Quick Starting

Summarization

This chapter introduces how to use the emulator of TOPICE quickly.

Compiling Source and Debugging

Creating a New Project

Select main menu "Project", then choose the submenu "New" to create a new project. The project file has the name with extension ".prj". The project file must be put at the same subdirectory as the source files. The path and file name should be DOS compatible.

New Project			×
Project Name:	D:\sample\OMF51\	123.prj	Browse
	ок	Cancel]

(picture 1)

Project Name

Type the name (with path) for the new project.

Browse

Display the "Browse Project" dialog box to select or change the name (with path).

•OK

Click to open the "Edit Project" dialog box to create the project files.

Edit a New Project

Select main menu "Project", then choose the submenu "Edit" to edit a project. The files in the project must be put at the same directory of the project file.(The C51 file has the name with extension ".c", the assemble file with extension ".asm" and PL/M51 file with extension ".plm")

Edit Project	×
File Name:	Path:
 prime1.c prime2.c prime3.c sample1.c sample2.asm 	fzqgraph hardfzq ijzfson ieic51 ig500 ieica mycode ieicst ig300 ieica pd300 ieicast ieicast
File Type:	
Source [*.asm; *.plm;*.c;]	ADD DELETE
Files in Project: D:\sample\OMF51\sample1.c	OK Cancel

(picture 2)

•File Name

List files with the filename extension selected in the "File Type" list box.

Path

Select the path you want to contain in the project. After double-clicking on a path, all files with the filename extension selected in the "File Type" list box will be shown in the File Name box.

•File Type

Select the type of file you want to see in the File Name box.

•File in Project

List all files selected from File Name box

•Add

Add the highlighted item in the File Name box project box.

Delete

Remove the highlighted file in Files in Project box from current project.

Setting Compiler Options

Select main menu "Option", then choose the submenu "Environment" to set compiler options. The needed files must be copied to the special path, which include the Assembler named "A51.EXE", C Compiler named "C51.EXE", PL/M51 Compiler named "PLM51.EXE", Linker named "L51.EXE", etc.

Environment	×			
Assembler Path:	C Compiler Path:			
D:\topICE	D: topICE			
PL/M51 Compiler Path:	Linker Path :			
D:\topICE	D: topICE			
Include Path:	Library Path :			
D:ttopICE	D: topICE			
Assembler Options: C Compiler Options:	SB XREF DB COND NOREGPARMS OE OT(0) DB PR RB(0) L/			
PL/M51 Compiler Options:	DB SB OT(0)			
Linker Options:	DP DL DS SY			
Delay in Step(Trace into) continually: 10 OK Clear code buffer when loading code Cancel				

(picture 3)

Assembler Path

Type the path that contains the Assembler named "A51.EXE".

•C Compiler Path

Type the path that contains the C Compiler named "C51.EXE".

•PL/M51 Compiler Path

Type the path that contains the PL/M51 Compiler named "PLM51.EXE".

Linker Path

Type the path that contains the Linker named "L51.EXE".

Include Path

Type the path that contains include files that are provided with the standard library.

•Library Path

Type the path that contains the compiler library.

Assembler Options

Type the Assembler options separated by space.

•C Compiler Options

Type the C Compiler options separated by space.

•PL/M51 Compiler Options

Type the PL/M51 Compiler options separated by space.

Linker Options

Type the Linker options separated by space.

Delay

Type the interval between two step when step continually.

•Clear Code

Clear the code buffer when load the program code.

Suggested Compilers

Franklin(Keil) tool chain C51(*.OMF) versions V3.20 or above C Compiler Version 3.20 or above Assembler Version 4.86 or above PL/M51 Compiler Version 1.40 or above Linker Version 3.11 or above

Build Project

Select main menu "Project", then choose the submenu "Build Project" to compile project (according the options made in the previous session). If no error, "Build" window displays the message "Build OK" and the system loads the code and downloads the code to the emulator. Otherwise, "Build" window displays error messages. You can locate the error line in the source window when double-click the error message in the "Build" window, modify the error and build project again.

Note: Please make sure that all used compiler files.

Load Code

Select main menu "File", then choose the submenu "Load as".

Source Level Debugging

Choose any submenu of main menu "Run" according to your need. Source Level Debugging works only with files in Intel OMF or Extended OMF format which contain necessary debugging messages including LINE numbers..

Debugging from Source Window

Select main menu "File", then choose the submenu "Load File" to open source window. In source window you can view the source file, start and stop emulation, set breakpoints, step over and trace into, and watch variables. You can also modify the source file and rebuild the project.

NOTE: Programs start in assembly code and not in main() in C source level debugging. You can set breakpoints in source window and run to breakpoint.

Mixed Source and Disassemble Window

Select main menu "View", then choose the submenu "Source and Disassemble" to open the window that displays the program source

intermixed with the assembly language compiled for each source line.

Setting Breakpoint

Select main menu "Run", then choose the submenu "Insert/Remove Breakpoint" or move the mouse to the left of the source line and double-clik the mouse. If a breakpoint is set, a breakpoint symbol will appear to the left margin of the source line.

Setting breakpoint must be done after loading code. If you try to set a breakpoint on a non-executable statement, an invalid breakpoint symbol appears.



(picture 4)

Emulating to Breakpoint

Select main menu "Run", then choose the submenu "Go". (When the breakpoint is reached, emulation halts before the instruction at the breakpoint address is executed). When emulating, you can open windows including every source window, disassemble window, mixed source and disassemble window. You can switch the window from one to another.

Stepping Source

This function allows single step of the program in source or assembly mode, including "trace into", "step over", "run to cursor", "Go over return", "trace into continually" and "step over continually".

View Register

Select main menu "Window", then choose the submenu "CPU" to open CPU window in which you can view and modify the register. You can also open the Peripheral Window and Bit Window.

View Memory

Memory includes internal Ram, XDATA (external data memory) and Code (program memory). Select main menu "Window", then choose the submenu "Internal Ram" to open internal Ram window. The XDATA and Code are displayed in "Output" window.

View Variables

Select main menu "Window", then choose the submenu "Variable" to open the window in which you can watch and modify the parameters and variables when emulating.

Discussion about Function

Summarization

The goal of this chapter is to introduce you to the TOPICE user interface that provides a quick and convenient way to create, edit a project, build a project, load the object file and download the code to the emulation.

Menu File

From the File menu, you can open source files:

New File

Create the source file saved as a file with extension name ".asm",".c" or ".plm".

Open File...

Open the existing file.

Save File

Save the editing file.

Save File as...

Save the editing file with another name.



(picture 5)

You can also load object file:

Load as

Load the object file with format:

Intel – OMF51 and Extension OMF51 (Franklin/Keil) Intel – Hex

Binary

FileType	×
Type OMF(51) HEX(INTEL) BINARY AUTO File: D:\sample\OMF51\1	Start Address: 0000 OK Cancel 23.omf

(picture 6)

•Туре

Choose the format of file.

•File

Display the name of the selected file.

•Start Address

The start address in the buffer from which code will be loaded.

Save as

Save the code buffer into a file with format:

Intel – Hex Binary

Exit Exit the system.

Menu Edit

The Edit menu offers the following commands: **Undo** Reverse previous editing operation. **Redo** Repeats previously reversed editing operation.

Cut

Deletes data from the document and moves it to the clipboard.

Сору

Copies data from the document to the clipboard.

Paste

Pastes data from the clipboard into the document.

Delete

Deletes data from the document.

Select All

Selects the entire text within the active document.

Find

Finds the pattern within the active document.

Find Next

Finds the next occurrence of the pattern within the active document.

Find Previous

Finds the previous occurrence of the pattern within the active document.

Replace...

Replaces the pattern for other one within the active document.

Note: When emulating, any editing operation is disallowed in source window.



(picture 7)

Menu Run

Choose commands in the Run menu to start and stop emulation or to step. You can switch the window between source window, disassemble window, source and disassemble window. When emulating, click the right mouse to pop up the menu to switch window.



Start Emulation

Emulation is started with following commands followed as:

GO

Start the emulation and halt at a breakpoint address.

Trace into

This function allows single step of the program in source or disassemble window. If a step begins on a source statement containing a function call for which source is available, it steps into the source for that function and stops at the first executable line in the function.

Step over

This function allows single step of the program in source or disassemble window. If a step begins on a source statement containing a function call for which source is available, it steps over the source for that function and stops at the first executable line in the function.

Run to cursor

Start the emulation and halt at the executable line the cursor locates.

Go over Return

Execute from the current Program Counter and halt at the first executable statement or line (in the calling function) after a return.

Run

Start the emulation freely and ignore any breakpoint.

Trace into continually

This function allows you to trace into continually the program in source or disassemble window.

Step over continually

This function allows you to step over continually the program in source or disassemble window.

Insert/Remove Breakpoint

If the line where the cursor is blinking has been set as a breakpoint, this function allows to remove the breakpoint; otherwise, insert a breakpoint. You can also insert/remove a breakpoint with the mouse. Move the mouse to the left of the line and double-click.



(picture 9)

Remove All Breakpoints

Remove all program breakpoints and XDATA breakpoints. Refer the menu "option" and submenu "set break".



(picture 10)

Halt and Stop Emulation

Halt

Halt the emulation and display the position, you can continue emulating from here.

Stop

This ends the emulation.



(picture 11)

Menu Project

New

Opens the New Project dialog box to create new compiling project.

Open

Displays the Open Project dialog box to open an existing project.

Edit

Displays the Edit Project dialog box to add or remove files in current project.

Close

Close the project, stop the emulation, and close all opened windows.

Build Project

Compile the project according to options, generate a OMF-format file with extension ".omf". If no error, load this file and download the code to the emulator.

Download Code

Download the code to the emulator.



(picture 12)

Menu Option

This function allows you to make setting of software and hardware of emulator.

Select MCU

Select the type of MCU.

	Internal Ram: 0,,7E	
8xC51	internarRant. UX7F	
8xC52	On Chip Rom: 0x0000	
8xC54		
0XU50 90024	XData: 0xFFFF	
80037	Program Memory: 0xFFFF	
W78E51		
W78E52		
VV78E54		
VV78E58 🔍		
VV78E54 VV78E58		

If the MCU has internal ROM, the P0 and P2 are available as I/O when PC is in the internal program memory and serves as a multiplexed address/data bus when PC is in the external program memory.

Set Emulator

Set the option of emulator.

Set Emulater		×
Clock 40MHz(OSC) 20MHz(OSC/2) 10MHz(OSC/4) 5MHz(OSC/8) 2.5MHz(OSC/16)	Rom Emulator Target Set I/O	Ram C Emulator C Target Default
 Emulator Target 	I Using P3.6 as A I Using P3.7 as A	WR OK RD Cancel

(picture 14)

Clock

Select the emulator clock.

•ROM

Emulator: use the program memory in emulator. Target: use the program memory in user board.

Internal Ram

Emulator: use the data memory in emulator. Target: use the data memory in user board.

•Set I/O

Set P3.6, P3.7

Set Break

Set the option of break

• Page "Set"

Add or delete breakpoints of program/XDATA. When the statement reads/writes the XDATA at the address where a breakpoint is set, the emulation halts.

Set Breakpoint 🗙
Set Enable Trigger Mode (External Break)
Breakpoint: Choose(Breakpoint): Program C Read/Write XData Offset: Add Remove
確定 取消

(picture 15)

Breakpoint

List the address of breakpoints.

Choose

Program: list the program breakpoints Read/Write XDATA: list the XDATA breakpoints

Offset

Type the address

•ADD

Set breakpoint at the current address

Remove

Remove the breakpoint at the highlighted address

•Page "Enable"

Set Breakpoint	×
Set Enable Trigger Mode(External Break)	
- Enable	_
🔽 Program Breakpoint 🦵 External Break	
🔽 Read XData Breakpoint	
🖵 Write XData Breakpoint	
When removeall, remove all Program Breakpoint Read/Write XData Breakpoint	
確定取	肖

(picture 16)

• Page "Trigger Mode(External Break)"

Set Breakpoint	×
Set Enable Trigger	Mode(External Break)
Probe1:	Probe2:
C Rising Edge	C Rising Edge
Falling Edge	Falling Edge
_ Probe3:	Probe4:
C Rising Edge	C Rising Edge
🖲 Falling Edge	Falling Edge
	确定取消

(picture 17)

The emulator has four external break probe. When trigger condition is satified , emulation halts.

Enviroment

Set option of compiler.

Environment	×			
Assembler Path:	C Compiler Path:			
D:\topICE	D:\topICE			
PL/M51 Compiler Path:	Linker Path :			
D:\topICE	D: topICE			
Include Path:	Library Path :			
D:\topICE	D: topICE			
Assembler Options: C Compiler Options:	SB XREF DB COND NOREGPARMS OE OT(0) DB PR RB(0) L/			
PL/M51 Compiler Options:	DB SB OT(0)			
Linker Options:	DP DL DS SY			
Delay in Step(Trace into) continually: 10 OK Clear code buffer when loading code Cancel				

(picture 18)

Assembler Path

Type the path that contains the Assembler named "A51.EXE".

•C Compiler Path

Type the path that contains the C Compiler named "C51.EXE".

•PL/M51 Compiler Path

Type the path that contains the PL/M51 Compiler named "PLM51.EXE".

Linker Path

Type the path that contains the Linker named "L51.EXE".

Include Path

Type the path that contains include files provided by the standard library.

Library Path

Type the path that contains the compiler library.

•Assembler Options

Type the Assembler options separated by space.

•C Compiler Options

Type the C Compiler options separated by space.

•PL/M51 Compiler Options

Type the PL/M51 Compiler options separated by space.

Linker Options

Type the Linker options separated by space.

•Delay

Type the interval between two steps.

•Clear Code

Clear the code buffer before loading code.

Refresh XDATA

No useful, you can ignore it.

(picture 19)

Load Target ROM

If target board has an external ROM with program code, you can use this function to load code and disassemble the code, then you can emulate target ROM in disassemble window.

How to use?

Take 89C51 for example.

If user has 8K program code, 4K in the ROM of 89C51 and another 4K in the external ROM. Use this function to load external ROM before emulation,. When emulating, if PC is larger than 4K, you can view in the disassemble window. Follow as:

- 1. Select the type of CPU: 8xC51
- 2. Set the option of emulator, select ROM: Target
- 3. Set environment , uncheck the Box "Clear code buffer when loading code"
- 4. If you want to debug the source level for 4K in the ROM of 89C51, you must load the OMF file.(create a project and build)
- 5. Before emulation, you first load the 4K code in the ROM of 89C51, then use this function to load 4K code in the external ROM.
- 6. Power on the target board.

Note: Load 4K external code, Size of Target ROM is 4+4=8K

Target Rom	×
Size of Target Rom:	8 K bytes
ок	Cancel

<u>(picture 20)</u>

Work Offline

Software simulation only.

Menu View

ToolBar

Show/Hide the toolbar.

StatusBar

Show/Hide the statusbar.

Source and Disassemble

Open the Mixed Source and Disassemble window.



<u>(picture 21)</u>

Disassemble

Open the Disassemble window.

)isAssemb	ole				_ 🗆 🗵
0	0000	0200D4	LJMP	00D4H	_	
ă	opda	900000	NOV	DPTR,#0000H	Dis	
e	000	E4	CLR	A)-	ă	
á	0001	FO	MOVX	0dptr, A	ŝ	
	8000	A3	INC	DPTR	Ë	
ŝ	0 00 9	E4	CLR	A	ble	
•	4004	FO	MOVX	@DPTR,A	ő	
	doop	900001	MOV	DPTR,#0001H	g	
υ	OODE	EO	MOVX	A,@DPTR	ø	
ê	OOOF	0/1	INC	A		
Ĕ	0010	P O	MOVX	@DPTR,A		
ab	0011	7006	JNZ	0019H		
e	0013	900000	MOV	DPTR,#OOOOH		
8	0016	EO	MOVX	A,@DPTR		
đ	0017	04	INC	A		
	0018	FO	MOVX	@DPTR,A		
	0019	900001	MOV	DPTR,#0001H		
	001C	bo∕	MOVX	A,@DPTR		
	001D	24FF	ADD	A,#FFH		•

<u>(picture 22)</u>

Output

Show/Hide the "Output" window which is composed of Build, XDATA and Code subwindow.s

Build window shows the message of building a project. If no error, the window displays the correct compiling and linking messages. In the meantime, the system generates the OMF file and load it, then download code to emulation if working online.

Compile Build OK - 0 error(s)	
	Compile and link correctly
M BUILD 0001 XDATA	oio CODE
Ready	=========== 0VR

<u>(picture 23)</u>

If the compiled file has error, the window shows error messages and the lines where the error occurs.

Compile	
Eile 0 : ERROR 141 IN	LINE 16 OF D:\SAMPLE\OMF51\DISPLAY.C: syntax error near '1)
Build Error - 1 error(s)	
	ouble click the left mouse button to locate error
•	<u> </u>
M BUILD 0101 XDATA	ton CODE

<u>(picture 24)</u>

XDATA window shows external RAM. After the emulation halts, the XDATA refreshes.

00000000	31	32	33	34	35	36	37	38	39	30	FF	FF	FF	FF FF FI	F 1234567890
00000010	74	65	73	74	20	58	44	41	54	41	FF	FF	FF	FF FF FI	f test XDATA <mark>.</mark>
00000020	FF		· · · · · · · · · · · · · · · · · · ·												
00000030	FF	Locat	•												
00000040	FF	(Search	h)												
00000050	FF	Seard	h Neva												

Single click the right mouse button to pop up menu

<u>(picture 25)</u>

Code window shows the executed code.

I	00000000	02	00	74	75	08	AA	75	OA	01	75	09	00	AE	09	AF	ΟÀ	OOt	uD.uD	0น	
l	00000010	74	DO	C3	9F	74	07	9E	40	11	85	08	90	78	OA	74	01	t	.tD.0	٥.١	p.xoto
l	00000020	26	F6	50	04	18	E4	36	F6	50	E2	75	08	55	75	0.	J	ocate	-		սուրոսը
l	00000030	75	09	00	AE	09	AF	ΟÀ	74	DO	CЗ	9F	74	07	9E	4	(s	earch		Υ.	to.00
l	00000040	85	08	90	78	ΟÀ	74	01	26	F6	50	04	18	E4	36	F	1	o or al-	House	/	ϸ⊡.6.Р
l	00000050	E 2	80	B0	22	75	0C	00	75	0B	01	E5	0B	DЗ	94	0_		ar ch	-		loop
Å		_							_			_			_	_		_		_	
L	👭 BUILD J	0101	XD/	ATA	010	n Ci	ODE	:													
					_			_													

Single click the right mouse button to pop up menu

<u>(picture 26)</u>

Locate

Locate to the address in the "Locate to Address" box.

Locate		×
Locate to Address :	2000	
ок	Cancel	

<u>(picture 27)</u>

Search

Locate to the address where the string in the "Text to search" can be founded.

Hex Editor - Search							
Bytes to search :	73 61 6D 70 6C 65						
Text to search :	sample						
	<u>O</u> K <u>C</u> ancel						

<u>(picture 28)</u>

Search Next

Go on searching the address which the string in the "Text to search" box can be founded.

Menu Window

Cascade

Arranges open debugging windows in an overlapping pattern so that the title bar of each window is visible.

Title Horizontally

Arranges opened debugging windows side by side so that all windows are visible.

Title Vertically

Arranges opened debugging windows side by side so that all windows are visible.

CPU

Open the CPU window in which value of all registers are displayed. Changed values are addressed with red color.

CPU				_ 🗆 ×
PC=0000	DPTR=0000	PSW=00	ACC=00	
B=00	SP=07	R0=00	R1=00	
R2=00	R3=00	R4=00	R5=00	
R6=00	R7=00	TL0=00	TL1=00	
TH0=00	TH1=00	SBUF=00	SCON=00	
TMOD=00	TCON=00	PCON=00	IP=00	
IE=00	PO=FF	Pl=FF	P2=FF	
P3=FF				_
				•

Double click the left mouse key to pop up dialog box of modification

<u>(picture 29)</u>

Modify Dialog Box

Modify Register	<u>I</u>	×
Variable:	SP	
Current Value:	0x07	
New Value:		
0K	Cancel	

<u>(picture 30)</u>

•New Value

Type a new value.

Bit

Open the Bit window that displays the value of bit-addressable locations.



(picture 31)

Stack

Open stack window.

SP =	08			
51				ť
LA-	20	+1	89	_
-1	05	+7	00	
-2	00	+3	00	
-3	03	- 4	00	
-4	02	+5	00	
-5	00	+0	09	
Ŧ	00	+7	00	
- 7	00	+8	00	
-8	80	+9	00	

is SP-n

or referrence

<u>(picture 32)</u>

Peripheral

Opens the Peripheral window.

Periph	ieral			
	Symbol 🛛 🖂	Address	Value	Name
÷	IP	B8H	0x00	Interrupt Priority Control 🔺
🕂 🔁	IE	A8H	0×00	Interrupt Enable Control 🛛 🔤
	SCON	98H	0x00	Serial Control
	RI	0	0	
	TI	1	0	
	RB8	2	0	
	TB8	3	0	
	REN	4	0	
	SM2	5	0	
	SM1	6	0	
	SMO	7	0	
÷	TCON	88H	0×00	Timer/Counter Control 🚽 🚽
÷	TMOD	89H	0×00	Timer/Counter Mode Control
	PO	80H	OxFF	Port O
	PO. 0	0	Madifr	v Value
	PO. 1	1	modily	, rande
	P0.2	2	1	
				Þ

Single click the right mouse button to pop up menu

```
(picture 33)
```

Variable

Open the variable window that can be used to inspect or edit variables. If you want to watch the value of DATA, IDATA and XDATA, you can add an untyped variable as:

D: XXXX	DATA Value in address "XXXX"

- I: XXXX IDATA Value in address "XXXX"
- X: XXXX XDATA Value in address "XXXX"

💦 Variable	
Symbol	∆ Value
⊡ <mark>≣</mark> n.Sam	
	0
	1
	2 Add
	3 Zdit
	4 Delete
nSam [5]	5 Delete All
<mark>È</mark> nSam[6]	6
	7 Modify
	8 Haw begins I. Die Low
nSam [9]	9 nexadocimai inspiay
	4672
i	10
n.Sam [5]	5
Address: (XDATA)0x0014 Global	Type: signed int

Single click the right mouse button to pop up menu

(picture 34)

•Add

Add a variable which can be compose of C expression, struct or array.

• Edit

Edit a variable. Use it to change the type of variable which is not defined.

• Delete

Delete a variable.

Delete All

Delete all variables.

Modify

Modify the value of the variable.

Hexadecimal Display

The value is displayed in Hexadecimal.

When debugging C51 file, if you want to display value of variable with type "pointer", you can select the type "array". For example, for variable "char* Name" you can use "Name[0]" to "Name[n]", instead of "*Name".

Internal RAM

Open the internal RAM window. You can type the key "0-----9" and "A----F" in the HEX region. Using key "Tab" switches between two regions.

	ternal	Ram																	_ 🗆 🗵
0000	00000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
poor	91001	31	32	33	34	35	36	37	38	39	30	00	00	00	00	00	00	12345678	390000000
0000	00020	61	62	63	64	65	66	20	20	20	20	20	20	20	20	20	20	abcdef	
0000	00030	73	68	6F	77	20	65	78	61	6D	70	6C	65	20	20	20	20	show exa	ample
0000	0040	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20		
0000)005C	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20		
0000	0060	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20		
0000	0070	00	00	00	00	00	00	00	οÞ	00	00	00	00	00	00	00	00		
0000	0080	00	00	00	00	00	00	00	οþ	00	00	00	00	00	00	00	00		
0000	0090	00	00	00	00	00	00	00	οþ	00	00	00	00	00	00	00	00		
0000	0040	00	00	00	00	00	00	00	οþ	00	00	00	00	00	00	00	00		
0000	00ВО	00	00	00	00	00	00	00	οþ	00	00	00	00	00	00	00	00		
0000	poco	00	00	00	00	00	00	00	οþ	00	00	00	00	00	00	00	00		
0000	DODO -	00	00	00	00	00	00	00	οþ	00	00	00	00	00	00	00	00		
0000	DOEO	00	00	00	00	00	00	00	οÞ	00	00	00	00	00	00	00	00		
0000	DOFO	00	00	00	00	00	00	00	οÞ	00	00	00	00	00	00	00	00		
Offset Address Hevadeci									cin	nal	C	de						ASCT	I Codo

(picture 35)

Project Parameters

Open the internal Project Parameters window that shows the structure of OMF file. You can use "drag/drop" to add the variable into Variable window.

🔁 Parameter of Project				
Parameter	Type	Address		
B ch	signed char	DATA: 0008 📃 🔺		
i i	signed int	XDATA:0000		
PROCEDURE NAME: SET_P3				
主 📄 PROCEDURE NAME: DELAYN				
🖻 📲 MODULE NAME:C:\TOPICE52\sample4\DEMO.c				
P1	unsigned char	DATA:0090		
B P3	unsigned char	DATA:00B0		
main	function	CODE:0011		
	bit	BIT:00B0		
	bit	BIT:00B1		
	bit	BIT:00B2		
	bit	BIT:00B3		
	bit	BIT:00B4 📃		



Appendix

Error Message

1. Not a project file

Because TOPICE software creates a project file in special format, you must

close the project or quit system normally to save the project file, Otherwise, an error may occurs.

2. There is an error type file in project

There are three correct types of file in project, C51 file with extension name ".c", assembly file with extension name ".asm" and PL/M51 file with extension name ".plm".

3. Not an OMF51 File

You can create a project including your files and build the project to generate a correct OMF51 file.

4. Invalid type value

Type the C type value. For example:

In Decimal, you can type: 1234, 45, 78, 09, etc.

In Hexadecimal, you can type: 0x1234, 0x6b, 0xabcd, etc.

Float: 12.345, 0.567, 2e-5, 3e+10

Long: 0x12345678L

5. Code size is too large

For example, if the code size is 4.5K while the MCU is AT89C51, this warning will happen when loading the OMF file.

6. Can not load all debug messages

If you debug a PL/M file, the name must be same as the file name.

Example: *Pdemo*.plm , you must write as:

PDEMO: do;

end *PDEMO*;