

FR FAMILY SOFTUNE™ WORKBENCH COMMAND REFERENCE MANUAL

for V6

FR FAMILY
SOFTUNE™ WORKBENCH
COMMAND REFERENCE
MANUAL
for V6

FUJITSU MICROELECTRONICS LIMITED

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.

■ Trademarks

SOFTUNE is a trademark of FUJITSU MICROELECTRONICS LIMITED.

REALOS (REALtime Operating System) is a trademark of FUJITSU MICROELECTRONICS LIMITED.

Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation in the U.S. and 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 one 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.

- The contents of this document are subject to change without notice.
Customers are advised to consult with sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of FUJITSU MICROELECTRONICS device; FUJITSU MICROELECTRONICS does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. FUJITSU MICROELECTRONICS assumes no liability for any damages whatsoever arising out of the use of the information.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of FUJITSU MICROELECTRONICS or any third party or does FUJITSU MICROELECTRONICS warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU MICROELECTRONICS assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).
Please note that FUJITSU MICROELECTRONICS will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.
- Exportation/release of any products described in this document may require necessary procedures in accordance with the regulations of the Foreign Exchange and Foreign Trade Control Law of Japan and/or US export control laws.
- The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

Command Reference Notation Format

The command reference notation format is given below.

Command name (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.

☉: 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.

Example

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

CONTENTS

CHAPTER 1	Environment Setup Commands	1
1.1	INITIALIZE	3
1.2	EXIT	4
1.3	RESET	5
1.4	SET RUNMODE	6
1.5	SHOW RUNMODE	8
1.6	SET MODE	9
1.7	SHOW MODE	11
1.8	SET RADIX	12
1.9	SHOW RADIX	13
1.10	SET SOURCE	14
1.11	SHOW SOURCE	15
1.12	SHOW SYSTEM	16
1.13	SET MAP (type 1)	17
1.14	SET MAP (type 2)	18
1.15	SHOW MAP (type 1)	19
1.16	SHOW MAP (type 2)	20
1.17	ENABLE MAP	21
1.18	DISABLE MAP	22
1.19	CANCEL MAP (type 1)	23
1.20	CANCEL MAP (type 2)	24
1.21	ENABLE VERIFYMODE	25
1.22	DISABLE VERIFYMODE	26
1.23	SHOW VERIFYMODE	27
1.24	SET INPORT	28
1.25	SHOW INPORT	30
1.26	CANCEL INPORT	31
1.27	SET OUTPORT	32
1.28	SHOW OUTPORT	34
1.29	CANCEL OUTPORT	35
1.30	SET INTERRUPT	36
1.31	SHOW INTERRUPT	37
1.32	CANCEL INTERRUPT	38
1.33	COPY VECTOR	39
1.34	SET VECTOR	40
1.35	SHOW VECTOR	41
1.36	ENABLE WATCHDOG	42
1.37	DISABLE WATCHDOG	43
1.38	SHOW WATCHDOG	44
1.39	SET EXTERNALMEMORY	45
1.40	ENABLE EXTERNALMEMORY	46
1.41	DISABLE EXTERNALMEMORY	47
1.42	SHOW EXTERNALMEMORY	48
1.43	SET WATCH	49

1.44	CANCEL WATCH	51
1.45	SET FREQUENCY	52
1.46	SHOW FREQUENCY	53
1.47	SET REALTIMEMEMORYAREA	54
1.48	SHOW REALTIMEMEMORYAREA	55
CHAPTER 2 Program Execution Commands		57
2.1	GO	58
2.2	STEP	59
2.3	SET STEP	60
2.4	SHOW STEP	61
2.5	CALL	62
2.6	CLEAR CALL	64
2.7	SHOW STATUS	65
CHAPTER 3 Break/Event Control Command		67
3.1	SET BREAK (type 1)	69
3.2	SET BREAK (type 2)	71
3.3	SET BREAK (type 3)	72
3.4	SHOW BREAK	74
3.5	CANCEL BREAK	76
3.6	ENABLE BREAK	77
3.7	DISABLE BREAK	78
3.8	SET DATABREAK (type 1)	79
3.9	SET DATABREAK (type 2)	81
3.10	SHOW DATABREAK	83
3.11	CANCEL DATABREAK	84
3.12	ENABLE DATABREAK	85
3.13	DISABLE DATABREAK	86
3.14	SET EVENT	87
3.15	SHOW EVENT	90
3.16	CANCEL EVENT	91
3.17	ENABLE EVENT	92
3.18	DISABLE EVENT	93
3.19	SET CODEEVENT	94
3.20	SHOW CODEEVENT	96
3.21	CANCEL CODEEVENT	97
3.22	ENABLE CODEEVENT	98
3.23	DISABLE CODEEVENT	99
3.24	SET DATAEVENT	100
3.25	SHOW DATAEVENT	102
3.26	CANCEL DATAEVENT	104
3.27	ENABLE DATAEVENT	105
3.28	DISABLE DATAEVENT	106
3.29	SET SEQUENCE (type 1)	107
3.30	SET SEQUENCE (type 2)	109
3.31	SHOW SEQUENCE	110

3.32	CANCEL SEQUENCE	111
3.33	ENABLE SEQUENCE	112
3.34	DISABLE SEQUENCE	113
3.35	SET TRIGGER	114
3.36	SHOW TRIGGER	115
3.37	ENABLE ALIGNMENTBREAK	116
3.38	DISABLE ALIGNMENTBREAK	117
3.39	SHOW ALIGNMENTBREAK	118
3.40	SET BREAKCONDITION (type 1)	119
3.41	SET BREAKCONDITION (type 2)	121
3.42	SHOW BREAKCONDITION	123
CHAPTER 4 Program Execution Analysis Commands		125
4.1	SET PERFORMANCE	127
4.2	SHOW PERFORMANCE	128
4.3	CLEAR PERFORMANCE	130
4.4	SET COVERAGE	131
4.5	SHOW COVERAGE	132
4.6	CANCEL COVERAGE	135
4.7	CLEAR COVERAGE	136
4.8	SHOW CALLS	137
4.9	SHOW TIMER	138
4.10	CLEAR TIMER	140
4.11	SET TRACE (type 1)	141
4.12	SET TRACE (type 2)	142
4.13	SHOW TRACE (type 1)	144
4.14	SHOW TRACE (type 2)	146
4.15	CLEAR TRACE	147
4.16	ENABLE TRACE	148
4.17	DISABLE TRACE	149
4.18	SEARCH TRACE	150
4.19	SET DATATRACEAREA	152
4.20	SHOW DATATRACEAREA	154
4.21	CANCEL DATATRACEAREA	155
4.22	SHOW DETAILTRACE	156
4.23	SET TRACETRIGGER	158
4.24	SHOW TRACETRIGGER	160
4.25	CANCEL TRACETRIGGER	161
4.26	ENABLE TRACETRIGGER	162
4.27	DISABLE TRACETRIGGER	163
4.28	SET RAMCHECK	164
4.29	SHOW RAMCHECK	166
4.30	CANCEL RAMCHECK	167
4.31	ENABLE RAMCHECK	169
4.32	DISABLE RAMCHECK	170

CHAPTER 5	Memory/Register Operation Commands	171
5.1	EXAMINE	172
5.2	ENTER	174
5.3	SET MEMORY	175
5.4	SHOW MEMORY	177
5.5	SEARCH MEMORY	179
5.6	SET REGISTER	180
5.7	SHOW REGISTER	181
5.8	COMPARE	182
5.9	FILL	183
5.10	MOVE	184
5.11	DUMP	185
CHAPTER 6	Line Assemble and Disassemble Commands	187
6.1	ASSEMBLE	188
6.2	DISASSEMBLE	189
CHAPTER 7	Load and Save Commands	191
7.1	LOAD	192
7.2	SAVE	194
CHAPTER 8	Source File/Symbol Commands	197
8.1	LIST	198
8.2	SET PATH	200
8.3	SHOW PATH	201
8.4	SHOW SCOPE	202
8.5	UP	203
8.6	DOWN	204
CHAPTER 9	Command Procedure Commands	205
9.1	BATCH	206
9.2	QUIT	207
CHAPTER 10	Replacement Commands	209
10.1	SET ALIAS	210
10.2	SHOW ALIAS	211
10.3	CANCEL ALIAS	212
10.4	SET VARIABLE	213
10.5	SHOW VARIABLE	214
10.6	CANCEL VARIABLE	215

CHAPTER 11 Utility Commands	217
11.1 SET LOGGING	218
11.2 SHOW LOGGING	219
11.3 CANCEL LOGGING	220
11.4 ENABLE LOGGING	221
11.5 DISABLE LOGGING	222
11.6 PRINTF	223
11.7 SET OUTPUT	225
11.8 SHOW OUTPUT	226
CHAPTER 12 Control Commands	227
12.1 IF	228
12.2 REPEAT	229
12.3 WHILE	230
12.4 BREAK	231
CHAPTER 13 Built-in Variables and Functions	233
13.1 %CALL	234
13.2 %ERRNUM	235
13.3 %ENTRY	236
13.4 %STKTOP	237
13.5 %RADIX	238
13.6 %SCPADR	239
13.7 %LOADNUM	240
13.8 %BIT, %B, %H, %W, %S, %D	241
13.9 %STRGET	242
13.10 %STRSTR	243
13.11 %STRCMP	244
13.12 %STRLEN	245
13.13 %STRCAT	246
13.14 %SYMLen	247
13.15 %TOVAL	248
13.16 %TOSTR	249
13.17 %EVAL	250
13.18 %EVENTNUM	251
13.19 %TRIGGERNUM	252
13.20 %DEBUGFUNCTION	253
APPENDIX	255
APPENDIX A MANAGER-RELATED MESSAGES	256
APPENDIX B ERROR MESSAGE FOR DEBUGGERS	272
APPENDIX C EXECUTION SUSPENSION MESSAGES LIST	293
INDEX.....	297

CHAPTER 1

Environment Setup Commands

This chapter describes the Environment Setup commands.

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

- 1.22 DISABLE VERIFYMODE
- 1.23 SHOW VERIFYMODE
- 1.24 SET INPORT
- 1.25 SHOW INPORT
- 1.26 CANCEL INPORT
- 1.27 SET OUTPORT
- 1.28 SHOW OUTPORT
- 1.29 CANCEL OUTPORT
- 1.30 SET INTERRUPT
- 1.31 SHOW INTERRUPT
- 1.32 CANCEL INTERRUPT
- 1.33 COPY VECTOR
- 1.34 SET VECTOR
- 1.35 SHOW VECTOR
- 1.36 ENABLE WATCHDOG
- 1.37 DISABLE WATCHDOG
- 1.38 SHOW WATCHDOG
- 1.39 SET EXTERNALMEMORY
- 1.40 ENABLE EXTERNALMEMORY
- 1.41 DISABLE EXTERNALMEMORY
- 1.42 SHOW EXTERNALMEMORY
- 1.43 SET WATCH
- 1.44 CANCEL WATCH
- 1.45 SET FREQUENCY
- 1.46 SHOW FREQUENCY
- 1.47 SET REALTIMEMEMORYAREA
- 1.48 SHOW REALTIMEMEMORYAREA

1.1 INITIALIZE

The INITIALIZE command initializes the debugger.

■ INITIALIZE

- Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<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)	⊙
Monitor		⊙

- Format

EXIT

- Description

The EXIT command terminates the debugger.

- Example

>EXIT

1.3 RESET

The RESET command inputs the reset signal to the MCU.

■ RESET

- Debugger

Simulator		☉
Emulator	(MB2197)	☉
	(MB2198)	☉
Monitor		✕

- Format

RESET

- Description

The RESET command inputs the reset signal to the MCU.

- Example

>RESET

1.4 SET RUNMODE

The SET RUNMODE command sets an MCU operation mode.

■ SET RUNMODE

● Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

● Format

SET RUNMODE

Command qualifiers

Setting operation (run) 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.

There are four operation modes: full trace, real time, internal trace, and external trace.

These modes differ as follows:

- 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 DSU3 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 DSU3 chip integrated with the function.

- Example

```
>SET RUNMODE/TRACE
```

Note:

In this function, the command qualifiers that can be specified vary depending on the emulator or its connections.

For details, refer to SOFTUNE Workbench USER'S MANUAL "MCU Operation Mode".

1.5 SHOW RUNMODE

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

■ SHOW RUNMODE

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
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
```

1.6 SET MODE

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

■ SET MODE

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

[Format 1] SET MODE

[Format 2] SET MODE debug-function-number

• Parameter

- Format 2

debug-function-number

Specify the debug function to be enabled.

Using the SHOW MODE command allows you to check selectable debug function and its function number.

• Command qualifiers

- Format 1

Setting of event mode

/NORMAL (default at start-up)

Sets event mode to TRACE.

/PERFORMANCE

Sets event mode to PERFORMANCE.

- Format 2

Setting of debug function

/CONFIG

Sets debug function.

● Description

• 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 SEQUENCE, DATAWATCH BREAK, and TRACETRIGGER 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 used real-time memory 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

```
>SET MODE/PERFORMANCE
```

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

Note:

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

1.7 SHOW MODE

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

■ SHOW MODE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
Monitor		×

- Format

SHOW MODE

- Description

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

- Example

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

Note:

This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench COMMAND REFERENCE MANUAL "Measuring Performance" and "Debug mode".

1.8 SET RADIX

The SET RADIX command sets default base number.

■ SET RADIX

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
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
```

1.9 SHOW RADIX

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

■ SHOW RADIX

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		⊙

- Format

SHOW RADIX

- Description

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

- Example

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

1.10 SET SOURCE

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

■ SET SOURCE

● Debugger

Simulator		☉
Emulator	(MB2197)	☉
	(MB2198)	☉
Monitor		☉

● Format

SET SOURCE

Command qualifiers

/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 the added source line.

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

● Example

```
>SET SOURCE/DISPLAY
```

1.11 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)	⊙
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
```

1.12 SHOW SYSTEM

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

■ SHOW SYSTEM

- Debugger

Simulator		☉
Emulator	(MB2197)	☉
	(MB2198)	☉
Monitor		○

- Format

SHOW SYSTEM

- Description

The SHOW SYSTEM command displays system information.

- Example

```
>SHOW SYSTEM
FR Family Softune Workbench V60L07
Debugger type      = Emulator Debugger
MCU type           = MB91V240
VCpu dll name      = C:\Softune\Bin\Wv911e2.dll
DSU type           = DSU4
Common version     = V02L09
Monitor version    = V03L02R09
MCU frequency      = 2.000MHz
Communication device= USB
```

1.13 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)	×
Monitor		×

● Format

SET MAP {address | address-range}

Parameters

address (address formula)

Specify the memory address where access attribute to be set.

address-range (address formula)

Specify 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. (settable by increments of one byte)

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 00001000...00001FFF
```

1.14 SET MAP (type 2)

The SET MAP command sets an area which accessing prohibited.

■ SET MAP (type 2)

● Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		○

● Format

SET MAP/INACCESSIBLE {address | address-range}

Parameters

address (address formula)

Specify the memory address.

Address-range (address formula)

Specify the memory area.

Command qualifiers

/INACCESSIBLE

Set inaccessible area.

● Description

Sets an 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
```


1.15 SHOW MAP (type 1)

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

■ SHOW MAP (type 1)

● Debugger

Simulator		⊙
Emulator	(MB2197)	×
	(MB2198)	×
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      ..      FFFEFFFF      undefined
FFFF0000      ..      FFFFFFFF      read code
```

1.16 SHOW MAP (type 2)

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

■ SHOW MAP (type 2)

● Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		⊙

● Format

SHOW MAP/INACCESSIBLE

Command qualifiers

/INACCESSIBLE

Displays an inaccessible area.

● Description

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

● Example

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

1.17 ENABLE MAP

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

■ ENABLE MAP

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		○

- Format

ENABLE MAP/INACCESSIBLE {map number}

Parameter

map number

Specify a number of inaccessible area.

Command qualifiers

/INACCESSIBLE

Enables an inaccessible area.

- Description

Enables an inaccessible area of a specified number.

- Example

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

1.18 DISABLE MAP

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

■ DISABLE MAP

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		○

- Format

DISABLE MAP/INACCESSIBLE {map number}

Parameter

map number

Specify a number of inaccessible area.

Command qualifiers

/INACCESSIBLE

Disable an inaccessible area.

- Description

Disables an inaccessible area of a specified number.

- Example

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

1.19 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)	×
Monitor		×

- Format

CANCEL MAP { address | address-range }

Parameters

address (address formula)

Specify the address where undefined attribute to be assigned.

address-range (address formula)

Specify the address range where undefined attribute to be assigned.

Command qualifier

/ALL

Assigns undefined attribute to all set maps.

- Description

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

- Example

>CANCEL MAP/ALL

1.20 CANCEL MAP (type 2)

The CANCEL MAP command deletes an inaccessible area of a specified address-range.

■ CANCEL MAP (type 2)

● Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		○

● Format

CANCEL MAP {address | address-range}

Parameters

address (address formula)

Specify the memory address.

Address-range (address formula)

Specify the memory area.

Command qualifiers

/INACCESSIBLE

Deletes an inaccessible area.

/ALL

Deletes all inaccessible area.

● Description

The CANCEL MAP command deletes an inaccessible area of a specified address-range.

● Example

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

1.21 ENABLE VERIFYPAGE

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

■ ENABLE VERIFYPAGE

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

- Format

ENABLE VERIFYPAGE

- Description

The **ENABLE VERIFYPAGE** 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 VERIFYPAGE
```

1.22 DISABLE VERIFYMODE

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

■ DISABLE VERIFYMODE

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

- Format

DISABLE VERIFYMODE

- Description

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

The verify mode is disabled when the debugger is started.

- Example

```
>DISABLE VERIFYMODE
```


1.23 SHOW VERIFYMODE

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

■ SHOW VERIFYMODE

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		×

- Format

SHOW VERIFYMODE

- Description

The **SHOW VERIFYMODE** command displays the status of the verify mode (mode in which verify operation is enabled or displayed when memory is written by a command).

- Example

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

1.24 SET INPORT

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.

■ SET INPORT

● Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

● Format

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

Parameters

port-address

Specify the port address.

mask-data

Specify the mask data.

Only 1 mask data bit can be used as port data.

data-input-source

Specify 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.

Specify \$TERMINAL as input terminal.

\$TERMINAL cannot be specified as file name.

cycle-count (default: decimal number)

Specify the count of program instruction execution cycles (H'1 to H'FFFFFFFF).

Command qualifiers

Specifying access size

/BYTE (default when omitted)

Specifies port access when specified address accessed 1 byte.

/HALFWORD

Specifies port access when specified address accessed 2 bytes.

/WORD

Specifies port access when specified address accessed 4 bytes.

/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

1.25 SHOW INPORT

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

■ SHOW INPORT

- Debugger

Simulator		☉
Emulator	(MB2197)	×
	(MB2198)	×
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
```

1.26 CANCEL INPORT

The CANCEL INPORT command cancels simulation of specified port address.

■ CANCEL INPORT

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

- Format

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

Parameter

port-address

Specify the port address.

Command qualifier

/ALL

Cancels all data set by SET INPORT command.

- Description

The CANCEL INPORT command cancels simulation of specified port address.

- Example

```
>CANCEL INPORT/ALL
```

1.27 SET OUTPUT

The **SET OUTPUT** 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 OUTPUT

● Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

● Format

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

Parameters

port-address

Specify the port address.

Mask-data

Specify the mask data.

Only 1 mask data bit can be used as port data.

data-output-destination

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

Specify \$TERMINAL as the output terminal.

\$TERMINAL cannot be specified as the file name.

The same file name cannot be used.

Command qualifiers

Specifying access attribute

/BYTE (default when omitted)

Specifies port access when specified address accessed 1 byte.

/HALFWORD

Specifies port access when specified address accessed 2 bytes.

/WORD

Specifies port access when specified address accessed 4 bytes.

/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 a 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

1.28 SHOW OUTPUT

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

■ SHOW OUTPUT

- Debugger

Simulator		☉
Emulator	(MB2197)	×
	(MB2198)	×
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
```


1.29 CANCEL OUTPORT

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

■ CANCEL OUTPORT

- Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

- Format

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

Parameter

port-address

Specify the port address.

Command qualifier

/ALL

Cancels all setting by SET OUTPORT command.

- Description

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

- Example

>CANCEL OUTPORT/ALL

1.30 SET INTERRUPT

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.

■ SET INTERRUPT

● Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

● Format

SET INTERRUPT interrupt number, cycle count

Parameters

interrupt-number (default: decimal number)

Specify the interrupt vector number.

For usable interrupts, refer to the manual for the chip being used.

cycle-count (default: decimal number)

Specify the count of program instruction execution cycles (H'1 to H'FFFFFFFF).

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
```

1.31 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)	×
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
```

1.32 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)	×
Monitor		×

- Format

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

Parameter

interrupt-number (default: decimal number)

Specify 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
```

1.33 COPY VECTOR

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

■ COPY VECTOR

● Debugger

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

● Format

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

Parameters

vector-number (default: decimal number)

Specify the number of vector to be copied.

TBR-value

Specify the area address to which 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

Specify the area address to which vector table to be copied.

The COPY VECTOR command copies the value in the initial area (area relative to 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
```

1.34 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"/>
Monitor		<input type="radio"/>

● Format

SET VECTOR vector-number, address-value

Parameters

vector-number

Specify the number of vector to be set.

address-value

Specify the starting address 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
```

1.35 SHOW VECTOR

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

■ SHOW VECTOR

● Debugger

Simulator		☉
Emulator	(MB2197)	☉
	(MB2198)	☉
Monitor		○

● Format

SHOW VECTOR [vector-number-range]

Parameter

vector-number-range

Specify the range of vector numbers to be displayed.

Specify range 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

1.36 ENABLE WATCHDOG

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

■ ENABLE WATCHDOG

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

- Format

ENABLE WATCHDOG

- Description

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

Only the DSU3/DSU4 chip can specify this command.

- Example

```
>ENABLE WATCHDOG
```


1.37 DISABLE WATCHDOG

The **DISABLE WATCHDOG** command enables a watchdog timer.

■ DISABLE WATCHDOG

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

- Format

DISABLE WATCHDOG

- Description

The **DISABLE WATCHDOG** command enables a watchdog timer.

Only the DSU3/DSU4 chip can specify this command.

- Example

```
>DISABLE WATCHDOG
```

1.38 SHOW WATCHDOG

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

■ SHOW WATCHDOG

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		×

- Format

SHOW WATCHDOG

- Description

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

Only the DSU3/DSU4 chip can specify this command.

- Example

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

1.39 SET EXTERNALMEMORY

Specify the chip select number to set memory acting as external ROM or external RAM.

■ SET EXTERNAL MEMORY

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
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 the chip select function, refer to your LSI Specifications.

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
```

1.40 ENABLE EXTERNALMEMORY

Enable the external memory emulation function.

■ ENABLE EXTERNAL MEMORY

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

- Format

ENABLE EXTERNALMEMORY

- Description

This command is used to enable the disabled external memory emulation function.

- Example

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

1.41 DISABLE EXTERNALMEMORY

Disable the external memory emulation function.

■ DISABLE EXTERNAL MEMORY

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

- Format

DISABLE EXTERNALMEMORY

- Description

This command is used to disable the enabled external memory emulation function.

- Example

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

1.42 SHOW EXTERNALMEMORY

Display the setting state of the external memory emulation function.

■ SHOW EXTERNAL MEMORY

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
Monitor		×

● Format

SHOW EXTERNALMEMORY

● Description

This command is used to display the setting state of the external memory emulation function.

● Example

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

1.43 SET WATCH

Sets specified variables in the watch window.

■ SET WATCH

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		⊙

● Format

SET WATCH variable [, watch-window-number]

Parameters

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

Command qualifier

- Base Number
 - /BINARY
Specify that variable values will be displayed in binary.
 - /OCTAL
Specify that variable values will be displayed in octal.
 - /DECIMAL
Specify that variable values will be displayed in decimal.
 - /HEXADECIMAL
Specify 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
Specify that display will be provided with 1-byte length in assembler mode.
 - /HALFWORD
Specify that display will be provided with 2-byte length in assembler mode.
 - /WORD
Specify that display will be provided with 4-byte length in assembler mode.
 - /DWORD
Specify that display will be provided with 8-byte length in assembler mode.
 - /SINGLE
Specify that display will be provided with single-precision floating-point number in assembler mode.
 - /DOUBLE
Specify 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 will be valid only when the setting of/ASSEMBLER is specified.

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

● Example

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


1.44 CANCEL WATCH

Deletes specified variables from the watch window.

■ CANCEL WATCH

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
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 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
```

1.45 SET FREQUENCY

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

■ SET FREQUENCY

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

SET FREQUENCY /MAX Maximum operating frequency

Parameters

Maximum operating frequency (default: decimal number)

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

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

Command qualifier

/MAX (Cannot be omitted)

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

/RESET

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

● Description

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.

● Example

```
>SET FREQUENCY/MAX64
```

Note:

There is a possibility that the emulator malfunctions when a value that is lower than the operation frequency actually used is set.

1.46 SHOW FREQUENCY

The **SHOW FREQUENCY** command displays the maximum operating frequency set by the **SET FREQUENCY** command.

■ SHOW FREQUENCY

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
Monitor		×

- Format

SHOW FREQUENCY

Command qualifier

/MAX (Cannot be omitted)

The **SHOW FREQUENCY** command displays the maximum operating frequency for CPU.

- Description

The **SHOW FREQUENCY** command displays the maximum operating frequency for CPU set by the **SET FREQUENCY** command.

max frequency: Maximum operating frequency of current CPU

- Example

```
>SHOW FREQUENCY/MAX
```

```
max frequency:33MHz
```

1.47 SET REALTIMEMEMORYAREA

Sets a real time memory area.

■ SET REALTIMEMEMORYAREA

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

SET REALTIMEMEMORYAREA area number, area address

Parameters

- Area number (default: decimal number)
Specify the real time memory area number to be set (1, 2).
- Area address (default: decimal number)
Specify 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, it will be overwritten.
 - If a watch variable is set in the area that has been disabled as a real time memory area by overwriting, the following message will appear:
"There is a watch variable that has been disabled for real time monitoring."
-

1.48 SHOW REALTIMEMEMORYAREA

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

■ SHOW REALTIMEMEMORYAREA

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
Monitor		×

- Format

SHOW REALTIMEMEMORYAREA

- Description

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

If no such area is set, it will display "not found".

- Example

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


CHAPTER 2

Program Execution Commands

This chapter describes the Program Execution commands.

- 2.1 GO
- 2.2 STEP
- 2.3 SET STEP
- 2.4 SHOW STEP
- 2.5 CALL
- 2.6 CLEAR CALL
- 2.7 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"/>
Monitor		<input type="radio"/>

● Format

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

Parameters

starting-address (address formula)

Specify the address at which program execution started.

break-address (address formula)

Specify the address at which program execution stopped.

Command qualifiers

Return setting

/RETURN

Executes program from function currently being executed to parent function return location.

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

The optimized program may not be stopped normally.

● Description

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

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

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

If a command qualifiers is omitted, program execution will start as set by the SET GO command.

● Example

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


2.2 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"/>
Monitor		<input type="radio"/>

● Format

STEP [step count]

Parameter

step-count (default: decimal number)

Specify the count of times STEP command executed (H'1 to H'FFFFFFFF).

If step-count specifying is omitted, the count of times is 1.

Command qualifiers

/INSTRUCTION

Executes program in units of machine instructions.

/LINE

Executes program in units of source lines.

/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 units of source lines (/LINE).

-When the source window display mode is another display mode, the program is executed in units of machine instructions (/INSTRUCTION).

/INTO

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

/OVER

Executes function call and subroutine call instructions (i.e., CALL) 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.

● Description

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

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

● Example

```
>STEP
>STEP/INSTRUCTION
```

2.3 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"/>
Monitor		<input type="radio"/>

● Format

SET STEP

Command qualifiers

- Step unit specification

/INSTRUCTION

Executes program in units of machine instructions.

/LINE

Executes program in units of source lines.

/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 units of source lines (/LINE).

-When the source window display mode is another display mode, the program is executed in units of machine instructions (/INSTRUCTION).

- Step operation specification

/INTO (default at start-up)

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

/OVER

Executes function call and subroutine call instructions (i.e., CALL) 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.

● 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 AUTOMATIC, INTO.

● Example

```
>SET STEP/INSTRUCTION
```

2.4 SHOW STEP

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

■ SHOW STEP

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		⊙

- Format

SHOW STEP

- Description

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

- Example

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

2.5 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"/>
Monitor		<input type="radio"/>

● Format

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

Parameters

function-name

Specify the name of function to be called.

argument

Compiles with C/C++ arguments.

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

Command qualifiers

/DISPLAY (Default at start-up)

Sets made in which a return value displayed.

/NODISPLAY

Sets made in which a return value not displayed.

● Description

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

Evaluate the argument of the specified function in dummy argument format, and execute it.

If the count of specified actual arguments is more than the count of dummy arguments, evaluate as many counts of actual arguments as that exceeds the dummy arguments 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 for the current PC, and sets the return address so as to return to that point, calling 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

Break at address by Invalid call termination

Re-starts execution using the GO command to continue the execution of the CALL 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 are not restored.

● Example

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

2.6 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"/>
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
```

2.7 SHOW STATUS

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

■ SHOW STATUS

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		⊙

● Format

SHOW STATUS

● Description

The SHOW STATUS command displays the MCU execution status.

If the MCU breaks, the command displays the break factor of the immediately-preceding program execution.

● Example

```
>SHOW STATUS
MCU status : executing
>SHOW STATUS
break at 0000FF00 by breakpoint
```


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 SHOW BREAK
- 3.5 CANCEL BREAK
- 3.6 ENABLE BREAK
- 3.7 DISABLE BREAK
- 3.8 SET DATABREAK (type 1)
- 3.9 SET DATABREAK (type 2)
- 3.10 SHOW DATABREAK
- 3.11 CANCEL DATABREAK
- 3.12 ENABLE DATABREAK
- 3.13 DISABLE DATABREAK
- 3.14 SET EVENT
- 3.15 SHOW EVENT
- 3.16 CANCEL EVENT
- 3.17 ENABLE EVENT
- 3.18 DISABLE EVENT
- 3.19 SET CODEEVENT
- 3.20 SHOW CODEEVENT
- 3.21 CANCEL CODEEVENT

- 3.22 ENABLE CODEEVENT
- 3.23 DISABLE CODEEVENT
- 3.24 SET DATAEVENT
- 3.25 SHOW DATAEVENT
- 3.26 CANCEL DATAEVENT
- 3.27 ENABLE DATAEVENT
- 3.28 DISABLE DATAEVENT
- 3.29 SET SEQUENCE (type 1)
- 3.30 SET SEQUENCE (type 2)
- 3.31 SHOW SEQUENCE
- 3.32 CANCEL SEQUENCE
- 3.33 ENABLE SEQUENCE
- 3.34 DISABLE SEQUENCE
- 3.35 SET TRIGGER
- 3.36 SHOW TRIGGER
- 3.37 ENABLE ALIGNMENTBREAK
- 3.38 DISABLE ALIGNMENTBREAK
- 3.39 SHOW ALIGNMENTBREAK
- 3.40 SET BREAKCONDITION (type 1)
- 3.41 SET BREAKCONDITION (type 2)
- 3.42 SHOW BREAKCONDITION

3.1 SET BREAK (type 1)

The SET BREAK command sets a breakpoint at the specified break address.

■ SET BREAK (type 1)

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input checked="" 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

SET BREAK break-address [, pass-count] [, command: command...]

Parameters

break-address (address formula)

Specify the address at which breakpoint set.

pass-count (default: decimal number)

Specify the number of times breakpoint to be hit.

Simulator debugger: 1 to 65535, emulator debugger (FR60Lite or FR80S): 1 to 255.

Program execution specifying is stopped when this number of times is reached.

If pass-count is omitted, 1 is assumed.

Pass-count is valid only in the simulator debugger or emulator debugger (FR60Lite or FR80S); it is ignored in the emulator and monitor debuggers.

command

Specify 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. This function is ignored in the emulator debugger and the monitor debugger.

Command qualifiers

[Simulator]

/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/ Monitor]

/SOFT

Specifies software breakpoint.

/COUNT (FR60Lite or FR80S)

Specifies hardware/count breakpoint. This function can be used only when the FR60Lite or FR80S is used.

/HARD (default at start-up)

Specifies hardware breakpoint (valid only in emulator debugger).

● Description

The SET BREAK command sets a breakpoint at the specified break address.

The count of breakpoints to be specified is as follows:

Simulator	65535
Emulator	4096 (software), 5 (hardware)
Monitor	16

● Example

[Simulator]

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

[Emulator]

```
>SET BREAK 00ff0200
>SET BREAK 00ff0300, 3
```

Notes:

- 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 SOFTUNE Workbench Operation Manual "4.6.4 Breakpoints".
 - 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.
 - 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 command sets a data monitoring breakpoint (software) at the specified break address.

■ SET BREAK (type 2)

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

SET BREAK/BREAKCONDITION break-address [, pass-count]

Parameters

break-address (address formula)

Specify the address at which breakpoint set.

pass-count (default: decimal number)

Specify the number of times breakpoint to be hit (1 to 255).

Program execution specifying is stopped when this number of times is reached. If pass-count is omitted, 1 is assumed.

Function type command qualifiers

/BREAKCONDITION

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

Command qualifiers

/HARD (default at start-up)

Specifies hardware breakpoint

/COUNT (FR60Lite or FR80S)

Specification of pass-count is omitted.

This function can be used only when the Emulator debugger (FR60Lite or FR80S) is used.

● Description

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

■ SET BREAK (type3)

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

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

Parameters

- code-address (address formula)
Specify the address of code that serves as a data monitoring condition.
- data-address (address formula)
Specify the address of data that serves as a data monitoring condition.
- address-mask (data formula)
Specify 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.
- data (data formula)
Specify the data of the address 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)
Specify 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
Specify byte access as a data monitoring condition.
/HALFWORD
Specify halfword (2-byte) access as a data monitoring condition.
/WORD
Specify word (4-byte) access as a data monitoring condition.
/ANYTHING (Default when omitted)
A break is applied when the specified address is accessed, regardless of the access data length.

Specifying access attribute

/READ

Specify data read access as a data monitoring condition.

/WRITE

Specify data write access as a data monitoring condition.

When omitted, /READ/WRITE is specified.

● Description

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

This command is used to determine whether to set breakpoints by referring data in the data address when execution arrives at the specified code address.

Up to four points can be set. The count of points that can be used varies depending on the settings of SET EVENT and SET TRACETRIGGER.

● Example

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

Notes:

- This command cannot use except the FR60Lite. For details, refer to SOFTUNE Workbench USER'S MANUAL "Break".
- 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.

3.4 SHOW BREAK

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

■ **SHOW BREAK**

● Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		⊙

● Format

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

Parameter

breakpoint-number (default: decimal number)

Specify the breakpoint number.

Command qualifier

/ALL (default when omitted)

Displays all breakpoints.

/NORMAL (FR60Lite or FR80S)

Only the hardware or software break information is displayed.

/COUNT (FR60Lite or FR80S)

Only the hardware/count break information is displayed.

/DATAWATCH (FR60Lite)

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

● Description

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

● Example

[Simulator]

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

[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	

Note:

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.

3.5 CANCEL BREAK

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

■ CANCEL BREAK

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input checked="" 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)

Specify the breakpoint number.

Use the SHOW BREAK command to reference the set breakpoint numbers.

Command qualifier

/ALL

Cancels all breakpoints.

/NORMAL (FR60Lite or FR80S)

The hardware or software break is canceled.

/COUNT (FR60Lite or FR80S)

The hardware/count break is canceled.

/DATAWATCH (FR60Lite)

The data-monitoring break is canceled.

● Description

The CANCEL BREAK command cancels the specified breakpoint(s).

● Example

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

3.6 ENABLE BREAK

The **ENABLE BREAK** command enables the specified breakpoint(s).

■ ENABLE BREAK

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input checked="" 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)

Specify the breakpoint number.

Use the SHOW BREAK command to reference the set breakpoint numbers.

Command qualifier

/ALL

Enables all breakpoints.

/NORMAL (FR60Lite or FR80S)

The hardware or software break is enabled.

/COUNT (FR60Lite or FR80S)

The hardware/count break is enabled.

/DATAWATCH (FR60Lite)

The data-monitoring break is enabled.

● Description

The **ENABLE BREAK** command enables the specified breakpoint(s).

● Example

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

3.7 DISABLE BREAK

The **DISABLE BREAK** command disables the specified breakpoint(s).

■ DISABLE BREAK

● Debugger

Simulator		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input checked="" 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)

Specify the breakpoint number.

Use the **SHOW BREAK** command to reference the set breakpoint numbers.

Command qualifier

/ALL

Disables all breakpoints.

/NORMAL (FR60Lite or FR80S)

The hardware or software break is disabled.

/COUNT (FR60Lite or FR80S)

The hardware/count break is disabled.

/DATAWATCH (FR60Lite)

The data-monitoring break is disabled.

● Description

The **DISABLE BREAK** command disables the specified breakpoint(s).

● Example

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

3.8 SET DATABREAK (type 1)

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

■ SET DATABREAK (type 1)

● Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

● Format

SET DATABREAK data-access-address [, pass-count] [, command: command...]

Parameters

data-access-address (address formula)

Specify the address at which data access breakpoint set.

pass-count (default: decimal number)

Specify the number of times data access breakpoint hit (1 to 65535).

Program execution is stopped when this number of times is reached.

If pass-count is specifying omitted, 1 is assumed.

command

Specify 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. This function is ignored in the emulator debugger and the monitor debugger.

Command qualifiers

Specify of STUB function

/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.

Specify 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 irrespective of data read or write access.

When such a 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++ 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.
-

3.9 SET DATABREAK (type 2)

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

■ SET DATABREAK (type 2)

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊖*
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)
Specify 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
Specify 8 bits for the data width for causing a break condition.
/HALFWORD
Specify 16 bits for the data width for causing a break condition.
/WORD
Specify 32 bits for the data width for causing a break condition.
- Specifying function
/BREAKCONDITION (only MB2198 DSU4)
The conditions set in SET BREAKCONDITION are combined into break conditions.

- 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.

A maximum number which can be set up is 2 points.

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,3
```

Note:

This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Break".

3.10 SHOW DATABREAK

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

■ SHOW DATABREAK

● Debugger

Simulator		⊙
Emulator	(MB2197)	×
	(MB2198)	⊙
Monitor		×

● Format

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

Parameter

breakpoint-number (default: decimal number)

Specify the breakpoint number.

Command qualifier

/ALL (default when omitted)

Displays all breakpoints.

● Description

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

● Example

[Simulator]

>SHOW DATABREAK

no.	en/dis	address	read/write	pass-count	symbol
1	enable	00001000	write only	5 (2)	

Control: BREAK

Command: show status

[MB2198]

>SHOW DATABREAK

no.	en/dis	address	read/write	pass-count	symbol
1	enable	00002000	read only	1 (0)	\trac
4	disable	00002052	write only	65535 (2345)	

Note:

On the Emulator Debugger, this command cannot use except the FR60Lite or FR80S. For details, refer to **SOFTUNE Workbench USER'S MANUAL "Break"**.

3.11 CANCEL DATABREAK

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

■ CANCEL DATABREAK

● Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	⊙*
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)

Specify the breakpoint number.

Use the SHOW DATABREAK command to reference the set breakpoint numbers.

Command qualifier

/ALL

Cancels all data access breakpoints.

● Description

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

● Example

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

Note:

On the Emulator Debugger, this command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Break".

3.12 ENABLE DATABREAK

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

■ ENABLE DATABREAK

● Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	⊙*
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)

Specify the breakpoint number.

Use the SHOW DATABREAK command to reference the set breakpoint numbers.

Command qualifier

/ALL

Enables all data breakpoints.

● Description

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

● Example

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

Note:

On the Emulator Debugger, this command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Break".

3.13 DISABLE DATABREAK

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

■ DISABLE DATABREAK

● Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	⊙*
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)

Specify the a breakpoint number.

Use the SHOW DATABREAK command to reference the set breakpoint numbers.

Command qualifier

/ALL

Disables all data breakpoints temporarily.

● Description

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

● Example

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

Note:

On the Emulator Debugger, this command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Break".

3.14 SET EVENT

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

■ SET EVENT

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

[Event mode: Normal]

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

[Event mode: Performance]

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

Parameters

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

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

p=pass count (default: decimal number)

Specify the event occurrence count (1 to 16777215).

If omitted, it is set to 1.

It can be specified when the event mode is normal.

b=event number (performance)

Specify the event number (1 to 4)

When the event mode is performance, the specification of this parameter cannot be omitted.

Command qualifiers

- Access attributes

`/CODE`

Takes code access to specified address as event generating condition. Cannot be specified for FR80S.

`/READ`

Takes read access to specified address as event generating condition. Cannot be specified for FR80S.

`/WRITE`

Takes write access to specified address as event generating condition.

`/CODE/WRITE` cannot be omitted for FR60lite. If omitted, it means `/CODE` is specified.

For FR80S, the omission means `/WRITE` is specified.

- Data length specifying

`/WORD`

Specifies handling of event condition data as word data.

`/HALFWORD`

Specifies handling of event condition data as halfword data.

`/BYTE`

Specifies handling of event condition data as byte data.

`/ANYTHING` (Default when omitted)

An event generation condition is applied when the specified address is accessed, regardless of the access data length.

● Description

[Event mode: Normal]

The SET EVENT command sets the event that triggers a sequencer. If data is omitted, it will be all "don't care".

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.

You can reference the event number using of the SHOW EVENT command, or access using the embedded function `% EVENTNUM`.

[Event mode: Performance]

Specify the starting/ending addresses which are the conditions for measuring performance using the event number.

The following meanings are applied to each 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

● Example

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

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

Notes:

- This command cannot use except the FR60Lite or FR80S.
 - For FR80S, this is valid only when the setting is for the built-in RAM area.
 - The setting contents differ by event mode.
For details, refer to SOFTUNE Workbench USER'S MANUAL "Control by Sequencer" or "Measuring Performance".
 - 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 EVENT/ANYTHING 0x10000, D=0x12000000&=0xff000000
Ex.) For 16-bit data (0x4567)
SET EVENT/ANYTHING 0x10000, D=0x45670000&=0xffff0000
-

3.15 SHOW EVENT

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

■ SHOW EVENT

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
Monitor		×

● Format

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

Parameter

event-number

Specify the event number (1 to 4)

Command qualifier

/ALL (default when omitted)

Displays all events.

● Description

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

● Example

>SHOW EVENT

Note:

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

3.16 CANCEL EVENT

The **CANCEL EVENT** command cancels the event corresponding to a specified event number.

■ CANCEL EVENT

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

- Format

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

Parameter

event-number

Specify the event number (1 to 4)

Command qualifier

/ALL (default when omitted)

Cancels all events.

- Description

The CANCEL EVENT command cancels the event corresponding to a specified event number.

- Example

>CANCEL EVENT

Note:

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

3.17 ENABLE EVENT

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

■ **ENABLE EVENT**

● **Debugger**

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● **Format**

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

Parameter

event-number

Specify the event number (1 to 4).

Command qualifier

/ALL (default when omitted)

Enables all events.

● **Description**

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

● **Example**

>ENABLE EVENT

Note:

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

3.18 DISABLE EVENT

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

■ DISABLE EVENT

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

- Format

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

Parameter

event-number

Specify the event number (1 to 4).

Command qualifier

/ALL (default when omitted)

Disables all events.

- Description

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

- Example

>DISABLE EVENT

Note:

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

3.19 SET CODEEVENT

The SET CODEEVENT command sets a code event.

■ SET CODEEVENT

● Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

● Format

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

Parameters

event-number

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

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

Specify 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++ cannot be set.

pass-count (default: decimal number)

Specify 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 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:

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

For details, refer to SOFTUNE Workbench USER's MANUAL "BREAK".

3.20 SHOW CODEEVENT

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

■ SHOW CODEEVENT

● Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		×

● Format

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

Parameter

event-number

Specify 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 format:

```
no. en/dis      addr mask pass symbol
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  0000EFFF  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.
- This function cannot be used only when the FR60Lite is used.
For details, refer to SOFTUNE Workbench USER's MANUAL "BREAK".

3.21 CANCEL CODEEVENT

The CANCEL CODEEVENT command cancels the specified event.

■ CANCEL CODEEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

- Format

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

Parameter

event-number

Specify the event number (1 or 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:

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

For details, refer to SOFTUNE Workbench USER's MANUAL"BREAK".

3.22 ENABLE CODEEVENT

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

■ ENABLE CODEEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

- Format

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

Parameter

event-number

Specify the event number (1 or 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:

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

For details, refer to SOFTUNE Workbench USER's MANUAL "BREAK".

3.23 DISABLE CODEEVENT

The **DISABLE CODEEVENT** command disables the specified event.

■ DISABLE CODEEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

- Format

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

Parameter

event-number

Specify the event number (1 or 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:

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

For details, refer to SOFTUNE Workbench USER's MANUAL "BREAK".

3.24 SET DATAEVENT

The SET DATAEVENT command sets a data event.

■ SET DATAEVENT

● Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

● Format

SET DATAEVENT event-number, address [& = address-mask] [, [!] D = data [& = data-mask]]

Parameters

event-number

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

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

Specify 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 cannot be set.

[!] D=data [&=data-mask] (data formula, data formula)

Specify 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 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:

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

For details, refer to SOFTUNE Workbench USER's MANUAL "BREAK".

3.25 SHOW DATAEVENT

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

■ SHOW DATAEVENT

● Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		×

● Format

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

Parameter

event-number

Specify 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 format:

```
no. en/dis addr mask status size symbol
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_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  ize symbol
1  enable  R/W  00500000  -----  0000000A  FFFFFFFF word \R_Tpri_Lev
2  enable  R/W  00508038  -----  0051017C  FFFFFFFF word \task0
```

Note:

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

For details, refer to SOFTUNE Workbench USER's MANUAL "BREAK".

3.26 CANCEL DATAEVENT

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

■ CANCEL DATAEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

- Format

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

Parameter

event-number

Specify the event number (1 or 2).

Command qualifier

/ALL

Cancels all dataevents.

- Description

The CANCEL DATAEVENT command cancels the specified event.

- Example

```
>CANCEL DATAEVENT 1
>CANCEL DATAEVENT/ALL
```

Note:

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

For details, refer to SOFTUNE Workbench USER's MANUAL "BREAK".

3.27 ENABLE DATAEVENT

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

■ ENABLE DATAEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

- Format

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

Parameter

event-number

Specify the event number (1 or 2).

Command qualifier

/ALL

Enables all dataevents.

- Description

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

- Example

```
>ENABLE DATAEVENT 2
>ENABLE DATAEVENT/ALL
```

Note:

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

For details, refer to SOFTUNE Workbench USER's MANUAL "BREAK".

3.28 DISABLE DATAEVENT

The **DISABLE DATAEVENT** command disables the specified event.

■ DISABLE DATAEVENT

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

- Format

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

Parameter

event-number

Specify the event number (1 or 2).

Command qualifier

/ALL

Disables all dataevents.

- Description

The **DISABLE DATAEVENT** command disables the specified event.

- Example

```
>DISABLE DATAEVENT 2
>DISABLE DATAEVENT/ALL
```

Note:

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

For details, refer to SOFTUNE Workbench USER's MANUAL "BREAK".

3.29 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)	○
Monitor		×

● Format

SET SEQUENCE

Command qualifiers

/ON

Specify the delay count start triggered when event 1 condition and event 2 condition established in this order.

/OFF

Specify the delay count start triggered when event 1 condition or event 2 condition established.

/CODE

Specify the processing of code events.

/DATA

Specify the processing of data events.

/ALL

Specify the processing of code and data events.

● Description

The SET SEQUENCE command sets a SEQUENTIAL operation mode for code 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:

This function cannot be used only when the FR60Lite is used.
For details, refer to SOFTUNE Workbench USER's MANUAL "Control by sequencer".

3.30 SET SEQUENCE (type 2)

The SET SEQUENCE command sets a sequencer.

■ SET SEQUENCE (type 2)

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

SET SEQUENCE event-number[,event-number[,event-number]] [,R=event-number]

Parameters

event-number (default: decimal number)

Specify the number (D'1 to D'4) of the event to be set as a trigger factor.

R=event-number (default: decimal number)

Specify the numbers (D'1 to D'4) of the events that serve as the restarting conditions for the sequencer.

● Description

The sequencer is set. Three levels of sequence with RESTART are set.

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

```
>SET SEQUENCE 2,3,1,R=4
```

Notes:

- This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Control by Sequencer".
- This function cannot be used when the performance mode is set as the event mode.

3.31 SHOW SEQUENCE

The **SHOW SEQUENCE** command displays the event sequential operation mode set by the **SET SEQUENCE** command.

■ SHOW SEQUENCE

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		×

- Format

SHOW SEQUENCE

- Description

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.

- Example

```
>SHOW SEQUENCE
code sequence mode = ON
data sequence mode = OFF

[FR60Lite or FR80S]
>SHOW SEQUENCE
level1    -> level2      -> level3      -> end  restart
00FF0010  00FF0020      -----      -----
```

3.32 CANCEL SEQUENCE

The **CANCEL SEQUENCE** command cancels a sequencer.

■ CANCEL SEQUENCE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
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:

- This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Control by Sequencer".
 - This function cannot be used when the performance mode is set as the event mode.
-

3.33 ENABLE SEQUENCE

The **ENABLE SEQUENCE** command enables a sequencer.

■ ENABLE SEQUENCE

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

ENABLE SEQUENCE

● Description

The **ENABLE SEQUENCE** command enables a sequencer.

● Example

>ENABLE SEQUENCE

Notes:

- This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Control by Sequencer".
 - This function cannot be used when the performance mode is set as the event mode.
-

3.34 DISABLE SEQUENCE

The **DISABLE SEQUENCE** command disables a sequencer.

■ DISABLE SEQUENCE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

- Format

DISABLE SEQUENCE

- Description

The **DISABLE SEQUENCE** command disables a sequencer.

- Example

>DISABLE SEQUENCE

Notes:

- This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Control by Sequencer".
 - This function cannot be used when the performance mode is set as the event mode.
-

3.35 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)	○
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
```


3.36 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)	⊙
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
```

3.37 ENABLE ALIGNMENTBREAK

The **ENABLE ALIGNMENTBREAK** command enables alignment error break.

■ ENABLE ALIGNMENTBREAK

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
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
```

3.38 DISABLE ALIGNMENTBREAK

The **DISABLE ALIGNMENTBREAK** command disables alignment error break.

■ DISABLE ALIGNMENTBREAK

● Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
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
```

3.39 SHOW ALIGNMENTBREAK

The **SHOW ALIGNMENTBREAK** command displays the alignment error break status.

■ SHOW ALIGNMENTBREAK

- Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		×

- Format

SHOW ALIGNMENTBREAK

- Description

The **SHOW ALIGNMENTBREAK** command displays the alignment error break status.

- Example

```
>SHOW ALIGNMENTBREAK  
code = enable  
data = disable
```

3.40 SET BREAKCONDITION (type 1)

The SET BREAKCONDITION sets a data monitoring break (software).

■ SET BREAKCONDITION (type 1)

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

SET BREAKCONDITION[/ADDRESS] address, data

Parameters

address (address formula)

Specify the address at which monitoring is performed.

data (data formula)

Specify the data value that serves as the criteria for comparison for the data of the above address.

Function type command qualifiers

/ADDRESS

Specify 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)

Specify that data to be monitored is 8-bit length.

/HALFWORD

Specify that data to be monitored is 16-bit length.

/WORD

Specify 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
```

Note:

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).

3.41 SET BREAKCONDITION (type 2)

The SET BREAKCONDITION sets a data monitoring break (software).

■ SET BREAKCONDITION (type 2)

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
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)

Specify the data value that serves as the criteria for comparison for the data of the above address.

Function type command qualifiers

/REGISTERINDIRECT

Specify 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)

Specify that data to be monitored is 8-bit length.

/HALFWORD

Specify that data to be monitored is 16-bit length.

/WORD

Specify 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
```

Note:

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).

3.42 SHOW BREAKCONDITION

Setting information specified by the SET BREAKCONDITION command is displayed.

■ SHOW BREAKCONDITION

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
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
- 4.2 SHOW PERFORMANCE
- 4.3 CLEAR PERFORMANCE
- 4.4 SET COVERAGE
- 4.5 SHOW COVERAGE
- 4.6 CANCEL COVERAGE
- 4.7 CLEAR COVERAGE
- 4.8 SHOW CALLS
- 4.9 SHOW TIMER
- 4.10 CLEAR TIMER
- 4.11 SET TRACE (type 1)
- 4.12 SET TRACE (type 2)
- 4.13 SHOW TRACE (type 1)
- 4.14 SHOW TRACE (type 2)
- 4.15 CLEAR TRACE
- 4.16 ENABLE TRACE
- 4.17 DISABLE TRACE
- 4.18 SEARCH TRACE
- 4.19 SET DATATRACEAREA
- 4.20 SHOW DATATRACEAREA
- 4.21 CANCEL DATATRACEAREA

- 4.22 SHOW DETAILTRACE
- 4.23 SET TRACETRIGGER
- 4.24 SHOW TRACETRIGGER
- 4.25 CANCEL TRACETRIGGER
- 4.26 ENABLE TRACETRIGGER
- 4.27 DISABLE TRACETRIGGER
- 4.28 SET RAMCHECK
- 4.29 SHOW RAMCHECK
- 4.30 CANCEL RAMCHECK
- 4.31 ENABLE RAMCHECK
- 4.32 DISABLE RAMCHECK

4.1 SET PERFORMANCE

The **SET PERFORMANCE** command sets the operation of a performance measurement buffer when it is full.

■ SET PERFORMANCE

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

SET PERFORMANCE

Command qualifiers

/BREAK (default when omitted)

Causes break when performance measurement buffer becomes full.

/NOBREAK

Does not cause break when performance measurement buffer becomes full.

● 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 SOFTUNE Workbench USER'S MANUAL "Measuring Performance".
- This function cannot be used when the trace mode is set as the event mode.

4.2 SHOW PERFORMANCE

The state of the setting, and the measured results are displayed.

■ SHOW PERFORMANCE

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
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]

Command qualifiers

/STATUS

Displays operation setting state of performance measurement buffer when full.

/COUNT

Displays count result of times that event occurs.

/TIME (default when omitted)

Displays the result of time measurement in a specified area.

Parameters

- event number

Specify 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.

- Area number

Specify an area number (1 or 2) to display the result of its time measurement.

- lower limit (default: decimal number)

Specify the lower limit at which the measured time is displayed graphically. The unit is 1 ns.

- upper limit (default: decimal number)

Specify the upper limit at which the measured time is displayed graphically. The unit is 1 ns.

- Display interval (default: decimal number)

Specify the interval at which the measured time is displayed graphically. The unit is 1 ns.

- Description

- When /STATUS specified

The SHOW PERFORMANCE command displays the setting state of the operation of a performance measurement buffer when it is full.

- When /COUNT specified

The SHOW PERFORMANCE command displays the result of counting the time an event occurs.

- When /TIMER specified

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. The specified unit follows the setting in the SET TIMERSCALE command.

- Example

```
>SHOW PERFORMANCE/COUNT
```

Notes:

- This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Measuring Performance".
 - This function cannot be used when the trace mode is set as the event mode.
-

4.3 CLEAR PERFORMANCE

The **CLEAR PERFORMANCE** command cancels the performance measurement values.

■ CLEAR PERFORMANCE

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

- Format

CLEAR PERFORMANCE

- Description

The **CLEAR PERFORMANCE** command cancels the performance measurement values.

- Example

>CLEAR PERFORMANCE

Notes:

- This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Measuring Performance".
 - This function cannot be used when the trace mode is set as the event mode.
-

4.4 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)	×
Monitor		×

*: It is possible to use only for the high-speed version simulator debugger.

● Format

SET COVERAGE [measurement range]

Parameter

measurement range (address formula)

Specify 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 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
```

4.5 SHOW COVERAGE

The **SHOW COVERAGE** command displays the result of coverage measurement in a specified measurement range.

■ SHOW COVERAGE

● Debugger

Simulator		○*
Emulator	(MB2197)	×
	(MB2198)	×
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)

Specify 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 at start-up)

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%)
```

4.6 CANCEL COVERAGE

The **CANCEL COVERAGE** cancels the coverage measurement area and disables the coverage measurement function.

■ CANCEL COVERAGE

● Debugger

Simulator		○*
Emulator	(MB2197)	×
	(MB2198)	×
Monitor		×

*: It is possible to use only for the high-speed version simulator debugger.

● Format

CANCEL COVERAGE [measurement range]

Parameter

measurement range (address formula)

Specify 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
```

4.7 CLEAR COVERAGE

The **CLEAR COVERAGE** clears a coverage buffer.

■ CLEAR COVERAGE

- Debugger

Simulator		○*
Emulator	(MB2197)	×
	(MB2198)	×
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

4.8 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"/>
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 255 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++ 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++ function called.

Note the following when using the SHOW CALLS command:

- 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
cheker (12, 8)
main (3, 4)
```

4.9 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		<input type="radio"/>
Emulator	(MB2197)	<input type="radio"/>
	(MB2198)	<input type="radio"/>
Monitor		<input type="radio"/>

● 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 numeric values displayed are those after the RESET command has been executed and those when and after program execution is started by the preceding GO, STEP, or CALL command.

The SHOW TIMER command in the simulator debugger displays the cycle count and step count.

The SHOW TIMER command in the emulator debugger displays the cycle count.

The SHOW TIMER command in the monitor debugger displays time.

[MB2198]

The instruction execution cycle count and execution time of the executed program are displayed.

The resolution of the execution time is 25 ns, up to $(56^{\text{th}} \text{ power of } 2 - 1) \times 25 \text{ ns}$, and the maximum cycle count is $(56^{\text{th}} \text{ power of } 2 - 1)$; that is, measurements can be made up to 72057594037927935 cycles.

● Example

```
>SHOW TIMER
<timer>  From reset      :    12345678
          From Last Executed :    987654
<step>   From reset      :    6010712
          From Last Executed :    501142

[MB2198]
>SHOW TIMER
<cycle>   From initialize:  4210826410 [Cycle]
          From Last Executed: 362387415 [Cycle]
<timer>   From initialize:  0h00m42s108ms264us250ns [Time]
          From Last Executed: 0h00m03s623ms874us375ns [Time]
```


Note:

On the 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.

Pipeline execution times are not simulated on the Simulator Debugger.

4.10 CLEAR TIMER

The **CLEAR TIMER** command clears the timer measurement execution result.

■ CLEAR TIMER

- Debugger

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

- Format

CLEAR TIMER

- Description

The **CLEAR TIMER** command clears the timer measurement execution result.

- Example

```
>CLEAR TIMER
```

4.11 SET TRACE (type 1)

Controls the trace buffer full break.

■ SET TRACE (type 1)

● Debugger

Simulator		○
Emulator	(MB2197)	×
	(MB2198)	×
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
```

4.12 SET TRACE (type 2)

Controls the trace buffer full break.

■ SET TRACE (type 2)

● Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

● Format

SET TRACE

- Command qualifiers

Trace buffer-full break specified

/BREAK

The program is suspended when the trace buffer is full.

/NOBREAK (default at start-up)

The program is not suspended when the trace buffer is full. Trace measurements are continued.

[MB2198 (FR60Lite, FR80S)]

/STOP

The program is not suspended when the trace buffer is full. Trace measurements are stopped.

/NOSTOP

The program is not suspended when the trace buffer is full. Trace measurements are kept.

Trace sampling mode specified.

/FULL (default at start-up)

Measures the trace from program execution start to stop.

This mode cannot be specified with FR60Lite or FR80S.

/TRIGGER

The code event, data event and trace trigger are used as the causes of trace control.

This mode cannot be specified with FR60Lite or FR80S.

● Description

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.

- Example

```
>SET TRACE/TRIGGER /BREAK
```

4.13 SHOW TRACE (type 1)

The **SHOW TRACE** command displays the trace data stored in the trace buffer.

■ SHOW TRACE (type 1)

● Debugger

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

● Format

SHOW TRACE [/DATA] [trace-number [.. trace-number]]

SHOW TRACE FILE [/APPEND] file-nane

Command qualifiers classified by function

/DATA (default when omitted)

Displays traced data.

This qualifiers is specified in Format 1.

Parameters

trace-number (default: decimal number)

Specify the number of trace data to be displayed with decimal number.

file-nane

The file that saves the displayed trace data is specified.

Command qualifiers

/CYCLE (default when omitted)

Displays data in valid bus cycle. For the DSU3, this qualifier cannot be specified.

/INSTRUCTION

Executes disassemble display. For the DSU3, this qualifier cannot be specified.

/SOURCE

Displays trace result in units of source lines. For the DSU3, this qualifier cannot be specified.

/ONEFRAME

Displays trace data only by one line.

/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.

When /INSTRUCTION is specified, the command executes disassemble display according to the sampled trace data.

When /SOURCE is specified, the command displays source lines according to the sampled trace data.

For the DSU3 chip, PC and data access information at branch are displayed. This command does not display the information about instructions being executed while branch instruction. So, /CYCLE, /INSTRUCTION and /SOURCE qualifiers cannot be specified. Use the SHOW DETAILTRACE command when displaying the information about instructions being executed while branch instruction.

When trace-number is omitted, the command starts trace data display from the oldest trace data or the trace data of the trace number next to the previously-displayed last trace number.

When only the display start trace number is specified, the command displays 12 trace data, starting from the display start trace number.

When the display start trace number is less than the trace number of the oldest trace data, the command starts trace data displays from the oldest trace data.

When /STATUS is specified, the command displays the current trace status.

In the MB2198, the time stamp is displayed when the real time trace interface is provided.

When "/FILE" is specified, trace data is overwritten to a file.

When "/APPEND" is specified, trace data is added and saved.

● Example

```
>SHOW TRACE/SOURCE -65
frame no.          source
-00065 :demo3.c$489  if (sy->str[0] == ab1)
-00059 :demo3.c$491   }
-00055 :demo3.c$487   for ( i = 0 ; i < 12 ; i++ ) {
-00052 :demo3.c$492   ackdat += 5;
-00047 :demo3.c$493   nckdat = ackdat;
-00043 :demo3.c$494   return (ab1);
-00042 :demo3.c$495   }
-00038 :demo3.c$464   if (rc != 0)
-00035 :demo3.c$465   sy->dat1 = 0x21;
-00031 :demo3.c$467   }
-00027 :demo3.c$460   for ( i = 0 ; i < NUM ; i++) {
-00024 :demo3.c$468   return (0);
>SHOW TRACE/FILE/APPEND  C:\sample.log
```

4.14 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"/>
Monitor		<input checked="" type="radio"/>

● 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

```
>SHOW TRACE/STATUS
en/dis= enable
buffer full          = nobreak
frame no.           = -00120 to 00000
```


4.15 CLEAR TRACE

The **CLEAR TRACE** command clears the trace buffer.

■ CLEAR TRACE

- Debugger

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

- Format

CLEAR TRACE

- Description

The **CLEAR TRACE** command clears the trace buffer.

- Example

```
>CLEAR TRACE
```

4.16 ENABLE TRACE

The **ENABLE TRACE** command enables the trace function.

■ ENABLE TRACE

- Debugger

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

- Format

ENABLE TRACE

- Description

The **ENABLE TRACE** command enables the trace function.

- Example

```
>ENABLE TRACE
```

4.17 DISABLE TRACE

The **DISABLE TRACE** command disables the trace function.

■ DISABLE TRACE

● Debugger

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

● 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
```

4.18 SEARCH TRACE

The **SEARCH TRACE** command searches for trace data according to the specified condition.

■ SEARCH TRACE

● Debugger

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

● 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)

Specify the address to be searched.

data (data formula)

Specify the data transmission register access data to be searched.

This parameter is valid only when the debugger type is an emulator debugger.

mask-data (data formula)

Specify 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)

Specify 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 with DSU3.

/INSTRUCTION

If trace data cannot be rearranged in machine instruction execution units, it is searched for in machine cycles.

/WORD (DSU3)

Specifies that event condition data to be treated as 4-byte data.

/HALFWORD (DSU3)

Specifies that event condition data to be treated as 2-byte data.

/BYTE (DSU3)

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.

/CODE (DSU3)

Searches trace frame or step where code access made to specified address.

/READ (DSU3)

Searches trace frame or step where read access made to specified address.

/WRITE (DSU3)

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:        000F0AD4A     DDSP      #4
```

4.19 SET DATATRACEAREA

The **SET DATATRACEAREA** command specifies the data access area where trace measurement is performed.

■ SET DATATRACEAREA

● Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

● Format

SET DATATRACEAREA address [& = address-mask]

Parameters

address (address formula)

Specify the address for trace measurement.

address mask (data formula)

Specify the bit mask of address. It is specified by checking only the bits set to 1s of the address mask.

Command qualifiers

/READ

Performs trace measurement upon read access to a specified address area.

/WRITE

Performs trace measurement upon write access to a specified address area.

/CODE

Performs trace measurement upon code access to a specified address area.

● 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 assuming /READ/WRITE/CODE as being specified.

If no address mask is specified, processing will be performed assuming the address mask as being H'0FFFFFFF.

● Example

```
>SET DATATRACEAREA/WRITE 180 & 7f
```

Note:

Specification for this command can be different depending on MCU mode and DSU type.

MCU mode	DSU Type	
	DSU3	DSU4
Full trace	○	○
Real time	○	○
External trace	○	Adapter board connection : ◎
		cable connection : ○
Internal trace	×	×

◎ :Command is valid

Attribution of code can be specified.

○ :Command is valid

Attribution of 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.

4.20 SHOW DATATRACEAREA

The set data trace measurement area is displayed.

■ SHOW DATATRACEAREA

● Debugger

Simulator		×
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		×

● Format

SHOW DATATRACEAREA

● Description

The set data trace measurement area is displayed.

This function cannot be used in Internal trace mode, as the MCU mode.

● Example

```
>SHOW DATATRACEAREA
address      mask      access
00000000    FFFFFFFF  read/write/code
```


4.21 CANCEL DATATRACEAREA

The set data trace measurement area is deleted.

■ CANCEL DATATRACEAREA

- Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

- Format

CANCEL DATATRACEAREA

- Description

The set data trace measurement area is deleted.

This function cannot be used when Internal trace mode as the MCU mode.

- Example

```
>SHOW DATATRACEAREA
address      mask      access
00001000    0001FFFF    read
>CANCEL DATATRACEAREA
address      mask      access
00000000    FFFFFFFF    read/write/code
```

4.22 SHOW DETAILTRACE

The trace is displayed in detail.

■ SHOW DETAILTRACE

● Debugger

Simulator		×
Emulator	(MB2197)	○
	(MB2198)	○
Monitor		×

● Format

SHOW DETAILTRACE [frame-display-starting-line] [, f = frame-number]

Parameters

frame display starting line (default: decimal number)

Specify the count of lines on which the detailed display of a specified frame is started.

frame number (default: decimal number)

Specify the number of the frame to be displayed. When omitted, the same frame number as the previous one is used.

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

The SHOW DETAILTRACE command can be specified only by the DSU3 chip. The command performs the following operations on the branch address information and data access information displayed by the SHOW TRACE command and displays the results.

- 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, 12 lines of information will be displayed.

● Example

```
>SHOW DETAILTRACE 1, f=-198
frame no.      address      mnemonic (-00198 .. -00194)
-00198   :      000F0052      BNE          000F0048
          :      000F0054      LDI:32      #00005632,R14
          :      000F005A      LDUH        @R14,R0
-00197   :      read      1234 at 00005632
          :      000F005C      LDUB        @(R14,2),R1
-00196   :      read      98 at 00005634
          :      000F005E      LDI:8       #10,R2
          :      000F0060      STB         R2,@(R14,4)
-00195   :      write     10 at 00005636
-00194   :      000F0062      BRA          000F0048
```

4.23 SET TRACETRIGGER

When the specified conditions are satisfied, the state of trace measurements is controlled.

■ SET TRACETRIGGER

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

SET TRACETRIGGER address [&=address mask][,!d=data[&=data mask]]

Parameters

address (address expression)

Specify the address that serves as a trace trigger condition.

address mask (data expression)

Specify 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.

data (data expression)

Specify data that serves as a trace trigger condition.

Data cannot be specified under the code execution condition.

mask data (data expression)

Specify 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.

Command qualifiers

• Access attributes

/READ

Specify data read access as a trace trigger condition. Cannot be specified for FR80S.

/WRITE (default when FR80S is omitted)

Specify data write access as a trace trigger condition.

/CODE (default when FR60Lite is omitted)

Specify the code execution as a trace trigger condition. Cannot be specified for FR80S.

• Data length specifying

/BYTE

Specify byte access as a trace trigger condition.

/HALFWORD

Specify halfword (2-byte) access as a trace trigger condition. This parameter cannot be specified if /CODE is set.

/ANYTHING (Default when omitted)

A trace trigger condition is applied when the specified address is accessed, regardless of the access data length.

/WORD

Specify word (4-byte) access as a trace trigger condition. This parameter cannot be specified if /CODE is set.

- Trace control

/ENABLETRACE (default when omitted)

When the trace trigger conditions are satisfied, trace measurements are started.

/DISABLETRACE

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.

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.

● Example

```
>SET TRACETRIGGER/CODE/ENABLETRACE FF0000
```

```
>SET TRACETRIGGER/READ/WORD/ENABLETRACE 120046!D=1234&=FFFC
```

Notes:

- This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Trace".
- For FR80S, this is valid only when the setting is for the built-in RAM area.
- This function cannot 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=12000000&=ff000000
```

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

```
SET TRACETRIGGER/ANYTHING 0x10000, D=45670000&=ffff0000
```

4.24 SHOW TRACETRIGGER

The state of the trace trigger set by the SET TRACETRIGGER command is displayed.

■ SHOW TRACETRIGGER

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
Monitor		×

● Format

SHOW TRACETRIGGER[trigger-number[,...]]

Parameter

trigger-number (default: decimal)

Specify 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.

● Example

```
>SHOW TRACETRIGGER
no. trace-ctrl  address  address-mask data  d-mask size access  symbol
1   enable     00FF0010 00FFFFFF0    ...   ...   ... code
2   disable     01200430 ...          !A5A5 F5FF  word read/write symbol
```

Notes:

- This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Trace".
- This function cannot be used when the performance mode is set as the event mode.

4.25 CANCEL TRACETRIGGER

The trace trigger set by the SET TRACETRIGGER command is deleted.

■ CANCEL TRACETRIGGER

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

CANCEL TRACETRIGGER[trigger-number[,...]]

Parameter

trigger-number (default: decimal)

Specify the number of the trace trigger to be deleted.

Command qualifiers

/ALL (default when omitted)

All trace triggers set by the SET TRACETRIGGER command are deleted.

● Description

The trace trigger set by the SET TRACETRIGGER command is deleted.

● Example

```
>CANCEL TRACETRIGGER/ALL
```

Notes:

- This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Trace".
 - This function cannot be used when the performance mode is set as the event mode.
-

4.26 ENABLE TRACETRIGGER

The trace trigger set by the SET TRACETRIGGER command is enabled.

■ ENABLE TRACETRIGGER

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

ENABLE TRACETRIGGER [trigger-number [, . . .]]

Parameter

trigger-number (default: decimal)

Specify the number of the trace trigger to be enabled.

Command qualifiers

/ALL (default when omitted)

All trace triggers set by the SET TRACETRIGGER command are enabled.

● Description

The trace trigger set by the SET TRACETRIGGER command is enabled.

● Example

```
>ENABLE TRACETRIGGER/ALL
```

Notes:

- This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Trace".
 - This function cannot be used when the performance mode is set as the event mode.
-

4.27 DISABLE TRACETRIGGER

The trace trigger set by the SET TRACETRIGGER command is disabled.

■ DISABLE TRACETRIGGER

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

DISABLE TRACETRIGGER[trigger-number[,...]]

Parameter

trigger-number (default: decimal)

Specify the number of the trace trigger to be disabled.

Command qualifiers

/ALL (default when omitted)

All trace triggers set by the SET TRACETRIGGER command are disabled.

● Description

The trace trigger set by the SET TRACETRIGGER command is disabled.

● Example

```
>DISABLE TRACETRIGGER/ALL
```

Notes:

- This command cannot use except the FR60Lite or FR80S. For details, refer to SOFTUNE Workbench USER'S MANUAL "Trace".
 - This function cannot be used when the performance mode is set as the event mode.
-

4.28 SET RAMCHECK

Sets the RAM Checker.

■ SET RAMCHECK

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

[Format 1] SET RAMCHECK address

[Format 2] SET RAMCHECK file-name

Parameter

- Format 1
address
Specify the monitoring address to be set.
- Format 2
file-name
Specify the name of the log file that stores sampling data.

Command qualifiers

- Format 1

Data size

/BYTE

Specify a data size in bytes.

/HALFWORD

Specify a data size in halfwords. The lower 1 bit of the monitoring address is ignored.

/WORD (default)

Specify a data size in words. The lower 2 bits of the monitoring address are ignored.

Access attribute

/READ

Specify data read access as a data monitoring condition. Cannot be specified for FR80S.

/WRITE

Specify data write access as a data monitoring condition.

When the specification FR60Line is omitted, the access attribute is set to /READ/WRITE.

For FR80S, the omission means /WRITE is set.

- Format 2

/SOFTUNE

Specify when storing files in the SOFTUNE format.

When an extension is omitted, ".SRL" is added.

/CSV

Specify 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

Specify 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 "RAM Checker" section in "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.
-

4.29 SHOW RAMCHECK

Displays the setting of the RAM Checker.

■ SHOW RAMCHECK

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
Monitor		×

● Format

SHOW RAMCHECK

Parameter

None

Command qualifiers

None

● Description

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 "RAM Checker" section in "SOFTUNE Workbench User's Manual".
 - This command can be used only when debug mode is RAM Checker mode.
-

4.30 CANCEL RAMCHECK

Deletes the setting of the RAM Checker.

■ CANCEL RAMCHECK

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
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

Deletes the monitoring address 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 "RAM Checker" section in "SOFTUNE Workbench User's Manual".
 - This command can be used only when debug mode is RAM Checker mode.
-

4.31 ENABLE RAMCHECK

Enables the logging status of the RAM Checker.

■ ENABLE RAMCHECK

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

ENABLE RAMCHECK

Parameter

None

Command qualifiers

None

● Description

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 "RAM Checker" section in "SOFTUNE Workbench User's Manual".
- This command can be used only when debug mode is RAM Checker mode.

4.32 DISABLE RAMCHECK

Disables the logging status of the RAM Checker.

■ DISABLE RAMCHECK

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	○
Monitor		×

● Format

DISABLE RAMCHECK

Parameter

None

Command qualifiers

None

● Description

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 "RAM Checker" section in "SOFTUNE Workbench User's Manual".
 - This command can be used only when debug mode is RAM Checker mode.
-

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)	⊙*
Monitor		○

*: The DSU3 chip cannot specify this command during the program running.

● Format

EXAMINE expression [, ...]

Parameter

expression (address formula)

Specify expression to be analyzed.

Command qualifiers

/BINARY

Specifies that value to be examined to be displayed as binary number.

/OCTAL

Specifies that value to be examined to be displayed as octal number.

/DECIMAL

Specifies that value to be examined to be displayed as decimal number.

/HEXADECIMAL

Specifies that value to be examined to be displayed as a hexadecimal 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
```

5.2 ENTER

The ENTER command assigns the specified data to the specified variable.

■ ENTER

● Debugger

Simulator		☉
Emulator	(MB2197)	☉
	(MB2198)	☉*
Monitor		○

*: The DSU3 chip cannot specify this command during the program running.

● Format

ENTER variable = data

Parameters

variable (address formula)

Specify the variable where data to be stored.

data (data formula)

Specify the data to be stored.

Command qualifiers

/BYTE

Stores specified value in specified memory location as 1-byte data.

/HALFWORD

Stores specified value in specified memory location as 2-byte data.

/WORD

Stores specified value in specified memory location as 4-byte data.

/DWORD

Stores specified value in specified memory location as 8-byte data.

/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)	⊙*
Monitor		○

*: The DSU3 chip cannot specify this command during the program running.

● Format

SET MEMORY [storage-address] = data [, ...]

Parameters

storage-address (address formula)

Specify the memory location where specified data to be stored.

data (data formula)

Specify the value to be stored.

Command qualifiers

/BIT

Stores specified value in specified memory location as bit-length data.

/BYTE (default when omitted)

Stores specified value in specified memory location as 1-byte data.

/HALFWORD

Stores specified value in specified memory location as 2-byte data.

/WORD

Stores specified value in specified memory location as 4-byte data.

/DWORD

Stores specified value in specified memory location as 8-byte data.

/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)	⊙*
Monitor		○

*: The DSU3 chip cannot specify this command during the program running.

● Format

SHOW MEMORY [{address | address-range} [, ...]]

Parameters

address (address formula)

Specify the address in memory location to be checked.

address-range (address formula)

Specify the memory area range to be checked.

Command qualifiers

/BIT

Specifies that value to be checked to be displayed as 1-bit data.

/BYTE (default when omitted)

Specifies that value to be checked to be displayed as 1-byte data.

/HALFWORD

Specifies that value to be checked to be displayed as 2-byte data.

/WORD

Specifies that value to be checked to be displayed as 4-byte data.

/DWORD

Specifies that value to be checked to be displayed as 8-byte data.

/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)	○
Monitor		○

● Format

SEARCH MEMORY address-range = data [, ...] [, S = skip-byte-count]

Parameters

address-range (address formula)

Specify the memory area to be searched.

data (data formula)

Specify the data to be searched.

skip-byte-count (data formula)

Specify 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

/BYTE (default when omitted)

Searches for specified data as byte-length data.

/HALFWORD

Skips 2 bytes and searches for specified data as 2-byte data.

/WORD

Skips 4 bytes and searches for specified data as 4-byte data.

/DWORD

Skips 8 bytes and searches for specified data as 8-byte data.

/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"/>
Monitor		<input type="radio"/>

● Format

SET REGISTER register name = data

Parameters

register-name

Specify the name of register or flag to be modified.

data (data formula)

Specify 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:

- 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'0FFFFFFC00 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"/>
Monitor		<input type="radio"/>

● Format

SHOW REGISTER register name

Parameter

register-name

Specify the name of register or flag to be checked.

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)	○
Monitor		○

● Format

COMPARE compare-origin-address-range, comparison-destination-address

Parameters

compare-origin-address-range (address formula)

Specify the memory area of compare origin.

comparison-destination-address (address formula)

Specify 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)	○
Monitor		○

● Format

FILL address-range = data [, ...]

Parameters

address-range (address formula)

Specify the memory range to be filled with data.

data (data formula)

Specify the data filling specified memory area.

Command qualifiers

/BYTE (default when omitted)

Specifies filling of memory area with 1-byte data.

/HALFWORD

Specifies filling of memory area with 2-byte data.

/WORD

Specifies filling of memory area with 4-byte data.

/DWORD

Specifies filling of memory area with 8-byte data.

/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)	○
Monitor		○

● Format

MOVE transfer-source-address-range, transfer-destination-address

Parameters

transfer-source-address-range (address formula)

Specify the memory area from where data transferred.

transfer-destination-address (address formula)

Specify 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)	☉*
Monitor		○

*: The DSU3 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

/BIT

Dumps data in bits.

/BYTE (default when omitted)

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 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"/>
Monitor		<input type="radio"/>

● Format

ASSEMBLE [starting-address] = assemble-character-string

Parameters

starting-address (address formula)

Specify a starting address of memory containing line-assembled codes.

assemble-character-string (character string)

Specify 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)	⊙*
Monitor		○

*: The DSU3 chip cannot specify this command during the program running.

● Format

DISASSEMBLE [{starting-address | address-range}]

Parameters

starting-address (address formula)

Specify a starting address of memory to be disassembled.

address-range (address formula)

Specify 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"/>
Monitor		<input type="radio"/>

● Format

LOAD file-name [, address] [, file-offset [, byte-count]]

Parameters

file-name

Specify a name of file to be loaded.

The default extension depends on the command qualifier to be specified.

address (address formula)

Specify 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)

Specify an offset of read 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)

Specify a count of data to be read.

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

/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 command definition).

The default extension is ".lst".

/AUTOMATIC (default when omitted)

For simulator debugger, automatically sets map area at loading.

/MANUAL

For simulator debugger, does not automatically set map area at loading.

A map area must be set by the SET MAP command.

/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.

/ONDEMAND

The load module file is on-demand loaded. This modifier can be specified when /OBJECT or /DEBUG is given.

- Data length

/BYTE (default when omitted)

Specifies loading to memory area with 1-byte data.

/HALFWORD

Specifies loading to memory area with 2-byte data.

/WORD

Specifies loading to memory area with 4-byte data.

/DWORD

Specifies loading to memory area with 8-byte data.

● Description

The LOAD command loads the specified file.

This command can load the following files:

- 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
```

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"/>
Monitor		<input type="radio"/>

● Format

SAVE file-name [, {address - range | module - name}]

Parameters

file-name

Specify 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)

Specify 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)	⊙
Monitor		⊙

● Format

LIST [{ [file-name] line-number [.. line-number] | address }]

Parameters

file-name

Specify a name of source file to be displayed.

When file-name is omitted, the previously-specified file name is assumed.

line-number

Specify 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)

Specify an address (memory location) where code attribute stored.

Specify 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)	⊙
Monitor		⊙

● Format

SET PATH [source-search-directory-name [, ...]]

Parameter

source-search-directory-name

Specify 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)	☉
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"/>
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"/>
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"/>
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)	⊙
Monitor		⊙

● Format

BATCH file-name [, actual-parameter [, ...]]

Parameters

file-name

Specify a name of file where command procedure to be executed written.

The default extension is ".prc".

actual-parameter

Specify 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)	⊙
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)	⊙
Monitor		⊙

● Format

SET ALIAS alias = command-character-string

Parameters

alias (identifier)

Specify a command alias.

command-character-string

Specify 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)	⊙
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)	⊙
Monitor		⊙

● Format

CANCEL ALIAS [alias [,...]]

Parameter

alias (identifier)

Specify 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)	⊙
Monitor		⊙

● Format

SET VARIABLE debug-variable-name = replacing-character-string

Parameters

debug-variable-name (identifier)

Specify a debug variable to be defined.

replacing-character-string

Specify 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.

● Example

```
>SET VARIABLE ADDR = 0309+12
>SET VARIABLE STR = "ABCDEF"
>SET MEMORY/STRING %ADDR = %STR can be replaced with SET MEMORY/
STRING 0309+12 = "ABCDE"
>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)	⊙
Monitor		⊙

● Format

SHOW VARIABLE [debug-variable-name [,...]]

Parameter

debug-variable-name (identifier)

Specify 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)	⊙
Monitor		⊙

● Format

CANCEL VARIABLE [debug-variable-name [,...]]

Parameter

debug-variable-name (identifier)

Specify 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 Replacement 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)	⊙
Monitor		⊙

● Format

SET LOGGING [file-name]

Parameter

file-name

Specify a log file name.

The default extension is ".LOG".

When file-name is omitted, the DEBUG.LOG file is used to log data.

Command qualifiers

/OPEN (default when omitted)

Newly opens specified file.

/APPEND

Appends log data to end of specified file.

/EXPANSION (default when omitted)

Logs command list and its result.

/UNEXPANSION

Logs only result.

/COMMAND

Logs only user-entered data.

● 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.

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.

● Example

```
>SET LOGGING filename.log
>
>SET LOGGING/COMMAND filename.log
```


11.2 SHOW LOGGING

The **SHOW LOGGING** command displays the logging status.

■ SHOW LOGGING

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		⊙

- Format

SHOW LOGGING

- Description

The **SHOW LOGGING** command displays the logging status.

- Example

```
>SHOW LOGGING
en/dis      :    ENABLE
logging file : logfile.log
logging data : EXPANSION
```

11.3 CANCEL LOGGING

The **CANCEL LOGGING** command cancels the logging setup and closes the logging file.

■ CANCEL LOGGING

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		⊙

- Format

CANCEL LOGGING

- Description

The CANCEL LOGGING command cancels the logging setup and closes the logging file.

- Example

```
>CANCEL LOGGING
```

11.4 ENABLE LOGGING

The **ENABLE LOGGING** command enables logging again.

■ ENABLE LOGGING

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		⊙

- Format

ENABLE LOGGING

- Description

The **ENABLE LOGGING** command enables logging again.

- Example

```
>ENABLE LOGGING
```

11.5 DISABLE LOGGING

The **DISABLE LOGGING** command temporarily disables logging.

■ DISABLE LOGGING

- Debugger

Simulator		⊙
Emulator	(MB2197)	⊙
	(MB2198)	⊙
Monitor		⊙

- Format

DISABLE LOGGING

- Description

The **DISABLE LOGGING** command temporarily disables logging.

The **ENABLE LOGGING** command can be used to enable logging again.

- Example

```
>DISABLE LOGGING
```

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)	⊙
Monitor		⊙

● Format

PRINTF "format-control-string" [, expression [, ...]]

Parameters

format-control-string

Specify character strings to be displayed on screen and format for expression value display.

Enclose format specification in double quotation marks (").

"% [flag] [width] [.precision] [l] type"

%

Specify this parameter when displaying data according to format specification.

The PRINTF command displays characters that are not format specification after % as they are.

flag

Specify 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

Specify 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

Specify 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.

l

Specify whether to display the language expression value as the long, unsigned long type when the conversion display format is d, u, o, x, or X.

When l is omitted, the language expression value is assumed to be the short, unsigned short type.

type

Specify 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.)

c: One character

b: Unsigned binary number

s: Character string (Only addressing is valid. The maximum number of characters is 128 bytes.)

expression

Specify 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)	⊙
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)	⊙
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)	⊙
Monitor		⊙

● Format

```
IF formula
    command-list
[ELSEIF formula
    command-list]
[ELSE
    command-list]
```

ENDIF

Parameters

formula

Specify the execution condition formula of specified command list.

command-list

Specify 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)	⊙
Monitor		⊙

● Format

REPEAT

command-list

UNTIL formula

Parameters

command-list

Specify the commands to be executed.

formula

Specify 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)	⊙
Monitor		⊙

● Format

```
WHILE formula
    command-list
```

```
ENDW
```

Parameters

formula

Specify the execution condition formula of specified command list.

command-list

Specify 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)	⊙
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 %SYMLN
- 13.15 %TOVAL
- 13.16 %TOSTR
- 13.17 %EVAL
- 13.18 %EVENTNUM
- 13.19 %TRIGGERNUM
- 13.20 %DEBUGFUNCTION

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"/>
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)	⊙
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)	⊙
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)	⊙
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)	⊙
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)	⊙
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)	⊙
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)	⊙*
Monitor		○

*: The DSU3 chip cannot specify this command during the program running.

● Format

%BIT(address)

%B(address)

%H(address)

%W(address)

%S(address)

%D(address)

Parameter

address

Specify 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 data

%H : Halfword data

%W : Word 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)	⊙
Monitor		⊙

- Format

`%STRGET(character-string, character-position, character-count)`

Parameters

character-string

Specify a replacing character string.

character-position

Specify a character position where get processing to be started (character position relative to first character).

character-count

Specify 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)	☉
Monitor		☉

● Format

%STRSTR(character-string-1, character-string-2)

Parameters

character-string-1

Specify a character string including character string to be retrieved.

character-string-2

Specify 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)	⊙
Monitor		⊙

● Format

%STRCMP(character-string-1, character-string-2)

Parameter

character-string-1,character-string-2

Specify 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)	⊙
Monitor		⊙

● Format

%STRLEN(character-string)

Parameter

character-string

Specify 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)	⊙
Monitor		⊙

● Format

`%STRCAT(character-string-1, character-string-2)`

Parameter

`character-string-1, character-string-2`

Specify 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)	⊙
Monitor		⊙

● **Format**

`%SYMLEN(symbol-name)`

Parameter

symbol-name

Specify 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)	⊙
Monitor		⊙

- Format

%TOVAL(character-string)

Parameter

character-string

Specify 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)	⊙
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)	⊙
Monitor		⊙

● Format

%EVAL(expression)

Parameter

expression

Specify 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)	⊙
Monitor		×

● Format

%EVENTNUM(address)

Parameter

address

Specify the address at which the event number automatically set is read.

● Description

The number of the event set at the specified address is returned.

If the address at which no event is set is specified, 0 is returned.

● Example

```
>PRINTF "event no : %0d\n", %EVENTNUM(F0000)
event no : 1
```

13.19 %TRIGGERNUM

%TRIGGERNUM returns the number of trace trigger.

■ %TRIGGERNUM

● Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
Monitor		×

● Format

%TRIGGERNUM(address)

Parameter

address

Specify the address at which the trace trigger number automatically set is read.

● Description

The number of the trace trigger set at the specified address is returned.

If the address at which no trace trigger is set is specified, 0 is returned.

● Example

```
>PRINTF "trace trigger no : %0d\n", % TRIGGERNUM(F0000)
trace trigger no : 2
```

13.20 %DEBUGFUNCTION

This replaces the current debug function number.

■ %DEBUGFUNCTION

- Debugger

Simulator		×
Emulator	(MB2197)	×
	(MB2198)	⊙
Monitor		×

- Format

%DEBUGFUNCTION

- Description

This replaces the current debug function number.

- Example

```
>PRINTF "debug function : %d\n",%DEBUGFUNCTION
debug function : 0
```


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 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	Function call failed. 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]	Click the OK button to create a directory.
E4206W	List file directory not found. Create a directory?
[Explanation]	The target list file directory is not specified.
[Operator response]	Click the OK button to create a directory.
E4207W	Object file directory not found. Create a directory?
[Explanation]	The target object file directory is not specified.
[Operator response]	Click the OK button to create a directory.
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 0xFFFFFF.
[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.

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.

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.

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.
[Operator response]	Use the Setup Wizard to restart the SOFTUNE debugger.
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.
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]	The monitor program loaded into the emulator is incorrect.
[Operator response]	Start the loader program attached to this product to load the monitor program, then restart the SOFTUNE debugger. For details, refer to the Installation Manual.

F9403S	Emulator hardware error.
[Explanation]	The emulator hardware cannot operate normally. Or, processing is stopped because an exception in detailed information is detected.
[Operator response]	<p>- "RAM checker over-flow"</p> <p>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.</p> <p>- No detailed information is displayed</p> <p>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.</p>
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 Installation Manual.
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 ICE environment.
[Explanation]	The ICE settings specified in the setup file are different from the ICE environment in connection.
[Operator response]	Start the setup wizard to change the ICE type in the setup file or correct the ICE environment.
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.

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]	"WINSOCK.DLL" should correspond to the LAN software to be used. Store "WINSOCK.DLL" to the Windows directory or PATH-set directory, referring to the manual for LAN software.
F9901S	Memory allocation error.
[Explanation]	Sufficient memory for startup cannot be allocated.
[Operator response]	Increase the host PC empty memory and then restart this program.
F9902S	System error.
[Explanation]	This program could not startup normally because of system error.
[Operator response]	Restart the system and then restart this program.
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.

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] If this message is output when the LOAD command is executed, 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.

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 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]	Re-enter the GO command in Run menu.

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 DLL file.

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.

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 0x1 to 0xff).
[Explanation]	The value of the specified bit pattern is other than 0x1 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
[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 this program.
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 there are too many device drivers.
[Operator response]	Increase the memory space in the host PC to restart this program.
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.
[Operator response]	Increase the memory space in the host PC memory, then restart this program.

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 the program. 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 MB2198 emulator, display the message.
[Operator response]	Check the contents of break points and maximum count of break points. In case of MB2198 emulator, the break point, the trace trigger and the sequencer may be used same hardware point. Check the status in event lists.
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 [Environment] - [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 manual of the chip.

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 data watch break or RAM checker is set with MB2198 emulator.
[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]	This message appears when the break condition cannot be set for reasons of short on the event resource with MB2198 emulator.
[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]	Please check the LSI specification or the Command Reference Manual based on MBXXXX in the error message.

E4144S	Please use the RAM checker function with an 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.

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. 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.

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] Or, load the load module with debug data.

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.

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.

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.

E4405S	Reserved
--------	----------

[Explanation]

[Operator response]

E4406S	Reserved
--------	----------

[Explanation]

[Operator response]

E4407S	Reserved
--------	----------

[Explanation]

[Operator response]

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]

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	Command error (user reset).
[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.
[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 LSI specifications based on detailed information.

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.

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.

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.

E4602S	Communication: Parallel adapter not connected.
--------	--

[Explanation] The parallel communication adapter is not connected.

[Operator response] Connect the parallel communication adapter correctly, then re-execute this program.

E4603S	Communication: Mismatch parallel adapter version.
--------	---

[Explanation] Communication cannot be performed because the version of the parallel communication adapter is old.

[Operator response] Use the latest parallel communication 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 SOFTUNE Workbench Operation Manual Appendix C [Setting LAN Interface].

E4605S	Communication: Cannot find port number.
--------	---

[Explanation] The port number of ICE is not defined in the services file.

[Operator response] Please register the port number in the services file.

For details, refer to the SOFTUNE Workbench Operation Manual Appendix C [Setting LAN Interface].

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.

APPENDIX

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.

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]	<p>This message is displayed when a break is caused by a trace buffer full.</p> <p>Address indicates the address of the next instruction to be executed where execution was suspended.</p>
---------------	--

Break at address by alignment error break (code)
--

[Explanation]	<p>This message is displayed when a break is caused by a code fetch alignment error.</p> <p>Address indicates the address of the next instruction to be executed where execution was suspended.</p>
---------------	---

Break at address by alignment error break (data)
--

[Explanation]	<p>This message is displayed when a break is caused by a data access alignment error.</p> <p>Address indicates the address of the next instruction to be executed where execution was suspended.</p>
---------------	--

Break at address by external trigger break
--

[Explanation]	<p>This message is displayed when a break is caused by the input of an external signal to the TRIG pin of the emulator.</p> <p>Address indicates the address of the next instruction to be executed where execution was suspended.</p>
---------------	--

Break at address by trace lost break

[Explanation]	<p>This message is displayed when a break is caused by the trace data loss.</p> <p>Address indicates the address of the next instruction to be executed where execution was suspended.</p>
---------------	--

Break at address by data break at access-address
--

[Explanation]	<p>This message is displayed when a break is caused by a data breakpoint.</p> <p>Address indicates the address of the next instruction to be executed where execution was suspended.</p> <p>Access-address indicates the address where the access that caused the break was made.</p>
---------------	---

Break at address by guarded access access-type at access-address
--

[Explanation]	<p>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.</p> <p>There may be an error in the memory attribute or the program.</p> <p>Address indicates the address of the next instruction to be executed where execution was suspended.</p> <p>Access-type indicates the type of the access that caused the break.</p> <p>Access-address indicates the address where the access that caused the break was made.</p>
---------------	---

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

[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 by the SHOW STATUS command when a break is caused by step (OVER) execution.</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>
---------------	--

INDEX

The index follows on the next page.
This is listed in alphabetic order.

Index

Symbols

%B.....	241
%BIT	241
%CALL	234
%D	241
%DEBUGFUNCTION.....	253
%ENTRY	236
%ERRNUM	235
%EVAL.....	250
%EVENTNUM.....	251
%H	241
%LOADNUM	240
%RADIX	238
%S.....	241
%SCPADR	239
%STKTOP.....	237
%STRCAT.....	246
%STRCMP	244
%STRGET	242
%STRLEN.....	245
%STRSTR.....	243
%SYMLen	247
%TOSTR.....	249
%TOVAL	248
%TRIGGERNUM.....	252
%W.....	241

A

ASSEMBLE.....	188
---------------	-----

B

BATCH.....	206
BREAK.....	231

C

CALL	62
CANCEL ALIAS	212
CANCEL BREAK	76
CANCEL CODEEVENT	97
CANCEL COVERAGE	135
CANCEL DATABREAK.....	84
CANCEL DATAEVENT.....	104
CANCEL DATATRACEAREA.....	155
CANCEL EVENT	91
CANCEL INPORT	31
CANCEL INTERRUPT	38
CANCEL LOGGING.....	220
CANCEL MAP (type 1)	23
CANCEL MAP (type 2)	24
CANCEL OUTPORT	35
CANCEL RAMCHECK.....	167
CANCEL SEQUENCE	111
CANCEL TRACETRIGGER.....	161
CANCEL VARIABLE.....	215
CANCEL WATCH	51
CLEAR CALL	64
CLEAR COVERAGE.....	136
CLEAR PERFORMANCE	130
CLEAR TIMER.....	140
CLEAR TRACE.....	147
COMPARE.....	182
COPY VECTOR.....	39

D

DISABLE ALIGNMENTBREAK.....	117
DISABLE BREAK.....	78
DISABLE CODEEVENT	99
DISABLE DATABREAK.....	86
DISABLE DATAEVENT	106
DISABLE EVENT	93
DISABLE EXTERNAL MEMORY.....	47

DISABLE LOGGING	222
DISABLE MAP	22
DISABLE RAMCHECK	170
DISABLE SEQUENCE	113
DISABLE TRACE	149
DISABLE TRACETRIGGER	163
DISABLE VERIFYMODE	26
DISABLE WATCHDOG	43
DISASSEMBLE	189
DOWN	204
DUMP	185

E

ENABLE ALIGNMENTBREAK	116
ENABLE BREAK	77
ENABLE CODEEVENT	98
ENABLE DATABREAK	85
ENABLE DATAEVENT	105
ENABLE EVENT	92
ENABLE EXTERNAL MEMORY	46
ENABLE LOGGING	221
ENABLE MAP	21
ENABLE RAMCHECK	169
ENABLE SEQUENCE	112
ENABLE TRACE	148
ENABLE TRACETRIGGER	162
ENABLE VERIFYMODE	25
ENABLE WATCHDOG	42
ENTER	174
Error Message for Debuggers	272
EXAMINE	172
Execution Suspension Messages List	293
EXIT	4

F

FILL	183
------------	-----

G

GO	58
----------	----

I

IF	228
INITIALIZE	3

L

LIST	198
LOAD	192

M

Manager-Related Messages	256
MOVE	184

P

PRINTF	223
--------------	-----

Q

QUIT	207
------------	-----

R

REPEAT	229
RESET	5

S

SAVE	194
SEARCH MEMORY	179
SEARCH TRACE	150
SET ALIAS	210
SET BREAK (type 1)	69
SET BREAK (type 2)	71
SET BREAK (type3)	72
SET BREAKCONDITION (type 1)	119
SET BREAKCONDITION (type 2)	121
SET CODEEVENT	94
SET COVERAGE	131
SET DATABREAK (type 1)	79
SET DATABREAK (type 2)	81
SET DATAEVENT	100
SET DATATRACEAREA	152
SET EVENT	87
SET EXTERNAL MEMORY	45
SET FREQUENCY	52
SET INPORT	28
SET INTERRUPT	36
SET LOGGING	218
SET MAP (type 1)	17
SET MAP (type 2)	18
SET MEMORY	175
SET MODE	9
SET OUTPORT	32
SET OUTPUT	225
SET PATH	200
SET PERFORMANCE	127
SET RADIX	12
SET RAMCHECK	164
SET REGISTER	180
SET RUNMODE	6

INDEX

SET SEQUENCE (type 1)	107	SHOW OUTPUT	34
SET SEQUENCE (type 2)	109	SHOW OUTPUT	226
SET SOURCE	14	SHOW PATH	201
SET STEP	60	SHOW PERFORMANCE	128
SET TRACE (type 1)	141	SHOW RADIX	13
SET TRACE (type 2)	142	SHOW RAMCHECK	166
SET TRACETRIGGER	158	SHOW REALTIMEMEMORYAREA	55
SET TRIGGER	114	SHOW REGISTER	181
SET VARIABLE	213	SHOW RUNMODE	8
SET VECTOR	40	SHOW SCOPE	202
SET WATCH	49	SHOW SEQUENCE	110
SHOW ALIAS	211	SHOW SOURCE	15
SHOW ALIGNMENTBREAK	118	SHOW STATUS	65
SHOW BREAK	74	SHOW STEP	61
SHOW BREAKCONDITION	123	SHOW SYSTEM	16
SHOW CALLS	137	SHOW TIMER	138
SHOW CODEEVENT	96	SHOW TRACE (type 1)	144
SHOW COVERAGE	132	SHOW TRACE (type 2)	146
SHOW DATABREAK	83	SHOW TRACERIGGER	160
SHOW DATAEVENT	102	SHOW TRIGGER	115
SHOW DATATRACEAREA	154	SHOW VARIABLE	214
SHOW DETAILTRACE	156	SHOW VECTOR	41
SHOW EVENT	90	SHOW VERIFYMODE	27
SHOW EXTERNAL MEMORY	48	SHOW WATCHDOG	44
SHOW FREQUENCY	53	STEP	59
SHOW INPORT	30		
SHOW INTERRUPT	37	U	
SHOW LOGGING	219	UP	203
SHOW MAP (type 1)	19		
SHOW MAP (type 2)	20	W	
SHOW MEMORY	177	WHILE	230
SHOW MODE	11		

CM71-00330-4E

FUJITSU MICROELECTRONICS • CONTROLLER MANUAL
FR FAMILY
SOFTUNE™ WORKBENCH
COMMAND REFERENCE MANUAL
for V6

June 2008 the fourth edition

Published **FUJITSU MICROELECTRONICS LIMITED**
Edited Business & Media Promotion Dept.
