

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 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 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

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

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

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
  - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
  - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

# R8C Tiny Series E8a Emulator

## Write Facility of the Microcomputer's Internal Flash Memory

### Summary

The E8a emulator, not only usable as a debugger for the user system, can also be used as a writer or programmer for the microcomputer's internal flash memory.

This document describes the procedure for writing load module data into the flash memory after starting the E8a emulator in writer mode.

Explained in this document is for the case where the user system incorporating an R8C family microcomputer and the E8a emulator are used in combination. The content of this document may be used in common for even a different target microcomputer providing it is one of the R8C family microcomputers.

### Table of Contents

1.	Specification .....	2
2.	Facilities Used .....	2
2.1	Outline of the E8a Emulator's Writer Facility .....	2
2.2	Outline of the ID Code Check Facility .....	2
3.	Preparing the Software .....	2
3.1	Getting Started .....	2
3.2	Installing the Software Supplied with the E8a Emulator .....	3
3.3	Installing Other Necessary Software.....	3
4.	Operational Description.....	3
4.1	Starting the High-performance Embedded Workshop .....	4
4.2	Opening a Workspace.....	4
4.3	Setting the ID Code.....	9
4.4	Writing Data into the Flash Memory.....	13
4.5	Using the E8a As a Write Tool .....	15
5.	Frequently Asked Questions .....	26
5.1	Are there any precautions to take when verifying the program behavior with a microcomputer alone after debugging is complete? .....	26
5.2	When a dialog box is displayed asking for ID code, what should be input? .....	27
6.	Frequency Encountered Errors and the Corrective Actions to Take .....	28
7.	Related Documents.....	29

## 1. Specification

Broadly classified, the E8a emulator has two operation modes: a debug mode and a writer mode in which the E8a emulator is used as a flash writer. If the user program is downloaded in debug mode, the emulator control program also, not just the user program alone, is written into the microcomputer. Therefore, the program cannot be run with the MCU alone, without using the E8a emulator (by removing it) after debugging. If the E8a emulator is used as a writer, it must be started in writer mode.

When the E8a emulator is started in writer mode, the user program data only can be written into the microcomputer's internal flash memory after being erased. In this case, since there are no E8a emulator programs in the flash memory, it is impossible to debug the user program using the E8a emulator, but it is possible to run the program with the MCU alone after a write. The load module to be downloaded must be registered in the workspace.

## 2. Facilities Used

### 2.1 Outline of the E8a Emulator's Writer Facility

The E8a emulator's writer mode may be used for such purposes as to write the user program into the microcomputer's internal flash memory singly and then verify how it actually behaves. In this case, since there are no E8a emulator programs in the flash memory, program behaviors are verified without the intervention of the E8a emulator.

In this document, a range of operations up to writing into the flash memory is described using the sample program included in the CD-ROM supplied with the E8a emulator or the downloaded package from the Renesas website.

The versions of respective tools used here are given below.

Integrated development environment

High-performance Embedded Workshop Version 4.03.00

Compiler package

Free evaluation edition for the M16C series

Compiler Package Version 5.43.00

Emulator software

E8a Emulator Software Version 1.01 Release 00

### 2.2 Outline of the ID Code Check Facility

The ID code check facility of the R8C family is the facility to prohibit reading out or erasing the microcomputer's internal flash memory. The ID code is the one written into the internal flash memory by the user and has such a characteristic that unless a matching ID code for it is supplied when using the debugger or flash programmer, the internal flash memory can neither be read out nor erased at all. This is a security facility of the microcomputer, so that should the ID code be forgotten, Renesas cannot do anything about it.

## 3. Preparing the Software

### 3.1 Getting Started

Install the software included in the CD-ROM that is supplied with the E8a emulator.

The sample program (tutorial workspace) used in this document will be expanded in your PC.

The software included in the CD-ROM that is supplied with the E8a emulator may also be installed in a PC that has had the High-performance Embedded Workshop already installed. In this case, part of the dialogs displayed during the installation work will be omitted.

### 3.2 Installing the Software Supplied with the E8a Emulator

Execute HewInstMan.exe present in the CD-ROM supplied with the E8a emulator.

For details on how to install, see the guide to E8a emulator introduction published at the Renesas website. During installation work, follow the instructions displayed on the screen. The installation procedure is omitted here.

### 3.3 Installing Other Necessary Software

- (1) In this document, part of the sample program is modified, with which to verify the program behavior. For this reason, we'll use the C/C++ compiler package for the M16C series. If you've already purchased a product-edition compiler package, install the product-edition compiler package.
- (2) If you've not purchased a product-edition compiler package yet, the free evaluation-edition compiler package included in the CD-ROM that is supplied with the E8a emulator.

A free evaluation-edition compiler package may also be downloaded from the Renesas website. To find a free evaluation edition of the M16C series C/C++ compiler package, choose Support → Download → Download Search from Renesas Top Page and then open category selection, in which select "Free Evaluation Edition" and search for the product you want. Links to the Renesas website are given in the last section of this document. Information about the limitations of and the method for installing the free evaluation edition may be obtained from the download page.

- (3) If you've selected the auto update utility when installing the software, it is possible to confirm via the Internet whether the latest version of each product is available.

## 4. Operational Description

This section describes the procedure for operating in flash memory writer mode after starting the High-performance Embedded Workshop (HEW). The procedure is shown below.

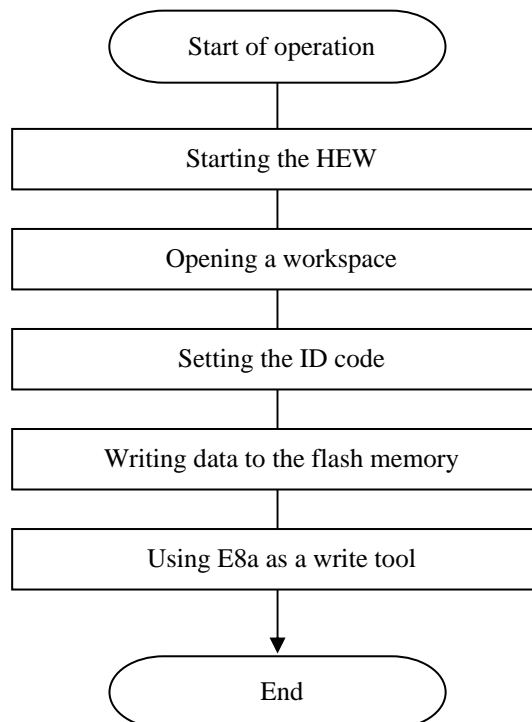


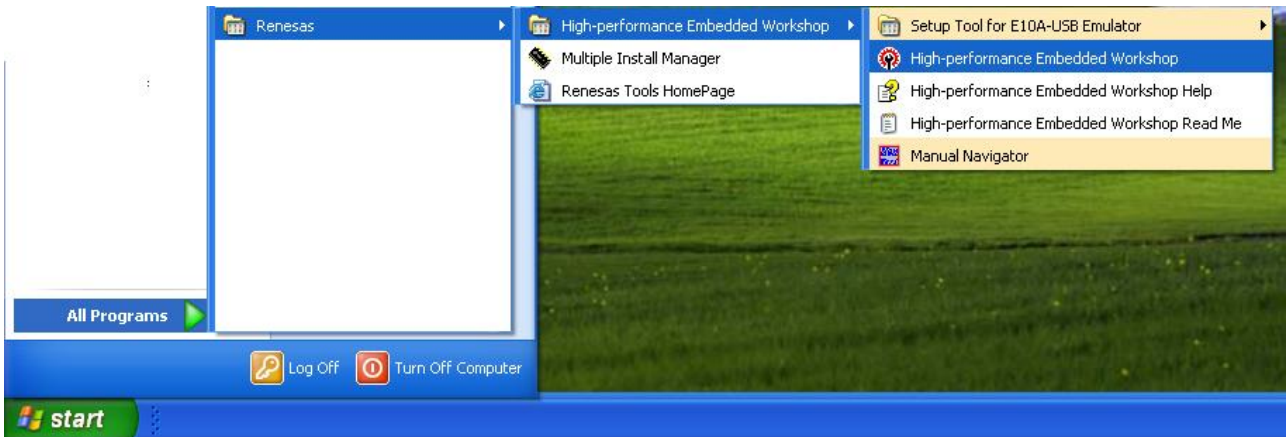
Figure 4.1 Example Program Execution Procedure

### 4.1 Starting the High-performance Embedded Workshop

To begin with, first connect the E8a emulator that has the user system connected to it and the host computer with USB cable, and check to see that everything is ready to debug.

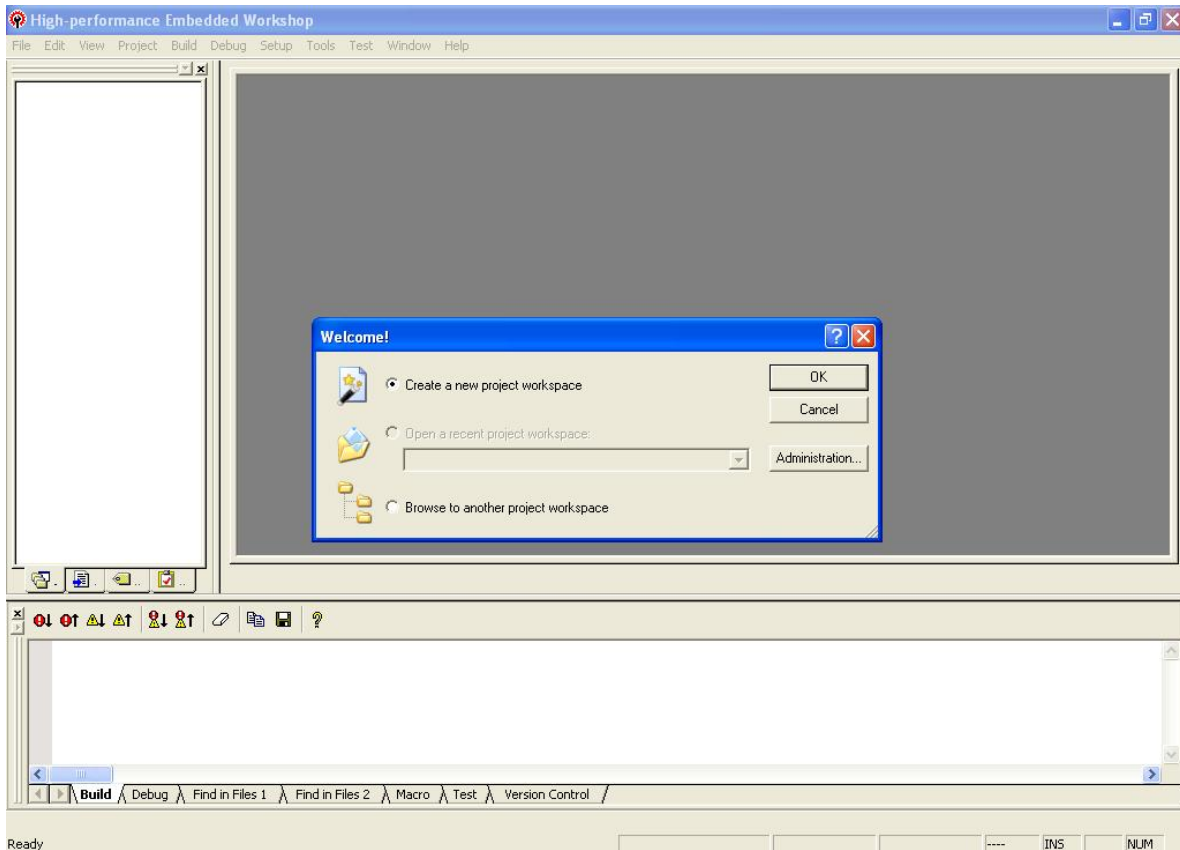
Next, start the High-performance Embedded Workshop.

From All Programs on the Start menu, choose Renesas → High-performance Embedded Workshop → High-performance Embedded Workshop, to start.

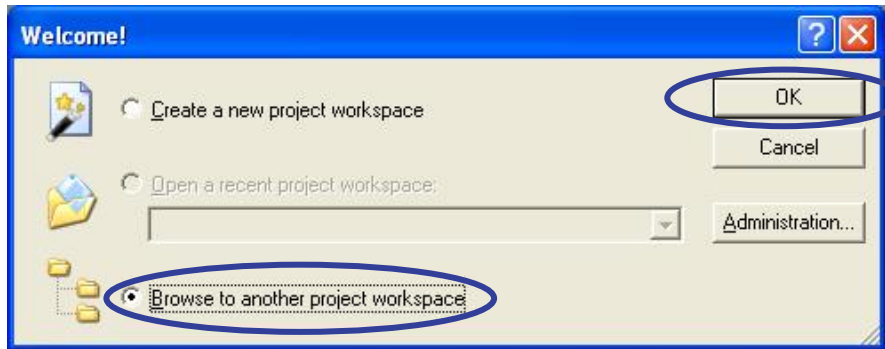


### 4.2 Opening a Workspace

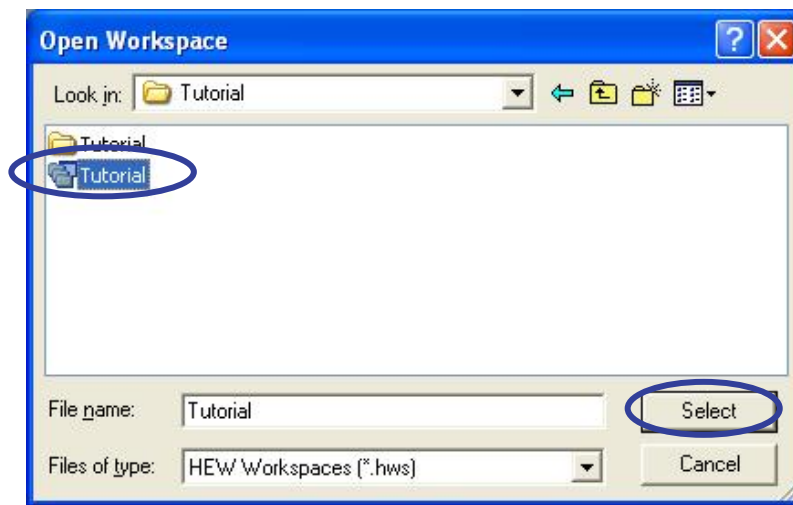
(1) The welcome dialog box will be displayed in the High-performance Embedded Workshop.



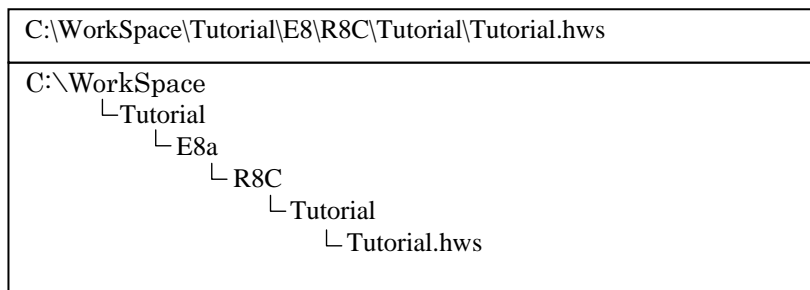
Select the "Browse Another Project Space" radio button in the welcome dialog box and click the OK button.



(2) The Open a Workspace dialog box will be displayed.



If the installation of the CD-ROM of this product is complete, the workspace "Tutorial.hws" is stored as standard in the folder position shown below. Check folder positions in order while you locate. When the workspace "Tutorial.hws" is found, specify it and click the Select button.



Note: Depending on the software version used, it will occur that the above directory cannot be specified. In such a case, select the directory given below.

<Directory in which the High-performance Embedded Workshop is installed>

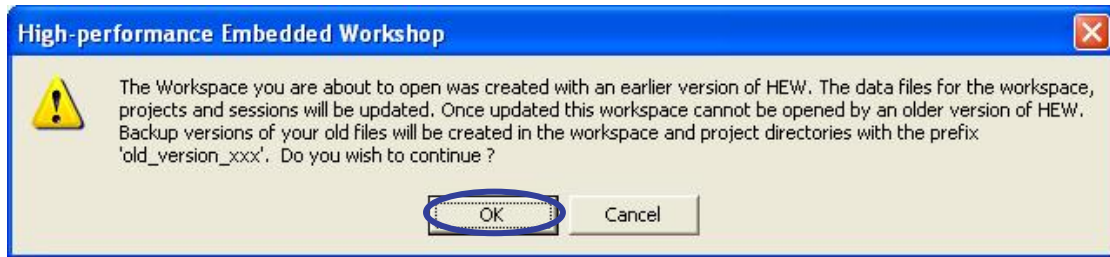
\Tools\Renesas\DebugComp\Platform\E8a\R8C\Tutorial

Examples of directory:

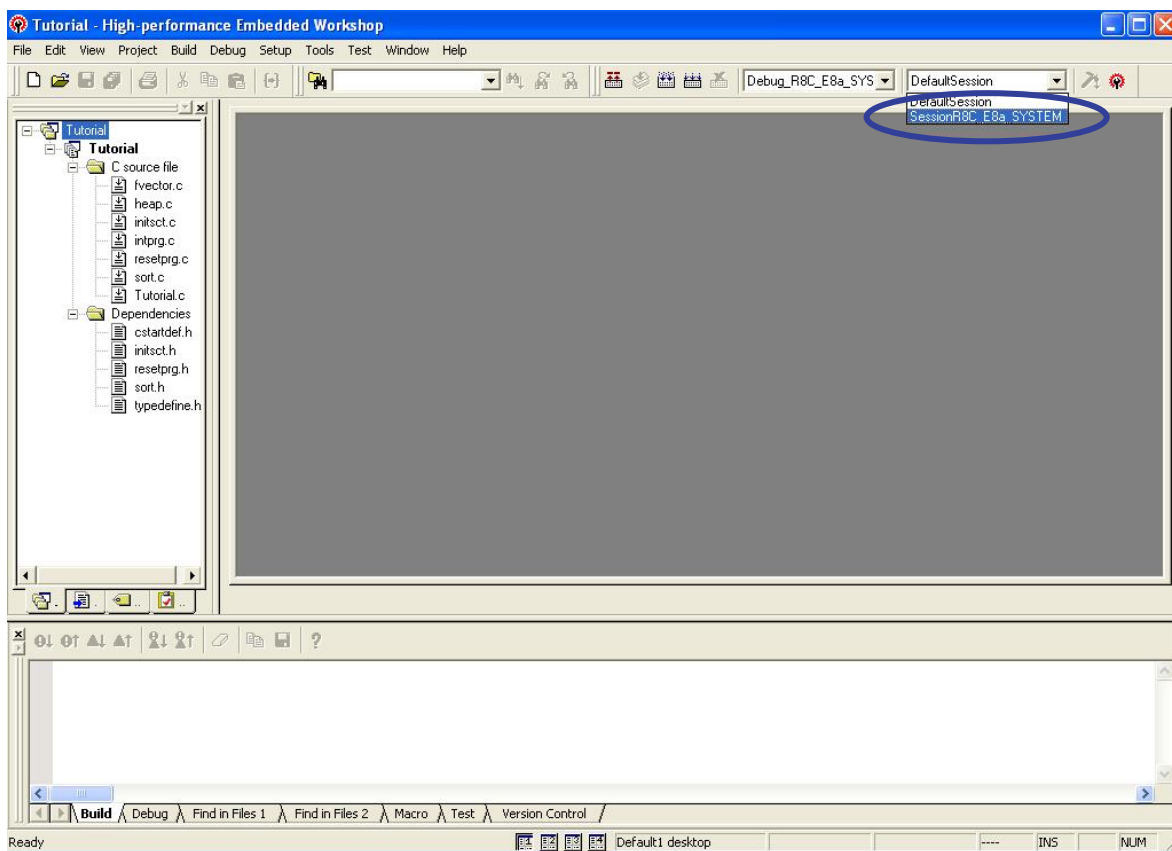
C:\hew3\Tools\Renesas\DebugComp\Platform\E8a\R8C\Tutorial

C:\hew2\Tools\Renesas\DebugComp\Platform\E8a\R8C\Tutorial

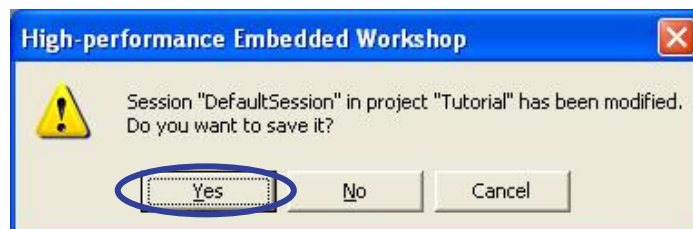
- (3) If the workspace version is old, the dialog box shown below is displayed. To update it to a new version, click the OK button.



- (4) When the workspace is opened, you are ready to use the High-performance Embedded Workshop. Change sessions to connect the E8a emulator. To do it, change "DefaultSession" in the toolbar to "SessionR8C\_E8a\_SYSTEM."

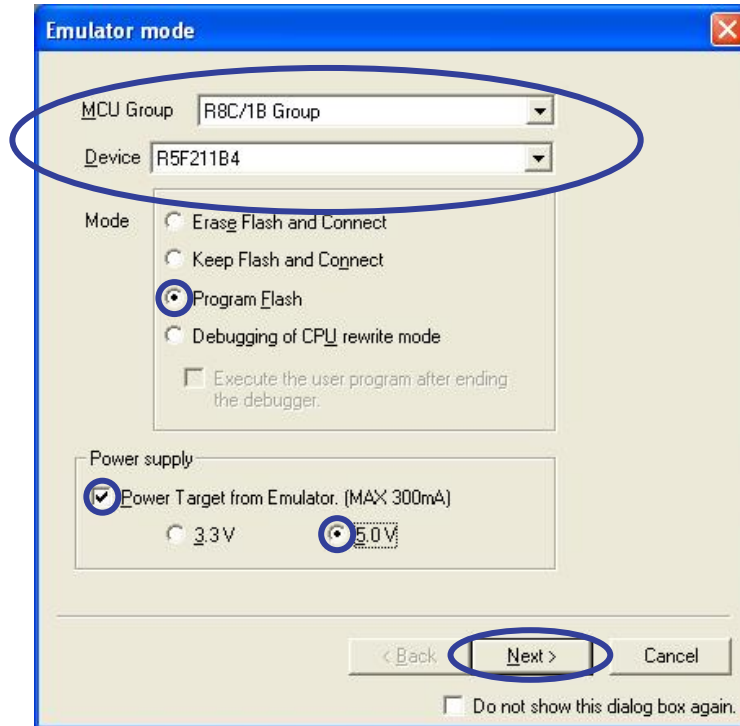


If the dialog box shown below is displayed prompting for your confirmation of whether to save the previous session, click the Yes button.





- (5) When a connection of the E8a emulation begins, the Emulator Mode dialog box is opened.

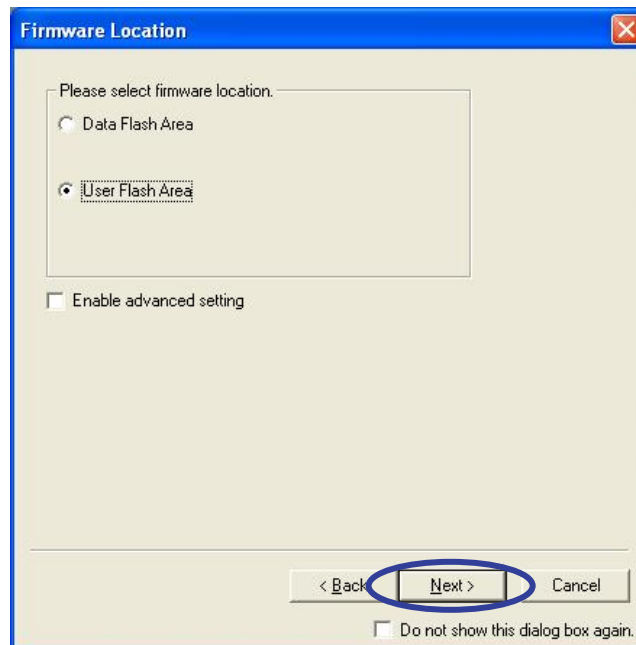


Select the microcomputer name incorporated in the user system from the Device menu of the dialog box. In the example here, "R5F211B4" is selected.

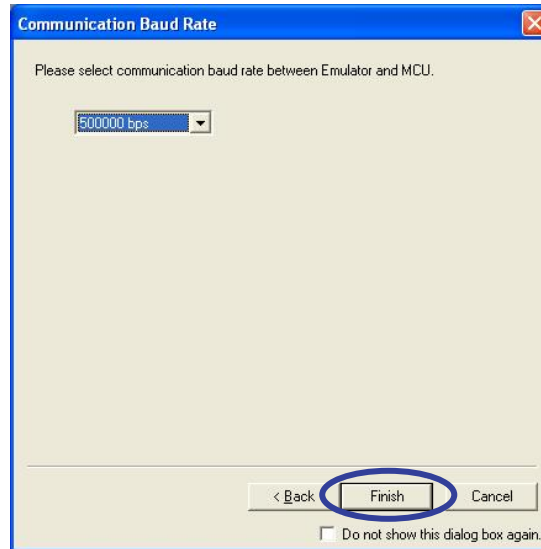
For Mode, select "Write to Flash Memory." This is the mode in which the E8a is used as a writer.

Furthermore, select options in the Power Supply column according to the user system's power supply condition. In the example here, because the power is fed from the E8a to the user system, select "Power Supply from Emulator" and then "5.0V." Finally, click the Next button.

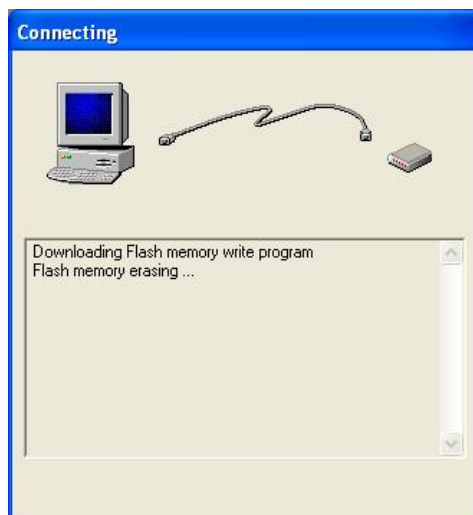
- (6) The Firmware Location dialog box will be displayed. Since this setting has no effect in writer mode, simply click Next leaving default settings intact.



- (7) The Communication Baud Rate dialog box will be displayed. There is specifically no need to change the communication baud rate from the default "500000 bps." Click the Finish button.

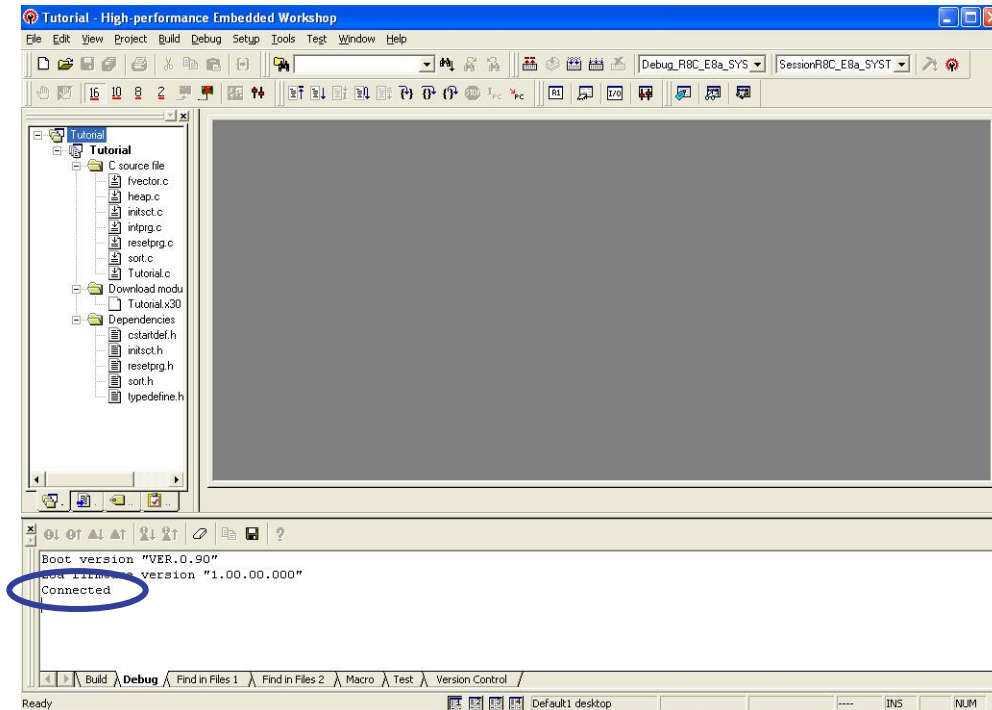


- (8) When a connection of the E8a emulator is in progress, the Connecting dialog box shown below is displayed.



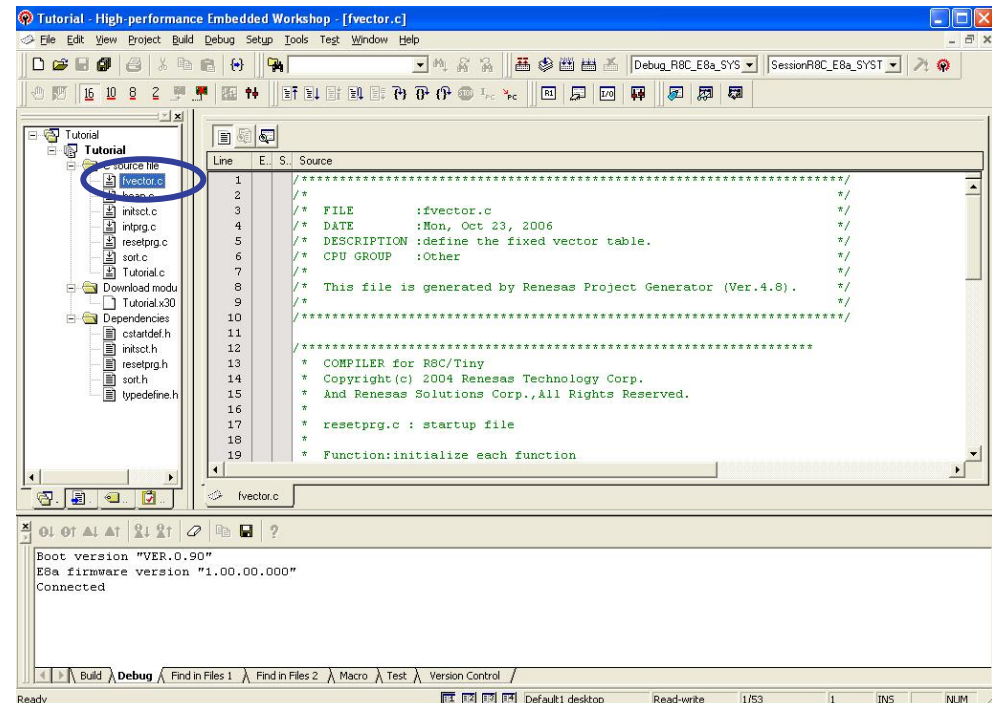
- When a connection in the "Write to Flash Memory" mode is complete, the entire area of the microcomputer's internal flash memory is erased (blanked) to get it prepared for a write.
- The R8C family microcomputer is provided with the ID code check facility to prevent the flash memory from being read or rewritten easily by a third party.
- This ID code check facility ensures that unless the ID code sent from the E8a matches the 7-byte ID code written in the flash memory, the E8a can neither read out nor erase the flash memory. Each byte of the ID code consists of 8-bit data, located at the addresses 00FFDFh, 00FFE3h, 00FFEBh, 00FFEFh, 00FFF3h, 00FFF7h and 00FFFBh beginning with the first byte. Before the ID code check facility can be used, a program that has had an ID code set in it must be written into the flash memory at these addresses.
- If any value other than FF is written at the ID code store address of the microcomputer used, a dialog box is displayed prompting for the confirmation of the ID code before a connection of the E8a emulator begins. If this dialog box is displayed, enter the ID written into the flash memory. Unless the correct ID code value is supplied, next operations cannot be performed. Therefore, the ID codes set by the user need to be managed carefully. In this document, explanation is made on condition that when the E8a emulator is connected for the first time, the microcomputer in a blank state where the ID code value consists entirely of FF is used.

- (9) When the E8a emulator is successfully connected, a message "Connected" is displayed on the Debug tab of the output window.



### 4.3 Setting the ID Code

- (1) Double-click the file name "fvector.c" in the workspace window to open the source file.



If the file "fvector.c" is one of the source files automatically generated by the project generator, it should contain the assembler directive ".ID" that sets an ID code. This assembler directive ".ID" permits you to set an ID code easily.

- (2) Scroll the source file "fvector.c" until the 46th line where the directive ".ID" is written appears.

Line	E..	S..	Source
33			<code>#pragma interrupt/v _dummy_int //address_match</code>
34			<code>#pragma interrupt/v _dummy_int //single_step</code>
35			<code>#pragma interrupt/v _dummy_int //wdt</code>
36			<code>#pragma interrupt/v _dummy_int //reserved</code>
37			<code>#pragma interrupt/v _dummy_int //reserved</code>
38			<code>#pragma interrupt/v start</code>
39			
40			<code>#if __WATCH_DOG__ != 0</code>
41			<code>  _asm(" .ofsreg OFEH");</code>
42			<code>#else</code>
43			<code>  _asm(" .ofsreg OFFH");</code>
44			<code>#endif</code>
45			
46			<code>_asm(" .id ""\`"#FFFFFFFFFFFFFFF\`"");</code>
47			
48			<code>#pragma interrupt _dummy_int()</code>
49			<code>void _dummy_int(void){}</code>

Initially, the ID begins with “#” and consists of a 14-digit hexadecimal value “FFFFFFFFFFFFFFF.”

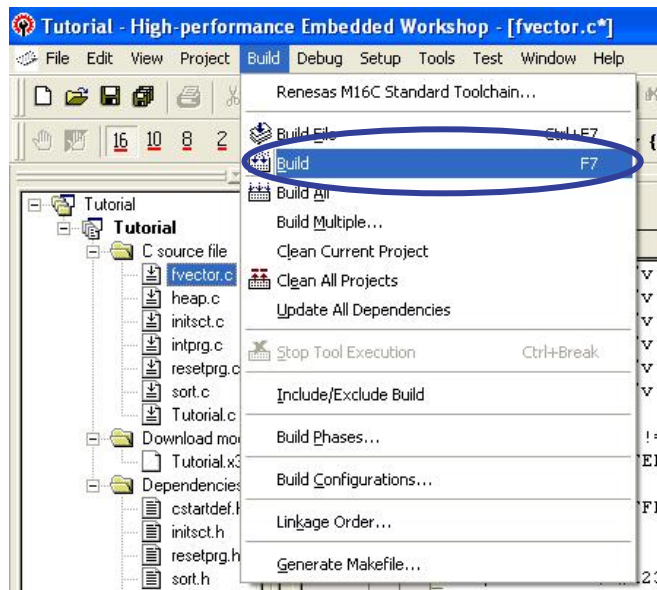
(3) Edit the ID code part to set your desired value.

Line	E..	S..	Source
33			<code>#pragma interrupt/v _dummy_int //address_match</code>
34			<code>#pragma interrupt/v _dummy_int //single_step</code>
35			<code>#pragma interrupt/v _dummy_int //wdt</code>
36			<code>#pragma interrupt/v _dummy_int //reserved</code>
37			<code>#pragma interrupt/v _dummy_int //reserved</code>
38			<code>#pragma interrupt/v start</code>
39			
40			<code>#if __WATCH_DOG__ != 0</code>
41			<code>  _asm(" .ofsreg OFEH");</code>
42			<code>#else</code>
43			<code>  _asm(" .ofsreg OFFH");</code>
44			<code>#endif</code>
45			
46			<code>_asm(" .id ""\`"#123456789ABCDE\`"");</code>
47			
48			<code>#pragma interrupt _dummy_int()</code>
49			<code>void _dummy_int(void){}</code>

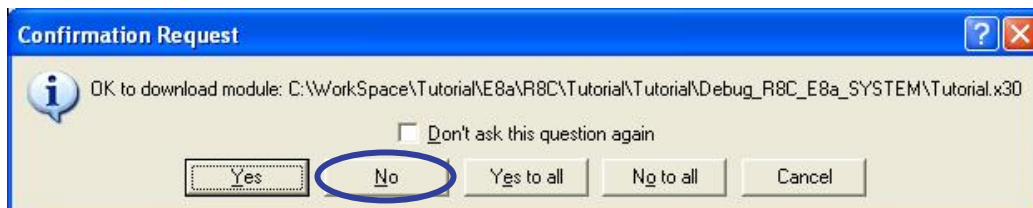
In the example here, a hexadecimal value “123456789ABCDE” is used.

For details on how to set the ID code, see the relevant sections in the assembler user’s manual of the C compiler package where “In30,” “lmc30” and “directive commands” are described.

(4) Choose Build from the Build menu.

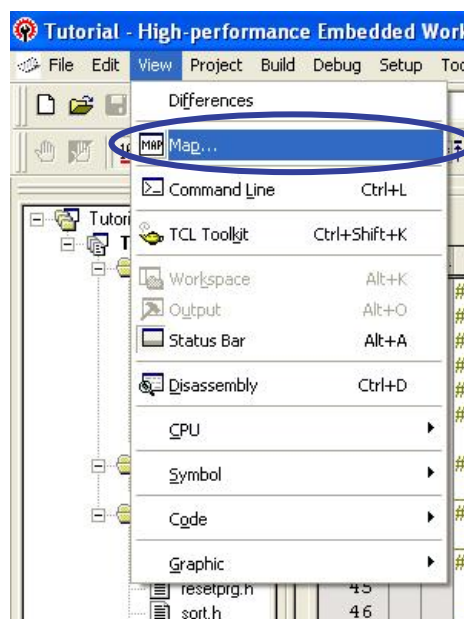


(5) When a build is complete, the dialog box shown below is displayed asking for your confirmation of whether to proceed to perform a download into the flash memory. Here, select “No or “No for All.”

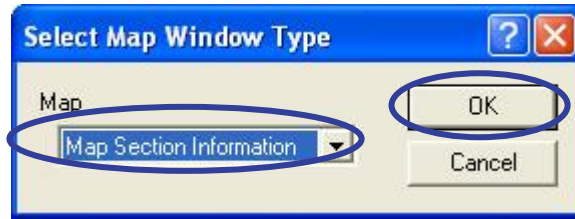


In the example presented in this document data is written into the flash memory after checking map information, so that automatic download is canceled here. In cases where it is known beforehand that data can be written in correctly, you might select “Yes” or “Yes for All” and proceed to perform a download into the flash memory.

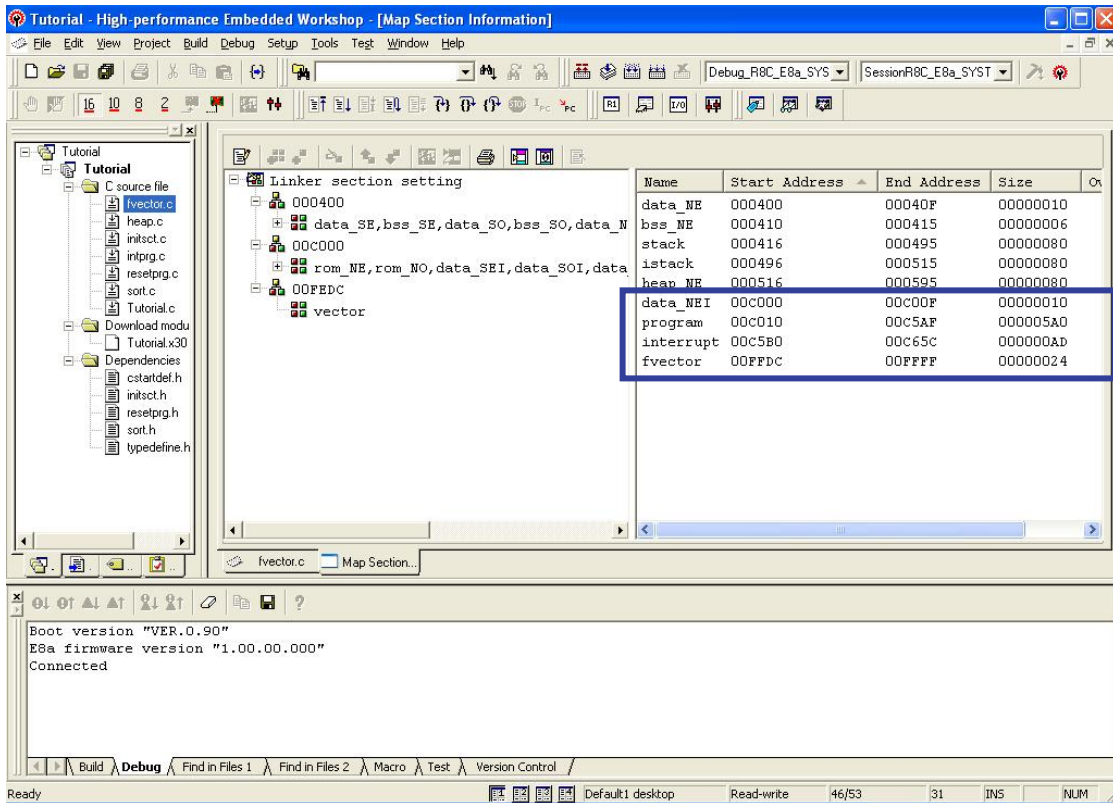
(6) Choose Map from the View menu.



- (7) The Select Type of Map dialog box will be displayed. In this dialog box, select "Map Section Information" and click the OK button.



- (8) Section information will be displayed on the Map Section Information tab.

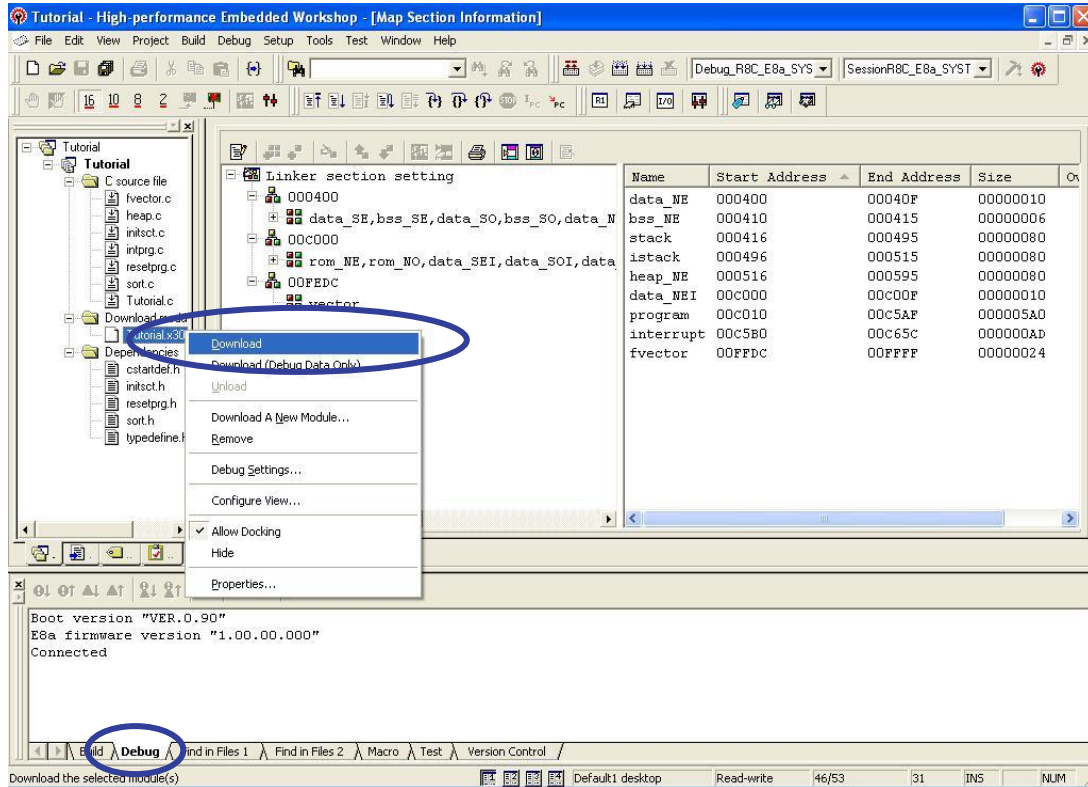


Sorting the displayed "Start Address" in ascending order, we know that the data with ROM attribute to be stored in the flash memory area is present in a location from the address C000 to the address FFFF.

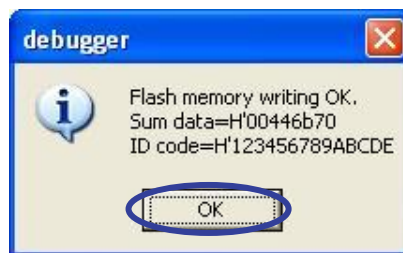
Note that the map window displayed in this example has its docking view turned off.

#### 4.4 Writing Data into the Flash Memory

- (1) Select the Debug tab of the output window. Next, right-click the load module filename displayed in the Download Modules part of the workspace display column and then choose Download from the context menu. Or a load module can be downloaded directly by double-clicking its filename.

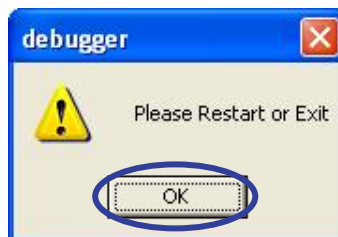


- (2) When a download to the flash memory is complete, the "Sum Data" value and "ID Code" to check are displayed. After confirming the displayed contents, click the OK button.



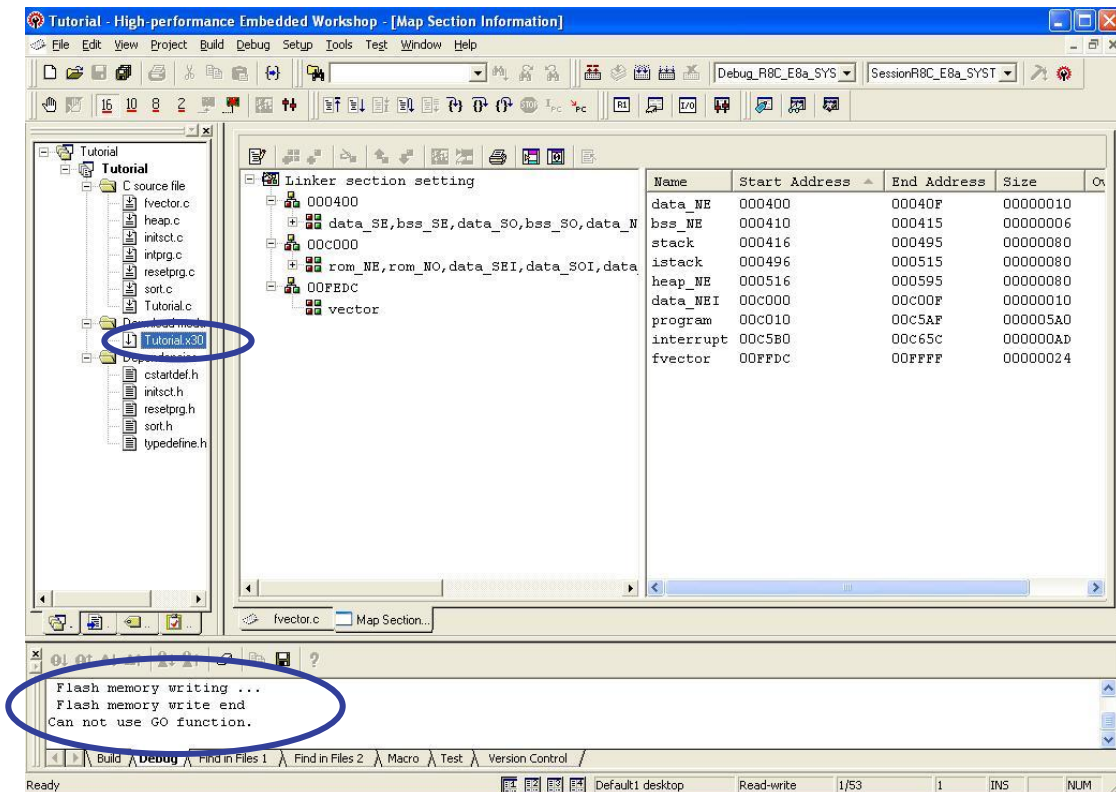
Shown for ID Code here is the user-set value that is specified in the source file.

This completes writing data into the flash memory. When the message box shown below is displayed, click the OK button.



No ID codes are reflected by simply writing one in the program. An ID code is written into the flash memory only when it is downloaded. Note also that in other than writer mode, the ID code is always FFFFFFFFFFFFFFFF in hexadecimal for convenience's sake, no matter what value is set in the user program. In other words, it is in only writer mode that ID codes are reflected by downloading.

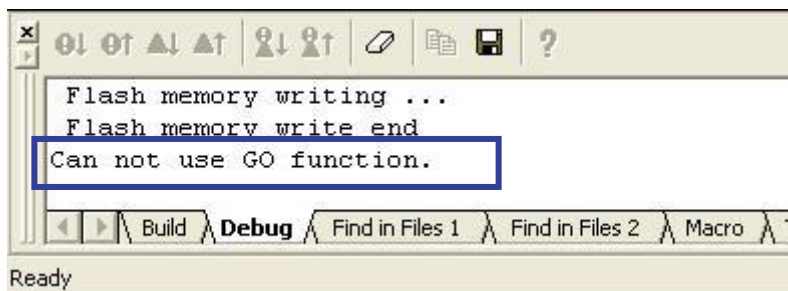
- (3) While a write to the flash memory is underway, a message “Flash Memory Writing...” is displayed on the Debug tab of the output window. When the write is complete, “Flash Memory Write End” is displayed and the icon representing the load module file in the workspace is marked with a down arrow.



In this state, you cannot debug yet because there are no E8a emulator programs in the flash memory.

The values displayed in the memory window, etc. are dummy.

If any debug operation is attempted, an error message is displayed on the Debug tab of the output window.



Shown above is an example of an error display that appears when program execution is specified.

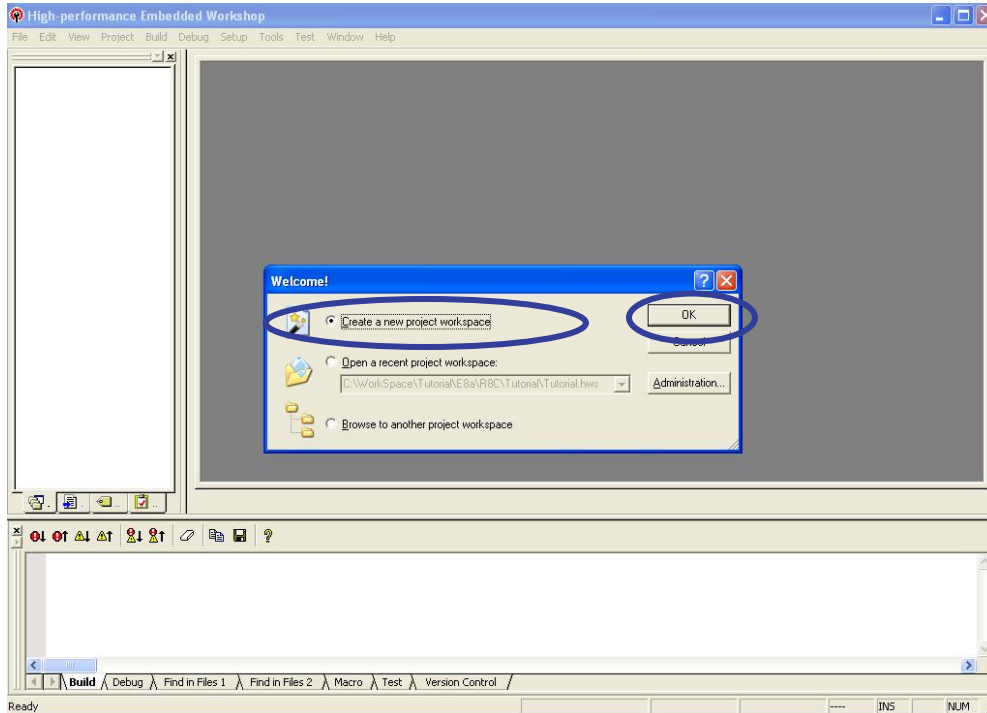
Here, close the E8a emulator. The next section describes the procedure for making use of the E8a emulator's writer facility to write existing data into the flash memory.



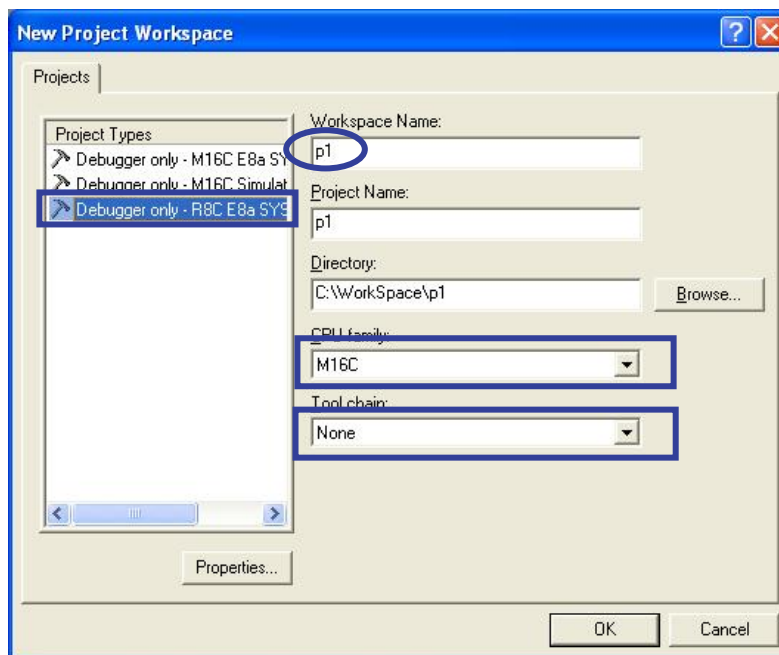
## 4.5 Using the E8a As a Write Tool

So far we've explained how to write into the flash memory in an environment where there is already a workspace available for debugging use. From now, we'll explain the method for using the E8a as and only as a write tool to write existing data into the flash memory.

- (1) In the welcome dialog box of High-performance Embedded Workshop displayed on startup, select "Create New Project Workspace" and click the OK button. If you already have a project created for development use, download it and after switching sessions proceed to step (6).

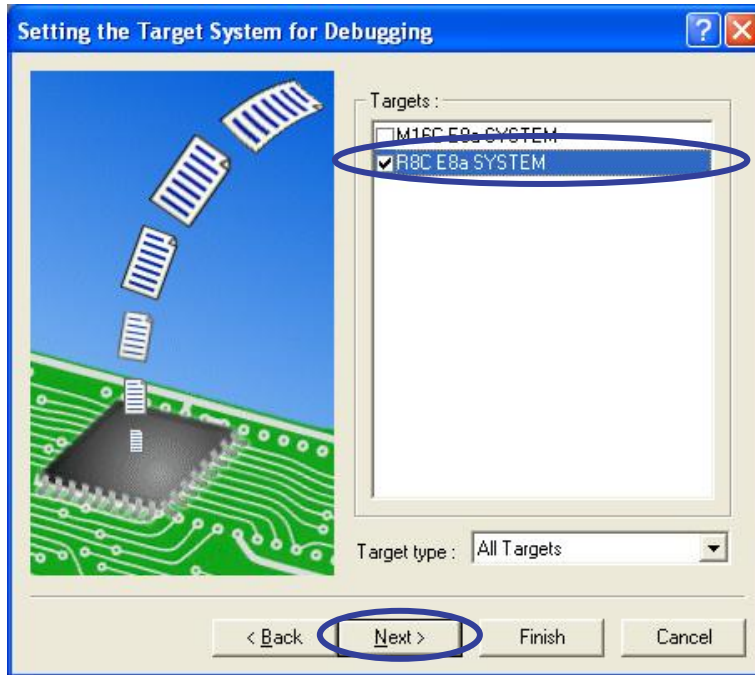


- (2) The New Project Workspace dialog box will be displayed. Select "M16C" for CPU Type and "None" for Toolchain. For Project Type, select "Debugger Only - R8C E8a SYSTEM," type a name in the Workspace Name text box and then click the OK button.



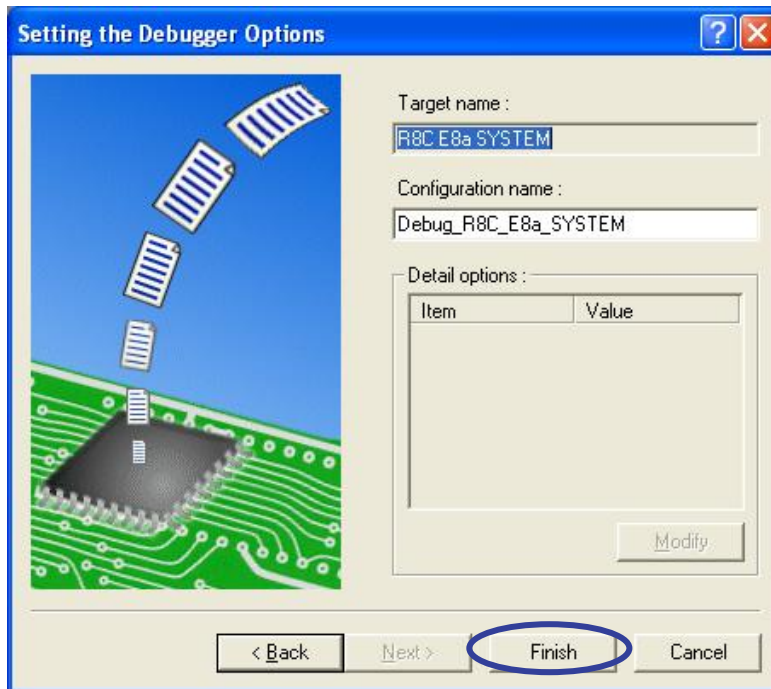
In the example here, the workspace name "p1" is set.

(3) Select the target concerned and click Next.

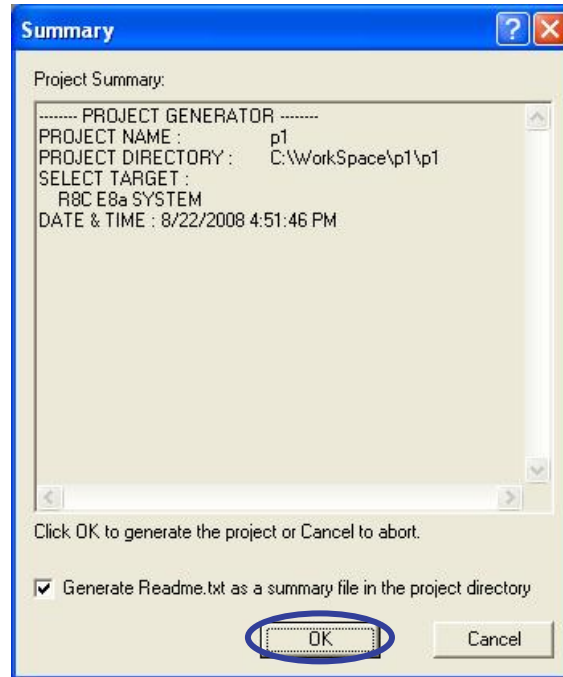


In the example here, "R8C E8a SYSTEM" is selected.

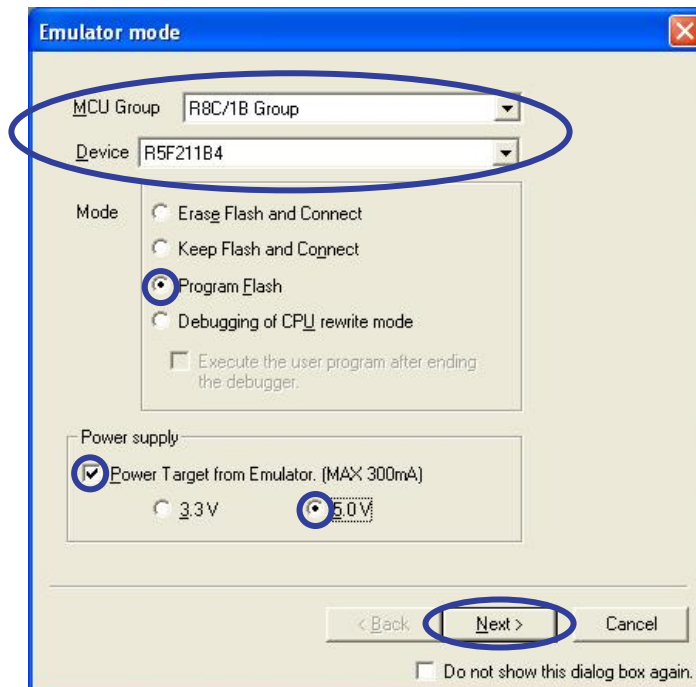
(4) The Debugger Options dialog box will be displayed. Leaving the default intact, simply click Finish.



(5) "Outline of Project" will be displayed. Click the OK button.



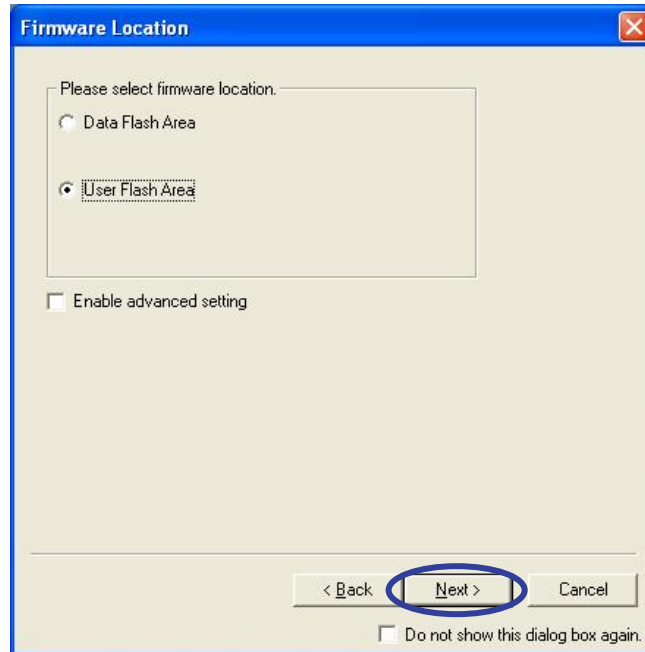
(6) The dialog box for connection to the E8a will be displayed. Select the microcomputer's type name to be written into Device, and select "Flash Memory Data Write" for Mode. Select options in the Power Supply column according to the user system's power supply condition. Finally, click the OK button. Before performing this work, be sure that the E8a and the user system are connected.



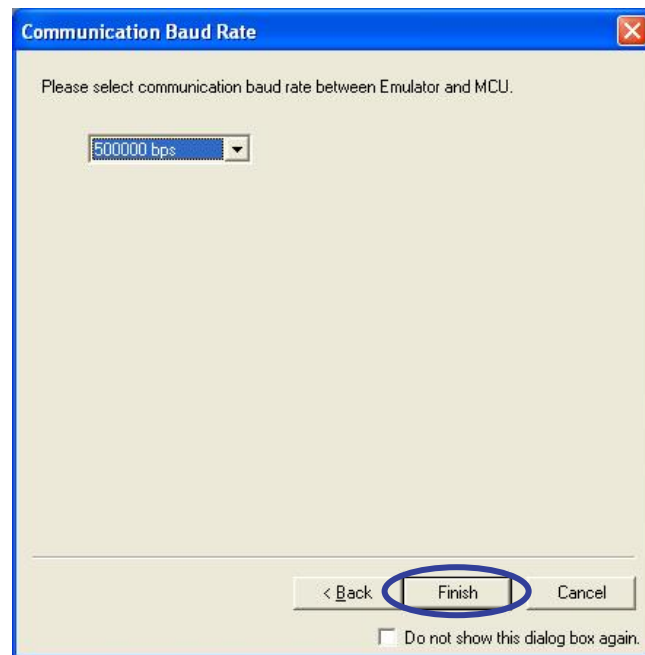
If the dialog box for connection to the E8a is not displayed, choose Connect from the Debug menu.

If a write with higher reliability is required (e.g., as used for mass-production writes), supply the power from the user system without using the power supply facility of the E8a. (\* Since the voltage fed from the E8a depends on the PC's USB power supply performance, the accuracy of the supplied voltage cannot be guaranteed.)

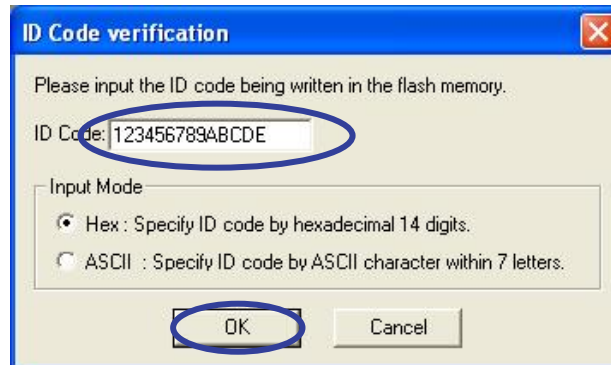
- (7) The Firmware Location dialog box will be displayed. Leaving the default intact, simply click the Next button. (In writer mode, settings made here have no effect.)



- (8) The Communication Baud Rate dialog box will be displayed. Leaving the default intact, simply click the Finish button.



- (9) If the microcomputer's internal flash memory has had any user-set value (other than FFFFFFFFFFFFFFFF in hexadecimal) written into its ID code part, the ID Code Confirmation dialog box is displayed. Enter the value that you set when specifying ID code and click the OK button.

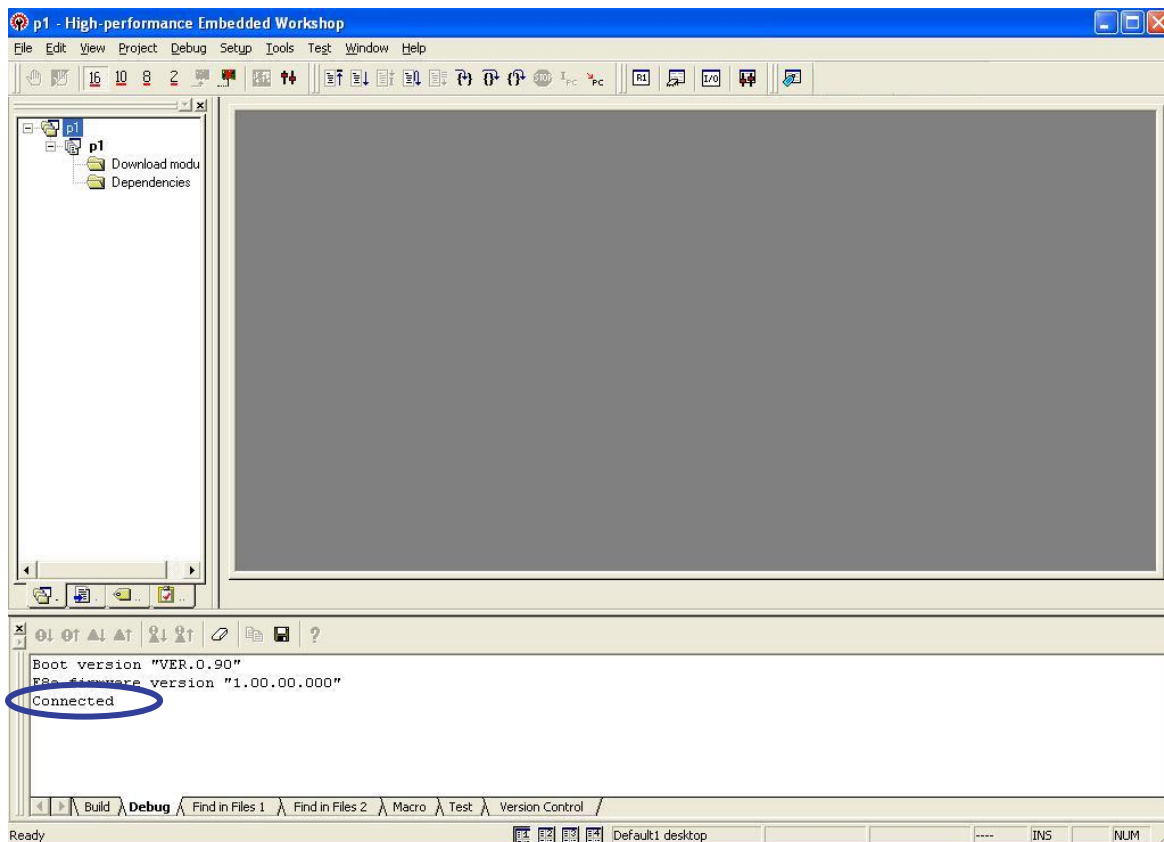


Shown in the above example is the ID code confirmation screen that is displayed when the E8a emulator is connected to a microcomputer that has had ID code written in it following the procedure in Section 4.3 of this document.

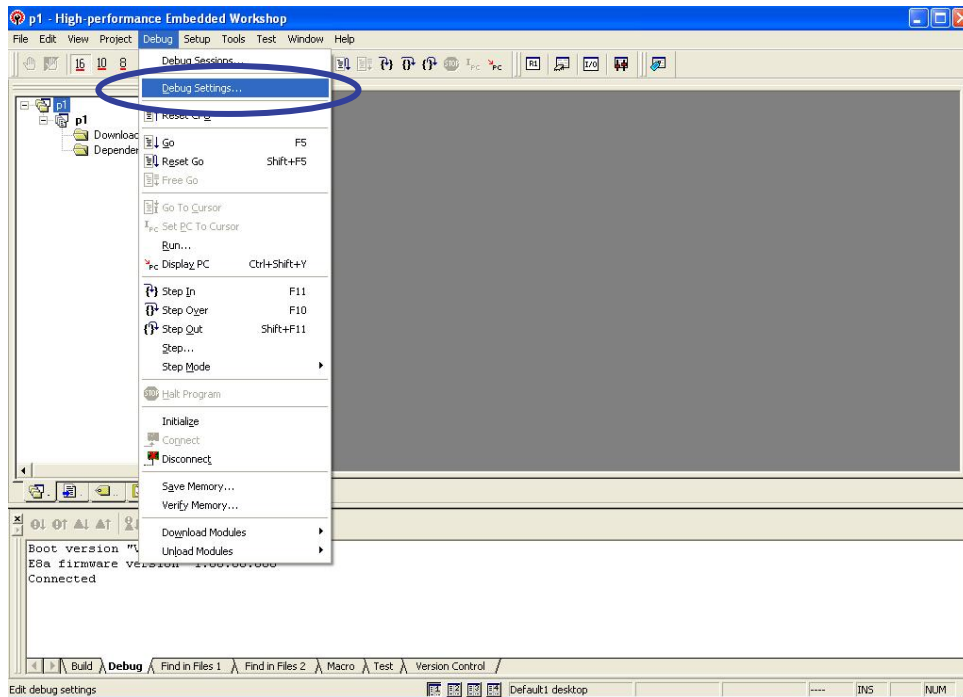
Unless the correct ID code is supplied in the ID code confirmation dialog box, you cannot proceed to the next step. Therefore, the ID code that you set needs to be managed carefully.

If the ID code is FFFFFFFFFFFFFFFF in hexadecimal, the ID code confirmation dialog box is not displayed and you are brought to the next screen.

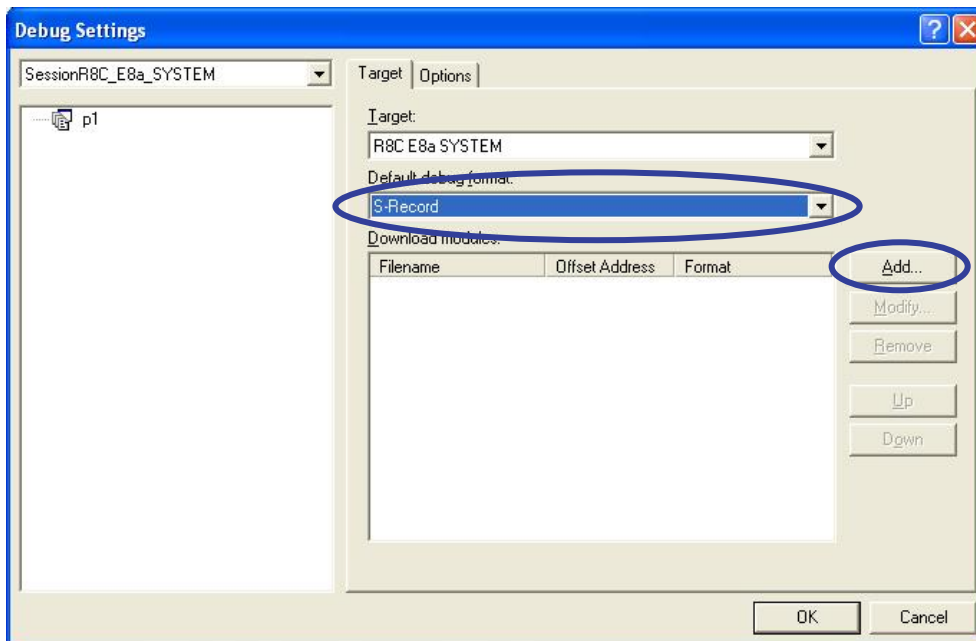
- (10) When the E8a emulator is successfully connected, a message "Connected" is displayed on the Debug tab of the output window.



(11) Choose "Debug Setting" from the Debug menu.



(12) The Debug Setting dialog box will be displayed. Specify the file format to download (write) for Format To Debug and click the Add button. This setting is saved in the project, so there is no need to set from the next time on.

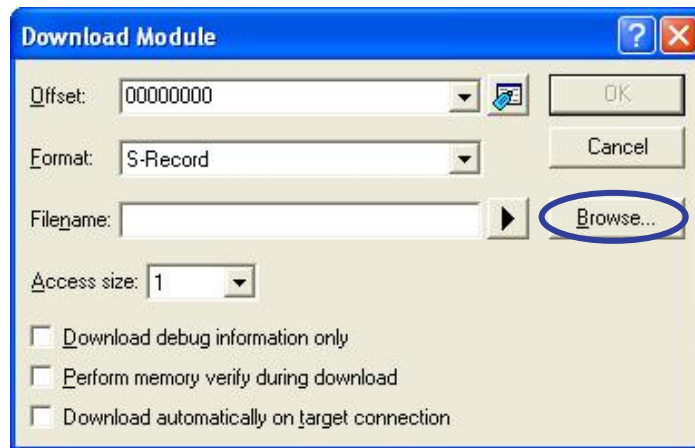


The selectable formats are listed below.

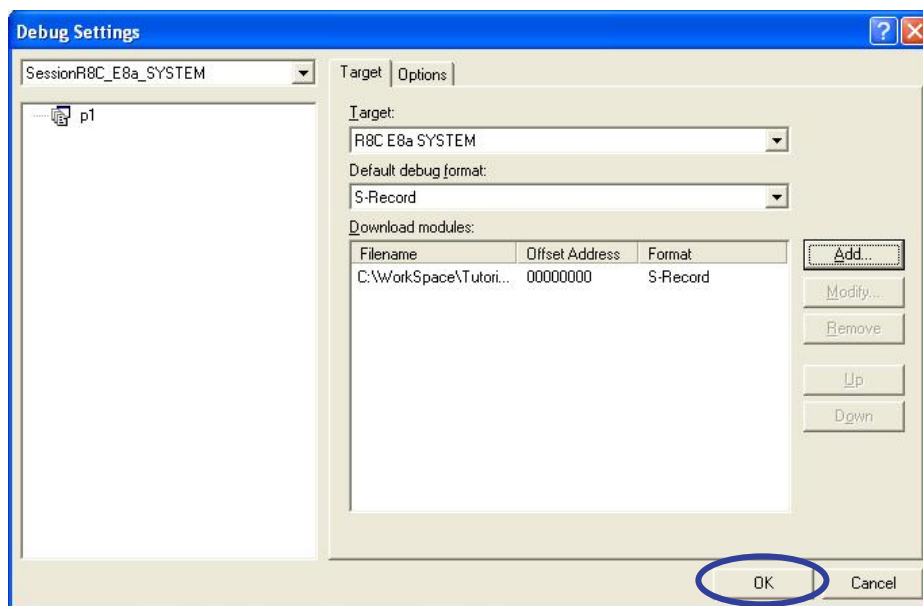
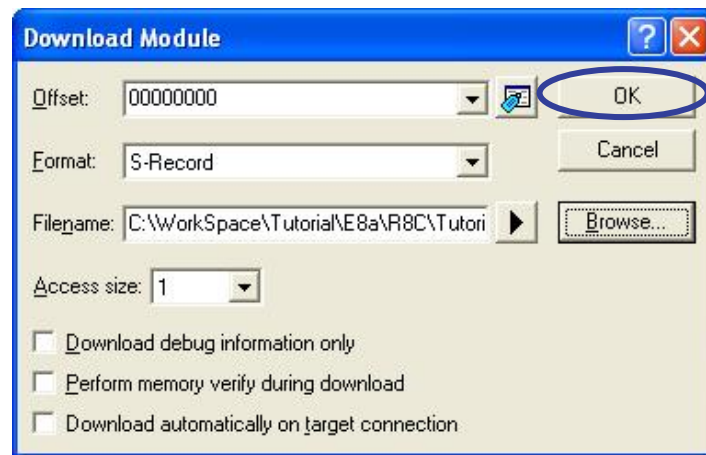
- IEEE-695 RENESAS: Renesas object format (x30 files)
- Intel Hex: Intel HEX format (hex files)
- S-Record: Motorola S format (mot files, s files)

In the example here, Motorola S format is selected as the format to debug.

(13) The Download Module dialog box will be displayed. Click the Browse button and select a file from the ensuing list.

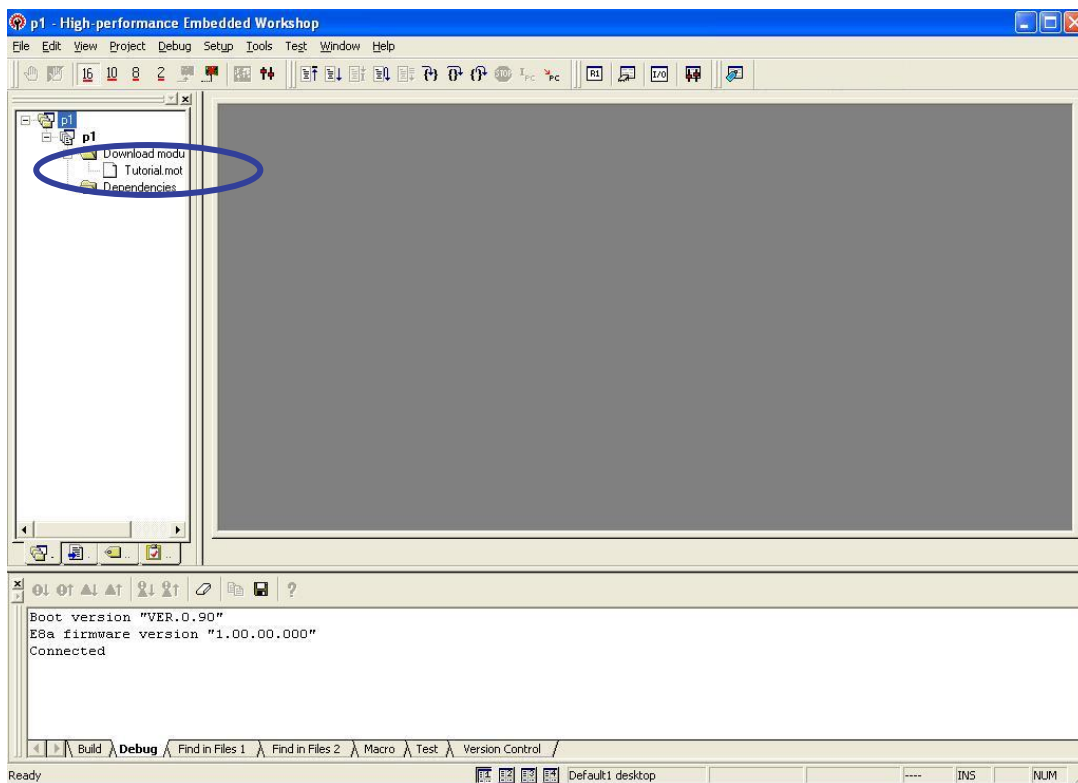


(14)After confirming the file to download, click the OK button for each.

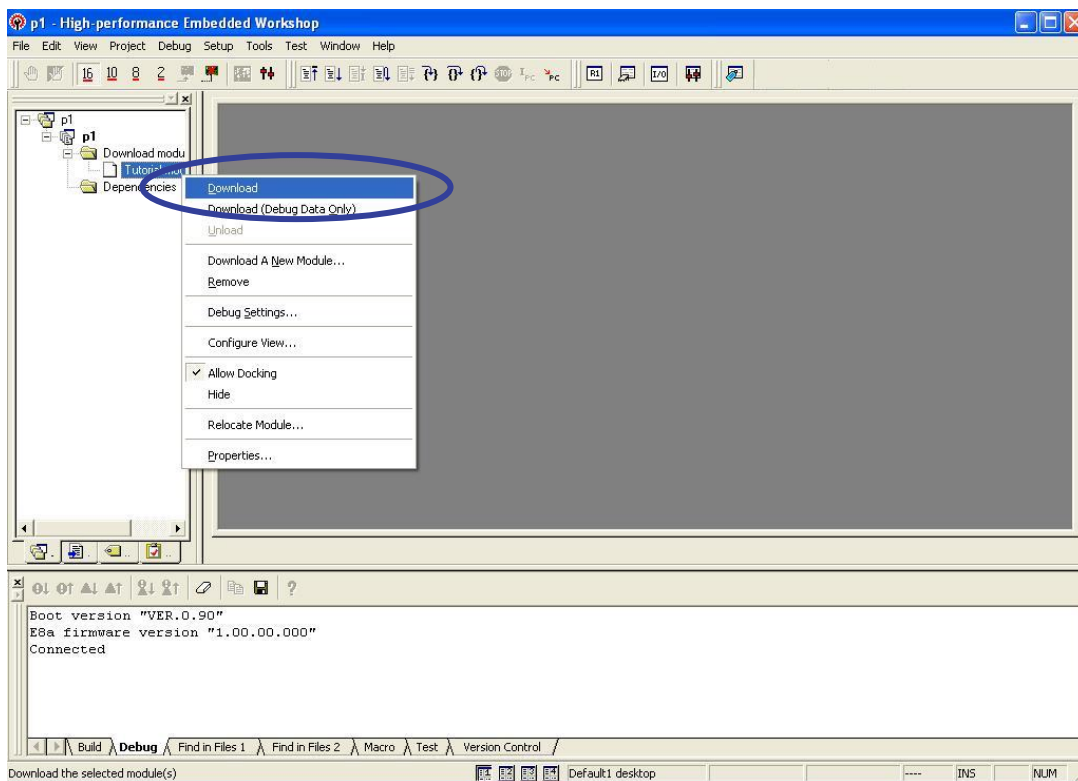


Here, an output file in Motorola S format of the sample program (tutorial workspace) used in this document is specified to be the file to debug.

(15) The file to download is added to the list in the workspace window.



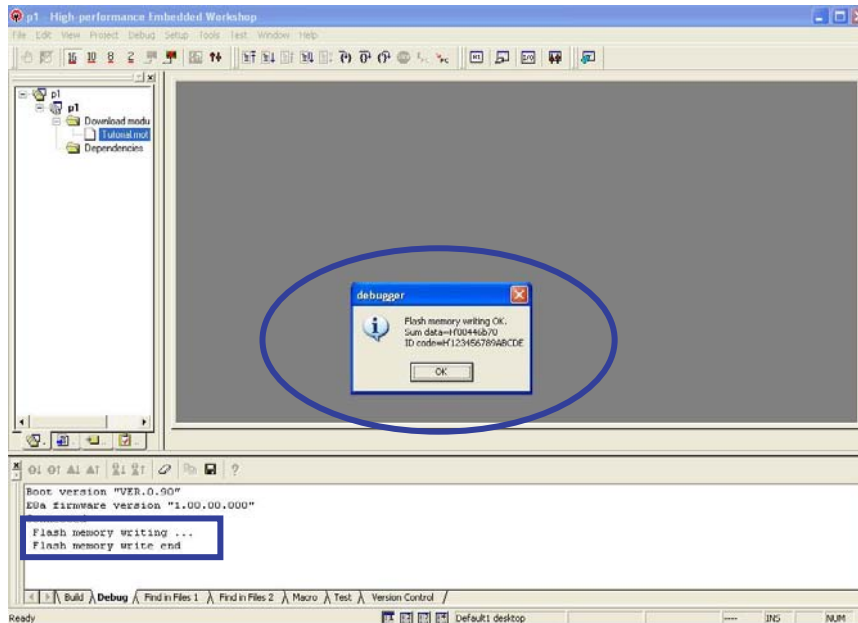
(16) Right-click the filename part of the file in the workspace window and select "Download" from the context menu.



In the Flash Memory Data Write mode, only the download facility works. The other facilities do not show the correct value.



(17)When a flash memory data write is complete, a message box “Flash Memory Writing OK” is displayed along with “Sum Data” and “ID Code.” After confirming the content, click the OK button.



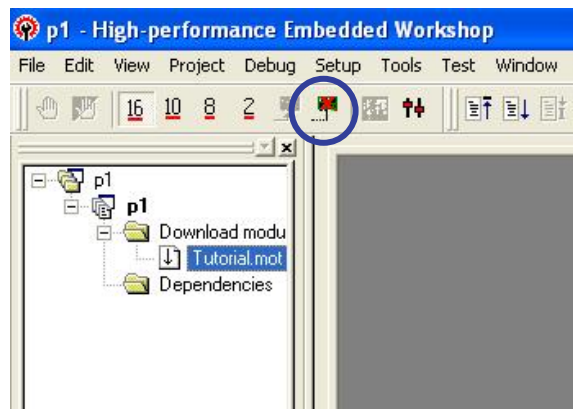
When a flash memory write begins, a message “Flash Memory Writing...” is displayed on the Debug tab of the output window. When the write is complete, a message “Flash Memory Write End” is displayed.

(18)The message box “Please Reset or Exit” follows. Click the OK button.

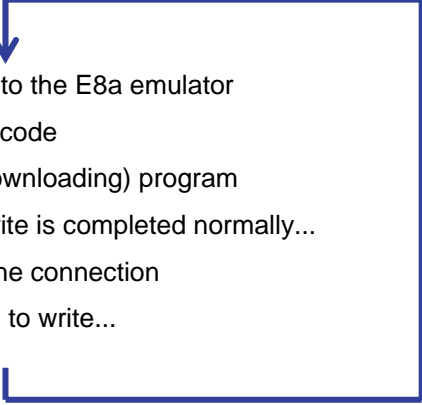


This completes writing to the flash memory. In the Flash Memory Data Write mode, no programs can be debugged using the E8a emulator. Restart or close the E8a emulator.

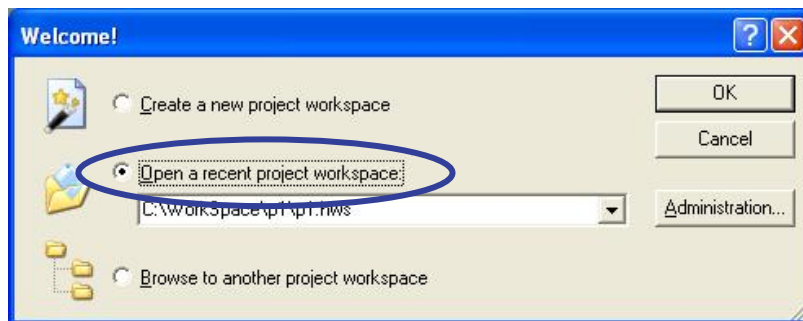
(19)Click the Disconnect HEW button in the toolbar to sever a connection with the E8a emulator before removing the user system. To terminate a write, close the HEW here, too. To continue to write, proceed to the next step. Even when you continue to write, be sure to disconnect the E8a once.



(20)To continue to write, click the Connect HEW button in the toolbar and repeat steps (6), (9), (16), (17), (19) and (20).

- 
- (6) Connecting to the E8a emulator
  - (9) Entering ID code
  - (16) Writing (downloading) program
  - (17) When a write is completed normally...
  - (19) Severing the connection
  - (20) Continuing to write...

(21) To write the same file, launch the workspace you've created above by opening it the next time you start the HEW.



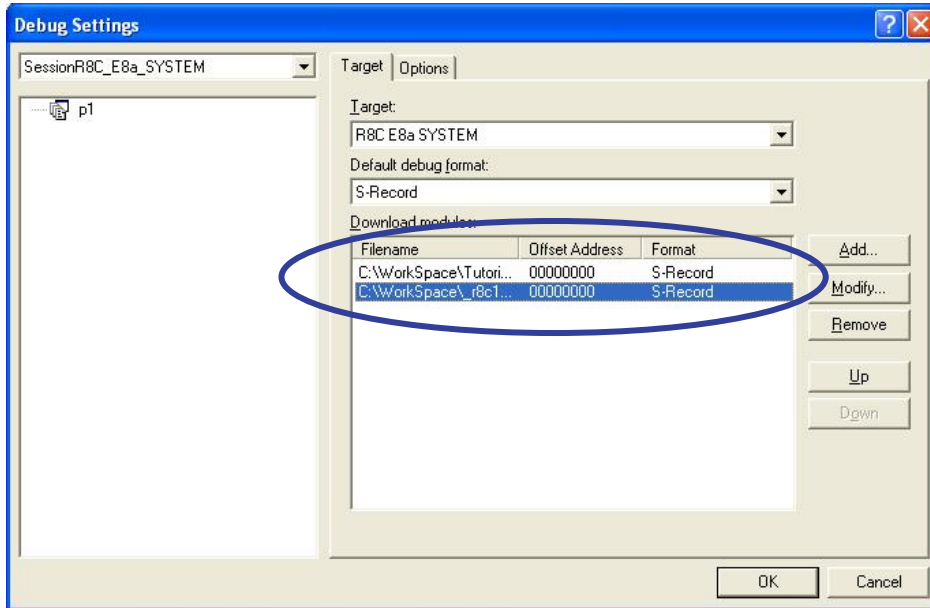
The rest is the same as described in paragraph (6) of Section 4.5.

Workspaces can also be restarted by double-clicking a workspace file for the HEW (extension ".hws") directly from Windows Explorer, etc.

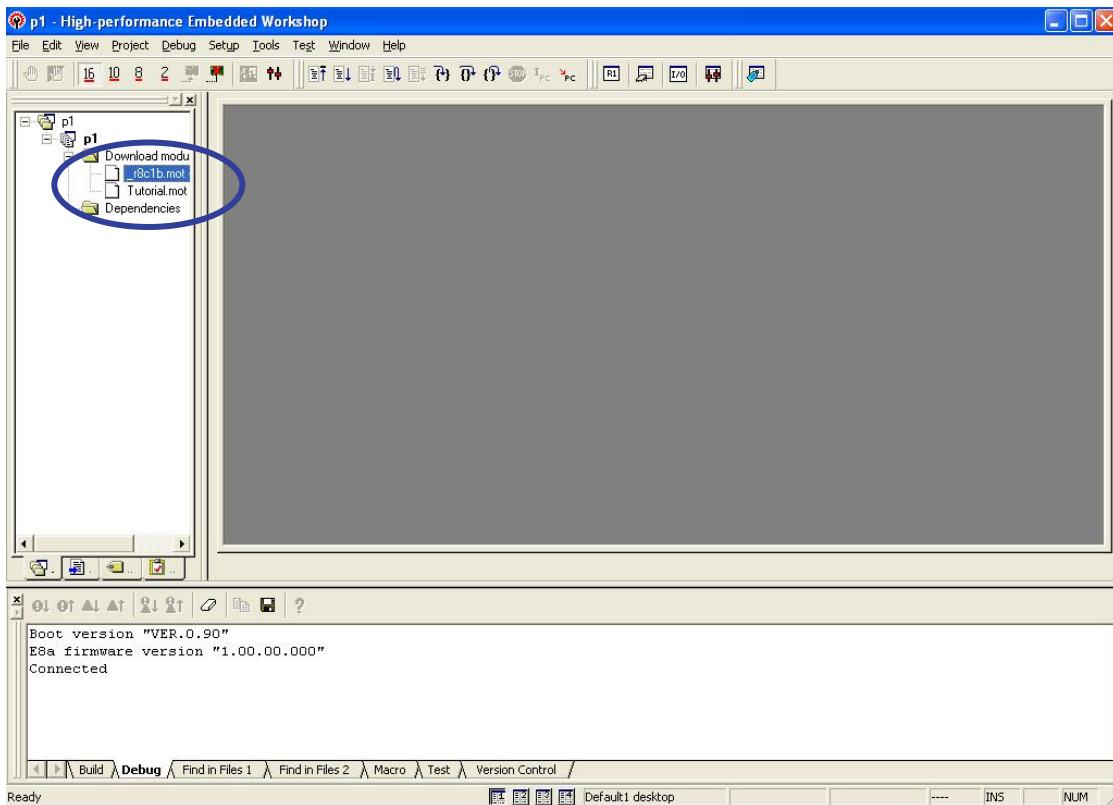
Here, we've presented the method for writing to the flash memory using the E8a and the High-performance Embedded Workshop. To write a large amount of data as in a mass-production process, we recommend using the dedicated flash write software "Flash Development Toolkit" instead of the High-performance Embedded Workshop.

The Flash Development Toolkit comes with the facility to read out flash memory contents, the facility to erase the flash memory one block at a time and the facility to perform a series of write operations with the press of a button, the facilities that are not available in the High-performance Embedded Workshop.

(22) You might want to register multiple files in advance and select one of those files when writing to the flash memory. In that case, register two or more files as you wish in the Debug Setting dialog box.



That way, you can select the file you want from the workspace window and download (write) it.



In the Flash Memory Data Write mode, only one file can be downloaded (written in) at a time. You cannot write multiple files, one appended to the other.

## 5. Frequently Asked Questions

### 5.1 Are there any precautions to take when verifying the program behavior with a microcomputer alone after debugging is complete?

The following presents the cases (involving the use of the R8C family microcomputer) where problems occurred in verifying the program behavior with a microcomputer alone, specifically for reasons of functional specifications of the E8a. In the Flash Memory Data Write mode, data can be written to the entire ROM area as possible with an ordinary ROM writer. To verify the program behavior with the actual chip alone, pay attention to the following.

#### **(1) When the E8a is unconnected, whether the RESET and NMI pins are processed correctly**

There was a case in which when the E8a was removed, the RESET or NMI pin was stuck in the low state, making the microcomputer unable to operate normally.

(Example: Unless there is a pullup resistor for the NMI pin on the board side, the pin may mistakenly be processed as NC when the E8a is removed.)

#### **(2) Whether the E8a occupied area is taken into account in a sum check**

Some ROM area is occupied the E8a as stated in the manual of each device. If all of ROM areas are sum-checked in the initialize part of the user program, as the memory contents of these areas differ between the Flash Memory Erase Before Start and the Flash Memory Data Write modes, so does the SUM value. To sum-check the ROM, take this difference into account when you create a program.

#### **(3) About WDT settings**

The E8a, for reasons of emulator specifications, sometimes turns the WDT off during debug. Therefore, a reset may happen to occur while operating with the actual chip alone, even when no resets are generated.

#### **(4) Whether the stack pointer is initialized in the user program**

When debugged with the E8a emulator, the debugger initializes the stack pointer. Unless the stack pointer is initialized, a symptom will occur that the program, although it operated normally when debugged with the E8a, does not operate normally when run singly. Always be sure that the stack pointer is initialized in the user program.

## 5.2 When a dialog box is displayed asking for ID code, what should be input?

When the E8a emulator starts up, it compares the ID code written in the microcomputer's internal flash memory and the ID code supplied by the user in the ID code confirmation dialog box. The ID code value written in the microcomputer's internal flash memory differs depending on the .ID set in the source file or option specification by the user or whether the -ID option or the fixed interrupt vector is set. Note 1

See the table below to ensure that the correct ID code is set. The ID code in a blank-state microcomputer is FFFFFFFF in hexadecimal.

.ID or -ID specification		ID code value
Specified <sup>Note 2</sup>		Values set by user
Not specified <sup>Note 3</sup>	Fixed interrupt vector set	00000000000000 in hex
	Fixed interrupt vector not set	FFFFFFFFFFFFFF in hex

Note 1 There are following methods to set ID code.

- Use the assembler directive ".ID" (recommended)

Example: To set ID code to 1234567890abcd

(1) For the assembler source

```
.id "#1234567890abcd"
```

(2) For the C source

```
-asm(".id\"#1234567890abcd\"");
```

- Use the "-ID" option of the load module converter LMC30

-ID is an optional facility of the LMC30. If the assembler directive ".ID," ".OFSREG" or ".PROTECT" is written in the file, this option has no effect.

Note 2 Be aware that the values set using the directive command ".ID" of the assembler and the "-ID" option of the LMC30 command are reflected differently in x30 and mot files. Therefore, we recommend using the assembler directive ".ID."

When specified by .ID → The value is reflected in both x30 and mot files

When specified by -ID → The value is reflected in only mot file

Due to this difference, if the -ID option and a x30 file are used when writing data to the flash, a problem will occur that the ID code set in the user program is not reflected in the actual chip.

Note 3 If no ID codes are specified by the user, ID code normally is 00000000000000 in hex because the top address of the fixed interrupt vector is assigned to the ID. Furthermore, if the vector that contains ID code is not written in the user program, the initial value of the flash, or FFFFFFFF in hex, is assumed.

## 6. Frequency Encountered Errors and the Corrective Actions to Take

### (1) Driver Error : Illegal driver inter face select

Unable to communicate with the E8a. Unplug the USB cable from the PC briefly and plug it back into position.

### (2) Driver Error: No available communication devices found.

The E8a is not connected. Check the USB cable.

### (3) Connector disconnected. Please, connect and press <Enter> key.

The user board and the E8a are not connected. Connect the user board and the E8a, and then press the Enter key.

### (4) Flash memory erase error ! Change device

Failed to erase the flash memory. Replace the microcomputer with another.

### (5) Flash memory write error

Failed to write to the flash memory.

This error often occurs in cases where the microcomputer is operating with a voltage lower than the guaranteed write voltage.

### (6) Communication timeout error

No responses are received from the microcomputer.

Visit the Renesas website and see the "Frequently Asked Questions" regarding the E8a.

## 7. Related Documents

The E8a emulator and the HEW have numerous other convenient facilities and features not discussed in this document. Detailed specification, technical information, limitations and other useful information on each product are described in the related documents listed below. Please see these manuals along with this document.

### [E8a emulator related documents]

- E8a Emulator User's Manual
- E8a Emulator User's Manual, Separate Volume (Supplementary explanation about the R8C and the precautions to take when connecting it)

### [High-performance Embedded Workshop related documents]

- High-performance Embedded Workshop User's Manual
- High-performance Embedded Workshop Release Notes

### [CPU related documents]

- R8C Family Hardware Manual
- R8C Family Software Manual

### [M16C series C compiler related documents]

- M3T-NC30WA C Compiler Guidebook (C Compiler Package for the M16C Series)
- M3T-NC30WA Assembler User's Manual (C Compiler Package for the M16C Series)

To see more information on the E8a emulator, please visit the Renesas websites given below.

Japan site: <http://japan.renesas.com/e8a>

Global site: <http://www.renesas.com/e8a>

## Home Page and Where to Contact for Support

Renesas Technology home page  
<http://www.renesas.com/>Where to contact  
<http://www.renesas.com/inquiry>

## Revision Record

Rev.	Issue date	Contents of revision	
		Page	Points
1.00	2009.2.13	—	First edition issued



Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
  - (1) artificial life support devices or systems
  - (2) surgical implantations
  - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
  - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.