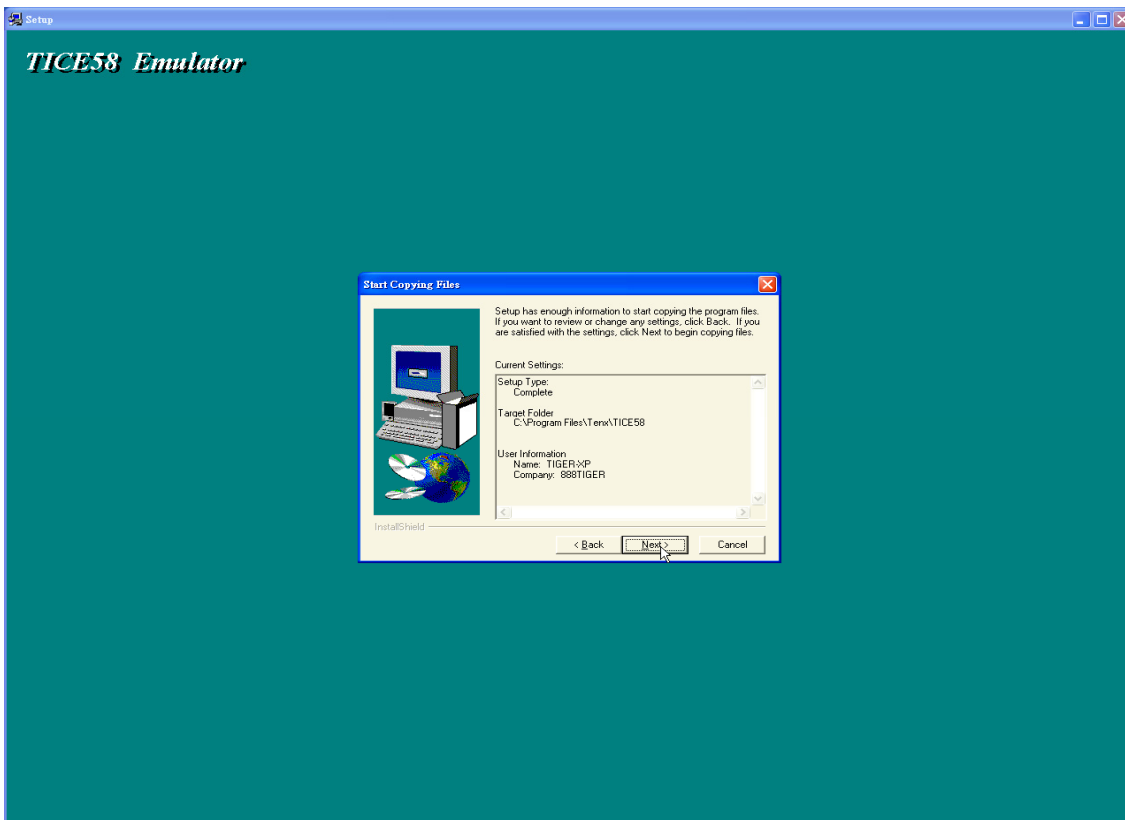
【Step 3】 Click **NEXT** to start installation:



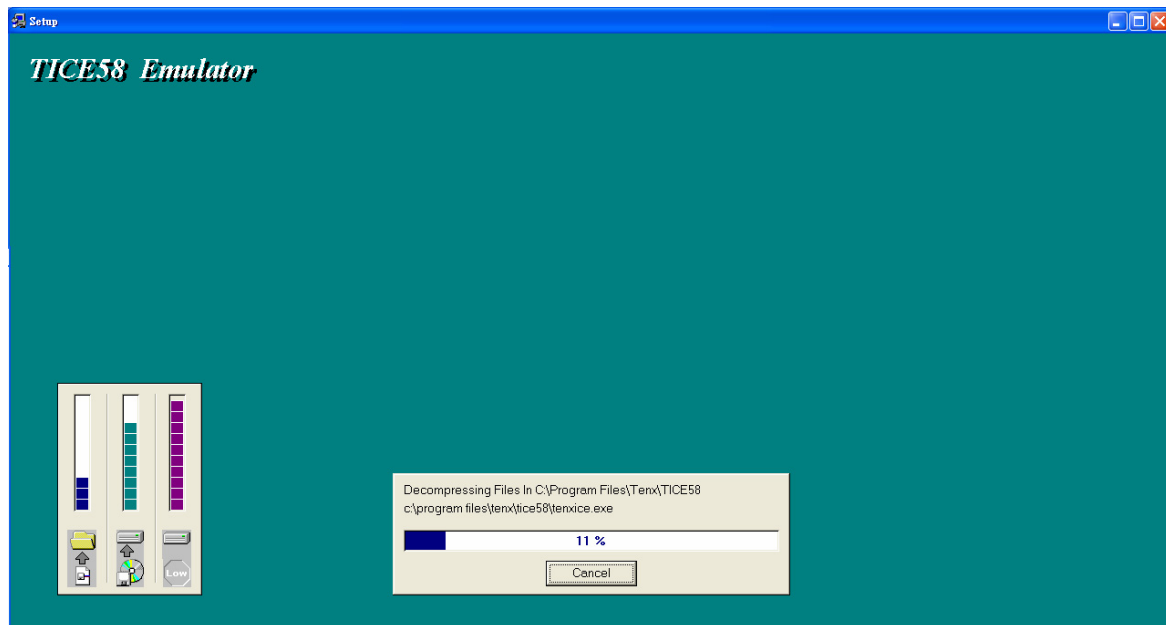
【Step 4】 Click **NEXT** to continue:



【Step 5】 Click **NEXT** to start installation:

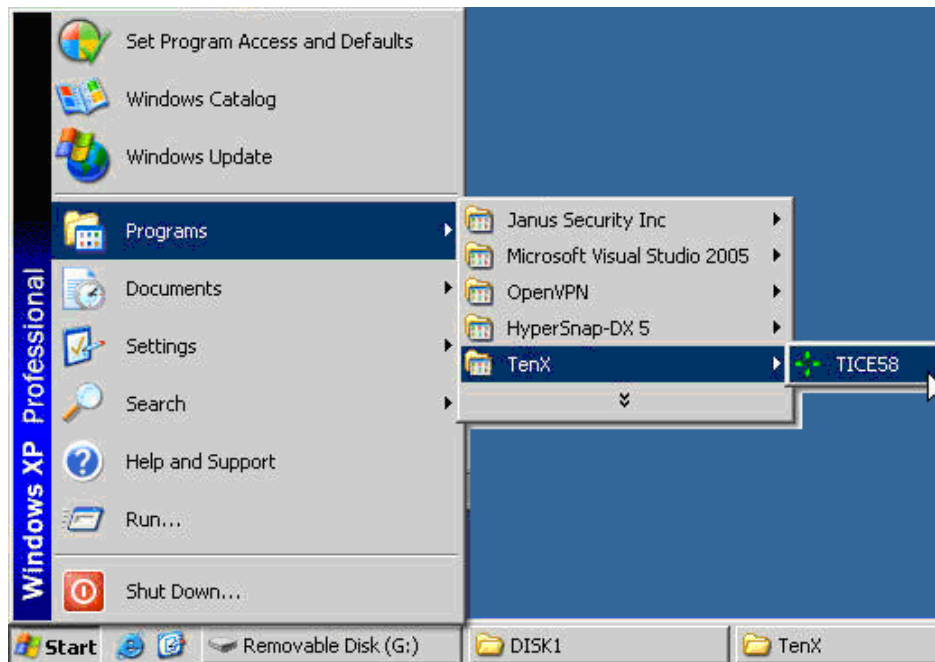


【Step 6】 Software installation screen (when the figure in red column reaches 100%, the installation is complete):

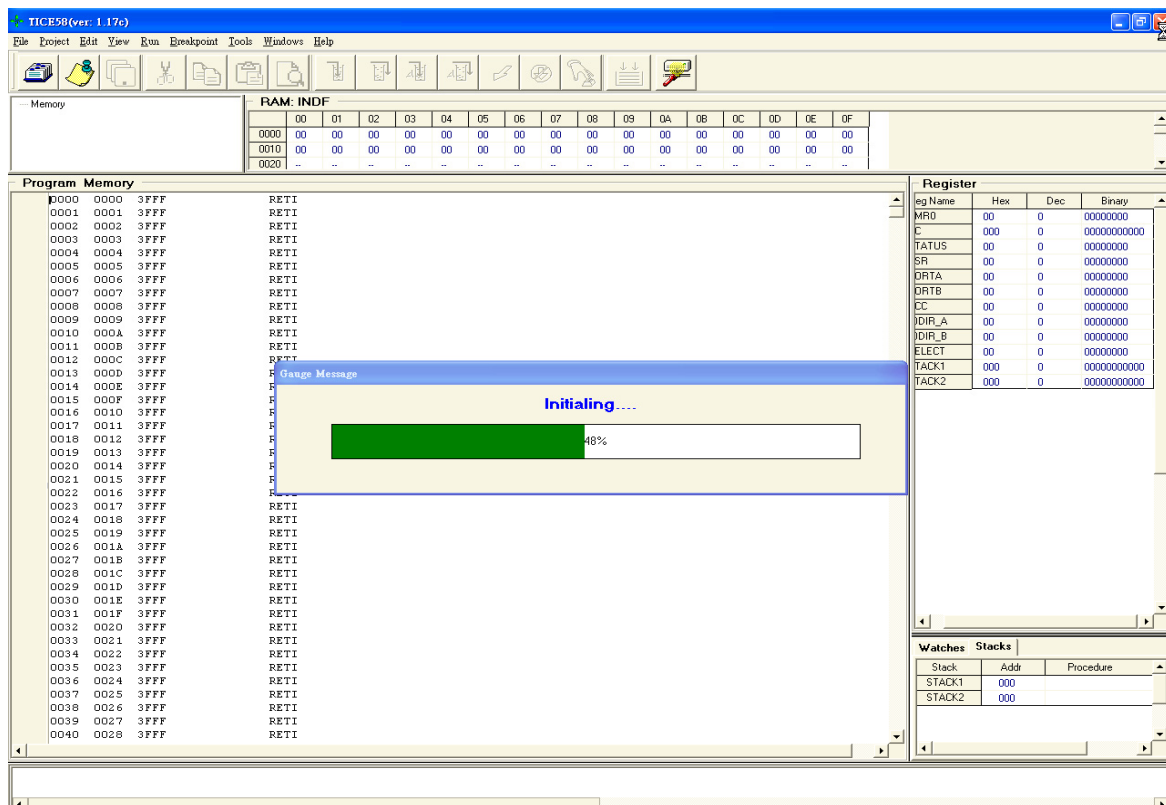


Quick Start

【Step 1】Run TICE58 program

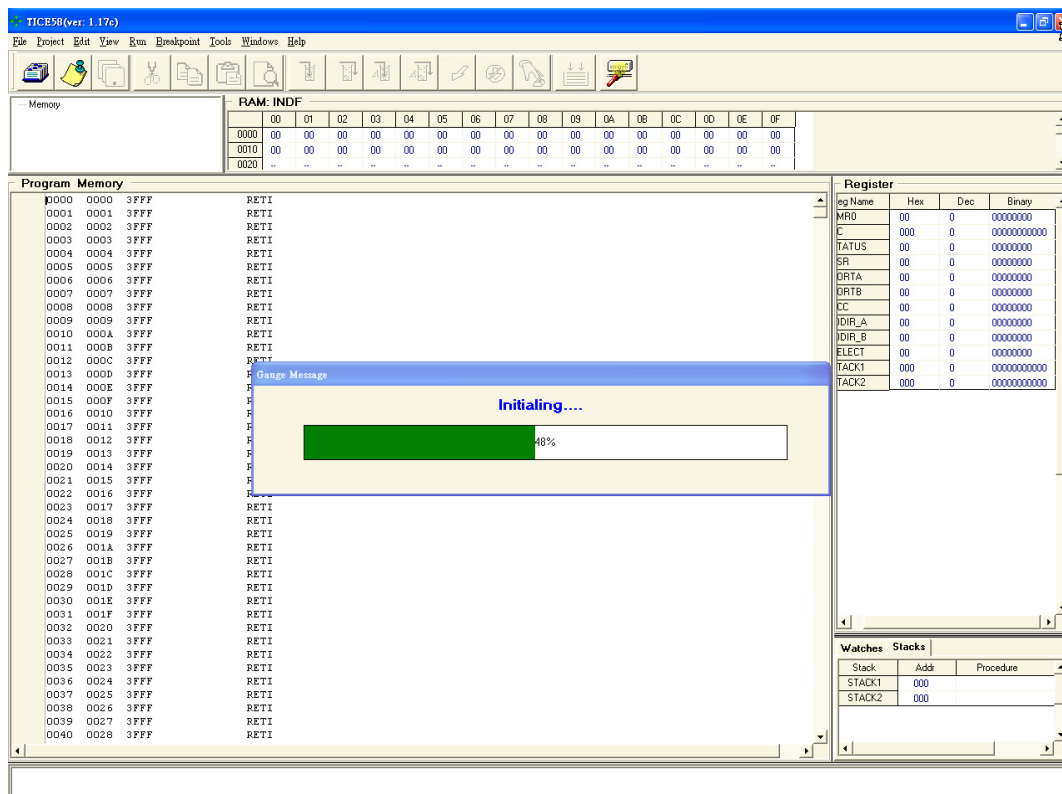


【Step 2】When the TICE58 software is running, it starts downloading from USB device and TICE58 unit.

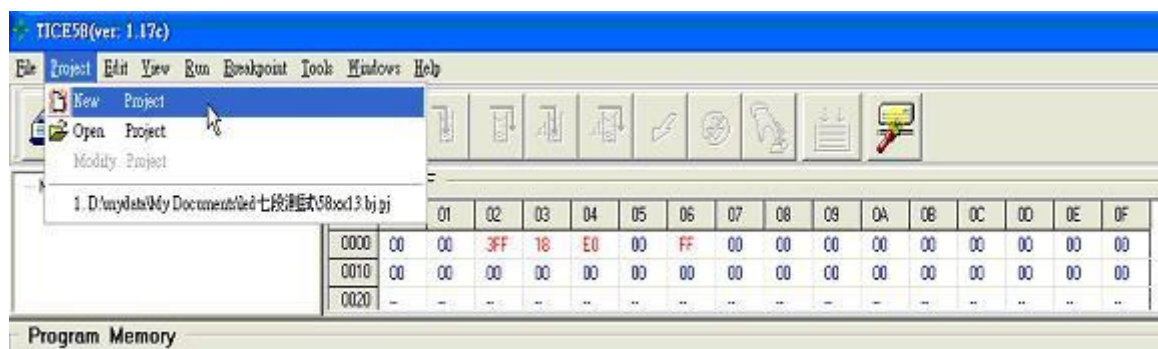


✗ When Initialing indicates 100%, the download is complete, and the program is ready for use.

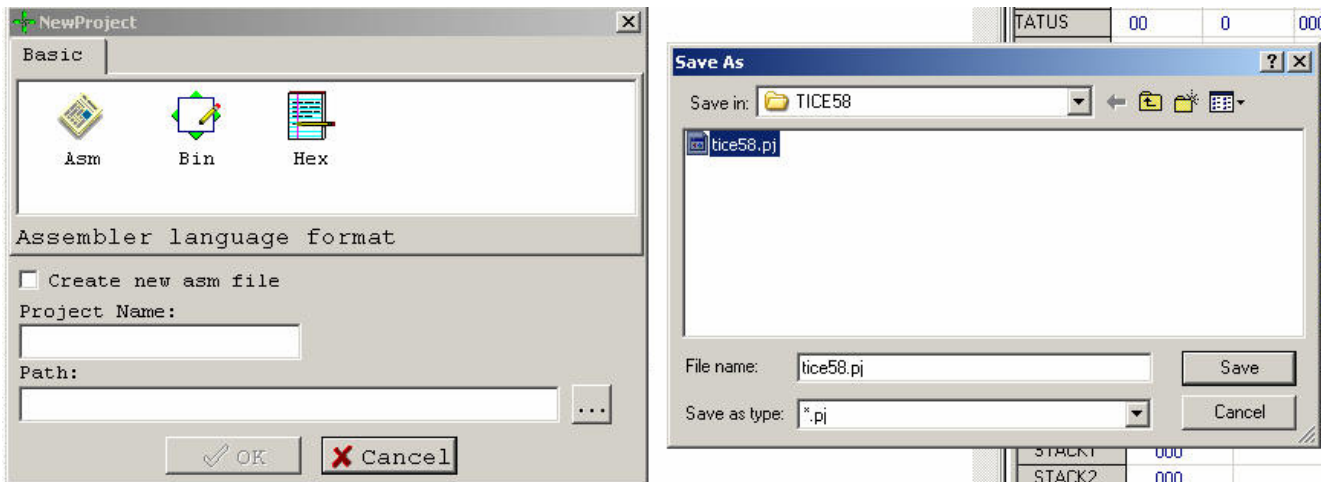
✗ If the downloading fails, press the  key on the far right to enable TICE58 software to download again.



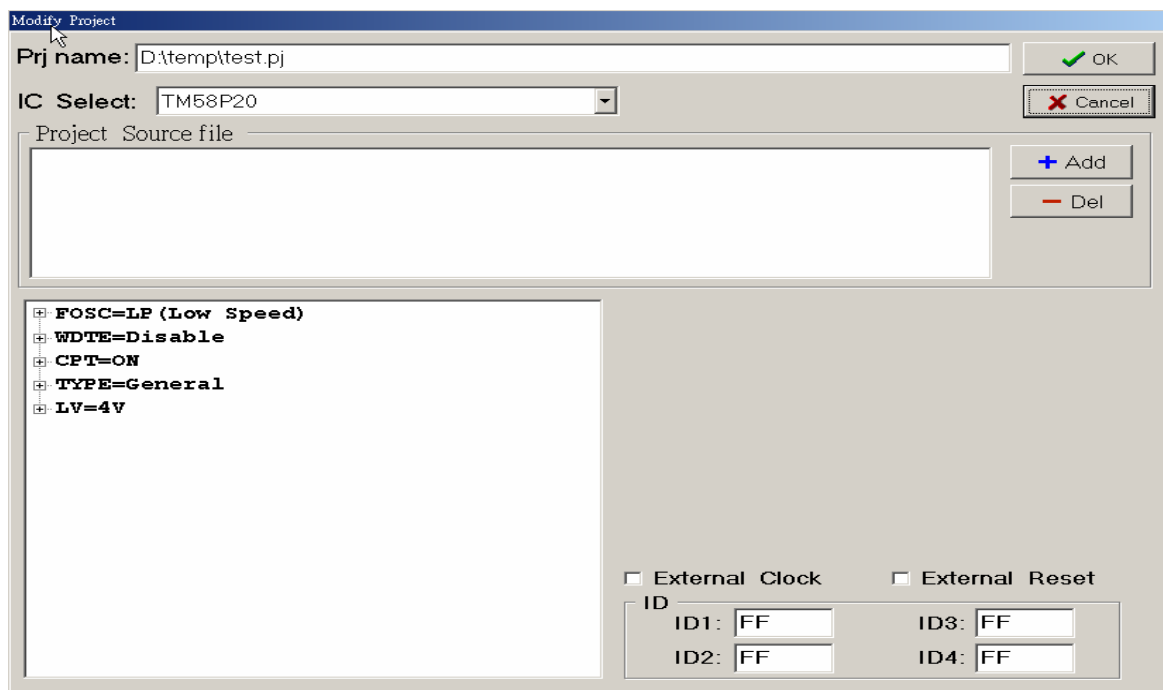
【Step 3】 Choose project → open project



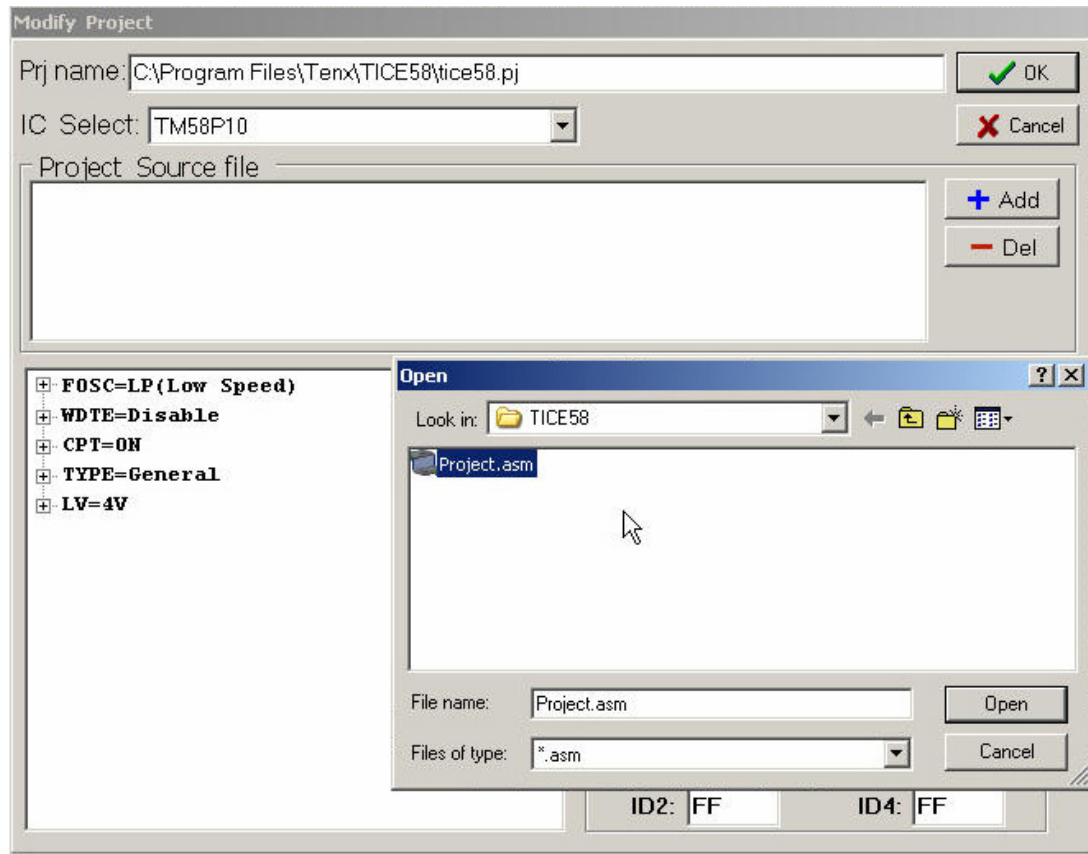
【Step 4】 Choose path, and enter project name (for example: test.pj), then click save button as shown below:



【Step 5】 After clicking OK button, a screen with 'IC Select', 'Configuration' and 'Add Source Code' will appear (See below). (IC model needs to match the corresponding POD board, please refer to related POD board)



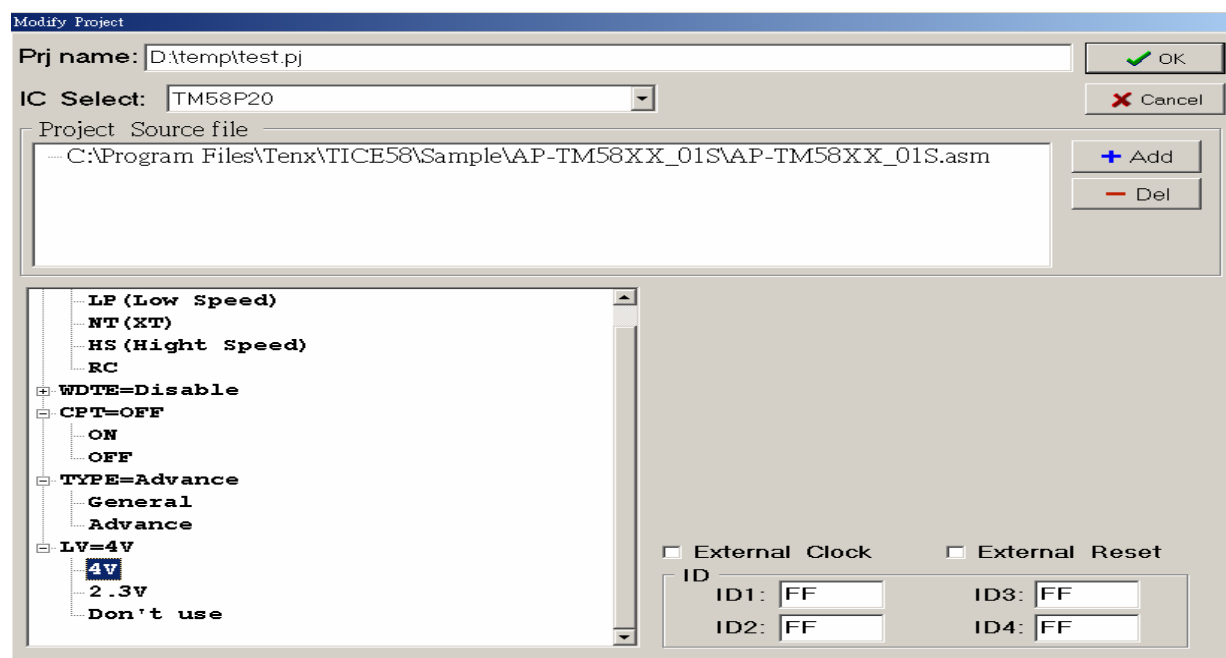
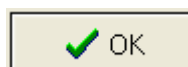
【Step 6】 Firstly, select IC model, then click **+ Add** to add Source Code (Here we add TICE58 sample file, the path is C:\Program Files\Tenx\TICE58\example\AP-TM58XX_01S, <C: may be different due to various configurations>) Select AP-TM58XX_01S.asm, and then press Open.



【Step 7】 After 'Add Source Code' is finished, the screen comes back to configuration, see below. For the sample program, our settings are:

Type: Advanced
FOSC: HS
WATCHDOG: DISABLE

After finishing with the settings, press



✘ External Clock:

To switch between internal/external clock, un-tick box for internal clock (the fixed internal frequency is 4MHz). When using external clock, the figures of Y1, R1, C1 on TM58POD board need to be adjusted, see table below:

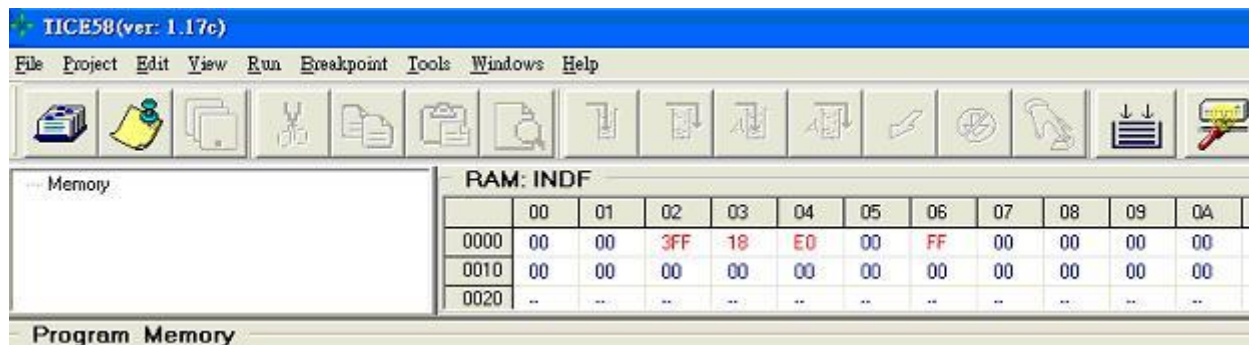
Y1	R1	C1	C2
32.768K	4.7K	4.7n	
455K	4.7K	680p	
1M	27K	20p	
2M	20K~27K	20p	
4M	20K~27K	20p	
8M	1K	20p	
10M	1K~470	20p	
12M	470	20p	
14.374M	100	20p	
16M	3.3	20p	

✘ External Reset:

To use the internal/external RESET, un-tick box for internal RESET

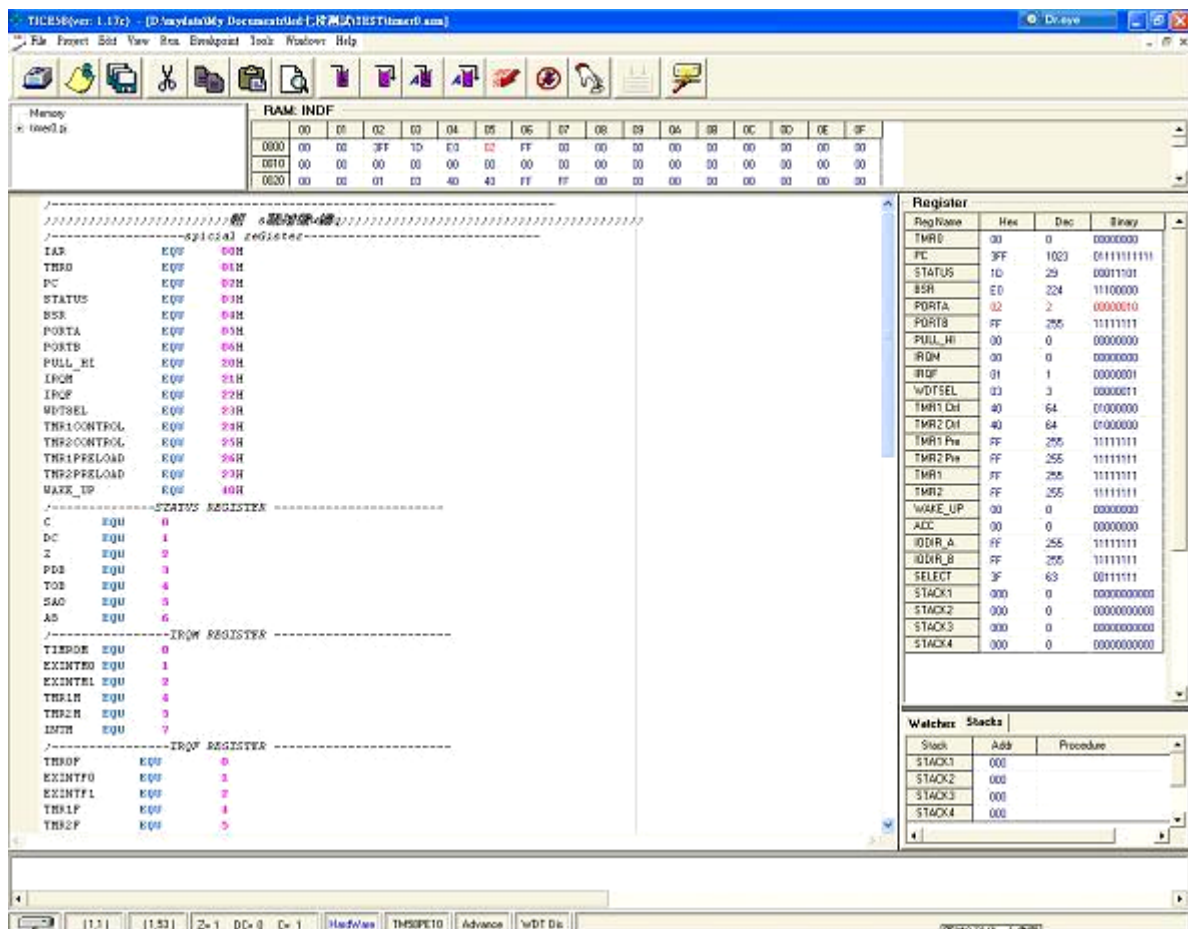
✘ ID: There are 4 selections, users can self define ID, default is FF

【Step 8】 Press  to compile, debug and download files to TICE58.



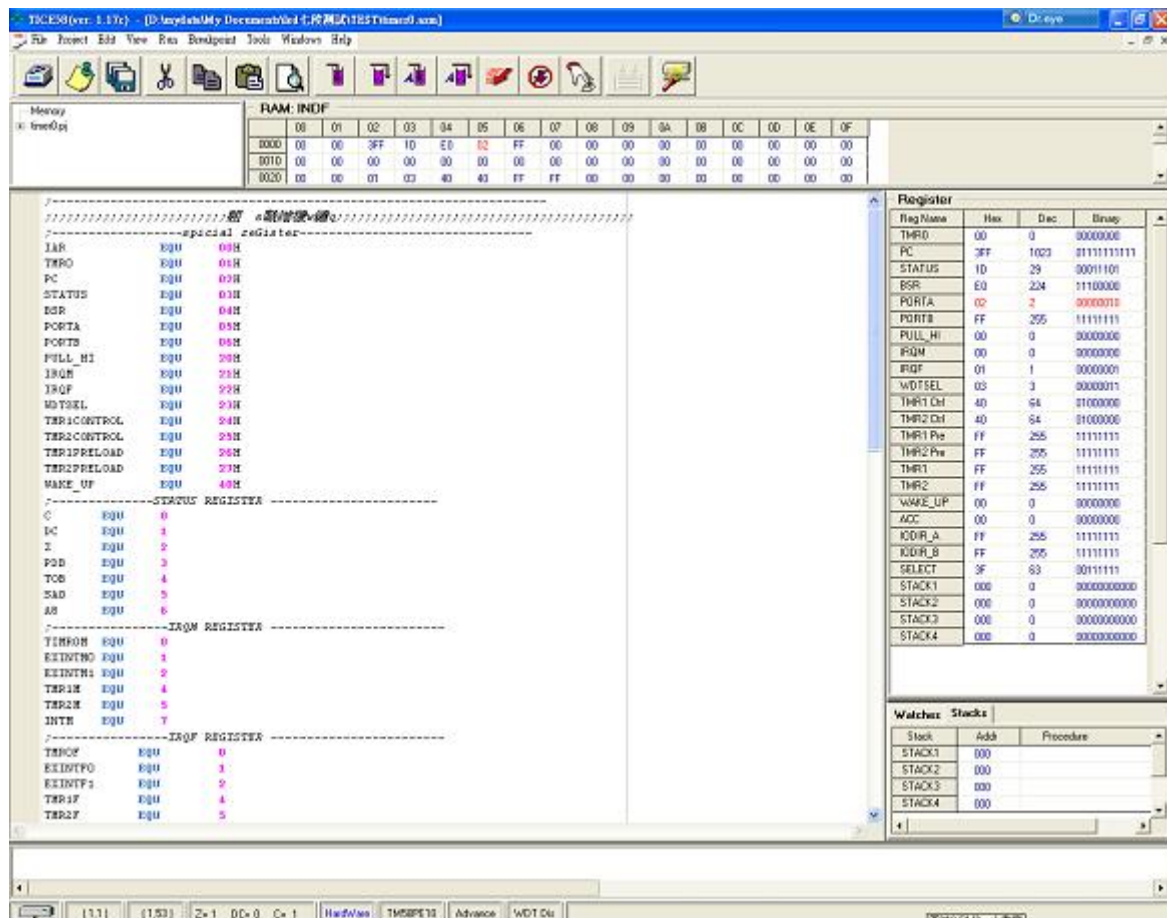
✘ If no errors are found in compiler and program is fully downloaded, press  to free run the program, or press  for single step function.

✘ If errors are found during downloading, press  to restart downloading.
Below is the screen after downloading is completed.



Menu Functions

1. Main Screen-Functions in Menu



File:

- **Create new file:** Creates new ASM files.
- **Open file:** Opens existing files.
- **Save file:** Saves the current file.
- **Save as file:** Saves the current file with another file name.
- **Save all file:** Saves all files in a project.
- **Close all file:** Closes all files in a project.
- **Printer:** Printer settings.
- **Printer set:** Configures printer settings.
- **Exit:** Closes TICE58 program.

Project:

- **New project:** Creates new projects.
- **Open project:** Opens existing projects.
- **Modify project:** Modifies settings in projects, including configurations, IC models...

Edit:

- Copy
- Cut
- Paste
- Find
- Replace

View:

- Register**: Window that displays the Register.
- Watches and Stacks**: Window that displays Watches and Stacks.
- Message**: Window that displays the message.
- RAM**: Window that displays RAM.
- Circuit Output** (only applicable in software simulator): Please refer to 'Tips on using the software' 4.3
- Remote Controlled** (only applicable in software simulator)

RUN:

- Free run**: Runs the program in full.
- Auto step into**: Runs (all programs) with single steps automatically.
- Auto step over**: Runs (the program only) with single steps automatically.
- Step over**: Runs (all programs) with single steps.
- Step into**: Runs (the program only) with single steps.
- Run to cursor**: Runs the program only to where the cursor is.
- Stop**: Stops running the program
- Warm reset**: Restarts the program

BreakPoint:

- Break Point List**: List of BreakPoint.

Tools:

- Compiler**: Compiles files
- Compiler & Loading** (only applicable in hardware simulator) : After compiling, programs are downloaded to EV.
- Download Program** (only applicable in hardware simulator) : Downloads programs to EV.
- Software Simulator**: When software simulator is in use, no POD board required. (some functions will be disabled)
- Hardware Simulator**: When hardware simulator is in use, POD board required.
- Environment Set**: Runs environment settings

2. Main Screen-Tool Bar

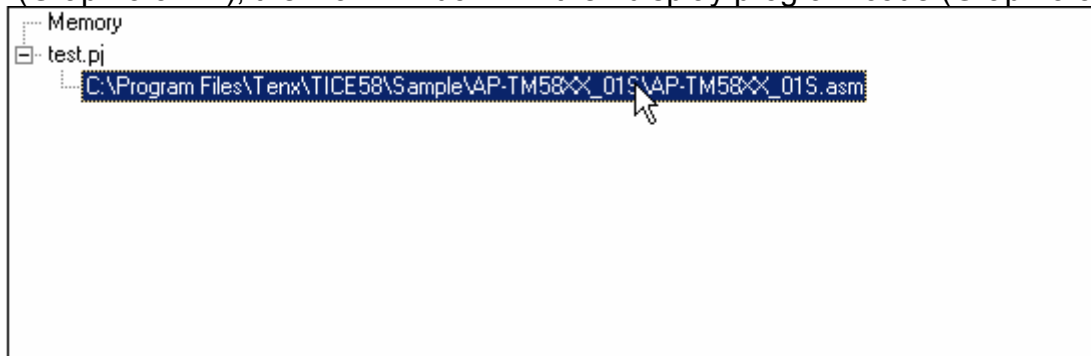
Button Icons	Function Keys	Remarks	
		Open existing files	Opens last or previous files
		Build new source files	Builds new source files (such as .asm)
		Save all files in a project	Saves all files in a project
		Cut	Cuts selection
		Paste	Pastes selection
		Search	Searches the program
	F3	Single step run (all programs)	Runs (all programs) with single steps
	F4	Single step run (only runs the main program)	Runs (the main program only) with single steps, excludes CALLcommand
	F10	Automatic single step run (all programs)	Runs (all programs) with single steps automatically
	F11	Automatic single step run (only runs the main program)	Runs (the program only) with single steps automatically, excludes CALLcommand
	F9	Run program	Free runs the program
	F3	Stop	Stops running the program
	F4	Reset	Program returns to initial starting point
		Run Compiler and Download	Runs Compiler, and downloads files to ICE under no error conditions.
		Initial ICE again	Initializes ICE again.

3. Main Screen Window

【Project File】

This window is devised to display source files or Memory format. Users can configure their preferred display information.

- (1). To display program code, (*.asm) please double click on asm file, as shown below (Graph 3.3.1-1), the main window will then display program code (Graph 3.3.1-2).



(Graph 3.3.1-1)

```

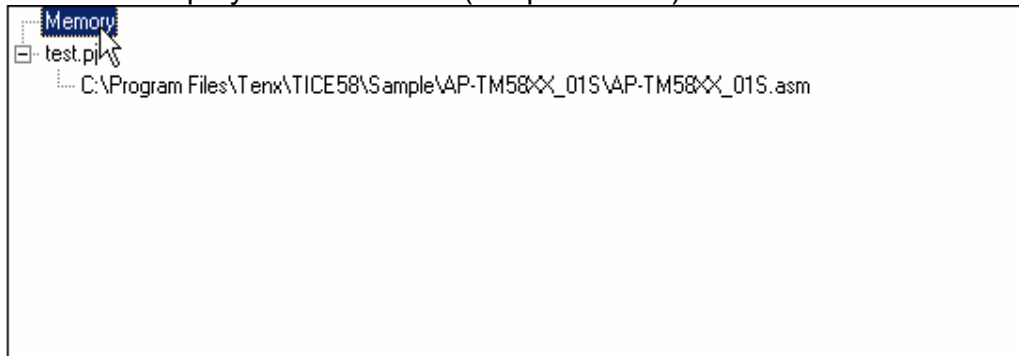
;=====
;=====
org      7ffh
* goto
org      7feh
* goto   extinct
org      000h
;
;
;=====
;tmr0
;=====
readtime:
* movm    tmr0,w
* xoram   time_temp,w
* movam   time_buf
* xoram   time_temp,f
* btms    time_buf,7
* ret
* btmsc   del_flag,0
* incm    del_cnt
* incm    count1

* movla   3dh
* xoram   count1,w
* btms    status,z
* ret
* clrm    count1
* btmsc   led_flag,led1
* goto    turn_on_led1
* btmsc   led_flag,led2

```

(Graph 3.3.1-2)

- (2). To display machine code
Please double click Memory, as shown below (Graph 3.3.1-3), the main window will then display machine code (Graph 3.3.1-4).



(Graph 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		MOVW	TMRO, 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

(Graph 3.3.1-4)

【RAM】

Displays IC RAM information, please refer to IC Data Sheet for corresponding positions.

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

✘ In Debug, if modification is required, double click on required field (see below), enter value, and then press **OK**.

Address: 12 H

Hex

Ok Cancel

The above graph shows the field to be modified.

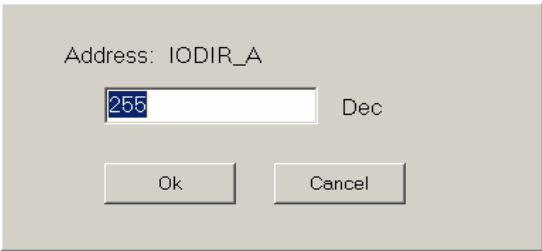
【Register】

Displays IC Register information; please refer to IC Data Sheet for definition of each Register.

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

✘ In Debug, if modification is required, double click on required field (see below), enter value, and then press **OK**.

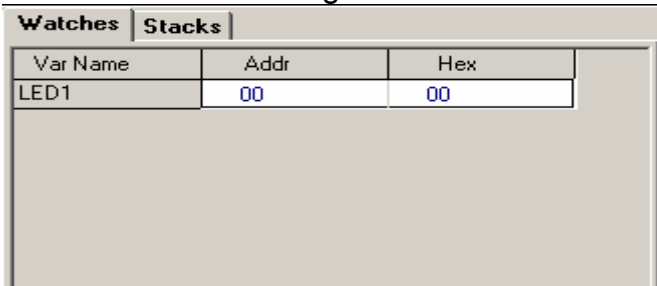


The above graph shows the Register to be modified.

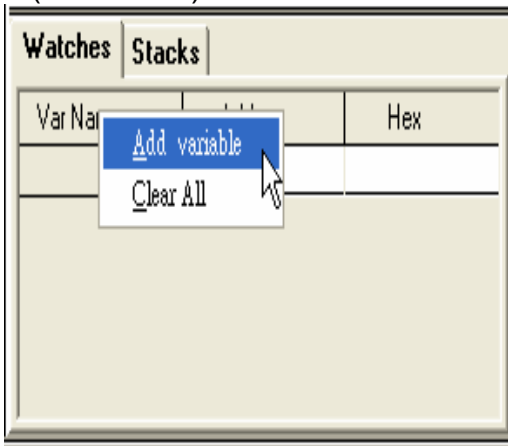
【Watches/Stacks】

This window is divided into Watches and Stacks.

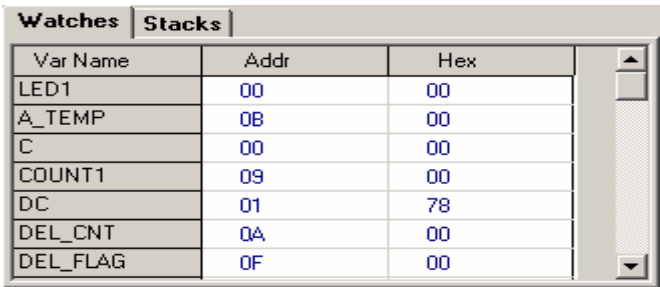
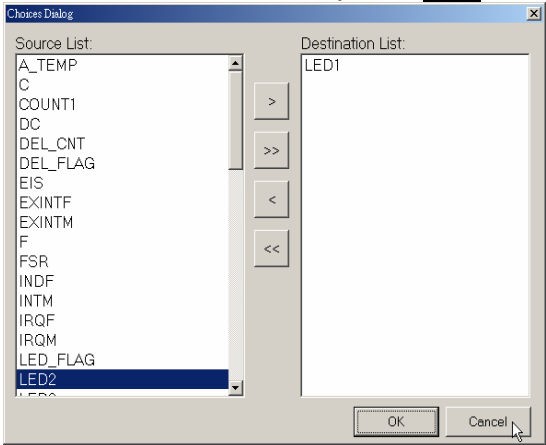
- a. 'Watches' allows users to trace the changes of defined variables in the program.



- ✖ To add variables:
1. Position cursor on Var Name, and right click mouse button (See below).



- 2. Click Add variable, the window below will appear, add the variable to be observed, then press **OK**.



b. Stacks: allows users to observe the values in stacks.

Watches Stacks		
Stack	Addr	Procedure
STACK1	000	
STACK2	000	
STACK3	000	
STACK4	000	

【Operation Messages】

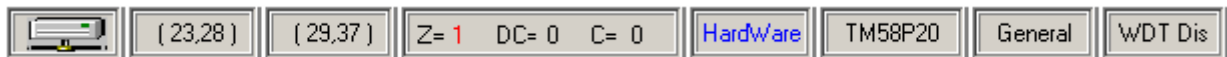
Displays messages during operation, editing...etc.

```
Building C:\Program Files\Tenx\TICE58\Sample\AP-TM58XX_01S\AP-TM58XX_01S.Hex , AP-TM58XX_01S.Bin
Build completed successfully.
```

【Other Messages】

Shown as following:

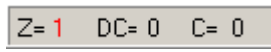
a. Hardware Simulator



 → Shows connection with TICE58 unit.

 → Position of the cursor, when main screen displays under machine code.

 → Position of the cursor, when main screen displays under program code.

 → Value of each flag register.

(Please refer to IC Data Sheet for definition of each flag register)

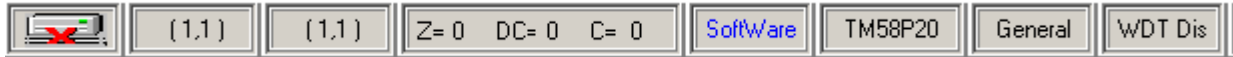
 → Currently in hardware simulator.

 → The IC model to be simulated.

 → The values defined by TYPE in TM58P20 Configurations.

(Each IC provides different configurations; please refer to IC Data Sheet.)

 → WatchDog function is disabled.

b. Software Simulator

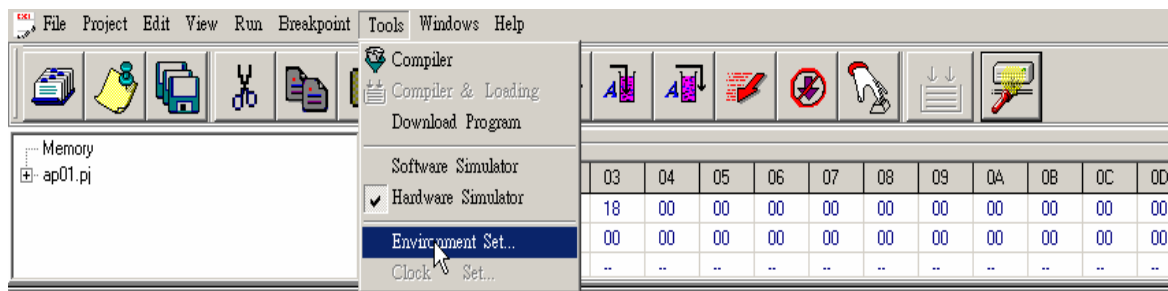
→ Unable to connect to TICE58 unit.



→ Currently in software simulator.

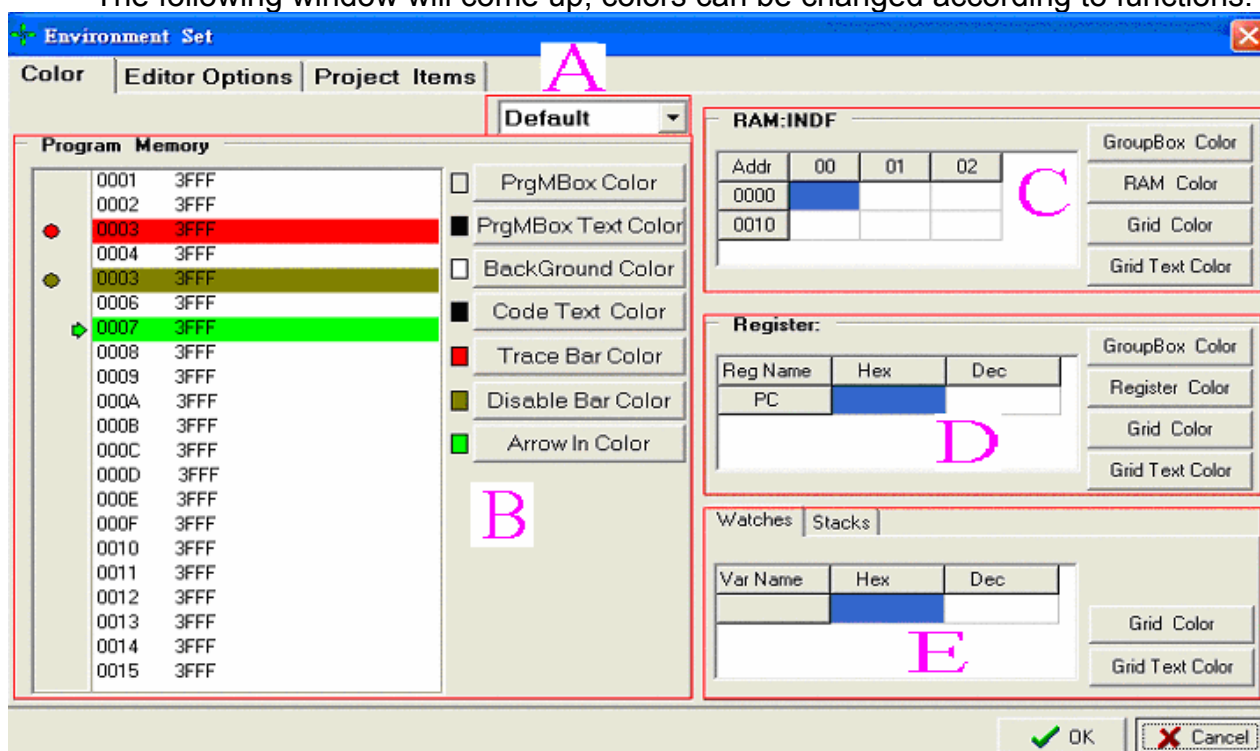
4. Environment Set Window

✖ Under Tools→ Environment set



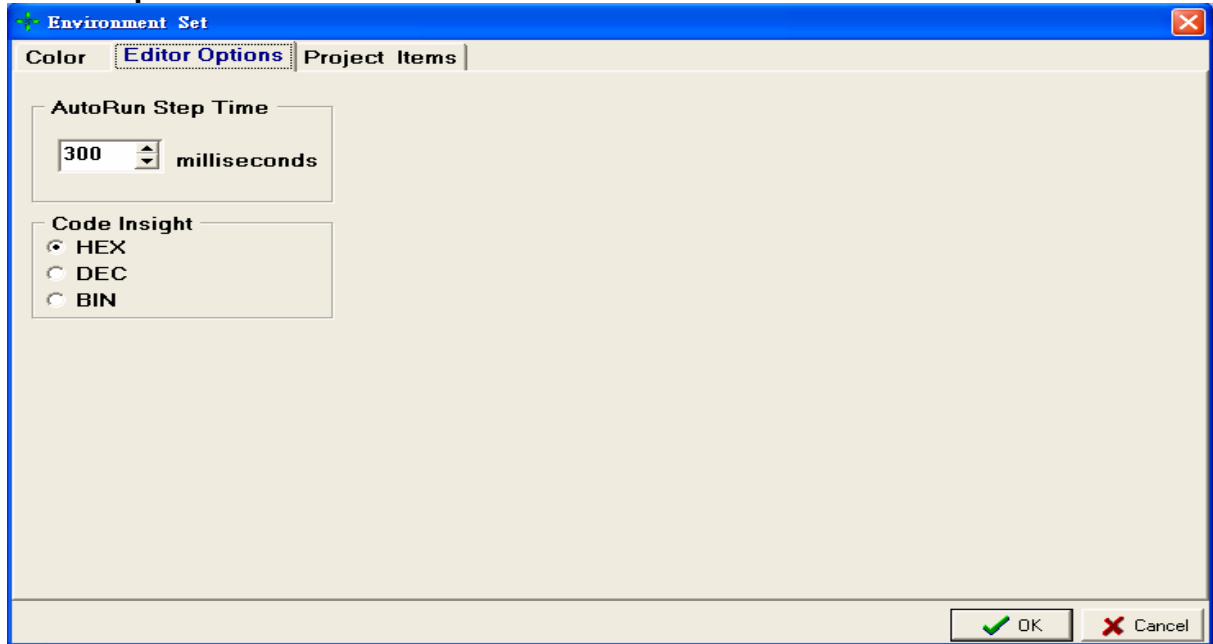
a. Color

The following window will come up, colors can be changed according to functions.



- A:** Set up main screen background color
- B:** PrgMBox Color Sets up box color
 PrgMBox Text Color Sets up text color
 BackGround Color Sets up background color
 Code Text Color Sets up code text color
 Trace Bar Color Sets up BreakPoint color when enabled
 Disable Bar Color Sets up BreakPoint color when disabled
 Arrow In Color Sets up color when program is running to a certain address
- C:** Adjust RAM window color
- D:** Adjust Register window color
- E:** Adjust Watches and Stacks window color

b. Editor Options



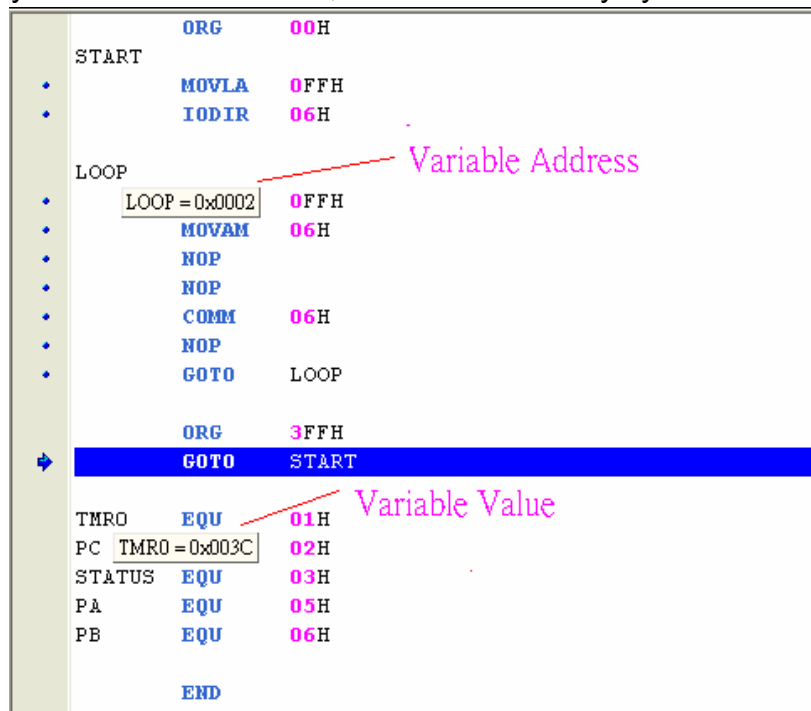
✖ AutoRun Setup Time:

When running 'Auto step' functions, it sets up the time for command to move down from line to line. (unit: millisecond)

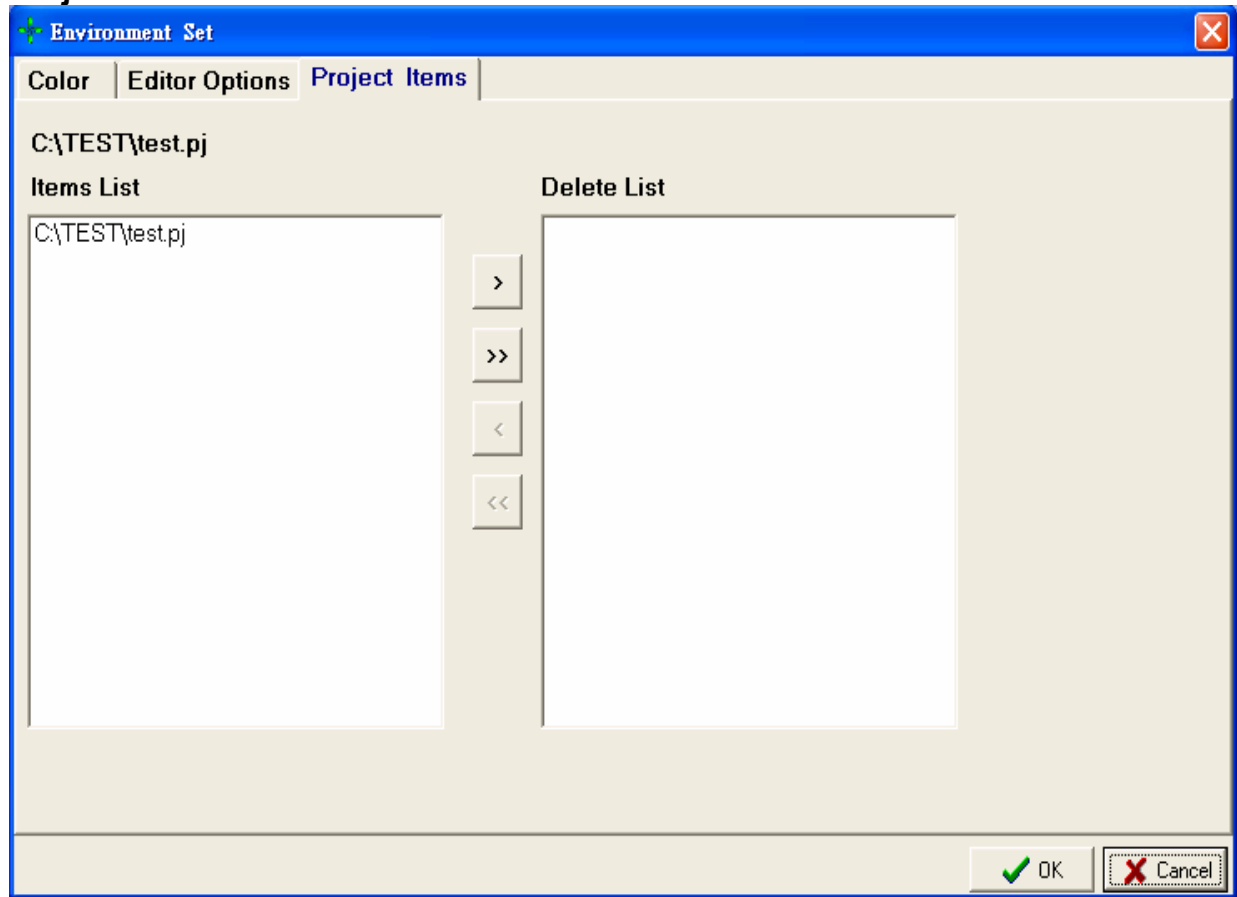
✖ Code Insight:

Only applicable when main screen is under source window, it displays symbol addresses and variable values.

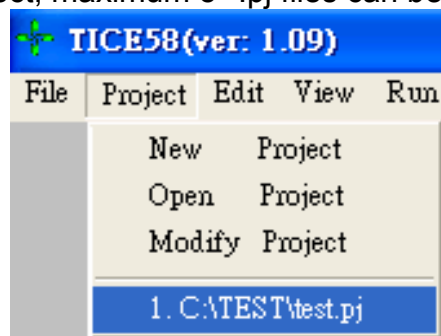
✖ 3 display modes: hexadecimal, decimal and binary systems



c. Project items



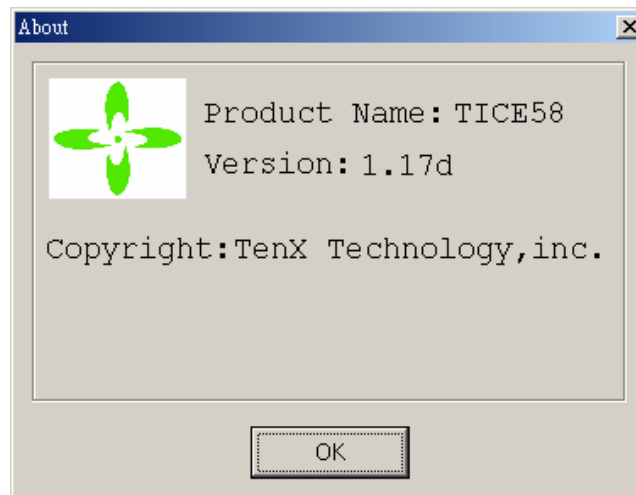
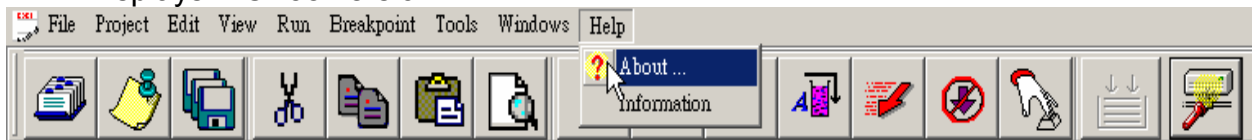
- ✘ When opening each different project, the 'Quick Open' option is added under 'Project' (see below); for those no longer required *.pj files, they can be deleted from here. (Under Project, maximum 5 *.pj files can be listed)



5. Help Window

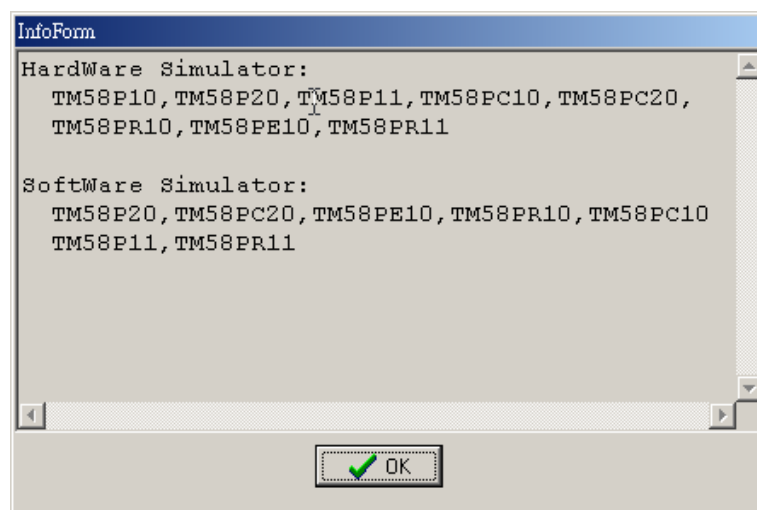
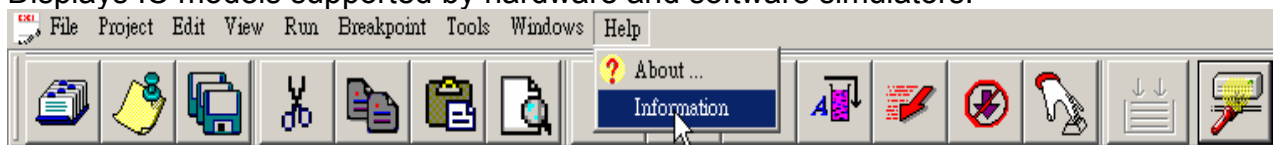
a. About

Displays TICE58 version



b. Information

Displays IC models supported by hardware and software simulators.



Tips on using the software

1. Use of BreakPoint

- a. For easy Debug run, TICE58 software provides several ways of using BreakPoint, users can:
- press F2 key
 - use BreakPoint
 - use Run to Cursor
 - use Break Point List
 - go to the main screen (when BreakPoint occurs)

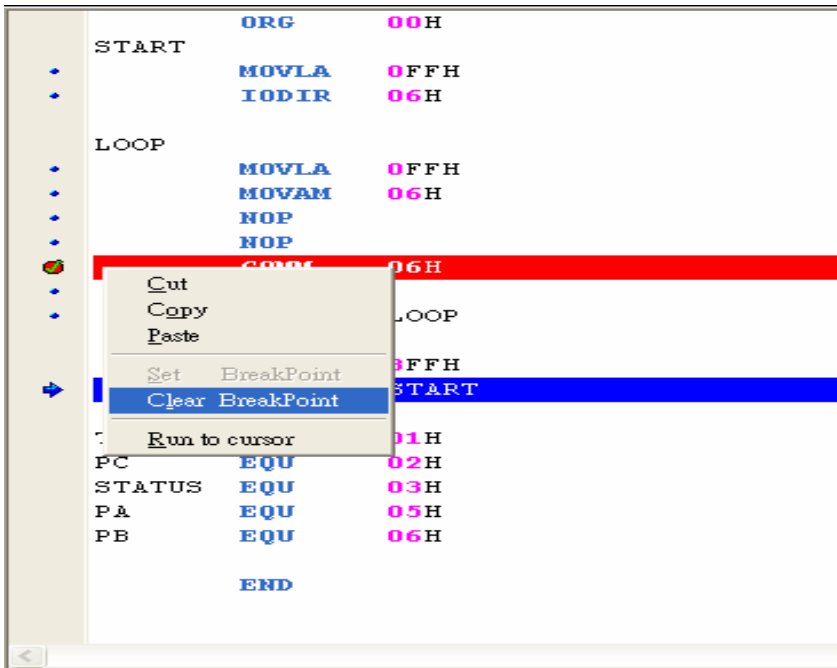
```

clr
iodir    portb
movam    porta
movla    0fh
iodir    porta
movla    05h
select

● movla    b'00100001'    ;
movam    wakeup
movla    b'10000100'
movam    irqm
movm     tmr0,w
movam    time_temp
;
;
start:
call     readtime
● btmisc    irqm,exintm
goto     start
btmss    porta,0
goto     start
bsm      del_flag,0
movla    05h
xoram    del_cnt,w
● btmss    status,z
goto     start
● clrm     del_cnt
btmss    porta,0
goto     start
bcm      del_flag,0
● bcm      irqf,exintf
bsm      irqm,exintm
● goto     start
;
;
;=====
;
;=====

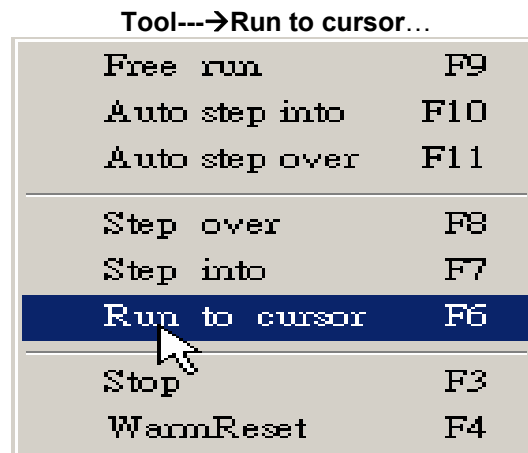
```

- [illegible]



【Use Run to Cursor】

Allows the program to run to where the cursor is positioned.



✘ Move the cursor to where the program code should stop (where the red arrow is pointing to), as below:

```

movla    b'00100001'
movam    wakeup
movla    b'10000100'
movam    irqm

```

✘ Press Run to cursor or F6, the program will stop where is cursor is, see below:

Where the red arrow points to will be highlighted in blue.

```

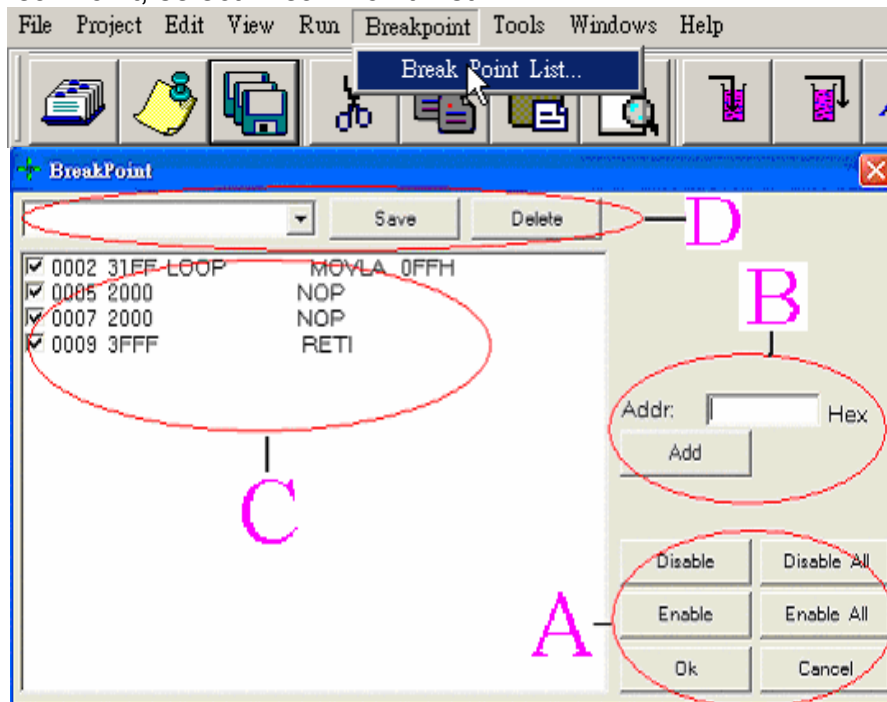
• movla          05h
• select

• movla    b'00100001'
• movam    wakeup
• movla    b'10000100'
• movam    irqm

```

【Use Break Point List】

Click on BreakPoint, select Break Point List



A: Used for monitoring purposes

B: Enter the address value for desired Breakpoint then press

Add

C: Address information of all set Breakpoints

D: This function saves the addresses of set Breakpoints; easy for users to set multiple sets of Breakpoints; users can self define names to be saved; saved data will be stored in *.pj files.

【Main Screen (When Breakpoint occurs)】

A. Displays under source file

```

• nop
• nop
• bsm      portb, led7
• movla    01h
• movam    led_flag
• ret
/
/
/=====
;MAIN PROGRAM
/=====
main:
• clrm     count1
• clrm     del_cnt
• clrm     del_flag
• movla    02h
• movam    led_flag
• movla    01h
• movam    portb
• clra
• iodir    portb
• movam    porta
• movla    0fh
• iodir    porta
• movla    05h
• select
•
• movla    b'00100001'
• movam    wakeup
• movla    b'10000100'
• movam    irqm
• movm     tmr0, w
• movam    time_temp
/
/
start:
• call     readtime
• btmsc    irqm, exintm
• goto     start

```

※Red indicates the set Breakpoint; grey indicates Breakpoint failed to be set up; blue indicates where the program is running.

B. Displays under machine code

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

※Red indicates the set Breakpoint; green indicates where the program is running. <Note: Users can self-define colors. >

2. Use of INCLUDE

By using 'INCLUDE', the main program's program code is shortened; users can read with ease.

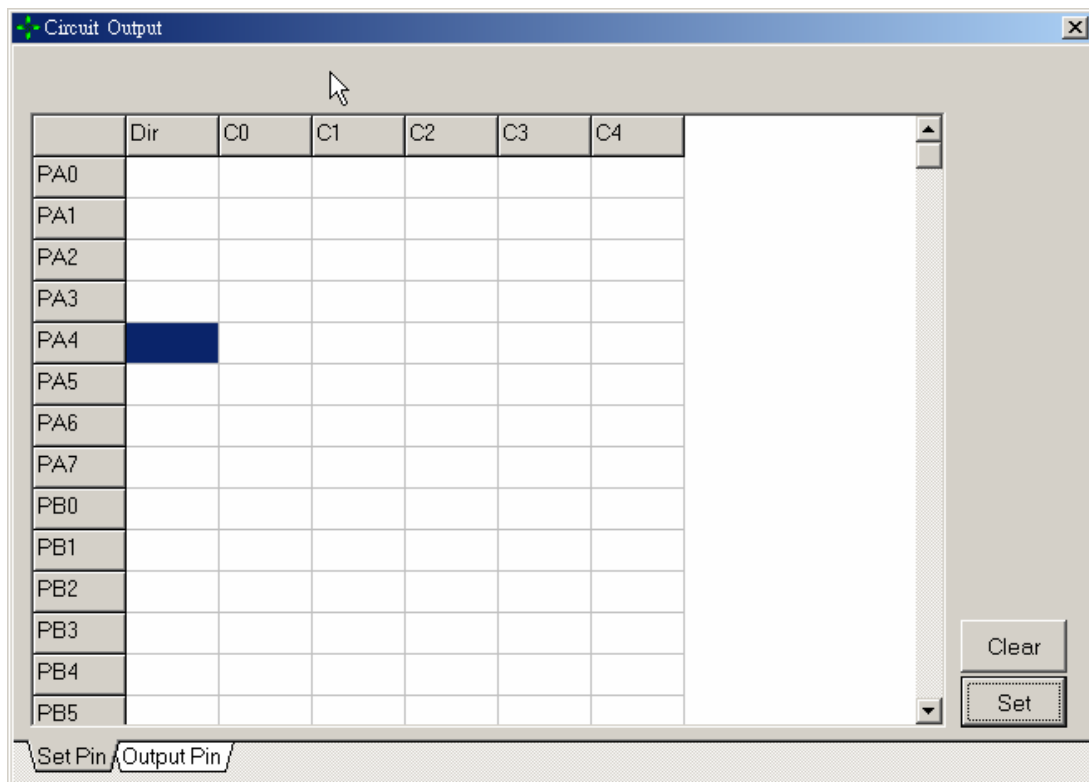
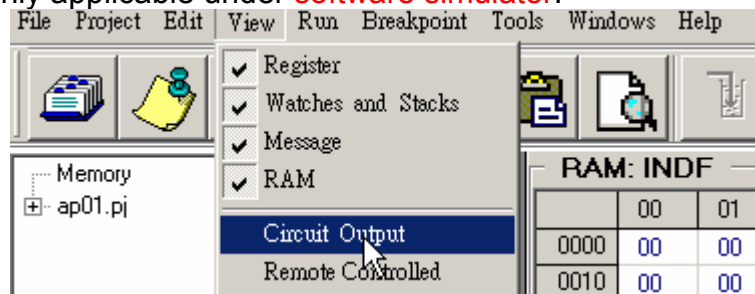
```

;osc1,osc2:clock pin(0)
;reseth0:pa0 be normal i/o pin(0)
;reseth1:reseth1(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     3fih
    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"

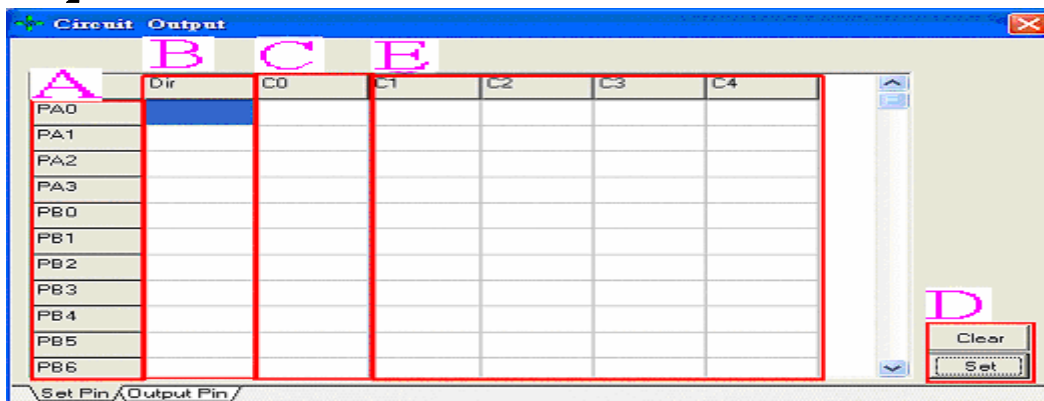
```

3. Use of Virtual Circuit

This function is only applicable under **software simulator**.

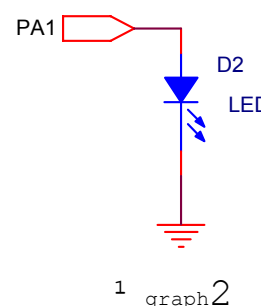
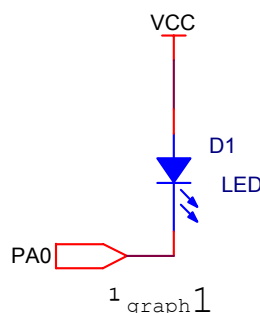
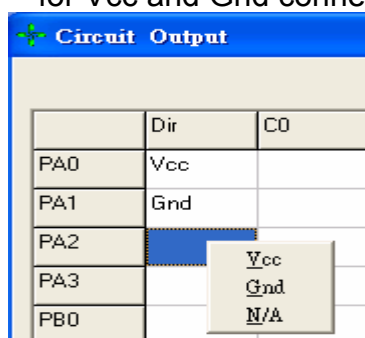


【Set up Pin】

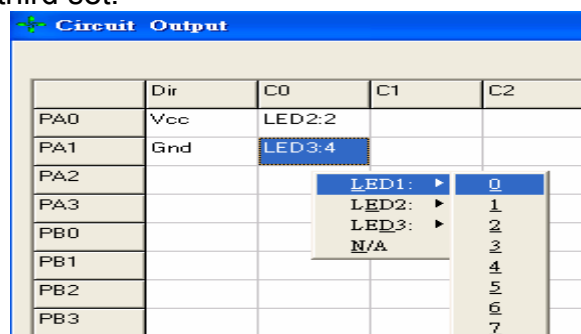


A: IO PIN: this will differ, depending on IC models

B: IO PIN directions: to set up simulated LED connections, right click on mouse to select, "N/A" indicates disconnection; please refer to graph 1 & 2 for Vcc and Gnd connections

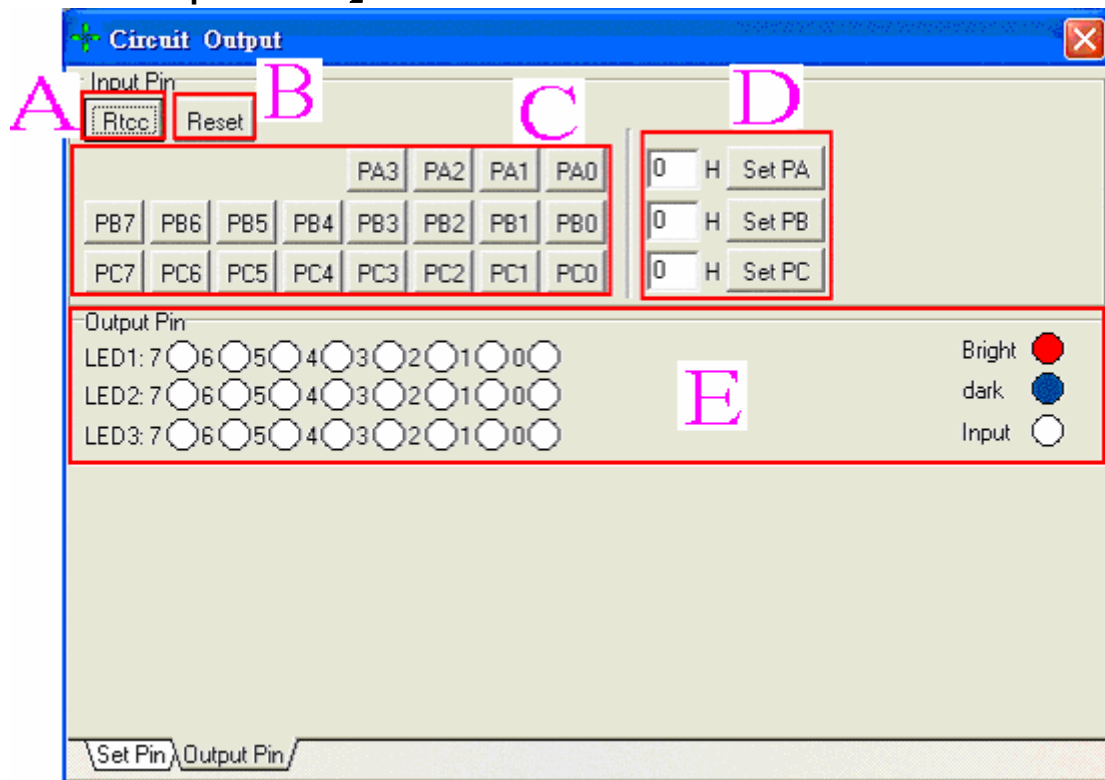


C: C0: Right click on mouse to set up desired position for LED, there are 3 sets of LEDs to choose from, each set consists of 8 LEDs, see below graph; LED2:2 indicates the 2nd LED of the second set; LED3:4 indicates the 4th LED of the third set.



D: After the set-up is done, press **Set** to save, if the key is not pressed, the settings will not be functional. To delete all settings, press **Clear**.

E: Those 4 columns are left blank intentionally.

【Observe PIN Output Status】

A: Rtcc: simulated signals from external clock; press once to add value 1 to TMR0.

B: Reset button, press the button to return the program to initial status.

C: All buttons in the box are functional under Wake-Up or Interrupt; when the configured Wake-Up or Interrupt keys are pressed, corresponding functions will be performed. The function of each key is dependent on various IC models, and not every key is functional. For detailed information, please refer to specifications of each IC.

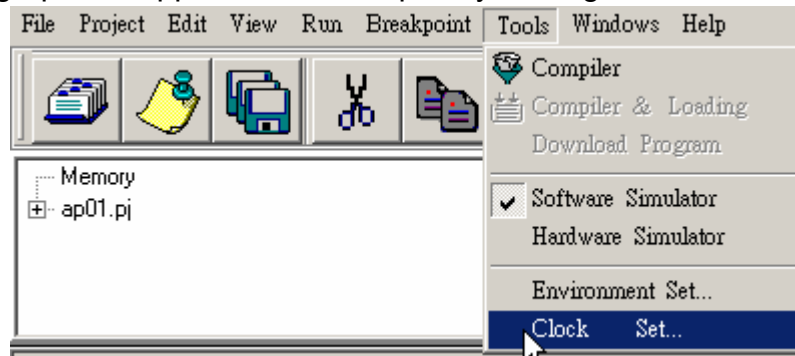
D: Sets up "In/Out value" for I/O port; the In/Out status in I/O is defined by the program. Enter hexadecimal values in the white box, then press 'Set' to export/import the entered values to I/O port.

E: Simulates LED Output, red indicates on, blue indicates off.

【Frequency Settings】

Only functional under **software simulator**.

✘The following graph will appear when 'Frequency Settings' is clicked.



Frequency: Sets up Clock Frequency under simulator.

RTCC: Simulates external TRM Clock frequency; when un-ticked, it is not in use.

INRC Clock: When the chosen IC uses Internal RC, an extra window will appear for Internal RC values to be set up; if the chosen IC has no Internal RC function, the window will not appear.

Points to note

【Note 1】

☐ **External Clock** When selecting internal/external timing pulse, if choosing internal oscillator, there is no need to position it to “ √ ”.

☐ **External Reset** When selecting internal/external timing pulse, if choosing external oscillator, it is necessary to position it to “ √ ”.

【Note 2】

During compiling or debugging, files are automatically saved as *.ASM files, therefore, *.ASM files should not be set as 'read only' files. If they are set as 'read only' files, errors will occur during compiling or debugging.