To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

RENESAS

Notice

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

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

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

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

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

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

User's Manual

SuperH Family E6000 Emulator HS7000EPI60HE

User's Manual

Renesas Microcomputer Development Environment System

Renesas Electronics
www.renesas.com

Rev.2.0 2003.06

IMPORTANT INFORMATION

READ FIRST

- **READ this user's manual before using this E6000 emulator.**
- **KEEP the user's manual handy for future reference.**

Do not attempt to use the E6000 emulator until you fully understand its mechanism.

E6000 emulator:

Throughout this document, the term “E6000 emulator” shall be defined as the E6000 emulator, user system interface cable, PC interface board, and optional SIMM memory module produced only by Renesas Technology Corp. excluding all subsidiary products.

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

Purpose of the E6000 emulator:

This E6000 emulator is a software and hardware development tool for systems employing the Renesas microcomputer SH series (hereafter referred to as MCU). This E6000 emulator must only be used for the above purpose.

Improvement Policy:

Renesas Technology Corp. (including its subsidiaries, hereafter collectively referred to as Renesas) pursues a policy of continuing improvement in design, functions, performance, and safety of the E6000 emulator. 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 E6000 emulator:

This E6000 emulator 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 E6000 emulator 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 E6000 emulator.

LIMITED WARRANTY

Renesas warrants its E6000 emulators to be manufactured in accordance with published specifications and free from defects in material and/or workmanship. Renesas, at its option, will repair or replace any E6000 emulators returned intact to the factory, transportation charges prepaid, which Renesas, upon inspection, determine to be defective in material and/or workmanship. The foregoing shall constitute the sole remedy for any breach of Renesas's 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 E6000 EMULATOR, THE USE OF ANY E6000 EMULATOR, OR ITS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. EXCEPT AS EXPRESSLY STATED OTHERWISE IN THIS WARRANTY, THIS E6000 EMULATOR IS SOLD “AS IS ”, AND YOU MUST ASSUME ALL RISK FOR THE USE AND RESULTS OBTAINED FROM THE E6000 EMULATOR.

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 E6000 emulator without Renesas's prior written consent or any problems caused by the user system.

All Rights Reserved:

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's 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.

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 E6000 emulator are therefore not all inclusive. Therefore, you must use the E6000 emulator safely at your own risk.

SAFETY PAGE

READ FIRST

- **READ this user's manual before using this E6000 emulator.**
- **<u>KEEP the user's manual handy for future reference.</u>**

Do not attempt to use the E6000 emulator 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.

⚠ 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 E6000 emulator and user system before connecting or disconnecting any CABLES or PARTS.**
3. **Always before connecting any CABLES, make sure that pin 1 on both sides are correctly aligned.**
4. **Supply power according to the power specifications and do not apply an incorrect power voltage. Use only the provided power cable.**

About This Manual

This manual explains how to set up and use the E6000 Emulator for the SH series microcomputers. It is the Debugging Platform User's Manual for all SH series E6000 emulators. For detailed specifications on each E6000 emulator, refer to the supplementary information supplied with the E6000 emulator.

Section 1, Introduction, gives a rapid introduction to the system's facilities, including an overview of the main emulation features provided by the E6000 emulator and the HDI software that provides access to them.

Section 2, Setting Up, describes how to set up the E6000 emulator and prepare it for use in conjunction with the HDI.

Section 3, Hardware, explains how to connect the E6000 emulator to an external user system.

Section 4, Tutorial, then introduces each of the E6000 emulator's main features by showing how to load and debug a simple C program. The tutorial program is supplied on disk so that you can follow the steps on your own system to learn first-hand how it operates.

Assumptions

This manual assumes that you already have a working knowledge of the procedures for running and using programs for MS-DOS® and Microsoft® Windows® operating system.

This manual also assumes that the operating environment is the English version of Microsoft® Windows® 98 operating system running on the IBM PC.

Related Manuals

- Supplementary Information
- HDI User's Manual
- User System Interface Cable User's Manual
- PC Interface Board User's Manual
- SIMM Memory Module User's Manual

Conventions

This manual uses the following typographical conventions:

Style	Used for
`computer`	Text that you type in, or that appears on the screen.
`parameter`	A label representing the actual value you should type as part of a command.
bold	Names of menus, menu commands, buttons, dialog boxes, and windows that appear on the screen.

Trademarks

Microsoft®, MS®, MS-DOS®, Windows® and WindowsNT® are registered trademarks of Microsoft Corporation in the United States and/or in other countries.

IBM is a registered trademark of International Business Machines Corporation.

This manual assumes the operating environment to be the English version of Microsoft® Windows® 98 operating system.

Contents

Figures

Tables

Section 1 Introduction

The E6000 emulator is an advanced realtime in-circuit emulator which allows programs to be developed and debugged for the SH series microcomputers.

The E6000 emulator can either be used without a user system, for developing and debugging software, or connected via a user system interface cable to a user system, for debugging user hardware.

The E6000 emulator works with the HDI, an interface program based on Microsoft® Windows® operating system. This provides a powerful range of commands for controlling the emulator hardware, with a choice of either fully interactive or automated debugging.

1.1 Debugging Features

1.1.1 Breakpoints

The E6000 emulator provides a comprehensive range of alternative types of breakpoints, to give you the maximum flexibility in debugging applications and user system hardware.

Hardware Break Conditions (Type 1 and Type 2): Up to 12 break conditions can be defined using the event and range channels in the complex event system (CES). For more information about the hardware break conditions, see section 1.2, Complex Event System (CES).

On-chip Breakpoint (Type 3): In target ROM, three breakpoints (on-chip break) can be set.

Program Breakpoints (PC Breakpoints): Up to 256 program breakpoints can be defined. These program breakpoints are set by replacing the user instruction by a BREAK instruction.

1.1.2 Trace

The E6000 emulator incorporates a powerful realtime trace facility which allows you to examine MCU activity in detail. The realtime trace buffer holds up to 65535 bus cycles, and it is continuously updated during execution. The buffer is configured as a rolling buffer, which can be stopped during execution and read back by the host computer without halting emulation.

The data stored in the trace buffer is displayed in both source program and assembly languages for ease of debugging. However, if trace filtering is used, only assembly language can be displayed.

The buffer can be set up to store all bus cycles or just selected cycles. This is called trace acquisition and uses the complex event system (CES) to select the parts of the program you are interested in; see section 1.2, Complex Event System (CES), for more information.

It is also possible to store all bus cycles and then just look at selected cycles. This is called trace filtering.

1.1.3 Execution Time Measurements

The E6000 emulator allows you to measure the total execution time, or to measure the time of execution between specified events in the complex event system. You can set the resolution of the timer to any of the following values:

20 ns, 125 ns, 250 ns, 500 ns, 1 µs, 2 µs, 4 µs, 8 µs, or 16 µs.

At 20 ns the maximum time that can be measured is about six hours, and at 16µs the maximum time is about 200 days.

1.1.4 Performance Analysis

The E6000 emulator provides functions for measuring the performance of a program. The performance of the specified program range can be displayed either as a histogram or in percentage form. A timer resolution of 20 ns, 40 ns, or 160 ns can be selected. In addition, the execution count of the specified program range can be measured (1 to 65535).

1.2 Complex Event System (CES)

In most practical debugging applications, the program or hardware errors that you are trying to debug occur under a certain restricted set of circumstances. For example, a hardware error may only occur after a specific area of memory has been accessed. Tracking down such problems using simple PC breakpoints can be very time consuming.

The E6000 emulator provides a very sophisticated system for giving a precise description of the conditions you want to examine, called the complex event system. This allows you to define events which depend on the state of a specified combination of the MCU signals.

The complex event system provides a unified way of controlling the trace, break, and timing functions of the E6000 emulator.

1.2.1 Event Channels

The event channels allow you to detect when a specified event has occurred. The event can be defined as a combination of one or more of the following:

- Address or address range
- Address outside range
- Data, with an optional mask
- Read or Write or either
- MCU access type (e.g., DMAC and instruction prefetch)
- MCU access area (e.g., on-chip ROM and on-chip RAM)
- A signal state on one or more of the four external probes
- A certain number of times that the event must be triggered
- Delay cycles after an event

Up to eight events can be combined into a sequence, in which each event is either activated or deactivated by the occurrence of the previous event in the sequence. For example, you can cause a break if an I/O register is written to after a specified area of RAM has been accessed.

1.2.2 Range Channels

The range channels can be set up to be triggered on a combination of one or more of the following:

- Address or address range (inside the range)
- Data, with an optional mask
- Read or Write or either
- MCU access type (e.g., DMAC and instruction prefetch)
- MCU access area (e.g., on-chip ROM and on-chip RAM)
- A signal state on one or more of the four external probes
- Delay cycles after an event

The complex event system can be used to control the following functions of the E6000 emulator:

1.2.3 Breaks

You use breaks to interrupt program execution when a specified event, or sequence of events, is activated. For example, you can set up a break to halt execution when the program reads from one address, and then writes to another address. The break can also optionally be delayed by up to 65535 bus cycles.

1.2.4 Timing

You can set up two events and then measure the execution time of the program between the activation of the first event and second event.

1.3 Hardware Features

1.3.1 Memory

The E6000 emulator provides standard emulation memory as the substitute for on-chip ROM memory and on-chip RAM memory. When a device type or device mode without an on-chip ROM or on-chip RAM is selected, the standard emulation memory is disabled. When debugging with only the E6000 emulator and the user program and data are stored in an external address space, an optional SIMM memory module must be used. The optional SIMM memory modules can be separately purchased.

The emulation memory can be mapped in units to any number of separate memory blocks in the MCU address space according to table 1.1. Each memory block can be specified using the **Memory Mapping…** function as user (Target) or emulator (SIMM memory module) and, in each case, the access can be specified as read-write, read-only, or guarded.

Table 1.1 Emulation Memory

Type	High-speed emulation memory	Low-speed emulation memory
HS6000EMS21H	128 kbyte × 4 area	512 kbyte × 4 area
HS6000EMS22H		512 kbyte × 12 area

The definition of each type of memory is as follows:

Table 1.2 Memory Types

Memory Type	Description
On chip	Uses the MCU on-chip memory.
Target	Accesses the user system memory.
Emulator	Accesses the E6000 emulator SIMM memory module.

The contents of a specified block of memory can be displayed using the **Memory Window…** function. The contents of memory can be modified at any time, even during program execution and the results are immediately reflected in all other appropriate windows.

Note that the time taken to modify memory contents during program execution may differ depending on the settings, but approximately has the following time requirements:

1. MCU on-chip or ROM, or emulator SIMM memory module

   The E6000 emulator modifies the memory contents by temporarily switching the memory bus to the emulator side without stopping the user program execution. For both memory read and memory write accesses, the HDI stores a maximum of 256 bytes of memory contents in the buffer. Therefore, the emulator uses the memory bus for up to 80 μs (25 MHz, on-chip ROM)

2. MCU on-chip RAM, or I/O user system memory

   The E6000 emulator stops the user program execution, then modifies the memory contents. As stated above, a maximum of 256 bytes of memory contents are accessed. Therefore, the user program stops for a maximum of 2 ms (25 MHz, emulation memory).

1.3.2 Clocks

The clock can be specified as E6000 emulator internal clock or target clock. The frequencies that can be specified as the emulation clock depend on the MCU. For details, refer to the supplementary information supplied together with the emulator.

1.3.3 Probes

External probe (EXT) can be connected to the E6000 emulator, to make use of signals from other parts of your user system hardware, and can be used to trigger the complex event system depending on whether the probe signal is low or high.

1.3.4 Environment Conditions

Observe the conditions listed in table 1.3 when using the E6000 emulator.

Table 1.3 Environment Conditions

Item	Specifications
Temperature	Operating: +10 to +35°C
	Storage: –10 to +50°C
Humidity	Operating: 35 to 80% RH; no condensation
	Storage: 35 to 80% RH; no condensation
Ambient gases	No corrosive gases
AC Power supply voltage	100 V to 240 V AC 50/60 Hz 0.6 A max.
User system voltage (UVcc)	Depends on the target MCU within the range 2.7 V to 5.5 V

1.3.5 Emulator External Dimensions and Weight

Item	Specifications
Dimensions	219 × 170 × 54 mm
Weight	1,000 g

Section 2 Setting Up

This section explains how to:

- Set up the PC interface board (HS6000EII01H separately purchased).
- Set up the E6000 emulator.
- Install the HDI software and use it to check correct operation of the entire system.

To use another interface board, such as a PC card (PCMCIA), refer to the user's manual for that interface board.

The E6000 emulator communicates with the HDI through the PC interface board, and therefore, the PC interface board must be inserted into the host computer.

The PC interface board is a memory-mapped board, and before inserting it you first need to reserve a block of memory addresses for use by the board. This ensures that other programs do not inadvertently use the PC interface hardware.

The allocated memory area must not overlap memory already allocated to other board. If attempted, the PC interface board and the E6000 emulator product will not operate correctly. At shipment, the memory area of PC interface board is allocated to the address range from H'D0000 to H'D3FFF.

When using Microsoft® Windows® 95 or Microsoft® Windows® 98 operating system, refer to section 2.2, Setting Up the PC Interface Board on Windows® 95 or Windows® 98. When using Microsoft® WindowsNT® operating system, refer to section 2.3, Setting Up the PC Interface Board on WindowsNT® 4.0.

2.1 Package Contents

The E6000 emulator is supplied in a package containing the following components.

- E6000 emulator
- 5V and 5A E6000 emulator power supply (AC adapter)
- Test program disk
- HDI installation disks
- External probes
- Supplementary Information
- SH Series E6000 Emulator User’s Manual (this manual)
- HDI User’s Manual

Before proceeding you should check that you have all the items listed above, and contact your supplier if any are missing.

2.2 Setting Up the PC Interface Board on Windows® 95 or Windows® 98

2.2.1 Setting Up the PC Interface Board

- Start Windows® 95 or Windows® 98.
- Click the **My Computer** icon with the right mouse button and select **Properties** from the pop-up menu.

The **System Properties** dialog box will be displayed.

- Double-click the **Computer** icon in the **Device Manager** panel to open the **Computer Properties** dialog box.
- Click the **Memory** in the **View Resources** panel to display the memory resources.

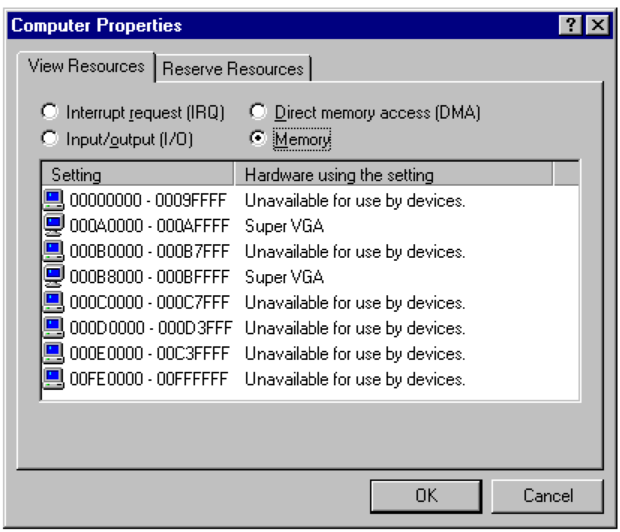


Figure 2.1 Computer Properties Dialog Box (Before Setting)

A memory area that is not listed in the dialog box can be assigned to the PC interface board. Table 2.1 lists the address ranges that can be set by the switch on the rear panel of the PC interface board. Select one of the address ranges that is not listed in the **Computer Properties** dialog box. For example, if you select the range H'D8000 to H'DBFFF, the corresponding switch number will be 6.

Table 2.1 Address Map of PC Interface Board and Memory Switch Setting

Address Range	Switch Setting
From H'C0000 to H'C3FFF	0
From H'C4000 to H'C7FFF	1
From H'C8000 to H'CBFFF	2
From H'CC000 to H'CFFFF	3
From H'D0000 to H'D3FFF (at shipment)	4
From H'D4000 to H'D7FFF	5
From H'D8000 to H'DBFFF	6
From H'DC000 to H'DFFFF	7
From H'E0000 to H'E3FFF	8
From H'E4000 to H'E7FFF	9
From H'E8000 to H'EBFFF	A
From H'EC000 to H'EFFFF	B

Define the memory area so that Windows® 95 or Windows® 98 does not use the area as follows:

- Click **Memory** in the **Reserve Resources** panel and click **Add**.

The **Edit Resource Setting** dialog box will be displayed.

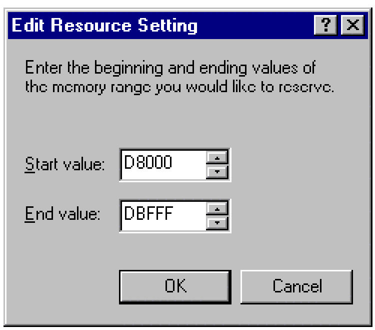


Figure 2.2 Edit Resource Setting Dialog Box

- Enter the memory area addresses in **Start value** and **End value**.
- Shut down the host computer (do not restart it) and turn off the power switch.
- Using a small screwdriver, rotate the switch in the rear panel of the PC interface board so that the arrow points to the number corresponding to the memory area you have selected.
- Remove the cover from the host computer and install the PC interface board in a spare ISA slot.

- Replace the host computer cover.
- Connect the PC interface cable between the PC interface board and the PC IF connector on the E6000 emulator. Press each plug firmly home until it clicks into position.
- Switch on the host computer.
- Open the **Computer Properties** dialog box and check that the memory area you have selected is listed as `System Reserved`.

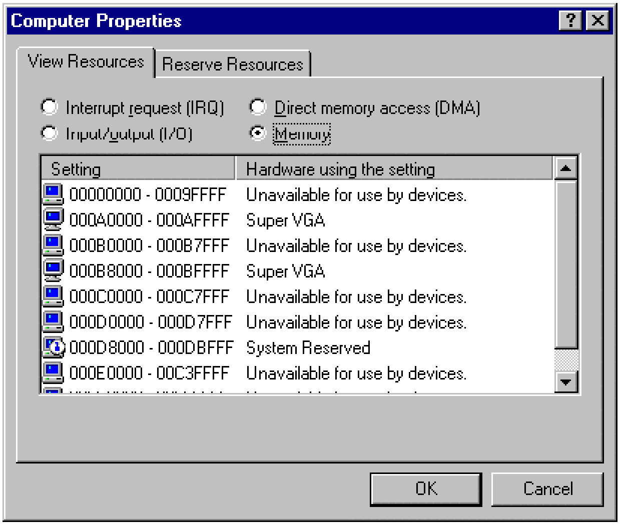


Figure 2.3 Computer Properties Dialog Box (After Setting)

2.2.2 Modifying the CONFIG.SYS File

Prevent the memory area for the PC interface board being accessed by another program as follows:

- Select **Run** from the **Start** menu.
- Type `SYSEDIT` and click **OK.**

When `EMM386.EXE` is used in the `CONFIG.SYS` file, the `CONFIG.SYS` file must be modified. If the `CONFIG.SYS` file is not used, or if `EMM386.EXE` is not used even when the `CONFIG.SYS` file is used, go to Section 2.2.3, Modifying the SYSTEM.INI File.

- Locate the line in the `CONFIG.SYS` file that reads:

```
DEVICE=C:\WINDOWS\EMM386.EXE
```

- Change the line so that it reads as shown below.

  `DEVICE=C:\WINDOWS\EMM386.EXE X=`*`aaaa–bbbb`*

  Here, *aaaa* is the upper four digits of **Start value** and *bbbb* is the upper four digits of **End value**. For example, for the switch set to 6, you would set the line to read:

  `DEVICE=C:\WINDOWS\EMM386.EXE X=D800–DBFF`

- Save the `CONFIG.SYS` file.

2.2.3 Modifying the SYSTEM.INI File

- Add the following line to the `[386enh]` section in the `SYSTEM.INI` file:

  `EMMExclude=`*`aaaa-bbbb`*

  Here, *aaaa* is the upper four digits of **Start value** and *bbbb* is the upper four digits of **End value**. For example, for the switch set to 6, you would set the line to read:

  `EMMExclude = D800-DBFF`

- Save the `SYSTEM.INI` file and exit the `SYSEDIT`.
- Restart the host computer.

  This ensures that Windows® will not use this block of memory. You are ready to connect up the E6000 emulator and run the HDI to check communication to it.

2.3 Setting Up the PC Interface Board on WindowsNT® 4.0

The PC interface board uses the ISA bus slot, and therefore the host computer must have a spare ISA bus slot.
This section describes the general procedure for installing the PC interface board in the host computer. For details, refer to the manual of your host computer.

Starting WindowsNT®:

- Execute **Start/Programs/Administrative Tools (Common)/WindowsNT Diagnostics**.
- Click the **Memory** button in the **Resource** tab and, in the following form, make a note of the upper memory areas that have already been used.

#	Start	End	#	Start	End	#	Start	End
0			4			8		
1			5			9		
2			6			A		
3			7			B		

- Shut down WindowsNT®.

Starting the Host Computer in Setup Mode:

For details on the setup mode, refer to the manual of your host computer.

- Check which upper memory areas have already been used.

#	Start	End	#	Start	End	#	Start	End
0			4			8		
1			5			9		
2			6			A		
3			7			B		

The memory areas being used should be the same as those checked for WindowsNT® above.

- Define the memory area for the PC interface board. Select one of the memory areas that correspond to the following PC interface board switch settings, and no other devices can access the selected memory area.

#	Start	End	#	Start	End	#	Start	End
0	H'C0000	H'C3FFF	4*	H'D0000	H'D3FFF	8	H'E0000	H'E3FFF
1	H'C4000	H'C7FFF	5	H'D4000	H'D7FFF	9	H'E4000	H'E7FFF
2	H'C8000	H'CBFFF	6	H'D8000	H'DBFFF	A	H'E8000	H'EBFFF
3	H'CC000	H'CFFFF	7	H'DC000	H'DFFFF	B	H'EC000	H'EFFFF

Note: 4 is the setting at shipment.

If the **Intel P&P BIOS** disk is supplied with the host computer, define the memory area as follows:

- Start the host computer with the Intel P&P BIOS disk.
- Check the upper memory areas that have already been used, with **View/System Resources**.
- Add **Unlisted Card** with **Configure/Add Card/Others...**.
- Click **No** in the dialog box displayed because there is no .CFG file.
- Move to the **Memory [hex]** list box in the **Configure Unlisted Card** dialog box.
- Click the **Add Memory...** button to display the **Specify Memory** dialog box.
- Enter a memory area range that is not used by any other device and that corresponds to one of the PC interface board switch settings.

- Save the file.
- Exit the current setup program.
- Shut down the host computer (do not restart it) and turn off the power switch.
- Using a small screwdriver, rotate the switch in the rear panel of the PC interface board so that the arrow points to the number corresponding to the memory area you have selected.
- Remove the cover from the host computer and install the PC interface board in a spare ISA slot.
- Replace the host computer cover.
- Connect the PC interface cable between the PC interface board and the PC IF connector on the E6000 emulator. Press each plug firmly home until it clicks into position.
- Switch on the host computer.

Starting WindowsNT® in the Administrator Mode:

- Install the HDI Software as described in section 2.4, Installing the HDI Software.
- Execute **Start/Programs/Hdi/Setup ISA bus Board**.
  If the DOS prompt window does not open, open the DOS prompt window first, move to the directory where the HDI has been installed, then execute `SETUPISA.EXE`.

2.4 Installing the HDI Software

This section describes how to install the HDI software by using the SH7010 E6000, for example. For another type of E6000 emulator, change the file and directory names to the target ones.

2.4.1 HDI Installation Procedure

- Start the host computer.
- Close all other applications that are running.
- Insert HDI installation disk #1 into the floppy disk drive of the host computer.
- Choose **Run…** from the **Start** menu.
- Type `A:setup.exe` and click **OK**:

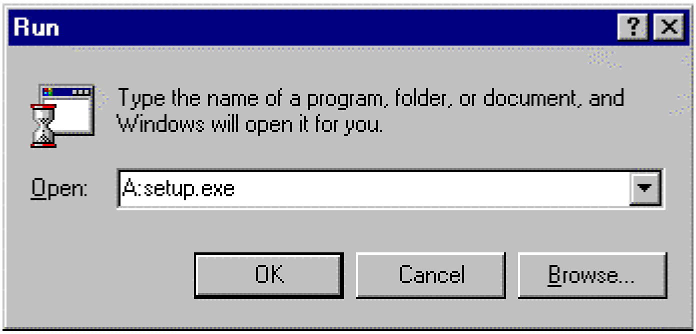


Figure 2.4 Run Dialog Box

This runs the HDI installer, and the following **Welcome!** dialog box will be displayed:

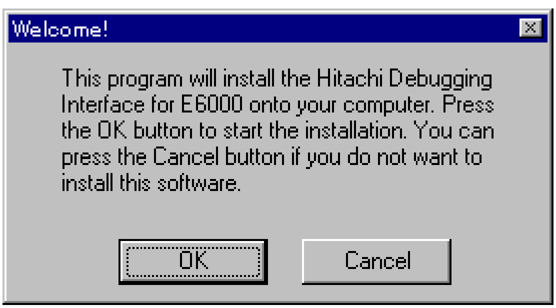


Figure 2.5 HDI Installer [Welcome!] Dialog Box

- Click **OK** to proceed with the installation.

The following dialog box displays the **Read Me** file for the version of the HDI you are installing:

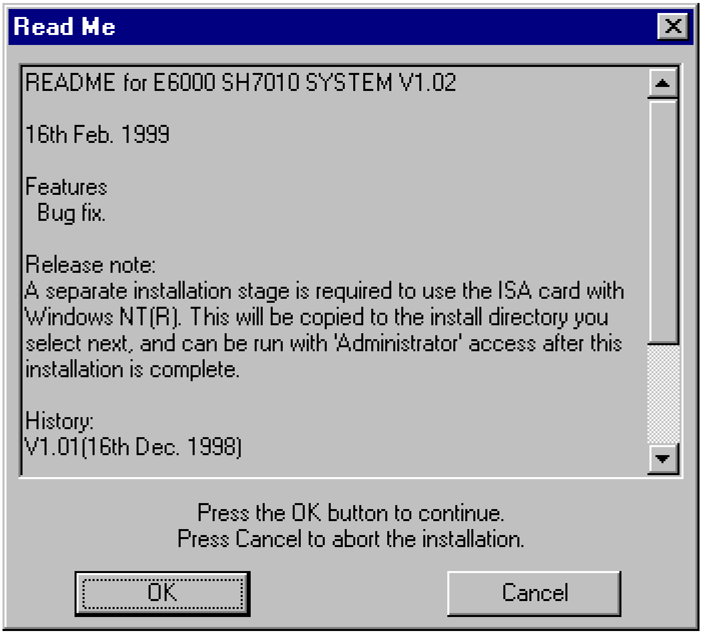


Figure 2.6 Read Me Dialog Box

- Read the **Read Me** file for any important information concerning the installation and then click **OK** to proceed.

The following dialog box then allows you to select a directory in which to install HDI:

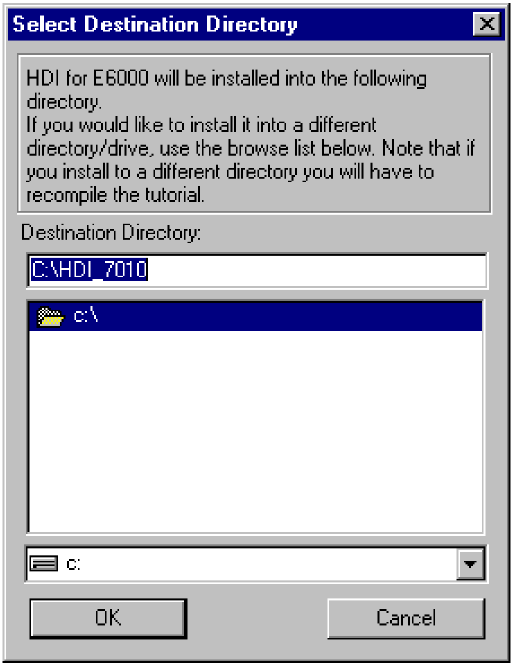


Figure 2.7 Select Destination Directory Dialog Box

- Click **OK** to install into the default directory `C:\HDI_7010`, or specify an alternative directory and click **OK**. When a directory other than the default directory is specified, file `tutorial.abs` will not be installed.

The following dialog box then asks you whether backups should be made for files replaced by the installation:

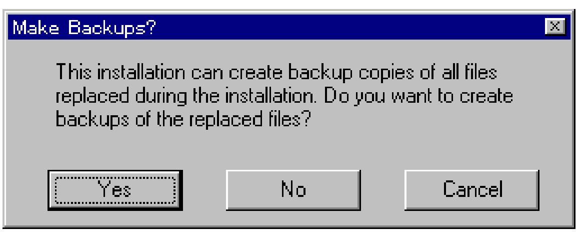


Figure 2.8 Make Backups? Dialog Box

- Click **Yes** to save any files that may be replaced as part of the installation (recommended), or **No** if you do not want to make a backup.

If you chose **Yes,** the following dialog box allows you to specify the backup directory:

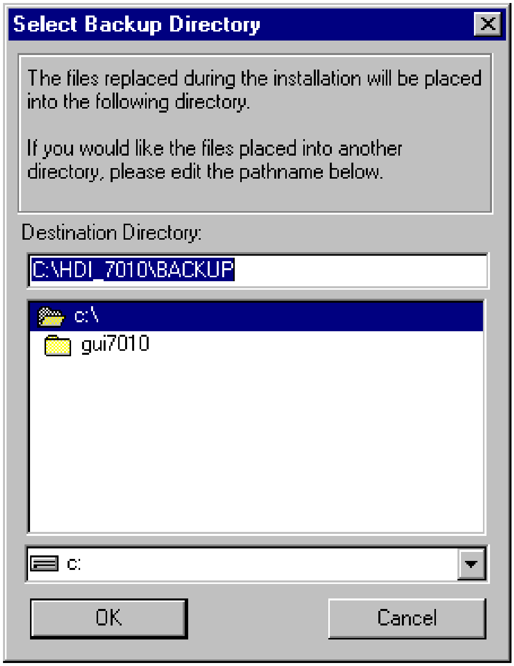


Figure 2.9 Select Backup Directory Dialog Box

- Enter the directory you want to use and click **OK**.

The installer then copies the HDI files to the specified directory:

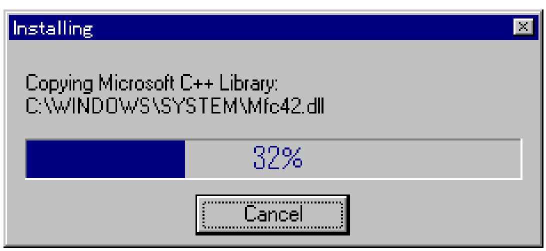


Figure 2.10 Installing Dialog Box

RENESAS

When first disk (#1) installation is completed, the installer displays this dialog box:

Figure 2.11 Insert New Disk Dialog Box

- Insert installation disk #2 and press the **OK** button.
- In the same way, insert the next installation disk according to the dialog box message and press the **OK** button. In the installation procedure, specify the target communication interface according to the dialog box message.

After the necessary files have been copied, the following dialog box allows you to specify the program group for the HDI icons:

Figure 2.12 Select Program Manager Group Dialog Box

- Select an existing group or enter the name of a new group, and click **OK** to proceed.

This completes the HDI installation.

The installer creates the following short-cuts in the program group you specified, by default **Hdi**:

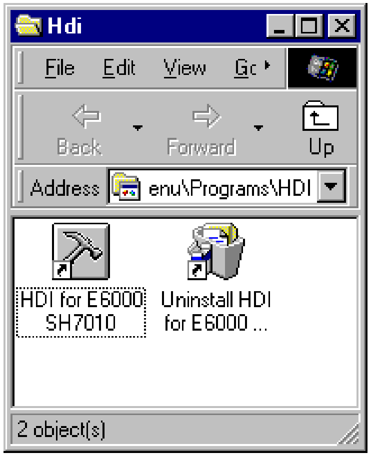


Figure 2.13 HDI Program Group

These short-cuts have the following functions:

HDI for E6000 SH7010 is the HDI software.

Uninstall HDI for E6000 SH7010 will remove HDI, and its associated files, if you need to uninstall it at any stage.

2.4.2 Checking the System

The next step is to run the HDI software to check that the E6000 emulator is working correctly.

- Switch on the E6000 emulator and check that the red LED is illuminated.
- Select **HDI for E6000 SH7010** from the **Start** menu.

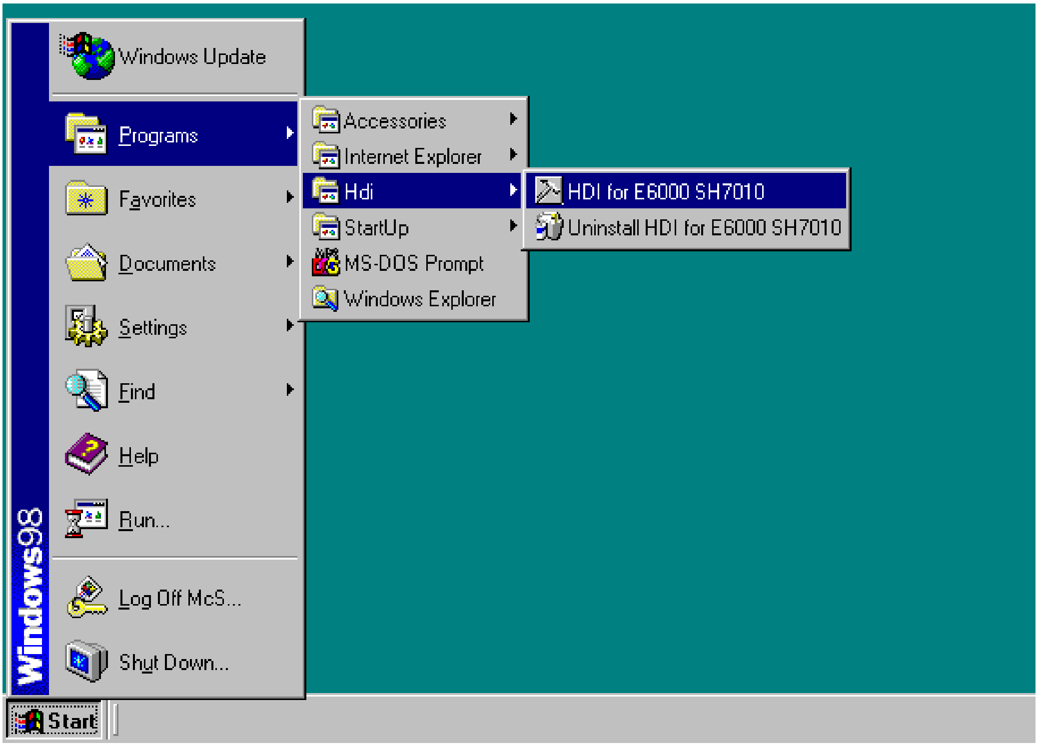


Figure 2.14 HDI Start Menu

With everything set up correctly the HDI window will be displayed, and the following messages will be shown in the status bar at the bottom of the window:

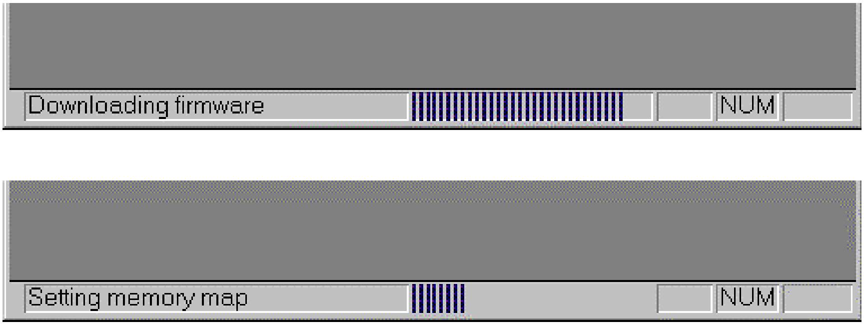


Figure 2.15   HDI Start-Up Messages

Finally the status bar will display `Link up` to indicate that everything is set up correctly, and the HDI window will be displayed as shown below.

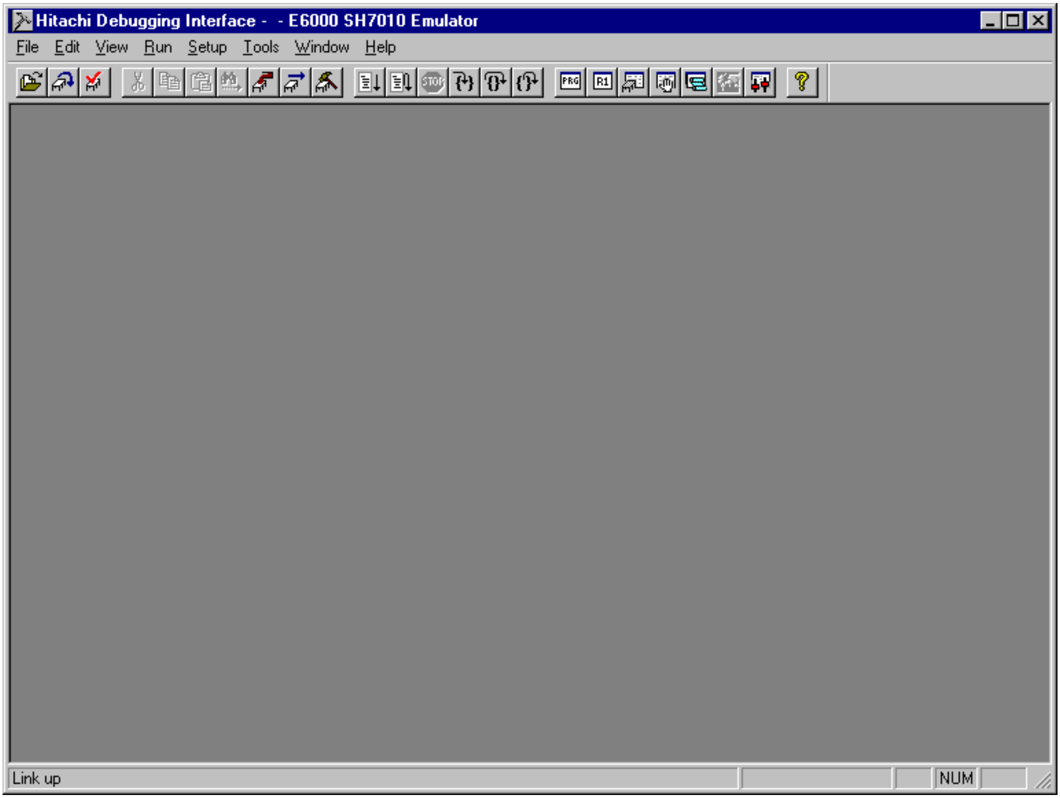


Figure 2.16 HDI Window

2.5 What Next?

The E6000 emulator is now correctly set up and ready for use. We recommend you work through section 4, Tutorial, to familiarize yourself with the key features of the E6000 emulator, and to learn how to use the E6000 emulator to develop and debug programs for the MCU.

2.6 Uninstalling the HDI Software

This section describes how to uninstall the HDI software on Windows® 95 or Windows® 98, for example.

- Select **Uninstall HDI for E6000 SH7010** from the **Start** menu.

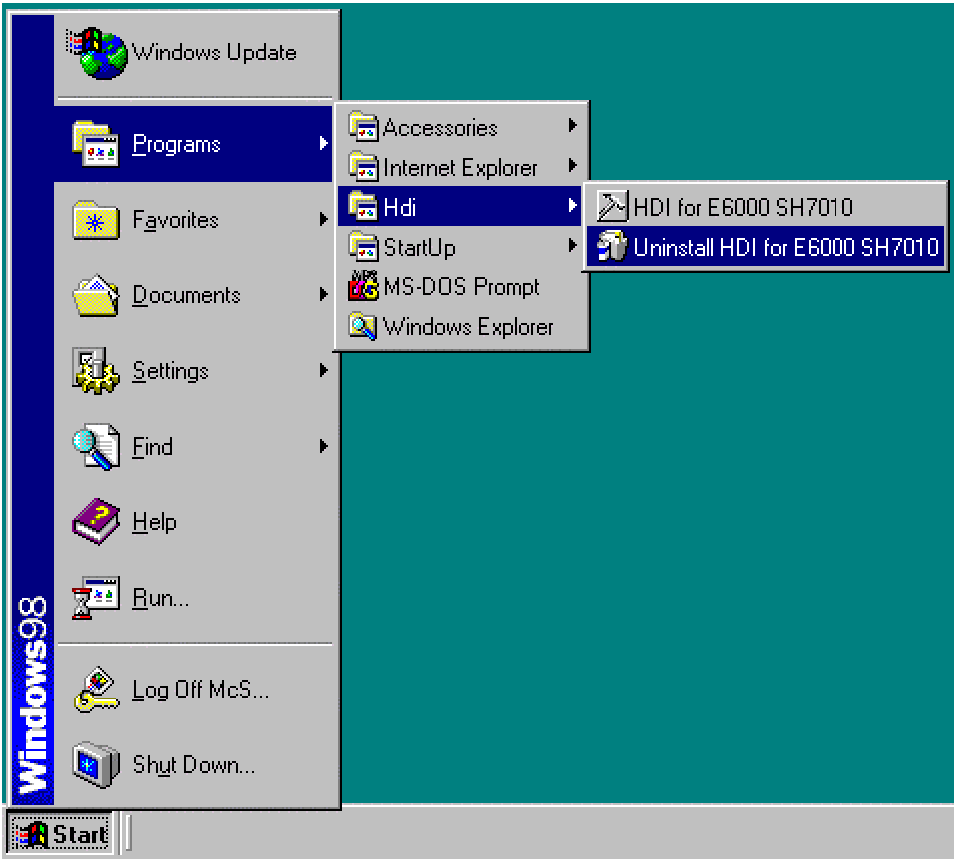


Figure 2.17 Start Menu (Uninstaller)

The uninstaller is initiated and the following dialog box will be displayed.

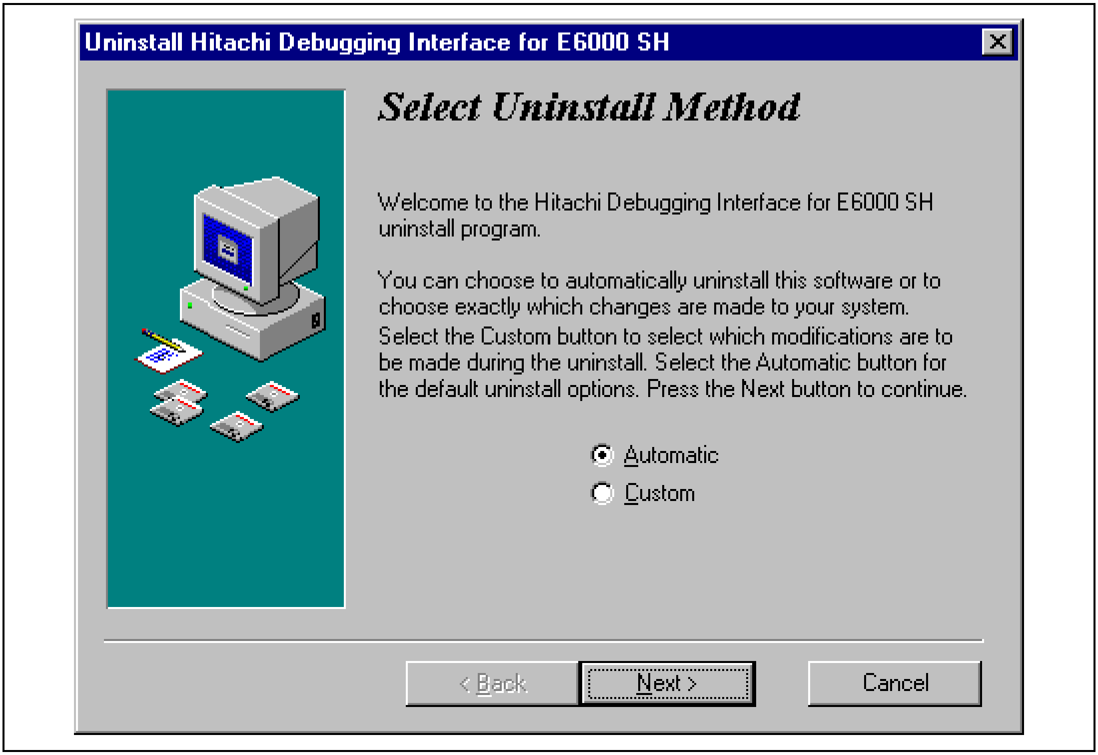


Figure 2.18 Select Uninstall Method Dialog Box

- To automatically uninstall the HDI, select the **Automatic** radio button and click **Next**.
- To select the files to delete, select **Custom** and click **Next.**
- To cancel uninstallation, click **Cancel**.

When backup files were made at installation, the dialog box to confirm whether to roll back the backup files will be displayed.

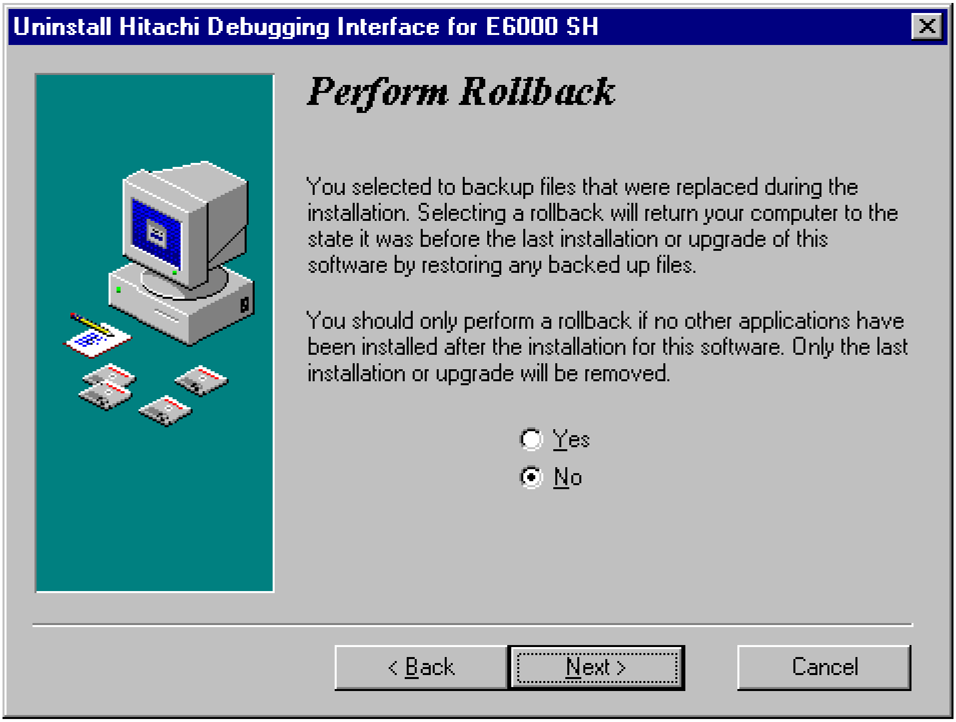


Figure 2.19 Perform Rollback Dialog Box

- To perform rollback, select the **Yes** radio button and click **Next**.
- To not perform rollback, select the **No** radio button and click **Next**.
- To cancel uninstallation, click **Cancel**.
- To go back to the **Select Uninstall Method** dialog box, click **Back**.

Notes: 1. By performing rollback, the backup files are restored.

2. If no backup files have been made or if no backup files are found, the **Perform Rollback** dialog box will not be displayed.

- The dialog box to confirm whether to start uninstallation will be displayed.

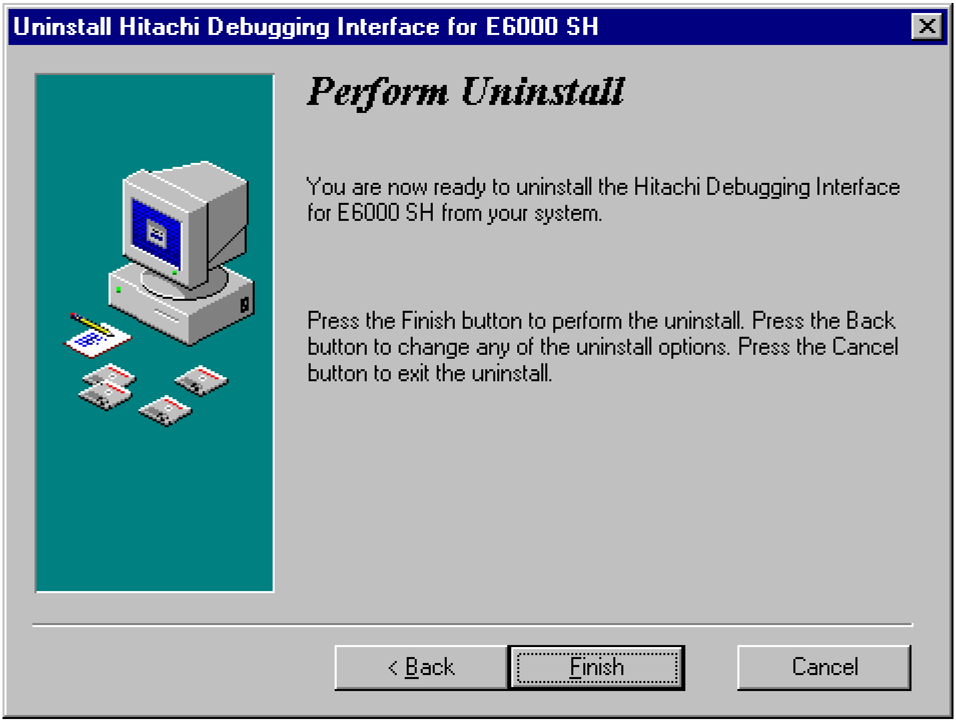


Figure 2.20 Perform Uninstall Dialog Box

- To start uninstallation, click **Finish**.
- To cancel uninstallation, click **Cancel**.
- To go back to the **Select Uninstall Method** dialog box, click **Back**.

When uninstallation is successfully completed, the directories and files created by the installer are deleted.

Note: 1. Any subdirectory or file that you have created in the HDI directory will not be deleted by the uninstaller.

2. When rollback was not performed, backup directory and files will not be deleted.

2.7 Troubleshooting

2.7.1 Faulty Connection

If the following message box appears during initialization, the PC interface board was not able to detect the E6000 emulator.

Figure 2.21 Faulty Connection Message

This indicates:

- Power supply not connected to the E6000 emulator, or the emulator not switched on. Check the power LED on the E6000 emulator.
- The PC interface cable is not correctly connected between the PC interface board and the E6000 emulator.

2.7.2 Communication Problems

The following message box indicates that the HDI was not able to set up the E6000 emulator correctly:

Figure 2.22 Communication Problem Message

This indicates:

- The memory area reserved in the `CONFIG.SYS` file does not match the interface switch setting on the rear panel of the PC interface board.
- Selected area of memory is in use by another application.

Section 3 Hardware

This section explains how to connect the E6000 emulator to a user system.

3.1 Connecting to the User System

To connect the E6000 emulator to a user system, proceed as follows:

- Connect the user system interface cable head to the user system.
- Plug the cable body into the E6000 emulator.
- Plug the cable body into the cable head.

For details of these steps, refer to the User System Interface Cable User's Manual.

Figure 3.1 gives details of the connectors provided on the E6000 emulator.

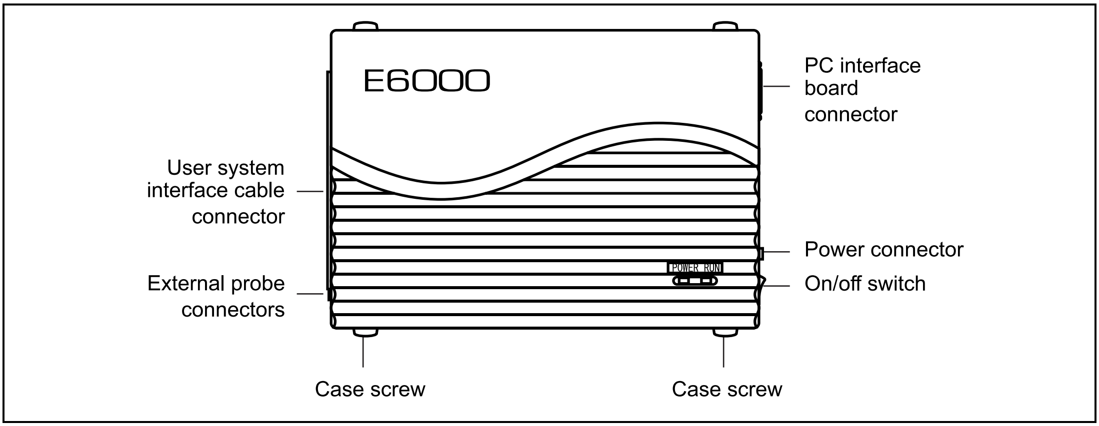


Figure 3.1 E6000 Emulator Connectors

3.1.1 Example of Connecting the User System Interface Cable Head to the User System

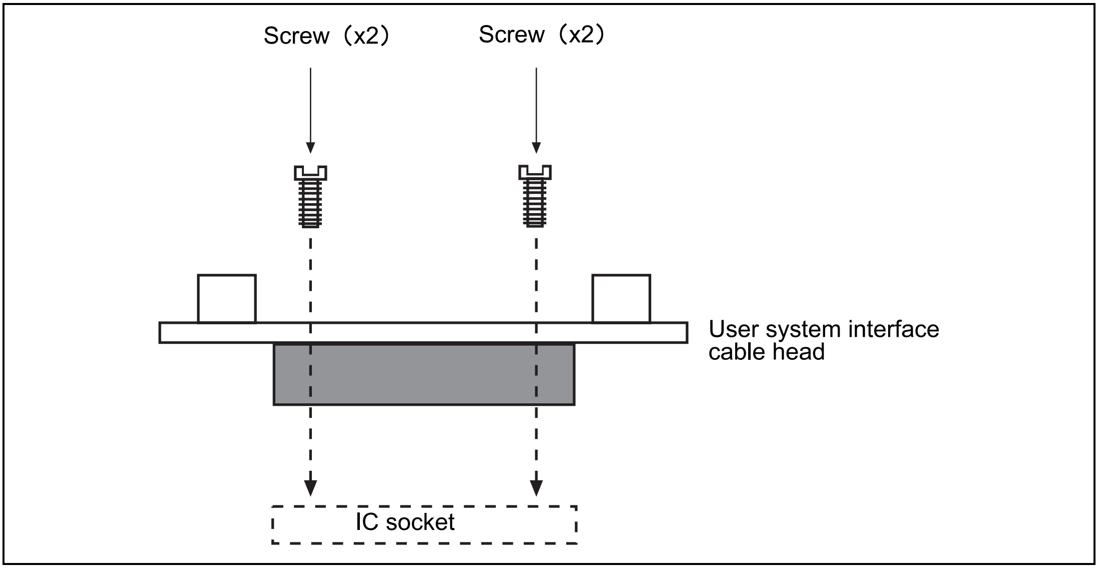


Figure 3.2 Example of Connecting User System Interface Cable Head to User System

- Ensure that all power is off to the E6000 emulator, user hardware, and associated equipment.
- Insert the cable head into the socket on the user system hardware.

Depending upon the package, it may be possible to orientate this cable head in any position on the socket, so care should be taken to correctly identify pin 1 on the E6000 emulator and socket when installing.

- Screw the cable head to the socket with the screws provided. Progressively tighten the screws in the sequence shown in figure 3.3 until all are ‘finger tight’.

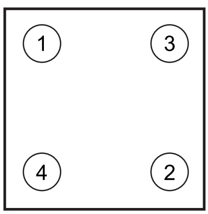


Figure 3.3 Sequence of Screw Tightening

Note: Be careful not to over-tighten the screws as this may result in contact failure on the user system hardware or damage the cable head. Where provided, use the ‘solder lugs’ on the QFP socket to provide extra strength to the E6000 emulator/user system connection.

3.1.2 Plugging the User System Interface Cable Body into the E6000 Emulator

Plug the cable body into the E6000 emulator, taking care to insert it straight, and push it firmly into place.

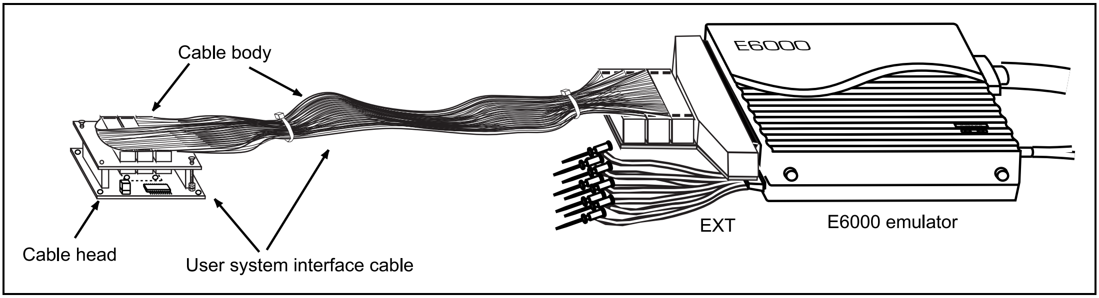


Figure 3.4 Plugging User System Interface Cable Body to E6000 Emulator

3.1.3 Plugging the User System Interface Cable Body into the Cable Head

Plug the cable body into the cable head connected to the user system hardware.

3.2 Power Supply

3.2.1 AC Adapter

The AC adapter supplied with the E6000 emulator must be used at all times.

3.2.2 Polarity

Figure 3.5 shows the polarity of the power-supply plug.

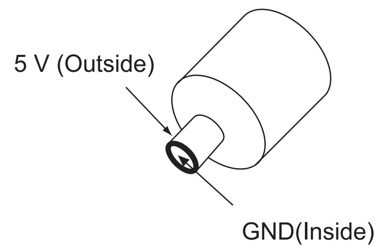


Figure 3.5 Polarity of Power Supply Plug

3.2.3 Power Supply Monitor Circuit

The E6000 emulator incorporates a power supply monitor circuit which only lights the red LED when a voltage higher than 4.75 V is supplied. If this LED does not light, you should check the E6000 emulator voltage level. An input voltage less than 4.75 V could indicate that enough current cannot be supplied to the E6000 emulator.

Note: Use the provided AC adapter for the E6000 emulator.

3.3 SIMM Memory Module

E6000 emulator optional SIMM memory modules are available which provide emulation memory for user code without needing a user system. The optional SIMM memory modules are available in different memory size, but all are partitioned into the areas as shown in table 3.1. These banks can be relocated on page boundaries anywhere in the user area. For details on SIMM Memory Module, refer to the user's manual provided with the SIMM Memory Module.

Table 3.1 Emulation Memory

Type	High-speed emulation memory	Low-speed emulation memory
HS6000EMS21H	128 kbyte × 4 area	512 kbyte × 4 area
HS6000EMS22H		512 kbyte × 12 area

3.3.1 Optional SIMM Memory Module Configuration

The configuration of the optional SIMM memory module is controlled by the mapping RAM. Opening the **Memory Map** dialog box allows you to check which optional SIMM memory module, if any, is installed.

3.4 Hardware Interface

All signals are directly connected to the MCU in the E6000 emulator with no buffering with the exception of those listed in the Supplementary Information:

3.4.1 Signal Protection on the E6000 Emulator

All signals are over/under voltage protected by use of diode arrays. The only exceptions being the AV_{CC} and Vref.

All ports have pull-up resistors except for analog port.

All V_{CC} pins on the cable head assembly are connected together (with the exception of the AV_{CC} pin), and are then monitored by the E6000 emulator to detect powered user system hardware presence.

3.4.2 User System Interface Circuits

The interface circuit between the MCU in the E6000 emulator and the user system has a signal delay of about 8 ns due to the user system interface cable and it includes pull-up resistors. Therefore, high-impedance signals will be pulled up to the high level. When connecting the E6000 emulator to a user system, adjust the user system hardware to compensate for propagation delays.

The following diagrams show the equivalent circuit examples of the interface signals. The interface circuits depend on the MCU type. For details, refer to the supplementary information supplied together with the E6000 emulator.

General Ports:

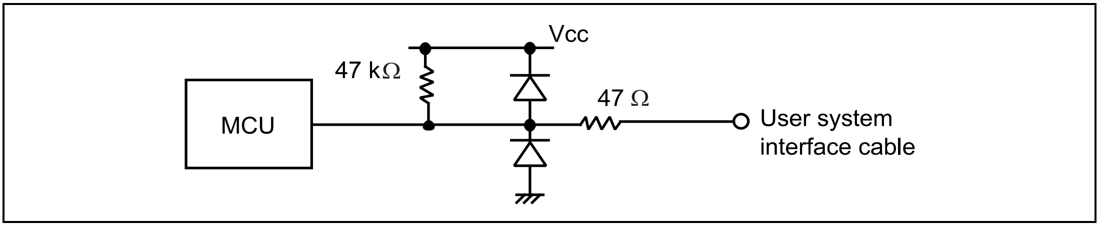


Figure 3.6 User System Interface Circuit for General Ports

Mode Pins (MD3, MD2, MD1, and MD0), WAIT, NMI, and STBY: The WAIT and NMI signals are input to the MCU through the emulator control circuit. The rising/falling time of these signals must be 8 ns/V or less. The STBY signal and mode pins are only monitored. The CPU mode depends on the HDI settings.

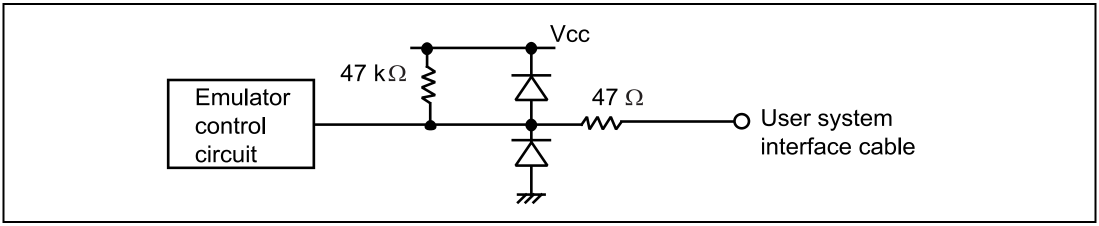


Figure 3.7 User System Interface Circuit for MD2, MD1, MD0, WAIT, NMI, and STBY

RES:

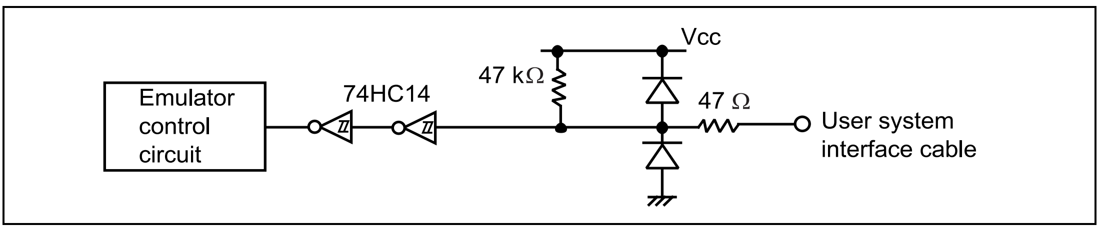


Figure 3.8 User System Interface Circuit for RESET

Analog Port Control Signals:

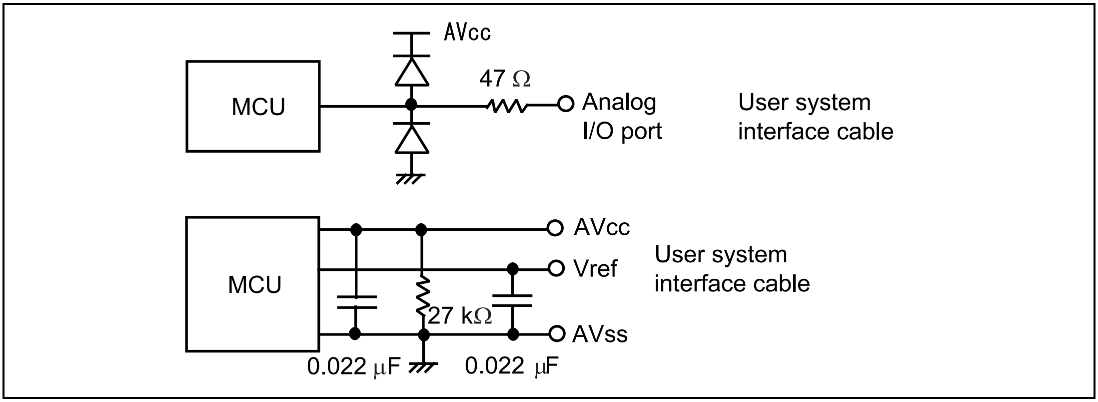


Figure 3.9 User System Interface Circuit for Analog Port Control Signals

IRQ0–IRQ7: The IRQ0 to IRQ7 signals are input to the MCU and also to the trace acquiring circuit. Therefore, the rising and falling time of these signals must be within 8 ns/v or shorter.

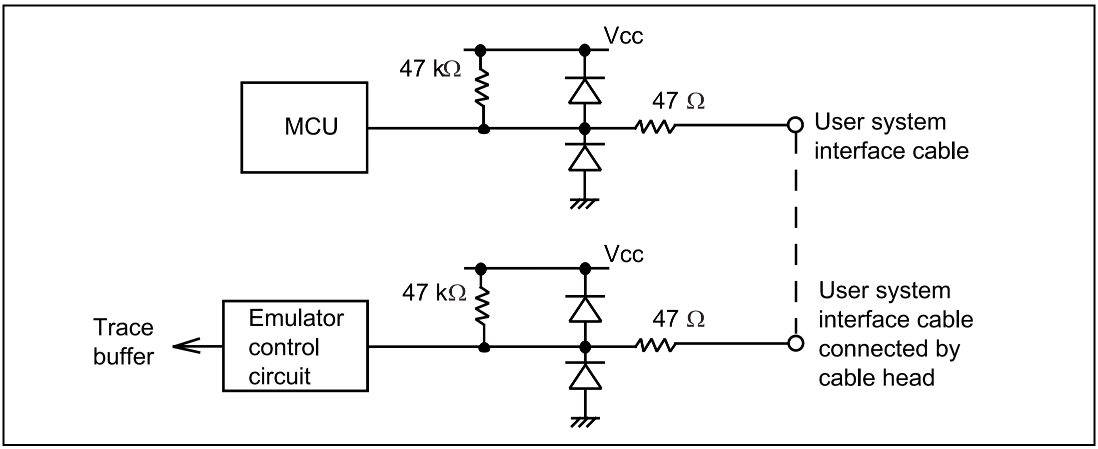


Figure 3.10 IRQ0–IRQ7 User System Interface Circuit

3.4.3 Clock Oscillator

Figure 3.11 shows the oscillator circuit example that has been implemented on the E6000 emulator cable head.

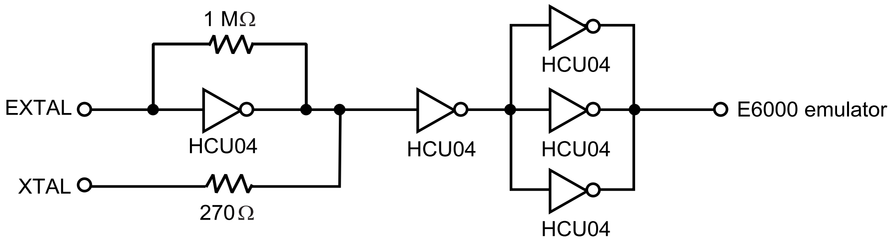


Figure 3.11 Oscillator Circuit

3.4.4 External Probe (EXT)/Trigger Output

An 8-pin connector, marked EXT (on the right under the user system interface cable connector), on the E6000 emulator case accommodates four external probe inputs and two trigger outputs. The pin assignment of this connector is shown in figure 3.12.

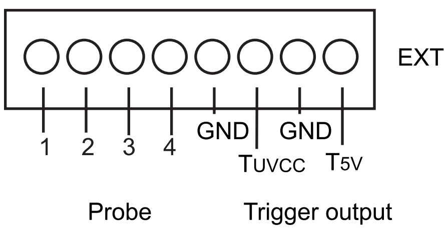


Figure 3.12 External Probe Connector

The interface circuit for the external probes 1-4 is shown in figure 3.13.

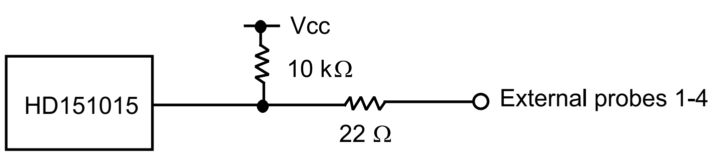


Figure 3.13 Interface Circuit for External Probes 1-4

The trigger output is controlled by event channel 8 and is active low. The trigger output is available as either T5V (within the range from 2.5 V to 5 V; does not depend on the user V_{CC} level) or TUV_{CC} (the user V_{CC} level).

3.4.5 Voltage Follower Circuit

CAUTION

1. Do not connect the user system interface cable to the E6000 emulator without user system connection.

2. Turn on the user system before starting up the E6000 emulator.

A voltage follower circuit is implemented on the E6000 emulator which allows the user system voltage level from the user system to be monitored. This monitored voltage level is automatically supplied to the logic on the E6000 emulator and is derived from the E6000 emulator power supply unit. This means that no power is taken from the user system board.

If no user system interface cable is connected to the E6000 emulator, the E6000 emulator will operate at a specified voltage and all clock frequencies will be available to the user. If the user system interface cable is attached, the E6000 emulator will match the voltage supplied to the user target in all cases; i.e. even when the user V_{CC} is below the operating voltage for the MCU. You must be careful not to select an invalid clock frequency. When the E6000 emulator is connected to the user system and the user system is turned off, the voltage follower circuit output voltage level is 0 V. In this case, the E6000 emulator will not operate correctly.

You can set a user V_{CC} threshold in the range Vcc max. – 0 V by using the E6000 emulator configuration dialog box. If the user V_{CC} drops below this threshold, the **User System Voltage** in the **System Status** window will display `Down`, otherwise `OK` is displayed.

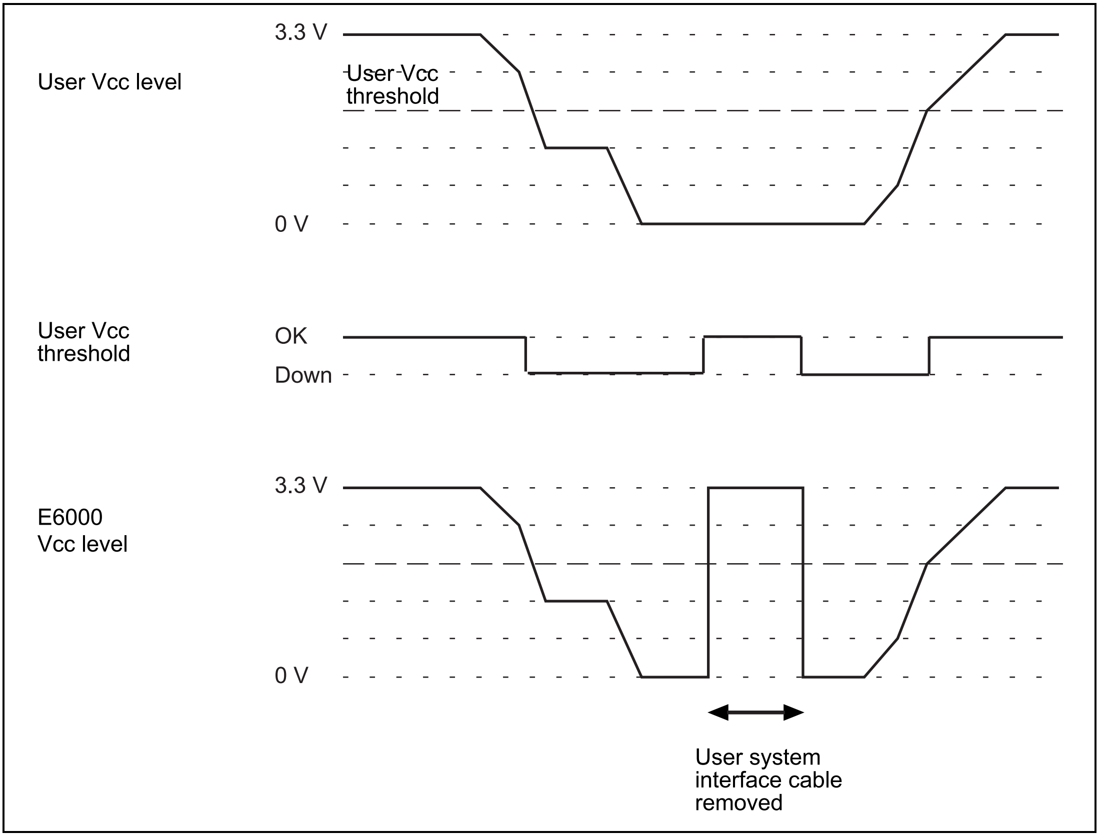


Figure 3.14 Voltage Level Monitoring (Example for Vcc = 3.3 V)

3.5 Differences between MCU and E6000 Emulator

When the E6000 emulator is initialized or the system is reset, there are some differences in the initial values in some of the general registers between the MCU and E6000 emulator as shown in table 3.2.

Table 3.2 Initial Value Differences between MCU and E6000 Emulator

Status	Register	E6000 Emulator	MCU
Power-on and reset command	PC	Reset vector value	Reset vector value
	R0 to R14	H'00000000	Undefined
	R15 (SP)	Reset vector value	Reset vector value
	SR	H'000000F0	I0 to I3 bits are 1
		I0 to I3 bits are 1	Reserved bit is 0
			Others are undefined
	PR	H'00000000	Undefined
	VBR	H'00000000	H'00000000
	GBR	H'00000000	Undefined
	MACH	H'00000000	Undefined
	MACL	H'00000000	Undefined
	Others	Value before reset	Undefined

Please refer to the supplied supplementary information for details of the protection circuit used on the I/O ports of the E6000 emulator.

3.5.1 A/D Converter and D/A Converter

Due to the use of a user system interface cable, there is a slight degradation in the A/D and D/A conversion than that quoted in the Hardware Manual for the MCU being emulated.

Section 4 Tutorial

The following describes a sample debugging session, designed to introduce the main features of the E6000 emulator used in conjunction with the HDI software. Therefore, the SIMM memory module must be installed in the E6000 emulator.

The tutorial is designed to run in the E6000 emulator's resident memory so that it can be used without connecting the E6000 emulator to a user system.

The tutorial assumes that the SH7010 E6000 is used. When using another type of E6000 emulator, change the file and directory names to your target ones.

4.1 Introduction

The tutorial is based on a simple C program.

Before reading this chapter:

- Set up the E6000 emulator from the HDI software. See section 2, Setting Up. You do not need to connect the E6000 emulator to a user system to use this tutorial.
- Make sure you are familiar with the architecture and instruction set of the MCU. For more information, refer to the Hardware Manual and the Programming Manual for the target MCU.

The tutorial program starts the Direct Memory Access Controller (DMAC) by the MTU and transfers Name ("Hitachi, Ltd.") string to the memory start address Destination Address (H'200000). The source program (`tutorial.C`), and the object file in the Sysrof format (`tutorial.abs`) are provided in the HDI installation disk.

4.2 Starting HDI

To start the HDI:

- Select **HDI for E6000 SH7010** from the **Start** menu.

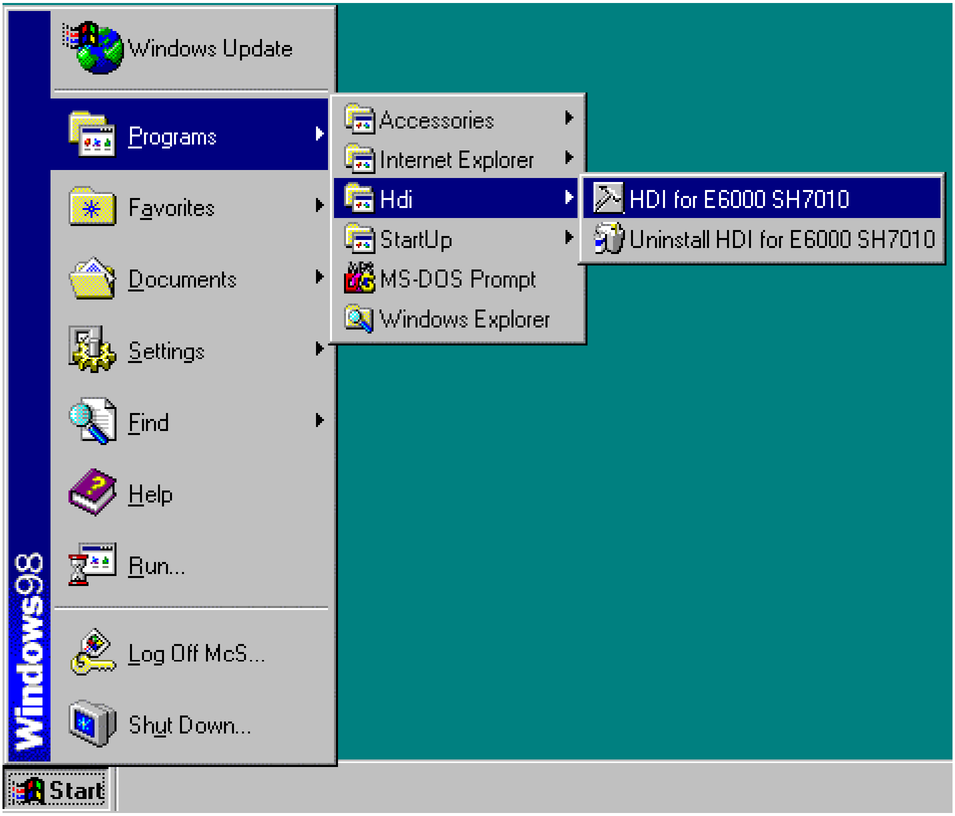


Figure 4.1 HDI Start Menu

4.2.1 Selecting the Target Platform

The HDI has extended functions for supporting multiple target platforms, and if your system is set up for more than one platform you will first be prompted to choose the target platform.

Note that you can change the target platform at any time by choosing **Select Platform…** from the **Setup** menu. If you have only one platform installed, this menu option will not be available.

The tutorial selects the E6000 SH7016/7017 Emulator.

When the emulator has been successfully set up, the **HDI** window will be displayed, with the message `Link up` in the status bar.

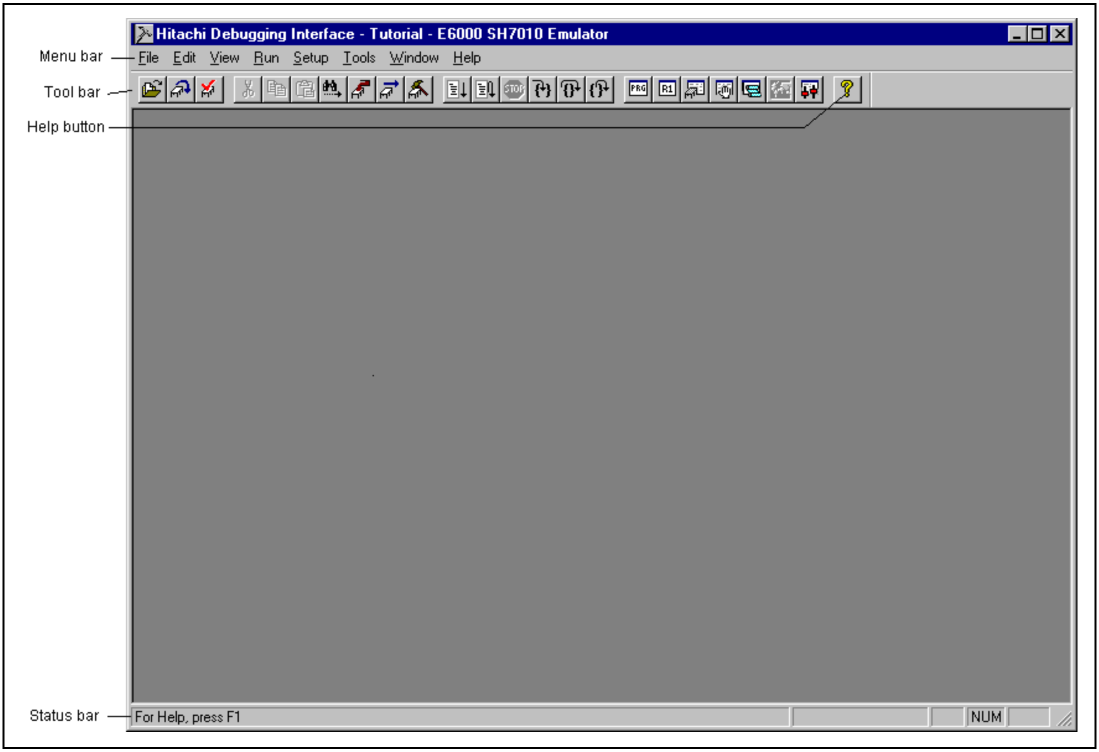


Figure 4.2 HDI Window

For the key features of HDI, see HDI User's Manual. For the functions specialized for the E6000 emulator, refer to the on-line help.

Menu Bar: Gives you access to the HDI commands for setting up the E6000 emulator and using the HDI debugging functions.

Toolbar: Provides convenient buttons as shortcuts for the most frequently used menu commands.

Status Bar: Displays the status of the E6000 emulator. For example, progress information about downloads, snapshots of address bus in run mode.

Help Button: Activates context sensitive help about any feature of the HDI user interface.

4.3 Setting up the E6000 Emulator

Before downloading a program to the E6000 emulator, you first need to set up the target MCU conditions. The following items need to be configured:

- The device type
- The operating mode
- The clock source
- The user signals
- The memory map

The following sections describe how to set up the E6000 emulator as appropriate for the tutorial program.

4.3.1 Configuring the Platform

To set up the target configuration:

- Choose **Configure Platform…** from the **Setup** menu to set up the conditions for the selected platform.
- The following dialog box will be displayed:

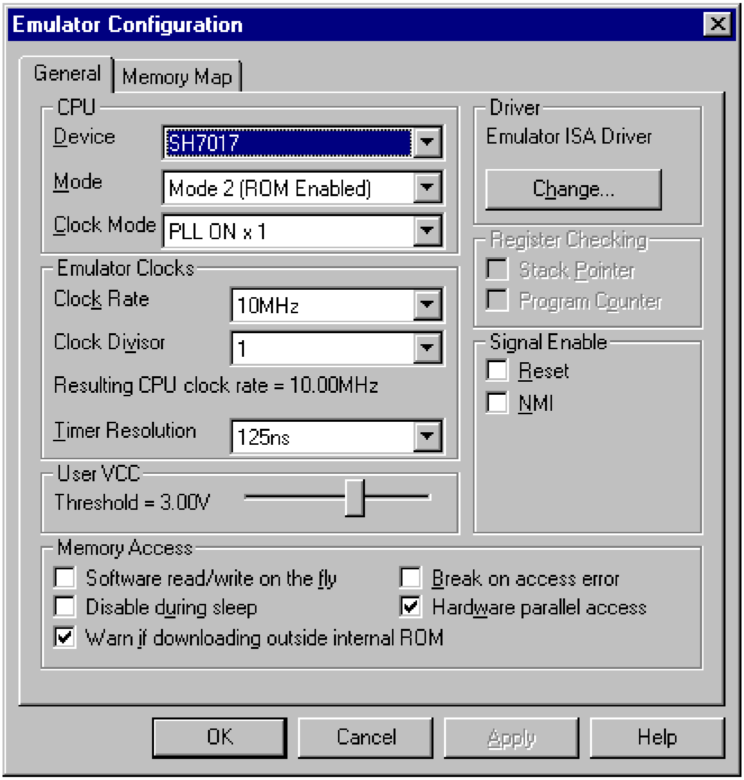


Figure 4.3 Emulator Configuration Dialog Box (General)

- Set up the options as shown in table 4.1.

Table 4.1 Configuration Options

Option	Value (Depending on Evaluation Chip)
Device	SH7017
Mode	2 (with on-chip ROM)
Clock mode	PLL ON x 1
Clock rate	10 MHz
Clock divisor	1
Timer resolution	125 ns
User system voltage monitoring level (User VCC Threshold)	3.00 V
All other options	Default

4.3.2 Mapping the Memory

After you have selected the device and mode in the **Configuration Dialog Box**, the HDI automatically maps the E6000 emulator memory for the device and mode you have selected.

- To display the current memory map, click the **Memory Map** tag.

The dialog box shown in figure 4.4 is displayed.

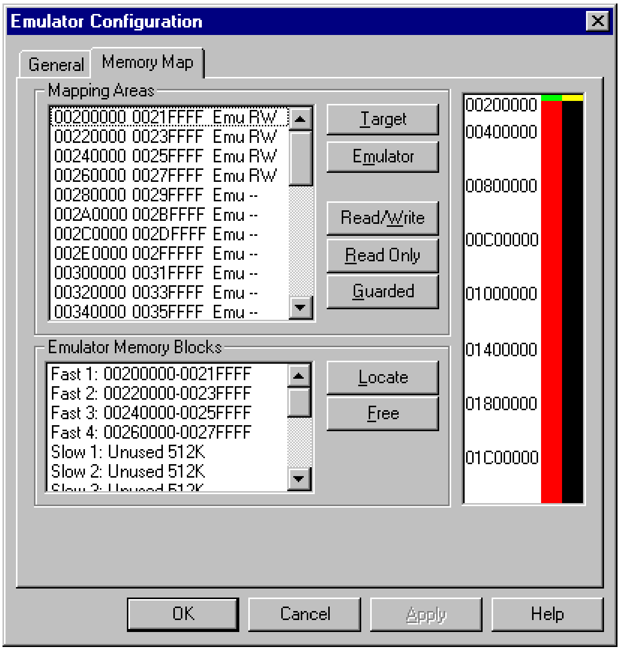


Figure 4.4 Emulator Configuration Dialog Box (Memory Map)

Table 4.2 lists the three memory types available in the E6000 emulator.

Table 4.2 Memory Types

Memory Type	Description
On-chip	Not displayed.
Target	Accesses the memory on the user system hardware.
Emulator	Accesses the optional SIMM memory module.

Table 4.3 lists the three access types.

Table 4.3 Access Types

Access Type	Description
Read-write	RAM
Read-only	ROM
Guarded	No access allowed

For this tutorial, we can use the default mapping, but you can edit the mapping as follows:

Note: The memory map of internal ROM, internal RAM, internal I/O, and reserved area differ depending on the target MCU. For details, refer to the hardware manual of the MCU.

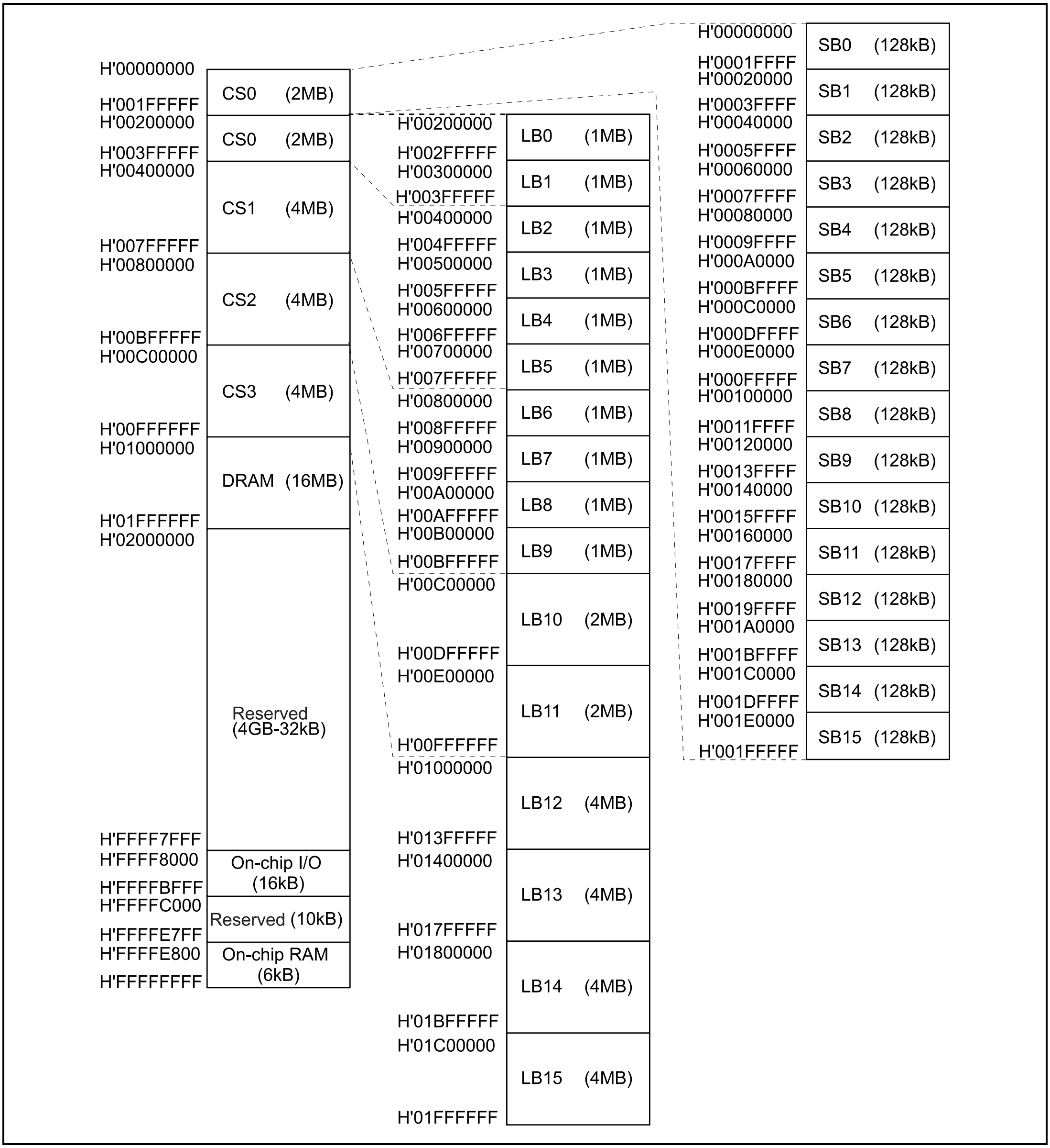


Figure 4.5 Memory Block in Extended Mode without ROM

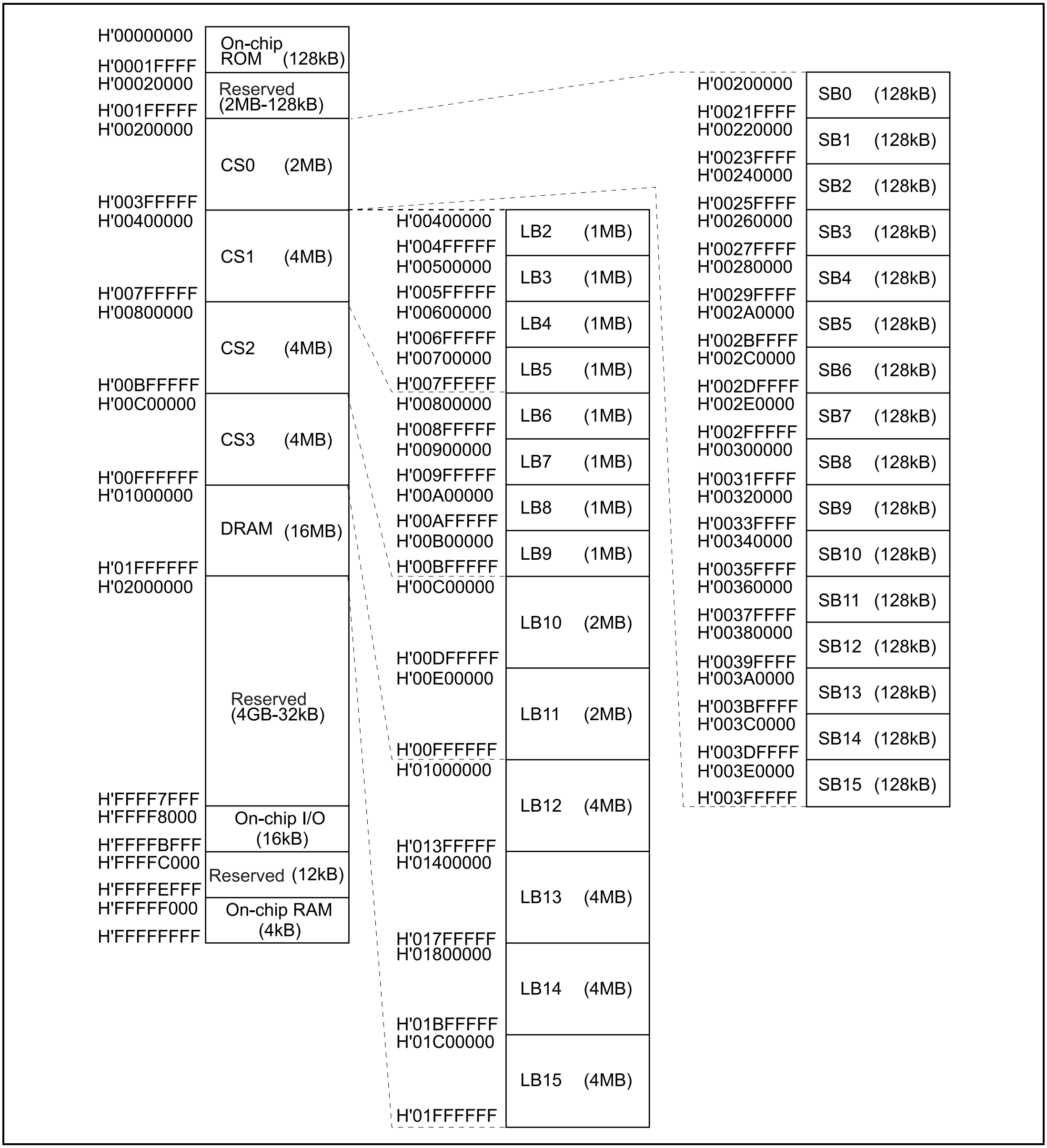


Figure 4.6 Memory Block in Extended Mode with ROM

- Do not modify memory allocation in the **Emulator Configuration** Dialog Box. .
- Click **OK** to close the dialog box.

The device type, operating mode, and memory map settings have completed.

4.4 Downloading the Tutorial Program

After the E6000 emulator is set up, you can download the object program you want to debug.

4.4.1 Loading the Object File

First load the Sysrof-format object file, as follows:

- Choose **Load Program** from the **File** menu, or click the **Load Program** button in the toolbar.

- Select the file `tutorial.abs`, in the `tutorial` directory, and click **OK**.

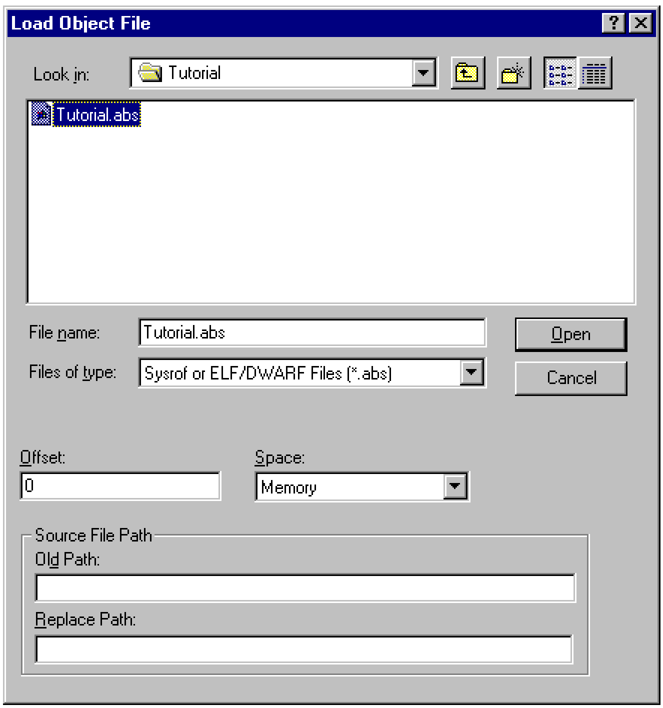


Figure 4.7 Load Object File Dialog Box

- The file `tutorial.abs` is created only when the HDI is installed in the default directory. If the HDI is installed in another directory, the file `tutorial.abs` is created when the file `tutorial.bat` is executed. Modify `tutorial.bat` or `tutorial.sub` according to the system environment.

When the file has been loaded, the message box shown in figure 4.8 displays information about the memory areas that have been filled with the program code.

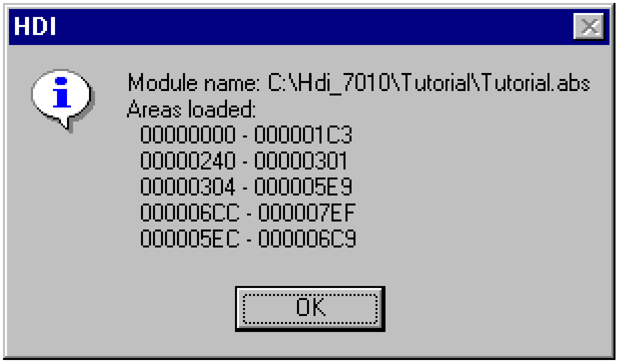


Figure 4.8 HDI Information Message Box

- Click **OK** to continue.

The program has been loaded into the on-chip ROM.

4.4.2 Displaying the Program Listing

The HDI allows you to view a program at source level and in assembly-language mnemonic.

- Choose **Program Window…** from the **View** menu, or click the **Program Window** button in the toolbar.

You will be prompted for the C source file corresponding to the object file you have loaded.

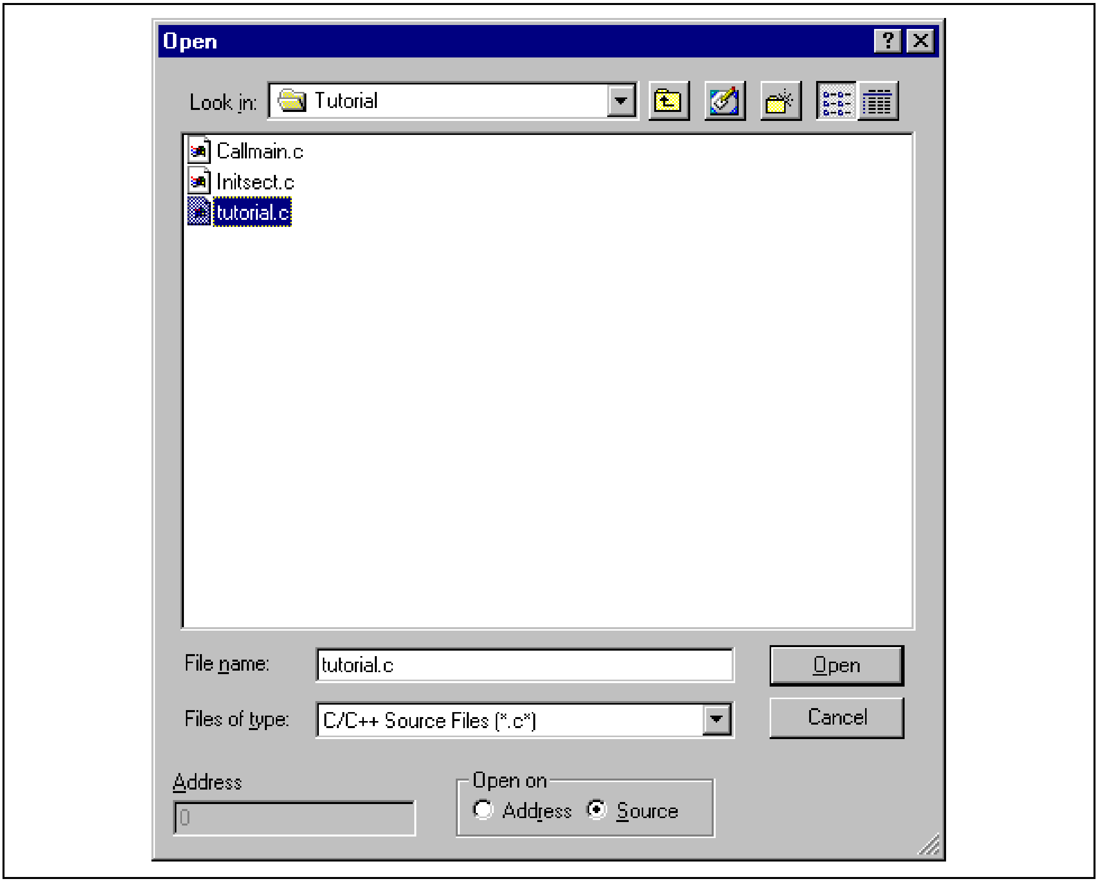


Figure 4.9 Open Dialog Box

- Select `tutorial.c` and click **OK** to display the program window.

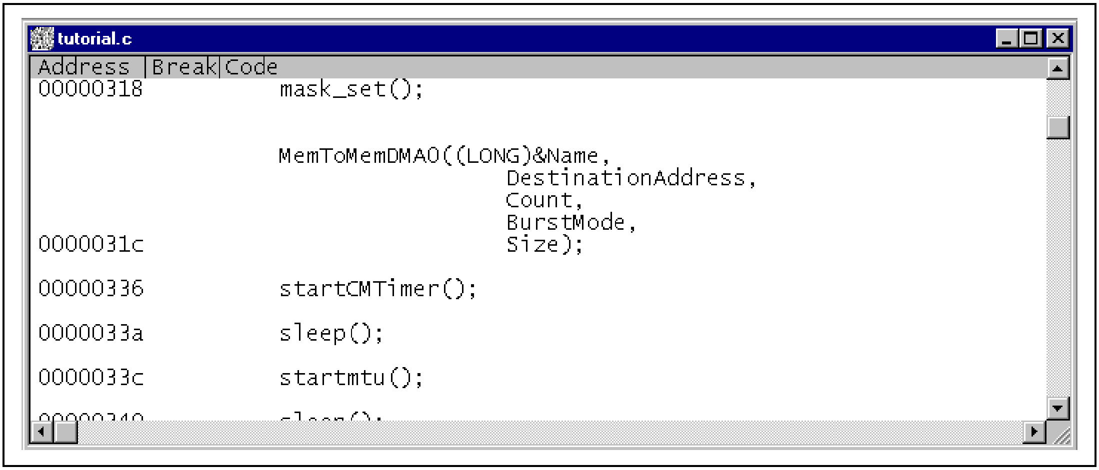


Figure 4.10 Tutorial Program Window

- If necessary, choose **Font** option from the **Customize** submenu on the **Setup** menu to choose a font and size suitable for your host computer.

Initially the program window opens showing the beginning of the `main` program, but you can scroll through the program with the scroll bars to see the definitions and include statements.

4.5 Using Breakpoints

The simplest debugging aid is the PC break, which lets you halt execution when a particular point in the program is reached. You can then examine the state of the MCU and memory at that point in the program.

4.5.1 Setting a PC Break

The program window provides a very simple way of setting a PC break. For example, set a PC break at address H'336 as follows:

- Double-click in the **Break** column on the line containing address H'336.

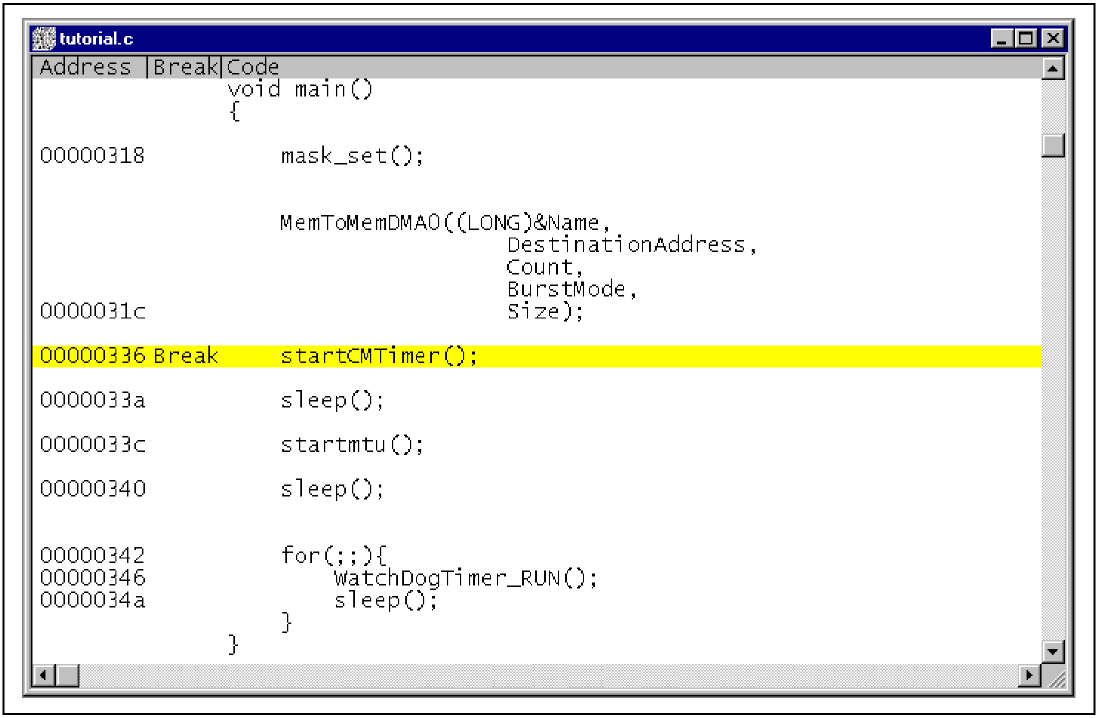


Figure 4.11 Setting a Breakpoint

The word `Break` will be displayed there to show that a PC break is set at that address. Although not performed in this tutorial, double-clicking repeatedly in the **Break** column can change the display in the cyclic order shown below to set the event for measuring the execution time between events (`+Timer`: start time measurement; `-Timer`: stop time measurement), point-to-point trace (`+Trace`: start trace; `-Trace`: temporarily stop trace), or trace stop (`TrStop`: stop trace). When `-Trace` is followed by `+Trace`, trace is resumed. However, when `-Trace` is followed by `TrStop`, trace will not resume even after `+Trace` appears.

(Blank) → `Break` → `+Timer` → `-Timer` → `+Trace` → `-Trace` → `TrStop` → (Blank) → ...

4.5.2 Executing the Program

To run the program from the address pointed to by the reset vector:

- Choose **Go Reset** from the **Run** menu, or click the **Go Reset** button in the toolbar.

The program will be executed up to the PC break you inserted, and the statement will be highlighted in the program window to show that the program has halted.

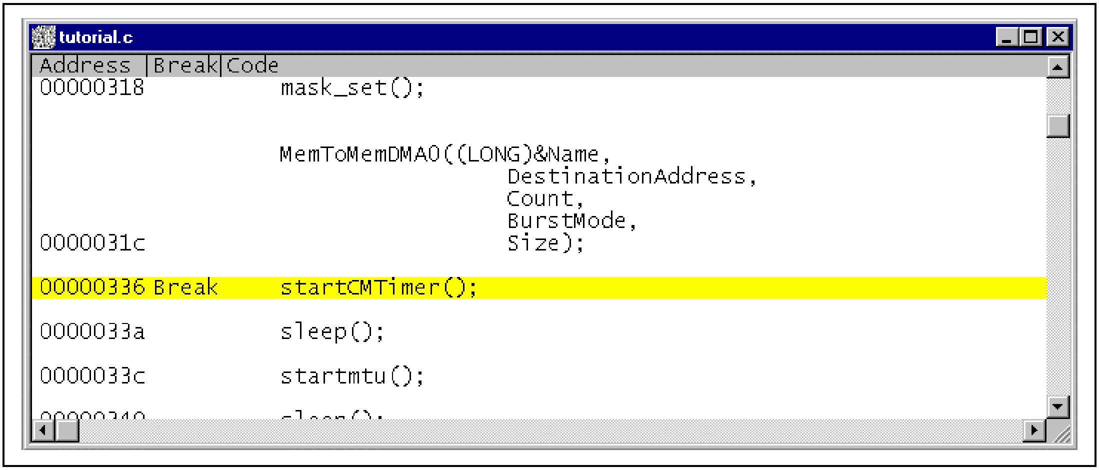
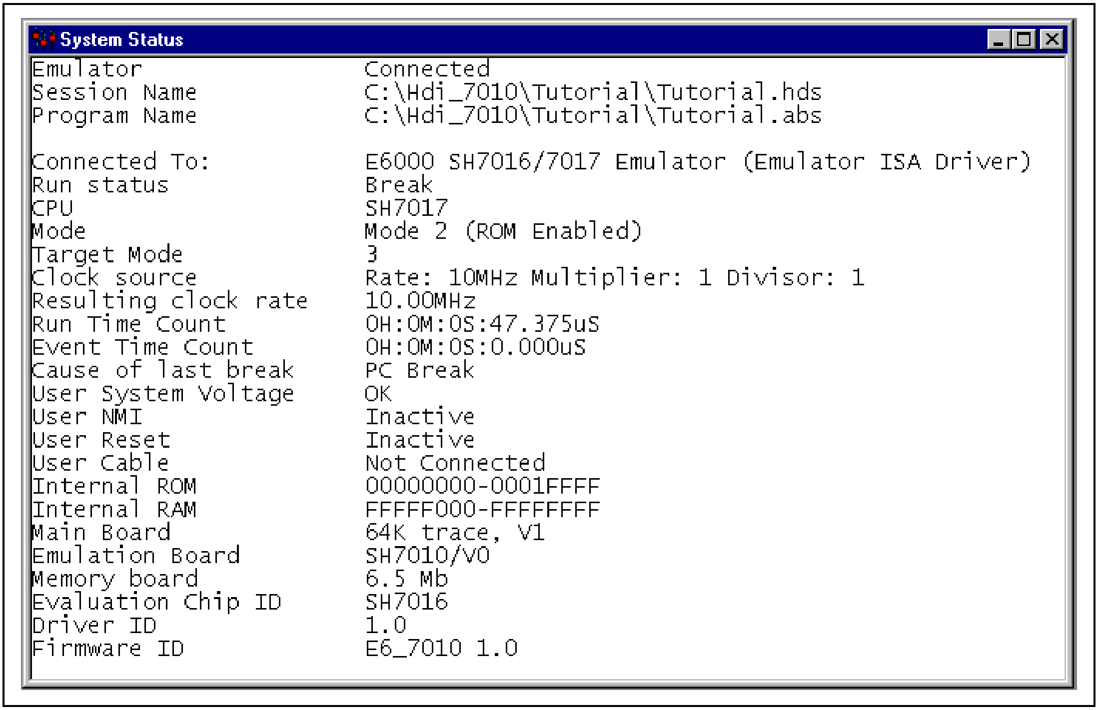


Figure 4.12 Program Break

The message `Break=PC Break` is displayed in the status bar to show the cause of the break.

You can also see the cause of the last break in the **System Status** window.

- Choose **Status Window** from the **View** menu, or click the **Status Window** button in the toolbar:



System Status

Emulator	Connected
Session Name	C:\Hdi_7010\Tutorial\Tutorial.hds
Program Name	C:\Hdi_7010\Tutorial\Tutorial.abs
Connected To:	E6000 SH7016/7017 Emulator (Emulator ISA Driver)
Run status	Break
CPU	SH7017
Mode	Mode 2 (ROM Enabled)
Target Mode	3
Clock source	Rate: 10MHz Multiplier: 1 Divisor: 1
Resulting clock rate	10.00MHz
Run Time Count	0H:0M:0S:47.375uS
Event Time Count	0H:0M:0S:0.000uS
Cause of last break	PC Break
User System Voltage	OK
User NMI	Inactive
User Reset	Inactive
User Cable	Not Connected
Internal ROM	00000000-0001FFFF
Internal RAM	FFFFF000-FFFFFFFF
Main Board	64K trace, V1
Emulation Board	SH7010/V0
Memory board	6.5 Mb
Evaluation Chip ID	SH7016
Driver ID	1.0
Firmware ID	E6_7010 1.0

Figure 4.13 System Status Window

The `Cause of last break` line shows that the break was a PC break. The `Run Time Count` line shows that the user program executing time (from user program start to break) is 47.375 µs. The timer resolution of the event time (set by `+Timer` and `–Timer`) and the run time timer's resolution is decided by the **Timer Resolution** option in the target **Configuration** dialog box. When using a small resolution (e.g. 20 ns) for a long time measurement, the inaccuracy may be large. Select the timer resolution suitable for the length of measurement time.

4.5.3 Examining Registers

While the program is halted you can refer to the contents of the MCU registers. These are displayed in the **Registers** window.

- Choose **Registers Window** from the **View** menu, or click the **Registers Window** button in the toolbar:

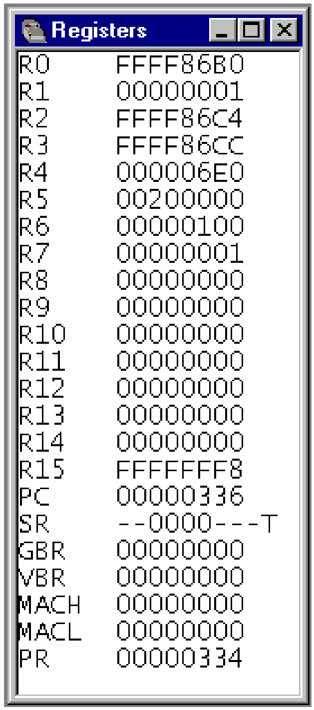


Figure 4.14 Registers Window

As expected, the value of the program counter, PC, is the same as the highlighted statement, `H'336`.

(Note: The values of the other registers may differ from those shown in the above figure.)

You can also change the registers from the **Registers** window. For example, to change the value of the PC:

- Double-click `PC` in the **Registers** window.

The **Register-PC** dialog box allows you to edit the value.

RENESAS

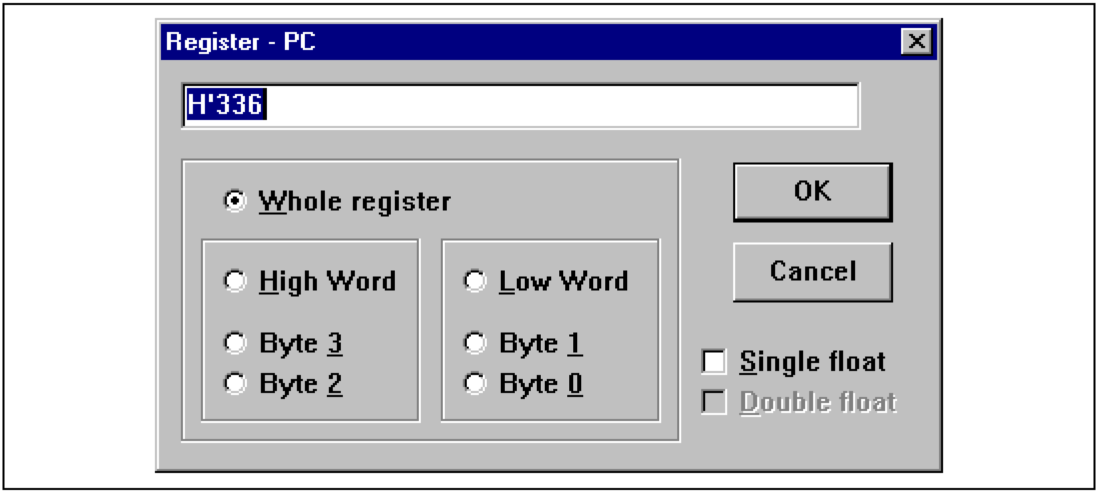


Figure 4.15 Register Dialog Box

- Edit the value to `H'31C`, the address of the previous statement, and click **OK**.

The highlighted bar will move to the previous statement in the program window to show the new PC value.

- Choose **Go** from the **Run** menu, or click the **Go** button in the toolbar, to execute up to the breakpoint again.

RENESAS

4.5.4 Reviewing the Breakpoints

You can see a list of all the breakpoints set in the program in the **Breakpoints** window.

- Choose **Breakpoints Window** from the **View** menu, or click the **Breakpoint Window** button in the toolbar:

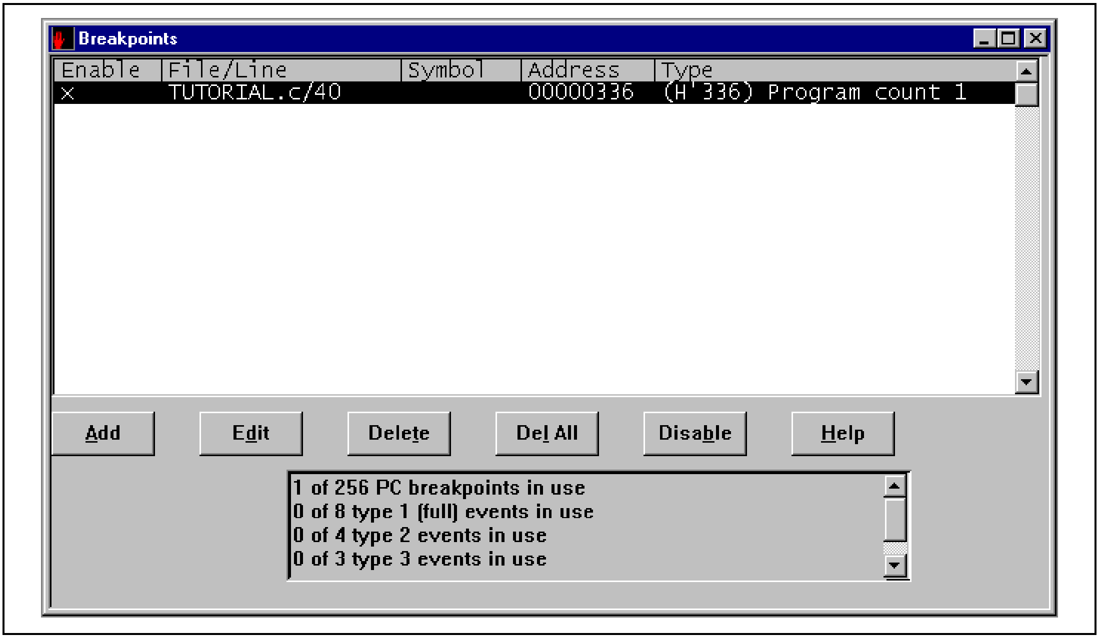


Figure 4.16 Breakpoints Window

The **Breakpoints** window also allows you to enable or disable breakpoints, define new breakpoints, and delete breakpoints.

Before proceeding, remove the breakpoint as follows:

- Highlight the breakpoint in the **Breakpoints** window and click **Delete**.
- Close the **Breakpoints** window.

4.6 Examining Memory and Variables

You can monitor the behavior of a program by examining the contents of an area of memory, or by displaying the values of variables used in the program.

4.6.1 Viewing Memory

You can view the contents of a block of memory in the **Memory** window.

For example, to view the memory corresponding to the array `Name` in ASCII:

- Choose **Memory Window…** from the **View** menu, or click the **Memory Window** button in the toolbar.

- Enter `Name` in the **Address** field, and set **Format** to `ASCII`.

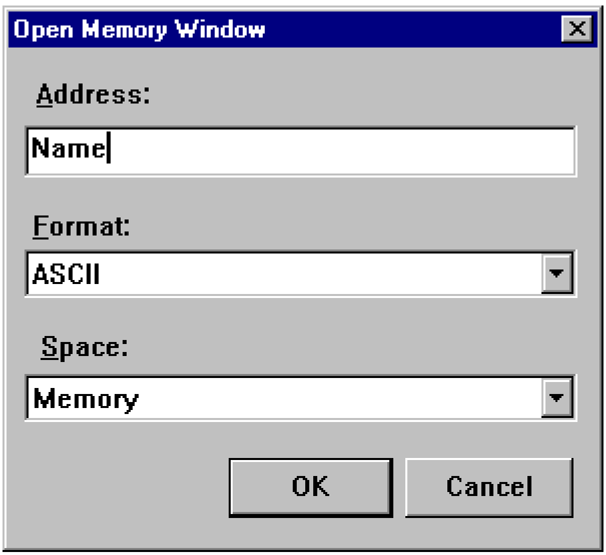


Figure 4.17 Open Memory Window Dialog Box

- Clicking **OK** opens the **Memory** window showing the specified area of memory and enables to check the contents of the memory block.

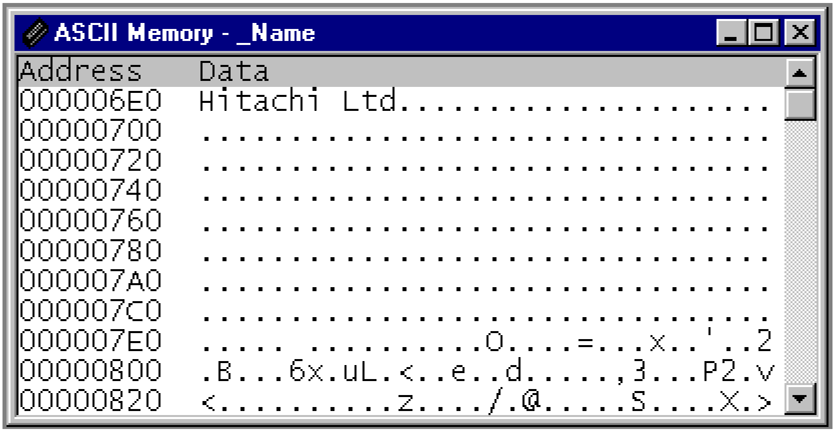


Figure 4.18 Memory Window (ASCII)

4.6.2 Watching Variables

As you step through a program, it is useful to be able to watch the values of variables used in your program, to verify that they change in the way that you expected.

For example, set a watch on the char variable `Name`, declared at the beginning of the program, using the following procedure:

- Scroll up in the program window until you see the line:

```
const char Name [0x100] = “Hitachi, Ltd”
```

- Click to position the cursor to the left of `Name` in the program window.
- Click in the program window with the right mouse button to display a pop-up menu, and choose **Add Watch**.

The **Watch** window will display the variable.

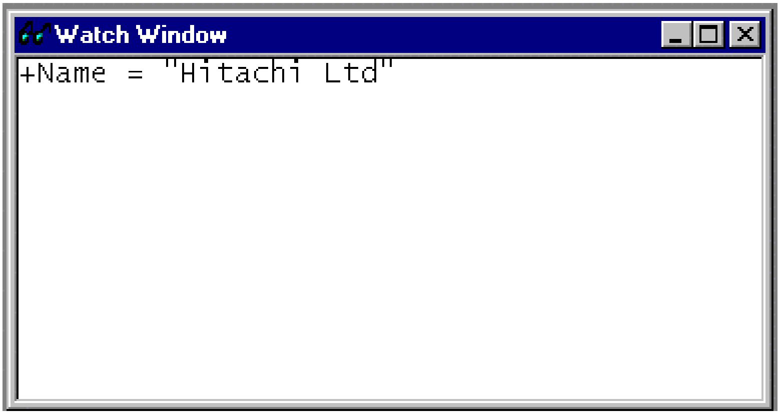


Figure 4.19 Watch Window (After Adding Variables)

You can double-click the + symbol to the left of any symbol in the **Watch** window to expand it and display the individual elements in the array.

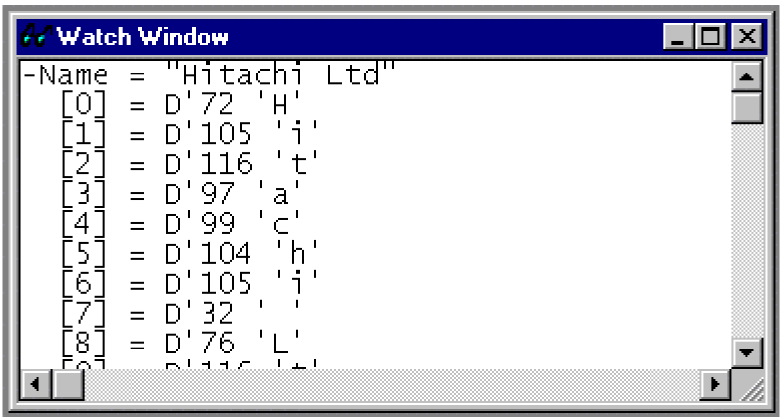


Figure 4.20 Watch Window (Symbol Expansion)

4.7 Stepping Through a Program

The E6000 emulator provides a range of options to perform step execution by executing an instruction or statement at a time. The alternative step commands listed in table 4.4 are provided.

Table 4.4 Step Commands

Command	Description
Step in	Executes every statement, including statements within functions.
Step Over	Executes a function call in a single step.
Step out	Exits a function and stops at the next statement of the calling program.
Step…	Allows you to step repeatedly the specified number of times.

4.7.1 Single Stepping

- Set a PC break at `H'318`.
- Select **Go Reset** from the **Run** menu or click the **Go Reset** button in the toolbar.

The statement of `mask_set()` will be highlighted.

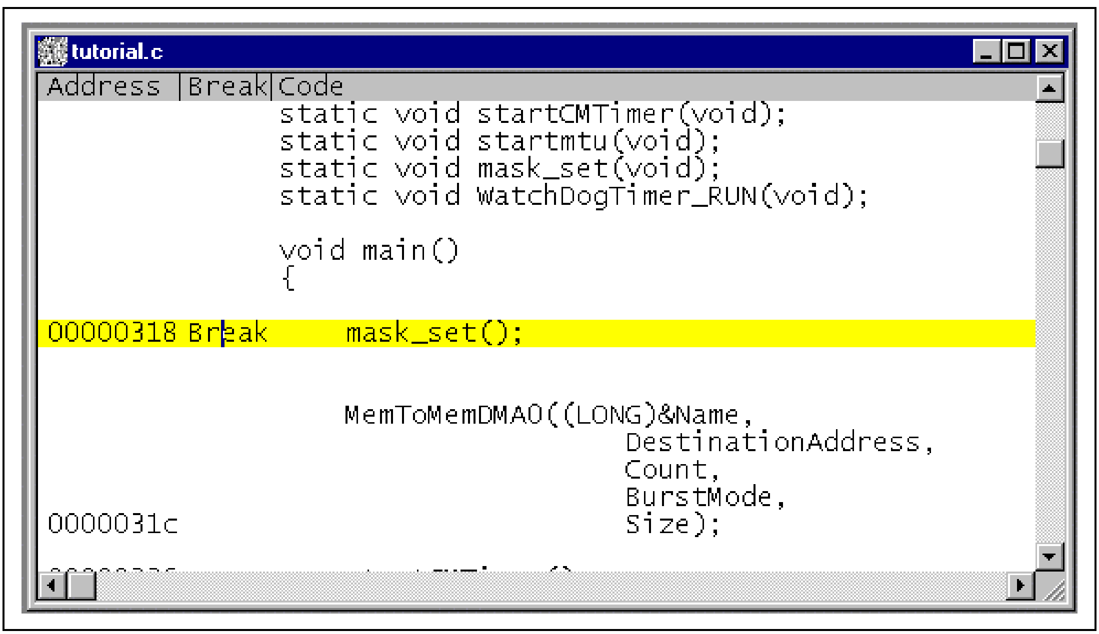


Figure 4.21 Program Window after Executing the Step In Command (1)

- Choose **Step In** from the **Run** menu, or click on the **Step In** button in the toolbar, to step through the `mask_set()` statement.

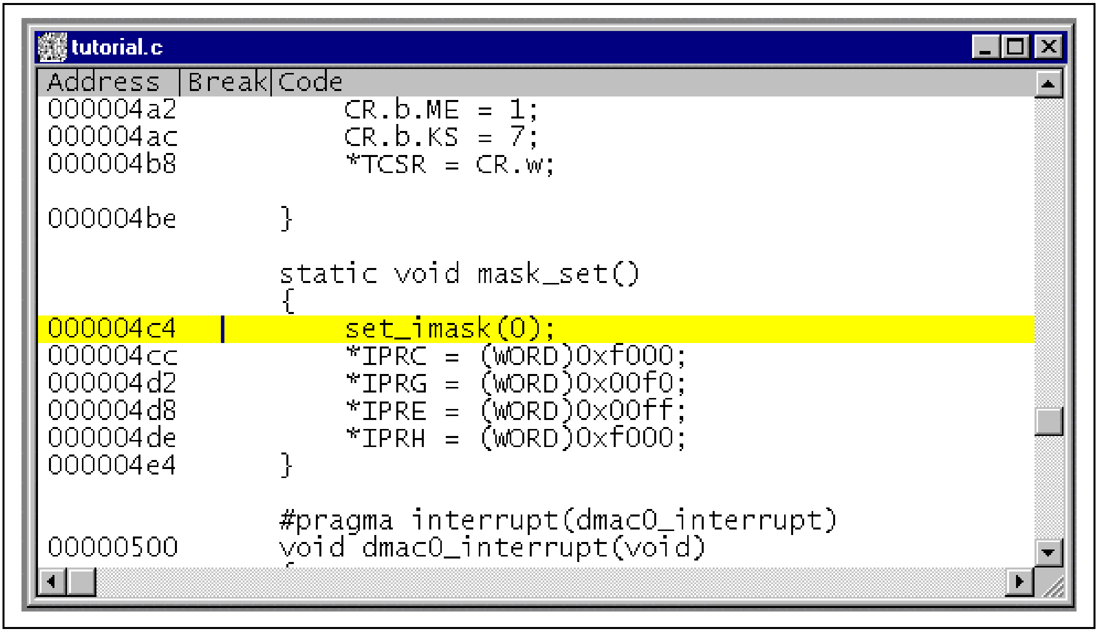


Figure 4.22 Program Window after Executing the Step In Command (2)

Exit the function, and back to the next statement in the `main` program, by choosing **Step Out** from the **Run** menu, or clicking the **Step Out** button.

RENESAS

Address `H'31c` will be highlighted showing that the emulator has exit from the function.

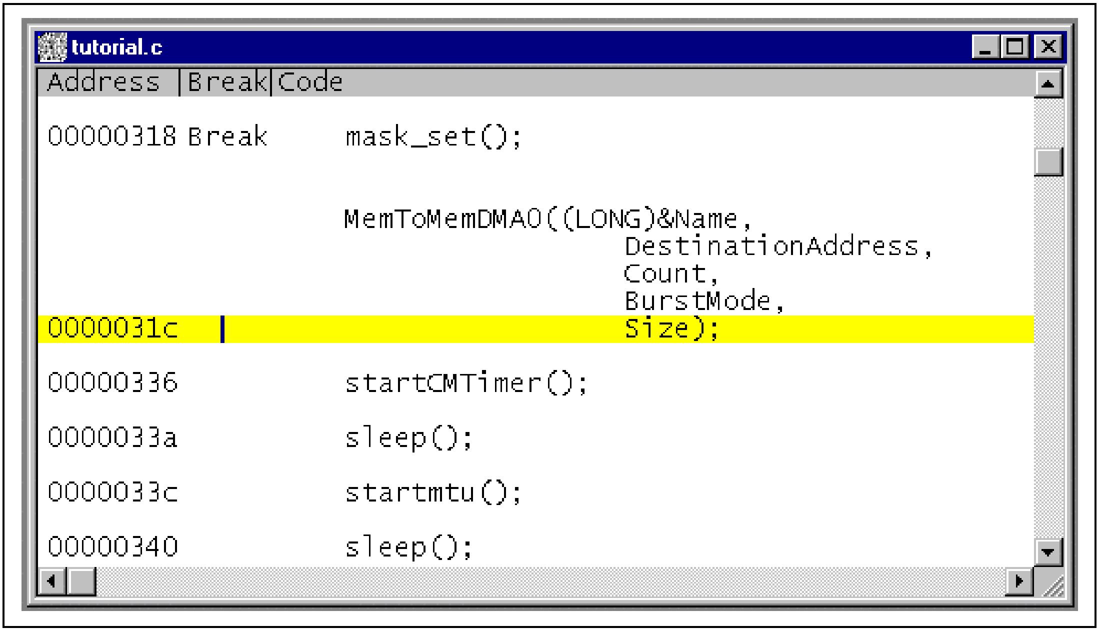


Figure 4.23 Program Window after Executing the Step Out Command

4.7.2 Stepping Over a Function

The **Step Over** command executes a function, without single-stepping through the body of the function, and stops at the next statement in the `main` program.

- Choose **Step Over** from the **Run** menu, or click the **Step Over** button in the toolbar.

The program executes the `MemToMemDMA0` function and stops at the beginning of the next address, `H'336`.

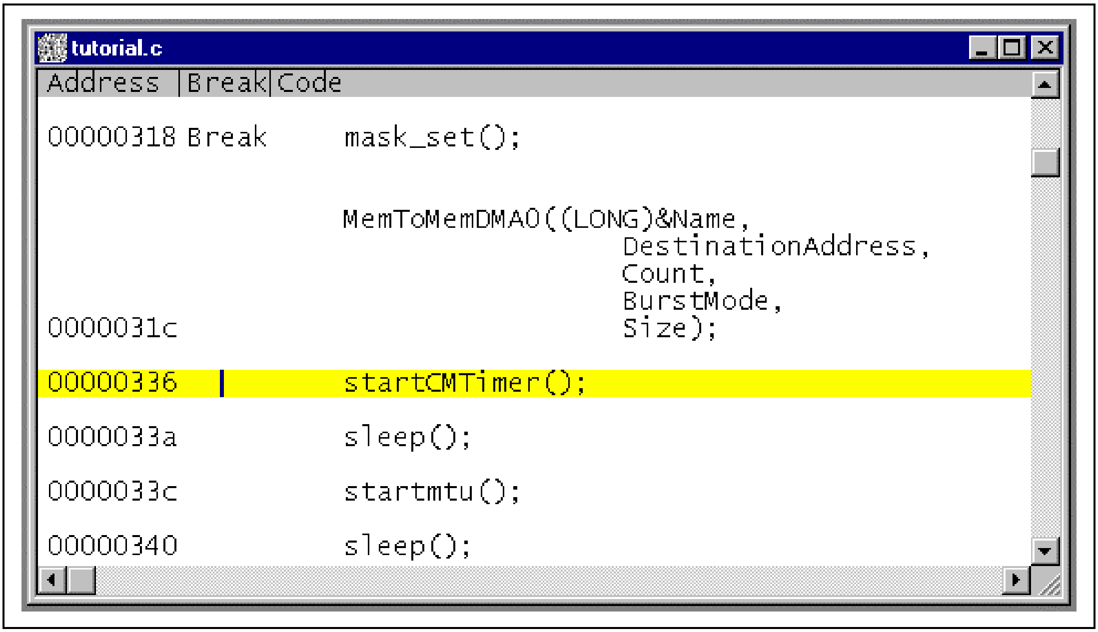


Figure 4.24 Program Window after Executing the Step Over Command

4.7.3 Displaying Local Variables

You can display local variables of a function using the **Local Variables** window. For example, we will examine the local variables in the function `startCMTimer`.

- Choose **Step In** from the **Run** menu to start executing the function `startCMTimer`, or click the **Step In** button in the toolbar one time.

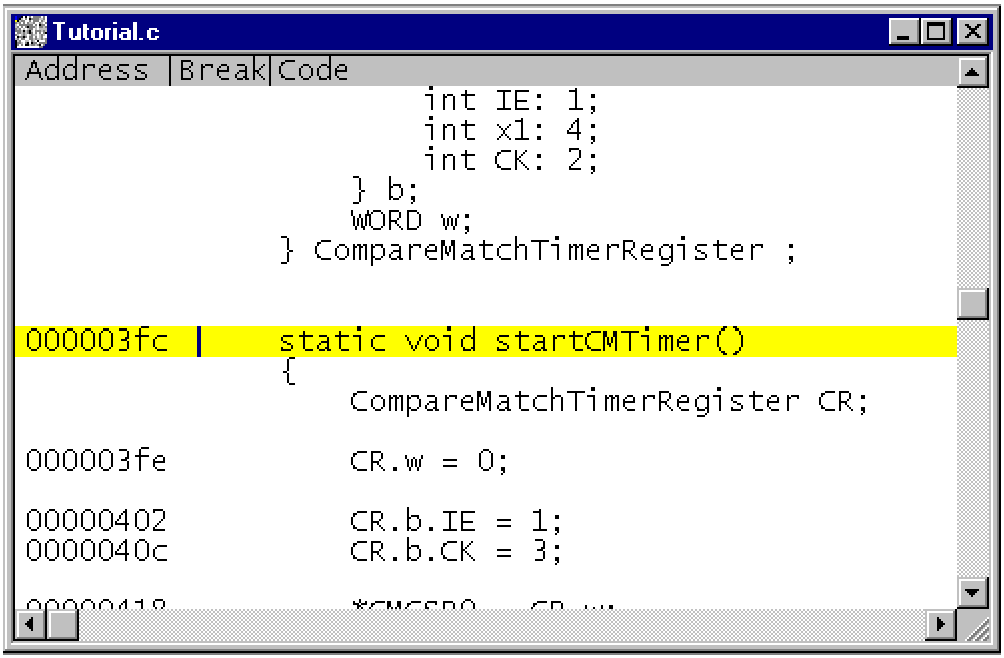


Figure 4.25 Program Window after Executing the Step In Command (4)

- Open the **Locals** window by choosing **Local Variable Window** from the **View** menu.

Initially, the **Locals** window will not show correct values because the local variables declarations have not yet been executed.

- Choose **Step In** from the **Run** menu or click the **Step In** button in the toolbar to perform step execution one time.

The **Locals** window will now show the local variables and their values.

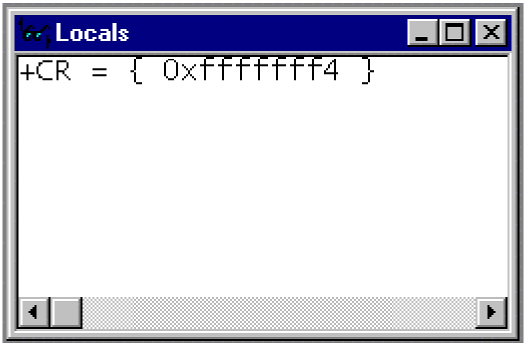


Figure 4.26 Displaying Local Variables

- Double-click the + symbol to the left of the variable `CR` in the **Locals** window to display the individual elements of the array `CR`.

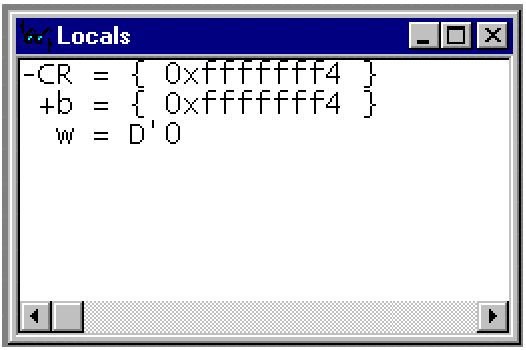


Figure 4.27 Displaying Local Variables (Elements in an Array)

- Choose **Step Out** from the **Run** menu to return to the `main` program, or click the **Step Out** button in the toolbar and return to the `main` program.

4.8 Using the Complex Event System

So far in this tutorial we have monitored the behavior of the program by observing the contents of an area of memory in the **Memory** window, or the values of variables in the **Watch** and **Locals** windows.

Sometimes the action of a program is too complex to allow us to do this. Using the emulator's complex event system, you can detect the timing when a program accesses address `H'3c4`.

4.8.1 Defining an Event Using the Complex Event System

Now define an event using the complex event system to monitor a part of the program as follows:

- Choose **Breakpoint Window** from the **View** menu to display the **Breakpoints** window, or click the **Breakpoint Window** button in the toolbar.

- Click **Add** to define a new breakpoint.

The **Select Event Type** dialog box allows you to define the event type.

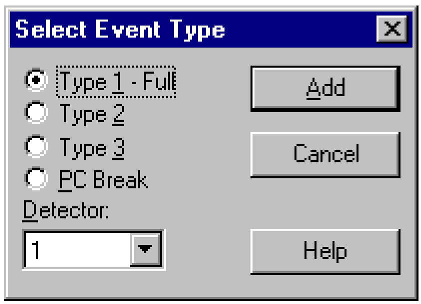


Figure 4.28 Select Event Type Dialog Box

Select **Type1-Full** and click **Add**.

The **Breakpoint/Event Properties** dialog box allows you to define the breakpoint's properties.

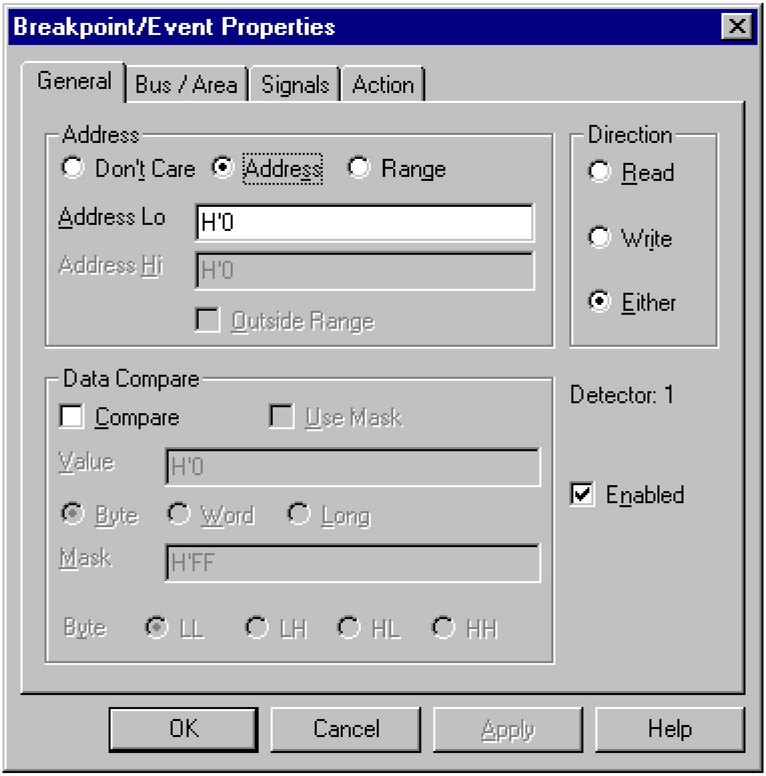


Figure 4.29 Breakpoint/Event Properties Dialog Box

- Select **Address** in the **Address** Section and enter the address `H'3c4` into the **Address Lo** box as a condition.
- Click **OK** to define the breakpoint.

This will cause a break whenever address `H'3c4` is accessed, either for a read or a write.

The **Breakpoints** window shows the new event you have defined.

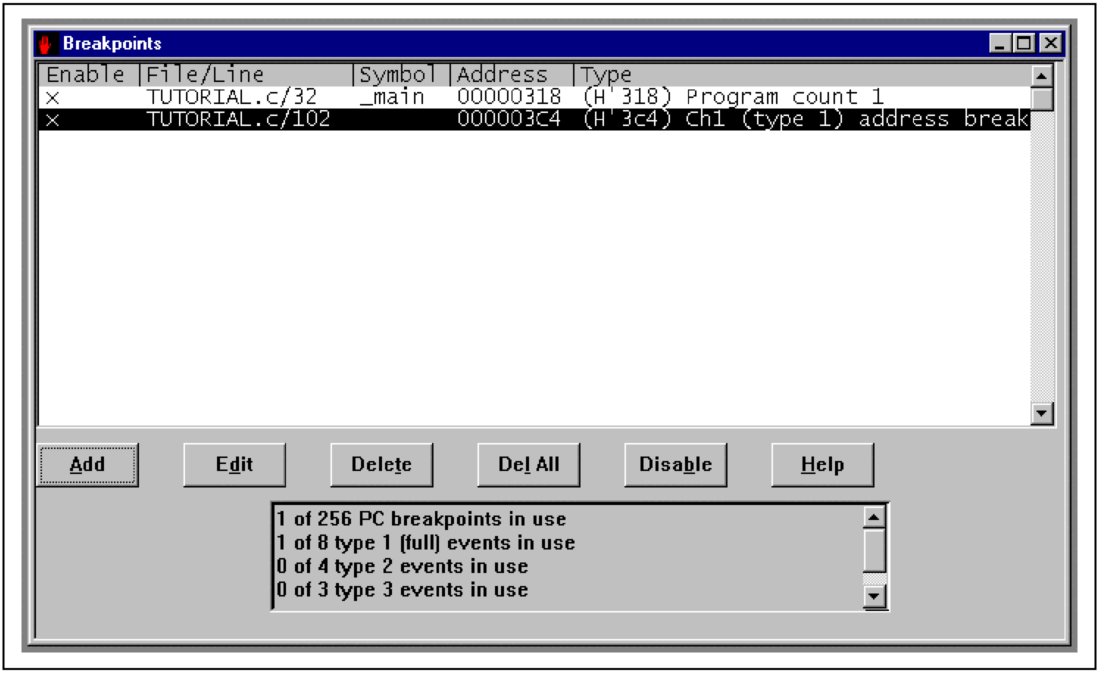


Figure 4.30 Breakpoints Window

- Select the line of the address H'00000318 in the **Breakpoints** window and click **Delete** to delete the PC breakpoint set in address H'318.

- Select **Go Reset** from the **Run** menu or click the **Go Reset** button in the toolbar to execute the program from the reset vector.

Execution will stop at address H'3c4.

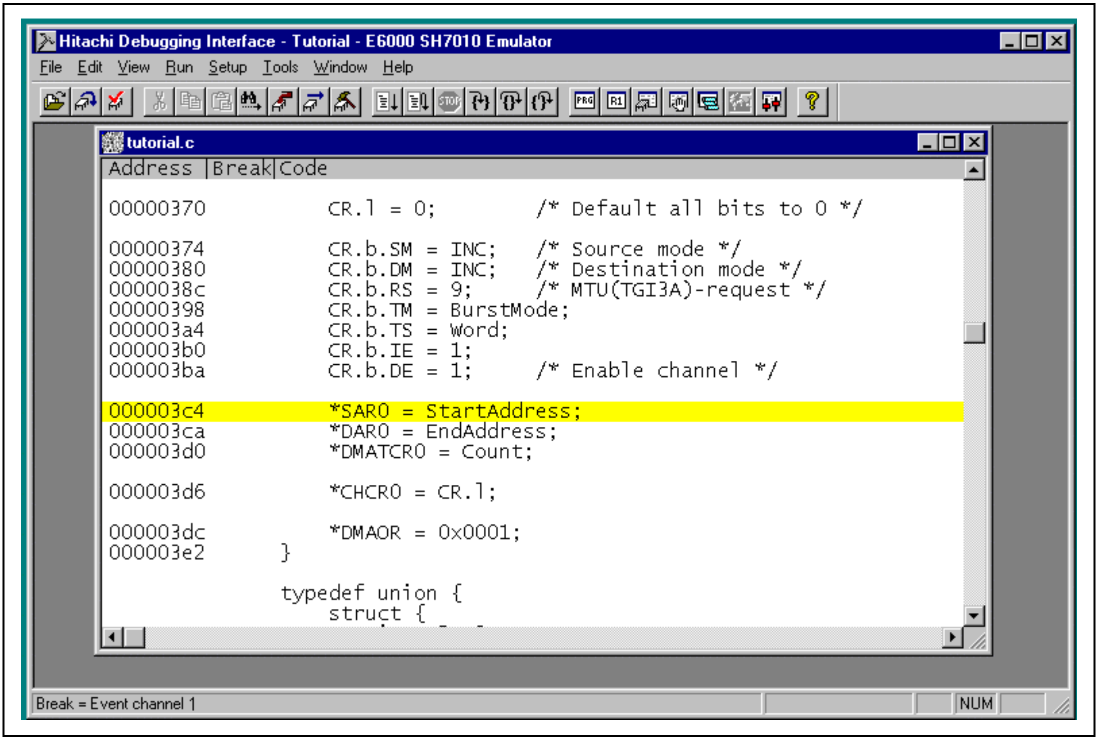


Figure 4.31 Stopping the Program by a Breakpoint

The status bar will display Break = Event channel 1 to indicate that the break was caused by satisfaction of the event condition.

4.9 Using the Trace Buffer

The trace buffer allows you to look back over previous MCU cycles to see exactly what the MCU was doing prior to a specified event.

4.9.1 Displaying the Trace Buffer

You can specify the address accessed by the program to use the trace buffer to look back to see what accesses took place.

- Open the **Trace** window by choosing **Trace Window** from the **View** menu, or click the **Trace Window** button in the toolbar.

If necessary scroll the window down so that you can see the last few cycles. The **Trace** window is displayed, as shown in figure 4.32.

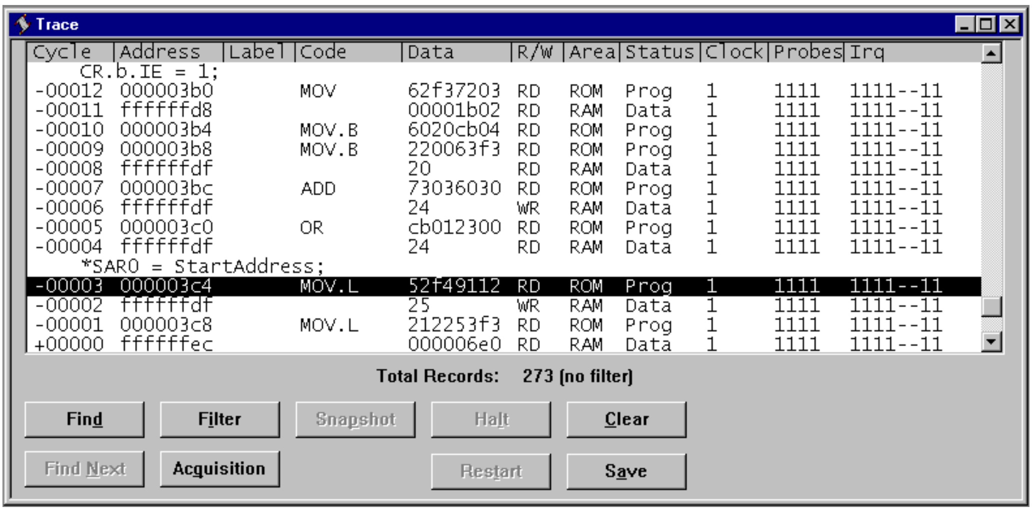


Figure 4.32 Trace Window

- If necessary, adjust the width of each column by dragging the column dividers on either side of the labels just below the title bar.

In cycle `–00003`, you can see that address `H'3c4` has been accessed.

4.9.2 Setting a Trace Filter

Currently the **Trace** window shows all the MCU cycles.

- Click **Filter** to display the **Trace Filter** dialog box.

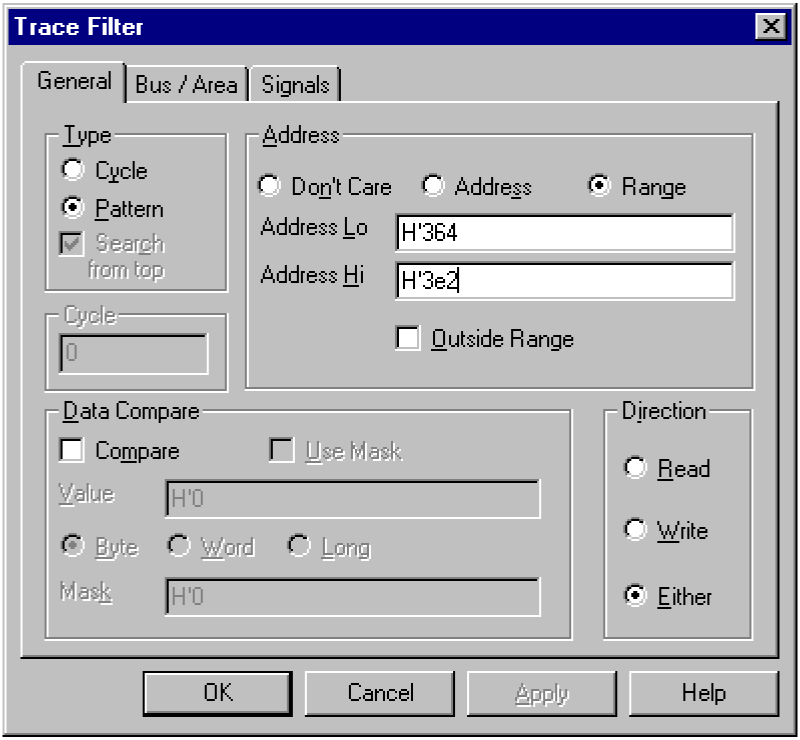


Figure 4.33 General Panel in Trace Filter Dialog Box

This allows you to define a filter to restrict which cycles are displayed in the trace buffer.

- If necessary, click **General** to show the **General** panel.
- Select **Pattern** in the **Type** section.
- In the **Address** section click **Range** and type `H'364` in the **Address Lo** field and `H'3e2` in the **Address Hi** field.
- Click **Bus / Area** to display the **Bus / Area** panel.
- Set **Bus State** to **Instruction Fetch**.

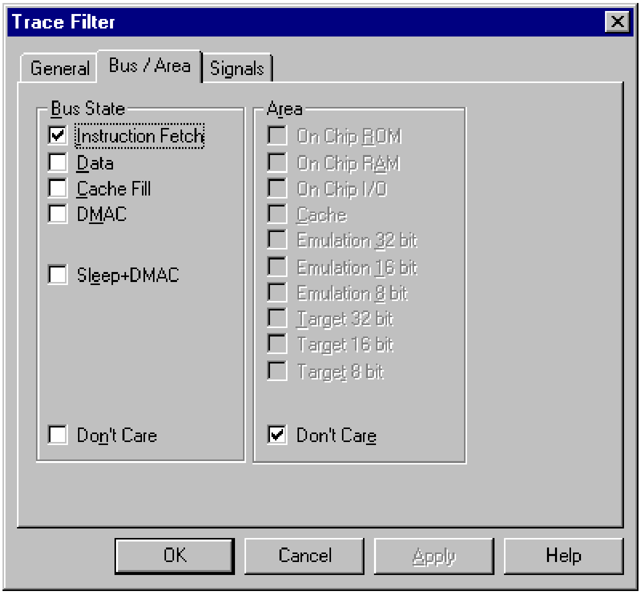


Figure 4.34 Bus / Area Panel in Trace Filter Dialog Box

- Click **OK** to save the trace filter.

In the **Trace** window, only the cycles in which the MCU accessed address range `H'364` to `H'3e2` are displayed.

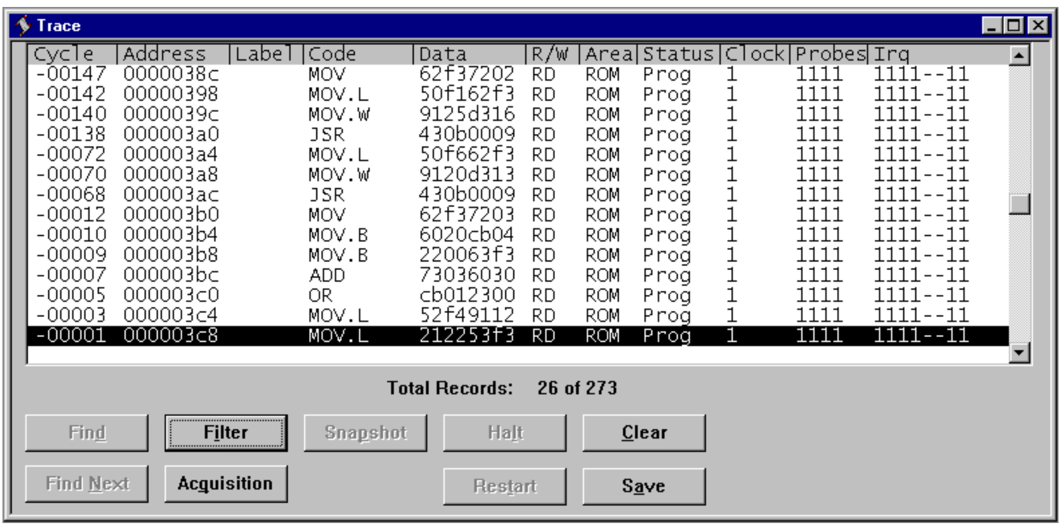


Figure 4.35 Showing Trace Buffer Contents

4.10 Measuring the Performance

By using the performance analysis function in the HDI, you can measure the performance of a program. The results are displayed as a histogram or as percentages.

4.10.1 Selecting the Measurement Conditions

Select the conditions for measurement as follows:

- Select **Performance Analysis Window** from the **View** menu or click the **Performance Analysis Window** button in the toolbar and open the **Performance Analysis** dialog box.

- Click the **Conditions** button and open the **Performance Analysis Conditions** window.
- After clicking **No. 1** in the **Performance Analysis Conditions**, click the **Edit** button and open the **Performance Analysis Properties** dialog box.

The following dialog box will be displayed to allow selection of the measuring conditions.

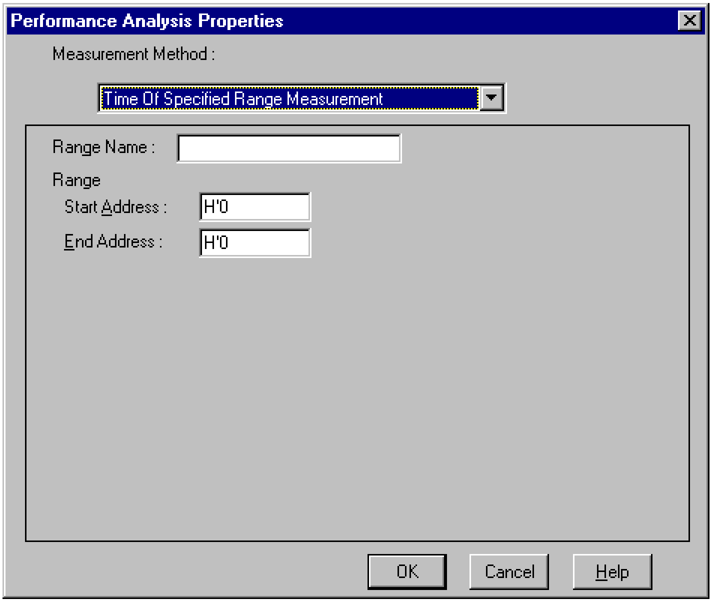


Figure 4.36   Selecting the Conditions for Measurement

- Select **Time Of Specified Range Measurement** from the **Measurement Method** and select the performance analysis condition for specified range measurement.
- Input `Analysis` as the **Range Name**.
- Input address `H'3fc` as the **Start Address** and address `H'428` as the **End Address**.
- Click **OK** to select the conditions.

This completes the selection.

In the **Performance Analysis Conditions** window, the conditions selected in the **Performance Analysis Properties** dialog box are displayed.

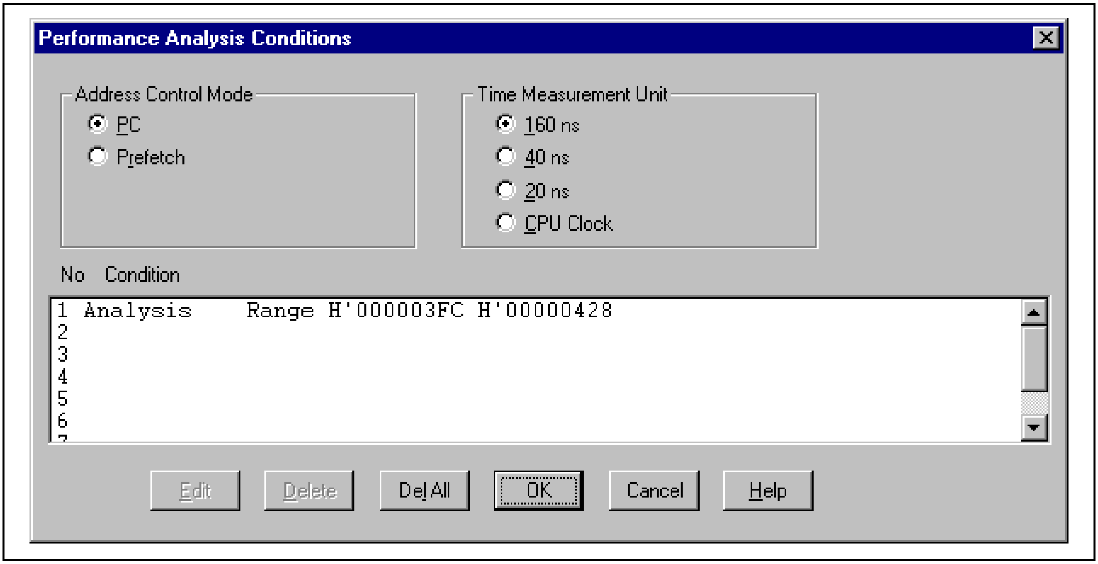


Figure 4.37 Displaying the Measurement Conditions

- Click **OK** to set the measurement conditions.

Now, the performance of the execution in the address range `H'3fc` to `H'428` can be measured.

- Click **Close** and close the **Performance Analysis** dialog box.
- Open the **Breakpoints** window from the **View** menu and cancel all breakpoints by clicking the **Del All** button. Then double-click the **Break** column of the line that includes address `H'34a` and set a PC break.
- Select **Go Reset** from the **Run** menu or click the **Go Reset** button in the toolbar and execute the program from the beginning.

The program will stop at address `H'34a`.

4.10.2 Displaying the Analysis Results

The performance analysis results are displayed as a histogram or as percentages.

- Select **Performance Analysis Window** from the **View** menu or click the **Performance Analysis Window** button in the toolbar and open the **Performance Analysis** dialog box.

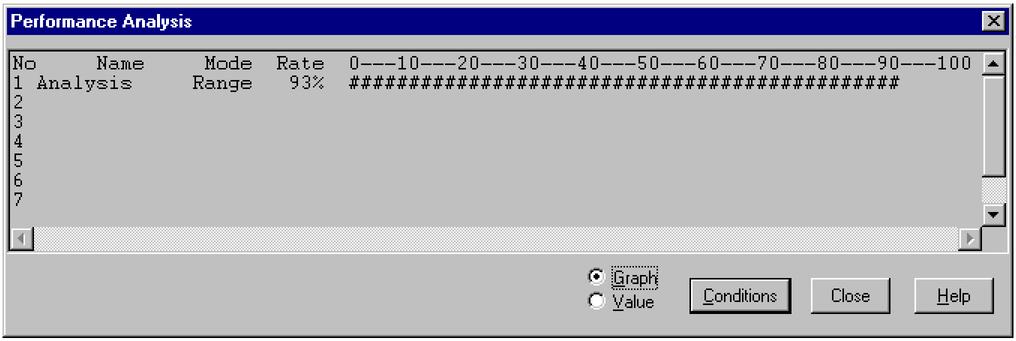


Figure 4.38 Displaying the Analysis Results (1)

The performance analysis results are displayed as a histogram and as percentages.

- Click **Value**.

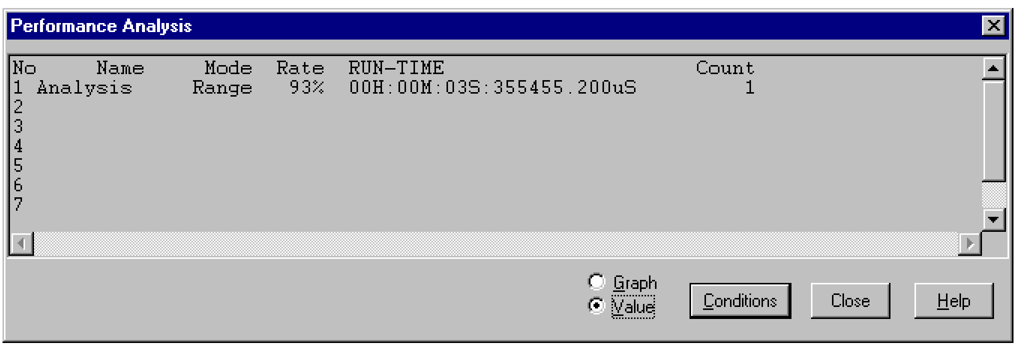


Figure 4.39 Displaying the Analysis Results (2)

The analysis results are displayed as percentages and as the actual time measured.

4.11 Saving the Session

Before exiting, it is good practice to save your session, so that you can resume with the same E6000 emulator and HDI configuration at your next debugging session.

- Choose **Save Session…** from the **File** menu.
- Choose **Exit** from the **File** menu to exit HDI.

4.12 What Next?

This tutorial has introduced you to some of the key features of the E6000 emulator, and their use in conjunction with the HDI. By combining the emulation tools provided in the E6000 emulator you can perform extremely sophisticated debugging, allowing you to track down hardware and software problems efficiently by precisely isolating and identifying the conditions under which they occur. For details on HDI operation, refer to the HDI User's Manual, supplied separately.

Note: For details on each function, refer to the online help. Online help can be displayed by clicking the help key or F1 button on each window or dialog box.

Appendix A Command Line Functions

This section lists the E6000 emulator command line functions.

Command Type:
General: HDI general commands
Specific: Commands specific to the E6000 emulator

For HDI general command line functions, refer to the HDI User's Manual or the on-line help. For E6000-specific commands, refer to the on-line help. To display the on-line help, enter the following in the **Command Line** window:

`helpΔ <command>`

<command>: Command name or its abbreviation

Table A.1 Command List

Command Name	Abbrevia-tion	Command Type	Description
!	—	General	Comments
ACCESS	AC	General	Sets operation for invalid access
ANALYSIS	AN	Specific	Enables or disables the performance analysis range
ANALYSIS_RANGE	AR	Specific	Sets or displays the performance analysis range
ANALYSIS_RANGE_DELETE	AD	Specific	Cancels the performance analysis range
ASSEMBLE	AS	General	Assembles a program
ASSERT	—	General	Checks conditions
BREAKPOINT / EVENT	BP, EN	Specific	Sets a breakpoint or an event
BREAKPOINT_CLEAR, EVENT_CLEAR	BC, EC	Specific	Clears a breakpoint or an event
BREAKPOINT_DISPLAY, EVENT_DISPLAY	BD, ED	Specific	Displays breakpoints or events
BREAKPOINT_ENABLE, EVENT_ENABLE	BE, EE	Specific	Enables or disables a breakpoint or an event
BREAKPOINT_SEQUENCE, EVENT_SEQUENCE	BS, ES	Specific	Defines or clears a breakpoint or event sequence

Table A.1 Command List (cont)

Command Name	Abbreviation	Command Type	Description
CLOCK	CK	Specific	Sets the CPU clock rate in the E6000 emulator
CLOCK_MODE	CM	Specific	Sets and displays clock mode
CLOCK_DIVISOR	CD	Specific	Sets and displays clock divisor
CONFIGURE_PLATFORM	CP	Specific	Sets and displays configuration option
DEVICE_TYPE	DE	Specific	Selects the target device in the E6000 emulator
DISASSEMBLE	DA	General	Disassembles and displays a program
ERASE	ER	General	Clears the contents of the Command Line window
EVALUATE	EV	General	Evaluates an expression
FILE_LOAD	FL	General	Loads an object program file
FILE_SAVE	FS	General	Saves memory contents in a file
FILE_VERIFY	FV	General	Verifies memory contents against file contents
GO	GO	General	Executes a user program
GO_RESET	GR	General	Executes a user program from the reset vector
GO_TILL	GT	General	Executes a user program until a temporary breakpoint
HALT	HA	General	Stops user program execution
HELP	HE	General	Displays the help message for the command line or the command
INITIALISE	IN	General	Initializes the platform
INTERRUPT	IR	General	Validates/invalidates interrupt on the platform. (this command is not supported for some products.)
LOG	LO	General	Manipulates the logging file

Table A.1 Command List (cont)

Command Name	Abbrevia-tion	Command Type	Description
MAP_DISPLAY	MA	General	Displays the memory map information
MAP_LOCATE	ML	Specific	Displays memory mapping information
MAP_SET	MS	Specific	Sets memory mapping
MEMORY_DISPLAY	MD	General	Displays memory contents
MEMORY_EDIT	ME	General	Modifies memory contents
MEMORY_FILL	MF	General	Fills the memory with the specified data
MEMORY_MOVE	MV	General	Moves a memory block
MEMORY_TEST	MT	General	Tests a memory block
MEMORY_UPDATE	MU	Specific	Updates windows related to memory
MODE	MO	Specific	Sets or displays the MCU mode
QUIT	QU	General	Terminates the HDI
RADIX	RA	General	Sets a radix for input value
REGISTER_DISPLAY	RD	General	Displays the MCU register values
REGISTER_SET	RS	General	Sets the MCU register values
RESET	RE	General	Resets the MCU
SLEEP	—	General	Delays command execution
STEP	ST	General	Performs single-step execution in instruction unit or source line unit
STEP_OUT	SP	General	Step out of the current function
STEP_OVER	SO	General	Performs step-over execution
STEP_RATE	SR	General	Set rate for multiple steps

Table A.1 Command List (cont)

Command Name	Abbreviation	Command Type	Description
SUBMIT	SU	General	Executes an emulator command file
SYMBOL_ADD	SA	General	Adds a symbol
SYMBOL_CLEAR	SC	General	Deletes a symbol
SYMBOL_LOAD	SL	General	Loads a symbol information file
SYMBOL_SAVE	SS	General	Saves a symbol information file
SYMBOL_VIEW	SV	General	Displays a symbol
TEST_EMULATOR	TE	Specific	Tests the E6000 emulator hardware
TIMER	TI	Specific	Sets or displays the timer minimum measurement unit for execution time measurement
TRACE	TR	General	Displays trace data
TRACE_ACQUISITION	TA	Specific	Sets or displays trace acquisition information
TRACE_COMPARE	TC	Specific	Compares trace data
TRACE_SAVE	TV	Specific	Saves trace data
TRACE_SEARCH	TS	Specific	Searches for trace data
USER_SIGNALS	US	Specific	Enables or disables user signals

SuperH Family E6000 Emulator User's Manual

Publication Date: Rev.2.00, June 26, 2003
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Technical Documentation & Information Department
Renesas Kodaira Semiconductor Co., Ltd.

SuperH Family E6000 Emulator
User's Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ10B0005-0200H