User’s Manual

NEC

ID850QB Ver. 2.80

Integrated Debugger

Operation

Target Device
V850 Series

Document No. U16973EJ1V0UM00 (1st edition)
Date Published February 2004 CP(K)

Printed in Japan

[MEMO]

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

These commodities, technology or software, must be exported in accordance
with the export administration regulations of the exporting country.
Diversion contrary to the law of that country is prohibited.

- **The information in this document is current as of January, 2004. 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

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC Electronics product in your application, please contact the NEC Electronics office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

[GLOBAL SUPPORT]
http://www.necel.com/en/support/support.html

NEC Electronics America, Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65030

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00
- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41
- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45
- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 80 820
- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-558-3737

NEC Electronics Shanghai Ltd.
Shanghai, P.R. China
Tel: 021-5888-5400

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 6253-8311

J04.1

[MEMO]

PREFACE

Target Readers

This manual is intended for user engineers who design and develop application systems of the V850 Series.

Purpose

This manual is intended for users to understand the functions of the ID850QB in the organization below.

Organization

This manual consists of the following chapters:

- OVERVIEW
- INSTALLATION
- STARTING AND TERMINATING
- ASSOCIATION WITH PM plus
- DEBAG FUNCTIONS
- 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 V850 Series
→ Refer to Hardware User's Manual for each product..

To understand the instruction functions of the V850 Series
→ Refer to **V850ES Architecture User's Manual (U15943E)** or **V850E1 Architecture User's Manual (U14559E)**.

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

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.
IE-703002-MC (In-circuit emulator for V853, V850/SA1, V850/SB1, V850/SB2, V850/SC1, V850/SC2, V850/SC3, V850/SF1, V850/SV1)		U11595E
IE-V850E-MC (In-circuit emulator for V850E/IA1, V850E/IA2), IE-V850E-MC-A (In-circuit emulator for V850E/MA1, V850E/MA2)		U14487E
IE-V850E1-CD-NW(PCNCIA Card Type On-Chip Debug Emulator)		U16647E
IE-703003-MC-EM1 (In-circuit emulator option board for V853)		U11596E
IE-703017-MC-EM1 (In-circuit emulator option board for V850/SA1)		U12898E
IE-703037-MC-EM1 (In-circuit emulator option board for V850/SB1, V850/SB2)		U14151E
IE-703040-MC-EM1 (In-circuit emulator option board for V850/SV1)		U14337E
IE-703079-MC-EM1 (In-circuit emulator option board for V850/SF1)		U15447E
IE-703102-MC (In-circuit emulator for V850E/MS1, V850E/MS2)		U13875E
IE-703102-MC-EM1, IE-703102-MC-EM1-A (In-circuit emulator option board for V850E/MS1, V850E/MS2)		U13876E
IE-703107-MC-EM1 (In-circuit emulator option board for V850E/MA1)		U14481E
IE-703116-MC-EM1 (In-circuit emulator option board for V850E/IA1)		U14700E
CA850 Ver. 2.50 C Compiler Package	Operation	U16053E
	C Language	U16054E
	PM plus	U16055E
	Assembly Language	U16042E
ID850 Ver. 2.50 Integrated Debugger	Operation	U16217E
ID850NW Ver. 2.51 Integrated Debugger	Operation	U16454E
ID850NWC Ver. 2.51 Integrated Debugger	Operation	U16525E
ID850QB Ver. 2.80 Integrated Debugger	Operation	This manual
SM850 Ver. 2.50 System Simulator	Operation	U16218E
SM850 Ver. 2.00 or Later System Simulator	External Part User Open Interface Specifications	U14873E
RX850 Ver. 3.13 or Later Real-Time OS	Basics	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro Ver. 3.13 Real-Time OS	Fundamental	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 Ver. 3.01 Task Debugger		U13737E
RD850 Pro Ver. 3.01 Task Debugger		U13916E
AZ850 Ver. 3.10 System Performance Analyzer		U14410E
PG-FP4 Flash Memory Programmer		U15260E

CONTENTS

LIST OF FIGURES

Figure No Title, Page

LIST OF TABLE

CHAPTER 1 OVERVIEW

The Integrated Debugger ID850QB for the V850 Series (hereafter referred to as the ID850QB) is a software tool developed for NEC Electronics V850 Series of microcontrollers for embedded control. This software tool is intended to enable efficient debugging of user programs.

This chapter explains the following items regarding the ID850QB.

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

1. 1 Features

The ID850QB has the following features:

- New functions, enhanced functions
- Other

1. 1. 1 New functions, enhanced functions

(1) Enhanced RRM function (when IECUBE connected)

Division setting for the sampling range of the real-time RAM monitor area is now possible (refer to "5. 12 The real-time RAM monitor function").

Up to 8 locations can be set in 256-byte units in the 2 KB RRM area of the ID850QB.

Moreover, in the Memory window, the access status (read, write, read & write) can be displayed in different colors using this function. This can also be used as a simple coverage function.

(2) Enhanced timer function (when IECUBE connected)

Using an external clock of 50 MHz, measurement of up to 7 segments + Run-Break time (time from program execution start until break) is possible (refer to "5. 9 Timer Function").

Other than during Run-Break, the maximum time, minimum time, pass count, and average time are displayed.

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

(3) Fail-safe break support (when IECUBE connected)

In addition to the traditional guarded areas and I/O register areas, fail-safe break is also supported for the guarded areas of internal ROM/internal RAM (refer to "5. 4. 5 Fail-safe break").

(4) Enhanced command function

Script file specification is possible at ID startup (refer to "3. 2 Startup Option and Argument Specification").

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

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

(5) Settings during program execution (when IECUBE connected)

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

(6) Support of both IECUBE and N-Wire CARD

Two emulators (IECUBE, N-Wire CARD) can be connected with 1 debugger (ID850QB) (refer to "1. 2 System Configuration").

In the case of IECUBE, USB2.0 is supported for the first time by NEC Electronics.

(7) Trace complementation function (when IECUBE connected)

This function performs complementary display of instructions between branch instructions that cannot be traced by the hardware (refer to "5. 10. 2 Setting trace data"). Complementation/non-complementation can be selected. In the complementation mode, assemble display in the internal ROM area is possible even during program execution (while the tracer is stopped).

(8) 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. Pasting to support mail is now easy.

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. 11 Event function").

(2) Support of on-chip debugging (when N-Wire CARD connected)

A debugging function implemented by the on-chip debug unit of the Nx85ET（RCU0+TEU+TRCU）, Nx85E901（RCU0）, RCU1 is provided.

(3) Flash memory writing function (when N-Wire CARD 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. 5 Flash Memory Writing Function (When N-Wire CARD connected)").

(4) Security function (when N-Wire CARD connected)

The ID code stored in the internal ROM or internal flash memory of a product with a security unit can be authenticated (refer to " Configuration dialog box").

(5) Function expansion through Tcl

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

(6) Function expansion through TIP or ToolLink

By associating with a task debugger (RD) and system performance analyzer (AZ) supporting TIP (Tool Interface Protocol) or ToolLink, the debugging efficiency of the user program using a real-time OS (RX) can be dramatically improved.

1. 2 System Configuration

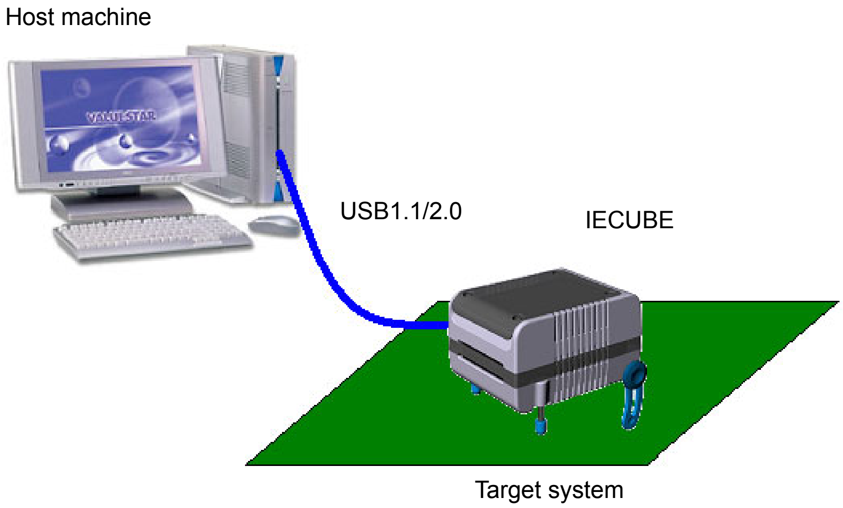
The ID850QB can be connected to the following two types of emulators.

As a result, user programs developed for the V850 Series and a pleasant debugging environment for target systems are provided.

(1) IECUBE (in-circuit emulator)

The IECUBE can be manipulated from the ID850QB by connected it to the ID850QB via a USB cable.

Figure 1-1 Example of ID850QB System Configuration (IECUBE)

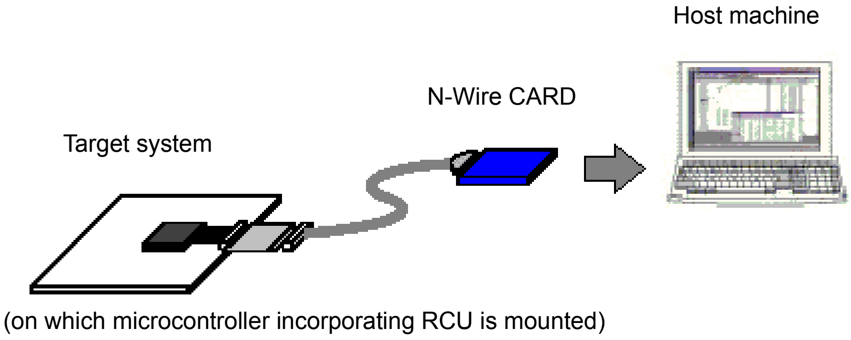


(2) N-Wire CARD (on-chip debug emulator)

The N-Wire CARD, a PC card emulator, can be manipulated from the ID850QB by directly inserting it to the host machine.

The N-Wire CARD can provide debugging functions by connecting it to a target system on which the V850ES incorporating an RCU (Run Control Unit) and the V850E1 are mounted.

Figure 1-2 Example of ID850QB System Configuration (N-Wire CARD)



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

Caution When the N-Wire CARD is connected, because it is assumed that the IECUBE is used with a notebook PC, the host machine must have a PC card slot (TYPEII).

(2) In-circuit emulator (any of the following)

- IECUBE
- N-Wire CARD (IE-V850E1-CD-NW)

1. 3. 2 Software environment

(1) OS (any of the following)

Windows98, Windows2000, Windows NT4.0 (Service Pack6 or above)**Note**, WindowsMe, WindowsXP Home Edition, WindowsXP Professional Edition

Note Can be used only when N-Wire CARD is connected.

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

(2) Device file (Individual acquisition)

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

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

http://www.necel.com/micro

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 plus

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

(2) When using CA850 on standalone basis

Add the -g option.

1. 4. 2 Security ID

The object file used when N-Wire CARD is connected must include the security ID information.

For the security ID850QB settings, refer to "CA850 C Compiler Package Manipulation".

For details about the security ID, refer to the N-Wire CARD user's manual.

The security ID (ID code) from the ID850QB is specified in the Configuration dialog box.

CHAPTER 2 INSTALLATION

This chapter explains the following items about installation of ID850QB.

- Installing
- Uninstalling

2. 1 Installing

The following items must be installed, when the ID850QB is used.

Table 2-1 Install

Item	Procedure
ID850QBsystem 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] -> [NEC Tools32] -> [Device File Installer].

Caution 1 To install the ID850QB again after the ID850QB has been installed once, be sure to uninstall the ID850QB. If the ID850QB is installed in a directory different from that, without uninstalling, the ID850QB that has already been installed cannot be uninstalled.

Caution 2 When ID850NWC is already installed, ID850QB cannot be installed in the same machine. In this case, uninstall ID850NWC before installing ID850QB.

2. 2 Uninstalling

Perform uninstall using [Add/Remove Programs] in the Control Panel.

CHAPTER 3 STARTING AND TERMINATING

This chapter explains the procedure of starting and terminating the ID850QB.

- Cautions Before Starting (When N-Wire CARD connected)
- Startup Option and Argument Specification
- Starting
- Terminating
- Error Messages at Start up

3. 1 Cautions Before Starting (When N-Wire CARD connected)

When the N-Wire CARD is connected, start the N-Wire Checker before the starting the ID850QB to check that the in-circuit emulator and the target system can be normally debugged. (Refer to "APPENDIX B N-WIRE CHECKER".)

Caution For the connection between the in-circuit emulator and the target system and the power application sequence, refer to "IE-V850E1-CD-NW User's Manual". Incorrect connection may damage the in-circuit emulator and the target system.

3. 2 Startup Option and Argument Specification

The procedure for specifying the startup options and arguments for the ID850QB is described below.

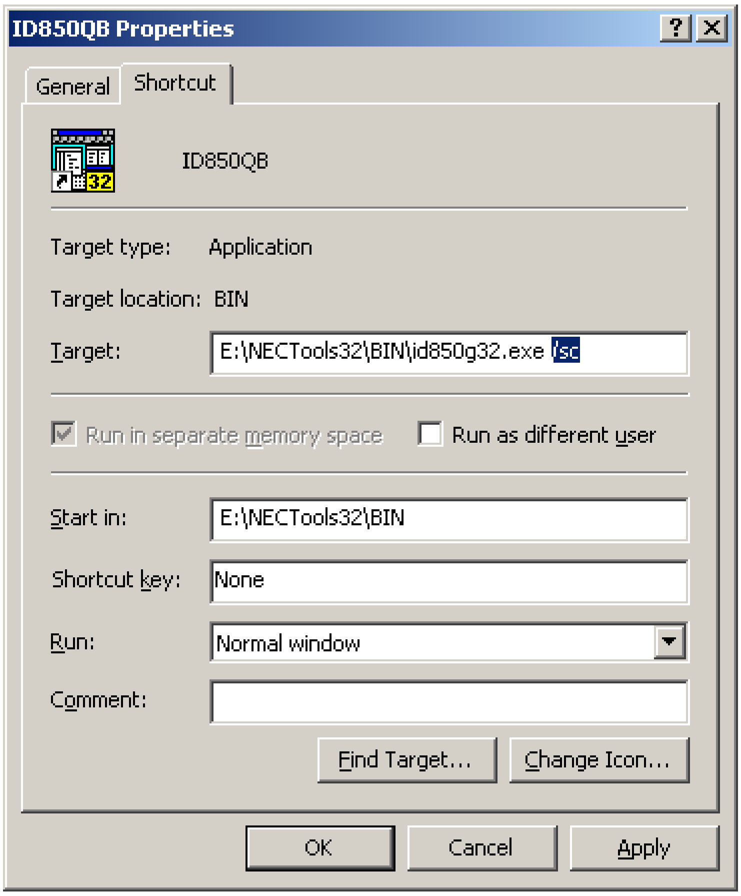
By specifying the startup options and arguments, it is possible to specify the script file at startup and the project file.

Remark When starting up the ID850QB from PM plus, the startup option and argument settings are performed in [Debugger Settings...] in the Tool Menu of PM plus (refer to "CHAPTER 4 ASSOCIATION WITH PM plus"). The debugger startup option can be set to the option column.

3. 2. 1 Specification method

1) Create an ID850QB shortcut on the desktop.
   The ID850QB execution file (*.exe) is located in the bin folder in the folder to which the installation was performed (refer to "Table 3-2 Execution File").
2) Open the properties of the created shortcut and after the execution file name displayed in [Target:], specify the option and argument (refer to "3. 2. 2 Specification format and options").

Figure 3-1 Startup Option



ID850QB Properties

General | Shortcut

ID850QB

Target type: Application

Target location: BIN

Target: E:\NECTools32\BIN\id850g32.exe /sc

☑ Run in separate memory space ☐ Run as different user

Start in: E:\NECTools32\BIN

Shortcut key: None

Run: Normal window

Comment:

3. 2. 2 Specification format and options

(1) Specification format

```
xxx.exe ?options?
xxx.exe ?options? project
```

Each option and argument is separated by a space. No distinction is made between uppercase and lowercase in the character string.

Arguments enclosed between '?' can be omitted.

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

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

(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

Example 1) Specification of script file only

```
xxx.exe /script:c:/work/script.tcl
```

Example 2) Specification of script file and project file

```
xxx.exe /script:c:/work/script.tcl c:/work/project.prj
```

Caution Make the script file (*.tcl file) name excluding the extension different from the load module file name and project file name.

In the case of this example, do not place a file named project.tcl in c:/work.

(4) Execution file

Execution files and names displayed from the [Start] menu are as follows.

Table 3-2 Execution File

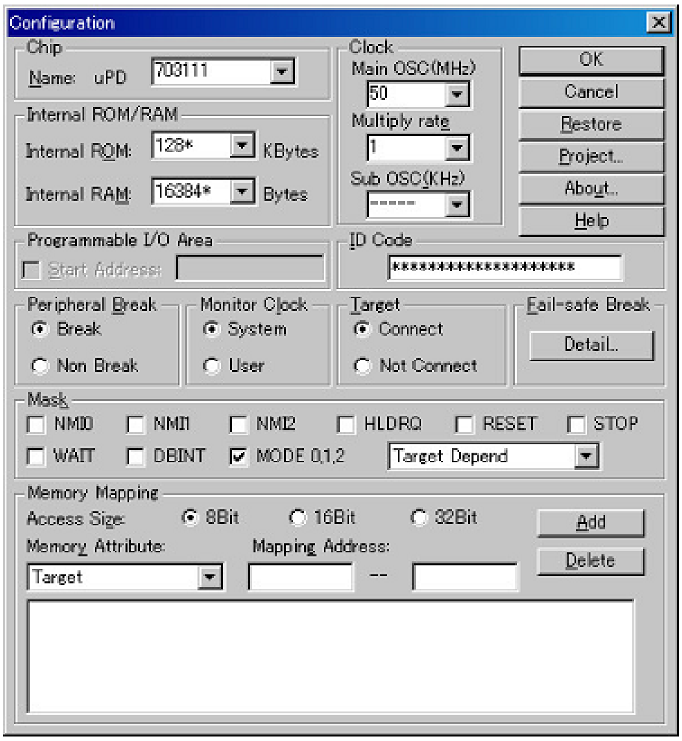
Connected IE	Execution file	Name ([Start] Menu)
IECUBE, N-Wire CARD	id850g32.exe	ID850QB

3. 3 Starting

1) ID850QB startup is executed from the [Start] menu or by clicking the shortcut created on the desktop. Start the ID850QB, the Configuration dialog box will be opened.

Caution In this case, the Configuration dialog box should not be displayed, but an error message should be displayed, please cope with it with reference to "3. 5 Error Messages at Start up" .

Figure 3-2 Configuration Dialog Box



Chip
Name: uPD 703111

Internal ROM/RAM
Internal ROM: 128* KBytes
Internal RAM: 16384* Bytes

Clock
Main OSC(MHz): 50
Multiply rate: 1
Sub OSC(KHz): -----

Programmable I/O Area
☐ Start Address:

ID Code: ********************

Peripheral Break: ● Break ○ Non Break

Monitor Clock: ● System ○ User

Target: ● Connect ○ Not Connect

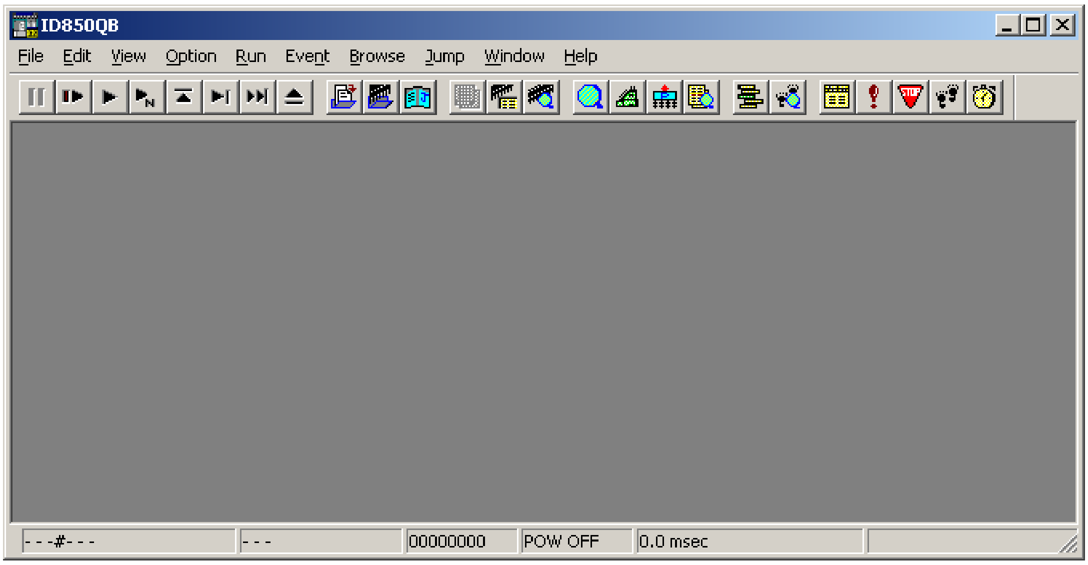
Mask
☐ NMI0 ☐ NMI1 ☐ NMI2 ☐ HLDRQ ☐ RESET ☐ STOP
☐ WAIT ☐ DBINT ☑ MODE 0,1,2 Target Depend

Memory Mapping
Access Size: ● 8Bit ○ 16Bit ○ 32Bit
Memory Attribute: Target
Mapping Address: --

2) Set the items related to the operating environment of the ID850QB 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 ID850QB can be operated. Mainly use this window for debugging.

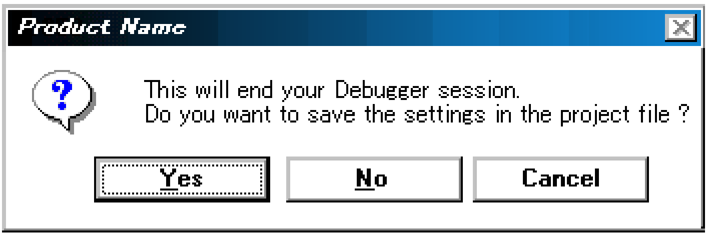
Figure 3-3 Main Window (Startup)



3. 4 Terminating

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

Figure 3-4 Exit Debugger Dialog Box



This will end your Debugger session.
Do you want to save the settings in the project file ?

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 ID850QB terminated.

3. 5 Error Messages at Start up

Error messages that may be output when ID850QB starts up are listed below (by order of occurrence).

When these messages are output, refer to "APPENDIX E MESSAGES".

If both IECUBE and N-Wire CARD are connected, the USB connection (IECUBE) has priority.

(1) When the IECUBE is connected

F0100 : Can not communicate with ICE. Please confirm the installation of the device driver for the PC interface board.
F0c43: Can not communicate with ICE. (N-Wire CARD) Can not communicate with ICE. Please confirm the power of ICE, connection of the interface cable. (IECUBE)
F0c70: DCU cannot be accessed.
F0c76 : Initial state at the time of DCU access start is unusual.
F0c77 : DCU access is unusuall.
Ff606: Please check connection with the target board, and power on it.
Wf607 : Please check connection of the exchange adapter.
Ff608 : Please disconnect the target board.
A0105: Failed in reading device file (d3xxx.800).
F0ca2: Device file which does not correspond to on tip debug.
F0ca4 : Device file which does not correspond to IECUBE.
F0c71: Reset cannot be performed.
F0c72: Monitor memory cannot be accessed.
F0c73: Monitor execution cannot be performed.
F0c74: CPU register cannot be accessed.
F0c23: Bus hold under continuation.
A0c01: During access of register, CPU did time out.
A0c02: During access of memory, CPU did time out.
A0c03: During access of I/O register, CPU did time out.
A01a0: No response from the evachip. Please confirm the signal of the CLOCK or RESET WAIT, HLDRQ and so on.(IECUBE) No response from the CPU. Please confirm the signal of the CLOCK or RESET WAIT, HLDRQ and so on.(N-Wire CARD) No response from the emulation CPU. Please confirm the signal of the CLOCK or RESET,WAIT and so on.(78K0)

(2) When the N-Wire CARD is connected

F0100 : Can not communicate with ICE. Please confirm the installation of the device driver for the PC interface board.
F0c43: Can not communicate with ICE. (N-Wire CARD) Can not communicate with ICE. Please confirm the power of ICE, connection of the interface cable. (IECUBE)
F03a0: Target is not turned on.
F0c70: DCU cannot be accessed.
F0c76 : Initial state at the time of DCU access start is unusual.
F0c77 : DCU access is unusuall.
A0105: Failed in reading device file (d3xxx.800).
F0ca2: Device file which does not correspond to on tip debug.
F0ca3: Information that it does not support is included in on tip debug information on device file.
F0c24: It cannot shift to debug mode.
F0c72: Monitor memory cannot be accessed.
F0c73: Monitor execution cannot be performed.
F0c74: CPU register cannot be accessed.
F0c23: Bus hold under continuation.
A0c01: During access of register, CPU did time out.
A0c02: During access of memory, CPU did time out.
A0c03: During access of I/O register, CPU did time out.
A01a0: No response from the evachip. Please confirm the signal of the CLOCK or RESET WAIT, HLDRQ and so on.(IECUBE) No response from the CPU. Please confirm the signal of the CLOCK or RESET WAIT, HLDRQ and so on.(N-Wire CARD) No response from the emulation CPU. Please confirm the signal of the CLOCK or RESET,WAIT and so on.(78K0)

CHAPTER 4 ASSOCIATION WITH PM plus

The ID850QB 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 plus.

This chapter explains the following items about association with the PM plus.

- Setting Build Mode
- Registering Debugger to PM plus Project
- To Start ID850QB from PM plus
- Auto Load

Caution If a load module file is created by using the Windows command prompt, the function to associate the ID850QB with the PM plus cannot be used.

4. 1 Setting Build Mode

To debug the load module file created by the PM plus on the ID850QB 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 plus.

4. 2 Registering Debugger to PM plus Project

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

4. 2. 1 Selecting debugger

The procedure for selecting the debugger is as follows:

(1) Creating a new workspace

1) Select [File] menu -> [New Workspace...] on the PM plus.
-> 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 [New Workspace - Step 7/8 (Select Debugger)] dialog box will be opened. Specify ID850QB in this dialog box. (Refer to "Table 3-2 Execution File".) For details of the setting, refer to the "PM plus online help" or "User's manual".

(2) Using an existing workspace

1) Select [Tool] menu -> [Debugger Settings...] on the PM plus. -> The [Debugger Settings] dialog box will be opened.

2) Specify ID850QB and click the <OK> button in this dialog box. (Refer to "Table 3-2 Execution File".)

The ID850QB is registered as the debugger of the active project. The ID850QB icon is displayed on the toolbar of the PM plus.

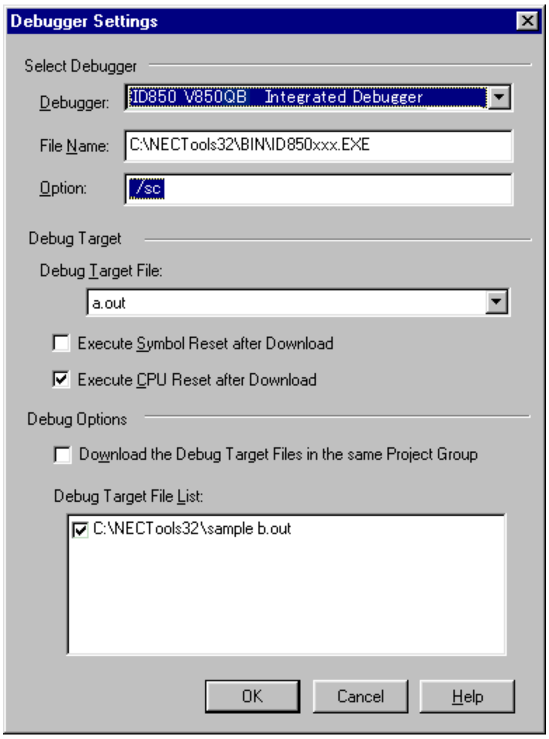
4. 2. 2 Downloading multiple load module files

The load module files of the same project group are downloaded to the ID850QB.
For details of the project group, refer to the "PM plus online help" or "User's manual".

Download multiple load module files in the following steps.

1) Select [Tool] menu -> [Debugger Settings...] on the PM plus. -> The [Debugger Settings] dialog box will be opened.

Figure 4-1 [Debugger Settings] Dialog Box (PM plus)



Debugger Settings

Select Debugger

Debugger: ID850 V850QB Integrated Debugger

File Name: C:\NECTools32\BIN\ID850xxx.EXE

Option: /sc

Debug Target

Debug Target File:

a.out

☐ Execute Symbol Reset after Download

☑ Execute CPU Reset after Download

Debug Options

☐ Download the Debug Target Files in the same Project Group

Debug Target File List:

☑ C:\NECTools32\sample b.out

2) Specify the ID850QB in the [Debugger:]. (Refer to "Table 3-2 Execution File".)
-> The execution file is automatically displayed in the [File Name:].

3) Specify the option in the [Option:]. (Refer to "3. 2. 2 Specification format and options".)

4) Specify the load module file to be downloaded first in the [Debug Target File:] when the ID850QB is started.

5) Check the [Download the debug target files in the same project group.] check box, and select the debug target files in the same group to be downloaded in the [Debug Target File List:].
->The selected file will be downloaded right after the file specified in the [Debug Target File:] when the ID850QB is started. Note that when these files are downloaded, symbol reset and CPU reset are not executed.

6) Click the <OK> button.

Remark Multiple load module files that are being downloaded can be checked in the Load Module List dialog box of the ID850QB.

Caution If [Execute symbol reset after dowmload] in the [Debugger Settings] dialog box of the product with a internal flash memory is checked, the contents of the internal flash memory are erased before downloading (when N-Wire CARD connected).

4. 3 To Start ID850QB from PM plus

The ID850QB can be started from the PM plus as follows:

- Click the ID850QB starting button on the toolbar of the PM plus.
- Select the [Build] menu -> [Debug] on the PM plus.
- Select the [Build] menu -> [Build and Debug] on the PM plus.
- Select the [Build] menu -> [Rebuild and Debug] on the PM plus.

If the debugging environment of the ID850QB is saved to a project file currently being used by the PM plus, it will be started in the debugging environment saved in the project file.

If the debugging environment of the ID850QB is not saved to a project file being used by the PM plus, the Configuration dialog box is opened. At this time, the device type (chip name) cannot be changed.

4. 3. 1 Restoring debugging environment

The previous debugging environment can be restored by the following procedure when the ID850QB is started from the PM plus:

1) Create a new workspace (project file: e.g., sample.prj) on the PM plus**Note**.

2) Start the ID850QB from the PM plus. 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 ID850QB is started.

3) Download the load module file to be debugged with the Download dialog box of the ID850QB.

4) Debug the load module file on the ID850QB.

5) Click the <Yes> button on the Exit Debugger dialog box when the ID850QB is terminated.
-> The debugging environment will be saved to the project file (sample.prj) for the PM plus when the ID850QB 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 ID850QB debugging).

6) When the ID850QB is next started up after the sample.prj file is read by PM plus, the debug environment at the point when the project file was saved is automatically restored.

Note In the ID850QB and PM plus, the environment information is saved to a project file and referenced. The extension of the project file that can be used by the ID850QB and PM plus is "prj". For the information that is saved or restored by the project file, refer to the "Online help" or the "Uer's manual" of each product.

4. 4 Auto Load

If a bug is found while the load module file is being debugged by the ID850QB, 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 ID850QB by compiling and linking the file on the PM plus with the activated ID850QB. (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 plus.

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 ID850QB (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 -> [Correct Source] on the ID850QB.
-> An editor will be opened and the specified source file will be read.

3) Correct the source file on the editor.

4) Terminate the editor.

Caution The CPU reset is not performed when the load module file is automatically downloaded. The debug window that was opened when the editor was called, and each event setting will be restored. If the previously used line or symbol has been deleted as a result of correcting the source file, the following happens:
- A variable that was displayed is dimmed.
- The event mark of an event condition is displayed in yellow.
- A software break point may be deleted.

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

4. 4. 2 Auto load by starting debugger

If the following operation is performed on the PM plus with the ID850QB started, the load module will be automatically downloaded to the ID850QB.

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

Caution The CPU reset is not performed after the load module has been downloaded.

CHAPTER 5 DEBUG FUNCTION

This chapter explains about debug function of ID850QB.

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 break point	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
To check the trace data	5. 10 Trace Function (When IECUBE Connected)
To manage the events	5. 11 Event function
RRM function	5. 12 The real-time RAM monitor function
To save the debug environment and window status	5. 13 Load/Save Function
Jump function, linking window function	5. 14 Functions Common to Various Windows

5. 1 Setting debugging environment

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

- Setting operating environment
- Option settings
- Mapping settings
- To change the value of a register required for access of an external memory

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 ID850QB starts up.

If a project file already exists, the debugging environment can be restored by clicking the <Project...> button. (Refer to "5. 13. 1 Debugging environment (project file)".)

5. 1. 2 Option settings

The expanded options and debug options are set in the Extended Option dialog box, the Debugger Option dialog box .

5. 1. 3 Mapping settings

The mapping settings are performed in the Configuration dialog box.

The following types of mapping attributes are available:

Table 5-2 Details of Mapping Attributes

Attribute	Meaning
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.
Internal ROM	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 RAM	A memory area specified as the internal RAM is equivalent to the internal RAM of the target device (core). The actual memory configuration depends on the target system.
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 IOR window or the Watch window.

5. 1. 4 To change the value of a register required for access of an external memory

When mapping has been performed for external memory, must change the values of the registers required for accessing external memory prior to downloading, using the IOR window or the hook procedure.

For how to change register values using the hook procedure, refer to "7. 8 Hook Procedure".

For the registers to be changed, refer to the hardware manual of the CPU that is used.

5. 2 Download Function / Upload Function

ID850QB allows downloading and uploading of object files of the formats listed in the following table: Table 5-3 , Table 5-4 .

- Download
- Upload

Remark When the N-Wire CARD is connected, the internal flash memory can be written and the load module can be downloaded (refer to "5. 7. 5 Flash Memory Writing Function (When N-Wire CARD connected)").

5. 2. 1 Download

Object file download is done in the Download dialog box.

Multiple load module files can be downloaded.

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

Loaded files can be checked in the Load Module List dialog box that is opened by selecting [File] menu -> [Load Module] .As a result, loading results when loading multiple load module files can be checked.

Table 5-3 Type of File That Can Be Downloaded

Format	Extension
Load module (ELF (.OUT))	Load Module(*.out)
Intel Hex format (00, 01, 02, 03 - 20bit address)	Hex Format (*.hex)**Note**
Motorola Hex format S type (S0, S3, S7, S8)	
Extended Tektronix Hex format	
Binary data	Binary Data (*.bin)

Note Automatic format detection

5. 2. 2 Upload

Uploading of memory contents, etc., is performed in the Upload dialog box.

The saving range can be set.

Table 5-4 Type of File That Can Be Uploaded

Format	Extension
Intel Hex format (00, 01, 02, 03 - 20bit-address)	Hex Format (*.hex)[Note]
Motorola Hex format S type (S0, S3, S7 - 32bit-address)	
Extended Tektronix Hex format	
Binary data	Binary Data (*.bin)

Note Specify saving format.

Remark In the Intel hex format, the format is always extended (20 bits).

5. 3 Source Display, Disassemble Display Function

Source file display is performed in the Source window. Disassemble display and online assemble are performed in the Assemble window.

- Source display
- Disassemble display
- Mixed display mode (Source window)
- Convert symbol (Symbol to Address)

5. 3. 1 Source display

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

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

Specifications related to the tab size, display font, etc., and specification of the source path are done 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)	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 done in the Assemble window.

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

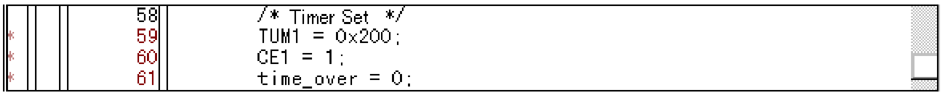
Offset display and register name display specification are performed in the Debugger Option dialog box.

Specify a serching 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)

Disassemble display of programs combined with the source file can be done by selecting [View] menu -> [Mix] in the Source window. The display contents in the mixed display mode can be saved as a view file.

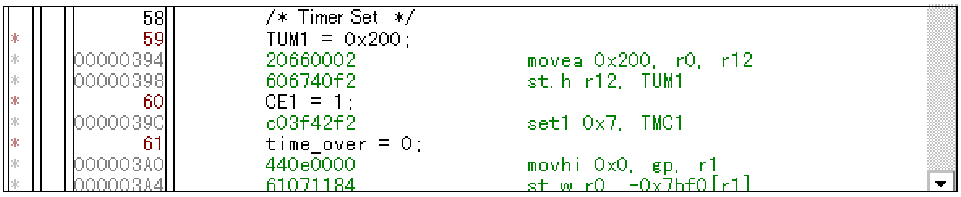
Normal display mode



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

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

Mixed display mode



```
        58          /* Timer Set  */
*       59          TUM1 = 0x200;
*  00000394         20660002                movea 0x200, r0, r12
*  00000398         606740f2                st.h r12, TUM1
*       60          CE1 = 1;
*  0000039C         c03f42f2                set1 0x7, TMC1
*       61          time_over = 0;
*  000003A0         440e0000                movhi 0x0, gp, r1
*  000003A4         61071184                st.w r0, -0x7bf0[r1]
```

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

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

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)
Line number of source file	file#no prog$file#no
I/O port name	portname
I/O register name	I/O regname
Register name	regname
PSW flag name	pswname

Remark 1 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.

Remark 2 Separator ’$’

To specify a load module name when two or more load modules are read, use "$" as a separator to delimit the load module name from a file name, variable, function name, or symbol name. In the default status, a symbol name takes precedence. To temporarily change the priority, prefixing "$" to a symbol gives the priority to a register name.

5. 4 Break Function

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

- Break Types
- Breakpoint setting
- Setting break to variable
- Hardware break and software break
- Fail-safe break

5. 4. 1 Break Types

The 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 even 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 stop condition of each command ([Step In], [Next Over], [Return Out], [Slow Motion]).
Forced break	Function to forcibly stop execution by selecting [Run] menu -> [Stop], and selecting the STOP button. It is valid for all the execution commands.
Fail-safe break	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	Function to stop user program execution when the measurement time exceeds the specified timeout 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 the 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. 11. 4 Number of enabled events for each event condition".)

(1) Breakpoint setting method

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

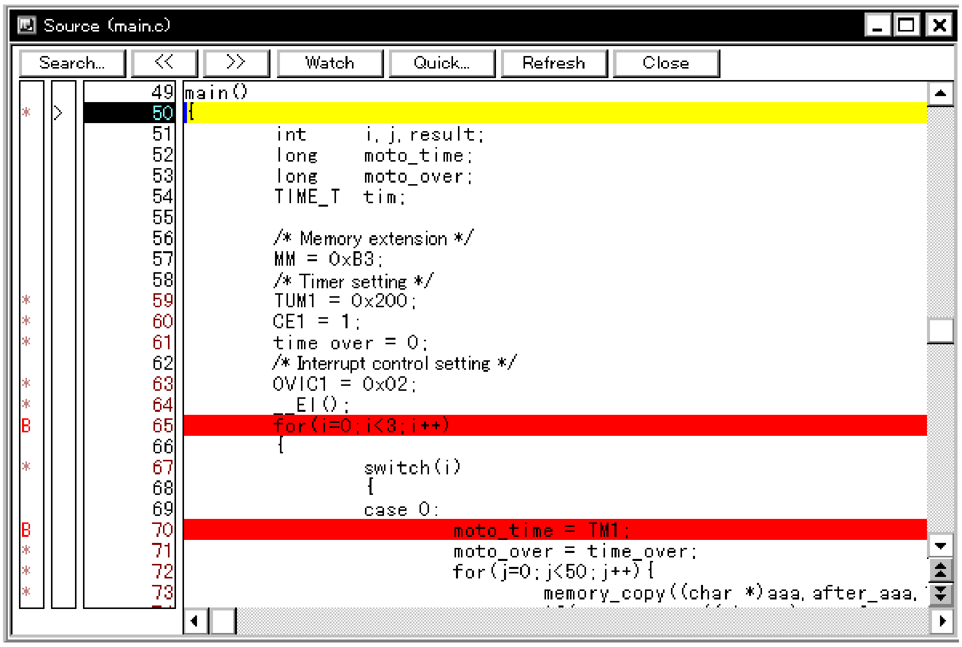
In the default setting, software breakpoint (B) is set, but if [Breakpoint] is selected in the context menu, hardware breakpoint (B, or B) is set.

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 6-11 Event Setting Status (Event Mark)").

Caution A software breakpoint cannot be set 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



```
main()
{
        int     i, j, result;
        long    moto_time;
        long    moto_over;
        TIME_T  tim;

        /* Memory extension */
        MM = 0xB3;
        /* Timer setting */
        TUM1 = 0x200;
        CE1 = 1;
        time over = 0;
        /* Interrupt control setting */
        OVIC1 = 0x02;
        __EI();
        for(i=0;i<3;i++)
        {
                switch(i)
                {
                case 0:
                        moto_time = TM1;
                        moto_over = time_over;
                        for(j=0;j<50;j++){
                                memory_copy((char *)aaa, after_aaa,
```

(2) Deleting a breakpoint method

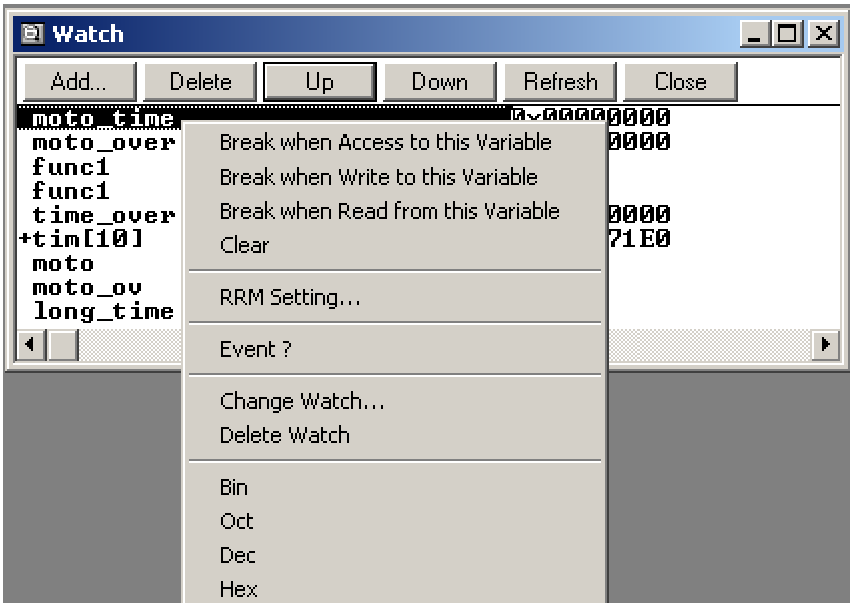
Click the position at which the breakpoint to be deleted is set.

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 break to variable

Access break setting to variables can easily be done from the context menu in the Source window or the Watch window.

Figure 5-2 Setting Break to Variable



5. 4. 4 Hardware break and software break

(1) Hardware break

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

Therefore, in the ID850QB, they are managed using "5. 11 Event function" as break event conditions.

The number of valid break points varies depending on the device (refer to "5. 11. 4 Number of enabled events for each event condition").

(2) Software break

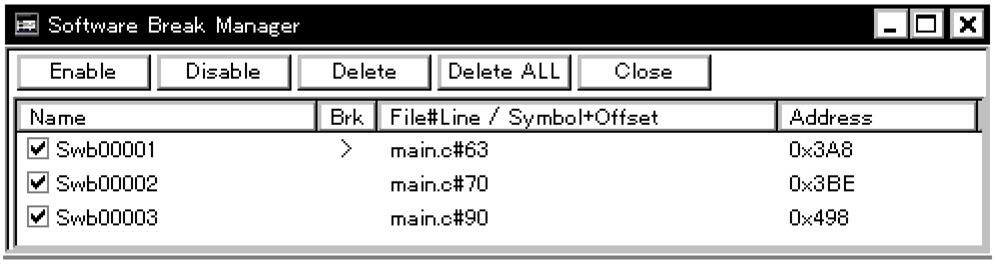
Software breaks are breaks that are set by rewriting instructions of specified addresses to software break instructions. Therefore, the number of software breaks that can be set is not limited, but settings to external ROM, stopping at variable access timing, etc., cannot be specified.

Table 5-8 The Number of Valid Software Break

Connected IE	Valid Number
IECUBE	2000
N-Wire CARD	2000 **Note:** Software breaks in relation to internal ROM and internal flash memory are automatically set by the ROM collection function. The maximum number of software breaks that can be set with the ROM collection function depends on the product (0, 4, or 8). Moreover, the software breaks set with the ROM collection function are temporarily disabled by target reset or internal reset, but they are enabled when a break occurs.

Software break is managed by the Software Break Manager.

Figure 5-3 Management of Software Breaks



Name	Brk	File#Line / Symbol+Offset	Address
☑ Swb00001	>	main.c#63	0x3A8
☑ Swb00002		main.c#70	0x3BE
☑ Swb00003		main.c#90	0x498

5. 4. 5 Fail-safe break

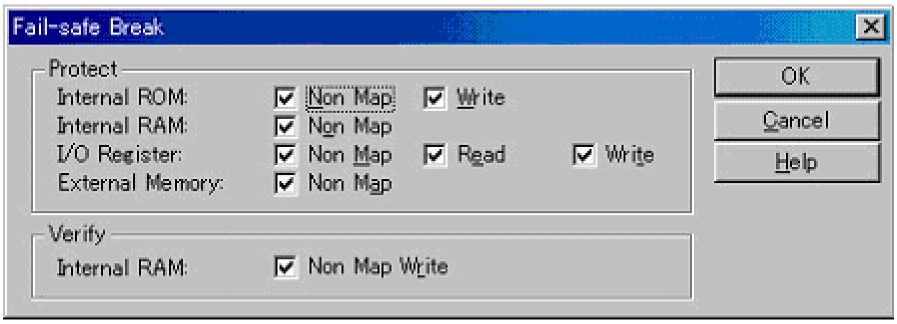
The fail-safe break settings are performed in the Fail-safe Break dialog box.

Individual settings are possible by selecting check boxes.

However, the protect setting for the internal RAM area is performed through verify processing by software, and therefore a warning is displayed during breaks.

During IOR Illegal and internal RAM verify checks, the address is displayed on the status bar.

Figure 5-4 Fail-safe Break Setting



Fail-safe Break

Protect

Internal ROM: ☑ Non Map ☑ Write

Internal RAM: ☑ Non Map

I/O Register: ☑ Non Map ☑ Read ☑ Write

External Memory: ☑ Non Map

Verify

Internal RAM: ☑ Non Map Write

5. 5 Program Execution Function

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

Through user program execution, the PC advances until the set breakpoint or forced break. (Refer to "5. 4 Break Function".)

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 ID850QB execution functions are provided. They are operated using the execution buttons on the toolbar , or from the [Run] menu.

Figure 5-5 Execution Button

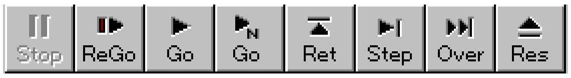


Figure 5-6 [Run] Menu

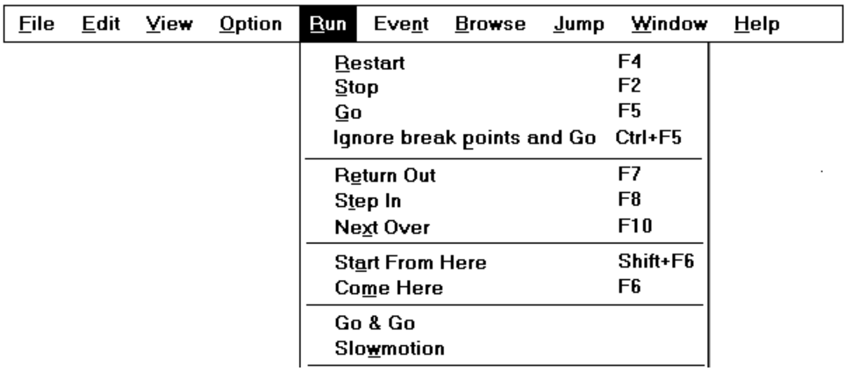


Table 5-9 Type of Execution

Items	Contents
[Restart]	The CPU is reset and the user program is executed starting from address 0. This is the same operation as "resetting the CPU before execution of the user program and executing [Go]".
[Go]	The user program is executed starting from the address indicated by the current PC register and is stopped if a set break event condition is satisfied.
[Ignore break points and Go]	The user program is executed starting from the address indicated by the current PC register Execution of the user program continues, ignoring set breakpoints.
[Return Out]	The user program is executed until execution returns to the calling function described in C language.

Table 5-9 Type of Execution

Items	Contents
[Step In]	In the Source window,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 Assemble window,One instruction is executed from the current PC register value and the contents of each window are updated.
[Next Over]	**jarl instruction** Next step execution is performed, assuming the function or subroutine called by the jarl instruction as one step (step execution continues until the nesting level becomes the same as when the jarl instruction was executed). **Instruction other than jarl** 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 the 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].

5. 6 Watch Function

This section explains the following items about watch function.

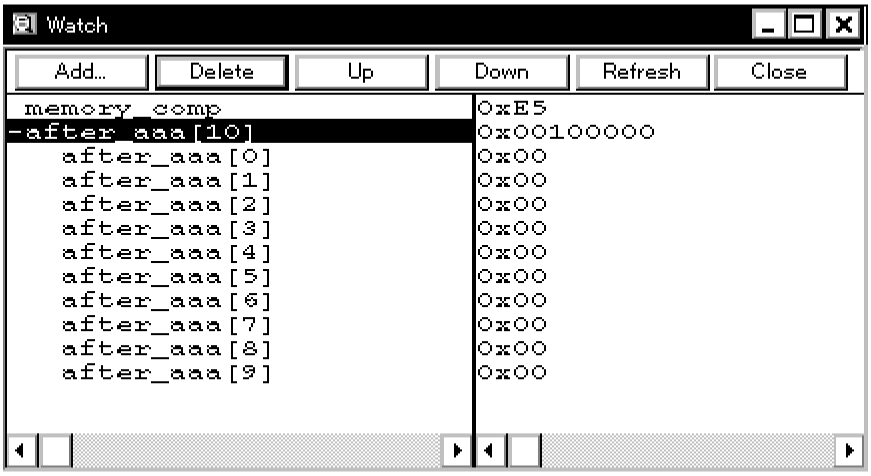
- Displaying and changing data value
- Displaying and changing local variable values
- Registering and deleting watch data
- Changing watch data
- Temporarily displaying and changing data values
- Toolbar watch function
- Stack trace display function

5. 6. 1 Displaying and changing data value

Displaying and changing data values is done in the Watch window. Shifts in data values can be checked by registering watch data.

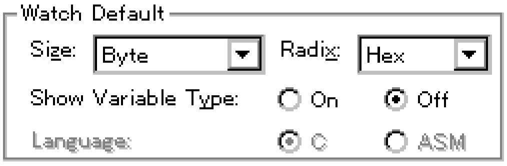
Specification of the display format is done in the Debugger Option dialog box.

Figure 5-7 Watch Window



```
 memory_comp          0xE5
-after_aaa[10]        0x00100000
    after_aaa[0]      0x00
    after_aaa[1]      0x00
    after_aaa[2]      0x00
    after_aaa[3]      0x00
    after_aaa[4]      0x00
    after_aaa[5]      0x00
    after_aaa[6]      0x00
    after_aaa[7]      0x00
    after_aaa[8]      0x00
    after_aaa[9]      0x00
```

Figure 5-8 Specification of the Display Format (Debugger Option dialog box)



Watch Default

Size: Byte Radix: Hex

Show Variable Type: ○ On ● Off

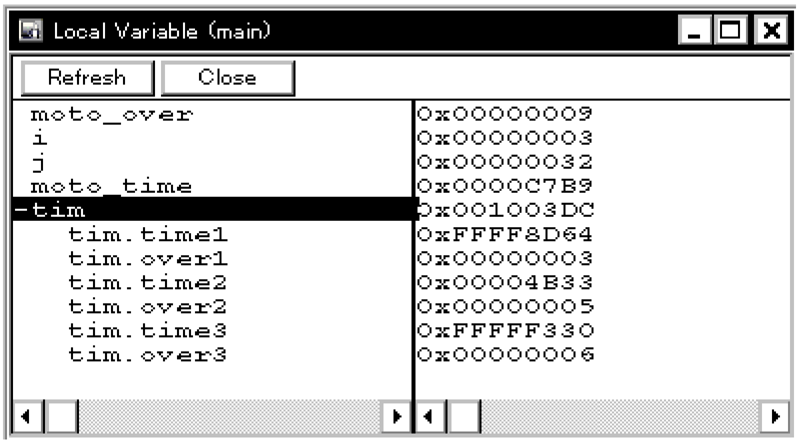
Language: ● C ○ ASM

5. 6. 2 Displaying and changing local variable values

Displaying and changing local variable values is done 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



moto_over	0x00000009
i	0x00000003
j	0x00000032
moto_time	0x0000C7B9
-tim	0x001003DC
tim.time1	0xFFFF8D64
tim.over1	0x00000003
tim.time2	0x00004B33
tim.over2	0x00000005
tim.time3	0xFFFFF330
tim.over3	0x00000006

5. 6. 3 Registering and deleting watch data

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

- Drag and drop the selected variable or symbol name directly on the Watch window. (Refer to "5. 14. 4 Drag & drop function".)
- Click the <Add> button in the Quick Watch dialog box or the Add Watch dialog box.

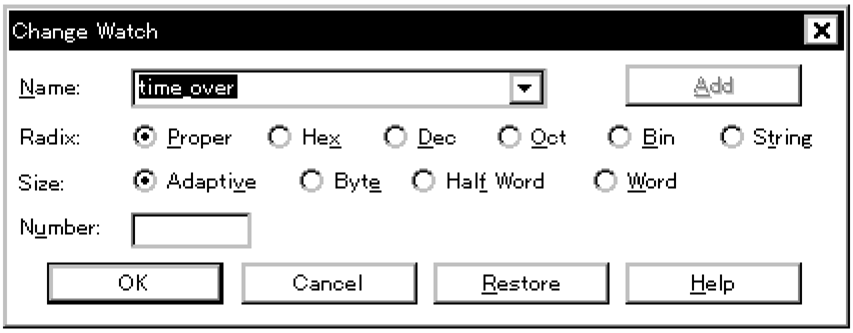
To delete watch data, click the variable name or symbol name (multiple selections can also be made using the Shift key or Ctrl key), and then click the <Delete> button. However, lines with an expanded hierarchy, such as elements of an array, and members of structures and unions, cannot be deleted.

5. 6. 4 Changing watch data

Changing watch data is done in the Change Watch dialog box.

Changing a symbol name is permitted even when any existing data of symbols of the same name has already existed.

Figure 5-10 Change Watch Dialog Box



Name: time_over

Radix: ● Proper ○ Hex ○ Dec ○ Oct ○ Bin ○ String

Size: ● Adaptive ○ Byte ○ Half Word ○ Word

Number:

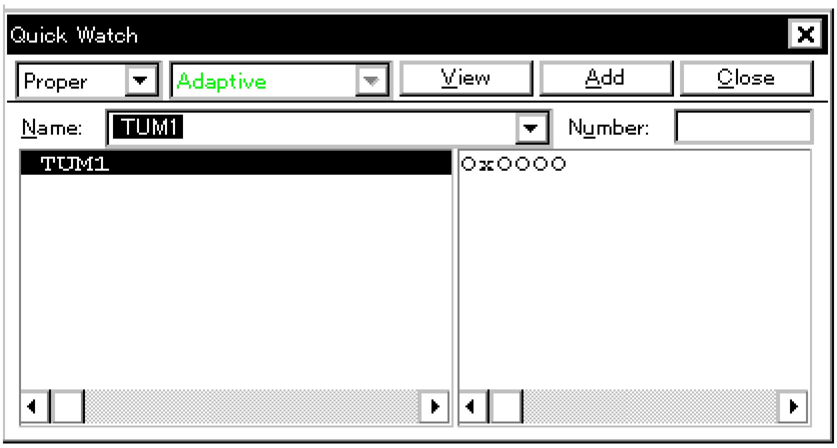
5. 6. 5 Temporarily displaying and changing data values

Temporarily displaying and changing data values is done in the Quick Watch dialog box.

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

The display code type, display size, and display count can all be changed in this window.

Figure 5-11 Quick Watch dialog box



Proper Adaptive

Name: TUM1 Number:

TUM1 0x0000

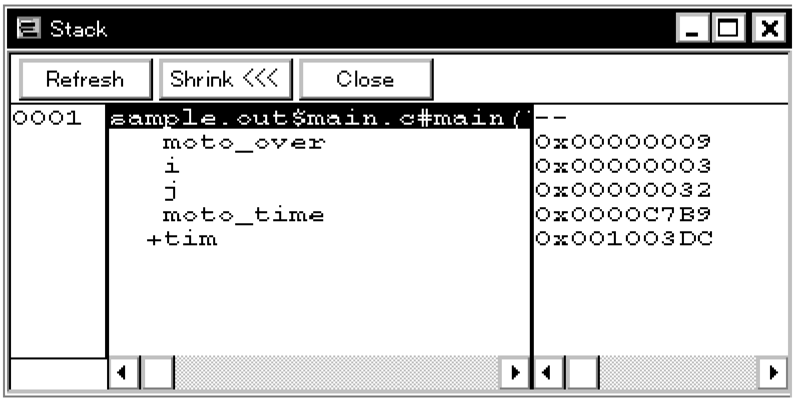
5. 6. 6 Toolbar watch function

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

5. 6. 7 Stack trace display function

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

Figure 5-12 Stack Trace Window



```
0001 sample.out$main.c#main(   --
        moto_over              0x00000009
        i                      0x00000003
        j                      0x00000032
        moto_time              0x0000C7B9
      +tim                     0x001003DC
```

5. 7 Memory Manipulation Function

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

Verify check specification, etc., is done in the Extended Option dialog box.

- Displaying and changing memory contents
- Filling, copying, and comparing memory contents
- Modifying memory contents (DMM function)
- Access monitor function (when IECUBE connected)
- Flash Memory Writing Function (When N-Wire CARD connected)

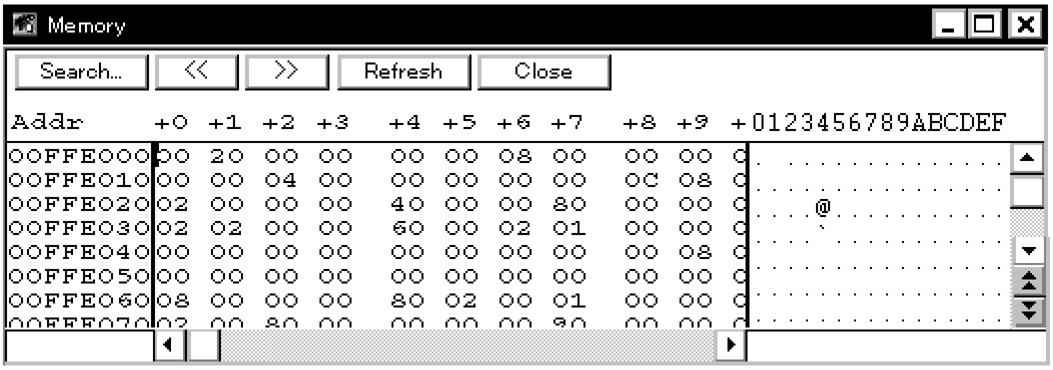
5. 7. 1 Displaying and changing memory contents

In the Memory window, the memory contents can be displayed or changed by using mnemonic codes, hexadecimal codes, and ASCII codes. Search is done in the Memory Search dialog box displayed by clicking the <Search...> button. The result 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 real-time internal RAM sampling range can be displayed in real time even during program execution. (Refer to "5. 12 The real-time RAM monitor function".)

Figure 5-13 Displaying and Changing Memory Contents



```
Addr      +0 +1 +2 +3  +4 +5 +6 +7  +8 +9 +0123456789ABCDEF
00FFE000  00 20 00 00  00 00 08 00  00 00 0 ................
00FFE010  00 00 04 00  00 00 00 00  0C 08 0 ................
00FFE020  02 00 00 00  40 00 00 80  00 00 0 ....@...........
00FFE030  02 02 00 00  60 00 02 01  00 00 0 ....`...........
00FFE040  00 00 00 00  00 00 00 00  00 08 0 ................
00FFE050  00 00 00 00  00 00 00 00  00 00 0 ................
00FFE060  08 00 00 00  80 02 00 01  00 00 0 ................
00FFE070  02 00 80 00  00 00 00 90  00 00 0 ................
```

5. 7. 2 Filling, copying, and comparing memory contents

Filling, copying, and comparing memory contents is done in the Memory Fill dialog box, the Memory Copy dialog box, and the Memory Compare dialog box displayed by selecting [Edit] menu -> [Memory] -> [Fill.../Copy.../ Compare....].

The comparison results are displayed in the Memory Compare Result dialog box.

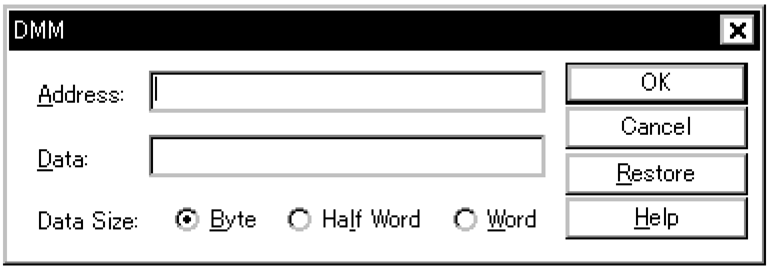
5. 7. 3 Modifying memory contents (DMM function)

Memory contents can be modified during user program execution in the DMM dialog box displayed by clicking the <Modify...> button in the Memory window.

The following points must be noted when using the DMM function.

- If data is written to an address for which a software break is set, the program may not run correctly.
- If the program is stopped immediately before rewriting the memory because a break event is generated, the generated break event is invalid because program execution is resumed, regardless of the break event.
- If data is written to an address for which a software break is set, the program may not run correctly.
- Verify check is always invalid. Even if writing has failed, a verify error is not generated. (When N-Wire CARD connected)
- No data can be written to the internal flash memory during user program execution (when N-Wire CARD connected).

Figure 5-14 Modifying Memory Contents (DMM Dialog Box)



5. 7. 4 Access monitor function (when IECUBE connected)

The access monitor function displays the access status (read, write, read & write) for the sampling range of the The real-time RAM monitor function using different colors in the Memory window.

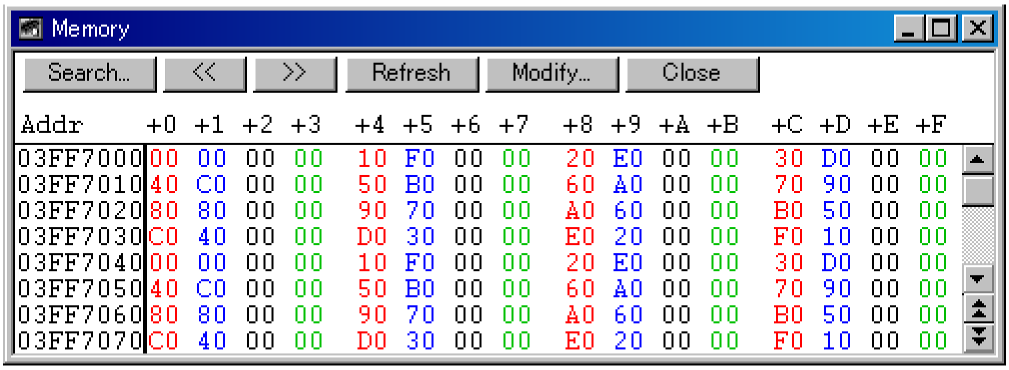
The access monitor color-marking function is executed only during byte display. Colors are not displayed in the ASCII display area.

Cumulative display setting and access status display can be cleared by selecting [View] menu -> [Access Monitoring].

Caution1 The value of memory rewritten via DMA during program execution, and the value of memory rewritten from the debugger cannot be displayed on the access monitor.

Caution2 This function is enabled only when [Option] menu -> [RRM Function] is selected.

Figure 5-15 Access Monitor function (Memory Window)



```
Addr     +0 +1 +2 +3  +4 +5 +6 +7  +8 +9 +A +B  +C +D +E +F
03FF7000 00 00 00 00  10 F0 00 00  20 E0 00 00  30 D0 00 00
03FF7010 40 C0 00 00  50 B0 00 00  60 A0 00 00  70 90 00 00
03FF7020 80 80 00 00  90 70 00 00  A0 60 00 00  B0 50 00 00
03FF7030 C0 40 00 00  D0 30 00 00  E0 20 00 00  F0 10 00 00
03FF7040 00 00 00 00  10 F0 00 00  20 E0 00 00  30 D0 00 00
03FF7050 40 C0 00 00  50 B0 00 00  60 A0 00 00  70 90 00 00
03FF7060 80 80 00 00  90 70 00 00  A0 60 00 00  B0 50 00 00
03FF7070 C0 40 00 00  D0 30 00 00  E0 20 00 00  F0 10 00 00
```

5. 7. 5 Flash Memory Writing Function (When N-Wire CARD connected)

With the ID850QB, the internal flash memory can be written and the load module can be downloaded by the same access method as an ordinary memory operation.

The data on the internal flash memory can be changed from the Memory window, Assemble window, Watch window, Memory Fill dialog box, and Memory Copy dialog box, without having to be aware that the data is that of the internal flash memory. The load module can also be downloaded to the internal flash memory by using the flash self-programming function.

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

Remark With the ID850QB, the remaining area contents after the load module was downloaded to the internal flash memory are erased.

5. 8 Register Manipulation Function

This section explains the following items about register manipulation function.

- Displaying and changing register contents
- Displaying and changing peripheral I/O registers contents
- Displaying and changing I/O port contents

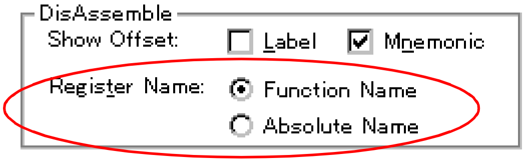
5. 8. 1 Displaying and changing register contents

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.

Display register selection is done in the Register Select dialog box.

Figure 5-16 Absolute Name/Function Name Switching



DisAssemble

Show Offset: □ Label ☑ Mnemonic

Register Name: ● Function Name ○ Absolute Name

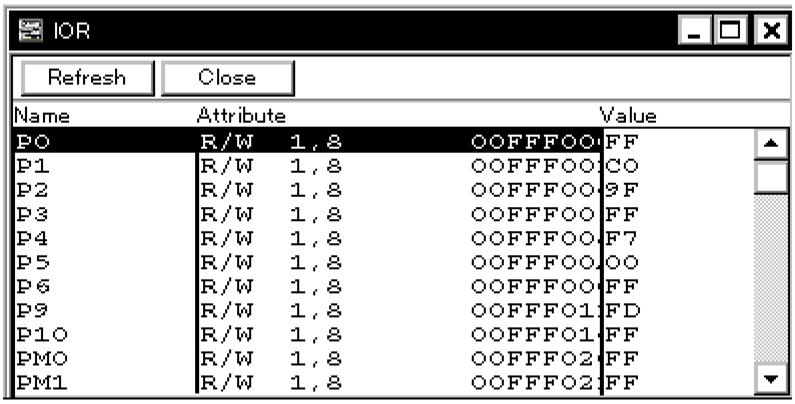
5. 8. 2 Displaying and changing peripheral I/O registers contents

peripheral I/O registers contents can be displayed and changed in the IOR window.

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

Display register selection is done in the IOR Select dialog box.

Figure 5-17 Displaying peripheral I/O registers Contents



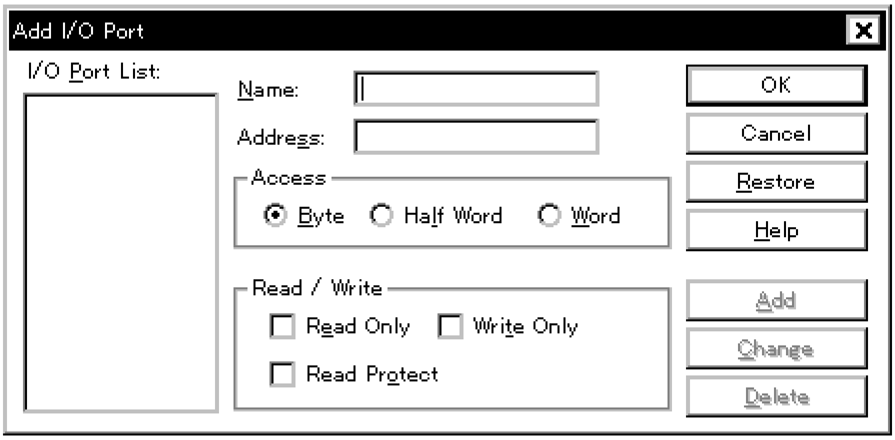
Name	Attribute			Value
P0	R/W	1,8	00FFF00	FF
P1	R/W	1,8	00FFF00	C0
P2	R/W	1,8	00FFF00	9F
P3	R/W	1,8	00FFF00	FF
P4	R/W	1,8	00FFF00	F7
P5	R/W	1,8	00FFF00	00
P6	R/W	1,8	00FFF00	FF
P9	R/W	1,8	00FFF01	FD
P10	R/W	1,8	00FFF01	FF
PM0	R/W	1,8	00FFF02	FF
PM1	R/W	1,8	00FFF02	FF

5. 8. 3 Displaying and changing I/O port contents

User-defined I/O ports can be displayed and changed in the IOR 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 5-18 Register I/O Port



5. 9 Timer Function

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 ID850QB timer function performs measurements using an external clock. Therefore, the measurable time differs based on the setting in "Table 6-10 Relationship Between Division Ratio and Maximum Measurement Time (Timer Count (Timer))" (when IECUBE connected).

- Timer event conditions
- Run-Break event

Remark1 Timer event condition setting/enable/disable/delete operations are possible even during user program execution.

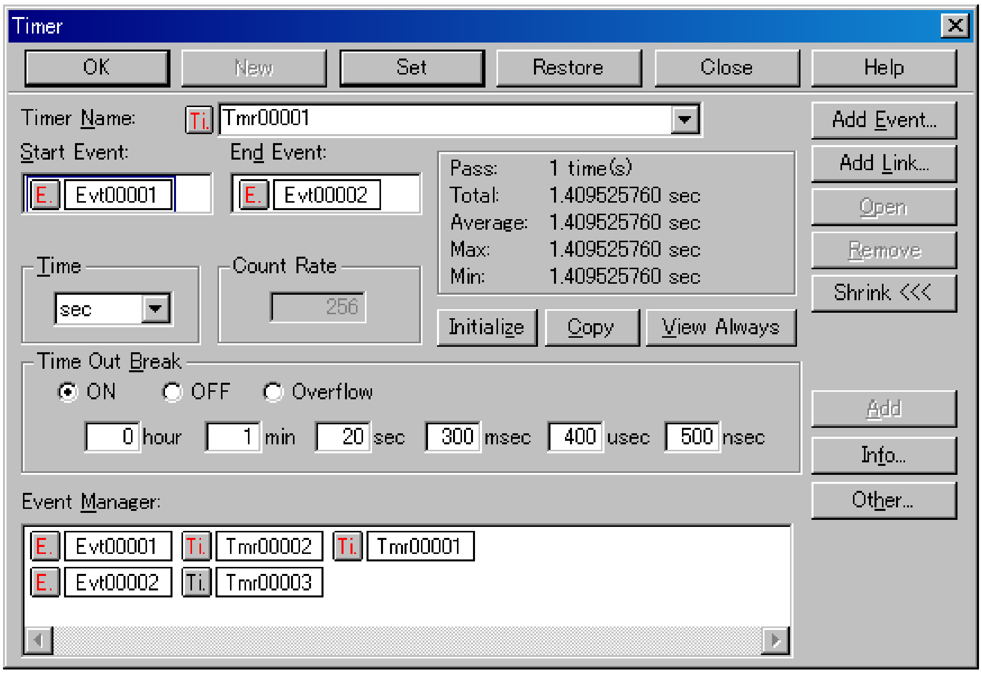
Remark2 When the N-Wire CARD is connected, there is no timer event setting function and only Run-Break time measurement is possible. The Run-Break measurement result is displayed in the status bar of the Main window.

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. 11 Event function".)

In the ID850QB, timeout break settings can be performed in the Time Out Break area.

Figure 5-19 Sets and DisplaysTimer Event (Timer Dialog Box)



Timer Name: Ti Tmr00001

Start Event: E. Evt00001

End Event: E. Evt00002

Pass: 1 time(s)
Total: 1.409525760 sec
Average: 1.409525760 sec
Max: 1.409525760 sec
Min: 1.409525760 sec

Time: sec

Count Rate: 256

Time Out Break: ● ON ○ OFF ○ Overflow

0 hour 1 min 20 sec 300 msec 400 usec 500 nsec

Event Manager:

E. Evt00001 Ti Tmr00002 Ti Tmr00001
E. Evt00002 Ti Tmr00003

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] menu -> [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. 11. 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 Connected)

The trace function of the ID850QB traces the branch instructions of the user program and saves the execution flow to the trace memory. The Trace function is enabled by selecting [Option] menu -> [Trace Function].

In the ID850QB, the DMA (Direct Memory Access) start point and end point are traced regardless of the trace condition (realized/not realized, and within-range/out-of-range) (DMA point trace function priority).

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

Caution This function is used on a mutually exclusive basis with the RRM function (refer to "5. 12 The real-time RAM monitor function").

The RRM function cannot be used when [Option] menu -> [Trace Function] is selected.

When [RRM Function] is selected, the trace data displayed in the Trace window consists only of the branch information and the RRM area range access information.

Remark1 Trace event condition setting/enable/disable/delete operations are possible even during user program execution. In this case, the tracer operation is momentarily stopped during manipulation.

5. 10. 1 Trace memory

ID850QB has trace memory with a ring structure. Size specification is done in the Extended Option dialog box.

The maximum trace memory capacity is as follows.

Table 5-10 Trace memory size

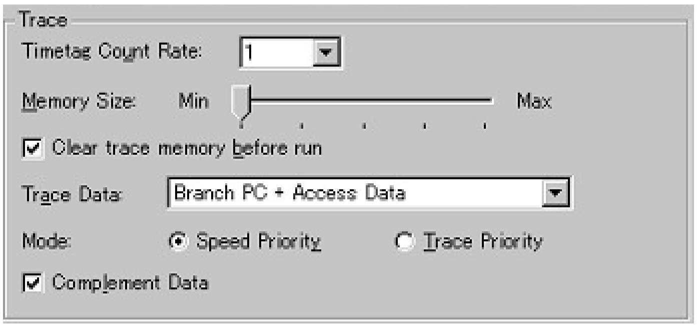
Connected IE	Maximum value
IECUBE	256 KB

5. 10. 2 Setting trace data

The detailed settings for the collected traced data are done in the Extended Option dialog box.

Complement display of instructions between branch instructions that cannot be traced by hardware is possible in the complement mode (enabled by selecting the Complement Data area checkbox). When the complement mode is enabled, assemble display of the internal ROM area is possible during user program execution (while the tracer is stopped).

Figure 5-20 Trace Data Settings (The Extended Option Dialog Dox)



Trace

Timetag Count Rate: 1

Memory Size: Min Max

☑ Clear trace memory before run

Trace Data: Branch PC + Access Data

Mode: ● Speed Priority ○ Trace Priority

☑ Complement Data

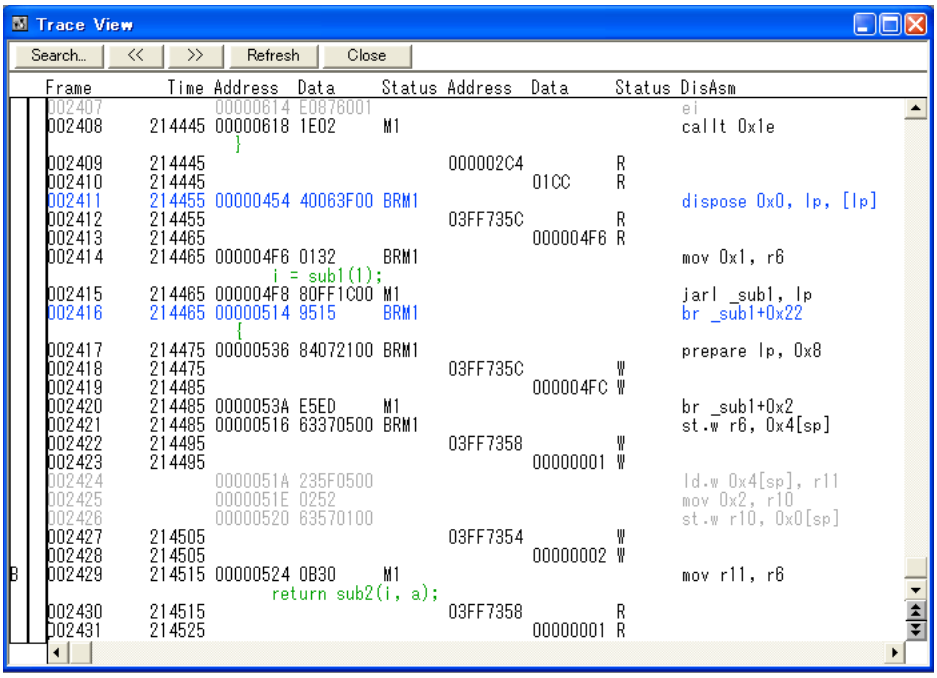
5. 10. 3 Checking trace data

The trace data saved to the trace memory can be checked in the Trace 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] menu -> [Move...].

The display items in the Trace window can be selected in the Trace Data Select dialog box.

Figure 5-21 Checking Trace Data

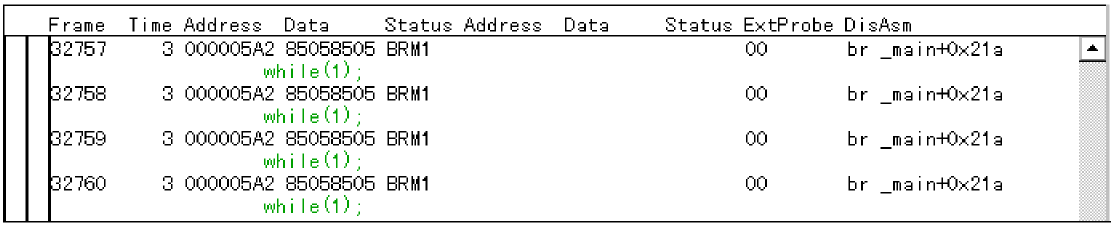


```
  Frame        Time Address  Data     Status Address  Data     Status DisAsm
  002407            00000614 E0876001                                 ei
  002408     214445 00000618 1E02     M1                              callt 0x1e
                       }
  002409     214445                          000002C4          R
  002410     214445                                   01CC     R
  002411     214455 00000454 40063F00 BRM1                            dispose 0x0, lp, [lp]
  002412     214455                          03FF735C          R
  002413     214465                                   000004F6 R
  002414     214465 000004F6 0132     BRM1                            mov 0x1, r6
                          i = sub1(1);
  002415     214465 000004F8 80FF1C00 M1                              jarl _sub1, lp
  002416     214465 00000514 9515     BRM1                            br _sub1+0x22
                       {
  002417     214475 00000536 84072100 BRM1                            prepare lp, 0x8
  002418     214475                          03FF735C          W
  002419     214485                                   000004FC W
  002420     214485 0000053A E5ED     M1                              br _sub1+0x2
  002421     214485 00000516 63370500 BRM1                            st.w r6, 0x4[sp]
  002422     214495                          03FF7358          W
  002423     214495                                   00000001 W
  002424            0000051A 235F0500                                 ld.w 0x4[sp], r11
  002425            0000051E 0252                                     mov 0x2, r10
  002426            00000520 63570100                                 st.w r10, 0x0[sp]
  002427     214505                          03FF7354          W
  002428     214505                                   00000002 W
B 002429     214515 00000524 0B30     M1                              mov r11, r6
                          return sub2(i, a);
  002430     214515                          03FF7358          R
  002431     214525                                   00000001 R
```

5. 10. 4 Mixed display mode (Trace window)

Source file display combined with trace results can be done by selecting [View] menu -> [Mix] in the Trace 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.



```
Frame  Time Address  Data     Status Address  Data     Status ExtProbe DisAsm
32757     3 000005A2 85058505 BRM1                            00       br _main+0x21a
                     while(1);
32758     3 000005A2 85058505 BRM1                            00       br _main+0x21a
                     while(1);
32759     3 000005A2 85058505 BRM1                            00       br _main+0x21a
                     while(1);
32760     3 000005A2 85058505 BRM1                            00       br _main+0x21a
                     while(1);
```

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. 5 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] menu -> [Tracer Start/Tracer Stop].

(1) Operation during execution

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

Table 5-11 Type of Trace Modes

Item	Contents
Unconditional trace	Trace is started when execution of user program, and ends when a break occurs. At this time, the set trace event conditions are ignored.
Conditional trace	Trace is started or stopped by the condition set in the Trace dialog box (refer to "5. 10. 6 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) jarl disp22, [lp] instruction

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

(b) Other instructions

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

(4) Tracer control mode

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

Table 5-12 Types of Tracer Control Mode

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

5. 10. 6 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. 11 Event function").

There are the following types of conditional trace.

Table 5-13 Types of Conditional Trace

Item	Contents, Setting method
Section trace	Executes a trace between two specified conditions (in a specific zone). A section trace can be executed by setting a trace start event and trace end event in the Trace dialog box and selecting [Run] menu -> [Cond. Trace ON].
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] menu -> [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] menu -> [Cond. Trace ON]. In this case, select [Delay Trigger Stop] .

5. 11 Event function

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

In ID850QB, such events are used as action triggers for each debugging function, such as break and trace.

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

5. 11. 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 5-14 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 Connected)"). -> Trace dialog box
Timer event	Ti	Condition for specifying the time measurement start timing and stop timing (refer to "5. 9 Timer Function"). -> Timer dialog box

5. 11. 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 the 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.

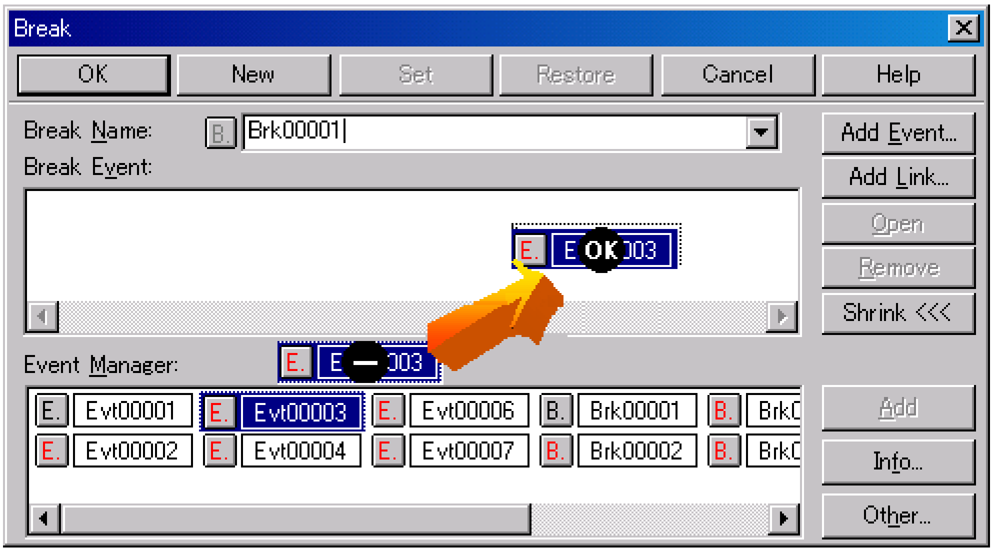
5. 11. 3 Setting event conditions

Various event conditions listed in Table 5-14 are individually created in the corresponding dialog box.

(1) Setting of various event conditions

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

Figure 5-22 Setting of Various Event Conditions



Break Name: B. Brk00001

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 and 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. 11. 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 or event link conditions exceed the maximum number that can be used simultaneously, it is necessary to disable the set various event conditions once and then register them again. (Refer to "5. 11. 5 Managing events".)

Table 5-15 Number of enabled events for each event condition

Connected IE		Event		Event Link	Break	Trace	Timer
		Execute	Access				
IECUBE		10[a]	6[b]	1	10+6	1	7
N-Wire CARD	Nx85ET（RCU0+TEU+TRCU）	10[a]	4[c]	1[d]	10+4	-	-
	Nx85E901（RCU0），RCU1	2[e]		1[f]	2[e]	-	-

a. 2 before executions (usable only for breaks, address range not specifiable), 8 post-execution events (4 when address range is specified, because 2 events are used for range specification)

b. 3 when address range is specified, because 2 events are used for range specification

c. 2 when address range is specified, because 2 events are used for range specification

d. Use from Phase 1 to Phase 4

e. Address range specification is not possible.

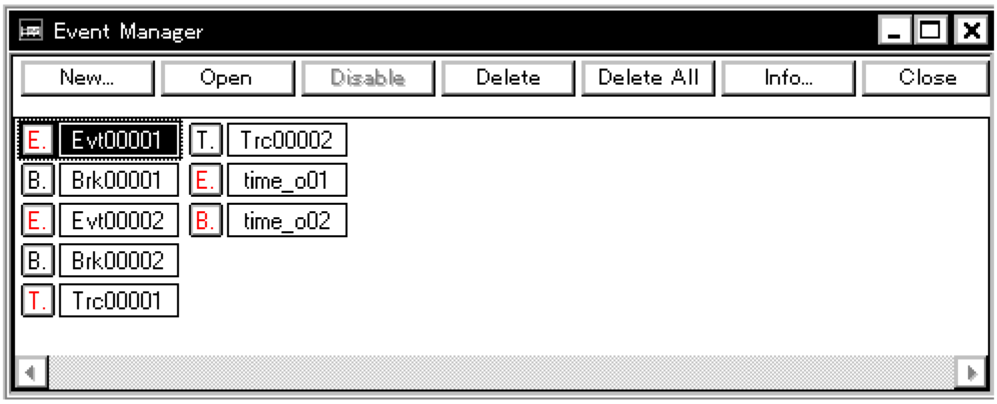
f. Can be set only for Phase 1 and Phase 2

5. 11. 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 (the Event Manager)



(1) Event icons

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

Enable/disable is switched by clicking the mark part.

Table 5-16 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 valid.
	B.T.Ti.	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 invalid.
	B.T.Ti.	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.	Indicates that the Various Event Conditions is held pending. The various events do not occur even when its condition is satisfied.

5. 12 The real-time RAM monitor function

The real-time RAM monitor function (RRM function) displays in real time the variables allocated to the sampling range, data, etc., in the Watch window or the Memory window.

Access statuses (read, write, read & write) are displayed using different colors in the Memory window using this function (refer to "5. 7. 4 Access monitor function (when IECUBE connected)").

The RRM function is enabled by selecting [Option] menu -> [RRM Function].

- Sampling range setting
- Sampling time setting
- RAM monitor function

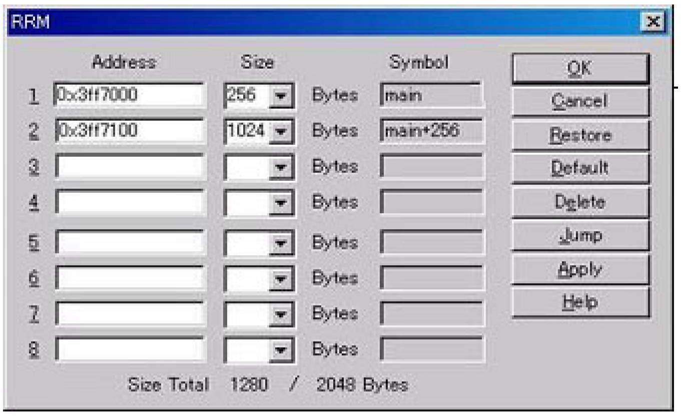
Caution This function is used on a mutually exclusive basis with the trace function (refer to "5. 10 Trace Function (When IECUBE Connected)").The trace function cannot be used when [Option] menu -> [RRM Function] is selected.

5. 12. 1 Sampling range setting

The sampling range for the RRM function is set in the RRM dialog box.

Up to 8 locations can be set in 256-byte units in the 2 KB RRM area.

Figure 5-24 Sampling Range Setting (RRM Dialog Box)



RRM

	Address	Size		Symbol
1	0x3ff7000	256	Bytes	main
2	0x3ff7100	1024	Bytes	main+256
3			Bytes	
4			Bytes	
5			Bytes	
6			Bytes	
7			Bytes	
8			Bytes	

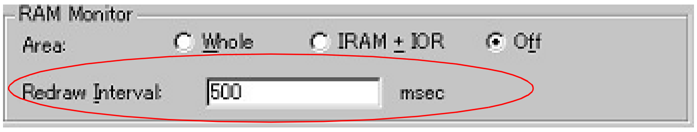
Size Total 1280 / 2048 Bytes

5. 12. 2 Sampling time setting

Setting of the sampling time for the RRM function is done in the Extended Option dialog box.

Values are updated at the specified time interval.

Figure 5-25 Sampling Time Setting



RAM Monitor

Area: ○ Whole ○ IRAM + IOR ● Off

Redraw Interval: 500 msec

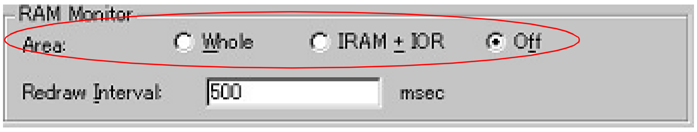
5. 12. 3 RAM monitor function

The RAM monitor function realizes the RRM function through software emulation.

There are no sampling range restrictions, but user program execution is temporarily paused during sampling. Therefore, the variables and data allocated to this range are displayed almost in real time.

The RAM monitor function settings are performed in the Extended Option dialog box.

Figure 5-26 RAM Monitor Function Setting



RAM Monitor

Area: ○ Whole ○ IRAM + IOR ● Off

Redraw Interval: 500 msec

5. 13 Load/Save Function

ID850QB 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. 14. 1 Active Status and Static Status".)

5. 13. 1 Debugging environment (project file)

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

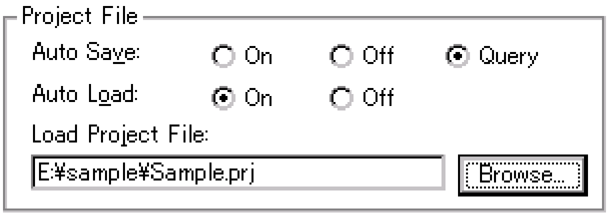
A project file is created when the debugging environment at a particular point in time is saved, and that debugging environment can be restored by loading this file at a subsequent time.

Loading and saving project files are done in the Project File Save dialog box and the Project File Load dialog box, respectively. To load a project file at startup, press the <Project...> button in the Configuration dialog box.

(1) Automatic save/automatic load of project file

The project file to be loaded/saved when performing startup and end in the Debugger Option dialog box can also be set in advance.

Figure 5-27 Automatic Save/Automatic Load Setting for Project File



Project File

Auto Save: ○ On ○ Off ● Query

Auto Load: ● On ○ Off

Load Project File: E:¥sample¥Sample.prj

(2) Contents saved to project File

The following contents are saved to the project file:

Table 5-17 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, trace on/off information
Download dialog box	File information to be downloaded
Extended Option dialog box Debugger Option dialog box Fail-safe Break dialog box RRM dialog box	Set information
Source window	Display information of window
Assemble window Memory window	Display information of window, display start address
Stack Trace window IOR window Local Variable window Trace window Event Manager Console window Expansion window	Display information of window
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
Register window	Display information of window
Watch window	Display information of window, 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

5. 13. 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 static status) is displayed and the display information at the time of saving is displayed.

Loading and saving of view files are done in the View File Load dialog box and the View File Save dialog box, respectively.

Table 5-18 Type of the View Files

File Type	Target Window File Name
Source Text (*.svw)	Source window View file of the Source window
Assemble (*.dis)	Assemble window View file of the Disassemble window
Memory (*.mem)	Memory window View file of the Memory window
Watch (*.wch)	Watch window View file of the Watch window
Register (*.rgw)	Register window View file of the Register window
I/O Register (*.ior)	IOR window View file of the IOR window
Local Variable (*.loc)	Local Variable window View file of the Local Variable window
Stack Trace (*.stk)	Stack Trace window View file of the Stack Trace window
Trace (*.tvw)	Trace window View file of the Trace window
Console (*.log)	Console window View file of the Console window
All (*.*)	All files
Source (*.c, *.s) (During downloading)	Source file**Note**
Text (*.txt)	Text file

Note The extension of the source file can be changed in the Extended Option dialog box.

5. 13. 3 Window setting information (setting file)

A setting file is a file that records the window setting information (watch data settings, peripheral I/O registers 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.

Loading and saving of setting files are done in the Environment Setting File Load dialog box and the Environment Setting File Save dialog box, respectively.

Table 5-19 Type of the Setting Files

File Type	Target Window File Name
Watch (*.wch)[Note]	Watch window Setting file of the Watch window
I/O Register (*.ior)[Note]	IOR window Setting file of the IOR window
Event (*.evn)	Event Manager Setting file of event

Note A variable value can be loaded.

5. 14 Functions Common to Various Windows

The target windows have the following common functions.

- Active Status and Static Status
- Jump function
- Trace Result with Linking Window (when IECUBE connected)
- Drag & drop function

5. 14. 1 Active Status and Static Status

The Source window that is displaying the source file to which symbol information is read, Assemble window and Memory window has two statuses:

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

Select this status by the [Window] menu.

(1) Active status

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

This window is also the jump destination of the jump function. If this window is linked with the Trace window, the contents displayed in the active window are updated in association with the Trace 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 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. 14. 2 Jump function

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

You can jump among the above windows, or from Trace window, Stack Trace 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.

Source window -> Select [Source Text]

Assemble window -> Select [Assemble]

Memory window -> Select [Memory]

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

(2) Details of jump source address

The details of jump source address is as follows:

Table 5-20 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

Table 5-20 Details of Jump Source Address

Target Window	Details of Jump Pointer	
From the Stack Trace 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 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 Assemble window	Fetch address	

5. 14. 3 Trace Result with Linking Window (when IECUBE connected)

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

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

(1) Linking method

1) Set the Trace window as the current window.
2) Select [View] menu -> [Window Synchronize] to select a window to be linked.
   Source window -> Select [Source Text]
   Assemble window -> Select [Assemble]
   Memory window ->Select [Memory]
3) Move the cursor to the line to be linked in the trace result display area of the Trace 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.

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

5. 14. 4 Drag & drop function

Line numbers, addresses, and text highlighted by selecting them 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 5-21 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. An external sense data condition and path count are 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. An external sense data condition and path count are 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 5-22 Details of Drag & Drop Function (Character String)

<table>
<tr><th>Window/Area to Drop to</th><th colspan="2">Operation After Drop</th></tr>
<tr><td rowspan="7">The Event Manager or the event manager area in each various event setting dialog box</td><td colspan="2">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, an external sense 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:</td></tr>
<tr><td><b>Symbols</b></td><td><b>Status</b></td></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>

Table 5-22 Details of Drag & Drop Function (Character String)

<table>
<tr><th>Window/Area to Drop to</th><th colspan="2">Operation After Drop</th></tr>
<tr><td rowspan="7">Condition setting area in each various event setting dialog box (other than address and data setting areas)</td><td colspan="2">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, an external sense 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:</td></tr>
<tr><td><b>Symbols</b></td><td><b>Status</b></td></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>
<tr><td>Condition setting area in each various event setting dialog box (address and data setting areas)</td><td colspan="2">The dropped text is set in the area.</td></tr>
<tr><td>Watch window</td><td colspan="2">If the dropped text is recognizable as a symbol, the contents of the symbol are displayed.</td></tr>
</table>

Caution The various event setting dialog boxes include the Event dialog box, Event Link dialog box, Break dialog box, Trace dialog box and Timer dialog box.

CHAPTER 6 WINDOW REFERENCE

This chapter explains in detail the functions of the windows and dialog boxes of ID850QB.

- Window List
- Explanation of Windows

6. 1 Window List

The Table 6-1 lists the windows of the ID850QB.

Table 6-1 Window List

Window Name	Contents
Main window	This window is displayed first, when the ID850QB is started. It controls execution of the user program. Various windows are opened from this window.
Configuration dialog box	Displays and sets the ID850QB operation environment.
Extended Option dialog box	Displays and sets the extended options of the ID850QB.
Fail-safe Break dialog box	Sets the fail-safe breaks (IECUBE)
RRM dialog box	Sets the RRM sampling range
Debugger Option dialog box	Displays and sets other options.
Project File Save dialog box	Saves the current debug environment to project file
Project File Load dialog box	Loads the debug environment.
Download dialog box	Loads an object file and binary file.
Upload dialog box	Saves the memory contents to a file.
Load Module List dialog box	Lists the names of the downloaded load module files.
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 online 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

Table 6-1 Window List

Window Name	Contents
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 Trace window	Displays the current stack contents
Memory window	Display the contents of memory.
Memory Search dialog box	Searches in the Memory window
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 result of comparing the memory.
DMM dialog box	Sets addresses and data subject to DMM.
Register window	Displays the contents of registers.
Register Select dialog box	Selects registers to be displayed in the Register window.
IOR window	Displays the contents of peripheral I/O registers
IOR Select dialog box	Selects peripheral I/O registers and I/O ports to be displayed in the IOR window
Add I/O Port dialog box	Registers an I/O port to be displayed in the IOR window.
Timer dialog box	Registers and sets timer event conditions, and displays execution time measurement result, and . (IECUBE)
Timer Result dialog box	Displays execution time measurement results. (IECUBE)
Trace 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 window. (IECUBE)
Trace move dialog box	Specifies the start address to display the contents of the Trace 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)
Event Manager	Displays, enables/disables, and deletes each event condition.
Software Break Manager	Display, enable or disable, and delete software breaks.
Event dialog box	Registers event conditions.
Event Link dialog box	Registers event link conditions.
Break dialog box	Registers and sets break event conditions.

Table 6-1   Window List

Window Name	Contents
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 ID850QB,CPU, and symbol information.
Exit Debugger dialog box	Terminate the ID850QB.
About dialog box	Displays the version of the ID850QB.
Console window	Inputs commands.
Font dialog box	Displays the types of fonts displayed.
Browse dialog box	Selects the file to be set

6. 2 Explanation of Windows

This section explains each window or dialog box as follows:

Window Name / Dialog box Name

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

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 ID850QB is started up and initialized.

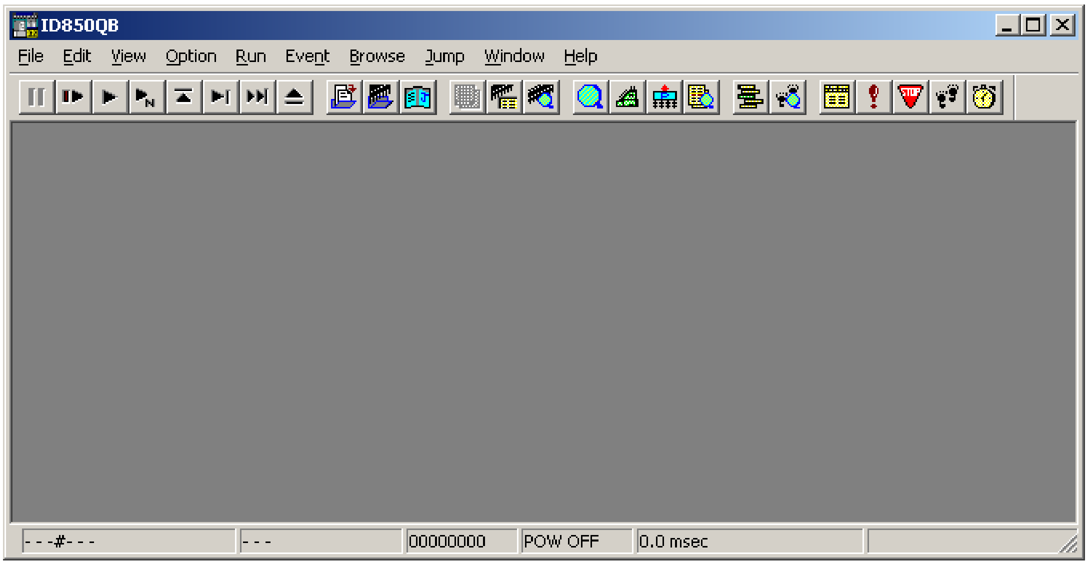
In ID850QB, other windows are manipulated from this window (refer to "Table 6-1 Window List").

Execution of the user program is controlled in this window.

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

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

Figure 6-1 Main Window



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

Menu bar

(1) [File] menu

(2) [Edit] menu

(3) [View] menu

(4) [Option] menu

(5) [Run]menu

(6) [Event] menu

(7) [Browse] menu

(8) [Jump] menu

(9) [Window] menu

(10) [Help] menu

(1) [File] menu

Open...		Loads a view file, source file, or text file. Opens the 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.
Load Module...		Lists the names of the files that have been downloaded. Opens the Load Module List 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 ID850QB.
	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 ID850QB. Opens the Reset Debugger dialog box.
Exit		Terminate the ID850QB. 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.
	Modify...	Rewrites the memory contents during user program execution. Opens the DMM dialog box.
Edit Source		Opens the source file displayed in the active Source window with the editor specified by the PM plus when the PM plus 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 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 Watch window.
Delete Watch	Deletes the selected watch point from the Watch window.This menu is valid only when a variable is selected in 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 status bar is displayed (default) or not.
Button	Selects whether the buttons on each window are displayed (default) or not.
Source Mode	Executes step execution at the source level (in line units).
Instruction Mode	Executes step execution at the instruction level (in instruction units).

Auto Mode	Automatically selects step execution at the source level or step execution at the instruction level (default). Step execution is performed at the source level (in a mode other than mixed display mode) if 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.
Extended Option...	Sets extended options. Opens the Extended Option dialog box.
RRM Setting...	Sets the sampling range for the real-time RAM monitor function (RRM function). Opens the RRM dialog box (when IECUBE connected).
Debugger Option...	Sets ID850QB 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 window is active (when IECUBEconnected).
RRM Function	Selects the real-time RAM monitor function (RRM function) (used on a mutually exclusive basis with the trace function) (when IECUBE connected). When the RRM function is selected, the data displayed in the Trace window consists only of the branch information and the RRM area range access information. Also, when this function is selected, the menus related to tracing become invalid.
Trace Function	Selects the trace function (used on a mutually exclusive basis with the RRM function) (when IECUBE connected). When the trace function is selected, the RRM function cannot be used.

(5) [Run]menu

Restart	Resets the CPU and executes the program. ▮▶ Same operation as the ReGo button
Stop	Forcibly stops program execution. ❚❚ Same operation as the Stop button
Go	Executes the program from the current PC. ▶ Same operation as the Go button
Ignore break points and Go	Ignores break points being set, and executes the program. ▶N Same operation as the GoN button
Return Out	The user program is executed until execution returns. **Note:** This command is used for a function described in C language. ▲ Same operation as the Ret button
Step In	Executes the instructions in the program one by one (step execution). If a function or subroutine is called, its instructions are executed one by one. ▶❘ Same operation as the Step button

Next Over	Executes the instructions in the program one by one (Next step execution). If a function or subroutine is called, its instructions are not executed on a step-by-step basis. ⏭ Same operation as the Over button
Start From Here	Executes the program from the cursor position on Source window or Assemble window .
Come Here	Executes the program from the current PC to the cursor position in the Source window or 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 the Go each time a break has occurred.
Slowmotion	Continues step execution. Each time step execution has been performed, the window is updated and then step execution is performed again. ▶❘ Same operation as clicking the Step each time a break has occurred.
CPU Reset	Resets the CPU. ⏏ Same operation as the Res button
Change PC	Sets the address at the cursor position in the Source window or Assemble window to the PC.
Break Point	Sets or deletes a breakpoint at the cursor position in the Source window or Assemble window.
Software Break Point	Sets or deletes a software breakpoint at the cursor position in the Source window or Assemble window.
Delete All Breakpoints	Deletes all the set software break events.
Uncond. Trace ON	Validates unconditional trace so that trace can always be executed during program execution (default). (When IECUBE connected). At this time, the set trace event conditions are ignored. The trace mode cannot be changed while the tracer is activated.
Cond. Trace ON	Validates conditional trace and traces in accordance with the trace event condition during program execution (when IECUBE connected). The trace mode cannot be changed while the tracer is activated.
Tracer Control Mode	Sets trace control mode. (when IECUBE 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.

Timer Start/Timer Stop	Starts timer measurement when it is stopped, or stops it when it is in progress (when IECUBE 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 connected) . This item is invalid if the program is not being executed. Immediately after program execution has been started , the tracer is executed.

(6) [Event] menu

Event Manager	Manages various event conditions. Opens the Event Manager. Same operation as the Mgr 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 the Evn button
Event Link...	Registers an event link condition. Opens the Event Link dialog box.
Break...	Registers and sets a break event condition. Opens the Break dialog box. Same operation as the Brk button
Trace...	Registers and sets a trace event condition (when IECUBE connected) . Opens the Trace dialog box. Same operation as the Trc button
Timer...	Registers and sets a timer event condition (when IECUBE connected) . Opens the Timer dialog box. Same operation as the Tim button
Delay Count...	Sets a delay count and displays the delay value (when IECUBE 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 the Src 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 the Asm 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 the Mem button

Watch	Displays the watch contents. Opens the Watch window. Same operation as the Wch button
Register	Displays the register contents. Opens the Register window. Same operation as the Reg button
I/O Register	Displays the contents of the Peripheral I/O registers. Opens the IOR window. Same operation as the IOR button
Local Variable	Displays the local variable. Opens the Local Variable window. Same operation as the Loc button
Stack Trace	Displays the stack trace results. Opens the Stack Trace window. Same operation as the Stk button
Trace	Displays the trace results (when IECUBE connected) . Opens the Trace window. Same operation as the TrW button
Console	Opens the Console window.
Others	Displays other windows. 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 an active Source window 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 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 jump destination address specified by the data value selected in the current window. Opens the Memory window. If an active Memory window is open, that window is displayed in the forefront (so that it can be manipulated).

(9) [Window] menu

New Window	Opens a new window displaying the same contents as those of the current window.This menu is valid only when the current window is Source window, Assemble window or Memory window.
Cascade	Cascade display of the windows in the Main window.
Tile	Tile display of the windows in the Main window.
Arrange Icons	Rearranges the icons in the Main window.

Close All	Closes all windows, except the Main window.
Refresh	Updates the contents of the window with the latest data.
Active	Sets the window in the active status.
Static	Sets the window in the static status.
(Open Window)	Lists the windows that are open. The window with the check mark shown on the side of the figure is the current window. By selecting a window name, the selected window is used as the current window.

(10) [Help] menu

ID850QB Help	Displays the help of ID850QB.
Command Reference	Opens the Help window of COMMAND REFERENCE.
Main Window Help	Displays the help of the Main window.
Current Window Help	Displays the help of the current window.
About...	Displays the version of the ID850QB. Opens the About dialog box.

Toolbar

(1) Meaning of each button

(2) Operation of Toolbar

(1) Meaning of each button

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

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 break points being set, and executes the user program. Same function as [Run] menu -> [Ignore break points and Go] .
Ret	The user program is executed until execution returns Same function as [Run] menu -> [Return Out] . **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 -> [Watch] .
Reg	Displays the register contents. Opens the Register window. Same function as [Browse] menu -> [Register] .
IOR	Displays the contents of the Peripheral I/O registers. Opens the IOR window. Same function as [Browse] menu -> [I/O Register] .
Loc	Displays the local variable contents. Opens the Local Variable window. Same function as [Browse] menu -> [Local Variable] .
Stk	Displays the current stack contents. Opens the Stack Trace window. Same function as [Browse] menu -> [Stack Trace] .
TrW	Displays the trace results (when IECUBE connected) . Opens the Trace window. Same function as [Browse] menu -> [Trace] .
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 connected) . Opens the Trace dialog box. Same function as [Event] menu -> [Trace...].
Tim	Registers and sets a timer event. (when IECUBE 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)

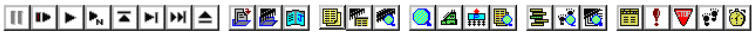
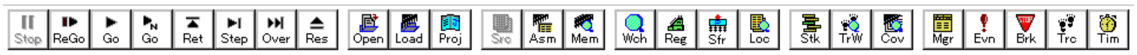


Figure 6-3 Toolbar (Picture and Text)



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.

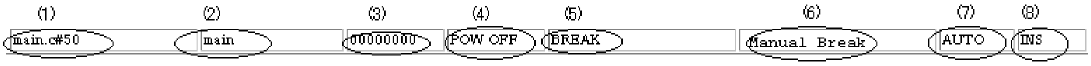
Status bar

The status bar displays the status of the ID850QB and in-circuit emulator.

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

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

Figure 6-4 Status Bar



(1) Program name	Displays the program file name indicated by the PC value.
Source name	Displays the source file name indicated by the PC value.
Line number	Displays the line number indicated by the PC value.
(2) Function name	Displays the function name indicated by the PC value.
(3) PC value	Displays the current PC value.
(4) CPU status	Refer to "Table 6-2 CPU status".
(5) 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/IDLE	Software stop mode, hardware stop mode or idle mode
HOLD	Bus hold 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	Tracer operating (when IECUBE connected) .
TIM	Timer operating (when IECUBEconnected)
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) (refer to "Table 6-16 Measurable Values") .
TIMER OVERFLOW	Measurement result overflowed.

Table 6-4 Break Cause

Display	Meaning
Manual Break	Forced break
Temporary Break	Temporary break
Event Break	Break by event
Software Break	Software break
Trace Full Break	Break due to trace full (when IECUBE connected)
Non Map Break	Non-mapped area is accessed (when IECUBE connected) .

Table 6-4   Break Cause

Display	Meaning
Write Protect	An attempt has been made to write to a write-protected area (when IECUBE connected) .
IOR Illegal (xxx)	Illegal access to peripheral I/O register was performed (when IECUBE connected). xxx indicates the corresponding address.
Timer Over Break	Execution time-over detected (when IECUBE connected)
Flash Macro Service	Flash macro service in progress (when N-Wire CARD connected)
IRAM Write Protect (xxx xxx)	During break, performed verify check of IRAM guarded area and rewrote value (when IECUBE is connected). xxx xxx indicates the relevant address and data (in case of multiple items, only first item is displayed).

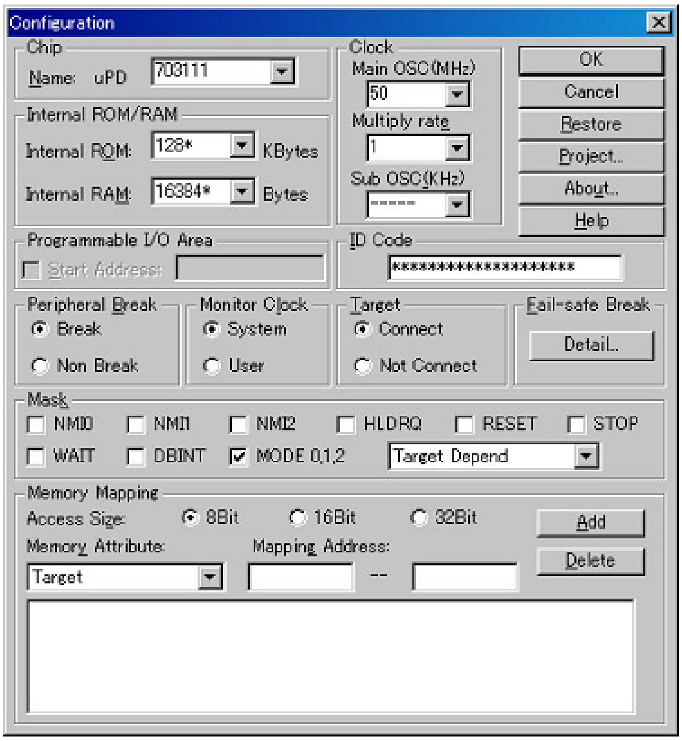
Configuration dialog box

This dialog box is used to display and set the ID850QB operation environment. (Refer to "5. 1 Setting debugging environment".)

This dialog box is automatically displayed after the ID850QB 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. 13. 1 Debugging environment (project file)".)

Figure 6-5 Configuration Dialog Box (When IECUBE connected)



Chip
Name: uPD 703111

Internal ROM/RAM
Internal ROM: 128* KBytes
Internal RAM: 16384* Bytes

Clock
Main OSC(MHz) 50
Multiply rate 1
Sub OSC(KHz) ------

Programmable I/O Area
□ Start Address:

ID Code
****

Peripheral Break
● Break
○ Non Break

Monitor Clock
● System
○ User

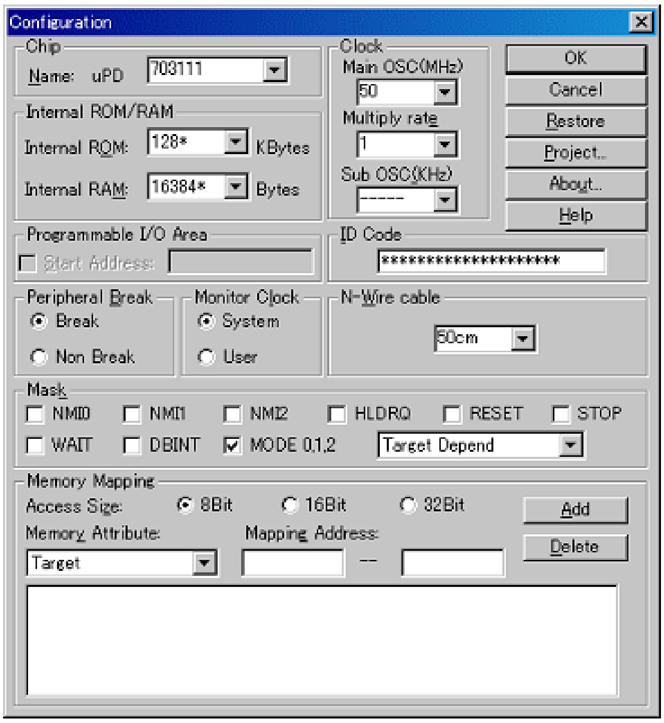
Target
● Connect
○ Not Connect

Fail-safe Break

Mask
□ NMI0 □ NMI1 □ NMI2 □ HLDRQ □ RESET □ STOP
□ WAIT □ DBINT ☑ MODE 0,1,2 Target Depend

Memory Mapping
Access Size: ● 8Bit ○ 16Bit ○ 32Bit
Memory Attribute: Target
Mapping Address: --

Figure 6-6 Configuration Dialog Box (When N-Wire CARD connected)



Configuration

Chip
Name: uPD 703111

Internal ROM/RAM
Internal ROM: 128* KBytes
Internal RAM: 16384* Bytes

Clock
Main OSC(MHz) 50
Multiply rate 1
Sub OSC(KHz) -----

Programmable I/O Area
□ Start Address:

ID Code
****

Peripheral Break
● Break
○ Non Break

Monitor Clock
● System
○ User

N-Wire cable
50cm

Mask
□ NMI0 □ NMI1 □ NMI2 □ HLDRQ □ RESET □ STOP
□ WAIT □ DBINT ☑ MODE 0,1,2 Target Depend

Memory Mapping
Access Size: ● 8Bit ○ 16Bit ○ 32Bit
Memory Attribute: Target
Mapping Address: --

- Opening
- Explanation of each area
- Function buttons

Opening

(Automatically when the ID850QB is started up)

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

Explanation of each area

(1) Chip

(2) Internal ROM/RAM

(3) Clock

(4) Programmable I/O Area

(5) ID Code

(6) Peripheral Break

(7) Monitor Clock

(8) N-Wire cable

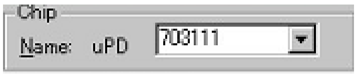
(9) Target

(10) Fail-safe Break

(11) Mask

(12) Memory Mapping

(1) Chip



Chip

Name: uPD 703111

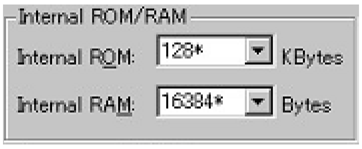
This area is used to select the chip name.

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



Internal ROM/RAM

Internal ROM: 128* KBytes

Internal RAM: 16384* Bytes

This area is used to set the size of the internal RAM and internal ROM of the CPU. (Refer to "5. 1. 3 Mapping settings".)

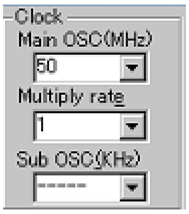
A mapping size is selected from the drop-down list.

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

Area	Meaning	Settable range
Internal ROM	Sets the Internal ROM size (When IECUBE connected)	0,8,32,64,128,256,512,1024 (KB)
	Displays the Internal ROM size (When N-Wire CARD connected)	-
Internal RAM	Sets the Internal RAM size	4096,12288,28672,61440 bytes

(3) Clock



Clock

Main OSC(MHz) 50

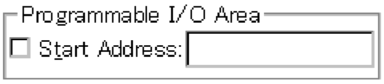
Multiply rate 1

Sub OSC(KHz) -----

Settings related to the main clock and subclock are performed here.

Main OSC (MHz)	Specifies a frequency before the main clock is multiplied. A frequency can be selected from the drop-down list (5.000, 8.000, 13.500, or 18.000) or directory input.
Multiply rate	Specifies the main clock multiplication rate. This value can be selected from the drop-down list (1 to 10), or directly input.
Sub OSC (KHz)	Specifies the subclock frequency (setting this parameter is disabled for types that do not have a subclock). A frequency can be selected from the drop-down list or directory input.

(4) Programmable I/O Area



Programmable I/O Area

□ Start Address:

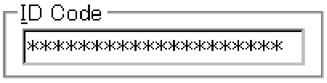
This area is used to specify usage of the programmable I/O area and the start address.

The start address of the programmable I/O area can be specified, only if the device selected by (1) Chip supports the programmable I/O area. The start address of the programmable I/O area can be input by checking the check box when the programmable I/O area is used. The address is aligned to 16 KB.

Caution To use a programmable I/O area or extended I/O area, mapping that area as a target is required.

Remark In the case of a device with an extended I/O area with fixed addresses, the device selected by (1) Chip , Setting of this area is performed automatically.

(5) ID Code



ID Code

****

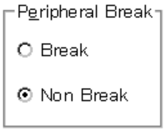
This area is used to input the ID code to be used when the code on the internal ROM or internal flash memory is read by ID850QB (ID code authentication) (When N-Wire CARDconnected)

This area does not have to be set with a ROMless product or a product without a security 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 ID850QB is forcibly terminated.

For the details of ID code authentication, refer to "N-Wire CARDUser's Manual".

(6) Peripheral Break



Peripheral Break

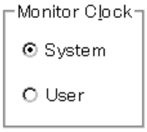
○ Break

● Non Break

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

Break	Stopped
Non Break	Not stopped (default)

(7) Monitor Clock



Monitor Clock

● System

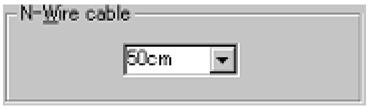
○ User

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). **Caution** In theID850QB, 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.

(8) N-Wire cable



N-Wire cable

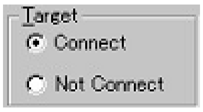
50cm

This area is used to select the length of the PC interface cable that connects the in-circuit emulator and host machine. (When N-Wire CARD connected)

Caution If the actual PC interface cable length and the value set in this area do not match, the ID850QB will not start (for example, if [50 cm] is selected when a 2-meter cable is connected).

50cm	50 cm cable connection (default) DCK = 20 MHz
2m	2 m cable connection DCK = 10 MHz **Caution** When 2 m is selected, the maximum value of the execution time measurement function becomes double, and the resolution becomes half.

(9) Target



Target

● Connect

○ Not Connect

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

Remark It is used to detect an illegal power supply status.

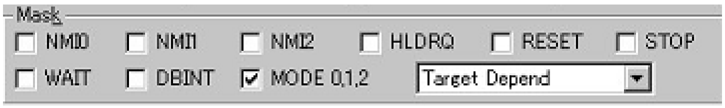
The default is determination by detecting the power (TVDD) of the target.

Connect	be connected
Not Connect	not be connected

(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 IECUBEconnected) .

(11) Mask



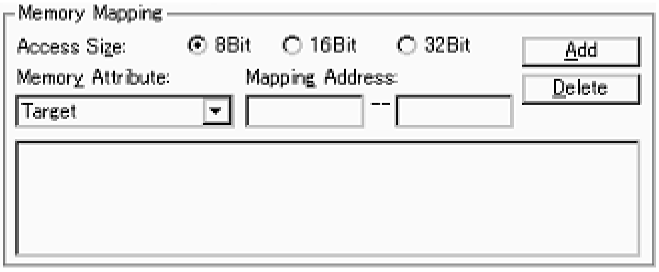
This area is used to mask the signal sent from the target.

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

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

Remark If a device file supporting TM tag is used, however, select Mode00 to Mode1F (the modes to be displayed are determined by the definition of the device file). When the IECUBE is connected and when the target is connected, Target Depend can be selected.

(12) Memory Mapping



access size, memory attribute, and address.

(a) Access Size

Selects memory access size.

This setting is used to specify the access size on the ID850QBsoftware; the operation of the external bus hardware is set in accordance with the settings of the MODE pin and I/O register.

8Bit	Accesses memory with ld.b instruction/st.b instruction.
16Bit	Accesses memory with ld.h instruction/st.h instruction.
32Bit	Accesses memory with ld.w instruction/st.w instruction.

(a) Memory Attribute

The following mapping attributes can be selected. Select a mapping attribute according to the usage. (Refer to "5. 1. 3 Mapping settings".)

Table 6-6 Type of Mapping Attribute

mapping attribute	Meaning
Target	Selects the target memory.
I/O Protect[Note1]	Selects the I/O protect area. (This area can be set only in the area set as Target .)

Note1 The area set as I/O Protect is not read unless it is registered to the IOR 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 and (b) Mapping Address is set and displayed.

Remark Moreover, mapping unit adjustment is performed except I/O Protect, and if the mapping units is not matched, the minimum settable range including the specified address becomes the mapping target.

Table 6-7 Mapping Unit

Mapping Area	Mapping Unit
Target	1Mbyte units
I/O Protect Stack	1-byte units

Caution If the external memory is mapped, change the value of a register required for access of an external memory.(refer to "5. 1. 4 To change the value of a register required for access of an external memory").

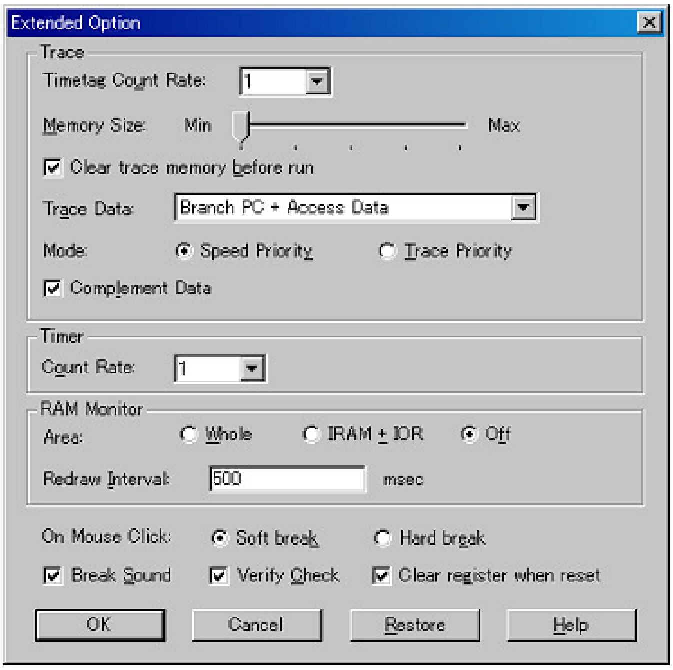
Function buttons

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 environment settings status 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 ID850QB 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 ID850QB. (Refer to "5. 1 Setting debugging environment".)

Figure 6-7 Extended Option Dialog Box



Extended Option

Trace

Timetag Count Rate: 1

Memory Size: Min Max

☑ Clear trace memory before run

Trace Data: Branch PC + Access Data

Mode: ● Speed Priority ○ Trace Priority

☑ Complement Data

Timer

Count Rate: 1

RAM Monitor

Area: ○ Whole ○ IRAM + IOR ● Off

Redraw Interval: 500 msec

On Mouse Click: ● Soft break ○ Hard break

☑ Break Sound ☑ Verify Check ☑ Clear register when reset

- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

(1) Trace

(2) Timer

(3) RAM Monitor

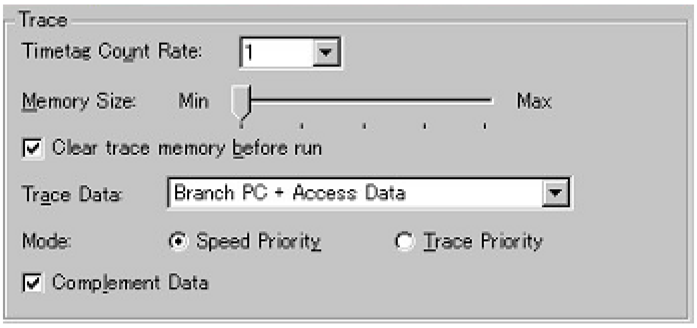
(4) On Mouse Click

(5) Break Sound

(6) Verify Check

(7) Clear register when reset

(1) Trace



Trace

Timetag Count Rate: 1

Memory Size: Min Max

☑ Clear trace memory before run

Trace Data: Branch PC + Access Data

Mode: ● Speed Priority ○ Trace Priority

☑ Complement Data

This area is used to set about trace (refer to "5. 10 Trace Function (When IECUBE Connected)") (when IECUBEconnected).

(a) Timetag Count Rate:

This area is used to set the division ratio of the counter used for time tag display in Trace window.

This division ratio is selected from a drop-down list.

If the division ratio is set, the number of clocks necessary for counting up the counter displayed for time tag is changed.

The relationship between the time tag counter division ratio and maximum measurement time is as follows.

Table 6-8 Relationship Between Division Ratio and Maximum Measurement Time (Time Tag Counter (Trace))

Division Ratio	Resolution (ns)	Maximum Measurable Time	Remark
TMCLK (1/1)	20	1.4 minutes	Timetag counter is 32 bits, In case of 50 MHz external clock
TMCLK (1/2)	40	2.8 minutes	
TMCLK (1/4)	80	5.7 minutes	
TMCLK (1/8)	160	11.4 minutes	
TMCLK (1/16)	320	22.8 minutes	
TMCLK (1/32)	640	45.6 minutes	
TMCLK (1/64)	1280	1.5 hours	
TMCLK (1/128)	2560	3 hours	
TMCLK (1/256)	5120	6 hours	
TMCLK (1/512)	10240	12.2 hours	
TMCLK (1/1024)	20480	24.4 hours	
TMCLK (1/2048)	40960	48.8 hours	
TMCLK (1/4096)	81920	97.6 hours	

(b) Memory Size

Set the size of the trace memory (buffer).

In other words, specify a memory size by dragging the knob. The sizes that can be specified are 8K (min.), 32K, 64K, 128K, and 256K (max).

Caution The larger the value that is set, the greater the number of trace data that are recorded. However, the response when reading trace data becomes correspondingly slower.

(c) Clear trace memory before run

Select this checkbox to clear the trace memory prior to program execution.

(d) Trace Data

Select the trace data to be collected.

Caution When [Option] menu -> [RRM Function] is selected, the setting is fixed to Branch PC.

Table 6-9 Relationship Between Meaning of Trace Data to Be Collected and Trace Collection Mode

Item	Branch PC Collects PC values of branch origin and branch destination instructions	All PC Collects PC values of all instructions[Note1]	Access PC Collects PC values of instructions that caused access	Access Data Collects access address and access data[Note2]
Branch PC	Traced	-	-	-
All PC	Traced			-
Access Data	-	-	-	Traced
Branch PC + Access Data	Traced	-	-	Traced
Branch PC + Access Data	Traced.			Traced
Access Data + Access PC	-	-	Traced	Traced
Branch PC + Access Data + Access PC	Traced	-	Traced	Traced

Note1 When selected, unconditional trace results, and qualify trace and section trace cannot be set simultaneously (conditional trace setting is invalid).

Note2 In the high-speed priority mode (select Speed Priority in (e) Mode:), when access to the internal RAM area is performed 32 times in succession, data may be missed.

(e) Mode:

Specify the trace collection mode.

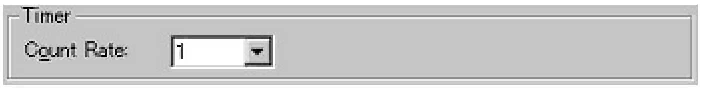
Caution When [Option] menu -> [RRM Function] is selected, the setting is fixed to Speed Priority.

Speed Priority	This mode performs tracing by prioritizing speed (real-time operation). In this mode, data may be missed depending on the trace data to be collected (refer to "Table 6-9 Relationship Between Meaning of Trace Data to Be Collected and Trace Collection Mode").
Trace Priority	This mode performs tracing by prioritizing data collection (non-real-time). Since, in order to reliably collect all the trace data, the CPU's execution pipeline is momentarily stopped when data is likely to be missed, the real-time characteristic of operation in relation to the user program is lost. Trace Priority cannot be selected when Branch PC or All PC is selected in (d) Trace Data .

(f) Complement Data

Select this checkbox to perform complementary display of trace data.

(2) Timer



Timer

Count Rate: 1

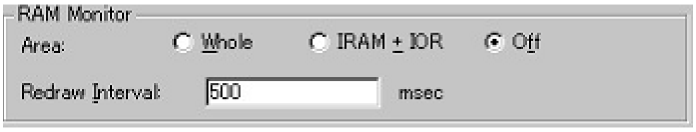
Set the rate value for the timer counter.

The rate value is selected from a drop-down list.

Table 6-10 Relationship Between Division Ratio and Maximum Measurement Time (Timer Count (Timer))

Division Ratio	Resolution (ns)	Maximum Measurable Time	Remark
TMCLK (1/1)	20	2.8 minutes	Timer counter is 33bits, In case of 50 MHz external clock
TMCLK (1/2)	40	5.7 minutes	
TMCLK (1/4)	80	11.4 minutes	
TMCLK (1/8)	160	22.8 minutes	
TMCLK (1/16)	320	45.6 minutes	
TMCLK (1/32)	640	1.5 hours	
TMCLK (1/64)	1280	3 hours	
TMCLK (1/128)	2560	6 hours	
TMCLK (1/256)	5120	12.2 hours	
TMCLK (1/512)	10240	24.4 hours	
TMCLK (1/1024)	20480	48.8 hours	
TMCLK (1/2048)	40960	97.6 hours	
TMCLK (1/4096)	81920	195.2 hours	

(3) RAM Monitor



RAM Monitor

Area: ○ Whole ○ IRAM + IOR ● Off

Redraw Interval: 500 msec

(a) Area

This area is used to set about RAM monitor function (refer to "5. 12. 3 RAM monitor function").

Whole	RAM sampling is to be performed, the entire memory space is the target. **Note:** 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.
IRAM+IOR	RAM sampling is to be performed, only the internal RAM area and the peripheral I/O register area are the target .
Off	Disables RAM monitor function. (default)

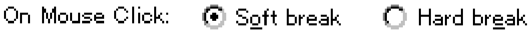
(b) Redraw Interval

Specify the sampling time (ms) of the RRM function (refer to "5. 12 The real-time RAM monitor function").

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

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

(4) On Mouse Click



On Mouse Click: ● Soft break ○ Hard break

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 red

(5) Break Sound

☑ Break Sound

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

(6) Verify Check

☑ Verify Check

This area is used to specify whether a verify check is performed when data has been written to memory.

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

Caution When N-Wire CARD 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).

(7) Clear register when reset

☑ Clear register when reset

Select this checkbox in order to clear the program registers (r1 to r31) and registers EIPC, EIPSW, FEPC, FEPSW, CTPC, CTPSW, and CTBP upon CPU reset.

Under the default setting, the registers are not cleared.

Function buttons

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

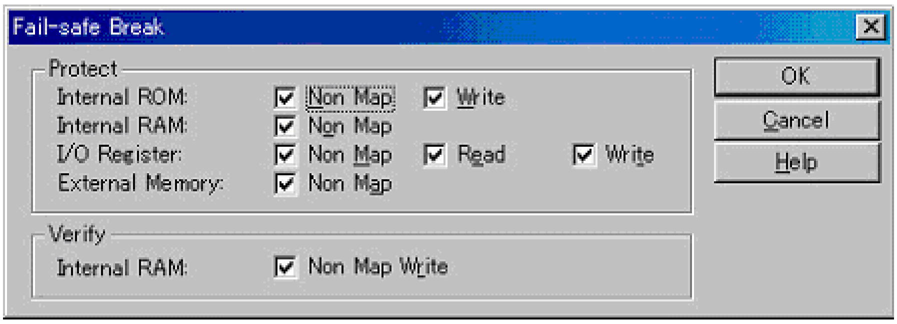
Fail-safe Break dialog box

IECUBE

This dialog box is used to perform the fail-safe break settings. (Refer to "5. 1 Setting debugging environment".)
When a project file is read, the results obtained by reading this project file are reflected in this dialog box.

Remark For details on the fail-safe break function, refer to the user's manuals of the in-circuit emulator and the emulation board that are used.

Figure 6-8 Fail-safe Break dialog box



Fail-safe Break

Protect
Internal ROM: ☑ Non Map ☑ Write
Internal RAM: ☑ Non Map
I/O Register: ☑ Non Map ☑ Read ☑ Write
External Memory: ☑ Non Map

Verify
Internal RAM: ☑ Non Map Write

- Opening
- Explanation of each area
- Function buttons

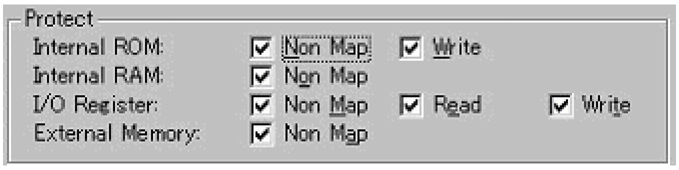
Opening

Click the <Detail...> button in the Configuration dialog box.

Explanation of each area

(1) Protect
(2) Verify

(1) Protect



Protect
Internal ROM: ☑ Non Map ☑ Write
Internal RAM: ☑ Non Map
I/O Register: ☑ Non Map ☑ Read ☑ Write
External Memory: ☑ Non Map

The fail-safe break protect settings are performed in this area.
The fail-safe breaks corresponding to the selected checkboxes are protected.
Under the default setting, all checkboxes are selected.

(a) Internal ROM:

This area is used to perform the protect settings for the internal ROM area.

Non Map	Access to access prohibited area
Write	Write to write prohibited area

(b) Internal RAM:

This area is used to perform the protect settings for the internal RAM area.

Non Map	Access to access prohibited area

(c) I/O Register:

This area is used to perform the protect settings for the peripheral I/O registers area.

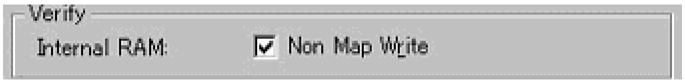
Non Map	Access to access prohibited area
Read	Read to read prohibited area
Write	Write to write prohibited area

(d) External Memory:

This area is used to perform the protect settings for the external memory area.

Non Map	Access to access prohibited area

(2) Verify



Verify

Internal RAM: ☑ Non Map Write

This area is used to perform the verify settings for fail-safe breaks.

Verify check is performed during access to the items corresponding to the selected checkboxes.

(a)Internal RAM:

This area is used to perform the verify check setting in the internal RAM area.

Non Map Write	Write to write prohibited area

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

IECUBE

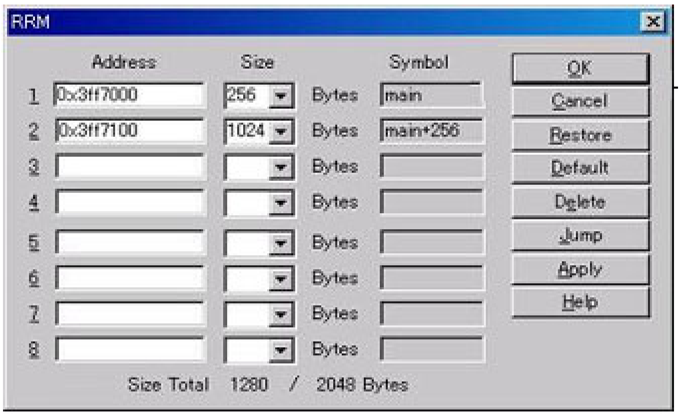
This dialog box is used to set the sampling range for the RRM function. (Refer to "5. 12 The real-time RAM monitor function".)

Up to 8 locations can be specified in 256-byte units as the sampling range.

The total of the sizes specified for the 8 locations cannot exceed 2048 bytes.

Caution This dialog box cannot be opened when [Option] menu -> [Trace Function] is selected.

Figure 6-9 RRM dialog box



RRM

	Address	Size		Symbol
1	0x3ff7000	256	Bytes	main
2	0x3ff7100	1024	Bytes	main+256
3			Bytes	
4			Bytes	
5			Bytes	
6			Bytes	
7			Bytes	
8			Bytes	

Size Total 1280 / 2048 Bytes

- 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, 256 bytes is displayed in an empty row in (2) Size, and the value obtained by converting the address to a symbol is displayed in an empty row in (3) Symbol. However, if the total of the sizes specified for the 8 locations already exceeds 2048 bytes, the dialog box opens but no value can be set.

(c) When settings are performed from the Watch window

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

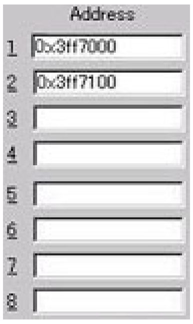
In this case, the value obtained by converting the variable into an address is displayed in an empty row in (1) Address, 256 bytes is displayed in an empty row in (2) Size, and the value obtained by converting the variable to a symbol is displayed in an empty row in (3) Symbol.

However, if the total of the sizes specified for the 8 locations already exceeds 2048 bytes, the dialog box opens but no value can be set.

Explanation of each area

- (1) Address
- (2) Size
- (3) Symbol
- (4) Size Total

(1) Address



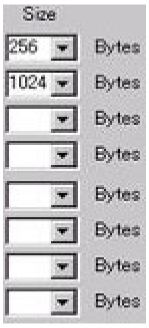
	Address
1	0x3ff7000
2	0x3ff7100
3	
4	
5	
6	
7	
8	

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 or the <Apply> button to enable the settings. When the settings are enabled, the addresses are aligned in 256-byte units, but if an address is duplicate, it is not enabled.

(2) Size

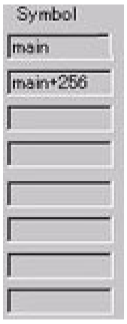


Size	
256	Bytes
1024	Bytes
	Bytes
	Bytes
	Bytes
	Bytes
	Bytes
	Bytes

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

The values that can be selected are 256, 512, 768, 1024, 1280, 1536, 1792, and 2048. However, the total of the sizes specified for the 8 locations cannot exceed 2048 bytes.

(3) Symbol



Symbol

main

main+256

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

Size Total 1280 / 2048 Bytes

This area displays the total of the sizes specified in (2) Size. If the total exceeds 2048 bytes, it is displayed in red.

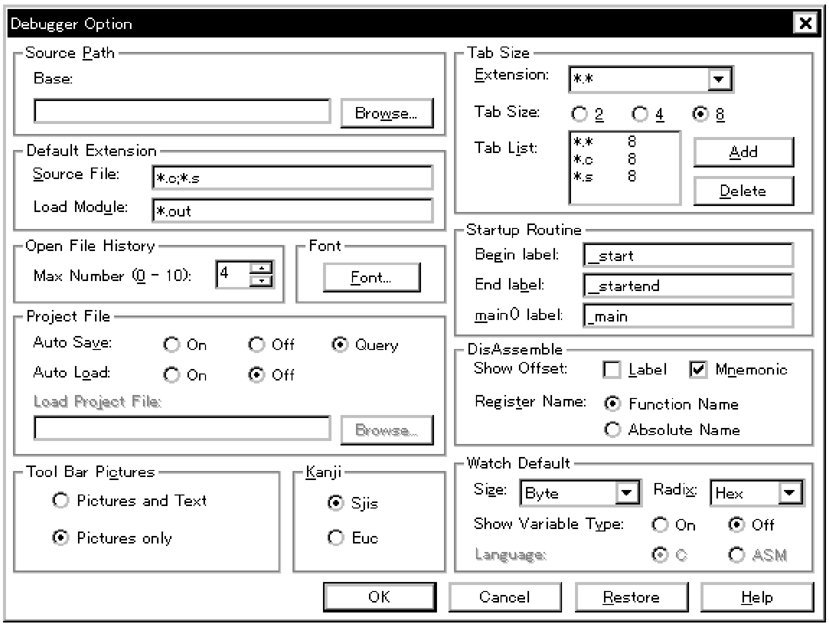
Function buttons

OK	Validates the settings and closes this dialog box.
Cancel	Closes this dialog box.
Restore	Restores the input data to the original status.
Default	Clears the current setting and sets the internal RAM start address in the first row in (1) Address, and 2048 in the first row in (2) Size.
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. 14. 1 Active Status and Static Status")
Apply	Enables the settings.
Help	Displays this dialog box online help files.

Debugger Option dialog box

This dialog box is used to display and set the various options of the ID850QB.

Figure 6-10 Debugger Option Dialog Box



Debugger Option

Source Path
Base:

Default Extension
Source File: *.c;*.s
Load Module: *.out

Open File History
Max Number (0 - 10): 4

Project File
Auto Save: ○ On ○ Off ● Query
Auto Load: ○ On ● Off
Load Project File:

Tool Bar Pictures
○ Pictures and Text
● Pictures only

Kanji
● Sjis
○ Euc

Tab Size
Extension: *.*
Tab Size: ○ 2 ○ 4 ● 8
Tab List:

.	8
*.c	8
*.s	8

Startup Routine
Begin label: _start
End label: _startend
main0 label: _main

DisAssemble
Show Offset: □ Label ☑ Mnemonic
Register Name: ● Function Name ○ Absolute Name

Watch Default
Size: Byte Radix: Hex
Show Variable Type: ○ On ● Off
Language: ● C ○ ASM

- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

(1) Source Path

(2) Default Extension

(3) Open File History

(4) Font

(5) Project File

(6) Tool Bar Pictures

(7) Kanji

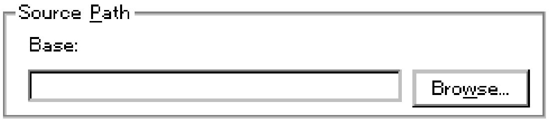
(8) Tab Size

(9) Startup Routine

(10) DisAssemble

(11) Watch Default

(1) Source Path



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

(a) Base :

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

1) Directory to which the project file has been loaded

2) Directory to which a load module or hex file has been loaded last

3) Current directory of Windows

(b) Text box

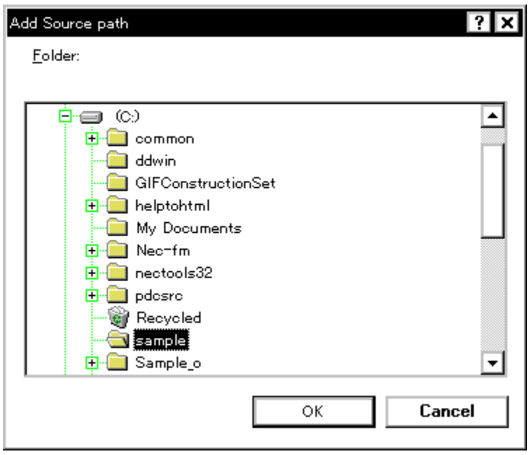
This area is used to specify the directory searched.

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

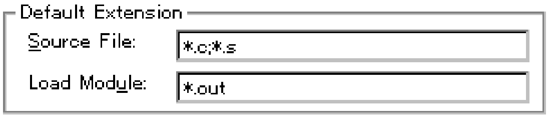
Opens the Add Source Path Dialog Box by clicking the <Browse...> button. To delimit paths, use ";" (semicolon) or "," (comma).

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

Figure 6-11 Add Source Path Dialog Box



(2) Default Extension



Default Extension

Source File: *.c;*.s

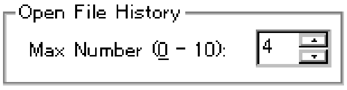
Load Module: *.out

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 ".
Load Module	Set the extension of a load module that is displayed when the Download dialog box is opened. The default extension is " *.out ".

(3) Open File History



Open File History

Max Number (0 - 10): 4

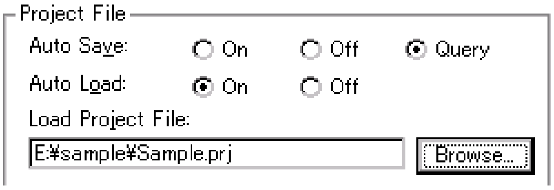
This area is used to set the number of histories of the open file displayed in the bottom field of the [File] menu. The default value is 4. If 0 is set, no history is displayed on the menu.

(4) Font

This area is used to specify the font displayed on the Source window, the Watch window, the Quick Watch dialog box, the Local Variable window, and the Stack Trace window. Clicking the <Font...> button opens the Font dialog box in which the font to be displayed and its size can be set.

(5) Project File

This area is used to set automatic saving and loading of the project file. (Refer to "5. 13. 1 Debugging environment (project file)".)



Project File

Auto Save: ○ On ○ Off ● Query

Auto Load: ● On ○ Off

Load Project File: E:¥sample¥Sample.prj

(a) Auto Save

Sets whether the project file is automatically saved at the ID850QB termination.

On	Automatically saves the project file at the ID850QB termination.
Off	Does not automatically save the project file at the ID850QB termination.
Query	Displays the Exit Debugger dialog box at the ID850QB termination (default).

(b)Auto Load

Sets whether the project file is automatically loaded at the ID850QB start up.

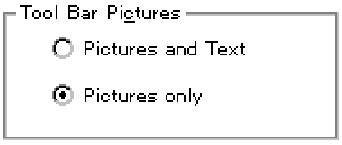
On	Automatically loads the project file at the ID850QB start up. Specify the project file to be loaded in the (c) Load Project File.
Off	Does not automatically loads the project file at the ID850QB start up. (default)

(c) Load Project File

Specify the project file to be loaded automatically.

Set a project file name by inputting from the keyboard or clicking the <Browse...> button. Clicking the <Browse...> button displays the Browse dialog box.

(6) Tool Bar Pictures



Tool Bar Pictures

○ Pictures and Text

● Pictures only

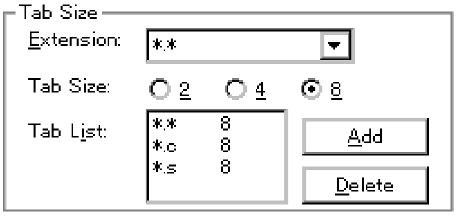
This area sets the buttons to be displayed on the toolbar. (Refer to " Toolbar".)

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

(7) Kanji

Cannot be selected.

(8) Tab Size



Tab Size

Extension: *.*

Tab Size: ○ 2 ○ 4 ● 8

Tab List:

.	8
*.c	8
*.s	8

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

(a) Extension

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

(b) Tab Size

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

(c) Tab List

Displays the tab size set for each extension.

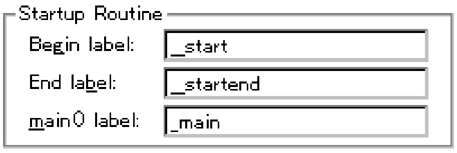
(d) <Add> button

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

(e) <Delete> button

To delete the tab size setting, select the setting to be deleted from Tab List and click <Delete> button.

(9) Startup Routine



Startup Routine

Begin label: _start

End label: _startend

main() label: _main

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

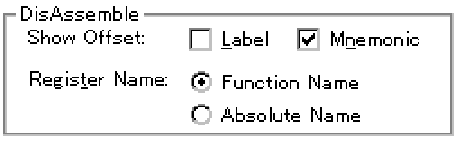
The source file can be opened if an object file in the load module format is downloaded in the Download dialog box.

Begin label	Specifies the symbol of the first address (default : _start)
End label	Specifies the symbol of the end address (default : _startend)
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.

(10) DisAssemble



DisAssemble

Show Offset: □ Label ☑ Mnemonic

Register Name: ● Function Name ○ Absolute Name

This area is used to set for disassemble display.

(a) Show Offset

Specifies whether an offset (symbol + offset) is displayed during disassemble display.

When the offset is not displayed, only a symbol that matches a numeric value is displayed, if any. If no matching symbol is found, the numeric value is displayed as a hexadecimal number unchanged.

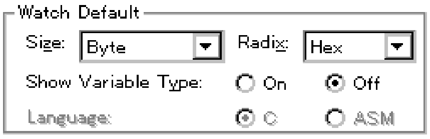
Label	Specifies whether the offset is displayed in the Label field. In the default condition, the offset is not displayed.
Mnemonic	Specifies whether the offset is displayed in the Mnemonic field. In the default condition, the offset is displayed.

(b) Register Name

This area is used to select the method of displaying register names in mnemonics during disassemble display.

Function Name	Displays register names as function names or nicknames (default).
Absolute Name	Displays register names as absolute names.

(11) Watch Default



Watch Default

Size: Byte

Radix: Hex

Show Variable Type: ○ On ● Off

Language: ● C ○ ASM

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

(a) Size

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

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

(b) Radix

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

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

(c) Show Variable Type

Select the display/non-display of variable type is specified.

On	Displays the type of a variable.
Off	Does not display the type of a variable (default).

(d) Language

Select the display/non-display of type of variable is specified.

C	Displays a C-like base number (default).
ASM	Cannot be selected.

Function buttons

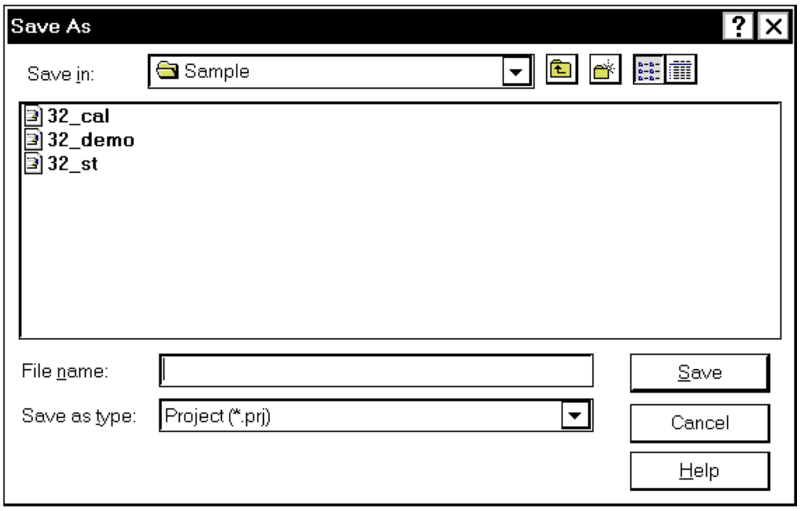
OK	Validates the settings and closes this dialog box.
Cancel	Cancels the changings and closes this dialog box.
Restore	Restores the previous settings before this dialog box was opened.
Help	Displays this dialog box online help files.

Project File Save dialog box

This dialog box is used to save the current debugging environment to a project file. (Refer to "5. 13. 1 Debugging environment (project file)".)

Project files can be newly saved or saved as in this dialog box.

Figure 6-12 Project File Save Dialog Box



- Opening
- Explanation of each area
- Function buttons

Opening

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

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

Explanation of each area

(1) Save in:, File name:

(2) Save as type:

(1) Save in:, File name:

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

Up to 257 character 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.

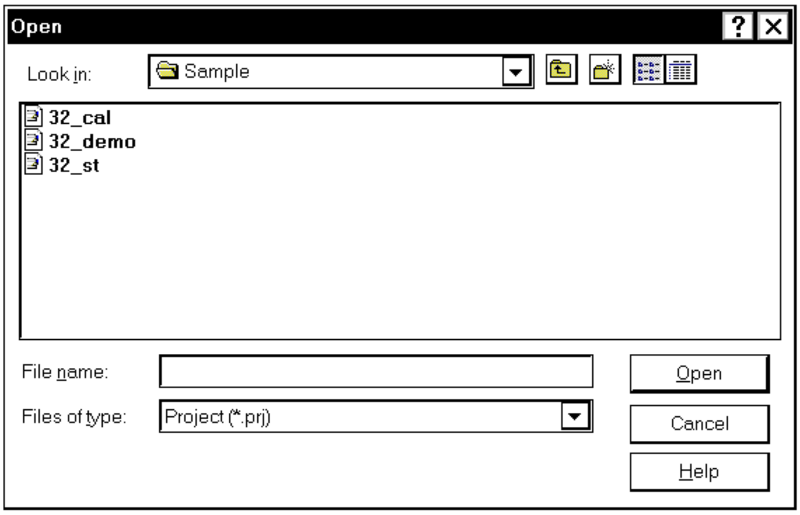
Project File Load dialog box

This dialog box is used to restore the debugging environment to the debugging environment saved to the project file. (Refer to "5. 13. 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 If, following ID850QB startup, 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 6-13 Project File Load Dialog Box



- Opening
- Explanation of each area
- Function buttons

Opening

Click the **Proj** button, 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 character string with a extension can be specified.

(2) Files of type :

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

Function buttons

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

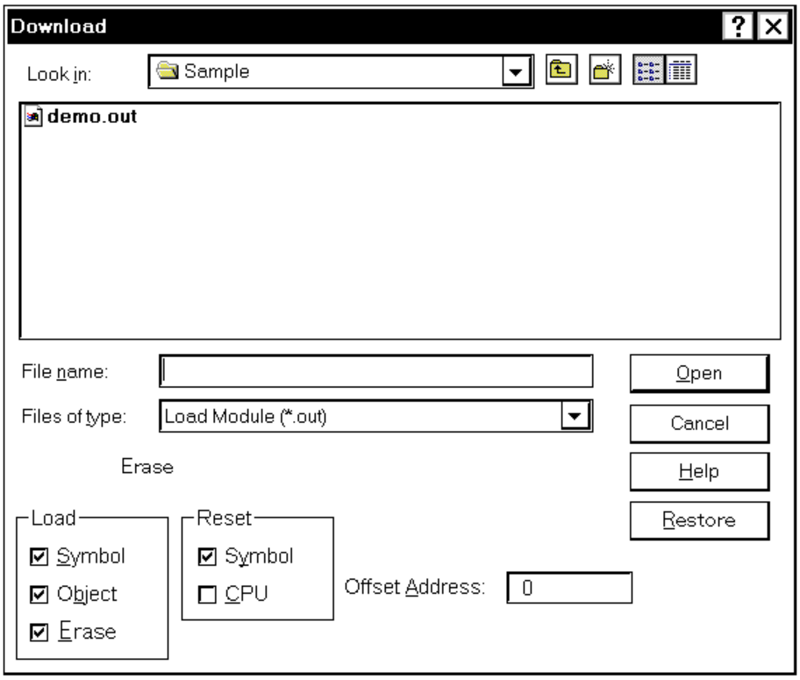
Download dialog box

This dialog box is used to select the name and format of a file to be downloaded, and downloads memory contents to the in-circuit emulator and the target system. (Refer to "5. 2 Download Function / Upload Function").

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

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

Figure 6-14 Download Dialog Box



Look in: Sample

Files of type: Load Module (*.out)

Load

☑ Symbol

☑ Object

☑ Erase

Reset

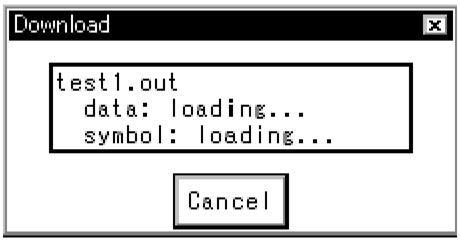
☑ Symbol

☐ CPU

Offset Address: 0

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.

Figure 6-15 The Progress of Download



```
test1.out
  data: loading...
  symbol: loading...
```

- Opening
- Explanation of each area

- Function buttons

Opening

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

Explanation of each area

(1) Look In:, File name:

(2) Files of type:

(3) Load

(4) Reset

(5) Offset Address

(1) Look In:, File name:

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

Two or more files can be specified in this dialog box. To specify two or more files, delimit each file name with " " (double quotation mark). Files can also be specified by clicking the mouse button while holding down the Shift or Ctrl key.

Up to 20 load module files can be downloaded.

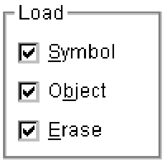
Up to 257 character string with a extension can be specified.

(2) Files of type:

This area is used to specify the type (extension) of the file to be 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



Load

☑ Symbol

☑ Object

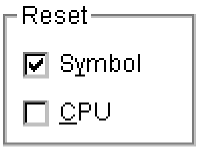
☑ Erase

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

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

Symbol	Specifies whether symbol information is read (when checked, default) or not. The memory capacity can be saved by not reading symbol information when a program consisting of two or more load module files is to be debugged and if the symbol information of some modules does not have to be read.
Object	Specifies whether object information is read (when checked, default) or not. (The object information is read even if this button is not checked when a HEX file is loaded.)
Erase	Specifies whether the contents of the internal flash memory are erased all before download (when checked, default) or not . (N-Wire CARD)

(4) Reset



Reset

☑ Symbol

□ CPU

This area is used to set a reset condition.

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

Symbol	Specifies whether symbol information is reset (when checked, default) or not. **Caution** When debugging a program consisting of two or more load module files, load each one of the load module files without resetting the symbol information. When downloading two or more load module files, take care that location addresses do not overlap.
CPU	Specifies whether the CPU is reset or not (not checked, default).

(5) Offset Address

Offset Address: 0

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

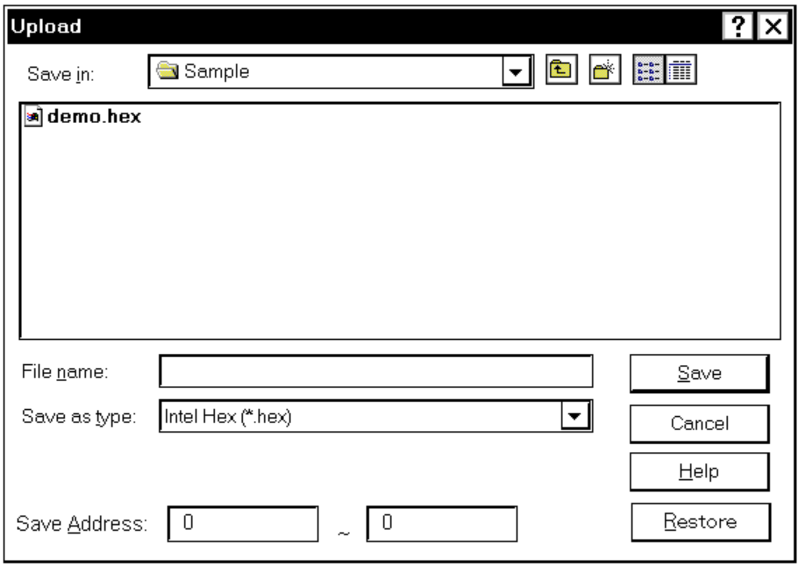
Function buttons

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")

Figure 6-16 Upload Dialog Box



Save in: Sample

File name:

Save as type: Intel Hex (*.hex)

Save Address: 0 ~ 0

- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

(1) Save in :, File name :

(2) Files of type :

(3) Save Address

(1) Save in :, File name :

This area is used to specify the file name to be saved. A file name can be directly input from the keyboard, or selected from the list.

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

(2) Files of type :

This area is used to specify the type (extension) of the file to be saved.

The format of the data to be saved is determined by the extension. (Refer to "Table 5-3 Type of File That Can Be Downloaded")

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

Remark Extensions other than those listed can also be used.

(3) Save Address

Save Address: 0 ~ 0

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

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

The default radix for inputting a numeric value is hexadecimal.

Function buttons

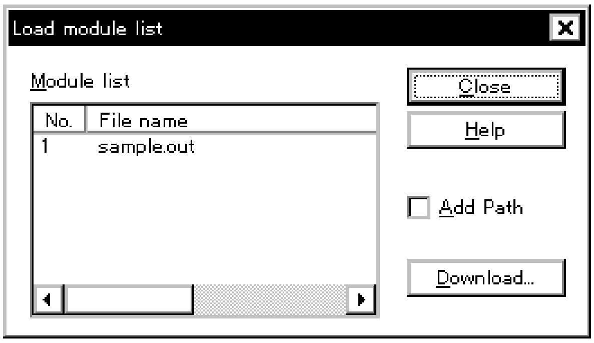
Save	Saves the file according to the setting.
Cancel	Closes this dialog box without executing anything.
Help	Displays this dialog box online help files.
Restore	Restores the status before this dialog box was opened.

Load Module List dialog box

This dialog box can be used to check the file name and file path name of a load module that has already been downloaded. (Refer to "5. 2 Download Function / Upload Function")

By using the <Download...> button, The Download dialog box can be opened and a load module can be downloaded.

Figure 6-17 Load Module List Dialog Box



Load module list

Module list

No.	File name
1	sample.out

- Opening
- Explanation of each area
- Function buttons

Opening

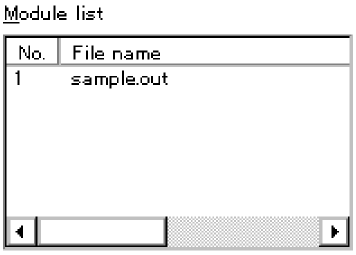
Select [File] menu -> [Load Module...].

Explanation of each area

(1) Module list

(2) Add Path

(1) Module list



Module list

No.	File name
1	sample.out

This area displays the names of the load module files that have already been downloaded.

(a) No.

The numbers displayed indicate the sequence in which the load module file names were read.

(b) File name

The file names are displayed with the full path if (2) Add Path is checked; otherwise, only the file names will be displayed.

Caution If symbol information has been reset in the Reset Debugger dialog box, or if symbol information has been reset in the Download dialog box, the load module file names downloaded before that are cleared.

(2) Add Path

This should be checked to specify load module file names are displayed with the path in the (1) Module list.

Function buttons

Close	Closes this dialog box.
Help	Displays this dialog box online help files.
Download...	Opens the Download dialog box. A new load module can be downloaded. The file name of the newly downloaded module will be added to the file name display area when the Download dialog box is closed.

Source window

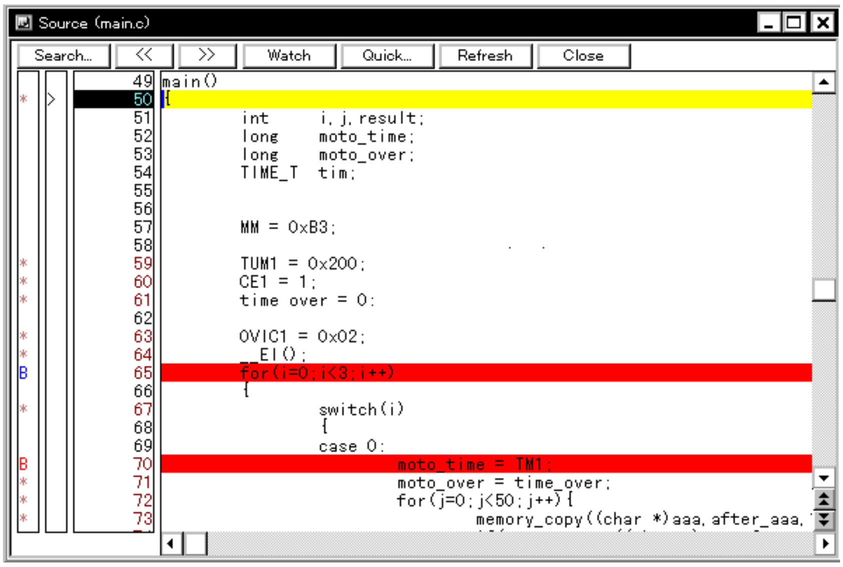
This dialog box is used to displays source files or text files. (Refer to "5. 3 Source Display, Disassemble Display Function".)

In addition to Breakpoint setting and Mixed display mode (Source window), a number of other operations using Context menu, Function buttons, etc., can be performed in this window.

Moreover, there are two statuses, Active Status and Static Status, for this window. When the window is in the active status, it has Trace Result with Linking Window (when IECUBE connected). Moreover, the items selected in the window with Drag & drop function can be used in another window. (Refer to "5. 14 Functions Common to Various Windows".)

Caution 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.

Figure 6-18 Source window



```
main()
{
        int     i, j, result;
        long    moto_time;
        long    moto_over;
        TIME_T  tim;


        MM = 0xB3;

        TUM1 = 0x200;
        CE1 = 1;
        time over = 0;

        OVIC1 = 0x02;
        __EI();
        for(i=0;i<3;i++)
        {
                switch(i)
                {
                case 0:
                        moto_time = TM1;
                        moto_over = time_over;
                        for(j=0;j<50;j++){
                                memory_copy((char *)aaa, after_aaa,
```

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

Opening

Click the **Src** button, select [Browse]menu -> [Source Text]. (This window is automatically opened if the corresponding source file exists after the download module file has been downloaded.)

Explanation of each area

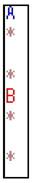
(1) Point mark area

(2) Current PC mark area

(3) Line number/address display area

(4) Source text display area

(1) Point mark area



This area is used for Event Setting Status (Event Mark), program code display, and Breakpoint setting.

Caution The program code is displayed only when the symbol information downloaded by the load module is read. Breakpoints can be set or deleted by clicking with the mouse on this program code. (if "*" is not displayed for the line, the breakpoint is set on the line above or below the line, whichever has "*" displayed).

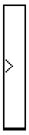
If an event has been set for the corresponding line, one of the marks listed in the following table is displayed. The color of the "B" mark differs according to the breakpoint type and status. (When a breakpoint is set in this area, it is enabled at the same time that it is set.)

Table 6-11 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.
T	Trace event is set..(IECUBE)
Ti	Timer 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

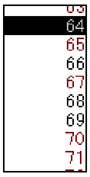


The mark '>', which indicates the current PC value (PC register value), is displayed in this area.

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.

(3) Line number/address display area

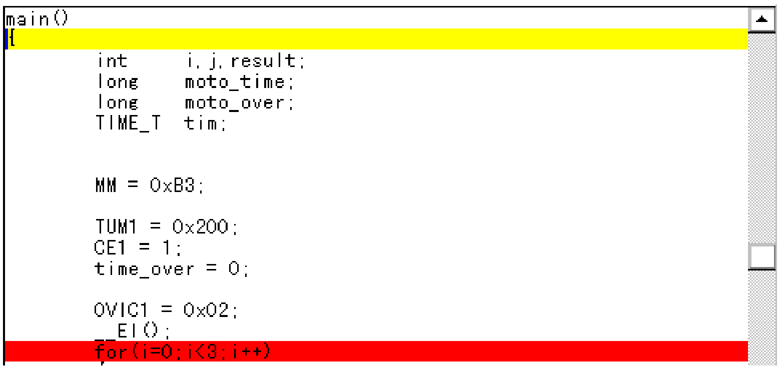


```
63
64
65
66
67
68
69
70
71
```

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

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

(4) Source text display area



```
main()
{
        int     i, j, result;
        long    moto_time;
        long    moto_over;
        TIME_T  tim;


        MM = 0xB3;

        TUM1 = 0x200;
        CE1 = 1;
        time_over = 0;

        OVIC1 = 0x02;
        __EI();
        for(i=0;i<3;i++)
```

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?		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).

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?	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.
Assemble	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position. (Refer to "5. 14. 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. 14. 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

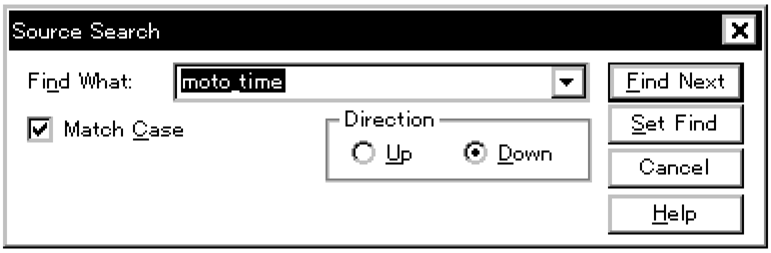
Search...	Opens the Source Search dialog box and searches a character string of the source text. If a character string is selected in the source text display area, the Source Search dialog box is opened to search the character string. If no character string is selected, the Source Search dialog box is opened with nothing specified to be searched. Specify a search method in the Source Search dialog box. The result 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.

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 6-19 Source Search Dialog Box



Find What: moto_time

☑ Match Case

Direction: ○ Up ● Down

- Opening
- Explanation of each area
- Function buttons

Opening

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

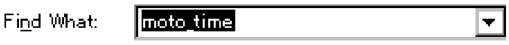
Explanation of each area

(1) Find What

(2) Match Case

(3) Direction

(1) Find What



Find What: moto_time

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.

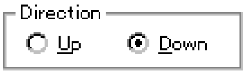
Up to 16 input histories can be recorded.

(2) Match Case

☑ Match Case

This should be checked to distinguish between uppercase and lowercase.

(3) Direction



Direction

○ Up ● Down

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).

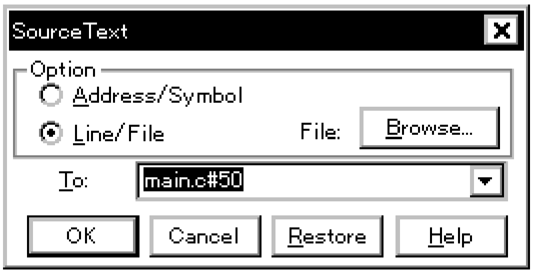
Function buttons

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.
Cancel	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 6-20 Source Text Move Dialog Box



- Opening
- Explanation of each area
- Function buttons

Opening

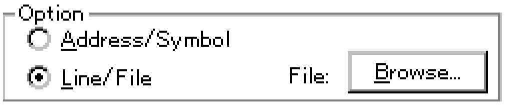
When the Source window is the current window, select [View] menu -> [Move...].

Explanation of each area

(1) Option

(2) To

(1) Option



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

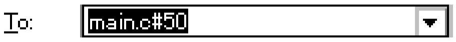
(a) Address/Symbol

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

(b) Line/File

This should be selected to specify by 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.

(a) When the (a) Address/Symbol is selected

Specifies the address from which display is to be started.

The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or a expression. (Refer to "Table 5-6 Specifying Symbols".)

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

(b) When the (b) Line/File is selected

Specifies the line number (or a file name) from which display is to be started.

Specification of the line number is done with **[[path name] file name]# line number**.

The default radix for inputting a numeric value is decimal.

File name specification can be done just with the file name, or using the absolute path and relative path.

If just the file name or the relative path was specified, the file in the source path specified in the Debugger Option dialog box is searched.

The file whose specified line number was specified as the first line is displayed by clicking the <OK> button. When the file name is omitted, the currently displayed file is displayed from the specified line. If the line number is omitted, display is done from the first line of the file.

Function buttons

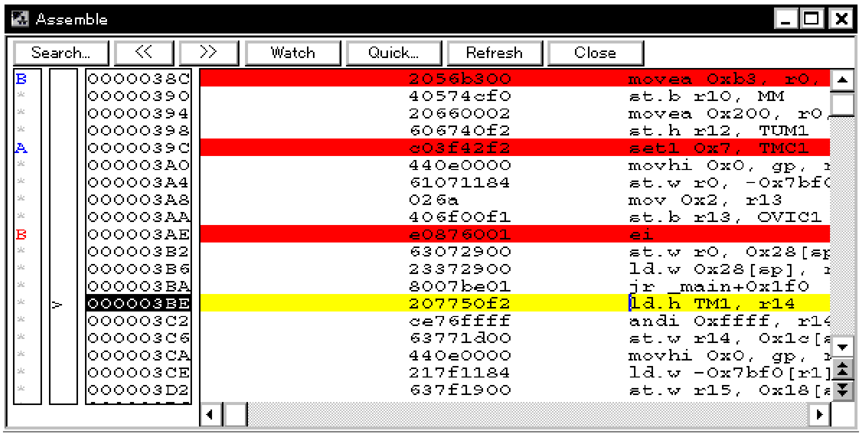
OK	Starts displaying the source text from the specified position.
Cancel	Closes this dialog box.
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

Assemble window

This window is used to disassemble and display programs. It is also used to execute online assembly. (Refer to "5. 3 Source Display, Disassemble Display Function".) The results of online assembly are also reflected in the Memory window. In addition to Breakpoint setting , a number of other operations using Context menu, Function buttons, etc., can be performed in this window.

Moreover, there are two statuses, Active Status and Static Status, for this window. When the window is in the active status, it has a Trace Result with Linking Window (when IECUBE connected). Moreover, the items selected in the window with the Drag & drop function can be used in another window. (Refer to "5. 14 Functions Common to Various Windows".)

Figure 6-21 Assemble window



```
B  0000038C  2056b300  movea 0x3b3, r0,
*  00000390  40574cf0  st.b r10, MM
*  00000394  20660002  movea 0x200, r0,
*  00000398  606740f2  st.h r12, TUM1
A  0000039C  c03f42f2  set1 0x7, TMC1
*  000003A0  440e0000  movhi 0x0, gp, r
*  000003A4  61071184  st.w r0, -0x7bf0
*  000003A8  026a      mov 0x2, r13
*  000003AA  406f00f1  st.b r13, OVIC1
B  000003AE  e0876001  ei
*  000003B2  63072900  st.w r0, 0x28[sp
*  000003B6  23372900  ld.w 0x28[sp], r
*  000003BA  8007be01  jr _main+0x1f0
* > 000003BE  207750f2  ld.h TM1, r14
*  000003C2  ce76ffff  andi 0xffff, r14
*  000003C6  63771d00  st.w r14, 0x1c[s
*  000003CA  440e0000  movhi 0x0, gp, r
*  000003CE  217f1184  ld.w -0x7bf0[r1]
*  000003D2  637f1900  st.w r15, 0x18[s
```

- Opening
- Explanation of each area
- [View] menu (Assemble window-dedicated items)
- Context menu
- Function buttons
- Related operations

Opening

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

Explanation of each area

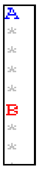
(1) Point mark area

(2) Current PC mark area

(3) Address specification area

(4) Disassemble display area

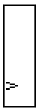
(1) Point mark area



```
A
*
*
*
*
B
*
*
```

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

(2) Current PC mark area

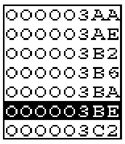


```
>
```

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

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

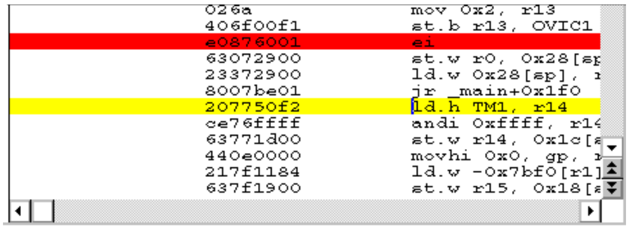
(3) Address specification area



```
000003AA
000003AE
000003B2
000003B6
000003BA
000003BE
000003C2
```

This area displays the disassembly start address.

(4) Disassemble display area



```
026a        mov 0x2, r13
406f00f1    st.b r13, OVIC1
e0876001    ei
63072900    st.w r0, 0x28[sp
23372900    ld.w 0x28[sp], r
8007be01    jr _main+0x1f0
207750f2    ld.h TM1, r14
ce76ffff    andi 0xffff, r14
63771d00    st.w r14, 0x1c[s
440e0000    movhi 0x0, gp, r
217f1184    ld.w -0x7bf0[r1]
637f1900    st.w r15, 0x18[s
```

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

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

It can be Online assembly in the mnemonic field.

This area also provides the following functions:

- [Come Here] , [Start From Here] (Refer to "Table 5-9 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?	Displays the event information of the address at the cursor position. If an event is set, the Event dialog box is opened.

Context menu

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.
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. 14. 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. 14. 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 result 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) Online assembly

To change the disassembled contents, move the cursor to the mnemonic field (the overwrite and insertion modes are alternately selected by pressing the Insert key).

If an attempt is made to move the cursor to another line after the disassembled contents have been changed in the mnemonic field, the new contents are checked. If the new contents are illegal, the code data on the line where the contents have been changed is indicated as '*'.

The contents changed in the mnemonic field are written into the memory by pressing the Enter key. By pressing the Enter key, the new contents are checked. If even one line is illegal, the new contents are not written into the memory. To discard the contents, press the ESC key.

If the contents are correct and if the Enter key is pressed, the contents are written to the memory, and then the cursor moves to the next line in the mnemonic field, so that the data on the next line can be changed.

Caution If the number of new instruction bytes is less than the number of previous instruction bytes as a result of changing, as many 'NOP' instructions as necessary are inserted. If the number of new instruction bytes is more than the number of previous instruction bytes, the next instruction is overwritten. In this case also, as many 'NOP' instructions as necessary are inserted. The same applies to instructions that straddle over source lines.

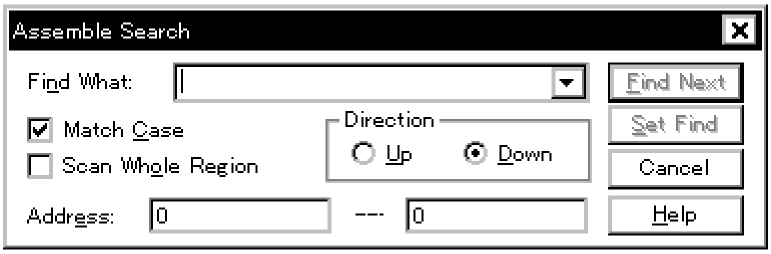
Assemble Search dialog box

This dialog box is used to search the contents in the Assemble window . (Refer to "5. 3. 2 Disassemble display".)

Successive character strings included in an input character string and disassembler character string are compared as one blank character.

By setting each item and then clicking the <Find Next> button, searching can be started. By clicking the <Set Find> button, the direction buttons ("<<" and ">>") in the Assemble window can be used for the search.

Figure 6-22 Assemble Search Dialog Box



- Opening
- Explanation of each area
- Function buttons

Opening

When the Assemble window is the current window, select [View] menu -> [Search...], or click the <Search...> button in the same window.

Explanation of each area

(1) Find What

(2) Match Case

(3) Scan Whole Region

(4) Direction

(5) Address

(1) Find What

This area is used to specify the data to be searched (up to 256 character).

In the default condition, the string selected in the window that called this dialog box is displayed. As necessary, the character string displayed can be changed.Up to 16 input histories can be recorded.

(2) Match Case

☑ Match Case

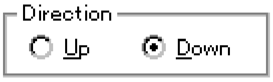
This should be checked to distinguish between uppercase and lowercase.

(3) Scan Whole Region

□ Scan Whole Region

This should be checked to search the entire specified range.

(4) Direction



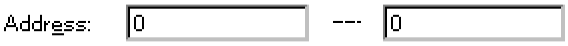
Direction

○ Up ● Down

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



Address: 0 --- 0

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

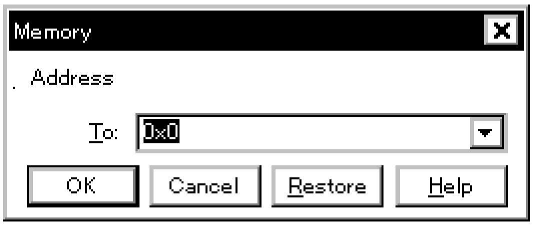
Find Next	Searches the specified data in accordance with a given condition. If the specified character string is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.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

Figure 6-23 Address Move Dialog Box (Example: When Memory Window Is Open)



To: 0x0

- 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

To: 0x0

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.

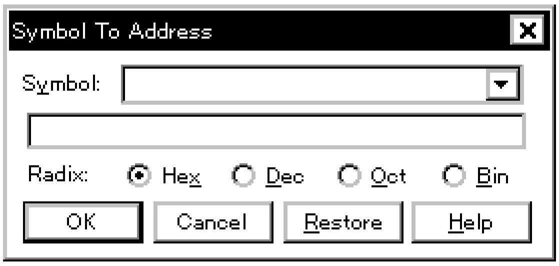
Function buttons

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 Source Display, Disassemble Display Function".)

Figure 6-24 Symbol To Address Dialog Box



- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

(1) Symbol

(2) Conversion result display area

(3) Radix

(1) Symbol

This area is used to specify the variable, function name, symbol name, or line number to be converted. (Refer to "Table 5-6 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 the (2) Conversion result display area.

(2) Conversion result display area

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

The variable, address of the function, value of the symbol, address of the line number, or value of the expression specified in the (1) Symbol is displayed. The address value of an I/O port name or peripheral I/O registers name, the register contents of a register name, or flag value of a PSW flag name is displayed.

(3) Radix

Radix: ● Hex ○ Dec ○ Oct ○ Bin

This area is used to select the radix of the data to be displayed in the (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 the (1) Symbol have been changed, converts the symbol. After conversion, closes the dialog box if the contents of the (1) Symbolhave not been changed.
Cancel	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.

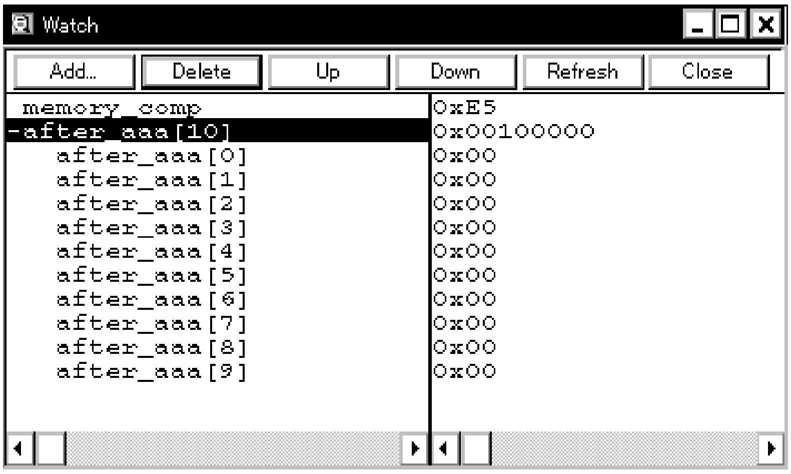
The result of updating and rewriting data in this window will be reflected in the Memory window.

Watch data registration is done by pressing the <Watch...> button in the Source window or the Assemble window. (Refer to "5. 6. 3 Registering and deleting watch data".)

This window allows easy setting of breakpoints to variables with a Context menu.

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

Figure 6-25 Watch Window



```
 memory_comp          0xE5
-after_aaa[10]        0x00100000
    after_aaa[0]      0x00
    after_aaa[1]      0x00
    after_aaa[2]      0x00
    after_aaa[3]      0x00
    after_aaa[4]      0x00
    after_aaa[5]      0x00
    after_aaa[6]      0x00
    after_aaa[7]      0x00
    after_aaa[8]      0x00
    after_aaa[9]      0x00
```

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

Opening

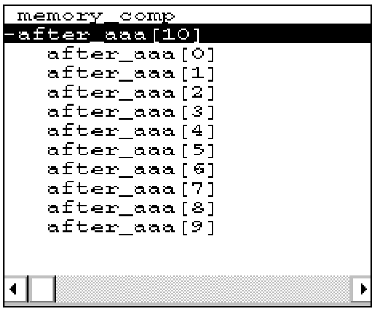
Click the **Wch** button, or select [Browse] menu -> [Watch].

Explanation of each area

(1) Left field (symbol name display area)

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

(1) Left field (symbol name display area)



```
 memory_comp
-after_aaa[10]
    after_aaa[0]
    after_aaa[1]
    after_aaa[2]
    after_aaa[3]
    after_aaa[4]
    after_aaa[5]
    after_aaa[6]
    after_aaa[7]
    after_aaa[8]
    after_aaa[9]
```

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 6-12 Watch Window Display Format (Symbol)

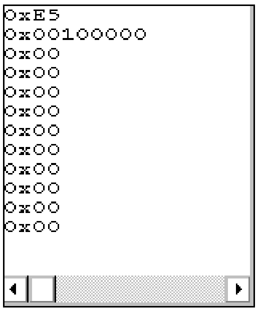
First characte r	Meaning	
+	Array, pointer variable, or structure/union Expanded display is performed by double-clicking "+" (first character changes from "+" to "-").	
	Array	By double-clicking the '+', all the elements of the variable are displayed in accordance with the type of the array variable.
	Pointer variable	By double-clicking the '+', the data indicated by the pointer is displayed.
	Structure/union	By double-clicking the '+', all the members of the structure/union are displayed in accordance with the type of the member variable. If a structure or union is defined in the structure or union, the structure name or union name of the internal structure or union is also displayed. The internal structure or union can be also expanded by using '+'.
-	Expanded display variable Expanded display is canceled by double-clicking "-" (first character changes from "-" to "+").	

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

Registered watch data changes are performed in the Change Watch dialog box opened by selecting the item to be changed and then selecting Context menu -> [Change Watch...]. A line with an expanded hierarchy, such as the elements of an array, and members of structures and unions cannot be deleted.

If an access breakpoint is set for a variable or a symbol in the Watch window, the symbol name display area is highlighted in gold.

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



```
0xE5
0x00100000
0x00
0x00
0x00
0x00
0x00
0x00
0x00
0x00
0x00
0x00
```

This area is used to display and change watch data values. A value is updated when execution is stopped.

To save a value, select [File] menu -> [Save As...]. This area is blank if getting data has failed.

Values are changed through direct input. The location to be changed is displayed in red and the contents of the change are written into the target memory when the Enter key is pressed. The previous value can be canceled by the ESC key.

The display format is as follows:

Table 6-13 Watch Window Display Format (Data)

Display Data	Contents
Integer	Hexadecimal (**0x***xxxx*) Decimal (*xxxx*) Octal (**0***xxxx*\) Binary (**0b***xxxx*)
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.

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?		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 octal 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.
Half Word		Displays in 16-bit units.
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 peripheral I/O registers 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

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 real-time RAM monitor function (RRM function). Opens the RRM dialog box (when IECUBE is connected).
Event?	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.
Half Word	Displays the selected line in 16-bit units.
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.

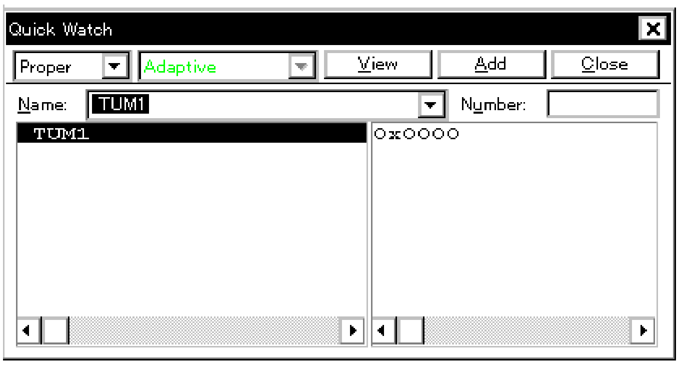
Function buttons

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 6-26 Quick Watch Dialog Box



Name: TUM1

TUM1	0x0000

- Opening
- Explanation of each area
- Function buttons

Opening

When the Source window or Assemble window is the current window, select [View] menu -> [Quick Watch...], or click the <Quick...> button in same window.

Explanation of each area

(1) Name

(2) Symbol name display area

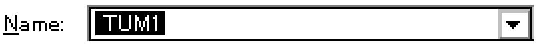
(3) Data value display/setting area

(4) Display radix selection area (upper left field)

(5) Display size selection area (second upper left field)

(6) Number

(1) Name



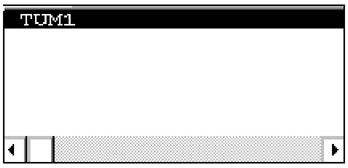
Name: TUM1

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

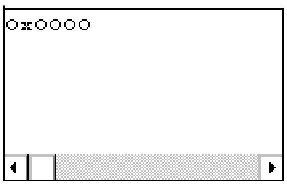


TUM1

This area is used to display watch data (variable names, symbol names and types, and tag names of structures or unions). (Refer to "(1) Left field (symbol name display area)" in the Watch window.)

This area cannot be edited.

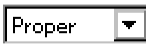
(3) Data value display/setting area



0x0000

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

(4) Display radix selection area (upper left field)



Proper

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 (second upper left field)

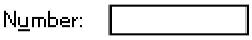
Adaptive

This area is used to select the display size.

If the display size is fixed, such as when a variable in C 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.
Half Word	Displays in 16-bit units.
Word	Displays in 32-bit units.

(6) Number



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 or register is to be displayed, the specified number of data is invalid.

Function buttons

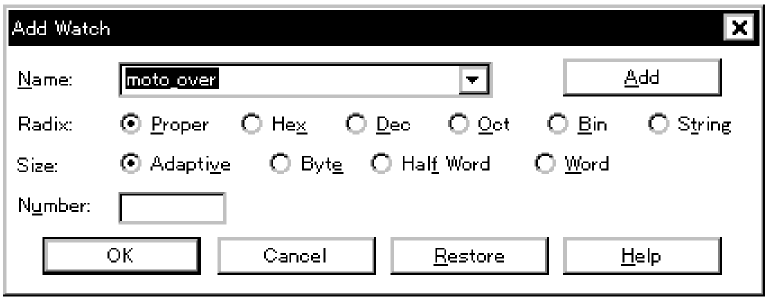
View	Displays the data specified in (1) Name in the field below.
Add	Adds the data specified in (1) Name to the Watch window.
Close	Closes this dialog box. Data that has not actually been written to the target memory will be canceled.

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 6-27 Add Watch Dialog Box



Add Watch

Name: moto_over

Radix: ● Proper ○ Hex ○ Dec ○ Oct ○ Bin ○ String

Size: ● Adaptive ○ Byte ○ Half Word ○ Word

Number:

- Opening
- Explanation of each area
- Function buttons

Opening

Select [View] menu -> [Add Watch...], or click the <Add...> button in the Watch window.

Explanation of each area

(1) Name

(2) Radix

(3) Size

(4) Number

(1) Name

Name: moto_over

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 6-14 Watch Window Input Format

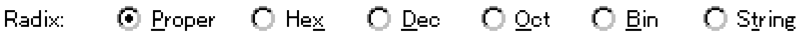
<table>
<tr><td colspan="3">- Variable Name of C language</td></tr>
<tr><td></td><td colspan="2">Variable expression : Variable Name</td></tr>
<tr><td></td><td>|Variable expression [Constant value | Variable Name]</td><td>Elements of array</td></tr>
<tr><td></td><td>|Variable expression . Member name</td><td>Entity members of structure/union</td></tr>
<tr><td></td><td>|Variable expression -> Member name</td><td>Members of structure/union indicated by pointer</td></tr>
<tr><td></td><td>|*Variable expression</td><td>Value of pointer variable</td></tr>
<tr><td></td><td>|&Variable expression</td><td>Address where variable is located</td></tr>
<tr><td colspan="3">- Register name</td></tr>
<tr><td colspan="3">- peripheral I/O registers name, peripheral I/O registers bit name</td></tr>
<tr><td colspan="3">- Label and address of immediate value</td></tr>
<tr><td colspan="3">- Register name.bit</td></tr>
<tr><td colspan="3">- peripheral I/O registers name. bit</td></tr>
<tr><td colspan="3">- Label name.bit , address of immediate value.bit</td></tr>
<tr><td colspan="3">- Specification of scope</td></tr>
</table>

How a variable is handled when a scope Is specified is as follows:

Table 6-15 How a Variable Is Handled When a Scope Is Specified

Scope Specification	Program name	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

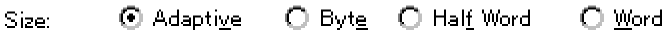
(2) Radix



Radix: ● Proper ○ Hex ○ Dec ○ Oct ○ Bin ○ String

This area is used to select the display radix. (Refer to "(4) Display radix selection area (upper left field)" in the Quick Watch dialog box.)

(3) Size



Size: ● Adaptive ○ Byte ○ Half Word ○ Word

This area is used to select the display size. (Refer to "(5) Display size selection area (second upper left field)" in the Quick Watch dialog box.)

(4) Number

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

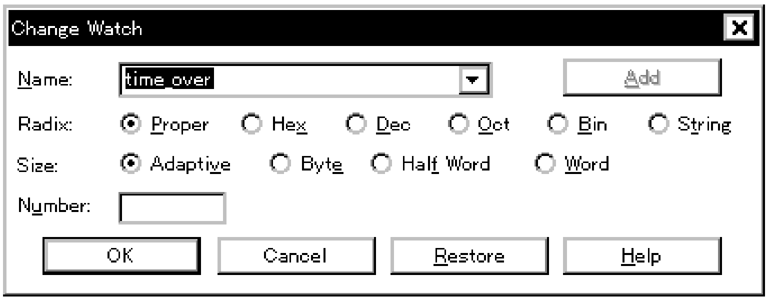
This window is used to change the data on a line selected in the Watch window. (Refer to "5. 6 Watch Function".)

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 6-28 Change Watch Dialog Box



Change Watch

Name: time_over

Radix: ● Proper ○ Hex ○ Dec ○ Oct ○ Bin ○ String

Size: ● Adaptive ○ Byte ○ Half Word ○ Word

Number:

- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

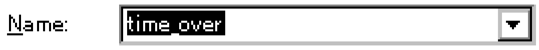
(1) Name

(2) Radix

(3) Size

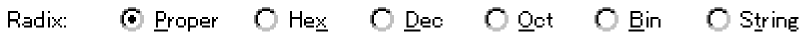
(4) Number

(1) Name


Name: time_over

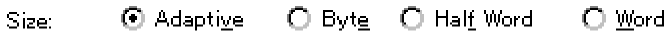
This area is used to change a symbol name on a line selected in the Watch window. (Refer to "(1) Name" in the Add Watch dialog box.)

(2) Radix


Radix: ● Proper ○ Hex ○ Dec ○ Oct ○ Bin ○ String

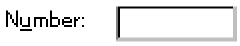
This area is used to change the display radix on a line selected in the Watch window .(Refer to "(4) Display radix selection area (upper left field)"in the Quick Watch dialog box.)

(3) Size


Size: ● Adaptive ○ Byte ○ Half Word ○ Word

This area is used to change the display size on a line selected in the Watch window.(Refer to "(5) Display size selection area (second upper left field)" in the Quick Watch dialog box.)

(4) Number


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.

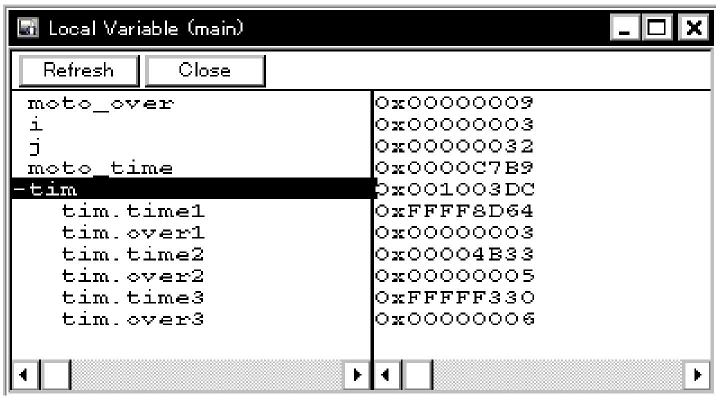
Local Variable window

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 Trace 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 6-29 Local Variable Window



```
moto_over        0x00000009
i                0x00000003
j                0x00000032
moto_time        0x0000C7B9
-tim             0x001003DC
   tim.time1     0xFFFF8D64
   tim.over1     0x00000003
   tim.time2     0x00004B33
   tim.over2     0x00000005
   tim.time3     0xFFFFF330
   tim.over3     0x00000006
```

- Opening
- Explanation of each area
- Context menu
- Function buttons

Opening

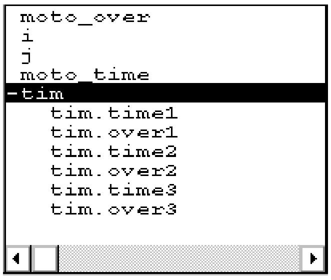
Click the **Loc** button, or select [Browse] menu -> [Local Variable].

Explanation of each area

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

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

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



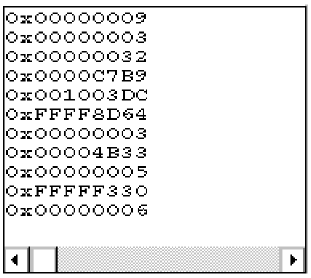
```
moto_over
i
j
moto_time
-tim
   tim.time1
   tim.over1
   tim.time2
   tim.over2
   tim.time3
   tim.over3
```

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

Auto, internal Static, and Register variables can be displayed.

This area cannot be edited.

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



```
0x00000009
0x00000003
0x00000032
0x0000C7B9
0x001003DC
0xFFFF8D64
0x00000003
0x00004B33
0x00000005
0xFFFFF330
0x00000006
```

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

[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

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 Trace 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".)

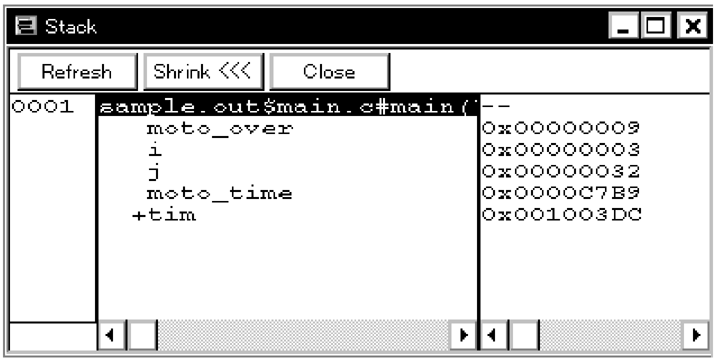
The window corresponding to the stack contents can be jumped to with the Jump function.

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

Caution1 The stack trace display function may not operate correctly if there is a function that does not create a stack frame.

Caution2 [ERROR] may be displayed during prologue or epilogue processing of a function.

Figure 6-30 Stack Trace Window



```
0001  sample.out$main.c#main(   --
         moto_over               0x00000009
         i                       0x00000003
         j                       0x00000032
         moto_time               0x0000C7B9
       +tim                      0x001003DC
```

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

Opening

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

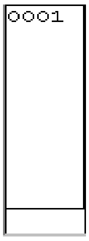
Explanation of each area

(1) Left field (stack frame number display area)

(2) Center field (stack frame contents display area)

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

(1) Left field (stack frame number display area)

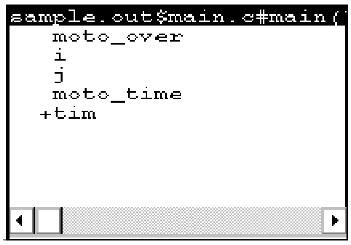


```
0001
```

This area assigns numbers to and displays the stack contents.

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

(2) Center field (stack frame contents display area)



```
sample.out$main.c#main(
    moto_over
    i
    j
    moto_time
  +tim
```

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]

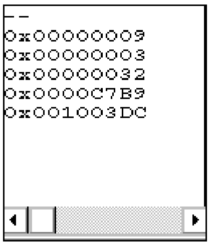
If this line is double-clicked, the operation will be the same as jumping to the Source window of the Jump function (i.e., the local variable in the function to which execution has jumped will be displayed in the Local Variable window).If the function has a local variable, the local variable will be 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 6-12 Watch Window Display Format (Symbol)".)

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

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



```
--
0x00000009
0x00000003
0x00000032
0x0000C7B9
0x001003DC
```

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 "(2) Right field (data value display/setting area)" in the Watch window.)

[View] menu (Stack Trace 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. 14. 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. 14. 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. 14. 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".)

The memory access status (read, write, read & write) can be displayed using different colors (refer to "5. 7. 4 Access monitor function (when IECUBE connected)").

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 Trace Result with Linking Window (when IECUBE connected), Jump function . (Refer to "5. 14 Functions Common to Various Windows".)

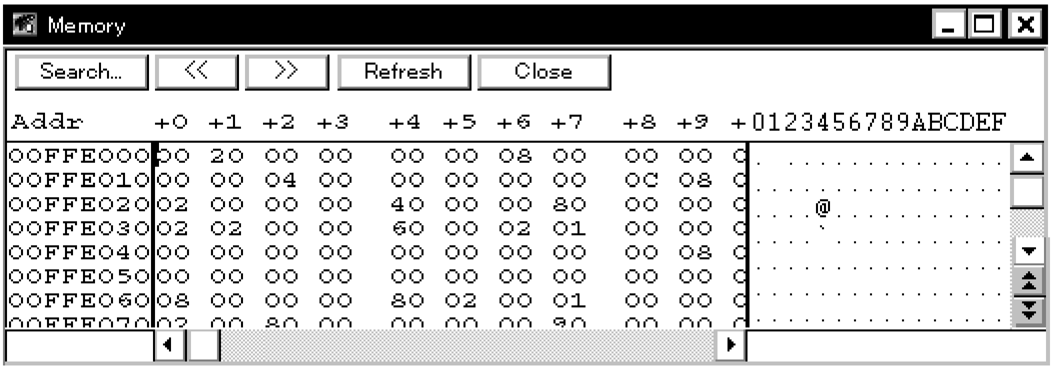
Remark The display start position when the this window is opened is as follows:

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

Second and subsequent times : Display starts from the address at which an active status window was closed.

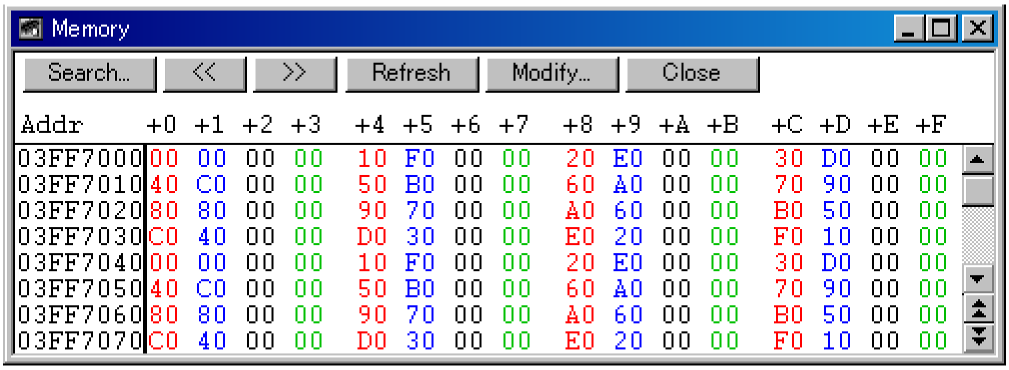
(if an active status window has never been closed, display starts from the first display start position).

Figure 6-31 Memory Window



```
Addr     +0 +1 +2 +3  +4 +5 +6 +7  +8 +9 +0123456789ABCDEF
00FFE000 00 20 00 00  00 00 08 00  00 00 0 ................
00FFE010 00 00 04 00  00 00 00 00  0C 08 0 ................
00FFE020 02 00 00 00  40 00 00 80  00 00 0 ....@...........
00FFE030 02 02 00 00  60 00 02 01  00 00 0 ....`...........
00FFE040 00 00 00 00  00 00 00 00  00 08 0 ................
00FFE050 00 00 00 00  00 00 00 00  00 00 0 ................
00FFE060 08 00 00 00  80 02 00 01  00 00 0 ................
00FFE070 02 00 80 00  00 00 00 90  00 00 0 ................
```

Figure 6-32 Memory Window (When "5. 12 The real-time RAM monitor function" Selected)



```
Addr     +0 +1 +2 +3  +4 +5 +6 +7  +8 +9 +A +B  +C +D +E +F
03FF7000 00 00 00 00  10 F0 00 00  20 E0 00 00  30 D0 00 00
03FF7010 40 C0 00 00  50 B0 00 00  60 A0 00 00  70 90 00 00
03FF7020 80 80 00 00  90 70 00 00  A0 60 00 00  B0 50 00 00
03FF7030 C0 40 00 00  D0 30 00 00  E0 20 00 00  F0 10 00 00
03FF7040 00 00 00 00  10 F0 00 00  20 E0 00 00  30 D0 00 00
03FF7050 40 C0 00 00  50 B0 00 00  60 A0 00 00  70 90 00 00
03FF7060 80 80 00 00  90 70 00 00  A0 60 00 00  B0 50 00 00
03FF7070 C0 40 00 00  D0 30 00 00  E0 20 00 00  F0 10 00 00
```

- Opening
- Explanation of each area

- [View]menu (Memory window-dedicated items)
- Context menu
- Function buttons

Opening

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

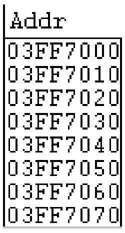
Explanation of each area

(1) Addr

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

(3) 0 1 2 3 ... ...

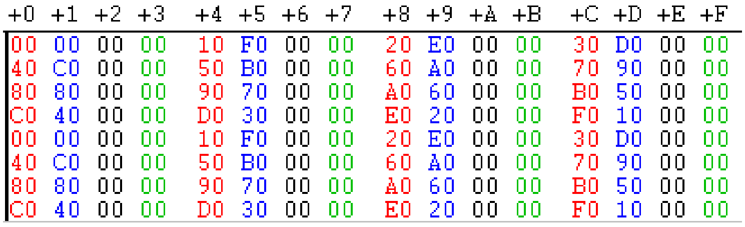
(1) Addr



```
Addr
03FF7000
03FF7010
03FF7020
03FF7030
03FF7040
03FF7050
03FF7060
03FF7070
```

This area displays memory addresses.

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



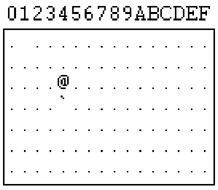
```
+0 +1 +2 +3   +4 +5 +6 +7   +8 +9 +A +B   +C +D +E +F
00 00 00 00   10 F0 00 00   20 E0 00 00   30 D0 00 00
40 C0 00 00   50 B0 00 00   60 A0 00 00   70 90 00 00
80 80 00 00   90 70 00 00   A0 60 00 00   B0 50 00 00
C0 40 00 00   D0 30 00 00   E0 20 00 00   F0 10 00 00
00 00 00 00   10 F0 00 00   20 E0 00 00   30 D0 00 00
40 C0 00 00   50 B0 00 00   60 A0 00 00   70 90 00 00
80 80 00 00   90 70 00 00   A0 60 00 00   B0 50 00 00
C0 40 00 00   D0 30 00 00   E0 20 00 00   F0 10 00 00
```

This area is used to display and change memory contents, and to display the access status (refer to "5. 7. 4 Access monitor function (when IECUBE connected)").

Black	No access
Green	Read
Red	Write
Blue	Read & write

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

(3) 0 1 2 3 ... ...



```
0123456789ABCDEF
. ..............
................
....@...........
....`...........
................
................
................
```

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

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

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

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

[View]menu (Memory window-dedicated items)

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

Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays hexadecimal numbers (default).
Nibble	Displays in 4-bit units.
Byte	Displays in 8-bit units (default).
Half Word	Displays in 16-bit units.
Word	Displays in 32-bit units.
Ascii	Selects whether ASCII characters are displayed or not. Checked: Displayed Not checked: No display (default)
Little Endian	Displays in little endian (default).
Big Endian	Displays in big endian.
Access Monitoring	This area is used to set about Access monitor function (when IECUBE connected).
Clear	Clears the display color through the access monitor function (when IECUBE connected).
Accumulative	Enables/disables cumulative display of access status (memory content change) (when IECUBE is connected). Checked : Cumulative display of memory contents changes Not checked : Display of only memory contents changes from previous update

Context menu

Move...	Moves the display position. Opens the Address move dialog box.
RRM Setting...	Opens the RRM dialog box.

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).
Half Word	Displays in 16-bit units.
Word	Displays in 32-bit units.
Ascii	Selects whether ASCII characters are displayed or not. Checked: Displayed Not checked: No display (default)
Clear Access Monitor	Clears the display color through the access monitor function (when IECUBE connected).
Accumulative	Enables/disables cumulative display of access status (memory content change) (when IECUBE is connected). Enables/disables cumulative display of access status (memory content change). Checked : Cumulative display of memory contents changes Not checked : Display of only memory contents changes from previous update

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 result 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 (during search)	Stops searching.
Refresh	Updates the contents of the window with the latest data.
Modify...	Opens the DMM dialog box.
Close	Closes this window.

Memory Search dialog box

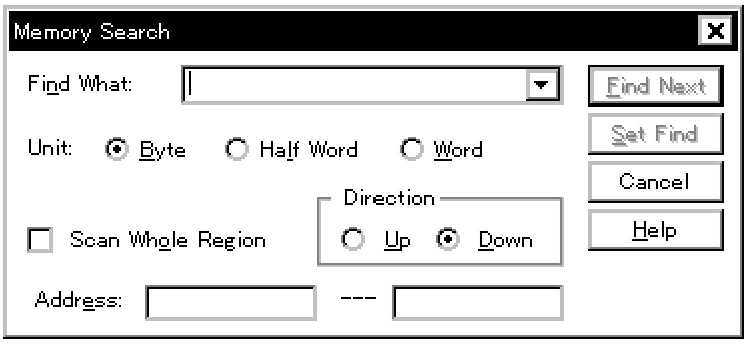
This dialog box is used to search the memory contents of the part of the Memory window at which the cursor is located. (Refer to "5. 7 Memory Manipulation Function".)

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 ASCI character string, and the contents of these respective areas are searched.

By setting each item and then clicking the <Find Next> button, searching can be started. By clicking the <Set Find> button, the direction buttons ("<<" and ">>") in the Memory window can be used for the search.

Caution Non-mapped, peripheral I/O registers, and I/O protect areas are not searched.

Figure 6-33 Memory Search Dialog Box



- Opening
- Explanation of each area
- Function buttons

Opening

When the Memory window is the current window, select [View] menu -> [Search...], or click the <Search...> button in the same window.

Explanation of each area

(1) Find What

(2) Unit

(3) Scan Whole Region

(4) Direction

(5) Address

(1) Find What

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

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

(b) When searching (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

Unit: ● Byte ○ Half Word ○ Word

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

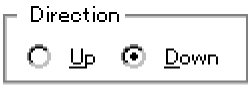
Byte	Searches the data as 8-bit data (default).
Half Word	Searches the data as 16-bit data.
Word	Searches the data as 32-bit data.

(3) Scan Whole Region

□ Scan Whole Region

This should be checked to search the entire specified range.

(4) Direction



Direction

○ Up ● Down

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

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".)

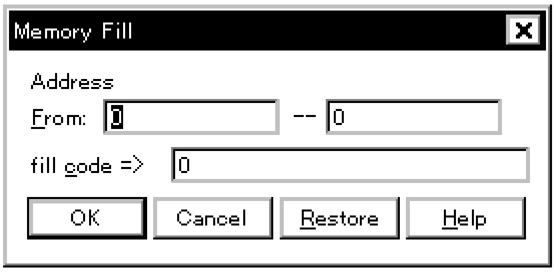
Function buttons

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(during searching)	Stops searching.
Cancel	Closes this dialog box. (During searching, this button is replaced by the <Stop> button.)
Help	Displays this dialog box online help files.

Memory Fill dialog box

This dialog box is used to fill the memory contents in the Memory window with specified codes (fill code). (Refer to "5. 7 Memory Manipulation Function".)

Figure 6-34 Memory Fill Dialog Box



From: 0 -- 0

fill code => 0

- Opening
- Explanation of each area
- Function buttons

Opening

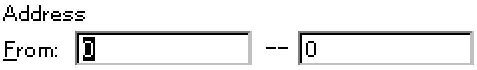
Select [Edit] menu -> [Memory] -> [Fill...].

Explanation of each area

(1) Address

(2) fill code

(1) Address



From: 0 -- 0

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

The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol or expression. (Refer to "Table 5-6 Specifying Symbols".)

(2) fill code

fill code => 0

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

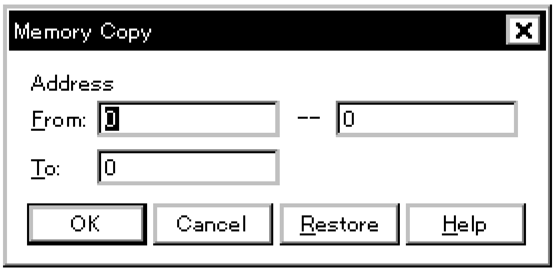
Function buttons

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

Memory Copy dialog box

This dialog box is used to copy the memory contents in the Memory window . (Refer to "5. 7 Memory Manipulation Function".)

Figure 6-35 Memory Copy Dialog Box



Address

From: 0 -- 0

To: 0

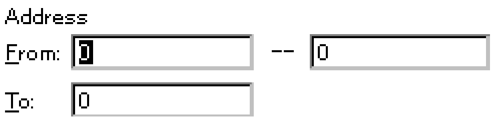
- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

(1) Address



Address

From: 0 -- 0

To: 0

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".)

(a) From:

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

(b) To:

Specify start address of the copy destination.

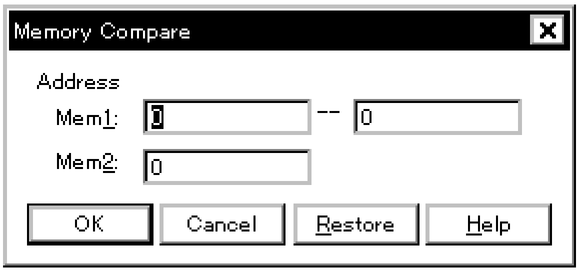
Function buttons

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

Memory Compare dialog box

This dialog box is used to compare the memory contents in the Memory window .(Refer to "5. 7 Memory Manipulation Function".)

Figure 6-36 Memory Compare Dialog Box



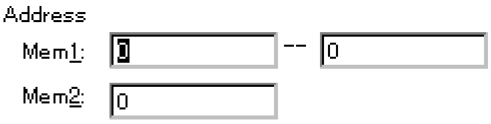
- Opening
- Explanation of each area
- Function buttons

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".)

(a) Mem1

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

(b) Mem2

Specify the start address of the comparison destination.

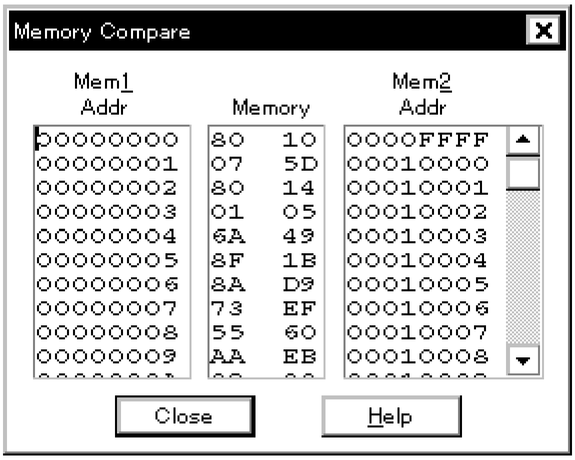
Function buttons

OK	Compares the memory contents in accordance with a given condition. If no difference is found as a result of comparison, " Wf200: No difference encountered." is displayed. If a difference is found, the Memory Compare Result dialog box is opened.
Stop(during comparison)	Stops memory comparison.
Cancel	Closes this dialog box. (During comparison, this button is replaced by the <Stop> button.)
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

Memory Compare Result dialog box

This dialog box is displayed if any difference is found in the memory contents when the memory has been compared in the Memory Compare dialog box .(Refer to "5. 7 Memory Manipulation Function".)

Figure 6-37 Memory Compare Result Dialog Box

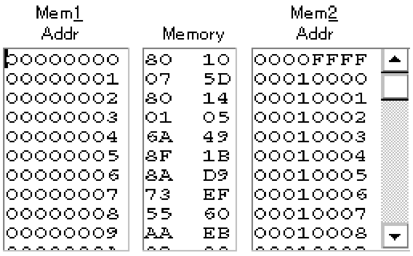


Mem1 Addr	Memory		Mem2 Addr
00000000	80	10	0000FFFF
00000001	07	5D	00010000
00000002	80	14	00010001
00000003	01	05	00010002
00000004	6A	49	00010003
00000005	8F	1B	00010004
00000006	8A	D9	00010005
00000007	73	EF	00010006
00000008	55	60	00010007
00000009	AA	EB	00010008

- Explanation of each area
- Function buttons

Explanation of each area

(1) (comparison result display area)



Mem1 Addr	Memory		Mem2 Addr
00000000	80	10	0000FFFF
00000001	07	5D	00010000
00000002	80	14	00010001
00000003	01	05	00010002
00000004	6A	49	00010003
00000005	8F	1B	00010004
00000006	8A	D9	00010005
00000007	73	EF	00010006
00000008	55	60	00010007
00000009	AA	EB	00010008

This area displays the results of comparing the memory. Only differences that have been found as a result of comparison are displayed.

(a) Mem1 Addr

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

(b) Memory

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

(c) Mem2 Addr

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

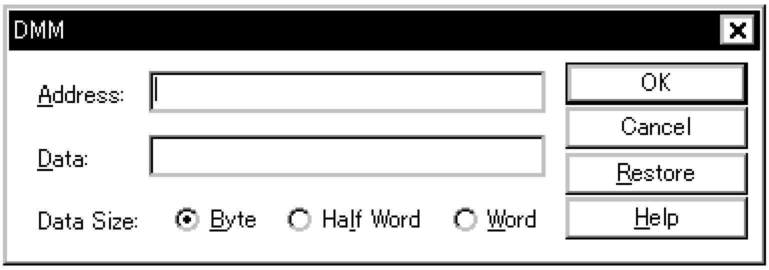
Function buttons

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

DMM dialog box

This dialog box is used to set addresses and data for DMM (Dynamic Memory Modification). (Refer to "5. 7. 3 Modifying memory contents (DMM function)".)

Figure 6-38 DMM Dialog Box



- Opening
- Explanation of each area
- Function buttons

Opening

Select [Edit] menu -> [Memory] -> [Modify...], or click the <Modify...> button in the Memory window.

Explanation of each area

(1) Address

(2) Data

(3) Data Size

(1) Address

This area is used to specify the 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".)

(2) Data

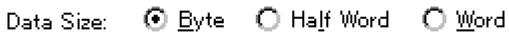
Data:

This area is used to specify the data to be written to the memory address specified in (1) Address.

The default radix for inputting a numeric value is hexadecimal.

One binary data string can be specified at one time. Specify the data size in (3) Data Size.

(3) Data Size



Data Size: ● Byte ○ Half Word ○ Word

This area is used to specify the size of the data specified in (2) Data to be written.

Byte	Writes data as 8-bit data.
Half Word	Writes data as 16-bit data.
Word	Writes data as 32-bit data.

Function buttons

OK	Writes the data in accordance with a given condition.
Cancel	Closes this dialog box.
Restore	Restores the status before this dialog box was opened.
Help	Displays this dialog box online help files.

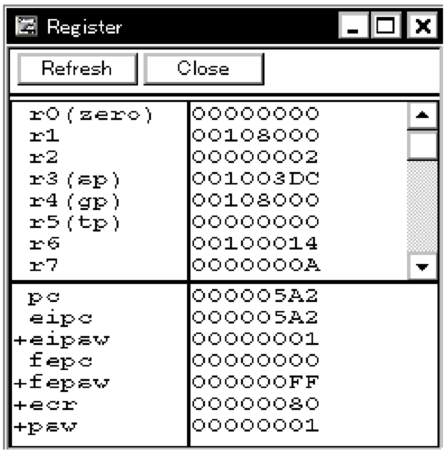
Register window

This window is used to display and change registers (program registers, system registers) (refer to "5. 8 Register Manipulation Function").

Other operations using Context menu, Function buttons, etc., can be performed in this window.

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

Figure 6-39 Register Window



```
 r0(zero)   00000000
 r1         00108000
 r2         00000002
 r3(sp)     001003DC
 r4(gp)     00108000
 r5(tp)     00000000
 r6         00100014
 r7         0000000A

 pc         000005A2
 eipc       000005A2
+eipsw      00000001
 fepc       00000000
+fepsw      000000FF
+ecr        00000080
+psw        00000001
```

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

Opening

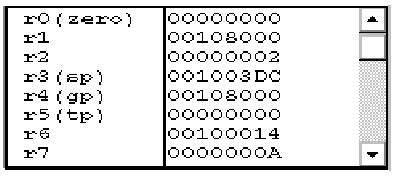
Click the **Reg** button, or select [Browse] menu -> [Register].

Explanation of each area

(1) Upper field (program register display area)

(2) Lower field (system register display area)

(1) Upper field (program register display area)



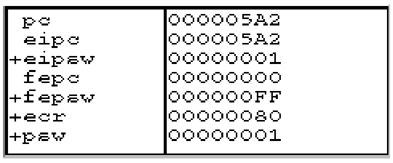
```
r0(zero)   00000000
r1         00108000
r2         00000002
r3(sp)     001003DC
r4(gp)     00108000
r5(tp)     00000000
r6         00100014
r7         0000000A
```

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

Caution When overflow of a register occurs due to an illegal value entered by the user, the register will be updated with a value of 0xFFFFFFFF.

(2) Lower field (system register display area)



```
 pc       000005A2
 eipc     000005A2
+eipsw    00000001
 fepc     00000000
+fepsw    000000FF
+ecr      00000080
+psw      00000001
```

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

Caution When overflow of a register occurs due to an illegal value entered by the user, the register will be updated with a value of 0xFFFFFFFF.

[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).
Pick Up	Displays only the registers selected in the Register Select dialog box.
Select...	Opens the Register Select dialog box.

Context menu

Add Watch...	Registers a selected character string to the Watch window. 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 (default).
Pick Up	Displays only the registers selected in the Register Select dialog box.
Select...	Selects the register to be displayed Opens the Register Select dialog box.

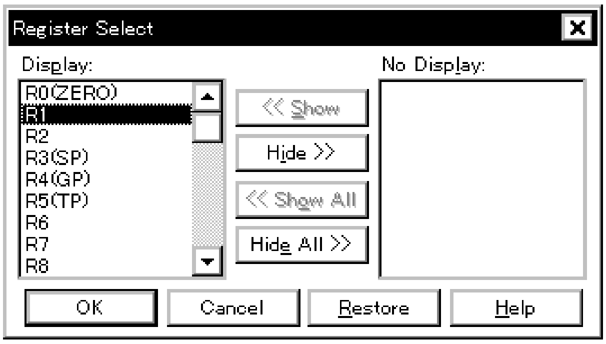
Function buttons

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

Register Select dialog box

This dialog box is used to select registers that are not displayed in the Register window .(Refer to "5. 8 Register Manipulation Function".)

Figure 6-40 Register Select Dialog Box



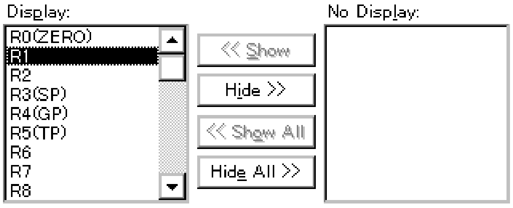
- Opening
- Explanation of each area
- Function buttons

Opening

When the Register window is the current window, select [View] menu -> [Select...].

Explanation of each area

(1) Display, No Display



This area is used to select registers that are displayed in the Register window, and those that are not.

(a) Display

Registers displayed in the Register window.

(b) No Display

Registers not displayed in the Register window.

(c) Button

The following buttons are used to change register to be displayed in the Register window.

<< Show	Moves the register selected from the (b) No Display list to (a) Display.
Hide >>	Moves the register selected from the (a) Display list to (b) No Display.
<< Show All	Moves all registers to (a) Display.
Hide All >>	Moves all registers to (b) No Display.

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

Function buttons

OK	Reflects the selection in this dialog box in the Register 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.

IOR window

This window is used to display and change the contents of peripheral I/O registers 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.

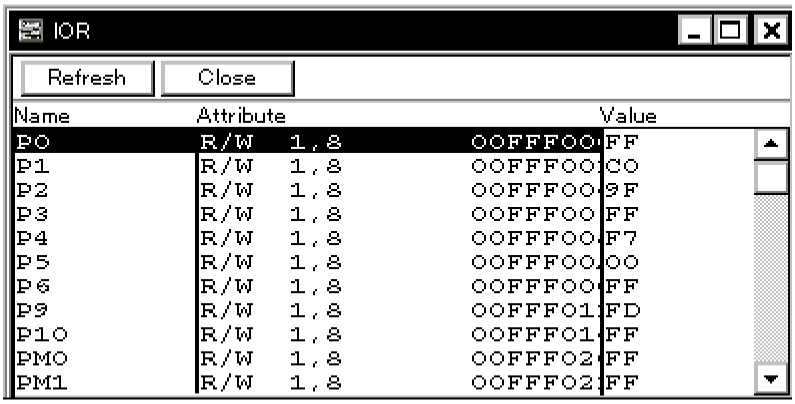
Caution1 However, that the values of read-only peripheral I/O registers and I/O ports cannot be changed. In addition, the peripheral I/O registers and I/O ports that cause the device to operate when they are read are read-protected and therefore cannot be read. To read these registers, select a register, and select and execute [Compulsion Read] from the context menu.

Caution2 During user program execution, the IOR contents are updated at every sampling time set in the Extended Option dialog box (refer to "5. 12 The real-time RAM monitor function"). However, updating is not performed when the RAM monitor function is OFF.

Remark1 The display start position when the window is opened is as follows.
First time: Display from peripheral I/O registers of minimum address
Second and subsequent times: Display from first peripheral I/O registers when window was last closed

Remark2 If the device supports programmable I/O registers and a programmable I/O area has been set in the Configuration dialog box, the programmable I/O registers and expansion peripheral I/O registers are also displayed. If the value of an I/O port address is defined, the I/O port name is displayed in light color.

Figure 6-41 IOR Window



Name	Attribute			Value
P0	R/W	1,8	00FFF00	FF
P1	R/W	1,8	00FFF00	C0
P2	R/W	1,8	00FFF00	9F
P3	R/W	1,8	00FFF00	FF
P4	R/W	1,8	00FFF00	F7
P5	R/W	1,8	00FFF00	00
P6	R/W	1,8	00FFF00	FF
P9	R/W	1,8	00FFF01	FD
P10	R/W	1,8	00FFF01	FF
PM0	R/W	1,8	00FFF02	FF
PM1	R/W	1,8	00FFF02	FF

- Opening
- Explanation of each area
- Context menu
- Function buttons

Opening

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

Explanation of each area

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

(1) Name

This area displays the names of peripheral I/O registers 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 peripheral I/O registers and I/O ports.

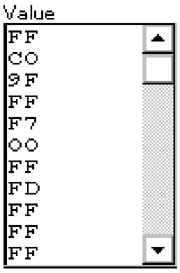
This area displays the read/write attributes, access types, and displays and absolute addresses from the left side.

When the bit peripheral I/O registers 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 peripheral I/O registers, execute [View] menu -> [Compulsion Read]. Even a write-only peripheral I/O registers 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 Half Word units.
32	Can be accessed in Word units.

(3) Value



```
Value
FF
C0
9F
FF
F7
00
FF
FD
FF
FF
FF
```

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

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

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

Values are changed through direct input. The location to be changed is displayed in red and the contents of the contents of the change are written into the target memory when the Enter key is pressed. The previous value can be canceled by the ESC key.

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

The value of read-protected peripheral I/O registers and I/O ports can be read by selecting Context menu -> [Compulsion Read].

[View] menu (IOR 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 the (2) Attribute.
Pick Up	Displays only the registers selected in the IOR Select dialog box.
Select...	Opens the IOR Select dialog box.
Compulsion Read	Forcibly reads the peripheral I/O registers 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 IOR Select dialog box.
Select...	Opens the IOR Select dialog box.
Compulsion Read	Forcibly reads the peripheral I/O registers 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

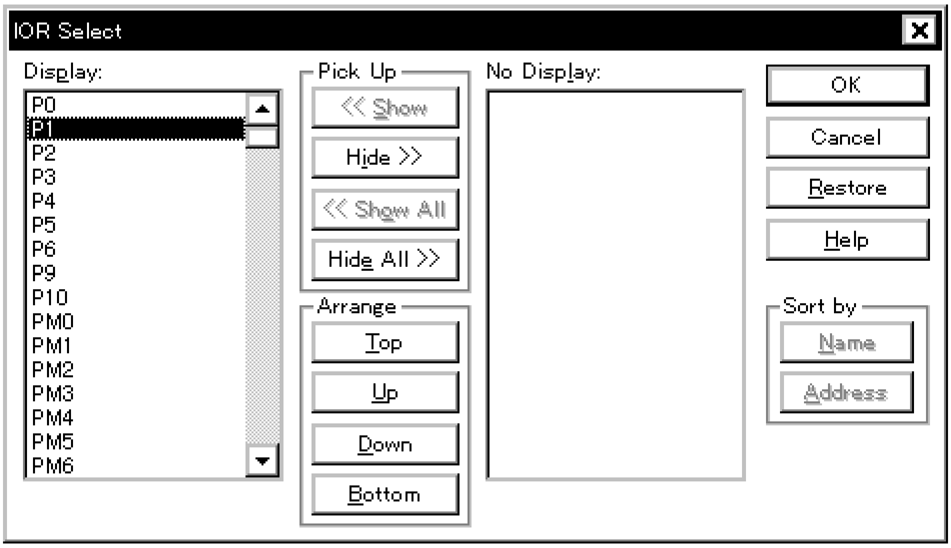
Refresh	Updates the contents of this window with the latest watch data.
Close	Closes this window.

IOR Select dialog box

This dialog box is used to select peripheral I/O registers and I/O ports that are not displayed the IOR 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 IOR Select Dialog Box



- Opening
- Explanation of each area
- Function buttons

Opening

When the IOR window is the current window, select [View] menu -> [Select...].

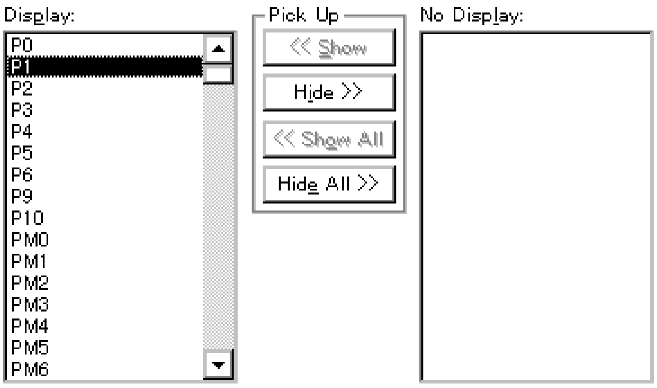
Explanation of each area

(1) Display, Pick Up, No Display

(2) Arrange

(3) Sort by

(1) Display, Pick Up, No Display



This area is used to select peripheral I/O registers / I/O ports that are displayed in the IOR window, and those that are not.

(a) Display

The peripheral I/O registers / I/O ports displayed in the IOR window.

(b) No Display

The peripheral I/O registers / I/O ports not displayed in the IOR window.

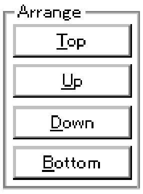
(c) Pick Up

The following buttons are used to change peripheral I/O registers/ I/O ports to be displayed in the IOR window.

<< Show	Moves peripheral I/O registers / I/O ports selected from the (b) No Display list to (a) Display.
Hide >>	Moves peripheral I/O registers / I/O ports selected from the (a) Display list to (b) No Display.
<< Show All	Moves all peripheral I/O registers / I/O ports to (a) Display.
Hide All >>	Moves all peripheral I/O registers / I/O ports to (b) No Display.

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

(2) Arrange

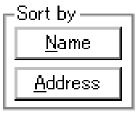


The following buttons are used to change the display sequence in the (a) Display.

Top	Moves the selected peripheral I/O registers or I/O port to the top of the list.
Up	Moves the selected peripheral I/O registers or I/O port one line up.

Down	Moves the selected peripheral I/O registers or I/O port one line down.
Bottom	Moves the selected peripheral I/O registers or I/O port to the bottom of the list.

(3) Sort by



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

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

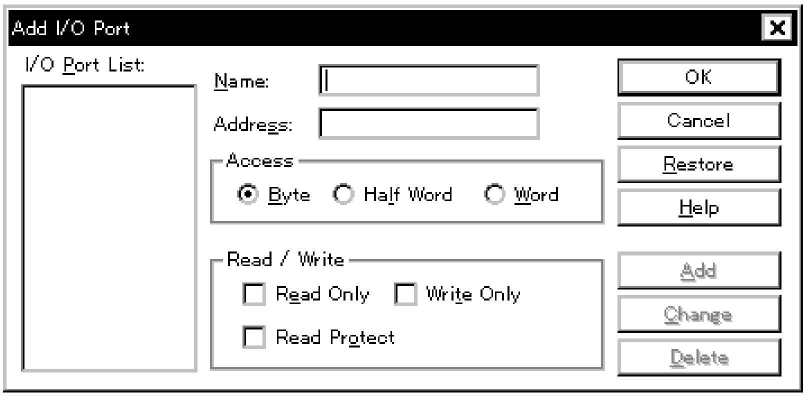
Function buttons

OK	Reflects the selection in this dialog box in the IOR 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 IOR 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

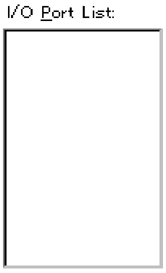
(2) Name

(3) Address

(4) Access

(5) Read/Write

(1) I/O Port List



This area lists the I/O ports currently registered.

If a new I/O port is registered, it is added to this list. An I/O port already registered can be selected and changed or deleted by Function buttons.

(2) Name

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

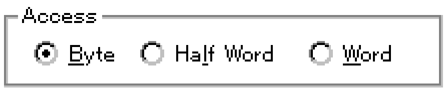
(3) Address

This area is used to specify the address of the I/O port to be added.

The default radix for inputting a numeric value is hexadecimal. An address can be also specified by a symbol. (Refer to "Table 5-6 Specifying Symbols".)

Area can be specified	Target area , peripheral I/O registers area

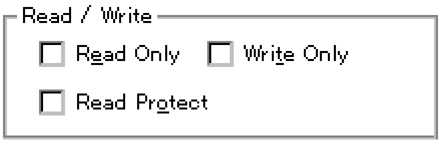
(4) Access



This area is used to select the access size of the I/O port to be added.

Byte	8-bit unit (default)
Half Word	16-bit unit
Word	32-bit unit

(5) Read/Write



This area is used to specify the access attribute of the I/O port to be added.

In the default condition, all the attributes are not checked (i.e., the I/O port can be both read and written).

Function buttons

OK	Reflects the result of addition in the IOR 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 the (1) I/O Port List.
Delete	Deletes the I/O port selected in the (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. 11 Event function", "5. 9 Timer Function".)

The execution time display area can be constantly displayed as the Timer Result dialog box by clicking the <View Always> button.

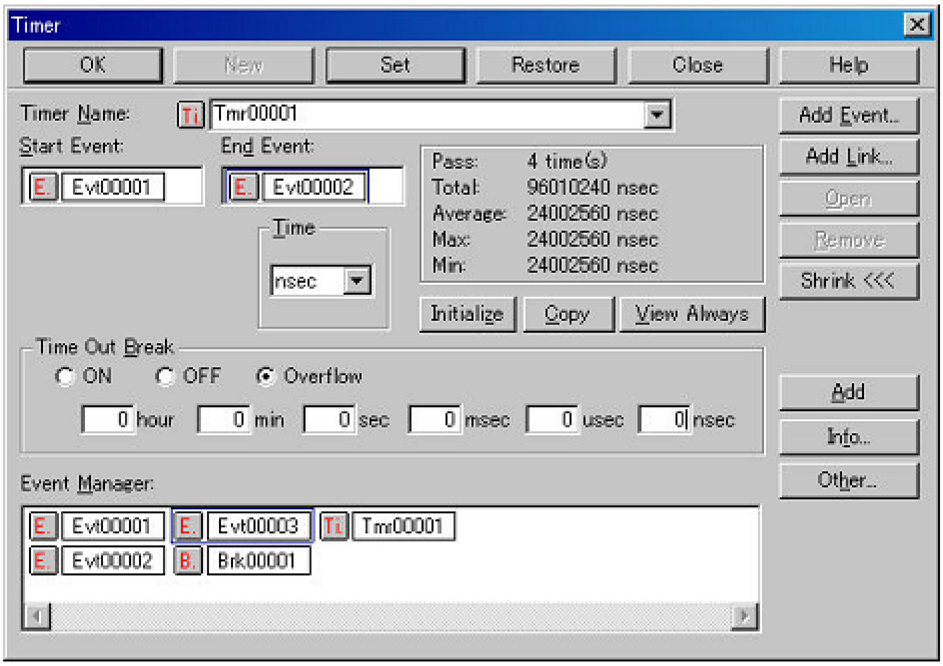
Registration and setting of timer event conditions is done by setting each item (256 items max.) in this dialog box and then pressing the <OK> button. The registered timer event conditions are managed by the Event Manager.

The number of timer event conditions that can be simultaneously used (validated) is limited (refer to "5. 11. 4 Number of enabled events for each event condition").

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

Remark The measurement result display contents are updated at each sampling time of the The real-time RAM monitor function, even during user program execution.

Figure 6-44 Timer Dialog Box



Timer Name: Ti Tmr00001

Start Event: E. Evt00001

End Event: E. Evt00002

Time: nsec

Pass: 4 time(s)
Total: 96010240 nsec
Average: 24002560 nsec
Max: 24002560 nsec
Min: 24002560 nsec

Time Out Break
○ ON ○ OFF ● Overflow
0 hour 0 min 0 sec 0 msec 0 usec 0 nsec

Event Manager:
E. Evt00001 E. Evt00003 Ti Tmr00001
E. Evt00002 B. Brk00001

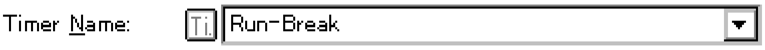
- Opening
- Explanation of each area
- Function buttons (Related event function)

Opening

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

Explanation of each area

(1) Timer Name



Timer Name: Ti Run-Break

This area is used to set a timer event name.

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

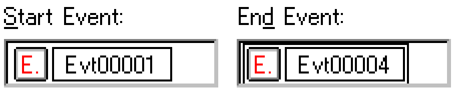
To display the contents of an already created event condition, select from the drop-down list.

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 5-16 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



Start Event: E. Evt00001

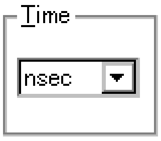
End Event: E. Evt00004

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

The number of event conditions that can be registered in this area is one for each of the start and end conditions.

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. 11. 3 Setting event conditions".

(3) Time



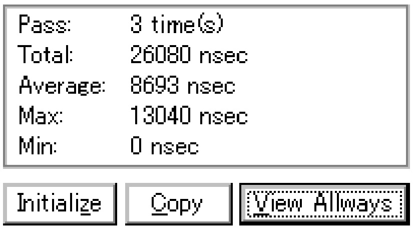
Time

nsec

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

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

(4) Execution time display area



Pass: 3 time(s)
Total: 26080 nsec
Average: 8693 nsec
Max: 13040 nsec
Min: 0 nsec

This area displays the result of measuring the execution time of the program (refer to "Table 6-16 Measurable Values").

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

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

Table 6-16 Measurable Values

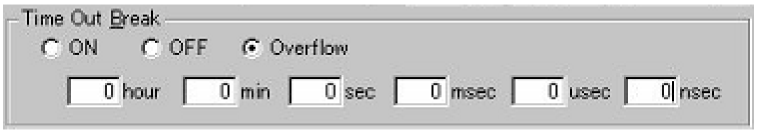
Connected IE	Measurable Execution Time	Measurable Execution Count
IECUBE	Approx. 2.8 minutes max. (External clock = 50 MHz, 1 division, resolution = 20 ns) Approx. 195.2 hours max. (External clock = 50 MHz, 4K division, resolution = 81920 ns) **Note:** The measurable execution time differs according to "Table 6-10 Relationship Between Division Ratio and Maximum Measurement Time (Timer Count (Timer))".	65535 times max. (16 bit)
N-Wire CARD	Approx. 3 minutes and 30 seconds max. (Resolution = 100 ns) **Note**	-

Note Only during display to the status bar in the Main window

(a) Function buttons

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

(5) Time Out Break



Time Out Break

○ ON ○ OFF ● Overflow

0 hour 0 min 0 sec 0 msec 0 usec 0 nsec

This area is used to set the operation and its duration when the measurement time in the block specified in the (2) Start Event, End Event (time from occurrence of start event until occurrence of end event) measurement time, exceed the specified time (timeout time).

(a) Time Out Break settings

Settings related to timeout breaks.

ON	If the timeout time specified in (b) Timeout time setting is exceeded, a timeout break occurs.
OFF	No timeout break occurs (default).
Overflow	A timeout break occurs at the largest measurable time (refer to "Table 6-16 Measurable Values").

(b) Timeout time setting

Timeout time setting can be done through direct input.

Input up to the measurable maximum time (refer to "Table 6-16 Measurable Values") is possible.

This setting is invalid when "OFF" or "Overflow" was selected in (a) Time Out Break settings .

(6) Event Manager

This area is used to display the list of the events registered. (Refer to "Table 5-16 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

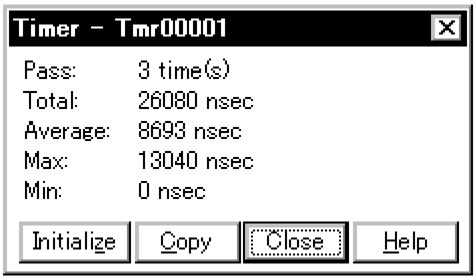
This dialog box displays the result of measuring the execution time (Refer to "5. 9 Timer Function".)

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. 11. 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 The real-time RAM monitor function, even during user program execution.

Figure 6-45 Timer Result Dialog Box



Timer - Tmr00001

Pass: 3 time(s)
Total: 26080 nsec
Average: 8693 nsec
Max: 13040 nsec
Min: 0 nsec

- 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 "(4) 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 window

IECUBE

This window used to display the trace results. (Refer to "5. 10 Trace Function (When IECUBE Connected)".)

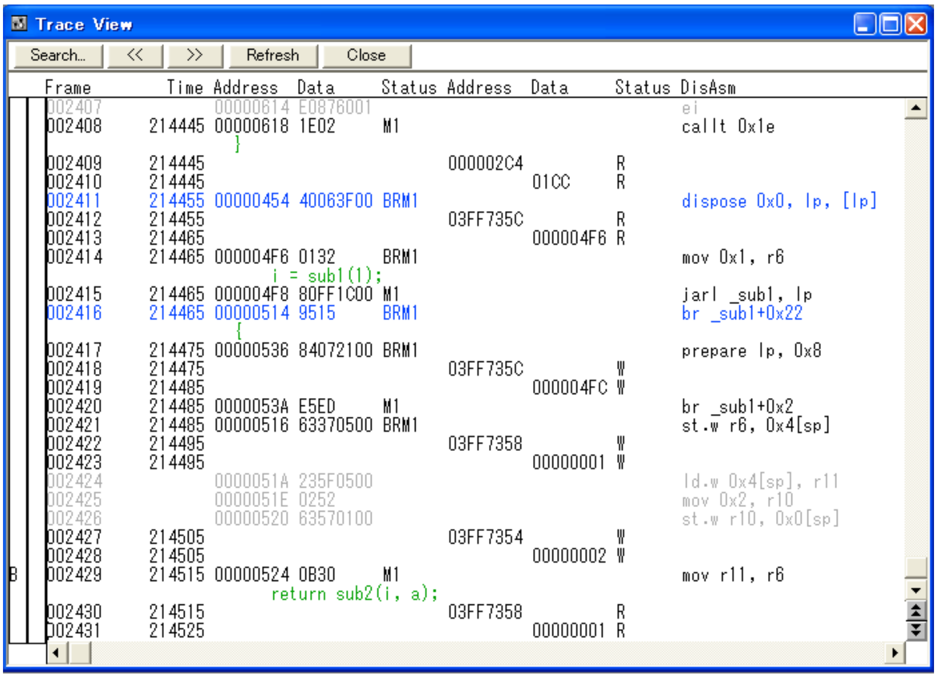
Display updates are performed during breaks or during step execution.

This window has Mixed display mode (Trace window).

Also, It has "5. 14. 3 Trace Result with Linking Window (when IECUBE connected)".

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

Figure 6-46   Trace Window



```
  Frame         Time Address  Data     Status Address  Data     Status DisAsm
  002407             00000614 E0876001                                 ei
  002408      214445 00000618 1E02     M1                              callt 0x1e
                        }
  002409      214445                          000002C4          R
  002410      214445                                   01CC     R
  002411      214455 00000454 40063F00 BRM1                            dispose 0x0, lp, [lp]
  002412      214455                          03FF735C          R
  002413      214465                                   000004F6 R
  002414      214465 000004F6 0132     BRM1                            mov 0x1, r6
                                i = sub1(1);
  002415      214465 000004F8 80FF1C00 M1                              jarl _sub1, lp
  002416      214465 00000514 9515     BRM1                            br _sub1+0x22
                        {
  002417      214475 00000536 84072100 BRM1                            prepare lp, 0x8
  002418      214475                          03FF735C          W
  002419      214485                                   000004FC W
  002420      214485 0000053A E5ED     M1                              br _sub1+0x2
  002421      214485 00000516 63370500 BRM1                            st.w r6, 0x4[sp]
  002422      214495                          03FF7358          W
  002423      214495                                   00000001 W
  002424             0000051A 235F0500                                 ld.w 0x4[sp], r11
  002425             0000051E 0252                                     mov 0x2, r10
  002426             00000520 63570100                                 st.w r10, 0x0[sp]
  002427      214505                          03FF7354          W
  002428      214505                                   00000002 W
B 002429      214515 00000524 0B30     M1                              mov r11, r6
                                return sub2(i, a);
  002430      214515                          03FF7358          R
  002431      214525                                   00000001 R
```

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

Opening

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

Explanation of each area

(1) Point mark display area

(2) Trace mode display area

(3) Trace result display 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.

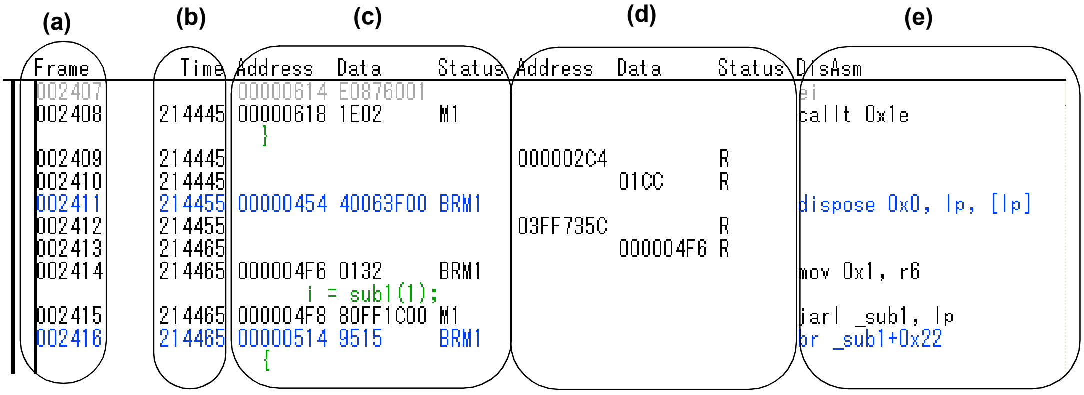
The mark displayed is not that during trace but an event mark that is set when the trace result is displayed.

(2) Trace mode display area

This area displays the type of trace mode.

A	a start or an end frame (section trace or qualify trace)
T	Delay trigger frame
M	DMA point access frame (DMA start point and end point)
N	Frame for which not all the trace data was fetched

(3) Trace result display area



```
Frame        Time Address   Data     Status Address   Data     Status DisAsm
002407            00000614 E0876001                                   ei
002408     214445 00000618 1E02     M1                                callt 0x1e
                      }
002409     214445                          000002C4            R
002410     214445                                    01CC      R
002411     214455 00000454 40063F00 BRM1                              dispose 0x0, lp, [lp]
002412     214455                          03FF735C            R
002413     214465                                    000004F6  R
002414     214465 000004F6 0132     BRM1                              mov 0x1, r6
                            i = sub1(1);
002415     214465 000004F8 80FF1C00 M1                                jarl _sub1, lp
002416     214465 00000514 9515     BRM1                              br _sub1+0x22
                      {
```

This area displays the trace results.

Complement frames are displayed in gray. In the case of complement frames, only (a) Frame , (c) Address Data Status (fetch access display) , and (e) DisAsm are displayed.

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

(a) Frame

This area displays the trace flame number.

(b) Time

This area displays the number of clocks required for the target chip from the execution start of the program to the execution start of an instruction of each frame or generation of memory access cause.

Complement frames are not displayed.

The display contents can be switched between clock count display and time display in the Trace Data Select dialog box.

For the relationship between the timer count division ratio and maximum measurement time, refer to "Table 6-8 Relationship Between Division Ratio and Maximum Measurement Time (Time Tag Counter (Trace))").

Caution If overflow occurs, the time tag maximum value is displayed in red.

(c) Address Data Status (fetch access display)

This area displays the result of fetching the program.

In the case of a complement frame, only (ii) Address Data is displayed.

(i) Status

The following types of statuses are available:

BRM1	Fetching of first byte of first instruction after branch If the fetch address is the start of the symbol, the first line is highlighted in blue. (The frame which is BRM1 and is also M1 is included.)
M1	Fetching of first byte of instruction
Blank	Any of the following - 6th byte and 8th byte fetch of the instruction - Data access frame (3 frames max., 2 frames when RRM function is selected) - Frame of the second instruction (when two instructions are executed at the same time) - Invalid flame

Remark When a 6-byte or 8-byte instruction code is displayed, the first 4 bytes are displayed in the first frame, and the other bytes are displayed in the second and third frames. If two instructions are executed at the same time, one frame is displayed on two lines. The instruction code at the lower address is displayed on the first line and the instruction code at the higher address is displayed on the second line.

(ii) Address Data

This area displays the address and data.

Address	Displays the fetch address
Data	Displays the fetch data

(d) Address Data Status (data access display)

This area displays the result of accessing data.

Complement frames are not displayed.

Status

R	Data read
W	Data write

(e) DisAsm

This area displays the result of disassembly (only when the (i) Status is BRM1 or M1).

To display instruction codes when two instructions are simultaneously executed, one instruction is displayed on the first line, and the other instruction is displayed on the second line.

[View]menu (Trace window-dedicated items)

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

Select...	Selects the contents to be displayed. Opens the Trace Data Select dialog box.
Mix	Specifies whether the source file are displayed in mixed display mode, or not displayed. Checked: Mixed display Not checked: No display (default)
Window Synchronize	Links the Trace window with the following windows: (Refer to "5. 14. 3 Trace Result with Linking Window (when IECUBE connected)".) A checked 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.
Mix	Specifies whether the source file are displayed in mixed display mode, or not displayed. Checked: Mixed display Not checked: No display (default)
Window Synchronize	Links the Trace window with the following windows: (Refer to "5. 14. 3 Trace Result with Linking Window (when IECUBE 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. 14. 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 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. 14. 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. 14. 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 Trace Search dialog box and searches trace results. The searched result will be highlighted in the Trace 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.
>>	Searches backward (downward on screen) for a trace result that satisfies the search condition set in the Trace Search dialog box.
Refresh	Updates the contents of the window with the latest data.
Close	Closes this window.

Trace Search dialog box

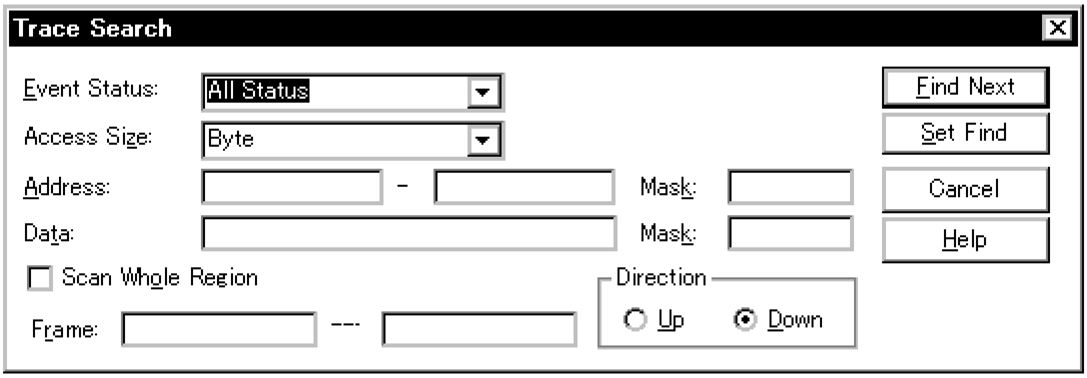
IECUBE

This dialog box is used to search in the Trace window. (Refer to "5. 10 Trace Function (When IECUBE 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 window can be used for the search.

Figure 6-47   Trace Search Dialog Box



Trace Search

Event Status: All Status

Access Size: Byte

Address: – Mask:

Data: Mask:

☐ Scan Whole Region

Frame: ---

Direction ○ Up ● Down

- Opening
- Explanation of each area
- Function buttons

Opening

When the Trace 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

(2) Access Size

(3) Address, Mask

(4) Data, Mask

(5) Scan Whole Region

(6) Direction

(7) Frame

(1) Event Status

Event Status: All 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

Access Size: Byte

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.

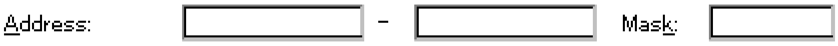
Byte	Searches for a data condition with 8-bit width (only during 8-bit access).
Half Word	Searches for a data condition with 16-bit width (only during 16-bit access).
Word	Searches for a data condition with 32-bit width (only during 32-bit access).
No Condition	Does not search based on access size (nothing can be input to the Data area).
Bit	Searches for a data condition with 1-bit width (only during 8-bit access). In this case, a data condition is searched with 1-bit width. Because of the operation of the in-circuit emulator, access to a bit is not directly traced; the ID850QB searches a dummy bit access by internally setting address conditions and data conditions as follows: **Input example:** Address: FE20.1 Data: 1 | **Setting of trace search** Address: FE20 Data: 00000010B Mask: 11111101B If another bit of the same address is accessed or if all the 8 bits of the same address are accessed, therefore, a trace data is searched in accordance with the specified status if the address and bit match the specified value of [address.bit].

Caution 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.

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
- Half Word if the data condition is set in 16-bit units
- Word if the address condition is set in 32-bit units
- No Condition if no data condition is specified

(3) Address, Mask



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 ("Table 5-6 Specifying Symbols").

The following can be set:

Table 6-17 Settable Range of Address Condition (Trace)

Settable range	Condition
0 <= address value <= 0xFFFFFFFF	None
0 <= mask value <= 0xFFFFFFFF	None

(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 the lower address and the higher address. Mask cannot be set.

(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

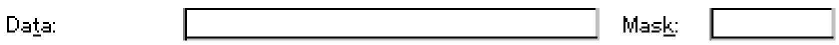
With this setting, addresses 0x4000 to 0x40FF satisfy the condition.

Example 2:

Address	0x4000 to 0x4000
Mask	0x101

With this setting, addresses 0x4000, 0x4001, 0x4100, and 0x4101 satisfy the condition.

(4) Data, Mask



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 the (2) Access Size. (Refer to "Table 6-23 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:

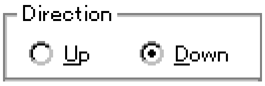
Data	0x4000
Mask	0x101

With this setting, addresses 0x4000, 0x4001, 0x4100, and 0x4101 satisfy the condition.

(5) Scan Whole Region

This should be checked to search the entire specified range.

(6) Direction



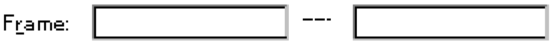
Direction

○ Up ● Down

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



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

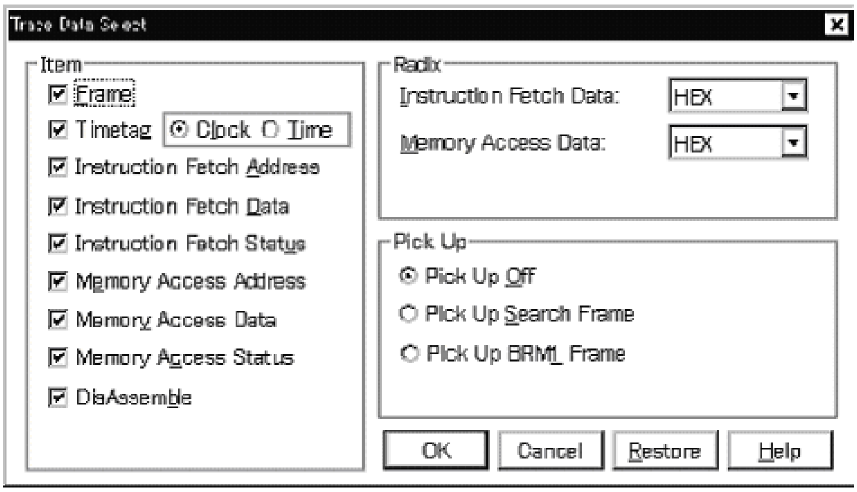
Find Next	Searches the specified data in accordance with a given condition. If the specified frame is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.
Cancel	Closes this dialog box.
Help	Displays this dialog box online help files.

Trace Data Select dialog box

IECUBE

This dialog box is used to select items to be displayed in the Trace window. (Refer to "5. 10 Trace Function (When IECUBE Connected)".)

Figure 6-48 Trace Data Select Dialog Box



Trace Data Select

Item
☑ Frame
☑ Timetag ● Clock ○ Time
☑ Instruction Fetch Address
☑ Instruction Fetch Data
☑ Instruction Fetch Status
☑ Memory Access Address
☑ Memory Access Data
☑ Memory Access Status
☑ DisAssemble

Radix
Instruction Fetch Data: HEX
Memory Access Data: HEX

Pick Up
● Pick Up Off
○ Pick Up Search Frame
○ Pick Up BRM1_Frame

- Opening
- Explanation of each area
- Function buttons

Opening

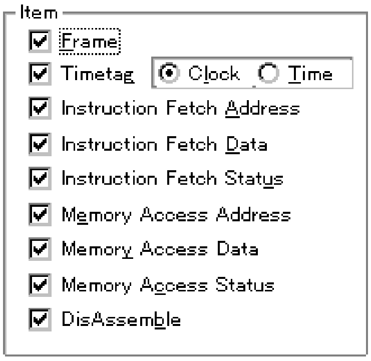
When the Trace window is the current window, select [View] menu -> [Select...].

Explanation of each area

(1) Item

(2) Radix

(1) Item

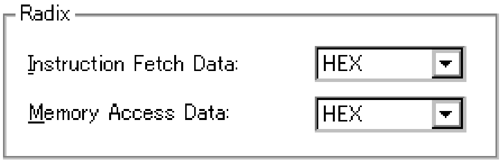


Item
☑ Frame
☑ Timetag ● Clock ○ Time
☑ Instruction Fetch Address
☑ Instruction Fetch Data
☑ Instruction Fetch Status
☑ Memory Access Address
☑ Memory Access Data
☑ Memory Access Status
☑ DisAssemble

This area is used to select items to be displayed in the Trace window. Displaying the following items may or may not be selected. The field checked is displayed.

Frame	(a) Frame field
Timetag	(b) Time field Whether the clock or time is displayed can be selected.
Instruction Fetch Address	Address field in (c) Address Data Status (fetch access display)
Instruction Fetch Data	Data field in (c) Address Data Status (fetch access display)
Instruction Fetch Status	Status field in (c) Address Data Status (fetch access display)
Memory Access Address	Address field in (d) Address Data Status (data access display)
Memory Access Data	Data field in (d) Address Data Status (data access display)
Memory Access Status	Status field in (d) Address Data Status (data access display)
DisAssemble	(e) DisAsm field

(2) Radix



Instruction Fetch Data: HEX

Memory Access Data: HEX

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 (c) Address Data Status (fetch access display)
Memory Access Data	Data field in (d) Address Data Status (data access display)

HEX	Displays hexadecimal numbers. (default)
DEC	Displays decimal numbers.
OCT	Displays octal numbers.
Bin	Displays binary numbers.

Function buttons

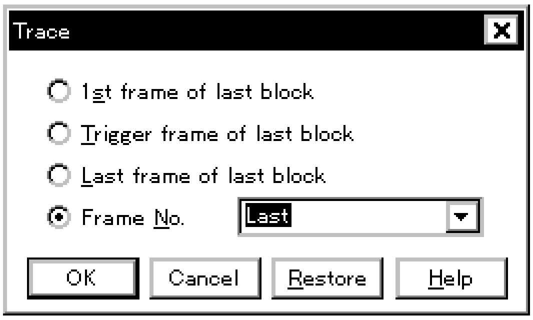
OK	Reflects the results of selection in this dialog box in the Trace window.
Cancel	Closes this dialog box.
Restore	Restores the original status.
Help	Displays this dialog box online help files.

Trace move dialog box

IECUBE

This dialog box is used to specify the position from which displaying the Trace window is started. (Refer to "5. 10 Trace Function (When IECUBE Connected)".)

Figure 6-49 Trace Move Dialog Box



○ 1st frame of last block
○ Trigger frame of last block
○ Last frame of last block
● Frame No. Last

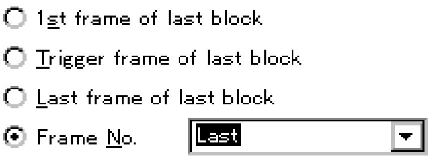
- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

(1) Frame selection area



○ 1st frame of last block
○ Trigger frame of last block
○ Last frame of last block
● Frame No. Last

This area is used to specify the frame at the destination.

1st frame of last block	Moves the display start position to the first frame in the newest block frame of trace data. The display start position is moved to the first frame of the trace data when using an in-circuit emulator without a block frame.
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. 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. In addition, frame number can also be specified in the following format. Up to 16 input histories can be recorded.

Table 6-18 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	Moves the display start position to the first 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]
Bottom	B	Moves the display start position to the last frame

Function buttons

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

Trace dialog box

IECUBE

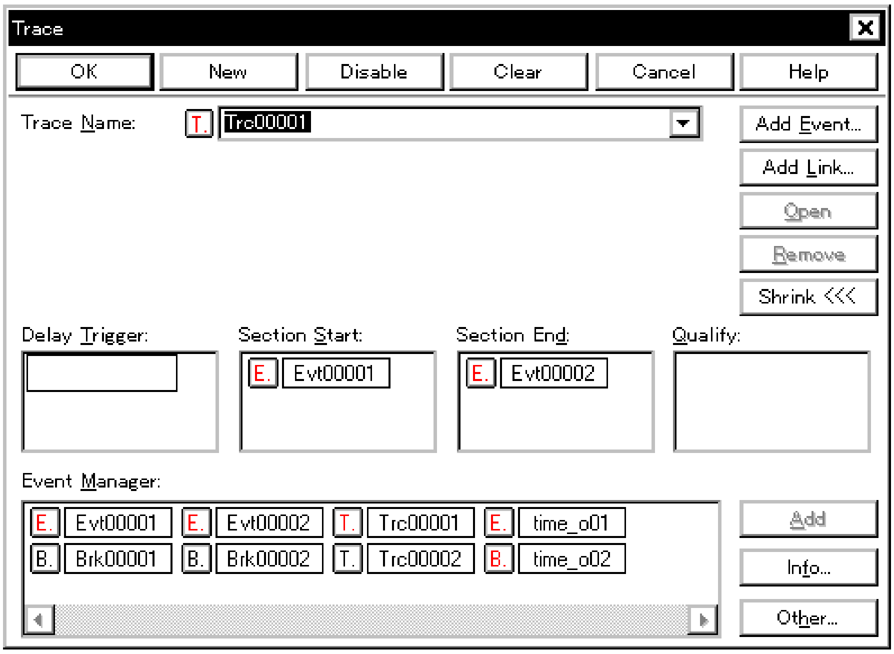
This dialog box is used to register, set, and display trace event conditions. (Refer to "5. 11 Event function", "5. 10 Trace Function (When IECUBE Connected)".)

The trace event conditions for when performing conditional trace are specified in this dialog box (refer to "Table 5-13 Types of Conditional Trace").

Registration and setting of trace event conditions is done by setting each item (256 items max.) in this dialog box and then pressing 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. 11. 4 Number of enabled events for each event condition").

Figure 6-50 Trace Dialog Box



Trace Name: T. Trc00001

Delay Trigger:

Section Start: E. Evt00001

Section End: E. Evt00002

Qualify:

Event Manager:

E. Evt00001 E. Evt00002 T. Trc00001 E. time_o01

B. Brk00001 B. Brk00002 T. Trc00002 B. time_o02

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

Opening

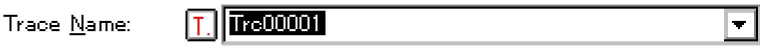
Click the **Trc** button, or select [Event] menu -> [Trace...].

Explanation of each area

(1) Trace Name

(2) Delay Trigger

(1) Trace Name



Trace Name: T. Trc00001

This area is used to set a trace event name.

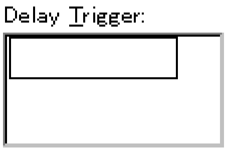
Directly input an alphanumeric string of up to eight characters as a name.

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 5-16 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) Delay Trigger



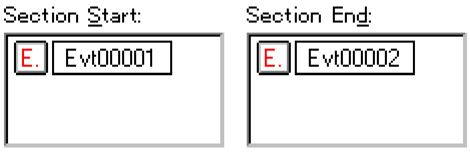
Delay Trigger:

This area is used to set an event condition for a delay trigger (refer to "5. 10. 6 Setting conditional trace").

For the number of items that can be set to this area, refer to "Table 6-19 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. 11. 3 Setting event conditions".

(3) Section Start, Section End



Section Start: E. Evt00001

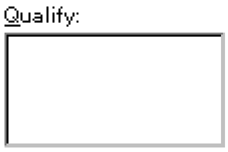
Section End: E. Evt00002

This area is used to set event conditions for starting and stopping a section trace (refer to "5. 10. 6 Setting conditional trace").

For the number of items that can be set to this area, refer to "Table 6-19 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. 11. 3 Setting event conditions".

(4) Qualify



Qualify:

This area is used to set an event condition for a qualify trace (refer to "5. 10. 6 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: The number of event link conditions that can be set in this area, refer to "5. 11. 4 Number of enabled events for each event condition"

Table 6-19 Number of Events Settable

Connected IE	Total (execution/access)
IECUBE	14 (8[a]/6)

a. Can be used to events after execution

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. 11. 3 Setting event conditions".

(5) Event Manager

This area is used to display the list of the events registered. (Refer to "Table 5-16 Event icon", "(4) Manipulation in event manager area".)

Function Buttons

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

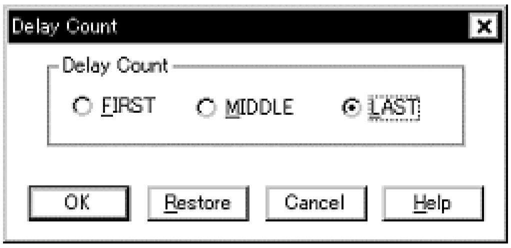
Delay Count dialog box

IECUBE

This dialog box is used to set or display delay count values. (Refer to "5. 10 Trace Function (When IECUBE 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. 6 Setting conditional trace".)

Figure 6-51 Delay Count Dialog Box



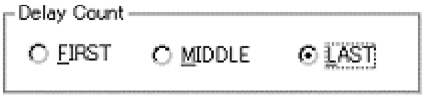
- Opening
- Explanation of each area
- Function buttons

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.

Caution The delay count value differs according to the trace memory size specified in the Extended Option dialog box.

Example: When the trace memory size is 256 K frames,

FIRST.......256K - 1frames

MIDDLE...(256K / 2) - 1frames

LAST........5frames

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.

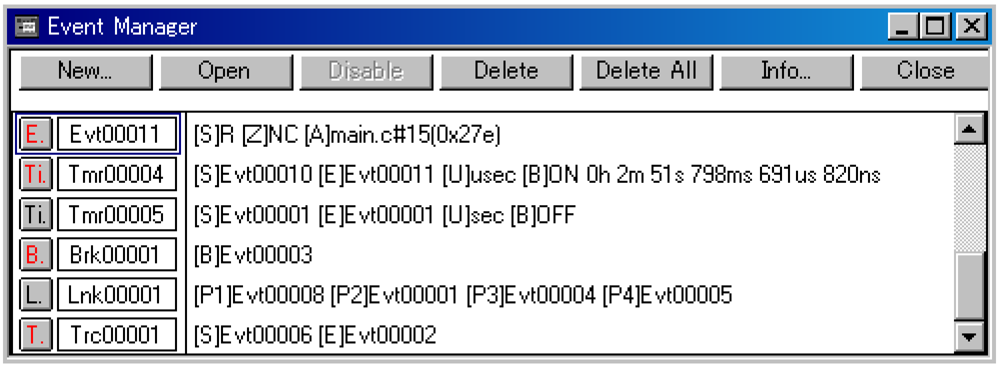
Event Manager

This window is used to manage event conditions. This window allows display, enabling/disabling, and deletion of the Various Event Conditions. (Refer to "5. 11 Event function".)

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

The event icon is the jump pointer of the Jump function.

Figure 6-52 Event Manager (In detailed display mode)



```
E.  Evt00011   [S]R [Z]NC [A]main.c#15(0x27e)
Ti. Tmr00004   [S]Evt00010 [E]Evt00011 [U]usec [B]ON 0h 2m 51s 798ms 691us 820ns
Ti. Tmr00005   [S]Evt00001 [E]Evt00001 [U]sec [B]OFF
B.  Brk00001   [B]Evt00003
L.  Lnk00001   [P1]Evt00008 [P2]Evt00001 [P3]Evt00004 [P4]Evt00005
T.  Trc00001   [S]Evt00006 [E]Evt00002
```

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

Opening

Click the **Mgr** button, or select [Event] menu -> [Event Manager].

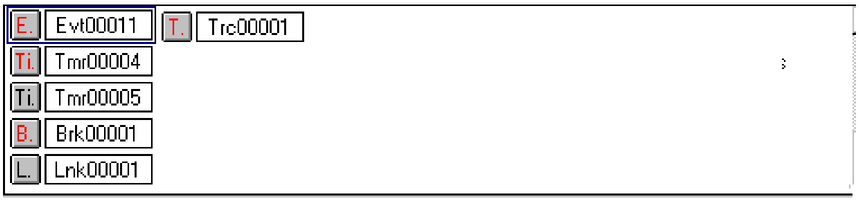
Explanation of each area

(1) Event display area

This area displays the icons (event icons) of the registered Various Event Conditions.

By selecting the context menu -> [Detail], the details can be displayed.

(a) [In list displayed]

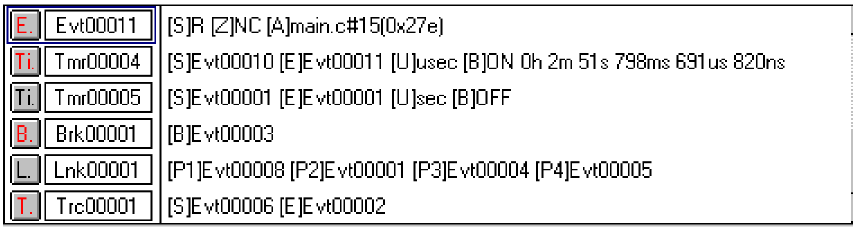


```
E.  Evt00011   T. Trc00001
Ti. Tmr00004
Ti. Tmr00005
B.  Brk00001
L.  Lnk00001
```

Displays event icon (refer to "Table 5-16 Event icon").

The event icon is the jump pointer (refer to "5. 14. 2 Jump function").

(b) In detailed display



E.	Evt00011	[S]R [Z]NC [A]main.c#15(0x27e)
Ti.	Tmr00004	[S]Evt00010 [E]Evt00011 [U]usec [B]ON 0h 2m 51s 798ms 691us 820ns
Ti.	Tmr00005	[S]Evt00001 [E]Evt00001 [U]sec [B]OFF
B.	Brk00001	[B]Evt00003
L.	Lnk00001	[P1]Evt00008 [P2]Evt00001 [P3]Evt00004 [P4]Evt00005
T.	Trc00001	[S]Evt00006 [E]Evt00002

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

Table 6-20 Separator for Displaying Event Details

Key Information	Contents
Event condition	
[S]	Status condition
[Z]	Access size condition
[A]	Address condition Symbol or expression: (actual address)
[D]	Data condition Symbol or expression: (actual address)
[M]	Mask condition
Event link condition	
[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 connected)	
[M]	Trace mode
[D]	Delay Count
[S]	Trace start condition
[E]	Trace end condition
[S1] - [S4]	Section trace start condition
[E1] - [E4]	Section trace end condition
[Q]	Qualify trace condition
Timer condition (When IECUBE Connected)	
[S]	Timer measurement start condition
[E]	Timer measurement end condition

Table 6-20 Separator for Displaying Event Details

Key Information	Contents
[U]	Timer measurement unit
[B]	Timeout break condition

[View]menu (Event manager-dedicated items)

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

Select All Event	Selects all the registered events.
Delete Event	Deletes a selected event.
Sort By Name	Displays icons in the order of event names.
Sort By Kind	Displays icons in the order of event types.
Unsort	Does not sort icons (default).
Detail	Displays the details
Overview	List display (default)

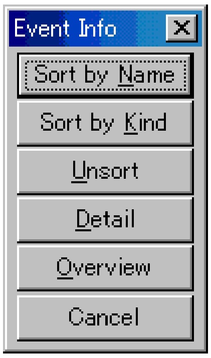
Context menu

Sort By Name	Displays icons in the order of event names.
Sort By Kind	Displays icons in the order of event types.
Unsort	Does not sort icons (default).
Detail	Displays the details.
Overview	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. 14. 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. 14. 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. 14. 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, the Event Link dialog box,the Break dialog box, the Trace dialog box and the Timer dialog box.)

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 of <Event...>,<Event.Link...>,<Break...>,<Trace...> and <Timer...> 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.
Break...	Opens the Break dialog box.
Trace...	Opens the Trace dialog box (when IECUBE connected) .
Timer...	Opens the Timer dialog box (when IECUBE connected) .
Manager	Opens the Event Manager.

Add Event...	Opens theEvent dialog boxin the select mode, and selects or newly creates an event condition to be set. The event condition will be added to the area selected when the < Add Event...> button is pressed.
Add Link...	Opens the Event Link dialog box in the select mode, and selects or newly creates an event link condition. The event condition will be added to the area selected when the < Add Link...> button is pressed.
Open	Opens the various event setting dialog box corresponding to the selected event condition (one). Each setting dialog box displays the contents of the selected event condition. Same operation as double-clicking the event icon or pressing the Enter key.
Remove / Delete	Deletes the selected event. 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 6-53 Select Display Information Dialog Box  Event Info Sort by Name Sort by Kind Unsort Detail Overview Cancel <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 of <Event...>,<Event.Link...>,<Break...>,<Trace...> and <Timer...> 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.

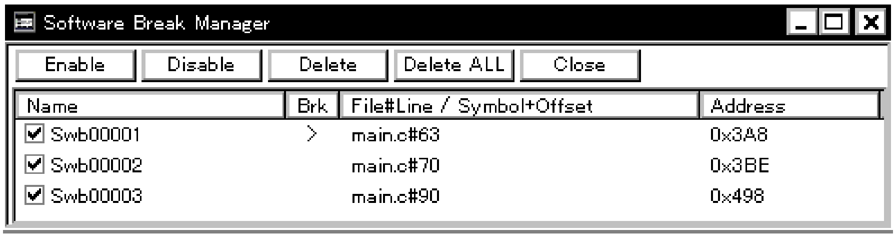
Software Break Manager

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 the Assemble window. (Refer to "5. 4. 2 Breakpoint setting".)

Caution Software breaks can be set or deleted while the user program is being executed. While the user program is being executed, software breaks can be set ,deleted, enable or disable. The warning of a purport which makes a user program once take a break is displayed.

Figure 6-54 Software Break Manager



Name	Brk	File#Line / Symbol+Offset	Address
☑ Swb00001	>	main.c#63	0x3A8
☑ Swb00002		main.c#70	0x3BE
☑ Swb00003		main.c#90	0x498

- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

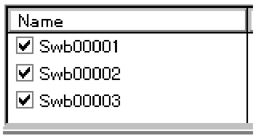
(1) Name

(2) Brk

(3) File#Line / Symbol+Offset

(4) Address

(1) Name



Name
☑ Swb00001
☑ Swb00002
☑ Swb00003

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

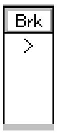
An event name is displayed in the form of Swb+[number] in the default condition. It can be changed to an alphanumeric string of up to 256 characters. To change an event name, select and click a name. Then directly edit the name. To set the editing, press the Enter key.

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

Furthermore, the name jumps to the Source window by double-clicking an event name if the event name corresponds to the source line, whereas the name jumps to the Assemble window if it does not correspond to the source line.

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

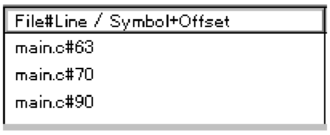
(2) Brk



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



File#Line / Symbol+Offset
main.c#63
main.c#70
main.c#90

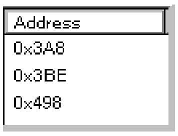
This area displays the location at which a software break event was set as follows:

- 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.

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

(4) Address



Address
0x3A8
0x3BE
0x498

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

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

Function buttons

Enable	Enables the selected event.
Disable	Disables the selected event.
Delete	Deletes the selected event.
Delete ALL	Deletes all the set software break events.
Close	Closes this window.

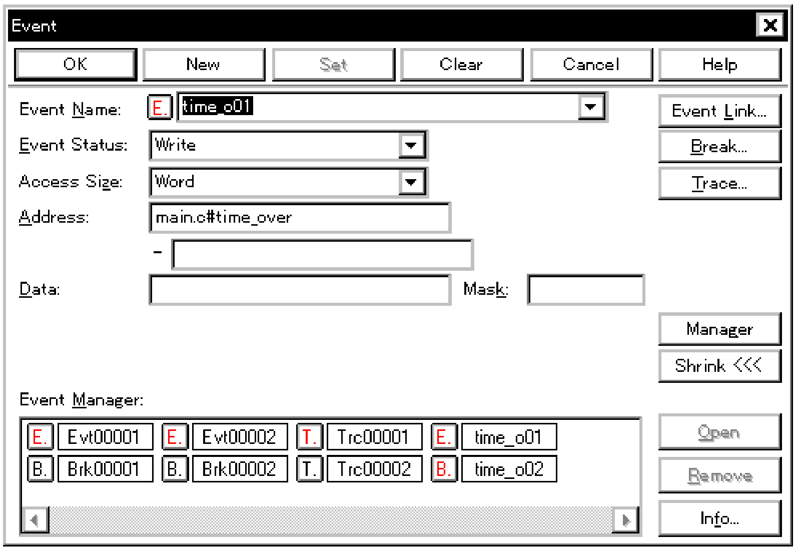
Event dialog box

This dialog box is used to register and display event conditions. (Refer to "5. 11 Event function".)

Setting of event conditions is done by setting each item in this dialog box and then pressing 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. 11. 4 Number of enabled events for each event condition").

Figure 6-55 Event Dialog Box



Event Name: E. time_o01

Event Status: Write

Access Size: Word

Address: main.c#time_over

-

Data:

Mask:

Event Manager:

E. Evt00001	E. Evt00002	T. Trc00001	E. time_o01
B. Brk00001	B. Brk00002	T. Trc00002	B. time_o02

- 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] menu -> [Event...].

In select mode

If the <OK> button is pressed when the Event dialog box has been opened as follows, an event condition can be registered in the setting dialog box from which this dialog box was opened (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

Event Name: E. time_o01

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 5-16 Event icon"). The gray E. mark indicates that the event condition is being edited and has not been registered yet.

(2) Event Status

Event Status: Write

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 6-21 Status condition

Status	Abbreviation	Meaning
Execution event		
Execution	EX	Program execution
Before Execution	EX-B	Program execution (break before execution) Note
Access event		
R/W	RW	Data read/write
Read	R	Data read
Write	W	Data write

Note Multiple items can be specified, but only two items, including access events, can be enabled. The address range cannot be specified. Can be used only for break event conditions.

(3) Access Size

Access Size: Word

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).
Half Word	Detects data condition with 16-bit width (only during 16-bit access).
Word	Detects data condition with 32-bit width (only during 32-bit access).
No Condition	Does not detect access size (nothing can be input to the Data area).
Bit	Detects data condition with 1-bit width (only during 8-bit access). 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 ID850QB detects a dummy bit access by internally setting address conditions and data conditions as follows: **Input example:** Address: FE20.1 Data: 1 ↓ **Setting of trace search** Address: FE20 Data: 00000010B Mask: 11111101B If another bit of the same address is accessed or if all the 8 bits of the same address are accessed, therefore, an event is detected in accordance with the specified status if the address and bit match the specified value of [address.bit].

Caution 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.

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
- Half Word if the data condition is set in 16-bit units
- Word if the address condition is set in 32-bit units
- No Condition if no data condition is specified

(4) Address

Address: main.c#time_over

–

This area is used to specify an address condition (address value, mask value).

The default radix for inputting a numeric value is hexadecimal. A data can be also specified by a symbol (refer to

"Table 5-6 Specifying Symbols").

Table 6-22 Settable Range of Address Condition (Event)

Settable range	Remark
0 <= address value <= 0xFFFFFFFF	None

(a) Address

Set an address condition (lower address - higher address) (may be omitted).

The following can be set:

Caution Specify a 28-bit address, since physical address and image space are distinguished in setting event.

(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

(iii) Set a value to the lower address and the higher address. Setting as a bit

Set a value to only the lower address, or set the same value to the lower address and the higher address.

Specify a value in the form of address.bit.

The value of bit, which indicates the bit position, must be 0 >= bit >= 7.

(5) Data, Mask

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 (3) Access Size.

Table 6-23 Settable Range of Data Condition

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

(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.

(6) Event Manager

This area is used to display the list of the events registered. (Refer to "Table 5-16 Event icon", "(4) Manipulation in event manager area".)

Function Buttons

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

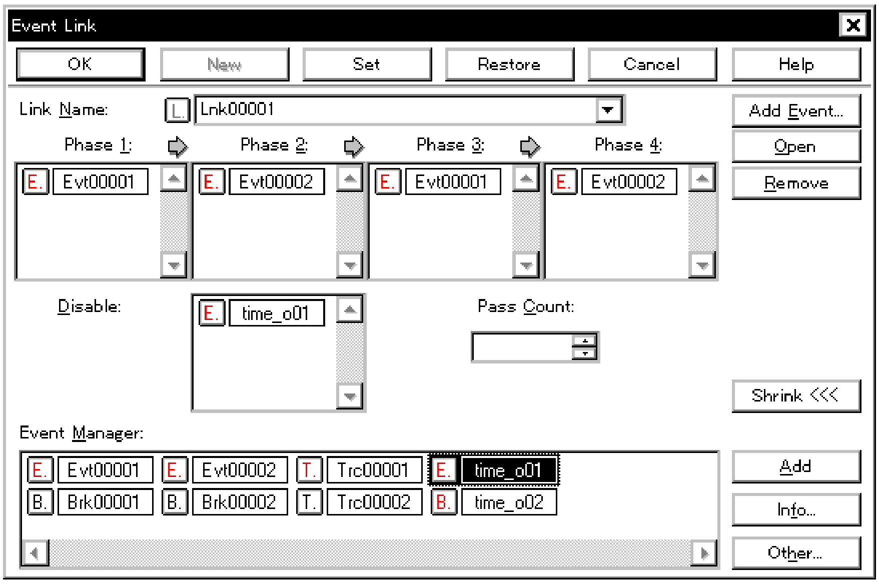
Event Link dialog box

This dialog box is used to register and display event link conditions. (Refer to "5. 11 Event function".)

Registration of event link conditions is done by setting each item (256 items max.) in this dialog box and then pressing 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. 11. 4 Number of enabled events for each event condition").

Figure 6-56 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

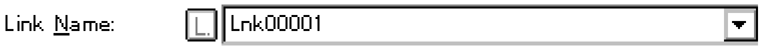
If the <OK> button is pressed when the Event Link dialog box has been opened as follows, an event link condition can be registered in the setting dialog box from which this dialog box was opened.

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



Link Name: L. Lnk00001

This area is used to set a event link name.

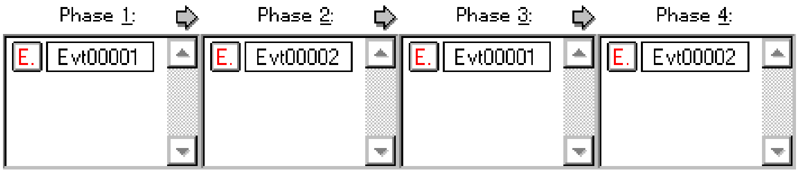
Directly input an alphanumeric string of up to eight characters as a name.

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 5-16 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



Phase 1: E. Evt00001

Phase 2: E. Evt00002

Phase 3: E. Evt00001

Phase 4: E. Evt00002

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 6-24 The Number of Event Conditions in Event Link Dialog Box

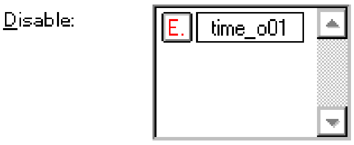
Connected IE		Each phase	Disable area	Total (execution/ access)
IECUBE		10	10	14 (8/6)
N-Wire CARD	Nx85ET （RCU0+TEU+TRCU）	10[a]	12(max)	12(max)
	Nx85E901（RCU0）, RCU1	1[b]	-	2[c]

a. Can be set in Phase1-4 (expect before execution)

b. Can be set only in Phase 1 and Phase 2 (must always be set in two stages)

c. Only events before execution

(3) Disable



This area is used to set an event condition that invalidates the event conditions that have so far been satisfied ("Table 6-24 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. 11. 3 Setting event conditions".

(4) Pass Count

This area is used to set a pass count condition (settable range: 1 to 32767).

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 5-16 Event icon", "(4) Manipulation in event manager area".)

Function Buttons

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

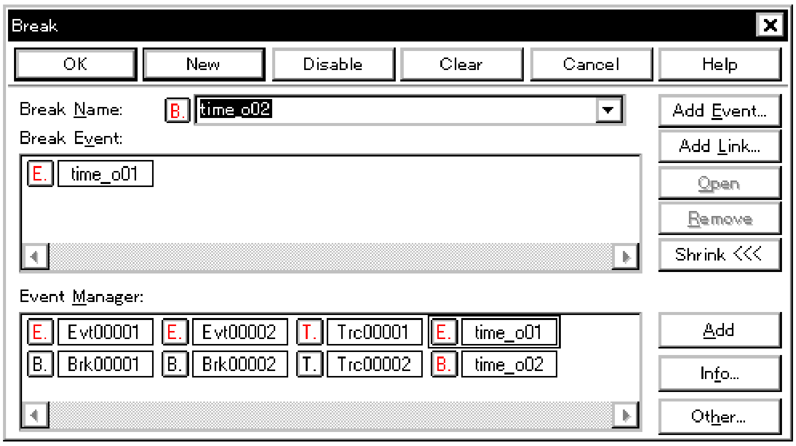
Break dialog box

This dialog box is used to register; set, and display break event conditions. (Refer to "5. 11 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 pressing 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. 11. 4 Number of enabled events for each event condition".)

Figure 6-57 Break Dialog Box



Break Name: B. time_o02

Break Event: E. time_o01

Event Manager: E. Evt00001, E. Evt00002, T. Trc00001, E. time_o01, B. Brk00001, B. Brk00002, T. Trc00002, B. time_o02

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

Opening

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

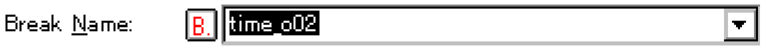
Explanation of each area

(1) Break Name

(2) Break Event

(3) Event Manager

(1) Break Name



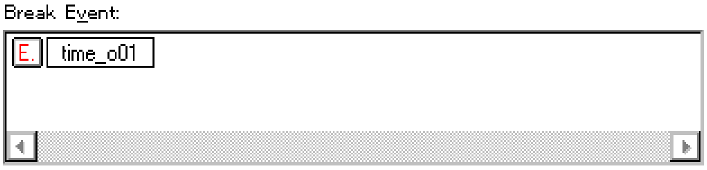
Break Name: B. time_o02

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 5-16 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



Break Event:

E. time_o01

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

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

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 section "5. 11. 3 Setting event conditions".

Table 6-25 Number of Events Settable in Break Condition Setting Area

Connected IE		Total (before execution / after execution / access)
IECUBE		16 (2 / 8 / 6)
N-Wire CARD	Nx85ET （RCU0+TEU+TRCU）	14 (2 / 8 / 4)
	Nx85E901（RCU0），RCU1	2

(3) Event Manager

This area is used to display the list of the events registered. (Refer to "Table 5-16 Event icon", "(4) Manipulation in event manager area".)

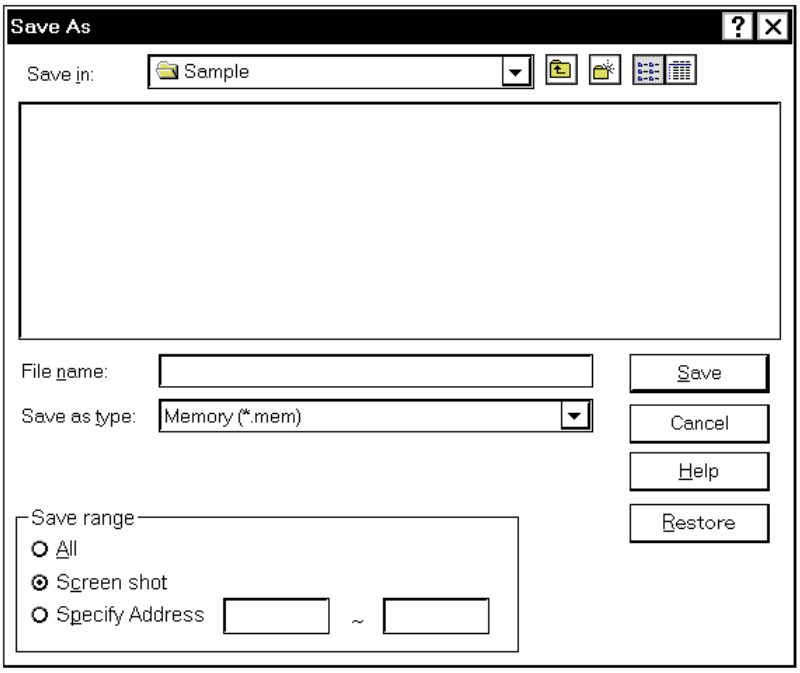
Function Buttons

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 display file. (Refer to "5. 13. 2 Window display information (view file)".)

Figure 6-58 View File Save Dialog Box



- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

(1) Save in :, File name :

(2) Save as type :

(3) Save range

(1) Save in :, File name :

This area is used to specify the file name to be saved. A file name can be directly input from the keyboard, or selected from the list.

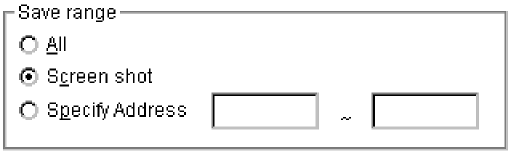
Up to 257 character string with a extension can be specified.

(2) Save as type :

This area is used to specify the type (extension) of the file to be saved. (Refer to "Table 5-18 Type of the View Files".)

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

(3) Save range



Save range

○ All

● Screen shot

○ Specify Address ~

Specify the range of data to be saved.

This area is displayed if the current window to be saved is the Assemble window, Memory window, Source window, Trace 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 6-18 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 tne 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. 13. 2 Window display information (view file)".)

When a display 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.

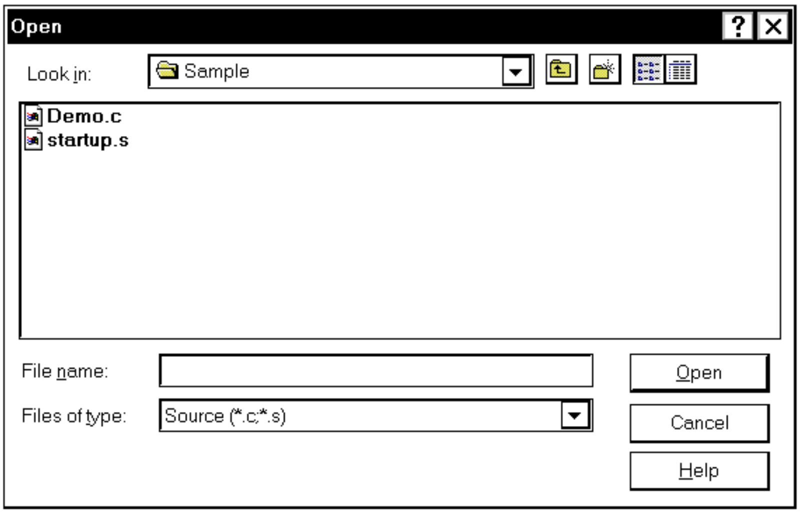
- **Loading source file to which symbol information has been read**

  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 6-59 View File Load Dialog Box



- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

(1) Look In:, File name :

(2) Files of type :

(1) Look In:, File name :

This area is used to specify the file name to be loaded. A file name can be directly input from the keyboard, or selected from the list.

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

(2) Files of type :

This area is used to specify the type (extension) of the file to be loaded. (Refer to "Table 5-18 Type of the View Files".)

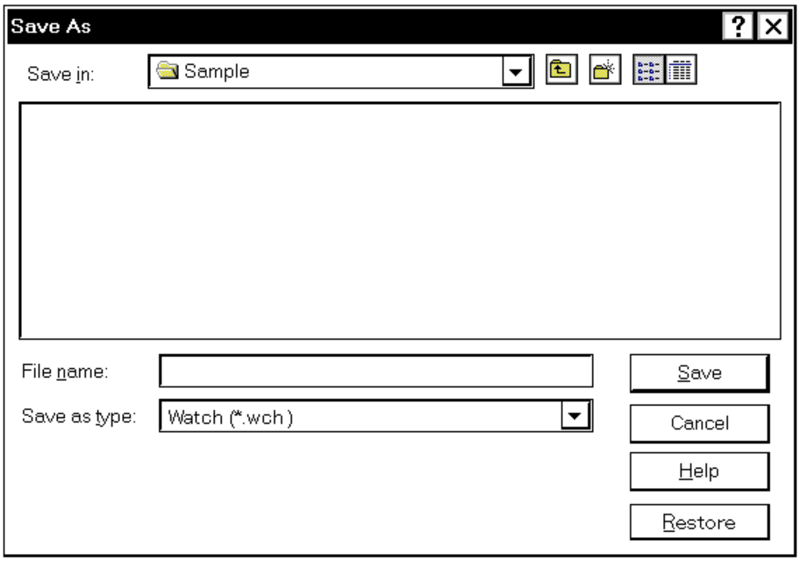
Function buttons

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. 13. 3 Window setting information (setting file)".)

Figure 6-60 Environment Setting File Save Dialog Box



- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

(1) Save in :, File name :

(2) Save as type :

(1) Save in :, File name :

This area is used to specify the file name to be saved. A file name can be directly input from the keyboard, or selected from the list.

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

(2) Save as type :

This area is used to specify the type (extension) of the file to be saved. (Refer to "Table 5-19 Type of the Setting Files").

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

Function buttons

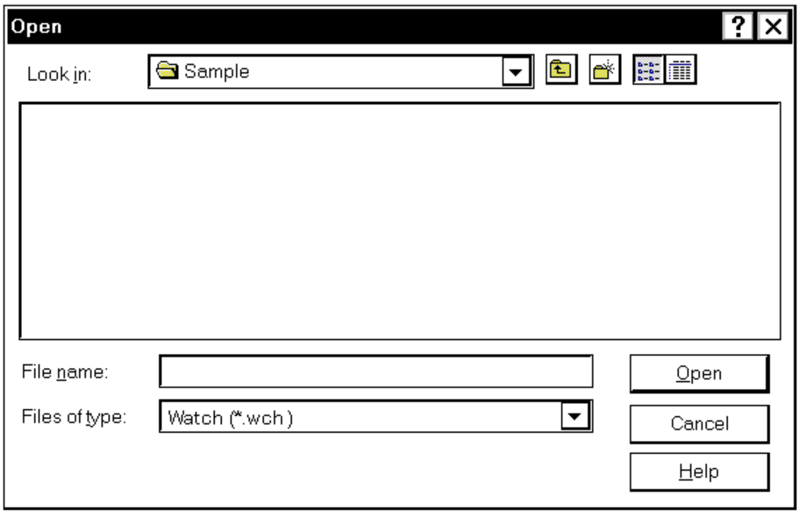
Save	Saves the setting information of the current window to the selected file. After saving, this dialog box is closed.
Cancel	Closes this dialog box without executing anything.
Help	Displays this dialog box online help files.
Restore	Restores the status before this dialog box was opened.

Environment Setting File Load dialog box

This dialog box is used to read the setting files. (Refer to "5. 13. 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 6-61 Environment Setting File Load Dialog Box



Look in: Sample

File name:

Files of type: Watch (*.wch)

- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

(1) Look In:, File name :

(2) Files of type :

(1) Look In:, File name :

This area is used to specify the file name to be loaded. A file name can be directly input from the keyboard, or selected from the list.

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

(2) Files of type :

This area is used to specify the type (extension) of the file to be loaded. (Refer to "Table 5-19 Type of the Setting Files".)

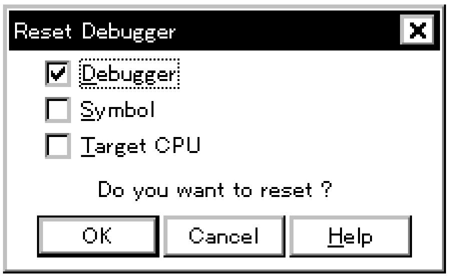
Function buttons

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 ID850QB, CPU, and symbol information.

Figure 6-62 Reset Debugger Dialog Box



☑ Debugger
□ Symbol
□ Target CPU

Do you want to reset ?

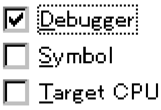
- Opening
- Explanation of each area
- Function buttons

Opening

Select [File] menu -> [Debugger Reset...].

Explanation of each area

(1) Reset subject selection area



☑ Debugger
□ Symbol
□ Target CPU

This area is used to specify what is to be Initialized. Initializes the checked item.

Debugger	Initializes the ID850QB (default).
Symbol	Initializes the symbol information.
Target CPU	Initializes the CPU.

Function buttons

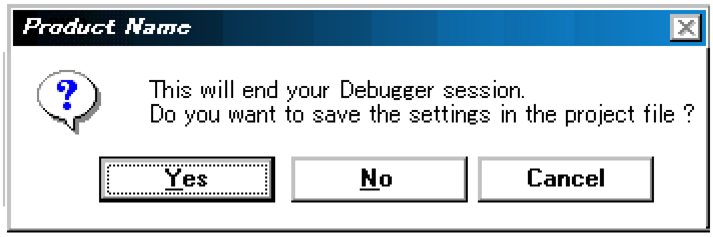
OK	Initializes according to the setting.
Cancel	Cancels the changes and closes this dialog box.
Help	Displays this dialog box online help files.

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 ID850QB. (Refer to "5. 13. 1 Debugging environment (project file)".)

It can be specified in the Debugger Option dialog box that the ID850QB is terminated without this confirmation dialog box being opened.

Figure 6-63 Exit Debugger Dialog Box



Product Name

This will end your Debugger session.
Do you want to save the settings in the project file ?

- 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 ID850QB. 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 ID850QB 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 ID850QB. (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

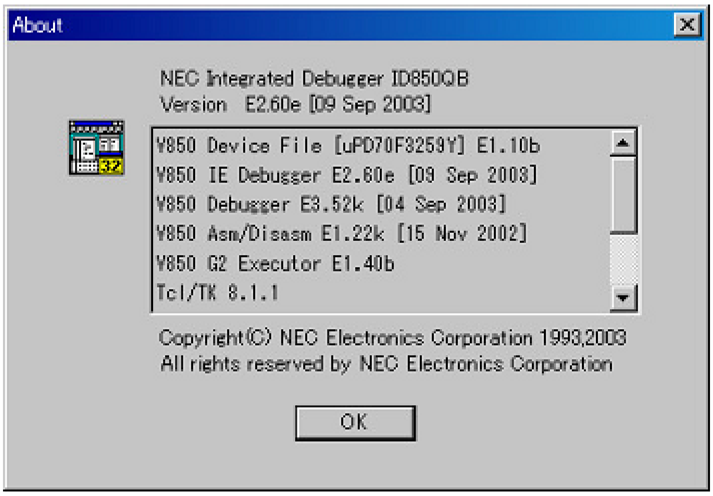
This dialog box displays the version information of the ID850QB (the year is displayed in 4 digits).

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 ID850QB
- Version of device file
- Version of GUI
- Version of debugger DLL
- Version of assembler DLL
- Version of executor
- Version of Tcl/Tk
- Product ID and product version of in-circuit emulator

Figure 6-64 About Dialog Box



NEC Integrated Debugger ID850QB
Version E2.60e [09 Sep 2003]

```
V850 Device File [uPD70F3259Y] E1.10b
V850 IE Debugger E2.60e [09 Sep 2003]
V850 Debugger E3.52k [04 Sep 2003]
V850 Asm/Disasm E1.22k [15 Nov 2002]
V850 G2 Executor E1.40b
Tcl/TK 8.1.1
```

Copyright(C) NEC Electronics Corporation 1993,2003
All rights reserved by NEC Electronics Corporation

- 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

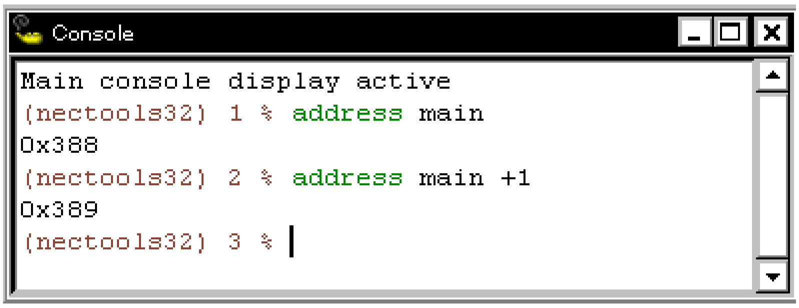
This window is used to input commands that control the ID850QB .

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 6-65 Console Window



```
Main console display active
(nectools32) 1 % address main
0x388
(nectools32) 2 % address main +1
0x389
(nectools32) 3 %
```

- Opening

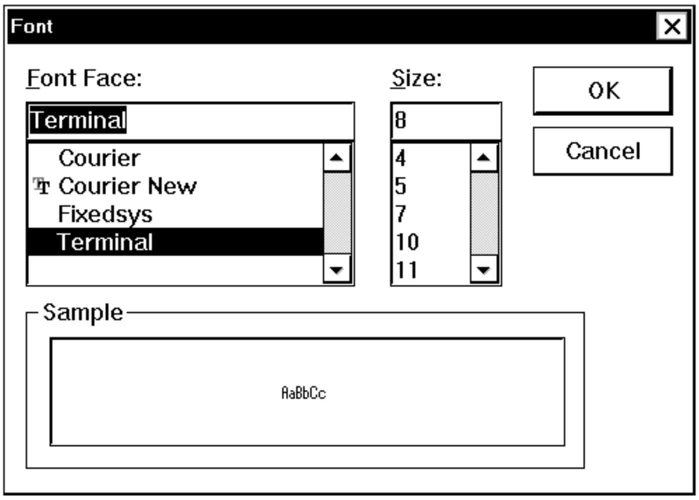
Opening

Select [Browse] menu -> [Console].

Font dialog box

This dialog box is used to select the font and font size to be displayed in the Source window, the Watch window, the Quick Watch dialog box, the Local Variable window, the Stack Trace window.

Figure 6-66 Font Dialog Box



Font Face: Terminal

Size: 8

- Opening
- Explanation of each area
- Function buttons

Opening

Click the <Font> button in the Debugger Option dialog box.

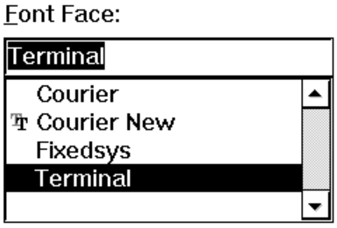
Explanation of each area

(1) Font Face :

(2) Size :

(3) Sample

(1) Font Face :

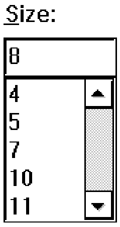


Font Face: Terminal

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

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

(2) Size :

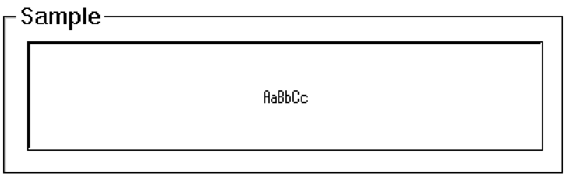


Size: 8

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

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

(3) Sample



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

Function buttons

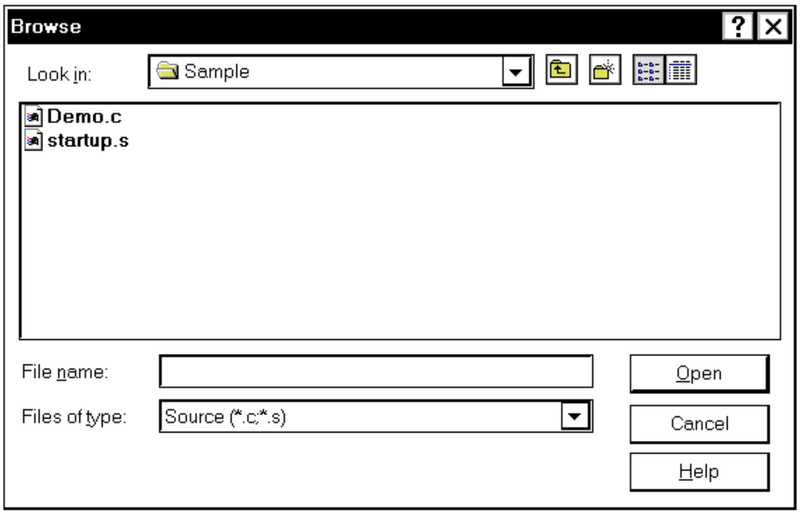
OK	Validates the settings and closes this dialog box.
Cancel	Cancels the changes and closes this dialog box.

Browse dialog box

This dialog box is used to select the file to be set in the Source Text move dialog box.

Remark If this dialog box is opened for the first time after the system has been started up, the directory first specified by the source path is displayed. When the dialog box is opened the second and subsequent times, the previously displayed directory is recorded and displayed again. If the <Cancel> button is pressed, however, the previously displayed directory is not recorded.

Figure 6-67 Browse Dialog Box



Browse

Look in: Sample

Demo.c
startup.s

File name:

Files of type: Source (*.c;*.s)

- Opening
- Explanation of each area
- Function buttons

Opening

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

Explanation of each area

(1) Look In:, File name :

(2) Files of type :

(1) Look In:, File name :

This area is used to specify the file name to be opened. A file name can be directly input from the keyboard, or selected from the list.

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

(2) Files of type :

This area is used to specify the type (extension) of the file to be opened (refer to "Table 5-5 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 ID850QB.

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 ID850QBList of Debugger Control Commands

Command name	Function
address	Evaluation of address expression Converts the address expression specified with expression into address.
assemble	Disassemble/line assemble (a) Assembles the character strings specified by code from the address specified by address.
batch	Executing batch (with echo) Executes in batch with displaying files specified by script name on the screen.
breakpoint	Setting/deletion of breakpoint (b) Operates the breakpoint specified by options and address.
cache	Setting of cache Sets I cache and D cache.
dbgexit	Terminating ID850QB Terminate the ID850QB.
dbgopt	Selecting debugger option Selects the debugger option.
download	Download of files (l) Downloads files specified with filename according to options.
erase	Deletion of the internal flash memory Deletes the internal flash memory
extwin	Creation of expansion window Creates expansion window with script file.
finish	Returning from function Executes until it returns to the program that called the current function.
flop	Manipulation related to internal flash memory Selects whether the clock-related settings during internal flash memory write are the device file settings or user settings.
go	Continuous execution (g) Executes program continuously.
help	Display of help Displays Dcl help.
hook	Setting of hook Sets the procedure for hook with script file.
ie	Display/setting of IE register The ie command depends on the IE.
inspect	Symbol inspect Searches and displays the symbol of the load module specified with progname using the regular expression of *pattern*.
jump	Jump to window (j) Displays the window specified by options.
map	Setting / deleting memory mapping Sets, deletes, and displays memory mapping.

Table 7-1   ID850QBList of Debugger Control Commands

Command name	Function
mdi	Setting of expansion window Sets the size and title name of the expansion window.
memory	Display/setting of memory (m) Sets value in the memory of the address specified with address according to options.
module	Display of the list of files and functions Displays the list of files and functions of the load module specified by progname.
next	Procedure step (n) Executes the procedure steps.
refresh	Redrawing of window Redraws the window and updates the data.
register	Display/setting of register value and I/O register value Sets value in the register specified with regname.
reset	Reset Resets the CPU, ID850QB debugger or symbols.
run	Reset and execution of CPU (r) Resets the program and executes it.
step	Step execution (s) Executes step execution.
stop	Stop execution Stops the program forcibly.
upload	Upload Saves the memory data within the specified range in a file.
version	Display of the version information Displays the version of the ID850QB.
watch	Display/setting of variables (w) Displays and sets the variables.
where	Trace of stack Executes the back-trace of the stack.
wish	Start of Tclet Starts up the script using Tk (Tclet).
xtime	Operation of timer Operates timer.
xtrace	Operation of tracer(when IECUBEconnected) Operates tracer.

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
Other	Based on Tcl/Tk 8.4

7. 3 List of Variables

- 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. 4 List of Packages

- tcltest - Restoration test
- cwind - Automatic window control
- BWidget - Toolkit
- Tcllib - Tcl library
- mclistbox - Multi-column list box
- combobox - Combo box

7. 5 Key Bind

- tcsh + emacs like
- Complement of command name [Tab]
- Complement of file name [Tab]
- HTML help [F1]

7. 6 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 expansion windows, Tk menu commands cannot be used because of the restrictions of MDI windows.

7. 7 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 7-3 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 ID850QB
17	After changing extended option
18	After changing debugger option
19	After downloading
20	After changing memory or register
36	Before starting tracer(when IECUBEconnected)
37	After stopping tracer (when IECUBEconnected)
40	Before starting timer (when IECUBEconnected)
41	After stopping timer (when IECUBEconnected)
42	After clearing trace (when IECUBE connected)
45	After resetting symbol
46	After change from RRM function to trace function, and change from trace function to RRM function.

7. 8 Hook Procedure

A hook can be set in the ID850QB using the hook procedure.

The hook procedures are shown below:

- BeforeDownload (Hook before downloading)
- AfterDownload (Hook after downloading)
- AfterCpuReset(Hook after CPU reset)
- 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 ID850QB is closed.

[When hook is set with ID850QB control command]

1) Create script file a. with an editor.
2) Start up the ID850QB, 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

[When hook is set on downloading of project file]

1) Create script file a. with an editor.

   Caution Be sure that the script file name is the same as the project file.

   Example:

   The script file corresponding to test.prj is test.tcl.

   Allocate test.prj, test.pri, and test.tcl in the same directory.

2) Start up the ID850QBand 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
}
```

7. 9 Related Files

- Executes when the aliases.tcl console is opened.
  Sets the default alias etc.
- Executes when the project file name.tcl project is opened.
  The hook of BeforeDownload, AfterDownload, AfterCpuReset, BeforeCpuRun and AfterCpuStop can be used.
- Executes when the load module name.tcl load module is downloaded.
  The hook of BeforeDownload, AfterDownload, AfterCpuReset, BeforeCpuRun and AfterCpuStop can be used.

7. 10 Cautions

- The separator for file and path is a slash (/).
- When a console or an expansion window is active, the accelerator key cannot be acknowledged.
- 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. 11 Explanation of Commands

In this section, each command is explained using the format shown below.

Command name

Describes the command name.

Input format

Describes the input format of the command.

In the following explanation, italics indicate an Argument to be supplied by the user, while the argument enclosed in "?" may be omitted.

When a command name and an option are entered to the point of identifiability, they are recognized.

Functions

Explains the functions of the command.

Usage example

Shows an example of the usage of the command.

address

address - Evaluation of address expression

Input format

address *expression*

Functions

Converts the address expression specified by *expression* into address.

Usage example

```
(IDCON) 1 % address main
0xaa
(IDCON) 2 % address main+1
0xab
```

assemble

assemble - Disassemble/line assemble

Input format

assemble *?options? address ?code?*

Functions

Assembles the character strings specified by *code*from the *address* specified by address.

When '.' is specified for address, it is understood as an *address* continuing from the immediately previous assemble.

When *code* is omitted, it is assembled from the *address* specified by address.

The following are *options*: They are ignored for assembly.

-code	Command code is also displayed.It is ignored for assembly.
-number *number*	*number* line is displayed. It is ignored for assembly.

Usage example

```
(IDCON) 1 % assemble -n 5 main
0x000000aa  B7          PUSH  HL
0x000000ab  B1          PUSH  AX
0x000000ac  891C        MOVW  AX,SP
0x000000ae  D6          MOVW  HL,AX
0x000000af  A100        MOV   A,#0H
(IDCON) 2 % assemble main mov a,b
(IDCON) 3 % assemble . mov a,b
```

batch

batch - Executing batch (with echo)

Input format

batch *scriptname*

Functions

Executes in batch with displaying files specified by *scriptname* on the screen.

Nesting is possible.

Usage example

```
(IDCON) 1 % clear
(IDCON) 2 % batch bat_file.tcl
(IDCON) 3 % tkcon save a:/log.txt
```

breakpoint

breakpoint - Setting/deletion of breakpoint

Input format

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

breakpoint -*delete brkno*

breakpoint -*delete 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, 16, or 32bit).
-data *value*	The data condition is set.
-datamask *value*	The data mask is set.
-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
```

cache

When N-Wire CARD connected

cache - Setting of cache

Input format

cache

cache *config ?-icache itype? ?-dcache dtype?*

cache *clear ?-icache bool? ?-dcache bool?*

Functions

When *config* is specified for the subcommand, the cache type is set.

When *clear* is specified for the subcommand, whether cache clear is to be done by EXEC, or not, is set (default is clear)

When subcommand or below is omitted, the current status is displayed.

itype is selected from the following:

NB85E212	NB85E212 is used.
NB85E213	NB85E213 is used.
nouse	Not used (default).

dtype is selected from the following:

NB85E252	NB85E252 is used.
NB85E263	NB85E263 is used.
nouse	Not used (default).

bool is selected from the following:

0, false, or off	Off
11, true, or on	On

Usage example

```
(IDCON) 1 % cache config -i NB85E212 -d NB85E252
(IDCON) 2 % cache
i-cache: NB85E212
i-cache: NB85E252
(IDCON) 3 % cache clear -i false
(IDCON) 4 % cache
i-cache: NB85E212 (persist)
d-cache: nouse
```

dbgexit

Tdbgexit - Terminating ID850QB

Input format

dbgexit *?options?*

Functions

Terminate the ID850QB.

The following are *options*:

-saveprj	Project is saved on terminating ID850QB.

Usage example

```
(IDCON) 1 % dbgexit -saveprj
```

dbgopt

dbgopt - Selecting the debugger option

Input format

dbgopt *options ?value?*

Functions

Selects the option of the debugger.

The following are *options*:

-function *?func?*	Switches the RRM function and trace function (when IECUBE connected). When *func* is omitted, the current function is displayed.

func is selected from the following:

rrm	RRM function is selected.
trace	Trace function is selected.

Usage example

```
(IDCON) 1 % dbgopt ?function trace
```

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.
-append	Additional download is executed.
-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) (when N-Wire CARD connected).
-reset	CPU is reset after download.
-information	Download information is displayed.

Usage example

```
(IDCON) 1 % download test.lmf
```

erase

When N-Wire CARD connected

erase - Deletion of the internal flash memory

Input format

extwin *scriptfile*

Functions

Deletes 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
```

flop

When N-Wire CARD connected

flop - Manipulation related to internal flash memory

Input format

```
flop -init
flop -user clock
```

Functions

Selects whether the clock-related settings during internal flash memory write are the device file settings or user settings.

When the -init option is specified, processing is performed using the device file setting.

If the device file contains no information, an error occurs during internal flash memory write.

In this case, use the -user option.

When the -user option is specified, IOR of the clock generation function is manipulated.

Specify for *clock* the frequency (MHz) of the CPU clock (fCPU) determined by the IOR manipulation.

In this case, perform the IOR setting using the hook procedure.

Usage example

```
(IDCON) 1 % flop -user 13.5
```

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 ID850QB is reset.

Usage example

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

ie

ie - Display/setting of IE register

Input format

ie reg *address ?value?*

ie dcu *address ?value?*

Functions

The **ie** command depends on the IE.

When reg is specified for the subcommand, referencing and setting of the IE register is executed.

When dcu is specified for the subcommand, referencing and setting of the DCU register is executed.

Caution The value of a register will be reset by 0 if a DCU register is referred to.

Usage example

```
(IDCON) 1 % ie reg 0x100 1
(IDCON) 2 % ie dcu 0x100 1
```

inspect

inspect - Symbol inspect

Input format

inspect *?options? progname pattern*

Functions

Searches and displays the load module symbol specified with *progname* using the regular expression of *pattern*.

The following regular expressions can be used.

?	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	Does not distinguish between upper and lowercase.
-address	Displays in pair with symbol address.

Usage example

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

jump

jump - Jump to window

Input format

jump -source -line *filename ?line?*

jump *?options? address*

Functions

Displays the window specified by *options*.

-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.
-protect	I/O protect area is mapped.
-rrm	Start address of real-time RAM monitor area is set. If performed during user program execution, CPU is stopped for an instant. Real-time RAM monitor area can be divided into 8 partitions. The start address and size are specified in pairs in list format as follows. {{address size} {address size} {address size} ...} size is one of 256, 512, 768, 1024, 1280, 1536, 1792, 2048 bytes, and the total size is up to 2048.
-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 {Saddr}
7: 0xfee0 0xfeff 8 {Register}
8: 0xff00 0xffff 8 {SFR}
(IDCON) 2 % map -erom 0x100000 0x10ffff
(IDCON) 3 % map -c
```

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).
-halfword	Displayed/set in halfword units.
-word	Displayed/set in word units.
-fill	The data is filled in.
-copy	The data is copied.
-noverify	Verification is not executed on writing.

If memory referencing is performed for other than the real-time RAM monitor area during user program execution, the CPU is stopped for an instant. The CPU is stopped for an instant even if the memory setting is performed.

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 I/O register 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 is executed.

If register reference/setting is performed during user program execution, the CPU stops for an instant.

Usage example

```
(IDCON) 1 % register pc
0x0100
(IDCON) 2 % register pc 200
(IDCON) 3 % register pc
0x200
```

reset

reset - Reset

Input format

reset *?options?*

Functions

Resets the ID850QB , CPU or symbols.

If options are omitted, the CPU is reset.

The following are *options*:

-cpu	CPU is reset (default).
-debugger	The ID850QB is reset.
-symbol	Symbol is reset.
-event	Event is reset.

Usage example

```
(IDCON) 1 % reset
```

run

run - Reset and execution of CPU

Input format

run *?options?*

Functions

Resets the program and executes it.

If -waitbreak is not specified, the command does not wait until the program stops.

The following are *options*:

-**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 command units.

Usage example

```
(IDCON) 1 % step -i
(IDCON) 2 % step -s
```

stop

stop - Stop executing

Input format

```
stop
```

Functions

Stops the program forcibly.

Usage example

```
(IDCON) 1 % run
(IDCON) 2 % stop
```

upload

upload - Upload

Input format

upload *?options? filename address1 address2*

Functions

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

The following are *options*:

-binary	The data is saved in binary format.
-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 ID850QB.

Usage example

```
(IDCON) 1 % version
GUI              : E2.00y [31-May-99]
Devicefile       : 78K0[uPD780034] E1.01a
Debugger         : 78K/0 Debugger E2.50c [02-Apr-99]
Executer          : 78K/0 Executer E1.3c
Packet translator : 78K/0 Packet E2.00w
Assembler        : 78K/0 Asm/Disasm E1.15a [01-Apr-99]
Tcl/Tk            : 8.1.1
```

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 - Trace of stack

Input format

where

Functions

Executes the back-trace of the stack.

Usage example

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

wish

wish - Startup of Tclet

Input format

wish *scriptname*

Functions

Starts up the script using Tk (Tclet).

The expansion window can be created with Tclet.

Usage example

```
(IDCON) 1 % wish test.tcl
```

xtime

xtime - Operation of timer

Input format

xtime *option*

Functions

Operates timer.

The following are ***options***:

-start	Timer starts on executing the program. (when IECUBE connected)
-stop	Timer stops on executing the program. (when IECUBE connected)
-gobreak	Time from Go to Break is displayed in nsec.

Usage example

```
(IDCON) 1 % xtime -start
(IDCON) 2 % xtime -stop
```

xtrace

When IECUBE connected

xtrace - Operation of tracer

Input format

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

xtrace -start

xtrace -stop

xtrace -clear

xtrace -*addup ?bool*?

xtrace -mode *?mode?*

xtrace -*complement ?bool*?

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.
-addup *?bool?*	Whether the time tag is totaled or not is selected. (Addition only) When *bool* is omitted, the current mode is displayed.
-mode *?mode?*	The trace control mode (any one of: all, cond, nonstop, fullstop, fullbreak, delaystop, delaybreak, machine, or event) is selected. When *mode* is omitted, the current mode is displayed.
-complement *?bool?*	Selects whether to perform trace complementation. When *bool* is omitted, the current mode is displayed.

Usage example

```
(IDCON) 1 % xtrace -start
(IDCON) 2 % xtrace -stop
(IDCON) 3 % xtrace -dump 3
_ 01685 2 000000BC M1 br _sub2+0x2
_ 01686 4 0000009A BRM1 st.w r6, 0x8[sp]
_ 01687 3 0000009E BRM1 st.w r0, 0x0[sp]
(IDCON) 4 % xtrace -clear
(IDCON) 5 % xtrace -addup true
```

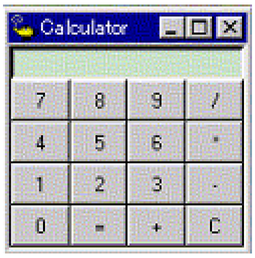
7. 12 Samples (Calculator Script)

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 $top.f -fill both -expand 1set i 0; set v {}; set r
0.25
foreach n {7 8 9 / 4 5 6 * 1 2 3 - 0 = + C} {
        if {$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



APPENDIX A EXPANSION WINDOW

A. 1 Overview

With the ID850QB, 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 ID850QB. Users can create windows using this Tcl.

The ID850QB 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.
RRM window	This is the memory window for real-time RAM monitoring.
Hook window	Sets the hook procedure.
Memory Mapped I/O window	Writes to or reads from the specified address.

A. 3 Activation

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

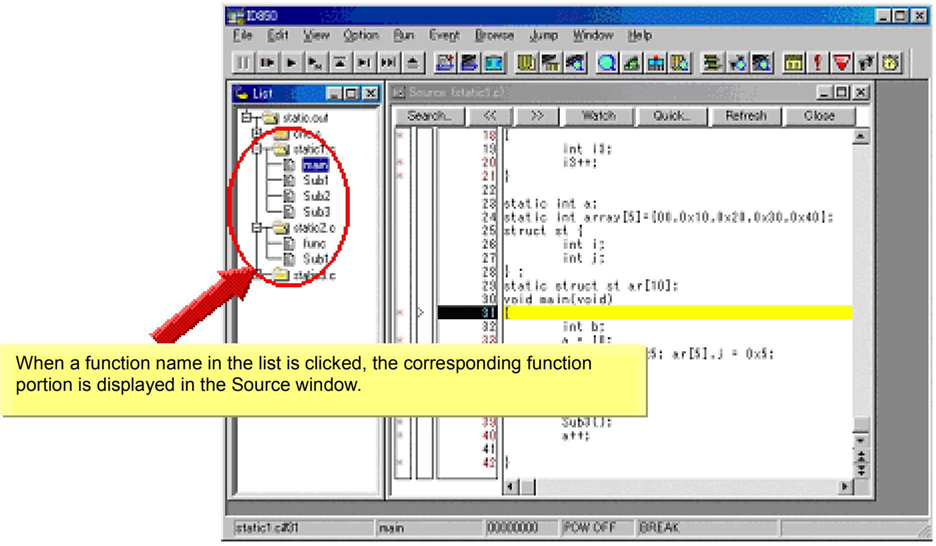
Remark Each .tcl file is installed in NECTools32\BIN\idtcl\tools.

A. 4 Explanation of each sample window

List window

The lists of the source files and functions are displayed in a tree format in this window. When a function name in the list is clicked, the corresponding source is displayed.

Figure A-1   List window



```
{
        int i3;
        i3++;
}

static int a;
static int array[5]={00,0x10,0x20,0x30,0x40};
struct st {
        int i;
        int j;
} ;
static struct st ar[10];
void main(void)
{
        int b;
        a = 10;
```

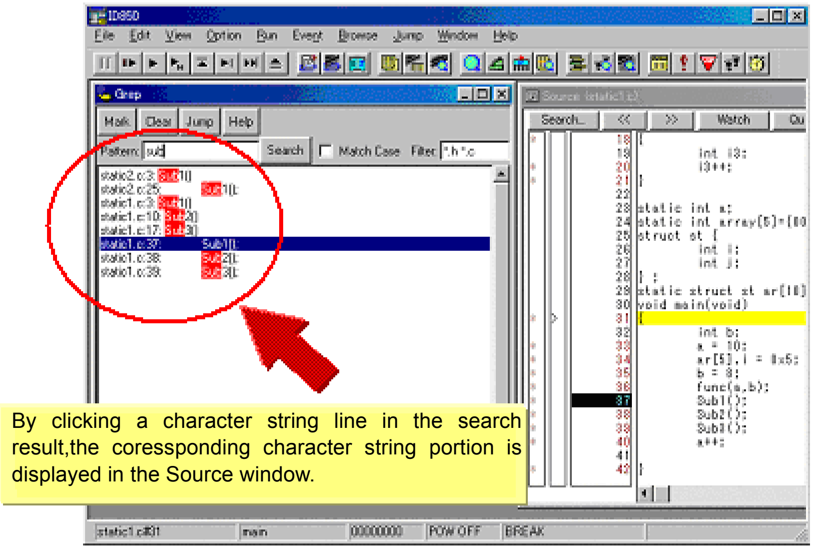
When a function name in the list is clicked, the corresponding function portion is displayed in the Source window.

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



```
static2.c:3: Sub1()
static2.c:25:        Sub1();
static1.c:3: Sub1()
static1.c:10: Sub2()
static1.c:17: Sub3()
static1.c:37:        Sub1();
static1.c:38:        Sub2();
static1.c:39:        Sub3();
```

```
{
        int i3;
        i3++;
}

static int a;
static int array[5]=[00
struct st {
        int i;
        int j;
} ;
static struct st ar[10]
void main(void)
{
        int b;
        a = 10;
        ar[5].i = 0x5;
        b = 8;
        func(a,b);
        Sub1();
        Sub2();
        Sub3();
        a++;

}
```

By clicking a character string line in the search result,the coressponding character string portion is displayed in the Source 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.

RRM window

This is a dedicated window for RAM monitoring.

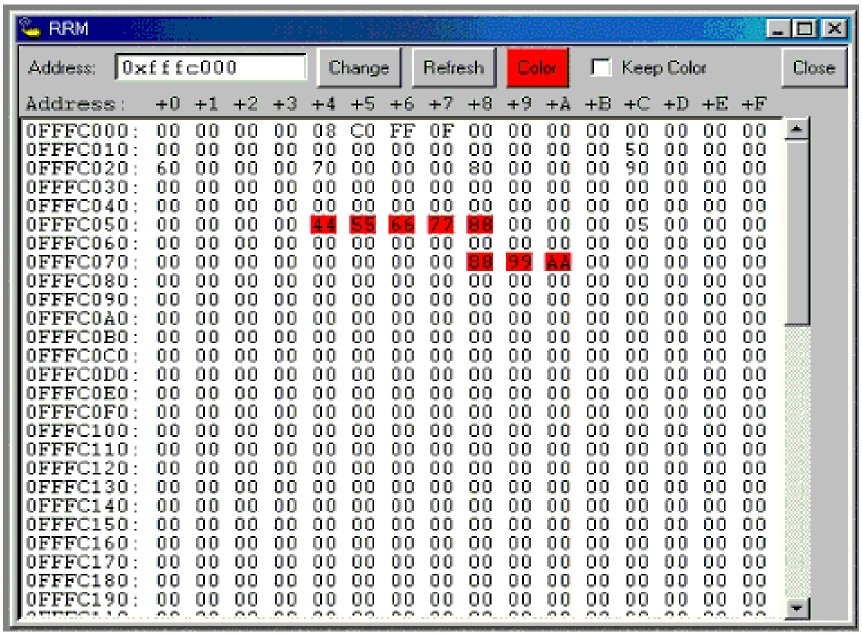
The address area in which a value was changed in the RAM area during program execution is highlighted with a color. The display range is 1 KB. With reading RAM, execution of the user program momentarily breaks. On this window, the start address of the RAM area can be changed while the user program is being executed.

Caution 1 All data are not read at the same time (because data of 1 KB is divided and read in word units).

Caution 2 This RRM window is opened even when the RAM monitor function is set to OFF in the Extended Option dialog box.

Caution 3 The sampling interval is about 0.3 - 0.7 seconds (20MHz), but it depend on the frequency of the CPU operation (when the N-Wire CARD is connected).

Figure A-3 RRM window



Address: 0xfffc000

```
Address:  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
0FFFC000: 00 00 00 00 08 C0 FF 0F 00 00 00 00 00 00 00 00
0FFFC010: 00 00 00 00 00 00 00 00 00 00 00 00 50 00 00 00
0FFFC020: 60 00 00 00 70 00 00 00 80 00 00 00 90 00 00 00
0FFFC030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC050: 00 00 00 00 44 55 66 77 88 00 00 00 05 00 00 00
0FFFC060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC070: 00 00 00 00 00 00 00 00 88 99 AA 00 00 00 00 00
0FFFC080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC0A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC0B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC0C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC0D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC0E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC0F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0FFFC190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Object	Function
Address	Input the start address to be displayed (automatically aligned to 1 KB.)
<Change>button	Switch the start address display.
<Refresh>button	Reads data from the memory.
<Color>button	The color can be customized. The default color is red.
Keep Color	Specify whether or not to hold the color highlighting. Selected: Once a value is changed, the color highlighting is held until a break occurs. Unselected: The color is cleared if there is no change of values(default).
<Close> button	Closes this window.

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.

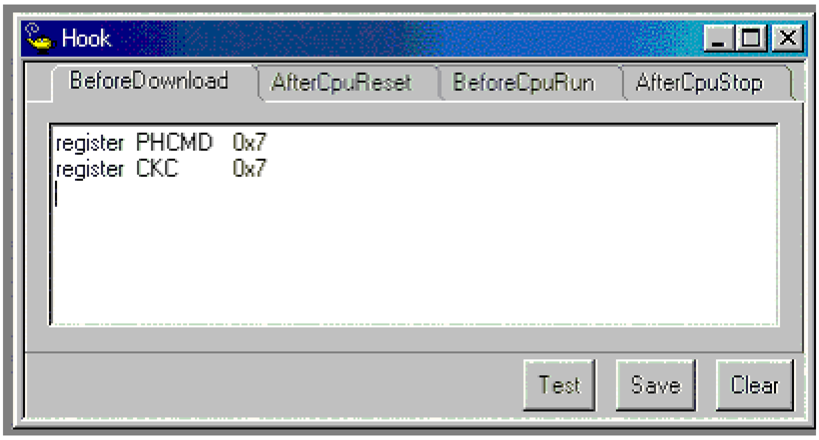
On this window, a hook can be set by using the following four tabs.

- [BeforeDownload] tab: Hook before downloading
- [AfterCpuReset] tab: Hook after CPU reset
- [BeforeCpuRun] tab: Hook before start of execution
- [AfterCpuStop] tab: Hook after break

By setting a peripheral I/O registers by using the [BeforeDownload] tab before downloading the load module, for example, downloading can be executed at high speeds. Access to the external memory is also facilitated by using this tab.

If the setting is saved as "project-file-name.tcl" in the directory where the project is stored, the setting is executed when the project is next opened.

Figure A-4   Hook window



```
register PHCMD  0x7
register CKC    0x7
```

Object	Function
[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 resetting CPU 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.

Object	Function
[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 ID850QB was activated from a project file, the file is saved as "project-file-name.tcl".
<Clear>button	Clears all the descriptions on the tabs.

Caution Specify the program register and the peripheral I/O registers for the register name.

Memory Mapped I/O window

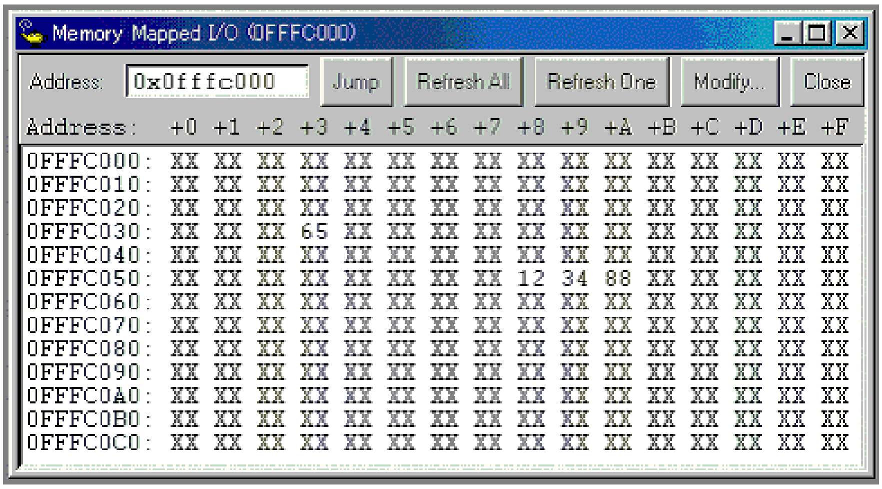
Data is explicitly read or written at a specified address in this window.

When a write is performed in the Memory window, the data is internally read and verified by the ID850QB. In addition, the memory can also be read simply by scrolling in the Memory window. On the Memory Mapped I/O window, however, the above operations are not performed.

Therefore, this window is useful for reading or writing a specific address.

While the user program is being executed, it momentarily breaks before data is written in this window.

Figure A-5 Memory Mapped I/O window



Address: 0x0fffc000

```
Address:  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
0FFFC000: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0FFFC010: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0FFFC020: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0FFFC030: XX XX XX 65 XX XX XX XX XX XX XX XX XX XX XX XX
0FFFC040: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0FFFC050: XX XX XX XX XX XX XX XX 12 34 88 XX XX XX XX XX
0FFFC060: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0FFFC070: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0FFFC080: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0FFFC090: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0FFFC0A0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0FFFC0B0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0FFFC0C0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
```

Object	Function
Address	Input the address to display. The display target address changes by pressing the Enter key or clicking the <Jump> button.The data contents are not read at this time, so the address (numerical value) is displayed in the address display section, but "XX" is displayed in the data section.
<Jump>button	Jumps to the address input in the Address field.
<Refresh All >button	Reads all the areas currently displayed only once. "ZZ" will be displayed in the data section when an attempt is made to read an unmapped area, or when an error occurs upon a read.
<Refresh One >button	Reads data in the memory of the address at which the cursor is placed only once. The read data size depends on the display format. "ZZ" will be displayed in the data section when an attempt is made to read an unmapped area, or when an error occurs upon a read.
<Modify... >button	Opens the Memory Mapped I/O dialog box. The address at which the cursor is placed is the input address displayed in the Memory Mapped I/O dialog box.If this button is clicked after the cursor position is changed in the Memory Mapped I/O dialog box, the Address field in the Memory Mapped I/O dialog box is also changed.
<Close> button	Closes this window.
Context menu	Select the display format from Byte, HalfWord, and Word.

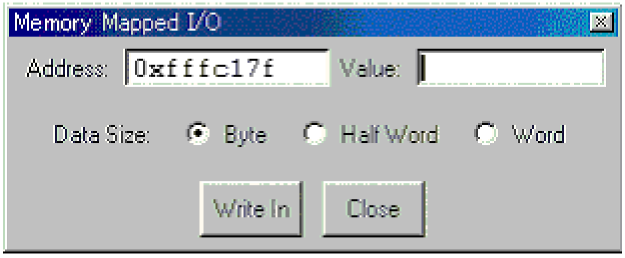
Memory Mapped I/O dialog box

This dialog box can be opened by clicking the <Modify... > button on the Memory Mapped I/O window. It is used to write data to any address.

Caution 1 When the area to which data is written is displayed in the Memory window or the Watch window , data is read in these windows after the <Write in> button is clicked.

Caution 2 If Data Size is less than Access Size specified in the Configuration dialog box, ID850QB reads data in Access Size once, changes the corresponding part of the read data, and writes the changed data in Access Size.

Figure A-6 Memory Mapped I/O dialog box



Address: 0xfffc17f Value:

Data Size: ● Byte ○ Half Word ○ Word

Object	Function
Address	Input the address to be written. The address corresponding to the data for which the cursor is placed in the Memory Mapped I/O window is displayed by default.
Value	Input the value to be written.
Data Size	Select the size of the data to be written. The size specified in the Memory Mapped I/O window is selected by default.
<Write in> button	Data is written to the specified address with the specified size.
<Close> button	Closes this dialog box.

APPENDIX B N-WIRE CHECKER

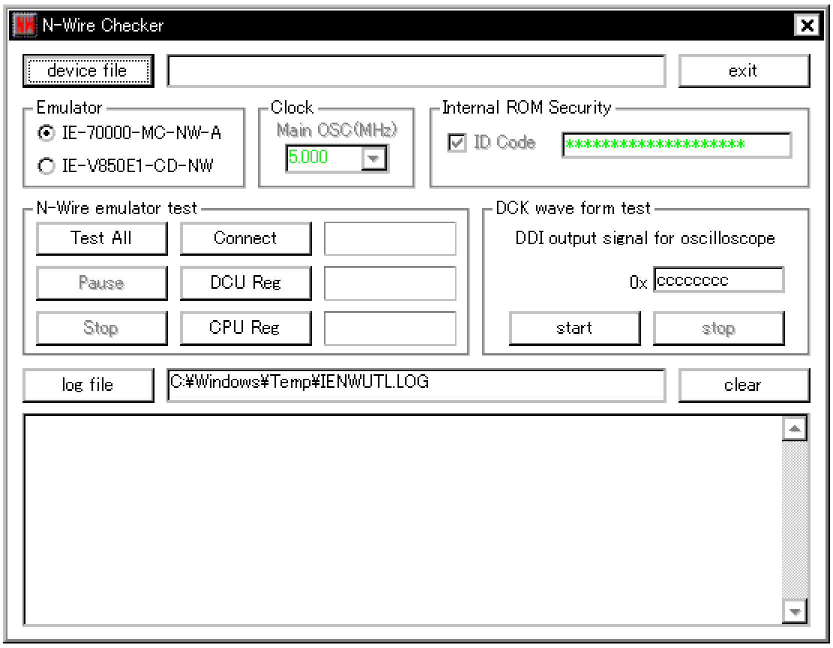
- Overview
- Operation
- Explanation of each area and button
- Corrective Actions in Case of NG
- Restrictions

B. 1 Overview

N-Wire Checker is an application for simply testing the PC connected N-Wire emulator and the target CPU operation.

N-Wire Checker is installed at the same time as the debugger.

Figure B-1 N-Wire Checker



● IE-70000-MC-NW-A
○ IE-V850E1-CD-NW

Main OSC(MHz) 5.000

☑ ID Code ********************

0x cccccccc

C:¥Windows¥Temp¥IENWUTL.LOG

B. 1. 1 Kind of test

The following tests are possible with N-Wire Checker.

(1) N-Wire Emulator test

Checks N-Wire emulator connection, initialization, and target CPU register read/write.

The test results can be output to the log viewer and a log file.

(2) DCK wave form test

Outputs the data set from the N-Wire emulator to DDI signal in order to check the waveform of the DCK signal on the target board with an oscilloscope.

B. 2 Operation

(1) Starting and preparation N-Wire Checker

1) A N-Wire emulator and a target board are turned on.
   Please refer to the user's manual of each N-Wire emulator about connection of a N-Wire emulator, a target board, and a host machine.
2) Select [Start] menu -> [Program] -> [NEC Tools32] -> [N-Wire Checker]. This starts N-Wire Checker. If, at this time, the debugger is already started up, end it.
3) Click the <device file> button. Specified a device file name of target CPU.
4) In the Emulator area, select the N-Wire emulator that is connected (IE-70000-MC-NW-A or N-Wire CARD).
5) In the Clock area, specify the frequency of the main clock input to the device (only when the N-Wire CARD is connected).
6) In the Internal ROM Security area, check the check box and input an ID code when a product with security unit is used in the single-chip mode (only when the N-Wire CARD is connected).

(2) N-Wire Emulator test

1) Press the <Test All> button. This is the same operation as pressing the <Connect>, <DCU reg>, and <CPU reg> buttons in sequence.
2) The test results (OK, NG) and log are displayed.
   Check that all the test are OK.
   If NG is displayed, refer to "B. 4 Corrective Actions in Case of NG".
   For details of the dialog box, refer to "B. 3 Explanation of each area and button".

(3) DCK waveform test

1) Connect the DCK signal of the target board to the oscilloscope probe.

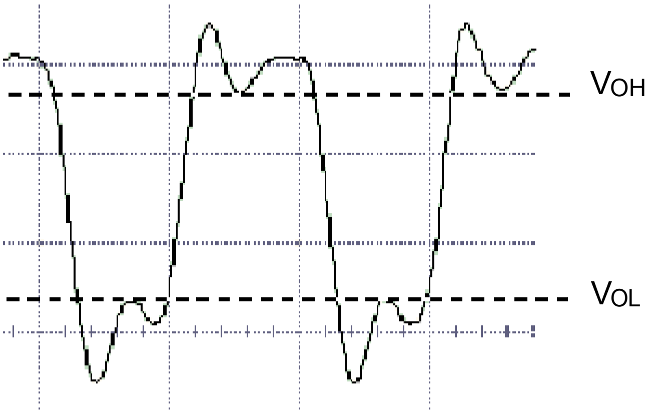
Caution Switch off the power supply of the N-wire emulator and target board before performing this connection.

2) Input 4-byte data (0xcccccccc at the time of shipping) in the DDI output signal for oscilloscope area and then click the <start> button.

3) Monitor the waveform on the oscilloscope.

4) Check whether the waveform noise (reflection) is within the standard range below.
If it is not within the standard range, refer to "B. 4 Corrective Actions in Case of NG".
For details of the dialog box, refer to "B. 3 Explanation of each area and button".

Figure B-2 Waveform of N-Wire Checker



VOH, VOH : Refer to the data sheet supplied with the emulator.

Caution The data is output from the lowermost bit.

-**In the IE-70000-MC-NW-A**

If data from the target board changes, the changed data is subsequently output because data is shifted via the target board.

-**In the N-Wire CARD**

Even if data from the target board changes, the same data is always output because data is shifted inside the emulator.

B. 3 Explanation of each area and button

(1) device file (operating environment setting area)

Object	Function
<device file>	Displays the dialog box for specifying the device file name of the target CPU.
Text area	Displays the device file name of the target CPU and its path. Specify this at N-Wire Checker startup.

(2) Emulator (emulator selecting area)

Object	Function
IE-70000-MC-NW-A	Select this to connect IE-70000-MC-NW-A.
IE-V850E1-CD-NW	Select this to connect N-Wire CARD.

(3) Clock (frequency of the main clock setting area)

Object	Function
Main OSC (MH z)	Input the frequency of the main clock input to the device.

(4) Internal ROM Security (ID code specification area)

Object	Function
ID Code	Check the check box when a product with security unit is used in the single-chip mode.
Input area	Input an ID code when a product with security unit is used in the single-chip mode. For details of the ID code, refer to the "N-Wire CARD User's Manual".

(5) N-Wire emulator test (N-Wire emulator test area)

Object	Function
<Test All>	Tests Connect, DCR Reg, and CPU Reg in sequence.
<Pause>	Pauses the test. (Testing resumes when this button is clicked again.)
<Stop>	Stops the test.
<Connect>	Opens communication with the target and executes testing for initialization.
<DCU Reg>	Performs testing for DCU register read/write.
<CPU Reg>	Performs testing for CPU register read/write.
Text area	A test result is displayed.

(6) DCK wave form test (DCK waveform test area)

Object	Function
DDI output signal for ocsilloscopearea	Sets the data of the DDI signal output from the N-Wire emulator. When a value of 0xffffffff or greater is set, an error occurs when the <start> button is pressed.
<start>	Outputs the value (0 to 0xffffffff) specified in the DDI output signal for oscilloscope text area.
<stop>	Stops the signal for value testing specified in the DDI output signal for oscilloscope text area.

(7) Log viewer area

Object	Function
<log file>	Displays the dialog box for specifying the test result log file name. (Window\TEMP\IENWULT.log at the time of shipping) The log file is overwritten when N-Wire Checker is started up. The following occurs when the file path name is not specified. - When environment variable TEMP is not defined -> A log file is created under environment variable TMP. - When environment variable TMP is not defined -> A log file is created in the current folder.
Text area	Displays the log file name and its path.
<clear>	Clears the log viewer display. If a log file name is specified in the log file text area, the log file contents are not cleared.
Log viewer area	Displays the test results as a log (up to approx. 21,000 characters). The same contents are output to the log file specified in the log file text area.

(8) Other

Object	Function
<exit>	Ends N-Wire Checker.

B. 4 Corrective Actions in Case of NG

B. 4. 1 NG in N-Wire Emulator test

(1) If [Connect] is NG

Take action to correct the error by referring to next table.

Table B-1 Errors of N-Wire Checker

Error Number	Error Messages	Corrective Action
0x0100	Can not communicate with ICE.	Please confirm the installation of the device driver for the PC interface board.
0x0c43	Connection of emulator cannot be performed.	**In the IE-70000-MC-NW-A** Check the power to the in-circuit emulator. Or, the PC-IF cable may not be correctly connected. Check the connection. **In the N-Wire CARD** 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.
0x03a0	Power supply of target is OFF.	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.
0x0c70	DCU cannot be accessed.	Check the power to the chip. Check the connection of the signal lines (DCK, DMS, DDI, DDO, and DRSTZ). Check the noise level by a DCK waveform test.
0x0105	Failed in reading device file (dxxxx.800).	The necessary file may be damaged. Reinstall the device file.
0x0c71	Reset cannot be performed.	Check the clock signal. This may be caused by a stopped clock or a slow clock.
0x0c24	It cannot shift to debug mode.	Check the clock signal. This may be caused by a stopped clock or a slow clock.
0x0c72	Monitor memory cannot be accessed.	Check the noise level by a DCK waveform test. The problem may also be caused by an internal chip problem.
0x0c73	Monitor execution cannot be performed.	Check the noise level by a DCK waveform test. The problem may also be caused by an internal chip problem.
0x0c74	CPU register cannot be accessed.	Check the noise level by a DCK waveform test. The problem may also be caused by an internal chip problem.
0xf603	Incorrect ID Code.	This may be caused by the following (when the N-Wire CARD 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 7 of address 0x79) is 0. ->Erase the internal flash memory once.

Table B-1 Errors of N-Wire Checker

Error Number	Error Messages	Corrective Action
0xf604	Incorrect ID Code. Abort the N-Wire checker.	This may be caused by the following (when the N-Wire CARD 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 7 of address 0x79) is 0. ->Erase the internal flash memory once.

(2) If [DCU reg] is NG

Normal access to the DCU register seems impossible.

Perform the DCK waveform test.

If no anomaly is found with the DCK waveform test, the register may be damaged, so try again with a different chip .

(3) If [CPU reg] is NG

Normal access to the CPU register seems impossible.

Perform the DCK waveform test.

If no anomaly is found with the DCK waveform test, the register may be damaged, so try again with a different chip .

B. 4. 2 NG in DCK waveform test

Check the following items regarding the design of the target board, in accordance with the User's Manual of the N-Wire emulator.

(1) IE-70000-MC-NW-A

- Is the pattern of the N-Wire interface signal 100 mm or less?
- Is the DCK signal shielded through a 4.7 k Ohm pull-up resistor + GND?
- Is a 4.7 k Ohm pull-up resistor connected to the DMS signal and the DDI signal?
- Is a 22 Ohm damping resistor connected to the DDO signal?
- Is a 50 k Ohm pull-down resistor connected to the DRSTZ signal?

(2) N-Wire CARD

- Is the pattern of the N-Wire interface signal 100 mm or less?
- Is the DCK signal shielded through a pull-up resistor* + GND?
- Is a pull-up resistor* connected to the DMS, DDI, and DDO signals?
- Is a damping resistor* connected to the DDO signal?
- Is a pull-down resistor* connected to the _DRSTZ signal (or DRST signal)?

Caution The resistance of the resistor marked * above must be as specified for each device. Some devices have on-chip pull-up/down resistors. With these devices, an external resistor is not necessary.

Also check the following items regarding the power supply capacity of the target board.

- Is a power supply with a sufficient capacity used?
- Are sufficient capacitors provided on the target board?
- Is a bypass capacitor connected to all the power supply pins of the CPU?
- Is the GND plane on the target board sufficiently large?

B. 5 Restrictions

Up to 21000 characters can be displayed in the log view area. If this is exceeded, the displayed information is overwritten starting from the oldest data. if the system resources are running extremely short, the N-Wire Checker stops output to the log view area. For accurate information, therefore, use the log file. Information is continuously recorded in the log file even while screen output is stopped. The log file is overwritten once when the N-Wire checker is started.

APPENDIX C INPUT CONVENTIONS

C. 1 Usable Character Set

Table C-1 List of Character Set

Classification	Character
Alphabetic characters	Uppercase : A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Lowercase : a b c d e f g h i j k l m n o p q r s t u v w x y z
Numerals	0123456789
Character equivalent to alphabetic character	@ _

Table C-2 List of Special Characters

Character	Name	Usage
(	Left parenthesis	Changes operation order.
)	Right parenthesis	Changes operation order.
+	Plus	Addition operator or positive sign
-	Minus	Subtraction operator or negative sign
*	Asterisk	Multiplication operator or indirect reference operator
/	Slash	Division operator
%	Percent	Remainder operator
~	Tilde	Complement operator
|	Vertical line	Bit sum operator
^	Circumflex	Bit difference operator
&	Ampersand	Bit product operator or address operator
[	Left bracket	Array subscript operator or base register specification symbol
]	Right bracket	
.	Period	Direct member operator or bit position specifier
,	Comma	Delimiter between operands

C. 2 Symbols

(a) A symbol consists of characters A to Z, a to z, @, _ (underbar), . (period) and 0 to 9.

(b) A symbol must start with a character other than numerals 0 to 9.

(c) Uppercase characters (A to Z) and lowercase characters (a to z) are distinguished.

(d) 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).

(e) A symbol is defined by loading a load module file.

(f) Symbols are classified into the following types by the valid range:

- Global symbol (assembly language, C language)

- Static symbol (C language)
 In-file static symbol
 In-function static symbol

- Local symbol (C language)
 In-file local symbol
 In-function local symbol
 In-block local symbol

(g) The following symbols are available for each language used:

- Assembly language
  Label 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)

(h) A symbol can be described instead of an address or numeric value.

(i) The valid range of a symbol is determined based on the source debug information when the source file is assembled or compiled.

(j) Describe only the symbol name of a global symbol.

(k) A local symbol is expressed in pairs with a file name.

C. 3 Numeric Values

The following four types of numeric values can be used. The input format of each type is as shown below.

The suffix (bold) and the alphabetic characters of hexadecimal numbers may be uppercase or lowercase characters. If the first character is A to F, 0 must be prefixed to it.

In the input field of ID850QB, decimal numbers or hexadecimal numbers are alternately selected, depending on the default radix.

Table C-3 Input Format of Numeric Values

Numeric Value	Input Format
Binary number	n**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)

C. 4 Expressions and Operators

(1) Expressions

An expression consists of constants, register names, IOR names, and symbols coupled by operators.

If IOR name, label name, function name, or variable name is described as a symbol, an address is calculated as the value of the symbol.

The elements making up an expression, except operators, are called terms (constants and labels). Terms are called the first term, the second term, and so on, starting from the left.

(2) Operators

The following operators of the C language can be used:

Table C-4 List of Operators

Symbol	Meaning	Explanation
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 result.
MOD %	Remainder	Divides the value of the first term by the value of the second term, and returns the remainder of the result.
- 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 result.
AND &	Logical product	Obtains the logical product of the values of the first and second terms on each bit, and returns the result.
OR |	Logical sum	Obtains the logical sum of the values of the first and second terms on each bit, and returns the result.
XOR ^	Exclusive logical sum	Obtains the exclusive logical sum of the values of the first and second terms on each bit, and returns the result.
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 result. 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 result. 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.

Table C-4 List of Operators

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 C-5 Operator Priority

Priority	Operators
1 Higher	(,)
2	+ sign, - sign, NOT, ~, HIGHT, LOW, HIGHW, LOWW
3	*, /, MOD, %, SHR, >>, SHL, <<
4	+, -
5	AND, &
6 Lower	OR, |, XOR, ^

- If the priorities of the operators are the same, the operation is performed from the left toward the right.
- Performs the operation in ( ) before the operation outside ( ).
- Each term in an operation is treated as unsigned 32-bit data.
- All operation results are treated as unsigned 32-bit data.
- If an overflow occurs during operation, the lower 32 bits are valid, and the overflow is not detected.

(4) Terms

To describe a constant for a term, the following numeric values can be described.

Table C-6 Range of Radixes

Radix	Range
Binary number	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 <= 0FFFFFFFF**H**

APPENDIX D KEY FUNCTION LIST

Table D-1 Key Function List

Key	Function
BackSpace	Deletes one character before the cursor and moves the cursor to the position of the deleted character. At this time, the character string following the cursor moves forward.
Delete	- 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.
F2	Forcibly stops program execution. Same function as [Run] menu -> [Stop].
F3	Resets the CPU. Same function as [Run] menu -> [CPU Reset].

Table D-1 Key Function List

Key	Function
F4	Resets the CPU and executes the program. Same function as [Run] menu -> [Restart].
F5	Executes the program. Same function as [Run] menu -> [Go].
F6	Executes the program to the cursor position in the Source or Assemble window. Same function as [Run] menu -> [Come Here].
F7	The user program is real-time executed until execution returns. Same function as [Run] menu -> [Return Out].
F8	Step execution. Same function as [Run] menu -> [Step In].
F9	Sets a breakpoint at cursor position in Source or Assemble window. Same function as [Run] menu -> [Break Point].
F10	Next step execution. Same function as [Run] menu -> [Next Over].
F11	Sets or deletes a software breakpoint. Same function as [Run] menu -> [Software Break Point].
Shift+End	Expands the selection range to the end of the line.
Shift+Home	Expands the selection range to the beginning of the line.
Shift+Left arrow key	Expands the selection range one character to the left.
Shift+Right arrow key	Expands the selection range one character to the right.
Shift+F6	Executes the program from the cursor position in the Source or Assemble window. Same function as [Run] menu -> [Start From Here].
Shift+F9	Resets the CPU. Same function as [Run] menu -> [CPU Reset].
Ctrl+End	Displays the last line. The cursor will also move to the last line.
Ctrl+Home	Displays the first line. The cursor will also move to the first line.
Ctrl+Left arrow key	Moves the cursor one word to the left. If the cursor at the left most position on the screen, scrolls the screen one column to the right.
Ctrl+Right arrow key	Moves the cursor one word to the right. If the cursor at the right most position on the screen, scrolls the screen one column to the left.
Ctrl+F5	Ignores break points being set, and executes the program. Same function as [Run] menu -> [Ignore break points and Go].
Ctrl+F9	Sets the address at the cursor position in the Source window or Assemble window to the PC. Same function as [Run] menu -> [Change PC].
Ctrl+A	Selects all the events registered to the Event Manager. Same function as [View] menu -> [Select All Event] in the Event Manager.
Ctrl+C	Copies a selected character string and saves it to the clipboard buffer.
Ctrl+D	Disassembles and displays the results from the jump destination address specified by the data value selected in the current window.Opens the Assemble window. Same function as [Jump] menu -> [Assemble].
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].

Table D-1 Key Function List

Key	Function
Ctrl+G	Performs a search.Opens the search dialog box corresponding to the current window. Same function as [View] menu -> [Search...].
Ctrl+J	Moves the display position. Opens the each dialog box, depending on the current window. Same function as [View] menu -> [Move...].
Ctrl+M	Displays the memory contents from the jump destination address specified by the data value selected in the current window. Opens the Memory window. Same function as [Jump] menu -> [Memory...].
Ctrl+O	Loads a view file, source file, or text file. Opens the View File Load dialog box. The operation will differ depending on the extension of the file. view file: Displays the file in the corresponding window. Others: Displays the file in the Source window. Same function as [File] menu -> [Open...].
Ctrl+S	Saves the data displayed in the current window to the view file. Same function as [View] menu -> [Save...].
Ctrl+U	Displays the corresponding source text and source line, using the data value selected in the current window as the jump destination address. Opens the Source window. Same function as [Jump] menu -> [Source Text].
Ctrl+V	Pastes the contents of the clipboard buffer to the text cursor position.
Ctrl+W	Temporarily displays the contents of the specified data. Opens the Quick Watch dialog box. Same function as [View] menu -> [Quick Watch...].
Ctrl+X	Cuts a selected character string and saves it to the clipboard buffer. Same function as [Edit] menu -> [Cut].
Ctrl+Shift+Left arrow key	Expands the selection range one word to the left.
Ctrl+Shift+Right arrow key	Expands the selection range one word to the right.

APPENDIX E MESSAGES

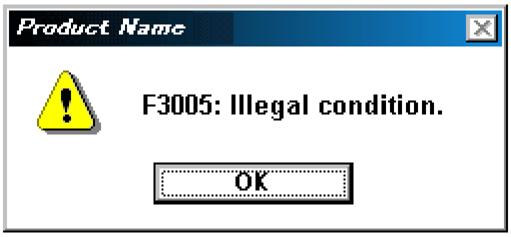
- Display Format
- Types of Messages
- Message Lists

E. 1 Display Format

Messages are output to the error/warning dialog box.

By pressing the F1 key while the error/warning dialog box is open, the related online help files are displayed.

Figure E-1 Error/Warning Messages



F3005: Illegal condition.

E. 2 Types of Messages

The ID850QB outputs the following types of messages.

Table E-1 Types of Messages

Type	Meaning
A*xxxx*	Fatal error messages Stops processing, and terminates the debugger. If this error occurs, debugging cannot be continued.
F*xxxx*	Syntax error messages Stops processing, and opened windows and dialog boxes are closed.
W*xxxx*	Warning messages or question messages Stops processing, but opened windows and dialog boxes are not closed.

E. 3 Message Lists

< X0000~ > < X1000~ > < X2000~ > < X3000~ > < X4000~ > < X5000~ > < X6000~ > < X7000~ > < X8000~ > < X9000~ > < Xa000~ > < Xb000~ > < Xc000~ > <Xd000~ > < Xe000~ > < Xf000~ >

(1) *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 (d3xxx.800). 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.
F010a:	Can not communicate. Please confirm the availability of the communication port.
A01a0:	No response from the evachip. Please confirm the signal of the CLOCK or RESET WAIT, HLDRQ and so on.(IECUBE) No response from the CPU. Please confirm the signal of the CLOCK or RESET WAIT, HLDRQ and so on.(N-Wire CARD) No response from the emulation CPU. Please confirm the signal of the CLOCK or RESET,WAIT and so on.(78K0) 1) Check the HOLD signal, WAIT signal, clock signal, etc. The IOR value may not be correct.
A01a1:	Failed in reading ie703000.ie.
A01a2:	Break board is not connected.
A01a3:	Emulation board is not connected.
A01a4:	Board configuration of ICE is not consistent.
A01a5:	POD/EM1 board is not connected.

A01a6:	Executor is running.
A01a7:	Failed in reading micro program file (m0xxx.78k).
A01a8:	Failed in reading initialization file (expc.ini).(78K) Failed to find configuration file (lv8hw.ini). (V850)
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).(78K) Failed in reading configuration file (lv8hw.ini). (V850)
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.
A01d0:	Not enough memory in starting simulator.
A01d1:	Not enough memory in starting simulator.
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 IOR 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 IOR 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.

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.

F0b61: Section Trace event conditions overflow.

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 IOR 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.

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. (N-Wire CARD)

Can not communicate with ICE. Please confirm the power of ICE, connection of the interface cable. (IECUBE)

1) The switch setting may be wrong if a desktop computer is used and two or more PC cards are inserted. Check the setting. Or it may have malfunctioned (when the N-Wire CARD is connected).

2) Check the power to the in-circuit emulator and cable connections The switch setting may be wrong if a desktop computer is used and two or more PC cards are inserted. Check the setting. Or it may have malfunctioned (when the IECUBE is connected).

F0c44: Coverage test is being executed.

F0c45: Inside of Power off reset emulation cannot carry out program execution.

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) The device file selection may be incorrect. Select a device file that supports the target chip in Chip Selection in the Configuration dialog box. Check the power to the chip. Check the connection of the signal lines (DCK, DMS, DDI, DDO, and DRSTZ). Check the noise level by a DCK waveform test on the N-Wire Checker (when the IE-70000-MC-NW-A, N-Wire CARD is connected).

2) 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. Check the setting of main clock in the Configuration dialog box , the noise level by a DCK waveform test on the N-Wire Checker. The problem may also be caused by an internal chip problem (when the is N-Wire CARD, IECUBE connected).

2) 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).

3) Check the noise level by a DCK waveform test on the N-Wire Checker. The problem may also be caused by an internal chip problem (when the IE-70000-MC-NW-A is connected).

F0c73: Monitor execution cannot be performed.

1) Check the noise level by a DCK waveform test on the N-Wire Checker. The problem may also be caused by an internal chip problem (when the IE-70000-MC-NW-A , N-Wire CARD is connected).

2) IE may be malfunctioning (when the IECUBE is connected).

F0c74: CPU register cannot be accessed.

1) Check the noise level by a DCK waveform test on the N-Wire Checker. The problem may also be caused by an internal chip problem (when the N-Wire CARD is connected).

2) 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) Initial state at the time of DCU access start is unusual (does not start and remains in the reset state). Check the connection of the signal lines (DCK, DMS, DDI, DDO, and DRSTZ). Check the noise level by a DCK waveform test on the N-Wire Checker (when the N-Wire CARD is connected).

2) 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) DCU access is unusuall (verify error). Check the connection of the signal lines (DCK, DMS, DDI, DDO, and DRSTZ). Check the noise level by a DCK waveform test on the N-Wire Checker (when the N-Wire CARD is connected).

2) 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.(V850)

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: Device file which does not correspond to on tip debug.

1) An attempt was made to start with a device file not supporting on-chip debugging. The device file may be old. Install the latest device file (when the N-Wire CARD is connected).

2) IE may be malfunctioning (when the IECUBE is connected).

F0ca3: Information that it does not support is included in on tip debug information on 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 : Device file which does not correspond to IECUBE.

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 (when the IECUBE is connected).

F0caf: Trace block can not be stepped over.

(2) *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.
F1019:	Illegal Internal ROM size or Internal RAM size.
A10ff:	Can not communicate with ICE.
A1dbe:	Error occurred inside debugger.

(3) *X*2000~

F2000:	Illegal IOR name.
A2001:	Illegal address.
F2002:	User program is running.
F2003:	Illegal IOR number.
F2004:	Illegal bit number.
W2005:	IOR of Read Protect attribute was specified.
F2006:	Hidden IOR was specified.
F2007:	IOR of ban read or write was specified.
F2008:	IOR not existing was specified.
A2009:	Device file is damaged or error is in file.
F200a:	Illegal value specified for IOR.
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 IOR.
F200e:	IOR area can not be accessed.
A20ff:	Can not communicate with ICE.
A2222:	Illegal condition.

(4) *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 IOR 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) *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) *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 ie703000.ie.

F500a: Specified device file is illegal version.

1) Necessary files may be damaged. Reinstall the device file.

W500b: Specified device file does not relocate 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) *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) *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) *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) *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) *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.

(12) *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) *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 (ELF) 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 CA 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.

Fc101e: Flash erase is not supported on IECUBE.

Fc100: This feature is not supported.

(14) *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) *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) *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.
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 IOR 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/peripheral IOR - 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 N-Wire CARD 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 7 of address 0x79) is 0.

->Erase the internal flash memory once.

2) 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 N-Wire CARD 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 7 of address 0x79) is 0.

->Erase the internal flash memory once.

2) 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.

Ff606: Please check connection with the target board, and power on it.

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.
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 ID850QB is terminated. <No> button is selected, execution of the user program is not stopped and the Exit Debugger dialog box is not displayed. The ID850QB is not terminated.

Ffc00:	Online help window cannot be started. Please install HTML Help environment with reference to a users manual.
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 F INDEX