



十速科技股份有限公司  
tenx technology inc.

**Advance  
Information**

---

# TICE59

# Software

# User's Manual

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

**tenx technology inc.**

TEL: 886-2-82571700

886-3-5737682

FAX: 886-2-82571701

886-3-5737632

[www.tenx.com.tw](http://www.tenx.com.tw)

---

**Preliminary**

**tenx technology, inc.**

Rev.1.5, 2007/08/28

## Index

<b>1. Supported Chip Types</b>	<b>3</b>
<b>2. Hardware Requirements</b>	<b>3</b>
<b>3. Software Requirements</b>	<b>3</b>
<b>4. Software Installation</b>	<b>3</b>
<b>5. User's Guide</b>	<b>4</b>
<b>6. Operation Instructions</b>	<b>7</b>
<b>6.1. Project Management</b>	<b>7</b>
6.1.1. New	7
6.1.2. Save	8
6.1.3. Add File	8
6.1.4. Delete File	8
6.1.5. Configuring Compile Include Directory	8
<b>6.2. Cross Assembler Syntax and Format</b>	<b>9</b>
6.2.1. Functions and Features	9
6.2.2. Compiling	12
<b>6.3. Text Editor Guide</b>	<b>13</b>
6.3.1. The Text Editor included with ICE offers the following features	13
6.3.2. Mouse Actions	13
6.3.3. Keyboard Action	14
<b>6.4. LCD Panel Editor Guide</b>	<b>15</b>
6.4.1. Add Component	15
6.4.2. Add Dot Matrix	15
6.4.3. Query attribute	16
6.4.4. Select Pattern	16
6.4.5. Display Grid	16
6.4.6. Set grid size in LCD window	16
6.4.7. LCD attribute dialog	16
6.4.8. LCD Dot Matrix Definition	17
6.4.9. LCD Pattern Editor	17
<b>6.5. Write .OBJ Code To ICE Board</b>	<b>17</b>
<b>6.6. Code Debugging</b>	<b>18</b>
6.6.1. Debug Mode	18
6.6.2. Single Step Selection	17
6.6.3. Set/Clear Break Point	18

---

6.6.4. View Data During Debug Process-----	17
6.6.5. Control Register View -----	21
6.6.6. Watch Window -----	22
<b>6.7. User Definition -----</b>	<b>23</b>
6.7.1. Set Color/Font-----	23
6.7.2. Hot Key Definition -----	24
<b>6.8. System Config Register-----</b>	<b>26</b>

**1. Supported Chip Types:**

- TM59PA20
- TM59PA40
- TM59PA80
- TM59PE40

**2. Hardware Requirements:**

- IBM PC 586 compatible or above
- TICE59 Board
- Parallel Port Cable
- 9V adapter

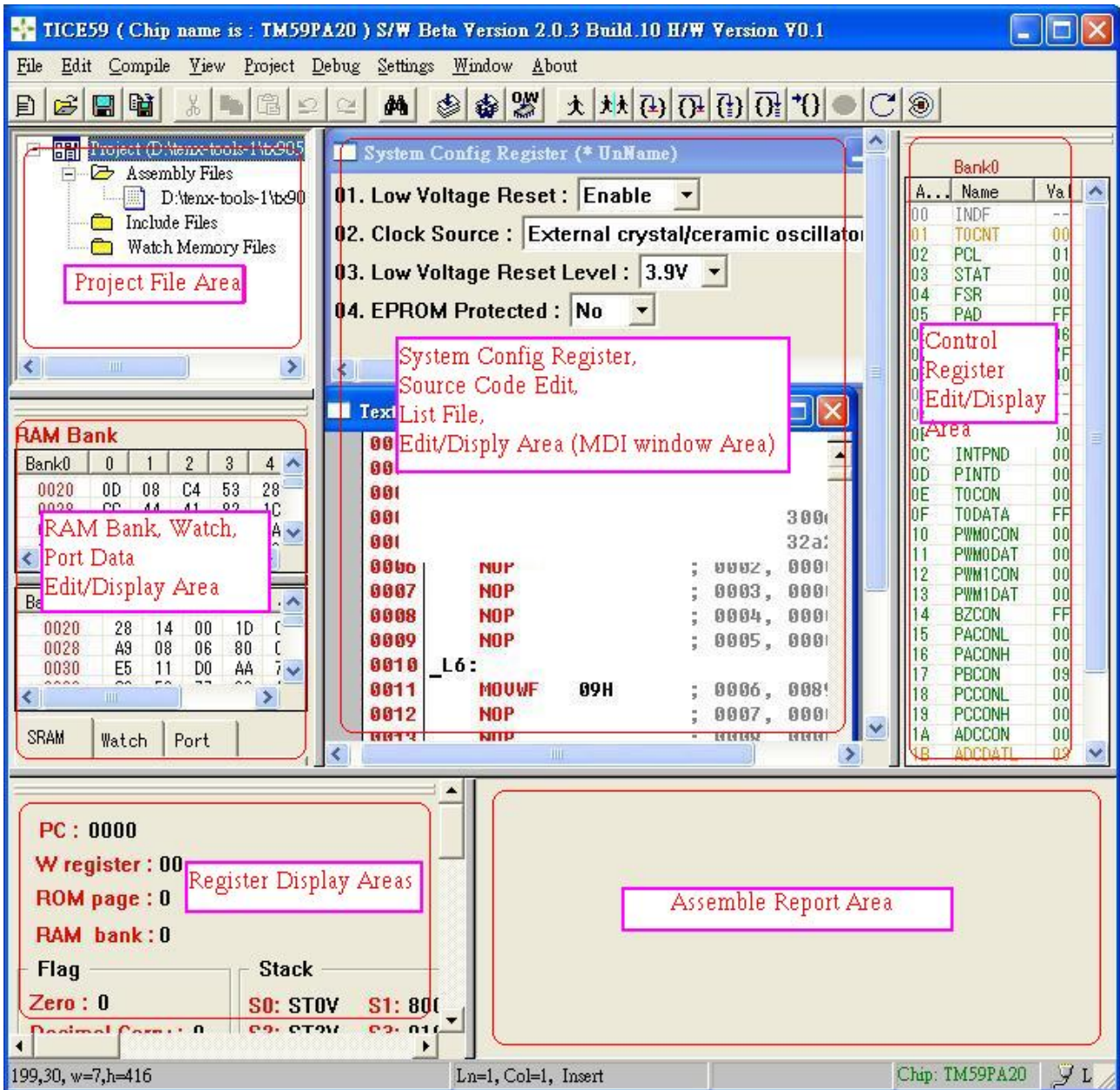
**3. Software Requirements:**

- Microsoft Windows Operating System 9x/ME/2000/XP
- In PC BIOS the Parallel Port must be set as EPP Mode or EPP + ECP Mode
- ICE Windows Version Setup Program: "setup\_TICE59\_Vxxx.exe" (where xxx is the version ID)

**4. Software Installation:**

- 4.1. Run "setup\_TICE59\_Vxxx.exe" to install the TICE59 ICE program. Once the installation is complete the TICE59 executable and associated documentation will be placed within a program group with a shortcut added to the desktop for convenient access.
- 4.2. Run TICE59 exe file to start the ICE program.

5. User's Guide:



Menu Bar Function Description:

5.1. File:

- 5.1.1. New: Create new file: for example, "\*.asm".
- 5.1.2. Open...: Open an existing file: such as "\*.asm" or another text file.
- 5.1.3. Save: Save the "\*.asm" or other text file.
- 5.1.4. Save As...: Save the "\*.asm" or other data as a new file.
- 5.1.5. Exit: Exit the ICE program.

5.2. Edit:

- 5.2.1. While Text Edit window is active:
  - 5.2.1.1. Undo: Revert the last change made.

- 5.2.1.2. Redo: Restore the last Undo action.
- 5.2.1.3. Cut: Cut out the selected area.
- 5.2.1.4. Copy: Copy out the selected area.
- 5.2.1.5. Paste: Paste the data cut or copied to the Clipboard to the cursor's current position.
- 5.2.1.6. Find...: Text string search.
- 5.2.1.7. Find Next: Find the next similar text string.
- 5.2.1.8. Replace: Find and replace a text string with another.
- 5.2.1.9. Goto Line: Jump to the start of the specified Line.
- 5.2.1.10. Select All: Select all characters.
- 5.2.1.11. Set/Clear break point : Set or clear Break point.
- 5.2.1.12. Clear all break point : Clear all Break point.
- 5.2.2. When Program ROM List File window is active:
  - 5.2.2.1. Search: Search through the List File data.
  - 5.2.2.2. Set/Clear break point: Set or Clear Break Point.
  - 5.2.2.3. Clear all break point: Clear all the break points.

### 5.3. Compile:

- 5.3.1. Compile: Compile the assembly language program.
- 5.3.2. Compile and Download code to ICE: After source code is compiling successfully, AP will download OBJ code to ICE directly.
- 5.3.3. LABEL Case Sensitive: When compiling, treat LABEL names (Alphabetical characters) as case sensitive or not.
- 5.3.4. Sorting LABEL and CONSTANT name while outputting file: When generating the output files, optionally sort content of "\*.LBL" and "\*.EQU" into ascending order.

### 5.4. View:

- 5.4.1. RAM Window: Check and save the setting to display RAM window.
- 5.4.2. Register Window: Check and save the setting to display Register window.
- 5.4.3. Control Register Window: Check and save the setting to display Control Register window.
- 5.4.4. Minimize MDI Window: Check and save the setting to display Minimize MDI Window.

### 5.5. Project:

- 5.5.1. New: Create a new project.
- 5.5.2. Open: Open an existing project.
- 5.5.3. Save: Save all project data.
- 5.5.4. Close: Close current project.
- 5.5.5. Add File: Select a file on disk and add it to project.
- 5.5.6. Delete File: Deleted the selected file from the project.
- 5.5.7. Recent File: The project previously created or opened.
- 5.5.8. Download OBJ Code to ICE: Open the "\*.OBJ" file and write its content to the ICE Program ROM.
- 5.5.9. Compile include directories: Set the directories to be used when including files during compiling.

**5.6. Debug:**

- 5.6.1. Go: Debug program in near Real Time mode stopping only at break points or when the Pause or Reset function are used.
- 5.6.2. Free Run: Same as Go but does not stop at break points.
- 5.6.3. Single Step: Debug the program in single step mode. After executing each instruction the program will pause at its current PC address.
- 5.6.4. Auto Single Step: Debug the program in single step mode but do not pause unless a break point is reached, or when the Pause or Reset functions are used.
- 5.6.5. Pause: In Go, Free Run or Auto Single Step mode use this function temporarily suspend the program for debugging.
- 5.6.6. Step Over: If program is being debugged using Single Step Mode, after executing each instruction it will pause at its current PC address. If a Call instruction is encountered it will launch the subroutine then wait until it has completed execution before jumping to the next instruction.
- 5.6.7. Auto Single Step Over: Use Single Step Mode to debug the program but does not pause after executing each instruction. If a Call instruction is encountered it will launch the sub-program then wait until it has completed execution before jumping to the next instruction. Only stop if a break point is encountered, or when the Pause or Reset functions are used.
- 5.6.8. Run to Cursor: Continue executing program until it reaches the address where the input cursor (Caret) is.
- 5.6.9. Goto PC: Jump to the specified Program Counter address.
- 5.6.10. Reset ICE: Reset ICE Board.
- 5.6.11. Initialize ICE: Initialize ICE Board.
- 5.6.12. Pause when Stack Over/Under Flow: When this function is checked, if a Stack Over/Under Flow occurs in Go or Auto Single Step mode then the ICE will pause the program. If this option is not checked the ICE will continue running.
- 5.6.13. Source Code Debug into Macro File: Check this function for source code occurring Marco if debugging is into Macro function or not in Single Step Running status.

**5.7. Settings:**

- 5.7.1. Color/Font...: Set up the font or colors to use in the Text Editor window.
- 5.7.2. Hot Key Definition: Set hot keys for commonly used functions.
- 5.7.3. LCD Dot Matrix Definition: Set the Segment and Common initial values, whether to arrange in ascending or descending order and the interval between lines.
- 5.7.4. Debug Definition: Set file type of debug mode, space amount of List File and timer speed in Auto Step Into.
- 5.7.5. Update USB Firmware: When USB board is update mode, you can update firmware code of USB board.

**5.8. Window:**

Select how sub-windows are arranged and select the currently active window. If a window is minimized or hidden, click on it to restore the window to its original size. If the sub-window is the Text Editor (\*.asm) it will append the opened filename.

**5.9. About:**

Show software version and related information.

**6. Operation Instructions:**

Step1: Check that the Parallel Port has been set to EPP Mode or EPP+ECP Mode in the computer's BIOS.

Step2: Connect the ICE to the computer's Parallel Port.

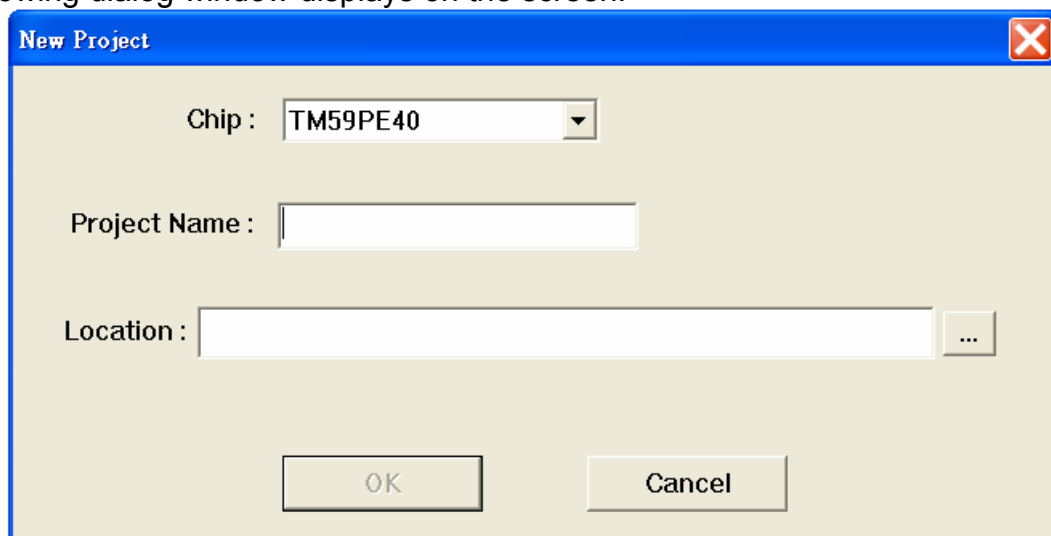
Step3: Turn on the ICE board power supply.

Step4: Run the ICE control program.

If the ICE Board had been powered for the first time, or the last attempt at initializing the board failed then the system will attempt to initialize the ICE Board again.

**6.1. Project Management:**

6.1.1. New...: Under Menu Function, choose Project and the New... function. The following dialog window displays on the screen:



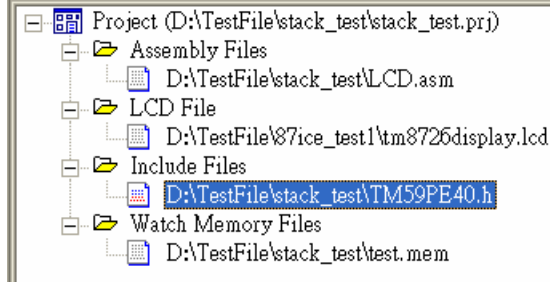
At this point the file type is "\*.prj" and you must complete the 3 following steps:

- (1). Select the Chip No. (e.g. "TM59PA20").
- (2). Enter the Project Name (e.g. "project").
- (3). Location (Path Name): This is where the project will be stored. If the provided path does not exist it will be automatically created by the system.



After completing all 3 of the above steps, please press the [OK] button open the new project.

The Project window will look like the following picture (Example):



Under the project there will be four kinds of file. These are:

- (1). Assembly File: Names of one or more files saved after being edited in the Text Editor. (e.g. “test.asm”).
- (2). LCD File : Names of one files saved after being edited in the LCD Editor. (e.g. “test.lcd”). If chip do not support LCD function, this kind of file will be not displayed.
- (3). Include File: Names of one or more files used in “.asm” using Include File.
- (4). Watch Memory File: Names of one or more files recording the Watch Memory Data displayed in the Debug View window.

6.1.2. Save: After pressing the system will save the project.

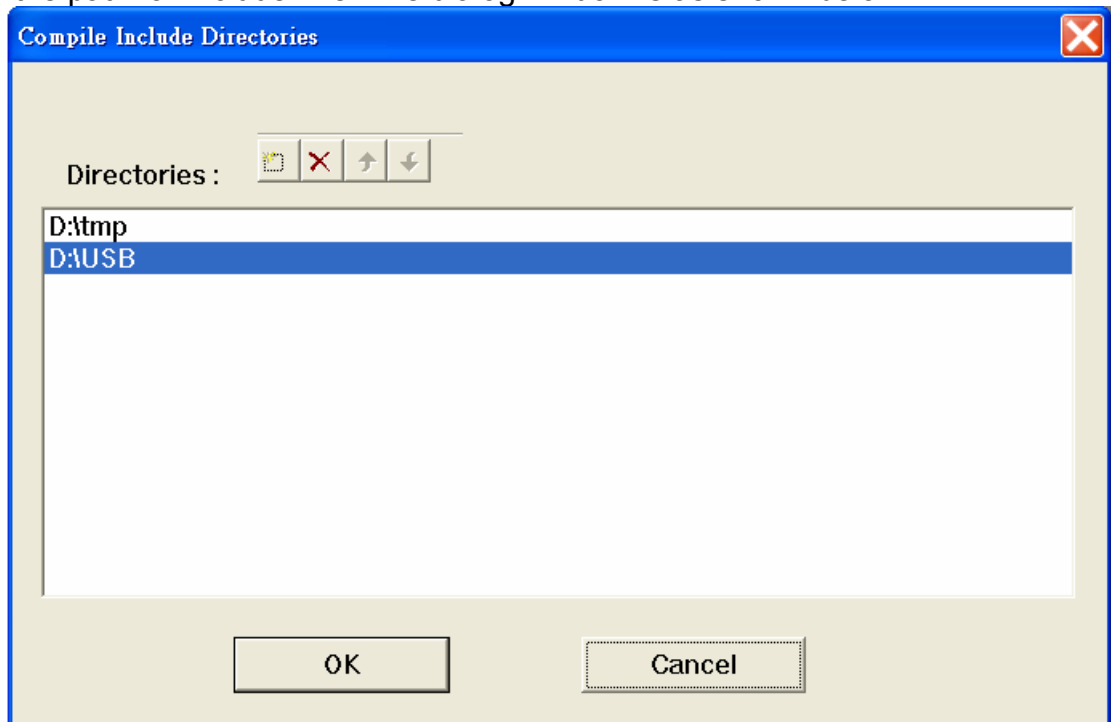
6.1.3. Add File...: Use the mouse cursor to select the file category to add to (e.g. “Assembly File”) then press the Insert key to add the selected file to project.

6.1.4. Delete File: Use the mouse cursor to select the filename under File then press the Delete button to remove that filename. If the Project’s File category was changed, please select Project → Save to save the changes to “.prj” (e.g. “test.prj”). To edit or view a file within the project, move the selection bar to the desired file then press the Enter key or double click the left mouse button to open.

6.1.5. Configuring Compile Include Directory: When the \*.asm code uses the #include instruction, if no path is specified for the file (e.g. #include “MyLIB.LIB” then Include File will search in the following order:

- (1). Project folder.
- (2) ICE system folder.
- (3) User defined path.

If the include file is stored in another path you can use this function to define the path for Include File. The dialog window is as shown below:



**6.2. Cross Assembler Syntax and Format:**

6.2.1. Functions and Features: (1) Macro support. (2). Conditional Assembly.

6.2.1.1. Code can include Label, mnemonic code, Macro or Control Command (e.g. .BYTE).

Examples :

```

PORTA .EQUAL 00H
PORTB .EQUAL 01H
PORTC .EQUAL 02H
MOVA  MACRO  SOURCE, TARGET
MOVF  SOURCE
MOVW  TARGET
ENDM
LABEL:
NOP
MOVA
.BYTE  "HELLO"
.END
    
```

6.2.1.2. Usable Mathematical and Logical Operators:

Operators	Description
~	Unary bitwise not
*	Multiplication
/	Division
%	Remainder
&	Bitwise AND

^	Bitwise XOR
<<	Shift left operator
>>	Shift right operator
+	Binary plus
-	Binary minus
	Binary OR
= (EQ)	Compare operation ( equal )
<> (NE)	Compare operation ( not equal )
< (LT)	Compare operation ( less )
> (GT)	Compare operation ( greater )
<= (LE)	Compare operation ( less or equal )
>= (GE)	Compare operation ( greater or equal )
&& (AND)	Boolean AND
(OR)	Boolean OR
! (NOT)	Boolean NOT

6.2.1.3. Any numbers included within ( ) are given priority in calculations.

For example, in  $2^{(3+6)}$  the (3+6) will be calculated first so  $2^{(3+6)}=20$ .

Binary Expression: 17 can be expressed as 00010001B  
or B'00010001'.

Decimal Expression: 17 can be expressed as 17, 17D or .17.

Hexadecimal Expression: 11 can be expressed as 11H, \$11  
or 0x11.

27 can be expressed as 1BH, \$1B or 0x1B.

255 can be expressed as 0FFH, \$FF or 0xFF.

6.2.1.4. Control Commands:

(1). .BYTE: Define as Byte size data and this can be expressed as a string.

Example: .BYTE "Hello" , \$11, \$22

(2). .DW: Define as Word size data.

Example: .DW \$1234, \$5678

(3). . DEFINE: Define a certain value.

Example: .Define TRUE 1  
.Define FALSE 0

(4). .DB: Same as "(1). .BYTE".

(5). .DS: Reserve blank space in the program.

Example: .DW 30 ; 30 bytes space.

(6). . ELSE: Conditional compilation – execute if current condition not met.

- (7). `.ELSEIF`: Conditional compilation - if current condition not met then check against new condition.
- (8). `.END`: End compilation. Disregard any remaining assembly code.
- (9). `.ENDIF`: Conditional compilation – terminate `.IF` or `.ELSE` conditional branch.
- (10). `.ENDM` ("`.ENDMAC`", "`.ENDMACRO`"): End of Macro definition.
- (11). `.EXITMAC` ("`.EXITMACRO`"): Immediately terminate Macro expansion.
- (12). `.IF`: Conditional compilation – evaluate whether the condition specified in the express is met before continuing with compilation. The expression must be a Constant.  
Examples: `.IF MONTH>6`
- (13). `.IFDEF`: Conditional compilation – check if a Symbol is defined. If a Symbol is defined then the condition is TRUE; if not then the condition is FALSE.  
Example: `.DEFINE ALARM 1`  
`.IFDEF ALARM ; condition is TRUE`
- (14). `.IFDIFF`: Conditional compilation – check if the two values are different. If different then the condition is TRUE; if not then the condition FALSE.  
Example: `.IFDIFF "A", "B" ; Condition is TRUE`  
`.IFDIFF "A", "B" ; Condition is FALSE`
- (15). `.IFNDEF`: Conditional compilation – check if a Symbol is defined. If a Symbol is not defined then the condition is TRUE; if defined then the condition is FALSE.
- (16). `.IFSAME`: Conditional compilation – check if the two values are the same. If the same then the condition is TRUE; if not then the condition FALSE.  
Example: `.IFDIFF "A", "B" ; Condition is TRUE`  
`.IFSAME "A", "B" ; Condition is FALSE`
- (17). `.MAC` ("`.MACRO`"): Begin defining a Macro. The Command must be preceded by a Macro Name and Parameters can be defined after the Command. If there are multiple Parameters to be passed, then each Parameter must be separated by a "Comma".  
Example: `MOVA MACRO SOURCE, TARGET`  
`CLRF SOURCE`  
`INCF TARGET, 0`  
`ENDM`

(18). .ORG: Redefine the Program Counter (PC) address.

Example: .ORG \$0300

(19). .WORD: Same as "(2). .DW".

(20). .EQU (" .EQUAL"): Declare a constant.

Example: PORTA .EQU 10H

(21). .DEFSTRING or DEFSTR: Define a text string.

Example:

USBE .defstring 10H, 7

USBE2 .defstr bsf 10h, 7

bsf USBE ; equal to: bsf 10h, 7

USBE2 ; equal to: bsf 10h, 7

(22). .INCLUDE or #INCLUDE:

► Include another File. Included files may be nested up to a depth of 8.

Syntax: .INCLUDE "TEST.LIB"

► When Include other files, if no path is specified for the file then the order of search is:

(1). Project folder.

(2). Path where ICE is executed.

(3) User defined path.

6.2.1.5. When editing the code, if a Chip (e.g. TM59PA40) directive (e.g. : BSF 10h,7) is used then the instruction must start from column 2 or after of that Line.

### 6.2.2. Compiling:

When you use the built-in Text Editor or other text editors and finish editing the assembly code, you must save the code in the "\*.ASM" (e.g. "TEST.ASM") format and load it in the Text Edit window (if an existing file). You can now press function key F9 (default) or choose the Compile function from the Menu Bar to begin compiling the code. Once compilation is complete, the following files (with "TEST.ASM" as an example) will be generated:

- (1). TEST.ERR: Record any Error or Warning results. Once the compilation process is finished this file will be automatically opened and displayed on screen. If there is any Error or Warning messages you can double click on it with your mouse cursor or use the ↑ ↓ to move to the location to edit then press the Enter key. This moves your edit cursor to the corresponding line in the source code so you can quickly locate and fix the problem.
- (2). TEST.LST: Records information such as number of lines, PC Address, Machine Code and Error Messages. This is used for Trace during Debug.
- (3). TEST.HEX: The compiled Machine Code is saved as an Intel Hexadecimal format file.

- (4). TEST.LBL: Records Label Name related information.
- (6). TEST.EQU: Records Constant Name (EQU defined information).
- (6). TEST.OBJ: This is a Binary file compose of 16K Bytes of ROM Code for downloading to the ICE Program ROM.
- (7). TEST.ERR: File recording Error or Warning messages.

**6.3. Text Editor Guide:**

6.3.1. The Text Editor included with ICE offers the following features:

- (1). Assemble Code Syntax Highlighting: Instructions, keywords, remarks, numbers and text strings can all be highlighted using different colors.
- (2). Drag and Drop Text Operation: Highlighted text can be copied or moved. Support is also provide for drag and drop operations between windows that support OLE Text Drag and Drop.
- (3) Show Split Windows: The same Text Edit window can be split in to a maximum of 4 views (same data). Each split window can be scrolled independently.
- (4) Unlimited Undo/Redo: Undo/Redo is available for all changes with no maximum limit.
- (5). Clipboard Support. Text can be cut, copied and pasted to the clipboard using the right-click menu or a keystroke.
- (6). Microsoft IntelliMouse Support: When using Microsoft IntelliMouse easier scrolling, word selection and line selection is supported.

6.3.2. Mouse Actions (Some may require the Ctrl or Alt key to be held down at the same time):

Mouse Action:	Result:
L-Button click over text	Change the cursor location
R-Button click	Show the right click menu
L-Button down over selection, and drag	Moves text
Ctrl + L-Button down over selection, and drag	Copies text
L-Button click over left margin	Selects line
L-Button click over left margin, and drag	Selects multiple lines
Alt + L-Button or Ctrl + L-Button down, and drag	Select columns of text (selects block)
L-Button double click over text	Select word under cursor
Spin IntelliMouse mouse wheel	Scroll the window vertically
Single click IntelliMouse mouse wheel	Select the word under the cursor
Double click IntelliMouse mouse wheel	Select the line under the cursor
Click and drag splitter bar	Split the window into multiple views or adjust the current splitter position
Double click splitter bar	Split the window in half into multiple views or unsplit the window if already split

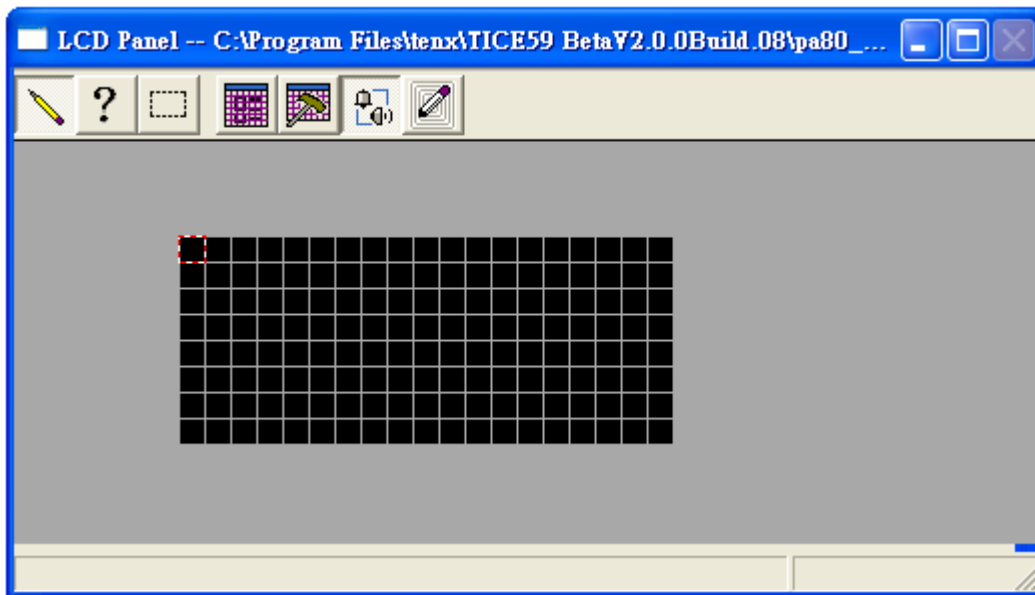
## 6.3.3. Keyboard Action:

Char Left	Left
Char Left Extend	Shift + Left (mark char)
Char Right	Right
Char Right Extend	Shift + Right (mark char)
Copy	Ctrl + C, Ctrl + Insert
Cut	Shift + Delete, Ctrl+X
Delete char or block	Delete
Delete Back	Backspace
Document End	Ctrl + End
Document End Extend	Ctrl + Shift + End
Document Start	Ctrl + Home
Document Start Extend	Ctrl + Shift + Home
Find	Alt + F3, Ctrl + F
Find Next Word	Ctrl + F3
Find Prev	Shift + F3
Find Prev Word	Ctrl + Shift + F3
Find Replace	Ctrl + F3, Ctrl + Alt + F3
GoTo Line	Ctrl + L
Home	Home
HomeExtend	Shift + Home
IndentSelection	Tab
LineCut	Ctrl + BackSpace
LineDown	Down
LineDowNextend	Shift + Down
LineEnd	End
LineEndExtend	Shift + End
LineOpenAbove	Ctrl + Shift + N
LineUp	Up
LineUpExtend	Shift + Up
LowercaseSelection	Alt + L
PageDown	Next
PageDowNextend	Shift + Next
PageUp	Prior
PageUpExtend	Shift + Prior
Paste	Ctrl + V, Shift + Insert
Properties	Alt + Enter
RecordMacro	Ctrl + Shift + R
Redo	Ctrl + Y
SelectLine	Ctrl + Alt + F8
SelectSwapAnchor	Ctrl + Shift + X
SentenceCut	Ctrl + Alt + K
SentenceLeft	Ctrl + Alt + Left
SentenceRight	Ctrl + Alt + Right
Line Repeat	Ctrl + R
TabifySelection	Ctrl + Shift + T


ToggleOvertyp	Insert
ToggleWhitesp	Ctrl + Alt + T
Undo	Ctrl + Z, Ctrl + U
UnindentSele	Shift + Tab
UppercaseSele	Ctrl + U
WindowScroll	Ctrl + Up
WindowScroll	Ctrl + PageUp
WindowScroll	Ctrl + PageDown
WindowScroll	Ctrl + Down
WordDeleteTo	Ctrl + Delete
WordDeleteTo	Ctrl + Backspace
WordLeft	Ctrl + Left
WordLeftExt	Ctrl + Shift + Left
WordRight	Ctrl + Right
WordRightExt	Ctrl + Shift + Right

**Note:** If you wish to save the edited data, use the Save or Save As... functions under the Menu Bar Function (File).

6.4. LCD Panel Editor Guide:

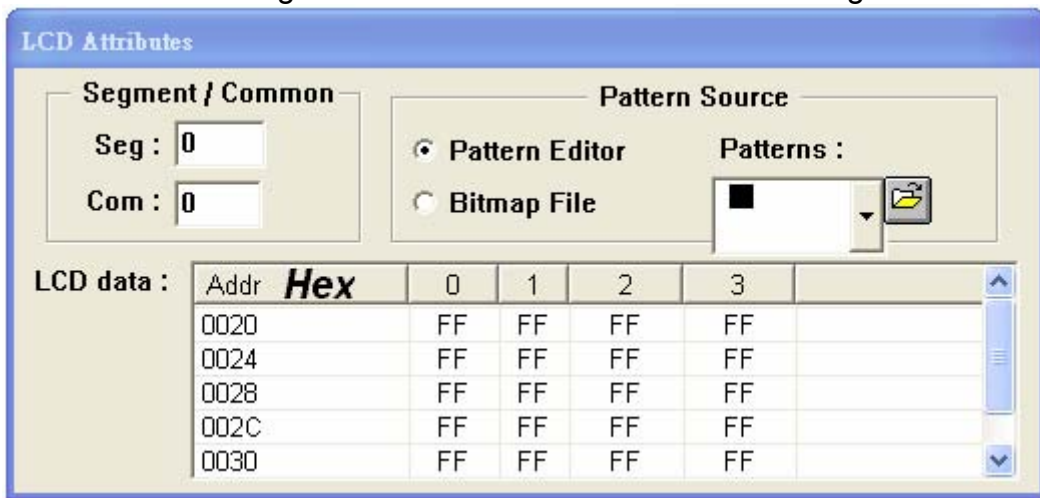



6.4.1. Add Component: In the blank area, click the left mouse button once to add a new component using the pattern or bitmap file set in the LCD attribute dialog. The Segment and Common values from the LCD attribute dialog are then used to configure the new component.

6.4.2. Add Dot Matrix: If the input cursor is within the LCD Panel Editor, from the Menu Bar select Edit → Add Dot Matrix. The cursor becomes . Now hold down the left mouse button and drag until the desired size is reached. The system will use Settings -> LCD Dot Matrix Definition settings and the pattern selected from the LCD attribute dialog to fill the click-and-drag defined area.



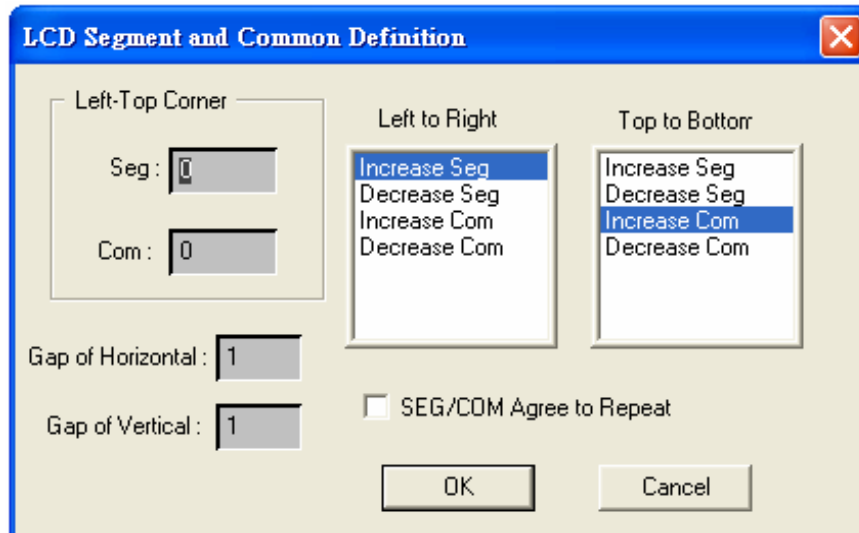
- 6.4.3. Query attribute: Use the left mouse button to select a single component and it will be highlighted by a red/white dotted line border. After making the selection, press the up and down arrow keys to adjust the length and width of the dotted line. If the Ctrl key is held down at the same time then the adjustments will be at 1 pixel intervals. After double-clicking the left mouse button on a component, the system will display and record the Segment and Common values of this component to the LCD attribute dialog.
- 6.4.4. Select Pattern: Hold down the left mouse button and drag until it has reached the desired size. Components within the red selected border can be cut, copied or dragged to the required position.
- 6.4.5. Display Grid: Display or hide the grid lines.
- 6.4.6. Set grid size in LCD window: Set the length and width of the grid lines displayed within the LCD window.
- 6.4.7. LCD attribute dialog: Show or hide the LCD Attributes dialog window.



- 6.4.7.1. Segment/Common: Apart from entering the Segment and Common values manually, the Query Attribute function can be used in conjunction with a double click of the left mouse button on any component within the edit panel. This sends the component's Segment and Common values to the attribute dialog for later use.
- 6.4.7.2. Pattern Source: The user checks the desired pattern or bitmap file.
  - (1). When choosing the pattern to use, the  button the right can be used to open the pattern file (\*.pat) and import the patterns. The pull-down menu can then be used to select the required pattern. Now use the Add Component to add the component in the editor.
  - (2). When a bitmap file is selected for use with the Add Component function, if the bitmap file has not yet been loaded then the dialog window for opening a bitmap file will be displayed so the user can select the required bitmap file.
- 6.4.7.3. LCD Data: Click on the text area before the LCD data to edit the selected LCD Data. Click on the **Addr Hex** button to switch the LCD Data from hexadecimal to binary mode. Conversely, click

on **Addr Bin** to switch from binary back to hexadecimal mode.

6.4.8. LCD Dot Matrix Definition: On the Menu Bar select Settings → LCD Dot Matrix Definition to bring up the LCD Segment and Common Definition Dialog. As below:



6.4.8.1. Configure the Segment and Common values of the component in the top left corner of the Dot Matrix.

6.4.8.2. In the list box at the top right of the above picture, choose whether the Segment and Common direction of the Dot Matrix increases by row or decreases.

6.4.8.3. Set the row interval of the Dot Matrix.

6.4.8.4. Check to indicate if the Segment and Common values within the Dot Matrix are allowed to be repeated.

6.4.9. LCD Pattern Editor: After selecting this function, the LCD Pattern Editor program will be launched so the required pattern can be edited. Please see the LCD Pattern Editor User's Guide for instructions on use.

**Restriction:**

- (1). Only when the Chip supports the LCD function can the LCD Panel Editor be used.
- (2). When the system is in Go or Free Run Debug mode, display LCD function is not supported due to hardware (EV Chip) restriction.

**6.5. Write .OBJ Code To ICE Board:**

When the source code (\*.asm) has been compiled with no errors, the system will generate a ".OBJ file" (e.g. "Test.OBJ"). Before debugging the code the .OBJ code must be downloaded to the Program ROM on the ICE (select Menu Bar → Project → Write OBJ Code to ICE). After all the data has been written to the ROM memory, the system will verify the correctness of the loaded data. If there is an error in the loaded data please perform the write function again. If error persists, check the ICE Board.

Once the download is successful, the system will Reset Board. The program will start again from the Reset address (PC=0) (So during the Debug process (e.g. Go, Free Run...) do not download code data to the Program ROM Memory). The system will then automatically open the ".LST" file (e.g. "TEST.LST") and show this in the Program ROM List File window to serve as a reference during debugging.

## 6.6. Code Debugging:

6.6.1. Debug Mode: From the Menu Bar, select under [Debug] the debug mode:

- (1). Go: Run the program in near real-time mode and pause at break points. The [Pause] function can be used to suspend the execution of the program to view the current PC Address, CPU register and RAM data.
- (2). Free Run: Same as "GO" though the program is not interrupted when it encounters a break point.
- (3). Single Step: Execute the program step by step. The current PC Address, CPU register and RAM data can be viewed as well. To continue executing the program, press the F7 key (default) or the Single Step button (on the Tool Bar).
- (4). Auto Single Step: Operates in the same way as Single Step though no button is required to automatically go to the next step. To pause the program, set a break point or at the Menu Bar select Debug → Pause.

### **Attention:**

You can cycle between the four modes described in (1), (2), (3) and (4) without having to pause or hit Reset.

6.6.2. Single Step Selection:

At the Menu Bar check [RUN] → [Step Into Subroutine] to let the currently running program go into a subroutine (Call); uncheck to not allow subroutine calls.

6.6.3. Set/Clear Break Point:

Break Points can be set or cleared in the \*.ASM(Source Code File) and \*.LST (Program ROM List File) window, and both break point correspond to each other. Break points can only be set for a maximum of 8 addresses, with the maximum Break Point Counter value for each address being 1

(ON/OFF). There are 2 ways to set a Break Point in the List File window:

- (1). First move the caret (using the keyboard or left mouse button) to the desired address (In Source Code Editor, user can set break point anywhere. In List File, user must set an effective address or the break point can't be set.) then use the Action key (default is F6) to set or clear the Break Point at the address.
- (2) On the address to set or clear a Break Point for double left click with the mouse.

### **Attention:**

At Debug: Break Points have no effect under the Go mode.

After ASM File compiling, AP will clear all break point that set on ineffective address.

6.6.4. View Data During Debug Process:

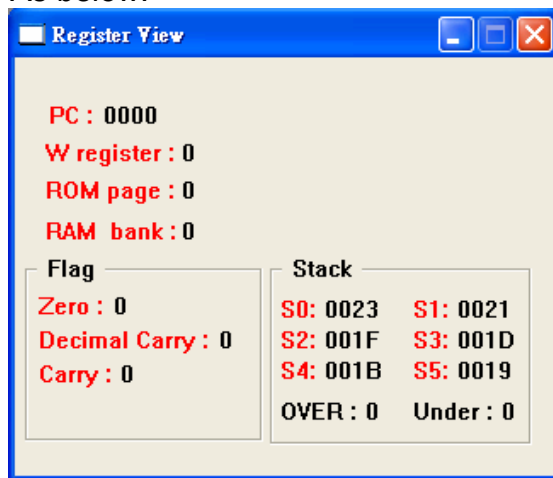
In Single Step (including Auto Single Step) mode of the Pause state the changes in Register and RAM Data can be viewed. This is not available during Free Run (including Go) mode.

(1). CPU Register Data:

The data that can be viewed in the Register View window include:

- (A). Program Counter (PC).
- (B). W register.
- (C). ROM page.
- (D). RAM bank.
- (E). Flag: Zero, Decimal carry, Carry.
- (F). Stack: Stack0~Stack5.
- (G).Stack Over/Under Flag.

As below:



(2). SRAM/Watch/Port Data:

- (A). SRAM window: Show the Bank0 and Bank1 data. The user can change the data at any address. Use the left mouse button to select and edit an address. When editing, use the arrow keys Up, Down, Left and Right or the Tab key to change address.

**Note:**

As RAM data can't be modified in Debug: [GO] and [Free Run] modes, the data displayed is the memory state before entering the [GO] and [Free Run] modes.

As below:

**RAM Bank**

B...	0	1	2	3	4	5	6	7
0020	05	20	80	20	C7	D5	02	5C
0028	D2	0C	90	24	81	4D	90	00
0030	83	05	A4	48	A8	57	02	4C
0038	83	24	68	F3	8C	60	00	C5
0040	A5	84	70	08	A4	02	C4	2D
0048	45	A8	52	01	34	21	5E	3D
0050	4D	70	A4	98	44	03	FC	03
0058	1B	2D	84	B0	49	23	5C	80
0060	14	A1	30	40	11	2A	43	14
0068	F5	08	12	A4	90	62	90	00

B...	0	1	2	3	4	5	6	7
0028	10	62	70	83	14	84	02	CA
0030	03	50	AC	41	04	90	48	04
0038	47	0A	F1	03	88	F3	24	16
0040	39	C0	20	6A	91	02	10	22
0048	4E	34	02	E0	0B	54	A8	54
0050	10	40	9C	24	23	88	44	96
0058	01	8A	81	3E	10	D2	28	90
0060	30	AB	00	30	01	08	64	30
0068	48	24	88	16	E0	23	97	40
0070	C8	40	8B	38	C0	11	40	98

(B). Watch Data: Display the data at RAM addresses (20~7Fh) defined by the user including Bank0 and Bank1, and user can enter value to change address as require. (Please refer to features of address.) As below:

The screenshot shows a window titled "DEBUG View" with a table of watch data. The table is divided into two sections: Bank0 and Bank1. Each section has columns for Name, Addr, and Val. In Bank0, the row for 'clkcon\_1' at address 02 has a value of 1B. In Bank1, the row for 'bzcon' at address 13 has a value of FF. At the bottom of the window, there are tabs for "SRAM", "Watch", and "Port", with "Watch" currently selected.

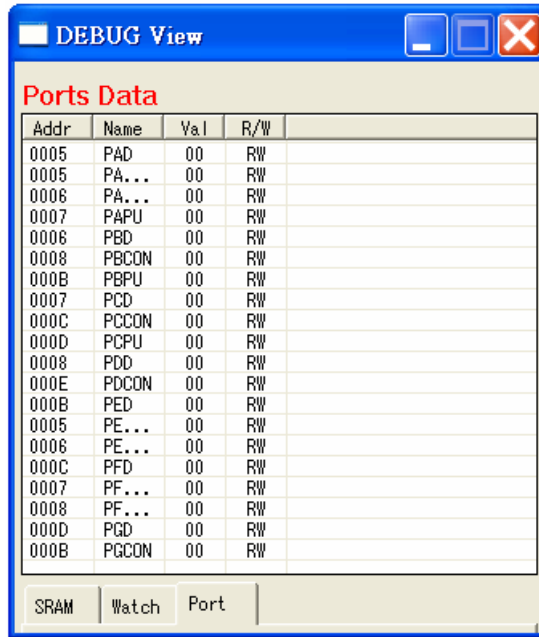
Bank0:Name	Addr	Val
bzcon	13	FF
clkcon_1	02	1B
intcon0_2	10	10
intpnd0	17	00

Bank1:Name	Addr	Val
gpr0	1A	00
bzcon	13	FF

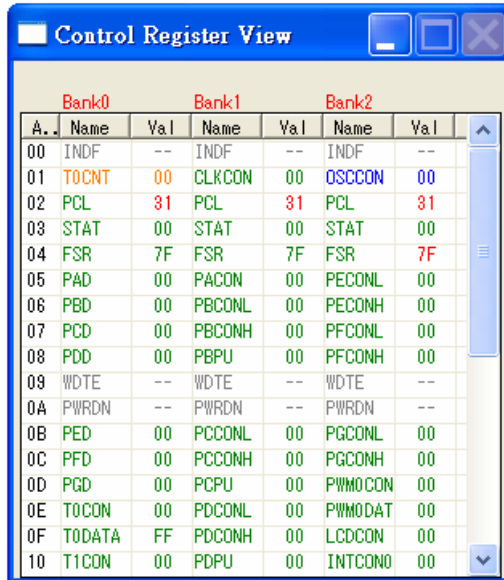
The memory Name and Address to be viewed can be set in the Watch Window to display the current value (0~FFh) of address (20~7Fh). For a more detailed explanation please see "Section 6.6.5".

(C). Port: Show the data at A, B and C Ports. As below:



6.6.5. Control Register View:

Show the Control Register RAM data. If ICE Board initialization failed then this dialog window will be hidden. This window displays the data at debug runtime and distinguishes their properties through color. Orange or gray data can't be modified directly. As below (using the TM59PA80 as the example)



The properties represented by each color are as follow:

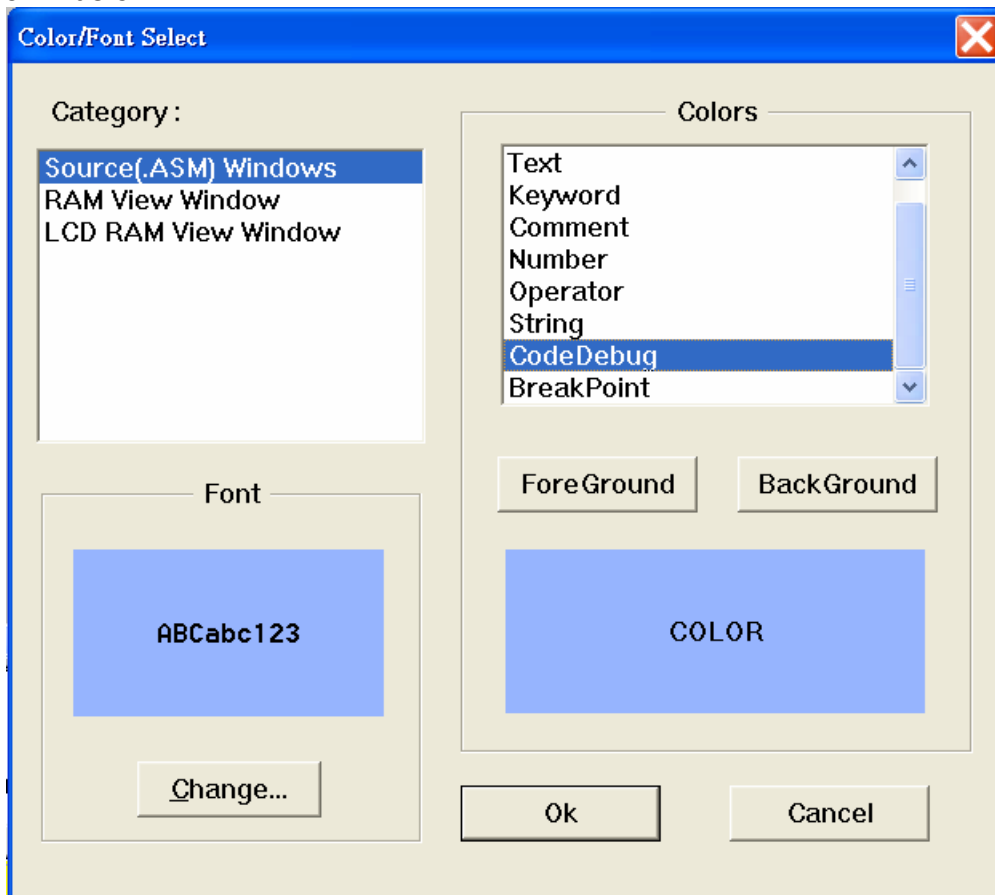
- (1). Red: Presently changing Data.
- (2). Orange: Read Only Data.
- (3). Green: Read/Write Data.
- (4). Blue: Data contains Read only and Read/Write elements.
- (5). Gray: Unused Data.



- (2). Save: Save all the items in the Watch Window under their current filename. If there is no current filename then it must be saved as another file.
- (3). Save As: Save all the entries under Watch Window under another filename (save as new file).
- (4). Open” Open an existing “\*.mem” file and show its data in the Watch window.

**6.7. User Definition**

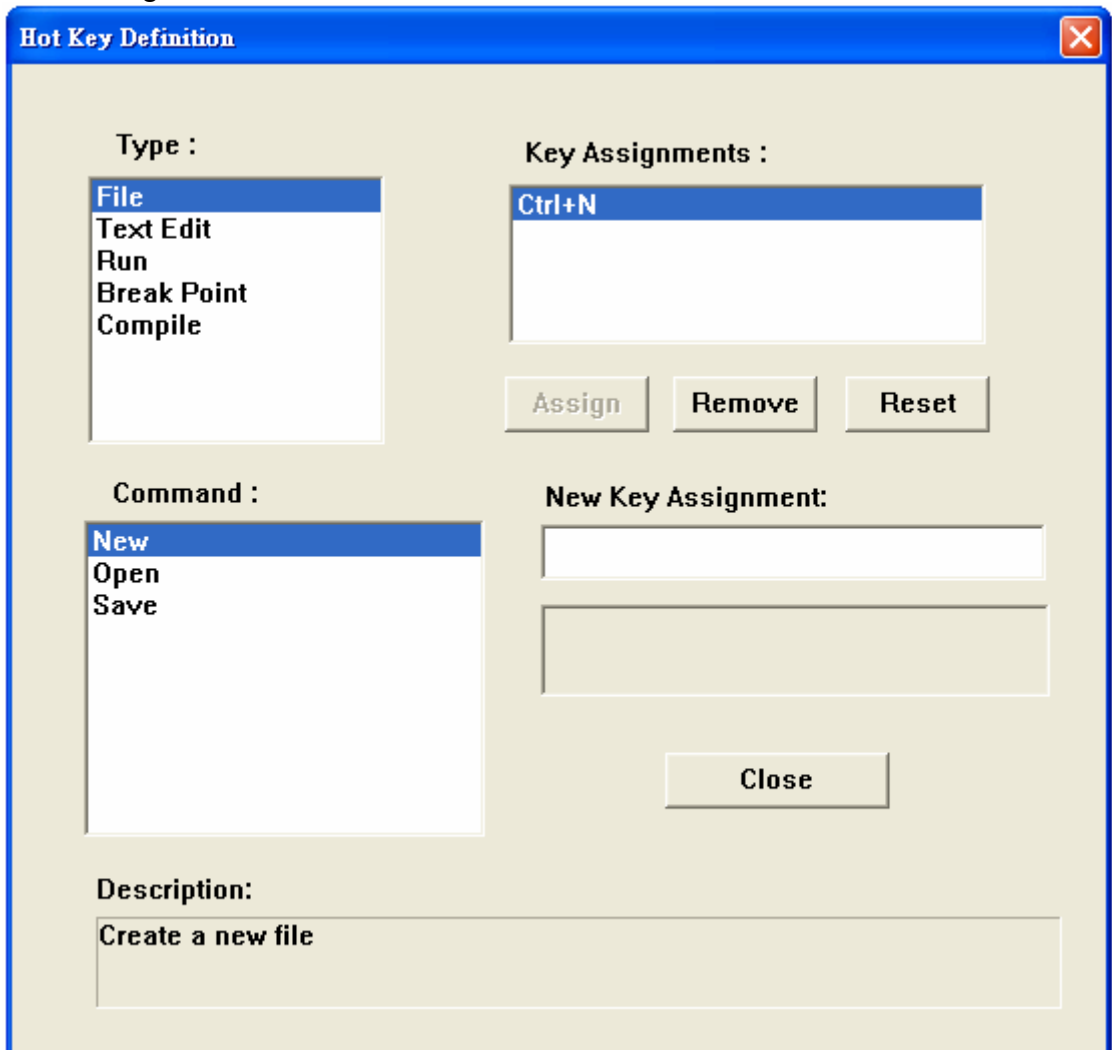
6.7.1. Set Color/Font: On the Menu Bar at Settings → Color Font... the function can be used to set the color and font used in the Text Editor. The dialog window is as shown below:





6.7.2. Hot Key Definition (User Keyboard Definition):

If the keyboard is used to operate the ICE system, some of the frequently used keys can be re-defined using this function or other keys added. The dialog window is as shown below:

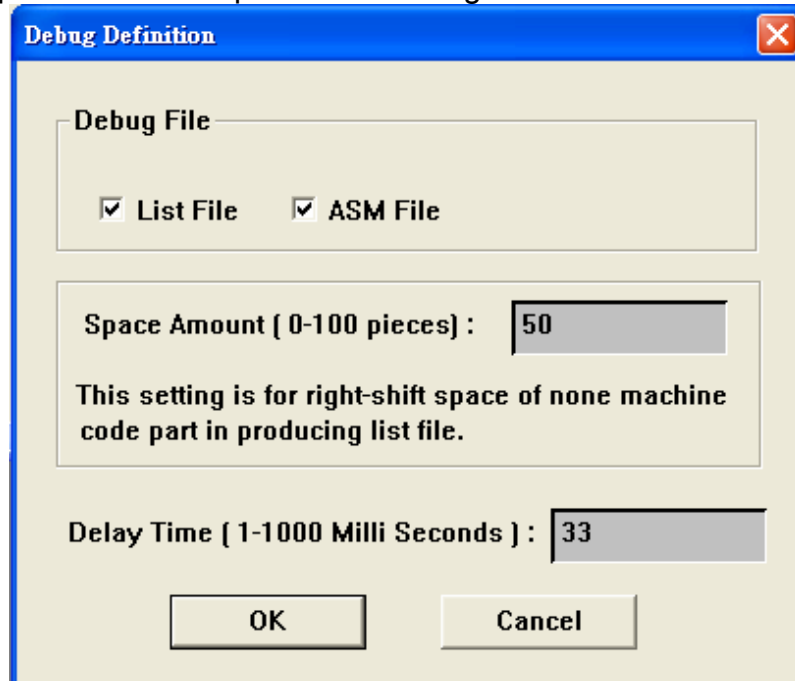


Step:

- (1). Select operation type.
- (2). Select Command.
- (3). Below New Key Assignment, enter the Hot Key.
- (4). Press Assign Button to apply the change.

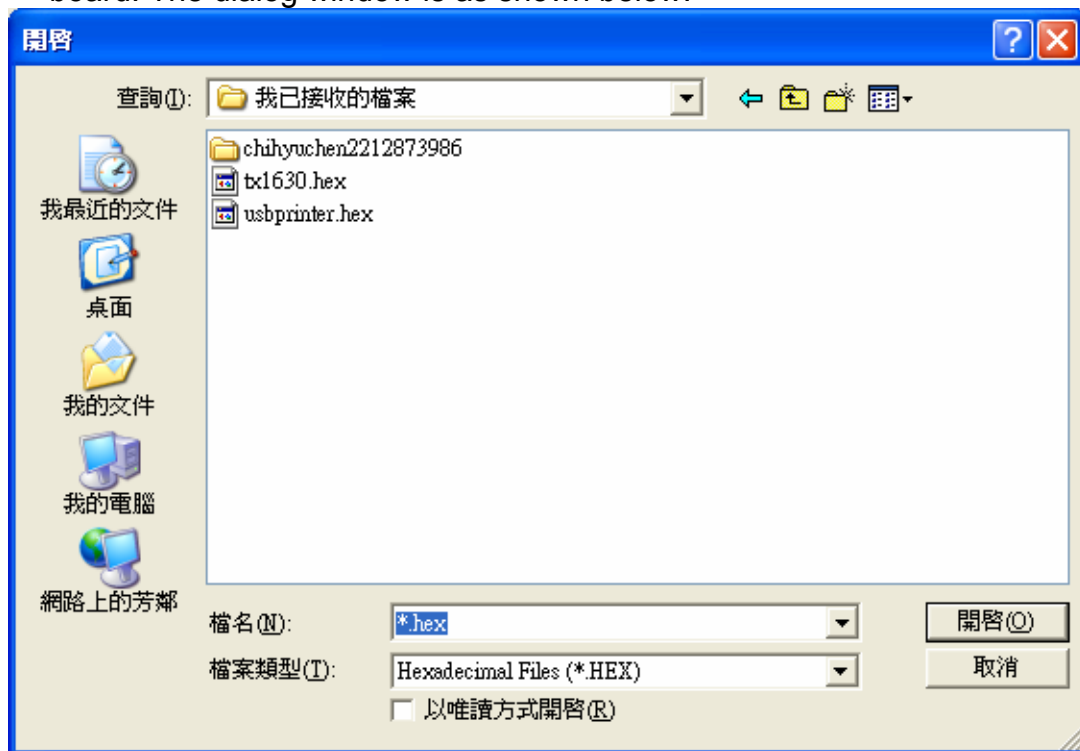
6.7.3. Debug Definition:

On the Menu Bar Menu bar at Settings --> Debug Definition ...the function can be used to set file type of debug mode, space amount of List File and timer speed in Auto Step Into. The dialog window is as shown below:



6.7.4. Update USB Firmware:

When USB board is update mode, you can update firmware code of USB board. The dialog window is as shown below:

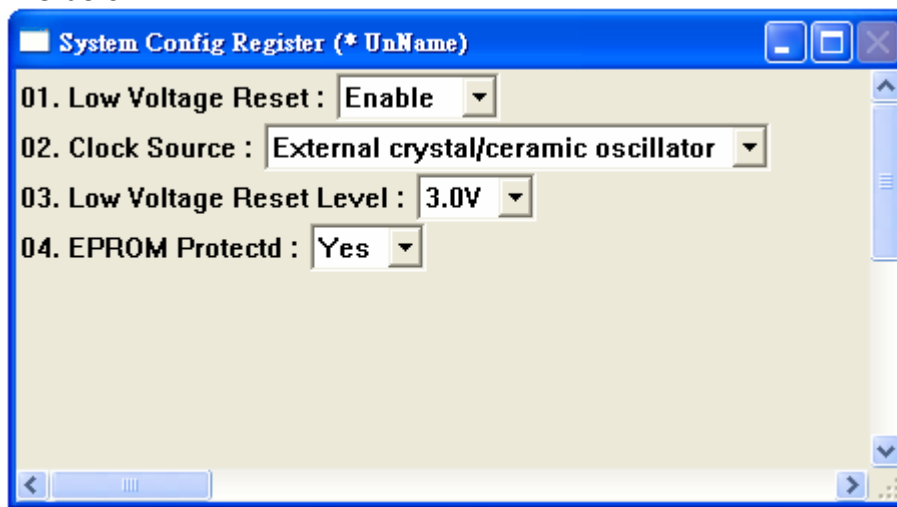


**6.8. System Config Register:**

Depending on the EV Chip there are two different options for the System Config Register. Users can choose between the options before hardware is identified. Once the hardware is identified then it will show the DIP switch settings, which can't be changed through the software.

6.8.1. TM59PA20 and TM59PA40: In the System Config Register window there are four options. These are used to set the System Config Register and Write Protected data. The options displayed are the DIP Switch settings on the ICE board and can't be modified through the software, "01. Low Voltage Reset" is always Enabled. Please refer to the hardware user's guide on the other settings. When the assembly code is successfully compiled, the selected option value (2 bytes) will be written to ".OBJ code (address 0x10)" and ".HEX code (address 0x2000)".

As below:



6.8.2. TM59PA80 and TM59PE40: In the System Config Register window there is an option used for setting Write Protected data. The option displayed is the DIP switch setting on the ICE Board and can't be modified through the software. When the assembly code is successfully compiled, the selected option value (2 bytes) will be written to ".OBJ code (address 0x10)" and ".HEX code (address 0x2000)".

As below:

