To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 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 1st, 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.

RENESAS

User's Manual

ID78K0-QB Ver. 2.94

Integrated Debugger

Operation

Target Device
78K0 Microcontrollers

Document No. U18330EJ1V0UM00 (1st edition)
Date Published October 2006 CP(K)

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.
Pentium is a trademark of Intel Corporation.

- **The information in this document is current as of October, 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).

M8E 02. 11-1

[MEMO]

PREFACE

Target Readers

This manual is intended for user engineers who design and develop application systems of the 78K0 microcontrollers.

Purpose

This manual is intended for users to understand the functions of the ID78K0-QB in the organization below.

Organization

This manual consists of the following chapters:

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

How to Use This Manual

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, microcontrollers, C language, and assemblers.

To understand the functions of the 78K0 microcontrollers
→ Refer to Hardware User's Manual for each product.

To understand the instruction functions of the 78K0 microcontrollers
→ Refer to 78K/0 Series Instructions User's Manual (U12326E).

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	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-78K0KX1H In-Circuit Emulator		U17081E
QB-78K0KX2 In-Circuit Emulator		U17341E
QB-780714 In-Circuit Emulator		U17366E
QB-78K0LX2 In-Circuit Emulator		U17468E
QB-78K0MINI On-Chip Debug Emulator		U17029E
RA78K0 Assembler Package Ver. 3.80	Operation	U17199E
	Assembly Language	U17198E
	Structured Assembly Language	U17197E
CC78K0 C Compiler Package Ver. 3.70	Operation	U17201E
	Language	U17200E
ID78K0-QB Ver. 2.94 Integrated Debugger	Operation	This manual
PM plus Ver.5.20		U16934E

[MEMO]

CONTENTS

LIST OF FIGURES

Figure No. Title, Page

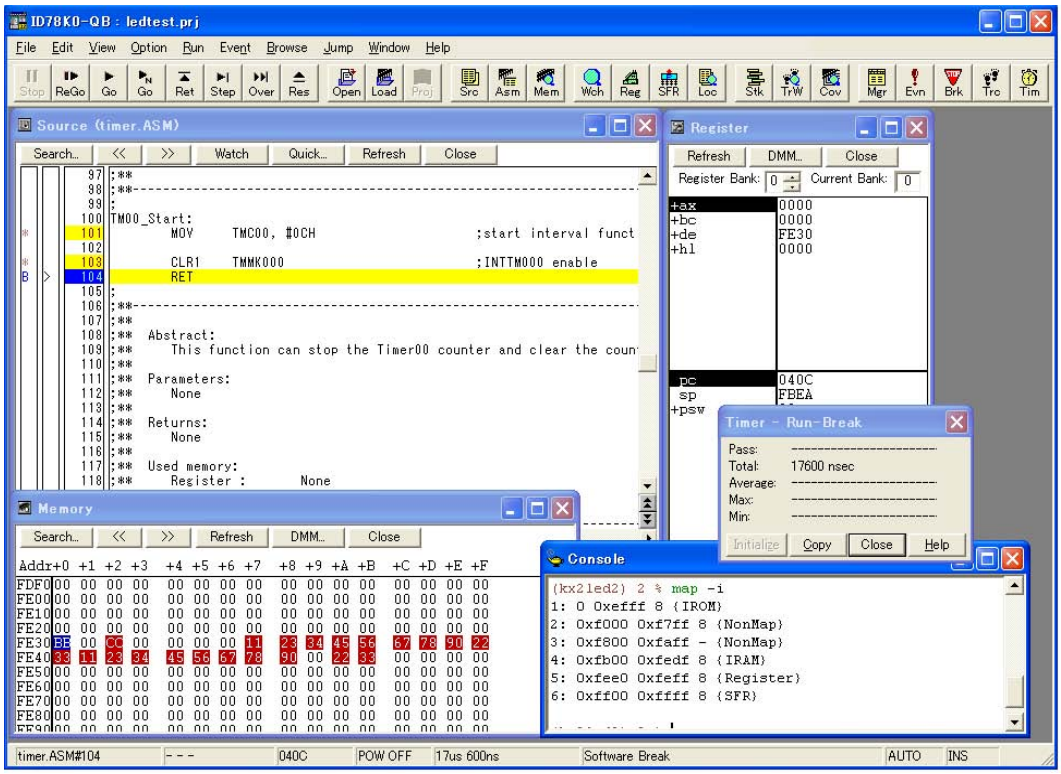
LIST OF TABLES

Table No. Title, Page

CHAPTER 1 OVERVIEW

The Integrated Debugger ID78K0-QB for the 78K0 microcontrollers(hereafter referred to as the ID78K0-QB) is a software tool developed for NEC Electronics 78K0 microcontrollers for embedded control. This software tool is intended to enable efficient debugging of user programs.

Figure 1-1 ID78K0-QB



This chapter explains the following items regarding the ID78K0-QB.

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

1.1 Features

The ID78K0-QB has the following features:

- New functions, enhanced functions
- Other

1.1.1 New functions, enhanced functions

(1) Support of IECUBE™, MINICUBE™ and MINICUBE2

Three emulators (IECUBE, MINICUBE and MINICUBE2) can be connected with 1 debugger (ID78K0-QB) (refer to "1.2 System Configuration").

USB2.0 is supported for the first time by NEC Electronics.

The term "MINICUBE" that appears in this document includes "MINICUBE2".

(2) Enhanced RRM function (when MINICUBE is connected)

Division setting for the sampling range of the real-time RAM monitor area is now possible (refer to "5.15 RRM Function"). This setting can be set to 16-byte RAM areas (up to eight locations in 1-byte units).

(3) Enhanced timer function (when IECUBE is connected)

Other than during Run-Break, the maximum time, minimum time, pass count, and average time are displayed (refer to "5.9 Timer Function (When IECUBE Is Connected)"). Measured time display during user program execution and timeout break are supported.

(4) Fail-safe break support

In addition to the traditional guarded areas and SFR areas, fail-safe break is also supported for the guarded areas of internal ROM/internal RAM (refer to "5.4.5 Fail-safe break").

(5) Enhanced command function

Script file specification is possible at ID startup (refer to "3.1 Startup Option And Argument Specification").

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

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

(6) Settings during program execution (when IECUBE is connected)

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

(7) Hardware detailed version display

The hardware detailed version is displayed in the About Dialog Box.

The version can also be confirmed in the Configuration dialog box prior to startup, and the display information can be copied and pasted.

(8) Supports one-byte spaces

Single-byte spaces can now be used for folder names.

(9) Support of code coverage measurement (when IECUBE is connected)

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

The code coverage can now be displayed in the Code Coverage Window, and sections for which coverage measurement is executed can now be displayed in the Source Window and Assemble Window (refer to "5.11 Coverage Measurement Function (When IECUBE Is Connected)").

(10) Addition of snapshot function (when IECUBE is connected)

The contents of the register memory and SFRs in the user program execution process can now be saved as snap data in the trace memory (refer to "5.13 Snapshot Function (When IECUBE Is Connected)").

(11) Addition of stub function (when IECUBE is connected)

A user program (subprogram) that has been downloaded or written via line assembling to a vacant space in the memory when an event is established can now be executed (refer to "5.14 Stub Function (When IECUBE Is Connected)").

(12) Enhanced DMM function

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

Real-time writing to the memory can be performed with the DMM function during user program execution (refer to "5.16 DMM Function").

(13) Support of flash self programming error emulation (when IECUBE is connected)

Flash self programming can now be debugged (refer to " Flash Option Dialog Box").

1.1.2 Other

(1) Using function of in-circuit emulator

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

(2) Support of on-chip debugging (when MINICUBE is connected)

A debugging function implemented by the on-chip debug unit.

(3) Flash memory writing function (when MINICUBE is connected)

The internal flash memory can be written and the load module can be downloaded by the same access method as an ordinary memory operation (refer to "5.7.4 Flash memory writing function (when MINICUBE is connected)").

(4) Security function (when MINICUBE is connected)

The ID code stored in the internal flash memory of a product with a security unit can be authenticated (refer to " Configuration Dialog Box", "(5) ID Code").

(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").

1.2 System Configuration

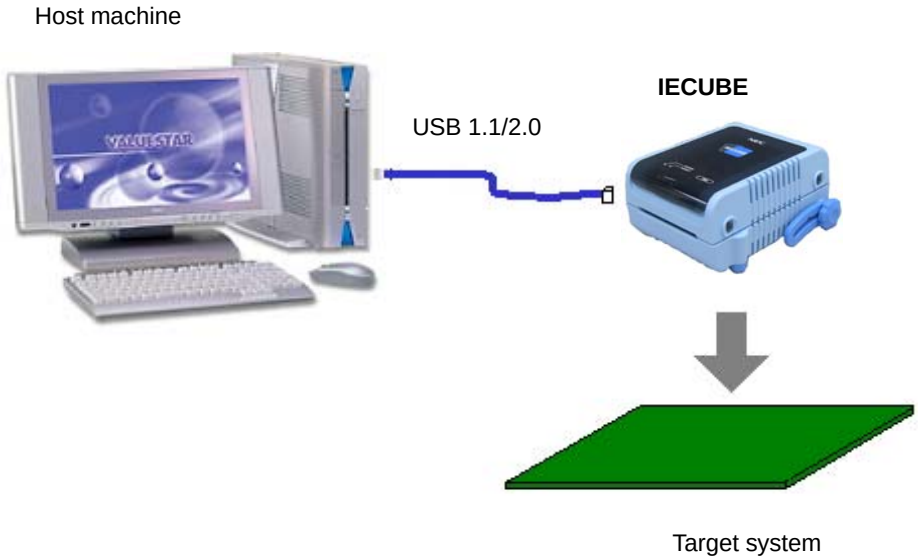
The ID78K0-QB can be connected to the following two types of emulators.

User programs developed for the 78K0 microcontrollers and a pleasant debugging environment for target systems are provided.

(1) IECUBE

IECUBE can be manipulated from the ID78K0-QB by connected it to the ID78K0-QB via a USB cable.

Figure 1-2 Example of ID78K0-QB System Configuration (IECUBE)

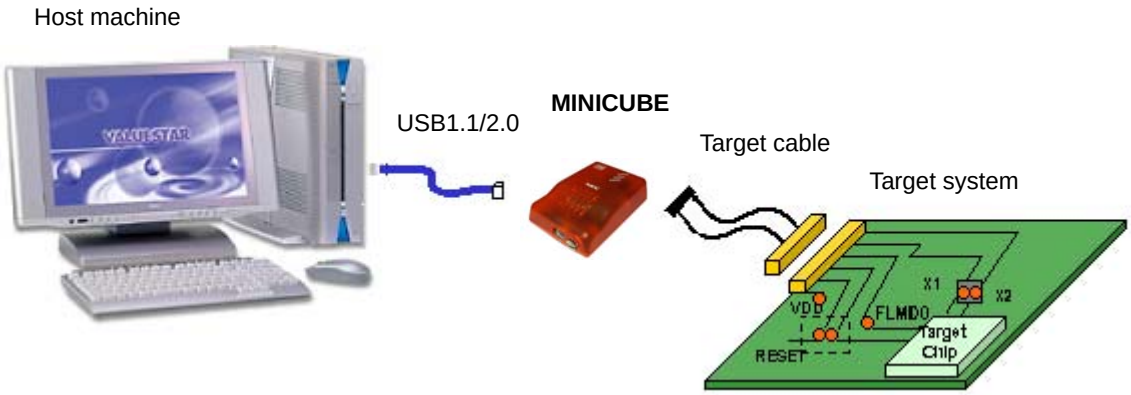


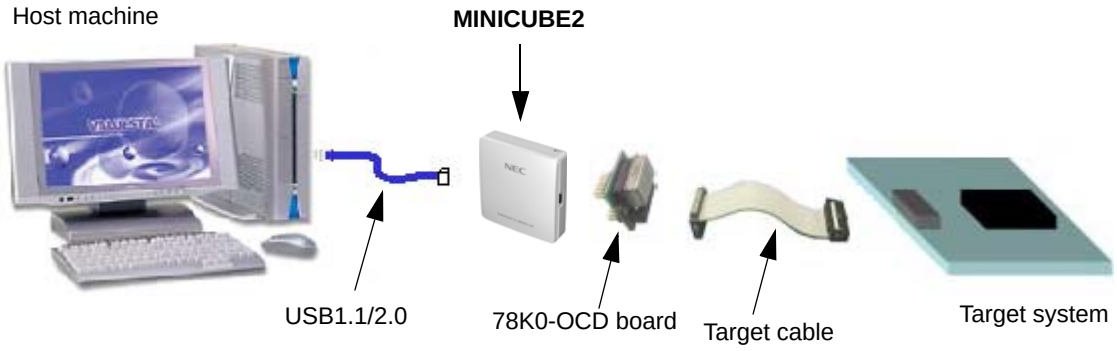
(2) MINICUBE, MINICUBE2

MINICUBE, MINICUBE2 is operated via the ID78K0-QB when it is connected to the host machine with the USB cable.

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

Figure 1-3 Example of ID78K0-QB System Configuration (MINICUBE)





1.3 Operating Environment

This section explains the following items regarding the operating environment.

- Hardware environment
- Software environment

1.3.1 Hardware environment

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

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

(2) In-circuit emulator

- IECUBE (QB-78K0KX2, other)
- MINICUBE (QB-78K0MINI)
- MINICUBE2 (QB-MINI2)

1.3.2 Software environment

(1) OS (any of the following)

Windows®2000, Windows XP(Home Edition, 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.

Remark: This file is available from the following Web site of NEC Electronics (ODS).
http://www.necel.com/micro/ods/eng/

(3) Supported tools (manufactured by NEC Electronics)

- Assembler package RA78K0 (Version 3.80 or later)
- C compiler package CC78K0 (Version 3.70 or later)
- Project manager PM+ (Version 5.20 or later)

1.4 Cautions During Debugging

The cautions to be observed during debugging are described below.

- When performing source level debugging
- Security ID

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 LK78K0 on standalone basis

Add the -g option.

1.4.2 Security ID

The object file used when MINICUBE is connected must include the security ID information.

For the security ID78K0-QB settings, refer to "RA78K0 Assembler Package Manipulation".

For details about the security ID, refer to the MINICUBE user's manual.

The security ID (ID code) from the ID78K0-QB is set in the Configuration Dialog Box.

CHAPTER 2 INSTALLATION

This chapter explains the following items about installation of ID78K0-QB:

- Installing
- Uninstalling

2.1 Installing

The following items must be installed, when the ID78K0-QB is used.

Table 2-1 Install

Item	Procedure
ID78K0-QB system disk	Install the contents of this disk according to the automatically executed installer.
Used device file	Install this file according to the DFINST.exe dedicated startup installer by selecting [Start] menu -> [Program] -> [NECTools32] -> [Device File Installer].

Caution: To install the ID78K0-QB again after the ID78K0-QB has been installed once, be sure to uninstall the ID78K0-QB. If the ID78K0-QB is installed in a folder different from that, without uninstalling, the ID78K0-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 ID78K0-QB:

- Startup Option And Argument Specification
- Starting
- Terminating
- Error Messages At Start Up

3.1 Startup Option And Argument Specification

The procedure for specifying the startup options and arguments for the ID78K0-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 ID78K0-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.

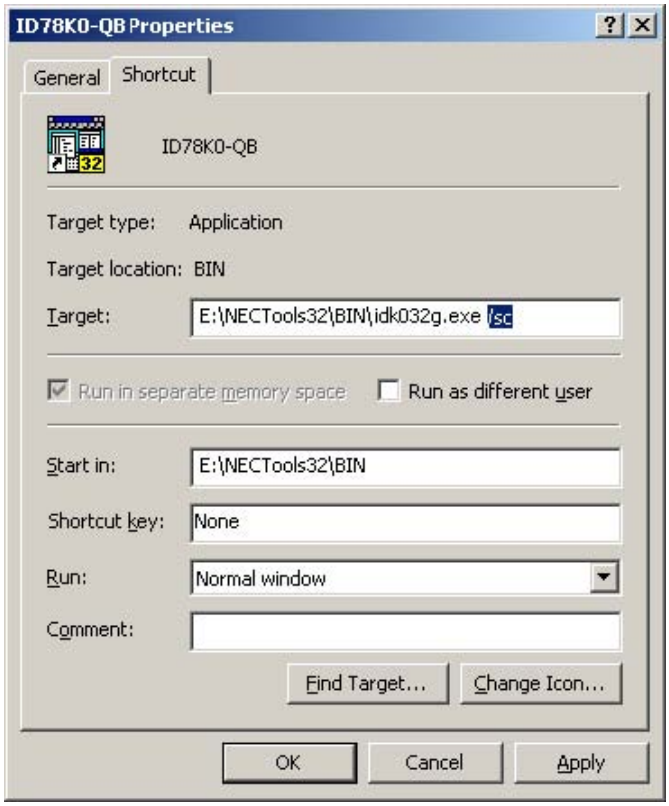
3.1.1 Specification method

1) Create the ID78K0-QB shortcut on the desktop.

   The ID78K0-QB execution file is located in the bin folder in the folder to which the installation was performed.

2) Open the properties of the created shortcut and after the execution file name displayed in [Target:], specify the option and argument (refer to "3.1.2 Specification format and options").

Figure 3-1 Startup Option (Example)



3.1.2 Specification format and options

(1) Specification format

```
idk032g.exe ?options?
idk032g.exe ?options? project
```

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

Arguments enclosed between '?' can be omitted.

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

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

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

(2) Specification options

The following options can be specified.

Table 3-1 Startup Options

Options	Meaning
/SC	Change background color of window to system color.
/SCRIPT:*script file name*	Specify the script file to be executed at startup.

(3) Specification example

Example1) Specification of script file only

```
idk032g.exe /script:c:\work\script.tcl
```

Example2) Specification of script file and project file

```
idk032g.exe /script:c:\work\script.tcl c:\work\project.prj
```

Example3) Specification when there are spaces in the path

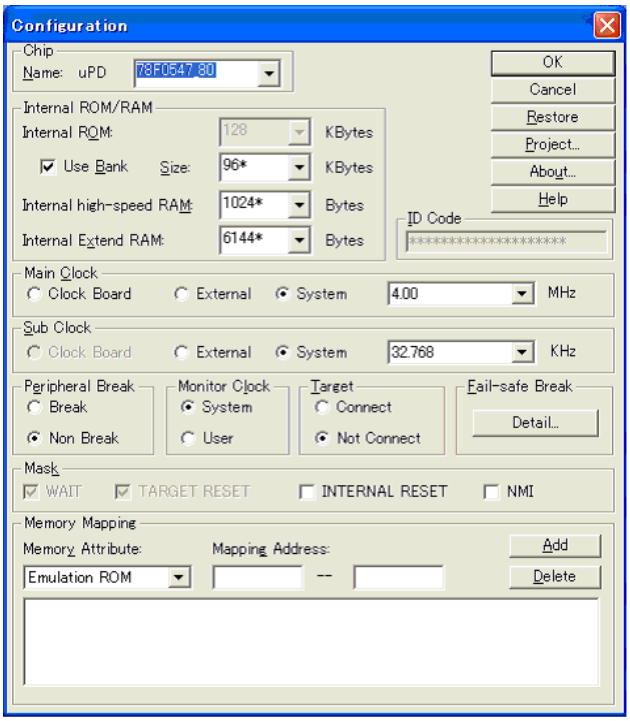
```
idk032g.exe /script:"c:\work folder\script.tcl" "c:\work folder\project.prj"
```

3.2 Starting

1) Start ID78K0-QB from the [Start] menu of PM+ or by clicking the shortcut created on the desktop.
Refer to "4.3 To Start ID78K0-QB From PM+" when starting from PM+.
Start the ID78K0-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 "3.4 Error Messages At Start Up".

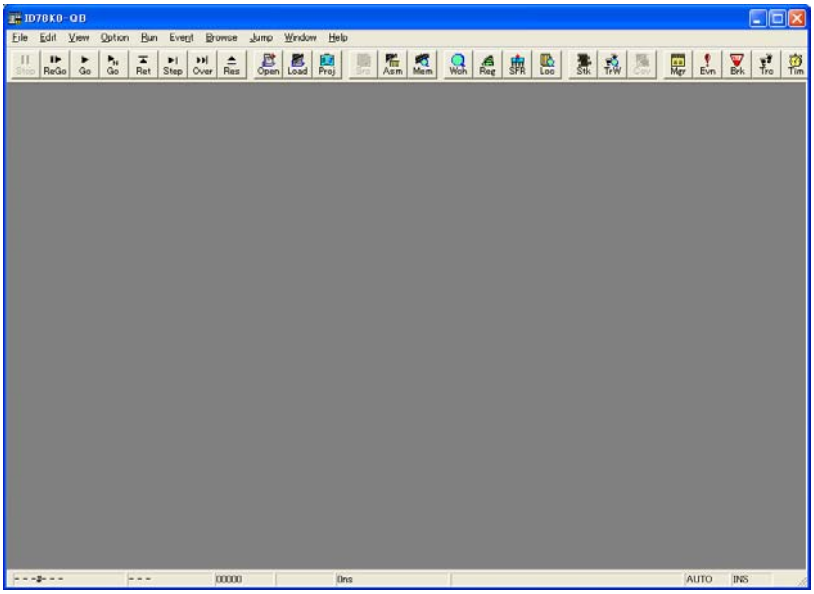
Figure 3-2 Configuration Dialog Box



2) Set the items related to the operating environment of the ID78K0-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 ID78K0-QB can be operated. Mainly use this window for debugging.

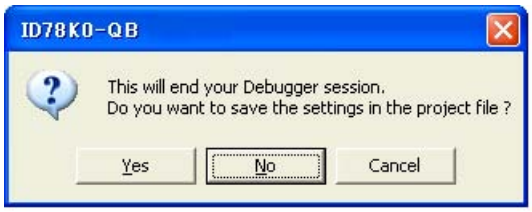
Figure 3-3 Main Window (At Startup)



3.3 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 ID78K0-QB terminated.

3.4 Error Messages At Start Up

Error messages that may be output when ID78K0-QB starts up are listed below (by order of occurrence). When these messages are output, refer to "APPENDIX D MESSAGES".

3.4.1 When IECUBE is connected

The pattern of the output error message differs as follows depending on the connection status with the target and the settings in the Configuration Dialog Box.

Table 3-2 Error Message Output Pattern

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

3.4.2 When MINICUBE is connected

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

CHAPTER 4 ASSOCIATION WITH PM+

The ID78K0-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 ID78K0-QB From PM+
- Auto Load

Caution: If a load module file is created by using the Windows command prompt, the function to associate the ID78K0-QB with the PM+ cannot be used.

4.1 Setting Build Mode

To debug the load module file created by the PM+ on the ID78K0-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 ID78K0-QB is registered as the debugger of the active project. The ID78K0-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 ID78K0-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 ID78K0-QB and click the <OK> button in this dialog box. For details of the setting, refer to the User's manual.

Caution: If [Execute symbol reset after download] in the [Debugger Settings] dialog box of the product with a internal flash memory is selected, the contents of the internal flash memory are erased before downloading (when MINICUBE is connected).

4.3 To Start ID78K0-QB From PM+

The ID78K0-QB can be started from the PM+ as follows:

- Click the ID78K0-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 ID78K0-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 ID78K0-QB is not saved to a project file being used by the PM+, the Configuration Dialog Box is opened. At this time, the device type (chip name) cannot be changed.

Caution: With the PM+, if too many source files are registered in a project, the number of files may exceed the upper limit of the source path length that can be registered to the ID78K0-QB, the source files consequently may not be displayed automatically.
For details on the source path length, refer to "(1) Source Path" in the Debugger Option Dialog Box.

4.3.1 Restoring debugging environment

The previous debugging environment can be restored by the following procedure when the ID78K0-QB is started from the PM+:

1) Create a new workspace (project file: e.g., sample.prj) on the PM+[Note].
2) Start the ID78K0-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 ID78K0-QB is started.
3) Download the load module file to be debugged with the Download Dialog Box of the ID78K0-QB.
4) Debug the load module file on the ID78K0-QB.
5) Click the <Yes> button on the Exit Debugger Dialog Box when the ID78K0-QB is terminated.
   -> The debugging environment will be saved to the project file (sample.prj) for the PM+ when the ID78K0-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 ID78K0-QB debugging).
6) When the ID78K0-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 ID78K0-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 ID78K0-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 ID78K0-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 ID78K0-QB by compiling and linking the file on the PM+ with the activated ID78K0-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 ID78K0-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 ID78K0-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 breakpoint 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 ID78K0-QB started, the load module will be automatically downloaded to the ID78K0-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 ID78K0-QB.

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

Item	Refer To
To set the debugging environment	5.1 Setting Debugging Environment
To download the load module	5.2 Download Function, Upload Function
To display the source file and the disassemble result	5.3 Source Display, Disassemble Display Function
To set a breakpoint	5.4 Break Function
To execute the user program	5.5 Program Execution Function
To check the variable value	5.6 Watch Function
To check and edit the memory contents	5.7 Memory Manipulation Function
To check and change the register variable	5.8 Register Manipulation Function
To check the execution time	5.9 Timer Function (When IECUBE Is Connected)
To check the trace data	5.10 Trace Function (When IECUBE Is Connected)
To check the code coverage measurement results	5.11 Coverage Measurement Function (When IECUBE Is Connected)
To manage the events	5.12 Event Function
Snapshot function	5.13 Snapshot Function (When IECUBE Is Connected)
Stub function	5.14 Stub Function (When IECUBE Is Connected)
RAM Sampling	5.15 RRM Function
DMM Function	5.16 DMM Function
To save the debug environment and window status	5.17 Load/Save Function
Jump function, linking window and cautions	5.18 Functions Common To Each Window

5.1 Setting Debugging Environment

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

- Setting operating environment
- Setting option
- Setting mapping

5.1.1 Setting operating environment

The in-circuit emulator operating environment settings are performed in the Configuration Dialog Box that is automatically displayed when ID78K0-QB starts up.

If a project file already exists, the debugging environment can be restored by clicking the <Project...> button. (refer to "5.17.1 Debugging environment (project file)").

5.1.2 Setting option

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

- Configuration Dialog Box
- Extended Option Dialog Box
- Fail-safe Break Dialog Box
- RRM Dialog Box
- Mask Option Dialog Box
- Flash Option Dialog Box
- Debugger Option Dialog Box
- Pseudo Emulation Dialog Box

5.1.3 Setting mapping

The mapping settings are performed in the Configuration Dialog Box.

The following types of mapping attributes are available:

Table 5-2 Mapping Attribute

Attribute	Meaning
Internal ROM	Internal RAM (When IECUBE is connected) A memory area specified as the internal ROM is equivalent to the internal ROM of the target device (core).If the target device attempts writing to this memory area, a write protect break occurs.
Internal High-speed RAM	A memory area specified as the internal high-speed RAM is equivalent to the internal high-speed RAM of the target device (core). The actual memory configuration depends on the target system.
Internal Extend RAM	A memory area specified as the internal extend RAM is equivalent to the internal extend RAM of the target device (core). The actual memory configuration depends on the target system.
Target	User area mapping The memory area specified for user area mapping becomes the area to accesses the memory in the target system or memory incorporated in the CPU.
I/O Protect	I/O protect area An I/O Protect area can be set in the area specified for the "target". The I/O protect area is displayed in the same manner as an area that is not mapped (display symbol: ??), on the Memory Window. By mapping an area with this attribute, data cannot be read or written from/to this area by the Memory Window, on the area can therefore be protected from an illegal access.To read or write the value of the area mapped with this attribute, register the value in the SFR Window or Watch Window.
Stack	Only the Internal high-speed RAM area can be set as the stack area (the Internal Extend RAM area cannot be set as the Stack area) (When IECUBE is connected). Note that if the Stack area is not specified, the entire Internal high-speed RAM area (excluding the register area) is specified as the Stack area by default.

5.2 Download Function, Upload Function

ID78K0-QB allows downloading and uploading of object files in the formats listed in the following table: Table 5-3, Table 5-4.

This section explains the following items:

- Download
- Upload

Remark: When MINICUBE is connected, the internal flash memory can be written and the load module can be downloaded (refer to "5.7.4 Flash memory writing function (when MINICUBE is connected)").

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.

(1) Format of file that can be downloaded

Format of file that can be downloaded is as follows:

Table 5-3 Type of File That Can Be Downloaded

Format	Extension
Load module (XCOFF(.lnk, .lmf))	Load Module (*lnk:*.lmf)
Intel Hex format (Standard and extension)	Hex Format (*.hex;*.hxb;*.hxf)[Note1] Hex Format [Bank](*.hex;*.hxb;*.hxf)[Note1] Hex Format [64KB] (*.hex;*.hxb;*.hxf)[Note1]
Motorola Hex format S type (S0, S2, S8)	
Extended Tektronix Hex format	
Binary data	Binary Data (*.bin)[Note2] Binary Data [Bank](*.bin)[Note2] Binary Data [64KB](*.bin)[Note2]
Coverage result (when IECUBE is connected)	Coverage (*.cvb)

Note1: Includes the Hex format compatible with flash memory self-programming mode.
The format of a HEX file is automatically determined.
Hex Format(*.hex;*.hxb;*.hxf): When memory banks are not used
Hex Format [Bank] (*.hex;*.hxb;*.hxf): When memory banks are used (for memory banks)
Hex Format [64KB](*.hex;*.hxb;*.hxf): When memory banks are used (when memory capacity is 64 KB or lower)

Note2: Binary Data (*.bin): When memory banks are not used

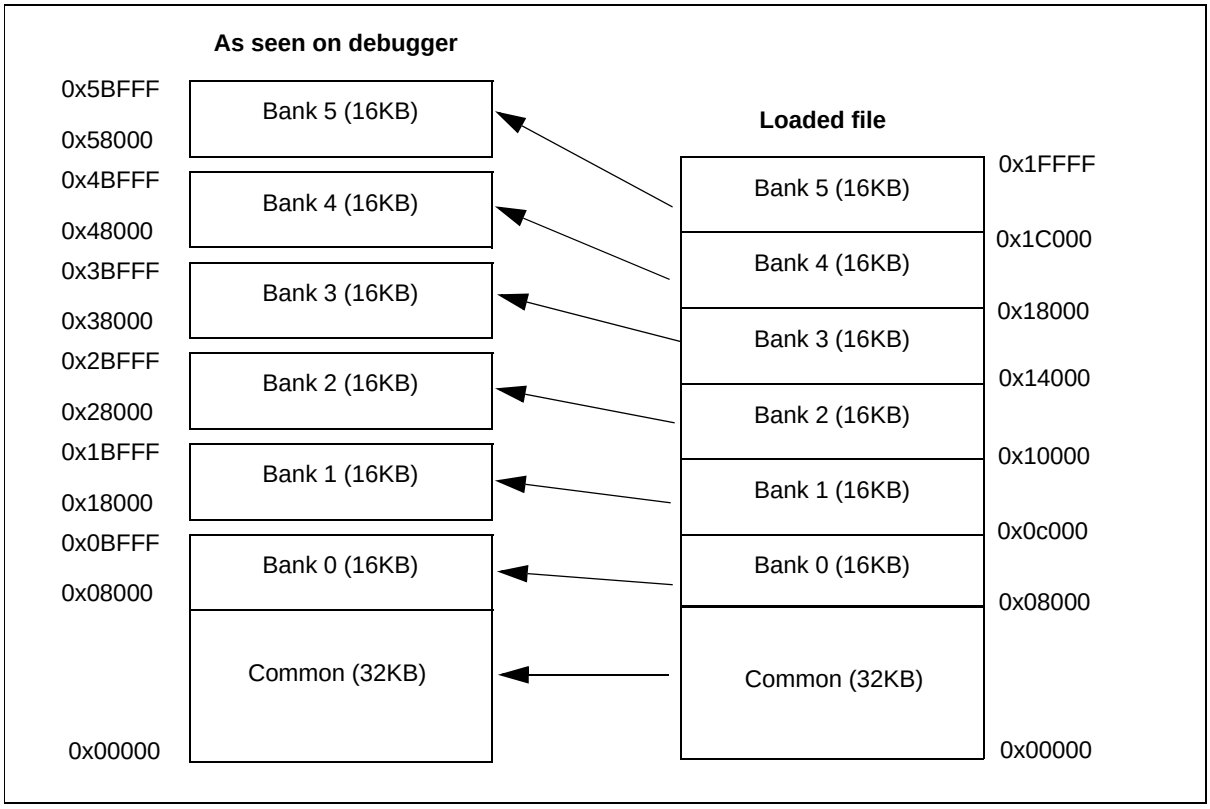
Binary Data [Bank](*.bin): When memory banks are used (for memory banks)

Binary Data [64KB](*.bin): When memory banks are used (when memory capacity is 64 KB or lower)

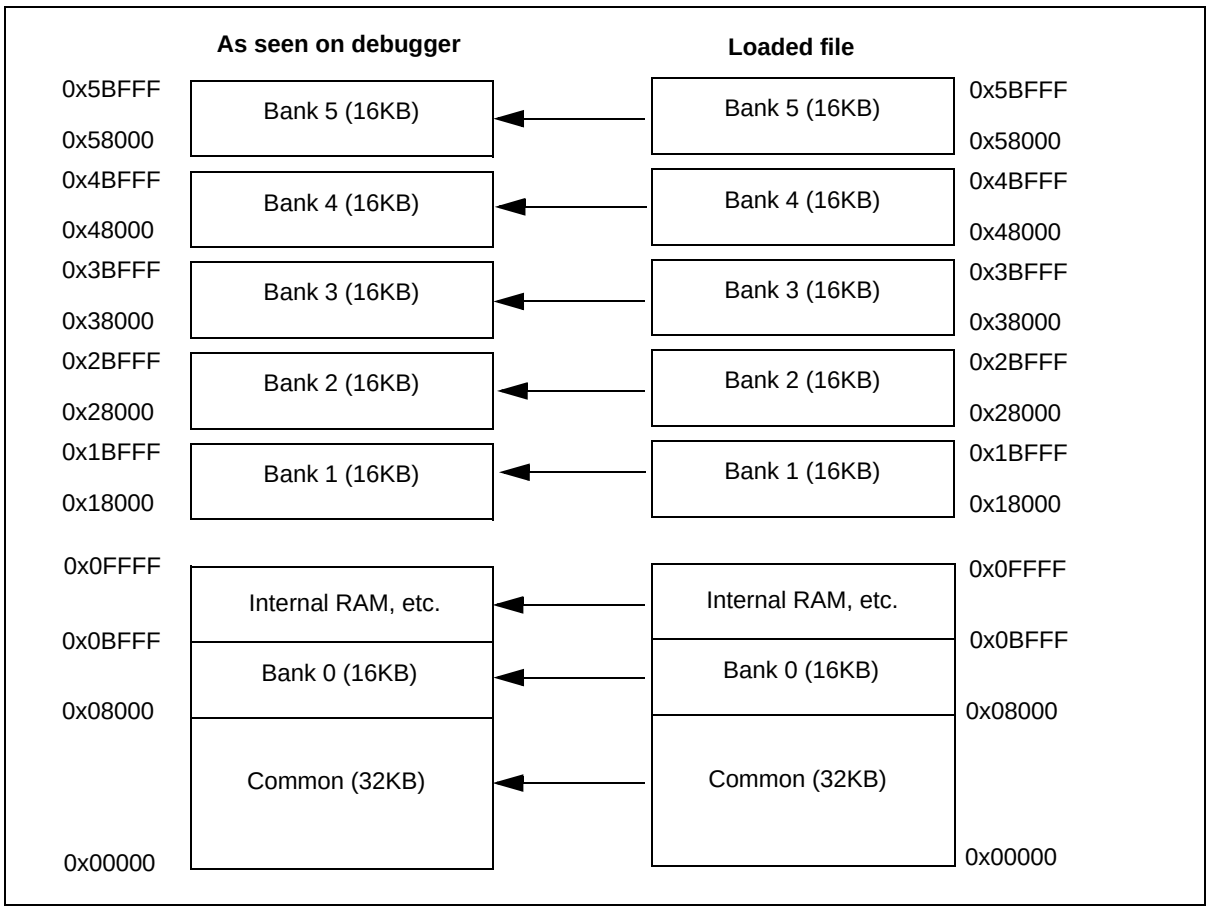
(2) Allocation image when memory banks are used

The figure below shows an allocation image when a Hex format or binary data format file is downloaded when memory banks are used.

[When "Hex Format [Bank] (*.hex)" or "Binary Data [Bank] (*.bin)" is specified]
(Internal Bank ROM Size: 96 KB)



[When "Hex Format [64KB] (*.hex)" or "Binary Data [64KB] (*.bin)" is specified]



5.2.2 Upload

Uploading of memory contents, etc., is performed in the Upload Dialog Box. The saving range can be set.

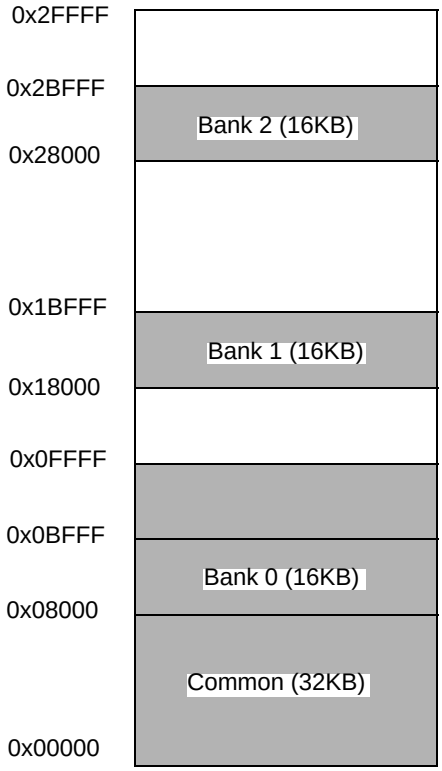
(1) Format of file that can be uploaded

Format of file that can be uploaded is as follows:

Table 5-4 Type of File That Can Be Uploaded

Extension	Format
Intel Hex (*.hex)	Intel Hex format Extension (20-bit addresses)
Motorola Hex (*.hex)	Motorola Hex format S type (S0,S2,S8 -24 bit-address)
Tektro Hex (*.hex)	Extended Tektronix Hex format
Binary Data (*.bin)	Binary data
Coverage (*.cvb)	Coverage results (when IECUBE is connected)

Remark: If a coverage result is uploaded when memory banks are used, all the memory areas (shaded portions in the figure below) are saved (when internal bank ROM size is 48 KB) (when IECUBE is connected).

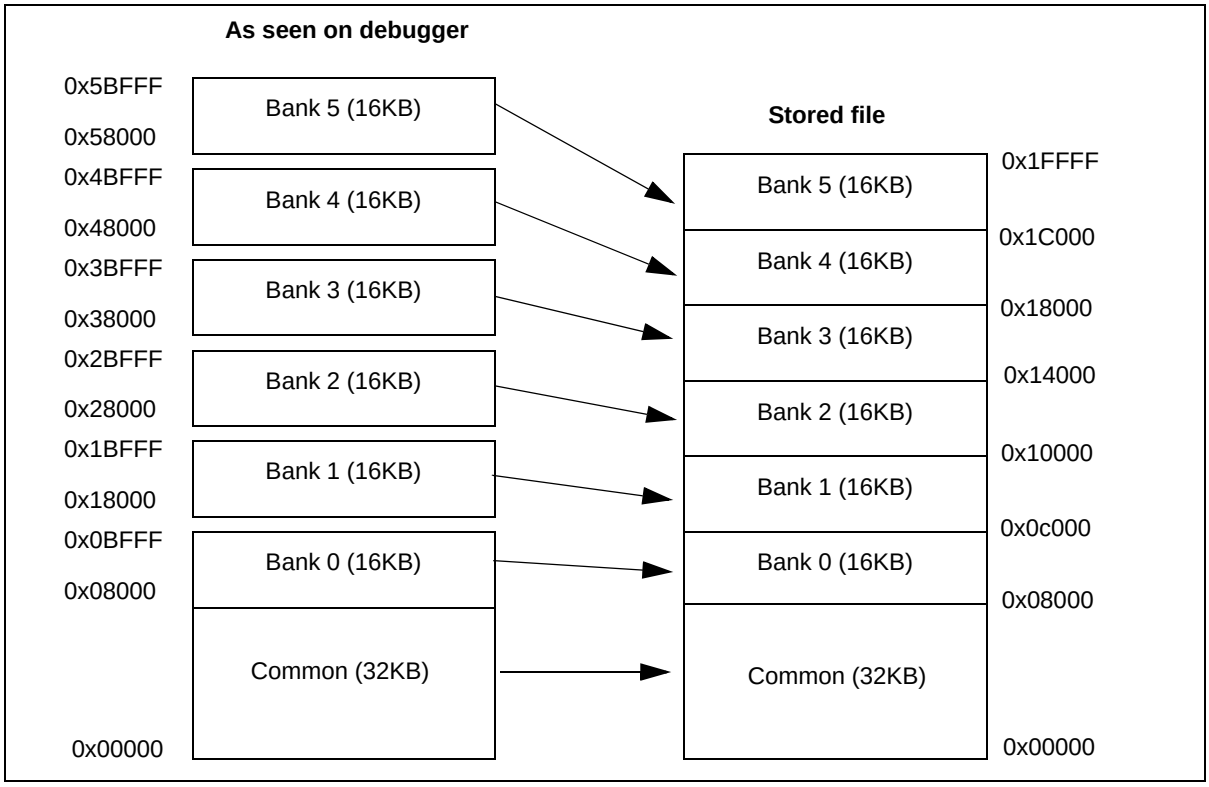


(2) Storage image when memory banks are used

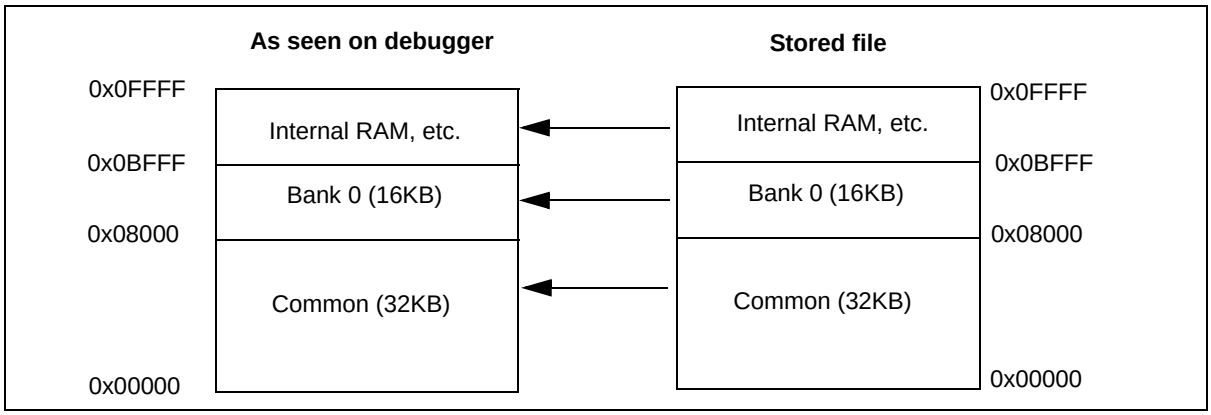
The figure below shows a storage image when a Hex format or binary data format file is uploaded when memory banks are used.

The storage result differs depending on whether a 0x10000 or higher address or an address lower than 0x10000 is specified as the end address of the stored address range, as shown below.

[When a 0x10000 or higher address is specified as the end address]



[When an address lower than 0x10000 is specified as the end address]



5.3 Source Display, Disassemble Display Function

Source file display is performed in the Source Window. Disassemble display and line assembly are performed in the Assemble Window.

This section explains the following items:

- Source display
- Disassemble display
- Mixed display mode (Source Window)
- Convert symbol (symbol to address)

Remark: The locations for which coverage measurement is executed in the user program are displayed in the Source Window and Assemble Window (refer to "5.11.3 Display of locations for which coverage measurement is executed").

5.3.1 Source display

The corresponding text file is displayed in the Source Window by downloading a load module file having debug information.

The display start position can be changed in the Source Text Move Dialog Box displayed by selecting [View] menu -> [Move...].

Specifications related to the tab size, display font, etc., and specification of the source path are made in the Debugger Option Dialog Box. Specify a searching method in the Source Search Dialog Box opened by clicking the <Search...> button. The search result is highlighted in the Source Window.

Table 5-5 File Type Can Be Displayed

File Type (Extension)	Meaning
Source (*.c, *.s, *.asm)	Source file (The extension can be changed in the Debugger Option Dialog Box.)
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.

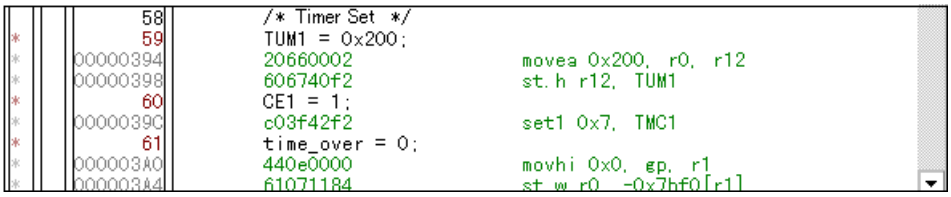
5.3.3 Mixed display mode (Source Window)

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

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

Mixed display mode



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

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

Remark: When scrolling is performed using the cursor keys in the Mixed display mode, excessive scrolling may occur. Also, scrolling down to the last line may not be possible using the cursor keys.

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 5-6 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	equsym file#equsym(to specify a local EQU symbol with file name)
Bit symbol	bitsym file#bitsym(to specify a local bit symbol with file name)
Line number of source file	file#no prog$file#no
I/O port name	portname
SFR name	sfrname
Register name	regname
PSW flag name	pswname

Remark: Separator "#"

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

5.4 Break Function

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

This section explains the following items:

- Break types
- Breakpoint setting
- Setting breaks to variables
- Hardware break and software break
- Fail-safe break

5.4.1 Break types

The ID78K0-QB has the following break functions.

Table 5-7 Break Types

Item	Contents
Hardware break **Note1** (Event detection break)	Function to stop user program execution upon detection of the set break event condition. -> Refer to "5.4.2 Breakpoint setting".
Software break **Note1**	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"). -> Refer to "5.4.2 Breakpoint setting".
[Come Here] break **Note2** (Simple break)	Function to stop user program execution selected by selecting [Run] menu -> [Come Here] upon detection of address specified in the Source Window the Assemble Window.
Break on satisfaction of condition of step execution	Function to stop execution upon satisfaction of the stop condition of each command ([Step In], [Next Over], [Return Out], [Slowmotion]).
Forced break	Function to forcibly stop execution by selecting [Run] menu -> [Stop], or selecting the STOP button. It is valid for all the execution commands.
Fail-safe break (when IECUBE is connected)	Function to forcibly stop execution when the user program performs an illegal operation in relation to the memory or registers (refer to "5.4.5 Fail-safe break"). -> Refer to " Fail-safe Break Dialog Box".
Time-out break (when IECUBE is connected)	Function to stop user program execution when the measurement time exceeds the specified time-out time (refer to " Timer Dialog Box").

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

Breakpoints can simply be set to the desired location by clicking in the Source Window or Assemble Window.

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

Caution: When MINICUBE is connected, no breakpoints can be set or deleted during user program execution.

(1) Breakpoint setting method

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

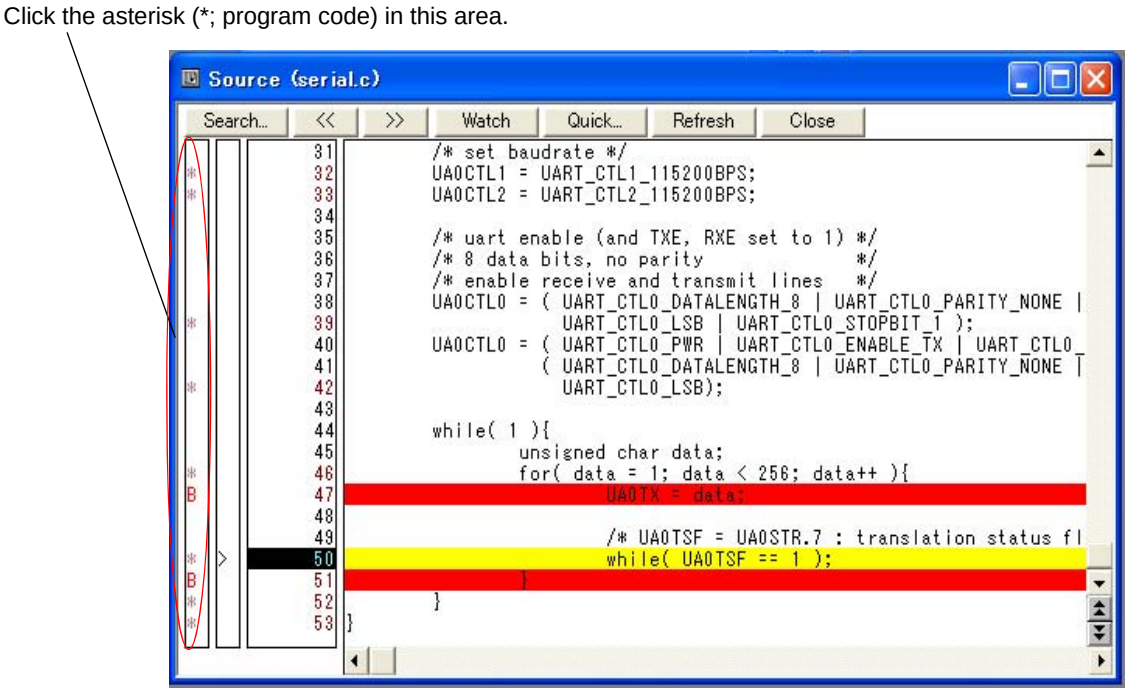
In the default setting, software breakpoint (B) is set, but if [Breakpoint] is selected in the context menu, hardware breakpoint (B, or B) is set.

If a breakpoint is set on a line on which an event breakpoint has already been set, "A" indicating that multiple events have been set is marked (refer to "Table 6-7 Event Setting Status (Event Mark)").

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

Remark: Breaks set by default can also be changed in the Extended Option Dialog Box.

Figure 5-1 Breakpoint Setting



(2) Deleting a breakpoint method

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

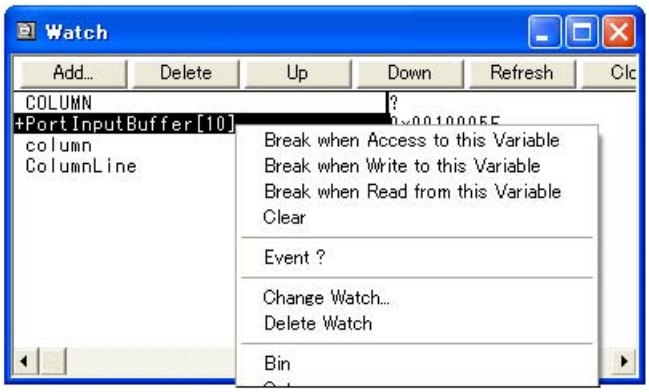
At the same time as setting is performed, in the default setting, software breakpoint (B) is deleted, but if [Breakpoint] is selected in the context menu, hardware breakpoint (B, or B) is deleted.

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

5.4.3 Setting breaks to variables

Access breaks can easily be set to variables from the context menu in the Source Window or Watch Window.

Figure 5-2 Setting Break to Variable



5.4.4 Hardware break and software break

(1) Hardware break

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

Therefore, in the ID78K0-QB, they are managed using "5.12 Event Function" as break event conditions.

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

(2) Software break

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

Caution: If a software break is set to the execution start address in order to re-execute the code in the execution start address, the following events which are set to the address are not generated (when IECUBE is connected).

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

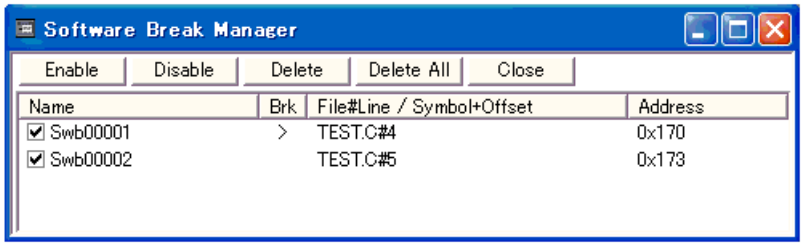
To re-execute the code in the execution start address, use a break for the event before execution.

Table 5-8 The Number of Valid Software Break

Connected IE	Valid Number
IECUBE MINICUBE	2000

Software break is managed by the Software Break Manager.

Figure 5-3 Management of Software Breaks



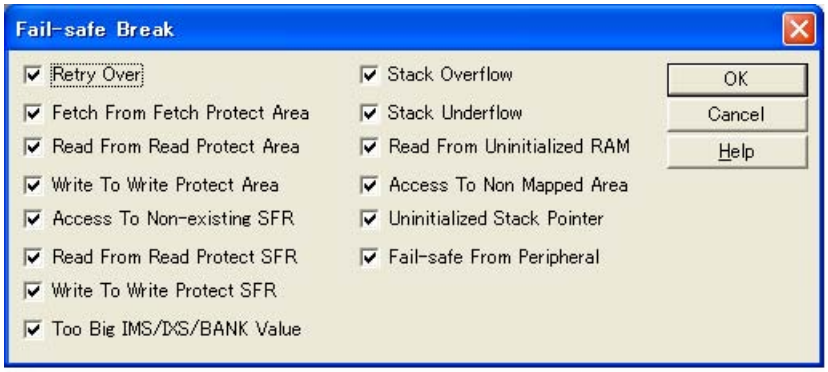
5.4.5 Fail-safe break

Fail-safe break function to forcibly stop execution when the user program performs an illegal operation in relation to the memory or registers.

The fail-safe break settings are performed in the Fail-safe Break Dialog Box.

Individual settings are possible by selecting check boxes.

Figure 5-4 Fail-safe Break Setting



5.5 Program Execution Function

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

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

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

The following types of ID78K0-QB execution functions are provided. They are operated using the execution buttons on the tool bar , or from the [Run] menu.

Figure 5-5 Execution Button

Figure 5-6 [Run] Menu

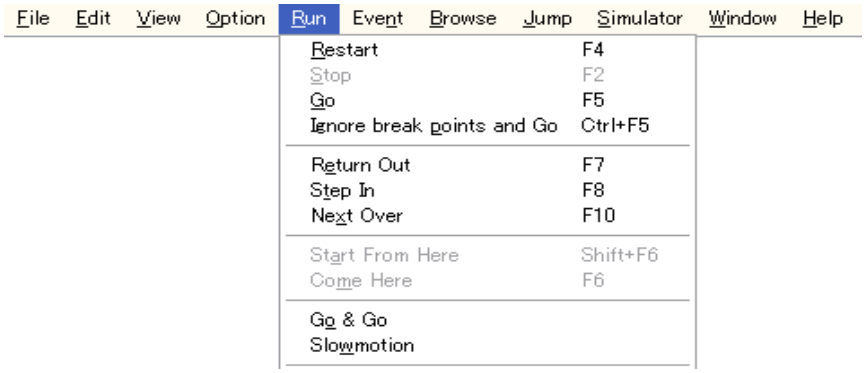


Table 5-9 Type of Execution

Items	Contents
[Restart] 	The CPU is reset and the user program is executed starting from address RESET. This is the same operation as "executing [CPU Reset]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 execution continues until a break condition (refer to "5.4.1 Break types") is established.
[Ignore break points and Go] 	The user program is executed starting from the address indicated by the current PC register Execution of the user program continues, ignoring set breakpoints.
[Return Out] 	The user program is executed until execution returns to the calling function described in C language.
[Step In] 	In the source mode, step execution of one line of the source text is performed starting from the current PC register value and the contents of each window are updated. In the instruction mode, one instruction is executed from the current PC register value and the contents of each window are updated.
[Next Over] 	**CALL / CALLT / CALLF instruction** 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). **Instruction other than CALL / CALLT / CALLF** The same processing as [Step In] is performed.
[Start From Here]	This command executes the user program starting from the specified address. Execution of the user program is stopped when a set break event condition is satisfied.
[Come Here]	The user program is executed from the address indicated by the current PC register to the address selected in the line/address display area of the Source Window or Assemble Window, and then a break occurs. While the user program is being executed, the break event currently set does not occur.
[Go & Go]	The user program is executed starting from the address indicated by the current PC register and stopped if a set break event condition is satisfied. The contents of each window are updated, and execution of the user program is resumed from the address at which the program was stopped. This operation is repeated until the user executes [Stop].
[Slowmotion]	Step execution of one line is performed from the address indicated by the current PC register value in the source mode. In the instruction mode, step execution of one instruction is performed. The contents of each window are updated each time step execution is performed. This operation is repeated until the user executes [Stop].
[CPU Reset] 	Resets the CPU.
[Stop] 	Forcibly stops program execution.

5.6 Watch Function

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

- Displaying and changing data values
- Displaying and changing local variable values
- Registering and deleting watch data
- Changing watch data
- Temporarily displaying and changing data values
- Callout watch function
- Stack trace display function

5.6.1 Displaying and changing data values

Data values are displayed and changed in the Watch Window. Shifts in data values can be checked by registering watch data.

The display format is specified in the Debugger Option Dialog Box.

Figure 5-7 Watch Window

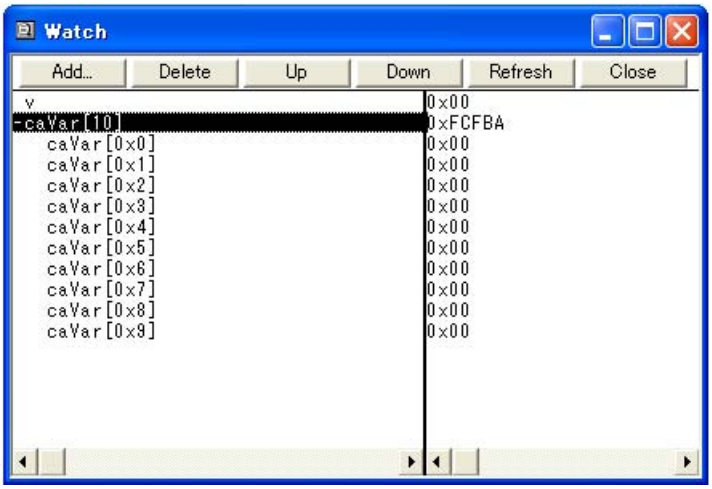
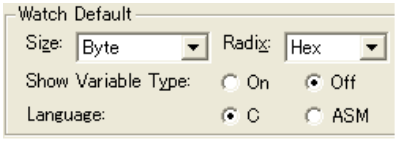


Figure 5-8 Specification of the Display Format (Debugger Option Dialog Box)

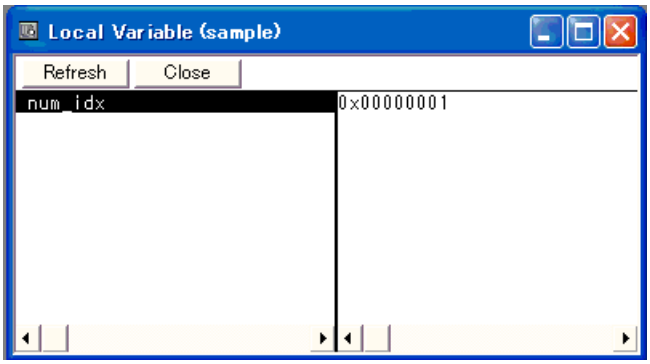


5.6.2 Displaying and changing local variable values

Local variables are displayed and changed in the Local Variable Window.

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

Figure 5-9 Local Variable Window



5.6.3 Registering and deleting watch data

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

- Drag and drop the selected variable or symbol name directly on the Watch Window. (Refer to "5.18.4 Drag & drop function".)
- Click the <Add> button in the Quick Watch Dialog Box or Add Watch Dialog Box.

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

5.6.4 Changing watch data

Watch data is changed in the Change Watch Dialog Box.

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

Figure 5-10 Change Watch Dialog Box

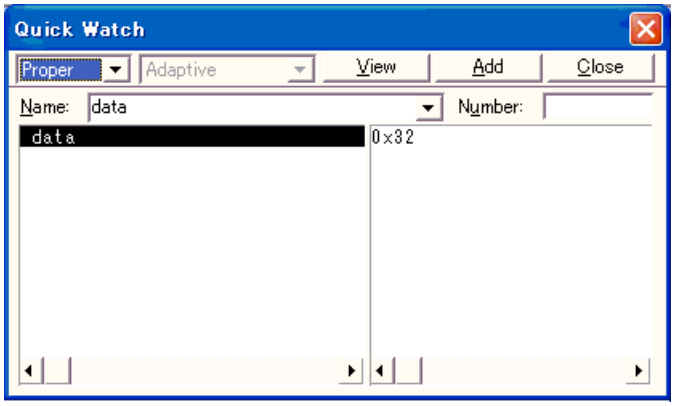
5.6.5 Temporarily displaying and changing data values

Data values are temporarily displayed and changed in the Quick Watch Dialog Box.

Select the desired variable or symbol name in the Source Window or Assemble Window and click the <Quick...> button to perform watch data registration.

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

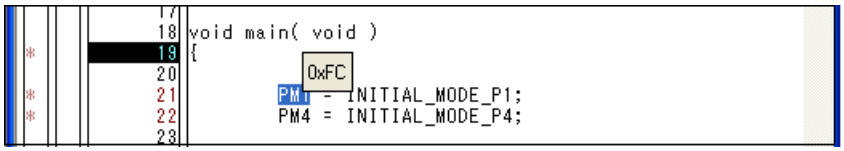
Figure 5-11 Quick Watch Dialog Box



5.6.6 Callout watch function

The corresponding variable value pops up when the mouse cursor is placed over a selected variable in the Source Window or Assemble Window.

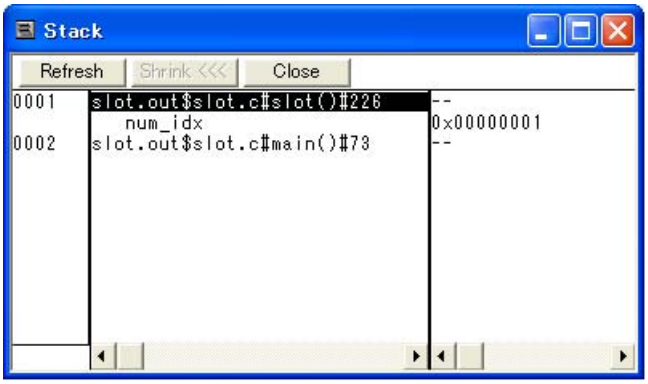
Figure 5-12 Callout Watch Function



5.6.7 Stack trace display function

This function displays the stack contents of the current user program in the Stack Window.

Figure 5-13 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
- Access monitor function (when IECUBE is connected)
- Flash memory writing function (when MINICUBE is connected)

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 "5.15 RRM Function".)

Remark: The address width changes when memory banks are used.

5.7.2 Filling, copying, and comparing memory contents

Memory contents are Filled, copied, and compared in the Memory Fill Dialog Box, Memory Copy Dialog Box, and 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.

Remark: The address width changes when memory banks are used.

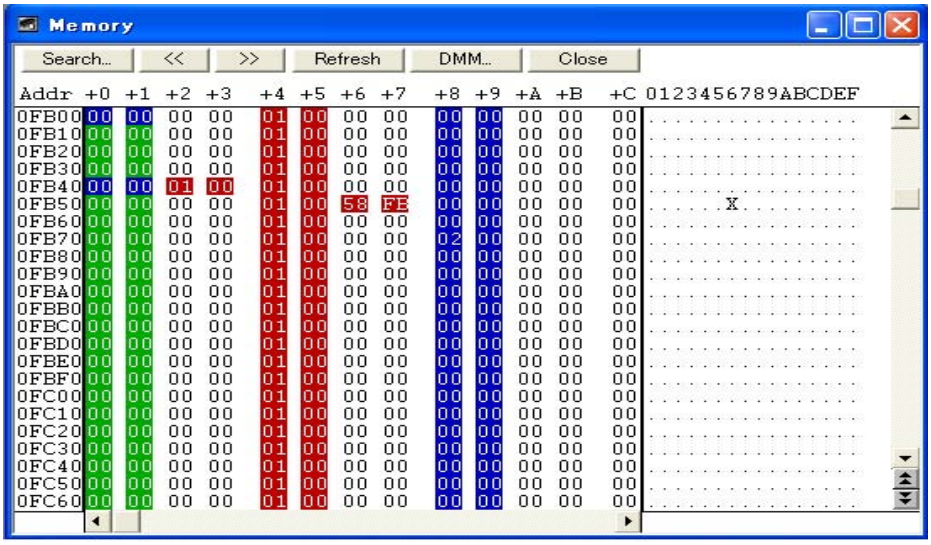
5.7.3 Access monitor function (when IECUBE is connected)

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

Caution: The value of memory rewritten via DMA during program execution, and the value of memory rewritten from the ID78K0-QB cannot be displayed on the access monitor.

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

Figure 5-14 Access Monitor Function (Memory Window)



5.7.4 Flash memory writing function (when MINICUBE is connected)

With the ID78K0-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 (refer to " Flash Option Dialog Box").

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 5-15 Absolute Name/Function Name Switching

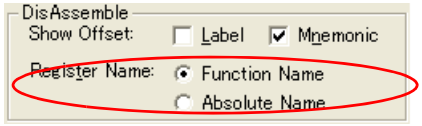


Table 5-10 Absolute Name to Function Name Correspondence

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

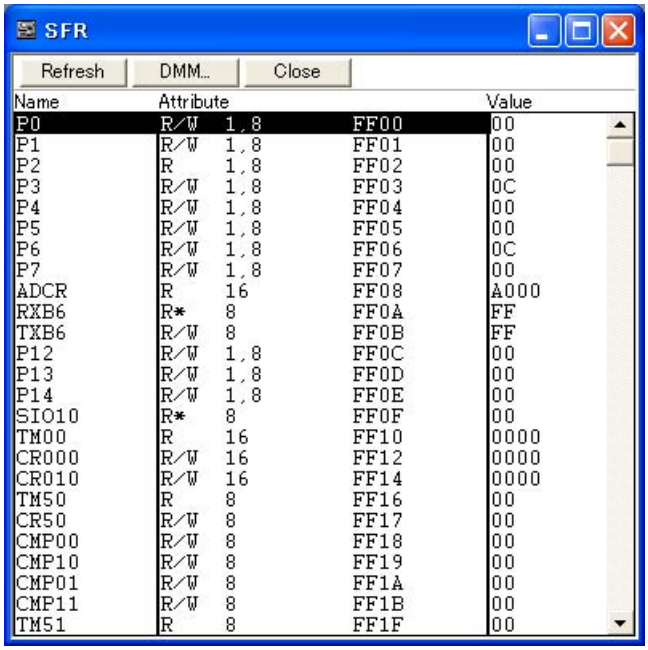
5.8.2 Displaying and changing SFR contents

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 5-16 Displaying SFR Contents

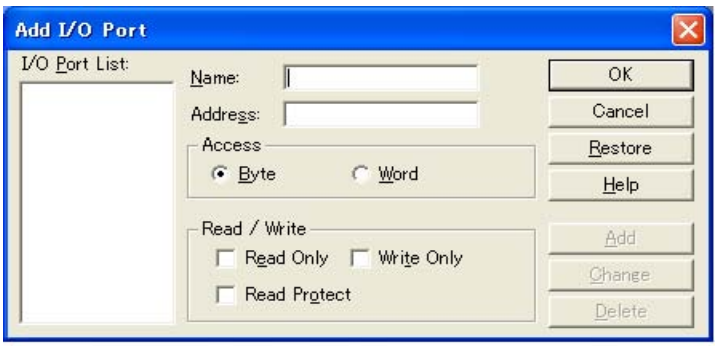


5.8.3 Displaying and changing I/O port contents

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 5-17 Register I/O Port



5.9 Timer Function (When IECUBE Is Connected)

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

The Run-Break time is also displayed in the status bar in the Main window.

This section explains the following items:

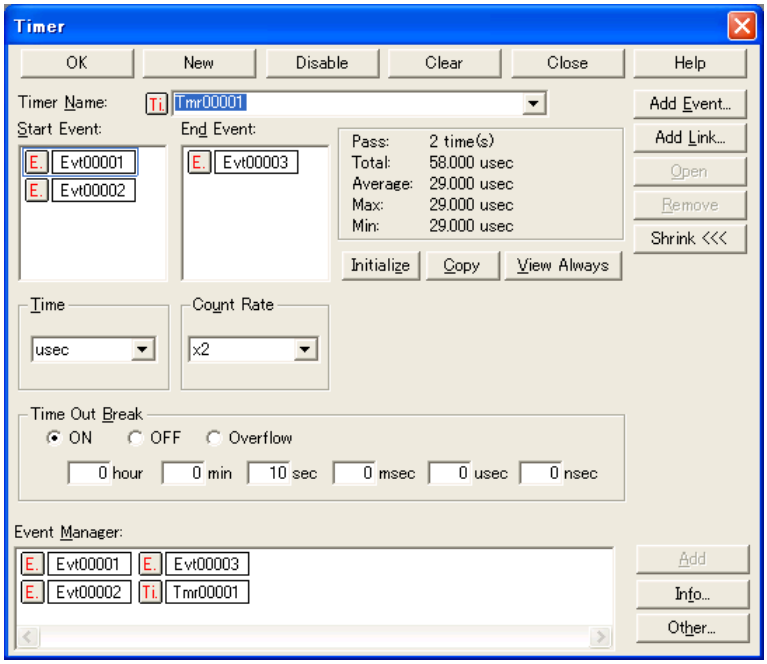
- Timer event conditions
- Run-Break event

5.9.1 Timer event conditions

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

In the ID78K0-QB, timeout break settings can be performed in the Time Out Break area.

Figure 5-18 Sets and Displays Timer Event (Timer Dialog Box)



Continuous display in the Timer Result Dialog Box can be selected by clicking the <View Always> button.

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

5.9.2 Run-Break event

Run-Break event is a timer event name given to a timer event condition that measures the execution time from execution to break. Run-break events are registered in advance and the run-break time can be displayed through specification in the Timer Dialog Box.

The Run-Break time is also displayed in the status bar in the Main window.

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

5.10 Trace Function (When IECUBE Is Connected)

The trace function is used to save the history of the data indicating the execution process of the user program to the trace memory. This section explains the following items:

- Trace memory
- Checking Trace Data
- Mixed display mode (Trace View Window)
- Tracer operation
- Setting conditional trace

5.10.1 Trace memory

ID78K0-QB has trace memory with a ring structure. The maximum trace memory capacity is as follows.

Table 5-11 Trace Memory Size

Connected IE	Maximum Value
IECUBE	128 KB

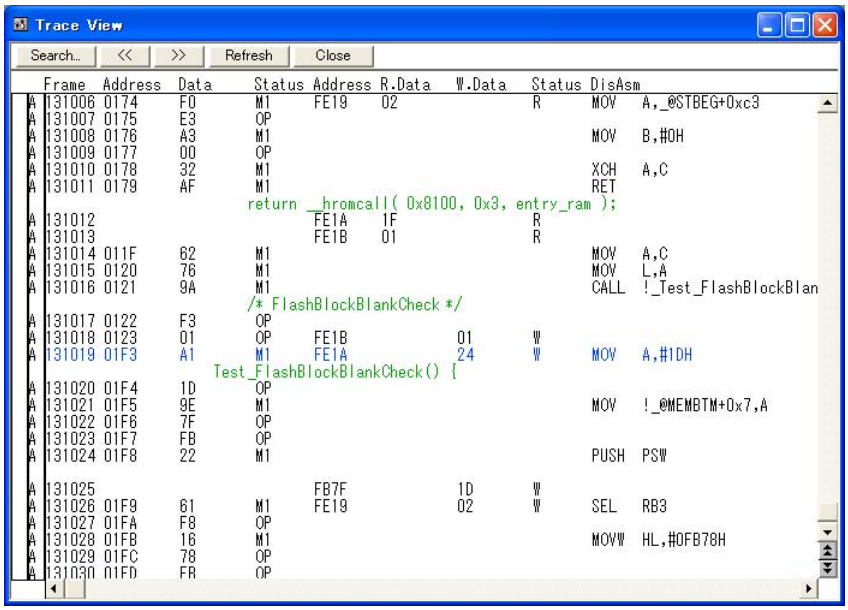
5.10.2 Checking Trace Data

The trace data saved to the trace memory can be checked in the Trace View Window. Trace data can be searched in the Trace Search Dialog Box displayed by clicking the < Search...> button.

The display start position can be changed in the Trace Move Dialog Box displayed by selecting [View] -> [Move].

The display items in the Trace View Window can be selected in the Trace Data Select Dialog Box.

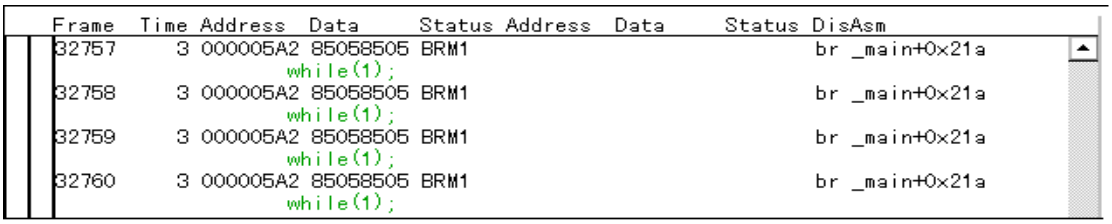
Figure 5-19 Checking Trace Data



5.10.3 Mixed display mode (Trace View 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.

5.10.4 Tracer operation

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

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

(1) Operation during execution

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

Table 5-12 Types of Trace Mode

Item	Contents
Unconditional trace	Trace is started when execution of user program, and ends when a break occurs. At this time, the set trace event conditions are ignored.
Conditional trace	Trace is started or stopped by the condition set in the Trace Dialog Box (refer to "5.10.5 Setting conditional trace"). If a break occurs while a trace is being executed, however, trace is stopped immediately.

(2) Operation during Step In execution

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

(3) Operation during Next Over execution

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

(a) CALL/CALLT/CALLF instructions

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

(b) Other instructions

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

(4) Tracer control mode

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

Table 5-13 Types of Tracer Control Mode

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

5.10.5 Setting conditional trace

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

By setting a trace event condition in the Trace Dialog Box, the conditional trace can be set (refer to "5.12 Event Function").

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

There are the following types of conditional trace.

In the ID78K0-QB, all the following conditional traces can be set as one trace event condition.

Table 5-14 Types of Conditional Trace

Item	Contents, Setting Method
Section trace	Executes a trace between two specified conditions (in a specific zone). A section trace can be executed by setting a trace start event and trace end event in the Trace Dialog Box and selecting [Run] -> [Cond. Trace ON]. In the ID78K0-QB, up to 4 ranges can be specified simultaneously.
Qualify trace	Executes a trace only when a condition is satisfied. If two or more events are set as qualify trace events, a qualify trace can be executed by executing a conditional trace. A qualify trace can be executed by setting a qualify trace event in the Trace Dialog Box and selecting [Run] -> [Cond. Trace ON].
Delay trigger trace	Executes a trace by the number of delay counts after a condition has been satisfied. A delay trigger trace can be executed by setting a delay trigger event in the Trace Dialog Box, setting a delay count in the Delay Count Dialog Box and selecting [Run] -> [Cond. Trace ON]. In this case, select [Delay Trigger Stop] , [Delay Trigger Break].

5.11 Coverage Measurement Function (When IECUBE Is Connected)

Although there are several types of coverage measurement, the ID78K0-QB performs measurement for C0 coverage.

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

Download or upload the coverage measurement result (coverage data) via the Download Dialog Box or Upload Dialog Box, respectively.

Remark: Coverage data is cleared when the debugger is started.

This section explains the following items:

- Coverage mesurement result display
- Coverage measurement range
- Display of locations for which coverage measurement is executed

5.11.1 Coverage mesurement result display

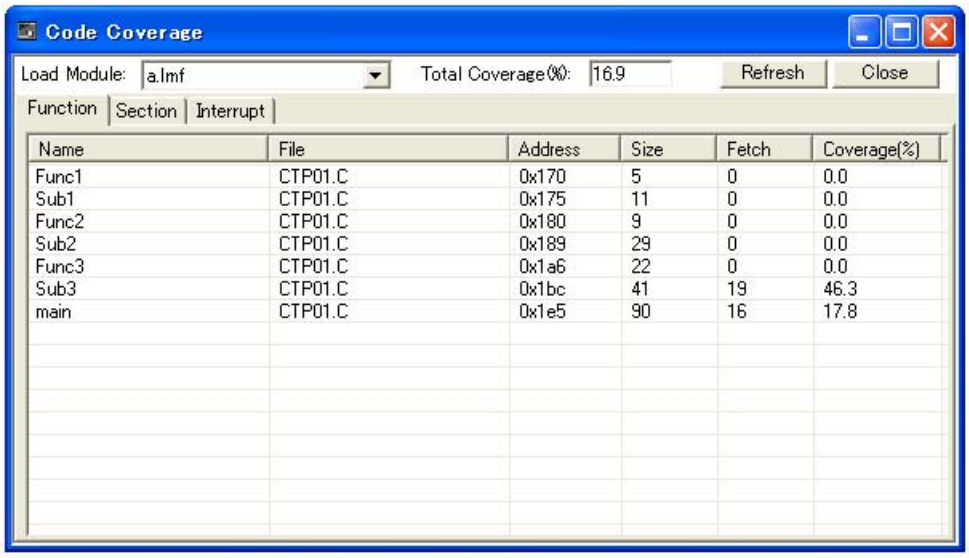
The coverage measurement result can be checked in the Code Coverage Window.

In the Code Coverage Window, the measurement result is displayed individually for functions, sections, and interrupt handlers (vectors). The coverage measurement result is updated at a break (it is not updated automatically during user program execution).

Clear the coverage data by selecting [Coverage Clear] from the [Option] menu.

Coverage data can be saved in the CSV format (refer to "5.17.2 Window display information (view file)").

Figure 5-20 Coverage Mesurement Result Display



5.11.2 Coverage measurement range

The coverage measurement range is as follows.

Table 5-15 Code Coverage Measurement Range

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

5.11.3 Display of locations for which coverage measurement is executed

The locations for which coverage measurement is executed in the user program are displayed in the Source Window and Assemble Window based on the coverage measurement information.

The display result can be saved as view files for the Source Window and Assemble Window (refer to "5.17.2 Window display information (view file)").

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

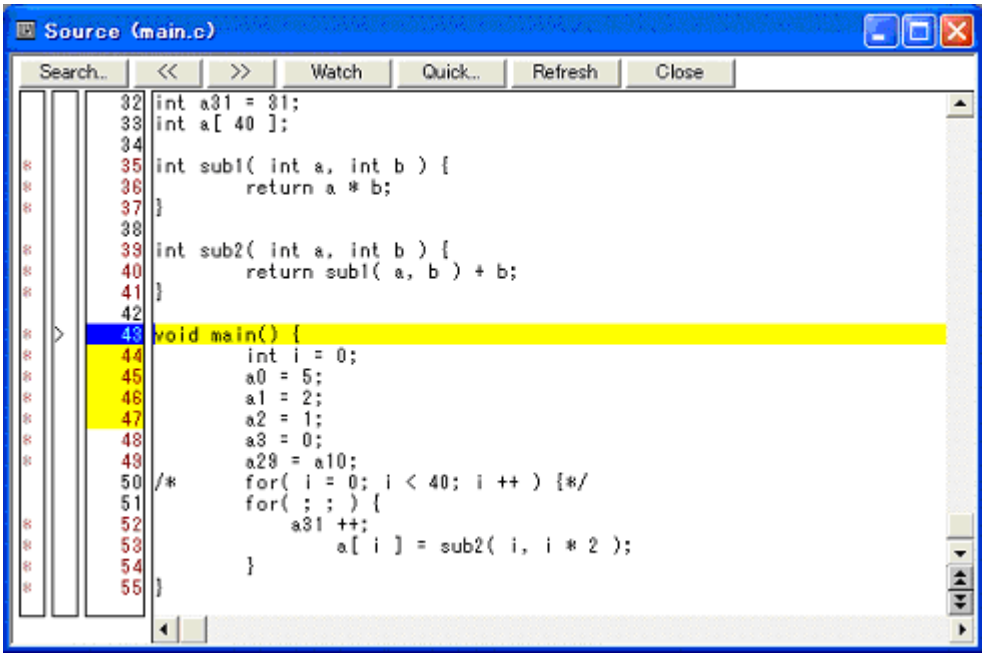
Table 5-16 Format of View of Locations for Which Coverage Measurement Executed

Coverage	Source Window		Assemble Window	
	Background Color	Mark	Background Color	Mark
Code on this line has been executed by 100%	Yellow	'@'	Yellow	'@'
Code on this line has been executed by 1 to 99%	Orange	'+'		
Code on this line has been executed by 0% (not yet executed)	No highlight color	None	No highlight color	None

Caution: The instruction (unexecuted instruction) pointed to by the PC is judged to be executed partially and displayed for coverage measurement information (it is highlighted with a background color in the Source Window and Assemble Window, or its coverage becomes other than 0 in the Code Coverage Window).

Remark: With the ID78K0-QB, the cumulative view of the locations for which coverage measurement has been executed can be displayed and coverage data can be cleared using the context menu on the Source Window or Assemble Window.

Figure 5-21 View of Locations for Which Coverage Measurement Executed



5.12 Event Function

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

In ID78K0-QB, such events are used as action triggers for each debugging function, such as break and trace.

This section explains the following items:

- Using event function
- Creating events
- Setting event conditions
- Number of enabled events for each event condition
- Managing events

5.12.1 Using event function

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

Table 5-17 Various Event Conditions

Event Condition	Mark	Contents ->Setting Dialog Box
Break event	B	Condition in which the execution of the user program or operation of a tracer is stopped. (refer to "5.4 Break Function"). ->Break Dialog Box
Trace event	T	Condition in which the process of user program execution is saved to the trace memory (refer to "5.10 Trace Function (When IECUBE Is Connected)"). ->Trace Dialog Box
Timer event	Ti	Condition for specifying the time measurement start timing and stop timing (refer to "5.9 Timer Function (When IECUBE Is Connected)"). ->Timer Dialog Box
Stub event	U	Condition for specifying the timing of inserting the program (refer to "5.14 Stub Function (When IECUBE Is Connected)"). ->Stub Dialog Box
Snapshot event	S	Condition for specifying the timing of executing a snapshot (refer to "5.13 Snapshot Function (When IECUBE Is Connected)"). ->Snap Shot Dialog Box
Event DMM	M	Condition for specifying the timing of executing DMM (refer to "5.16 DMM Function"). ->Event DMM Dialog Box

5.12.2 Creating events

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

(1) Creating and registering events

The creation of event conditions is done in the Event Dialog Box.

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

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

(2) Creating and registering event links

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

To create an event link condition, use the Event Link Dialog Box.

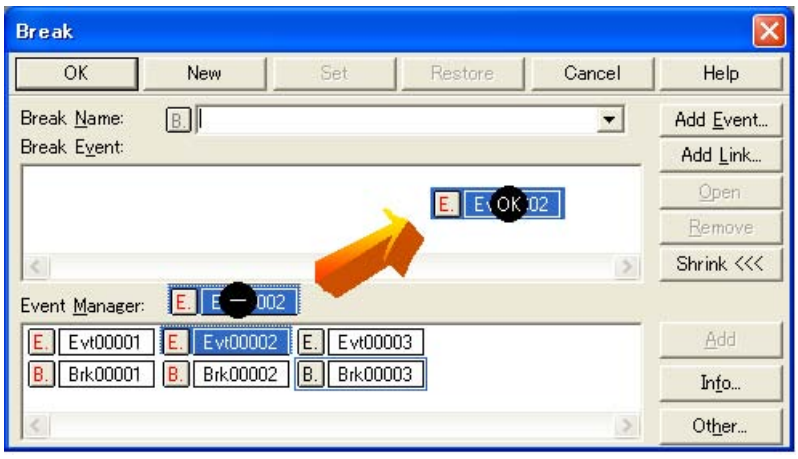
5.12.3 Setting event conditions

Various event conditions listed in Table 5-17 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 5-22 Setting of Various Event Conditions



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

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

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

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

(3) Copying and moving event icons

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

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

(4) Manipulation in event manager area

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

Event setting content display

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

Deletion

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

Changing display mode and sorting

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

Area non-display

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

5.12.4 Number of enabled events for each event condition

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

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

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

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

Caution: Except for execution before events, event conditions for which the address range or data range has been set use 2 event conditions internally, so the number of event conditions that can be used simultaneously will be reduced according to the number of those event conditions.

For an event before execution, however, only one event condition can be set, even if the address range is set.

Table 5-18 Number of Enabled Events for Each Event Condition

Connected IE	Event		Event Link	Break	Trace	Snapshot	Stub	Timer	Event DMM
	Execute	Access							
IECUBE	8	10[a.]	2	34[b.]	1[c.]	3	3	2	4
MINICUBE	1	1	-	2[d.]	-	-	-	-	-

a. 8 dedicated byte access events and 2 dedicated word access events

b. Including 16 break before executions

c. Up to 4 section trace events, 1 qualify trace, and 1 delay trigger trace can be set simultaneously.

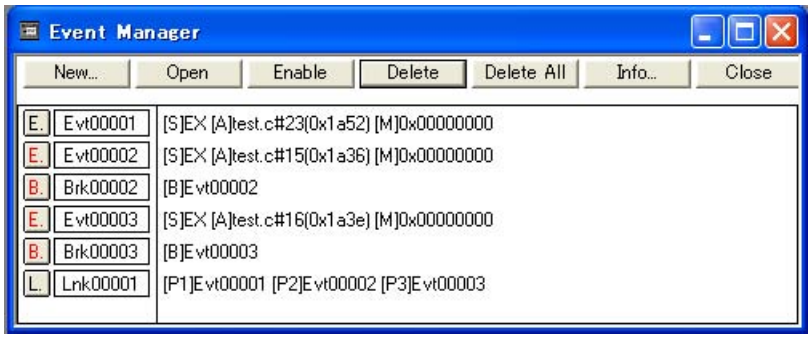
d. One is a break before execution (exclusive with software break)

5.12.5 Managing events

Managing events is done with the Event Manager.

The Event Manager allows display, enabling/disabling, and deletion of the Various Event Conditions.

Figure 5-23 Managing Events (Event Manager)



(1) Event icon

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

Enable/disable is switched by clicking the mark part.

Table 5-19 Event Icon

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

5.13 Snapshot Function (When IECUBE Is Connected)

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

This section explains the following items:

- Snapshot event conditions
- Snap data

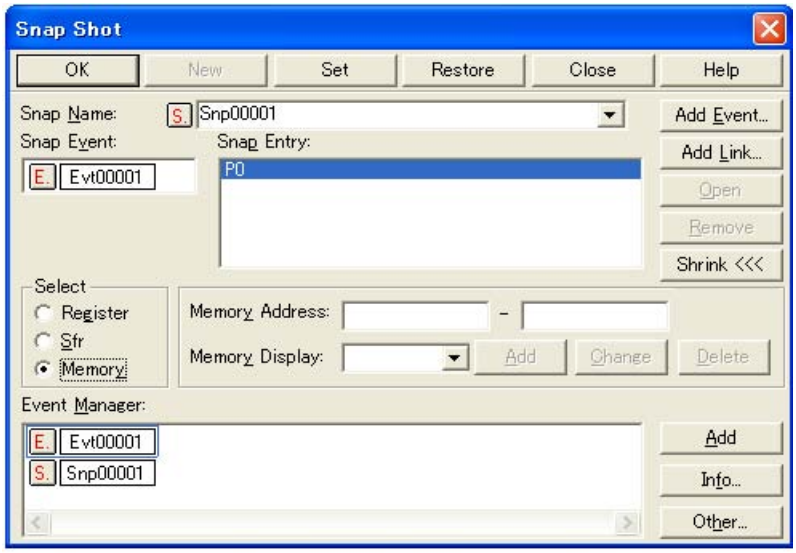
5.13.1 Snapshot event conditions

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

Snapshot event conditions are set in the Snap Shot Dialog Box. (Refer to "5.12 Event Function".)

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

Figure 5-24 Snap Shot Dialog box



5.13.2 Snap data

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

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

- Register value
- SFR value
- Memory contents

5.14 Stub Function (When IECUBE Is Connected)

The stub function is used to execute the user program (sub-program) that has been downloaded or written to a vacant memory area in advance when an event occurs.

Caution: Append the RETB instruction to the end of the sub-program that is executed when a stub event occurs, otherwise, malfunction may occur.

This section explains the following items:

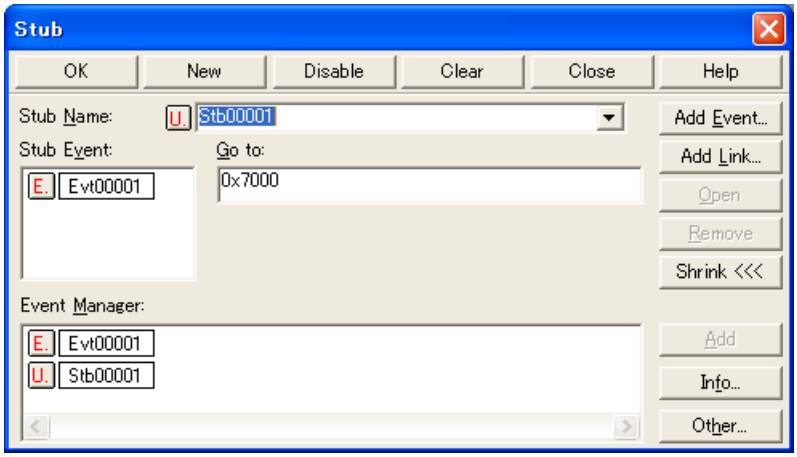
- Setting stub event conditions
- Flow of stub function

5.14.1 Setting stub event conditions

A stub event condition specifies the trigger by which the stub function is executed. Stub event conditions are set in the Stub Dialog Box. (Refer to "5.12 Event Function".)

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

Figure 5-25 Setting Stub Function Conditions



The start address of the function executed when a stub event occurs is as follows.

Table 5-20 Start Address of Function to Be Executed (Stub Function)

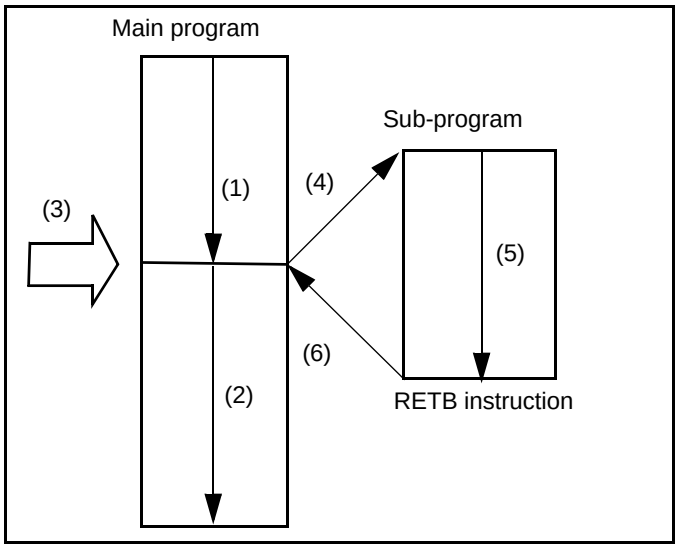
Setting Range
0 <= Start address of function <= 0xFFFF

Remark: The specifiable address range does not change even when memory banks are used. If an address between 0x8000 and 0xBFFF is set, the execution jumps to the corresponding address (current bank) when a stub event is established.

5.14.2 Flow of stub function

A flow of stub function is shown below.

Figure 5-26 Flow of Stub Function



(1) If stub function is not used

(1) and (2) are executed.

(2) If stub function is used

1) (1) is executed.
2) A stub event occurs in (3) and a break occurs.
3) The simulator forms a call stack in the current PC in (4), and rewrites the PC value to the entry address of the sub-program.
4) (5) is executed.
5) Execution is returned to the main program by the RETB instruction at the end of the sub-program in (6).

5.15 RRM Function

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

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

5.15.1 Real-time RAM monitor function

Table 5-21 shows the range of data that can be loaded using the real-time RAM monitor function.

The variables and data allocated to this area can be displayed in real time in the Watch Window and Memory Window.

The sampling interval can be specified in the Extended Option Dialog Box.

Table 5-21 Real-Time RAM Monitor Function Sampling Range

Connected IE	Sampling Range
IECUBE	Internal ROM/internal RAM area, general-purpose registers
MINICUBE	None

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

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 instantaneously 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 Memory Window[Note].

Specify turning on/off of the pseudo real-time RAM monitor function and the sampling range in the Extended Option Dialog Box.

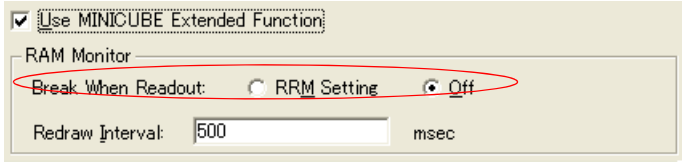
Note: In this case, the target memory area is displayed as "**".

Remark: This function is enabled only when (1) Use MINICUBE Extended Function in the Extended Option Dialog Box is selected (when MINICUBE is connected).

Figure 5-27 Specification of Pseudo Real-Time RAM Monitor Function

(When IECUBE Is Connected)

(When MINICUBE Is Connected)



5.16 DMM Function

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

The DMM Dialog Box is opened by clicking the <DMM... > button on the Memory Window, Register Window, or SFR Window. Specify the DMM target address and data.

Remark1: Pseudo DMM, with which a break occurs instantaneously upon a write during the user program execution, can be selected by checking "Break When Write" in the DMM Dialog Box (when IECUBE is connected).
Use of pseudo DMM is fixed if an area for which a real-time write is not permitted (such as registers and SFR) is specified.

Remark2: When MINICUBE is connected, pseudo DMM can be performed. The pseudo DMM function is enabled only when (1) Use MINICUBE Extended Function is selected in the Extended Option Dialog Box. With pseudo DMM, a break occurs instantaneously upon a write during the user program execution.

Figure 5-28 DMM Dialog Box

This section explains the following items:This section explains the following items:

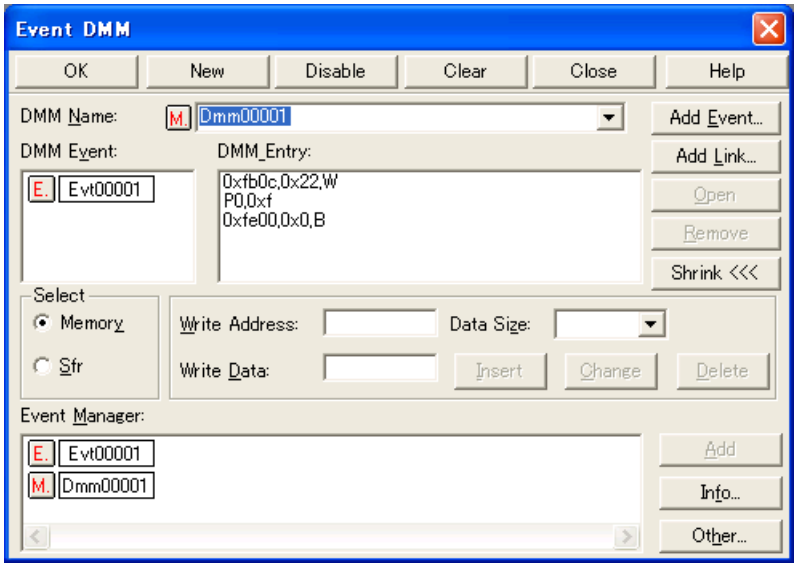
- Event DMM condition (when IECUBE is connected)

5.16.1 Event DMM condition (when IECUBE is connected)

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

Event DMM conditions are set in the Event DMM Dialog Box (refer to "5.12 Event Function").

Figure 5-29 Event DMM Dialog Box



5.17 Load/Save Function

ID78K0-QB allows saving and loading the following types of information as files. As a result, recovery of these various types of information is possible.

- Debugging environment (project file)
- Window display information (view file)
- Window setting information (setting file)

Remark: The simple window status can be maintained by selecting [Window] menu -> [Static]. (Refer to "5.18.1 Active status and static status".)

5.17.1 Debugging environment (project file)

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

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

Project files are loaded and saved in the Project File Save Dialog Box and Project File Load Dialog Box, respectively. To load a project file at startup, click the <Project...> button in the Configuration Dialog Box.

The following contents are saved to the project file:

Table 5-22 Contents Saved to Project File

Window Name	Saved Contents
Configuration Dialog Box	All items (target device, clock setting, pin mask setting, mapping information)
Main window	Display position, tool bar/status bar/button display information, execution mode information
Download Dialog Box	File information to be downloaded
Extended Option Dialog Box Debugger Option Dialog Box Fail-safe Break Dialog Box Pseudo Emulation Dialog Box Mask Option Dialog Box RRM Dialog Box	Set information
Assemble Window Memory Window	Display information of window, display start address
Source Window Stack Window SFR Window Local Variable Window Trace View Window Code Coverage Window Event Manager Console Window Expansion Window Register Window	Display information of window

Window Name	Saved Contents
Event Dialog Box	Display information of window, event information
Event Link Dialog Box	Display information of window, link event information
Break Dialog Box	Display information of window, break event information
Trace Dialog Box	Display information of window, trace event information
Timer Dialog Box	Display information of window, timer event information
Snap Shot Dialog Box	Display information of window, snapshot event information
Stub Dialog Box	Display information of window, stub event information
Event DMM Dialog Box	Display information of window, event DMM information
Watch Window	Display information of window**Note**, watch registration information
Add I/O Port Dialog Box	Added I/O port information
DMM Dialog Box	DMM information
Delay Count Dialog Box	Delay count value
Software Break Manager	Display information of window, software break information

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

5.17.2 Window display information (view file)

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

View files can be loaded and saved for each window.

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

View files are loaded and saved in the View File Load Dialog Box and View File Save Dialog Box, respectively.

Table 5-23 Type of the View Files

File Type	Current Window Name, File Name
Source Text (*.svw)	Source Window **Note1**
Assemble (*.dis)	Assemble Window **Note1**
Memory (*.mem)	Memory Window
Watch (*.wch)	Watch Window
Register (*.rgw)	Register Window
SFR (*.sfr)	SFR Window
Local Variable (*.loc)	Local Variable Window
Stack Trace (*.stk)	Stack Window
Trace (*.tvw)*	Trace View Window
Code Coverage (*.csv)	Code Coverage Window (Data is saved separately for the tab selected)
Console (*.log)	Console Window
All (*.*)	All files
Source (*.c, *.s, *.asm)	Source file**Note2**
Text (*.txt)	Text file

Note1: When IECUBE is connected, the mark for indicating the code coverage measurement result (executed/not yet executed) is added to the contents of the displayed file (refer to "Table 5-16 Format of View of Locations for Which Coverage Measurement Executed").

Note2: The extension of the source file can be changed in the Extended Option Dialog Box.

5.17.3 Window setting information (setting file)

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

Setting files can be loaded and saved for each window.

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

Setting files are loaded and saved in the Environment Setting File Load Dialog Box and Environment Setting File Save Dialog Box, respectively.

Table 5-24 Type of the Setting Files

File Type	Current Window Name
Watch (*.wch)	Watch Window[Note]
SFR (*.sfr)	SFR Window[Note]
Event (*.evn)	Event Manager

Note: A variable value can not be loaded.

5.18 Functions Common To Each Window

The windows have the following common functions.

- Active status and static status
- Jump function
- Trace Result with Linking Window (when IECUBE is connected)
- Drag & drop function
- Cautions

5.18.1 Active status and static status

Each of the Windows below has two statuses: The Active status and Static status.

- Source Window (that is displaying the source file to which symbol information is read)
- Assemble Window
- Memory Window

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

Select this status by the [Window] menu.

(1) Active status

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

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

Only one window can be opened in the active status.

(2) Static status

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

The static window is not used as the jump destination of the Jump function. In addition, it is not linked with the Trace View Window.

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

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

5.18.2 Jump function

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

- Source Window
- Assemble Window
- Memory Window

You can jump among the above windows, or from Trace View Window, Stack Window, Event Manager and Register Window to the above windows.

(1) Jump method

The jump method is as follows:

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

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

(2) Details of jump source address

The details of jump source address is as follows:

Table 5-25 Details of Jump Source Address

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

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

5.18.3 Trace Result with Linking Window (when IECUBE is connected)

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

If the cursor is moved on the Trace View Window, the corresponding part on the linked window is highlighted or indicated by the cursor position.

(1) Linking method

The linking method is as follows:

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

Remark: The linking source address differs as follows depending on the cursor position in the trace result display area if the Memory Window is linked.
Access address, access data, access status -> Access address, Others -> Fetch address
When the Source Window or the Assemble Window is linked, the fetch address is always used as the pointer.

5.18.4 Drag & drop function

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

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

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

(1) Drag & drop details

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

Table 5-26 Details of Drag & Drop Function (Line/Address)

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

Table 5-27 Details of Drag & Drop Function (Character String)

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

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

- Event Dialog Box
- Event Link Dialog Box
- Break Dialog Box
- Trace Dialog Box
- Timer Dialog Box
- Snap Shot Dialog Box
- Stub Dialog Box
- Event DMM Dialog Box

5.18.5 Cautions

(1) The number of characters that can be displayed on 1 line in each area of a window is 319.

(2) If the width of the display area is narrow, the display may become corrupted. In this case, increase the width of the window.

(3) Redrawing may not successfully be performed in a window with a <Refresh> button when the cursor position is changed while the window is active. Click the <Refresh> button to perform redrawing.

(4) The help that is opened using the F1 key is the help corresponding to the window on which the cursor is placed. Consequently, because the cursor cannot be placed on the Trace View Window in which no trace results are displayed, such as immediately after startup, the help may not open even if the F1 key is pressed. In this case, open the help by selecting [Current Window Help] from the [Help] menu.

(5) Do not select [Slowmotion] from the [Run] menu during Go & Go execution. [Slowmotion] on the [Run] menu is usually dimmed during Go & Go execution, but there is a moment when it can be selected, so if [Slowmotion] is selected at this time, the program will not be able to be stopped even if [Stop] is selected from the [Run] menu (or the STOP button is clicked).

(6) If for some reason or other the application switches while event icons are in the process of being dragged, the icons will no longer be able to be dropped.
Use the ESC key to escape from drag, then reattempt the drag.

(7) If a header file in which functions are defined is included into a user program, the following restrictions apply to the defined functions.
(This item derives from cautions for the compiler (CC78K0).)
- No breakpoints can be set to the functions defined in the header file.
- The functions defined in the header file cannot be stepped into by step-wise execution ([Run] menu -> [Step In]).
- The functions defined in the header file are not listed in the function list in the Code Coverage Window.

CHAPTER 6 WINDOW REFERENCE

This chapter explains in detail the functions of the windows and dialog boxes of ID78K0-QB.

- Window List
- Explanation Of Windows

6.1 Window List

The list is the windows of the ID78K0-QB.

Table 6-1 Window List

Window Name	Contents
Main window	This window is displayed first, when the ID78K0-QB is started. It controls execution of the user program. Various windows are opened from this window.
Configuration Dialog Box	Displays and sets the ID78K0-QB operation environment.
Extended Option Dialog Box	Displays and sets the extended options of the ID78K0-QB.
Fail-safe Break Dialog Box	Sets the fail-safe breaks
RRM Dialog Box	Sets the RRM sampling range (MINICUBE)
Mask Option Dialog Box	Sets the mask options and pin mode (IECUBE)
Flash Option Dialog Box	Sets the the flash self programming emulation (IECUBE)
Debugger Option Dialog Box	Displays and sets other options.
Pseudo Emulation Dialog Box	Generation of pseudo emulation (IECUBE)
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 Source Window
Source Text Move Dialog Box	Specifies a file to be displayed in the Source Window and the position from which displaying the file is to be started.
Assemble Window	Disassembles the program and executes line assembly.
Assemble Search Dialog Box	Searches in the Assemble Window
Address Move Dialog Box	Specifies the start address to display the contents of the Memory Window or Assemble Window.
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 todisplay in the Watch Window
Change Watch Dialog Box	Changes watch data todisplay in the Watch Window
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 Memory Window

Window Name	Contents
Memory Fill Dialog Box	Fills the memory contents with specified data.
Memory Copy Dialog Box	Copies the memory.
Memory Compare Dialog Box	Compares the memory.
Memory Compare Result Dialog Box	Displays the results of comparing the memory.
DMM Dialog Box	Sets addresses and data subject to DMM.
Register Window	Displays the contents of registers.
SFR Window	Displays the contents of SFR
SFR Select Dialog Box	Selects SFR and I/O ports to be displayed in the SFR Window
Add I/O Port Dialog Box	Registers an I/O port to be displayed in the SFR Window.
Timer Dialog Box	Registers and sets timer event conditions, and displays execution time measurement result (IECUBE)
Timer Result Dialog Box	Displays execution time measurement results. (IECUBE)
Trace View Window	Displays trace results. (IECUBE)
Trace Search Dialog Box	Searches trace data. (IECUBE)
Trace Data Select Dialog Box	Selects items to be displayed in the Trace View Window.(IECUBE)
Trace Move Dialog Box	Specifies the start address to display the contents of the Trace View Window. (IECUBE)
Trace Dialog Box	Registers and sets trace event conditions. (IECUBE)
Delay Count Dialog Box	Sets the delay count of a delay trigger trace event.(IECUBE)
Code Coverage Window	Display of code coverage results (IECUBE)
Software Break Manager	Display, enable or disable, and delete software breaks.
Event Manager	Displays, enables/disables, and deletes each event condition.
Event Dialog Box	Registers event conditions.
Event Link Dialog Box	Registers event link conditions.(IECUBE)
Break Dialog Box	Registers and sets break event conditions.
Snap Shot Dialog Box	Registers and sets snapshot event conditions. (IECUBE)
Stub Dialog Box	Registers and sets stub event conditions. (IECUBE)
Event DMM Dialog Box	Registers and sets event DMM conditions. (IECUBE)
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 ID78K0-QB,CPU, and symbol information.
Exit Debugger Dialog Box	Terminates the ID78K0-QB.
About Dialog Box	Displays the version of the ID78K0-QB.

Window Name	Contents
Console Window	Inputs commands.
Browse Dialog Box	Selects the file to be set

6.2 Explanation Of Windows

This section explains each window or dialog box as follows:

Window Name / Dialog box Name

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

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

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

Items of related operation are also explained.

Opening

Explains how to open the window or dialog box.

Explanation Of Each Area

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

Context Menu

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

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 ID78K0-QB is started up and initialized.

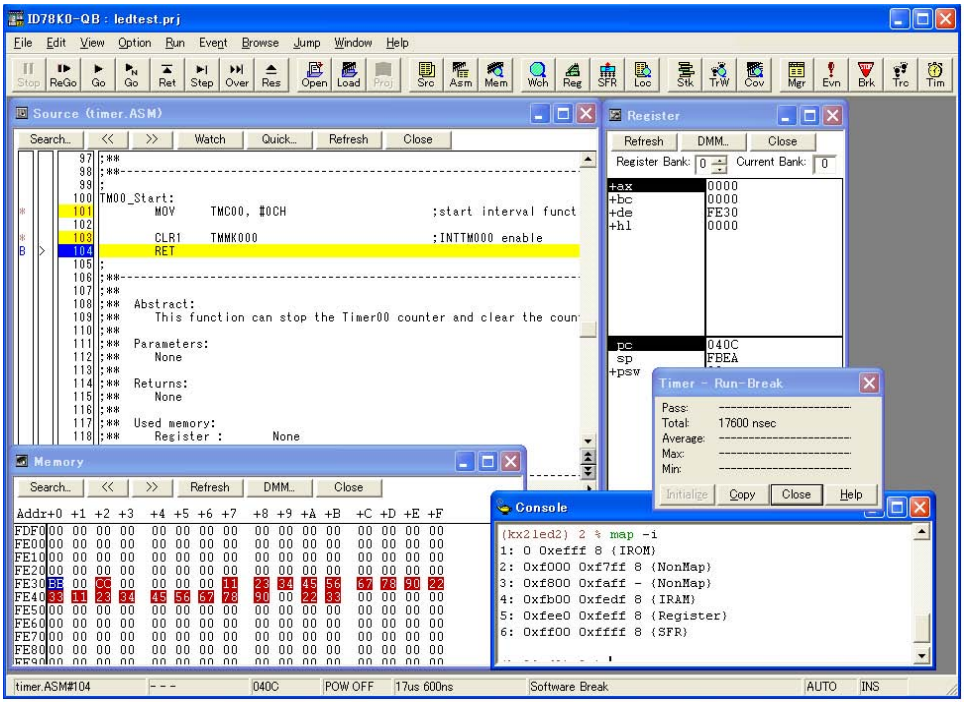
In ID78K0-QB, other windows are manipulated from this window (refer to "Table 6-1 Window List").

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.) (default)

Figure 6-1 Main Window



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

Menu bar

(1) [File] menu

Open...	Loads a view file, source file, or text file. Opens the View File Load Dialog Box. 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 View File Save Dialog Box.
Close	Closes the current window.
Download...	Downloads a file. Opens the Download Dialog Box.
Upload...	Uploads a program. Opens the Upload Dialog Box.
Project	Manipulates a project file.
Open...	Opens a project file. Opens the Project File Load Dialog Box.
Save	Overwrites the current status to the project file currently being read to the ID78K0-QB.
Save As...	Saves the current status to a specified project file. Opens the Project File Save Dialog Box.
Environment	Manipulates a setting file.
Open...	Opens a setting file. Opens the Environment Setting File Load Dialog Box.
Save As...	Saves the setting in the current window to the setting file. Opens the Environment Setting File Save Dialog Box.
Debugger Reset...	Initializes the CPU, symbols, and ID78K0-QB. Opens the Reset Debugger Dialog Box.
Exit	Terminate the ID78K0-QB. Opens the Exit Debugger Dialog Box.
(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 Memory Fill Dialog Box.
Copy...	Copies the memory. Opens the Memory Copy Dialog Box.
Compare...	Compares the memory. Opens the Memory Compare Dialog Box.
DMM...	Rewrites the memory contents to real time during user program execution. Opens the DMM Dialog Box. (Cannot be selected, when MINICUBE is connected and (1) Use MINICUBE Extended Function is cleared.)
Edit Source	Opens the source file displayed in the active Source Window 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 Quick Watch Dialog Box.
Add Watch...	Registers the specified data to the Watch window. Opens the Add Watch Dialog Box.
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 Debugger Option Dialog Box.
Change Watch...	Changes the data on the line selected by the Watch window. Opens the Change Watch Dialog Box. This menu is valid only when a variable is selected in the Watch Window.
Delete Watch	Deletes the selected watch point from the Watch Window.This menu is valid only when a variable is selected in the Watch Window.
Symbol...	Displays the address of the specified variable or function, or the value of the specified symbol. Opens the Symbol To Address Dialog Box.

(4) [Option] menu

Tool Bar	Selects whether the tool bar is displayed (default) or not.
Status Bar	Selects whether the tool 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 Source Window is active. It is performed at the instruction level if Assemble Window is active. If neither window is active, step execution is performed at the source level.
Configuration...	Sets the environment. Opens the Configuration Dialog Box.
Mask Option...	Performs the mask option settings and pin mode settings (when IECUBE is connected). Opens the Mask Option Dialog Box.
Extended Option...	Sets extended options. Opens the Extended Option Dialog Box.
RRM Setting...	Sets the sampling range for RRM function. Opens the RRM Dialog Box (when MINICUBE is connected).
Flash Option...	This dialog box is used to make the flash self programming emulation settings. (When IECUBEis connected.) Opens the Flash Option Dialog Box.
Debugger Option...	Sets ID78K0-QB options. Opens the Debugger Option Dialog Box.
Add I/O Port...	Adds user-defined I/O ports. Opens the Add I/O Port Dialog Box.
Trace Clear	Clears the trace data. This item is displayed only when Trace View Window is active (when IECUBE is connected).
Coverage Clear	Clears the coverage measurement results (when IECUBE is connected).

(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 breakpoints being set, and executes the program.  Same operation as this button.

Return Out	The user program is executed until execution returns  Same operation as this button. **Note:** 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	Executes the instructions in the program one by one (Next step execution). If a function or subroutine is called, its instructions are not executed on a step-by-step basis.  Same operation as this button.
Start From Here	Executes the program from the cursor position on the Source Window or Assemble Window.
Come Here	Executes the program from the current PC to the cursor position in the Source Window or the Assemble Window.
Go & Go	Continues executing the program. If a break occurs because a break condition is satisfied, the window is updated and the program is executed again.  Same operation as clicking this button each time a break has occurred.
Slowmotion	Continues step execution. Each time step execution has been performed, the window is updated and then step execution is performed again.  Same operation as clicking this button each time a break has occurred.
CPU Reset	Resets the CPU.  Same operation as this button.
Change PC	Sets the address at the cursor position in the Source Window or the Assemble Window to the PC.
Break Point	Sets or deletes a breakpoint at the cursor position in the Source Window or the Assemble Window.
Software Break Point	Sets or deletes a software breakpoint at the cursor position in the Source Window or the Assemble Window.
Delete All Breakpoints	Deletes all the set breakpoints (includes the software breakpoints).
Uncond. Trace ON	Validates unconditional trace so that trace can always be executed during program execution (default). (When IECUBE is connected.) At this time, the set trace event conditions are ignored.
Cond. Trace ON	Validates conditional trace and traces in accordance with the trace event condition during program execution (when IECUBE is connected).
Tracer control mode	Sets trace control mode (when IECUBE is connected).

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

Timer Start/Timer Stop	Starts timer measurement when it is stopped, or stops it when it is in progress (when IECUBE is connected). This item is invalid if the program is not being executed and if a timer event is not used. Immediately after program execution has been started, timer measurement is in progress.
Tracer Start/Tracer Stop	Starts the tracer when it is stopped, or stops it when it is in progress (when IECUBE is connected). This item is invalid if the program is not being executed. Immediately after program execution has been started , the tracer is executed.
Pseudo Emulation...	Opens the Pseudo Emulation Dialog Box (when IECUBE is connected).

(6) [Event] menu

Event Manager	Manages various event conditions. Opens the Event Manager.  Same operation as this button.
Software Break Manager	Manages software break event conditions. Opens the Software Break Manager.
Event...	Registers an event condition. Opens the Event Dialog Box.  Same operation as this button.
Event Link...	Registers an event link condition (when IECUBE is connected). Opens the Event Link Dialog Box.
Break...	Registers and sets a break event condition. Opens the Break Dialog Box.  Same operation as this button.
Trace...	Registers and sets a trace event condition (when IECUBE is connected). Opens the Trace Dialog Box.  Same operation as this button.
Snapshot...	Registers and sets a snapshot event condition (when IECUBE is connected). Opens the Snap Shot Dialog Box.
Stub...	Registers and sets a stub event condition (when IECUBE is connected). Opens the Stub Dialog Box.
Timer...	Registers and sets a timer event condition (when IECUBE is connected). Opens the Timer Dialog Box.  Same operation as this button.
Event DMM...	Registers and sets an event DMM condition (when IECUBE is connected). Opens the Event DMM Dialog Box.

Delay Count...	Sets the delay count (when IECUBE is connected). Opens the Delay Count Dialog Box.

(7) [Browse] menu

Source Text	Displays a source text. Opens the Source Window. If there is this window already open in the active status, it is opened in the static status.  Same operation as this button.
Assemble	Displays the disassemble results. Opens the Assemble Window. If there is this window already open in the active status, it is opened in the static status.  Same operation as this button.
Memory	Displays the contents of the memory. Opens the Memory Window. If there is this window already open in the active status, it is opened in the static status.  Same operation as this button.
Watch	Displays the watch contents. Opens the Watch Window.  Same operation as this button.
Register	Displays the register contents. Opens the Register Window.  Same operation as this button.
SFR	Displays the SFR. Opens the SFR Window.  Same operation as this button.
Local Variable	Displays the local variable. Opens the Local Variable Window.  Same operation as this button.
Stack Trace	Displays the stack trace results. Opens the Stack Window.  Same operation as this button.
Trace	Displays the trace results (when IECUBE is connected). Opens the Trace View Window.  Same operation as this button.
Code Coverage	Displays the code coverage measurement results (when is IECUBE is connected). Opens the Code Coverage Window.  Same operation as this button.
Console	Opens the Console Window.
Others	Displays other windows. (refer to "A.2 Sample List Of Expansion Window") Displays a user-defined window list.

(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 Source Window. If the Source Window in active 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 Assemble Window. If the Assemble Window in active 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 Memory Window. If the Memory Window in active is open, that window is displayed in the forefront (so that it can be manipulated).
Coverage	Cannot be selected.

(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 Source Window, Assemble Window or Memory Window.
Cascade	Cascade display of the windows in the Main window.
Tile	Cascade display of the windows in the Main window.
Arrange Icons	Rearranges the icons in the Main window.
Close All	Closes all windows, except the Main window.
Refresh	Updates the contents of the window with the latest data.
Active	Sets the window in the active status.
Static	Sets the window in the static status.
(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

ID78K0-QB Help	Displays the help.
Command Reference	Opens the Help window of COMMAND REFERENCE.
Main Window	Displays the help of the Main window.
Current Window	Displays the help of the current window.
About...	Displays the version of the ID78K0-QB. Opens the About Dialog Box.

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.

 **Stop**	Stops execution of the user program. Same function as [Run] menu -> [Stop].
 **ReGo**	Resets the CPU and executes the user program. Same function as [Run] menu - > [Restart].
 **Go**	Executes the user program from the current PC without resetting the CPU. Same function as [Run] menu -> [Go].
 **Go**	Ignores breakpoints being set, and executes the user program. Same function as [Run] menu -> [Ignore break points and Go].
 **Ret**	The user program is executed until execution returns Same function as [Run] menu -> [Return Out]. **Note:** This command is used for a function described in C language.
 **Step**	Step execution (executes instructions in the program one by one.) If a function or subroutine is called, its instructions are executed one by one. Same function as [Run] menu -> [Step In].
 **Over**	Next step execution (executes the program, assuming a function/call statement as one step.) If a function or subroutine is called, its instructions are not executed on a step-by-step basis. Same function as [Run] menu -> [Next Over].
 **Res**	Resets the CPU. Same function as [Run] menu -> [CPU Reset].
 **Open**	Opens the View File Load Dialog Box. Same function as [File] menu -> [Open...].
 **Load**	Opens the Download Dialog Box. Same function as [File] menu -> [Download...].
 **Proj**	Opens the Project File Load Dialog Box. Same function as [File] menu -> [Project] -> [Open...].
 **Src**	Displays the source text. Opens the Source Window. Same function as [Browse] menu -> [Source Text].
 **Asm**	Displays the disassemble results. Opens the Assemble Window. Same function as [Browse] menu -> [Assemble].
 **Mem**	Displays the contents of the memory. Opens the Memory Window. Same function as [Browse] menu -> [Memory].
 **Wch**	Displays the watch contents. Opens the Watch Window. Same function as [Browse] menu -> [Warch].
 **Reg**	Displays the register contents. Opens the Register Window. Same function as [Browse] menu -> [Register].
 **SFR**	Displays the contents of SFR. Opens the SFR Window. Same function as [Browse] menu -> [SFR].

Loc	Displays the local variable contents. Opens the Local Variable Window. Same function as [Browse] menu -> [Local Variable].
Stk	Displays the stack trace results. Opens the Stack Window. Same function as [Browse] menu -> [Stack Trace].
TrW	Displays the trace results. (when IECUBE is connected) Opens the Trace View Window. Same function as [Browse] menu -> [Trace].
Cov	Displays the code coverage measurement results. (when IECUBE is connected) Opens the Code Coverage Window. Same function as [Browse] menu -> [Code Coverage].
Mgr	Opens the Event Manager. Same function as [Event] menu -> [Event Manager...].
Evn	Registers and sets events. Opens the Event Dialog Box. Same function as [Event] menu -> [Event...].
Brk	Registers and sets break events. Opens the Break Dialog Box. Same function as [Event] menu -> [Break...].
Trc	Registers and sets a trace event (when IECUBE is connected). Opens the Trace Dialog Box. Same function as [Event] menu -> [Trace...].
Tim	Registers and sets a timer event (when IECUBE is connected). Opens the Timer Dialog Box. Same function as [Event] menu -> [Timer...].

(2) Operation of Toolbar

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

This toolbar can be displayed in the following two modes. The modes are selected in the Debugger Option Dialog Box.

Figure 6-2 Toolbar (Picture Only)

Figure 6-3 Toolbar (Picture and Text)

Window display area

This area displays various debug windows.

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

Status bar

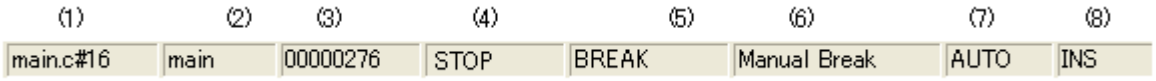
The status bar displays the status of the ID78K0-QB and in-circuit emulator.

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].

Remark: If the screen resolution is low (800 - 600, etc.), all the statuses may not be displayed on the 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) IE 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 Memory Window is fixed to OVR mode.

Table 6-2 CPU Status

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

Table 6-3 IE Status

Display	Meaning
RUN	User program execution in progress (the color of the status bar changes).
STEP	Step execution in progress.
TRC	Trace operating
TIM	Timer operating
BREAK	Break occurring.
Time	Displays the result of measuring the time from the start of user program execution to the occurrence of break (Run-Break time).**Note**
TIMER OVERFLOW	Measurement result overflowed.

Note: The measurable range is 20 ns (min., with undivided clock) to 48 hours 50 minutes (max., with clock divided by 2K). (When IECUBE is connected.)
The measurable range is 100 us (min.) to 119 hours 18 minutes (max.). (When MINICUBE2 is connected.)

Table 6-4 Break Cause

Display	Meaning
Manual Break	Forced break
Temporary Break	Temporary break
Software Break	Software break
Trace Full Break	Break due to trace full
Trace Delay Break	Break due to trace delay
Non Map Break	Non-mapped area is accessed.
Write Protect	An attempt has been made to write to a write-protected area.
Read Protect	Read was attempted from read protected area
SFR Illegal	SFR was illegally access
SFR Write Protect	Read from prohibited SFR was attempted
SFR Read Protect	Write to write-prohibited SFR was attempted.
Stack Overflow	Break due to stack overflow
Stack Underflow	Break due to stack underflow
Uninitialize Stack Pointer	Break due to failure to perform stack pointer initialization
Uninitialize Memory Read	Memory not initialized has been read.
Timer Over Break	Execution time-over detected
Fetch Break	Guarded area or fetch prohibited area was fetched

Display	Meaning
Unspecified Illegal	Illegal action of user program related to peripheral chip function occurred. Refer to documentation on peripheral emulation board.
IMS (IXS) Illegal	Break due to illegal write to IMS, IXS register
Retry Over	RETRY count over break
Flash Illegal	Flash illegal break
Peripheral Break	Break from peripheral
Break Event: "*Event name*"	Stopped due to event cause of displayed event name.
Event Break "*Event Name*"	Stopped due to event cause of displayed event name.
Before Execution "*Event Name*"	Break before execution due to event cause of displayed event name.

Configuration Dialog Box

This dialog box is used to display and set the ID78K0-QB operation environment. (Refer to "5.1 Setting Debugging Environment".)

This dialog box is automatically displayed after the ID78K0-QB is started up.

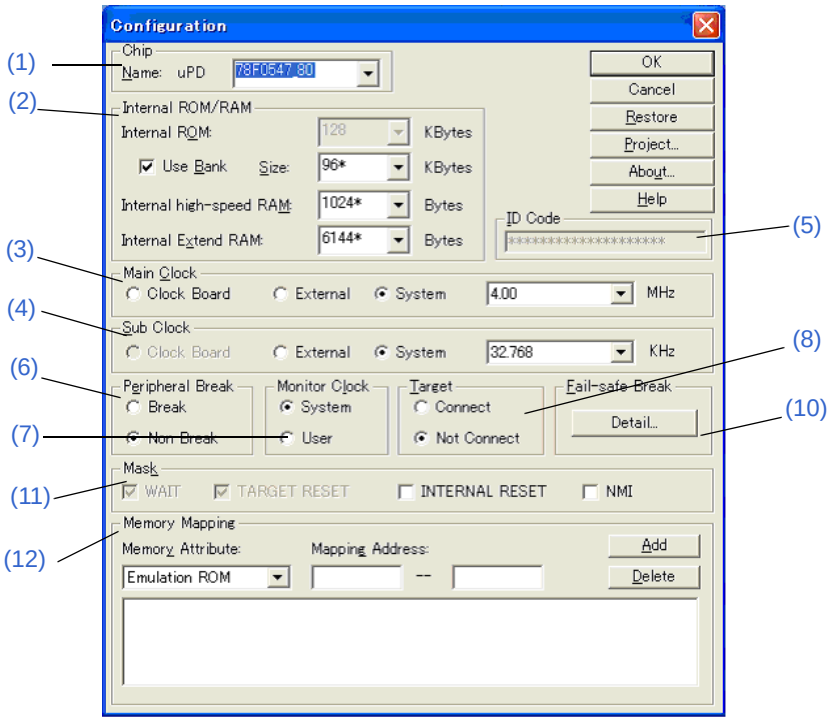
However, no setting is required to read a project as the results of reading the project file are reflected in this dialog box. (Refer to "5.17.1 Debugging environment (project file)".)

Caution: Setting to the "Chip" and "Target" areas can be made only when the Configuration Dialog Box opens automatically when the ID78K0-QB is started (if the debug environment is saved into a project file after setting these areas, the Configuration Dialog Box will no longer open when the ID78K0-QB is started).

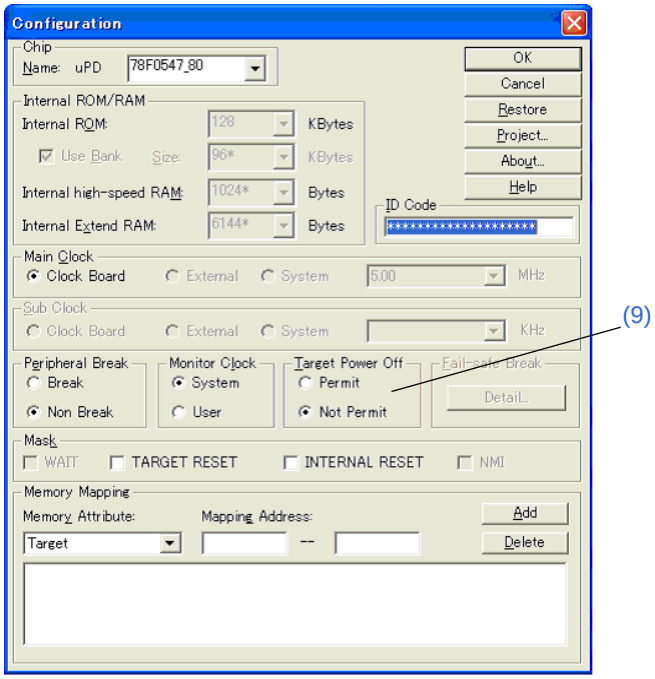
To change the settings of these areas, open the folder in which the project file is stored and delete the file with the extension ".pri" (file with ".pri" contains the debugger settings).

Figure 6-5 Configuration Dialog Box

(When IECUBE Is Connected)



(When MINICUBE Is Connected)



- Opening
- Explanation Of Each Area
- Function Buttons

Opening

(Automatically when the ID78K0-QB is started up)

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

Explanation Of Each Area

(1) Chip

This area is used to select the chip name.

A chip name is selected from the drop-down list.

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

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

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 RAM and internal ROM of the CPU. (Refer to "Table 5-2 Mapping Attribute".)

When MINICUBE is connected, this area is fixed to device file value.

A mapping size is selected from the drop-down list.

The default size is obtained from the device file through selection in (1) Chip , and displayed (value with '*').

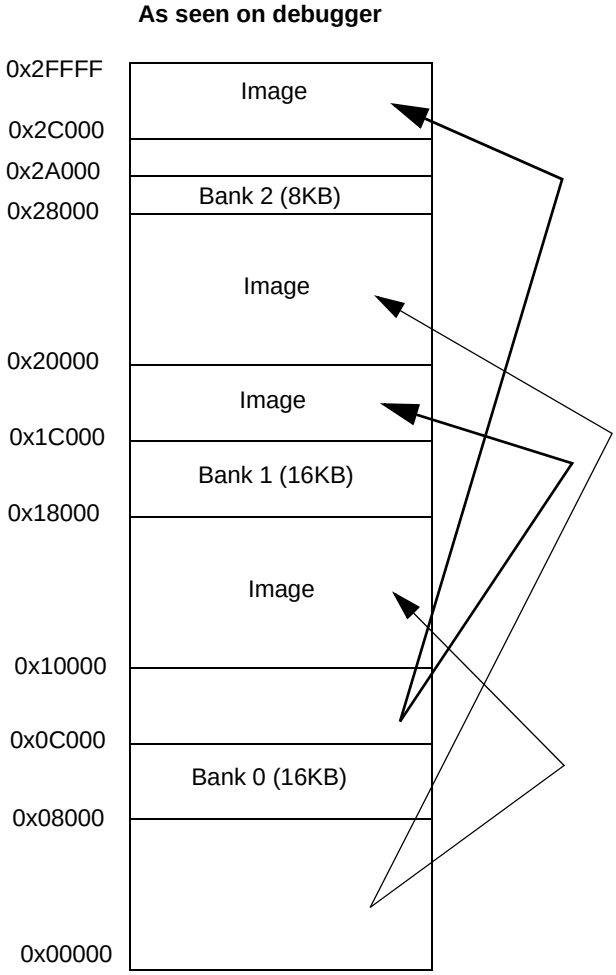
Table 6-5 Range and Unit of Internal ROM/RAM Setting (When IECUBE Is Connected)

Area	Meaning	Settable Range
Internal ROM**Note1**	Sets the Internal ROM size	When memory bank is not used: 0 to 60 KB When memory bank is used: 52 to 256 (gray) (in 4 KB units)
Internal Bank ROM**Note2**	Sets the Internal bank ROM size	When memory bank is not used: 0 (gray) When memory bank is used: 20 to 224 (gray) (in 4 KB units)
Internal High-speed RAM	Sets the Internal high-speed RAM size	64 to 1024 bytes (64-byte units)
Internal Extend RAM	Sets the Internal Extend RAM size	0 to 14336 bytes (512-byte units)

Note1: If "Use Bank" is selected, the internal Bank ROM area size can be changed.
At this time, the "Internal ROM" area cannot be selected and the total of the internal bank ROM area size plus 32 KB (Common area) is grayed.
If memory bank information is included in the device file, this item is selected by default. This item cannot be selected if there is no memory bank information.

Note2: When the internal bank ROM is used, the address space seen in the ID78K0-QB is as shown in Figure 6-6.
Consequently, the address range cannot be specified for address spaces of address 0x10000 or higher (except for bank areas).

Figure 6-6 Diagram of Address Space When Internal ROM Bank Is Used (With Bank ROM Size of 40 KB)



(3) Main Clock

Select either of the following as a main clock source.

Clock Board	Clock mounted in the emulator
External	Clock mounted on the target system (rectangular wave)
System	Clock generated in the emulator Select the relevant clock from the drop-down list. - 1.00, 2.00, 3.00, 3.57, 4.00 (default), 4.19, 4.91, 5.00, 6.00, 8.00, 8.38, 10.00, 12.00, 16.00, 20.00 (when IECUBE is connected) - 5.00 (default) (when MINICUBE is connected) - 4.00 (default),8.00, 16.00 (when MINICUBE2 is connected)

When IECUBE is connected, the default status and available options in this area vary as follows, depending on the detection result of the target power supply (TVDD) (O: Available, X: Results in an error when selected).

	TVDD ON	TVDD OFF
Clock Board	O	O
External	O (default)	X
System	O	O (default)

When MINICUBE is connected, this area is fixed to "Clock Board" when an oscillator is mounted in the clock socket in the emulator; otherwise, it is fixed to "System".

(4) Sub Clock

This area is used to select the subclock source input to the CPU.

Clock Board	Clock mounted in the clock socket on the emulator
External	Clock mounted on the target system (rectangular wave)
System	Clock generated in the emulator Select the relevant clock from the drop-down list. 32.762 (default), 38.400

When IECUBE is connected, the default status and available options in this area vary as follows, depending on whether a clock (oscillator or main clock board with a resonator) is mounted on the emulator and the detection result of the target power supply (TVDD) (O: Available, X: Results in an error when selected, --: Unavailable).

	Clock Mounted		No Clock Mounted	
	TVDD ON	TVDD OFF	TVDD ON	TVDD OFF
Clock Board	O (fixed)		--	--
External	--		O (default)	X
System	--		O	O (default)

When is MINICUBE is connected, cannot be selected.

(5) ID Code

This area is used to input the ID code to be used when the code on the internal ROM or internal flash memory is read by ID78K0-QB (ID code authentication) (When MINICUBE is connected)

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

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

The ID code is saved to the registry. If inputting the ID code fails three times, the ID78K0-QB is forcibly terminated.

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

(6) Peripheral Break

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

Break	The peripheral emulation function of in-circuit emulator is stopped during a break (when IECUBE is connected). During breaks, operations are stopped except for the following peripheral functions (when MINICUBE is connected). - 8-bit timer/event counter 50
Non Break	During breaks, these operations are maintained except for the following peripheral functions (default). - Watchdog timer - 8-bit timer H1 **Note:** 8-bit timer H1 stops operation only when the internal high-speed oscillator is selected for the operation clock, and continues operation if a clock other than the internal high-speed oscillator clock is selected.

(7) Monitor Clock

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

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

System	The operation clock is switched to the main clock and the monitor program is executed (default). **Note:** In theID78K0-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.

The operation clock during a break varies as follows, depending on the setting made in this area (when MINICUBE is connected) (User: When a break occurs, the monitor program operates with a clock before the break occurrence. Main: the monitor program operates with a clock supplied from MINICUBE.)

Table 6-6 Operation Clock During Break (when MINICUBE Is Connected)

Device Version[Note1]	Setting	When System Is Selected		When User Is Selected	
		Flash Memory Overwriting[Note2]	Other	Flash Memory Overwriting[Note2]	Other
V2.00 or later	Peripheral	User		User	
	CPU	Main		Main	User
V1.20 or later but earlier than V2.00	Peripheral	Main		Main	User
	CPU	Main		Main	User
V1.10 or earlier	Peripheral	Main		Main	
	CPU	Main		Main	

Note1: Vx.xx in "OCD Control Code Vx.xx", which is displayed by selecting [About...] from the [Help] menu in the ID78K0-QB, shows the device version.

Note2: Processing that involves flash memory overwriting operations such as downloading data and writing to memory

(8) Target

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

Remark: It is used to detect an illegal power supply status.
The default is determination by detecting the power (TVDD) of the target. (Refer to "Table 3-2 Error Message Output Pattern".)

Connect	Be connected
Not Connect	Not be connected

(9) Target Power Off

This area is used to select whether the Power Off emulation function is valid (when MINICUBE is connected).

Do not turn on/off the power to the target system during a break.

A reset from the target system is masked during a break.

Setting Of Target Power Off	Meaning	Function
Permit	Enables Power Off emulation function.	If the power from the target system is turned on while a user program is running, the user program is reexecuted immediately after a reset operation without generating a break. If a reset is input from the target system while a user program is running, the user program is reexecuted immediately after a reset operation without generating a break
Not Permit	Disables Power Off emulation function.	If the power from the target system is turned on while a user program is running, a break is generated after a reset operation. If a reset is input from the target system while a user program is running, a break is generated after a reset operation, and then the user program is forcibly reexecuted by the debugger.

(10) Fail-safe Break

Clicking the <Detail...> button opens the Fail-safe Break Dialog Box, so that the fail-safe break function can be individually set. (When IECUBE is connected.)

(11) 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 system is not stable at the debugging stage.

Caution1: When MINICUBE is connected, only "TARGET RESET" and "INTERNAL RESET" can be specified.

Caution2: When is IECUBE is connected, and when "Not Connect" is specified in (8) Target, "WAIT" and "INTERNAL RESET" are fixed to the selected state, and the mask status cannot be changed.

(12) Memory Mapping

This area is used to set mapping by specifying the memory attribute, and address.

(a) Memory Attribute

The following mapping attributes can be selected. Select a mapping attribute according to the usage. (Refer to "Table 5-2 Mapping Attribute".)

The mapping unit is 1 byte.

Emulation ROM**Note**	Selects the in-circuit emulator alternate ROM. (When IECUBE is connected) (Up to four items in the total of Emulation ROM/Target can be mapped)
Target **Note**	Selects the target memory. (Up to four items in the total of Emulation ROM/Target can be mapped)
I/O Protect**Note**	Selects the I/O protect area. (This area can be set only in the area set as Target or External SFR area.)
Stack	Selects the memory of the stack area (when IECUBE is connected)

Note: This item cannot be selected when using a device that does not have an external space.

Remark: If the Stack area is not specified, the entire Internal high-speed RAM area (except for the register area) is specified as the Stack area by default.

Caution: The area set as I/O Protect is not read unless it is registered to the SFR Window or Watch Window as an I/O port. To read this area, forcibly read it on these windows.

(b) Mapping Address

Specify the address to be mapped.

Input the higher and lower addresses from the keyboard.

(c) <Add>button, <Delete> button

These buttons are used to set and delete mapping.

When the <Add> button is clicked, the mapping specified with (a) Memory Attribute is set and displayed.

Function Buttons

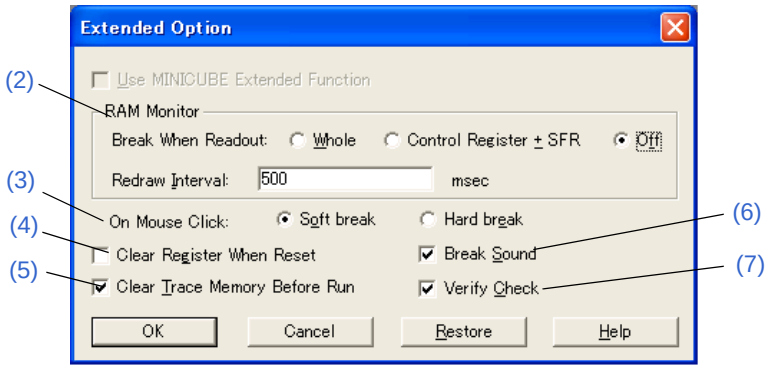
OK	Validates the current environment. Sets the environment and closes this dialog box.
Cancel	Cancels the changes and closes this dialog box.
Restore	Restores the previous settings before this dialog box was opened.
Project...	Opens the Project File Load Dialog Box. If an error occurs while a project file is being opened or read, the ID78K0-QB can no longer continue and is terminated.
About...	Opens the About Dialog Box.
Help	Displays the help window of this window.

Extended Option Dialog Box

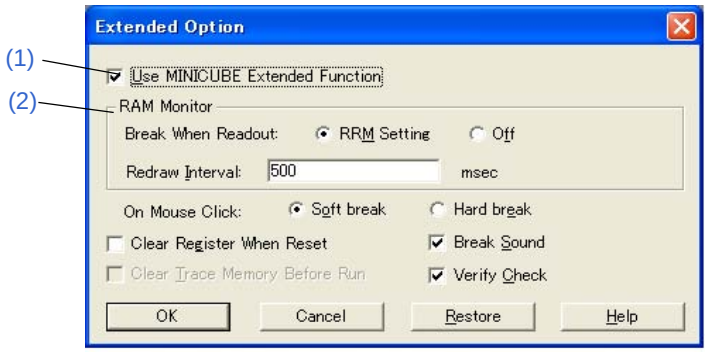
This dialog box is used to display and set the extended options of the ID78K0-QB. (Refer to "5.1 Setting Debugging Environment".)

Figure 6-7 Extended Option Dialog Box Extended Option Dialog Box

(When IECUBE Is Connected)



(When MINICUBE Is Connected)



- Opening
- Explanation Of Each Area
- Function Buttons

Opening

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

Explanation Of Each Area

(1) Use MINICUBE Extended Function

Check this item when using the MINICUBE extended functions (when MINICUBE is connected).

If this item is selected, the RRM and DMM functions are enabled (refer to "5.15.2 Pseudo real-time RAM monitor function (Break When Readout)" and "5.16 DMM Function").

If this item is selected, (2) RAM Monitor is enabled (this item is selected by default).

(2) RAM Monitor

(a) Break When Readout:

Select this item to specify the target range of RAM sampling by instantaneously generating a break in the user program execution (refer to "5.15.2 Pseudo real-time RAM monitor function (Break When Readout)").

Whole	Whole memory space. (when IECUBE is connected)[Note] **Remark:** The user program execution is stopped for a long time when a large number of windows are opened because the range from which memory is read out is wid.
Control Register + SFR	Control registers and SFR area (when IECUBE is connected)
RRM Setting	Specified area in the RRM Dialog Box.(when MINICUBE is connected)
Off	Disables the Pseudo real-time RAM monitor function (Break When Readout). **Remark:** Memory areas other than those shown in "Table 5-21 Real-Time RAM Monitor Function Sampling Range" are displayed as "**".

Note: Except for "Table 5-21 Real-Time RAM Monitor Function Sampling Range".

(b) Redraw Interval:

Specify the sampling time (ms) of the RAM sampling.

It 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.

(3) On Mouse Click

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

Soft break	Sets a software breakpoint (default). The mark of breakpoint is displayed in blue.
Hard break	Sets a hardware breakpoint. The mark of breakpoint is displayed in green (before execution).

(4) Clear Register When Reset

Select this check box to set the SP register to 0xFEE0 (0xFED0 in the case of MINICUBE) and all the banks of the general-purpose registers (X, A, C, B, E, D, L, H) to 0x0 after a CPU reset.

Under the default setting, the registers are not cleared.

(5) Clear Trace Memory Before Run

Select this checkbox to clear the trace memory prior to program execution (when IECUBE is connected).

(6) Break Sound

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

(7) 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 Memory Window and is written to memory.

Caution: When MINICUBE is connected, during write to the internal flash memory (including download), verify check is not performed whether or not the checkbox in this area is selected, and internal verify of flash self-write is always performed (read verify is not performed).

Function Buttons

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

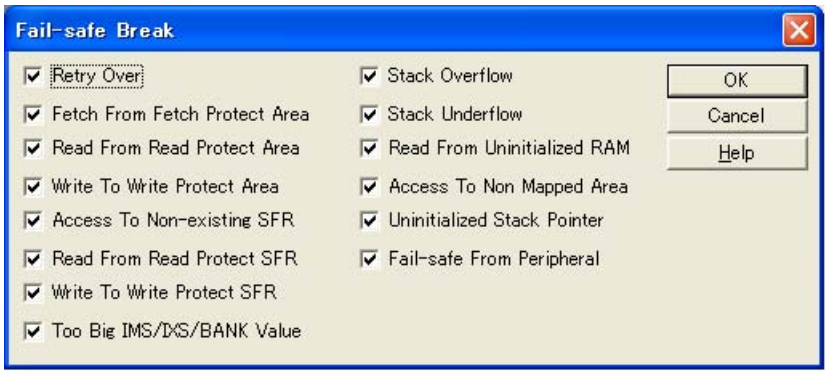
Fail-safe Break Dialog Box

IECUBE

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

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

Figure 6-8 Fail-safe Break Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons

Opening

Click the <Detail...> button in the Configuration Dialog Box.

Explanation Of Each Area

(1) Fail-safe break setting area

When a checkbox is selected, the corresponding fail-safe break function is enabled. Under the default setting, all the checkboxes are selected.

When MINICUBE is connected, only "Uninitialized Stack Pointer" is always abled.

Retry Over	Maximum allowable number of retries from peripheral exceeded
Fetch From Fetch Protect Area	Fetch of fetch prohibited area
Read From Read Protect Area	Read from read prohibited area
Write To Write Protect Area	Write to write prohibited area
Access To None-existing SFR	Access to non-existent SFR
Read From Read Protect SFR	Read of read prohibited SFR
Write To Write Protect SFR	Write to write prohibited SFR
Too Big IMS/IXS/BANK Value	IMS (memory size switching register), IXS (internal expansion RAM size switching register), BANK setting error
Stack Overflow	User stack limit exceeded (upper limit)
Stack Underflow	User stack limit not reached (lower limit)
Read From Uninitialized RAM	Failure to perform RAM initialization
Access To Non Mapped Area	Access to non-mapped area
Uninitialized Stack Pointer	Failure to perform stack pointer initialization
Fail-safe From Peripheral	Fail-safe from peripheral

Function Buttons

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

RRM Dialog Box

MINICUBE

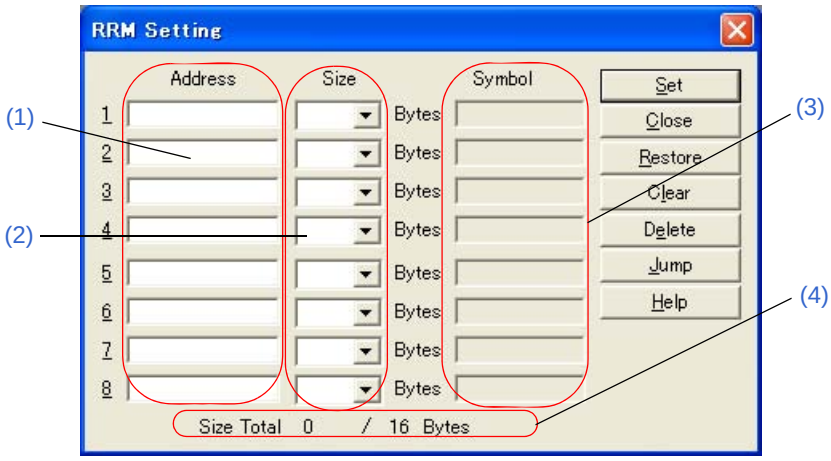
This dialog box is used to set the sampling range for the RRM function. (Refer to "5.15 RRM Function".

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.

Caution: This dialog box can be opened only when (1) Use MINICUBE Extended Function is selected in the Extended Option Dialog Box.

Figure 6-9 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" 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.

Remark: If the total size of the eight locations specified in the Address field already exceeds 16 bytes, or setting for 8 locations has already been made, the RRM Setting Dialog Box can be opened but the values cannot 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" 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.

Remark: If the total size of the eight locations specified in the Address field already exceeds 16 bytes, or setting for 8 locations has already been made, the RRM Setting Dialog Box can be opened but the values cannot be set.

Explanation Of Each Area

(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 5-6 Specifying Symbols".)

Following input, click the <OK> 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 - 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	Determine the specified sampling range.
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 Memory Window and displays the addresses in (1) Address whose numbers have a focus. Jump is performed for Memory Window that are in the active status. If multiple memory windows are to be opened, they must be set in the static status. (Refer to "5.18.1 Active status and static status")
Help	Displays this dialog box online help files.

Mask Option Dialog Box

IECUBE

This dialog box is used to perform mask option and pin mode settings (refer to "5.1 Setting Debugging Environment").

However, it is not necessary to perform such settings if reading a project file, and the results of reading a project file are reflected in this dialog box (refer to "5.17.1 Debugging environment (project file)").

Remark: For details about the mask options and pin modes, refer to the user's manuals for the in-circuit emulator and emulation board that are used.

Figure 6-10 Mask Option dialog box

- Opening
- Explanation of each area
- Function Buttons

Opening

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

Explanation of each area

(1) Pin group name:

This area is used to select the pin group for which a mask option and pin mode are to be set.

(2) Option name:

This area is used to set mask options and pin modes.

The setting items differ depending on the pin group selected in "(1) Pin group name:".

The first item displayed as a setting item is the current setting.

Function Buttons

OK	Validates the settings and closes this dialog box.
Cancel	Closes this dialog box.
Set	Validates the settings. Consecutive settings can be made (dialog box does not close).
Restore	Restores the previous settings before this dialog box was opened.
Help	Displays this dialog box online help files.

Flash Option Dialog Box

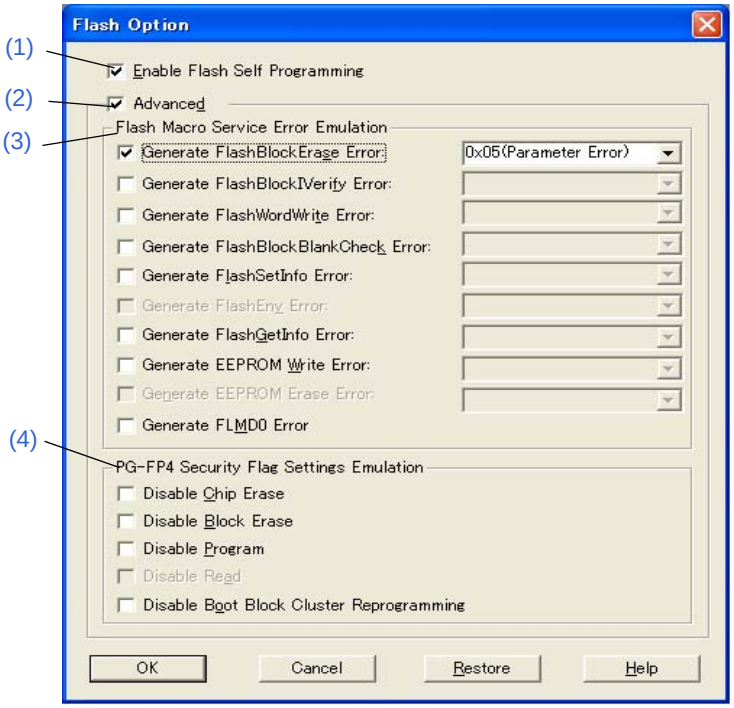
IECUBE

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

This dialog box cannot be opened during user program execution.

Caution: This dialog box cannot be opened when using a device that does not incorporate the flash memory.

Figure 6-11 Flash Option Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons
- Cautions

Opening

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

Explanation Of Each Area

(1) Enable Flash Self Programming

If this item is selected, flash self emulation is enabled.

This enables setting of (2) Advanced. This item is not selected by default.

(2) Advanced

If this item is selected, detailed settings for flash self emulation are enabled.

This enables setting of (3) Flash Macro Service Error Emulation and (4) PG-FP4 Security Flag Settings Emulation. This item is not selected by default.

(3) Flash Macro Service Error Emulation

These items set the operation of the self library function. By checking the item, the error value for the error set below can be forcibly returned (it is not returned in the normal emulation).

The combo boxes become valid if the corresponding check box is selected. This item is not selected by default.

Direct inputting is possible in the combo boxes, but the prescribed error values shown below are also selectable. The prescribed error values for each flash macro type vary as shown below.

	CZ6 series (Kx1+)	MF2 series (Kx2)
Generate FlashBlockErase Error	0x05 (Parameter Error) 0x1A (Erase Error)	0x05 (Parameter Error) 0x10 (Protect Error) 0x1A (Erase Error) 0x1F (Stop by Interrupt)
Generate FlashBlockIVerify Error	0x05 (Parameter Error) 0x1B (Inside Verify Error)	0x05 (Parameter Error) 0x1B (Inside Verify Error) 0x1F (Stop by Interrupt)
Generate FlashWordWrite Error	0x05 (Parameter Error) 0x18 (FLMD Error) 0x1C (Write Error)	0x05 (Parameter Error) 0x10 (Protect Error) 0x1C (Write Error) 0x1F (Stop by Interrupt)
Generate FlashBlockBlankCheck Error	0x05 (Parameter Error) 0x1B (Blank Check Error)	0x05 (Parameter Error) 0x1B (Blank Check Error) 0x1F (Stop by Interrupt)
Generate FlashSetInfo Error	0x05 (Parameter Error) 0x18 (FLMD Error) 0x1B (Inside Verify Error) 0x1C (Write Error)	0x05 (Parameter Error) 0x10 (Protect Error) 0x1A (Erase Error) 0x1B (Inside Verify Error) 0x1C (Write Error) 0x1F (Stop by Interrupt)
Generate FlashEnd Error	0x05 (Parameter Error)	--
Generate FlashGetInfo Error	0x05 (Parameter Error)	0x05 (Parameter Error) 0x20 (Read Error)
Generate EEPROM Write Error	0x05 (Parameter Error) 0x18 (FLMD Error) 0x1C (Write Error) 0x1D (MRG12 Error) 0x1E (Blank Error)	0x05 (Parameter Error) 0x10 (Protect Error) 0x1C (Write Error) 0x1D (MRG12 Error) 0x1E (Blank Error) 0x1F (Stop by Interrupt)

	CZ6 series (Kx1+)	MF2 series (Kx2)
Generate EEPROM Erase Error	0x05 (Parameter Error) 0x1A (Erase Error)	--
Generate FLMD0 Error	When the target is connected:Unavailable (the operation varies depending on the FLMD0 pin status)	
	When the target is not connected: If selected, FLMD0 operates at Low level. If cleared, FLMD0 operates at High level.	

(4) PG-FP4 Security Flag Settings Emulation

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

The settable items are as follows: All items are not selected by default.

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

Note: This item is valid only for MF2 firmware.

Function Buttons

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

Cautions

(1) Flash self emulation in the following case.

- The internal ROM size is not set to the default size

  Workaround: Set the default size to the internal ROM size.

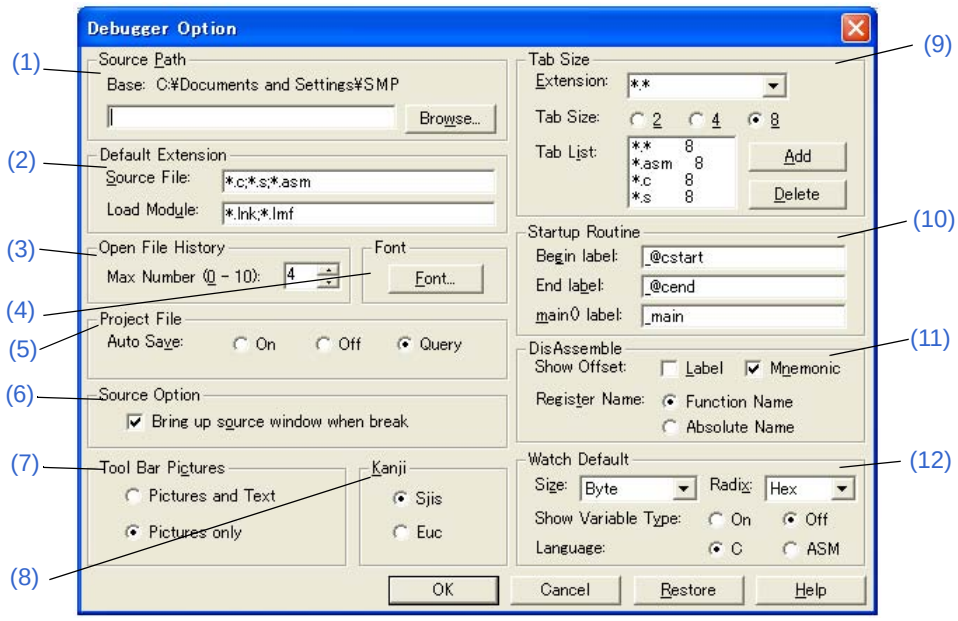
(2) The following restriction applies if flash self emulation is enabled.

The internal ROM size cannot be changed.

Debugger Option Dialog Box

This dialog box is used to display and set the various options of the ID78K0-QB.

Figure 6-12 Debugger Option Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons

Opening

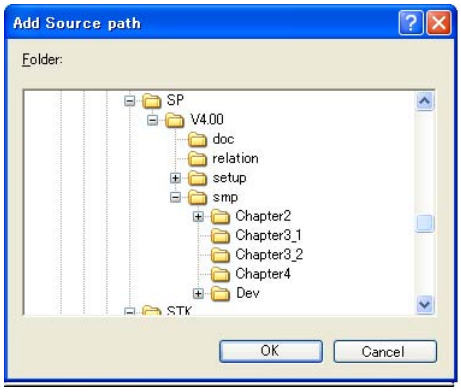
Select [Option] menu -> [Debugger Option...].

Explanation Of Each Area

(1) Source Path

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

Base:	The folder is the basis of a relative path is displayed. The base folder is determined in the following sequence: **1)** Folder to which the project file has been loaded **2)** Folder to which a load module or hex file has been loaded last **3)** Current folder of Windows

Text box	This area is used to specify the folder searched. To specify a folder, either directly input one to the text box, or click the <Browse...> button. A relative path can also be specified. Opens the [Add Source path] Dialog Box by clicking the <Browse...> button. To delimit paths, use ";" (semicolon) or "," (comma). Figure 6-13 [Add Source path] Dialog Box 

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

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

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

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

(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 Browse Dialog Box 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 Download Dialog Box is opened. The default extension is "*.lnk, *.lmf".

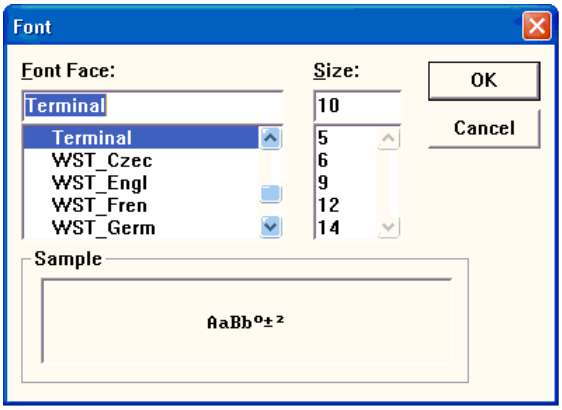
(3) Open File History

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

(4) Font

This area is used to specify the font displayed on the Source Window, Watch Window, Quick Watch Dialog Box, Local Variable Window, and Stack Window. Clicking the <Font...> button opens the [Font] Dialog Box in which the font to be displayed and its size can be set.

Figure 6-14 [Font] Dialog Box



(5) Project File

This area is used to set automatic saving of the project file. (Refer to "5.17.1 Debugging environment (project file)".)

Auto Save:	Sets whether the project file is automatically saved at the ID78K0-QB termination.	
	On	Automatically saves the project file.
	Off	Does not Automatically saves the project file.
	Query	Displays the Exit Debugger Dialog Box at the ID78K0-QB termination. (default)

(6) Source Option

This area is used to set the Source Window operation at a break.

By selecting this check box, a Source Window that is active at a break is displayed at the front.

If there is no active Source Window or no debug information in the load module file, the active Assemble Window is displayed at the front.

(7) Tool Bar Pictures

This area sets the buttons to be displayed on the toolbar. (Refer to "(2) Operation of Toolbar".)

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

(8) Kanji

Cannot be selected in this area.

(9) Tab Size

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

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

(10) Startup Routine

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

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

Begin label:	Specifies the symbol of the first address (default: _@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.

Show Offset:	Specifies whether an offset (symbol + offset) is displayed during disassemble display. When the offset is not displayed, only a symbol that matches a numeric value is displayed, if any. If no matching symbol is found, the numeric value is displayed as a hexadecimal number unchanged.	
	Label	Specifies whether the offset is displayed in the Label field. In the default condition, the offset is not displayed.
	Mnemonic	Specifies whether the offset is displayed in the Mnemonic field. In the default condition, the offset is displayed.

Register Name:	This area is used to select the method of displaying register names in mnemonics during disassemble display.	
	Function Name	Displays register names as function names or nicknames. (default)
	Absolute Name	Displays register names as absolute names.

(12) Watch Default

This area is used to specify a symbol to be watched in the Watch Window etc..

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

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.

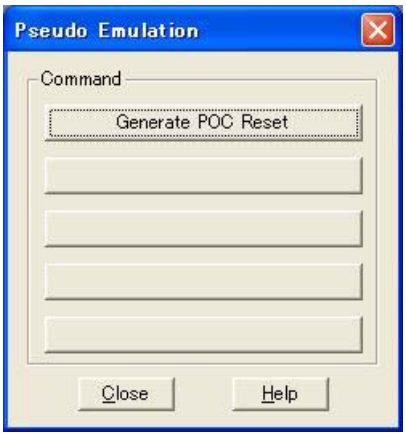
Pseudo Emulation Dialog Box

IECUBE

This dialog box is used for generation of pseudo emulation.

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

Figure 6-15 Pseudo Emulation Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons

Opening

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

Explanation Of Each Area

(1) Command

The targeted pseudo emulation can be generated by clicking the button with the name of pseudo emulation to be generated.The number of buttons and types depend on the device file.

Button with no pseudo emulation name cannot be clicked.

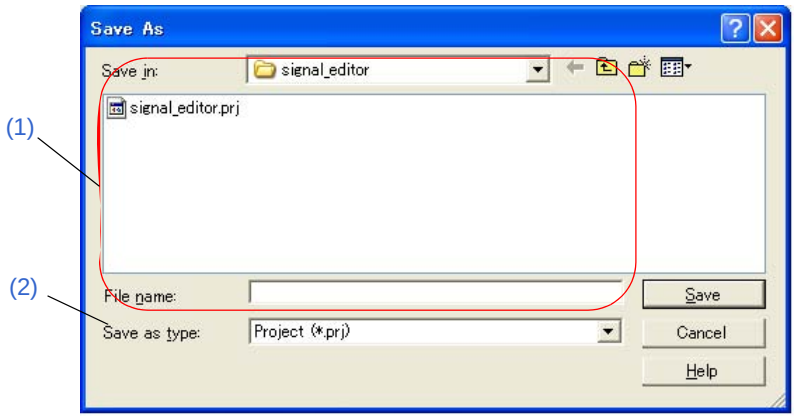
Function Buttons

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

Project File Save Dialog Box

This dialog box is used to save the current debugging environment to a project file. (Refer to "5.17.1 Debugging environment (project file)".) Project files can be newly saved or saved under an existing file name in this dialog box.

Figure 6-16 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:

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.

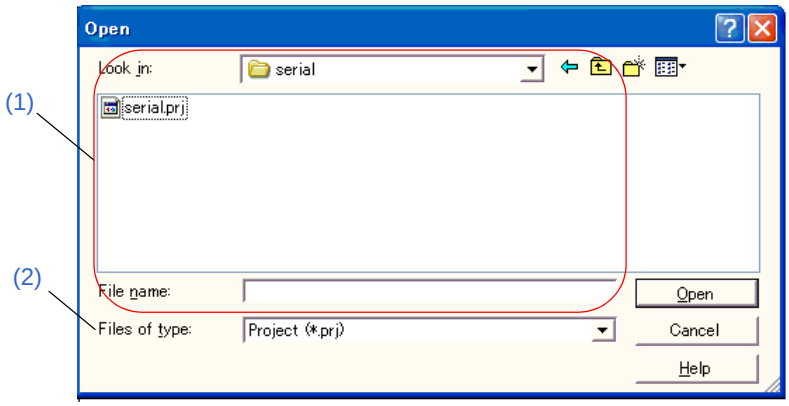
Project File Load Dialog Box

This dialog box is used to restore the debugging environment to the debugging environment saved to the project file. (Refer to "5.17.1 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 ID78K0-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 6-17 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:

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

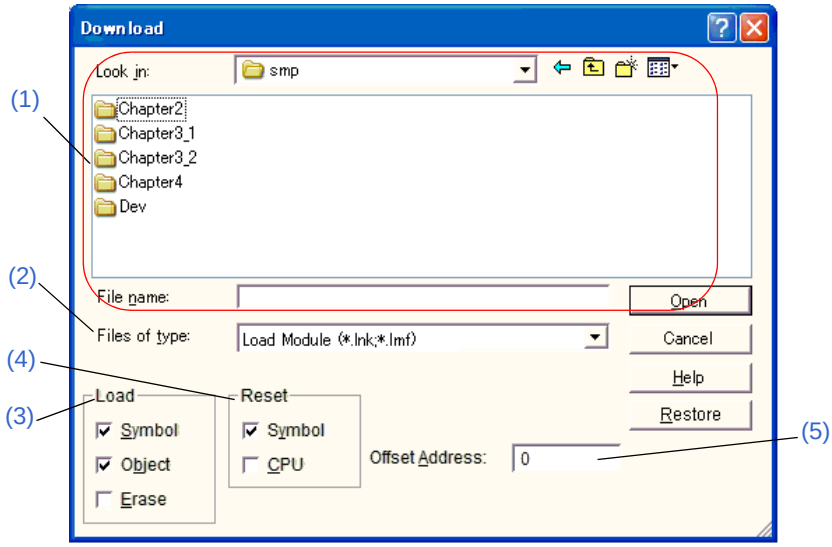
This dialog box is used to select the name and format of a file to be downloaded, and downloads memory contents to the in-circuit emulator and the target system. (Refer to "5.2 Download Function, Upload Function".)

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.

Remark: Refer to "(2) Allocation image when memory banks are used" for the file allocation image when a Hex format or binary data format has been downloaded when memory banks are used.

Figure 6-18 Download Dialog Box



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

- 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:

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 5-3 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.

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

(4) Reset

This area is used to set a reset condition.

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

Symbol	Specifies whether symbol information is reset or not.
CPU	Specifies whether the CPU is reset or not. (Selected, 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 or code coverage result file is specified. An address can be also specified by a symbol or expression. (Refer to "Table 5-6 Specifying Symbols".) The default radix for inputting a numeric value is hexadecimal.

Function Buttons

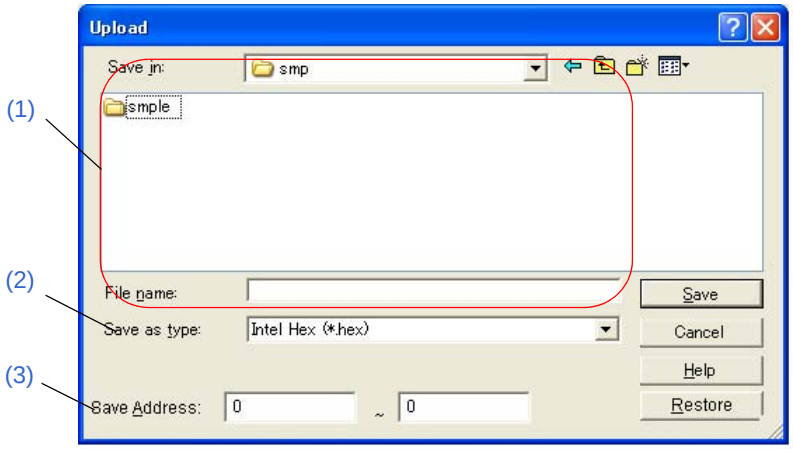
Open	Loads the selected file. After loading the file, this dialog box is closed.
Cancel	Closes this dialog box without loading the file.
Help	Displays this dialog box online help files.
Restore	Restores the input data to the original status.

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".)

Caution: Refer to "(2) Storage image when memory banks are used" for the file storage image when a Hex format or binary data format has been selected in (2) Save as type: when memory banks are used (when IECUBE is connected).

Figure 6-19 Upload Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons

Opening

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

Explanation Of Each Area

(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 5-4 Type of File That Can Be Uploaded")

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.

All the ranges are saved (this area cannot be set) when coverage data (*.cvb) is selected in (2) Save as type:.

An address can be also specified by a symbol or expression. (Refer to "Table 5-6 Specifying Symbols".)

The default radix for inputting a numeric value is hexadecimal.

Caution: If the file is saved with an end address of 0x10000 or higher specified, specify "Hex Format [Bank] (*.hex;*.hxb;*.hxf)" or "Binary Data [Bank] (*.bin)" for downloading.
If the file is saved with an address lower than 0x10000 specified, specify "Hex Format [64KB] (*.hex;*.hxb;*.hxf)" or "Binary Data [64KB] (*.bin)" for downloading (refer to "(2) Storage image when memory banks are used").

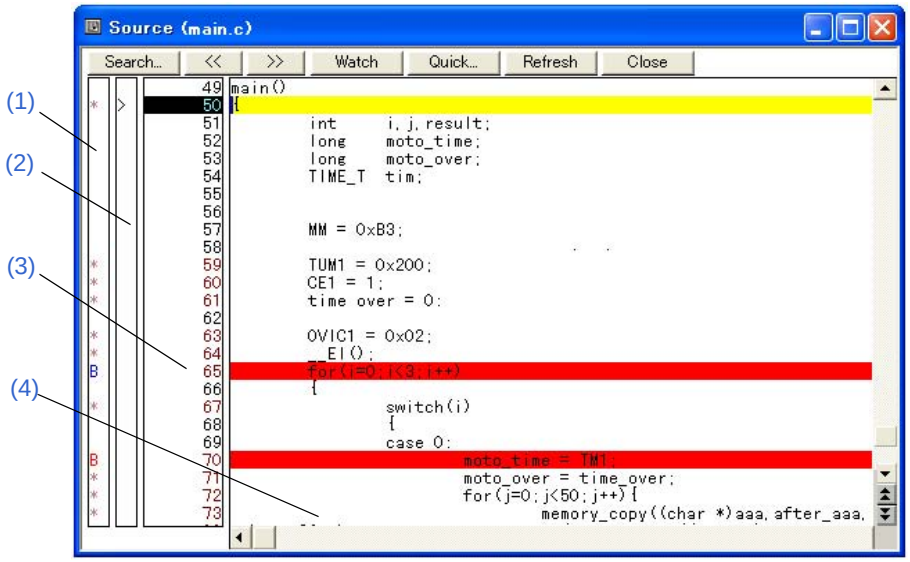
Function Buttons

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 , Display of locations for which coverage measurement is executed and Mixed display mode (Source Window), a number of other operations using Context Menu, Function Buttons, etc., can be performed in this window. Moreover, there are two statuses, Active status and static status, for this window. When the window is in the active status, it has the Trace Result with Linking Window (when IECUBE is connected). Moreover, the items selected in the window with Drag & drop function can be used in another window (refer to "5.18 Functions Common To Each Window").

Figure 6-20 Source Window



- Opening
- Explanation Of Each Area
- [View] menu (Source Window-dedicated items)
- Context Menu
- Function Buttons
- Cautions

Opening

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

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

Explanation Of Each Area

(1) Point mark area

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

Remark: The program code is displayed only when the symbol information downloaded by the load module file 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, whichever 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 6-7 Event Setting Status (Event Mark)

Mark	Meaning
B (blue)	Software breakpoint is set.
B (red)	Valid hardware breakpoint (after execution) is set.
B (green)	Valid hardware breakpoint (before execution) is set. **Note:** Breaks before execution are set with priority.
B (black)	Invalid hardware breakpoint is set. This hardware breakpoint can be validated on the Event Manager or in the Break Dialog Box.
E	Event condition is set.
L	Event link condition is set.(IECUBE)
T	Trace event is set.(IECUBE)
Ti	Timer event is set. (IECUBE)
S	Snapshot event is set. (IECUBE)
U	Stub event is set. (IECUBE)
M	DMM event is set. (IECUBE)
A	Multiple events are set.

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

(2) Current PC mark area

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

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

By double-clicking the current PC mark area, the program can be executed up to a specified line. (Refer to "

[Come Here]".)

(3) Line number/address display area

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

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

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

(4) Source text display area

This area displays 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 5-9 Type of Execution")
- Drag & drop function
- Context Menu

Caution: If a Program code does not exist on the source line, the top address of the line above or below the line on which a program code exists is manipulated by these functions.
These functions cannot be performed in the following cases. The corresponding menu will be dimmed and cannot be selected.
- If a file other than a source file is displayed
- While the user program is being executed

[View] menu (Source Window-dedicated items)

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

Create Break Event		Sets a break event that occurs if the selected variable is accessed.
	Break when Access to this Variable	Sets a break event that occurs if the selected variable is accessed for read/ write.
	Break when Write to this Variable	Sets a break event that occurs if the selected variable is accessed for write.
	Break when Read from this Variable	Sets a break event that occurs if the selected variable is accessed for read.
	Clear	Deletes a break event corresponding to the selected variable.
Event Information		Displays the event information of a line at the cursor position or a selected variable name. If an event is set, the Event Dialog Box is opened.
Mix		Turns on/off Mixed display mode (Source Window).

Execution Monitoring	Sets about code coverage display (when IECUBE is connected).
Clear	Clears highlights in code coverage display.
Accumulative	Cumulatively displays highlights in code coverage display.

Context Menu

Move...	Moves the display position. Opens the Source Text Move Dialog Box.
Mix	Turns on/off Mixed display mode (Source Window).
Add Watch...	Adds the specified data to the Watch Window. Opens the Add Watch Dialog Box.
Symbol...	Displays the address of the specified variable or function, or the value of the specified symbol. Opens the Symbol To Address Dialog Box.
Break when Access to this Variable	Sets a break event that occurs if the selected variable is accessed for read/write.
Break when Write to this Variable	Sets a break event that occurs if the selected variable is accessed for write.
Break when Read from this Variable	Sets a break event that occurs if the selected variable is accessed for read.
Clear	Deletes a break event corresponding to the selected variable.
Event Information	Displays the event information of a line at the cursor position or a selected variable name. If an event is set, the Event Dialog Box is opened.
Come Here	Executes the program from the current PC to the cursor position. (Refer to "Table 5-7 Break Types".)
Change PC	Sets the address at the cursor position to the PC.
Break Point	Sets or deletes a hardware breakpoint at the cursor position. **Remark:** Breaks before execution (B) are set with priority.
Software Break Point	Sets or deletes a software breakpoint at the cursor position.
Clear Execution Monitoring	Clears highlights in code coverage display (when IECUBE is connected).
Accumulative	Cumulatively displays highlights in code coverage display (when IECUBE is connected).
Assemble	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position. (Refer to "5.18.2 Jump function".) Opens the Assemble Window. If an active Assemble Window 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 "5.18.2 Jump function".) Opens the Memory Window. If an active Memory Window is open, that window is displayed in the forefront (so that it can be manipulated).

Function Buttons

Search...	Opens the Source Search Dialog Box 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. If no character string is selected, the Source Search Dialog Box is opened with nothing specified to be searched. Specify a search method in the Source Search Dialog Box. The results of search is highlighted in the Source window. This is the same operation as selecting [View] menu -> [Search...].
<<	Searches forward (upward on screen) for the text that satisfies the search condition set in the Source Search Dialog Box, 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 text that satisfies the search condition set in the Source Search Dialog Box, 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 variables selected in the source text display area to the Watch Window. If the Watch Window is not opened, it is opened. If no text is selected in the source text display area, the Watch Window is only opened. This is the same operation as selecting [View] menu -> [View Watch].
Quick...	Temporarily displays the contents, such as a variable, selected in the source text display area in the Quick Watch Dialog Box. If no text is selected in the source text 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.

Cautions

(1) 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.

(2) 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, ID78K0-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 ID78K0-QB is terminated.

(3) 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.

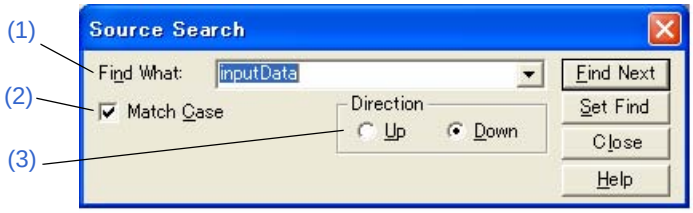
(4) If step-wise execution is performed in the source mode, the beginning of the function (prologue processing) is not skipped. Displaying the local variables or stack trace, and return execution cannot be performed in prologue processing. To skip prologue processing, perform the step-wise execution again.

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 6-21 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

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 selected 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).

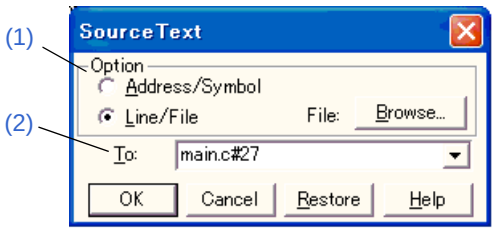
Function Buttons

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 6-22 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

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 "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 5-6 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 "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 bythe 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

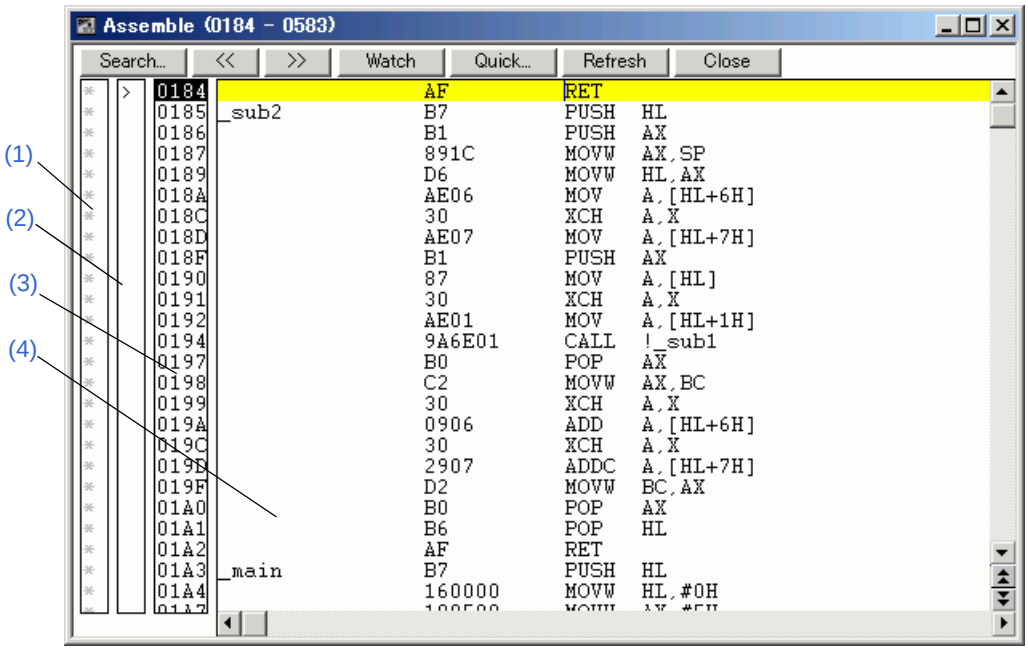
OK	Starts displaying the source text from the specified position.
Cancel	Closes this dialog box.
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

Assemble Window

This window is used to disassemble and display programs. It is also used to execute Line assembly. (Refer to "5.3 Source Display, Disassemble Display Function".) The results of line assembly are also reflected in the Memory Window.

In addition to Breakpoint setting and Display of locations for which coverage measurement is executed, a number of other operations using Context Menu, Function Buttons, etc., can be performed in this window. Moreover, there are two statuses, Active status and static status, for this window. When the window is in the active status, it has the Trace Result with Linking Window (when IECUBE is connected). Moreover, the items selected in the window with the Drag & drop function can be used in another window. (Refer to "5.18 Functions Common To Each Window".)

Figure 6-23 Assemble Window



- Opening
- Explanation Of Each Area
- [View] menu (Assemble Window-dedicated items)
- Context Menu
- Function Buttons
- Related Operations
- Cautions

Opening

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

Explanation Of Each Area

(1) Point mark area

This area is used for Event Setting Status (Event Mark) and Breakpoint setting.

(2) Current PC mark area

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

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

(3) Address specification area

This area displays the disassembly start address.

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

Caution: The end address is not display. The end address is 0xFFFF, if the memory bank function is not used.

(4) Disassemble display area

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

Yellow indicates the current PC line, and red indicates lines where a valid breakpoint is set.

It can be Line assembly in the mnemonic field.

This area also provides the following functions:

- [Come Here], [Start From Here] (Refer to "Table 5-9 Type of Execution")
- Drag & drop function
- Context Menu

[View] menu (Assemble Window-dedicated items)

The following items are added in the [View] menu, when the Assemble Window is active.

Event Information	Displays the event information of the address at the cursor position. If an event is set, the Event Dialog Box is opened. (when IECUBE is connected.)
Execution Monitoring	Sets about code coverage display (when IECUBE is connected).
Clear	Clears highlights in code coverage display.
Accumulative	Cumulatively displays highlights in code coverage display.

Context Menu

The menu items are effective for the selected line or item, not the position where the mouse pointer was clicked (same operation as when selecting the main menu with the same name).

Move...	Moves the display position. Opens the Address Move Dialog Box.
Add Watch...	Adds the specified data to the Watch Window. Opens the Add Watch Dialog Box.
Symbol...	Displays the address of the specified variable or function, or the value of the specified symbol. Opens the Symbol To Address Dialog Box.
Come Here	Executes the program from the current PC to the cursor position. (Refer to "Table 5-7 Break Types".)
Change PC	Sets the address at the cursor position to the PC.
Break Point	Sets or deletes a hardware breakpoint at the cursor position. **Remark:** Breaks before execution (B) are set with priority.
Software Break Point	Sets or deletes a software breakpoint at the cursor position.
Clear Execution Monitoring	Clears highlights in code coverage display (when IECUBE is connected).
Accumulative	Cumulatively displays highlights in code coverage display (when IECUBE is connected).
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 "5.18.2 Jump function".) If no line information exists at the jump destination address, however, you cannot jump. Opens the Source Window. If an active Source Window 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 "5.18.2 Jump function".) Opens the Memory Window. If an active Memory Window is open, that window is displayed in the forefront (so that it can be manipulated).

Function Buttons

Search...	Opens the Assemble Search Dialog Box and searches for a character string of mnemonics. Specify a search method in the Assemble Search Dialog Box. 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 Assemble Search Dialog Box, 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 Assemble Search Dialog Box, 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) Disassemble display area to the Watch Window. If the Watch Window is not opened, it is opened. If no text is selected in (4) Disassemble display area, 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) Disassemble display area on the Quick Watch Dialog Box. 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) Line 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 over-written. In this case also, as many 'nop' instructions as necessary are inserted. The same applies to instructions that straddle over source lines.

Cautions

(1) The symbol name of a memory bank area (0x8000 to 0xbfff), indicated by the mnemonic of the instruction that is assigned to the Common area, is the symbol name of the memory bank indicated by the BANK register upon display.

(2) Return execution ([Run] menu -> [Return Out]) cannot be performed when the program pointer (PC) points to an address other than the address of the source line. In addition, return execution cannot be performed successively because the PC does not point to the address of the source line after return execution. To perform return execution again, therefore, perform step-wise execution in the source mode so that the PC points to the address of the source line.

(3) If an attempt is made to scroll up the Assemble window toward the upper direction from the address at which the disassemble display starts, the display view may become invalid (this is because the debugger cannot locate the disassemble display start address).

In such a case, click the <Refresh> button; the address displayed at the top at that time is treated as the start address, and then the disassemble display is refreshed.

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 6-24 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:

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 selected to distinguish between uppercase and lowercase.

(3) Scan Whole Region

This should be selected 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 5-6 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.

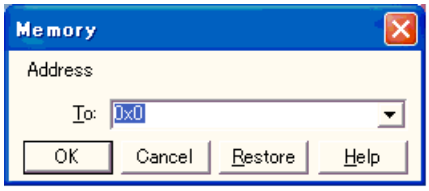
Address Move Dialog Box

This dialog box is used to specify the start address from which displaying, as follows.

- Memory Window
- Assemble Window
- SFR Window

Caution: When a memory bank is used, the specification of addresses for the address space of 0x10000 or higher cannot be made for other than bank areas. (Refer to "Figure 6-6 Diagram of Address Space When Internal ROM Bank Is Used (With Bank ROM Size of 40 KB)".)

Figure 6-25 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 5-6 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 "5.3.4 Convert symbol (symbol to address)".)

Figure 6-26 Symbol To Address Dialog Box

- Opening
- Explanation Of Each Area
- Function Buttons

Opening

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

Explanation Of Each Area

(1) Symbol:

This area is used to specify the variable, function name, symbol name, or line number to be converted. (Refer to "Table 5-6 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

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.

If bit symbol have been specified, they are converted to the Address.bit format. Also, equations that include bit symbols cannot be specified.

(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 (1) Symbol: have been changed, converts the symbol. After conversion, closes the dialog box if the contents of (1) Symbol: have not been changed.
Close	Closes this dialog box.
Restore	Restores the input data to the original status. If the <OK> button has already been clicked, the data is restored to the status immediately after the <OK> button was clicked.
Help	Displays this dialog box online help files.

Watch Window

This window is used to display and change specified watch data. (Refer to "5.6 Watch Function".)

This window can also display wide-ranging watch data (such as global variables and public symbols) in real time even during program execution, in the same way as the Memory Window.

The results of updating and rewriting data in this window will be reflected in the Memory Window.

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

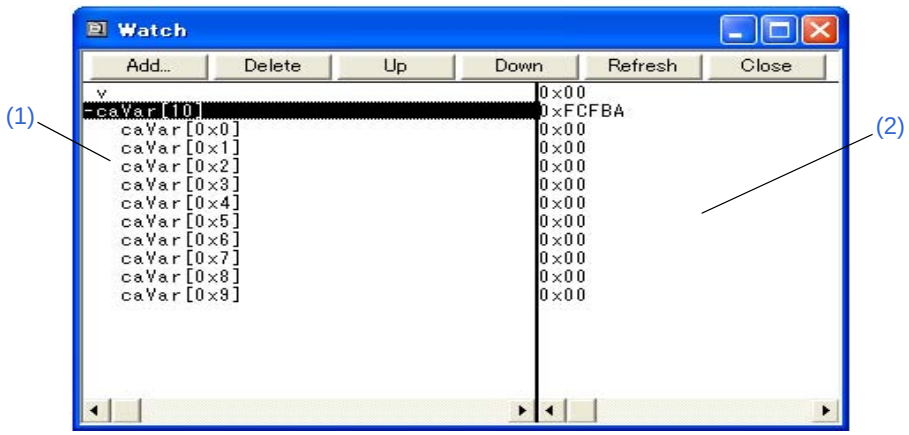
This window allows easy setting of breakpoints to variables via a Context Menu.

Remark1: If a local variable and a global variable exist with the same name, the local variable takes priority.

Remark2: A maximum of 10,000 lines can be displayed in the Watch Window.

Caution: When MINICUBE is connected, DMM (writing) to registers or SFRs cannot be performed during user program execution.

Figure 6-27 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) 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 6-8 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) 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.

Caution: If an attempt is made to change a data value during user program execution, symbols that can be written without causing a temporary break[Note1] are written as is. For symbols that need a temporary break[Note2], a dialog box appears to inform occurrence of a temporary break, a temporary break is generated, and then the symbols are written. For symbols that cannot be written even if a temporary break occurs[Note3], an error occurs and the symbols cannot be written.

Note1:

Variables allocated to the internal high-speed RAM or general-purpose registers (when IECUBE is connected)

Note 2:

Control registers, SFR, or the like (when IECUBE is connected)

Variables allocated to the internal high-speed RAM (when MINICUBE is connected and extended function is enabled)

Note 3:

Variables allocated to the internal high-speed RAM (when MINICUBE is connected and MINICUBE extended function is disabled)

The display format is as follows:

Table 6-9 Watch Window Display Format (Data)

Display Data	Contents
Integer	Hexadecimal (**0x***xxxx*) or (*xxxx***H**) Decimal (*xxxx*) or (*xxxx***T**) Octal (**0***xxxx*) or (*xxxx***Q**) Binary (**0b***xxxx*) or (*xxxx***Y**)
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	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	Displays the event information of the variable selected. If an event is set, the Event Dialog Box 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 Debugger Option Dialog Box (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 Debugger Option Dialog Box.
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 Add I/O Port Dialog Box.

Context Menu

The menu items are effective for the selected line or item, not the position where the mouse pointer was clicked (same operation as when selecting the main menu with the same name).

Beak when Access to this Variable	Creates a break event that can be accessed for read/write by using the selected item.

Break when Write to this Variable	Creates a break event that can be accessed for write by using the selected item.
Break when Read from this Variable	Creates a break event that can be accessed for read by using the selected item.
Clear	Deletes a break event corresponding to the selected item.
RRM Setting...	Sets the sampling range of the RRM function. Opens the RRM Dialog Box (when IECUBE is connected).
Event Information	Displays the event information of the variable selected. If an event is set, the Event Dialog Box is opened.
Change Watch...	Changes the selected watch data. Opens the Change Watch Dialog Box.
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 Debugger Option Dialog Box (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 Debugger Option Dialog Box.
Up	Moves the selected line one line up.
Down	Moves the selected line one line down.

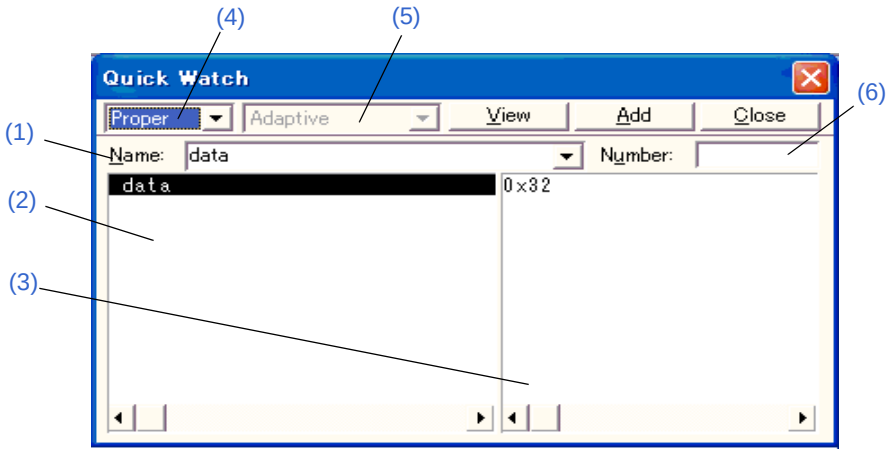
Function Buttons

Add...	Opens the Add Watch Dialog Box. If watch data is specified and the <Add...> button is clicked in the Add Watch Dialog Box, 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 6-28 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) 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.

(2) 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) Symbol name display area" in the Watch Window.) This area cannot be edited.

(3) Data value display/setting area

This area is used to display and change data values. (Refer to "(2) Data value display/setting area" in the Watch Window.)

(4) Display radix selection area

This area is used to select the display radix.

Proper	Variable: Displays the default value of each variable. Symbol: Displays data with the radix set in the Debugger Option Dialog Box.
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.

(5) Display size selection area

This area is used to select the display size.

If the display size is fixed, such as when a variable in C language or register is to be displayed, it cannot be changed.

Adaptive	Variable: Displays the default value of each variable. Symbol: Displays data with the size set in the Debugger Option Dialog Box.
Byte	Displays in 8-bit units.
Word	Displays in 16-bit units.
Double Word	Displays in 32-bit units.

(6) 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.

Function Buttons

View	Displays the data specified in (1) Name: in the field below.
Add	Adds the data specified in (1) Name: to the Watch Window.
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 6-29 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:

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 6-10 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 6-11 How a Variable Is Handled When a Scope Is Specified

Scope Specification	Program nNme	File Name	Function Name	Variable Name
prog$file#func#var	prog	file	func	var
prog$file#var	prog	file	global	var
prog$func#var	prog	global	func	var
prog$var	prog	global	global	var
file#func#var	current	file	func	var
file#var	current	file	global	var
func#var	current	current	func	var
var	current	current	current	var

(2) Radix:

This area is used to select the display radix. (Refer to "(4) Display radix selection area" in the Quick Watch Dialog Box.)

(3) Size:

This area is used to select the display size. (Refer to "(5) Display size selection area" in the Quick Watch Dialog Box.)

(4) Number:

This area is used to specify the number of data to be displayed. (Refer to "(6) Number:" in the Quick Watch Dialog Box.)

Function Buttons

Add	Adds the specified data to the Watch Window. The dialog box remains open.
OK	Adds the specified data to the Watch Window. 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 6-30 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:

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 "(4) Display radix selection area" 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 "(5) Display size selection area" 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 "(6) Number:" in the Quick Watch Dialog Box.)

Function Buttons

Add	Cannot be selected.
OK	Replaces the data on a line selected in the Watch Window 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.

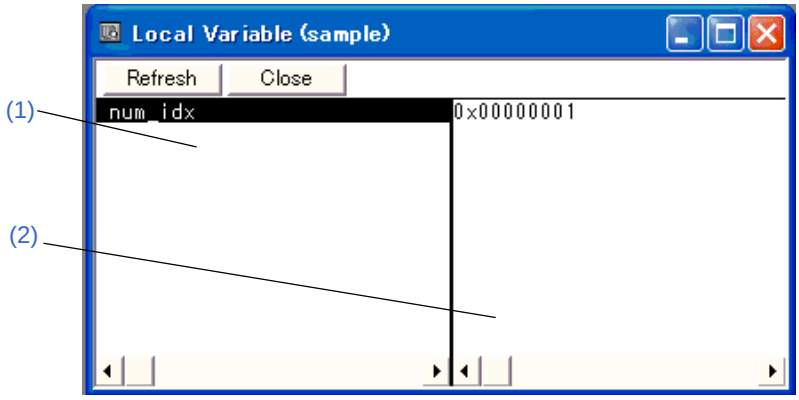
Local Variable Window

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 6-31 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) Local variable name display area

This area displays local variable name. (Refer to "(1) Symbol name display area" in the Watch Window.)

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

(2) Local variable value display/setting area

This area is used to display and change local variable values. (Refer to "(2) Data value display/setting area" in the Watch Window.) 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 (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 Add Watch Dialog Box.
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 Debugger Option Dialog Box (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".)

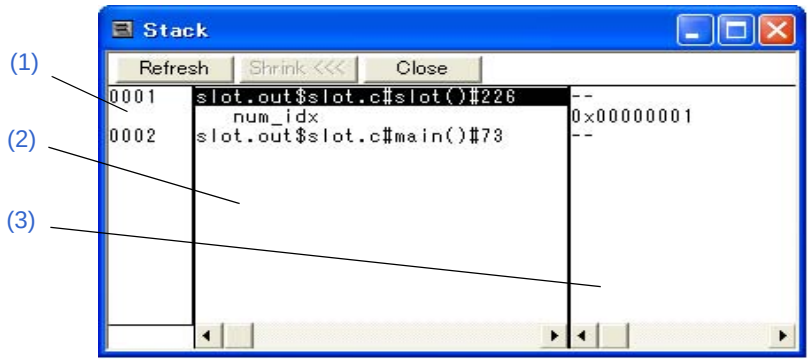
The window corresponding to the stack contents can be jumped to using the Jump function.

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 6-32 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) Stack frame number display area

This area assigns numbers to and displays the stack contents.

A stack frame number is a natural number starting from 1. The shallower the nesting of the stack, the higher the number. This means that a function having stack number one higher than that of a certain function is the function that calls the certain function.

(2) Stack frame contents display area

This area displays the stack frame contents.

It displays function names or local variable names. Note, however, that this area cannot be edited.

(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 6-8 Watch Window Display Format (Symbol)".)

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

(3) Stack contents display/setting area

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

(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 6-9 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 Debugger Option Dialog Box (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 "5.18.2 Jump function".) If no line information exists at the jump destination address, however, you cannot jump. Opens the Source Window. If an active Source Window is open, that window is displayed in the forefront (so that it can be manipulated).
Assemble	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position. (Refer to "5.18.2 Jump function".) Opens the Assemble Window. If an active Assemble Window 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 "5.18.2 Jump function".) Opens the Memory Window. If an active Memory Window 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 Trace Result with Linking Window (when IECUBE is connected), Jump function. (Refer to "5.18 Functions Common To Each Window".)

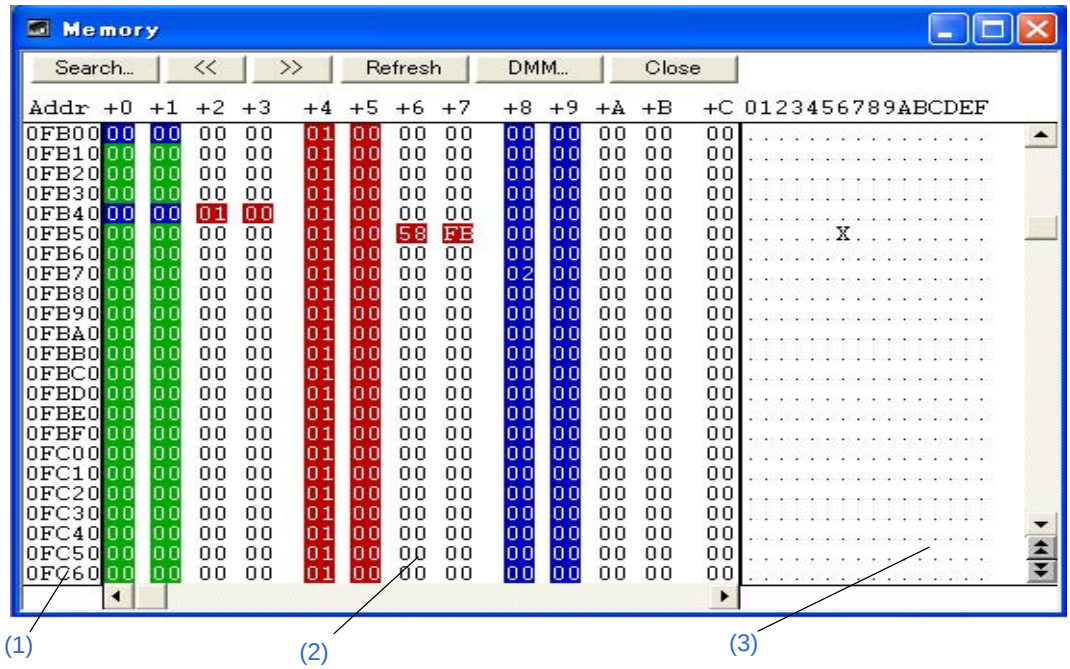
Remark1: The memory access status (read, write, read & write) can be displayed using different colors (refer to "5.7.3 Access monitor function (when IECUBE is connected)").

Remark2: The display start position when the this window is opened is as follows:

First time: Display starts from the first address of the RAM area.

Second and subsequent times: Display starts from the address at which an active status window was closed. (if an active status window has never been closed, display starts from the first display start position).

Figure 6-33 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

This area displays memory addresses.

Caution: The address width changes when memory banks are used.

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

This area is used to display and change memory contents, and to display the access status (refer to "5.7.3 Access monitor function (when IECUBE is connected)").

Display		Symbol Substituted When Display Information Is Saved in View File	Meaning
00	Green	R	Read
00	Red	W	Write
00	Blue	A	Read & write

Memory contents are changed through direct input.

The location to be changed is displayed in red and the contents of the contents of the change are written into the target memory when the Enter key is pressed. The previous value can be canceled by the ESC key. Up to 256 bytes can be specified at one time.

Remark: To change the memory contents during user program execution, open the DMM Dialog Box by clicking the <DMM...> button.

(3) 0 1 2 3....

This area is used to display and change the memory contents in ASCII characters.

This area is displayed when [View] menu -> [Ascii] is selected.

Data can be changed in this area in the same manner as in the memory display area.

The changing method is the same as in (2) +0 +1 +2.....

Remark: When the display address is changed, the position of the cursor in the ASCII display area is not synchronized.

[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. Selected: Displayed Cleared: No display (default)
Little Endian	Displays in little endian (default).
Big Endian	Displays in big endian.
Access Monitoring	This area is used to set about Access monitor function (when IECUBE is connected).
Clear	Clears the display color through the access monitor function.
Accumulative	Enables/disables cumulative display of access status (memory content change). Selected: Cumulative display of memory contents changes Cleared: Display of only memory contents changes from previous update.

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 Address Move Dialog Box.
RRM Setting...	Opens the RRM Dialog Box. (MINICUBE)
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. Selected: Displayed Cleared: No display (default)
Clear Access Monitor	Clears the display color through the access monitor function (when IECUBE is connected).
Accumulative	Enables/disables cumulative display of access status (memory content change) (when IECUBE is connected). Selected: Cumulative display of memory contents changes Cleared: Display of only memory contents changes from previous update

Function Buttons

Search...	Opens the Memory Search Dialog Box 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 Memory Search Dialog Box, 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 Memory Search Dialog Box, backward (downward on screen) from the address at the cursor position. This button is displayed as the <Stop> button during a search.
Stop(searching)	Stops searching.
Refresh	Updates the contents of the window with the latest data.
DMM...	Opens the DMM Dialog Box. (Cannot be selected when (1) Use MINICUBE Extended Function is cleared in the Extended Option Dialog Box (when MINICUBE is connected).)
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".)

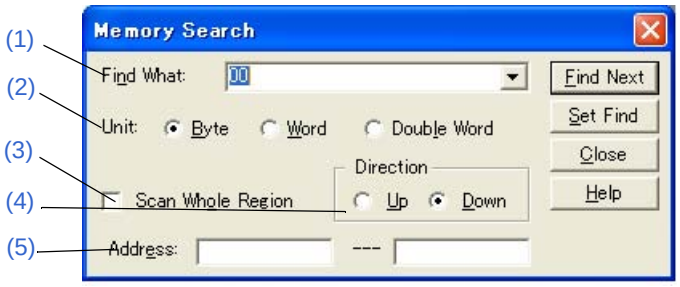
If the cursor is placed in (2) +0 +1 +2.... in the Memory Window, the specified data is treated as a binary data string, and if the cursor is placed in (3) 0 1 2 3...., the specified data is treated as an ASCII character string, and the contents of these respective areas are searched.

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.

Caution1: Non-mapped, SFR, and I/O protect areas are not searched.

Caution2: When a memory bank is used, the specification of addresses for the address space of 0x10000 or higher cannot be made for other than bank areas. (Refer to "Figure 6-6 Diagram of Address Space When Internal ROM Bank Is Used (With Bank ROM Size of 40 KB)".)

Figure 6-34 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:

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 selected 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 5-6 Specifying Symbols".)

Function Buttons

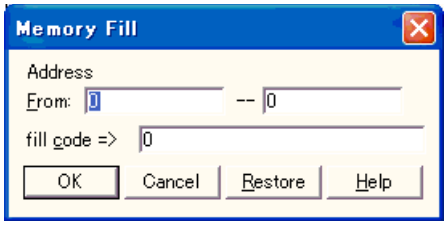
Find Next	Searches the specified data in accordance with a given condition. If the specified character string is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.
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".)

Caution: When memory banks are used, the address range cannot be specified for address spaces of address 0x10000 or higher (except for bank areas) (refer to "Figure 6-6 Diagram of Address Space When Internal ROM Bank Is Used (With Bank ROM Size of 40 KB)").

Figure 6-35 Memory Fill Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons

Opening

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

Explanation Of Each Area

(1) Address

This area is used to specify the filling range and fill code.

From:	Specifies the filling range (start address -- end address). The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to "Table 5-6 Specifying Symbols".)
fill code =>	Specify the data (fill code) used when filling the range specified in "From:". Up to 16 binary data strings (byte data strings) can be specified. Delimit each data with a "blank character".

Function Buttons

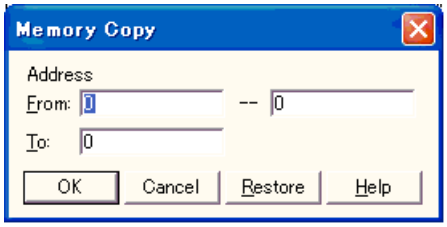
OK	Fills the specified data in accordance with a given condition.
Stop (filling)	Stops filling.
Cancel	Closes this dialog box. (During filling, this button is replaced by the <Stop> button.)
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

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".)

Caution: When memory banks are used, the address range and copy destination cannot be specified for address spaces of address 0x10000 or higher (except for bank areas) (refer to "Figure 6-6 Diagram of Address Space When Internal ROM Bank Is Used (With Bank ROM Size of 40 KB)").

Figure 6-36 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 5-6 Specifying Symbols".)

From:	Specify the address range (start address -- end address) of the copy source.
To:	Specify start address of the copy destination.

Function Buttons

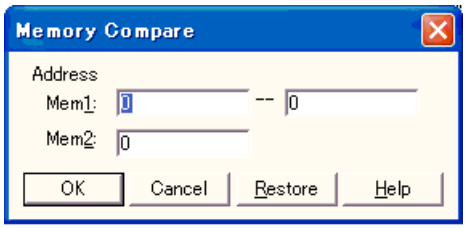
OK	Copies the memory contents in accordance with a given condition.
Stop (copying)	Stops copying.
Cancel	Closes this dialog box. (During copying, this button is replaced by the <Stop> button.)
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

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".)

Caution: When memory banks are used, the address range and compare target cannot be specified for address spaces of address 0x10000 or higher (except for bank areas) (refer to "Figure 6-6 Diagram of Address Space When Internal ROM Bank Is Used (With Bank ROM Size of 40 KB)").

Figure 6-37 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 5-6 Specifying Symbols".)

Mem1:	Specify the address range (start address -- end address) of the comparison source.
Mem2:	Specify the start address of the comparison destination.

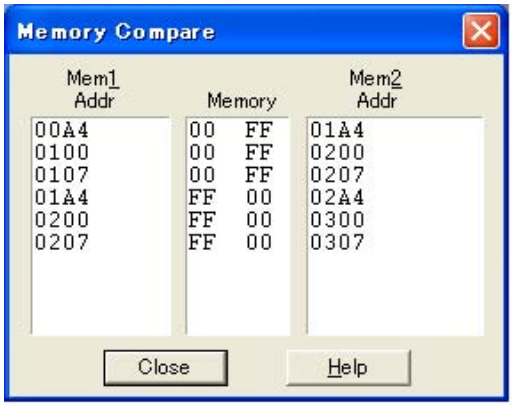
Function Buttons

OK	Compares the memory contents in accordance with a given condition.If no difference is found as a result of comparison, " Wf200: No difference encountered." is displayed. If a difference is found, the Memory Compare Result Dialog Box 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 6-38 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.

Mem1 Addr	Displays a comparison source address in which a difference has been found.
Memory	Displays the data in which a difference has been found (Left: Comparison source data, Right: Comparison destination data).
Mem2 Addr	Displays the comparison destination address at which a difference has been found.

Caution: The address width changes when memory banks are used.

Function Buttons

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

DMM Dialog Box

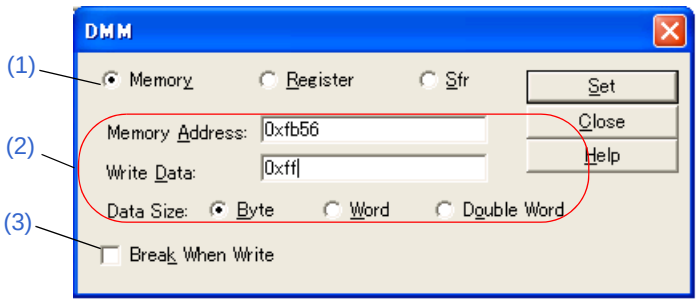
This dialog box is used to set addresses and data for DMM (Dynamic Memory Modification). (Refer to "5.16 DMM Function".) The memory contents are rewritten via the DMM function in real time during user program execution.

Caution1: This dialog box can be opened only when (1) Use MINICUBE Extended Function is selected in the Extended Option Dialog Box (when MINICUBE is connected).

Caution2: When a memory bank is used, the specification of addresses for the address space of 0x10000 or higher cannot be made for other than bank areas. (Refer to "Figure 6-6 Diagram of Address Space When Internal ROM Bank Is Used (With Bank ROM Size of 40 KB)".)

Caution3: When MINICUBE is connected, DMM (writing) to registers or SFRs cannot be performed during user program execution.

Figure 6-39 DMM Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons

Opening

Select [Edit] menu -> [DMM...], or click the <DMM...> button in the Memory Window/Register Window/SFR Window.

Explanation Of Each Area

(1) DMM target selection area

This area is used to select the target for DMM. The items displayed in (2) DMM setting area change by selecting the option button.

Memory	Memory
Register	Register (when IECUBE is connected)
Sfr	SFR (when IECUBE is connected)

Remark: If the DMM Dialog Box is opened via the Memory Window, Register Window, or SFR Window, the corresponding option button has already been selected.

(2) DMM setting area

The items displayed vary as follows, depending on the selection in (1) DMM target selection area.

(a) When Memory is selected

Memory Address:	This area is used to specify the memory address to which data is to be written. The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to "Table 5-6 Specifying Symbols".)	
Write Data:	This area is used to specify the data to be written to the memory address specified in "Memory Address:".	
Data Size:	This area is used to specify the size of the data specified in "Write Data:" to be written.	
	Byte	Writes the data as 8-bit data.
	Word	Writes the data as 16-bit data.
	Double Word	Writes the data as 32-bit data.

(b) When Register is selected (when IECUBE is connected)

Register Name:	This area is used to specify the register name to which data is to be written. The case is distinguished. Both functional and absolute names can be used for specification.
Write Data:	This area is used to specify the data to be written to the register specified in "Register Name:".
Register Bank:	Specify the register bank (0 <= setting range <=3). If this area is left blank, the current bank is specified. Specification of the control register is ignored.

(c) When Sfr is selected (when IECUBE is connected)

Sfr Name:	This area is used to specify the SFR name to which data is to be written. The case is distinguished. The read-only SFRs cannot be specified.
Write Data:	This area is used to specify the data to be written to the register specified in "SFR Name:".

(3) Break When Write

Check this item when performing DMM via software emulation (pseudo DMM function) (this item is not selected by default).

By checking this item, a break occurs instantaneously upon a write via the DMM function during the user program execution.

When IECUBE is connected, use of pseudo DMM is fixed if "Register" or "SFR" is selected in (1) DMM target selection area (not selectable when this item is selected). When MINICUBE is connected, use of pseudo DMM is fixed.

Function Buttons

Set	Writes the data in accordance with a given condition.
Close	Closes this dialog box.
Help	Displays this dialog box online help files.

Register Window

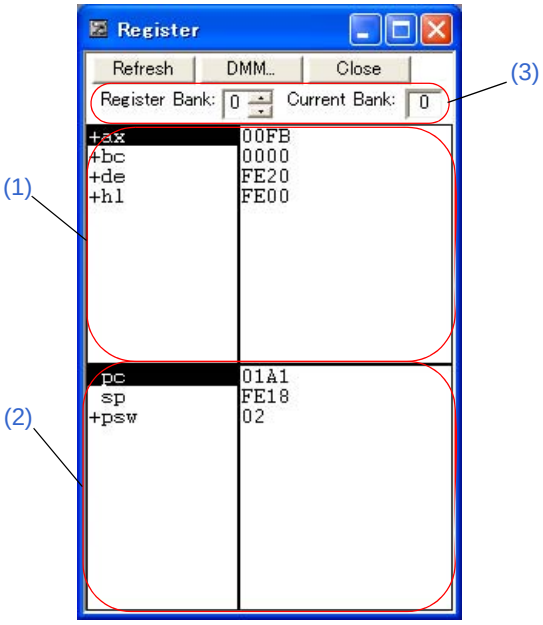
This window is used to display and change registers (general-purpose registers/control registers). (Refer to "5.8 Register Manipulation Function".)

Other operations using Context Menu, Function Buttons, etc., can be performed in this window.

Each area in this window are the jump pointer of the Jump function.

Caution: The register contents are displayed using the method set in (2) RAM Monitor in the Extended Option Dialog Box user program execution (when IECUBE is connected) (refer to "5.15 RRM Function").

Figure 6-40 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) The 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 contents of the change are written into the target memory when the Enter key is pressed. During user program execution, however, the change cannot be written (if attempted, an error occurs).

The previous value can be canceled by the ESC key.

(2) The 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 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.

Remark: When memory banks are used, the address width of the PC register increases by 8 bits.

(3) Register bank setting area

This area is used to display and specify general-purpose register bank numbers.

Register Bank:	Specifies the register bank number displayed.
Current Bank:	Displays the register bank number that is currently set.

[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 Add Watch Dialog Box.
Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays hexadecimal numbers (default).

Function Buttons

Refresh	Updates the contents of the window with the latest data.
DMM...	Opens the DMM Dialog Box. (when IECUBE is connected)
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".)

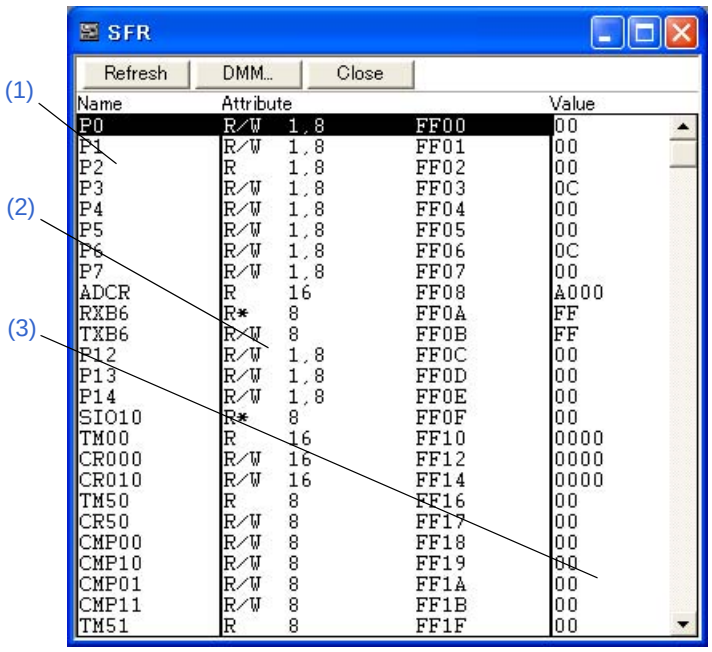
A number of other operations using Context Menu, Function Buttons, etc., can be performed in this window.

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.

Remark1: During program execution, this window is displayed in accordance with the settings of (2) RAM Monitor in the Extended Option Dialog Box (when is IECUBE is connected) (refer to "5.15 RRM Function").

Remark2: The display start position when the window is opened is as follows.
First time: Display from SFR of minimum address
Second and subsequent times: Display from first SFR when window was last closed

Figure 6-41 SFR Window



- 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

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.
32	Can be accessed in Double 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. During user program execution, however, the change cannot be written (if attempted, an error occurs). The previous value can be canceled by the ESC key.

Note that the values of read-only SFR and I/O ports cannot be changed.

The value of read-protected SFR and I/O ports can be read by selecting 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 (2) Attribute.
Pick Up	Displays only the registers selected in the SFR Select Dialog Box.
Select...	Opens the SFR Select Dialog Box.
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 Add I/O Port Dialog Box.

Context Menu

Move...	Opens the Address Move Dialog Box.
Add Watch...	Opens the Add Watch Dialog Box.
Add I/O Port...	Opens the Add I/O Port Dialog Box.
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) Attribute.
Pick Up	Displays only the registers selected in the SFR Select Dialog Box.
Select...	Opens the SFR Select Dialog Box.
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 Add I/O Port Dialog Box.

Function Buttons

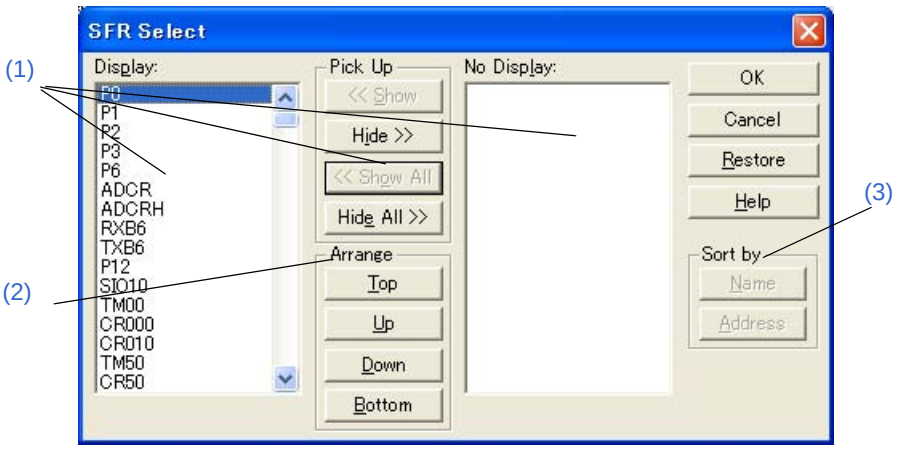
Refresh	Updates the contents of this window with the latest watch data.
DMM...	Opens the DMM Dialog Box. (when IECUBE is connected)
Close	Closes this window.

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 6-42 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:

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

(2) Arrange

The following buttons are used to change the display sequence in (a) Display:.

If the display arrangement is changed, multiple lines cannot be selected. Select one line at a time.

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 SFR Window and closes this dialog box.
Cancel	Cancels the changes and closes this dialog box.
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

Add I/O Port Dialog Box

This dialog box is used to register an I/O port to be added to the SFR Window. (Refer to "5.8 Register Manipulation Function".)

Figure 6-43 Add I/O Port Dialog Box

- Opening
- Explanation Of Each Area
- Function Buttons

Opening

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

Explanation Of Each Area

(1) I/O Port List:

This area lists the I/O ports currently registered.

If a new I/O port is registered, it is added to this list. An I/O port already registered can be selected and changed or deleted by 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 5-6 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 (Selectable only when SFR or external SFR is specified in (2) Name:)

(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 cleared (i.e., the I/O port can be both read and written).

Function Buttons

OK	Reflects the results of addition in the SFR Window 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 (1) I/O Port List:.
Delete	Deletes the I/O port selected in (1) I/O Port List:.

Timer Dialog Box

IECUBE

This dialog box is used to register and set timer event conditions, and display execution time measurement results. (Refer to "5.12 Event Function" and "5.9 Timer Function (When IECUBE Is Connected)".)

The (3) Execution time display area can be constantly displayed as the Timer Result Dialog Box by clicking the <View Always> button.

Registration and setting of timer event conditions is done by setting each item (256 items max.) in this dialog box and then clicking the <OK> button. The registered timer event conditions are managed by the Event Manager.

The number of timer event conditions that can be simultaneously used (validated) is limited (refer to "5.12.4 Number of enabled events for each event condition").

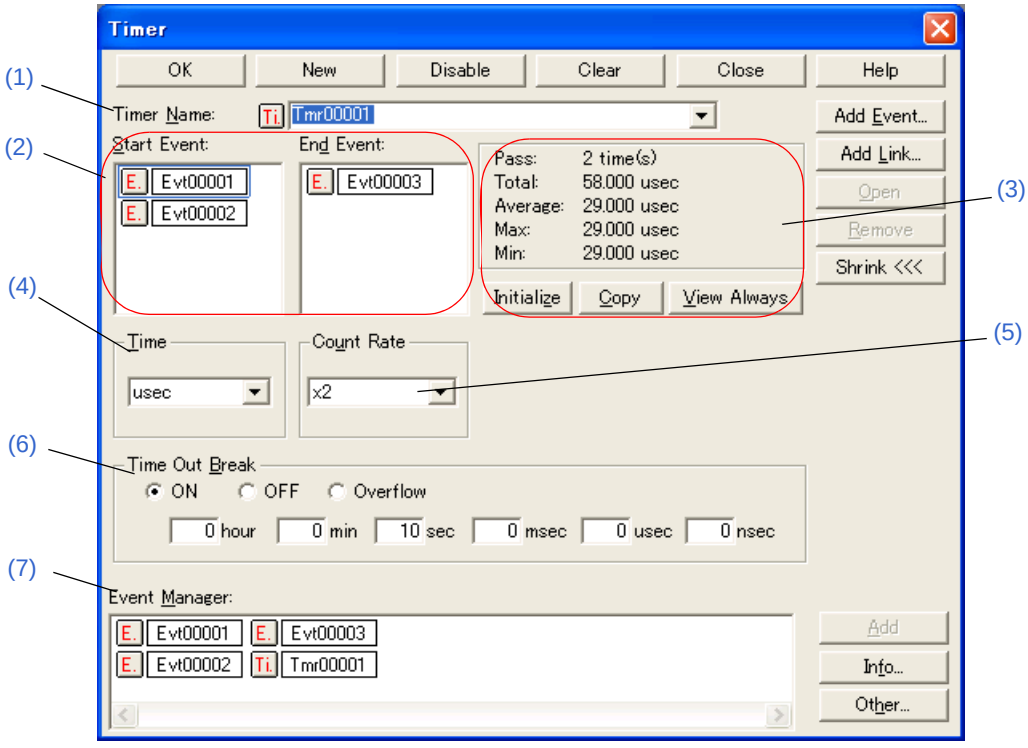
The execution time measurement result is displayed when the set timer event condition is selected.

The timer event conditions can be set, deleted, validated, or invalidated even during user program execution.

Remark1: The measurement result display contents are updated at each sampling time of the RRM Function, even during user program execution.

Remark2: Timer event condition setting/enable/disable/delete operations are possible even during user program execution.

Figure 6-44 Timer Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons (Related event function)

Opening

 Click the **Tim** button, or select [Event] menu -> [Timer...].

Explanation Of Each Area

(1) Timer Name:

This area is used to set a timer event name.

Directly input an alphanumeric string of up to eight characters as a name.

To display the contents of an already created event condition, select from the drop-down list.

To display from user program execution until break, specify "Run-Break" (refer to "5.9.2 Run-Break event").

The mark on the left of this area indicates the utilization status of events (refer to "Table 5-19 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) Start Event:, End Event:

This area is used to set an event condition for the timer.

For the number of items that can be set to this area, refer to "Table 6-17 Number of Events Settable". Setting of event conditions is easily done by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions".

(3) Execution time display area

This area displays the result of measuring the execution time of the program (refer to "Table 6-12 Measurable Values").

Pass	Number of passes
Total	Total execution time in the measurement zone specified by start event and end event conditions
Average	Average execution time
Max	Maximum execution time
Min	Minimum execution time

Caution: Measurement results that cannot be trusted due to counter overflow are displayed in red.

Table 6-12 Measurable Values

Connected IE	Measurable Execution Time	Measurable Execution Count
IECUBE	1 minute and 25 seconds max. (Resolution = 20 nsec) 48 hours and 50 minutes max. (2K division, resolution = 41 usec)	4294967295 times max. (32 bit)

(a) Button

Initialize	Clears the measurement results.
Copy	Copies the measurement result to the clipboard in text format.
View Always	Opens the Timer Result Dialog Box.

(4) Time

This area is used to select the unit in which the timer measurement result is to be displayed.

"Nsec" is displayed when a new event is created.

nsec	Nanoseconds (default)
usec	Microseconds
msec	Milliseconds
sec	Seconds
min	Minutes

(5) Count Rate

This area is used to set the timer count rate value used for execution time measurement.

The timer count rate value can be set for each timer event condition.

The rate value can be selected from x1 to x2048 from a drop-down list.

(6) Time Out Break

This area is used to set the timeout break for the section measurement time specified in (2) Start Event:, End Event: (time from the establishment of timer start event to the establishment of timer end event).

ON	A timeout break occurs (execution is terminated) if the section measurement time exceeds the specified timeout time. Specify the time-out time in the text boxes. The values up to the maximum measurable time can be specified.
OFF	No timeout break occurs (default).
Overflow	A timeout break occurs (execution is terminated) if the section measurement time exceeds the maximum measurable time (refer to "Table 6-12 Measurable Values").

Caution: "ON" cannot be selected for Run-Break event, and the time-out break cannot be set to specify the time-out time (the time-out break can be set when "Overflow" is selected).

(7) Event Manager:

This area is used to display the list of the events registered. (Refer to "Table 5-19 Event Icon", "(4) Manipulation in event manager area".)

Function Buttons

Refer to " Function Buttons (Related event function)" in the Event Manager.

Timer Result Dialog Box

IECUBE

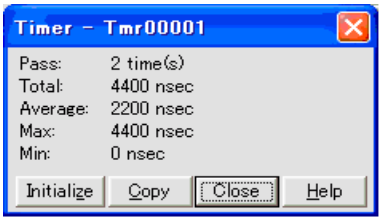
This dialog box displays the results of measuring the execution time. (Refer to "5.9 Timer Function (When IECUBE Is Connected)".)

By clicking the <View Always> button in the Timer Dialog Box, this dialog box is opened corresponding to a timer event condition on a one-to-one basis. Two or more of this dialog box can be simultaneously opened.

Up to 256 + 1 (Run-Break event) Timer Result Dialog Boxes can be opened, the number of events that can be measured at the same time is the number of valid events described in "5.12.4 Number of enabled events for each event condition" + 1 (Run-Break event).

Remark: The measurement result display contents are updated at each sampling time of the RRM Function, even during user program execution.

Figure 6-45 Timer Result Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons

Opening

Select a timer event condition in the Timer Dialog Box , click the <View Always> button.

Explanation Of Each Area

(1) Execution time display area

Same area is Timer Dialog Box. Refer to "(3) Execution time display area".

Function Buttons

Initialize	Clears the measurement results.
Copy	Copies the measurement result to the clipboard in text format.
Close	Closes this dialog box.
Help	Displays this dialog box online help files.

Trace View Window

IECUBE

This window used to display the trace results. (Refer to "5.10 Trace Function (When IECUBE Is Connected)".)

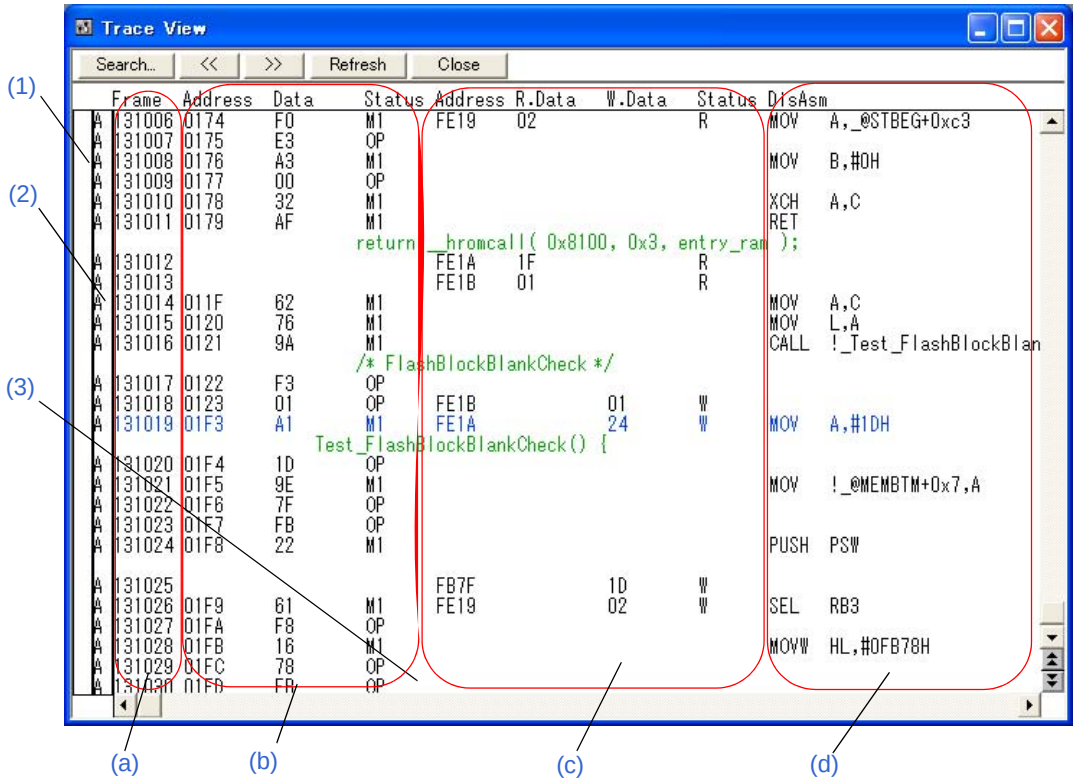
Display updates are performed during breaks or during step execution.

This window has Mixed display mode (Trace View Window).

Also, It has "5.18.3 Trace Result with Linking Window (when IECUBE is connected)".

Context Menu, A number of other operations using Function Buttons, etc., can be performed in this window.

Figure 6-46 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] -> [Trace] on the menu.

Explanation Of Each Area

(1) Point mark display area

This area displays the Event Setting Status (Event Mark).

If an execution event or access event is set at the corresponding trace address, the mark corresponding to the type of the event is displayed.

The mark displayed is not that during trace but an event mark that is set when the trace result is displayed.

(2) Trace mode display area

This area displays the type of trace mode.

A	unconditional traced or section traced frame
Q	Qualify traced frame
S	Step execute frame
T	Delay trigger frame

(3) Trace result display area

This area displays the trace results.

In theID78K0-QB, a horizontal line (block information) is displayed to indicate the end of program execution.

In addition, the cause of stopping the tracer and the reset source during the trace operation shown in "Table 6-13 Break Causes When Tracer Is Stopped" and "Table 6-14 Reset Causes During Trace Operation" are also displayed.If there are multiple break causes, all of the causes are displayed.

Table 6-13 Break Causes When Tracer Is Stopped

Cause	Meaning
Event Break	Break by event
Trace Full Break	Break because trace memory is full
Trace Delay Break	Break by trace delay
Non Map Break	Non-mapped area is accessed.
Read Protect	Read was attempted from read protected area
Write Protect	An attempt has been made to write to a write-protected area.
SFR Illegal	SFR was illegally access
SFR Read Protect	Read from prohibited SFR was attempted
SFR Write Protect	Write to write-prohibited SFR was attempted.
Step Break	Step execution break
Manual Break	Manual break
Stack Overflow	Break by stack overflow
Stack Underflow	Break by stack underflow
Uninitialize Stack Pointer	Break due to failure to perform stack pointer initialization

Cause	Meaning
Software Break	Break by software break
Uninitialize Memory read	Memory not initialized has been read.
Time Over Break	Break because execution time is over.
Unspecified Illegal	Others
Fetch Guard	Fetch guard break
Trace Stop	Trace stop
IMS(IXS) Illegal	IMS, IXS Illegal break
Flash Illegal	Flash Illegal break
Retry Over	RETRY count over break
Peripheral Break	Break from peripheral
Before Execution	Before Execution

Table 6-14 Reset Causes During Trace Operation

Cause	Meaning
Internal Reset	Reset by peripheral chip
External Reset	Reset by target
Pseudo Emulation Reset	Reset due to pseudo emulation
POC Reset	Reset due to POC

Whether each of the following sub-areas is displayed or not can be selected in the Trace Data Select Dialog Box.

(a) Frame

This area displays the trace flame number.

(b) Address Data Status (Mnemonic display)

This area displays the result of fetching the program.

(i) Status

The following types of statuses are available:

Program fetch display

M1	Fetching of first byte of instruction If the fetch address is the start of the symbol, the first line is highlighted in blue. (The frame which is BRM1 and is also M1 is included.)
OP	Opecode fetch of 2nd and subsequent bytes
IF	Invalid fetch or status unknown

Snap display

SNAP	Snap display

(ii) Address Data

This area displays the address and data.

Address	Displays the fetch address
Data	Displays the fetch data

(c) Address R.Data W.Data(Data access display)

This area displays the result of accessing data.

Status

VECT	Vector read
RW	Data read/write
R	Data read
W	Data write

(d) DisAsm (Mnemonic display)

This area displays the disassemble results.

When (i) Status is only M1, this area is displayed.

[View] menu (Trace View Window-dedicated items)

The following items are added in the [View] menu , when the Trace View Window is active.

Select...	Selects the contents to be displayed. Opens the Trace Data Select Dialog Box.
Pick Up	Performs the setting for pickup display.
Off	Does not pick up and display (default).
Search	Picks up and displays a frame that satisfies the search condition.
Snap	Picks up and displays a snap frame.
Mix	Specifies whether the source file are displayed in mixed display mode, or not displayed. Selected: Mixed display Cleared: No display (default)
Window Synchronize	Links the Trace View Window with the following windows: (Refer to "5.18.3 Trace Result with Linking Window (when IECUBE is connected)".) A selected window is linked.
Source Text	Links the Source Window.
Assemble	Links the Assemble Window.
Memory	Links the Memory Window.

Context Menu

Move...	Moves the display position. Opens the Trace Move Dialog Box.
Trace Clear	Clears the trace data.
Select...	Selects the contents to be displayed. Opens the Trace Data Select Dialog Box.
Pick Up	Performs the setting for pickup display.
Off	Does not pick up and display (default).
Search	Picks up and displays a frame that satisfies the search condition.
Snap	Picks up and displays a snap frame.
Mix	Specifies whether the source file are displayed in mixed display mode, or not displayed. Selected: Mixed display Cleared: No display (default)
Window Synchronize	Links the Trace View Window with the following windows: (Refer to "5.18.3 Trace Result with Linking Window (when IECUBE is connected)".)
Source Text	Links the Source Window.
Assemble	Links the Assemble Window.
Memory	Links the Memory Window.
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 "5.18.2 Jump function"). If no line information exists at the jump destination address, however, you cannot jump. Opens the Source Window. If the Source Window 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 "5.18.2 Jump function"). Opens the Assemble Window. If the Assemble Window 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 "5.18.2 Jump function"). Opens the Memory Window. If the Memory Window in active is open, that window is displayed in the forefront (so that it can be manipulated).

Function Buttons

Search...	Opens the Trace Search Dialog Box and searches or picks up trace results. The searched result will be highlighted in the Trace View Window. Same function as [View] menu -> [Search...].
<<	Searches forward (upward on screen) for a trace result that satisfies the search condition set in the Trace Search Dialog Box. This button cannot be selected during pickup display.
>>	Searches backward (downward on screen) for a trace result that satisfies the search condition set in the Trace Search Dialog Box. This button cannot be selected during pickup display.
Refresh	Updates the contents of the window with the latest data.
Close	Closes this dialog box.

Trace Search Dialog Box

IECUBE

This dialog box is used to search in the Trace View Window. (Refer to "5.10 Trace Function (When IECUBE Is Connected)".)

By setting each item and then clicking the <Find Next> button, searching can be started.

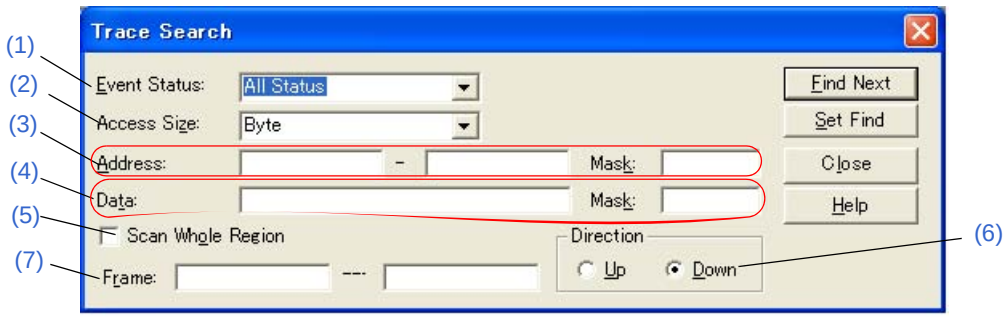
By clicking the <Set Find> button, the direction buttons (<< and >>) in the Trace View Window can be used for the search.

Remark: This dialog box is used to search trace data if it is opened by selecting [View] menu -> [Pick Up] -> [Off]. It is used to pick up and display trace data if it is opened by selecting [View] menu -> [Pick Up] -> [Search].

Caution1: This dialog box cannot be called if picking up the first M1 fetch frame (BRM1) after program branch is specified using the menu bar or in the Trace Data Select Dialog Box.

Caution2: When a memory bank is used, the specification of addresses for the address space of 0x10000 or higher cannot be made for other than bank areas (Refer to "Figure 6-6 Diagram of Address Space When Internal ROM Bank Is Used (With Bank ROM Size of 40 KB)".)

Figure 6-47 Trace Search Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons

Opening

When the Trace View Window is the current window, select [View] menu -> [Search...], or click the <Search... > button in the same window.

Explanation Of Each Area

(1) Event Status:

This area is used to select a status condition. If a status condition is omitted, all frames (All status) are searched.

All Status	All frames (default)
M1 Fetch	M1 fetch (including BRM1)
R/W	Data read/write (including R, W)
Read	Data read
Write	Data write

(2) Access Size:

This area is used to select an access size condition. By specifying an access size condition, the access width of a data condition to be detected by an access event is determined.

Byte	Searches for a data condition with 8-bit width (only during 8-bit access).
Bit	Searches for a data condition with 1-bit width (only during 8-bit access)[Note1,2]

Note1: If an access event is specified as a status condition, the alternative of Bit is not displayed. If Bit or 1 is specified, an error occurs.

Note2: In this case, a search is made for a data condition with 1-bit width. Because of the operation of the simulator, access to a bit is not directly detected; the simulator searches a dummy bit access by internally setting address conditions and data conditions as follows:

Input example:		**Setting of trace search:**
Address: FE20.1	->	Address: FE20
Data: 1		Data: 00000010B
		Mask: 11111101B

If another bit of the same address is accessed or if all the 8 bits of the same address are accessed, therefore, a trace data is searched in accordance with the specified status if the address and bit match the specified value of [address.bit].

Remark: If no access size condition is specified, a judgment is automatically made from the address condition and data condition, and the following is set:

- Bit if the address condition is set in bit units
- Byte if the data condition is set in 8-bit units

(3) Address:, Mask:(Address setting area)

This area is used to specify an address condition.

The default radix for inputting a numeric value is hexadecimal. A symbol can be also specified by a symbol or expression (refer to "Table 5-6 Specifying Symbols"). The following can be set:

Table 6-15 Settable Range of Address Condition (Trace)

Settable Range	Condition
0 <= address value <= 0xFFFF	When banks not used
0 <= mask value <= 0xFFFF	
0 <= address value <= (n<<16) | 0xFFFF	When banks used (n = max. value of bank number used)
0 <= mask value <= (n<<16) | 0xFFFF	

(a) Address:

Set an address condition (lower address - higher address) (may be omitted).

The following can be set:

(i) Setting as a point

Set a value to only the lower address, or set the same value to the lower address and the higher address.

(ii) Setting as a range

Set a value to only the lower address, or set the same value to the lower address and the higher address.

(iii) Setting as a bit

Set a value to only the lower address, or set the same value to the lower address and the higher address.

Specify a value in the form of "address.bit". Mask cannot be set.

The value of bit, which indicates the bit position, must be 0 <= bit<= 7.

(b) Mask:

Set a mask value for an address value (only when (i) Setting as a point).

Mask may be omitted.

The address value of a bit whose mask value is 1 may be 0 or 1.

Example 1:

Address	0x4000 to 0x4000
Mask	0xFF

With this setting, addresses 0x4000 to 0x40FF satisfy the condition.

Example 2:

Address	0x4000 to 0x4000
Mask	0x101

With this setting, addresses 0x4000, 0x4001, 0x4100, and 0x4101 satisfy the condition.

(4) Data:, Mask:(Data setting area)

This area is used to set data conditions.

The default radix for inputting a numeric value is hexadecimal.

The settable range differs as follows depending on the access size condition specified in (2) Access Size:. (Refer to "Table 6-21 Settable Range of Data Condition".)

(a) Data:

Set a data value as data conditions. A data can be also specified by a symbol (refer to "Table 5-6 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	0x4000
Mask	0xFF

With this setting, addresses 0x4000 to 0x40FF satisfy the condition.

Example 2:

Data	0x4000
Mask	0x101

With this setting, addresses 0x4000, 0x4001, 0x4100, and 0x4101 satisfy the condition.

(5) Scan Whole Region

This should be selected to search the entire specified range.

(6) 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 (default). Searches data backward (downward on screen) from the current position of the cursor.

(7) Frame:

This area is used to specify a frame number to be searched.

The default radix for inputting a numeric value is decimal. A symbol can be also specified by Frame Number Specification Format.

Function Buttons

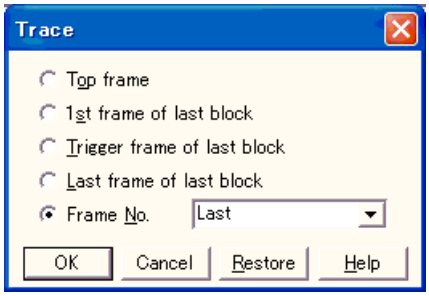
Find Next	Searches the specified data in accordance with a given condition. If the specified frame is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.
Pick Up (Stop (during search))	Picks up according to the specified condition of data search. If a frame that satisfies the condition is found as a result of a search, it is picked up. To pick up a frame that satisfies a different condition, click this button again.
Close	Closes this dialog box.
Help	Displays this dialog box online help files.

Trace Move Dialog Box

IECUBE

This dialog box is used to specify the position from which displaying the Trace View Window is started. (Refer to "5.10 Trace Function (When IECUBE Is Connected)".)

Figure 6-48 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.

Top frame	Moves the display start position to a first frame of trace data.
1st frame of last block	Moves the display start position to a first frame in the newest block frame of trace data.
Trigger frame of last block	Moves the display start position to the trigger frame in the newest block frame of trace data.
Last frame of last block	Moves the display start position to the last frame of trace data.
Frame No.	Moves the display start position to the specified frame number. (Refer to "Table 6-16 Frame Number Specification Format".) In the default condition, the character string selected in the window that called this dialog box or "Last" is selected. The default radix for inputting a numeric value is decimal. If 0 is specified, the display start position is moved to the first frame of trace data. Up to 16 input histories can be recorded.

Remark: The "block frame" is a unit of a frame that is traced from start to end of the tracer operation. An area divided by a horizontal line (block information) in the Trace View Window is one block frame.

Table 6-16 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).
Top	O	Same as "Top frame"
First	S	Same as "1st frame of last block"
Trigger	T	Same as "Trigger frame of last block"
Last	L	Same as "Last frame of last block"

Function Buttons

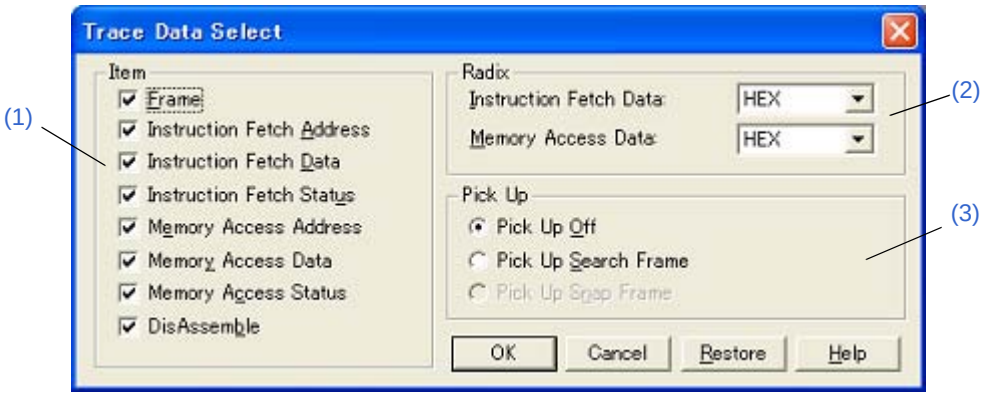
OK	Starts trace display from the specified position.
Cancel	Closes this dialog box.
Restore	Restores the input data to the original status.
Help	Displays this dialog box online help files.

Trace Data Select Dialog Box

IECUBE

This dialog box is used to select items to be displayed in the Trace View Window. (Refer to "5.10 Trace Function (When IECUBE Is Connected)".)

Figure 6-49 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

This area is used to select items to be displayed in the Trace View Window. Displaying the following items may or may not be selected. The field selected is displayed.

Frame	(a) Frame field
Instruction Fetch Address	Address field in (b) Address Data Status (Mnemonic display)
Instruction Fetch Data	Data field in (b) Address Data Status (Mnemonic display)
Instruction Fetch Status	Status field in (b) Address Data Status (Mnemonic display)
Memory Access Address	Address field in (c) Address R.Data W.Data(Data access display)
Memory Access Data	Data field in (c) Address R.Data W.Data(Data access display)
Memory Access Status	Status field in (c) Address R.Data W.Data(Data access display)
DisAssemble	(d) DisAsm (Mnemonic display) 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 field in (b) Address Data Status (Mnemonic display)
Memory Access Data	Data field in (c) Address R.Data W.Data(Data access display)

HEX	Displays hexadecimal numbers. (default)
DEC	Displays decimal numbers.
OCT	Displays octal numbers.
BIN	Displays binary numbers.

(3) Pick Up

This area is used to select a pick up condition.

Pick Up Off	No pick up display (default)
Pick Up Search Frame	Picks up and displays a frame that satisfies the search condition.
Pick Up Snap Frame	Picks up and displays a snap frame.

Function Buttons

OK	Reflects the results of selection in this dialog box in the Trace View Window.
Cancel	Closes this dialog box.
Restore	Restores the original status.
Help	Displays this dialog box online help files.

Trace Dialog Box

IECUBE

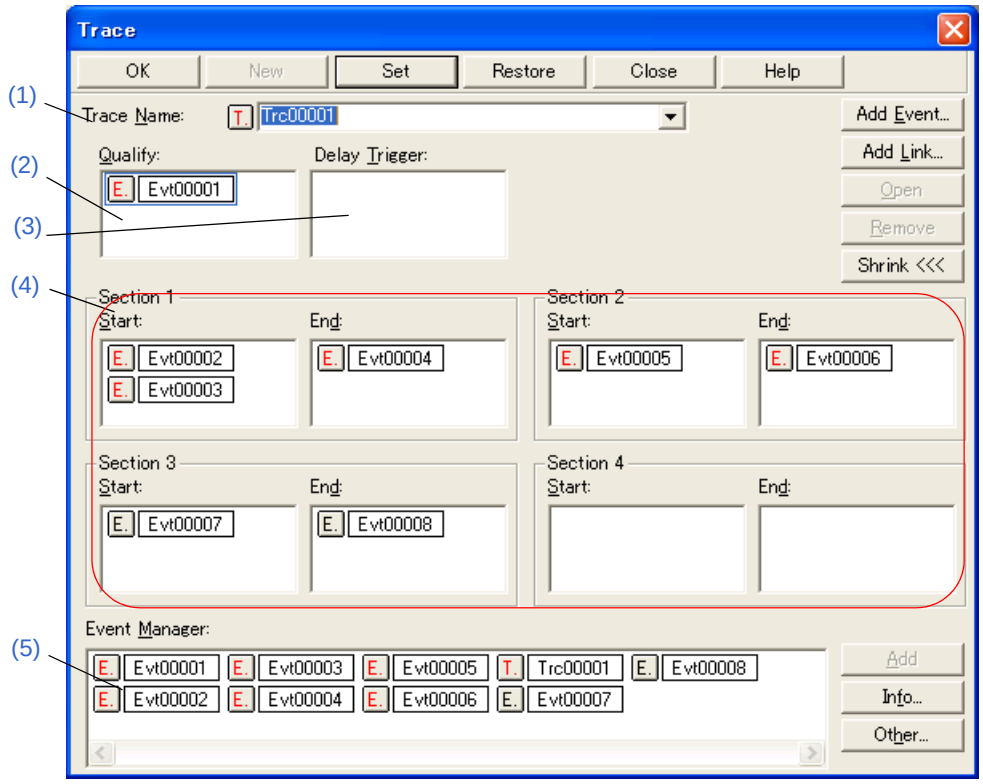
This dialog box is used to register, set, and display trace event conditions. (Refer to "5.12 Event Function", "5.10 Trace Function (When IECUBE Is Connected)".)

The trace event conditions for when performing conditional trace are specified in this dialog box (refer to "Table 5-14 Types of Conditional Trace").

Registration and setting of trace event conditions is done by setting each item (256 items max.) in this dialog box and then clicking the <OK> button. The registered trace event conditions are managed by the Event Manager.

The number of trace event conditions that can be simultaneously used (validated) is limited (refer to "5.12.4 Number of enabled events for each event condition").

Figure 6-50 Trace Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons (Related event function)

Opening

 Click the **Trc** button, or select [Event] menu -> [Trace...].

Explanation Of Each Area

(1) Trace Name:

This area is used to set a trace event name.

Directly input an alphanumeric string of up to eight characters as a name.

To display the contents of an already created event condition, select from the drop-down list.

The mark on the left of this area indicates the utilization status of events (refer to "Table 5-19 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) Qualify:

This area is used to set an event condition for a qualify trace (refer to "5.10.5 Setting conditional trace").

If two or more events are set, trace is performed when each event occurs.

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

Table 6-17 Number of Events Settable

Connected IE	Event Conditions Total (execution/access)	Event Link Conditions
IECUBE	18 (8/10)	2

Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions".

(3) Delay Trigger:

This area is used to set an event condition for a delay trigger (refer to "5.10.5 Setting conditional trace").

For the number of items that can be set to this area, refer to "Table 6-17 Number of Events Settable".

Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions".

(4) Section Start:, Section End:

Up to four sections can be set at the same time.

This area is used to set event conditions for starting and stopping a section trace (refer to "5.10.5 Setting conditional trace").

For the number of items that can be set to this area, refer to "Table 6-17 Number of Events Settable".

Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions".

(5) Event Manager:

This area is used to display the list of the events registered. (Refer to "Table 5-19 Event Icon", "(4) Manipulation in event manager area".)

Function Buttons

Refer to " Function Buttons (Related event function)" in the Event Manager.

Delay Count Dialog Box

IECUBE

This dialog box is used to set or display delay count values. (Refer to "5.10 Trace Function (When IECUBE Is Connected)".)

By setting a delay count value, a trace can be executed the number of times specified by the delay count value after the delay trigger event condition set in the Trace Dialog Box has been satisfied. (Refer to "5.10.5 Setting conditional trace".)

Figure 6-51 Delay Count Dialog Box

- Opening
- Explanation Of Each Area
- Function Buttons

Opening

Select [Event] menu -> [Delay Count...].

Explanation Of Each Area

(1)Delay Count

The following items can be selected.

FIRST	Places the trigger pointer at the first of the trace data, traces all frames, and then stops the tracer.
MIDDLE	Places the trigger pointer at the center of the trace data, traces a half of all frames, and then stops the tracer.
LAST	Places the trigger pointer at the end of the trace data and immediately stops the tracer.

Function Buttons

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

Code Coverage Window

IECUBE

This dialog box displays the code coverage measurement result (C0 coverage) (refer to "5.11 Coverage Measurement Function (When IECUBE Is Connected)").

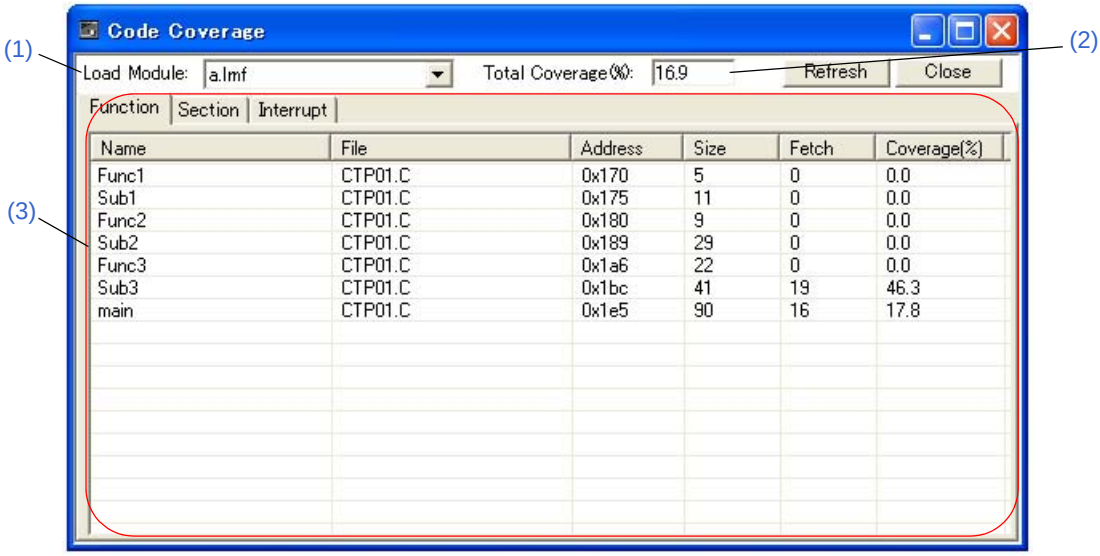
The lines where the user program has been executed or not yet executed can be checked in the Source Window or Assemble Window.

The coverage measurement result is updated at a break (it is not updated automatically during user program execution).

Caution: The coverage measurement result is inaccurate if the on-chip flash memory data is replaced via emulation of flash self programming (refer to " Flash Option Dialog Box").

Other operations using Context Menu, Function Buttons, etc., can be performed in this window.

Figure 6-52 Code Coverage Window



- Opening
- Explanation Of Each Area
- Context Menu
- Function Buttons

Opening

 Click the **Cov** button, or select [Browse] menu -> [Code Coverage].

Explanation Of Each Area

(1) Load Module:

This area is used to select the load module file that has been downloaded.

This area is blank when no load module file has been downloaded.

(2) Total Coverage (%):

This area displays the coverage for the area for which code coverage has been measured.

Total coverage = Total executed (fetched) function size/total function size

(excluding sections outside the coverage measurement range)

This area is blank when no load module file has been downloaded.

Caution: Even if all of the codes are executed, the coverage may not be 100%. For example, the jump table for case in a switch statement is data and not the code to be executed, so it is not subject to the code coverage measurement.

(3) Measurement result display area

This area displays the code coverage per tab (function, section, interrupt handler).

This area is blank when no load module file has been downloaded.

Remark: The displayed items are sorted by clicking the title (on the label) in each column (ascending/ descending order is switched each time the title is clicked).

(a) [Function] tab

Name	Function name (displayed as function in segment units in case of assembler source file)
File	Name of file in which the function is defined
Address	Function start address
Size	Function size (unit: bytes)
Fetch	Number of bytes executed (fetched)
Coverage (%)	Coverage of the function (0 - 100%) ----: When the function is outside the coverage measurement range

(b) [Section] tab

Name	Section name
Type	Section type (code, data)
Address	Section start address
Size	Section size (unit: bytes)
Fetch	Number of bytes executed (fetched)
Coverage (%)	Coverage of the section (0 - 100%) ----: When the section is outside the coverage measurement range

(c) [Interrupt] tab

Name	Interrupt request name
Type	Interrupt type (nonmaskable, maskable, software, security id, flash mask option)
Status	Utilization status in the program ----: Unknown
Address	Starting address of the interrupt handler
Size	Size of the interrupt handler (unit: bytes) Maximum size for statuses other than "use"
Fetch	Number of fetched bytes
Coverage (%)	Coverage of the interrupt handler (0 - 100%) ----: When the interrupt handler is outside the coverage measurement range

The display jumps from this tab to the Source Window or Assemble Window using the start address value of the selected line as a jump pointer. The jump destination window will be displayed from the jump pointer.

The jump function is executed by selecting a jump source line then selecting [Source Text/Assemble] in the [Jump] menu. Jump can also be performed by double-clicking the jump source line.

Remark: Update data using the <Refresh> button during user program execution.

Context Menu

Source Text	Displays the corresponding source text and source line, using the data value at the cursor position as the jump destination address. (Refer to "5.18.2 Jump function".) If no line information exists at the jump destination address, however, you cannot jump. Opens the Source Window. If an active Source Window is open, that window is displayed in the forefront (so that it can be manipulated).
Assemble	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position (refer to "5.18.2 Jump function"). Opens the Assemble Window. If an active Assemble Window is open, that window is displayed in the forefront (so that it can be manipulated).
Clear	Clears the coverage measurement results.

Function Buttons

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

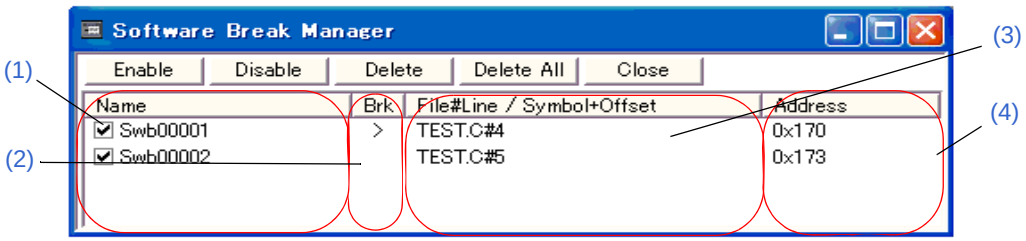
Software Break Manager

This window is used to display, enable or disable, and delete software breaks. (Refer to "5.4.4 Hardware break and software break".)

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

Remark: The displayed items are sorted by clicking the title (on the label) in each column (ascending/ descending order is switched each time the title is clicked).

Figure 6-53 Software Break Manager



- Opening
- Explanation Of Each Area
- Function Buttons

Opening

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

Explanation Of Each Area

(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 selected. To be disable, the check box is cleared.

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.

(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 dose not correspond to the source line.)

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

(4) Address

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

Function Buttons

Enable	Enables the selected event.
Disable	Disables the selected event.
Delete	Deletes the selected event.
Delete All	Deletes all the set software break events.
Close	Closes this window.

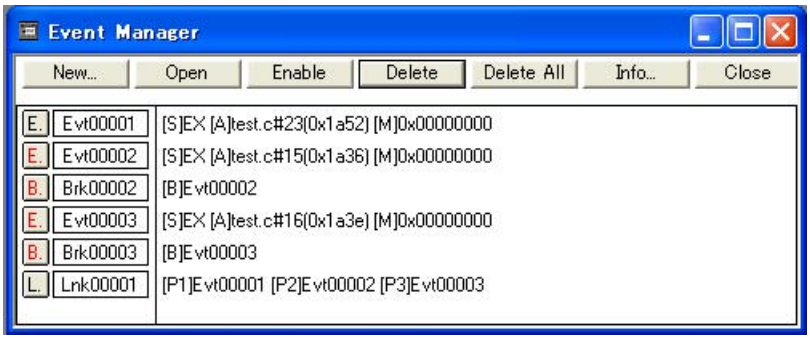
Event Manager

This window is used to manage event conditions. This window allows display, enabling/disabling, and deletion of the Various Event Conditions. (Refer to "5.12 Event Function".)

Other operations using Context Menu, Function Buttons (Related event function), etc., can be performed in this window.

The event icon is the jump pointer of the Jump function.

Figure 6-54 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] -> [Event Manager] on the menu.

Explanation Of Each Area

(1) Event display area

This area displays the icons (event icons) of the registered Various Event Conditions.

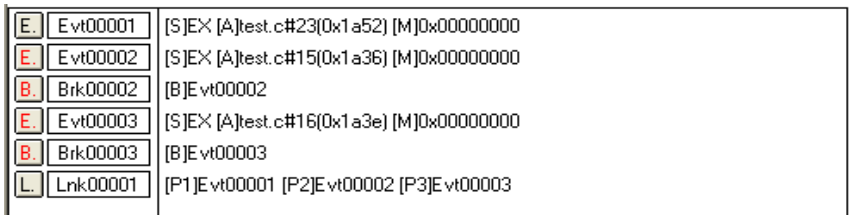
By selecting the context menu -> [Detail], the details can be displayed.

(a) [In list displayed]

Displays event icon (refer to "Table 5-19 Event Icon").

The event icon is the jump pointer (refer to "5.18.2 Jump function").

(b) In detailed display



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

Table 6-18 Separator for Displaying Event Details

Key Information	Contents
Event condition	
[S]	Status condition
[Z]	Access size condition
[AR]	Address range condition
[A]	Address condition Symbol or expression: (actual address)
[D]	Data condition Symbol or expression: (actual address)
[DR]	Data range condition
[P]	Pass count condition (when IECUBE is connected)
[M]	Mask condition
Event link condition (when IECUBE is connected)	
[P1] - [P4]	Event link condition on "n"th line
[D]	Disable condition
[P]	Pass count condition
Break condition	
[B]	Break condition
Trace condition (when IECUBE is connected)	
[M]	Tracer control mode
[T]	Delay trigger condition
[D]	Delay Count
[S1] - [S4]	Section trace start condition
[E1] - [E4]	Section trace end condition
[Q]	Qualify trace condition
Timer condition (when IECUBE is connected)	
[S]	Timer measurement start condition
[E]	Timer measurement end condition

Key Information	Contents
[R]	Timer division ratio
[U]	Timer measurement unit
[B]	Timeout break condition
Snapshot condition (when IECUBE is connected)	
[SN]	Snapshot condition
[R]	Register condition
[B]	Register bank condition
[M]	Memory condition Symbol or expression: (actual address)
[Z]	Access size condition
[F]	SFR condition
Stub condition (when IECUBE is connected)	
[SU]	Stub condition
[A]	Jump address Symbol or expression: (actual address)
Event DMM condition (when IECUBE is connected)	
[DM]	Event DMM condition
[E]	SFR condition
[D]	Modified data
[M]	Memory condition
[A]	Data modification address
[Z]	Size of modified data

[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	Detail display
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	List display (default).
Source Text	Displays the corresponding source text and source line, using the position of the selected event as the jump destination address. (Refer to "5.18.2 Jump function".) If no line information exists at the jump destination address, however, you cannot jump. Opens the Source Window. If an active Source Window is open, that window is displayed in the forefront (so that it can be manipulated).
Assemble	Displays the Assemble window from the position of the selected event, which is used as the jump destination address. (Refer to "5.18.2 Jump function".) Opens the Assemble Window. If an active Assemble Window is open, that window is displayed in the forefront (so that it can be manipulated).
Memory	Displays the memory contents from the position of the selected event, which is used as the jump destination address. (Refer to "5.18.2 Jump function".) Opens the Memory Window. If an active Memory Window 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, Event Dialog Box and each various event setting dialog boxes (refer to "Table 5-17 Various Event Conditions").

OK	(Event Dialog Box, Event Link Dialog Box) Automatically registers the event condition being edited, if any, and closes this dialog box. **In the select mode** 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.
	(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.
New...	(Event Manager) Opens the dialog box to create new event condition. By clicking each button, the corresponding event setting dialog box can be opened with the new event name set. After the event setting dialog box has been opened, this dialog box is closed. Returns to Event Manager by clicking the <Cancel> button.
	(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, Event Link Dialog Box) Registers the various event conditions. Because the dialog box is not closed even after an event has been registered, new event conditions can be registered. **In the select mode** 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 conditions. Because the dialog box is not closed even after an event has been registered, new event conditions can be registered. Each event condition becomes valid as soon as it has been registered.
Enable / Disable	Validates (enables) or invalidates (disables) the selected event condition. However, event conditions and event link conditions cannot be enabled or disabled. Same operation as the clicking the mark of event icon.
Clear	Clears the contents of the event condition.
Restore	Restores the contents of an edited event condition. If an event condition not registered is displayed, all the fields other than the event name field are blank or the default values are set.
Cancel / Close	Closes this dialog box. Even if an event condition is being edited, it is not registered and the dialog box is closed.
Help	Displays the help window of this window.
Event Link...	Opens the Event Link Dialog Box (when IECUBE is connected).
Break...	Opens the Break Dialog Box.
Trace...	Opens the Trace Dialog Box (when IECUBE is connected).
Snap Shot...	Opens the Snap Shot Dialog Box (when IECUBE is connected).
Stub...	Opens the Stub Dialog Box (when IECUBE is connected).
Timer...	Opens the Timer Dialog Box (when IECUBE is connected).
Event DMM...	Opens the Event DMM Dialog Box (when IECUBE is connected).
Manager	Opens the Event Manager.
Add Event...	Opens the Event Dialog Box in the select mode, and selects or newly creates an event condition to be set. The event condition will be added to the area selected when the < Add Event...> button is clicked.
Add Link...	Opens the Event Link Dialog Boxin the select mode, and selects or newly creates an event link condition. The event condition will be added to the area selected when the < Add Link...> button is clicked.
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. When an event condition or an event link condition is to be deleted, an error occurs and the event condition or event link condition cannot be deleted if the event is used as a various event condition.
Delete All	Deletes all event conditions except software break events
Expand >>> / Shrink <<<	Turns on (Expand>>>) or off (Shrink<<<) display of the event manager area. The size of the dialog box is expanded or reduced.

Add	The event condition and event link condition selected in Event Manager area add to setting area with a focus.
Info...	Opens the Select Display Information dialog box. This dialog box is used to change the display mode and rearrange event names. Figure 6-55 Select Display Information Dialog Box  <Sort by Name> .....Sorts events into name order. <Sort by Kind>.....Sorts events into type order. <Unsort>.....Displays events in the order in which they have been registered without sorting the events. <Detail> .....Sets the detailed display mode. <Overview> .....Sets the list display mode. <Cancel> .....Closes this dialog box (same as ESC key).
Other...	Opens the dialog box for selecting the event type. By clicking each button, the corresponding event setting dialog box can be opened with the new event name set. After the event setting dialog box has been opened, this dialog box is closed. <Manager...> .....Opens the Event Manager. <Cancel> .....Closes the dialog box to create event condition.

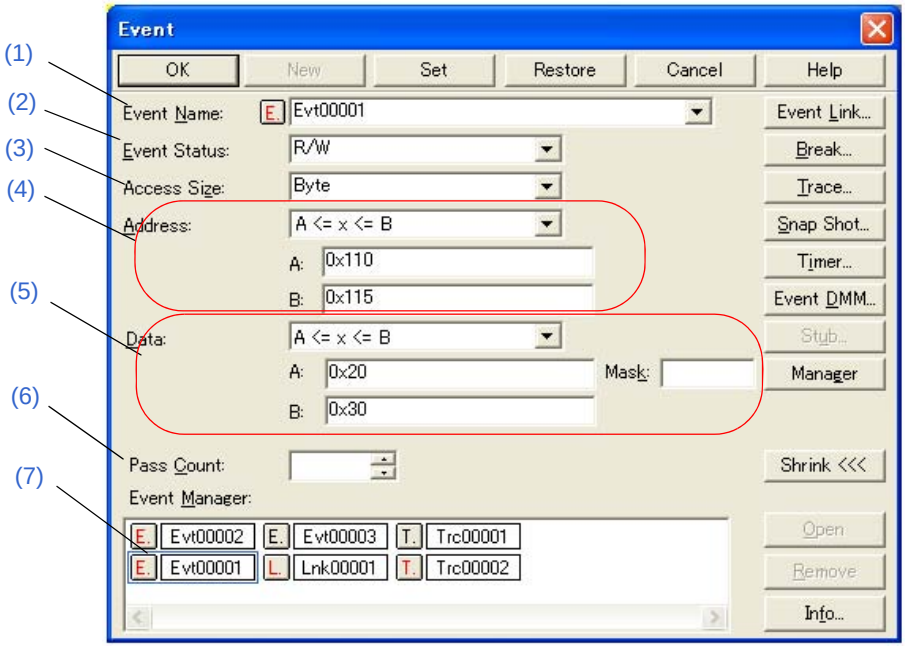
Event Dialog Box

This dialog box is used to register and display event conditions. (Refer to "5.12 Event Function".)

Setting of event conditions is done by setting each item in this dialog box and then clicking the <OK> button. The registered event conditions are managed by the Event Manager.

One event condition can be set for multiple Various Event Conditions. However, the number of event conditions that can be simultaneously used is limited (refer to "5.12.4 Number of enabled events for each event condition").

Figure 6-56 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 the **Evn** button, or select [Event] -> [Event...] on the menu.

In select mode

If the <OK> button is clicked when the Event Dialog Box has been opened as follows, an event condition can be registered in the setting dialog box from which this dialog box was opened (the setting dialog box from which the this box was opened is displayed on the title bar.).

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

Explanation Of Each Area

(1) Event Name:

This area is used to set an event name.

Directly input an alphanumeric string of up to eight characters as a name.

To display the contents of an already created event condition, select from the drop-down list.

In the select mode, the selected event condition can be set in the event condition setting area of the setting dialog box that called the Event Dialog Box.

The mark on the left of this area indicates the utilization status of events (refer to "Table 5-19 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 select a status condition.

By specifying a status condition, the type of the execution event and an access event is determined (if an execution event is specified, nothing can be input to the (3) Access Size: and (5) Data:, Mask:).

The status conditions that can be specified are listed below.

Table 6-19 Status Condition

Status	Abbreviation	Meaning
Execution event		
Execution	EX	Program execution (when IECUBE is connected)
Before Execution	EX-B	Program execution (break before execution) **Note** (when IECUBE is connected)
Access event		
R/W	RW	Memory read/write
Read	R	Memory read
Write	W	Memory write

Note: Only usable for break conditions

(3) Access Size:

This area is used to select an access size condition.

By selecting an access size condition from the drop-down list, the access width of a data condition to be detected by an access event is determined.

Byte	Detects data condition with 8-bit width (only during 8-bit access).
Word	Detects data condition with 16-bit width (only during 16-bit access). (IECUBE)

Bit	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 ID78K0-QB detects a dummy bit access by internally setting address conditions and data conditions as follows: **Input example:** -> **Setting of emulator:** Address: FE20.1 -> Address: FE20 Data: 1 Data: 00000010B Mask: 11111101B If another bit of the same address is accessed or if all the 8 bits of the same address are accessed, therefore, an event is detected in accordance with the specified status if the address and bit match the specified value of [address.bit]. 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.

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
- Word if the data condition is set in 16-bit units

(4) Address:

This area is used to specify an address condition (address value).

Caution: When a memory bank is used, the specification of addresses for the address space of 0x10000 or higher cannot be made for other than bank areas. (Refer to "Figure 6-6 Diagram of Address Space When Internal ROM Bank Is Used (With Bank ROM Size of 40 KB)".)

Table 6-20 Settable Range of Address Condition (Event)

Settable Range	Remark
When IECUBE is connected (when bank not used)	
0 <= address value <= 0xFFFF	-
0 <= mask value <= 0xFFFF	-
When IECUBE is connected (when bank used)	
0 <= address value <= (n <<16) | 0xFFFF	n = max. value of bank number used
0 <= mask value <= (n << 16) | 0xFFFF	n = max. value of bank number used
When MINICUBEis connected	
0xF800 <= address value <= 0xFFFF	-

(a) Address:

Select the address range (specification method) from a drop-down list and specify the address value in areas **A:** and **B:**:.

When MINICUBE is connected, only x == A can be selected.

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).

x == A	Specify address value specified for **A:**.
x >= A	Specify address value higher than address value specified for **A:**.
x <= B	Specify address value lower than address value specified for **B:**.
A <= x <= B	Specify address value within address range from **A:** to **B:**. In this case, 2 events are used when item other than before execution event is specified.
x < A && B < x	Specify value outside address range from **A:** to **B:**.

(5) Data:, Mask:

This area is used to specify an data condition (data value, mask value).

The default radix for inputting a numeric value is hexadecimal.

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

Table 6-21 Settable Range of Data Condition

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

(a) Data:

Select the data range (specification method) from a drop-down list and specify the data value in areas **A:** and **B:**:.

When MINICUBE is connected, "No Condition" or "x== A" can be selected.

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).

No Condition	Don’t care (specify all data values).
x != A	Specify data value not matching data value specified for **A:**.
x == A	Specify data value matching data value specified for **A:**.
x >= A	Specify data value higher than data value specified for **A:**.
x <= B	Specify data value lower than data value specified for **B:**.
A <= x <= B	Specify data value within data range from **A:** to **B:**. In this case, 2 events are used.
x < A && B < x	Specify data value outside data range from **A:** to **B:**.

(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	0x4000
Mask	0xFF

With this setting, addresses 0x4000 to 0x40FF satisfy the condition.

Example 2:

Data	0x4000
Mask	0x101

With this setting, addresses 0x4000, 0x4001, 0x4100, and 0x4101 satisfy the condition.

(6) Pass Count

A pass count specifies how many times an event condition must be satisfied during user program execution before a given condition is satisfied (When IECUBE is connected). If no pass count is specified, 1 is assumed (the condition is satisfied as soon as the event condition is satisfied). This area is invalid if Before Execution is specified in the (2) Event Status:.

Table 6-22 Pass Count

Device	Range
IECUBE	1 - 255

(7) Event Manager:

This area is used to display the list of the events registered. (Refer to "Table 5-19 Event Icon", "(4) Manipulation in event manager area".)

Function Buttons

Refer to " Function Buttons (Related event function)" in the Event Manager.

Event Link Dialog Box

IECUBE

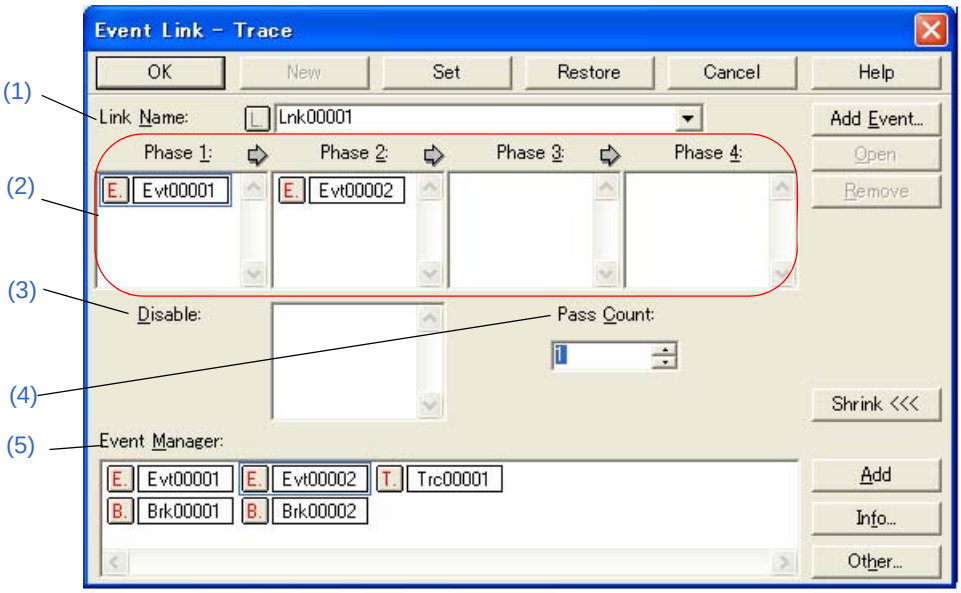
This dialog box is used to register and display event link conditions. (Refer to "5.12 Event Function".)

Registration of event link conditions is done by setting each item (256 items max.) in this dialog box and then clicking the <OK> button. The registered event link conditions are managed by the Event Manager.

However, the number of event link conditions that can be simultaneously used is limited ("5.12.4 Number of enabled events for each event condition").

Caution: Event conditions that can be specified in the Event Link Dialog Box are: other than events before execution, and with a pass count value of "0" or "1" (event conditions with a pass count value of 2 or larger cannot be used).

Figure 6-57 Event Link Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons (Related event function)

Opening

In normal mode

If the Event Link Dialog Box is opened as follows, an event link condition can be registered without its purpose being specified.

Select [Event] menu -> [Event Link...].

In select mode

If the <OK> button is clicked when the Event Link Dialog Box has been opened as follows, an event link condition can be registered in the setting dialog box from which this dialog box was opened.

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

(the setting dialog box from which the Event Link Dialog Box was opened is displayed on the title bar.)

Explanation Of Each Area

(1) Link Name:

This area is used to set a event link name.

Directly input an alphanumeric string of up to eight characters as a name.

To display the contents of an already created event link condition, select from the drop-down list.

In the select mode, the selected event condition can be set in the event link condition setting area of the setting dialog box that called the Event Link Dialog Box.

The mark on the left of this area indicates the utilization status of event link conditions ("Table 5-19 Event Icon"). The mark "L" in gray indicates that an event link condition is being edited and has not been registered yet.

(2) Phase1:, Phase2:, Phase3:, Phase4:

This area is used to specify the sequence in which event conditions and events are detected.

Up to four sequences can be specified. If a disable condition is detected while the program is being executed, however, the event conditions that have so far been satisfied are initialized, and the event conditions are detected again starting from the first event condition. If a link condition and a disable condition are detected at the same time, the disable condition takes precedence.

Set Phase 1 -> Phase 2 -> Phase 3 -> Phase 4, in that order.Phase 4 does not have to be set. In this case, an event occurs when the event condition set for the last phase has been detected. An event condition can be set for only Phase 1 or the same event condition can be set for two or more Phases.

The number of event conditions that can be set to each phase of this area and while the dialog box, is as follows:

Table 6-23 The Number of Event Conditions in Event Link Dialog Box

Connected IE	Each Phase	Disable Area (Execution/ Access)	Total (Execution/ Access)
IECUBE	18	18	18 (8/10)

(3) Disable:

This area is used to set an event condition that invalidates the event conditions that have so far been satisfied (refer to "Table 6-23 The Number of Event Conditions in Event Link Dialog Box").

Setting of event conditions is easily done by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions".

(4) Pass Count:

This area is used to set a pass count condition (settable range: 0 - 255).

A pass count condition specifies how many times an event condition must be satisfied during user program execution before a given condition is satisfied.

If no pass count is specified, 1 is assumed (the condition is satisfied as soon as the event condition is satisfied).

(5) Event Manager:

This area is used to display the list of the events registered. (Refer to "Table 5-19 Event Icon", "(4) Manipulation in event manager area".)

Function Buttons

Refer to " Function Buttons (Related event function)" in the Event Manager.

Break Dialog Box

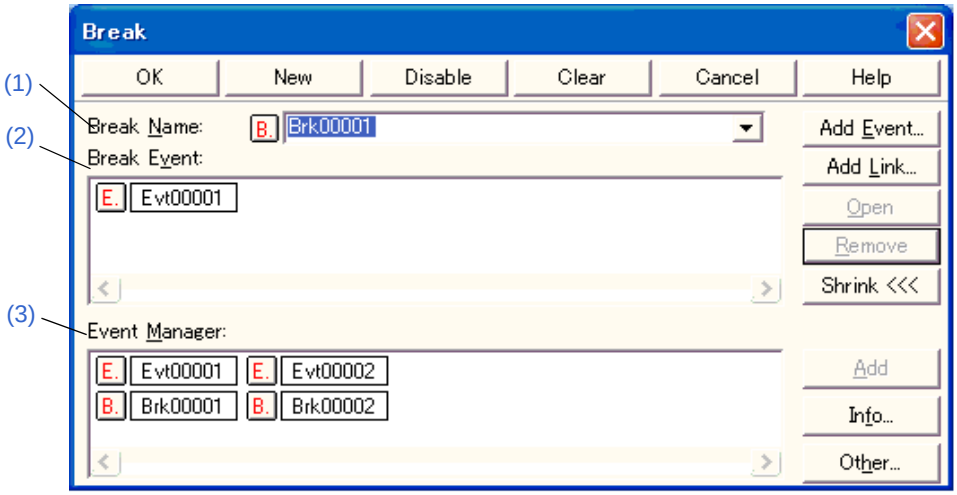
This dialog box is used to register; set, and display break event conditions. (Refer to "5.12 Event Function", "5.4 Break Function".)

Registration and setting of break event conditions is done by setting each item (256 items max.) in this dialog box and then clicking the <OK> button. The registered break event conditions are managed by the Event Manager.

There are restrictions on the number of break event conditions that can be simultaneously set (enabled). (Refer to "5.12.4 Number of enabled events for each event condition".)

Remark: Break event condition setting/enable/disable/delete operations are possible even during user program execution.

Figure 6-58 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:

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 5-19 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 and event link conditions that can be set in this area is as follows:

Table 6-24 Number of Events Settable in Condition Setting Area

Connected IE	Total Event Conditions (Before Execution / After Execution / Access)	Event Link Conditions
IECUBE	34 (16 / 8 / 10)	2
MINICUBE	2 (1[a] / 0 / 1)	-

a. When the software break is used, it is 0

Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions".

(3) Event Manager:

This area is used to display the list of the events registered. (Refer to "Table 5-19 Event Icon", "(4) Manipulation in event manager area".)

Function Buttons

Refer to " Function Buttons (Related event function)" in the Event Manager.

Snap Shot Dialog Box

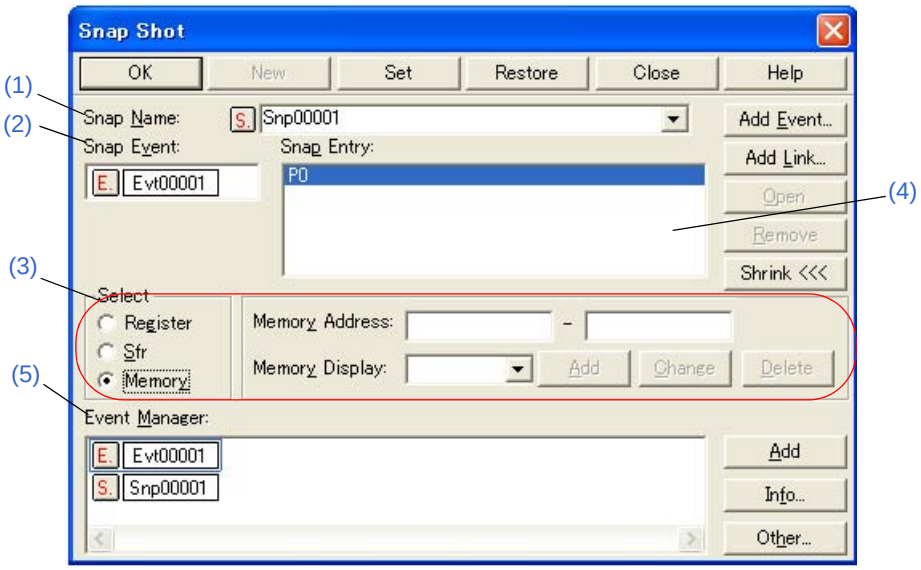
IECUBE

This dialog box is used to register; set, and display snapshot event conditions. (Refer to "5.12 Event Function", "5.13 Snapshot Function (When IECUBE Is Connected)".)

Registration and setting of snapshot event conditions is done by setting each item (256 items max.) in this dialog box and then clicking the <OK> button. The registered snapshot event conditions are managed by the Event Manager.

There are restrictions on the number of snapshot event conditions that can be simultaneously set (enabled) (refer to "5.12.4 Number of enabled events for each event condition").

Figure 6-59 Snap Shot Dialog Box (When "Register" Is selected)



- Opening
- Explanation Of Each Area
- Function Buttons (Related event function)

Opening

Select [Event] menu -> [Snap Shot...].

Explanation Of Each Area

(1) Snap Name:

This area is used to set a snapshot 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 5-19 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) Snap Event:

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

The number of event condition and event link condition that can be set in this area, refer to "Table 6-17 Number of Events Settable".

Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions".

(3) Select

This area is used to register, change, and delete the snap data, selecting type of data subject to snapshot.

By selecting a data type, the item to be displayed in the snap data setting area on the right of this area changes in accordance with the selected area.

As snap data, up to 16 registers, SFR, and memory addresses each can be registered; therefore, a total of 48 registers, SFR, and memory addresses can be registered.

The snap data registered, changed, or deleted in this area is reflected in (4) Snap Entry:. If snap data is selected in (4) Snap Entry:, the contents of the selected snap data are displayed in this area.

(a) Register

This should be selected to set by a register as snap data.

(i) Register Name: (Register name setting area)

To specify a register name, either directly input one to the text box, or select one from the drop-down list. The case is distinguished.

A register name can be specified as both a function name and an absolute name.

Remark1: Select "All" to specify all registers.

Remark2: The names of general-purpose registers and control registers can be specified as Register Name.

Remark3: The snap data displayed in the Trace View Window is unified in uppercase letters and in the format of 'absolute name'

(ii) Register Bank (Register bank setting area)

To specify a register bank, either directly input one to the text box, or select one from the drop-down list (0 - 3).

Remark: To specify the current bank, select 'Current'. However, the current bank will be specified even when the specification is omitted.

(b) Sfr

This should be selected to set by a SFR as snap data.

(i) Sfr name:(SFR snap data setting area)

To specify SFR name, either directly input one to the text box, or select one from the drop-down list.

Only the SFR that can be read can be specified. The case is distinguished.

Note that all the register names are displayed in uppercase characters in the Trace View Window.

(c) Memory

This should be selected to set by a Memory as snap data.

(i) Memory Address: (Memory address setting area

Start address - End address

This area is used to specify an address range of the memory. If a value is input as only the start address and specifying the end address is omitted, it is assumed that the same value as the start address is specified as the end address. If the specified address range cannot be divided by the access size, the address range is rounded up to a range that can be divided by the access size.

An address can be also specified by a symbol or expression (refer to "Table 5-6 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

If addresses are registered or changed in this area by using expressions or symbols, the converted address values are displayed in (4) Snap Entry: along with the specified expressions and symbols.

In the Trace View Window, only the converted address values are displayed.

Table 6-25 Address Settable Range (Snap Shot Dialog Box)

Range
0 <= address value <= 0xFFFF

(ii) Memory Display: (Memory display size setting area)

To specify the access size, either directly input the size or select the size from the drop-down list. The case is distinguished.

Byte	B	Snapshot of memory in 8-bit units
Word	W	Snapshot of memory in 16-bit units
Double Word	DW	Snapshot of memory in 32-bit units

(d) Buttons to manipulate snap data

The following buttons are used to register, change, and delete snap data.

Insert	Registers as snap data. The registered snap data is inserted and displayed at the selected position in (4) Snap Entry:.
Change	Changes the contents of the snap data selected in (4) Snap Entry: area to the contents of the snap data specified in this area.
Delete	Deletes the snap data selected in (4) Snap Entry:. The DEL key performs the same operation.

(4) Snap Entry:

This area displays a list of registered snap data.

The registered snap data is written into the tracer when a snapshot event occurs.

If snap data is selected in this area, the contents of the selected snap data are displayed in each setting area.

Snap data is displayed as follows:

Table 6-26 Snap Data Display Format

Display Example	Contents
Register snap data:	
RP0[0] RP[Current] All[2]	Register name [bank number or 'Current']
SFR snap data:	
PM0 PM1	SFR name
Memory snap data:	
0xFE20<byVar>,B0xFE22<wVar> ,W0xFE30<szVar> - 0xFE2F<szVar+0x10>,B	
Start address <symbol expression> - End address <symbol expression>, Access size	

(5) Event Manager:

This area is used to display the list of the events registered. (Refer to "Table 5-19 Event Icon", "(4) Manipulation in event manager area".)

Function Buttons

Refer to " Function Buttons (Related event function)" in the Event Manager.

Stub Dialog Box

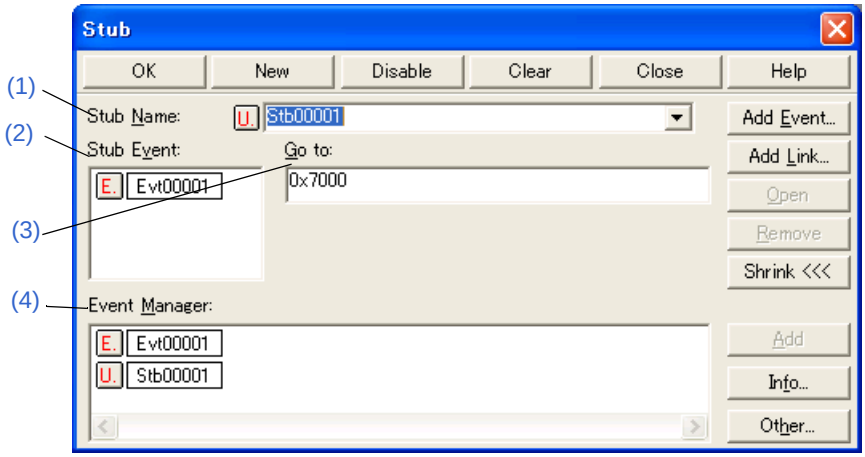
IECUBE

This dialog box is used to register; set, and display stub event conditions. (Refer to "5.12 Event Function", "5.14 Stub Function (When IECUBE Is Connected)".)

Registration and setting of stub event conditions is done by setting each item (256 items max.) in this dialog box and then clicking the <OK> button. The registered stub event conditions are managed by the Event Manager.

There are restrictions on the number of stub event conditions that can be simultaneously set (enabled). (Refer to "5.12.4 Number of enabled events for each event condition".)

Figure 6-60 Stub Dialog Box



- Opening
- Explanation Of Each Area
- Function Buttons (Related event function)

Opening

Select [Event] menu -> [Stub...].

Explanation Of Each Area

(1) Stub Name:

This area is used to set a stub 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 5-19 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) Stub Event:

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

The number of event condition and event link condition that can be set in this area, refer to "Table 6-17 Number of Events Settable".

Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions".

(3) Go to:

This area is used to specify the start address of the function that is executed when a stub event occurs. (Refer to "Table 5-20 Start Address of Function to Be Executed (Stub 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 5-6 Specifying Symbols".)

(4) Event Manager:

This area is used to display the list of the events registered. (Refer to "Table 5-19 Event Icon", "(4) Manipulation in event manager area".)

Function Buttons

Refer to " Function Buttons (Related event function)" in the Event Manager.

Event DMM Dialog Box

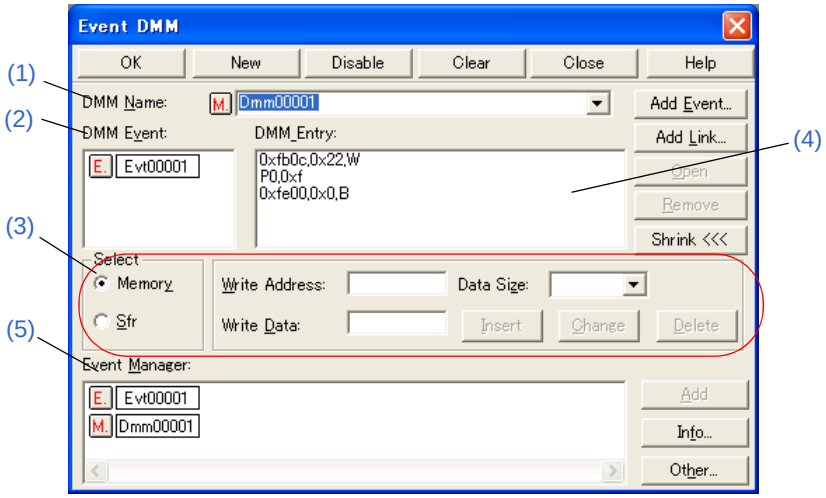
This dialog box is used to register; set, and display event DMM conditions. (Refer to "5.12 Event Function", "5.16.1 Event DMM condition (when IECUBE is connected)".)

Registration and setting of event DMM conditions is done by setting each item (256 items max.) in this dialog box and then clicking the <OK> button. The registered event DMM conditions are managed by the Event Manager.

There are restrictions on the number of event DMM conditions that can be simultaneously set (enabled). (Refer to "5.12.4 Number of enabled events for each event condition".)

Remark: Up to 100 SFR data items and memory data items can be registered per event DMM.

Figure 6-61 Event DMM Dialog Box (When "Memory" Is Selected)



- Opening
- Explanation Of Each Area
- Function Buttons (Related event function)

Opening

Select [Event] menu -> [Event DMM...].

Explanation Of Each Area

(1) DMM Name:

This area is used to set a event DMM 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 5-19 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) DMM Event:

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

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

- Event conditions:18 (execution: 8 , access: 10)
- Event link conditions:2

Event conditions are easily set by dragging the icon of the event to be set from the event manager area and dropping it in this area. For details, refer to "5.12.3 Setting event conditions".

(3) Select

This area is used to register, change, or delete the target data for DMM (event DMM data) after establishment of an event.

The memory or SFR can be registered as event DMM data, and up to 100 event DMM data items can be registered per event DMM.

By selecting a data type, the item to be displayed in the snap data setting area on the right of this area changes in accordance with the selected area.

The event DMM data registered, changed, or deleted in this area is reflected in (4) DMM Entry:. If event DMM data is selected in (4) DMM Entry:, the contents of the selected event DMM data are displayed in this area.

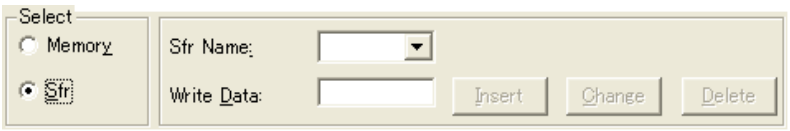
(a) Memory

This area is used to specify the start address, data value to be written, and data size when writing data to the memory after establishment of an event.

Write Address:	Specifies the start address An address can be also specified by a symbol or expression. (Refer to "Table 5-6 Specifying Symbols".)
Write Data:	Specifies the data value
Data Size:	Specifies the data size (Byte, Word)

(b) Sfr

This area is used to specify the SFR name and write data value for writing data to the SFR after event establishment.



Sfr Name:	Specifies the SFR name
Write Data:	Specifies the data value

(c) Buttons to manipulate event DMM data

The following buttons are used to register, change, and delete event DMM data.

Insert	Registers as event DMM data. The registered event DMM data is inserted and displayed at the selected position in (4) DMM Entry:.
Change	Changes the contents of the event DMM data selected in (4) DMM Entry: area to the contents of the event DMM data specified in this area.
Delete	Deletes the event DMM data selected in (4) DMM Entry:. The DEL key performs the same operation.

(4) DMM Entry:

This area displays a list of registered event DMM data.

If event DMM data is selected in this area, the contents of the selected event DMM data are displayed in each setting area.

Event DMM data is displayed as follows:

Table 6-27 Event DMM Data Display Format

Display Example	Contents
Memory data:	
0xFE20<byVar>, 0xff, B 0xFE22<wVar>, 0xff00ff00, W	Start address <symbol expression> ,Write data, Access size (If a symbol or expression is used for address specification, it is enclosed with "< >".)
SFR data:	
P0, 0xff TMS, 0x0000	SFR name, Write data

(5) Event Manager:

This area is used to display the list of the events registered. (Refer to "Table 5-19 Event Icon", "(4) Manipulation in event manager area".)

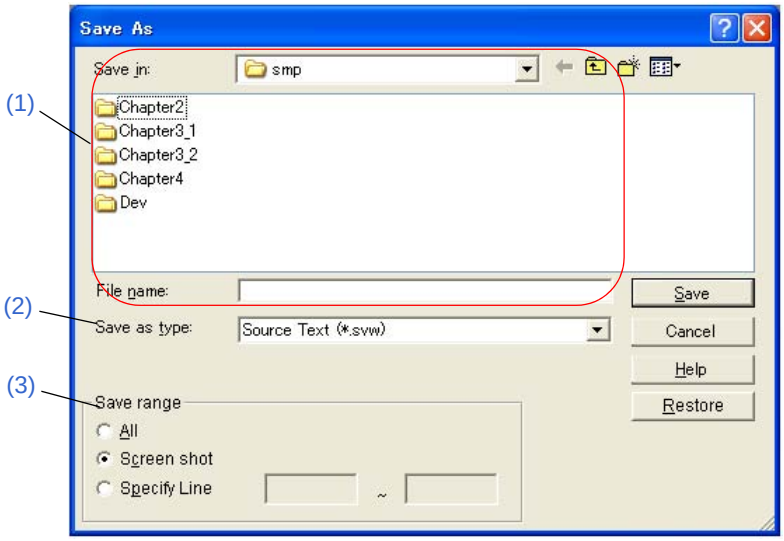
Function Buttons

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 "5.17.2 Window display information (view file)".)

Figure 6-62 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:

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 5-23 Type of the 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
- Code Coverage Window
- Source Window
- Trace View Window

(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 / 100 frames / 256 bytes or more is specified, a message dialog box is displayed to indicate the progress of saving. To stop saving midway, click the <Stop> button in the message dialog box.

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 Source Window is in the mixed display mode, the mixed displayed part on the specified line is also saved.
Specify Frame	Specify the range of trace frames to be saved. (Refer to "Table 6-16 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 5-6 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 "5.17.2 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.

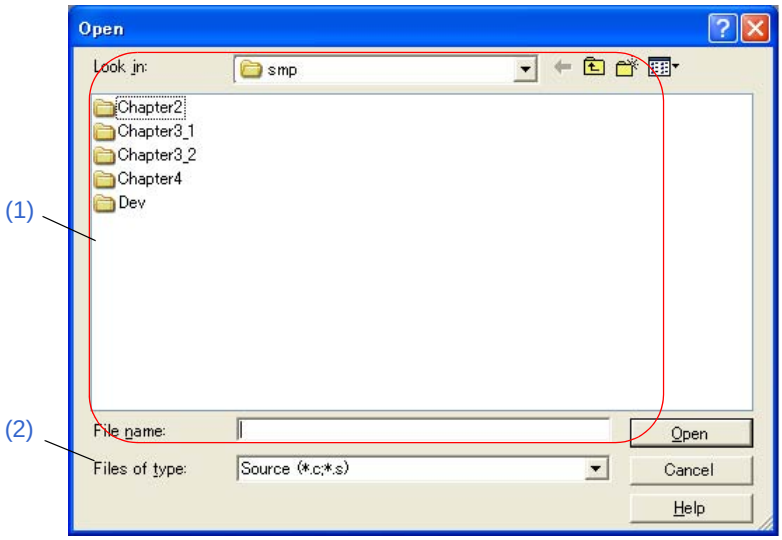
- **Loading source file to which symbol information has been read**

  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 6-63 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:

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 5-23 Type of the View Files".)

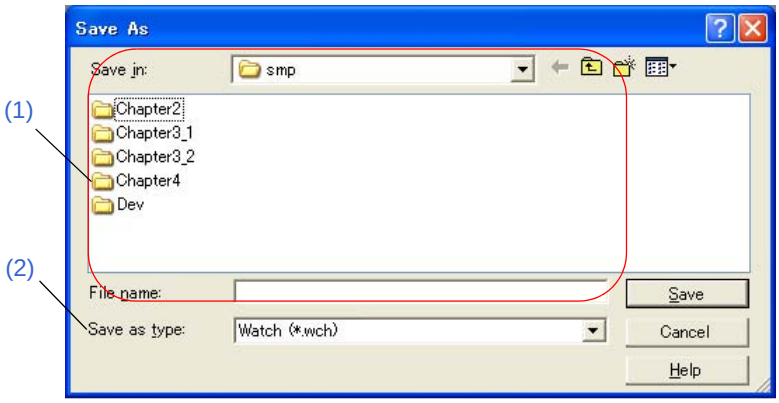
Function Buttons

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 "5.17.3 Window setting information (setting file)".)

Figure 6-64 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:

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 5-24 Type of the Setting Files"). The extension of the file corresponding to the current window is displayed.

Function Buttons

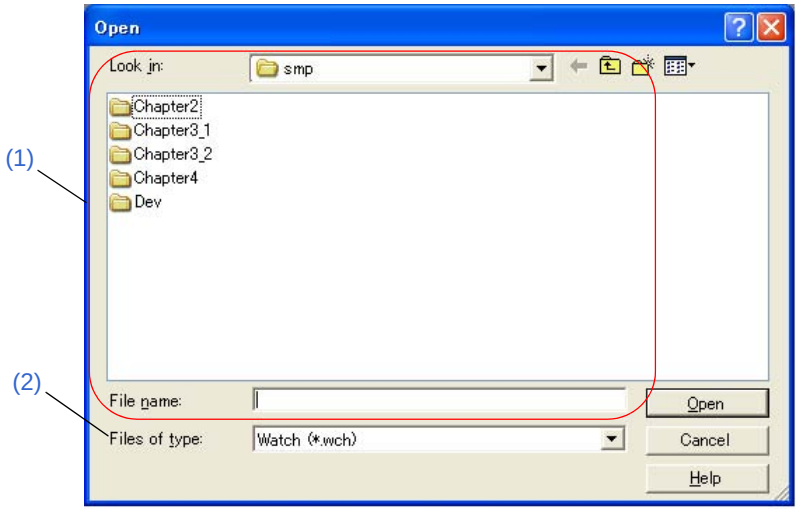
Save	Saves the setting information of the current window to the selected file. After saving, this dialog box is closed.
Cancel	Closes this dialog box without executing anything.
Help	Displays this dialog box online help files.

Environment Setting File Load Dialog Box

This dialog box is used to read the setting files. (Refer to "5.17.3 Window setting information (setting file)".)

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

Figure 6-65 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:

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 5-24 Type of the Setting Files".)

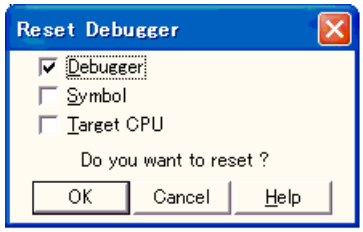
Function Buttons

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 ID78K0-QB, CPU, and symbol information.

Figure 6-66 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 selected item.

Debugger	Initializes the ID78K0-QB (default).
Symbol	Initializes the symbol information.
Target CPU	Initializes the CPU.

Function Buttons

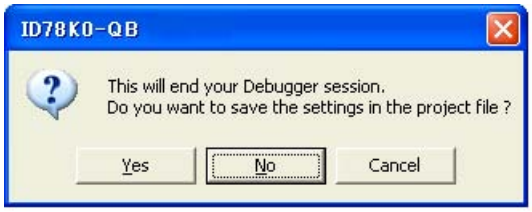
OK	Initializes according to the setting.
Cancel	Cancels the changes and closes this dialog box.
Help	Displays this dialog box online help files.

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 ID78K0-QB. (Refer to "5.17.1 Debugging environment (project file)".)

It can be specified in the Debugger Option Dialog Box that the ID78K0-QB is terminated without this confirmation dialog box being opened.

Figure 6-67 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 ID78K0-QB. If a project file name is not specified, the Project File Save Dialog Box is opened. If the <Cancel> button is selected on the Project File Save Dialog Box, the environment is neither saved to a project file nor is the ID78K0-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 ID78K0-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 ID78K0-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 ID78K0-QB
- Version of device file
- Version of GUI
- Version of debugger DLL
- Version of assembler DLL
- Version of executor
- Version of monitor
- Version of Tcl/Tk
- Product ID and product version of in-circuit emulator
- Version of OCD Control Code (MINICUBE)

Figure 6-68 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

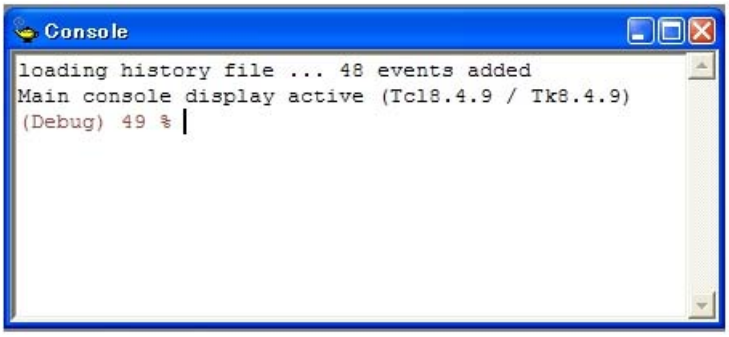
This window is used to input commands that control the ID78K0-QB.

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.

Refer to "CHAPTER 7 COMMAND REFERENCE" for details on the command specifications.

Figure 6-69 Console Window



- Opening
- Explanation Of Each Area

Opening

Select [Browse] menu -> [Console].

Explanation Of Each Area

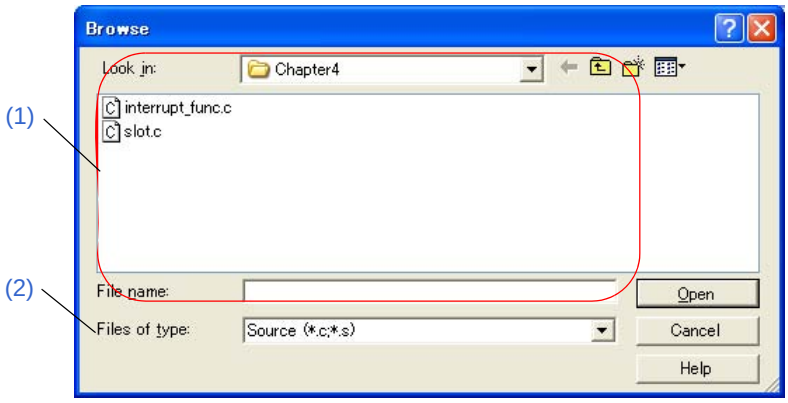
Refer to "CHAPTER 7 COMMAND REFERENCE" for details on the command specifications.

Browse Dialog Box

This dialog box is used to select the file to be set in the Source Text Move Dialog Box.

Remark: If this dialog box is opened for the first time after the system has been started up, the folder first specified by the source path is displayed. When the dialog box is opened the second and subsequent times, the previously displayed folder is recorded and displayed again. If the <Cancel> button is clicked, however, the previously displayed folder is not recorded.

Figure 6-70 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:

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 5-5 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 ID78K0-QB.

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
address	Evaluation of address expression
assemble	Disassemble/line assemble
batch	Executing batch (with echo)
breakpoint	Setting/deletion of breakpoint
dbgexit	Terminating ID78K0-QB
download	Download of files
erase	Deletion of the internal flash memory
extwin	Creation of expansion window
finish	Returning from function
go	Continuous execution
help	Display of help
hook	Setting of hook
inspect	Symbol inspect
jump	Jump to window
map	Setting/deleting memory mapping
mdi	Setting of expansion window
memory	Display/setting of memory
module	Display of the list of files and functions
next	Procedure step
refresh	Redrawing of window
register	Display/setting of register value and SFR value

Command Name	Function
reset	Reset
run	Reset and execution of CPU
step	Step execution
stop	Stop execution
upload	Upload
version	Display of the version information
watch	Display/setting of variables
where	Stack trace
wish	Start of Tclet
xcoverage	Operation of coverage (IECUBE)
xtime	Operation of timer (IECUBE)
xtrace	Operation of tracer (IECUBE)

Table 7-2 List of Console/Tcl Commands

Command Name	Function
alias	Creation of another name
cd	Change of directory
clear	Clears the screen
echo	Echo
exit	Close/end
history	Display of history
ls	Display of files
pwd	Check of the directory
source	Execution of batch
time	Measurement of time for commands
tkcon	Console control
unalias	Deletion of another name
which	Display of the command path or another name
Others	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/idtcl/aliases.tcl".

The aliases are described by default as shown below.

This file can be edited with an editor.

Table 7-3 Contents of File aliases.tcl

```
alias a assemble
alias b breakpoint
alias g go
alias i step -i
alias j jump
alias l download
alias m memory
alias n next
alias r run
alias s step
alias w watch
```

7.4 List Of Variables

Table 7-4 List of Variables

Variable	Function
dcl (chip)	Chip name read only
dcl (prjfile)	Project file name read only
dcl (srcpath)	Source path read only
dcl (ieid)	IE type read only
dcl (iestat)	IE status read only
dcl (bkstat)	Break status read only
env (LANG)	Language
dcl_version	Dcl version read only

7.5 List Of Packages

Table 7-5 List of Packages

Package	Function
tcltest	Restoration test
cwind	Automatic window control
BWidget	Toolkit
tcllib	Tcllibrary
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/idtcl/tools/, an expansion window is added on selecting [Browse] menu -> [Others]. The mdi command, an exclusive command for expansion windows, has been added.

```
# Sample.tcl
wm protocol .dcl WM_DELETE_WINDOW { exit }
mdi geometry 100 50
button .dcl.b -text Push -command exit
pack .dcl.b
```

Caution: In the expansion windows, Tk menu commands cannot be used because of the restrictions of MDI windows.

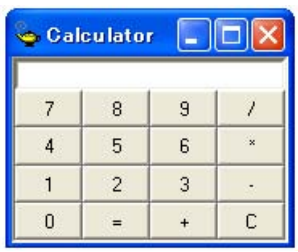
7.7.1 Samples (Calculator Script)

The script of the expansion window in which the calculator script is described and its execution screen are shown below.

Script of expansion window

```
# Calculator.tcl
mdi geometry 100 100
set top .dcl
entry $top.e -relief sunken -textvariable v
frame $top.f -height 120 -width 120; pack $top.e -fill x; pack $top.f -fill both -expand 1
set i 0; set v {}; set r 0.25
foreach n {7 8 9 / 4 5 6 * 1 2 3 - 0 = + C} {
        if {$n == "=" || $n == "C"} {
                button $top.f.b$n -text $n
        } else {
                button $top.f.b$n -text $n -command "$top.e insert end $n"
        }
        place $top.f.b$n -relx [expr ($i%4)*$r] -rely [expr ($i/4)*$r] -relw $r -relh $r
        incr i
}
bind $top.f.bC <1> {$top.e delete 0 end}
bind $top.f.b= <1> {catch {expr $v} v}
```

Figure 7-1 Execution Screen



7.8 Callback Procedure

Expansion windows can hold dcl_asyncproc procedures called by asynchronous messages.

```
proc dcl_asyncproc {mid} {
    if {$mid == 19} {
        redraw
    }
}
```

The asynchronous message ID is passed for the argument of the dcl_asyncproc procedure

The message IDs are shown below:

Table 7-6 Message ID

Message ID	Meaning
9	After changing configuration
10	After registering event
11	After deleting event
12	Before executing
13	After breaking
14	After resetting CPU
15	After resetting ID78K0-QB
17	After changing extended option
18	After changing debugger option
19	After downloading
20	After changing memory or register
36	Before starting tracer (IECUBE)
37	After stopping tracer (IECUBE)
40	Before starting timer (IECUBE)
41	After stopping timer (IECUBE)
42	After clearing trace (IECUBE)
45	After resetting symbol

7.9 Hook Procedure

A hook can be set in the ID78K0-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 ID78K0-QB is closed.

(1) [When hook is set with ID78K0-QB control command]

1) Create script file a. with an editor.
2) Start up the ID78K0-QB, select [Browse] menu -> [Console], and open the Console Window.
3) If the script file is executed in the window as below, the hook in the script file is set.

   %hook test.tcl

(2) [When hook is set on downloading of project file]

1) Create script file a. with an editor.[Note]
2) Start up the ID78K0-QBand 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 folder.

7.10 Related Files

Table 7-7 List of Related Files

File Name	Function
aliases.tcl	Executes when the aliases.tcl console is opened. Sets the default alias etc.
Projectfilename.tcl	Executes when the project file name.tcl project is opened. The following hooks can be used. BeforeDownload AfterDownload AfterCpuReset BeforeCpuRun AfterCpuStop
*Loadmodulefilename.*tcl	Executes when the load module name.tcl load module file is downloaded. The following hooks can be used. BeforeDownload AfterDownload AfterCpuReset BeforeCpuRun AfterCpuStop

7.11 Cautions

- The separator for file and path is a slash (/).
- When a console is open, error messages are output to the console.
- To terminate the command forcibly, close the console.
- The execution of external commands (DOS commands) is OFF by default.

7.12 Explanation Of Commands

In this section, each command is explained using the format shown below.

Command name

Describes the command name.

Input format

Describes the input format of the command.

In the following explanation, italics indicate an Argument to be supplied by the user, while the argument enclosed in "?" may be omitted.

When a command name and an option are entered to the point of identifiability, they are recognized.

Functions

Explains the functions of the command.

Usage example

Shows an example of the usage of the command.

address

address - Evaluation of address expression

Input format

address *expression*

Functions

Converts the address expression specified by *expression* into address.

Usage example

```
(IDCON) 1 % address main
0xaa
(IDCON) 2 % address main+1
0xab
```

assemble

assemble - Disassemble/line assemble

Input format

assemble *?options? address ?code?*

Functions

Line 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 disassembled from the *address* specified by address.

The following are *options*: They are ignored for line assembly.

-code	Command code is also displayed.It is ignored for line assembly.
-number *number*	*Number* line is displayed. It is ignored for line assembly.

Usage example

```
(IDCON) 1 % assemble -n 5 main
0x000002ad PUSH  HL
0x000002ae PUSH  AX
0x000002af PUSH  AX
0x000002b0 MOVW  AX, SP
0x000002b2 MOVW  HL, AX
(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 all
```

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*:

-software	A software break is specified.
-hardware	A hardware break is specified (default).
-execute	The *address* execution break is set (default).
-beforeexecute	The break before *address* execution is set.
-read	An *address* data read break is set.
-write	An *address* data write break is set.
-access	An *address* data access break is set.
-size *size*	The access size is set (8, 16bit). (IECUBE)
-data *value*	The data condition is set.
-datamask *value*	The data mask is set.
-pass *value*	The path count is set. (IECUBE)
-information	The list of breakpoints is displayed.
-delete	The breakpoint whose number is specified is deleted.
-disable	The breakpoint whose number is specified is disabled.
-enable	The breakpoint whose number is specified is enabled.

Usage example

```
(IDCON) 1 % breakpoint main
1
(IDCON) 2 % breakpoint -i
1 Brk00001 enable rammon.c#17

(IDCON) 3 % breakpoint -software sub
2
(IDCON) 4 % breakpoint -i
1 Brk00001 enable rammon.c#17
2 Brk00001 enable rammon.c#8

(IDCON) 5 % breakpoint -disable 2
(IDCON) 6 % breakpoint -i
1 Brk00001 enable rammon.c#17
2 Brk00001 disable rammon.c#8

(IDCON) 7 % breakpoint -delete 1
2 Brk00001 disable rammon.c#8
```

dbgexit

dbgexit - Terminating ID78K0-QB

Input format

dbgexit *?options?*

Functions

Terminate the ID78K0-QB.

The following are *options*:

-saveprj	Project is saved on terminating ID78K0-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*.

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*).

-binary	Binary format data is downloaded.
-coverage	Coverage data is downloaded.
-nosymbol	Download is executed. Symbol information is not read.
-symbolonly	Symbol information is read.
-erase	The contents of the internal flash memory are erased all before download (only a product with internal flash memory).
-reset	CPU is reset after download.
-information	Download information is displayed.
-64kb	A file is downloaded in the format for the memory capacity of 64 KB or lower. Specify this option for loading a binary or HEX-format file.
-bank	A file is downloaded in the format for the memory bank. Specify this option for loading a binary or HEX-format file.

Usage example

(IDCON) 1 % download test.lmf

erase

When MINICUBE is connected

erase - Deletion 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*:

-ignorebreak	Breakpoint is ignored.
-waitbreak	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 ID78K0-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*:

-nocase	The case is distinguished.
-address	Displays in pair with symbol address.

Usage example

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

jump

jump - Jump to window

Input format

jump -source -line *filename ?line?*

jump *?options? address*

Functions

Displays the window specified by *options*.

-source	The Source Window is displayed from the address specified by *address*.
-assemble	The Assemble Window is displayed from the address specified by *address*.
-memory	The Memory Window is displayed from the address specified by *address*.
-line	The command is moved to the line specified by *line*.
-focus	The Focus is moved to the window displayed.

Usage example

(IDCON) 1 % jump -s main
(IDCON) 2 % jump -s -l mainfile.c 10
(IDCON) 3 % jump -m array

map

map - Setting/deletion of memory mapping

Input format

map *options address1 address2 ?accsize?*

Functions

Sets, deletes, and displays memory mapping.

The access size of 8, 16, or 32 is specified by accsize (unit:byte, the default is 8).

The following are *options*:

-target	Target area is mapped.
-**stack**	Stack area is mapped. (IECUBE)
-protect	I/O protect area is mapped.
-rrm	Start address of RRM area is set. When MINICUBE is connected, RRM area can be divided. In this case, 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 to 16 bytes, and the total size is up to 16.
-clear	All the settings for the mapping are deleted.
-information	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 {IRAM}
7: 0xfee0 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*:

-byte	Displayed/set in one-byte units (default).
-word	Displayed/set in word units.
-fill	The data is filled in.
-copy	The data is copied.
-noverify	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 (MINICUBE)
- Referencing a memory area in the RRM area (MINICUBE)
- Setting of a memory area other than the internal ROM/RAM area (IECUBE)
- Setting of a memory area (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 *progname ?filename?*

Functions

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

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*:

-source	The command is executed in source line units (default).
-instruction	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*:

-force	Compulsory reading or writing is executed.
-bankno *bankno*	Specifies the bank number.

Remark: If either of the following operations is performed during user program execution, the CPU is stopped momentarily and execution continues.(when IECUBE is connected)

- Setting a register or SFR
- Referencing a control register

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 ID78K0-QB , CPU, symbols or events.

If options are omitted, the CPU is reset.

The following are *options*:

-cpu	CPU is reset (default).
-debugger	The ID78K0-QB is reset.
-symbol	Symbol is reset.
-event	All events and software breaks are reset.

Usage example

(IDCON) 1 % reset

run

run - Reset and execution of CPU

Input format

run *?options?*

Functions

Resets the program and executes it.

If -waitbreak is not specified, the command does not wait until the program stops.

The following are *options*:

-**waitbreak**	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*:

-source	The command is executed in source line units (default).
-instruction	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*

upload -coverage *filename*

Functions

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

When saving coverage data, all the specified range of the coverage data is saved in the file (specification of start/end addresses not required).

Remark: Data is saved in the following format when specifying addresses with any one of the options -binary, -intel, -motorola, or -tektronix is specified with a device incorporating the memory bank.
When an address lower than 0x10000 is specified as the end address:
Saved in the format for memory capacity of 64 KB or lower
When a 0x10000 or higher address is specified as the end address:
Saved in the format for the memory bank

The following are *options*:

-binary	The data is saved in binary format.
-coverage	The coverage data is saved.
-intel	The data is saved in Intel HEX format (default).
-motorola	The data is saved in Motorola HEX format.
-tektronix	The data is saved in Tektronix HEX format.
-force	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 ID78K0-QB.

Usage example

```
(IDCON) 1 % version
GUI             : VX.XX [XX XXXX 200X]
Devicefile      : 78K/0[uPD78FXXXX] VX.XX
Debugger        : 78K0-QB Debugger VX.XX XX XXXX 200X]
Executer        : 78K0 MINICUBE Executer VX.XX
Micro program : OCD Control Code VX.XX
Assembler       : 78K0 Asm/Disasm VX.XX 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*:

-binary	The value is displayed in binary digits.
-octal	The value is displayed in octal digits.
-decimal	The value is displayed in decimal digits.
-hexdecimal	The value is displayed in hexdecimal digits.
-string	The value is displayed in character strings.
-sizeof	The size, instead of the value, of variables is displayed in decimal digits.
-encoding *name*	Encoding during character string display is specified. By default, system encoding is used. *name* (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

xcoverage

When IECUBE is connected

xcoverage - Operation of coverage

Input format

xcoverage *option*

Functions

Operates coverage.

The following are *options*:

-clear	Clears the coverage memory.

Usage example

(IDCON) 1 % xcoverage -clear

xtime

xtime - Operation of timer

Input format

xtime *option*

Functions

Operates timer.

The following are *options:*

-start	Timer starts on executing the program. (when IECUBE is connected)
-stop	Timer stops on executing the program. (when IECUBE is connected)
-gobreak	Time from Go to Break is displayed in nsec.

Usage example

(IDCON) 1 % xtime -start
(IDCON) 2 % xtime -stop

xtrace

When IECUBE is connected

xtrace - Operation of tracer

Input format

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

xtrace -start

xtrace -stop

xtrace -clear

xtrace -mode *?mode?*

Functions

Operates tracer.

The following are *options:*

-start	The tracer starts on executing the program.
-stop	The tracer stops on executing the program.
-clear	Clears the trace memory.
-dump	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.
-append	The dump result is added to a file.
-mode *?mode?*	The trace control mode (any one of: all, cond, nonstop, fullstop, fullbreak, delaystop, delaybreak) is selected. When *mode* is omitted, the current mode is displayed.

Usage example

```
(IDCON) 1 % xtrace -start
(IDCON) 2 % xtrace -stop
(IDCON) 3 % xtrace -dump 3
_ A 000015 0023B 9D M1          BNC $_sub_Global+0x2b
_ A 000016 0023C 1F OP
_               Software Break
(IDCON) 4 % xtrace -clear
(IDCON) 5 % xtrace -addup true
```

tkcon

tkcon - Console control

Input format

tkcon cmd ?arg?

Functions

Controls the Console window.

This command is one of the console/Tcl commands.(Refer to "Table 7-2 List of Console/Tcl Commands".)

tkcon buffer ?size?	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.
tkcon close **tkcon destroy**	Close the Console window.
tkcon font ?fontname?	Sets and references the fonts used in the Console window.
tkcon gets	Performs standard inputs such as Stdin. Opens a dialog box.
tkcon history ?-newline?	Displays the command history.
tkcon save ?filename? ?type	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.
tkcon version	Displays the console version.

Usage example

(IDCON) 1 % tkcon save c:/temp/logfile.txt all

APPENDIX A EXPANSION WINDOW

A.1 Overview

With the ID78K0-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 ID78K0-QB. Users can create windows using this Tcl.

The ID78K0-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
List Window	Displays a list of the source files and functions.
Grep Window	Searches a character string.
Hook Window	Sets the hook procedure.
SymInspect Window	Searches through a list of properly described symbols.

A.3 Activation

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

Remark: Each .tcl file is installed in NECTools32\ID78K0-QB\bin\idtcl\tools

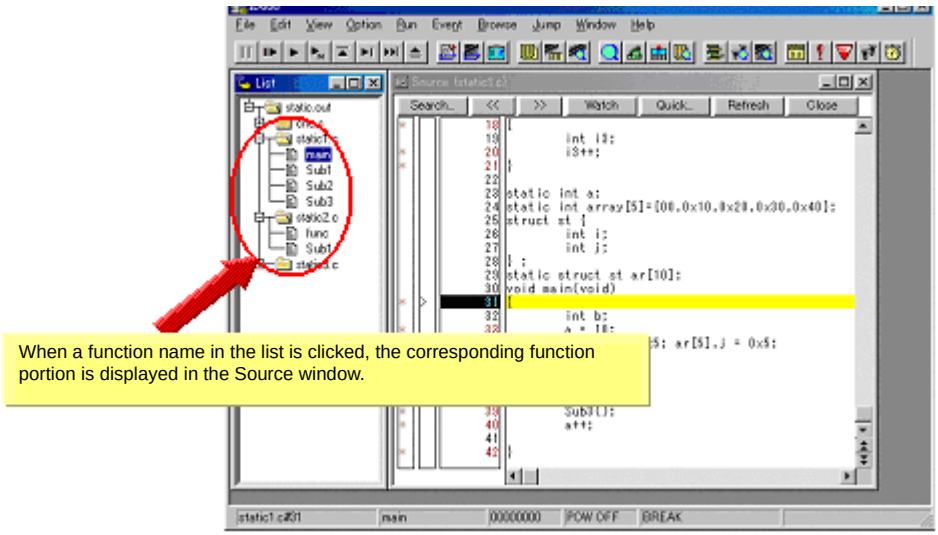
A.4 Explanation Of Each Sample Window

The ID78K0-QB provides the sample window below.

List Window

The lists of the source files and functions are displayed in a tree format in this window. When a function name in the list is clicked, the corresponding source is displayed.

Figure A-1 List Window

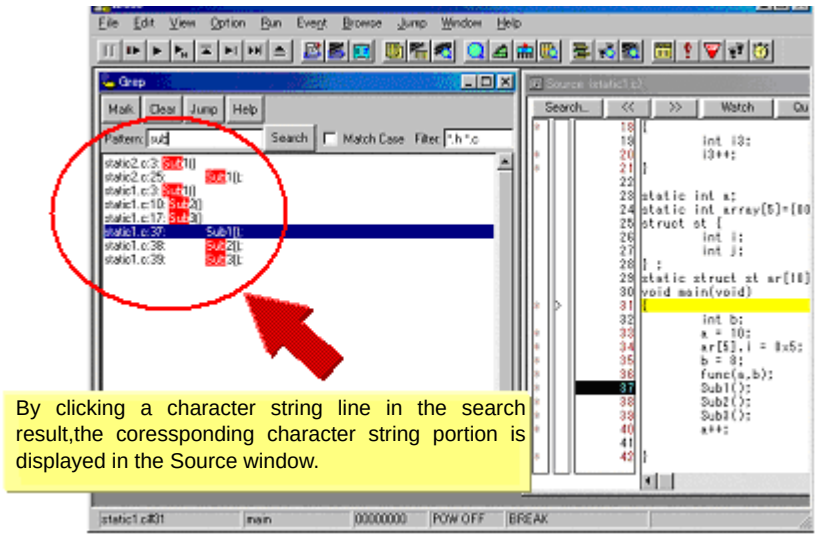


Grep Window

Search for a character string is performed in the files under the source path.

When the search result is clicked, the corresponding source is displayed.

Figure A-2 Grep Window



Object	Function
Pattern	Input the character string to be searched.
<Mark> button	Marks the searched character string.
<Clear> button	Clears the marking.
<Jump> button	Put the cursor on a section in the search result and click this button to open the corresponding file.
Match Case	Select whether or not to distinguish uppercase and lowercase.
Filter	Specify the type of the file to be searched.

Hook Window

This window is used to set a hook to the debugger, using a hook procedure.

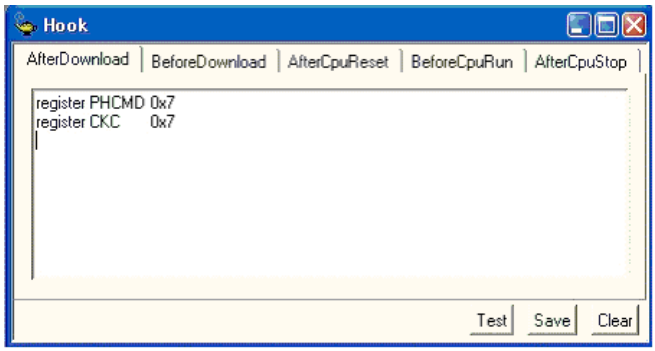
The hook procedure enables changing the register value before downloading a program, or after a CPU reset.

On this window, a hook can be set by using the following five tabs.

- [AfterDownload] tab: Hook after downloading
- [BeforeDownload] tab: Hook before downloading
- [AfterCpuReset] tab: Hook after CPU reset during break
- [BeforeCpuRun] tab: Hook before start of execution
- [AfterCpuStop] tab: Hook after break

If the setting is saved as "project-file-name.tcl" in the folder where the project is stored, the setting is executed when the project is next opened. Specify the general-purpose register and the SFR for the register name.

Figure A-3 Hook Window

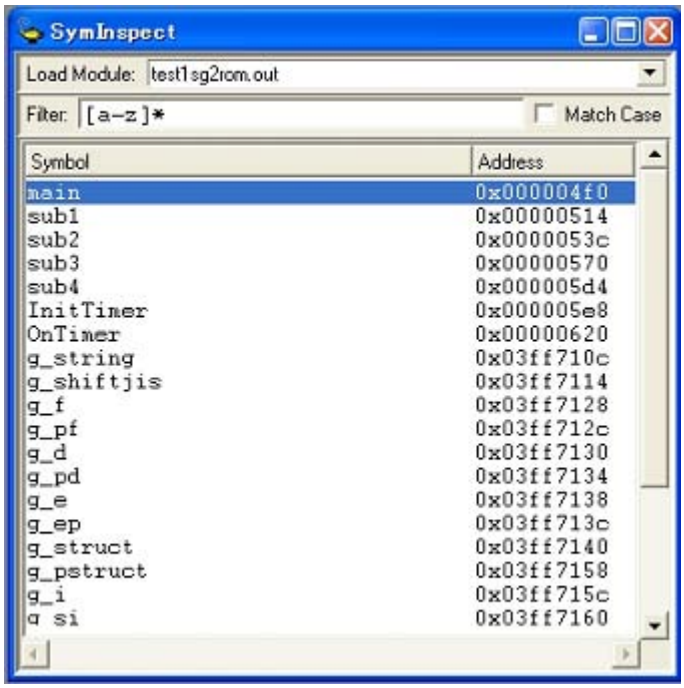


Object	Function
[AfterDownload]tab	Hook after downloading. After downloading is performed, the register values input to the tab are automatically overwritten by the specified value.
[BeforeDownload]tab	Hook before downloading. Before downloading is performed, the register values input to the tab are 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.
<Test> button	All the commands described on the tabs are tested.
<Save> button	Saves all the tab contents to a file. If the ID78K0-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.

SymInspect Window

This window displays the list of the symbols and addresses of loaded module files, and is used for searching the list for the properly described symbol.

Figure A-4 Sym Inspect Window



Object	Function
Load Module:	Selects a load module file.
Filter:	Specifies a properly described symbol so that the symbol is retrieved.
Match Case	In Filter:, specify to differentiate or not differentiate case sensitivity. Check this box to differentiate case sensitivity.
Symbol	Displays the symbols. Clicking this icon has the symbols sorted in alphabetical order.
Address	Displays the addresses. Clicking this icon has the addresses sorted in ascending numerical order.

Context Menu	Function
Copy	Copies the selected address to the clipboard.
Jump to Source	Jumps from the address in the selected line to the identical address displayed in the Source Window.
Jump to Assemble	Jumps from the address in the selected line to the identical address displayed in the Assemble Window.
Jump to Memory	Jumps from the address in the selected line to the identical address displayed in the Memory Window.

APPENDIX B INPUT CONVENTIONS

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
.	Period	Direct member operator or bit position specifier
,	Comma	Delimiter between operands

Character	Name	Usage
[	Left bracket	Array subscript operator or indirect display symbol
]	Right bracket	
!	Exclamation	Absolute addressing start symbol
$	Dollar	Relative addressing start symbol
#	Sharp mark	Symbol indicating an immediate value

B.2 Symbols

(1) A symbol is composed of either of the following characters.

A to Z, a to z, @, _ (underbar), . (period) and 0 to 9

(2) A symbol must start with a character other than numerals 0 to 9.

(3) Uppercase characters (A to Z) and lowercase characters (a to z) are distinguished.

(4) A symbol must be no more than 2048 characters long (if a symbol of more than 2048 characters is defined, only the first 2048 characters are valid).

(5) A symbol is defined by loading a load module file.

(6) Symbols are classified into the following types by the valid range:

Global symbol (assembly language, structured assembly language, C language)

Static symbol (C language)
- In-file static symbol
- In-function static symbol

Local symbol (C language)
- In-module local symbol (assembly language, structured assembly language)]
- In-file local symbol
- In-function local symbol
- In-block local symbo

(7) The following symbols are available for each language used:

Assembly language, structured assembly language
label name, constant name, bit symbol name

C language
Variable name (including pointer variable name, enumeration type variable name, array name, structure name, and union name)
Function name, label name
Array element, structure element, union element, bit field (if the symbol is an array, structure, or union)

(8) A symbol can be described instead of an address or numeric value.

(9) The valid range of a symbol is determined based on the source debug information when the source file is assembled or compiled.

(10) Describe only the symbol name of a global symbol.

(11) A local symbol is expressed in pairs with a file name.

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 ID78K0-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**Y** n...n**Y** (n=0,1)
Octal number	n**O** n...n**O** (n=0,1,2,3,4,5,6,7) n**Q** n...n**Q** (n=0,1,2,3,4,5,6,7)
Decimal number	n n...n n**T** n...n**T** (n=0,1,2,3,4,5,6,7,8,9)
Hexadecimal numbers	n n...n n**H** n...n**H** **0x**n **0x**n...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
Arithmetic operator		
+	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.
Logical operator		
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.
Shift operator		
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.
Byte separation operator		
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.
Word separation operator		
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.

Symbol	Meaning	Explanation
Other		
(	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	0**Y** <= value <= 11111111111111111111111111111111**Y** (32 digits)
Octal number	0**O** <= value <= 37777777777**O**
Decimal number	-2147483648 <= value <= 4294967295 (A negative decimal number is internally converted into a 2's complement.)
Hexadecimal numbers	0**H** <= value <= 0FFFFFFF**H**

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 ".".

File names cannot be prefixed or suffixed by a period (.) or space.

File names are not case-sensitive.

A file names consists of up to 259 characters including the path.

(2) Other file names

Other file names comply with Windows file name regulations.

The following characters cannot be used in file names.

\ / : * ? " < > | ;

File names cannot be prefixed or suffixed by a period (.) or space.

File names are not case-sensitive.

A file name consists of up to 259 characters including the path.

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	- Deletes one character after the cursor and move the character string following the cursor forward. - Deletes a various event condition selected in the Event Manager or each event dialog box. - Deletes the data selected in the Watch Window.
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	- Closes the pull-down menu. - Closes the modal dialog box. - Restores the input data.
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	- Sets the input data. - Presses the default push button.
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 breakpoints 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 plus when the PM plus 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.

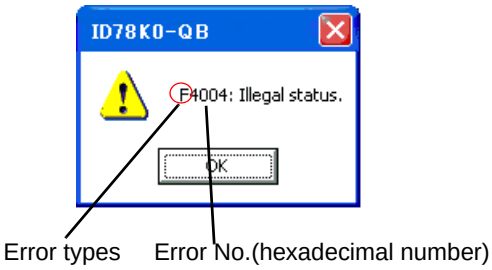
APPENDIX D MESSAGES

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 Messages



D.2 Types Of Messages

The ID78K0-QB outputs the following types of messages.

Table D-1 Types of Messages

Types	Meaning
A*xxxx*	A: Abort Error Stops processing, and terminates the debugger. If this error occurs, debugging cannot be continued.
F*xxxx*	F: Fail Stops processing, and opened windows and dialog boxes are closed.
W*xxxx*	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 *X*0000

F0002:	This feature is not supported.
F0100:	Can not communicate with ICE. Please confirm the installation of the device driver for the PC interface board. 1) The driver may not be correctly installed. Reinstall the driver.
A0101:	Can not find initialization file (expc.ini).
A0102:	Host name not found.
F0103:	Data transfer to ICE is timed out. Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board.
F0104:	Data receive from ICE is timed out. Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board.
A0105:	Failed in reading device file (d1xxx.78k). 1) Necessary files may be damaged. Reinstall the device file.
A0106:	Illegal data received. 1) Check the power of the in-circuit emulator, cable connections, and setting of the interface board and restart the debugger.
A0107:	Can not communicate with ICE.
A0108:	Failed in reading initialization file (expc.ini).
A0109:	Can not communicate with ICE. Please terminate the debugger and check the power of ICE or the connection of cable then restart the debugger. 1) An error may have occurred during USB communication (such as disconnection of power or cable) or IECUBE is faulty (IECUBE).
F010a:	Can not communicate. Please confirm the availability of the communication port.
A01a0:	No response from the emulation CPU. Please confirm the signal of the CLOCK or RESET and so on. 1) Check the HOLD signal, WAIT signal, clock signal, etc. The IOR value (or SFR value) may not be correct.
A01a1:	Failed in reading EX78K4.OM0.
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.

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 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 SFR Window or Hook Procedure 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 breaked.
F0302:	User program is being traced.
F0303:	Not traced.
F0304:	Trace memory is not set.
F0306:	No trace block exists.
F0307:	No event condition exists.
F0308:	No timer measurement is done.
F0309:	No trigger frame exists.
F030a:	Tracer is being stopped.
F030b:	Specified snap-event has not been registered.
F030c:	Specified stub-event has not been registered.
F030d:	Timer is running.
F030e:	Memory copy area is overlapped.
F030f:	Trace has been already set.
F0310:	Event condition is not set.
F0311:	Too many valid timer event conditions.
F0312:	Specified timer event is not set.

F0313: Illegal map range. 1) Check the map range in the Configuration Dialog Box. When mapping to external memory has been performed, change the register values necessary for accessing the external memory using the SFR Window or Hook Procedure before download).
F0314: Only trace delay mode can set with delay trigger.
F0315: Delay trigger cannot set without trace delay mode.
F0316: Overflowed the number of mapping.
F03a0: Target is not turned on. 1) Check the target power supply. Check the cable connecting the in-circuit emulator and target board. Check that the VDD signal is input to the connector of the target board.
F03a1: Step execution is being done.
F03a2: Timer and Tracer are running.
F03a3: Event link and BRS events are mixed.
F03d0: Back-trace is being executed.
F03d1: Back-trace is being stopped.
F03d2: Back-trace execution point overrun oldest frame.
F03d3: Register status or Memory status cannot be set up other than Phase1 of event link.
F03d4: No back-trace information exists.
F03d5: Last command can not be backstepped.
F0400: Illegal condition. 1) Settings of the used in-circuit emulator and those of the Configuration Dialog Box may not match. Check the Chip selection.
F0401: Result of timer measurement overflowed.
F0402: Too many event conditions with path count.
F0403: Too many address range conditions.
F0404: Too many simultaneously-usable-event conditions.
F0405: Too many snap-events.
F0406: Too many stub-events.
F0407: Too many initialization data.
F0408: Too large search data (> 16 byte).
F0409: Too large search data (> search range).
F040a: Too many Linking-event conditions.
F04a0: Software break conditions number overflow.
F04a1: Not enough memory for emulation.
F04a2: Too many partition of bus size.
F04a3: Too many execution-event conditions.
F04a4: Too many bus-event conditions.

A0600: Not enough memory for buffer. 1) There is not enough system memory. Close the applications being executed and the open files.
A0601: Not enough resource of operating system.
F0b20: This event number can not be used.
F0b61: Section Trace event conditions overflow.
F0b66: Cannot use the break before execution event and the software break at the same time. 1) This is because a break before execution is used for implementing a software break (when MINICUBE is connected).
F0b80: Reset by hardware error.
F0c00: Monitor file read error. 1) Necessary files may be damaged. Reinstall the debugger.
A0c01: During access of register, CPU did time out. 1) Check the clock signal, etc. The register value may not be correct.
A0c02: During access of memory, CPU did time out. 1) Check the HOLD signal, WAIT signal, clock signal, etc. The memory value may not be correct.
A0c03: During access of I/O register, CPU did time out. 1) Check the HOLD signal, WAIT signal, clock signal, etc. The I/O register value may not be correct.
F0c20: Guarded area can not be accessed.
F0c21: Memory was unready status.
F0c22: Memory unready status was canceled.
F0c23: Bus hold under continuation. 1) Check the setting of the target board, or mask the HOLD pin.
F0c24: It cannot shift to debug mode. 1) Check the clock signal. This may be caused by a stopped clock or a slow clock.
F0c25: Flash macro service ROM was accessed or stepped in. 1) Please perform [Go] execution or CPU reset.
F0c26: FLMD terminal is in a write-protected state. 1) FLMD is not in the write-enabled status. Check the status of the FLMD0 and FLMD1 pins.
F0c27: Security flag is in a write-protected state. 1) The security flag of the flash memory has disabled writing, block erasure, or chip erasure. Nothing can be written to the flash memory.
F0c28: Internal RAM is not enough, the writing to flash memory is not made. 1) The internal RAM size is less than 4 KB and flash self-programming cannot be executed.
F0c29: The blank check of flash memory failed.
F0c2a: The erasing of flash memory failed.
F0c2b: The writing of flash memory failed.
F0c2c: The internal verification of flash memory failed.

F0c2d: Failed in writing flash memory.
F0c2e: There is no response from flash macro service.
F0c2f : Response from flash macro service is not right.
F0c30: Flash I/O register operation prohibition setup needs to be canceled.
F0c31: STOP mode under continuation. Can not compulsory break. Please release STOP mode or reset the CPU.
F0c32: Please write in flash memory in the single chip mode 0.
F0c33: Disabling the on-chip debug function is prohibited.
F0c34: Writing to the on-chip debug reserved area is prohibited.
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.
F0c39: Real-time RAM monitoring failed.
F0c40: Status of effective event conditions cannot be changed.
F0c41: Coverage test is being executed.
F0c42: Monitor has failed in shift in the debugging mode.Please reset the CPU.
F0c43: Can not communicate with ICE. Please confirm the power of ICE, connection of the interface cable. 1) Check the power to the in-circuit emulator and cable connections The switch setting may be wrong if a desktop computer is used and two or more PC cards are inserted. Check the setting. Or it may have malfunctioned (when IECUBE connected).
F0c44: Coverage test is being executed.
F0c45: Inside of Power off reset emulation cannot carry out program execution.
F0c46: This function is not valid during Flash Self Emulation.
F0c60: Event before execution cannot be set up other than break conditions.
F0c61: Can not register event numbers which can not be used for hardware break.
F0c62: Event numbers reserved for hardware breaks can not be used.
F0c63: Event link conditions cannot set.
F0c64: Too many ROM-emulation-RAM areas.
F0c67: Writing of flash memory during block is not made.
F0c70: DCU cannot be accessed. 1) IE may be malfunctioning (when the IECUBE is connected).
F0c71: Reset cannot be performed. 1) Check the clock signal. This may be caused by a stopped clock or a slow clock.
F0c72: Monitor memory cannot be accessed. 1) Revise the Main OSC value in the Configuration Dialog Box. If this does not solve the problem, IE may be malfunctioning (when the IECUBE is connected).
F0c73: Monitor execution cannot be performed. 1) IE may be malfunctioning (when the IECUBE is connected).

F0c74:	CPU register cannot be accessed. 1) The device file selection may be incorrect. Select a device file that supports the target chip in Chip Selection in the Configuration Dialog Box. If this does not solve the problem, the IE may be malfunctioning (when the IECUBE is connected).
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. 1) The device file selection may be incorrect. Select a device file that supports the target chip in Chip Selection in the Configuration Dialog Box. If this does not solve the problem, the IE may be malfunctioning (when the IECUBE is connected).
F0c77:	DCU access is unusuall. 1) IE may be malfunctioning (when the IECUBE is connected).
F0c78:	Failed in reading of trace data.
F0ca0:	Can not communicate with ICE. Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board. Error occurred inside debugger. (IECUBE) 1) Can not communicate with in-circuit emulator. Check the power of the in-circuit emulator, cable connections, and setting of the interface board etc.
F0ca1:	Monitor file not found. 1) Necessary files may be damaged. Reinstall the debugger.
F0ca2:	This device file does not include the on-chip debug information. 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 MINICUBE is connected). 2) IE may be malfunctioning (when the IECUBE is connected).
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 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 *X*1000

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 *X*2000

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 *X*3000

F3000: No mapped address was accessed. 1) The allocation addresses of the program and the addresses of the debugger may not match. Set the mapping to the external memory in the Configuration Dialog Box 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 SFR Window or Hook Procedure before download.
F3001: Memory has different value.
F3002: Illegal start address.
F3003: Illegal end address
F3004: Illegal start address and end address.
F3005: Illegal condition.
F3006: User program is running.
F3007: Verification error.
F3008: No condition specified.
F3009: Parameter size does not align with access size alignment.
F300a: Specified address does not align with access size alignment.
F300b: Source address does not align with access size alignment.
F300c: Destination address does not align with access size alignment.
F300d: Illegal end address.
F300e: Different access size in specified area.
F300f: Different access size both in source and destination areas.
F3010: Different access size in destination area.
F3011: Different access size, source & destination.
A3012: Not enough memory. 1) There is not enough system memory. Close the applications being executed and the open files.
F3013: Failed in writing DMM.
F3014: Oveflowed mapping area.
F3015: Processing was interrupted.

F3016: This feature is not supported.
A30ff: Can not communicate with ICE.

(5) From *X*4000

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 (>= 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 *X*5000

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 EX78K4.OM0.
F500a: Specified device file is illegal version. 1) Necessary files may be damaged. Reinstall the device file.
W500b: Specified device file does not relocate IRAM.
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 *X*6000

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 *X*7000

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 *X*8000

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 *X*9000

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 *X*a000

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 *X*b000

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 Debugger Option Dialog Box, or check if the source is in the same directory as the out file. Refer to the Assemble Window 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 *X*c000

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 Configuration Dialog Box 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 *X*d000

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 *X*e000

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 *X*f000

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 SFR Window or Hook Procedure before download .
Wf203:	When a program is running, while rewriting a memory, 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:	This variable cannot be set as a break. 1) Break cannot be set for the following variables. - Local variables, static variables - Array variables, member variables of structures/unions - Register/SFR - Variable expressions
Wf600:	Save project file?
Wf601:	When connecting the target system, please turn on the target system. 1) When a target is not connected, simply click the <OK> button.
Wf602:	Please change a MODE mask condition or connect the target system.
Ff603:	Incorrect ID Code. 1) This may be caused by the following (when the MINICUBE is connected). - The ID code is incorrect. -> Input the correct ID code. - The internal flash memory is in the write mode because the FLMD0 pin is high. -> Make the FLMD0 pin low. - The emulator connection prohibition mode is set because the ID code (bit 0,1 of address 0x84) is 0. ->Erase the internal flash memory once by flash writer.
Af604:	Incorrect ID Code. Abort the debugger. 1) This may be caused by the following (when the MINICUBE is connected). - The ID code is incorrect. -> Input the correct ID code. - The internal flash memory is in the write mode because the FLMD0 pin is high. -> Make the FLMD0 pin low. - The emulator connection prohibition mode is set because the ID code (bit 0,1 of address 0x84) is 0. ->Erase the internal flash memory once by flash writer.
Ff605:	Please check connection with the target board. 1) Check the connection of the target connector (TC). If a target is not connected, review the Target setting in the Configuration Dialog Box.
Ff606:	Please check connection with the target board, and power on it. 1) Check the target power supply. If a target is not connected, review the Target setting in the Configuration Dialog Box.
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 Configuration Dialog Box 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).
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 (when MINICUBE is connected).
Af60c:	During break Target was not turned on.
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 Add I/O Port Dialog Box.
Ff902:	Illegal access size.
Ff903:	Illegal access type.
Ff904:	There is the same name.
Wf905:	[XXX] is already exist. Do you replace it?
Wf906:	Would you like to register the change made in [XXX]?
Ffa00:	The [XXX] function of current program on PC position not found. 1) The symbol specified in main() label: in the Debugger Option Dialog Box could be found. Set a symbol of the main routine of the program. Default is _main.

Ffa01:	The line information on PC position not found. 1) The source file corresponding to program counter (PC) value when the program was stopped could not be found. The following reasons are possible. -The source file exists in a location that the source path does not connect to. -The program stopped where the source files, such as library or RX, do not exist. -The program looped, jumped to an address that is not used by the program, and stopped there.
Wfb00:	User program is running. Do you want to stop user program? 1) <Yes> button is selected, execution of the user program is stopped and then the Exit Debugger Dialog Box is displayed. If it is specified in the Debugger Option Dialog Box that the Exit Debugger dialog box is not to be displayed, however, the ID78K0-QB is terminated. <No> button is selected, execution of the user program is not stopped and the Exit Debugger Dialog Box is not displayed. The ID78K0-QB is not terminated.
Wfb01:	Since bit 7 of address 0x79 in the ID code are 0, The N-Wire emulator becomes prohibition of use henceforth. Do you exit the debugger as it is?
Ffc00:	Online help window cannot be started. Please install HTML Help environment with reference to a users manual.
Ffd00:	Failed to specify [XXX].
Ffe00:	The maximum size of RRM was exceeded.
Wfe01:	There is a duplicate RRM address.
Wfe0b:	It shift to the flash mode. Is it completely cleared but is the present event. Doesn't it care?
Fffff:	Interrupted.

APPENDIX E INDEX

V

W

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
Podbielskistrasse 166 B
30177 Hannover
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 Esplandiu, 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
Steijgerweg 6
5616 HS 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: 02-8175-9600
http://www.tw.necel.com/

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