RENESAS

H8S, H8SX Family E10A-USB Emulator

User's Manual

(HS0005KCU01H, HS0005KCU02H)

H8S, H8SX Family E10A-USB

HS0005KCU01HE-H8S

Renesas Electronics
www.renesas.com

Rev. 14.01   Mar. 2016

Regulatory Compliance Notices

● European Union regulatory notices

This product complies with the following EU Directives. (These directives are only valid in the European Union.)

CE Certifications:

• Electromagnetic Compatibility (EMC) Directive 2014/30/EU

　　　　EN 55022:2010 Class A

WARNING:　This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

　　　　EN 55024:2010

• Information for traceability
  • Authorised representative
    Name:　　　Renesas Electronics Corporation
    Address:　　Toyosu Foresia, 3-2-24, Toyosu, Koto-ku, Tokyo, 135-0061, Japan
  • Manufacturer
    Name:　　　Renesas System Design Co.,Ltd.
    Address:　　5-20-1, Josuihon-cho, Kodaira-shi, Tokyo 187-8588, Japan
  • Person responsible for placing on the market
    Name:　　　Renesas Electronics Europe GmbH
    Address:　　Arcadiastrasse 10, 40472 Dusseldorf, Germany
  • Trademark and Type name
    Trademark:　Renesas
    Product name:　E10A-USB Emulator
    Type name:　HS0005KCU01H / HS0005KCU02H

Environmental Compliance and Certifications:

• Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment (RoHS)

　Directive 2002/95/EC

• Waste Electrical and Electronic Equipment (WEEE) Directive 2002/96/EC

● United States Regulatory notices

This product complies with the following EMC regulation. (This is only valid in the United States.)

FCC Certifications:

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:
(1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

CAUTION: Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

IMPORTANT INFORMATION

READ FIRST

• READ this user's manual before using this emulator product.

• <u>KEEP the user's manual handy for future reference.</u>

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 Renesas Electronics Corp. 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 Renesas 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 Renesas sales company. Such use includes, but is not limited to, use in life support systems. Buyers of this emulator product must notify the relevant Renesas sales offices before planning to use the product in such applications.

Improvement Policy:

Renesas Electronics Corp. (including its subsidiaries, hereafter collectively referred to as Renesas) pursues a policy of continuing improvement in design, performance, and safety of the emulator product. Renesas 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

Renesas warrants its emulator products to be manufactured in accordance with published specifications and free from defects in material and/or workmanship. Renesas, at its option, will replace any emulator products returned intact to the factory, transportation charges prepaid, which Renesas, upon inspection, shall determine to be defective in material and/or workmanship. The foregoing shall constitute the sole remedy for any breach of Renesas' warranty. See the Renesas 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. Renesas is not liable for any claim made by a third party or made by you for a third party.

DISCLAIMER

RENESAS 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 RENESAS 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:

Renesas 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 Renesas' 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 Renesas. 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 Renesas' 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 Renesas' semiconductor products. Renesas 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 Renesas.

Figures:

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

Device names:

Sections 1 to 6 of the Debugger Part in this user's manual use H8S/xxxx as an example of the device names.

Limited Anticipation of Danger:

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

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

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

CAUTION   **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.

RENESAS

⚠ 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.**

CAUTION

Caution to Be Taken for Disposal:

Penalties may be applicable for incorrect disposal of this waste, in accordance with your national legislation.

European Union regulatory notices:

The WEEE (Waste Electrical and Electronic Equipment) regulations put responsibilities on producers for the collection and recycling or disposal of electrical and electronic waste. Return of WEEE under these regulations is applicable in the European Union only. This equipment (including all accessories) is not intended for household use. After use the equipment cannot be disposed of as household waste, and the WEEE must be treated, recycled and disposed of in an environmentally sound manner.

Renesas Electronics Europe GmbH can take back end of life equipment, register for this service at " http://www.renesas.eu/weee ".

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.**

Introduction

The High-performance Embedded Workshop is a powerful development environment for embedded applications targeted at Renesas microcontrollers. The main features are:

- A configurable build engine that allows you to set-up compiler, assembler and linker options via an easy to use interface.
- An integrated text editor with user customizable syntax coloring to improve code readability.
- A configurable environment to run your own tools.
- An integrated debugger which allows you to build and debug in the same application.
- Version control support.

The High-performance Embedded Workshop has been designed with two key aims; firstly to provide you, the user, with a set of powerful development tools and, secondly, to unify and present them in a way that is easy to use.

About This Manual

This manual describes preparation before using the emulator, emulator functions, debugging functions specific to the emulator, tutorial, and emulator's hardware and software specifications.

Refer to the High-performance Embedded Workshop User's Manual for details on the information on the basic usage of the High-performance Embedded Workshop, customization of the environment, build functions, and debugging functions common to each High-performance Embedded Workshop product.

This manual does not intend to explain how to write C/C++ or assembly language programs, how to use any particular operating system or how best to tailor code for the individual devices. These issues are left to the respective manuals.

This manual applies to v.3.00 Release 00 and later versions of the E10A-USB emulator software. The devices groups shown below are not supported by v.3.00 Release 00 and later versions of the E10A-USB emulator software.

Table 1   E10A-USB Devices Groups which are No Longer Supported

H8S/2437 device group
H8S/2114 device group
H8S/2172 device group
H8S/2158 device group
H8S/2189R device group
H8S/2116 device group
H8S/2125 device group

Microsoft® and Windows® are registered trademarks of Microsoft Corporation.

Visual SourceSafe is a trademark of Microsoft Corporation.

IBM is a registered trademark of International Business Machines Corporation.

All brand or product names used in this manual are trademarks or registered trademarks of their respective companies or organizations.

Document Conventions

This manual uses the following typographic conventions:

Table 2   Typographic Conventions

Convention	Meaning
[Menu->Menu Option]	Bold text with '->' is used to indicate menu options (for example, **[File->Save As...]**).
FILENAME.C	Uppercase names are used to indicate filenames.
"<u>enter this string</u>"	Used to indicate text that must be entered (excluding the "" quotes).
Key + Key	Used to indicate required key presses. For example, **CTRL+N** means press the **CTRL** key and then, whilst holding the **CTRL** key down, press the **N** key.
➲ (The "how to" symbol)	When this symbol is used, it is always located in the left hand margin. It indicates that the text to its immediate right is describing "how to" do something.

User Registration

When you install debugger software, a text file for user registration is created on your PC. Fill it in and email it to your local distributor. If you have replaced an emulator main unit or emulation probe, rewrite an emulator name and serial number in the text file you filled in earlier to register your new hardware products.

Your registered information is used for only after-sale services, and not for any other purposes. Without user registration, you will not be able to receive maintenance services such as a notification of field changes or trouble information. So be sure to carry out the user registration.

For more information about user registration, please contact your local distributor.

Table of Contents

Section 1   Overview

The E10A-USB emulator (hereafter referred to as the emulator) is a support tool for developing application systems to run on Renesas original microcomputers.

The main unit of the emulator is connected, through the dedicated debugging interface, 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 USB 1.1/2.0 (Full-Speed).

Figure 1.1 shows the system configuration using the emulator.
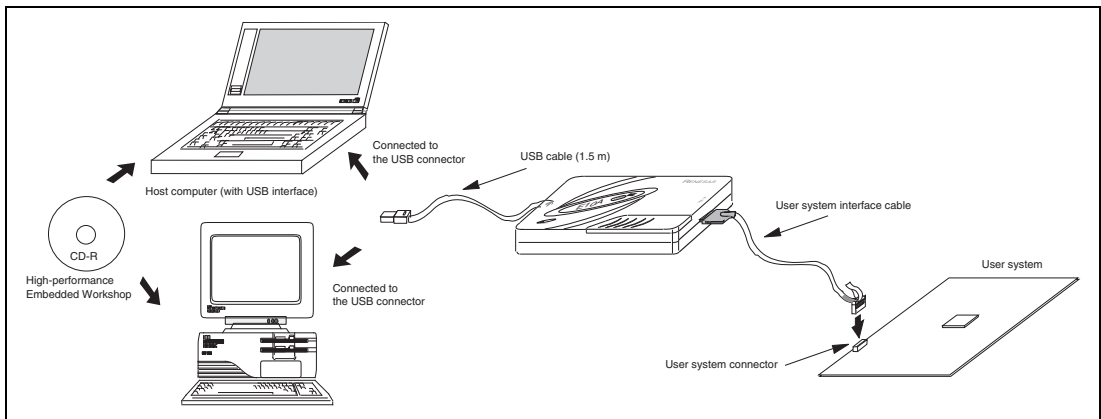
Figure 1.1   System Configuration with the Emulator

The emulator provides the following features:

- Excellent cost-performance emulator

  Compactness and connection to the USB are obtained.

- Realtime emulation

  Realtime emulation of the user system is enabled at the maximum operating frequency of the CPU.

- Excellent operability

  Using the High-performance Embedded Workshop enables user program debugging using a pointing device such as a mouse. The High-performance Embedded Workshop 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.

- 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 emulator is used, a laptop computer can be used as a host computer, creating a debugging environment in any place.

1.1   Warnings

<div style="border: 2px solid black; padding: 20px;">

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.

</div>

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. When moving the host computer or user system, take care not to vibrate or damage it.
5. After connecting the cable, check that it is connected correctly. For details, refer to section 3, Preparation before Use.
6. 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

<div style="border:1px solid black; padding:1em;">

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.

</div>

Table 1.1   Environmental Conditions

Item	Specifications	
Temperature	Operating: Storage:	+10°C to +35°C −10°C to +50°C
Humidity	Operating: Storage:	35% RH to 80% RH, no condensation 35% RH to 80% RH, no condensation
Vibration	Operating: Storage: Transportation:	2.45 m/s^2 max. 4.9 m/s^2 max. 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	32-Bit Editions of Windows® XP	32-Bit Editions of Windows Vista® or 32-Bit or 64-Bit Editions of Windows® 7
Host computer	Built-in Pentium® III or higher-performance CPU (1 GHz or higher recommended); IBM PC or compatible machine with USB 1.1/2.0 (Full-Speed).	
CPU	Pentium® III (1 GHz) or higher recommended	Pentium® 4 (3 GHz), Core™ 2 Duo (1 GHz), or higher recommended
Minimum memory capacity	1 Gbyte or more recommended	1.5 Gbyte or more recommended (at least 10 times the size of the load module file)
Hard-disk capacity	Installation disk capacity: 600 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® XP, Windows Vista®, or Windows® 7	
Display	Monitor resolution: 1024 x 768 or higher	
Power voltage	5.0 ± 0.25 V (USB-bus power type)	
Current consumption	HS0005KCU01H: 260 mA (max.) HS0005KCU02H: 420 mA (max.)	
CD-ROM drive	Required to install the High-performance Embedded Workshop for the emulator or refer to the emulator user's manual.	

Microsoft, Windows, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and or other countries. All other brand and product names are trademarks, registered trademarks or service marks of their respective holders.

1.3   Components

Check that all of the components are present when unpacking the product. For details on the emulator components, refer to section 1.1 in the additional document, Supplementary Information on Using the H8S/xxxx and Supplementary Information on Using the H8SX/xxxx. If all of the components are not present, contact your nearest Renesas sales office.

Section 2   Emulator Functions

This section describes the emulator functions. They differ according to the device supported by the emulator. For the usage of each function, refer to section 6, Tutorial.

2.1   Overview

Table 2.1 gives a functional overview of the emulator.

For details on the functions of each product, refer to the online help.

Table 2.1 Emulator Functions

No.	Item	Function
1	User program execution function	• Executing user programs at an operating frequency within the range guaranteed for the MCU. • Reset emulation • Step functions: Single step (one step: one instruction) Source-level step (one step: one-line source) Step over (a break did not occur in a subroutine) Step out (when the PC points to a location within a subroutine, execution continues until it returns to the calling function)
2	Reset function	• Issues a power-on reset from the High-performance Embedded Workshop to the MCU during break.
3	Trace functions	• Trace function incorporated in the MCU • Branch trace (conditions and the number of branches differ depending on the MCU)
4	Break functions	• Hardware break function (functions and the number of functions differ depending on the MCU) • PC break function (255 points) • Forced break function

Table 2.1   Emulator Functions (cont)

No.	Item	Function
5	Memory access functions	• Downloading to RAM
		• Downloading to flash memory
		• Single-line assembly
		• Reverse assembly (disassembly)
		• Reading of memory
		• Writing to memory
		• Automatic updating of a display of selected variables during user program execution
		• Fill
		• Search
		• Move
		• Copy
6	General/control register access function	Reads or writes the general/control registers.
7	Internal I/O register access function	Reads or writes the internal I/O registers.
8	Source-level debugging function	Various source-level debugging functions.
9	Command line function	Supports command input.
		Batch processing is enabled when a file is created by arranging commands in input order.
10	Help function	Describes the usage of each function or command syntax input from the command line window.

The specific functions of the emulator are described in the next section.

Note: The number of hardware break conditions, trace conditions, branches, and MCU internal-trace functions differs according to the MCU. For the specifications of a given MCU, refer to the online help or the supplementary document.

2.2   Trace Functions

The emulator has a trace function.

Note: The number of branch instructions that can be acquired by a trace differs according to the
        product. For the number that can be specified for each product, refer to the online help.

2.2.1   Branch Trace Function

The branch source addresses, mnemonics, operands, and source lines are displayed. Since this
function uses the trace buffer built into the MCU, a realtime trace can be acquired.

2.3   Break Function

The emulator has the following three break functions.

(1) Hardware break function (Break condition)

   Uses a break controller incorporated in the MCU.

   The access address, instruction fetch address, data, or bus cycle condition can be set. The
   logical address is the address condition.

   This function can be also set from the [Event] column in the [Editor] or [Disassembly]
   window. For the setting, refer to section 5.2, Downloading a Program.

(2) PC break function (BREAKPOINT)

   Breaks when the dedicated instruction at the specified address that has been replaced is
   executed.

   This function can be set in the [Breakpoint] page of the [Event] dialog box.

   It can also be set when the [S/W breakpoint] column for the line to be set is double-clicked in
   the [Source] or [Disassembly] window.

(3) Forced break function

   Forcibly breaks the user program.

2.4   Memory Access Functions

The emulator has the following memory access functions.

(1) Memory read/write function

[Memory] window: The memory contents are displayed in the window. Only the amount specified when the [Memory] window is opened can be read. Since there is no cache in the emulator, read cycles are always generated. If the memory is written in the [Memory] window, read cycles in the range displayed in the [Memory] window will occur for updating the window. When the [Memory] window is not to be updated, change the setting in [Lock Refresh] from the popup menu.

me (memory_edit) command: A command line function that reads or writes the specified amount of memory at the specified address.

(2) User program downloading function

A load module registered in the workspace can be downloaded. Such module can be selected from [Download Module] in the [Debug] menu. Downloading is also possible by a popup menu that is opened by right-clicking on the mouse at the load module in the workspace. The user program is downloaded to the RAM or flash memory.

This function also downloads information required for source-level debugging such as debugging information.

(3) Memory data uploading function

The specified amount of memory from the specified address can be saved in a file. Select [Save Memory] from the [Debug] menu or select the [Save] item from the popup menu in the [Memory] window.

(4) Memory data downloading function

The memory contents saved in a file can be downloaded. Select [Load] from the popup menu in the [Memory] window.

(5) Displaying the variable contents

The variable contents specified in the user program are displayed.

(6) Other memory operation functions

Other functions are as follows:

- Memory fill
- Memory copy
- Memory save
- Memory verify
- Memory search
- Internal I/O display
- Displaying label and variable names and their contents

For details, refer to the online help.

Note: Memory access during user program execution:
        When memory is accessed from the memory window, etc. during execution of the user
        program, the user program is resumed after it has stopped in the emulator to access the
        memory. Therefore, realtime emulation cannot be performed.
        The stopping time of the user program is as follows:
        Environment:
            Host computer: Core^{TM} 2 Duo 2 GHz
            H8S/2377F: 33 MHz (system clock frequency)
        When a one-byte memory is read from the command-line window, the stopping time will
        be about 65 ms.

2.5   Stack Trace Function

The emulator uses the information on the stack to display the names of functions in the sequence
of calls that led to the function to which the program counter is currently pointing. This function
can be used only when the load modules that have the following formats are loaded:
  – H8SX, H8S, H8 family series C/C++ compiler (such as [R0C40008XSW06R]): Elf/Dwarf2
For the usage of this function, refer to section 6.19, Stack Trace Function.

2.6   Online Help

An online help explains the usage of each function or the command syntax that can be entered
from the command line window.

Select [Emulator Help] from the [Help] menu to view the emulator help.

Section 3   Preparation before Use

3.1   Emulator Preparation

Unpack the emulator and prepare it for use as follows:

⚠ WARNING

READ the reference sections shaded in figure 3.1 before using the emulator product. Incorrect operation will damage the user system and the emulator product. The USER PROGRAM will be LOST.
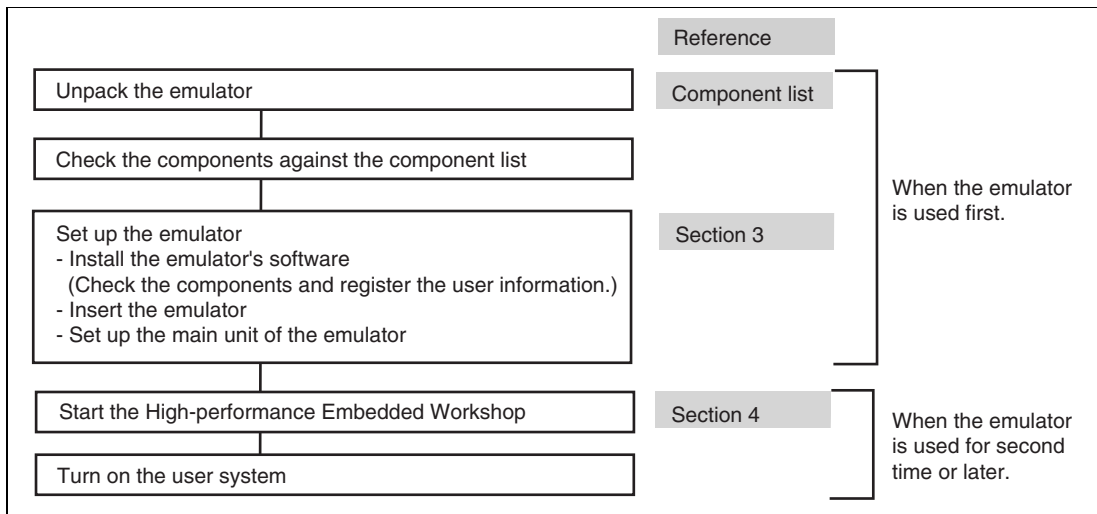
Figure 3.1   Emulator Preparation Flow Chart

3.2   Emulator Hardware Configuration

As shown in figure 3.2, the emulator consists of an emulator, a USB cable, and a user system interface cable. The emulator is connected to the host computer via USB 1.1, and also to the USB port conforming to USB 2.0.

Figure 3.2   Emulator Hardware Configuration (when the 14-pin Type Connector is Used)

The names of each section of the emulator are explained next.

Emulator Top View:
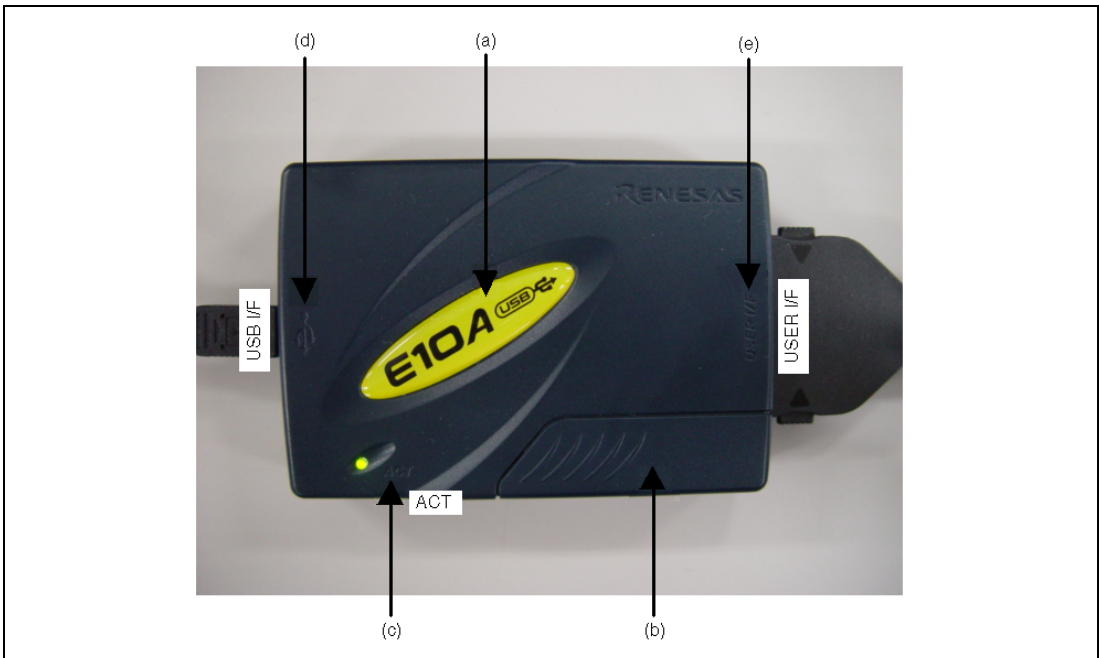
Figure 3.3   Emulator Top View

(a) E10A-USB logo plate: A yellow plate (for HS0005KCU01H) or a red plate (for HS0005KCU02H) dedicated for the emulator is provided to be easily distinguished from other E-series emulators.

(b) Sliding switch cover: A cover to protect switches for setting the emulator, which is closed to prevent incorrect operation. Be sure to close this cover during emulation.

(c) ACTION LED: Marked 'ACT'. When this LED is lit, the E10A-USB control software is in operation.

(d) Host connector: Marked '⟵⟶'. A connector for the host computer is provided at the side of this mark.

(e) User connector: Marked 'USER I/F'. A connector for the user system interface cable is provided at the side of this mark.

Note: Even if the LED is not lit, the USB is not disconnected or malfunctioned.

Emulator Host-side View:
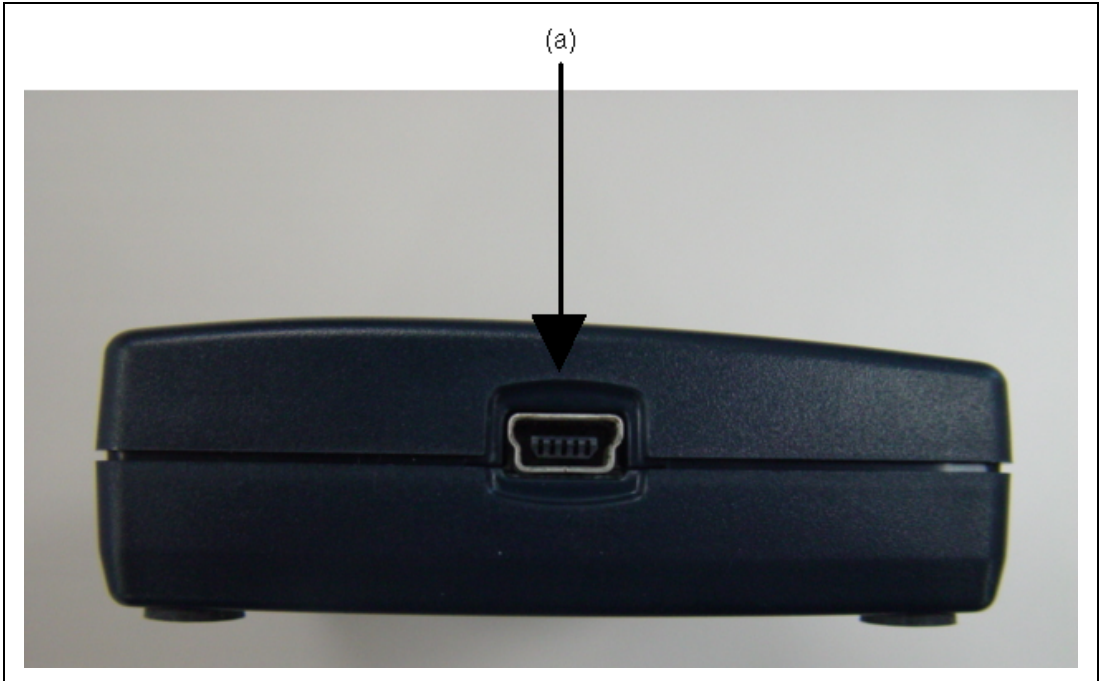
Figure 3.4　Emulator Host-side View

(a) Host-side connector:　A USB connector for the host computer. Be sure to connect the provided USB cable.

Emulator User-side View:
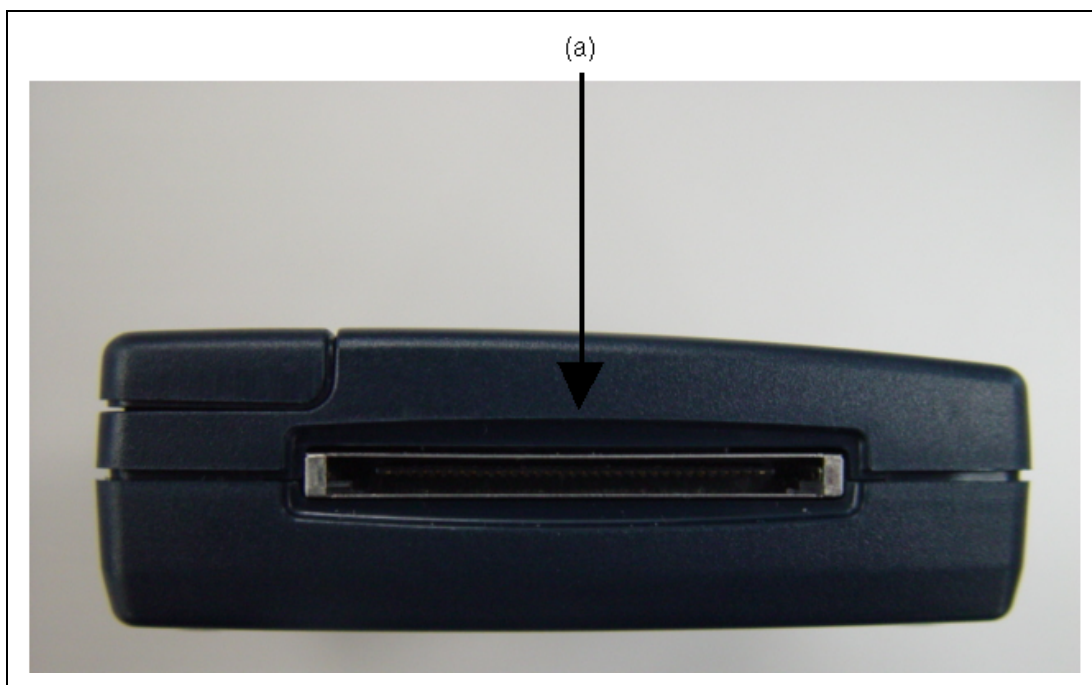
Figure 3.5   Emulator User-side View

(a) User-side connector: A user system interface cable is connected.

Emulator Bottom View:
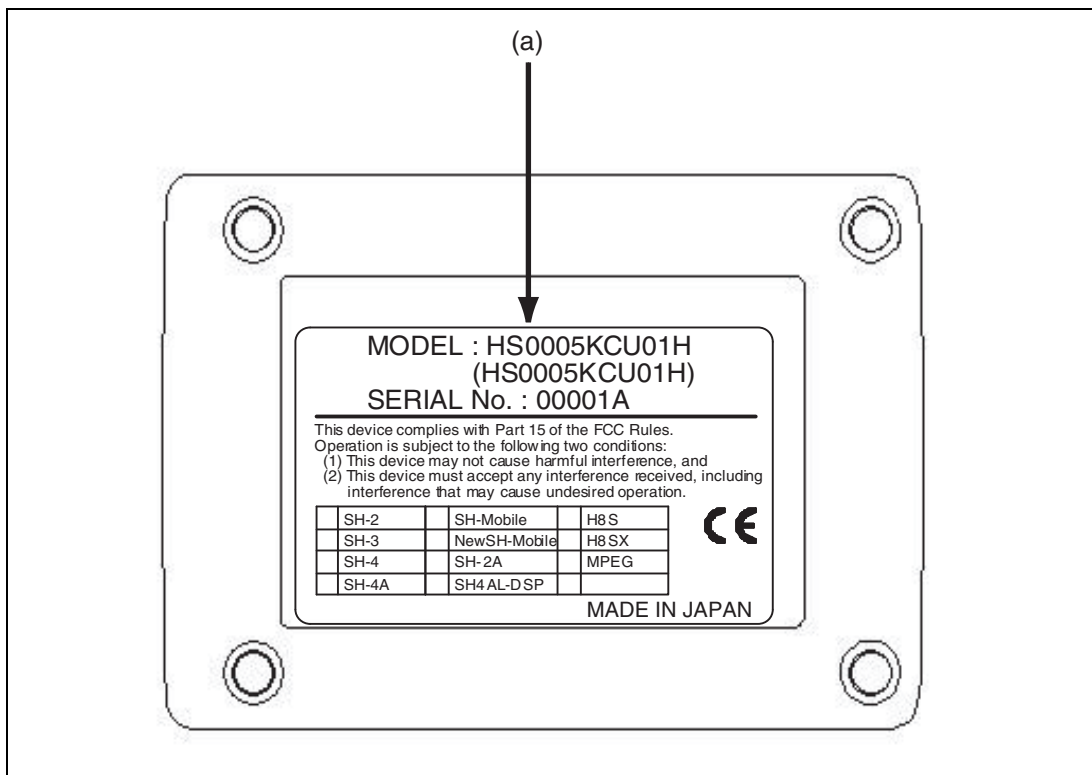
Figure 3.6　Emulator Bottom View

(a) Label for product management: The serial number, revision, and safety standard, etc. of the emulator are written to. The contents differ depending on the time when you purchased the product.
Only one device group can be set up using the setup tool when the emulator is purchased. Be sure to check the device group you have selected on the label for product management.

3.3   CD-R

The root directory of the CD-R contains a setup program for installing the emulator's software.
The folders contain the files and programs listed below.

Table 3.1   Contents of the CD-R Directories

Directory Name	Contents	Description
Dlls	Microsoft® runtime library	A runtime library for the High-performance Embedded Workshop. The version is checked at installation and this library is copied to the hard disk as part of the installation process.
Drivers	E10A-USB emulator driver	USB drivers for the E10A-USB emulator.
Help	Online help for the E10A-USB emulator	An online help file. This is copied to the hard disk as part of the installation process.
Manuals	E10A-USB emulator manuals	E10A-USB emulator user's manuals. They are provided as PDF files.

3.4   Installing Emulator's Software

Execute HewInstMan.exe from the root directory of the CD-R to start the installation manager.

Follow the cues given by the installation manager to install the software.

Note: When a driver is installed in Windows® XP, a warning message on the Windows® logo
        test may be displayed, but it is not a problem. Select [Continue Anyway] to proceed with
        driver installation.
        Download the latest version of "Renesas E-Series USB driver" from the following URL
        and install the USB driver.
        http://www.renesas.com/products/tools/emulation_debugging/onchip_debuggers/e10ausb/
        downloads.jsp

3.5   Connecting the Emulator to the Host Computer

This section describes how to connect the emulator to the host computer. For the position of each connector of the emulator, refer to section 3.2, Emulator Hardware Configuration.

Note: Be sure to install the software for the emulator before putting the emulator in place.

⚠ WARNING

**Always switch OFF the emulator product and the user system before connecting or disconnecting any CABLES except for the USB interface cable. 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.**

The emulator is connected to the host computer via the USB 1.1, and also to the USB port conforming to USB 2.0. Figure 3.7 shows the system configuration.
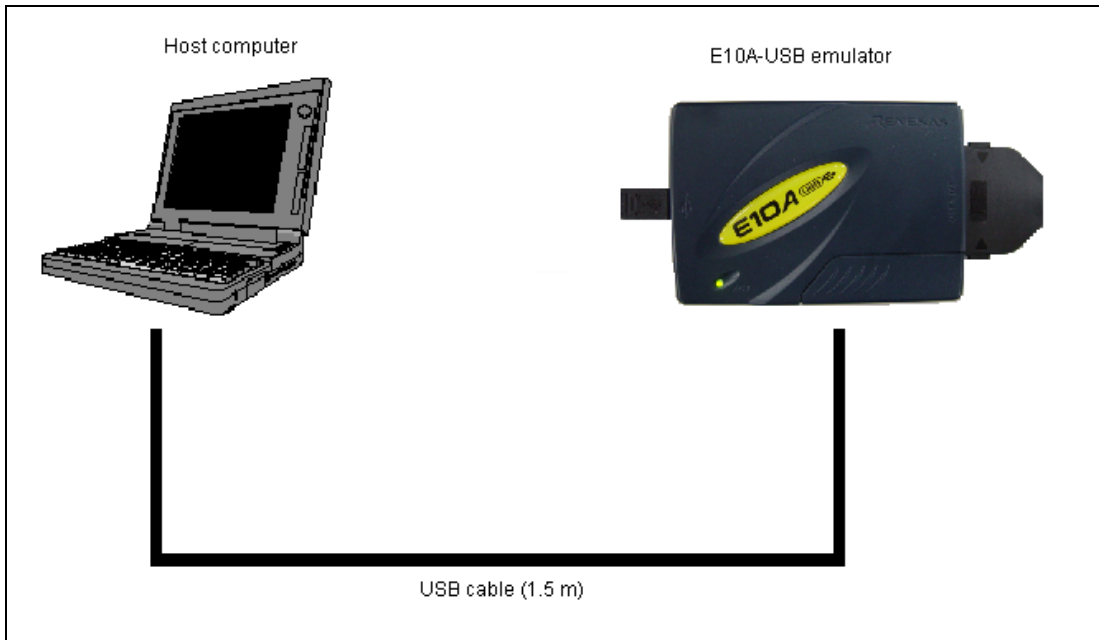
Figure 3.7　System Configuration when Connecting the Emulator to the Host Computer

3.6   Connecting the Emulator to the User System

Use the procedure below 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.

1. Check that the host computer is turned off or the emulator is not connected to the host computer with the USB cable.
2. Connect the user system interface cable to the user-side connector of the emulator.
3. Connect the USB cable to the host-side connector of the emulator.

Figure 3.8 shows the position of the connector.
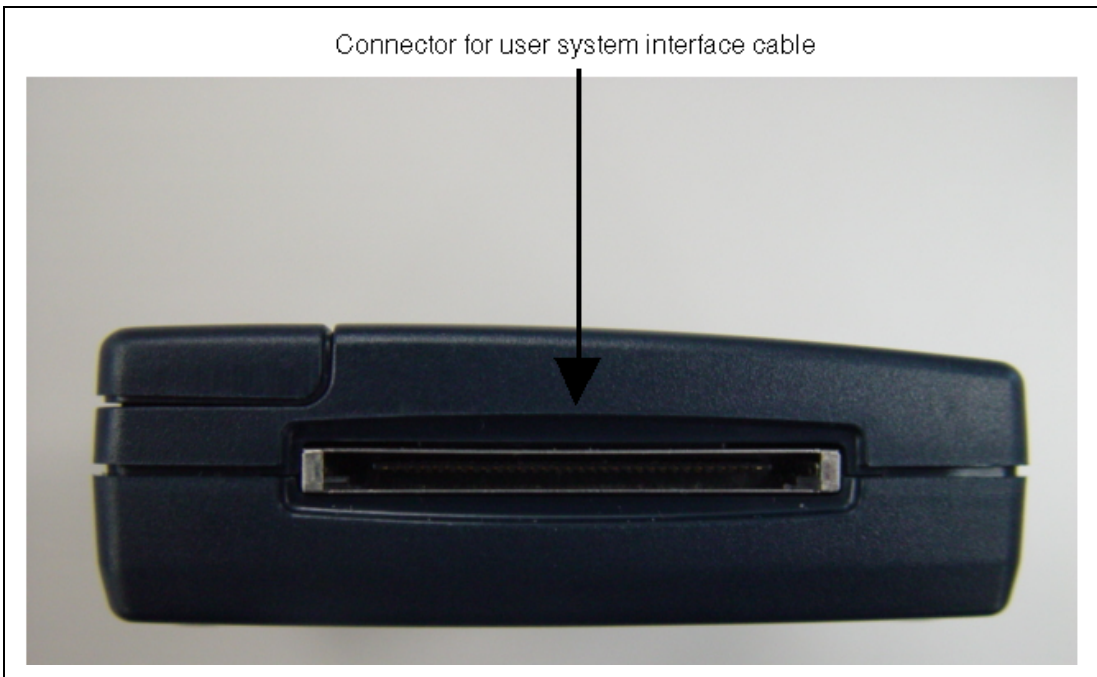
Figure 3.8   Position of the Connector

(1) The user system connector must be installed to the user system. Table 3.2 shows the recommended user system connector for the emulator.

Table 3.2   Recommended User System Connector

Connector	Type Number	Manufacturer	Specifications
14-pin connector	2514-6002	3M Limited	14-pin straight type

Note: When designing the 14-pin connector layout on the user board, do not place any components within 3 mm of the user system connector.

(2) The pin assignments of the user system connector are shown in section 1 in the additional document, Supplementary Information on Using the H8S/xxxx, or Supplementary Information on Using the H8SX/xxxx.

(3) Connect pins 8, 9, 10, 12, 13, and 14 (when using the 14-pin user system interface cable) of the user system connector to GND firmly on the PCB. These pins are used as electrical GND and to monitor the connection of the user system connector. Note the pin assignments of the user system connector.
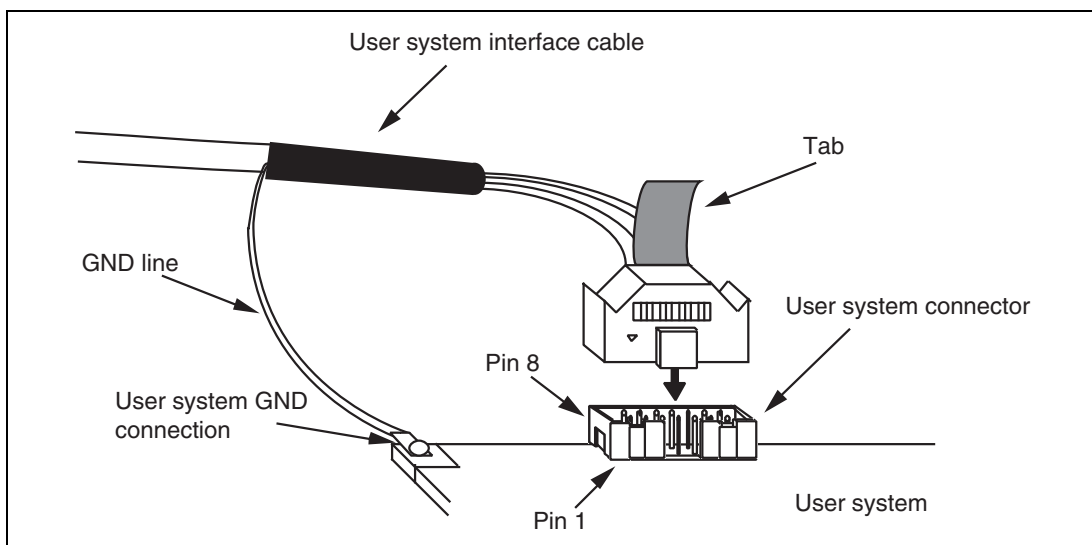
Figure 3.9   Connecting the User System Interface Cable to the User System when the 14-pin Type Connector is Used

CAUTION

Note that the pin number assignments of the user system connector differ from those of the connector manufacturer.

Notes:　1.　Connection of the signals differs depending on the package. For details, refer to the MCU pin assignments.

2.　To remove the 14-pin type user system interface cable from the user system, pull the tab on the connector upward.

3.　The range of communication that the emulator operates at is different depending on the MCU used.

4.　To connect the signals from the connector, refer to section 1 in the additional document, Supplementary Information on Using the H8S/xxxx, or Supplementary Information on Using the H8SX/xxxx.

3.7   Connecting System Ground

⚠ **WARNING**
Separate the frame ground from the signal ground at the user system. 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 emulator's signal ground is connected to the user system's signal ground. In the emulator, the signal ground and frame ground are connected. In the user system, connect the frame ground only; do not connect the signal ground to the frame ground (figure 3.10).

If it is difficult to separate the frame ground from the signal ground in the user system, set the GND for DC power input (AC adapter) of the host computer and the frame ground of the user system as the same potential. If the GND potential is different between the host computer and the user system, an overcurrent will flow in the low-impedance GND line and thin lines might be burned.
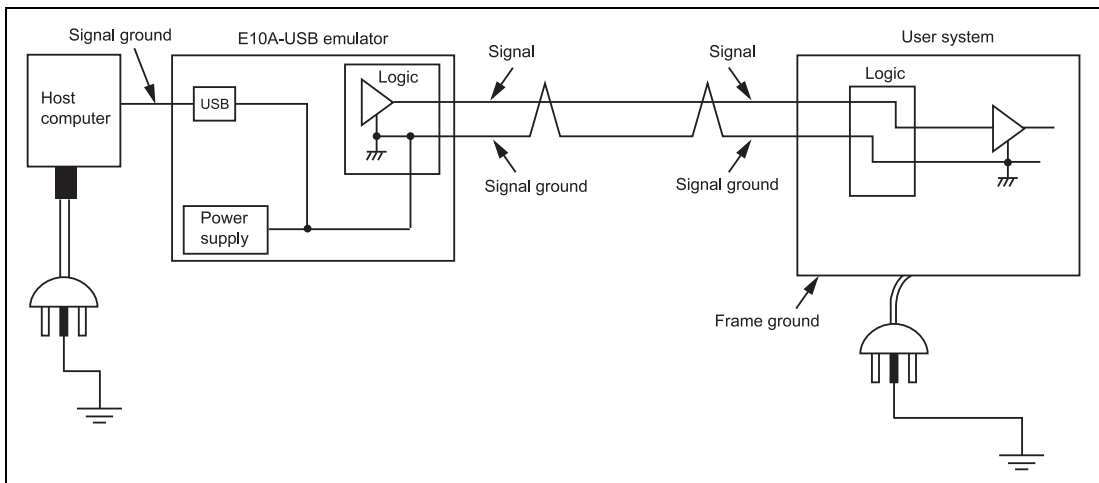
Figure 3.10   Connecting System Ground

3.8   Setting the DIP Switches

⚠ WARNING

Do not change switches (SW2 and SW3) while the emulator and the user sytem are turned on. The changing of switches (SW2 and SW3) will result in a FIRE HAZARD and will damage the user system and the emulator product. The USER PROGRAM will be LOST.

The emulator incorporates three switches; set all of them to 1 in general, and set SW1 to 0 only when the emulator is set up.
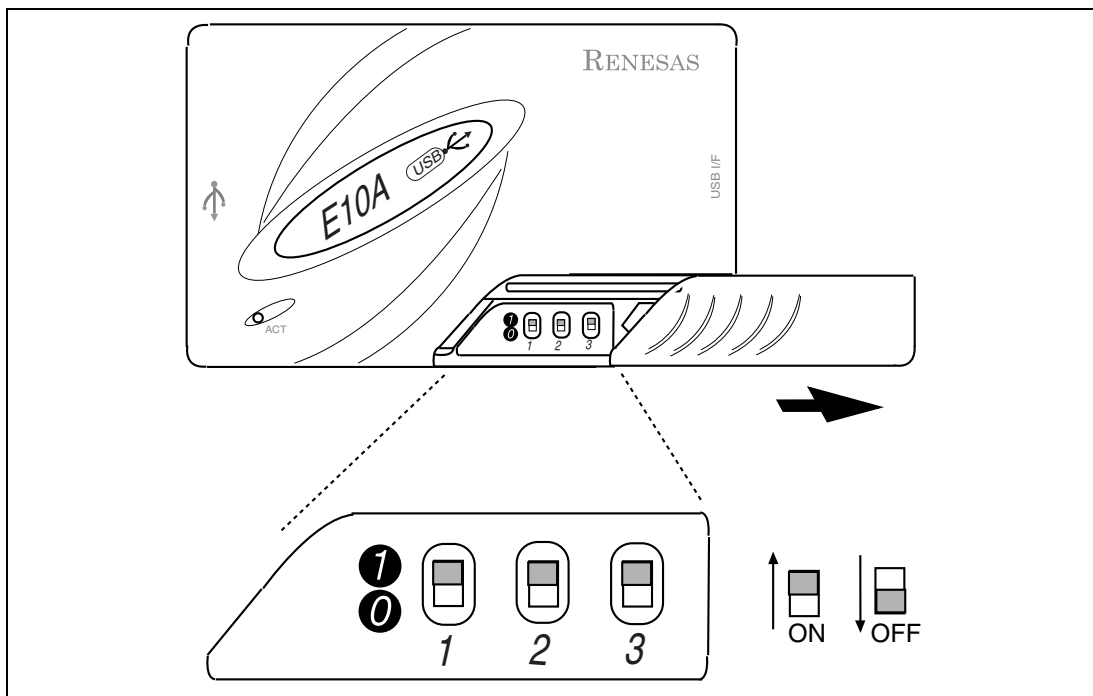
Figure 3.11   DIP Switches

Table 3.3 shows the relationships between settings and functions of DIP switches 1 to 3. Use the settings depending on the usage of the user system.

<div style="border:1px solid">

CAUTION

Use only the settings shown in table 3.3. Use of other settings will not activate the emulator. The USER PROGRAM will be LOST.

</div>

- Settings for use of the 14-pin user interface cable

Description: The settings of the UVCC pin of the user interface connector apply to SW2. In the H8S, H8SX family E10A-USB emulator, connect the UVCC pin to the VCC power supply that is connected to the VCC pin of the MCU for the user system.

Table 3.3　Switch Settings of the E10A-USB

Switch Settings			State of the E10A-USB	
SW1	SW2	SW3	UVCC	Condition
0 (off)	1	1	-	The emulator is set up
1 (on)	1 (on)	1 (on)	1.8 V to 5.0 V*	Normal

Note:　The range differs depending on the MCU.

RENESAS

3.9   Interface Circuits in the Emulator

Figure 3.12 shows interface circuits in the emulator. Use them as a reference to determine the value of the pull-up resistance.

Note: The 74LVC2G125 operates at VCC (1.8 to 5.0 V) from the user system connector.

Notes: 1. In the H8S, H8SX family, these pin names differ depeding on the MCU. Refer to each additional document.
       2. When the H8S, H8SX family is connected, the emulator outputs the reset signal to pin 4. Therefore, pin 4 cannot be directly
          connected to pin RES# of the MCU.
       3. When the H8S, H8SX family is connected, the emulator only monitors pin 7.
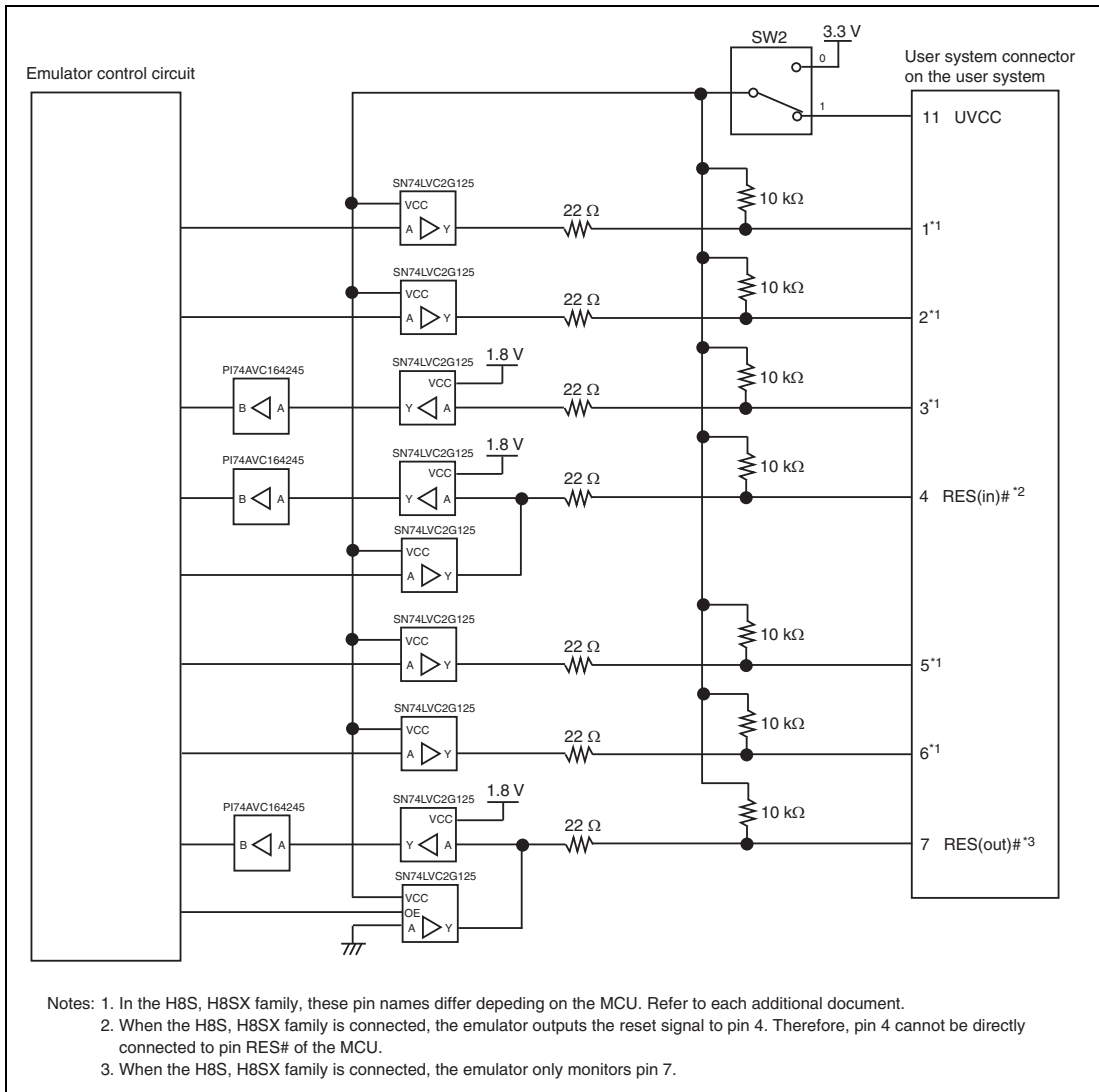
Figure 3.12   Interface Circuits in the Emulator

3.10   Setting up the Emulator

Set up the emulator's firmware using the following procedures. Here describes how to set up the H8S device group.

Note: Only one device group can be set up using the setup tool when the emulator is purchased. Be sure to check the device group you have selected on the label for product management attached to the back of the emulator box. To use the emulator for another device group after set up, purchase the license tool to add a device group.

CAUTION

Do not disconnect the USB cable unless instructed to do so by an on-screen message. Incorrect operation will damage the emulator product.

3.10.1   Setting up at Purchasing the Emulator or Updating the Version of Software

Note: If you are using the HS0005KCU01H (serial No.: 03311C or later) or HS0005KCU02H
         (serial No.: 04146E or later) emulator hardware, the below procedure may not be required;
         follow the procedure only when the dialog box shown in figure 3.13 or 3.14 is displayed
         by using the procedure described in section 3.11, System Check.

Figure 3.13   [Emulator firmware error (1)] Dialog Box

Figure 3.14   [Emulator firmware error (2)] Dialog Box

(1)   Open the sliding switch cover and check that the switch (SW1) for setting the emulator is
       turned to '1'.

(2)   Select [Renesas] -> [High-performance Embedded Workshop] -> [Setup tool for E10A-USB
     Emulator] -> [H8S/xxxx Device Group] from [Programs] in the [Start] menu. A tool for setting
     up the emulator is activated.

Note: When the H8SX device group is used, select [H8SX Device Group].
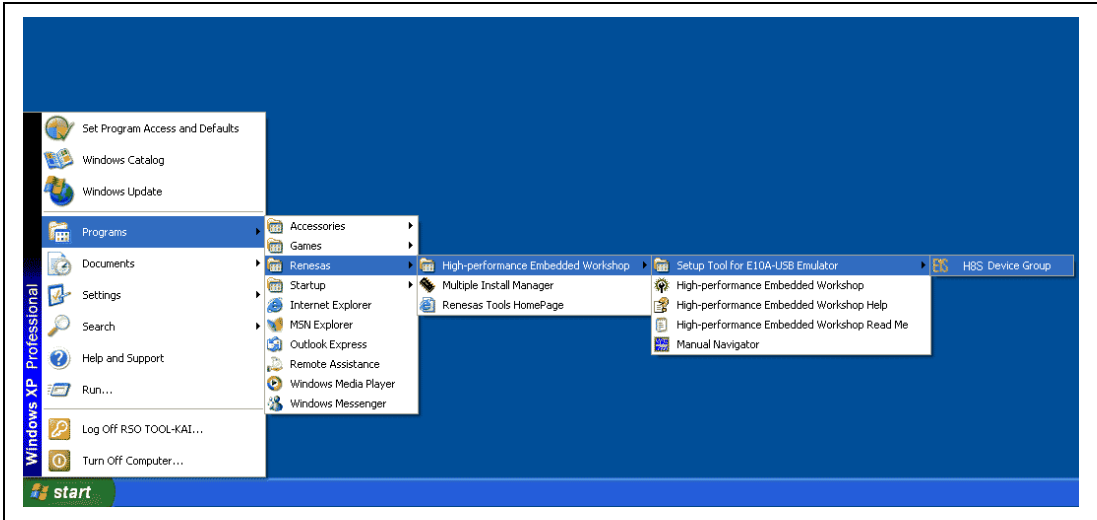
Figure 3.15   [Start] Menu
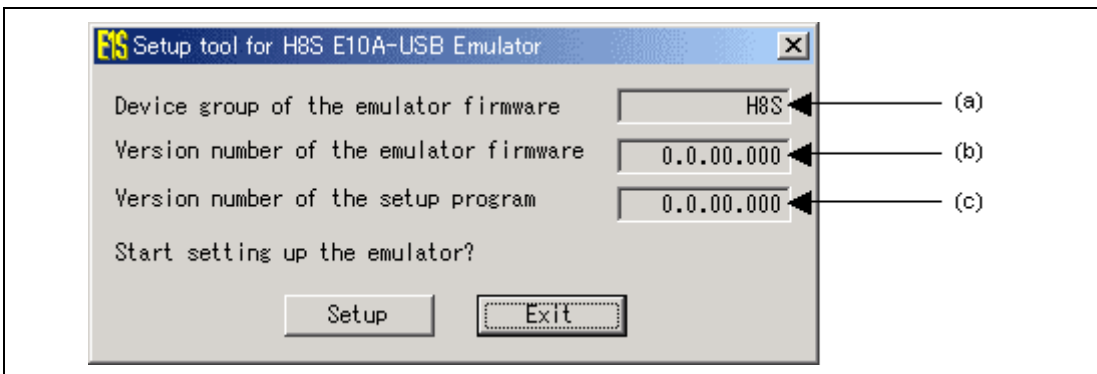
Figure 3.16   Setup Tool for Emulator

(a) Device group of the emulator firmware: Name of the device group currently set.

(b) Version number of the emulator firmware: The version number of software for controlling the
                                                 H8S Device Group in the emulator. This item is
                                                 displayed only when the H8S Device Group is
                                                 available.

(c) Version number of the setup program: The version number of the setup program.

Notes: 1. If the version numbers shown in (b) and (c) are the same, setup of the emulator is not required. Setup the emulator only when "-.-.--.---" is shown in (b) or the version number of (b) is older than that of (c).

2. If an emulator other than the H8S/xxxx E10A-USB is connected, the following error message will be displayed to exit the setup tool.

Figure 3.17   Error Message (1)

3. If the following error message is displayed, the host computer is not connected to the emulator or the setup switch (SW1) is turned to '0'.

Figure 3.18   Error Message (2)

If the setup switch (SW1) is turned to '0', set it to '1' and connect the USB cable again.

(3)   Clicking the [Setup] button displays the following dialog box.
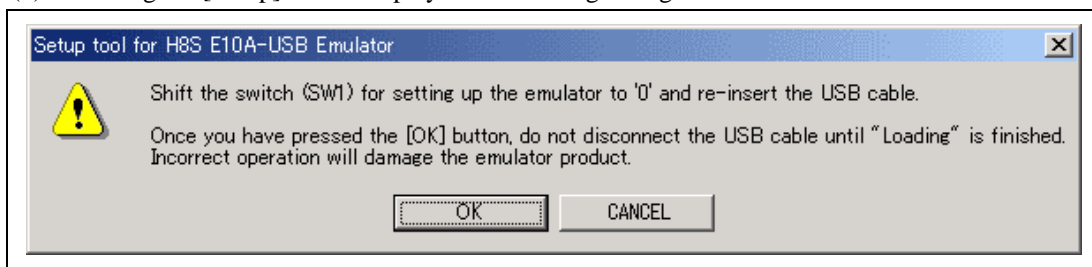
Figure 3.19   [Setup tool for H8S E10A-USB Emulator] Dialog Box

(4)   Turn the setup switch (SW1) to '0', connect the USB cable again, and click the [OK] button.
      Setting up the emulator's firmware is started.

Notes: 1. If the following dialog message is displayed, insert the USB cable again.
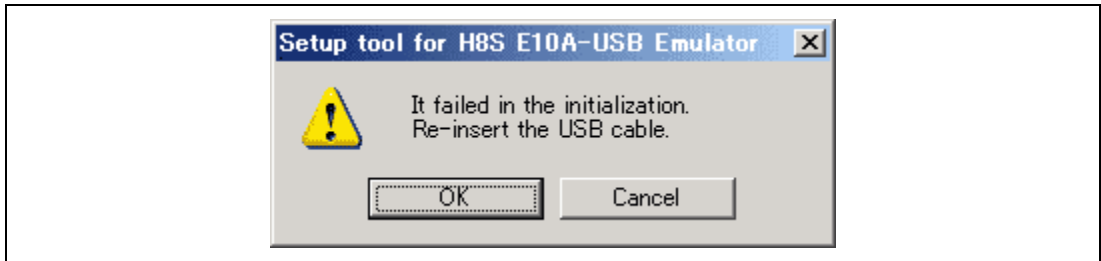
Figure 3.20   [Setup tool for H8S E10A-USB Emulator] Dialog Box (3)

2. When [Add New Hardware Wizard] is displayed, select [Install the software
    automatically].
    The above case only applies when the emulator is setup initially or the USB port on the
    host computer side is changed and used for first connection. The time may be required
    until [Add New Hardware Wizard] is displayed.
3. Although a dialog box will be displayed to indicate disconnection of the USB, this is
    not a problem.

CAUTION

**Do not turn off the host computer or disconnect the USB
cable while setting up the emulator. The emulator may be
damaged.**
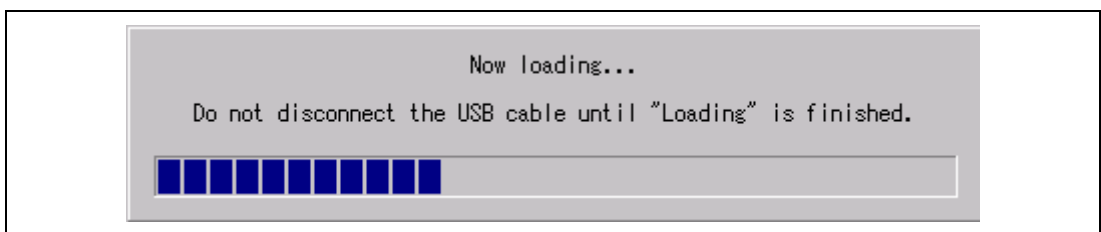
Figure 3.21   Start of Setting up the Emulator

(5)   When the following dialog box is displayed, setting up the emulator is completed.

Figure 3.22   Message for Completion of Setting up the Emulator

(6)   When setting up the emulator has been completed, the following message will be displayed. Turn the setup switch (SW1) to '1', connect the USB cable again, and click the [OK] button.
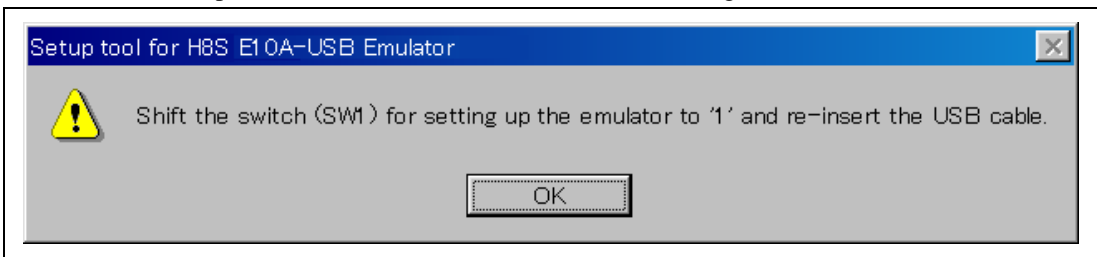
Figure 3.23   [Setup Tool for H8S E10A-USB Emulator] Dialog Box

Notes: 1. Be sure to turn the setup switch (SW1) to '1' except when the setup tool is used.
        2. To use the license tool for another device group, it is necessary to set up the firmware by using the setup tool or license tool that corresponds to the device group. To use the setup tool, however, the license tool must also be installed. For details on the license tool, refer to the following section.

3.10.2   Setting up the Emulator by Using the License Tool to Add a Device Group

In the license tool, the emulator for the current product group can be used for debugging another product group (device group such as SHxxxx Device Group, H8S Device Group, or H8SX Device Group that is supported by the emulator).

In this section, the names of the device group that have already been installed and another device group to be added by the license tool are shown as SHxxxx Device Group and H8S Device Group, respectively. Replace these names according to your environment when you read this section.

The license tool to add a device group does not include software for the E10A-USB emulator.

Install the software for your product group by using the CD-R provided for the emulator or accessing the data on the web site.

Note: If no product groups have been installed in the emulator at the time of purchase, do not use the license tool.
Refer to section 3.10.1, Setting up at Purchasing the Emulator or Updating the Version of Software, and use the setup tool.

(1) Installing the emulator
Inserting the CD-R into the CD-ROM drive automatically activates the installation wizard (to prevent automatic activation, insert the CD-R by pressing the Shift key). If the installation wizard is not automatically activated, execute setup.exe from the root directory of the CD-R.

Follow the instructions by the installation wizard.

(2) Setting up the emulator

1. Open the sliding switch cover and check that the switch (SW1) for setting the emulator is turned to '1'.

2. Select [Renesas] -> [License tool for E10A-USB] -> [H8S Device Group] from [Programs] in the [Start] menu. This starts up the license tool to add a device group to the emulator.
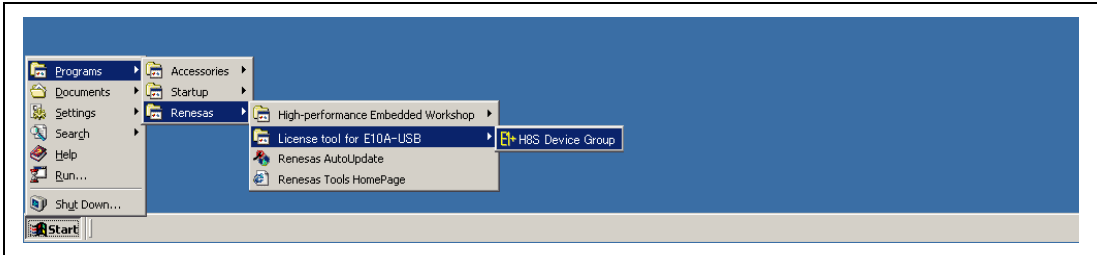
Figure 3.24   [Start] Menu
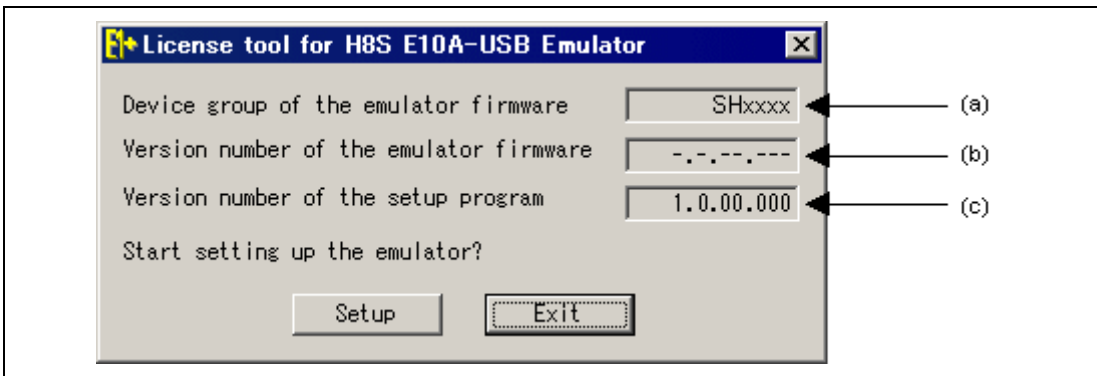
Figure 3.25   License Tool for Emulator

(a) Device group of the emulator firmware: Name of the device group currently set.

(b) Version number of the emulator firmware: The version number of software for controlling the H8S Device Group in the emulator. This item is displayed only when the H8S Device Group is available.

(c) Version number of the setup program: The version number of the setup program.

Notes: 1. If the version numbers shown in (b) and (c) are the same, setup of the emulator is not required. Set up the emulator only when "-.-.--.---" is shown in (b) or the version number of (b) is older than that of (c).

2. If the following error message is displayed, the host computer is not connected to the emulator or the setup switch (SW1) is turned to '0'.
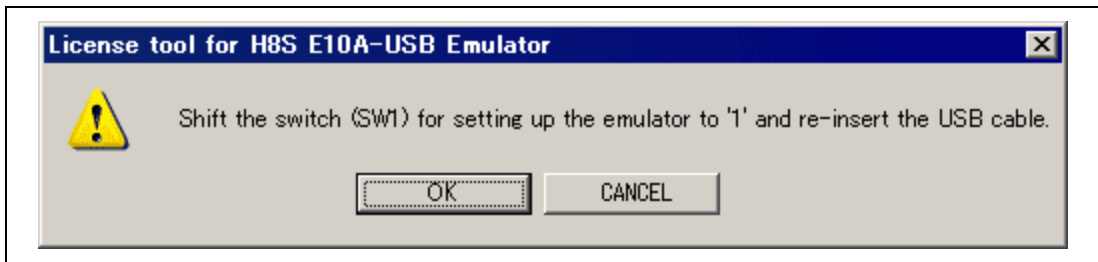
Figure 3.26　Error Message (4)

If the setup switch (SW1) is turned to '0', set it to '1' and connect the USB cable again.

(3)　Click the [Setup] button.
When the following dialog box is displayed, turn the setup switch (SW1) to '0', connect the USB cable again, and click the [OK] button.
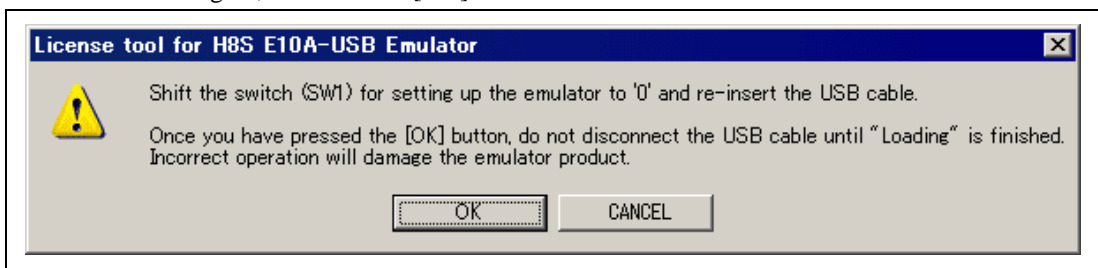
Figure 3.27　[License tool for H8S E10A-USB Emulator] Dialog Box

RENESAS

(4)   Setting up the emulator's control software is started.

Notes: 1.   If the following dialog message is displayed, insert the USB cable again.

Figure 3.28   [License tool for H8S E10A-USB Emulator] Dialog Box

2. When [Add New Hardware Wizard] is displayed, select [Install the software automatically].
    The above case only applies when the USB port on the host computer side is changed and used for first connection. The time may be required until [Add New Hardware Wizard] is displayed.
3. Although a dialog box will be displayed to show the disconnection of the USB, this is not a problem.

CAUTION

Do not turn off the host computer or disconnect the USB cable while setting up the emulator. The emulator may be damaged.
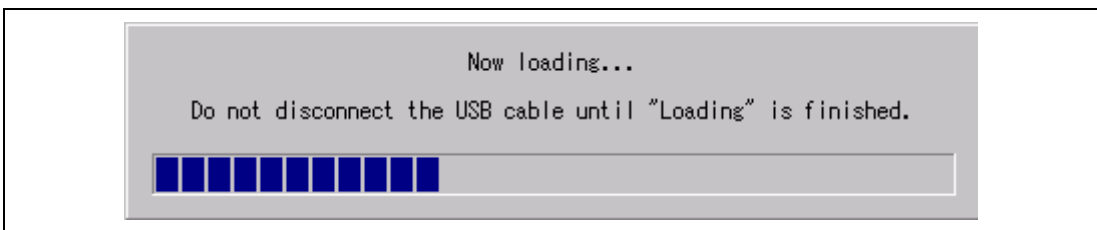
Figure 3.29   Start of Setting up the Emulator

(5)   When the following dialog box is displayed, setting up the emulator is completed.

Figure 3.30   Message for Completion of Setting up the Emulator

(6)   When setting up the emulator has been completed, the following message will be displayed.
         Turn the setup switch (SW1) to '1', connect the USB cable again, and click the [OK] button.
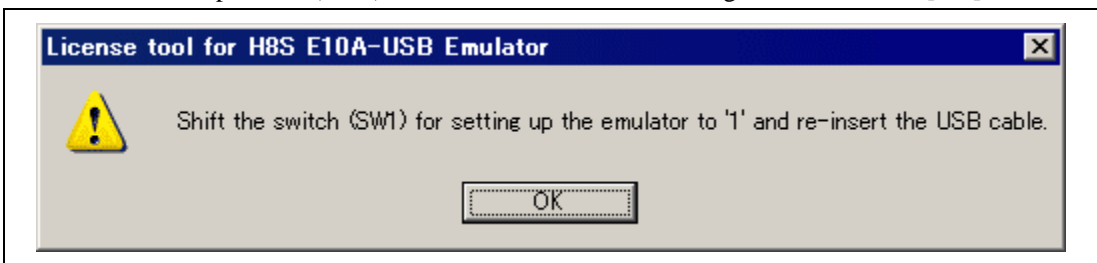
Figure 3.31   [License tool for H8S E10A-USB Emulator] Dialog Box

Notes: 1. Be sure to turn the setup switch (SW1) to '1' except when the license tool for adding
           device groups is used.
      2. To use the license tool for another device group, in which the license has been installed
           in the emulator, it is necessary to set up the firmware by using the setup tool or license
           tool that corresponds to the device group. For details on the setup tool, refer to section
           3.10, Setting up the Emulator, in this manual.
      3. After you have added a device group by using the license tool, place the attached
           device-group sticker onto the back of the emulator box. Otherwise, the emulator may
           not be considered as a target product when repair is required. The license tool provides
           a license to use the target device group. Be sure to acquire a license for each of the
           E10A-USB emulators being used with that device group.

3.11   System Check

When the software is executed, use the procedure below to check that the emulator is connected correctly. Here, use the workspace for a tutorial provided on the product.

Refer to section 4, Preparations for Debugging, for the other activating method to create a new project or use an existing workspace.

1. Connect the emulator to the host computer.
2. Connect the user system interface cable to the connector of the emulator.
3. Connect the user system interface cable to the user system connector.
4. Select [Renesas] -> [High-performance Embedded Workshop] -> [High-performance Embedded Workshop] from [Programs] in the [Start] menu.
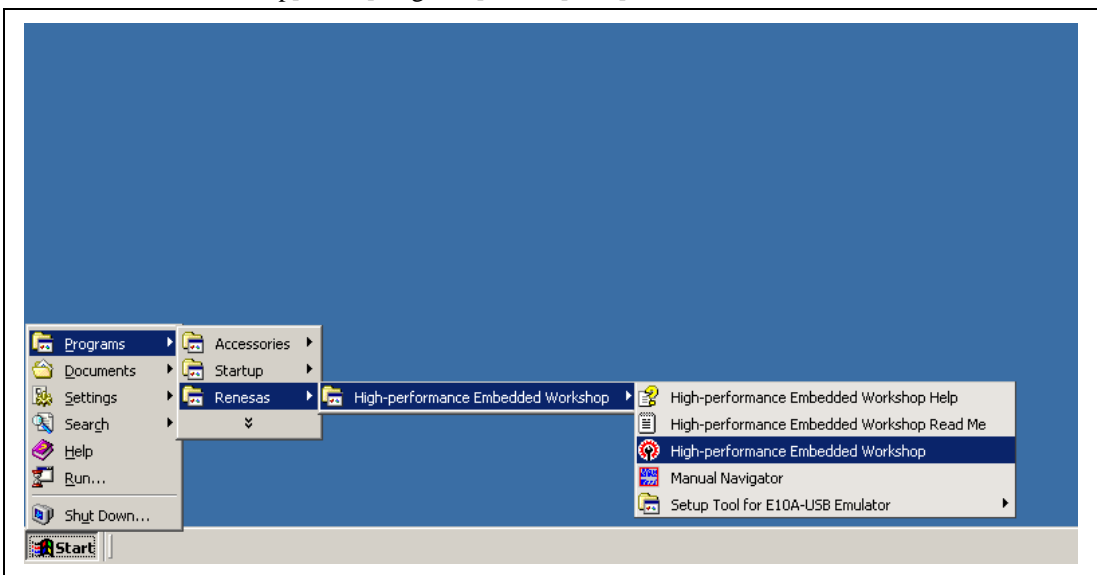
Figure 3.32   [Start] Menu

Note: The [High-performance Embedded Workshop] -> [Tools] is not displayed depending on the user's environment.

5. The [Welcome!] dialog box is displayed.
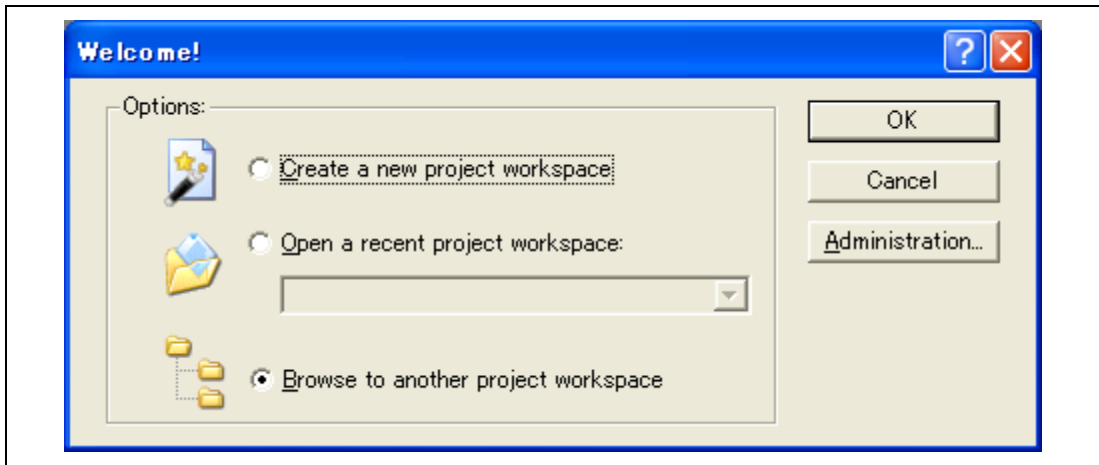
Figure 3.33   [Welcome!] Dialog Box

[Create a new project workspace] radio button: Creates a new workspace.

[Open a recent project workspace] radio button: Uses an existing workspace and displays
                                                        the history of the opened workspace.

[Browse to another project workspace] radio button: Uses an existing workspace; this radio
                                                        button is used when the history of the
                                                        opened workspace does not remain

To use a workspace for the tutorial, select the [Browse to another project workspace] radio button
and click the [OK] button.

When the [Open workspace] dialog box is opened, specify the following directory:
<Drive where the OS has been installed>:
\WorkSpace\Tutorial\E10A-USB\H8S\xxxx\Tutorial

Here, 'xxxx' means the target product group.

Note: When the H8SX product group is used, specify \E10A-USB\H8SX\xxxx\Tutorial. The
        following explains the case of H8S.

After the directory has been specified, select the following file and click the [Open] button.
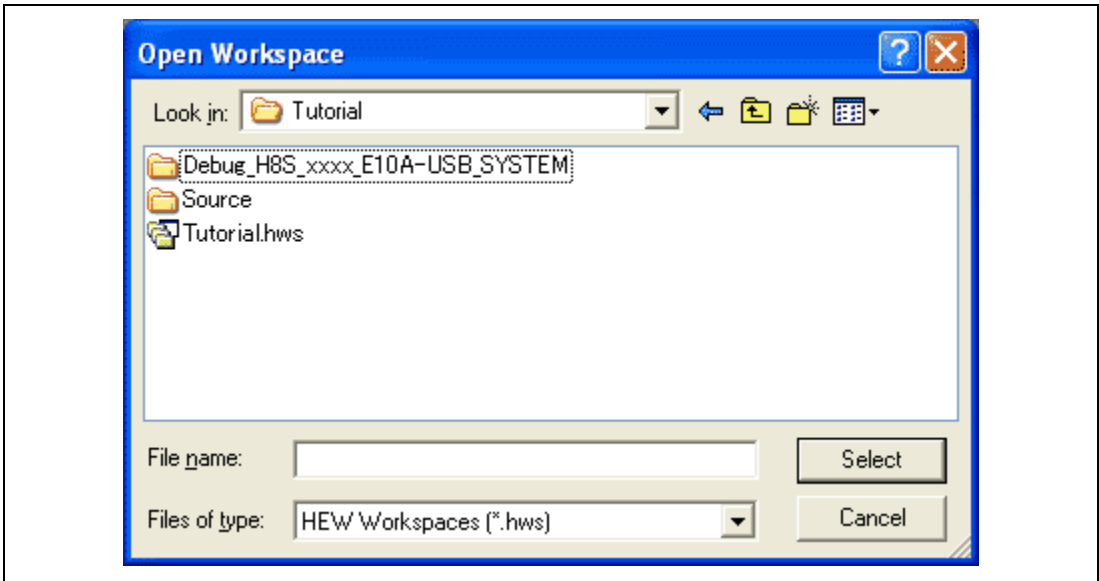
Figure 3.34   [Open Workspace] Dialog Box

6. The [Select Emulator mode] dialog box is displayed.
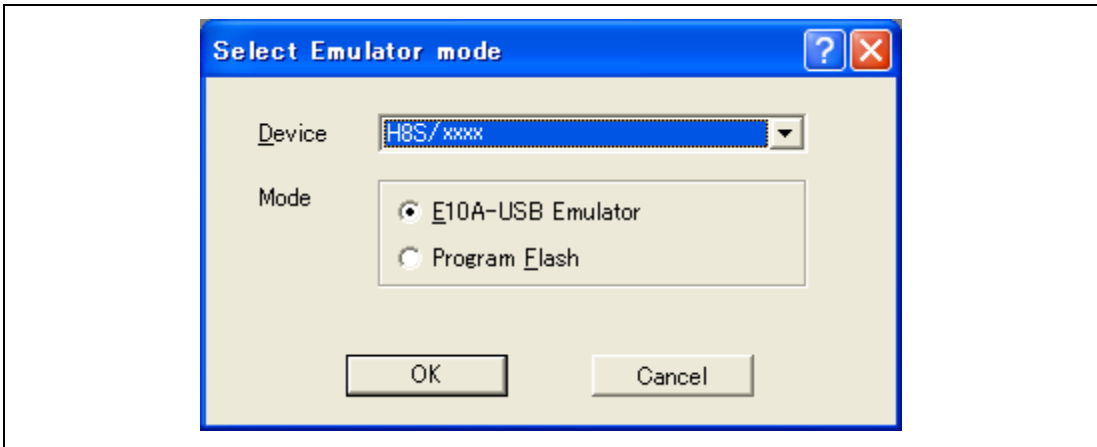
Figure 3.35　[Select Emulator mode] Dialog Box

Select the MCU name in use from the [Device] drop-down list box. The following items are selected in the [Mode] group box.

— E10A-USB Emulator

The E10A-USB emulator for the specified MCU is activated. Debugging the program is enabled.

— Program Flash

The user program is programmed to the internal flash memory. Debugging the program is disabled. To download the load module, register it in the workspace.

7. The [Connecting] dialog box is displayed and the emulator connection is started.
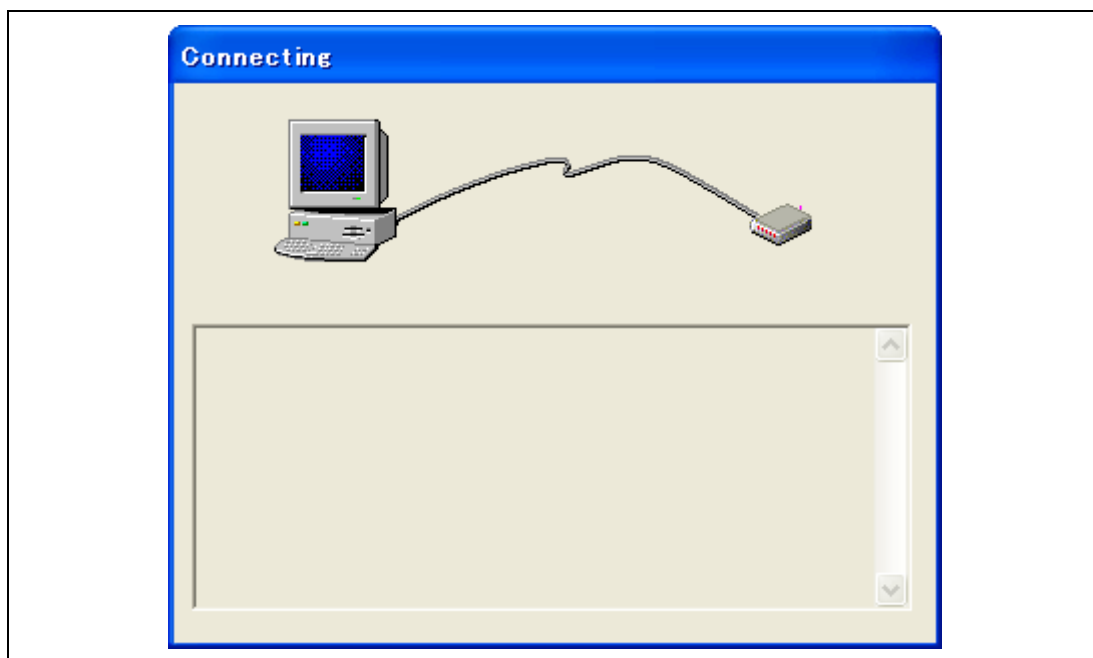
Figure 3.36   [Connecting] Dialog Box

8. The dialog box shown in figure 3.37 is displayed if no product groups have been installed in the emulator at the time of purchase or if the H8S or H8SX license has been installed in the emulator but the emulator firmware has been set up for a different device group. The dialog box shown in figure 3.38 is displayed if an old version of the emulator firmware has been set up in the emulator. Clicking the [OK] button sets up the emulator firmware.
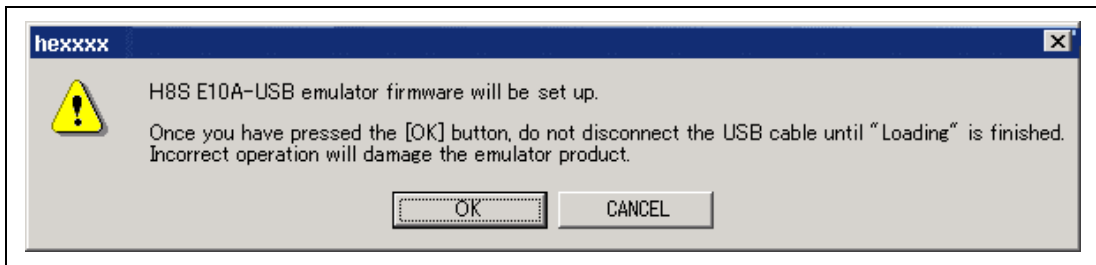
Figure 3.37 Dialog Box to Confirm Setting up of the Emulator Firmware
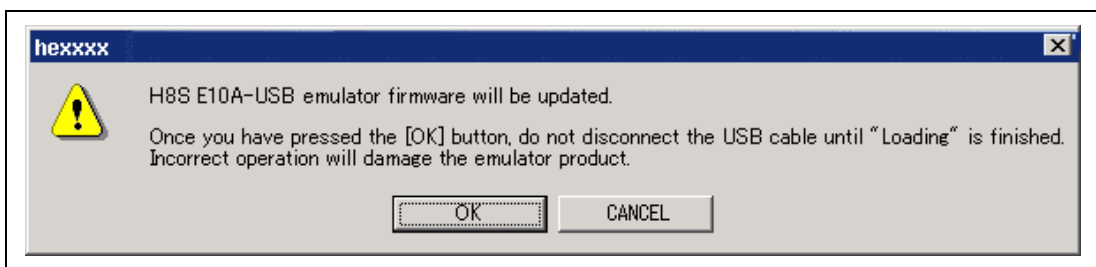
Figure 3.38 Dialog Box to Confirm Updating of the H8S Emulator Firmware

CAUTION

The USB cable must not be disconnected until writing is complete.
Early disconnection may damage the emulator.

Note: The above dialog boxes are only displayed if you are using the HS0005KCU01H (serial No.: 03311C or later) or HS0005KCU02H (serial No.: 04146E or later) emulator hardware. In this case, follow the procedure described in section 3.10.1, Setting up at Purchasing the Emulator or Updating the Version of Software.

9. The dialog box is displayed as shown in figure 3.39 or figure 3.40.
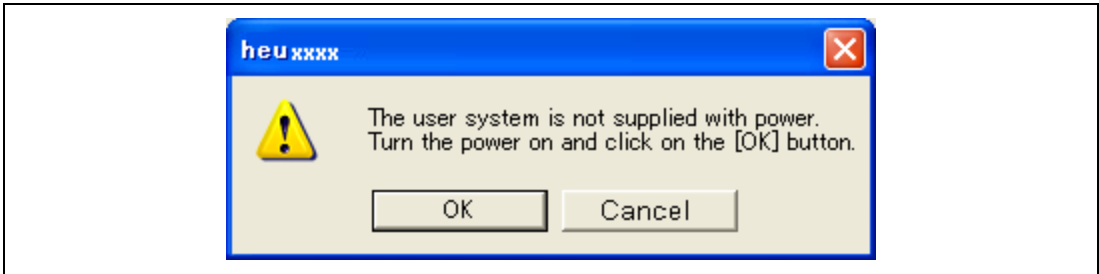
Figure 3.39　[Turn Power on] Message Box
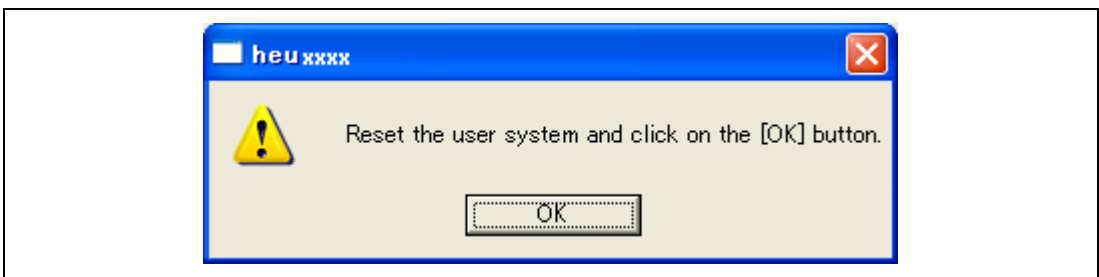
Figure 3.40　Dialog Box of the Power-on Request Message

10. Turn on the user system.
11. When using the H8S family, the dialog box shown in figure 3.41 is displayed. Input the system clock value.
    The system clock is the input clock.
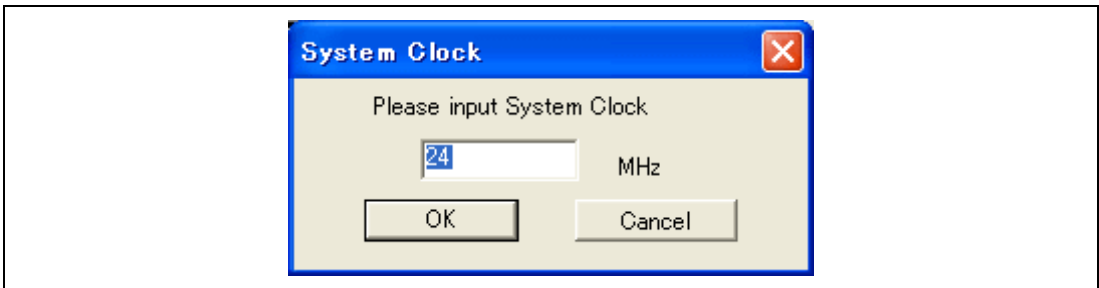
Figure 3.41　[System Clock] Dialog Box

12. Set an eight-digit hexadecimal ID code as a security code for the flash memory. To read the data that has been written to the internal flash memory, input this ID code when the [E10A-USB Emulator] is selected in the [Select Emulator mode] dialog box and the [New ID code] check box is unselected on activating the emulator. If the ID code is not matched or the [New ID code] check box is selected, the flash memory contents are erased.
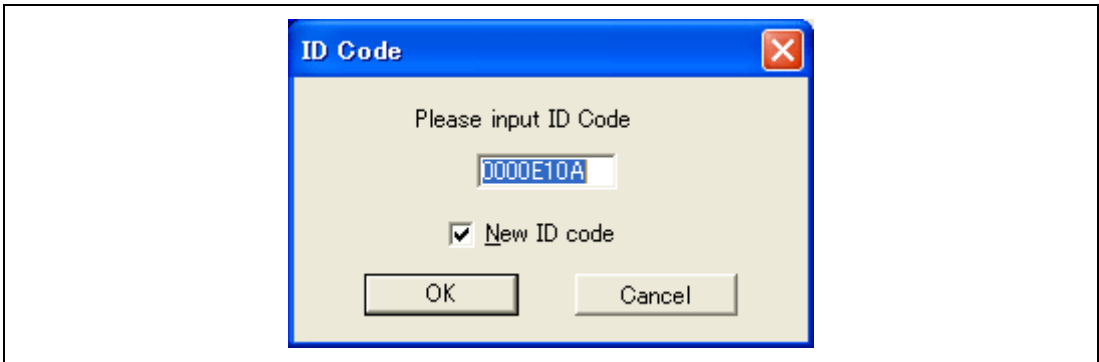
Figure 3.42　[ID Code] Dialog Box

13. When "Connected" is displayed in the [Output] window of the High-performance Embedded Workshop, the emulator initiation is completed.
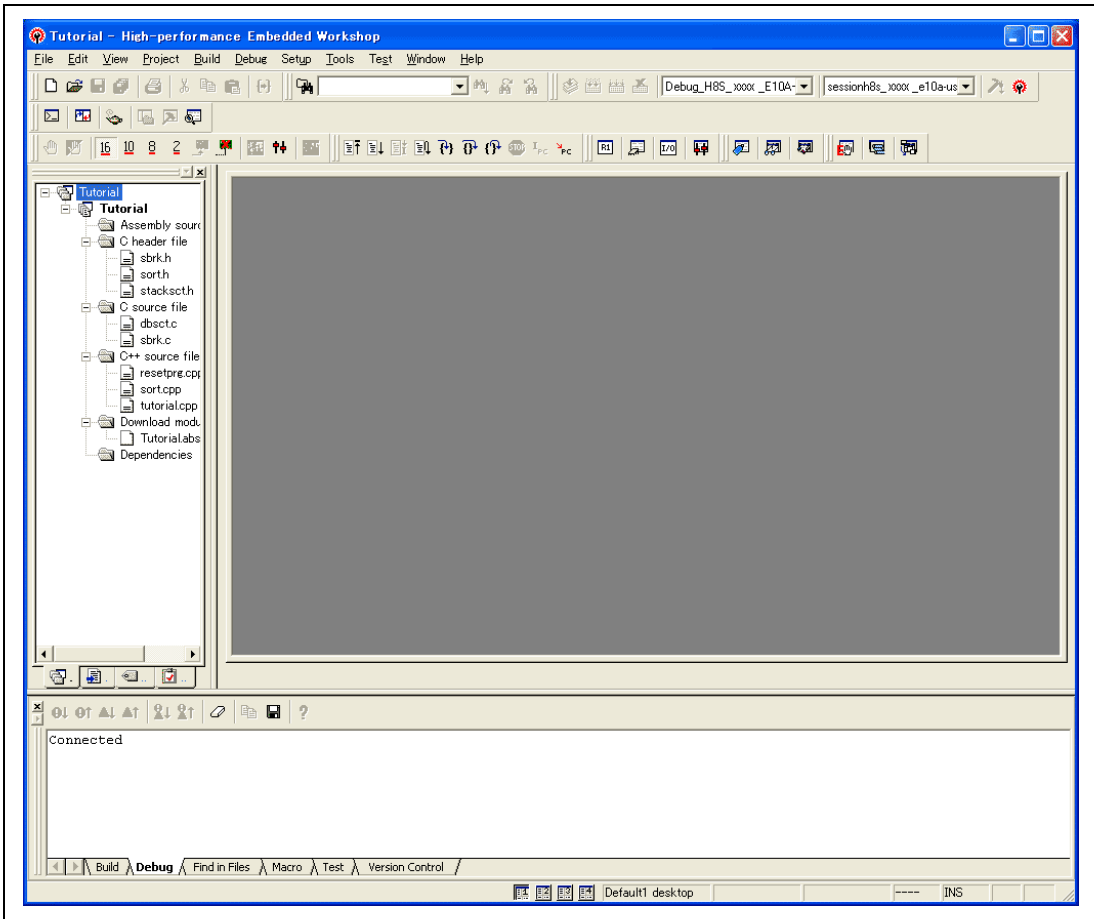
Figure 3.43   High-performance Embedded Workshop Window

Note: When the user program has already been downloaded to the flash memory, source-level debugging cannot be executed because there is no debugging information on the user program after the emulator has been activated. Be sure to load the debugging information file. For details, refer to section 4.2.1, Setting at Emulator Activation.

Notes: 1. If the user system interface cable is disconnected from the connector on the user
          system, the following dialog box will appear.
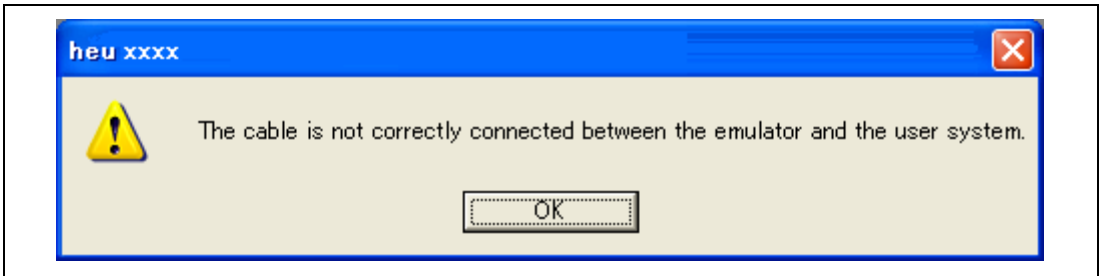
Figure 3.44   [Cable disconnected from connector] Dialog Box

2. If the emulator is not properly initialized, one of the dialog boxes shown in figures 3.45
    through 3.51 will appear.

(a) The following dialog box is displayed when the flash memory cannot be erased.
     Exchange the MCU since the flash memory has been reprogrammed more times than
     the limitation.

Figure 3.45   [Flash memory erase error!] Dialog Box

Note: If a mode is illegally set, the error message shown in above figure will be displayed.

(b) The following dialog box is displayed when an incorrect ID code has been input. For security, the flash memory is completely erased.
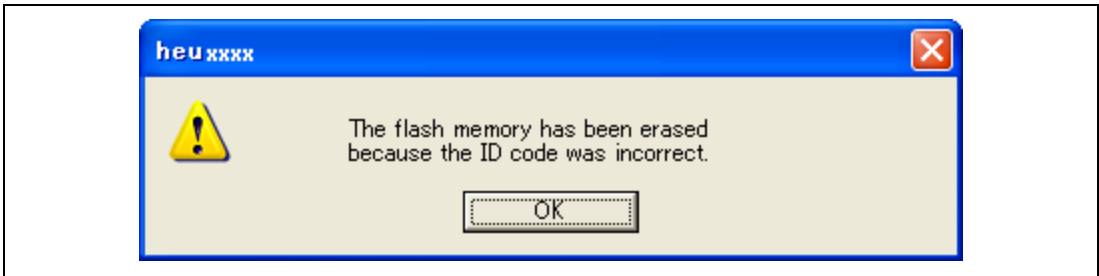
Figure 3.46   [ID code error!] Dialog Box

(c) The following dialog box is displayed when the MCU cannot communicate with the emulator. The MCU may not operate correctly; check the MCU settings.
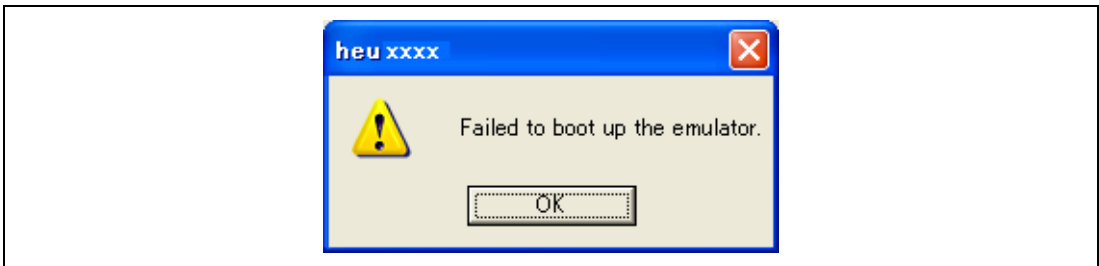
Figure 3.47   [Error H-UDI boot] Dialog Box

(d) If the following dialog box is displayed, check that the user system connector on the user system is correctly connected.
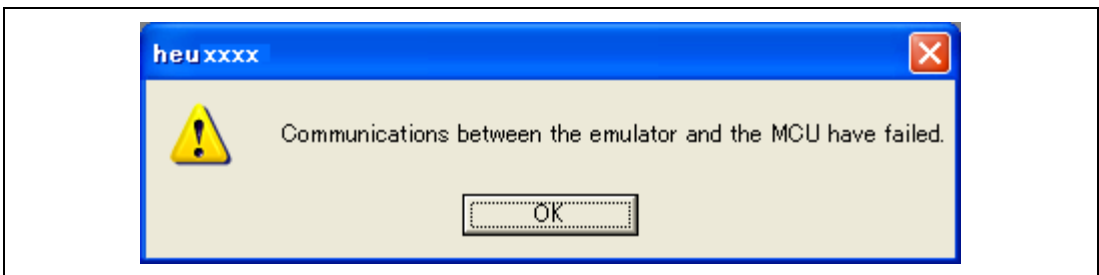
Figure 3.48   [Check the connection] Dialog Box

(e) If the following dialog box is displayed, the version of the firmware in the emulator may be old. Set up the firmware by using the setup tool. For the usage of the setup tool, refer to section 3.10, Setting up the Emulator.
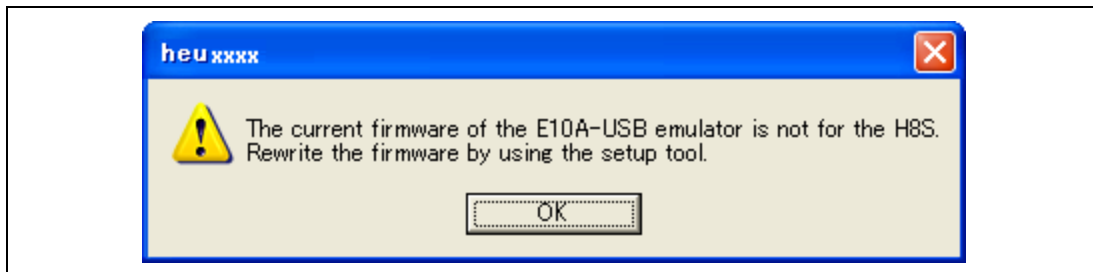
Figure 3.49   [Emulator firmware error (3)] Dialog Box

(f) If the following dialog box is displayed, the version of the firmware in the emulator may be old. Set up the firmware by using the setup tool.

Figure 3.50   [Emulator version error] Dialog Box

(g) If the following dialog box is displayed, the MCU selected on activating the emulator does not match the actual MCU. Select the correct MCU.
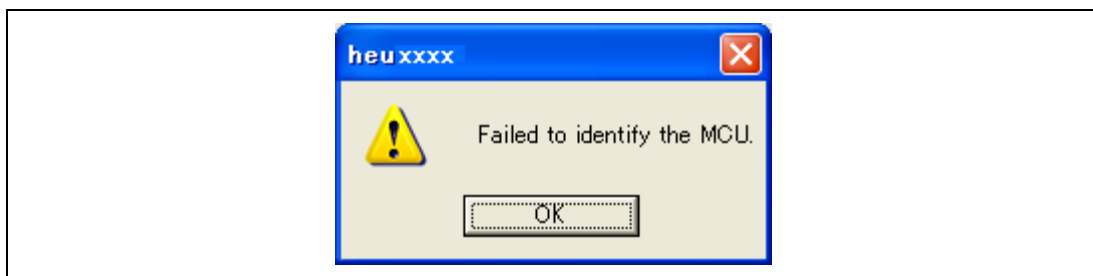
Figure 3.51   [MCU identification error] Dialog Box

3. If an incorrect driver has been selected, the following dialog box will appear.

Figure 3.52　[Driver Communications error] Dialog Box

Figure 3.53　[Driver Communications error] Dialog Box

Section 4   Preparations for Debugging

4.1   Method for Activating High-performance Embedded Workshop

This section describes the ways to create and modify workspaces using H8S family (H8S/xxxx E10A-USB SYSTEM). Replace 'H8S/xxxx E10A-USB SYSTEM' in the description (including figures) as 'H8SX/xxxx E10A-USB SYSTEM' when you use H8SX family.

To activate the High-performance Embedded Workshop, follow the procedure listed below.

1. Connect the emulator to the host computer and the user system, then turn on the user system.
2. Select [High-performance Embedded Workshop] from [Renesas High-performance Embedded Workshop] of [Programs] in the [Start] menu.
3. The [Welcome!] dialog box is displayed.
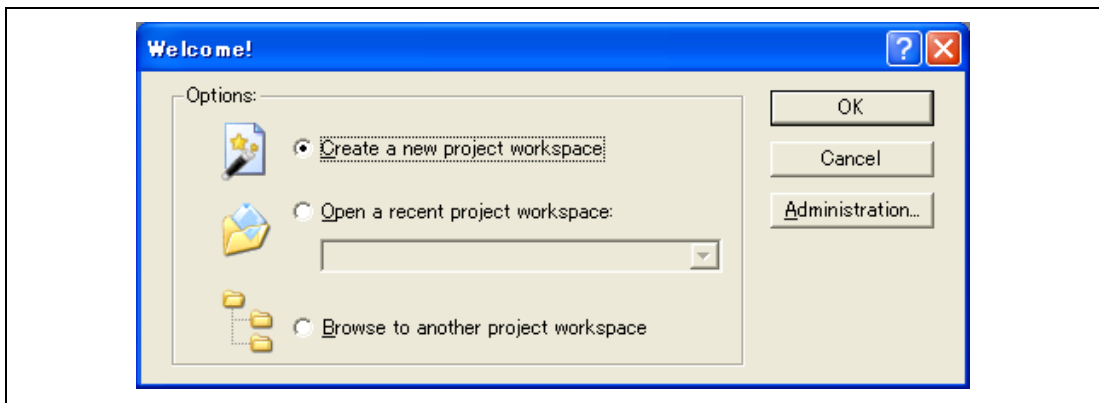
Figure 4.1   [Welcome!] Dialog Box

[Create a new project workspace] radio button: Creates a new workspace.

[Open a recent project workspace] radio button: Uses an existing workspace and displays
                                                          the history of the opened workspace.

[Browse to another project workspace] radio button: Uses an existing workspace; this radio
                                                          button is used when the history of the
                                                          opened workspace does not remain.

In this section, we describe the following three ways to start up the High-performance Embedded Workshop:

- [Create a new project workspace] - a toolchain is not in use
- [Create a new project workspace] - a toolchain is in use
- [Browse to another project workspace]

The operation of [Open a recent project workspace] radio button is same as the operation without specifying the workspace file when [Browse to another project workspace] is selected.

4.1.1   Creating the New Workspace (Toolchain Not Used)

1. In the [Welcome!] dialog box that is displayed when the High-performance Embedded Workshop is activated, select [Create a new project workspace] radio button and click the [OK] button.

Figure 4.2   [Welcome!] Dialog Box

2. The Project Generator is started. In this section, we omit description of the settings for the toolchain.
   If you have not purchased the toolchain, the following dialog box is displayed.

Figure 4.3   [New Project Workspace] Dialog Box

[Workspace Name] edit box: Enter the new workspace name. Here, for example, enter 'test'.

[Project Name] edit box: Enter the project name. When the project name is the same as the workspace name, it needs not be entered.

Other list boxes are used for setting the toolchain; the fixed information is displayed when the toolchain has not been installed.

3.　The following dialog box is displayed.
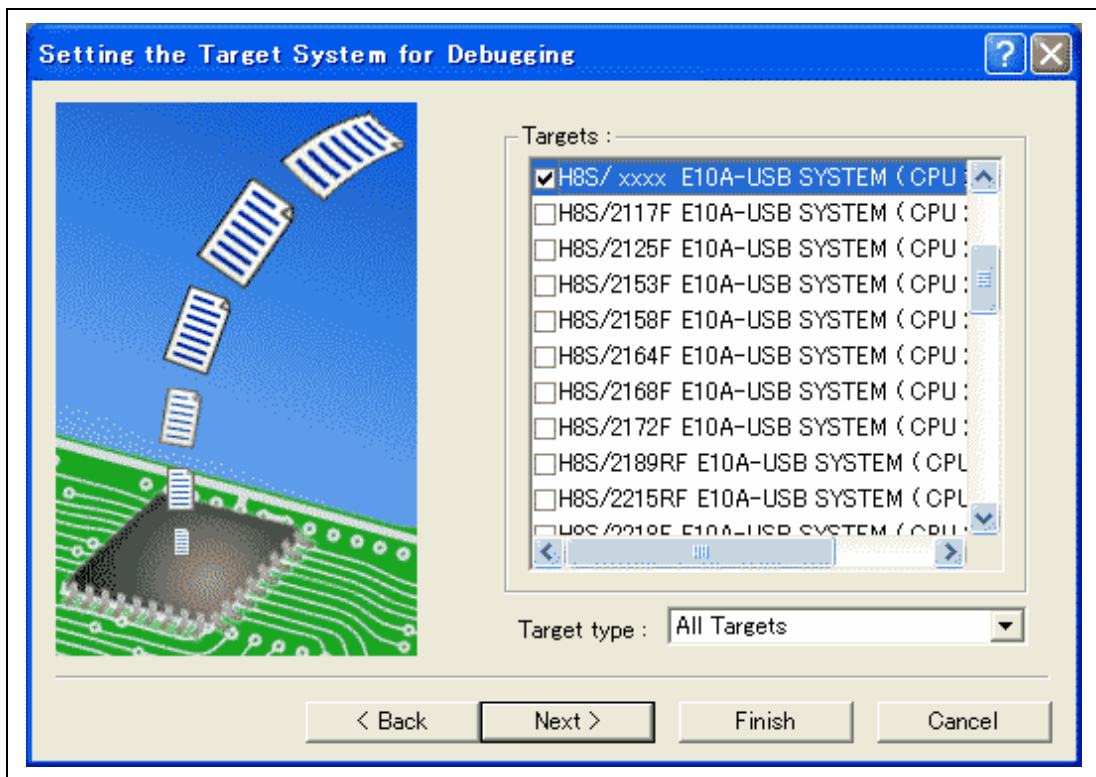
Figure 4.4　[Debugger Target] Dialog Box

Check [H8S/xxxx E10A-USB SYSTEM] and click the [Next] button.

4. Set the configuration file name. The configuration file saves the state of High-performance
   Embedded Workshop except for the emulator.
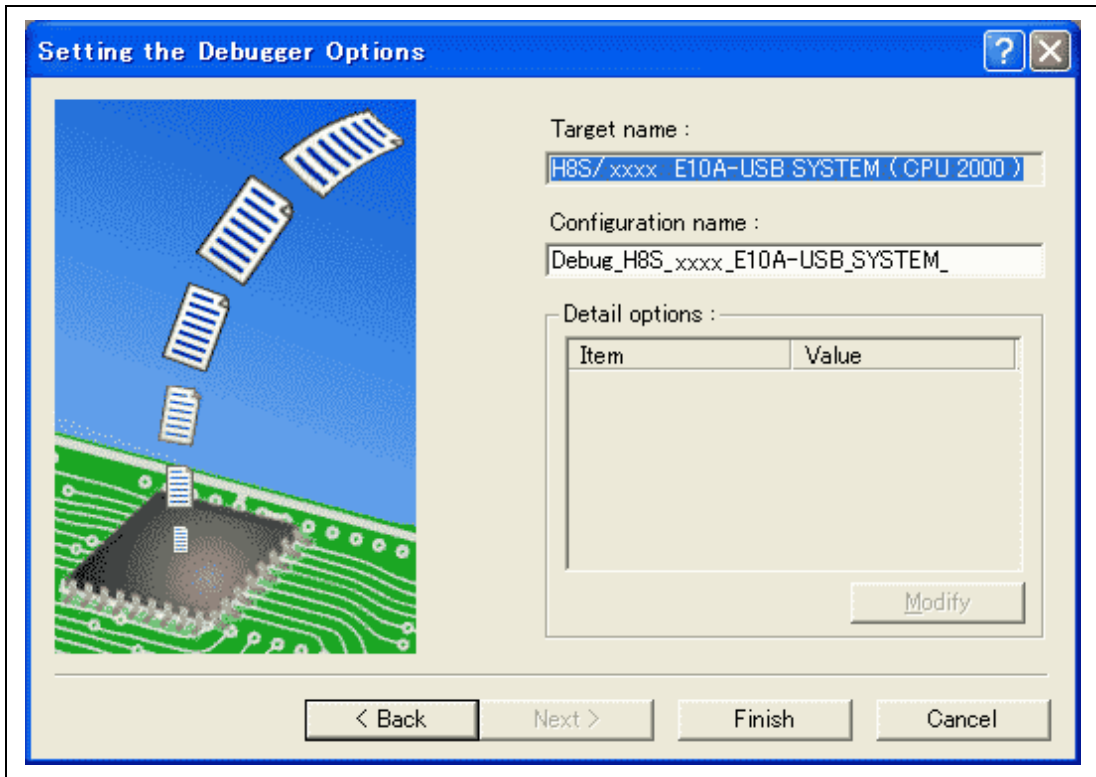
Figure 4.5   [Debugger Options] Dialog Box

This is the end of the emulator setting.

Click the [Finish] button to exit the Project Generator. The High-performance Embedded
Workshop is activated.

5. After the High-performance Embedded Workshop has been activated, the emulator is
   automatically connected. For operation during connection, refer to section 3.11, System
   Check.

4.1.2　Creating the New Workspace (Toolchain Used)

1. In the [Welcome!] dialog box that is displayed when the High-performance Embedded
   Workshop is activated, select [Create a new project workspace] radio button and click the [OK]
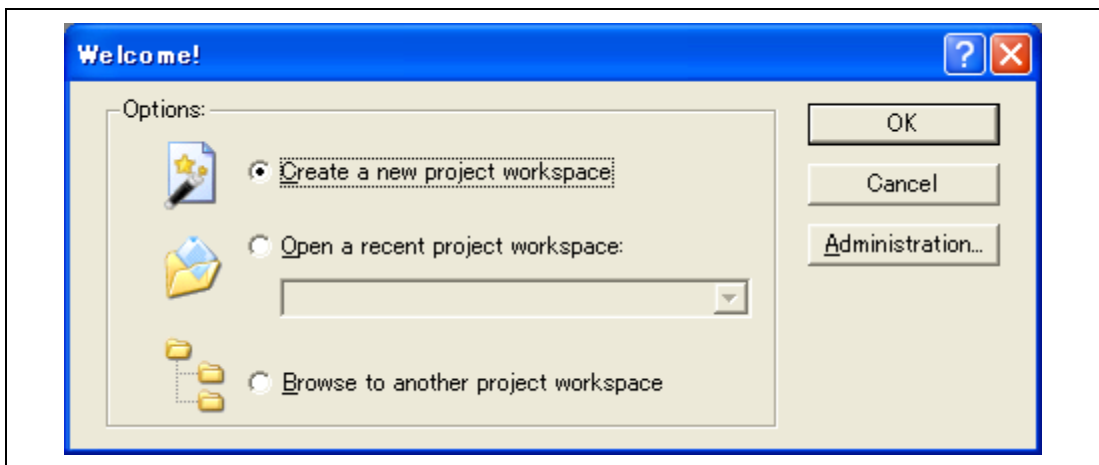   button.
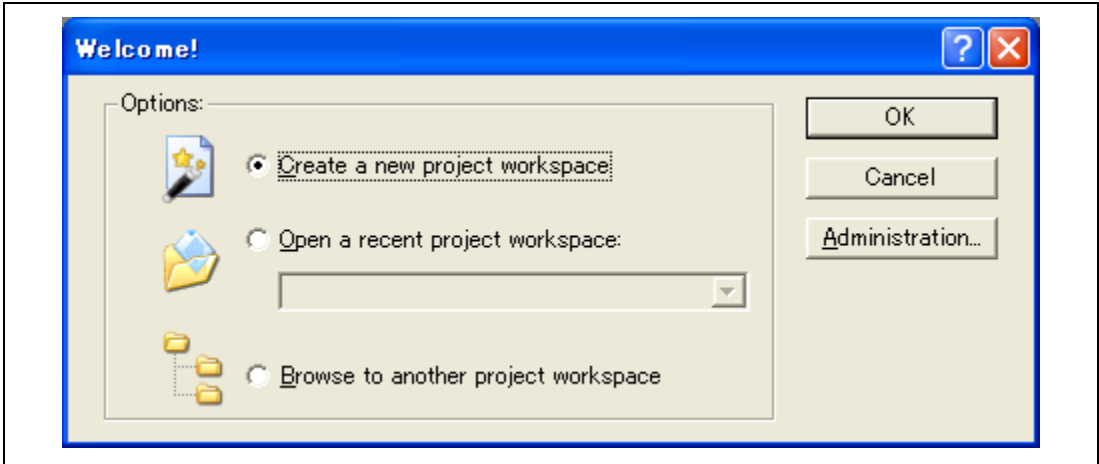
Figure 4.6　[Welcome!] Dialog Box

2. The Project Generator is started.
    If you have purchased the toolchain, the following dialog box is displayed.
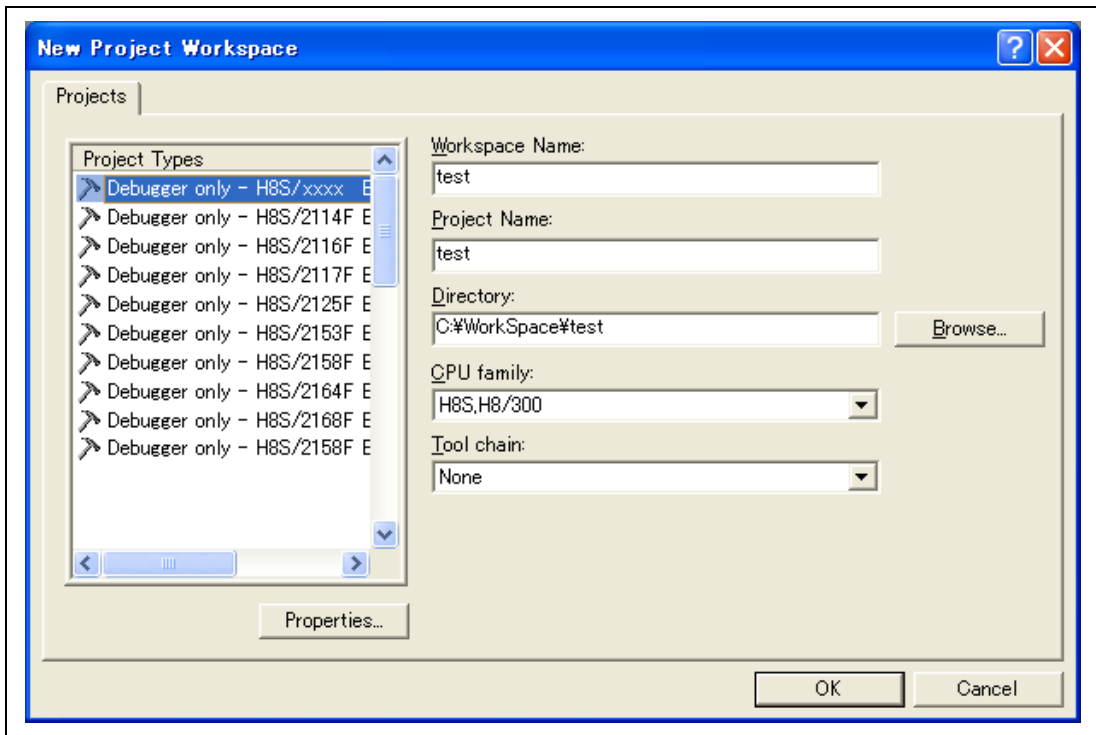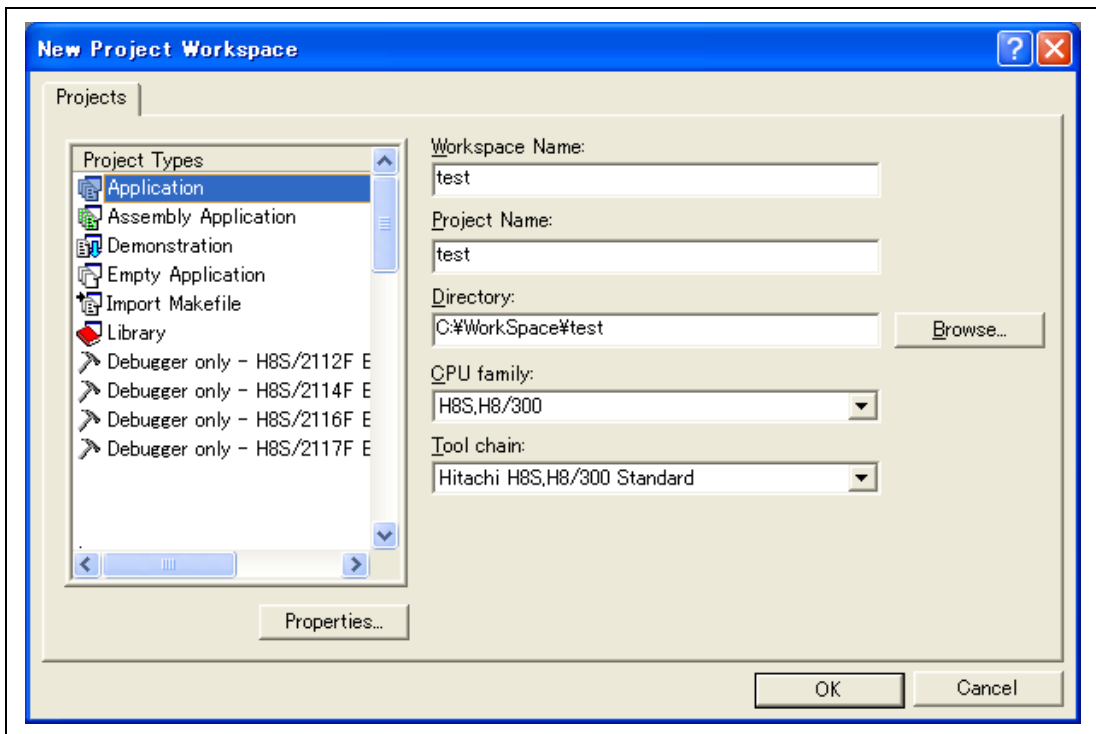

Figure 4.7   [New Project Workspace] Dialog Box

[Workspace Name] edit box:	Enter the new workspace name. Here, for example, enter 'test'.
[Project Name] edit box:	Enter the project name. When the project name is the same as the workspace name, it needs not be entered.
[CPU family] drop-down list box:	Select the target CPU family.
[Tool chain] drop-down list box:	Select the target toolchain name when using the toolchain. Otherwise, select [None].
[Project type] list box:	Select the project type to be used.

Note: When [Demonstration] is selected in the emulator, note the following:
        The [Demonstration] is a program for the simulator. When the generated program is used
        by the emulator, delete the Printf statement.

3. Make the required setting for the toolchain. When the setting has been completed, the
    following dialog box is displayed.
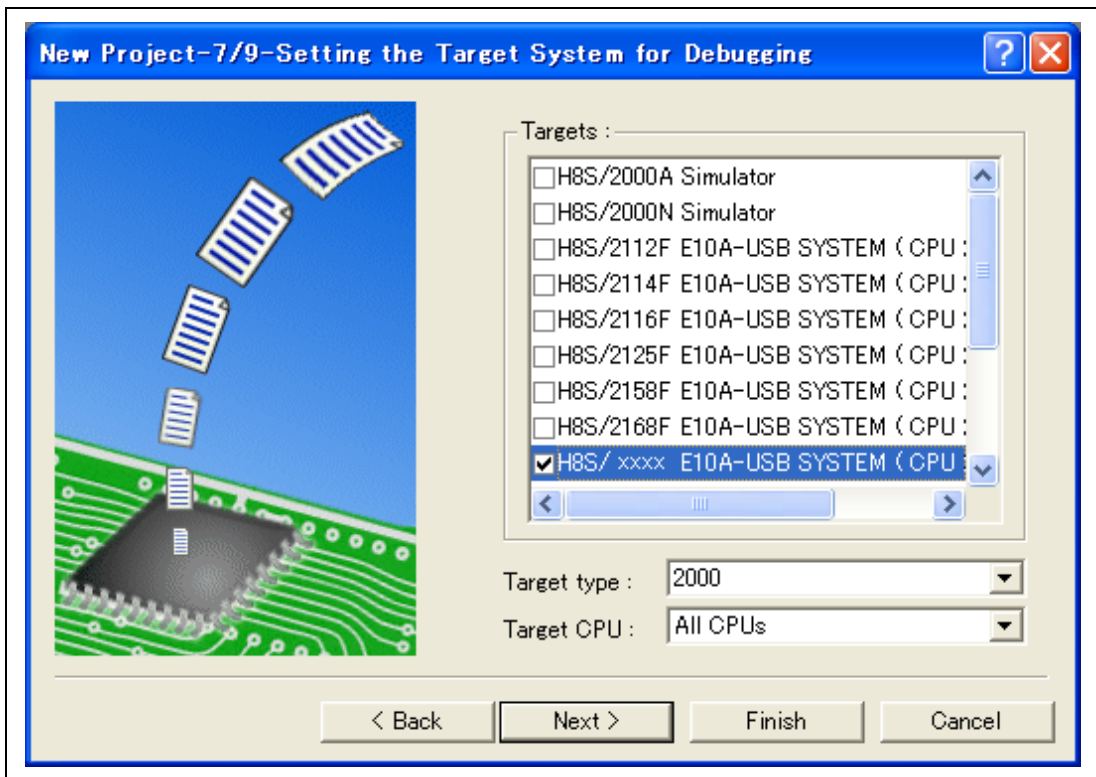
Figure 4.8   [New Project -7/9- Setting the Target System for Debugging] Dialog Box

Check [H8S/xxxx E10A-USB SYSTEM] and click the [Next] button. Mark other products as
required.

4. Set the configuration file name. The configuration file saves the state of High-performance Embedded Workshop except for the emulator.
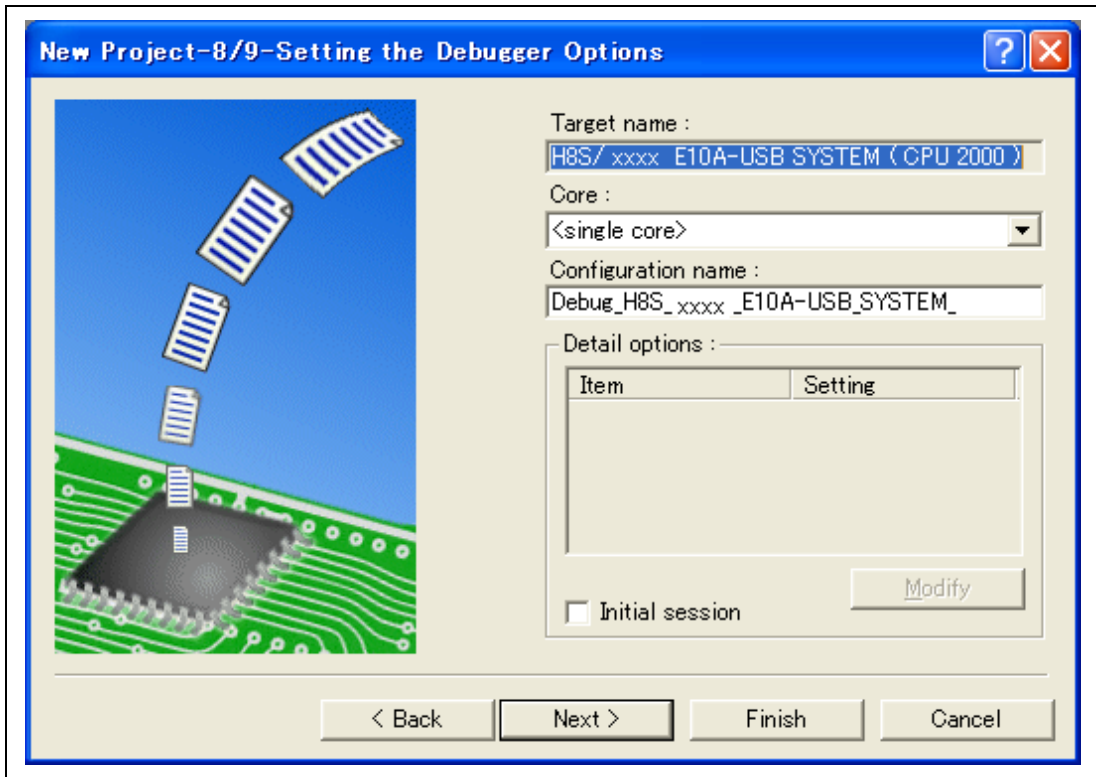
**Figure 4.9   [New Project -8/9- Setting the Debugger Options] Dialog Box**

This is the end of the emulator setting.

Exit the Project Generator according to the instructions on the screen. The High-performance Embedded Workshop is activated.

5. After the High-performance Embedded Workshop has been activated, connect the emulator. However, it is not needed to connect the emulator immediately after the High-performance Embedded Workshop has been activated.

   To connect the emulator, use one of the methods (a) and (b) below. For operation during connection, refer to section 3.11, System Check.

   (a) Connecting the emulator after the setting at emulator activation

   Select [Debug settings] from the [Debug] menu to open the [Debug Settings] dialog box. It is possible to register the download module or the command chain that is automatically executed at activation. For details on the [Debug Settings] dialog box, refer to section 4.2, Setting at Emulator Activation.

   After the [Debug Settings] dialog box has been set, when the dialog box is closed, the emulator is connected.

   (b) Connecting the emulator without the setting at emulator activation

   The emulator can be easily connected by switching the session file that the setting for the emulator use has been registered.
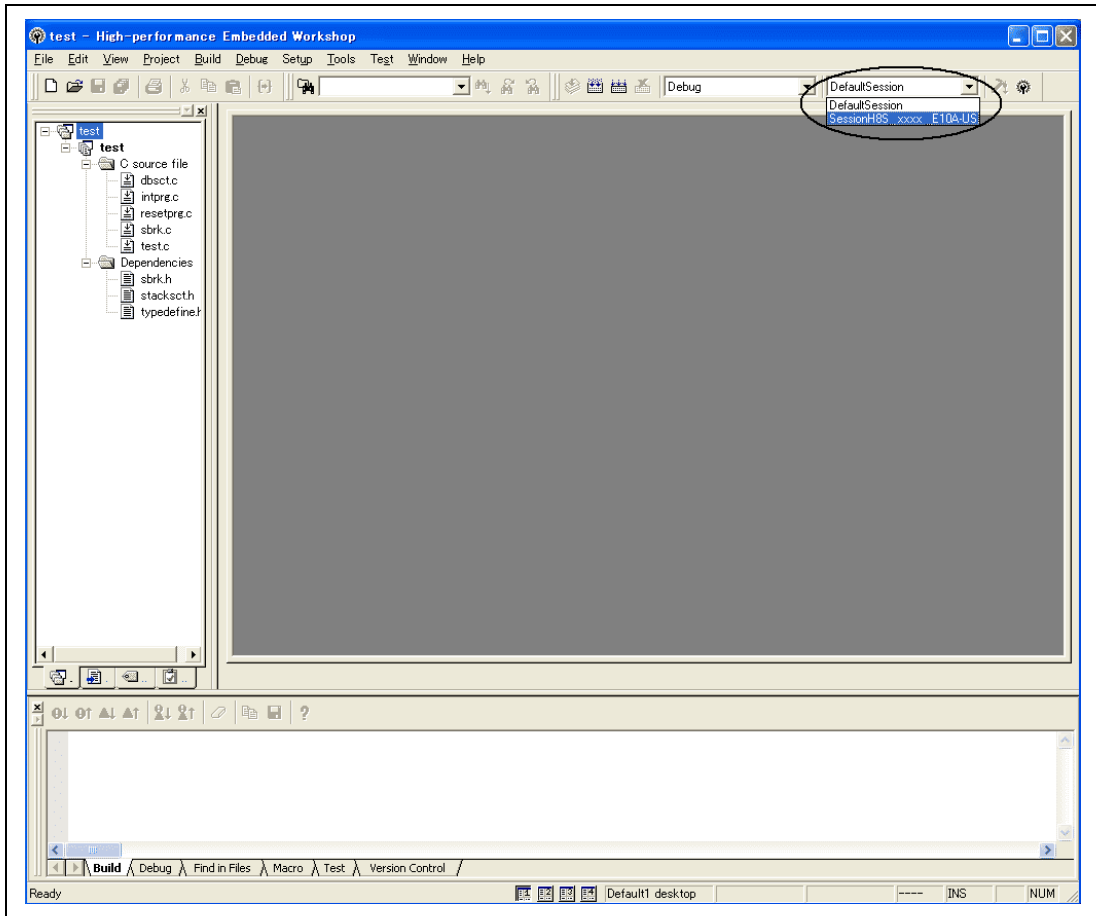
Figure 4.10   Selecting the Session File

In the list box that is circled in figure 4.10, select the session file name including the character string that has been set in the [Target name] text box in figure 4.9, [New Project -8/9- Setting the Debugger Options] dialog box. The setting for using the emulator has been registered in this session file.

After selected, the emulator is automatically connected.

4.1.3   Selecting an Existing Workspace

1. In the [Welcome!] dialog box that is displayed when the High-performance Embedded
    Workshop is activated, select [Browse to another project workspace] radio button and click the
    [OK] button.
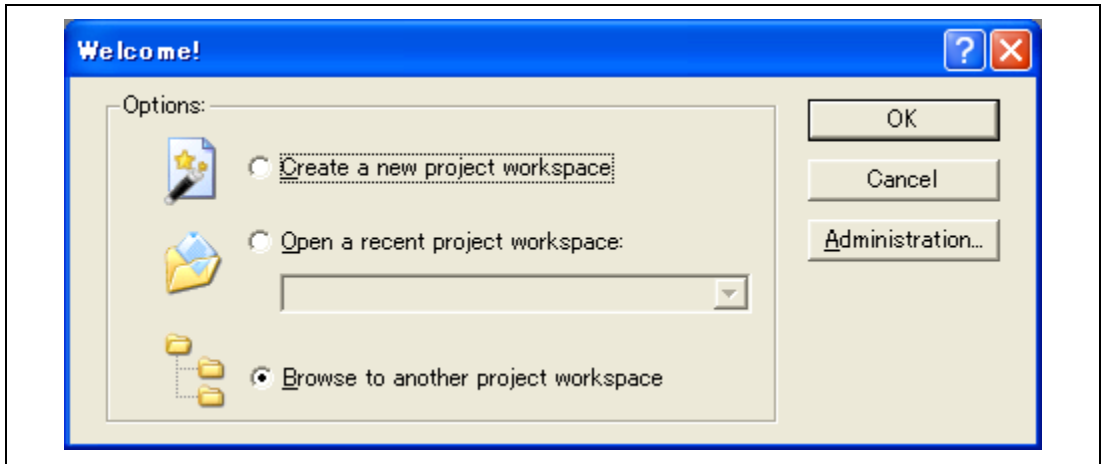
Figure 4.11   [Welcome!] Dialog Box

2. The [Open Workspace] dialog box is displayed. Select a directory in which you have created a workspace.
   After that, select the workspace file (.hws) and press the [Open] button.
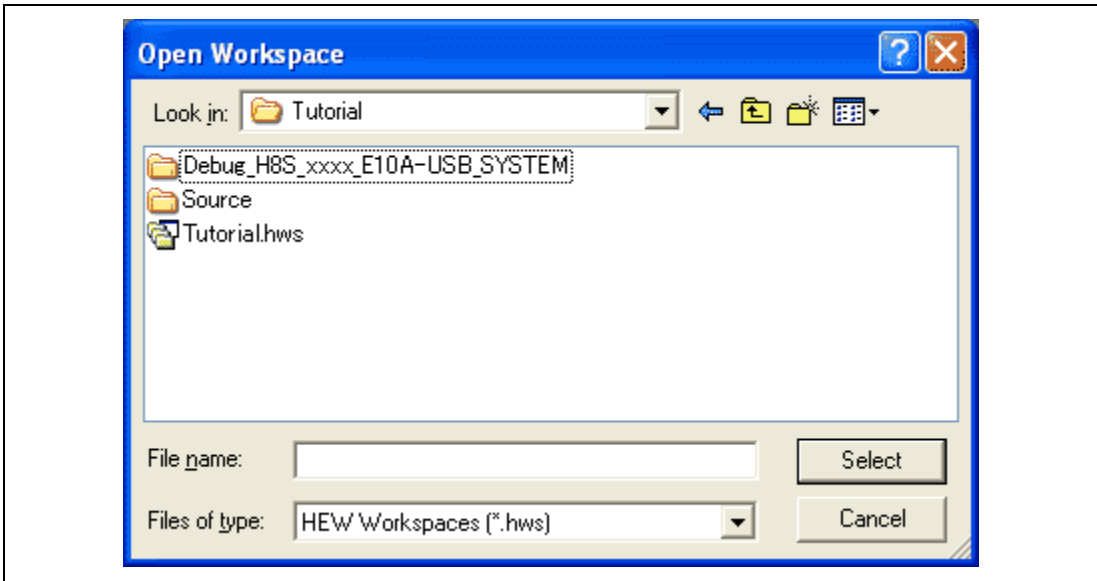
Figure 4.12　[Open Workspace] Dialog Box

Note:　In figure 4.12, 'xxxx' means the name of the MCU.

3. This activates the High-performance Embedded Workshop and recovers the state of the selected workspace at the time it was saved.
   When the saved state information of the selected workspace includes connection to the emulator, the emulator will automatically be connected. To connect the emulator when the saved state information does not include connection to the emulator, refer to section 4.4, Connecting the Emulator.

4.2   Setting at Emulator Activation

4.2.1   Setting at Emulator Activation

A batch file of commands can be specified for automatic execution when the emulator is activated. It is also possible to register multiple load modules to be downloaded. The registered load modules are displayed on the workspace window.

1. Select [Debug Settings…] from the [Debug] menu to open the [Debug Settings] dialog box.
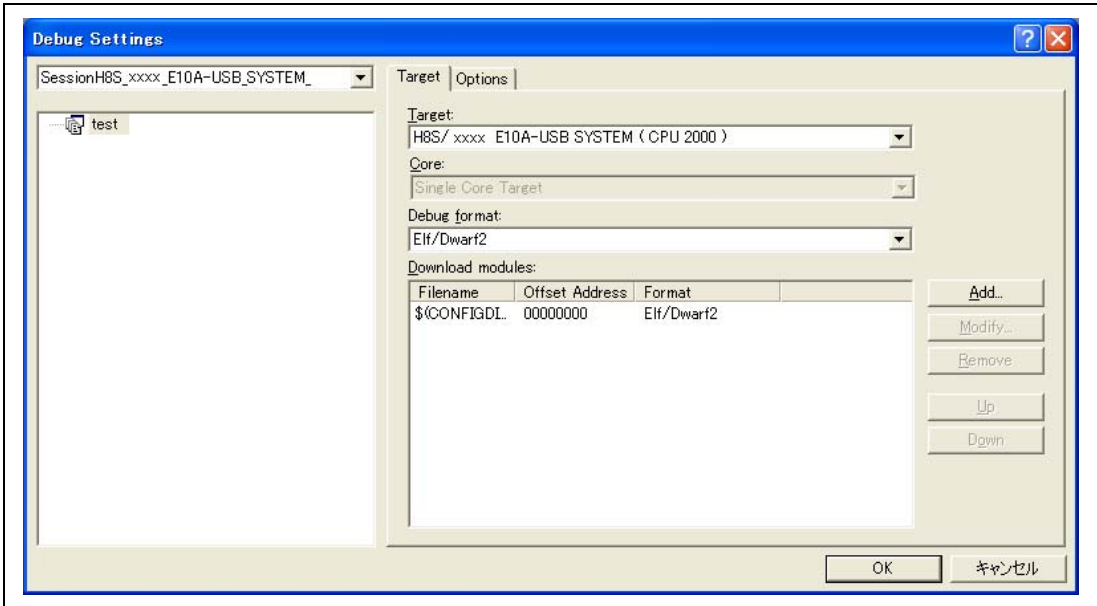
Figure 4.13   [Debug Settings] Dialog Box ([Target] Page)

2. Select the product name to be connected in the [Target] drop-down list box.
3. Select the format of the load module to be downloaded in the [Default Debug Format] drop-down list box, then register the corresponding download module in the [Download Modules] list box.

Note: Here, no program has been downloaded. For downloading, refer to section 5.2,
         Downloading a Program.

4. Click the [Options] tab.
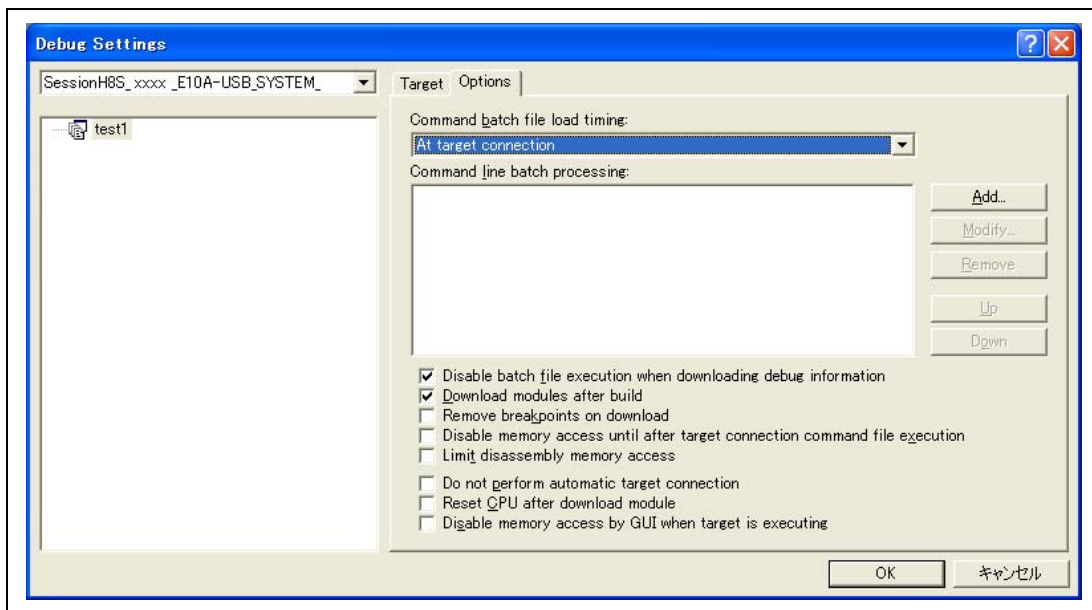
Figure 4.14   [Debug Settings] Dialog Box ([Options] Page)

The command batch file that is automatically executed at the specified timing is registered. The following four timings can be specified:

- At connecting the emulator (At target connection)
- Immediately before downloading (Before download of modules)
- Immediately after downloading (After download of modules)
- Immediately after reset (After reset)

Specify the timing for executing the command batch file in the [Command batch file load timing] drop-down list box. In addition, register the command batch file to be executed with the specified timing in the [Command Line Batch Processing] list box.

4.2.2   Downloading a Program

A download module is added under [Download modules] in the [Workspace] window.

Open the load module of [Download modules] in the [Workspace] window by clicking the right-hand mouse button and select [Download module] to start downloading the module.
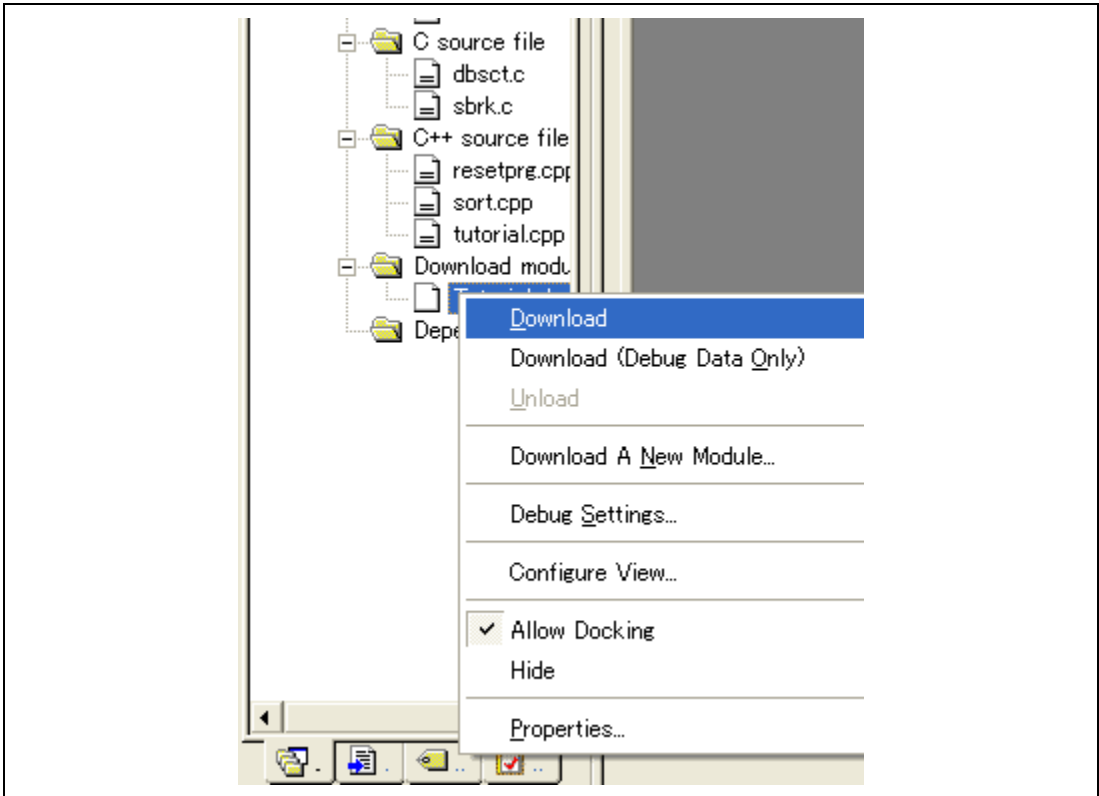
Figure 4.15   Download Menu of the [Workspace] Window ([Projects])

Notes: 1. When load modules are downloaded, select [Debug] -> [Download] -> [All DownLoad Modules].
   2. The emulator downloads programs to the MCU flash memory just before execution of the user program.

4.2.3   Setting the Program Flash Mode

The following describes the procedures when the emulator is used as the programming tool. The load module to be downloaded to the new workspace is registered and programmed.

(a) Select the new project workspace.
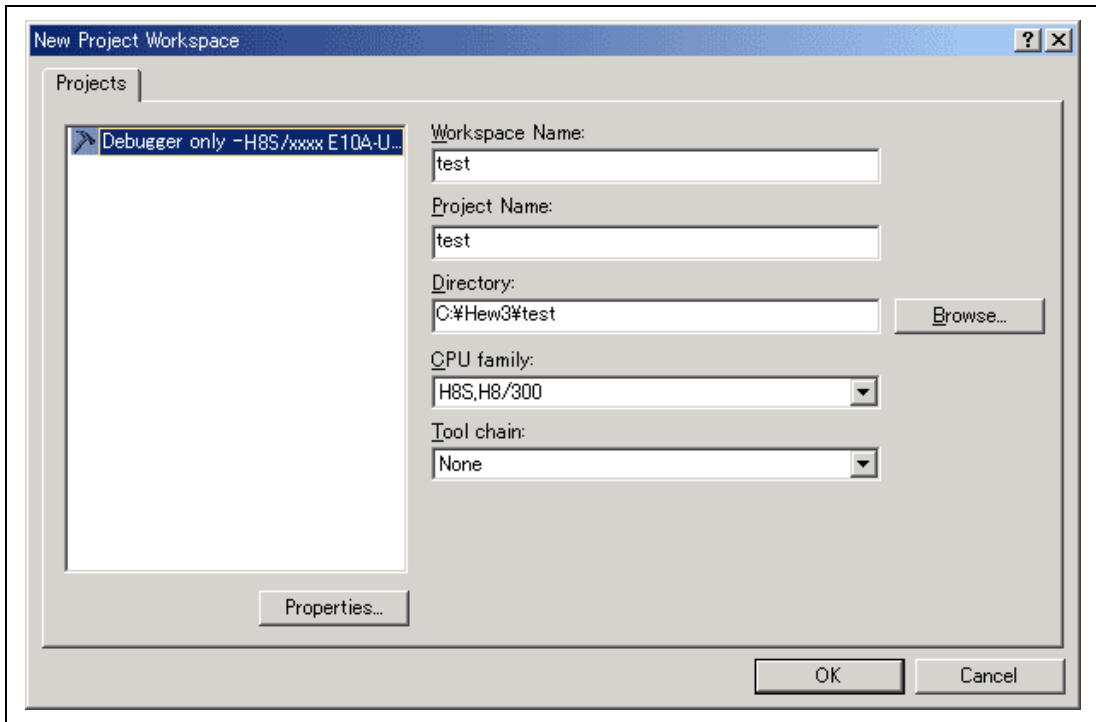
Figure 4.16   [New Project Workspace] Dialog Box

(b) Select the target MCU and click the [Finish] button.
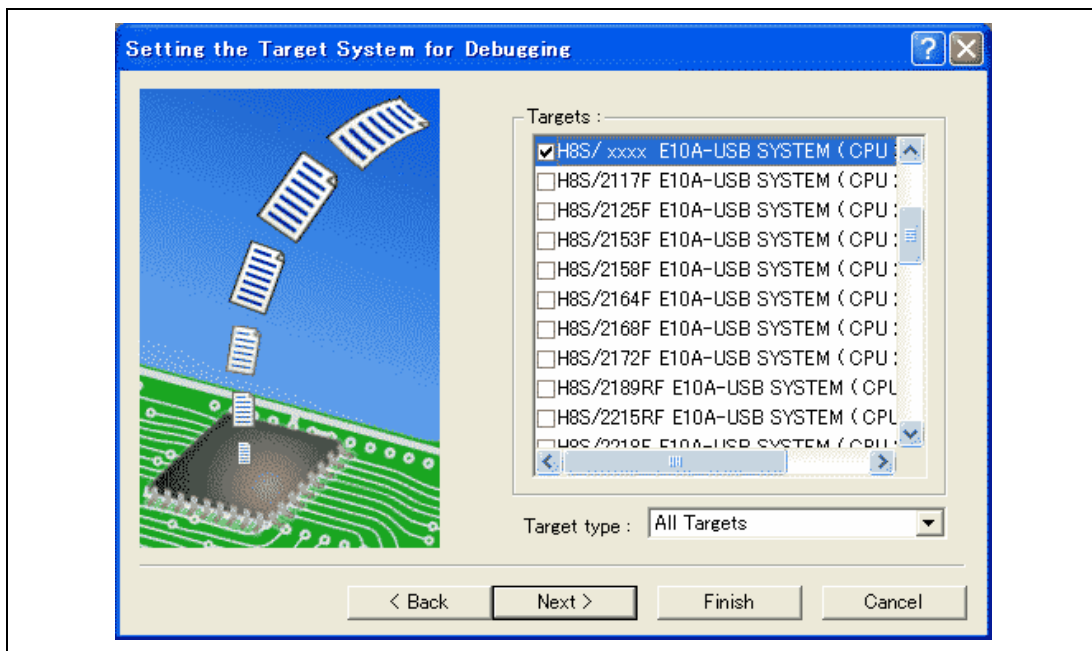
Figure 4.17   [Setting the Target System for Debugging] Dialog Box

(c) The [Select Emulator mode] dialog box is displayed.
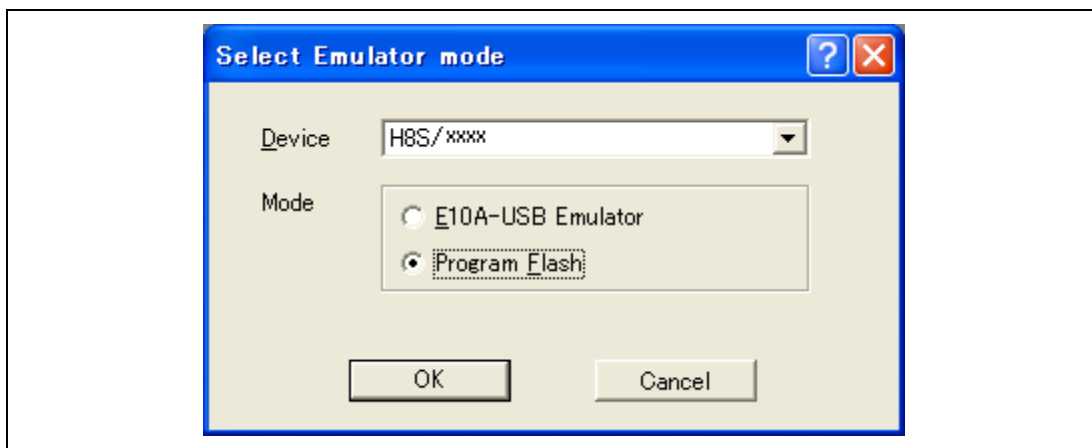
Figure 4.18   [Select Emulator mode] Dialog Box

Select the [Program Flash] mode.

Note: The contents to be displayed differ from each MCU. This is dedicated mode for each
         MCU, and is incorporated in the internal ROM. This is not available for ROM-less
         products. Normal debugging is not possible.

(d) If the dialog box shown below is displayed, turn on the target board and press the [OK] button.
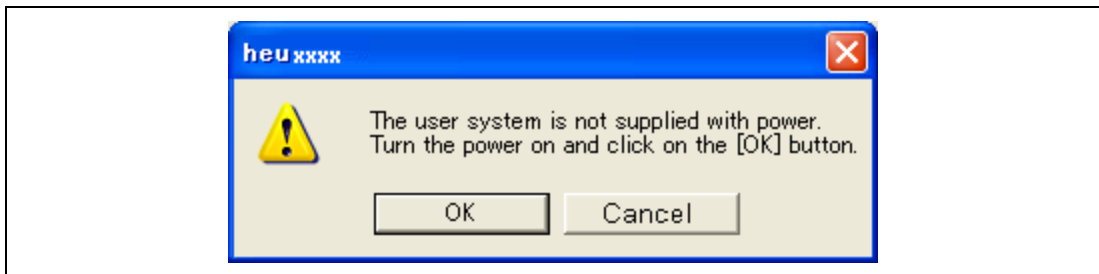
Figure 4.19   [Turn Power on] Message Box
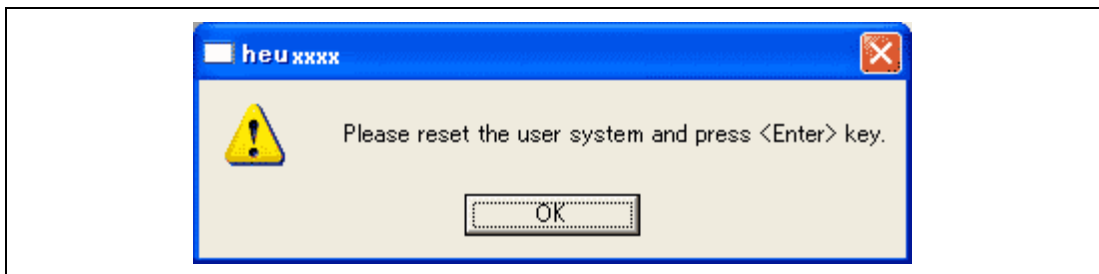
Figure 4.20   Dialog Box of the Reset Request Message

(e) Input the system clock value.
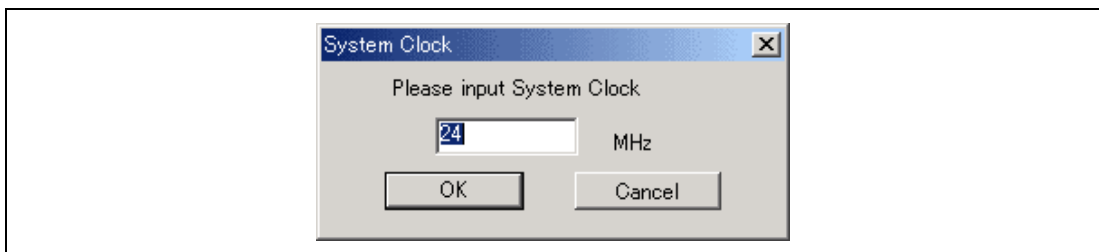
Figure 4.21   [System Clock] Dialog Box

(f) Select [Debug Setting] from the [Debug] menu.
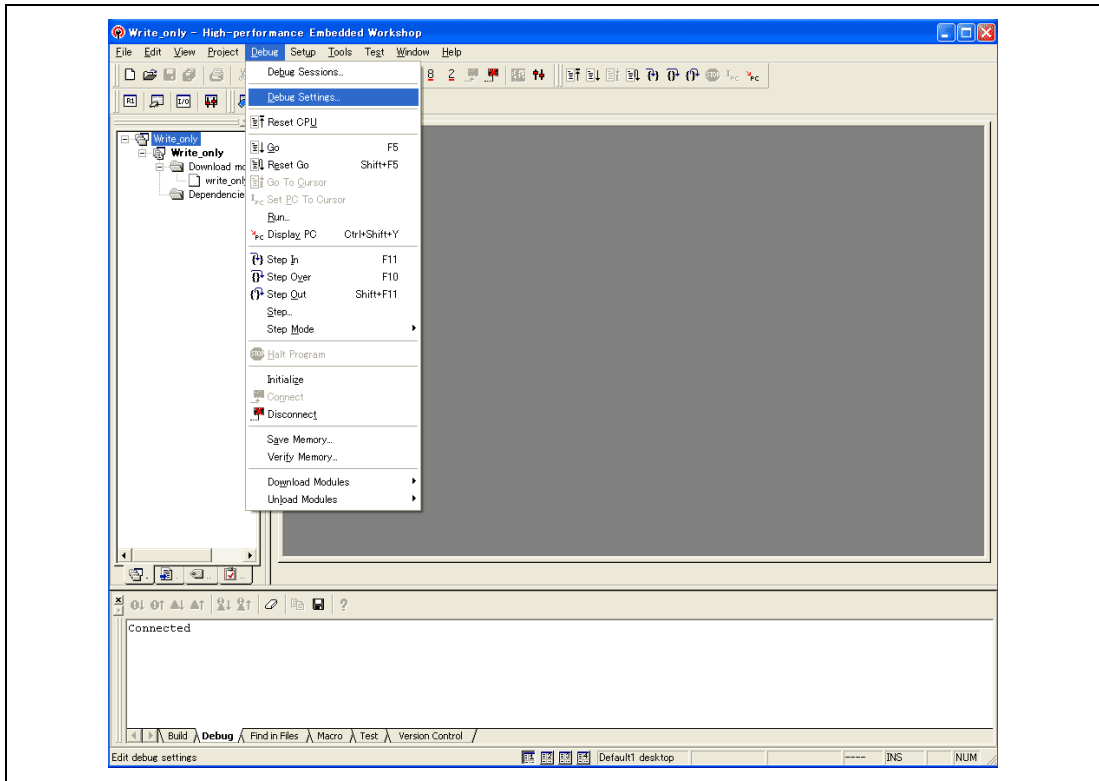
Figure 4.22   High-performance Embedded Workshop Window

(g) Select the download module with the [Add…] button.
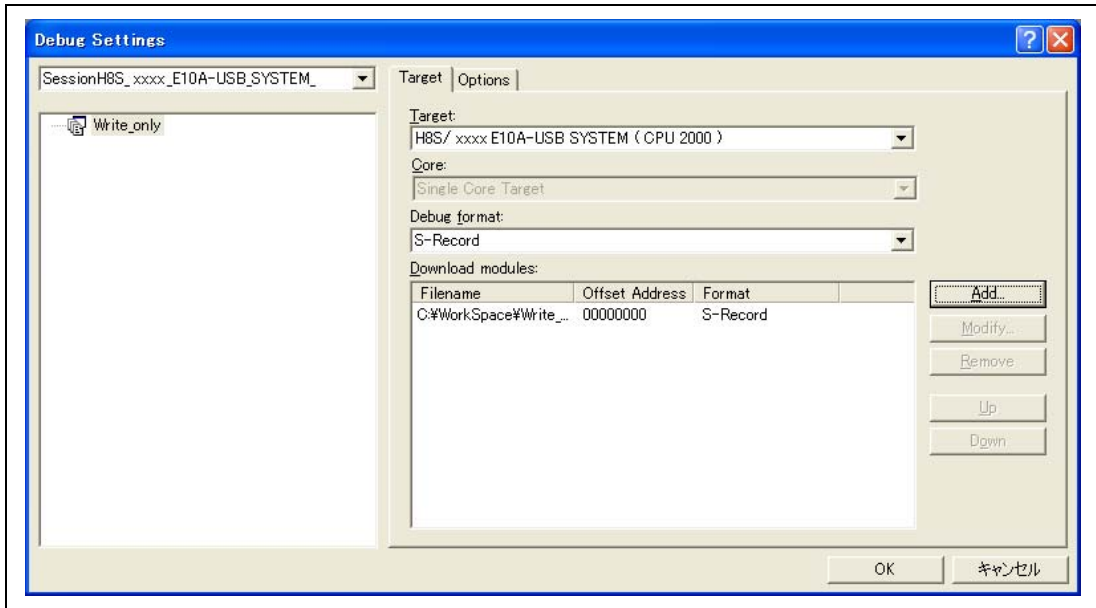
Figure 4.23   [Debug Settings] Dialog Box ([Target] Page)

(h) The download file is displayed on [Projects].
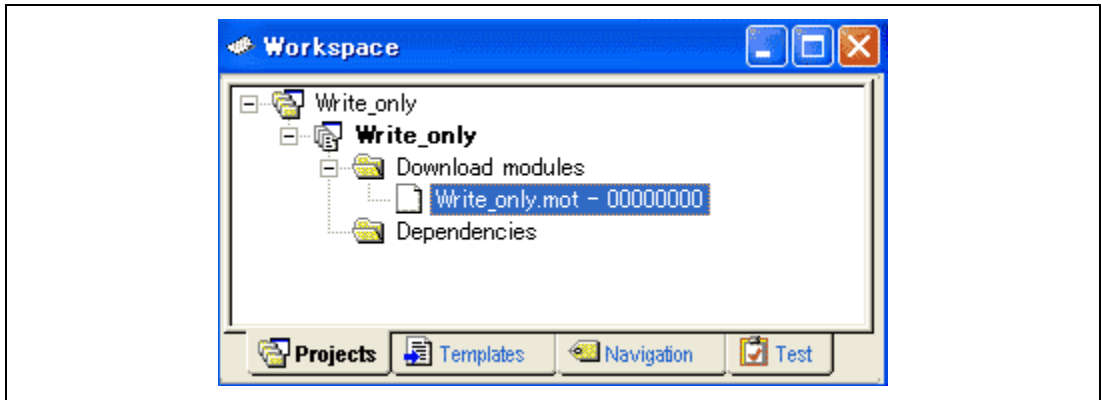
Figure 4.24   [Workspace] Window ([Projects])

(i) Select and download the file with the right-hand mouse button.
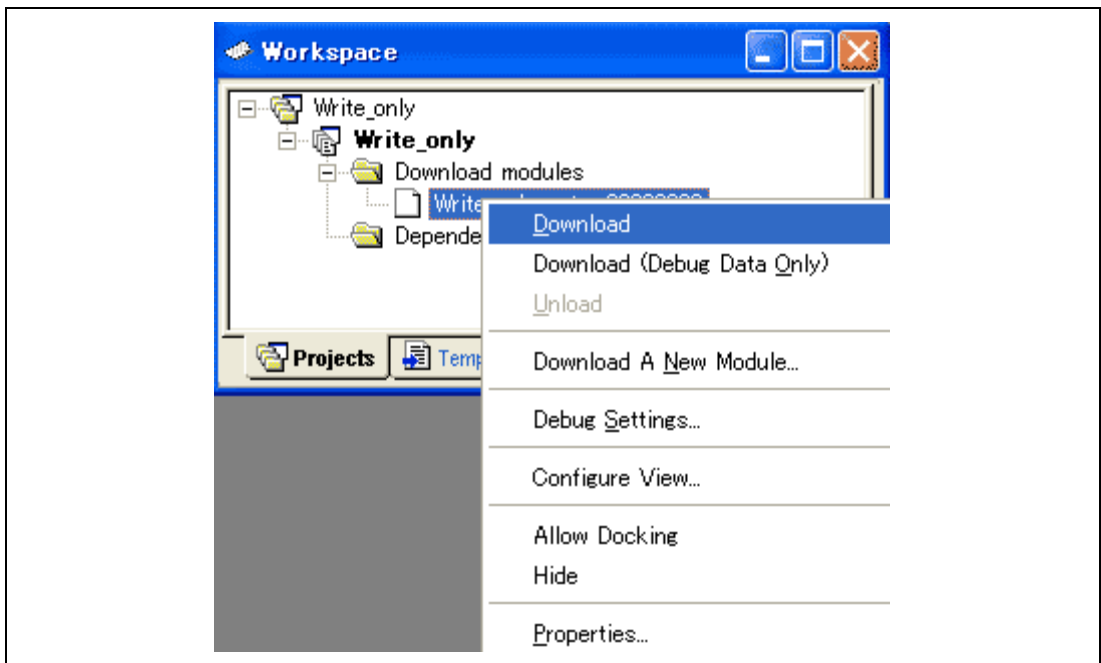
Figure 4.25   Download Menu of the [Workspace] Window ([Projects])

(j) The dialog box for sum checking is displayed and programming is completed. Checksum is a value that data in the internal ROM area has been added by byte. If no user program exists, the value is calculated by H'FF.
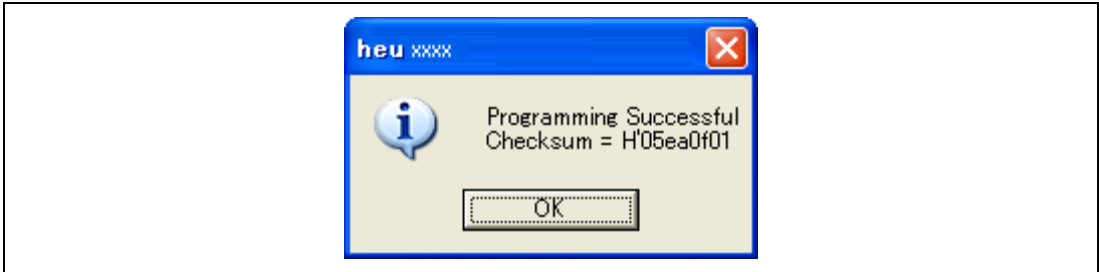
Figure 4.26　Message for Completion of Flash Memory Programming

(k) When the following dialog box is displayed, close and restart or exit the workspace.
To write the same data to the MCU continuously, select [Debug -> Initialize…] and go back to step (c) (displaying [Select Emulator mode]). To exit, close the workspace.

Figure 4.27　Message for Restarting or Exiting Program Flash Mode

4.3　　Debug Sessions

The High-performance Embedded Workshop stores all of your builder options into a configuration.　In a similar way, the High-performance Embedded Workshop stores your debugger options in a session.　The debugging platforms, the programs to be downloaded, and each debugging platform's options can be stored in a session.

Sessions are not directly related to a configuration.　This means that multiple sessions can share the same download module and avoid unnecessary program rebuilds.

Each session's data should be stored in a separate file in the High-performance Embedded Workshop project.　Debug sessions are described in detail below.

4.3.1　　Selecting a Session

The current session can be selected in the following two ways:

- From the toolbar
  Select a session from the drop-down list box (figure 4.28) in the toolbar.
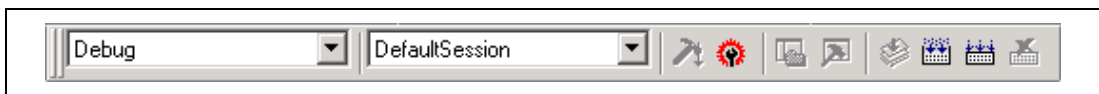
Figure 4.28　Toolbar Selection

- From the dialog box
    1. Select [Debug -> Debug Sessions…]. This will open the [Debug Sessions] dialog box (figure 4.29).
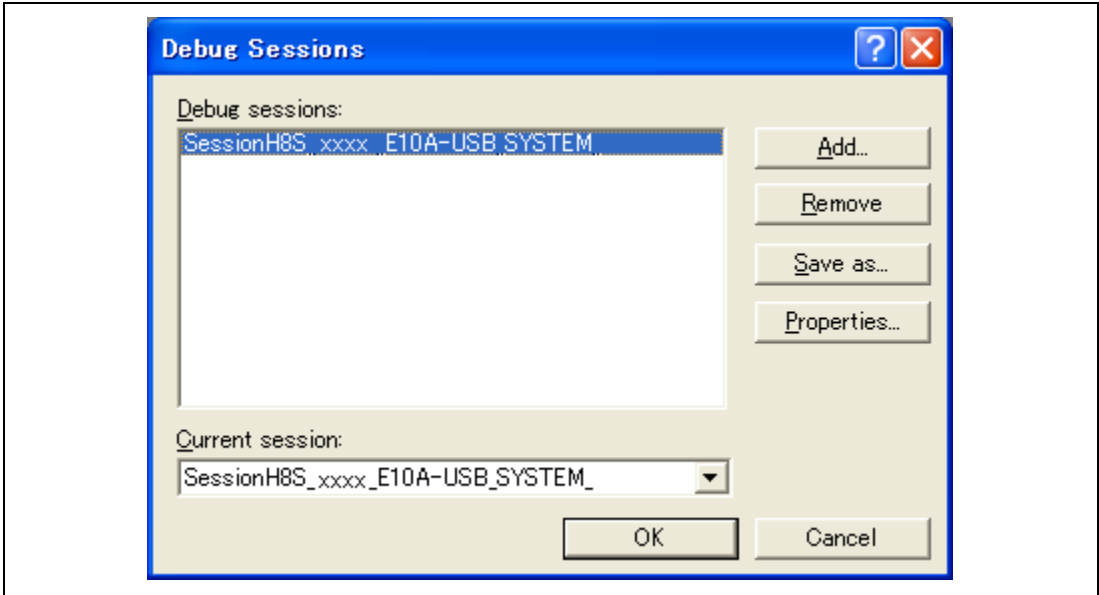
Figure 4.29   [Debug Sessions] Dialog Box

    2. Select the session you want to use from the [Current session] drop-down list.
    3. Click the [OK] button to set the session.

4.3.2   Adding and Removing Sessions

A new session can be added by copying settings from another session or removing a session.

- To add a new empty session
    1. Select [Debug -> Debug Sessions…] to display the [Debug Sessions] dialog box (figure 4.29).
    2. Click the [Add…] button to display the [Add new session] dialog box (figure 4.30).
    3. Check the [Add new session] radio button.
    4. Enter a name for the session.
    5. Click the [OK] button to close the [Debug Sessions] dialog box.
    6. This creates a file with the name entered in step 4. If a file with this name already exists, an error is displayed.
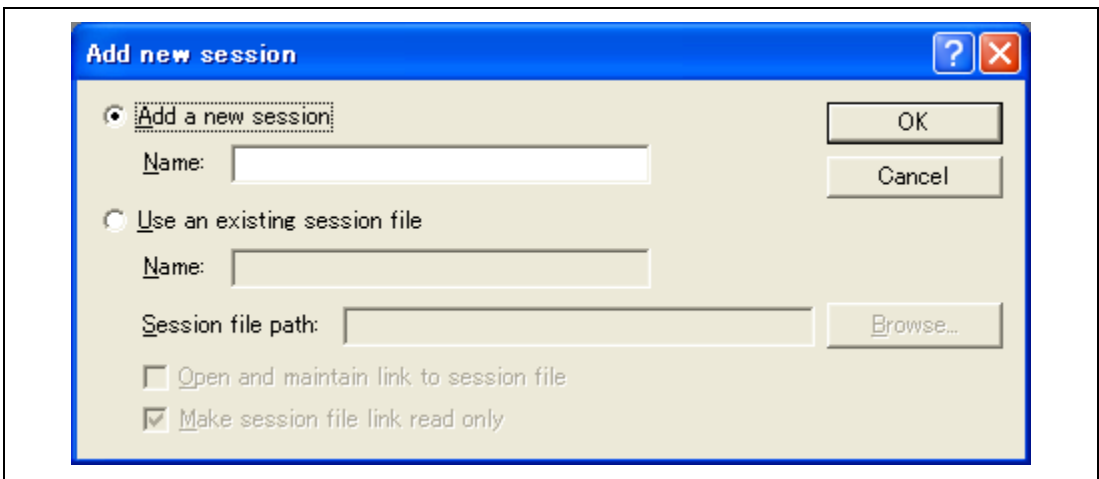
Figure 4.30   [Add new session] Dialog Box

- To import an existing session into a new session file
    1. Select [Debug -> Debug Sessions…] to display the [Debug Sessions] dialog box (figure 4.29).
    2. Click the [Add…] button to display the [Add new session] dialog box (figure 4.30).
    3. Check the [Use an existing session file] radio button.
    4. Enter a name for the session.
    5. Enter the name of an existing session file that you would like to import into the existing project or click the [Browse] button to select the file location.

       If the [Open and maintain link to session file] check box is not checked, the imported new session file is generated in the project directory.

       If the [Open and maintain link to session file] check box is checked, a new session file is not generated in the project directory but is linked to the existing session file.

       If the [Make session file link read only] check box is checked, the linked session file is used as read-only.
    6. Click the [OK] button to close the [Debug Sessions] dialog box.

- To remove a session
    1. Select [Debug -> Debug Sessions…] to display the [Debug Sessions] dialog box (figure 4.29).
    2. Select the session you would like to remove.
    3. Click the [Remove] button.

       Note that the current session cannot be removed.
    4. Click the [OK] button to close the [Debug Sessions] dialog box.

- To view the session properties
  1. Select [Debug -> Debug Sessions…] to display the [Debug Sessions] dialog box (figure 4.29).
  2. Select the session you would like to view the properties for.
  3. Click the [Properties] button to display the [Session Properties] dialog box (figure 4.31).
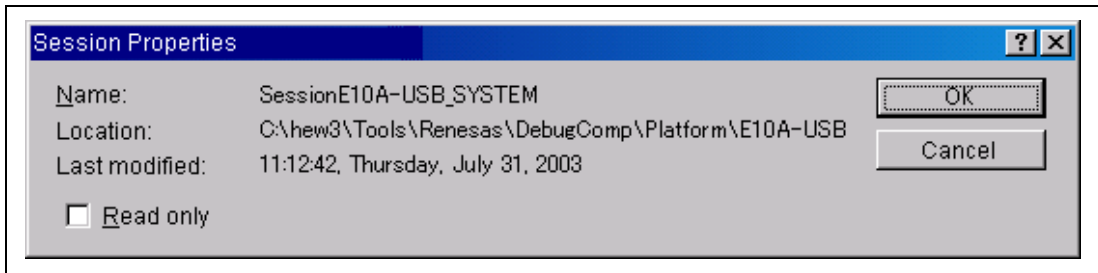
Figure 4.31   [Session Properties] Dialog Box

- To make a session read-only
  1. Select [Debug -> Debug Sessions…] to display the [Debug Sessions] dialog box (figure 4.29).
  2. Select the session you would like to make read-only.
  3. Click the [Properties] button to display the [Session Properties] dialog box (figure 4.31).
  4. Check the [Read only] check box to make the link read-only. This is useful if you are sharing debugger-setting files and you do not want data to be modified accidentally.
  5. Click the [OK] button.

- To save a session with a different name
    1. Select [Debug -> Debug Sessions…] to display the [Debug Sessions] dialog box (figure 4.29).
    2. Select the session you would like to save.
    3. Click the [Save as…] button to display the [Save Session] dialog box (figure 4.32).
    4. Specify the location to save the new file.
    5. If you want to export the session file to another location, leave the [Maintain link] check box unchecked. If you would like the High-performance Embedded Workshop to use this location instead of the current session location, check the [Maintain link] check box.
    6. Click the [Save] button.
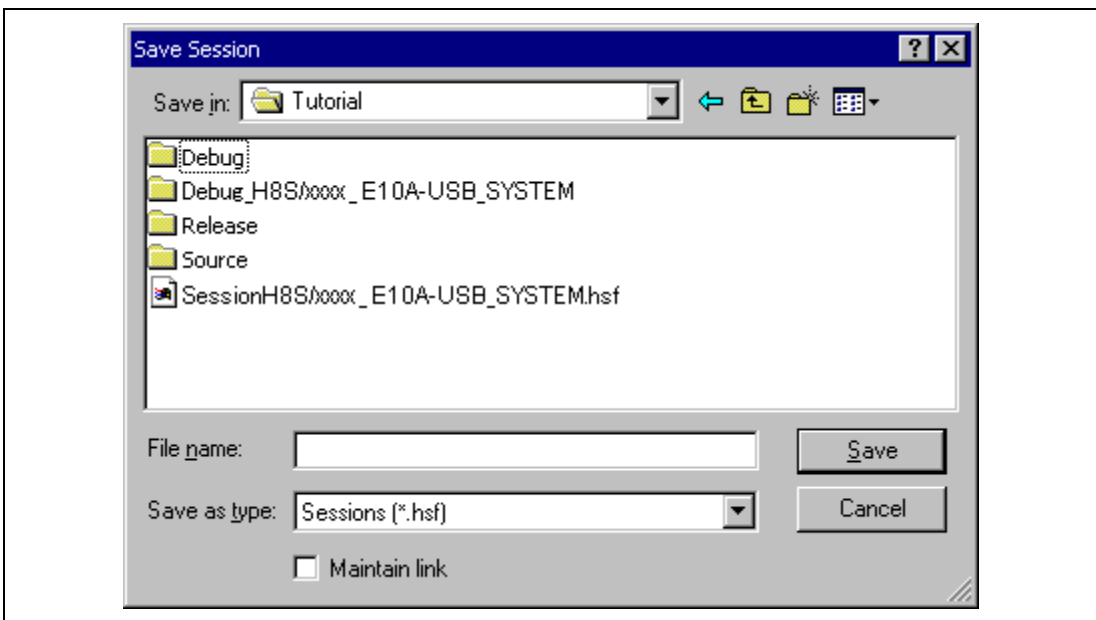
Figure 4.32 [Save Session] Dialog Box

4.3.3 Saving Session Information

➲ To save a session
Select [File -> Save Session].

4.4   Connecting the Emulator

Select either of the following two ways to connect the emulator:

(1) Connecting the emulator after the setting at emulator activation

Select [Debug settings] from the [Debug] menu to open the [Debug Settings] dialog box. It is possible to register the download module or the command batch file that is automatically executed at activation. For details on the [Debug Settings] dialog box, refer to section 4.2, Setting at Emulator Activation.

When the dialog box is closed after setting the [Debug Settings] dialog box, the emulator will automatically be connected.

(2) Connecting the emulator without the setting at emulator activation

Connect the emulator by simply switching the session file to one in which the setting for the emulator use has been registered.
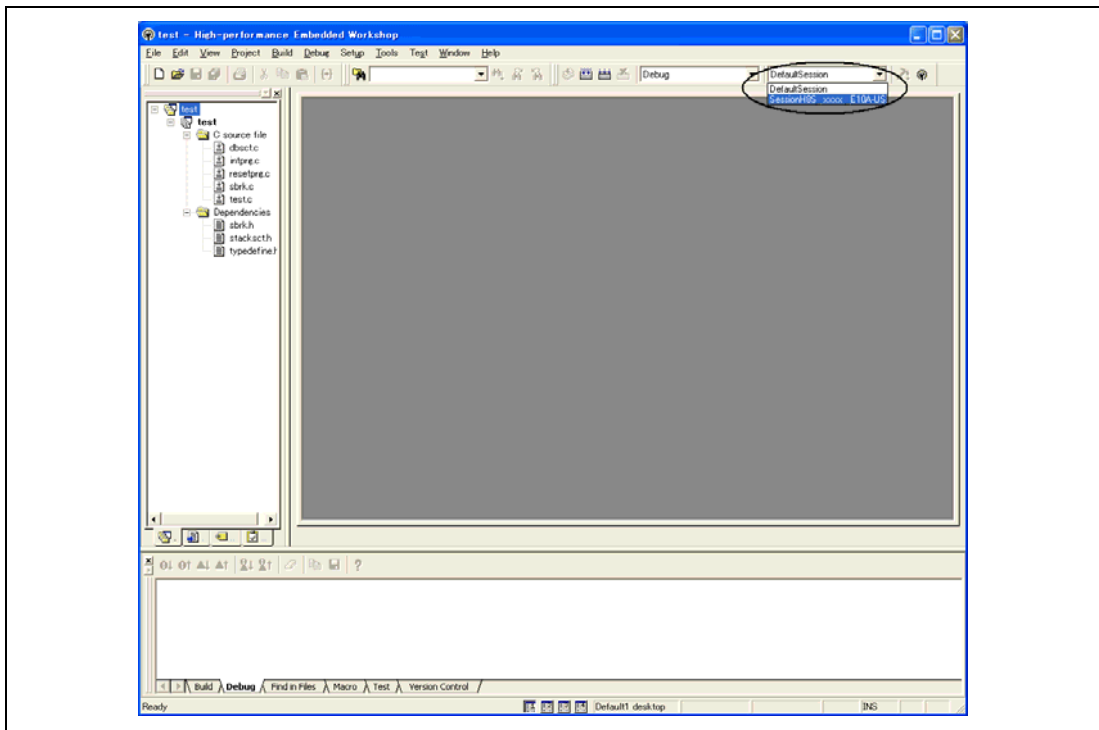
Figure 4.33   Selecting the Session File

In the list box that is circled in figure 4.33, select the session file name including the character string that has been set in the [Target name] text box in figure 4.9, [New Project -8/9- Setting the Debugger Options] dialog box. The setting for using the emulator has been registered in this session file.

After the session file name is selected, the emulator will automatically be connected. For details on the session file, refer to section 4.3, Debug Sessions.

4.5   Reconnecting the Emulator

When the emulator is disconnected, use the following way for reconnection:

Select [Debug -> Connect] or click the [Connect] toolbar button ( ). The emulator is connected.

Note: The emulator must be selected in the [Target] drop-down list box of the [Debug Settings] dialog box that is opened by selecting [Debug settings] from the [Debug] menu (see 4.2.1, Setting at Emulator Activation).

4.6   Ending the Emulator

When using the toolchain, the emulator can be exited by using the following two methods:

- Canceling the connection of the emulator being activated
- Exiting the High-performance Embedded Workshop

(1) Canceling the connection of the emulator being activated
Select [Disconnect] from the [Debug] menu or click the [Disconnect] toolbar button ( ).

(2) Exiting the High-performance Embedded Workshop
Select [Exit] from the [File] menu.

A message box may be displayed. If necessary, click the [Yes] button to save a session. After saving a session, the High-performance Embedded Workshop exits. If not necessary, click the [No] button to exit the High-performance Embedded Workshop.
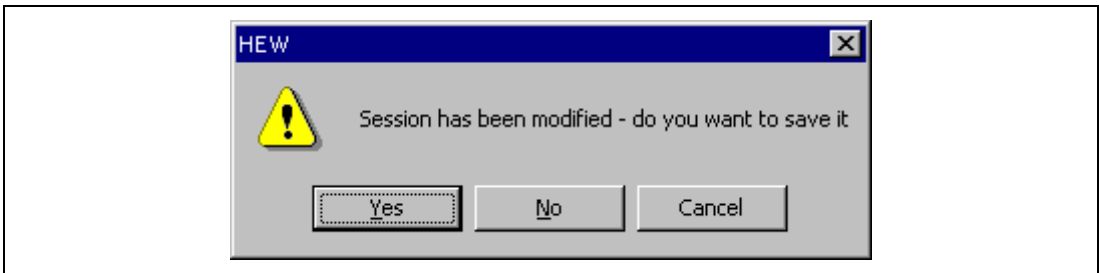
Figure 4.34   [Session has been modified] Message Box

Section 5   Debugging

This section describes the debugging operations and their related windows and dialog boxes.

5.1   Setting the Environment for Emulation

5.1.1   Opening the [Configuration] Dialog Box

Selecting [Options -> Emulator -> System…] or clicking the [Emulator System] toolbar button
(⊞) opens the [Configuration] dialog box.

5.1.2   [Emulator] Page

Sets the emulator operation conditions.
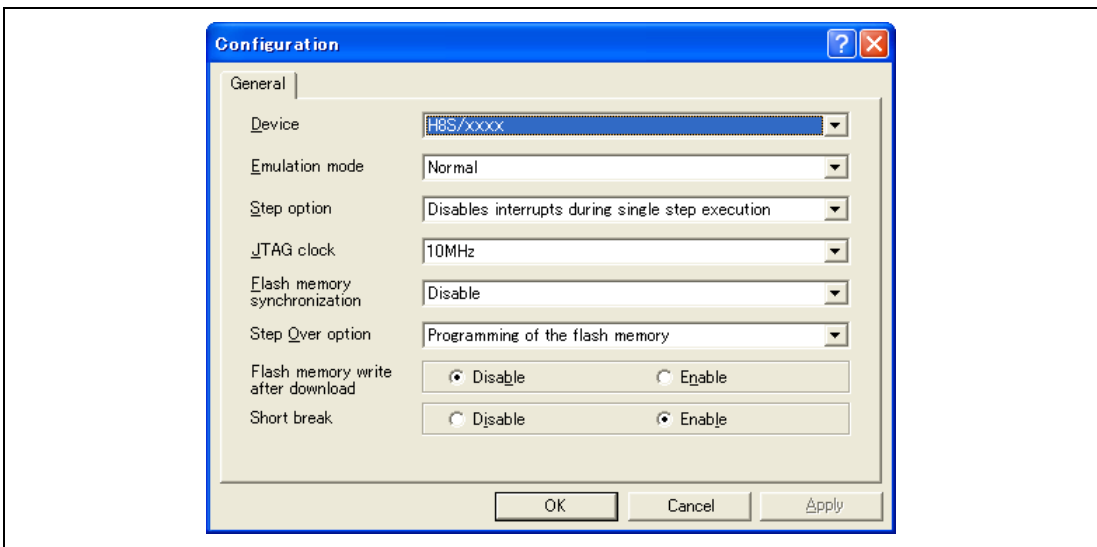
Figure 5.1   [Configuration] Dialog Box ([General] Page)

Note: The contents to be displayed differ depending on the MCU.

Items that can be displayed in the sheet are listed below.

Note: The items to be set differ between H8S, H8SX family and MCUs. For functions to be set
          in each MCU, refer to the online help.

[Device] combo box   Displays the MCU name.

[Emulation mode] combo box   Selects the emulation mode at user program execution.

- Normal:
  Performs normal emulation.
- No break:
  Temporarily disables breakpoint settings and executes user program.
- Sequential Break Condition 2-1[*1]:
  Program is halted when Break Condition 2 and Break Condition 1 are satisfied in that order.
- Sequential Break Condition 3-2-1[*1]:
  Program is halted when Break Condition 3, Break Condition 2, and Break Condition 1 are satisfied in that order.
- Sequential Break Condition 4-3-2-1[*1]:
  Program is halted when Break Condition 4, Break Condition 3, Break Condition 2, and Break Condition 1 are satisfied in that order.

[Step option] combo box   Enables or disables interrupts during step operation.

- Disables interrupts during single step execution:
  Interrupts during step operation are masked.
- Enables interrupts during single step execution:
  Interrupts during step operation are released.

[JTAG clock] combo box   Sets the frequency[*2] of the JTAG clock.

[Flash memory synchronization]   Selects whether or not the contents of the flash memory are
combo box   acquired by the emulator when the user program is stopped or
the position where the PC break is set is put back as the
original code.

When the flash memory is not reprogrammed by the user
program, its contents need not be acquired by the emulator.

If there is no problem with the state that the program in the
flash memory has been replaced as the PC break code, the
position where the PC break is set needs not be put back as
the original code.

- Disable:
  Read or program is not performed for the flash memory
  except when the emulator is activated, the flash memory
  area is modified, and the settings of the PC break to the
  flash memory area are changed.

- PC to flash memory:
  When the user program is stopped, the specified PC break
  code is replaced as the original instruction. Select this
  option if there is a problem with the state that the program in
  the flash memory has been replaced as the PC break code.

- Flash memory to PC:
  When the user program is stopped, the contents of the flash
  memory are read by the emulator. Select this option if the
  flash memory is reprogrammed by the user program.

- PC to flash memory, Flash memory to PC:
  When the user program is stopped, the contents of the flash
  memory are read by the emulator and the specified PC
  break code is replaced as the original instruction. Select this
  option if the flash memory is reprogrammed by the user
  program and there is a problem with the state that the
  program in the flash memory has been replaced as the PC
  break code.

RENESAS

[Step Over option] combo box   Enables or disables programming of the flash memory while executing step operation.

- Programming of the flash memory:
  Uses a software break for step operation (e.g. Step Over) and enables programming of the flash memory.

- No Programming of the flash memory:
  Uses Break Condition for step operation (e.g. Step Over) and disables programming of the flash memory.[3]

[Flash memory write after download] group box   Enables or disables programming of the flash memory after downloading the program.

- Disable:
  Disables programming of the flash memory after downloading the program.

- Enable:
  Enables programming of the flash memory after downloading the program.

[Parallel] group box   Selects the setting of memory access during user program execution.[1]

- Disable:
  Uses no emulator debugging functions to access memory. A short break will occur.

- Enable:
  Uses emulator debugging functions to access memory.

[Short break] group box   Selects whether or not memory should be accessed during user program execution.[4]

- Disable:
  Disables access to memory during user program execution.

- Enable:
  Enables access to memory during user program execution.

[Memory access] group box   Selects whether or not memory should be accessed during user program execution.[1]

- Disable:
  Disables access to memory during user program execution.

- Enable:
  Enables access to memory during user program execution.

RENESAS

Notes: 1. This function is supported by the H8SX family.

2. The range of frequency for the JTAG clock differs depending on the supported MCU.

3. The H8S family supports the optional Step Over function. When Step Over is performed after selecting [No Programming of the flash memory] for [Step Over option], the instruction that follows a JSR, BSR, or TRAPA instruction will also be executed. If there are consecutive JSR, BSR, or TRAPA instructions, execution proceeds until it reaches an instruction that is not JSR, BSR, or TRAPA. (When there are consecutive function calls in C language, the program steps all of these functions.)

4. This function is supported by the H8S family.

5.2　　Downloading a Program

This section describes how to download a program and view it as source code or assembly-language mnemonics.

Note:　After a break has been detected, the High-performance Embedded Workshop displays the location of the program counter (PC). In most cases, for example if an Elf/Dwarf2-based project is moved from its original path, the source file may not be automatically found. In this case, the High-performance Embedded Workshop will open a source file browser dialog box to allow you to manually locate the file.

5.2.1　　Downloading a Program

A load module to be debugged must be downloaded.

To download a program, select the load module from [Debug -> Download] or select [Download] from the popup menu opened by clicking the right-hand mouse button on the load module in [Download modules] of the [Workspace] window.

Note:　Before downloading a program, it must be registered to the High-performance Embedded Workshop as a load module. For registration, refer to section 4.2, Setting at Emulator Activation.

5.2.2   Viewing the Source Code

Select a source file name in the [Workspace] window and click [Open] in the popup menu to make
the High-performance Embedded Workshop display the source file in the editor. It is also possible
to display your source files by double-clicking on their names in the [Workspace] window.
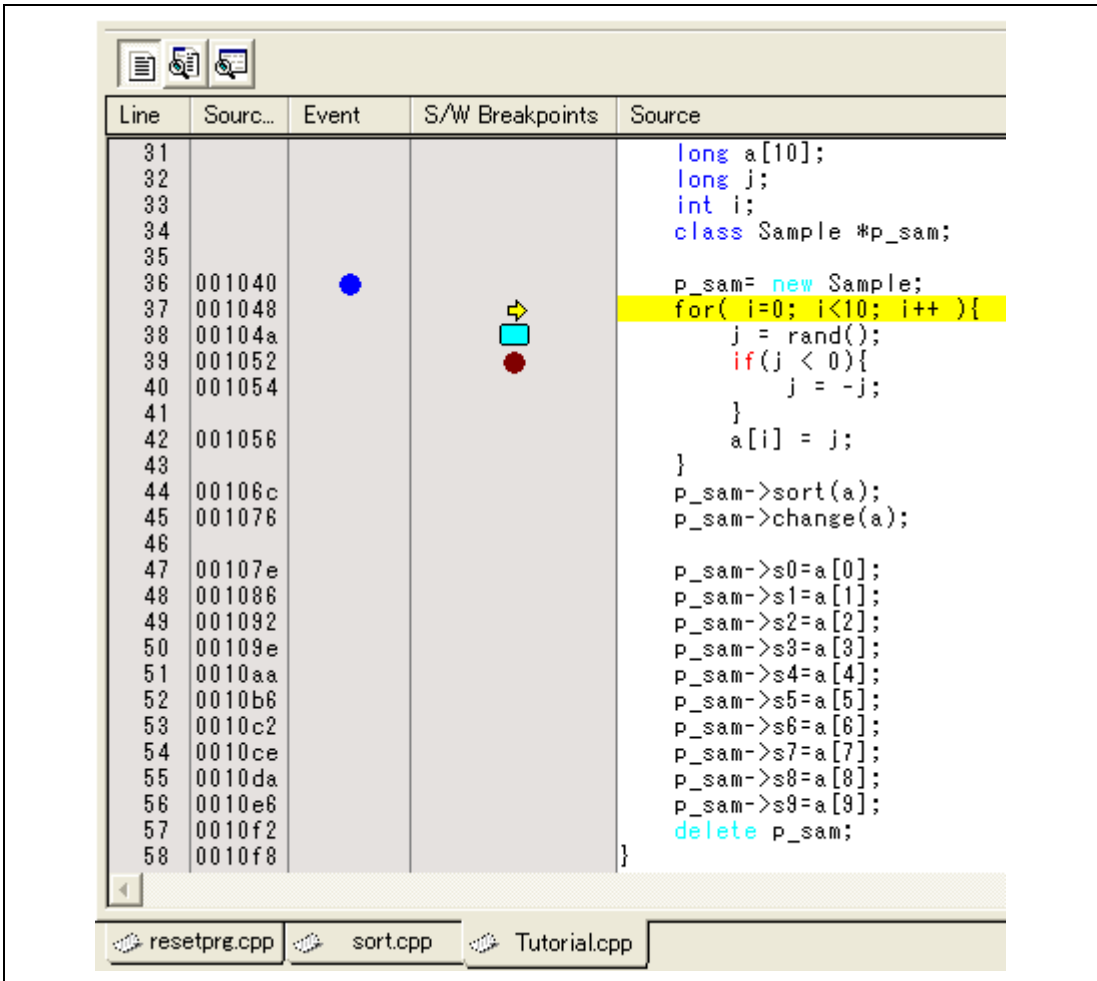

Figure 5.2   [Editor] Window

In this window, the following items are shown on the left as line information.

> The first column (Line number column): Source-line information

> The second column (Source address column): Address information

> The third column (Event column): Event information (break condition)

> The fourth column (S/W breakpoint column): PC, bookmark, and breakpoint information

Line number column
The line number that corresponds to the source file is displayed.

Source address column
When a program is downloaded, an address for the current source file is displayed on the Source address column. These addresses are helpful when setting the PC value or a breakpoint.

Event column
The Event column displays the following item:

●: An address condition as a break condition; only one such a position can be set.

This is also set by using the popup menu.

The bitmap symbol above is shown by double-clicking the Event column. This is also set by using the popup menu.
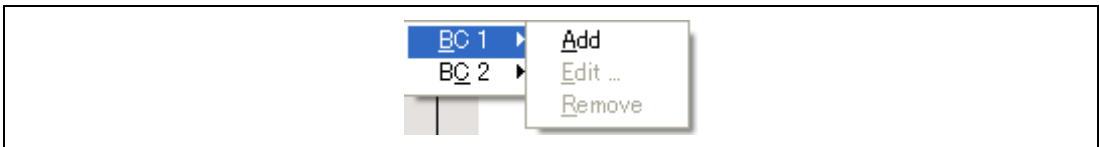

Figure 5.3   Popup Menu

Notes: 1. The contents of the Event column are erased when conditions other than the address condition are added to each channel by using the [Edit] menu or in the [Event] window.
2. The conditions for setting the break condition differ depending on the MCU. For details, refer to the online help.

S/W breakpoint column

S/W breakpoint column displays the following items:

🟦: A bookmark is set.

🔴: A breakpoint is set.

➡: PC location

● To switch off a column in all source files

1. Click the right-hand mouse button on the [Editor] window or select the [Edit] menu.
2. Click the [Define Column Format…] menu item.
3. The [Global Editor Column States] dialog box is displayed.
4. A check box indicates whether the column is enabled or not. If it is checked, the column is enabled. If the check box is gray, the column is enabled in some files and disabled in others. Deselect the check box of a column you want to switch off.
5. Click the [OK] button for the new column settings to take effect.
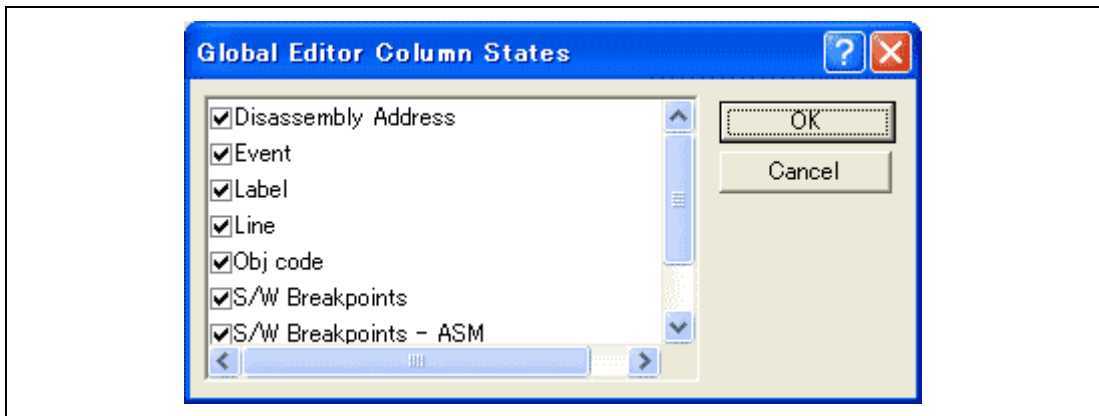
Figure 5.4   [Global Editor Column States] Dialog Box

● To switch off a column in one source file

1. Open the source file which contains the column you want to remove and right-click on the [Edit] window.
2. Click the [Columns] menu item to display a cascaded menu item. The columns are displayed in this popup menu. If a column is enabled, it has a tick mark next to its name. Clicking the entry will toggle whether the column is displayed or not.

5.2.3   Viewing the Assembly-Language Code

If there is an address at the cursor position in the [Editor] window, selecting [Go to Disassembly] in the pop-up menu opens the [Disassembly] window.

The first address displayed in the [Disassembly] window is the address where the cursor is placed in the [Editor] window.

If you do not have a source file, but wish to view code in the assembly-language level, either choose [View] -> [Disassembly…] or click on the [Disassembly] window's toolbar button (🖳). The [Disassembly] window opens at the current PC location and shows [Address] and [Code] (optional) which show the disassembled mnemonics (with labels when available).

Selecting the [Mixed display] toolbar button (🖳) displays both the source and the code. The following shows an example in this case.
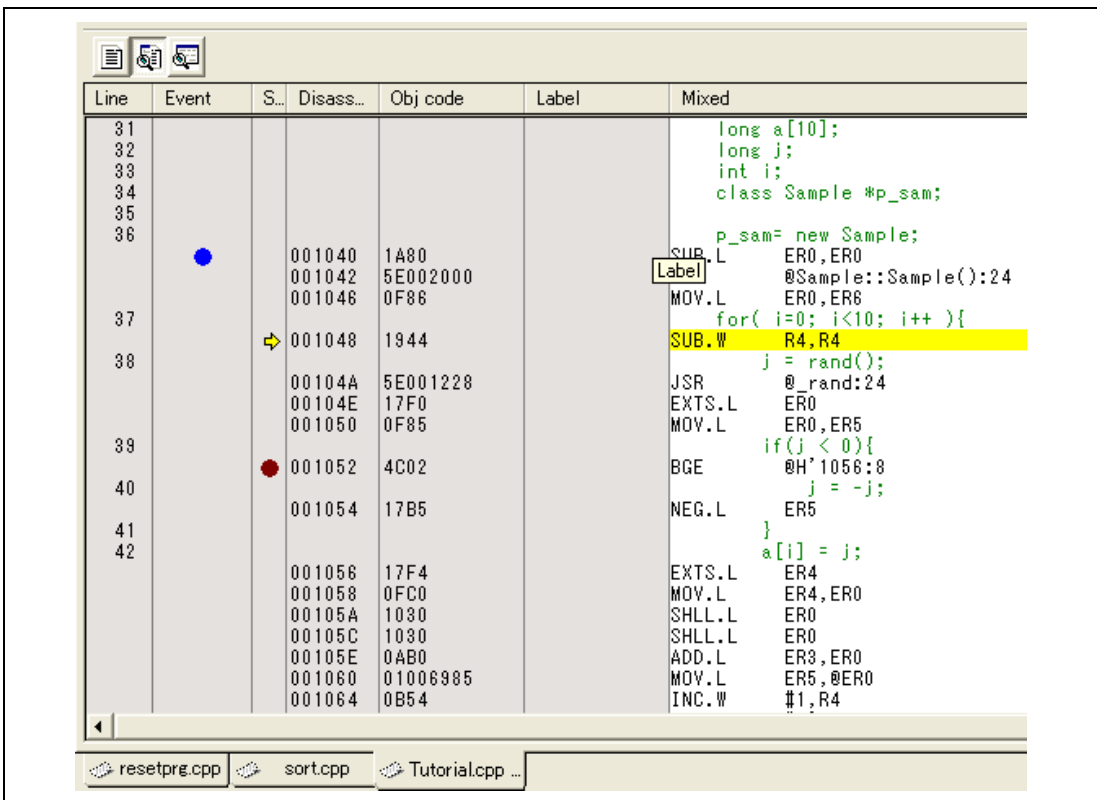

Figure 5.5   [Disassembly] Window

5.2.4   Modifying the Assembly-Language Code

You can modify the assembly-language code by double-clicking on the instruction that you want to change. The [Assembler] dialog box will be opened.
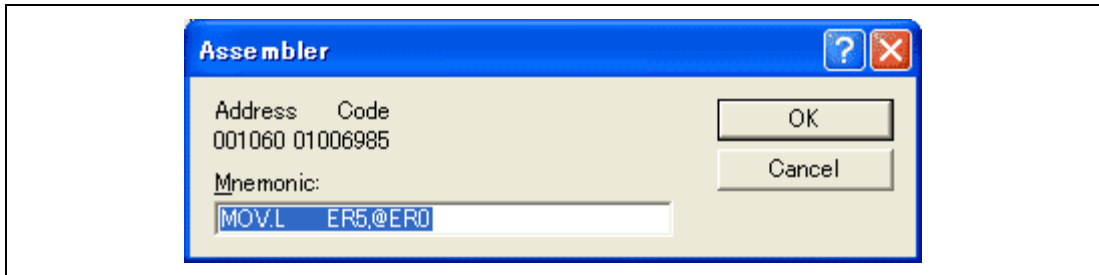
Figure 5.6   [Assembler] Dialog Box

The address, instruction code, and mnemonic are displayed. Enter the new instruction or edit the old instruction in the [Mnemonics] field. Pressing the [Enter] key will replace the memory content with the new instruction and move on to the next instruction. Clicking the [OK] button will replace the memory content with the new instruction and close the dialog box. Clicking the [Cancel] button or pressing the [Esc] key will close the dialog box without modifying the memory contents.

Note: The assembly-language code being displayed is the current memory content. If the memory contents are changed the [Assembler] dialog box and the [Disassembly] window will show the new assembly-language code, but the source file displayed in the [Editor] window will be unchanged. This is the same even if the source file contains assembly codes.

5.2.5   Viewing a Specific Address

When you are viewing your program in the [Disassembly] window, you may wish to look at another area of your program's code. Rather than scrolling through a lot of code in the program, you can go directly to a specific address. Select [Set Address…] from the popup menu, and the dialog box shown in figure 5.7 is displayed.
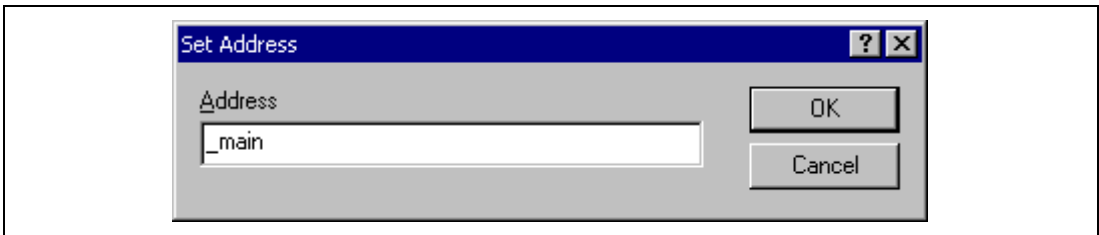

Figure 5.7   [Set Address] Dialog Box

Enter the address in the [Address] edit box and either click on the [OK] button or press the [Enter] key. A label name can also be specified as the address. The [Disassembly] window will be updated to show the code at the new address. When an overloaded function or a class name is entered, the [Select Function] dialog box opens for you to select a function.

5.2.6   Viewing the Current Program Counter Address

Wherever you can enter an address or value into the High-performance Embedded Workshop, you can also enter an expression. If you enter a register name prefixed by the hash character (#), the contents of that register will be used as the value in the expression. Therefore, if you enter the expression #pc in the [Set Address] dialog box, the [Editor] or [Disassembly] window will display the current PC address. It also allows the offset of the current PC to be displayed by entering an expression with the PC register plus an offset, e.g., #PC+0x100.

5.3   Using the Event Points

The emulator has the event point function that performs breaking by specifying higher-level conditions along with the PC breakpoints standard for the High-performance Embedded Workshop.

5.3.1   PC Breakpoints

When the instruction of the specified address is fetched, the user program is stopped. Up to 255 points can be set.

5.3.2   Break Conditions

Break conditions can be used for higher-level conditions such as the data condition as well as specification of the single address.

5.3.3   Opening the [Event] Window

Select [View -> Code -> Eventpoints] or click the [Eventpoints] toolbar button () to open the [Event] window.

The [Event] window has the following two sheets:

[Breakpoint] sheet: Displays the settings made for PC breakpoints. It is also possible to set, modify, and cancel PC breakpoints.

[Breakcondition] sheet: Displays or sets the settings made for break condition channels.

5.3.4   Setting PC Breakpoints

It is possible to display, modify, and add PC breakpoints on the [Breakpoint] sheet.
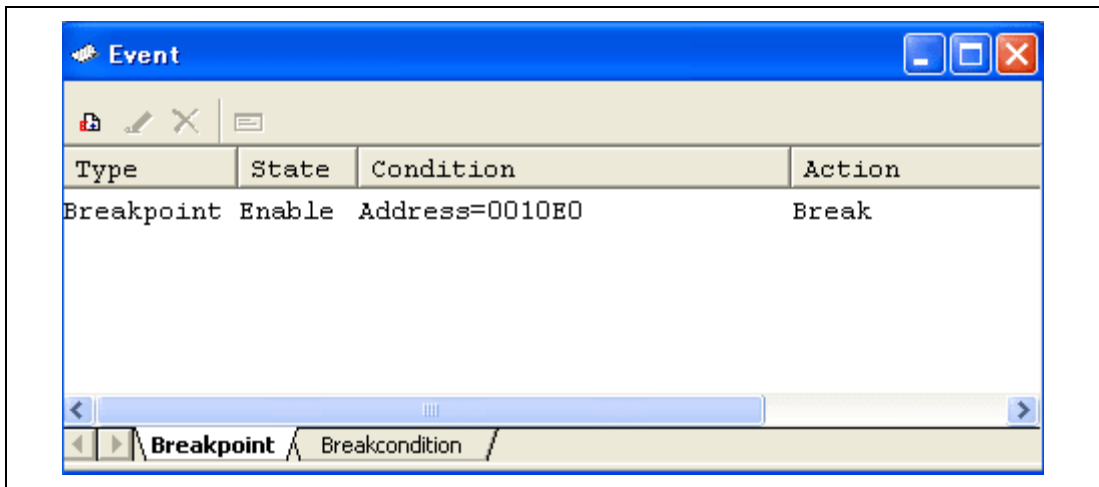

Figure 5.8   [Event] Window ([Breakpoint] Sheet)

This window displays and sets the breakpoints. Items that can be displayed in the sheet are listed below.

[Type]   Breakpoint

[State]   Whether the breakpoint is enabled or disabled

         [Condition]   An address that the breakpoint is set
            Address = Program counter (Corresponding file name, line, and symbol name)

[Action]   Operation of the emulator when a break condition is satisfied
            Break: Breaks program execution

When a breakpoint is double-clicked in this window, the [Set Break] dialog box is opened and break conditions can be modified.

A popup menu containing the following options is available by right-clicking within the window.

5.3.5   Add

Sets breakpoints. Clicking this item will open the [Set Break] dialog box and break conditions can be specified.

5.3.6   Edit

Only enabled when one breakpoint is selected. Select a breakpoint to be edited and click this item. The [Set Break] dialog box will open and break conditions can be changed.

5.3.7   Enable

Enables the selected breakpoint(s).

5.3.8   Disable

Disables the selected breakpoint(s). When a breakpoint is disabled, the breakpoint will remain in the list; when specified conditions have been satisfied, a break will not occur.

5.3.9   Delete

Removes the selected breakpoint. To retain the details of the breakpoint but not have it cause a break when its conditions are met, use the Disable option (see section 5.3.8, Disable).

5.3.10   Delete All

Removes all breakpoints.

5.3.11   Go to Source

Only enabled when one breakpoint is selected. Opens the [Editor] window at the address of the breakpoint.
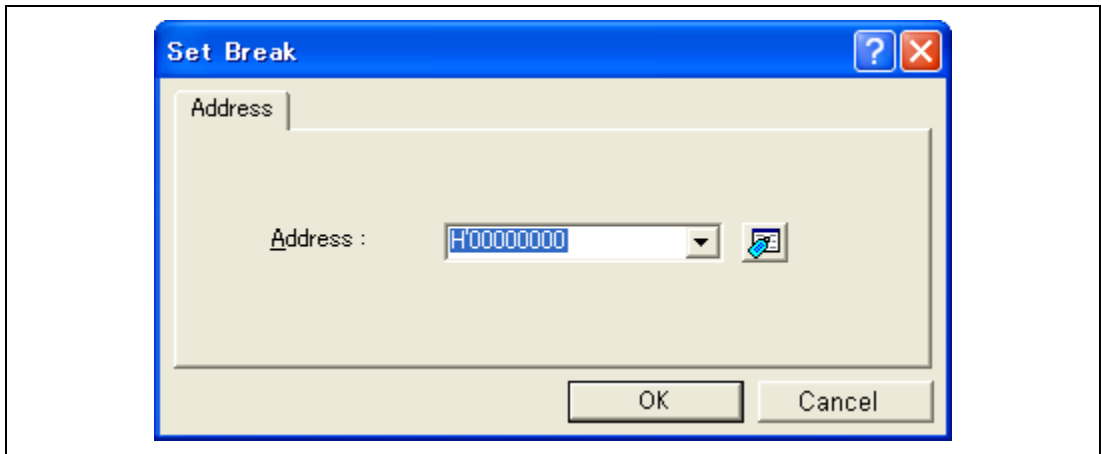
5.3.12   [Set Break] Dialog Box

Figure 5.9   [Set Break] Dialog Box

This dialog box specifies break conditions.

A breakpoint address to be set is specified in the [Address] edit box. The PC register can also be specified such as #PC. Up to 255 breakpoints can be specified.

The contents to be set differ depending on the product. For details, refer to the on-line help for each product.

When [Address] is selected, if an overloaded function or class name including a member function is specified in address, the [Select Function] dialog box opens.

Clicking the [OK] button sets the break conditions. Clicking the [Cancel] button closes this dialog box without setting the break conditions.

Notes on Setting the [Set Break] 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 flash memory or RAM area. However, a software break cannot be set to the following addresses:
   — An area other than the flash memory or RAM
   — An area occupied by the E10A-USB emulator program
   — An instruction in which Break Condition 1 is satisfied
3. During step execution, a software breakpoint is disabled.
4. A condition set at Break Condition 1 is disabled immediately after starting execution when an instruction at a software breakpoint is executed. A break does not occur even if a condition of Break Condition 1 is satisfied immediately after starting the execution.
5. When execution resumes from the breakpoint address after the program execution stops at the software breakpoint, single-step execution is performed at the address before execution resumes. Therefore, realtime operation cannot be performed.
6. Settings of software breakpoint and Break Condition 1 are invalid while the STEP OVER function is being used.

5.3.13   Setting Break Conditions

On the [Breakcondition] sheet, the settings for break conditions are displayed, modified, and added.
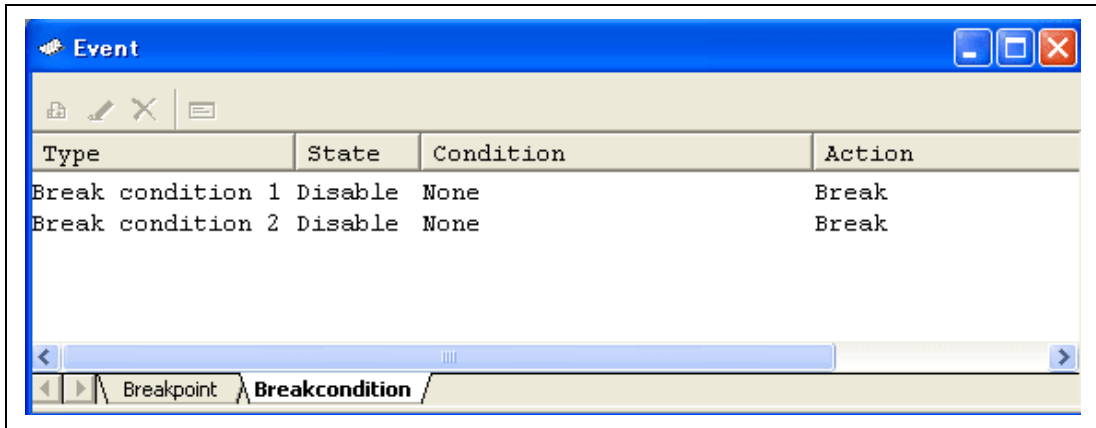

Figure 5.10   [Event] Window ([Breakcondition] Sheet)

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

This window displays and sets the break condition.

Notes on Setting the Break Condition:

1. When [Step In], [Step Over], or [Step Out] is selected, the settings of Break Condition 1 are disabled.
2. Setting of Break Condition 1 is disabled when an instruction to which a BREAKPOINT has been set is executed.
3. When step over function is used, the settings of BREAKPOINT and Break Condition 1 are disabled.

Items that can be displayed in the sheet are listed below.

[Type]   Break channel number

[State]   Whether the breakpoint is enabled or disabled

[Condition]   A condition that satisfies a break

[Action]   Operation of the emulator when a break condition is satisfied
                  Break: Breaks program execution

When a Break condition 1 is double-clicked in this window, the [Break condition 1] dialog box is opened and break conditions can be modified. For details on the [Break condition 1] dialog box, refer to the online help for each product.

A popup menu containing the following options is available by right-clicking within the window.

5.3.14   Edit...

Only enabled when one break channel is selected. Select a break channel to be edited and click this item. The [Break condition 1] dialog box will open and break conditions can be changed.

5.3.15   Enable

Enables the selected break channel(s). A break channel that the condition has not been set is not enabled.

5.3.16   Disable

Disables the selected break channel(s). When a break channel is disabled, a break will not occur even if specified conditions have been satisfied.

5.3.17   Delete

Initializes the condition of the selected break channel. To retain the details of the break channel but not have it cause a break when its conditions are met, use the Disable option (see section 5.3.16, Disable).

5.3.18   Delete All

Initializes conditions of all break channels.

5.3.19   Go to Source

Only enabled when one break channel is selected. Opens the [Editor] window at address of break channel.

If an address value has not been set to the break channel, this option cannot be used.

5.3.20   Sequential Conditions

Sets the sequential condition of the break channel.

Note: This function is set in [Emulation mode] of the [Configuration] dialog box (refer to section 5.1.2, [Emulator] Page).

5.3.21   Editing Break Conditions

Handlings for settings other than PC breakpoints and break conditions are common. The following describes examples of such handling.

5.3.22   Modifying Break Conditions

Select a break condition to be modified, and choose [Edit...] from the popup menu to open the dialog box for the event, which allows the user to modify the break conditions. The [Edit...] menu is only available when one break condition is selected.

5.3.23   Enabling Break Conditions

Select a break condition and choose [Enable] from the popup menu to enable the selected break condition.

5.3.24   Disabling Break Conditions

Select a break condition and choose [Disable] from the popup menu to disable the selected break condition. When a break condition is disabled, the break condition will remain in the list, but an event will not occur when the specified conditions have been satisfied.

5.3.25   Deleting Break Conditions

Select a break condition and choose [Delete] from the popup menu to remove the selected break condition. To retain the break condition but not have it cause an event when its conditions are met, use the [Disable] option (see section 5.3.24, Disabling Break Conditions).

5.3.26   Deleting All Break Conditions

Choose [Delete All] from the popup menu to remove all break conditions.

5.3.27   Viewing the Source Line for Break Conditions

Select a break condition and choose [Go to Source] from the popup menu to open the [Editor] or [Disassembly] window at the address of the break condition. The [Go to Source] menu is only available when break conditions that have the corresponding source file are selected.

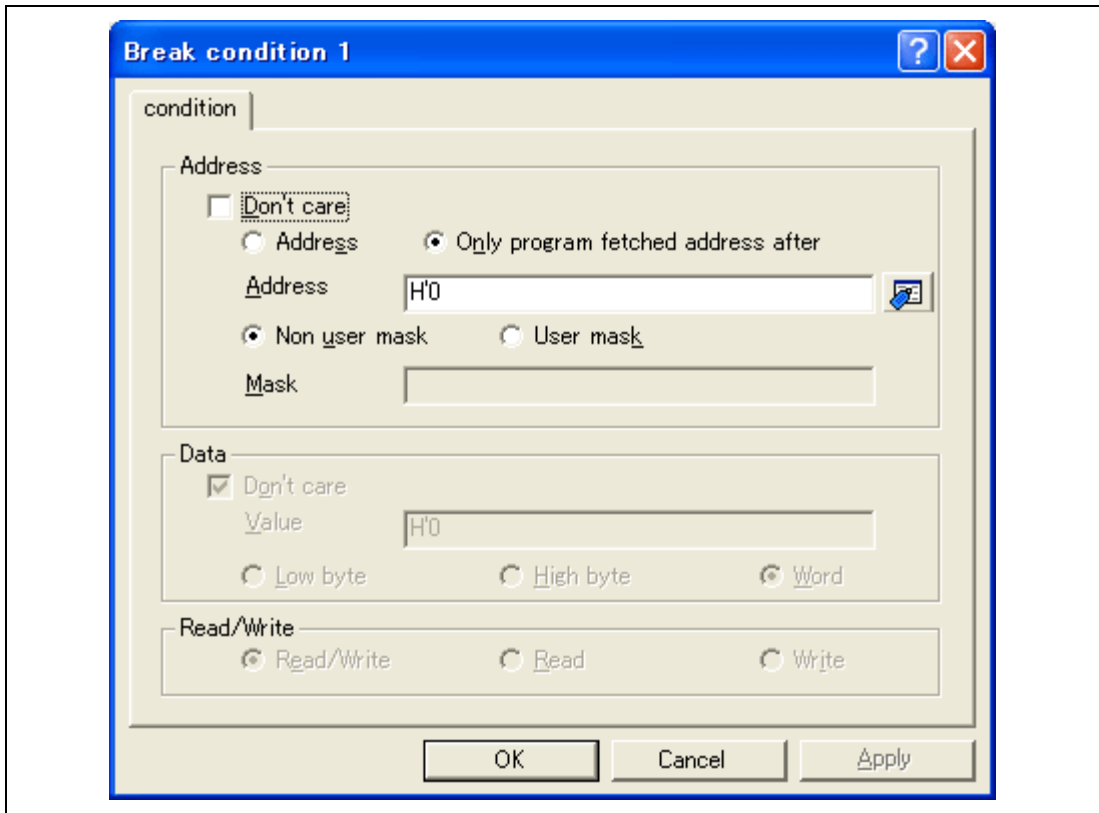### 5.3.28   [Break condition 1] Dialog Box

Figure 5.11   [Break condition 1] Dialog Box (H8S)

This page sets the address bus, data bus, and read/write cycle conditions.

Note: The items that can be set in this dialog box differ depending on the MCU. For details on the settings for each MCU, refer to the online help.

The [Address] group box sets address bus conditions.

Option	Description
[Don't care] check box	Does not set address conditions.
[Address] radio button	Sets the use of the normal address bus 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.
[Mask] edit box	After the radix has been set as ″H″ (hexadecimal) or ″B″ (binary), the setting ″∗′″ selects masking of the corresponding digit.

The contents of an option that can be will change depending on the radio button selected.

Option	Description
[Address] radio button	All options can be set.
[Only program fetched address after] radio button	The [Address] option can be set.

The [Data] group box sets the data bus conditions.

Option	Description
[Don't care] check box	No data conditions are set.
[Value] edit box	Sets the data bus with a number.
[Low byte] radio button	Sets data access size as the lower 8-bit access.
[High byte] radio button	Sets data access size as the upper 8-bit access.
[Word] radio button	Sets data access size as the word access.

The [Read/Write] group box sets the read/write cycle conditions.

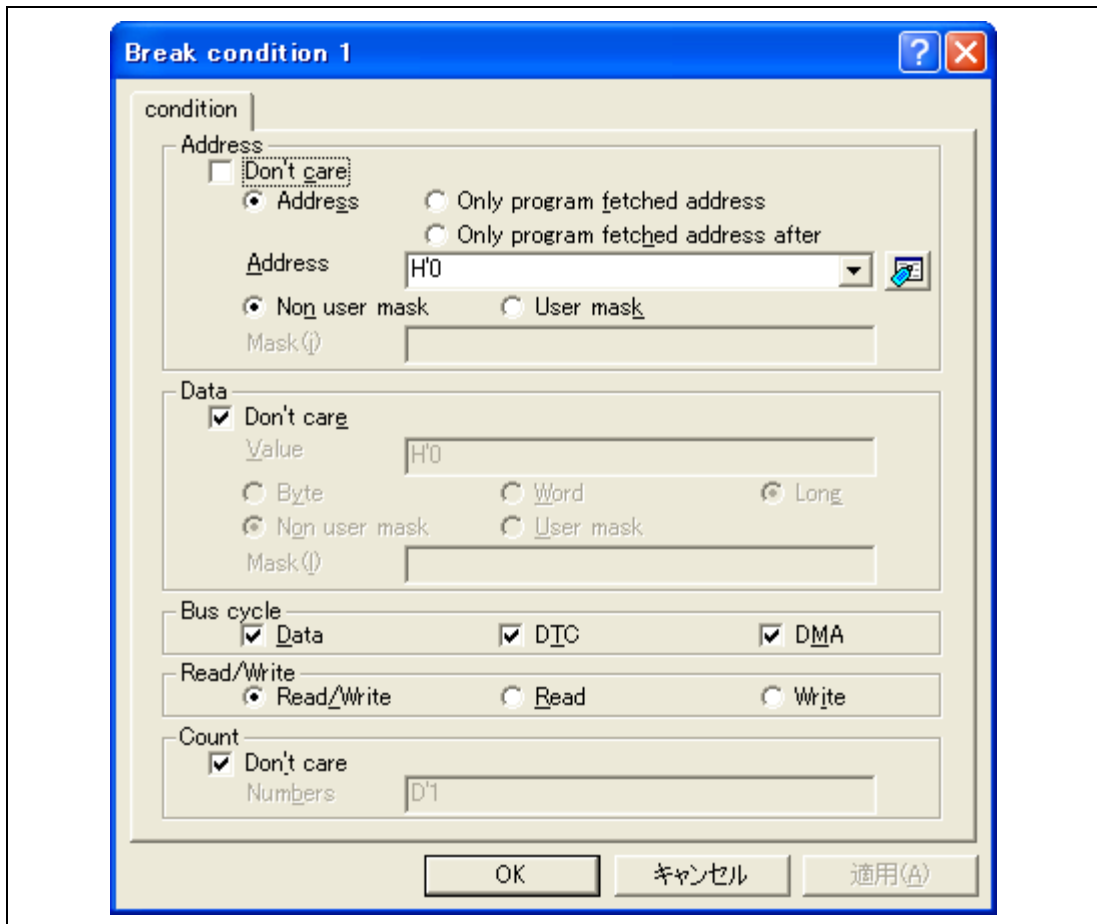| Option	Description
[Read/Write] radio button	Sets the read/write cycle conditions as break conditions.
[Read] radio button	Sets read cycles as break conditions.
[Write] radio button	Sets write cycles as break conditions.

Figure 5.12　[Break condition 1] Dialog Box (H8SX)

This page sets the address bus, data bus, and read/write cycle conditions.

Note:　The items that can be set in this dialog box differ depending on the MCU. For details on the settings for each MCU, refer to the online help.

The [Address] group box sets address bus conditions.

Option	Description
[Don't care] check box	Does not set address conditions.
[Address] radio button	Sets the use of the normal address bus as break conditions.
[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	Does not set mask conditions.
[User mask] radio button	Sets mask conditions.
[Mask] edit box	Sets a value to be masked when [User mask] is selected. For masked bits, the break condition is satisfied for any values.

The contents of an option that can be changed depending on the radio button selected.

Option	Description
[Address] radio button	All options can be set.
[Only program fetched address] radio button	The [Address] and [Count] options can be set.
[Only program fetched address after] radio button	The [Address] and [Count] options can be set.

The [Data] group box sets the data bus conditions.

Option	Description
[Don't care] check box	No data conditions are set.
[Value] edit box	Sets the data bus 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.
[Mask enabled] radio button	Sets mask conditions.
[Mask] edit box	Sets a value to be masked when [Mask enabled] is selected. For masked bits, the break condition is satisfied for any values.

The [Bus state] group box sets the bus state conditions.

Option	Description
[Data] check box	Sets the execution cycle as break conditions.
[Dtc] check box	Specifies DTC cycle as break conditions.
[Dma] check box	Specifies DMA cycle as break conditions.

The [Read/Write] group box sets the read/write cycle conditions.

Option	Description
[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.

The [Count] group box sets the break condition satisfaction conditions.

Option	Description
[Don't care] check box	Sets no satisfaction count conditions.
[Numbers] edit box	Sets the satisfaction count as a break condition. The maximum count is 65,535. Breaks when the conditions set by the [Break Condition] dialog box for the specified times are satisfied. The default is H'1.

5.4   Viewing the Trace Information
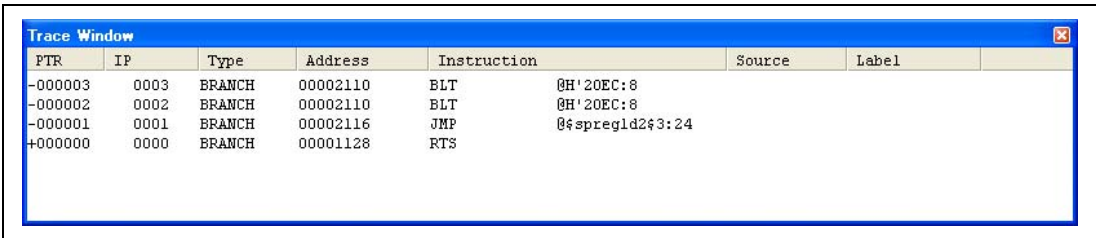
For the description on the trace function, refer to section 2.2, Trace Functions.

5.4.1   Opening the [Trace] Window

To open the [Trace] window, choose [View -> Code -> Trace] or click the [Trace] toolbar button
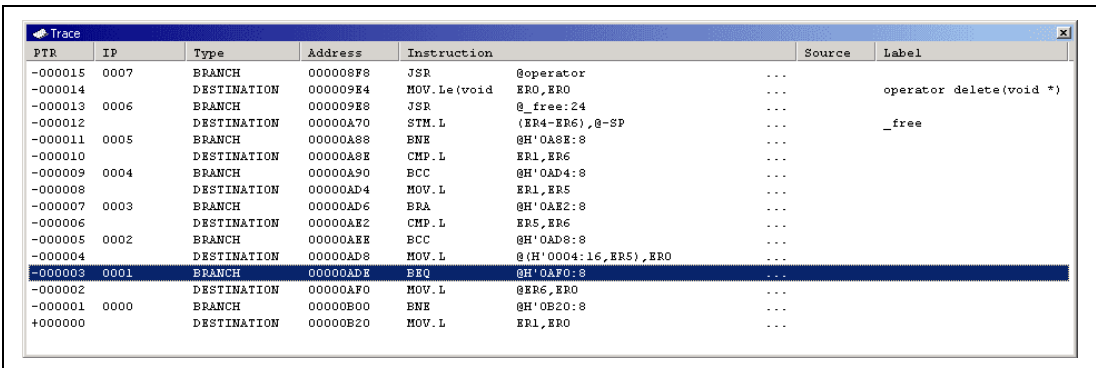(☐).

5.4.2   Acquiring Trace Information

The acquired trace information is displayed in the [Trace] window. The four-channel branch-
source addresses are displayed in H8 family. The eight-channel branch-source or branch-
destination addresses are displayed in H8SX family.

Figure 5.13   [Trace] Window (H8S)

Figure 5.14   [Trace] Window (H8SX)

This window displays the following trace information items:

[PTR]   Pointer to a location in the trace buffer (+0 for the last executed instruction)

[IP]   The amount of acquired trace information

[Type]   Type of branch:
                              BRANCH: branch source
                              DESTINATION: branch destination

[Address]   Instruction address

[Instruction]   Instruction mnemonic

[Source]   The C/C++ or assembly-language source program

[Label]   Label information

It is possible to hide any column not necessary in the [Trace] window. Selecting a column you want to hide from the popup menu displayed by clicking the right-hand mouse button on the header column hides that column. To display the hidden column, select the column from the said popup menu again.

Note: The number of branch instructions that can be acquired by a trace and the trace display differ depending on the product. For details on the specifications of each product, refer to the online help.

5.4.3   Clearing the Trace Information

When [Clear] is selected from the popup menu, the trace buffer that stores the trace information becomes empty. If several [Trace] windows are open, all [Trace] windows will be cleared as they all access the same buffer.

5.4.4   Saving the Trace Information in a File

Select [Save...] from the popup menu to open the [Save As] file dialog box, which allows the user to save the information displayed in the [Trace] window as a text file. A range can be specified based on the [PTR] number. Note that this file cannot be reloaded into the [Trace] window.

5.4.5   Viewing the [Editor] Window

The [Editor] window corresponding to the selected trace record can be displayed in the following two ways:

- Select a trace record and choose [View Source] from the popup menu.
- Double-click a trace record

The [Editor] or [Disassembly] window opens and the selected line is marked with a cursor.

5.4.6   Trimming the Source

Choose [Trim Source] from the popup menu to remove the white space from the left side of the source.

When the white space is removed, a check mark is shown to the left of the [Trim Source] menu. To restore the white space, choose [Trim Source] while the check mark is shown.

5.5   Using the Start/Stop Function

The start/stop functions are useful if you wish to control the user system in synchronization with the user program execution being started or stopped.

5.5.1   Opening the [Start/Stop Function Setting] Dialog Box

The start/stop functions are useful if you wish to control the user system in synchronization with the user program execution being started or stopped. To open the [Start/Stop Function Setting] dialog box, select [Setup -> Emulator -> Start/Stop Function Setting] or click on the [Start/Stop Function dialog box] toolbar button ( ).

Figure 5.15   [Start/Stop Function Setting] Dialog Box

The [Start/Stop Function Setting] dialog box contains the following options.

[Run the specified routine just before the user program execution is started.] checkbox :	Select this box if you wish to run a user routine just before the user program execution is started.	
	[Start address]:	Specify an address where a routine will start to run just before the user program execution is started. You can also specify a label.
[Run the specified routine just after the user program execution is stopped.] checkbox :	Select this box if you wish to run a user routine just after the user program execution is stopped.	
	[Start address]:	Specify an address where a routine will start to run just after the user program execution is stopped. You can also specify a label.

5.5.2   Specifying the Routine to be Executed

It is possible to specify the respective routines immediately before starting and immediately after halting the user program execution.

When the [The specified routine is executed immediately before execution of the user's program.] check box is selected, the routine specified in the [starting address] combo box, which is below this check box, is executed immediately before starting user program execution.

If no specified routine is executed immediately before starting user program execution, do not select the [The specified routine is executed immediately before execution of the user's program.] check box.

When the [The specified routine is executed immediately after execution of the user's program.] check box is selected, the routine specified in the [starting address] combo box, which is below this check box, is executed immediately after stopping user program execution.

If no specified routine is executed immediately before starting user program execution, do not select the [The specified routine is executed immediately after execution of the user's program.] check box.

5.5.3   Restrictions on the Start/Stop Function

The start/stop function has the following restrictions.
- An RTS instruction must be written at the end of the user routine.
- The user routine must end.
- No break should occur within the user routine.
- No interrupt should be accepted in the user routine.
- The user routine to run immediately before the user program execution must end within 2 msec.
- The user routine to run after the user program execution is stopped must end within 30 sec.
- As the stack area for the routine specified for execution before the user program starts, use the firmware stack of the emulator. The stack area is no larger than H'80 bytes.
- The user stack should be used by the routine specified for execution immediately after execution of the user program ends.
- If the user routine is to use a general-purpose register, saving the contents of the register, making the initial setting, and restoring the contents should all be done within the routine.
- The specified routine is also executed when the following operations are performed:
  — Step operations
  — Watching or changing memory during user program execution. ∗

Note: When parallel access is enabled and memory access is disabled, the user routine is not executed.

- The device groups listed below do not support this function.
  — -H8S/2118 device group
  — -H8S/2378 device group
  — -H8S/2339 device group
  — -H8S/2215R device group

5.6   Operation with Multiple Debugging Platforms

The High-performance Embedded Workshop is capable of operation with multiple debugging platforms.

When selecting [Renesas High-performance Embedded Workshop] -> [High-performance Embedded Workshop] from [Programs] in the [Start] menu to initiate another High-performance Embedded Workshop, two emulators can be used separately for debugging.

5.6.1   Distinguishing Two Emulators

Connect two emulators to the USB connector. Then, initiate the High-performance Embedded Workshop using the tutorial workspace. The following message is displayed.

Figure 5.16   Message for Driver Selection

Click the [OK] button. The following dialog box will appear.
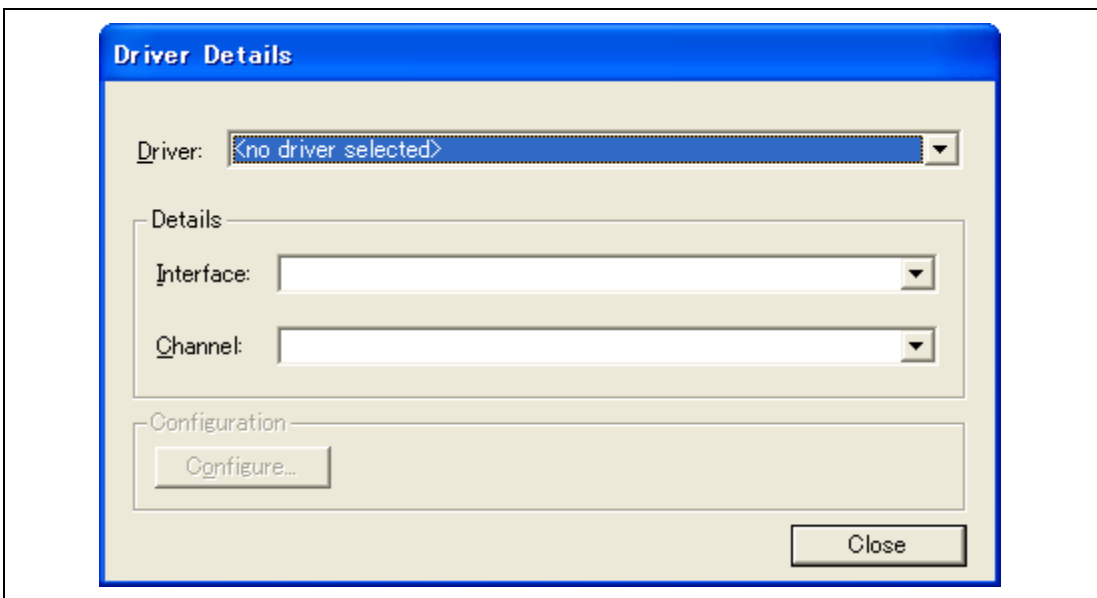
Figure 5.17   [Driver Details] Dialog Box (1)

Select [Renesas E-Series USB Driver] from the [Driver] drop-down list box and open the [Channel] drop-down list box. Channel information for two emulators is displayed in the [Channel] drop-down list box as shown in figure 5.18.
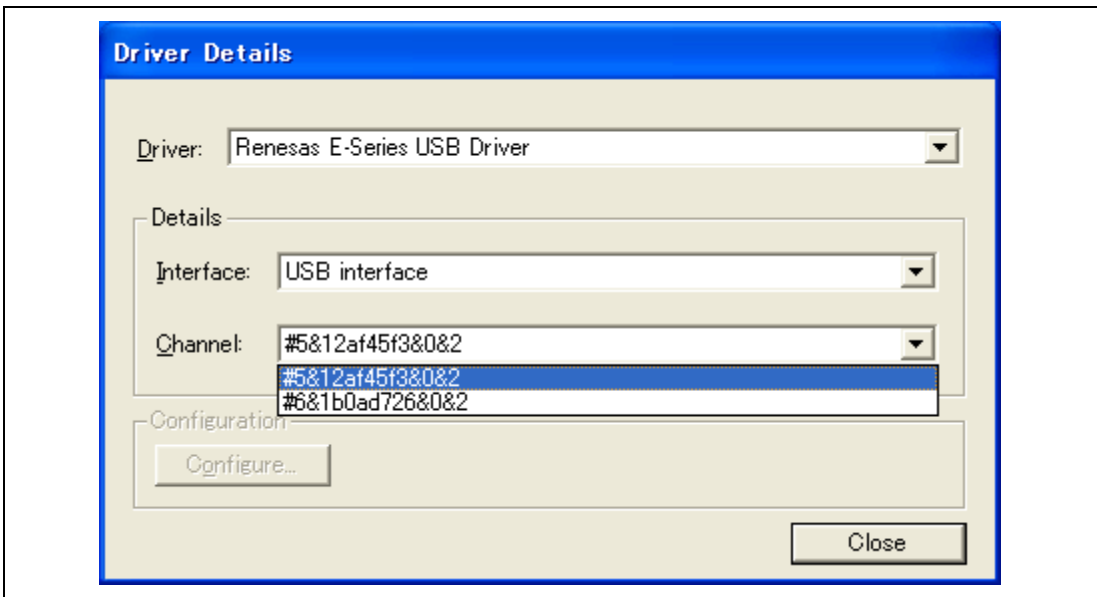
Figure 5.18   [Driver Details] Dialog Box (2)

Information displayed in figure 5.18 is the information of the USB connector to which the emulators are connected.

Note: The displayed character strings of the information differ according to the host computer's environment.

Check which of the character strings of information show emulators.

Select [<no driver selected>] in the [Driver] drop-down list box and disconnect one emulator from the USB connector. After that, select [Renesas E-Series USB Driver] in the [Driver] drop-down list box. Only information on the USB connector that the emulator is connected to is displayed in the [Channel] drop-down list box.

The above procedures are used to discern which of the emulators are indicated by the displayed character strings of information in the [Channel] drop-down list box.

When initiating the master High-performance Embedded Workshop, in the [Channel] drop-down list box, select the information on the USB connector of the emulator that is connected to the master CPU. Initiate the High-performance Embedded Workshop using the normal procedures.

When initiating the slave High-performance Embedded Workshop, in the [Channel] drop-down list box, select the information on the USB connector of the emulator that is connected to the slave CPU.

Display example:

- E10A-USB: [Cont0] – [Port1]

    The emulator is connected to port 1 of USB controller 0.

- E10A-USB: [Cont4] – [Port7] [Port3]

    The emulator is connected to general hub port 3 which has been connected to port 7 of USB controller 4 (this is displayed when the USB hub is connected).

Note: Connecting Two or More Emulators
        If two or more emulators are to be connected to a single personal computer, the emulator that was connected earlier should be in the break state whenever you connect another emulator.

5.7   Saving of Hardware-Break Settings

5.7.1   Saving and Loading Hardware-Break Settings

This facility allows you to save or load hardware-break settings, i.e. the "Break condition" settings for all channels shown on the [Breakcondition] sheet of the [Eventpoints] window (note that the breakpoints that have been set will not be saved). The hardware-break settings are saved in a file with the ".bcs" extension and, after that, you can load these settings. This makes it easier to switch the hardware-break settings for use in debugging.

5.7.2   How to Save Hardware-Break Settings

Use the [Save Break condition] dialog box to save hardware-break settings. To open the [Save Break condition] dialog box, select [Setup -> Emulator -> Save Break condition] or click on the [Save Break condition] toolbar button (   ).
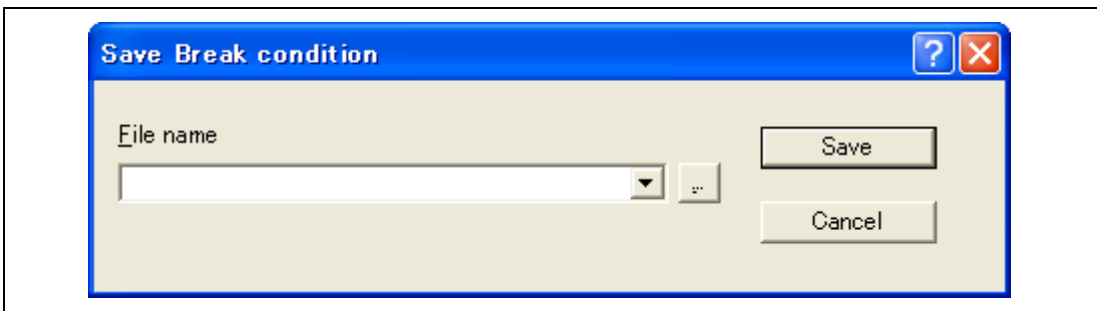
Figure 5.19   [Save Break condition] Dialog Box

Select a file name from the [File name] combo box. If you wish to create a new file, click on the [▪▪] button and enter the name of the file. Clicking on the [Save] button saves the settings.

5.7.3   How to Load Hardware-Break Settings

Use the [Load Break condition] dialog box to load hardware-break settings. To open the [Load Break condition] dialog box, select [Setup -> Emulator -> Load Break condition] or click on the [Load Break condition] toolbar button (▒).
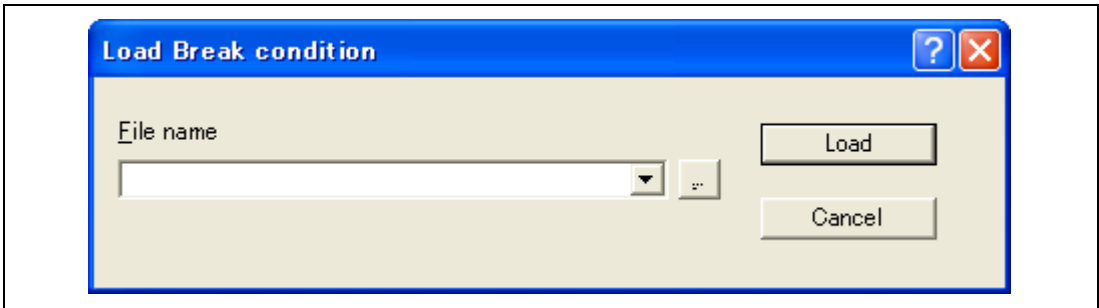
Figure 5.20   [Load Break condition] Dialog Box

Select a file name from the [File name] combo box. If you wish to specify a file that is not included in the list, click on the [▪▪] button and enter the name of the file. Clicking on the [Load] button loads the settings.

Section 6   Tutorial

6.1   Introduction

This section describes the main functions of the emulator by using a tutorial program. The explanation of the tutorial program will be given in this section using the H8S device. When the H8SX device is used, the displays of some dialog boxes will differ. However, the operation similar to that of the H8S device can be performed.

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.cpp` contains source code for the tutorial program of the H8S, H8SX family devices. The file `Tutorial.abs` is a compiled load module in the Elf/Dwarf2 format.

Note: After recompilation, the addresses may differ from those given in this section.

6.2   Running the High-performance Embedded Workshop

To run the High-performance Embedded Workshop, refer to section 3.11, System Check.

6.3   Setting up the Emulator

This section describes the basic settings of the emulator.

6.4   Setting the [Configuration] Dialog Box

Select [Emulator] then [Systems…] from the [Setup] menu for the basic settings of the emulator.
The [Configuration] dialog box is displayed.
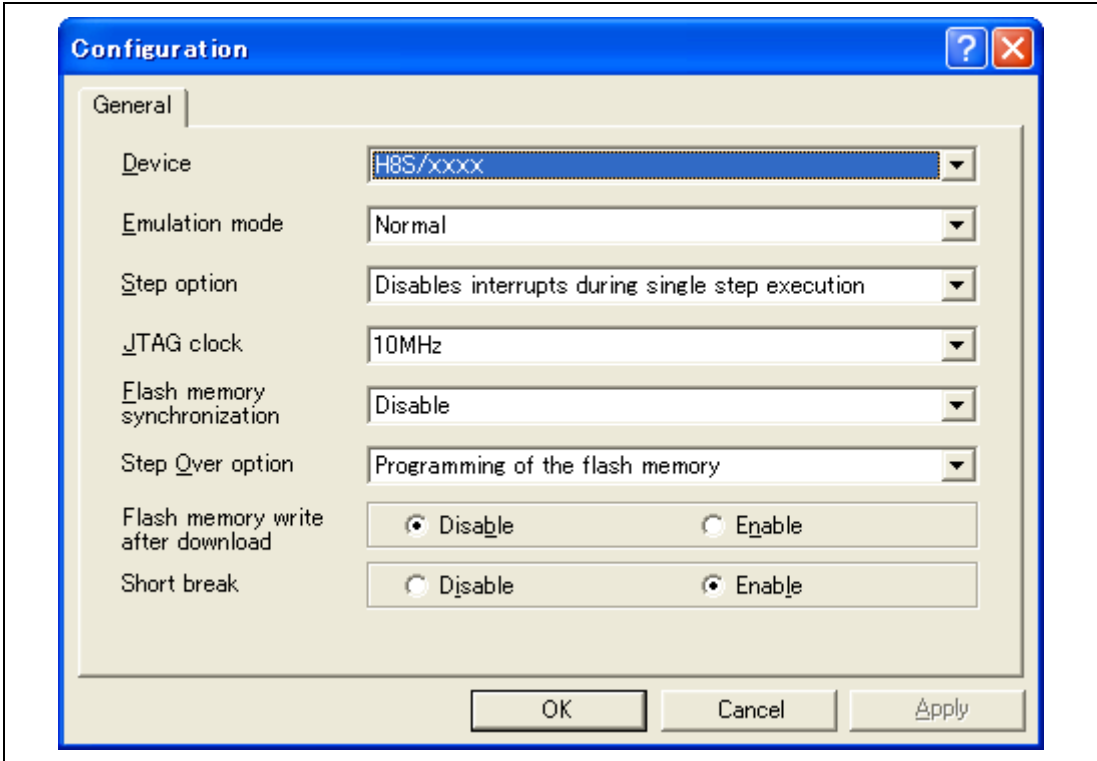
Figure 6.1   [Configuration] Dialog Box

Note: The items that can be set in this dialog box differ depending on the product. For details on
        the settings for each product, refer to the online help.

Set options as follows:

Table 6.1   [General] Page Options

Option	Value (Default)
Device	H8S/xxxx
Emulation mode	Normal (normal execution)
Step option	Disables interrupts during single step execution
JTAG clock	Sets the frequency* of the JTAG clock.
Flash memory synchronization when the user program is stopped	Disable
Step Over option	Executes the step over and step out by using the breakpoint.
Flash memory write after download	Disable
Short break	Enable

Note: The range of frequency for the JTAG clock differs depending on the supported MCU.

Click the [OK] button to set a configuration.

6.5   Checking the Operation of RAM

Check that RAM is operating correctly.

Display and edit the contents of the memory in the [Memory] window to check that the memory is operating correctly.

Note: The memory can be installed on the board in some MCUs. In this case, however, the above way of checking the operation of memory may be inadequate. It is recommended that a program for checking the memory be created.

- Select [Memory…] from the [CPU] submenu of the [View] menu and enter the RAM address in the [Display Address] edit box. Here, enter *H'FFC000*. Then enter *H'000000* and *H'FFFFFF* in the [Scroll Start Address] and [Scroll End Address] edit boxes, respectively.
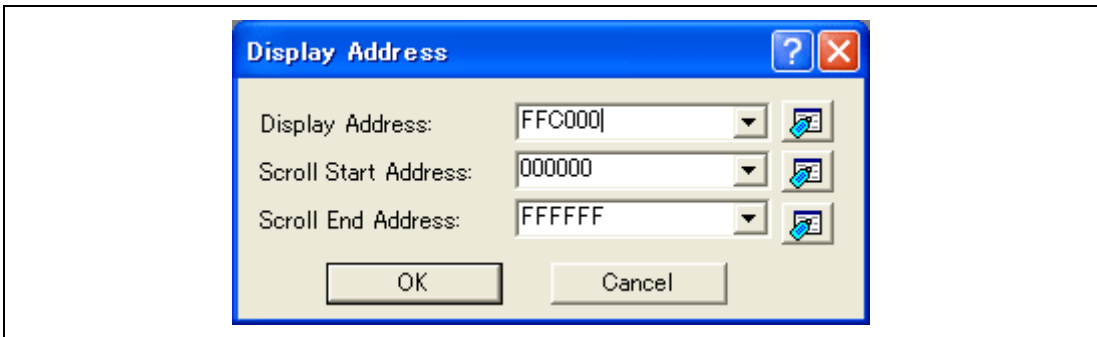
Figure 6.2   [Display Address] Dialog Box

Note: The settings of the RAM area differ depending on the product. For details, refer to the hardware manual.

- Click the [OK] button. The [Memory] window is displayed and shows the specified memory area.
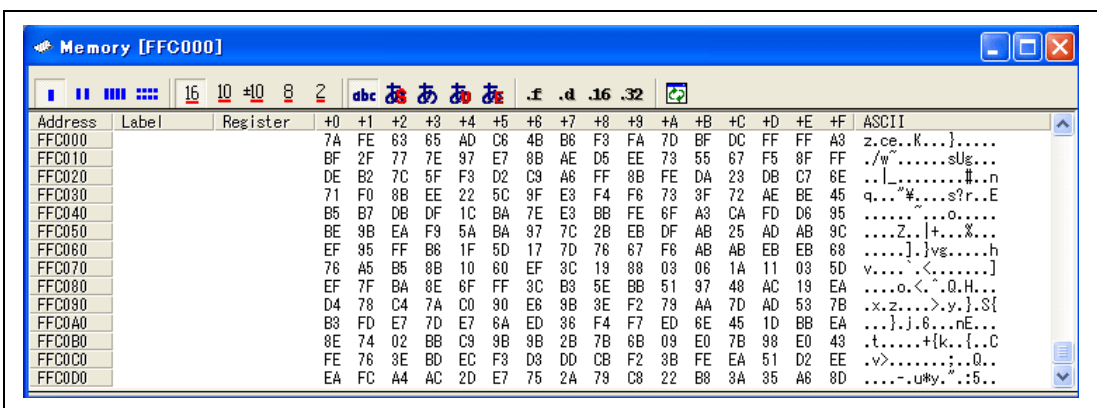
Figure 6.3   [Memory] Window

- Placing the mouse cursor on a point in the display of data in the [Memory] window and double-clicking allows the values at that point to be changed. Data can also be directly edited around the current position of the text cursor.

6.6   Downloading the Tutorial Program

6.6.1   Downloading the Tutorial Program

Download the object program to be debugged.

In this emulator, it is enabled to download the program and set the PC breakpoint in the internal ROM area. For the method to set the PC breakpoint, refer to section 6.16.1, PC Break Function.

- Select [Download module] by right-clicking on "Tutorial.abs" under "Download modules".
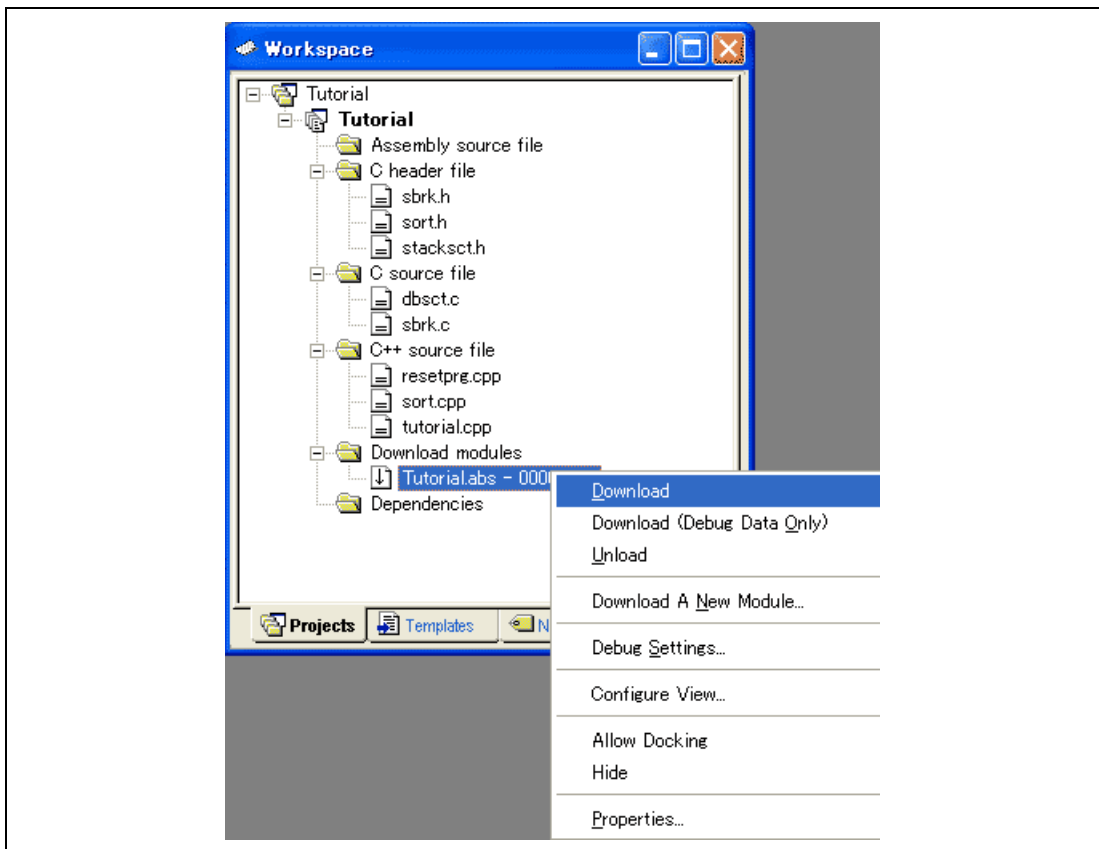
Figure 6.4   Downloading the Tutorial Program

6.6.2   Displaying the Source Program

The High-performance Embedded Workshop allows the user to debug a user program at the source level.

- Double-click [tutorial.cpp] under [C++ source file].
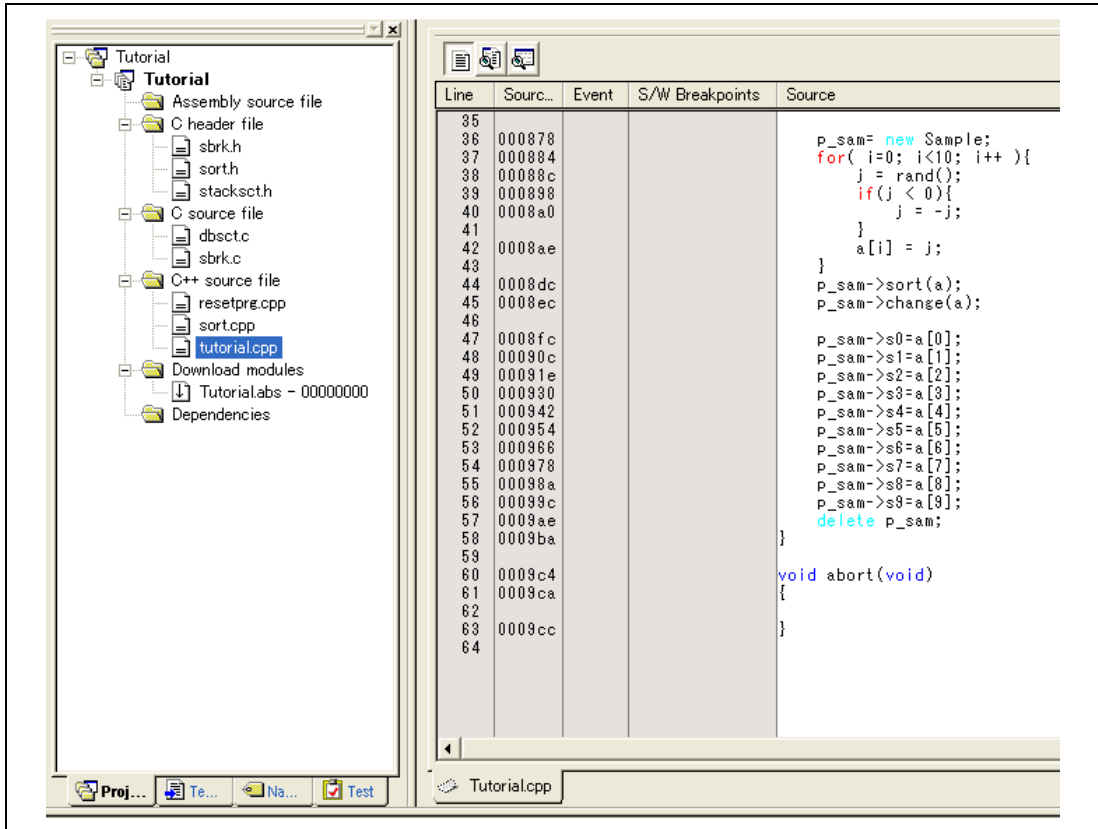
Figure 6.5   [Editor] Window (Displaying the Source Program)

- Select the [Display] option from the [Setup] menu to set a font and size that are legible, if necessary.

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

6.7   Setting a PC Breakpoint

A PC breakpoint is a simple debugging function.

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

- Select by double-clicking the [S/W breakpoint] column on the line containing the sort function call.
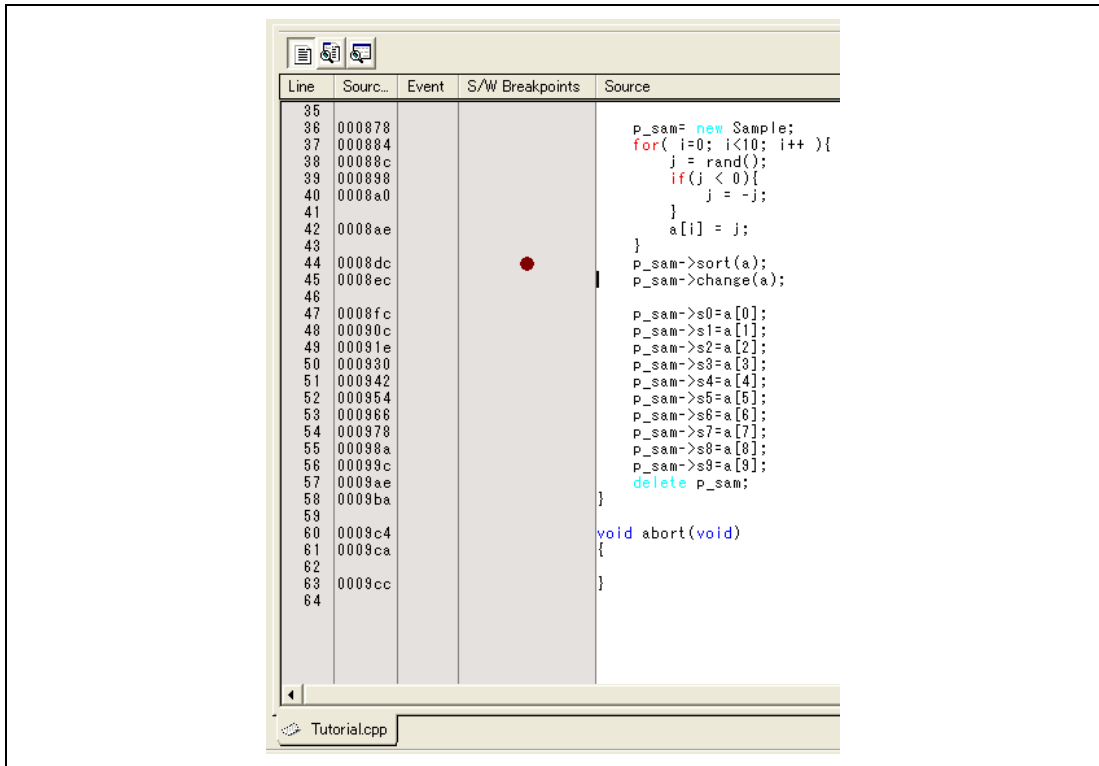
Figure 6.6   [Editor] Window (Setting a PC Breakpoint)

The symbol ● will appear on the line containing the sort function. This shows that a PC breakpoint has been set.

6.8   Setting Registers

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

- Select [Registers] from the [CPU] submenu of the [View] menu. The [Register] window is displayed.
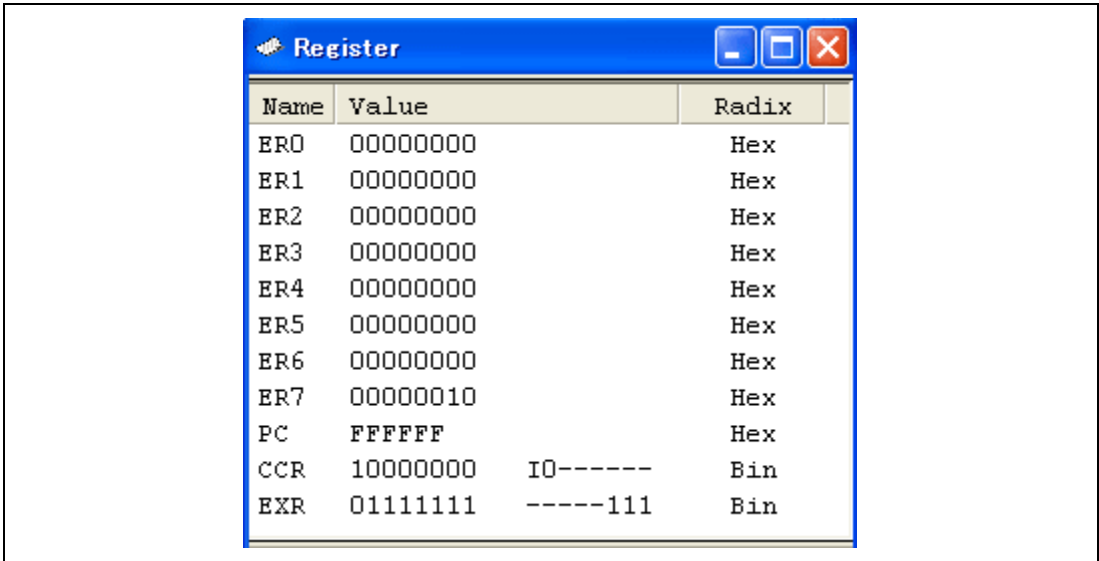
Figure 6.7   [Register] Window

To change the value of the program counter (PC), double-click the value area in the [Register] window with the mouse. The following dialog box is then displayed, and the value can be changed. Set the program counter to H'400 in this tutorial program, and click the [OK] button.
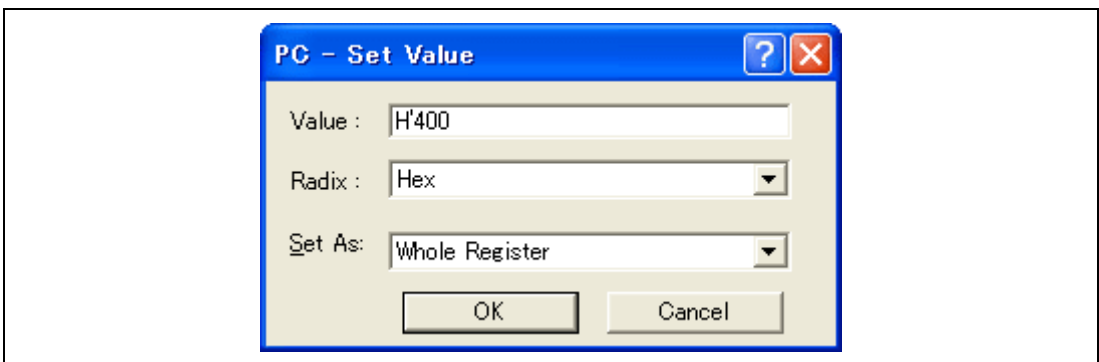
Figure 6.8   [Register] Dialog Box (PC)

6.9   Executing the Program

Execute the program as described in the following:

- To execute the program, select [Go] from the [Debug] menu, or click the [Go] button on the toolbar.
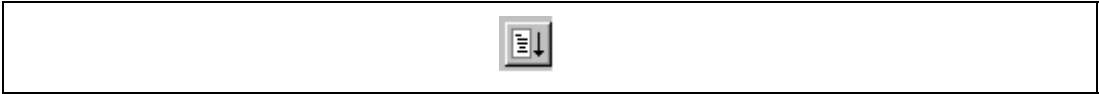
Figure 6.9   [Go] Button

When the program execution is started, '**RUNNING' is displayed on the status bar.

The program will be executed up to the breakpoint that has been set, and an arrow will be displayed in the [S/W breakpoint] column to show the position that the program has halted, with the message [BREAKPOINT] in the status bar.

Note: When the source file is displayed after a break, a path of the source file may be inquired. The location of the source file is as follows:
<Drive where the OS has been installed>:
\WorkSpace\Tutorial\E10A-USB\H8S\xxxx\Tutorial

Here, 'xxxx' means the target product group.
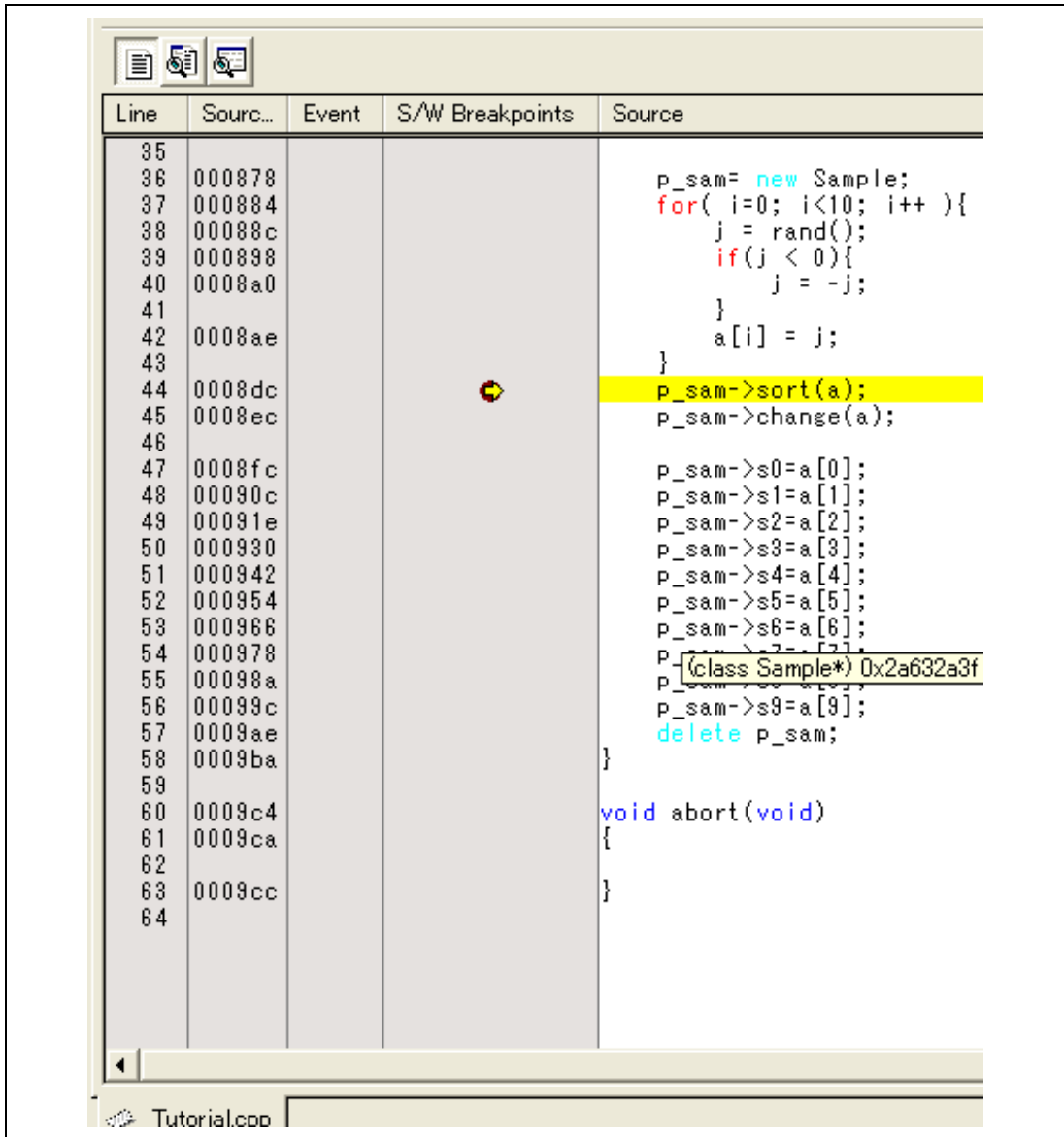
Figure 6.10   [Editor] Window (Break State)

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

- Select [Status] from the [CPU] submenu of the [View] menu. After the [Status] window is displayed, open the [Platform] sheet, and check the `Status` of `Cause of last break`.
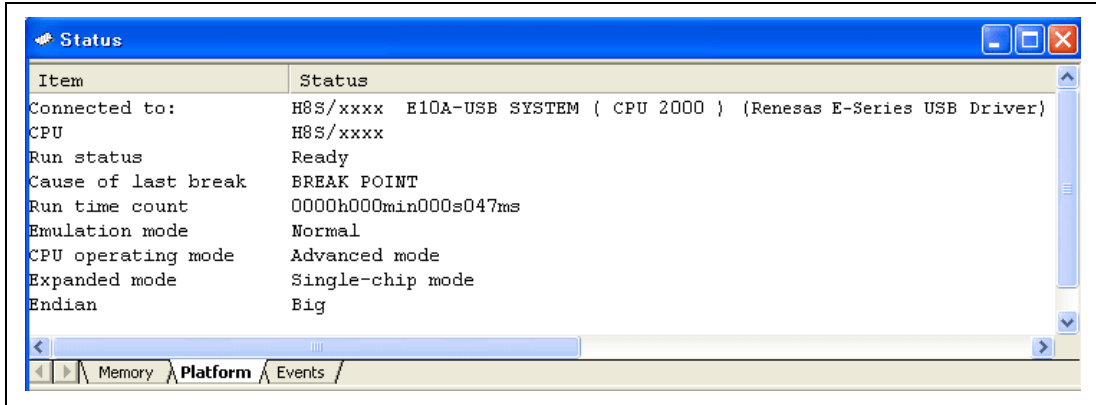
Figure 6.11   [Status] Window

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

6.10   Reviewing Breakpoints

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

- Select [Eventpoints] from the [Code] submenu of the [View] menu. The [Event] window is displayed. Select the [Breakpoint] sheet.
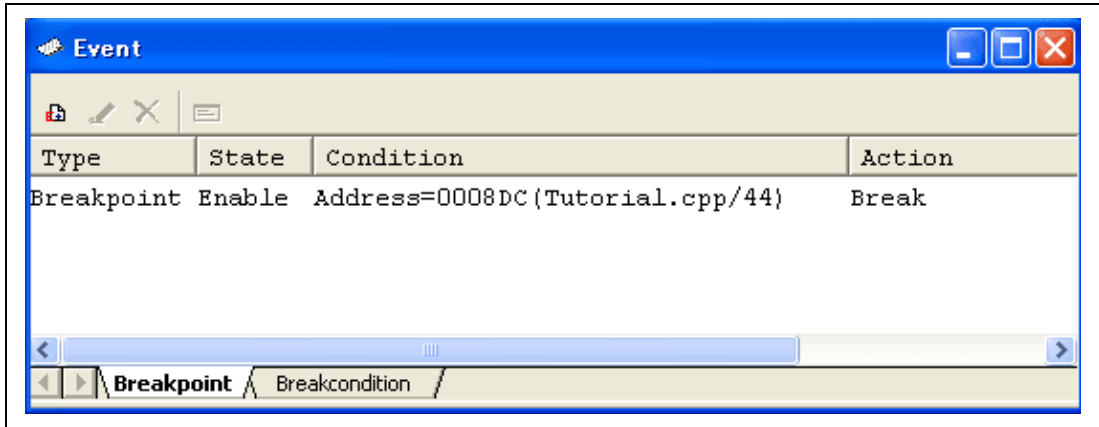
Figure 6.12   [Event] Window

The popup menu, opened by clicking the [Event] window with the right-hand mouse button, allows the user to set or change breakpoints, add new breakpoints, and delete, enable, or disable breakpoints.

6.11   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 `_main` in word size:

- Select [Memory…] from the [CPU] submenu of the [View] menu and enter the RAM address in the [Display Address] edit box. Here, enter *H'FFC000*. Then enter *H'000000* and *H'FFFFFF* in the [Scroll Start Address] and [Scroll End Address] edit boxes, respectively.
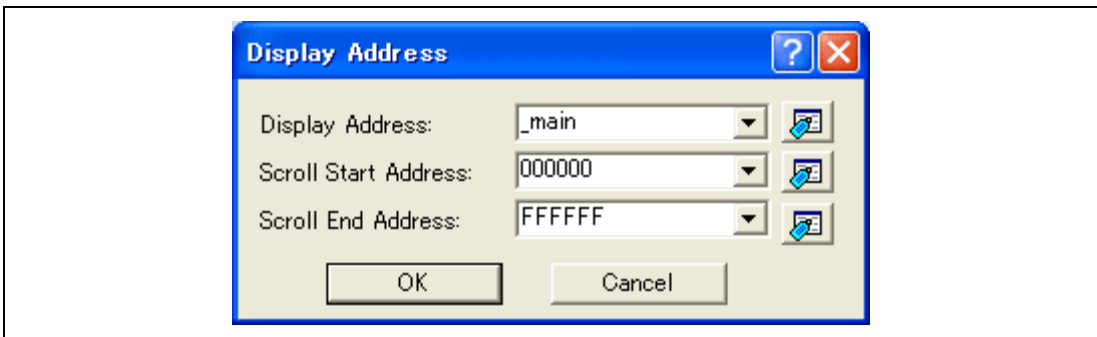
Figure 6.13   [Display Address] Dialog Box

- Click the [OK] button. The [Memory] window showing the specified area of memory is displayed.
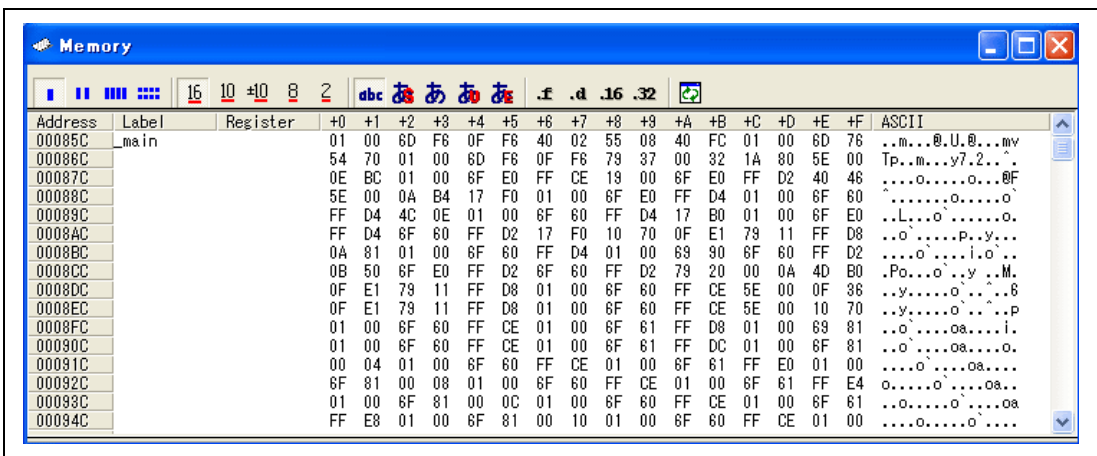
Figure 6.14   [Memory] Window

6.12   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 [Editor] window to position the cursor.
- Select [Instant Watch...] with the right-hand mouse button.

The following dialog box will be displayed.
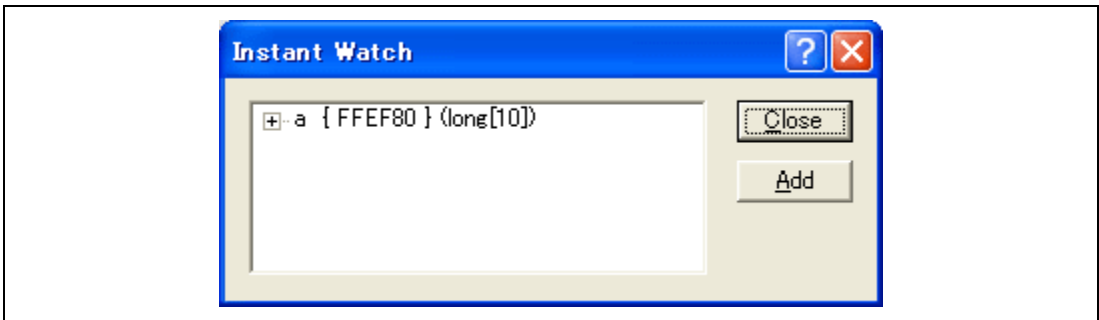
Figure 6.15   [Instant Watch] Dialog Box

- Click the [Add] button to add a variable to the [Watch] window.
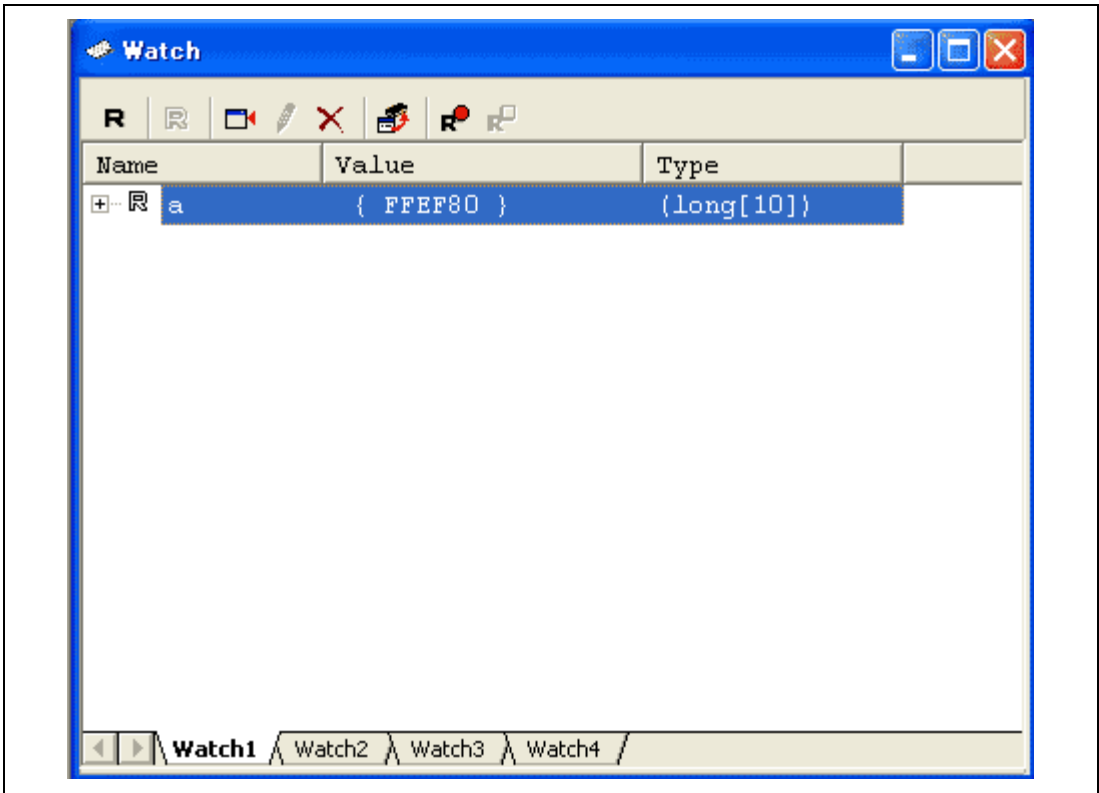
Figure 6.16   [Watch] Window (Displaying the Array)

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

- Click the [Watch] window with the right-hand mouse button and select [Add Watch…] from the popup menu.

The following dialog box will be displayed. Enter variable i.
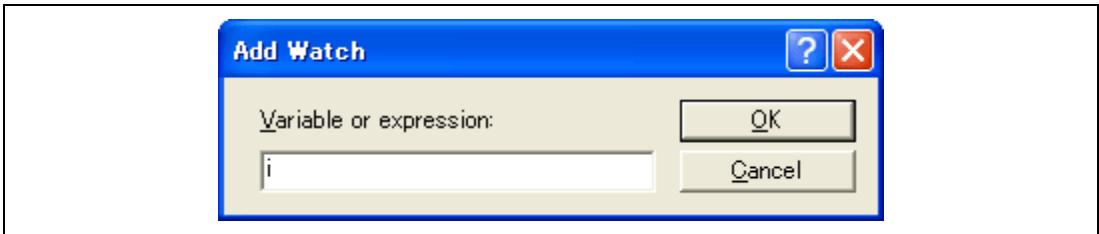
Figure 6.17   [Add Watch] Dialog Box

- Click the [OK] button.

The [Watch] window will now also show the int-type variable i.
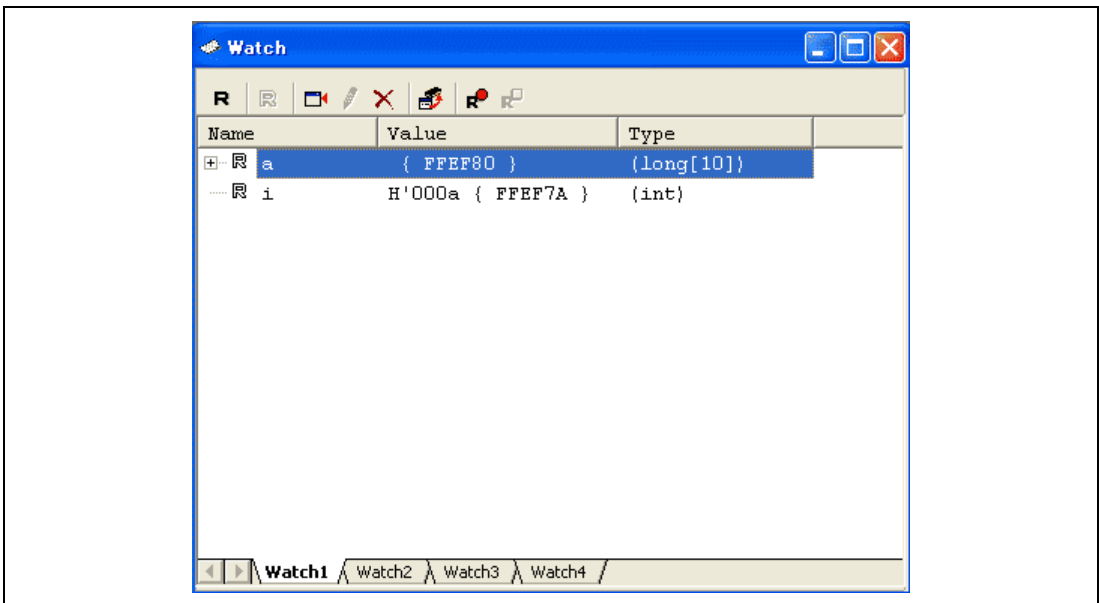
Figure 6.18   [Watch] Window (Displaying the Variable)

The user can click mark '+' at the left side of array a in the [Watch] window to watch all the elements.
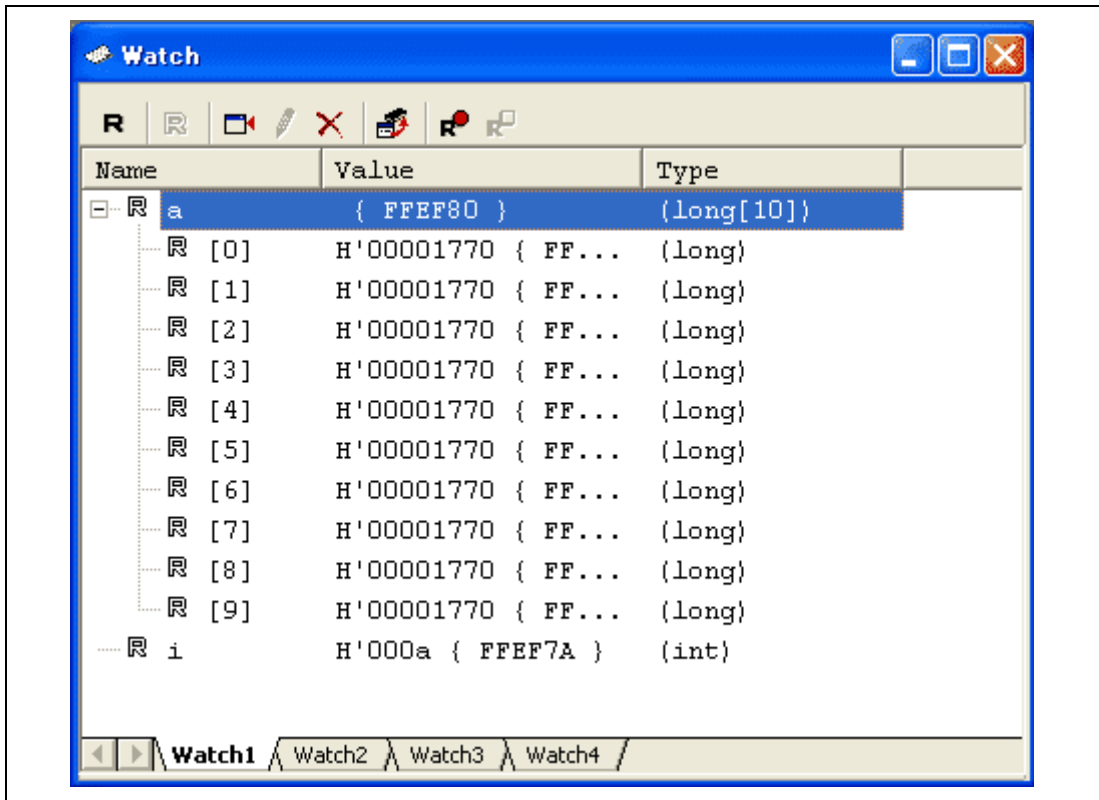
Figure 6.19   [Watch] Window (Displaying Array Elements)

6.13   Stepping Through a Program

The High-performance Embedded Workshop provides a range of step menu commands that allow efficient program debugging.

Table 6.2 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.

6.13.1   Executing [Step In] Command

The [Step In] command 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 [Debug] menu, or click the [Step In] button on the toolbar.
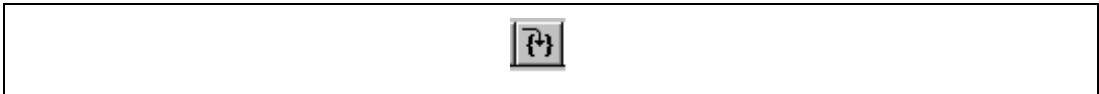

Figure 6.20   [Step In] Button
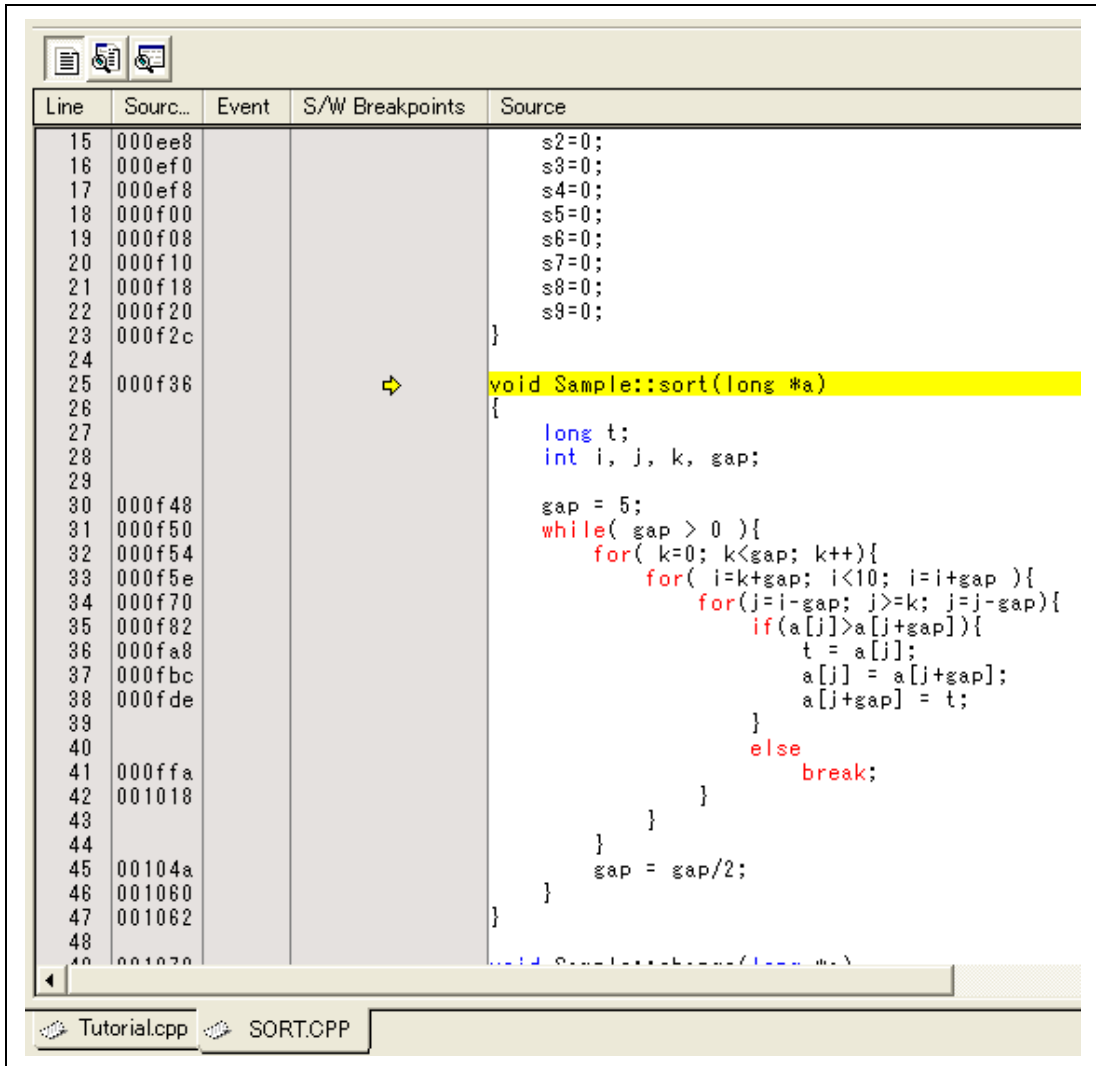
Figure 6.21   [Editor] Window (Step In)

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

6.13.2   Executing [Step Out] Command

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

• To step out of the `sort` function, select [Step Out] from the [Debug] menu, or click the [Step Out] button on the toolbar.

Note: It takes time to execute this function. When the calling source is clarified, use [Go To Cursor].

Figure 6.22   [Step Out] Button
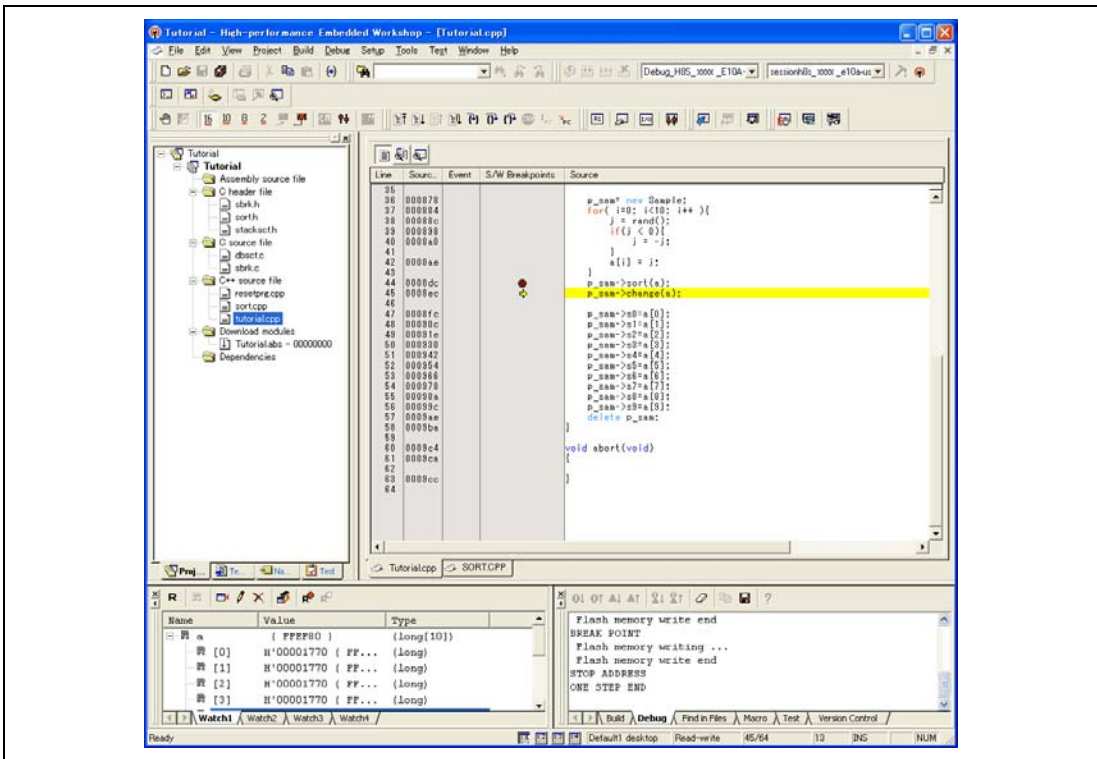
Figure 6.23   [High-performance Embedded Workshop] Window (Step Out)

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

6.13.3 Executing [Step Over] Command

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

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

Figure 6.24 [Step Over] Button
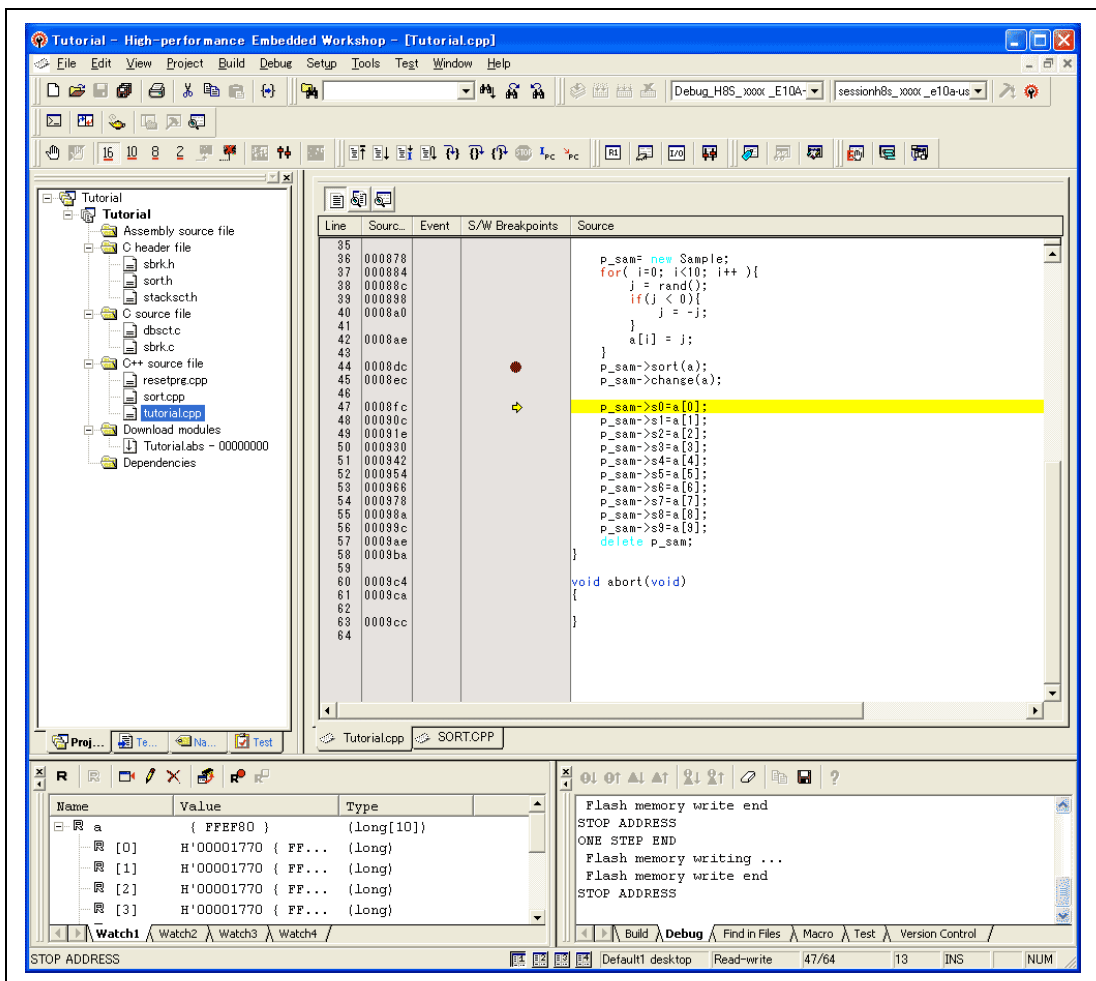
Figure 6.25 [High-performance Embedded Workshop] Window (Step Over)

6.14   Forced Breaking of Program Executions

The High-performance Embedded Workshop can force a break in the execution of a program.

- Cancel all breakpoints settings.
- To execute the remaining sections of the `main` function, select [Go] from the [Debug] menu or the [Go] button on the toolbar.
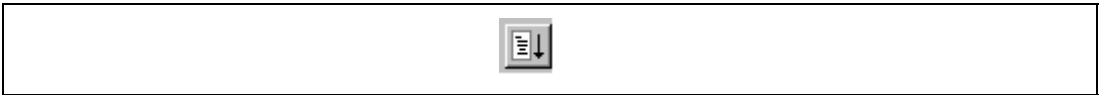
Figure 6.26   [Go] Button

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

Figure 6.27   [Stop] Button

6.15   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 four local variables: a, j, i, and p_sam.

- Select [Locals] from the [Symbol] submenu of the [View] menu. The [Locals] window is displayed.

The [Locals] window shows the local variables in the function currently pointed to by the program counter, along with their values. Note, however, that the [Locals] window is initially empty because local variables are yet to be declared.
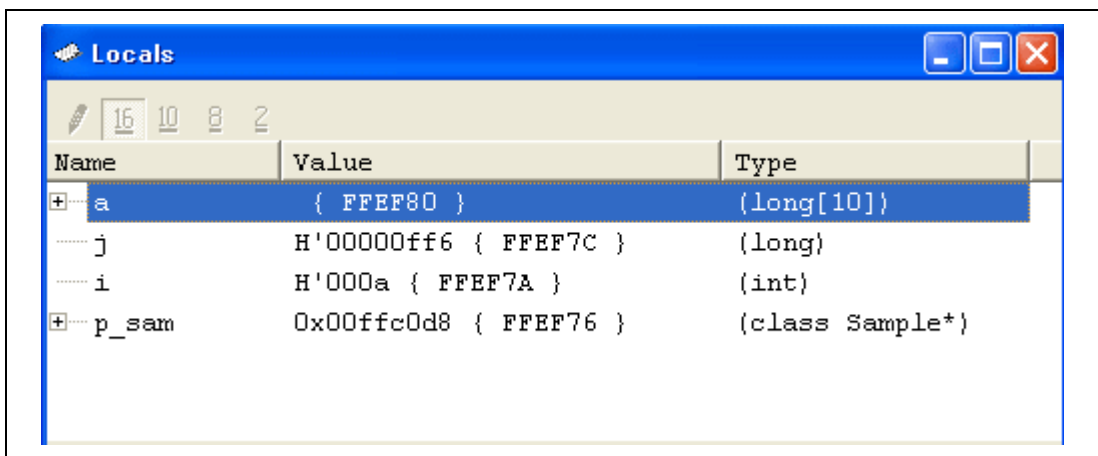
Figure 6.28   [Locals] Window

- Click mark '+' at the left side of array a in the [Locals] window to display the elements.
- When the user refers to the elements of array a before and after the execution of the sort function, it is clarified that random data is sorted in descending order.

6.16   Break Function

The emulator has PC and hardware break functions. With the High-performance Embedded Workshop, a PC breakpoint can be set using the [Breakpoint] sheet of the [Event] window, and a hardware break condition can be set using the [Breakcondition] sheet.

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

6.16.1   PC Break Function

The emulator can set up to 255 PC breakpoints. Other methods for setting a PC breakpoint than in section 6.7, Setting a PC Breakpoint, are described below.

• Select [Eventpoints] from the [Code] submenu of the [View] menu. The [Event] window is displayed.

• Select the [Breakpoint] sheet.
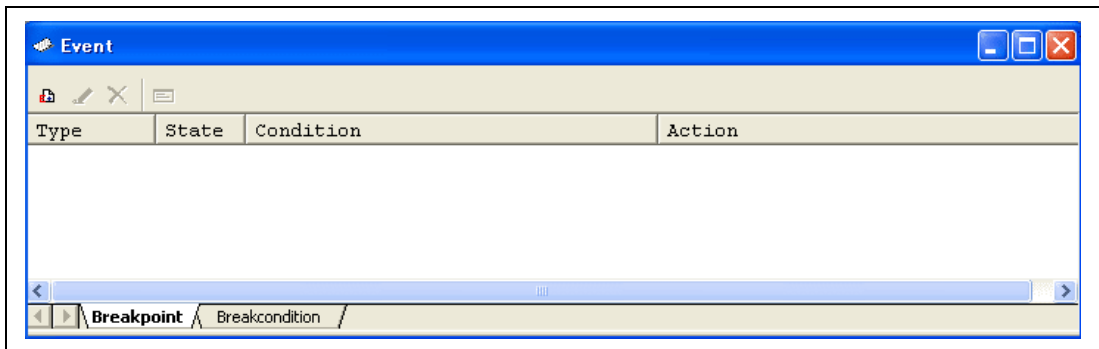
Figure 6.29   [Event] Window (Before Setting a PC Breakpoint)

- Double-click the [S/W breakpoint] column on the [Editor] window. A PC breakpoint is set.
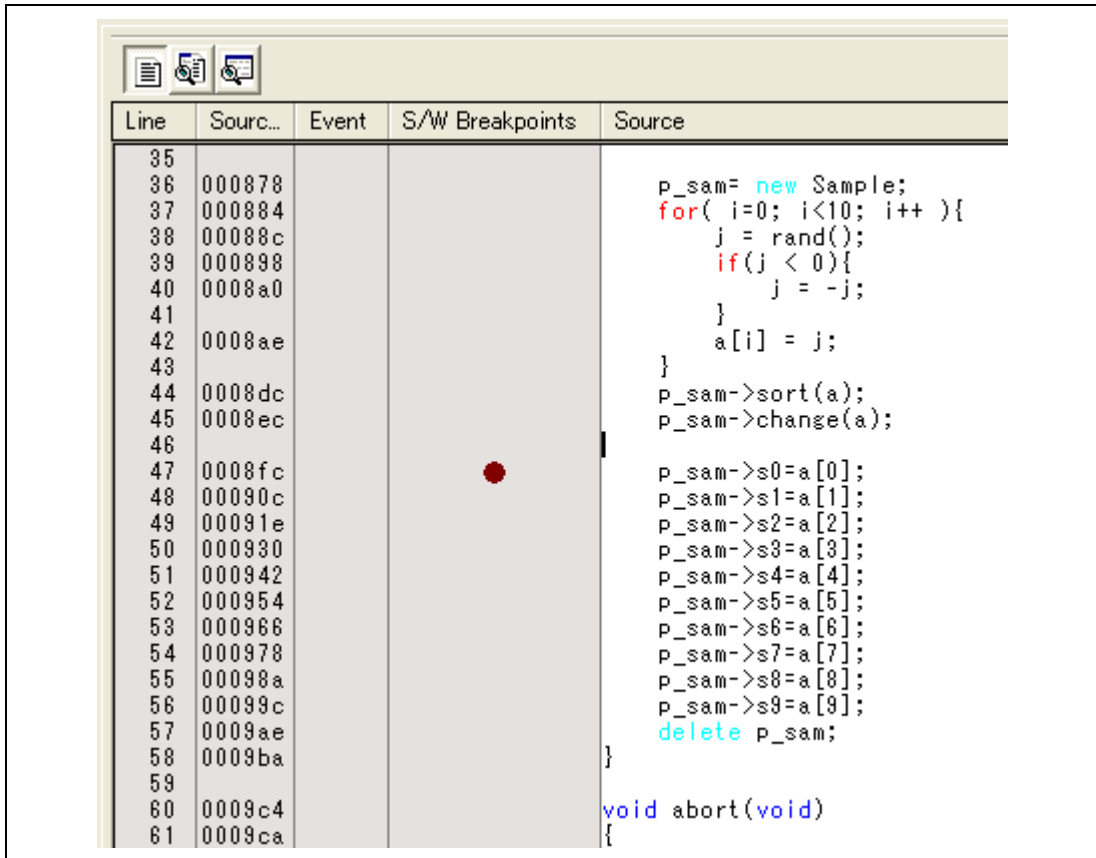
Figure 6.30   [Editor] Window (Setting a PC Breakpoint)

- When the PC breakpoint is set in the [Event] window, click the [Event] window with the right-hand mouse button and select [Add…] from the popup menu.

Figure 6.31   [Set Break] Dialog Box

- Click the [OK] button.

Note: The address values described in this section will differ depending on the tutorial for each product.

The PC breakpoint that has been set is displayed in the [Event] window.
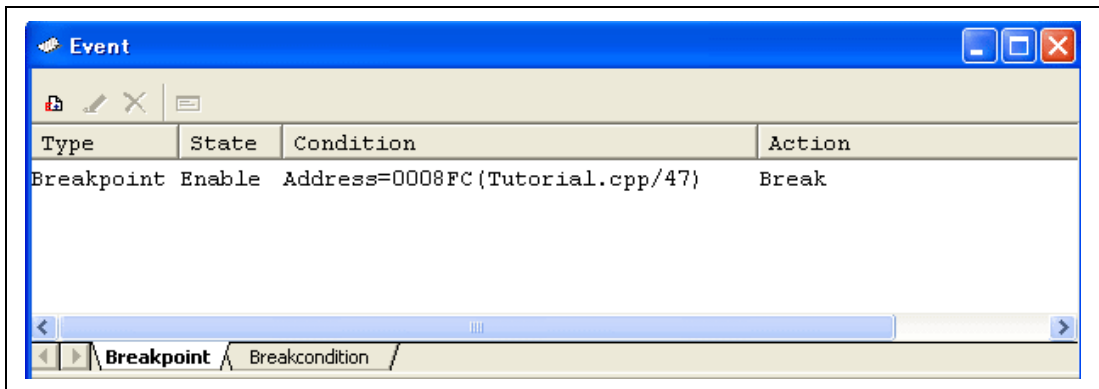
Figure 6.32   [Event] Window (PC Breakpoint Setting)

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.

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

- Close the [Event] window.
- Set the program counter value (PC = H'400) that was set in section 6.8, Setting Registers, in
  the [Register] window. Click the [Go] button.
- If program execution is failed, reset the device and execute again the procedures above.

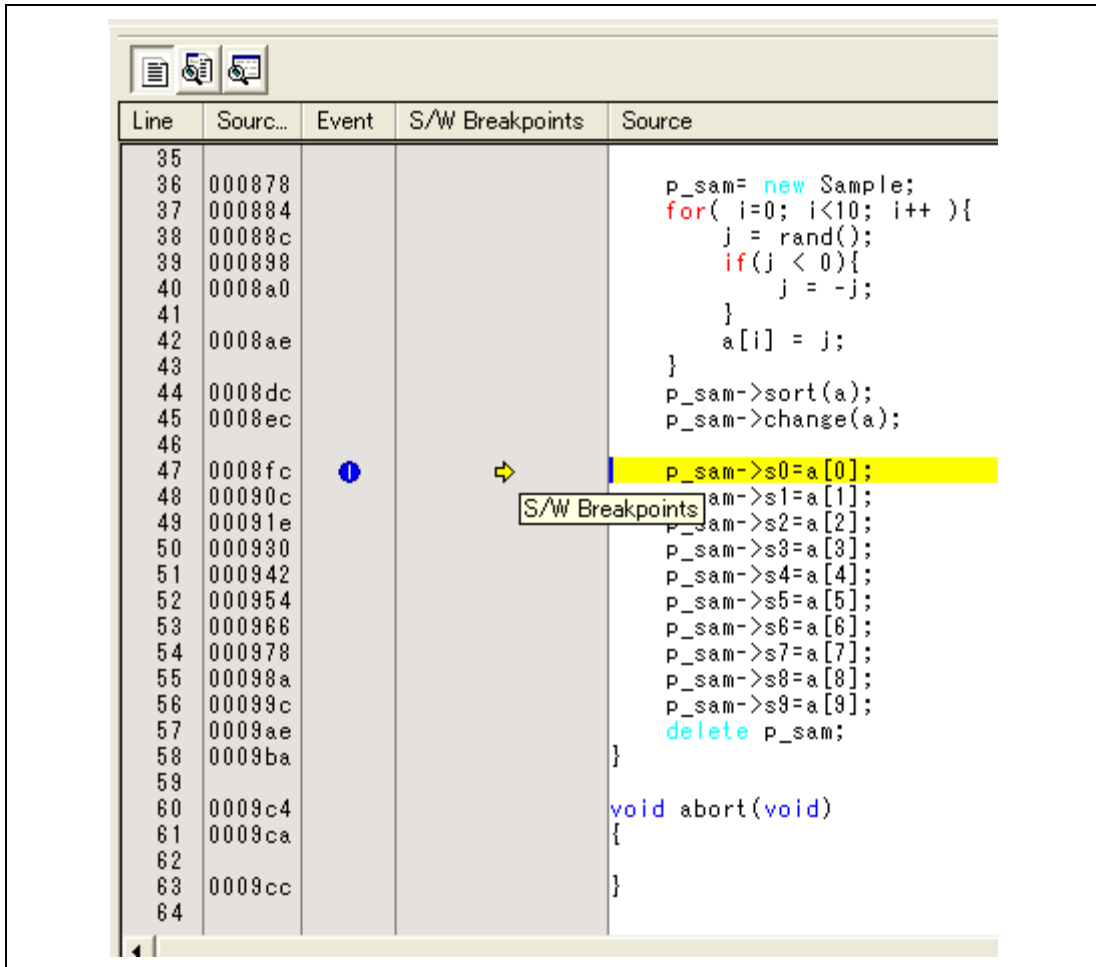The program runs, and stops at the set PC breakpoint.
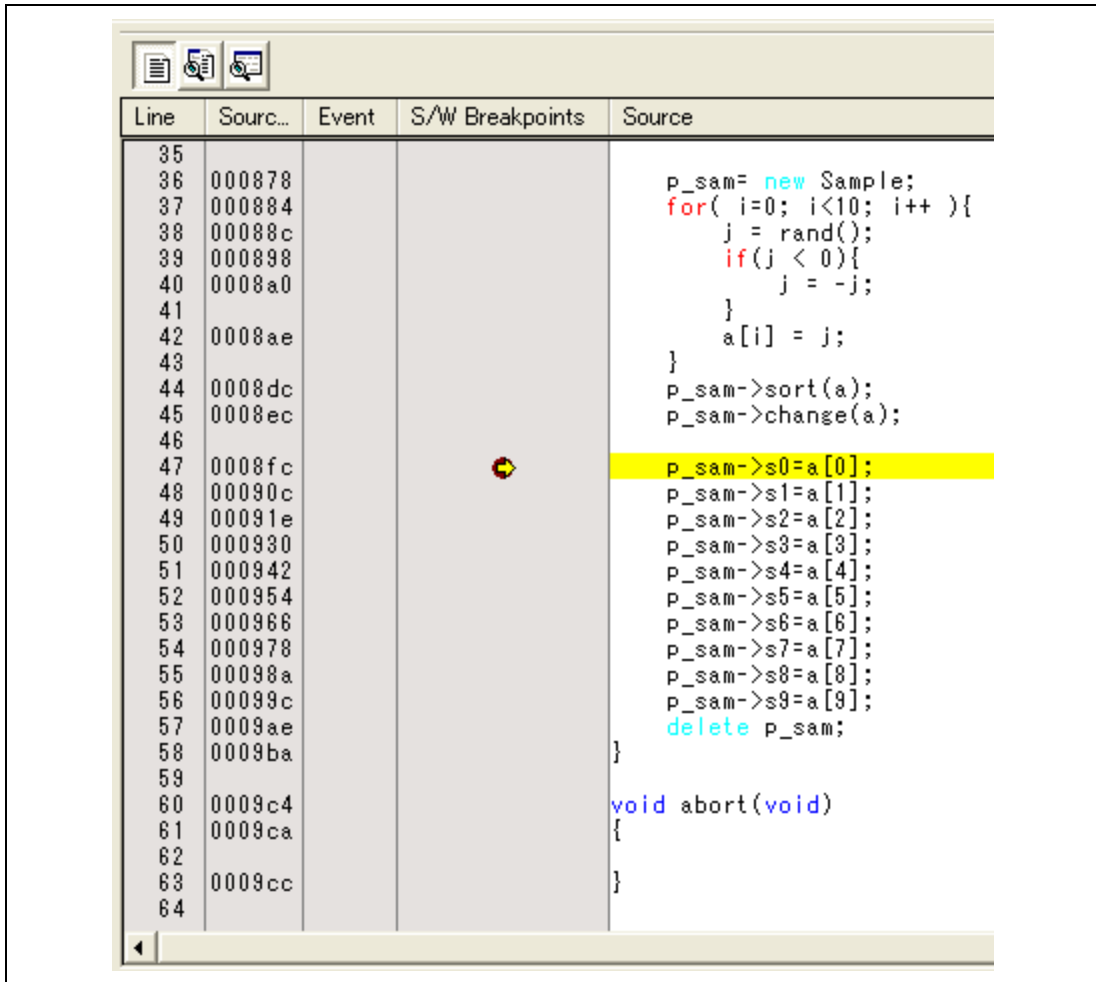
Figure 6.33   [Editor] Window at Execution Stop (PC Break)
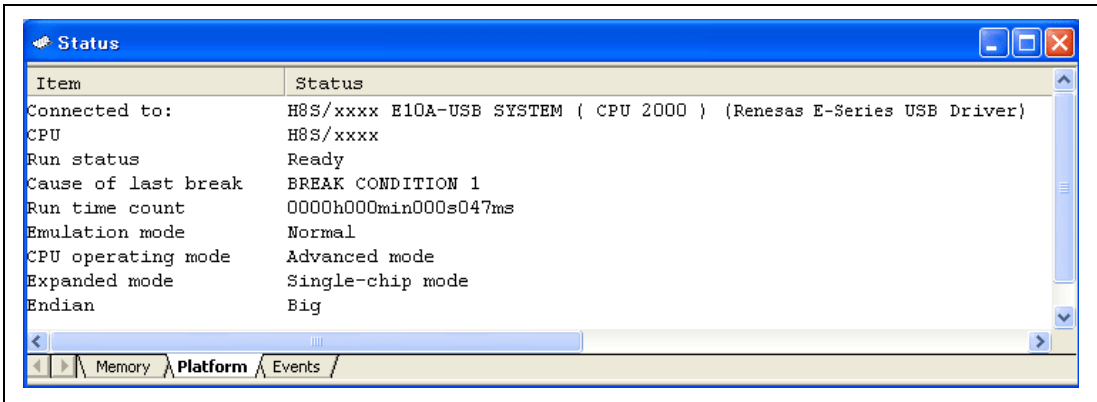
The [Status] window displays the following contents.
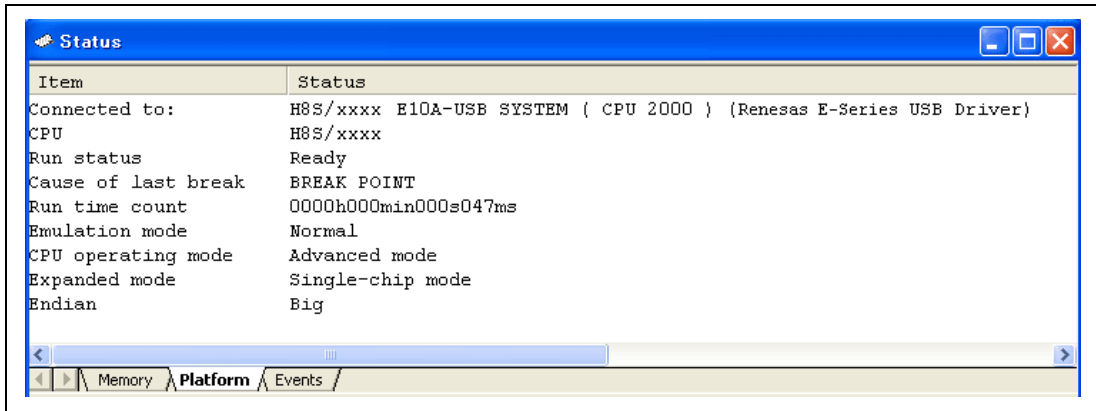
Figure 6.34   Displayed Contents of the [Status] Window (PC Break)

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

6.17   Hardware Break Function

A method is given below in which the address bus condition is set under Break Condition 1 as hardware break conditions.

- Select [Eventpoints] from the [Code] submenu of the [View] menu. The [Event] window is displayed.
- The PC breakpoint that has been previously set is deleted. Click the [Event] window with the right-hand mouse button and select [Delete All] from the popup menu to cancel all PC breakpoints that have been set.
- Double-click the [Event] column in the [Editor] window. The specified address shows 🔵 and a Break Condition is set.
- Click the [Break condition] tab.
- Select a line of Break condition 1 in the [Event] window. When highlighted, double-click this line.
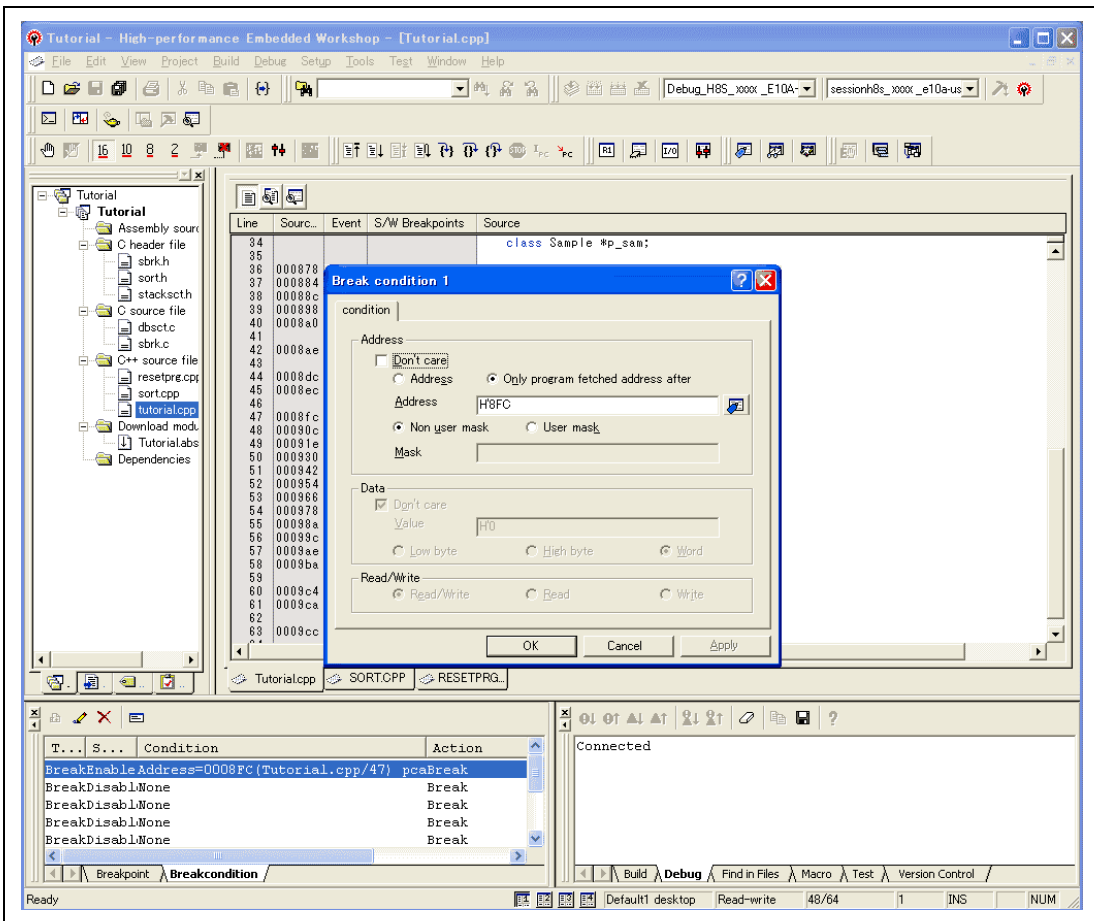
Figure 6.35   [High-performance Embedded Workshop] Window ([Break condition 1])

- The [Break condition 1] dialog box is displayed.
- The data condition can be set besides the address condition.
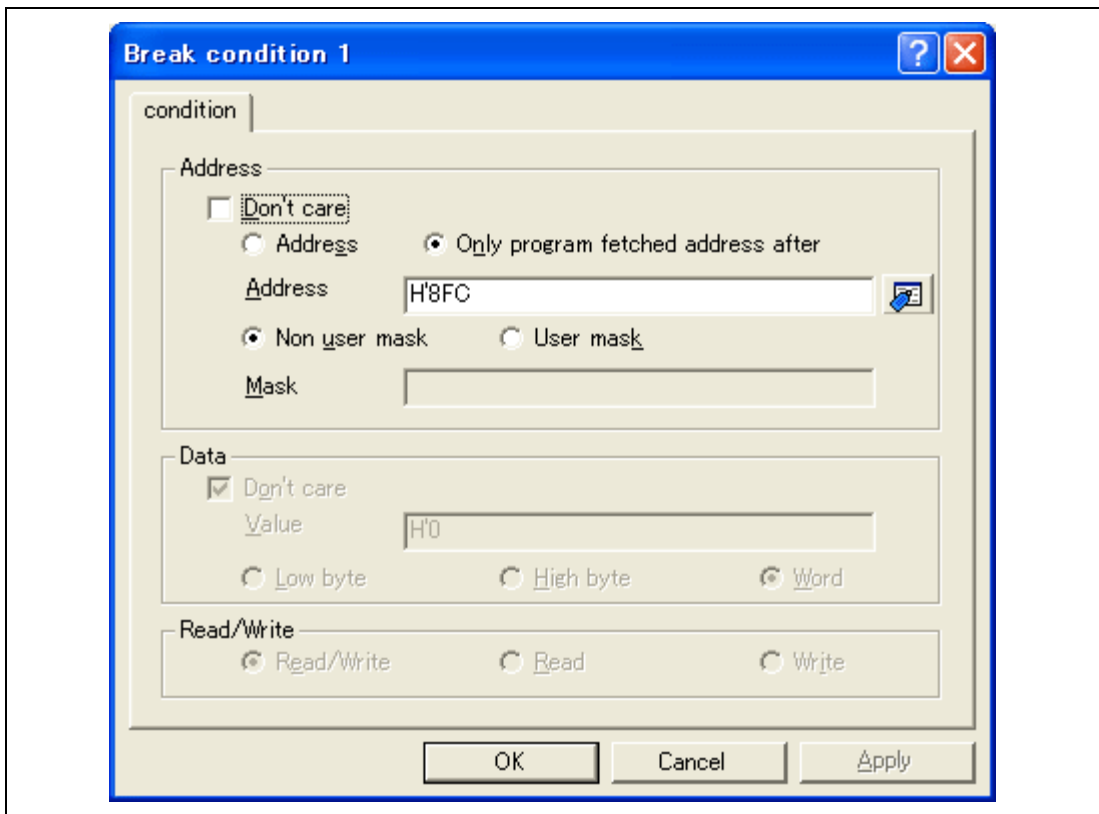
Figure 6.36 [condition] Page ([Break condition 1] Dialog Box)

- Click the [OK] button.
- The content that has been set is reflected in the [Event] window.
- The first point display in the State line changes from Disable to Enable.
- The first point display in the Condition line changes from None to Address = H'00008FC (Tutorial.cpp/47) pcafter.
- Set the program counter value (PC = H'400) that was set in section 6.8, Setting Registers, in the [Register] window. Click the [Go] button.

Note: The items that can be set in this page differ depending on the product. For details on the settings for each product, refer to the online help.
The address values described in this section will differ depending on the tutorial for each product.

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

Figure 6.37   [Editor] Window at Execution Stop (Break Condition 1)

The [Status] window displays the following contents.

Figure 6.38   Displayed Contents of the [Status] Window (Break Condition 1)

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

6.18   Trace Functions

The emulator has a branch-instruction trace function.
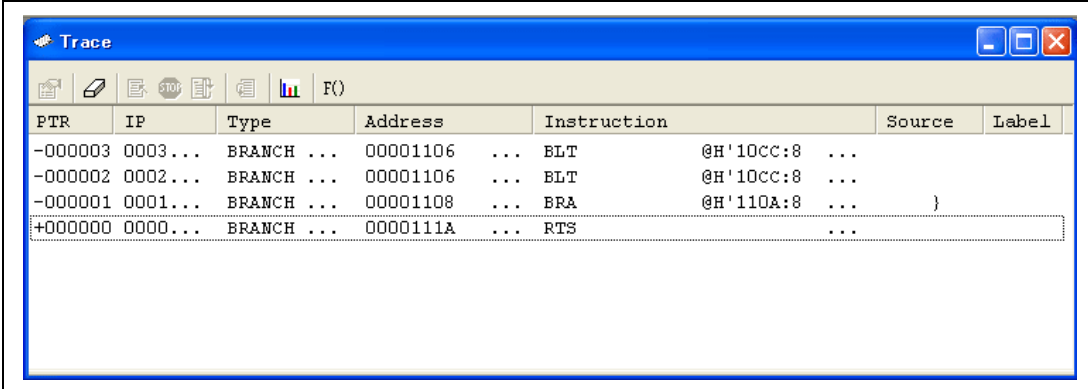
In H8S family, the four-channel branch source addresses, mnemonics, operands, source lines, and labels are displayed.

In H8SX family, the eight-channel branch source or destination addresses, mnemonics, operands, source lines, and labels are displayed.

6.18.1   Displaying the Trace Window

Select [Trace] from the [Code] submenu of the [View] menu.

Run the program as shown in the example of section 6.16.1, PC Break Function. The trace results are displayed in the [Trace] window after the program execution is completed.

Figure 6.39   [Trace] Window

If necessary, adjust the column widths by dragging borders in the header bar (immediately below the title bar).

Note: The number of branch instructions that can be acquired by a trace and the trace display differ depending on the product. For details on the specifications of each product, refer to the online help.

6.19   Stack Trace Function

The emulator uses the information on the stack to display the names of functions in the sequence of calls that led to the function to which the program counter is currently pointing.

Note: This function can be used only when the load module that has the Elf/Dwarf2-type debugging information is loaded. Such load modules are supported in H8C/C++ compiler V3.0 or later.

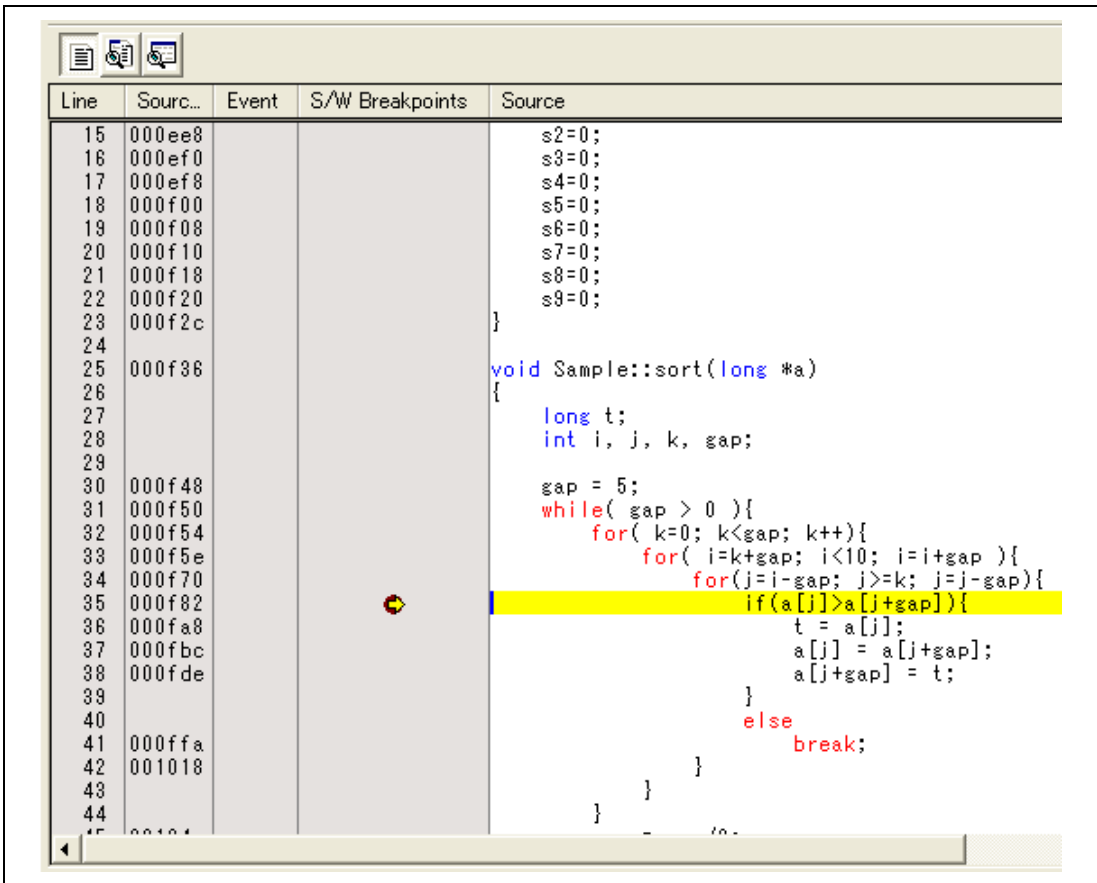• Double-click the [S/W breakpoint] column in the sort function and set a PC breakpoint.

Figure 6.40   [Editor] Window (PC Breakpoint Setting)

- Set the same program counter value (PC = H'400) as was set in section 6.8, Setting Registers (again, use the [Register] window). Click the [Go] button.
- After the break in program execution, select [Stack Trace] from the [Code] submenu of the [View] menu to open the [Stack Trace] window.
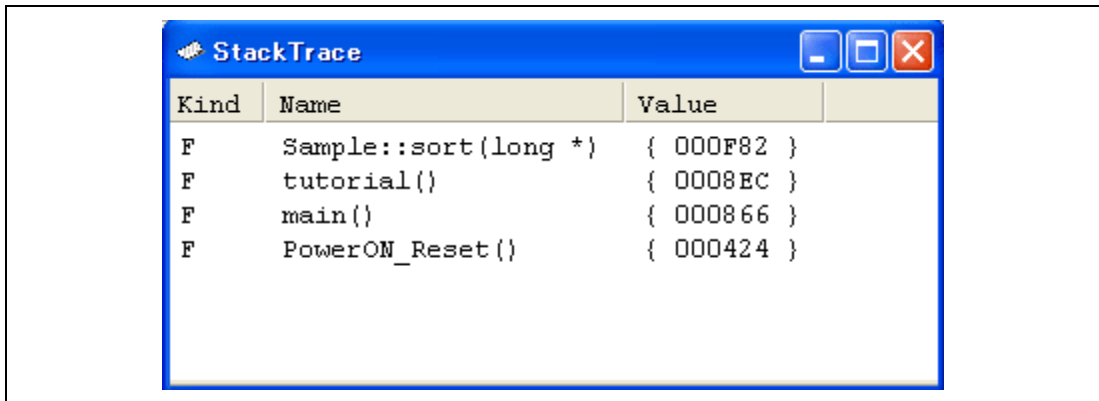
Figure 6.41   [Stack Trace] Window

Figure 6.41 shows that the position of the program counter is currently at the selected line of the sort() function, and that the sort() function is called from the tutorial() function.

6.20   What Next?

This tutorial has described the major features of the emulator and the use of the High-performance Embedded Workshop.

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.

Section 7   Maintenance and Guarantee

This section describes maintenance, guarantee, repair provisions, and how to request for repair of the emulator.

7.1   User Registration

When you purchase our product, be sure to register as a user. For user registration, refer to the section of 'User Registration' (p. iii) of this user's manual.

7.2   Maintenance

1. If dust or dirt collects on any equipment of this product, wipe the board dry with a soft cloth. Do not use thinner or other solvents because these chemicals can cause the equipment's surface coating to separate.
2. When you do not use this product for a long period, for safety purposes, disconnect the power cable from the power supply.

7.3   Guarantee

If your product becomes faulty within one year after its purchase while being used under good conditions by observing 'IMPORTANT INFORMATION' described in this user's manual, we will repair or replace your faulty product free of charge. Note, however, that if your product's fault is raised by any one of the following causes, we will repair it or replace it with new one with extra-charge:

*   Misuse, abuse, or use under extraordinary conditions
*   Unauthorized repair, remodeling, maintenance, and so on
*   Inadequate user's system or misuse of it
*   Fires, earthquakes, and other unexpected disasters

In the above cases, contact your local distributor. If your product is being leased, consult the leasing company or the owner.

RENESAS

7.4   Repair Provisions

7.4.1   Repair with Extra-Charge

The products elapsed more than one year after purchase can be repaired with extra-charge.

7.4.2   Replacement with Extra-Charge

If your product's fault falls in any of the following categories, the fault will be corrected by replacing the entire product instead of repair, or you will be advised to purchase new one, depending on the severity of the fault.

- Faulty or broken mechanical parts
- Flaw, separation, or rust in coated or plated parts
- Flaw or cracks in plastic parts
- Faults or breakage caused by improper use or unauthorized repair or modification
- Heavily damaged electric circuits due to overvoltage, overcurrent or shorting of power supply
- Cracks in the printed circuit board or burnt-down patterns
- Wide range of faults that makes replacement less expensive than repair
- Unlocatable or unidentified faults

7.4.3   Expiration of the Repair Period

When a period of one year elapses after the model was dropped from production, repairing products of the model may become impossible.

7.4.4   Transportation Fees at Sending Your Product for Repair

Send your product to us for repair at your expense.

7.5   How to Make a Request for Repair

If your product is found faulty, follow the procedure below to send your product for repair.

Fill in the Repair Request Sheet included with this product, then send it along with this product for repair to your local distributor. Make sure that information in the Repair Request Sheet is written in as much detail as possible to facilitate repair.

⚠ CAUTION

Note on Transporting the Product:

When sending your product for repair, use the packing box and cushion material supplied with this product when delivered to you and specify handling caution for it to be handled as precision equipment. If packing of your product is not complete, it may be damaged during transportation. When you pack your product in a bag, make sure to use conductive polyvinyl supplied with this product (usually a blue bag). When you use other bags, they may cause a trouble on your product because of static electricity.

Appendix A   Troubleshooting

1. *I have a text file open in the editor but syntactic color-coding is not being displayed.*

    Ensure that you have named the file (i.e. saved it) and that the "Syntax coloring" check box is set on the "Editor" tab of the "Options" dialog box, which is launched via **[Setup -> Options...]**. The High-performance Embedded Workshop looks up the filename extension to determine the group to which the file belongs and decides whether or not coloring should be applied to the file. To view the currently defined filename extensions and file groups, select **[Project -> File Extensions...]** to launch the "File Extensions" dialog box. To view the coloring information, select **[Setup -> Format]** to display the "Color" tab of the "Format" dialog box.

2. *I want to change the settings of a tool but the [Tools->Administration…] menu option is not selectable.*

    [Tools->Administration...] is not selectable while a workspace is open. To open the "Tool Administration" dialog box, close the current workspace.

3. *I opened a workspace from my PC, and one of my colleagues opened the same workspace simultaneously from another PC. I changed the settings of the workspace and saved it. My colleague saved the workspace after me. I opened the workspace again and found that the settings of the workspace differed from those I had made.*

    The last settings to be saved are effective. While a workspace is open in the High-performance Embedded Workshop, updating of the workspace is within the memory. The settings are not saved in a file unless the user intentionally saves the workspace.

In addition to above, refer to FAQs on the emulator and High-performance Embedded Workshop on the Renesas web site (www.renesas.com).

Appendix B   Window Functions

This section describes the window functions that can be used with the E10A-USB emulator High-performance Embedded Workshop. Here, the High-performance Embedded Workshop common functions are omitted, and only the emulation functions are described.

Table B.1   Window Functions

Menu	Option		Shortcut	Toolbar Button	Remarks
View	Disassembly		Ctrl + D		Opens the [Disassembly] window.
	Command Line		Ctrl + L		Opens the [Command Line] window.
	TCL toolkit		Shift + Ctrl + L		Opens the [Console] window.
	Workspace		Alt + K		Opens the [Workspace] window.
	Output		Alt + U		Opens the [Output] window.
	Difference				Opens the [Difference] window.
	CPU	Registers	Ctrl + R		Opens the [Register] window.
		Memory…	Ctrl + M		Opens the [Memory] window.
		IO	Ctrl + I		Opens the [IO] window.
		Status	Ctrl + U		Opens the [Status] window.
	Sym-bol	Labels	Shift + Ctrl + A		Opens the [Labels] window.
		Watch	Ctrl + W		Opens the [Watch] window.
		Locals	Shift + Ctrl + W		Opens the [Locals] window.

Table B.1   Window Functions (cont)

Menu	Option		Shortcut	Toolbar Button	Remarks
View (cont)	Code	Eventpoints	Ctrl + E		Opens the [Event] window.
		Trace	Ctrl + T		Opens the [Trace] window.
		Stack Trace	Ctrl + K		Opens the [Stack Trace] window.
	Gra-phic	Image…	Shift + Ctrl + G		Opens the [Image] window.
		Waveform…	Shift + Ctrl + V		Opens the [Waveform] window.
Setup	Radix	Hexadecimal			Uses a hexadecimal for displaying a radix in which the numerical values will be displayed and entered by default.
		Decimal			Uses a decimal for displaying a radix in which the numerical values will be displayed and entered by default.
		Octal			Uses an octal for displaying a radix in which the numerical values will be displayed and entered by default.
		Binary			Uses a binary for displaying a radix in which the numerical values will be displayed and entered by default.
	Emu-lator	System…			Opens the [Configuration] dialog box allowing the user to modify the debugging platform settings.
		Start/Stop Function Settings…			Displays the [Start/Stop function settings] dialog box.
		Save Break condition …			Opens the [Save Break condition] dialog box
		Load Break condition …			Opens the [Load Break condition] dialog box

Table B.1   Window Functions (cont)

Menu	Option	Shortcut	Toolbar Button	Remarks
Debug	Debug Sessions…			Opens the [Debug Sessions] dialog box to list, add, or remove the debug session.
	Debug Settings…			Opens the [Debug Settings] dialog box to set the debugging conditions or download modules.
	Reset CPU			Resets the target MCU and sets the PC to the reset vector address.
	Go	F5		Starts executing the user program at the current PC.
	Reset Go	Shift + F5		Resets the target MCU and executes the user program from the reset vector address.
	Go To Cursor			Starts executing the user program at the current PC until the PC reaches the address indicated by the current text cursor position.
	Set PC To Cursor			Sets the PC to the address at the row of the text cursor.
	Display PC	Shift + Ctrl + Y		Displays the current PC value.
	Run…			Launches the [Run Program] dialog box allowing the user to enter the PC or PC breakpoint during executing the user program.
	Step In	F11		Executes a block of user program before breaking.
	Step Over	F10		Executes a block of user program before breaking. If a subroutine call is reached, then the subroutine will not be entered.
	Step Out	Shift + F11		Executes the user program to reach the end of the current function.

Table B.1 Window Functions (cont)

Menu	Option		Shortcut	Toolbar Button	Remarks
Debug	Step…				Launches the [Step Program] dialog box allowing the user to modify the settings for stepping.
	Step Mode	Auto			Steps only one source line when the [Editor] window is active. When the [Disassembly] window is active, stepping is executed in a unit of assembly instructions.
		Assembly			Executes stepping in a unit of assembly instructions.
		Source			Steps only one source line.
	Halt Program		Esc		Stops the execution of the user program.
	Connect				Connects the debugging platform.
	Initialize				Disconnects the debugging platform and connects it again.
	Disconnect				Disconnects the debugging platform. This option cannot be used in some products.
	Download Modules				Downloads the object program.
	Unload Modules				Unloads the object program.

Appendix C   Command-Line Functions

The emulator supports the commands that can be used in the command-line window.

For details, refer to the online help.

Appendix D   Notes on High-performance Embedded Workshop

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 downloading the load module or saving in the [Verify Memory], [Save Memory], or [Trace] window because this will not allow correct file accessing to be performed.

4. 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 =

5. Line Assembly

   — Input radix

Regardless of the radix setting, the default for line assembly input is decimal. Specify H' or 0x for hexadecimal input on H8-series MCUs.

6. 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

   — File specification by commands

The current directory may be altered by file specifications in commands. It is recommended to use absolute paths to specify the files in a command file so that the current directory alteration is not affected.

      Example: FILE_LOAD C\HEW\Tools\Renesas\DebugComp\Platform \E10A-USB\Tutorial\Debug_H8S_xxxx_E10A-USB_SYSTEM \Tutorial.abs

7. Memory Save during User Program Execution

Do not execute saving, verifying, comparing, or searching of memory during user program execution.

8. Load of Motorola S-type Files

This High-performance Embedded Workshop 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.

9. [I/O] Window
   — Display and modification

   Do not change values in the [I/O Registers] window because the emulator uses the address break controller.

   Note that the emulator does not support the display of the invalid module or bit information in the [I/O] window.

10. Note on [Register] Window Operation during Program Execution

   The register value cannot be changed in the [Register] window during program execution. Even if the changed value is displayed, the register contents are not changed actually.

11. Break Functions

   When the PC breakpoint is set in the flash memory area, the program is written to the flash memory each time the user program is executed. At this time, note that the number of rewritable times will be decreased.

   — BREAKPOINT cancellation

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

   Checking of data at the address of the breakpoint where the user program execution stopped has failed. This is because the data at the address has been changed, or writing or reading was not successful. So the original data has not been written back to the address. Address = H′xxxx

12. Number of BREAKPOINT and [Stop At] Settings in the [Run...] Menu

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

13. 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.

14. Note on Displaying `Timeout error`

   If `Timeout error` is displayed, the emulator cannot communicate with the target MCU. Turn off the emulator and the user system and connect the emulator again by using the High-performance Embedded Workshop.

RENESAS

15. 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 [Length] specification in the [Move] dialog box is ignored. Memory is copied in a byte unit.

16. Note on Using the [Run Program] Dialog Box

When [Run...] is selected from the [Debug] 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.

17. Memory Access during User Program Execution

When a memory is accessed from the memory window, etc. during user program execution, the user program is resumed after it has stopped in the emulator to access the memory. Therefore, realtime emulation cannot be performed.

When [Short break] or [Memory access] (default: 'Enable') is set to 'Disable' in the [Configuration] dialog box, temporary halting of the user program execution by a memory access is disabled. However, even if you attempt to view memory data during the user program execution via window [IO] or [Memory], automatic update of the watch function, or the use of the tool-chip watch function, all values are shown as H'FF.

The stopping time of the user program is as follows:

Environment:

    Host computer: Core^{TM} 2 Duo 2 GHz

    H8S/2377F: 33 MHz (system clock frequency)

When a one-byte memory is read from the command-line window, the stopping time will be about 65 ms (the same result will apply to H8SX).

18. Setting Hardware Breakpoints While the User Program is Running

When you set a hardware breakpoint while the user program is running, the emulator suspends and then resumes the program execution. The program is thus not running in realtime.

19. MCU for use in debugging

The emulator is also applicable as a programming tool in mass production when it is operating in the [Program Flash] mode.
However, the flash memory of an actual MCU which has been used in connection with the emulator in the [E10A-USB Emulator] mode for debugging will have been programmed and subjected to stress accordingly. Do not use an MCU that has been used for debugging in a mass-produced product.

20. Suspending PC Operation while Using the Emulator

Do not place the host computer in sleep mode or suspended mode while using the emulator. The emulator is not usable when the host computer is in these modes. Reconnect the emulator after the host computer has returned from sleep mode or suspended mode.

21. Verification of Memory

Use the FILE_VERIFY command to verify memory. Do not use [Verify Memory…] in the [File] menu.

22. Overwrite of a File

In command line interface, a file that has the same name as the output file is overwritten without asking the user.

23. Usage with Other Products

To install the SuperH™ RISC engine C/C++ compiler package or H8SX, H8S, or H8 Family C/C++ compiler package after installing the emulator software for High-performance Embedded Workshop, select [High-performance Embedded Workshop] as the component to be installed in [Select Components] during installation of the compiler package.

24. Download of Load Modules

When load modules are downloaded, select [Debug] -> [Download] -> [All Download modules].

25. Program Flash Mode

When MCUs are continuously programmed, be sure to turn the target on or off.

26. Memory Access in the Program Flash Mode

Memory cannot be accessed in the Program Flash mode. In this mode, values displayed in the [Memory] or [IO] window are dummy.

27. Memory Access during Flash Memory Programming

During flash memory programming (e.g., user program execution), operation for memory accessing such as opening the [Memory] window is not allowed. Values displayed here are dummy. Access the memory again after flash memory programming has been completed.

28. Download of Programs

Verification cannot be enabled by selecting [Perform memory verify during download] checkbox in the [Download Module] dialog box in which download modules can be specified. To perform verification after downloading, specify verification in the [Verify Memory] dialog box opened by selecting [Verify Memory] from the [Debug] menu.

29. Loading Sessions

Information in [JTAG clock] of the [Configuration] dialog box cannot be saved by sessions. The JTAG clock value becomes the initial value when starting up the emulator.

30. Connecting Two or More Emulators

If two or more emulators are to be connected to a single personal computer, the emulator that was connected earlier should be in the break state whenever you connect another emulator.

31. Manual Navigator

Follow the procedure below to execute this program under Windows Vista® or Windows® 7.

Work-around:

(1) Log in with administrative rights.
(2) Open the properties window for file man_navi.exe in the Manuals folder under the installation folder for the High-performance Embedded Workshop.
(3) On the [Compatibility] tabbed page, check the [Run this program as an administrator] box.

Appendix E   Diagnostic Test Procedure

For the diagnostic test procedure using the emulator test program, refer to the E10A-USB Emulator Test Program Manual (file name: E10A-USBTME.PDF) in the 'E10A-USB Emulator software' CD.

Appendix F   Repair Request Sheet

Thank you for purchasing the E10A-USB emulator (HS0005KCU01H or HS0005KCU02H).

In the event of a malfunction, fill in the repair request sheet on the following pages and send it to your distributor.

Repair Request Sheet

To Distributor

Your company name:

Person in charge:

Tel.:

Item	Symptom
1. Date and time when the malfunction occurred	Month/Day/Year {at system initiation, in system operation} *Circle either of items in the braces { }.
2. Frequency of generation of the malfunction	( ) times in ( ) {day(s), week(s), or month(s)} *Enter the appropriate numbers in the parentheses ( ) and circle one of the three items in the braces { }.

3. System configuration when the malfunction occurred	System configuration of the emulator:
	• E10A-USB emulator (HS0005KCU01H or HS0005KCU02H):
	    Serial No.:
	    Revision:
	    The above items are written on the label for product management at the bottom of the emulator unit; the serial no. is the five-digit number and the revision is the string of letters following the number.
	• Provided CD-R (HS0005KCU01SR):
	    Version: V.
	    Shown as 'V.x.xx release xx' on the CD-R (x: numeral).
	• Host computer in use:
	    Manufacturer:
	    Type number:
	    OS: (Windows® XP, Windows Vista®, or Windows® 7)

Item	Symptom
4. Settings when the malfunction occurred	(1) MCU: Part number: (2) Operating frequency: MHz
5. Failure phenomenon	
6. Error in debugging	
7. Error in the diagnostic program	
8. The High-performance Embedded Workshop does not link-up with the emulator.	Content of the error message

For errors other than the above, fill in the box below.

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

H8S, H8SX Family E10A-USB Emulator
User's Manual