



FR Family SOFTUNE™ Workbench Command Reference Manual for V6

Spec. # 002-04576 Rev. *A

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intl): 408.943.2600
<http://www.cypress.com>



Copyrights

© Cypress Semiconductor Corporation, 2002-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you under its copyright rights in the Software, a personal, non-exclusive, nontransferable license (without the right to sublicense) (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units. Cypress also grants you a personal, non-exclusive, nontransferable, license (without the right to sublicense) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely to the minimum extent that is necessary for you to exercise your rights under the copyright license granted in the previous sentence. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and Company shall and hereby does release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. Company shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

Preface

■ What is the SOFTUNE Workbench?

SOFTUNE Workbench is support software for developing programs for the microprocessors / microcontrollers.

It is a combination of a development manager, simulator debugger, emulator debugger, monitor debugger, and an integrated development environment for efficient development.

■ Purpose of this manual and target readers

This manual explains the command and built-in variable/function of the SOFTUNE Workbench in the reference format.

This manual is intended for engineers developing various types of products using SOFTUNE Workbench.

Be sure to read this manual completely.

■ Trademarks

SOFTUNE is a trademark of Cypress Semiconductor Corp..

Microsoft, Windows and Windows Media are either registered trademarks of Microsoft Corporation in the United States and/or other countries.

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

■ Organization of Manual

This manual consists of 13 chapters and an appendix.

CHAPTER 1 Environment Setup Commands

This chapter describes the Environment Setup commands.

CHAPTER 2 Program Execution Commands

This chapter describes the Program Execution commands.

CHAPTER 3 Break/Event Control Command

This chapter describes the Break/Event Control commands.

CHAPTER 4 Program Execution Analysis Commands

This chapter describes the Program Execution Analysis commands.

CHAPTER 5 Memory/Register Operation Commands

This chapter describes the Memory/Register Operation commands.

CHAPTER 6 Line Assemble and Disassemble Commands

This chapter describes the Line Assemble and Disassemble commands.

CHAPTER 7 Load and Save Commands

This chapter describes the Load and Save commands.

CHAPTER 8 Source File/Symbol Commands

This chapter describes the Source File/Symbol commands.

CHAPTER 9 Command Procedure Commands

This chapter describes the Command Procedure commands.

CHAPTER 10 Replacement Commands

This chapter describes the Replacement commands.

CHAPTER 11 Utility Commands

This chapter describes the Replacement commands.

CHAPTER 12 Control Commands

This chapter describes the Control commands.

CHAPTER 13 Built-in Variables and Functions

This chapter describes the Built-in Variables and Functions commands.

APPENDIX

These appendixes describe the Manager-Related Messages, Error Message for Debuggers, and Execution Suspension Message List.

Command Reference Notation Format

The command reference notation format is given below.

<p>■ Command name</p> <ul style="list-style-type: none"> ● Debuggers ● Format ● Description ● Example

Command name:

Name of command to be explained

Debuggers:

Usable commands depend on the debugger type. Which debugger can use the command and which debugger cannot use it are described.

[Debugger type]

Simulator : Simulator debugger

Emulator : Emulator debugger

Monitor : Monitor debugger

[Symbol for usable command]

⊙: Can use command

○: Can use command except when instruction being executed

×: Cannot use command

– : There is no debugger.

Format:

The format, parameters, and command qualifiers of the command are explained. Enter the command in this format.

Description:

The command function is explained.

When there is a description by "default n-adic number" in "Parameter", the base number used when the prefix is omitted is shown.

Example:

Command coding example. This example may differ slightly from the actual coding.

CHAPTER 1 Environment Setup Commands	13
1.1 INITIALIZE	15
1.2 EXIT	16
1.3 RESET	17
1.4 SET RESET	18
1.5 SHOW RESET	19
1.6 SET RUNMODE	20
1.7 SHOW RUNMODE	22
1.8 SET MODE (type 1)	23
1.9 SET MODE (type 2)	25
1.10 SHOW MODE	27
1.11 SET RADIX	28
1.12 SHOW RADIX	29
1.13 SET SOURCE	30
1.14 SHOW SOURCE	31
1.15 SHOW SYSTEM	32
1.16 SET MAP (type 1)	34
1.17 SET MAP (type 2)	35
1.18 SHOW MAP (type 1)	36
1.19 SHOW MAP (type 2)	37
1.20 ENABLE MAP	38
1.21 DISABLE MAP	39
1.22 CANCEL MAP (type 1)	40
1.23 CANCEL MAP (type 2)	41
1.24 ENABLE VERIFYMODE	42
1.25 DISABLE VERIFYMODE	43
1.26 SHOW VERIFYMODE	44
1.27 SET INPORT	45
1.28 SHOW INPORT	47
1.29 CANCEL INPORT	48
1.30 SET OUTPORT	49
1.31 SHOW OUTPORT	51
1.32 CANCEL OUTPORT	52
1.33 SET INTERRUPT	53
1.34 SHOW INTERRUPT	54
1.35 CANCEL INTERRUPT	55
1.36 COPY VECTOR	56
1.37 SET VECTOR	57
1.38 SHOW VECTOR	58

1.39	ENABLE WATCHDOG	59
1.40	DISABLE WATCHDOG	60
1.41	SHOW WATCHDOG	61
1.42	SET EXTERNALMEMORY	62
1.43	ENABLE EXTERNALMEMORY	63
1.44	DISABLE EXTERNALMEMORY	64
1.45	SHOW EXTERNALMEMORY	65
1.46	SET WATCH	66
1.47	CANCEL WATCH	68
1.48	SET FREQUENCY	69
1.49	SHOW FREQUENCY	71
1.50	CANCEL FREQUENCY	72
1.51	SET REALTIMEMEMORYAREA	73
1.52	SHOW REALTIMEMEMORYAREA	74
1.53	ENABLE POWERONDEBUG	75
1.54	DISABLE POWERONDEBUG	76
1.55	SHOW POWERONDEBUG	77
1.56	SYNCHRONIZE FLASH	78
1.57	ERASE FLASH	80
1.58	ENABLE DMA	81
1.59	DISABLE DMA	82
1.60	SHOW DMA	83
1.61	ENABLE CLOCKSUPERVISOR	84
1.62	DISABLE CLOCKSUPERVISOR	85
1.63	SHOW CLOCKSUPERVISOR	86
1.64	SET TIMERSCALE	87
1.65	SHOW TIMERSCALE	89
1.66	CANCEL TIMERSCALE	90

CHAPTER 2 Program Execution Commands 91

2.1	GO	92
2.2	ABORT	93
2.3	SET ABORT	94
2.4	SHOW ABORT	95
2.5	STEP	96
2.6	SET STEP	98
2.7	SHOW STEP	100
2.8	CALL	101
2.9	CLEAR CALL	103
2.10	SHOW STATUS	104

CHAPTER 3 Break/Event Control Command 107

3.1	SET BREAK (type 1)	109
3.2	SET BREAK (type 2)	112
3.3	SET BREAK (type 3)	113
3.4	SET BREAK (type 4)	116
3.5	SHOW BREAK	117

3.6	CANCEL BREAK	119
3.7	ENABLE BREAK	120
3.8	DISABLE BREAK	121
3.9	SET DATABREAK (type 1)	122
3.10	SET DATABREAK (type 2)	124
3.11	SHOW DATABREAK	126
3.12	CANCEL DATABREAK	127
3.13	ENABLE DATABREAK	128
3.14	DISABLE DATABREAK	129
3.15	SET EVENT	130
3.16	SHOW EVENT	134
3.17	CANCEL EVENT	136
3.18	ENABLE EVENT	138
3.19	DISABLE EVENT	139
3.20	SET CODEEVENT	140
3.21	SHOW CODEEVENT	142
3.22	CANCEL CODEEVENT	144
3.23	ENABLE CODEEVENT	145
3.24	DISABLE CODEEVENT	146
3.25	SET DATAEVENT	147
3.26	SHOW DATAEVENT	149
3.27	CANCEL DATAEVENT	150
3.28	ENABLE DATAEVENT	151
3.29	DISABLE DATAEVENT	152
3.30	SET SEQUENCE (type 1)	153
3.31	SET SEQUENCE (type 2)	155
3.32	SHOW SEQUENCE	157
3.33	CANCEL SEQUENCE	158
3.34	ENABLE SEQUENCE	159
3.35	DISABLE SEQUENCE	160
3.36	SET TRIGGER	161
3.37	SHOW TRIGGER	162
3.38	ENABLE ALIGNMENTBREAK	163
3.39	DISABLE ALIGNMENTBREAK	164
3.40	SHOW ALIGNMENTBREAK	165
3.41	SET BREAKCONDITION (type 1)	166
3.42	SET BREAKCONDITION (type 2)	168
3.43	SHOW BREAKCONDITION	170
 CHAPTER 4 Program Execution Analysis Commands		171
4.1	SET PERFORMANCE (type 1)	173
4.2	SET PERFORMANCE (type 2)	175
4.3	SET PERFORMANCE (type 3)	176
4.4	SHOW PERFORMANCE	177
4.5	CANCEL PERFORMANCE	182
4.6	CLEAR PERFORMANCE	183
4.7	SET COVERAGE	184

4.8	SHOW COVERAGE	185
4.9	CANCEL COVERAGE	188
4.10	CLEAR COVERAGE	189
4.11	SHOW CALLS	190
4.12	SHOW TIMER	191
4.13	CLEAR TIMER	193
4.14	SET TRACE (type 1)	194
4.15	SET TRACE (type 2)	195
4.16	SHOW TRACE (type 1)	197
4.17	SHOW TRACE (type 2)	199
4.18	CLEAR TRACE	200
4.19	ENABLE TRACE (type 1)	201
4.20	ENABLE TRACE (type 2)	202
4.21	DISABLE TRACE (type 1)	203
4.22	DISABLE TRACE (type 2)	204
4.23	SEARCH TRACE	205
4.24	SET DATATRACEAREA	207
4.25	SHOW DATATRACEAREA	209
4.26	CANCEL DATATRACEAREA	210
4.27	SHOW DETAILTRACE	211
4.28	SET TRACETRIGGER	213
4.29	SHOW TRACETRIGGER	216
4.30	CANCEL TRACETRIGGER	217
4.31	ENABLE TRACETRIGGER	218
4.32	DISABLE TRACETRIGGER	219
4.33	SET DELAY	220
4.34	SHOW DELAY	221
4.35	SET RAMCHECK	222
4.36	SHOW RAMCHECK	224
4.37	CANCEL RAMCHECK	225
4.38	ENABLE RAMCHECK	227
4.39	DISABLE RAMCHECK	229
4.40	SET SEMIHOSTING	230
4.41	SHOW SEMIHOSTING	231
CHAPTER 5 Memory/Register Operation Commands		233
5.1	EXAMINE	234
5.2	ENTER	236
5.3	SET MEMORY	238
5.4	SHOW MEMORY	240
5.5	SEARCH MEMORY	242
5.6	SET REGISTER	244
5.7	SHOW REGISTER	245
5.8	COMPARE	246
5.9	FILL	247
5.10	MOVE	248
5.11	DUMP	249

CHAPTER 6 Line Assemble and Disassemble Commands	251
6.1 ASSEMBLE	252
6.2 DISASSEMBLE	253
CHAPTER 7 Load and Save Commands	255
7.1 LOAD	256
7.2 SAVE	259
CHAPTER 8 Source File/Symbol Commands	261
8.1 LIST	262
8.2 SET PATH	264
8.3 SHOW PATH	265
8.4 SHOW SCOPE	266
8.5 UP	267
8.6 DOWN	268
CHAPTER 9 Command Procedure Commands	269
9.1 BATCH	270
9.2 QUIT	271
CHAPTER 10Replacement Commands	273
10.1 SET ALIAS	274
10.2 SHOW ALIAS	275
10.3 CANCEL ALIAS	276
10.4 SET VARIABLE	277
10.5 SHOW VARIABLE	278
10.6 CANCEL VARIABLE	279
CHAPTER 11Utility Commands	281
11.1 SET LOGGING	282
11.2 SHOW LOGGING	284
11.3 CANCEL LOGGING	285
11.4 ENABLE LOGGING	286
11.5 DISABLE LOGGING	287
11.6 PRINTF	288
11.7 SET OUTPUT	290
11.8 SHOW OUTPUT	291
CHAPTER 12Control Commands	293
12.1 IF	294
12.2 REPEAT	295
12.3 WHILE	296
12.4 BREAK	297
CHAPTER 13Built-in Variables and Functions	299
13.1 %CALL	300

13.2 %ERRNUM	301
13.3 %ENTRY	302
13.4 %STKTOP	303
13.5 %RADIX	304
13.6 %SCPADR	305
13.7 %LOADNUM	306
13.8 %BIT, %B, %H, %W, %S, %D	307
13.9 %STRGET	308
13.10 %STRSTR	309
13.11 %STRCMP	310
13.12 %STRLEN	311
13.13 %STRCAT	312
13.14 %SYMLen	313
13.15 %TOVAL	314
13.16 %TOSTR	315
13.17 %EVAL	316
13.18 %EVENTNUM	317
13.19 %TRIGGERNUM	319
13.20 %PFMEVENTNUM	320
13.21 %DEBUGFUNCTION	321
13.22 %GET_CORESTATE	322
13.23 %GET_TRACESTATE	323
13.24 %GET_EVENTSTATE	324
13.25 %GET_PERFORMANCESTATE	325
APPENDIX	327
APPENDIX A MANAGER-RELATED MESSAGES	328
APPENDIX B ERROR MESSAGE FOR DEBUGGERS	344
APPENDIX C EXECUTION SUSPENSION MESSAGES LIST	370
APPENDIX D MAJOR CHANGES	375
Index	377
Revision History	381

CHAPTER 1 Environment Setup Commands



This chapter describes the Environment Setup commands.

- 1.1 INITIALIZE
- 1.2 EXIT
- 1.3 RESET
- 1.4 SET RESET
- 1.5 SHOW RESET
- 1.6 SET RUNMODE
- 1.7 SHOW RUNMODE
- 1.8 SET MODE (type 1)
- 1.9 SET MODE (type 2)
- 1.10 SHOW MODE
- 1.11 SET RADIX
- 1.12 SHOW RADIX
- 1.13 SET SOURCE
- 1.14 SHOW SOURCE
- 1.15 SHOW SYSTEM
- 1.16 SET MAP (type 1)
- 1.17 SET MAP (type 2)
- 1.18 SHOW MAP (type 1)
- 1.19 SHOW MAP (type 2)
- 1.20 ENABLE MAP
- 1.21 DISABLE MAP
- 1.22 CANCEL MAP (type 1)
- 1.23 CANCEL MAP (type 2)
- 1.24 ENABLE VERIFYMODE
- 1.25 DISABLE VERIFYMODE
- 1.26 SHOW VERIFYMODE
- 1.27 SET INPORT
- 1.28 SHOW INPORT
- 1.29 CANCEL INPORT

- 1.30 SET OUTPORT
- 1.31 SHOW OUTPORT
- 1.32 CANCEL OUTPORT
- 1.33 SET INTERRUPT
- 1.34 SHOW INTERRUPT
- 1.35 CANCEL INTERRUPT
- 1.36 COPY VECTOR
- 1.37 SET VECTOR
- 1.38 SHOW VECTOR
- 1.39 ENABLE WATCHDOG
- 1.40 DISABLE WATCHDOG
- 1.41 SHOW WATCHDOG
- 1.42 SET EXTERNALMEMORY
- 1.43 ENABLE EXTERNALMEMORY
- 1.44 DISABLE EXTERNALMEMORY
- 1.45 SHOW EXTERNALMEMORY
- 1.46 SET WATCH
- 1.47 CANCEL WATCH
- 1.48 SET FREQUENCY
- 1.49 SHOW FREQUENCY
- 1.50 CANCEL FREQUENCY
- 1.51 SET REALTIMEMEMORYAREA
- 1.52 SHOW REALTIMEMEMORYAREA
- 1.53 ENABLE POWERONDEBUG
- 1.54 DISABLE POWERONDEBUG
- 1.55 SHOW POWERONDEBUG
- 1.56 SYNCHRONIZE FLASH
- 1.57 ERASE FLASH
- 1.58 ENABLE DMA
- 1.59 DISABLE DMA
- 1.60 SHOW DMA
- 1.61 ENABLE CLOCKSUPERVISOR
- 1.62 DISABLE CLOCKSUPERVISOR
- 1.63 SHOW CLOCKSUPERVISOR
- 1.64 SET TIMERSCALE
- 1.65 SHOW TIMERSCALE
- 1.66 CANCEL TIMERSCALE

1.1 INITIALIZE

The INITIALIZE command initializes the debugger.

■ INITIALIZE

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

- Format

INITIALIZE

- Description

The INITIALIZE command initializes the debugger.

This initialization nullifies all settings other than macro, alias and debug variable.

- Example

```
>INITIALIZE
```

1.2 EXIT

The EXIT command terminates the debugger.

■ EXIT

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

EXIT

- Command qualifiers

/ALL

Terminates SOFTUNE Workbench after terminating the debug session.

- Description

The EXIT command terminates the debug session.

- Example

[When terminating the debug session]

```
>EXIT
```

[When terminating the SOFTUNE Workbench after terminating the debug session]

```
>EXIT /ALL
```


1.3 RESET

The RESET command inputs the reset signal to the MCU.

■ RESET

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		×

- Format

RESET

Command qualifiers

- Reset level

/CHIP (default at start-up)

Chip reset is issued.

This qualifier can be applied/specified only to emulator debugger (MB2100-01).

/LOWLEVEL

The low level reset is issued.

This qualifier can be applied/specified only to emulator debugger (MB2100-01).

- Description

The RESET command inputs the reset signal to the MCU.

- Example

```
>RESET
```

```
[Emulator Debugger (MB2100-01)]
```

```
> RESET /LOWLEVEL
```

Note:

[Emulator Debugger (MB2100-01)]

- When the command qualifiers is omitted, the reset level set by SET RESET is applied.
 - For details of the chip reset and the low level reset, see the hardware manual.
-

1.4 SET RESET

The **SET RESET** command sets the operation that the reset level is not specified by the **RESET** command.

■ SET RESET

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SET RESET

Command qualifiers

- Reset level

/CHIP (Default when omitted)

/CHIP is set that chip reset is issued by default when the RESET command is used.

/LOWLEVEL

/LOWLEVEL is set that Low level reset is issued by default when the RESET command is used.

- Description

The SET RESET command sets the operation that the reset level is not specified by the RESET command.

- Example

```
>SET RESET /LOWLEVEL
```

1.5 SHOW RESET

The **SHOW RESET** command displays the reset level specified by **SET RESET** command.

■ SHOW RESET

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW RESET

- Description

The SHOW RESET command displays the reset level specified by SET RESET command.

- Example

```
>SET RESET/CHIP
>SHOW RESET
reset level : chip
>SET RESET/LOWLEVEL
>SHOW RESET
reset level : lowlevel
```

1.6 SET RUNMODE

The **SET RUNMODE** command sets an MCU operation mode.

■ SET RUNMODE

● Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

● Format

SET RUNMODE

Command qualifiers

• Setting operation mode

/TRACE (default at start-up)

Sets operation mode to full-trace mode.

/REALTIME

Sets operation mode to real-time mode.

/INTERNALTRACE

Sets operation mode to internal-trace mode.

/EXTERNALTRACE

Sets operation mode to external-trace mode.

● Description

The SET RUNMODE command sets an MCU operation mode.

The following four operation modes are available:

• Full-trace mode

In the full-trace mode, the trace function can be freely used but the program cannot be executed in real time.

• Real-time mode

In the real time mode, the program can be executed in real time.

However, in the following cases, trace data may be lost because trace information output is insufficient:

- Program branches three times or more within 11 cycles.

• Internal-trace mode

In the internal-trace mode, a trace data is stored into the trace dedicated memory built-in the chip. The program is executed in real time, but this mode can be specified only using the evaluation chip integrated with the function.

• External-trace mode

In the external-trace mode, a trace data is stored into the trace dedicated memory installed on the adapter board. The program is executed in real time, but this mode can be specified only using the evaluation chip integrated with the function.

● Example

```
>SET RUNMODE/TRACE
```

Note:

The command qualifiers that can be specified vary depending on the emulator used or its connection and the DSU.

For details, refer to "2.2.1.1 MCU Operation Mode" or "2.3.1.2 MCU Operation Mode" of "SOFTUNE WORKBENCH USER'S MANUAL".

1.7 SHOW RUNMODE

The **SHOW RUNMODE** command displays the MCU operation mode set by the **SET RUNMODE** command.

■ SHOW RUNMODE

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	×
Monitor		×

- Format

SHOW RUNMODE

- Description

The **SHOW RUNMODE** command displays the MCU operation mode set by the **SET RUNMODE** command.

- Example

```
>SHOW RUNMODE  
run mode : TRACE
```

1.8 SET MODE (type 1)

The **SET MODE** command sets the event mode and debug function.

■ SET MODE (type 1)

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

<Format 1>

SET MODE

<Format 2>

SET MODE debug-function-number

Parameter

- Format 2

debug-function-number

Specifies the debug function to be enabled.

Specifies one of the following numbers:

0: Real Time Memory function

1: RAM Checker function

Command qualifiers

- Format 1

Setting of event mode

/NORMAL (default at start-up)

Sets event mode to TRACE mode.

/PERFORMANCE

Sets event mode to PERFORMANCE mode.

- Format 2

Setting of debug function

/CONFIG

Sets debug function.

- Description

The event mode or the debug feature is set. The setting of two modes simultaneously is also possible.

- Format 1

The SET MODE command sets the event mode as follows:

- TRACE mode

The event function is used for trace control. Command setting related to , datawatch break, tracetrigger, and sequence is enabled.

- PERFORMANCE mode

The event function is used for measuring performance. Command setting related to performance is enabled.

The commands related to event can be used in all modes, each of which has different values. If a mode is changed, the value will return to the value previously set in the mode.

A mode change will also clear the trace and performance buffers.

The default is "/NORMAL".

- Format 2

The SET MODE command sets the debug function as follows:

- Real Time Memory mode

This mode enables real-time monitor function. This mode enables to display data for a "256 bytes × 2" area in the real-time memory window without breaking the MCU at all during program execution.

- RAM Checker mode

This mode enables the RAM Checker function. This mode allows you to record the access history of the monitoring address in the log file.

Selectable debug function varies with the emulator or its connection configuration.

Only selected functions can be used. In addition, changing a mode clears all the trace and performance data.

At startup time, debug mode is set to Real-time Memory mode.

- Example

<Format 1>

```
>SET MODE /PERFORMANCE
```

<Format 2>

```
>SET MODE /CONFIG 1
```

Note:

This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.8 Measuring Performance" and "2.3.1.6 Debug mode" of "SOFTUNE WORKBENCH USER'S MANUAL".

1.9 SET MODE (type 2)

The **SET MODE** command sets the execution time mode and the pass count mode.

■ SET MODE (type 2)

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SET MODE

Command qualifiers

- Select execution time mode
/TIME (default at start-up)
Changes to the time measurement mode.
/PERFORMANCE
Changes to PERFORMANCE mode.
- Select the pass count mode
/SEQUENCE
Changes to SEQUENCE mode.
/PASSCOUNTBREAK (default at start-up)
Changes to the pass count break mode.

- Description

The execution time mode or the pass count mode is set. The setting of two modes simultaneously is also possible.

- Execution time mode
 - Time measurement mode
Enables the time measurement function for measure between the user program execution and the break point.
 - Performance mode
Enables the performance function for measure the pass time between 2 points of the event.
- Pass count mode
 - Sequence mode
The event to which the pass count can be set is used as the sequence function.
 - Pass count break mode
The event to which the pass count can be set is used as the pass count break function.

- Example

```
>SET MODE /PERFORMANCE /PASSCOUNTBREAK
```

Note:

When the pass count mode is modified, all events under setting are deleted.

1.10 SHOW MODE

The **SHOW MODE** command displays the status of each debug function.

■ SHOW MODE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW MODE

- Description

[Emulator Debugger (MB2198)]

The SHOW MODE command displays the setting state of the event mode and debug function.

[Emulator Debugger (MB2100-01)]

The SHOW MODE command displays the information of the execution time mode and pass count mode.

- Example

[Emulator Debugger (MB2198)]

```
>SHOW MODE
event mode      :normal
debug function  :
    *0:Real Time Memory
    1:RAM Checker
```

[Emulator Debugger (MB2100-01)]

```
>SHOW MODE
timer : performance
passcount mode : passcount break
>SHOW MODE
timer : time measurement
passcount mode : sequence
```

Note:

[Emulator Debugger (MB2198)]

This command cannot be used except the FR60Lite or FR80S. For details, refer to "2.3.8 Measuring Performance" and "2.3.1.6 Debug mode" of "SOFTUNE WORKBENCH USER'S MANUAL".

1.11 SET RADIX

The **SET RADIX** command sets default base number.

■ SET RADIX

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SET RADIX

Command qualifiers

/BINARY

Sets default base number to binary number.

/OCTAL

Sets default base number to octal number.

/DECIMAL

Sets default base number to decimal number.

/HEXADECIMAL (default)

Sets default base number to hexadecimal number.

- Description

The SET RADIX command sets default base number.

- Example

```
>SET RADIX/HEXADECIMAL
```

1.12 SHOW RADIX

The **SHOW RADIX** command displays the current base number.

■ SHOW RADIX

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SHOW RADIX

- Description

The SHOW RADIX command displays the current base number.

- Example

```
>SHOW RADIX
default radix : hexadecimal
```

1.13 SET SOURCE

The **SET SOURCE** command sets the source line display mode.

■ SET SOURCE

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SET SOURCE

Command qualifiers

- Source line display mode

/DISPLAY (default at start-up)

Sets mode in which source lines displayed.

/NODISPLAY

Sets mode in which source lines not displayed.

- Description

When the disassemble list is displayed, the SET SOURCE command sets whether to display with the source lines.

When the debugger is started, the mode in which source lines are displayed is set.

- Example

```
>SET SOURCE/DISPLAY
```

1.14 SHOW SOURCE

The **SHOW SOURCE** command displays the source line display mode set by the **SET SOURCE** command.

■ SHOW SOURCE

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SHOW SOURCE

- Description

The SHOW SOURCE command displays the source line display mode set by the SET SOURCE command.

- Example

```
>SHOW SOURCE
source mode : display
```

1.15 SHOW SYSTEM

The **SHOW SYSTEM** command displays system information.

■ SHOW SYSTEM

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		○

- Format

SHOW SYSTEM

- Description

The SHOW SOURCE command displays system information.

What is displayed varies depending on the debugger type.

For details, refer to the following sections of "SOFTUNE WORKBENCH USER'S MANUAL".

Simulator debugger : "2.1.12 Checking Debugger Status"

Emulator Debugger (MB2197) : "2.2.8 Checking Debugger Status"

Emulator Debugger (MB2198) : "2.3.13 Checking Debugger Status"

Emulator Debugger (MB2100-01) : "2.4.11 Checking Debugger Status"

- Example

[Emulator Debugger (MB2198)]

```
>SHOW SYSTEM
FR Family Softune Workbench V60L10
Debugger type           = Emulator Debugger
MCU type                = MB91647
VCpu dll name           = C:\Softune\Bin\wv911e3.dll
DSU type                = DSU4
Common version          = V02L09
Monitor version         = V03L03
Configuration board ID  = 790-01
Configuration board version = V01
MCU frequency           = 6.252 MHz
Communication device     = USB
```


[Emulator Debugger (MB2100-01)]

```
>SHOW SYSTEM
FR Family Softune Workbench V60L10
Debugger type           = Emulator Debugger
MCU type                = MB91F777
VCpu dll name           = C:\Softune\Bin\wv911e3.dll
VCpu dll version        = VxxLxx
SiDRVo dll version      = VxxLxx
DSU type                = OCDU
Adapter version         = V--L-- (V01L01*)
FPGA version            = V01L01
Maker ID                = 0x0400
CPU family ID           = 0x0200
DSU type ID             = 0x1
DSU version ID          = 0x1
Device ID               = 0x0010
Device version ID       = 0x1
OSC clock                = xxxxxxxx (xxxxxxxx) Hz
PLL clock                = --- (---)
Clock mode               = RC
Communication mode       = Normal (High)
Communication type       = 0xx
Communication device     = USB
```

1.16 SET MAP (type 1)

The SET MAP command sets a memory space area type and access attribute.

■ SET MAP (type 1)

● Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

● Format

SET MAP {address | address-range}

Parameters

address (address formula)

Specifies the memory address where access attribute to be set.

address-range (address formula)

Specifies the memory area where access attribute to be set.

Command qualifiers

• Specifying access attribute

/READ

Enables data read access.

/WRITE

Enables data write access.

/CODE

Enables code read access.

If command qualifier is omitted, /READ/WRITE is set.

● Description

The SET MAP command sets a memory space area type and access attribute.

Up to 31 memory areas can be set. (Set in a 1-byte boundary for the normal speed simulator debugger and set in a 2-byte boundary for the high-speed simulator debugger.)

When the load module file is loaded by the LOAD command, appropriate access attributes are automatically set according to the file information.

Up to 128 Mbytes can be specified in the total of each area.

● Example

```
>SET MAP/READ/WRITE 1000..1FFF
```

1.17 SET MAP (type 2)

The **SET MAP** command sets the memory area which accessing is prohibited.

■ SET MAP (type 2)

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	○
Monitor		○

- Format

SET MAP/INACCESSIBLE {address | address-range}

Parameters

address (address formula)

Specifies the memory address.

Address-range (address formula)

Specifies the memory area.

Command qualifiers

- Specifying access attribute

/INACCESSIBLE

Sets the memory area which accessing is prohibited.

- Description

Sets the memory area which accessing is prohibited.

A maximum of 16 regions can be set (settable by increments of one byte).

- Example

```
>SET MAP/INACCESSIBLE 1000..1FFF
```

1.18 SHOW MAP (type 1)

The **SHOW MAP** command displays the set memory space access attributes.

■ SHOW MAP (type 1)

- Debugger

Simulator		⊙
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

SHOW MAP

- Description

The **SHOW MAP** command displays the set memory space access attributes.

- Example

```
>SHOW MAP
      address          attribute
00000000 .. 000011FF  read write
00001200 .. FFFFFFFF  undefined
FFFF0000 .. FFFFFFFF  read code
```

1.19 SHOW MAP (type 2)

The **SHOW MAP** command displays an inaccessible area.

■ SHOW MAP (type 2)

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SHOW MAP /INACCESSIBLE

Command qualifiers

- Specifying access attribute

/INACCESSIBLE

Displays an inaccessible memory area.

- Description

The SHOW MAP command displays an inaccessible memory area of a specified number.

- Example

```
>SHOW MAP/INACCESSIBLE
no.  en/dis  address
1    enable  00000000..0000FFFF
2    enable  00044000..0007FFFF
```

1.20 ENABLE MAP

The **ENABLE MAP** command enables an inaccessible area of specified number.

■ ENABLE MAP

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	○
Monitor		○

- Format

ENABLE MAP/INACCESSIBLE [map number]

Parameter

map number

Specifies a number of inaccessible memory area.

Command qualifiers

/INACCESSIBLE

Enables an inaccessible memory area.

- Description

Enables an inaccessible area of a specified number.

- Example

```
>ENABLE MAP/INACCESSIBLE 2
```

1.21 DISABLE MAP

The **DISABLE MAP** command disables an inaccessible area of a specified number.

■ DISABLE MAP

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	○
Monitor		○

- Format

DISABLE MAP/INACCESSIBLE [map number]

Parameter

map number

Specifies a number of inaccessible area.

Command qualifiers

/INACCESSIBLE

Disables an inaccessible memory area.

- Description

Disables an inaccessible area of a specified number.

- Example

```
>DISABLE MAP/INACCESSIBLE 1
```

1.22 CANCEL MAP (type 1)

The **CANCEL MAP** command assigns the undefined attribute to the specified address area.

■ CANCEL MAP (type 1)

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

CANCEL MAP [address | address-range]

Parameters

address (address formula)

Specifies the address where undefined attribute to be assigned.

address-range (address formula)

Specifies the address range where undefined attribute to be assigned.

Command qualifier

/ALL

Assigns undefined attribute to all map settings.

- Description

The **CANCEL MAP** command assigns the undefined attribute to the specified address area.

- Example

```
>CANCEL MAP/ALL
```


1.23 CANCEL MAP (type 2)

The **CANCEL MAP** command deletes the specified inaccessible address-range.

■ CANCEL MAP (type 2)

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	○
Monitor		○

- Format

CANCEL MAP [address | address-range]

Parameters

address (address formula)

Specifies the memory address.

address-range (address formula)

Specifies the memory area.

Command qualifiers

/INACCESSIBLE

Deletes an inaccessible memory area.

/ALL

Deletes all inaccessible area.

- Description

The CANCEL MAP command deletes the specified inaccessible address-range.

- Example

```
>CANCEL MAP/INACCESSIBLE 00044000..0007FFFF
```

1.24 ENABLE VERIFYPYMODE

The **ENABLE VERIFYPYMODE** command enables the verify mode used when memory is written by a command.

■ ENABLE VERIFYPYMODE

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	○
Monitor		×

- Format

ENABLE VERIFYPYMODE

- Description

The **ENABLE VERIFYPYMODE** command enables the verify mode used when memory is written by a command.

The verify mode is enabled when the debugger is started.

- Example

```
>ENABLE VERIFYPYMODE
```

1.25 DISABLE VERIFYPAGE

The **DISABLE VERIFYPAGE** command disables the verify mode used when memory is written by a command.

■ DISABLE VERIFYPAGE

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	○
Monitor		×

- Format

DISABLE VERIFYPAGE

- Description

The **DISABLE VERIFYPAGE** command disables the verify mode used when memory is written by a command.

The verify mode is enabled when the debugger is started.

- Example

```
>DISABLE VERIFYPAGE
```

1.26 SHOW VERIFYMODE

The **SHOW VERIFYMODE** command displays the status of the verify mode.

■ SHOW VERIFYMODE

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW VERIFYMODE

- Description

The SHOW VERIFYMODE command displays the state of the verify mode, which shows the enabled or disabled verify operation at the memory writing with a command.

- Example

```
>SHOW VERIFYMODE  
verify mode : enable
```

1.27 SET INPORT

The **SET INPORT** command specifies data input to a specified port.

■ SET INPORT

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

SET INPORT port-address, mask-data, data-input-source [, cycle-count]

Parameters

port-address

Specifies the port address.

mask-data

Specifies the mask data.

Only the data with the mask data bit of "1" can be valid as a port data.

data-input-source

Specifies the data-input-source (file name, input terminal) when program reads data from input port or when count of program instruction execution cycles exceeds specified cycle count.

The input terminal is specified as \$TERMINAL.

\$TERMINAL cannot be used as file name.

cycle-count (decimal number)

Specifies the count of program instruction execution cycles (D'1 to D'4294967295).

Command qualifiers

- Specifying access size

/BYTE (default when omitted)

Specifies a port access when specified address is accessed through 1 byte access.

/HALFWORD

Specifies a port access when specified address is accessed through 2-byte access.

/WORD

Specifies a port access when specified address is accessed through 4-byte access.

/ASCII

Uses character codes of input data as input values.

When /ASCII is specified, the access size is always /BYTE.

● Description

The SET INPORT command specifies data input to a port each time the program reads data from the specified port or each time the count of program instruction execution cycles exceeds the specified cycle count.

When cycle-count is not specified, data is read from the specified data input source each time the program reads data from the input port.

When cycle-count is specified, port values are updated per fixed cycle irrespective of port access.

If data-input-source is a file, data input processing returns to the beginning of the file when the last data is entered.

If data-input-source is a input terminal (\$TERMINAL), the dialog box for data input request is displayed when the set port is read-accessed. When this dialog box appears, enter the input data.

Up to 4096 port addresses can be simulated.

● Example

```
>SET INPORT 0, 1F, INBUF0. DAT
>SHOW INPORT
```

address	bitpattern	size	cycle	input
00000000	0000001F	byte	-----	INBUF0. DAT
0000004F	0000000F	ascii	-----	\$terminal

1.28 SHOW INPORT

The **SHOW INPORT** command displays the data set by the **SET INPORT** command.

■ SHOW INPORT

- Debugger

Simulator		⊙
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

SHOW INPORT

- Description

The **SHOW INPORT** command displays the data set by the **SET INPORT** command.

- Example

```
>SHOW INPORT
address  bitpattern    size   cycle      input
000000FF 000000FF    byte  -----   $terminal
0000004F 0000000F    ascii -----   $terminal
```

1.29 CANCEL INPORT

The **CANCEL INPORT** command cancels simulation of specified port address.

■ CANCEL INPORT

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

CANCEL INPORT [port-address [, ...]]

Parameter

port-address

Specifies the port address.

Command qualifier

/ALL

Cancels all data set by SET INPORT command.

- Description

The CANCEL INPORT command cancels the port simulation of specified port address.

- Example

```
>CANCEL INPORT/ALL
```


1.30 SET OUTPORT

The **SET OUTPORT** command specifies that data is to be stored in the specified data-output-destination each time the program writes data to the specified port.

■ SET OUTPORT

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

SET OUTPORT port-address, mask-data, data-output-destination

Parameters

port-address

Specifies the port address.

mask-data

Specifies the mask data.

Only the data with the mask data bit of "1" can be valid as a port data.

data-output-destination

Specifies the data-output-destination (file name, output terminal) where data written to output port by program to be stored.

The output terminal is specified as \$TERMINAL.

\$TERMINAL cannot be used as the file name.

The same file name cannot be used.

Command qualifiers

- Specifying access attribute

/BYTE (default when omitted)

Specifies a port access when specified address is accessed through 1 byte access.

/HALFWORD

Specifies a port access when specified address is accessed through 2-byte access.

/WORD

Specifies a port access when specified address is accessed through 4-byte access.

/ASCII

When data-output-destination is \$TERMINAL, the debugger converts the data output to the port (regarded as ASCII codes) to characters and displays them on the screen.

When data-output-destination is a file, the debugger outputs binary codes as they are.

When /ASCII is specified, the access size is always /BYTE.

- Description

The SET OUTPORT command specifies that data is to be stored in the specified data-output-destination each time the program writes data to the specified port.

If the disk becomes full when data-output-destination is a file, the debugger displays the error message and does not store the subsequent data in the file.

When data-output-destination is an output terminal (\$TERMINAL), the debugger displays the data output to the port on the screen in hexadecimal notation.

However, when command qualifier /ASCII is specified, the debugger displays the output data in characters, because the debugger treats the output data as character codes.

Up to 4096 port addresses can be simulated.

- Example

```
>SET OUTPORT 0, 3F, OU1. DAT
>SHOW OUTPORT
address    bitpattern    size    output
00000000  0000003F      byte    OU1. DAT
00000040  000000FF      ascii    $TERMINAL
```

1.31 SHOW OUTPUT

The **SHOW OUTPUT** command displays the data set by the **SET OUTPUT** command.

■ SHOW OUTPUT

- Debugger

Simulator		⊙
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

SHOW OUTPUT

- Description

The **SHOW OUTPUT** command displays the data set by the **SET OUTPUT** command.

- Example

```
>SHOW OUTPUT
address  bitpattern  size  output
00000000 0000003F   byte  OU1. DAT
00000040 000000FF  ascii  $TERMINAL
```

1.32 CANCEL OUTPORT

The **CANCEL OUTPORT** command cancels the simulation of the specified port address.

■ CANCEL OUTPORT

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

CANCEL OUTPORT [port-address [, . . .]]

Parameter

port-address

Specifies the port address.

Command qualifier

/ALL

Cancels all settings by SET OUTPORT command.

- Description

The CANCEL OUTPORT command cancels the port simulation of the specified port address.

- Example

```
>CANCEL OUTPORT/ALL
```

1.33 SET INTERRUPT

The **SET INTERRUPT** command sets the conditions to generate interrupts.

■ SET INTERRUPT

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

SET INTERRUPT interrupt-number, cycle-count

Parameters

interrupt-number (decimal number)

Specifies the interrupt vector number.

For information about how many interrupts can be used, refer to the Hardware Manual for the product type you are using.

cycle-count (decimal number)

Specifies the count of program instruction execution cycles (D'1 to D'4294967295).

Command qualifier

/INTERVAL

Specifies cyclic generation of interrupts.

- Description

When the program is executed for the specified count of instruction execution cycles or more, the SET INTERRUPT command causes the specified interrupt and cancels the interrupt-generation condition.

When /INTERVAL is specified, the specified interrupt is generated per specified count of instruction execution cycles during program execution.

The interrupt-generation condition is valid until it is canceled by the CANCEL INTERRUPT command.

- Example

```
>SET INTERRUPT 4, 12367
```

1.34 SHOW INTERRUPT

The **SHOW INTERRUPT** command displays the interrupt vector number, cycle count (decimal number), and **/INTERVAL** set by the **SET INTERRUPT** command in this order.

■ SHOW INTERRUPT

- Debugger

Simulator		⊙
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

SHOW INTERRUPT

- Description

The **SHOW INTERRUPT** command displays the interrupt vector number, cycle count (decimal number), and **/INTERVAL** set by the **SET INTERRUPT** command in this order.

/INTERVAL specifies any of the following:

shot: Specifies single interrupt generation (without **INTERVAL**).

interval: Specifies repetitive interrupt generation (with **INTERVAL**).

- Example

```
>SHOW INTERRUPT
no    cycle      kind
18    1258       shot
22    9823       interval
```

1.35 CANCEL INTERRUPT

The **CANCEL INTERRUPT** command cancels all the interrupt-generation conditions set by the **SET INTERRUPT** command.

■ CANCEL INTERRUPT

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

CANCEL INTERRUPT [interrupt-number [, . . .]]

Parameter

interrupt-number (decimal number)

Specifies the interrupt vector number.

Command qualifier

/ALL

Cancels all interrupt-generation conditions set by SET INTERRUPT command.

- Description

The CANCEL INTERRUPT command cancels all the interrupt-generation conditions set by the SET INTERRUPT command.

- Example

```
>CANCEL INTERRUPT/ALL
```

1.36 COPY VECTOR

The **COPY VECTOR** command copies the value in the initial area of the vector table to the specified area.

■ COPY VECTOR

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

- Format

COPY VECTOR [vector-number] [, TBR-value]

Parameters

vector-number (decimal number)

Specifies the number of vector to be copied.

TBR-value

Specifies the area addresses of the vector table to be copied.

If address H'FFFFFFC01 or higher is specified, an error occurs.

If TBR-value specifying is omitted, the TBR value of the current register is used.

Command qualifiers

/ALL (default when omitted)

Copies all EIT vector values.

/MINIMUM

Copies minimum of EIT vector values used by debugger.

- Description

The **COPY VECTOR** command copies the value in the initial area of the vector table to the specified area.

The **COPY VECTOR** command copies the value in the initial area (area specified by the TBR value of 0xffc00) of the vector table to the area specified by TBR-value.

When the vector table location is changed by rewriting the TBR register, the value in the initial area of the vector table must be copied.

- Example

```
>COPY VECTOR 11, 100C00
```

```
>COPY VECTOR /MINIMUM
```


1.37 SET VECTOR

The **SET VECTOR** command sets the address value of the vector number set in the specified area.

■ SET VECTOR

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

- Format

SET VECTOR vector-number, address-value

Parameters

vector-number

Specifies the number of vector to be set.

address-value

The value is the starting address value of routine corresponding to specified vector number.

- Description

The SET VECTOR command sets the address value of the vector number set in the specified area.

- Example

```
>SET VECTOR 11, 0FF100
>SHOW VECTOR 11..11
VectorNo.      Address      Symbol
11             00FF100
```

1.38 SHOW VECTOR

The **SHOW VECTOR** command displays vector number data.

■ SHOW VECTOR

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		◯

- Format

SHOW VECTOR [vector-number-range]

Parameter

vector-number-range

Specifies the range of vector numbers to be displayed.

The range is specified in "[starting-number..ending-number]" format.

- Description

The **SHOW VECTOR** command displays vector number data.

If vector-number-range specifying is omitted, vector number display is started from the next vector number.

- Example

```
>SHOW VECTOR 6..8
```

VectorNo.	Address	Symbol	Factor
6	00000000		System Reserved
7	FF201000	co_1000	Co-processor Absence
8	FF110000	CO_ERROR	Co-processor error

1.39 ENABLE WATCHDOG

The **ENABLE WATCHDOG** command enables the watchdog timer.

■ ENABLE WATCHDOG

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	○
Monitor		×

- Format

ENABLE WATCHDOG

- Description

The **ENABLE WATCHDOG** command enables the watchdog timer.

- Example

```
>ENABLE WATCHDOG
```

Note:

[Emulator Debugger (MB2197 or MB2198)]

Only the DSU3/DSU4 chip can specify this command.

1.40 DISABLE WATCHDOG

The **DISABLE WATCHDOG** command disables the watchdog timer.

■ DISABLE WATCHDOG

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	○
Monitor		×

- Format

DISABLE WATCHDOG

- Description

The **DISABLE WATCHDOG** command disables the watchdog timer.

- Example

```
>DISABLE WATCHDOG
```

Note:

[Emulator Debugger (MB2197 or MB2198)]

Only the DSU3/DSU4 chip can specify this command.

1.41 SHOW WATCHDOG

The **SHOW WATCHDOG** command displays the enabled/disabled state of the watchdog timer.

■ SHOW WATCHDOG

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW WATCHDOG

- Description

The SHOW WATCHDOG command displays the enabled/disabled state of the watchdog timer.

- Example

```
>SHOW WATCHDOG
watchdog : enable
```

Note:

[Emulator Debugger (MB2197 or MB2198)]
Only the DSU3/DSU4 chip can specify this command.

1.42 SET EXTERNALMEMORY

The **SET EXTERNALMEMORY** command specifies an alternative memory area (in the main memory) for the external ROM or RAM by setting the chip select number.

■ SET EXTERNALMEMORY

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

SET EXTERNALMEMORY {/ROM|/RAM} chip-select-number

Parameter

chip-select-number (default: hexadecimal)

Specify the chip select number of the memory area to which acting memory is allocated.

Command qualifiers

/ROM (default when omitted)

The specified chip select area is used as the ROM area.

/RAM

The specified chip select area is used as the RAM area.

- Description

Only one area can be allocated as acting memory to external memory.

The chip select number is specified to set the area. For details of the chip select, refer to the Hardware Manual for the product type you are using.

If /ROM is specified as a command qualifier, write from the user program is inhibited, which does not become the cause of a break.

If command qualifiers are omitted, /ROM is specified.

This function can be used only in evaluation chips with dedicated pins used only for external memory. The emulator automatically determines whether the function can be used or not.

Acting memory is mounted to the adapter unit. For the hardware setting and memory size, refer to the hardware manual for your adapter unit.

- Example

```
> SET EXTERNALMEMORY/ROM 1
> SHOW EXTERNALMEMORY
enable   CS1 : ROM
```

1.43 ENABLE EXTERNALMEMORY

The **ENABLE EXTERNALMEMORY** command enables the external memory emulation function.

■ ENABLE EXTERNALMEMORY

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

ENABLE EXTERNALMEMORY

- Description

The **ENABLE EXTERNALMEMORY** command is used to enable the disabled external memory emulation function.

- Example

```
> ENABLE EXTERNALMEMORY
> SHOW EXTERNALMEMORY
enable    CS1 : ROM
```

1.44 DISABLE EXTERNALMEMORY

The **DISABLE EXTERNALMEMORY** command disables the external memory emulation function.

■ DISABLE EXTERNALMEMORY

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

DISABLE EXTERNALMEMORY

- Description

The **DISABLE EXTERNALMEMORY** command is used to disable the enabled external memory emulation function.

- Example

```
> DISABLE EXTERNALMEMORY
> SHOW EXTERNALMEMORY
disable    CS1 : ROM
```


1.45 SHOW EXTERNALMEMORY

The **SHOW EXTERNALMEMORY** command displays the setting state of the external memory emulation function.

■ SHOW EXTERNALMEMORY

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	×
Monitor		×

- Format

SHOW EXTERNALMEMORY

- Description

The **SHOW EXTERNALMEMORY** command is used to display the setting state of the external memory emulation function.

- Example

```
>SHOW EXTERNALMEMORY
CS1 : ROM
disable CS1 : ROM
```

1.46 SET WATCH

The **SET WATCH** command sets specified variables in the watch window.

■ SET WATCH

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

● Format

SET WATCH variable [, watch-window-number]

Parameters

- variable
Specifies variables or expressions to be set in the watch window.
- watch-window-number
Specifies the number (1 to 4) of the watch window to which variables are added. When omitted, Watch Window 1 will be specified.

Command qualifier

- Base Number
 - /BINARY
Specifies that variable values will be displayed in binary.
 - /OCTAL
Specifies that variable values will be displayed in octal.
 - /DECIMAL
Specifies that variable values will be displayed in decimal.
 - /HEXADECIMAL
Specifies that variable values will be displayed in hexadecimal.
- Watch Mode
 - /AUTOMATIC
Interprets variables in the watch mode set in the debug environment.
If the setting in the debug environment is changed, the watch mode for variables is changed as well.
 - /C
Interprets variables as C/C++ language expressions.
 - /ASSEMBLER
Interprets variables as assembler expressions.

- Data Size

/BYTE

Specifies that display will be provided with 1-byte length in assembler mode.

/HALFWORD

Specifies that display will be provided with 2-byte length in assembler mode.

/WORD

Specifies that display will be provided with 4-byte length in assembler mode.

/DWORD

Specifies that display will be provided with 8-byte length in assembler mode.

/SINGLE

Specifies that display will be provided with single-precision floating-point number in assembler mode.

/DOUBLE

Specifies that display will be provided with double-precision floating-point number in assembler mode.

- Individual monitoring setting

/MONITORING (Default when omitted)

Sets the individual monitoring settings for variables to ON.

/NOMONITORING

Sets the individual monitoring settings for variables to OFF.

- Description

Sets variables in the specified watch window. If variables already set area specified, two or more variables of the same name are set.

If the command qualifier for a base number, watch mode, or data size is omitted, the setting specified in the debug environment is effective.

The specified data size is valid only when the setting of /ASSEMBLER is specified.

For the setting of a watch point, the previous setting is restored when the Debugger is started. If a watch point is set in the batch file once delete all the watch points by CANCEL WATCH/ALL before the Debugger is started.

- Example

```
>SET WATCH strsym.a,1
```

```
>SET WATCH/HEXADECIMAL/ASSEMBLER/WORD/NOMONITORING LABEL1,1
```

1.47 CANCEL WATCH

The **CANCEL WATCH** command deletes specified variables from the watch window.

■ CANCEL WATCH

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

● Format

CANCEL WATCH variable [, watch-window-number]

CANCEL WATCH/ALL [,watch-window-number]

Parameters

• Variable

Specifies variables or expressions to be deleted from the watch window.

• watch-window-number

Specifies the number (1 to 4) of the watch window from which variables are deleted.
When omitted, Watch Window 1 is specified.

Command qualifier

/ALL

Deletes all watch points from the specified window.

● Description

Deletes variables from the specified watch window.

If two or more variables of the same name exist, only the variables that first appear will be deleted.

● Example

```
>CANCEL WATCH flag,1
```

```
>CANCEL WATCH/ALL 2
```

1.48 SET FREQUENCY

The **SET FREQUENCY** command sets the maximum operating frequency for CPU or standard clock frequency for high speed communication.

■ SET FREQUENCY

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	○
Monitor		×

- Format

[Emulator Debugger (MB2198)]

SET FREQUENCY /MAX Maximum operating frequency

[Emulator Debugger (MB2100-01)]

SET FREQUENCY /MAX Standard clock frequency for high speed communication

Parameters

Maximum operating frequency (decimal number)

Sets the maximum operating frequency (MHz) for CPU. This setting is valid only on the emulator debugger (MB2198).

This parameter sets the maximum operation frequency, and does not change an actual operation frequency.

Standard clock frequency for high speed communication (decimal number)

Sets the standard clock frequency for high speed communication. This setting is valid only on the emulator debugger (MB2100-01).

The communication speed is automatically set to high speed during debugging based on the configured frequency.

Command qualifiers

/MAX (Cannot be omitted)

Sets the following values.

Emulator Debugger (MB2198): The maximum operating frequency of the CPU

Emulator Debugger (MB2100-01): The reference clock frequency for high speed communication

/RESET

The set maximum operating frequency is returned to the default value of CPU information file.

This setting is valid only on the emulator debugger (MB2198).

Qualifier specifying the units

The following command qualifiers are valid only for emulator debugger (MB2100-01).

/MHZ(Default when omitted)

Handles input parameter frequencies in MHz.

/KHZ

Handles input parameter frequencies in kHz.

/HZ

Handles input parameter frequencies in Hz.

- Description

[Emulator Debugger (MB2198)]

The SET FREQUENCY command sets the maximum operating frequency for CPU.

The best auto wait value is calculated and set automatically based on the set maximum operating frequency.

As a result, the best response speed is set at debugging.

[Emulator Debugger (MB2100-01)]

Set the standard clock frequency for high speed communication.

The communication speed is automatically set to high speed during debugging based on the configured frequency.

- Example

[Emulator Debugger (MB2198)]

```
>SET FREQUENCY /MAX 64
```

[Emulator Debugger (MB2100-01)]

```
>SET FREQUENCY /MAX /MHZ 64
```

Notes:

- There is a possibility that the emulator malfunctions when a frequency lower than the operating frequency is set.

[Emulator Debugger (MB2100-01)]

- If this command is executed while the high speed communication is running, the setting is not applied until a reset is input.
-

1.49 SHOW FREQUENCY

The **SHOW FREQUENCY** command displays the maximum operating frequency or standard clock frequency for high speed communication set by the **SET FREQUENCY** command.

■ SHOW FREQUENCY

- Debugger

Simulator		X
Emulator	(MB2197)	X
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		X

- Format

SHOW FREQUENCY

Command qualifiers

/MAX (Cannot be omitted)

Displays the maximum operating frequency for the CPU or the standard clock frequency for high speed communication.

- Description

Displays the frequency configured using the **SET FREQUENCY** command.

Emulator Debugger (MB2198): Current maximum operating frequency for the CPU

Emulator Debugger (MB2100-01): Standard clock frequency for high speed communication

- Example

[Emulator Debugger (MB2198)]

```
>SHOW FREQUENCY /MAX
Max frequency: 33 MHz
```

[Emulator Debugger (MB2100-01)]

(Normal communication)

```
>SHOW FREQUENCY /MAX
Communication mode: normal
Max frequency: 40 MHz
```

(High speed communication)

```
>SHOW FREQUENCY /MAX
Communication mode: high
Max frequency: 100 MHz
```

1.50 CANCEL FREQUENCY

The **CANCEL FREQUENCY** command inhibits the high speed communication, and switches to the normal communication mode.

■ CANCEL FREQUENCY

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	○
Monitor		×

- Format

CANCEL FREQUENCY

Command qualifiers

/MAX (Cannot be omitted)

The high speed communication is inhibited.

- Description

The **CANCEL FREQUENCY** command inhibits the high speed communication, and switches to the normal communication mode.

The debugger communication speed returns to normal speed.

- Example

```
>CANCEL FREQUENCY /MAX
```


1.51 SET REALTIMEMEMORYAREA

The **SET REALTIMEMEMORYAREA** command sets a real time memory area.

■ SET REALTIMEMEMORYAREA

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

SET REALTIMEMEMORYAREA area number, area address

Parameters

area number (default: decimal number)

Specifies the real time memory area number to be set (1, 2).

area address (default: decimal number)

Specifies the address of the real time memory area to be set.

- Description

The SET REALTIMEMEMORYAREA command sets a real time memory area that can be monitored in real time.

Two areas up to 256 bytes can be set based on the entered addresses.

- Example

```
>SET REALTIMEMEMORYAREA 1, 1000
```

Notes:

- If a real time memory area already exists at the specified area number, the setting is overwritten.
- If a watch variable is overwritten in an invalid real time memory area, the following message is displayed:
"Please check! There is watch-variable that the realtime-monitoring function becomes invalid."

1.52 SHOW REALTIMEMEMORYAREA

The **SHOW REALTIMEMEMORYAREA** command displays the real time memory area set by the **SET REALTIMEMEMORYAREA** command.

■ SHOW REALTIMEMEMORYAREA

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	×
Monitor		×

- Format

SHOW REALTIMEMEMORYAREA

- Description

Displays the real time memory area set by the **SET REALTIMEMEMORYAREA** command.

If no such area is set, "not found" is displayed.

- Example

```
>SHOW REALTIMEMEMORYAREA
no.      address
1    0003A200 .. 0003A2FF
2    0003B500 .. 0003B5FF
```

1.53 ENABLE POWERONDEBUG

The **ENABLE POWERONDEBUG** command enables the Power-on debug.

■ ENABLE POWERONDEBUG

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	○
Monitor		×

- Format

ENABLE POWERONDEBUG

- Description

The **ENABLE POWERONDEBUG** command enables the Power-on debug.

- Example

```
>ENABLE POWERONDEBUG
```

Note:

[Emulator Debugger (MB2100-01)]

Power-on debug cannot be used when security is enabled.

For details of the security, refer to "2.4.1.3 Security" of "SOFTUNE WORKBENCH USER'S MANUAL".

1.54 DISABLE POWERONDEBUG

The **DISABLE POWERONDEBUG** command disables the Power-on debug.

■ DISABLE POWERONDEBUG

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	○
Monitor		×

- Format

DISABLE POWERONDEBUG

- Description

The **DISABLE POWERONDEBUG** command disables the Power-on debug.

- Example

```
>DISABLE POWERONDEBUG
```

1.55 SHOW POWERONDEBUG

The **SHOW POWERONDEBUG** command displays the state of the current power-on debug.

■ SHOW POWERONDEBUG

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW POWERONDEBUG

- Description

The SHOW POWERONDEBUG command displays enable/disable state of the current power-on debug.

- Example

When Power-on debug is enabled.

```
>SHOW POWERONDEBUG
power on debug : enable
```

When Power-on debug is disabled.

```
>SHOW POWERONDEBUG
power on debug : disable
```

1.56 SYNCHRONIZE FLASH

The **SYNCHRONIZE FLASH** command synchronizes the contents of the debugger memory and the flash memory.

■ SYNCHRONIZE FLASH

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	○
Monitor		×

- Format

<Format 1>

SYNCHRONIZE FLASH /FROMDEBUGGER {address | address-range} [, ...]

<Format 2>

SYNCHRONIZE FLASH /TODEBUGGER {address | address-range} [, ...]

<Format 3>

SYNCHRONIZE FLASH /FROMDEBUGGER /ALL

<Format 4>

SYNCHRONIZE FLASH /TODEBUGGER /ALL

Parameters

address (address formula)

Specifies the address including the area where the flash memory is synchronized.

address-range (address formula)

Specifies the address of the area where the flash memory is synchronized.

Command qualifiers

- Range

/ALL

All areas are synchronized to the flash memory.

- Synchronization of the flash memory

/FROMDEBUGGER

Synchronizes by writing the contents of the debugger memory to the flash memory.

/TODEBUGGER

Synchronizes by writing the contents of the flash memory to the debugger memory.

- Description

The **SYNCHRONIZE FLASH** command synchronizes the contents of the debugger memory and the flash memory.

The area that can be synchronized is limited to the internal area only.

- Example

```
>SYNCHRONIZE FLASH /FROMDEBUGGER 1000..1FFF
```

Note:

If the memory content held by the debugger is slightly changed, writing is performed for the entire sector of the target when the data is reflected to the flash.

1.57 ERASE FLASH

The ERASE FLASH command erases the flash memory.

■ ERASE FLASH

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	○
Monitor		×

- Format

ERASE FLASH {address | address-range} [, ...]

ERASE FLASH /ALL

Parameters

- Reset level

address (address formula)

Specifies the address including the area where the flash memory is erased.

address-range (address formula)

Specifies the address of the area where the flash memory is erased.

Command qualifiers

- Range

/ALL

Erases all areas of the flash memory.

- Description

The ERASE FLASH command erases the flash memory.

The area that can be erased is limited to the internal area only.

- Example

```
>ERASE FLASH 1000..1FFF
```


1.58 ENABLE DMA

The **ENABLE DMA** command enables the DMA operation at break.

■ ENABLE DMA

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

ENABLE DMA

- Description

The **ENABLE DMA** command enables the DMA operation at break.

- Example

```
>ENABLE DMA
```

Note:

[Emulator Debugger (MB2100-01)]

Please do not access to the debug work RAM through the DMA transfer at the user program break. If accessed, results of the user DMA operation may not be correct.

For details of the debug work RAM, refer to the Hardware Manual for the product type you are using.

1.59 DISABLE DMA

The **DISABLE DMA** command disables the DMA operation at break.

■ DISABLE DMA

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

DISABLE DMA

- Description

The **DISABLE DMA** command disables the DMA operation at break.

- Example

```
>DISABLE DMA
```

1.60 SHOW DMA

The **SHOW DMA** command displays the setting of the DMA operation at break.

■ SHOW DMA

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW DMA

- Description

The SHOW DMA command displays the setting of the DMA operation at break.

- Example

When the DMA operation is enabled at break

```
>SHOW DMA
accepting DMA : enable
```

When the DMA operation is disabled at break

```
>SHOW DMA
accepting DMA : disable
```

1.61 ENABLE CLOCKSUPERVISOR

The **ENABLE CLOCKSUPERVISOR** command enables the response speed optimization at debugging.

■ ENABLE CLOCKSUPERVISOR

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

ENABLE CLOCKSUPERVISOR

- Description

The **ENABLE CLOCKSUPERVISOR** command enables the response speed optimization at debugging.

- Example

```
>ENABLE CLOCKSUPERVISOR
```

1.62 DISABLE CLOCKSUPERVISOR

The **DISABLE CLOCKSUPERVISOR** command disables the response speed optimization at debugging.

■ DISABLE CLOCKSUPERVISOR

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

DISABLE CLOCKSUPERVISOR

- Description

The **DISABLE CLOCKSUPERVISOR** command disables the response speed optimization at debugging.

- Example

```
>DISABLE CLOCKSUPERVISOR
```

1.63 SHOW CLOCKSUPERVISOR

The **SHOW CLOCKSUPERVISOR** command displays the setting state of the response speed optimization at debugging.

■ SHOW CLOCKSUPERVISOR

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW CLOCKSUPERVISOR

- Description

The **SHOW CLOCKSUPERVISOR** command displays the setting state of the response speed optimization at debugging.

- Example

When the response speed optimization at debugging is enabled.

```
>SHOW CLOCKSUPERVISOR
clock supervisor : enable
```

When the response speed optimization at debugging is disabled.

```
>SHOW CLOCKSUPERVISOR
clock supervisor : disable
```

1.64 SET TIMERSCALE

The **SET TIMERSCALE** command sets the execution time measurement mode and the execution cycle conversion frequency.

■ SET TIMERSCALE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SET TIMERSCALE [Execution cycle conversion frequency]

Command qualifiers

/NORMAL and /HIGH can be specified only with products which can perform the real time measurement. For details, refer to "On-chip debugger (OCD)" in the hardware manual for the product type you are using.

- Specifying the execution time measurement mode

/CPU (default at start-up)

Performs the measurement of cycle count.

/NORMAL

Performs the real time measurement based on the reference frequency of the standard communication speed.

/HIGH

Performs the real time measurement based on the reference frequency of high communication speed.

- Specifying the unit for the execution cycle conversion frequency

/MHZ (Default when omitted)

Measures input execution cycle conversion frequencies in MHz.

This is disabled when the execution cycle conversion frequency is not specified.

/KHZ

Measures input execution cycle conversion frequencies in kHz.

This is disabled when the execution cycle conversion frequency is not specified.

/HZ

Measures input execution cycle conversion frequencies in Hz.

This is disabled when the execution cycle conversion frequency is not specified.

Parameter

Execution cycle conversion frequency

Set the frequency in an integer used to display the acquired clock count converted to real time when measuring cycle count. When omitted, the execution cycle conversion frequency is not changed.

- Description

Sets the execution time measurement mode and the execution cycle conversion frequency.

- Example

When measuring based on the reference frequency of standard communication speed

```
> SET TIMERSCALE /NORMAL
```

When measuring based on the CPU clock

```
> SET TIMERSCALE /CPU
```

When specifying the execution cycle conversion frequency to 12.5 MHz

```
> SET TIMERSCALE /KHZ 12500
```

When specifying the execution cycle conversion frequency to 4 MHz based on the CPU clock

```
> SET TIMERSCALE /CPU 4
```

Notes:

- When the command qualifier /HIGH is specified, the measurement is not performed while high speed communication mode is disabled. A hyphen ("-") will be displayed instead of the value measured.
- When setting the measurement unit to the reference frequency of high communication speed
 - The measurement is not performed in normal communication mode.
 - The measurement result is cleared when changing the standard frequency for high speed communication.

For details of the measurement unit, refer to "4.4.14.2 Performance (Emulator Debugger [MB2100-01])" of "SOFTUNE Workbench Operation Manual".

For details of the standard frequency for high speed communication, refer to "4.7.2 Setting Debug Environment" of "SOFTUNE Workbench Operation Manual".

1.65 SHOW TIMERSCALE

The **SHOW TIMERSCALE** command displays the setting state of the execution time measurement mode and execution cycle conversion frequency.

■ SHOW TIMERSCALE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW TIMERSCALE

- Description

The SHOW TIMERSCALE command displays the setting state of the execution time measurement mode and execution cycle conversion frequency.

Displays the following contents.

- When the execution time measurement mode is set to cycle count
 Timer scale : <Execution time measurement mode>
 CPU clock : <Execution cycle conversion frequency/"None" when not set>
- When the execution time measurement mode is not set to cycle count
 Timer scale: <Execution time measurement mode> (<reference frequency of communication speed>)
 CPU clock : <Execution cycle conversion frequency/"None" when not set>

- Example

When specifying the execution cycle conversion frequency to 32768 Hz based on the CPU clock

```
>SET TIMERSCALE /CPU /HZ 32768
>SHOW TIMERSCALE
Timer scale : CPU clock
CPU clock : 32768Hz
```

When measuring based on the reference frequency of standard communication speed

```
>SET TIMERSCALE /NORMAL
>SHOW TIMERSCALE
Timer scale : Normal(4000003Hz)
```

1.66 CANCEL TIMERSCALE

The **CANCEL TIMERSCALE** command clears the setting of the execution cycle conversion frequency.

■ CANCEL TIMERSCALE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

CANCEL TIMERSCALE

- Description

Clears the setting of the execution cycle conversion frequency.

- Example

When clearing the setting of the execution cycle conversion frequency

```
> CANCEL TIMERSCALE
> SHOW TIMERSCALE
Timer scale : CPU clock
CPU clock : None
```

CHAPTER 2 Program Execution Commands



This chapter describes the Program Execution commands.

- 2.1 GO
- 2.2 ABORT
- 2.3 SET ABORT
- 2.4 SHOW ABORT
- 2.5 STEP
- 2.6 SET STEP
- 2.7 SHOW STEP
- 2.8 CALL
- 2.9 CLEAR CALL
- 2.10 SHOW STATUS

2.1 GO

The GO command executes the program from the specified starting address.

■ GO

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

● Format

GO [starting-address] [, break-address]

Parameters

starting-address (address formula)

Specifies the execution start address of the program.

This parameter is invalid if /RETURN is specified.

break-address (address formula)

Specifies the execution stop address of the program.

This parameter is invalid if /RETURN or /NOWAIT is specified.

Command qualifiers

• Return setting

/RETURN

Executes a program from the currently executing function to the return location to its parent function.

Only programs coded in C/C++ language can use this function.

The optimized program may not be stopped normally.

• Selection of command input during execution

/WAIT (default when not specified)

Once a user program starts, any command input can not be accepted until the program stops.

/NOWAIT

Commands can be input after the user program is run without waiting for the program to stop.

This qualifier cannot be specified if /RETURN is specified.

Temporary breaks cannot be specified.

● Description

The GO command executes the program from the specified starting address.

If starting-address specifying is omitted, the program is executed from the current address indicated by the program counter.

The break address set by the GO command is automatically deleted when program execution is stopped.

● Example

```
>GO power$20
Break at main$10
>GO power$20, main$5
```

2.2 ABORT

The **ABORT** command stops the currently executing program.

■ ABORT

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

ABORT

Command qualifiers

/CANCEL

Cancels the stop request of MCU.

On the emulator debugger (MB2100-01) only, can be specified.

- Description

[Except for Emulator Debugger (MB2100-01)]

The ABORT command stops the currently executing program.

This command is valid in the following situations.

- If /NOWAIT is specified by the GO command
- If the continuous execution button is pressed in the execution toolbar

[Emulator Debugger (MB2100-01)]

The ABORT command stops MCU according to the specified condition set by the SET ABORT command.

- Example

[Except for Emulator Debugger (MB2100-01)]

>ABORT

[Emulator Debugger (MB2100-01)]

> ABORT /CANCEL

Note:

If this command is input while the user program is stopped, the message "Command error(Now MCU stopping)." is displayed.

2.3 SET ABORT

The **SET ABORT** command specifies the stop request condition of the **ABORT** command.

■ SET ABORT

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SET ABORT Interrupt level of the stop request

SET ABORT /FORCE

Parameters

Interrupt level of the stop request

Specifies interrupt level (D'1 to D'30) of the stop request.

For details of the value of the interrupt level setting, refer to the Hardware Manual for the product type you are using.

Command qualifiers

/FORCE

Does not set the interrupt level, and sets the forced stop as the stop request condition.

- Description

The **SET ABORT** command specifies the stop request condition of the **ABORT** command.

- Example

```
>SET ABORT 20
```

2.4 SHOW ABORT

The **SHOW ABORT** command displays the stop request condition of the **ABORT** command set by the **SET ABORT** command.

■ SHOW ABORT

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW ABORT

- Description

The **SHOW ABORT** command displays the stop request condition of the **ABORT** command set by the **SET ABORT** command.

- Example

For forced stop:

```
>SHOW ABORT
abort level : force
```

For interrupt level specified:

```
>SHOW ABORT
abort level : 20
```

2.5 STEP

The **STEP** command executes the program in units of source lines or machine instructions according to the condition set by the **SET STEP** command.

■ STEP

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

● Format

STEP [step-count]

Parameter

step-count (default: decimal number)

Specifies the execution number of the STEP command (D'1 to D'4294967295).

If the parameter is omitted, the execution number is 1.

Command qualifiers

• Step unit specification

/INSTRUCTION

Executes program in the unit of machine instruction.

/LINE

Executes program in the unit of source line.

/AUTOMATIC (default at start-up)

Automatically changes execution unit according to source window display mode as follows:

- When the source window display mode is the source line display mode, the program is executed in the unit of source line (/LINE).
- When the source window display mode is another display mode, the program is executed in the unit of machine instruction (/INSTRUCTION).

• Step operation specification

/INTO

Executes program for each step in called function, subroutine, or interrupt handler.

/OVER

Executes function call at /LINE and subroutine call instructions (i.e., CALL) at /INSTRUCTION and software interrupt instructions (i.e., INT) as one step.

Function call is valid when /LINE is specified. Subroutine call instructions and software interrupt instructions are valid when /INSTRUCTION is specified.

- Interrupt mask in step (when omitted, set condition by SET STEP)

/MASK

Set mask interrupt.

On the emulator debugger (MB2100-01) only, can be specified.

/NOMASK

Does not set mask interrupt.

On the emulator debugger (MB2100-01) only, can be specified.

- Description

The STEP command executes the program in the unit of source line or machine instruction according to the condition set by the SET STEP command.

The condition set by the SET STEP command can be invalidated by specifying a command qualifier.

- Example

[Except for Emulator Debugger (MB2100-01)]

```
>STEP
```

```
>STEP/INSTRUCTION
```

[Emulator Debugger (MB2100-01)]

```
>STEP /NOMASK
```

Note:

[Emulator Debugger (MB2100-01)]

When omitting the specifying of "Interrupt mask in step" by the command qualifiers, the specifying of the interrupt mask set by SET RESET is applied.

2.6 SET STEP

The **SET STEP** command specifies the step execution condition when no command qualifier is specified in the **STEP** command.

■ SET STEP

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

● Format

SET STEP

Command qualifiers

• Step unit specification

/INSTRUCTION

Executes program in the unit of machine instruction.

/LINE

Executes program in the unit of source line.

/AUTOMATIC (default at start-up)

Automatically changes execution unit according to source window display mode as follows:

- When the source window display mode is the source line display mode, the program is executed in the unit of source line (/LINE).
- When the source window display mode is another display mode, the program is executed in the unit of machine instruction (/INSTRUCTION).

• Step operation specification

/INTO (default at start-up)

Executes program for each step in called function, subroutine, or interrupt handler.

/OVER

Executes the following instructions as one step:

- Function call when /LINE is specified
- Subroutine call instruction (e.g. CALL) when /INSTRUCTION is specified
- Software interrupt instruction (e.g. INT) when /INSTRUCTION is specified

• Interrupt mask during step

/MASK

Set mask interrupt.

On the emulator debugger (MB2100-01) only, can be specified.

/NOMASK (default at start-up)

Does not set mask interrupt.

On the emulator debugger (MB2100-01) only, can be specified.

- Description

The SET STEP command specifies the step execution condition when no command qualifier is specified in the STEP command.

When the SOFTUNE Workbench is started, the step execution condition is SET STEP /AUTOMATIC/INTO/MASK.

- Example

[Except for Emulator Debugger (MB2100-01)]

```
>SET STEP/INSTRUCTION
```

[Emulator Debugger (MB2100-01)]

```
>SET STEP/MASK
```

2.7 SHOW STEP

The **SHOW STEP** command displays the step execution condition of the current **STEP** command.

■ SHOW STEP

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SHOW STEP

- Description

The **SHOW STEP** command displays the step execution condition of the current **STEP** command.

For emulator debugger (MB2100-01), the command additionally displays the interrupt mask condition.

- Example

[Except for Emulator Debugger (MB2100-01)]

```
>SHOW STEP
step mode : instruction, into
```

[Emulator Debugger (MB2100-01)]

When set interrupt mask in step

```
>SHOW STEP
step mode : automatic, into, mask
```

When does not set interrupt mask in step

```
> SHOW STEP
step mode : automatic, into, nomask
```

2.8 CALL

The **CALL** command executes the specified function and displays a return value.

■ CALL

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

- Format

CALL function-name ([argument [, . . .]])

Parameters

function-name

Specifies the name of function to be called.

argument

Complies with C/C++ language arguments.

However, structures unions and class cannot be specified as variable names.

Command qualifiers

/DISPLAY (Default at start-up)

Specifies the display of a return value.

/NODISPLAY

Specifies the non-display of a return value.

- Description

The CALL command executes the specified functions to display return values. The return value is set to the built-in variable %CALL.

The argument of the specified function is evaluated in the dummy argument format, then executed.

If the count of specified actual arguments is more than the count of dummy arguments, the extra actual arguments are evaluated in an int type.

When the program hits a break point while the CALL command is executing a function, a break occurs at that position.

To execute the CALL command continuously, use the GO command.

To terminate the execution of the CALL command, use the CLEAR CALL command.

Note that it is not possible to nest this command.

The CALL command sets the break point in the current PC, and sets the return address so as to return to that point. Then, the command calls a function.

Therefore, if the function called by the CALL command passes through the current PC, a break occurs in the middle of executing the function.

In such cases, the following message is displayed.

Break at address by Invalid call termination

The execution of the CALL command is continued by re-starting the execution by the GO command.

Note:

The CALL command may change the resources such as the registers, memory or I/O from the state prior to the function call. To restore registers, hold contents prior to the function call and execute the functions, or use the CLEAR CALL command.
Other resources can not be restored.

- Example

```
>CALL debug (cmd, p)
return value is H'0001
```

2.9 CLEAR CALL

The **CLEAR CALL** command cancels the **CALL** command and restores the status set before the register is called.

■ CLEAR CALL

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

- Format

CLEAR CALL

- Description

The **CLEAR CALL** command cancels the **CALL** command and restores the status set before the register is called.

- Example

```
>CALL debug (cmd, p)
Break at 00FF0F20 by breakpoint
>CLEAR CALL
```

2.10 SHOW STATUS

The **SHOW STATUS** command displays the MCU execution status.

■ SHOW STATUS

- Debugger

Simulator		○
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	○
Monitor		○

- Format

SHOW STATUS

- Description

[Except for Emulator Debugger (MB2100-01)]

When program is executing

The SHOW STATUS command displays the MCU execution status.

When program is stopping

The command displays the break factor of the immediately-preceding program execution.

[Emulator Debugger (MB2100-01)]

When program is executing

The SHOW STATUS command displays the MCU execution status.

The command also displays the last branch PC value.

When program is stopping

The command displays all the detected break factors.

The execution status of MCU is displayed as follows.

Display contents		Debugger				
		Simulator	Emulator			Monitor
Display result	execution status of MCU		MB2197	MB2198	MB2100-01	
Break	Break (user state)	○	○	○	○	○
Execute	Execute (debug state)	○	○	○	○	○
Stop	Stop mode	○	×	×	○	×
Timer	Watch mode	×	×	×	○	×
Sleep	Sleep mode	○	×	×	×	×
Sleep(CPU)	CPU sleep mode	×	×	×	○	×
Sleep(Bus)	Bus sleep mode	×	×	×	○	×
Halt	CPU stop state	×	×	×	○	×
Unknown	The state of CPU cannot be detected.	×	×	×	○	×

○: Displaied ×: Not displaied

For the break factor, refer to "APPENDIX C EXECUTION SUSPENSION MESSAGES LIST".

- Example

[MCU breaking]

[Except for Emulator Debugger (MB2100-01)]

>SHOW STATUS

break at 0000FF00 by breakpoint

[Emulator Debugger (MB2100-01)]

>SHOW STATUS

break at 00060000 by hardware breakpoint / data event break

[MCU executing]

[Except for Emulator Debugger (MB2100-01)]

>SHOW STATUS

MCU status : executing

[Emulator Debugger (MB2100-01)]

>SHOW STATUS

MCU status : stop

PC=0008012E

CHAPTER 3 Break/Event Control Command



This chapter describes the Break/Event Control commands.

- 3.1 SET BREAK (type 1)
- 3.2 SET BREAK (type 2)
- 3.3 SET BREAK (type 3)
- 3.4 SET BREAK (type 4)
- 3.5 SHOW BREAK
- 3.6 CANCEL BREAK
- 3.7 ENABLE BREAK
- 3.8 DISABLE BREAK
- 3.9 SET DATABREAK (type 1)
- 3.10 SET DATABREAK (type 2)
- 3.11 SHOW DATABREAK
- 3.12 CANCEL DATABREAK
- 3.13 ENABLE DATABREAK
- 3.14 DISABLE DATABREAK
- 3.15 SET EVENT
- 3.16 SHOW EVENT
- 3.17 CANCEL EVENT
- 3.18 ENABLE EVENT
- 3.19 DISABLE EVENT
- 3.20 SET CODEEVENT
- 3.21 SHOW CODEEVENT
- 3.22 CANCEL CODEEVENT
- 3.23 ENABLE CODEEVENT
- 3.24 DISABLE CODEEVENT
- 3.25 SET DATAEVENT
- 3.26 SHOW DATAEVENT
- 3.27 CANCEL DATAEVENT

- 3.28 ENABLE DATAEVENT
- 3.29 DISABLE DATAEVENT
- 3.30 SET SEQUENCE (type 1)
- 3.31 SET SEQUENCE (type 2)
- 3.32 SHOW SEQUENCE
- 3.33 CANCEL SEQUENCE
- 3.34 ENABLE SEQUENCE
- 3.35 DISABLE SEQUENCE
- 3.36 SET TRIGGER
- 3.37 SHOW TRIGGER
- 3.38 ENABLE ALIGNMENTBREAK
- 3.39 DISABLE ALIGNMENTBREAK
- 3.40 SHOW ALIGNMENTBREAK
- 3.41 SET BREAKCONDITION (type 1)
- 3.42 SET BREAKCONDITION (type 2)
- 3.43 SHOW BREAKCONDITION

3.1 SET BREAK (type 1)

The **SET BREAK (type 1)** command sets a breakpoint at the specified break address.

■ SET BREAK (type 1)

● Debugger

Simulator		○
Emulator	(MB2197)	○
	(MB2198)	⊙*
	(MB2100-01)	⊙
Monitor		○

*: If "Setting break point while running" is disabled, the command cannot be used during execution of the user program.

● Format

SET BREAK break-address [& = address mask] [, pass-count] [, {command; command...}]

Parameters

break-address [& = address mask] (address formula, data formula)

Specifies the address at which breakpoint set.

With specifying the address mask, only the bit where the address mask is "1" becomes effective, and other bits are ignored.

Without specifying the address mask, all bits become effective.

pass-count (default: decimal number)

Sets the number of times breakpoints are to be hit to stop the execution. The pass count number is specified in the following ranges depending on the types of the debugger.

If pass-count is omitted, 1 is assumed.

Debugger		Pass-count	Remarks
Emulator	MB2198	1 to 255	Limitation FR60Lite / FR80S
		Invalid	Except for FR60Lite / FR80S
	MB2100-01	1 to 1048575	—
	MB2197	Invalid	—
Simulator		1 to 65535	—
Monitor		Invalid	—

command

Specifies the command list for executing when the break address is hit. Two or more commands can be specified by using the semicolon.

This function is valid when the debugger type is only a simulator debugger.

Command qualifiers

[Simulator Debugger]

/BREAK (default when omitted)

After the command list is processed, the instruction execution is stopped when the breakpoint is hit.

/NOBREAK

After the command list is processed, the instruction execution is restarted when the breakpoint is hit.

[Emulator Debugger]

- Specifies the type of breakpoint.

/SOFT

Specifies software breakpoint.

/COUNT

Specifies hardware/count break. This qualifier can be used only for the emulator debugger (MB2198 (FR60Lite, FR80S) or MB2100-01).

/HARD (default at start-up)

Specifies hardware breakpoint. For the chip with ROM patch function, the ROM patch function can be used as a hardware break when all the 5 hardware breakpoints are used. For details, refer to "2.3.4.1 Code Break" of "SOFTUNE WORKBENCH USER'S MANUAL". For emulator debugger (MB2198), it can be specified only when the setting of breakpoint in operating is valid.

[Monitor Debugger]

/SOFT

Specifies software breakpoint.

● Description

The SET BREAK (type 1) command sets a breakpoint at the specified break address.

The maximum settable count of breakpoints to be specified is as follows:

Debugger		Breakpoint		
		Hardware	Software	Hardware/count
Emulator	MB2197	5	4096	0
	MB2198	5	4096	2
	MB2100-01	8 (*1)	4096	0
Simulator (*2)		65535 (*3) 67 (*4)		0
Monitor		0	16	0

*1: The maximum set number may become few depending on the target MCU used for emulator debugger (MB2100-01).

The count of breakpoints specified above is the sum of the value set in the following commands in the emulator debugger (MB2100-01).

SET BREAK/DATAWATCH

SET DATABREAK

SET EVENT

SET TRACETRIGGER

*2: There is no distinction between hardware and software for the simulator debugger.

*3: For simulator debugger of normal speed

*4: For simulator debugger of high speed

●Example

[Simulator Debugger]

```
>SET BREAK/NOBREAK main, 1,{SHOW TRACE; SHOW TIMER}
```

[Emulator Debugger]

```
>SET BREAK 00ff0200
```

```
>SET BREAK 00ff0300, 3
```

Notes:

[Emulator Debugger (MB2197 or MB2198)]

- In software break, no breakpoint can be set in write-inhibit areas such as ROM. In this case, a verify error occurs during program execution.
- Set a software breakpoint at the starting address of an instruction. If a software breakpoint is set in other addresses, the program may malfunction.
- To use ROM patch break, it is necessary to set resources for the ROM patch to be used as break. For details, refer to the "4.6.4 Breakpoints" of "SOFTUNE Workbench Operation Manual".
- Internal ROM areas are the only memory areas in which ROM patch break can be set. No break can be set, when any other area is specified.
- No data monitoring condition can be set for ROM patch break.
- Always set ROM patch break at the starting address of an instruction. If a breakpoint is set in the middle of an instruction, the CPU cannot interpret the instruction correctly and may malfunction.
- When an address to which ROM patch break has been set is read during execution, it is read as a break instruction. Therefore, to read a break address during execution, delete or disable the ROM patch break beforehand.
- When ROM patch break is set during execution, the execution stop time becomes longer, compared with when a hardware break is set.

[Emulator Debugger (MB2100-01)]

- When the pass count mode is in the sequence mode, this command cannot be used.

For details, refer to "1.9 SET MODE (type 2)".

[Simulator debugger]

- The following execution type command cannot be specified for the command list.
 - GO
 - STEP
 - CALL
 - SYSTEMCALL
 - The STUB function (restart command execution and instruction execution) is not executed in STEP/INTO and STEP/OVER. After the breakpoint is hit, execution is stopped.
-

3.2 SET BREAK (type 2)

The SET BREAK (type 2) command sets a data monitoring breakpoint (software) at the specified break address.

■ SET BREAK (type 2)

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

● Format

SET BREAK/BREAKCONDITION break-address

Parameters

break-address (address formula)

Specifies the address at which breakpoint set.

Function type command qualifiers

/BREAKCONDITION

The conditions set in SET BREAKCONDITION are combined into break conditions.

Command qualifiers

- Specifies the breakpoint type

/HARD (default at start-up)

Specifies hardware breakpoint

● Description

The SET BREAK (type 2) command sets a data monitoring breakpoint (software) at the specified break address.

This command is used to combine the conditions set in SET BREAKCONDITION into break conditions.

Passing through the specified break address, the break conditions set in SET BREAKCONDITION are checked to determine whether to set breakpoints.

● Example

```
>SET BREAK /BREAKCONDITION 00ff0200
```


3.3 SET BREAK (type 3)

The **SET BREAK (type 3)** command sets a data monitoring breakpoint (hardware) at the specified break address.

■ SET BREAK (type 3)

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

SET BREAK/DATAWATCH code-address, data-address [&=address-mask] [,!]d=data
[&=data-mask]]

Parameters

code-address (address formula)

Specifies the address of code that serves as a data monitoring condition.

data-address (address formula)

Specifies the address of data that serves as a data monitoring condition.

address-mask (data formula)

Specifies the mask bit pattern for the address of data that serves as a data monitoring condition.

Comparison is made only on the address value in the bit position where 1 is set.

Cannot be specified on the emulator debugger (MB2100-01).

data (data formula)

Specifies the address of the data that serves as a data monitoring condition.

When the /ANYTHING qualifier is specified, 32-bit data is targeted for comparison. To compare 8/16-bit data, it is necessary to specify the mask data.

data-mask (data formula)

Specifies the mask bit pattern for the data that serves as a data monitoring condition.

Comparison is made only on the data value in the bit position where 1 is set.

Command qualifiers

- Specifying access size

/BYTE (Default on the emulator debugger (MB2100-01) when not specified)

Specifies 8-bit access as a data monitoring condition.

One byte is monitored from the specified data address for emulator debugger (MB2100-01).

/HALFWORD

Specifies 16-bit (2-byte) access as a data monitoring condition.

Two bytes are monitored from the specified data address for emulator debugger (MB2100-01).

/WORD

Specifies 32 bits (4-byte) access as a data monitoring condition.

Four bytes are monitored from the specified data address for emulator debugger (MB2100-01).

/ANYTHING (Default on the emulator debugger (MB2198) when not specified)

A break occurs when the specified address is accessed, regardless of the access data length.

Cannot be specified on the emulator debugger (MB2100-01).

Specifying access attribute

/READ

Specifies data read access as a data monitoring condition.

/WRITE

Specifies data write access as a data monitoring condition.

When omitted, /READ/WRITE is specified.

● Description

The SET BREAK (type 3) command sets a data monitoring break (hardware) at the specified break address.

The data monitoring break is a function that determines whether to set breakpoints by referring data in the data address when execution arrives at the specified code address.

[Emulator Debugger (MB2198)]

Up to 4 breakpoints can be set, but the count of breakpoints to be specified is the sum of the value set in the following commands.

SET EVENT

SET TRACETRIGGER

[Emulator Debugger (MB2100-01)]

Up to 8 breakpoints can be set, but the count of breakpoints to be specified is the sum of the value set in the following commands.

SET BREAK/HARD

SET DATABREAK

SET EVENT

SET TRACETRIGGER

● Example

[Emulator Debugger (MB2198)]

```
>SET BREAK/DATAWATCH/WRITE/WORD 00ff0200, 120034&==ffffffc, !d=1234&=feff
```

Notes:

[Emulator Debugger (MB2198)]

- This command can use only for the FR60Lite. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".
- When the /ANYTHING qualifiers is specified, 32-bit data is targeted for comparison. To compare 8/16-bit data, it is necessary to specify the mask data.

Ex.) For 8-bit data (0x12)

```
SET BREAK /DATAWATCH/ANYTHING 0x10000, D=0x12000000&=0xff000000
```

Ex.) For 16-bit data (0x4567)

```
SET BREAK /DATAWATCH/ANYTHING 0x10000, D=0x45670000&=0xffff0000
```

- When event mode is set to performance mode, this command cannot be used.

[Emulator Debugger (MB2100-01)]

The specified data is monitored by bus access. Therefore, when there is no access to the specified data address, break does not occur.

3.4 SET BREAK (type 4)

The SET BREAK (type 4) command configures the default attributes for setting breakpoints.

■ SET BREAK (type 4)

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		×

- Format

SET BREAK /DEFAULT [/SOFT | /HARD]

Command qualifiers

/ DEFAULT (Cannot be omitted)

Configures the attributes for setting breakpoints.

/ SOFT

Configures software breaks as the default for setting breakpoints.

/ HARD (default when omitted)

Configures hardware breaks as the default for setting breakpoints.

- Description

The SET BREAK (type 4) command configures the default attributes for setting breakpoints.

- Example

```
>SET BREAK /DEFAULT /SOFT
```

3.5 SHOW BREAK

The **SHOW BREAK** command displays the breakpoints set by the **SET BREAK** command.

■ SHOW BREAK

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

● Format

SHOW BREAK [breakpoint-number [, . . .]]

Parameter

breakpoint-number (default: decimal number)

Specifies the breakpoint number.

Command qualifier

/ALL (default when omitted)

Displays all breakpoints.

/NORMAL [Emulator debugger (MB2198 (FR60Lite, FR80S), MB2100-01)]

Only the hardware or software break information is displayed.

/COUNT [Emulator debugger (MB2198 (FR60Lite, FR80S), MB2100-01)]

Only the hardware/count break information is displayed.

/DATAWATCH [Emulator debugger (MB2198 (FR60Lite), MB2100-01)]

Only the data monitoring break information is displayed. When event mode is set to performance mode, this cannot be specified.

/DEFAULT [Emulator debugger (MB2198, MB2100-01)]

Displays attributes on setting the breakpoints.

/GUARDEDACCESS [Emulator debugger (MB2100-01)]

Displays information of the guarded access break only.

● Description

The **SHOW BREAK** command displays the breakpoints set by the **SET BREAK** command.

- Example

[Simulator debugger]

```
>SHOW BREAK
no.  en/dis  address      pass-count      symbol
1    enable  00FF0F00      1 (    1)
4    disable 00FF20DE      65535 ( 1234)
Control: BREAK
Command: show timer
```

[Emulator Debugger (MB2198)]

```
>SHOW BREAK/ALL
no.  en/dis  address  kind  cond.  symbol
1    enable  0000F000  hard
Count
no.  en/dis  address  kind  pass-count  cond. symbol
1    enable  00FF0F00  hard  1(    1)    *   main
4    disable 00FF20DE  hard  65535( 1234)  func
data watch
no.  en/dis  address  data-addr d-dd-mask  data      mask      size  access  symbol
1    enable  00FF3032  00008000  .....    00000034  !000000FF  byte  read
```

[Emulator Debugger (MB2100-01)]

When guarded access break is enabled

```
>SHOW BREAK /GUARDEDACCESS
guarded access break : enable
```

When guarded access break is disabled

```
>SHOW BREAK /GUARDEDACCESS
guarded access break : disable
```

Note:

The hit count of breakpoints is not updated while running user program. Therefore, the value of hit count indicated while running user program is the one before the program running starts.

3.6 CANCEL BREAK

The **CANCEL BREAK** command cancels a breakpoint at the specified number.

■ CANCEL BREAK

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input checked="" type="radio"/> *
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

*: If "Setting break point while running" is disabled, the command cannot be used during execution of the user program.

- Format

CANCEL BREAK [breakpoint-number [, . . .]]

Parameter

breakpoint-number (default: decimal number)

Specifies the breakpoint number.

Breakpoint numbers can be checked using the SHOW BREAK command.

Command qualifier

/ALL

Cancels all breakpoints.

/NORMAL [Emulator debugger (MB2198 (FR60Lite, FR80S), MB2100-01)]

The hardware or software break is canceled.

/COUNT [Emulator debugger (MB2198 (FR60Lite, FR80S), MB2100-01)]

The hardware/count break is canceled.

/DATAWATCH [Emulator debugger (MB2198 (FR60Lite), MB2100-01)]

The data-monitoring break is canceled.

- Description

The CANCEL BREAK command cancels the breakpoints with specified number.

- Example

```
>CANCEL BREAK 1
>CANCEL BREAK 3, 4
```

Note:

When /ALL command qualifier is specified, the breakpoint number cannot be specified.

When /ALL command qualifier is not specified, specify the breakpoint number.

3.7 ENABLE BREAK

The **ENABLE BREAK** command enables the breakpoints with specified number.

■ ENABLE BREAK

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input checked="" type="radio"/> *
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

*: If "Setting break point while running" is disabled, the command cannot be used during execution of the user program.

● Format

ENABLE BREAK [breakpoint-number [, ...]]

Parameter

breakpoint-number (default: decimal number)

Specifies the breakpoint number.

Breakpoint numbers can be checked using the SHOW BREAK command.

Command qualifier

/ALL

Enables all breakpoints.

/NORMAL [Emulator debugger (MB2198 (FR60Lite, FR80S), MB2100-01)]

The hardware or software break is enabled.

/COUNT [Emulator debugger (MB2198 (FR60Lite, FR80S), MB2100-01)]

The hardware/count break is enabled.

/DATAWATCH [Emulator debugger (MB2198 (FR60Lite), MB2100-01)]

The data-monitoring break is enabled.

/GUARDEDACCESS [Emulator debugger (MB2100-01)]

The guarded access break is enabled.

● Description

The **ENABLE BREAK** command enables the breakpoints with the specified number.

● Example

```
>ENABLE BREAK 2
>ENABLE BREAK 3, 4
```


3.8 DISABLE BREAK

The **DISABLE BREAK** command disables the breakpoints with the specified number.

■ DISABLE BREAK

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input checked="" type="radio"/> *
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

*: If "Setting break point while running" is disabled, the command cannot be used during execution of the user program.

- Format

DISABLE BREAK [breakpoint-number [, . . .]]

Parameter

breakpoint-number (default: decimal number)

Specifies the breakpoint number.

Breakpoint numbers can be checked using the **SHOW BREAK** command.

Command qualifier

/ALL

Disables all breakpoints.

/NORMAL [Emulator debugger (MB2198 (FR60Lite, FR80S), MB2100-01)]

The hardware or software break is disabled.

/COUNT [Emulator debugger (MB2198 (FR60Lite, FR80S), MB2100-01)]

The hardware/count break is disabled.

/DATAWATCH [Emulator debugger (MB2198 (FR60Lite), MB2100-01)]

The data-monitoring break is disabled.

/GUARDEDACCESS [Emulator debugger (MB2100-01)]

The guarded access break is disabled.

- Description

The **DISABLE BREAK** command disables the breakpoints with the specified number.

- Example

```
>DISABLE BREAK 2
>DISABLE BREAK 3, 4
```

3.9 SET DATABREAK (type 1)

The **SET DATABREAK (type 1)** command breaks the program when data at the specified address is accessed.

■ SET DATABREAK (type 1)

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

SET DATABREAK data-access-address [, pass-count] [{ command; command...}]

Parameters

data-access-address (address formula)

Specifies the address at which data access breakpoint set.

pass-count (default: decimal number)

Sets the number of times breakpoints are to be hit to stop the execution.

If pass-count is omitted, 1 is assumed.

command

Specifies the command list to be executed when the break address is hit. Two or more commands can be specified by using the semicolon.

This function is valid when the debugger type is only a simulator debugger. This function is ignored in the emulator debugger and the monitor debugger.

Command qualifiers

- Specifying of STUB function

/BREAK (default when omitted)

When the breakpoint is hit, the instruction execution is stopped after the command list is processed.

/NOBREAK

When the breakpoint is hit, the instruction execution is restarted after the command list is processed.

- Specifying of Attribute

/READ

Breaks program when data read-accessed.

/WRITE

Breaks program when data write-accessed.

- Description

The SET DATABREAK command breaks the program when data at the specified address is accessed.

Use a command qualifier to set a break access type.

If no command qualifier is specified, /READ/WRITE is assumed. /READ/WRITE breaks the program at either access of data.

When the command qualifier size is specified, a break occurs under the following condition.

- A break occurs when a specified-size access is made to the specified address.

The pass count value is set each time the program is executed.

Data breakpoints can be specified are shown below.

Simulator debugger : 65535

If an automatic variable in the function is specified, the current address at which the variable is stored is set as the data access address (take care when using automatic variables).

To break the program when a C/C++ language variable is accessed, specify "&" before the variable as the variable address.

- Example

```
>SET DATABREAK/NOBREAK &checkflg,3,{SHOW TRACE; SHOW TIMER}
```

Notes:

- The following execution type command cannot be specified for the command list.
 - GO
 - STEP
 - CALL
 - SYSTEMCALL
 - The STUB function (restart command execution and instruction execution) is not executed in STEP/INTO and STEP/OVER. After the breakpoint is hit, execution is stopped.
-

3.10 SET DATABREAK (type 2)

The **SET DATABREAK (type 2)** command breaks the program when data at the specified address is accessed.

■ SET DATABREAK (type 2)

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙ *
	(MB2100-01)	⊙
Monitor		×

*: If "Setting break point while running" is disabled, the command cannot be used during execution of the user program.

● Format

SET DATABREAK data-access-address [,d=break data[&=break data mask]]

Parameters

- data-access-address (address formula)
Specifies the address at which data access breakpoint set.
- break data (Default: hexadecimal)
Specifies the data value that is the condition for a break.
- break data mask (Default: hexadecimal)
Specifies the valid bit of the data value that is the condition for a break.

Command qualifier

- Specifying access attribute
/READ
Breaks program when data read-accessed.
/WRITE
Breaks program when data write-accessed.
- Specifying access size
/BYTE (default when omitted)
Specifies 8 bits for the data width for causing a break condition.
/HALFWORD
Specifies 16 bits for the data width for causing a break condition.
/WORD
Specifies 32 bits for the data width for causing a break condition.

- Description

The SET DATABREAK command breaks the program when data at the specified address is accessed.

Use a command qualifier to set a break access type.

If no command qualifier is specified, this command breaks the program at the time of both data read access and data write access.

[Emulator Debugger (MB2198)]

Up to 2 breakpoints can be set.

[Emulator Debugger (MB2100-01)]

Up to 8 breakpoints can be set, but the count of breakpoints to be specified is the sum of the value set in the following commands.

SET BREAK/HARD

SET BREAK/DATAWATCH

SET EVENT

SET TRACETRIGGER

If an automatic variable in the function is specified, the current address at which the variable is stored is set as the data access address (take care when using automatic variables).

To break the program when a C/C++ variable is accessed, specify "&" before the variable as the variable address.

- Example

```
>SET DATABREAK &checkflg
```

Note:

[Emulator Debugger (MB2198)]

This command can be used only for the FR60Lite and FR80S. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.11 SHOW DATABREAK

The **SHOW DATABREAK** command displays the data access points set by the **SET DATABREAK** command.

■ SHOW DATABREAK

- Debugger

Simulator		⊙
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW DATABREAK [breakpoint-number [, ...]]

Parameter

breakpoint-number (default: decimal number)

Specifies the breakpoint number.

Command qualifier

/ALL (default when omitted)

Displays all data breakpoints.

- Description

The **SHOW DATABREAK** command displays the data access points set by the **SET DATABREAK** command.

- Example

[Simulator Debugger]

```
>SHOW DATABREAK
no.   en/dis   address   read/write   pass-count   symbol
1     enable   00001000   write only    5( 2)
Control: BREAK
Command: show status
```

[Emulator Debugger (MB2198)]

```
>SHOW DATABREAK
no. en/dis address      data      d-mask      size  access  symbol
1   enable 00000100  00000010  FFFFFFFF   byte  read/write
```

Note:

[Emulator Debugger (MB2198)]

This command can be used only for the FR60Lite and FR80S. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.12 CANCEL DATABREAK

The **CANCEL DATABREAK** command cancels the data access breakpoint(s) with specified number.

■ CANCEL DATABREAK

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	⊙ *
	(MB2100-01)	⊙
Monitor		×

*: If "Setting break point while running" is disabled, the command cannot be used during execution of the user program.

- Format

CANCEL DATABREAK [breakpoint-number [, . . .]]

Parameter

breakpoint-number (default: decimal number)

Specifies the breakpoint number.

Breakpoint numbers can be checked using the SHOW BREAK command.

Command qualifier

/ALL

Cancels all data access breakpoints.

- Description

The CANCEL DATABREAK command cancels the data access breakpoint(s) with specified number.

- Example

```
>CANCEL DATABREAK 1
>CANCEL DATABREAK 3, 4
```

Note:

[Emulator Debugger (MB2198)]

This command can be used only for the FR60Lite and FR80S. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.13 ENABLE DATABREAK

The **ENABLE DATABREAK** command enables the data access breakpoint(s) with specified number.

■ ENABLE DATABREAK

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	⊙ *
	(MB2100-01)	⊙
Monitor		×

*: If "Setting break point while running" is disabled, the command cannot be used during execution of the user program.

- Format

ENABLE DATABREAK [breakpoint-number [, . . .]]

Parameter

breakpoint-number (default: decimal number)

Specifies the breakpoint number.

Breakpoint numbers can be checked using the SHOW DATA BREAK command.

Command qualifier

/ALL

Enables all data breakpoints.

- Description

The **ENABLE DATABREAK** command enables the data access breakpoint(s) with specified number.

- Example

```
>ENABLE DATABREAK 2
>ENABLE DATABREAK 3, 4
```

Note:

[Emulator Debugger (MB2198)]

This command can be used only for the FR60Lite and FR80S. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.14 DISABLE DATABREAK

The **DISABLE DATABREAK** command disables the data access breakpoint(s) with specified number.

■ DISABLE DATABREAK

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	⊙*
	(MB2100-01)	⊙
Monitor		×

*: If "Setting break point while running" is disabled, the command cannot be used during execution of the user program.

- Format

DISABLE DATABREAK [breakpoint-number [, ...]]

Parameter

breakpoint-number (default: decimal number)

Specifies the breakpoint number.

Breakpoint numbers can be checked using the **SHOW DATA BREAK** command.

Command qualifier

/ALL

Disables all data breakpoints.

- Description

The **DISABLE DATABREAK** command disables the data access breakpoint(s) with specified number.

- Example

```
>DISABLE DATABREAK 2
>DISABLE DATABREAK 3, 4
```

Note:

[Emulator Debugger (MB2198)]

This command can be used only for the FR60Lite and FR80S. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.15 SET EVENT

The **SET EVENT** command sets the event that triggers a sequencer or performance.

■ SET EVENT

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

● Format

[Emulator Debugger (MB2198)]

[Event mode: Normal]

SET EVENT address [& = mask] [, [!] d = data [& = mask]] [,p = passcount]

[Event mode: Performance]

SET EVENT address [& = mask] [, [!] d = data [& = mask]] ,b = event number

[Emulator Debugger (MB2100-01)]

SET EVENT address

SET EVENT /SEQUENCE address [,p = passcount]

Parameters

address [& = mask] (address type, data type)

Specifies a memory location taken as an event generating condition. If a mask is specified, only one portion where the bit of the mask is 1 will be valid and the others will be "don't care".

If mask data is omitted, all the bits will be valid.

Automatic variables in C language cannot be specified.

On the emulator debugger (MB2100-01), mask data cannot be specified.

d = data [& = mask] (data type, data type)

Specify the data taken as an event generating condition. If a mask is specified, only one portion where the bit of the mask is 1 will be valid and the others will be "don't care".

If mask data is omitted, all the bits will be valid.

If ! is specified, the specified data will be assumed to be "not".

When the /ANYTHING qualifies is specified, 32-bit data is targeted for comparison. To compare 8/16-bit data, it is necessary to specify the mask data.

Cannot be specified on the emulator debugger (MB2100-01).

p = pass-count (default: decimal number)

Specifies the event occurrence count. Specify within the range of 1 to 16777215 on the emulator debugger (MB2198) or 1 to 1048575 on the emulator debugger (MB2100-01). If omitted, it is set to 1.

On the emulator debugger (MB2198), the pass count can be specified when the event mode is normal.

On the emulator debugger (MB2100-01), the pass count cannot be specified when /PERFORMANCE is specified.

b = event number

Specifies the event number (1 to 4).

On the emulator debugger (MB2198), when the event mode is performance, the specification of this parameter cannot be omitted.

Cannot be specified on the emulator debugger (MB2100-01).

Command qualifiers

- Access attributes

/CODE

Takes code access to specified address as event generating condition. On the emulator debugger (MB2198), cannot be specified for FR80S

/READ

Takes read access to specified address as event generating condition. On the emulator debugger (MB2198), cannot be specified for FR80S

/WRITE

Takes write access to specified address as event generating condition.

/CODE and /WRITE cannot be specified at the same time. Also, /CODE and /READ cannot be specified at the same time. If omitted, it means /CODE is specified.

For emulator debugger (MB2198)(FR80S), the omission means /WRITE is specified.

- Data length specifying

/WORD

Specifies handling of event condition data as 32 bits data.

/HALFWORD

Specifies handling of event condition data as 16 bits data.

/BYTE (default on the emulator debugger (MB2100-01))

Specifies handling of event condition data as 8 bits data.

/ANYTHING (default on the emulator debugger (MB2198))

Takes the accessing to the specified address as the event occurrence condition regardless of the data length. Cannot be specified on the emulator debugger (MB2100-01).

- Specifying function type

/SEQUENCE (default when omitted)

Sets the event for SEQUENCE.

On the emulator debugger (MB2100-01) only, can be specified.

/PERFORMANCE

Sets the event for PERFORMANCE.

On the emulator debugger (MB2100-01) only, can be specified.

- Description

[Emulator Debugger (MB2198)]

[Event mode: Normal]

The SET EVENT command sets the event that triggers a sequencer. If the data specifying is omitted, all data is ignored.

Up to four events can be set. Sharing hardware, however, the maximum settable count of events is the sum of the values specified in SET BREAK/DATAWATCH and in SET TRACETRIGGER.

Use the SET SEQUENCE command to set the sequencer by specifying the event number.

Event numbers can be checked using SHOW EVENT command, or from an address using the embedded function %EVENTUM.

[Event mode: Performance]

The starting/ending addresses, conditions for measuring performance, are specified using event numbers. The following meanings are applied to each event number.

Event 1 → Starting event in section 1

Event 2 → Ending event in section 1

Event 3 → Starting event in section 2

Event 4 → Ending event in section 2

[Emulator Debugger (MB2100-01)]

Sets the event that is that triggers a sequencer. Up to 3 events can be set. Because the hardware is shared, the maximum count is the sum of the values of the following commands.

SET BREAK

SET DATABREAK

SET BREAK/DATAWATCH

SET TRACETRIGGER

Configure the sequencer by using the SET SEQUENCE command to specify the event numbers. Event numbers can be checked using SHOW EVENT commands, or from an address using the embedded function %EVENTUM.

- Example

[Emulator Debugger (MB2198)]

```
>SET EVENT /READ, func1, b=1
```

```
>SET EVENT /WRITE &data[2], !d=h'10, b=2
```

Notes:

[Emulator Debugger (MB2198)]

- This command can be used only for the FR60Lite or FR80S. For details, refer to "2.3.5 Control by Sequencer" or "2.3.8 Measuring Performance" of "SOFTUNE WORKBENCH USER'S MANUAL".
- For FR80S, when the MCU operation mode is the external trace mode, only the written data in the built-in RAM area is stored in the trace buffer.
- The setting contents differ in each event mode.
- When the qualifier /ANYTHING is specified, 32-bit data is targeted for comparison. To compare 8/16-bit data, data masking is required.

Ex.) For 8-bit data(0x12)

```
SET BREAK/DATAWATCH /ANYTHING 0x10000, D=0x12000000&=0xff000000
```

Ex.) For 16-bit data(0x4567)

```
SET BREAK/DATAWATCH /ANYTHING 0x10000, D=0x45670000&=0xffff0000
```

- The sequence is released when resetting the address to the same address as the specified event.
Reset the sequence using the SET SEQUENCE command.

[Emulator Debugger (MB2100-01)]

- The sequence is released when the command qualifier /SEQUENCE is reset to the same address as the address having been set as the sequence.
Reset the sequence using the SET SEQUENCE command.
- The performance trigger is released when the command qualifier /PERFORMANCE is reset to the same address as the address having been set as the performance trigger.
Reset by the SET PERFORMANCE command.

3.16 SHOW EVENT

The **SHOW EVENT** command shows the contents set by the **SET EVENT** command.

■ SHOW EVENT

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		×

● Format

SHOW EVENT [event-number[,...]]

Parameter

event-number

Specifies the event number.

Specifies within the range of 1 to 4 on the emulator debugger (MB2198) or 1 to 16 on the emulator debugger (MB2100-01).

Command qualifier

/ALL (default when omitted)

Displays all events.

● Specifying function type

/SEQUENCE (default when not specified)

Displays the events for SEQUENCE.

On the emulator debugger (MB2100-01) only, can be specified.

/PERFORMANCE

Displays the events for PERFORMANCE.

On the emulator debugger (MB2100-01) only, can be specified.

● Description

The SHOW EVENT command shows the contents set by the SET EVENT command.

[Emulator Debugger (MB2198)]

The specified event number can also be referenced from the address by using the built-in function %EVENTNUM.

[Emulator Debugger (MB2100-01)]

When /SEQUENCE is specified:

The specified event number can also be referenced from the address by using the built-in function %EVENTNUM.

When /PERFORMANCE is specified:

The specified event number can also be referenced from the address by using the built-in function %PFMEVENTNUM.

- Example

```
>SHOW EVENT
```

Note:

[Emulator Debugger (MB2198)]

This command can be used only for FR60Lite and FR80S. For details, refer to "2.3.5 Control by Sequencer" or "2.3.8 Measuring Performance" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.17 CANCEL EVENT

The **CANCEL EVENT** command cancels the events with the specified number.

■ CANCEL EVENT

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

CANCEL EVENT [event-number [...]]

Parameter

event-number

Specifies the event number.

Specifies within the range of 1 to 4 on the emulator debugger (MB2198) or 1 to 16 on the emulator debugger (MB2100-01).

Command qualifier

/ALL (default when omitted)

Cancels all events.

- Specifying function type

/SEQUENCE (default when not specified)

Cancels the events for SEQUENCE.

On the emulator debugger (MB2100-01) only, can be specified.

/PERFORMANCE

Cancels the events for PERFORMANCE.

On the emulator debugger (MB2100-01) only, can be specified.

- Description

The CANCEL EVENT command cancels the event with the specified number.

[Emulator Debugger (MB2198)]

The specified event number can also be referenced from the address by using the built-in function %EVENTNUM.

[Emulator Debugger (MB2100-01)]

When /SEQUENCE is specified:

The specified event number can also be referenced from the address by using the built-in function %EVENTNUM.

When /PERFORMANCE is specified:

The specified event number can also be referenced from the address by using the built-in function %PFMEVENTNUM.

- Example

>CANCEL EVENT

Notes:

[Emulator Debugger (MB2198)]

- This command cannot be used except the FR60Lite or FR80S. For details, refer to "2.3.5 Control by Sequencer" or "2.3.8 Measuring Performance" of "SOFTUNE WORKBENCH USER'S MANUAL".
- The sequence is released when specifying it to the event number that has already been set as a sequence.
Reset the sequence using the SET SEQUENCE command.

[Emulator Debugger (MB2100-01)]

- When command qualifiers/PERFORMANCE is specified, the sequence is released when resetting it to the event number that has already been set as the sequence.
Reset the sequence using the SET SEQUENCE command.
 - When command qualifier/SEQUENCE is specified, the event number set to the parameter as a performance trigger is specified, the event cannot be deleted.
 - When command qualifiers/PERFORMANCE is specified, the performance trigger is released when specifying it to the event number that has already been set as a performance trigger.
Reset by the SET PERFORMANCE command.
 - When command qualifier/PERFORMANCE is specified, the event number set to the parameter as a sequence is specified, the event cannot be deleted.
-

3.18 ENABLE EVENT

The **ENABLE EVENT** command temporarily enables the event.

■ ENABLE EVENT

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

ENABLE EVENT [event-number [,...]]

Parameter

event-number

Specifies the event number.

Specifies within the range of 1 to 4 on the emulator debugger (MB2198) or 1 to 8 on the emulator debugger (MB2100-01).

Command qualifier

/ALL (default when omitted)

Enables all events.

- Description

The **ENABLE EVENT** command temporarily enables the event.

The specified event number can also be referenced from the address by using the built-in function %EVENTNUM.

- Example

```
>ENABLE EVENT
```

Note:

[Emulator Debugger (MB2198)]

This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.5 Control by Sequencer" or "2.3.8 Measuring Performance" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.19 DISABLE EVENT

The **DISABLE EVENT** command temporarily disables the event.

■ DISABLE EVENT

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

DISABLE EVENT [event-number[,...]]

Parameter

event-number

Specifies the event number.

Specifies within the range of 1 to 4 on the emulator debugger (MB2198) or 1 to 8 on the emulator debugger (MB2100-01).

Command qualifier

/ALL (default when omitted)

Disables all events.

- Description

The **DISABLE EVENT** command temporarily disables the event.

The specified event number can also be referenced from the address by using the built-in function %EVENTNUM.

- Example

```
>DISABLE EVENT
```

Note:

[Emulator Debugger (MB2198)]

This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.5 Control by Sequencer" or "2.3.8 Measuring Performance" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.20 SET CODEEVENT

The SET CODEEVENT command sets a code event.

■ SET CODEEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

SET CODEEVENT event-number, address [& = address-mask] [, pass-count]

Parameters

event-number

Specifies the event number (D'1 or D'2).

address [&=address-mask] (address formula, data formula)

Specifies the address at which event generation condition to be stored.

When address-mask is specified, only 1 address mask bits are valid; other bits are "don't care".

When address-mask is not specified, all address mask bits are valid. Automatic variables coded in C/C++ language cannot be set.

pass-count (default: decimal number)

Specifies the number of times events generated (1 to 255).

When pass-count is not specified, the number of times of events generated is set to 1.

Command qualifier

/BREAKCONDITION (Only emulator debugger (MB2198) DSU4)

The conditions set in SET BREAKCONDITION are combined into event conditions.

- Description

The SET CODEEVENT command sets a code event. Only two code events can be set. The event for code break is set.

It can set an address, address mask, and pass count value.

For the DSU3 chip, the code event can be used for trace measurement-start factor in addition to program stop factor. This switching can be performed by the SET TRACE command.

- Example

```
>SET CODEEVENT 1, function
>SET CODEEVENT 2, loop, 3
```

Note:

[Emulator Debugger (MB2198)]

This function cannot be used only when the FR60Lite is used.

For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.21 SHOW CODEEVENT

The **SHOW CODEEVENT** command displays the setting data of the specified event.

■ SHOW CODEEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	×
Monitor		×

- Format

SHOW CODEEVENT [event-number [, . . .]]

Parameter

event-number

Specifies the event number (D'1 or D'2).

Command qualifier

/ALL (default when omitted)

Displays all code events.

- Description

The SHOW CODEEVENT command displays the setting data of the specified event in the following :

no. : Event number
en/dis : Enable or disable
addr : Indicates address in hexadecimal notation
mask : Indicates address mask data
pass : Indicates pass count in decimal notation
symbol : Indicates symbol or line number corresponding to address

- Example

```
>SHOW CODEEVENT/ALL
no.  en/dis  addr      mask      pass      symbol
1    enable  0000FF00  FFFFFFFF  1 ( 1)    \function
2    enable  0000EFF0  FFFFFFFF  5 ( 0)
```

Notes:

- The hit count of code event is not update while running user program. Therefore, the value of hit count indicated while running user program is the one before the program running starts.
- The hit count of code event is affected by a prefetch by the MCU.

[Emulator Debugger (MB2198)]

This function cannot be used only when the FR60Lite is used. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.22 CANCEL CODEEVENT

The **CANCEL CODEEVENT** command cancels the specified event.

■ CANCEL CODEEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

CANCEL CODEEVENT [event-number [, . . .]]

Parameter

event-number

Specifies the event number (D'1 or D'2).

Command qualifier

/ALL

Cancels all code events.

- Description

The **CANCEL CODEEVENT** command cancels the specified event.

- Example

```
>CANCEL CODEEVENT 1
>CANCEL CODEEVENT/ALL
```

Note:

[Emulator Debugger (MB2198)]

This function cannot be used only when the FR60Lite is used. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.23 ENABLE CODEEVENT

The **ENABLE CODEEVENT** command enables the specified event.

■ ENABLE CODEEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

ENABLE CODEEVENT [event-number [, . . .]]

Parameter

event-number

Specifies the event number (D'1 or D'2).

Command qualifier

/ALL

Enables all code events.

- Description

The **ENABLE CODEEVENT** command enables the specified event.

- Example

```
>ENABLE CODEEVENT 2
>ENABLE CODEEVENT/ALL
```

Note:

[Emulator Debugger (MB2198)]

This function cannot be used only when the FR60Lite is used. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.24 DISABLE CODEEVENT

The **DISABLE CODEEVENT** command disables the specified event.

■ DISABLE CODEEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

DISABLE CODEEVENT [event-number [, . . .]]

Parameter

event-number

Specifies the event number (D'1 or D'2).

Command qualifier

/ALL

Disables all code events.

- Description

The **DISABLE CODEEVENT** command disables the specified event.

- Example

```
>DISABLE CODEEVENT 2
>DISABLE CODEEVENT/ALL
```

Note:

[Emulator Debugger (MB2198)]

This function cannot be used only when the FR60Lite is used. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.25 SET DATAEVENT

The SET DATAEVENT command sets a data event.

■ SET DATAEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

SET DATAEVENT event-number, address [& = address-mask] [, [!] D = data [& = data-mask]]

Parameters

event-number

Specifies the event number (D'1 or D'2).

address [&=address-mask] (address formula, data formula)

Specifies the address in which event generation condition to be stored.

When address-mask is specified, only 1 address mask bits are valid; other bits are "don't care".

When address-mask is not specified, all address mask bits are valid. Automatic variables coded in C language cannot be set.

[!] D=data [&=data-mask] (data formula, data formula)

Specifies the data (32 bits) to be set as event generation condition.

When data-mask is specified, only 1 data mask bits are valid; other bits are "don't care".

When data-mask is not specified, all the bits are valid.

When " ! " is specified, the specified data and NOT specifying of data-mask are valid (If NOT is specified, the event generation condition is set when the bit pattern specified in data-mask does not match that specified in data).

Depending on the chip to be used, data and data-mask may not be specifiable.

Command qualifiers

/READ

Specifies event generation condition when specified address read-accessed.

/WRITE

Specifies event generation condition when specified address write-accessed.

If /READ and /WRITE are omitted, /READ/WRITE is assumed.

/BYTE (default when omitted)

Specifies event generation condition when specified address accessed 1 byte.

/HALFWORD

Specifies event generation condition when specified address accessed 2 bytes.

/WORD

Specifies event generation condition when specified address accessed 4 bytes.

/BREAKCONDITION (Only emulator debugger (MB2198) DSU4)

The conditions set in SET BREAKCONDITION are combined into event conditions.

- **Description**

The SET DATAEVENT command sets a data event. Only two data events can be set.

If no command identifier is specified, READ/WRITE and /BYTE are assumed. /READ/WRITE specifies that the event generation condition is to be set irrespective of read or write access. /BYTE specifies the event generation condition when the specified address is accessed 1 byte.

If the data value to be used as the event generation condition is specified, address-mask is disabled.

For the DSU3 chip, the data event can be used for trace measurement-start factor in addition to program stop factor. This switching can be performed by the SET TRACE command.

- **Example**

```
>SET DATAEVENT 1, flag
>SET DATAEVENT/READ 2, data
```

Note:

[Emulator Debugger (MB2198)]

This function cannot be used only when the FR60Lite is used. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.26 SHOW DATAEVENT

The **SHOW DATAEVENT** command displays setting data of the specified event.

■ SHOW DATAEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	×
Monitor		×

- Format

SHOW DATAEVENT [event-number [, . . .]]

Parameter

event-number

Specifies the event number (D'1 or D'2).

Command qualifier

/ALL (default when omitted)

Displays all dataevents.

- Description

The SHOW DATAEVENT command displays the setting data of the specified event in the following :

```

no.      : Event number
en/dis   : Enable or disable
status   : Indicates access attribute (R: Read, W: Write)
addr     : Indicates address in hexadecimal notation
addr_msk : Indicates address mask data
data     : Data
data_msk : Indicates data mask data
size     : Indicates data size
symbol   : Indicates symbol corresponding to address

```

- Example

```

>SHOW DATAEVENT/ALL
no.en/dis status addr      addr_msk  data      data_msk size symbol
1  enable  R/W  00500000  -----  0000000A FFFFFFFF word \R_Tpri_Lev
2  enable  R/W  00508038  -----  0051017C FFFFFFFF word \task0

```

Note:

[Emulator Debugger (MB2198)]

This function cannot be used only when the FR60Lite is used. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.27 CANCEL DATAEVENT

The CANCEL DATAEVENT command cancels the specified event.

■ CANCEL DATAEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

CANCEL DATAEVENT [event-number [, . . .]]

Parameter

event-number

Specifies the event number (D'1 or D'2).

Command qualifier

/ALL

Cancels all dataevents.

- Description

The CANCEL DATAEVENT command cancels the specified event.

- Example

```
>CANCEL DATAEVENT 1
>CANCEL DATAEVENT/ALL
```

Note:

[Emulator Debugger (MB2198)]

This function cannot be used only when the FR60Lite is used. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.28 ENABLE DATAEVENT

The **ENABLE DATAEVENT** command enables the specified event.

■ ENABLE DATAEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

ENABLE DATAEVENT [event-number [, . . .]]

Parameter

event-number

Specifies the event number (D'1 or D'2).

Command qualifier

/ALL

Enables all dataevents.

- Description

The **ENABLE DATAEVENT** command enables the specified event.

- Example

```
>ENABLE DATAEVENT 2
>ENABLE DATAEVENT/ALL
```

Note:

[Emulator Debugger (MB2198)]

This function cannot be used only when the FR60Lite is used. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.29 DISABLE DATAEVENT

The **DISABLE DATAEVENT** command disables the specified event.

■ DISABLE DATAEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

DISABLE DATAEVENT [event-number [, . . .]]

Parameter

event-number

Specifies the event number (D'1 or D'2).

Command qualifier

/ALL

Disables all dataevents.

- Description

The **DISABLE DATAEVENT** command disables the specified event.

- Example

```
>DISABLE DATAEVENT 2
>DISABLE DATAEVENT/ALL
```

Note:

[Emulator Debugger (MB2198)]

This function cannot be used only when the FR60Lite is used. For details, refer to "2.3.4 Break" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.30 SET SEQUENCE (type 1)

The SET SEQUENCE command sets a sequential operation mode for code and data events.

■ SET SEQUENCE (type 1)

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

SET SEQUENCE

Command qualifiers

/ON

Specifies the delay count start triggered when event 1 condition and event 2 condition established in this order.

/OFF

Specifies the delay count start triggered when event 1 condition or event 2 condition established.

/CODE

Specifies the processing of code events.

/DATA

Specifies the processing of data events.

/ALL

Specifies the processing of code events and data events.

- Description

The SET SEQUENCE command sets a sequential operation mode for code events and data events.

When /ON is specified, the SEQUENTIAL mode is set. In the SEQUENTIAL mode, when the event 1 condition and event 2 condition are established in this order, the delay count start is triggered.

When /OFF is specified, the OR mode is set. In the OR mode, when the event 1 condition or event 2 condition is established, the delay count start is triggered.

The sequential operation mode can be set individually for both the code and data events.

For the DSU3 chip, the code event and data event can be also used for trace measurement-start factor by the SET TRACE/TRIGGER command. In this case, the code event and data event are not a program stop factor, so this command cannot be used.

- Example

```
>SET SEQUENCE/ON
```

Note:

[Emulator Debugger (MB2198)]

This function cannot be used only when the FR60Lite is used. For details, refer to "2.3.5 Control by Sequencer" or "2.3.8 Measuring Performance" of "SOFTUNE WORKBENCH USER'S MANUAL".

3.31 SET SEQUENCE (type 2)

The SET SEQUENCE command sets a sequencer.

■ SET SEQUENCE (type 2)

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

[Emulator Debugger (MB2198)]

SET SEQUENCE event-number[, event-number[, event-number]][,R=event-number]

[Emulator Debugger (MB2100-01)]

SET SEQUENCE event-number[, event-number][,R=event-number]

Parameters

event-number (default: decimal number)

Specify the number of the event to be set as a trigger factor.

Specify within the range of 1 to 4 on the emulator debugger (MB2198) or 1 to 8 on the emulator debugger (MB2100-01).

R=event-number (default: decimal number)

Specify the numbers of the events that serve as the restarting conditions for the sequencer.

Specify within the range of 1 to 4 on the emulator debugger (MB2198) or 1 to 8 on the emulator debugger (MB2100-01).

- Description

The sequencer is set. The 3-level sequencer with RESTART is set on the emulator debugger (MB2198) and the 2-level sequencer with RESTART is set on the emulator debugger (MB2100-01).

Set the event point by SET EVENT beforehand.

The event number can be referred from the address by using the built-in function %EVENTNUM.

- Example

[Emulator Debugger (MB2198)]

```
>SET SEQUENCE 2,3,1,R=4
```

Notes:

- The same event number as the parameter cannot be specified two times or more.

[Emulator Debugger (MB2198)]

- This command can't use except the FR60Lite or FR80S. For details, refer to "2.3.5 Control by Sequencer" or "2.3.8 Measuring Performance" of "SOFTUNE WORKBENCH USER'S MANUAL".
- This function cannot be used when the performance mode is set as the event mode.

[Emulator Debugger (MB2100-01)]

- When the pass count mode is in the pass count break mode, this command cannot be used. For details, refer to "1.8 SET MODE (type 1)".
-

3.32 SHOW SEQUENCE

The **SHOW SEQUENCE** command displays the settings configured by the **SET SEQUENCE** command.

■ SHOW SEQUENCE

- Debugger

Simulator		X
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		X

- Format

SHOW SEQUENCE

- Description

[For the Emulator Debugger (MB2197 and MB2198 (Except for FR60Lite and FR80S))]

The SHOW SEQUENCE command displays the event sequential operation mode set by the SET SEQUENCE command.

For the DSU3 chip, the code event and data event can be also used for trace measurement-start factor by the SET TRACE/TRIGGER command. In this case, the code event and data event are not a program stop factor, so this command cannot be used.

[For the Emulator Debugger (MB2198 (FR60Lite, FR80S) and MB2100-01)]

Displays address information for the sequence configured using the SET SEQUENCE command.

- Example

[Emulator Debugger (MB2197 and MB2198 (Except for FR60Lite and FR80S))]

```
>SHOW SEQUENCE
code sequence mode = ON
data sequence mode = OFF
```

[Emulator Debugger (MB2198 (FR60Lite, FR80S), MB2100-01)]

```
>SHOW SEQUENCE
level1    -> level2    -> level3    -> end    restart
00FF0010  00FF0020  -----  -----
```

3.33 CANCEL SEQUENCE

The **CANCEL SEQUENCE** command cancels a sequencer.

■ CANCEL SEQUENCE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

CANCEL SEQUENCE

- Description

The settings of the sequencer specified by SET SEQUENCE are canceled all together. Individual event settings, however, are not deleted.

- Example

```
>CANCEL SEQUENCE
```

Notes:

[Emulator Debugger (MB2198)]

- This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.5 Control by Sequencer" of "SOFTUNE WORKBENCH USER'S MANUAL".
 - This function cannot be used when the performance mode is set as the event mode.
-

3.34 ENABLE SEQUENCE

The **ENABLE SEQUENCE** command enables a sequencer.

■ ENABLE SEQUENCE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

ENABLE SEQUENCE

- Description

The **ENABLE SEQUENCE** command enables a sequencer.

- Example

>ENABLE SEQUENCE

Notes:

[Emulator Debugger (MB2198)]

- This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.5 Control by Sequencer" of "SOFTUNE WORKBENCH USER'S MANUAL".
 - This function cannot be used when the performance mode is set as the event mode.
-

3.35 DISABLE SEQUENCE

The **DISABLE SEQUENCE** command disables a sequencer.

■ DISABLE SEQUENCE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

DISABLE SEQUENCE

- Description

The **DISABLE SEQUENCE** command disables a sequencer.

- Example

```
>DISABLE SEQUENCE
```

Notes:

[Emulator Debugger (MB2198)]

- This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.5 Control by Sequencer" of "SOFTUNE WORKBENCH USER'S MANUAL".
- This function cannot be used when the performance mode is set as the event mode.

3.36 SET TRIGGER

The **SET TRIGGER** command specifies whether to use the signal input from the TRIG pin as the break factor.

■ SET TRIGGER

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

SET TRIGGER

Command qualifiers

/BREAK

Uses signal input from TRIG pin as break factor.

/NOBREAK (default when omitted)

Does not use signal input from TRIG pin as break factor.

- Description

The SET TRIGGER command specifies whether to use the signal input from the TRIG pin as the break factor.

If a signal is input from the TRIG pin when /BREAK is specified, program execution is suspended.

- Example

```
>SET TRIGGER/BREAK
```

3.37 SHOW TRIGGER

The **SHOW TRIGGER** command displays whether the signal input from the TRIG pin is used as the break factor.

■ SHOW TRIGGER

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	×
Monitor		×

- Format

SHOW TRIGGER

- Description

The SHOW TRIGGER command displays whether the signal input from the TRIG pin is used as the break factor.

- Example

```
>SHOW TRIGGER
trigger = No Break
>
>SET TRIGGER/BREAK
>SHOW TRIGGER
trigger = Break
```

3.38 ENABLE ALIGNMENTBREAK

The **ENABLE ALIGNMENTBREAK** command enables alignment error break.

■ ENABLE ALIGNMENTBREAK

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

ENABLE ALIGNMENTBREAK

Command qualifiers

/CODE

Enables alignment error break by means of code fetch (access).

/DATA

Enables alignment error break by means of data access.

If no command qualifier is specified, /CODE/DATA is assumed.

- Description

The **ENABLE ALIGNMENTBREAK** command enables alignment error break.

This break can be set individually for both code access and data access.

- Example

```
>ENABLE ALIGNMENTBREAK/DATA
```

3.39 DISABLE ALIGNMENTBREAK

The **DISABLE ALIGNMENTBREAK** command disables alignment error break.

■ DISABLE ALIGNMENTBREAK

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

DISABLE ALIGNMENTBREAK

Command qualifiers

/CODE

Disables alignment error break by means of code fetch (access).

/DATA

Disables alignment error break by means of data access.

If no command qualifier is specified, /CODE/DATA is assumed.

- Description

The **DISABLE ALIGNMENTBREAK** command disables alignment error break.

This break can be set individually for both code access and data access.

- Example

```
>DISABLE ALIGNMENTBREAK/DATA
```

3.40 SHOW ALIGNMENTBREAK

The **SHOW ALIGNMENTBREAK** command displays the alignment error break status.

■ SHOW ALIGNMENTBREAK

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	×
Monitor		×

- Format

SHOW ALIGNMENTBREAK

- Description

The SHOW ALIGNMENTBREAK command displays the alignment error break status.

- Example

```
>SHOW ALIGNMENTBREAK  
code = enable  
data = disable
```

3.41 SET BREAKCONDITION (type 1)

The SET BREAKCONDITION sets a data monitoring break (software).

■ SET BREAKCONDITION (type 1)

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

● Format

SET BREAKCONDITION [/ADDRESS] address, data

Parameters

address (address formula)

Specifies the address at which monitoring is performed.

data (data formula)

Specifies the data value that serves as the criteria for comparison for the data of the above address.

Function type command qualifiers

/ADDRESS

Specifies the address as data to be monitored.

Command qualifiers

/EQ (default at start-up)

A break occurs when the data of the specified address agree with the data value.

/NE

A break occurs when the contents of the specified address disagree with the data value.

/BYTE (default when omitted)

Specifies that data to be monitored is 8-bit length.

/HALFWORD

Specifies that data to be monitored is 16-bit length.

/WORD

Specifies that data to be monitored is 32-bit length.

- Description

This command sets a data monitoring break (software).

This command is used in combination with SET BREAK/BREAKCONDITION, SET CODEEVENT/BREAKCONDITION, or SET DATAEVENT/BREAKCONDITION.

Immediately before executing the instruction at the address specified by SET BREAK/BREAKCONDITION, SET CODEEVENT/BREAKCONDITION, or SET DATAEVENT/BREAKCONDITION, the program execution is temporarily stopped to cause a break when the data of the address specified by this command agreed or disagree with the specified data value (if the conditions are not satisfied, the program execution is restarted).

Up to one point can be set.

- Example

```
>SET BREAKCONDITION/ADDRESS/WORD/NE &tskl_stat, 0
```

Notes:

- When using the monitoring function, do not execute this command. The error message "Command error (MCU is busy)." appears when the monitoring function is used after setting the data monitoring break (software).
 - When "Setting breakpoint while running" is enabled, this command cannot be used. For details of "Setting breakpoint while running", refer to "4.7.2.3 Setting Debug Environment" of "SOFTUNE Workbench Operation Manual".
-

3.42 SET BREAKCONDITION (type 2)

The SET BREAKCONDITION sets a data monitoring break (software).

■ SET BREAKCONDITION (type 2)

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

● Format

SET BREAKCONDITION /REGISTERINDIRECT register-name, offset, data

Parameters

register-name

Evaluate the values of the register whose number is specified as an address at which monitoring is performed. R0 to R15 can be specified.

offset (default: decimal number)

Evaluate the specified offset value with the values of the above register as an address and specify the address at which monitoring is performed.

data (data formula)

Specifies the data value that serves as the criteria for comparison for the data of the above address.

Function type command qualifiers

/REGISTERINDIRECT

Specifies the address indicated by "values of register + offset" as data to be monitored.

Command qualifiers

/EQ (default at start-up)

A break occurs when the data of the specified address agree with the data value.

/NE

A break occurs when the data of the specified address disagree with the data value.

/BYTE (default when omitted)

Specifies that data to be monitored is 8-bit length.

/HALFWORD

Specifies that data to be monitored is 16-bit length.

/WORD

Specifies that data to be monitored is 32-bit length.

- Description

This command sets a data monitoring break (software).

This command is used in combination with SET BREAK/BREAKCONDITION, SET CODEEVENT/BREAKCONDITION, or SET DATAEVENT/BREAKCONDITION.

Immediately before executing the instruction at the address specified by SET BREAK/BREAKCONDITION, SET CODEEVENT/BREAKCONDITION, or SET DATAEVENT/BREAKCONDITION, the program execution is temporarily stopped to cause a break when the values of "register + offset" specified by this command agreed or disagree with the specified data value (if the conditions are not satisfied, the program execution is restarted).

Up to one point can be set.

- Example

```
>SET BREAKCONDITION/REGISTERINDIRECT/HALFWORD/NE R11, 58, 1354
```

Notes:

- When using the monitoring function, do not execute this command. The error message "Command error (MCU is busy)." appears when the monitoring function is used after setting the data monitoring break (software).
 - When "Setting breakpoint while running" is enabled, this command cannot be used. For details of "Setting breakpoint while running", refer to "4.7.2.3 Setting Debug Environment" of "SOFTUNE Workbench Operation Manual".
-

3.43 SHOW BREAKCONDITION

Setting information specified by the SET BREAKCONDITION command is displayed.

■ SHOW BREAKCONDITION

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	×
Monitor		×

- Format

SHOW BREAKCONDITION

- Description

Setting information specified by the SET BREAKCONDITION command is displayed.

- Example

```
>show breakcondition
addressing data      size  compare-condition
R7+8244      000000FF byte  ne(not-equal)
```

CHAPTER 4 Program Execution Analysis Commands



This chapter describes the Program Execution Analysis commands.

- 4.1 SET PERFORMANCE (type 1)
- 4.2 SET PERFORMANCE (type 2)
- 4.3 SET PERFORMANCE (type 3)
- 4.4 SHOW PERFORMANCE
- 4.5 CANCEL PERFORMANCE
- 4.6 CLEAR PERFORMANCE
- 4.7 SET COVERAGE
- 4.8 SHOW COVERAGE
- 4.9 CANCEL COVERAGE
- 4.10 CLEAR COVERAGE
- 4.11 SHOW CALLS
- 4.12 SHOW TIMER
- 4.13 CLEAR TIMER
- 4.14 SET TRACE (type 1)
- 4.15 SET TRACE (type 2)
- 4.16 SHOW TRACE (type 1)
- 4.17 SHOW TRACE (type 2)
- 4.18 CLEAR TRACE
- 4.19 ENABLE TRACE (type 1)
- 4.20 ENABLE TRACE (type 2)
- 4.21 DISABLE TRACE (type 1)
- 4.22 DISABLE TRACE (type 2)
- 4.23 SEARCH TRACE
- 4.24 SET DATATRACEAREA
- 4.25 SHOW DATATRACEAREA
- 4.26 CANCEL DATATRACEAREA
- 4.27 SHOW DETAILTRACE

- 4.28 SET TRACETRIGGER
- 4.29 SHOW TRACETRIGGER
- 4.30 CANCEL TRACETRIGGER
- 4.31 ENABLE TRACETRIGGER
- 4.32 DISABLE TRACETRIGGER
- 4.33 SET DELAY
- 4.34 SHOW DELAY
- 4.35 SET RAMCHECK
- 4.36 SHOW RAMCHECK
- 4.37 CANCEL RAMCHECK
- 4.38 ENABLE RAMCHECK
- 4.39 DISABLE RAMCHECK
- 4.40 SET SEMIHOSTING
- 4.41 SHOW SEMIHOSTING

4.1 SET PERFORMANCE (type 1)

The **SET PERFORMANCE** command sets the operation of a performance measurement buffer when it is full.

■ SET PERFORMANCE (type 1)

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

SET PERFORMANCE

Command qualifiers

- Specifies the buffer-full break

/BREAK (default when omitted)

A break occurs when the performance measurement buffer becomes full.

Cannot be specified on the emulator debugger (MB2100-01).

/NOBREAK

No break occurs even when the performance measurement buffer becomes full.

Cannot be specified on the emulator debugger (MB2100-01).

- Specifies the measurement mode

/TOTAL

Sets the measurement mode to accumulation measurement.

Cannot be specified on the emulator debugger (MB2198).

/FIRST

Sets the measurement mode to the first measurement.

Cannot be specified on the emulator debugger (MB2198).

- Description

The SET PERFORMANCE command sets the operation of a performance measurement buffer when it is full.

/BREAK can be specified to cause a break when a performance measurement buffer becomes full. The performance measurement buffer becomes full when an event occurs 65535 times.

This command setting is valid only when the event mode is set to PERFORMANCE.

The point at which performance is measured is set by the SET EVENT command. At performance measurement, the following items are measured:

Time measurement

The time between two events is measured in two sections. The starting and ending events are combined as follows:

Section 1: Starting event 1 - Ending event 2

Section 2: Starting event 3 - Ending event 4

Counting event occurrences

The time an event occurs is counted.

- Example

```
>SET PERFORMANCE/BREAK
```

Notes:

- This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.8 Measuring Performance" of "SOFTUNE WORKBENCH USER'S MANUAL".
 - This function cannot be used when the trace mode is set as the event mode.
-

4.2 SET PERFORMANCE (type 2)

The **SET PERFORMANCE** command sets the start/end trigger of the performance measurement.

■ SET PERFORMANCE (type 2)

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SET PERFORMANCE start event number, end event number

Parameters

start event number

Specifies the event number set as a trigger that starts the performance measurement.

end event number

Specifies the event number set as a trigger that ends the performance measurement.

Command qualifiers

/AREA

Sets an area for performance measurement.

- Description

The SET PERFORMANCE command sets the start/end trigger of the performance measurement.

Set the event previously by SET EVENT command before using this command. The specified event number can also be referenced from the address by using the built-in function %PFMEVENTNUM.

- Example

```
>SET PERFORMANCE /AREA 1, 2
```

Notes:

- When the execution time mode by set the SET MODE command is in the time measurement mode, this command cannot be used.
- The same event number as the parameter cannot be specified two times or more.

4.3 SET PERFORMANCE (type 3)

The SET PERFORMANCE command remeasures the performance.

■ SET PERFORMANCE (type 3)

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SET PERFORMANCE /RESTART

Command qualifiers

/RESTART

The SET PERFORMANCE command remeasures the performance.

- Description

The SET PERFORMANCE command remeasures the performance.

The remeasuring is done according to the following procedure.

1. Suspend performance measurement
2. Performance buffer clear
3. Start performance measurement

- Example

```
>SET PERFORMANCE /RESTART
```

Note:

This command cannot be used in the user program suspend.

4.4 SHOW PERFORMANCE

The **SHOW PERFORMANCE** command displays the state of the setting.

■ SHOW PERFORMANCE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		×

- Format

<Format 1>

SHOW PERFORMANCE/STATUS

<Format 2>

SHOW PERFORMANCE/COUNT [event-number]

<Format 3>

SHOW PERFORMANCE/TIME area-number [, lower limit, upper limit, display interval]

<Format 4>

SHOW PERFORMANCE/TIME

Command qualifiers

/STATUS

On the emulator debugger (MB2198), displays operation setting state of performance measurement buffer when full.

On the emulator debugger (MB2100-01), displays setting state of the performance measurement mode and the measurement area.

/COUNT

Displays count result of times that event occurs.

On the emulator debugger (MB2100-01) only, can be specified.

/TIME (default when omitted)

The performance measurement result is displayed.

Parameters (only Emulator Debugger (MB2198))

event-number

Specifies the number (1 to 4) of the event for displaying the setting contents.

When displaying the result of time measurement, the operation will be the same, even if the number of either the starting or ending event in the measurement section is specified.

When event-number is omitted, all events are displayed.

area-number

Specifies an area number (1 or 2) to display the result of its time measurement.

lower limit (default: decimal number)

Specifies the lower limit at which the measured time is displayed graphically. The unit is 1 ns.

upper limit (default: decimal number)

Specifies the upper limit at which the measured time is displayed graphically. The unit is 1 ns.

display interval (default: decimal number)

Specifies the interval at which the measured time is displayed graphically. The unit is 1 ns.

● Description

The SHOW PERFORMANCE command displays the state of performance settings.

- When /STATUS specified

[Emulator Debugger (MB2198)]

The SHOW PERFORMANCE command displays the setting state of the operation when a performance measurement buffer becomes the buffer full.

[Emulator Debugger (MB2100-01)]

The SHOW PERFORMANCE command displays the measurement mode and the measurement area set by the SHOW PERFORMANCE command.

- When /COUNT specified

The SHOW PERFORMANCE command displays the result of counting the time an event occurs.

- When /TIME specified

[Emulator Debugger (MB2198)]

The SHOW PERFORMANCE command displays the result of time measurement. The upper limit, lower limit, and interval at which the measured time is displayed graphically can be specified.

[Emulator Debugger (MB2100-01)]

The performance measurement result is displayed.

In measuring, the measurement count is updated only.

After the measurement ends, measurement count is updated.

The setting state of SET TIMERSCALE command is applied for the measurement unit.

The display results are as follows:

- When cycle count is performed

Pass Count: <number of counts>

<CPU Cycle>

Min: <minimum number of execution cycles> [Cycle] ^(*1)

Max: <maximum number of execution cycles> [Cycle] ^(*1)

Average: <average number of measurement cycles> ^(*4) [Cycle]

<Convert from CPU Cycle @<execution time cycle conversion frequency> Hz> ^(*2)

Min: <minimum number of execution cycles × execution cycle conversion frequency>

[Time@<execution time cycle conversion frequency> Hz] ^(*3)

Max: <maximum number of execution cycles × execution cycle conversion frequency>

[Time@<execution time cycle conversion frequency> Hz] ^(*3)

Average: <average number of measurement cycles × execution cycle conversion

frequency> ^(*4) [Time@<execution time cycle conversion frequency> Hz] ^(*2)

- When count in real time is performed

Pass Count: <number of counts>

<RealTime>

Min: <minimum execution time> [Time]

Max: <maximum execution time> [Time]

Average: <average measurement time><error display ^{*4}> [Time]

*1: This is displayed when using products which can measure the maximum/minimum values while measuring cycle count.

For details, refer to "On-chip debugger (OCD)" in the hardware manual for the product type you are using.

*2: This is displayed only when the execution cycle conversion function is enabled.

*3: This is displayed only when the conditions of both *1 and *2 are satisfied.

*4: One of errors below is displayed.

Abort: This indicates that the count start and stop events occur simultaneously.
In this case, "-----" is displayed for the count result.

Including invalid data: This indicates that counting is performed.

● Example

[Emulator Debugger (MB2198)]

```
>SHOW PERFORMANCE /COUNT
```

[Emulator Debugger (MB2100-01)]

When using /STATUS

<Accumulation measurement>

```
> SHOW PERFORMANCE /STATUS
```

```
measurement mode : total
```

```
measurement area : 1 -> 2
```

<first measurement>

```
> SHOW PERFORMANCE /STATUS
```

```
measurement mode : first
```

```
measurement area : 1 -> 2
```

When using /TIME

```
> SHOW PERFORMANCE /TIME
```

```
Total Cycle : 5158
```

```
Pass Count : 243
```

```
Average : 21
```

When using products which can measure the maximum/minimum values while measuring cycle count

When using /TIME

```
>SHOW PERFORMACE /TIME
Pass Count : 243
<CPU cycle>
Min          : 5[Cycle]
Max          : 35[Cycle]
Average      : 21[Cycle]
<Convert from CPU cycle@32MHz>
Min          : 12.340[us@32MHz]
Max          : 456.789[us@32MHz]
Average      : 123.450[us@32MHz]
```

When the start event and the end event of measurement occur simultaneously

```
> SHOW PERFORMANCE
Pass Count : 10
<CPU Cycle>
Min: 100 [Cycle]
Max: 1000 [Cycle]
Average: 500 (Abort)
```

When performing measurement

```
> SHOW PERFORMANCE
Pass Count : 10
<CPU Cycle>
Min: 100 [Cycle]
Max: 1000 [Cycle]
Average: 500 (Including invalid data)
```

When the minimum value overflowed

```
> SHOW PERFORMANCE
Min : Overflow [Cycle]
```

When the maximum value overflowed

```
> SHOW PERFORMANCE
Max : Overflow [Cycle]
```

When the performance timer overflowed

```
> SHOW PERFORMANCE
Average : Overflow [Cycle]
```

When the measurement mode is set to the first measurement

```
>SHOW PERFORMACE
Pass Count : 5
<CPU cycle>
Min          : -----[Cycle]
Max          : -----[Cycle]
Average      : 12[Cycle]
```

When using products which cannot measure the maximum/minimum values while measuring cycle count

When using /TIME

```
>SHOW PERFORMACE /TIME
Pass Count : 243
<CPU cycle>
Min          : -----[Cycle]
Max          : -----[Cycle]
Average      : 21[Cycle]
```

Notes:

[Emulator Debugger (MB2198)]

- This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.8 Measuring Performance" of "SOFTUNE WORKBENCH USER'S MANUAL".
- This function cannot be used when the trace mode is set as the event mode.
- When the upper bound, the lower bound, and the display interval are omitted, the best display interval is calculated and displayed based on the result of a measurement.

[Emulator Debugger (MB2100-01)]

- When the execution time mode is in the time measurement mode, this command cannot be used. For details, refer to "1.8 SET MODE (type 1)".
- When command qualifiers/TIME is specified, the error margin of about 10 cycles is usually caused in measurement cycle number in the following cases.

Condition of error margin: When the user program execution is suspended or is re-executed during the timer of the performance execution.

For details, refer to "2.4.7.2 Measuring Event-to-Event Execution Cycle Count [Performance Measurement]" of "SOFTUNE WORKBENCH USER'S MANUAL".

- When setting the measurement unit to the reference frequency of high communication speed, the measurement is not performed in normal communication mode. A hyphen ("-") will be displayed instead of the value measured. For details of the measurement unit, refer to "4.4.14.2 Performance (Emulator Debugger [MB2100-01])" of "SOFTUNE Workbench Operation Manual".
 - If the mode is switched to normal communication mode even once while the target program is running when the measurement unit is set to the reference frequency of high communication speed, an incorrect measurement value will be displayed because the emulator debugger cannot detect the communication mode switching.
-

4.5 CANCEL PERFORMANCE

The **CANCEL PERFORMANCE** command clears the start/end trigger of a performance measurement set by the **SET PERFORMANCE** command.

■ CANCEL PERFORMANCE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

CANCEL PERFORMANCE

- Description

The CANCEL PERFORMANCE command clears the start/end trigger of a performance measurement set by the SET PERFORMANCE command. The setting of the event set as start/end trigger remains.

- Example

```
>CANCEL PERFORMANCE
```

Note:

When this command is executed, the measurement result is cleared.

4.6 CLEAR PERFORMANCE

The **CLEAR PERFORMANCE** command clears the performance measurement values.

■ CLEAR PERFORMANCE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

CLEAR PERFORMANCE

- Description

The CLEAR PERFORMANCE command clears the performance measurement values.

- Example

```
>CLEAR PERFORMANCE
```

Notes:

[Emulator Debugger (MB2198)]

- This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.8 Measuring Performance" of "SOFTUNE WORKBENCH USER'S MANUAL".
- This function cannot be used when the trace mode is set as the event mode.

[Emulator Debugger (MB2100-01)]

- When the execution time mode is in the time measurement mode, this command cannot be used. For details, refer to "1.8 SET MODE (type 1)".
- While executing the user program, this command can be specified only when the measurement is terminated.

4.7 SET COVERAGE

The **SET COVERAGE** command specifies the coverage measurement area. Up to 32 areas can be set.

■ SET COVERAGE

- Debugger

Simulator		○*
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

*: It is possible to use only for the high-speed version simulator debugger.

- Format

SET COVERAGE [measurement range]

Parameter

measurement range (address formula)

Specifies the coverage measurement area.

If /AUTOMATIC is given, this parameter cannot be specified.

Command qualifier

/AUTOMATIC (default when omitted)

Automatically sets code area of currently-loaded module.

The C compiler library area is not set.

- Description

The SET COVERAGE command specifies the coverage measurement area. Up to 32 areas can be set.

- Example

```
>SET COVERAGE FE000000..FFFFFFFF
```


4.8 SHOW COVERAGE

The **SHOW COVERAGE** command displays the result of coverage measurement in a specified measurement range.

■ SHOW COVERAGE

- Debugger

Simulator		○*
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

*: It is possible to use only for the high-speed version simulator debugger.

- Format

<Format 1>

SHOW COVERAGE [/STATUS]

<Format 2>

SHOW COVERAGE [/GENERAL] [measurement range]

<Format 3>

SHOW COVERAGE [/TOTAL | /DETAIL] [measurement range]

<Format 4>

SHOW COVERAGE /SOURCE [{file-name} line number [..line-number] | {address | address-range}]

<Format 5>

SHOW COVERAGE /INSTRUCTION [{address | address-range}]

<Format 6>

SHOW COVERAGE /MODULE [{source-file-name | coverage-range} [, number-of-columns]]

Parameters

measurement range (address formula)

Specifies the coverage measurement area.

If this parameter is omitted, this command displays the area in order, from the first.

file-name

Specifies a name of source file to display a coverage measurement result.

When the file name is omitted, the previously-specified file name is used.

line-number

Specifies a line number of source to display a coverage measurement result.

"\$" must proceed a line number.

When line number is delimited by "..", the source within the specified range is displayed.

When the end line number is not specified, the result is displayed by 19 lines.

address (address formula)

Specifies a memory location of a code attribute.

Specifies this parameter to display a coverage measurement result corresponding to the memory location.

When a formula is specified, the coverage measurement result within the address range of the formula is displayed.

When other than a formula is specified, the result is displayed by 19 lines.

address-range (address formula)

Specifies a memory area of a code attribute.

Specifies this parameter to display a coverage measurement result corresponding to the memory location.

source-file-name

Specifies a name of source file to display a coverage rate.

If this parameter is omitted, the coverage rate of entire load module is displayed.

coverage-range

Specifies a range of coverage rate to be displayed.

If this parameter is omitted, the entire range is displayed.

number-of-columns

Specifies a column position for a coverage rate (number of characters from the beginning of a line).

When this parameter is omitted, the number of columns is 40.

Command qualifiers**/STATUS**

Displays enabled/disabled state of coverage measurement function and coverage measurement area.

/TOTAL

Displays coverage rate in entire specified measurement range.

/GENERAL (default when omitted)

Displays result of coverage measurement in 16 addresses.

/DETAIL

Displays result of coverage measurement in addresses.

/SOURCE

Displays a coverage measurement result in source lines.

/INSTRUCTION

Displays a coverage measurement result in machine instructions.

When SET SOURCE is set to the mode to add a source line and the memory location corresponds to the source line, this command also displays that source line.

/MODULE

Displays the coverage rate of the load module.

- Description

The SHOW COVERAGE command displays the result of coverage measurement in a specified measurement range. If a command qualifier is omitted, the operation assumes a previous qualifier is specified.

If /GENERAL is specified, the access count will be displayed as follows:

```
.      : No access
1 to F : Count of addresses accessed out of 16 addresses
*      : 16 addresses accessed
```

If /DETAIL is specified, the access count will be displayed as follows:

```
.      : No access
-      : Accessed
```

/SOURCE,/INSTRUCTION is specified, the access count will be displayed as follows:

```
.      : No access
*      : Accessed
```

Blank : Not generate the code or line outside measurement range

The coverage rate outside the all coverage measurement range is displayed as "(--%)", if specify /MODULE. The asterisk "*" is displayed next the coverage rate, when a part of the area is outside the coverage measurement range.

- Example

```
>SHOW COVERAGE
      (HEX) 0X0          +1X0          +2X0          -----
      +-----+-----+-----+-----+
address  0123456789ABCDEF0123456789ABCDEF0123456 .. ABCDEF  C0 (%)
FF000000  **3*F*.....                                32.0

>SHOW COVERAGE/SOURCE
*   70: {
      71:   int    i;
      72:   struct table *value[16];
      73:
*   74:   for (i=0; i<16; i++)
*   75:       value[i] = &target[i];
      76:
*   77:   sort_val(value, 16L);
      78: }

>SHOW COVERAGE/MODULE
sample.abs ..... (84.03%)
+- startup.asm ..... (90.43%)
+- sample.c ..... (95.17%)
+- samp.c ..... (100.00%)
```

4.9 CANCEL COVERAGE

The **CANCEL COVERAGE** cancels the coverage measurement area and disables the coverage measurement function.

■ CANCEL COVERAGE

- Debugger

Simulator		○*
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

*: It is possible to use only for the high-speed version simulator debugger.

- Format

CANCEL COVERAGE [measurement range]

Parameter

measurement range (address formula)

Specifies the measurement range to be deleted.

Command qualifier

/ALL

Deletes all coverage measurement areas.

- Description

The **CANCEL COVERAGE** cancels the coverage measurement area and disables the coverage measurement function.

- Example

```
>CANCEL COVERAGE/ALL
```

4.10 CLEAR COVERAGE

The **CLEAR COVERAGE** clears a coverage buffer.

■ CLEAR COVERAGE

- Debugger

Simulator		○*
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

*: It is possible to use only for the high-speed version simulator debugger.

- Format

CLEAR COVERAGE

- Description

The **CLEAR COVERAGE** clears a coverage buffer.

- Example

```
>CLEAR COVERAGE
```

4.11 SHOW CALLS

The **SHOW CALLS** command displays the calling history until current function.

■ SHOW CALLS

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

● Format

SHOW CALLS [call-frame-count]

Parameter

call-frame-count (default: decimal number)

Specifies count of call frames requiring information (D'1 to D'256).

● Description

The SHOW CALLS command displays the calling history until current function.

When call-frame-count is not specified, the command displays up to 256 frames.

When the function to be displayed contains an argument, the command displays the argument as a hexadecimal number.

If there is no C/C++ language debug information, the command displays the function address.

The command analyzes accumulated stack data and determines which data to display according to the analysis result. It analyzes accumulated stack data according to the stack format used when C/C++ language function called.

The following programs can not be displayed correctly, please note.

- The command cannot be used in the programs coded in assembler.
- In the optimized program, the command may be unable to display data normally.
- If the program is not compiled with debug information, the command displays the address instead of the function name. However, if the program breaks at the beginning of the function, the command cannot display data normally.

● Example

```
>SHOW CALLS
checker (12, 8)
main (3, 4)
```

4.12 SHOW TIMER

The **SHOW TIMER** command displays the instruction execution cycle count, step count, and time of the executed program in decimal notation.

■ SHOW TIMER

- Debugger

Simulator		○
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	○
Monitor		○

- Format

SHOW TIMER

- Description

The SHOW TIMER command displays the instruction execution cycle count, step count and time of the executed program in decimal notation.

The numbers displayed are those after the RESET command has been executed, and those after the execution has been started by the previous GO, STEP or CALL command.

The displayed contents vary depending on the debugger.

[Simulator Debugger]

Refer to Section "2.1.9 Measuring Execution Cycle Count" in "SOFTUNE Workbench User's Manual".

[Emulator Debugger (MB2197)]

Refer to Section "2.2.5 Measuring Execution Cycle Count" in "SOFTUNE Workbench User's Manual".

[Emulator Debugger (MB2198)]

Refer to Section "2.3.6 Measuring Execution Cycle Count" in "SOFTUNE Workbench User's Manual".

[Emulator Debugger (MB2100-01)]

- When cycle count is performed (the execution cycle conversion function is disabled)

<cycle> From Initialize: <number of execution cycles> [Cycle]
 From Last Executed: <cumulative number of execution cycles> [Cycle]

- When cycle count is performed (the execution cycle conversion function is enabled)

<cycle> From Initialize: <number of execution cycles> [Cycle]
 From Last Executed: <cumulative number of execution cycles> [Cycle]

<timer @<execution cycle conversion frequency> Hz>

From Initialize: <number of execution cycles × execution cycle conversion frequency> [Time@<execution cycle conversion frequency> Hz]

From Last Executed: <cumulative number of execution cycles × execution cycle conversion frequency> [Time@<execution cycle conversion frequency> Hz]

- When count in real time is performed

```
<timer> From Initialize:      <number of execution cycles> [Time]
        From Last Executed:  <cumulative number of execution cycles> [Time]
```

● Example

[Emulator Debugger (MB2198)]

```
>SHOW TIMER
<cycle>      From Initialize:      4210826410[Cycle]
              From Last Executed:  362387415[Cycle]
<timer>      From Initialize:      0h00m42s108ms264us250ns[Time]
              From Last Executed:  0h00m03s623ms874us375ns[Time]
```

[Emulator Debugger (MB2100-01)]

<When the measurement unit is CPU clock>

```
>SHOW TIMER
<cycle> From Initialize:      4210826410[Cycle]
        From Last Executed:  362387415[Cycle]
<timer @1MHz>
From Initialize:      0h00m42s108ms264us250ns[Time@1MHz]
From Last Executed:  0h00m03s623ms874us375ns[Time@1MHz]
```

<When the measurement unit is not CPU clock>

```
>SHOW TIMER
<timer> From Initialize:      0h00m42s108ms264us250ns[Time]
        From Last Executed:  0h00m03s623ms874us375ns[Time]
```

Note:

[Emulator Debugger]

Several cycles of errors occur with one execution. It is recommended to execute many at one time to the degree that errors are ignored.

[Simulator Debugger]

Pipeline execution times are not simulated.

[Emulator Debugger (MB2100-01)]

- Approx. 10 cycles of error occurs for the measured cycle counts normally, however, the error may be greater due to the state of bus.
 - When the execution time mode is in the time measurement mode, this command cannot be used. For details, refer to "1.8 SET MODE (type 1)".
 - When both of the following conditions are satisfied, this command cannot be used.
 - When the communication speed during debugging is normal
 - When setting the measurement unit to the reference frequency of high communication speed
 - When setting the measurement unit to the reference frequency of high communication speed, the measurement is not performed in normal communication mode. A hyphen ("-") will be displayed instead of the value measured. For details of the measurement unit, refer to "4.4.14.2 Performance (Emulator Debugger [MB2100-01])" of "SOFTUNE Workbench Operation Manual".
 - If the mode is switched to normal communication mode even once while the target program is running when the measurement unit is set to the reference frequency of high communication speed, an incorrect measurement value will be displayed because the emulator debugger cannot detect the communication mode switching.
-

4.13 CLEAR TIMER

The **CLEAR TIMER** command initializes the timer measurement execution result.

■ CLEAR TIMER

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

- Format

CLEAR TIMER

- Description

The CLEAR TIMER command initializes executed number of instruction execution cycles, number of steps of programs, and the result of a measurement at time are.

- Example

```
>CLEAR TIMER
```

Note:

[Emulator Debugger (MB2100-01)]

- When the execution time mode is in the time measurement mode, this command cannot be used. For details, refer to "1.8 SET MODE (type 1)".
-

4.14 SET TRACE (type 1)

The SET TRACE (type 1) command configures the trace buffer full break.

■ SET TRACE (type 1)

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	×
Monitor		×

- Format

SET TRACE

Command qualifiers

/BREAK

Enables trace buffer-full break.

/NOBREAK (default at start-up)

Disables trace buffer-full break.

- Description

Enabling the trace buffer-full break, suspends program execution when the trace buffer becomes full.

- Example

```
>SET TRACE/BREAK
```

4.15 SET TRACE (type 2)

The **SET TRACE (type 2)** command Configures a break when the trace buffer is full or the trace capture has finished.

■ SET TRACE (type 2)

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

SET TRACE

- Command qualifiers

Trace buffer-full break specified

/BREAK

The program is suspended when the trace buffer is full.

On the emulator debugger (MB2100-01), breaks program execution when the trace capture has finished. Specify this after /ENDTRACE.

/NOBREAK (default at start-up)

The program is not suspended when the trace buffer is full. Trace measurements are continued.

On the emulator debugger (MB2100-01), continues program execution after the trace capture has finished. Specify this after /ENDTRACE.

Trace sampling mode specified

/FULL (default at start-up)

Measures the trace from program execution start to stop.

Cannot be specified on the emulator debugger (MB2198 (FR60Lite and FR80S) or MB2100-01).

/TRIGGER

The code event, data event and trace trigger are used as the causes of trace control.

Cannot be specified on the emulator debugger (MB2198 (FR60Lite and FR80S) or MB2100-01).

[Emulator Debugger (MB2198 (FR60Lite, FR80S))]

Trace capture on trace buffer full specification

/STOP

Does not break program execution when the trace buffer is full.

When the buffer is full, trace measurement is stopped.

/NOSTOP

Does not break program execution when the trace buffer is full.

When the buffer is full, trace measurement continues.

[Emulator Debugger (MB2100-01)]

Operation specification when trace capture finished

/ENDTRACE

Specifies the operation when the trace capture has finished. Specify this together with /BREAK or /NOBREAK. /ENDTRACE cannot be specified by itself.

- Description

[Emulator Debugger (MB2198)]

Enabling the trace buffer-full break, suspends program execution when the trace buffer becomes full.

DSU3/DSU4 chip can associate code event or data event with trace measurement. The code event or data event can be associated with the trace measurement in either of the following two modes.

- Full mode
 - The trace from program execution start to stop is measured. Code event and data event are used for execution stop factors of program.
- Trigger mode
 - The trace measurement is not started at starting a program execution. The trace from code event or data event detection to program stop is measured. The trace sampling mode for code event and data event should not be individual-specified.

[Emulator Debugger (MB2100-01)]

Specifies whether or not to break program execution when the trace capture has finished.

- Example

[Emulator Debugger (MB2198)]

```
>SET TRACE/TRIGGER /BREAK
```

[Emulator Debugger (MB2100-01)]

```
>SET TRACE /ENDTRACE /BREAK
```

Note:

[Emulator Debugger (MB2100-01)]

This command can only be used while a user program is executing if the trace capture has finished.

4.16 SHOW TRACE (type 1)

The **SHOW TRACE** command displays the trace data stored in the trace buffer.

■ SHOW TRACE (type 1)

- Debugger

Simulator		○
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW TRACE [/DATA] [trace-number [.. trace-number]]

SHOW TRACE /FILE [/APPEND] [file-name]

- Command qualifiers classified by function

/DATA (default when omitted)

Displays traced data.

- Parameters

trace-number (default: decimal number)

Specify the number of trace data to be displayed with decimal number.

The oldest trace data or the following trace data of the trace number displayed at the end of last time is displayed. When only the display start trace number is specified, as many as 12 traces are displayed from the display start trace number. When the display start trace number is smaller than the trace number of the oldest trace data, the oldest trace data is displayed in the beginning.

file-name

The file that saves the displayed trace data is specified.

If omitted, the file is saved as "TRACE.LOG".

- Command qualifiers

Trace capture data specification (only DSU1 or DSU2)

/CYCLE (default when omitted)

Displays trace data without analyzing the trace data.

/INSTRUCTION

Displays the trace data by the disassemble form.

/SOURCE

Displays the trace data by the source lines form.

Trace capture size specification

/ONEFRAME

Displays trace data only by one line.

Trace data storage method specification

/FILE

Save trace data to a file.

/APPEND

Add and save trace data to a file. Effective when "/FILE" is specified.

● Description

The SHOW TRACE command displays the trace data stored in the trace buffer.

Sampled trace data is assigned numbers. Trace data in the execution stop location (trigger point) is assigned number 0. The sampled trace data is assigned negative numbers until the execution stop location is reached. These numbers are called frame numbers.

The DSU3 or DSU4 chip, PC and data access information at branch are displayed.

In the emulator debugger (MB2198), the time stamp is displayed when the real time trace interface is provided.

● Example

```
> SHOW TRACE/DATA
```

frame no.	b-addr	b-cause	a-addr	a-stat	a-size	a-type	a-data	r-id	r-data
-00009:	-----	-----	-----	-----	-----	-----	-----	-	-----
-00008:	->00060248->	Go	-----	-----	-----	-----	-----	-	-----
-00007:	->00060224	Branch	FFFFFFFFC	read	word	cpu	00000003 3	FFFFFFFF	
-00006:	-----	-----	FFFFFFFFC	read	word	cpu	00000000 0	FFFFFFFF	
-00005:	-----	-----	FFFFFFFB8	write	word	cpu	0001021C -	-----	
-00004:	-----	-----	FFFFFFFFC	read	word	cpu	000001 1	FFFFFFFF	
-00003:	-----	-----	FFFFFFFFC	write	word	cpu	00000000 -	-----	
-00002:	-----	-----	FFFFFFFFC	read	word	cpu	00000002 2	FFFFFFFF	
-00001:	00060248->	-----	-----	-----	-----	-----	-----	-	-----
00000:	-----	Break	-----	-----	-----	-----	-----	-	-----

Note:

[Emulator Debugger (MB2197 or MB2198)]

In DSU3 or DSU4 chip, use SHOW DETAILTRACE command to display information on the instruction executed between branch instructions.

[Emulator Debugger (MB2100-01)]

This command can only be used while a user program is executing if the trace capture has finished.

4.17 SHOW TRACE (type 2)

The **SHOW TRACE** command displays the trace conditions.

■ SHOW TRACE (type 2)

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input checked="" type="radio"/>
Monitor		<input checked="" type="checkbox"/>

- Format

SHOW TRACE /STATUS

Command qualifier

/STATUS

Displays trace measurement conditions, enabled/disabled state of trace function, and storage status of trace buffer.

- Description

The SHOW TRACE command displays the trace conditions. Displays trace measurement conditions, enabled/disabled state of trace function, and storage status of trace buffer.

- Example

[Emulator Debugger (MB2197)]

```
>SHOW TRACE/STATUS
en/dis= enable
buffer full          = nobreak
frame no.            = -00120 to 00000
```

[Emulator Debugger (MB2100-01)]

```
>SHOW TRACE/STATUS
end trace            = nobreak
frame no.            = -00120 to 00000
```

Note:

[Emulator Debugger (MB2100-01)]

This command can only be used while a user program is executing if the trace capture has finished.

4.18 CLEAR TRACE

The **CLEAR TRACE** command clears the trace buffer.

■ CLEAR TRACE

- Debugger

Simulator		○
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

CLEAR TRACE

- Description

The CLEAR TRACE command clears the trace buffer.

- Example

>CLEAR TRACE

Note:

[Emulator Debugger (MB2100-01)]

This command can only be used while a user program is executing if the trace capture has finished.

4.19 ENABLE TRACE (type 1)

The **ENABLE TRACE** command enables the trace function.

■ ENABLE TRACE (type 1)

- Debugger

Simulator		○
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

ENABLE TRACE

- Description

The **ENABLE TRACE** command enables the trace function.

- Example

```
>ENABLE TRACE
```

4.20 ENABLE TRACE (type 2)

The **ENABLE TRACE** command forcefully starts the trace function.

■ ENABLE TRACE (type 2)

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

ENABLE TRACE

Command qualifiers

/FORCE

Specifies that the trace function is forcefully started.

- Description

The **ENABLE TRACE** command forcefully starts the trace function while the user program is executing.

- Example

```
>ENABLE TRACE /FORCE
```

Note:

If this command is input while the user program is stopped, the message "This command cannot be used while the MCU is stopped" is displayed.

4.21 DISABLE TRACE (type 1)

The **DISABLE TRACE** command disables the trace function.

■ DISABLE TRACE (type 1)

- Debugger

Simulator		○
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

DISABLE TRACE

- Description

The **DISABLE TRACE** command disables the trace function.

However, when MCU operation mode is set to internal trace mode or external trace mode (DSU3 chip only), the trace function cannot be disabled.

- Example

```
>DISABLE TRACE
```

4.22 DISABLE TRACE (type 2)

The **DISABLE TRACE** command forcefully stops the trace function.

■ DISABLE TRACE (type 2)

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

DISABLE TRACE

Command qualifiers

/FORCE

Specifies that the trace function is forcefully stopped.

- Description

The **DISABLE TRACE** command forcefully stops the trace function while the user program is executing.

- Example

```
>DISABLE TRACE /FORCE
```

Note:

If this command is input while the user program is stopped, the message "This command cannot be used while the MCU is stopped" is displayed.

4.23 SEARCH TRACE

The **SEARCH TRACE** command searches for trace data according to the specified condition.

■ SEARCH TRACE

- Debugger

Simulator		○
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

<Format 1>

SEARCH TRACE [address [& = mask-data]] [, f = search-start-number]

<Format 2>

SEARCH TRACE [d = data [& = mask-data]] [, f = search-start-number]

Parameters

address (address formula)

Specifies the address to be searched.

data (data formula)

Specifies the data transmission register access data to be searched.

This parameter is valid only when the debugger type is an emulator debugger (DSU3 of MB2197 or MB2198).

mask-data (data formula)

Specifies the masking and searching of address and data.

Only bits set to 1 are to be compared for search.

search-start-number (default: decimal number)

Specifies the search start frame number with decimal number.

When this parameter is omitted, the command starts data search from the beginning of the trace buffer.

Command qualifiers

/ALL (default when omitted)

Searches for all associated frames.

/ONEFRAME

Terminates trace data search when one frame found.

/CYCLE

Searches for trace data in units of valid bus cycles. Cannot be specified on the DSU3 chip of the emulator debugger (MB2197 or MB2198).

/INSTRUCTION

Searches for data that is rearranged in machine instruction execution units.

If trace data cannot be rearranged in machine instruction execution units, it is searched for in machine cycles.

/BYTE (DSU3 of the emulator debugger (MB2197))

Specifies that event condition data to be treated as 1-byte data.

Data length specifying is valid only for Format 2.

When this specifying is omitted, event condition data is a data length.

/HALFWORD (DSU3 of the emulator debugger (MB2197))

Specifies that event condition data to be treated as 2-byte data.

/WORD (DSU3 of the emulator debugger (MB2197))

Specifies that event condition data to be treated as 4-byte data.

/CODE (DSU3 of the emulator debugger (MB2197))

Searches trace frame or step where code access made to specified address.

/READ (DSU3 of the emulator debugger (MB2197))

Searches trace frame or step where read access made to specified address.

/WRITE (DSU3 of the emulator debugger (MB2197))

Searches trace frame or step where write access made to specified address.

● Description

The SEARCH TRACE command searches for trace data according to the specified condition.

When the trace data matching the condition is found, the command displays it in the same format as the SHOW TRACE command.

When /ONEFRAME is specified, the debugger terminates this command when one frame is found.

● Example

```
>SEARCH TRACE/INSTRUCTION 0xF0AE6
frame no.      address      mnemonic
-00010:        000F0AE6      ENTER      #004
-00009:        000F0AE8      LEAVE
-00008:        000F0AEA      LD          @R15+, RP
-00007:        000F0AEC      RET
-00006:        000F0ADE      LEAVE
-00005:        000F0AE0      LD          @R15+, RP
-00004:        000F0AE2      RET
-00003:        000F0ACE      LD          @ (R14,-4) , R4
-00002:        000F0AD0      LEAVE
-00001:        000F0AD2      LD          @R15+, RP
-00000:        000F0AD4      ADDSP      #4
```

Note:

[Emulator Debugger (MB2100-01)]

This command can only be used while a user program is executing if the trace capture has finished.

4.24 SET DATATRACEAREA

The **SET DATATRACEAREA** command specifies the data access area where trace measurement is performed.

■ SET DATATRACEAREA

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

SET DATATRACEAREA address [& = address-mask]

Parameters

address (address formula)

The address where to be done the trace measurement accessed data is specified. This parameter cannot be specified for emulator debugger (MB2100-01).

address-mask (data formula)

The address and the bit mask are taken off, and the address area of the data access that wants to do the trace measurement is specified. Only bits of "1" set are compared, and the address area is set. This parameter cannot be specified for emulator debugger (MB2100-01).

Command qualifiers

/READ (default when omitted)

Performs trace measurement upon read access to a specified address area.

The specified address area is all area for emulator debugger (MB2100-01).

/WRITE (default when omitted)

Performs trace measurement upon write access to a specified address area.

The specified address area is all area for emulator debugger (MB2100-01).

/CODE (default when omitted)

Performs trace measurement upon code access to a specified address area.

The specified address area is all area for emulator debugger (MB2100-01).

/DMA

Performs trace measurement upon DMA access to a specified address area.

In emulator debugger (MB2100-01), only when /READ or /WRITE is specified, this command qualifier is effective.

- Description

The command specifies the data access area where trace measurement is performed. The data access area is specified by checking only the bits set to 1s of the address mask.

If no command qualifier is specified, processing will be performed as /READ/WRITE/CODE.

If no address mask is specified, processing will be performed assuming the address mask as being H'FFFFFFFF.

- Example

```
>SET DATATRACEAREA/WRITE 180 &= 7f
```

Note:

[Emulator Debugger (MB2197 or MB2198)]

The specification for this command can be different depending on MCU mode and DSU type as follows.

MCU operation mode	DSU Type	
	DSU3	DSU4
Full trace mode	○	○
Real time mode	○	○
External trace mode	○	Adapter board connection : ◎
		cable connection : ○
Internal trace mode	×	×

◎ :Command is valid

Command qualifier "/CODE" can be specified.

○ :Command is valid

Command qualifier "/CODE" cannot be specified.

(Note) Always obtain a data for attribution of code.

× :Command is invalid

Always obtain the data for attribution of read/write/code for all areas.

[Emulator Debugger (MB2100-01)]

This command can only be used during user program execution if the trace capture has finished.

4.25 SHOW DATATRACEAREA

The **SHOW DATATRACEAREA** command displays the set data trace measurement area.

■ SHOW DATATRACEAREA

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW DATATRACEAREA

- Description

The SHOW DATATRACEAREA command displays the set data trace measurement area.

- Example

[Emulator Debugger (MB2197 or MB2198)]

```
>SHOW DATATRACEAREA
address      mask      access
00000000    00000000    read write code
```

[Emulator Debugger (MB2100-01)]

```
>SHOW DATATRACEAREA
address      mask      access
-----      -----      read code write
```

Note:

[Emulator Debugger (MB2197 or MB2198)]

This function cannot be used in internal trace mode, as the MCU operation mode.

4.26 CANCEL DATATRACEAREA

The **CANCEL DATATRACEAREA** command deletes the set data trace measurement area.

■ CANCEL DATATRACEAREA

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

CANCEL DATATRACEAREA

- Description

The **CANCEL DATATRACEAREA** command deletes the set data trace measurement area.

- Example

```
>SHOW DATATRACEAREA
address      mask      access
00001000    0001FFFF    read
>CANCEL DATATRACEAREA
>SHOW DATATRACEAREA
address      mask      access
00000000    00000000    read write code
```

Notes:

[Emulator Debugger (MB2197 or MB2198)]

This function cannot be used in internal trace mode, as the MCU operation mode.

[Emulator Debugger (MB2100-01)]

This command can only be used during user program execution if the trace capture has finished.

4.27 SHOW DETAILTRACE

The **SHOW DETAILTRACE** command displays details of traces.

■ SHOW DETAILTRACE

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW DETAILTRACE [frame-display-starting-line] [, f = frame-number]

Parameters

frame-display-starting-line (default: decimal number)

Specifies the count of lines on which the detailed display of a specified frame is started.

frame-number (default: decimal number)

Specifies the number of the frame to be displayed. It becomes the oldest frame number that can be displayed when omitting the number of the frame.

Command qualifiers

/INSTRUCTION (default when omitted)

Displays the disassembling results.

/SOURCE

Displays the trace results in units of source lines.

/ONEFRAME

Displays trace data only by one line.

- Description

Detailed information on the trace is displayed. It displays it up to 12 frames or less at a time.

SHOW DETAILTRACE command is a command that can be specified only with the chip DSU3 for emulator debugger (MB2197).

SHOW DETAILTRACE does the following processing, and displays the result of divergence address information and data access information displayed by the SHOW TRACE command.

- Complement to branch address information
 - Determine, the instruction between branch instructions, by disassembling it from branch address information.
- Search for data access instruction
 - Determine the instruction, making data access, between branch instructions from the disassembling results.
 - If /ONEFRAME is omitted, up to 12 lines of information will be displayed.

● Example

```
>SHOW DETAILTRACE/INSTRUCTION
frame no.      b-addr      mnemonic      b-cause      a-addr      a-stat      a-size      a-type      a-data      r-id      r-data
-00009 : -----
sample.c$73      while (1) {
-00008 :      00060248      BRA      00060224      Go      -----
sample.c$72      value[i] = &target[i];
-00007 :      00060224      LD      @(R14,-4),R3      Branch      FFFFFFFFC      read      word      cpu      00000003      3      FFFFFFFF
:      00060226      LSL      #2,R3
:      00060228      LD      @(R14,-4),R0
:      0006022A      LDI:8      #14,R1
:      0006022C      MUL      R1,R0
:      0006022E      MOV      MDL,R0
:      00060230      ADDN      R2,R0
:      00060232      ADDN      R14,R3

>SHOW DETAILTRACE/SOURCE
frame no.      source (-00009 .. 00000)
-00008 :      sample.c$73      while (1) {
-00007 :      sample.c$72      value[i] = &target[i];
-00001 :      sample.c$73      while (1) {
```

Note:

[Emulator Debugger (MB2100-01)]

This command can only be used during user program execution if the trace capture has finished.

4.28 SET TRACETRIGGER

The **SET TRACETRIGGER** command controls the state of trace measurements when the specified conditions are satisfied.

■ SET TRACETRIGGER

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

[Emulator Debugger (MB2198)]

SET TRACETRIGGER address [&= address mask] [, [!] d=data [&= data mask]]

[Emulator Debugger (MB2100-01)]

SET TRACETRIGGER address

Parameters

address (address expression)

Specifies the address that serves as a trace trigger condition.

address mask (data expression)

Specifies the address mask bit pattern that serves as a trace trigger condition.

Comparison is made only on the address value in the bit position where 1 is set.

Cannot be specified on the emulator debugger (MB2100-01).

data (data expression)

Specifies data that serves as a trace trigger condition.

Data cannot be specified under the code execution condition.

When the /ANYTHING qualifier is specified, 32-bit data is targeted for comparison. (To compare 8/16-bit data, it is necessary to specify the mask data.)

Cannot be specified on the emulator debugger (MB2100-01).

mask data (data expression)

Specifies the data mask bit pattern that serves as a trace trigger condition.

Comparison is made only on the data value in the bit position where 1 is set.

The data mask bit pattern cannot be specified under the code execution condition.

Cannot be specified on the emulator debugger (MB2100-01).

Command qualifiers

- Specifying access attribute

/READ

Specifies data read access as a trace trigger condition.

For the emulator debugger (MB2198), cannot be specified on the FR80S.

/WRITE (default when emulator debugger (MB2198 (FR80S)) is omitted)

Specifies data write access as a trace trigger condition.

/CODE (default when emulator debugger (MB2198 (FR60Lite) or MB2100-01) is omitted)

Specifies the code execution as a trace trigger condition.

For the emulator debugger (MB2198), cannot be specified on the FR80S.

- Specifying access size

/BYTE (Default on the emulator debugger (MB2100-01) when not specified)

Specifies 8 bits access as a trace trigger condition.

/HALFWORD

Specifies 16 bits (2-byte) access as a trace trigger condition. This parameter cannot be specified if /CODE is set.

/WORD

Specifies 32 bits (4-byte) access as a trace trigger condition. This parameter cannot be specified if /CODE is set.

/ANYTHING (Default on the emulator debugger (MB2198) when not specified)

A trace trigger condition is applied when the specified address is accessed, regardless of the access data length. Cannot be specified on the emulator debugger (MB2100-01).

- Specifying trace control

/ENABLETRACE (default when emulator debugger (MB2198) omitted)

When the trace trigger conditions are satisfied, trace measurements are started.

Cannot be specified on the emulator debugger (MB2100-01).

/DISABLETRACE (Default on the emulator debugger (MB2100-01) when not specified)

When the trace trigger conditions are satisfied, trace measurements are stopped.

- Description

When the specified conditions are satisfied, the state of trace measurements is controlled.

[Emulator Debugger (MB2198)]

Up to four points can be set. The count of points that can be used varies depending on the settings of SET BREAK/DATAWATCH and SET EVENT.

[Emulator Debugger (MB2100-01)]

Only one can be set. Because the hardware is shared, it cannot be specified if the total number of the following commands is 8.

SET BREAK

SET DATABREAK

SET BREAK/DATAWATCH

SET TRACETRIGGER

SET EVENT

- Example

```
>SET TRACETRIGGER/CODE/ENABLETRACE FF0000
```

```
>SET TRACETRIGGER/READ/WORD/ENABLETRACE 120046 !D=1234&=FFFC
```

Notes:

[Emulator Debugger (MB2198)]

- This command can't use except the FR60Lite or FR80S. For details, refer to "2.3.7 Trace" of "SOFTUNE WORKBENCH USER'S MANUAL".
- For FR80S, this is valid only when the setting is for the built-in RAM area.
- This function can't be used when the performance mode is set as the event mode.
- When the /ANYTHING qualifiers is specified, 32-bit data is targeted for comparison. To compare 8/16-bit data, it is necessary to specify the mask data.

Ex.) For 8-bit data(0x12)

```
SET TRACETRIGGER /ANYTHING 0x10000, D=0x12000000&=ff000000
```

Ex.) For 16-bit data(0x4567)

```
SET TRACETRIGGER /ANYTHING 0x10000, D=0x45670000&=ffff0000
```

[Emulator Debugger (MB2100-01)]

This command can only be used while a user program is executing if the trace capture has finished.

4.29 SHOW TRACETRIGGER

The **SHOW TRACETRIGGER** command displays the state of the trace trigger set by the **SET TRACETRIGGER** command.

■ SHOW TRACETRIGGER

- Debugger

Simulator		X
Emulator	(MB2197)	X
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		X

- Format

SHOW TRACETRIGGER[trigger-number[,...]]

Parameter

trigger-number (default: decimal)

Specifies the number of the trace trigger to be displayed.

Command qualifiers

/ALL (default when omitted)

All trace triggers set by the SET TRACETRIGGER command are displayed.

- Description

The state of the trace trigger set by the SET TRACETRIGGER command is displayed.

The specified trace trigger number can also be referenced from the address by using the built-in function %TRIGGERNUM.

- Example

```
>SHOW TRACETRIGGER
```

no.	en/dis	ctrl	address	add-mask	data	d-mask	size	access	symbol
1	enable	stop	00FF0010	-----	-----	-----	-----	code	
2	disable	stop	01200430	-----	!0000A5A5	0000F5FF	word	R/W	

Notes:

[Emulator Debugger (MB2198)]

- This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.7 Trace" of "SOFTUNE WORKBENCH USER'S MANUAL".
- This function cannot be used when the performance mode is set as the event mode.

4.30 CANCEL TRACETRIGGER

The **CANCEL TRACETRIGGER** command deletes the trace trigger set by the **SET TRACETRIGGER** command.

■ CANCEL TRACETRIGGER

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

CANCEL TRACETRIGGER [trigger-number[,...]]

Parameter

trigger-number (default: decimal)

Specifies the number of the trace trigger to be deleted.

Command qualifiers

/ALL (default when omitted)

Deletes all trace triggers set by the SET TRACETRIGGER command.

- Description

The CANCEL TRACETRIGGER command deletes the trace trigger set by the SET TRACETRIGGER command.

The specified trace trigger number can also be referenced from the address by using the built-in function %TRIGGERNUM.

- Example

```
>CANCEL TRACETRIGGER/ALL
```

Notes:

[Emulator Debugger (MB2198)]

- This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.7 Trace" of "SOFTUNE WORKBENCH USER'S MANUAL".
- This function cannot be used when the performance mode is set as the event mode.

[Emulator Debugger (MB2100-01)]

- This command can only be used during user program execution if the trace capture has finished.

4.31 ENABLE TRACETRIGGER

The **ENABLE TRACETRIGGER** command enables the trace trigger set by the **SET TRACETRIGGER** command.

■ ENABLE TRACETRIGGER

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

ENABLE TRACETRIGGER [trigger-number [, . . .]]

Parameter

trigger-number (default: decimal)

Specifies the number of the trace trigger to be enabled.

Command qualifiers

/ALL (default when omitted)

Enables all trace triggers set by the SET TRACETRIGGER command.

- Description

The ENABLE TRACETRIGGER command enables the trace trigger set by the SET TRACETRIGGER command.

The specified trace trigger number can also be referenced from the address by using the built-in function %TRIGGERNUM.

- Example

```
>ENABLE TRACETRIGGER/ALL
```

Notes:

[Emulator Debugger (MB2198)]

- This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.7 Trace" of "SOFTUNE WORKBENCH USER'S MANUAL".
- This function cannot be used when the performance mode is set as the event mode.

[Emulator Debugger (MB2100-01)]

- This command can only be used during user program execution if the trace capture has finished.

4.32 DISABLE TRACETRIGGER

The **DISABLE TRACETRIGGER** command disables the trace trigger set by the **SET TRACETRIGGER** command.

■ DISABLE TRACETRIGGER

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		×

- Format

DISABLE TRACETRIGGER [trigger-number[,...]]

Parameter

trigger-number (default: decimal)

Specifies the number of the trace trigger to be disabled.

Command qualifiers

/ALL (default when omitted)

Enables all trace triggers set by the SET TRACETRIGGER command.

- Description

The **DISABLE TRACETRIGGER** command disables the trace trigger set by the **SET TRACETRIGGER** command.

The specified trace trigger number can also be referenced from the address by using the built-in function %TRIGGERNUM.

- Example

```
>DISABLE TRACETRIGGER/ALL
```

Notes:

[Emulator Debugger (MB2198)]

- This command cannot use except the FR60Lite or FR80S. For details, refer to "2.3.7 Trace" of "SOFTUNE WORKBENCH USER'S MANUAL".
- This function cannot be used when the performance mode is set as the event mode.

[Emulator Debugger (MB2100-01)]

- This command can only be used during user program execution if the trace capture has finished.

4.33 SET DELAY

The **SET DELAY** command configures the delay count when the trace trigger is hit.

■ SET DELAY

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SET DELAY delay count

Parameter

delay count (defaults to decimal)

Specifies the delay values from when the trace trigger is hit until when the trace is stopped. Delay count is executed for each frame.

Maximum delay value that can be set becomes the frame number is mounted of each kinds.

- Description

The SET DELAY command configures the delay count when the trace trigger is hit.

- Example

```
>SET DELAY 30
```

Note:

This command can only be used while a user program is executing if the trace capture has finished.

4.34 SHOW DELAY

The **SHOW DELAY** command shows the current setting of the delay count when the trace trigger is hit.

■ SHOW DELAY

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙
Monitor		×

- Format

SHOW DELAY

- Description

The SHOW DELAY command shows the current setting of the delay count.

- Example

```
>SHOW DELAY
delay count =30
```

4.35 SET RAMCHECK

The **SET RAMCHECK** command configures the RAM checker.

■ SET RAMCHECK

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

● Format

<Format 1>

SET RAMCHECK address

<Format 2>

SET RAMCHECK file-name

Parameter

- Format 1

address

Specifies the monitoring address to be set.

- Format 2

file-name

Specifies the name of the log file that stores sampling data.

Command qualifiers

- Format 1

Data size

/BYTE

Specifies a data size in 8 bits.

/HALFWORD

Specifies a data size in 16 bits. The lower 1 bit of the monitoring address is ignored.

/WORD (default when omitted)

Specifies a data size in 32 bits. The lower 2 bits of the monitoring address are ignored.

Access attribute

/READ

Specifies data read access as a data monitoring condition. Cannot be specified for FR80S.

/WRITE

Specifies data write access as a data monitoring condition.

When the specification FR60Lite is omitted, the access attribute is set to /READ/WRITE.

For FR80S, the omission means /WRITE is set.

- Format 2

/SOFTUNE

Specifies when storing files in the SOFTUNE format.

When an extension is omitted, ".SRL" is added.

/CSV

Specifies when storing files in the CSV format.

When an extension is omitted, ".CSV" is added.

- Description

- Format 1

Set the monitoring address, data size, and access attribute of the RAM Checker.

The monitoring addresses of up to eight points can be set.

These monitoring addresses are stored in a log in the order they are set.

- Format 2

Specifies the log file that obtains sampling data, and the file storage format.

Command qualifier cannot be omitted. Specify either /SOFTUNE or /CSV.

- Example

```
>SET RAMCHECK 0x0003C100
>SHOW RAMCHECK
Logging mode    : disable
Logging name    :
Logging format   : SOFTUNE
ch 0 : 0003C000 / word / RW : main\dat_0
>SET RAMCHECK /BYTE/WRITE 0x0003C110
>SHOW RAMCHECK
Logging mode    : disable
Logging name    :
Logging format   : SOFTUNE
ch 0 : 0003C100 / word / RW : main\dat_0
ch 1 : 0003C110 / byte / W  : main\dat_8
>SET RAMCHECK /SOFTUNE TEMP2
>SHOW RAMCHECK
Logging mode    : disable
Logging name    : TEMP2.SRL
Logging format   : SOFTUNE
ch 0 : 0003C100 / word / RW : main\dat_0
ch 1 : 0003C110 / byte / W  : main\dat_8
```

Notes:

- Use this command for only when the FR60Lite or FR80S is used. For details, refer to "2.3.12 RAM Checker" of "SOFTUNE WORKBENCH USER'S MANUAL".
 - This command can be used only when debug mode is RAM Checker mode.
 - For FR80S, this is valid only when the setting is for the built-in RAM area.
-

4.36 SHOW RAMCHECK

The **SHOW RAMCHECK** command displays the setting of the RAM Checker.

■ SHOW RAMCHECK

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	×
Monitor		×

- Format

SHOW RAMCHECK

Parameter

None

Command qualifiers

None

- Description

The **SHOW RAMCHECK** command displays the setting of the RAM Checker.

- Example

```
> SHOW RAMCHECK
Logging mode      : enable
Logging name      : TEMP.SRL
Logging format    : SOFTUNE
ch 0 : 0003C100 /word / RW : main\dat_0
ch 1 : 0003C102 /byte / RW : main\dat_1
ch 2 : 0003C104 /word / RW : main\dat_2
```

Notes:

- Use this command for only when the FR60Lite or FR80S is used. For details, refer to "2.3.12 RAM Checker" of "SOFTUNE WORKBENCH USER'S MANUAL".
- This command can be used only when debug mode is RAM Checker mode.

4.37 CANCEL RAMCHECK

The **CANCEL RAMCHECK** command deletes the setting of the RAM Checker.

■ CANCEL RAMCHECK

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

<Format 1>

CANCEL RAMCHECK address

<Format 2>

CANCEL RAMCHECK

Parameter

address

Input the monitoring address you want to delete.

When the address is specified, no command qualifier can be specified.

Command qualifiers

/ALL

Deletes all the set monitoring addresses.

When a command qualifier is specified, no parameter can be input.

- Description

The CANCEL RAMCHECK command deletes the setting of the RAM Checker.

- Example

```
> SHOW RAMCHECK
Logging mode      : enable
Logging name      : TEMP.SRL
Logging format    : SOFTUNE
ch 0 : 0003C100 /word / RW : main\dat_0
ch 1 : 0003C102 /byte / RW : main\dat_1
ch 2 : 0003C104 /word / RW : main\dat_2
> CANCEL RAMCHECK 0x0003C102
> SHOW RAMCHECK
Logging mode      : enable
Logging name      : TEMP.SRL
Logging format    : SOFTUNE
ch 0 : 0003C100 /word / RW : main\dat_0
ch 1 : 0003C104 /word / RW : main\dat_2
```

```
> CANCEL RAMCHECK /ALL
> SHOW RAMCHECK
  Logging mode      : enable
  Logging name      : TEMP.SRL
  Logging format    : SOFTUNE
```

Notes:

- Use this command for only when the FR60Lite or FR80S is used. For details, refer to "2.3.12 RAM Checker" of "SOFTUNE WORKBENCH USER'S MANUAL".
 - This command can be used only when debug mode is RAM Checker mode.
-

4.38 ENABLE RAMCHECK

The **ENABLE RAMCHECK** command enables the logging status of the RAM Checker.

■ ENABLE RAMCHECK

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

ENABLE RAMCHECK

Parameter

None

Command qualifiers

None

- Description

The **ENABLE RAMCHECK** command enables the logging status of the RAM Checker.

When event mode is set to performance mode, this command cannot be used.

- Example

```
> SHOW RAMCHECK
Logging mode      : disable
Logging name      : TEMP.SRL
Logging format    : SOFTUNE
ch 0 : 0003C100 /word / RW : main\dat_0
ch 1 : 0003C102 /byte / RW : main\dat_1
ch 2 : 0003C104 /word / RW : main\dat_2
> ENABLE RAMCHECK
> SHOW RAMCHECK
Logging mode      : enable
Logging name      : TEMP.SRL
Logging format    : SOFTUNE
ch 0 : 0003C100 /word / RW : main\dat_0
ch 1 : 0003C102 /byte / RW : main\dat_1
ch 2 : 0003C104 /word / RW : main\dat_2
```

Notes:

- Use this command for only when the FR60Lite or FR80S is used. For details, refer to "2.3.12 RAM Checker" of "SOFTUNE WORKBENCH USER'S MANUAL".
 - This command can be used only when debug mode is RAM Checker mode.
-

4.39 DISABLE RAMCHECK

The **DISABLE RAMCHECK** command disables the logging status of the RAM Checker.

■ DISABLE RAMCHECK

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
	(MB2100-01)	×
Monitor		×

- Format

DISABLE RAMCHECK

Parameter

None

Command qualifiers

None

- Description

The **DISABLE RAMCHECK** command disables the logging status of the RAM Checker.

- Example

```
> SHOW RAMCHECK
Logging mode      : enable
Logging name      : TEMP.SRL
Logging format    : SOFTUNE
ch 0 : 0003C100 /word / RW : main\dat_0
ch 1 : 0003C102 /byte / RW : main\dat_1
ch 2 : 0003C104 /word / RW : main\dat_2
> DISABLE RAMCHECK
> SHOW RAMCHECK
Logging mode      : disable
Logging name      : TEMP.SRL
Logging format    : SOFTUNE
ch 0 : 0003C100 /word / RW : main\dat_0
ch 1 : 0003C102 /byte / RW : main\dat_1
ch 2 : 0003C104 /word / RW : main\dat_2
```

Notes:

- Use this command for only when the FR60Lite or FR80S is used. For details, refer to "2.3.12 RAM Checker" of "SOFTUNE WORKBENCH USER'S MANUAL".
- This command can be used only when debug mode is RAM Checker mode.

4.40 SET SEMIHOSTING

The **SET SEMIHOSTING** command configures the method to acquire the semihosting data.

■ SET SEMIHOSTING

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	○*
Monitor		×

*: This can be used only for products whose semihosting function is enabled.

For details, refer to the Hardware Manual for the product type you are using.

● Format

SET SEMIHOSTING [Address]

Command qualifiers

- Specifying the method to acquire the data

/BUFFER

Transfer the data via the transfer buffer on the user memory, and outputs to the terminal window.

/MBR

Transfer the data one byte by one byte using MBR, and outputs to the terminal window.

/AUTO (default at start-up)

Automatically sets the data acquisition method.

Acquire the address of the label `__SemiHostBuffer` from the debug information, then set as follows by the result.

When the acquisition succeeded: The acquired address is used as the transfer buffer address for semihosting, and set with /BUFFER.

When the acquisition failed: Set with /MBR.

Parameter

address

Specifies the address of the transfer buffer.

Only valid when specifying /BUFFER.

● Description

The SET SEMIHOSTING command configures the method to acquire the semihosting data.

For details of the semihosting function, refer to Section "2.6.10 Displaying Messages Output from the User Program on Debugger" in "SOFTUNE Workbench User's Manual".

● Example

When automatically configuring the data acquisition method of semihosting

```
> SET SEMIHOSTING
```

4.41 SHOW SEMIHOSTING

The **SHOW SEMIHOSTING** command displays the method to acquire the semihosting data.

■ SHOW SEMIHOSTING

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	×
	(MB2100-01)	⊙*
Monitor		×

*: This can be used only for products whose semihosting function is enabled.

For details, refer to the Hardware Manual for the product type you are using.

- Format

SHOW SEMIHOSTING

- Description

The SHOW SEMIHOSTING command displays the method to acquire the semihosting data.

For details of the semihosting function, refer to Section "2.6.10 Displaying Messages Output from the User Program on Debugger" in "SOFTUNE Workbench User's Manual".

The displayed contents are as follows:

- For auto
 - auto/manual : (automatic setting state)
 - mode : <mode string: "MBR"/"buffer">
 - __SemiHostBuffer : <buffer address>
- For manual
 - auto/manual : (automatic setting state)
 - mode : <mode string: "MBR"/"buffer">
 - address : <buffer address>

- Example

When automatically configuring the data acquisition method of semihosting and it is set to MBR

```
>SET SEMIHOSTING /AUTO
>SHOW SEMIHOSTING
auto/manual : auto
mode : MBR
```

When automatically configuring the data acquisition method of semihosting and it is set to BUFFER

```
>SHOW SEMIHOSTING
auto/manual : auto
mode : buffer
__SemiHostBuffer : 00011000
```

When the data acquisition method of semihosting is set to MBR

```
>SET SEMIHOSTING /MBR
auto/manual : manual
mode : MBR
```

When the data acquisition method of semihosting is set to BUFFER

```
>SET SEMIHOSTING /MEMORY 11000
>SHOW SEMIHOSTING
auto/manual : manual
mode : buffer
address : 00011000
```


CHAPTER 5 Memory/Register Operation Commands



This chapter describes the Memory/Register Operation commands.

- 5.1 EXAMINE
- 5.2 ENTER
- 5.3 SET MEMORY
- 5.4 SHOW MEMORY
- 5.5 SEARCH MEMORY
- 5.6 SET REGISTER
- 5.7 SHOW REGISTER
- 5.8 COMPARE
- 5.9 FILL
- 5.10 MOVE
- 5.11 DUMP

5.1 EXAMINE

The EXAMINE command analyzes the specified C/C++ expression and displays the result.

■ EXAMINE

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙*
	(MB2100-01)	⊙
Monitor		○

*: The DSU3 evaluation chip cannot specify this command during the program running.

● Format

EXAMINE expression [, ...]

Parameter

expression (address formula)

Specifies expression to be analyzed.

Command qualifiers

/BINARY

Specifies the formula solution to be displayed as binary number.

/OCTAL

Specifies the formula solution to be displayed as octal number.

/DECIMAL

Specifies the formula solution to be displayed as decimal number.

/HEXADECIMAL

Specifies the formula solution to be displayed as a hexadecimal number.

/SINGLE

Specified to display the analyzed formula solution as a single-precision floating-point number.

/DOUBLE

Specified to display the analyzed formula solution as a double--precision floating-point number.

● Description

The EXAMINE command analyzes the specified C/C++ expression and displays the result.

When a variable is specified, the command displays the data.

When a variable of structure or union or class type is specified, the command displays all the member values.

When only an array name is specified, the command displays all the data of that array.

When the display base number of a command qualifier is omitted, the base number specified by the SET RADIX command is assumed.

- Example

```
>EXAMINE strsym
strsym =      {
    a = H'20
    b = H'4A30
    c = H'3012
}
>EXAMINE strsym.a
strsym.a = H'20
>EXAMINE flags [0]
flags [0] = H'03
>EXAMINE flags
flags [0] = H'05
flags [1] = H'50
flags [2] = H'10
flags [3] = H'2A
>EXAMINE/DECIMAL count
count = D'12
>EXAMINE/HEXADECIMAL count
count = H'0C
>EXAMINE/DECIMAL fwork
fwork = 2.36S+1
```

Note:

For the method of specifying formula, refer to the following sections of "SOFTUNE WORKBENCH OPERATION MANUAL".

- "2.1.1 Data and Address Formulas (Numerical constant)"
 - "2.1.2 Data and Address Formulas (Symbols, Line Numbers, Character Constants)"
 - "2.1.3 Data and Address Formulas (Register name, Flag name)"
-

5.2 ENTER

The ENTER command assigns the specified data to the specified variable.

■ ENTER

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙*
	(MB2100-01)	⊙
Monitor		○

*: The DSU3 evaluation chip cannot specify this command during the program running.

● Format

ENTER variable = data

Parameters

variable (address formula)

Specifies the variable where data to be stored.

data (data formula)

Specifies the data to be stored.

Command qualifiers

• Data length

/BYTE

Stores specified value in specified memory location as 1-byte length.

/HALFWORD

Stores specified value in specified memory location as 2-byte length.

/WORD

Stores specified value in specified memory location as 4-byte length.

/DWORD

Stores specified value in specified memory location as 8-byte length.

/SINGLE

Stores specified value in specified memory location as single-precision floating-point number.

/DOUBLE

Stores specified value in specified memory location as double-precision floating-point number.

● Description

The ENTER command assigns the specified data to the specified variable.

Specifying the type of command qualifier enables data to be assigned at the specified size.

- Example

```
>ENTER tmcnt = 10
>ENTER work = 6A5
>ENTER tmp = 1DF2BF
>ENTER fsymbol = F'10.55S+2
>ENTER/WORD work = 12345678
```

5.3 SET MEMORY

The **SET MEMORY** command stores the specified data in the specified memory location according to the type of the specified command qualifier.

■ SET MEMORY

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙*
	(MB2100-01)	⊙
Monitor		○

*: The DSU3 evaluation chip cannot specify this command during the program running.

● Format

SET MEMORY [storage-address] = data [, ...]

Parameters

storage-address (address formula)

Specifies the memory location where specified data to be stored.

data (data formula)

Specifies the value to be stored.

Command qualifiers

• Data length

/BIT

Stores specified value in specified memory location as bit-length length.

/BYTE (default when omitted)

Stores specified value in specified memory location as 1-byte length.

/HALFWORD

Stores specified value in specified memory location as 2-byte length.

/WORD

Stores specified value in specified memory location as 4-byte length.

/DWORD

Stores specified value in specified memory location as 8-byte length.

/SINGLE

Stores specified value in specified memory location as single-precision floating-point number.

/DOUBLE

Stores specified value in specified memory location as double-precision floating-point number.

/STRING

Stores value specified in character string in specified memory location as ASCII code data.

- Description

The SET MEMORY command stores the specified data in the specified memory location according to the type of the specified command qualifier.

When storage-address is omitted, the command stores the specified data in the memory location next to the memory location last accessed by the SHOW MEMORY or SET MEMORY commands. The type of the data to be stored is the same as that of the last accessed memory data.

When only a period (.) is specified in storage-address, the command stores the data in the memory location last accessed by the SHOW MEMORY or SET MEMORY commands.

In this case, the type of the data to be stored is also the same as that of the last accessed memory data.

If the type of command qualifier is omitted, /BYTE is assumed.

- Example

```
>SET MEMORY/BYTE 1000 = 10
>SET MEMORY/HALFWORD 1030 = 6A5
>SET MEMORY/WORD 1050 = 1DF2BF
>SET MEMORY/STRING 2000 = "ST"
>SET MEMORY . = 45
>SET MEMORY/BIT 8000:3 = 1
>SET MEMORY/SINGLE 2050 = F'10.55S+2
```

5.4 SHOW MEMORY

The **SHOW MEMORY** command displays data in the memory location, specified by address or address-range, according to the type of specified data.

■ SHOW MEMORY

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙*
	(MB2100-01)	⊙
Monitor		○

*: The DSU3 evaluation chip cannot specify this command during the program running.

- Format

SHOW MEMORY [{address | address-range} [, ...]]

Parameters

address (address formula)

Specifies the address in memory location to be checked.

address-range (address formula)

Specifies the memory area range to be checked.

Command qualifiers

/BIT

Specifies that value to be checked to be displayed as 1-bit length.

/BYTE (default when omitted)

Specifies that value to be checked to be displayed as 1-byte length.

/HALFWORD

Specifies that value to be checked to be displayed as 2-byte length.

/WORD

Specifies that value to be checked to be displayed as 4-byte length.

/DWORD

Specifies that value to be checked to be displayed as 8-byte length.

/SINGLE

Specifies that value to be checked to be displayed as single-precision floating-point number.

/DOUBLE

Specifies that value to be checked to be displayed as double-precision floating-point number.

/ASCII

Specifies that value to be checked to be displayed as ASCII characters.

/STRING

Specifies that value to be checked to be displayed as character string.

/BINARY

Specifies that value to be checked to be displayed as binary number.

/OCTAL

Specifies that value to be checked to be displayed as octal number.

/DECIMAL

Specifies that value to be checked to be displayed as decimal number.

/HEXADECIMAL

Specifies that value to be checked to be displayed as hexadecimal number.

- **Description**

The SHOW MEMORY command displays data in the memory location, specified by address or address-range, according to the type of specified data. However, when /BIT is specified, address-range cannot be specified.

When address and address-range are omitted, the command displays data in the memory location next to the memory location last accessed by the SHOW MEMORY or SET MEMORY commands.

The type of the data to be displayed is the same as that of the last-accessed memory data.

When only a period (.) is specified, the command displays the data in the memory location last accessed by the SHOW MEMORY or SET MEMORY commands.

In this case, the type of data to be displayed is also the same as that of the last accessed memory data.

If the command qualifier type is omitted, /BYTE is assumed.

If the display base number of a command qualifier is omitted, the base number specified by the SET RADIX command is assumed.

- **Example**

```
>SHOW MEMORY/DECIMAL 1000
00001000 = D'12
>SHOW MEMORY/BINARY 1000
00001000 = B'00001100
>SHOW MEMORY/HEXADECIMAL 1000..1001
00001000 = H'0C
00001001 = H'41
>SHOW MEMORY/HEXADECIMAL/HALFWORD 1000
0000      1000 = D'410C
>SHOW MEMORY/HEXADECIMAL/WORD 1000
00001000 = H'0030410C
>SHOW MEMORY/HEXADECIMAL 1000, 1020
00001000 = H'0C
00001020 = H'E3
>SHOW MEMORY/ASCII 1001
00001000 = 'A'
>SHOW MEMORY/SINGLE/DECIMAL 1030
00001030 = 2.36S+1
>SHOW MEMORY/BYTE 1000
0000      1000 = H'0C
>SHOW MEMORY .
00001000 = H'0C
>SHOW MEMORY
00001001 = H'41
```

5.5 SEARCH MEMORY

The **SEARCH MEMORY** command searches the specified memory for the specified data and displays the address matching the data.

■ SEARCH MEMORY

- Debugger

Simulator		⊙
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		○

- Format

SEARCH MEMORY address-range = data [, ...] [, S = skip-byte-count]

Parameters

address-range (address formula)

Specifies the memory area to be searched.

data (data formula)

Specifies the data to be searched.

skip-byte-count (data formula)

Specifies the number of bytes to be skipped.

H'1 to H'FFFF can be specified.

If this parameter is omitted, the data length is assumed.

Command qualifiers

- Data length

/BYTE (default when omitted)

Searches for specified data as byte-length data.

/HALFWORD

Skips 2 bytes and searches for specified data as 2-byte length.

/WORD

Skips 4 bytes and searches for specified data as 4-byte length.

/DWORD

Skips 8 bytes and searches for specified data as 8-byte length.

/ASCII

Searches for specified data as ASCII character strings.

- Description

The **SEARCH MEMORY** command searches the specified memory for the specified data and displays the address matching the data.

- Example

```
>SEARCH MEMORY 2000..3000 = 88  
found at = 00002050  
found at = 00002577  
found at = 00002BDF
```

5.6 SET REGISTER

The **SET REGISTER** command sets the specified value in the specified register or flag.

■ SET REGISTER

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

● Format

SET REGISTER register-name = data

Parameters

register-name

Specifies the name of register or flag to be modified.

For register and flag names, refer to the "Appendix A List of Register Names" of "SOFTUNE Workbench Operation Manual".

data (data formula)

Specifies the value to be set in specified register or flag.

● Description

The SET REGISTER command sets the specified value in the specified register or flag.

● Example

```
>SET REGISTER PC = 1000
>SET REGISTER C = 1
```

Notes:

Note on the following points in setting values in the TBR register.

- The program runs away when the I/O area and vector table overlap in the TBR register.
- A parameter error occurs when a value exceeding H'FFFFFFC00 is specified in the TBR register.

5.7 SHOW REGISTER

The **SHOW REGISTER** command displays the values of the specified register or flag in hexadecimal notation.

■ SHOW REGISTER

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

- Format

SHOW REGISTER register-name

Parameter

register-name

Specifies the name of register or flag to be checked. When the register name is omitted, all registers and content of the flag are displayed.

For register and flag names, refer to the "Appendix A List of Register Names" of "SOFTUNE Workbench Operation Manual".

Command qualifier

/ALL (default when omitted)

Displays values of all registers and flags.

- Description

The SHOW REGISTER command displays the values of the specified register or flag in hexadecimal notation.

When not set, each flag in the CCR register displays "-". When set, it displays the flag name.

- Example

```
>SHOW REGISTER PC
PC = 00FF0000
>SHOW REGISTER
R0 = 00000000 R1 = 00000000 R2 = 00000000 R3 = 00000000
R4 = 00000000 R5 = 00000000 R6 = 00000000 R7 = 00000000
R8 = 00000000 R9 = 00000000 R10 = 00000000 R11 = 00000000
R12 = 00000000 R13 = 00000000 R14 = 00000000 R15 = 0000FFC0
MDH = 00000000 MDL = 00000000 RP = 00000000 PS = FFFFFFFF
PC = 000FF000 USP = 0000E000 SSP = 0000FFC0 CCR = --SINZVC
SCR = --T      ILM = 1F      TBR = 000FFC00
```

5.8 COMPARE

The COMPARE command compares memory data.

■ COMPARE

- Debugger

Simulator		⊙
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		○

- Format

COMPARE compare-origin-address-range, comparison-destination-address

Parameters

compare-origin-address-range (address formula)

Specifies the memory area of compare origin.

comparison-destination-address (address formula)

Specifies the comparison destination address.

- Description

The COMPARE command compares memory data.

When no error is found as a result of the comparison, the COMPARE command displays "Not found".

When an error is found, the command displays (in hexadecimal notation) the memory location of the compare origin and the data to the left and the memory location of the comparison destination and the data to the right.

- Example

```
>COMPARE 2000..3000, 4000
address  source  destination  address
00002050 35      10      00004050
00002051 40      00      00004051
```

5.9 FILL

The FILL command fills the specified memory area with any data.

■ FILL

- Debugger

Simulator		⊙
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		○

- Format

FILL address-range = data [, ...]

Parameters

address-range (address formula)

Specifies the memory range to be filled with data.

data (data formula)

Specifies the data filling specified memory area.

Command qualifiers

- Data length

/BYTE (default when omitted)

Specifies filling of memory area with 1-byte length.

/HALFWORD

Specifies filling of memory area with 2-byte length.

/WORD

Specifies filling of memory area with 4-byte length.

/DWORD

Specifies filling of memory area with 8-byte length.

/ASCII

Specifies filling of memory area with ASCII character string data.

- Description

The FILL command fills the specified memory area with any data.

- Example

```
>FILL 2000..2FFF = 23
```

5.10 MOVE

The MOVE command transfers data from the specified memory area to the specified transfer destination.

■ MOVE

- Debugger

Simulator		⊙
Emulator	(MB2197)	○
	(MB2198)	○
	(MB2100-01)	⊙
Monitor		○

- Format

MOVE transfer-source-address-range, transfer-destination-address

Parameters

transfer-source-address-range (address formula)

Specifies the memory area from where data transferred.

transfer-destination-address (address formula)

Specifies the memory location to where data to be transferred.

- Description

The MOVE command transfers data from the specified memory area to the specified transfer destination.

- Example

```
>MOVE 2000..3000, 4000
```


5.11 DUMP

The DUMP command dumps the values of the specified memory area.

■ DUMP

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙*
	(MB2100-01)	⊙
Monitor		○

*: The DSU3 evaluation chip cannot specify this command during the program running.

- Format

DUMP [{starting-address | address-range}]

Parameters

starting-address (address formula)

Specify the memory address where dump to be started.

address-range (address formula)

Specify the memory area range to be dumped.

Command qualifiers

- Display units

/BIT

Dumps data in bits.

/BYTE (default at start-up)

Dumps data in bytes.

/HALFWORD

Dumps data in 2 bytes.

/WORD

Dumps data in 4 bytes.

/DWORD

Dumps data in 8 bytes.

- Description

The DUMP command dumps data in the specified memory area.

When only start-address is specified, the DUMP command dumps the first 16 lines in the output window.

When no parameter is specified, the command starts dumping from the memory location next to the memory location last-displayed as a result of previous command execution.

- **Example**

```
>DUMP 100..118
address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ---ascii--
00000100 00 00 41 42 43 00 00 00 00 00 00 00 00 00 00 00 ..ABC.....
00000110 53 49 4D 55 4C 41 54 4F 52                               SIMULATOR
>
>DUMP/HALFWORD 100..118
address  +0      +2      +4      +6      +8      +A      +C      +E      ---ascii--
00000100 0000    4241    0043    0000    0000    0000    0000    0000 ..ABC.....
00000110 4953    554D    414C    4F54    0052                               SIMULATOR
>
>DUMP/BIT 5
address  :7  :6  :5  :4  :3  :2  :1  :0 HEX
00000005 0   1   1   0   1   0   1   0  6A
00000006 1   1   1   1   0   1   0   0  F4
```

CHAPTER 6 Line Assemble and Disassemble Commands



This chapter describes the Line Assemble and Disassemble commands.

6.1 ASSEMBLE

6.2 DISASSEMBLE

6.1 ASSEMBLE

The **ASSEMBLE** command line-assembles the entered mnemonic and operand, and stores the instruction code in the specified memory location.

■ ASSEMBLE

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input checked="" type="radio"/>
Monitor		<input type="radio"/>

- Format

ASSEMBLE [starting-address] = assemble-character-string

Parameters

starting-address (address formula)

Specifies a starting address of memory containing line-assembled codes.

assemble-character-string (character string)

Specifies a character string to be line-assembled. Please enclose the string in double quotation marks ' ' (character).

- Description

The **ASSEMBLE** command line-assembles the entered mnemonic and operand, and stores the instruction code in the specified memory location.

When starting-address is omitted, the memory location containing the next address of the last executed instruction code is assumed.

- Example

```
>ASSEMBLE 1000 = "RET"
>ASSEMBLE 1006 = "ADD #1, R1"
>DISASSEMBLE 1000
00001000 RET
00001002 LDI #0, R0
00001004 LDUB @R0, R1
00001006 ADD #1, R1
00001008 STB R1, @R0
0000100A
```

6.2 DISASSEMBLE

The **DISASSEMBLE** command disassembles data in the specified memory location and displays it in the output window.

■ DISASSEMBLE

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙*
	(MB2100-01)	⊙
Monitor		○

*: The DSU3 evaluation chip cannot specify this command during the program running.

- Format

DISASSEMBLE [{starting-address | address-range}]

Parameters

starting-address (address formula)

Specifies a starting address of memory to be disassembled.

address-range (address formula)

Specifies a range of memory to be disassembled.

- Description

The **DISASSEMBLE** command disassembles data in the specified memory location and displays it in the output window.

When only starting-address is specified, the command disassembles and displays data by 16 lines.

When only a period (.) is specified in starting-address or address-range, the command starts disassembled data display from the address indicated by the current program counter.

When starting-address and address-range are omitted, the command displays disassembled data by 16 lines, starting from the line next to the last displayed line.

When **/DISPLAY** is specified in the **SET SOURCE** command and the memory location corresponds to the source line, the **DISASSEMBLE** command also displays the source line.

- Example

```
>DISASSEMBLE 1000..1002
00001000 9720      RET
00001002 C000      LDI:8      #0, R0
>DISASSEMBLE .
000FF000 1781      ST          RP, @-R15
000FF002 0F07      ENTER       #01C
000FF004 C010      LDI:8      #1, R0
000FF006 7FF0      STB         R0, @ (R14, -1)
```


CHAPTER 7 Load and Save Commands



This chapter describes the Load and Save commands.

7.1 LOAD

7.2 SAVE

7.1 LOAD

The **LOAD** command loads the specified file.

■ LOAD

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

● Format

LOAD file-name [, address] [, file-offset [, byte-count]]

Parameters

file-name

Specifies a name of file to be loaded.

The default extension depends on the command qualifier to be specified.

address (address formula)

Specifies a memory location (address) where memory image file to be loaded.

This parameter is valid only when command qualifier /BINARY is specified.

Specifying other command qualifiers results in an error.

file-offset (data formula)

Specifies an offset of load start data in specified file.

When file-offset is omitted, data is read from the beginning of the file.

This parameter is valid only when command qualifier /BINARY is specified.

Specifying other command qualifiers results in an error.

byte-count (data formula)

Specifies the loaded number of data by each bytes for the specified file name.

When byte-count is omitted, all data is read.

This parameter is valid only when command qualifier /BINARY is specified.

Specifying other command qualifiers results in an error.

Command qualifiers

Specifies the file format, the operation when loading it, and the data length.

- File format specification

/OBJECT (default when omitted)

Loads load module file.

The default extension is ".abs".

/DEBUG

Loads only debug information from load module file.

The default extension is ".abs".

/BINARY

Loads binary format memory image file.

The default extension is ".bin".

Addressing cannot be omitted.

/COVERAGE

Loads coverage data file.

The default extension is ".cov".

This qualifier is possible to use only for the high-speed version simulator debugger.

/ALIAS

Loads alias file (command alias definition, macro definition).

The default extension is ".lst".

- Operation specification at loading

/AUTOMATIC (default when omitted)

For simulator debugger, automatically sets map area at loading.

The command qualifiers can specify when the "/OBJECT" or "the /DEBUG" is specified.

/MANUAL

For simulator debugger, does not automatically set map area at loading.

A map area must be set by the SET MAP command. The command qualifiers can specify when the "/OBJECT" or "the /DEBUG" is specified.

/READ

For simulator debugger, sets ROM area for data segment as /READ attribute if /AUTOMATIC qualifier valid.

If this qualifier is omitted, the /READ/CODE attribute will be set. The command qualifiers can specify when the "/OBJECT" or "the /DEBUG" is specified.

/ONDEMAND

The load module file is on-demand loaded. This modifier can be specified when /OBJECT or /DEBUG is given.

/SYNCHRONIZE

The flash memory synchronizes immediately after loading the target file.

This command can specify when the "/OBJECT" or "/BINARY" is specified only on the emulator debugger (MB2100-01).

- Data length

/BYTE (default when omitted)

Specifies loading to memory area with 1-byte length.

/HALFWORD

Specifies loading to memory area with 2-byte length.

/WORD

Specifies loading to memory area with 4-byte length.

/DWORD

Specifies loading to memory area with 8-byte length.

- Description

The LOAD command loads the specified file.

This command can load the following four kinds of files. The file is opened adding the default extension respectively when the file name extension is omitted.

- Load module file
 - Absolute-format object file created by linker.
- Memory image file
 - Memory image file saved by SAVE command.
(An address should be always specified to load the files.)
- Coverage file
 - Coverage file saved by SAVE command.
- Alias file
 - File containing command alias definition and macro command definition.
(If a file extension is omitted, the default extension is added and the file is opened.)

- Example

```
>LOAD debug
>LOAD/BINARY data.bin, FE0000
```

Note:

When loading the target file with the LOAD command, the batch file specified with "Specification batch file before/after load" in the setup wizard is not executed.

For details, refer to "4.7.2.5 Setup Wizard" of "SOFTUNE Workbench Operation Manual".

7.2 SAVE

When all command qualifiers are omitted or when **/BINARY** is specified, the **SAVE** command saves data in the specified memory to the memory image file (binary format of data only).

■ SAVE

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

● Format

SAVE file-name [, {address-range | module-name}]

Parameters

file-name

Specifies a name of file where memory data to be saved.

When the file name extension is omitted, any of the following extensions is added:

- ".bin" (valid when memory data saved in memory image)
- ".cov" (valid when coverage data saved)
- ".csv" (valid when the coverage measurement result saved in CSV format)
- ".lst" (valid when command alias definition or macro command definition saved)

address-range (address formula)

Specifies a memory area to be saved.

Address-range is valid only when command qualifier **/BINARY** is specified.

Specifying other command qualifiers results in an error.

module-name

Specifies a module name of the coverage measurement result to be saved.

This is valid only when the command qualifier is **/COVERAGE**.

If this parameter is omitted, the coverage rate of entire module is saved.

Command qualifiers

• Kind of files

/BINARY (default when omitted)

Saves memory data to memory image file in binary format.

The default extension is ".bin".

Address-range specification cannot be omitted.

/COVERAGE

Saves coverage data in all areas specified by **SET COVERAGE** command.

The default extension is ".cov".

Address range specification is invalid.

/ALIAS

Saves command alias definition and macro command definition to alias file.

The default extension is ".lst".

Address-range specification is invalid.

- Saving format

/CSV

The file is saved in CSV format.

This is valid only when the command qualifier is /COVERAGE.

- **Description**

When all command qualifiers are omitted or when /BINARY is specified, the SAVE command saves data in the specified memory to the memory image file (binary format of data only).

In this case, address-range specification cannot be omitted.

If /COVERAGE is specified, this command will save coverage measurement data in all areas specified by the SET COVERAGE command.

If "/CSV" is specified at the same time, the coverage measurement result of the module is saved in CSV format.

When /ALIAS is specified, the command saves command alias definition and macro command definition to the alias file.

- **Example**

```
>SAVE memo.bin, 0..00000fff
```

```
>SAVE /COVERAGE/CSV cov.csv, sample.c
```

CHAPTER 8 Source File/Symbol Commands



This chapter describes the Source File/Symbol commands.

8.1 LIST

8.2 SET PATH

8.3 SHOW PATH

8.4 SHOW SCOPE

8.5 UP

8.6 DOWN

8.1 LIST

The LIST command displays the source line corresponding to the specified line number.

■ LIST

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

● Format

LIST [{ [file-name] line-number [.. line-number] | address}]

Parameters

file-name

Specifies a name of source file to be displayed.

When file-name is omitted, the previously-specified file name is assumed.

line-number

Specifies a number of source line to be displayed.

"\$" must always precede a line number.

When line numbers are delimited by ". . .", the source lines within the specified range are displayed.

address (address formula)

Specifies an address (memory location) where code attribute stored.

Specifies this parameter when displaying the source line corresponding to the address (memory location).

● Description

The LIST command displays the source line corresponding to the specified line number.

When only a period (.) is specified in file-name, line-number, or address, the command displays source lines of the count of lines in the output window, starting from the source line corresponding to the current program counter.

If the value in the program counter is rewritten due to program execution when all parameters are omitted, the command starts source line display from the source line corresponding to the current program counter.

In other cases, the command displays 19 source lines, starting from the line next to the previously-displayed last line.

- Example

```
>LIST PROGRAM.C$2..$3
2:   x = x+1 ;
3:   printf ("%d\n", x) ;
>LIST subdisp
30: subdisp ( )
31: {
32:   int i;
33:
34:   for (i = p; i >= 1; i--)
35:     printf ( "data [%d] = %d \n", i, data [i] );
36:
      .
      .
      .
>LIST.
53: switch (*s) {
54:   case '0' : z = " "; return (z) ;
55:   case '1' : z = "a"; return (z) ;
56:   case '2' : z = "b"; return (z) ;
```

8.2 SET PATH

The **SET PATH** command specifies the directories used to search for the source file.

■ SET PATH

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SET PATH [source-search-directory-name [, ...]]

Parameter

source-search-directory-name

Specifies a directory for which source file to be searched.

Command qualifier

/APPEND

Appends specified search directory to current setting.

- Description

The SET PATH command specifies the directories used to search for the source file.

The command searches the specified directories for the source file in sequence from the left.

When source-search-directory-name is omitted, the current directory is assumed.

- Example

```
>SET PATH A:\
```


8.3 SHOW PATH

The **SHOW PATH** command displays currently-enabled source file search directories.

■ SHOW PATH

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SHOW PATH

- Description

The SHOW PATH command displays currently-enabled source file search directories.

- Example

```
>SHOW PATH
source file search path = a:\
```

8.4 SHOW SCOPE

The **SHOW SCOPE** command displays the module and function names including the memory location indicated by the current program counter.

■ SHOW SCOPE

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

- Format

SHOW SCOPE

- Description

The **SHOW SCOPE** command displays the module and function names including the memory location indicated by the current program counter.

- Example

```
>SHOW SCOPE
current scope = SIEVE\sub_main\
```

8.5 UP

The UP command moves the scope to the parent function.

■ UP

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

- Format

UP

- Description

The UP command moves the scope to the parent function.

UP/DOWN information is cleared when the MCU is executed, RESET is performed, or the program counter is updated.

- Example

```
>UP
Current Scope = demo\sort\
```

8.6 DOWN

The **DOWN** command moves the scope to the child function.

■ DOWN

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

- Format

DOWN

- Description

The DOWN command moves the scope to the child function.

UP/DOWN information is cleared when the MCU is executed, RESET is performed, or the program counter is updated.

- Example

```
>DOWN
```

```
Current Scope = demo\check\
```

CHAPTER 9 Command Procedure Commands



This chapter describes the Command Procedure commands.

9.1 BATCH

9.2 QUIT

9.1 BATCH

The **BATCH** command executes the commands in the specified command procedure file.

■ BATCH

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

● Format

BATCH file-name [, actual-parameter [, ...]]

Parameters

file-name

Specifies a name of file where command procedure to be executed written.

The default extension is ".prc".

actual-parameter

Specifies an actual parameter required for command procedure.

Command qualifier

/ICON

Converts debugger to icon and executes it when command procedure executed.

When command procedure execution terminates, the icon is restored to the original size.

● Description

The **BATCH** command executes the commands in the specified command procedure file.

Batch processing (procedure file call) can be nested for up to 8 levels.

Actual parameters are replaced with temporary parameters (%P0 to %P9) in the order they were specified.

When the count of temporary parameters is greater than that of the specified actual parameters, the remaining temporary parameters are replaced by empty strings.

When the count of temporary parameters is less than that of the specified actual parameters, the remaining parameters are ignored.

The count of the specified actual parameters can be referenced by means of %NP.

● Example

```
>BATCH TST.PRC, 0, 0FFF, BRK
```

9.2 QUIT

Executing the QUIT command when the command procedure is being executed quits command procedure processing.

■ QUIT

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

QUIT

- Description

Executing the QUIT command when the command procedure is being executed quits command procedure processing.

If the QUIT command is executed during execution of a control command, the program exits all the control command loops.

When this command is executed in the command wait status, nothing is executed.

- Example

Data in command procedure file

```

IF %NP < 2
    QUIT
ENDIF
SET     VARIABLE I = 0
SET     VARIABLE ADDR = %P0
WHILE   %I<%P1
SET     MEMORY %ADDR = %I
SET     VARIABLE I = %I+1
IF %ADDR == H'FFFFFF
    QUIT
ELSE
    SET VARIABLE ADDR = %ADDR+1
ENDIF
ENDW

```


CHAPTER 10 Replacement Commands



This chapter describes the Replacement commands.

10.1 SET ALIAS

10.2 SHOW ALIAS

10.3 CANCEL ALIAS

10.4 SET VARIABLE

10.5 SHOW VARIABLE

10.6 CANCEL VARIABLE

10.1 SET ALIAS

The **SET ALIAS** command defines a command alias.

■ SET ALIAS

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

● Format

SET ALIAS alias = command-character-string

Parameters

alias (identifier)

Specifies a command alias.

command-character-string

Specifies a command character string (command name, command qualifier, and parameter) to be replaced with specified alias, enclosed in double quotation marks (").

● Description

The SET ALIAS command defines a command alias.

It is convenient to define command aliases for frequently-used commands.

No command alias can be nested.

Other command aliases cannot be included in command alias definition.

● Example

```
>SET ALIAS BP = "SET BREAK 00FF0300,3"
>SET ALIAS E = "ENTER"
>SET ALIAS R = "SHOW REGISTER"
```

10.2 SHOW ALIAS

The **SHOW ALIAS** command displays the defined command alias list.

■ SHOW ALIAS

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SHOW ALIAS

- Description

The SHOW ALIAS command displays the defined command alias list.

- Example

```
>SHOW ALIAS
T      :      STEP
D      :      EXAMINE
PC     :      SHOW REGISTER PC
>
```

10.3 CANCEL ALIAS

The **CANCEL ALIAS** command cancels the alias of the specified command character string.

■ CANCEL ALIAS

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

CANCEL ALIAS [alias [,...]]

Parameter

alias (identifier)

Specifies a command alias to be canceled.

Command qualifier

/ALL

Cancels aliases of all command character strings.

- Description

The CANCEL ALIAS command cancels the alias of the specified command character string.

- Example

```
>CANCEL ALIAS BP
>
```

10.4 SET VARIABLE

The **SET VARIABLE** command defines a debug variable.

■ SET VARIABLE

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SET VARIABLE debug-variable-name = replacing-character-string

Parameters

debug-variable-name (identifier)

Specifies a debug variable to be defined.

replacing-character-string

Specifies a character string replacing debug variable.

- Description

The SET VARIABLE command defines a debug variable.

The defined debug variable can be used as part of the parameter field when the command is specified.

The used debug variable is replaced with the replacing character string defined by this command as is.

All the variables that can be specified in the parameter field can be defined.

For example, a character string and an expression can be defined as they are.

The debug variable is replaced as defined without any change; therefore, care must be taken in the case of definitions using the same debug variable (e.g. increments).

- Example

```
>SET VARIABLE ADDR = 0309+12
>SET VARIABLE STR = "ABCDEF"
>SET MEMORY/STRING %ADDR = %STR
(SET MEMORY/STRING 0309+12 ="ABCDE" is replaced.)
>SET VARIABLE CNT = 1
>WHILE %CNT<5
*PRINTF "val [%d] = %d\n", %CNT, %CNT
*SET VARIABLE CNT = %EVAL (%CNT+1) (The %EVAL function is
defined so that the CNT character string will not exceed the
limit.)
*ENDW
```

10.5 SHOW VARIABLE

The **SHOW VARIABLE** command displays the definition of the specified debug variable.

■ SHOW VARIABLE

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

● Format

SHOW VARIABLE [debug-variable-name [,...]]

Parameter

debug-variable-name (identifier)

Specifies a debug variable name to be displayed.

Command qualifier

/ALL (default when omitted)

Displays all debug variables.

● Description

The **SHOW VARIABLE** command displays the definition of the specified debug variable.

● Example

```
>SET VARIABLE CNT = 1
>WHILE %CNT<5
*SHOW VARIABLE CNT
*PRINTF "CNT = %d\n", %CNT
*SET VARIABLE CNT = %CNT+1
*ENDW
CNT : 1
CNT = 1
CNT : 1+1
CNT = 2
CNT : 1+1+1
CNT = 3
CNT : 1+1+1+1
CNT = 4
>
```

10.6 CANCEL VARIABLE

The **CANCEL VARIABLE** command cancels the specified debug variable.

■ CANCEL VARIABLE

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

CANCEL VARIABLE [debug-variable-name [,...]]

Parameter

debug-variable-name (identifier)

Specifies a debug variable name to be canceled.

Command qualifier

/ALL

Cancels all debug variables.

- Description

The CANCEL VARIABLE command cancels the specified debug variable.

- Example

```
>CANCEL VARIABLE CHKADR, X, Y
>
```


CHAPTER 11 Utility Commands



This chapter describes the Utility commands.

11.1 SET LOGGING

11.2 SHOW LOGGING

11.3 CANCEL LOGGING

11.4 ENABLE LOGGING

11.5 DISABLE LOGGING

11.6 PRINTF

11.7 SET OUTPUT

11.8 SHOW OUTPUT

11.1 SET LOGGING

The **SET LOGGING** command opens the specified logging file and starts logging.

■ SET LOGGING

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

● Format

SET LOGGING [file-name]

Parameter

file-name

Specifies the name of the log file.

If the file name is omitted, perform logging with the following file name for each logging target window.

- Command window: DEBUG.LOG
- Terminal window: TERMINAL.LOG

If the path is omitted, create the file in the same location where the currently opened project file exists.

Command qualifiers

- Specifying for creating a logging file

/OPEN (default when omitted)

Newly opens specified file.

/APPEND

Appends log data to end of specified file.

- Specifying for the type of data acquisition of command window

When specifying /TERMINALWINDOW, the following command qualifiers cannot be specified.

/EXPANSION (default when omitted)

Perform logging of the command list and the result.

/UNEXPANSION

Perform logging of the result only.

/COMMAND

Perform logging of the user input only.

- Specifying the target window for logging
/COMMANDWINDOW(Default when omitted)
Starts logging for command window.
/TERMINALWINDOW
Starts logging for terminal window.
This can be specified only for products whose semihosting function is enabled.
For details, refer to "On-chip debugger (OCD)" in the hardware manual for the product type you are using.

- Description

The SET LOGGING command opens the specified logging file and starts logging.

When command qualifier /APPEND is specified, data in the previous file is not lost.

When the same file name is specified without "/APPEND", the previous file will be overwritten and its contents are lost.

The data to be logged can be selected.

Specifying command qualifier /COMMAND enables the entered command list to be used as the command procedure file because only the list is logged. Terminal window can be logged with the emulator debugger (MB2100-01).

- Example

```
>SET LOGGING filename.log  
>  
>SET LOGGING/COMMAND filename.log  
[Emulator Debugger (MB2100-01)]  
>SET LOGGING /TERMINALWINDOW
```

11.2 SHOW LOGGING

The **SHOW LOGGING** command displays the logging status.

■ SHOW LOGGING

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SHOW LOGGING

- Description

The SHOW LOGGING command displays the logging status.

The logging status of terminal window can be displayed with the products whose semihosting function is enabled.

For details, refer to "On-chip debugger (OCD)" in the hardware manual for the product type you are using.

- Example

```
>SHOW LOGGING
en/dis          :    ENABLE
logging file    :    logfile.log
logging data    :    EXPANSION
```

[Emulator Debugger (MB2100-01)]

```
>SHOW LOGGING
window          :    COMMAND
en/dis          :    ENABLE
logging file    :    comandlog.log
logging data    :    EXPANSION

window          :    TERMINAL
en/dis          :    ENABLE
logging file    :    terminallog.log
```

11.3 CANCEL LOGGING

The **CANCEL LOGGING** command cancels the logging setup and closes the logging file.

■ CANCEL LOGGING

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

CANCEL LOGGING

Command qualifiers

/COMMANDWINDOW (Default when omitted)

Cancels logging setting of command window, and closes the logging file.

/TERMINALWINDOW

Cancels logging setting of terminal window, and closes the logging file.

This can be specified only for products whose semihosting function is enabled.

For details, refer to "On-chip debugger (OCD)" in the hardware manual for the product type you are using.

- Description

Cancels logging setting of the specified window, and closes the logging file.

- Example

```
>CANCEL LOGGING
```

```
[Emulator Debugger (MB2100-01)]
```

```
>CANCEL LOGGING /TERMINALWINDOW
```

11.4 ENABLE LOGGING

The **ENABLE LOGGING** command enables logging again.

■ ENABLE LOGGING

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

ENABLE LOGGING

Command qualifiers

/COMMANDWINDOW (Default when omitted)

Enables logging for command window.

/TERMINALWINDOW

Enables logging for terminal window.

This can be specified only for products whose semihosting function is enabled.

For details, refer to "On-chip debugger (OCD)" in the hardware manual for the product type you are using.

- Description

Enables logging for the specified window.

- Example

```
>ENABLE LOGGING
```

```
[Emulator Debugger (MB2100-01)]
```

```
>ENABLE LOGGING /TERMINALWINDOW
```

11.5 DISABLE LOGGING

The **DISABLE LOGGING** command temporarily disables logging.

■ DISABLE LOGGING

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

DISABLE LOGGING

Command qualifiers

/COMMANDWINDOW (Default when omitted)

Disables logging for command window.

/TERMINALWINDOW

Disables logging for terminal window.

This can be specified only for products whose semihosting function is enabled.

For details, refer to "On-chip debugger (OCD)" in the hardware manual for the product type you are using.

- Description

Disables logging for the specified window.

The **ENABLE LOGGING** command can be used to enable logging again.

- Example

```
>DISABLE LOGGING
[Emulator Debugger (MB2100-01)]
>DISABLE LOGGING /TERMINALWINDOW
```

11.6 PRINTF

The PRINTF command displays the specified character string and the expression value of the specified format on the screen.

■ PRINTF

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

● Format

PRINTF "format-control-string" [, expression [, ...]]

Parameters

format-control-string

Specifies character strings to be displayed on screen and format for expression value display.

Enclose format specification in double quotation marks (").

"% [flag] [width] [.precision] [size] type"

%

Specifies this parameter when displaying data according to format specification.

The PRINTF command displays characters that are not format specification after % as they are.

flag

Specifies whether to right- or left-justify display, o (octal number) or 0 (hexadecimal number), and 0x, 0X output control.

When flag is omitted, the display is right-justified.

This parameter is invalid when the conversion display format is b or f.

-: Left-justification

: Adds 0, 0x, or 0X before numeric value 0 is added when the conversion display format is o. 0x is added when the format is x. 0X is added when the format is X.

width

Specifies minimum count of digits of integer to be output.

When the conversion result is less than the specified count of digits, the remaining areas are padded with 0s.

To pad with 0s at right-justification, add 0 to the beginning and specify the digits count.

When the conversion display format is b or f, width is invalid.

precision

Specifies minimum count of digits of integer to be output.

When the conversion result is less than the specified count of digits, the remaining areas are padded with 0s.

When the conversion display format is b or f, precision is invalid.

size

Specifies size of expression.

I: The expression values is assumed to be the long, unsigned long type.

II: The expression values is assumed to be the long long, unsigned long long type.

Even when size is omitted, the language expression value is assumed to be the long, unsigned long type.

type

Specifies one of following conversion display formats:

d: Signed decimal number

u: Unsigned decimal number

o: Unsigned octal number

x: Unsigned hexadecimal number (Lower-case characters a to f represent 10 to 15, respectively.)

X: Unsigned hexadecimal number (Upper-case characters A to F represent 10 to 15, respectively.)

f: floating-point number

c: One character

b: Unsigned binary number

s: Character string (Only addressing is valid. The maximum number of characters is 128 bytes.)

expression

Specifies the expression to be displayed.

- **Description**

The PRINTF command displays the specified character string and the expression value of the specified format on the screen.

- **Example**

```
>PRINTF "ABC = %d\n", datflg
ABC = 3
```

11.7 SET OUTPUT

When the user program stops, the **SET OUTPUT** command opens the source window according to the debug information at the position indicated by the PC.

■ SET OUTPUT

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SET OUTPUT

Command qualifiers

/SOURCE (default when omitted)

Opens source window in mixed mode, even if no file.

/INSTRUCTION

Opens source window as disassembly window, even if no file.

- Description

When the user program stops, the **SET OUTPUT** command opens the source window according to the debug information at the position indicated by the PC. In this case, the operation that is performed when no target source file can be found is set.

- Example

```
>SET OUTPUT /SOURCE
```

11.8 SHOW OUTPUT

The **SHOW OUTPUT** command shows the display mode set by the **SET OUTPUT** command.

■ SHOW OUTPUT

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

SHOW OUTPUT

- Description

The **SHOW OUTPUT** command shows the display mode set by the **SET OUTPUT** command.

- Example

```
>SHOW OUTPUT
source mode:  source
```


CHAPTER 12 Control Commands



This chapter describes the Control commands.

12.1 IF

12.2 REPEAT

12.3 WHILE

12.4 BREAK

12.1 IF

When formula is evaluated as true, the command list immediately after IF is executed. When formula is evaluated as false, the command list after ELSE is executed.

■ IF

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

● Format

IF formula

command-list

[ELSEIF formula

command-list]

[ELSE

command-list]

ENDIF

Parameters

formula

Specifies the execution condition formula of specified command list.

command-list

Specifies the commands to be executed.

● Description

When formula is evaluated as true, the command list immediately after IF is executed.

When formula is evaluated as false, the command list after ELSE is executed.

If formula is false when ELSE is omitted, nothing is executed.

Only macros or batch can use the IF command.

● Example

```
IF %R0 == 0
    print "OK!!"
else
    print "NG!!"
endif
```

12.2 REPEAT

The REPEAT command evaluates the UNTIL formula after the command list specified by command-list has been executed.

■ REPEAT

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

REPEAT

command-list

UNTIL formula

Parameters

command-list

Specifies the commands to be executed.

formula

Specifies the execution condition formula of specified command list.

- Description

The REPEAT command evaluates the UNTIL formula after the command list specified by command-list has been executed. This command repeats execution of the command list while the formula is false.

Only macros or batch can use the REPEAT command.

- Example

```
REPEAT
```

```
    STEP
```

```
UNTIL  %PC == main
```

12.3 WHILE

When the specified formula is evaluated as true, the WHILE command repeats execution of the specified command list.

■ WHILE

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

```
WHILE formula
    command-list
```

```
ENDW
```

Parameters

formula

Specifies the execution condition formula of specified command list.

command-list

Specifies the commands to be executed.

- Description

When the specified formula is evaluated as true, the WHILE command repeats execution of the specified command list.

Only macros or batch can use the WHILE command.

- Example

```
WHILE %PC! = function
    STEP
ENDW
```


12.4 BREAK

The **BREAK** command enables the program to exit the control structure.

■ BREAK

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

BREAK

- Description

The **BREAK** command enables the program to exit the control structure.

This command is valid only in the **REPEAT** and **WHILE** command lists.

Only macros or batch can use the **BREAK** command.

- Example

```

WHILE      1
  if %PC == main
    BREAK
  ENDIF
STEP
ENDW

```


CHAPTER 13 Built-in Variables and Functions



This chapter describes the Built-in Variables and Functions commands.

- 13.1 %CALL
- 13.2 %ERRNUM
- 13.3 %ENTRY
- 13.4 %STKTOP
- 13.5 %RADIX
- 13.6 %SCPADR
- 13.7 %LOADNUM
- 13.8 %BIT, %B, %H, %W, %S, %D
- 13.9 %STRGET
- 13.10 %STRSTR
- 13.11 %STRCMP
- 13.12 %STRLEN
- 13.13 %STRCAT
- 13.14 %SYMLen
- 13.15 %TOVAL
- 13.16 %TOSTR
- 13.17 %EVAL
- 13.18 %EVENTNUM
- 13.19 %TRIGGERNUM
- 13.20 %PFMEVENTNUM
- 13.21 %DEBUGFUNCTION
- 13.22 %GET_CORESTATE
- 13.23 %GET_TRACESTATE
- 13.24 %GET_EVENTSTATE
- 13.25 %GET_PERFORMANCESTATE

13.1 %CALL

%CALL returns the return value for the last-executed CALL command.

■ %CALL

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
	(MB2100-01)	<input type="radio"/>
Monitor		<input type="radio"/>

- Format

%CALL

- Description

%CALL returns the return value for the last-executed CALL command. If the function return values are void and double, 0 is returned.

- Example

```
>CALL func(100,200)
    return value is H'40
>ENTER val=%CALL+0x80
```

13.2 %ERRNUM

%ERRNUM replaces the error number with the last error number executed from the Command Window.

■ %ERRNUM

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%ERRNUM

- Description

%ERRNUM replaces the error number with the last error number executed from the Command Window.

0 indicates that there is no error.

- Example

```
>PRINTF "ERROR NO. = %d\n", %ERRNUM
ERROR NO. = 5
```

13.3 %ENTRY

%ENTRY replaces the execution starting address with the execution starting address of the load module being loaded.

■ %ENTRY

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%ENTRY

- Description

%ENTRY replaces the execution starting address with the execution starting address of the load module being loaded.

0 indicates that there is no execution starting entry.

- Example

```
>PRINTF "ENTRY = 0x%X\n", %ENTRY
ENTRY = 0x10000
```

13.4 %STKTOP

%STKTOP replaces the starting address with the starting address of the stack area of the load module being loaded.

■ %STKTOP

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%STKTOP

- Description

%STKTOP replaces the starting address with the starting address of the stack area of the load module being loaded.

0 indicates that there is no stack area.

- Example

```
>PRINTF "STACK = 0x%X\n", %STKTOP
STACK = 0x80000
```

13.5 %RADIX

%RADIX replaces the base number with the currently-set base number (BINARY, OCTAL, DECIMAL, or HEXADECIMAL).

■ %RADIX

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%RADIX

- Description

%RADIX replaces the base number with the currently-set base number (BINARY, OCTAL, DECIMAL, or HEXADECIMAL).

- Example

```
>PRINTF "base-number = "
>PRINTF %TOSTR(%RADIX)
base-number = HEXADECIMAL
```


13.6 %SCPADR

%SCPADR replaces the scope address with the current scope address.

■ %SCPADR

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%SCPADR

- Description

%SCPADR replaces the scope address with the current scope address.

- Example

```
>PRINTF " scope = 0x%X\n", %SCPADR
scope = 0x18300
```

13.7 %LOADNUM

%LOADNUM replaces the size with the size of the last loaded binary file.

■ %LOADNUM

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%LOADNUM

- Description

%LOADNUM replaces the size with the size of the last loaded binary file.

- Example

```
>PRINTF "byte-count = %d\n", %LOADNUM  
byte-count = 584
```

13.8 %BIT, %B, %H, %W, %S, %D

%BIT, %B, %H, %W, %S, or %D replaces the data with memory data read from the specified address.

■ %BIT, %B, %H, %W, %S, %D

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙*
	(MB2100-01)	⊙
Monitor		○

*: The DSU3 evaluation chip cannot specify this command during the program running.

● Format

%BIT(address)

%B(address)

%H(address)

%W(address)

%S(address)

%D(address)

Parameter

address

Specifies the address from where memory data to be read.

● Description

%BIT, %B, %H, %W, %S, or %D replaces the data with any of the following memory data read from the specified address:

%BIT : Bit data

%B : Byte (8 bits) data

%H : Halfword (16 bits) data

%W : Word (32 bits) data

%S : Single-precision floating-point number data

%D : Double-precision floating-point number data

● Example

```
>PRINTF "10000 = 0x%X\n", %W(10000)
10000 = 0xAABBAACC
```

13.9 %STRGET

%STRGET replaces the character string in the specified count of characters, starting from the specified character position in the specified character string.

■ %STRGET

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%STRGET(character-string, character-position, character-count)

Parameters

character-string

Specifies a replacing character string.

character-position

Specifies a character position where get processing to be started (character position relative to first character).

character-count

Specifies a count of characters to be gotten.

- Description

%STRGET replaces the character string in the specified count of characters, starting from the specified character position in the specified character string.

- Example

```
>PRINTF %TOSTR(%STRGET("abcdefghijklmn", 3, 4) )
cdef
```

13.10 %STRSTR

%STRSTR checks whether character-string-1 includes character-string-2.

■ %STRSTR

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%STRSTR(character-string-1, character-string-2)

Parameters

character-string-1

Specifies a character string including character string to be retrieved.

character-string-2

Specifies a character string to be retrieved.

- Description

%STRSTR checks whether character-string-1 includes character-string-2.

When character-string-1 includes character-string-2, %STRSTR replaces the character position number with the character position number in character-string-1.

When character-string-1 does not include character-string-2, %STRSTR replaces the character position number with 0.

- Example

```
>PRINTF "%d\n", %STRSTR("abcdefghijklmn", "fg")
6
```

13.11 %STRCMP

%STRCMP compares character-string-1 with character-string-2.

■ %STRCMP

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%STRCMP(character-string-1, character-string-2)

Parameter

character-string-1, character-string-2

Specifies a character strings to be compared.

- Description

%STRCMP compares character-string-1 with character-string-2.

When the character strings match, %STRCMP sets 0. When they do not match, %STRCMP sets 1.

- Example

```
>PRINTF "%d\n", %STRCMP("abcde", "fg")
1
>PRINTF "%d\n", %STRCMP("abcde", "abcde")
0
```

13.12 %STRLEN

%STRLEN replaces character string with the count of characters.

■ %STRLEN

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

`%STRLEN(character-string)`

Parameter

character-string

Specifies a replacing character string.

- Description

%STRLEN replaces character string with the count of characters.

- Example

```
>PRINTF "%d\n", %STRLEN("abcde")
5
```

13.13 %STRCAT

%STRCAT replaces character string with character string created by linking character-string-1 and character-string-2.

■ %STRCAT

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%STRCAT(character-string-1, character-string-2)

Parameter

character-string-1, character-string-2

Specifies a character strings to be linked.

- Description

%STRCAT replaces character string with character string created by linking character-string-1 and character-string-2.

- Example

```
>PRINTF %TOSTR(%STRCAT ("abcde", "fg") )
abcdefg
```


13.14 %SYMLEN

%SYMLEN returns the size of a specified symbol.

■ %SYMLEN

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%SYMLEN(symbol-name)

Parameter

symbol-name

Specifies a symbol.

- Description

%SYMLEN returns the size of a specified symbol.

- Example

```
>PRINTF "%d\n", %SYMLEN("abcde")
2
```

13.15 %TOVAL

%TOVAL deletes double quotation marks (") from both ends of the specified character string.

■ %TOVAL

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%TOVAL(character-string)

Parameter

character-string

Specifies a character string.

- Description

%TOVAL deletes double quotation marks (") from both ends of the specified character string.

This function is used when the character string enclosed in double quotation marks is specified in a field where only parameters other than character strings can be written.

- Example

```
>SET BREAK %TOVAL("main")
```

13.16 %TOSTR

%TOSTR encloses the specified character string in double quotation marks (").

■ %TOSTR

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%TOSTR(character-string)

Parameter

character-string

All parameter types can be specified.

- Description

%TOSTR encloses the specified character string in double quotation marks (").

This function is used when the specified character string is specified in a field where only character strings can be written as parameters.

- Example

```
>PRINTF %TOSTR(main)
main
```

13.17 %EVAL

%EVAL evaluates specified expression.

■ %EVAL

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

`%EVAL(expression)`

Parameter

expression

Specifies an expression to be evaluated.

- Description

%EVAL evaluates specified expression.

- Example

```
>PRINTF "%d\n", %EVAL(10+20+30)
60
```

13.18 %EVENTNUM

%EVENTNUM returns the number of event.

■ %EVENTNUM

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%EVENTNUM(address)

Parameter

address

Specifies the address at which the event number automatically set is read.

- Description

[Emulator Debugger (MB2198)]

%EVENTNUM returns the number of the event set by SET EVENT.

It can also be used as a parameter for the following commands:

SET SEQUENCE

SHOW EVENT

CANCEL EVENT

ENABLE EVENT

DISABLE EVENT

[Emulator Debugger (MB2100-01)]

%EVENTNUM returns the number of the event set by SET EVENT /SEQUENCE.

It can also be used as a parameter for the following commands:

SET SEQUENCE

SHOW EVENT /SEQUENCE

CANCEL EVENT /SEQUENCE

- Example

```
>PRINTF "event no : %0d\n", %EVENTNUM(F0000)
event no : 1
```

Note:

In the following cases, "0" returns regardless of the kind of the debugger.

- Specifies invalid address.
 - Specifies the address where the event is not set.
 - The event is not found.
-

13.19 %TRIGGERNUM

%TRIGGERNUM returns the number of trace trigger.

■ %TRIGGERNUM

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%TRIGGERNUM(address)

Parameter

address

Specifies the address at which the trace trigger number automatically set is read.

- Description

%TRIGGERNUM returns the number of the trace trigger set by SET TRACETRIGGER.

It can also be used as a parameter for the following commands:

SHOW TRACETRIGGER

CANCEL TRACETRIGGER

ENABLE TRACETRIGGER

DISABLE TRACETRIGGER

- Example

```
>PRINTF "trace trigger no : %0d\n", % TRIGGERNUM(F0000)
trace trigger no : 2
```

Note:

In the following cases, "0" returns regardless of the kind of the debugger.

- Specifies invalid address.
 - Specifies the address where the trace trigger is not set.
 - The trace trigger is not found.
-

13.20 %PFMEVENTNUM

%TRIGGERNUM returns the number of the performance trigger at the specified address.

■ %PFMEVENTNUM

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

● Format

%PFMEVENTNUM(address)

Parameter

address

Specifies the address at which the performance trigger number automatically set is read.

● Description

%PFMEVENTNUM returns the number of the performance trigger set by SET EVENT /PERFORMANCE.

1 to 8 refers to a code event, while 9 to 12 refers to a data event.

It can also be used as a parameter for the following commands:

SET PERFORMANCE

SHOW EVENT /PERFORMANCE

CANCEL EVENT /PERFORMANCE

● Example

```
>PRINTF "performance event no : %d\n", %PFMEVENTNUM (F0000)
performance event no : 1
```

Note:

In the following cases, "0" returns regardless of the kind of the debugger.

- Specifies invalid address.
- Specifies the address where the performance trigger is not set.
- Uses an emulator debugger other than MB2100-01.

In the following cases, "1" returns regardless of the kind of the debugger.

- Measurement state can not be attained.

13.21 %DEBUGFUNCTION

This replaces the current debug function number.

■ %DEBUGFUNCTION

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
	(MB2100-01)	×
Monitor		×

- Format

%DEBUGFUNCTION

- Description

This replaces the current debug function number.

- Example

```
>PRINTF "debug function : %d\n",%DEBUGFUNCTION
debug function : 0
```

13.22 %GET_CORESTATE

%GET_CORESTATE is a built-in variable for retrieving the run state of the core.

■ %GET_CORESTATE

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%GET_CORESTATE

- Description

%GET_CORESTATE retrieves the run state of the core.

This returns 1 if the core is running and 0 if the core is in a break.

- Example

```
> PRINTF " CORE_STATUS : %d\n", %GET_CORESTATE
CORE_STATUS : 0
```

13.23 %GET_TRACESTATE

%GET_TRACESTATE is a built-in variable for retrieving the state of the trace function.

■ %GET_TRACESTATE

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%GET_TRACESTATE

- Description

%GET_TRACESTATE retrieves the state of the trace function.

- If the trace function is retrieved: 1
- If the trace function is stopped: 0
- If the state of the trace function cannot be retrieved: -1

- Example

```
> PRINTF " TRACE_STATUS : %d\n", %GET_TRACESTATE
TRACE_STATUS : 0
```

13.24 %GET_EVENTSTATE

%GET_EVENTSTATE returns the hit information of the code event and the data event.

■ %GET_EVENTSTATE

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

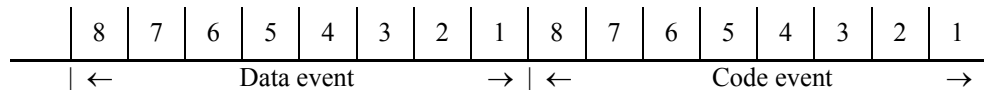
- Format

%GET_EVENTSTATE

- Description

Return the hit information of the set code event and the set data event.

- When becoming a hit to the event: 0
- When not becoming a hit to the event: 1



Confirm to which event to have become a hit by the following bit allocations.

- Bit 0 to bit 7: Code event (1 to 8)
- Bit 8 to bit 15: Data event (9 to 16)
- Bit 16 or more : Reserved bits

Confirm by SHOW EVENT command (See "3.16 SHOW EVENT") for which function the hit event number is used.

- Example

In case of the confirmation whether the set performance event was hit

```
> IF ((1<<(%PFMEVENTNUM (F0000) - 1)) & %GET_EVENTSTATE) != 0
> PRINTF "performance event hit\n"
> ENDIF
```

Notes:

- The value of the reserve bit is infinite.
- "0" returns in the following cases regardless of the hit state of the event.
 - Emulators other than emulator debugger (MB2100-01) are used.
 - The program is break.
 - The hit information can not be attained.

13.25 %GET_PERFORMANCESTATE

%GET_PERFORMANCESTATE returns the state whether the performance measurement count reaches in the maximum.

■ %GET_PERFORMANCESTATE

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
	(MB2100-01)	⊙
Monitor		⊙

- Format

%GET_PERFORMANCESTATE

- Description

The state whether the performance measurement count reaches in the maximum is returned. When the measurement count does not reach in the maximum, it is shown that the performance measurement can be continued.

The following numbers are returned according to the state.

- 1: Performance measurement is continuing.
- 0: Performance measurement ends.
- 1: The measurement state is not returned.

- Example

```
> PRINTF " PERFORMANCE_STATUS : %d\n", %GET_PERFORMANCESTATE
PERFORMANCE_STATUS : 1
```


APPENDIX



These appendixes describe the Manager-Related Messages, Error Message for Debuggers, and Execution Suspension Message List.

APPENDIX A MANAGER-RELATED MESSAGES

APPENDIX B ERROR MESSAGE FOR DEBUGGERS

APPENDIX C EXECUTION SUSPENSION MESSAGES LIST

APPENDIX D MAJOR CHANGES

APPENDIX A MANAGER-RELATED MESSAGES

This appendix describes the Manager-Related Messages.

■ Manager-Related Messages

E4002W	Insufficient memory.
[Explanation]	System memory is insufficient.
[Operator response]	Terminate another program and execute this program.

E4011W	Registration not possible.
[Explanation]	Data cannot be written to the system registry.
[Operator response]	Terminate another program and execute this program.

E4012W	Failed function call. Exe file is old.
[Explanation]	The version of the program file does not correspond to that of the DLL file.
[Operator response]	Install the latest version of SOFTUNE Workbench.

E4013W	Failed function call. DLL file is old.
[Explanation]	The version of the program file does not correspond to that of the DLL file.
[Operator response]	Install the latest version of SOFTUNE Workbench.

E4020W	CPU information file version is different. Contains uninterpretable information.
[Explanation]	The CPU information file is old and does not contain the required information.
[Operator response]	Get the latest CPU information file.

E4021W	Chip type in CPU information file is not applicable.
[Explanation]	Information for a different CPU is specified.
[Operator response]	Specify the correct CPU information file.

E4022W	Please enter CPU information file.
[Explanation]	The CPU information file cannot be found.
[Operator response]	Enter the CPU information file directory.

E4023W	Illegal tool option data. Default data is set.
[Explanation]	The project file has illegal tool option data.
[Operator response]	Reset the tool option data.

E4024W	Invalid CPU information. Set default value.
[Explanation]	The CPU information file has illegal data.
[Operator response]	Get the latest CPU information file.

E4100W	Access was denied.
[Explanation]	The file cannot be accessed.
[Operator response]	The file may be write- or read-disabled. Check the file attributes.
E4110W	Too many open files.
[Explanation]	The maximum number of files that can be opened is exceeded.
[Operator response]	Close other files.
E4120W	Directory does not exist.
[Explanation]	The directory cannot be found.
[Operator response]	Enter the correct directory name.
E4121W	Drive is not ready.
[Explanation]	The drive cannot be accessed.
[Operator response]	Check the drive.
E4122W	Path is invalid.
[Explanation]	The directory cannot be found.
[Operator response]	Enter the correct directory name.
E4123W	Unable to create directory.
[Explanation]	The directory cannot be created.
[Operator response]	The directory may be write-disabled, or a file in the directory may be in use by another process.
E4124W	Unable to delete directory.
[Explanation]	The directory cannot be deleted.
[Operator response]	The directory may be write-disabled, or a file in the directory may be in use by another process.
E4125W	Destination disk is full.
[Explanation]	The remaining capacity of the disk is insufficient.
[Operator response]	Delete unnecessary files.
E4126W	Could not be removed because it is the current directory.
[Explanation]	An attempt was made to delete the current directory.
[Operator response]	Move from the current directory to delete another directory.
E4127W	This directory cannot be access.
[Explanation]	Access to the directory is denied.
[Operator response]	Permission to access the directory may be denied.

E4130W	File cannot be open.
--------	----------------------

[Explanation] The file cannot be opened.

[Operator response] Permission to access the file or directory may be denied.

E4131W	File cannot be close.
--------	-----------------------

[Explanation] The file cannot be closed.

[Operator response] Permission to access the file or directory may be denied.

E4132W	File cannot be read.
--------	----------------------

[Explanation] The file cannot be read.

[Operator response] Permission to access the file or directory may be denied.

E4133W	File cannot be written.
--------	-------------------------

[Explanation] The file cannot be written.

[Operator response] Permission to access the file or directory may be denied.

E4134W	File cannot be create.
--------	------------------------

[Explanation] The file cannot be created.

[Operator response] Permission to access the file or directory may be denied.

E4135W	File cannot be delete.
--------	------------------------

[Explanation] The file cannot be deleted.

[Operator response] Permission to access the file or directory may be denied.

E4136W	File cannot be change name.
--------	-----------------------------

[Explanation] The file cannot be renamed.

[Operator response] Permission to access the file or directory may be denied.

E4137W	File cannot be copied.
--------	------------------------

[Explanation] The file cannot be copied.

[Operator response] Permission to access the file or directory may be denied.

E4138W	File not found.
--------	-----------------

[Explanation] The file cannot be found.

[Operator response] Check the file name.

E4140W	File not found. Do you create this file?
--------	--

[Explanation] The file cannot be found.

[Operator response] To create a new file, click the [OK] button.

E4142W	A sharing violation occurred while accessing.
--------	---

[Explanation] The same file is being used by another process.

[Operator response] Terminate the other program. In some rare cases, the file may remain in use even after the program is terminated. In this case, reboot Windows.

E4143W	A locking violation occurred while accessing.
[Explanation]	The same file is being used by another process.
[Operator response]	Terminate the other program. In some rare cases, the file may remain in use even after the program is terminated. In this case, reboot windows.
E4200W	The project file format is illegal.
[Explanation]	The projection file cannot be read properly.
[Operator response]	The projection file may be different from that for SOFTUNE Workbench or may be damaged. Create a new project file.
E4201W	Project file cannot be opened - CPU type is different.
[Explanation]	The projection file is different from that for the MCU.
[Operator response]	Create a new project file for the MCU.
E4202W	Unable to save project file.
[Explanation]	An error occurred at writing to the project file.
[Operator response]	The remaining disk capacity may be insufficient or the project file may be write-disabled.
E4204W	Illegal CPU information of project file. Setting default value.
[Explanation]	CPU information in the project file is illegal, and is substituted for the default.
[Operator response]	Check the set value for CPU information in the project file.
E4205W	Target file directory not found. Create a directory?
[Explanation]	The target project file directory is not specified.
[Operator response]	To create a directory, click the [OK] button.
E4206W	List file directory not found. Create a directory?
[Explanation]	The target list file directory is not specified.
[Operator response]	To create a directory, click the [OK] button.
E4207W	Object file directory not found. Create a directory?
[Explanation]	The target object file directory is not specified.
[Operator response]	To create a directory, click the [OK] button.
E4210W	Please specify the project name.
[Explanation]	The project name is not specified.
[Operator response]	Enter the project name.
E4211W	Please specify the project directory.
[Explanation]	The project directory is not specified.
[Operator response]	Enter the project directory name.

E4212W	Please specify the target file name.
--------	--------------------------------------

[Explanation] The target file name is not specified.

[Operator response] Enter the target file name.

E4213W	Includes characters that cannot be designated. ¥ / : , ; * ? " " < >
--------	--

[Explanation] These characters cannot be used.

[Operator response] Change the name.

E4214W	Includes characters that cannot be designated. , ; * ? " " < >
--------	--

[Explanation] These characters cannot be used.

[Operator response] Change the name.

E4215W	Includes characters that cannot be designated. , ; * ? " " < >
--------	--

[Explanation] These characters cannot be used.

[Operator response] Change the name.

E4220W	Please specify the target file name.
--------	--------------------------------------

[Explanation] The target file name is not specified.

[Operator response] Enter the target file name.

E4221W	Directory not found. Do you create this directory?
--------	--

[Explanation] The directory is not specified.

[Operator response] Enter the directory name.

E4222W	Unable to create directory.
--------	-----------------------------

[Explanation] The directory cannot be created.

[Operator response] The file may be write-disabled.

E4223W	Changed target MCU. CPU information changed to default value.
--------	--

[Explanation] When the target MCU is changed, the preset CPU information returns to the default.

[Operator response] Reset the CPU information.

E4224W	Specify target MCU.
--------	---------------------

[Explanation] The target MCU is not specified.

[Operator response] Enter the target MCU name.

E4225W	Specify project type.
--------	-----------------------

[Explanation] The project type is not specified.

[Operator response] Specify the project type.

E4226W	Includes characters that cannot be designated. , ; * ? " ' < >
[Explanation]	These characters cannot be used.
[Operator response]	Change the name.
E4227W	Please specify Object File Directory.
[Explanation]	The target object file directory is not specified.
[Operator response]	Enter the directory name.
E4228W	Please specify List File Directory.
[Explanation]	The target list file directory is not specified.
[Operator response]	Enter the directory name.
E4230W	Double specification.
[Explanation]	The same specification is already in use.
[Operator response]	Change the specification.
E4232W	Setup file is not registered. Registered automatically.
[Explanation]	Starting the debugger requires a setup file. If a setup file is not specified, create it with the same name as that of the project file.
[Operator response]	Use [Project]-[Setup Project]-[Debug] to set the items required for the automatically-created setup file.
E4233W	Available setup file is not registered. Registered automatically.
[Explanation]	Starting the debugger requires a setup file. If a setup file is not specified, create it with the same name as that of the project file.
[Operator response]	Use [Project]-[Setup Project]-[Debug] to set the items required for the automatically-created setup file.
E4234W	Please specify the title.
[Explanation]	The title is not specified.
[Operator response]	Specify the title.
E4240W	Already a registered member.
[Explanation]	The specified file is already saved in the project.
[Operator response]	Check the file name.
E4241W	This file name has already been registered.
[Explanation]	The specified file is already saved in the project.
[Operator response]	Check the file name.
E4242W	File not found. Do you registered?
[Explanation]	An attempt was made to save a non-existent file in the project.
[Operator response]	If the file name is correct, save the file. An inquiry is made as to whether to create a new file when starting the editor.

E4243W	Too many select files.
[Explanation]	The count of selected files exceeds the maximum value.
[Operator response]	Decrease the count of selected files.

E4301W	Unable to create command line.
[Explanation]	The option file to start the language tool cannot be created.
[Operator response]	Check the access permission for the OPT subdirectory under the project directory, or the disk capacity.

E4302W	Failed during start.
[Explanation]	The tool cannot be started.
[Operator response]	The tool name may be incorrect. Check the tool settings.

E4303W	Command Line too long.
[Explanation]	The command line is too long (max. 2048 characters).
[Operator response]	Check the option parameters.

E4304W	Failed during start editor.
[Explanation]	The saved external editor cannot be started.
[Operator response]	Check the executable file name of the editor.

E4305W	Compiler/Assembler is started.
[Explanation]	An attempt is made to close the project during tool start up.
[Operator response]	Use the Suspend button to terminate the tool and close the project.

E4306W	Make function is started.
[Explanation]	An attempt is made to close the project during tool start up.
[Operator response]	Use the Suspend button to terminate the tool and close the project.

E4307W	Build function is started.
[Explanation]	An attempt is made to close the project during tool start up.
[Operator response]	Use the Suspend button to terminate the tool and close the project.

E4308W	Include Dependencies is started.
[Explanation]	An attempt is made to close the project during tool start up.
[Operator response]	Use the Suspend button to terminate the tool and close the project.

E4309W	Tool is started.
[Explanation]	An attempt is made to close the project during tool start up.
[Operator response]	Use the Suspend button to terminate the tool and close the project.

E4400W	Setup file is read only. Setup information is not saved.
[Explanation]	The setup file cannot be written.
[Operator response]	Set the setup file to the write-enabled state.

E4401W	Setup file not found.
[Explanation]	The setup file was not found.
[Operator response]	It may not be that the SOFTUNE Workbench was installed completely. Install it again.
E4402W	The setup file format is illegal.
[Explanation]	The setup file has invalid data.
[Operator response]	It may not be that the SOFTUNE Workbench was installed completely. Install it again.
E4420W	Maximum of address is xxxx.
[Explanation]	The address exceeds the maximum value.
[Operator response]	Check the address specification.
E4421W	The start address exceeds the end address.
[Explanation]	The specified address range is incorrect.
[Operator response]	Check the address range specification.
E4422W	The designated address is already designated.
[Explanation]	The specified address range has already been saved.
[Operator response]	Check the address range.
E4601W	Double specification.
[Explanation]	The specified item has already been saved.
[Operator response]	Check the specification contents.
E4603W	Illegal tool option data.
[Explanation]	The tool option data does not have the necessary data.
[Operator response]	Open the Tool Option Check dialog and click the OK button. When the control data is displayed, input the necessary data.
E4604W	There is no control data.
[Explanation]	Unspecified control data is found.
[Operator response]	Specify the control data.
E4605W	Includes characters that cannot be designated.
[Explanation]	These characters cannot be used.
[Operator response]	Change the name.
E4606W	Includes characters that cannot be designated. , ; * ? " ' < >
[Explanation]	These characters cannot be used.
[Operator response]	Change the name.
E4607W	Includes characters that cannot be designated. , ; * ? " ' < >
[Explanation]	These characters cannot be used.

[Operator response] Change the name.

E4610W	The range of the number of lines is 20-255.
--------	---

[Explanation] The count of lines exceeds the limit.

[Operator response] Change the count of lines.

E4611W	The range of the number of columns is 80-1023.
--------	--

[Explanation] The count of lines exceeds the limit.

[Operator response] Change the count of lines.

E4612W	The range of the number of columns is 70-1023.
--------	--

[Explanation] The count of lines exceeds the limit.

[Operator response] Change the count of lines.

E4613W	The range of the number of tabs is 0-32.
--------	--

[Explanation] The count of lines exceeds the limit.

[Operator response] Change the count of lines.

E4614W	Please specify the macro name.
--------	--------------------------------

[Explanation] The macro name is not specified.

[Operator response] Specify the macro name.

E4615W	Please specify the include path.
--------	----------------------------------

[Explanation] The install path is not specified.

[Operator response] Specify the install path.

E4616W	Already a registered macro name. Do you change contents?
--------	--

[Explanation] The specified macro name has already been saved.

[Operator response] To change the setting, click the [OK] button.

E4620W	Please specify the start address.
--------	-----------------------------------

[Explanation] The start address is not specified.

[Operator response] Specify the start address.

E4621W	Please specify the end address.
--------	---------------------------------

[Explanation] The end address is not specified.

[Operator response] Specify the end address.

E4622W	The start address is larger than the end address.
--------	---

[Explanation] The address range is incorrect.

[Operator response] Specify the address range.

E4623W	Please specify a correct start address.
--------	---

[Explanation] The start address is incorrect.

[Operator response] Specify the correct start address.

E4624W	Please specify a correct end address.
--------	---------------------------------------

[Explanation] The end address is incorrect.

[Operator response] Specify the correct end address.

E4625W	Please specify the ROM/RAM area name.
--------	---------------------------------------

[Explanation] The ROM/RAM area name is not specified.

[Operator response] Specify the ROM/RAM area name.

E4626W	Please specify the section name.
--------	----------------------------------

[Explanation] The section name is not specified.

[Operator response] Specify the section name.

E4627W	Maximum of address is 0xFFFFFFFF.
--------	-----------------------------------

[Explanation] The address exceeds the maximum value.

[Operator response] Check the address specification.

E4628W	Maximum of address is 0xFFFFF.
--------	--------------------------------

[Explanation] The address exceeds the maximum value.

[Operator response] Check the address specification.

E4629W	Maximum of address is 0xFFFF.
--------	-------------------------------

[Explanation] The address exceeds the maximum value.

[Operator response] Check the address specification.

E4630W	Cannot specify address over bank.
--------	-----------------------------------

[Explanation] The specified address crosses several banks.

[Operator response] Specify an address within one bank.

E4631W	Specify symbol name.
--------	----------------------

[Explanation] The symbol name is not specified.

[Operator response] Specify the symbol name.

E4632W	Specify set value.
--------	--------------------

[Explanation] The set value is not specified.

[Operator response] Specify the set value.

E4633W	Incorrect setting in area list. Please change setting.
--------	--

[Explanation] Some ROM/RAM area settings cannot be converted to address.

[Operator response] Check the address.

E4635W	This symbol name has already been registered. Change the setting?
--------	--

[Explanation] The specified symbol name has already been saved.

[Operator response] To change the setting, click the [OK] button.

E4636W	This ROM/RAM area name has already been registered. Change the setting?
--------	--

[Explanation] The specified ROM/RAM area name has already been saved.

[Operator response] To change the setting, click the [OK] button.

E4637W	This section name has already been registered. Change the setting?
--------	---

[Explanation] The specified section name has already been saved.

[Operator response] To change the setting, click the [OK] button.

E4638W	Address must be specified to leader section name.
--------	---

[Explanation] The address is not specified in the leading section name.

[Operator response] Specify the address.

E4639W	This section name has already been specified in another ROM/RAM area.
--------	---

[Explanation] The specified ROM/RAM area name has already been saved.

[Operator response] Check the ROM/RAM area name.

E4640W	Specify exact address.
--------	------------------------

[Explanation] The address specification is incorrect.

[Operator response] Specify the correct address.

E4641W	Maximum of value is 127.
--------	--------------------------

[Explanation] A value more than the maximum value (127) or an invalid value is specified for the number of lines for the target function of inline expansion.

[Operator response] Specify the number of lines between 0 to 127.

E4642W	The name of the ROM/RAM area across the internal ROM/RAM address range is entered. Do you want to enter this name?
[Explanation]	The area other than the internal ROM/RAM area is specified.
[Operator response]	<p>The areas other than the internal ROM/RAM area are shown below.</p> <ul style="list-style-type: none"> • Areas which can be accessed (the outer bus area where ROM, RAM or I/O is placed) • Other areas <ul style="list-style-type: none"> - Areas which cannot be accessed - Reserved areas for debug system (only for F2MC-16FX, the address between 0xDF0100 and 0xDF01FF) <p>Press the Yes button to register only for areas which can be accessed.</p>
E4701W	Specified directory does not exist. Specify?
[Explanation]	A non-existent directory is specified.
[Operator response]	If there is no error, click the OK button.
E4702W	Cannot specify multiple directories.
[Explanation]	Only one directory can be specified.
[Operator response]	Specify only one directory.
E4703W	Illegal Environment Variable.
[Explanation]	The set value is illegal.
[Operator response]	Check the set value.
E4740W	This executable file does not exist. Register in the list?
[Explanation]	The file in the execution file name cannot be found.
[Operator response]	Check the file name.
E4741W	Title is not specified.
[Explanation]	The title is not specified.
[Operator response]	Specify the title.
E4742W	Executable file is not specified.
[Explanation]	An execution file name is not specified.
[Operator response]	Specify an execution file name.
E4743W	The registration count is maximum. You cannot register any more.
[Explanation]	No more settings can be saved.
[Operator response]	Delete unnecessary settings.

E4744W	Syntax error. Illegal macro is specified.
[Explanation]	An undefined option and macro description are found in the execution directory.
[Operator response]	Check the syntax.
E4745W	Title is too long.
[Explanation]	The title is too long.
[Operator response]	Shorten the title.
E4746W	Execute file name is too long.
[Explanation]	The execution file name is too long.
[Operator response]	Shorten the file name.
E4747W	Option too long.
[Explanation]	The option string is too long.
[Operator response]	Shorten the option string.
E4748W	The executing directory too long.
[Explanation]	The directory name is too long.
[Operator response]	Shorten the directory name.
E4749W	Directory not found. Create this directory?
[Explanation]	The specified directory cannot be found.
[Operator response]	If the directory is correct, click the [OK] button.
E4750W	Already a registered title. Do you change contents?
[Explanation]	The specified title has already been saved.
[Operator response]	To change the setting, click the [OK] button.
E4752W	Start tool does not exist.
[Explanation]	The tool to be started cannot be found.
[Operator response]	Check the saved tool name and directory name.
E4760W	The registered error syntax format cannot be converted.
[Explanation]	The error message in the output window cannot be analyzed.
[Operator response]	Check the setting in the syntax list in [Setup]-[Error].
E4761W	Syntax error. Undefined Macro.
[Explanation]	An undefined macro is specified.
[Operator response]	Check the syntax.

E4762W	Syntax error. Undefined separate of '%f', '%*'. [Explanation] The delimiter indicating the end of %f and %* is not input. [Operator response] The description of the macros, %f and %*, needs the delimiter to identify the end of %f and %*. The next character in the macro description is regarded as the delimiter.
E4763W	Syntax error. Duplicate Macro syntax. [Explanation] The macros, %f, %l, and %h, are duplicated. [Operator response] Check the syntax.
E4764W	Syntax error. Invalid '¥' syntax . [Explanation] ¥ is used for other than ¥t, ¥], and ¥¥. [Operator response] Check the syntax.
E4765W	Syntax error. Invalid '%[]' syntax. [Explanation] The description of the macro, %[], is illegal. [Operator response] There may be no correspondence in []. Check the syntax.
E4766W	Syntax error. Do not describe '%f'. [Explanation] The macro, %f or %h, is not described. [Operator response] Always describe %f or %h in the error jump setting syntax.
E4767W	Syntax error. Invalid Macro into '%[...]'. [Explanation] An illegal macro is described in the macro, %[]. [Operator response] Only the macro, %% or %[] can be described in the macro, %[].
E4768W	Already a registered syntax. Do you change contents? [Explanation] The same syntax has already been saved. [Operator response] To change the contents, click the [OK] button.
E4769W	Syntax not specified. [Explanation] The syntax is not specified. [Operator response] Specify the syntax.
E4771W	Syntax too long. [Explanation] The character string in the syntax is too long. [Operator response] Shorten the syntax.
E4772W	Comment too long. [Explanation] The comment is too long. [Operator response] Shorten the comment.
E4773W	The registration count is maximum. You cannot register any more.

[Explanation] The count of saved settings exceeds the maximum value.
[Operator response] Check unnecessary settings.

E4774W	The same syntax has already been set in the SYSTEM. It cannot be changed.
--------	--

[Explanation] The same syntax has already been set in the SYSTEM.
[Operator response] Syntax that has already been saved in the SYSTEM cannot be changed.

E4780W	Title not specified.
--------	----------------------

[Explanation] The title is not specified.
[Operator response] Specify the title.

E4781W	Execute filename not specified.
--------	---------------------------------

[Explanation] The execution file name is not specified.
[Operator response] Specify the execution file name.

E4782W	Option not specified.
--------	-----------------------

[Explanation] The option is not specified.
[Operator response] Specify the option.

E4783W	Already a registered title. Do you change contents?
--------	---

[Explanation] The specified title has already been saved.
[Operator response] To change the setting, click the [OK] button.

E4784W	Syntax error. Undefined Macro.
--------	--------------------------------

[Explanation] An undefined macro is specified.
[Operator response] Check the syntax.

E4785W	Syntax error. Duplicate Macro syntax.
--------	---------------------------------------

[Explanation] The macros, %f, %l, and %h, are duplicated.
[Operator response] Check the syntax.

E4786W	Syntax error. Do not describe '%f'.
--------	-------------------------------------

[Explanation] The macro, %f or %h, is not described.
[Operator response] Always describe %f or %h in the error jump setting syntax.

E4789W	The registration count is maximum. You cannot register any more.
--------	---

[Explanation] The count of saved settings exceeds the maximum value.
[Operator response] Delete unnecessary settings.

E4790W	Editor in list not selected.
--------	------------------------------

[Explanation] The editor to be operated is not specified.
[Operator response] Select the required editor from the editor list and operate it.

E4791W	The standard editor cannot delete and change.
[Explanation]	An attempt was made to delete or change the standard editor.
[Operator response]	The standard editor is built into SOFTUNE Workbench. It cannot be deleted or changed.
E4792W	This executable file does not exist. Register in the list?
[Explanation]	The specified execution file cannot be found.
[Operator response]	If the execution file name or directory name has no error, save it as it is.
E4793W	The valid editor cannot delete.
[Explanation]	An attempt was made to delete the editor selected as the "editor to be used."
[Operator response]	Change the "editor to be used" to another before deleting it.
E4794W	Directory not found. Create this directory?
[Explanation]	The specified director cannot be found.
[Operator response]	To create a directory, click the [OK] button.
E4795W	Title too long.
[Explanation]	The title exceeds the maximum count of characters.
[Operator response]	Shorten the title.
E4796W	Execute file name too long.
[Explanation]	The execution file name is too long.
[Operator response]	Shorten the execution file name.
E4797W	Option string too long.
[Explanation]	The option string is too long.
[Operator response]	Shorten the option string.
E4798W	The executing directory too long.
[Explanation]	The directory name is too long.
[Operator response]	Shorten the directory name.
E4804W	The new version workspace/project file was opened. It can not be read by SOFTUNE V3/V6.
[Explanation]	The new version workspace/project file was opened. It can not be read by SOFTUNE V3/V6
[Operator response]	Please use SOFTUNE V7 or later to open this workspace/project file.

APPENDIX B ERROR MESSAGE FOR DEBUGGERS

This appendix describes the Error Message for Debuggers.

■ Error Message for Debuggers

F9201S	Invalid setup file (not found).
[Explanation]	The specified setup file could not be found.
[Operator response]	Check that the file specified in the startup option setup file specification exists.
F9202S	Invalid command or parameter (in setup file).
[Explanation]	An invalid command or parameter exists in the setup file. Alternately, the parameter or command must be changed because the MCU configuration is changed.
[Operator response]	Use the Setup Wizard to restart the SOFTUNE Workbench.
F9203S	Invalid machine program (execution error).
[Explanation]	The machine program is already executed or it cannot be executed because the system resources are insufficient.
[Operator response]	Check the execution state of the machine program. If the machine program is not executed, close the View Window or terminate another startup program.
F9204S	The disk space is insufficient.
[Explanation]	During logging, the free disk space of the storage destination disk is less than 500MB.
[Operator response]	Increase disk space of the disk.
F9205S	The Boot-ROM file was not found. Please check the specified folder. "Boot ROM file name"
[Explanation]	The Boot ROM file of the MCU was not found.
[Operator response]	Check the existence of Boot ROM file in the folder specified by [Boot ROM] category in [Debug] tab of the setting dialog box for the project.
F9401S	Invalid emulation pod or MCU cable (unmatch or no- connected).
[Explanation]	The emulation pod or the MCU cable is incorrect. Alternatively, the MCU cable is not connected correctly.
[Operator response]	Turn off the emulator, then check the emulation pod and MCU cable. If the cable is not connected correctly, connect it correctly, then restart the SOFTUNE debugger.

F9402S	Invalid emulator hardware monitor program (unmatch).
[Explanation]	There are differences in the following monitor program. - The monitor program loaded into the emulator - The monitor program in the SOFTUNE install directory
[Operator response]	Load the monitor attached to this product using the monitor loader program. For details, refer to the "SOFTUNE Workbench Installation Manual (release note)". When using the emulator debugger (MB2100-01), enable the automatic loading of the monitor program with the setup wizard.
F9403S	Emulator hardware error. ["detailed information"]
[Explanation]	The emulator hardware cannot operate normally. Or, processing is stopped because an exception in detailed information is detected.
[Operator response]	- "RAM checker over-flow" Log data cannot be obtained at the specified interval due to the effect of other applications, etc., and so logging by the RAM Checker is stopped. During logging, do not perform operation that burdens the machine. - When the detailed information is empty, or ERRID is displayed, Check whether MCU operates normally. Reset and restart the emulator. If the problem occurs frequently, the emulator hardware, MCU, or target system may be down.
F9404S	Invalid emulator hardware monitor program version (old).
[Explanation]	Invalid install commands or parameters are found in the install file.
[Operator response]	Use the monitor loader program to load the monitor attached to this product. For details, refer to the "SOFTUNE Workbench Installation Manual (release note)". "Monitor program auto-loading" is enabled by the setup wizard in emulator debugger(MB2100-01).
F9405S	A bus error occurs. To issue reset is necessary for restore.
[Explanation]	When the emulator accessed the memory, the bus error occurred.
[Operator response]	Because the bus error occurred, the emulator cannot control MPU. Please issue reset to restore.
F9406S	Invalid either chip classification in project or chip on board.
[Explanation]	The CPU on board is different from the kind of CPU in the project file.
[Operator response]	Check the CPU on board or the kind of CPU in the project file.
F9407S	Cannot recognize a communication device and terminate debugger.
[Explanation]	Cannot recognize a communication device connecting with the debugger. And terminate the debugger.
[Operator response]	Check the connection between PC and emulator.

F9408S	Don't match target for emulator environment.
[Explanation]	The emulator settings specified in the setup file are different from the emulator environment in connection.
[Operator response]	Start the setup wizard to change the emulator type in the setup file or correct the emulator environment.
F9409S	Emulator hardware error. Please turn off the power immediately and check the connection.
[Explanation]	The emulator hardware cannot operate normally.
[Operator response]	Turn off the power immediately and check whether the MCU is operating normally. If this error occurs frequently, the emulator hardware, MCU, and/or target system may be faulty.
F9410S	A configuration board is not connected.
[Explanation]	A configuration board is not connected properly.
[Operator response]	Turn the emulator off, check the configuration board, make sure it is connected properly, and then restart the SOFTUNE debugger.
F9411S	Cannot continue debugging due to a CR trimming value error.
[Explanation]	The preset CR trimming value is abnormal.
[Operator response]	Please restart the debug system or change the target device.
F9412S	Invalid supply voltage.
[Explanation]	The supply voltage supplied from the user system is found abnormal.
[Operator response]	Review the supply voltage of the user system.
F9413S	MB2100-01 is not connected.
[Explanation]	Emulator debugger (MB2100-01) is not connected, or emulator debugger (MB2100-01) is not turned on.
[Operator response]	Connect emulator debugger (MB2100-01), or turn on emulator debugger (MB2100-01).
F9414S	DLL version is different. "file name"
[Explanation]	DLL file version of the "file name" is invalid.
[Operator response]	Install the latest SOFTUNE Workbench.
F9415S	The FPGA configuration is incompleteness.
[Explanation]	The FPGA configuration is not performed properly.
[Operator response]	Turn off the unit, and then turn on again. If the similar message appears again, set the automatic loading of the monitor program and restart again. If this message still appears, contact our sales representative.

F9416S	The connection cannot be established to the target MCU.
[Explanation]	Debugging could not start because the connection with the target could not be established.
[Operator response]	Check the setting items below for starting. · Reference frequency of communication speed (setup wizard) · DEBUG I/F cable length (setup wizard) · Chip type (project setting)
F9417S	The USB driver is not found.
[Explanation]	Failed to load the USB driver.
[Operator response]	Check whether the USB driver is properly installed. If this error occurs again after checking, reinstall the SOFTUNE Workbench.
F9418S	The user program execution was canceled, because chip reset was detected.
[Explanation]	Because a chip reset was detected during a break, the program was not executed.
[Operator response]	Perform the user system reset process, if necessary. Then, execute the user program again.
F9601S	Invalid communication status (or cable connection).
[Explanation]	The state of communication line is abnormal, or the cable is not connected correctly.
[Operator response]	Check the state of communication line. Then terminate debugging, turn the emulator off, and turn the emulator on again. When the USB cable is pulled out during debugging, this message appears.
F9602S	Invalid communication device name (or not specified).
[Explanation]	The specified communication device name is incorrect.
[Operator response]	Check the communication device name in install file.
F9603S	Invalid INTERFACE (not specified in install file).
[Explanation]	Invalid INTERFACE (not specified in install file).
[Operator response]	Check the install file.
F9604S	Cannot initialize "WINSOCK.DLL".
[Explanation]	"WINSOCK.DLL" cannot be initialized.
[Operator response]	Check if the TCP/IP protocol is installed on your Windows PC. If it is not installed, refer to Windows Help to install it.
F9901S	Memory allocation error.
[Explanation]	Debugging cannot be continued because memory of the host PC is insufficient.
[Operator response]	Immediately terminate SOFTUNE Workbench and increase the host PC empty memory and then restart SOFTUNE Workbench.

F9902S	System error.
[Explanation]	This program could not startup normally because of system error.
[Operator response]	Restart the system and then restart SOFTUNE Workbench.
F9903S	A necessary DLL file was not found.
[Explanation]	The required DLL file cannot be loaded.
[Operator response]	Re-install SOFTUNE Workbench.
F9904S	The version of CPU information file is an old version.
[Explanation]	The version of the CPU information file is old, so information cannot be set properly.
[Operator response]	Update the CPU information file to a new version.
F9905S	A necessary file for the USB communication is not found.
[Explanation]	A necessary dll-file for the USB communication is not found.
[Operator response]	Re-install a SOFTUNE Workbench.
F9906S	Failed in connection because BGM adaptor is too old.
[Explanation]	The BGM adapter is too old a version to be connected to the MCU board.
[Operator response]	Check the versions of the BGM adapter and MCU board, and connect the acceptable combination.
F9907S	Password is not correct. Please input the correct password by setup wizard.
[Explanation]	The password required for starting the debugger is wrong.
[Operator response]	Use Setup Wizard to enter the correct password for starting the debugger.
W1001S	Invalid data value (underflow).
[Explanation]	Data underflowed the specified precision.
[Operator response]	Recheck the precision or data.
W1002S	Invalid data value (overflow).
[Explanation]	Data overflowed the specified precision.
[Operator response]	Recheck the precision or data.
W1101S	Invalid symbol (multiple).
[Explanation]	Duplicate symbols are found.
[Operator response]	Recheck the source file corresponding to the load module.
W1102S	Invalid code section or entry data (not found in load module).
[Explanation]	The code section and input data are not in the loaded load module. The program counter (PC) is not set.
[Operator response]	Set the program counter (PC) and then execute the program.

W1103S	Command history buffer allocation error (in host memory).
[Explanation]	Buffer memory for the command history cannot be allocated to an internal memory area in the host PC.
[Operator response]	Expand the internal memory area in the host PC. If the SOFTUNE debugger is used as is, the command history function cannot be used.
W1104S	Invalid address (mis-alignment).
[Explanation]	In the FR family MCU, 16-bit data must be aligned on a 16-bit boundary and 32-bit data on a 32-bit boundary, respectively.
[Operator response]	Review the specified address.
W1201S	Invalid HELP command file (not found).
[Explanation]	The HELP command file is not placed in a correct location.
[Operator response]	Place the HELP command file in a correct location.
W1202S	Loaded different series's file.
[Explanation]	The load module file made with the tool of a series different from the chip specification of the installation file was loaded.
[Operator response]	Do the reload after the file is confirmed when the specified load module file is not a file of the purpose.
W1203S	Invalid file format.
[Explanation]	The format of the file to be loaded is illegal.
[Operator response]	Check the file.
W1204S	Loaded different series's file. (FPU instructions are included)
[Explanation]	The load module file made with the tool of a series different from the chip specification of the installation file was loaded (FPU instructions are included). In the load module of a different MCU series, the instruction might not operate correctly.
[Operator response]	Do the reload after the file is confirmed when the specified load module file is not a file of the purpose.
W1401S	Invalid timer (overflow).
[Explanation]	The execution-time timer overflowed during program execution.
[Operator response]	Shorten the measurement time.
W1402S	Invalid performance measuring data (buffer full).
[Explanation]	The buffer that stores performance measuring data became buffer full during program execution. Performance is not subsequently measured.
[Operator response]	Reduce the measurement count.
W1403S	Invalid pass count (overflow).
[Explanation]	The pass count overflowed.
[Operator response]	Check the term in the expression, then re-enter the command.

W1404S	User reset.
--------	-------------

[Explanation] An user reset is specified in MCU during command execution.

[Operator response] Select the GO command in [Debug]-[Run]-[Go] menu.

W1405S	The CR trimming value was adjusted due to a CR trimming value error. To continue debugging, adjust the CR trimming value.
--------	---

[Explanation] The CR trimming value was temporarily adjusted because the preset CR trimming value was abnormal. To continue debugging without encountering the same problem, the CR trimming value of the system has to be adjusted again.

[Operator response] The CR trimming value has to be adjusted again. Regarding the adjusting method, please refer to the support web page or contact our sales representative. After the preset CR trimming value has been adjusted, the debug can be continued; however, Cypress cannot guarantee the proper operation of the same device as a mass production device.

W1406S	Command error(Now MCU stopping).
--------	----------------------------------

[Explanation] The command that cannot be executed during MCU stopping has been issued.

[Operator response] Check the command.

W1407S	Do not access to debug resource.
--------	----------------------------------

[Explanation] When DMAC accesses the area (H'10000..H'103FF) of debugging during the break in the user program, read and write operations cannot be normally processed.

[Operator response] Prohibit the DMA transfer at the break when there is a possibility that DMAC accesses the area of the debug resource.

W1408S	Enabled the software break. The software break is written as an instruction code in the memory.
--------	---

[Explanation] If the software break is set in the FLASH area, the content in the memory of the setting address is temporarily rewritten.

[Operator response] None

W1601S	Failed to switch to the high-speed communication of DEBUG I/F.
--------	--

[Explanation] Because a high-speed communication can not be done, the setting of the high-speed communication was changed to the invalidity and switched to the normal communication mode.

[Operator response] Confirm the operating frequency under the setting and make the high-speed communication effective again.

W1901S	The setup file is read-only. The change in setup information cannot be preserved.
--------	---

[Explanation] The setup file is read-only. Changes to the setup information cannot be saved.

[Operator response] Remove the read-only attribute from the attributes for the setup file corresponding to the setup file name.

W1902S	Invalid CPU information data.
[Explanation]	Data in the CPU information file is invalid.
[Operator response]	Obtain the latest CPU information file.
W1903S	There is a possibility with an old version of DLL.
[Explanation]	The version of the program does not match that of the DLL file.
[Operator response]	Install the latest SOFTUNE Workbench.
W1904S	Start "Setup Wizard" to update the setup file.
[Explanation]	It is necessary to update information in the setup file, because that information was modified.
[Operator response]	Update the setup file on the setup wizard.
W1906S	Information file is not the latest. "Information file name"
[Explanation]	Information file is not the latest.
[Operator response]	Install the latest SOFTUNE Workbench.
E4001S	Command error.
[Explanation]	The command or line assembler syntax is incorrect.
[Operator response]	Check the syntax and parameters, then re-enter the command.
E4002S	Command qualifier error.
[Explanation]	The specified command qualifier is incorrect or it does not exist in the command.
[Operator response]	Check the command qualifier, then re-enter the command.
E4003S	Syntax error.
[Explanation]	An error is found in the command or line assembler syntax.
[Operator response]	Check the syntax and parameters and then re-enter.
E4004S	Invalid parameter count (over limit).
[Explanation]	The parameter count is too large.
[Operator response]	Check the command syntax and then re-enter.
E4005S	Invalid parameter omission.
[Explanation]	A required parameter is omitted.
[Operator response]	Check the command syntax and then re-enter the parameter.
E4006S	Parameter error.
[Explanation]	Illegal parameters are specified. The parameter name is illegal or parameters cannot be recognized as numeric values.
[Operator response]	Check the command syntax or input radix and then re-enter.
E4007S	Invalid operand.

[Explanation] There are invalid operands in the expression. Attempts were made to perform arithmetic operations using floating-point numbers. Arithmetic operations using floating-point numbers cannot be performed.

[Operator response] Check the operands in the statement and then re-enter.

E4008S	Invalid operator.
--------	-------------------

[Explanation] There are invalid operators in the expression.

[Operator response] Check the operators in the expression and then re-enter.

E4009S	Syntax error (operand not found).
--------	-----------------------------------

[Explanation] The operand is not found in the polynomial operator in the expression.

[Operator response] Check the expression and then input the operand correctly.

E4010S	Syntax error (' ' ' or ' ' ' not found).
--------	---

[Explanation] "or ' ' " on the right side of ' " or ' ' " is not found in the expression.

[Operator response] Check the expression and then input quotation marks correctly.

E4011S	Invalid nest level (over limit).
--------	----------------------------------

[Explanation] The nest level of (), *, and [] in the expression exceeds 16. Or, the nest level of the structure or union exceeds 16.

[Operator response] Check the expression.

E4012S	Syntax error (dividing by zero).
--------	----------------------------------

[Explanation] Division by 0 is found in the expression.

[Operator response] Check the operand in the expression and then re-enter.

E4013S	Invalid address specifying.
--------	-----------------------------

[Explanation] The ending address may be less than the starting address or the specified address range may extend over two or more areas.

[Operator response] Check the addresses, then re-enter the command.

E4014S	Invalid bit pattern (over 0x01 to 0xff).
--------	--

[Explanation] The value of the specified bit pattern is other than 0x01 to 0xff.

[Operator response] Check the bit pattern and then re-enter.

E4015S	Invalid bit offset (over 0 to 31).
--------	------------------------------------

[Explanation] The specified bit offset is not 0 to 31.

[Operator response] Check the bit offset, then re-enter the command.

E4016S	Invalid register or flag name (not found).
--------	--

[Explanation] The specified register or flag name is not found.

[Operator response] Check the register or flag name and then re-enter.

E4017S	Invalid symbol (not found).
--------	-----------------------------

[Explanation] The specified symbol is not found in the symbol table. Or, the specified symbol is a local variable and the symbol path name is not entered in the current scope.

[Operator response] Check whether the invalid symbol name is specified or whether the symbol data in the module to which the symbol belongs is entered in the symbol table, and then re-enter. If the symbol data in the module to which the symbol belongs is entered in the symbol table, specify the data with the symbol path name assigned, or enter the symbol path name in the current scope.

E4018S	Invalid command alias (not found).
--------	------------------------------------

[Explanation] The specified command alias does not exist.

[Operator response] Check the command alias, then re-enter the command.

E4019S	Invalid line number (not found).
--------	----------------------------------

[Explanation] The specified line number is not found in the source file. Or, the load module file (line number data) corresponding to the source file is not loaded.

[Operator response] Check the source file and then re-enter. Or, load the load module file corresponding to the source file.

E4020S	Invalid starting display line number (over ending line number).
--------	---

[Explanation] The source line start line number is larger than the display end line number.

[Operator response] Check the line number and then re-enter.

E4021S	Invalid cycle count (0).
--------	--------------------------

[Explanation] 0 was specified as the cycle count.

[Operator response] Check the cycle count, then re-enter the command.

E4022S	Invalid break point number (not found).
--------	---

[Explanation] The specified break point number is not found.

[Operator response] Check the break point number.

E4023S	Invalid data break point number (not found).
--------	--

[Explanation] The specified data break point number is not found.

[Operator response] Check the data break point number.

E4024S	Invalid watch point number (not found).
--------	---

[Explanation] The specified watch point number is not found.

[Operator response] Check the watch point number.

E4025S	Invalid starting display trace number (over ending number).
[Explanation]	The starting display trace number is larger than the display ending trace number.
[Operator response]	Check the trace number, then re-enter.
E4026S	Invalid format statement characters.
[Explanation]	The specified format statement character string is incorrect.
[Operator response]	Check the format statement character string, then re-enter the command.
E4027S	Invalid symbol (not found) path name.
[Explanation]	The specified symbol path name is not found.
[Operator response]	Check the symbol path name and then re-enter.
E4028S	Invalid function (not found, or argument error).
[Explanation]	The specified function is not found. Or, the invalid argument of the function is specified.
[Operator response]	Check the function or argument and then re-enter.
E4029S	Invalid expression (used variable of structure or union type).
[Explanation]	The structure or union variable cannot be used as the operand in the language expression.
[Operator response]	Recheck the data format. Prefix the operator & to the variable.
E4030S	Invalid address (not found).
[Explanation]	The corresponding address is not found in the line number.
[Operator response]	Recheck the line number.
E4031S	Invalid automatic variable reference.
[Explanation]	Attempts are made to reference the automatic variable out of the function in which the variable is set.
[Operator response]	The automatic variable can be referenced only within the function in which the variable is set.
E4032S	Invalid variable specifying.
[Explanation]	The specified variable is not the member of the structure or union variable.
[Operator response]	Check the structure or union member.
E4033S	Floating point data format error.
[Explanation]	The floating-point data format is illegal.
[Operator response]	Recheck the floating-point data format.

E4034S	Invalid macro command definition (not found).
[Explanation]	The specified macro command name is not found.
[Operator response]	Check the macro command name, then re-enter the command.

E4035S	Invalid symbol address (not found).
[Explanation]	The address of a register variable or address of a variable assigned to a register is referenced. Example: SET DATABREAK &r When the register with variable exists on the CPU address space, the address can be referred.
[Operator response]	The address of a register variable or address of a variable assigned to a register cannot be referenced.

E4101S	Invalid command list nest level (over 8).
[Explanation]	The nesting level of command list of the command procedure, command macro, or break point exceeds 8.
[Operator response]	Review the execution of the command.

E4102S	Symbol definition error.
[Explanation]	The free area allocated in host PC memory is insufficient to execute commands. This error occurs when too many device drivers are incorporated under the MS-DOS (PC) environment.
[Operator response]	Expand the free area allocated in host PC memory, then restart SOFTUNE Workbench.

E4103S	OS command error.
[Explanation]	An OS command cannot be executed. The command shell format is incorrect.
[Operator response]	Start the command shell of correct format.

E4104S	Invalid command shell (not found).
[Explanation]	The command shell could not be found.
[Operator response]	Review the environment variable, etc., so that the command shell can be started.

E4105S	Invalid alias string.
[Explanation]	The command alias includes an unregistrable character.
[Operator response]	Review command alias registration, then re-enter the command.

E4106S	Invalid macro command name (registered already).
[Explanation]	The same macro command is already registered.
[Operator response]	Review the macro command name, then re-enter the command.

E4107S	Invalid memory map definition.
[Explanation]	Memory mapping is too complex to define the area. When setting the memory area attributes, the areas with different attributes are excessive, causing the internal table to overflow.
[Operator response]	Simplify the memory mapping.
E4108S	Memory allocation error.
[Explanation]	There is insufficient memory space for command execution by the host PC. This error occurs when many software applications are started.
[Operator response]	Increase the memory space in the host PC to restart SOFTUNE Workbench.
E4109S	Object loading error.
[Explanation]	The object load destination exceeds the maximum value of address.
[Operator response]	Check the object size and object load destination or the specified address.
E4110S	Log file open error (already).
[Explanation]	The log file is already open.
[Operator response]	Close the current log file, then open a new log file.
E4111S	Memory access error.
[Explanation]	Attempts were made to access undefined memory. The address where access causing an error is made is displayed in the address part.
[Operator response]	Check the memory mapping.
E4112S	Invalid nest level of structure or union (over 16).
[Explanation]	The debug data table could not be created in the host PC memory. The maximum value of the nest level is 16.
[Operator response]	Increase the memory space in the host PC memory, then restart SOFTUNE Workbench.
E4113S	Debug data table creation error.
[Explanation]	The debug data table cannot be created in memory of the host PC or in the directory specified in TMP in the install file.
[Operator response]	Increase the memory space in the host PC and restart SOFTUNE Workbench. Or, check the condition of the directory specified in TMP in the install file.
E4114S	Logging control command error.
[Explanation]	The log file was operated although it is not open.
[Operator response]	Check that the log file is open.
E4115S	Invalid alias name (registered already).
[Explanation]	The same command alias is already registered.
[Operator response]	Review the command alias, then re-enter the command.
E4116S	Invalid alias name (not found).

[Explanation] The specified command alias does not exist.
 [Operator response] Check the command alias, then re-enter the command.

E4117S	Data type error.
--------	------------------

[Explanation] The data type is unmatched.
 [Operator response] Check the data type and then re-enter.

E4118S	Invalid member name (not specified).
--------	--------------------------------------

[Explanation] The structure or union name cannot be specified.
 [Operator response] Specify the structure or union name together with the member name.

E4119S	Break point and data break point setting error.
--------	---

[Explanation] Breakpoints and data breakpoints cannot be set. When the data monitoring condition cannot set in the emulator debugger (MB2198 or MB2147-01), display the message.
 [Operator response] Check the contents of break points and maximum count of break points.
 Check the contents of breakpoints and maximum count of breakpoints. In case of the emulator debugger (MB2198 or MB2147-01), the breakpoint, the trace trigger and the data watch break may be used same hardware point. Check the status in event lists.
 The emulator pod (MB2145-506, MB2145-507) or the emulator debugger (MB2147-05), the specified breakpoint be set at address out of debug area and breakpoints exceeds maximum count. Check the debug area.

E4120S	CALL command error.
--------	---------------------

[Explanation] The CALL command is already executing; it cannot be nested.
 [Operator response] Suspend the CALL command with a CLEAR CALL command. Alternatively, execute the GO or STEP command until the call operation terminates, then execute the CALL command.

E4121S	Invalid function (at the top).
--------	--------------------------------

[Explanation] There is no higher-level function than this function or this function is called from a program other than a C program.
 [Operator response] Check the current function.

E4122S	Invalid function (at the bottom).
--------	-----------------------------------

[Explanation] There is no lower-level function than this function or this function is called from a program other than a C program.
 [Operator response] Check the current function.

E4123S	Invalid coverage map (over-full).
--------	-----------------------------------

[Explanation] The coverage area cannot be set any more.
 [Operator response] Simplify coverage area specification.

E4124S	Coverage area setting error.
--------	------------------------------

[Explanation] The coverage area is not set.
 [Operator response] Set the coverage area.

E4125S	Invalid coverage area.
[Explanation]	An area outside the coverage area was specified.
[Operator response]	Check and specify the coverage area.
E4126S	Invalid coverage file.
[Explanation]	A file other than the coverage file was specified.
[Operator response]	Check file data.
E4127S	Invalid debug data (not loaded).
[Explanation]	The debug data file has not been loaded.
[Operator response]	Load the debug data file, then specify a coverage.
E4128S	Mapping overlap.
[Explanation]	The specified map area overlaps another area.
[Operator response]	Check map specification, then re-enter the command.
E4129S	Invalid address (mis-alignment).
[Explanation]	In the FR family MCU, 16-bit data must be aligned on a 16-bit boundary and 32-bit data on a 32-bit boundary, respectively.
[Operator response]	Review the specified address.
E4130S	Cannot open current source window.
[Explanation]	The source window that displays the current location could not be found in the set source search directory.
[Operator response]	Set the directory containing the source file.
E4131S	Cannot be used in current mode of debugger.
[Explanation]	The current debugger type cannot be used. The functions that can be used depend on the type of the debugger.
[Operator response]	Check the type of debugger.
E4132S	Command error (debug function violation).
[Explanation]	The function that can be used varies with the debug function.
[Operator response]	Check the setting of debug function by selecting [Setup] - [Debug environment] - [Select Debug function] menu.

E4133S	Task debugging cannot be used.
[Explanation]	The task debug function cannot be used if an object with that function is not loaded.
[Operator response]	Load the object with the task debug function.
E4134S	MMU data access exception.
[Explanation]	An MMU access error occurs.
[Operator response]	Check whether invalid memory was accessed.
E4135S	Register exception.
[Explanation]	An undefined register is accessed.
[Operator response]	Turn on the target again if recovery fails. Check whether a valid register was accessed.
E4136S	Not the first address of a parallel instruction.
[Explanation]	The specified address is not the first an address of the instruction executed in parallel.
[Operator response]	Please specify the first address of the instruction executed in parallel.
E4137S	Register access error.
[Explanation]	Cannot access the register.
[Operator response]	There is an inaccessible register according to the kind or the state of the chip. Please refer to the hardware manual of model to be used.
E4138S	Invalid combination of size and address beyond 64 bits boundary.
[Explanation]	Cannot specify the address beyond 64 bits boundary.
[Operator response]	Please specify the address matched 64 bits boundary.
E4139S	Over point.
[Explanation]	The set point count is above maximum. This message appears when the trace trigger or the break condition is set with emulator debugger (MB2198 or MB2147-01).
[Operator response]	Please check current set point count and state, then delete the unnecessary points and set the point again.
E4140S	Invalid point number.
[Explanation]	Cannot find the specified point number. If the invalid point number is specified (e.g. deleting event point), this message appears.
[Operator response]	Please check the point number.

E4141S	The same address is set.
[Explanation]	It is duplicate that the function is set at one address. For example, the same address is specified on each sequencer level.
[Operator response]	Please check current set point count and state, then delete the unnecessary points and set the point again.
E4142S	Invalid break condition.
[Explanation]	The data monitoring condition is not set. This message appears when the condition cannot be set for reasons of short on the event resource with emulator debugger (MB2198).
[Operator response]	Please check current set point count and state, then delete the unnecessary points and set the point again.
E4143S	No supported function with this device. (MBXXXX)
[Explanation]	This message appears when the evaluation chip or the emulator doesn't have the specified function.
[Operator response]	This message appears when the evaluation chip/OCDE or the emulator doesn't have the specified function. Check the hardware manual of the product you are using or the "SOFTUNE Workbench Command Reference Manual" based on MB number/OCD information in the error message.
E4144S	Please use the RAM checker function with a USB device.
[Explanation]	The RAM Checker cannot be used because other than USB (RS or LAN) is used as the communication device.
[Operator response]	When using the RAM Checker, use USB as the communication device.
E4201S	File access error.
[Explanation]	The file cannot be accessed.
[Operator response]	Check the condition of the disk in the host.
E4202S	File close error.
[Explanation]	The file cannot be closed.
[Operator response]	Check the condition of the disk in the host PC.
E4203S	File open error.
[Explanation]	The file cannot be opened.
[Operator response]	Check the file name or the condition of the disk in the host PC. Or, check the file and directory.
E4204S	Data write error.
[Explanation]	Data cannot be written to the file.
[Operator response]	Check the condition of the disk in the host PC.

E4205S	Invalid line number (not found).
[Explanation]	The corresponding source line is not found at the specified address. Even if the corresponding source line is not found, the source line is displayed in the source window.
[Operator response]	Please review the address. Or, load the load module with debug information.
E4206S	Alias file load error.
[Explanation]	The specified alias file cannot be loaded.
[Operator response]	Check the alias file name or the disk state of the host PC. Alternatively, check the directory containing the alias file.
E4207S	Alias file save error.
[Explanation]	The specified alias file cannot be saved.
[Operator response]	Check the condition of the disk in the host PC.
E4208S	Invalid file format.
[Explanation]	The format of the file to be loaded is illegal.
[Operator response]	Check the file.
E4209S	Open file read error.
[Explanation]	An error occurred during reading of the opened file.
[Operator response]	Check the file (drive) being read.
E4301S	Invalid interrupt factor number.
[Explanation]	The specified interrupt factor number does not exist.
[Operator response]	Check the interrupt factor number and enter it once again (IRQ0 to IRQ47).
E4302S	Invalid I/O buffer number.
[Explanation]	The specified I/O buffer number does not exist.
[Operator response]	The simulator debugger provides 0 to 3 I/O buffers. Refer to the Hardware Manual for the product type you are using to specify the correct I/O buffer number.
E4303S	Invalid port address.
[Explanation]	An address was specified beyond the port address range.
[Operator response]	A port address can be specified only in the MCU I/O area. Specify an address in the MCU I/O area.
E4304S	Invalid output destination.
[Explanation]	A data output destination, which is already in use as the data output destination, was specified.
[Operator response]	Specify a data output destination not in use.

E4305S	Invalid port count.
[Explanation]	The count of specified ports exceeds 4096.
[Operator response]	Specify 4096 ports or less.
E4306S	Simulation memory allocation error.
[Explanation]	Simulation memory cannot be allocated to an internal memory area in the host PC.
[Operator response]	Expand the internal memory area in the host PC.
E4307S	Invalid input data file.
[Explanation]	The file name assigned to the input port is incorrect or the file does not exist.
[Operator response]	Check the general format of the file.
E4401S	Verify error.
[Explanation]	A verify error occurred when data was being written to memory by a command.
[Operator response]	Check that data was written to the I/O area where values change and that memory is mounted. Also check whether or not a memory error occurred.
E4402S	Parity error (at emulation memory).
[Explanation]	A parity error occurred at accessing to the emulation memory.
[Operator response]	Reset the emulator body, then restart this. If the error occurs frequently, it may be an emulation memory malfunction.
E4403S	Parity error (at debug memory).
[Explanation]	A parity error occurred at accessing to the memory for emulator operation.
[Operator response]	Reset the emulator body, then restart this. If the error occurs frequently, it may be a malfunction of the memory for emulator operation.
E4404S	Command error (MCU is busy).
[Explanation]	An inexecutable command was tried to execute during MCU execution.
[Operator response]	Check the command.
E4408S	Invalid event number (not found).
[Explanation]	The specified event number is not found.
[Operator response]	Check the event number.
E4409S	Invalid level number (not found).
[Explanation]	The specified level number is not found.
[Operator response]	Check the level number.
E4410S	Command error (event mode violation).
[Explanation]	A command was specified that violates the event mode.
[Operator response]	Check the event mode setting with the [Setup]-[Debug Environment]-[Setup debug environment] menu.

E4411S	Invalid latch number (not found).
[Explanation]	The specified latch number is not found.
[Operator response]	Check the latch number.

E4412S	Invalid supply voltage.
[Explanation]	The supply voltage supplied from the user system is found abnormal.
[Operator response]	Review the supply voltage of the user system.

E4413S	MCU is in STOP mode, or invalid system clock.
[Explanation]	The system clock supplied from the user system is found abnormal.
[Operator response]	Review the system clock of the user system.

E4414S	MCU reset error.
[Explanation]	The MCU reset cannot be executed normally.
[Operator response]	The mode data and the reset vector read at reset may be an incorrect value. Set a correct value and retry this command. When this error occurs if the mode data is read from the user memory, the user memory cannot be read. Therefore, map it in the emulation memory before executing the [Debug]-[Reset of MCU] menu.

E4415S	Invalid MCU.
[Explanation]	Commands cannot be executed because MCU is not an operational state.
[Operator response]	<ol style="list-style-type: none"> 1.Set the reset vector and the mode data, then execute the [Debug]-[Reset of MCU] menu. 2.Release the SLEEP, STOP or HOLD state on the user system side, or set the reset vector and the mode data, then execute the [Debug]-[Reset of MCU] menu. Note that the HOLD state cannot be released by the [Debug]-[Reset of MCU] menu. 3.Check the execution result of the command.

E4416S	Invalid jump level number.
[Explanation]	The jump destination level number of the sequencer is incorrect.
[Operator response]	Review the jump destination level number. The sequencer cannot jump to the same level as the level to be specified.

E4417S	Command error (on internal ROM real-time mode).
[Explanation]	The command cannot be executed because the MCU execution mode is native.
[Operator response]	Change the MCU execution mode to debug.

E4418S	The user program execution was canceled, because chip reset was detected.
[Explanation]	This command cannot be executed because user reset is specified. This error occurs even if user reset is already released.
[Operator response]	Release user reset, execute the [Debug]-[Reset of MCU] menu, then execute this command.

E4419S	Abort command error.
[Explanation]	The ABORT command cannot be executed due to the SLEEP or STOP state.
[Operator response]	Release the SLEEP or STOP state.
E4420S	Command error (hardware standby).
[Explanation]	This command cannot be executed due to the hardware standby state. This error occurs even if the hardware standby state is already released.
[Operator response]	Release the hardware standby state, execute the [Debug]-[Reset of MCU] menu, then execute this command.
E4421S	Command error (timer-mode violation).
[Explanation]	When the timer mode is "timer", the SHOW CYCLE command and CLEAR CYCLE command cannot be executed. When the timer mode is "cycle", the SHOW TIMER command and CLEAR TIMER command cannot be executed.
[Operator response]	Check the timer mode, then re-enter the command.
E4422S	Invalid break point (not found).
[Explanation]	The software break point became invalid because data in the address where the software break point is set was rewritten by program execution. Alternatively, the software break point remained in memory because an error occurred when the point was being reburied. In this case, data in the program being loaded and setting data at the software break point are not guaranteed.
[Operator response]	Delete all software breaks, then review the program data. If some software breaks still remain in memory, reload the program.
E4423S	Monitor hit stack-check function.
[Explanation]	A stack-check exception occurred within the monitor at returning to the user-program.
[Operator response]	Invalidate a stack-check function or increase usable stack area.
E4424S	Exception occurred while accessing user resource. ["detailed information"]
[Explanation]	Because the exception shown in detailed information had been detected while accessing a specified user resource, processing was interrupted.
[Operator response]	Please confirm the corresponding matter from the exception generation factor described in the hardware manual of the product used based on detailed information. In FR-V, the address and trap type (TT) at the exception occurrence are displayed as detailed information in the form of "address TT:nn". Please refer to "4.7.2.3 Debug Environment" in "SOFTUNE Workbench Operation Manual" for the factor which relates to MMU because the operation by the debugger might be necessary.

E4425S	Invalid area number.
[Explanation]	The specified area number is not found.
[Operator response]	Specify an area number which is possible to set.
E4426S	Command error (event mode violation<Performance>).
[Explanation]	Event mode is set to performance mode, and so the command cannot be executed.
[Operator response]	Change event mode to a mode where the command can be used.
E4427S	Command error (event mode violation<Single trace>).
[Explanation]	Event mode is set to single trace mode, and so the command cannot be executed.
[Operator response]	Change event mode to a mode where the command can be used.
E4428S	Command error (event mode violation<Multi trace>).
[Explanation]	Event mode is set to multi trace mode, and so the command cannot be executed.
[Operator response]	Change event mode to a mode where the command can be used.
E4429S	Command error(The trace function is operating).
[Explanation]	The command that cannot be executed during the trace function has been issued.
[Operator response]	Check the command.
E4430S	If Access size is word, this event cannot be set. Recommend 'Don't care'.
[Explanation]	Even though Access size is word, the odd-numbered address is set.
[Operator response]	Specify the access size to "size free".
E4431S	Data on the RAM is broken.
[Explanation]	<p>The recovery of RAM area that the debugger has used temporarily failed.</p> <p>The debugger uses a part of internal RAM under the following conditions.</p> <ul style="list-style-type: none"> •When the execution starts •When the break is executed •When the reset is executed •When the debug completes
[Operator response]	Set the RAM contents again if necessary.
E4432S	Command error(pass count mode violation).
[Explanation]	The command that can not be used in a current pass count mode was input.
[Operator response]	Switch the pass count mode.
E4433S	Command error(time measurement mode violation).
[Explanation]	The command that can not be used in a current execution time mode was input.
[Operator response]	Switch the execution time mode.

E4434S	Detected debug resource access violation (by user operation [program, DMA]).
[Explanation]	The access to the debugging resource was detected during the break in the user program. The access to the debugging resource is not correctly processed during the break in the user program.
[Operator response]	When there is a possibility that the DMA transfer is done to the debug resource during the break in the user program, [DMA operation at the break] in [Emulation] tab of the debug environment setting dialog box is set to the prohibition and do not generate the DMA transfer.
E4435S	Command error(The software break is not allowed).
[Explanation]	The software break tried to be set though the setting of the software break was prohibited.
[Operator response]	Switch the software break to the use permission.
E4436S	Command error(The performance function is operating).
[Explanation]	The command that could not be used while the performance measuring tried to be executed.
[Operator response]	Confirm the command.
E4437S	The peripheral resources of Sleep mode was accessed.
[Explanation]	The software break tried to be set though the setting of the software break was prohibited.
[Operator response]	Do not access to the peripheral resources in sleep mode.
E4438S	Failed to switch to the high-speed communication of DEBUG I/F.
[Explanation]	The current reference clock frequency for the high-speed communication is outside the range of the clock that can be set.
[Operator response]	Confirm whether to find the mistake in the set reference clock for the high-speed communication.
E4439S	Since connection with the emulator is lost, the emulator is reconnected. Please initialize the user system.
[Explanation]	The USB connection with the emulator was re-connected.
[Operator response]	<p>In the following cases, the user system may also be affected.</p> <p>In that case, debugging may not be continued normally.</p> <p>Reset the user system or restart the debugger.</p> <ul style="list-style-type: none"> • When the emulator is disconnected by noise • When a user system is reset during power loss of the emulator

E4501S	Verify error.
[Explanation]	A verify error occurred when data was being written to memory by a command.
[Operator response]	Check that data was written to the I/O area where values change and that memory is mounted. Also check whether or not a memory error occurred.
E4502S	Illegal stack area.
[Explanation]	The stack area used by the monitor debugger cannot be accessed.
[Operator response]	Secure the correct stack area.
E4503S	System call error (cannot execute).
[Explanation]	In this state, a system call cannot be executed normally.
[Operator response]	Execute a system call in the state in which system calls can be issued. Interrupts may be disabled.
E4504S	This command is not built-in.
[Explanation]	The associated function is not built in a target side.
[Operator response]	Built the associated function in the target-side program.
E4505S	FLASH area can be changed only by LOAD command.
[Explanation]	Only the LOAD command can be used to access the Flash area.
[Operator response]	To update the Flash area, create an update file and load it using the LOAD command.
E4506S	The file of mixed ROM/RAM areas can not be loaded.
[Explanation]	The same file cannot be loaded into the ROM and RAM areas.
[Operator response]	The file to be loaded into the Flash area and the one to be loaded into the RAM area must be created separately.
E4601S	Invalid communication status (or cable connection).
[Explanation]	The communication line state is abnormal or the cable connection is incorrect.
[Operator response]	Check the line connection state. Push the reset button on the target board when CPU is running in the sub-clock mode.
E4602S	Communication: Parallel adapter not connected.
[Explanation]	The parallel adapter is not connected.
[Operator response]	Connect the parallel adapter correctly, then re-execute this program.
E4603S	Communication: Mismatch parallel adapter version.
[Explanation]	Communication cannot be performed because the version of the parallel adapter is old.
[Operator response]	Use the latest parallel adapter.
E4604S	Communication: Cannot find host name.

[Explanation] The specified host name is not registered in the hosts file.

[Operator response] Please register the host name in the hosts file.
For details, refer to the "Appendix C Setting LAN Interface" in "SOFTUNE Workbench Operation Manual".

E4605S	Communication: Cannot find port number.
--------	---

[Explanation] The port number of emulator is not defined in the services file.

[Operator response] Please register the port number in the services file.
For details, refer to the "Appendix C Setting LAN Interface" in "SOFTUNE Workbench Operation Manual".

E4606S	Communication: Cannot open device.
--------	------------------------------------

[Explanation] Abnormality is found in the specified device or not connected correctly.

[Operator response] Please confirm whether the specified device is correctly connected.

E4607S	Communication: Time out.
--------	--------------------------

[Explanation] Reception information on transmission information was not received within the fixed time.

[Operator response] Please confirm whether the specified device is correctly connected.

E4608S	Communication : DEBUG I/F error.
--------	----------------------------------

[Explanation] The problem occurred by the communication in DEBUG I/F.

[Operator response] Confirm the connection of the DEBUG I/F cable.

E4609S	Communication : USB error.
--------	----------------------------

[Explanation] The problem occurred by the USB communication.

[Operator response] Confirm the connection of the USB cable.

E4610S	Communication : Time out(DEBUG I/F).
--------	--------------------------------------

[Explanation] It is not possible to communicate with the target. The possibility that there is a problem in the target is high.

[Operator response] Reactivate the target.

E4611S	Communication : Time out(debug program).
--------	--

[Explanation] It is not possible to communicate with the target. The possibility that there is a problem in the target is high.

[Operator response] Reactivate the target.

E4901S	Not enough timer resource.
--------	----------------------------

[Explanation] The timer resource of Windows cannot be used.

[Operator response] End other applications, then re-execute this command.

E4902S	The key code cannot be defined.
--------	---------------------------------

[Explanation] The key code cannot be defined.

[Operator response] Define another key code.

E4903S	Write error(at flash memory).
--------	-------------------------------

[Explanation] The time limit was exceeded while writing to the flash memory.

[Operator response] Consult the Sales Department immediately when this error appears.

E4904S	Erase error(at flash memory).
--------	-------------------------------

[Explanation] The time limit was exceeded while erasing data from the flash memory.

[Operator response] Consult the Sales Department immediately when this error appears.

E4905S	Don't use because enabled MCU security.
--------	---

[Explanation] When the security of MCU was effective, the function that can not be used was operated.

[Operator response] Release the security of MCU.

APPENDIX C EXECUTION SUSPENSION MESSAGES LIST

This appendix describes the Execution Suspension Message List.

■ Execution Suspension Messages List

Break at address by breakpoint

[Explanation] This message is displayed when a break is caused by a software breakpoint.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by hardware breakpoint

[Explanation] This message is displayed when a break is caused by a hardware breakpoint (including breakpoint specified by GO command).
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by code event break (No. code-event-number)

[Explanation] This message is displayed when a break is caused by a code event.
Address indicates the address of the next instruction to be executed where execution was suspended.
Code-event-number indicates the number of the code event that caused the break.

Break at address by code event break (sequential)

[Explanation] This message is displayed when a sequential break is caused by code event 1 or 2.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by data event break (No. data-event-number)

[Explanation] This message is displayed when a break is caused by a data event.
Address indicates the address of the next instruction to be executed where execution was suspended.
Data-event-number indicates the number of the data event that caused the break.

Break at address by data event break (sequential)

[Explanation] This message is displayed when a sequential break is caused by data event 1 or 2.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by trace buffer full

[Explanation]	This message is displayed when a break is caused by a trace buffer full. Address indicates the address of the next instruction to be executed where execution was suspended.
---------------	---

Break at address by alignment error break (code)
--

[Explanation]	This message is displayed when a break is caused by a code fetch alignment error. Address indicates the address of the next instruction to be executed where execution was suspended.
---------------	--

Break at address by alignment error break (data)
--

[Explanation]	This message is displayed when a break is caused by a data access alignment error. Address indicates the address of the next instruction to be executed where execution was suspended.
---------------	---

Break at address by external trigger break
--

[Explanation]	This message is displayed when a break is caused by the input of an external signal to the TRIG pin of the emulator. Address indicates the address of the next instruction to be executed where execution was suspended.
---------------	---

Break at address by trace lost break

[Explanation]	This message is displayed when a break is caused by the trace data loss. Address indicates the address of the next instruction to be executed where execution was suspended.
---------------	---

Break at address by data break at access-address
--

[Explanation]	This message is displayed when a break is caused by a data breakpoint. Address indicates the address of the next instruction to be executed where execution was suspended. Access-address indicates the address where the access that caused the break was made.
---------------	--

Break at address by guarded access access-type at access-address
--

[Explanation]	This message is displayed when a break is caused by code fetch access to a code fetch inhibited area, read access to a read-inhibited area, or write access to a write-inhibited area. There may be an error in the memory attribute or the program. Address indicates the address of the next instruction to be executed where execution was suspended. Access-type indicates the type of the access that caused the break. Access-address indicates the address where the access that caused the break was made.
---------------	--

Break at address by dispatch task from task ID=<dispatch-source-task-ID> to task ID=<dispatch-destination-task-ID>

[Explanation] This message is displayed when a break is caused by task dispatch. Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by system call <system-call-name> on task ID=<task-ID>

[Explanation] This message is displayed when a break is caused by a system call. Address indicates the address of the next instruction to be executed where execution was suspended. System-call-name indicates the name of the system call that caused the break. Task-ID indicates the ID of the task that issued the system call.

Break at address by command abort request

[Explanation] This message is displayed when a break is caused by the [Debug]-[Abort] menu. Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by output file overflow

[Explanation] This message is displayed when a break occurs because data could not be written to the data output file of an output port. Check the data output file of the output port. Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by stop abnormal action

[Explanation] This message is displayed when a break occurs because a non-executable instruction was added after a prefix instruction. Check the program because it may be incorrect. Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by invalid call termination

[Explanation] The CALL command is executed after a breakpoint is set in the address indicated by the current PC and the RP register is set so that control will return to the address. For this reason, a break occurs if the address of the original PC is executed during execution of the CALL command. In this way, this message is displayed when a break occurs before execution of the CALL command is completed. Restart execution of the CALL command with the GO command as is or suspend execution with the CLEAR CALL command. Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by EIT (attached information)
--

[Explanation]	<p>This message is displayed when a break is caused by EIT.</p> <p>Address indicates the address of the next instruction to be executed where execution was suspended.</p>
---------------	--

Break at address by step command

[Explanation]	<p>This message is displayed by the SHOW STATUS command when a break is caused by step (INTO) execution.</p> <p>Address indicates the address of the next instruction to be executed where execution was suspended.</p>
---------------	---

Break at address by call command

[Explanation]	<p>This message is displayed when a break occurs after execution of the CALL command is completed.</p> <p>Address indicates the address of the next instruction to be executed where execution was suspended.</p>
---------------	---

Break at address by violation to combine instructions

[Explanation]	<p>Displayed when a combined instruction that is not allowed in 1 parallel instruction is executed and there is a break in the simulator debugger.</p> <p>The address is the one where the execution stopped (the instruction for the next execution).</p>
---------------	--

Break at address by slot issue violation
--

[Explanation]	<p>Displays when an instruction that cannot be issued to a slot in 1 parallel instruction is executed and there is a break in the simulator debugger.</p> <p>The address is the one where the execution stopped (the instruction for the next execution).</p>
---------------	---

Break at address by resource write-write confliction
--

[Explanation]	<p>Displayed when executing a plurality of write access of the same memory or register of 1 parallel instruction in the simulator debugger.</p> <p>The address is the one of the parallel instruction executed after the instruction that had the cause of the break.</p>
---------------	---

Break at address by datawatchbreak

[Explanation]	<p>Displayed when a break is caused by a data watch breakpoint.</p> <p>Address indicates the address of the break factor instruction.</p>
---------------	---

Break at address by unknown break factor
--

[Explanation]	<p>Displayed when a break is caused by indefinite factor.</p> <p>Address indicates the address of the next instruction to be executed where execution was suspended.</p>
---------------	--

Break at address by trace end break

[Explanation] Displayed when the break is caused by the break on the completion of trace.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by breakpoint (data watch)

[Explanation] Displayed when the break is caused by the data watch break.
Address indicates the address where the command has caused the break.

Break at address by sequential break

[Explanation] This message is displayed when a break is caused by a data event. Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by data event break

[Explanation] Displayed when the break is caused by the sequencer.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by sequential or pass count break

[Explanation] Displayed when the break is caused by sequence or hardware/count break.
Address indicates the address of the next instruction to be executed where execution was suspended.

Break at address by guarded access

[Explanation] This message is displayed when a break is caused by code fetch access to a code fetch inhibited area, read access to a read-inhibited area, or write access to a write-inhibited area. There may be an error in the memory attribute or the program. Address indicates the address of the next instruction to be executed where execution was suspended.

Note:

If the CPU pause state is released during execution of the user program, that information is also displayed as additional information.

Example: When stopped due to the forced break:

Break at [address] by command abort request (exit CPU pause)

It should be noted that there are four types of CPU pause state:

CPU sleep/Bus sleep/Timer mode/Stop mode

APPENDIX D MAJOR CHANGES

Page	Section	Change Results
Revision 8.1		
-	-	Company name and layout design change
Revision 9.0		
356	APPENDIX B ERROR MESSAGE FOR DEBUGGERS	Added the error message (E4439S)
Revision 10.0		
-	-	Company name and layout design change
289	11.6 PRINTF	Added the explanation of the floating-point number
343	APPENDIX A Manager-Related Messages	Added the error message (E4804S)

Index

**The index follows on the next page.
This is listed in alphabetic order.**

Index

Symbols

%B	298
%BIT	298
%CALL	291
%D	298
%DEBUGFUNCTION	312
%ENTRY	293
%ERRNUM.....	292
%EVAL	307
%EVENTNUM	308
%GET_CORESTATE	313
%GET_EVENTSTATE.....	315
%GET_PERFORMANCESTATE.....	316
%GET_TRACESTATE	314
%H	298
%LOADNUM.....	297
%PFMEVENTNUM.....	311
%RADIX.....	295
%S	298
%SCPADR.....	296
%STKTOP	294
%STRCAT	303
%STRCMP	301
%STRGET	299
%STRLEN	302
%STRSTR	300
%SYMLEN.....	304
%TOSTR	306
%TOVAL.....	305
%TRIGGERNUM.....	310
%W	298

A

ABORT	83
ASSEMBLE	242

B

BATCH	260
BREAK	287

C

CALL	91
CANCEL ALIAS.....	266
CANCEL BREAK.....	109
CANCEL CODEEVENT	134
CANCEL COVERAGE	178
CANCEL DATABREAK	117
CANCEL DATAEVENT	140
CANCEL DATATRACEAREA	200
CANCEL EVENT	126
CANCEL FREQUENCY	61
CANCEL INPORT	37
CANCEL INTERRUPT	44
CANCEL LOGGING	275
CANCEL MAP (type 1).....	29
CANCEL MAP (type 2).....	30
CANCEL OUTPORT	41
CANCEL PERFORMANCE.....	172
CANCEL RAMCHECK	215
CANCEL SEQUENCE.....	148
CANCEL TRACETRIGGER.....	207
CANCEL VARIABLE	269
CANCEL WATCH.....	57
CLEAR CALL	93
CLEAR COVERAGE	179
CLEAR PERFORMANCE	173
CLEAR TIMER	183
CLEAR TRACE	190
COMPARE	236
COPY VECTOR	45

D

DISABLE ALIGNMENTBREAK	154
DISABLE BREAK	111
DISABLE CLOCKSUPERVISOR	74
DISABLE CODEEVENT	136

Index

DISABLE DATABREAK	119
DISABLE DATAEVENT	142
DISABLE DMA	71
DISABLE EVENT	129
DISABLE EXTERNALMEMORY	53
DISABLE LOGGING	277
DISABLE MAP	28
DISABLE POWERONDEBUG	65
DISABLE RAMCHECK	219
DISABLE SEQUENCE	150
DISABLE TRACE (type 1)	193
DISABLE TRACE (type 2)	194
DISABLE TRACETRIGGER	209
DISABLE VERIFYMODE	32
DISABLE WATCHDOG	49
DISASSEMBLE	243
DOWN	258
DUMP	239

E

ENABLE ALIGNMENTBREAK	153
ENABLE BREAK	110
ENABLE CLOCKSUPERVISOR	73
ENABLE CODEEVENT	135
ENABLE DATABREAK	118
ENABLE DATAEVENT	141
ENABLE DMA	70
ENABLE EVENT	128
ENABLE EXTERNALMEMORY	52
ENABLE LOGGING	276
ENABLE MAP	27
ENABLE POWERONDEBUG	64
ENABLE RAMCHECK	217
ENABLE SEQUENCE	149
ENABLE TRACE (type 1)	191
ENABLE TRACE (type 2)	192
ENABLE TRACETRIGGER	208
ENABLE VERIFYMODE	31
ENABLE WATCHDOG	48
ENTER	226
ERASE FLASH	69
Error Message for Debuggers	334
EXAMINE	224
Execution Suspension Messages List	360
EXIT	5

F

FILL	237
------------	-----

G

GO	82
----------	----

I

IF	284
INITIALIZE	4

L

LIST	252
LOAD	246

M

Manager-Related Messages	318
MOVE	238

P

PRINTF	278
--------------	-----

Q

QUIT	261
------------	-----

R

REPEAT	285
RESET	6

S

SAVE	249
SEARCH MEMORY	232
SEARCH TRACE	195
SET ABORT	84
SET ALIAS	264
SET BREAK (type 1)	99
SET BREAK (type 2)	102
SET BREAK (type 3)	103
SET BREAK (type 4)	106
SET BREAKCONDITION (type 1)	156
SET BREAKCONDITION (type 2)	158
SET CODEEVENT	130
SET COVERAGE	174
SET DATABREAK (type 1)	112
SET DATABREAK (type 2)	114
SET DATAEVENT	137
SET DATATRACEAREA	197
SET DELAY	210
SET EVENT	120
SET EXTERNALMEMORY	51
SET FREQUENCY	58

SET INPORT	34	SHOW MAP (type 2)	26
SET INTERRUPT	42	SHOW MEMORY	230
SET LOGGING	272	SHOW MODE	16
SET MAP (type 1).....	23	SHOW OUTPORT.....	40
SET MAP (type 2).....	24	SHOW OUTPUT	281
SET MEMORY.....	228	SHOW PATH.....	255
SET MODE	12	SHOW PERFORMANCE	167
SET OUTPORT	38	SHOW POWERONDEBUG	66
SET OUTPUT	280	SHOW RADIX	18
SET PATH	254	SHOW RAMCHECK.....	214
SET PERFORMANCE.....	163, 165, 166	SHOW REALTIMEMEMORYAREA	63
SET RADIX.....	17	SHOW REGISTER.....	235
SET RAMCHECK	212	SHOW RESET	8
SET REGISTER	234	SHOW RUNMODE.....	11
SET RESET	7	SHOW SCOPE.....	256
SET RUNMODE	9	SHOW SEQUENCE	147
SET SEQUENCE (type 1)	143	SHOW SOURCE	20
SET SEQUENCE (type 2)	145	SHOW STATUS	94
SET SOURCE	19	SHOW STEP.....	90
SET STEP	88	SHOW SYSTEM	21
SET TRACE (type 1)	184	SHOW TIMER	181
SET TRACE (type 2)	185	SHOW TRACE (type 1).....	187
SET TRACETRIGGER	203	SHOW TRACE (type 2).....	189
SET TRIGGER	151	SHOW TRACERIGGER.....	206
SET VARIABLE	267	SHOW TRIGGER.....	152
SET VECTOR.....	46	SHOW VARIABLE.....	268
SET WATCH.....	55	SHOW VECTOR	47
SHOW ABORT	85	SHOW VERIFYMODE	33
SHOW ALIAS	265	SHOW WATCHDOG.....	50
SHOW ALIGNMENTBREAK	155	STEP.....	86
SHOW BREAK	107	SYNCHRONIZE FLASH.....	67
SHOW BREAKCONDITION	160		
SHOW CALLS	180	U	
SHOW CLOCKSUPERVISOR.....	75	UP	257
SHOW CODEEVENT	132		
SHOW COVERAGE	175	W	
SHOW DATABREAK.....	116	WHILE	286
SHOW DATAEVENT	139		
SHOW DATATRACEAREA.....	199		
SHOW DELAY	211		
SHOW DETAILTRACE.....	201		
SHOW DMA.....	72		
SHOW EVENT.....	124		
SHOW EXTERNALMEMORY	54		
SHOW FREQUENCY	60		
SHOW INPORT	36		
SHOW INTERRUPT	43		
SHOW LOGGING.....	274		
SHOW MAP (type 1).....	25		

Revision History



Revision History

Document Title: FR Family SOFTUNE™ Workbench Command Reference Manual for V6				
Document Number: 002-04576				
Revision	ECN#	Issue Date	Origin of Change	Description of Change
**	-	12/09/2014	MITK	Migrated to Cypress and assigned document number 002-04576. No change to document contents or format.
*A	5194300	29/03/2016	RYOK	Company name and layout design change Added the explanation of the floating-point number Added the error message (E4804S)