

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

## 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.



# User's Manual

## **ID78K0S-QB Ver. 2.90**

### **Integrated Debugger**

### **Operation**

---

### **Target Device** **78K0S Microcontrollers**

Document No. U18247EJ1V0UM00 (1st edition)

Date Published August 2006 CP(K)

© NEC Electronics Corporation 2006

Printed in Japan

[MEMO]

**IECUBE is a registered trademark of NEC Electronics Corporation in Japan and Germany.**

**MINICUBE is a registered trademark of NEC Electronics Corporation in Japan and Germany or a trademark in the United States of America.**

**Windows are either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.**

- **The information in this document is current as of August, 2006. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC 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 NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".  
The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.  
"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.  
"Special": 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, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

- (1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
- (2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

[MEMO]

[MEMO]



## PREFACE

<b>Target Readers</b>	This manual is intended for user engineers who design and develop application systems of the 78K0S microcontrollers.	
<b>Purpose</b>	This manual is intended for users to understand the functions of the ID78K0S-QB in the organization below.	
<b>Organization</b>	This manual consists of the following chapters: <ul style="list-style-type: none"><li>• OVERVIEW</li><li>• INSTALLATION</li><li>• STARTING AND TERMINATING</li><li>• ASSOCIATION WITH PM+</li><li>• DEBUG FUNCTION</li><li>• WINDOW REFERENCE</li><li>• COMMAND REFERENCE</li></ul>	
<b>How to Use This Manual</b>	<p>It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, microcontrollers, C language, and assemblers.</p> <p>To understand the functions of the 78K0S microcontrollers → Refer to Hardware User's Manual for each product.</p> <p>To understand the instruction functions of the 78K0S microcontrollers → Refer to 78K/0S Series Instructions User's Manual (U11047E).</p>	
<b>Conventions</b>	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-78K0SKX1 In-Circuit Emulator		U18219E
QB-78K0SKX1MINI In-Circuit Emulator		U17307E
RA78K0S Assembler Package	Operation	U16656E
	Language	U14877E
	Structured Assembly Language	U11623E
CC78K0S C Compiler	Operation	U16654E
	Language	U14872E
ID78K0S-QB Ver. 2.90 Integrated Debugger	Operation	This manual
PM plus Ver.5.10		U16569E

# CONTENTS

CHAPTER 1 OVERVIEW ...	16
1.1 Features ...	17
1.2 System Configuration ...	18
1.3 Operating Environment ...	20
1.3.1 Hardware environment ...	20
1.3.2 Software environment ...	20
1.4 Cautions During Debugging ...	21
1.4.1 When performing source level debugging ...	21
CHAPTER 2 INSTALLATION ...	22
2.1 Installing ...	22
2.2 Uninstalling ...	22
CHAPTER 3 STARTING AND TERMINATING ...	23
3.1 Cautions Before Starting <b>[MINICUBE2]</b> ...	23
3.2 Startup Option and Argument Specification ...	24
3.2.1 Specification method ...	24
3.2.2 Specification format and options ...	25
3.3 Starting ...	26
3.4 Terminating ...	27
3.5 Error Messages at Start up ...	28
CHAPTER 4 ASSOCIATION WITH PM+ ...	29
4.1 Setting Build Mode ...	30
4.2 Registering Debugger to PM+ Project ...	30
4.2.1 Selecting debugger ...	30
4.3 To Start ID78K0S-QB from PM+ ...	31
4.3.1 Restoring debugging environment ...	31
4.4 Auto Load ...	32
4.4.1 Auto load by correcting source code ...	32
4.4.2 Auto load by starting debugger ...	33
CHAPTER 5 DEBUG FUNCTION ...	34
5.1 Setting Debugging Environment ...	35
5.1.1 Setting operating environment ...	35
5.1.2 Setting option ...	35
5.2 Download Function, Upload Function ...	36
5.2.1 Download ...	36
5.2.2 Upload ...	36
5.3 Source Display, Disassemble Display Function ...	37
5.3.1 Source display ...	37
5.3.2 Disassemble display ...	37
5.3.3 Mixed display mode (Source Window) ...	38
5.3.4 Convert symbol (symbol to address) ...	39
5.4 Break Function ...	40
5.4.1 Break types ...	40
5.4.2 Breakpoint setting ...	41
5.4.3 Setting breaks to variables <b>[IECUBE]</b> <b>[MINICUBE+]</b> ...	42
5.4.4 Hardware break and software break <b>[IECUBE]</b> <b>[MINICUBE+]</b> ...	43
5.5 Program Execution Function ...	44
5.6 Watch Function ...	46
5.6.1 Displaying and changing data values ...	46
5.6.2 Displaying and changing local variable values ...	47
5.6.3 Registering and deleting watch data ...	47
5.6.4 Changing watch data ...	48
5.6.5 Temporarily displaying and changing data values ...	48
5.6.6 Callout watch function ...	49

5.6.7	Stack trace display function ...	49
5.7	Memory Manipulation Function ...	50
5.7.1	Displaying and changing memory contents ...	50
5.7.2	Filling, copying, and comparing memory contents ...	50
5.7.3	Flash memory writing function <b>[MINICUBE2]</b> ...	51
5.8	Register Manipulation Function ...	52
5.8.1	Displaying and changing register contents ...	52
5.8.2	Displaying and changing SFR contents ...	53
5.8.3	Displaying and changing I/O port contents ...	53
5.9	Trace Function <b>[IECUBE]</b> ...	54
5.9.1	Checking trace data ...	54
5.9.2	Mixed display mode (Trace Window) ...	55
5.10	Event Function <b>[IECUBE]</b> <b>[MINICUBE+]</b> ...	56
5.10.1	Using event function ...	56
5.10.2	Creating events ...	56
5.10.3	Setting various event conditions ...	57
5.10.4	Number of enabled events for each event condition ...	58
5.10.5	Managing events ...	59
5.11	Pseudo Real-time Monitor Function (Break When Readout) <b>[IECUBE]</b> <b>[MINICUBE+]</b> ...	60
5.11.1	Sampling range setting ...	60
5.11.2	Sampling time setting ...	61
5.12	Load/Save Function ...	62
5.12.1	Debugging environment (project file) ...	62
5.12.2	Window display information (view file) ...	63
5.12.3	Window setting information (setting file) ...	64
5.13	Functions Common to Each Window ...	65
5.13.1	Active status and static status ...	65
5.13.2	Jump function ...	66
5.13.3	Drag & drop function ...	68
5.13.4	Trace result with linking window <b>[IECUBE]</b> ...	69
5.13.5	Cautions ...	70
CHAPTER 6 WINDOW REFERENCE ...		71
6.1	Window List ...	72
6.2	Explanation of Windows ...	74
	Main Window ...	75
	Configuration Dialog Box ...	86
	Extended Option Dialog Box ...	90
	RRM Dialog Box ...	92
	Debugger Option Dialog Box ...	95
	Project File Save Dialog Box ...	101
	Project File Load Dialog Box ...	103
	Download Dialog Box ...	105
	Upload Dialog Box ...	108
	Source Window ...	110
	Source Search Dialog Box ...	115
	Source Text Move Dialog Box ...	117
	Assemble Window ...	119
	Assemble Search Dialog Box ...	123
	Address Move Dialog Box ...	125
	Symbol To Address Dialog Box ...	126
	Watch Window ...	128
	Quick Watch Dialog Box ...	133
	Add Watch Dialog Box ...	136
	Change Watch Dialog Box ...	139
	Local Variable Window ...	141
	Stack Window ...	143
	Memory Window ...	146
	Memory Search Dialog Box ...	149
	Memory Fill Dialog Box ...	151
	Memory Copy Dialog Box ...	153
	Memory Compare Dialog Box ...	155
	Memory Compare Result Dialog Box ...	157

Register Window ...	158
SFR Window ...	161
SFR Select Dialog Box ...	165
Add I/O Port Dialog Box ...	167
Trace View Window ...	169
Trace Data Select Dialog Box ...	173
Trace Move Dialog Box ...	175
Software Break Manager ...	177
Event Manager ...	180
Event Dialog Box ...	185
Break Dialog Box ...	189
View File Save Dialog Box ...	191
View File Load Dialog Box ...	194
Environment Setting File Save Dialog Box ...	196
Environment Setting File Load Dialog Box ...	198
Reset Debugger Dialog Box ...	200
Exit Debugger Dialog Box ...	201
About Dialog Box ...	202
Console Window ...	204
Font Dialog Box ...	205
Browse Dialog Box ...	207
 CHAPTER 7 COMMAND REFERENCE ...	 209
7.1 Command Line Rules ...	210
7.2 Command List ...	210
7.3 List of Aliases ...	212
7.4 List of Variables ...	212
7.5 List of Packages ...	213
7.6 Key Bind ...	213
7.7 Expansion Window ...	214
7.7.1 Samples (Calculator Script) ...	214
7.8 Callback Procedure ...	215
7.9 Hook Procedure ...	216
7.10 Related Files ...	217
7.11 Cautions ...	217
7.12 Explanation of Commands ...	217
address ...	218
assemble ...	219
batch ...	220
breakpoint ...	221
dbgexit ...	223
download ...	224
erase ...	225
extwin ...	226
finish ...	227
go ...	228
help ...	229
hook ...	230
inspect ...	231
jump ...	232
map ...	233
mdi ...	234
memory ...	235
module ...	236
next ...	237
refresh ...	238
register ...	239
reset ...	240
run ...	241
step ...	242
stop ...	243
upload ...	244
version ...	245

- watch ... 246
- where ... 247
- wish ... 248
- xtime ... 249
- xtrace ... 250
- tkcon ... 251

## APPENDIX A EXPANSION WINDOW ... 252

- A.1 Overview ... 252
- A.2 Sample List of Expansion Window ... 252
- A.3 Activation ... 252
- A.4 Explanation of Each Sample Window ... 252
  - List Window ... 253
  - Grep Window ... 254
  - Hook Window ... 255

## APPENDIX B INPUT CONVENTIONS ... 257

- B.1 Usable Character Set ... 257
- B.2 Symbols ... 258
- B.3 Numeric Values ... 259
- B.4 Expressions and Operators ... 259
- B.5 File Names ... 262

## APPENDIX C KEY FUNCTION LIST ... 263

## APPENDIX D MESSAGES ... 266

- D.1 Display Format ... 266
- D.2 Types of Messages ... 266
- D.3 Message Lists ... 267

## APPENDIX E INDEX ... 289

# LIST OF FIGURES

Figure No	Title, Page
1-1	ID78K0S-QB ... 16
1-2	Example of ID78K0S-QB System Configuration [IECUBE] ... 18
1-3	Example of ID78K0S-QB System Configuration [MINICUBE+] ... 18
1-4	Example of ID78K0S-QB System Configuration [MINICUBE2] ... 19
3-1	Startup Option (Example) ... 24
3-2	Configuration Dialog Box ... 26
3-3	Main Window (at Startup) ... 27
3-4	Exit Debugger Dialog Box ... 27
5-1	Breakpoint Setting ... 41
5-2	Setting Break to Variable ... 42
5-3	Management of Software Breaks ... 43
5-4	Execution Button ... 44
5-5	[Run] Menu ... 44
5-6	Watch Window ... 46
5-7	Specification of the Display Format (Debugger Option Dialog Box) ... 46
5-8	Local Variable Window ... 47
5-9	Change Watch Dialog Box ... 48
5-10	Quick Watch Dialog Box ... 48
5-11	Stack Window ... 49
5-12	Displaying and Changing Memory Contents ... 50
5-13	Absolute Name/Function Name Switching ... 52
5-14	Displaying SFR Contents ... 53
5-15	Register I/O Port ... 53
5-16	Checking Trace Data ... 54
5-17	Setting of Various Event Conditions ... 57
5-18	Managing Events (Event Manager) ... 59
5-19	Sampling Range Setting (RRM Dialog Box) ... 60
5-20	Sampling Time Setting ... 61
6-1	Main Window ... 75
6-2	Toolbar (Picture Only) ... 83
6-3	Toolbar (Picture and Text) ... 83
6-4	Status Bar ... 84
6-5	Configuration Dialog Box ... 86
6-6	Extended Option Dialog Box ... 90
6-7	RRM Dialog Box ... 92
6-8	Debugger Option Dialog Box ... 95
6-9	Add Source Path Dialog Box ... 96
6-10	Project File Save Dialog Box ... 101
6-11	Project File Load Dialog Box ... 103
6-12	Download Dialog Box ... 105
6-13	The Progress of Download ... 105
6-14	Upload Dialog Box ... 108
6-15	Source Window ... 110
6-16	Source Search Dialog Box ... 115
6-17	Source Text Move Dialog Box ... 117
6-18	Assemble Window ... 119
6-19	Assemble Search Dialog Box ... 123
6-20	Address Move Dialog Box (Example: When Memory Window Is Open) ... 125
6-21	Symbol To Address Dialog Box ... 126
6-22	Watch Window ... 128
6-23	Quick Watch Dialog Box ... 133
6-24	Add Watch Dialog Box ... 136
6-25	Change Watch Dialog Box ... 139
6-26	Local Variable Window ... 141
6-27	Stack Window ... 143

6-28	Memory Window ...	146
6-29	Memory Search Dialog Box ...	149
6-30	Memory Fill Dialog Box ...	151
6-31	Memory Copy Dialog Box ...	153
6-32	Memory Compare Dialog Box ...	155
6-33	Memory Compare Result Dialog Box ...	157
6-34	Register Window ...	158
6-35	SFR Window ...	161
6-36	SFR Select Dialog Box ...	165
6-37	Add I/O Port Dialog Box ...	167
6-38	Trace View Window ...	169
6-39	Trace Data Select Dialog Box ...	173
6-40	Trace Move Dialog Box ...	175
6-41	Software Break Manager ...	177
6-42	Event Manager (In Detailed Display Mode) ...	180
6-43	Select Display Information Dialog Box ...	184
6-44	Event Dialog Box ...	185
6-45	Break Dialog Box ...	189
6-46	View File Save Dialog Box ...	191
6-47	View File Load Dialog Box ...	194
6-48	Environment Setting File Save Dialog Box ...	196
6-49	Environment Setting File Load Dialog Box ...	198
6-50	Reset Debugger Dialog Box ...	200
6-51	Exit Debugger Dialog Box ...	201
6-52	About Dialog Box ...	202
6-53	Console Window ...	204
6-54	Font Dialog Box ...	205
6-55	Browse Dialog Box ...	207
7-1	Execution Screen ...	214
A-1	List Window ...	253
A-2	Grep Window ...	254
A-3	Hook Window ...	255
D-1	Error/Warning Dialog Box ...	266



# LIST OF TABLES

Table No	Title, Page
2-1	Install ... 22
3-1	Startup Options ... 25
5-1	Debug Function List (Flow of Debugging Operations) ... 34
5-2	Type of File That Can Be Downloaded ... 36
5-3	Type of File That Can Be Uploaded ... 36
5-4	File Type Can Be Displayed ... 37
5-5	Specifying Symbols ... 39
5-6	Break Types ... 40
5-7	The Number of Valid Software Break ... 43
5-8	Type of Execution ... 44
5-9	Absolute Name to Function Name Correspondence ... 52
5-10	Various Event Conditions ... 56
5-11	Number of Enabled Events for Each Event Condition ... 58
5-12	Event Icon ... 59
5-13	Areas for Which Sampling Can Be Performed with Pseudo Real-time RAM Monitor Function ... 60
5-14	Contents Saved to Project File ... 62
5-15	Type of View Files ... 63
5-16	Type of Setting Files ... 64
5-17	Details of Jump Source Address ... 67
5-18	Details of Drag & Drop Function (Line/Address) ... 68
5-19	Details of Drag & Drop Function (Character String) ... 68
6-1	Window List ... 72
6-2	CPU Status ... 84
6-3	IE Status ... 85
6-4	Break Cause ... 85
6-5	Event Setting Status (Event Mark) ... 111
6-6	Watch Window Display Format (Symbol) ... 129
6-7	Watch Window Display Format (Data) ... 130
6-8	Watch Window Input Format ... 137
6-9	How a Variable Is Handled When a Scope Is Specified ... 137
6-10	Frame Number Specification Format ... 176
6-11	Separator for Displaying Event Details ... 181
6-12	Status Condition ... 186
6-13	Settable Range of Data Condition ... 188
6-14	Number of Events Settable in Break Condition Setting Area ... 190
7-1	List of Debugger Control Commands ... 210
7-2	List of Console/Tcl Commands ... 211
7-3	Contents of File aliases.tcl ... 212
7-4	Message ID ... 215
A-1	List of Expansion Window (Sample) ... 252
B-1	List of Character Set ... 257
B-2	List of Special Characters ... 257
B-3	Input Format of Numeric Values ... 259
B-4	List of Operators ... 260
B-5	Operator Priority ... 261
B-6	Range of Radixes ... 261
C-1	Key Function List ... 263
D-1	Types of Messages ... 266

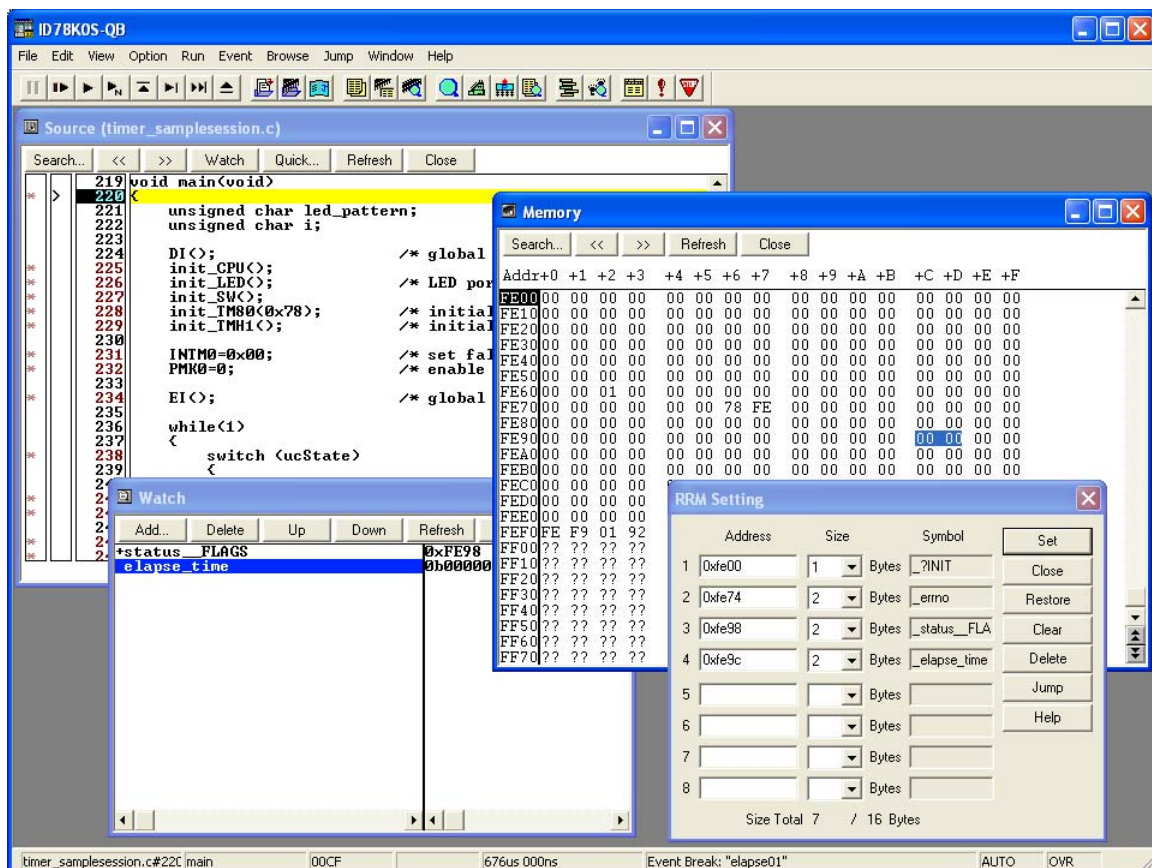
# CHAPTER 1 OVERVIEW

The Integrated Debugger ID78K0S-QB for the 78K0S microcontrollers (hereafter referred to as the ID78K0S-QB) is a software tool developed for NEC Electronics 78K0S microcontrollers 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 ID78K0S-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 MINICUBE™+ is connected ( [MINICUBE+] )
- When the MINICUBE2 is connected ( [MINICUBE2] )

Figure 1-1 ID78K0S-QB



This chapter explains the following items regarding the ID78K0S-QB.

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

## 1.1 Features

The ID78K0S-QB has the following features:

### (1) Source debugging

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

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

By using the event setting function of an in-circuit emulator, break events can be set (refer to ["5.10 Event Function \[IECUBE\] \[MINICUBE+\]"](#)).

### (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 ["5.12 Load/Save Function"](#)).

### (5) 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"](#)).

### (6) Support of on-chip debugging [MINICUBE2]

A debugging function implemented by the on-chip debug program (monitor program) is provided.

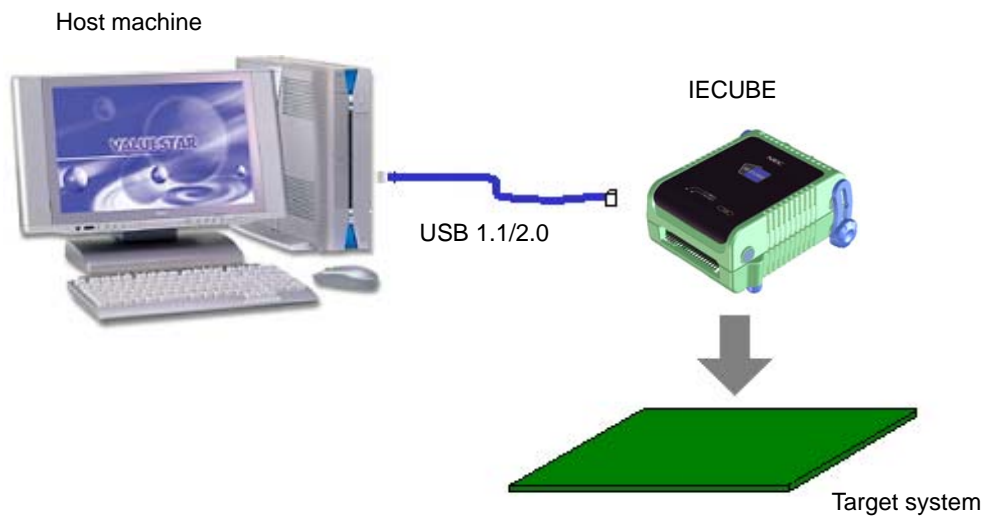
## 1.2 System Configuration

The ID78K0S-QB provides optimal debugging environments based on the following system configuration for user programs and target systems that were developed for the 78K0S microcontrollers.

### (1) IECUBE : QB-78K0SKX1 (In-circuit emulator)

The IECUBE can be manipulated from the ID78K0S-QB by connecting it to the host machine via a USB cable.

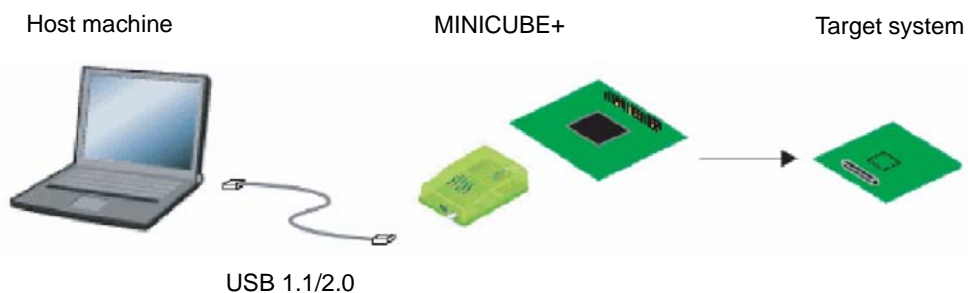
Figure 1-2 Example of ID78K0S-QB System Configuration [IECUBE]



### (2) MINICUBE+ : QB-78K0SKX1MINI (In-circuit emulator)

The MINICUBE+ can be manipulated from the ID78K0S-QB by connecting it to the host machine via a USB cable.

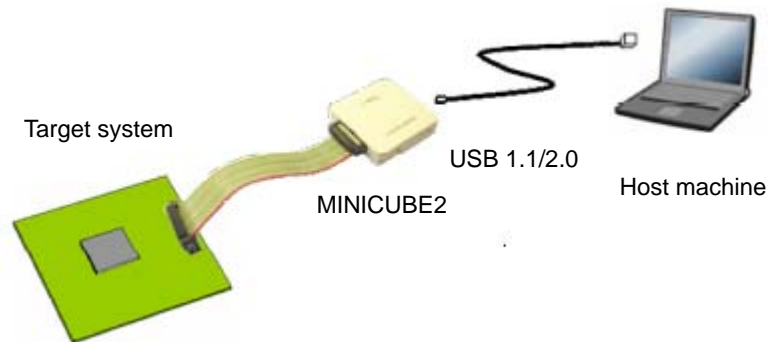
Figure 1-3 Example of ID78K0S-QB System Configuration [MINICUBE+]



**(3) MINICUBE2 : QB-MINI2 (on-chip debug emulator with programming function)**

The MINICUBE2 can be manipulated from the ID78K0S-QB by connecting it to the host machine via a USB cable.

Figure 1-4 Example of ID78K0S-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

#### (2) In-circuit emulator

IECUBE : QB-78K0SKX1

MINICUBE+ : QB-78K0SKX1MINI (QB-78K0SMINI + QB-78K0SKX1-DA)

MINICUBE2 : QB-MINI2

### 1.3.2 Software environment

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

Windows®98, Windows 2000, Windows Me, Windows XP Home Edition, Windows XP Professional

**Caution** Regardless of which of the OS above is used, we recommend that the latest Service Pack is installed.

#### (2) Device file (Individual acquisition)

The device file of the target device to be used.

This file is available from the following Web site of NEC Electronics.(ODS)

<http://www.necel.com/micro/ods/eng/>

## 1.4 Cautions During Debugging

The cautions to be observed during debugging are described below.

- [When performing source level debugging](#)

### 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.

#### (1) When using PM+

Specify [Debug Build] when the Build mode is selected.

#### (2) When using LK78K0S on standalone basis

Add the -g option.

# CHAPTER 2 INSTALLATION

This chapter explains the following items about installation of ID78K0S-QB:

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

## 2.1 Installing

The following items must be installed, when the ID78K0S-QB is used.

Table 2-1 Install

Item	Procedure
ID78K0S-QB system disk	Install the contents of this disk according to the automatically executed installer.
Used device file	Install this file according to dedicated startup installer by selecting [Start] menu -> [Program] -> [NEC Tools32] -> [Device File Installer].

**Caution** To install the ID78K0S-QB again after the ID78K0S-QB has been installed once, be sure to uninstall the ID78K0S-QB. If the ID78K0S-QB is installed in a directory different from that, without uninstalling, the ID78K0S-QB that has already been installed cannot be uninstalled.

## 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 ID78K0S-QB:

- [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 ID78K0S-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 ID78K0S-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 ID78K0S-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.

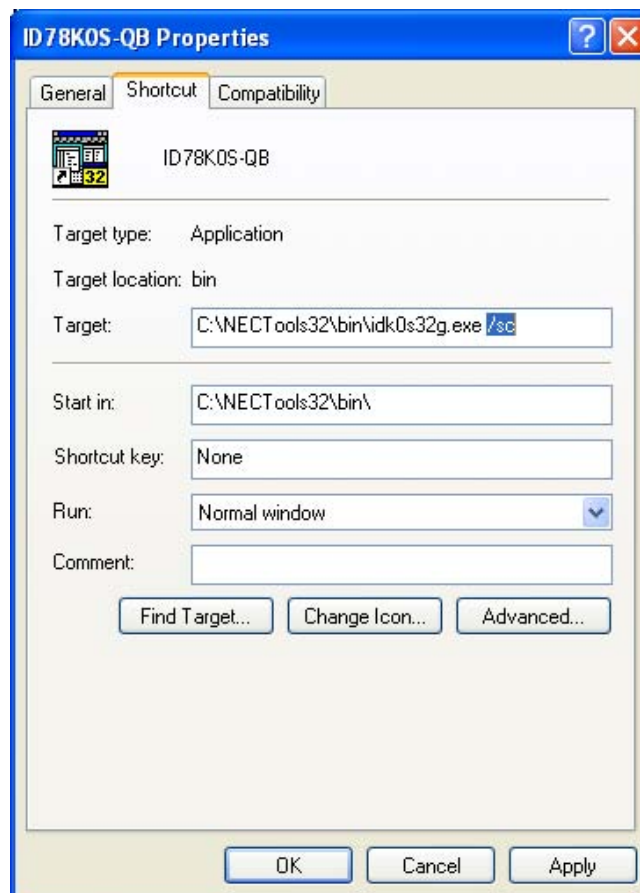
### 3.2.1 Specification method

- 1) Create an ID78K0S-QB shortcut on the desktop.

The ID78K0S-QB execution file (\*.exe) 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

```
idk0s32g.exe ?options?
idk0s32g.exe ?options? project
```

Each option and argument is separated by a space. No distinction is made between uppercase and lowercase 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 " [Example 3\) 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.
/SCRIPT: <i>script file name</i>	Specify the script file to be executed at startup.

### (3) Specification example

#### Example 1) Specification of script file only

```
idk0s32g.exe /script:c:\work\script.tcl
```

#### Example 2) Specification of script file and project file

```
idk0s32g.exe /script:c:\work\script.tcl c:\work\project.prj
```

#### Example 3) Specification when there are spaces in the path

```
idk0s32g.exe /script:"c:\work folder\script.tcl" "c:\work folder\project.prj"
```

### 3.3 Starting

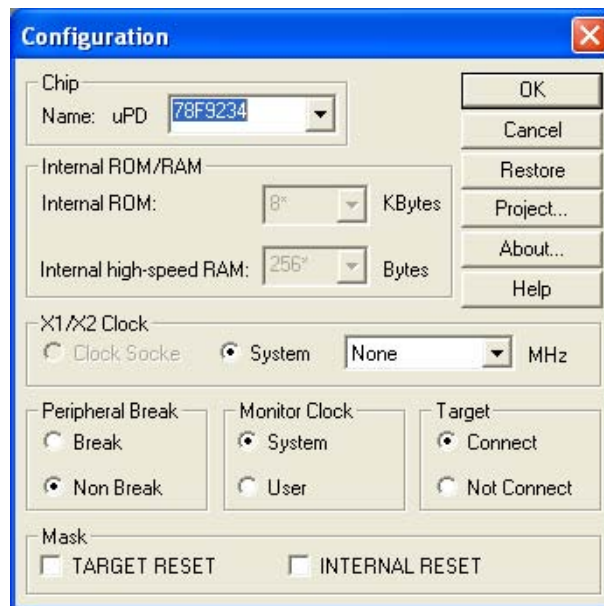
- 1) Start ID78K0S-QB from the [Start] menu of PM+ or by clicking the shortcut created on the desktop.

Refer to ["4.3 To Start ID78K0S-QB from PM+"](#) when starting from PM+.

Start the ID78K0S-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"](#).

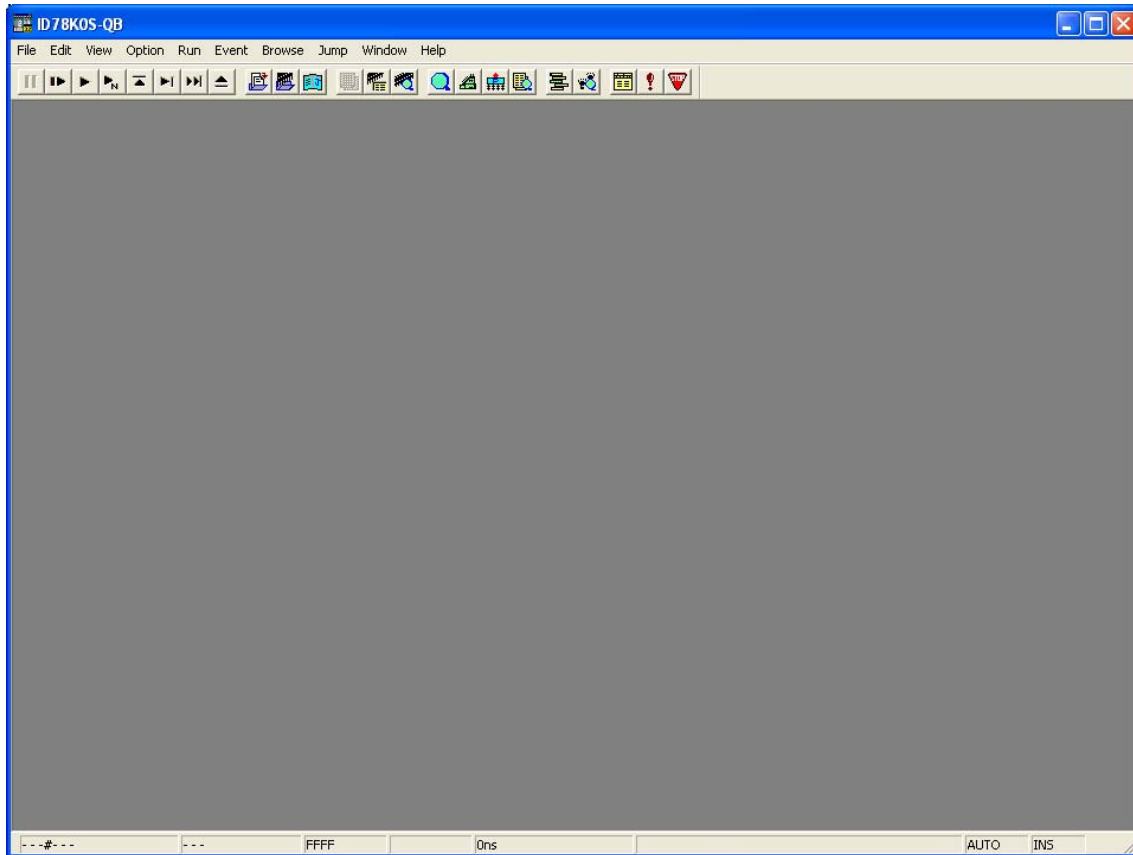
Figure 3-2 Configuration Dialog Box



- 2) Set the items related to the operating environment of the ID78K0S-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 ID78K0S-QB can be operated. Mainly use this window for debugging.

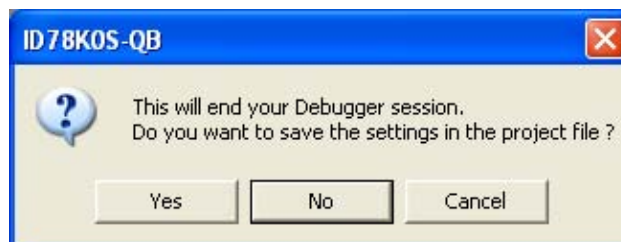
Figure 3-3 Main Window (at Startup)



## 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



- 2) 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 ID78K0S-QB terminated.

### 3.5 Error Messages at Start up

Error messages that may be output when ID78K0S-QB starts up are listed below (by order of occurrence). When these messages are output, refer to "APPENDIX D MESSAGES".

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 (d9xxx.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. <b>[MINICUBE2]</b>
F0c48: Flash programming is not permitted by the debugger setting. <b>[MINICUBE2]</b>

# CHAPTER 4 ASSOCIATION WITH PM+

The ID78K0S-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 ID78K0S-QB from PM+](#)
- [Auto Load](#)

**Caution** If a load module file is created by using the Windows command prompt, the function to associate the ID78K0S-QB with the PM+ cannot be used.

## 4.1 Setting Build Mode

To debug the load module file created by the PM+ on the ID78K0S-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 ID78K0S-QB is registered as the debugger of the active project. The ID78K0S-QB icon is displayed on the tool bar 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 ID78K0S-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 ID78K0S-QB and click the <OK> button in this dialog box. For details of the setting, refer to the User's manual.

**Caution** The contents of the internal flash memory are erased all before download (only a product with internal flash memory) . **[MINICUBE2]**



## 4.3 To Start ID78K0S-QB from PM+

The ID78K0S-QB can be started from the PM+ as follows:

- Click the ID78K0S-QB starting button on the tool bar 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 ID78K0S-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 ID78K0S-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.

### 4.3.1 Restoring debugging environment

The previous debugging environment can be restored by the following procedure when the ID78K0S-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 ID78K0S-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 ID78K0S-QB is started.
- 3) Download the load module file to be debugged with the [Download Dialog Box](#) of the ID78K0S-QB.
- 4) Debug the load module file on the ID78K0S-QB.
- 5) Click the <Yes> button on the [Exit Debugger Dialog Box](#) when the ID78K0S-QB is terminated.  
-> The debugging environment will be saved to the project file (sample.prj) for the PM+ when the ID78K0S-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 ID78K0S-QB debugging).
- 6) When the ID78K0S-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 ID78K0S-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 ID78K0S-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 ID78K0S-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 ID78K0S-QB by compiling and linking the file on the PM+ with the activated ID78K0S-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 ID78K0S-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 ID78K0S-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 ID78K0S-QB started, the load module will be automatically downloaded to the ID78K0S-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 ID78K0S-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 trace data	<a href="#">5.9 Trace Function [IECUBE]</a>
To manage the events	<a href="#">5.10 Event Function [IECUBE] [MINICUBE+]</a>
RRM function	<a href="#">5.11 Pseudo Real-time Monitor Function (Break When Readout) [IECUBE] [MINICUBE+]</a>
To save the debug environment and window status	<a href="#">5.12 Load/Save Function</a>
Jump function and cautions	<a href="#">5.13 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](#)

### 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 ID78K0S-QB starts up.

If a project file already exists, the debugging environment can be restored by clicking the <Project...> button. (Refer to "[5.12.1 Debugging environment \(project file\)](#)".)

### 5.1.2 Setting option

The extended options and debug options are set in the [Extended Option Dialog Box](#), the [Debugger Option Dialog Box](#) .

## 5.2 Download Function, Upload Function

ID78K0S-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](#)

### 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.

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)	Hex Format (*.hex)
Motorola Hex format S type (S0, S2, S3, S5, S7, S8)	
Extended Tektronix Hex format	
Binary data	Binary Data (*.bin)

**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.

Table 5-3 Type of File That Can Be Uploaded

Extension	Format
Intel Hex (*.hex)	Intel Hex format Extension (16-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

**Remark** One of the format 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 \(symbol to address\)](#)

### 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

```

* 225 init_CPU();
* 226 init_LED();          /* LED port initialization */
* 227 init_SW();
* 228 init_TM80(0x78);     /* initialization of timer80 (interval ti
* 229 init_TMH1();        /* initialization of timerH1 */
230
* 231 INTM0=0x00;         /* set falling edge detection for INTP0 */
* 232 PMK0=0;            /* enable external interrupt INTP0 */
233
* 234 EI();              /* global interrupt enable */

```

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

#### Mixed display mode

```

* 0213 224801 CALL !_init_CPU
* 226 init_LED();          /* LED port initialization */
* 0216 228401 CALL !_init_LED
* 227 init_SW();
* 0219 226501 CALL !_init_SW
* 228 init_TM80(0x78);     /* initialization of timer80 (interval ti
* 021C F07800 MOVW AX,#78H
* 021F 22A901 CALL !_init_TM80
* 229 init_TMH1();        /* initialization of timerH1 */
* 0222 22C001 CALL !_init_TMH1
230
* 231 INTM0=0x00;         /* set falling edge detection for INTP0 */
* 0225 F7EC00 MOV INTM0,#0H
* 232 PMK0=0;            /* enable external interrupt INTP0 */
* 0228 0AA6E4 CLR1 MK0.2H
233
* 234 EI();              /* global interrupt enable */
* 022B 0A7A1E EI

```

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.



### 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) func#var (to specify a static variable with function 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	equesym file#equesym (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.

This section explains the following items:

- Break types
- Breakpoint setting
- Setting breaks to variables [IECUBE] [MINICUBE+]
- Hardware break and software break [IECUBE] [MINICUBE+]

### 5.4.1 Break types

The ID78K0S-QB has the following break functions.

Table 5-6 Break Types

Item	Contents
Hardware break <b>Note1</b> (Event detection break) [IECUBE] [MINICUBE+]	Function to stop user program execution upon detection of the set break event condition. -> Refer to "5.4.3 Setting breaks to variables [IECUBE] [MINICUBE+]".
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 "5.4.4 Hardware break and software break [IECUBE] [MINICUBE+]"). -> Refer to "5.4.2 Breakpoint setting".
[Come Here] break <b>Note2</b> (Temporary 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> the <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.
Break due to stack pointer initialization failure (fail-safe break) [IECUBE] [MINICUBE+]	Function to stop the user program if an interrupt occurs or a stack manipulation instruction, such as a PUSH instruction, is executed while the initial setting has not been made for the stack pointer

**Note1** This break is valid for [Go], [Go & GO], [Come Here] and [Restart].

**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

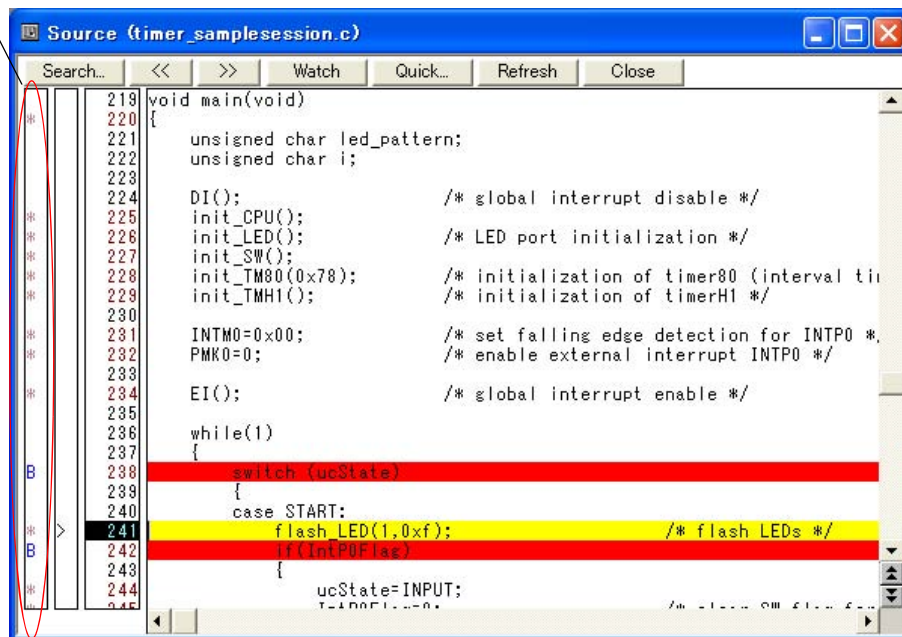
Software breakpoints can simply be set to the desired location by clicking in the [Source Window](#) or the [Assemble Window](#).

### (1) Breakpoint setting method

Breakpoints are executed by clicking lines in which " \* " is displayed (lines where program code exists). "B" is displayed in the line where a breakpoint is set.

Figure 5-1 Breakpoint Setting

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



### (2) Deleting a breakpoint method

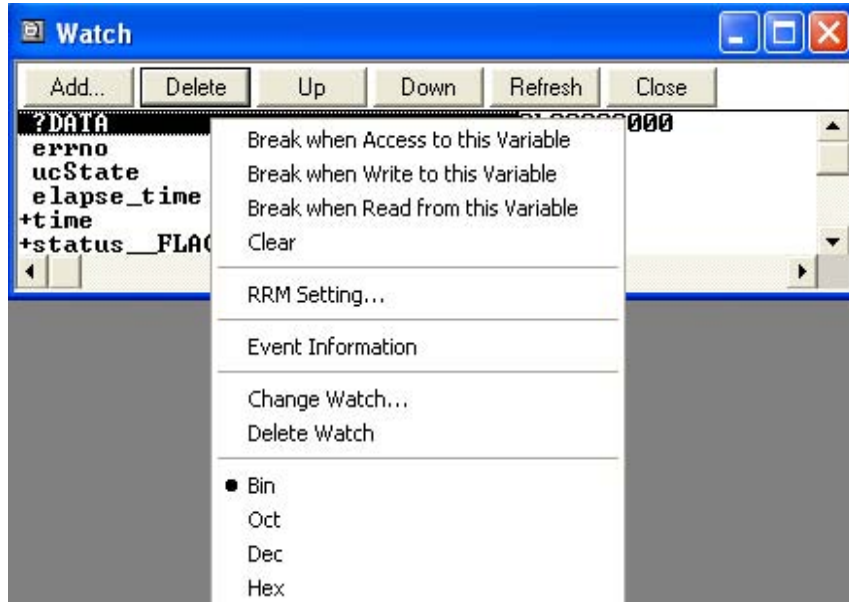
Click the position at which the breakpoint to be deleted is set.

"B" is deleted.

### 5.4.3 Setting breaks to variables [IECUBE] [MINICUBE+]

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

Figure 5-2 Setting Break to Variable



## 5.4.4 Hardware break and software break [IECUBE] [MINICUBE+]

### (1) Hardware break

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

Therefore, in the ID78K0S-QB, they are managed using "5.10 Event Function [IECUBE] [MINICUBE+]" as break event conditions.

The number of valid break event condition varies depending on the device (refer to "5.10.4 Number of enabled events for each event condition").

Access events can only be set as hardware breaks in the ID78K0S-QB.

### (2) Software break

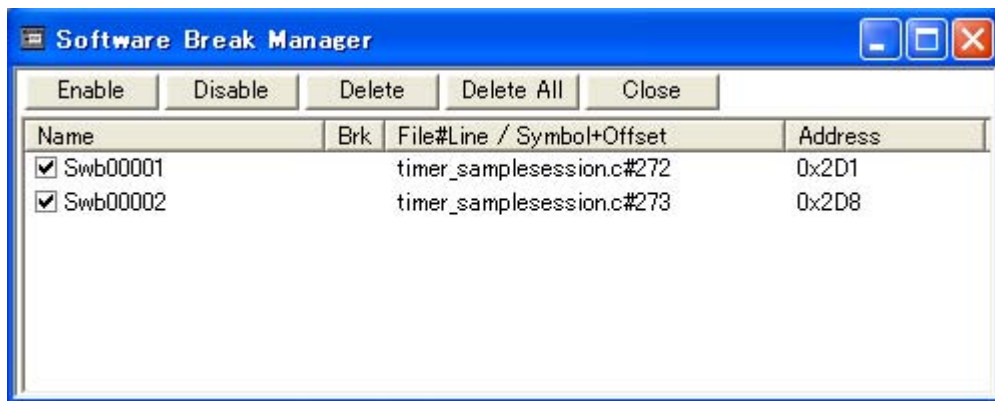
Software breaks are breaks that are set by rewriting instructions of specified addresses to software break instructions. Stopping at variable access timing, etc., cannot be specified.

Table 5-7 The Number of Valid Software Break

Product Name	Valid Number
ID78K0S-QB	2000

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

Figure 5-3 Management of Software Breaks



## 5.5 Program Execution Function

The program execution function is used to start execution of the user program by the CPU.

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

The following types of ID78K0S-QB execution functions are provided. They are operated using the execution buttons on the tool bar , or from the [Run] menu.

Figure 5-4 Execution Button

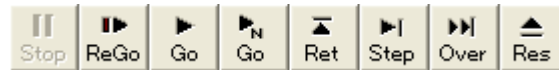


Figure 5-5 [Run] Menu

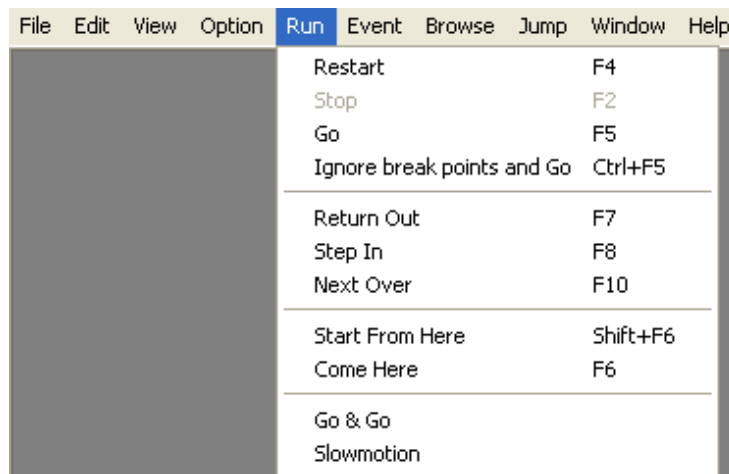


Table 5-8 Type of Execution

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 <a href="#">Source Window</a> , 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 <a href="#">Assemble Window</a> , One instruction is executed from the current PC register value and the contents of each window are updated.

Items	Contents
[Next Over]	<p><b>CALL / CALLT / CALLF instruction</b>  Next step execution is performed, assuming the function or subroutine called by the CALL / CALLT / CALLF instruction as one step (step execution continues until the nesting level becomes the same as when the CALL / CALLT / CALLF instruction was executed).</p> <p><b>Instruction other than CALL / CALLT / CALLF</b>  The same processing as [Step In] is performed.</p>
[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]	<p>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 the <a href="#">Assemble Window</a>, and then a break occurs.</p> <p>While the user program is being executed, the break event currently set does not occur.</p>
[Go & Go]	<p>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.</p> <p>The contents of each window are updated, and execution of the user program is resumed from the address at which the program was stopped.</p> <p>This operation is repeated until the user executes [Stop].</p>
[Slowmotion]	<p>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.</p> <p>This operation is repeated until the user executes [Stop].</p>
[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 5-6 Watch Window

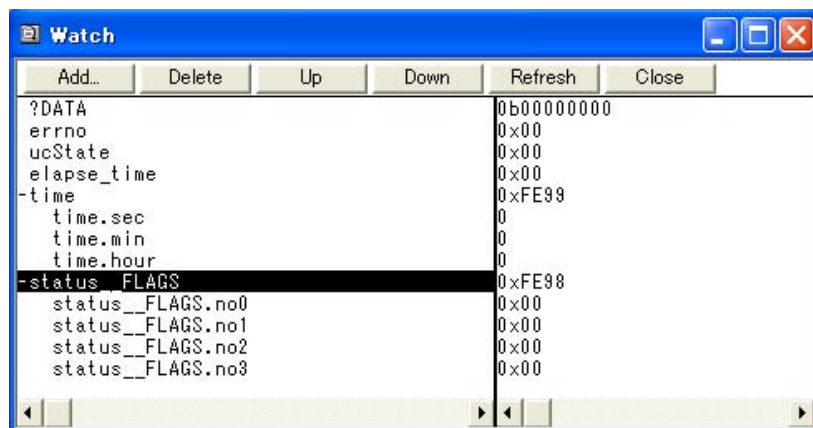
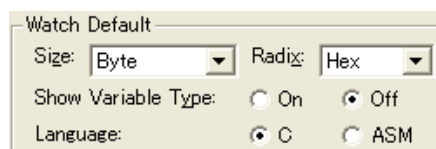


Figure 5-7 Specification of the Display Format (Debugger Option Dialog Box)



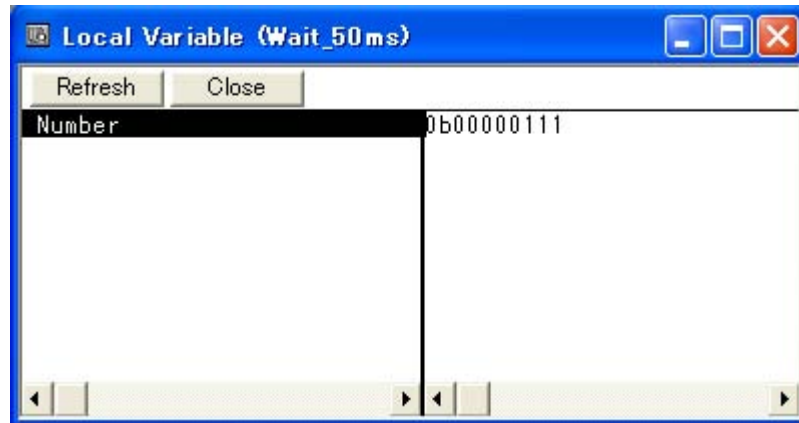


## 5.6.2 Displaying and changing local variable values

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 5-8 Local Variable Window



## 5.6.3 Registering and deleting watch data

Data can be registered to the [Watch Window](#) from the [Source Window](#) or the [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.

- Click the <Add> button in the [Quick Watch Dialog Box](#) or the [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 5-9 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 the [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 5-10 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 the [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 5-11 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](#)
- [Filling, copying, and comparing memory contents](#)
- [Flash memory writing function \[MINICUBE2\]](#)

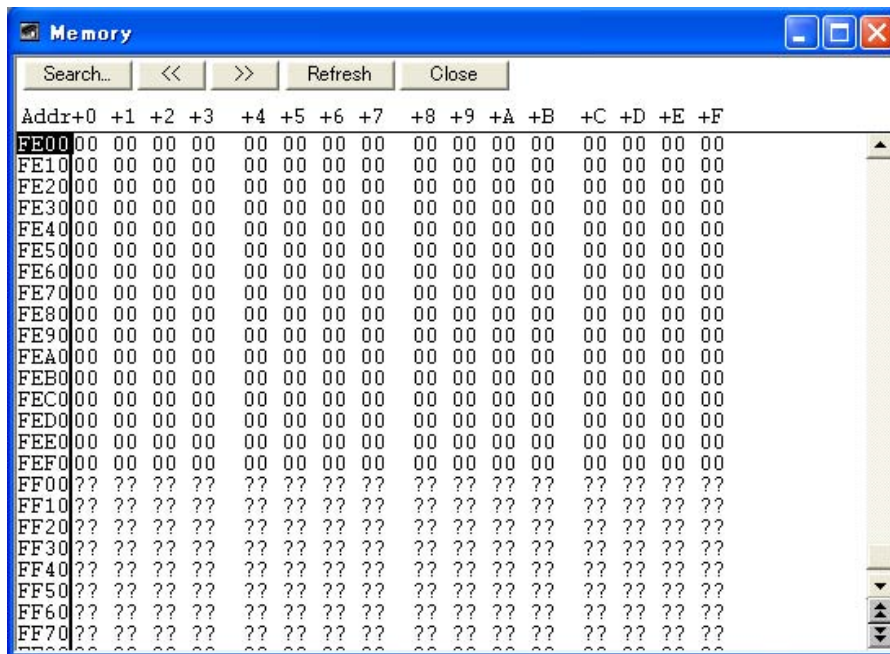
### 5.7.1 Displaying and changing memory contents

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 "5.11 Pseudo Real-time Monitor Function (Break When Readout) [IECUBE] [MINICUBE+]".)

Figure 5-12 Displaying and Changing Memory Contents



### 5.7.2 Filling, copying, and comparing memory contents

Memory contents are Filled, copied, and compared in the [Memory Fill Dialog Box](#), the [Memory Copy Dialog Box](#), and the [Memory Compare Dialog Box](#) displayed by selecting [Edit] menu -> [Memory] -> [Fill.../Copy.../Compare...].

The comparison results are displayed in the [Memory Compare Result Dialog Box](#).

### 5.7.3 Flash memory writing function [MINICUBE2]

With the ID78K0S-QB, the internal flash memory can be written and the load module can be downloaded by the same access method as an ordinary memory operation.

The data on the internal flash memory can be changed from the [Memory Window](#), [Assemble Window](#), [Watch Window](#), [Memory Fill Dialog Box](#) and [Memory Copy Dialog Box](#), without having to be aware that the data is that of the internal flash memory.

The load module can also be downloaded to the internal flash memory by using the flash self-programming function.

**Caution** No data can be written to the internal flash memory during user program execution.

## 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 5-13 Absolute Name/Function Name Switching

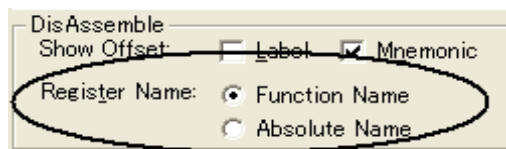


Table 5-9 Absolute Name to Function Name Correspondence

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

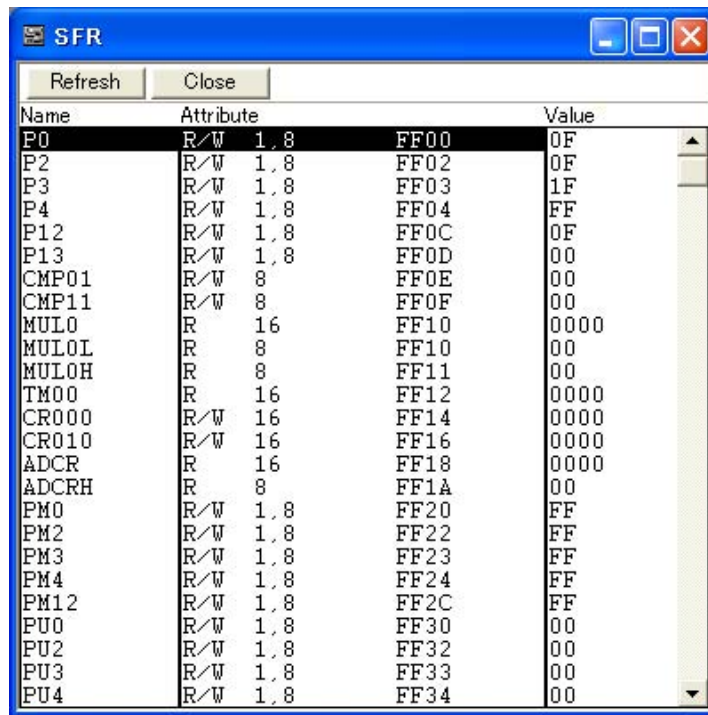
## 5.8.2 Displaying and changing SFR contents

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](#).

Figure 5-14 Displaying SFR Contents



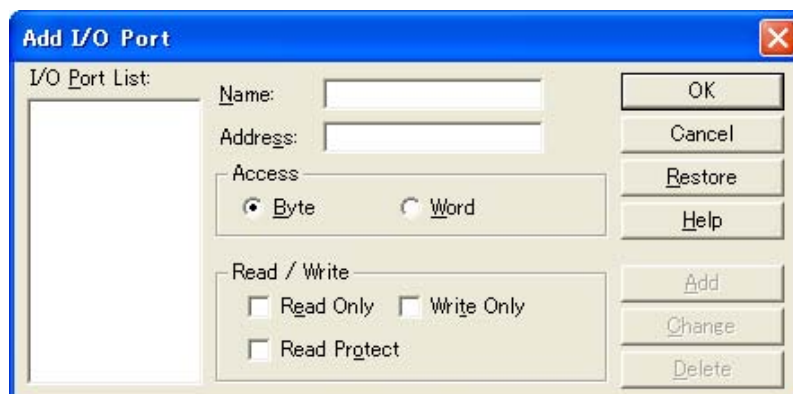
Name	Attribute	Value
P0	R/W 1, 8	FF00
P2	R/W 1, 8	FF02
P3	R/W 1, 8	FF03
P4	R/W 1, 8	FF04
P12	R/W 1, 8	FF0C
P13	R/W 1, 8	FF0D
CMP01	R/W 8	FF0E
CMP11	R/W 8	FF0F
MUL0	R 16	FF10
MUL0L	R 8	FF10
MUL0H	R 8	FF11
TM00	R 16	FF12
CR000	R/W 16	FF14
CR010	R/W 16	FF16
ADCR	R 16	FF18
ADCRH	R 8	FF1A
PM0	R/W 1, 8	FF20
PM2	R/W 1, 8	FF22
PM3	R/W 1, 8	FF23
PM4	R/W 1, 8	FF24
PM12	R/W 1, 8	FF2C
PU0	R/W 1, 8	FF30
PU2	R/W 1, 8	FF32
PU3	R/W 1, 8	FF33
PU4	R/W 1, 8	FF34

## 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



**Add I/O Port**

I/O Port List:

Name:

Address:

Access:

☒ Byte ☐ Word

Read / Write:

☐ Read Only ☐ Write Only

☐ Read Protect

Buttons: OK, Cancel, Restore, Help, Add, Change, Delete

## 5.9 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.

Only fetch data can be traced with the trace function of the ID78K0S-QB.

Tracing from start to end of user program execution can be performed.

When the program is re-executed, the previous trace result will be cleared.

This section explains the following items:

- [Checking trace data](#)
- [Mixed display mode \(Trace Window\)](#)

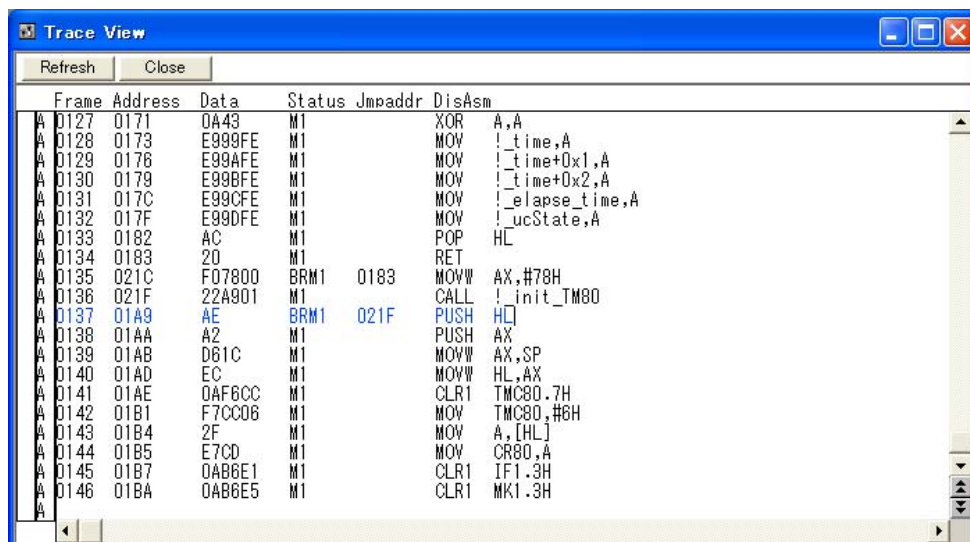
### 5.9.1 Checking trace data

The trace data saved to the trace memory can be checked in the [Trace View Window](#).

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](#).

Figure 5-16 Checking Trace Data



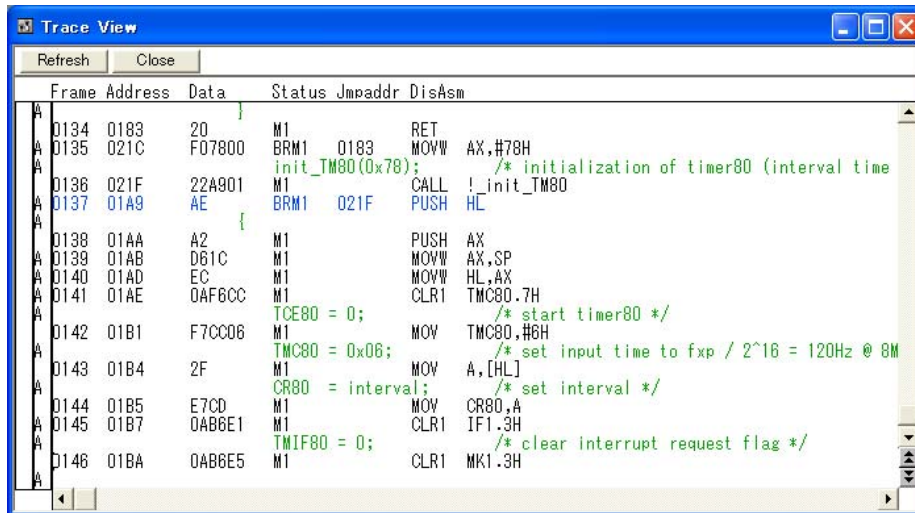


## 5.9.2 Mixed display mode (Trace Window)

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](#).



**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 Event Function [IECUBE] [MINICUBE+]

Events specify specific states of the target system during debugging ,such as "wrote data to address 0x2000".  
The ID78K0S-QB uses an event to trigger an action of functions such as break functions.

This section explains the following items:

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

### 5.10.1 Using event function

Events (event conditions) are assigned various debugging functions and consist of the event conditions listed in the following table.

Table 5-10 Various Event Conditions

Event Condition	Mark	Contents ->Setting Dialog Box
Break event condition	B	Condition in which the execution of the user program is stopped (refer to " <a href="#">5.4 Break Function</a> "). -> <a href="#">Break Dialog Box</a>

### 5.10.2 Creating events

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

#### (1) Creating and registering events

Event conditions are usually created 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.

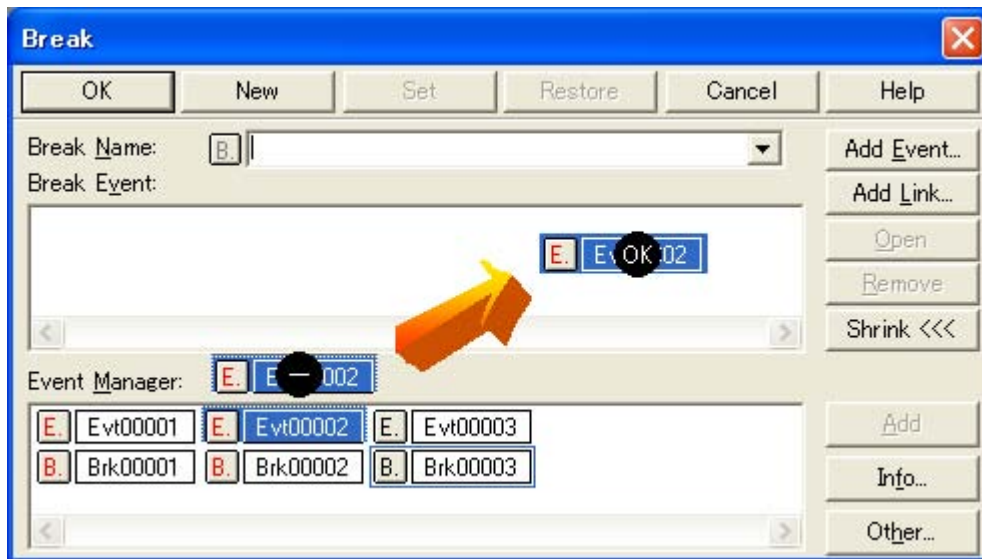
### 5.10.3 Setting various event conditions

Various event conditions listed in [Table 5-10](#) 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 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 5-17 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 <Set> or <OK> button in the Setting dialog box. After the event has been set, a debugging action occurs as various event conditions.

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

The [Event Dialog Box](#) is open in the "selection mode" by placing the focus on the condition area to be set and then clicking the <Add Event...> button. The corresponding detailed condition is displayed when a condition set in the dialog box is selected, so 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 the event condition was dropped using only the mouse, move the 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.

### 5.10.4 Number of enabled events for each event condition

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

However, the number of event conditions that can be set (enabled) at the same time is limited as follows:

Therefore, if the valid number is exceeded or if the used event 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 ["5.10.5 Managing events"](#).)

Table 5-11 Number of Enabled Events for Each Event Condition

Connected IE	Event		Event Link	Break	Trace	Timer
	Execute	Access				
IECUBE	0	1	-	1	-	-
MINICUBE+	0	1	-	1	-	-
MINICUBE2	-	-	-	-	-	-

## 5.10.5 Managing events

Events are managed by the [Event Manager](#).

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

Figure 5-18 Managing Events (Event Manager)



### (1) Event icons

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 5-12 Event Icon

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

## 5.11 Pseudo Real-time Monitor Function (Break When Readout) [IECUBE] [MINICUBE+]

This is a function that performs pseudo real-time RAM monitoring via software emulation. When this function is used for reading data, a break occurs momentarily upon a read during the user program execution.

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

Specify turning on/off of the pseudo real-time RAM monitor function and the sampling time setting in the [Extended Option Dialog Box](#), and the sampling range in the [RRM Dialog Box](#).

Table 5-13 Areas for Which Sampling Can Be Performed with Pseudo Real-time RAM Monitor Function

Connected IE	Sampling Range
[IECUBE]	Internal RAM(Internal high-speedRAM) area
[MINICUBE+]	None

This section explains the following items:

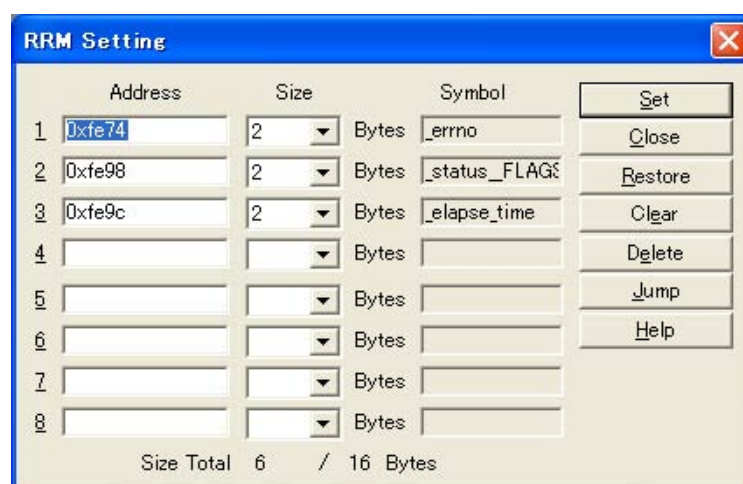
- [Sampling range setting](#)
- [Sampling time setting](#)

### 5.11.1 Sampling range setting

The sampling range for the RRM function is set in the [RRM Dialog Box](#).

Up to 8 locations can be set in 1-byte units in the 16 bytes RRM area.

Figure 5-19 Sampling Range Setting (RRM Dialog Box)



### 5.11.2 Sampling time setting

The sampling time for the RRM function is set in the [Extended Option Dialog Box](#).

Values are updated at the specified time interval.

Figure 5-20 Sampling Time Setting



## 5.12 Load/Save Function

ID78K0S-QB allows saving and loading the following types of information as files. As a result, recovery of these various types of information is possible.

**Remark** The simple window status can be maintained by selecting [Window] menu -> [Static]. (Refer to "5.13.1 Active status and static status".)

This section explains the following items:

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

### 5.12.1 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 the [Project File Load Dialog Box](#), respectively. To load a project file at startup, press the <Project...> button in the [Configuration Dialog Box](#).

The following contents are saved to the project file:

Table 5-14 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/status bar/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>	Set information
<a href="#">Source Window</a>	Display information of window
<a href="#">Assemble Window</a> <a href="#">Memory Window</a>	Display information of window, display start address
<a href="#">Stack Window</a> <a href="#">SFR Window</a> <a href="#">Local Variable Window</a> <a href="#">Event Manager</a> <a href="#">Console Window</a> <a href="#">Expansion Window</a>	Display information of window
<a href="#">Event Dialog Box</a>	Display information of window, event information
<a href="#">Break Dialog Box</a>	Display information of window, break event information



Window Name	Saved Contents
<a href="#">Register Window</a>	Display information of window
<a href="#">Watch Window</a>	Display information of window, watch registration information
<a href="#">Add I/O Port Dialog Box</a>	Added I/O port information
<a href="#">Software Break Manager</a>	Display information of window, software break information

### 5.12.2 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 the [View File Save Dialog Box](#), respectively.

Table 5-15 Type of View Files

File Type	Target Window File Name
Source Text (*.svw)	<a href="#">Source Window</a> View file of the Source Window
Assemble (*.dis)	<a href="#">Assemble Window</a> View file of the Assemble Window
Memory (*.mem)	<a href="#">Memory Window</a> View file of the Memory Window
Watch (*.wch)	<a href="#">Watch Window</a> View file of the Watch Window
Register (*.rgw)	<a href="#">Register Window</a> View file of the Register Window
SFR (*.sfr)	<a href="#">SFR Window</a> View file of the SFR Window
Local Variable (*.loc)	<a href="#">Local Variable Window</a> View file of the Local Variable Window
Stack Trace (*.stk)	<a href="#">Stack Window</a> View file of the Stack Window
All (*.*)	All files
Trace (*.tvw) [IECUBE]	<a href="#">Trace View Window</a> View file of the Trace View Window
Event (*.evn) [IECUBE] [MINICUBE+]	<a href="#">Event Manager</a> View file of the Event Manager
Console(*.log)	<a href="#">Console Window</a> View file of the Console Window

File Type	Target Window File Name
Source (*.c, *.s, *.asm) (During downloading)	Source file <sup>Note</sup>
Text (*.txt)	Text file

**Note** The extension of the source file can be changed in the [Extended Option Dialog Box](#).

### 5.12.3 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 the [Environment Setting File Save Dialog Box](#), respectively.

Table 5-16 Type of Setting Files

File Type	Target Window File Name
Watch (*.wch) <sup>Note</sup>	<a href="#">Watch Window</a> Setting file of the Watch window
SFR (*.sfr) <sup>Note</sup>	<a href="#">SFR Window</a> Setting file of the SFR window
Event (*.evn) [IECUBE] [MINICUBE+]	<a href="#">Event Manager</a> Setting file of event

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

## 5.13 Functions Common to Each Window

The windows have the following common functions.

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

### 5.13.1 Active status and static status

The [Source Window](#) that is displaying the source file to which symbol information is read, [Assemble Window](#) and [Memory Window](#) has two statuses:

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](#).

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](#).

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.

### 5.13.2 Jump function

The jump function is used to jump to the [Source Window](#), the [Assemble Window](#) or the [Memory Window](#) from a jump pointer that is the line or address on which the cursor is placed, and display the contents of the jump destination window from the jump pointer.

You can jump among the above windows, or from the [Stack Window](#), the [Event Manager](#) and the [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.

[Source Window](#) -> Select [Source Text]

[Assemble Window](#) -> Select [Assemble]

[Memory Window](#) -> Select [Memory]

**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 5-17 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
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.
Jump to the <a href="#">Source Window</a> or the <a href="#">Assemble Window</a>	Fetch address	

### 5.13.3 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 5-18 Details of Drag & Drop Function (Line/Address)

Window/Area to Drop to	Operation After Drop
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 5-19 Details of Drag & 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) 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:	
	<b>Symbols</b>	<b>Status</b>
	Variable	Access (R/W)
	Symbol in data section	Access (R/W)
	Others	Access (R/W)

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

**Caution** The various event setting dialog boxes include the [Event Dialog Box](#) and the [Break Dialog Box](#).

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

By linking the [Trace View Window](#) with each window ([Source Window](#) or [Assemble 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** When the [Source Window](#) or [Assemble Window](#) is linked, the fetch address is always used as the pointer.

### 5.13.5 Cautions

- The number of characters that can be displayed on 1 line in each area of a window is 319.
- If the width of the display area is narrow, the display may become corrupted. In this case, increase the width of the window.
- When scrolling is performed using the cursor keys in the [Mixed display mode \(Source Window\)](#), excessive scrolling may occur. Also, scrolling down to the last line may not be possible using the cursor keys.
- If the screen resolution is low (800 - 600, etc.), all the statuses may not be displayed on the status bar.
- 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).
- Memory Window  
When the display address is changed in the Memory Window, the position of the caret in the ASCII display area is not synchronized.
- SFR select Dialog Box  
If the display arrangement is changed in the SFR select Dialog Box, multiple lines cannot be selected. Select one line at a time.
- The number of lines displayed in the Watch Window  
A maximum of 10,000 lines can be displayed in the Watch Window.
- The number of C and assembly language source file support lines  
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.
- The switching of applications during dragging  
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.



# CHAPTER 6 WINDOW REFERENCE

This chapter explains in detail the functions of the windows and dialog boxes of ID78K0S-QB.

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

## 6.1 Window List

The list is the windows of the ID78K0S-QB.

Table 6-1 Window List

Window Name	Contents
Main Window	This window is displayed first, when the ID78K0S-QB is started. It controls execution of the user program. Various windows are opened from this window.
Configuration Dialog Box	Displays and sets the ID78K0S-QB operation environment.
Extended Option Dialog Box	Displays and sets the extended options of the ID78K0S-QB.
RRM Dialog Box	Sets the RRM sampling range.
Debugger Option Dialog Box	Displays and sets other options.
Project File Save Dialog Box	Saves the current debug environment to project file.
Project File Load Dialog Box	Loads the debug environment.
Download Dialog Box	Loads an object file and binary file.
Upload Dialog Box	Saves the memory contents to a file.
Source Window	Displays a source file and text file.
Source Search Dialog Box	Searches in the <a href="#">Source Window</a> .
Source Text Move Dialog Box	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.
Assemble Window	Disassembles the program and executes online assembly.
Assemble Search Dialog Box	Searches in the <a href="#">Assemble Window</a> .
Address Move Dialog Box	Specifies the start address to display the contents of the <a href="#">Memory Window</a> or <a href="#">Assemble Window</a> .
Symbol To Address Dialog Box	Displays the address of the specified variable or function, or the value of the specified symbol.
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.

Window Name	Contents
Memory Compare Dialog Box	Compares the memory.
Memory Compare Result Dialog Box	Displays the results of comparing the memory.
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> .
Trace View Window	Displays trace results. <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>
Event Manager	Displays, enables/disables, and deletes each event condition. <b>[IECUBE] [MINICUBE+]</b>
Software Break Manager	Display, enable or disable, and delete software breaks.
Event Dialog Box	Registers event conditions. <b>[IECUBE] [MINICUBE+]</b>
Break Dialog Box	Registers and sets break event conditions. <b>[IECUBE] [MINICUBE+]</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 ID78K0S-QB,CPU, and symbol information.
Exit Debugger Dialog Box	Terminates the ID78K0S-QB.
About Dialog Box	Displays the version of the ID78K0S-QB.
Console Window	Inputs commands.
Font Dialog Box	Displays the types of fonts displayed.
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.

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).

### Function buttons

---

Explains the operation of each button in the window or dialog box.

### Related operations

---

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

## Main Window

This window is automatically opened when the ID78K0S-QB is started up and initialized.

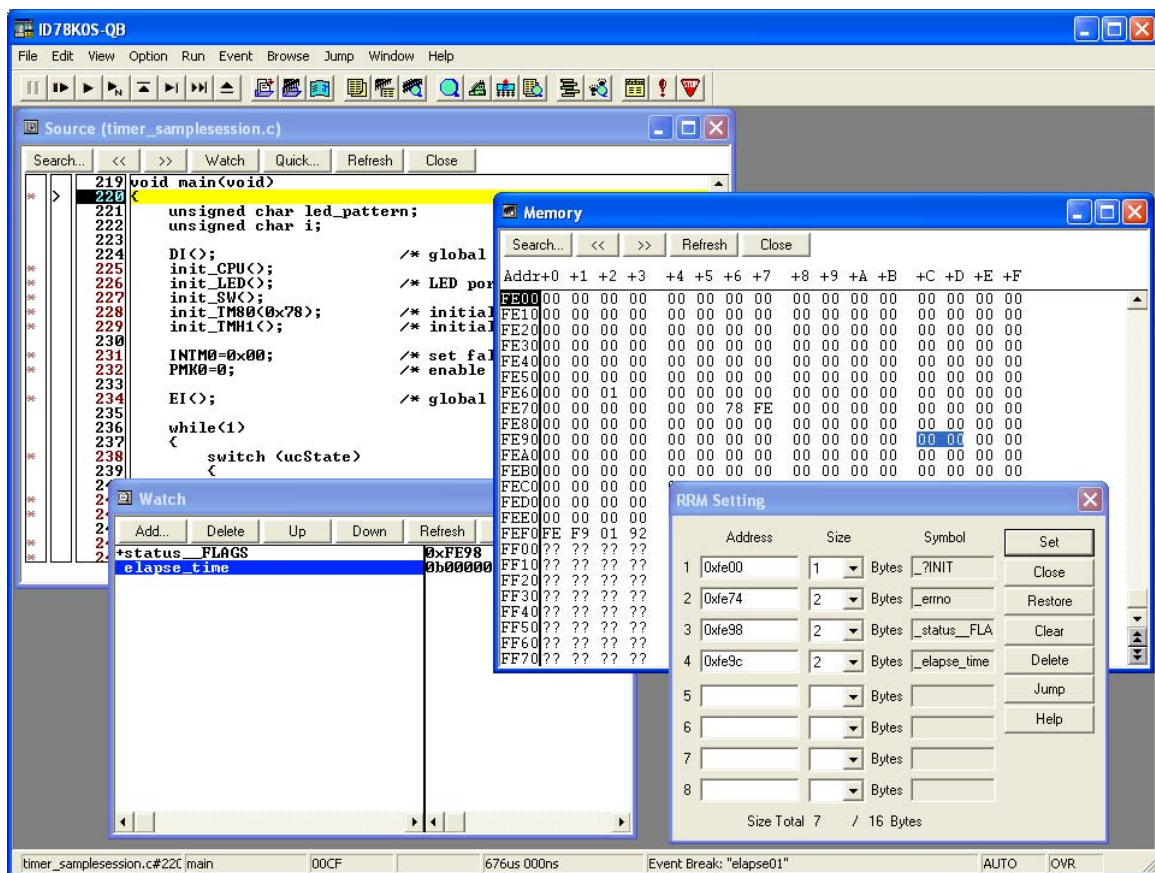
In the ID78K0S-QB, other windows are manipulated from this window. (Refer to "Table 6-1 Window List".)

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.)

Figure 6-1 Main Window



- Menu bar
- Toolbar
- Window display area
- Status bar

## Menu bar

- (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> .
Upload...	Uploads a file. 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 ID78K0S-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 target CPU, symbols, and the ID78K0S-QB. Opens the <a href="#">Reset Debugger Dialog Box</a> .
Exit	Terminate the ID78K0S-QB. Opens the <a href="#">Exit Debugger Dialog Box</a> .
(Open file)	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	Cancels 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 contents. Opens the <a href="#">Memory Copy Dialog Box</a> .
Compare...	Compares the memory contents. Opens the <a href="#">Memory Compare Dialog Box</a> .
Edit Source	Opens the source file displayed in the active the <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 tool bar is displayed (default) or not.
Status Bar	Selects whether the status bar 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 functions. Opens the <a href="#">Extended Option Dialog Box</a> .
RRM Setting... [IECUBE] [MINICUBE+]	Sets the sampling range for the RRM function. Opens the <a href="#">RRM Dialog Box</a> .
Debugger Option...	Sets the ID78K0S-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> .




**(5) [Run] menu**

Restart	Resets the CPU and executes the program.  Same operation as this button.
Stop	Forcibly stops program execution.  Same operation as this button.
Go	Executes the program from the current PC.  Same operation as this button.
Ignore break points and Go	Ignores break points being set, and executes the program.  Same operation as this button.
Return Out	The user program is executed until execution returns  Same operation as the this button. <b>Note:</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 this button.












Next Over	<p>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.</p> <p> Same operation as this button.</p>
Start From Here	<p>Executes the program from the cursor position on the <a href="#">Source Window</a> or the <a href="#">Assemble Window</a>.</p>
Come Here	<p>Executes the program from the current PC to the cursor position in the <a href="#">Source Window</a> or the <a href="#">Assemble Window</a>.</p>
Go & Go	<p>Continues executing the program. If a break occurs because a break condition is satisfied, the window is updated and the program is executed again.</p> <p> Same operation as clicking this button each time a break has occurred.</p>
Slowmotion	<p>Continues step execution. Each time step execution has been performed, the window is updated and then step execution is performed again.</p> <p> Same operation as clicking this button each time a break has occurred.</p>
CPU Reset	<p>Resets the CPU.</p> <p> Same operation as the this button.</p>
Change PC	<p>Sets the address at the cursor position in the <a href="#">Source Window</a> or the <a href="#">Assemble Window</a> to the PC.</p>
Software Break Point	<p>Sets or cancels a software breakpoint at the cursor position in the <a href="#">Source Window</a> or the <a href="#">Assemble Window</a>.</p>
Delete All Software Break	<p>Deletes all the set software break points.</p>

**(6) [Event] menu**

Event Manager <b>[IECUBE] [MINICUBE+]</b>	<p>Manages various event conditions. Opens the <a href="#">Event Manager</a>.</p> <p> Same operation as this button.</p>
Software Break Manager	<p>Manages software break event conditions. Opens the <a href="#">Software Break Manager</a>.</p>
Event... <b>[IECUBE] [MINICUBE+]</b>	<p>Registers an event condition. Opens the <a href="#">Event Dialog Box</a>.</p> <p> Same operation as this button.</p>
Break... <b>[IECUBE] [MINICUBE+]</b>	<p>Registers and sets a break condition. Opens the <a href="#">Break Dialog Box</a>.</p> <p> Same operation as this button.</p>

**(7) [Browse] menu**

Source Text	<p>Displays a source text. Opens the <a href="#">Source Window</a>. If there is a <a href="#">Source Window</a> already open in the active status, it is opened in the static status.</p> <p> Same operation as this button.</p>
Assemble	<p>Displays the disassemble results. Opens the <a href="#">Assemble Window</a>. If there is a <a href="#">Assemble Window</a> already open in the active status, it is opened in the static status.</p> <p> Same operation as this button.</p>
Memory	<p>Displays the contents of the memory. Opens the <a href="#">Memory Window</a>. If there is a <a href="#">Memory Window</a> already open in the active status, it is opened in the static status.</p> <p> Same operation as this button.</p>
Watch	<p>Displays the watch contents. Opens the <a href="#">Watch Window</a>.</p> <p> Same operation as this button.</p>
Register	<p>Displays the register contents. Opens the <a href="#">Register Window</a>.</p> <p> Same operation as this button.</p>
SFR	<p>Opens SFR. Opens the <a href="#">SFR Window</a>.</p> <p> Same operation as this button.</p>
Local Variable	<p>Displays the local variable. Opens the <a href="#">Local Variable Window</a>.</p> <p> Same operation as this button.</p>
Stack Trace	<p>Displays the stack trace results. Opens the <a href="#">Stack Window</a>.</p> <p> Same operation as this button.</p>
Trace [IECUBE]	<p>This area displays the trace results. Opens the <a href="#">Trace View Window</a>.</p> <p> Same operation as this button.</p>
Console	<p>Opens the <a href="#">Console Window</a>.</p>
Others	<p>Displays other windows. Displays a user-defined window list.</p>

**(8) [Jump] menu**

Source Text	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 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 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 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 jump destination address specified by the data value selected in the current window. 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).

**(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 the <a href="#">Source Window</a> , the <a href="#">Assemble Window</a> , or the <a href="#">Memory Window</a> .
Cascade	Cascade display of the windows in the main window.
Tile	Tile 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.
(Open Window)	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**

ID78K0S-QB Help	Displays the help of the debugger.
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 ID78K0S-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>Go</b>	Ignores break points being set, and executes the user program. Same function as [Run] menu -> [Ignore break points and Go] .
 <b>Ret</b>	The user program is executed until execution returns. Same function as [Run] menu -> [Return Out] . <b>Note:</b> This command is used for a function described in C language.
 <b>Step</b>	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].
 <b>Over</b>	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].
 <b>Res</b>	Resets the CPU. Same function as [Run] menu -> [CPU Reset].
 <b>Open</b>	Opens the <a href="#">View File Load Dialog Box</a> . Same function as [File] menu -> [Open].
 <b>Load</b>	Opens the <a href="#">Download Dialog Box</a> . Same function as [File] menu -> [Download].
 <b>Proj</b>	Opens the <a href="#">Project File Load Dialog Box</a> . Same function as [File] menu -> [Project] -> [Open].
 <b>Src</b>	Displays the source text. Opens the <a href="#">Source Window</a> . Same function as [Browse] menu -> [Source Text].
 <b>Asm</b>	Displays the disassemble results. Opens the <a href="#">Assemble Window</a> . Same function as [Browse] menu -> [Assemble].
 <b>Mem</b>	Displays the contents of the memory. Opens the <a href="#">Memory Window</a> . Same function as [Browse] menu -> [Memory].
 <b>Wch</b>	Displays the watch contents. Opens the <a href="#">Watch Window</a> . Same function as [Browse] menu -> [Watch].
 <b>Reg</b>	Displays the register contents. Opens the <a href="#">Register Window</a> . Same function as [Browse] menu -> [Register].

 <b>SFR</b>	Displays the contents of SFR. Opens the <a href="#">SFR Window</a> . Same function as [Browse] menu -> [SFR].
 <b>Loc</b>	Displays the local variable contents. Opens the <a href="#">Local Variable Window</a> . Same function as [Browse] menu -> [Local Variable].
 <b>Stk</b>	Displays the stack trace results. Opens the <a href="#">Stack Window</a> . Same function as [Browse] menu -> [Stack Trace].
 <b>TrW</b> [IECUBE]	This area displays the trace results. Opens the <a href="#">Trace View Window</a> . Same function as [Browse] menu -> [Trace].
 <b>Mgr</b> [IECUBE] [MINICUBE+]	Opens the <a href="#">Event Manager</a> . Same function as [Event] menu -> [Event Manager].
 <b>Evn</b> [IECUBE] [MINICUBE+]	Registers and sets events. Opens the <a href="#">Event Dialog Box</a> . Same function as [Event] menu -> [Event].
 <b>Brk</b> [IECUBE] [MINICUBE+]	Registers and sets break events. Opens the <a href="#">Break Dialog Box</a> . Same function as [Event] menu -> [Break].

## (2) Operation of toolbar

Whether the tool bar 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 6-2 Toolbar (Picture Only)

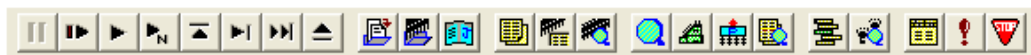


Figure 6-3 Toolbar (Picture and Text)



## Window display area

This area (at the center) displays various debug windows.

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

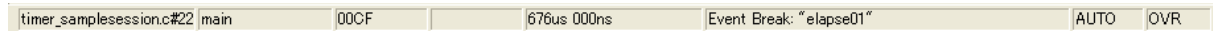
## Status bar

The status bar displays the status of the ID78K0S-QB and in-circuit emulator.

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

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

Figure 6-4 Status Bar



(1) Program name	Displays the program file name indicated by the PC value.
Source name	Displays the source file name indicated by the PC value.
Line number	Displays the line number indicated by the PC value.
(2) Function name	Displays the function name indicated by the PC value.
(3) PC value	Displays the current PC value.
(4) CPU status	Refer to "Table 6-2 CPU Status".
(5) CPU status	Refer to "Table 6-3 IE Status". (If there are two or more the statuses, they delimited with ' ' and displayed.)
(6) Break Cause	Refer to "Table 6-4 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.

Table 6-2 CPU Status

Display	Meaning
HALT	Halt mode <b>[IECUBE]</b>
STOP	Stop mode <b>[IECUBE]</b>
RESET	Reset mode
POW OFF	Power is not supplied to the target

Table 6-3 IE Status

Display	Meaning
RUN	User program execution in progress (the color of the status bar changes).
STEP	Step execution in progress
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) The measurable values is as follows: 1 hours 12 minutes max. (Resolution: 145 nsec) <b>[IECUBE]</b> 119 hours 18 minutes max. (Resolution: 100 usec) <b>[MINICUBE2]</b>
TIMER OVERFLOW	Measurement result overflowed. <b>[IECUBE]</b> <b>[MINICUBE2]</b>

Table 6-4 Break Cause

Display	Meaning
Manual Break	Forced break
Temporary Break	Temporary break
Event Break " <i>Event Name</i> "	Stopped due to event cause of displayed event name. <b>[IECUBE]</b> <b>[MINICUBE+]</b>
Software Break	Software break
Uninitialize Stack Pointer	Break due to failure to perform stack pointer initialization

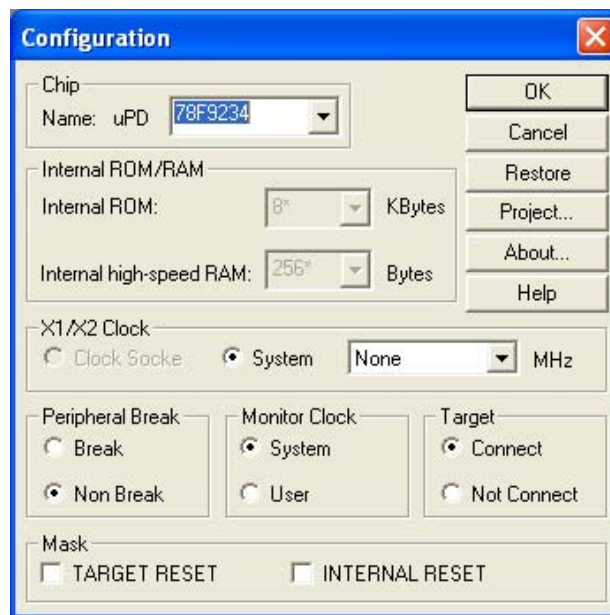
## Configuration Dialog Box

This dialog box is used to display and set the ID78K0S-QB operation environment. (Refer to "[5.1 Setting Debugging Environment](#)".)

This dialog box is automatically displayed after the ID78K0S-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 "[5.12.1 Debugging environment \(project file\)](#)".)

Figure 6-5 Configuration Dialog Box



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

### Opening

(Automatically when the ID78K0S-QB is started up)

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



## Explanation of each area

- (1) Chip
- (2) Internal ROM/RAM
- (3) X1/X2 Clock [IECUBE] [MINICUBE+]
- (4) Peripheral Break [IECUBE] [MINICUBE+]
- (5) Monitor Clock
- (6) Target [IECUBE] [MINICUBE+]
- (7) Mask

### (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.

**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 ROM/RAM

This area is used to set the size of the internal ROM and internal RAM of the CPU.

The sizes of the internal ROM and RAM for the device files are fixed in the ID78K0S-QB.

### (3) X1/X2 Clock [IECUBE] [MINICUBE+]

This area is used to select a clock source input to the X1 pin and X2 pin.

In this area, if the oscillator is not connected to the clock socket on the emulator, the setting is fixed to "System".

If the oscillator is connected, the setting is fixed to "Clock Socket".

Clock Socket	The clock for the clock socket on the emulator is input to the X1 pin and X2 pin.
System	<p>A divided frequency from the internal emulator system clock is input to the X1 pin and X2 pin.</p> <p>The selectable frequencies are as follows.</p> <p><b>[IECUBE]</b> None (default), 5.00, 8.00, 10.00</p> <p><b>[MINICUBE+]</b> None (default), 5.00</p>

**(4) Peripheral Break [IECUBE] [MINICUBE+]**

This area is used to specify whether the peripheral emulation function of in-circuit emulator is stopped during a break.

Break	Peripheral emulation functions are stopped during breaks.
Non Break	Peripheral emulation functions are not stopped during breaks (default).

**Remark** The following peripheral macros are compatible with Peripheral Breaks.

- 16-bit timer/event counter 00
- 8-bit timer 80
- 8-bit timer H1

Regardless of the settings in this area, the operations of the watchdog timer are stopped during breaks.

Peripheral macros other than the above, operate during breaks.

**(5) Monitor Clock**

This area is used to specify whether the operation clock of the monitor program is switched from the subclock to the main clock during a break.

This area does not have to be set with a product without a subclock.

System	The operation clock is switched to the main clock and the monitor program is executed (default). <b>Caution:</b> In the ID78K0S-QB, the clock is changed by manipulating PCC, but not while the main clock is stopped. If the operation clock is switched to the main clock during a break, the clock is returned to the previous setting when execution returns to the user program.
User	The monitor program is executed with the clock selected by the user program.

**(6) Target [IECUBE] [MINICUBE+]**

This area is used to select whether the target board is to be connected to the in-circuit emulator or not.

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).

**(7) Mask**

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

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

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

**Remark** When the target system is unconnected, "TARGET RESET" is masked regardless of the settings. When "Not Connect" is specified in (6) [Target \[IECUBE\] \[MINICUBE+\]](#), "INTERNAL RESET" are fixed to the selected state, and the mask status cannot be changed. [\[IECUBE\] \[MINICUBE+\]](#)

**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 ID78K0S-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 ID78K0S-QB. (Refer to "5.1 Setting Debugging Environment".)

Figure 6-6 Extended Option Dialog Box



- Opening
- Explanation of each area
- Function buttons

### Opening

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

### Explanation of each area

- (1) RAM Monitor [IECUBE] [MINICUBE+]
- (2) Break Sound
- (3) Verify Check
- (4) Clear Register When Reset

**(1) RAM Monitor [IECUBE] [MINICUBE+]**

Set the real-time monitor (refer to ["5.11 Pseudo Real-time Monitor Function \(Break When Readout\) \[IECUBE\] \[MINICUBE+\]"](#)).

**(a) Break When Readout****(i) RRM Setting**

Specified area in the [" RRM Dialog Box"](#), execution of the user program momentarily breaks upon a read.

**(ii) Off**

Disables the Pseudo real-time monitor function.

**(b) Redraw Interval**

Specify the sampling time (ms) of the real-time monitor function.

The sampling time can be specified in 100-ms units from 0 to 65500.

If 0 is specified, or if this area is blank, the data is not displayed in real time.

**(2) Break Sound**

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

This item is selected by default.

**(3) Verify Check**

This area is used to specify whether a verify check is performed 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 the [Memory Window](#) and is written to memory.

This item is selected by default.

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

**(4) Clear Register When Reset**

Select this checkbox to set the SP register to 0xfef0 and the general-purpose registers (X, A, C, B, E, D, L, H) to 0x0 at CPU reset.

Under the default setting, the registers are not cleared.

**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.

## RRM Dialog Box

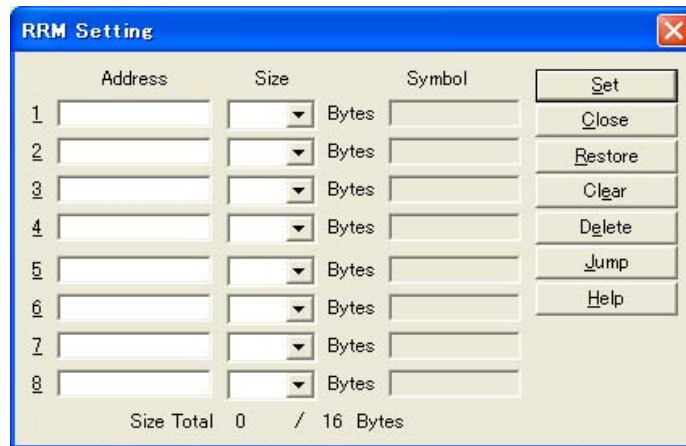
[IECUBE] [MINICUBE+]

This dialog box is used to set the sampling range for the RRM function. (Refer to "5.11 Pseudo Real-time Monitor Function (Break When Readout) [IECUBE] [MINICUBE+]".)

Up to 8 locations can be specified in 1-byte units as the sampling range.

The total of the sizes specified for the 8 locations cannot exceed 16 bytes.

Figure 6-7 RRM Dialog Box



- Opening
- Explanation of each area
- Function buttons

### Opening

The settings of this dialog box when it is opened differ depending on the opening method.

- (a) When settings are performed from RRM Dialog Box

The dialog box is opened by selecting [Option] menu -> [RRM Setting...].

In this case, the data in (1) Address and (2) Size are input manually.

- (b) When settings are performed from the Memory Window

This dialog box is opened by opening the Memory Window, selecting an address in the window, and then selecting [RRM Setting...] from the context menu.

In this case, the selected address is displayed in an empty row in (1) Address, 1 bytes is displayed in an empty row in (2) Size, and the value obtained by converting the address to a symbol is displayed in an empty row in (3) Symbol.

However, if the total of the sizes specified for the 8 locations already exceeds 16 bytes, the dialog box opens but no value can be set.

(c) When settings are performed from the [Watch Window](#)

This dialog box is opened by opening the [Watch Window](#), selecting a variable in the window, and then selecting [RRM Setting...] from the context menu.

In this case, the value obtained by converting the variable into an address is displayed in an empty row in (1) [Address](#), 1 bytes is displayed in an empty row in (2) [Size](#), and the value obtained by converting the variable to a symbol is displayed in an empty row in (3) [Symbol](#).

However, if the total of the sizes specified for the 8 locations already exceeds 16 bytes, the dialog box opens but no value can be set.

---

## Explanation of each area

---

(1) [Address](#)

(2) [Size](#)

(3) [Symbol](#)

(4) [Size Total](#)

### (1) Address

This area is used to specify the sampling start address for the RRM function.

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](#)".)

Following input, click the <Set> button to enable the settings.

### (2) Size

This area is used to specify the sampling range from (1) [Address](#).

The values that can be selected are 1 to 16.

However, the total of the sizes specified for the 8 locations cannot exceed 16 bytes.

### (3) Symbol

This area displays the symbols of the addresses specified in (1) [Address](#).

The specified addresses are displayed as a symbol or as a symbol + offset.

If the address has not been set, nothing is displayed.

### (4) Size Total

This area displays the total of the sizes specified in (2) [Size](#). If the total exceeds 16 bytes, it is displayed in **red**.

---

## Function buttons

---

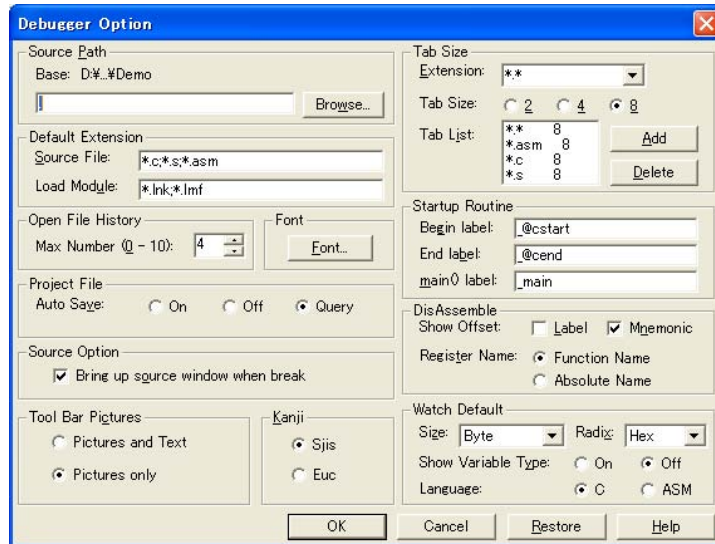
Set	Validates the settings and closes this dialog box.
Close	Closes this dialog box.
Restore	Restores the previous settings before this dialog box was opened.
Clear	Clears the current setting.
Delete	Deletes the setting for the numbers with a focus.
Jump	Opens the <a href="#">Memory Window</a> and displays the addresses in <a href="#">(1) Address</a> whose numbers have a focus. Jump is performed for <a href="#">Memory Window</a> that are in the active status. If multiple memory windows are to be opened, they must be set in the static status. (Refer to " <a href="#">5.13.1 Active status and static status</a> ".)
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 ID78K0S-QB.

Figure 6-8 Debugger Option Dialog Box



- Opening
- Explanation of each area
- Function buttons

### Opening

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

### Explanation of each area

- (1) Source Path
- (2) Default Extension
- (3) Open File History
- (4) Font
- (5) Project File
- (6) Source Option
- (7) Tool Bar Pictures
- (8) Kanji
- (9) Tab Size
- (10) Startup Routine
- (11) DisAssemble
- (12) Watch Default

**(1) Source Path**

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

**(a) Base**

The directory is the basis of a relative path is displayed. The base directory is determined in the following sequence:

- (i) Directory to which the project file has been loaded
- (ii) Directory to which a load module or hex file has been loaded last
- (iii) Current directory of Windows

**(b) Text box**

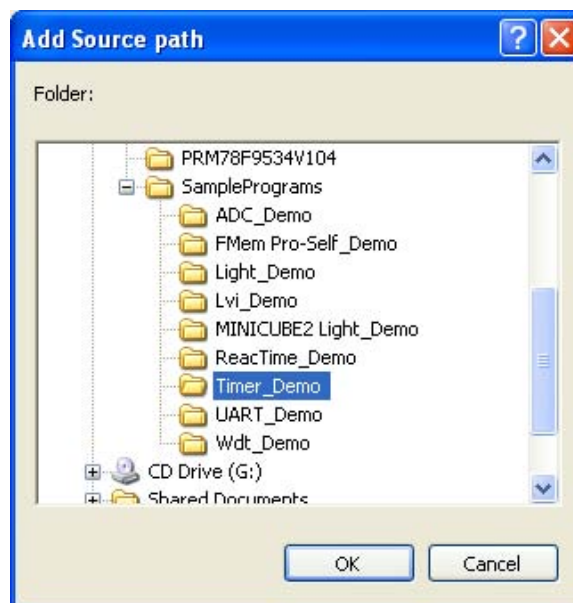
This area is used to specify the directory searched.

To specify a directory, either directly input one to the text box, or click the <Browse...> button. A relative path can also be specified.

Opens the [Add Source Path Dialog Box](#) by clicking the <Browse...> button. To delimit paths, use ";" (semicolon) or "," (comma).

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

Figure 6-9 Add Source Path Dialog Box



**(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](#), the [Watch Window](#), the [Quick Watch Dialog Box](#), the [Local Variable Window](#), and the [Stack Window](#). Clicking the <Font...> button opens the [Font Dialog Box](#) in which the font to be displayed and its size can be set.

**(5) Project File**

This area is used to set automatic saving and loading of the project file. (Refer to "[5.12.1 Debugging environment \(project file\)](#)".)

**(a) Auto Save**

Sets whether the project file is automatically saved at the ID78K0S-QB termination.

On	Automatically saves the project file at the ID78K0S-QB termination.
Off	Does not automatically save the project file at the ID78K0S-QB termination.
Query	Displays the <a href="#">Exit Debugger Dialog Box</a> at the ID78K0S-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 tool bar. (Refer to "[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.

**(a) Extension**

Set an extension. Input an extension from the keyboard, or select one from the drop-down list.

**(b) Tab Size**

Select the tab size. Select how many spaces are displayed as a tab code (2, 4, or 8).

**(c) Tab List**

Displays the tab size set for each extension.

**(d) <Add> button**

To change the tab size setting, select **(a) Extension** and **(b) Tab Size** and click the <Add> button.

**(e) <Delete> button**

To delete the tab size setting, select the setting to be deleted from **(c) Tab List** and click <Delete> 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](#).

Begin label	Specifies the symbol of the first address (default: @_cstart)
End label	Specifies the symbol of the end address (default: @_cend)
main() label	Specifies the display start symbol (default _main)

**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.

**(a) 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.

**(b) 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. .

**(a) Size**

Sets the default display size of data if [Adaptive] is specified.

Byte	8-bit display(default)
Word	16-bit display
Double Word	32-bit display

**(b) Radix**

Sets the default radix in which data is to be displayed if [Proper] is specified.

The item selected from this list is also reflected in the subscript for watch data such as array variables (or labels) in the Watch Window (hexadecimal by default), which are registered in the Watch Window after this setting is changed.

Hex	Displays data in hexadecimal numbers (default).
Dec	Displays data in decimal numbers.
Oct	Displays data in octal numbers.
Bin	Displays data in binary numbers.
String	Displays data in character strings.

**(c) 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).

**(d) 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.

---

**Function buttons**

---

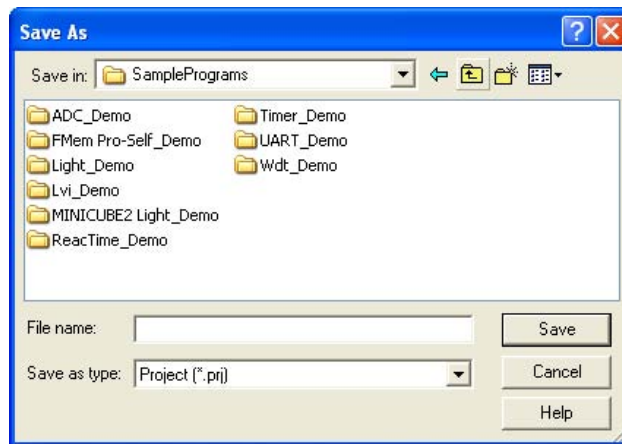
OK	Validates the settings and closes this dialog box.
Cancel	Cancels the changings and closes this dialog box.
Restore	Restores the previous settings before this dialog box was opened.
Help	Displays this dialog box online help files.

## Project File Save Dialog Box

This dialog box is used to save the current debugging environment to a project file. (Refer to ["5.12.1 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
- Function buttons

### 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 in, File name
- (2) Save as type

#### (1) Save in, File name

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.

Up to 257 characters string with a extension can be specified.

#### (2) 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.

## 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.



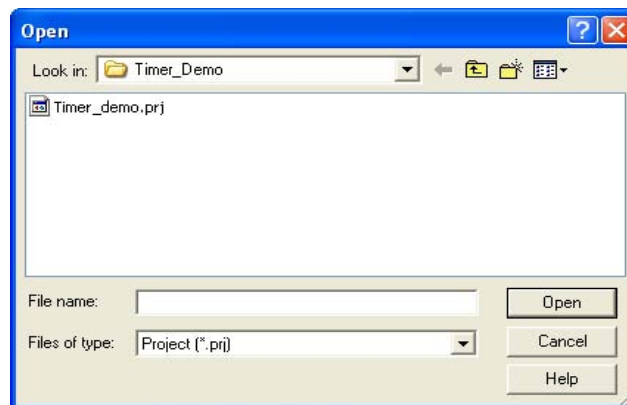
## Project File Load Dialog Box

This dialog box is used to restore the debugging environment to the debugging environment saved to the project file. (Refer to "5.12.1 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 ID78K0S-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
- Function buttons

### Opening



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

### Explanation of each area

- (1) Look in, File name
- (2) Files of type

#### (1) Look in, File name

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.

Up to 257 characters string with a extension can be specified.

#### (2) Files of type

This area is used to specify the extension (\*.prj) of the file to be loaded.

## 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.

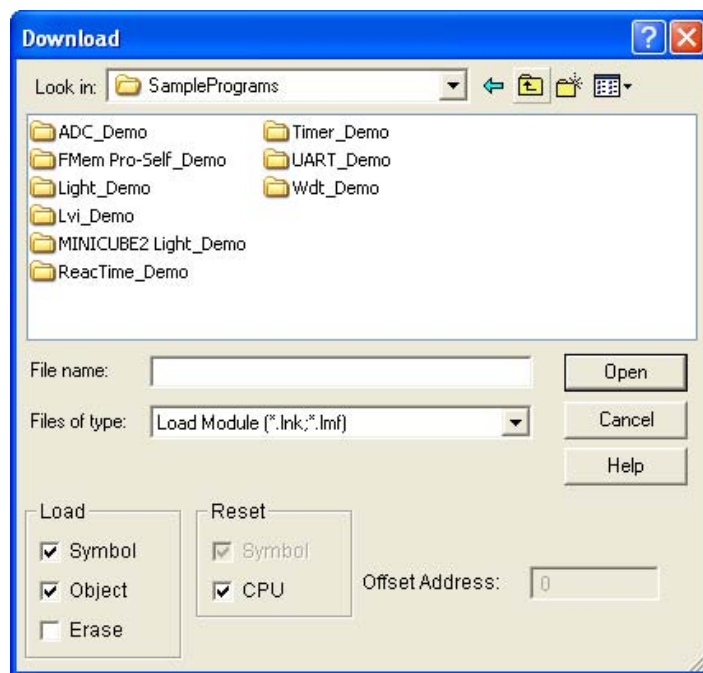
## Download Dialog Box

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".)

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

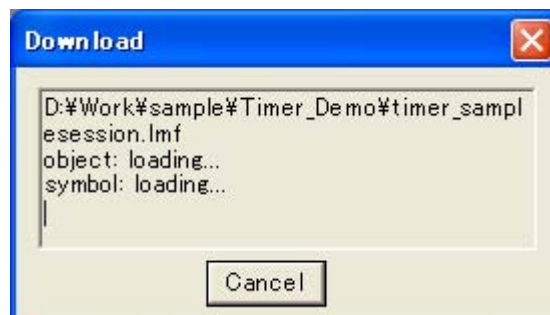
**Caution** If a file other than a load module file is loaded, source debugging cannot be executed.

Figure 6-12 Download Dialog Box



**Remark** The following dialog box appears while downloading and the downloading can be cancelled at any time. This dialog box is closed automatically after completing downloading.

Figure 6-13 The Progress of Download



- Opening
- Explanation of each area
- Function buttons

## Opening



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

## Explanation of each area

- (1) Look in, File name
- (2) Files of type
- (3) Load
- (4) Reset
- (5) Offset Address

### (1) Look in, File name

This area is used to specify a file name. A file name can be directly input from the keyboard, or selected from the list at the upper part of this area.

Up to 257 characters string with a extension can be specified.

### (2) Files of type

This area is used to specify the type (extension) of the file to be downloaded. (Refer to "[Table 5-2 Type of File That Can Be Downloaded](#)".)

**Remark** These are default extensions; other extensions can also be used.

The default extension of the displayed load module can also be specified in the [Debugger Option Dialog Box](#).

### (3) Load

This area is used to set a load condition. This setting is valid only if a file in the load module format is specified.

Symbol	Specifies whether symbol information is read or not. (This setting is valid only if a file in the load module format is specified. )
Object	Specifies whether object information is read or not. (This setting is valid only if a file in the load module format is specified. )
Erase	Specifies whether the contents of the internal flash memory are erased all before download or not.

**(4) Reset**

This area is used to set a reset condition.

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

Symbol	Specifies whether symbol information is reset or not. Fixed to checked status (always reset).
CPU	Specifies whether the CPU is reset or not. (Checked, default.)

**(5) 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 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.

---

**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.

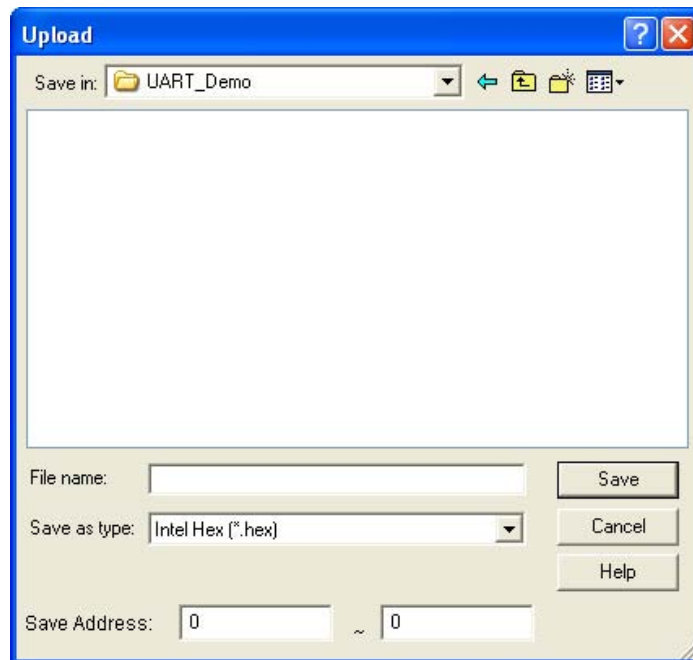
---

## 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 6-14 Upload Dialog Box



- Opening
- Explanation of each area
- Function buttons

---

### Opening

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

---

### Explanation of each area

- (1) Save in, File name
- (2) Save as type
- (3) Save Address

#### (1) Save in, File name

This area is used to specify the file name to be saved. 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.

**(2) Save as type**

This area is used to specify the type (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"](#))

However, if "All (\*.\*)" is selected, the data is saved in the default Intel extended Hex format.

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

**(3) Save Address**

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

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.

---

**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.

## 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](#) and [Mixed display mode \(Source Window\)](#), a number of other operations using [Context menu](#), [Function buttons](#), etc., can be performed in this window.

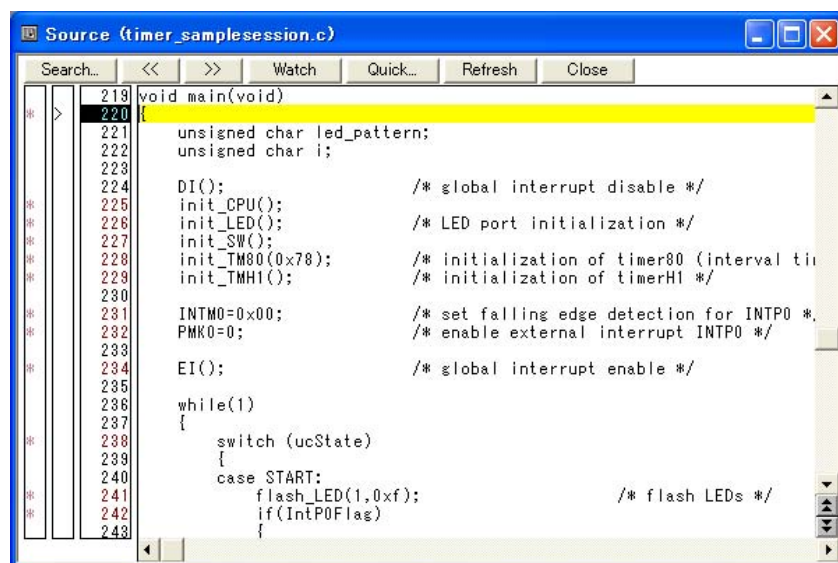
Moreover, there are two statuses, [Active status](#) and [static status](#), for this window. When the window is in the active status, it has the [Trace result with linking window \[IECUBE\]](#) and [Jump function](#). Moreover, the items selected in the window with [Drag & drop function](#) can be used in another window. (Refer to "5.13 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, ID78K0S-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 ID78K0S-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
- Function buttons

## 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
- (2) Current PC mark area
- (3) Line number/address display area
- (4) Source text display 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](#).

**Caution** The program code is displayed only when the symbol information downloaded by the load module is read. 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, which ever has "\*" displayed.)

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 6-5 Event Setting Status (Event Mark)

Mark	Meaning
B (blue)	Software breakpoint is set.
B (black)	Invalid software breakpoint is set. This software breakpoint can be validated on the <a href="#">Software Break Manager</a> .

**(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.

**(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.

**(4) Source text display area**

This area displays source files and text files.

Yellow indicates the current PC line, and red indicates lines where a valid breakpoint is set. In the [Mixed display mode \(Source Window\)](#), source lines are displayed in the regular color.

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](#)")
- [Jump function](#)
- [Drag & drop function](#)
- [Watch 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

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

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

Create Break Event <b>[IECUBE] [MINICUBE+]</b>	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.
Mix	Turns on/off <a href="#">Mixed display mode (Source Window)</a> .

## 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. <b>[IECUBE] [MINICUBE+]</b>
Break when Write to this Variable	Sets a break event that occurs if the selected variable is accessed for write. <b>[IECUBE] [MINICUBE+]</b>
Break when Read from this Variable	Sets a break event that occurs if the selected variable is accessed for read. <b>[IECUBE] [MINICUBE+]</b>
Clear	Deletes a break event corresponding to the selected variable. <b>[IECUBE] [MINICUBE+]</b>
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.
Software Break Point	Sets or deletes a software breakpoint at the cursor position.
Assemble	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position. (Refer to " <a href="#">5.13.2 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	<p>Displays the memory contents starting from the jump destination address specified by the data value at the cursor position. (Refer to "<a href="#">5.13.2 Jump function</a>".)</p> <p>Opens the <a href="#">Memory Window</a>.</p> <p>If an active <a href="#">Memory Window</a> is open, that window is displayed in the forefront (so that it can be manipulated).</p>
--------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Function buttons

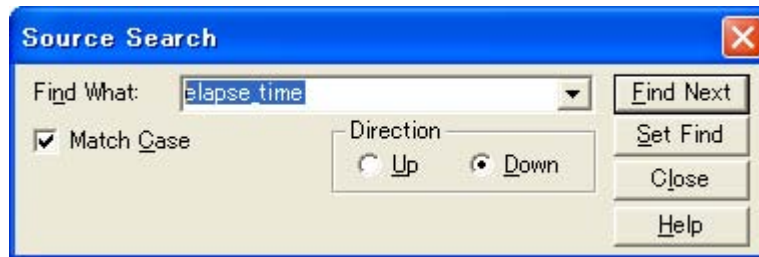
Search...	<p>Opens the <a href="#">Source Search Dialog Box</a> and searches a character string of the source text. If a character string is selected in the source text display area, the Source Search Dialog Box is opened to search the character string.</p> <p>If no character string is selected, the 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 &lt;Stop&gt; 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 &lt;Stop&gt; 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.

## 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
- Function buttons

### 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) Find What
- (2) Match Case
- (3) Direction

#### (1) Find What

This area is used to specify the data to be searched. (Up to 256 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.

#### (2) Match Case

This should be checked to distinguish between uppercase and lowercase.

**(3) 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).

**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.

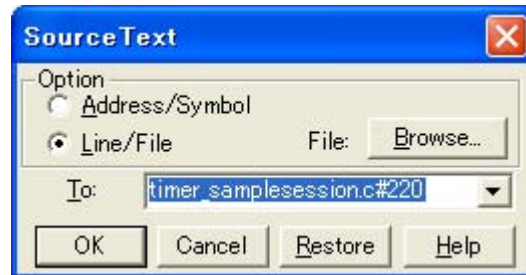
---

## 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](#)
- [Function buttons](#)

---

### Opening

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

---

### Explanation of each area

- (1) [Option](#)
- (2) [To](#)

#### (1) Option

This area is used to select the input mode when the display start position is specified.

##### (a) Address/Symbol

This should be selected to specify by an address (or symbol).

##### (b) Line/File

This should be selected to specify by a line number (or file name). To search the file name, use the <Browse...> button.

**(2) To**

This area is used to specify the file name or address to be displayed.

Up to 16 input histories can be recorded.

- When the **(a) 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 "[Table 5-5 Specifying Symbols](#)".)

Clicking the <OK> button displays the source text so that the source line corresponding to the specified address value can be viewed.

- When the **(b) 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 **[[path name] file name]# line number**.

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 [Debugger Option Dialog Box](#) 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.

## 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.

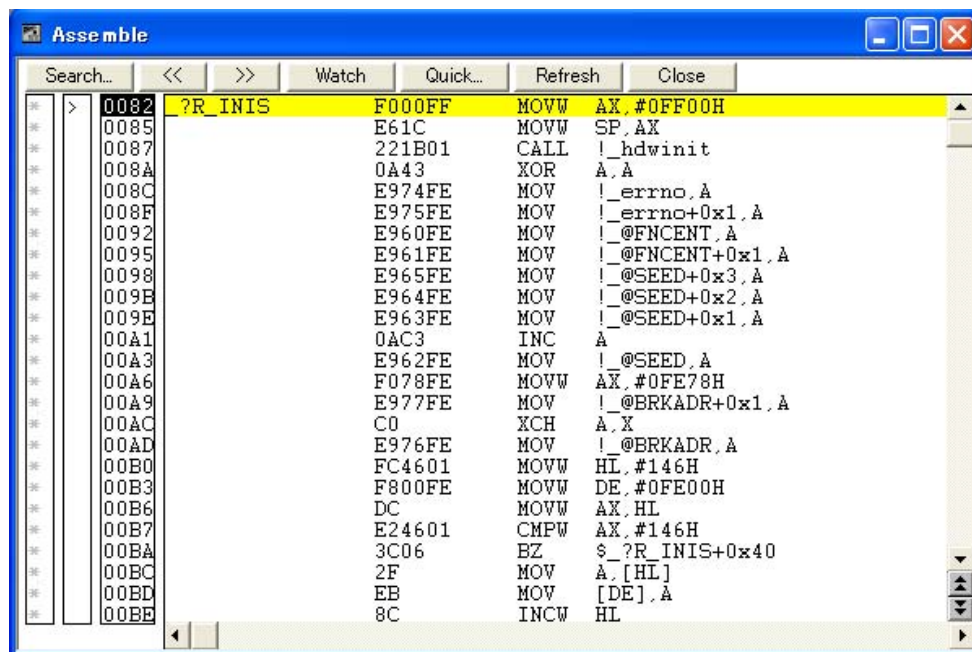


## Assemble Window

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](#), a number of other operations using [Context menu](#), [Function buttons](#), etc., can be performed in this window. When the window is in the active status, it has the [Trace result with linking window \[IECUBE\]](#) and [Jump function](#). Moreover, there are two statuses, [Active status](#) and [static status](#), for this window. Moreover, the items selected in the window with the [Drag & drop function](#) can be used in another window. (Refer to "5.13 Functions Common to Each Window".)

Figure 6-18 Assemble Window



- [Opening](#)
- [Explanation of each area](#)
- [Context menu](#)
- [Function buttons](#)
- [Related operations](#)

### Opening



Click the **Asm** button, or select [Browse] menu -> [Assemble] .

## Explanation of each area

- (1) Point mark area
- (2) Current PC mark area
- (3) Address specification area
- (4) Disassemble display 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.

### (3) Address specification area

This area displays the disassembly start address.

### (4) Disassemble display area

This area displays the labels and code data of addresses, and disassembled mnemonics.

This area displays source files and text files. Yellow indicates the current PC line, and **red** indicates lines where a valid breakpoint is set.

It can be [Online assembly](#) in the mnemonic field.

This area also provides the following functions:

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

## Context menu

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.
Software Break Point	Sets or deletes a software breakpoint at the cursor position.

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 "5.13.2 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).
Memory	Displays the memory contents starting from the jump destination address specified by the data value at the cursor position. (Refer to "5.13.2 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).

## 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.

## 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.

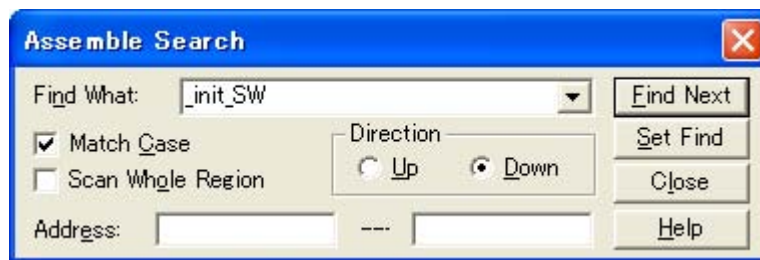
## Assemble Search Dialog Box

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.

Figure 6-19 Assemble Search Dialog Box



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

### 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) [Find What](#)
- (2) [Match Case](#)
- (3) [Scan Whole Region](#)
- (4) [Direction](#)
- (5) [Address](#)

#### (1) Find What

This area is used to specify the data to be searched. (Up to 256 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.

#### (2) Match Case

This should be checked to distinguish between uppercase and lowercase.

**(3) Scan Whole Region**

This should be checked to search the entire specified range.

**(4) 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).

**(5) 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 ["Table 5-5 Specifying Symbols"](#).)

---

**Function buttons**

---

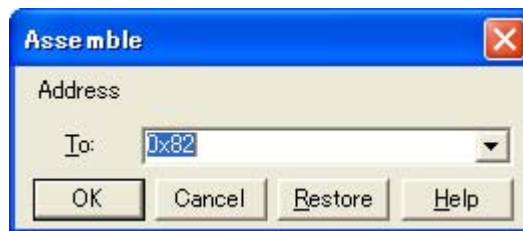
Find Next	Searches the specified data in accordance with a given condition. If the specified character string is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.
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 6-20 Address Move Dialog Box (Example: When Memory Window Is Open)



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

### Opening

When the target window is the current window, select [View] menu -> [Move...].

### Explanation of each area

#### (1) To

This area is used to specify an address.

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.

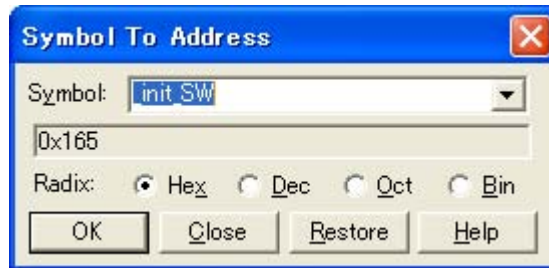
### Function buttons

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 Source Display, Disassemble Display Function".)

Figure 6-21 Symbol To Address Dialog Box



- Opening
- Explanation of each area
- Function buttons

### Opening

Select [View] menu -> [Symbol...].

### Explanation of each area

- (1) Symbol
- (2) Conversion result display area
- (3) Radix

#### (1) 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".)

The default radix for inputting a numeric value is decimal. Up to 16 input histories can be recorded.

To change the contents of this area, click the <OK> button. The conversion result will be displayed in (2) Conversion result display area.

#### (2) 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 (1) 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.



**(3) Radix**

This area is used to select the radix of the data to be displayed in [\(2\) Conversion result display area](#).

Hex	Hexadecimal number (default)
Dec	Decimal number
Oct	Octal number
Bin	Binary number

**Function buttons**

---

OK	If the contents of <a href="#">(1) Symbol</a> have been changed, converts the symbol. After conversion, closes the dialog box if the contents of <a href="#">(1) Symbol</a> 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.

## 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](#).

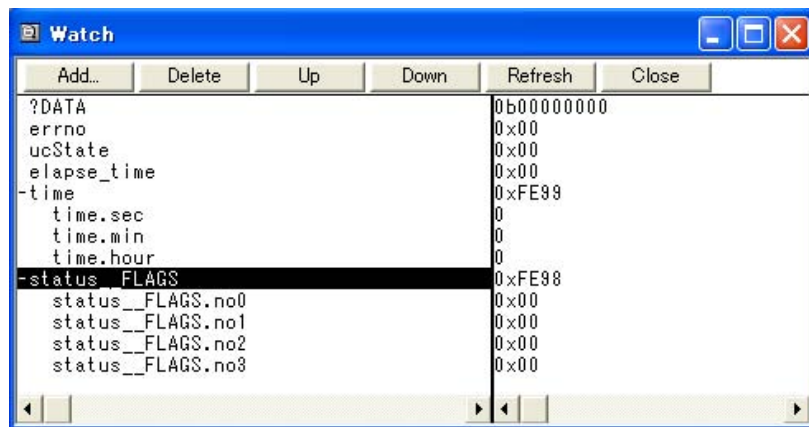
Watch data is registered by clicking the <Watch...> button in the [Source Window](#) or the [Assemble Window](#). (Refer to "5.6.3 Registering and deleting watch data".)

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** A maximum of 10,000 lines can be displayed in the Watch Window.

Figure 6-22 Watch Window



- Opening
- Explanation of each area
- [View] menu (Watch Window-dedicated items)
- Context menu
- Function buttons

## Opening



Click the **Wch** button, or select [Browse] menu -> [Watch].

## Explanation of each area

(1) Left field (symbol name display area)

(2) Right field (data value display/setting area)

### (1) Left field (symbol name display area)

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 as follows when they are double-clicked:

Table 6-6 Watch Window Display Format (Symbol)

First character	Meaning	
+	Array, pointer variable, or structure/union Expanded display is performed by double-clicking "+" (first character changes from "+" to "-").	
	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 "+".
-	Expanded display variable Expanded display is canceled by double-clicking "-" (first character changes from "-" to "+").	

**Remark** If an array has too many variables and takes too long to expand, a warning message is displayed.

Registered watch data changes are performed in the [Change Watch Dialog Box](#) opened by selecting the item to be changed and then selecting [Context menu](#) -> [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.

### (2) Right field (data value display/setting area)

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 -> [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.

The display format is as follows:

Table 6-7 Watch Window Display Format (Data)

Display Data	Contents
Integer	Hexadecimal ( <b>0</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

**Remark** The radix of a data value can be changed on the [Context menu](#) for each variable. The display format of "integer" can be changed on the [Debugger Option Dialog Box](#).

## [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 <b>[IECUBE] [MINICUBE+]</b>	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.
Break when Write to this Variable	Creates a break event that can be accessed for write.
Break when Read from this Variable	Creates a break event that can be accessed for read.
Clear	Deletes a break event corresponding to the selected item.
Event Information <b>[IECUBE] [MINICUBE+]</b>	Displays the event information of the variable selected. If an event is set, the <a href="#">Event Dialog Box</a> is opened.
Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays hexadecimal numbers.

String	Displays character strings.
Proper	Displays 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 in 8-bit units.
Word	Displays in 16-bit units.
Double Word	Displays in 32-bit units.
Adaptive	Displays 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 up one line.
Down	Moves down one line.
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. <b>[IECUBE] [MINICUBE+]</b>
Break when Write to this Variable	Creates a break event that can be accessed for write by using the selected item. <b>[IECUBE] [MINICUBE+]</b>
Break when Read from this Variable	Creates a break event that can be accessed for read by using the selected item. <b>[IECUBE] [MINICUBE+]</b>
Clear	Deletes a break event corresponding to the selected item. <b>[IECUBE] [MINICUBE+]</b>
RRM Setting...	Sets the sampling range of the RRM function). Opens the <a href="#">RRM Dialog Box</a> . <b>[IECUBE] [MINICUBE+]</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. <b>[IECUBE] [MINICUBE+]</b>
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.

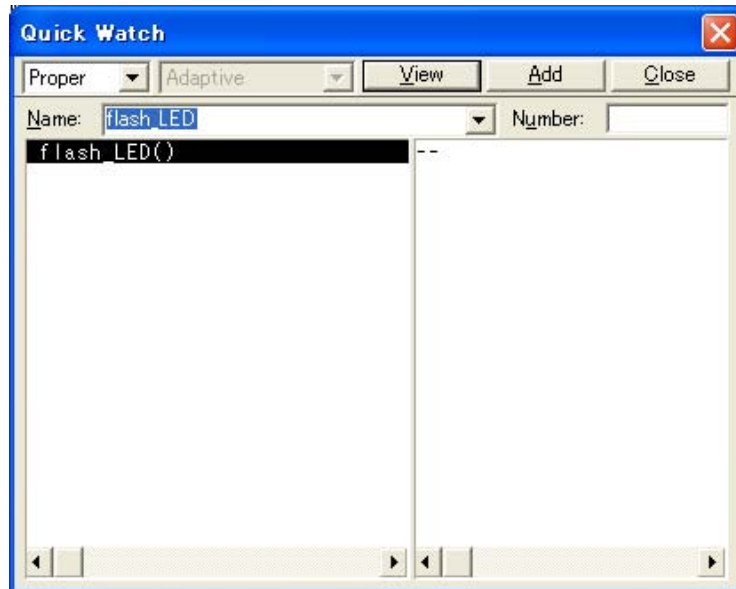
## Function buttons

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.

## Quick Watch Dialog Box

This dialog box is used to temporarily display or change specified watch data. (Refer to "5.6 Watch Function".)

Figure 6-23 Quick Watch Dialog Box



- Opening
- Explanation of each area
- Function buttons

### Opening

When the [Source Window](#) or the [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 radix selection area (upper left field)
- (2) Display size selection area (second upper left field)
- (3) Name
- (4) Number
- (5) Symbol name display area
- (6) Data value display/setting area

**(1) Display radix selection area (upper left field)**

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.

**(2) Display size selection area (second upper left field)**

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.

**(3) 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.

**(4) 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 [Watch Window](#).

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.



**(5) 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 "(1) Left field (symbol name display area)" in the [Watch Window](#).)

This area cannot be edited.

**(6) Data value display/setting area**

This area is used to display and change data values. (Refer to "(2) Right field (data value display/setting area)" in the [Watch Window](#).)

---

**Function buttons**

---

View	Displays the data specified in (3) Name in the field below.
Add	Adds the data specified in (3) Name 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 6-24 Add Watch Dialog Box



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

---

### Opening

---

Select [View] menu -> [Add Watch...], or click the <Add...> button in the [Watch Window](#).

---

### Explanation of each area

---

- (1) [Name](#)
- (2) [Radix](#)
- (3) [Size](#)
- (4) [Number](#)

**(1) Name**

This area is used to specify symbol to be added to the [Watch Window](#).

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.

The input format is as follows:

Table 6-8 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 6-9 How a Variable Is Handled When a 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

Scope Specification	Program name	File name	Function name	Variable Name
<b>var</b>	current	current	current	var

**(2) Radix**

This area is used to select the display radix. (Refer to "[\(1\) Display radix selection area \(upper left field\)](#)" in the [Quick Watch Dialog Box](#).)

**(3) Size**

This area is used to select the display size. (Refer to "[\(2\) Display size selection area \(second upper left field\)](#)" in the [Quick Watch Dialog Box](#).)

**(4) Number**

This area is used to specify the number of data to be displayed. (Refer to "[\(4\) Number](#)" in the [Quick Watch Dialog Box](#).)

## Function buttons

Add	Adds the specified data to the <a href="#">Watch Window</a> . The dialog box remains open.
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 6-25 Change Watch Dialog Box



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

### Opening

When the [Watch Window](#) is the current window, select [View] menu -> [Change Watch...].

### Explanation of each area

- (1) [Name](#)
- (2) [Radix](#)
- (3) [Size](#)
- (4) [Number](#)

#### (1) Name

This area is used to change a symbol name on a line selected in the [Watch Window](#). (Refer to "(1) Name" in the [Add Watch Dialog Box](#).)

#### (2) Radix

This area is used to change the display radix on a line selected in the [Watch Window](#). (Refer to "(1) Display radix selection area (upper left field)" in the [Quick Watch Dialog Box](#).)

**(3) Size**

This area is used to change the display size on a line selected in the [Watch Window](#). (Refer to "(2) Display size selection area (second upper left field)" in the [Quick Watch Dialog Box](#).)

**(4) Number**

This area is used to change the number of data to be displayed on a line selected in the [Watch Window](#). (Refer to "(4) Number" in the [Quick Watch Dialog Box](#).)

---

**Function buttons**

---

Add	Cannot be selected.
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.

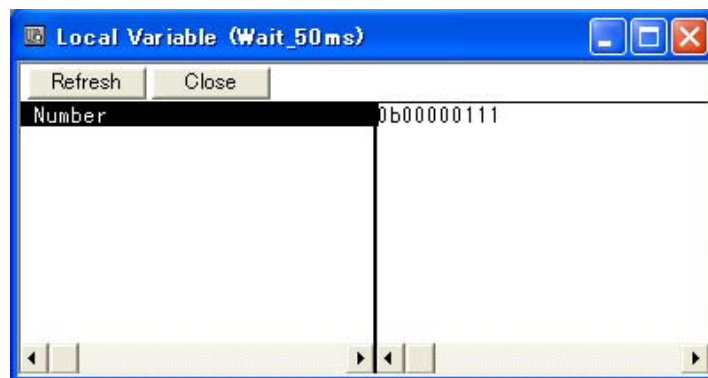
## Local Variable Window

This window is used to display the local variable in the current function and change the local variable values. (Refer to "5.6 Watch Function".)

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 6-26 Local Variable Window



- [Opening](#)
- [Explanation of each area](#)
- [Context menu](#)
- [Function buttons](#)

### Opening



Click the **Loc** button, or select [Browse] menu -> [Local Variable].

### Explanation of each area

- (1) Left field (local variable name display area)
- (2) Right field (local variable value display/setting area)

#### (1) Left field (local variable name display area)

This area displays local variable name. (Refer to "(1) Left field (symbol name display area)" in the [Watch Window](#).)

Auto, Internal Static, and Register variables can be displayed. This area cannot be edited.

**(2) Right field (local variable value display/setting area)**

This area is used to display and change local variable values. (Refer to "(2) Right field (data value display/setting area)" in the [Watch Window](#).)

During user program execution, however, the change cannot be written (if attempted, an error occurs).

**[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).

**Function buttons**

Refresh	Updates the contents of this window with the latest watch data.
Close	Closes this window.



## 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, norec, 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 6-27 Stack Window



- Opening
- Explanation of each area
- [View] menu (Stack Window-dedicated items)
- Context menu
- Function buttons

### Opening



Click the **Stk** button, or select [Browse] menu -> [Stack Trace].

### Explanation of each area

- (1) Left field (stack frame number display area)
- (2) Center field (stack frame contents display area)
- (3) Right field (stack contents display/setting area)

**(1) Left field (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) Center field (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.

(a) If the stack contents consist of a function

They are displayed as follows:

**[program name\$file name#function name (argument list) #line number]**

If this line is double-clicked, the operation will be the same as jumping to the [Source Window](#) of the [Jump function](#) (i.e., the local variable in the function to which execution has jumped will be displayed in the [Local Variable Window](#)). If the function has a local variable, the local variable will be displayed on the next and subsequent lines.

(b) If the stack contents consist of a local variable

Its type and name are displayed. (Refer to "[Table 6-6 Watch Window Display Format \(Symbol\)](#)".)

Note that the internal Static and Register variables are not displayed.

**(3) Right field (stack contents display/setting area)**

This area is used to display or change the stack contents.

(a) If the stack contents are a function

"--" is displayed and the function cannot be changed.

(b) If the stack contents are a local variable

The variable value is displayed. (Refer to "[Table 6-7 Watch Window Display Format \(Data\)](#)".)

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. 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.

**[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 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).
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="#">5.13.2 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="#">5.13.2 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="#">5.13.2 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).

## 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.

## 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 [Jump function](#). (Refer to "5.13 Functions Common to Each Window".)

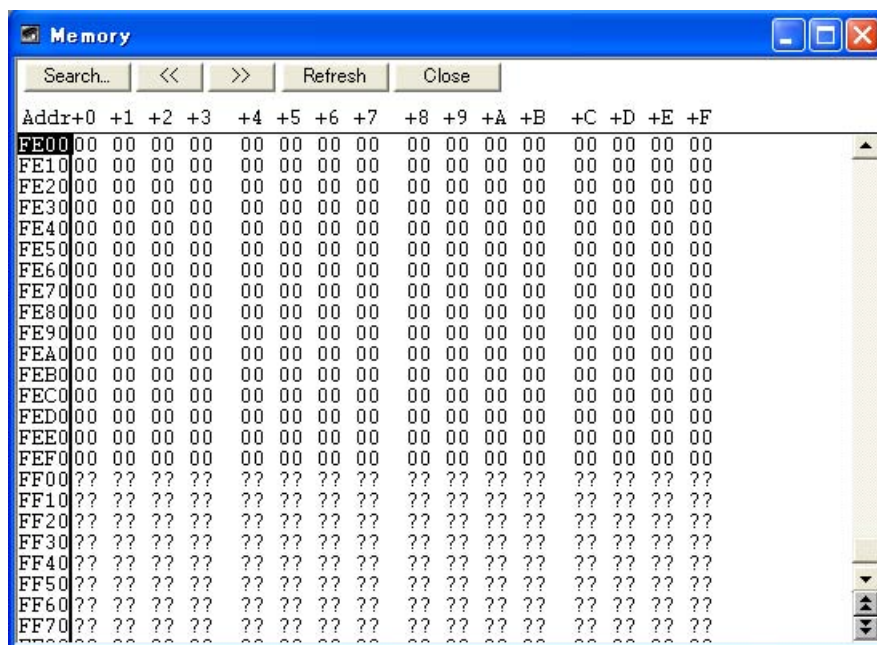
**Remark** The display start position when the this window is opened is as follows:

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 6-28 Memory Window



- Opening
- Explanation of each area
- [View] menu (Memory Window-dedicated items)
- Context menu
- Function buttons

### Opening



Click the **Mem** button, or select [Browse] menu -> [Memory].

## Explanation of each area

(1) Addr

(2) +0 +1 +2....

(3) 0 1 2 3....

### (1) Addr

This area displays memory addresses.

### (2) +0 +1 +2....

This area is used to display and change memory contents.

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. The previous value can be canceled by the ESC key. Up to 256 bytes can be specified at one time.

### (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....

## [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: No display (default)
Little Endian	Displays in little endian (default).
Big Endian	Displays in big endian.

## 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> .
RRM Setting...	Opens the <a href="#">RRM 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: No display (default)

## 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.
Close	Closes this window.

## 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 (2) +0 +1 +2.... in the Memory Window, the specified data is treated as a binary data string, and if the cursor is placed in (3) 0 1 2 3...., 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 6-29 Memory Search Dialog Box



- Opening
- Explanation of each area
- Function buttons

### 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) Find What
- (2) Unit
- (3) Scan Whole Region
- (4) Direction
- (5) Address

**(1) Find What**

This area is used to specify the data to be searched.

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.

(a) When searching in (2) +0 +1 +2....

Up to 16 data items can be specified. Delimit each data with a "blank character".

(b) When searching in (3) 0 1 2 3....

Up to 256 characters can be specified. A "blank character" in the data is treated as a blank character.

**(2) Unit**

This area is used to specify the number of bits of the data to be searched in (2) +0 +1 +2.....

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.

**(3) Scan Whole Region**

This should be checked to search the entire specified range.

**(4) 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).

**(5) 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 "Table 5-5 Specifying Symbols".)

**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 6-30 Memory Fill Dialog Box



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

### Opening

Select [Edit] menu -> [Memory] -> [Fill...].

### Explanation of each area

- (1) [Address](#)
- (2) [fill code](#)

#### (1) Address

This area is used to specify the memory address range whose contents are filled.

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"](#).)

#### (2) fill code

This area is used to specify the data (fill code) used when filling the range specified in (1) [Address](#). Up to 16 binary data strings (byte data strings) can be specified. Delimit each data with a "blank character".

## 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.

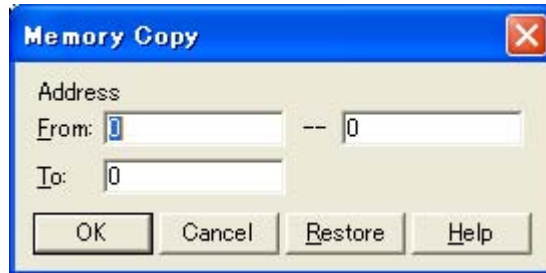
---

## 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 6-31 Memory Copy Dialog Box



- Opening
- Explanation of each area
- Function buttons

---

### Opening

Select [Edit] menu -> [Memory] -> [Copy...].

---

### Explanation of each area

#### (1) 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](#)".)

##### (a) From

Specify the address range (start address -- end address) of the copy source.

##### (b) To

Specify start address of the copy destination.

## Function buttons

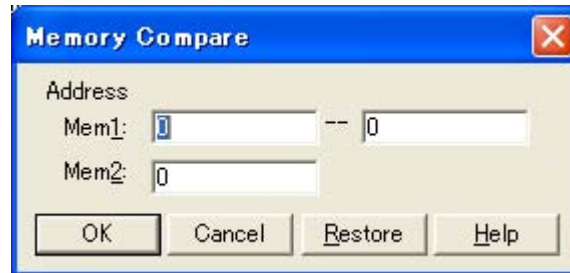
---

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 6-32 Memory Compare Dialog Box



- Opening
- Explanation of each area
- Function buttons

### Opening

Select [Edit] menu -> [Memory] -> [Compare...].

### Explanation of each area

#### (1) 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 "[Table 5-5 Specifying Symbols](#)".)

##### (a) Mem1

Specify the address range (start address -- end address) of the comparison source.

##### (b) Mem2

Specify the start address of the comparison destination.

---

## Function buttons

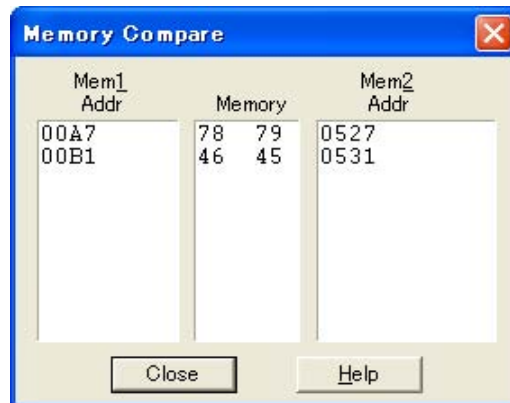
---

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 6-33 Memory Compare Result Dialog Box



- [Explanation of each area](#)
- [Function buttons](#)

### 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.

##### (a) Mem1 Addr

Displays a comparison source address in which a difference has been found.

##### (b) Memory

Displays the data in which a difference has been found (Left: Comparison source data, Right: Comparison destination data).

##### (c) Mem2 Addr

Displays the comparison destination address at which a difference has been found.

### Function buttons

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

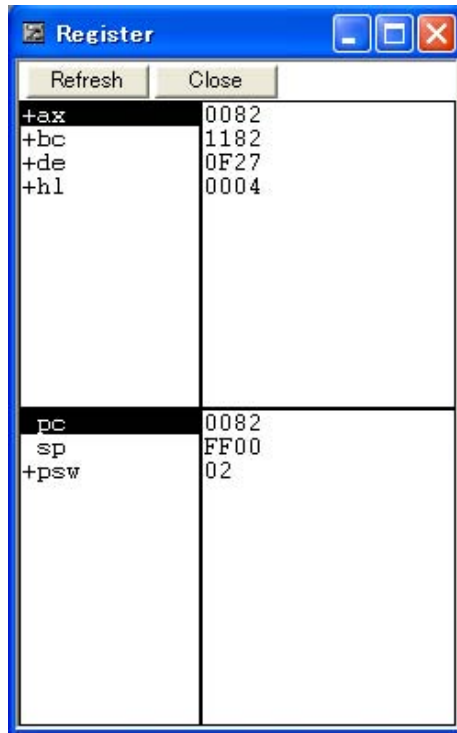
## Register Window

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](#).

Figure 6-34 Register Window



- Opening
- Explanation of each area
- [View]menu (Register Window-dedicated items)
- Context menu
- Function buttons

### Opening



Click the **Reg** button, or select [Browse] menu -> [Register].

### Explanation of each area

- (1) Upper field (general-purpose registers display area)
- (2) Lower field (control registers display area)



**(1) Upper field (general-purpose registers display area)**

This area is used to display and change the general-purpose registers.

Register 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.

During user program execution, however, the change cannot be written (if attempted, an error occurs).

**(2) Lower field (control registers display 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 "+").

Register 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.

During user program execution, however, the change cannot be written (if attempted, an error occurs).

**[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 the selected line in binary numbers.
Oct	Displays the selected line in binary numbers.
Dec	Displays the selected line in decimal numbers.
Hex	Displays the selected line in hexadecimal numbers (default).
Absolute Name	Displays register names as absolute names.
Function Name	Displays register names as function names (default).

## Function buttons

---

Refresh	Updates the contents of the window with the latest data.
Close	Closes this window.

## 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.

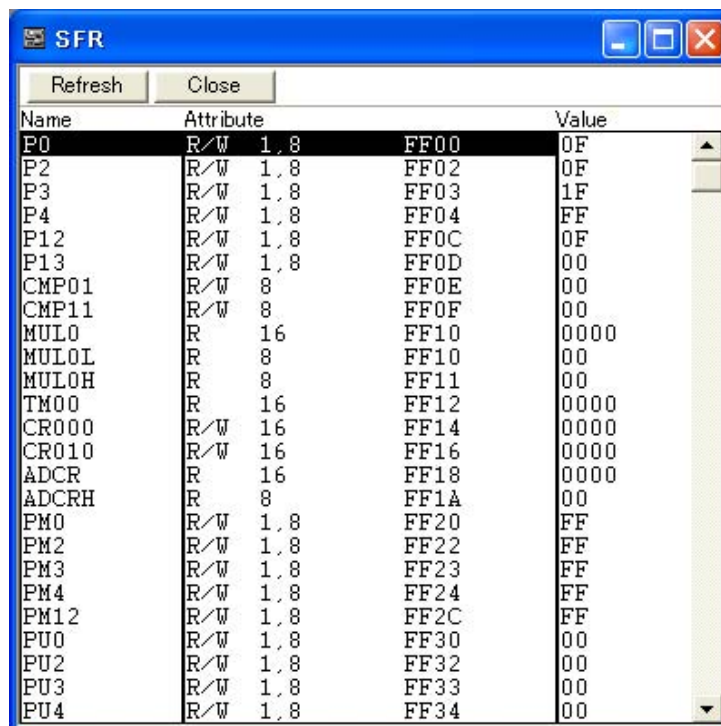
**Caution** However, that the values of read-only SFR and I/O ports cannot be changed. In addition, the SFR and I/O ports that cause the device to operate when they are read are read-protected and therefore cannot be read. To read these registers, select a register, and select and execute [Compulsion Read] from the [Context menu](#).

**Remark** 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

Figure 6-35 SFR Window



Name	Attribute	Value
P0	R/W 1, 8	FF00 0F
P2	R/W 1, 8	FF02 0F
P3	R/W 1, 8	FF03 1F
P4	R/W 1, 8	FF04 FF
P12	R/W 1, 8	FF0C 0F
P13	R/W 1, 8	FF0D 00
CMP01	R/W 8	FF0E 00
CMP11	R/W 8	FF0F 00
MUL0	R 16	FF10 0000
MUL0L	R 8	FF10 00
MUL0H	R 8	FF11 00
TM00	R 16	FF12 0000
CR000	R/W 16	FF14 0000
CR010	R/W 16	FF16 0000
ADCR	R 16	FF18 0000
ADCRH	R 8	FF1A 00
PM0	R/W 1, 8	FF20 FF
PM2	R/W 1, 8	FF22 FF
PM3	R/W 1, 8	FF23 FF
PM4	R/W 1, 8	FF24 FF
PM12	R/W 1, 8	FF2C FF
PU0	R/W 1, 8	FF30 00
PU2	R/W 1, 8	FF32 00
PU3	R/W 1, 8	FF33 00
PU4	R/W 1, 8	FF34 00

- Opening
- Explanation of each area
- [View] menu (SFR Window-dedicated items)
- Context menu
- Function buttons

## Opening



Click the **SFR** button, or select [Browse] menu -> [SFR].

## Explanation of each area

- (1) Name
- (2) Attribute
- (3) Value

### (1) 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.

### (2) Attribute

This area displays the attributes of SFR and I/O ports.

This area displays the read/write attributes, access types, and displays and absolute addresses 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.

**(3) Value**

This area is used to display and change the contents of a SFR and I/O port.

The contents are displayed differently as follows, depending on the attribute:

Black Display	Read only or read/write
--	Write only
**	Value changes if read

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. The previous value can be canceled by the ESC key.

During user program execution, however, the change cannot be written (if attempted, an error occurs).

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 [Context menu](#) -> [Compulsion Read].

**[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 <a href="#">(2) Attribute</a> .
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> .

## Context menu

Move...	Opens the <a href="#">Address Move Dialog Box</a> .
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 (2) <a href="#">Attribute</a> .
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> .

## Function buttons

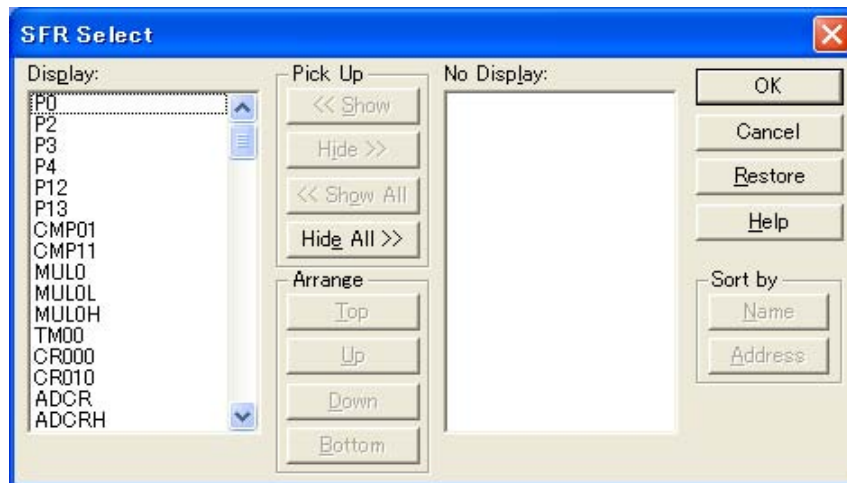
Refresh	Updates the contents of this window with the latest watch data.
Close	Closes this window.

## SFR Select Dialog Box

This dialog box is used to select SFR and I/O ports that are not 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 6-36 SFR Select Dialog Box



- Opening
- Explanation of each area
- Function buttons

### Opening

When the [SFR Window](#) is the current window, select [View] menu -> [Select...].

### Explanation of each area

- (1) Display, Pick Up, No Display
- (2) Arrange
- (3) Sort by

**(1) Display, Pick Up, No Display**

This area is used to select SFR or I/O ports that are displayed in the [SFR Window](#), and those that are not.

**(a) Display**

The SFR or I/O ports displayed in the [SFR Window](#).

**(b) No Display**

The SFR or I/O ports not displayed in the [SFR Window](#).

**(c) Pick Up**

The following buttons are used to change SFR or I/O ports to be displayed in the [SFR Window](#).

Two or more registers can be moved by clicking any of the above buttons while holding down the Ctrl or Shift key.

<< Show	Moves SFR or I/O ports selected from <a href="#">(b) No Display</a> list to <a href="#">(a) Display</a> .
Hide >>	Moves SFR or I/O ports selected from <a href="#">(a) Display</a> list to <a href="#">(b) No Display</a> .
<< Show All	Moves all SFR or I/O ports to <a href="#">(a) Display</a> .
Hide All >>	Moves all SFR or I/O ports to <a href="#">(b) No Display</a> .

**(2) Arrange**

The following buttons are used to change the display sequence in [\(a\) Display](#).

Top	Moves the selected SFR or I/O port to the top of the list.
Up	Moves the selected SFR or I/O port one line up.
Down	Moves the selected SFR or I/O port one line down.
Bottom	Moves the selected SFR or I/O port to the bottom of the list.

**(3) Sort by**

The following buttons are used to change the display sequence in [\(b\) No Display](#).

Name	Displays in alphabetical order.
Address	Displays in address order.

**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 6-37 Add I/O Port Dialog Box



- Opening
- Explanation of each area
- Function buttons

### Opening

Select [Option] menu -> [Add I/O Port...].

### Explanation of each area

- (1) I/O Port List
- (2) Name
- (3) Address
- (4) Access
- (5) Read / Write

#### (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 [Function buttons](#).

#### (2) Name

This area is used to specify an I/O port name to be added (up to 15 characters long).

**(3) 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 "[Table 5-5 Specifying Symbols](#)".)

The address that can be set in this area is either a Target area address or SFR area address.

**(4) 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

**(5) 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 not checked (i.e., the I/O port can be both read and written).

**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.
Add	Adds an I/O port of the specified address.
Change	Changes the setting of the I/O port selected in <a href="#">(1) I/O Port List</a> .
Delete	Deletes the I/O port selected in <a href="#">(1) I/O Port List</a> .

## Trace View Window

[IECUBE]

This window used to display the trace results. (Refer to "5.9 Trace Function [IECUBE]").

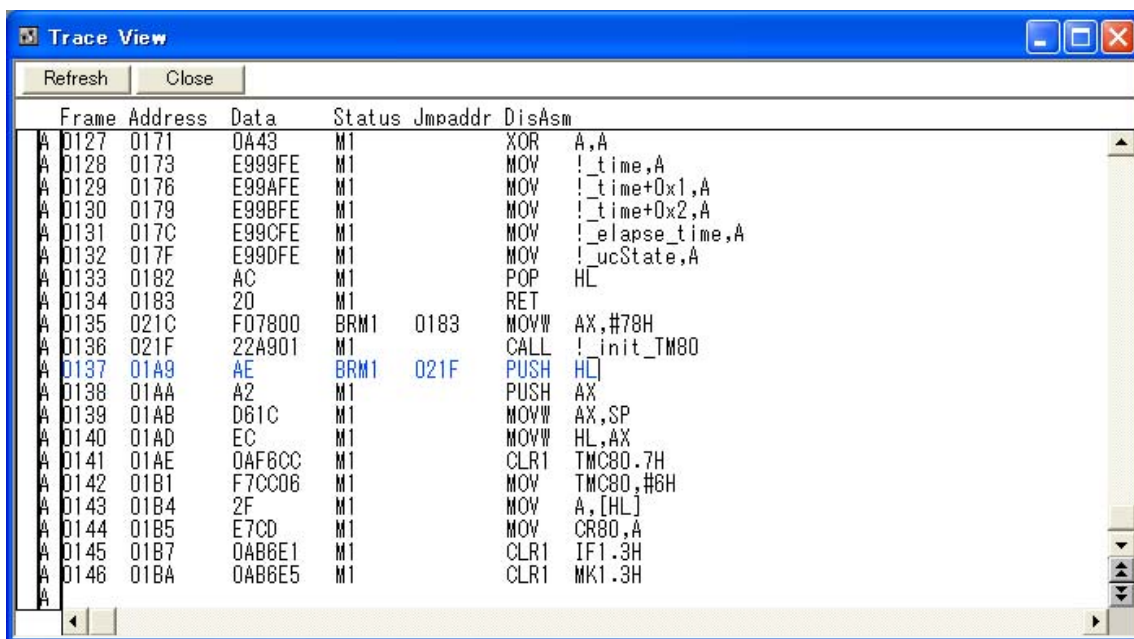
Display updates are performed during breaks or during step execution.

This window has [Mixed display mode \(Trace Window\)](#). Also, It has "5.13.4 Trace result with linking window [IECUBE]".

[Context menu](#), A number of other operations using [Function buttons](#), etc., can be performed in this window.

**Caution** Before starting execution, clears the trace data.

Figure 6-38 Trace View Window



- Opening
- Explanation of each area
- [View] menu (Trace View Window-dedicated items)
- Context menu
- Function buttons

## Opening



Click the **TrW** button, or select [Browse] menu -> [Trace].

## Explanation of each area

(1) Trace mode display area

(2) Trace result display area

### (1) Trace mode display area

This area displays the type of tracer mode.

Mark	Meaning
T	Delay trigger frame

### (2) 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 frame number.

#### (b) Fetch access display area

This area displays the result of fetching the program.

Address	Displays the fetched address.	
Data	Displays the fetched data.	
Status	BRM1	Fetching of first byte of first instruction after branch If the fetch address is the start of the symbol, the first line is highlighted in blue.
	M1	Fetching of first byte of instruction If the fetch address is the start of the symbol, the first line is highlighted in blue.

#### (c) Jmpaddress

This area is used to display the last address of the branch source. (This item is displayed only when "BRM1" is displayed in the "Status" column.)

#### (d) DisAsm

This area displays the disassemble results. (This item is displayed only when "BRM1" is displayed in the "Status" column.)

Global symbols are displayed for addresses that have been accessed and frames in which only data is displayed.

## [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> .
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="#">5.13.4 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> .

## Context menu

Move...	Moves the display position. Opens the <a href="#">Trace Move Dialog Box</a> .
Select...	Selects the contents to be displayed. Opens the <a href="#">Trace Data Select Dialog Box</a> .
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 Trace View Window with the following windows: (Refer to " <a href="#">5.13.4 Trace result with linking window [IECUBE]</a> ".)
Source Text	Links the <a href="#">Source Window</a> .
Assemble	Links the <a href="#">Assemble 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="#">5.13.2 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="#">5.13.2 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="#">5.13.2 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).

## Function buttons

---

Refresh	Updates the contents of the window with the latest data.
Close	Closes this dialog box.

## Trace Data Select Dialog Box

[IECUBE]

This dialog box is used to select items to be displayed in the [Trace View Window](#). (Refer to "[5.9 Trace Function \[IECUBE\]](#)".)

Figure 6-39 Trace Data Select Dialog Box



- Opening
- Explanation of each area
- Function buttons

### Opening

When the [Trace View Window](#) is the current window, select [View] -> [Select...] menu.

### Explanation of each area

- (1) Item
- (2) Radix

#### (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 "[\(2\) Trace result display area](#)".)

Frame	Frame Field
Instruction Fetch Address	Address ((b) <a href="#">Fetch access display area</a> ) Field
Instruction Fetch Data	Data ((b) <a href="#">Fetch access display area</a> ) Field
Instruction Fetch Status	Status ((b) <a href="#">Fetch access display area</a> ) Field
Jump Address	Jump Address 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 ((b) <a href="#">Fetch access display area</a> ) Field
------------------------	-------------------------------------------------------------

Hex	Displays hexadecimal numbers. (Default)
Dec	Displays decimal numbers.
Oct	Displays octal numbers.
Bin	Displays binary numbers.

**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.

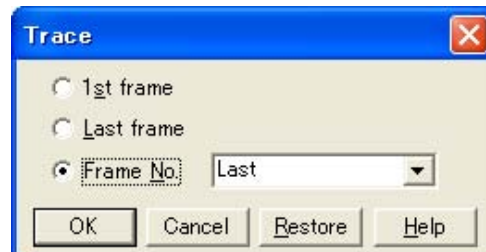


## Trace Move Dialog Box

[IECUBE]

This dialog box is used to specify the position from which displaying the [Trace View Window](#) is started. (Refer to "5.9 Trace Function [IECUBE]".)

Figure 6-40 Trace Move Dialog Box



- Opening
- Explanation of each area
- Function buttons

### 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.

1st frame	Moves the display start position to a first frame of trace data.
Last frame	Moves the display start position to the last frame of trace data.
Frame No.	<p>Moves the display start position to the specified frame number. (Refer to "Table 6-10 Frame Number Specification Format".)</p> <p>Default is Last.</p> <p>The default radix for inputting a numeric value is decimal.</p> <p>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.</p>

Table 6-10 Frame Number Specification Format

Specification	Abbreviation	Contents
+numeric value	None	Moves backward (downward on screen) the display start position from the frame at the cursor by the specified number of frames (numeric value).
-numeric value	None	Moves forward (upward on screen) the display start position from the frame at the cursor by the specified number of frames (numeric value).
First	S	Same as " 1st frame "
Last	L	Same as " Last frame "

## Function buttons

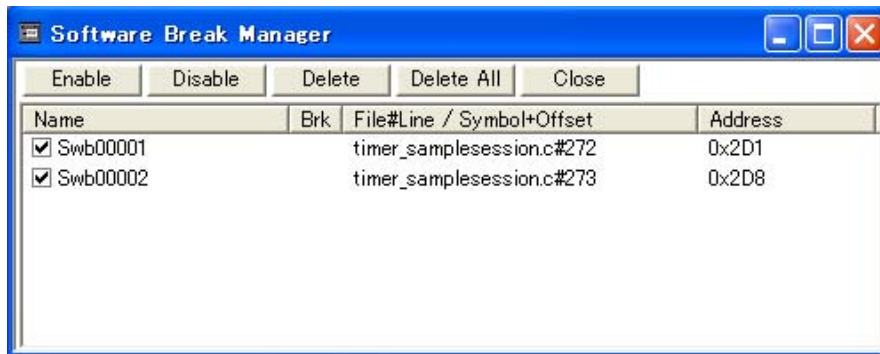
OK	Starts trace display from the specified position.
Cancel	Closes this dialog box.
Restore	Restores the input data to the original status.
Help	Displays this dialog box online help files.

## 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 [IECUBE] [MINICUBE+]".)

Software breakpoints cannot be set in this window; they can be set in the [Source Window](#) or the [Assemble Window](#). (Refer to "5.4.2 Breakpoint setting".)

Figure 6-41 Software Break Manager



- Opening
- Explanation of each area
- Function buttons

### Opening

Select [Event] menu -> [Software Break Manager].

### Explanation of each area

- (1) Name
- (2) Brk
- (3) File#Line / Symbol+Offset
- (4) Address

**(1) Name**

This area displays the names of registered events, and the check boxes that indicate whether each event is enabled or disabled.

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.

When an event is enabled, the check box is checked. To be disabled, the check box is not checked.

Furthermore, the name jumps to the [Source Window](#) by double-clicking an event name if the event name corresponds to the source line, whereas the name jumps to the [Assemble Window](#) if it does not correspond to the source line.

**Remark** By clicking "Name" (on the label), the character strings of the displayed items can be compared and sorted lexicographically (in alphabetical order). Whether the character strings are compared or sorted in ascending or descending order can be alternately selected by clicking the mouse.

**(2) Brk**

The ">" 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).

**(3) File#Line / Symbol+Offset**

This area displays the location at which a software break event was set as follows:

- [Program]\$file name#line number (If the event corresponds to the source line.)
- [Program]\$file name#symbol+offset (If the event does not correspond to the source line.)

Events are evaluated based on this when a symbol is re-downloaded.

**Remark** By clicking "File#Line/Symbol+Offset" (on the label), the character strings of the displayed items can be compared and sorted lexicographically (in alphabetical order). Whether the character strings are compared or sorted in ascending or descending order can be alternately selected by clicking the mouse.

**(4) Address**

This area displays the address at which a software break event is set.

**Remark** By clicking "Address" (on the label), the numeric values of the displayed items can be compared and sorted. Whether the values are compared or sorted in ascending or descending order can be alternately selected by clicking the mouse.

---

## 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

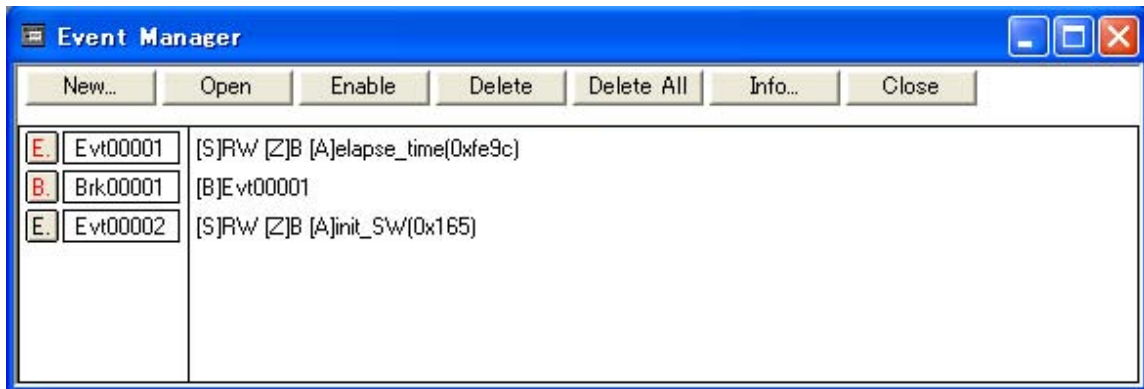
[IECUBE] [MINICUBE+]

This window is used to manage event conditions. This window allows display, enabling/disabling, and deletion of the [Various Event Conditions](#). (Refer to "5.10 Event Function [IECUBE] [MINICUBE+]".)

A number of other operations using [Context menu](#), [Function buttons \(Related event function\)](#), etc., can be performed in this window.

The event icon is the jump pointer of the [Jump function](#).

Figure 6-42 Event Manager (In Detailed Display Mode)



- [Opening](#)
- [Explanation of each area](#)
- [\[View\]menu \(Event manager-dedicated items\)](#)
- [Context menu](#)
- [Function buttons \(Related event function\)](#)

### 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.

(a) In list displayed



Displays event icon (refer to ["Table 5-12 Event Icon"](#)).

The event icon is the jump pointer (refer to ["5.13.2 Jump function"](#)).

(b) In detailed display

E. Evt00001	[S]RW [Z]B [A]elapse_time(0xfe9c)
B. Brk00001	[B]Evt00001
E. Evt00002	[S]RW [Z]B [A]init_Sw(0x165)

Details of event contents are displayed by using the following key information as a separator.

Table 6-11 Separator for Displaying Event Details

Key Information	Contents
<b>Event condition</b>	
[S]	Status condition
[Z]	Access size condition
[A]	Address condition Symbol or expression: (actual address)
[D]	Data condition Symbol or expression: (actual address)
[M]	Mask condition
<b>Break condition</b>	
[B]	Break condition

## [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	List display (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	Displays a list (default).
Memory	Displays the memory contents from the position of the selected event, which is used as the jump destination address. (Refer to " <a href="#">5.13.2 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).


## Function buttons (Related event function)

Describes the all function buttons the related event dialogs (the Event Manager, the [Event Dialog Box](#) and the [Break Dialog Box](#).)

OK	<p>(<a href="#">Event Dialog Box</a>) Automatically registers the event condition being edited, if any, and closes this dialog box.</p> <p><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.</p>
	<p>(Other than above dialog boxes) 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.</p>



New...	(Event Manager) Opens the dialog box to create new event condition. By clicking each button of <Event...> and <Break...>, 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.
	(Other than above dialog boxes) Newly creates an event condition in this dialog box. An event condition name is automatically created and a new event condition is prepared.
Set	(Event Dialog Box) Registers the various event condition. 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.
	(Other than above dialog boxes) Registers 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 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/Abort	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	Opens the <a href="#">Break Dialog Box</a> .
Manager	Opens the <a href="#">Event Manager</a> .
Add Event...	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 pressed.
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/Delete	Deletes the selected event condition. When an event condition is to be deleted, an error occurs and the event condition cannot be deleted if the event is used as a various event condition.
Delete All	Deletes all event conditions.
Expand >>>/ Shrink <<<	Turns on (Expand>>>) or off (Shrink<<<) display of the event manager area. The size of the dialog box is expanded or reduced. The size of the dialog box is expanded or reduced.
Add	The event condition selected in Event Manager area add to setting area with a focus.

Info...	<p>Opens the Select Display Information dialog box. This dialog box is used to change the display mode and rearrange event names.</p> <p>Figure 6-43 Select Display Information Dialog Box</p>  <p>&lt;Sort by Name&gt;.. Sorts events into name order.        &lt;Sort by Kind&gt;..... Sorts events into type order.        &lt;Unsort&gt;..... Displays events in the order in which they have been registered without sorting the events.        &lt;Detail&gt;..... Sets the detailed display mode.        &lt;Overview&gt;..... Sets the list display mode.        &lt;Cancel&gt;..... Closes this dialog box (same as ESC key).</p>
Other...	<p>Opens the dialog box for selecting the event type. By clicking each button of &lt;Event...&gt; and &lt;Break...&gt;, 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.        &lt;Manager...&gt; Opens the Event Manager.        &lt;Cancel&gt; Closes the dialog box to create new event condition.</p>

## Event Dialog Box

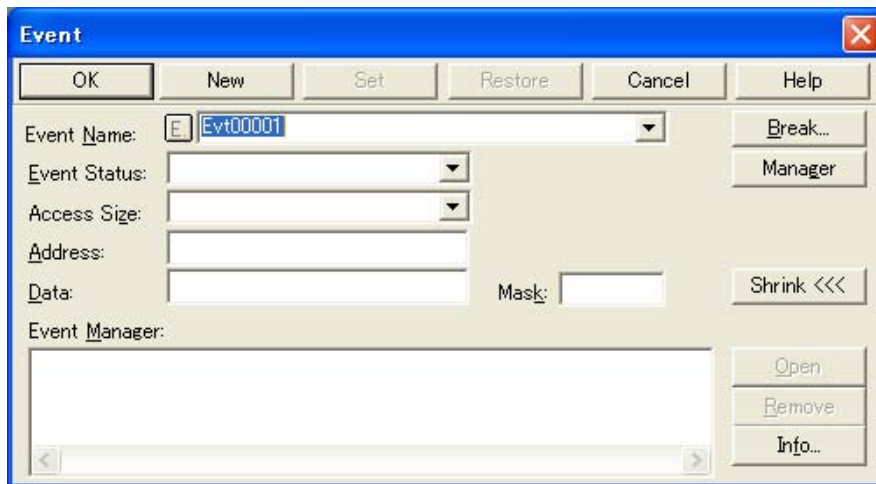
[IECUBE] [MINICUBE+]

This dialog box is used to register and display event conditions. (Refer to "5.10 Event Function [IECUBE] [MINICUBE+]".)

Registering 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 "5.10.4 Number of enabled events for each event condition".)

Figure 6-44 Event Dialog Box




- [Opening](#)
- [Explanation of each area](#)
- [Function buttons \(Related event function\)](#)

## 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 this button, or select [Event] menu -> [Event].

### In select mode

If the <OK> button is pressed 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.

- In each various event setting dialog box, click the <Add Event... > button.

## Explanation of each area

- (1) Event Name
- (2) Event Status
- (3) Access Size
- (4) Address
- (5) Data, Mask
- (6) Event Manager

### (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 ("Table 5-12 Event Icon"). The gray E. mark indicates that the event condition is being edited and has not been registered yet.

### (2) Event Status

This area is used to set 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 (3) Access Size and (5) Data, Mask ).

The status conditions that can be specified are listed below.

Table 6-12 Status Condition

Status	Abbreviation	Meaning
<b>Access event</b>		
R/W	RW	Memory read/write
Read	R	Memory read
Write	W	Memory write

**(3) Access Size**

This area is used to set 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.

Byte	Detects data condition with 8-bit width (only during 8-bit access).												
No Condition	Does not detect access size (nothing can be input to the Data area).												
Bit	<p>Detects data condition with 1-bit width (only during 8-bit access). In this case, a data condition is detected with 1-bit width. Because of the operation of the In-circuit emulator, access to a bit is not directly detected; the ID78K0S-QB detects a dummy bit access by internally setting address conditions and data conditions as follows:</p> <table><tr><td><b>Input example:</b></td><td></td><td><b>Setting of trace search:</b></td></tr><tr><td>Address: FE20.1</td><td>----&gt;</td><td>Address: FE20</td></tr><tr><td>Data: 1</td><td></td><td>Data: 00000010B</td></tr><tr><td></td><td></td><td>Mask: 11111101B</td></tr></table> <p>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]. When data is written to a bit, all the 8 bits are read/written. If read or read/write is specified as the status, an event occurs if a read operation is performed at this time if the value of the specified [Address.bit] matches.</p>	<b>Input example:</b>		<b>Setting of trace search:</b>	Address: FE20.1	---->	Address: FE20	Data: 1		Data: 00000010B			Mask: 11111101B
<b>Input example:</b>		<b>Setting of trace search:</b>											
Address: FE20.1	---->	Address: FE20											
Data: 1		Data: 00000010B											
		Mask: 11111101B											

**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

**(4) Address**

This area is used to specify an address condition.

A specific point (matches the specified address) can be specified.

Settable range of address condition (Event):  $0 \leq \text{address value} \leq 0\text{xFFFFF}$

The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression (refer to [Specifying Symbols](#) ).

**(5) Data, Mask**

This area is used to set data conditions.

A specific point (matches the specified address) can be specified.

The default radix for inputting a numeric value is hexadecimal.

The settable range differs as follows depending on the access size condition specified in [\(3\) Access Size](#).

Table 6-13 Settable Range of Data Condition

Access Size	Settable range
Byte	0 <= data value <= 0xFF 0 <= mask value <= 0xFF
Bit	Data value = 0 or 1 Mask value = Cannot be specified.

**(a) Data**

The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression (refer to [Specifying Symbols](#)).

**(b) Mask**

Set a mask value for the data value.

When a mask is set, the data value for the bit whose mask value is 1 may be 0 or 1.

**Example 1:**

Data	0x40
Mask	0x0F

With this setting, addresses 0x40 to 0x4F satisfy the condition.

**Example 2:**

Data	0x40
Mask	0x11

With this setting, addresses 0x40, 0x41, 0x50, and 0x51 satisfy the condition.

**(6) Event Manager**

This area is used to display the list of the events registered. (Refer to ["Table 5-12 Event Icon"](#), ["\(4\) Manipulation in event manager area"](#).)

## Function Buttons

Refer to ["Function buttons \(Related event function\)"](#) in the [Event Manager](#).

## Break Dialog Box

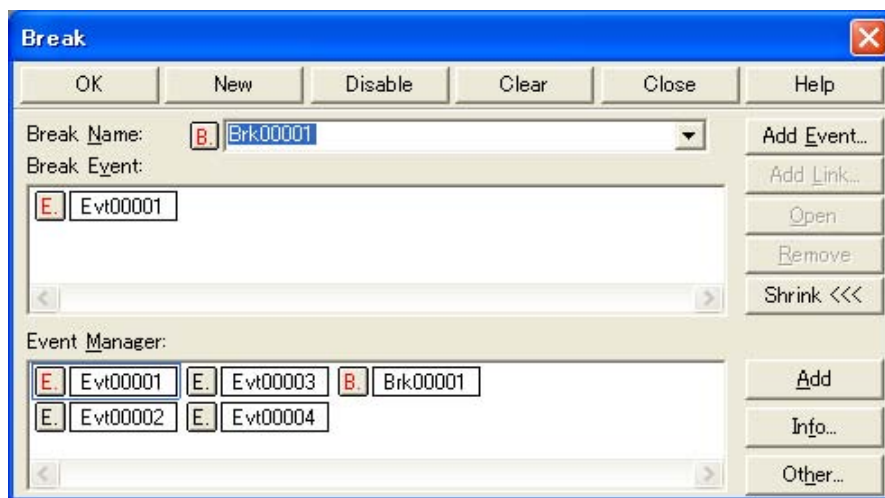
[IECUBE] [MINICUBE+]

This dialog box is used to register; set, and display break event conditions. (Refer to "5.10 Event Function [IECUBE] [MINICUBE+]", "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 "5.10.4 Number of enabled events for each event condition".)

Figure 6-45 Break Dialog Box



- Opening
- Explanation of each area
- Function buttons (Related event function)

### Opening



Click the **Brk** button, or select [Event] menu -> [Break...].

### Explanation of each area

- (1) Break Name
- (2) Break Event
- (3) Event Manager

**(1) Break Name**

This area is used to set a break 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 5-12 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.

**(2) Break Event**

This area is used to set an event condition for break.

The number of event conditions that can be set in this area is as follows:

Table 6-14 Number of Events Settable in Break Condition Setting Area

Connected IE	Total (execution / access)
[IECUBE] [MINICUBE+]	1 (0 / 1)

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 ["5.10.3 Setting various event conditions"](#).

**(3) Event Manager**

This area is used to display the list of the events registered. (Refer to ["Table 5-12 Event Icon"](#), ["\(4\) Manipulation in event manager area"](#).)

## Function Buttons

---

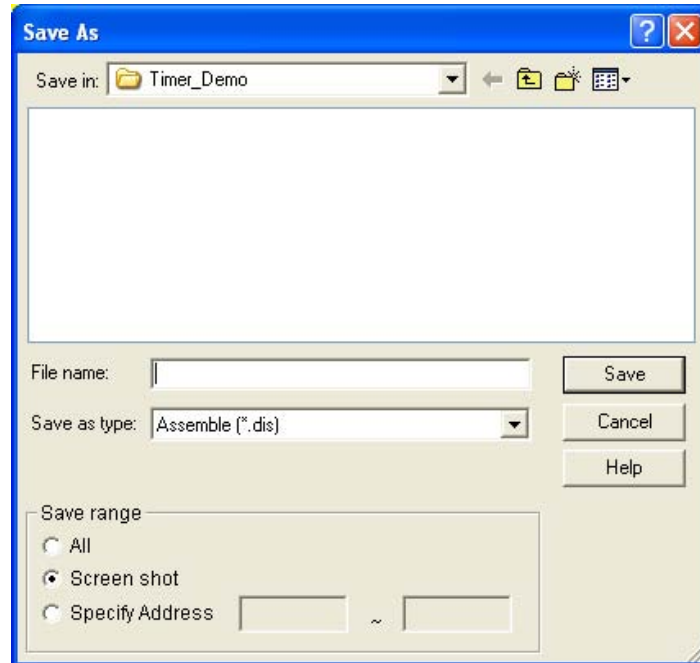
Refer to ["Function buttons \(Related event function\)"](#) in the [Event Manager](#).



## 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 "5.12.2 Window display information (view file)".)

Figure 6-46 View File Save Dialog Box



- Opening
- Explanation of each area
- Function buttons

### Opening

When the window to be saved is the current window, select [File] menu -> [Save As...].

### Explanation of each area

- (1) Save in, File name
- (2) Save as type
- (3) Save range

#### (1) Save in, File name

This area is used to specify the file name to be saved. 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.

**(2) Save as type**

This area is used to specify the type (extension) of the file to be saved. (Refer to "Table 5-15 Type of View Files".)

The extension of the file corresponding to the current window is displayed.

**(3) Save range**

Specify the range of data to be saved.

This area is displayed if the current window to be saved is the following.

- [Assemble Window](#)
- [Memory Window](#)
- [Source Window](#)
- [Trace View Window \[IECUBE\]](#)

**(a) All**

This should be selected to save the entire range, from the first line to the last line.

**(b) 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 [Source Window](#) is in the mixed display mode, however, the window contents are saved from the source line that includes the area visible on the screen.

**(c) 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.

If a range of 100 lines / 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.

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 [IECUBE]	Specify the range of trace frames to be saved. (Refer to "Table 6-10 Frame Number Specification Format".) 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 "Table 5-5 Specifying Symbols".) The default radix for inputting a numeric value is hexadecimal.

## 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 "5.12.2 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.

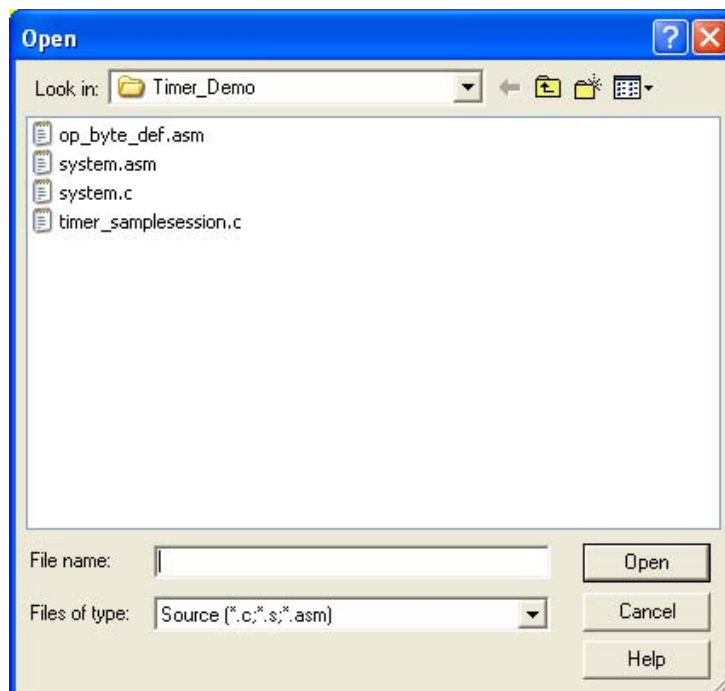
### - Loading source file to which symbol information has been read

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 6-47 View File Load Dialog Box



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

## Opening



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

---

## Explanation of each area

---

(1) [Look in, File name](#)

(2) [Files of type](#)

### (1) Look in, File name

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.

Up to 257 character string with a extension can be specified.

### (2) Files of type

This area is used to specify the type (extension) of the file to be loaded. (Refer to "[Table 5-15 Type of View Files](#)".)

---

## 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.

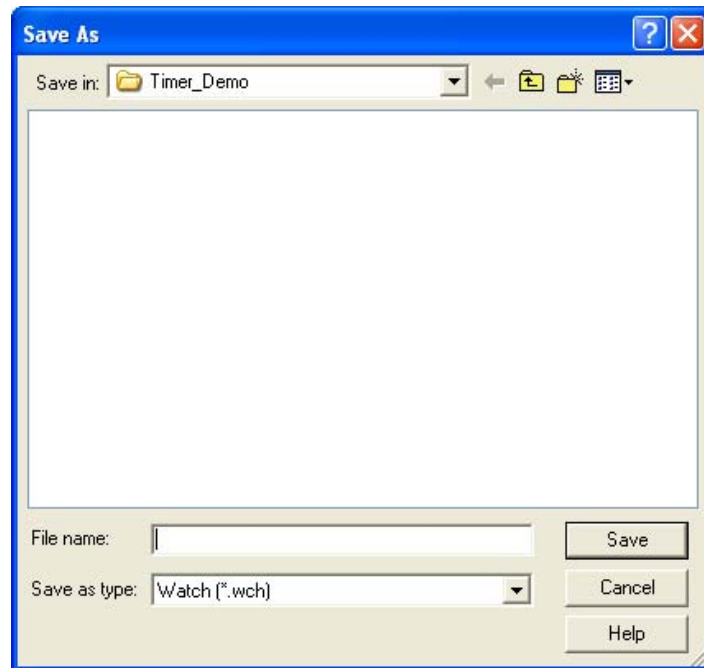
---

## 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 ["5.12.3 Window setting information \(setting file\)"](#).)

Figure 6-48 Environment Setting File Save Dialog Box



- Opening
- Explanation of each area
- Function buttons

---

### Opening

When the window to be saved is the current window, select [File] menu -> [Environment] -> [Save As...].

---

### Explanation of each area

- (1) Save in, File name
- (2) Save as type

#### (1) Save in, File name

This area is used to specify the file name to be saved. 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.

**(2) Save as type**

This area is used to specify the type (extension) of the file to be saved. (Refer to ["Table 5-16 Type of Setting Files"](#)).

The extension of the file corresponding to the current window is displayed.

**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.

---

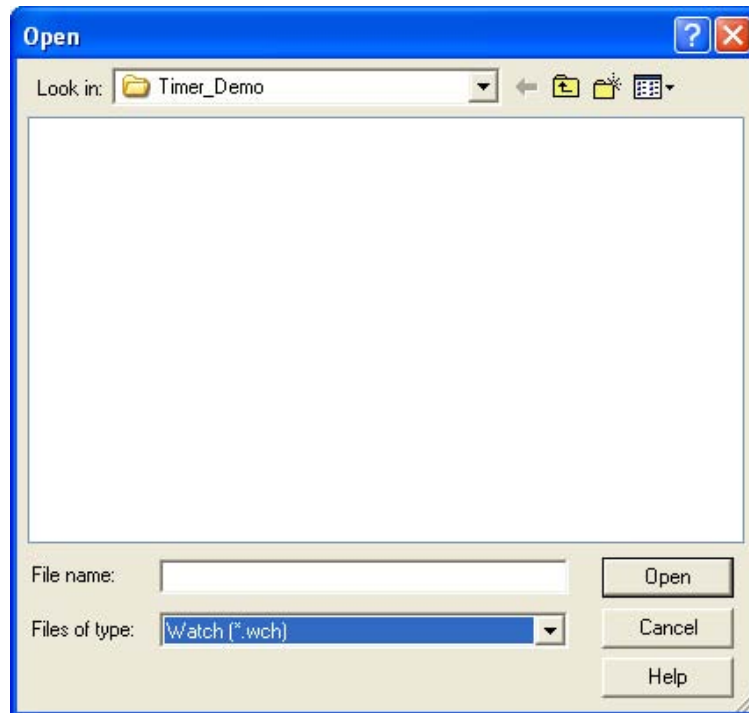
## Environment Setting File Load Dialog Box

---

This dialog box is used to read the setting files. (Refer to "5.12.3 Window setting information (setting file)".)

When a setting file is loaded, the target window opens and the setting information at saving is restored.

Figure 6-49 Environment Setting File Load Dialog Box



- Opening
- Explanation of each area
- Function buttons

---

### Opening

---

select [File] menu -> [Environment] -> [Open...].

---

### Explanation of each area

---

- (1) Look in, File name
- (2) Files of type

#### (1) Look in, File name

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.

Up to 257 character string with a extension can be specified.



**(2) Files of type**

This area is used to specify the type (extension) of the file to be loaded. (Refer to ["Table 5-16 Type of Setting Files"](#).)

**Function buttons**

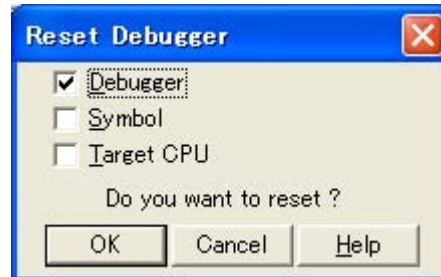
---

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.

## Reset Debugger Dialog Box

This dialog box is used to initialize the ID78K0S-QB, CPU, and symbol information.

Figure 6-50 Reset Debugger Dialog Box



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

### 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 checked item.

Debugger	Initializes the ID78K0S-QB (default).
Symbol	Initializes the symbol information.
Target CPU	Initializes the CPU.

### Function buttons

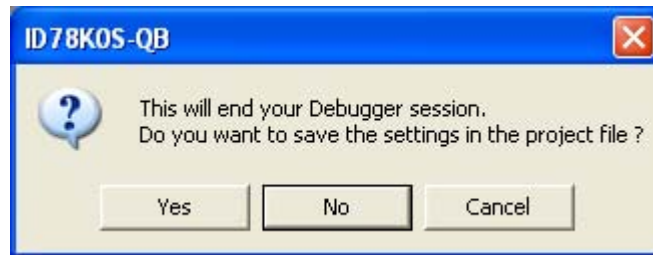
OK	Initializes according to the setting.
Cancel	Cancels the changes and closes this dialog box.
Help	Displays this dialog box online help files.

## 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 ID78K0S-QB. (Refer to "[5.12.1 Debugging environment \(project file\)](#)".)

It can be specified in the [Debugger Option Dialog Box](#) that the ID78K0S-QB is terminated without this confirmation dialog box being opened.

Figure 6-51 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 ID78K0S-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 ID78K0S-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 ID78K0S-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 ID78K0S-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 ID78K0S-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 OCD Control Code

Figure 6-52 About Dialog Box



- Opening
- Function buttons

## Opening

Select [Help] menu -> [About...], or click the <About...> button in the [Configuration Dialog Box](#).

## Function buttons

---

OK	Closes this dialog box.
----	-------------------------

---

## Console Window

---

This window is used to input commands that control the ID78K0S-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 6-53 Console Window



- Opening

---

### Opening

---

Select [Browse] menu -> [Console].

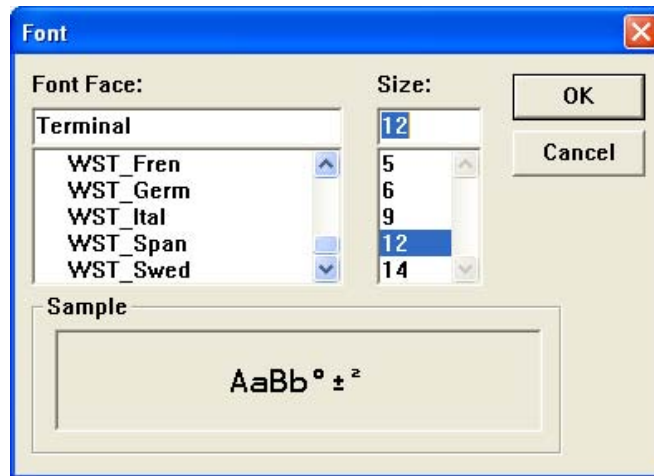
---

## Font Dialog Box

---

This dialog box is used to select the font and font size to be displayed in the [Source Window](#), the [Watch Window](#), the [Quick Watch Dialog Box](#), the [Local Variable Window](#) and the [Stack Window](#).

Figure 6-54 Font Dialog Box



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

---

### Opening

Click the <Font> button in the [Debugger Option Dialog Box](#).

---

### Explanation of each area

- (1) [Font Face](#)
- (2) [Size](#)
- (3) [Sample](#)

#### (1) Font Face

This area is used to select a font from the fonts currently usable.

Only fonts with equal width (fonts with a constant stroke width and a fixed pitch) are enumerated. When a font name is selected from the list, the font name is displayed in the text box, and the font size that can be used with that font is displayed under (2) [Size](#).

**(2) Size**

This area is used to specify the font size (unit: pt.).

On the drop-down list, the font size usable for the font specified in [\(1\) Font Face](#) is displayed in point units. When the font size is selected from the drop-down list, the selected font size is displayed in this area. The font size can also be directly input to the text box from the keyboard.

**(3) Sample**

This area displays a sample character string of the specified font and size.

---

**Function buttons**

---

OK	Validates the settings and closes this dialog box.
Cancel	Cancels the changes and closes this dialog box.



---

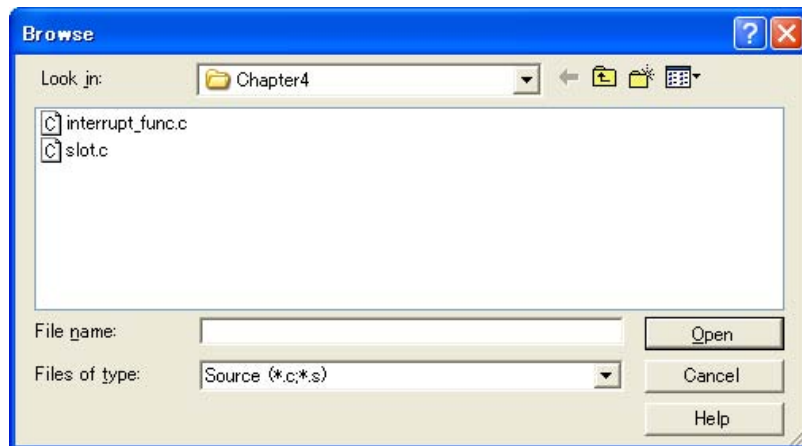
## Browse Dialog Box

---

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 directory first specified by the source path is displayed. When the dialog box is opened the second and subsequent times, the previously displayed directory is recorded and displayed again. If the <Cancel> button is clicked, however, the previously displayed directory is not recorded.

Figure 6-55 Browse Dialog Box



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

---

### Opening

Click the <Browse...> button in the target dialog box.

---

### Explanation of each area

- (1) [Look in, File name](#)
- (2) [Files of type](#)

#### (1) Look in, File name

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.

**(2) Files of type**

This area is used to specify the type (extension) of the file to be opened (refer to ["Table 5-4 File Type Can Be Displayed"](#)).

**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 ID78K0S-QB.

- [Command Line Rules](#)
- [Command List](#)
- [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 Command List

Table 7-1 List of Debugger Control Commands

Command name	Function
<a href="#">address</a>	Evaluation of address expression
<a href="#">assemble</a>	Disassemble/line assemble
<a href="#">batch</a>	Executing batch (with echo)
<a href="#">breakpoint</a>	Setting/deletion of breakpoint
<a href="#">dbgexit</a>	Terminating ID78K0S-QB
<a href="#">download</a>	Download of files
<a href="#">erase</a>	Erase of the internal flash memory
<a href="#">extwin</a>	Creation of expansion window
<a href="#">finish</a>	Returning from function
<a href="#">go</a>	Continuous execution
<a href="#">help</a>	Display of help
<a href="#">hook</a>	Setting of hook
<a href="#">inspect</a>	Symbol inspect
<a href="#">jump</a>	Jump to window
<a href="#">map</a>	Setting / deleting memory mapping
<a href="#">mdi</a>	Setting of expansion window
<a href="#">memory</a>	Display/setting of memory
<a href="#">module</a>	Display of the list of files and functions
<a href="#">next</a>	Procedure step

Command name	Function
<a href="#">refresh</a>	Redrawing of window
<a href="#">register</a>	Display/setting of register value and SFR value
<a href="#">reset</a>	Reset
<a href="#">run</a>	Reset and execution of CPU
<a href="#">step</a>	Step execution
<a href="#">stop</a>	Stop execution
<a href="#">upload</a>	Upload
<a href="#">version</a>	Display of the version information
<a href="#">watch</a>	Display/setting of variables
<a href="#">where</a>	Stack trace
<a href="#">wish</a>	Start of Tclet
<a href="#">xtime</a>	Operation of timer <b>[IECUBE]</b> <b>[MINICUBE2]</b>
<a href="#">xtrace</a>	Operates tracer <b>[IECUBE]</b>

Table 7-2 List of Console/Tcl Commands

Command name	Function
<a href="#">alias</a>	Creation of another name
<a href="#">cd</a>	Change of directory
<a href="#">clear</a>	Clears the screen
<a href="#">echo</a>	Echo
<a href="#">exit</a>	Close/end
<a href="#">history</a>	Display of history
<a href="#">ls</a>	Display of files
<a href="#">pwd</a>	Check of the directory
<a href="#">source</a>	Execution of batch
<a href="#">time</a>	Measurement of time for commands
<a href="#">tkcon</a>	Console control
<a href="#">unalias</a>	Deletion of another name
<a href="#">which</a>	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/ldtcl/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

- dcl(chip) - Chip name read only
- dcl(prjfile) - Project file name read only
- dcl(srcpath) - Source path read only
- dcl(ieid) - IType read only
- dcl(iestat) - IEstatus read only
- dcl(bkstat) - Break status read only
- env(LANG) - Language
- dcl\_version - Dclversion read only

## 7.5 List of Packages

- tcltest - 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/itcl/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.

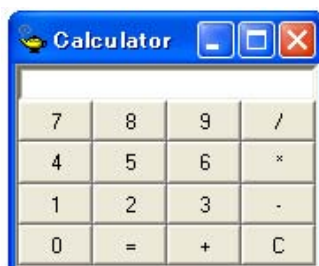
### 7.7.1 Samples (Calculator Script)

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 7-4 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 ID78K0S-QB
17	After changing extended option
18	After changing debugger option
19	After downloading
20	After changing memory or register
45	After resetting symbol

## 7.9 Hook Procedure

A hook can be set in the ID78K0S-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 ID78K0S-QB is closed.

### (1) [When hook is set with ID78K0S-QB control command]

- 1) Create script file a. with an editor.
- 2) Start up the ID78K0S-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 a script file. with an editor.<sup>Note</sup>
- 2) Start up the ID78K0S-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

- Executes when the aliases.tcl console is opened.  
Sets the default alias etc.
- Executes when the project file name.tcl project is opened. The following hooks can be used.  
BeforeDownload, AfterDownload, AfterCpuReset, BeforeCpuRun, AfterCpuStop
- Executes when the load module name.tcl load module 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
(IDCON) 3 % tkcon save a:/log.txt
```

## breakpoint

breakpoint - Setting/deletion of breakpoint

### Input format

**breakpoint** *?options? ?address1? ?address2?*

**breakpoint** -delete *brkno*

**breakpoint** -enable *brkno*

**breakpoint** -disable *brkno*

**breakpoint** -information

### 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>[IECUBE] [MINICUBE+]</b>
<b>-read</b>	An <i>address</i> data read break is set. <b>[IECUBE] [MINICUBE+]</b>
<b>-write</b>	An <i>address</i> data write break is set. <b>[IECUBE] [MINICUBE+]</b>
<b>-access</b>	An <i>address</i> data access break is set. <b>[IECUBE] [MINICUBE+]</b>
<b>-size</b> <i>size</i>	The access size is set (8, 16, 32).(Unit: bit) <b>[IECUBE] [MINICUBE+]</b>
<b>-data</b> <i>value</i>	The data condition is set. <b>[IECUBE] [MINICUBE+]</b>
<b>-datamask</b> <i>value</i>	The data mask is set. <b>[IECUBE] [MINICUBE+]</b>
<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.

## Usage example

---

(IDCON) 1 % breakpoint -software main

1

(IDCON) 2 % breakpoint -access v

1

(IDCON) 3 % breakpoint -i

1 Brk00001 enable \_v

(IDCON) 4 % breakpoint -disable 1

(IDCON) 5 % breakpoint -read glvar

2

(IDCON) 6 % breakpoint -i

1 Brk00001 disable \_v

2 Brk00002 enable \_glvar

(IDCON) 7 % breakpoint -delete 1

(IDCON) 8 % breakpoint -i

2 Brk00002 enable \_glvar



## dbgexit

---

dbgexit - Terminating ID78K0S-QB

### Input format

---

**dbgexit** *?options?*

### Functions

---

Terminate the ID78K0S-QB.

The following are *options*:

<b>-saveprj</b>	Project is saved on terminating ID78K0S-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>-binary</b>	Binary format data is downloaded.
<b>-nosymbol</b>	Download is executed. Symbol information is not read.
<b>-symbolonly</b>	Symbol information is read.
<b>-erase [MINICUBE2]</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

## 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 ID78K0S-QB is reset.

### Usage example

---

(IDCON) 1 % hook d:/foo.tcl



## 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.

?	Match 1 character
*	Match characters other than 0
[chars]	Match chars character. (Range specification such as [a-z/0-9] also possible.)
\x	Match character x. (? * [ ] \ specification also possible.)

The following are *options*:

<b>-nocase</b>	Does not distinguish between upper and lowercase.
<b>-address</b>	Displays in pair with symbol address.

### Usage example

---

(IDCON) 1 % inspect test1.out {[a-z]\*}

## 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*

### Functions

Sets, deletes, and displays memory mapping.

The following are *options*:

<b>-rrm</b>	Start address of RRM area is set. RRM area can be divided into 8 partitions. <b>[IECUBE] [MINICUBE+]</b> The start address and size are specified in pairs in list format as follows. {address size} {address size} {address size} ...} size is one of 1-16 bytes, and the total size is up to 16. <b>[IECUBE] [MINICUBE+]</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 0x7fff 8 {IROM}
2: 0x8000 0x87ff 8 {Target RRM}
3: 0x8800 0x9fff 8 {Target}
4: 0xa000 0xf7ff 8 {NonMap}
5: 0xf800 0xfaff - {NonMap}
6: 0xfb00 0xfedf 8 {Saddr}
7: 0xfef0 0xfeff 8 {Register}
8: 0xff00 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
- Referencing a memory area in the RRM area **[IECUBE] [MINICUBE+]**

### 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** *progrname* *?filename?*

### Functions

---

Displays the list of files and functions of the load module specified by *progrname*.

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.
---------------	--------------------------------------------

### 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 ID78K0S-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 ID78K0S-QB is reset.
<b>-symbol</b>	Symbol is reset.
<b>-event</b>	Event is 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

## upload

---

---

upload - Upload

### Input format

---

**upload** *?options? filename address1 address2*

### Functions

---

Saves the memory data within the specified range in a file.

The following are *options*:

<b>-binary</b>	The data is saved in binary format.
<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 ID78K0S-QB.

## Usage example

---

```
(IDCON) 1 % version
GUI       : V2.81 [XX XXXX 200X]
Devicefile : 78K/0S[uPD78F9222] V9.99
Debugger   : 78K0S-QB Debugger V3.33 XX XXXX 200X]
Executer   : 78K0S MINICUBE Executer V1.00f
Micro program : OCD Control Code V1.00
Assembler  : 78K0S Asm/Disasm V2.81 XX XXXX 200X]
Tcl/Tk     : 8.4.5
```

## 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.

## Usage example

---

```
(IDCON) 1 % where
1: test2.c#sub2(int i)#13
2: test.c#num(int i)#71
3: test.c#main()#82
```

## 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

---

## xtime

[IECUBE] [MINICUBE2]

---

xtime - Operation of timer

---

### Input format

**xtime** *option*

---

### Functions

Operates timer.

The following are *options*:

<b>-gobreak</b>	Time from Go to Break is displayed in nsec.
-----------------	---------------------------------------------

---

### Usage example

(IDCON) 1 % xtime -gobreak  
2198375 nsec

## xtrace

**[IECUBE]**

xtrace - Operation of tracer

### Input format

**xtrace** -dump *?-append?* *frameno* *?filename?*

### Functions

Operates tracer.

The following are *options*:

<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.

### Usage example

```
(IDCON) 1 % xtrace -dump 3
_ A 0219 01C1 0A5A04 M1 SET1 P4.5H
_ A 0220 01C4 0A3A0C M1 SET1 P12.3H
_ A 0221 01C7 20      M1 RET
```

## 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 ?size?</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.
<b>tkcon close</b> <b>tkcon destroy</b>	Close the Console window.
<b>tkcon font ?fontname?</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 ?-newline?</b>	Displays the command history.
<b>tkcon save ?filename? ?type?</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

# APPENDIX A EXPANSION WINDOW

- [Overview](#)
- [Sample List of Expansion Window](#)
- [Activation](#)
- [Explanation of Each Sample Window](#)

## A.1 Overview

With the ID78K0S-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 ID78K0S-QB. Users can create windows using this Tcl.

The ID78K0S-QB is supplied with samples of the following expansion windows.

## A.2 Sample List of Expansion Window

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.3 Activation

The expansion window can be activated by selecting List, Grep, or Hook in [Others] on the [Browse] menu.

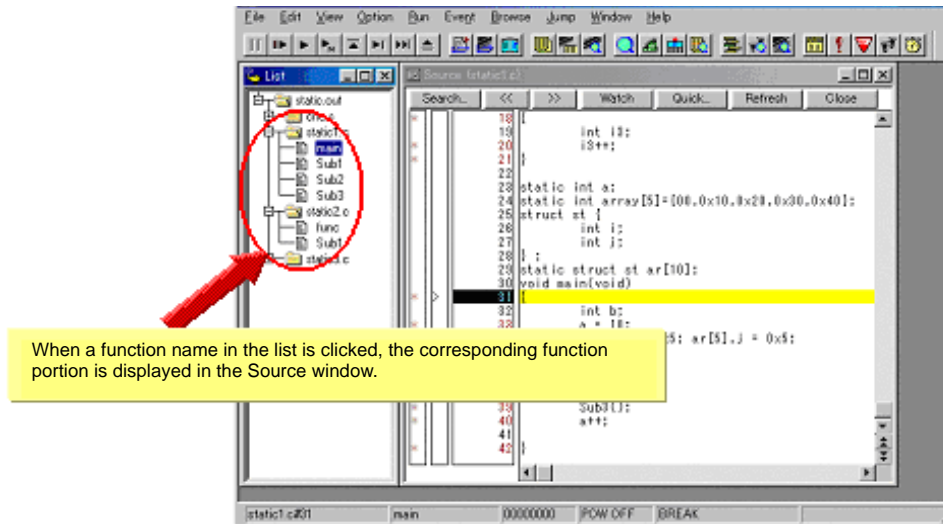
**Remark** Each .tcl file is installed in NECTools32\BIN\idctl\tools.

## A.4 Explanation of Each Sample Window

## 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

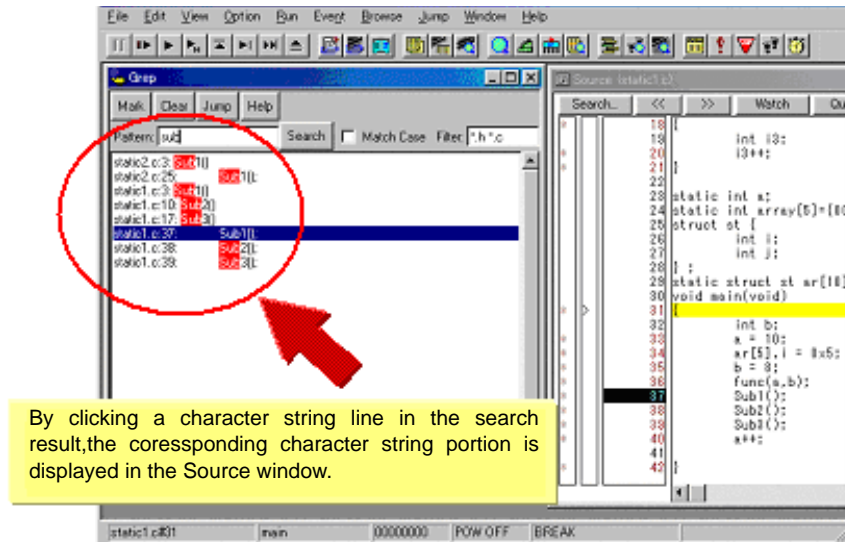


## Grep Window

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.

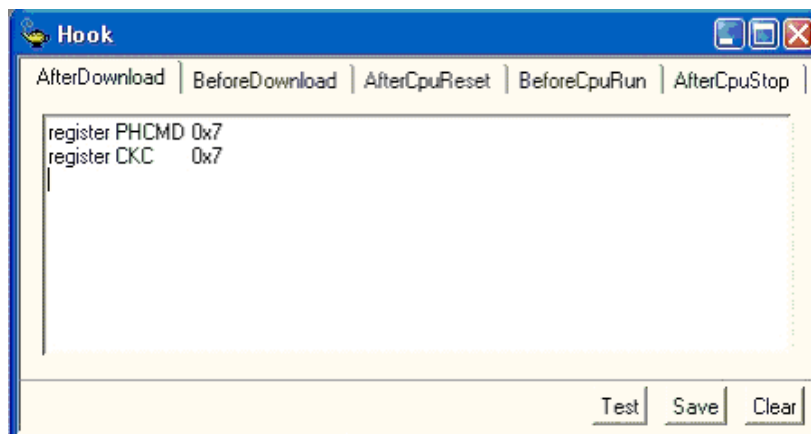
The hook procedure enables changing the register value before downloading a program, or after a CPU reset.

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 directory 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 ID78K0S-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.

**Caution** Specify the general-purpose register and the SFR for the register name.

# APPENDIX B INPUT CONVENTIONS

- Usable Character Set
- Symbols
- Numeric Values
- Expressions and Operators
- File Names

## B.1 Usable Character Set

Table B-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 B-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
[	Left bracket	Array subscript operator or indirect display symbol
]	Right bracket	

Character	Name	Usage
!	Exclamation	Absolute addressing start symbol
\$	Dollar	Relative addressing start symbol
#	Sharp mark	Symbol indicating an immediate value
.	Period	Direct member operator or bit position specifier
,	Comma	Delimiter between operands

## B.2 Symbols

- A symbol consists of characters A to Z, a to z, @, \_ (underbar), ?, and 0 to 9.
- A symbol must start with a character other than numerals 0 to 9.
- Uppercase characters (A to Z) and lowercase characters (a to z) are distinguished.
- 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).
- A symbol is defined by loading a load module file.
- Symbols are classified into the following types by the valid range:
  - (i) Global symbol (assembly language, structured assembly language, C language)
  - (ii) Static symbol (C language)
    - In-file static symbol
    - In-function static symbol
  - (iii) Local symbol (C language)
    - In-module local symbol (assembly language, structured assembly language)
    - In-file local symbol
    - In-function local symbol
    - In-block local symbol
- The following symbols are available for each language used:
  - (i) Assembly language, structured assembly language
    - label name, constant name, bit symbol name
  - (ii) 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)
- A symbol can be described instead of an address or numeric value.
- The valid range of a symbol is determined based on the source debug information when the source file is assembled or compiled.
- Describe only the symbol name of a global symbol.

- A local symbol is expressed in pairs with a file name.

## B.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 ID78K0S-QB, decimal numbers or hexadecimal numbers are alternately selected, depending on the default radix.

Table B-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)

## B.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 B-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.
<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>		

Symbol	Meaning	Explanation
(	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 B-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 B-6 Range of Radixes

Radix	Range
Binary number	0Y <= value <= 11111111111111111111111111111111Y (32 digits)
Octal number	0O <= value <= 37777777777O
Decimal number	-2147483648 <= value <= 4294967295 (A negative decimal number is internally converted into a 2's complement.)
Hexadecimal numbers	0H <= value <= 0FFFFFFFH

## B.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 ., +, or -.
- 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.

### (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.



# APPENDIX C KEY FUNCTION LIST

Table C-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 IOR 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 menu bar.
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.

Key	Function
F2	Forcibly stops program execution. Same function as [Run] menu -> [Stop].
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.

Key	Function
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].
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 D MESSAGES

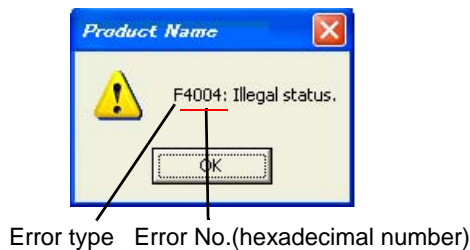
- Display Format
- Types of Messages
- Message Lists

## D.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.

Figure D-1 Error/Warning Dialog Box



## D.2 Types of Messages

The ID78K0S-QB outputs the following types of messages.

Table D-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.

## D.3 Message Lists

< From X0000 > < From 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.
<p>F0100: Can not communicate with ICE. Please confirm the installation of the device driver for the PC interface board.</p> <p>1) The driver may not be correctly installed. Reinstall the driver.</p> <p>2) Check the power supply to the emulator and the cable connections. <b>[MINICUBE2]</b></p>
A0101: Can not find initialization file (expc.ini).
A0102: Host name not found.
<p>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.</p> <p>1) Check the cable connections and the switches on the emulator.</p>
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.
<p>A0105: Failed in reading device file (d9xxx.78k).</p> <p>1) Necessary files may be damaged. Reinstall the device file.</p>
<p>A0106: Illegal data received.</p> <p>1) Check the power of the in-circuit emulator, cable connections, and setting of the interface board and restart the debugger.</p>
A0107: Can not communicate with ICE.
A0108: Failed in reading initialization file (expc.ini).
<p>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.</p> <p>1) USB communication was disconnected. Terminate the debugger, check the USB connection, and then restart the debugger. <b>[MINICUBE+] [MINICUBE2]</b></p>
<p>A010a: Cannot run debugger and a utility at the same time.</p> <p>1) The QB-Programmer, MINICUBE2 self-testing tool, or OCD Checker is running, so terminate it first. <b>[MINICUBE2]</b></p> <p>2) MINICUBE2 is being used by ID78K0S-QB, so terminate it first. <b>[MINICUBE2]</b></p> <p>3) Another debugger is running. Terminate it first and then restart the ID78K0S-QB. <b>[IECUBE] [MINICUBE+]</b></p>

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 ID78K0S-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.
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.
<p>F0313: Illegal map range.</p> <p>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).</p>
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.
<p>F03a0: Target is not turned on.</p> <p>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.</p>
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.
<p>F0400: Illegal condition.</p> <p>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.</p>
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 MINICUBE 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.
F0c05: Failed in updating monitor code.
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. 1) Enable writing to or erasing of entire block in the protect byte setting. <b>[MINICUBE2]</b>
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.
F0c3f : The intended value is not permitted for on-chip debugging.(address:XXXX)
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, MINICUBE connected). <b>[MINICUBE2]</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.
<p>F0c48: Flash programming is not permitted by the debugger setting.</p> <p>1) Flash programming is disabled in the debugger. Select "Permit" in the Flash Programming area in the <a href="#">Configuration Dialog Box</a>. <b>[MINICUBE2]</b></p>
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.
<p>F0c71: Reset cannot be performed.</p> <p>1) Check the clock signal. This may be caused by a stopped clock or a slow clock.</p>
F0c72: Monitor memory cannot be accessed.
F0c73: Monitor execution cannot be performed.
F0c74: CPU register cannot be accessed.
F0c75: Monitor has failed in shift in the debugging mode. Please reset the CPU.
F0c76: Initial state at the time of DCU access start is unusual.
F0c77: DCU access is unusuall.
F0c78: Failed in reading of trace data.
<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. <b>[MINICUBE2]</b></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, MINICUBE is connected). <b>[MINICUBE2]</b></p>

F0ca3:	Unsupported information is included in the on-chip debug information in the device file. 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.
F0ca4:	This device file does not include the IECUBE information. 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.
F0caf:	Trace block can not be stepped over.

**(2) From 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**

<p>F3000: No mapped address was accessed.</p> <p>1) The allocation addresses of the program and the addresses of the debugger may not match.</p> <p>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.</p>
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.
Ff504:	<p>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?
Wf601:	<p>When connecting the target system, please turn on the target system.</p> <p>1) When a target is not connected, simply click the &lt;OK&gt; button.</p>
Wf602:	Please change a MODE mask condition or connect the target system.
Ff603:	<p>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>
Af604:	<p>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>
Ff605:	<p>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>
Ff606:	<p>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>



Wf607: Please check connection of the exchange adapter. 1) Check the connection of the exchange adapter (EA). Recommend wearing of the exchange adapter, if the target is not connected.
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 MINICUBE 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:	<p>The [XXX] function of current program on PC position not found.</p> <p>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:	<p>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>
Wfb00:	<p>User program is running. Do you want to stop user program?</p> <p>1) &lt;Yes&gt; 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 ID78K0S-QB is terminated.</p> <p>&lt;No&gt; button is selected, execution of the user program is not stopped and the <a href="#">Exit Debugger Dialog Box</a> is not displayed. The ID78K0S-QB is not terminated.</p>
Wfb01:	<p>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>
Ffc00:	<p>Online help window cannot be started. Please install HTML Help environment with reference to a users manual.</p>
Ffd00:	<p>Failed to specify [XXX].</p>
Ffe00:	<p>The maximum size of RRM was exceeded.</p>
Wfe01:	<p>There is a duplicate RRM address.</p>
Wfe0b:	<p>It shift to the flash mode. Is it completely cleared but is the present event. Doesn't it care?</p>
Ffff:	<p>Interrupted.</p>

# APPENDIX E INDEX

## A

About Dialog Box ... 202  
Active status and static status ... 65  
Add I/O Port Dialog Box ... 167  
Add Watch Dialog Box ... 136  
Address move Dialog Box ... 125  
Assemble Search Dialog Box ... 123  
Assemble Window ... 119

## B

break  
    Breakpoint setting ... 41  
    setting break to variable ... 42  
Break Dialog Box ... 189  
break function ... 40  
Browse Dialog Box ... 207

## C

callback procedure ... 215  
Cautions before starting ... 23  
Change Watch Dialog Box ... 139  
character set ... 257  
clock ... 88  
Come Here ... 45  
command ... 204  
    command list ... 210  
Command reference ... 209  
Configuration Dialog Box ... 86  
Console Window ... 204  
Contents saved to project file ... 62  
Context menu ... 74

## D

Debug function list ... 34  
Debugger Option Dialog Box ... 95  
download ... 36  
Download Dialog Box ... 105  
Download Function / Upload Function ... 36  
drag & drop function ... 68

## E

Environment Setting File Load Dialog Box ... 198  
Environment Setting File Save Dialog Box ... 196  
Error messages at start up ... 28  
Errors ... 266  
Event function ... 56  
Event icon ... 59  
Event Setting Status (Event Mark) ... 111  
Exit Debugger Dialog Box ... 201  
Expressions ... 259  
Extended Option Dialog Box ... 90

## F

Font Dialog Box ... 205

## G

-g option ... 21

## H

hook procedure ... 216

## I

In-circuit emulator ... 20  
input conventions ... 257  
Installing ... 22  
IOR Select Dialog Box ... 165  
IOR Window ... 161

## J

jump function ... 66

## L

Load/Save Function ... 62  
Local Variable Window ... 141

## M

Main window ... 75  
Managing events ... 59  
Mask ... 89  
Memory Compare Dialog Box ... 155  
Memory Compare Result Dialog Box ... 157  
Memory Copy Dialog Box ... 153  
Memory Fill Dialog Box ... 151  
Memory manipulation function ... 50  
Memory Search Dialog Box ... 149  
Memory Window ... 146  
Messages ... 266  
Mixed display mode  
    Source window ... 38  
    Trace View Window ... 55

## O

Operating Environment ... 20  
Operators ... 260

## P

PM+ ... 29  
Point mark area ... 111, 120  
program code ... 112  
Program execution function ... 44  
project file ... 62, 201

Project File Load Dialog Box ... 103  
Project File Save Dialog Box ... 101

## Q

Quick Watch Dialog Box ... 133

## R

Range of Radixes ... 261  
Real-time monitor function (RRM function) ... 60  
Register Window ... 158  
reset ... 200  
Reset Debugger Dialog Box ... 200  
RRM Dialog Box ... 92

## S

Setting debugging environment ... 35  
setting file ... 64  
Software Break Manager ... 177  
Source Search Dialog Box ... 115  
Source Text move Dialog Box ... 117  
Source Window ... 110  
Stack trace display function ... 49  
Stack Window ... 143  
Start From Here ... 45  
Startup option ... 24  
Startup Routine ... 98  
Status bar ... 84  
Symbol To Address Dialog Box ... 126

## T

### Tcl

assemble ... 219  
batch ... 220  
breakpoint ... 221  
dbgexit ... 223  
download ... 224  
erase ... 225  
extwin ... 226  
finish ... 227  
go ... 228  
help ... 229  
hook ... 230  
inspect ... 231  
jump ... 232  
map ... 233  
mdi ... 234  
memory ... 235  
module ... 236  
next ... 237  
refresh ... 238  
register ... 239  
reset ... 240  
run ... 241  
step ... 242  
stop ... 243  
tkcon ... 251  
upload ... 244  
version ... 245  
watch ... 246

where ... 247  
wish ... 248  
xtime ... 249  
xtrace ... 250

Trace Data Select Dialog Box ... 173  
Trace function ... 54  
Trace Move Dialog Box ... 175  
Trace Result with Linking Window ... 69  
Types of Messages ... 266

## U

Uninstalling ... 22  
Upload ... 36  
Upload Dialog Box ... 108

## V

Verify check ... 91  
view file ... 63  
View File Load Dialog Box ... 194  
View File Save Dialog Box ... 191

## W

Watch function ... 46  
Watch Window ... 128  
window list ... 72  
window reference ... 71

*For further information,  
please contact:*

**NEC Electronics Corporation**  
1753, Shimonumabe, Nakahara-ku,  
Kawasaki, Kanagawa 211-8668,  
Japan  
Tel: 044-435-5111  
<http://www.necel.com/>

**[America]**

**NEC Electronics America, Inc.**  
2880 Scott Blvd.  
Santa Clara, CA 95050-2554, U.S.A.  
Tel: 408-588-6000  
800-366-9782  
<http://www.am.necel.com/>

**[Europe]**

**NEC Electronics (Europe) GmbH**  
Arcadiastrasse 10  
40472 Düsseldorf, Germany  
Tel: 0211-65030  
<http://www.eu.necel.com/>

**Hanover Office**  
Podbielski Strasse 166 B  
30177 Hanover  
Tel: 0 511 33 40 2-0

**Munich Office**  
Werner-Eckert-Strasse 9  
81829 München  
Tel: 0 89 92 10 03-0

**Stuttgart Office**  
Industriestrasse 3  
70565 Stuttgart  
Tel: 0 711 99 01 0-0

**United Kingdom Branch**  
Cygnus House, Sunrise Parkway  
Linford Wood, Milton Keynes  
MK14 6NP, U.K.  
Tel: 01908-691-133

**Succursale Française**  
9, rue Paul Dautier, B.P. 52180  
78142 Velizy-Villacoublay Cédex  
France  
Tel: 01-3067-5800

**Sucursal en España**  
Juan Esplandiú, 15  
28007 Madrid, Spain  
Tel: 091-504-2787

**Tyskland Filial**  
Täby Centrum  
Entrance S (7th floor)  
18322 Täby, Sweden  
Tel: 08 638 72 00

**Filiale Italiana**  
Via Fabio Filzi, 25/A  
20124 Milano, Italy  
Tel: 02-667541

**Branch The Netherlands**  
Limburglaan 5  
5616 HR Eindhoven  
The Netherlands  
Tel: 040 265 40 10

**[Asia & Oceania]**

**NEC Electronics (China) Co., Ltd**  
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian  
District, Beijing 100083, P.R.China  
TEL: 010-8235-1155  
<http://www.cn.necel.com/>

**NEC Electronics Shanghai Ltd.**  
Room 2509-2510, Bank of China Tower,  
200 Yincheng Road Central,  
Pudong New Area, Shanghai P.R. China P.C:200120  
Tel: 021-5888-5400  
<http://www.cn.necel.com/>

**NEC Electronics Hong Kong Ltd.**  
12/F., Cityplaza 4,  
12 Taikoo Wan Road, Hong Kong  
Tel: 2886-9318  
<http://www.hk.necel.com/>

**Seoul Branch**  
11F., Samik Lavied'or Bldg., 720-2,  
Yeoksam-Dong, Kangnam-Ku,  
Seoul, 135-080, Korea  
Tel: 02-558-3737

**NEC Electronics Taiwan Ltd.**  
7F, No. 363 Fu Shing North Road  
Taipei, Taiwan, R. O. C.  
Tel: 02-2719-2377

**NEC Electronics Singapore Pte. Ltd.**  
238A Thomson Road,  
#12-08 Novena Square,  
Singapore 307684  
Tel: 6253-8311  
<http://www.sg.necel.com/>