

To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

"Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

To all our customers

Regarding the change of names mentioned in the document, such as Hitachi Electric and Hitachi XX, to Renesas Technology Corp.

The semiconductor operations of Mitsubishi Electric and Hitachi were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Renesas Technology Home Page: <http://www.renesas.com>

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

Cautions

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.

Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.

SH7047F E10A Emulator

User's Manual

Renesas Microcomputer
Development Environment
System

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

IMPORTANT INFORMATION

READ FIRST

- **READ this user's manual before using this emulator product.**
- **KEEP the user's manual handy for future reference.**

Do not attempt to use the emulator product until you fully understand its mechanism.

Emulator Product:

Throughout this document, the term "emulator product" shall be defined as the following products produced only by Hitachi, Ltd. excluding all subsidiary products.

- Emulator
- User system interface cable

The user system or a host computer is not included in this definition.

Purpose of the Emulator Product:

This emulator product is a software and hardware development tool for systems employing the Hitachi microcomputer. This emulator product must only be used for the above purpose.

Limited Applications:

This emulator product is not authorized for use in MEDICAL, atomic energy, aeronautical or space technology applications without consent of the appropriate officer of a Hitachi sales company. Such use includes, but is not limited to, use in life support systems. Buyers of this emulator product must notify the relevant Hitachi sales offices before planning to use the product in such applications.

Improvement Policy:

Hitachi, Ltd. (including its subsidiaries, hereafter collectively referred to as Hitachi) pursues a policy of continuing improvement in design, performance, and safety of the emulator product. Hitachi reserves the right to change, wholly or partially, the specifications, design, user's manual, and other documentation at any time without notice.

Target User of the Emulator Product:

This emulator product should only be used by those who have carefully read and thoroughly understood the information and restrictions contained in the user's manual. Do not attempt to use the emulator product until you fully understand its mechanism.

It is highly recommended that first-time users be instructed by users that are well versed in the operation of the emulator product.

LIMITED WARRANTY

Hitachi warrants its emulator products to be manufactured in accordance with published specifications and free from defects in material and/or workmanship. Hitachi, at its option, will replace any emulator products returned intact to the factory, transportation charges prepaid, which Hitachi, upon inspection, shall determine to be defective in material and/or workmanship. The foregoing shall constitute the sole remedy for any breach of Hitachi's warranty. See the Hitachi warranty booklet for details on the warranty period. This warranty extends only to you, the original Purchaser. It is not transferable to anyone who subsequently purchases the emulator product from you. Hitachi is not liable for any claim made by a third party or made by you for a third party.

DISCLAIMER

HITACHI MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, ORAL OR WRITTEN, EXCEPT AS PROVIDED HEREIN, INCLUDING WITHOUT LIMITATION THEREOF, WARRANTIES AS TO MARKETABILITY, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR USE, OR AGAINST INFRINGEMENT OF ANY PATENT. IN NO EVENT SHALL HITACHI BE LIABLE FOR ANY DIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY NATURE, OR LOSSES OR EXPENSES RESULTING FROM ANY DEFECTIVE EMULATOR PRODUCT, THE USE OF ANY EMULATOR PRODUCT, OR ITS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. EXCEPT AS EXPRESSLY STATED OTHERWISE IN THIS WARRANTY, THIS EMULATOR PRODUCT IS SOLD "AS IS", AND YOU MUST ASSUME ALL RISK FOR THE USE AND RESULTS OBTAINED FROM THE EMULATOR PRODUCT.

State Law:

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may have other rights which may vary from state to state.

The Warranty is Void in the Following Cases:

Hitachi shall have no liability or legal responsibility for any problems caused by misuse, abuse, misapplication, neglect, improper handling, installation, repair or modifications of the emulator product without Hitachi's prior written consent or any problems caused by the user system.

All Rights Reserved:

This user's manual and emulator product are copyrighted and all rights are reserved by Hitachi. No part of this user's manual, all or part, may be reproduced or duplicated in any form, in hard-copy or machine-readable form, by any means available without Hitachi's prior written consent.

Other Important Things to Keep in Mind:

1. Circuitry and other examples described herein are meant merely to indicate the characteristics and performance of Hitachi's semiconductor products. Hitachi assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein.
2. No license is granted by implication or otherwise under any patents or other rights of any third party or Hitachi.

Figures:

Some figures in this user's manual may show items different from your actual system.

Device names:

Sections 1 to 5 in this user's manual use SHxxxx as an example of the device names.

Limited Anticipation of Danger:

Hitachi cannot anticipate every possible circumstance that might involve a potential hazard. The warnings in this user's manual and on the emulator product are therefore not all inclusive. Therefore, you must use the emulator product safely at your own risk.

SAFETY PAGE

READ FIRST

- **READ** this user's manual before using this emulator product.
- **KEEP the user's manual handy for future reference.**

Do not attempt to use the emulator product until you fully understand its mechanism.

DEFINITION OF SIGNAL WORDS



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.



DANGER indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.



WARNING indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.



CAUTION indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.



CAUTION used without the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in property damage.

NOTE emphasizes essential information.

WARNING

Observe the precautions listed below. Failure to do so will result in a FIRE HAZARD and will damage the user system and the emulator product or will result in PERSONAL INJURY. The USER PROGRAM will be LOST.

- 1. Do not repair or remodel the emulator product by yourself for electric shock prevention and quality assurance.**
- 2. Always switch OFF the host computer and user system before connecting or disconnecting any CABLES or PARTS.**
- 3. Connect the connectors in the user system and in the user interface cable by confirming the correct direction.**
- 4. If the PCI interface board for the E6000 or E8000 emulator (HS6000EIC01H) and the E10A emulator PCI card are mounted on the same host computer, the connectors may be illegally connected.**

Warnings on Emulator Usage

Be sure to read and understand the warnings below before using this emulator. Note that these are the main warnings, not the complete list.

WARNING

Always switch OFF the host computer and user system before connecting or disconnecting any CABLES or PARTS.

Failure to do so will result in a FIRE HAZARD and will damage the user system and the emulator product or will result in PERSONAL INJURY. The USER PROGRAM will be LOST.

CAUTION

Place the host computer and user system so that no cable is bent or twisted. A bent or twisted cable will impose stress on the user interface leading to connection or contact failure.

Make sure that the host computer and the user system are placed in a secure position so that they do not move during use nor impose stress on the user interface.

Preface

Thank you for purchasing the E10A emulator.

CAUTION

READ section 2, Preparation before Use, of this User's Manual before using the emulator product. Incorrect operation will damage the user system and the emulator product.

This emulator is an efficient development tool for software and hardware of user systems based on Hitachi's original microprocessor. The emulator operates using the Hitachi debugging interface (hereafter referred to as the HDI), which is the interface program that runs on Microsoft® Windows® 98, Microsoft® Windows® ME, Microsoft® Windows® 2000, or Microsoft® Windows NT® operating system.

This manual describes the functions and operating procedures of the E10A emulator. Sections 1 to 5 describe common features of all types of E10A emulators. Section 6 describes supplements to the E10A emulator. Read section 1.1, Warnings, carefully before using the emulator.

This manual consists of six sections. The information contained in each section is summarized below:

- Section 1, Overview, gives the emulator overview.
- Section 2, Preparation before Use, gives instructions for first-time users, such as preparation before use and system connection.
- Section 3, Tutorial, describes HDI operating examples.
- Section 4, Descriptions of Windows, describes HDI windows for operating the emulator.
- Section 5, Command-line Functions describes how to input HDI commands and command types.
- Section 6, SHxxxx E10A Emulator Specifications describes the features of the E10A emulator for each MCU. Section 7 describes the important information of the E10A emulator according to emulator products. Read these sections before using the E10A emulator.

The HDI installation disks are provided by the CD-R. Refer to the descriptions in the manuals of the host computer or operating system.

Related Manuals:

- SH Series Cross Assembler User's Manual
- H Series Linkage Editor User's Manual
- H Series Librarian User's Manual
- SuperH RISC Engine C/C++ Compiler User's Manual
- Hitachi Debugging Interface User's Manual
- Hardware Manual for each MCU
- Programming Manual for each MCU

Notes: 1. IBM PC is a registered trademark of International Business Machines Corporation.

2. Microsoft[®], Windows[®], and Windows NT[®] are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Microsoft[®] Windows[®] 98 operating system is referred to as Windows[®] 98 in this user's manual.

Microsoft[®] Windows[®] ME operating system is referred to as Windows[®] ME in this user's manual.

Microsoft[®] Windows[®] 2000 operating system is referred to as Windows[®] 2000 in this user's manual.

Contents

Section 1 Overview	1
1.1 Warnings	3
1.2 Environmental Conditions	4
1.3 Components	6
Section 2 Preparation before Use.....	7
2.1 Emulator Preparation	7
2.2 HDI Installation	8
2.2.1 Installing under Windows®98 and Windows®ME Operating Systems	8
2.2.2 Installing under Windows NT® 4.0 Operating System.....	9
2.2.3 Installing under Windows®2000 Operating System	10
2.3 Connecting the Card Emulator to the Host Computer	11
2.4 Connecting the Card Emulator to the User System.....	12
2.5 System Check.....	15
2.6 Ending the HDI.....	22
2.7 Uninstalling the HDI.....	23
2.8 CD-R.....	24
2.8.1 Configuration of the CD-R	24
2.9 Support.....	24
Section 3 Tutorial.....	25
3.1 Introduction.....	25
3.2 Running the HDI.....	27
3.3 [HDI] Window	28
3.4 Setting up the Emulator.....	29
3.5 Setting the [Configuration] Dialog Box	30
3.6 Downloading the Tutorial Program	31
3.6.1 Downloading the Tutorial Program	31
3.6.2 Displaying the Source Program	32
3.7 Setting a Software Breakpoint	34
3.8 Setting Registers	35
3.9 Executing the Program.....	37
3.10 Reviewing Breakpoints.....	39
3.11 Viewing Memory	40
3.12 Watching Variables.....	41
3.13 Stepping Through a Program	44
3.13.1 Executing [Step In] Command.....	44
3.13.2 Executing [Step Out] Command.....	46
3.13.3 Executing [Step Over] Command	48

3.14	Forced Breaking of Program Executions	50
3.15	Displaying Local Variables.....	51
3.16	Break Function.....	52
3.16.1	Software Break Function	52
3.17	Hardware Break Function	58
3.17.1	Setting the Sequential Break Condition	66
3.18	Trace Functions.....	70
3.18.1	AUD Trace Function.....	71
3.19	Stack Trace Function	73
3.20	What Next?	75
 Section 4 Descriptions of Windows		77
4.1	HDI Windows	77
4.2	Descriptions of Each Window	80
4.2.1	[Configuration] Dialog Box	80
4.2.2	[Breakpoints] Window.....	87
4.2.3	[Break] Dialog Box.....	90
4.2.4	[Break Point] Dialog Box	96
4.2.5	[Break Condition] Dialog Box.....	98
4.2.6	[Break Condition] Dialog Box Pages.....	100
4.2.7	[Trace] Window.....	109
4.2.8	[Trace Acquisition] Dialog Box.....	111
4.2.9	[System Status] Window.....	113
 Section 5 Command-line Functions		115
5.1	Table and Symbol Description.....	115
5.1.1	Format.....	115
5.1.2	Parameter Input.....	115
5.1.3	Examples.....	116
5.1.4	Related Items	116
5.2	Command Descriptions.....	117
5.2.1	AUD_MODE:AUM.....	118
5.2.2	AUD_TRACE:AUT	120
5.2.3	BREAKCONDITION_CLEAR: BCC.....	122
5.2.4	BREAKCONDITION_DISPLAY: BCD.....	123
5.2.5	BREAKCONDITION_ENABLE: BCE	124
5.2.6	BREAKCONDITION_SET: BCS	125
5.2.7	BREAKPOINT: BP	129
5.2.8	BREAKPOINT_CLEAR: BC.....	130
5.2.9	BREAKPOINT_DISPLAY: BD.....	131
5.2.10	BREAKPOINT_ENABLE: BE	132
5.2.11	DEVICE_TYPE: DE	133
5.2.12	GO_OPTION: GP.....	134

5.2.13	JTAG_CLOCK: JCK	136
5.2.14	REFRESH: RF	138
5.2.15	RESTART: RST	139
5.2.16	STATUS: STS	140
5.3	RAM Monitor Commands	141
5.3.1	RAM_R: RR	141
5.3.2	RAM_W: RW	143
Section 6 SH7047F E10A Emulator Specifications		145
6.1	Overview of the SH7047F E10A Emulator	145
6.2	Pin Arrangement of the Hitachi-UDI Port Connector	147
6.3	User System Interface Circuit	150
6.4	Differences between the SH7047F and the Emulator	156
6.5	The SH7047F E10A Emulator Functions	158
6.5.1	Emulator Driver Selection	158
6.5.2	Hardware Break Functions	158
6.5.3	Notes on Setting the [Breakpoint] Dialog Box	161
6.5.4	Notes on Using the JTAG Clock (TCK)	161
6.5.5	AUD Functions	162
6.5.6	Notes on Displaying the [Trace] Window	164
6.5.7	Notes on HDI	164

Figures

Figure 1.1	System Configuration with the Emulator (PCMCIA Card Emulator Used).....	1
Figure 1.2	System Configuration with the Emulator (PCI Card Emulator Used).....	2
Figure 2.1	Emulator Preparation Flow Chart.....	7
Figure 2.2	Inserting the PCMCIA Card Emulator in the Host Computer's Slot.....	11
Figure 2.3	Inserting the PCI Card Emulator in the Host Computer's Slot.....	11
Figure 2.4	Connecting the User System Interface Cable to the User System when the 14-pin Straight Type Connector is Used	13
Figure 2.5	User System Example	14
Figure 2.6	[Start] Menu	15
Figure 2.7	[Select Session] Dialog Box	16
Figure 2.8	[E10A Driver Details] Dialog Box.....	17
Figure 2.9	Dialog Box of the RESET Signal Input Request Message.....	18
Figure 2.10	[System Clock] Dialog Box.....	18
Figure 2.11	[HDI] Status Bar.....	18
Figure 2.12	[JTAG Connector Disconnected] Dialog Box.....	19
Figure 2.13	[Can not find /RESET signal] Dialog Box	19
Figure 2.14	[Check the connection] Dialog Box	19
Figure 2.15	[COMMUNICATION TIMEOUT ERROR] Dialog Box	20
Figure 2.16	[INVALID ASERAM FIRMWARE!] Dialog Box	20
Figure 2.17	[Error JTAG boot] Dialog Box.....	20
Figure 2.18	[Unable to restore the previous driver settings] Dialog Box	20
Figure 2.19	[Exit HDI] Dialog Box	22
Figure 2.20	[Save session] Dialog Box.....	22
Figure 3.1	[Start] Menu	27
Figure 3.2	[HDI] Window	28
Figure 3.3	[Configuration] Dialog Box	30
Figure 3.4	[Load Program] Dialog Box.....	31
Figure 3.5	[HDI] Dialog Box.....	31
Figure 3.6	[Open] Dialog Box	32
Figure 3.7	[Source] Window (Displaying the Source Program).....	33
Figure 3.8	[Source] Window (Setting a Software Breakpoint).....	34
Figure 3.9	[Registers] Window.....	35
Figure 3.10	[Register] Dialog Box (PC).....	36
Figure 3.11	[Go] Button	37
Figure 3.12	[Source] Window (Break Status).....	37
Figure 3.13	[System Status] Window	38
Figure 3.14	[Breakpoints] Window	39
Figure 3.15	[Open Memory Window] Dialog Box.....	40
Figure 3.16	[Memory] Window.....	40
Figure 3.17	[Instant Watch] Dialog Box.....	41
Figure 3.18	[Watch] Window (Displaying the Array).....	42

Figure 3.19	[Add Watch] Dialog Box.....	42
Figure 3.20	[Watch] Window (Displaying a Variable).....	43
Figure 3.21	[Watch] Window (Displaying Array Elements).....	43
Figure 3.22	[Step In] Button.....	44
Figure 3.23	[Source] Window (Step In).....	45
Figure 3.24	[Step Out] Button.....	46
Figure 3.25	[HDI] Window (Step Out).....	46
Figure 3.26	[HDI] Window (Step In → Step In).....	47
Figure 3.27	[Source] Window (Before Step Over Execution).....	48
Figure 3.28	[Step Over] Button.....	48
Figure 3.29	[HDI] Window (Step Over).....	49
Figure 3.30	[Go] Button.....	50
Figure 3.31	[Stop] Button.....	50
Figure 3.32	[Locals] Window.....	51
Figure 3.33	[Breakpoints] Window (Before Software Breakpoint Setting).....	52
Figure 3.34	[Point] Page ([Break] Dialog Box).....	53
Figure 3.35	[Break Point] Dialog Box.....	54
Figure 3.36	[Point] Page ([Break] Dialog Box) (After Software Breakpoint Setting).....	55
Figure 3.37	[Breakpoints] Window (Software Breakpoint Setting).....	56
Figure 3.38	[Source] Window at Execution Stop (Software Break).....	57
Figure 3.39	Displayed Contents of the [System Status] Window (Software Break).....	57
Figure 3.40	[Breakpoints] Window (Before Hardware Break Condition Setting).....	58
Figure 3.41	[Condition] Page ([Break] Dialog Box).....	59
Figure 3.42	[Address] Page ([Break Condition 1] Dialog Box).....	60
Figure 3.43	[Bus State] Page ([Break Condition 1] Dialog Box).....	61
Figure 3.44	[Break] Dialog Box (After Hardware Break Condition Setting).....	62
Figure 3.45	[Breakpoints] Window ([Break Condition 1] Setting).....	63
Figure 3.46	[Source] Window at Execution Stop (Break Condition 1).....	64
Figure 3.47	Displayed Contents of the [System Status] Window (Break Condition 1).....	65
Figure 3.48	[Configuration] Dialog Box (Sequential Break Setting).....	66
Figure 3.49	[Breakpoints] Window (After Sequential Break Condition Setting).....	67
Figure 3.50	[Source] Window at Execution Stop (Sequential Break).....	68
Figure 3.51	Displayed Contents of the [System Status] Window (Sequential Break).....	69
Figure 3.52	[Trace mode] Window.....	71
Figure 3.53	[Trace] Window in the SH7047F E10A Emulator.....	72
Figure 3.54	[Source] Window (Software Breakpoint Setting).....	73
Figure 3.55	[Stack Trace] Window.....	74
Figure 4.1	[Configuration] Dialog Box.....	80
Figure 4.2	[General] Page ([Configuration] Dialog Box).....	82
Figure 4.3	Warning Message Box.....	85
Figure 4.4	[E10A Driver Details] Dialog Box.....	86
Figure 4.5	[Breakpoints] Window.....	87
Figure 4.6	[Break] Dialog Box.....	90

Figure 4.7	[Point] Page ([Break] Dialog Box).....	92
Figure 4.8	[Condition] Page ([Break] Dialog Box)	94
Figure 4.9	[Break Point] Dialog Box	96
Figure 4.10	[Break Condition] Dialog Box	98
Figure 4.11	[Address] Page ([Break Condition 1] Dialog Box)	102
Figure 4.12	[Data] Page ([Break Condition 1] Dialog Box).....	104
Figure 4.13	[Bus State] Page ([Break Condition] Dialog Box)	106
Figure 4.14	[Count] Page ([Break Condition] Dialog Box).....	108
Figure 4.15	[Trace] Window	109
Figure 4.16	[Trace mode] Page ([Trace Acquisition] Dialog Box)	112
Figure 4.17	[System Status] Window	113
Figure 6.1	Pin Arrangement of the Hitachi-UDI Port Connector (14 Pins).....	147
Figure 6.2	Pin Arrangement of the Hitachi-UDI Port Connector (36 Pins).....	148
Figure 6.3	Circuit between User Interface Connector (36 Pins) and SH7047F	149
Figure 6.4	User System Interface Circuit (HS7047KCM01H).....	150
Figure 6.5	User System Interface Circuit of the Hitachi-UDI Pin (HS7047KCM02H)	151
Figure 6.6	User System Interface Circuit of the AUD Pin (HS7047KCM02H).....	152
Figure 6.7	User System Interface Circuit (HS7047KCI01H).....	153
Figure 6.8	User System Interface Circuit of the Hitachi-UDI Pin (HS7047KCI02H).....	154
Figure 6.9	User System Interface Circuit of the AUD Pin (HS7047KCI02H)	155

Tables

Table 1.1	Environmental Conditions.....	4
Table 1.2	Operating Environments.....	5
Table 2.1	Recommended Hitachi-UDI Port Connector.....	12
Table 2.2	Contents of the CD-R Directories	24
Table 3.1	Tutorial Program: Configuration and Parts	25
Table 3.2	Step Option.....	44
Table 3.3	Sequential Break Conditions	66
Table 3.4	AUD Trace Functions	70
Table 4.1	HDI Window Menus and Related Manual Entries.....	77
Table 4.2	[Configuration] Dialog Box Page.....	81
Table 4.3	[General] Page Options	83
Table 4.4	Options for the [E10A Driver Details] Dialog Box.....	86
Table 4.5	[Breakpoints] Window Display Items	88
Table 4.6	[Breakpoints] Window Pop-up Menu Operation.....	89
Table 4.7	[Break] Dialog Box Pages.....	91
Table 4.8	[Point] Page Options	93
Table 4.9	[Condition] Page Options.....	95
Table 4.10	[Address] Page Options.....	97
Table 4.11	Setting Conditions in [Break Condition] Dialog Boxes	100
Table 4.12	[Break Condition] Dialog Box Pages	101
Table 4.13	[Address] Page Options (BREAK_CONDITION1 to 4).....	103
Table 4.14	[Address] Page Options (BREAK_CONDITION R)	103
Table 4.15	Address Options	103
Table 4.16	[Data] Page Options	105
Table 4.17	[Bus State] Page Options.....	107
Table 4.18	[Count] Page Options	108
Table 4.19	[Trace] Window Display Items	110
Table 4.20	[Trace Acquisition] Dialog Box Page Options.....	111
Table 4.21	[System Status] Window Display Items	114
Table 5.1	E10A HDI Commands	117
Table 5.2	AUD_MODE Command Parameter.....	118
Table 5.3	AUD_TRACE Command Parameter	120
Table 5.4	BREAKCONDITION_CLEAR Command Parameter.....	122
Table 5.5	BREAKCONDITION_DISPLAY Command Parameter	123
Table 5.6	BREAKCONDITION_ENABLE Command Parameters	124
Table 5.7	BREAKCONDITION_SET Command Parameters	126
Table 5.8	BREAKPOINT Command Parameters	129
Table 5.9	BREAKPOINT_CLEAR Command Parameters	130
Table 5.10	BREAKPOINT_DISPLAY Command Parameter	131
Table 5.11	BREAKPOINT_ENABLE Command Parameters.....	132
Table 5.12	DEVICE_TYPE Command Parameter.....	133

Table 5.13	GO_OPTION Command Parameter.....	134
Table 5.14	JTAG_CLOCK Command Parameter.....	136
Table 5.15	REFRESH Command Parameter.....	138
Table 5.16	RESTART Command Parameter.....	139
Table 5.17	STATUS Command Parameter.....	140
Table 5.19	RAM_R Command Parameter.....	141
Table 5.20	RAM_W Command Parameter.....	143
Table 6.1	Components of the Emulator (HS7047KCM01H, HS7047KCM02H, HS7047KCI01H, or HS7047KCI02H).....	146
Table 6.2	Register Initial Values at Emulator Power-On.....	156
Table 6.3	Type Name and Driver.....	158
Table 6.4	Hardware Break Condition Specification Items.....	158
Table 6.5	Conditions Set in the [Break Condition] Dialog Box.....	159
Table 6.6	Conditions Set with the BREAKCONDITION_SET Command.....	159
Table 6.7	AUD Functions.....	162
Table 6.8	Type Name and AUD Function.....	163
Table 6.9	Watchdog Timer Registers.....	168

Section 1 Overview

The E10A emulator (hereafter referred to as the emulator) is a software and hardware development support tool for application systems using the microprocessor developed by Hitachi, Ltd.

The PCMCIA card emulator or PCI card emulator (hereafter referred to as the card emulator), which is the main unit of the emulator, is connected, through the Hitachi-UDI (user debug interface) port*, to the user system. The user system can be debugged under the conditions similar to the actual application conditions. The emulator enables debugging anywhere indoors or out. The host computer for controlling the emulator must be an IBM PC compatible machine with a PCMCIA type II or PCI slot.

Figures 1.1 and 1.2 show the system configuration using the emulator.

Note: The Hitachi-UDI is an interface compatible with the Joint Test Action Group (JTAG) specifications.

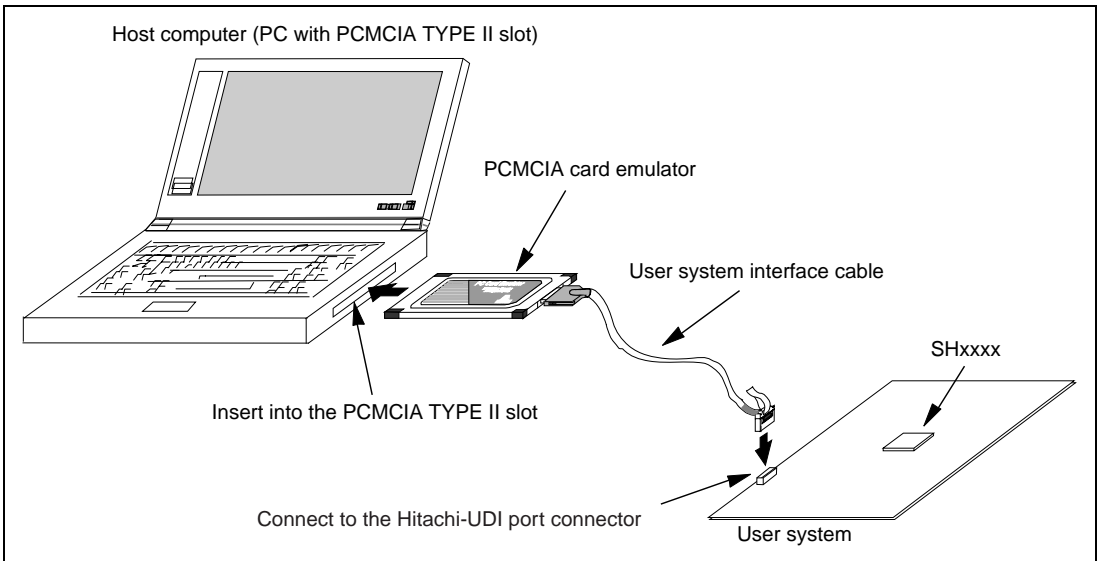


Figure 1.1 System Configuration with the Emulator (PCMCIA Card Emulator Used)

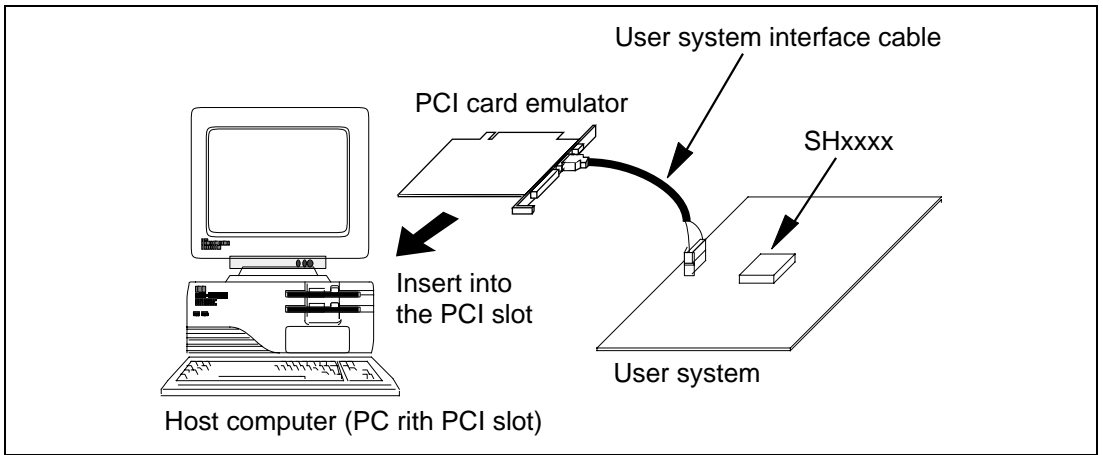


Figure 1.2 System Configuration with the Emulator (PCI Card Emulator Used)

The emulator provides the following features:

- Excellent cost-performance card emulator
Compactness and low price are implemented using the PCMCIA interface or the PCI interface.
- Realtime emulation
Realtime emulation of the user system is enabled at the maximum operating frequency of the CPU.
- Excellent operability
Using the Hitachi Debugging Interface (HDI) on the Microsoft® Windows® 98, Microsoft® Windows® ME, Microsoft® Windows® 2000, and Microsoft® Windows NT® operating systems enables user program debugging using a pointing device such as a mouse. The HDI enables high-speed downloading of load module files.
- Various debugging functions
Various break and trace functions enable efficient debugging. Breakpoints and break conditions can be set by the specific window, trace information can be displayed on a window, and command-line functions can be used.
- Memory access during emulation
During emulation, the memory contents can be read and modified.
- Debugging of the user system in the final development stage
The user system can be debugged under conditions similar to the actual application conditions.
- Compact debugging environment
When the card emulator specific to the PCMCIA interface is used, a laptop computer can be used as a host computer, creating a debugging environment in any place.
- AUD trace function*
The AUD trace function enables realtime trace.

Note: The AUD is an abbreviation of the Advanced User Debugger. Support for the AUD varies with the product.

1.1 Warnings

CAUTION

READ the following warnings before using the emulator product. Incorrect operation will damage the user system and the emulator product. The USER PROGRAM will be LOST.

1. Check all components against the component list after unpacking the emulator.
2. Never place heavy objects on the casing.
3. Protect the emulator from excessive impacts and stresses. For details, refer to section 1.2, Environmental Conditions.
4. Do not insert the emulator into any slot (PCMCIA TYPE II slot or PCI slot) other than the specified one.
5. When moving the host computer or user system, take care not to vibrate or damage it.
6. After connecting the cable, check that it is connected correctly. For details, refer to section 2, Preparation before Use.
7. Supply power to the connected equipment after connecting all cables. Cables must not be connected or removed while the power is on.

1.2 Environmental Conditions

CAUTION

Observe the conditions listed in tables 1.1 and 1.2 when using the emulator. Failure to do so will cause illegal operation in the user system, the emulator product, and the user program.

Table 1.1 Environmental Conditions

Item	Specifications
Temperature	Operating: +10°C to +35°C Storage: -10°C to +50°C
Humidity	Operating: 35% RH to 80% RH, no condensation Storage: 35% RH to 80% RH, no condensation
Vibration	Operating: 2.45 m/s ² max. Storage: 4.9 m/s ² max. Transportation: 14.7 m/s ² max.
Ambient gases	No corrosive gases may be present

Table 1.2 lists the acceptable operating environments.

Table 1.2 Operating Environments

Item	Description
Host computer	Built-in Pentium or higher-performance CPU (200 MHz or higher recommended); IBM PC or compatible machine with the PCMCIA TYPE II slot or the PCI slot.
OS	Windows® 98, Windows® ME, Windows® 2000, or Windows NT®
Minimum memory capacity	32 Mbytes or more (double of the load module size recommended)
Hard-disk capacity	Installation disk capacity: 10 Mbytes or more. (Prepare an area at least double the memory capacity (four-times or more recommended) as the swap area.)
Pointing device such as mouse	Connectable to the host computer; compatible with Windows® 98, Windows® ME, Windows® 2000, and Windows NT®.
Power voltage	5.0 ± 0.25 V
Current consumption	HSxxxxKCM01H: 110 mA (max) HSxxxxKCM02H: 230 mA (max) HSxxxxKCI01H: 340 mA (max) HSxxxxKCI02H: 600 mA (max)
CD-ROM drive	Required to install the HDI for the emulator or refer to the emulator user's manual.

1.3 Components

Check all the components unpacking. For details on the E10A emulator components, refer to table 6.1, Components of the Emulator. If the components are not complete, contact a Hitachi sales agency.

Section 2 Preparation before Use

2.1 Emulator Preparation



READ the reference sections shaded in figure 2.1 before using the emulator product. Incorrect operation will damage the user system and the emulator product. The USER PROGRAM will be LOST.

Unpack the emulator and prepare it for use as follows:

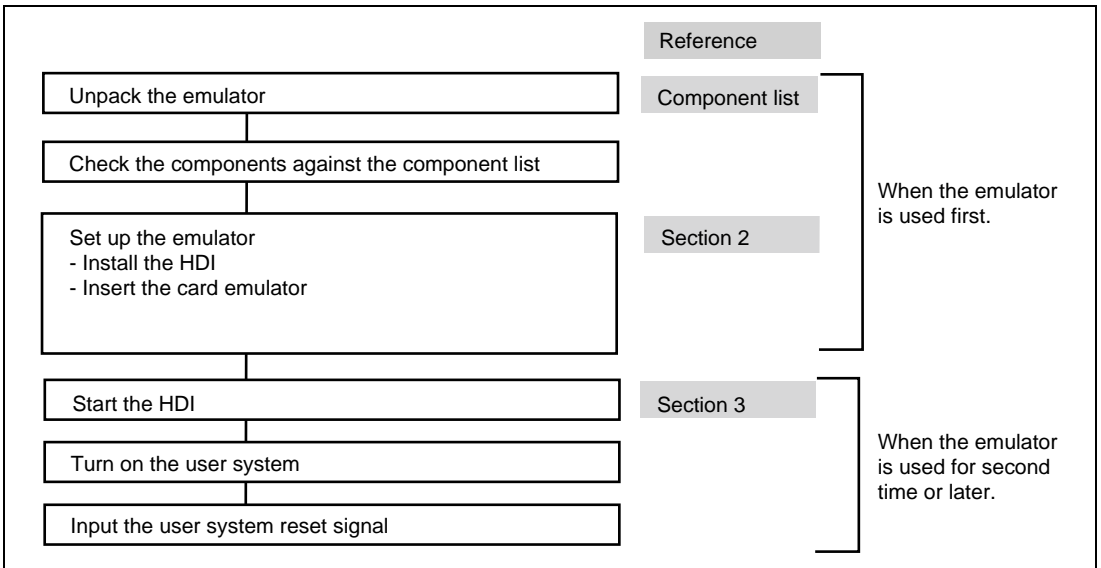


Figure 2.1 Emulator Preparation Flow Chart

2.2 HDI Installation

When the CD-R is inserted in the host computer's CD-ROM drive, the HDI installation wizard is automatically activated (holding the Shift key down while the CD-R is inserted cancels this automatic activation). To run the installation wizard when it has not been automatically activated, execute Setup.exe from the root directory of the CD-R.

Follow the cues given by the installation wizard to install the HDI.

Since hardware settings are also made during installation, the installation procedure differs according to the operating system or interface (PCI or PCMCIA) being used. Follow the installation steps carefully according to the environment you are using.

2.2.1 Installing under Windows®98 and Windows®ME Operating Systems

(1) When the emulator is a PCI card:

1. Install the HDI (when the component type has to be selected during installation, be sure to select [PCI Card Driver]).
2. Shut the operating system down and turn off the power to the host computer.
3. Insert the PCI-card emulator in a slot on the host computer. Refer to section 2.3, Connecting the Card Emulator to the Host Computer.
4. Restart the host computer. The hardware is now recognized and the driver is automatically installed.*

(2) When the emulator is a PCMCIA card:

1. Install the HDI (when the component type has to be selected during installation, be sure to select [PC Card Driver (PCMCIA)]).
2. Insert the PCMCIA-card emulator in the host computer's slot. Refer to section 2.3, Connecting the Card Emulator to the Host Computer.
3. The hardware is now recognized and the driver is automatically installed.*

Note: When [Add New Hardware Wizard] is displayed, select the [Search for the best driver for your device. (Recommended)] radio button and then the [Specify a location] check box to select the path to be searched for drivers. The location must be specified according to the emulator type, as indicated below:

When using the PCI-card emulator: <Drive>:\DRIVERS\PCI\95

When using the PCMCIA-card emulator: <Drive>:\DRIVERS\PCMCIA\95

(<Drive> is the CD-ROM drive name.)

2.2.2 Installing under Windows NT® 4.0 Operating System

(1) When the emulator is a PCI card:

1. Shut the operating system down and turn off the power to the host computer.
2. Insert the PCI-card emulator in a slot on the host computer. Refer to section 2.3, Connecting the Card Emulator to the Host Computer.
3. Start the host computer and log-on with an administrator-level user name.
4. Install the HDI. (For a component, be sure to select [PCI Card Driver]. There is a check box for selecting the type name of the product under the [PCI Card Driver] component. Select the appropriate type name. If the correct name is not selected, the correct driver will not be installed, and the emulator will not operate.)
5. Restart the host computer.

(2) When the emulator is a PCMCIA card:

1. Shut the operating system down and turn off the power to the host computer.
2. Insert the PCMCIA-card emulator in the host computer's slot. Refer to section 2.3, Connecting the Card Emulator to the Host Computer.
3. Start the host computer and log-on with an administrator-level user name.
4. During HDI installation, the setting value should be checked beforehand because inquiries are made about the resource used by the PCMCIA-card emulator.
Start the [Start] menu -> [Programs] -> [Administrative Tools (Common)] -> [Windows NT Diagnostics], check the status of the IRQ, I/O port, and memory from the resource panel, and determine the setting values that do not conflict with other devices. (The following resources are used: IRQ: one channel, I/O port: H'F byte, and memory: H'4000 byte.)
5. Install the HDI. (For a component, be sure to select [PC Card Driver (PCMCIA)]. There is a check box for selecting the type name of each product under the [PC Card Driver (PCMCIA)] component. Select the appropriate type name. If the correct name is not selected, the correct driver will not be installed and the emulator will not operate.)
6. Restart the host computer.

Notes: The driver that has been selected in the [Drivers] component starts after the host computer is initiated. If the host computer is initiated with the card disconnected or with the incorrect driver installed, the driver cannot initiate and the service control manager informs the system of an error. This, however, is not a problem.

2.2.3 Installing under Windows® 2000 Operating System

(1) When the emulator is a PCI card:

1. Log-on with an administrator-level user name.
2. Install the HDI. (When a component is selected, be sure to select [PCI Card Driver].)
3. Shut the operating system down and turn off the power to the host computer.
4. Insert the PCI-card emulator in a slot on the host computer. Refer to section 2.3, Connecting the Card Emulator to the Host Computer.
5. Restart the host computer and log-on with an administrator-level user name. The hardware is now recognized and the driver is automatically installed.*

(2) When the emulator is a PCMCIA card:

1. Log-on with an administrator-level user name.
2. Install the HDI. (When a component is selected, be sure to select [PC Card Driver (PCMCIA)].)
3. Insert the PCMCIA-card emulator in the host computer's slot. Refer to section 2.3, Connecting the Card Emulator to the Host Computer.
4. The hardware is now recognized and the driver is automatically installed.*

Note: When [Found New Hardware Wizard] is displayed, select the [Search for a suitable driver for my device (recommended).] radio button and then the [Specify a location] check box to select the path to be searched for drivers. The location must be specified according to the emulator type, as indicated below:

When using the PCI-card emulator: <Drive>:\DRIVERS\PCI\2000

When using the PCMCIA-card emulator: <Drive>:\DRIVERS\PCMCIA\2000

(<Drive> is the CD-ROM drive name.)

2.3 Connecting the Card Emulator to the Host Computer

Insert the card emulator, according to its type, in a PCMCIA TYPE II slot or PCI slot on the host computer (figures 2.2 and 2.3).

Note: When using Windows®98, Windows®ME, or Windows®2000, be sure to install the HDI before putting the card emulator in place.

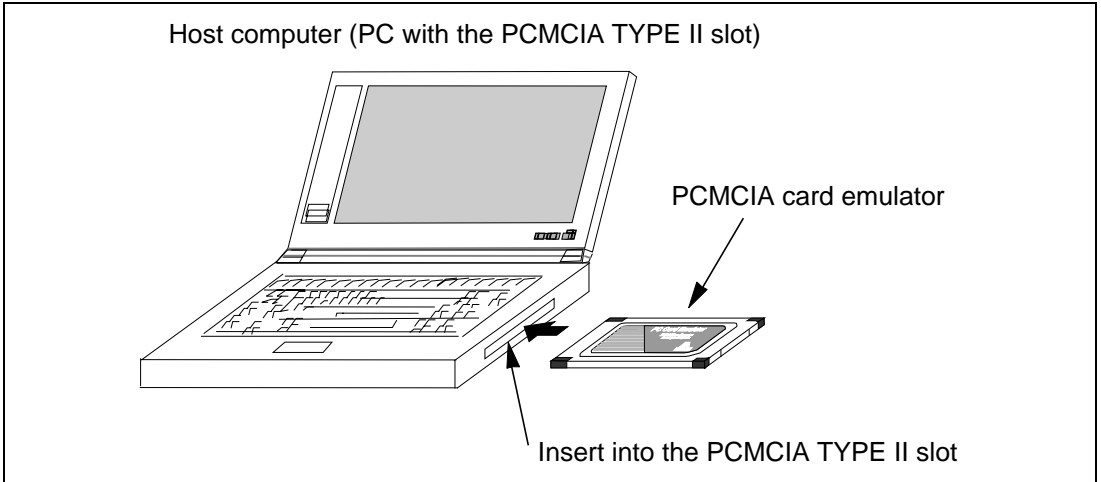


Figure 2.2 Inserting the PCMCIA Card Emulator in the Host Computer's Slot

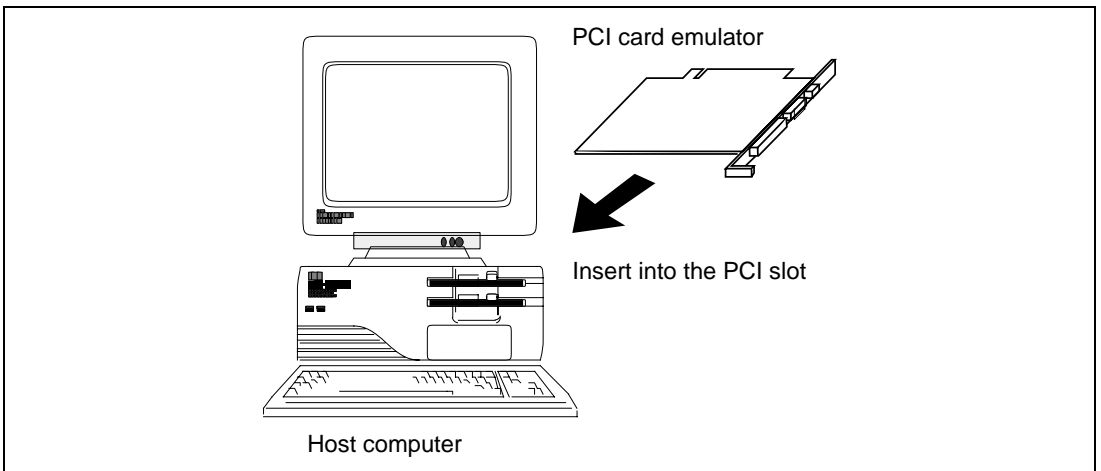


Figure 2.3 Inserting the PCI Card Emulator in the Host Computer's Slot

Use the procedure, described in section 2.4, to connect the emulator to the user system with the user system interface cable, or to disconnect them when moving the emulator or the user system.

WARNING

When inserting the PCI-card emulator, note the following. Failure to do so will damage the host computer.

1. Turn off the host computer.
2. Insert the emulator into the PCI slot in parallel.
3. Screw in the emulator after checking the connector and cable positions.

2.4 Connecting the Card Emulator to the User System

(1) The Hitachi-UDI port connector must be installed to the user system. Table 2.1 shows the recommended Hitachi-UDI port connector for the emulator.

Table 2.1 Recommended Hitachi-UDI Port Connector

Connector	Type Number	Manufacturer	Specifications
14-pin connector	2514-6002	Minnesota Mining & Manufacturing Ltd.	14-pin straight type
36-pin connector	DX10M-36S	Hirose Electric Co., Ltd.	Screw type
	DX10M-36SE, DX10GM-36SE		Lock-pin type

Note: When the 14-pin connector is used, do not install any components within 3 mm of the Hitachi-UDI port connector.
When the 36-pin connector is used, do not connect other signal lines to the Hitachi-UDI port connector.

- (2) Note that the TDO signal of the user system interface cable connector must be connected to the TDI pin of the Hitachi-UDI port connector and the TDI signal of the user system interface cable connector must be connected to the TDO pin of the Hitachi-UDI port connector. Section 6.2 shows the pin arrangement of the Hitachi-UDI port connector.
- (3) Figure 2.4 shows how to connect the user system interface cable to the user system when the 14-pin straight type connector is used. Connect the ground line of the cable to the user system ground. The end of the ground line has a hole having a diameter of 3 mm, and therefore, when the ground line is screwed to the user system, the screw diameter must be 3 mm.

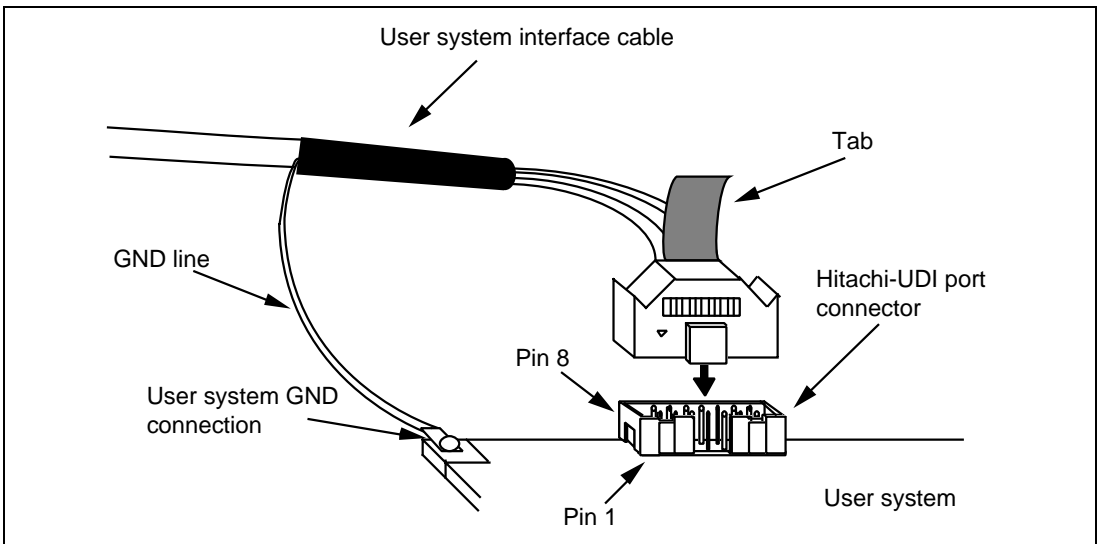
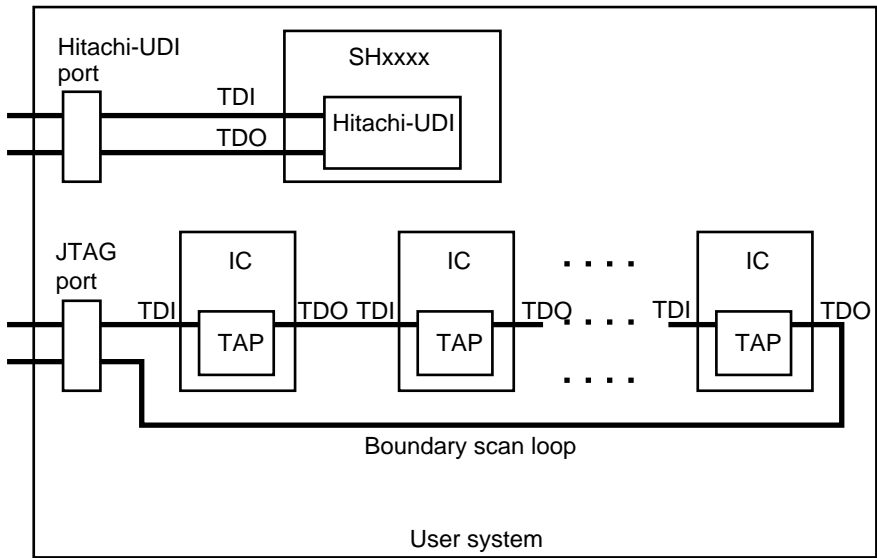


Figure 2.4 Connecting the User System Interface Cable to the User System when the 14-pin Straight Type Connector is Used

- Notes:**
1. To connect the signals output from the Hitachi-UDI port connector, refer to the device pin alignment.
 2. To remove the user system interface cable from the user system, pull the tab on the connector upward.
 3. The range of frequencies that the Hitachi-UDI operates at is different according to the devices used. For details, refer to section 6.5.4, Notes on Using the JTAG Clock (TCK).
 4. Connect the Hitachi-UDI signals from the Hitachi-UDI port connector directly to the device.
 5. When developing user systems, do not connect the TDI and TDO signals of the device to the boundary scan loop, or separate them by using a switch (figure 2.5).



TDI: Test data input
 TDO: Test data output
 TAP: Test access port

Figure 2.5 User System Example

2.5 System Check

When the HDI program is executed, use the procedure below to check that the emulator is operating correctly.

1. Check that the emulator card is inserted in the host computer's slot.
2. Connect the user system interface cable to the connector of the card emulator.
3. Connect the user system interface cable to the Hitachi-UDI port connector.
4. Supply power to the host computer and select [HDI for E10A SHxxxx] -> [Hitachi Debugging Interface] from the [Start] menu.

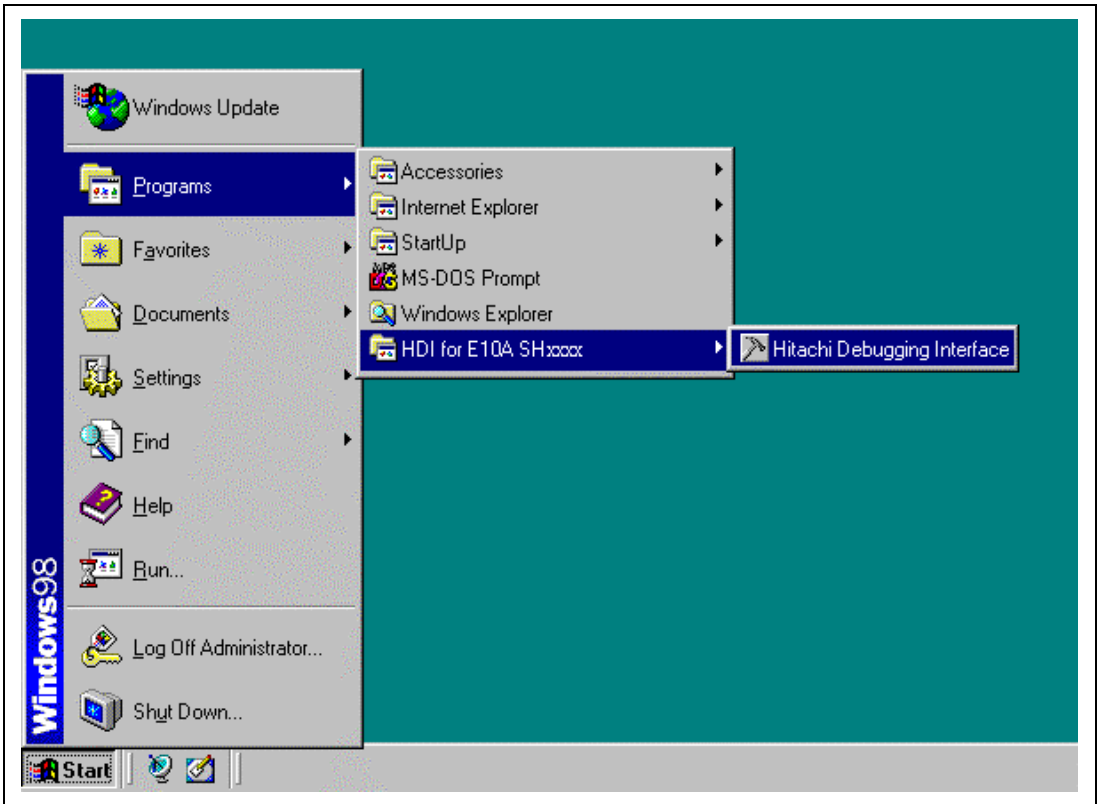


Figure 2.6 [Start] Menu

5. Select the setting to be used.

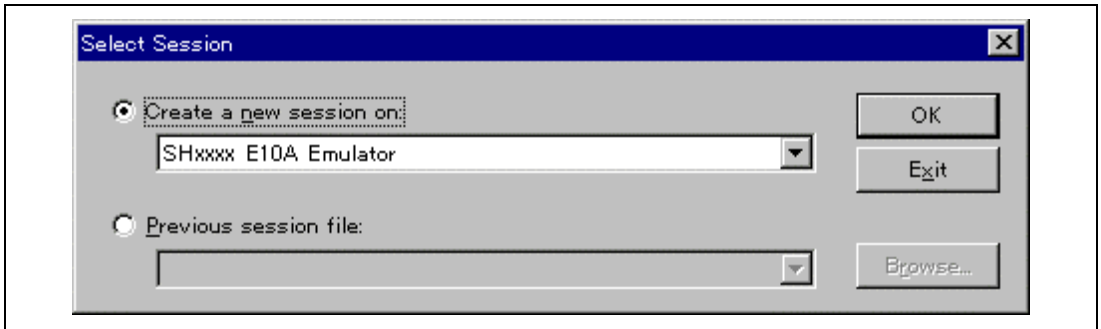


Figure 2.7 [Select Session] Dialog Box

- The [E10A Driver Details] dialog box is displayed. With the [Driver] combo box, select the driver to connect the HDI with the emulator. [Interface] displays the interface name of the PC interface board to be connected, and [Channel] displays the interface to which the board is connected. Once the driver is selected in the [E10A Driver Details] dialog box, this dialog box is not displayed when the HDI is run next time. (This procedure will not be executed by target devices.)

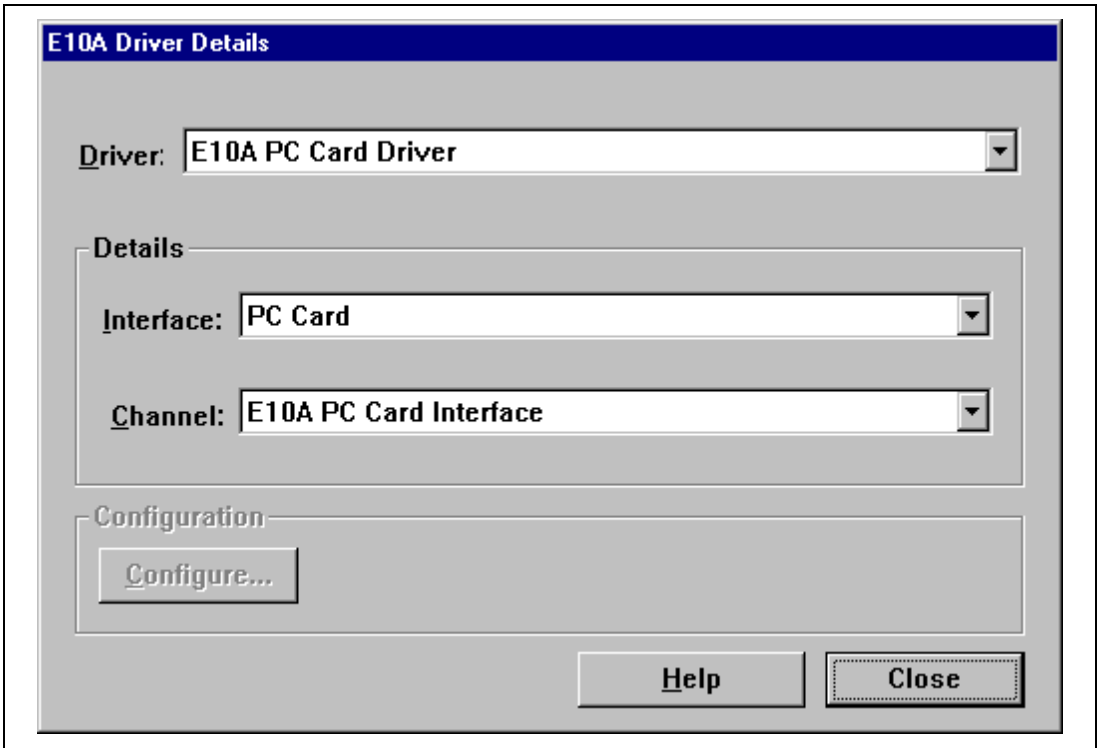


Figure 2.8 [E10A Driver Details] Dialog Box

- With the [Driver] combo box, select the driver to connect the HDI with the emulator.
- [Interface] displays the interface name of the card emulator to be connected, and [Channel] displays the interface to which the board is connected.
 - [Driver] combo box: Select [E10A PC Card Driver] to use the PCMCIA card emulator. Select [E10A PCI Card Driver] to use the PCI card emulator. For details, refer to table 6.3 in section 6.5.1, Emulator Driver Selection.
 - [Interface] combo box: Select [PC Card] to use the PCMCIA card emulator. [PCI] is displayed to use the PCI card emulator. (If the driver is not installed, the [PC Card] or [PCI] is not displayed.)
- Click the [Close] button.

- The HDI window is displayed, and the dialog box is displayed as shown in figures 2.9 and 2.10. In the [System Clock] dialog box, enter the system clock value that is determined by the input clock and the clock mode.



Figure 2.9 Dialog Box of the RESET Signal Input Request Message

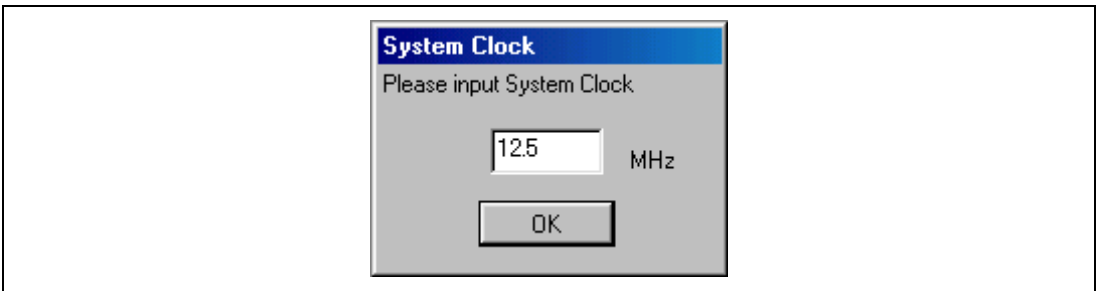


Figure 2.10 [System Clock] Dialog Box

- Power on the user system.
- Input the reset signal from the user system, and click the [OK] button.
- When "Link Up" is displayed on the status bar, the HDI initiation is completed.

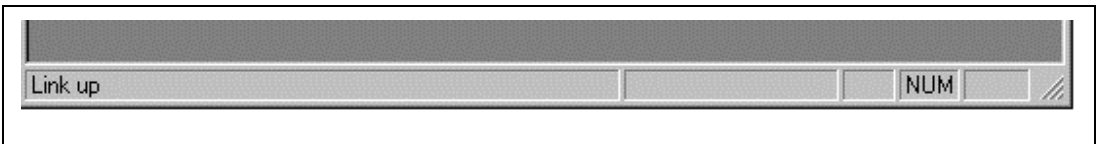


Figure 2.11 [HDI] Status Bar

Notes: 1. When the HDI is not linked up even if the above procedure has been executed, the driver will not be set correctly. Install drivers provided under the /DRIVERS directory in the CD-R according to the screen instructions. For instructions on how to check the driver has been correctly set, refer to the OS manual for your host computer or the following URL:

<http://www.hitachisemiconductor.com/sic/jsp/japan/eng/products/mpumcu/tool/qa/e10aqa.html>

2. If the user system interface cable is disconnected to the Hitachi-UDI port connector on the user system during user program execution, the following dialog box will be displayed.



Figure 2.12 [JTAG Connector Disconnected] Dialog Box

3. If the emulator is not initiated, the following dialog boxes shown in figures 2.13 through 2.17 will be displayed.
 - (a) If the following dialog box is displayed, the power of the user system may not be input or the RESET signal may not be input to the device. Check the input circuits for the power of the user system and the reset pin.

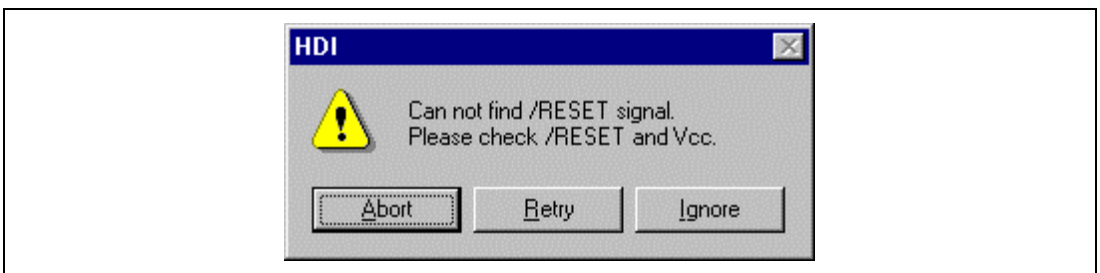


Figure 2.13 [Can not find /RESET signal] Dialog Box

- (b) If the following dialog box is displayed, check that the Hitachi-UDI port connector on the user system is correctly connected.

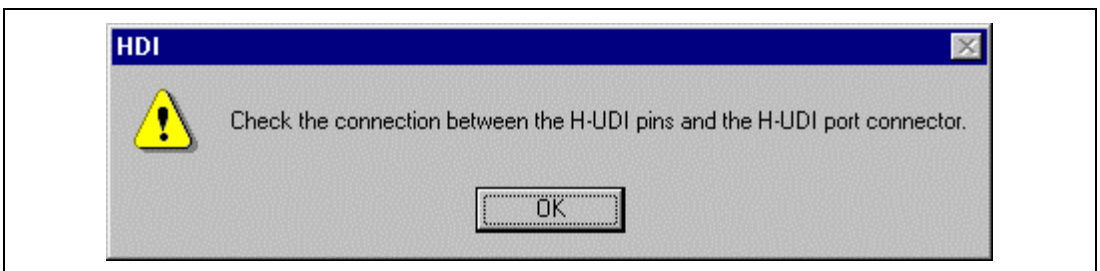


Figure 2.14 [Check the connection] Dialog Box

- (c) If the following dialog box is displayed, the device may not correctly operate. Check if there are reasons for illegal device operation.

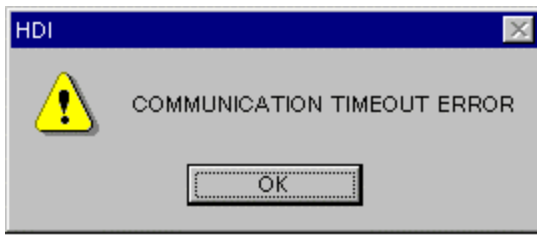


Figure 2.15 [COMMUNICATION TIMEOUT ERROR] Dialog Box

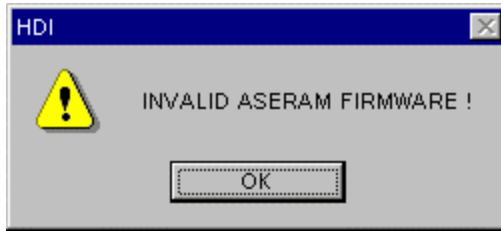


Figure 2.16 [INVALID ASERAM FIRMWARE!] Dialog Box



Figure 2.17 [Error JTAG boot] Dialog Box

4. If the driver is not correctly connected, the following dialog box will be displayed.

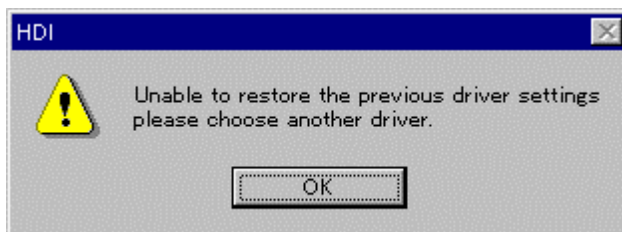


Figure 2.18 [Unable to restore the previous driver settings] Dialog Box

The [E10A Driver Details] dialog box is displayed when the [OK] button is clicked. Select the correct driver. For details, refer to section 6.5.1, Emulator Driver Selection.

2.6 Ending the HDI

Exit the HDI by using the following procedure:

1. Select [Exit] from the [File] menu to end the HDI. When the [Exit HDI] dialog box is displayed, click the [Yes] button.

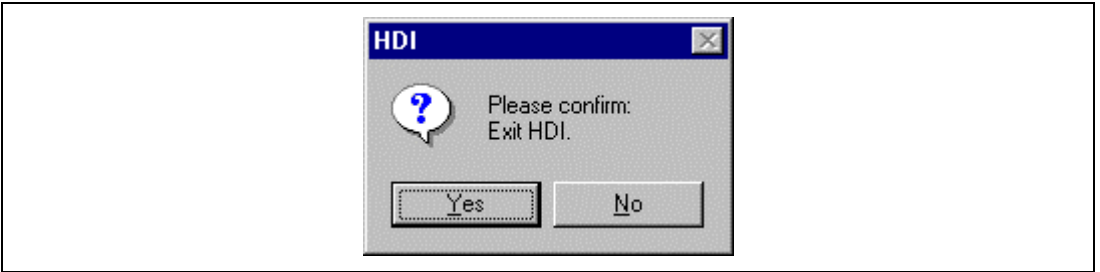


Figure 2.19 [Exit HDI] Dialog Box

2. Then, the [Save session] dialog box is displayed. If necessary, click the [Yes] button to save session. After saving session, the HDI ends. If not necessary, click the [No] button to end the HDI.

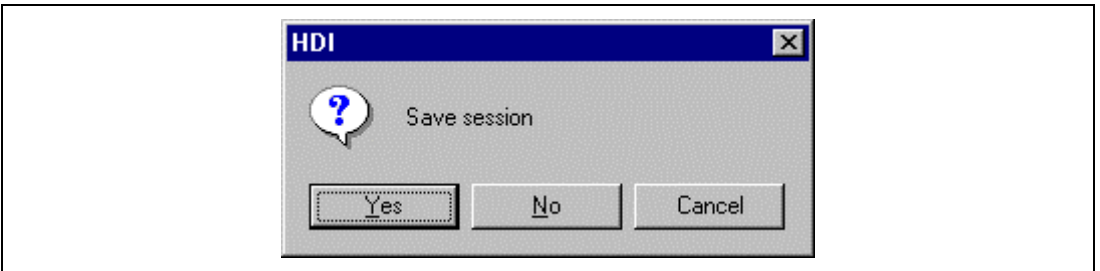


Figure 2.20 [Save session] Dialog Box

3. Turn the user system off.

2.7 Uninstalling the HDI

Follow this procedure to remove the installed HDI from the user's host computer.

1. Open [Add/Remove Programs Properties] from the control panel. Select the HDI program from the list and click the [Add/Remove...] button.
2. The setup program is executed again and the installed application can be changed, modified, or removed. When the application is to be uninstalled, select removal.

CAUTION

A shared file may be detected while the program is being removed. If another HDI may be using the shared file, do not remove the file. When Microsoft® Windows NT®4.0 operating system is used, the removal of the registry information on the driver may be asked. If other HDI may use the target driver, do not remove the registry information. If another HDI does not start up after the removal process, re-install that HDI.

2.8 CD-R

2.8.1 Configuration of the CD-R

The root directory of the CD-R contains a setup program for HDI installation. The folders contain the files and programs listed below.

Table 2.2 Contents of the CD-R Directories

Directory Name	Contents	Description
Dlls	Microsoft® runtime library	A runtime library for the HDI. The version is checked at installation and this library is copied to the hard disk as part of the installation process.
Drivers	E10A emulator driver	The E10A emulator drivers.
Help	Online help for the E10A emulator	An online help file. This is copied to the hard disk as part of the installation process.
Manual	E10A emulator manual	Precautions on Using the E10A Emulator and the user's manual. These are provided as PDF files.
Pdf_read	Adobe® Acrobat® Reader setup program	Adobe® Acrobat® Reader is an application for displaying, viewing, and printing PDF files.

2.9 Support

Information on the latest version of the HDI and other supporting information for the emulator can be found on the web site. Access the following URL:

<http://www.hitachisemiconductor.com/sic/jsp/japan/eng/products/mpumcu/tool/download/e10a.html>

Section 3 Tutorial

3.1 Introduction

The following describes the main functions of the HDI by using a tutorial program.

The tutorial program is based on the C program that sorts ten random data items in ascending or descending order. The tutorial program performs the following actions:

- The `main` function generates random data to be sorted.
- The `sort` function sorts the generated random data in ascending order.
- The `change` function then sorts the data in descending order.

The file `tutorial.c` contains source code for the tutorial program. The file `tutorial.abs` is a compiled load module in the Dwarf2 format.

Table 3.1 is a list of the parts of the tutorial program and an outline of their configuration on the hard disk.

Table 3.1 Tutorial Program: Configuration and Parts

Item	Contents
Workspace for HEW V1.2	[Installation directory]\tutorial\tutorial.hws
Load module	[Installation directory]\tutorial\tutorial\Debug\tutorial.abs
Main program (source file)	[Installation directory]\tutorial\tutorial\tutorial.c
Stack information file	[Installation directory]\tutorial\tutorial\Debug\tutorial.sni

- Notes:**
- 1. This program was created by using Hitachi Embedded Workshop (hereafter referred to as HEW) V1.2. Older versions of HEW will not open the workspace included with the package, so create a new workspace in such situations.**
 - 2. This program was compiled without optimization for the SH2 CPU. If recompiled with different settings, the addresses may differ from those given in this section.**
 - 3. tutorial.abs is a load module in the Dwarf2 format. If a load module is recreated in the Sysrof format, the amount of information displayed on the HDI screen during the program's execution will be reduced.**
 - 4. This section describes general usage examples for the emulator. For the specifications of particular products, refer to section 6 or the online help file.**

3.2 Running the HDI

To run the HDI, select the [HDI for E10A SHxxxx] -> [Hitachi Debugging Interface] from the [Start] menu.

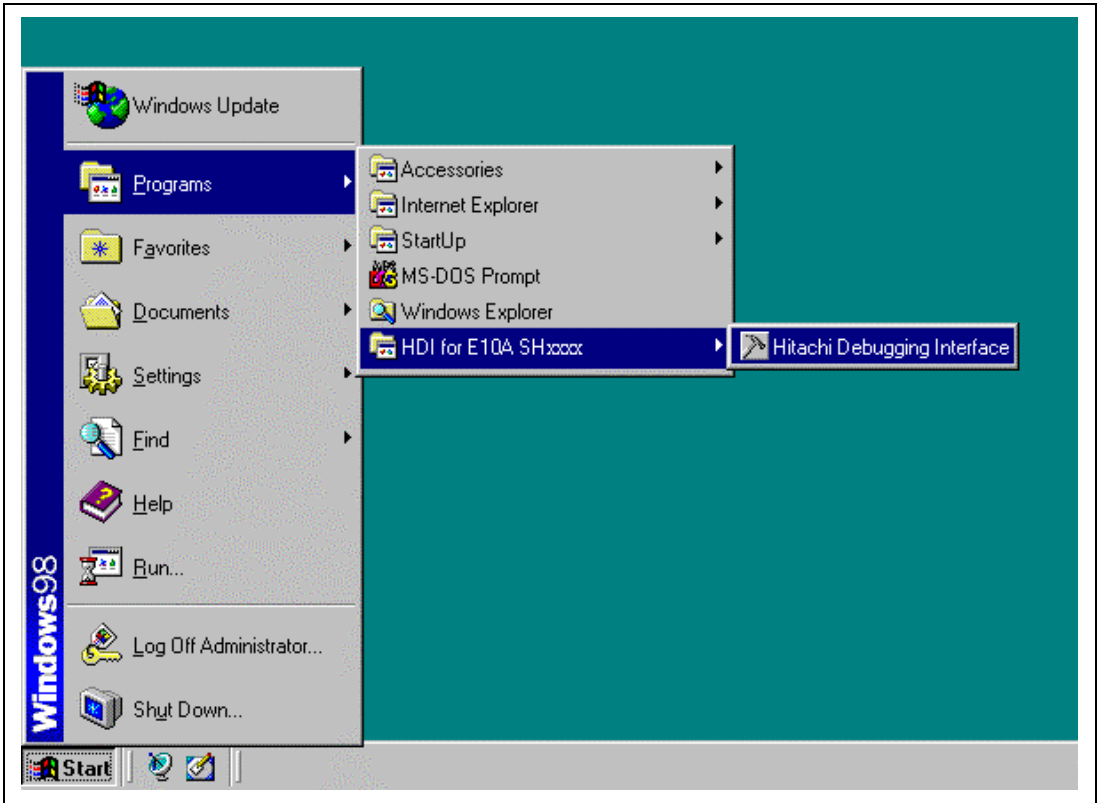


Figure 3.1 [Start] Menu

For the procedure of running the HDI, refer to section 2.5, System Check.

3.3 [HDI] Window

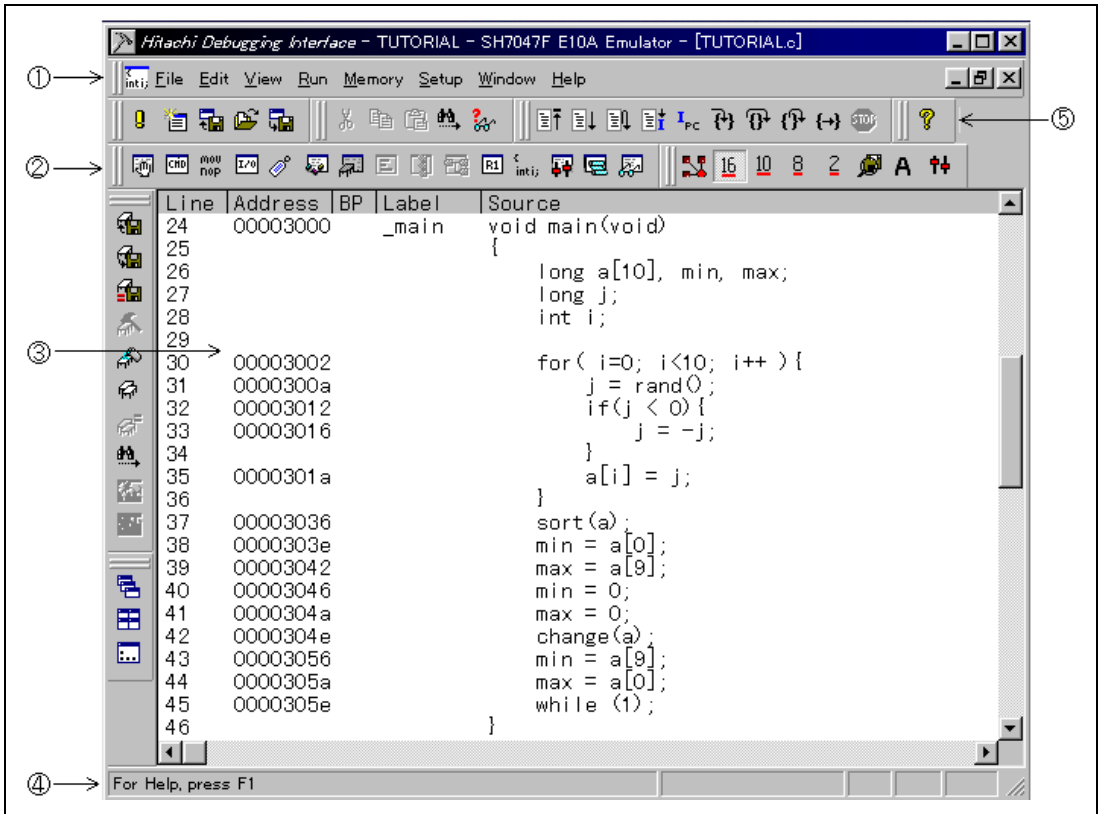


Figure 3.2 [HDI] Window

The key functions of the HDI are described in section 4, Descriptions of Windows. Numbers in figure 3.2 indicate the following:

1. Menu bar: Gives the user access to the HDI commands for using the HDI debugger.
2. Toolbar: Provides convenient buttons as shortcuts for the most frequently used menu commands.
3. Source window: Displays the source program being debugged.
4. Status bar: Displays the status of the emulator, and progress information about downloading.
5. [Help] button: Activates online help about any features of the HDI user interface.

3.4 Setting up the Emulator

The clocks which are used for data communications must be set up on the emulator before the program is downloaded.

- JTAG clock (TCK)

A communication clock for downloading data to the emulator except for acquiring AUD trace.

If its frequency is set too low, the speed of downloading will be lowered.

If its frequency is set too high, the upper limit for the device's TCK clock may be exceeded.

For details, refer to section 6.5.4, Notes on Using the JTAG Clock (TCK). The following is a description of the procedure used to set the clocks.

3.5 Setting the [Configuration] Dialog Box

- Select [Configure Platform...] from the [Setup] menu to set a communication clock. The [Configuration] dialog box is displayed.

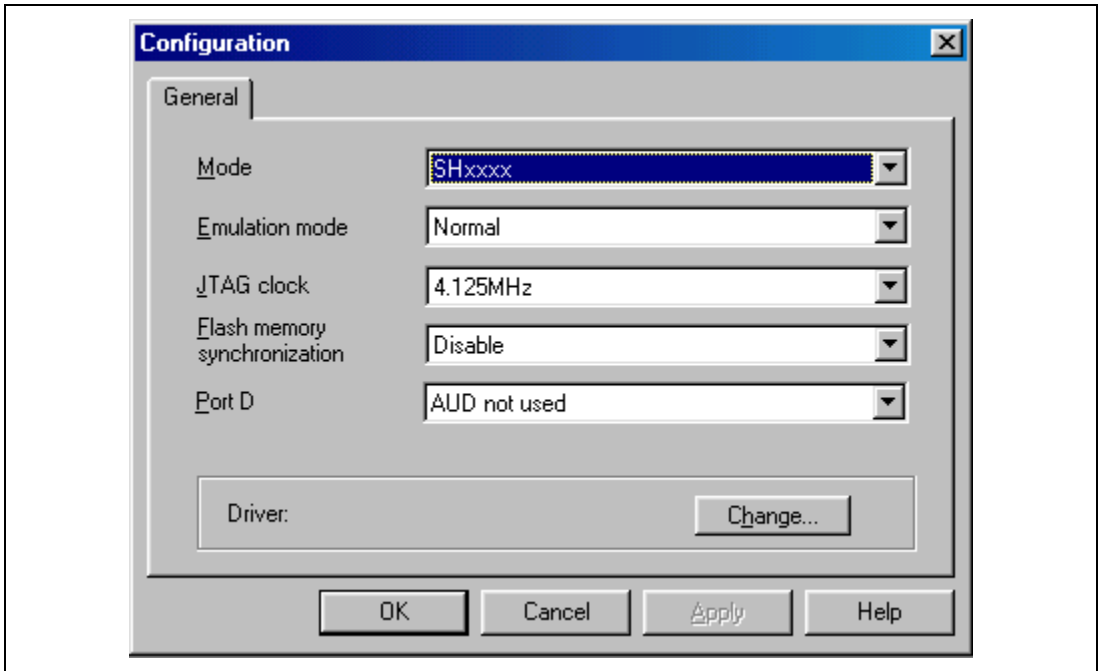


Figure 3.3 [Configuration] Dialog Box

- Set any value in the [JTAG clock] combo boxes. The clock also operates with the default value.

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

- Click the [OK] button to set a configuration.

3.6 Downloading the Tutorial Program

3.6.1 Downloading the Tutorial Program

Download the object program to be debugged.

- Select [Load Program...] from the [File] menu. The [Load Program] dialog box is displayed. Enter '[installation directory]\tutorial\tutorial\Debug\tutorial.abs' in the [File name] list box as shown in figure 3.4, then click the [Open] button.

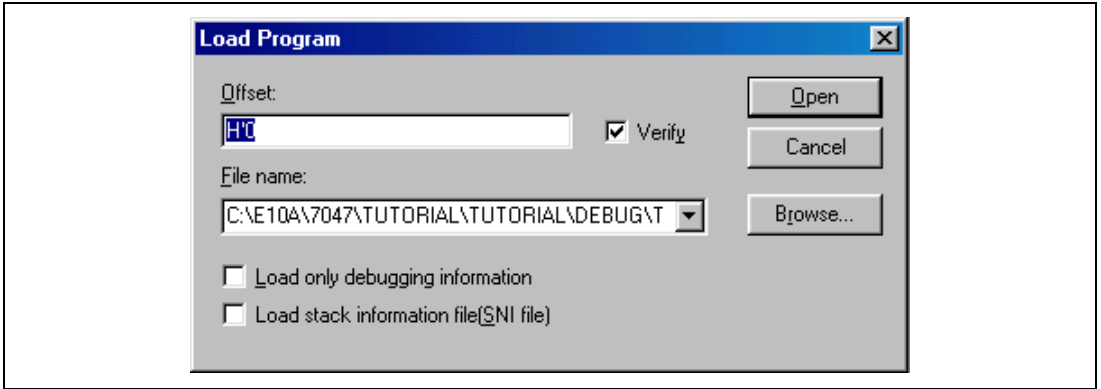


Figure 3.4 [Load Program] Dialog Box

- Notes:
1. When installing the emulator, if no directory is specified and the HEW is in use, the program is installed under "\Hew\hdi5\e10a". When the HEW is not in use, the program is installed under "\root directory\E10A".
 2. The SNI file is required to use the profiler function.
 3. The [Verify] check box is disabled in this product.

After the file has been loaded, the following dialog box displays information about the memory areas to which the program code has been transferred.

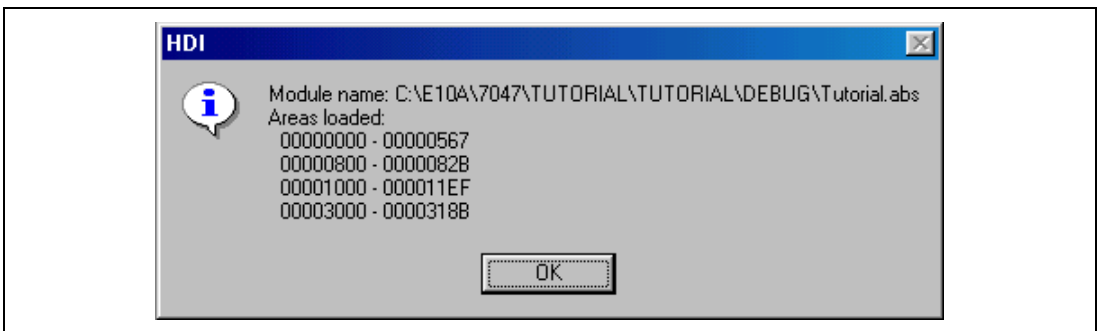


Figure 3.5 [HDI] Dialog Box

- Click the [OK] button to continue.

3.6.2 Displaying the Source Program

The HDI allows the user to debug a program on the source level.

- Select [Source...] from the [View] menu. The [Open] dialog box is displayed.
- Select the C source file that corresponds to the object file the user has loaded.

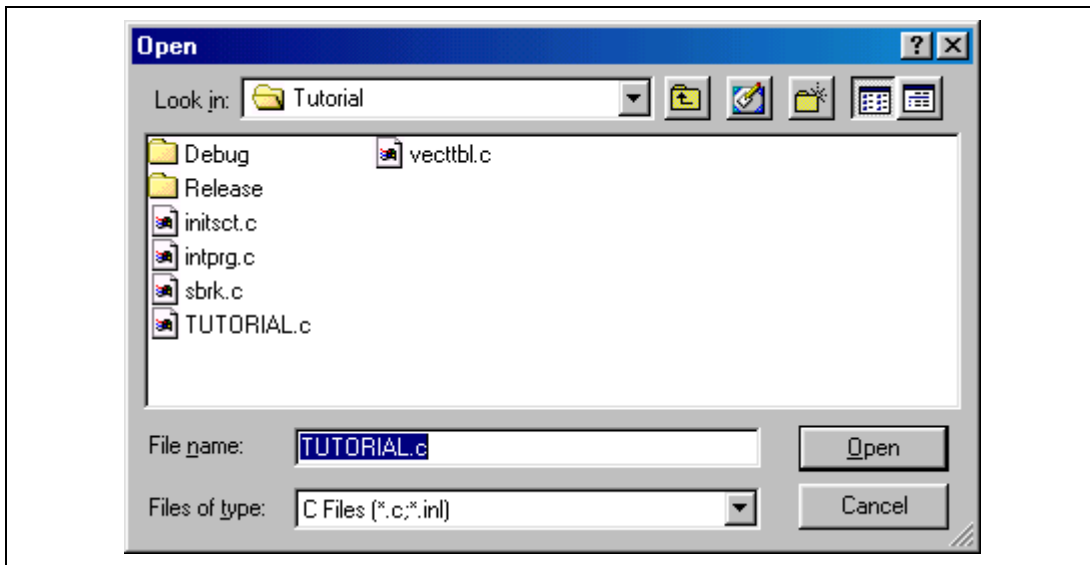


Figure 3.6 [Open] Dialog Box

- Select [tutorial.c] and click the [Open] button. The [Source] window is displayed.

Line	Address	BP	Label	Source
24	00003000		_main	void main(void)
25				{
26				long a[10], min, max;
27				long j;
28				int i;
29				
30	00003002			for(i=0; i<10; i++){
31	0000300a			j = rand();
32	00003012			if(j < 0){
33	00003016			j = -j;
34				}
35	0000301a			a[i] = j;
36				}
37	00003036			sort(a);
38	0000303e			min = a[0];
39	00003042			max = a[9];
40	00003046			min = 0;
41	0000304a			max = 0;
42	0000304e			change(a);
43	00003056			min = a[9];
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				
48	0000306c		_sort	void sort(long *a)
49				{
50				long t;
51				int i, j, k, gap;
52				

Figure 3.7 [Source] Window (Displaying the Source Program)

- If necessary, select the [Font] option from the [Customise] submenu on the [Setup] menu to select a font and size that are legible.

Initially the [Source] window shows the start of the main program, but the user can use the scroll bar to scroll through the program and look at the other statements.

3.7 Setting a Software Breakpoint

A breakpoint is one of simple debugging functions.

The [Source] window provides a very simple way of setting a software breakpoint at any point in a program. For example, to set a breakpoint at the `sort` function call:

- Select by double-clicking the [BP] column on the line containing the `sort` function call.

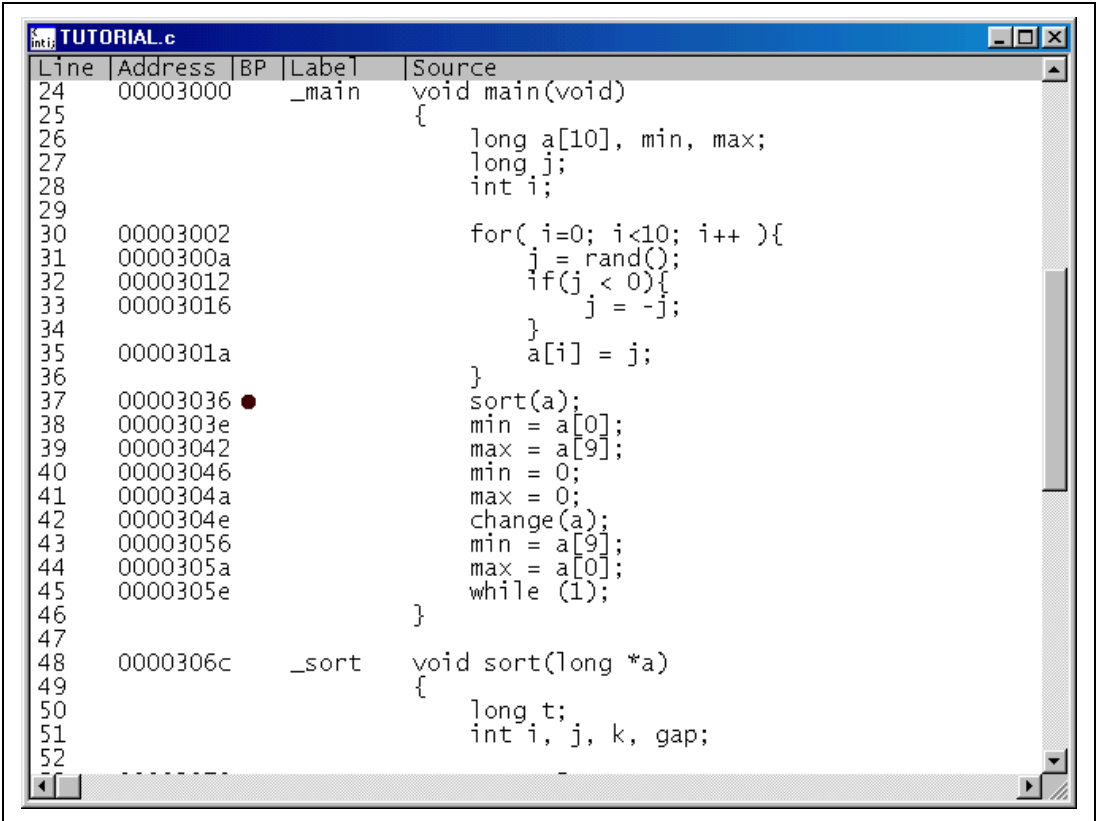


Figure 3.8 [Source] Window (Setting a Software Breakpoint)

The symbol • will appear on the line containing the `sort` function, and the word • Break will appear when the [BP] column is extended. This shows that a software breakpoint has been set.

Note: The software breakpoint cannot be set anywhere other than in the flash memory and RAM areas.

3.8 Setting Registers

Set values of the program counter and the stack pointer before executing the program.

- Select [Registers] from the [View] menu. The [Registers] window is displayed.

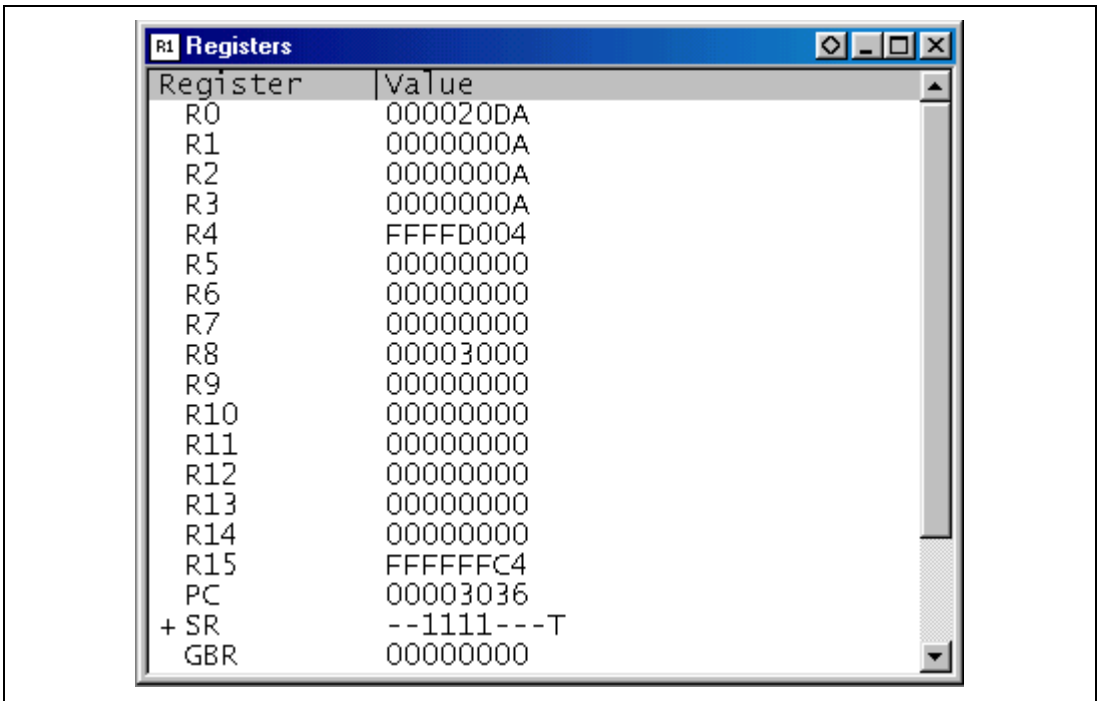


Figure 3.9 [Registers] Window

- To change the value of the program counter (PC), double-click the value area in the [Registers] window with the mouse. The following dialog box is then displayed, and the value can be changed. Set the program counter to H'00000800 in this tutorial program, and click the [OK] button.
- Move the mouse pointer on the value to be changed in the [PC] value area and enter the new value by the keyboard.

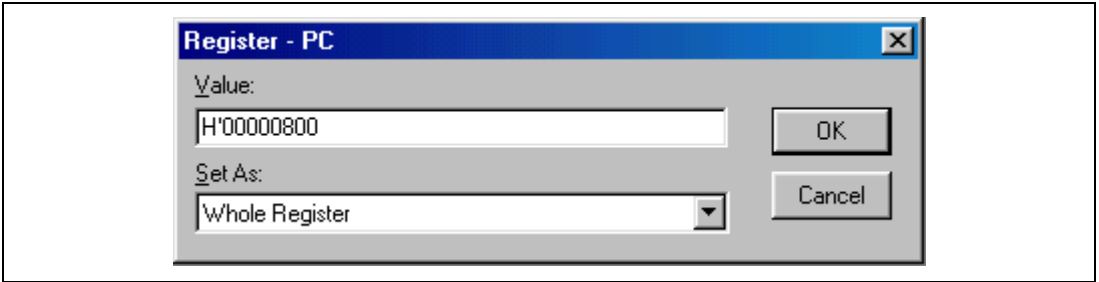


Figure 3.10 [Register] Dialog Box (PC)

- Change the value of the stack pointer (SP) in the same way. Set H'FFFFFFFC for the value of the stack pointer in this tutorial program.

3.9 Executing the Program

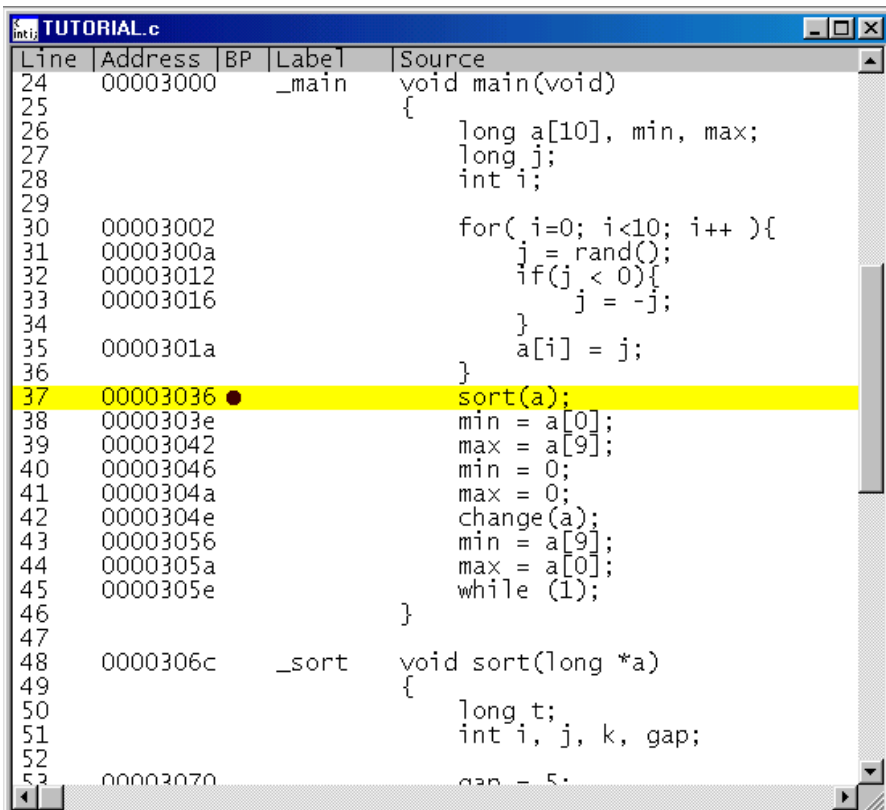
Execute the program as described in the following:

- To execute the program, select [Go] from the [Run] menu, or click the [Go] button on the toolbar.



Figure 3.11 [Go] Button

The program will be executed up to the breakpoint that has been inserted, and a statement will be highlighted in the [Source] window to show the position that the program has halted, with the message [Break=BREAKPOINT] in the status bar.



Line	Address	BP	Label	Source
24	00003000		_main	void main(void)
25				{
26				long a[10], min, max;
27				long j;
28				int i;
29				
30	00003002			for(i=0; i<10; i++){
31	0000300a			j = rand();
32	00003012			if(j < 0){
33	00003016			j = -j;
34				}
35	0000301a			a[i] = j;
36				}
37	00003036	●		sort(a);
38	0000303e			min = a[0];
39	00003042			max = a[9];
40	00003046			min = 0;
41	0000304a			max = 0;
42	0000304e			change(a);
43	00003056			min = a[9];
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				
48	0000306c		_sort	void sort(long *a)
49				{
50				long t;
51				int i, j, k, gap;
52				
53	00003070			gap = 5;

Figure 3.12 [Source] Window (Break Status)

The user can see the cause of the break that occurred last time in the [System Status] window.

- Select [Status] from the [View] menu. After the [System Status] window is displayed, open the [Platform] page, and check the status of Cause of last break.

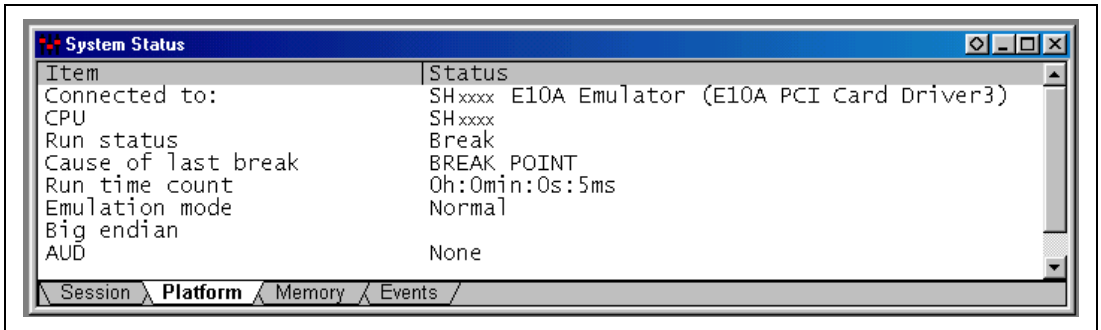


Figure 3.13 [System Status] Window

Note: The items that can be displayed in this window differ according to the product. For the items that can be displayed, refer to the online help.

3.10 Reviewing Breakpoints

The user can see all the breakpoints set in the program in the [Breakpoints] window.

- Select [Breakpoints] from the [View] menu.

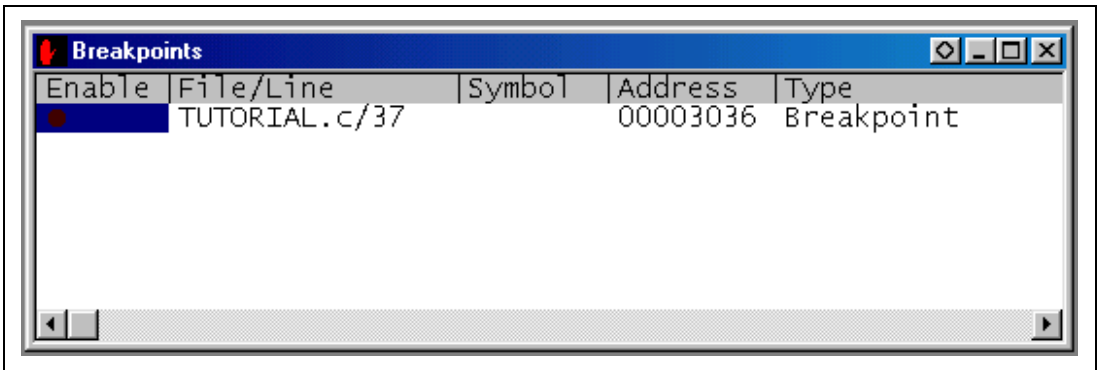


Figure 3.14 [Breakpoints] Window

The pop-up menu, opened by clicking the [Breakpoints] window with the right mouse button, also allows the user to set or change breakpoints, define new breakpoints, and delete, enable, or disable breakpoints.

3.11 Viewing Memory

When the label name is specified, the user can view the memory contents that the label has been registered in the [Memory] window. For example, to view the memory contents corresponding to the `_main` in word size:

- Select [Memory ...] from the [View] menu, enter `_main` in the [Address] edit box, and set word in the [Format] combo box.

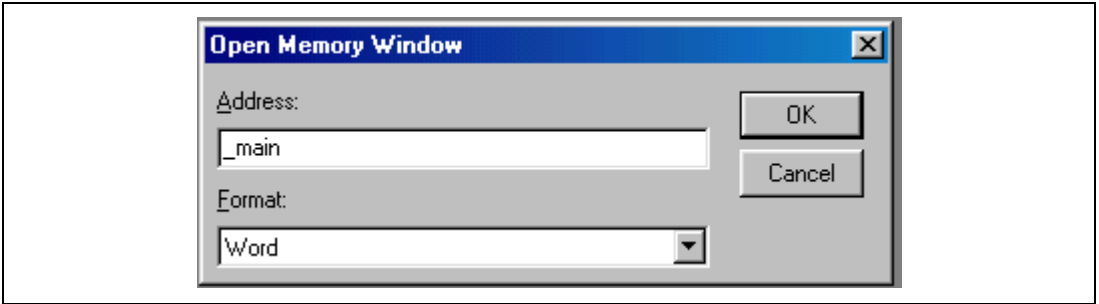


Figure 3.15 [Open Memory Window] Dialog Box

- Click the [OK] button. The [Memory] window showing the specified area of memory is displayed.

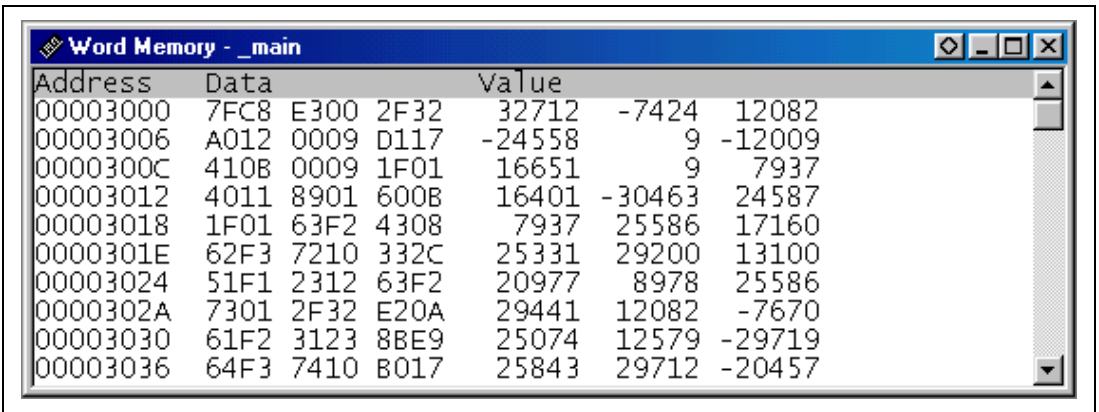


Figure 3.16 [Memory] Window

3.12 Watching Variables

As the user steps through a program, it is possible to watch that the values of variables used in the user program are changed. For example, set a watch on the long-type array `a` declared at the beginning of the program, by using the following procedure:

- Click the left of displayed array `a` in the [Source] window to position the cursor.
- Click the [Source] window with the right mouse button and select [Instant Watch...] from the pop-up menu.

The following dialog box will be displayed.

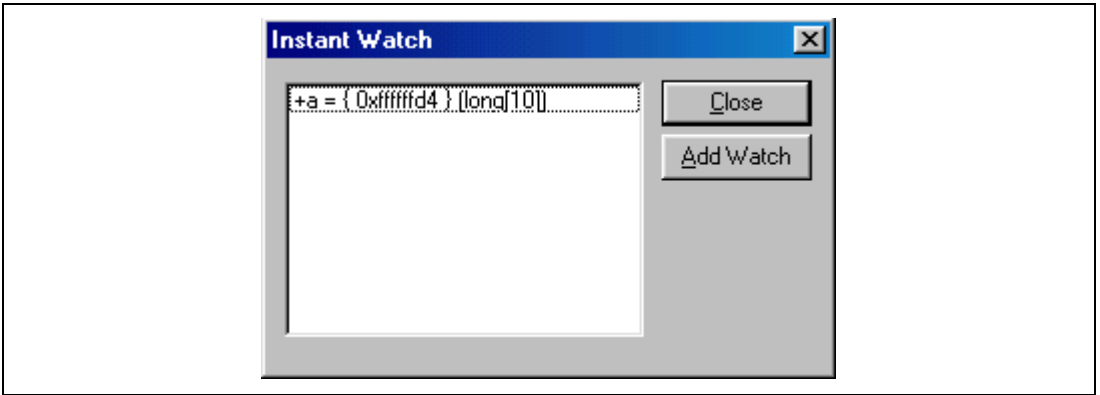


Figure 3.17 [Instant Watch] Dialog Box

- Click [Add Watch] button to add a variable to the [Watch] window.

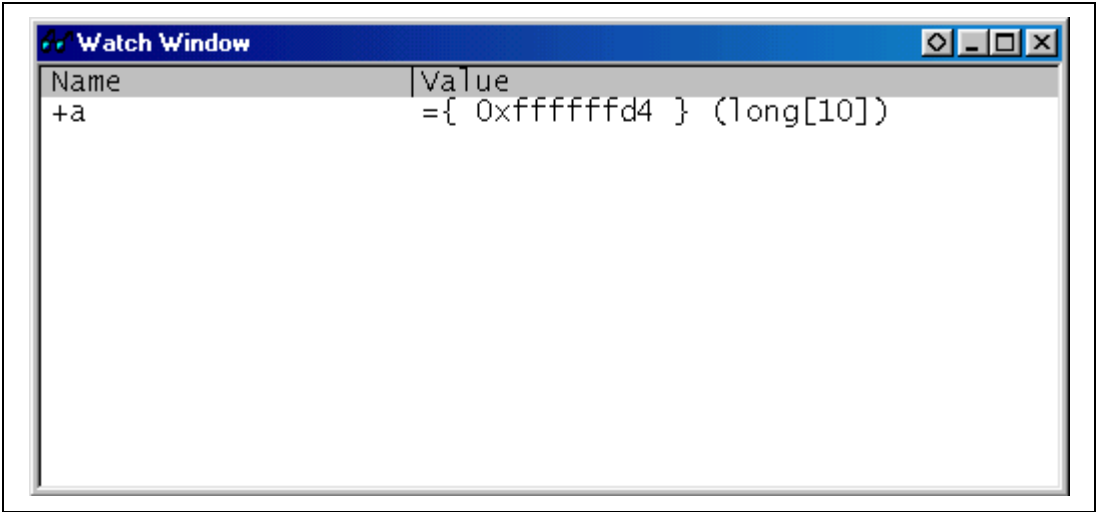


Figure 3.18 [Watch] Window (Displaying the Array)

The user can also add a variable to the [Watch] window by specifying its name.

- Click the [Watch] window with the right mouse button and select [Add Watch...] from the pop-up menu.

The following dialog box will be displayed.

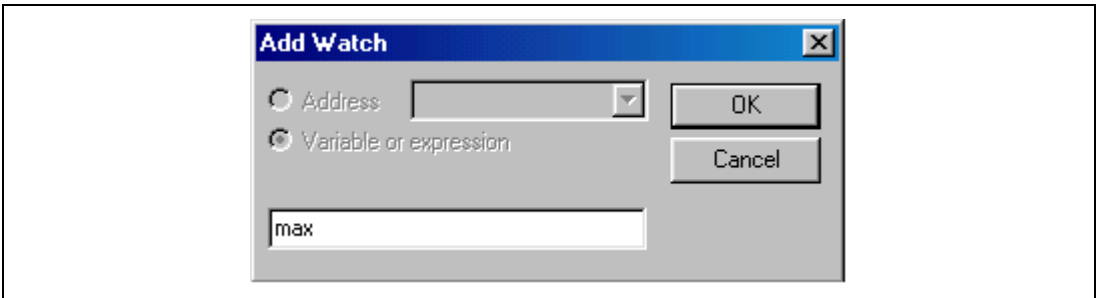


Figure 3.19 [Add Watch] Dialog Box

- Input variable **max** and click the [OK] button.

The [Watch] window will now also show the int-type variable max.

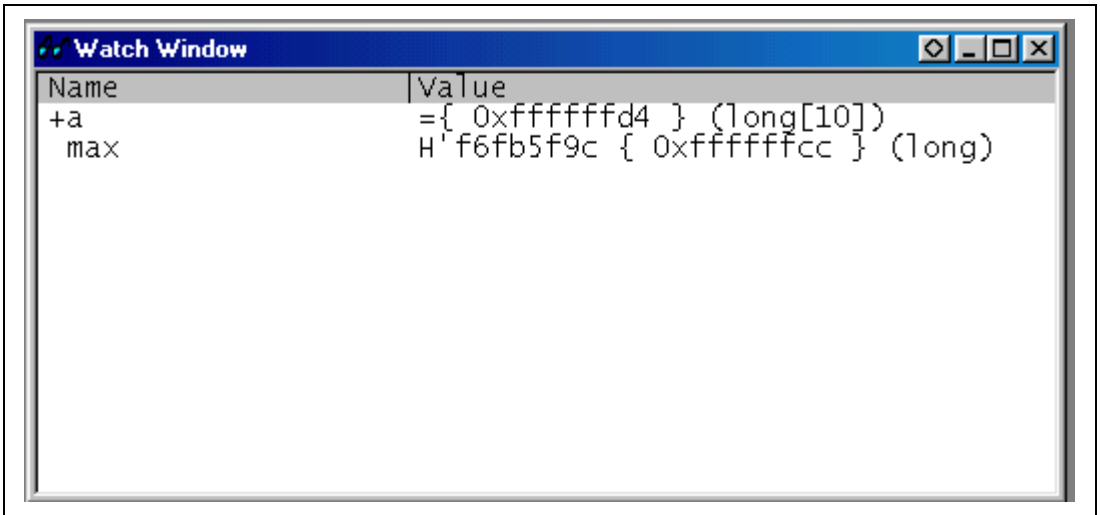


Figure 3.20 [Watch] Window (Displaying a Variable)

The user can double-click the + symbol to the left of any variable in the [Watch] window to watch the all elements in array a.

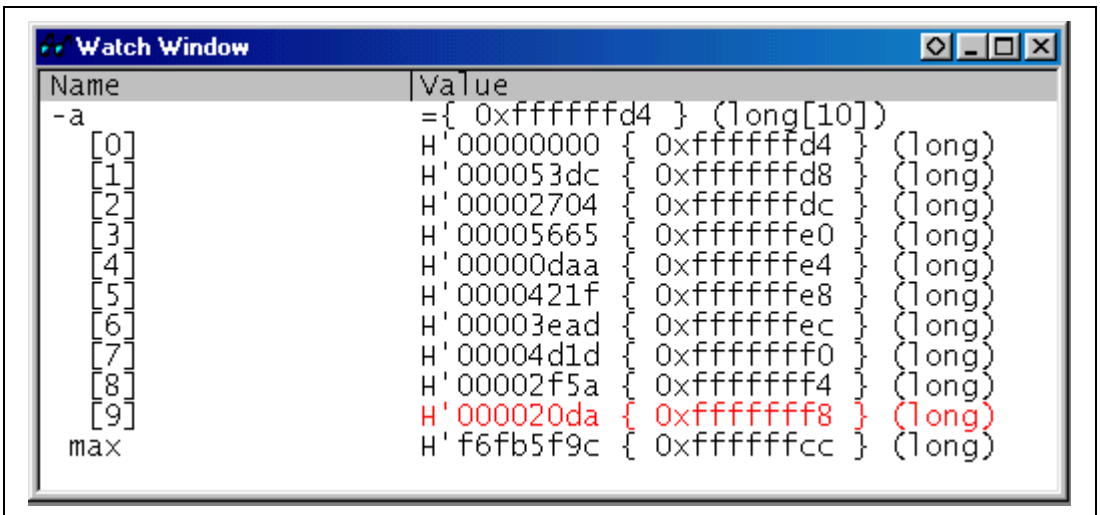


Figure 3.21 [Watch] Window (Displaying Array Elements)

3.13 Stepping Through a Program

The HDI provides a range of step menu commands that allow efficient program debugging.

Table 3.2 Step Option

Menu Command	Description
Step In	Executes each statement, including statements within functions.
Step Over	Executes a function call in a single step.
Step Out	Steps out of a function, and stops at the statement following the statement in the program that called the function.
Step...	Steps the specified times repeatedly at a specified rate.

3.13.1 Executing [Step In] Command

The [Step In] steps into the called function and stops at the first statement of the called function.

- To step through the `sort` function, select [Step In] from the [Run] menu, or click the [Step In] button in the toolbar.



Figure 3.22 [Step In] Button

The image shows a debugger window titled "TUTORIAL.c" with a table of source code. The table has columns for Line, Address, BP, Label, and Source. Line 48 is highlighted in yellow. A red dot is visible on line 37. The code includes a main function and a sort function.

Line	Address	BP	Label	Source
24	00003000		_main	void main(void)
25				{
26				long a[10], min, max;
27				long j;
28				int i;
29				
30	00003002			for(i=0; i<10; i++){
31	0000300a			j = rand();
32	00003012			if(j < 0){
33	00003016			j = -j;
34				}
35	0000301a			a[i] = j;
36				}
37	00003036			sort(a);
38	0000303e			min = a[0];
39	00003042			max = a[9];
40	00003046			min = 0;
41	0000304a			max = 0;
42	0000304e			change(a);
43	00003056			min = a[9];
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				
48	0000306c		_sort	void sort(long *a)
49				{
50				long t;
51				int i, j, k, gap;
52				
53	00003070			gap = 5;

Figure 3.23 [Source] Window (Step In)

- The highlighted line moves to the first statement of the `sort` function in the [Source] window.

3.13.2 Executing [Step Out] Command

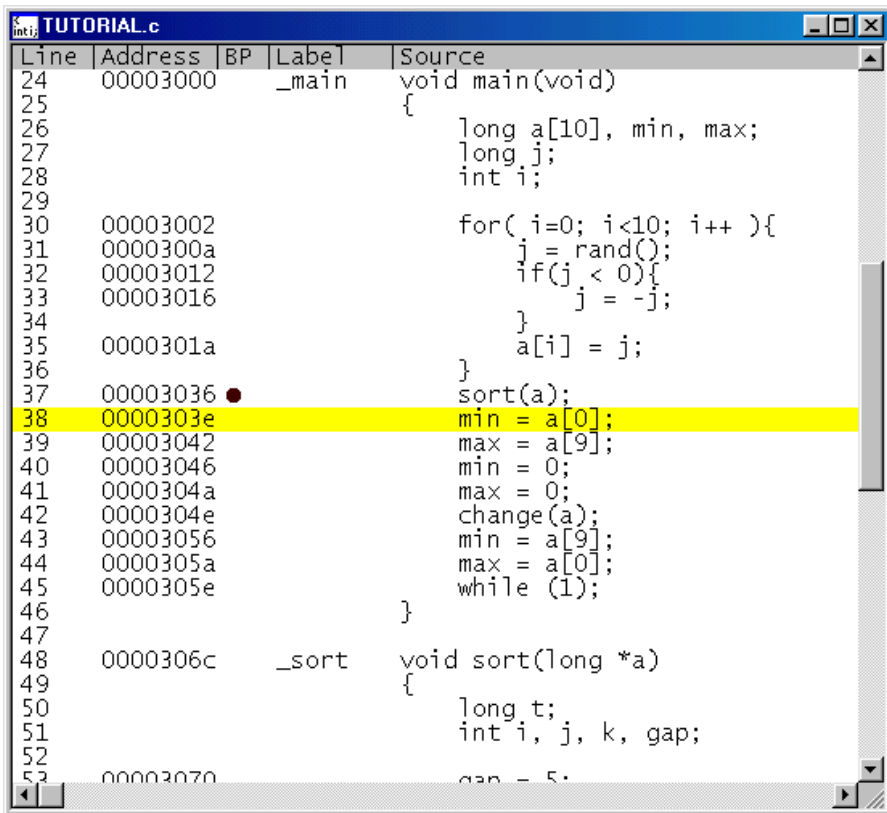
The [Step Out] steps out of the called function and stops at the next statement of the calling statement.

- To step out of the `sort` function, select [Step Out] from the [Run] menu, or click the [Step Out] button in the toolbar.

Note: It takes time to execute this function.



Figure 3.24 [Step Out] Button



```
int: TUTORIAL.c
Line Address BP Label Source
24 00003000 _main void main(void)
25 {
26     long a[10], min, max;
27     long j;
28     int i;
29
30     for( i=0; i<10; i++ ){
31         j = rand();
32         if(j < 0){
33             j = -j;
34         }
35         a[i] = j;
36     }
37     sort(a);
38     min = a[0];
39     max = a[9];
40     min = 0;
41     max = 0;
42     change(a);
43     min = a[9];
44     max = a[0];
45     while (1);
46 }
47
48 0000306c _sort void sort(long *a)
49 {
50     long t;
51     int i, j, k, gap;
52
53     gap = 5;
```

Figure 3.25 [HDI] Window (Step Out)

- The data of variable `a` displayed in the [Watch] window is sorted in ascending order.

- To execute two steps, use [Step In] twice.

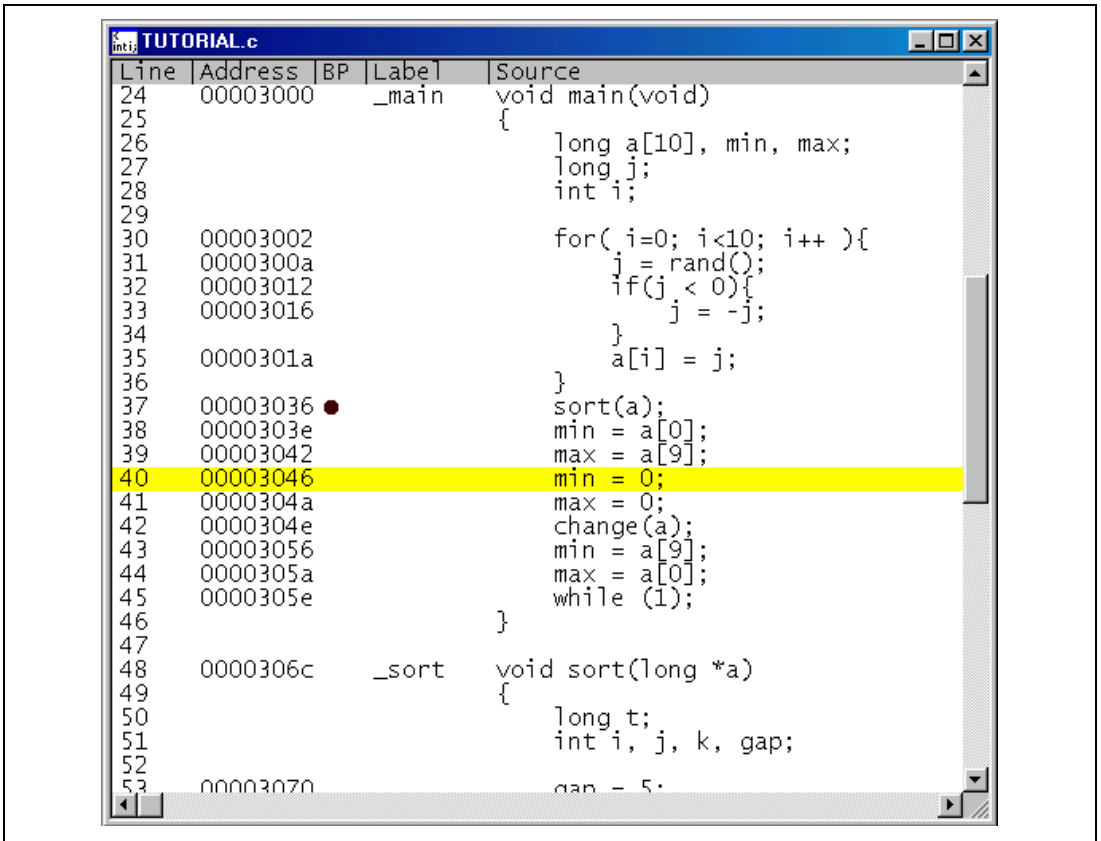


Figure 3.26 [HDI] Window (Step In → Step In)

- The value of max displayed in the [Watch] window is changed to the maximum data value.

3.13.3 Executing [Step Over] Command

The [Step Over] executes a function call as a single step and stops at the next statement of the main program.

- Using [Step Over], execute two steps to reach the change function statement.

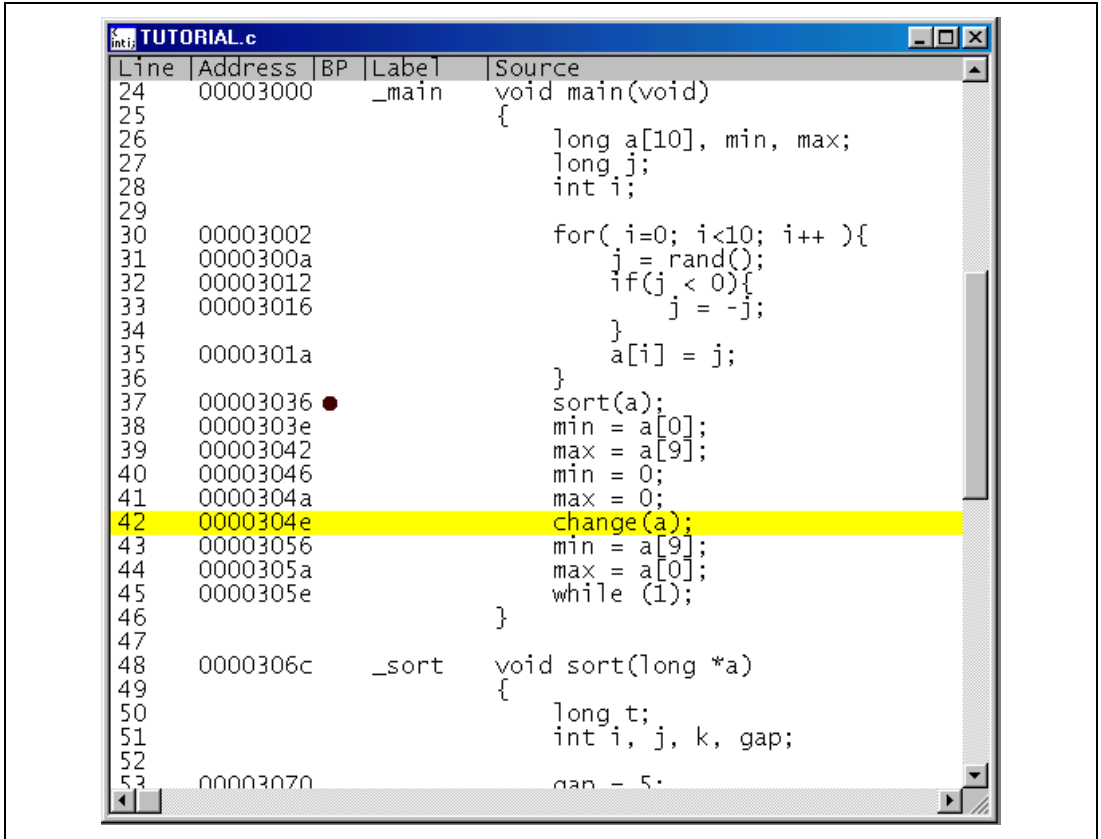


Figure 3.27 [Source] Window (Before Step Over Execution)

- To step through all statements in the change function at a single step, select [Step Over] from the [Run] menu, or click the [Step Over] button in the toolbar.



Figure 3.28 [Step Over] Button

Line	Address	BP	Label	Source
24	00003000		_main	void main(void)
25				{
26				long a[10], min, max;
27				long j;
28				int i;
29				
30	00003002			for(i=0; i<10; i++){
31	0000300a			j = rand();
32	00003012			if(j < 0){
33	00003016			j = -j;
34				}
35	0000301a			a[i] = j;
36				}
37	00003036			sort(a);
38	0000303e			min = a[0];
39	00003042			max = a[9];
40	00003046			min = 0;
41	0000304a			max = 0;
42	0000304e			change(a);
43	00003056			min = a[9];
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				
48	0000306c		_sort	void sort(long *a)
49				{
50				long t;
51				int i, j, k, gap;
52				
53	00003070			gap = 5;

Figure 3.29 [HDI] Window (Step Over)

3.14 Forced Breaking of Program Executions

The HDI can force a break in the execution of a program.

- To execute the remaining sections of the main function, select [Go] from the [Run] menu or the [Go] button on the toolbar.



Figure 3.30 [Go] Button

- The program goes into an endless loop. To force a break in execution, select [Halt] from the [Run] menu or the [Stop] button on the toolbar.



Figure 3.31 [Stop] Button

- The highlighted line of the [Program] window moves to the `while` statement, and the value of `max` displayed in the [Watch] window is updated to the most recent value.

3.15 Displaying Local Variables

The user can display local variables in a function using the [Locals] window. For example, we will examine the local variables in the `main` function, which declares five local variables: `a`, `j`, `i`, `min`, and `max`.

- Select [Locals] from the [View] menu. The [Locals] window is displayed. Initially, the [Locals] window is empty because local variables have not yet been declared. The [Locals] window will now show the local variables and their values.

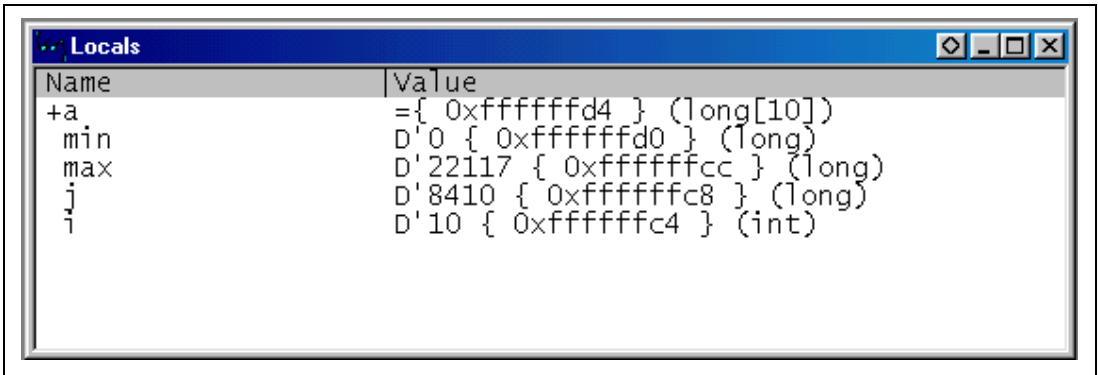


Figure 3.32 [Locals] Window

- Double-click the + symbol in front of array `a` in the [Locals] window to display the elements of array `a`.
- Refer to the elements of array `a` before and after the execution of the `sort` function, and confirm that random data is sorted in descending order.

3.16 Break Function

The emulator has software and hardware break functions. With the HDI, a software breakpoint can be set using the [Breakpoints] window, and a hardware break condition can be set using the [Break Condition] dialog box.

An overview and setting of the break function are described below.

3.16.1 Software Break Function

The emulator can set up to 255 software breakpoints. Other methods for setting a software breakpoint than in section 3.7 are described below.

- Select [Breakpoints] from the [View] menu. The [Breakpoints] window is displayed.
- Click the [Breakpoints] window with the right mouse button and select [Delete All] from the pop-up menu to cancel all the breakpoints that have been set.

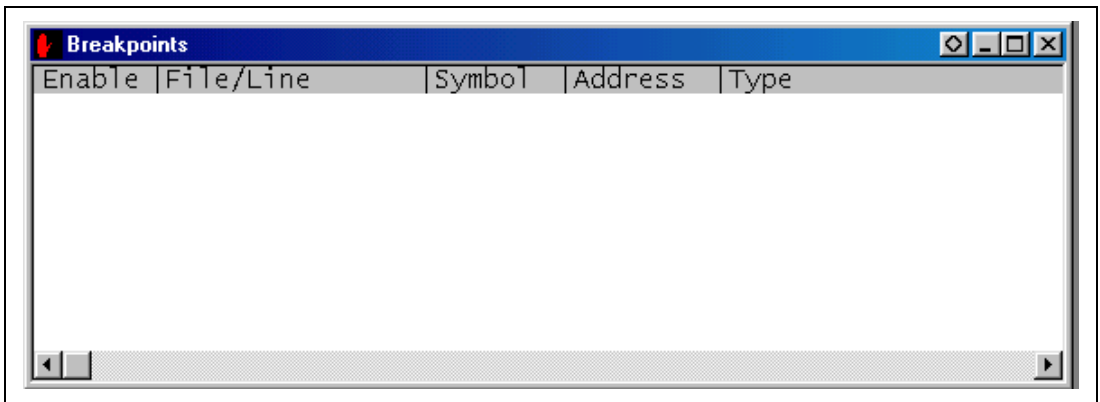


Figure 3.33 [Breakpoints] Window (Before Software Breakpoint Setting)

- Click the [Breakpoints] window with the right mouse button and select [Add] from the pop-up menu.

The [Break] dialog box is displayed. The [Point] page is displayed by default.

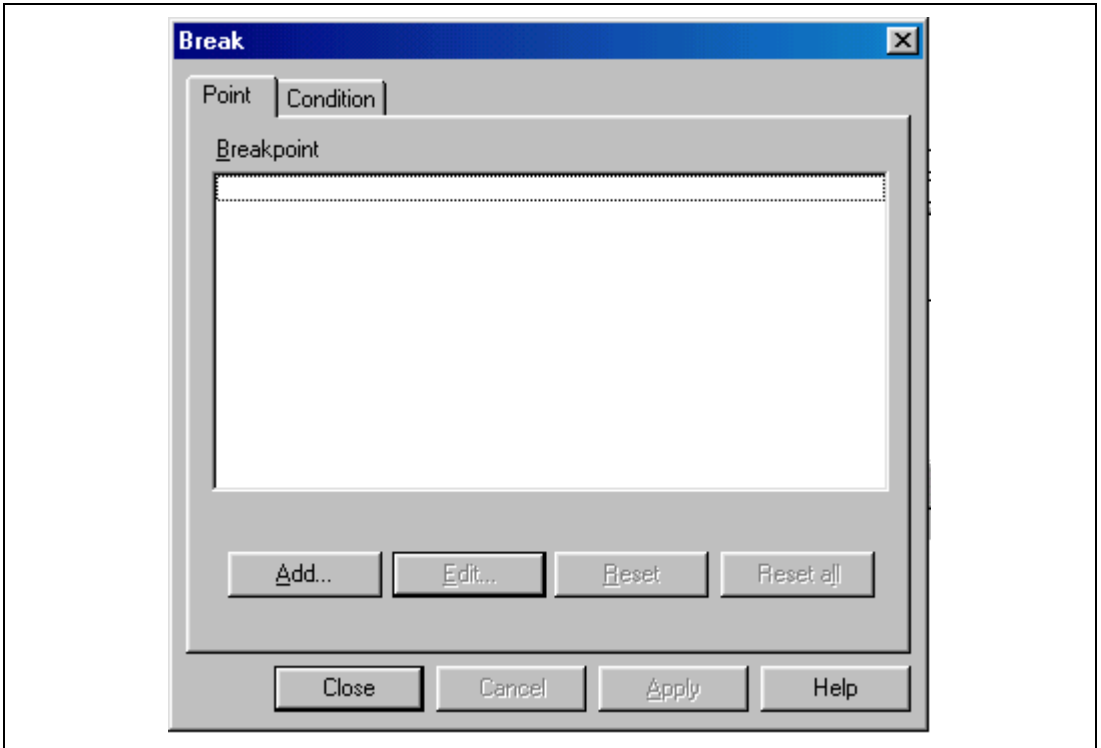


Figure 3.34 [Point] Page ([Break] Dialog Box)

- Click the [Add...] button to display the [Breakpoint] dialog box.
- Enter *H'00003056* in the [Value] edit box.

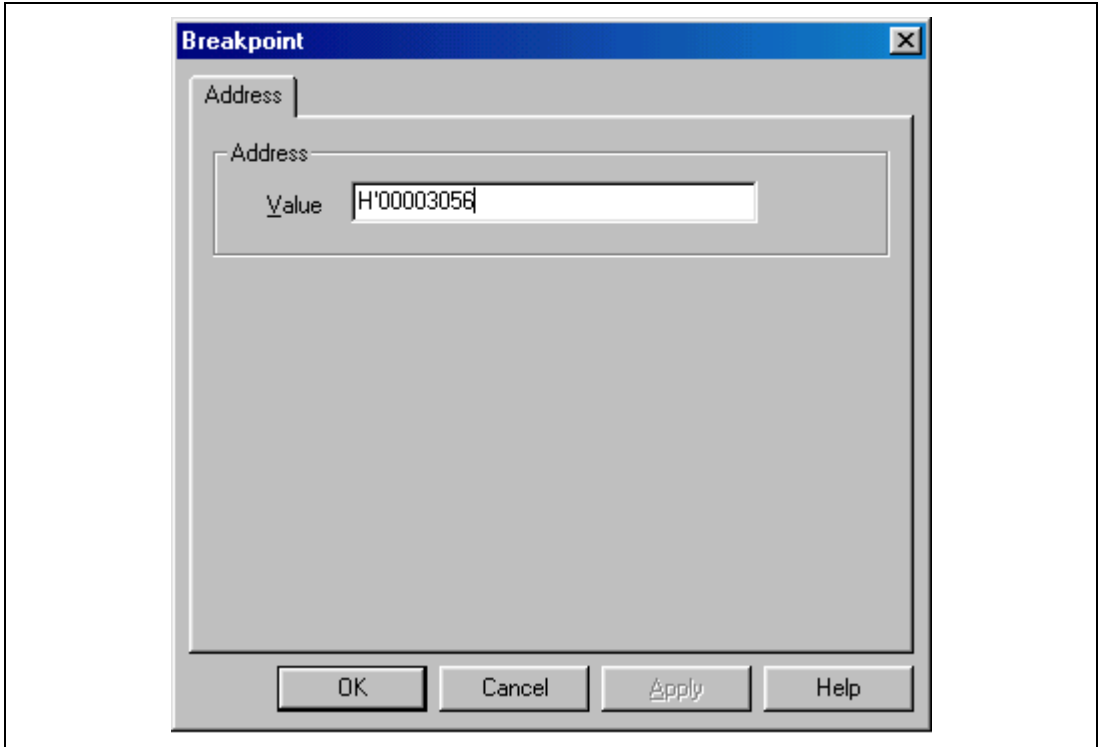


Figure 3.35 [Break Point] Dialog Box

- Click the [OK] button.

The [Break] dialog box is displayed. The address set in the value field of [Breakpoint] and the memory space are displayed.

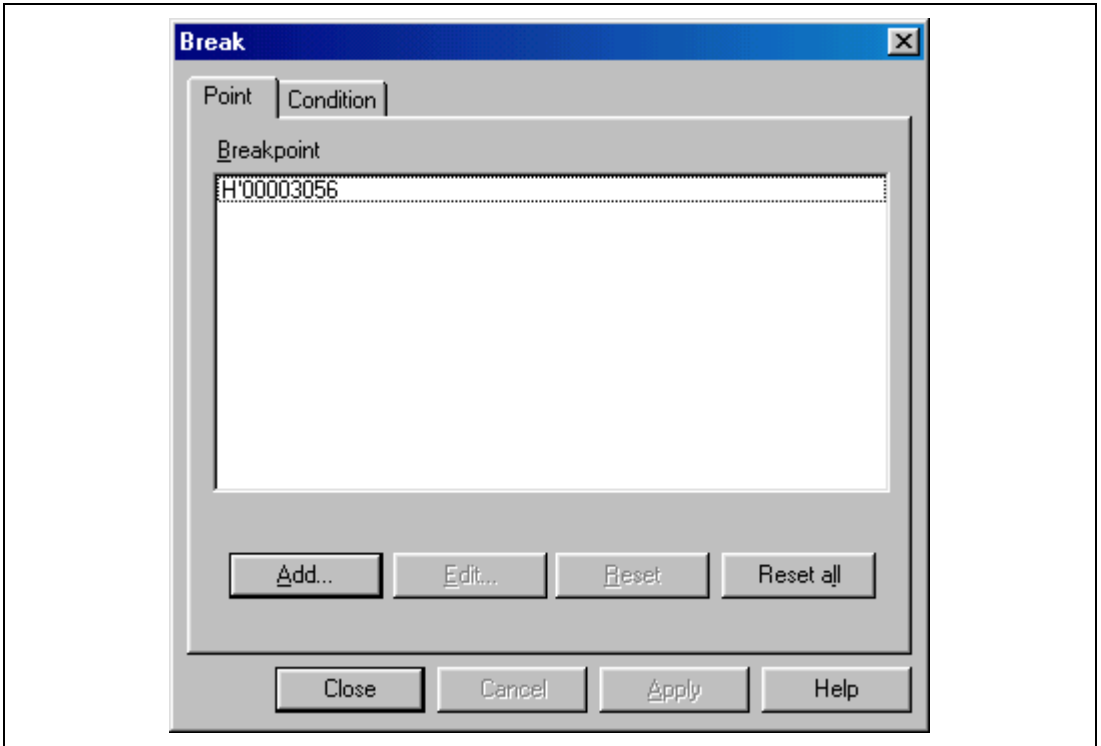


Figure 3.36 [Point] Page ([Break] Dialog Box) (After Software Breakpoint Setting)

- Click the [Close] button (or the [OK] button in some emulator products).

The software breakpoint that has been set is displayed in the [Breakpoints] window.

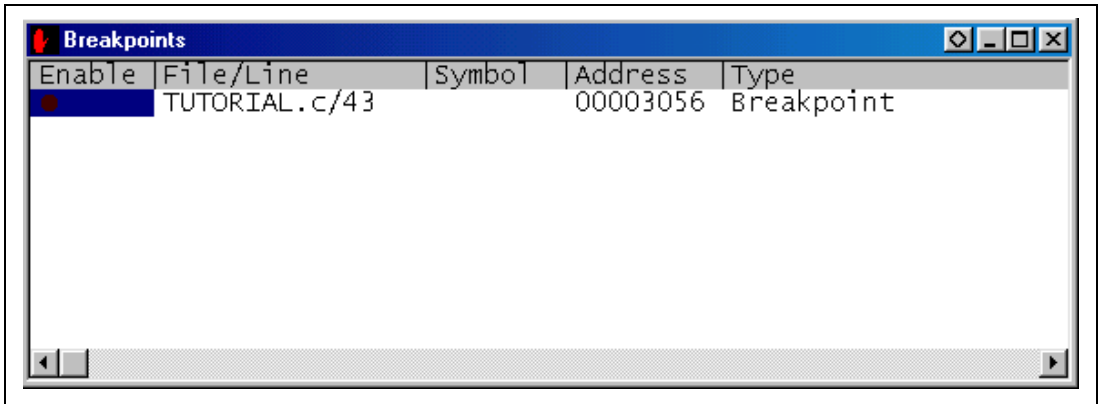


Figure 3.37 [Breakpoints] Window (Software Breakpoint Setting)

To stop the tutorial program at the breakpoint, the following procedure must be executed:

- Close the [Breakpoints] window.
- Set the program counter and stack pointer values (PC = H'00000800 and R15 = H'FFFFFFFC) that have been set in section 3.8, Setting Registers, in the [Registers] window. Click the [Go] button.

The program runs, and stops at the set breakpoint.

Line	Address	BP	Label	Source
24	00003000		_main	void main(void)
25				{
26				long a[10], min, max;
27				long j;
28				int i;
29				
30	00003002			for(i=0; i<10; i++){
31	0000300a			j = rand();
32	00003012			if(j < 0){
33	00003016			j = -j;
34				}
35	0000301a			a[i] = j;
36				}
37	00003036			sort(a);
38	0000303e			min = a[0];
39	00003042			max = a[9];
40	00003046			min = 0;
41	0000304a			max = 0;
42	0000304e			change(a);
43	00003056			min = a[9];
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				
48	0000306c		_sort	void sort(long *a)
49				{
50				long t;
51				int i, j, k, gap;
52				
53	00003070			gap = 5;

Figure 3.38 [Source] Window at Execution Stop (Software Break)

The [System Status] window displays the following contents.

Item	Status
Connected to:	SHxxxx E10A Emulator (E10A PCI Card Driver3)
CPU	SHxxxx
Run status	Break
Cause of last break	BREAK POINT
Run time count	0h:0min:0s:10ms
Emulation mode	Normal
Big endian	
AUD	None

Figure 3.39 Displayed Contents of the [System Status] Window (Software Break)

Note: The items that can be displayed in this window differ according to the product. For the items that can be displayed, refer to the online help.

3.17 Hardware Break Function

A method is given below in which the address bus condition and the read cycles for the bus status condition are set under Break Condition 1 as hardware break conditions.

- Select [Breakpoint Window] from the [View] menu. The [Breakpoints] window is displayed.
- Click the [Breakpoints] window with the right mouse button and select [Delete All] from the pop-up menu to cancel all breakpoints that have been set.
- Click the [Breakpoints] window with the right mouse button and select [Add] from the pop-up menu.

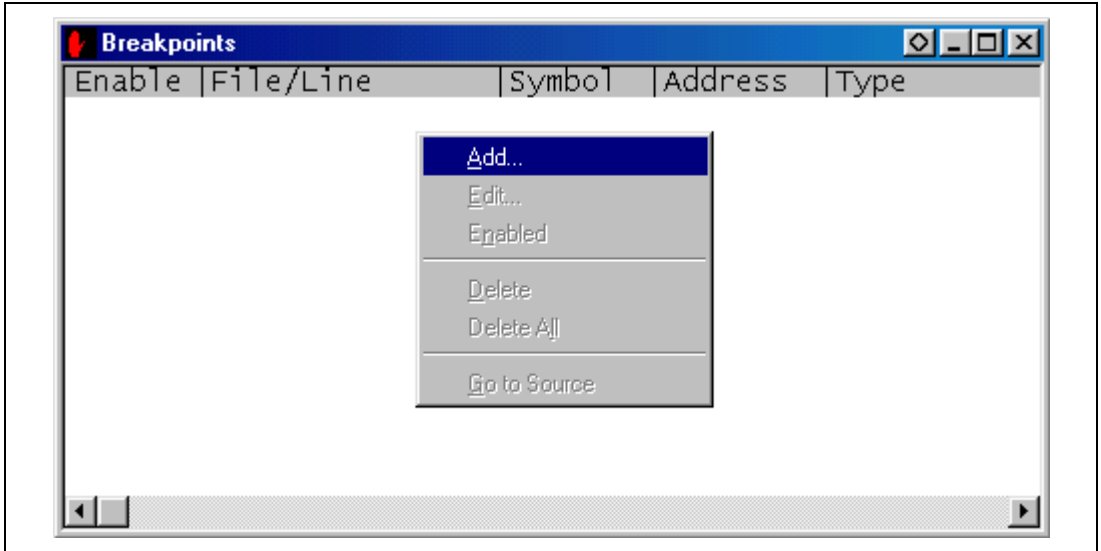


Figure 3.40 [Breakpoints] Window (Before Hardware Break Condition Setting)

The [Break] dialog box is displayed. To set hardware break conditions, select [Condition] in the [Break] dialog box to display the [Condition] page.

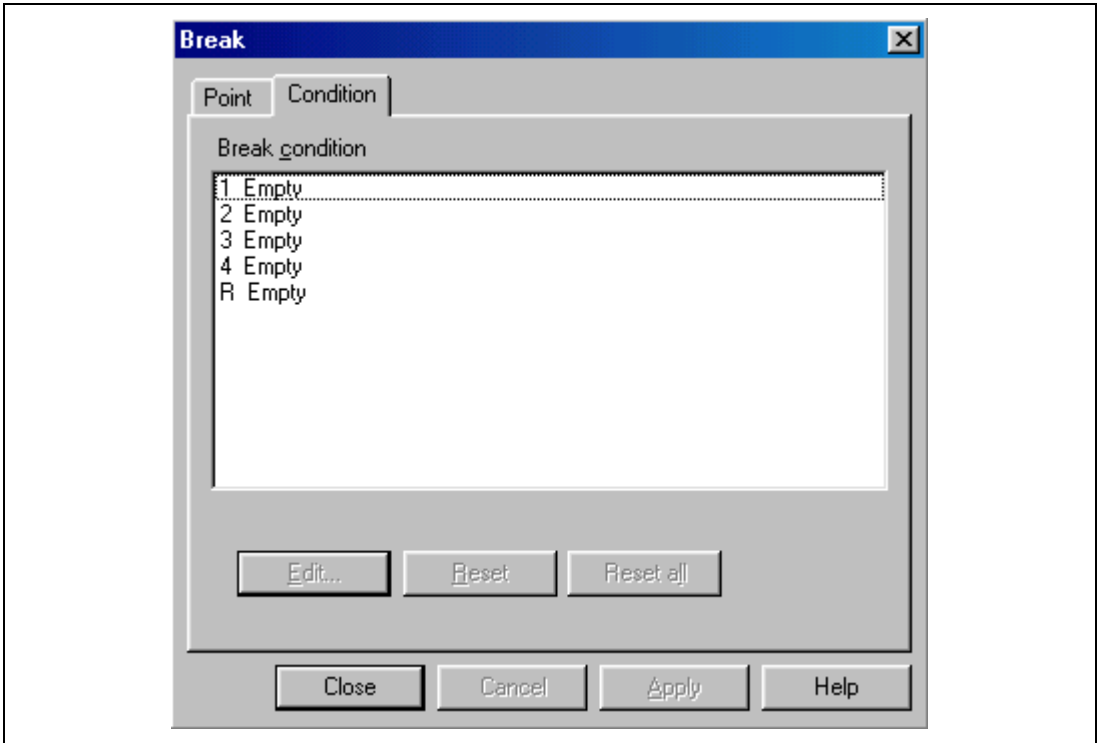


Figure 3.41 [Condition] Page ([Break] Dialog Box)

Up to five breakpoints can be set independently for the Break Condition hardware break condition. However, Break Condition R can be set as the reset point only when the sequential break function is used. In this example, set the hardware break condition for Break Condition 1.

Note: The number of hardware break conditions differs according to the product. For the number that can be specified for each product, refer to the online help.

- Highlight the first point in the [Break Condition] list box.
- Click the [Edit...] button. The [Break Condition 1] dialog box is displayed.

- Clear the [Don't Care] check box in the [Address] page.
- Select the [Address] radio button and enter *H'00003046* as the value in the [Address] edit box.

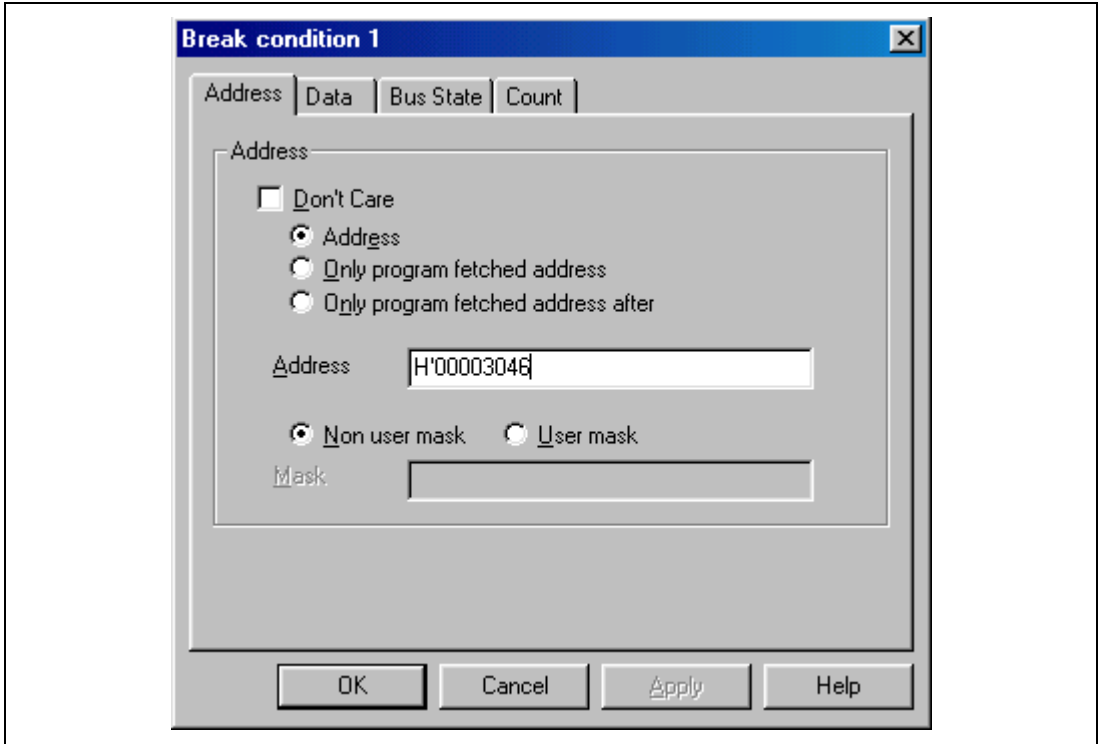


Figure 3.42 [Address] Page ([Break Condition 1] Dialog Box)

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

- Select [Bus State] to display the [Bus State] page.
- Select the [Read] radio button in the [Read/Write] group box.

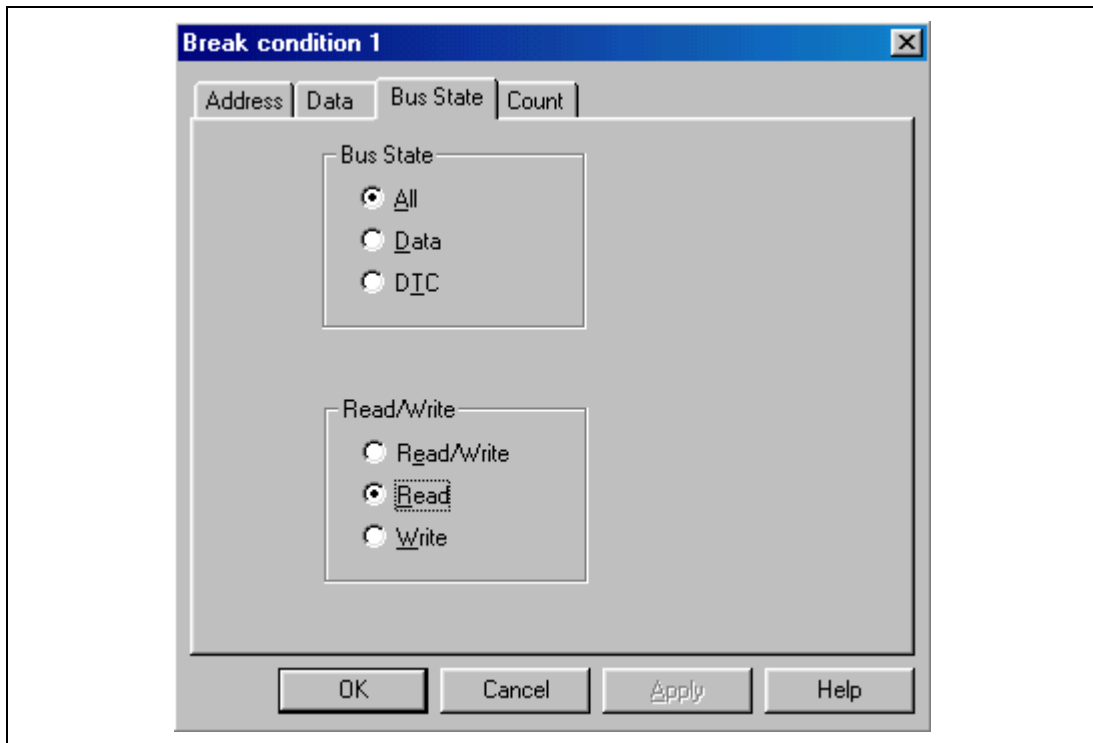


Figure 3.43 [Bus State] Page ([Break Condition 1] Dialog Box)

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

- Click the [OK] button.
- The [Break] dialog box is displayed, and the first point display in the [Break Condition] list box changes from Empty to Enable.

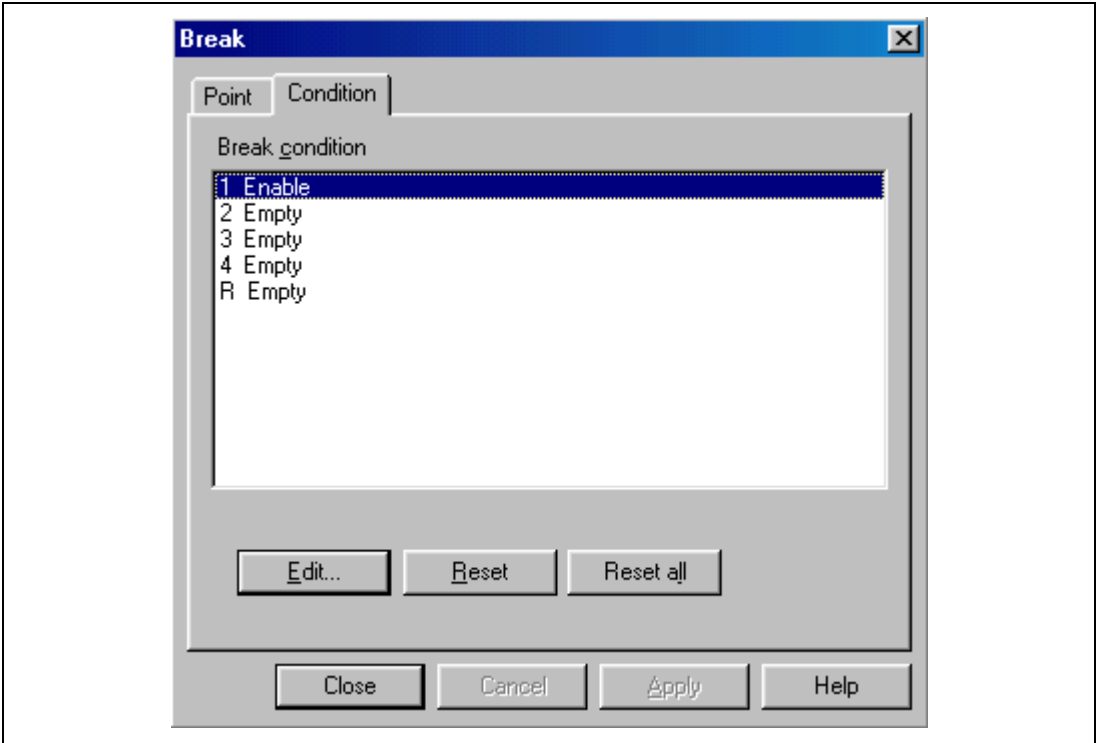


Figure 3.44 [Break] Dialog Box (After Hardware Break Condition Setting)

Note: The number of hardware break conditions differs according to the product. For the number that can be specified for each product, refer to the online help.

- Click the [Close] button.

The newly set hardware breakpoint is displayed in the [Breakpoints] window. With this setting, Break Condition 1 is displayed in [Type] in the [Breakpoints] window.

This completes the setting of the Break Condition 1 hardware break condition. When the program is executed, a break will occur when address H'00003046 is accessed in a read cycle.

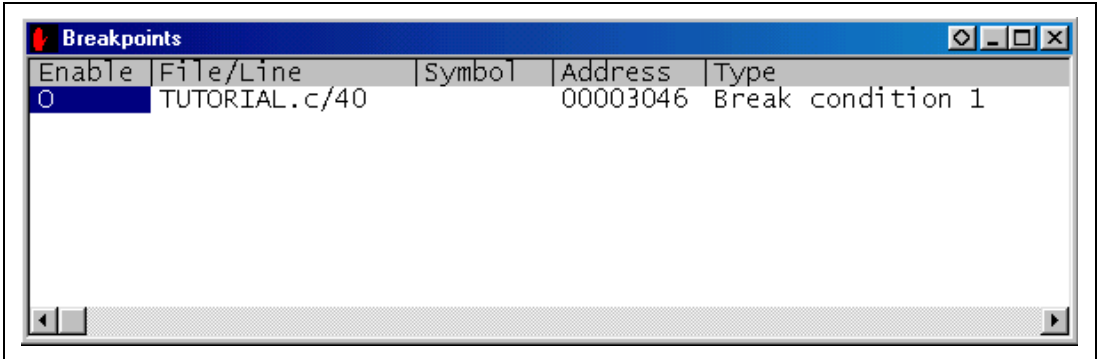
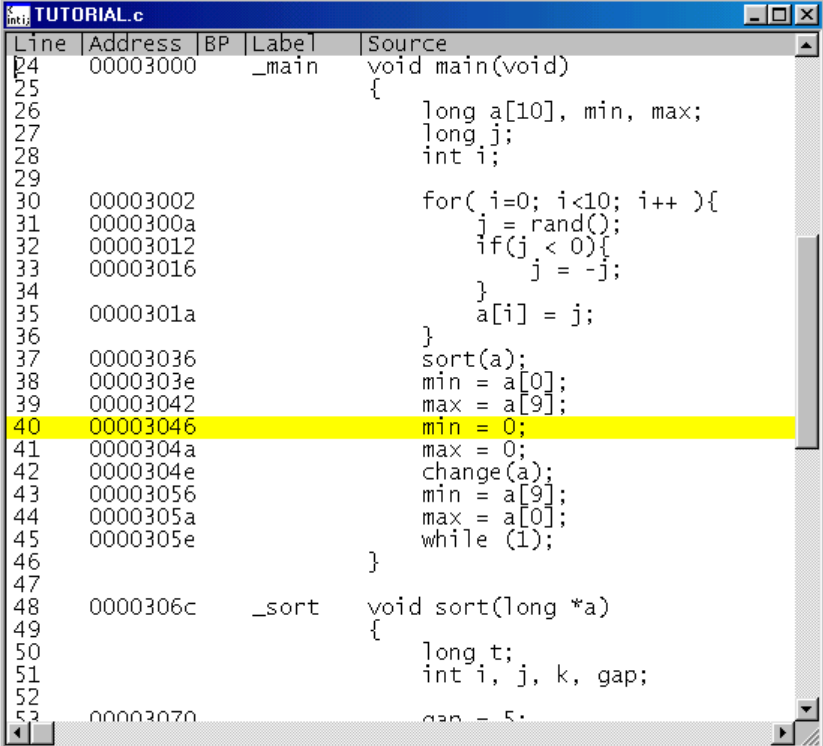


Figure 3.45 [Breakpoints] Window ([Break Condition 1] Setting)

- Close the [Breakpoints] window.
- Set the program counter and stack pointer values (PC = H'00000800 and R15 = H'FFFFFFFC) that have been set in section 3.8, Setting Registers, in the [Registers] window. Click the [Go] button.

The program runs and then stops at the condition specified under Break Condition 1.



Line	Address	BP	Label	Source
24	00003000		_main	void main(void)
25				{
26				long a[10], min, max;
27				long j;
28				int i;
29				
30	00003002			for(i=0; i<10; i++){
31	0000300a			j = rand();
32	00003012			if(j < 0){
33	00003016			j = -j;
34				}
35	0000301a			a[i] = j;
36				}
37	00003036			sort(a);
38	0000303e			min = a[0];
39	00003042			max = a[9];
40	00003046			min = 0;
41	0000304a			max = 0;
42	0000304e			change(a);
43	00003056			min = a[9];
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				}
48	0000306c		_sort	void sort(long *a)
49				{
50				long t;
51				int i, j, k, gap;
52				
53	00003070			gap = 5;

Figure 3.46 [Source] Window at Execution Stop (Break Condition 1)

The [System Status] window displays the following contents.

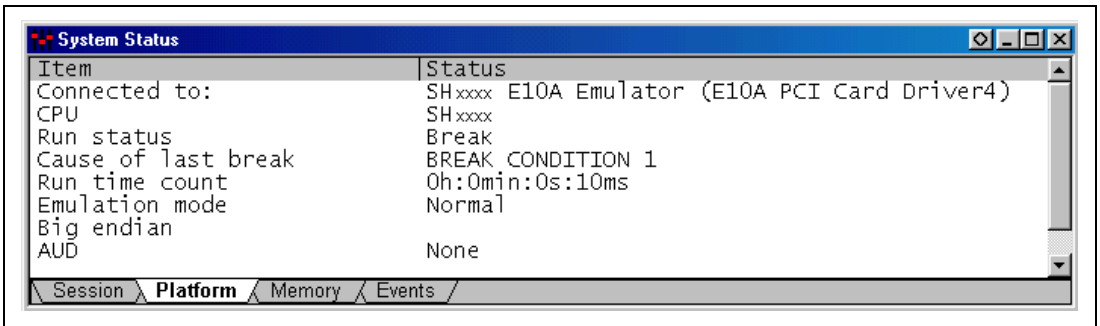


Figure 3.47 Displayed Contents of the [System Status] Window (Break Condition 1)

Note: The items that can be displayed in this window differ according to the product. For the items that can be displayed, refer to the online help.

3.17.1 Setting the Sequential Break Condition

The emulator has sequential break functions. When the hardware break conditions listed in table 3.3 are satisfied, program execution halts. This mode is called sequential break.

Table 3.3 Sequential Break Conditions

Break Condition	Description
Sequential break condition 2-1	Program is halted when Break Condition 2 and Break Condition 1 are satisfied in that order.

Sequential break condition 2-1 is described below as an example.

Before executing the program, change setting in the [Configuration] dialog box. Otherwise, the sequential break does not work.

- Select [Configure Platform...] from the [Setup] menu. The [Configuration] dialog box is displayed.
- Select Sequential break condition 2-1 from the [Emulation mode] combo box.

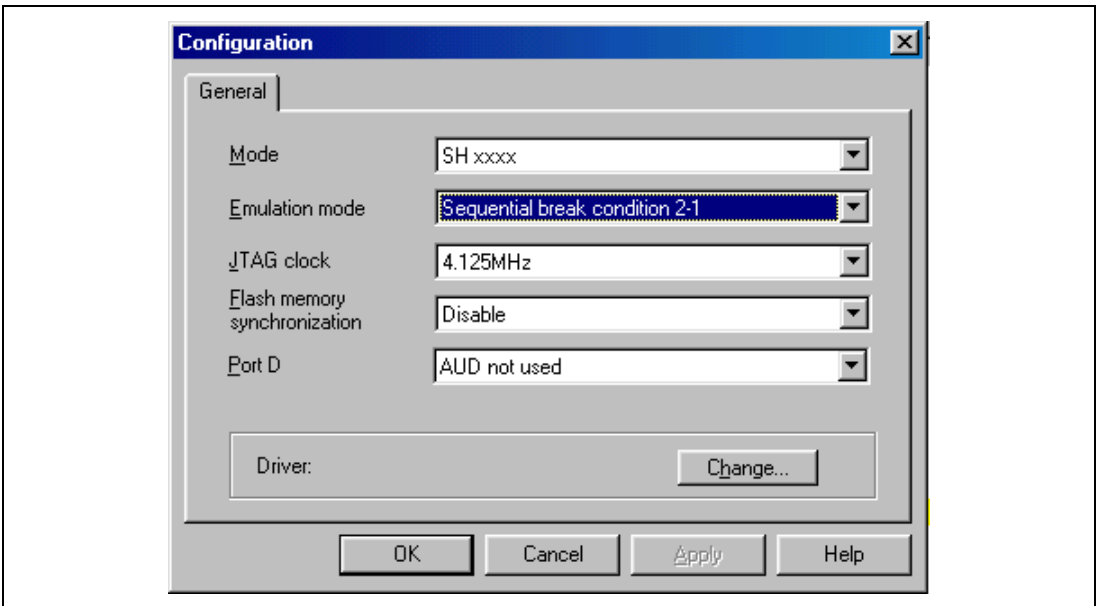


Figure 3.48 [Configuration] Dialog Box (Sequential Break Setting)

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

Click the [OK] button and close the [Configuration] dialog box.

Set hardware break conditions as follows:

Break condition 1: When address H'0000305A is accessed in a read cycle, a break condition is satisfied.

Break condition 2: When address H'00003046 is accessed in a read cycle, a break condition is satisfied.

Follow the setting method described in the previous section.

- When Break Condition 1,2 setting has been completed, the state of the [Breakpoints] window is as follows.

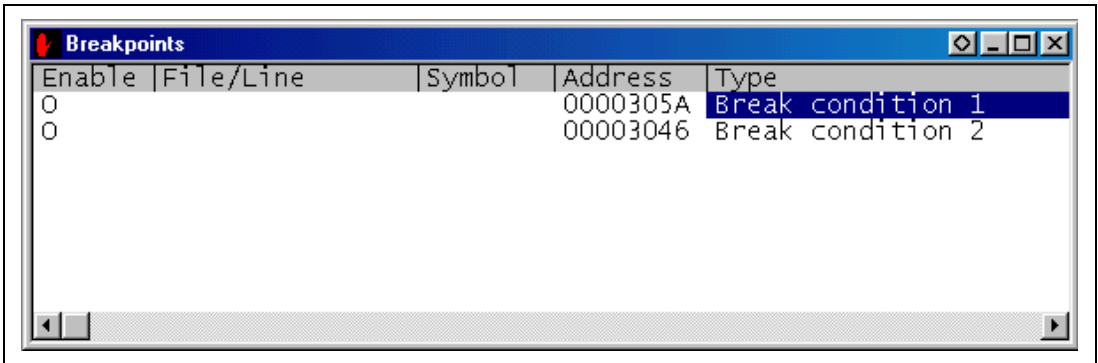
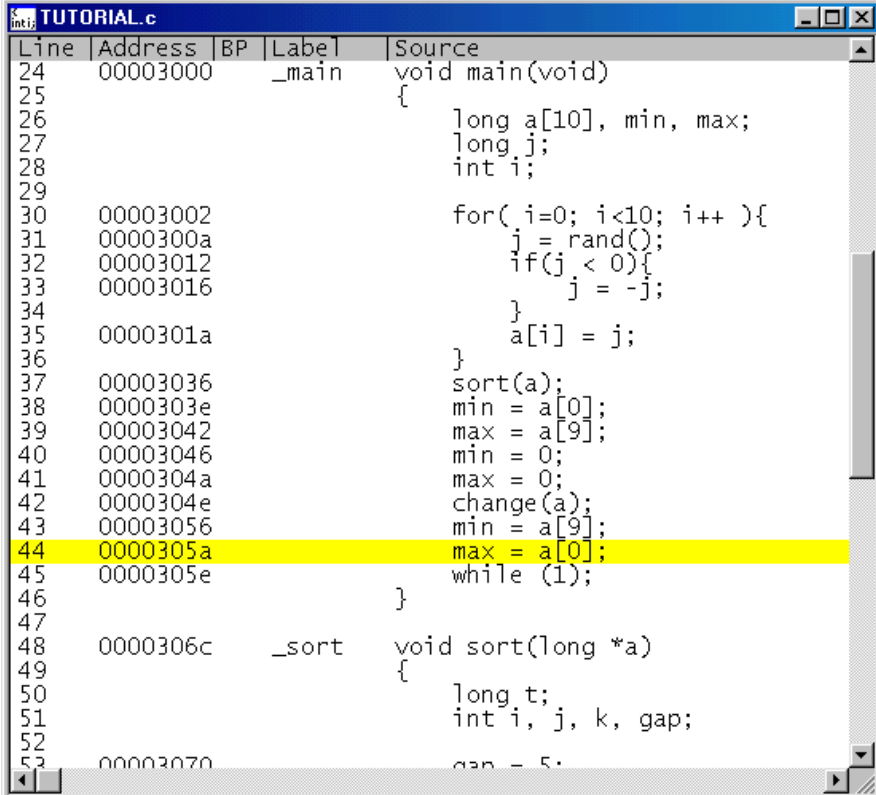


Figure 3.49 [Breakpoints] Window (After Sequential Break Condition Setting)

- Close the [Breakpoints] window.
- Set the program counter and stack pointer values (PC = H'800 and R15 = H'FFFFFFC) that have been set in section 3.8, Setting Registers, in the [Registers] window. Click the [Go] button.

The program runs and then stops at the condition specified under Break Condition 1.



```
TUTORIAL.c
Line Address BP Label Source
24 00003000 _main void main(void)
25 {
26     long a[10], min, max;
27     long j;
28     int i;
29
30 00003002     for( i=0; i<10; i++ ){
31 0000300a         j = rand();
32 00003012         if(j < 0){
33 00003016             j = -j;
34
35 0000301a         a[i] = j;
36     }
37 00003036     sort(a);
38 0000303e     min = a[0];
39 00003042     max = a[9];
40 00003046     min = 0;
41 0000304a     max = 0;
42 0000304e     change(a);
43 00003056     min = a[9];
44 0000305a     max = a[0];
45 0000305e     while (1);
46 }
47
48 0000306c _sort void sort(long *a)
49 {
50     long t;
51     int i, j, k, gap;
52
53     gap = 5;
```

Figure 3.50 [Source] Window at Execution Stop (Sequential Break)

The [System Status] window displays the following contents.

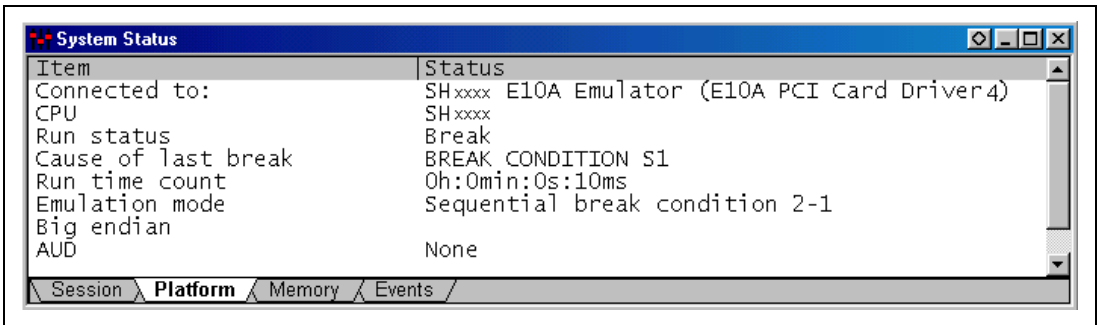


Figure 3.51 Displayed Contents of the [System Status] Window (Sequential Break)

Note: The items that can be displayed in this window differ according to the product. For the items that can be displayed, refer to the online help.

3.18 Trace Functions

AUD Trace Function

This is the large-capacity trace function that is enabled when the AUD pin is connected to the emulator. This function displays the branch destination addresses, mnemonics, operands, and source lines.

When the branch destination instructions are one branch, the number of branch instructions acquired by a trace is a maximum of 16,383 in the PCMCIA-type emulator and a maximum of 65,535 in the PCI-type emulator.

Table 3.4 shows the AUD trace function.

- Notes:
1. The AUD trace function is not supported for all products. For the specifications of each product, refer to the section related to the trace functions in section 6, SHxxxx E10A Emulator Specifications, or to the online help.
 2. The AUD trace function is extended for some products. For the specifications of each product or the number of acquired branches, refer to the section related to the trace functions in section 6, SHxxxx E10A Emulator Specifications, or to the online help.
 3. In some products, the AUD pin may be invalid. In such case, set the AUD pin to make it valid before using the AUD trace function.

Table 3.4 AUD Trace Functions

Type	Mode	Description
Acquisition mode when the trace buffer of the emulator becomes full	Trace continue	This function always overwrites the oldest trace information to acquire the latest trace information.
	Trace stop	The trace information is not acquired. The user program is continuously executed.

3.18.1 AUD Trace Function

This function is operational when the AUD pin of the MCU is connected to the emulator.

The following is the procedure for setting the AUD trace function (this function need not to be set in an emulator that does not support the AUD trace function):

1. Select [Trace] from the [View] menu.
2. Click the [Trace] window with the right mouse button and select [Acquisition] from the pop-up menu to display the [Trace Acquisition] window.
3. Select the [Trace continue] radio button in the [Trace mode] group box.

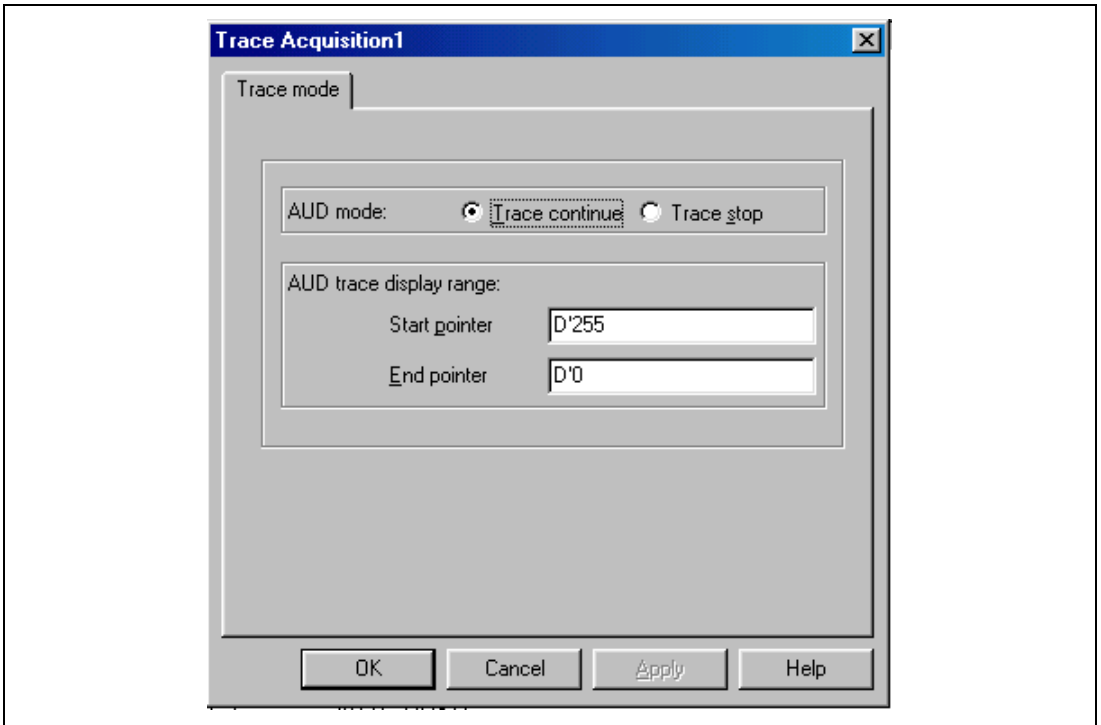


Figure 3.52 [Trace mode] Window

Note: For a description of each option, refer to table 3.4.

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

The trace results are displayed in the [Trace] window after the program execution is completed.

The following is an example of the display in the SH7047F E10A emulator.

The screenshot shows a trace window titled "Trace - 156 records (no filter)". The window contains a table of assembly instructions with columns for No., IP, TYPE, ADDR, MNEMONIC, OPERAND, and Source. The instructions are MOV.L operations with various operands and destinations. The source code on the right side of the window includes several "for" loops and a "min = a[0];" statement.

No.	IP	TYPE	ADDR	MNEMONIC	OPERAND	Source
-000035	-D'000036	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000034	-D'000035	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000033	-D'000034	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000032	-D'000033	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000031	-D'000032	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000030	-D'000031	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000029	-D'000030	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000028	-D'000029	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000027	-D'000028	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000026	-D'000027	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000025	-D'000026	DESTINATION	000030E4	MOV.L	@R15,R3	for
-000024	-D'000025	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000023	-D'000024	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000022	-D'000023	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000021	-D'000022	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000020	-D'000021	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000019	-D'000020	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000018	-D'000019	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000017	-D'000018	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000016	-D'000017	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000015	-D'000016	DESTINATION	000030E4	MOV.L	@R15,R3	for
-000014	-D'000015	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000013	-D'000014	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000012	-D'000013	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000011	-D'000012	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000010	-D'000011	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000009	-D'000010	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000008	-D'000009	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000007	-D'000008	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000006	-D'000007	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000005	-D'000006	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000004	-D'000005	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000003	-D'000004	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000002	-D'000003	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000001	-D'000002	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
+000000	-D'000001	DESTINATION	0000303E	MOV.L	@(H'10:4,R15),R3	min = a[0];

Figure 3.53 [Trace] Window in the SH7047F E10A Emulator

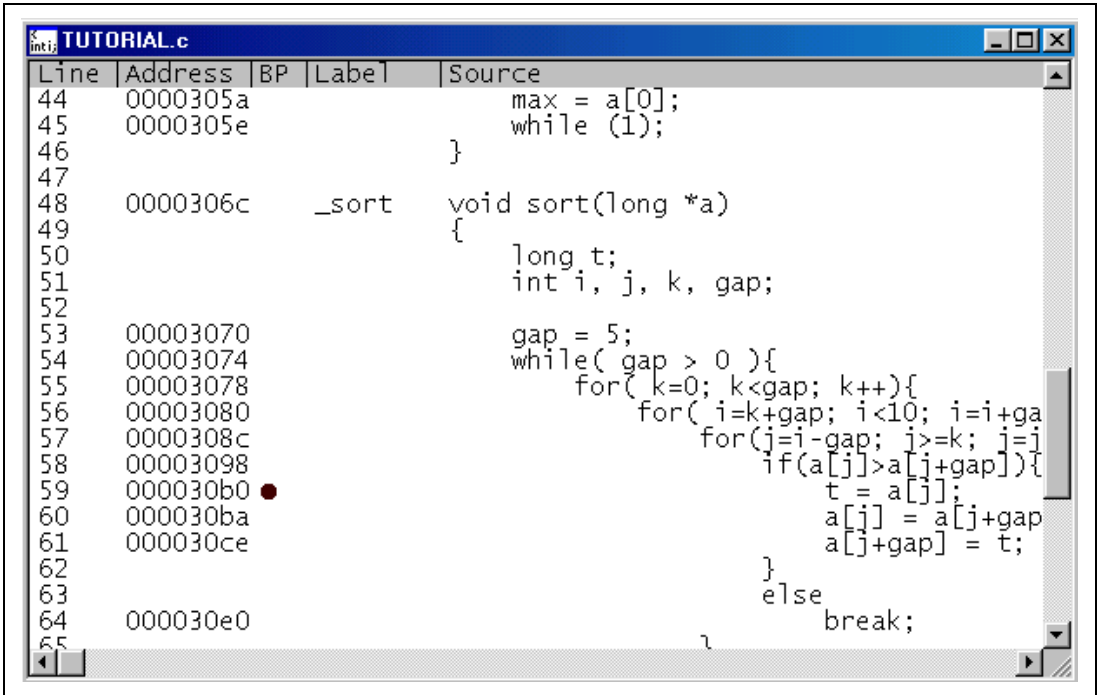
3.19 Stack Trace Function

The emulator uses the stack information to display the name of the calling function for a function at which the program counter is currently pointing.

Notes: 1. This function can be used only when the load module with the Dwarf2-type debugging information is loaded.

2. For details on the stack trace function, refer to the online help.

- Double-click the [BP] column in the `sort` function and set a software breakpoint.



Line	Address	BP	Label	Source
44	0000305a			max = a[0];
45	0000305e			while (1);
46				}
47				
48	0000306c		_sort	void sort(long *a)
49				{
50				long t;
51				int i, j, k, gap;
52				
53	00003070			gap = 5;
54	00003074			while(gap > 0){
55	00003078			for(k=0; k<gap; k++){
56	00003080			for(i=k+gap; i<10; i=i+ga
57	0000308c			for(j=i-gap; j>=k; j=j
58	00003098			if(a[j]>a[j+gap]){
59	000030b0			t = a[j];
60	000030ba			a[j] = a[j+gap
61	000030ce			a[j+gap] = t;
62				}
63				else
64	000030e0			break;
65				}

Figure 3.54 [Source] Window (Software Breakpoint Setting)

- Set the same program counter and stack pointer values (PC = H'00000800 and R15 = H'FFFFFFFC) as were set in section 3.8, Setting Registers (again, use the [Registers] window). Click the [Go] button.
- After the break in execution, select [Stack Trace] from the [View] menu to open the [Stack Trace] window.

Kind	Name	Value
F	sort(long*)	{ 0x000030b0 }
F	main()	{ 0x0000303e }
F	main()	{ 0x0000303e }
F	main()	{ 0x0000303e }
F	main()	{ 0x0000303e }
F	main()	{ 0x0000303e }
F	main()	{ 0x0000303e }
F	main()	{ 0x0000303e }
F	main()	{ 0x0000303e }
F	main()	{ 0x0000303e }

Figure 3.55 [Stack Trace] Window

Figure 3.55 shows that the position of the program counter is currently in the `sort()` function, and that the `sort()` function is called from the `main()` function.

- Notes:
1. If the function is not deeply nested (lower than 10), the `main()` function will be displayed multiple times.
 2. For details on this function, refer to the online help. The online help is opened when the [F1] key is pressed in the [Stack Trace] window.

3.20 What Next?

This tutorial has described the major features of the emulator and the use of the HDI.

Sophisticated debugging can be carried out by using the emulation functions that the emulator offers. This provides for effective investigation of hardware and software problems by accurately isolating and identifying the conditions under which such problems arise.

Further details on the use of the HDI can be found in the separately issued Hitachi Debugging Interface User's Manual.

Section 4 Descriptions of Windows

4.1 HDI Windows

HDI window menu bars and the corresponding pull-down menus are listed in table 4.1. Where a description of a menu is included in the Hitachi Debugging Interface User's Manual or in this manual, a O mark or the relevant section number is shown. Related commands in the E10A Emulator User's Manual are also shown.

Table 4.1 HDI Window Menus and Related Manual Entries

Menu Bar	Pull-Down Menu	Hitachi Debugging Interface User's Manual	This Manual
File menu	New Session...	O	—
	Load Session...	O	—
	Save Session	O	2.6
	Save Session As...	O	—
	Load Program...	O	3.7.1
	Initialize	O	—
	Exit	O	—
Edit Menu	Cut	O	—
	Copy	O	—
	Paste	O	—
	Find...	O	—
	Evaluate...	O	—

Table 4.1 HDI Window Menus and Related Manual Entries (cont)

Menu Bar	Pull-Down Menu	Hitachi Debugging Interface User's Manual	This Manual
View Menu	Breakpoints	O	3.11, 3.17.1, 4.2.4, 6.5.5
	Command Line	O	—
	Disassembly...	O	—
	I/O Area	O	—
	Labels	O	—
	Locals	O	3.16
	Memory...	O	3.12
	Performance Analysis	O	—
	Profile-List	O	—
	Profile-tree	O	—
	Registers	O	3.9
	Source...	O	3.7.2
	Stack Trace	O	3.20
	Status	O	3.10, 3.17.1, 4.2.9
	Trace	O	4.2.7, 6.5.3, 6.5.7
Watch	O	3.13	
Run Menu	Reset CPU	O	—
	Go	O	3.10
	Reset Go	O	—
	Go to Cursor	O	—
	Set PC To Cursor	O	—
	Run...	O	—
	Step In	O	3.14.1
	Step Over	O	3.14.3
	Step Out	O	3.14.2
	Step...	O	—
Halt	O	—	

Table 4.1 HDI Window Menus and Related Manual Entries (cont)

Menu Bar	Pull-Down Menu	Hitachi Debugging Interface User's Manual	This Manual
Memory Menu	Refresh	O	—
	Load	O	—
	Save	O	—
	Verify	O	—
	Test	O	—
	Fill	O	—
	Copy	O	—
	Compare	O	—
Setup Menu	Status bar	O	—
	Options	O	—
	Radix	O	—
	Customise	O	—
	Configure Platform...	O	3.5, 4.2
Window Menu	Cascade	O	—
	Tile	O	—
	Arrange Icons	O	—
	Close All	O	—
Help Menu	Index	O	—
	Using Help	O	—
	Search for Help on	O	—
	About HDI	O	—

4.2 Descriptions of Each Window

This section describes each window. Figures in this section are used as examples. Each E10A emulator type has explanatory notes. Read section 6, SHxxxx E10A Emulator Specifications.

4.2.1 [Configuration] Dialog Box

Function:

This dialog box sets the emulation conditions of the emulator.

Window:

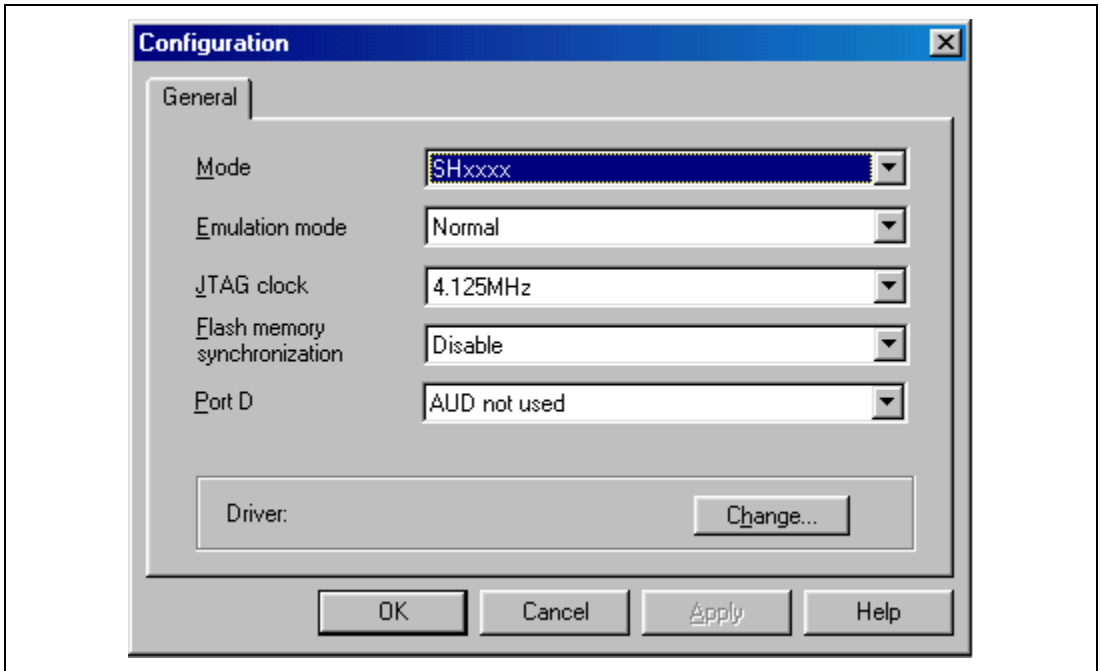


Figure 4.1 [Configuration] Dialog Box

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

Description:

The [Configuration] dialog box consists of the [General] page listed in table 4.2.

Table 4.2 [Configuration] Dialog Box Page

Page Name	Description
[General]	Sets and displays the emulation mode conditions.
[Loading flash memory]	Sets the download function for the flash memory.

Clicking the [OK] button sets the emulation conditions. If the [Cancel] button is clicked, this dialog box is closed without setting the emulation conditions.

(1) [General] Page ([Configuration] Dialog Box)

Function:

This page sets the emulator operation conditions, displays the device name, sets the emulation mode, JTAG clock (TCK), flash memory synchronization, port D, and selects the driver.

Window:

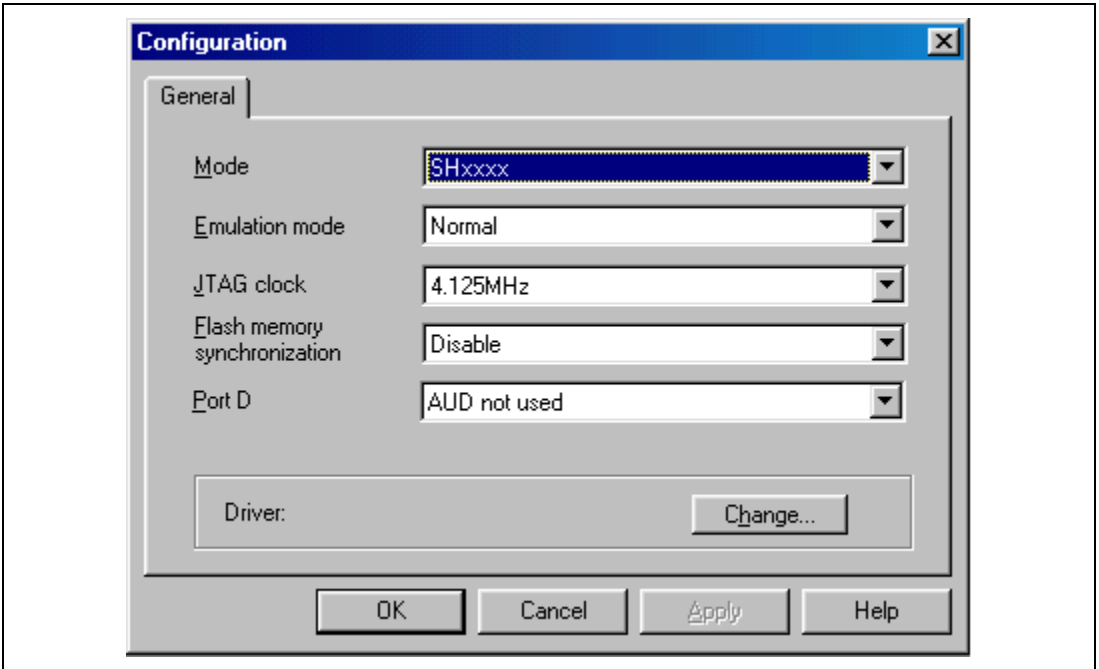


Figure 4.2 [General] Page ([Configuration] Dialog Box)

Note: The items and displayed contents that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

Description:

Table 4.3 [General] Page Options

Option	Description
[Mode] combo box	Displays the device name.
[Emulation mode] combo box	Selects the execution mode. Select Normal to perform normal emulation. Select No Break to disable breakpoint settings. Select Sequential break Condition 2-1, etc. to use the sequential break function* ¹ . (For Sequential break Condition 2-1, execution stops when conditions are satisfied in the order of Break Condition 2 and Break Condition 1.)
[JTAG clock] combo box	Sets the JTAG frequency* ³ .
[Flash memory synchronization] combo box	<p>Synchronization method is set between the host computer and the flash memory.</p> <p>When synchronization is performed from the host computer to the flash memory, a waiting time will be generated to write the flash memory during user program halting, but the displayed contents and the flash memory are always matched.</p> <p>When synchronization is performed from the flash memory to the host computer, the rewritten contents in the user program mode will be reflected since the flash memory is read during user program halting.</p> <p>Disable: Synchronization is not performed except when the E10A emulator is activated and the flash memory area is modified.</p> <p>PC to Flash memory: Synchronization is performed from the host computer to the flash memory.</p> <p>Flash memory to PC: Synchronization is performed from the flash memory to the host computer.</p> <p>PC to Flash memory, Flash memory to PC: Synchronization is performed between the host computer and the flash memory.</p>
[Port D] combo box	<p>Sets whether the multiplexed port D is used in the AUD.</p> <p>AUD not used: Port D is not used in the AUD.</p> <p>AUD used: Port D is used in the AUD.</p>
[Driver] group box	Displays the driver currently selected.
[Change...] button	Displays the [E10A Driver Details] dialog box. Use when a driver currently connected is changed.

- Notes:
1. When using the sequential break function, set the corresponding hardware break conditions.
 2. The range of frequencies that the AUD operates under is different according to the devices used. For details, refer to section 6.5.5, AUD Functions.
 3. The range of frequencies that the JTAG operates at is different according to the devices used. For details, refer to section 6.5.4, Notes on Using the JTAG Clock (TCK).

When a driver is to be changed with the [Change..] button, the following message is displayed.

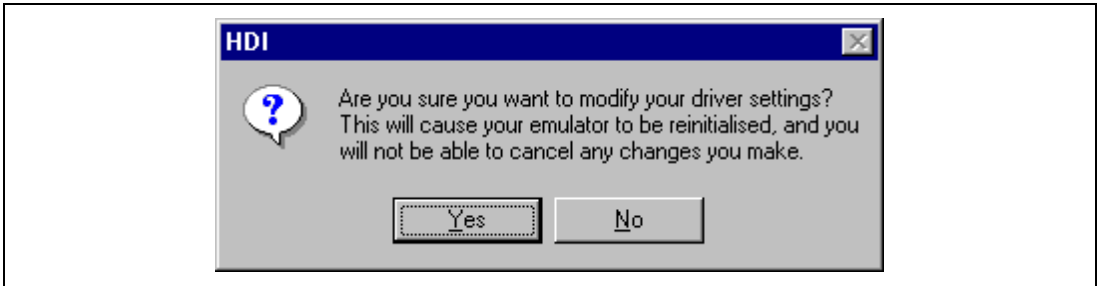


Figure 4.3 Warning Message Box

When the [Yes] button is clicked, the [E10A Driver Details] dialog box is displayed. When the [No] button is clicked, the display returns to the [Configuration] dialog box.

Related Command:

GO_OPTION command

(2) [E10A Driver Details] Dialog Box Function

Function:

When the [Change] button in the [Driver] group box is clicked on the [General] page in the [Configuration] dialog box, the [E10A Driver Details] dialog box is displayed.

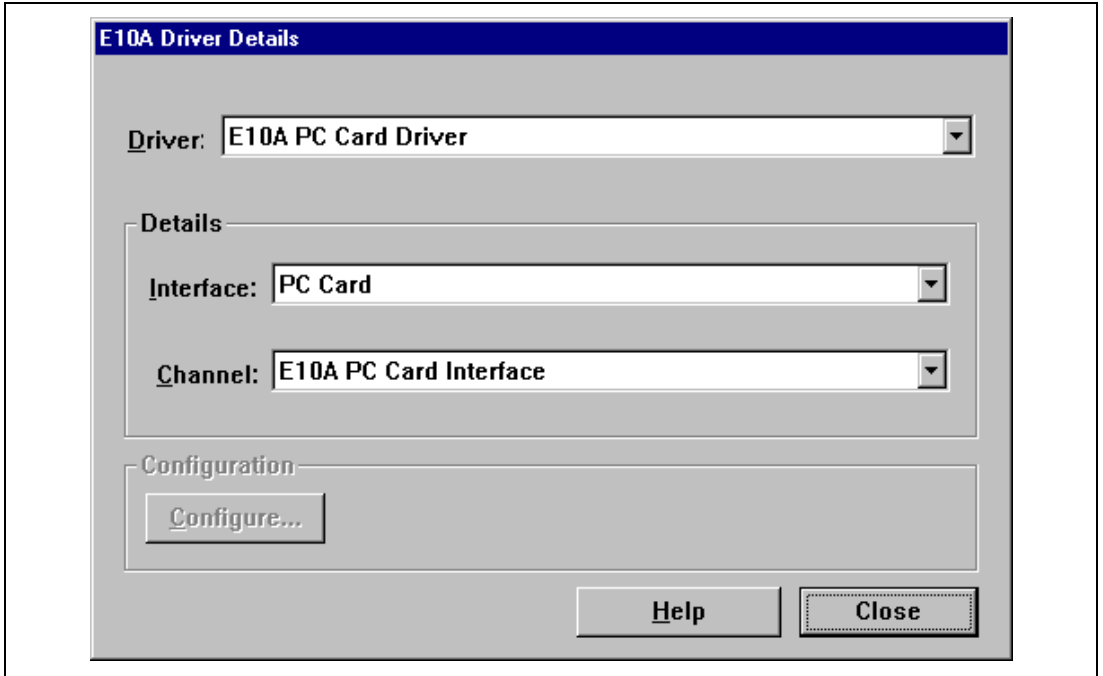


Figure 4.4 [E10A Driver Details] Dialog Box

Description:

Table 4.4 Options for the [E10A Driver Details] Dialog Box

Option	Description
[Driver] combo box	Selects the driver to connect the HDI with the emulator. Selects [E10A PC Card Driver] to use the PCMCIA card emulator. Selects [E10A PCI Card Driver] to use the PCI card emulator. For details, refer to section 6.5.1, Emulator Driver Selection.
[Interface] combo box	Displays the interface name of the card emulator to be connected. Selects [PC Card] to use the PCMCIA card emulator. Selects [PCI] to use the PCI card emulator. (If the driver is not installed, the [PC Card] or [PCI] is not displayed.)
[Channel] combo box	Displays the interface to which the board is connected.

4.2.2 [Breakpoints] Window

Function:

This window lists all break conditions that have been set.

Window:

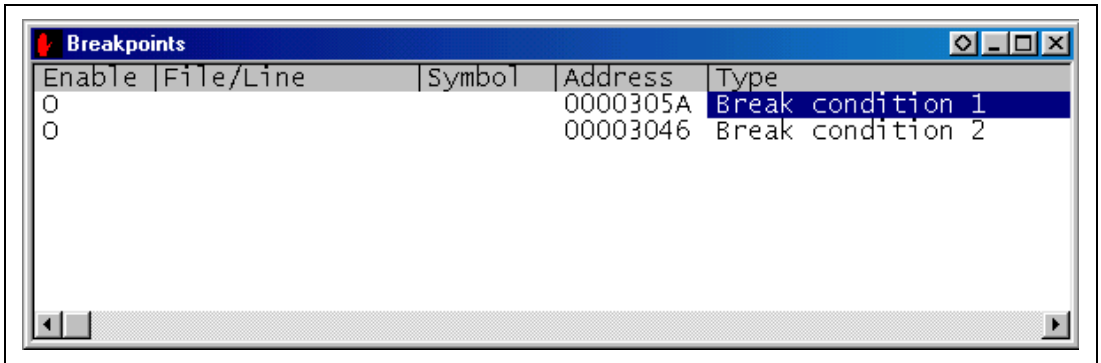


Figure 4.5 [Breakpoints] Window

Description:

The [Breakpoints] window displays breakpoint setting information. The items listed in the following tables are displayed.

Table 4.5 [Breakpoints] Window Display Items

Item	Description
[Enable]	Displays whether the break condition is enabled or disabled. BREAKPOINT: ● Break Condition: ○ (If the address is the same as the one that has been set to the BREAKPOINT, the mark is ●.)
[File/Line]	Displays the file name and line number where the breakpoint is set.
[Symbol]	Displays the symbol corresponding to the breakpoint address. If no symbol has been defined for the address, nothing is displayed.
[Address]	Displays the address where the breakpoint is set.
[Type]	Displays the break condition type as follows: Break Point: Software breakpoint Break Condition 1 to Break Condition 4, and Break Condition R: Hardware break condition

Note: The types of break conditions differ according to the product. For the specification of each product, refer to the online help.

The pop-up menu, which is opened by clicking the right mouse button, can be used to set, change, and clear breakpoints, and to enable or disable break conditions. The pop-up menu functions are described in the following table.

Table 4.6 [Breakpoints] Window Pop-up Menu Operation

Menu Name	Description
[Add]	Sets break conditions. Clicking this button will display the [Break] dialog box, enabling break conditions to be set.
[Edit]	Changes break conditions. Select break conditions to be changed and click this button. The break condition setting dialog box will be displayed, enabling the break condition to be changed.
[Disable] ([Enable])	Enables or disables break conditions. Select break conditions to be enabled or disabled and click this button.
[Delete]	Clears break conditions. Select break conditions to be cleared and click this button.
[Delete All]	Clears all break conditions.
[Go to Source]	Jumps to the address which sets the break in the [Source] window.

4.2.3 [Break] Dialog Box

Function:

This dialog box displays the break condition settings.

Window:

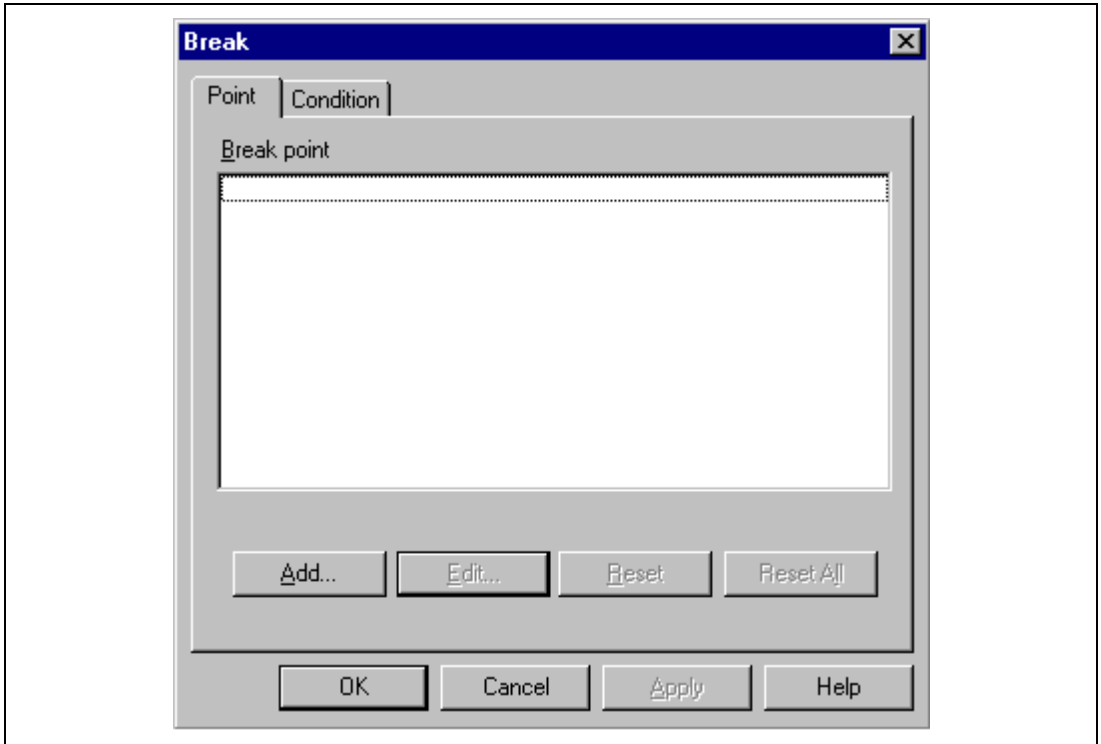


Figure 4.6 [Break] Dialog Box

Description:

The [Break] dialog box consists of the pages listed in table 4.7.

Table 4.7 [Break] Dialog Box Pages

Page Name	Description
[Point]	Displays software breakpoint settings.
[Condition]	Displays Break Condition settings.

The dialog boxes which set or modify break conditions can be displayed from the pages above.

Clicking the [OK] button (or [Close] button in some emulator products) will close this dialog box.

(1) [Point] Page ([Break] Dialog Box)

Function:

This page displays software breakpoint settings. In this page, software breakpoints can be set, changed, and cleared.

Window:

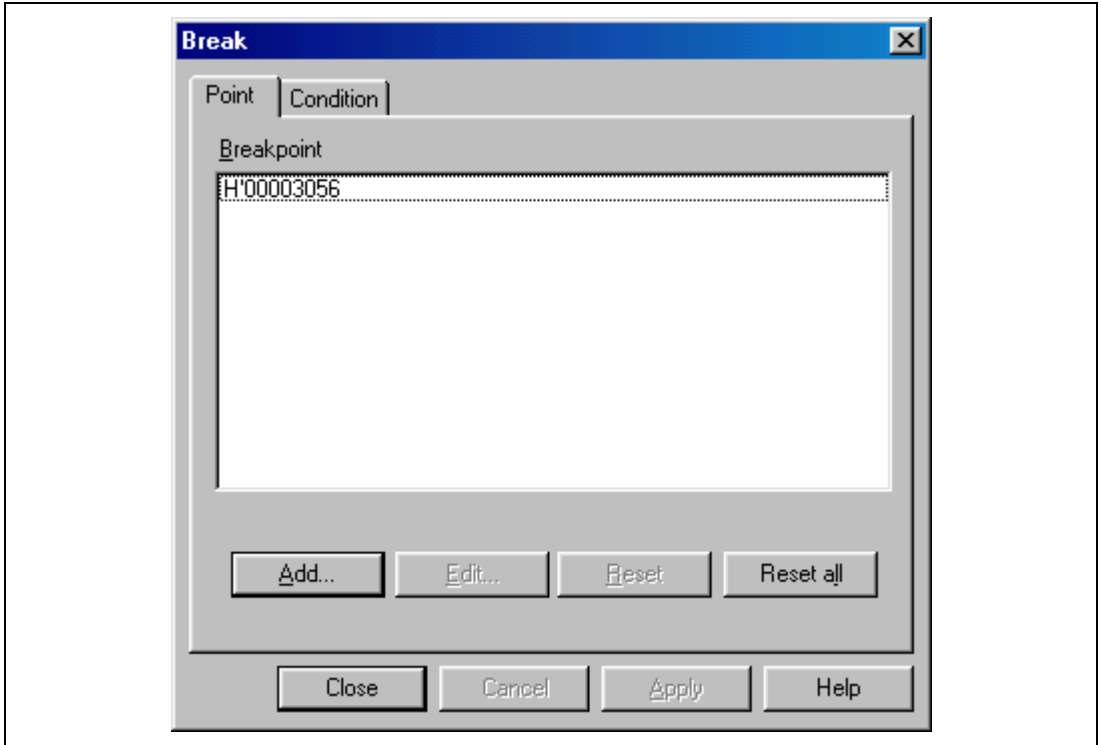


Figure 4.7 [Point] Page ([Break] Dialog Box)

Description:

Table 4.8 [Point] Page Options

Option	Description
[Break point] list box	Lists the software breakpoints currently being set. The display contents are <breakpoint address>.
[Add...] button	Sets software breakpoints. Clicking this button displays the [Break Point] dialog box.
[Edit...] button	Changes the software breakpoint selected in the [Break point] list box. Clicking this button displays the [Break Point] dialog box.
[Reset] button	Clears the software breakpoint selected in the [Break Point] list box.
[Reset All] button	Clears all software breakpoints displayed in the [Break Point] list box.

Related Commands:

BREAKPOINT command

BREAKPOINT_CLEAR command

BREAKPOINT_ENABLE command

BREAKPOINT_DISPLAY command

(2) [Condition] Page ([Break] Dialog Box)

Function:

This page displays the Break Condition settings. These conditions can also be set or cleared in this page.

Window:

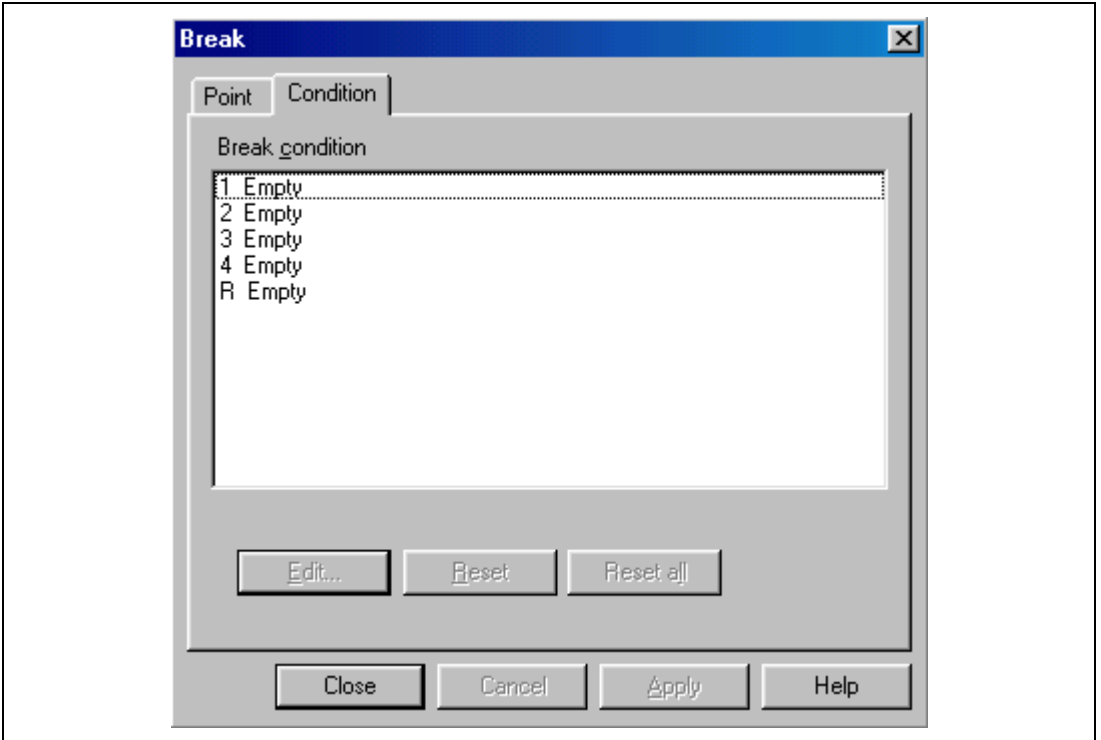


Figure 4.8 [Condition] Page ([Break] Dialog Box)

Note: The number of hardware break conditions differs according to the product. For the number that can be specified for each product, refer to the online help.

Description:

Table 4.9 [Condition] Page Options

Option	Description
[Break Condition] list box	Displays the Break Condition settings. The display at system initiation is as follows: When conditions are set, Enable is displayed. When no conditions are set, Empty is displayed. <ol style="list-style-type: none">1 Empty (setting of Break Condition 1)2 Empty (setting of Break Condition 2) <p style="text-align: center;">:</p>
[Edit...] button	Changes the Break Condition settings selected in the [Break Condition] list box. Clicking this button displays the [Break Condition] dialog boxes.
[Reset] button	Clears the Break Condition settings selected in the [Break Condition] list box.
[Reset All] button	Clears all Break Condition settings in the [Break Condition] list box.

Related Commands:

BREAKCONDITION_CLEAR command
BREAKCONDITION_DISPLAY command
BREAKCONDITION_ENABLE command
BREAKCONDITION_SET command

4.2.4 [Break Point] Dialog Box

Function:

This dialog box sets software breakpoints.

Window:

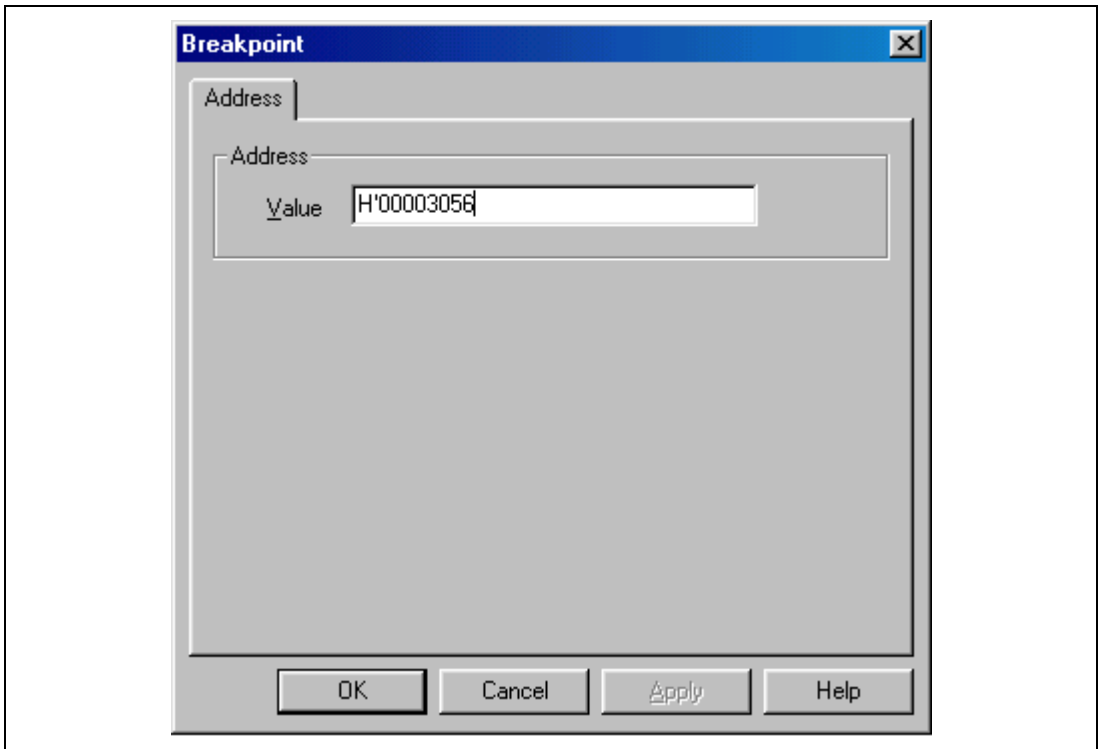


Figure 4.9 [Break Point] Dialog Box

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

Description:

The [Break Point] dialog box consists only of the [Address] page. This dialog box sets address conditions and address areas. The [Address] page options are as follows:

Table 4.10 [Address] Page Options

Option	Description
[Value] edit box	Sets a breakpoint address with a number or a symbol.
[Normal] radio button	Does not set an address area.*

Clicking the [OK] button enables breakpoints to be set. If the [Cancel] button is clicked, this dialog box is closed without setting breakpoints.

Related Commands:

BREAKPOINT command
BREAKPOINT_CLEAR command
BREAKPOINT_DISPLAY command
BREAKPOINT_SET command

4.2.5 [Break Condition] Dialog Box

Function:

This dialog box sets hardware break conditions.

Window:

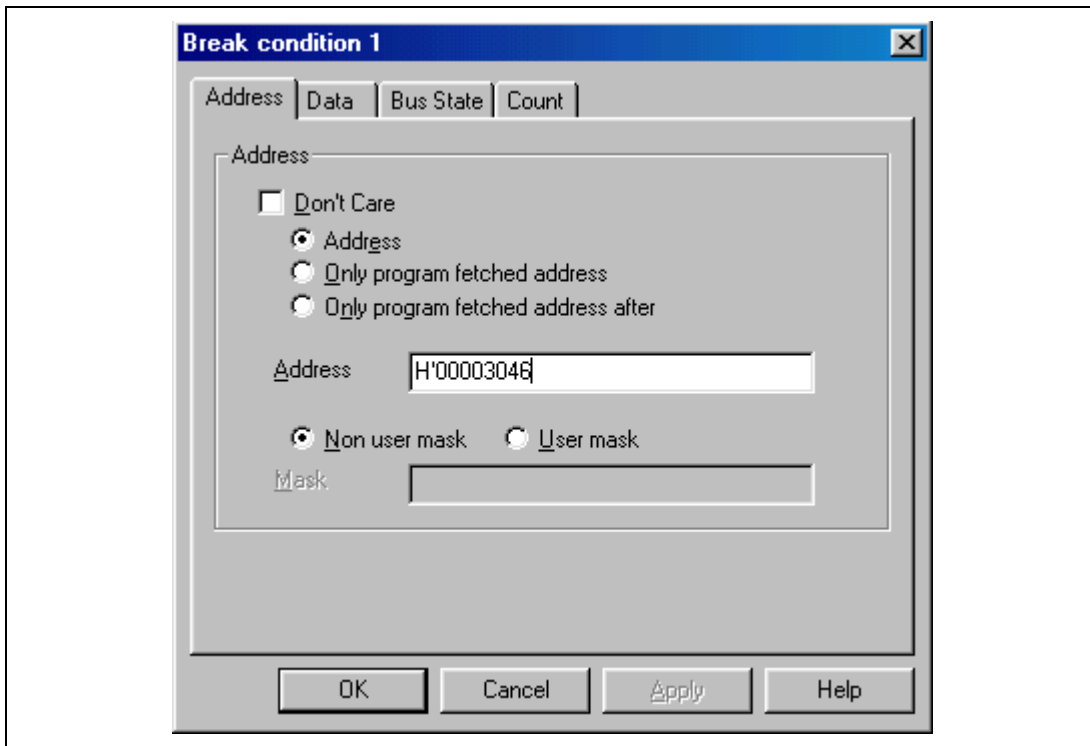


Figure 4.10 [Break Condition] Dialog Box

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

Description:

The [Break Condition] dialog box consists of multiple pages. A condition to halt the program can be set in each page.

Contents to be set by each page are described in section 4.2.6, [Break Condition] Dialog Box Pages.

Clicking the [OK] button sets the hardware break conditions. If the [Cancel] button is clicked, the dialog box is closed without setting the hardware break conditions.

Related Commands:

BREAKCONDITION_CLEAR command
BREAKCONDITION_DISPLAY command
BREAKCONDITION_ENABLE command
BREAKCONDITION_SET command

4.2.6 [Break Condition] Dialog Box Pages

Function:

The [Break Condition] dialog box pages allow a number of hardware break conditions to be set. Some functions may not be supported by some types of emulators. The setting conditions may differ from the dialog box name in table 4.11. For details, refer to section 6.5.2, Hardware Break Functions.

Table 4.11 Setting Conditions in [Break Condition] Dialog Boxes

Dialog Box	Type			
	Address Bus Condition	Data Bus Condition	Bus State and Read/Write Conditions	Count Condition
[Break Condition 1] dialog box	O	O	O	O
[Break Condition 2] dialog box	O	O	O	X
[Break Condition 3] dialog box	O	O	O	X
[Break Condition 4] dialog box	O	O	O	X
[Break Condition R] dialog box	O	X	X	X

Note: O: Can be set by checking the radio button in the dialog box.

X: Cannot be set in the dialog box.

Table 4.12 shows all the [Break Condition] dialog box pages.

Table 4.12 [Break Condition] Dialog Box Pages

Page Name	Function
[Address]	Sets the address conditions of Break Condition 1 to Break Condition 4, and Break Condition R.
[Data]	Sets the data conditions of Break Condition 1 to Break Condition 4.
[Bus State]	Sets the bus state conditions and read/write cycle conditions of Break Condition 1 to Break Condition 4.
[Count]	Sets the satisfaction count conditions of Break Condition 1. (Count condition is not displayed in the [Break Condition 2] to [Break Condition 4] dialog box pages.)

Note: This function differs according to the product. For the specifications of each product, refer to section 6.5.2, Hardware Break Functions, or to the online help.

(1) [Address] Page ([Break Condition] Dialog Box)

Function:

This page sets the address bus conditions.

Window:

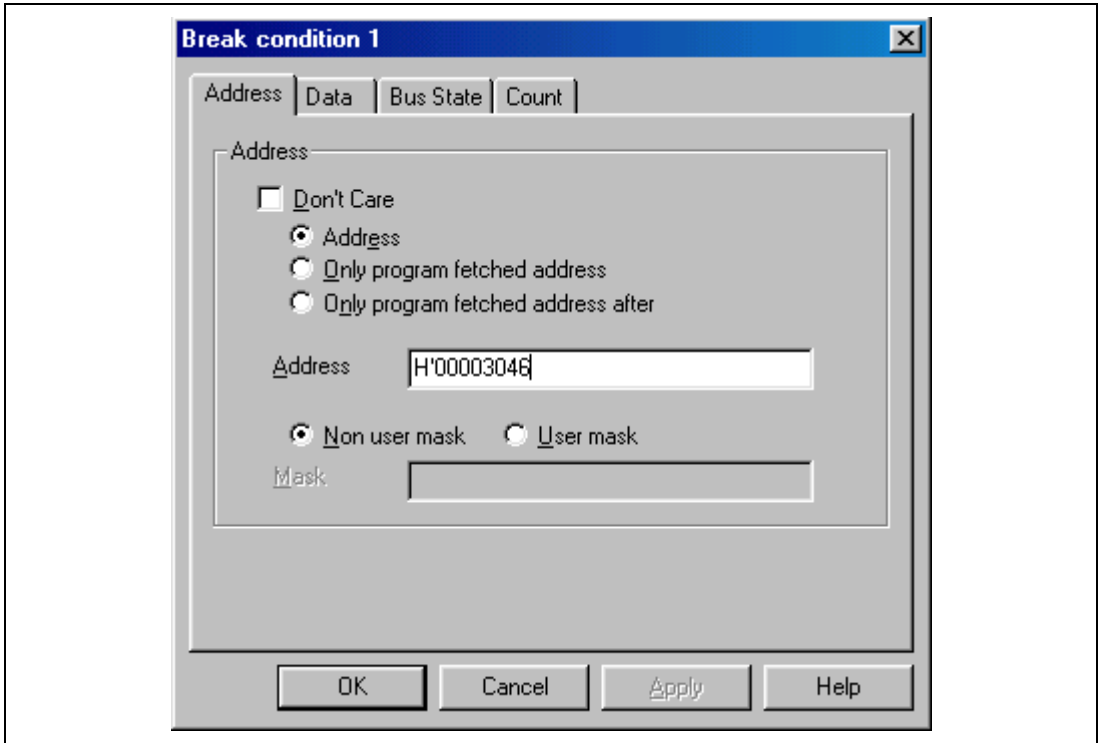


Figure 4.11 [Address] Page ([Break Condition 1] Dialog Box)

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

Description:

Table 4.13 [Address] Page Options (BREAK_CONDITION1 to 4)

Option	Description
[Only program fetched address] radio button	Sets a break before prefetched address execution as break conditions.
[Only program fetched address after] radio button	Sets a break after prefetched address execution as break conditions.
[Address] edit box	Sets the address bus value with a number or a symbol.
[Non user mask] radio button	Sets no mask conditions.
[User mask] radio button	Sets mask conditions.
[Mask] edit box	Sets the mask bits if [User mask] is selected. For masked bits, the break condition is satisfied regardless of the address values.

Note: This page is displayed when the conditions of Break Condition 1 to Break Condition 4 are set.

Table 4.14 [Address] Page Options (BREAK_CONDITION R)

Option	Description
[Don't Care] check box	Does not set address conditions.
[Address] radio button	Sets use of the normal address bus as break conditions.

A page name to be displayed and the contents of an option that can be set will change depending on the radio button selected.

Table 4.15 Address Options

Option	Description
[Address] radio button	All pages can be selected and masks can be set.
[Only program fetched address] radio button	The [Address] can be set; however, no mask can be set.
[Only program fetched address after] radio button	The [Address] can be set.

Note: This function differs according to the product. For the specifications of each product, refer to section 6.5.2, Hardware Break Functions, or to the online help.

(2) [Data] Page ([Break Condition] Dialog Box)

Function:

This page sets the data bus conditions.

Window:

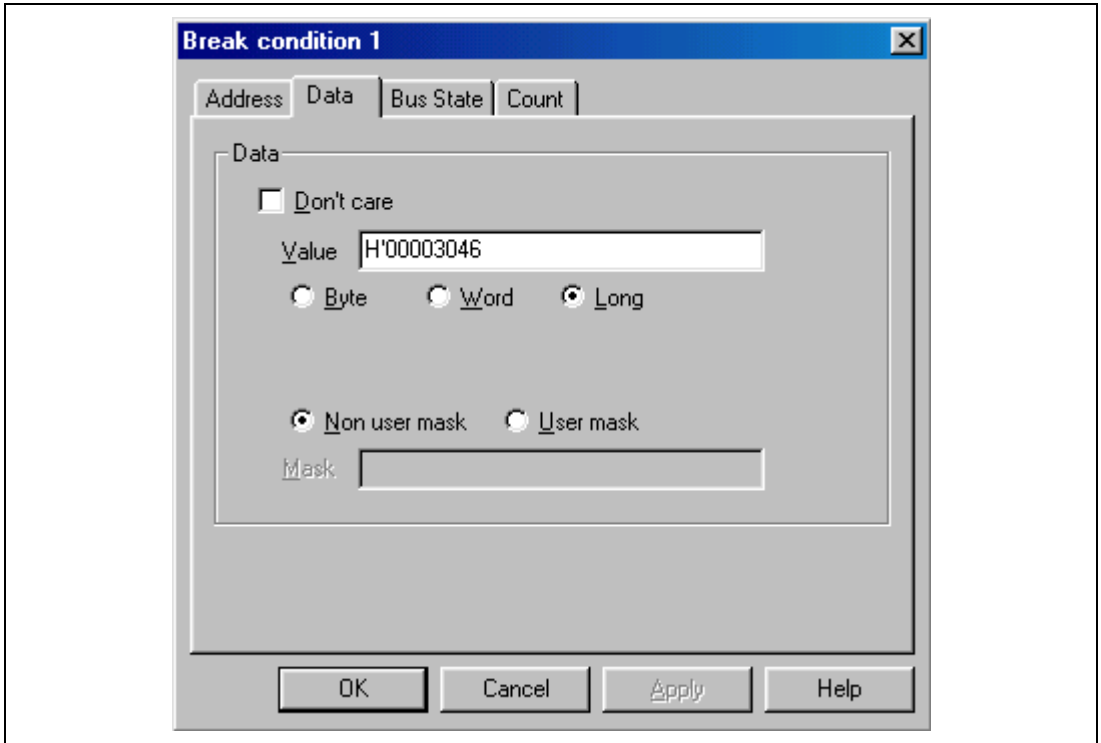


Figure 4.12 [Data] Page ([Break Condition 1] Dialog Box)

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

Description:**Table 4.16 [Data] Page Options**

Option	Description
[Don't Care] check box	Does not set data conditions.
[Value] edit box	Sets the data bus value with a number.
[Byte] radio button	Sets byte data access cycles.
[Word] radio button	Sets word data access cycles.
[Long] radio button	Sets longword data access cycles.
[Non user mask] radio button	Does not set mask conditions.
[User mask] radio button	Sets mask conditions.
[Mask] edit box	Sets the mask bits when [User mask] is selected. Mark a bit to be masked with *. For masked bits, the break conditions will be satisfied regardless of the data values.

Note: This page is displayed when the conditions of Break Condition 1 to Break Condition 4 are set.

(3) [Bus State] Page ([Break Condition] Dialog Box)

Function:

This page sets bus state conditions and read/write cycle conditions.

Window:

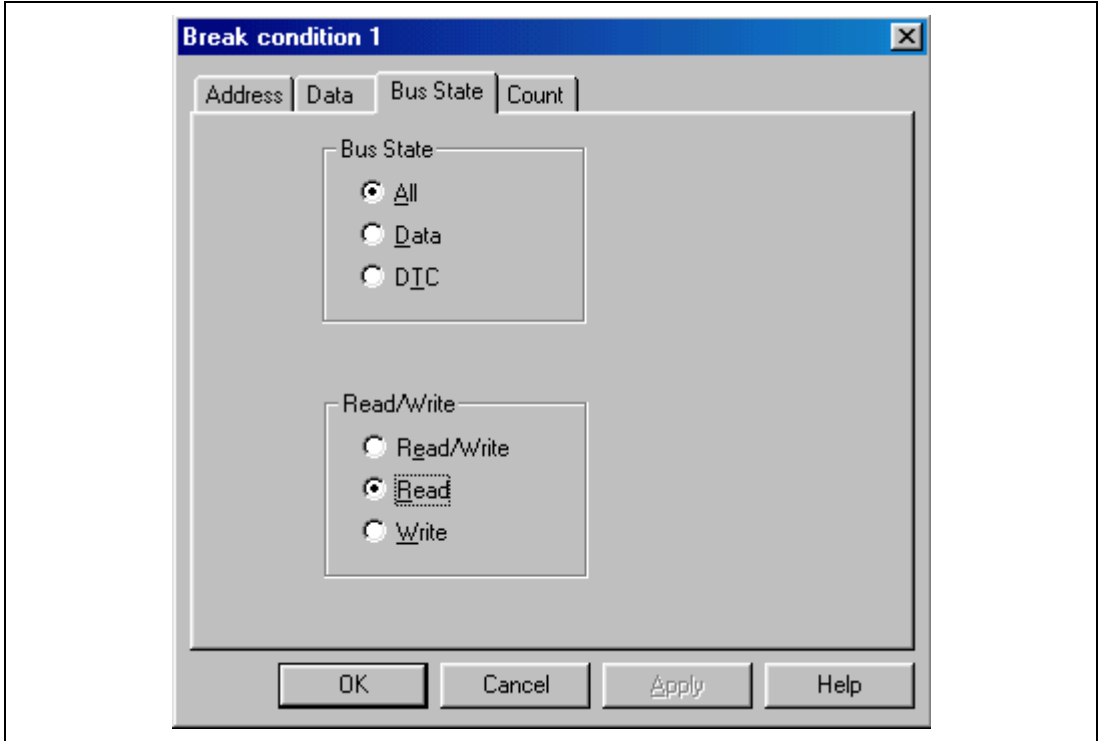


Figure 4.13 [Bus State] Page ([Break Condition] Dialog Box)

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the online help.

Description:

Table 4.17 [Bus State] Page Options

Group Box	Option	Description
[Bus State] group box	[All] radio button	Sets the bus state conditions as break conditions.
	[Data] radio button	Sets the execution cycle as break conditions.
	[DTC] radio button	Sets DTC cycles as a break condition.
[Read/Write] group box	[Read/Write] radio button	Sets the read/write cycle conditions as break conditions.
	[Read] radio button	Sets read cycles as break conditions.
	[Write] radio button	Sets write cycles as break conditions.

Note: This page is displayed when the conditions of Break Condition 1 to Break Condition 4 are set.

(4) [Count] Page ([Break Condition] Dialog Box)

Function:

This page sets the conditions for Break Condition 1.

Window:

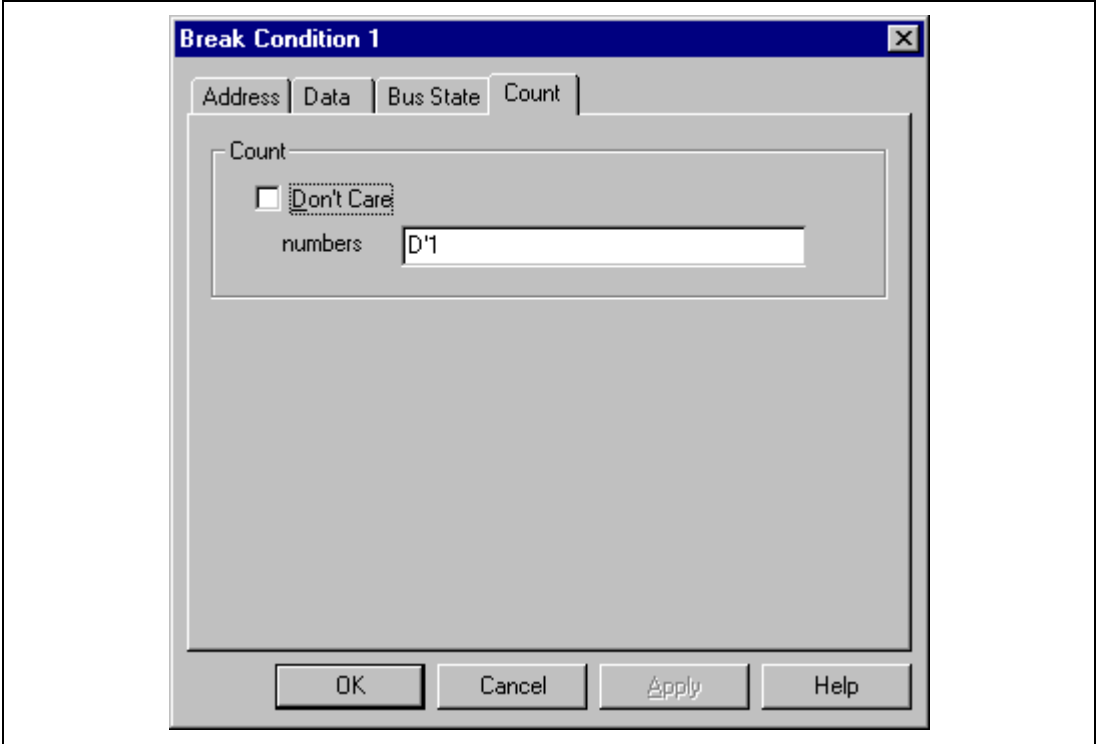


Figure 4.14 [Count] Page ([Break Condition] Dialog Box)

Table 4.18 [Count] Page Options

Option	Description
[Don't Care] check box	Sets no satisfaction count conditions.
Input area	Sets the satisfaction count as a break condition. The maximum count is 4,095. Breaks when the conditions set by the [Break Condition] dialog box for the specified times are satisfied. The default is D'1.

Note: Some products are not supported by this function. For the specifications of each product, refer to the online help.

4.2.7 [Trace] Window

Function:

This window displays the trace buffer contents.

Window:

No.	IP	TYPE	ADDR	MNEMONIC	OPERAND	Source
-000035	-D'000038	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000034	-D'000035	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000033	-D'000034	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000032	-D'000033	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000031	-D'000032	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000030	-D'000031	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000029	-D'000030	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000028	-D'000029	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000027	-D'000028	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000026	-D'000027	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000025	-D'000026	DESTINATION	000030E4	MOV.L	@R15,R3	for
-000024	-D'000025	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000023	-D'000024	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000022	-D'000023	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000021	-D'000022	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000020	-D'000021	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000019	-D'000020	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000018	-D'000019	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000017	-D'000018	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000016	-D'000017	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000015	-D'000016	DESTINATION	000030E4	MOV.L	@R15,R3	for
-000014	-D'000015	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000013	-D'000014	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000012	-D'000013	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000011	-D'000012	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000010	-D'000011	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000009	-D'000010	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000008	-D'000009	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000007	-D'000008	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000006	-D'000007	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000005	-D'000006	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
-000004	-D'000005	DESTINATION	0000308C	MOV.L	@R15,R3	for
-000003	-D'000004	DESTINATION	000030EC	MOV.L	@(H'04:4,R15),R3	
-000002	-D'000003	DESTINATION	00003098	MOV.L	@(H'08:4,R15),R3	
-000001	-D'000002	DESTINATION	000030F4	MOV.L	@R15,R3	for(i=
+000000	-D'000001	DESTINATION	0000303E	MOV.L	@(H'10:4,R15),R3	min = a[0];

Figure 4.15 [Trace] Window

Note: The types of information and the number of branch instructions differ according to the product. For the settings for each product, refer to the online help.

Description:

This window displays the trace buffer contents. The items listed in table 4.18 are displayed.

Table 4.19 [Trace] Window Display Items

Item	Description
[No.]	Displays the number in ascending order as the trace stop point is 0 (signed decimal).
[IP]	Displays the instruction pointer (signed decimal).
[TYPE]	For the branch instruction trace, displays the information type, that is, branch destination. DESTINATION: Branch destination
[ADDR]	For the branch instruction trace, displays the branch destination address.
[MNEMONIC]	Displays the execution instruction mnemonic.
[OPERAND]	Displays the execution instruction operand.
[Source]	Displays the C-source line of the address that the trace has been acquired.

The pop-up menu, opened by clicking the right mouse button, can be used to set, change, and clear trace conditions. For details, refer to the Hitachi Debugging Interface User's Manual.

Notes:

1. In some cases, the emulator address may be acquired by trace. In such a case, the following message will be displayed. Ignore this address because it is not a user program address.

*** EML ***

2. The [Halt] menu in the pop-up menu is active only when the [Trace] window is open during user program execution.

Related Command:

TRACE_DISPLAY command

4.2.8 [Trace Acquisition] Dialog Box

Function:

This dialog box sets trace acquisition conditions. When the [Acquisition] menu is selected from the pop-up menu, which is displayed by clicking the right mouse button in the [Trace] window, the [Trace Acquisition] dialog box is displayed.

Table 4.20 [Trace Acquisition] Dialog Box Page Options

Page Name	Description
[Trace Mode]	Sets the conditions of trace mode.

(1) [Trace Mode] Page ([Trace Acquisition] Dialog Box)

Function:

This page sets the conditions for trace mode.

Window:

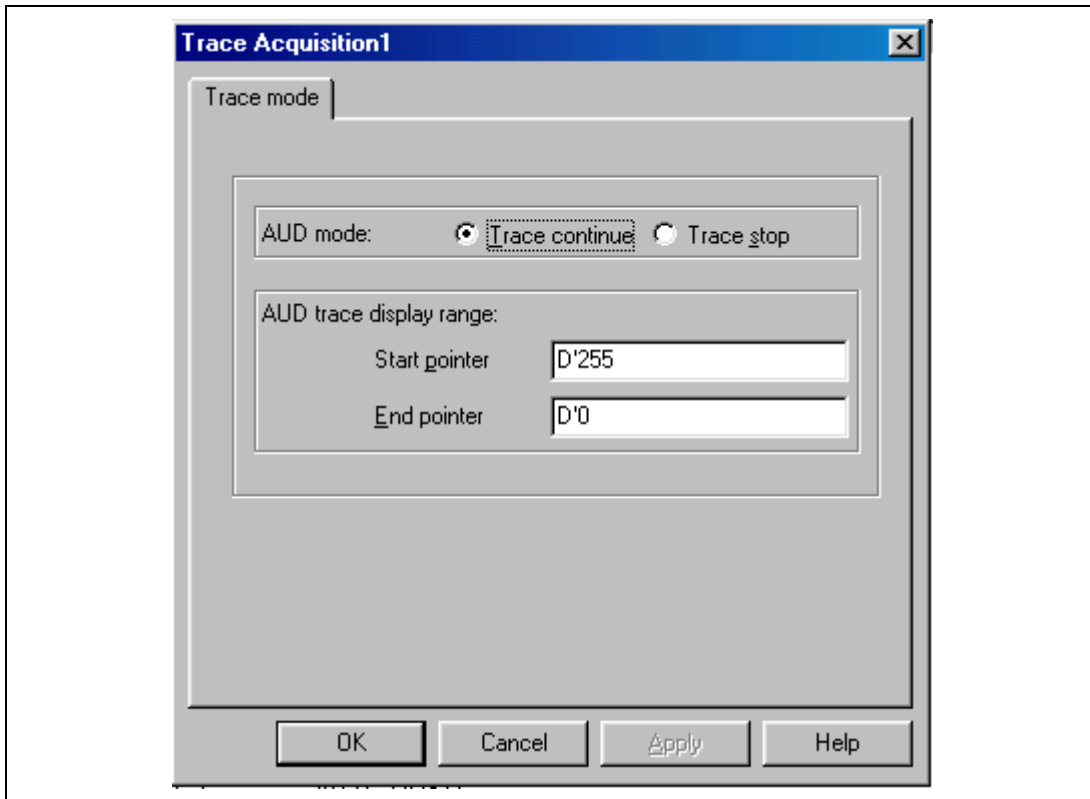


Figure 4.16 [Trace mode] Page ([Trace Acquisition] Dialog Box)

Note: This function differs according to the product. For the specifications of each product, refer to the section related to the trace functions in section 6, SHxxxx E10A Emulator Specifications, or to the online help.

4.2.9 [System Status] Window

Function:

This window lists information, such as conditions that have been set to the emulator and execution results.

Window:

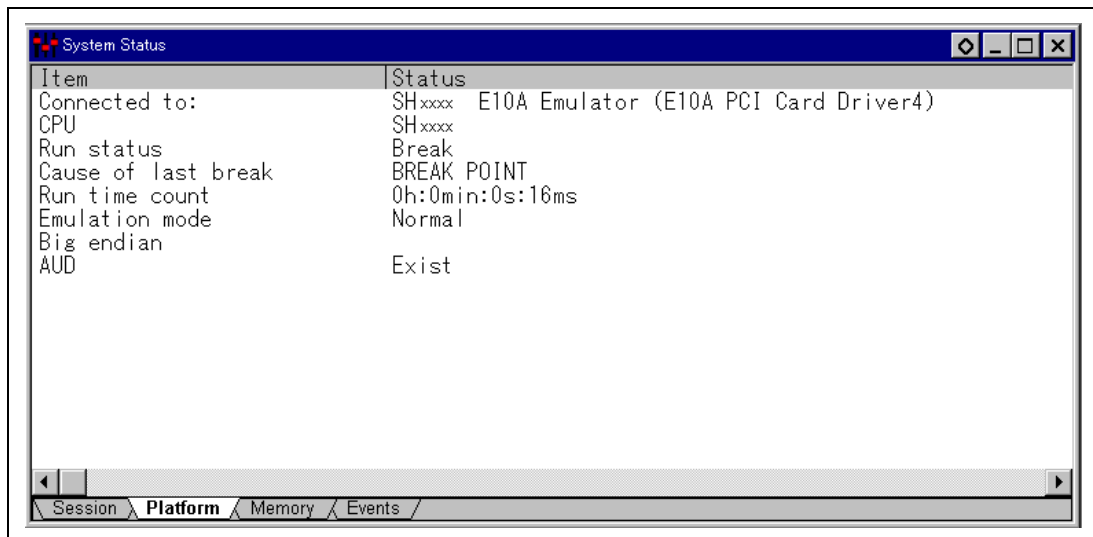


Figure 4.17 [System Status] Window

Note: The items that can be displayed in this window differ according to the product. For the items that can be displayed, refer to the online help.

The items listed in the following table are displayed in the [System Status] window.

Table 4.21 [System Status] Window Display Items

Page	Item	Description
[Session]	Target System	Always displays Connected.
	Session Name	Displays the session file name.
	Program Name	Displays the load module file name.
[Platform]	Connected To	Displays the name of the connected emulator and the selected driver name.
	CPU	Displays the target device name.
	Run status	Displays the execution status: RUNNING: Being executed Break: Stopped
	Cause of last break	Displays the cause of the emulator stopping at break. In this example, the cause of the stop is BREAK POINT.
	Run time count	Displays the program execution time. The display format is h: hours, min: minutes, s: seconds, and ms: milliseconds. In this example, 0h:0min:0s:16ms is displayed.
	Emulator mode	Displays the emulator operating mode (setting information for [Emulation Mode] of the [Configuration] dialog box).
	Big Endian/Little Endian	Displays the endian state (Big Endian or Little Endian). In this example, Big Endian is displayed.
	AUD	Displays whether the AUD function can be used. This item is displayed by the emulator with the AUD function.
[Memory]	Loaded Memory Areas	Displays the loaded area of the load module.
[Events]	Resources	Displays the usage states of BREAK POINT and Break Condition.

Section 5 Command-line Functions

5.1 Table and Symbol Description

This section describes the format used in section 5.2, Command Descriptions. The descriptions of some commands are given over two or more pages.

5.1.1 Format

The input format for each command is as follows. Characters shown in bold-italics are to be input.

[] : Parameters enclosed by [] can be omitted.

< > : Contents shown in < > are set.

< >=: The parameter to the left of the "=" sign is input in the format shown to the right.

| : This represents a non-exclusive selection.

|| : This represents an exclusive selection.

The command parameter details are described in the parameter table.

5.1.2 Parameter Input

Numerical Parameters:

A binary, octal, decimal, or hexadecimal value, a symbol, or a formula can be input. A symbol can contain up to 32 characters. Terms in a formula are separated with operators (such as + or -).

Keyword Parameters:

One of the bold characters given in the description column of the table can be input. If a character string not shown in the description is input, an error will occur.

Character-String Parameters:

Character-string parameters are used to input mask data or a file name. In the mask data, set a radix (H': hexadecimal or B': binary) at the top of a character string and set "*" at the digit to be masked.

5.1.3 Examples

These are actual input examples. For commands whose execution results in a specific display output, an example of the display is given.

5.1.4 Related Items

Related E10A HDI commands (abbreviations) and dialog boxes are shown. (Refer to section 4, Descriptions of Windows.)

5.2 Command Descriptions

The command list of the E10A emulator is shown below.

Table 5.1 E10A HDI Commands

No.	Command	Abb.	Function
1	AUD_MODE	AUM	Sets AUD trace conditions.
2	AUD_TRACE	AUT	Displays trace information.
3	BREAKCONDITION_ CLEAR	BCC	Clears hardware breakpoints that have been set.
4	BREAKCONDITION_ DISPLAY	BCD	Displays hardware breakpoints that have been set.
5	BREAKCONDITION_ ENABLE	BCE	Enables or disables hardware breakpoints that have been set.
6	BREAKCONDITION_ SET	BCS	Sets hardware breakpoints.
7	BREAKPOINT	BP	Sets software breakpoints.
8	BREAKPOINT_ CLEAR	BC	Clears software breakpoints that have been set.
9	BREAKPOINT_ DISPLAY	BD	Displays software breakpoints that have been set.
10	BREAKPOINT_ ENABLE	BE	Enables or disables software breakpoints that have been set.
11	DEVICE_TYPE	DE	Displays device type currently selected.
12	GO_OPTION	GP	Displays or sets the emulation mode during user program execution.
13	JTAG_CLOCK	JCK	Displays or sets a JTAG clock (TCK) frequency.
14	RAM_R	RR	Displays the specified address contents on the status bar during user program execution. Reads the memory by using the RAM monitor function.
15	RAM_W	RW	Writes the specified contents of the specified address to the memory. Writes the memory by using the RAM monitor function.
16	REFRESH	RF	Updates the HDI memory information to the latest contents.
17	RESTART	RST	Restarts the emulator.
18	STATUS	STS	Displays emulator state information.

Note: Support for these commands varies with the product. For the specifications of each product, refer to the online help.

5.2.1 AUD_MODE:AUM

Description:

Displays AUD trace acquisition conditions.

Format:

```
aum [<option1>]
```

```
<option1> = full<full>
```

Table 5.2 AUD_MODE Command Parameter

Parameter	Type	Description
<full>	Keyword	Continues or stops emulation when the trace memory is full. C: Always overwrites the oldest information to acquire the latest information. S: When the trace buffer memory is full, information acquisition stops.

Note: When <option1> is omitted, the current setting conditions are displayed.

Note: The parameters for this command differ according to the product. For the specifications of each product, refer to the online help.

Examples:

To set continue option:

```
>aum full c (RET)
```

To display settings:

```
>aum (RET)  
mode = Realtime trace, continue
```

Related Item:

[Trace Acquisition] dialog box

5.2.2 AUD_TRACE:AUT

Description:

Displays the trace information.

Format:

```
aut [<option1>] [<option2>]
```

<option1> = start<start_pointer>

<option2> = end<end_pointer>

Table 5.3 AUD_TRACE Command Parameter

Parameter	Type	Description
<start_pointer>	Numerical value (-n)	Start pointer value for trace display.
<end_pointer>	Numerical value (-m)	End pointer value for trace display.

- Notes:
1. In the PCMCIA card emulator, -D'16383 to D'0 can be set to the trace pointer. In the PCI card emulator, -D'65535 to D'0 can be set.
 2. When the internal trace is selected, the AUT command displays the information that has been acquired by using the AUD function.

Example:

To display trace information according to the information acquired during user program execution:

```
>AUD_TRACE (RET)
```

IP	TYPE	ADDR	MNEMONIC	OPERAND
-D'xxxxxxx	DESTINATION	01000020	MOV.L	R1, @R1
(a)	(b)	(c)	(d)	(e)

- (a) Instruction pointer (signed decimal)
- (b) Types of branch destination
DESTINATION: Branch destination
- (c) Address of instruction word
- (d) Instruction mnemonic
- (e) Instruction operand

Related Item:

[Trace] dialog box

5.2.3 BREAKCONDITION_CLEAR: BCC

Description:

Clears hardware breakpoints that have been set.

Format:

```
bcc [<channel>]
```

<channel> = channel <channel_number>

Table 5.4 BREAKCONDITION_CLEAR Command Parameter

Parameter	Type	Description
<channel number>	Numerical value	Specifies hardware break channel number

Note: When <channel> is omitted, all hardware breakpoints that have been set are cleared.

Examples:

To clear all hardware breakpoints:

```
>bcc (RET)
```

To clear a hardware breakpoint set at channel 2:

```
>bcc channel 2 (RET)
```

Related Items:

BCD, BCE, and BCS commands

[Breakpoints] window

[Break] and [Break Condition] dialog boxes

5.2.4 BREAKCONDITION_DISPLAY: BCD

Description:

Displays hardware breakpoints that have been set. The display contents include a hardware breakpoint channel number, enable or disable of the setting, and setting conditions.

Format:

```
bcd [<channel>]
```

<channel> = channel <channel_number>

Table 5.5 BREAKCONDITION_DISPLAY Command Parameter

Parameter	Type	Description
<channel_number>	Numerical value	Hardware breakpoint channel number

Note: When <channel> is omitted, all hardware breakpoints that have been set are displayed.

Examples:

To display all hardware breakpoint settings:

```
>bcd (RET)
```

Break Condition 1:Enable data 20 long

Break Condition 2:Disable address 126

To display the hardware breakpoint set at channel 1:

```
>bcd channel 1 (RET)
```

Break Condition 1:Enable data 20 long

Note: The items displayed with this command differ according to the product. For the display specifications of each product, refer to the online help.

Related Items:

BCC, BCE, and BCS commands

[Breakpoints] window

[Break] and [Break Condition] dialog boxes

5.2.5 BREAKCONDITION_ENABLE: BCE

Description:

Enables or disables hardware breakpoints that have been set.

Format:

```
bce [<channel>] <mode>
```

<channel> = channel <channel_number>

Table 5.6 BREAKCONDITION_ENABLE Command Parameters

Parameter	Type	Description
<channel_number>	Numerical value	Specifies hardware break channel number.
<mode>	Keyword	Enables or disables hardware break settings. Set either of the following keywords: enable : Enables hardware break settings. disable : Disables hardware break settings.

Note: When <channel> is omitted, all hardware breakpoints that have been set are enabled or disabled.

Examples:

To enable all hardware breakpoints:

```
>bce enable (RET)
```

To disable the hardware breakpoints set at channel 1:

```
>bce channel 1 disable (RET)
```

Related Items:

BCC, BCD, and BCS commands

[Breakpoints] window

[Break] and [Break Condition] dialog boxes

5.2.6 BREAKCONDITION_SET: BCS

Description:

Sets hardware breakpoints.

Note: The function will differ according to the devices used. For functions of each emulator product, refer to section 6.5.2, Hardware Break Functions.

Format:

```
bcs <channel> <option> [ <option> ... ]
```

```
<channel>    = channel <channel_number>
<option>     = [ <addropt> | <dataopt> | <r/wopt> | <accessopt> ] [ <countopt> ]
<addropt>    = address <address> [ <addrcycle> ] || address mask <maskdata>
               <addrcycle>
<dataopt>    = data <data> <datawidth> ||
               data mask <maskdata> <datawidth>
<r/wopt>     = direction <r/w>
<accessopt>  = access <access>
<countopt>   = count <count>
```

Table 5.7 BREAKCONDITION_SET Command Parameters

Parameter	Type	Description
<channel_number>	Numerical value	<p>Hardware break condition channel number</p> <p>Specifiable options change depending on the channel number. For details, refer to section 6.5.2, Hardware Break Functions.</p> <p>1: <addropt>, <dataopt>, <r/wopt>, <accessopt>, and <countopt> can be set.</p> <p>2 to 4: <addropt>, <dataopt>, <r/wopt>, and <accessopt> can be set.</p> <p>R: address <address> can be set.</p>
<address>	Numerical value	Virtual address as an address bus value
<addrcycle>	Keyword	<p>Address bus access conditions for program fetch cycles</p> <p>Set either of the following keywords:</p> <p>pc: Breaks before the address set by the <address> parameter is executed. When this keyword is set, only <addropt> cannot be set as a condition. In addition, when pc is set, the <maskdata> parameter cannot be set.</p> <p>pcafter: Breaks after the address set by the <address> parameter is executed. When this keyword is set, only <addropt> cannot be set as a condition. When pcafter is not set, the address bus during data access cycles and program fetch cycles is targeted as the access condition.</p>
<maskdata>	Character string	<p>Mask specification for desired bits in the data</p> <p>Set a radix (H' for hexadecimal or B' for binary) at the top of a character string and set * in the bit to the masked. Conditions are satisfied regardless of the values of masked bits.</p>
<data>	Numerical value	Data bus value

Table 5.7 BREAKCONDITION_SET Command Parameters (cont)

Parameter	Type	Description
<datawidth>	Keyword	Data bus access conditions Set one of the following keywords: byte : byte access word : word access long : longword access
<r/w>	Keyword	Bus cycle read/write conditions Set either of the following keywords: read : read cycles write : write cycles
<access>	Keyword	Bus cycle access type dat : execution cycles
<count>	Numerical value	Set satisfaction count from 1 to H'FFFF

Note: The parameters for this command differ according to the product. For the specifications of each product, refer to the online help.

Examples:

To set the following conditions for channel 1 hardware breakpoint:

<addropt> item: An address bus value of H'1000,
<dataopt> item: D0 bit of the byte access data is 0,
<r/wopt> item: write cycle.

```
>bcs channel 1 address H'1000 data mask B'*****0 byte  
direction write (RET)
```

To set the following conditions for channel 2 hardware breakpoint:

<addropt> item: Sets an address bus value of H'10000 during the program fetch cycles,
and breaks before execution.

```
>bcs channel 2 address H'10000 pc(RET)
```

To set the following conditions for channel 2 hardware breakpoint:

<accessopt> item: Execution cycles,
<r/wopt> item: Read cycles.

```
>bcs channel 2 access dat direction read (RET)
```

Related Items:

BCC, BCD, and BCE commands

[Breakpoints] window

[Break] and [Break Condition] dialog boxes

5.2.7 BREAKPOINT: BP

Description:

Sets software breakpoints.

Note: The function will differ according to the devices used.

Format:

```
bp <address>
```

Table 5.8 BREAKPOINT Command Parameters

Parameter	Type	Description
<address>	Numerical value	Breakpoint address When an odd address is set, the address is rounded down to an even address.

Note: The parameters for this command differ according to the product. For the specifications of each product, refer to the online help.

Examples:

To set a software breakpoint at address H'12C8:

```
>bp H'12C8 (RET)
```

Related Items:

BC, BD, and BE commands

[Breakpoints] window

[Break] dialog box

5.2.8 BREAKPOINT_CLEAR: BC

Description:

Clears software breakpoints that have been set.

Format:

```
bc [<address>]
```

Table 5.9 BREAKPOINT_CLEAR Command Parameters

Parameter	Type	Description
<address>	Numerical value	Breakpoint address

Notes: When no parameters are set, all software breakpoints are cleared.

Note: The parameters for this command differ according to the product. For the specifications of each product, refer to the online help.

Examples:

To clear all breakpoints:

```
>bc (RET)
```

To clear software breakpoints whose address value is H'1000:

```
>bc H'1000 (RET)
```

Related Items:

BP, BD, and BE commands

[Breakpoints] window

[Break] dialog box

5.2.9 BREAKPOINT_DISPLAY: BD

Description:

Displays software breakpoints that have been set.

Format:

```
bd
```

Table 5.10 BREAKPOINT_DISPLAY Command Parameter

Parameter	Type	Description
None		

Example:

To display the software breakpoints that have been set:

```
>bd (RET)
```

```
H'00000110 enable  
H'0000011c disable  
H'00000250 enable
```

Note: The items displayed with this command differ according to the product. For the display specifications of each product, refer to the online help.

Related Items:

BP, BC, and BE commands

[Breakpoints] window

[Break] dialog box

5.2.10 BREAKPOINT_ENABLE: BE

Description:

Enables or disables software breakpoints that have been set.

Format:

```
be <address> <mode>
```

Table 5.11 BREAKPOINT_ENABLE Command Parameters

Parameter	Type	Description
<address>	Numerical value	Breakpoint address
<mode>	Keyword	Enables or disables breakpoints. Set either of the following keywords: enable : Enables breakpoints. disable : Disables breakpoints.

Examples:

To enable a software breakpoint at physical address H'1002:

```
>be H'1002 enable (RET)
```

- Notes:**
- 1. The parameters for this command differ according to the product. For the specifications of each product, refer to the online help.**
 - 2. The items displayed with this command differ according to the product. For the display specifications of each product, refer to the online help.**

Related Items:

BC, BD, and BP commands

[Breakpoints] window

[Break] dialog box

5.2.11 DEVICE_TYPE: DE

Description:

Displays the currently selected device.

Format:

de

Table 5.12 DEVICE_TYPE Command Parameter

Parameter	Type	Description
None		

Example:

To display the currently selected device:

```
>de (RET)
```

```
Current device = SHxxxx
```

5.2.12 GO_OPTION: GP

Description:

Displays or sets the emulation mode.

Format:

Displays emulation mode.

gp

Sets emulation mode.

gp <eml_opt>

<eml_opt> = eml_mode <eml_mode>

Table 5.13 GO_OPTION Command Parameter

Parameter	Type	Description
<eml_mode>	Keyword	<p>Specifies the emulation mode.</p> <p>normal: Normal execution</p> <p>sequence2-1: Stops the user program only when the conditions are satisfied in the order of hardware breakpoints 2 to 1. Hardware breakpoints 1 and 2 must be set.</p> <p>sequence3-2-1: Stops the user program only when the conditions are satisfied in the order of hardware breakpoints 3 to 1. Hardware breakpoints 1 to 3 must be set.</p> <p>sequence4-3-2-1: Stops the user program only when the conditions are satisfied in the order of hardware breakpoints 4 to 1. Hardware breakpoints 1 to 4 must be set.</p> <p>no_break: Makes software breakpoints and hardware breakpoints temporarily invalid and executes the user program.</p>

Examples:

To display the currently set emulation mode for user program execution:

```
>gp (RET)
```

```
Emulator execution mode = Sequential break Condition 2-1
```

To set the normal emulation mode for user program execution:

```
>gp eml_mode normal (RET)
```

- Notes:
1. The sequential break function differs according to emulator products. For details, refer to online help.
 2. The parameters for this command differ according to the product. For the specifications of each product, refer to the online help.

Related Items:

BCS and BS commands,
[Breakpoints] window,
[Break], [Break Condition], and [Configuration] dialog boxes

5.2.13 JTAG_CLOCK: JCK

Description:

Displays or sets the JTAG clock (TCK) frequency.

Format:

Displays the JTAG clock (TCK) frequency.

```
jck
```

Sets the JTAG clock (TCK) frequency.

```
jck <jck_opt>
```

Table 5.14 JTAG_CLOCK Command Parameter

Parameter	Type	Description
<jck_opt>	Numerical value	Sets one of the JTAG clock (TCK) frequency. (PCMCIA used: 3.75 MHz, 7.5 MHz, or 15 MHz) 3: 3.75 MHz 7: 7.5 MHz 15: 15 MHz (PCI used: 4.125 MHz, 8.25 MHz, or 16.5 MHz) 4: 4.125 MHz 8: 8.25 MHz 16: 16.5 MHz

Note: The range of frequencies that the Hitachi-UDI operates at differs according to the devices used. For details, refer to section 6.5.4, Notes on Using the JTAG Clock (TCK).

Examples:

(when PCMCIA used):

To set the JTAG clock (TCK) frequency:

```
> jck 15 (RET)
```

```
JTAG Clock 15MHz
```

To display the JTAG clock (TCK) frequency:

```
> jck (RET)
```

```
JTAG Clock 15MHz
```

(when PCI used):

To set the JTAG clock (TCK) frequency:

```
> jck 16 (RET)
```

```
JTAG Clock 16.5MHz
```

To display the JTAG clock (TCK) frequency:

```
> jck (RET)
```

```
JTAG Clock 16.5MHz
```

5.2.14 REFRESH: RF

Description:

Updates the HDI memory information.

Format:

rf

Table 5.15 REFRESH Command Parameter

Parameter	Type	Description
None		

Example:

To update the HDI memory information:

```
>rf (RET)
```

5.2.15 RESTART: RST

Description:

Restarts the emulator. The settings of breakpoints or trace acquisition conditions are not reset here.

Format:

`rst`

Table 5.16 RESTART Command Parameter

Parameter	Type	Description
None		

Example:

To restart the emulator:

```
>rst (RET)
```

5.2.16 STATUS: STS

Description:

Displays status information of the emulator.

Format:

sts

Table 5.17 STATUS Command Parameter

Parameter	Type	Description
None		

Example:

To display status information of the emulator:

```
>sts (RET)
Emulator Status
Connected to:      SHxxxxx E10A Emulator (E10A PC Card Driver)
CPU               SHxxxxx
Run status        Break
Cause of last break BREAK POINT
Run time count    0h:0min:0s:14ms
Emulator mode     Normal
Big endian
AUD              Exist
```

Note: The items displayed with this command differ according to the product. For the display specifications of each product, refer to the online help.

5.3 RAM Monitor Commands

5.3.1 RAM_R: RR

Description:

Displays the specified address contents on the status bar during user program execution. The memory can be read by using the RAM monitor function (a maximum of three addresses are displayed).

Format:

```
RR  [[[<option1> [<option2>]] [<option1> [<option2>]] [<option1>
[<option2>]]][<init>]]
```

<option1> = address <address>

<option2> = size <size>

<init> = <init>

Table 5.18 RAM_R Command Parameter

Parameter	Type	Description
<address>	Numerical value	Sets the address to be displayed on the status bar.
<size>	Keyword	Sets the size to be read. B : Byte size W : Word size L : Longword size
<init>	Keyword	Clears the setting address. I : Clears the setting address.

Note: When only <option2> is omitted, the address is read in a byte size. When <option1>, <option2>, and <init> are omitted, the current setting address is displayed.

- Notes:
1. While the memory is read by using the RAM monitor function at a fixed interval, no trace is acquired.
 2. This command can be accepted during user program execution.

Examples:

To set the address to be displayed on the status bar:

```
>rr address 00000100 size B address 00000102 size L
```

To display the current setting address:

```
>rr (RET)  
address 00000100 size B address 00000102 size L
```

To clear settings:

```
>rr i (RET)
```

5.3.2 RAM_W: RW

Description:

Writes the specified contents in the specified address to the memory. Memory can be written by using the RAM monitor function (a maximum of three addresses are written).

Format:

```
RW <option1> <option2> [<option3>] [<option1> <option2> [<option3>]]  
[<option1> <option2> [<option3>]]
```

<option1> = address <address>

<option2> = data <data>

<option3> = size <size>

Table 5.19 RAM_W Command Parameter

Parameter	Type	Description
<address>	Numerical value	Sets the address to be written.
<data>	Numerical value	Sets the data to be written.
<size>	Keyword	Sets the size to be written. B : Byte size W : Word size L : Longword size

Note: When <option3> is omitted, the address is written in a byte size.

- Notes:
1. While the memory is written by using the RAM monitor function, no trace is acquired.
 2. This command can be accepted during the user program execution.

Example:

To write memory by using the RAM monitor function:

```
>rr address 00000100 data 0009 size W address 00000200 data  
FF000000 size L
```


Section 6 SH7047F E10A Emulator Specifications

6.1 Overview of the SH7047F E10A Emulator

In the SH7047F E10A emulator, the SH7047F enables realtime emulation. In addition, realtime data tuning with the AUD function is also enabled.

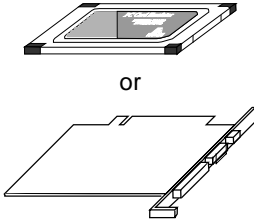
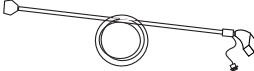
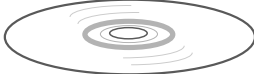
The supported operating modes are as follows:

- MCU expansion mode 2
- Single-chip mode

Note: When the SH7047F E10A emulator is used, set the FWP pin to 0 (low). To select MCU expansion mode 2, select the clock mode in FWP = 0 and MD3,2, and set MD1 = 1 and MD0 = 0. To select the single-chip mode, select the clock mode in FWP = 0 and MD3,2, and set MD1 = 1 and MD0 = 1.

Table 6.1 lists the components of the emulator.

Table 6.1 Components of the Emulator (HS7047KCM01H, HS7047KCM02H, HS7047KCI01H, or HS7047KCI02H)

Classification	Component	Appearance	Quantity	Remarks
Hard-ware	Card emulator (HS7047KCM01H, HS7047KCM02H, HS7047KCI01H, or HS7047KCI02H)		1	HS7047KCM01H (PCMCIA: 14-pin type): Depth: 85.6 mm, Width: 54.0 mm, Height: 5.0 mm, Mass: 27.0 g
				HS7047KCM02H (PCMCIA: 36-pin type): Depth: 85.6 mm, Width: 54.0 mm, Height: 5.0 mm, Mass: 28.0 g
				HS7047KCI01H (PCI: 14-pin type): Depth: 144.0 mm, Width: 105.0 mm, Mass: 93.0 g
				HS7047KCI02H (PCI: 36-pin type): Depth: 122.0 mm, Width: 96.0 mm, Mass: 90.0 g
User system interface cable			1	HS7047KCM01H (PCMCIA: 14-pin type): Length: 80 cm, Mass: 45.0 g
				HS7047KCM02H (PCMCIA: 36-pin type): Length: 30 cm, Mass: 55.0 g
				HS7047KCI01H (PCI: 14-pin type): Length: 150 cm, Mass: 86.0 g
				HS7047KCI02H (PCI: 36-pin type): Length: 80 cm, Mass: 69.0 g
Soft-ware	SH7047F E10A emulator setup program, SH7047F E10A Emulator User's Manual, and Hitachi Debugging Interface User's Manual		1	HS7047KCM01SR, HS7047KCM01HJ, HS7047KCM01HE, HS6400DIIW5SJ, and HS6400DIIW5SE (provided on a CD-R)

6.2 Pin Arrangement of the Hitachi-UDI Port Connector

Figure 6.1 shows the pin arrangement of the Hitachi-UDI port connector (14 pins).

CAUTION

Note that the pin number assignment of the Hitachi-UDI differs from that of the connector manufacturer.

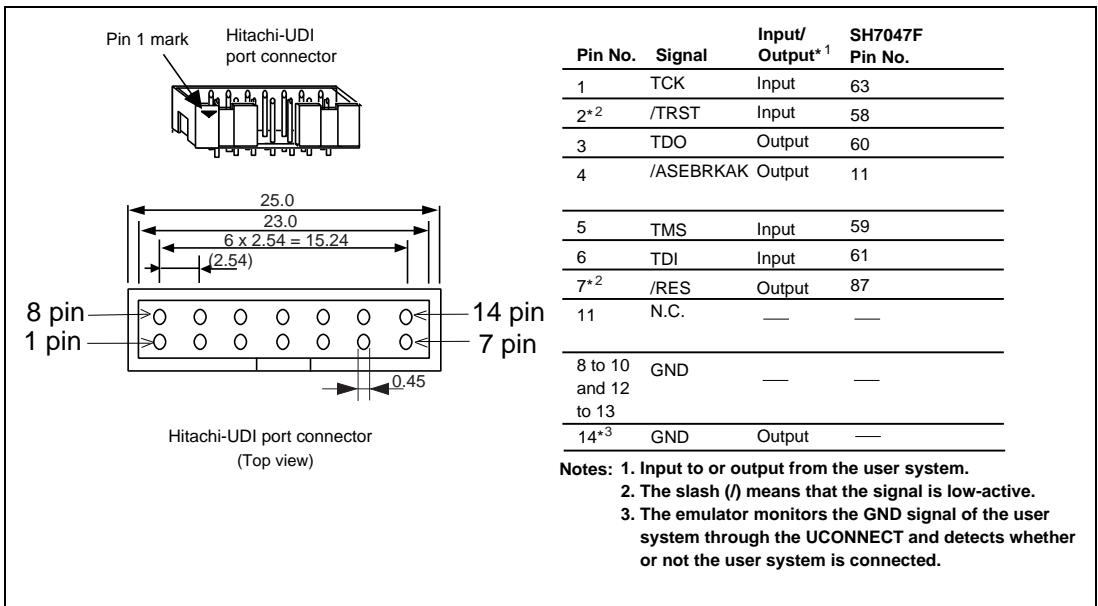


Figure 6.1 Pin Arrangement of the Hitachi-UDI Port Connector (14 Pins)

- Notes: 1. Handling of the TCK, TMS, TDI, TDO, /TRST, and /ASEBRKAK pins depends on the use conditions of the Hitachi-UDI as follows:
- When the user system is used by connecting the emulator, the TCK, TMS, TDI, TDO, /TRST, and /ASEBRKAK pins must be pulled up by a resistance of several kilo-ohms. The /DBGMD pin must be grounded.
 - When the user system is independently used without using the emulator and Hitachi-UDI, the TCK, TMS, TDI, TDO, /TRST, and /ASEBRKAK pins must be pulled up by a resistance of several kilo-ohms. The /DBGMD pin must be also pulled up by a resistance of several kilo-ohms.
2. The /RESET signal in the user side is input to the 87 pin of the SH7047F. Connect this pin to the Hitachi-UDI port connector as the output from the user system.

Figure 6.2 shows the pin arrangement of the Hitachi-UDI port connector (36 pins).

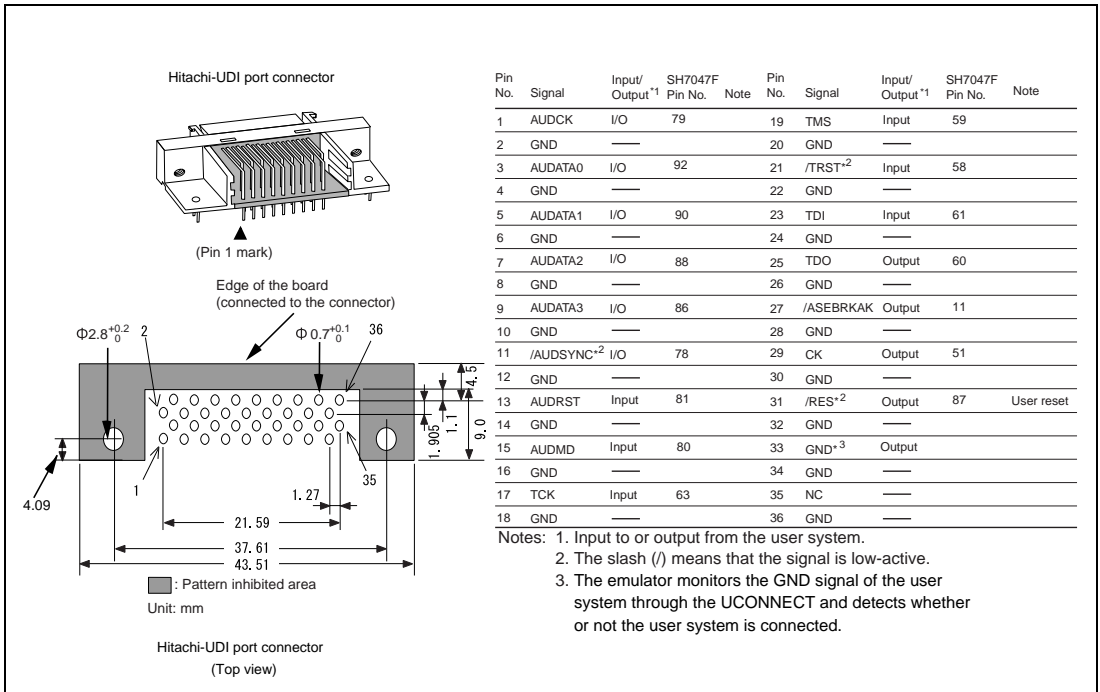


Figure 6.2 Pin Arrangement of the Hitachi-UDI Port Connector (36 Pins)

Notes: 1. Handling of the TCK, TMS, TDI, TDO, /TRST, and /ASEBRKAK pins depends on the use conditions of the Hitachi-UDI as follows:

(a) When the user system is used by connecting the emulator, the TCK, TMS, TDI, TDO, /TRST, and /ASEBRKAK pins must be pulled up by a resistance of several kilo-ohms. The /DBGMD pin must be grounded.

The AUDATA3 to AUDATA0, /AUDSYNC, and AUDMD pins must be pulled up by a resistance of several kilo-ohms. The resistor of the AUDCK pin must be terminated (pulled up and pulled down by a resistance of several kilo-ohms).

The AUDRST pin must be pulled up by a resistance of several kilo-ohms (5 kilo-ohms or less).

(b) When the user system is used independently without using the emulator and Hitachi-UDI, the TCK, TMS, TDI, TDO, /TRST, and /ASEBRKAK pins must be pulled up by a resistance of several kilo-ohms. The /DBGMD pin must be also pulled up by a resistance of several kilo-ohms. The AUDCK, AUDATA3 to AUDATA0, /AUDSYNC, and AUDMD pins must be pulled up by a resistance of several kilo-ohms.

The AUDRST pin must be pulled down by a resistance of several kilo-ohms.

2. The /RESET signal in the user side is input to the 87 pin of the SH7047F. Connect this pin to the Hitachi-UDI port connector as the output from the user system.
3. The CK pin is used by the emulator. When the CK pin is directly connected to the interface, operation may be affected by the load capacity and reflection of the interface cable. When the emulator is used, it is recommended that the CK pin between the user interface connector (36 pins) and the SH7047F is connected via the buffer (e.g., 74LVC125) as shown in figure 6.3. The buffer voltage should be Vcc.

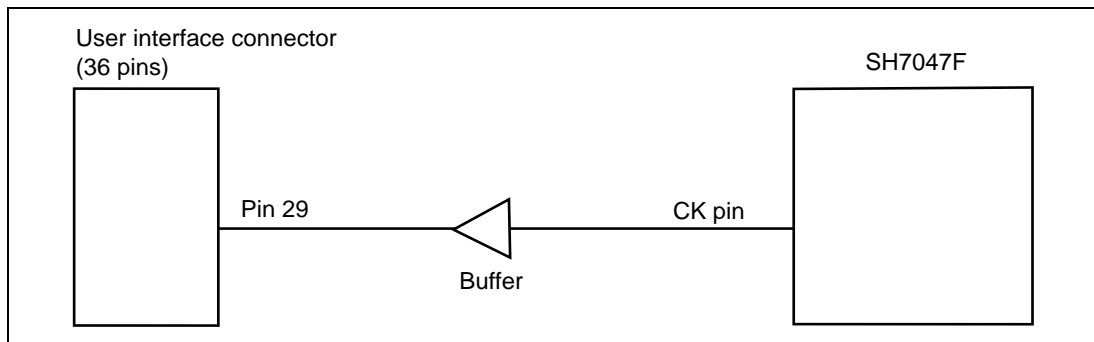


Figure 6.3 Circuit between User Interface Connector (36 Pins) and SH7047F

6.3 User System Interface Circuit

The emulator is connected to the user system via the user interface cable. Figure 6.4 shows the user system interface circuit of the emulator (HS7047KCM01H).

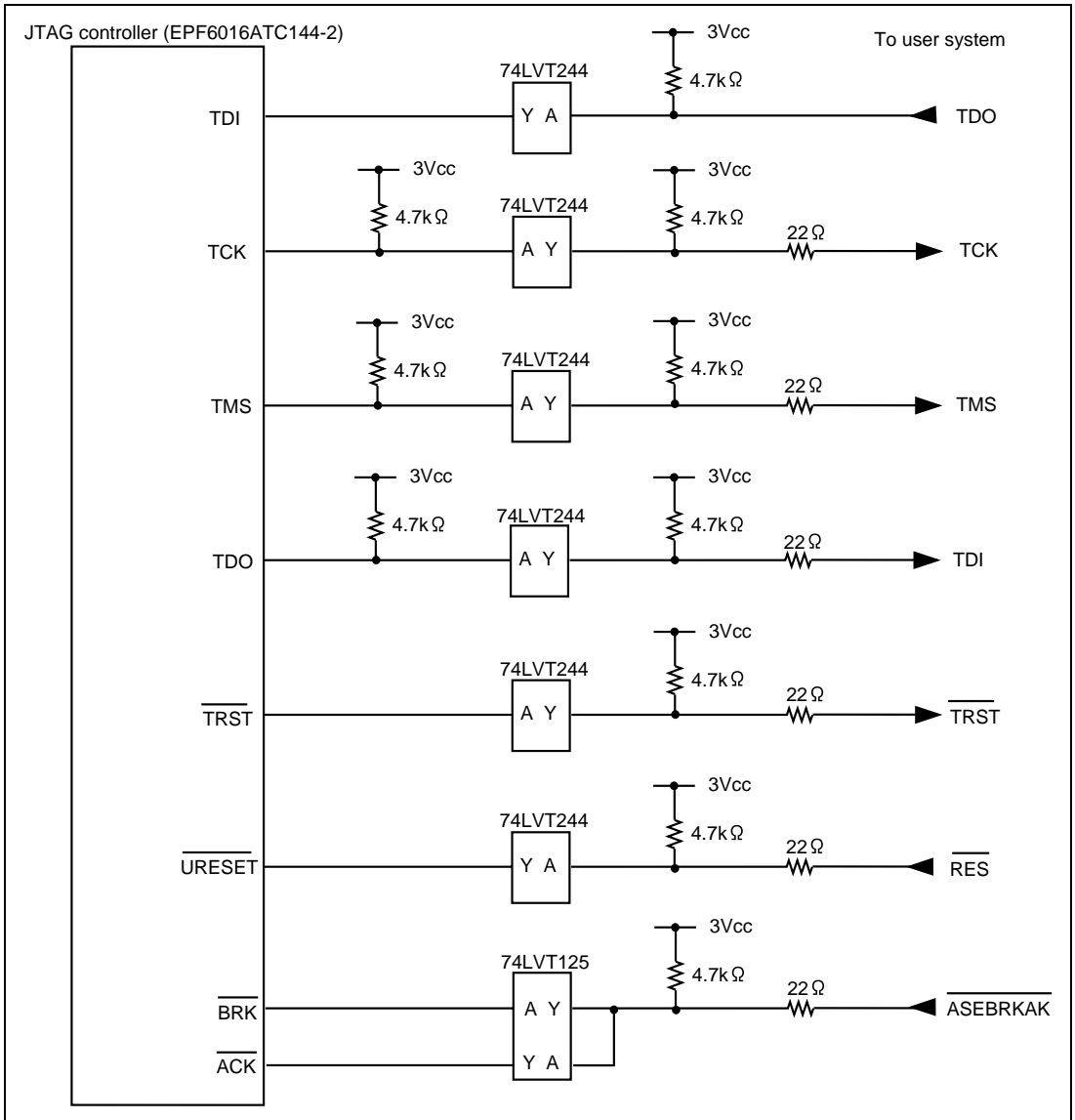


Figure 6.4 User System Interface Circuit (HS7047KCM01H)

The user system interface circuits of the emulator (HS7047KCM02H) are shown. Figures 6.5 and 6.6 show the circuits of the Hitachi-UDI pin and AUD pin, respectively.

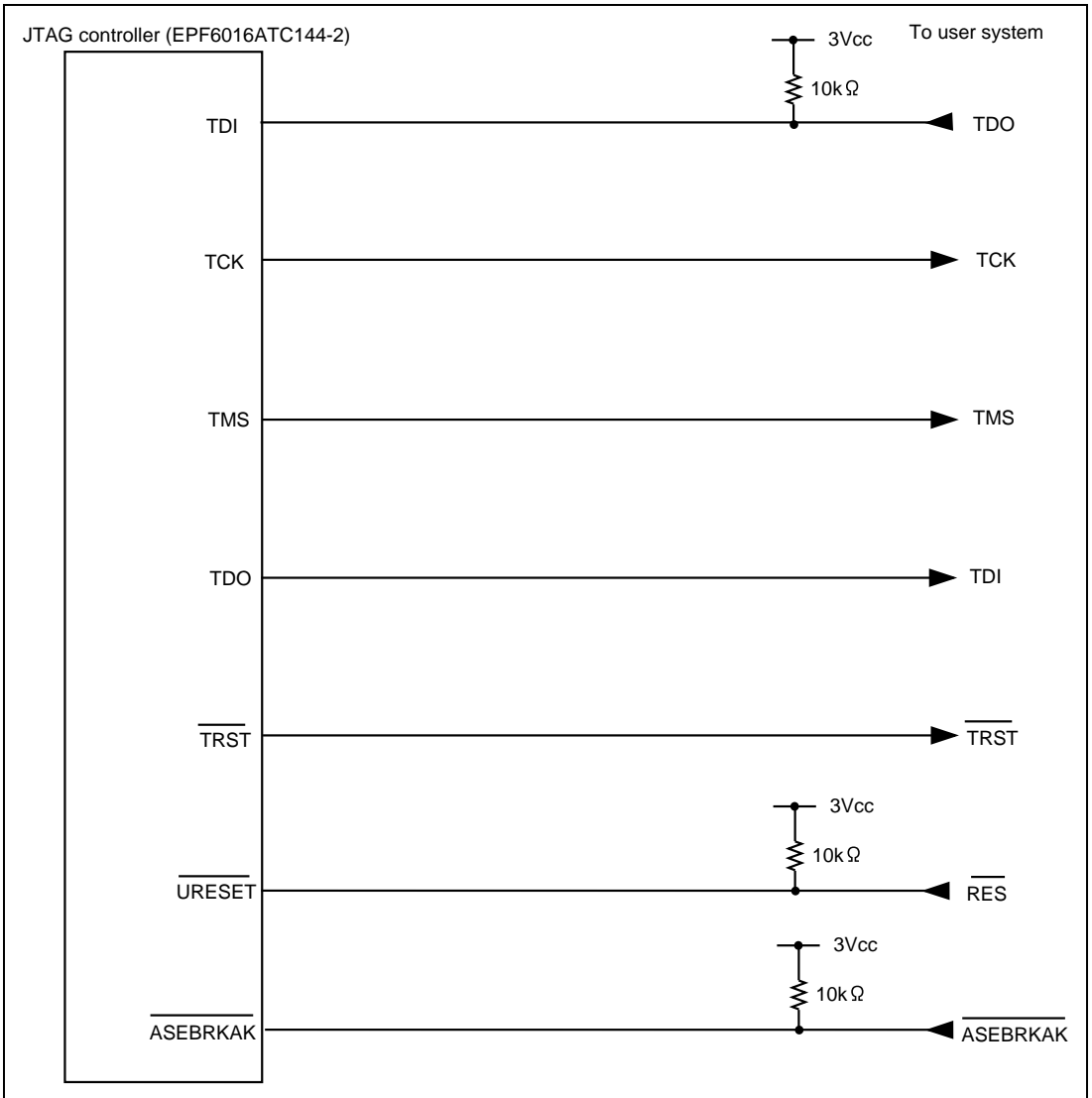


Figure 6.5 User System Interface Circuit of the Hitachi-UDI Pin (HS7047KCM02H)

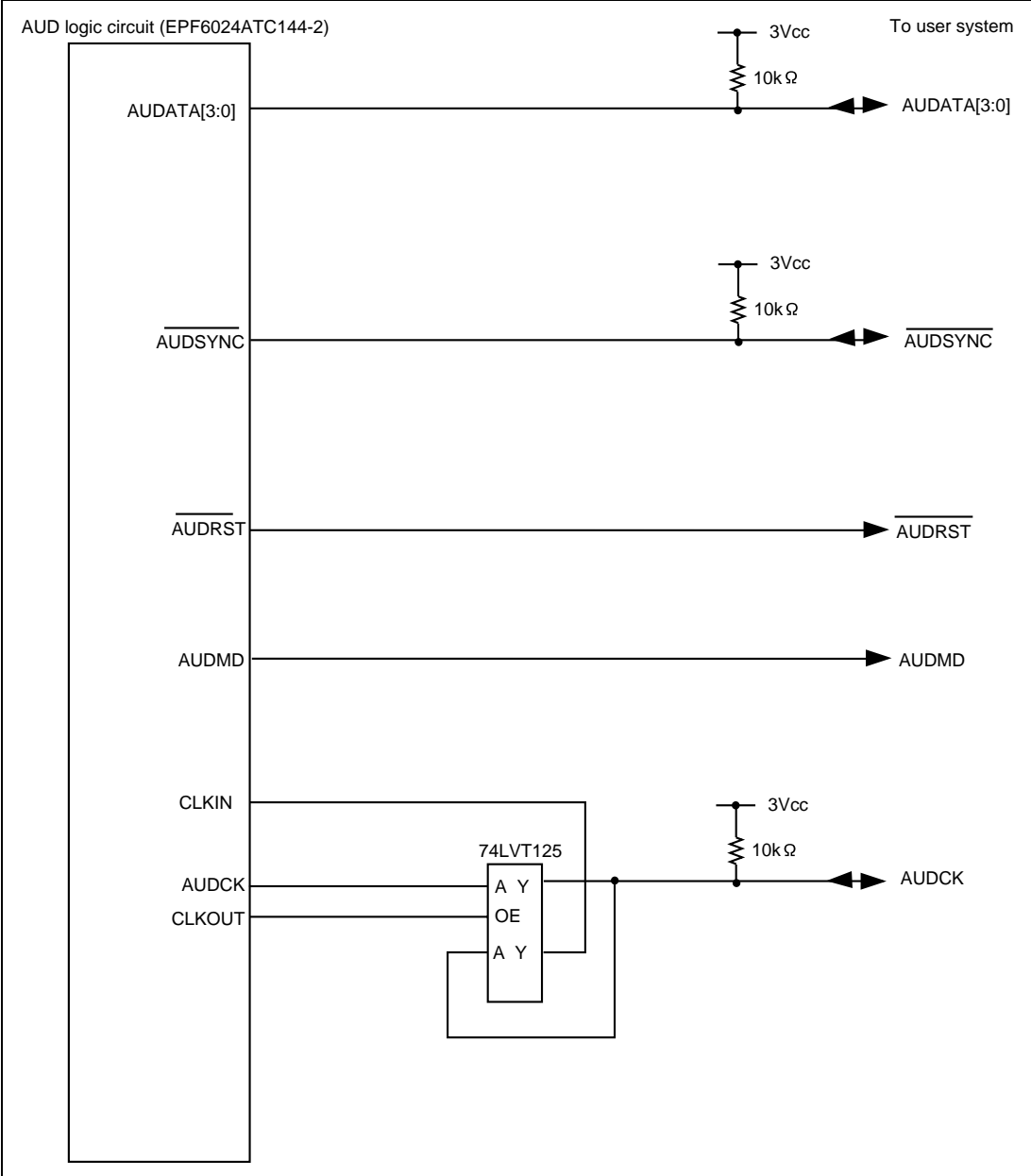


Figure 6.6 User System Interface Circuit of the AUD Pin (HS7047KCM02H)

Figure 6.7 shows the user system interface circuit of the emulator (HS7047KCI01H).

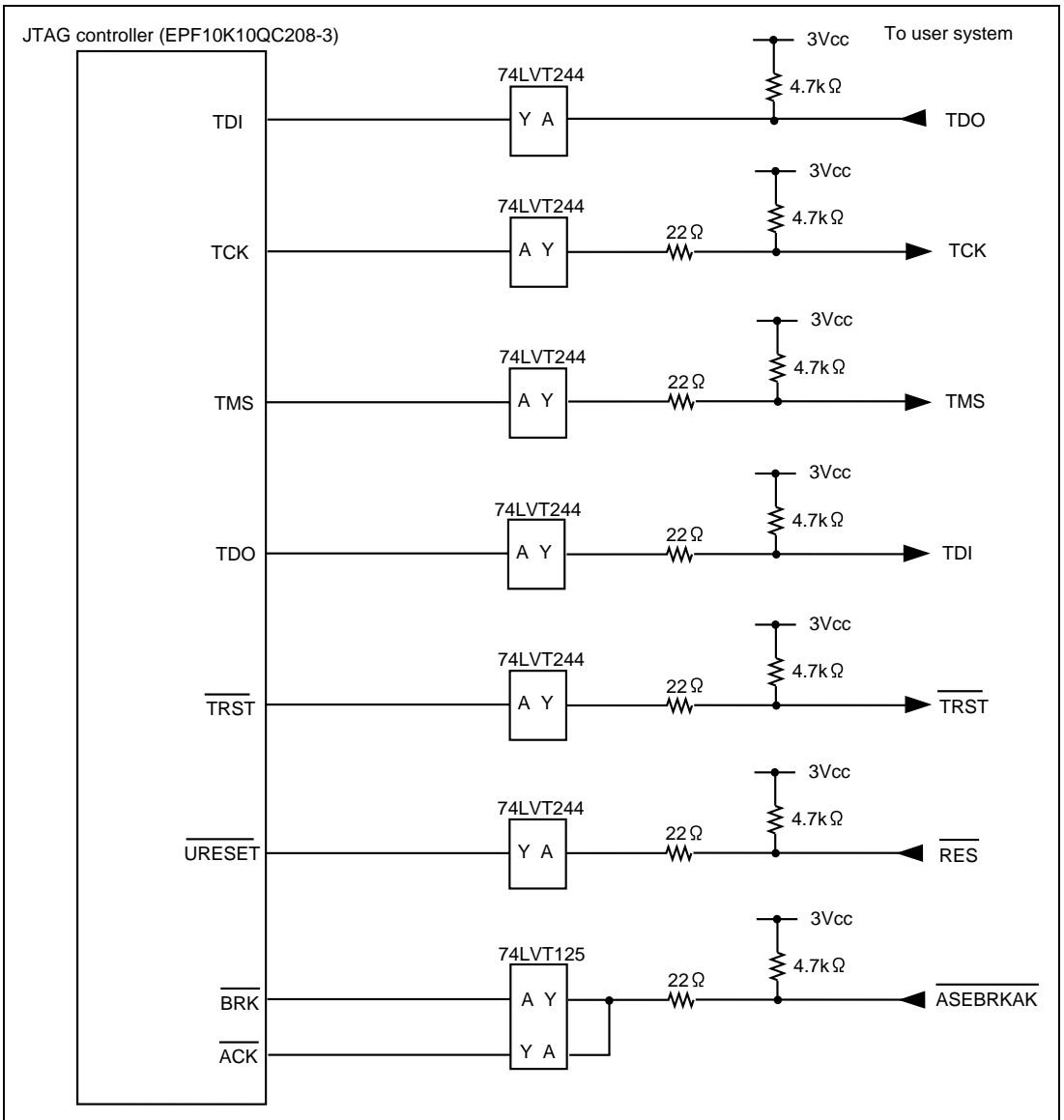


Figure 6.7 User System Interface Circuit (HS7047KCI01H)

The user system interface circuits of the emulator (HS7047KCI02H) are shown. Figures 6.8 and 6.9 show the circuits of the Hitachi-UDI pin and AUD pin, respectively.

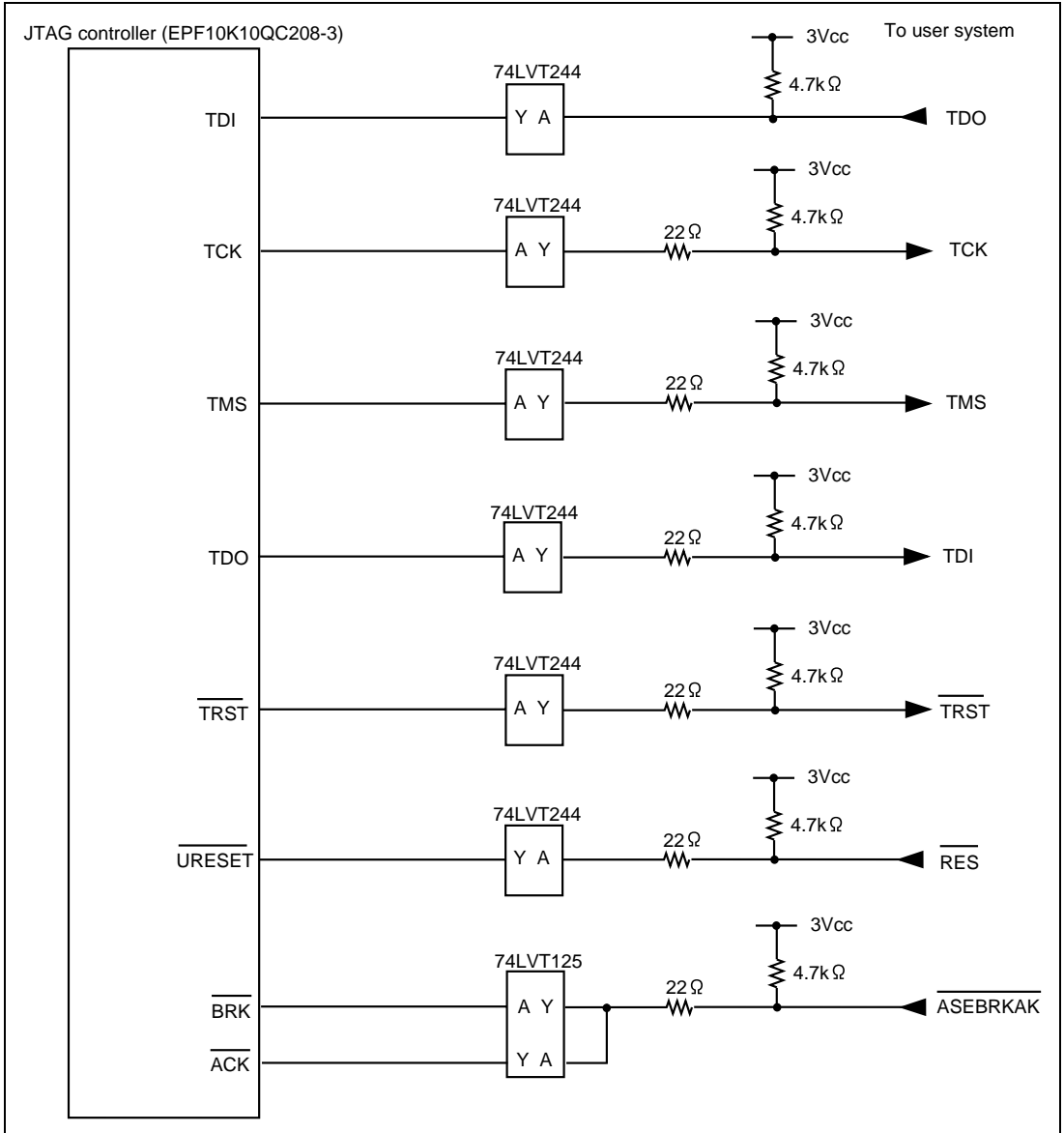


Figure 6.8 User System Interface Circuit of the Hitachi-UDI Pin (HS7047KCI02H)

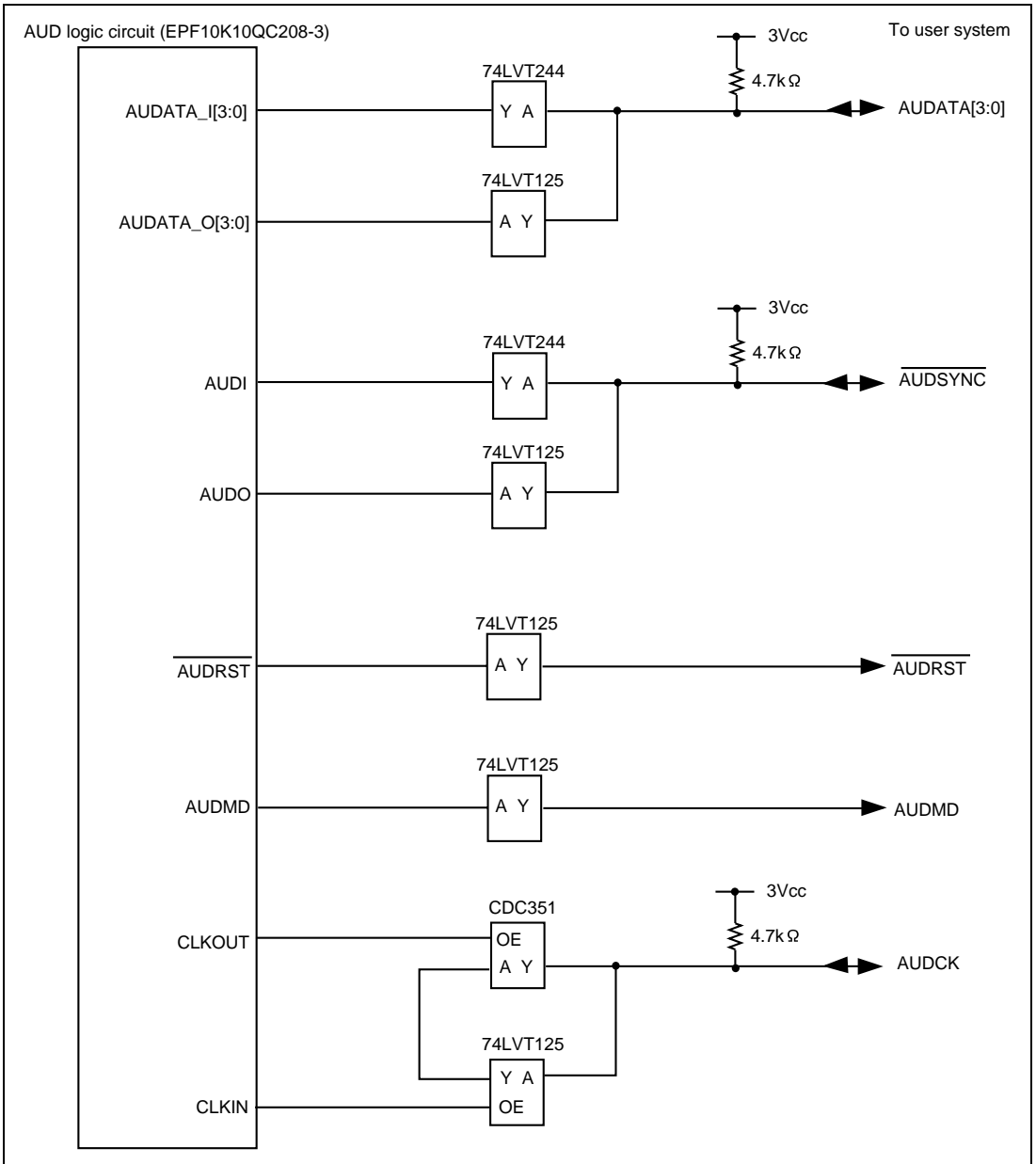


Figure 6.9 User System Interface Circuit of the AUD Pin (HS7047KCI02H)

6.4 Differences between the SH7047F and the Emulator

- When the emulator system is initiated, it initializes the general registers and part of the control registers as shown in table 6.2. When the emulator is not used, the initial values of the actual SH7047F general register is undefined.

Table 6.2 Register Initial Values at Emulator Power-On

Register	Initial value when the emulator is used
R0 to R14	H'00000000
R15 (SP)	Value of the SP in the vector address table
PC	Value of the PC in the vector address table
SR	H'000000F0
GBR	H'00000000
VBR	H'00000000
MACH	H'00000000
MACL	H'00000000
PR	H'00000000

- The emulator uses the Hitachi-UDI; do not access the Hitachi-UDI.
- Low-Power Modes (Sleep, Standby, and Module Standby)

For low-power consumption, the SH7047F has sleep, standby, and module standby modes.

When the emulator is used, the sleep mode can be cleared with either the normal clearing function or with the forced break. Note that, however, if a command has been entered in standby mode or module standby mode, no commands can be used from the emulator since the mode is cleared only with the normal clearing function.

Notes: The memory must not be accessed or modified in software standby mode.

1. **Do not set the MSTP2 or MSTP27 bit to 1 in the MSTCR2 register (address H'FFFF861E) nor the RAME bit to 0 in the SYSCR register (address H'FFFF8618), since doing so may prevent the emulator from operating correctly.**
2. **When the HS7047KCM02H or HS7047KCI02H is used, do not set the following bits to 0: the MSTP3 bit in the MSTCR2 register (address H'FFFF861E) and the AUDSRST bit in the SYSCR register (address H'FFFF8618). The AUD function cannot be used.**

- RESET Signal ($\overline{\text{RES}}$)

The RESET signal can be accepted during user program break, when the peripheral modules are reset. Since the PC, SR, and SP registers are not initialized, reset them and execute the GO command when the user program is executed from the reset vector.

Note: Do not break the user program when the RES, BREQ, and WAIT pins are being low. A TIMEOUT error will occur. The TIMEOUT error will also occur at memory access when the BREQ and WAIT pins are fixed low during break.

- Data Transfer Controller (DTC)

The DTC operates even in the command wait state. When a data transfer request is generated, the DTC executes a DMA transfer.

- Interrupt

All interrupts except for NMI are masked during user program break.

6.5 The SH7047F E10A Emulator Functions

In the SH7047F E10A emulator, the realtime memory access function by using the RAM monitor function has been added.

The following functions are not supported:

- Performance analysis function
- Profile function

6.5.1 Emulator Driver Selection

Table 6.3 shows drivers which can be selected in the [E10A Driver Details] dialog box.

Table 6.3 Type Name and Driver

Type Name	Driver
HS7047KCM01H	E10A PC Card Driver 3
HS7047KCM02H	E10A PC Card Driver 4
HS7047KCI01H	E10A PCI Card Driver 3
HS7047KCI02H	E10A PCI Card Driver 4

6.5.2 Hardware Break Functions

The emulator can set hardware break conditions. Table 6.4 lists these hardware break conditions.

Table 6.4 Hardware Break Condition Specification Items

Items	Description
Address bus condition (Address)	Breaks when the MCU address bus value or program counter value matches the specified value.
Data size condition (Size)	Breaks when the data size that has been accessed matches the specified value. Byte, word, or longword can be specified as the access data size.
Read or write condition (Read or Write)	Breaks in the read or write cycle.
Access type	Breaks when the bus cycle is the specified cycle.

Table 6.5 lists the combinations of conditions that can be set in the [Break Condition 1] dialog box.

Table 6.5 Conditions Set in the [Break Condition] Dialog Box

Dialog Box	Type	
	Address Bus Condition ([Address] page)	Access Type Condition, Read or Write Condition, Data Size Condition ([Bus state] page)
[Break Condition 1] dialog box	<input type="radio"/>	<input type="radio"/>
[Break Condition 2] dialog box	<input type="radio"/>	<input type="radio"/>
[Break Condition 3] dialog box	<input type="radio"/>	<input type="radio"/>
[Break Condition 4] dialog box	<input type="radio"/>	<input type="radio"/>
[Break Condition R] dialog box	<input type="radio"/>	-

Note: : Can be set by clicking the radio button in the dialog box.

Table 6.6 lists the combinations of conditions that can be set with the BREAKCONDITION_SET command.

Table 6.6 Conditions Set with the BREAKCONDITION_SET Command

Channel	Type	
	Address Bus Condition (<addropt> option)	Access Type Condition (<accessopt> option), Read or Write Condition (<r/wopt> option), Data Size Condition (<sizeopt> option)
Break Condition 1	<input type="radio"/>	<input type="radio"/>
Break Condition 2	<input type="radio"/>	<input type="radio"/>
Break Condition 3	<input type="radio"/>	<input type="radio"/>
Break Condition 4	<input type="radio"/>	<input type="radio"/>
Break Condition R	<input type="radio"/>	-

Note: : Can be set by the BREAKCONDITION_SET command.

Notes on Setting the [Break Condition] Dialog Box and BREAKCONDITION_SET Command:

1. When [Go to cursor], [Step In], [Step Over], or [Step Out] is selected, the settings of Break Condition 3 are disabled.
2. Break Condition 3 is disabled when an instruction to which a BREAKPOINT has been set is executed. Accordingly, do not set a BREAKPOINT to an instruction which satisfies Break Condition 3.
3. When a Break Condition is satisfied, emulation may stop after two or more instructions have been executed.
4. If a PC break before execution is set to the slot instruction after a delayed branch instruction, user program execution cannot be terminated before the slot instruction execution; execution stops before the branch destination instruction.

6.5.3 Notes on Setting the [Breakpoint] Dialog Box

1. When an odd address is set, the address is rounded down to an even address.
2. A software break is accomplished by replacing instructions. Accordingly, it can be set only to the RAM area. However, a software break cannot be set to the following addresses:
 - An area other than the CS0 areas and the internal RAM area
 - An instruction in which Break Condition 3 is satisfied
 - A slot instruction of a delayed branch instruction
3. During step execution, software breakpoints are disabled.
4. A condition set at Break Condition 3 is disabled when an instruction at a software breakpoint is executed. Do not set a software breakpoint to an instruction in which Break Condition 3 is satisfied.
5. When execution resumes from the breakpoint address after the program execution stops at a software breakpoint, single-step execution is performed at the address before execution resumes. Therefore, realtime operation cannot be performed.
6. When a software breakpoint is set to the slot instruction of a delayed branch instruction, the PC value becomes an illegal value. Accordingly, do not set a software breakpoint to the slot instruction of a delayed branch instruction.

6.5.4 Notes on Using the JTAG Clock (TCK)

When the JTAG clock (TCK) is used, set the frequency to lower than that of the system clock.

6.5.5 AUD Functions

In the SH7047F E10A emulator, the following functions using the AUD function can be used. These functions are operational when the AUD pin is connected to the emulator

Table 6.7 AUD Functions

Function	Description
Branch trace function	Displays the addresses and instruction words at the branch destination.
RAM monitor function	Enables realtime memory reading or writing during user program execution.

- Notes:
1. When HS7047KCM01H and HS7047KCI01H are used, the AUD function cannot be used.
 2. When the MCU operating mode is MCU expansion mode 2, the AUD function cannot be used.
 3. Set the following before using the AUD function in the user program. The AUD function does not operate correctly until the following is set after user program execution is started:
 - Set H'00FF to the PDCRL1 and PDCRL2 registers to select the following pins: AUDSYNC input/output, AUDCK input/output, AUDMD input, AUDRST input, AUDATA3 input/output, AUDATA2 input/output, AUDATA1 input/output, and AUDATA0 input/output.
 - Set 1 to the PB5MD2 bit in the PBCRL1 register, 0 to the PB5MD1 bit in the PBCRL2 register, and 1 to the PBMD0 bit in the PBCRL2 register to select the CK output.
 - Set 1 to the AUDSRST bit in the SYSCR register and cancel the AUD reset.

(1) Branch Trace Function

When a branch occurs during user program execution, the branch destination address is acquired. When the next branch occurs while the trace information is being output, the information is stopped and the next trace information is output. The user program can be executed in realtime, but some trace information will not be output.

Note: After [Trace stop] is selected in [AUD mode] of the [Trace mode] page in the [Trace Acquisition] dialog box and the trace buffer of the emulator becomes full, the trace information is not acquired. The user program is continuously executed.

(2) Realtime Memory Access Function

Realtime memory reading or writing is enabled during user program execution. The specified memory address contents (maximum three) can be displayed on the status bar. The memory contents can be changed by the command line, and can be referenced in the [Memory] window.

Memory can be read or written as follows:

1. When the [Memory] window is used:

Memory can be read or written during user program execution. Open the address to be referenced in the [Memory] window. At reference, select Refresh from the [Memory] menu or issue the Refresh command in the command-line window.

2. When the command line is used:

MEMORY_EDIT command: Memory can be read or written during user program execution.

RAM_R command: Address and size displayed on the status bar are specified during user program execution.

RAM_W command: Memory can be changed during user program execution. One command can change a maximum of three addresses.

Note: Note that the flash memory area cannot be written by the RAM_W command.

(3) Products Using the AUD Function and Note

Table 6.8 Type Name and AUD Function

Type Name	AUD Function
HS7047KCM01H	Not available
HS7047KCM02H	Available
HS7047KCI01H	Not available
HS7047KCI02H	Available

Notes: 1. To use the AUD function, set the system clock to 40 MHz or lower.

2. Trace cannot be acquired while memory read/write is being performed by using the RAM monitor function during user program execution.

6.5.6 Notes on Displaying the [Trace] Window

1. The AUD trace outputs the differences between newly output branch destination addresses and the previously output branch destination addresses. If the previous branch destination address is the same as the upper 16 bits, the lower 16 bits are output. If it matches the upper 24 bits, the lower 8 bits are output. If it matches the upper 28 bits, the lower 4 bits are output. The emulator regenerates the 32-bit address from these differences and usually displays it in the [Trace] window. If the emulator cannot display the 32-bit address, it displays the difference from the previously displayed 32-bit address.
2. When a completion-type exception occurs during exception branch acquisition, the next address is the address in which an exception occurs is acquired.
3. When the [Halt] option is used from the popup menu in the [Trace] window, realtime emulation is retained.
4. In the SH7047F E10A emulator, the maximum number of trace display pointers is as follows:
When HS7047KCM02H is used: D'16383 to 0
When HS7047KCI02H is used: D'65535 to 0
However, the maximum number of trace display pointers differs according to the AUD trace information to be output. Therefore, the above pointers cannot be always acquired.
5. When only one line is acquired by trace, the display data is not updated. In this case, open the [Trace] window again.

6.5.7 Notes on HDI

1. Note on Moving Source File Position after Creating Load Module
When the source file is moved after creating the load module, the [Open] dialog box may be displayed to specify the source file during the debugging of the created load module. Select the corresponding source file and click the [Open] button.

2. Source-level Execution

— Source file

Do not display source files that do not correspond to the load module in the program window. For a file having the same name as the source file that corresponds to the load module, only its addresses are displayed in the program window. The file cannot be operated in the program window.

— Step

Even standard C libraries are executed. To return to a higher-level function, enter Step Out. In a for statement or a while statement, executing a single step does not move execution to the next line. To move to the next line, execute two steps.

3. Operation During Accessing Files

Do not perform other operations during saving in the [Load Program], [Verify Memory], [Save Memory], or [Trace] window because this will not allow correct saving to be performed.

4. Source Window at Program Change

When a program being displayed in the source window is changed and the source file and load module are reloaded, close and reopen the source window once. If the window is not closed and reopened, the display will be illegal.

5. Watch

— Local variables at optimization

Depending on the generated object code, local variables in a C source file that is compiled with the optimization option enabled will not be displayed correctly. Check the generated object code by displaying the [Disassembly] window.

If the allocation area of the specified local variable does not exist, displays as follows.

Example: The variable name is asc.

 asc = ? - target error 2010 (xxxx)

— Variable name specification

When a name other than a variable name, such as a symbol name or function name, is specified, no data is displayed.

Example: The function name is main.

 main =

— Array display

When array elements exceed 1000, elements from after 1000 will not be displayed.

6. Memory Load Function

When [Load...] is selected from the [Memory] menu, the Memory Load function can be used although it takes time to download. It is recommended that the File Load function ([Load Program...] selected from the [File] menu) is used to load the S-type file.

Note: The File Load function deletes the debugging information of the previously loaded program. When other load modules are loaded after the program to be debugged has been loaded, use the following sequence: Link the program to be debugged with the Sdebug option and save the debugging information in another file. Load the debugging information file after all the load modules have been loaded.

7. Line Assembly

— Input radix

Regardless of the Radix setting, the default for line assembly input is decimal. Specify H' or 0x as the radix for a hexadecimal input.

8. Command Line Interface

— Batch file

To display the message “Not currently available” while executing a batch file, enter the sleep command. Adjust the sleep time length which differs according to the operating environment.

Example: To display “Not currently available” during memory_fill execution:

```
sleep d'3000
```

```
memory_fill 0 ffff 0
```

— Overwrite file

In Command Line Interface, a file having the same name as the output file is overwritten without asking the user.

— File specification by commands

The current directory may be altered by file specifications in commands. Absolute paths should be used to specify the files in a command file so that the current directory alteration is not affected

Example: FILE_LOAD
C:\\HEW\\HDI5\\E10A\\Tutorial\\Debug\\TUTORIAL.ABS

9. About Hitachi Debugging Interface User's Manual

This version of HDI does not support section 10, Selecting Functions, written in Hitachi Debugging Interface User's Manual.

10. Note on Initiating HDI

When the emulator is initiated by using another card emulator after it has been initiated by using the PCI card emulator, delete the [TARGET] line from the C:\windows\HDI.INI file.

11. Usage with Another HDI

— Automatic load of session files

Since the emulator cannot use another HDI, re-install this HDI whenever another HDI has been previously installed is used.

If another HDI has been used, initiate this HDI with “Run” as follows without using the session files.

```
<Directory path name in which HDI is installed>\hdi /n (RET)
```

/n initiates the HDI without loading the recently used session files.

If there is another session file in the different debug platform, the following error message is displayed:

```
invalid target system: <recently used debug platform name>
```

— Uninstallation of Another HDI

If another HDI is uninstalled after this HDI has been installed, the stack trace function cannot be used. In this case, re-install this HDI.

12. [Select Function] Dialog Box

This HDI does not support software breakpoint setting in the [Select Function] dialog box (described in section 10, Selecting Functions, in the Hitachi Debugging Interface User's Manual).

13. Memory Save During User Program Execution

Do not execute memory save verifying during user program execution.

14. [Performance Analysis] Window

This HDI does not support the [Performance Analysis] window (described in section 13.7, Performance Analysis, in the Hitachi Debugging Interface User's Manual).

15. Load of Motorola S-type Files

This HDI does not support Motorola S-type files with only the CR code (H'0D) at the end of each record. Load Motorola S-type files with the CR and LF codes (H'0D0A) at the end of each record.

16. [Memory] Window

If the following memory contents are displayed, they will be incorrect.

Word access from address $2n + 1$

Longword access from address $4n + 1$, $4n + 2$, or $4n + 3$

17. Scrolling Window During User Program Execution

Do not scroll the [Memory] and [Disassembly] windows by dragging the scroll box during user program execution. This generates many memory reads causing the user program to stop execution until the memory reads have been completed.

18. [I/O Registers] window

— Display and modification

- For each Watchdog Timer register, there are two registers to be separately used for write and read.

Table 6.9 Watchdog Timer Registers

Register Name	Usage	Register
TCSR (R)	Read	Watchdog timer control/status register
TCNT (R)	Read	Watchdog timer counter
RSTCSR (R)	Read	Reset control/status register
TCSR (W)	Write	Watchdog timer control/status register
TCNT (W)	Write	Watchdog timer counter
RSTCSR (W)	Write	Reset control/status register

- Note that the E10A emulator does not support the bit-field function described in the Hitachi Debugging Interface User's Manual.
- Verify
In the [I/O Registers] window, the verify function of the input value is invalid.

19. Note on [Registers] Window Operation During Program Execution

Although a dialog box is displayed in which the register contents can be changed by double-clicking the [Registers] window, do not change the register contents during program execution.

20. Note on Radix in the [Register] Dialog Box

The default input radix in the [Register] dialog box is hexadecimal irrespective of the Radix display. When a radix other than a hexadecimal is input, specify the prefix code such as B'.

21. Software Break

— Session file

When the software breakpoint address set in the session file is H'0, the breakpoint will not be set. If the address set as the breakpoint is wrong, the error message is not output. The breakpoint is registered as DISABLE in the [Breakpoints] window.

— Breakpoint cancellation

When the contents of the software breakpoint address is modified during user program execution, the following message is displayed when the user program stops.

```
BREAKPOINT IS DELETED A=xxxxxxx
```

If the above message is displayed, cancel all software breakpoint settings with the [Delete All] or [Disable] button in the [Breakpoints] window.

— [Run program] dialog box

If an invalid software breakpoint address is specified as a stop address in the [Run Program] dialog box, the invalid software breakpoint will become valid after the user program has stopped.

— [Breakpoints] window

During user program execution, it is impossible to jump from the breakpoint to the source or address line on the [Source] or [Disassembly] window by using Go to Source in the popup menu displayed on the [Breakpoints] window.

22. Number of Software Breakpoint and [Stop At] Settings in the [Run...] Menu

The maximum number of software breakpoints and [Stop At] settings allowed in the [Run...] menu is 255. Therefore, when 255 software breakpoints are set, specification by [Stop At] in the [Run...] menu becomes invalid. Use the software breakpoints and [Stop At] in the [Run...] menu with 255 or less total settings.

23. Note on RUN-TIME Display

The execution time of the user program displayed in the [Status] window is not a correct value since the timer in the host computer has been used.

24. Note on Displaying COMMUNICATION TIMEOUT ERROR

If COMMUNICATION TIMEOUT ERROR is displayed, the emulator cannot communicate with the MCU. Select [Initialize] from the [File] menu to initialize the emulator. Before the emulator is restarted, turn off the user system and disconnect the interface cable that has been connected with the emulator.

25. Note on Downloading Program

In the [Load Program] dialog box, which is opened when [Load Program...] is selected, the verify function is invalid. After downloading the program, perform verify in the [Verify S-Record File with Memory] dialog box, which is opened when [Verify] is selected from the [Memory] menu.

26. Support of Double Float Format

In the following memory operations, the double float format is not supported:

- [Fill Memory] dialog box
- [Search Memory] dialog box
- MEMORY_FILL command

The [Format] specification in the [Copy Memory] dialog box is ignored. Memory is copied in a byte unit.

27. Note on Continuous Step Execution

When the step is continuously executed by selecting [Step...] from the [Run] menu, do not use the BREAKPOINT because this will cause the HDI to abnormally operate.

28. Note on Using the [Run program] Dialog Box

When [Run...] is selected from the [Run] menu to specify the stop address, there is the following note:

- When the breakpoint that has been set as Disable is specified as the stop address, note that the breakpoint becomes Enable when the user program stops.

29. When the user program is loaded in the CS0 area by loading a session, the program will not be correctly loaded if the bus size for CS0 is illegal. Set the bus size by using the BCR1 register, then load the program again.

30. Processing Time for Updating the Flash Memory Contents

When the contents of the flash memory area is modified by the program loading, memory window, or memory command, or when a software break is set, a waiting time will be generated to write or read the flash memory before executing the user program.

The processing time for updating the flash memory contents will be about a maximum of 60 seconds under the following environments (reference values):

- Host computer: 500 MHz (Pentium® III)
- SH7047F: 40 MHz (system clock frequency)