

# ID78K0R-QB Ver.3.61

Integrated Debugger

User's Manual: Development Tool (Operation)

RENESAS MCU  
78K0R Microcontroller

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
  - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
  - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

## General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

### 1. Handling of Unused Pins

Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

### 2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.  
In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

### 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

### 4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

### 5. Differences between Products

Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

- The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

## How to Use This Manual

**Target Readers** This manual is intended for user engineers who design and develop application systems of the 78K0R Microcontroller.

**Purpose** This manual is intended for users to understand the functions of the ID78K0R-QB in the organization below.

**Organization** This manual consists of the following chapters:

- OVERVIEW
- INSTALLATION
- STARTING AND TERMINATING
- ASSOCIATION WITH PM+
- DEBUG FUNCTION
- WINDOW REFERENCE
- COMMAND REFERENCE

**How to Use This Manual** It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, microcontrollers, C language, and assemblers.

To understand the functions of the 78K0R Microcontroller  
→ Refer to Hardware User's Manual for each product.

**Conventions**

Data significance:	Higher digits on the left and lower digits on the right
<b>Note:</b>	Footnote for item marked with <b>Note</b> in the text
<b>Caution:</b>	Information requiring particular attention
<b>Remark:</b>	Supplementary information
Numerical representation:	Binary ... XXXX or XXXXB
	Decimal ... XXXX
	Hexadecimal ... 0XXXXX
Prefix indicating the power of 2 (address space, memory capacity):	
	K (Kilo): $2^{10} = 1024$
	M (Mega): $2^{20} = 1024^2$
	G (Giga): $2^{30} = 1024^3$

**Related Documents**

Refer to the documents listed below when using this manual.

The related documents indicated in this publication may include preliminary versions.

However, preliminary versions are not marked as such.

**Documents related to development tools (User's Manuals)**

Document Name		Document No.
QB-78K0RIX3 In-Circuit Emulator		U19228E
QB-78K0RKX3 In-Circuit Emulator		U17866E
QB-78K0RKX3C In-Circuit Emulator		U19324E
QB-78K0RLX3 In-Circuit Emulator		U19336E
RA78K0R Ver. 1.20 Assembler Package	Operation	U18547E
	Language	U18546E
CC78K0R Ver. 2.00 C Compiler Package	Operation	U18549E
	Language	U18548E
ID78K0R-QB Ver. 3.61 Integrated Debugger	Development Tool (Operation)	This manual
PM+ Ver.6.30		U18416E

[MEMO]

[MEMO]

[MEMO]

# CONTENTS

CHAPTER 1 GENERAL ...	18
1.1 Features ...	19
1.2 System Configuration ...	21
1.3 Operating Environment ...	22
1.3.1 Hardware environment ...	22
1.3.2 Software environment ...	22
1.4 Cautions During Debugging ...	23
1.4.1 When performing source level debugging ...	23
1.4.2 Security ID and on-chip debug option byte <b>[MINICUBE2]</b> ...	23
CHAPTER 2 INSTALLATION ...	24
2.1 Installing ...	24
2.2 Uninstalling ...	24
CHAPTER 3 STARTING AND TERMINATING ...	25
3.1 Cautions Before Starting <b>[MINICUBE2]</b> ...	25
3.2 Startup Option And Argument Specification ...	25
3.2.1 Specification method ...	26
3.2.2 Specification format and options ...	26
3.3 Starting ...	28
3.4 Terminating ...	29
3.5 Error Messages At Start Up ...	30
3.5.1 When the IECUBE is connected ...	30
3.5.2 When the MINICUBE2 is connected ...	30
CHAPTER 4 ASSOCIATION WITH PM+ ...	32
4.1 Setting Build Mode ...	32
4.2 Registering Debugger To PM+ Project ...	32
4.2.1 Selecting debugger ...	32
4.3 To Start ID78K0R-QB From PM+ ...	33
4.3.1 Restoring debugging environment ...	33
4.4 Auto Load ...	34
4.4.1 Auto load by correcting source code ...	34
4.4.2 Auto load by starting debugger ...	35
CHAPTER 5 DEBUG FUNCTION ...	36
5.1 Setting Debugging Environment ...	37
5.1.1 Setting operating environment ...	37
5.1.2 Setting option ...	37
5.1.3 Setting mapping ...	37
5.2 Download Function, Upload Function ...	38
5.2.1 Download ...	38
5.2.2 Upload ...	39
5.3 Source Display, Disassemble Display Function ...	40
5.3.1 Source display ...	40
5.3.2 Disassemble display ...	40
5.3.3 Mixed display mode (Source window) ...	41
5.3.4 Convert symbol (symbol to address) ...	42
5.4 Break Function ...	43
5.4.1 Break types ...	43
5.4.2 Breakpoint setting ...	44
5.4.3 Setting breaks to variables ...	45
5.4.4 Hardware break and software break ...	45
5.4.5 Fail-safe break function <b>[IECUBE]</b> ...	46
5.5 Program Execution Function ...	47
5.6 Watch Function ...	49
5.6.1 Displaying and changing data values ...	49

5.6.2	Displaying and changing local variable values ...	50
5.6.3	Registering and deleting watch data ...	50
5.6.4	Changing watch data ...	50
5.6.5	Temporarily displaying and changing data values ...	51
5.6.6	Callout watch function ...	51
5.6.7	Stack trace display function ...	51
5.7	Memory Manipulation Function ...	52
5.7.1	Displaying and changing memory contents ...	52
5.7.2	Access monitor function <b>[IECUBE]</b> ...	52
5.7.3	Filling, copying, and comparing memory contents ...	53
5.7.4	Flash memory writing function <b>[MINICUBE2]</b> ...	53
5.8	Register Manipulation Function ...	54
5.8.1	Displaying and changing register contents ...	54
5.8.2	Displaying and changing SFR contents ...	54
5.8.3	Displaying and changing I/O port contents ...	55
5.9	Timer Function <b>[IECUBE]</b> ...	56
5.9.1	Timer event conditions ...	56
5.9.2	Run-Break event ...	57
5.10	Trace Function <b>[IECUBE]</b> ...	58
5.10.1	Trace memory ...	58
5.10.2	Checking trace data ...	58
5.10.3	Mixed display mode (Trace window) ...	59
5.10.4	Tracer Operation ...	59
5.10.5	Setting conditional trace ...	61
5.11	Coverage Measurement Function <b>[IECUBE]</b> ...	62
5.11.1	Coverage measurement result display ...	62
5.11.2	Coverage measurement range ...	63
5.11.3	Display of locations for which coverage measurement is executed ...	63
5.12	Event Function ...	65
5.12.1	Using event function ...	65
5.12.2	Creating events ...	66
5.12.3	Setting event conditions ...	66
5.12.4	Number of enabled events for each event condition ...	67
5.12.5	Managing events ...	68
5.13	Snapshot Function <b>[IECUBE]</b> ...	70
5.13.1	Snapshot event conditions ...	70
5.13.2	Snap data ...	70
5.14	RRM Function ...	71
5.14.1	Real-time RAM monitor function <b>[IECUBE]</b> ...	71
5.14.2	Pseudo real-time RAM monitor function (Break When Readout) ...	71
5.15	DMM Function ...	73
5.15.1	Event DMM condition <b>[IECUBE]</b> ...	73
5.16	Load/Save Function ...	75
5.16.1	Debugging environment (project file) ...	75
5.16.2	Window display information (view file) ...	76
5.16.3	Window setting information (setting file) ...	77
5.17	Functions Common to Each Window ...	78
5.17.1	Active status and static status ...	78
5.17.2	Jump function ...	79
5.17.3	Trace result with linking window <b>[IECUBE]</b> ...	80
5.17.4	Drag & drop function ...	81
5.17.5	Cautions ...	83
	Main window ...	84
CHAPTER 6	WINDOW REFERENCE ...	96
6.1	Window List ...	96
6.2	Explanation of Windows ...	98
	Configuration dialog box ...	100
	Extended Option dialog box ...	107
	Fail-safe Break dialog box ...	112
	Flash Option dialog box ...	114
	DataFlash Option dialog box ...	118
	Debugger Option dialog box ...	120

Pseudo Emulation dialog box ... 127  
 Project File Save dialog box ... 128  
 Project File Load dialog box ... 129  
 Download dialog box ... 130  
 Upload dialog box ... 133  
 Load Module List dialog box ... 135  
 Source window ... 137  
 Source Search dialog box ... 143  
 Source Text Move dialog box ... 145  
 Assemble window ... 147  
 Assemble Search dialog box ... 153  
 Address Move dialog box ... 155  
 Symbol To Address dialog box ... 156  
 List window ... 158  
 Watch window ... 161  
 Quick Watch dialog box ... 167  
 Add Watch dialog box ... 169  
 Change Watch dialog box ... 172  
 Local Variable window ... 174  
 Stack window ... 177  
 Memory window ... 180  
 Memory Search dialog box ... 184  
 Memory Fill dialog box ... 186  
 Memory Copy dialog box ... 187  
 Memory Compare dialog box ... 188  
 Memory Compare Result dialog box ... 189  
 DMM dialog box ... 190  
 Register window ... 192  
 SFR window ... 195  
 SFR Select dialog box ... 199  
 Add I/O Port dialog box ... 201  
 Timer dialog box ... 203  
 Trace View window ... 207  
 Trace Search dialog box ... 214  
 Timer Result dialog box ... 218  
 Trace Data Select dialog box ... 220  
 Trace Move dialog box ... 222  
 Trace dialog box ... 224  
 Code Coverage window ... 227  
 Delay Count dialog box ... 230  
 Coverage-Color dialog box ... 232  
 Software Break manager ... 234  
 Event manager ... 236  
 Event Link dialog box ... 241  
 Break dialog box ... 244  
 Snap Shot dialog box ... 248  
 Event DMM dialog box ... 253  
 Event dialog box ... 257  
 View File Save dialog box ... 263  
 View File Load dialog box ... 265  
 Environment Setting File Save dialog box ... 267  
 Reset Debugger dialog box ... 268  
 Exit Debugger dialog box ... 269  
 About dialog box ... 270  
 Console window ... 271  
 Environment Setting File Load dialog box ... 272  
 Browse dialog box ... 273

## CHAPTER 7 COMMAND REFERENCE ... 275

7.1 Command Line Rules ... 275  
 7.2 List of Command ... 276  
 7.3 List of Aliases ... 278  
 7.4 List of Variables ... 278

- 7.5 List of Packages ... 279
- 7.6 Key Bind ... 279
- 7.7 Expansion Window ... 279
  - 7.7.1 Samples (Calculator Script) ... 280
- 7.8 Callback Procedure ... 281
- 7.9 Hook Procedure ... 282
- 7.10 Related Files ... 283
- 7.11 Cautions ... 283
- 7.12 Explanation of Commands ... 284
  - address ... 285
  - assemble ... 286
  - batch ... 287
  - breakpoint ... 288
  - dbgexit ... 290
  - download ... 291
  - erase ... 292
  - extwin ... 293
  - finish ... 294
  - go ... 295
  - help ... 296
  - hook ... 297
  - ie ... 298
  - inspect ... 299
  - ioport ... 300
  - jump ... 301
  - map ... 302
  - mdi ... 303
  - memory ... 304
  - module ... 305
  - next ... 306
  - refresh ... 307
  - register ... 308
  - reset ... 309
  - run ... 310
  - step ... 311
  - stop ... 312
  - tkcon ... 313
  - upload ... 314
  - version ... 315
  - watch ... 316
  - where ... 317
  - wish ... 318
  - xcoverage ... 319
  - xtime ... 320
  - xtrace ... 321

## APPENDIX A EXPANSION WINDOW ... 322

- A.1 Overview ... 322
- A.2 Sample Window List ... 322
- A.3 Activation ... 322
- A.4 Explanation of Each Sample Window ... 322
  - List window ... 323
  - Grep window ... 324
  - Hook window ... 325
  - SymInspect window ... 327
  - OpenBreak window ... 328

## APPENDIX B CAUTIONS ... 329

## APPENDIX C INPUT CONVENTIONS ... 336

- C.1 Usable Character Set ... 336
- C.2 Symbols ... 337
- C.3 Numeric Values ... 338

- C.4 Expressions And Operators ... 339
- C.5 File Names ... 341
- C.6 Quantitative Upper Limit List ... 342

APPENDIX D KEY FUNCTION LIST ... 343

APPENDIX E MESSAGES ... 346

- E.1 Display Format ... 346
- E.2 Types of Messages ... 347
- E.3 Message Lists ... 347

APPENDIX F INDEX ... 370

# LIST OF FIGURES

Figure No.	Title	Page
1-1	ID78K0R-QB ...	18
1-2	Example of ID78K0R-QB System Configuration [ <b>IECUBE</b> ] ...	21
1-3	Example of ID78K0R-QB System Configuration [ <b>MINICUBE2</b> ] ...	21
3-1	Startup Option (Example) ...	26
3-2	Configuration Dialog Box ...	28
3-3	Main Window ...	29
3-4	Exit Debugger Dialog Box ...	29
5-1	Breakpoint Setting ...	44
5-2	Setting Break to Variable ...	45
5-3	Management of Software Breaks ...	46
5-4	Fail-safe Break Setting ...	46
5-5	Execution Button ...	47
5-6	[Run] Menu ...	47
5-7	Watch Window ...	49
5-8	Specification of the Display Format (Debugger Option Dialog Box) ...	49
5-9	Local Variable Window ...	50
5-10	Change Watch Dialog Box ...	50
5-11	Quick Watch Dialog Box ...	51
5-12	Stack Window ...	51
5-13	Access Monitor Function (Memory Window) ...	53
5-14	Absolute Name/Function Name Switching ...	54
5-15	Register I/O Port ...	55
5-16	Sets and Displays Timer Event (Timer Dialog Box) ...	56
5-17	Checking Trace Data ...	59
5-18	Coverage Measurement Result Display ...	62
5-19	View of Locations for Which Coverage Measurement Executed ...	64
5-20	Setting of Various Event Conditions ...	66
5-21	Managing Events (Event Manager) ...	68
5-22	Snap Shot Dialog Box ...	70
5-23	Specification of Interval for Sampling with Real-time RAM Monitor Function ...	71
5-24	Specification of Pseudo Real-time RAM Monitor Function [ <b>IECUBE</b> ] ...	72
5-25	Specification of Pseudo Real-time RAM Monitor Function [ <b>MINICUBE2</b> ] ...	72
5-26	Modifying Memory Contents (DMM Dialog Box) ...	73
5-27	Event DMM Dialog Box ...	74
5-28	Main Window ...	84
5-29	Toolbar (Picture Only) ...	93
5-30	Toolbar (Picture and Text) ...	93
5-31	Statusbar ...	93
6-1	Configuration Dialog Box ...	100
6-2	Extended Option Dialog Box ...	107
6-3	Fail-safe Break Dialog Box ...	112
6-4	Flash Option Dialog Box ...	114
6-5	DataFlash Option Dialog Box ...	118
6-6	Debugger Option Dialog Box ...	120
6-7	Add Source path Dialog Box ...	121
6-8	Font Dialog Box ...	123
6-9	Pseudo Emulation Dialog Box ...	127
6-10	Project File Save Dialog Box ...	128
6-11	Project File Load Dialog Box ...	129
6-12	Download Dialog Box ...	130
6-13	Upload Dialog Box ...	133
6-14	Load Module List Dialog Box ...	135
6-15	Source Window ...	137
6-16	Source Search Dialog Box ...	143
6-17	Source Text Move Dialog Box ...	145

6-18	Assemble Window ...	147
6-19	Example of Assemble Window ...	149
6-20	Assemble Search Dialog Box ...	153
6-21	Address Move Dialog Box (Example: When Memory Window Is Open) ...	155
6-22	Symbol To Address Dialog Box ...	156
6-23	List Window ...	158
6-24	Watch Window ...	161
6-25	Example of Watch Window 1 ...	166
6-26	Example of Watch Window 2 ...	166
6-27	Quick Watch Dialog Box ...	167
6-28	Add Watch Dialog Box ...	169
6-29	Change Watch Dialog Box ...	172
6-30	Local Variable Window ...	174
6-31	Stack Window ...	177
6-32	Memory Window ...	180
6-33	Memory Search Dialog Box ...	184
6-34	Memory Fill Dialog Box ...	186
6-35	Memory Copy Dialog Box ...	187
6-36	Memory Compare Dialog Box ...	188
6-37	Memory Compare Result Dialog Box ...	189
6-38	DMM Dialog Box ...	190
6-39	Register Window ...	192
6-40	SFR Window ...	195
6-41	Example of SFR Window ...	197
6-42	SFR Select Dialog Box ...	199
6-43	Add I/O Port Dialog Box ...	201
6-44	Timer Dialog Box ...	203
6-45	Trace View Window ...	207
6-46	Trace Search Dialog Box ...	214
6-47	Timer Result Dialog Box ...	218
6-48	Trace Data Select Dialog Box ...	220
6-49	Trace Move Dialog Box ...	222
6-50	Trace Dialog Box ...	224
6-51	Code Coverage Window ...	227
6-52	Delay Count Dialog Box ...	230
6-53	Coverage-Color dialog box ...	232
6-54	Software Break Manager ...	234
6-55	Event Manager (In Detailed Display Mode) ...	236
6-56	Event Link Dialog Box ...	241
6-57	Break Dialog Box ...	244
6-58	Snap Shot Dialog Box (When "Memory" Is Selected) ...	248
6-59	Event DMM Dialog Box (When "Memory" Is Selected) ...	253
6-60	Event Dialog Box ...	257
6-61	View File Save Dialog Box ...	263
6-62	View File Load Dialog Box ...	265
6-63	Environment Setting File Save Dialog Box ...	267
6-64	Reset Debugger Dialog Box ...	268
6-65	Exit Debugger Dialog Box ...	269
6-66	About Dialog Box ...	270
6-67	Console Window ...	271
6-68	Environment Setting File Load Dialog Box ...	272
6-69	Browse Dialog Box ...	273
7-1	Execution Screen ...	280
A-1	List Window ...	323
A-2	Grep Window ...	324
A-3	Hook Window ...	325
A-4	SymInspect Window ...	327
A-5	OpenBreak Window ...	328
E-1	Error/Warning Dialog Box ...	346

# LIST OF TABLES

Table No.	Title	Page
2-1	Install...	24
3-1	Startup Options...	27
3-2	Error Message Output Pattern <b>[IECUBE]</b> ...	30
5-1	Debug Function List (Flow of Debugging Operations)...	36
5-2	Type of File That Can Be Downloaded...	38
5-3	Type of File That Can Be Uploaded...	39
5-4	File Type Can Be Displayed...	40
5-5	Specifying Symbols...	42
5-6	Break Types...	43
5-7	The Number of Valid Software Break...	46
5-8	Type of Execution...	48
5-9	Absolute Name to Function Name Correspondence...	54
5-10	Trace Memory Size...	58
5-11	Type of Trace Modes...	60
5-12	Types of Tracer Control Mode...	60
5-13	Types of Conditional Trace...	61
5-14	Code Coverage Measurement Range...	63
5-15	Format of View of Locations for Which Coverage Measurement Executed...	63
5-16	Various Event Conditions...	65
5-17	Number of Enabled Events for Each Event Condition...	68
5-18	Event Icon...	68
5-19	Areas for Which Sampling Can Be Performed with Real-time RAM Monitor Function...	71
5-20	Contents Saved to Project File...	75
5-21	Type of View Files...	76
5-22	Type of Setting Files...	77
5-23	Details of Jump Source Address...	79
5-24	Details of Drag & Drop Function (Line/Address)...	81
5-25	Details of Drag & Drop Function (Character String)...	82
5-26	CPU Status...	94
5-27	IE Status...	94
5-28	Break Cause...	95
6-1	Window List...	96
6-2	Relationship Between Division Ratio and Maximum Measurable Time (Time Tag Count)...	110
6-3	Event Setting Status (Event Mark)...	138
6-4	Watch Window Input Format...	170
6-5	How Variable Is Handled When Scope Is Specified...	170
6-6	Measurable Values...	205
6-7	Reset Causes During Trace Operation...	211
6-8	Break Causes When Tracer Is Stopped (Status)...	211
6-9	Settable Range of Address Condition (Trace)...	216
6-10	Frame Number Specification Format...	223
6-11	Number of Events Settable...	225
6-12	Separator for Displaying Event Details...	237
6-13	The Number of Event Conditions in Event Link Dialog Box...	242
6-14	Number of Events Settable in Condition Setting Area...	245
6-15	Settable Range of Address Condition (Event)...	259
7-1	Debugger Control Command List...	276
7-2	List of Console/Tcl Commands...	277
7-3	Contents of File aliases.tcl...	278
7-4	List of Variables...	278
7-5	List of Packages...	279
7-6	Message ID...	281
7-7	List of Related Files...	283
A-1	List of Expansion Window (Sample)...	322
C-1	List of Character Set...	336

C-2	List of Special Characters...	336
C-3	Input Format of Numeric Values...	338
C-4	List of Operators...	339
C-5	Operator Priority...	340
C-6	Range of Radixes...	341
C-7	Quantitative upper limit list...	342
D-1	Key Function List...	343
E-1	Types of Messages...	347

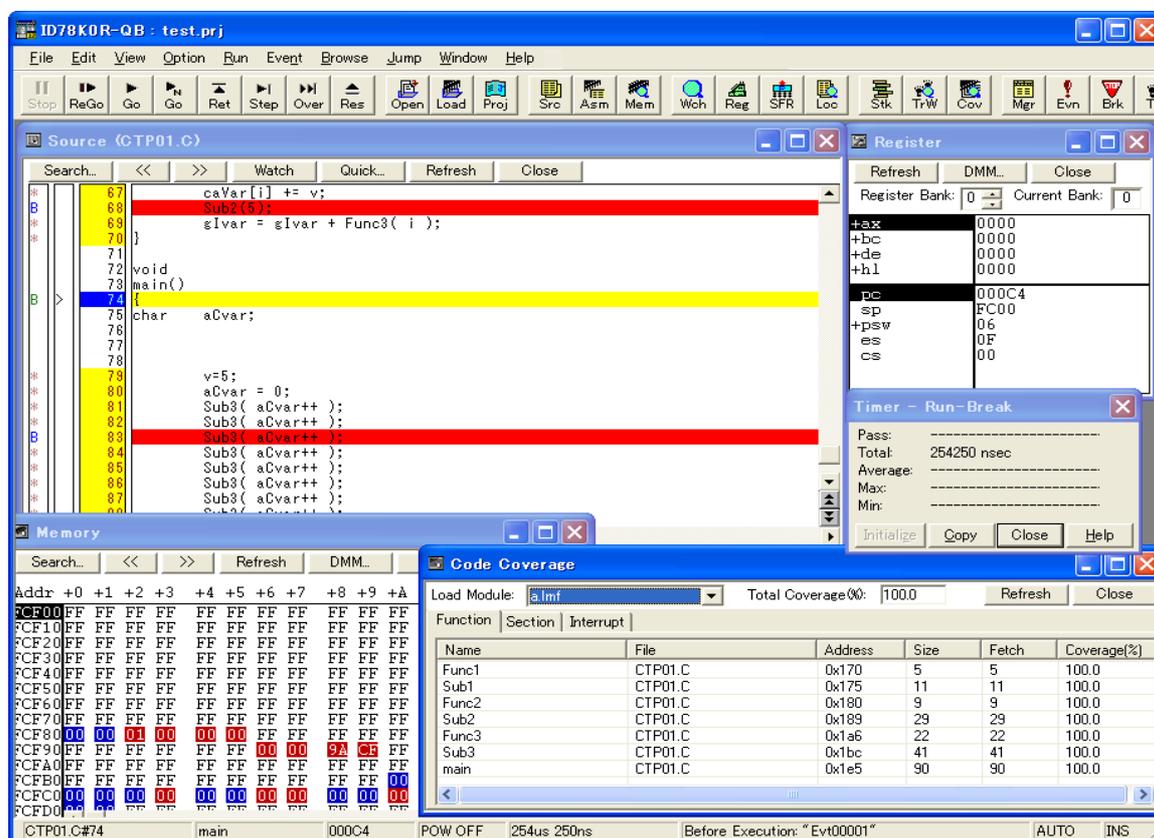
# CHAPTER 1 GENERAL

The Integrated Debugger ID78K0R-QB for the 78K0R microcontrollers (hereafter referred to as the ID78K0R-QB) is a software tool developed for Renesas Electronics 78K0R of microcontrollers for embedded control. This software tool is intended to enable efficient debugging of user programs.

The functions that can be implemented with the ID78K0R-QB vary depending on the emulator connected. (descriptions in parentheses show the symbols used to indicate each device in this manual).

- When the IECUBE is connected ( [IECUBE] )
- When the MINICUBE2 is connected ( [MINICUBE2] )

Figure 1-1 ID78K0R-QB



This chapter explains the following items regarding the ID78K0R-QB.

- Features
- System Configuration
- Operating Environment
- Cautions During Debugging

## 1.1 Features

The ID78K0R-QB has the following features:

### (1) Using function of in-circuit emulator

By using the event setting function of an in-circuit emulator, break events can be set, the user program can be traced, and time can be measured, and so on. (Refer to ["5.12 Event Function"](#).)

### (2) Source debugging

A C source program and source program in assembly language can be debugged.

### (3) Automatic display updating function when execution is stopped

If execution of the user program is stopped, the values of the window displayed on the screen are automatically updated.

### (4) Saving/restoring debugging environment

The debugging environment including information on setting of breakpoints and events, downloading files, and display status and position of windows can be saved to a file (project file).

By loading this project file, the debugging environment can be restored. (Refer to ["5.16 Load/Save Function"](#).)

### (5) Security function [MINICUBE2]

The ID code stored in the internal flash memory of a product with a security unit can be authenticated. (Refer to ["Configuration dialog box"](#), ["\(7\) ID Code \[MINICUBE2\]"](#).)

### (6) Function expansion through Tcl

The batch processing and hook processing, and the creation of original user custom windows are possible using the command line with Tcl/Tk (Tool Command Language). (Refer to ["CHAPTER 7 COMMAND REFERENCE"](#), ["APPENDIX A EXPANSION WINDOW"](#).)

### (7) Timer function [IECUBE]

Other than during Run-Break, the maximum time, minimum time, pass count, and average time are displayed. (Refer to ["5.9 Timer Function \[IECUBE\]"](#).)

Measured time display during user program execution and timeout break are supported.

### (8) Fail-safe break support [IECUBE]

In addition to the guarded areas and SFR areas, fail-safe break is also supported for the guarded areas of internal ROM/internal RAM. (Refer to ["5.4.5 Fail-safe break function \[IECUBE\]"](#).)

### (9) Command function

Script file specification is possible at ID startup. (Refer to ["3.2 Startup Option And Argument Specification"](#).)

Testing can be done with 1 click by specifying a project file at the same time as the script file.

The Tcl/Tk core was updated to the latest version 8.4.

### (10) Settings during program execution [IECUBE]

Timer event conditions and trace event conditions can now be set during user program execution.

**(11) Support of both IECUBE and MINICUBE2**

Two emulators (IECUBE, MINICUBE2) can be connected with 1 debugger (ID78K0R-QB) (Refer to "[1.2 System Configuration](#)".) USB2.0 is supported.

**(12) Snapshot function [IECUBE]**

The contents of the register memory and SFRs in the user program execution process can now be saved as snap data in the trace memory. (Refer to "[5.13 Snapshot Function \[IECUBE\]](#)".)

**(13) Support of code coverage measurement [IECUBE]**

Code coverage measurement (C0 coverage) can now be performed.

The code coverage can now be displayed in the Code Coverage window, and sections for which coverage measurement is executed can now be displayed in the Source window and Assemble window. (Refer to "[5.11 Coverage Measurement Function \[IECUBE\]](#)".)

**(14) DMM function**

DMM (Dynamic Memory Modification) is now possible with memory, registers, or SFRs specified.

Real-time writing to the memory can be performed with the DMM function during user program execution. (Refer to "[5.15 DMM Function](#)".)

**(15) Support of flash self programming error emulation [IECUBE]**

Flash self programming can now be debugged. (Refer to "[Flash Option dialog box](#)".)

**(16) Support of DataFlash error emulation [IECUBE]**

DataFlash can now be debugged. (Refer to "[DataFlash Option dialog box](#)".)

## 1.2 System Configuration

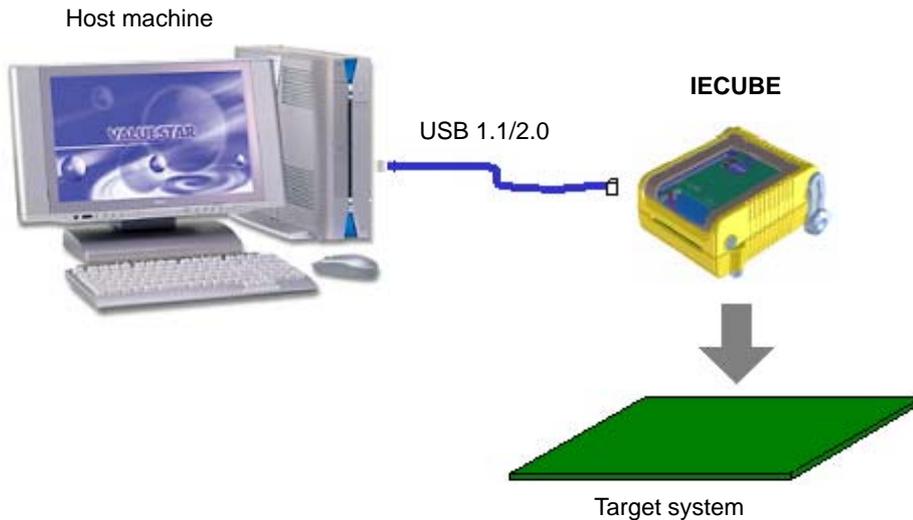
The ID78K0R-QB can be connected to the following two types of emulators.

User programs developed for the 78K0R microcontrollers and a pleasant debugging environment for target systems are provided.

### (1) IECUBE

The IECUBE can be manipulated from the ID78K0R-QB by connected it to the ID78K0R-QB via a USB cable.

Figure 1-2 Example of ID78K0R-QB System Configuration [IECUBE]

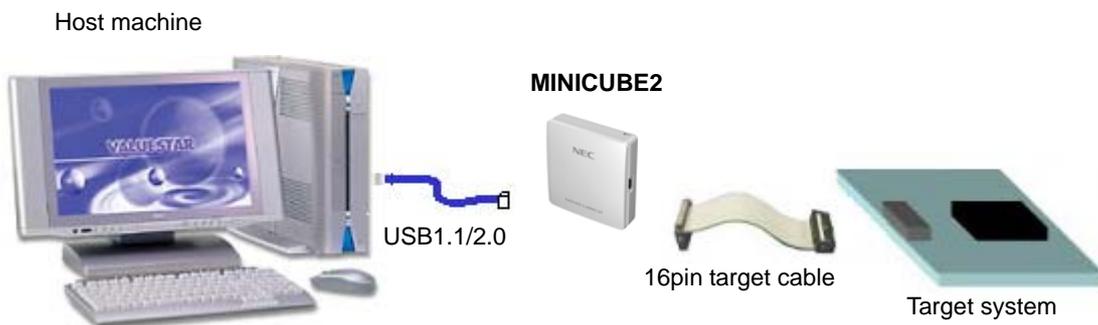


### (2) MINICUBE2

The MINICUBE2 is operated via the ID78K0R-QB when it is connected to the host machine with the USB cable.

The MINICUBE2 can provide the debug function when it is connected to a microcontroller with the on-chip debug function.

Figure 1-3 Example of ID78K0R-QB System Configuration [MINICUBE2]



## 1.3 Operating Environment

This section explains the following items regarding the operating environment.

- [Hardware environment](#)
- [Software environment](#)

### 1.3.1 Hardware environment

(1) **Host machine** (The machine by which the target OS operates)

CPU	Pentium II™ 400MHz or above
Main memory	256MB or above

(2) **In-circuit emulator**

- IECUBE
- MINICUBE2

### 1.3.2 Software environment

(1) **OS (any of the following)**

- Windows® 2000
- Windows XP (Home Edition, Professional)
- Windows Vista (Business Edition, Enterprise Edition)

**Caution:**

- Regardless of which of the OS above is used, we recommend that the latest Service Pack is installed.
- Only 32-bit OS is supported.

(2) **Device file (Individual acquisition)**

- The device file of the target device to be used.

**Remark:** This file is available from the following Web site of Renesas Electronics (Version-up Service).  
<http://www2.renesas.com/micro/en/ods/>

(3) **Supported tools (manufactured by Renesas Electronics)**

- Assembler package
- C compiler package
- Project manager

**Caution:** For versions of tools, refer to Operating Precautions.

## 1.4 Cautions During Debugging

The cautions to be observed during debugging are described below.

- [When performing source level debugging](#)
- [Security ID and on-chip debug option byte \[MINICUBE2\]](#)

### 1.4.1 When performing source level debugging

The object file for which source level debugging is performed must include symbol information or other information for debugging (debugging information).

Therefore, perform the following processing during source file compiling.

<b>When using the PM+</b>	Specify [Debug Build] when the Build mode is selected.
<b>When not using the PM+</b>	Do not add the -ng option.

### 1.4.2 Security ID and on-chip debug option byte [MINICUBE2]

The security ID, and information which enables the on-chip debug operation with the on-chip debug option byte, must be included in the object file that is used when connecting MINICUBE2.

For the settings of the security ID and on-chip debug option byte, refer to the RA78K0R Assembler Package Operation User's Manual.

For details on the security ID and on-chip debug option byte, refer to the user's manual of MINICUBE2 or each device.

The security ID (ID code) can also be set in the [Configuration dialog box](#) of the ID78K0R-QB.

# CHAPTER 2 INSTALLATION

This chapter explains the following items about installation of ID78K0R-QB:

- [Installing](#)
- [Uninstalling](#)

## 2.1 Installing

The following items must be installed, when the ID78K0R-QB is used.

Table 2-1 Install

Item	Procedure
ID78K0R-QB Disk	Install the contents of this disk according to the automatically executed installer.
Device file	Install this file according to the "DFINST" dedicated startup installer by selecting [start] menu -> [All Programs] -> [NEC Electronics Tools] -> [DeviceFile Installer].

## 2.2 Uninstalling

Perform uninstallation using [Add/Remove Programs] on the Control Panel.

# CHAPTER 3 STARTING AND TERMINATING

This chapter explains the following items related to the starting and terminating the ID78K0R-QB:

- [Cautions Before Starting \[MINICUBE2\]](#)
- [Startup Option And Argument Specification](#)
- [Starting](#)
- [Terminating](#)
- [Error Messages At Start Up](#)

## 3.1 Cautions Before Starting [MINICUBE2]

When the MINICUBE2 is connected, start the following check tools before the starting the ID78K0R-QB to check that the in-circuit emulator and the target system can be normally debugged.

- OCD Checker

**Caution:** For the connection between the in-circuit emulator and the target system and the power application sequence, refer to the MINICUBE2 User's Manual.  
Incorrect connection may damage the in-circuit emulator and the target system.

## 3.2 Startup Option And Argument Specification

The procedure for specifying the startup options and arguments for the ID78K0R-QB is described below.

By specifying the startup options and arguments, it is possible to specify the script file at startup and the project file.

**Remark:** When starting up the ID78K0R-QB from PM+, the startup option and argument settings are performed in [Debugger Settings...] in the [Tool] menu of PM+. (Refer to "[CHAPTER 4 ASSOCIATION WITH PM+](#)".) The debugger startup option can be set to the option column.

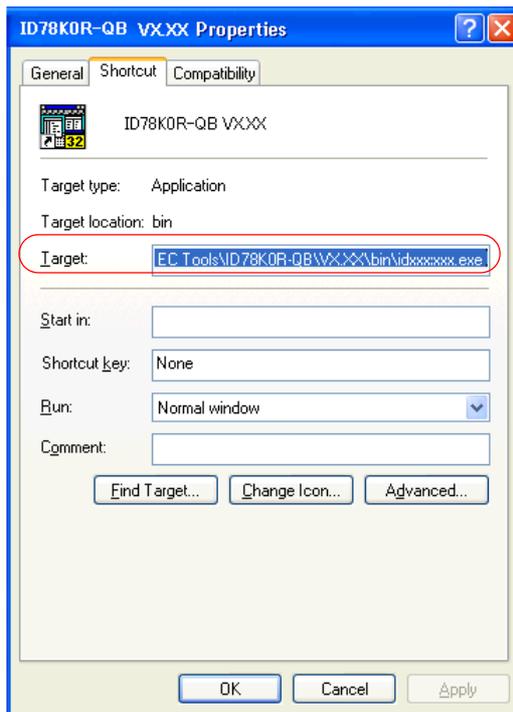
### 3.2.1 Specification method

- 1) Create the ID78K0R-QB shortcut on the desktop.

The ID78K0R-QB execution file is located in the bin folder in the folder to which the installation was performed.

- 2) Open the properties of the created shortcut and after the execution file name displayed in [Target:], specify the option and argument. (Refer to "3.2.2 Specification format and options".)

Figure 3-1 Startup Option (Example)



### 3.2.2 Specification format and options

#### (1) Specification format

```
idk0r32g.exe ?options?
idk0r32g.exe ?options? project
```

Each option and argument is separated by a space. The case is distinguished in the character string.

Arguments enclosed between '?' can be omitted.

When a project file is specified, that project file is read at startup.

However, during PM+ startup, the project file specification is ignored.

When there are spaces in the file names and paths, specify the project file names and script file names enclosed in double quotation marks (" "). (Refer to " [Example3](#) Specification when there are spaces in the path".)

**(2) Specification options**

The following options can be specified.

Table 3-1 Startup Options

Options	Meaning
/sc	Change background color of window to system color.
/RPC [IECUBE]	The program name, source file name, line number and PC value are updated on the status bar while the debugger is running.
/script: <i>script file name</i>	Specify the script file to be executed at startup.

**(3) Specification example****Example1) Specification of script file only**

```
idk0r32g.exe /script:c:\work\script.tcl
```

**Example2) Specification of script file and project file**

```
idk0r32g.exe /script:c:\work\script.tcl c:\work\project.prj
```

**Example3) Specification when there are spaces in the path**

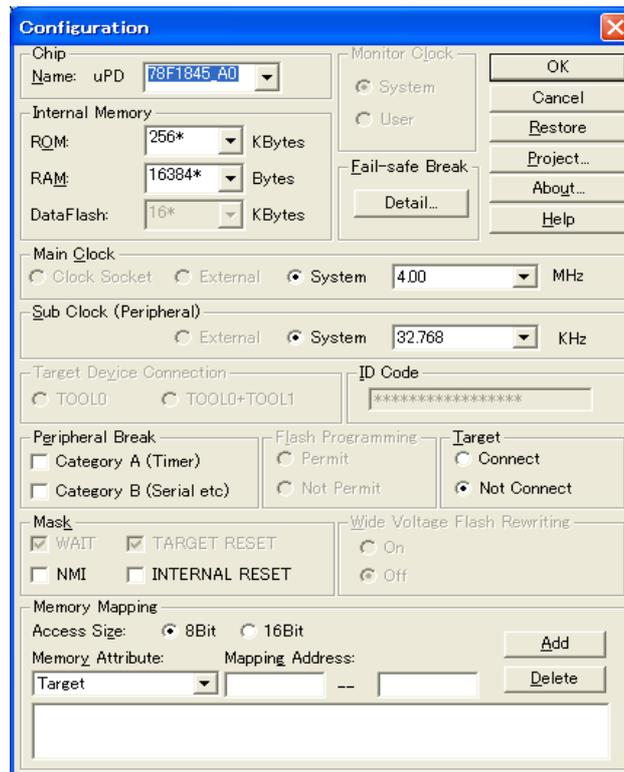
```
idk0r32g.exe /script:"c:\work folder\script.tcl" "c:\work folder\project.prj"
```

### 3.3 Starting

- 1) Start the ID78K0R-QB from the [Start] menu of PM+ or by clicking the shortcut created on the desktop.  
Refer to "4.3 To Start ID78K0R-QB From PM+" when starting from PM+.  
Start the ID78K0R-QB, the [Configuration dialog box](#) will be opened.

**Caution:** In this case, the Configuration dialog box should not be displayed, but an error message should be displayed, please cope with it with reference to "3.5 Error Messages At Start Up".

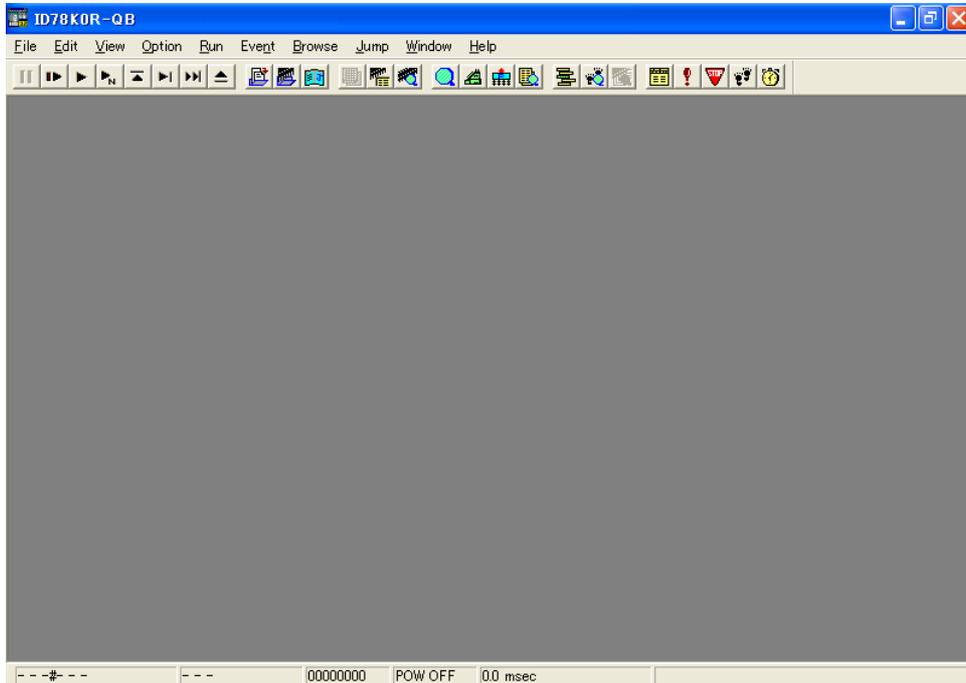
Figure 3-2 Configuration Dialog Box



- 2) Set the items related to the operating environment of the ID78K0R-QB in the [Configuration dialog box](#).  
After setting each item, click the [OK] button in the dialog box.

- 3) The [Main window](#) will be opened and the ID78K0R-QB can be operated. Mainly use this window for debugging.

Figure 3-3 Main Window



### 3.4 Terminating

- 1) Select [File] menu -> [Exit] on the [Main window](#). The following the [Exit Debugger dialog box](#) will be opened:  
(An execution stop confirmation message is displayed when stop operation is performed during program execution.)

Figure 3-4 Exit Debugger Dialog Box



To save the current debugging environment to a project file, click the [Yes] button. If the [No] button is clicked, all the windows are closed the ID78K0R-QB terminated.

## 3.5 Error Messages At Start Up

Error messages that may be output when the ID78K0R-QB starts up are listed below (by order of occurrence). When these messages are output, refer to "[APPENDIX E MESSAGES](#)".

### 3.5.1 When the IECUBE is connected

The pattern of the output error message differs as follows depending on the connection status with the target and the settings in the [Configuration dialog box](#).

Table 3-2 Error Message Output Pattern [IECUBE]

Error Message	[Target] Area in the Configuration dialog box		Target		Exchange Adapter		Target Power Supply	
	Connect	Not Connect	Connect	Not Connect	Used	Not Used	ON	OFF
Ff606: Please check connection with the target board, and power on it.	Checked	-	-	-	-	-	-	Checked
Wf607: Please check connection of the exchange adapter.	-	Checked	-	Checked	-	Checked	-	Checked
Ff608: Please disconnect the target board.	-	Checked	Checked	-	-	-	-	Checked
Ff609: Please power off the target board, and disconnect it.	-	Checked	-	-	-	-	Checked	-

### 3.5.2 When the MINICUBE2 is connected

F0100: Can not communicate with ICE. Please confirm the installation of the device driver for the PC interface board.
F03a0: Target is not turned on.
A0105: Failed in reading device file (d1xxx.78k).
F0ca2: This device file does not include the on-chip debug information.
F0ca3: Unsupported information is included in the on-chip debug information in the device file.
A01a0: No response from the CPU. Please confirm the signal of the CLOCK or RESET WAIT, HLDRQ and so on.
F0ca0: Error occurred inside debugger.

A0109: Can not communicate with ICE. Please terminate the debugger and check the power of ICE or the connection of cable then restart the debugger.

A01a6: Executor is running.

A010a: Cannot run debugger and a utility at the same time.

F0103: Data transfer to ICE is timed out. Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board.

F0c79: The on-chip debug function had been disabled in the device.

F0c48: Flash programming is not permitted by the debugger setting.

# CHAPTER 4 ASSOCIATION WITH PM+

The ID78K0R-QB can automatically perform a series of operations in development processes, such as creating source files -> compiling -> debugging -> correcting source files, in association with the PM+.

This chapter explains the following items related to association with the PM+.

For details of the PM+ functions, refer to the PM+ User's Manual.

- [Setting Build Mode](#)
- [Registering Debugger To PM+ Project](#)
- [To Start ID78K0R-QB From PM+](#)
- [Auto Load](#)

**Caution:** If a load module file is created by using the Windows command prompt, the function to associate the ID78K0R-QB with the PM+ cannot be used.

## 4.1 Setting Build Mode

To debug the load module file created by the PM+ on the ID78K0R-QB at the source level, build to output symbol information for debugging must be performed to create a load module file. This setting can be performed by selecting [Debug Build] on the PM+.

## 4.2 Registering Debugger To PM+ Project

The debugger to be used or the load module files to be downloaded can be specified for each project in the PM+.

### 4.2.1 Selecting debugger

The procedure for selecting the debugger is as follows:

The ID78K0R-QB is registered as the debugger of the active project. The ID78K0R-QB icon is displayed on the toolbar of the PM+.

#### (1) Creating a new workspace

- 1) Select [File] menu -> [New Workspace...] on the PM+.  
-> This opens the dialog box to create a new workspace using the wizard format.
- 2) Creating the necessary settings for the workspace with the wizard, the [Select Debugger] dialog box will be opened. Specify ID78K0R-QB in this dialog box.

For details of the setting, refer to the User's manual.

## (2) Using an existing workspace

- 1) Select [Tool] menu -> [Debugger Settings...] on the PM+.  
-> The [Debugger Settings] dialog box will be opened.
- 2) Specify ID78K0R-QB and click the [OK] button in this dialog box. For details of the setting, refer to the User's manual.

## 4.3 To Start ID78K0R-QB From PM+

The ID78K0R-QB can be started from the PM+ as follows:

- Click the ID78K0R-QB starting button on the toolbar of the PM+.
- Select the [Build] menu -> [Debug] on the PM+.
- Select the [Build] menu -> [Build and Debug] on the PM+.
- Select the [Build] menu -> [Rebuild and Debug] on the PM+.

If the debugging environment of the ID78K0R-QB is saved to a project file currently being used by the PM+, it will be started in the debugging environment saved in the project file.

If the debugging environment of the ID78K0R-QB is not saved to a project file being used by the PM+, the [Configuration dialog box](#) is opened. At this time, the device type (chip name) cannot be changed.

**Caution:** With the PM+, if too many source files are registered in a project, the number of files may exceed the upper limit of the source path length that can be registered to the ID78K0R-QB, the source files consequently may not be displayed automatically.

For details on the source path length, refer to "(1) Source Path" in the [Debugger Option dialog box](#).

### 4.3.1 Restoring debugging environment

The previous debugging environment can be restored by the following procedure when the ID78K0R-QB is started from the PM+:

- 1) Create a new workspace (project file: e.g., sample.prj) on the PM+<sup>Note</sup>.
- 2) Start the ID78K0R-QB from the PM+. Because a new project file is created, set items other than the device type (chip name) in the [Configuration dialog box](#) in the same manner as when only the ID78K0R-QB is started.
- 3) Download the load module file to be debugged with the [Download dialog box](#) of the ID78K0R-QB.
- 4) Debug the load module file on the ID78K0R-QB.
- 5) Click the [Yes] button on the [Exit Debugger dialog box](#) when the ID78K0R-QB is terminated.  
-> The debugging environment will be saved to the project file (sample.prj) for the PM+ when the ID78K0R-QB is terminated (the debug environment can also be saved to the sample.prj file by overwriting the project file at times other than the completion of ID78K0R-QB debugging).
- 6) When the ID78K0R-QB is next started up after the sample.prj file is read by PM+, the debug environment at the point when the project file was saved is automatically restored.

**Note:** In the ID78K0R-QB and PM+, the environment information is saved to a project file and referenced. The extension of the project file that can be used by the ID78K0R-QB and PM+ is "prj". For the information that is saved or restored by the project file, refer to the "User's manual" of each product.

## 4.4 Auto Load

If a bug is found while the load module file is being debugged by the ID78K0R-QB, correct the source file using the following procedure. Compiling and re-downloading the file can be automatically executed. (Refer to "[4.4.1 Auto load by correcting source code](#)".)

The load module is downloaded again to the ID78K0R-QB by compiling and linking the file on the PM+ with the activated ID78K0R-QB. (Refer to "[4.4.2 Auto load by starting debugger](#)".)

**Caution:** This processing cannot be performed if it is selected that the standard editor (idea-L) is used with the PM+.

### 4.4.1 Auto load by correcting source code

Correct the source file for auto load as follows:

- 1) Open the source file to be corrected in the [Source window](#). Select [File] menu -> [Open] and specify the file to be corrected on the ID78K0R-QB (if the file is already open in the Source window, that window is displayed in the forefront).  
-> The specified file will be opened in the Source window.
- 2) Select [Edit] menu -> [Edit Source] on the ID78K0R-QB.  
-> An editor will be opened and the specified source file will be read.
- 3) Correct the source file on the editor.
- 4) Terminate the editor.

**Caution:** The CPU reset is not performed when the load module file is automatically downloaded. The debug window that was opened when the editor was called, and each event setting will be restored. If the previously used line or symbol has been deleted as a result of correcting the source file, the following happens:

- A variable that was displayed is dimmed.
- The event mark of an event condition is displayed in [yellow](#).
- A software break point may be deleted.

- 5) Select [Build] menu -> [Build and Debug], or [Build] menu -> [Rebuild and Debug] on the PM+.

## 4.4.2 Auto load by starting debugger

If the following operation is performed on the PM+ with the ID78K0R-QB started, the load module will be automatically downloaded to the ID78K0R-QB.

- Selecting the [Build] menu -> [Build and Debug] on the PM+.
- Selecting the [Build] menu -> [Rebuild and Debug] on the PM+.

**Remark:** Specify whether to use a CPU reset after downloading from [Debugger Settings...] on the [Tool] menu of PM+ (a CPU reset is performed by default).

# CHAPTER 5 DEBUG FUNCTION

This chapter explains about debug function of ID78K0R-QB.

Table 5-1 Debug Function List (Flow of Debugging Operations)

Item	Refer To
To set the debugging environment	<a href="#">5.1 Setting Debugging Environment</a>
To download the load module	<a href="#">5.2 Download Function, Upload Function</a>
To display the source file and the disassemble result	<a href="#">5.3 Source Display, Disassemble Display Function</a>
To set a break point	<a href="#">5.4 Break Function</a>
To execute the user program	<a href="#">5.5 Program Execution Function</a>
To check the variable value	<a href="#">5.6 Watch Function</a>
To check and edit the memory contents	<a href="#">5.7 Memory Manipulation Function</a>
To check and change the register variable	<a href="#">5.8 Register Manipulation Function</a>
To check the execution time	<a href="#">5.9 Timer Function [IECUBE]</a>
To check the trace data	<a href="#">5.10 Trace Function [IECUBE]</a>
To check the coverage measurement results	<a href="#">5.11 Coverage Measurement Function [IECUBE]</a>
To manage the events	<a href="#">5.12 Event Function</a>
Snapshot function	<a href="#">5.13 Snapshot Function [IECUBE]</a>
RRM sampling	<a href="#">5.14 RRM Function</a>
DMM function	<a href="#">5.15 DMM Function</a>
To save the debug environment and window status	<a href="#">5.16 Load/Save Function</a>
Jump function, linking window function	<a href="#">5.17 Functions Common to Each Window</a>

## 5.1 Setting Debugging Environment

This section explains the following items related to the setting debugging environment:

- [Setting operating environment](#)
- [Setting option](#)
- [Setting mapping](#)

### 5.1.1 Setting operating environment

The in-circuit emulator operating environment settings are performed in the [Configuration dialog box](#) that is automatically displayed when the ID78K0R-QB starts up.

If a project file already exists, the debugging environment can be restored by clicking the [Project...] button. (Refer to "[5.16.1 Debugging environment \(project file\)](#)".)

### 5.1.2 Setting option

Perform setting related to the debugger or in-circuit emulator in the following setting dialog boxes.

- [Configuration dialog box](#)
- [Extended Option dialog box](#)
- [Fail-safe Break dialog box](#)
- [Flash Option dialog box](#)
- [DataFlash Option dialog box](#)
- [Debugger Option dialog box](#)
- [Pseudo Emulation dialog box](#)

### 5.1.3 Setting mapping

The mapping settings are performed in the [Configuration dialog box](#).

## 5.2 Download Function, Upload Function

The ID78K0R-QB allows downloading and uploading of object files in the formats listed in the following table:

[Table 5-2](#), [Table 5-3](#).

This section explains the following items:

- [Download](#)
- [Upload](#)

**Remark:** The internal flash memory can be written and the load module can be downloaded. (Refer to "[5.7.4 Flash memory writing function \[MINICUBE2\]](#)".) **[MINICUBE2]**

### 5.2.1 Download

Object files are downloaded in the [Download dialog box](#).

The corresponding source text file ([Source window](#)) is displayed by downloading load module files with debug information.

**Remark:** Loaded files can be selected in the [Load Module List dialog box](#) that is opened by selecting [File] menu -> [Load Module].

Format of file that can be downloaded is as follows:

Table 5-2 Type of File That Can Be Downloaded

Format	Extension
Load module (XCOFF(.lnk, .lmf))	Load Module (*.lnk;*.lmf)
Intel Hex format (Standard and extension)	Hex Format (*.hex;*.hxb;*.hxf) (Includes the Hex format compatible with flash memory self-programming mode)
Motorola Hex format S type (S0, S2, S8)	
Extended Tektronix Hex format	
Binary data	Binary Data (*.bin)
Coverage result <b>[IECUBE]</b>	Coverage (*.cvb)

**Remark:** The format of a HEX file is automatically determined.

## 5.2.2 Upload

Uploading of memory contents, etc., is performed in the [Upload dialog box](#). The saving range can be set.

Format of file that can be uploaded is as follows:

Table 5-3 Type of File That Can Be Uploaded

Extension	Format
Intel Hex (*.hex)	Intel Hex format Extension (20-bit addresses)
Motorola Hex (*.hex)	Motorola Hex format S type (S0,S2,S8 -24 bit-address)
Tektro Hex (*.hex)	Extended Tektronix Hex format
Binary Data (*.bin)	Binary data
Coverage (*.cvb)	Coverage results <b>[IECUBE]</b>

**Remark:** One of the formats can be specified when saving a \*.hex file.

## 5.3 Source Display, Disassemble Display Function

Source file display is performed in the [Source window](#). Disassemble display and online assembly are performed in the [Assemble window](#).

This section explains the following items:

- [Source display](#)
- [Disassemble display](#)
- [Mixed display mode \(Source window\)](#)
- [Convert symbol \(symbol to address\)](#)

**Remark:** The locations for which coverage measurement is executed in the user program are displayed in the [Source window](#) and [Assemble window](#). (Refer to "5.11.3 Display of locations for which coverage measurement is executed".)

### 5.3.1 Source display

The corresponding text file is displayed in the [Source window](#) by downloading a load module file having debug information.

The display start position can be changed in the [Source Text Move dialog box](#) displayed by selecting [View] menu -> [Move...].

Specifications related to the tab size, display font, etc., and specification of the source path are made in the [Debugger Option dialog box](#). Specify a searching method in the [Source Search dialog box](#) opened by clicking the [Search...] button. The search result is highlighted in the [Source window](#).

Table 5-4 File Type Can Be Displayed

File Type (Extension)	Meaning
Source (*.c, *.s, *.asm)	Source file (The extension can be changed in the <a href="#">Debugger Option dialog box</a> .)
Text (*.txt)	Text file
All (*.*)	All files

### 5.3.2 Disassemble display

Disassemble display is performed in the [Assemble window](#).

The display start position can be changed in the [Address Move dialog box](#) opened by selecting [View] menu -> [Move...].

Offset display and register name display are specified in the [Debugger Option dialog box](#).

Specify a searching method in the [Assemble Search dialog box](#) opened by clicking the [Search...] button. The search result is highlighted in the [Assemble window](#).

### 5.3.3 Mixed display mode (Source window)

Programs can be disassembled and displayed combined with the source file by selecting [View] menu -> [Mix] in the [Source window](#). The contents displayed in the mixed display mode can be saved as a view file.

#### Normal display mode

```

58      /* Timer Set */
*      59      TUM1 = 0x200;
*      60      CE1 = 1;
*      61      time_over = 0;

```

In the normal display mode, general text files can be displayed as well as source files.

#### Mixed display mode

```

58      /* Timer Set */
*      59      TUM1 = 0x200;
*      00000394 20660002  movea 0x200, r0, r12
*      00000398 606740f2  st.h r12, TUM1
*      60      CE1 = 1;
*      0000039c c03f42f2  setl 0x7, TMC1
*      61      time_over = 0;
*      000003a0 440e0000  movhi 0x0, gp, r1
*      000003a4 61071184  st.w r0, -0x7bf0[r1]

```

If a program code corresponds to the line of the displayed source file, the disassembly line is displayed next to the source line. The label of the address, code data, and disassembled mnemonic are displayed (the display start position of the mnemonic is adjusted by the set value of the tab size).

**Caution:** The mixed display mode is valid only when the load module is downloaded and the symbol information is read, and the corresponding source file is displayed.

**Remark:** The disassemble section cannot be edited in the Source window even if the Source window is set to the mixed display mode. Edit the disassemble section in the Assemble window.

### 5.3.4 Convert symbol (symbol to address)

In the [Symbol To Address dialog box](#), can be displayed the address of the specified variable or function, or the value of the specified symbol.

Convert symbol is performed by selecting the character string to be converted in the [Source window](#) or [Assemble window](#), and then selecting context menu -> [Symbol...].

The Specification symbols is indicated below.

Table 5-5 Specifying Symbols

Conversion Target	Specification Method
Variable	var file#var (to specify a static variable with file name) file#func#var (to specify a static variable with file name and function name)
Function	func file#func (to specify a static function with file name)
Label	label file#label (to specify a local label with file name)
EQU symbol	equsym file#equsym (to specify a local EQU symbol with file name)
Bit symbol	bitsym file#bitsym(to specify a local bit symbol with file name)
Line number of source file	file#no prog\$file#no
I/O port name	portname
SFR name	sfrname
Register name	regname
PSW flag name	pswname

**Remark:** Separator "#"

"#" is used as a separator for file names, variables, function names, and line numbers. If a specified symbol is not found in the scope, all symbols (static variables, static functions, local labels) are searched.

## 5.4 Break Function

The break function is used to stop execution of the user program by the CPU and operation of the tracer.

This section explains the following items:

- [Break types](#)
- [Breakpoint setting](#)
- [Setting breaks to variables](#)
- [Hardware break and software break](#)
- [Fail-safe break function \[IECUBE\]](#)

### 5.4.1 Break types

The ID78K0R-QB has the following break functions.

Table 5-6 Break Types

Item	Contents
Hardware break <b>Note1</b> (Event detection break)	Function to stop user program execution upon detection of the set break event condition. -> Refer to " <a href="#">5.4.2 Breakpoint setting</a> ".
Software break <b>Note1</b>	Function to replace the instruction at the specified address software break instruction and stop the user program executed. (Refer to " <a href="#">5.4.4 Hardware break and software break</a> ".) -> Refer to " <a href="#">5.4.2 Breakpoint setting</a> ".
[Come Here] break <b>Note2</b> (Simple break)	Function to stop user program execution selected by selecting [Run] menu -> [Come Here] upon detection of address specified in the <a href="#">Source window</a> or <a href="#">Assemble window</a> .
Break on satisfaction of condition of step execution	Function to stop execution upon satisfaction of the stop condition of each command ([Step In], [Next Over], [Return Out], [Slowmotion]).
Forced break	Function to forcibly stop execution by selecting [Run] menu -> [Stop], or selecting the STOP button. It is valid for all the execution commands.
Fail-safe break <b>[IECUBE]</b>	Function to forcibly stop execution when the user program performs an illegal operation in relation to the memory or registers. (Refer to " <a href="#">5.4.5 Fail-safe break function [IECUBE]</a> ".) -> Refer to " <a href="#">Fail-safe Break dialog box</a> ".
Time-out break <b>[IECUBE]</b>	Function to stop user program execution when the measurement time exceeds the specified time-out time. (Refer to " <a href="#">5.9 Timer Function [IECUBE]</a> ".) -> Refer to " <a href="#">Timer dialog box</a> ".

**Note1:** This break is valid for [Go], [Go & GO], [Come Here] and [Restart]. (Refer to "[Table 5-8 Type of Execution](#)".)

**Note2:** After user program execution has been stopped, the breakpoint by this function is eliminated.  
During execution of a user program by this function, break events set before the cursor position does not occur.

## 5.4.2 Breakpoint setting

Breakpoints can simply be set to the desired location by clicking in the [Source window](#) or [Assemble window](#).

Since breakpoints are set as break event conditions and managed using the [Event Function](#), restrictions apply to the number of breakpoints that can be set. (Refer to "[5.12.4 Number of enabled events for each event condition](#)".)

### (1) Breakpoint setting method

Breakpoints are executed by clicking lines in which "\*" is displayed (lines where program code exists).

In the default setting, software breakpoint (B) is set, but if [Breakpoint] is selected in the context menu, hardware breakpoint (B, or B) is set. (Refer to "[5.4.4 Hardware break and software break](#)".)

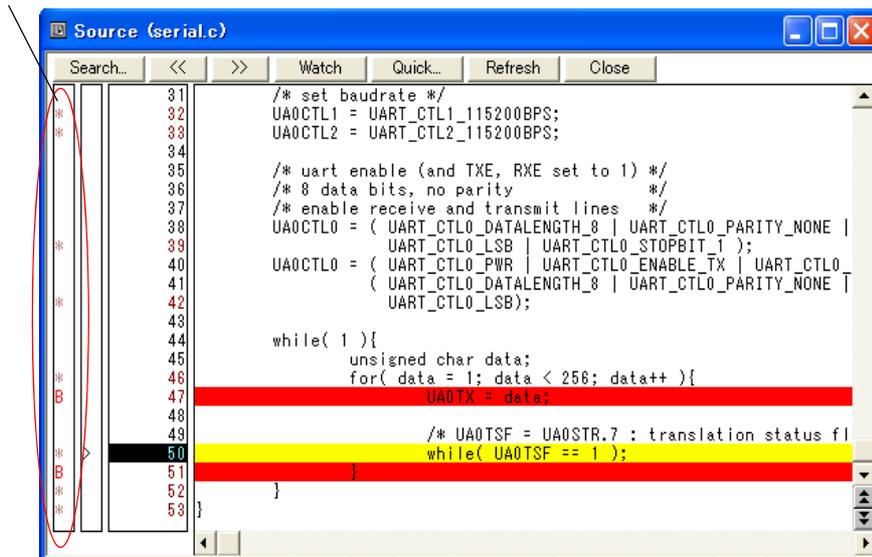
If a breakpoint is set on a line on which an event breakpoint has already been set, "A" indicating that multiple events have been set is marked. (Refer to "[Table 6-3 Event Setting Status \(Event Mark\)](#)".)

**Caution:** A software breakpoint cannot be set/delete in an externally mapped ROM area.

**Remark:** Breaks set by default can also be changed in the [Extended Option dialog box](#).

Figure 5-1 Breakpoint Setting

Click the asterisk (\*; program code) in this area.



### (2) Deleting a breakpoint method

Click the position at which the breakpoint to be deleted is set.

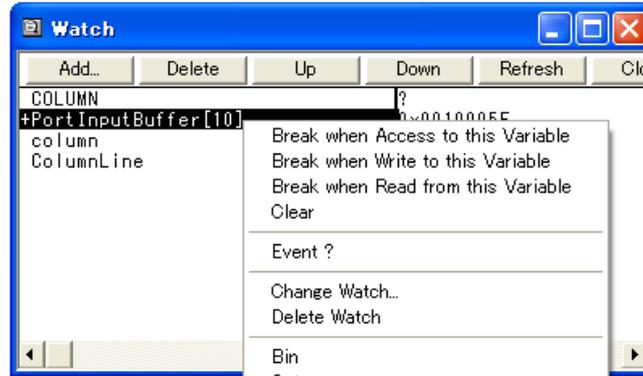
At the same time as setting is performed, in the default setting, software breakpoint (B) is deleted, but if [Breakpoint] is selected in the context menu, hardware breakpoint (B, or B) is deleted.

As a result of deletion, If another event remains, however, the mark of that event is displayed.

### 5.4.3 Setting breaks to variables

Access breaks can easily be set to variables from the context menu in the [Source window](#) or [Watch window](#).

Figure 5-2 Setting Break to Variable



### 5.4.4 Hardware break and software break

#### (1) Hardware break

Hardware breaks are breaks that are set using one hardware resource per event condition.

Therefore, in the ID78K0R-QB, they are managed using ["5.12 Event Function"](#) as break event conditions.

The number of valid break points varies depending on the device. (Refer to ["5.12.4 Number of enabled events for each event condition"](#).)

#### (2) Software break

Software breaks are breaks that are set by rewriting instructions of specified addresses to software break instructions. Settings to external ROM, stopping at variable access timing, etc., cannot be specified.

**Caution1:** If a software break is set to the execution start address in order to re-execute the code in the execution start address, the following events which are set to the address are not generated. **[IECUBE].**

- (1) Start of section trace
- (2) Start of section measurement
- (3) Trace delay trigger
- (4) Event after execution
- (5) Access event

To re-execute the code in the execution start address, use a break for the event before execution.

**Caution2:** Do not over write programs in which a software break has been set to the interval RAM area or external RAM area; otherwise, the break may not occur normally. Even if the break occurs, the program before being overwritten is automatically restored. Therefore, use software breaks to set breaks to programs in the RAM area.

**Caution3:** Do not assign a software break to addresses that are accessed.  
Use a hardware break for addresses to be accessed

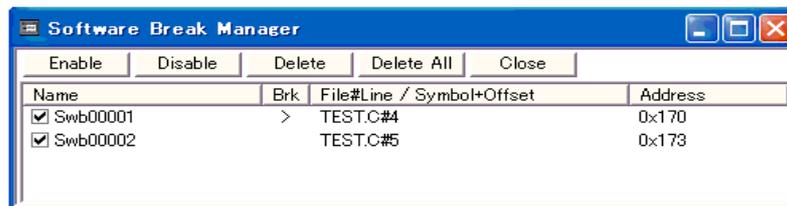
The number of valid software break is as follows:

Table 5-7 The Number of Valid Software Break

Connected IE	Valid Number
[IECUBE]	2000
[MINICUBE2]	2000

Software break is managed by the [Software Break manager](#).

Figure 5-3 Management of Software Breaks

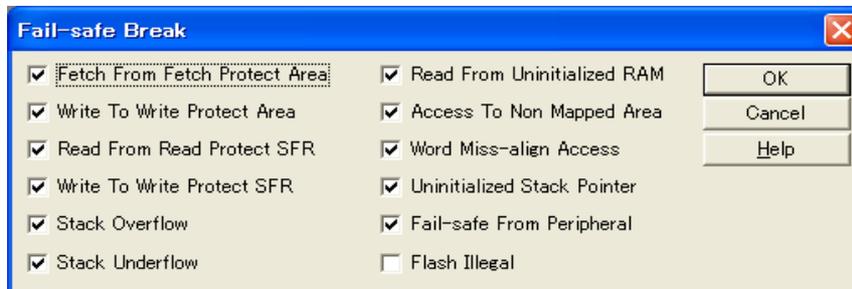


#### 5.4.5 Fail-safe break function [IECUBE]

The fail-safe break settings are performed in the [Fail-safe Break dialog box](#).

Individual settings are possible by selecting check boxes.

Figure 5-4 Fail-safe Break Setting



## 5.5 Program Execution Function

The program execution function is used to start/stop execution of the user program by the CPU and operation of the tracer.

Through user program execution, the program counter (PC) advances until the set breakpoint or forced break. (Refer to "5.4 Break Function".)

**Remark:** While the user program is being executed, trace event condition and timer event condition can be set. (Refer to " Trace dialog box", " Timer dialog box".)

The following types of ID78K0R-QB execution functions are provided. They are operated using the execution buttons on the toolbar, or from the [Run] menu.

Figure 5-5 Execution Button



Figure 5-6 [Run] Menu

File	Edit	View	Option	Run	Event	Browse	Jump	Simulator	Window	Help
				Restart				F4		
				Stop				F2		
				Go				F5		
				Ignore break points and Go				Ctrl+F5		
				Return Out				F7		
				Step In				F8		
				Next Over				F10		
				Start From Here				Shift+F6		
				Come Here				F6		
				Go & Go						
				Slowmotion						

Table 5-8 Type of Execution

Items	Contents
[Restart] 	The CPU is reset and the user program is executed starting from RESET address. This is the same operation as "resetting the CPU before execution of the user program and executing [Go]".
[Go] 	The user program is executed starting from the address indicated by the current PC register and is stopped if a set break event condition is satisfied.
[Ignore break points and Go] 	The user program is executed starting from the address indicated by the current PC register. Execution of the user program continues, ignoring set breakpoints.
[Return Out] 	The user program is executed until execution returns to the calling function described in C language.
[Step In] 	In the source mode, Step execution of one line of the source text is performed starting from the current PC register value and the contents of each window are updated. In the instruction mode, One instruction is executed from the current PC register value and the contents of each window are updated.
[Next Over] 	Next step execution is performed, assuming the function or subroutine called by the CALL / CALLT instruction as one step (step execution continues until the nesting level becomes the same as when the CALL / CALLT instruction was executed). <b>Instruction other than CALL / CALLT</b> The same processing as [Step In] is performed.
[Start From Here]	This command executes the user program starting from the specified address. Execution of the user program is stopped when a set break event condition is satisfied.
[Come Here]	The user program is executed from the address indicated by the current PC register to the address selected in the line/address display area of the <a href="#">Source window</a> or <a href="#">Assemble window</a> , and then a break occurs. While the user program is being executed, the break event currently set does not occur.
[Go & Go]	The user program is executed starting from the address indicated by the current PC register and stopped if a set break event condition is satisfied. The contents of each window are updated, and execution of the user program is resumed from the address at which the program was stopped. This operation is repeated until the user executes [Stop].
[Slowmotion]	Step execution of one line is performed from the address indicated by the current PC register value in the source mode. In the instruction mode, step execution of one instruction is performed. The contents of each window are updated each time step execution is performed. This operation is repeated until the user executes [Stop].
[CPU Reset] 	Resets the CPU.
[Stop] 	Forcibly stops program execution.

## 5.6 Watch Function

This section explains the following items related to the watch function:

- [Displaying and changing data values](#)
- [Displaying and changing local variable values](#)
- [Registering and deleting watch data](#)
- [Changing watch data](#)
- [Temporarily displaying and changing data values](#)
- [Callout watch function](#)
- [Stack trace display function](#)

### 5.6.1 Displaying and changing data values

Data values are displayed and changed in the [Watch window](#). Shifts in data values can be checked by registering watch data.

The display format is specified in the [Debugger Option dialog box](#).

Figure 5-7 Watch Window

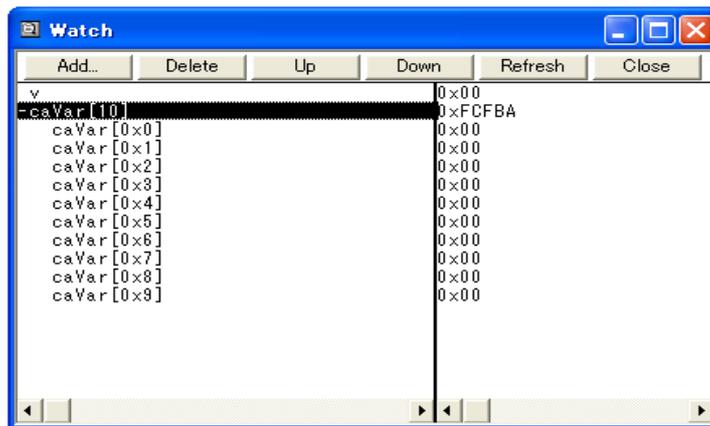


Figure 5-8 Specification of the Display Format (Debugger Option Dialog Box)

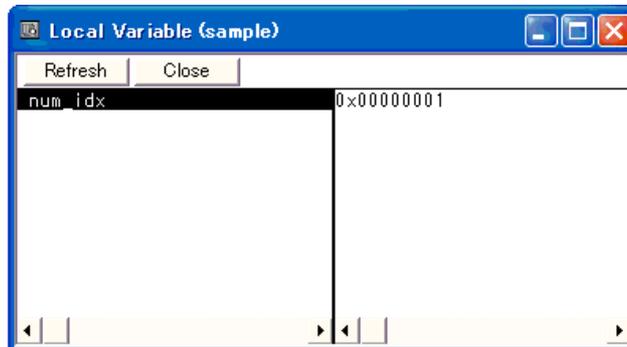


## 5.6.2 Displaying and changing local variable values

Local variables are displayed and changed in the [Local Variable window](#).

Local variables within the current function are automatically displayed in this window. (Variable addition/deletion is not possible.)

Figure 5-9 Local Variable Window



## 5.6.3 Registering and deleting watch data

Data can be registered to the [Watch window](#) from the [Source window](#) or [Assemble window](#). This is simply done by selecting the variable or symbol name in the respective window, and then clicking the [Watch] button. Registration is also possible with the following method.

- Drag and drop the selected variable or symbol name directly on the Watch window. (Refer to "[5.17.4 Drag & drop function](#)".)
- Click the [Add] button in the [Quick Watch dialog box](#) or [Add Watch dialog box](#).

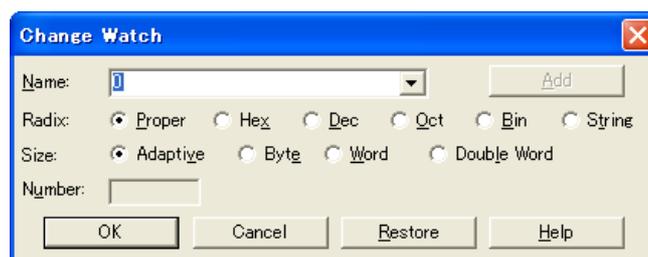
To delete watch data, click the variable name or symbol name (multiple selections can also be made using the Shift key or Ctrl key), and then click the [Delete] button. However, lines with an expanded hierarchy, such as elements of an array, and members of structures and unions, cannot be deleted.

## 5.6.4 Changing watch data

Watch data is changed in the [Change Watch dialog box](#).

Note that the symbol name can be changed even if it results in duplication of a name already in use with existing data.

Figure 5-10 Change Watch Dialog Box



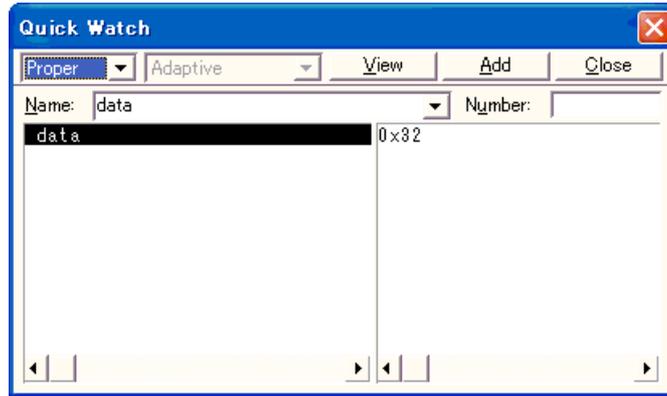
### 5.6.5 Temporarily displaying and changing data values

Data values are temporarily displayed and changed in the [Quick Watch dialog box](#).

Select the desired variable or symbol name in the [Source window](#) or [Assemble window](#) and click the [Quick...] button to perform watch data registration.

The display radix, display size, and display number can be changed in this window.

Figure 5-11 Quick Watch Dialog Box



### 5.6.6 Callout watch function

The corresponding variable value pops up when the mouse cursor is placed over a selected variable in the [Source window](#) or [Assemble window](#).

### 5.6.7 Stack trace display function

This function displays the stack contents of the current user program in the [Stack window](#).

Figure 5-12 Stack Window



## 5.7 Memory Manipulation Function

This section explains the following items related to the memory manipulation:

Verify check, etc., is specified in the [Extended Option dialog box](#).

- [Displaying and changing memory contents](#)
- [Access monitor function \[IECUBE\]](#)
- [Filling, copying, and comparing memory contents](#)
- [Flash memory writing function \[MINICUBE2\]](#)

### 5.7.1 Displaying and changing memory contents

In the [Memory window](#), the memory contents can be displayed or changed by using mnemonic codes, hexadecimal codes, and ASCII codes. Searching is done in the [Memory Search dialog box](#) displayed by clicking the [Search...] button. The results of search is highlighted in the [Memory window](#).

The display start position can be changed in the [Address Move dialog box](#) displayed by selecting [View] menu -> [Move...].

The variables and data allocated to the sampling range can be displayed in real time even during program execution. (Refer to "[5.14 RRM Function](#)".)

### 5.7.2 Access monitor function [IECUBE]

The access monitor function displays the access status (read, write, read & write) for the sampling range of the [RRM Function](#) using different colors in the [Memory window](#). The access monitor view is available only when "Byte" is selected for the display units. Colors are not displayed in the ASCII display area. Cumulative display setting and access status display can be cleared by selecting [View] menu -> [Access Monitoring].

**Caution:** The value of memory rewritten via DMA during program execution, and the value of memory rewritten from the ID78K0R-QB cannot be displayed on the access monitor.

**Remark:** Data fetched from the internal ROM area is displayed as read information.



## 5.8 Register Manipulation Function

This section explains the following items related to the register manipulation function.

- [Displaying and changing register contents](#)
- [Displaying and changing SFR contents](#)
- [Displaying and changing I/O port contents](#)

### 5.8.1 Displaying and changing register contents

Register contents can be displayed and changed in the [Register window](#).

Register name display switching (absolute name/function name) can be done in the [Debugger Option dialog box](#).

Figure 5-14 Absolute Name/Function Name Switching



Table 5-9 Absolute Name to Function Name Correspondence

Function Name		Absolute Name	
Pair Register	Register	Pair Register	Register
ax	x	rp0	r0
	a		r1
bc	c	rp1	r2
	b		r3
de	e	rp2	r4
	d		r5
hl	l	rp3	r6
	h		r7

### 5.8.2 Displaying and changing SFR contents

The SFR contents can be displayed and changed in the [SFR window](#).

The display start position can be changed in the [Address Move dialog box](#) displayed by selecting [View] menu -> [Move...].

The display register is selected in the [SFR Select dialog box](#).

### 5.8.3 Displaying and changing I/O port contents

User-defined I/O ports can be displayed and changed in the [SFR window](#) once they have been registered in the [Add I/O Port dialog box](#).

In the case of products that support programmable I/O registers, programmable I/O register contents can be displayed and changed by setting programmable I/O area use in the [Configuration dialog box](#).

Figure 5-15 Register I/O Port



## 5.9 Timer Function [IECUBE]

The timer function measures the execution time (run-break time) from the start of user program execution until a break, or the execution time in a specific user program interval using timer events.

The run-break time is also displayed in the statusbar in the [Main window](#).

This section explains the following items:

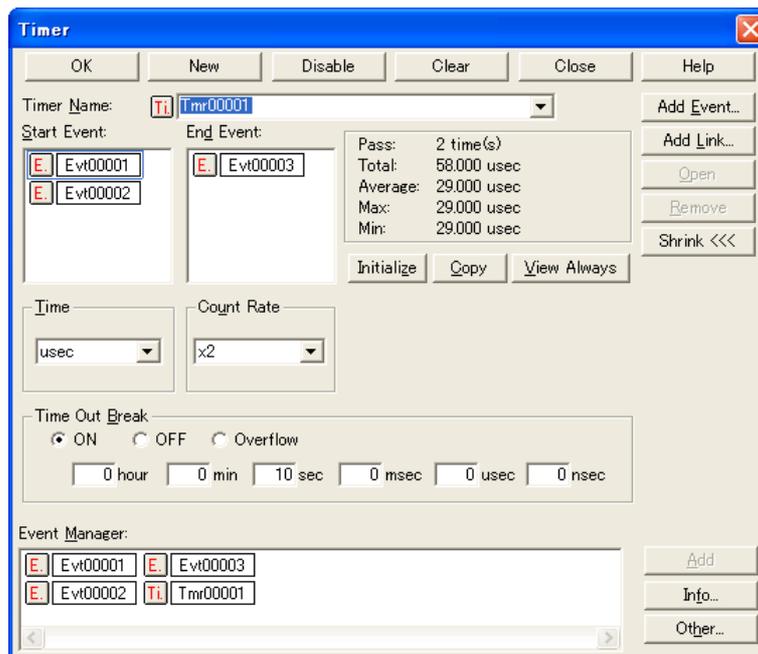
- [Timer event conditions](#)
- [Run-Break event](#)

### 5.9.1 Timer event conditions

A timer event condition specifies the trigger by which time measurement is started or stopped. Timer event conditions are set in the [Timer dialog box](#). (Refer to "5.12 Event Function".)

In the ID78K0R-QB, timeout break settings can be performed in the Time Out Break area.

Figure 5-16 Sets and Displays Timer Event (Timer Dialog Box)



Continuous display in the [Timer Result dialog box](#) can be selected by clicking the [View Always] button.

Timer manipulations during program execution are performed by selecting [Run] -> [Timer Start/Timer Stop].

## 5.9.2 Run-Break event

Run-Break event is a timer event name given to a timer event condition that measures the execution time from execution to break. Run-break events are registered in advance and the run-break time can be displayed through specification in the [Timer dialog box](#).

The run-break time is also displayed in the statusbar in the [Main window](#).

Since Run-Break events are included in the number of timer events that can be simultaneously enabled (refer to "[5.12.4 Number of enabled events for each event condition](#)"), they can be used added to the number of valid timer event conditions.

## 5.10 Trace Function [IECUBE]

The trace function is used to save the history of the data indicating the execution process of the user program to the trace memory.

This section explains the following items:

- [Trace memory](#)
- [Checking trace data](#)
- [Mixed display mode \(Trace window\)](#)
- [Tracer Operation](#)
- [Setting conditional trace](#)

### 5.10.1 Trace memory

The ID78K0R-QB has trace memory with a ring structure.

The maximum trace memory capacity is as follows.

Table 5-10 Trace Memory Size

Connected IE	Maximum Value
[IECUBE]	128 KB

### 5.10.2 Checking trace data

The trace data saved to the trace memory can be checked in the [Trace View window](#). Trace data can be searched in the [Trace Search dialog box](#) displayed by clicking the [ Search...] button.

The display start position can be changed in the [Trace Move dialog box](#) displayed by selecting [View] -> [Move].

The display items in the [Trace View window](#) can be selected in the [Trace Data Select dialog box](#).

This area is used to set the division ratio of the counter used for time tag display in the [Trace View window](#). The division ratio is set in the [Extended Option dialog box](#).

Figure 5-17 Checking Trace Data

Frame	h	m	s	ms	us	ns	Address	Data	Status	Address	R.Data	W.Data	Status	DisAsm
A	000910	0:00:00.000	402	250	0013E		47	M1						CMPW AX,HL
A	000911	0:00:00.000	402	500	0013F		DF8	M1						BNZ \$_@cstart+0x75
A	000912	0:00:00.000	403	000	00141		4100	M1						MOV ES,#0H
A	000913	0:00:00.000	403	250	00143		362400	M1						MOVW HL,#24H
A	000914	0:00:00.000	403	500	00146		3420FE	M1						MOVW DE,#0FE20H
A	000915	0:00:00.000	403	750	00149		EF05	M1						BR \$_@cstart+0x8c
A	000916	0:00:00.000	404	500	00150		17	M1						MOVW AX,HL
A	000917	0:00:00.000	404	750	00151		442400	M1						CMPW AX,#24H
A	000918	0:00:00.000	405	000	00154		DF5	M1						BNZ \$_@cstart+0x87
A	000919	0:00:00.000	405	500	00156		3620FF	M1						MOVW HL,#0FE20H
A	000920	0:00:00.000	405	750	00159		3020FE	M1						MOVW AX,#0FE20H
A	000921	0:00:00.000	406	000	0015C		EF04	M1						BR \$_@cstart+0x9e
A	000922	0:00:00.000	406	750	00162		47	M1						CMPW AX,HL
A	000923	0:00:00.000	407	000	00163		DF9	M1						BNZ \$_@cstart+0x9a
A	000924	0:00:00.000	407	500	00165		FC9F0100	M1						CALL !_main
A	000925	0:00:00.000	407	750						FFE1E		00	W	
A	000926	0:00:00.000	408	000						FFE1C		0169	W	
A	000927	0:00:00.000	408	250	0019F		C7	M1						PUSH HL
A	000928	0:00:00.000	408	500	001A0		2006	M1		FFE1A		FE20	W	SUBW SP,#6H
A	000929	0:00:00.000	408	750	001A2		FBF8FF	M1						MOVW HL,SP
A	000930	0:00:00.000	409	000	001A5		F6	M1						CLRW AX
A	000931	0:00:00.000	409	250	001A6		BC04	M1						MOVW [HL+4H],AX
A	000932	0:00:00.000	409	500	001A8		E6	M1		FFE18		0000	W	ONW AX
A	000933	0:00:00.000	409	750	001A9		A1	M1						INCW AX
A	000934	0:00:00.000	410	000	001AA		F7	M1						CLRW BC
A	000935	0:00:00.000	410	250	001AB		BB	M1						MOVW [HL],AX
A	000936	0:00:00.000	410	500	001AC		33	M1		FFE14		0002	W	XCHW AX,BC
A	000937	0:00:00.000	410	750	001AD		BC02	M1						MOVW [HL+2H],AX
A	000938	0:00:00.000	411	000						FFE16		0000	W	
Software Break														

### 5.10.3 Mixed display mode (Trace window)

Source file display combined with trace results can be done by selecting [View] -> [Mix] in the [Trace View window](#) (mixed display mode).

If a program code corresponds on the program fetch address to be displayed, a source file line is displayed before the line indicating the result of tracing that program fetch address.

The source file line is displayed, emphasized in [green](#).

Frame	Time	Address	Data	Status	Address	Data	Status	DisAsm
32757	3	000005A2	85058505	BRM1				br _main+0x21a
			while(1);					
32758	3	000005A2	85058505	BRM1				br _main+0x21a
			while(1);					
32759	3	000005A2	85058505	BRM1				br _main+0x21a
			while(1);					
32760	3	000005A2	85058505	BRM1				br _main+0x21a
			while(1);					

**Caution:** The mixed display mode is valid only when the load module has been downloaded and symbol information is read, and when a fetch address, fetch data, fetch status, or result of disassembly is displayed.

### 5.10.4 Tracer Operation

The trace operation differs as follows according to the user program execution format and the tracer control mode.

Tracer manipulations during program execution are performed by selecting [Run] -> [Tracer Start/Tracer Stop].

#### (1) Operation during execution

The tracer operation differs as follows according to [Run] -> [Cond. Trace ON/Uncond. Trace ON] selection.

Table 5-11 Type of Trace Modes

Item	Contents
Unconditional trace	Trace is started when execution of user program, and ends when a break occurs. At this time, the set trace event conditions are ignored.
Conditional trace	Trace is started or stopped by the condition set in the <a href="#">Trace dialog box</a> . (Refer to " <a href="#">5.10.5 Setting conditional trace</a> ".) If a break occurs while a trace is being executed, however, trace is stopped immediately.

**(2) Operation during Step In execution**

The tracer operates every step execution, and trace data of one step is successively added to the trace memory.

**(3) Operation during Next Over execution**

The operation of the tracer differs depending on the instruction to which Next Over is to be executed.

## (a) CALL / CALLT Instruction

CALL / CALLT The jarl instruction and the subroutine that was called are traced.

## (b) Other instructions

The same operation as that during Step In execution is performed.

**(4) Tracer control mode**

There are the following types of trace control mode. These trace mode settings are performed from the [Run] menu.

Table 5-12 Types of Tracer Control Mode

Mode	Contents
Non Stop	Goes around the trace memory and overwrites data from the oldest frame (default).
Full Stop	Goes around the trace memory and then stops the tracer.
Full Break	Goes around the trace memory and then stops the tracer and program execution.
Delay Trigger Stop	Traces data by the number of delay count frames and stops the tracer when a delay trigger event has occurred.
Delay Trigger Break	Traces data by the number of delay count frames and stops the tracer and program execution when a delay trigger event has occurred.

## 5.10.5 Setting conditional trace

A trace event condition triggers starting/stopping trace execution when a conditional trace is set.

By setting a trace event condition in the [Trace dialog box](#), the conditional trace can be set. (Refer to "[5.12 Event Function](#)".)

To use the conditional trace, select [Run] menu -> [Cond. Trace ON].

There are the following types of conditional trace.

In the ID78K0R-QB, all the following conditional traces can be set as one trace event condition.

Table 5-13 Types of Conditional Trace

Item	Contents, Setting Method
Section trace	Executes a trace between two specified conditions (in a specific zone). A section trace can be executed by setting a trace start event and trace end event in the <a href="#">Trace dialog box</a> and selecting [Run] -> [Cond. Trace ON]. In the ID78K0R-QB, up to 4 ranges can be specified simultaneously.
Qualify trace	Executes a trace only when a condition is satisfied. If two or more events are set as qualify trace events, a qualify trace can be executed by executing a conditional trace. A qualify trace can be executed by setting a qualify trace event in the <a href="#">Trace dialog box</a> and selecting [Run] -> [Cond. Trace ON].
Delay trigger trace	Executes a trace by the number of delay counts after a condition has been satisfied. A delay trigger trace can be executed by setting a delay trigger event in the <a href="#">Trace dialog box</a> , setting a delay count in the <a href="#">Delay Count dialog box</a> and selecting [Run] -> [Cond. Trace ON]. In this case, select [Delay Trigger Stop], or [Delay Trigger Break]

## 5.11 Coverage Measurement Function [IECUBE]

Although there are several types of coverage measurement, the ID78K0R-QB performs measurement for C0 coverage.

C0 coverage (instruction coverage): A percentage that all statements in a code are executed at least once

Download or upload the coverage measurement result (coverage data) via the [Download dialog box](#) or [Upload dialog box](#), respectively.

This section explains the following items:

- [Coverage measurement result display](#)
- [Coverage measurement range](#)
- [Display of locations for which coverage measurement is executed](#)

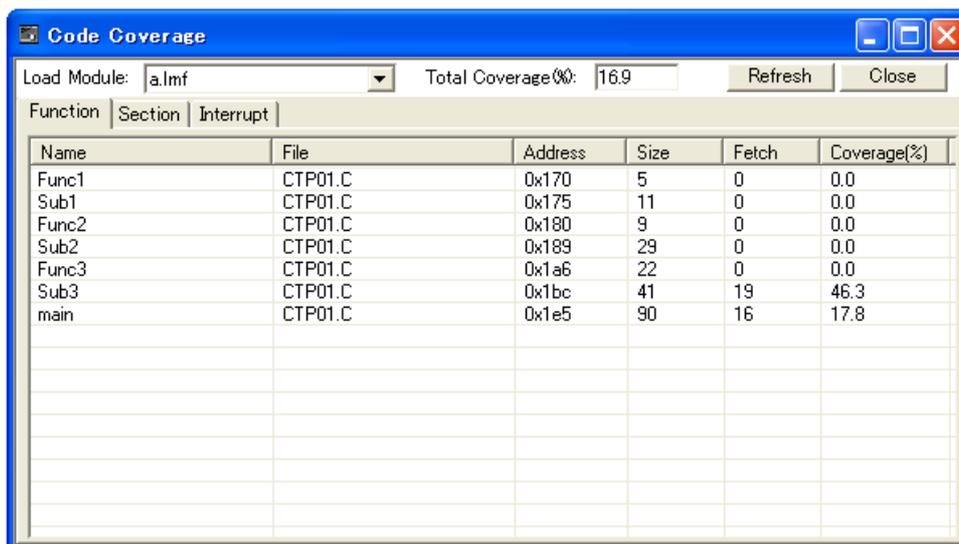
### 5.11.1 Coverage measurement result display

The coverage measurement result can be checked in the [Code Coverage window](#).

In the [Code Coverage window](#), the measurement result is displayed individually for functions, sections, and interrupt handlers (vectors). The coverage measurement result is updated at a break (it is not updated automatically during user program execution). Clear the coverage data by selecting [Option] menu. Coverage data can be saved in the CSV format. (Refer to "[5.16.2 Window display information \(view file\)](#)".)

**Remark:** Coverage data is cleared when the ID78K0R-QB is started.

Figure 5-18 Coverage Measurement Result Display



The screenshot shows a window titled "Code Coverage" with a blue title bar. Below the title bar, there is a "Load Module:" dropdown menu set to "a.lmf", a "Total Coverage%" field showing "16.9", and "Refresh" and "Close" buttons. Below this, there are three tabs: "Function", "Section", and "Interrupt", with "Function" selected. The main area contains a table with the following data:

Name	File	Address	Size	Fetch	Coverage(%)
Func1	CTP01.C	0x170	5	0	0.0
Sub1	CTP01.C	0x175	11	0	0.0
Func2	CTP01.C	0x180	9	0	0.0
Sub2	CTP01.C	0x189	29	0	0.0
Func3	CTP01.C	0x1a6	22	0	0.0
Sub3	CTP01.C	0x1bc	41	19	46.3
main	CTP01.C	0x1e5	90	16	17.8

### 5.11.2 Coverage measurement range

The coverage measurement range is as follows.

Table 5-14 Code Coverage Measurement Range

Connected IE	Code Coverage Measurement Range
[IECUBE]	Internal ROM space + External memory space

### 5.11.3 Display of locations for which coverage measurement is executed

The locations for which coverage measurement is executed in the user program are displayed in the [Source window](#) and [Assemble window](#) based on the coverage measurement information.

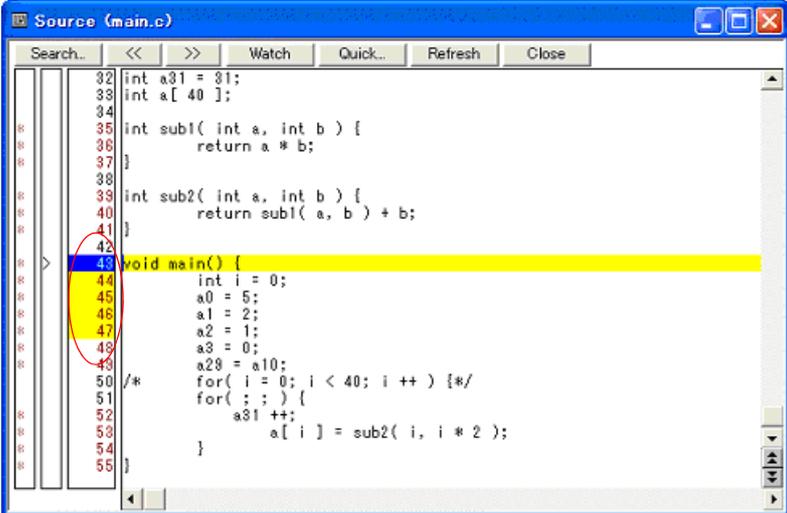
The display result can be saved as view files for the [Source window](#) and [Assemble window](#). (Refer to "5.16.2 Window display information (view file)".)

The numbers of line or addresses for which coverage measurement is executed are highlighted as shown in the table below. In the view file, the marks in the table below are appended to the relevant line number or address instead of using the background color.

Table 5-15 Format of View of Locations for Which Coverage Measurement Executed

Coverage	Source Window		Assemble Window	
	Background Color	Mark	Background Color	Mark
Code on this line has been executed by 100%	Yellow	@	Yellow	@
Code on this line has been executed by 1 to 99%	Orange	+	-	-
Code on this line has been executed by 0% (not yet executed)	No highlight color	-	No highlight color	-

Figure 5-19 View of Locations for Which Coverage Measurement Executed



```
Source (main.c)
Search... << >> Watch Quick... Refresh Close
32 int a31 = 31;
33 int a[ 40 ];
34
35 int sub1( int a, int b ) {
36     return a * b;
37 }
38
39 int sub2( int a, int b ) {
40     return sub1( a, b ) + b;
41 }
42
43 void main() {
44     int i = 0;
45     a0 = 5;
46     a1 = 2;
47     a2 = 1;
48     a3 = 0;
49     a29 = a10;
50     /* for( i = 0; i < 40; i ++ ) {*/
51     for( ; ; ) {
52         a31 ++;
53         a[ i ] = sub2( i, i * 2 );
54     }
55 }
```

**Remark:** With the ID78K0R-QB, the cumulative view of the locations for which coverage measurement has been executed can be displayed and coverage data can be cleared using the context menu on the [Source window](#) or [Assemble window](#).

## 5.12 Event Function

Events specify specific states of the target system during debugging ,such as "fetched address 0x1000" or "Wrote data to address 0x2000".

In the ID78K0R-QB, such events are used as action triggers for each debugging function, such as break and trace.

This section explains the following items:

- [Using event function](#)
- [Creating events](#)
- [Setting event conditions](#)
- [Number of enabled events for each event condition](#)
- [Managing events](#)

### 5.12.1 Using event function

Events (event conditions and event rink conditions) consist of the event conditions listed in the following table, by assigning various debugging functions. As a result, event conditions can be utilized according to the debugging purpose.

Table 5-16 Various Event Conditions

Event Condition	Mark	Contents ->Setting Dialog Box
Break event	B	Condition in which the execution of the user program or operation of a tracer is stopped. (Refer to " <a href="#">5.4 Break Function</a> ".) -> <a href="#">Break dialog box</a>
Trace event	T	Condition in which the process of user program execution is saved to the trace memory. (Refer to " <a href="#">5.10 Trace Function [IECUBE]</a> ".) -> <a href="#">Trace dialog box</a>
Timer event	Ti	Condition for specifying the time measurement start timing and stop timing. (Refer to " <a href="#">5.9 Timer Function [IECUBE]</a> ".) -> <a href="#">Timer dialog box</a>
Snapshot event condition	S	Condition for specifying the timing of executing a snapshot. (Refer to " <a href="#">5.13 Snapshot Function [IECUBE]</a> ".) -> <a href="#">Snap Shot dialog box</a>
Event DMM	M	Condition for specifying the timing of executing DMM. (Refer to " <a href="#">5.15 DMM Function</a> ".) -> <a href="#">Event DMM dialog box</a>

## 5.12.2 Creating events

Events can be used as action triggers of various event conditions described before through registration of event conditions and event link conditions, individually naming states called events.

### (1) Creating and registering events

The creation of event conditions is done in the [Event dialog box](#).

Set an address condition, status condition, and data condition in this dialog box. Specify a combination of these as one event condition and name and register this event condition.

A simple method consists in using event conditions generated by setting breakpoint in the [Source window](#) and [Assemble window](#). (Refer to "5.4.2 Breakpoint setting".)

### (2) Creating and registering event links

Event link conditions are conditions for single events that provide ordered restrictions for event conditions, and are generated when user programs are executed according to the specified sequence.

To create an event link condition, use the [Event Link dialog box](#).

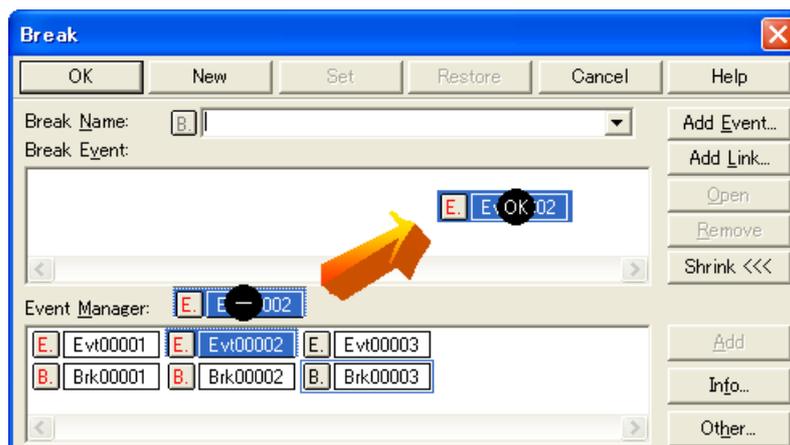
## 5.12.3 Setting event conditions

Various event conditions listed in [Table 5-16](#) are individually created in the corresponding dialog box.

### (1) Setting of Various Event Conditions

The setting of the various event conditions is done by selecting the event icon of the desired event condition or event link condition displayed in the event manager area (or [Event manager](#)) in the respective setting dialog box, and dragging and dropping this icon in the condition area to be set.

Figure 5-20 Setting of Various Event Conditions



The shape of the mouse cursor changes to "OK" when it is dragged over a settable condition area.

Regarding the created event conditions, the event icon mark becomes red and the setting is enabled by clicking the [Set] button or the [OK] button in the Setting dialog box. After the event has been set, a debugging action occurs as various event conditions.

## (2) Settings using selection mode (settings after checking contents)

The [Event dialog box](#) or the [Event Link dialog box](#) are open in the selection mode by placing the focus on the condition area to be set and then clicking the [Add Event...] button or the [Add Link...] button. Because when a condition set in the dialog box is selected, the corresponding detailed condition is displayed, conditions can be set after checking the contents.

## (3) Copying and moving event icons

In the event condition setting area, event conditions can be copied and moved through drag & drop operation using the following methods.

- If event condition was dropped using only the mouse, move event condition.
- If the event condition was dropped while pressing the Ctrl key, copy the event condition.

## (4) Manipulation in event manager area

Event conditions can be set by clicking the [Add] button after placing the focus on the condition area to be set and selecting an event icon.

### Event setting content display

Select an event and click the [Open] button or double-click the event. The setting dialog box corresponding to the selected event will be opened and the set contents of the event will be displayed.

### Deletion

An event can be deleted by selecting the event and then clicking the [Remove/Delete] button or pressing the Delete key.

### Changing display mode and sorting

The display mode of and sorting in the event manager area can be selected by clicking the [Info...] button.

### Area non-display

An area can be hidden by clicking the [Shrink<<< ] button.

## 5.12.4 Number of enabled events for each event condition

Up to 256 conditions can be registered as event conditions or various event conditions.

One event condition or link event condition can be set for multiple types of events such as break and trace.

One event condition or link event condition can be set for multiple types of events such as break, trace and timer.

However, the number of event conditions that can be simultaneously set (enabled) is limited as follows.

Therefore, if the valid number is exceeded or if the used event conditions or event link conditions exceed the maximum number that can be used simultaneously, it is necessary to disable the set various event conditions once and then register them again. (Refer to "[5.12.5 Managing events](#)".)

Table 5-17 Number of Enabled Events for Each Event Condition

Connected IE	Event		Event Link	Break	Trace	Snapshot	Timer	Event DMM
	Execute	Access						
[IECUBE]	8	8	2	20 - 24 <sup>a.</sup>	1 <sup>b.</sup>	3	2	4
[MINICUBE2]	1 <sup>c.</sup>	1 <sup>c.</sup>	-	1 <sup>c.</sup>	-	-	-	-

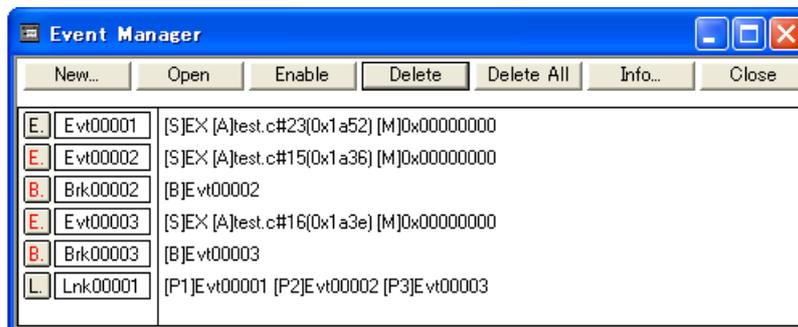
- a. Among the enabled events, four to eight breaks are "execution before break" (the number of settable breaks varies depending on the address and instruction to be set).
- b. Up to 4 section trace events, 1 qualify trace, and 1 delay trigger trace can be set simultaneously.
- c. One event can be set as an "execution after break" or an access event.

### 5.12.5 Managing events

Managing events is done with the [Event manager](#).

The Event Manager allows display, enabling/disabling, and deletion of the [Various Event Conditions](#).

Figure 5-21 Managing Events (Event Manager)



#### (1) Event icon

Event icons consist of a mark and an event name indicating the type of event. The color of each event icon indicates the setting status of that event.

Enable/disable is switched by clicking the mark part.

Table 5-18 Event Icon

Character Color	Mark	Meaning
Red	E.L.	Indicates that the event condition or event link condition which is used for various event conditions is enabled.
	B.T.Ti.S.M.	Indicates that the <a href="#">Various Event Conditions</a> is valid. The various events occur when its condition is satisfied.

Character Color	Mark	Meaning
Black	E.L.	Indicates that the event condition or event link condition which is used for various event conditions is disabled.
	B.T.Ti.S.M.	Indicates that the <a href="#">Various Event Conditions</a> is invalid. The various events do not occur even when its condition is satisfied.
Yellow	E.L.	Indicates that the symbol specified for an event is held pending because it cannot be recognized by the program currently loaded.
	B.T.Ti.S.M.	Indicates that the <a href="#">Various Event Conditions</a> is held pending. The various events do not occur even when its condition is satisfied.

## 5.13 Snapshot Function [IECUBE]

The snapshot function is used to save the contents of registers, memory, and SFR of the user program execution process to the trace memory as snap data.

This section explains the following items:

- [Snapshot event conditions](#)
- [Snap data](#)

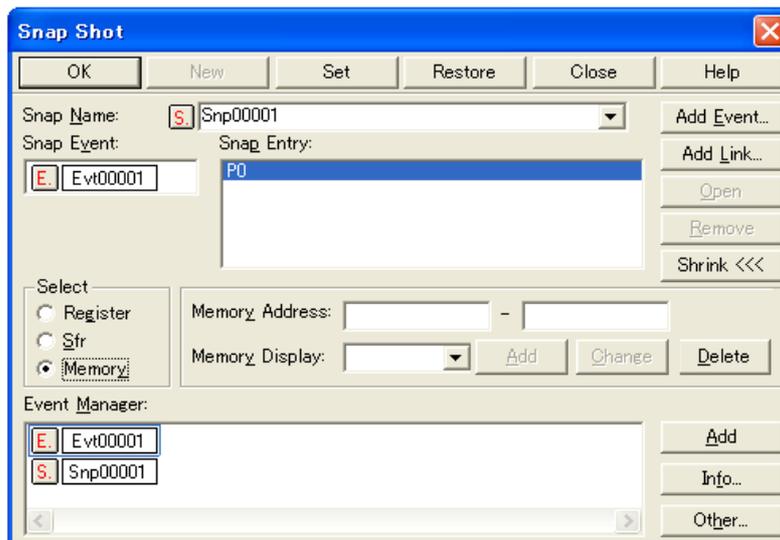
### 5.13.1 Snapshot event conditions

A snapshot event condition specifies the trigger by which a snapshot is to be executed.

Snapshot event conditions are set in the [Snap Shot dialog box](#). (Refer to "5.12 Event Function".)

When the snapshot event is executed, execution of the user program momentarily breaks.

Figure 5-22 Snap Shot Dialog Box



### 5.13.2 Snap data

One snap event condition can specify the collection of up to 16 snap data.

The following types of data can be collected as snap data.

- Register value
- SFR value
- Memory contents

## 5.14 RRM Function

This section explains the following items related to the RRM function.

- [Real-time RAM monitor function \[IECUBE\]](#)
- [Pseudo real-time RAM monitor function \(Break When Readout\)](#)

### 5.14.1 Real-time RAM monitor function [IECUBE]

[Table 5-19](#) shows the range of data that can be loaded using the real-time RAM monitor function.

The variables and data allocated to this area can be displayed always in real time in the [Watch window](#) and [Memory window](#).

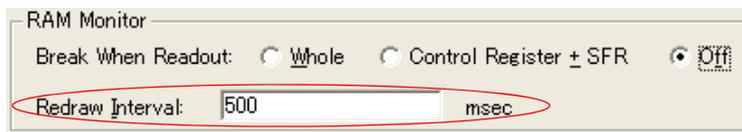
The sampling interval can be specified in the [Extended Option dialog box](#).

Table 5-19 Areas for Which Sampling Can Be Performed with Real-time RAM Monitor Function

Connected IE	Sampling Range
[IECUBE]	PC and Internal ROM area <sup>Note</sup> / Internal RAM area

**Note:** If "OFF" is selected for "Break When Readout" for the internal ROM area setting, data before starting execution is displayed in real time.

Figure 5-23 Specification of Interval for Sampling with Real-time RAM Monitor Function



### 5.14.2 Pseudo real-time RAM monitor function (Break When Readout)

To read areas that cannot be read using the real-time RAM monitor function, the pseudo real-time RAM monitor function can be used instead.

The memory area is read by software simulation while the pseudo real-time RAM monitor function is executed, so execution of the user program momentarily breaks upon a read.

The variables and data allocated to this area can be displayed in close to real time in the [Watch window](#) and the [Memory window](#)<sup>Note</sup>.

Specify turning on/off of the pseudo real-time RAM monitor function and the sampling range in the [Extended Option dialog box](#).

**Note:** In this case, the target memory area is displayed as "\*\*\*".

Figure 5-24 Specification of Pseudo Real-time RAM Monitor Function [IECUBE]



RAM Monitor

Break When Readout:  Whole  Control Register ± SFR  Off

Redraw Interval: 500 msec

Figure 5-25 Specification of Pseudo Real-time RAM Monitor Function [MINICUBE2]



RAM Monitor And DMM

Break When Access:  Whole  IRAM  Off

Redraw Interval: 500 msec

## 5.15 DMM Function

DMM (Dynamic Memory Modification) is a function that rewrites the contents of the memory (RAM) in real time during user program execution.

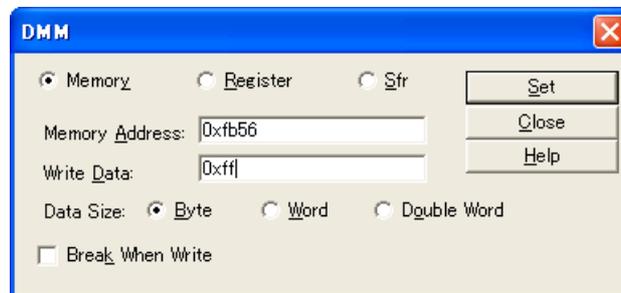
The [DMM dialog box](#) is opened by clicking the [DMM... ] button on the [Memory window](#), [Register window](#) or [SFR window](#). Specify the DMM target address and data.

**Remark:** After "Break When Write" is selected in the [DMM dialog box](#), DMM (pseudo DMM) by software simulation is selectable (execution of the user program momentarily breaks upon a write).

If an area for which a real-time write is not permitted (register or SFR) is specified in the [DMM dialog box](#), or when MINICUBE2 is used, the setting is fixed to pseudo DMM.

**Caution:** DMM cannot be used for DataFlash areas.

Figure 5-26 Modifying Memory Contents (DMM Dialog Box)



This section explains the following items:

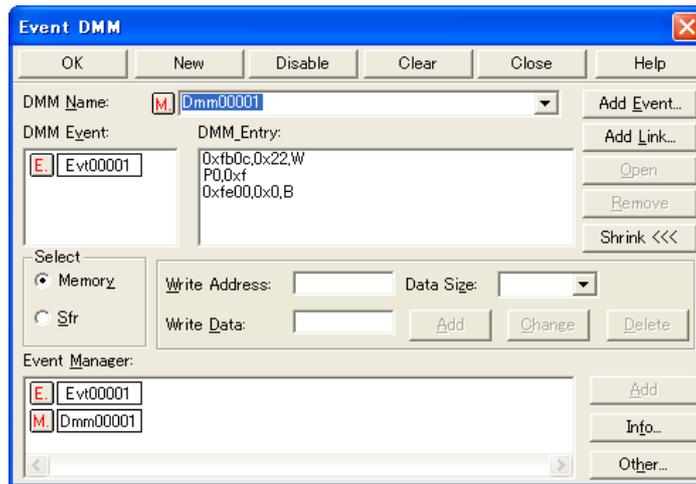
- [Event DMM condition \[IECUBE\]](#)

### 5.15.1 Event DMM condition [IECUBE]

The event DMM is a function that is used to write any data to the specified address (memory or SFR) when the specified event is established during user program. A break occurs instantaneously upon a write during the user program execution.

Event DMM conditions are set in the [Event DMM dialog box](#). (Refer to "5.12 Event Function".)

Figure 5-27 Event DMM Dialog Box



## 5.16 Load/Save Function

The ID78K0R-QB allows saving and loading the following types of information as files. As a result, recovery of these various types of information is possible.

- [Debugging environment \(project file\)](#)
- [Window display information \(view file\)](#)
- [Window setting information \(setting file\)](#)

**Remark:** The simple window status can be maintained by selecting [Window] menu -> [Static]. (Refer to ["5.17.1 Active status and static status"](#).)

### 5.16.1 Debugging environment (project file)

A project file (\*.prj) is a file that records the debugging environment.

A project file is created when the debugging environment at a particular point in time is saved, and that debugging environment can be restored by loading this file at a subsequent time.

Project files are loaded and saved in the [Project File Save dialog box](#) and [Project File Load dialog box](#), respectively. To load a project file at startup, click the [Project...] button in the [Configuration dialog box](#).

The following contents are saved to the project file:

Table 5-20 Contents Saved to Project File

Window Name	Saved Contents
<a href="#">Configuration dialog box</a>	All items (target device, clock setting, pin mask setting, mapping information)
<a href="#">Main window</a>	Display position, toolbar/statusbar/button display information, execution mode information
<a href="#">Download dialog box</a>	File information to be downloaded
<a href="#">Extended Option dialog box</a> <a href="#">Debugger Option dialog box</a> <a href="#">Fail-safe Break dialog box</a> <a href="#">Flash Option dialog box</a> <a href="#">DataFlash Option dialog box</a> <a href="#">Pseudo Emulation dialog box</a>	Set information
<a href="#">Assemble window</a> <a href="#">Memory window</a>	Display information of window, display start address
<a href="#">Source window</a> <a href="#">Stack window</a> <a href="#">SFR window</a> <a href="#">Local Variable window</a> <a href="#">Trace View window</a> <a href="#">Code Coverage window</a> <a href="#">Event manager</a> <a href="#">Console window</a> <a href="#">Expansion Window</a> <a href="#">Register window</a>	Display information of window

Window Name	Saved Contents
<a href="#">Event dialog box</a> <a href="#">Event Link dialog box</a> <a href="#">Break dialog box</a> <a href="#">Trace dialog box</a> <a href="#">Snap Shot dialog box</a> <a href="#">Timer dialog box</a> <a href="#">Event DMM dialog box</a>	Display information of window, variable event information
<a href="#">List window</a>	Display position of window
<a href="#">Watch window</a>	Display information of window <sup>Note</sup> , watch registration information
<a href="#">Add I/O Port dialog box</a>	Added I/O port information
<a href="#">Delay Count dialog box</a>	Delay count value
<a href="#">Software Break manager</a>	Display information of window, software break information

**Note:** The display status of members of a structure pointer, array pointer and so on, and radix for displaying individual member are not saved.

## 5.16.2 Window display information (view file)

A view file is a file that records window display information.

View files can be loaded and saved for each window.

When a view file is loaded, a reference window ([Source window](#) in the static status) is displayed and the display information at the time of saving is displayed.

View files are loaded and saved in the [View File Load dialog box](#) and [View File Save dialog box](#), respectively.

Table 5-21 Type of View Files

File Type	Current Window Name, File Name
Source Text (*.sw)	<a href="#">Source window</a> <sup>Note1</sup>
Assemble (*.dis)	<a href="#">Assemble window</a> <sup>Note1</sup>
Memory (*.mem)	<a href="#">Memory window</a>
Watch (*.wch)	<a href="#">Watch window</a>
Register (*.rgw)	<a href="#">Register window</a>
SFR (*.sfr)	<a href="#">SFR window</a>
Local Variable (*.loc)	<a href="#">Local Variable window</a>
Stack Trace (*.stk)	<a href="#">Stack window</a>
Trace (*.tvw)	<a href="#">Trace View window</a>
Code Coverage (*.csv)	<a href="#">Code Coverage window</a> (Data is saved separately for the tab selected)
List (*.csv)	<a href="#">List window</a> (Data is saved separately for the tab selected)

File Type	Current Window Name, File Name
Console (*.log)	<a href="#">Console window</a>
All (*.*)	All files
Source (*.c, *.s, *.asm)	Source file <sup>Note2</sup>
Text (*.txt)	Text file

**Note1:** The mark for indicating the code coverage measurement result (executed/not yet executed) is added to the contents of the displayed file. (Refer to "[Table 5-15 Format of View of Locations for Which Coverage Measurement Executed](#)".) [IECUBE]

**Note2:** The extension of the source file can be changed in the [Extended Option dialog box](#).

### 5.16.3 Window setting information (setting file)

A setting file is a file that records the window setting information (watch data settings, SFR settings, and event settings).

Setting files can be loaded and saved for each window.

When a setting file is loaded, the target window is displayed and the setting information that was saved is restored.

Setting files are loaded and saved in the [Environment Setting File Load dialog box](#) and [Environment Setting File Save dialog box](#), respectively.

Table 5-22 Type of Setting Files

File Type	Current Window Name
Watch (*.wch)	<a href="#">Watch window</a> <sup>Note</sup>
SFR (*.sfr)	<a href="#">SFR window</a> <sup>Note</sup>
Event (*.evn)	<a href="#">Event manager</a>

**Note:** A variable value can not be loaded.

## 5.17 Functions Common to Each Window

The windows have the following common functions.

- [Active status and static status](#)
- [Jump function](#)
- [Trace result with linking window \[IECUBE\]](#)
- [Drag & drop function](#)
- [Cautions](#)

### 5.17.1 Active status and static status

Each of the windows below has two statuses: The [Active status](#) and [Static status](#).

- [Source window](#) (that is displaying the source file to which symbol information is read)
- [Assemble window](#)
- [Memory window](#)

Only one window can be opened in the active status. However, because two or more windows in the static status can be opened, the current status of the windows can be temporarily held.

Select this status by the [Window] menu.

#### (1) Active status

The display position and contents of the window in the active status are automatically updated in association with the current PC value.

This window is also the jump destination of the [Jump function](#). If this window is linked with the [Trace View window](#), the contents displayed in the active window are updated in association with the [Trace View window](#).

Only one window can be opened in the active status.

#### (2) Static status

The display position of the window in the static status does not move in association with the current PC value, but the displayed contents are updated.

The static window is not used as the jump destination of the [Jump function](#). In addition, it is not linked with the [Trace View window](#).

If an active window is already open, the next window is opened in the static status.

Two or more static windows can be opened at the same time.

## 5.17.2 Jump function

This is a function that jumps to any of the windows below from a line or address (a jump pointer) on which the cursor is put. The window to which the jump is made is displayed on the jump pointer.

- [Source window](#)
- [Assemble window](#)
- [Memory window](#)

You can jump among the above windows, or from the [Trace View window](#), [Stack window](#), [Event manager](#) and [Register window](#) to the above windows.

### (1) Jump method

The jump method is as follows:

- 1) Move the cursor to the line or address that is to be used as the jump pointer, on the window from which jumping is possible (select an event icon on the Event Manager).
- 2) Select the following menu item to which execution is to jump from the [Jump] menu.

**Caution:** If a program code does not exist on the line at the cursor position, the first address of the line with a program code above or below that line is used as the jump pointer.

### (2) Details of jump source address

The details of jump source address is as follows:

Table 5-23 Details of Jump Source Address

Target Window	Details of Jump Pointer	
From the <a href="#">Register window</a>	Registers selected	
From the <a href="#">Memory window</a>	Address at the cursor position	
From the <a href="#">Event manager</a>	If the selected event icon is that of an event condition, an address condition is used as the jump pointer.	
	If the address condition is set in point	Jump to specified address
	If the address condition is set in range	Jump to lower address (point address before the mask if a mask is specified)
	If the address condition is set in bit	Jump to address at the bit position

Target Window	Details of Jump Pointer	
From the <a href="#">Stack window</a>	A function at the cursor position that stack frame number indicates is used as the jump pointer.	
	<b>With current function</b>	
	If the jump destination is the <a href="#">Source window</a>	Jumps to the current PC line
	Other than above	Jumps to the current PC address
	<b>With function other than current function</b>	
	If the jump destination is the <a href="#">Source window</a>	Jump to the line that calls a nested function.
	Other than above	Jump to the address next to the instruction that calls a nested function.
From the <a href="#">Trace View window</a>	Jump to the <a href="#">Memory window</a>	
	If the cursor position is at an access address, access data, or access status	Access address
	Other than above	Fetch address
Jump to the <a href="#">Source window</a> or <a href="#">Assemble window</a>	Fetch address	

### 5.17.3 Trace result with linking window [IECUBE]

By linking the [Trace View window](#) with each window ([Source window](#), [Assemble window](#) or [Memory window](#)), the corresponding part can be displayed on the linked window, by using the address at the cursor position on the [Trace View window](#) as a pointer.

If the cursor is moved on the [Trace View window](#), the corresponding part on the linked window is highlighted or indicated by the cursor position.

#### (1) Linking method

The linking method is as follows:

- 1) Set the [Trace View window](#) as the current window.
- 2) Select [View] menu -> [Window Synchronize] to select a window to be linked.
- 3) Move the cursor to the line to be linked in the trace result display area of the [Trace View window](#).
- 4) Using the address of the line selected in 3) as a pointer, the corresponding part is highlighted (or indicated by the cursor position) in the display area of the window selected in 2).

**Remark:** The linking source address differs as follows depending on the cursor position in the trace result display area if the [Memory window](#) is linked.

- Access address, access data, access status -> Access address
- Others -> Fetch address

When the [Source window](#) or [Assemble window](#) is linked, the fetch address is always used as the pointer.

## 5.17.4 Drag & drop function

Selected and highlighted line numbers, addresses, and text can be dragged and dropped in another window using the following method.

- 1) Drag the selected line number, address, or text.
  - > The shape of the mouse cursor changes from an arrow to "-".
- 2) Drop the selection in a window or area where it can be dropped.
  - > The shape of the cursor changes from "-" to "OK" when the cursor is placed over a window or area where the selection can be dropped.

In the window in which the line number of the address has been dropped, an operation is performed on the dropped address or the address that is obtained from the dropped line number. For example, a variable can be simply registered by dragging and dropping in the [Watch window](#) such a variable located in the [Source window](#).

### (1) Drag & drop details

The operation to be performed after dropping the line number or address differs, depending on the window or area in which the line number or address has been dropped.

Table 5-24 Details of Drag & Drop Function (Line/Address)

Window/Area to Drop to	Operation After Drop
The <a href="#">Event manager</a> or the event manager area in each various event setting dialog box	Automatically creates an execution event condition by using the dropped line number or address as an address condition. Event condition names are automatically created as Evt00001, Evt00002, and so on. A path count is not specified. The address condition is set for the closest symbol in the format of symbol name + offset value.
Condition setting area in each various event setting dialog box (other than address and data setting areas)	Automatically creates an execution event condition by using the dropped line number or address as an address condition. The automatically created event condition is set in each condition setting area in which the line number or address has been dropped. Event condition names are automatically created as Evt00001, Evt00002, and so on. A path count is not specified. The address condition is set for the closest symbol in the format of symbol name + offset value.
Condition setting area in each various event setting dialog box (address and data setting areas)	The text of the dropped line number or address is set in the area in which the line number or address has been dropped. The address condition is set for the closest symbol in the format of symbol name + offset value.

Table 5-25 Details of Drag &amp; Drop Function (Character String)

Window/Area to Drop to	Operation After Drop												
The <a href="#">Event manager</a> or the event manager area in each various event setting dialog box	If the dropped text can be converted as a symbol into an address value, an event condition in the Access status (all access statuses) or Execute status is automatically created, using the converted address value as an address condition. Event condition names are automatically created as Evt00001, Evt00002, and so on. A data condition and path count are not specified. The address condition is set by the dropped text. The relationship between the event condition to be created and the symbol is as follows:												
	<table border="1"> <thead> <tr> <th>Symbols</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Variable</td> <td>Access (R/W)</td> </tr> <tr> <td>Function</td> <td>Execute</td> </tr> <tr> <td>Symbol in data section</td> <td>Access (R/W)</td> </tr> <tr> <td>Symbol in code section</td> <td>Execute</td> </tr> <tr> <td>Others</td> <td>Access (R/W)</td> </tr> </tbody> </table>	Symbols	Status	Variable	Access (R/W)	Function	Execute	Symbol in data section	Access (R/W)	Symbol in code section	Execute	Others	Access (R/W)
	Symbols	Status											
	Variable	Access (R/W)											
	Function	Execute											
	Symbol in data section	Access (R/W)											
	Symbol in code section	Execute											
Others	Access (R/W)												
Variable	Access (R/W)												
Function	Execute												
Symbol in data section	Access (R/W)												
Symbol in code section	Execute												
Others	Access (R/W)												
Condition setting area in each various event setting dialog box (other than address and data setting areas)	If the dropped text can be converted as a symbol into an address value, an event condition in the Access status (all access statuses) or Execute status is automatically created, using the converted address value as an address condition. The automatically created event condition is set in each condition setting area in which the line number or address has been dropped. Event condition names are automatically created as Evt00001, Evt00002, and so on. A data condition and path count are not specified. The address condition is set by the dropped text. The relationship between the event condition to be created and the symbol is as follows:												
	<table border="1"> <thead> <tr> <th>Symbols</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Variable</td> <td>Access (R/W)</td> </tr> <tr> <td>Function</td> <td>Execute</td> </tr> <tr> <td>Symbol in data section</td> <td>Access (R/W)</td> </tr> <tr> <td>Symbol in code section</td> <td>Execute</td> </tr> <tr> <td>Others</td> <td>Access (R/W)</td> </tr> </tbody> </table>	Symbols	Status	Variable	Access (R/W)	Function	Execute	Symbol in data section	Access (R/W)	Symbol in code section	Execute	Others	Access (R/W)
	Symbols	Status											
	Variable	Access (R/W)											
	Function	Execute											
	Symbol in data section	Access (R/W)											
	Symbol in code section	Execute											
Others	Access (R/W)												
Variable	Access (R/W)												
Function	Execute												
Symbol in data section	Access (R/W)												
Symbol in code section	Execute												
Others	Access (R/W)												
Condition setting area in each various event setting dialog box (address and data setting areas)	The dropped text is set in the area.												
<a href="#">Watch window</a>	If the dropped text is recognizable as a symbol, the contents of the symbol are displayed.												

**Remark:** Each various event setting dialog box are as follows.

- [Event dialog box](#)
- [Event Link dialog box](#)
- [Break dialog box](#)
- [Trace dialog box](#)
- [Timer dialog box](#)
- [Snap Shot dialog box](#)
- [Event DMM dialog box](#)

### 5.17.5 Cautions

- (1) The number of characters that can be displayed on 1 line in each area of a window is 319.
- (2) If the width of the display area is narrow, the display may become corrupted. In this case, increase the width of the window.
- (3) Redrawing may not successfully be performed in a window with a [Refresh] button when the cursor position is changed while the window is active. Click the [Refresh] button to perform redrawing.
- (4) The help that is opened using the F1 key is the help corresponding to the window on which the cursor is placed. Consequently, because the cursor cannot be placed on the [Trace View window](#) in which no trace results are displayed, such as immediately after startup, the help may not open even if the F1 key is pressed. In this case, open the help by selecting [Current Window Help] from the [Help] menu.
- (5) Do not select [Slowmotion] from the [Run] menu during Go & Go execution. [Slowmotion] on the [Run] menu is usually dimmed during Go & Go execution, but there is a moment when it can be selected, so if [Slowmotion] is selected at this time, the program will not be able to be stopped even if [Stop] is selected from the [Run] menu (or the STOP button is clicked).
- (6) If for some reason or other the application switches while event icons are in the process of being dragged, the icons will no longer be able to be dropped.  
Use the ESC key to escape from drag, then reattempt the drag.

## Main window

This window is automatically opened when the ID78K0R-QB is started up and initialized.

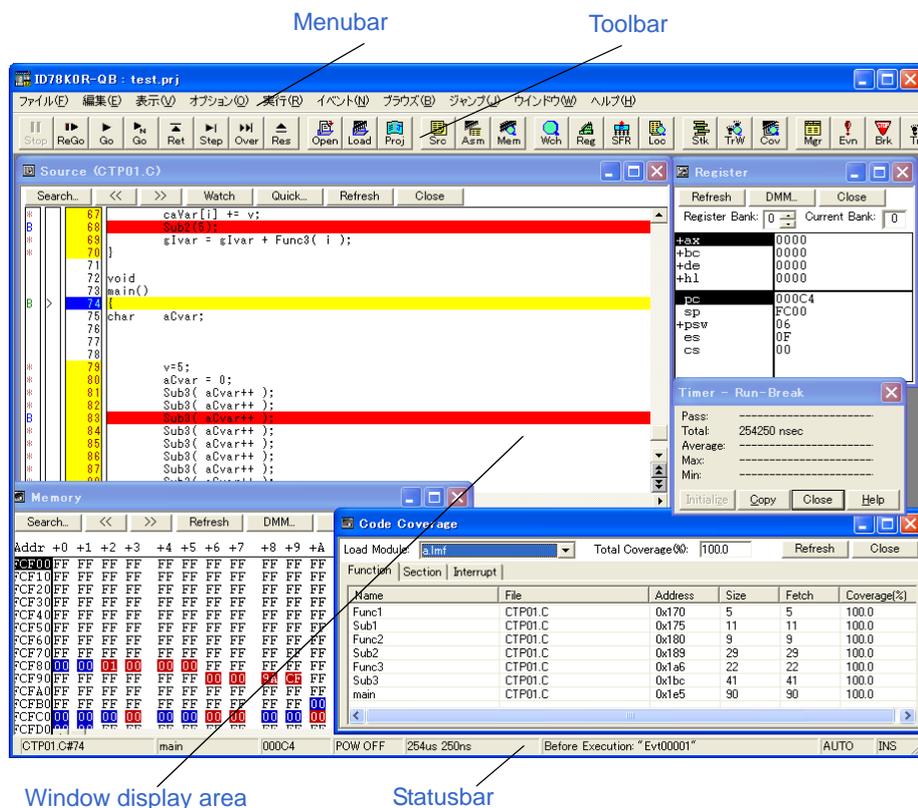
In the ID78K0R-QB, other windows are manipulated from this window. (Refer to "6.1 Window List".)

Execution of the user program is controlled in this window.

Execution of the user program is controlled in the following three modes:

- Source mode (Debugs the user program at the source level.)
- Instruction mode (Debugs the user program at the instruction level.)
- Auto mode (Automatically selects the source mode or instruction mode.) (default)

Figure 5-28 Main Window



Window display area

Statusbar

## Menubar

- (1) [File] menu
- (2) [Edit] menu
- (3) [View] menu
- (4) [Option] menu
- (5) [Run] menu
- (6) [Event] menu
- (7) [Browse] menu
- (8) [Jump] menu
- (9) [Window] menu
- (10) [Help] menu

### (1) [File] menu

Open...	Loads a view file, source file, or text file. Opens the <a href="#">View File Load dialog box</a> . The operation differs depending on the extension of the file selected in the dialog box.
Save As...	Saves the contents displayed on the current window to the file whose name is specified. Opens the <a href="#">View File Save dialog box</a> .
Close	Closes the current window.
Download...	Downloads a file. Opens the <a href="#">Download dialog box</a> .
Load Module...	Lists the names of the files that have been downloaded. Opens the <a href="#">Load Module List dialog box</a> .
Upload...	Uploads a program. Opens the <a href="#">Upload dialog box</a> .
Project	Manipulates a project file.
Open...	Opens a project file. Opens the <a href="#">Project File Load dialog box</a> .
Save	Overwrites the current status to the project file currently being read to the ID78K0R-QB.
Save As...	Saves the current status to a specified project file. Opens the <a href="#">Project File Save dialog box</a> .
Environment	Manipulates a setting file.
Open...	Opens a setting file. Opens the <a href="#">Environment Setting File Load dialog box</a> .
Save As...	Saves the setting in the current window to the setting file. Opens the <a href="#">Environment Setting File Save dialog box</a> .
Debugger Reset...	Initializes the CPU, symbols, and the ID78K0R-QB. Opens the <a href="#">Reset Debugger dialog box</a> .
Exit	Terminate the ID78K0R-QB. (Refer to " <a href="#">3.4 Terminating</a> ".) Opens the <a href="#">Exit Debugger dialog box</a> .
<i>Open file</i>	Lists the names of the files opened.

**(2) [Edit] menu**

Cut	Cuts a selected character string and saves it to the clipboard buffer.
Copy	Copies a selected character string and saves it to the clipboard buffer.
Paste	Pastes the contents of the clipboard buffer to the text cursor position.
Write in	Writes the modified contents to the target.
Restore	Cancel the modification.
Memory	Manipulates the memory contents.
Fill...	Fills the memory contents with specified codes. Opens the <a href="#">Memory Fill dialog box</a> .
Copy...	Copies the memory. Opens the <a href="#">Memory Copy dialog box</a> .
Compare...	Compares the memory. Opens the <a href="#">Memory Compare dialog box</a> .
DMM...	Rewrites the memory contents in real time during user program execution. Opens the <a href="#">DMM dialog box</a> .
Edit Source	Opens the source file displayed in the active <a href="#">Source window</a> with the editor specified by the PM+ when the PM+ runs.

**(3) [View] menu**

The [View] menu contains common parts as well as dedicated parts added according to the active window. For details about the dedicated parts, refer to the description of each window.

**(a) Common items**

Search...	Performs a search. Opens the search dialog box corresponding to the current window. Same operation as the [Search] button.
Move...	Moves the display position. Opens the specification dialog box corresponding to the current window.
Quick Watch...	Temporarily displays the contents of the specified data. Opens the <a href="#">Quick Watch dialog box</a> .
Add Watch...	Registers the specified data to the Watch window. Opens the <a href="#">Add Watch dialog box</a> .
View Watch	Adds the selected data to the Watch window. If the data is a symbol, it is added in accordance with the setting of the <a href="#">Debugger Option dialog box</a> .
Change Watch...	Changes the data on the line selected by the Watch window. Opens the <a href="#">Change Watch dialog box</a> . This menu is valid only when a variable is selected in the <a href="#">Watch window</a> .
Delete Watch	Deletes the selected watch point from the <a href="#">Watch window</a> . This menu is valid only when a variable is selected in the <a href="#">Watch window</a> .
Symbol...	Displays the address of the specified variable or function, or the value of the specified symbol. Opens the <a href="#">Symbol To Address dialog box</a> .

**(4) [Option] menu**

Tool Bar	Selects whether the toolbar is displayed (default) or not.
Status Bar	Selects whether the statusbar is displayed (default) or not.
Button	Selects whether the buttons on each window are displayed (default) or not.
Source Mode	Executes step execution at the source level (in line units).
Instruction Mode	Executes step execution at the instruction level (in instruction units).
Auto Mode	Automatically selects step execution at the source level or step execution at the instruction level (default). Step execution is performed at the source level (in a mode other than mixed display mode) if the <a href="#">Source window</a> is active. It is performed at the instruction level if the <a href="#">Assemble window</a> is active. If neither window is active, step execution is performed at the source level.
Configuration...	Sets the environment. Opens the <a href="#">Configuration dialog box</a> .
Extended Option...	Sets extended options. Opens the <a href="#">Extended Option dialog box</a> .
Flash Option ... [IECUBE]	This dialog box is used to make the flash self programming emulation settings. Opens the <a href="#">Flash Option dialog box</a> .
DataFlash Option... [IECUBE]	This dialog box is used to make the DataFlash error emulation settings. Opens the <a href="#">DataFlash Option dialog box</a> .
Debugger Option...	Sets the ID78K0R-QB options. Opens the <a href="#">Debugger Option dialog box</a> .
Add I/O Port...	Adds user-defined I/O ports. Opens the <a href="#">Add I/O Port dialog box</a> .
Trace Clear [IECUBE]	Clears the trace data. This item is displayed only when <a href="#">Trace View window</a> is active.
Coverage Clear [IECUBE]	Clears the coverage measurement results.

**(5) [Run] menu**

Restart	Resets the CPU and executes the program. Same operation as the  button.
Stop	Forcibly stops program execution. Same operation as the  button.
Go	Executes the program from the current PC. Same operation as the  button.
Ignore break points and Go	Ignores break points being set, and executes the program. (Both hard and soft.) Same operation as the  button.
Return Out	The user program is executed until execution returns. Same operation as the  button. <b>Caution:</b> This command is used for a function described in C language.

Step In	Executes the instructions in the program one by one (step execution). If a function or subroutine is called, its instructions are executed one by one.  Same operation as the  button.
Next Over	Executes the instructions in the program one by one (Next step execution). If a function or subroutine is called, its instructions are not executed on a step-by-step basis.  Same operation as the  button.
Start From Here	Executes the program from the cursor position on <a href="#">Source window</a> or <a href="#">Assemble window</a> .
Come Here	Executes the program from the current PC to the cursor position in the <a href="#">Source window</a> or <a href="#">Assemble window</a> .
Go & Go	Continues executing the program. If a break occurs because a break condition is satisfied, the window is updated and the program is executed again.  Same operation as clicking the  button each time a break has occurred.
Slowmotion	Continues step execution. Each time step execution has been performed, the window is updated and then step execution is performed again.  Same operation as clicking the  button each time a break has occurred.
CPU Reset	Resets the CPU.  Same operation as the  button.
Change PC	Sets the address at the cursor position in the <a href="#">Source window</a> or <a href="#">Assemble window</a> to the PC.
Break Point [IECUBE]	Sets or deletes a breakpoint at the cursor position in the <a href="#">Source window</a> or <a href="#">Assemble window</a> .
Software Break Point	Sets or deletes a software breakpoint at the cursor position in the <a href="#">Source window</a> or <a href="#">Assemble window</a> .
Delete All Breakpoints	Deletes all the set break events.
Uncond. Trace ON [IECUBE]	Validates unconditional trace so that trace can always be executed during program execution. (default) At this time, the set trace event conditions are ignored.
Cond. Trace ON [IECUBE]	Validates conditional trace and traces in accordance with the trace event condition during program execution.
Tracer Control Mode [IECUBE]	To set trace control mode. <b>Caution:</b> Delay trigger event condition is enabled only when [Delay Trigger Stop] or [Delay Trigger Break] is selected. In such a case, switching to another mode is impossible.
Non Stop	Goes around the trace memory and overwrites data from the oldest frame (default).
Full Stop	Goes around the trace memory and then stops the tracer.
Full Break	Goes around the trace memory and then stops the tracer and program execution.
Delay Trigger Stop	Traces data by the number of delay count frames and stops the tracer when a delay trigger event has occurred.
Delay Trigger Break	Traces data by the number of delay count frames and stops the tracer and program execution when a delay trigger event has occurred.

Timer Start/Timer Stop <b>[IECUBE]</b>	Starts timer measurement when it is stopped, or stops it when it is in progress. This item is invalid if the program is not being executed and if a timer event is not used. Immediately after program execution has been started, timer measurement is in progress.
Tracer Start/Tracer Stop <b>[IECUBE]</b>	Starts the tracer when it is stopped, or stops it when it is in progress. This item is invalid if the program is not being executed. Immediately after program execution has been started, the tracer is executed.
Pseudo Emulation... <b>[IECUBE]</b>	Opens the <a href="#">Pseudo Emulation dialog box</a> .

**(6) [Event] menu**

Event Manager	Manages various event conditions. Opens the <a href="#">Event manager</a> . Same operation as the  button.
Software Break Manager	Manages software break event conditions. Opens the <a href="#">Software Break manager</a> .
Event...	Registers an event condition. Opens the <a href="#">Event dialog box</a> . Same operation as the  button.
Event Link... <b>[IECUBE]</b>	Registers an event link condition. Opens the <a href="#">Event Link dialog box</a> .
Break...	Registers and sets a break event condition. Opens the <a href="#">Break dialog box</a> . Same operation as the  button.
Trace... <b>[IECUBE]</b>	Registers and sets a trace event condition. Opens the <a href="#">Trace dialog box</a> . Same operation as the  button.
Snapshot... <b>[IECUBE]</b>	Registers and sets a snap event condition. Opens the <a href="#">Snap Shot dialog box</a> .
Timer... <b>[IECUBE]</b>	Registers and sets a timer event condition. Opens the <a href="#">Timer dialog box</a> . Same operation as the  button.
Event DMM... <b>[IECUBE]</b>	Registers and sets an event DMM condition. Opens the <a href="#">Event DMM dialog box</a> .
Delay Count... <b>[IECUBE]</b>	Sets the delay count. Opens the <a href="#">Delay Count dialog box</a> .

**(7) [Browse] menu**

List	Lists functions, variables, symbols, sections, and interrupt requests. Opens the <a href="#">List window</a> .
Source Text	Displays a source text. Opens the <a href="#">Source window</a> . If there is this window already open in the active status, it is opened in the static status. Same operation as the  button.
Assemble	Displays the disassemble results. Opens the <a href="#">Assemble window</a> . If there is this window already open in the active status, it is opened in the static status. Same operation as the  button.

Memory	<p>Displays the contents of the memory. Opens the <a href="#">Memory window</a>. If there is this window already open in the active status, it is opened in the static status.</p> <p>Same operation as the  button.</p>
Watch	<p>Displays the watch contents. Opens the <a href="#">Watch window</a>.</p> <p>Same operation as the  button.</p>
Register	<p>Displays the register contents. Opens the <a href="#">Register window</a>.</p> <p>Same operation as the  button.</p>
SFR	<p>Displays the SFR. Opens the <a href="#">SFR window</a>.</p> <p>Same operation as the  button.</p>
Local Variable	<p>Displays the local variable. Opens the <a href="#">Local Variable window</a>.</p> <p>Same operation as the  button.</p>
Stack Trace	<p>Displays the stack trace results. Opens the <a href="#">Stack window</a>.</p> <p>Same operation as the  button.</p>
Trace [IECUBE]	<p>This area displays the trace results. Opens the <a href="#">Trace View window</a>.</p> <p>Same operation as the  button.</p>
Code Coverage [IECUBE]	<p>Displays code coverage measurement results. Opens the <a href="#">Code Coverage window</a>.</p> <p>Same operation as the  button.</p>
Console	<p>Opens the <a href="#">Console window</a>.</p>
Others	<p>Displays other windows. (Refer to "<a href="#">APPENDIX A EXPANSION WINDOW</a>".) Displays a user-defined window list.</p>

**(8) [Jump] menu**

Source Text	<p>Displays the corresponding source text and source line, using the data value selected in the current window as the jump destination address. If no line information exists at the jump destination address, however, you cannot jump. Opens the <a href="#">Source window</a>. If the <a href="#">Source window</a> in active is open, that window is displayed in the forefront (so that it can be manipulated).</p>
Assemble	<p>Disassembles and displays the results from the jump destination address specified by the data value selected in the current window. Opens the <a href="#">Assemble window</a>. If the <a href="#">Assemble window</a> in active is open, that window is displayed in the forefront (so that it can be manipulated).</p>
Memory	<p>Displays the memory contents from the jump destination address specified by the data value selected in the current window. Opens the <a href="#">Memory window</a>. If the <a href="#">Memory window</a> in active is open, that window is displayed in the forefront (so that it can be manipulated).</p>

**(9) [Window] menu**

New Window	Opens a new window displaying the same contents as those of the current window. This menu is valid only when the current window is <a href="#">Source window</a> , <a href="#">Assemble window</a> or <a href="#">Memory window</a> .
Cascade	Cascade display of the windows in the Main window.
Tile	Cascade display of the windows in the Main window.
Arrange Icons	Rearranges the icons in the Main window.
Close All	Closes all windows, except the Main window.
Refresh	Updates the contents of the window with the latest data.
Active	Sets the window in the active status.
Static	Sets the window in the static status.
<i>Open Window</i>	Lists the windows that are open. The window with the check mark shown on the side of the figure is the current window. By selecting a window name, the selected window is used as the current window.

**(10) [Help] menu**

ID78K0R-QB Help	Displays the help.
Command Reference	Opens the Help window of <a href="#">COMMAND REFERENCE</a> .
Main Window	Displays the help of the Main window.
Current Window	Displays the help of the current window.
About...	Displays the version of the ID78K0R-QB. Opens the <a href="#">About dialog box</a> .

**Toolbar**(1) [Meaning of each button](#)(2) [Operation of toolbar](#)**(1) Meaning of each button**

The meaning of each button on the toolbar is as follows. When the mouse cursor is placed on a button of the toolbar, a tool hint pops up several seconds later.

 <b>Stop</b>	Stops execution of the user program. Same function as [Run] menu -> [Stop].
 <b>ReGo</b>	Resets the CPU and executes the user program. Same function as [Run] menu -> [Restart].
 <b>Go</b>	Executes the user program from the current PC without resetting the CPU. Same function as [Run] menu -> [Go].
 <b>GoN</b>	Ignores break points being set, and executes the user program. Same function as [Run] menu -> [Ignore break points and Go].

 Ret	The user program is executed until execution returns Same function as [Run] menu - [Return Out]. <b>Caution:</b> This command is used for a function described in C language.
 Step	Step execution (executes instructions in the program one by one.) If a function or subroutine is called, its instructions are executed one by one. Same function as [Run] menu -> [Step In].
 Over	Next step execution (executes the program, assuming a function/call statement as one step.) If a function or subroutine is called, its instructions are not executed on a step-by-step basis. Same function as [Run] menu -> [Next Over].
 Res	Resets the CPU. Same function as [Run] menu -> [CPU Reset].
 Open	Opens the <a href="#">View File Load dialog box</a> . Same function as [File] menu -> [Open].
 Load	Opens the <a href="#">Download dialog box</a> . Same function as [File] menu -> [Download].
 Proj	Opens the <a href="#">Project File Load dialog box</a> . Same function as [File] menu -> [Project] -> [Open].
 Src	Displays the source text. Opens the <a href="#">Source window</a> . Same function as [Browse] menu -> [Source Text].
 Asm	Displays the disassemble results. Opens the <a href="#">Assemble window</a> . Same function as [Browse] menu -> [Assemble].
 Mem	Displays the contents of the memory. Opens the <a href="#">Memory window</a> . Same function as [Browse] menu -> [Memory].
 Wch	Displays the watch contents. Opens the <a href="#">Watch window</a> . Same function as [Browse] menu -> [Watch].
 Reg	Displays the register contents. Opens the <a href="#">Register window</a> . Same function as [Browse] menu -> [Register].
 SFR	Displays the contents of SFR. Opens the <a href="#">SFR window</a> . Same function as [Browse] menu -> [SFR].
 Loc	Displays the local variable contents. Opens the <a href="#">Local Variable window</a> . Same function as [Browse] menu -> [Local Variable].
 Stk	Displays the stack trace results. Opens the <a href="#">Stack window</a> . Same function as [Browse] menu -> [Stack Trace].
 TrW [IECUBE]	This area displays the trace results. Opens the <a href="#">Trace View window</a> . Same function as [Browse] menu -> [Trace].
 Cov [IECUBE]	Displays code coverage measurement results. Opens the <a href="#">Code Coverage window</a> . Same function as [Browse] menu -> [Code Coverage].
 Mgr	Opens the <a href="#">Event manager</a> . Same function as [Event] menu -> [Event Manager].
 Evn	Registers and sets events. Opens the <a href="#">Event dialog box</a> . Same function as [Event] menu -> [Event].
 Brk	Registers and sets break events. Opens the <a href="#">Break dialog box</a> . Same function as [Event] menu -> [Break].
 Trc [IECUBE]	Registers and sets trace events. Opens the <a href="#">Trace dialog box</a> . Same function as [Event] menu -> [Trace].

 <b>Tim</b> <b>[IECUBE]</b>	Registers and sets timer events. Opens the <a href="#">Timer dialog box</a> . Same function as [Event] menu -> [Timer].
-----------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------

**(2) Operation of toolbar**

Whether the toolbar is displayed or not can be specified by selecting [Option] menu -> [Tool Bar].

This toolbar can be displayed in the following two modes. The modes are selected in the [Debugger Option dialog box](#).

Figure 5-29 Toolbar (Picture Only)



Figure 5-30 Toolbar (Picture and Text)



**Window display area**

This area displays various debug windows.

The displayed window can be changed in size or an icon can be created in this area.

**Statusbar**

The statusbar displays the status of the ID78K0R-QB and in-circuit emulator.

While the user program is being executed, the statusbar is displayed in **red**.

Whether the toolbar is displayed or not can be specified by selecting [Option] menu -> [Status Bar].

**Remark:** If the screen resolution is 800 - 600 or lower, some statuses are not be displayed.  
 With 800 - 600, for example, the enclosed area is (7) and (8) in the following figure.

Figure 5-31 Statusbar



(1) Program name	Displays the program file name indicated by the PC value. <b>Note</b>
Source name	Displays the source file name indicated by the PC value. <b>Note</b>
Line number	Displays the line number indicated by the PC value. <b>Note</b>
(2) Function name	Displays the function name indicated by the PC value. <b>Note</b>
(3) PC value	Displays the current PC value. <b>Note</b>
(4) CPU status	Refer to " <a href="#">Table 5-26 CPU Status</a> ".

(5) IE status	Refer to "Table 5-27 IE Status". (If there are two or more the statuses, they delimited with ' ' and displayed.)
(6) Break Cause	Refer to "Table 5-28 Break Cause".
(7) STEP mode	Displays the step execution mode. Displays that the following modes are selected from the [Option] menu: SRC:..... Source mode INST:..... Instruction mode AUTO: ..... Automatic mode
(8) Key input mode	Displays the key input mode. INS: ..... Insertion mode OVR:..... Overwrite mode The <a href="#">Memory window</a> is fixed to OVR mode.

**Note:** These are displayed in real time, even during user program execution. **[IECUBE]**

Table 5-26 CPU Status

Display	Meaning
HALT	Halt mode
STOP	Stop mode
WAIT	Wait mode
RESET	Reset mode
POW OFF	Power is not supplied to the target

Table 5-27 IE Status

Display	Meaning
RUN	User program execution in progress (the color of the statusbar changes)
STEP	Step execution in progress
TRC	Tracer operating
TIM	Timer operating
BREAK	Break occurring
<i>Time</i>	Displays the result of measuring the time from the start of user program execution to the occurrence of break. (run-break time) <b>Note</b>
TIMER OVERFLOW	Measurement result overflowed. The measurement result is not accurate during execution of Step In, Next Over, or Return Out. <b>[MINICUBE2]</b>

**Note:** The measurable range is 17 ns (min., with undivided clock) to 40 hours 43 minutes (max., with clock divided by 2K). **[IECUBE]**

The measurable range is 100 us (min.) to 119 hours 18 minutes (max.). **[MINICUBE2]**

Table 5-28 Break Cause

Display	Meaning
Manual Break	Forced break
Temporary Break	Temporary break
Software Break	Software break
Trace Full Break	Break due to trace full
Trace Delay Break	Break by trace delay
Non Map Break	Non-mapped area is accessed.
Write Protect	An attempt has been made to write to a write-protected area
SFR Write Protect	Read from prohibited SFR was attempted
SFR Read Protect	Write to write-prohibited SFR was attempted
Stack Overflow	Break by stack overflow
Stack Underflow	Break by stack underflow
Uninitialize Stack Pointer	Break due to failure to perform stack pointer initialization
Uninitialize Memory Read	Memory not initialized has been read
Timer Over Break	Execution time-over detected
Fetch Break	Guarded area or fetch prohibited area was fetched
Unspecified Illegal	Illegal action of user program related to peripheral chip function occurred Refer to documentation on peripheral emulation board.
Flash Illegal	Flash Illegal break
Peripheral Break	Break from peripheral
Step Break	Step execution break
Event Break: " <i>Event Name</i> "	Stopped due to event cause of displayed event name
Before Execution: " <i>Event Name</i> "	A break before execution caused by the event cause <i>Event Name</i>
Word Miss-align Access	A word access for an odd number address was made

# CHAPTER 6 WINDOW REFERENCE

This chapter explains in detail the functions of the windows and dialog boxes of ID78K0R-QB.

- [Window List](#)
- [Explanation of Windows](#)

## 6.1 Window List

The list is the windows of the ID78K0R-QB.

Table 6-1 Window List

Window Name	Contents
<a href="#">Main window</a>	This window is displayed first, when the ID78K0R-QB is started. It controls execution of the user program. Various windows are opened from this window.
<a href="#">Configuration dialog box</a>	Displays and sets the ID78K0R-QB operation environment.
<a href="#">Extended Option dialog box</a>	Displays and sets the extended options of the ID78K0R-QB.
<a href="#">Fail-safe Break dialog box</a>	Sets the fail-safe breaks. <b>[IECUBE]</b>
<a href="#">Flash Option dialog box</a>	Sets the the flash self programming emulation. <b>[IECUBE]</b>
<a href="#">DataFlash Option dialog box</a>	Sets the DataFlash error emulation. <b>[IECUBE]</b>
<a href="#">Debugger Option dialog box</a>	Displays and sets other options.
<a href="#">Pseudo Emulation dialog box</a>	Generation of pseudo emulation. <b>[IECUBE]</b>
<a href="#">Project File Save dialog box</a>	Saves the current debug environment to project file.
<a href="#">Project File Load dialog box</a>	Loads the debug environment.
<a href="#">Download dialog box</a>	Downloads.
<a href="#">Upload dialog box</a>	Uploads.
<a href="#">Load Module List dialog box</a>	Lists the names of the downloaded load module files.
<a href="#">Source window</a>	Displays a source file and text file.
<a href="#">Source Search dialog box</a>	Searches in the <a href="#">Source window</a> .
<a href="#">Source Text Move dialog box</a>	Specifies a file to be displayed in the <a href="#">Source window</a> and the position from which displaying the file is to be started.
<a href="#">Assemble window</a>	Disassembles the program and executes online assembly.
<a href="#">Assemble Search dialog box</a>	Searches in the <a href="#">Assemble window</a> .
<a href="#">Address Move dialog box</a>	Specifies the start address to display the contents of the <a href="#">Memory window</a> , <a href="#">Assemble window</a> or <a href="#">SFR window</a> .

Window Name	Contents
Symbol To Address dialog box	Displays the address of the specified variable or function, or the value of the specified symbol.
List window	Lists functions, variables, symbols, sections, and interrupt requests.
Watch window	Displays and changes specified watch data.
Quick Watch dialog box	Displays temporarily specified watch data.
Add Watch dialog box	Registers watch data to display in the <a href="#">Watch window</a> .
Change Watch dialog box	Changes watch data to display in the <a href="#">Watch window</a> .
Local Variable window	Displays and changes the local variable in the current function.
Stack window	Displays the current stack contents.
Memory window	Displays the contents of memory.
Memory Search dialog box	Searches in the <a href="#">Memory window</a> .
Memory Fill dialog box	Fills the memory contents with specified data.
Memory Copy dialog box	Copies the memory.
Memory Compare dialog box	Compares the memory.
Memory Compare Result dialog box	Displays the results of comparing the memory.
DMM dialog box	Sets addresses and data subject to DMM.
Register window	Displays the contents of registers.
SFR window	Displays the contents of SFR.
SFR Select dialog box	Selects SFR and I/O ports to be displayed in the <a href="#">SFR window</a> .
Add I/O Port dialog box	Registers an I/O port to be displayed in the <a href="#">SFR window</a> .
Timer dialog box	Registers and sets timer event conditions, and displays execution time measurement result. <b>[IECUBE]</b>
Timer Result dialog box	Displays execution time measurement results. <b>[IECUBE]</b>
Trace View window	Displays trace results. <b>[IECUBE]</b>
Trace Search dialog box	Searches trace data. <b>[IECUBE]</b>
Trace Data Select dialog box	Selects items to be displayed in the <a href="#">Trace View window</a> . <b>[IECUBE]</b>
Trace Move dialog box	Specifies the start address to display the contents of the <a href="#">Trace View window</a> . <b>[IECUBE]</b>
Trace dialog box	Registers and sets trace event conditions. <b>[IECUBE]</b>
Delay Count dialog box	Sets the delay count of a delay trigger trace event. <b>[IECUBE]</b>
Code Coverage window	Displays the coverage results. <b>[IECUBE]</b>
Software Break manager	Displays, enables/disables, and deletes software breaks.
Event manager	Displays, enables/disables, and deletes each event condition.
Event dialog box	Registers event conditions.
Event Link dialog box	Registers event link conditions. <b>[IECUBE]</b>

Window Name	Contents
Break dialog box	Registers and sets break event conditions.
Snap Shot dialog box	Registers and sets snapshot event conditions. <b>[IECUBE]</b>
Event DMM dialog box	Registers and sets event DMM conditions. <b>[IECUBE]</b>
View File Save dialog box	Saves the display information of the current window to a view file.
View File Load dialog box	Loads the view file of each window.
Environment Setting File Save dialog box	Saves the setting information of the current window to a setting file.
Environment Setting File Load dialog box	Loads the setting file of each window.
Reset Debugger dialog box	Initializes the ID78K0R-QB, CPU, and symbol information.
Exit Debugger dialog box	Terminate the ID78K0R-QB.
About dialog box	Displays the version of the ID78K0R-QB.
Console window	Inputs commands.
Browse dialog box	Selects the file to be set.

## 6.2 Explanation of Windows

This section explains each window or dialog box as follows:

### Window Name / Dialog Box Name

Briefly explains the function of the window or dialog box and points to be noted.

If an invalid window/dialog box exists due to a connected IE, the name of the valid connected IE is indicated at the lower right of the window/dialog box name.

In addition, the display image of the window or dialog box is also illustrated.

Items of related operation are also explained.

### Opening

Explains how to open the window or dialog box.

### Explanation of each area

Explains items to be set to or displayed in each area of the window or dialog box.

## **Context menu**

---

Explains the context menu that is displayed in the window when the right mouse button is clicked. From the context menu, convenient functions often used in this window can be selected with a single action (window only).

## **Related operations**

---

Explains the operation of a window or dialog box related to this window or dialog box.

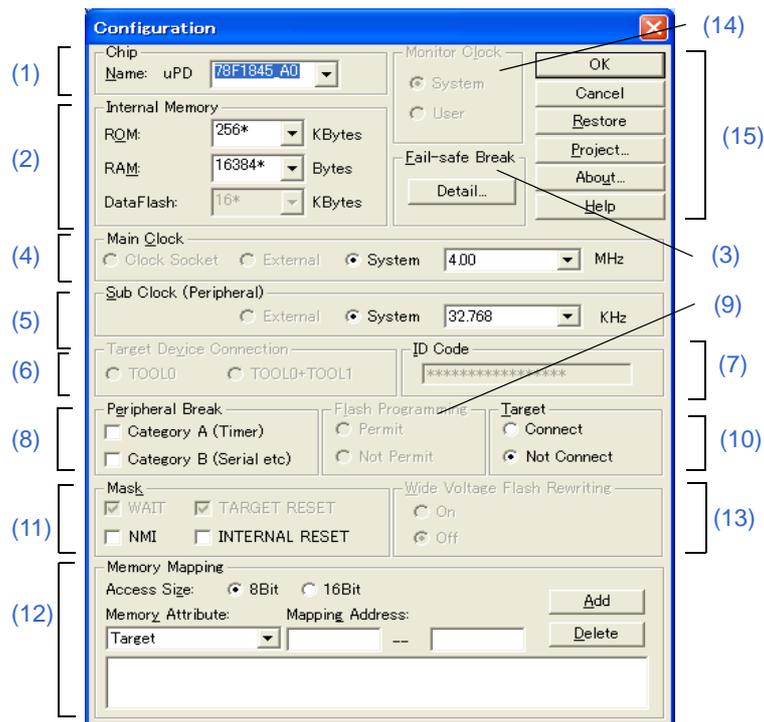
## Configuration dialog box

This dialog box is used to display and set the ID78K0R-QB operation environment. (Refer to "5.1 Setting Debugging Environment".)

This dialog box is automatically displayed after the ID78K0R-QB is started up.

However, no setting is required to read a project as the results of reading the project file are reflected in this dialog box. (Refer to "5.16.1 Debugging environment (project file)".)

Figure 6-1 Configuration Dialog Box



- Opening
- Explanation of each area

## Opening

(Automatically when the ID78K0R-QB is started up)

Select [Option] menu -> [Configuration...].

## Explanation of each area

### (1) Chip

This area is used to select the chip name. A chip name is selected from the drop-down list.

On the drop-down list, only the chip names registered to the registry from the device file installer are displayed.

This area can be specified only when the debugger is started up. (After the debugger is started, this area appears dimmed and the setting cannot be changed.)

If a project file to which a different chip is specified is loaded after the debugger startup, the chip name specified in the project file is ignored.

**Remark:** By default, the type selected at the previous startup is displayed, but if that type is not registered, the first type registered is displayed.

### (2) Internal Memory

This area is used to set the size of the internal RAM and internal ROM of the CPU. It is selected from the drop-down list, or input from the key board. **[IECUBE]**

The size of the data flash area in the device is displayed in this area.

The default size is obtained from the device file through selection in "Chip" , and displayed (value with "\*").

**Remark:** This area is fixed to device file value. **[MINICUBE2]**

Internal ROM:	Selects the Internal ROM size. Values can also be input from the keyboard in 1-byte unit (0 to 960 KB).
Internal RAM:	Selects the Internal RAM size. Values can also be input from the keyboard in 128-byte units (128 to 63,232 bytes). The fractions are rounded up.

### (3) Fail-safe Break **[IECUBE]**

This area is used to select a fail-safe break function.

Clicking the [Detail...] button opens the [Fail-safe Break dialog box](#), so that the fail-safe break function can be individually set.

### (4) Main Clock

This area is used to select a clock source input to the main CPU (main CPU clock).

Clock Socket	The clock connected to the clock socket is used as the main CPU clock (this item is shaded if no oscillator is connected).
External	The main clock (rectangular wave) on the target system is used as the main CPU clock (this item is shaded if the main clock on the target system is not detected by the debugger).
System	The clock generated in the in-circuit emulator is used as the main CPU clock. The selectable frequencies are as follows. <b>[IECUBE]</b> 2.00, 3.00, 3.57, 4.00 (default), 4.19, 4.91, 5.00, 6.00, 8.00, 8.38, 10.00,12.00, 16.00, 20.00

**Caution1:** The setting is fixed to "External". **[MINICUBE2]**

Input the correct clock frequency into the combo box.

Select one of the following items, or input a value from 0.001 to 99.999 (to three decimal places).

None (clock not used) (default), 2.00, 3.00, 3.57, 4.00, 4.19, 4.91, 5.00, 6.00, 8.00, 8.38, 10.00, 12.00, 16.00, 20.00

**Caution2:** For products that include a PLL, be sure to specify the pre-PLL frequency (that is, the frequency of the oscillator or resonator before being multiplied by the PLL.)

The default setting varies depending on whether the oscillator is used and whether the power supply of the target system is detected, as follows. (O: Selectable, x: Not selectable, Default: default setting). **[IECUBE]**

	Oscillator Used		Oscillator Not Used	
	Power Detected on Target System	Power Not Detected on Target System	Power Detected on Target System	Power Not Detected on Target System
Clock Socket	Default	Default	x	x
External	O	x	Default	x
System	O	O	O	Default

#### (5) Sub Clock (Peripheral)

This area is used to select a clock source input to the sub CPU and the peripheral equipment (sub CPU clock).

The default setting is "External" when the power supply on the target system is ON, and "System" when the power supply on the target system is OFF. **[IECUBE]**

External	The main clock (rectangular wave) on the target system is used as the sub CPU clock.
System	The clock generated in the in-circuit emulator is used as the sub CPU clock. The selectable frequencies are as follows. <b>[IECUBE]</b> 32.768 (default), 38.400

**Caution:** The setting is fixed to "External". **[MINICUBE2]**

Input the correct clock frequency into the combo box.

Select one of the following items, or input a value from 0.001 to 99.999 (to three decimal places).

None (clock not used) (default), 32.768, 38.400

**(6) Target Device Connection [MINICUBE2]**

This area is used to select the port to be connected for serial communication between MINICUBE2 and the device on the target system.

The 1-wire mode (TOOL0) and 2-wire mode (TOOL0 + TOOL1) are supported as the communication interface.

The type of port that can be selected depends on the device used.

This area can be specified only when the debugger is started up.

If the 1-wire mode (TOOL0) is selected, the RRM function can be set to the 1-wire mode (TOOL0) in the Extend Option dialog box. For details, refer to the descriptions of Extended Option dialog box.

**(7) ID Code [MINICUBE2]**

This area does not have to be set with a ROMless product or a product without a OCD (on-chip debug unit).

This area is used to input the ID code to be used when the code on the internal ROM or internal flash memory is read by ID78K0R-QB (ID code authentication).

Input a hexadecimal number of 20 digits (10 bytes) as the ID code (all 'F' by default).

The ID code is saved to the registry.

For the details of ID code authentication, refer to MINICUBE2 User's Manual.

**(8) Peripheral Break**

This area is used to select the operation of the peripheral emulation function of the in-circuit emulator during a break.

The peripheral emulation function corresponding to the selected item will stop during a break.

No items are selected by default.

Category A (Timer)	The peripheral emulation function of group A (timer-related) is stopped during a break.
Category B (Serial etc.)	The peripheral emulation function of group B (serial communication-related) is stopped during a break.

**Caution:** If the target device is the 78K0R/Ix3, the open break function is enabled by selecting Category A. **[MINICUBE2]**

The open break function is used to set Hi-Z the timer pins, which control a motor, so as to stop the motor safely if the motor control signal is not fed back due to CPU stoppage (break) and the signal may have an adverse effect on the motor.

The device and pins subject to manipulation by the open break function are listed in the following table. When the open break function is used, the motor stops upon a break and thus the program cannot be re-executed. In such a case, reset the CPU.

Target Device	Function	Target Pins
78K0R/Ix3	6-phase PWM output function	TO02, TO03, TO04, TO05, TO06, TO07
	Triangular wave PWM output function	TO02, TO03, TO06, TO07

**(9) Flash Programming [MINICUBE2]**

This area is used to select whether to enable flash programming.

Permit	Permits flash programming (default).
Not Permit	Prohibits flash programming.

**(10) Target [IECUBE]**

This area is used to select whether the target board is to be connected to the in-circuit emulator or not. (Refer to "Table 3-2 Error Message Output Pattern [IECUBE].")

Connect	Be connected
Not Connect	Not be connected

**Remark:** This setting is used to detect the abnormal power supply status. The default setting is determined in accordance with the power supply detection on the target system (target LED on the in-circuit emulator is lit).

**(11) Mask**

This area is used to mask the signal sent from the target.

Select the pins to be masked.

- WAIT [IECUBE]
- TARGET RESET
- INTERNAL RESET
- NMI [IECUBE]

The signal of a masked pin is not input to the in-circuit emulator.

Mask a pin only when the operation of the target system is not stable at the debugging stage.

**Caution1:** When the IECUBE is connected, and when "Not Connect" is specified in (10) Target [IECUBE], "WAIT" and "INTERNAL RESET" are fixed to the selected state, and the mask status cannot be changed. [IECUBE]

**Caution2:** "TARGET RESET" and "INTERNAL RESET" can be specified. [MINICUBE2]

**(12) Memory Mapping**

This area is used to set a memory mapping.

Specify the memory access size with "Access Size", specify the mapping attribute with "Memory Attribute", and then specify the address range with "Mapping Address".

The mapping unit is 1 byte. Up to four locations can be mapped for the Target area.

The memory setting is performed by clicking the [Add] button, and the result is listed in the area at the bottom of the window.

The mappable area depends on the product type.

Access Size	Selects memory access size. This setting is used to specify the access size on the ID78K0R-QB software; the operation of the external bus hardware is set in accordance with the settings of the MODE pin and I/O register.	
	8Bit	Assesses the memory using the MOV instruction.
	16Bit	Assesses the memory using the MOVW instruction.
Memory Attribute	The following mapping attributes can be selected. Select a mapping attribute according to the usage.	
	Target	Selects the target memory.
	I/O Protect	Selects the I/O protect area. The I/O protect area can be set in the area that is specified in the Target field. Since the area set as the I/O protect area is displayed with symbol "??" in the <a href="#">Memory window</a> like unmapped areas, read/write to this area cannot be performed freely in the <a href="#">Memory window</a> , which protects this area from being erroneously read or written.
	Stack	Selects the memory of the stack area. The stack area can be set in the area that is specified in the internal RAM field.
Mapping Address	Specify the address to be mapped. Input the higher and lower addresses from the keyboard.	
[Add] [Delete]	These buttons are used to set and delete mapping. By clicking the [Add] button, mapping is set as specified in each area and the result is listed in the area at the bottom of the window. To delete a mapped item, select the item from the list and click the [Delete] button.	

**Caution:** The area set as "I/O Protect" is not read unless it is registered to the [SFR window](#) or [Watch window](#) as an I/O port. To read this area, forcibly read it on these windows.

**Remark:** Addresses in the external memory can be specified for "Emulation ROM", "Emulation RAM", "Target", and "I/O Protect".

If the internal ROM area is set to a value smaller than that of the device, the remaining area can be specified in these areas

### (13) Wide Voltage Flash Rewriting

This area is used to select whether to enable flash rewriting at 1.8V or higher. (Only devices supported Wide Voltage mode)

On (default)	Enables downloading of load module files or software break rewriting at 1.8V or higher. (Wide Voltage mode)
Off	Enables rewriting at 2.7V or higher. (Full Speed mode)

**(14) Monitor Clock**

This area is used to select whether to use the high frequency for the monitor program during a break.

System (default)	Switch the clock to the high frequency one for the monitor program during a break (default). This accelerates the processing (such as displaying variables) during a break. <b>Caution:</b> The ID78K0R-QB manipulates the CKC, RMC and CSC registers. The frequency is reset to the original value when the execution returns to the user's program.
User	Does not switch the frequency for the monitor program during a break. However, rewriting to the flash memory cannot be performed if the frequency is too low. Therefore, the frequency is switched when rewriting must be performed, such as when downloading a load module file.

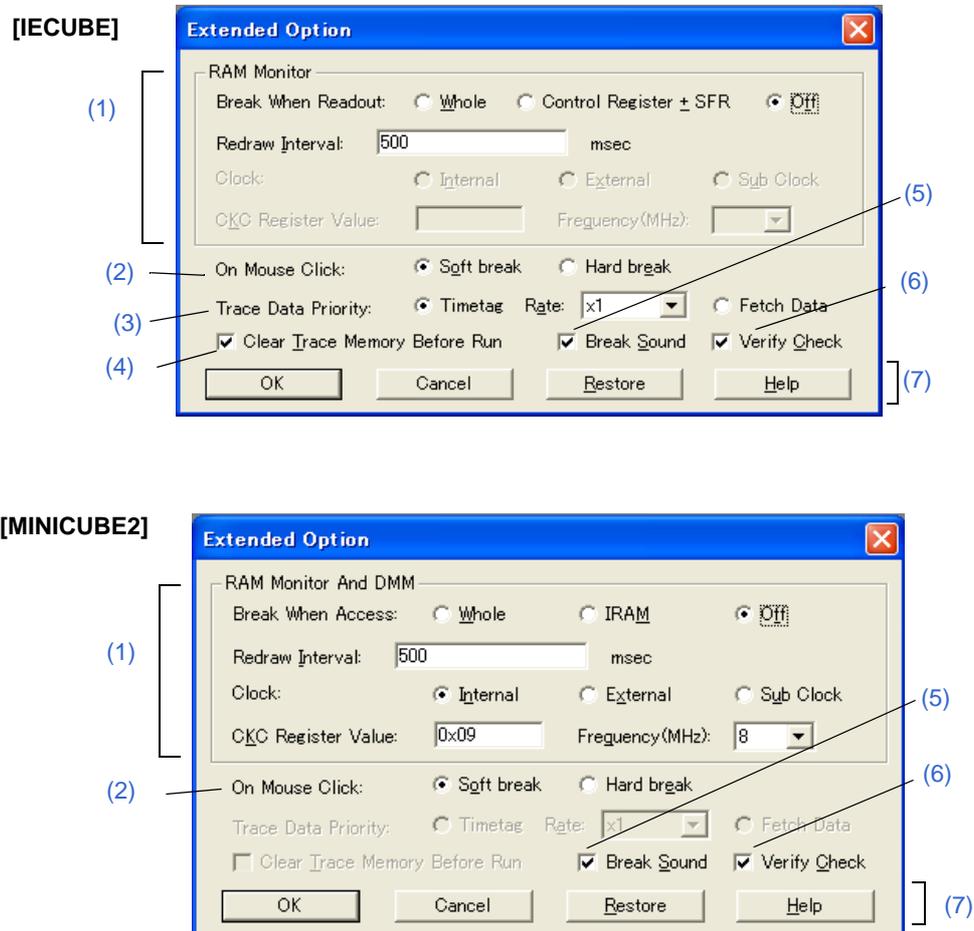
**(15) Function buttons**

OK	Validates the current environment. Sets the environment and closes this dialog box.
Cancel	Cancels the changes and closes this dialog box.
Restore	Restores the previous settings before this dialog box was opened.
Project...	Opens the <a href="#">Project File Load dialog box</a> . If an error occurs while a project file is being opened or read, the ID78K0R-QB can no longer continue and is terminated.
About...	Opens the <a href="#">About dialog box</a> .
Help	Displays the help window of this window.

## Extended Option dialog box

This dialog box is used to display and set the extended options of the ID78K0R-QB. (Refer to "5.1.2 Setting option".)

Figure 6-2 Extended Option Dialog Box



- Opening
- Explanation of each area

## Opening

Select [Option ] menu -> [Extended Option ...] .

## Explanation of each area

### (1) RAM Monitor [IECUBE] and RAM Monitor And DMM [MINICUBE2]

This area is used to set RAM monitor function, and DMM functions (for MINICUBE2 only). (Refer to "5.14 RRM Function" and "5.15 DMM Function".)

**Caution1:** If the pseudo real-time RAM monitor function is used while the Memory window is open, the operability is degraded significantly (due to an excessive amount of communication data).

When using the pseudo real-time RAM monitor function, therefore, closing the Memory window is highly recommended. [MINICUBE2]

**Caution2:** The Clock option buttons are used to select the CPU operation clock for RAM monitoring.

This setting determines the communication speed, so the CPU operation clock cannot be changed during RAM monitoring. [MINICUBE2]

**Caution3:** The data flash area cannot be specified as RRM area.

Break When Readout: [IECUBE] Break When Access: [MINICUBE2]	Select this item to specify the target range of RAM sampling by instantaneously generating a break in the user program execution. (Refer to "5.14.2 Pseudo real-time RAM monitor function (Break When Readout)".) The area where sampling by the real time monitor function is possible doesn't become it within this range of the object. (Reading by the real time monitor function is always done.)(Refer to "Table 5-19 Areas for Which Sampling Can Be Performed with Real-time RAM Monitor Function".) Select " Off " if execution of the user program will not be stopped momentarily to perform sampling.	
	Whole	Whole memory space. [IECUBE] Internal RAM, general-purpose registers and SFRs [MINICUBE2] When this function is used, the user program execution is stopped for a long time if the memory range subject to read is wide or several windows are open.
	Control Register + SFR [IECUBE]	Control registers (except PC) and SFR area
	IRAM [MINICUBE2]	Internal RAM area and SFR area
	Off	Sampling is not performed during user program execution. ROM and internal RAM areas and PC values are displayed as "***".
Clock: [MINICUBE2] Available in 1-wire mode (TOOL0) only	Internal	Selects internal high-speed oscillator
	External	Selects external clock
	Sub Clock	Selects subsystem clock
CKC Register Value: [MINICUBE2] Available in 1-wire mode (TOOL0) only	Sets the CKC value for RAM monitoring. Selectable only when 1-wire mode (TOOL0) is selected.	

Frequency: <b>[MINICUBE2]</b> Available in 1-wire mode (TOOL0) only	Sets the frequency used during RRM when the internal high-speed oscillator is used. Selectable only when 1-wire mode (TOOL0) is selected. For products that include a PLL, be sure to specify the PLL frequency (that is, the frequency of the oscillator or resonator after being multiplied by the PLL.)
Redraw Interval	Specify the sampling interval of the real-time RAM monitor function. (Refer to " <a href="#">5.14.1 Real-time RAM monitor function [IECUBE]</a> ".) It can be specified, in 100-ms units from 0 to 65500. If is specified, or if this area is blank, the data is not displayed in real time.

**(2) On Mouse Click:**

This area is used to select whether a software breakpoint or hardware breakpoint is set as the default breakpoint if a breakpoint is set in the point mark area by clicking the mouse button in the [Source window](#) or [Assemble window](#). (Refer to "[5.4.2 Breakpoint setting](#)".)

Soft break	Sets a software breakpoint (default).
Hard break	Sets a hardware breakpoint.

**(3) Trace Data Priority [IECUBE]**

This area is used to select which the following trace data to be given priority and to be recorded during trace.

Timetag	<p>Records the time tag during trace (default). Fetch data and mnemonics are displayed in the trace view in accordance with the memory contents at the time. When flash self programming is performed or if the memory contents are rewritten after the tracer has stopped, therefore, what is displayed may differ from data actually fetched or actual mnemonics. If Timetag is selected, the division ratio of the time tag counter (60 MHz) becomes selectable. One of the following ratios can be selected (refer to "<a href="#">Table 6-2 Relationship Between Division Ratio and Maximum Measurable Time (Time Tag Count)</a>"). x1 (default), x2, x4, x8, x16, x32, x64, x128, x256, x512, x1024</p>
Fetch Data	<p>Records fetch data during trace. If trace data is recorded with this mode selected, the time tag becomes empty.</p>

The relationship between a division ratio of the time tag count and the maximum measurable time is as follows.

Table 6-2 Relationship Between Division Ratio and Maximum Measurable Time (Time Tag Count)

Division Ratio	Resolution (nsec)	Maximum Measurable Time
x1	17	1.2 minutes
x2	33	2.4 minutes
x4	67	4.8 minutes
x8	133	9.5 minutes
x16	267	19.1 minutes
x32	533	38.2 minutes
x64	1067	1.3 hours
x128	2133	2.5 hours
x256	4267	5.1 hours
x512	8533	10.2 hours
x1024	17067	20.4 hours

**(4) Clear Trace Memory Before Run [IECUBE]**

Select this check box to clear the trace memory before user program execution.

It is selected by default.

**(5) Break Sound**

If the check box is checked, a beep sound is issued when a break occurs.

It is selected by default.

**(6) Verify Check**

Select this check box to be performed a verify check when data has been written to memory.

A verify check is performed when download, memory fill, or memory copy is executed. A verify check is also performed when a variable or data is changed in the [Watch window](#) or [Memory window](#) is written to memory.

**Caution:** During write to the internal flash memory (including download), verify check is not performed whether or not the checkbox in this area is selected, and internal verify of flash self-write is always performed (read verify is not performed). **[MINICUBE2]**

**(7) Function buttons**

OK	Validates the settings and closes this dialog box.
Cancel	Cancels the changes and closes this dialog box.
Restore	Restores the previous settings before this dialog box was opened.
Help	Displays the help window of this window.

## Fail-safe Break dialog box

[IECUBE]

This dialog box is used to perform the fail-safe break settings. (Refer to "5.4.5 Fail-safe break function [IECUBE]").

When a project file is read, the results obtained by reading this project file are reflected in this dialog box.

Figure 6-3 Fail-safe Break Dialog Box



- Opening
- Explanation of each area

### Opening

Click the [Detail...] button in the [Configuration dialog box](#).

### Explanation of each area

#### (1) Fail-safe break setting area

When a check box is selected, the corresponding fail-safe break function is enabled.

All these boxes are selected by default, except Flash Illegal.

Fetch From Fetch Protect Area	Fetch of fetch prohibited area
Write To Write Protect Area	Write to write prohibited area
Read From Read Protect SFR	Read of read prohibited SFR
Write To Write Protect SFR	Write to write prohibited SFR
Stack Overflow	User stack limit exceeded (upper limit)
Stack Underflow	User stack limit not reached (lower limit)
Read From Uninitialized RAM	Failure to perform RAM initialization
Access To Non Mapped Area	Access to non-mapped area
Word Miss-align Access	A word access for an odd number address was made.
Uninitialized Stack Pointer	Failure to perform stack pointer initialization
Fail-safe From Peripheral	Fail-safe from peripheral

Flash Illegal	Flash Illegal break
---------------	---------------------

**(2) Function buttons**

OK	Validates the settings and closes this dialog box.
Cancel	Closes this dialog box.
Help	Displays this dialog box online help files.

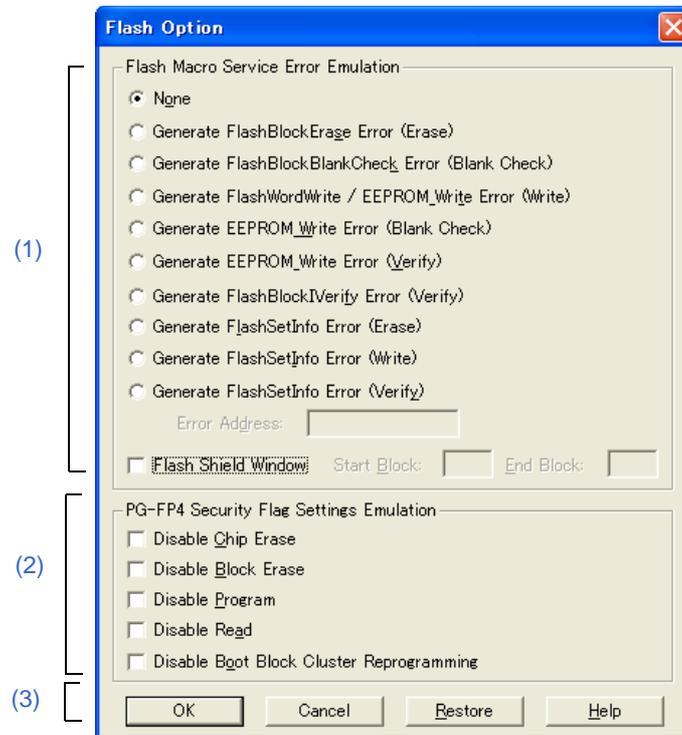
## Flash Option dialog box

[IECUBE]

This dialog box is used to make the flash self programming emulation settings.

This dialog box cannot be opened during user program execution.

Figure 6-4 Flash Option Dialog Box



- Opening
- Explanation of each area

## Opening

Select [Option] menu -> [Flash Option...].

## Explanation of each area

### (1) Flash Macro Service Error Emulation

These items set the operation of the self library function.

By selecting the item, the error value for the error set below can be forcibly returned (it is not returned in the normal emulation).

If an address at which an error is to be generated can be set depending on the selected item, input to the "Error Address" text box is available.

Error to Be Generated	Description	Address Setting
None	No errors are generated	Unavailable
Generate FlashBlockErase Error (Erase)	An erase error is returned at FlashBlockErase.	Available
Generate FlashBlockBlankCheck Error (Blank Check)	A blank check error is returned at FlashBlockBlankCheck.	Available
Generate FlashWordWrite / EEPROM_Write Error (Write)	A write error is returned at FlashWordWrite and EEPROM_Write.	Available
Generate EEPROM_Write Error (Blank Check)	A blank check error is returned at EEPROM_Write.	Available
Generate EEPROM_Write Error (Verify)	An internal verify error is returned at EEPROM_Write.	Available
Generate FlashBlockIVerify Error (Verify)	An internal verify error is returned at FlashBlockIVerify.	Available
Generate FlashSetInfo Error (Erase)	An erase error is returned at FlashSetInfo.	Unavailable
Generate FlashSetInfo Error (Write)	A write error is returned at FlashSetInfo.	Unavailable
Generate FlashSetInfo Error (Verify)	An internal verify error is returned at FlashSetInfo.	Unavailable

Error Address:	Error Address: Input an address at which an error is to be generated. Availability of this text box depends on the selected item.
----------------	-----------------------------------------------------------------------------------------------------------------------------------

Flash Shield Window	If this check box is selected, the area that can be written or erased by flash self programming can be specified. This item is not selected by default (no limitation).	
	Start Block	Specify the start block number (0 to 0xff).
	End Block	Specify the end block number (0 to 0xff).

**(2) PG-FP4 Security Flag Settings Emulation**

The initial value of the security flag is emulated when the security has been set to the flash memory using flash memory programmer PG-FP4.

The settable items are listed in the following table. All items are not selected by default.

Disable Chip Erase	Disables/enables chip erase
Disable Block Erase	Disables/enables block erase
Disable Program	Disables/enables programming
Disable Read	Disables/enables read (cannot be selected)
Disable Boot block Cluster Reprogramming	Disables/enables boot area programming

**(3) Function buttons**

OK	Validates the settings and closes this dialog box.
Cancel	Closes this dialog box.
Restore	Restores the previous settings before this dialog box was opened.
Help	Displays this dialog box online help files.



## DataFlash Option dialog box

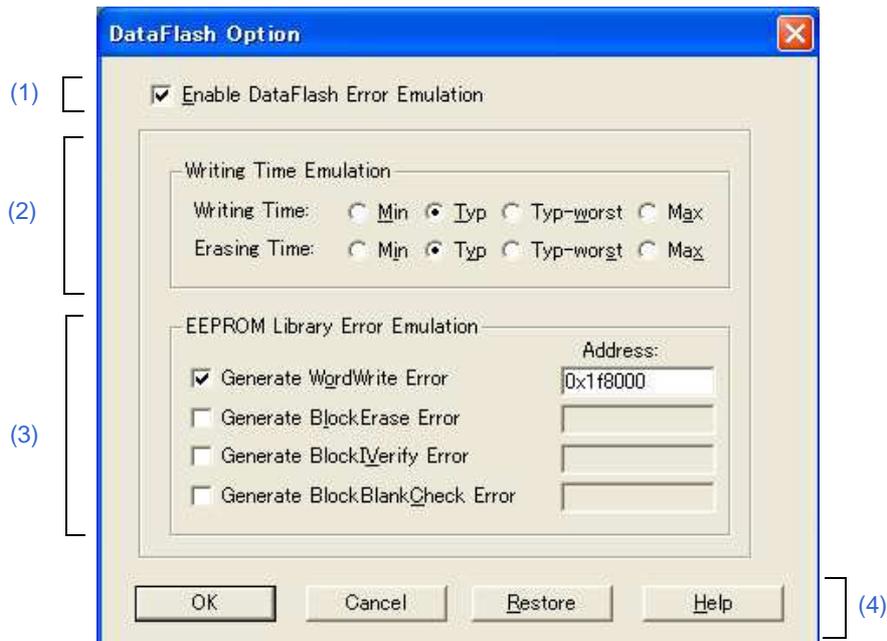
[IECUBE]

This dialog box is used to specify the data flash error emulation settings.

If the device does not include data flash area, this dialog box cannot be opened.

This dialog box also cannot be opened while the user-created program is running.

Figure 6-5 DataFlash Option Dialog Box



- Opening
- Explanation of each area

### Opening

Select [Option] menu -> [DtaFlash Option...].

## Explanation of each area

### (1) Enable DataFlash Error Emulation

Enable or disable data flash error emulation in this area.

Select the check box to enable emulation. When emulation is enabled, the Writing Time Emulation and EEPROM Library Error Emulation area become available. These checkboxes are cleared by default.

### (2) Writing Time Emulation

Specify the write / erase time in this area.

The write / erase time can be selected from the following four options.

Min	No retries
Typ	Number of retries assumed in the flash macro specifications (default)
Typ-worst	Maximum number of retries assumed in the flash macro specifications i
Max	Maximum number of retries

### (3) EEPROM Library Error Emulation

Specify the operation of the EEPROM library functions in this area.

The functions selected below can be specified to return an error (these functions do not usually return errors during emulation)

The following items can be specified. The checkboxes are all cleared by default.

After selecting an item, be sure to specify the address in the data flash area to which the error is to be returned.

Generate WordWrite Error	The WordWrite function returns an error.
Generate BlockErase Error	The BlockErase function returns an error.
Generate BlockIVerify Error	The BlockIVerify function returns an error
Generate BlockBlankCheck Error	The BlockBlankCheck function returns an error.

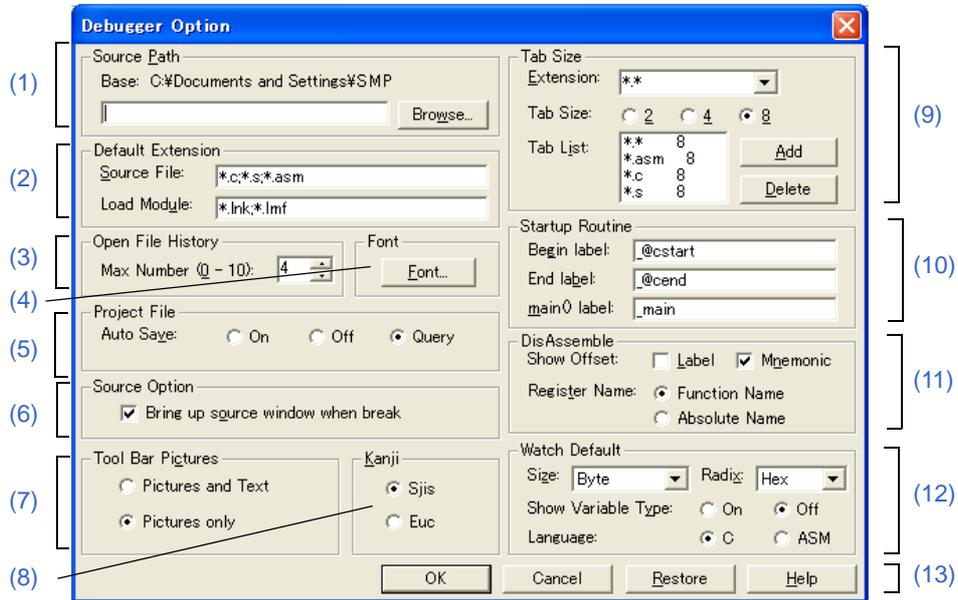
### (4) Function buttons

OK	Validates the settings and closes this dialog box.
Cancel	Closes this dialog box.
Restore	Restores the previous settings before this dialog box was opened.
Help	Displays this dialog box online help files.

## Debugger Option dialog box

This dialog box is used to display and set the various options of the ID78K0R-QB.

Figure 6-6 Debugger Option Dialog Box



- Opening
- Explanation of each area

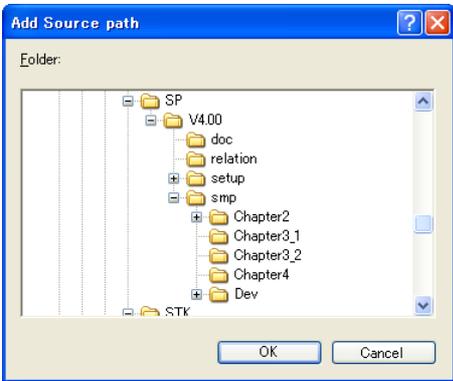
## Opening

Select [Option] menu -> [Debugger Option...].

## Explanation of each area

### (1) Source Path

This area is used to specify the folder in which a source file or text file is searched.

Base:	<p>The folder is the basis of a relative path is displayed. The base folder is determined in the following sequence:</p> <ol style="list-style-type: none"> <li>1) Folder to which the project file has been loaded</li> <li>2) Folder to which a load module or hex file has been loaded last</li> <li>3) Current folder of windows</li> </ol>
Text box	<p>This area is used to specify the folder searched. To specify a folder, either directly input one to the text box, or click the [Browse...] button. A relative path can also be specified. Opens the <a href="#">Add Source path Dialog Box</a> by clicking the [Browse...] button. To delimit paths, use ";" (semicolon) or "," (comma).</p> <p style="text-align: center;">Figure 6-7 Add Source path Dialog Box</p> 

**Remark1:** Directories that contain ";" and/or "," in the source path can be specified. Non-existent directories cannot be specified.

**Remark2:** Immediately after this dialog box has been opened, the base folder is selected and opened. If the selected folder has already been set for the source path, a source path is not added.

**Remark3:** Up to 4,095 characters can be set for the source path length in total, including a dot (.) before extension.

If more than 4,096 characters are used to specify a source path, the valid paths until the 4,095th character are set as a source path and characters that follow are ignored.

**Caution1:** When a project file is open, the [Base:] setting specifies a folder to which the project file has been load. Even if a load module file is downloaded from another folder, the source file is therefore searched for from the folder to which the project file has been loaded.

The following shows an example.

C:\TEST\                    <-- Folder to which the project file has been load  
D:\TEST\TEST.LMF <-- Downloaded load module file

Under such conditions, if source files with the same name exist under both

C:\TEST\ and D:\TEST and D:\TEST and D:\TEST\TEST.LMF is downloaded, the source file stored under C:\TEST\ is opened

**Caution2:** When a project file is not open and the source path has been set to the text box, the source file is searched for from the folder set as the source path even if a load module file is downloaded from a folder different from the one set as the source path.

The following shows an example.

C:\TEST\                    <-- Folder set as the source path  
D:\TEST\TEST.LMF <-- Downloaded load module file

Under such conditions, if source files with the same name exist under both

C:\TEST\ and D:\TEST and D:\TEST and D:\TEST\TEST.LMF is downloaded, the source file stored under C:\TEST\ is opened

**(2) Default Extension**

This area is used to specify the default extension.

Delimit extensions with " " (blank), ";" (semicolon) or "," (comma).

Source File:	Set the extension of a source file that is displayed when the <a href="#">Browse dialog box</a> is opened by selecting [File] menu -> [Open...]. The default extension is "*.c, *.s, *.asm".
Load Module:	Set the extension of a load module that is displayed when the <a href="#">Download dialog box</a> is opened. The default extension is "*.lnk, *.lmf".

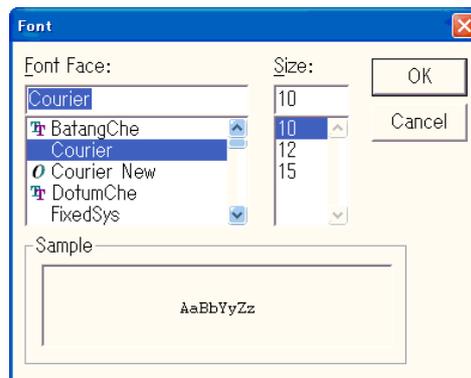
**(3) Open File History**

This area is used to set the number of histories of the open file displayed in the bottom field of the [File] menu. The default value is 4. If 0 is set, no history is displayed on the menu.

**(4) Font**

This area is used to specify the font displayed on the [Source window](#), [Watch window](#), [Quick Watch dialog box](#), [Local Variable window](#), and [Stack window](#). Clicking the [Font...] button opens the [Font Dialog Box](#) in which the font to be displayed and its size can be set.

Figure 6-8 Font Dialog Box

**(5) Project File**

This area is used to set automatic saving of the project file. (Refer to "[5.16.1 Debugging environment \(project file\)](#)".)

Auto Save:	Sets whether the project file is automatically saved at the ID78K0R-QB termination.	
	On	Automatically saves the project file.
	Off	Does not Automatically saves the project file.
	Query	Displays the <a href="#">Exit Debugger dialog box</a> at the ID78K0R-QB termination. (default)

**(6) Source Option**

This area is used to set the [Source window](#) operation at a break.

By selecting this check box, a [Source window](#) that is active at a break is displayed at the front.

If there is no active [Source window](#) or no debug information in the load module file, the active [Assemble window](#) is displayed at the front.

**(7) Tool Bar Pictures**

This area sets the buttons to be displayed on the toolbar. (Refer to "[\(2\) Operation of toolbar](#)".)

Pictures and Text	Displays a button on which a graphic and character are displayed.
Pictures only	Displays a button with only graphic. (default)

**(8) Kanji**

Cannot be selected in this area.

**(9) Tab Size**

This area is used to set the tab size for each extension when files are displayed.

Extension:	Set an extension. Input an extension from the keyboard, or select one from the drop-down list.
Tab Size:	Select the tab size. Select how many spaces are displayed as a tab code (2, 4, or 8).
Tab List:	Displays the tab size set for each extension.
[Add]	To change the tab size setting, select "Extension:" and "Tab Size:", and click this button.
[Delete]	To delete the tab size setting, select the setting from "Tab List:" and click this button.

**(10) Startup Routine**

This area is used to specify the first address, end address, and display start symbol of the text area (code area) of the start-up routine by symbols.

The source file can be opened if an object file in the load module format is downloaded in the [Download dialog box](#). (If the PC locates the address between "Begin label:" and "End label:" at this time, the ID850QB displays the code starting from the address specified in "main() label:".)

Begin label:	Specifies the symbol of the first address (default: <code>_@cstart</code> )
End label:	Specifies the symbol of the end address (default: <code>_@cend</code> )
main() label:	Specifies the display start symbol (default <code>_main</code> )

**Caution1:** If the specified symbol is not correct, the source file cannot be opened until the PC reaches the address range of the corresponding source file. In addition, the start-up routine cannot be skipped by step execution.

**Caution2:** Be sure to specify this area. If this area is blank, the dialog box cannot be closed.

**(11) DisAssemble**

This area is used to set for disassemble display.

Show Offset:	Specifies whether an offset (symbol + offset) is displayed during disassemble display. When the offset is not displayed, only a symbol that matches a numeric value is displayed, if any. If no matching symbol is found, the numeric value is displayed as a hexadecimal number unchanged.	
	Label	Specifies whether the offset is displayed in the Label field. In the default condition, the offset is not displayed.
	Mnemonic	Specifies whether the offset is displayed in the Mnemonic field. In the default condition, the offset is displayed.
Register Name:	This area is used to select the method of displaying register names in mnemonics during disassemble display.	
	Function Name	Displays register names as function names or nicknames. (default)
	Absolute Name	Displays register names as absolute names.

**(12) Watch Default**

This area is used to specify a symbol to be watched in the [Watch window](#) etc..

Size:	Selects the default display size of data if [Adaptive] is specified from the drop-down list.	
	Byte	Displays in 8 bits.(default)
	Word	Displays in 16 bits.
	Double Word	Displays in 32 bits.
Radix:	Sets the default radix in which data is to be displayed if [Proper] is specified from the drop-down list. The item selected from this list is also reflected in the subscript for watch data such as array variables (or labels) in the <a href="#">Watch window</a> (hexadecimal by default), which are registered in the <a href="#">Watch window</a> after this setting is changed.	
	Hex	Displays in hexadecimal numbers. (default)
	Dec	Displays in decimal numbers.
	Oct	Displays in octal numbers.
	Bin	Displays in binary numbers.
	String	Displays as a character string.
Show Variable Type:	Select the display/non-display of variable type is specified.	
	On	Displays the type of a variable.
	Off	Does not display the type of a variable (default).
Language:	Select the display/non-display of type of variable is specified.	
	C	Displays a C-like base number (default).
	ASM	Displays an assembly language-like base number.

**(13) Function buttons**

OK	Validates the settings and closes this dialog box.
Cancel	Cancel the changings and closes this dialog box.
Restore	Restores the previous settings before this dialog box was opened.
Help	Displays this dialog box online help files.

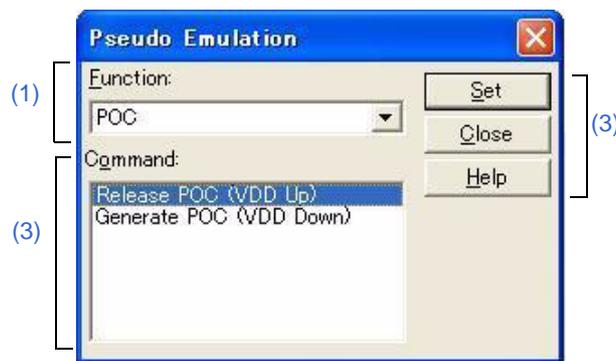
## Pseudo Emulation dialog box

[IECUBE]

This dialog box is used for generation of pseudo emulation.

Since the types of pseudo emulation that can be generated depend on the device file, this dialog box cannot be opened if the device file does not include the relevant information.

Figure 6-9 Pseudo Emulation Dialog Box



- Opening
- Explanation of each area

### Opening

Select [Run] menu -> [Pseudo Emulation...].

### Explanation of each area

#### (1) Function:

The targeted pseudo emulation can be generated by clicking the button with the name of pseudo emulation to be generated.

#### (2) Command:

Select the pseudo emulation commands to be generated.

This area lists the commands that can be generated for the pseudo emulation function specified in the "Function" drop-down list.

Select the name of emulation to be generated and then click the [Set] button.

#### (3) Function buttons

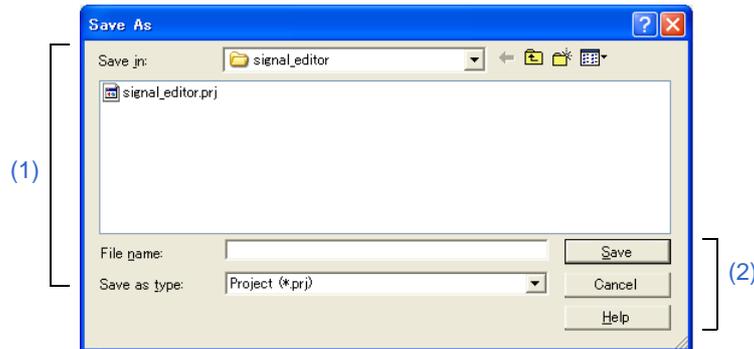
Set	Generates the pseudo emulation that is selected in the "Command" area. This button is available during user program execution only.
Close	Closes this dialog box.
Help	Displays this dialog box online help files.

## Project File Save dialog box

This dialog box is used to save the current debugging environment to a project file. (Refer to "5.16.1 Debugging environment (project file)".)

Project files can be newly saved or saved under an existing file name in this dialog box.

Figure 6-10 Project File Save Dialog Box



- Opening
- Explanation of each area

### Opening

Select [File] menu -> [Project] -> [Save As...].

(To save a file of same name as a project file previously loaded or saved, select [File] menu -> [Project] -> [Save].)

### Explanation of each area

#### (1) Save file setting area

Save in:	This area is used to specify a file name. A file name can be directly input, or selected from the list at the upper part of this area.
File name:	A string up to 259 characters with a path name and an extension can be specified.
Save as type:	This area is used to specify the extension (*.prj) of the project file to be saved. If the extension is omitted, "*.prj" is appended as the default extension.

#### (2) Function buttons

Save	Saves the debugging environment to the selected file. After saving, the dialog box is closed.
Cancel	Closes this dialog box without saving the file.
Help	Displays this dialog box online help files.

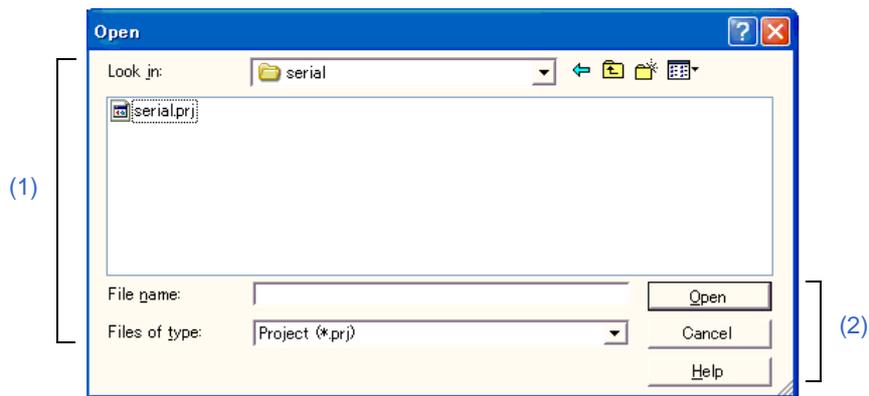
## Project File Load dialog box

This dialog box is used to restore the debugging environment to the debugging environment saved to the project file. (Refer to "5.16.1 Debugging environment (project file)".)

If there is an active [Source window](#) after a project file has been loaded, it is displayed at the top.

**Caution:** Following the ID78K0R-QB startup, if a project file with settings that differ from those of the target device at startup has been loaded, the target device specified at startup is used.

Figure 6-11 Project File Load Dialog Box



- [Opening](#)
- [Explanation of each area](#)

## Opening

Click the  **Proj** button, or select [File] menu -> [Project] -> [Open...].

## Explanation of each area

### (1) Load file setting area

Look in:	This area is used to specify the file name to be loaded. A file name can be directly input from the keyboard, or selected from the list. A string up to 259 characters with a path name and an extension can be specified.
File name:	
Files of type:	This area is used to specify the extension (*.prj) of the file to be loaded.

### (2) Function buttons

Open	Loads the selected file. After loading the file, this dialog box is closed.
Cancel	Closes this dialog box without executing anything.
Help	Displays this dialog box online help files.

## Download dialog box

This dialog box is used to select the name and format of a file to be downloaded, and downloads memory contents to the in-circuit emulator and the target system. (Refer to "5.2 Download Function, Upload Function".)

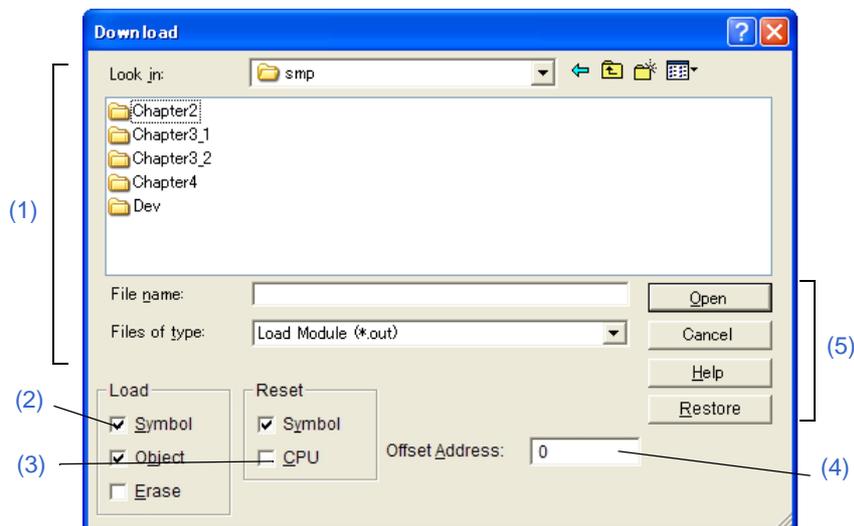
This dialog box is used to select the name and format of a file to be downloaded, and downloads memory contents to the in-circuit emulator. (Refer to "5.2 Download Function, Upload Function".)

If a load module file has been downloaded, the corresponding source file is searched, and the [Source window](#) is automatically opened.

**Caution:** If a file other than a load module file is loaded, source debugging cannot be executed.

**Remark:** The following dialog box appears while downloading and the downloading can be cancelled at any time.

Figure 6-12 Download Dialog Box



- Opening

- Explanation of each area

## Opening

Click the  **Load** button, or select [File] menu -> [Download...].

## Explanation of each area

### (1) Load file setting area

Look in:	This area is used to specify the file name to be loaded. A file name can be directly input from the keyboard, or selected from the list.
File name:	
Files of type:	This area is used to specify the extension (*.prj) of the file to be loaded. (Refer to <a href="#">"Table 5-2 Type of File That Can Be Downloaded"</a> .) These are default extensions; other extensions can also be used. The default extension of the displayed load module can also be specified in the <a href="#">Debugger Option dialog box</a> .

### (2) Load

Sets a load condition.

This setting is valid only when loading a load module file.

(If an item other than "Load Module" is selected as a file type, the [Load] area appears dimmed and the setting becomes invalid. Even if "Load Module" is selected as a file type, the [Load] area setting is ignored if a file other than a load module file (such as a hex file) is specified as a file type by directly typing it.)

Symbol	Specifies whether symbol information is read or not. This check box appears dimmed and unavailable if a file other than the load module format file is specified.
Object	Specifies whether object information is read (when selected, default) or not. (The object information is read even if this button is cleared when a HEX file is loaded.) This check box appears dimmed and unavailable if a file other than the load module format file is specified.
Erase	Specifies whether the contents of the internal flash memory are erased all before download (when selected, default) or not. <b>[MINICUBE2]</b>

### (3) Reset

Sets a reset condition. This setting is valid only if a file in the load module format is specified.

This area appears dimmed and unavailable if a file other than the load module format file is specified.

Symbol	Specifies whether symbol information is reset or not. <b>Note</b>
CPU	Specifies whether the CPU is reset or not. (Default: Reset)

### (4) Offset Address:

This area is used to specify the offset address that is used when a file is loaded (for binary data, specify the start address).

This area appears dimmed and unavailable if a file other than the load module format file or code coverage result file is specified.

An address can be also specified by a symbol or expression. (Refer to ["Table 5-5 Specifying Symbols"](#).)

The default radix for inputting a numeric value is hexadecimal.

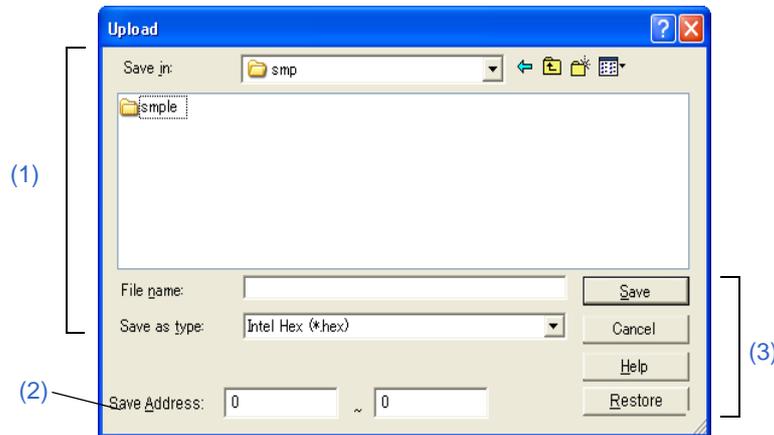
**(5) Function buttons**

Open	Loads the selected file. After loading the file, this dialog box is closed.
Cancel	Closes this dialog box without loading the file.
Help	Displays this dialog box online help files.
Restore	Restores the input data to the original status.

## Upload dialog box

This dialog box is used to set the name and format of the file to be saved, and save the set memory contents, etc., to that file. (Refer to "5.2 Download Function, Upload Function".)

Figure 6-13 Upload Dialog Box



- Opening
- Explanation of each area

## Opening

Select [File] menu -> [Upload...].

## Explanation of each area

### (1) Upload file setting area

Save in:	This area is used to specify a file name. A file name can be directly input, or selected from the list at the upper part of this area.
File name:	A string up to 259 characters with a path name and an extension can be specified.
Save as type:	This area is used to specify the extension of the file to be saved. The format of the data to be saved is determined by the extension. (Refer to "Table 5-3 Type of File That Can Be Uploaded".)

**Remark:** Extensions other than those listed can also be used.

### (2) Save Address:

This area is used to specify the range of address to be saved.

All the ranges are saved (this area cannot be set) when coverage data (\*.cvb) is selected.

An address can be also specified by a symbol or expression. (Refer to "Table 5-5 Specifying Symbols".)

The default radix for inputting a numeric value is hexadecimal.

**(3) Function buttons**

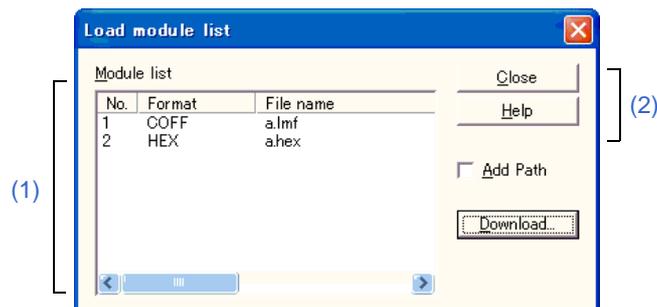
Save	Saves the file according to the setting.
Cancel	Closes this dialog box without executing anything.
Help	Displays this dialog box online help files.
Restore	Restores the status before this dialog box was opened.

## Load Module List dialog box

This dialog box displays the list of the files and file paths that have been downloaded from [Download dialog box](#). (Refer to "5.2 Download Function, Upload Function".)

The listed files (excluding the coverage data files) are saved in the project file; they are downloaded when the project file is opened next. By using the [Download...] button, the [Download dialog box](#) can be opened and a file can be downloaded.

Figure 6-14 Load Module List Dialog Box



- [Opening](#)
- [Explanation of each area](#)

## Opening

Select [File] menu -> [Load Module...].

## Explanation of each area

### (1) Load module file display area

Module list	This area displays the names of the files that have already been downloaded.		
	No.	The numbers displayed indicate the sequence in which the load module file names were read.	
	Format	The file formats are displayed	
		BIN	Binary file
		HEX	Hex file
		COV	Coverage data file
		XCOFF	Load module file (XCOFF)
unknown	Unknown		
File name	The file names are displayed with the full path if "Add Path" is checked; otherwise, only the file names will be displayed.		
Add Path	This should be checked to specify file names are displayed with the path.		
[Download...] button	Opens the <a href="#">Download dialog box</a> . A new load module can be downloaded. The file name of the newly file will be added to the file name display area when the Download dialog box is closed.		

**Caution:** If symbol information has been reset in the [Reset Debugger dialog box](#), or if symbol information has been reset in the [Download dialog box](#), the file names downloaded before that are cleared.

### (2) Function buttons

Close	Closes this dialog box.
Help	Displays this dialog box online help files.

## Source window

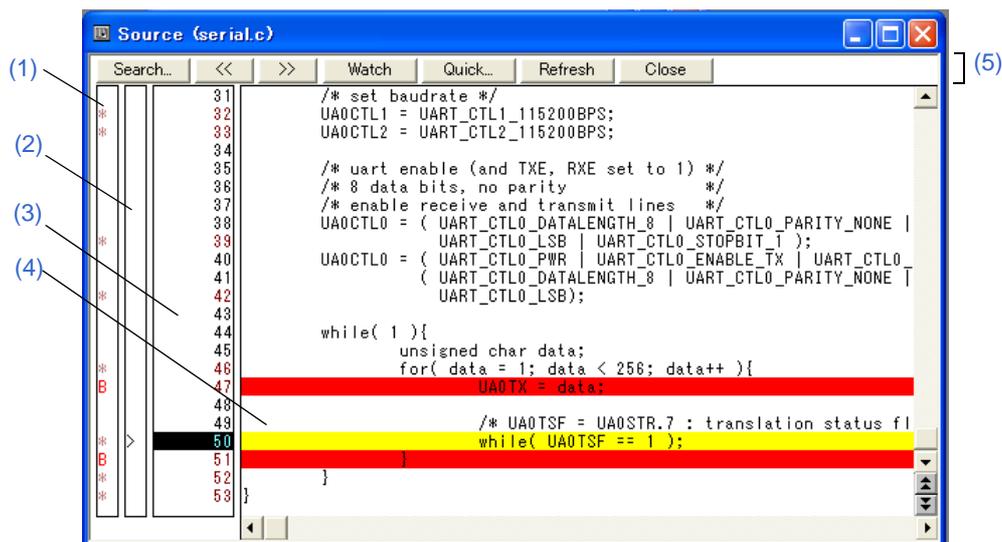
This window is used to displays source files or text files. (Refer to "5.3 Source Display, Disassemble Display Function".) In addition to [Breakpoint setting](#) , [Display of locations for which coverage measurement is executed](#) and [Mixed display mode \(Source window\)](#), a number of other operations using [Context menu](#), [Function buttons](#), etc., can be performed in this window. Moreover, there are two statuses, [Active status and static status](#), for this window. When the window is in the active status, it has the [Trace result with linking window \[IECUBE\]](#). Moreover, the items selected in the window with [Drag & drop function](#) can be used in another window. (Refer to "5.17 Functions Common to Each Window".)

**Caution1:** If program codes is described in an include file and these codes are included in multiple files, the line numbers and addresses do not correspond on a one-to-one bases. In such an include file, function that indicates the correspondence relationship between line numbers and addresses dose not correctly operate.

**Caution2:** If a source file that includes the "main" function cannot be found in the source path after a load module file is downloaded, or if the source file cannot be found during step execution, ID78K0R-QB opens a dialog box to select the source file and prompt the user to select the source path for the source file displayed in the dialog box. If the [Cancel] button is clicked, the displayed file name is memorized, so the source file name will no longer be asked, until the ID78K0R-QB is terminated.

**Remark:** Up to 65,535 lines of C and assembly language source files can be displayed. If the source files exceed 65,535 lines, partition them.

Figure 6-15 Source Window



- [Opening](#)
- [Explanation of each area](#)
- [\[View\] menu \(Source window-dedicated items\)](#)
- [Context menu](#)

## Opening

Click the  **Src** button, or select [Browse] menu -> [Source Text].

(This window is automatically opened if the corresponding source file exists after the download module file has been downloaded.)

## Explanation of each area

### (1) Point mark area

This area is used for the [Event Setting Status \(Event Mark\)](#) and program codes (\*) display, as well as [Breakpoint setting](#).

Breakpoints can be set or deleted by clicking with the mouse on this program code.

(if "\*" is not displayed for the line, the breakpoint is set on the line above or below the line, whichever has "\*" displayed.)

The program code is displayed only when the symbol information downloaded by the load module file is read.

If an event has been set for the corresponding line, one of the marks listed in the following table is displayed. The color of the "B" mark differs according to the breakpoint type and status. (When a breakpoint is set in this area, it is enabled at the same time that it is set.)

Table 6-3 Event Setting Status (Event Mark)

Mark	Meaning
B (blue)	Software breakpoint is set.
B (red)	Valid hardware breakpoint (after execution) is set.
B (green)	Valid hardware breakpoint (before execution) is set. <b>Note:</b> Breaks before execution are set with priority.
B (black)	Invalid hardware breakpoint is set. This hardware breakpoint can be validated on the <a href="#">Event manager</a> or in the <a href="#">Break dialog box</a> .
E	Event condition is set.
L	Event link condition is set. <b>[IECUBE]</b>
T	Trace event is set. <b>[IECUBE]</b>
Ti	Timer event is set. <b>[IECUBE]</b>
S	Snapshot event is set. <b>[IECUBE]</b>
M	DMM event is set. <b>[IECUBE]</b>

Mark	Meaning
A	Multiple events are set.

**Remark:** If an address range is specified as the address condition of the event, the lower addresses of the range are displayed. The mask specification of the address condition is not reflected.

## (2) Current PC mark area

The mark ">", which indicates the current PC value (PC register value), is displayed in this area.

Clicking this mark with the mouse displays a pop-up window that shows the PC register value.

By double-clicking the current PC mark area, the program can be executed up to a specified line. (Refer to "[\[Come Here\]](#)".)

## (3) Line number/address display area

This area displays the line numbers of a source file or text file.

**Red** indicates line numbers for which corresponding program code exists, and black indicates line numbers for which corresponding program code does not exist. In the [Mixed display mode \(Source window\)](#), disassemble display addresses are displayed in gray.

In addition, executed addresses are highlighted based on code coverage measurement information. (Refer to "[5.11.3 Display of locations for which coverage measurement is executed](#)".)

## (4) Source text display area

This area displays the line numbers of a source file or text file.

The following line is emphasized.

Current PC line ( <b>yellow</b> ) <sup>Note</sup>	<b>Yellow</b> indicates the current PC line (or disassemble line)('>'). If there is no line number information at the PC position when a break occurs, the execution automatically jumps to the <a href="#">Assemble window</a> . In the <a href="#">Mixed display mode (Source window)</a> , the color of only the disassemble display line is changed (the source line is displayed in the normal color).
Break point setting line ( <b>red</b> )	<b>Red</b> indicates lines where a valid breakpoint is set. In the <a href="#">Mixed display mode (Source window)</a> , the color of only the disassemble display line is changed (the source line is displayed in the normal color).

**Note:** The PC position does not move during user program execution (stays at the position before starting execution), and moves to the latest PC position when a break occurs.

Moreover, this area also provides the following functions for lines (start address of program code) and addresses where the cursor has been placed.

- [\[Come Here\]](#), [\[Start From Here\]](#) (Refer to "[Table 5-8 Type of Execution](#)".)
- [Drag & drop function](#)
- [Context menu](#)

**Caution:** If a Program code does not exist on the source line, the top address of the line above or below the line on which a program code exists is manipulated by these functions.

These functions cannot be performed in the following cases. The corresponding menu will be dimmed and cannot be selected.

- If a file other than a source file is displayed
- While the user program is being executed

## (5) Function buttons

Search...	<p>Opens the <a href="#">Source Search dialog box</a> and searches a character string of the source text. If a character string is selected in the source text display area, the Source Search dialog box is opened to search the character string.</p> <p>If no character string is selected, the Source Search dialog box is opened with nothing specified to be searched.</p> <p>Specify a search method in the Source Search dialog box.</p> <p>The results of search is highlighted in the Source window.</p> <p>This is the same operation as selecting [View] menu -&gt; [Search...].</p>
<<	<p>Searches forward (upward on screen) for the text that satisfies the search condition set in the <a href="#">Source Search dialog box</a>, starting from the address at the cursor position.</p> <p>This button is displayed as the [Stop] button during a search.</p>
>>	<p>Searches backward (downward on screen) for the text that satisfies the search condition set in the <a href="#">Source Search dialog box</a>, starting from the address at the cursor position.</p> <p>This button is displayed as the [Stop] button during a search.</p>
Stop (during a search)	Stops searching.
Watch	<p>Adds the variables selected in the source text display area to the <a href="#">Watch window</a>.</p> <p>If the <a href="#">Watch window</a> is not opened, it is opened.</p> <p>If no text is selected in the source text display area, the Watch window is only opened.</p> <p>This is the same operation as selecting [View] menu -&gt; [View Watch].</p>
Quick...	<p>Temporarily displays the contents, such as a variable, selected in the source text display area in the <a href="#">Quick Watch dialog box</a>. If no text is selected in the source text display area, the Quick Watch dialog box is only opened.</p> <p>This is the same operation as selecting [View] menu -&gt; [Quick Watch...].</p>
Refresh	Updates the contents of the window with the latest data.
Close	Closes this window.

## [View] menu (Source window-dedicated items)

The following items are added in the [\[View\] menu](#), when the Source window is active.

Create Break Event	Sets a break event that occurs if the selected variable is accessed.
Break when Access to this Variable	Sets a break event that occurs if the selected variable is accessed for read/write.
Break when Write to this Variable	Sets a break event that occurs if the selected variable is accessed for write.
Break when Read from this Variable	Sets a break event that occurs if the selected variable is accessed for read.
Clear	Deletes a break event corresponding to the selected variable.
Event Information	Displays the event information of a line at the cursor position or a selected variable name. If an event is set, the <a href="#">Event dialog box</a> is opened.
Mix	Turns on/off <a href="#">Mixed display mode (Source window)</a> .
Execution Monitoring [IECUBE]	Sets about code coverage display.
Clear [IECUBE]	Clears highlights in code coverage display.
Accumulative [IECUBE]	Cumulatively displays highlights in code coverage display.

## Context menu

Move...	Moves the display position. Opens the <a href="#">Source Text Move dialog box</a> .
Mix	Turns on/off <a href="#">Mixed display mode (Source window)</a> .
Add Watch...	Adds the specified data to the <a href="#">Watch window</a> . Opens the <a href="#">Add Watch dialog box</a> .
Symbol...	Displays the address of the specified variable or function, or the value of the specified symbol. Opens the <a href="#">Symbol To Address dialog box</a> .
Break when Access to this Variable	Sets a break event that occurs if the selected variable is accessed for read/write.
Break when Write to this Variable	Sets a break event that occurs if the selected variable is accessed for write.
Break when Read from this Variable	Sets a break event that occurs if the selected variable is accessed for read.
Clear	Deletes a break event corresponding to the selected variable.
Event Information	Displays the event information of a line at the cursor position or a selected variable name. If an event is set, the <a href="#">Event dialog box</a> is opened.
Come Here	Executes the program from the current PC to the cursor position. (Refer to <a href="#">"Table 5-6 Break Types"</a> .)

Change PC	Sets the address at the cursor position to the PC.
Break Point	Sets or deletes a hardware breakpoint at the cursor position.
Software Break Point	Sets or deletes a software breakpoint at the cursor position.
Clear Execution Monitoring [IECUBE]	Clears highlights in code coverage display.
Accumulative [IECUBE]	Cumulatively displays highlights in code coverage display.
Assemble	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position. (Refer to "5.17.2 Jump function".) Opens the <a href="#">Assemble window</a> . If an active <a href="#">Assemble window</a> is open, that window is displayed in the forefront (so that it can be manipulated).
Memory	Displays the memory contents starting from the jump destination address specified by the data value at the cursor position. (Refer to "5.17.2 Jump function".) Opens the <a href="#">Memory window</a> . If an active <a href="#">Memory window</a> is open, that window is displayed in the forefront (so that it can be manipulated).

**Remark1:** When code that includes too many instructions in one line exists in the Source window (case such as one line being repeatedly executed), stepwise execution of this one line takes time. As a result, processing of stepwise execution may be aborted in mid-flow. If aborted, the execution is suspended in the middle of the source line.

The following shows an example.

```
for( i = 0; i < 10000; i++ ){}
```

In the above example, the for statement in the second line is intended to repeat instruction processing 10,000 times. If such code is executed, processing of stepwise execution may be aborted mid-flow.

This does not apply if the processing involves multiple lines. In the above example, the following code can avoid the case.

```
int i;
for( i = 0; i < 10000; i++){
{
}
```

In this case, if stepwise execution is performed for the for statement in second line, the current PC line moves "}" in the fourth line, stepwise execution of the fourth line is performed, and then the current PC line moves again to the second line. The number of instructions performed in one stepwise execution is reduced, so stepwise execution processing is not aborted.

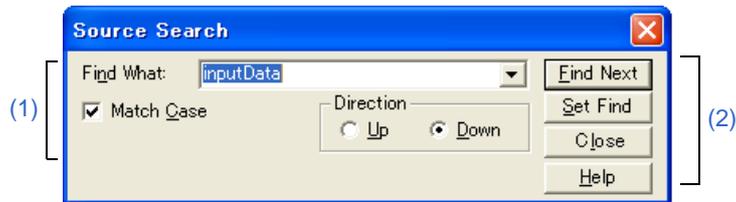
**Remark2:** When stepwise execution completes processing of the last prologue ("}") of the main function in the Source window, processing seems as if it returns to the beginning of the main function.

Since the main function is called from the startup routine, the current PC line moves to the startup routine, the current PC line is placed at the beginning of the main function in the Source window.

## Source Search dialog box

This dialog box is used to search the contents of a file in the [Source window](#). (Refer to "5.3.1 Source display".) By setting each item and then clicking the [Find Next] button, searching can be started. By clicking the [Set Find] button, the direction buttons ("<<" and ">>") in the Source window can be used for the search.

Figure 6-16 Source Search Dialog Box



- [Opening](#)
- [Explanation of each area](#)

### Opening

When the [Source window](#) is the current window, select [View] menu -> [Search...], or click the [Search...] button in the same window.

### Explanation of each area

#### (1) Search condition specification area

Find What:	This area is used to specify the data to be searched. (Up to 150 character.) In the default condition, the string selected in the window that called this dialog box is displayed. As necessary, the character string displayed can be changed. Up to 16 input histories can be recorded.	
Match Case	This should be selected to distinguish between uppercase and lowercase.	
Direction	This area is used to specify the direction of the search.	
	Up	Forward search. Searches data forward (upward on screen) from the current position of the cursor.
	Down	Backward search. Searches data backward (downward on screen) from the current position of the cursor. (default)

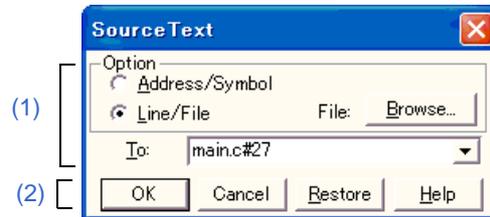
**(2) Function buttons**

Find Next	Searches the specified data in accordance with a given condition. If the specified character string is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.
Stop (during searching)	Stops searching.
Close	Closes this dialog box. (During searching, this button is replaced by the [Stop] button.)
Help	Displays this dialog box online help files.

## Source Text Move dialog box

This dialog box is used to specify a file to be displayed in the [Source window](#) and the position from which displaying the file is to be started. (Refer to "[5.3.1 Source display](#)".)

Figure 6-17 Source Text Move Dialog Box



- [Opening](#)
- [Explanation of each area](#)

## Opening

When the [Source window](#) is the current window, select [View] menu -> [Move...].

## Explanation of each area

### (1) Source file setting area

Option	Selects the input mode when the display start position is specified.	
	Address/Symbol	This should be selected to specify by an address (or symbol).
	Line/File	This should be selected to specify by a line number (or file name). To search the file name, use the [Browse...] button.

To:	Specifies the file name or address to be displayed. Up to 16 input histories can be recorded.	
	When "Address/ Symbol" is selected	Specifies the address from which display is to be started. The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or a expression. (Refer to <a href="#">"Table 5-5 Specifying Symbols"</a> .) Clicking the [OK] button displays the source text so that the source line corresponding to the specified address value can be viewed.
	When "Line/File" is selected	Specifies the line number (or a file name) from which display is to be started. The line number is specified by <b>[[path name] file name]# line number</b> . The default radix for inputting a numeric value is decimal. The file name can be specified just by the file name, or using the absolute path and relative path. If just the file name or the relative path was specified, the file in the source path specified in the <a href="#">Debugger Option dialog box</a> is searched. The file whose specified line number was specified as the first line is displayed by clicking the [OK] button. When the file name is omitted, the currently displayed file is displayed from the specified line. If the line number is omitted, the file is displayed from the first line.

**(2) Function buttons**

OK	Starts displaying the source text from the specified position.
Cancel	Closes this dialog box.
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

## Assemble window

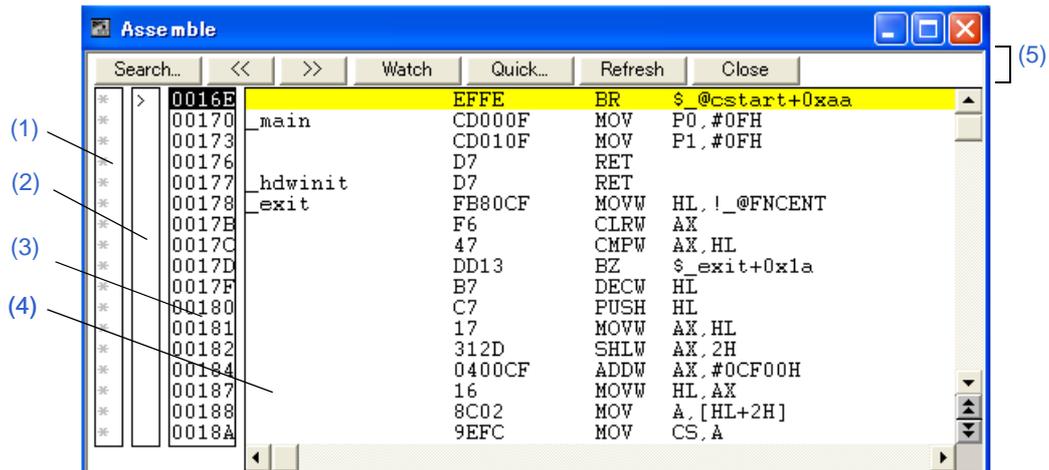
This window is used to disassemble and display programs. It is also used to execute [Online assembly](#). (Refer to "5.3 Source Display, Disassemble Display Function".) The results of online assembly are also reflected in the [Memory window](#).

In addition to [Breakpoint setting](#) and [Display of locations for which coverage measurement is executed](#), a number of other operations using [Context menu](#), [Function buttons](#), etc., can be performed in this window.

Moreover, there are two statuses, [Active status](#) and [static status](#), for this window. When the window is in the active status, it has the [Trace result with linking window \[IECUBE\]](#). Moreover, the items selected in the window with the [Drag & drop function](#) can be used in another window. (Refer to "5.17 Functions Common to Each Window".)

**Remark:** Disassemble display is available even during user program execution.

Figure 6-18 Assemble Window



- Opening
- Explanation of each area
- [View] menu (Assemble window-dedicated items)
- Context menu
- Related operations

## Opening

Click the  **Asm** button, or select [Browse] menu -> [Assemble].

## Explanation of each area

### (1) Point mark area

This area is used for [Event Setting Status \(Event Mark\)](#) and [Breakpoint setting](#).

### (2) Current PC mark area

The mark ">", which indicates the current PC value (PC register value), is displayed in this area.

By double-clicking the current PC mark area, the program can be executed up to a specified line. (Refer to "[\[Come Here\]](#)".)

### (3) Address specification area

This area displays the disassembly start address.

In addition, executed addresses are highlighted based on code coverage measurement information. (Refer to "[5.11.3 Display of locations for which coverage measurement is executed](#)".)

### (4) Disassemble display area

This area displays the labels and code data of addresses, and disassembled mnemonics.

It can be [Online assembly](#) in the mnemonic field.

The following line is emphasized.

Current PC line ((yellow) <sup>Note</sup> )	<b>Yellow</b> indicates the current PC line ('>'). If there is no line number information at the PC position when a break occurs, the execution automatically jumps to the <a href="#">Assemble window</a> .
Break point setting line (red)	<b>Red</b> indicates lines where a valid breakpoint is set.

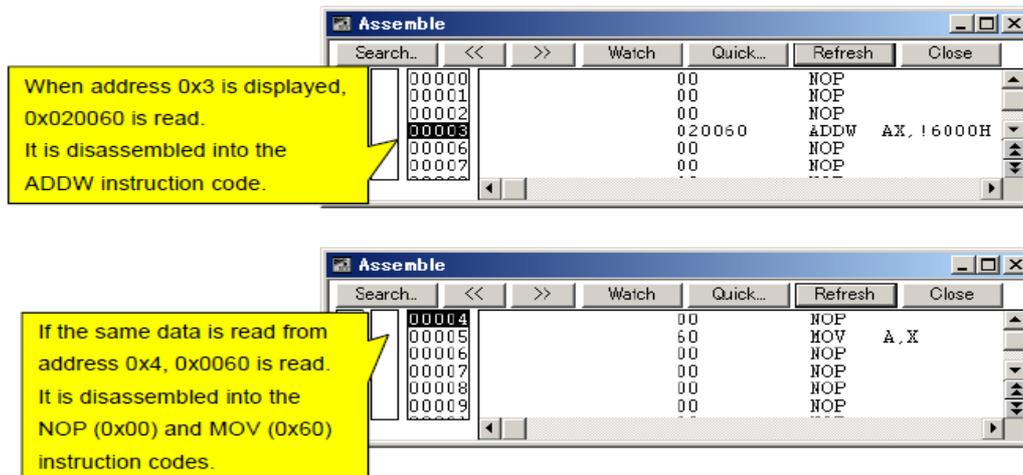
**Note:** The PC position does not move during user program execution (stays at the position before starting execution), and moves to the latest PC position when a break occurs.

**Caution1:** When displaying an SFR access instruction while an SFR with another name exists at the address where the accessed SFR exists, an SFR with a different name may be displayed in the disassemble display area.

**Caution2:** The disassemble display is a function for reading data at each address and converting it into assembly code. (With the ID78K0R-QB, for example, "0x00" is read and it is converted into "nop".) Since the instruction length may be multiple bytes, the disassemble data may not be display correctly under the circumstances where the start address of an instruction cannot be identified correctly.

The following shows an example of disassemble display output by the ID78K0R-QB.

Figure 6-19 Example of Assemble Window



Conditions where the start address of an instruction cannot be identified correctly include the following.

- If an address different from the start address of the instruction is specified in the Address Move dialog box. (Above example)
- A range that includes areas where no instructions are written (such as data area) is displayed.
- The window is scrolled up (toward lower address) for 500 bytes (equivalent to 0x00) or more.

If the display becomes incorrect as a result of scrolling the window up, click the [Refresh] button, or select the [Move...] command from the context menu to move to the start address of the instruction.

This area also provides the following functions:

- [Come Here], [Start From Here] (Refer to "Table 5-8 Type of Execution".)
- [Drag & drop function](#)
- [Context menu](#)

**(5) Function buttons**

Search...	Opens the <a href="#">Assemble Search dialog box</a> and searches for a character string of mnemonics. Specify a search method in the <a href="#">Assemble Search dialog box</a> . The results of search is highlighted in the Assemble window. This is the same operation as selecting [View] menu -> [Search...].
<<	Searches forward (upward on screen) for the contents that satisfy the search condition set in the <a href="#">Assemble Search dialog box</a> , starting from the address at the cursor position. This button is displayed as the [Stop] button during a search.
>>	Searches backward (downward on screen) for the contents that satisfy the search condition set in the <a href="#">Assemble Search dialog box</a> , starting from the address at the cursor position. This button is displayed as the [Stop] button during a search.
Stop(during a search)	Stops searching.
Watch	Adds the symbols selected in (4) <a href="#">Disassemble display area</a> to the <a href="#">Watch window</a> . If the Watch window is not opened, it is opened. If no text is selected in (4) <a href="#">Disassemble display area</a> , the Watch window is only opened. This is the same operation as selecting [View] menu -> [View Watch].
Quick...	Temporarily displays the contents, such as symbols, selected in (4) <a href="#">Disassemble display area</a> on the <a href="#">Quick Watch dialog box</a> . Opens the Quick Watch dialog box. If no text is selected in the disassemble display area, the Quick Watch dialog box is only opened. This is the same operation as selecting [View] menu -> [Quick Watch...].
Refresh	Updates the contents of the window with the latest data.
Close	Closes this window.

**[View] menu (Assemble window-dedicated items)**

The following items are added in the [\[View\] menu](#), when the Assemble window is active.

Event Information	Displays the event information of the address at the cursor position. If an event is set, the <a href="#">Event dialog box</a> is opened.
Execution Monitoring [IECUBE]	Sets about code coverage display.
Clear [IECUBE]	Clears highlights in code coverage display.
Accumulative [IECUBE]	Cumulatively displays highlights in code coverage display.

## Context menu

The menu items are effective for the selected line or item, not the position where the mouse pointer was clicked (same operation as when selecting the main menu with the same name).

Move...	Moves the display position. Opens the <a href="#">Address Move dialog box</a> .
Add Watch...	Adds the specified data to the <a href="#">Watch window</a> . Opens the <a href="#">Add Watch dialog box</a> .
Symbol...	Displays the address of the specified variable or function, or the value of the specified symbol. Opens the <a href="#">Symbol To Address dialog box</a> .
Come Here	Executes the program from the current PC to the cursor position. (Refer to " <a href="#">Table 5-6 Break Types</a> ".)
Change PC	Sets the address at the cursor position to the PC.
Break Point	Sets or deletes a hardware breakpoint at the cursor position.
Software Break Point	Sets or deletes a software breakpoint at the cursor position.
Clear Execution Monitoring [IECUBE]	Clears highlights in code coverage display.
Accumulative [IECUBE]	Cumulatively displays highlights in code coverage display.
Source Text	Displays the corresponding source text and source line, using the data value at the cursor position as the jump destination address. (Refer to " <a href="#">5.17.2 Jump function</a> ".) If no line information exists at the jump destination address, however, you cannot jump. Opens the <a href="#">Source window</a> . If an active <a href="#">Source window</a> is open, that window is displayed in the forefront (so that it can be manipulated).
Memory	Displays the memory contents starting from the jump destination address specified by the data value at the cursor position. (Refer to " <a href="#">5.17.2 Jump function</a> ".) Opens the <a href="#">Memory window</a> . If an active <a href="#">Memory window</a> is open, that window is displayed in the forefront (so that it can be manipulated).

## Related operations

---

### (1) Online assembly

To change the disassembled contents, move the cursor to the mnemonic field (the overwrite and insertion modes are alternately selected by pressing the Insert key).

If an attempt is made to move the cursor to another line after the disassembled contents have been changed in the mnemonic field, the new contents are checked. If the new contents are illegal, the code data on the line where the contents have been changed is indicated as "\*\*".

The contents changed in the mnemonic field are written into the memory by pressing the Enter key. By pressing the Enter key, the new contents are checked. If even one line is illegal, the new contents are not written into the memory. To discard the contents, press the ESC key.

If the contents are correct and if the Enter key is pressed, the contents are written to the memory, and then the cursor moves to the next line in the mnemonic field, so that the data on the next line can be changed.

**Caution:** If the number of new instruction bytes is less than the number of previous instruction bytes as a result of changing, as many 'nop' instructions as necessary are inserted. If the number of new instruction bytes is more than the number of previous instruction bytes, the next instruction is overwritten. In this case also, as many 'nop' instructions as necessary are inserted. The same applies to instructions that straddle over source lines.

## Assemble Search dialog box

This dialog box is used to search the contents in the [Assemble window](#). (Refer to "5.3.2 Disassemble display".) Successive character strings included in an input character string and disassembler character string are compared as one blank character.

By setting each item and then clicking the [Find Next] button, searching can be started. By clicking the [Set Find] button, the direction buttons ("<<" and ">>") in the Assemble window can be used for the search.

**Remark:** Search cannot be performed during program execution. (The function button appears dimmed.)

Figure 6-20 Assemble Search Dialog Box



- Opening
- Explanation of each area

### Opening

When the [Assemble window](#) is the current window, select [View] menu -> [Search...], or click the [Search...] button in the same window.

### Explanation of each area

#### (1) Search condition specification area

Find What:	This area is used to specify the data to be searched. (Up to 150 character.) In the default condition, the string selected in the window that called this dialog box is displayed. As necessary, the character string displayed can be changed. Up to 16 input histories can be recorded.	
Match Case	This should be selected to distinguish between uppercase and lowercase.	
Scan Whole Region	This should be selected to search the entire specified range.	
Direction	This area is used to specify the direction of the search.	
	Up	Forward search. Searches data forward (upward on screen) from the current position of the cursor.
	Down	Backward search. Searches data backward (downward on screen) from the current position of the cursor. (default)

Address:	This area is used to specify the address to be searched. The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to " <a href="#">Table 5-5 Specifying Symbols</a> ".)
----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**(2) Function buttons**

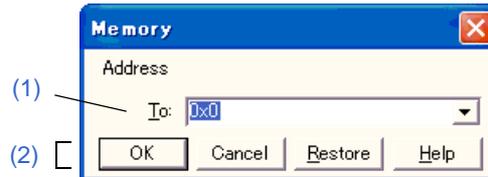
Find Next	Searches the specified data in accordance with a given condition. If the specified character string is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.
Stop (searching)	Stops searching.
Close	Closes this dialog box. (During searching, this button is replaced by the [Stop] button.)
Help	Displays this dialog box online help files.

## Address Move dialog box

This dialog box is used to specify the start address from which displaying, as follows.

- [Memory window](#)
- [Assemble window](#)
- [SFR window](#)

Figure 6-21 Address Move Dialog Box (Example: When Memory Window Is Open)



- [Opening](#)
- [Explanation of each area](#)

## Opening

When the target window is the current window, select [View] menu -> [Move...].

## Explanation of each area

### (1) Address

Specifies the start address from which displaying

In the default condition, the string selected in the window that called this dialog box, or the current PC value etc. is displayed. As necessary, the character string displayed can be changed.

The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to "[Table 5-5 Specifying Symbols](#)".) Up to 16 input histories can be recorded.

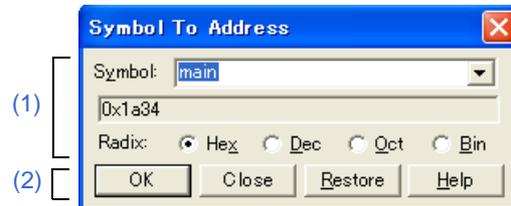
### (2) Function buttons

OK	The corresponding window is displayed from the address.
Cancel	Closes this dialog box.
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

## Symbol To Address dialog box

This dialog box is used to display the address of the specified variable or function, or the value of the specified symbol. (Refer to "5.3.4 Convert symbol (symbol to address)".)

Figure 6-22 Symbol To Address Dialog Box



- [Opening](#)
- [Explanation of each area](#)

## Opening

Select [View] menu -> [Symbol...].

## Explanation of each area

### (1) Symbol conversion area

Symbol:	This area is used to specify the variable, function name, symbol name, or line number to be converted. (Refer to "Table 5-5 Specifying Symbols".) To change the contents of this area, click the [OK] button. The conversion result will be displayed in the area below. The default radix for inputting a numeric value is decimal. Up to 16 input histories can be recorded.	
Conversion result display area	If bit symbol have been specified, they are converted to the Address.bit format. Also, equations that include bit symbols cannot be specified. The variable, address of the function, value of the symbol, address of the line number, or value of the expression specified in "Symbol:" is displayed. The address value of an I/O port name or SFR name, the register contents of a register name, or flag value of a PSW flag name is displayed.	
Radix:	This area is used to select the radix of the converted data to be displayed.	
	Hex	Hexadecimal number (default)
	Dec	Decimal number
	Oct	Octal number
	Bin	Binary number

**(2) Function buttons**

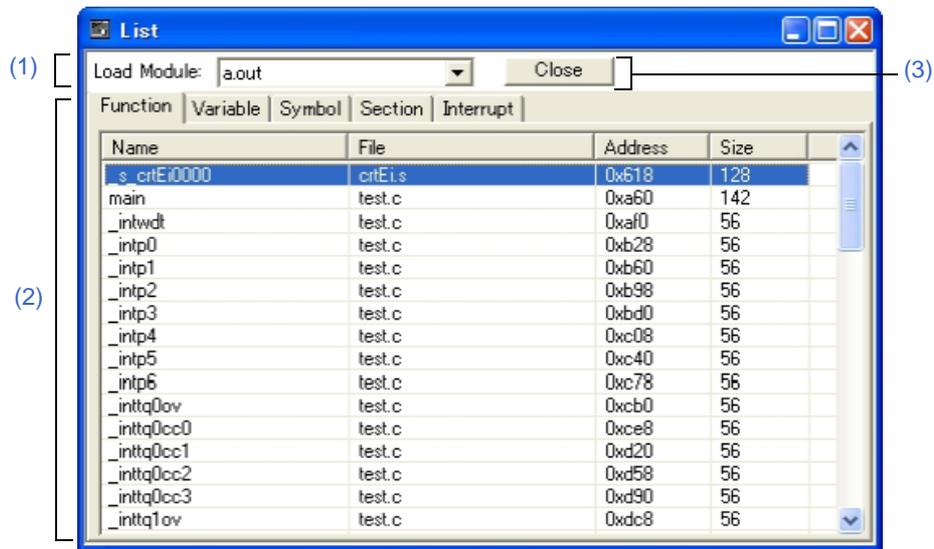
OK	If the contents of "Symbol:" have been changed, converts the symbol. After conversion, closes the dialog box if the contents of "Symbol:" have not been changed.
Close	Closes this dialog box.
Restore	Restores the input data to the original status. If the [OK] button has already been clicked, the data is restored to the status immediately after the [OK] button was clicked.
Help	Displays this dialog box online help files.

## List window

This window lists functions, variables, symbols, sections, and interrupt requests.

Display data can be saved in the CSV format, independently for a selected tab. (Refer to ["5.16.2 Window display information \(view file\)"](#).)

Figure 6-23 List Window



- Opening
- Explanation of Each Area
- Context menu

## Opening

Select [Browse] menu -> [List].

## Explanation of Each Area

### (1) Load Module:

This area is used to select the load module file that has been downloaded.

This area is blank when no load module file has been downloaded.

### (2) List view area

The items such as names, sizes, and addresses of functions, variables, symbols, sections, and interrupt requests are listed, by each tab. The displayed contents are updated automatically after a load module file is downloaded.

This area is blank when no load module file has been downloaded.

(a) When [Function] tab is selected

Name	Function name (displayed as function in segment units in case of assembler source file)
File	Name of file in which the function is defined
Address	Function start address
Size	Function size (unit: bytes)

(b) When [Variable] tab is selected

Name	Variable name
File	Name of file in which the variable is defined
Function	Name of function in which variable is defined
Address	Variable start address
Size	Variable size (unit: bytes)

(c) When [Symbol] tab is selected

Name	Symbol name
Address	Symbol address

(d) When [Section] tab is selected

Name	Section name
Type	Section type (code, data)
Address	Section start address
Size	Section size (unit: bytes)

(e) When [Interrupt] tab is selected

Name	Interrupt request name
Type	Interrupt type (nonmaskable, maskable, software, security id, flash mask option)
Status	Utilization status in the program (use, nonuse) - - - - : Unknown
Address	Starting address of the interrupt handler
Size	Size of the interrupt handler (unit: bytes) Maximum size for statuses other than "use"

The pointer can also jump to any of the above windows just by double-clicking the jump source address.

The display jumps from this tab to the [Source window](#), [Assemble window](#) or [Memory window](#) using the start address value of the selected line as a jump pointer. The jump destination window will be displayed from the jump pointer.

The jump function is executed by selecting a jump source line then selecting [Source Text/Assemble/Memory] in the [Jump] menu.

**Remark:** The displayed items are sorted by clicking the title (on the label) in each column (ascending/descending order is switched each time the title is clicked). The width of each column can be changed, but the change is lost when the data in the window is saved as the CSV file.

### (3) Function buttons

Close	Closes this window.
-------	---------------------

## Context menu

Source Text	Displays the corresponding source text and source line, using the data value at the cursor position as the jump destination address. (Refer to " <a href="#">5.17.2 Jump function</a> ".) If no line information exists at the jump destination address, however, you cannot jump. Opens the <a href="#">Source window</a> . If an active <a href="#">Source window</a> is open, that window is displayed in the forefront (so that it can be manipulated).
Assemble	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position. (Refer to " <a href="#">5.17.2 Jump function</a> ".) Opens the <a href="#">Assemble window</a> . If an active <a href="#">Assemble window</a> is open, that window is displayed in the forefront (so that it can be manipulated).
Memory	Displays the memory contents starting from the jump destination address specified by the data value at the cursor position. (Refer to " <a href="#">5.17.2 Jump function</a> ".) Opens the <a href="#">Memory window</a> . If an active <a href="#">Memory window</a> is open, that window is displayed in the forefront (so that it can be manipulated).

## Watch window

This window is used to display and change specified watch data. (Refer to "5.6 Watch Function".)

This window can also display wide-ranging watch data (such as global variables and public symbols) in real time even during program execution, in the same way as the [Memory window](#).

The results of updating and rewriting data in this window will be reflected in the [Memory window](#).

Watch data is registered by clicking the [Watch...] button in the [Source window](#) or [Assemble window](#). (Refer to "5.6.3 Registering and deleting watch data".)

This window allows easy setting of breakpoints to variables via a [Context menu](#).

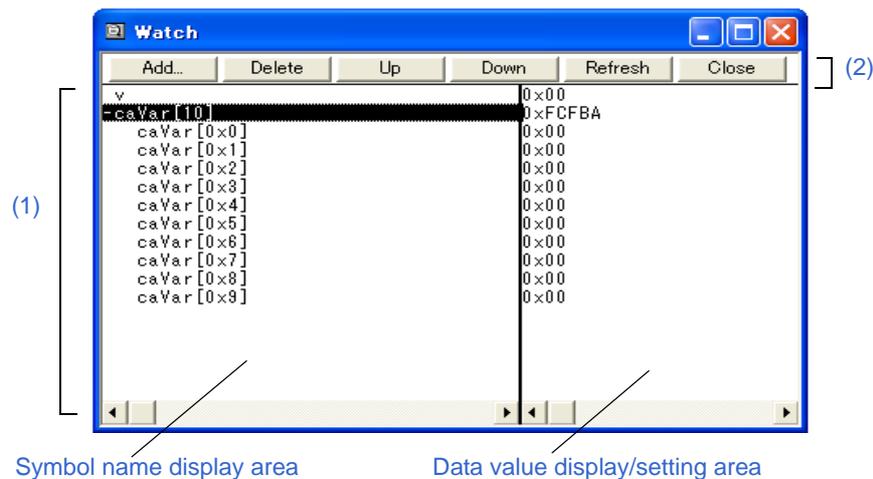
**Remark1:** If a local variable and a global variable exist with the same name, the local variable takes priority.

**Remark2:** When a variable is referenced in the Watch window during program execution, the data currently being changed may be displayed at the timing of the variable change.

**Remark3:** A maximum of 10,000 lines can be displayed in the Watch Window.

**Remark4:** Multiple lines cannot be selected in the Watch Window.

Figure 6-24 Watch Window



- Opening
- Explanation of each area
- [View] menu (Watch window-dedicated items)
- Context menu

## Opening

Click the  **Wch** button, or select [Browse] menu -> [Watch].

## Explanation of each area

### (1) Watch data display/change area

The left area displays symbol names, and the right area displays data values.

Symbol name display area	<p>This area is used to display variable names, symbol names and types, and tag names of structures or unions. '+' is prefixed to the displayed arrays, pointer variables, and structures or unions. These variables are expanded and displayed when they are double-clicked (first character changes from "+" to "-"). Registered watch data changes are performed in the <a href="#">Change Watch dialog box</a> opened by selecting the item to be changed and then selecting <a href="#">Context menu</a> -&gt; [Change Watch...]. A line with an expanded hierarchy, such as the elements of an array, and members of structures and unions cannot be deleted. If an access breakpoint is set for a variable or a symbol in the Watch window, the symbol name display area is highlighted in gold.</p>	
	Array	By double-clicking the "+", all the elements of the variable are displayed in accordance with the type of the array variable.
	Pointer variable	By double-clicking the "+", the data indicated by the pointer is displayed.
	Structure/union	By double-clicking the "+", all the members of the structure/union are displayed in accordance with the type of the member variable. If a structure or union is defined in the structure or union, the structure name or union name of the internal structure or union is also displayed. The internal structure or union can be also expanded by using "+".
Data value display/setting area	<p>This area is used to display and change watch data values. A value is updated when execution is stopped. To save a value, select [File ] menu -&gt; [Save As...]. This area is blank if getting data has failed. Values are changed through direct input. The location to be changed is displayed in red and the contents of the change are written into the target memory when the Enter key is pressed. The previous value can be canceled by the ESC key.<sup>Note</sup></p>	
	Display Data	Contents
	Integer	Hexadecimal ( <b>0x</b> xxxx)or (xxxx <b>H</b> ) Decimal (xxxx)or (xxxx <b>T</b> ) Octal ( <b>0</b> xxxx)or (xxxx <b>Q</b> ) Binary ( <b>0b</b> xxxx)or (xxxx <b>Y</b> )
	Character	"Character"
	Enumeration type	Member name
	If scope is specified	Displayed in accordance with specified scope.
	Floating-point type	Single precision/double precision supported. The input/display format is as follows: [ +   - ] inf [ +   - ] nan [ +   - ] integer e [ +   - ]exponent [ +   - ] integer.fraction[ e [ +   - ]exponent
	"?"	Data that has been invalidated because of a change in the scope or optimized compiling

**Note:** Symbols in the internal RAM area are written without momentarily breaking execution.  
 Symbols in the internal ROM area, target area, or registers including SFRs are written after writing is agreed upon in the confirmation dialog box, while execution is stopped momentarily. **[IECUBE]**  
 Symbols in the internal RAM area are written after writing is agreed upon in the confirmation dialog box, while execution is stopped momentarily.  
 Symbols other than the above cannot be written (if attempted, an error occurs). **[MINICUBE2]**

**Remark1:** If an array has too many variables and takes too long to expand, a warning message is displayed.

**Remark2:** The radix can be changed per variable using the [Context menu](#). The display format of integers and the number of elements for arrays can be changed in the [Debugger Option dialog box](#).

**Caution:** Even if Hex is selected for Radix in the Watch Default field in Debugger Option dialog box, the number of array variables is not displayed in hexadecimal in the Watch window. (The following shows examples when array "ary" is displayed. )

Desired display example:

```
- ary[0x2]  <-- Hexadecimal
  ary[0x0]  <-- Hexadecimal
  ary[0x1]  <-- Hexadecimal
```

Actual display example:

```
- ary[2]    <-- Decimal
  ary[0x0]  <-- Hexadecimal
  ary[0x1]  <-- Hexadecimal
```

## (2) Function buttons

Add...	Opens the <a href="#">Add Watch dialog box</a> . If watch data is specified and the [Add...] button is clicked in the <a href="#">Add Watch dialog box</a> , the specified watch data is added to the Watch window.
Delete	Deletes the selected watch data from the window.
Up	Moves the selected line one line up.
Down	Moves the selected line one line down.
Refresh	Updates the contents of this window with the latest watch data.
Close	Closes this window.

## [View] menu (Watch window-dedicated items)

When this window is the current window, The following items are added on [\[View\] menu](#).

Only the selected item is subject to this manipulation.

Create Break Event	Creates a break event by using the selected item as follows.
--------------------	--------------------------------------------------------------

Beak when Access to this Variable	Creates a break event that can be accessed for read/write by using the selected item.
Break when Write to this Variable	Creates a break event that can be accessed for write by using the selected item.
Break when Read from this Variable	Creates a break event that can be accessed for read by using the selected item.
Clear	Deletes a break event corresponding to the selected item.
Event Information	Displays the event information of the variable selected. If an event is set, the <a href="#">Event dialog box</a> is opened.
Bin	Displays the selected line in binary numbers.
Oct	Displays the selected line in octal numbers.
Dec	Displays the selected line in decimal numbers.
Hex	Displays the selected line in hexadecimal numbers.
String	Displays the selected line as a character string.
Proper	Displays the selected line as the default value of each variable. Symbols are displayed in accordance with the setting of the <a href="#">Debugger Option dialog box</a> (default).
Byte	Displays the selected line in 8-bit units.
Word	Displays the selected line in 16-bit units.
Double Word	Displays the selected line in 32-bit units.
Adaptive	Displays the selected line as the default value of each variable. (default) Only this item is valid for a symbol in C language. Symbols in assembly language are displayed in accordance with the setting of the <a href="#">Debugger Option dialog box</a> .
Up	Moves the selected line one line up.
Down	Moves the selected line one line down.
Compulsion Read	Forcibly reads SFR, that are disabled from being read because their values will be changed, or the data of the I/O ports and I/O protect area added in the <a href="#">Add I/O Port dialog box</a> .

## Context menu

The menu items are effective for the selected line or item, not the position where the mouse pointer was clicked (same operation as when selecting the main menu with the same name).

Beak when Access to this Variable	Creates a break event that can be accessed for read/write by using the selected item.
Break when Write to this Variable	Creates a break event that can be accessed for write by using the selected item.
Break when Read from this Variable	Creates a break event that can be accessed for read by using the selected item.
Clear	Deletes a break event corresponding to the selected item.

RRM Setting...	Sets the sampling range of RRM function. Opens the <a href="#">RRM Setting dialog box</a> . <b>[MINICUBE2]</b>
Event Information	Displays the event information of the variable selected. If an event is set, the <a href="#">Event dialog box</a> is opened.
Change Watch...	Changes the selected watch data. Opens the <a href="#">Change Watch dialog box</a> .
Delete Watch	Deletes the selected watch data from the window.
Bin	Displays the selected line in binary numbers.
Oct	Displays the selected line in octal numbers.
Dec	Displays the selected line in decimal numbers.
Hex	Displays the selected line in hexadecimal numbers.
String	Displays the selected line as a character string.
Proper	Displays the selected line as the default value of each variable. Symbols are displayed in accordance with the setting of the <a href="#">Debugger Option dialog box</a> (default).
Byte	Displays the selected line in 8-bit units.
Word	Displays the selected line in 16-bit units.
Double Word	Displays the selected line in 32-bit units.
Adaptive	Displays the selected line as the default value of each variable (default). Only this item is valid for a symbol in C language. Symbols in assembly language are displayed in accordance with the setting of the <a href="#">Debugger Option dialog box</a> .
Up	Moves the selected line one line up.
Down	Moves the selected line one line down.

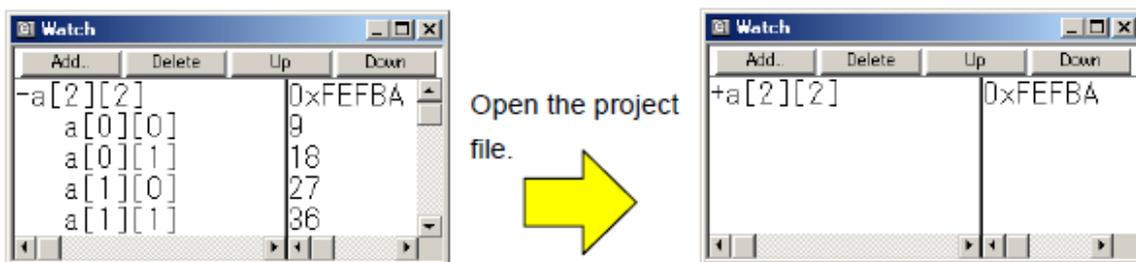
**Remark:** If elements such as structures and arrays are registered in the Add Which dialog box and they are referenced in the Watch window, the display with which each member is displayed and the display radix setting are not saved in the project file. To save these items to the project file, each member must also be registered in the Add Watch dialog box.

The following is an example of saving the display for each member of the array "a" shown below.

```
char a[2][2];
```

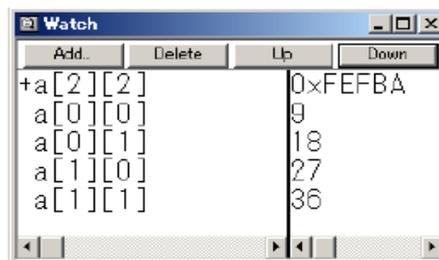
Register the array "a" in the Add Watch dialog box and then display its members in the Watch window; the following display appears. Even if this state is saved in the project file and the project file is reloaded, the members are not displayed.

Figure 6-25 Example of Watch Window 1



To save the state with members displayed, register in the Add Watch dialog box the members to be displayed. In this example, register not only the array "a", but also its members "a[0][0]", "a[0][1]", "a[1][0]" and "a[1][1]". By taking this measure, the following display can be saved.

Figure 6-26 Example of Watch Window 2



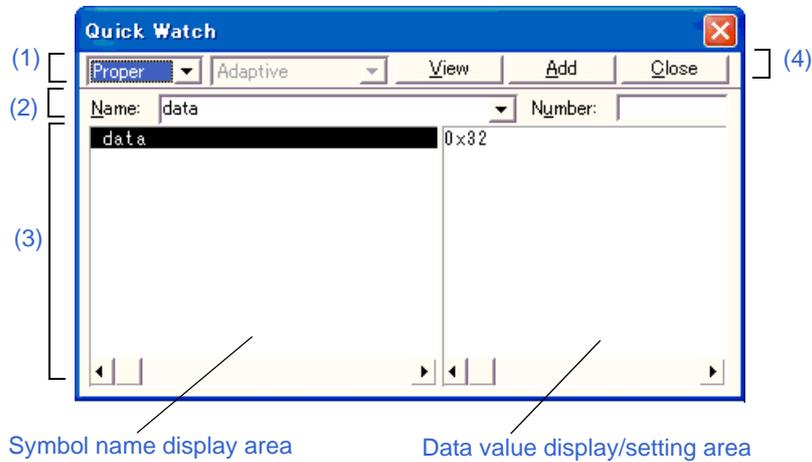
**Caution:** When Auto variables are used to display local variables, accurate values cannot be displayed at a prologue ("{"") or epilogue ("}") of a function.

(The Auto variable addresses are the relative addresses from the address pointed to by the stack pointer (SP), so their addresses are not determined until the SP value is determined in the function. The SP is manipulated via prologues or epilogues, so the accurate value cannot be displayed)

## Quick Watch dialog box

This dialog box is used to temporarily display or change specified watch data. (Refer to "5.6 Watch Function".)

Figure 6-27 Quick Watch Dialog Box



- Opening
- Explanation of each area

### Opening

When the [Source window](#) or [Assemble window](#) is the current window, select [View] menu -> [Quick Watch...], or click the [Quick...] button in same window.

### Explanation of each area

#### (1) Display form specification area

Display radix selection area 	This area is used to select the display radix.	
	Proper	Variable: Displays the default value of each variable. Symbol: Displays data with the radix set in the <a href="#">Debugger Option dialog box</a> .
	Hex	Displays in hexadecimal numbers.
	Dec	Displays in decimal numbers.
	Oct	Displays in octal numbers.
	Bin	Displays in binary numbers.
	String	Displays as a character string.

Display size selection area 	This area is used to select the display size. If the display size is fixed, such as when a variable in C language or register is to be displayed, it cannot be changed.	
	Adaptive	Variable: Displays the default value of each variable. Symbol: Displays data with the size set in the <a href="#">Debugger Option dialog box</a> .
	Byte	Displays in 8-bit units.
	Word	Displays in 16-bit units.
	Double Word	Displays in 32-bit units.

## (2) Watch data specification area

Name:	This area is used to specify the watch data to be displayed. In the default condition, the string selected in the window that called this dialog box is displayed. As necessary, the character string displayed can be changed. Up to 16 input histories can be recorded. If the contents of this area have been changed, the data specified can be displayed in the field below by clicking the [View] button.
Number:	This area is used to specify the number of data to be displayed (blank or a number of 1 to 256). If this area is blank, data is displayed as a simple variable. If a number of 1 or more is specified, data is displayed as an array variable in the <a href="#">Watch window</a> . If an array variable is displayed, "+" is prefixed to the data. By double-clicking this "+", all the elements of the data are expanded and displayed in accordance with the type of the data ("- is prefixed to the expanded data. If this "-" is double-clicked, the expanded display is canceled). If the number of data to be displayed is fixed, such as when a variable in C language or register is to be displayed, the specified number of data is invalid.

## (3) Watch data display area

The left area displays symbol names, and the right area displays data values.

Symbol name display area	This area is used to display watch data (variable names, symbol names and types, and tag names of structures or unions). (Refer to " <a href="#">Symbol name display area</a> " in the <a href="#">Watch window</a> .) This area cannot be edited.
Data value display/setting area	This area is used to display and change data values. (Refer to " <a href="#">Data value display/setting area</a> " in the <a href="#">Watch window</a> .)

## (4) Function buttons

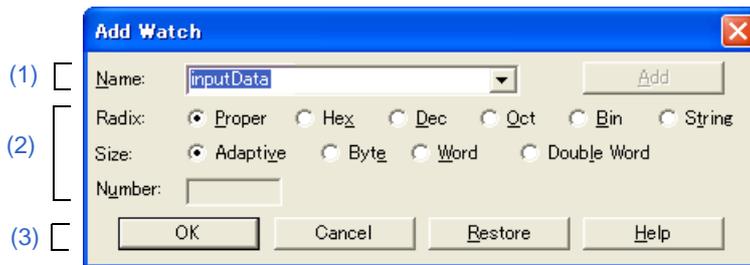
View	Displays the data specified in <a href="#">(2) Watch data specification area</a> in the field below.
Add	Adds the data specified in <a href="#">(2) Watch data specification area</a> to the <a href="#">Watch window</a> .
Close	Closes this dialog box. Data that has not actually been written to the target memory will be canceled.

## Add Watch dialog box

This dialog box is used to register watch data to be displayed in the [Watch window](#). (Refer to "[5.6 Watch Function](#)".)

Multiple data with the same symbol name can be registered.

Figure 6-28 Add Watch Dialog Box



- [Opening](#)
- [Explanation of each area](#)

### Opening

Select [View] menu -> [Add Watch...], or click the [Add...] button in the [Watch window](#).

### Explanation of each area

#### (1) Watch data registration area

Name:	This area is used to specify symbol to be added to the <a href="#">Watch window</a> . (Refer to " <a href="#">Table 6-4 Watch Window Input Format</a> ".) In the default condition, the string selected in the window that called this dialog box is displayed. As necessary, the character string displayed can be changed. This area is blank if no character string is selected. Up to 16 input histories can be recorded.
[Add]	Adds the specified data to the <a href="#">Watch window</a> . The dialog box remains open.

Table 6-4 Watch Window Input Format

Variable Name of C language		
	Variable expression : Variable Name	
	Variable expression [Constant value   Variable Name]	Elements of array
	Variable expression. Member name	Entity members of structure/union
	Variable expression -> Member name	Members of structure/union indicated by pointer
	*Variable expression	Value of pointer variable
	&Variable expression	Address where variable is located
Register name		
SFR name, SFR bit name		
Label, EQU and address of immediate value		
Register name.bit		
SFR name. bit		
Label name.bit , EQU symbol.bit, address of immediate value.bit		
Bit symbol		
Specification of scope		

How a variable is handled when a scope is specified is as follows:

Table 6-5 How Variable Is Handled When Scope Is Specified

Scope Specification	Program Name	File Name	Function Name	Variable Name
<b>prog\$file#func#var</b>	prog	file	func	var
<b>prog\$file#var</b>	prog	file	global	var
<b>prog\$func#var</b>	prog	global	func	var
<b>prog\$var</b>	prog	global	global	var
<b>file#func#var</b>	current	file	func	var
<b>file#var</b>	current	file	global	var
<b>func#var</b>	current	current	func	var
<b>var</b>	current	current	current	var

**(2) Display form change area**

Radix:	This area is used to select the display radix.	
	Proper	Variable: Displays the default value of each variable. Symbol: Displays data with the radix set in the <a href="#">Debugger Option dialog box</a> .
	Hex	Displays in hexadecimal numbers.
	Dec	Displays in decimal numbers.
	Oct	Displays in octal numbers.
	Bin	Displays in binary numbers.
	String	Displays in strings.
Size:	This area is used to select the display size. If the display size is fixed, such as when a variable in C language or register is to be displayed, it cannot be changed.	
	Adaptive	Variable: Displays the default value of each variable. Symbol: Displays data with the size set in the <a href="#">Debugger Option dialog box</a> .
	Byte	Displays in 8-bit units.
	Word	Displays in 16-bit units.
	Double Word	Displays in 32-bit units.
Number:	<p>This area is used to specify the number of data to be displayed (blank or a number of 1 to 256).</p> <p>If this area is blank, data is displayed as a simple variable. If a number of 1 or more is specified, data is displayed as an array variable in the <a href="#">Watch window</a>.</p> <p>If an array variable is displayed, "+" is prefixed to the data. By double-clicking this "+", all the elements of the data are expanded and displayed in accordance with the type of the data ("- " is prefixed to the expanded data. If this "-" is double-clicked, the expanded display is canceled).</p> <p>If the number of data to be displayed is fixed, such as when a variable in C language or register is to be displayed, the specified number of data is invalid.</p>	

**(3) Function buttons**

OK	Adds the specified data to the <a href="#">Watch window</a> . Closes this dialog box.
Cancel	Closes this dialog box.
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

## Change Watch dialog box

This window is used to change the data on a line selected in the [Watch window](#). (Refer to "5.6 Watch Function".)  
A line with an open hierarchy, such as the elements of an array, and members of structures and unions cannot be changed.

When watch data is changed, the contents of the selected line are replaced with the new data.

The symbol name can be changed even if it results in duplication of a name already in use with existing data.

Figure 6-29 Change Watch Dialog Box



- [Opening](#)
- [Explanation of each area](#)

### Opening

When the [Watch window](#) is the current window, select [View] menu -> [Change Watch...].

### Explanation of each area

#### (1) Watch data change area

Name:	This area is used to change a symbol name on a line selected in the <a href="#">Watch window</a> . (Refer to "Table 6-4 Watch Window Input Format".) The symbol name can be changed even if it results in duplication of a name already in use with existing data. In the default condition, the string selected in the window that called this dialog box is displayed. Up to 16 input histories can be recorded.
[Add]	Cannot be selected.

**(2) Display form change area**

Radix:	This area is used to change the display radix on a line selected in the <a href="#">Watch window</a> .	
	Proper	Variable: Displays the default value of each variable. Symbol: Displays data with the radix set in the <a href="#">Debugger Option dialog box</a> .
	Hex	Displays in hexadecimal numbers.
	Dec	Displays in decimal numbers.
	Oct	Displays in octal numbers.
	Bin	Displays in binary numbers.
	String	Displays in strings.
Size:	This area is used to change the display size on a line selected in the <a href="#">Watch window</a> . If the display size is fixed, such as when a variable in C language or register is to be displayed, it cannot be changed.	
	Adaptive	Variable: Displays the default value of each variable. Symbol: Displays data with the size set in the <a href="#">Debugger Option dialog box</a> .
	Byte	Displays in 8-bit units.
	Word	Displays in 16-bit units.
	Double Word	Displays in 32-bit units.
Number:	<p>This area is used to change the number of data to be displayed on a line selected in the <a href="#">Watch window</a> (blank or a number of 1 to 256).</p> <p>If this area is blank, data is displayed as a simple variable. If a number of 1 or more is specified, data is displayed as an array variable in the <a href="#">Watch window</a>.</p> <p>If an array variable is displayed, "+" is prefixed to the data. By double-clicking this "+", all the elements of the data are expanded and displayed in accordance with the type of the data ("- " is prefixed to the expanded data. If this "-" is double-clicked, the expanded display is canceled).</p> <p>If the number of data to be displayed is fixed, such as when a variable in C language or register is to be displayed, the specified number of data is invalid.</p>	

**(3) Function buttons**

OK	Replaces the data on a line selected in the <a href="#">Watch window</a> with the specified data, and then closes this dialog box.
Cancel	Closes this dialog box.
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

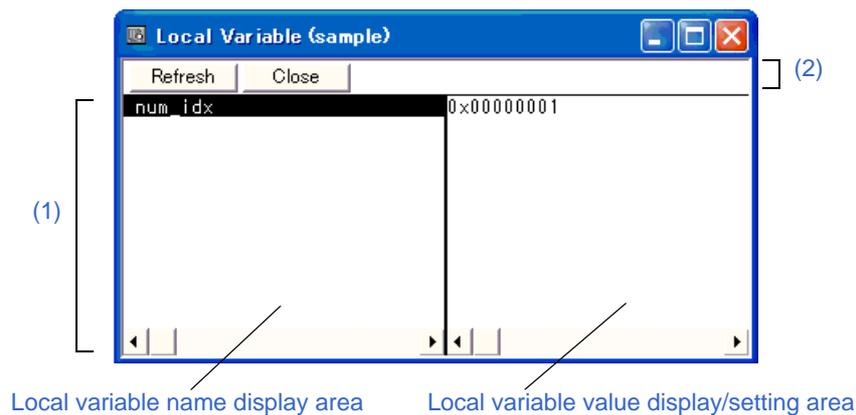
## Local Variable window

This window is used to display the local variable in the current function and change the local variable values. (Refer to "5.6.2 Displaying and changing local variable values".)

It is linked with the [Jump function](#) of the [Stack window](#), and displays the local variable in the function jumped when jumping to the [Source window](#).

A number of other operations using [Context menu](#), [Function buttons](#), etc., can be performed in this window.

Figure 6-30 Local Variable Window



- Opening
- Explanation of each area
- [View] menu (Local Variable window-dedicated items)
- Context menu

## Opening

Click the  **Loc** button, or select [Browse] menu -> [Local Variable].

## Explanation of each area

### (1) Local variable display/variable value changing area

Local variable name display area	This area displays local variable name. (Refer to " <a href="#">Symbol name display area</a> " in the <a href="#">Watch window</a> .) Auto, Internal Static, and Register variables can be displayed. Local variables within the current function are automatically displayed in this window. This area cannot be edited.
----------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Local variable value display/ setting area	This area is used to display and change local variable values. (Refer to " <a href="#">Data value display/setting area</a> " in the <a href="#">Watch window</a> .) Values are changed through direct input. The location to be changed is displayed in <b>red</b> and the contents of the change are written into the target memory when the Enter key is pressed. During user program execution, however, the change cannot be written (if attempted, an error occurs). The previous value can be canceled by the ESC key.
-----------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## (2) Function buttons

Refresh	Updates the contents of this window with the latest watch data.
Close	Closes this window.

**Caution:** When Auto variables are used to display local variables, accurate values cannot be display at a prologue ("{" or epilogue ("}") of a function.

(The Auto variable addresses are the relative addresses from the address pointed to by the stack pointer (SP), so their addresses are not determined until the SP value is determined in the function. The SP is manipulated via prologues or epilogues, so the accurate value cannot be displayed)

## [View] menu (Local Variable window-dedicated items)

When this window is the current window, the following items are added on [\[View\] menu](#).

Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays octal numbers.
String	Displays character strings.
Proper	Displays the default value of each variable. (default)

## Context menu

The menu items are effective for the selected line or item, not the position where the mouse pointer was clicked (same operation as when selecting the main menu with the same name).

Add Watch...	Opens the <a href="#">Add Watch dialog box</a> .
Bin	Displays the selected line in binary numbers.
Oct	Displays the selected line in octal numbers.
Dec	Displays the selected line in decimal numbers.
Hex	Displays the selected line in hexadecimal numbers.
String	Displays the selected line as a character string.

Proper	Displays the selected line as the default value of each variable. Symbols are displayed in accordance with the setting of the <a href="#">Debugger Option dialog box</a> . (default)
--------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Stack window

This window is used to display or change the current stack contents of the user program. (Refer to "5.6.7 Stack trace display function".)

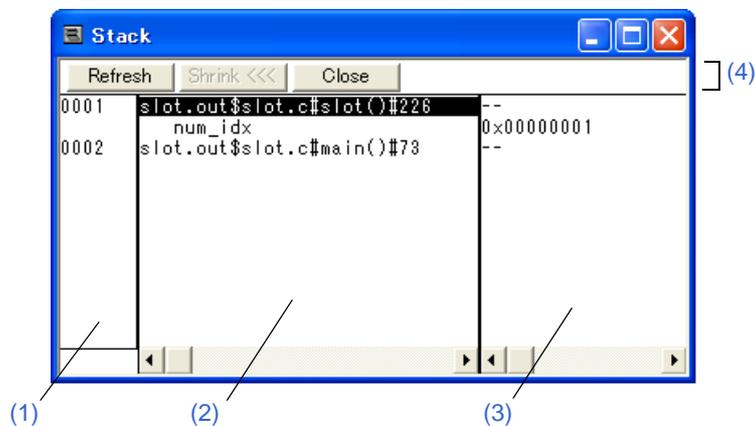
The window corresponding to the stack contents can be jumped to using the [Jump function](#).

A number of other operations using [Context menu](#), [Function buttons](#), etc., can be performed in this window.

**Caution:** The stack trace display function may not operate properly when there is a function (noauto, noexec, etc.) that does not push the frame pointer (HL) onto the stack.

**Remark:** [ERROR] may be displayed during prologue or epilogue processing of a function.

Figure 6-31 Stack Window



- Opening
- Explanation of each area
- [View] menu (Stack window-dedicated items)
- Context menu

## Opening

Click the  **Stk** button, or select [Browse] menu -> [Stack Trace].

## Explanation of each area

### (1) Stack frame number display area

This area assigns numbers to and displays the stack contents.

A stack frame number is a natural number starting from 1. The shallower the nesting of the stack, the higher the number. This means that a function having stack number one higher than that of a certain function is the function that calls the certain function.

### (2) Stack frame contents display area

This area displays the stack frame contents.

It displays function names or local variable names. Note, however, that this area cannot be edited.

If the stack contents consist of a function	They are displayed as follows: <b>[program name\$file name#function name (argument list) #line number]</b> If this line is double-clicked, the operation will be the same as jumping to the <a href="#">Source window</a> of the <a href="#">Jump function</a> (i.e., the local variable in the function to which execution has jumped will be displayed in the <a href="#">Local Variable window</a> ). If the function has a local variable, the local variable will be displayed on the next and subsequent lines.
If the stack contents consist of a local variable	Its type and name are displayed. (Refer to " <a href="#">Watch window</a> ".) Note that the internal Static and Register variables are not displayed.

### (3) Stack contents display/setting area

This area is used to display or change the stack contents.

If the stack contents are a function	"- -" is displayed and the function cannot be changed.
If the stack contents are a local variable	The variable value is displayed. (Refer to " <a href="#">Watch window</a> ".) Values are changed through direct input. The location to be changed is displayed in <b>red</b> and the contents of the change are written into the target memory when the Enter key is pressed. During user program execution, however, the change cannot be written (if attempted, an error occurs). The previous value can be canceled by the ESC key.

### (4) Function buttons

Refresh	Updates the contents of this window with the latest watch data.
Shrink <<<	Collapses the local variable list of the selected function.
Expand >>> (when the [Shrink<<<] button is clicked)	Displays the local variable list of the selected function.
Close	Closes this window.

## [View] menu (Stack window-dedicated items)

When this window is the current window, The following items are added on [\[View\] menu](#).

Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays octal numbers.
String	Displays character strings.
Proper	Displays the default value of each variable. (default)

## Context menu

Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays octal numbers.
String	Displays character strings.
Proper	Displays the default value of each variable. (default)
Source Text	Displays the corresponding source text and source line from the jump destination address specified by the data value at the cursor position. (Refer to " <a href="#">5.17.2 Jump function</a> ".) If no line information exists at the jump destination address, however, you cannot jump. Opens the <a href="#">Source window</a> . If an active <a href="#">Source window</a> is open, that window is displayed in the forefront (so that it can be manipulated).
Assemble	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position. (Refer to " <a href="#">5.17.2 Jump function</a> ".) Opens the <a href="#">Assemble window</a> . If an active <a href="#">Assemble window</a> is open, that window is displayed in the forefront (so that it can be manipulated).
Memory	Displays the memory contents starting from the jump destination address specified by the data value at the cursor position. (Refer to " <a href="#">5.17.2 Jump function</a> ".) Opens the <a href="#">Memory window</a> . If an active <a href="#">Memory window</a> is open, that window is displayed in the forefront (so that it can be manipulated).

## Memory window

This window is used to display and change the memory contents. (Refer to "5.7 Memory Manipulation Function".) Other operations using [Context menu](#), [Function buttons](#), etc., can be performed in this window.

Moreover, there are two statuses, [Active status](#) and [static status](#), for this window. When the window is in the active status, it has the [Trace result with linking window \[IECUBE\]](#), [Jump function](#). (Refer to "5.17 Functions Common to Each Window".)

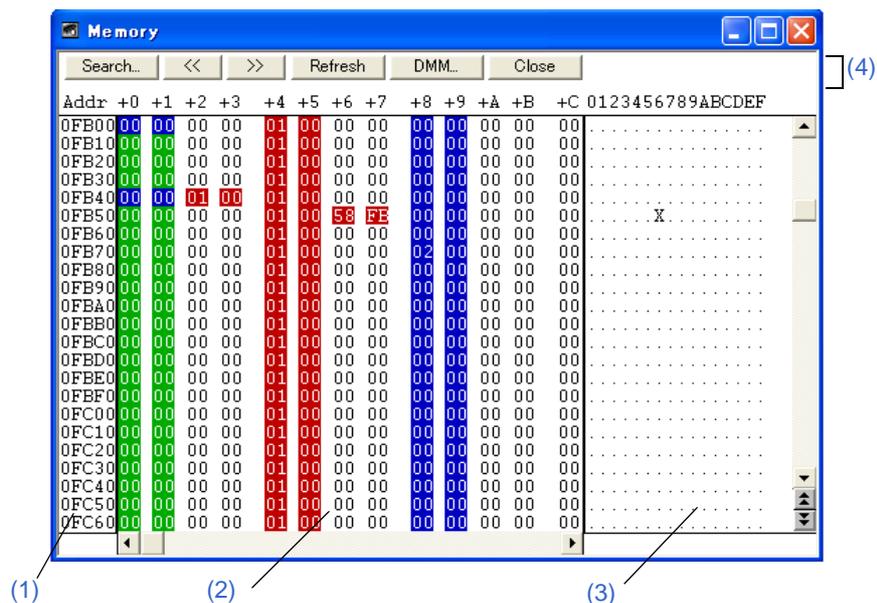
**Remark1:** The memory access status (read, write, read & write) can be displayed using different colors. (Refer to "5.7.2 Access monitor function [IECUBE]").

**Remark2:** The display start position when the this window is opened is as follows:

First time: Display starts from the first address of the RAM area.

Second and subsequent times: Display starts from the address at which an active status window was closed (if an active status window has never been closed, display starts from the first display start position).

Figure 6-32 Memory Window



- Opening
- Explanation of each area
- [View] menu (Memory window-dedicated items)
- Context menu

## Opening

Click the  **Mem** button, or select [Browse] menu -> [Memory].

## Explanation of each area

### (1) Addr

This area displays memory addresses.

### (2) +0 +1 +2....

This area is used to display and change memory contents, and to display the access status. (Refer to "5.7.2 Access monitor function [IECUBE].")

Display		Symbol Substituted When Display Information Is Saved in View File	Meaning
	Green	R	Read or Fetch (Internal RAM)
	Red	W	Write
	Blue	A	Read & write

Memory contents are changed through direct input. The location to be changed is displayed in red and the contents of the contents of the change are written into the target memory when the Enter key is pressed. The previous value can be canceled by the ESC key. Up to 256 bytes can be specified at one time.

**Remark:** To change the memory contents during user program execution, open the [DMM dialog box](#) by clicking the [DMM...] button. [IECUBE]

### (3) 0 1 2 3....

This area is used to display and change the memory contents in ASCII characters.

This area is displayed when [View] menu -> [Ascii] is selected.

Data can be changed in this area in the same manner as in the memory display area.

The changing method is the same as in (2) +0 +1 +2....

**Remark:** If a window is activated and then the Memory window is activated while the cursor is placed in the ASCII display area, the cursor position moves to the memory display area.

### (4) Function buttons

Search...	Opens the <a href="#">Memory Search dialog box</a> and searches for character strings from the displayed memory contents, or memory contents. Selected data (a memory value) is displayed in the Memory Search dialog box as data to be searched. If the Memory Search dialog box is opened without data specified, specify data from the keyboard. The results of the search is highlighted in the Memory window.
<<	Searches the memory contents satisfying the search condition set in the <a href="#">Memory Search dialog box</a> , forward (upward on screen) from the address at the cursor position. This button is displayed as the [Stop] button during a search.

>>	Searches the memory contents satisfying the search condition set in the <a href="#">Memory Search dialog box</a> , backward (downward on screen) from the address at the cursor position. This button is displayed as the [Stop] button during a search.
Stop(searching)	Stops searching.
Refresh	Updates the contents of the window with the latest data.
DMM...	Opens the <a href="#">DMM dialog box</a> . <b>[IECUBE]</b> Cannot be selected. <b>[MINICUBE2]</b>
Close	Closes this window.

## **[View] menu (Memory window-dedicated items)**

The following items are added in the [\[View\] menu](#) , when the Memory window is active.

Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays hexadecimal numbers (default).
Nibble	Displays in 4-bit units.
Byte	Displays in 8-bit units (default).
Word	Displays in 16-bit units.
Double Word	Displays in 32-bit units.
Ascii	Selects whether ASCII characters are displayed or not. Checked: Displayed Not checked: Hidden (default)
Little Endian	Displays in little endian (default).
Big Endian	Displays in big endian.
Access Monitoring <b>[IECUBE]</b>	Sets about Access monitor function.
Clear <b>[IECUBE]</b>	Clears the display color through the access monitor function.
Accumulative <b>[IECUBE]</b>	Enables/disables cumulative display of access status (memory content change). Checked: Cumulative display of memory contents changes Not checked: Display of only memory contents changes from previous update.

### **(5) Context menu**

The menu items are effective for the selected line or item, not the position where the mouse pointer was clicked (same operation as when selecting the main menu with the same name).

Move...	Moves the display position. Opens the <a href="#">Address Move dialog box</a> .
Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.

Hex	Displays hexadecimal numbers. (default)
Nibble	Displays in 4-bit units.
Byte	Displays in 8-bit units. (default)
Word	Displays in 16-bit units.
Double Word	Displays in 32-bit units.
Ascii	Selects whether ASCII characters are displayed or not. Checked: Displayed Not Checked: Hidden (default)
Clear Access Monitoring <b>[IECUBE]</b>	Clears the display color through the access monitor function.
Accumulative <b>[IECUBE]</b>	Enables/disables cumulative display of access status (memory content change). Checked: Cumulative display of memory contents changes Not Checked: Display of only memory contents changes from previous update.

## Memory Search dialog box

This dialog box is used to search the memory contents of the part of the [Memory window](#) at which the cursor is located. (Refer to "[5.7 Memory Manipulation Function](#)".)

If the cursor is placed in [memory display area](#) in the Memory window, the specified data is treated as a binary data string, and if the cursor is placed in [the ascii display area](#), the specified data is treated as an ASCII character string, and the contents of these respective areas are searched.

By setting each item and then clicking the [Find Next] button, searching can be started. By clicking the [Set Find] button, the direction buttons ("[<<](#)" and "[>>](#)") in the Memory window can be used for the search.

**Caution:** Non-mapped, SFR, and I/O protect areas are not searched.

Figure 6-33 Memory Search Dialog Box



- [Opening](#)
- [Explanation of each area](#)

## Opening

When the [Memory window](#) is the current window, select [View] menu -> [Search...], or click the [Search...] button in the same window.

## Explanation of each area

### (1) Search condition specification area

Find What:	This area is used to specify the data to be searched. (Up to 150 character.) In the default condition, the string selected in the window that called this dialog box is displayed. As necessary, the character string displayed can be changed. Up to 16 input histories can be recorded.	
	When searching in memory display area	Up to 16 data items can be specified. Delimit each data with a "(blank).
	When searching in ascii display area	Up to 150 characters can be specified. A "(blank) in the data is treated as a blank character.
Unit:	This area is used to specify the number of bits of the data to be searched in memory display area.	
	Byte	Searches the data as 8-bit data. (default)
	Word	Searches the data as 16-bit data.
	Double Word	Searches the data as 32-bit data.
Scan Whole Region	This should be selected to search the entire specified range.	
Direction	This area is used to specify the direction of the search.	
	Up	Forward search. Searches data forward (upward on screen) from the current position of the cursor.
	Down	Backward search. Searches data backward (downward on screen) from the current position of the cursor. (default)
Address:	This area is used to specify the address range to be searched. The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to " <a href="#">Table 5-5 Specifying Symbols</a> ".)	

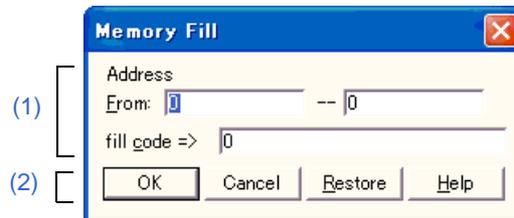
### (2) Function buttons

Find Next	Searches the specified data in accordance with a given condition. If the specified character string is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.
Stop (searching)	Stops searching.
Close	Closes this dialog box.(During searching, this button is replaced by the [Stop] button.)
Help	Displays this dialog box online help files.

## Memory Fill dialog box

This dialog box is used to fill the memory contents in the [Memory window](#) with specified codes (fill code). (Refer to ["5.7 Memory Manipulation Function"](#).)

Figure 6-34 Memory Fill Dialog Box



- [Opening](#)
- [Explanation of each area](#)

### Opening

Select [Edit] menu -> [Memory] -> [Fill...].

### Explanation of each area

#### (1) Memory fill range specification area

Address	This area is used to specify the filling range and fill code.	
	From:	Specifies the filling range (start address -- end address). The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to <a href="#">"Table 5-5 Specifying Symbols"</a> .)
	fill code =>	Specify the data (fill code) used when filling the range specified in "From: ". Up to 16 binary data strings (byte data strings) can be specified. Delimit each data with a " " (blank).

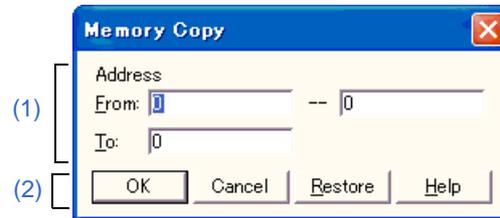
#### (2) Function buttons

OK	Fills the specified data in accordance with a given condition.
Stop (filling)	Stops filling.
Cancel	Closes this dialog box. (During filling, this button is replaced by the [Stop] button.)
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

## Memory Copy dialog box

This dialog box is used to copy the memory contents in the [Memory window](#). (Refer to "5.7 Memory Manipulation Function".)

Figure 6-35 Memory Copy Dialog Box



- [Opening](#)
- [Explanation of each area](#)

### Opening

Select [Edit] menu -> [Memory] -> [Copy...].

### Explanation of each area

#### (1) Copy range specification area

Address	This area is used to specify the copy source and copy destination addresses. The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to "Table 5-5 Specifying Symbols".)	
	From:	Specify the address range (start address -- end address) of the copy source.
	To:	Specify start address of the copy destination.

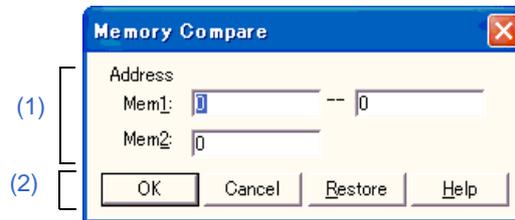
#### (2) Function buttons

OK	Copies the memory contents in accordance with a given condition.
Stop (copying)	Stops copying.
Cancel	Closes this dialog box.(During copying, this button is replaced by the [Stop] button.)
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

## Memory Compare dialog box

This dialog box is used to compare the memory contents in the [Memory window](#). (Refer to "[5.7 Memory Manipulation Function](#)".)

Figure 6-36 Memory Compare Dialog Box



- [Opening](#)
- [Explanation of each area](#)

### Opening

Select [Edit] menu -> [Memory] -> [Compare...].

### Explanation of each area

#### (1) Comparison range specification area

Address	This area is used to specify the comparison source address and comparison destination address. The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to " <a href="#">Table 5-5 Specifying Symbols</a> ".)	
	Mem1:	Specify the address range (start address -- end address) of the comparison source.
	Mem2:	Specify the start address of the comparison destination.

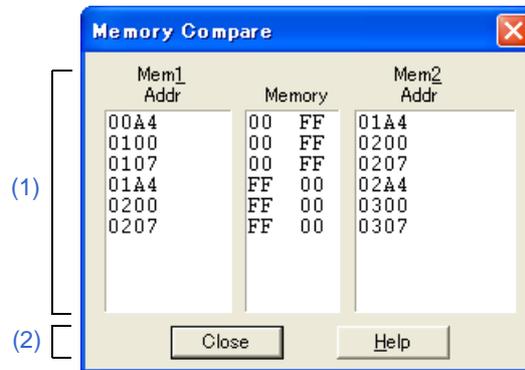
#### (2) Function buttons

OK	Compares the memory contents in accordance with a given condition. If no difference is found as a result of comparison, " <a href="#">Wf200: No difference encountered</a> ." is displayed. If a difference is found, the <a href="#">Memory Compare Result dialog box</a> is opened.
Stop (comparison)	Stops memory comparison.
Cancel	Closes this dialog box.(During comparison, this button is replaced by the [Stop] button.)
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

## Memory Compare Result dialog box

This dialog box is displayed if any difference is found in the memory contents when the memory has been compared in the [Memory Compare dialog box](#). (Refer to "5.7 Memory Manipulation Function".)

Figure 6-37 Memory Compare Result Dialog Box



- [Explanation of each area](#)

### Explanation of each area

#### (1) Comparison result display area

This area displays the results of comparing the memory. Only differences that have been found as a result of comparison are displayed.

Mem1 Addr	Displays a comparison source address in which a difference has been found.
Memory	Displays the data in which a difference has been found. (Left: Comparison source data, Right: Comparison destination data).
Mem2 Addr	Displays the comparison destination address at which a difference has been found.

#### (2) Function buttons

Close	Closes this dialog box.
Help	Displays this dialog box online help files.

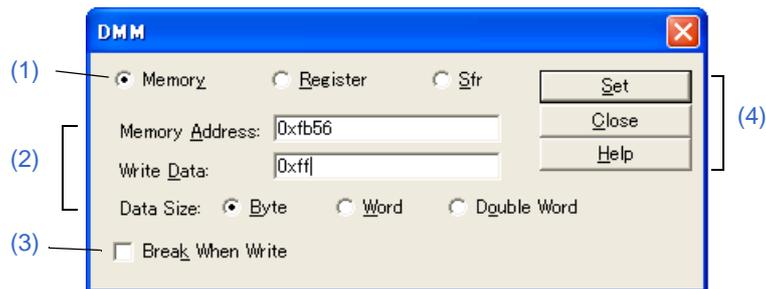
## DMM dialog box

This dialog box is used to set addresses and data for DMM (Dynamic Memory Modification). (Refer to "5.15 DMM Function".)

The memory contents are rewritten via the DMM function in real time during user program execution.

**Remark:** By selecting "Break When Write", pseudo DMM is enabled for areas for which DMM cannot be performed (execution of the user program breaks momentarily upon a write to the memory).

Figure 6-38 DMM Dialog Box



- Opening
- Explanation of each area

## Opening

Select [Edit] menu -> [DMM...], or click the [DMM...] button in the [Memory window](#) / [Register window](#) / [SFR window](#).

## Explanation of each area

### (1) DMM target selection area

This area is used to select the target for DMM. The items displayed in (2) [DMM setting area](#) change by selecting the option button.

Memory	DMM is performed for the memory.
Register	DMM is performed for the register.
Sfr	DMM is performed for the SFR.

**Remark:** If the DMM dialog box is opened via the [Memory window](#), [Register window](#), or [SFR window](#), the corresponding option button has already been selected.

**(2) DMM setting area**

The items displayed vary as follows, depending on the selection in [\(1\) DMM target selection area](#).

(a) When "Memory" is selected

Memory Address:	This area is used to specify the memory address to which data is to be written. The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to <a href="#">"Table 5-5 Specifying Symbols"</a> .)	
Write Data:	This area is used to specify the data to be written to the memory address specified in "Memory Address:".	
Data Size:	This area is used to specify the size of the data specified in "Write Data:" to be written.	
	Byte	Writes the data as 8-bit data.
	Word	Writes the data as 16-bit data.
	Double Word	Writes the data as 32-bit data.

(b) When "Register" is selected

Register Name:	This area is used to specify the register name to which data is to be written. The case is distinguished. Both functional and absolute names can be used for specification.
Write Data:	This area is used to specify the data to be written to the Register specified in "Register Name:".

(c) When "Sfr" is selected

Sfr Name:	This area is used to specify the SFR name to which data is to be written. The case is distinguished. The read-only SFRs cannot be specified.
Write Data:	SfrThis area is used to specify the data to be written to the register specified in "SFR Name:".

**(3) Break When Write**

Select this item when performing DMM via software emulation (pseudo DMM function) (this item is not selected by default). By selecting this item, a break occurs instantaneously upon a write via the DMM function during the user program execution.

Use of pseudo DMM is fixed if "Register" or "SFR" is selected in [\(1\) DMM target selection area](#) (not selectable when this item is selected).

**(4) Function buttons**

Set	Writes the data in accordance with a given condition.
Close	Closes this dialog box.
Help	Displays this dialog box online help files.

## Register window

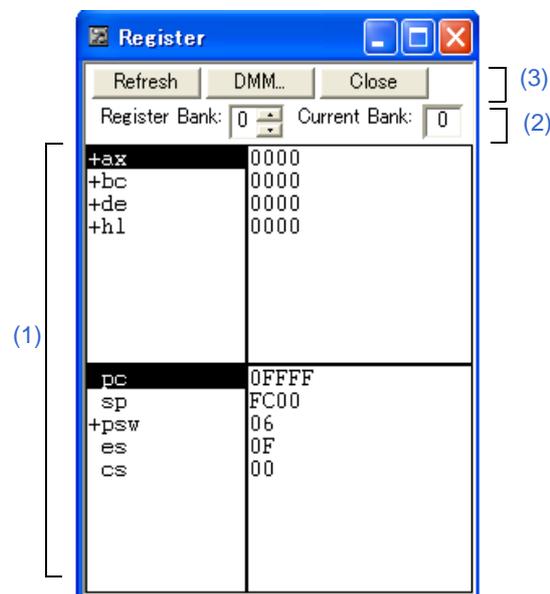
This window is used to display and change registers (general-purpose registers/control registers). (Refer to "5.8 Register Manipulation Function".)

Other operations using [Context menu](#), [Function buttons](#), etc., can be performed in this window.

Each area in this window are the jump pointer of the [Jump function](#).

**Caution:** The register contents are displayed using the method set in (1) [RAM Monitor \[IECUBE\]](#) and [RAM Monitor And DMM \[MINICUBE2\]](#) in the [Extended Option dialog box](#) user program execution. (Refer to "5.14 RRM Function".) **[IECUBE]**

Figure 6-39 Register Window



- Opening
- Explanation of each area
- [View] menu (Register window-dedicated items)
- Context menu

## Opening

Click the  **Reg** button, or select [Browse] menu -> [Register].

## Explanation of each area

### (1) Register value display /change area

The left area displays register names, and the right area displays register values.

Register values are changed through direct input. The location to be changed is displayed in **red** and the contents of the contents of the change are written into the target memory when the Enter key is pressed. During user program execution, however, the change cannot be written (if attempted, an error occurs). The previous value can be canceled by the ESC key.

The upper area displays the general-purpose registers, and the lower area displays the control registers.

The general-purpose registers display/change area	This area is used to display and change the general-purpose registers.
The control registers display/change area	This area is used to display and change the control registers. By double-clicking "+", flag name and flag value are displayed (first character changes from "+" to "-"). Expanded display is canceled by double-clicking "-" (first character changes from "-" to "+").

### (2) Register bank setting area

Register Bank:	Specifies the general-purpose register bank number displayed.
Current Bank:	Displays the general-purpose register bank number that is currently set.

### (3) Function buttons

Refresh	Updates the contents of the window with the latest data.
DMM...	Opens the <a href="#">DMM dialog box</a> .
Close	Closes this window.

## [View] menu (Register window-dedicated items)

The following items are added in the [\[View\] menu](#) , when the Register window is active.

Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays hexadecimal numbers (default).
Absolute Name	Displays register names as absolute names.
Function Name	Displays register names as function names (default).

---

## Context menu

---

Add Watch...	Registers a selected character string to the Watch window. Opens the <a href="#">Add Watch dialog box</a> .
Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays hexadecimal numbers (default).
Absolute Name	Displays register names as absolute names.
Function Name	Displays register names as function names (default).

## SFR window

This window is used to display and change the contents of SFR and the I/O ports that have been registered in the [Add I/O Port dialog box](#). (Refer to "5.8 Register Manipulation Function".)

A number of other operations using [Context menu](#), [Function buttons](#) etc., can be performed in this window.

**Caution:** However, that the values of read-only SFR and I/O ports cannot be changed. In addition, the SFR and I/O ports that cause the device to operate when they are read are read-protected and therefore cannot be read. To read these registers, select a register, and select and execute [Compulsion Read] from the [Context menu](#).

**Remark1:** During program execution, this window is displayed in accordance with the settings of (1) [RAM Monitor \[IECUBE\]](#) and [RAM Monitor And DMM \[MINICUBE2\]](#) in the [Extended Option dialog box](#). (Refer to "5.14 RRM Function".)

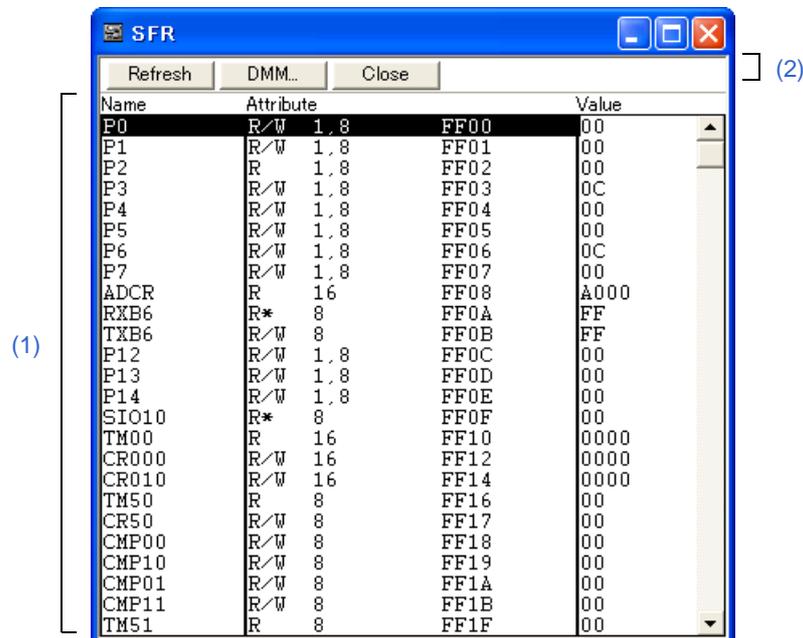
**Remark2:** The display start position when the window is opened is as follows.

First time: Display from SFR of minimum address

Second and subsequent times: Display from first SFR when window was last closed

**Remark3:** Multiple lines cannot be selected in the SFR window

Figure 6-40 SFR Window



- Opening
- Explanation of each area
- [View] menu (SFR window-dedicated items)
- Context menu

## Opening

Click the  **SFR** button, or select [Browse] menu -> [SFR].

## Explanation of each area

### (1) SFR display/change area

Name	This area displays the names of SFR and I/O ports. If the value of an I/O port address is not defined, the I/O port name displayed in light color.	
Attribute	This area displays the attributes of SFR and I/O ports. This area displays the <b>read/write attributes</b> , <b>access types</b> , and displays <b>absolute addresses</b> from the left side. When the bit SFR is displayed, bit-offset value is also displayed. It can be specified whether this area is displayed or not, by selecting [View] menu -> [Attribute].	
	Read/Write Attribute	
	R	Read only
	W	Write only
	R/W	Read/write
	*	Register that is read via an emulation register to prevent the device from operating when this register is read. To read this attribute directly from a SFR, execute [View] menu -> [Compulsion Read]. Even a write-only SFR can also be read via an emulation register. However, some devices do not support this function.
	Access Type	
	1	Can be accessed in Bit units.
	8	Can be accessed in Byte units.
	16	Can be accessed in Word units.
Value	This area is used to display and change the contents of a SFR and I/O port. Values are changed through direct input. The location to be changed is displayed in <b>red</b> and the contents of the contents of the change are written into the target memory when the Enter key is pressed. During user program execution, however, the change cannot be written (if attempted, an error occurs). The previous value can be canceled by the ESC key. Note that the values of read-only SFR and I/O ports cannot be changed. The value of read-protected SFR and I/O ports can be read by selecting <b>Context menu</b> -> [Compulsion Read]. The contents are displayed differently as follows, depending on the attribute:	
	Black Display	Read only or read/write
	--	Write only
	**	Value changes if read

**(2) Function buttons**

Refresh	Updates the contents of this window with the latest watch data.
DMM...	Opens the <a href="#">DMM dialog box</a> .
Close	Closes this window.

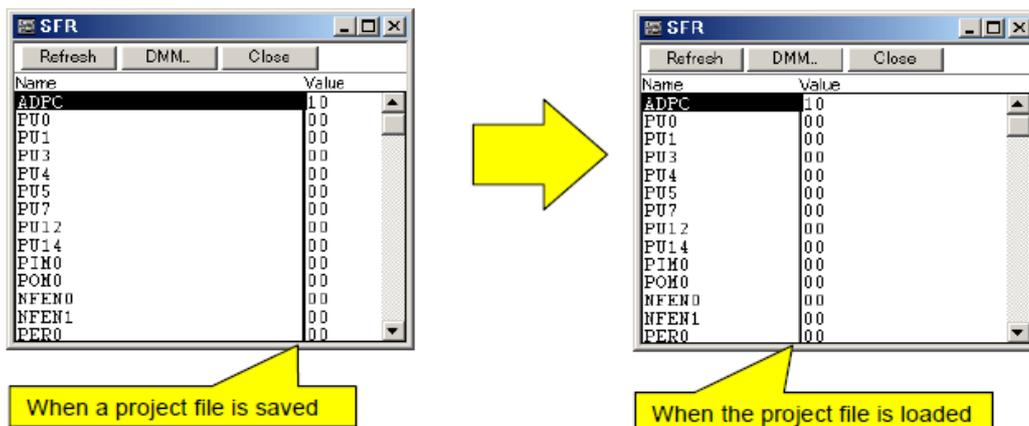
**[View] menu (SFR window-dedicated items)**

When this window is the current window, the following items are added on [\[View\] menu](#).

Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays octal numbers. (default)
Sort By Name	Displays in alphabetical order.
Sort By Address	Displays in address order. (default)
Unsort	Does not sort.
Attribute	Switches on/off display of "Attribute" <sup>Note</sup> .
Pick Up	Displays only the registers selected in the <a href="#">SFR Select dialog box</a> .
Select...	Opens the <a href="#">SFR Select dialog box</a> .
Compulsion Read	Forcibly reads the SFR that are disabled from being read because their values will be changed, or the data of the I/O ports and I/O protect area added in the <a href="#">Add I/O Port dialog box</a> .

**Note:** If the [Attribute] area is hidden, the information of the vertical boundary position in the SFR window is not stored into the project file. As a result, the vertical boundary position will not be restored when the project file is loaded.

Figure 6-41 Example of SFR Window



## Context menu

Move...	Opens the <a href="#">Address Move dialog box</a> . <b>Note</b>
Add Watch...	Opens the <a href="#">Add Watch dialog box</a> .
Add I/O Port...	Opens the <a href="#">Add I/O Port dialog box</a> .
Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays octal numbers.(default)
Sort By Name	Displays in alphabetical order.
Sort By Address	Displays in address order. (default).
Unsort	Does not sort.
Attribute	Switches on/off display of "Attribute".
Pick Up	Displays only the registers selected in the <a href="#">SFR Select dialog box</a> .
Select...	Opens the <a href="#">SFR Select dialog box</a> .
Compulsion Read	Forcibly reads the SFR that are disabled from being read because their values will be changed, or the data of the I/O ports and I/O protect area added in the <a href="#">Add I/O Port dialog box</a> .

**Note:** Open the Address Move dialog box.

If an SFR that satisfied the two condition shown below exists when an SFR name is input in the Address Move dialog box, the focus does not move to the specified SFR address but may move to the SFR that satisfied these two conditions.

- An SFR whose address is the same as that of the input SFR
- An SFR whose name is different from that of the input SFR

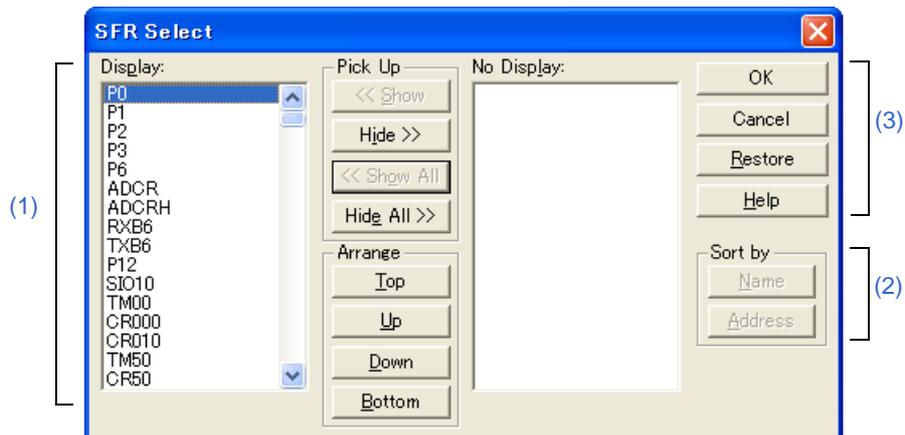
SFR names that satisfy the above two conditions include the I/O port names added in the Add I/O Port dialog box.

## SFR Select dialog box

This dialog box is used to select SFR and I/O ports that are displayed the [SFR window](#). (Refer to "5.8 Register Manipulation Function".)

It is also used to specify the sequence in which registers and ports are displayed.

Figure 6-42 SFR Select Dialog Box



- Opening
- Explanation of each area

## Opening

When the [SFR window](#) is the current window, select [View] menu -> [Select...].

## Explanation of each area

### (1) Displayed SFR selection/display order change area

Display:	Registers displayed in the <a href="#">SFR window</a> . Two or more registers can be selected by clicking any of the above buttons while holding down the Ctrl or Shift key.	
Pick Up	The following buttons are used to change SFR to be displayed.	
	<< Show	Moves the register selected from the "No Display:" list to "Display:".
	Hide >>	Moves the register selected from the "Display:" list to "No Display:".
	<< Show All	Moves all registers to "Display:".
	Hide All >>	Moves all registers to "No Display:".
Arrange	The following buttons are used to change the display sequence in "Display:". If the display arrangement is changed, multiple lines cannot be selected. Select one line at a time.	
	Top	Moves the selected register to the top of the list.
	Up	Moves the selected register one line up.
	Down	Moves the selected register one line down.
	Bottom	Moves the selected register to the bottom of the list.
No Display:	Registers not displayed in the <a href="#">SFR window</a> .	

### (2) Display order change buttons

Sort by	The following buttons are used to change the display sequence in "No Display:".	
	Name	Displays in alphabetical order.
	Address	Displays in address order.

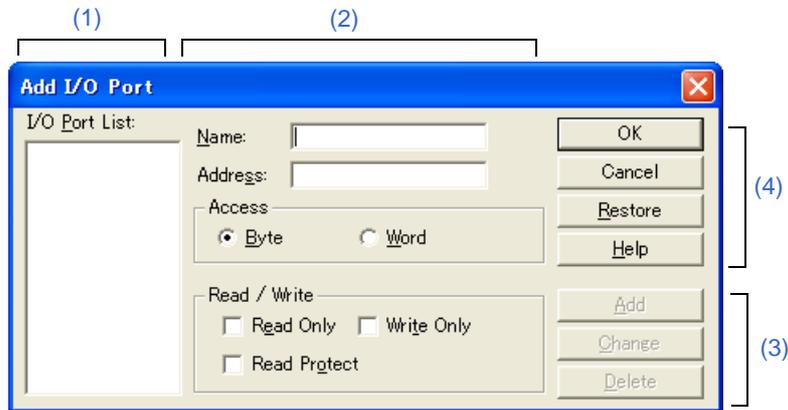
### (3) Function buttons

OK	Reflects the selection in this dialog box in the <a href="#">SFR window</a> and closes this dialog box.
Cancel	Cancels the changes and closes this dialog box.
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

## Add I/O Port dialog box

This dialog box is used to register an I/O port to be added to the [SFR window](#). (Refer to "[5.8 Register Manipulation Function](#)".)

Figure 6-43 Add I/O Port Dialog Box



- [Opening](#)
- [Explanation of each area](#)

### Opening

Select [Option] menu -> [Add I/O Port...].

### Explanation of each area

#### (1) I/O Port List:

This area lists the I/O ports currently registered.

If a new I/O port is registered, it is added to this list. An I/O port already registered can be selected and changed or deleted by (3) [Buttons](#).

**(2) I/O port specification area**

Name:	This area is used to specify an I/O port name to be added (up to 15 characters long).	
Address:	This area is used to specify the address of the I/O port to be added. The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol. (Refer to <a href="#">"Table 5-5 Specifying Symbols"</a> .) The address that can be set in this area is either a Target area address or SFR area address.	
Access	This area is used to select the access size of the I/O port to be added.	
	Byte	8-bit unit (default)
	Word	16-bit unit
Read / Write	This area is used to specify the access attribute of the I/O port to be added. In the default condition, all the attributes are cleared (i.e., the I/O port can be both read and written).	
	Read Only	Read Only
	Write Only	Write only
	Read Protect	Read-protected

**(3) Buttons**

Add	Adds an I/O port of the specified address.
Change	Changes the setting of the I/O port selected in "I/O Port List:".
Delete	Deletes the I/O port selected in "I/O Port List:".

**(4) Function buttons**

OK	Reflects the results of addition in the <a href="#">SFR window</a> and closes this dialog box.
Cancel	Cancels the changing, closes this dialog box.
Restore	Restores the original status.
Help	Displays this dialog box online help files.

## Timer dialog box

[IECUBE]

This dialog box is used to register and set timer event conditions, and display execution time measurement results. (Refer to "5.12 Event Function" and "5.9 Timer Function [IECUBE].")

The "Execution time display area" can be constantly displayed as the [Timer Result dialog box](#) by clicking the [View Always] button.

Registration and setting of timer event conditions is done by setting each item (256 items max.) in this dialog box and then clicking the [OK] button. The registered timer event conditions are managed by the [Event manager](#).

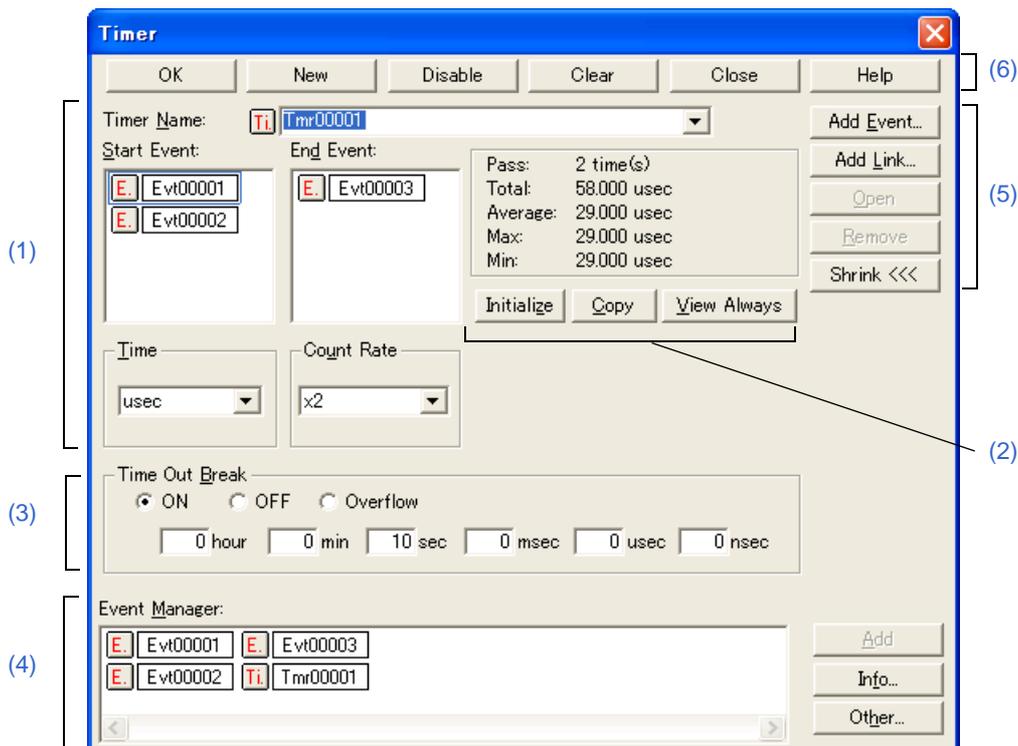
The number of timer event conditions that can be simultaneously used (validated) is limited. (Refer to "5.12.4 Number of enabled events for each event condition".)

The execution time measurement result is displayed when the set timer event condition is selected.

**Remark1:** The measurement result display contents are updated at each sampling time of the [RRM Function](#), even during user program execution.

**Remark2:** Timer event condition setting/enable/disable/delete operations are possible even during user program execution.

Figure 6-44 Timer Dialog Box



- [Opening](#)
- [Explanation of each area](#)

## Opening

Click the  **Tim** button, or select [Event] menu -> [Timer...].

## Explanation of each area

### (1) Timer event condition setting area

Timer Name:	<p>This area is used to set a timer event name. Directly input an alphanumeric string of up to eight characters as a name. To display the contents of an already created event condition, select from the drop-down list. To display from user program execution until break, specify "Run-Break". (Refer to <a href="#">"5.9.2 Run-Break event"</a>.) The mark on the left of this area indicates the utilization status of events. (Refer to <a href="#">"Table 5-18 Event Icon"</a>.) The gray mark indicates that an event condition is being edited and has not been registered yet. By clicking the left mark, an event condition can be validated or invalidated.</p>
Start Event: End Event:	<p>This area is used to set an event condition for the timer. The number of event conditions that can be set in this area is refer to <a href="#">"Table 6-11 Number of Events Settable"</a>. Setting of event conditions is easily done by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to <a href="#">"5.12.3 Setting event conditions"</a>.</p>
Time	<p>This area is used to select the unit in which the <a href="#">(2) Execution time display area</a> is to be displayed.</p>
Count Rate	<p>This area is used to set the timer count rate value used for execution time measurement. The timer count rate value can be set for each timer event condition. The rate value can be selected from x1 to x2048 from a drop-down list. For Run-Break events, this menu is fixed to x1 and selection is unavailable.</p>

### (2) Execution time display area

This area displays the result of measuring the execution time of the program.

Only "Total" is displayed for Run-Break.

Measurement results that cannot be trusted due to counter overflow are displayed in red.

Pass	Number of passes
Total	Total execution time in the measurement zone specified by start event and end event conditions (Up to 2,562,047 hours 47 minutes 16 seconds can be displayed.)
Average	Average execution time
Max	Maximum execution time
Min	Minimum execution time
[Initialize]	Clears the measurement results.
[Copy]	Copies the measurement result to the clipboard in text format.
[View Always]	Opens the <a href="#">Timer Result dialog box</a> .

Table 6-6 Measurable Values

Connected IE	Measurable Execution Time	Measurable Execution Count
<b>[IECUBE]</b> For zone measurement	1 minute 13 seconds max. (Resolution: 17 nsec) 40 hours 43 minutes max. (Resolution: 34 usec)	32 bit 4,294,967,295 times max.
<b>[IECUBE]</b> For Run-Break	1 minute 13 seconds max. (Resolution: 17 nsec) 40 hours 43 minutes max. (Resolution: 17 nsec)	- -

**Remark:** The time is displayed with resolution at 34 msec during user program execution. After the user program is stopped, the time is displayed with resolution at 17 nsec.

### (3) Time Out Break

This area is used to set the timeout break for the section measurement time specified in "Start Event:, End Event:" (time from the establishment of timer start event to the establishment of timer end event).

ON	A timeout break occurs (execution is terminated) if the section measurement time exceeds the specified timeout time. Specify the time-out time in the text boxes. The values up to the maximum measurable time can be specified.
OFF	No timeout break occurs. (default)
Overflow	A timeout break occurs (execution is terminated) if the section measurement time exceeds the maximum measurable time. (Refer to " <a href="#">Table 6-6 Measurable Values</a> ".)

**Caution:** "ON" cannot be selected for Run-Break event, and the time-out break cannot be set to specify the time-out time (the time-out break can be set when "Overflow" is selected).

### (4) Event manager area

This area displays the list of registered events.

Each event condition can be set easily just by dragging and dropping the event icon displayed in this area onto the event setting area in each event setting dialog box. (Refer to "[5.12.3 Setting event conditions](#)".)

This area is common to all event-related dialog boxes. (Refer to "[\(2\) Event manager area](#)" in the "[Break dialog box](#)".)

### (5) Function buttons (for event condition contents display, etc.)

These buttons are used to display or delete the event conditions displayed in the event condition setting area, and to display or hide the Event manager area.

This area is common to all event-related dialog boxes. (Refer to "[\(3\) Function buttons \(for event condition contents display, etc.\)](#)" in the "[Break dialog box](#)".)

**(6) Function buttons (for registering, deleting, validating, and invalidating event conditions)**

These buttons are used to register, delete, validate, and invalidate the events.

The event with the specified event condition is registered or set (validated) by clicking the [OK] (or [Set]) button. This area is common to all event-related dialog boxes. (Refer to "[\(4\) Function buttons \(for registering, deleting, validating, and invalidating event conditions\)](#)" in the "[Break dialog box](#)".)

## Trace View window

[IECUBE]

This window used to display the trace results. (Refer to "5.10 Trace Function [IECUBE]").

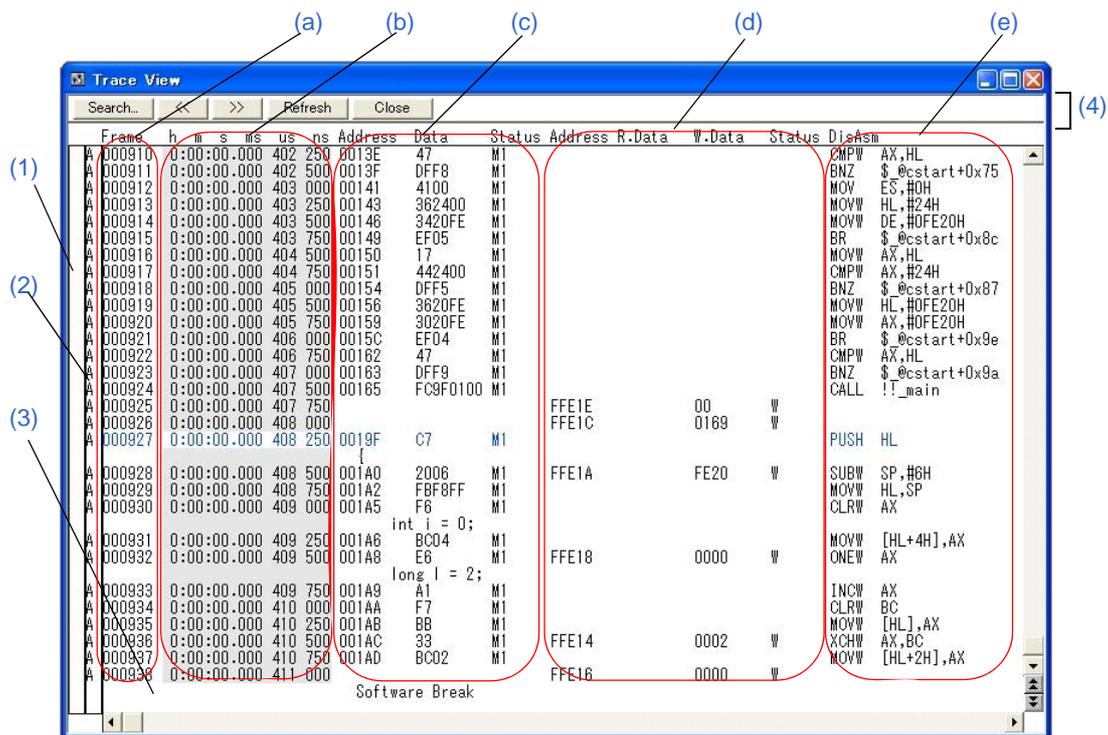
Display updates are performed during breaks or during step execution.

This window has [Mixed display mode \(Trace window\)](#). Also, It has "5.17.3 Trace result with linking window [IECUBE]".

A number of other operations using [Context menu](#), [Function buttons](#), etc., can be performed in this window.

**Remark:** To reference the trace data during program execution, stop the trace function by selecting the [Run] menu --> [Tracer Stop]/

Figure 6-45 Trace View Window



- Opening
- Explanation of each area
- [View] menu (Trace View window-dedicated items)
- Context menu

## Opening

Click the  TrW button, or select [Browse] menu -> [Trace].

## Explanation of each area

### (1) Point mark display area

This area displays the [Event Setting Status \(Event Mark\)](#).

If an execution event or access event is set at the corresponding trace address, the mark corresponding to the type of the event is displayed.

The mark displayed is not that during trace but an event mark that is set when the trace result is displayed.

### (2) Trace mode display area

This area displays the type of trace mode.

A	Unconditional traced or section traced frame
Q	Qualify traced frame
S	Step execute frame
T	Delay trigger frame
M	DMA access frame

**Caution:** The Help for the Trace View window is not displayed even if the F1 key on the keyboard is pressed while no trace result is displayed. In such a case, select the [Help] menu --> [Current Window] to open the help,

### (3) Trace result display area

This area displays the trace results.

Whether each of the following sub-areas is displayed or not can be selected in the [Trace Data Select dialog box](#).

(a) Frame	This area displays the trace flame number.
(b) Time display area	<p>This area displays how many clocks the target chip has required since execution of the preceding trace contents was started until execution of the current trace contents is started. In addition, the division rate of the number of counts can be selected up to 1M in the <a href="#">Extended Option dialog box</a>. The accumulative display can be set.</p> <p>For the relationship between a division ratio of the time tag counter and the maximum measurable time, refer to "<a href="#">Table 6-2 Relationship Between Division Ratio and Maximum Measurable Time (Time Tag Count)</a>".</p> <p>The time tag and the time during trace are measured with an external clock (60 MHz).</p> <p><b>Caution:</b> If the timer overflows, the maximum value of the time tag is displayed in red.</p>

(c) Fetch access display area <sup>Note1</sup>	This area displays the result of fetching the program and snapshot contents. The indication in the "Address" and "Data" columns varies depending on the "Status" display.	
Address	When "Status" type is Program fetch: Displays the fetched address. When "Status" type is Snap (SNAP): Displays the register names and the memory addresses.	
Data	When "Status" type is Program fetch: Displays the fetched data. When "Status" type is Snap (SNAP): Displays the register values and the memory contents.	
Status	The status has two types: Program fetch and Snap.	
	<b>Program fetch</b>	
	M1	Fetching of first byte of instruction If the fetch address is the start of the symbol, the first line is highlighted in blue.
	OP	Opecode fetch of 2nd and subsequent bytes
	IF	Invalid fetch or status unknown
	<b>Snap</b>	
	SNAP	Snap
	<b>VECT</b>	
	VECT	A vector read (Not shown in the case of a software interrupt) This line is highlighted in orennge
(d) Data access display area	This area displays the result of accessing data.	
Address	Displays access address	
R.Data W.Data	Displays access data	
Status	Displays access status	
	RW	Data read/write
	R	Data read
	W	Data write
(e) DisAsm <sup>Note2</sup>	This area displays the disassemble results. (This item is displayed only when "M1" is displayed in the "Status" column.) Global symbols are displayed for addresses that have been accessed and frames in which only data is displayed.	

**Note1:** If "Timetag" is selected in "(3) Trace Data Priority [IECUBE]" in the [Extended Option dialog box](#), fetch data is displayed in accordance with the memory contents of the time at which trace data is displayed. Therefore, the fetch data at the time of user program execution may not be displayed (in such a case than flash self programming is performed).

**Note2:** If "Timetag" is selected in "(3) Trace Data Priority [IECUBE]" in the [Extended Option dialog box](#), fetch data is displayed in accordance with the memory contents of the time at which trace data is displayed. Therefore, the disassemble result at the time of user program execution may not be displayed (in such a case than flash self programming is performed).

The causes of stopping tracer are as follows:

Table 6-7 Reset Causes During Trace Operation

Cause	Meaning
Internal Reset	Reset by peripheral chip
External Reset	Reset by target
Pseudo Emulation Reset	Reset due to pseudo emulation
POC Reset	Reset due to POC

These are displayed when the tracer is stopped.

If there are multiple break causes, all of the causes are displayed.

Table 6-8 Break Causes When Tracer Is Stopped (Status)

Cause	Meaning
Event Break	Break by event
Trace Full Break	Break because trace memory is full
Trace Delay Break	Break by trace delay
Non Map Break	Non-mapped area is accessed.
Write Protect	An attempt has been made to write to a write-protected area
SFR Read Protect	Read from prohibited SFR was attempted
SFR Write Protect	Write to write-prohibited SFR was attempted
Step Break	Step execution break
Manual Break	Manual break
Stack Overflow	Break by stack overflow
Stack Underflow	Break by stack underflow
Uninitialize Stack Pointer	Break due to failure to perform stack pointer initialization
Software Break	Break by software break
Uninitialize Memory read	Memory not initialized has been read
Timer Over Break	Break because execution time is over
Unspecified Illegal	Others
Fetch Guard	Fetch guard break
Trace Stop	Trace stop
IMS(IXS) Illegal	IMS, IXS Illegal break
Flash Illegal	Flash Illegal break
Retry Over	RETRY count over break
Peripheral Break	Break from peripheral

Cause	Meaning
Before Execution	Before Execution
Word Miss-align Access	A word access for an odd number address was made

#### (4) Function buttons

Search...	Opens the <a href="#">Trace Search dialog box</a> and searches trace results. The searched result will be highlighted in the Trace View window. Same function as [View] menu -> [Search...].
<<	Searches forward (upward on screen) for a trace result that satisfies the search condition set in the <a href="#">Trace Search dialog box</a> . This button cannot be selected during pickup display.
>>	Searches backward (downward on screen) for a trace result that satisfies the search condition set in the <a href="#">Trace Search dialog box</a> . This button cannot be selected during pickup display.
Refresh	Updates the contents of the window with the latest data.
Close	Closes this dialog box.

#### [View] menu (Trace View window-dedicated items)

The following items are added in the [\[View\] menu](#), when the Trace window is active.

Select...	Selects the contents to be displayed. Opens the <a href="#">Trace Data Select dialog box</a> .
Pick Up	Sets for pickup display.
Off	Does not pick up and display (default).
Search	Picks up and displays a frame that satisfies the search condition.
Snap	Picks up and displays a snap frame.
Mix	Specifies whether the source file are displayed in mixed display mode, or not displayed. Checked: Mixed display Not checked: Hidden (default)
Window Synchronize	Links the <a href="#">Trace View window</a> with the following windows: (Refer to " <a href="#">5.17.3 Trace result with linking window [IECUBE]</a> ".) A selected window is linked.
Source Text	Links the <a href="#">Source window</a> .
Assemble	Links the <a href="#">Assemble window</a> .
Memory	Links the <a href="#">Memory window</a> .

## Context menu

Move...	Moves the display position. Opens the <a href="#">Trace Move dialog box</a> .
Trace Clear [IECUBE]	Clears the trace data.
Select... [IECUBE]	Selects the contents to be displayed. Opens the <a href="#">Trace Data Select dialog box</a> .
Pick Up [IECUBE]	Sets for pickup display.
Off	Does not pick up and display (default).
Search	Picks up and displays a frame that satisfies the search condition.
Snap	Picks up and displays a snap frame.
Mix	Specifies whether the source file are displayed in mixed display mode, or not displayed. Checked: Mixed display Not checked: Hidden (default)
Window Synchronize [IECUBE]	Links the Trace View window with the following windows: (Refer to " <a href="#">5.17.3 Trace result with linking window [IECUBE]</a> ".)
Source Text	Links the <a href="#">Source window</a> .
Assemble	Links the <a href="#">Assemble window</a> .
Memory	Links the <a href="#">Memory window</a> .
Source Text	Displays the corresponding source text and source line, using the data value at the cursor position as the jump destination address. (Refer to " <a href="#">5.17.2 Jump function</a> ".) If no line information exists at the jump destination address, however, you cannot jump. Opens the <a href="#">Source window</a> . If the <a href="#">Source window</a> in active is open, that window is displayed in the forefront (so that it can be manipulated).
Assemble	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position. (Refer to " <a href="#">5.17.2 Jump function</a> ".) Opens the <a href="#">Assemble window</a> . If the <a href="#">Assemble window</a> in active is open, that window is displayed in the forefront (so that it can be manipulated).
Memory	Displays the memory contents starting from the jump destination address specified by the data value at the cursor position. (Refer to " <a href="#">5.17.2 Jump function</a> ".) Opens the <a href="#">Memory window</a> . If the <a href="#">Memory window</a> in active is open, that window is displayed in the forefront (so that it can be manipulated).

## Trace Search dialog box

[IECUBE]

This dialog box is used to search in the [Trace View window](#). (Refer to "5.10 Trace Function [IECUBE]".)

By setting each item and then clicking the [Find Next] button, searching can be started.

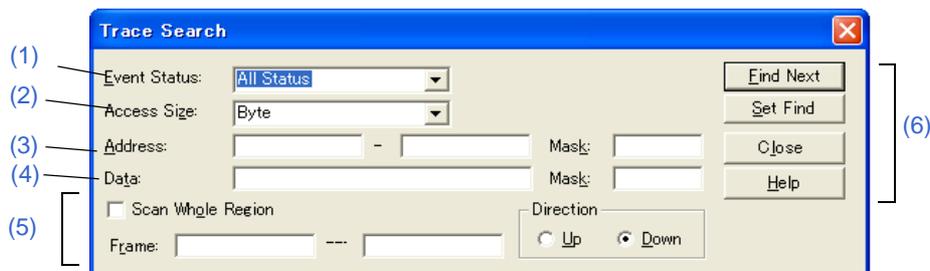
By clicking the [Set Find] button, the direction buttons (<< and >>) in the [Trace View window](#) can be used for the search.

**Remark:** This dialog box is used to search trace data if it is opened by selecting [View] menu -> [Pick Up] -> [Off]. It is used to pick up and display trace data if it is opened by selecting [View] menu -> [Pick Up] -> [Search].

**Caution1:** This dialog box cannot be called if picking up the first M1 fetch frame (BRM1) after program branch is specified using the menubar or in the [Trace Data Select dialog box](#).

**Caution2:** Search can be performed during program execution by executing the Trace Stop command.

Figure 6-46 Trace Search Dialog Box



- [Opening](#)
- [Explanation of each area](#)

## Opening

When the [Trace View window](#) is the current window, select [View] menu -> [Search...], or click the [Search... ] button in the same window.

## Explanation of each area

### (1) Event Status:

This area is used to select a status condition.

If a status condition is omitted, all frames (All status) are searched.

All status	All frames (default)
M1 Fetch	M1 fetch
R/W	Data read/write (Including Read, Write)
Read	Data read
Write	Data write

### (2) Access Size:

This area is used to select an access size condition.

By specifying an access size condition, the access width of a data condition to be detected by an access event is determined.

Byte	Searches for a data condition with 8-bit width (only during 8-bit access).
Word	Searches for a data condition with 16-bit width (only during 16-bit access).
No Condition	Does not search based on access size (nothing can be input to "Data" area).
Bit	Searches for a data condition with 1-bit width (only during 8-bit access)

**Note1:** If an access event is specified as a status condition, the alternative of Bit is not displayed. If Bit or 1 is specified, an error occurs.

**Note2:** In this case, a search is made for a data condition with 1-bit width. Because of the operation of the simulator, access to a bit is not directly detected; the simulator searches a dummy bit access by internally setting address conditions and data conditions as follows:

**Input example:**

Address: FE20.1 ---->  
Data: 1

**Setting of trace search:**

Address: FE20  
Data: 00000010B  
Mask: 11111101B

If another bit of the same address is accessed or if all the 8 bits of the same address are accessed, therefore, an event is detected in accordance with the specified status if the address and bit match the specified value of [address.bit].

**Remark:** If no access size condition is specified, a judgment is automatically made from the address condition and data condition, and the following is set:

- Bit if the address condition is set in bit units
- Byte if the data condition is set in 8-bit units
- No Condition if no data condition is specified

**(3) Address condition setting area**

This area is used to specify an address condition (may be omitted). The following can be set:

Table 6-9 Settable Range of Address Condition (Trace)

Settable Range	Condition
0 <= address value <= 0xFFFFF	None
0 <= mask value <= 0xFFFFF	None

Address:	Set an address condition (lower address - higher address) (may be omitted). The default radix for inputting a numeric value is hexadecimal. A symbol can be also specified by a symbol or expression. (Refer to "Table 5-5 Specifying Symbols".) The following can be set:	
	Setting as a point	Set a value to only the lower address, or set the same value to the lower address and the higher address.
	Setting as a range	Set a value to only the lower address, or set the same value to the lower address and the higher address.
	Setting as a bit	Set a value to only the lower address, or set the same value to the lower address and the higher address. Specify a value in the form of "address.bit". Mask cannot be set. The value of bit, which indicates the bit position, must be 0 <= bit <= 7.
Mask:	Set a mask value for an address value (only when "Setting as a point") (may be omitted). The address value of a bit whose mask value is 1 may be 0 or 1.	

**Example1: 0x4000 to 0x40FF satisfy the condition.**

Address:	0x4000 to 0x4000
Mask:	0xFF

**Example2: 0x4000, 0x4001, 0x4100, and 0x4101 satisfy the condition.**

Address:	0x4000 to 0x4000
Mask:	0x101

**(4) Data condition setting area**

This area is used to set data conditions (may be omitted).

The settable range differs as follows depending on the access size condition specified in (2) [Access Size](#):. (Refer to "(5) [Data condition setting area](#)" in the " [Event dialog box](#)".)

Data:	Set a data value as data conditions. The default radix for inputting a numeric value is hexadecimal. A data can be also specified by a symbol. (Refer to " <a href="#">Table 5-5 Specifying Symbols</a> ".)
Mask:	Set a mask value for the data value (may be omitted). When a mask is set, the data value for the bit whose mask value is 1 may be 0 or 1.

**Example1: 0x4000 to 0x40FF satisfy the condition.**

Data	0x4000
Mask	0xFF

**Example2: 0x4000, 0x4001, 0x4100, and 0x4101 satisfy the condition.**

Data	0x4000
Mask	0x101

**(5) Search condition setting area**

Scan Whole Region	This should be selected to search the entire specified range.	
Direction	This area is used to specify the direction of the search.	
	Up	Forward search. Searches data forward (upward on screen) from the current position of the cursor.
	Down	Backward search. Searches data backward (downward on screen) from the current position of the cursor. (default)
Frame:	This area is used to specify a frame number to be searched. The default radix for inputting a numeric value is decimal. A symbol can be also specified by <a href="#">Frame Number Specification Format</a> .	

**(6) Function buttons**

Find Next	Searches the specified data in accordance with a given condition. If the specified frame is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.
Pick Up (Stop (during search))	Picks up according to the specified condition of data search. If a frame that satisfies the condition is found as a result of a search, it is picked up. To pick up a frame that satisfies a different condition, click this button again.
Close	Closes this dialog box.
Help	Displays this dialog box online help files.

## Timer Result dialog box

[IECUBE]

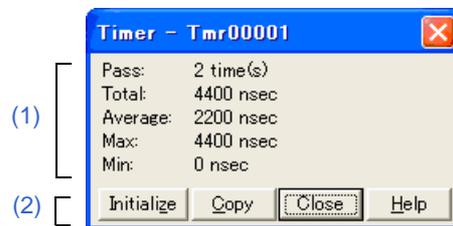
This dialog box displays the results of measuring the execution time. (Refer to "5.9 Timer Function [IECUBE].")

By clicking the [View Always] button in the [Timer dialog box](#), this dialog box is opened corresponding to a timer event condition on a one-to-one basis. Two or more of this dialog box can be simultaneously opened.

Up to 256 + 1 (Run-Break event) Timer Result dialog boxes can be opened, the number of events that can be measured at the same time is the number of valid events described in "5.12.4 Number of enabled events for each event condition" + 1 (Run-Break event).

**Remark:** The measurement result display contents are updated at each sampling time of the [RRM Function](#), even during user program execution.

Figure 6-47 Timer Result Dialog Box



- [Opening](#)
- [Explanation of each area](#)

## Opening

Select a timer event condition in the [Timer dialog box](#), click the [View Always] button.

## Explanation of each area

### (1) Execution time display area

Same area is [Timer dialog box](#).

Pass:	Number of passes
Total:	Total execution time in the measurement zone specified by start event and end event conditions
Average:	Average execution time
Max:	Maximum execution time
Min:	Minimum execution time

**(2) Function buttons**

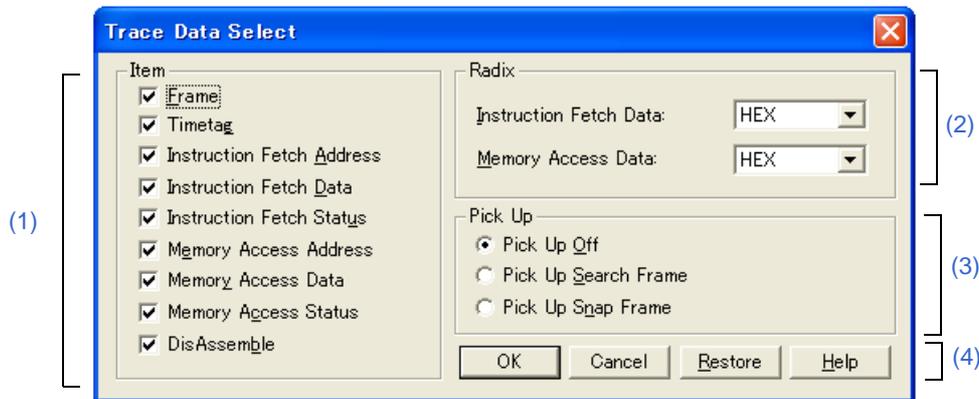
Initialize	Clears the measurement results.
Copy	Copies the measurement result to the clipboard in text format.
Close	Closes this dialog box.
Help	Displays this dialog box online help files.

## Trace Data Select dialog box

[IECUBE]

This dialog box is used to select items to be displayed in the [Trace View window](#). (Refer to "[5.10 Trace Function \[IECUBE\]](#)".)

Figure 6-48 Trace Data Select Dialog Box



- [Opening](#)
- [Explanation of each area](#)

### Opening

When the [Trace View window](#) is the current window, select [View] -> [Select...] menu.

### Explanation of each area

#### (1) Item

This area is used to select items to be displayed in the [Trace View window](#). Displaying the following fields may or may not be selected. The field selected is displayed. (Refer to "[\(3\) Trace result display area](#)".)

Frame	"Frame" Field
Timetag	"h m s ms us ns" Field
Instruction Fetch Address	"Address" ((c) <a href="#">Fetch access display area</a> <sup>Note1</sup> ) Field
Instruction Fetch Data	"Data" ((c) <a href="#">Fetch access display area</a> <sup>Note1</sup> ) Field
Instruction Fetch Status	"Status" ((c) <a href="#">Fetch access display area</a> <sup>Note1</sup> ) Field
Memory Access Address	"Address" ((d) <a href="#">Data access display area</a> ) Field
Memory Access Data	"R.Data W.Data" ((d) <a href="#">Data access display area</a> ) Field
Memory Access Status	"Status" ((d) <a href="#">Data access display area</a> ) Field
DisAssemble	"DisAsm" Field

**(2) Radix**

This area is used to select the radix in which data is to be displayed. Displaying the following items may or may not be selected.

Instruction Fetch Data	"Data" ((c) <a href="#">Fetch access display area</a> <sup>Note1</sup> ) Field
Memory Access Data	"R.Data W.Data" ((d) <a href="#">Data access display area</a> ) Field

HEX	Displays hexadecimal numbers. (default)
DEC	Displays decimal numbers.
OCT	Displays octal numbers.
Bin	Displays binary numbers.

**(3) Pick Up**

This area is used to select a pick up condition.

Pick Up Off	No pick up display (default)
Pick Up Search Frame	Picks up and displays a frame that satisfies the search condition.
Pick Up Snap Frame	Picks up and displays a snap frame.

**(4) Function buttons**

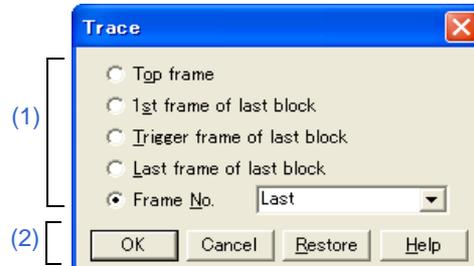
OK	Reflects the results of selection in this dialog box in the <a href="#">Trace View window</a> .
Cancel	Closes this dialog box.
Restore	Restores the original status.
Help	Displays this dialog box online help files.

## Trace Move dialog box

[IECUBE]

This dialog box is used to specify the position from which displaying the [Trace View window](#) is started. (Refer to "[5.10 Trace Function \[IECUBE\]](#)".)

Figure 6-49 Trace Move Dialog Box



- [Opening](#)
- [Explanation of each area](#)

## Opening

When the [Trace View window](#) is the current window, select [View] menu -> [Move...].

## Explanation of each area

### (1) Frame selection area

This area is used to specify the frame at the destination.

Top frame	Moves the display start position to a first frame of trace data.
1st frame of last block	Moves the display start position to a first frame in the newest block frame of trace data.
Trigger frame of last block	Moves the display start position to the trigger frame in the newest block frame of trace data.
Last frame of last block	Moves the display start position to the last frame of trace data.
Frame No.	Moves the display start position to the specified frame number. (Refer to " <a href="#">Table 6-10 Frame Number Specification Format</a> ".) In the default condition, the character string selected in the window that called this dialog box or "Last" is selected. The default radix for inputting a numeric value is decimal. If 0 is specified, the display start position is moved to the first frame of trace data. Up to 16 input histories can be recorded.

Table 6-10 Frame Number Specification Format

Specification	Abbreviation	Contents
<b>+numeric value</b>	None	Moves backward (downward on screen) the display start position from the frame at the cursor by the specified number of frames (numeric value).
<b>-numeric value</b>	None	Moves forward (upward on screen) the display start position from the frame at the cursor by the specified number of frames (numeric value).
<b>Top</b>	O	Same as "Top frame"
<b>First</b>	S	Same as "1st frame of last block"
<b>Trigger</b>	T	Same as "Trigger frame of last block"
<b>Last</b>	L	Same as "Last frame of last block"

**(2) Function buttons**

OK	Starts trace display from the specified position.
Cancel	Closes this dialog box.
Restore	Restores the input data to the original status.
Help	Displays this dialog box online help files.

## Trace dialog box

[IECUBE]

This dialog box is used to register, set, and display trace event conditions. (Refer to "5.12 Event Function", "5.10 Trace Function [IECUBE]".)

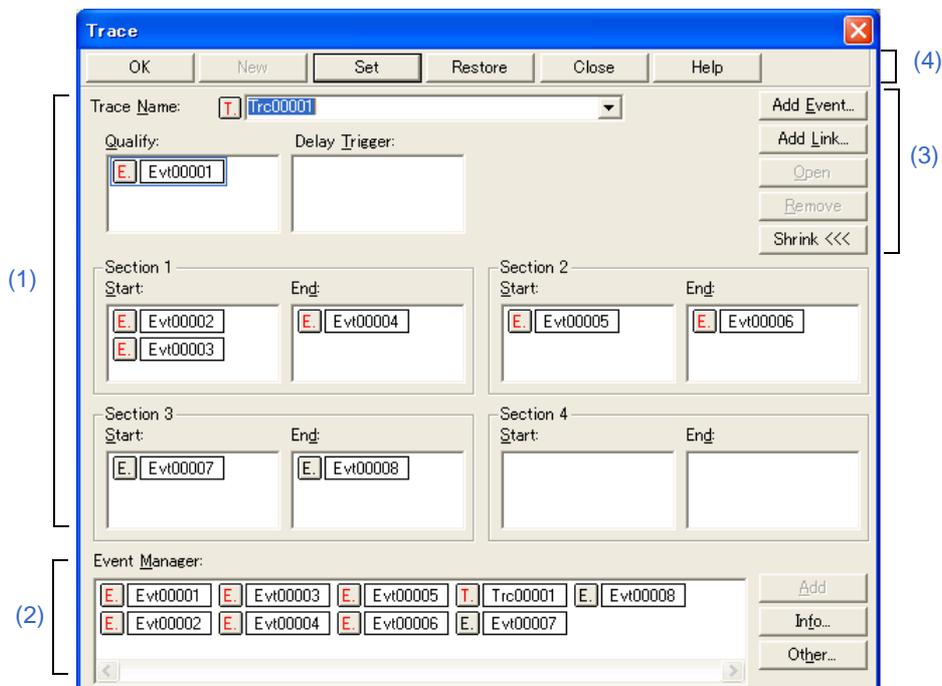
The trace event conditions for when performing conditional trace are specified in this dialog box. (Refer to "Table 5-13 Types of Conditional Trace".)

Registration and setting of trace event conditions is done by setting each item (256 items max.) in this dialog box and then clicking the [OK] button. The registered trace event conditions are managed by the [Event manager](#).

The number of trace event conditions that can be simultaneously used (validated) is limited. (Refer to "5.12.4 Number of enabled events for each event condition".)

**Remark:** Trace event condition setting/enable/disable/delete operations are possible even during user program execution. In this case, the tracer operation is momentarily stopped during manipulation.

Figure 6-50 Trace Dialog Box



- Opening
- Explanation of each area

## Opening

Click the  Trc button, or select [Event] menu -> [Trace...].

## Explanation of each area

### (1) Trace event condition setting area

Trace Name:	This area is used to set a trace event name. Directly input an alphanumeric string of up to eight characters as a name. To display the contents of an already created event condition, select from the drop-down list. The mark on the left of this area indicates the utilization status of events. (Refer to "Table 5-18 Event Icon".) The gray mark indicates that an event condition is being edited and has not been registered yet. By clicking the left mark, an event condition can be validated or invalidated.
Qualify:	This area is used to set an event condition for a qualify trace. (Refer to "5.10.5 Setting conditional trace".) If two or more events are set, trace is performed when each event occurs. Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions". The number of event conditions that can be set in this area is refer to "Table 6-11 Number of Events Settable".
Delay Trigger:	This area is used to set an event condition for a delay trigger. (Refer to "5.10.5 Setting conditional trace".) Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions". The number of event conditions that can be set in this area is refer to "Table 6-11 Number of Events Settable".
Start: End:	This area is used to set event conditions for starting and stopping a section trace. (Refer to "5.10.5 Setting conditional trace".) Up to four sections can be set at the same time in "Section1" - "Section4". Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions". The number of event conditions that can be set in this area is refer to "Table 6-11 Number of Events Settable".

Table 6-11 Number of Events Settable

Connected IE	Event Conditions Total (execution/access)	Event Link Conditions
[IECUBE]	16 (8/8)	2

### (2) Event manager area

This area displays the list of registered events.

Each event condition can be set easily just by dragging and dropping the event icon displayed in this area onto the event setting area in each event setting dialog box. (Refer to "5.12.3 Setting event conditions".)

This area is common to all event-related dialog boxes. (Refer to "(2) Event manager area" in the " Break dialog box".)

### (3) Function buttons (for event condition contents display, etc.)

These buttons are used to display or delete the event conditions displayed in the event condition setting area, and to display or hide the Event manager area. This area is common to all event-related dialog boxes. (Refer to "(3) Function buttons (for event condition contents display, etc.)" in the " Break dialog box".)

**(4) Function buttons (for registering, deleting, validating, and invalidating event conditions)**

These buttons are used to register, delete, validate, and invalidate the events.

The event with the specified event condition is registered or set (validated) by clicking the [OK] (or [Set]) button. This area is common to all event-related dialog boxes. (Refer to "[\(4\) Function buttons \(for registering, deleting, validating, and invalidating event conditions\)](#)" in the "[Break dialog box](#)".)

## Code Coverage window

[IECUBE]

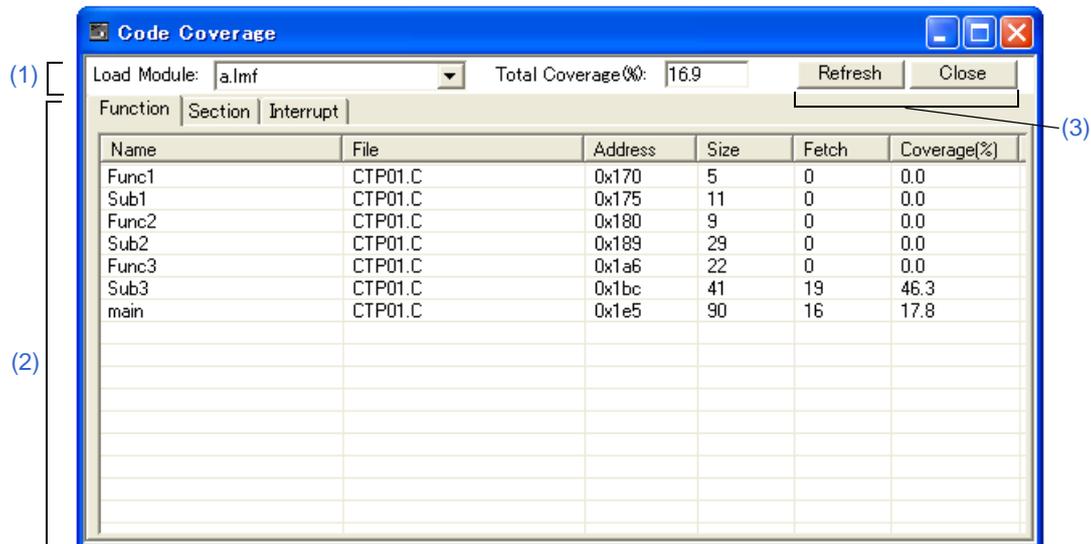
This dialog box displays the code coverage measurement result (C0 coverage). (Refer to "5.11 Coverage Measurement Function [IECUBE].")

The lines where the user program has been executed or not yet executed can be checked in the [Source window](#) or [Assemble window](#).

**Caution:** The coverage measurement result is inaccurate if the on-chip flash memory data is replaced via emulation of flash self programming. (Refer to "Flash Option dialog box".)

**Remark:** Coverage data is cleared when the ID78K0R-QB is started.

Figure 6-51 Code Coverage Window



- Opening
- Explanation of each area
- Context menu

### Opening

Click the  **Cov** button, or select [Browse] menu -> [Code Coverage].

## Explanation of each area

### (1) Coverage information display area

Load Module:	This area is used to select the load module file that has been downloaded. This area is blank when no load module file has been downloaded.
Total Coverage (%):	This area displays the coverage for the area for which code coverage has been measured. <b>Total coverage = Total executed (fetched) function size/total function size</b> (excluding sections outside the coverage measurement range) This area is blank when no load module file has been downloaded.

### (2) Measurement result display area

This area displays the measurement result per tab (function, section, interrupt handler).

The coverage measurement result is updated automatically at a break (it is not updated during user program execution).

This area is blank when no load module file has been downloaded.

The display jumps from this tab to the [Source window](#) or [Assemble window](#) using the start address value of the selected line as a jump pointer. The jump destination window will be displayed from the jump pointer.

The jump function is executed by selecting a jump source line then selecting [Source Text/Assemble] in the [Jump] menu. Jump can also be performed by double-clicking the jump source line.

**Remark:** The displayed items are sorted by clicking the title (on the label) in each column (ascending/descending order is switched each time the title is clicked).

#### (a) When [Function] tab is selected

Name	Function name (displayed as function in segment units in case of assembler source file)
File	Name of file in which the function is defined
Address	Function start address
Size	Function size (unit: bytes)
Fetch	Number of bytes executed (fetched)
Coverage (%)	Coverage of the function (0 - 100%) - - - : When the function is outside the coverage measurement range

#### (b) When [Section] tab is selected

Name	Section name
Type	Section type (code, data)
Address	Section start address
Size	Section size (unit: bytes)
Fetch	Number of bytes executed (fetched)

Coverage [%]	Coverage of the section (0 - 100%) - - - - : When the section is outside the coverage measurement range
--------------	------------------------------------------------------------------------------------------------------------

(c) When [Interrupt] tab is selected

Name	Interrupt request name
Type	Interrupt type (nonmaskable, maskable, software, security id, flash mask option)
Status	Utilization status in the program - - - - : Unknown
Address	Starting address of the interrupt handler
Size	Size of the interrupt handler (unit: bytes) Maximum size for statuses other than "use"
Fetch	Number of bytes executed (fetched)
Coverage [%]	Coverage of the interrupt handler (0 - 100%) - - - - : When the interrupt handler is outside the coverage measurement range

**Remark:** Update data using the [Refresh] button during user program execution.

### (3) Function buttons

Refresh	Updates the contents of this window with the latest watch data.
Close	Closes this window.

### Context menu

Source Text	Displays the corresponding source text and source line, using the data value at the cursor position as the jump destination address. (Refer to "5.17.2 Jump function".) If no line information exists at the jump destination address, however, you cannot jump. Opens the <a href="#">Source window</a> . If an active <a href="#">Source window</a> is open, that window is displayed in the forefront (so that it can be manipulated).
Assemble	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position. (Refer to "5.17.2 Jump function".) Opens the <a href="#">Assemble window</a> . If an active <a href="#">Assemble window</a> is open, that window is displayed in the forefront (so that it can be manipulated).
Clear	Clears the coverage measurement results.

## Delay Count dialog box

[IECUBE]

This dialog box is used to set or display delay count values. (Refer to "5.10 Trace Function [IECUBE]".)

By setting a delay count value, a trace can be executed the number of times specified by the delay count value after the delay trigger event condition set in the [Trace dialog box](#) has been satisfied. (Refer to "5.10.5 Setting conditional trace".)

Figure 6-52 Delay Count Dialog Box



- Opening
- Explanation of each area

### Opening

Select [Event ] menu -> [Delay Count...].

### Explanation of each area

#### (1) Delay count setting area

Delay Count	The following items can be selected.	
	FIRST	Places the trigger pointer at the first of the trace data, traces all frames, and then stops the tracer.
	MIDDLE	Places the trigger pointer at the center of the trace data, traces a half of all frames, and then stops the tracer.
	LAST	Places the trigger pointer at the end of the trace data and immediately stops the tracer.

#### (2) Function buttons

OK	Validates the settings and closes this dialog box.
Restore	Restores the previous settings.
Cancel	Closes this dialog box.
Help	Displays this dialog box online help files.

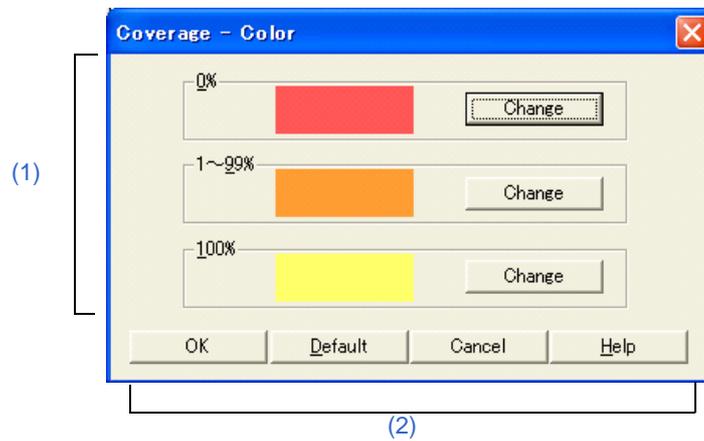


## Coverage-Color dialog box

[IECUBE]

This dialog box is used to select the coverage color to be displayed in the Source window and Assemble window/

Figure 6-53 Coverage-Color dialog box



- [Opening](#)
- [Explanation of each area](#)

### Opening

Click the [Color...] button in the Debugger Option dialog box.

---

## Explanation of each area

---

### (1) Coverage color setting area

Click the [Change] button for 0%, 1-99% or 100% to select the color from the color setting dialog box of a common dialog box.

### (2) Function buttons

OK	Applies the settings to the Source window or Assemble window and closes this dialog box.
Default	Restores the default color for coverage.
Cancel	Closes this dialog box without changing colors.
Help	Displays the help window of this dialog box.

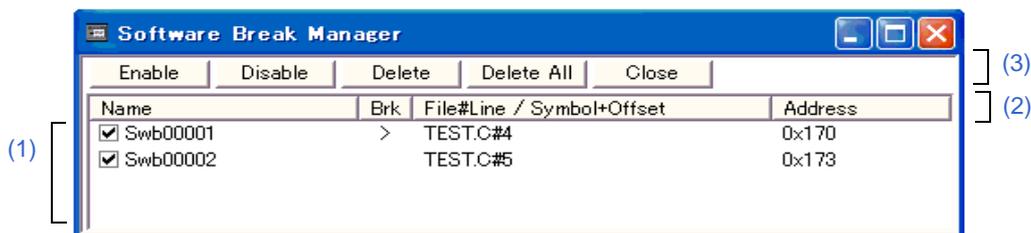
## Software Break manager

This window is used to display, enable or disable, and delete software breaks. (Refer to "5.4.4 Hardware break and software break".)

Software breakpoints cannot be set in this window; they can be set in the [Source window](#) or [Assemble window](#). (Refer to "5.4.2 Breakpoint setting".)

**Remark:** The displayed items are sorted by clicking the title (on the label) in each column (ascending/descending order is switched each time the title is clicked).

Figure 6-54 Software Break Manager



- Opening
- Explanation of each area

### Opening

Select [Event] menu -> [Software Break Manager].

### Explanation of each area

#### (1) Break information display area

Name	<p>This area displays the names of registered events, and the check boxes that indicate whether each event is enabled or disabled.</p> <p>An event name is displayed in the form of "Swb+[number]" in the default condition. It can be changed to an alphanumeric string of up to 256 characters. To change an event name, select and click a name. Then directly edit the name. To set the editing, press the Enter key.</p> <p>When an event is enabled, the check box is selected. To be disable, the check box is cleared.</p> <p>Furthermore, the name jumps to the <a href="#">Source window</a> by double-clicking an event name if the event name corresponds to the source line, whereas the name jumps to the <a href="#">Assemble window</a> if it does not correspond to the source line.</p>
Brk	<p>The "&gt;" mark is displayed for a software break event that is set at the current PC position (so that the software break event that caused a break can be easily identified).</p>

File#Line / Symbol+Offset	This area displays the location at which a software break event was set as follows: <b>Program\$file name#line number</b> (If the event corresponds to the source line.) <b>Program\$file name#symbol+offset</b> (If the event dose not correspond to the source line.) Events are evaluated based on this when a symbol is re-downloaded.
Address	This area displays the address at which a software break event is set.

### (2) Item label area

The displayed items are sorted by clicking each item label (ascending/descending order is switched each time the label is clicked).

Name	Sorts the character strings in alphabetic order (ascending/descending order)
Brk	Does not sort
File#Line / Symbol+Offset	Sorts the character strings in alphabetic order (ascending/descending order)
Address	Sorts the addresses based on higher/lower (ascending/descending order)

### (3) Function buttons

Enable	Enables the selected event.
Disable	Disables the selected event.
Delete	Deletes the selected event.
Delete All	Deletes all the set software break events.
Close	Closes this window.

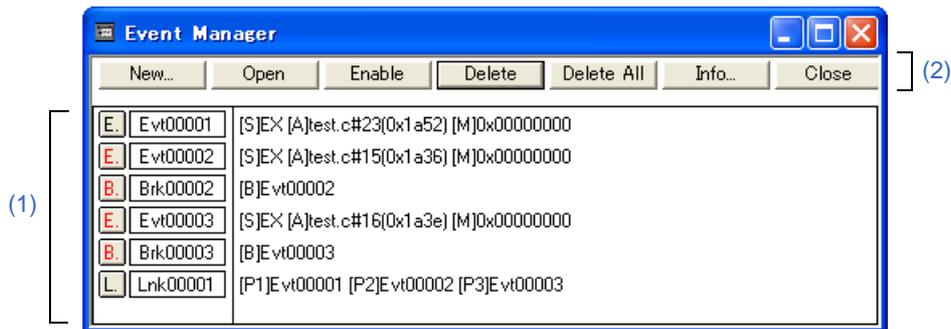
## Event manager

This window is used to manage event conditions. This window allows display, enabling/disabling, and deletion of the [Various Event Conditions](#). (Refer to "5.12 Event Function".)

Other operations using [Context menu](#), [Function button](#), etc., can be performed in this window.

The event icon is the jump pointer of the [Jump function](#).

Figure 6-55 Event Manager (In Detailed Display Mode)



- Opening
- Explanation of each area
- [View] menu (Event manager-dedicated items)
- Context menu

## Opening

Click the  **Mgr** button, or select [Event] menu -> [Event Manager].

## Explanation of each area

### (1) Event display area

This area displays the icons (event icons) of the registered [Various Event Conditions](#).

By selecting the context menu -> [Detail], the details can be displayed.

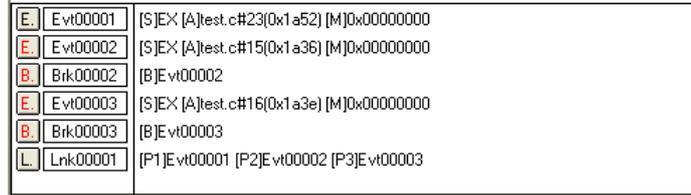
In list displayed	 <p>Displays event icon. (Refer to "<a href="#">Table 5-18 Event Icon</a>".) The event icon is the jump pointer. (Refer to "<a href="#">5.17.2 Jump function</a>".)</p>
In detailed display	 <p>Details of event contents are displayed by using the following key information as a separator. (Refer to "<a href="#">Table 6-12 Separator for Displaying Event Details</a>".)</p>

Table 6-12 Separator for Displaying Event Details

Key Information	Contents
<b>Event condition</b>	
[S]	Status condition
[Z]	Access size condition
[AR]	Address range condition
[A]	Address condition Symbol or expression: (actual address)
[D]	Data condition Symbol or expression: (actual address)
[DR]	Data range condition
[P]	Pass count condition <b>[IECUBE]</b>
[M]	Mask condition
<b>Event link condition [IECUBE]</b>	
[P1] - [P4]	Event link condition on "n" th line
[D]	Disable condition
[P]	Pass count condition
<b>Break condition</b>	
[B]	Break condition

Key Information	Contents
<b>Trace condition [IECUBE]</b>	
[M]	Tracer control mode
[T]	Delay trigger condition
[D]	Delay Count
[S1] - [S4]	Section trace start condition
[E1] - [E4]	Section trace end condition
[Q]	Qualify trace condition
<b>Timer condition [IECUBE]</b>	
[S]	Timer measurement start condition
[E]	Timer measurement end condition
[R]	Timer division ratio
[U]	Timer measurement unit
[B]	Timeout break condition
<b>Snapshot condition [IECUBE]</b>	
[SN]	Snapshot condition
[R]	Register condition
[M]	Memory condition Symbol or expression: (actual address)
[Z]	Access size condition
[F]	SFR condition
<b>Event DMM condition [IECUBE]</b>	
[DM]	Event DMM condition
[F]	SFR condition
[D]	Modified data
[M]	Memory condition
[A]	Data modification address
[Z]	Size of modified data

(2) Function button

OK	Automatically registers the event condition being edited, if any, and closes this dialog box. Each event condition becomes valid as soon as it has been registered.	
New...	Opens the New Event dialog box.	
		<p>Opens the dialog box to create new event condition. By clicking each button, the corresponding event setting dialog box can be opened with the new event name set. After the event setting dialog box has been opened, this dialog box is closed. Returns to Event Manager by clicking the [Cancel] button.</p>
Set	Registers the various event conditions. Because the dialog box is not closed even after an event has been registered, new event conditions can be registered. Each event condition becomes valid as soon as it has been registered.	
Open	Opens the various event setting dialog box corresponding to the selected event condition (one). Each setting dialog box displays the contents of the selected event condition. Same operation as double-clicking the event icon or pressing the Enter key.	
Enable Disable	Validates (enables) or invalidates (disables) the selected event condition. However, event conditions and event link conditions cannot be enabled or disabled. Same operation as the clicking the mark of event icon.	
Delete	Deletes the selected event. When an event condition or an event link condition is to be deleted, an error occurs and the event condition or event link condition cannot be deleted if the event is used as a various event condition.	
Delete All	Deletes all event conditions.	
Info...	Opens the Event Info dialog box. This dialog box is used to change the display mode and rearrange event names.	
		<p>[Sort by Name] ..... Sorts events into name order.                  [Sort by Kind] ..... Sorts events into type order.                  [Unsort] ..... Displays events in the order in which they have been registered without sorting the events.                  [Detail] ..... Sets the detailed display mode.                  [Overview] ..... Sets the list display mode.                  [Cancel] ..... Closes this dialog box (same as ESC key).</p>
Close	Closes this dialog box.	

## [View] menu (Event manager-dedicated items)

The following items are added in the [\[View\] menu](#) , when the Event Manager is active.

Select All Event	Selects all the registered events.
Delete Event	Deletes a selected event.
Sort By Name	Displays icons in the order of event names.
Sort By Kind	Displays icons in the order of event types.
Unsort	Does not sort icons. (default)
Detail	Displays the details.
Overview	Displays the List. (default)

## Context menu

Sort By Name	Displays icons in the order of event names.
Sort By Kind	Displays icons in the order of event types.
Unsort	Does not sort icons. (default)
Detail	Displays the details.
Overview	List display (default)
Source Text	Displays the corresponding source text and source line, using the position of the selected event as the jump destination address. (Refer to " <a href="#">5.17.2 Jump function</a> ".) If no line information exists at the jump destination address, however, you cannot jump. Opens the <a href="#">Source window</a> . If an active <a href="#">Source window</a> is open, that window is displayed in the forefront (so that it can be manipulated).
Assemble	Displays the Assemble window from the position of the selected event, which is used as the jump destination address. (Refer to " <a href="#">5.17.2 Jump function</a> ".) Opens the <a href="#">Assemble window</a> . If an active <a href="#">Assemble window</a> is open, that window is displayed in the forefront (so that it can be manipulated).
Memory	Displays the memory contents from the position of the selected event, which is used as the jump destination address. (Refer to " <a href="#">5.17.2 Jump function</a> ".) Opens the <a href="#">Memory window</a> . If an active <a href="#">Memory window</a> is open, that window is displayed in the forefront (so that it can be manipulated).

## Event Link dialog box

[IECUBE]

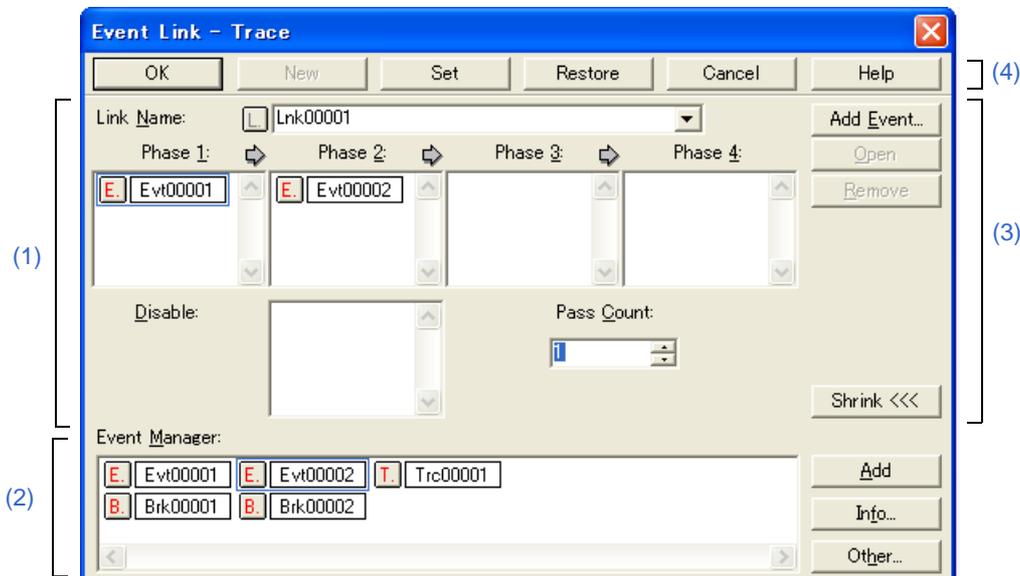
This dialog box is used to register and display event link conditions. (Refer to "5.12 Event Function".)

Registration of event link conditions is done by setting each item (256 items max.) in this dialog box and then clicking the [OK] button. The registered event link conditions are managed by the [Event manager](#).

However, the number of event link conditions that can be simultaneously used is limited ("5.12.4 Number of enabled events for each event condition").

**Caution:** Event conditions that can be specified in the Event Link dialog box are: other than events before execution, and with a pass count value of "0" or "1" (event conditions with a pass count value of 2 or larger cannot be used).

Figure 6-56 Event Link Dialog Box



- [Opening](#)
- [Explanation of each area](#)

## Opening

### In normal mode

If the Event Link dialog box is opened as follows, an event link condition can be registered without its purpose being specified.

Select [Event] menu -> [Event Link...].

### In select mode

If the [OK] button is clicked when the Event Link dialog box has been opened as follows, an event link condition can be registered in the setting dialog box from which this dialog box was opened.

In each various event setting dialog box, click the [Add Link... ] button (the setting dialog box from which the

Event Link dialog box was opened is displayed on the title bar).

## Explanation of each area

### (1) Event link condition setting area

Link Name:	<p>Directly input an alphanumeric string of up to eight characters as a name.</p> <p>To display the contents of an already created event link condition, select from the drop-down list.</p> <p>In the select mode, the selected event condition can be set in the event link condition setting area of the setting dialog box that called the Event Link dialog box.</p> <p>The mark on the left of this area indicates the utilization status of event link conditions ("<a href="#">Table 5-18 Event Icon</a>"). The mark "L" in gray indicates that an event link condition is being edited and has not been registered yet.</p>
Phase 1: Phase 2: Phase 3: Phase 4:	<p>This area is used to specify the sequence in which event conditions and events are detected. Up to four sequences can be specified. If a disable condition is detected while the program is being executed, however, the event conditions that have so far been satisfied are initialized, and the event conditions are detected again starting from the first event condition. If a link condition and a disable condition are detected at the same time, the disable condition takes precedence.</p> <p>Set "Phase 1" -&gt; "Phase 2" -&gt; "Phase 3" -&gt; "Phase 4", in that order. "Phase 4" does not have to be set. In this case, an event occurs when the event condition set for the last phase has been detected. An event condition can be set for only "Phase 1" or the same event condition can be set for two or more Phases.</p> <p>The number of event conditions that can be set to each phase of this area and while the dialog box, refer to "<a href="#">Table 6-13 The Number of Event Conditions in Event Link Dialog Box</a>".</p>
Disable:	<p>This area is used to set an event condition that invalidates the event conditions that have so far been satisfied.</p> <p>Refer to "<a href="#">Table 6-13 The Number of Event Conditions in Event Link Dialog Box</a>" for the number of event conditions that can be set in this area.</p> <p>Setting of event conditions is easily done by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "<a href="#">5.12.3 Setting event conditions</a>".</p>
Pass Count:	<p>This area is used to set a pass count condition (1 - 255).</p> <p>A pass count condition specifies how many times an event condition must be satisfied during user program execution before a given condition is satisfied.</p> <p>If no pass count is specified, 1 is assumed (the condition is satisfied as soon as the event condition is satisfied).</p>

Table 6-13 The Number of Event Conditions in Event Link Dialog Box

Connected IE	Each Phase	Disable Area	Total
[IECUBE]	16	16	16 (8/8)

### (2) Event manager area

This area displays the list of registered events.

Each event condition can be set easily just by dragging and dropping the event icon displayed in this area onto the event setting area in each event setting dialog box. (Refer to "[5.12.3 Setting event conditions](#)".)

This area is common to all event-related dialog boxes. (Refer to "[\(2\) Event manager area](#)" in the "[Break dialog box](#)".)

**(3) Function buttons (for event condition contents display, etc.)**

These buttons are used to display or delete the event conditions displayed in the event condition setting area, and to display or hide the Event manager area. This area is common to all event-related dialog boxes. (Refer to "(3) Function buttons (for event condition contents display, etc.)" in the " Break dialog box".)

**(4) Function buttons (for registering, deleting, validating, and invalidating event conditions)**

These buttons are used to register, delete, validate, and invalidate the events.

The event with the specified event condition is registered by clicking the [OK] (or [Set]) button.

OK	Automatically registers the event condition being edited, if any, and closes this dialog box. <b>In the select mode:</b> An event condition is selected and the setting dialog box (indicated on the title bar) that called the Event Link dialog box is displayed again. If the calling dialog box has already been closed, the select mode is returned to the normal mode, and the Event dialog box is not closed. Otherwise, this dialog box will be closed.
New	Newly creates an event condition in this dialog box. An event condition name is automatically created and a new event condition is prepared.
Set	Registers the various event conditions. Because the dialog box is not closed even after an event has been registered, new event conditions can be registered. <b>In the select mode:</b> An event condition is selected. If there is an event being edited, it is automatically registered and selected.
Enable/Disable	Validates (enables) or invalidates (disables) the selected event condition. However, event conditions and event link conditions cannot be enabled or disabled. Same operation as the clicking the mark of event icon.
Clear	Clears the contents of the event condition.
Restore	Restores the contents of an edited event condition. If an event condition not registered is displayed, all the fields other than the event name field are blank or the default values are set.
Cancel Close	Closes this dialog box. Even if an event condition is being edited, it is not registered and the dialog box is closed.
Help	Displays the help window of this window.

## Break dialog box

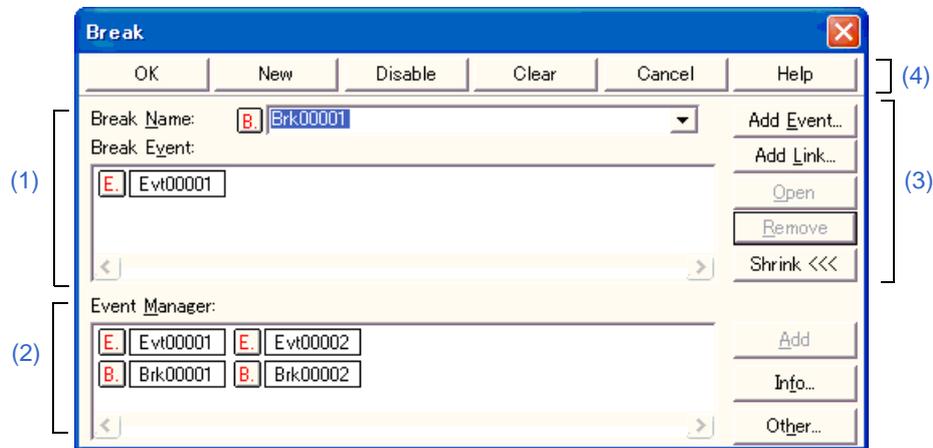
This dialog box is used to register; set, and display break event conditions. (Refer to "5.12 Event Function", "5.4 Break Function".)

Registration and setting of break event conditions is done by setting each item (256 items max.) in this dialog box and then clicking the [OK] button. The registered break event conditions are managed by the [Event manager](#).

There are restrictions on the number of break event conditions that can be simultaneously set (enabled). (Refer to "5.12.4 Number of enabled events for each event condition".)

**Remark:** Break event condition setting/enable/disable/delete operations are possible even during user program execution. **[IECUBE]**

Figure 6-57 Break Dialog Box



- Opening
- Explanation of each area

## Opening

Click the  **Brk** button, or select [Event] menu -> [Break...].

## Explanation of each area

### (1) Break event condition setting area

Break Name:	<p>This area is used to set a break event name. Directly input an alphanumeric string of up to eight characters as a name.</p> <p>To display the contents of an already created event condition, select from the drop-down list.</p> <p>The mark on the left of this area indicates the utilization status of events. (Refer to "Table 5-18 Event Icon".) The gray mark indicates that an event condition is being edited and has not been registered yet. By clicking the left mark, an event condition can be validated or invalidated.</p>
Break Event:	<p>This area is used to set an event condition for break.</p> <p>Refer to "Table 6-14 Number of Events Settable in Condition Setting Area" for the number of event conditions and event link conditions that can be set in this area.</p> <p>Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions".</p>

Table 6-14 Number of Events Settable in Condition Setting Area

Connected IE	Total Event Conditions (Before Execution / After Execution / Access)	Event Link Conditions
[IECUBE]	20 - 24 (4 - 8 <sup>a</sup> /8/8)	2
[MINICUBE2]	1 <sup>b</sup> (0/1/1)	-

- a. The number of settable events before execution varies depending on the address and instruction to be set. [IECUBE]
- b. Either of the event after execution or an access event can be set as an event. [MINICUBE2]

### (2) Event manager area

This area displays the list of registered events.

Each event condition can be set easily just by dragging and dropping the event icon displayed in this area onto the event setting area in each event setting dialog box. (Refer to "5.12.3 Setting event conditions".)

This area is common to all event-related dialog boxes.

[Add]	The event condition and event link condition selected in Event Manager area add to setting area with a focus.
-------	---------------------------------------------------------------------------------------------------------------

[Info...]	<p>Opens the Event Info dialog box. This dialog box is used to change the display mode and rearrange event names.</p> <div data-bbox="435 286 592 560" style="border: 1px solid black; padding: 5px;">  </div> <p>[Sort by Name] ..... Sorts events into name order. [Sort by Kind] ..... Sorts events into type order. [Unsort] ..... Displays events in the order in which they have been registered without sorting the events. [Detail] ..... Sets the detailed display mode. [Overview] ..... Sets the list display mode. [Cancel] ..... Closes this dialog box (same as ESC key).</p>
[Other...]	<p>Opens the Set Other dialog box.</p> <div data-bbox="435 651 592 987" style="border: 1px solid black; padding: 5px;">  </div> <p>By clicking each button, the corresponding event setting dialog box can be opened with the new event name set. After the event setting dialog box has been opened, this dialog box is closed. [Manager...] ..... Opens the Event Manager. [Cancel] ..... Closes the dialog box to create event condition.</p>

### (3) Function buttons (for event condition contents display, etc.)

These buttons are used to display or delete the event conditions displayed in the event condition setting area, and to display or hide the Event manager area. This area is common to all event-related dialog boxes.

Add Event...	<p>Opens the <a href="#">Event dialog box</a> in the select mode, and selects or newly creates an event condition to be set. The event condition will be added to the area selected when the [ Add Event...] button is clicked.</p>
Add Link...	<p>Opens the <a href="#">Event Link dialog box</a> in the select mode, and selects or newly creates an event link condition. The event condition will be added to the area selected when the [ Add Link...] button is clicked.</p>
Open	<p>Opens the various event setting dialog box corresponding to the selected event condition (one). Each setting dialog box displays the contents of the selected event condition. Same operation as double-clicking the event icon or clicking the Enter key.</p>
Remove Delete	<p>Deletes the selected event. When an event condition or an event link condition is to be deleted, an error occurs and the event condition or event link condition cannot be deleted if the event is used as a various event condition.</p>
Expand >>> Shrink <<<	<p>Turns on or off display of the event manager area. The size of the dialog box is expanded or reduced.</p>

**(4) Function buttons (for registering, deleting, validating, and invalidating event conditions)**

These buttons are used to register, delete, validate, and invalidate the events.

The event with the specified event condition is registered or set (validated) by clicking the [OK] (or [Set]) button. This area is common to all event-related dialog boxes.

OK	Automatically registers the event condition or the various event condition being edited, if any, and closes this dialog box. Each event condition becomes valid as soon as it has been registered.
New	Newly creates an event condition in this dialog box. An event condition name is automatically created and a new event condition is prepared.
Set	Registers the event condition or the various event condition. Because the dialog box is not closed even after an event has been registered, new event conditions can be registered. Each event condition becomes valid as soon as it has been registered.
Enable/Disable	Validates (enables) or invalidates (disables) the selected event condition. However, event conditions and event link conditions cannot be enabled or disabled. Same operation as the clicking the mark of event icon.
Clear	Clears the contents of the event condition.
Restore	Restores the contents of an edited event condition. If an event condition not registered is displayed, all the fields other than the event name field are blank or the default values are set.
Cancel Close	Closes this dialog box. Even if an event condition is being edited, it is not registered and the dialog box is closed.
Help	Displays the help window of this window.

## Snap Shot dialog box

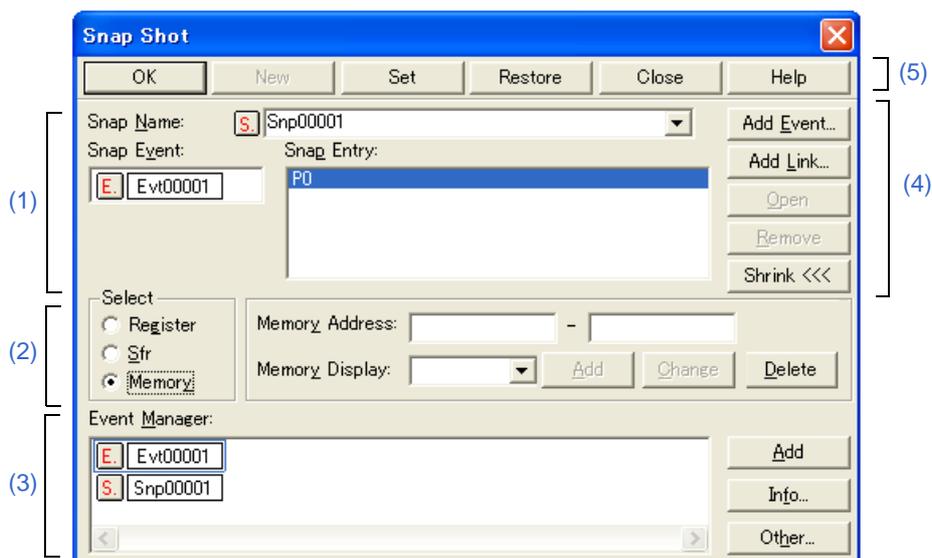
[IECUBE]

This dialog box is used to register; set, and display snapshot event conditions. (Refer to "5.12 Event Function", "5.13 Snapshot Function [IECUBE]").)

Registration and setting of snapshot event conditions is done by setting each item (256 items max.) in this dialog box and then clicking the [OK] button. The registered snapshot event conditions are managed by the [Event manager](#).

There are restrictions on the number of snapshot event conditions that can be simultaneously set (enabled). (Refer to "5.12.4 Number of enabled events for each event condition".)

Figure 6-58 Snap Shot Dialog Box (When "Memory" Is Selected)



- Opening
- Explanation of each area

### Opening

Select [Event] menu -> [Snap Shot...].

## Explanation of each area

### (1) Snapshot event condition setting area

Snap Name:	<p>This area is used to set a snapshot event name. Directly input an alphanumeric string of up to eight characters as a name.</p> <p>To display the contents of an already created event condition, select from the drop-down list.</p> <p>The mark on the left of this area indicates the utilization status of events. (Refer to "<a href="#">Table 5-18 Event Icon</a>".) The gray mark indicates that an event condition is being edited and has not been registered yet. By clicking the left mark, an event condition can be validated or invalidated.</p>	
Snap Event:	<p>This area is used to set an event condition for snapshot.</p> <p>The number of event condition and event link condition that can be set in this area, refer to "<a href="#">Table 6-11 Number of Events Settable</a>". Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "<a href="#">5.12.3 Setting event conditions</a>".</p>	
Snap Entry:	<p>This area displays a list of registered snap data.</p> <p>The registered snap data is written into the tracer when a snapshot event occurs.</p> <p>If snap data is selected in this area, the contents of the selected snap data are displayed in each setting area.</p> <p>Snap data is displayed as follows:</p>	
	Register	<p>Register name [Bank number or 'Current']</p> <p>Example: RP0[0] RP[Current] All[2]</p>
	SFR	<p>SFR name</p> <p>Example: PM0 PM1</p>
	Memory	<p>Start address &lt;symbol expression&gt; - End address &lt;symbol expression&gt;, Access size</p> <p>Example: 0xFE20&lt;byVar&gt;,B0xFE22&lt;wVar&gt; ,W0xFE30&lt;szVar&gt; - 0xFE2F&lt;szVar+0x10&gt;,B</p>

**(2) Snap data select area**

Select	<p>This area is used to register, change, and delete the snap data, selecting type of data subject to snapshot.</p> <p>As snap data, up to 16 registers, SFR, and memory addresses each can be registered; therefore, a total of 48 registers, SFR, and memory addresses can be registered.</p> <p>By selecting a data type in "Select" area, the item to be displayed in the snap data setting area on the right of this area changes in accordance with the selected area.</p> <p>Perform operations such as registering, changing or deleting of snap data using the Buttons to <a href="#">Buttons to manipulate snap data</a>.</p> <p>The snap data registered, changed, or deleted in this area is reflected in "Snap Entry:". If snap data is selected in "Snap Entry:", the contents of the selected snap data are displayed in this area.</p>	
	Register	This should be selected to set by a register as snap data.
	Sfr	This should be selected to set by a SFR as snap data.
	Memory	This should be selected to set by a Memory as snap data.

**(a) When "Register" is selected**

Register Name:	<p>To specify a register name, either directly input one to the text box, or select one from the drop-down list. The case is distinguished.</p> <p>A register name can be specified as both a function name and an absolute name.</p>
Register Bank:	<p>To specify a register bank, either directly input one to the text box, or select one from the drop-down list (0 -3). To specify the current bank, select 'Current'. However, the current bank will be specified even when the specification is omitted.</p>

**(b) When "Sfr" is selected**

Sfr Name:	<p>To specify SFR name, either directly input one to the text box, or select one from the drop-down list.</p> <p>Only the SFR that can be read can be specified. Uppercase and lowercase characters are not distinguished. The SFR bit names and the I/O port name registered in the <a href="#">Add I/O Port dialog box</a> cannot be specified.</p> <p>Note that all the register names are displayed in uppercase characters in the <a href="#">Trace View window</a>.</p>
-----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(c) When "Memory" is selected

Memory Address:	<p><b>Start address - End address</b></p> <p>This area is used to specify an address range of the memory. If a value is input as only the start address and specifying the end address is omitted, it is assumed that the same value as the start address is specified as the end address. If the specified address range cannot be divided by the access size, the address range is rounded up to a range that can be divided by the access size.</p> <p>An address can be also specified by a symbol or expression. (Refer to "<a href="#">Table 5-5 Specifying Symbols</a>".) The default radix for inputting a numeric value is hexadecimal.</p> <p>If addresses are registered or changed in this area by using expressions or symbols, the converted address values are displayed along with the specified expressions and symbols.</p> <p>In the <a href="#">Trace View window</a>, only the converted address values are displayed.</p> <p>Address settable range: 0 &lt;= Address value &lt;= 0xFFFFF</p>
Memory Display:	To specify the access size, either directly input the size or select the size from the drop-down list. The case is distinguished.

**Remark1** Select "All" to specify all registers.

**Remark2:** The names of general-purpose registers and control registers can be specified as Register Name.

**Remark3:** The snap data displayed in the [Trace View window](#) is unified in uppercase letters and in the format of 'absolute name'.

(d) Buttons to manipulate snap data

The following buttons are used to register, change, and delete snap data.

Add	Registers as snap data. Newly registered snap data is added to the end of the "Snap Entry:" area.
Change	Changes the contents of the snap data selected in "Snap Entry:" area to the contents of the snap data specified in this area.
Delete	Deletes the snap data selected in "Snap Entry:". The DEL key performs the same operation.

### (3) Event manager area

This area displays the list of registered events.

Each event condition can be set easily just by dragging and dropping the event icon displayed in this area onto the event setting area in each event setting dialog box. (Refer to "[5.12.3 Setting event conditions](#)".)

This area is common to all event-related dialog boxes. (Refer to "[\(2\) Event manager area](#)" in the "[Break dialog box](#)".)

**(4) Function buttons (for event condition contents display, etc.)**

These buttons are used to display or delete the event conditions displayed in the event condition setting area, and to display or hide the Event manager area. This area is common to all event-related dialog boxes. (Refer to "[\(3\) Function buttons \(for event condition contents display, etc.\)](#)" in the " [Break dialog box](#)".)

**(5) Function buttons (for registering, deleting, validating, and invalidating event conditions)**

These buttons are used to register, delete, validate, and invalidate the events.

The event with the specified event condition is registered or set (validated) by clicking the [OK] (or [Set]) button. This area is common to all event-related dialog boxes. (Refer to "[\(4\) Function buttons \(for registering, deleting, validating, and invalidating event conditions\)](#)" in the " [Break dialog box](#)".)

## Event DMM dialog box

[IECUBE]

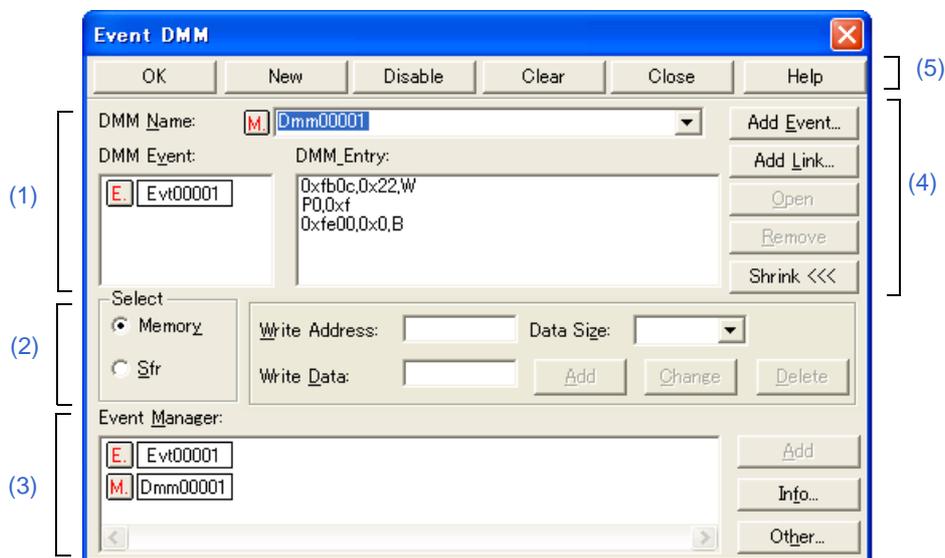
This dialog box is used to register; set, and display event DMM conditions. (Refer to "5.12 Event Function", "5.15.1 Event DMM condition [IECUBE]".)

Registration and setting of event DMM conditions is done by setting each item (256 items max.) in this dialog box and then clicking the [OK] button. The registered event DMM conditions are managed by the [Event manager](#).

There are restrictions on the number of event DMM conditions that can be simultaneously set (enabled). (Refer to "5.12.4 Number of enabled events for each event condition".)

**Remark:** Up to 100 SFR data items and memory data items can be registered per event DMM.

Figure 6-59 Event DMM Dialog Box (When "Memory" Is Selected)



- Opening
- Explanation of each area

## Opening

Select [Event] menu -> [Event DMM...].

## Explanation of each area

### (1) DMM event condition setting area

DMM Name:	<p>This area is used to set a event DMM name. Directly input an alphanumeric string of up to eight characters as a name.</p> <p>To display the contents of an already created event condition, select from the drop-down list.</p> <p>The mark on the left of this area indicates the utilization status of events. (Refer to "<a href="#">Table 5-18 Event Icon</a>".) The gray mark indicates that an event condition is being edited and has not been registered yet. By clicking the left mark, an event condition can be validated or invalidated.</p>	
DMM Event:	<p>This area is used to set an event condition for event DMM.</p> <p>The number of event condition and event link condition that can be set in this area, refer to "<a href="#">Table 6-11 Number of Events Settable</a>". Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "<a href="#">5.12.3 Setting event conditions</a>".</p>	
DMM Entry:	<p>This area displays a list of registered event DMM data.</p> <p>If event DMM data is selected in this area, the contents of the selected event DMM data are displayed in each setting area.</p> <p>Event DMM data is displayed as follows:</p>	
	Memory data:	<p>Start address &lt;symbol expression&gt; ,Write data, Access size (If a symbol or expression is used for address specification, it is enclosed with "&lt; &gt;".)</p>
		<p>Example: 0xFE20&lt;byVar&gt;, 0xff, B 0xFE22&lt;wVar&gt;, 0xff00ff00, W</p>
	SFR data:	<p>SFR name, Write data</p>
		<p>Example: P0, 0xff TMS, 0x0000</p>

### (2) Event DMM Data selection area

Select	<p>This area is used to register, change, or delete the target data for DMM (event DMM data) after establishment of an event.</p> <p>The memory or SFR can be registered as event DMM data, and up to 100 event DMM data items can be registered per event DMM.</p> <p>By selecting a data type in "Select" area, the item to be displayed in the snap data setting area on the right of this area changes in accordance with the selected area.</p> <p>The event DMM data registered, changed, or deleted in this area is reflected in "DMM Entry:". If event DMM data is selected in "DMM Entry:", the contents of the selected event DMM data are displayed in this area.</p>	
	Memory	This should be selected to set by a memory as DMM data.
	Sfr	This should be selected to set by a SFR as DMM data.

## (a) When "Memory" is selected

This area is used to specify the start address, data value to be written, and data size when writing data to the memory after establishment of an event.

Write Address:	Specifies the start address An address can be also specified by a symbol or expression. (Refer to " <a href="#">Table 5-5 Specifying Symbols</a> ".)
Write Data:	Specifies the data value
Data Size:	Specifies the data size (Byte, Word)

## (b) When "Sfr" is selected

This area is used to specify the SFR name and write data value for writing data to the SFR after event establishment.

Sfr Name:	Specifies the SFR name
Write Data:	Specifies the data value

## (c) Buttons to manipulate event DMM data

The following buttons are used to register, change, and delete event DMM data.

Add	Registers as event DMM data. Newly registered DMM data is added to the end of the "DMM Entry:" area.
Change	Changes the contents of the event DMM data selected in "DMM Entry:" area to the contents of the event DMM data specified in this area.
Delete	Deletes the event DMM data selected in "DMM Entry:". The DEL key performs the same operation.

**(3) Event manager area**

This area displays the list of registered events.

Each event condition can be set easily just by dragging and dropping the event icon displayed in this area onto the event setting area in each event setting dialog box. (Refer to "[5.12.3 Setting event conditions](#)".)

This area is common to all event-related dialog boxes. (Refer to "[\(2\) Event manager area](#)" in the "[Break dialog box](#)".)

**(4) Function buttons (for event condition contents display, etc.)**

These buttons are used to display or delete the event conditions displayed in the event condition setting area, and to display or hide the Event manager area. This area is common to all event-related dialog boxes. (Refer to "[\(3\) Function buttons \(for event condition contents display, etc.\)](#)" in the "[Break dialog box](#)".)

**(5) Function buttons (for registering, deleting, validating, and invalidating event conditions)**

These buttons are used to register, delete, validate, and invalidate the events.

The event with the specified event condition is registered or set (validated) by clicking the [OK] (or [Set]) button. This area is common to all event-related dialog boxes. (Refer to "[\(4\) Function buttons \(for registering, deleting, validating, and invalidating event conditions\)](#)" in the "[Break dialog box](#)".)

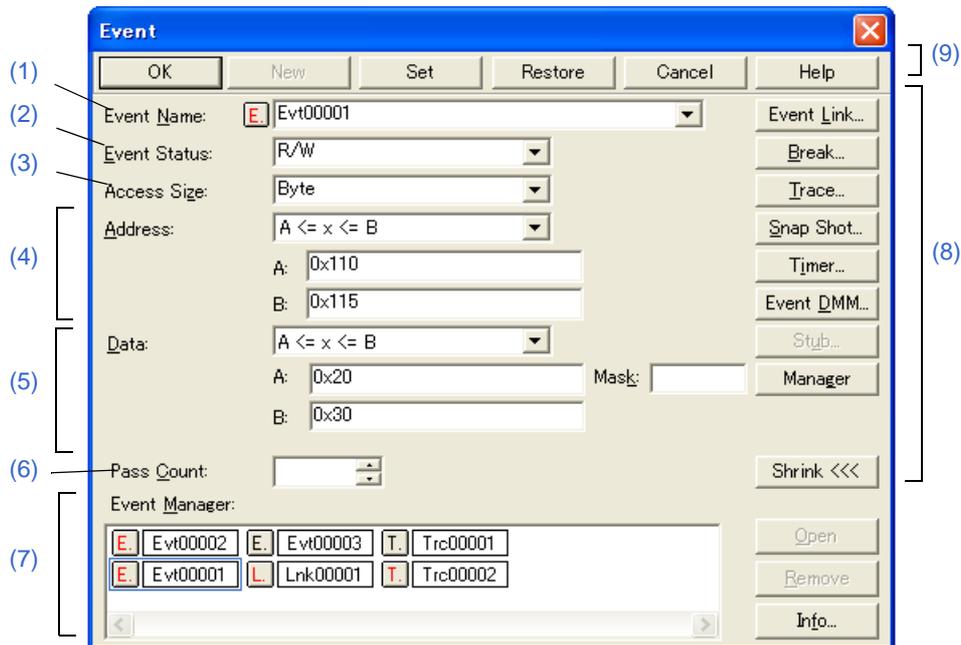
## Event dialog box

This dialog box is used to register and display event conditions. (Refer to "5.12 Event Function".)

Setting of event conditions is done by setting each item in this dialog box and then clicking the [OK] button. The registered event conditions are managed by the [Event manager](#).

One event condition can be set for multiple [Various Event Conditions](#). However, the number of event conditions that can be simultaneously used is limited. (Refer to "5.12.4 Number of enabled events for each event condition".)

Figure 6-60 Event Dialog Box



- Opening
- Explanation of each area

## Opening

### In normal mode

If the Event dialog box is opened as follows, an event condition can be registered without its purpose being specified.

Click the **Evn** button, or select [Event] menu -> [Event...].

### In select mode

If the [OK] button is clicked when the Event dialog box has been opened as follows, an event condition can be registered in the setting dialog box from which this dialog box was opened (the setting dialog box from which the this box was opened is displayed on the title bar).

In each various event setting dialog box, click the [Add Event... ] button.

## Explanation of each area

### (1) Event Name:

This area is used to set an event name.

Directly input an alphanumeric string of up to eight characters as a name.

To display the contents of an already created event condition, select from the drop-down list.

In the select mode, the selected event condition can be set in the event condition setting area of the setting dialog box that called the Event dialog box.

The mark on the left of this area indicates the utilization status of events. (Refer to "Table 5-18 Event Icon".)

The gray E. mark indicates that the event condition is being edited and has not been registered yet.

### (2) Event Status:

The status conditions that can be specified are listed below.

This area is used to select a status condition.

By specifying a status condition, the type of the execution event and an access event is determined (if an execution event is specified, nothing can be input to the "Access Size:" and "Data:, Mask:").

Execution event		
Execution	EX	Program execution
Before Execution	EX-B	Program execution (break before execution) <sup>Note</sup> [IECUBE]
Access event		
R/W	RW	Memory read/write
Read	R	Memory read
Write	W	Memory write

**Note:** "Before Execution" can only be set as a break event condition.

4 Before Execution items can be set.

**Caution:** Breaks before execution can be set internal ROM only. When setting them during execution, they cannot be set if flash self-programming has been executed.

### (3) Access Size:

This area is used to select an access size condition.

By selecting an access size condition from the drop-down list, the access width of a data condition to be detected by an access event is determined. If no access size condition is specified, a judgment is automatically made from the address condition and data condition.

Byte	Detects data condition with 8-bit width (only during 8-bit access). <b>Caution:</b> If this item is selected, accesses in word units are not detected. [IECUBE]
Word	Detects data condition with 16-bit width (only during 16-bit access).
No Condition	Does not detect access size (nothing can be input to the Data area).

Bit	Detects data condition with 1-bit width (only during 8-bit access) <sup>Note</sup> .
-----	--------------------------------------------------------------------------------------

**Note:** In this case, a search is made for a data condition with 1-bit width. Because of the operation of the simulator, access to a bit is not directly detected; the simulator searches a dummy bit access by internally setting address conditions and data conditions as follows:

<b>Input example:</b>		<b>Setting of trace search:</b>
Address: FE20.1	---->	Address: FE20
Data: 1		Data: 00000010B
		Mask: 11111101B

If another bit of the same address is accessed or if all the 8 bits of the same address are accessed, therefore, an event is detected in accordance with the specified status if the address and bit match the specified value of [address.bit].

#### (4) Address condition setting area

This area is used to specify an address condition (may be omitted).

The following can be set:

Table 6-15 Settable Range of Address Condition (Event)

Connected IE	Settable Range
[IECUBE]	0 <= address value <= 0xFFFFFFFF
	0 <= mask value <= 0xFFFFFFFF

(a) Address:

Select the address range (specification method) from a drop-down list and specify the address value in areas **A:** and **B:**.

The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to "Table 5-5 Specifying Symbols".)

x == A	Specify address value specified for <b>A:</b> .
x >= A [IECUBE]	Specify address value higher than address value specified for <b>A:</b> .
x <= B [IECUBE]	Specify address value lower than address value specified for <b>B:</b> .
A <= x <= B [IECUBE]	Specify address value within address range from <b>A:</b> to <b>B:</b> . In this case, 2 events are used.
x < A    B < x [IECUBE]	Specify value outside address range from <b>A:</b> to <b>B:</b> .

**(5) Data condition setting area**

This area is used to specify an data condition (data value, mask value).

The default radix for inputting a numeric value is hexadecimal.

The settable range differs as follows depending on the access size condition specified in "Access Size":

Access Size	Settable Range
Byte	0 <= data value <= 0xFF 0 <= mask value <= 0xFF
Word	0 <= data value <= 0xFFFF 0 <= mask value <= 0xFFFF
Bit	Data value = 0 or 1 Mask value = Cannot be specified.

**(a) Data:**

Select the data range (specification method) from a drop-down list and specify the data value in areas **A:** and **B:**.

The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to "Table 5-5 Specifying Symbols".)

No Condition [IECUBE]	Don't care (specify all data values).
x != A [IECUBE]	Specify data value not matching data value specified for <b>A:</b> .
x == A	Specify data value matching data value specified for <b>A:</b> .
x >= A [IECUBE]	Specify data value higher than data value specified for <b>A:</b> .
x <= B [IECUBE]	Specify data value lower than data value specified for <b>B:</b> .
A <= x <= B [IECUBE]	Specify data value within data range from <b>A:</b> to <b>B:</b> . In this case, 2 events are used.
x < A    B < x [IECUBE]	Specify data value outside data range from <b>A:</b> to <b>B:</b> .

**(b) Mask:**

Set a mask value for the data value (may be omitted).

When a mask is set, the data value for the bit whose mask value is 1 may be 0 or 1.

**Example1: 0x4000 to 0x40FF satisfy the condition.**

Data	0x4000
Mask	0xFF

**Example2: 0x4000, 0x4001, 0x4100, and 0x4101 satisfy the condition.**

Data	0x4000
Mask	0x101

**(6) Pass Count: [IECUBE]**

A pass count specifies how many times an event condition must be satisfied during user program execution before a given condition is satisfied (1 - 255).

If no pass count is specified, 1 is assumed (the condition is satisfied as soon as the event condition is satisfied).

**(7) Event manager area**

Event manager:	This area is used to display the list of the events registered. (Refer to "Table 5-18 Event Icon", "(4) Manipulation in event manager area".)	
[Open]	Opens the various event setting dialog box corresponding to the selected event condition (one). Each setting dialog box displays the contents of the selected event condition. Same operation as double-clicking the event icon or pressing the Enter key.	
[Remove]	Deletes the selected event. When an event condition or an event link condition is to be deleted, an error occurs and the event condition or event link condition cannot be deleted if the event is used as a various event condition.	
[Info...]	Opens the Event Info dialog box. This dialog box is used to change the display mode and rearrange event names.	
		<p>[Sort by Name] ..... Sorts events into name order.                  [Sort by Kind] ..... Sorts events into type order.                  [Unsort] ..... Displays events in the order in which they have been registered without sorting the events.                  [Detail] ..... Sets the detailed display mode.                  [Overview] ..... Sets the list display mode.                  [Cancel] ..... Closes this dialog box (same as ESC key).</p>

**(8) Function buttons (for event each event condition setting)**

Event Link... Break... Trace... Snap Shot... Timer... Event DMM...	By clicking each button, the corresponding event setting dialog box can be opened with the new event name set.
Manager...	Opens the <a href="#">Event manager</a> .
Expand >>> Shrink <<<	Turns on or off display of the event manager area. The size of the dialog box is expanded or reduced.

**(9) Function buttons (for registering, deleting, validating, and invalidating event conditions)**

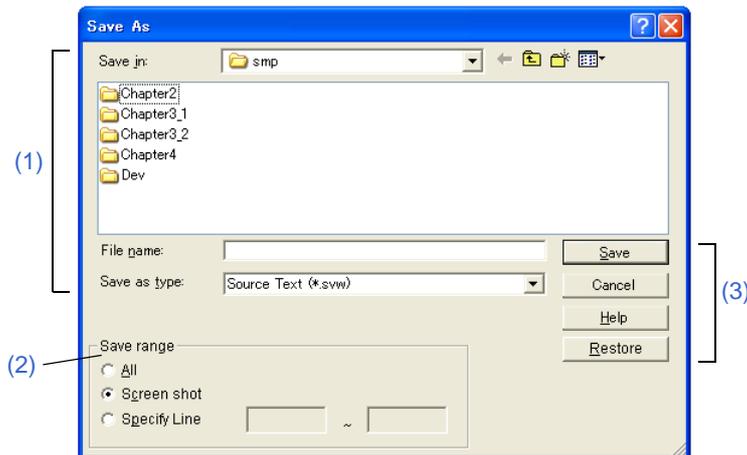
These buttons are used to register, delete, validate, and invalidate the events.

The event with the specified event condition is registered or set by clicking the [OK] (or [Set]) button. This area is common to all event-related dialog boxes. (Refer to "[\(4\) Function buttons \(for registering, deleting, validating, and invalidating event conditions\)](#)" in the "[Break dialog box](#)".)

## View File Save dialog box

This dialog box is used to save the current display information of the current window to a view file. (Refer to "5.16.2 Window display information (view file)".)

Figure 6-61 View File Save Dialog Box



- Opening
- Explanation of each area

### Opening

When the window to be saved is the current window, select [File] menu -> [Save As...].

### Explanation of each area

#### (1) Save file setting area

Save in:	This area is used to specify a file name. A file name can be directly input, or selected from the list at the upper part of this area.
File name:	
Save as type:	This area is used to specify the type (extension) of the file to be saved. (Refer to "Table 5-21 Type of View Files".) The extension of the file corresponding to the current window is displayed.

**(2) Save range setting area**

If a range of 100 lines / 100 frames / 256 bytes or more is specified, a message dialog box is displayed to indicate the progress of saving. To stop saving midway, click the [Stop] button in the message dialog box.

Save range	Specify the range of data to be saved. This area is displayed if the current window to be saved is the following. - <a href="#">Source window</a> - <a href="#">Assemble window</a> - <a href="#">Memory window</a> - <a href="#">Trace View window</a>	
	All	This should be selected to save the entire range, from the first line to the last line.
	Screen shot	This should be selected to save the area visible on the screen, from the top line on the screen to the bottom line. If the <a href="#">Source window</a> is in the mixed display mode, however, the window contents are saved from the source line that includes the area visible on the screen.
	Specify Line Specify Frame Specify Address	This should be selected to specify the start line and end line of the area to be saved. If the start line and end line are omitted, the first line and last line are assumed. Display any of the following corresponding to the current window:
	Specify Line	Specify the range of the line numbers to be saved. The default radix for inputting a numeric value is decimal. If the <a href="#">Source window</a> is in the mixed display mode, the mixed displayed part on the specified line is also saved.
	Specify Frame	Specify the range of trace frames to be saved. (Refer to " <a href="#">Table 6-10 Frame Number Specification Format</a> ".) The default radix for inputting a numeric value is decimal.
	Specify Address	Specify the range of address to be saved. An address can be also specified by a symbol or expression. (Refer to " <a href="#">Table 5-5 Specifying Symbols</a> ".) The default radix for inputting a numeric value is hexadecimal.

**(3) Function buttons**

Save	Saves the display information of the current window to the selected file. After saving, this dialog box is closed.
Cancel	Closes this dialog box without executing anything.
Help	Displays this dialog box online help files.
Restore	Restores the status before this dialog box was opened.

## View File Load dialog box

This dialog box is used to read the view files. (Refer to "5.16.2 Window display information (view file)".)

When a view file is loaded, the reference window ([Source window](#) in static status) opens and the display information at saving is displayed.

The window to be opened and its status differ as follows, depending on the file to be loaded.

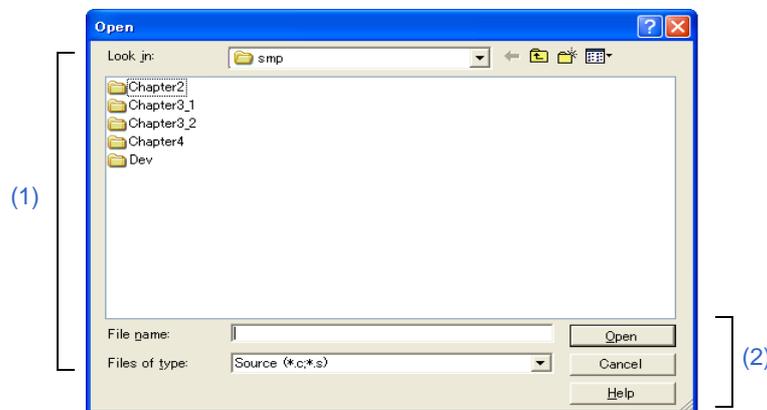
### - Loading source file to which symbol information has been read

If there is a [Source window](#) in the active status, it is opened in the static status; otherwise, the [Source window](#) is opened in the active status.

### - Loading source file to which symbol information has not been read, or view file

A window of text-format files is opened in the [Source window](#) in the static status.

Figure 6-62 View File Load Dialog Box



- Opening

- Explanation of each area

## Opening

Click the  **Open** button or select [File] menu -> [Open...].

## Explanation of each area

### (1) Load file setting area

Look in:	This area is used to specify the file name to be loaded. A file name can be directly input from the keyboard, or selected from the list.
File name:	A string of up to 259 characters with a path name and an extension can be specified.
Files of type:	This area is used to specify the type (extension) of the file to be loaded. (Refer to " <a href="#">Table 5-21 Type of View Files</a> ".)

**(2) Function buttons**

Open	Loads the selected file. After loading the file, this dialog box is closed.
Cancel	Closes this dialog box without executing anything.
Help	Displays this dialog box online help files.

## Environment Setting File Save dialog box

This dialog box is used to save the setting contents of the current window to a setting file. (Refer to "5.16.3 Window setting information (setting file)".)

Figure 6-63 Environment Setting File Save Dialog Box



- Opening
- Explanation of each area

### Opening

When the window to be saved is the current window, select [File] menu -> [Environment] -> [Save As...].

### Explanation of each area

#### (1) Save file setting area

Save in:	This area is used to specify a file name. A file name can be directly input, or selected from the list at the upper part of this area.
File name:	A string of up to 259 characters with a path name and an extension can be specified.
Save as type:	This area is used to specify the type (extension) of the file to be saved. (Refer to "Table 5-22 Type of Setting Files".) The extension of the file corresponding to the current window is displayed.

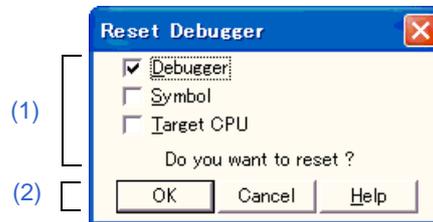
#### (2) Function buttons

Save	Saves the setting information of the current window to the selected file. After saving, this dialog box is closed.
Cancel	Closes this dialog box without executing anything.
Help	Displays this dialog box online help files.

## Reset Debugger dialog box

This dialog box is used to initialize the ID78K0R-QB, CPU, and symbol information.

Figure 6-64 Reset Debugger Dialog Box



- [Opening](#)
- [Explanation of each area](#)

### Opening

Select [File] menu -> [Debugger Reset...].

### Explanation of each area

#### (1) Reset subject selection area

This area is used to specify what is to be Initialized. Initializes the selected item.

Debugger	Initializes the ID78K0R-QB. (default)
Symbol	Initializes the symbol information.
Target CPU	Initializes the CPU.

#### (2) Function buttons

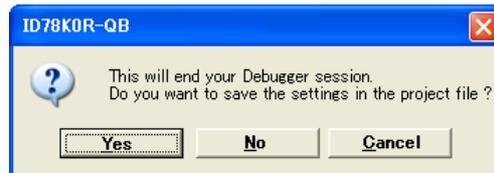
OK	Initializes according to the setting.
Cancel	Cancels the changes and closes this dialog box.
Help	Displays this dialog box online help files.

## Exit Debugger dialog box

This dialog box is used to select whether the current debug environment is saved to a project file or not before terminating the ID78K0R-QB. (Refer to "[5.16.1 Debugging environment \(project file\)](#)".)

It can be specified in the [Debugger Option dialog box](#) that the ID78K0R-QB is terminated without this confirmation dialog box being opened.

Figure 6-65 Exit Debugger Dialog Box



- [Opening](#)
- [Function buttons](#)

### Opening

- Select [File] menu -> [Exit].
- If forcible termination, such as to terminate the application, has been executed on the task list that terminates windows.

### Function buttons

Yes	Saves the current debug environment to a project file, closes all the windows, and terminates the ID78K0R-QB. If a project file name is not specified, the <a href="#">Project File Save dialog box</a> is opened. If the [Cancel] button is selected on the <a href="#">Project File Save dialog box</a> , the environment is neither saved to a project file nor is the ID78K0R-QB terminated. (If a project file is loaded or saved during debugger operation, this button has the default focus.)
No	Closes all the windows and terminates the ID78K0R-QB. (If a project file is not loaded or saved during debugger operation, this button has the default focus.)
Cancel	Closes this dialog box without executing anything.

## About dialog box

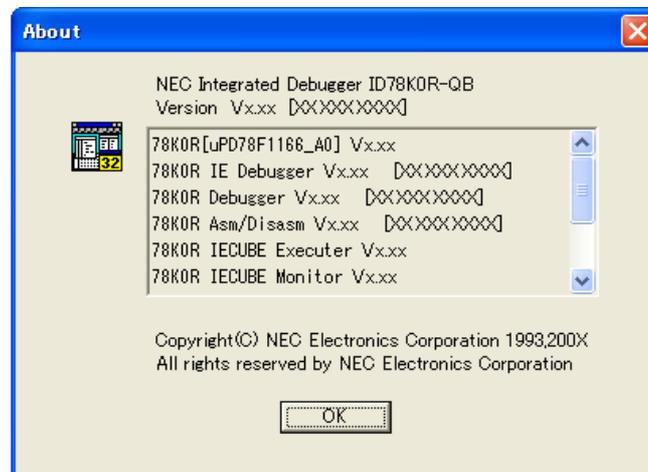
This dialog box displays the version information of the ID78K0R-QB (the year is displayed in 4 digits).

**Remark:** The version information can be copied to the clipboard by selecting [Select All and Copy (&C)] from the context menu in the dialog box.

The following version information is displayed:

- Product version of ID78K0R-QB
- Version of device file
- Version of GUI
- Version of debugger DLL
- Version of assembler DLL
- Version of executor
- Version of Tcl/Tk
- Product ID and product version of in-circuit emulator
- Version of Flash Firmware [**IECUBE**]
- Version of OCD Control Code [**MINICUBE2**]

Figure 6-66 About Dialog Box



- [Opening](#)

## Opening

Select [Help] menu -> [About...].

**Remark:** This dialog also be opened by clicking the [About...] button in the [Configuration dialog box](#).

---

## Console window

---

This window is used to input commands that control the ID78K0R-QB.

Because the key bind is Emacs-like, the accelerator key is not acknowledged if the Console window is active. However, the F1 key displays the online help files of the Console window.

While the Console window is open, an error message window with only an [OK] button is displayed in the Console window.

The command history is saved in file tkcon.hst when the Console window is closed, or the debugger is terminated with the Console window open.

Up to 100 command history items can be saved.

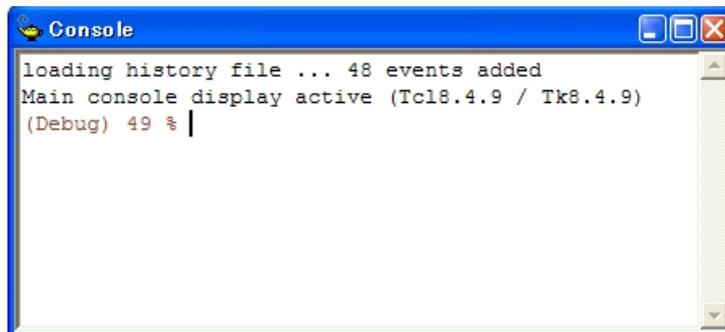
The command history is loaded the next time the Console window is opened.

Where to store the tkcon.hst file is determined by the following priority order.

- 1) Based on environment variable "HOME"
- 2) Based on environment variable "USERPROFILE"
- 3) Directly under the drive in which the OS is installed

Refer to "[CHAPTER 7 COMMAND REFERENCE](#)" for details on the command specifications.

Figure 6-67 Console Window



- [Opening](#)

---

## Opening

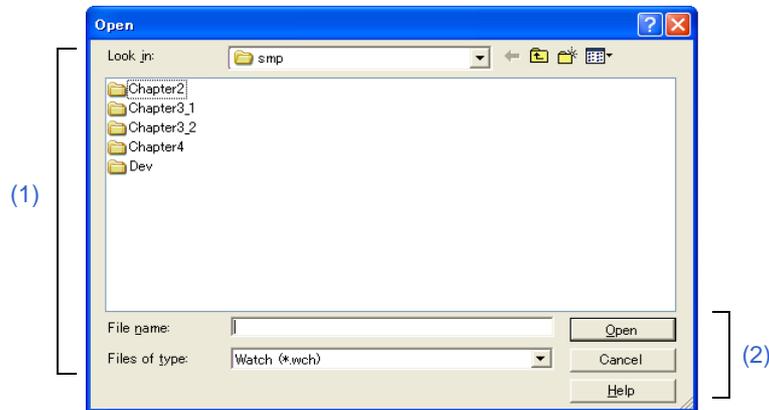
---

Select [Browse] menu -> [Console].

## Environment Setting File Load dialog box

This dialog box is used to read the setting files. (Refer to "5.16.3 Window setting information (setting file)").  
When a setting file is loaded, the target window opens and the setting information at saving is restored.

Figure 6-68 Environment Setting File Load Dialog Box



- Opening
- Explanation of each area

### Opening

select [File] menu -> [Environment] -> [Open...].

### Explanation of each area

#### (1) Load file setting area

Look in:	This area is used to specify the file name to be loaded. A file name can be directly input from the keyboard, or selected from the list.
File name:	A string of up to 259 characters with a path name and an extension can be specified.
Files of type:	This area is used to specify the type (extension) of the file to be loaded. (Refer to "Table 5-22 Type of Setting Files".)

#### (2) Function buttons

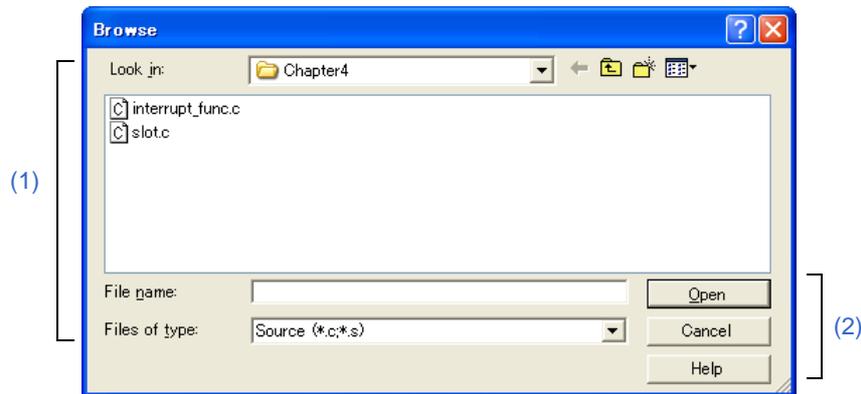
Open	Loads the selected file. After loading the file, this dialog box is closed.
Cancel	Closes this dialog box without executing anything.
Help	Displays this dialog box online help files.

## Browse dialog box

This dialog box is used to select the file to be set in the [Source Text Move dialog box](#).

**Remark:** If this dialog box is opened for the first time after the system has been started up, the folder first specified by the source path is displayed. When the dialog box is opened the second and subsequent times, the previously displayed folder is recorded and displayed again. If the [Cancel] button is clicked, however, the previously displayed folder is not recorded.

Figure 6-69 Browse Dialog Box



- [Opening](#)
- [Explanation of each area](#)

## Opening

Click the [Browse...] button in the target dialog box.

## Explanation of each area

### (1) Open file setting area

Look in:	This area is used to specify the file name to be opened. A file name can be directly input from the keyboard, or selected from the list. Up to 257 character string with a extension can be specified.
File name:	
Files of type:	This area is used to specify the type (extension) of the file to be opened. (Refer to " <a href="#">Table 5-4 File Type Can Be Displayed</a> ".)

**(2) Function buttons**

Open	Sets the selected file. After setting the file, this dialog box is closed.
Cancel	Closes this dialog box without executing anything.
Help	Displays this dialog box online help files.

# CHAPTER 7 COMMAND REFERENCE

This chapter explains the details of the command functions of the ID78K0R-QB.

- [Command Line Rules](#)
- [List of Command](#)
- [List of Aliases](#)
- [List of Variables](#)
- [List of Packages](#)
- [Key Bind](#)
- [Expansion Window](#)
- [Callback Procedure](#)
- [Hook Procedure](#)
- [Related Files](#)
- [Cautions](#)
- [Explanation of Commands](#)

## 7.1 Command Line Rules

The specification of command lines has the following rules:

- Command name, option, and argument are specified for command line.
- To divide words, a space (space key or tab key) is used.
- At the end of a line, a line feed character or a semicolon is used.
- When a command name and an option are entered to the point of identifiability, they are recognized.
- In script, command names have to be entered completely.

### Command format

```
command -options arg1 arg2 arg3 ...
```

## 7.2 List of Command

Table 7-1 Debugger Control Command List

Command Name	Function
<code>address</code>	Evaluation of address expression
<code>assemble</code>	Disassemble/line assemble
<code>batch</code>	Executing batch (with echo)
<code>breakpoint</code>	Setting/deletion of breakpoint
<code>dbgexit</code>	Terminating ID78K0R-QB
<code>download</code>	Download of files
<code>erase</code>	Erasure of the internal flash memory <b>[MINICUBE2]</b>
<code>extwin</code>	Creation of expansion window
<code>finish</code>	Returning from function
<code>go</code>	Continuous execution
<code>help</code>	Display of help
<code>hook</code>	Setting of hook
<code>ie</code>	Display/setting of IE register <b>[IECUBE]</b>
<code>inspect</code>	Symbol inspect
<code>ioport</code>	Add or delete an I/O port
<code>jump</code>	Jump to window
<code>map</code>	Setting / deleting memory mapping
<code>mdi</code>	Setting of expansion window
<code>memory</code>	Display/setting of memory
<code>module</code>	Display of the list of files and functions
<code>next</code>	Procedure step
<code>refresh</code>	Redrawing of window
<code>register</code>	Display/setting of register value and SFR value
<code>reset</code>	Reset
<code>run</code>	Reset and execution of CPU
<code>step</code>	Step execution
<code>stop</code>	Stop execution
<code>upload</code>	Upload
<code>version</code>	Display of the version information
<code>watch</code>	Display/setting of variables
<code>where</code>	Stack trace
<code>wish</code>	Start of Tclet

Command Name	Function
<a href="#">xcoverage</a>	Operation of coverage [IECUBE]
<a href="#">xtime</a>	Operation of timer [IECUBE]
<a href="#">xtrace</a>	Operation of tracer [IECUBE]

Table 7-2 List of Console/Tcl Commands

Command Name	Function
alias	Creation of another name
cd	Change of directory
clear	Clears the screen
echo	Echo
exit	Close/end
history	Display of history
ls	Display of files
pwd	Check of the directory
source	Execution of batch
time	Measurement of time for commands
<a href="#">tkcon</a>	Console control
unalias	Deletion of another name
which	Display of the command path or another name
Other	Based on Tcl/Tk 8.4

## 7.3 List of Aliases

The commands can be specified with other names by defining their aliases in the file "bin/tdtcl/aliases.tcl".

The aliases are described by default as shown below.

This file can be edited with an editor.

Table 7-3 Contents of File aliases.tcl

```
alias a assemble
alias b breakpoint
alias g go
alias i step -i
alias j jump
alias l download
alias m memory
alias n next
alias r run
alias s step
alias w watch
```

## 7.4 List of Variables

Table 7-4 List of Variables

Variable	Function
dcl (chip)	Chip name (read only)
dcl (prjfile)	Project file name (read only)
dcl (srcpath)	Source path (read only)
dcl (ieid)	IE type (read only)
dcl (iestat)	IE status (read only)
dcl (bkstat)	Break status (read only)
env (LANG)	Language
dcl_version	Dcl version (read only)

## 7.5 List of Packages

Table 7-5 List of Packages

Package	Function
tlctest	Restoration test
cwind	Automatic window control
BWidget	Toolkit
tcllib	Tcl library
mclistbox	Multi-column list box
combobox	Combo box

## 7.6 Key Bind

- tcsh + Emacs like
- Complement of command name [Tab]
- Complement of file name [Tab]
- HTML help [F1]

## 7.7 Expansion Window

The expansion windows can be created using Tk.

In the expansion windows, Widget is allocated with ".dcl" as a root instead of ".".

When the following script files are allocated in bin/tdtcl/tools/, an expansion window is added on selecting [Browse] menu -> [Others].

The mdi command, an exclusive command for expansion windows, has been added.

```
# Sample.tcl
wm protocol .dcl WM_DELETE_WINDOW { exit }
mdi geometry 100 50
button .dcl.b -text Push -command exit
pack .dcl.b
```

**Caution:** In the expansion windows, Tk menu commands cannot be used because of the restrictions of MDI windows.

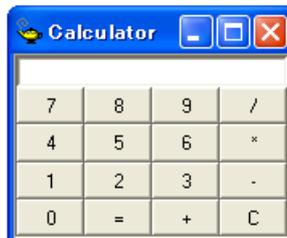
## 7.7.1 Samples (Calculator Script)

The script of the expansion window in which the calculator script is described and its execution screen are shown below.

### Script of expansion window

```
# Calculator.tcl
mdi geometry 100 100
set top .dcl
entry $top.e -relief sunken -textvariable v
frame $top.f -height 120 -width 120; pack $top.e -fill x; pack $top.f -fill both -expand 1
set i 0; set v {}; set r 0.25
foreach n {7 8 9 / 4 5 6 * 1 2 3 - 0 = + C} {
    if {$n == "=" || $n == "C"} {
        button $top.f.b$n -text $n
    } else {
        button $top.f.b$n -text $n -command "$top.e insert end $n"
    }
    place $top.f.b$n -relx [expr ($i%4)*$r] -rely [expr ($i/4)*$r] -relw $r -relh $r
    incr i
}
bind $top.f.bC <1> {$top.e delete 0 end}
bind $top.f.b= <1> {catch {expr $v} v}
```

Figure 7-1 Execution Screen



## 7.8 Callback Procedure

Expansion windows can hold `dcl_asyncproc` procedures called by asynchronous messages.

```
proc dcl_asyncproc {mid} {
  if {$mid == 19} {
    redraw
  }
}
```

The asynchronous message ID is passed for the argument of the `dcl_asyncproc` procedure.

The message IDs are shown below:

Table 7-6 Message ID

Message ID	Meaning
9	After changing configuration
10	After registering event
11	After deleting event
12	Before executing
13	After breaking
14	After resetting CPU
15	After resetting ID78K0R-QB
17	After changing extended option
18	After changing debugger option
19	After downloading
20	After changing memory or register
36	Before starting tracer <b>[IECUBE]</b>
37	After stopping tracer <b>[IECUBE]</b>
40	Before starting timer <b>[IECUBE]</b>
41	After stopping timer <b>[IECUBE]</b>
42	After clearing trace <b>[IECUBE]</b>
45	After resetting symbol

## 7.9 Hook Procedure

A hook can be set in the ID78K0R-QB using the hook procedure.

The hook procedures are shown below:

- BeforeDownload(Hook before downloading)
- AfterDownload(Hook after downloading)
- AfterCpuReset(Hook after CPU reset during break)
- BeforeCpuRun(Hook before starting execution)
- AfterCpuStop(Hook after breaking)

By using hook procedures, register values can be changed before downloading programs or after resetting the CPU.

An actual example of the procedure is shown below. A hook is valid till the ID78K0R-QB is closed.

### (1) [When hook is set with ID78K0R-QB control command]

- 1) Create script file a. with an editor.
- 2) Start up the ID78K0R-QB, select [Browse] menu -> [Console], and open the [Console window](#).
- 3) If the script file is executed in the window as below, the hook in the script file is set.

```
%hook test.tcl
```

### (2) [When hook is set on downloading of project file]

- 1) Create script file a. with an editor. **Note**
- 2) Start up the ID78K0R-QB and read test.prj. The hook in the script file is set.

```
proc BeforeDownload {} {
  register MM 0x7
  register PMC8 0xff
  register PMC9 0xff
  register PMCX 0xe0
}

proc AfterCpuReset {} {
  register MM 0x7
  register PMC8 0xff
  register PMC9 0xff
  register PMCX 0xe0
}
```

**Note:** Be sure that the script file name is the same as the project file.

Example:

The script file corresponding to test.prj is test.tcl.

Allocate test.prj, test.pri, and test.tcl in the same directory.

## 7.10 Related Files

Table 7-7 List of Related Files

File Name	Function
<code>aliases.tcl</code>	Executes when the <code>aliases.tcl</code> console is opened. Sets the default alias etc.
<i>Project filename.tcl</i>	Executes when the project file name.tcl project is opened. The following hooks can be used. BeforeDownload AfterDownload AfterCpuReset BeforeCpuRun AfterCpuStop
<i>Load modulefilename.tcl</i>	Executes when the load module name.tcl load module file is downloaded. The following hooks can be used. BeforeDownload AfterDownload AfterCpuReset BeforeCpuRun AfterCpuStop

## 7.11 Cautions

- The separator for file and path is a slash (/).
- When a console is open, error messages are output to the console.
- To terminate the command forcibly, close the console.
- The execution of external commands (DOS commands) is OFF by default.

## 7.12 Explanation of Commands

In this section, each command is explained using the format shown below.

### Command name

---

---

Describes the command name.

### Input format

---

Describes the input format of the command.

In the following explanation, italics indicate an Argument to be supplied by the user, while the argument enclosed in "?" may be omitted.

When a command name and an option are entered to the point of identifiability, they are recognized.

### Functions

---

Explains the functions of the command.

### Usage example

---

Shows an example of the usage of the command.

## address

---

---

address - Evaluation of address expression

### Input format

---

**address** *expression*

### Functions

---

Converts the address expression specified by *expression* into address.

### Usage example

---

```
(IDCON) 1 % address main
0xaa
(IDCON) 2 % address main+1
0xab
```

## assemble

assemble - Disassemble/line assemble

### Input format

**assemble** *?options? address ?code?*

### Functions

Assembles the character strings specified by *code* from the *address* specified by address.

When "." is specified for address, it is understood as an *address* continuing from the immediately previous assemble.

When *code* is omitted, it is assembled from the *address* specified by address.

The following are *options*: They are ignored for assembly.

<b>-code</b>	Command code is also displayed. It is ignored for assembly.
<b>-number</b> <i>number</i>	<i>Number</i> line is displayed. It is ignored for assembly.

### Usage example

```
(IDCON) 1 % assemble -n 5 main
0x000000aa B7      PUSH HL
0x000000ab B1      PUSH AX
0x000000ac 891C    MOVW AX,SP
0x000000ae D6      MOVW HL,AX
0x000000af A100    MOV  A,#0H
(IDCON) 2 % assemble main mov a,b
(IDCON) 3 % assemble . mov a,b
```

## batch

---

---

batch - Executing batch (with echo)

### Input format

---

**batch** *scriptname*

### Functions

---

Executes in batch with displaying files specified by *scriptname* on the screen.

Nesting is possible.

### Usage example

---

```
(IDCON) 1 % clear  
(IDCON) 2 % batch bat_file.tcl
```

## breakpoint

breakpoint - Setting/deletion of breakpoint

### Input format

**breakpoint** *?options? ?address1? ?address2?*

**breakpoint** -delete *brkno*

**breakpoint** -enable *brkno*

**breakpoint** -disable *brkno*

**breakpoint** -information

**breakpoint** -information *?-verbose?*

### Functions

Operates the breakpoint specified by *options* and *address*.

If a breakpoint can be set correctly, the breakpoint number is returned.

The following are *options*:

<b>-software</b>	A software break is specified.
<b>-hardware</b>	A hardware break is specified. (default)
<b>-execute</b>	The <i>address</i> execution break is set. (default)
<b>-beforeexecute</b>	The break before <i>address</i> execution is set.
<b>-read</b>	An <i>address</i> data read break is set.
<b>-write</b>	An <i>address</i> data write break is set.
<b>-access</b>	An <i>address</i> data access break is set.
<b>-size</b> <i>size</i>	The access size is set (8, 16).(Unit: bit)
<b>-data</b> <i>value</i>	The data condition is set.
<b>-datamask</b> <i>value</i>	The data mask is set.
<b>-pass</b> <i>value</i>	The path count is set.
<b>-information</b>	The list of breakpoints is displayed.
<b>-delete</b>	The breakpoint whose number is specified is deleted.
<b>-disable</b>	The breakpoint whose number is specified is disabled.
<b>-enable</b>	The breakpoint whose number is specified is enabled.
<b>-varbose</b>	If specified with -information, the fetailed contents of the breakpoint are displayed.

## Usage example

---

(IDCON) 1 % breakpoint main

1

(IDCON) 2 % breakpoint -i

1 Brk00001 enable rammon.c#17

(IDCON) 3 % breakpoint -software sub

2

(IDCON) 4 % breakpoint -i

1 Brk00001 enable rammon.c#17

2 Brk00001 enable rammon.c#8

(IDCON) 5 % breakpoint -disable 2

(IDCON) 6 % breakpoint -i

1 Brk00001 enable rammon.c#17

2 Brk00001 disable rammon.c#8

(IDCON) 7 % breakpoint -delete 1

2 Brk00001 disable rammon.c#8

(IDCON) 1 % breakpoint -information -verbose

1 Brk00001 enable [s]EX-B [Z]NC [AR]EQ [A]0x10D

## dbgexit

---

---

dbgexit - Terminating ID78K0R-QB

### Input format

---

**dbgexit** *?options?*

### Functions

---

Terminate the ID78K0R-QB.

The following are *options*:

<b>-saveprj</b>	Project is saved on terminating ID78K0R-QB.
-----------------	---------------------------------------------

### Usage example

---

(IDCON) 1 % dbgexit -saveprj

## download

download - Download of files

### Input format

**download** *?options? filename ?offset?*

### Functions

Downloads files specified with *filename* according to *options*. (The load module format files and HEX-format files are automatically recognized.)

If *offset* is specified, the address is shifted by the *offset* (if the data is in binary format, the load start address is specified for *offset*).

<b>-append</b>	Another HEX file is downloaded.
<b>-binary</b>	Binary format data is downloaded.
<b>-coverage</b>	Coverage data is downloaded.
<b>-nosymbol</b>	Download is executed. Symbol information is not read.
<b>-symbolonly</b>	Symbol information is read.
<b>-erase</b>	The contents of the internal flash memory are erased all before download. (Only a product with internal flash memory.)
<b>-reset</b>	CPU is reset after download.
<b>-information</b>	Download information is displayed.

### Usage example

(IDCON) 1 % download test.lmf

## **erase**

**[MINICUBE2]**

---

---

erase- Erasure of the internal flash memory

### **Input format**

---

erase

### **Functions**

---

Erases the internal flash memory.

### **Usage example**

---

(IDCON) 1 % erase

## extwin

---

---

extwin - Creation of expansion window

### Input format

---

**extwin** *scriptfile*

### Functions

---

Creates expansion window with *scriptfile*.

### Usage example

---

(IDCON) 1 % extwin d:/foo.tcl

## **finish**

---

---

finish - Returning from function

### **Input format**

---

**finish**

### **Functions**

---

Executes until it returns to the program that called the current function.

### **Usage example**

---

(IDCON) 1 % finish

## go

---

---

go - Continuous execution

### Input format

---

go *?options?*

### Functions

---

Executes program continuously. If -waitbreak is specified, the command waits until the program stops.

The following are *options*:

<b>-ignorebreak</b>	Breakpoint is ignored.
<b>-waitbreak</b>	The command waits for the program to stop.

### Usage example

---

(IDCON) 1 % go -w

## help

---

---

help - Display of help

### Input format

---

help

### Functions

---

Displays Dcl help.

### Usage example

---

(IDCON) 1 % help

For more information on a specific command, type HELP command-name

ASSOC Displays or modifies file extension associations.

AT Schedules commands and programs to run on a computer.

ATTRIB Displays or changes file attributes.

BREAK Sets or clears extended CTRL+C checking.

CACLS Displays or modifies access control lists (ACLs) of files.

CALL Calls one batch program from another.

CD Displays the name of or changes the current directory.

CHCP Displays or sets the active code page number.

CHDIR Displays the name of or changes the current directory.

CHKDSK Checks a disk and displays a status report.

CHKNTFS Displays or modifies the checking of disk at boot time.

CLS Clears the screen.

:  
:  
:

## hook

---

---

hook - Setting of hook

### Input format

---

**hook** *scriptfile*

### Functions

---

Sets the procedure for hook with *scriptfile*.

The hook setting is initialized when the project file is loaded and when the ID78K0R-QB is reset.

### Usage example

---

```
(IDCON) 1 % hook d:/foo.tcl
```

## ie

**[IECUBE]**

---

---

ie - Display/setting of IE register

### Input format

---

ie reg *address* ?*value*?

### Functions

---

The ie command depends on the IE.

When reg is specified for the subcommand, referencing and setting of the IE register is executed.

### Usage example

---

(IDCON) 1 % ie reg 0x100 1

## inspect

inspect - Symbol inspect

### Input format

**inspect** *?options? progname pattern*

### Functions

Searches and displays the load module symbol specified with *progname* using the regular expression of *pattern*.

The following regular expressions can be used.

<b>?</b>	Match 1 character
<b>*</b>	Match characters other than 0
<b>[chars]</b>	Match chars character. (Range specification such as [a-z/0-9] also possible.)
<b>\x</b>	Match character x. (? * [ ] \ specification also possible.)

The following are *options*:

<b>-nocase</b>	The case is distinguished.
<b>-address</b>	Displays in pair with symbol address.

### Usage example

(IDCON) 1 % inspect test1.out {[a-z]\*}

## ioport

ioport - Add or Delete an I/O port.

### Input format

**ioport** *options ?name? address ?attribute?*

### Functions

Add or delete an I/O port specified using name, address, size, or attribute.

Add an alias to the existing SFR.

Up to 15 one-byte alphanumeric characters and underscores can be specified for name. (The name is case sensitive.)

An address value or symbol can be specified for address.

An address in the internal memory area or SFR area can be specified.

The following are *attribute*.

<b>r</b>	Specify the read-only attribute.
<b>w</b>	Specify the write-only attribute.
<b>rw</b>	Specify the read/write attribute.
<b>rp</b>	Specify the read/read-protected attribute.
<b>rwp</b>	Specify the read/write/read-protected attribute.

The following are *options*:

<b>-add</b>	Add an I/O port specified using name, address, size, or attribute. Add an alias to the existing SFR. Attributes can be removed from the SFR attribute values, but only the read-protected attribute can be added.
<b>-delete</b>	Delete an I/O port specified using name or an SFR alias.
<b>-clear</b>	Delete all I/O ports or SFR aliases.
<b>-information</b>	Display a list of I/O ports or SFR aliases.

### Usage example

(IDCON) 1 % ioport -add PEMU0 0xffcf 8 w

(IDCON) 2 % ioport -delete PEMU0

(IDCON) 3 % ioport -add Port1 0xffcf 8 rwp

---

## jump

---

---

jump - Jump to window

---

### Input format

---

**jump** -source -line *filename* *?line?*

**jump** *?options?* *address*

---

### Functions

---

Displays the window specified by *options*.

<b>-source</b>	The Source window is displayed from the address specified by <i>address</i> .
<b>-assemble</b>	The Assemble window is displayed from the address specified by <i>address</i> .
<b>-memory</b>	The Memory window is displayed from the address specified by <i>address</i> .
<b>-line</b>	The command is moved to the line specified by <i>line</i> .
<b>-focus</b>	The Focus is moved to the window displayed.

---

### Usage example

---

(IDCON) 1 % jump -s main

(IDCON) 2 % jump -s -l mainfile.c 10

(IDCON) 3 % jump -m array

## map

map - Setting/deletion of memory mapping

### Input format

**map** *options address1 address2 ?accsize?*

### Functions

Sets, deletes, and displays memory mapping.

**Remark** The access size of 8 or 16 is specified by *accsize* (unit:byte, the default is 8).

The following are *options*:

<b>-target</b>	Target area is mapped.
<b>-protect</b>	I/O protect area is mapped.
<b>-rrm</b>	Start address of RRM area is set. <b>[MINICUBE2]</b>
<b>-stack</b>	Stack area is mapped. <b>[IECUBE]</b>
<b>-clear</b>	All the settings for the mapping are deleted.
<b>-information</b>	Refer to the setting for the mapping.

### Usage example

```
(IDCON) 1 % map -i
1: 0 0x3fff 8 {IROM}
2: 0x40000 0xefff - {NonMap}
3: 0xf0000 0xf07ff 8 {NonMap ExtSFR}
4: 0xf0800 0xfceff 8 {NonMap}
5: 0xfcf00 0xffed 8 {IRAM}
6: 0xf0000 0xf0fff 8 {Register}
7: 0xffff0 0xffff 8 {SFR}
```

## mdi

---

---

mdi - Setting of expansion window

### Input format

---

**mdi** geometry *?x y? width height*

**mdi** title *string*

### Functions

---

Sets the size and title name of the expansion window.

The command can be used only from the expansion window.

### Usage example

---

(IDCON) 1 % mdi geometry 0 0 100 100

(IDCON) 2 % mdi title foo

## memory

memory - Display/setting of memory

### Input format

**memory** *?options?* *address* *?value?*

**memory** *?options?* -fill *address1* *address2* *value*

**memory** *?options?* -copy *address1* *address2* *address3*

### Functions

Sets *value* in the memory of the *address* specified by *address* according to *options*.

If *value* is omitted, display the value of the memory of the address specified by *address*.

If -fill is specified, data from *address1* to *address2* is filled with *value*.

If -copy is specified, data from *address1* to *address2* is copied to *address3*.

The following are *options*:

<b>-byte</b>	Displayed/set in one-byte units. (default)
<b>-word</b>	Displayed/set in word units.
<b>-fill</b>	The data is filled in.
<b>-copy</b>	The data is copied.
<b>-noverify</b>	Verification is not executed on writing.

**Remark:** If one of the following operations is performed during user program execution, the CPU is stopped momentarily and execution continues.

- Referencing a memory area other than the RRM area [**IECUBE**]
- Setting of a memory area other than the internal RAM area [**IECUBE**]
- Referencing a memory area in the RRM area [**MINICUBE2**]
- Setting of a memory area [**MINICUBE2**]

### Usage example

```
(IDCON) 1 % memory 100
0x10
(IDCON) 2 % memory 100 2
(IDCON) 3 % memory 100
0x02
(IDCON) 4 % memory -fill 0 1ff 0
```

## module

---

---

module - Display of the list of files and functions

### Input format

---

**module** *programe* *?filename?*

### Functions

---

Displays the list of files and functions of the load module specified by *programe*.

If *filename* is not specified, the list of files is displayed.

If *filename* is specified, the list of functions of the specified files is displayed.

### Usage example

---

```
(IDCON) 1 % module rammon.lmf
1: rammon.c
(IDCON) 2 % module rammon.lmf rammon.c
1: rammon.c sub1
2: rammon.c main
```

## next

---

---

next - Procedure step

### Input format

---

**next** *?options?*

### Functions

---

Executes the procedure steps. If functions are called, the step stops after executing function.

The following are *options*:

<b>-source</b>	The command is executed in source line units. (default)
<b>-instruction</b>	The command is executed in command units.

### Usage example

---

(IDCON) 1 % next -i  
(IDCON) 2 % next -s

## **refresh**

---

---

refresh - Redrawing of window

### **Input format**

---

**refresh**

### **Functions**

---

Redraws the window and updates the data.

### **Usage example**

---

(IDCON) 1 % batch foo.tcl  
(IDCON) 2 % refresh

## register

register - Display/setting of register value and SFR value

### Input format

**register** *?options? regname ?value?*

### Functions

Sets *value* in the register specified with *regname*.

If *value* is omitted, displays the value of the register specified by *regname*.

The following are *options*:

<b>-force</b>	Compulsory reading or writing is executed.
<b>-bankno</b> <i>bankno</i>	Specifies the bank number.

**Remark:** If either of the following operations is performed during user program execution, the CPU is stopped momentarily and execution continues. **[IECUBE]**

- Setting a register or SFR
- Referencing a control register other than the PC or SFR

### Usage example

```
(IDCON) 1 % register pc
0x100
(IDCON) 2 % register pc 200
(IDCON) 3 % register pc
0x200
```

## reset

---

---

reset - Reset

### Input format

---

**reset** *?options?*

### Functions

---

Resets the ID78K0R-QB , CPU, symbols or events.

If options are omitted, the CPU is reset.

The following are *options*:

<b>-cpu</b>	CPU is reset. (default)
<b>-debugger</b>	The ID78K0R-QB is reset.
<b>-symbol</b>	Symbol is reset.
<b>-event</b>	All events and software breaks are reset.

### Usage example

---

(IDCON) 1 % reset

## run

---

---

run - Reset and execution of CPU

### Input format

---

`run ?options?`

### Functions

---

Resets the program and executes it.

If `-waitbreak` is not specified, the command does not wait until the program stops.

The following are *options*:

<b>-waitbreak</b>	The command waits for the program to stop.
-------------------	--------------------------------------------

### Usage example

---

(IDCON) 1 % run

(IDCON) 2 % run -w

## step

---

---

step - Step execution

### Input format

---

**step** *?options?*

### Functions

---

Executes step execution.

If functions are called, the command stops at the head of the functions.

The following are *options*.

<b>-source</b>	The command is executed in source line units. (default)
<b>-instruction</b>	The command is executed in instruction units.

### Usage example

---

(IDCON) 1 % step -i

(IDCON) 2 % step -s

## **stop**

---

---

stop - Stop executing

### **Input format**

---

**stop**

### **Functions**

---

Stops the program forcibly.

### **Usage example**

---

(IDCON) 1 % run  
(IDCON) 2 % stop

## tkcon

tkcon - Console control

### Input format

**tkcon** cmd *?arg?*

### Functions

Controls the Console window.

This command is one of the console/Tcl commands.

<b>tkcon buffer <i>?size?</i></b>	Sets and references the maximum buffer size (number of lines) of the console. If the specified buffer size is exceeded, the excessive lines are deleted from the oldest order. The default buffer size is 1,000 lines.
<b>tkcon close</b> <b>tkcon destroy</b>	Close the Console window.
<b>tkcon font <i>?fontname?</i></b>	Sets and references the fonts used in the Console window.
<b>tkcon gets</b>	Performs standard inputs such as Stdin. Opens a dialog box.
<b>tkcon history <i>?-newline?</i></b>	Displays the command history.
<b>tkcon save <i>?filename? ?type?</i></b>	Saves the buffer data for the Console window as a file. When the file name or the file type is omitted, a dialog box is opened. Select the type from all, history, stdin, stdout, and stderr.
<b>tkcon version</b>	Displays the console version.

### Usage example

(IDCON) 1 % tkcon save c:/temp/logfile.txt all

---

## upload

---

upload - Upload

### Input format

---

**upload** *?options? filename address1 address2*

**upload** -coverage *filename*

### Functions

---

Saves the memory data within the specified range or coverage data in a file.

The following are *options*:

<b>-binary</b>	The data is saved in binary format.
<b>-coverage</b>	The coverage data is saved. When saving coverage data, all the specified range of the coverage data is saved in the file (specification of start/end addresses not required).
<b>-intel</b>	The data is saved in Intel HEX format. (default).
<b>-motorola</b>	The data is saved in Motorola HEX format.
<b>-tektronix</b>	The data is saved in Tektronix HEX format.
<b>-force</b>	The file is overwritten.

### Usage example

---

(IDCON) 1 % upload -b foo.hex 0 0xffff

## version

---

---

version - Display of the version information

## Input format

---

**version**

## Functions

---

Displays the version of the ID78K0R-QB.

## Usage example

---

(IDCON) 1 % version

```
GUI           : VX.XX [XX XXXX 200X]
Devicefile    : 78K/0R[uPD78xxxxx] VX.XX
Debugger      : 78K0R-QB Debugger VX.XX XX XXXX 200X]
Executer      : 78K0R IECUBE Executer VX.XX
Micro program : OCD Control Code VX.XX
Assembler     : 78K0R Asm/Disasm VX.XX XX XXXX 200X]
Tcl/Tk        : 8.4.X
```

## watch

---

watch - Display/setting of variables

### Input format

---

**watch** *?options? variable ?value?*

### Functions

---

Displays and sets the variables.

The following are *options*:

<b>-binary</b>	The value is displayed in binary digits.
<b>-octal</b>	The value is displayed in octal digits.
<b>-decimal</b>	The value is displayed in decimal digits.
<b>-hexdecimal</b>	The value is displayed in hexadecimal digits.
<b>-string</b>	The value is displayed in character strings.
<b>-sizeof</b>	The size, instead of the value, of variables is displayed in decimal digits.
<b>-encoding <i>name</i></b>	Encoding during character string display is specified. By default, system encoding is used. <i>name</i> (encoding name) is based on the Tcl specification (shiftjis, euc-jp, etc.).

### Usage example

---

```
(IDCON) 1 % watch var
0x10
(IDCON) 2 % watch -d var
16
(IDCON) 3 % watch array[0] 0xa
```

## where

---

---

where - Stack trace

## Input format

---

**where**

## Functions

---

Executes the back-trace of the stack.

When the message "---Infomation below might be inaccurate" is displayed, the subsequent displays may be incorrect.

## Usage example

---

Example1

(IDCON) 1 % where

1: test2.c#sub2(int i)#13

2: test.c#num(int i)#71

3: test.c#main()#82

Example2

(IDCON) 1 % where

1: func2.c#func2()#8

---Infomation below might be inaccurate.

2: func2.c#func(register int\_i)#34

3: func2.c#main()#77

## wish

---

---

wish - Startup of Tclet

### Input format

---

**wish** *scriptname*

### Functions

---

Starts up the script using Tk (Tclet).

The expansion window can be created with Tclet.

### Usage example

---

(IDCON) 1 % wish test.tcl

## xcoverage

[IECUBE]

---

---

xcoverage - Operation of coverage

### Input format

---

**xcoverage** *option*

### Functions

---

Operates coverage.

The following are *options*:

<b>-clear</b>	Clears the coverage memory.
---------------	-----------------------------

### Usage example

---

(IDCON) 1 % xcoverage -clear

---

## xtime

**[IECUBE]**

---

---

xtime - Operation of timer

---

### Input format

---

**xtime** *option*

---

### Functions

---

Operates timer.

The following are *options*:

<b>-start</b>	Timer starts on executing the program.
<b>-stop</b>	Timer stops on executing the program.
<b>-gobreak</b>	Time from Go to Break is displayed in nsec.

---

### Usage example

---

(IDCON) 1 % xtime -start

(IDCON) 2 % xtime -stop

## xtrace

[IECUBE]

xtrace - Operation of tracer

### Input format

**xtrace** -dump *?-append? frameno ?filename?*

**xtrace** -start

**xtrace** -stop

**xtrace** -clear

**xtrace** -addup *?bool?*

**xtrace** -mode *?mode?*

### Functions

Operates tracer.

The following are *options*:

<b>-start</b>	The tracer starts on executing the program.
<b>-stop</b>	The tracer stops on executing the program.
<b>-clear</b>	Clears the trace memory.
<b>-dump</b>	The trace data is dumped. (default) The dump result is redirected to the console window. If the file name is specified, the dump result is written in the file.
<b>-append</b>	The dump result is added to a file.
<b>-addup <i>?bool?</i></b>	Whether the time tag is totaled or not is selected. When <i>bool</i> is omitted, the current mode is displayed.
<b>-mode <i>?mode?</i></b>	The trace control mode (any one of: all, cond, nonstop, fullstop, fullbreak, delaystop, delaybreak, machine, or event) is selected. When <i>mode</i> is omitted, the current mode is displayed.
<b>-priority <i>?mode?</i></b>	The trace data (timetag or data) to be given priority and to be recorded is selected. When <i>mode</i> is omitted, the current mode is displayed. The default setting is "timetag".

### Usage example

```
(IDCON) 1 % xtrace -start
(IDCON) 2 % xtrace -stop
(IDCON) 3 % xtrace -dump 3
_ 01685 2 000000BC M1 br _sub2+0x2
_ 01686 4 0000009A BRM1 st.w r6, 0x8[sp]
_ 01687 3 0000009E BRM1 st.w r0, 0x0[sp]
(IDCON) 4 % xtrace -clear
(IDCON) 5 % xtrace -addup true
```

# APPENDIX A EXPANSION WINDOW

- [Overview](#)
- [Sample Window List](#)
- [Activation](#)
- [Explanation of Each Sample Window](#)

## A.1 Overview

With the ID78K0R-QB, the user can create custom windows in addition to the existing windows.

The Tcl (Tool Command Language) interpreter and the commands for controlling the debugger are implemented in the ID78K0R-QB. Users can create windows using this Tcl.

The ID78K0R-QB is supplied with samples of the following expansion windows.

## A.2 Sample Window List

Table A-1 List of Expansion Window (Sample)

Window Name	Function
<a href="#">List window</a>	Displays a list of the source files and functions.
<a href="#">Grep window</a>	Searches a character string.
<a href="#">Hook window</a>	Sets the hook procedure.
<a href="#">SymInspect window</a>	Searches through a list of properly described symbols.
<a href="#">OpenBreak window</a>	OpenBreak functions

## A.3 Activation

The expansion window can be activated by selecting List, Grep, Hook, SymInspect in [Others] on the [Browse] menu.

**Remark:** Each .tcl file is installed in NEC Electronics Tools\ID78K0R-QB\Vx.xx\bin\idctl\tools

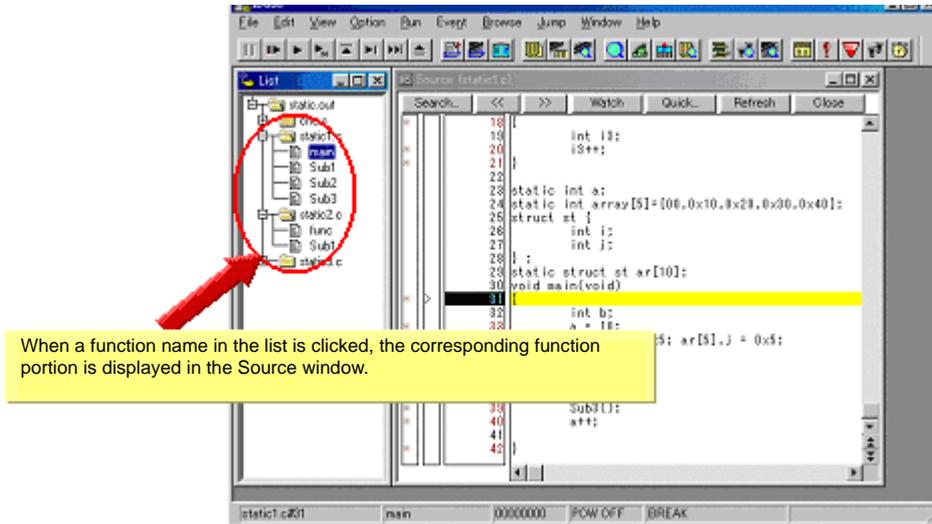
## A.4 Explanation of Each Sample Window

The ID78K0R-QB provides the sample window below.

## List window

The lists of the source files and functions are displayed in a tree format in this window. When a function name in the list is clicked, the corresponding source is displayed.

Figure A-1 List Window

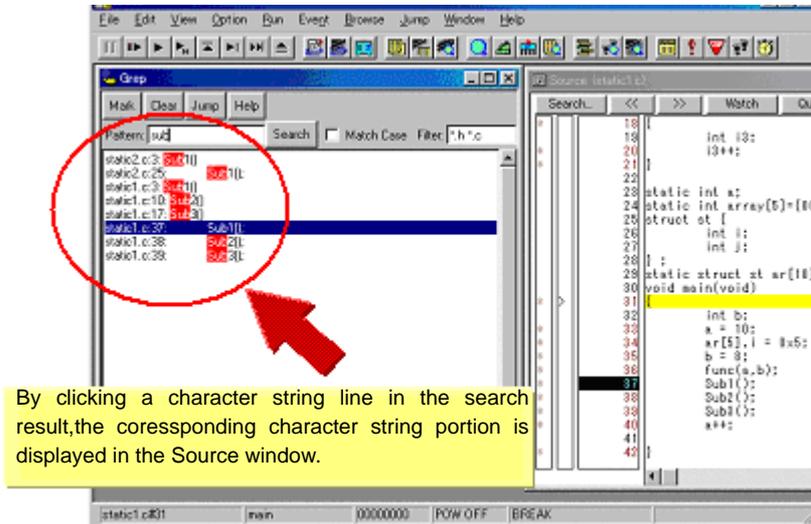


## Grep window

Search for a character string is performed in the files under the source path.

When the search result is clicked, the corresponding source is displayed.

Figure A-2 Grep Window



Object	Function
Pattern	Input the character string to be searched.
[Mark] button	Marks the searched character string.
[Clear] button	Clears the marking.
[Jump] button	Put the cursor on a section in the search result and click this button to open the corresponding file.
Match Case	Select whether or not to distinguish uppercase and lowercase.
Filter	Specify the type of the file to be searched.

## Hook window

This window is used to set a hook to the debugger, using a hook procedure.

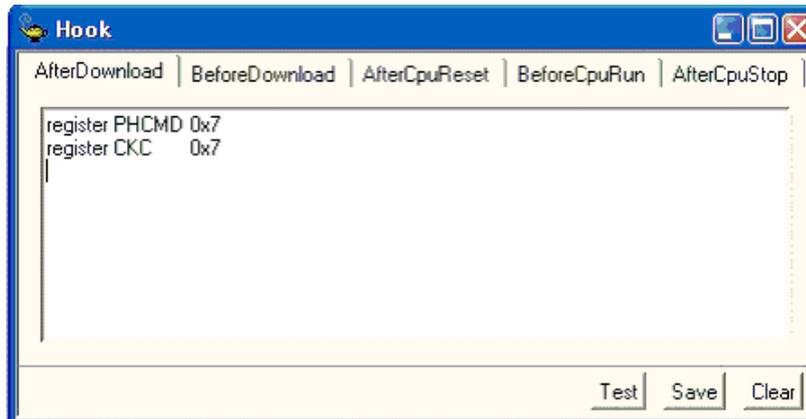
The hook procedure enables changing the register value before downloading a program, or after a CPU reset.

On this window, a hook can be set by using the following five tabs.

- [AfterDownload] tab: Hook after downloading
- [BeforeDownload] tab: Hook before downloading
- [AfterCpuReset] tab: Hook after CPU reset during break
- [BeforeCpuRun] tab: Hook before start of execution
- [AfterCpuStop] tab: Hook after break

If the setting is saved as "project-file-name.tcl" in the folder where the project is stored, the setting is executed when the project is next opened.

Figure A-3 Hook Window



Object	Function
[AfterDownload] tab	Hook after downloading After downloading is performed, the register values input to the tab are automatically overwritten by the specified value.
[BeforeDownload] tab	Hook before downloading Before downloading is performed, the register values input to the tab are automatically overwritten by the specified value.
[AfterCpuReset] tab	Hook after CPU reset during break after resetting CPU, the register values input to the tab are automatically overwritten by the specified value.
[BeforeCpuRun] tab	Hook before starting execution before starting execution, the register values input to the tab are automatically overwritten by the specified value.
[AfterCpuStop] tab	Hook after breaking After breaking, the register values input to the tab are automatically overwritten by the specified value.

---

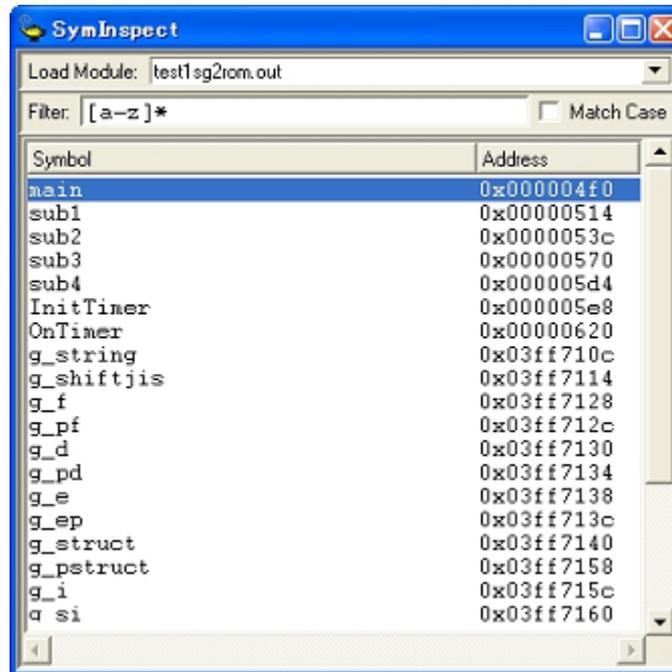
Object	Function
[Test] button	All the commands described on the tabs are tested.
[Save] button	Saves all the tab contents to a file. If the ID78K0R-QB was activated from a project file, the file is saved as "project-file-name.tcl".
[Clear] button	Clears all the descriptions on the tabs.

**Remark:** Specify the general-purpose register and the SFR for the register name.

## SymInspect window

This window displays the list of the symbols and addresses of loaded module files, and is used for searching the list for the properly described symbol.

Figure A-4 SymInspect Window



Object	Function
Load Module:	Selects a load module file.
Filter:	Specifies a properly described symbol so that the symbol is retrieved.
Match Case	In Filter:, specify to differentiate or not differentiate case sensitivity. Check this box to differentiate case sensitivity.
Symbol	Displays the symbols. Clicking this icon has the symbols sorted in alphabetical order.
Address	Displays the addresses. Clicking this icon has the addresses sorted in ascending numerical order.

Context Menu	Function
Copy	Copies the selected address to the clipboard.
Jump to Source	Jumps from the address in the selected line to the identical address displayed in the <a href="#">Source window</a> .
Jump to Assemble	Jumps from the address in the selected line to the identical address displayed in the <a href="#">Assemble window</a> .
Jump to Memory	Jumps from the address in the selected line to the identical address displayed in the <a href="#">Memory window</a> .

## OpenBreak window

[IECUBE]

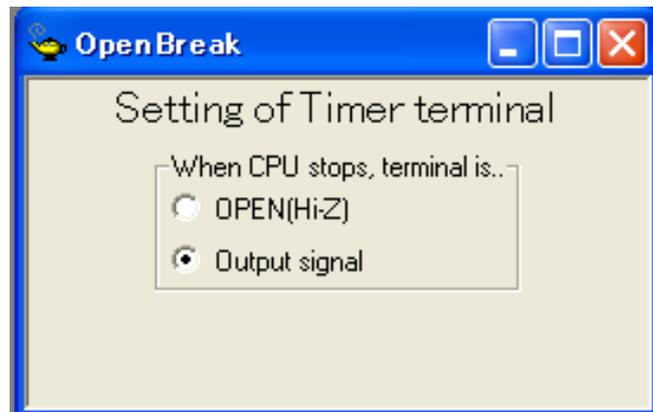
This window is used to set the open break function. [IECUBE]

The open break function is used to set to Hi-Z the timer pins, which control a motor, so as to stop the motor safely if the motor control signal is not fed back due to CPU stoppage (break) and the signal may have an adverse effect on the motor.

The device and pins subject to manipulation by the open break function are listed in the following table. When the open break function is used, the motor stops upon a break and thus the program cannot be re-executed. In such a case, reset the CPU.

Target Device	Function	Target Pins
78K0R/Ix3	6-phase PWM output function	TO02, TO03, TO04, TO05, TO06, TO07
	Triangular wave PWM output function	TO02, TO03, TO06, TO07

Figure A-5 OpenBreak Window



Bit	Function
OPEN [Hi-Z]	The open break target pin becomes the Hi-Z state after the CPU is stopped.
Output signal	The open break target pin outputs the signal even after the CPU is stopped.

- When the open break function is set, keep the OpenBreak window open or minimized. (The setting is cleared when the window is closed.)
- The setting made in the OpenBreak window can be saved into the current project file while the OpenBreak window is open. The setting is restored together with the settings made in other windows when the project file is opened next time.
- The open break target pins are set to the Hi-Z state by selecting “Category A (Timer)” in the Peripheral Break area in the Configuration dialog box. [MINICUBE2]

# APPENDIX B CAUTIONS

This chapter describes general cautions for this product. For details on individual restrictions or cautions, refer to the Operating Precautions document for each product.

## (1) Cautions on switching log-on user of Windows XP

If a log-on user is switched using the Windows XP Switch User function while the ID78K0R-QB is starting, the operation of the ID78K0R-QB that was used by the previous user is not guaranteed. The operation of the ID78K0R-QB used by the new user is also not guaranteed. (The ID78K0R-QB prohibits duplicated activation on one computer, but duplicated activation is possible by using the Windows XP Switch User function.)

## (2) Cautions related to Windows standby and hibernate modes

If Windows enters standby or hibernate mode while the ID78K0R-QB is starting, the operation of the ID78K0R-QB is not guaranteed after Windows returns from the mode. In such a case, restart the emulator and ID78K0R-QB.

## (3) Cautions on changing load module file names

If a project file is saved while PM+ and the ID78K0R-QB are active and the load module file name is changed in PM+, and the debug button of PM+ is clicked while the ID78K0R-QB has not been started, the load module file of the previous name will be downloaded.

If the load module file name is changed, the debug button of PM+ must therefore be clicked while the ID78K0R-QB is already started.

If the ID78K0R-QB is not already started, click the debug button once, start the ID78K0R-QB and then click the debug button again.

## (4) Cautions on Return Out execution for recursive functions

If Return Out is executed during processing of a recursive function (function that calls itself), the execution cannot return to the desired position. The operation varies as follows according to how many times the recursive function is called.

Current PC Line Before Return Out Execution	Return Out Execution Result
During processing of the 1st recursive function	Returns to the position at which the function was called (desired operation).
During processing of the 2nd to nth recursive function	Breaks during processing of the nth recursive function.
After processing of the nth recursive function ends	Returns to the position at which the function was called (desired operation).

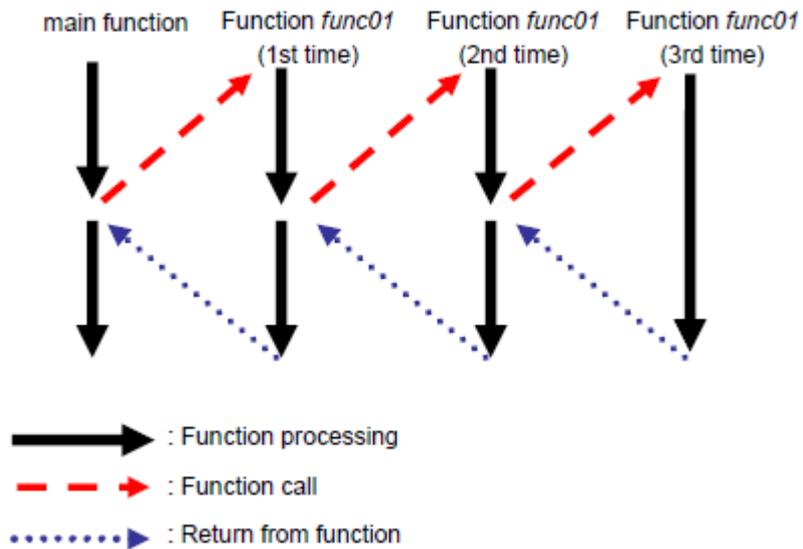
Even if the execution fails to return correctly, the subsequent program execution is performed correctly.

The following shows an operation example.

```
void func01(char count)
{
    count --;
    if (count > 0)
    {
        func01(count);    /* 2nd and 3rd call */
    }
}

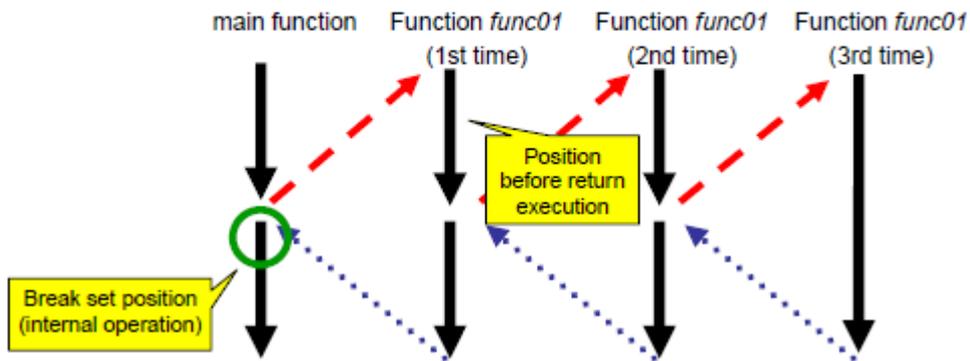
void main(void)
{
    func01(3);    /* 1st call */
}
```

In this example, "func01" is the recursive function. The above program flows as follows.



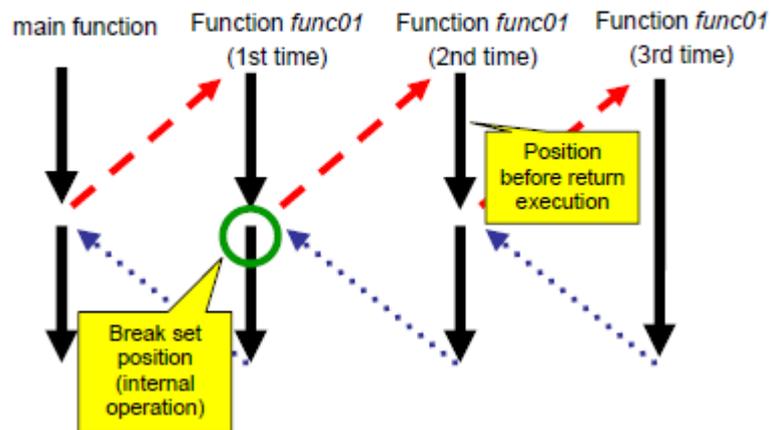
The following explains the Return Out operation in detail.

**[Return Out during processing of the 1st recursive function]**



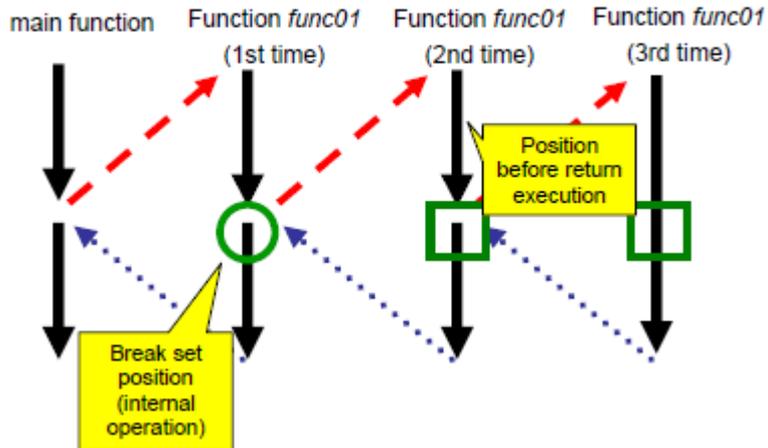
If Return Out is executed during the first processing of function *func01*, the execution returns to the circle-marked position. The ID78K0R-QB sets a breakpoint at this position internally and executes the program. In this case, the program will stop at the desired position (circle-marked position).

**[Return Out during processing of the 2nd to *n*th recursive function (1)]**



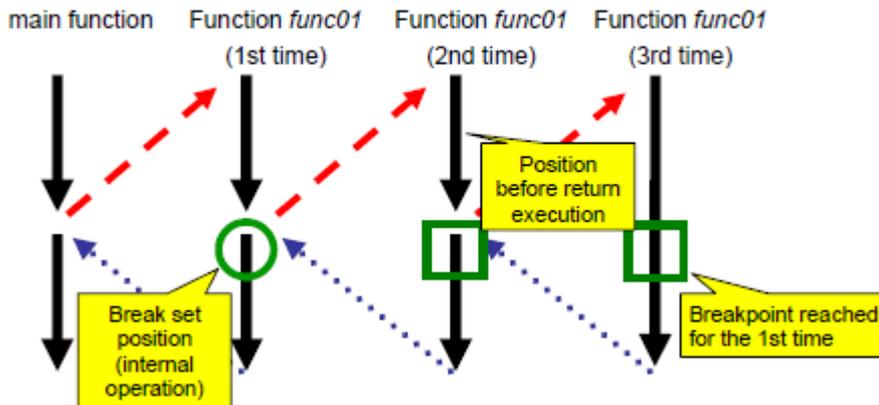
If Return Out is executed during the second and third processing of function *func01*, as in the above case, the ID78K0R-QB sets a breakpoint at the desired return position (circle-marked position).

[Return Out during processing of the 2nd to *n*th recursive function (2)]

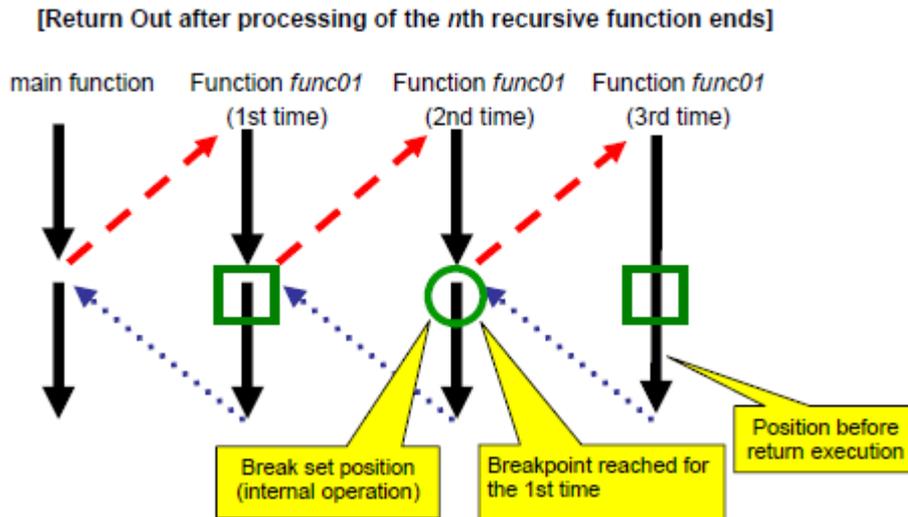


However, the set breakpoint is also included in the recursive processing, so breaks also occur at the square-marked positions.

[Return Out during processing of the 2nd to *n*th recursive function (3)]



As a result of program execution, a break occurs at the first breakpoint (the rightmost square-marked position in the above figure), which is not the desired position. This is because the execution passes through the desired position before reaching the desired position. Although it is not the desired result, program execution is performed as is coded. Even if the program is reexecuted from this position, the operation results are normal.



If Return Out is executed after the third `func01` processing (the rightmost square-marked position in the above figure), it is desired to return to the circle-marked position, but a breakpoint is also set at the square-marked position. This execution breaks at the desired position, however. This is because the first breakpoint to be reached during recursive processing is the circle-marked position.

#### (5) Caution when a function entity is defined in header file

If a function entity is defined in the header file, breakpoint setting, stepwise execution, or Return Out cannot be performed in the Source window. Moreover, information on this function is not displayed in the Code Coverage window.

The case where “a function entity is defined in the header file” means that function processing itself is described in the header file. It does not include the case where only function declaration, such as prototype declaration and EXTERN declaration, is described in the header file.

#### (6) Caution on `longjmp` function

If Step In or Next Over is performed for the `longjmp` function, execution processing may not complete and may wait for a time-out. If Return Out is performed in a function that called the `longjmp` function, breaks may not occur.

#### (7) Standby mode

The standby mode is released under any of the following conditions.

- <1> The standby mode is entered during stepwise execution.
- <2> Execution is stopped by a forced break in standby mode
- <3> A standby instruction is executed before the execution stops due to a break after execution or an access break, which takes several extra instructions until the execution stops.
- <4> A break after execution or an access break is used as an event of a snap-shot or event DMM, a standby instruction is executed before the execution stops due to the break, each processing is complete, and then the execution is restarted. (IECUBE)

**(8) Mirrored non-map area**

A mirrored non-map area is not indicated as “??” . If IROM = 64 KB and MAA = 1, for example, the non-map area starting from address 0x10000 is mirrored. At this time, 0x10000 to 0xF0FFF are indicated as “??” in the Memory window, but values can be read from or written to 0xF1000 to 0xFFEFF.

**(9) External memory**

<1> For external memory, the code coverage cannot be measured and memory accessing cannot be monitored. (IECUBE)

<2> When the RAM monitor function is off, data cannot be read from or written to the external RAM during execution. (IECUBE)

<3> The following two restrictions apply when a ROM is connected as an external memory. (IECUBE)

(a) If stepwise execution is performed for the following instructions in instruction mode, an extra instruction is executed.

a. RETI/RETB instruction

An instruction immediately after returning from the interrupt servicing will be executed.

b. Conditional skip instruction (condition is not satisfied)

The instruction next to the conditional skip instruction will be executed.

(b) Execution does not stop as is expected if one of the following operations is performed. The execution continues, or stops at the next breakpoint. (IECUBE, MINICUBE2)

a. Step In source mode, whose execution stops at a location in external ROM

b. Next Over whose execution stops at a location in external ROM

c. Return Out whose execution stops at a location in external ROM

d. Come Here executed with the cursor placed in external ROM

**(10) Caution on using project files with different emulators**

If a project file for IECUBE is opened by using MINICUBE2 or a project file for MINICUBE2 is opened by using IECUBE and then the project file is saved, information is lost partially and an event function or a timer function may not operate normally. Save project files for IECUBE or MINICUBE2 independently and use them according to the emulator used.

**(11) Cautions when IECUBE is connected**

<1> Execution operation while external reset signal is input

When the SFR window or the like is displayed and execution or step-wise execution is performed while an external reset is not masked (the TARGET RESET check box is not selected in the Configuration dialog box) and an external reset signal has been input, the program has to wait for a timeout in communication with the emulator, which drastically degrades the operability.

<2> Coverage function

(a) The PC points to the instruction that has not been executed, but for code coverage measurement, one or two bytes from the top of the instruction pointed to by the PC are counted by the code coverage execution. In addition, the address at which a break occurs is also counted.

(b) Display of an access being executed does not necessarily change in the order of access in the Memory window.

<3> Display of internal ROM area during execution

During execution, data in the internal ROM area before starting execution is saved temporarily and displayed. The display is therefore not updated until the execution breaks, even if the data is overwritten by flash self-programming or event DMM.

<4> Events

Up to two extra instructions after a snap-shot event, event DMM, trace event, or timer event is measured.

# APPENDIX C INPUT CONVENTIONS

- Usable Character Set
- Symbols
- Numeric Values
- Expressions And Operators
- File Names

## C.1 Usable Character Set

Table C-1 List of Character Set

Classification	Character
Alphabetic characters	Uppercase: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Lowercase: a b c d e f g h i j k l m n o p q r s t u v w x y z
Numerals	0123456789
Character equivalent to alphabetic character	@ _ ?

Table C-2 List of Special Characters

Character	Name	Usage
(	Left parenthesis	Changes operation order.
)	Right parenthesis	Changes operation order.
+	Plus	Addition operator or positive sign
-	Minus	Subtraction operator or negative sign
*	Asterisk	Multiplication operator or indirect reference operator
/	Slash	Division operator
%	Percent	Remainder operator
~	Tilde	Complement operator
	Vertical line	Bit sum operator
^	Circumflex	Bit difference operator
&	Ampersand	Bit product operator or address operator
.	Period	Direct member operator or bit position specifier
,	Comma	Delimiter between operands

Character	Name	Usage
[	Left bracket	Array subscript operator or indirect display symbol
]	Right bracket	
!	Exclamation	Absolute addressing start symbol
\$	Dollar	Relative addressing start symbol
#	Sharp mark	Symbol indicating an immediate value

## C.2 Symbols

(1) A symbol is composed of either of the following characters.

A to Z, a to z, @, \_ (underbar), . (period) and 0 to 9

(2) A symbol must start with a character other than numerals 0 to 9.

(3) Uppercase characters (A to Z) and lowercase characters (a to z) are distinguished.

(4) A symbol must be no more than 2048 characters long (if a symbol of more than 2048 characters is defined, only the first 2048 characters are valid).

(5) A symbol is defined by loading a load module file.

(6) Symbols are classified into the following types by the valid range:

Global symbol (assembly language, structured assembly language, C language)

Static symbol (C language)

- In-file static symbol
- In-function static symbol

Local symbol (C language)

- In-module local symbol (assembly language, structured assembly language)
- In-file local symbol
- In-function local symbol
- In-block local symbo

(7) The following symbols are available for each language used:

### **Assembly language, structured assembly language**

label name, constant name, bit symbol name

### **C language**

Variable name (including pointer variable name, enumeration type variable name, array name, structure name, and union name)

Function name, label name

Array element, structure element, union element, bit field (if the symbol is an array, structure, or union)

- (8) A symbol can be described instead of an address or numeric value.
- (9) The valid range of a symbol is determined based on the source debug information when the source file is assembled or compiled.
- (10) Describe only the symbol name of a global symbol.
- (11) A local symbol is expressed in pairs with a file name.

### C.3 Numeric Values

The following four types of numeric values can be used. The input format of each type is as shown below.

The suffix (**bold**) and the alphabetic characters of hexadecimal numbers may be uppercase or lowercase characters. If the first character is A to F, 0 must be prefixed to it.

In the input field of ID78K0R-QB, decimal numbers or hexadecimal numbers are alternately selected, depending on the default radix.

Table C-3 Input Format of Numeric Values

Numeric Value	Input Format
Binary number	n <b>Y</b> n...n <b>Y</b> (n=0,1)
Octal number	n <b>O</b> n...n <b>O</b> (n=0,1,2,3,4,5,6,7) n <b>Q</b> n...n <b>Q</b> (n=0,1,2,3,4,5,6,7)
Decimal number	n n...n n <b>T</b> n...n <b>T</b> (n=0,1,2,3,4,5,6,7,8,9)
Hexadecimal numbers	n n...n n <b>H</b> n...n <b>H</b> <b>0xn</b> <b>0xn</b> ...n (n=0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)

## C.4 Expressions And Operators

### (1) Expressions

An expression consists of constants, register names, SFR name and symbols coupled by operators.

If SFR name, label name, function name, or variable name is described as a symbol, an address is calculated as the value of the symbol.

The elements making up an expression, except operators, are called terms (constants and labels). Terms are called the first term, the second term, and so on, starting from the left.

### (2) Operators

The following operators of the C language can be used:

Table C-4 List of Operators

Symbol	Meaning	Explanation
<b>Arithmetic operator</b>		
+	Addition	Returns the sum of the first and second terms.
-	Subtraction	Returns the difference between the first and second terms.
*	Multiplication	Returns the product of the first and second terms.
/	Division	Divides the value of the first term by the value of the second term, and returns the integer of the results.
MOD %	Remainder	Divides the value of the first term by the value of the second term, and returns the remainder of the results.
- sign	Unary operator (negative)	Returns 2's complement of the value of the term.
+ sign	Unary operator (positive)	Returns the value of the term.
<b>Logical operator</b>		
NOT ~	Negation	Logically negates each bit of the term, and returns the results.
AND &	Logical product	Obtains the logical product of the values of the first and second terms on each bit, and returns the results.
OR 	Logical sum	Obtains the logical sum of the values of the first and second terms on each bit, and returns the results.
XOR ^	Exclusive logical sum	Obtains the exclusive logical sum of the values of the first and second terms on each bit, and returns the results.
<b>Shift operator</b>		
SHR >>	Right shift	Shifts the value of the first term by the value (number of bits) of the second term to the right, and returns the results. As many 0s as the number of shifted bits are inserted in the higher bits.
SHL <<	Left shift	Shifts the value of the first term by the value (number of bits) of the second term to the left, and returns the results. As many 0s as the number of shifted bits are inserted in the lower bits.

Symbol	Meaning	Explanation
<b>Byte separation operator</b>		
HIGH	Higher byte	Of the lowest 16 bits of the term, returns the higher 8 bits.
LOW	Lower byte	Of the lowest 16 bits of the term, returns the lower 8 bits.
<b>Word separation operator</b>		
HIGHW	Higher word	Of the 32 bits of the term, returns the higher 16 bits.
LOWW	Lower word	Of the 32 bits of the term, returns the lower 16 bits.
<b>Other</b>		
(	Left parenthesis	Performs the operation in ( ) before the operation outside ( ). '(' and ')' are always used in pairs.
)	Right parenthesis	

### (3) Rules of operation

Operations are performed according to the priority of the operators.

Table C-5 Operator Priority

Priority	Operators
1 Higher	(, )
2	+ sign, - sign, NOT, ~, HIGHT, LOW, HIGHW, LOWW
3	*, /, MOD, %, SHR, >>, SHL, <<
4	+, -
5	AND, &
6 Lower	OR,  , XOR, ^

- If the priorities of the operators are the same, the operation is performed from the left toward the right.
- Performs the operation in ( ) before the operation outside ( ).
- Each term in an operation is treated as unsigned 32-bit data.
- All operation results are treated as unsigned 32-bit data.
- If an overflow occurs during operation, the lower 32 bits are valid, and the overflow is not detected.

### (4) Terms

To describe a constant for a term, the following numeric values can be described.

Table C-6 Range of Radixes

Radix	Range
Binary number	0Y <= value <= 11111111111111111111111111111111Y (32 digits)
Octal number	0O <= value <= 3777777777O
Decimal number	-2147483648 <= value <= 4294967295 (A negative decimal number is internally converted into a 2's complement.)
Hexadecimal numbers	0H <= value <= 0FFFFFFFH

## C.5 File Names

The following regulations apply to the source file names and load module file names.

### (1) Source file names and load module file names

File names are composed of a to z, A to Z, 0 to 9, ., \_, +, and -.

File names must start with a character other than ".".

File names cannot be prefixed or suffixed by a period (.) or space.

File names are not case-sensitive.

A file names consists of up to 259 characters including the path.

### (2) Other file names

Other file names comply with windows file name regulations.

The following characters cannot be used in file names.

\ / : \* ? " < > | ;

File names cannot be prefixed or suffixed by a period (.) or space.

File names are not case-sensitive.

A file name consists of up to 259 characters including the path.

## C.6 Quantitative Upper Limit List

The following list is the ID78K0R-QB quantitative upper limits.

Table C-7 Quantitative upper limit list

Window Name		
(Common to windows)	File name (not including path name)	255 single-byte characters
	File name (including path name)	259 single-byte characters
	Number of character strings that can be displayed per line	400 single-byte characters
	Number of character strings that can be searched for	150 single-byte characters
	Number of characters that can be used for symbol names (function names, variable names, etc.)	2,048 single-byte characters
Download dialog box	Number of load module files that can be downloaded at a time	1
Trace View window	Number of frames that can be displayed	128K frames
Event dialog box	Number of character strings that can be used for event names	8 single-byte characters
Watch window	Number of lines that can be displayed	10,000 lines
	Number of array dimensions of that can be displayed	4
Source window	Number of source file and include file lines that can be displayed per line	65,535 lines
Assemble window	Highest address that can be displayed	0xFFFFE
Add I/O Port dialog box	Number of character strings that can be used for I/O register names and SFR names	15 single-byte characters
Software Break Manager	Number of software breaks that can be set	2,000 points

# APPENDIX D KEY FUNCTION LIST

Table D-1 Key Function List

Key	Function
BackSpace	Deletes one character before the cursor and moves the cursor to the position of the deleted character. At this time, the character string following the cursor moves forward.
Delete	<ul style="list-style-type: none"> <li>- Deletes one character after the cursor and move the character string following the cursor forward.</li> <li>- Deletes a various event condition selected in the Event Manager or each event dialog box.</li> <li>- Deletes the data selected in the Watch window.</li> </ul>
Insert	Alternately selects the insert mode and overwrite mode in the Source window and Assemble window. However, this key is invalid in the Memory, Register, and SFR windows, and only the overwrite mode can be used as an input mode.
PrintScreen	Loads the entire display screen to the clipboard as a bitmap image (function of windows).
Esc	<ul style="list-style-type: none"> <li>- Closes the pull-down menu.</li> <li>- Closes the modal dialog box.</li> <li>- Restores the input data.</li> </ul>
Alt	Moves the cursor to the menubar.
End	Moves the cursor to the end of the line.
Home	Moves the cursor to the beginning of the line.
PageUp	Scrolls the screen one screen up. The cursor also moves up to the top of the screen.
PageDown	Scrolls the screen one screen down. The cursor also moves up to the top of the screen.
Space	Inserts one blank character.
Tab	Moves the cursor to the next item.
Up arrow key	Moves the cursor up. If the cursor is at the bottom of the screen, scrolls the screen up one line at a time.
Down arrow key	Moves the cursor down. If the cursor is at the top of the screen, scrolls the screen down one line at a time.
Right arrow key	Moves the cursor to the left. If the cursor is at the left most position on the screen, scrolls the screen one column to the right.
Left arrow key	Moves the cursor to the right. If the cursor is at the right most position on the screen, scrolls the screen one column to the left.
Enter	<ul style="list-style-type: none"> <li>- Sets the input data.</li> <li>- Presses the default push button.</li> </ul>
F1	Opens the Help window.
F2	Forcibly stops program execution. Same function as [Run] menu -> [Stop].

Key	Function
F3	Resets the CPU. Same function as [Run] menu -> [CPU Reset].
F4	Resets the CPU and executes the program. Same function as [Run] menu -> [Restart].
F5	Executes the program. Same function as [Run] menu -> [Go].
F6	Executes the program to the cursor position in the Source or Assemble window. Same function as [Run] menu -> [Come Here].
F7	The user program is real-time executed until execution returns. Same function as [Run] menu -> [Return Out].
F8	Step execution. Same function as [Run] menu -> [Step In].
F9	Sets a breakpoint at cursor position in Source or Assemble window. Same function as [Run] menu -> [Break Point].
F10	Next step execution. Same function as [Run] menu -> [Next Over].
F11	Sets or deletes a software breakpoint. Same function as [Run] menu -> [Software Break Point].
Shift+End	Expands the selection range to the end of the line.
Shift+Home	Expands the selection range to the beginning of the line.
Shift+Left arrow key	Expands the selection range one character to the left.
Shift+Right arrow key	Expands the selection range one character to the right.
Shift+F6	Executes the program from the cursor position in the Source or Assemble window. Same function as [Run] menu -> [Start From Here].
Shift+F9	Resets the CPU. Same function as [Run] menu -> [CPU Reset].
Ctrl+End	Displays the last line. The cursor will also move to the last line.
Ctrl+Home	Displays the first line. The cursor will also move to the first line.
Ctrl+Left arrow key	Moves the cursor one word to the left. If the cursor at the left most position on the screen, scrolls the screen one column to the right.
Ctrl+Right arrow key	Moves the cursor one word to the right. If the cursor at the right most position on the screen, scrolls the screen one column to the left.
Ctrl+F5	Ignores break points being set, and executes the program. Same function as [Run] menu -> [Ignore break points and Go].
Ctrl+F9	Sets the address at the cursor position in the Source window or Assemble window to the PC. Same function as [Run] menu -> [Change PC].
Ctrl+A	Selects all the events registered to the Event Manager. Same function as [View] menu -> [Select All Event] in the Event Manager.
Ctrl+C	Copies a selected character string and saves it to the clipboard buffer.
Ctrl+D	Disassembles and displays the results from the jump destination address specified by the data value selected in the current window. Opens the Assemble window. Same function as [Jump] menu -> [Assemble].

Key	Function
Ctrl+E	Opens the source file displayed in the active Source window with the editor specified by the PM+ when the PM+ is running. Same function as [Edit] menu -> [Edit Source].
Ctrl+G	Performs a search. Opens the search dialog box corresponding to the current window. Same function as [View] menu -> [Search...].
Ctrl+J	Moves the display position. Opens the each dialog box, depending on the current window. Same function as [View] menu -> [Move...].
Ctrl+M	Displays the memory contents from the jump destination address specified by the data value selected in the current window. Opens the Memory window. Same function as [Jump] menu -> [Memory...].
Ctrl+O	Loads a view file, source file, or text file. Opens the View File Load dialog box. The operation will differ depending on the extension of the file. view file: Displays the file in the corresponding window. Others: Displays the file in the Source window. Same function as [File] menu -> [Open...].
Ctrl+S	Saves the data displayed in the current window to the view file. Same function as [View] menu -> [Save...].
Ctrl+U	Displays the corresponding source text and source line, using the data value selected in the current window as the jump destination address. Opens the Source window. Same function as [Jump] menu -> [Source Text].
Ctrl+V	Pastes the contents of the clipboard buffer to the text cursor position.
Ctrl+W	Temporarily displays the contents of the specified data. Opens the Quick Watch dialog box. Same function as [View] menu -> [Quick Watch...].
Ctrl+X	Cuts a selected character string and saves it to the clipboard buffer. Same function as [Edit] menu -> [Cut].
Ctrl+Shift+Left arrow key	Expands the selection range one word to the left.
Ctrl+Shift+Right arrow key	Expands the selection range one word to the right.

# APPENDIX E MESSAGES

- Display Format
- Types of Messages
- Message Lists

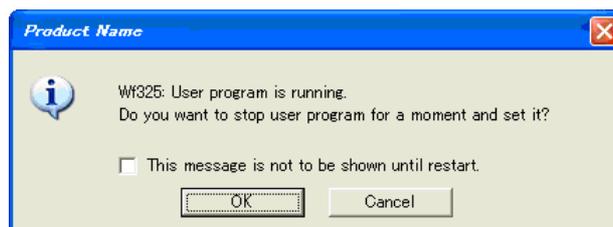
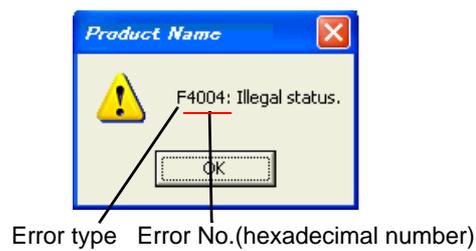
## E.1 Display Format

Messages are output to the error/warning dialog box.

By pressing the F1 key while the error/warning dialog box is open, the related online help files are displayed.

If the [This message is not to be shown until restart.] check box is selected in dialog boxes that appears with this check box, the message will no longer displayed while the ID78K0R-QB is running. When the ID78K0R-QB is started next time, this message can be displayed (default: cleared).

Figure E-1 Error/Warning Dialog Box



## E.2 Types of Messages

The ID78K0R-QB outputs the following types of messages.

Table E-1 Types of Messages

Types	Meaning
A	Abort Error Stops processing, and terminates the debugger. If this error occurs, debugging cannot be continued.
F	Fail Stops processing, and opened windows and dialog boxes are closed.
W	Warning Stops processing, but opened windows and dialog boxes are not closed.

## E.3 Message Lists

< From X0000 > < X1000~ > < From X2000 > < From X3000 > < From X4000 > < From X5000 > < From X6000 > < From X7000 > < From X8000 > < From X9000 > < From Xa000 > < From Xb000 > < From Xc000 > < From Xd000 > < From Xe000 > < From Xf000 >

### (1) From X0000

F0002: This feature is not supported.
F0100: Can not communicate with ICE. Please confirm the installation of the device driver for the PC interface board.  1) The driver may not be correctly installed. Reinstall the driver. 2) Check the power supply to the emulator and the cable connections. <b>[MINICUBE2]</b>
A0101: Can not find initialization file (expc.ini).
A0102: Host name not found.
F0103: Data transfer to ICE is timed out. Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board.  1) Check the cable connections and the switches on the emulator.
F0104: Data receive from ICE is timed out. Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board.
A0105: Failed in reading device file (d1xxx.78k).  1) Necessary files may be damaged. Reinstall the device file.
A0106: Illegal data received.  1) Check the power of the in-circuit emulator, cable connections, and setting of the interface board and restart the debugger.
A0107: Can not communicate with ICE.

A0108: Failed in reading initialization file (expc.ini).
A0109: Can not communicate with ICE. Please terminate the debugger and check the power of ICE or the connection of cable then restart the debugger. 1) An error may have occurred during USB communication (such as disconnection of power or cable) or IECUBE is faulty. <b>[IECUBE]</b> 2) USB communication was disconnected. Terminate the debugger, check the USB connection, and then restart the debugger. <b>[MINICUBE2]</b>
A010a: Cannot run debugger and a utility at the same time. 1) The QB-Programmer, MINICUBE2 self-testing tool, or OCD Checker is running, so terminate it first. 2) MINICUBE2 is being used by ID78K0R-QB, so terminate it first. 3) Another debugger is running. Terminate it first and then restart the ID78K0R-QB.
A01a0: No response from the CPU. Please confirm the signal of the CLOCK or RESET WAIT, HLDRQ and so on. 1) The target device is not responding. Terminate the debugger, check the connection, and then restart the debugger.
A01a1: Failed in reading ie703000.ie.
A01a2: Break board is not connected.
A01a3: Emulation board is not connected.
A01a4: Board configuration of ICE is not consistent.
A01a5: POD/EM1 board is not connected.
A01a6: Executor is running. 1) Another debugger is running. Terminate it first and then restart the ID78K0R-QB.
A01a7: Failed in reading micro program file (m0xxx.78k).
A01a8: Failed in reading initialization file (expc.ini).
A01ad: Please update the device driver for the PC interface board. 1) The device driver may be old. Install the latest device driver.
A01ae: Failed in reading configuration file (lv8hw.ini). Failed in reading initialization file (expc.ini).
A01af: Failed in executing monitor command.
A01b0: Can not communicate with monitor program. Please check the availability of communication port, the setting of CPU board or the type of cable.
A01b1: Can not communicate with monitor program. Please terminate the debugger and check the power of CPU board or the connection of cable then restart the debugger.
A01b1: Can not communicate the wireless utility. Click the setting of the wireless utility. 1) Click the setting of the wireless utility.

F0200: Verification error occurred. Failed in writing memory. 1) External memory could not be accessed, as it is not set. Change the register values necessary for accessing the external memory using the <a href="#">SFR window</a> or <a href="#">Hook Procedure</a> before download.
F02a0: Bus hold error. 1) CPU is in the bus-hold status. Reset the debugger.
F02a2: Can not compulsory break.
F02a3: Reset under continuation.
F02d2: Not enough memory for trace-buffer.
F0300: User program is running.
F0301: User program is being broken.
F0302: User program is being traced.
F0303: Not traced.
F0304: Trace memory is not set.
F0306: No trace block exists.
F0307: No event condition exists.
F0308: No timer measurement is done.
F0309: No trigger frame exists.
F030a: Tracer is being stopped.
F030b: Specified snap-event has not been registered.
F030c: Specified stub-event has not been registered.
F030d: Timer is running.
F030e: Memory copy area is overlapped.
F030f: Trace has been already set.
F0310: Event condition is not set.
F0311: Too many valid timer event conditions.
F0312: Specified timer event is not set.
F0313: Illegal map range. 1) Check the map range in the <a href="#">Configuration dialog box</a> . When mapping to external memory has been performed, change the register values necessary for accessing the external memory using the <a href="#">SFR window</a> or <a href="#">Hook Procedure</a> before download.
F0314: Only trace delay mode can set with delay trigger.
F0315: Delay trigger cannot set without trace delay mode.
F0316: Overflowed the number of mapping.
F03a0: Target is not turned on. 1) Check the target power supply. Check the cable connecting the in-circuit emulator and target board. Check that the VDD signal is input to the connector of the target board.
F03a1: Step execution is being done.

F03a2: Timer and Tracer are running.
F03a3: Event link and BRS events are mixed.
F03d0: Back-trace is being executed.
F03d1: Back-trace is being stopped.
F03d2: Back-trace execution point overrun oldest frame.
F03d3: Register status or Memory status cannot be set up other than Phase1 of event link.
F03d4: No back-trace information exists.
F03d5: Last command can not be backstepped.
F0400: Illegal condition. 1) Settings of the used in-circuit emulator and those of the <a href="#">Configuration dialog box</a> may not match. Check the Chip selection.
F0401: Result of timer measurement overflowed.
F0402: Too many event conditions with path count.
F0403: Too many address range conditions.
F0404: Too many simultaneously-usable-event conditions.
F0405: Too many snap-events.
F0406: Too many stub-events.
F0407: Too many initialization data.
F0408: Too large search data (> 16 byte).
F0409: Too large search data (> search range).
F040a: Too many Linking-event conditions.
F04a0: Software break conditions number overflow.
F04a1: Not enough memory for emulation.
F04a2: Too many partition of bus size.
F04a3: Too many execution-event conditions.
F04a4: Too many bus-event conditions.
A0600: Not enough memory for buffer. 1) There is not enough system memory. Close the applications being executed and the open files.
A0601: Not enough resource of operating system.
F0b20: This event number can not be used.
F0b61: Section Trace event conditions overflow.
F0b66: Cannot use the break before execution event and the software break at the same time. 1) This is because a break before execution is used for implementing a software break (when MINICUBE2 is connected). <b>[MINICUBE2]</b>
F0b80: Reset by hardware error.
F0c00: Monitor file read error. 1) Necessary files may be damaged. Reinstall the debugger.

A0c01: During access of register, CPU did time out. 1) Check the clock signal, etc. The register value may not be correct.
A0c02: During access of memory, CPU did time out. 1) Check the HOLD signal, WAIT signal, clock signal, etc. The memory value may not be correct.
A0c03: During access of I/O register, CPU did time out. 1) Check the HOLD signal, WAIT signal, clock signal, etc. The I/O register value may not be correct.
F0c20: Guarded area can not be accessed.
F0c21: Memory was unready status.
F0c22: Memory unready status was canceled.
F0c23: Bus hold under continuation. 1) Check the setting of the target board, or mask the HOLD pin.
F0c24: It cannot shift to debug mode. 1) Check the clock signal. This may be caused by a stopped clock or a slow clock.
F0c25: Flash macro service ROM was accessed or stepped in. 1) Please perform [Go] execution or CPU reset.
F0c26: FLMD terminal is in a write-protected state. 1) FLMD is not in the write-enabled status. Check the status of the FLMD0 and FLMD1 pins.
F0c27: Security flag is in a write-protected state. 1) The security flag of the flash memory has disabled writing, block erasure, or chip erasure. Nothing can be written to the flash memory.
F0c28: Internal RAM is not enough, the writing to flash memory is not made. 1) The internal RAM size is less than 4 KB and flash self-programming cannot be executed.
F0c29: The blank check of flash memory failed.
F0c2a: The erasing of flash memory failed.
F0c2b: The writing of flash memory failed.
F0c2c: The internal verification of flash memory failed.
F0c2d: Failed in writing flash memory.
F0c2e: There is no response from flash macro service.
F0c2f : Response from flash macro service is not right.
F0c30: Flash I/O register operation prohibition setup needs to be canceled.
F0c31: STOP mode under continuation. Can not compulsory break. Please release STOP mode or reset the CPU.
F0c32: Please write in flash memory in the single chip mode 0.
F0c33: Disabling the on-chip debug function is prohibited.
F0c34: Writing to the on-chip debug reserved area is prohibited.
F0c35: Abnormal Internal ROM size. The size is different from the default of the device.
F0c36: Abnormal Internal ROM size. The size is different from the default of the device.

F0c37: The voltage is too low to operate flash programming.
F0c38: Extended monitor area is not blank.
F0c39: Real-time RAM monitoring failed.
F0c40: Status of effective event conditions cannot be changed.
F0c41: Coverage test is being executed.
F0c42: Monitor has failed in shift in the debugging mode. Please reset the CPU.
F0c43: Can not communicate with ICE. Please confirm the power of ICE, connection of the interface cable. 1) The switch setting may be wrong if a desktop computer is used and two or more PC cards are inserted. Check the setting. Or it may have malfunctioned (when N-Wire CARD, MINICUBE2 connected). <b>[MINICUBE2]</b> 2) Check the power to the in-circuit emulator and cable connections. The switch setting may be wrong if a desktop computer is used and two or more PC cards are inserted. Check the setting. Or it may have malfunctioned. <b>[IECUBE]</b>
F0c44: Coverage test is being executed.
F0c45: Inside of Power off reset emulation cannot carry out program execution.
F0c46: Change of Internal ROM size or Internal RAM size or RAM monitor or DMM is not valid during Flash Self Emulation.
F0c48: Flash programming is not permitted by the debugger setting. 1) Flash programming is disabled in the debugger. Select "Permit" in the Flash Programming area in the <a href="#">Configuration dialog box</a> .
F0c60: Event before execution cannot be set up other than break conditions.
F0c61: Can not register event numbers which can not be used for hardware break.
F0c62: Event numbers reserved for hardware breaks can not be used.
F0c63: Event link conditions cannot set.
F0c64: Too many ROM-emulation-RAM areas.
F0c67: Writing of flash memory during block is not made.
F0c70: DCU cannot be accessed. 1) IE may be malfunctioning. <b>[IECUBE]</b>
F0c71: Reset cannot be performed. 1) Check the clock signal. This may be caused by a stopped clock or a slow clock.
F0c72: Monitor memory cannot be accessed. 1) Revise the Main OSC value in the <a href="#">Configuration dialog box</a> . If this does not solve the problem, IE may be malfunctioning. <b>[IECUBE]</b>
F0c73: Monitor execution cannot be performed. 1) IE may be malfunctioning. <b>[IECUBE]</b>

<p>F0c74: CPU register cannot be accessed.</p> <p>1) The device file selection may be incorrect. Select a device file that supports the target chip in Chip Selection in the <a href="#">Configuration dialog box</a>. If this does not solve the problem, the IE may be malfunctioning. <b>[IECUBE]</b></p>
<p>F0c75: Monitor has failed in shift in the debugging mode. Please reset the CPU.</p>
<p>F0c76: Initial state at the time of DCU access start is unusual.</p> <p>1) The device file selection may be incorrect. Select a device file that supports the target chip in Chip Selection in the <a href="#">Configuration dialog box</a>. If this does not solve the problem, the IE may be malfunctioning. <b>[IECUBE]</b></p>
<p>F0c77: DCU access is unusuall.</p> <p>1) IE may be malfunctioning. <b>[IECUBE]</b></p>
<p>F0c78: Failed in reading of trace data.</p>
<p>F0c79: The on-chip debug function had been disabled in the device.</p> <p>1) Codes have been written to the internal flash memory by the QB-Programmer (QBP) or PG-FP4 (FP4). Erase the internal flash memory using the QBP or FP4. If codes are written to the internal flash memory using the QBP or FP4, on-chip debugging is disabled even if it is enabled by the setting to address C3H.</p>
<p>F0ca0: Error occurred inside debugger.</p> <p>1) This message is displayed when the timer setting for an asynchronous message has failed, or an attempt was made to perform flash programming to an illegal address. Terminate the debugger and then restart it.</p>
<p>F0ca1: Monitor file not found.</p> <p>1) Necessary files may be damaged. Reinstall the debugger.</p>
<p>F0ca2: This device file does not include the on-chip debug information.</p> <p>1) An attempt was made to start with a device file not supporting on-chip debugging. The device file may be old. Install the latest device file (when the N-Wire CARD, MINICUBE2 is connected). <b>[MINICUBE2]</b></p> <p>2) IE may be malfunctioning. <b>[IECUBE]</b></p>
<p>F0ca3: Unsupported information is included in the on-chip debug information in the device file.</p> <p>1) An unknown flag is included in the on-chip debug information of the device file. The exec module may be old. Install the latest exec module.</p>
<p>F0ca4: This device file does not include the IECUBE information.</p> <p>1) An attempt was made to start with a device file not supporting IECUBE. The device file may be old. Install the latest device file.</p>
<p>F0caf: Trace block can not be stepped over.</p>

**(2) X1000~**

A1000: Failed in initializing ICE.

A1001: No entry exists for specified number.
A1002: Can not relocate internal RAM.
F1003: Illegal relocation address.
F1004: Illegal condition.
A1005: Invalid attribute.
F1006: Illegal address.
A1007: Not enough memory on ICE.
A1008: Not enough memory for tables. 1) There is not enough system memory. Close the applications being executed and the open files.
A1009: Already initialized.
A100a: Not initialized.
F100b: User program is running.
F100c: Different bus size has been already specified.
F100d: Too large bus size.
F100e: Too large bus partition size.
W100f: Target is not turned on.
F1010: Illegal map range.
F1011: Failed in setting internal ROM and RAM.
F1012: This feature is not supported.
F1013: No terminal name.
W1014: Data is not exist.
A1015: Programmable-IOR does not exist.
F1016: Programmable-IOR does not movable. 1) Necessary files may be damaged. Reinstall the latest device file.
F1017: I/O Protect mapping is possible a target attribute only.
F1018: Illegal Internal ROM size.
A10ff: Can not communicate with ICE.
A1dbe: Error occurred inside debugger.

**(3) From X2000**

F2000: Illegal SFR name.
A2001: Illegal address.
F2002: User program is running.
F2003: Illegal SFR number.
F2004: Illegal bit number.
W2005: SFR of Read Protect attribute was specified.

F2006: Hidden SFR was specified.
F2007: SFR of ban read or write was specified.
F2008: SFR not existing was specified.
A2009: Device file is damaged or error is in file.
F200a: Illegal value specified for SFR.
A200b: Can not copy.
A200c: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
W200d: No initialize data for SFR.
F200e: SFR area can not be accessed.
A20ff: Can not communicate with ICE.
A2222: Illegal condition.

**(4) From X3000**

F3000: No mapped address was accessed. 1) The allocation addresses of the program and the addresses of the debugger may not match. Set the mapping to the external memory in the <a href="#">Configuration dialog box</a> according to the allocation addresses specified in the link directive file on compilation. When mapping to external memory has been executed, change the register values necessary for accessing the external memory using the <a href="#">SFR window</a> or <a href="#">Hook Procedure</a> before download.
F3001: Memory has different value.
F3002: Illegal start address.
F3003: Illegal end address
F3004: Illegal start address and end address.
F3005: Illegal condition.
F3006: User program is running.
F3007: Verification error.
F3008: No condition specified.
F3009: Parameter size does not align with access size alignment.
F300a: Specified address does not align with access size alignment.
F300b: Source address does not align with access size alignment.
F300c: Destination address does not align with access size alignment.
F300d: Illegal end address.
F300e: Different access size in specified area.
F300f: Different access size both in source and destination areas.
F3010: Different access size in destination area.
F3011: Different access size, source & destination.

A3012: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
F3013: Failed in writing DMM.
F3014: Overflowed mapping area.
F3015: Processing was interrupted.
F3016: This feature is not supported.
A30ff: Can not communicate with ICE.

**(5) From X4000**

F4000: Can not delete specified event. 1) The specified event cannot be deleted as it is being used under another condition. Invalidate it for other usages before deleting.
F4001: Illegal table number.
F4002: Illegal start address.
F4003: Illegal end address.
F4004: Illegal status.
F4005: Illegal data.
F4006: Specified event number has been already used.
F4007: Too many same events are registered.
F4008: Specified event has not been registered.
F4009: Illegal data size.
F400a: Illegal mode.
F400b: Setting value is inaccurate.
F400c: Event link conditions cannot be used for section trace conditions.
F400d: Too many identical events are registered ( $\geq 32767$ ).
F400e: Specified event condition does not exist.
F400f: Illegal event link condition.
F4010: Function not found.
A4011: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
F4012: Timer is being disabled.
W4013: Access size is different from its mapped bus size.
F4014: Can not use software break.
F4015: Can not use event condition specifying address range.
F4016: Can not change event condition.
F4017: Can not access word at odd address.

A4018: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
F4019: This feature is not supported.
F401a: No Event.
F401b: Can not use tag-event.
W401c: Software break can not be set on this area.
F401d: Start event and end event of timer are not made to the same setup.
F401e: Too many trace-events.
F401f: Path count cannot be set up.
F4020: Address range cannot be set up in event before execution.
F4021: Event conditions number overflow.
F4022: Software DMM conditions number overflow.
F4023: Real-time call conditions number overflow.
F4024: Software break call conditions number overflow.
F4025: Illegal snap condition.
F4026: Too many event conditions cannot be set as Phase1 and Phase2 of event link conditions.
F4027: Software break conditions number which can be set as internal ROM was overflow.
F4318: Illegal memory bank setting.

**(6) From X5000**

A5000: Illegal device file type.
A5001: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
A5002: Can not open device file.
A5003: Reading of device file went wrong.
A5004: Can not close device file.
A5005: Illegal device file format. 1) Necessary files may be damaged. Reinstall the device file.
A5006: Failed in initializing ICE.
A5007: Device file has broken or error is in a file.
F5008: Can not open device file. 1) Necessary files may be damaged. Reinstall the device file.
F5009: Can not open ie703000.ie.
F500a: Specified device file is illegal version. 1) Necessary files may be damaged. Reinstall the device file.
W500b: Specified device file does not relocate internal RAM.

A500c: Failed in reading expc.ini.
A500d: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
W500e: No tag data which it was going to refer to device file.
A5300: Illegal device file type.
A5301: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
A5302: Can not open database file. 1) Necessary files may be damaged. Reinstall the debugger and device file.
A5303: Reading of database file went wrong.
A5304: Can not close database file.
A5305: Illegal database file format. 1) Necessary files may be damaged. Reinstall the debugger, and device file.
A5306: Database information has been already initialized.
A5307: Database information does not exist.
F5308: Can not open specified database file. 1) Necessary files may be damaged. Reinstall the debugger.
F5309: Specified database file is illegal version. 1) Necessary files may be damaged. Reinstall the debugger, and the device file.

**(7) From X6000**

F6000: Current function does not exist.
F6001: Illegal symbol name.
F6002: Illegal condition.
F6003: Illegal function name.
F6004: Overflowed output buffer size.
F6005: Illegal expression.

**(8) From X7000**

F7000: Illegal mode.
F7001: User program is running.
F7002: User program has been stopped.
F7003: Trace enabled.
F7004: Trace memory is not set.
F7005: Function return address does not exist, can not do step execution.
W7010: No source information exists.

W7011: Unknown result of step execution.
A7012: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
A70fe: Bus hold error. 1) CPU is in the bus-hold status. Reset the debugger.
A70ff: Can not communicate with ICE.
F7801: End waiting state of step execution was canceled.
F7802: End waiting state of step execution was canceled.
F7f00: Aborted step execution.
F7f02: Suspended step execution.
A7f03: Failed in canceling RUN/STEP.
F7f04: Can not execute non-mapped area.
F7f05: This feature is not supported.

**(9) From X8000**

F8000: Specified file was not found.
F8001: Illegal line number.
F8002: Current information is not set.
F8003: Illegal address.
F8004: This feature is not supported.

**(10) From X9000**

A9000: Specified register symbol does not exist.
A9001: Specified register symbol ID does not exist.
F9002: Illegal value.
A9003: Illegal condition.
A9004: Too large register size.
F9005: This feature is not supported.

**(11) From Xa000**

Fa001: Illegal expression.
Fa002: Start address is bigger than the end address.
Fa003: Illegal source path.
Fa004: Too long expression.
Aa005: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.

Fa006: Illegal argument.
Fa007: Illegal program number.
Fa008: Source path is not set.
Fa009: File not found.
Fa00a: Can not open file. 1) The file is damaged or does not exist. Recreate the file.
Aa00b: Can not close file.
Aa00c: Failed in reading file. 1) The file is damaged or does not exist. Recreate the file.
Fa00d: Not source file of load module.
Fa00e: Illegal line number.
Fa00f: Variable does not exist.
Aa010: Can not communicate with ICE.
Fa011: Can not access register.
Fa012: Can not access memory.
Aa013: Reading of file went wrong.
Fa014: It was going to open the binary file.
Fa015: Can not get temporary path. 1) The disk is full. Delete or move unnecessary files and increase the available memory in the disk.
Fa016: Can not create temporary file. 1) The disk is full. Delete or move unnecessary files and increase the available memory in the disk.
Fa017: Can not remove temporary file.
Fa020: This feature is not supported.
Fa021: Symbol assigned to register cannot be specified.
Fa022: The character which cannot be used for the folder is contained or the folder does not exist.

**(12) From Xb000**

Fb000: Illegal command line.
Fb001: Program information does not exist in specified load module file.
Fb002: File not found.
Fb003: Function not found.
Fb004: Selected load module different from kind (Chip) was loaded.
Fb005: Symbol not found. 1) The address could not be found. Specify a location holding address information.
Fb008: Illegal expression.

Ab009: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
Fb00a: Illegal symbol in load module file.
Fb00b: Current program does not exist.
Fb00c: Current file does not exist.
Ab00d: Current function does not exist.
Ab00e: Current line does not exist.
Ab00f: Tag not found.
Ab010: Failed in loading symbol table.
Ab011: Illegal line number.
Fb012: Too large line number.
Ab015: Reading of file went wrong. 1) The file is damaged or does not exist. Recreate the file.
Ab016: Can not open file. 1) The file is damaged or does not exist. Recreate the file.
Ab017: Failed in writing file. 1) The file is damaged or does not exist. Recreate the file.
Ab019: Reading of file went wrong.
Ab01a: Can not close file.
Fb01b: Too long load module file name.
Ab01c: Too many entries of the task kind.
Fb01d: Address not found.
Wb01e: No debug information (not compiled in Debug Build mode).
Fb01f: Can not find structure member.
Fb020: Can not find value.
Fb021: No debug information exists in load module file. 1) To create a load module with appended debug information, execute build in build mode of Debug Build.
Fb022: Illegal line number.
Ab023: Current stack frame is not active.
Ab024: Different section.
Fb026: Too many array dimensions (> 4).
Fb027: Found end of file. 1) The specified file may be damaged. Recreate the file.
Fb028: This feature is not supported.
Fb029: Illegal address.
Ab02a: Can not communicate with ICE.

Fb02b: Can not stack trace with current PC value.
Fb02c: Too many blocks for one function.
Fb02d: Illegal argument.
Fb02e: The file does not exist in the SOURCE PATH. 1) On stopping the program, the source that the debugger tried to display could not be found. Check if the path connects to the source in the <a href="#">Debugger Option dialog box</a> , or check if the source is in the same folder as the out file. Refer to the <a href="#">Assemble window</a> on which the error message is displayed, and check if the corresponding path connects.
Fb02f: Information has been deleted because of optimization.
Ab030: Monitor timed out. 1) Check the power of the in-circuit emulator, cable connections, and setting of the interface board and restart the debugger.
Ab031: Already set in memory.
Ab032: Out of scope.
Ab033: LP is not stored.
Fb034: Return execution from present PC position cannot be performed.
Fb037: Too Many Line-Numbers Information.
Fb038: Compiler version mismatch. 1) Recreate the load module with the latest compiler.
Ab039: Failed in loading debug information.
Ab03a: No more section information.
Fb040: Specified file is not load module. 1) This is not a linker output file. Source debug cannot be executed with the load module before output from the linker. Specify the load module output from the linker.
Ab041: Too many files in load module to download.
Wb042: Symbol module is not initialized.
Fb32e: Illegal port number.
Fb32f: Illegal port name.
Fb330: Illegal port position.
Fb331: Illegal increment number.
Fb332: Port for memory bank is not set.
Fb333: Illegal bank number.
Fb334: Area for memory bank is not set.
Wb335: Too long symbol name.

**(13) From Xc000**

Fc001: Can not open file. 1) The file is damaged or does not exist. Recreate the file.
Ac002: Can not close file.
Ac003: Reading of file went wrong. 1) The file is damaged or does not exist. Recreate the file.
Ac004: Reading of file went wrong.
Fc005: Illegal file type.
Fc006: Kind (Chip) of load module is illegal.
Fc007: Specified file is not load module. 1) This is not a linker output file. Source debug cannot be executed with the load module before output from the linker. Specify the load module output from the linker.
Fc008: Specified load module file (COFF) is old version.
Ac009: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
Fc00a: No mapped address was accessed.
Fc00b: Load module is not loaded.
Fc00c: Illegal argument.
Fc00d: User program is running.
Fc00e: User program is being traced.
Fc00f: Interrupted.
Ac010: Can not communicate with ICE.
Fc011: Illegal load module file format.
Fc012: Check sum error.
Fc013: Too wide address range to upload (> 1M byte).
Fc014: Failed in writing file. 1) The file is damaged or does not exist. Recreate the file.
Fc015: Illegal program number.
Fc016: Load information is full.
Wc017: Symbol information is duplicated, please reset symbols.
Fc018: Specified file is not load module. 1) This is not a linker output file. Source debug cannot be executed with the load module before output from the linker. Specify the load module output from the linker.
Fc019: Failed in writing memory.

Wc01a: BSS area is assigned to non-mapped area. 1) When the program is executed, a non-map break may occur. Either allocate the BSS area to the internal RAM by using a link directive, or map the emulation memory or target memory to the BSS area using the <a href="#">Configuration dialog box</a> of the debugger.
Fc01b: Programmable-IOR address not specified. 1) Necessary files may be damaged. Reinstall the debugger.
Wc01c: Programmable IOR address mismatch. 1) Necessary files may be damaged. Reinstall the debugger.
Wc01d: Selected load module different from kind (Chip) was loaded.
Fc01e: Flash erase is not supported.
Fc100: This feature is not supported.

**(14) From Xd000**

Ad000: Error occurred inside debugger.
Ad001: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
Ad002: Failed in reading initialization file (expc.ini).
Ad003: ICE is not connected.
Fd004: Can not find Dynamic Link Library.

**(15) From Xe000**

Fe000: Illegal argument.
Fe001: Illegal start address.
Fe002: Illegal end address.
Fe003: Too large size.
Fe004: Can not open file. 1) The file is damaged or does not exist. Recreate the file.
Fe005: Failed in reading file. 1) The file is damaged or does not exist. Recreate the file.
Fe006: Reading of file went wrong.
Fe007: Failed in writing file. 1) The file is damaged or does not exist. Recreate the file.
Ae008: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
Fe009: Illegal file format.
Fe00a: Verification error.

Fe010: This feature is not supported.
---------------------------------------

**(16) From Xf000**

Af000: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
Ff000: Not enough memory.
Ff001: [XXX] not found.
Wf002: Not found [XXX]. Search from the beginning?
Wf003: Already exceed search region.
Ff004: Missing parameter.
Ff005: Illegal function name.
Ff006: Illegal number.
Ff007: Start address is bigger than end address.
Ff008: Illegal symbol or expression.
Ff009: [XXX] This file is illegal type.
Ff00a: User Program is Running.
Ff100: Disk cannot write or full.
Ff101: File not found.
Ff102: File not Create.
Ff103: Old project file version.
Ff104: Illegal project file format.
Ff105: This file is a project file for [XXX].Please select a correct file.
Wf106: CPU in the Project File was Changed. You must exit the debugger for the new CPU. Do you exit the Debugger?
Wf107: CPU in the Project File was Changed. Do you start the Debugger with this CPU?
Wf108: Selected project file different [YYY] from chip [XXX] was opened. Does it open, although the chip cannot be changed?
Wf109: Project Manager cannot be used with the debugger of this version. Please use PMplus.
Wf200: No difference encountered.
Ff201: Memory mapping error.
Ff202: Verify error. 1) External memory could not be accessed, as it is not set. Change the register values necessary for accessing the external memory using the <a href="#">SFR window</a> or <a href="#">Hook Procedure</a> before download .
Wf203: When a program is running, while rewriting, program execution stops for a moment. Do you wish to rewrite a memory?
Wf300: Would you like to save the changes made in [XXX]?
Ff301: The symbol being used on the event condition can't be evaluated.

Wf302: Delete: [XXX]
Wf303: [XXX] is edited. Delete: [YYY]?
Wf304: [XXX] is edited. Save: [YYY]?
Wf305: [XXX] is already exist. Do you replace it?
Ff306: This name is too long.
Ff307: There is the same name in other kinds.
Ff308: An address can't be omitted.
Ff309: Illegal address mask.
Ff30a: Illegal data mask.
Ff30b: Illegal ext probe mask.
Ff30c: Illegal ext probe data.
Ff30d: Illegal pass count.
Ff30e: Illegal register name.
Ff30f: Illegal register bank.
Ff310: Illegal delay count.
Wf311: Only one [XXX] can be enabled. Do you make this [YYY] to enable?
Ff312: [XXX] is already there.
Ff313: Event number already exist.
Ff314: Event name is not set.
Ff315: [XXX] is already there.
Ff316: Max number of enabled [XXX] event is over. Please disable other enabled [YYY] event.
Ff317: Max number of set event is over.
Ff31e: Illegal start address.
Ff31f: Illegal end address.
Ff322: Illegal count rate.
Ff323: Illegal time out break count.
Ff324: Section and Qualify can be specified at the same time.
Wf325: User program is running. Do you want to stop user program for a moment and set it?
Wf326: User program is running. Do you want to stop user program for a moment and delete it?
Ff350: There is a phase which event are not in the middle.
Ff351: The same event is contained in Link and Disable.
Ff352: An event isn't specified.
Ff357: AND event is in Phase.
Ff400: Coverage mapping error.
Wf401: Clear coverage?
Ff500: Illegal symbol.

Ff501: Illegal value.
Ff502: Illegal parameter.
Ff503: Max number of symbol is over.
<p>Ff504: This variable cannot be set as a break.</p> <p>1) Break cannot be set for the following variables.</p> <ul style="list-style-type: none"> <li>- Local variables, static variables</li> <li>- Array variables, member variables of structures/unions</li> <li>- Register/SFR</li> <li>- Variable expressions</li> </ul>
Wf600: Save project file?
<p>Wf601: When connecting the target system, please turn on the target system.</p> <p>1) When a target is not connected, simply click the [OK] button.</p>
Wf602: Please change a MODE mask condition or connect the target system.
<p>Ff603: Incorrect ID Code.</p> <p>1) This may be caused by the following. <b>[MINICUBE2]</b></p> <ul style="list-style-type: none"> <li>- The ID code is incorrect.</li> <li>-&gt; Input the correct ID code.</li> <li>- The internal flash memory is in the write mode because the FLMD0 pin is high.</li> <li>-&gt; Make the FLMD0 pin low.</li> <li>- The emulator connection prohibition mode is set because the ID code (bit 0,1 of address 0x84) is 0.</li> <li>-&gt;Erase the internal flash memory once by flash writer.</li> </ul>
<p>Af604: Incorrect ID Code. Abort the debugger.</p> <p>1) This may be caused by the following. <b>[MINICUBE2]</b></p> <ul style="list-style-type: none"> <li>- The ID code is incorrect.</li> <li>-&gt; Input the correct ID code.</li> <li>- The internal flash memory is in the write mode because the FLMD0 pin is high.</li> <li>-&gt; Make the FLMD0 pin low.</li> <li>- The emulator connection prohibition mode is set because the ID code (bit 0,1 of address 0x84) is 0.</li> <li>-&gt;Erase the internal flash memory once by flash writer.</li> </ul>
<p>Ff605: Please check connection with the target board.</p> <p>1) Check the connection of the target connector (TC). If a target is not connected, review the Target setting in the <a href="#">Configuration dialog box</a>.</p>
<p>Ff606: Please check connection with the target board, and power on it.</p> <p>1) Check the target power supply. If a target is not connected, review the Target setting in the <a href="#">Configuration dialog box</a>.</p>
<p>Wf607: Please check connection of the exchange adapter.</p> <p>1) Check the connection of the exchange adapter (EA).</p> <p>Recommend wearing of the exchange adapter, if the target is not connected.</p>

Ff608: Please disconnect the target board. 1) A current may flow from the internal power supply to the target. Disconnect the target connector (TC) from the conversion adapter (EA). Review the setting in the <a href="#">Configuration dialog box</a> if the target is not connected.
Ff609: Please power off the target board, and disconnect it.
Af60a: Incorrect ID Code. Flash memory was erased. Abort the debugger. 1) This message is displayed if ID authentication address 0x84 results in failure when the mode to erase the flash memory contents has been set, and the flash memory contents are erased (when MINICUBE2 is connected). <b>[MINICUBE2]</b>
Af60b: Disabled ID Code. Flash memory was erased. Abort the debugger. 1) This message is displayed if the target connection cable is disconnected when using the debugger in power off emulation mode, when the debugger is activated after the target power is turned off, or when the flash memory contents are erased. <b>[MINICUBE2]</b>
Af60c: During break Target was not turned on.
Wf60d: Because the Source Path had exceeded 4095 characters, it rounded it down.
Wf700: Do you want to download Load Module File?
Wf701: Do you load symbol information only?
Wf800: Configuration of Memory Bank is not set.
Wf801: BANK address must be in target memory.
Ff802: All events are deleted. because the use of external probe was changed.
Ff803: This event address is invalid on current configuration.
Ff804: Invalid PC value.
Ff805: Cannot set temporary break on this address.
Ff806: External data is being used by Debugger.
Ff900: Illegal I/O port name.
Ff901: Memory mapping error. 1) The specification of the address is illegal. Check the addresses that can be specified in the <a href="#">Add I/O Port dialog box</a> .
Ff902: Illegal access size.
Ff903: Illegal access type.
Ff904: There is the same name.
Wf905: [XXX] is already exist. Do you replace it?
Wf906: Would you like to register the change made in [XXX]?
Ffa00: The [XXX] function of current program on PC position not found. 1) The symbol specified in main() label: in the <a href="#">Debugger Option dialog box</a> could be found. Set a symbol of the main routine of the program. Default is _main.

<p>Ffa01: The line information on PC position not found.</p> <p>1) The source file corresponding to program counter (PC) value when the program was stopped could not be found. The following reasons are possible.</p> <ul style="list-style-type: none"> <li>-The source file exists in a location that the source path does not connect to.</li> <li>-The program stopped where the source files, such as library or RX, do not exist.</li> <li>-The program looped, jumped to an address that is not used by the program, and stopped there.</li> </ul>
<p>Wfb00: User program is running. Do you want to stop user program?</p> <p>1) [Yes] button is selected, execution of the user program is stopped and then the <a href="#">Exit Debugger dialog box</a> is displayed. If it is specified in the <a href="#">Debugger Option dialog box</a> that the Exit Debugger dialog box is not to be displayed, however, the ID78K0R-QB is terminated.</p> <p>[No] button is selected, execution of the user program is not stopped and the <a href="#">Exit Debugger dialog box</a> is not displayed. The ID78K0R-QB is not terminated.</p>
<p>Wfb01: Since bit 7 of address 0x79 in the ID code are 0, The N-Wire emulator becomes prohibition of use henceforth. Do you exit the debugger as it is?</p>
<p>Ffc00: Online help window cannot be started. Please install HTML Help environment with reference to a users manual.</p>
<p>Ffd00: Failed to specify [XXX].</p>
<p>Ffe00: The maximum size of RRM was exceeded.</p>
<p>Wfe01: There is a duplicate RRM address.</p>
<p>Wfe0b: It shift to the flash mode. Is it completely cleared but is the present event. Doesn't it care?</p>
<p>Ffff: Interrupted.</p>

# APPENDIX F INDEX

## A

About dialog box ... 270  
access monitor ... 181  
Access monitor function ... 52  
Active status and static status ... 78  
Add I/O Port dialog box ... 201  
Add Watch dialog box ... 169  
Address Move dialog box ... 155  
Assemble Search dialog box ... 153  
Assemble window ... 147

## B

break  
    breakpoint setting ... 44  
    setting break to variable ... 45  
Break dialog box ... 244  
break function ... 43  
Browse dialog box ... 273

## C

C0 coverage ... 62  
callback procedure ... 281  
Cautions before starting ... 25  
Change Watch dialog box ... 172  
character set ... 336  
Code Coverage ... 62  
Code Coverage window ... 227  
Come Here ... 48  
command ... 271  
Command reference ... 275  
Conditional trace ... 60  
Configuration dialog box ... 100  
Console window ... 271  
Contents saved to project file ... 75  
Context menu ... 99  
Coverage-Color dialog box  
    ... 232

## D

DataFlash Option dialog box ... 118, 232  
Debug function list ... 36  
Debugger Option dialog box ... 120  
Delay Count dialog box ... 230  
Delay trigger trace ... 61  
DMM dialog box ... 190  
download ... 38  
Download dialog box ... 130  
drag & drop function ... 81

## E

Environment Setting File Load dialog box ... 272  
Environment Setting File Save dialog box ... 267  
Error messages at start up ... 30

Errors ... 346  
Event dialog box ... 257  
Event DMM condition ... 73  
Event DMM dialog box ... 253  
Event function ... 65  
Event icon ... 68  
Event Link dialog box ... 241  
Event manager ... 236  
Event manages ... 68  
Event Setting Status (Event Mark) ... 138  
Exit Debugger dialog box ... 269  
Expressions ... 339  
Extended Option dialog box ... 107

## F

Fail-safe break ... 101  
Fail-safe Break dialog box ... 112  
Flash Option dialog box ... 114

## G

Grep window ... 324

## H

hook procedure ... 282  
Hook window ... 325

## I

ID code ... 23  
In-circuit emulator ... 22  
input conventions ... 336  
Installing ... 24  
ioport ... 300

## J

jump function ... 79

## L

List window ... 158, 323  
Load Module List dialog box ... 135  
Load/Save Function ... 75  
Local Variable window ... 174

## M

Main window ... 84  
Mask ... 104  
Memory Compare dialog box ... 188  
Memory Compare Result dialog box ... 189  
Memory Copy dialog box ... 187  
Memory Fill dialog box ... 186  
Memory manipulation function ... 52

Memory Search dialog box ... 184  
 Memory window ... 180  
 Menubar ... 85  
 Messages ... 346  
 Mixed display mode  
   Source window ... 41  
   Trace View window ... 59

**N**

-ng option ... 23

**O**

OCD Checker ... 25  
 on-chip debug option byte ... 23  
 OpenBreak window ... 328  
 Operating Environment ... 22  
 Operators ... 339

**P**

pick up ... 221  
 PM+ ... 32  
 Point mark area ... 138, 148  
 program code ... 140  
 Program execution function ... 47  
 project file ... 75, 269  
 Project File Load dialog box ... 129  
 Project File Savedialog box ... 128  
 Pseudo Emulation dialog box ... 127

**Q**

Qualify trace ... 61  
 Quantitative upper limit list ... 342  
 Quick Watch dialog box ... 167

**R**

Range of Radixes ... 341  
 Register Manipulation Function ... 54  
 Register window ... 192  
 reset ... 268  
 Reset Debugger dialog box ... 268  
 right click menu ... 99  
 Run-Break event ... 57

**S**

Section trace ... 61  
 Setting debugging environment ... 37  
 setting file ... 77  
 Setting mapping ... 37  
 SFR Select dialog box ... 199  
 SFR window ... 195  
 Snap Shot dialog box ... 248  
 Software Break manager ... 234  
 Source Search dialog box ... 143  
 Source Text Move dialog box ... 145  
 Source window ... 137  
 Stack trace display function ... 51  
 Stack window ... 177

Start From Here ... 48  
 Startup option ... 25  
 Startup Routine ... 124  
 Statusbar ... 93  
 Sym Inspect window ... 327  
 Symbol To Address dialog box ... 156

**T**

## Tcl

assemble ... 286  
 batch ... 287  
 breakpoint ... 288  
 dbgexit ... 290  
 download ... 291  
 erase ... 292  
 extwin ... 293  
 finish ... 294  
 go ... 295  
 help ... 296  
 hook ... 297  
 ie ... 298  
 inspect ... 299  
 jump ... 301  
 map ... 302  
 mdi ... 303  
 memory ... 304  
 module ... 305  
 next ... 306  
 refresh ... 307  
 register ... 308  
 reset ... 309  
 run ... 310  
 step ... 311  
 stop ... 312  
 tkcon ... 313  
 upload ... 314  
 version ... 315  
 watch ... 316  
 where ... 317  
 wish ... 318  
 xcoverage ... 319  
 xtime ... 320  
 xtrace ... 321  
 Tcl command list ... 276  
 Timer dialog box ... 203  
 Timer function ... 56  
 Timer Result dialog box ... 218  
 Trace Data Select dialog box ... 220  
 Trace dialog box ... 224  
 Trace function ... 58  
 trace memory ... 58  
 Trace Move dialog box ... 222  
 Trace result with linking window ... 80  
 Trace Search dialog box ... 214  
 Trace View window ... 207  
 Tracer control mode ... 60  
 Types of Messages ... 347

**U**

Unconditional trace ... 60  
 Uninstalling ... 24

upload ... 39  
Upload dialog box ... 133

**V**

verify check ... 111  
view file ... 76  
View File Load dialog box ... 265  
View File Save dialog box ... 263

**W**

Watch function ... 49  
Watch window ... 161  
window list ... 96  
window reference ... 96

<b>REVISION HISTORY</b>	<b>ID78K0R-QB Ver.3.61 User's Manual: Development Tool (Operation)</b>
-------------------------	----------------------------------------------------------------------------

<b>Rev.</b>	<b>Date</b>	<b>Description</b>	
		<b>Page</b>	<b>Summary</b>
1.00	Apr, 2010	—	First Edition issued

---

ID78K0R-QB Ver.3.61 User's Manual: Development Tool  
(Operation)

Publication Date: Rev.1.00 Apr, 2010

Published by: Renesas Electronics Corporation

---

**SALES OFFICES****Renesas Electronics Corporation**<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.**Renesas Electronics America Inc.**2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130**Renesas Electronics Canada Limited**1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220**Renesas Electronics Europe Limited**Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: +44-1628-585-100, Fax: +44-1628-585-900**Renesas Electronics Europe GmbH**Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-65030, Fax: +49-211-6503-1327**Renesas Electronics (China) Co., Ltd.**7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679**Renesas Electronics (Shanghai) Co., Ltd.**Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China  
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898**Renesas Electronics Hong Kong Limited**Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2886-9318, Fax: +852 2886-9022/9044**Renesas Electronics Taiwan Co., Ltd.**7F, No. 363 Fu Shing North Road Taipei, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670**Renesas Electronics Singapore Pte. Ltd.**1 HarbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: +65-6213-0200, Fax: +65-6278-8001**Renesas Electronics Malaysia Sdn.Bhd.**Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510**Renesas Electronics Korea Co., Ltd.**11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

ID78K0R-QB Ver.3.61

