REJ10B0023-0200H Everywhere you imagine. RENESAS

SuperH RISC engine

High-Performance Embedded Workshop 3

Tutorial

User's Manual

Renesas Microcomputer Development Environment System

Rev.2.00
Revision Date: Jun. 25, 2004

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

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

Preface

This tutorial will introduce you to the basic usage of the High-Performance Embedded Workshop 3 (hereafter referred to as *HEW*). The tutorial describes a series of usage such as how to invoke the HEW, how to create and modify a project, how to build a program, and how to invoke a simulator/debugger from the HEW. For details on the HEW, the C/C++ compiler, assembler, optimizing linkage editor, and simulator/debugger, refer to the following manuals.

- SuperH™ RISC engine High-Performance Embedded Workshop 3 User’s Manual
- SuperH™ RISC engine C/C++ Compiler, Cross Assembler, Optimizing Linkage Editor User’s Manual

For details on the SuperH™ RISC engine CPU, refer to a programming manual or a hardware manual of the corresponding CPU.

This tutorial is written assuming that all the tools included in the SuperH™ RISC engine C/C++ compiler Package are installed.

If "->" is written inside [and], the left hand side of "->" means the menu title and the right hand side of "->" means a menu option of the menu. For example, in figure 1, the menu option [Exit] in the [File] menu is described as [File->Exit].

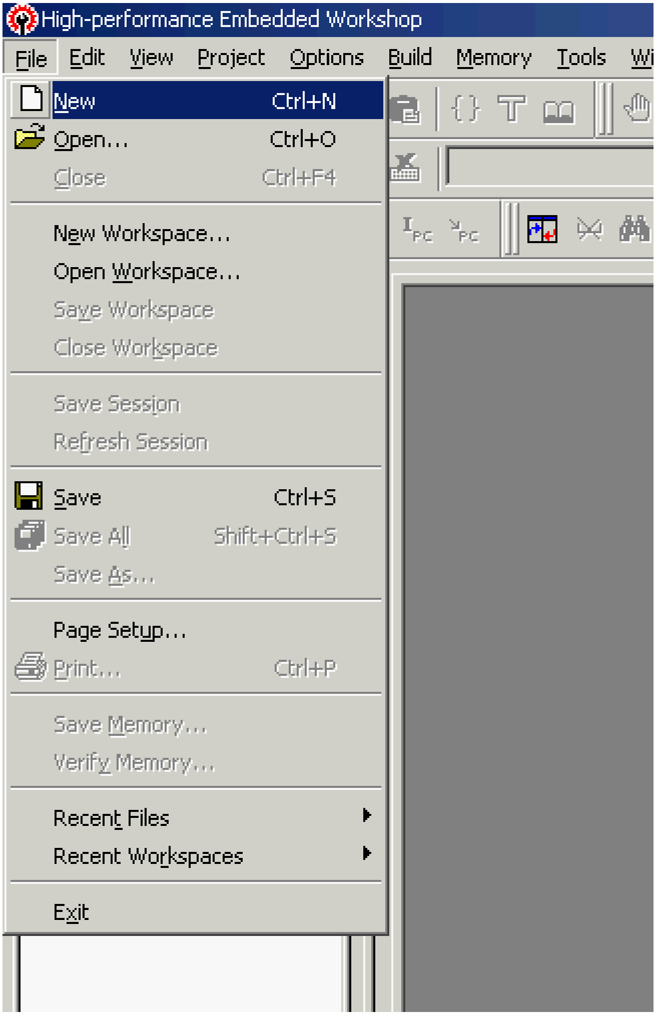


Figure 1 A Menu of the HEW

Contents

Section 1 Invoking the HEW

After the installation of the HEW, the installer creates a folder whose name is [Renesas High-Performance Embedded Workshop] in the [Program] folder of the [Start] menu of the Windows®. In the [Renesas High-Performance Embedded Workshop] folder, short cuts of the HEW support files will be registered.

If you click [High-Performance Embedded Workshop 3] from the menu, the HEW will be invoked and the [Welcome!] dialog box (figure 1.1) will be displayed. You can open the project promptly without the [Welcome!] dialog box if you change the setting via [Tools->Options]. For details of this setting, refer to chapter 6, "Customizing the Environment", of the High-Performance Embedded Workshop 3 User's Manual.

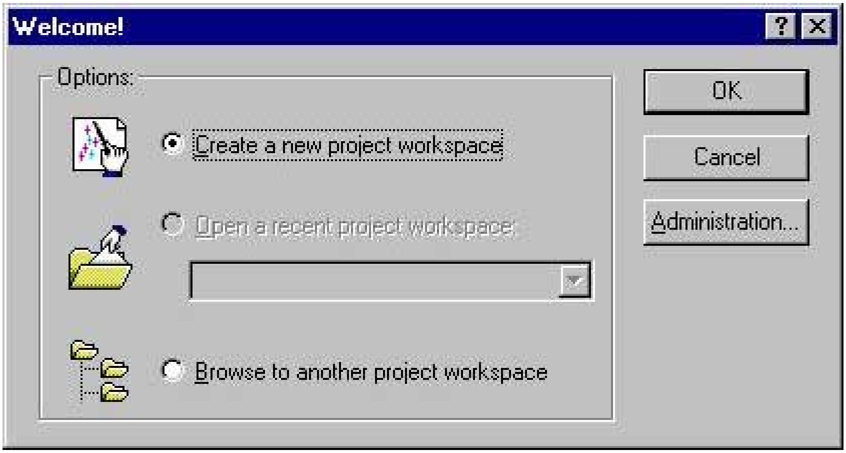

Welcome!
Options:
● Create a new project workspace
○ Open a recent project workspace:
○ Browse to another project workspace
OK
Cancel
Administration...

Figure 1.1 Welcome! Dialog Box

If you use the HEW for the first time or if you want to work on a new project, select the [Create a new project workspace] radio button and click [OK]. If you want to work on an existing project, select the [Open a recent project workspace] or [Browse to another project workspace] radio button and click [OK].

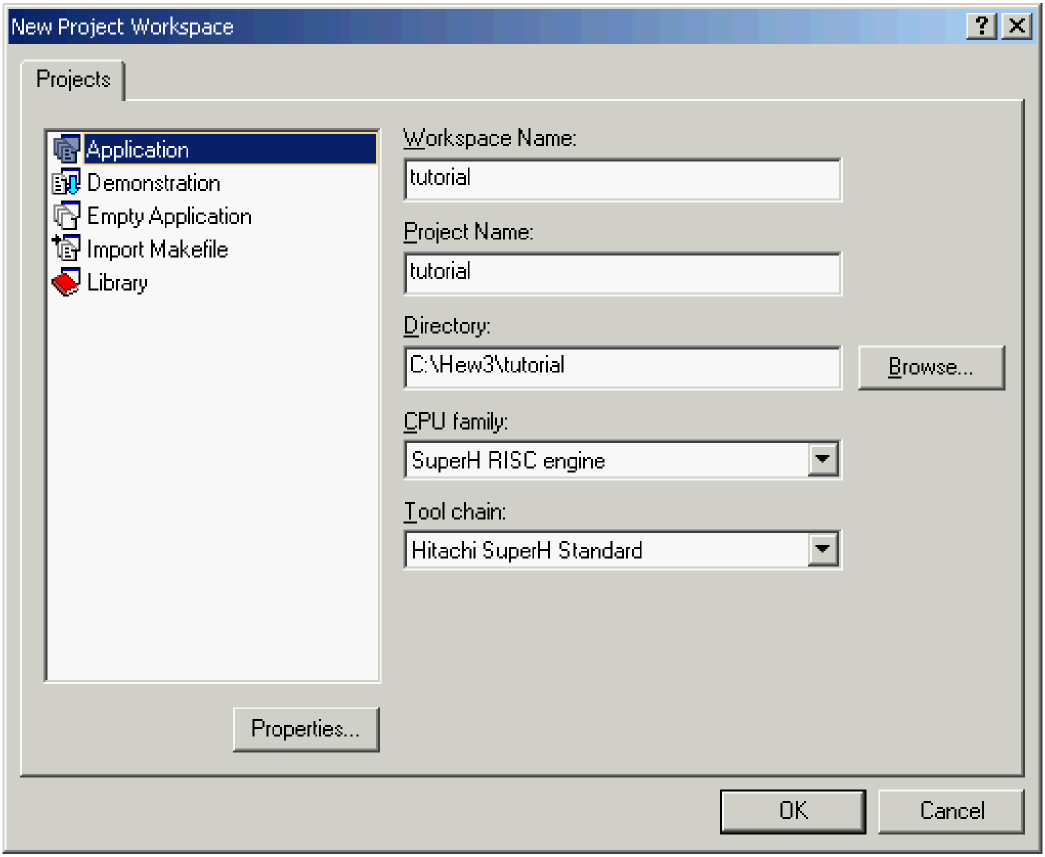
In this tutorial, select the [Create a new project workspace] radio button and click [OK].

Section 2 Creating a Project

2.1 Creating a New Workspace

When you have selected the [Create a new project workspace] radio button and clicked [OK] on the [Welcome!] dialog box, the [New Project Workspace] dialog box (figure 2.1), which is used to create a new workspace and project, will be launched. You will specify a workspace name (When a new workspace is created, the project name is the same as the default), a CPU family, a project type, and so on, on this dialog box.

Enter “tutorial”, for example, in the [Workspace Name] field, then the [Project Name] field will show “tutorial” and the [Directory] field will show “C:\Hew2\tutorial”. If you want to change the project name, enter a new project name manually in the [Project Name] field. If you want to change the directory used for the new workspace, click the [Browse…] button and specify a directory, or enter a directory path manually in the [Directory] field.



Workspace Name: tutorial

Project Name: tutorial

Directory: C:\Hew3\tutorial

CPU family: SuperH RISC engine

Tool chain: Hitachi SuperH Standard

Figure 2.1 New Project Workspace Dialog Box

Table 2.1 Items of [Project type:]

Items of [Project type:]	Description
Application	This project is used to create a program that includes the initial routine files written in C/C++ or assembly language.
Demonstration	This project is used to create the program for demonstration written in C/C++ or assembly language.
Empty Application	This project is used to set up the toolchain (no file is to be generated).
Library	This project is used to create a library file (no file is to be generated).

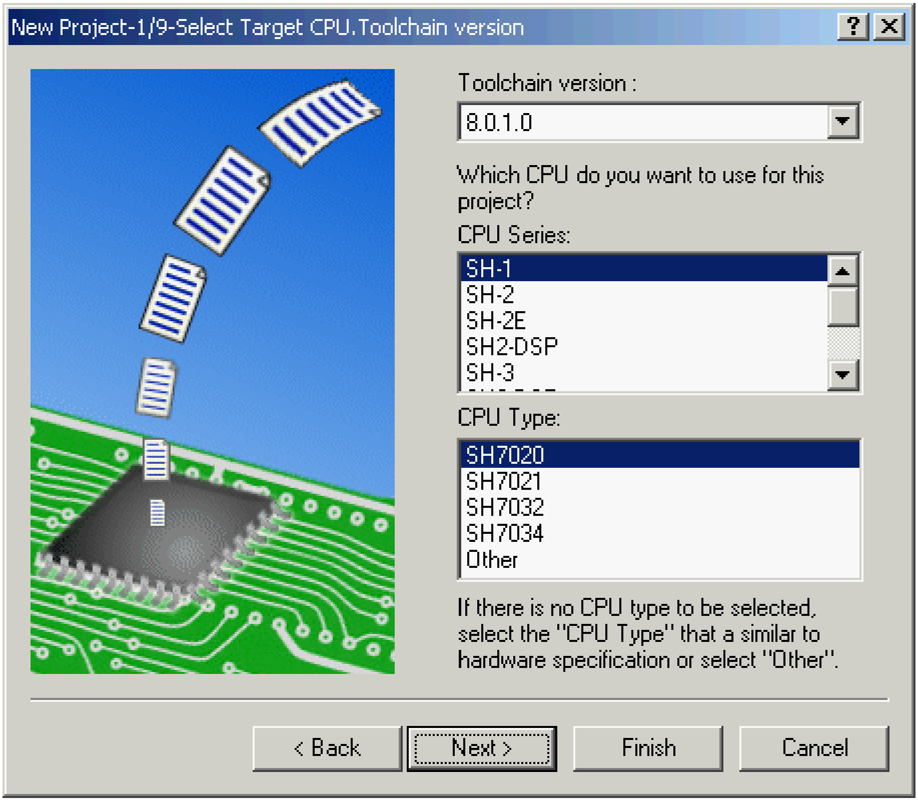
2.2 Selecting the Target CPU

When you click [OK] on the [New Project Workspace] dialog box, the project generator will be invoked. Start by selecting the CPU that you will be using. CPU types shown in the [CPU Type:] list are classified into the CPU series shown in the [CPU Series:] list.

The selected items in the [CPU Series:] list box and the [CPU Type:] list box specify the files to be generated. Select the CPU type of the program to be developed. If the CPU type which you want to select is not displayed in the [CPU Type:] list, select a CPU type with similar hardware specifications or select [Other].

Clicking [Next>] moves to the next display. Clicking [<Back] moves to the previous display or the previous dialog box. Clicking [Finish] opens the [Summary] dialog box. Clicking [Cancel] returns the display to the [New Project Workspace] dialog box. [<Back], [Next>], [Finish], and [Cancel] are common buttons of all the wizard dialog boxes.

In this tutorial, select [SH-1] in the [CPU Series:] list and select [SH7020] in the [CPU Type:] list (figure 2.2). Then click [Next >].



New Project-1/9-Select Target CPU.Toolchain version

Toolchain version : 8.0.1.0

Which CPU do you want to use for this project?

CPU Series:

- SH-1
- SH-2
- SH-2E
- SH2-DSP
- SH-3

CPU Type:

- SH7020
- SH7021
- SH7032
- SH7034
- Other

If there is no CPU type to be selected, select the "CPU Type" that a similar to hardware specification or select "Other".

Figure 2.2 Selecting the Target CPU (Step 1)

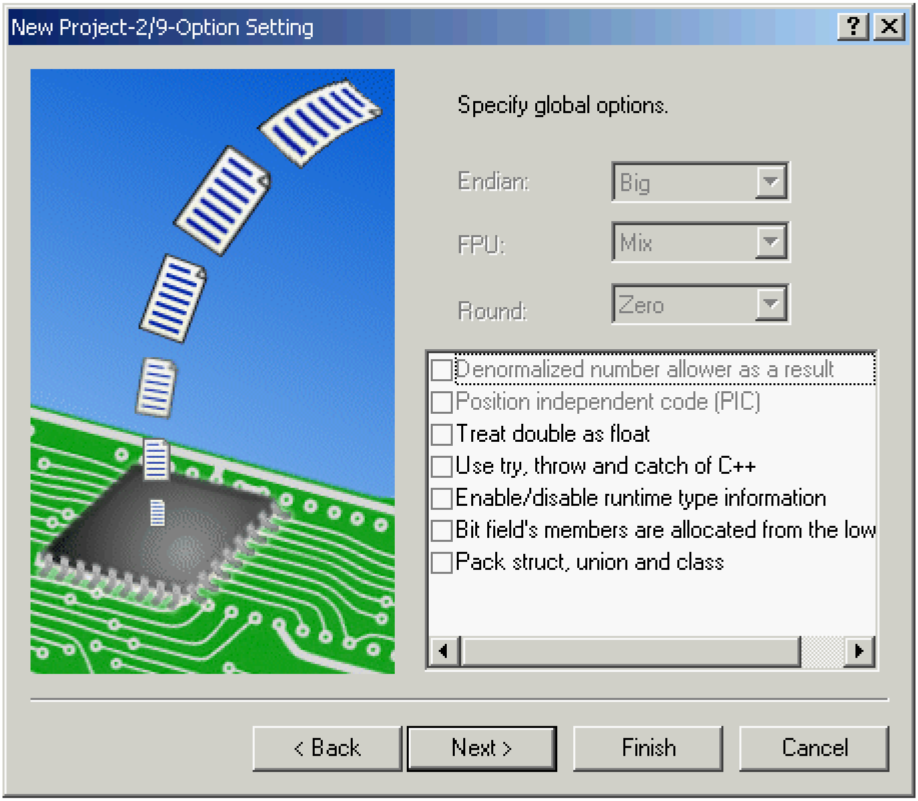
2.3 Option Setting

Clicking the [Next>] button on the Step-1 screen opens the dialog box shown in figure 2.3.

On this screen, you can specify options common to all project files.

Settings for these options can be modified in correspondence with the CPU series selected in step 1 screen. To change the option settings after a project has been created, select the CPU page from the [Options -> SuperH RISC engine Standard Toolchain] menu item of the HEW window.

Click the [Next>] button without changing the setting. The Step-3 screen will be displayed.



Specify global options.

Endian: Big

FPU: Mix

Round: Zero

□ Denormalized number allower as a result
□ Position independent code (PIC)
□ Treat double as float
□ Use try, throw and catch of C++
□ Enable/disable runtime type information
□ Bit field's members are allocated from the low
□ Pack struct, union and class

Figure 2.3 Option Setting (Step 2)

2.4 Setting the Contents of Files to be Generated

Click the [Next>] button on the Step-2 screen to display the screen shown in figure 2.4.

On this screen, specify information that is necessary to generate files.

If [Use I/O Library] is checked, the standard I/O library can be used.

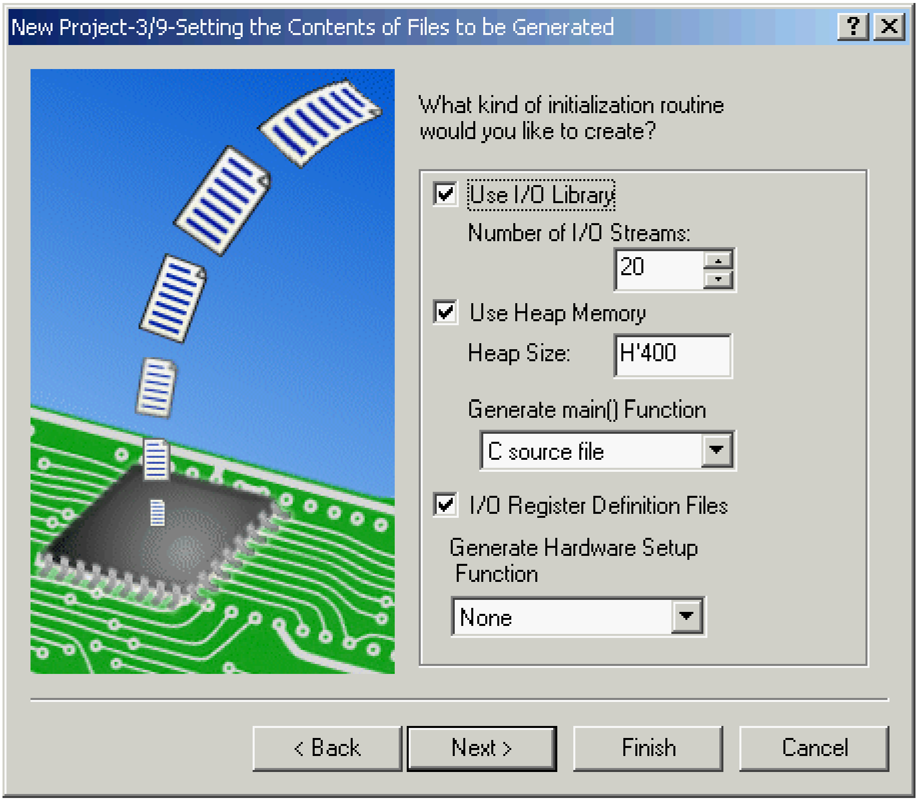
If [Use Heap Memory] is checked, the function sbrk() for heap area management will be generated. Specify the number of bytes of the heap area in the [Heap Size:] edit box.

Select whether or not to generate a sample code for the main function from the [Generate main() Function] drop-down list.

If [I/O Register Definition Files] is checked, an I/O register definition file, which is written in C language, can be created. Then select whether or not to generate a sample code for the program that makes initial settings of the I/O registers from the [Generate Hardware Setup Function] drop-down list.

In this tutorial, check [Use I/O Library] and click [Next >].

NOTE
If you want to use an existing main function, uncheck [Generate main() Function], add the file of the function after generating the project. If the name of the function differs from main, change the caller of the function in resetprg.c. For the contents of such sample files as a vector table definition file or I/O register definition file that will be generated by the project generator, check the description in the hardware manual for the target CPU.



New Project-3/9-Setting the Contents of Files to be Generated

What kind of initialization routine would you like to create?

☑ Use I/O Library

Number of I/O Streams: 20

☑ Use Heap Memory

Heap Size: H'400

Generate main() Function: C source file

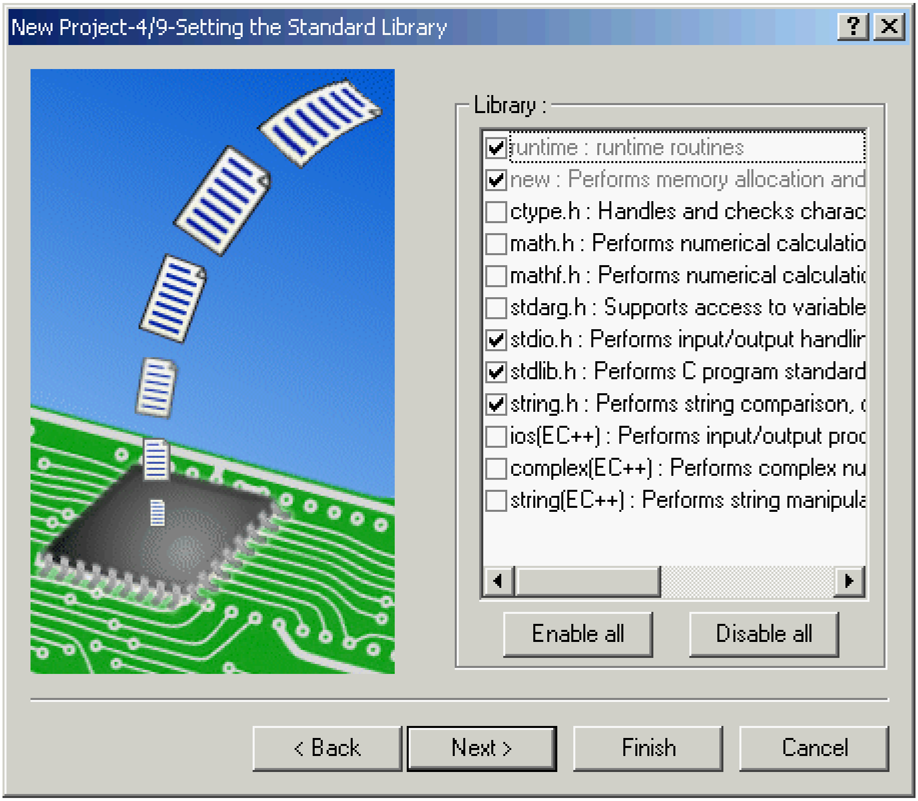
☑ I/O Register Definition Files

Generate Hardware Setup Function: None

Figure 2.4 Setting the Contents of Files to be Generated (Step 3)

2.5 Setting the Standard Library

The screen shown in figure 2.5 is displayed when the [Next>] button is clicked in the Step-3 screen. This screen is used to set details of compilation by the C/C++ compiler. To change the standard library configuration after a project has been created, select the CPU page in [Options -> SuperH RISC engine Standard Toolchain] of the HEW. In this tutorial, click the [Next>] button without changing the settings.



New Project-4/9-Setting the Standard Library

Library :

- ☑ runtime : runtime routines
- ☑ new : Performs memory allocation and
- □ ctype.h : Handles and checks charac
- □ math.h : Performs numerical calculatio
- □ mathf.h : Performs numerical calculati
- □ stdarg.h : Supports access to variable
- ☑ stdio.h : Performs input/output handlir
- ☑ stdlib.h : Performs C program standard
- ☑ string.h : Performs string comparison, c
- □ ios(EC++) : Performs input/output proc
- □ complex(EC++) : Performs complex nu
- □ string(EC++) : Performs string manipula

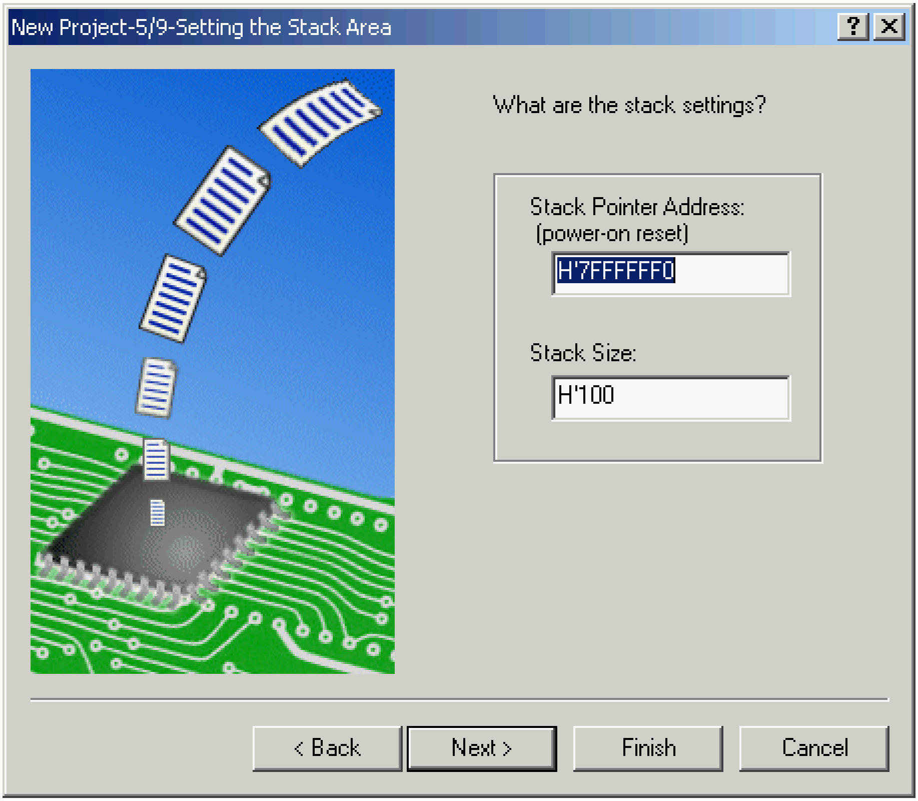
Figure 2.5 Setting the Standard Library (Step 4)

2.6 Setting the Stack Area

The screen shown in figure 2.6 is displayed when the [Next>] button is clicked in the Step-4 screen.

This screen is used to specify the stack area.

The initial value of the stack area differs depending on [CPU Type:] selected in the Step-1 screen. To change the stack size after a project has been created, select the [Project -> Edit Project Configuration] menu item of the HEW window. In this tutorial, since SH7020 has been selected for [CPU Type:] in the Step-1 screen, [Stack Pointer Address:] is set to H’7FFFFFF0 and [Stack Size:] is set to H’100. In this tutorial, click the [Next>] button without changing the settings.



New Project-5/9-Setting the Stack Area

What are the stack settings?

Stack Pointer Address:
(power-on reset)
H'7FFFFFF0

Stack Size:
H'100

Figure 2.6 Setting the Stack Area (Step 5)

The stack area is defined in stacksct.h that is generated by the HEW. If stacksct.h has been edited in an editor, its modification after you have selected the [Project -> Edit Project Configuration] menu item of the HEW will not be available.

2.7 Setting the Vector

Clicking the [Next>] button in the Step-5 screen displays the screen shown in figure 2.7.

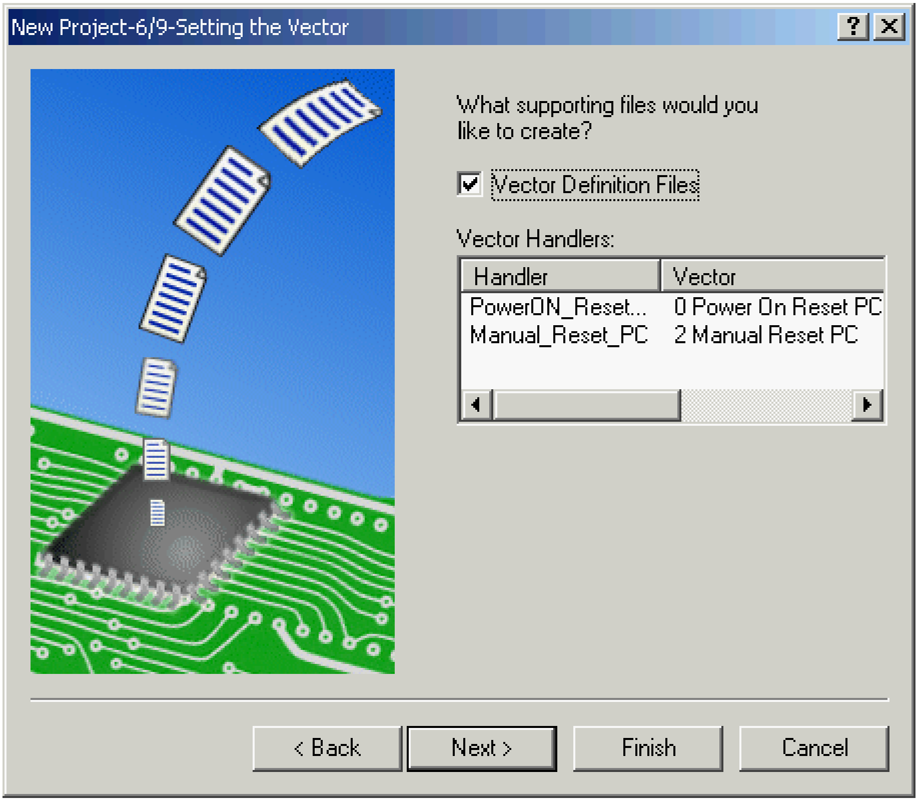
In this screen, a vector is specified.

If [Vector Table Definition Files] is checked, the HEW creates a vector table definition file.

The [Handler] column of [Vector Handlers:] displays the handler program name, and the [Vector] column displays the vector description. To change the handler program, click the name of the handler program to be changed, and enter a new name.

If the handler name in the [Vector Handlers:] list is changed, the HEW does not create the reset program (resetprg.c).

In this tutorial, click the [Next>] button without changing the settings.



New Project-6/9-Setting the Vector

What supporting files would you like to create?

☑ Vector Definition Files

Vector Handlers:

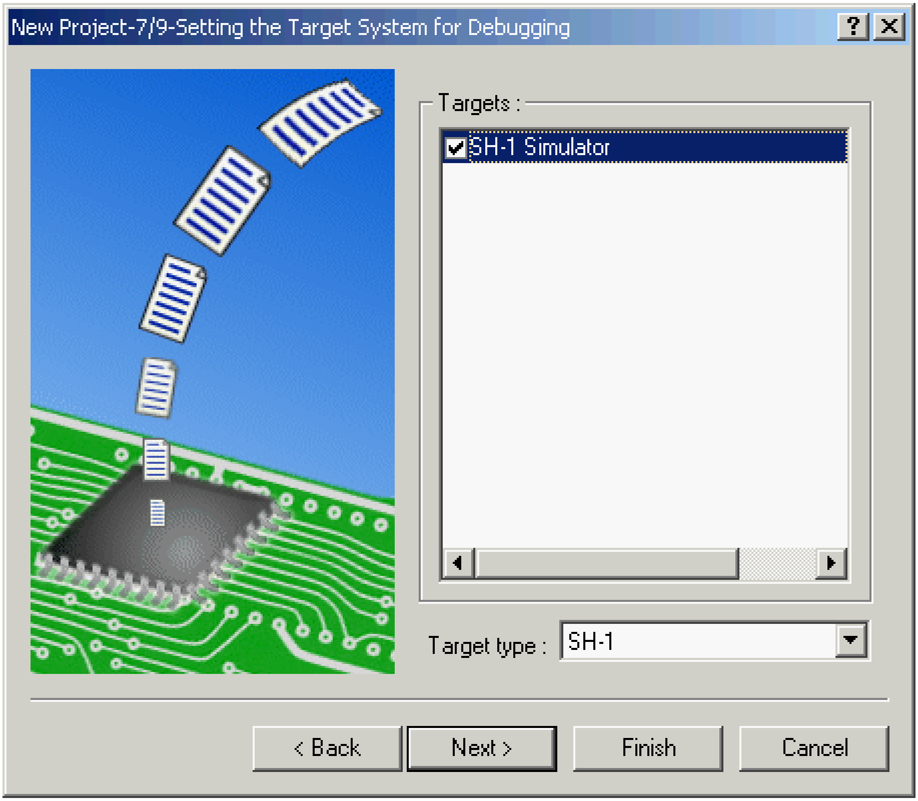
Handler	Vector
PowerON_Reset...	0 Power On Reset PC
Manual_Reset_PC	2 Manual Reset PC

Figure 2.7 Setting the Vector (Step 6)

2.8 Setting the Target System for Debugging

When the [Next>] button is clicked in the Step-6 screen, the screen shown in figure 2.8 is displayed. This screen is used to specify the target system for debugging. Select (check) the target for debugging from the list under [Targets:]. Selection of no target or of multiple targets is allowed.

In this tutorial, select [SH-1 Simulator] and then click the [Next>] button.



☑ SH-1 Simulator

Target type : SH-1

Figure 2.8 Setting the Target System for Debugging (Step 7)

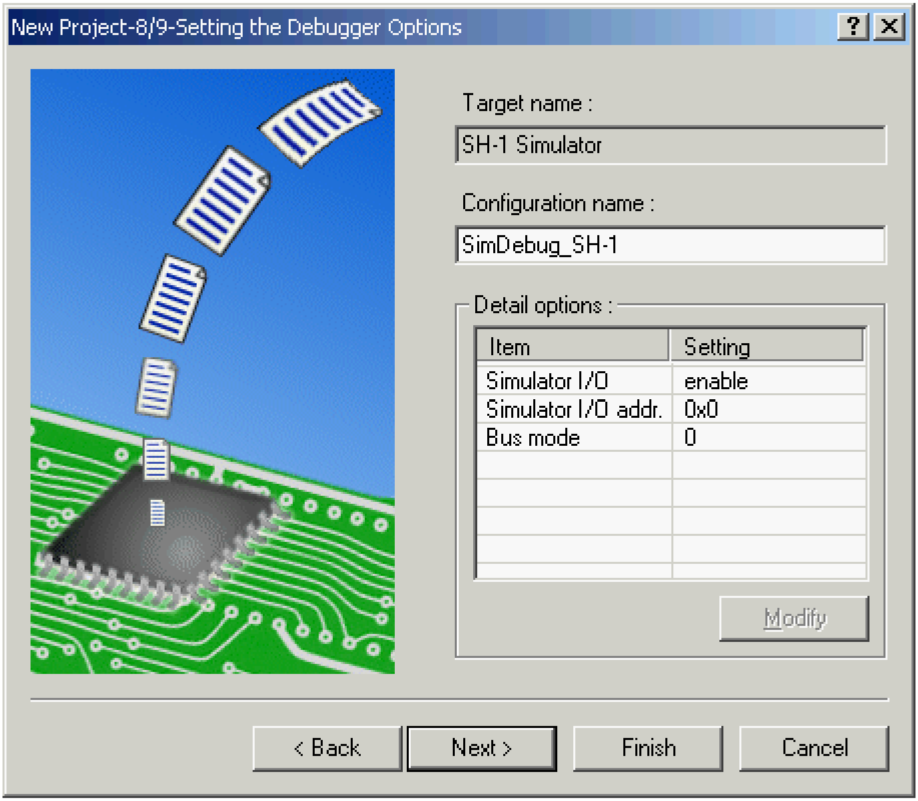
If you change [Target type:], you can specify the other target system for debugging.

2.9 Setting the Debugger Options

When the [Next>] button is clicked in the Step-7 screen, the screen shown in figure 2.9 is displayed. This screen is used to specify the optional settings for the selected target for debugging.

By default, the HEW creates two configurations, [Release] and [Debug]. When a target for debugging is selected, the HEW creates another configuration (The name of the target is included). The name of the configuration can be modified in [Configuration name:]. Options to do with the target for debugging are displayed under [Detail options:]. To change the settings, select [Item] and then click [Modify]. When items for which modification is not possible are selected, [Modify] remains grayed even if [Item] is selected.

In this tutorial, click the [Next>] button without changing the settings.



New Project-8/9-Setting the Debugger Options

Target name : SH-1 Simulator

Configuration name : SimDebug_SH-1

Detail options :

Item	Setting
Simulator I/O	enable
Simulator I/O addr.	0x0
Bus mode	0

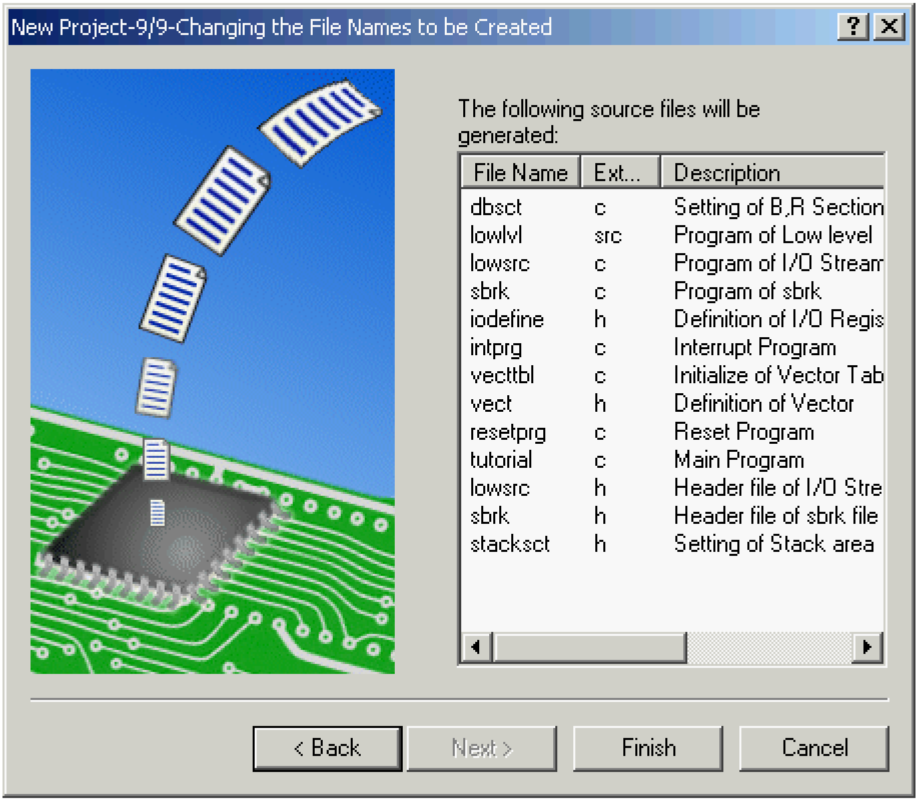
Figure 2.9 Setting the Debugger Options (Step 8)

2.10 Changing the File Names to be Created

When the [Next>] button is clicked in the Step-8 screen, the screen shown in figure 2.10 is displayed.

The screen displays a list of files created by HEW according to the previous settings. The [File Name] column in the list shows a file name, [Extension] shows an extension, and [Description] shows the description of a file. The file name can be changed. To change a file name, select it by clicking it and change it to a new file name.

In this tutorial, click the [Finish] button without changing the settings. When the button is clicked, the [Summary] dialog box will be displayed.



The following source files will be generated:

File Name	Ext...	Description
dbsct	c	Setting of B,R Section
lowlvl	src	Program of Low level
lowsrc	c	Program of I/O Stream
sbrk	c	Program of sbrk
iodefine	h	Definition of I/O Regis
intprg	c	Interrupt Program
vecttbl	c	Initialize of Vector Tab
vect	h	Definition of Vector
resetprg	c	Reset Program
tutorial	c	Main Program
lowsrc	h	Header file of I/O Stre
sbrk	h	Header file of sbrk file
stacksct	h	Setting of Stack area

Figure 2.10 Changing the File Names to be Created (Step 9)

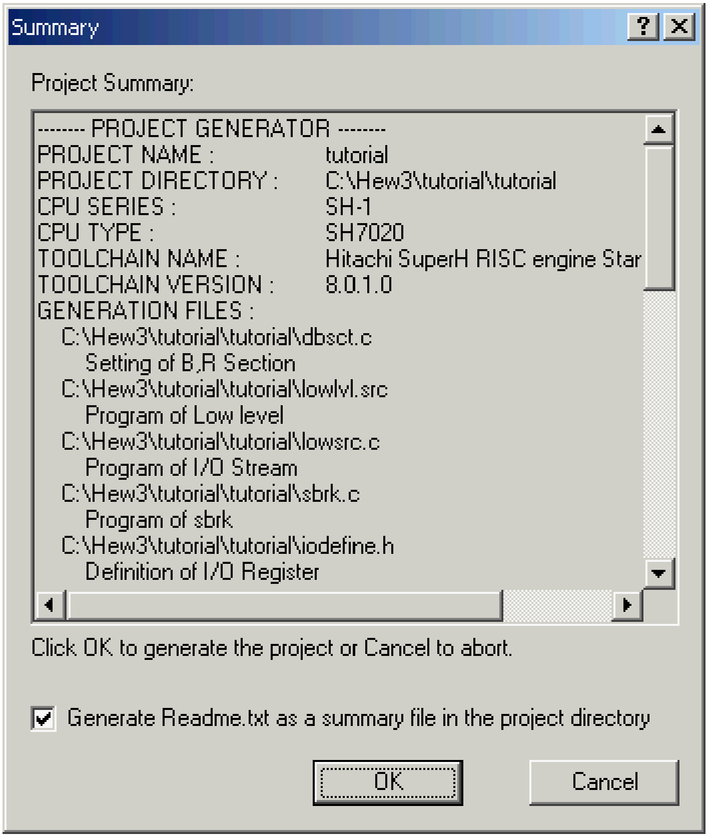
NOTE

A file with an extension “h” or “inc” (shown in the [Extension] column) is an include file. If you change the file name of an include file, the file name at the include directive have to be modified.

2.11 Confirming Settings (Summary Dialog Box)

The project generator shows a list of generated files on the [Summary] dialog box (figure 2.11). Confirm the contents of the dialog box and click [OK].

When [Generate Readme.txt as a summary file in the project directory] checkbox is checked, the project information displayed on the [Summary] dialog box will be stored in the project directory under the text file name "Readme.txt".



```
-------- PROJECT GENERATOR --------
PROJECT NAME :              tutorial
PROJECT DIRECTORY :         C:\Hew3\tutorial\tutorial
CPU SERIES :                SH-1
CPU TYPE :                  SH7020
TOOLCHAIN NAME :            Hitachi SuperH RISC engine Star
TOOLCHAIN VERSION :         8.0.1.0
GENERATION FILES :
   C:\Hew3\tutorial\tutorial\dbsct.c
      Setting of B,R Section
   C:\Hew3\tutorial\tutorial\lowlvl.src
      Program of Low level
   C:\Hew3\tutorial\tutorial\lowsrc.c
      Program of I/O Stream
   C:\Hew3\tutorial\tutorial\sbrk.c
      Program of sbrk
   C:\Hew3\tutorial\tutorial\iodefine.h
      Definition of I/O Register
```

Click OK to generate the project or Cancel to abort.

☑ Generate Readme.txt as a summary file in the project directory

Figure 2.11 Summary Dialog Box

Then the HEW will open a project generated by the project generator.

By default, the HEW has:

The [Workspace] window (on the left hand side of the HEW), which shows the contents of the project, the [Output] window (on the lower side of the HEW), which shows outputs from a build or a string search, and the editor window (on the right hand side of the window), which is used to edit text files. Refer to the High-Performance Embedded Workshop 3 User's Manual for how to change the state of the HEW and how to use each window such as the editor window.

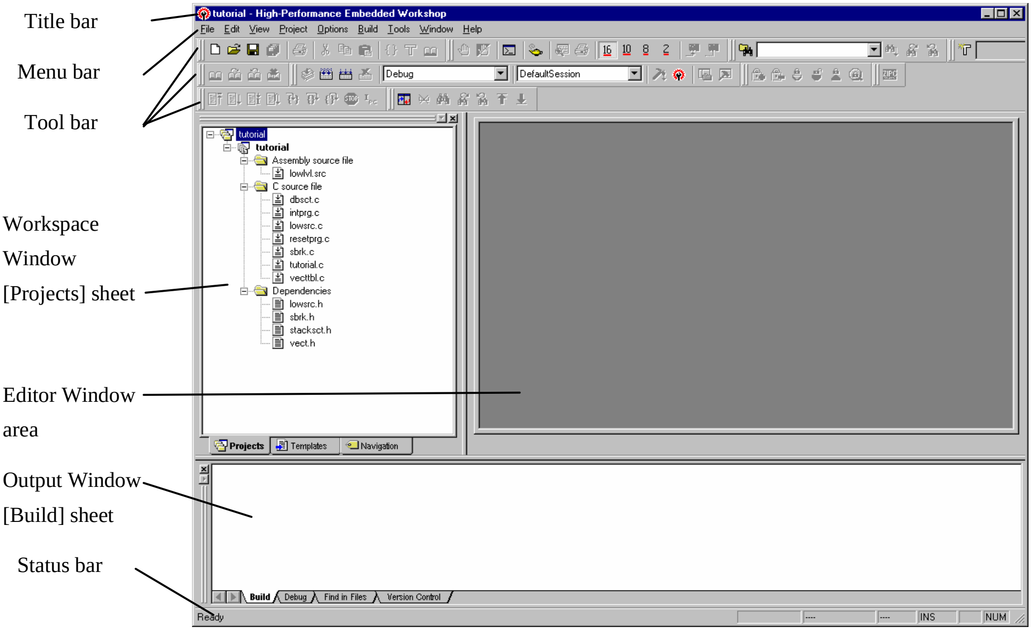
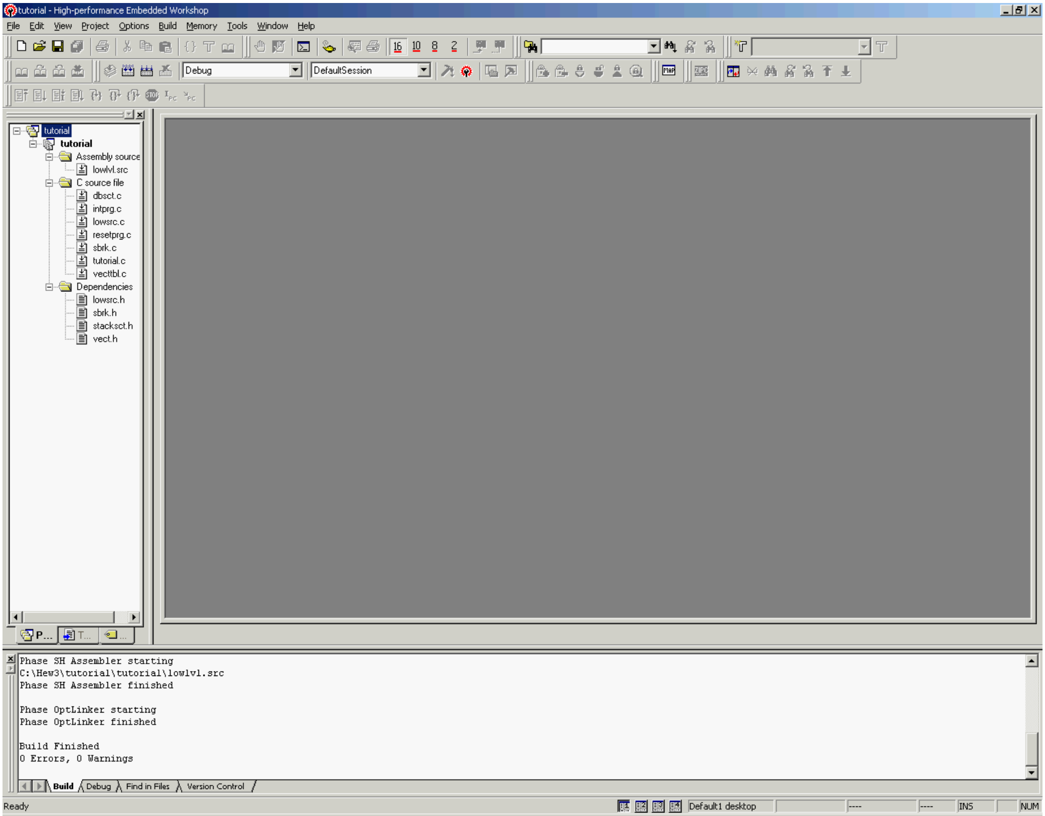


Figure 2.12 Sub-windows of the HEW

The project generated by the project generator includes minimum options for the C/C++ compiler, the assembler, the inter-module optimizer and the object converter. Thus you can build the project (figure 2.13) by selecting [Build->Build]. Selecting the button (shown below) on the toolbar works in the same way as selecting [Build->Build].

Those minimum options are set by default for each tool. When you add a new file to the project, the minimum options will automatically be added to the file. You do not have to specify options other than those that are specific to each such file.



```
Phase SH Assembler starting
C:\Hew3\tutorial\tutorial\lowlvl.src
Phase SH Assembler finished

Phase OptLinker starting
Phase OptLinker finished

Build Finished
0 Errors, 0 Warnings
```

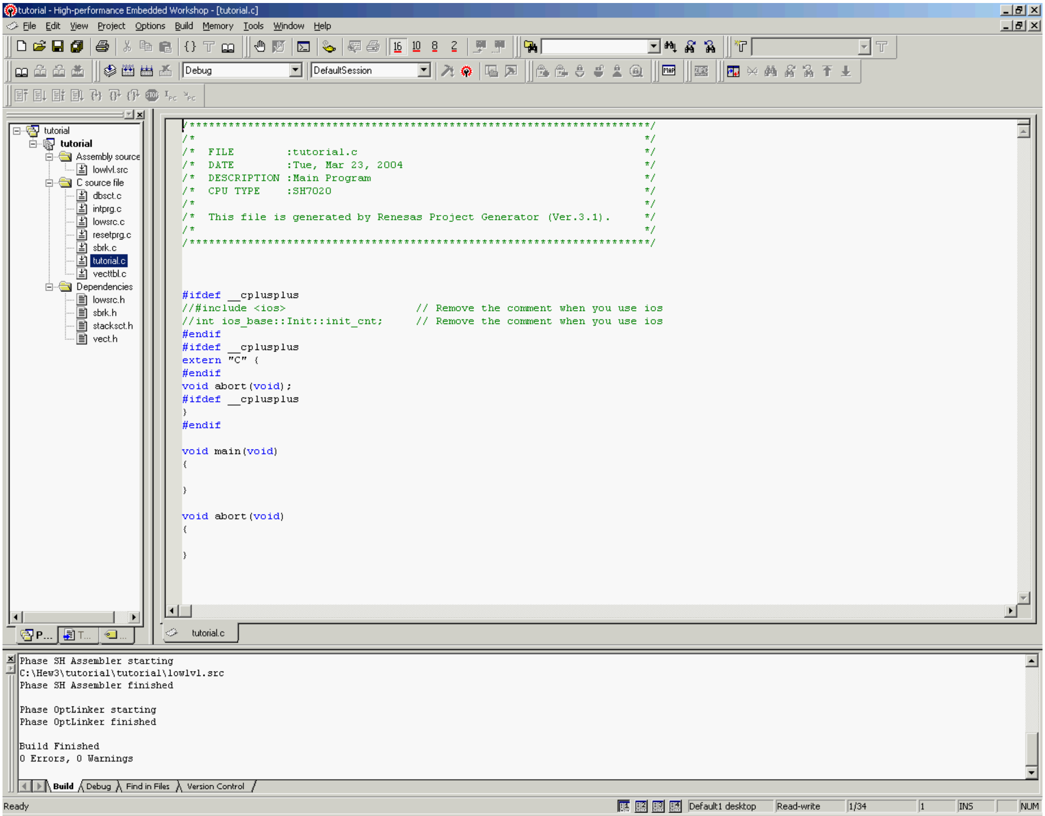
Figure 2.13 The HEW after Building a Project

Section 3 Modifying the Project

When you create a project using the project generator, basic program such as reset routine and initialization of RAM area will already be given in the project. This tutorial describes how to modify a file and how to add a file to the project.

3.1 Editing and Creating a Source Program File

To open and edit a file in the project, double click the file in the [Projects] sheet of the [Workspace] window. In this tutorial, double click “tutorial.c”. (figure 3.1)

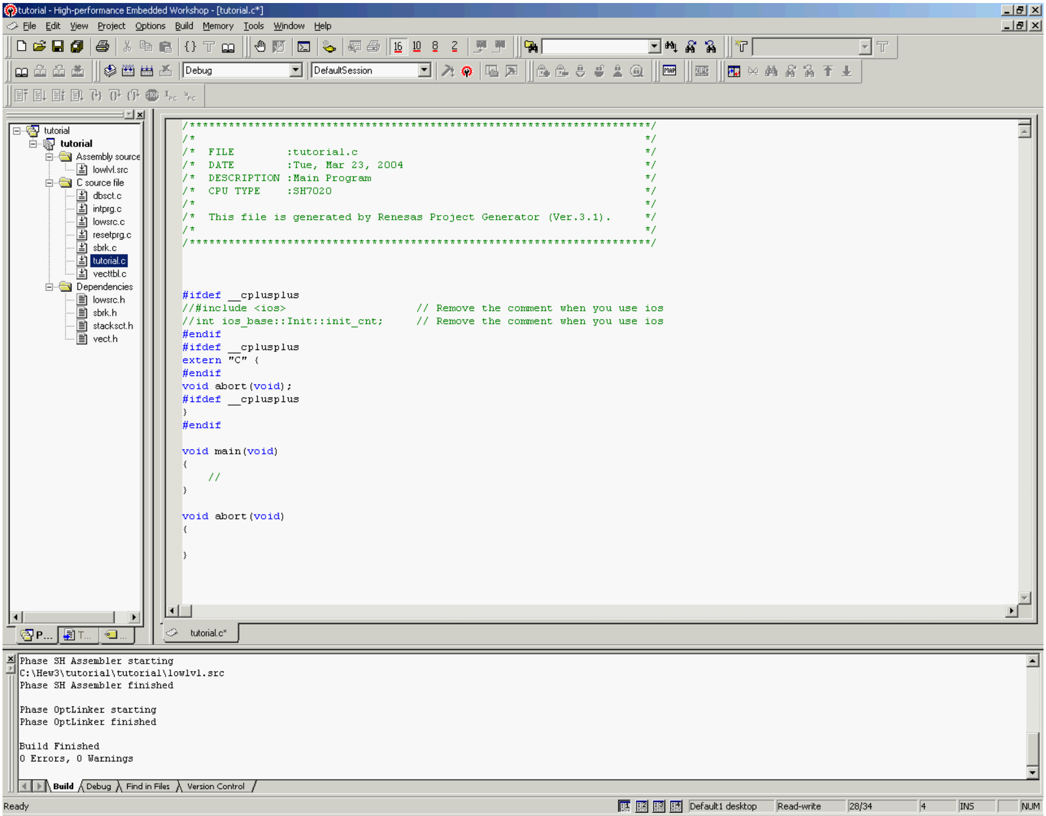


```
/***********************************************************************/
/*                                                                     */
/*  FILE        :tutorial.c                                            */
/*  DATE        :Tue, Mar 23, 2004                                     */
/*  DESCRIPTION :Main Program                                          */
/*  CPU TYPE    :SH7020                                                */
/*                                                                     */
/*  This file is generated by Renesas Project Generator (Ver.3.1).     */
/*                                                                     */
/***********************************************************************/



#ifdef __cplusplus
//#include <ios>                        // Remove the comment when you use ios
//int ios_base::Init::init_cnt;       // Remove the comment when you use ios
#endif
#ifdef __cplusplus
extern "C" {
#endif
void abort(void);
#ifdef __cplusplus
}
#endif

void main(void)
{

}

void abort(void)
{

}
```

```
Phase SH Assembler starting
C:\Hew3\tutorial\tutorial\lowlvl.src
Phase SH Assembler finished

Phase OptLinker starting
Phase OptLinker finished

Build Finished
0 Errors, 0 Warnings
```

Figure 3.1 Opening a Project File

Then you can edit the file opened. If you edit the file, an asterisk (*) will be added to the file path shown on the title bar of the HEW (figure 3.2). If you save the file, the asterisk will disappear.



```
/************************************************************************/
/*                                                                      */
/*  FILE        :tutorial.c                                             */
/*  DATE        :Tue, Mar 23, 2004                                      */
/*  DESCRIPTION :Main Program                                           */
/*  CPU TYPE    :SH7020                                                 */
/*                                                                      */
/*  This file is generated by Renesas Project Generator (Ver.3.1).      */
/*                                                                      */
/************************************************************************/



#ifdef __cplusplus
//#include <ios>                        // Remove the comment when you use ios
//int ios_base::Init::init_cnt;       // Remove the comment when you use ios
#endif
#ifdef __cplusplus
extern "C" {
#endif
void abort(void);
#ifdef __cplusplus
}
#endif

void main(void)
{
    //
}

void abort(void)
{

}
```

```
Phase SH Assembler starting
C:\Hew3\tutorial\tutorial\lowlvl.src
Phase SH Assembler finished

Phase OptLinker starting
Phase OptLinker finished

Build Finished
0 Errors, 0 Warnings
```

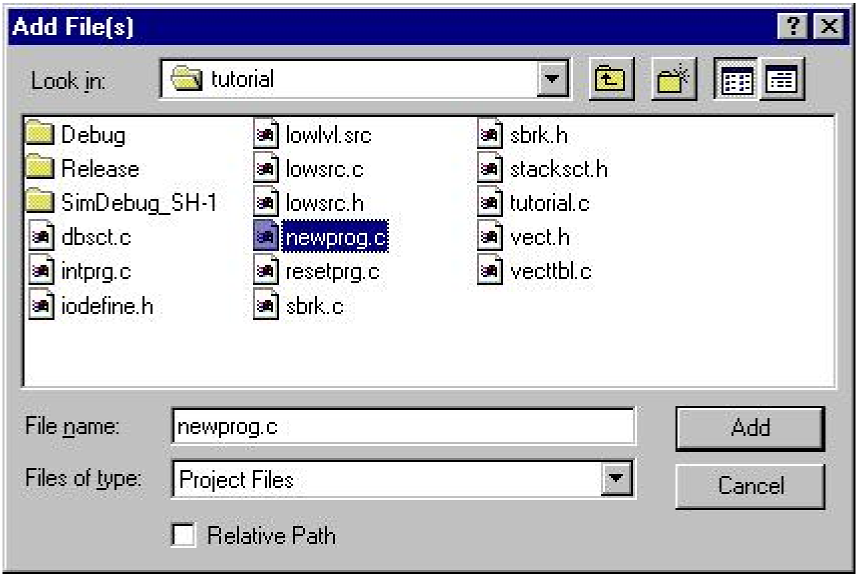
Figure 3.2 Editing a Project File

If you want to open an existing text file with the editor of the HEW, select [File->Open]. If you want to edit a new text file with the editor, select [File->New]. In this tutorial, select [File->New], input the three lines shown below, select [File->Save As…] and save the file as "newprog.c".

```
void NewProgram(void)
{
}
```

3.2 Adding/Removing a File To/From a Project

This section describes how to add a file to a project and how to delete a file from a project. In this tutorial, add the file, “newprog.c” created in the previous section to the project as follows. Select [Project->Add Files…], then the [Add File(s)] dialog box (figure 3.3) will be launched. Select a file, “newprog.c” in this example, to be added and click the [Add] button. Thus “newprog.c” will be added to the project.



Look in: tutorial

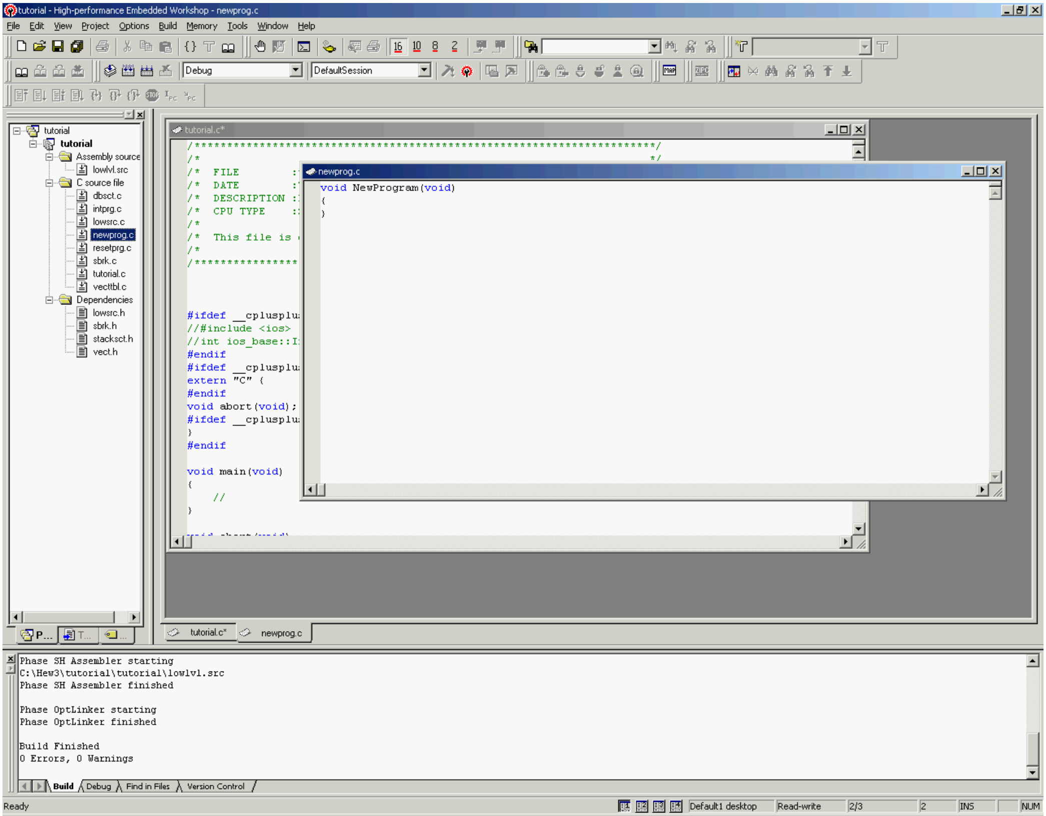
File name: newprog.c

Files of type: Project Files

□ Relative Path

Figure 3.3 Adding a File To a Project

The added file will be displayed on the [Projects] sheet of the [Workspace] window as shown in figure 3.4.



```
void NewProgram(void)
{
}
```

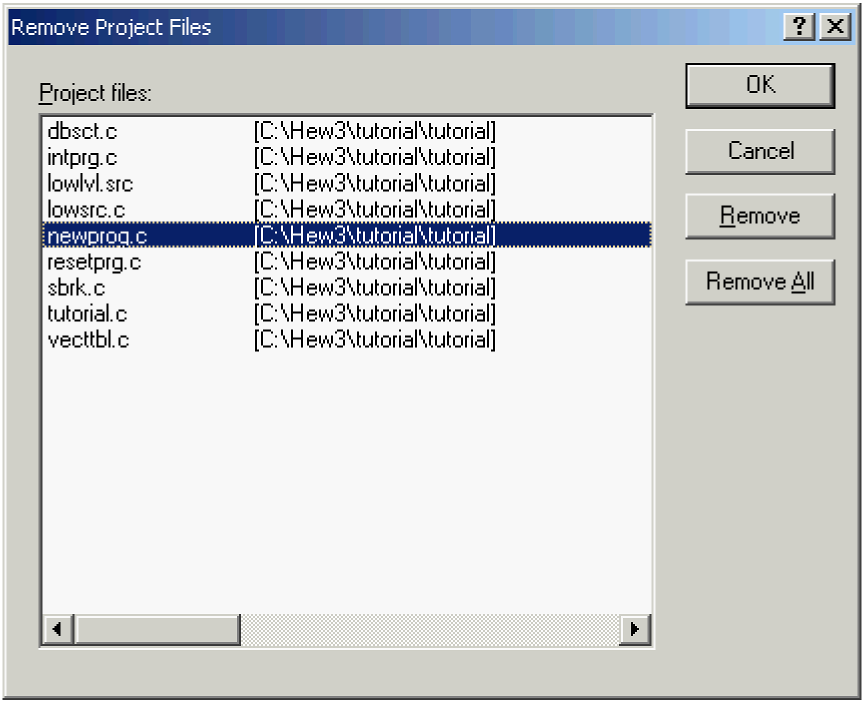
```
Phase SH Assembler starting
C:\Hew3\tutorial\tutorial\lowlvl.src
Phase SH Assembler finished

Phase OptLinker starting
Phase OptLinker finished

Build Finished
0 Errors, 0 Warnings
```

Figure 3.4 Project With a File Added

Next, how to delete a file from a project will be shown. Select [Project->Remove File…], then the [Remove Project Files] dialog box (figure 3.5) will be displayed. Select one or more project files on this dialog box and click the [Remove] button. Then the selected file(s) will disappear from the list of the dialog box. Clicking [OK] actually removes the file(s) from the project. Clicking [Cancel] instead will nullify the removal.

You have another way to delete a file from a project. Select the file in the [Projects] sheet of the [Workspace] window and press the [Delete] key. Then the file will be deleted from the project.

If you want to exclude a file from a build temporarily instead of removing the file from the project, refer to section 4.4, "Excluding a Project File from Build".



Project files:

dbsct.c	[C:\Hew3\tutorial\tutorial]
intprg.c	[C:\Hew3\tutorial\tutorial]
lowlvl.src	[C:\Hew3\tutorial\tutorial]
lowsrc.c	[C:\Hew3\tutorial\tutorial]
newprog.c	[C:\Hew3\tutorial\tutorial]
resetprg.c	[C:\Hew3\tutorial\tutorial]
sbrk.c	[C:\Hew3\tutorial\tutorial]
tutorial.c	[C:\Hew3\tutorial\tutorial]
vecttbl.c	[C:\Hew3\tutorial\tutorial]

Figure 3.5 Remove Project Files Dialog Box

3.3 File Groups Used in a Project

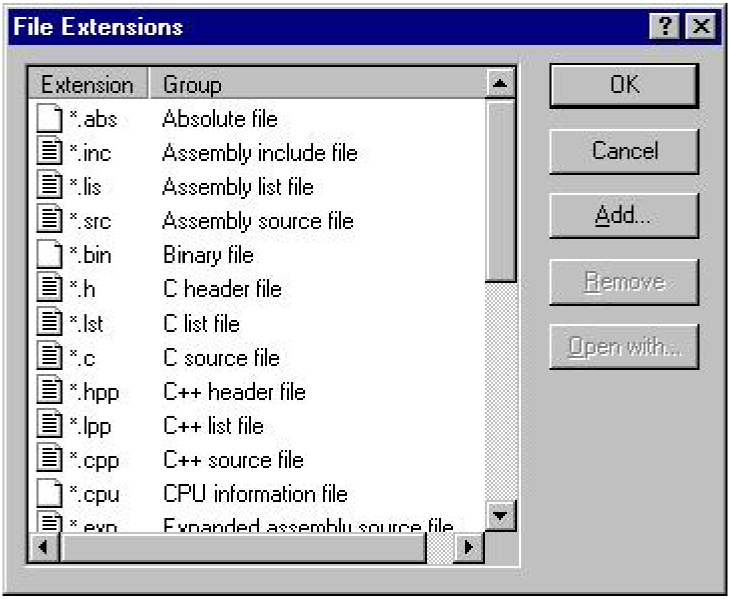
In this section, a *file group* to which a file added to a project belongs is described. For example, the project generator adds a file described in C language or assembly language to the project. A file with a file extension “.c” belongs to the [C source file] group, and a file with a file extension “.src” belongs to the [Assembly source file] group.

The HEW identifies a group to which a file belongs by checking the file extension. Select [Project->File Extensions…], then the [File Extensions] dialog box (figure 3.6) will be displayed. File extensions used in the project are shown on this dialog box. Every file extension belongs to a file group. A file group can be associated to a tool in a build process.

If you want to use your own file extension in the project, define a new file extension by clicking the [Add…] button.

The icon displayed in the [Extension] column of the [File Extensions] dialog box can show a tool which opens the file. If the icon of a file extension is the same as the icon shown below, a file with that file extension can be opened by the editor of the HEW.

🗎

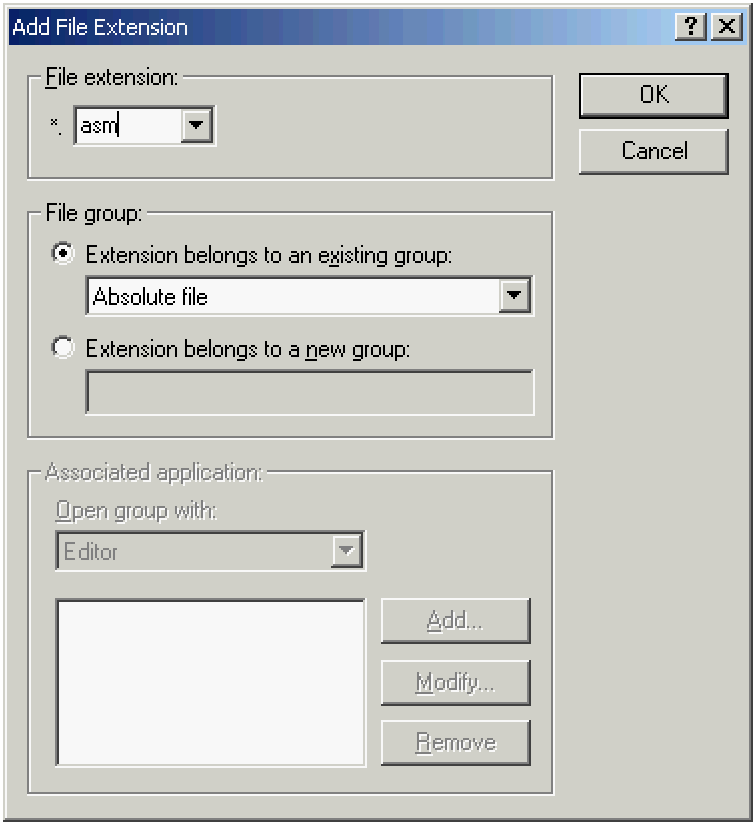


Extension	Group
*.abs	Absolute file
*.inc	Assembly include file
*.lis	Assembly list file
*.src	Assembly source file
*.bin	Binary file
*.h	C header file
*.lst	C list file
*.c	C source file
*.hpp	C++ header file
*.lpp	C++ list file
*.cpp	C++ source file
*.cpu	CPU information file
*.exp	Expanded assembly source file

Figure 3.6 File Extensions Dialog Box (Initial Screen)

Click the [Add…] button on the [File Extensions] dialog box, then the [Add File Extension] dialog box (figure 3.7) will be launched. Enter a new file extension in the [File extension] field and specify the [File group] group box. If you want to add an extension to an existing file group, select the [Extension belongs to an existing group] radio button and select a file group from the drop-down list. If you want to create a new file group, select the [Extension belongs to a new group] radio button and enter the name of the file group in the edit field.

In this tutorial, enter "asm" in the [File extension] field, select "Assembly source file" from the drop-down list and click [OK]. Then a file with the extension, "*.asm", will be treated as a file in the [Assembly source file] group (figure 3.8).



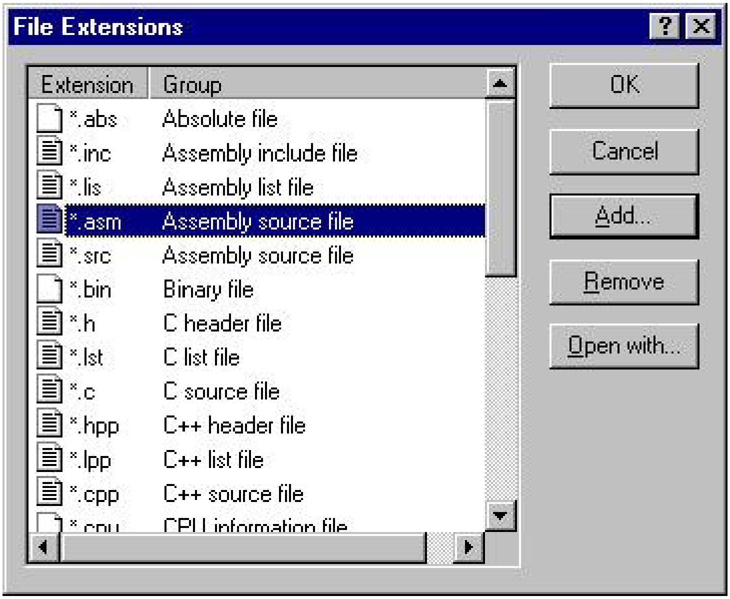
Add File Extension

File extension:
*. asm

File group:
● Extension belongs to an existing group:
Absolute file
○ Extension belongs to a new group:

Associated application:
Open group with:
Editor

Figure 3.7 Add File Extension Dialog Box

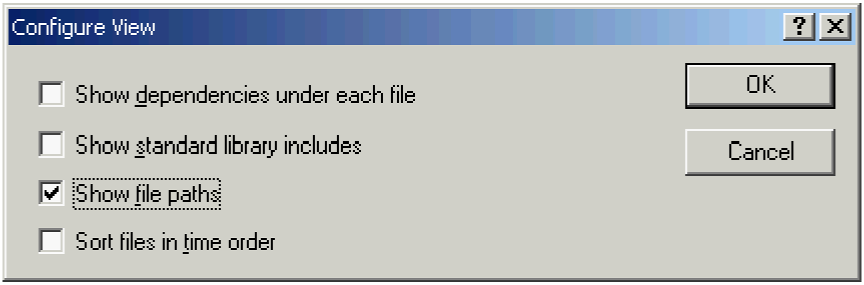


Extension	Group
*.abs	Absolute file
*.inc	Assembly include file
*.lis	Assembly list file
*.asm	Assembly source file
*.src	Assembly source file
*.bin	Binary file
*.h	C header file
*.lst	C list file
*.c	C source file
*.hpp	C++ header file
*.lpp	C++ list file
*.cpp	C++ source file
*.cpu	CPU information file

Figure 3.8 File Extension Dialog Box (After Adding an File Extension)

3.4 Customizing the Workspace Window

Click the right mouse button on the [Projects] sheet of the [Workspace] window, then a pop-up menu will be displayed. Select [Configure View…] on this menu, then [Configure View] dialog box (figure 3.9) will be launched. On this dialog box, you can specify whether the file dependencies are shown for each file or not, whether a file is shown in a file group folder (figure 3.10) or not, and so on.



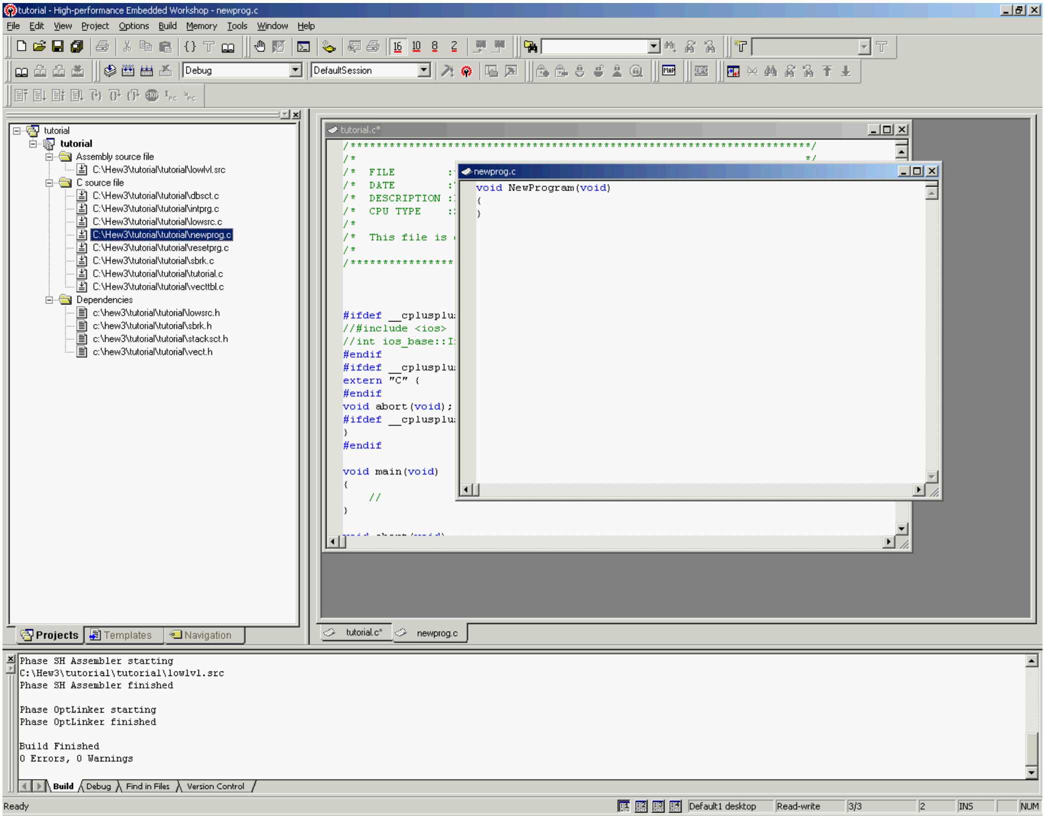
☐ Show dependencies under each file

☐ Show standard library includes

☑ Show file paths

☐ Sort files in time order

Figure 3.9 Configure View Dialog Box



```
tutorial
  tutorial
    Assembly source file
      C:\Hew3\tutorial\tutorial\lowlvl.src
    C source file
      C:\Hew3\tutorial\tutorial\dbsct.c
      C:\Hew3\tutorial\tutorial\intprg.c
      C:\Hew3\tutorial\tutorial\lowsrc.c
      C:\Hew3\tutorial\tutorial\newprog.c
      C:\Hew3\tutorial\tutorial\resetprg.c
      C:\Hew3\tutorial\tutorial\sbrk.c
      C:\Hew3\tutorial\tutorial\tutorial.c
      C:\Hew3\tutorial\tutorial\vecttbl.c
    Dependencies
      c:\hew3\tutorial\tutorial\lowsrc.h
      c:\hew3\tutorial\tutorial\sbrk.h
      c:\hew3\tutorial\tutorial\stacksct.h
      c:\hew3\tutorial\tutorial\vect.h
```

tutorial.c*

```
/***********************************************************************/
/*                                                                   */
/*  FILE        :
/*  DATE        :
/*  DESCRIPTION :
/*  CPU TYPE    :
/*
/*  This file is
/*
/****************



#ifdef __cplusplu
//#include <ios>
//int ios_base::I
#endif
#ifdef __cplusplu
extern "C" {
#endif
void abort(void);
#ifdef __cplusplu
}
#endif

void main(void)
{
    //
}
```

newprog.c

```
void NewProgram(void)
{
}
```

```
Phase SH Assembler starting
C:\Hew3\tutorial\tutorial\lowlvl.src
Phase SH Assembler finished

Phase OptLinker starting
Phase OptLinker finished

Build Finished
0 Errors, 0 Warnings
```

Figure 3.10   Configuration of the Workspace Window

Section 4 Building a Project

4.1 Build Overview

The HEW provides two ways of building a project.

1. "Build All" ([Build->Build All])

The series of tools including the compiler, assembler, and linkage editor are applied to all of the source program files.

2. "Build" [Build->Build]

The series of tools are applied to a source program file updated after the last build. (The update means not only the update of a source program file, but also an update of a file included by the source program file and an update of an option applied to the source program file.)

If you want to compile or assemble a single source program file, select the file on the [Projects] sheet of the [Workspace] window and select [Build->Build File].

If you want to execute "Build" or "Build All" for multiple programs or configurations, specify the target by selecting the [Build -> Build Multiple] menu item.

CAUTION ON BUILD

1. Depending of the setting of the editor window, a file being edited might be saved on executing a build. So close a file which you do NOT want to be saved beforehand. For details of customizing the editor, refer to the High-Performance Embedded Workshop 3 User's Manual.
2. Do not save a file used in a project during the build because doing so might disturb the build process.

4.2 Setting Options

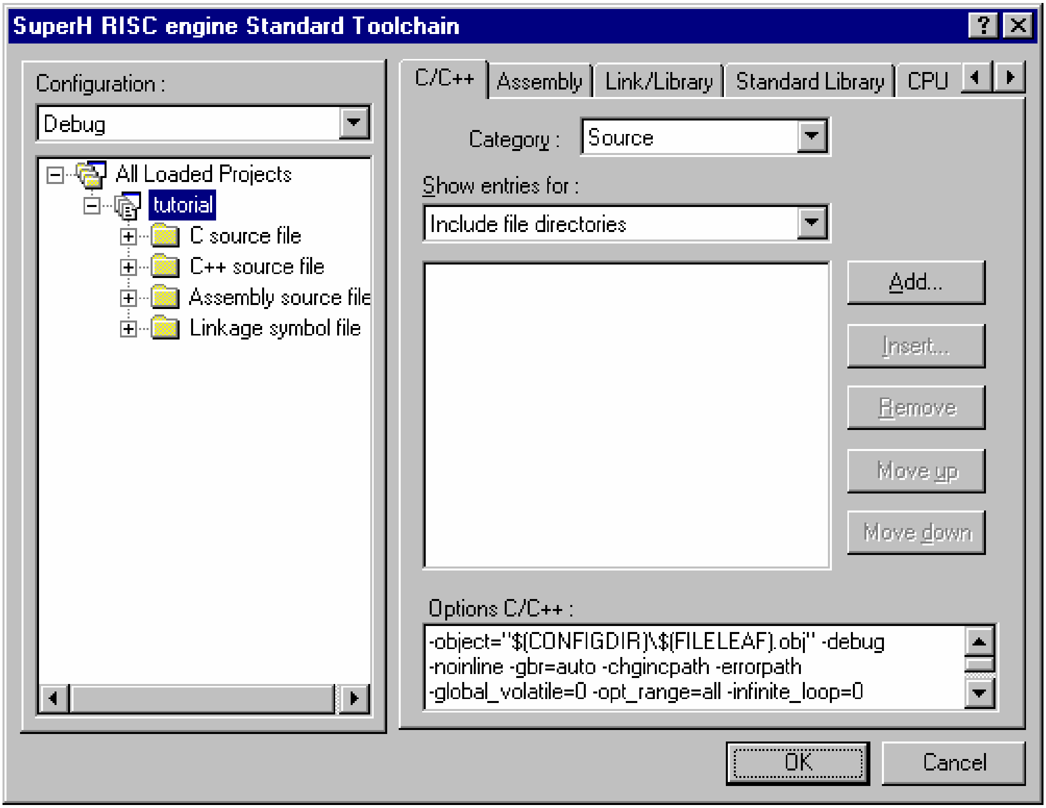
In this section, overview of the option setting is described.

When the project generator generates a project, the minimum options required for a build are already specified. If you want to modify the options, however, select [Options-> SuperH RISC engine Standard Toolchain] and set options on the option dialog box.

Figure 4.1 shows the option dialog box of a C/C++ compiler. On the left hand side of the dialog box is a file list, and on the right hand side are optional setting controls for each tool divided into some pages. If you select a file in the file list, you can specify options only to the selected file. If you select a file group folder in the file list, you can specify the same option to all the files in the file group. To specify the same options to specific files, select the files while pressing the [Ctrl] key or the [Shift] key.

If you select [Default Options] in a file group folder, you can specify initial options for the file group. The HEW automatically adds the options specified to the [Default Options] to a file of the corresponding file group of the corresponding file extension of the file. For example, if you add a file with an extension “C” (“*.C”) to the project, the settings of the [Default Options] of the [C source file] group will be added to the file automatically.

For the details of the options refer to the SuperH RISC engine C/C++ Compiler, Cross Assembler, or Optimizing Linkage Editor User’s Manual.



Configuration : Debug

Category : Source

Show entries for : Include file directories

Options C/C++ :

```
-object="$(CONFIGDIR)\$(FILELEAF).obj" -debug
-noinline -gbr=auto -chgincpath -errorpath
-global_volatile=0 -opt_range=all -infinite_loop=0
```

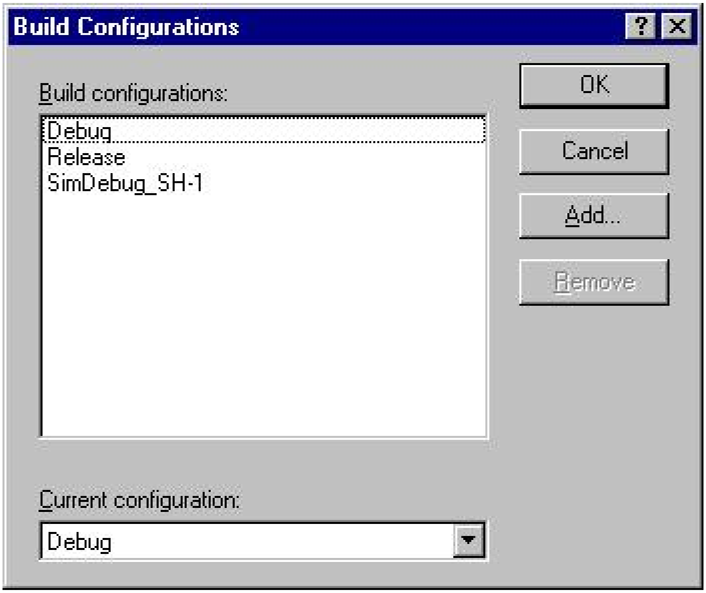
Figure 4.1   Option Dialog Box of C/C++ Compiler

4.3 Customizing the Configuration

In the previous section, how to set options to a project file is described. In this section, how to create more than one suite of option settings for a build will be described. A suite of option settings for a build is called a *configuration*.

The project generator creates two configurations, [Debug] and [Release]. Difference between them is only the setting of the debugging option. When a target for debugging is selected, the configuration for the target is also created. To switch the current configuration to another, launch the [Build Configurations] dialog box (figure 4.2) by selecting [Options->Build Configurations…], select a configuration from the [Current configuration] drop-down list, and click [OK]. You can also change the current configuration by selecting one from the drop-down list on the toolbar.

On the [Build Configurations] dialog box, you can create a new configuration or delete a configuration. For details, refer to chapter 2, “Build Basics”, of the High-Performance Embedded Workshop 3 User’s Manual.



Build configurations:
Debug
Release
SimDebug_SH-1

Current configuration: Debug

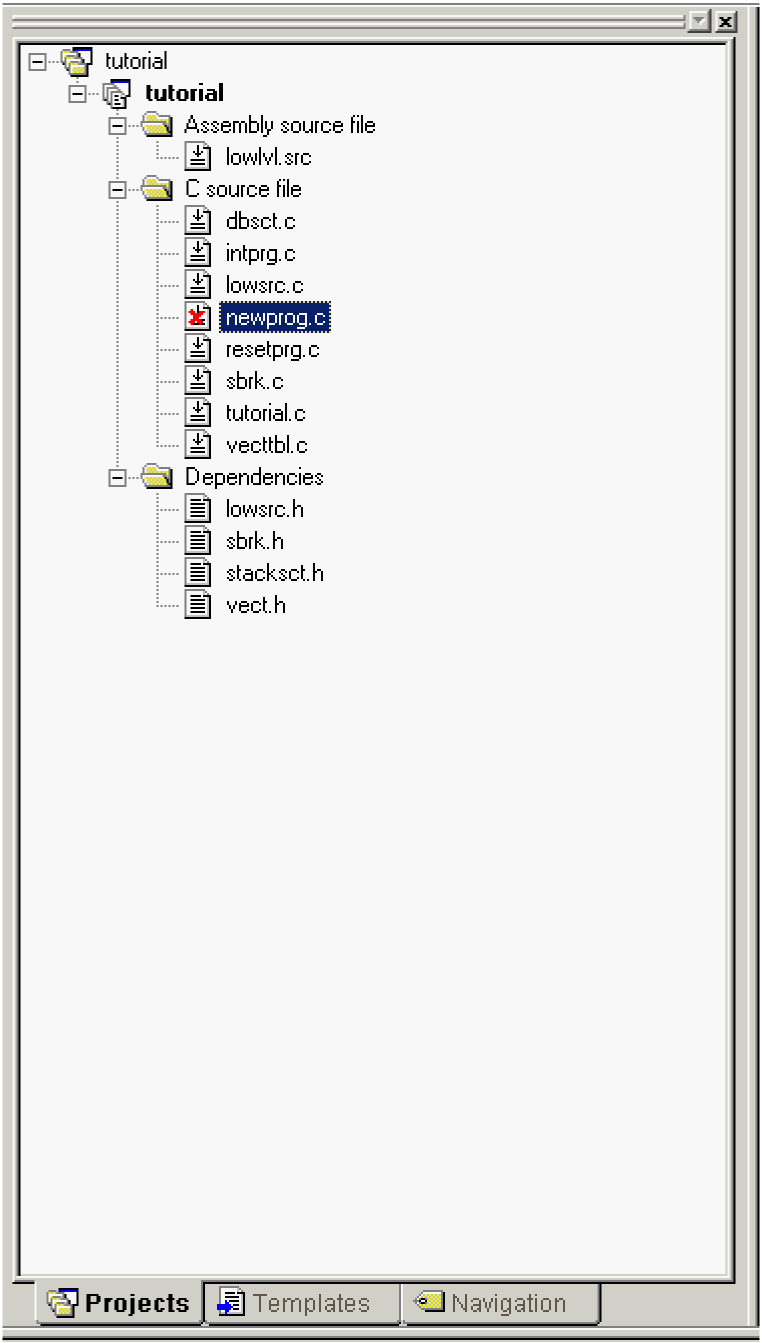
Figure 4.2 Build Configurations Dialog Box

4.4 Excluding a Project File from Build

In this section, how to exclude a project file from build will be described. Excluding a project file from build does not delete the file from a project, but exclude a project file from build in configuration by configuration basis.

Select a file on the [Projects] sheet of the [Workspace] window, click the right mouse button, and select [Exclude Build <file>] on the pop-up menu. Then a red cross will be added to the icon of this file as shown in figure 4.3. This file will be excluded from a build.

To include an excluded file into build, select the file on the [Projects] sheet of the [Workspace] window, click the right mouse button, and select [Include Build <file>] on the pop-up menu.



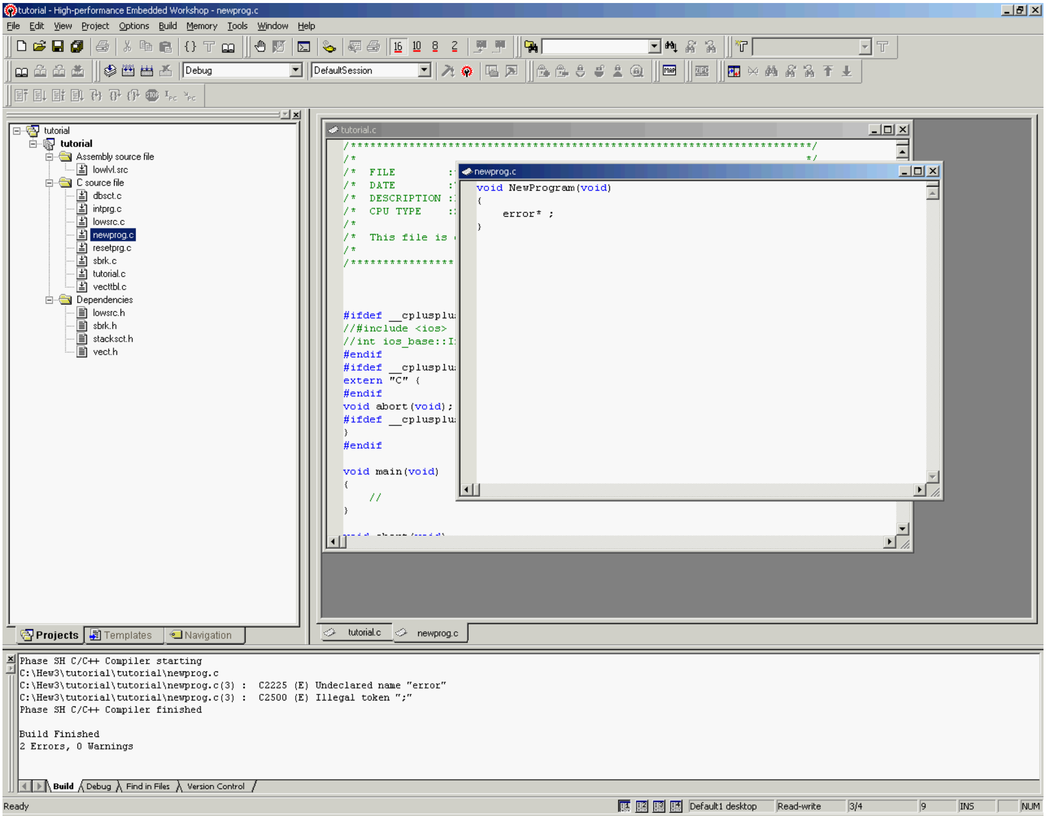
- tutorial
  - **tutorial**
    - Assembly source file
      - lowlvl.src
    - C source file
      - dbsct.c
      - intprg.c
      - lowsrc.c
      - newprog.c
      - resetprg.c
      - sbrk.c
      - tutorial.c
      - vecttbl.c
    - Dependencies
      - lowsrc.h
      - sbrk.h
      - stacksct.h
      - vect.h

Figure 4.3 Excluding a Project File from Build

4.5 Correcting Errors in a Build

In this section, how to jump to a line of a file which caused an error in compiling or assembling will be described.

Figure 4.4 shows an example in which a compiler detected a syntax error in a source program file. When you build a project, messages, including error messages, from a tool in a build will be displayed on the [Output] window. If you double click the line of the error message on the [Build] sheet of the [Output] window, the file which caused the error will be opened and the cursor will be placed on the line which caused the error. (The cursor might not be placed on the line which caused the error if you have already edited the file.) Placing the cursor on the error line of the output window and pressing the [F1] key displays help of that error.



```
void NewProgram(void)
{
    error* ;
}
```

```
Phase SH C/C++ Compiler starting
C:\Hew3\tutorial\tutorial\newprog.c
C:\Hew3\tutorial\tutorial\newprog.c(3) :  C2225 (E) Undeclared name "error"
C:\Hew3\tutorial\tutorial\newprog.c(3) :  C2500 (E) Illegal token ";"
Phase SH C/C++ Compiler finished

Build Finished
2 Errors, 0 Warnings
```

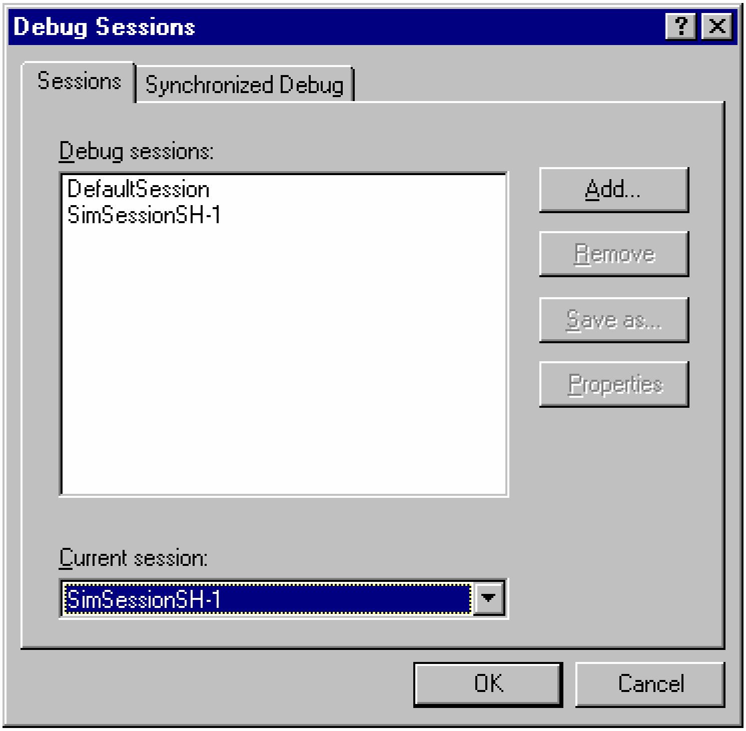
Figure 4.4 Correcting the Line which Caused an Error

4.6 Customizing the Session

You can save the options for debugger to the session file.

The project generator creates the session, [Default Session]. When a target for debugging is selected, the session for the target is also created. To use the a target, launch the [Debug Sessions] dialog box (figure 4.5) by selecting [Options->Debug Settings…], select a session [SimSessionSH-1] from the [Current session] drop-down list, and click [OK]. You can also change the current session by selecting one from the drop-down list on the toolbar.

On the [Debug Sessions] dialog box, you can create a new session or delete a session. For details, refer to chapter 1.9, “Debugger Sessions”, of the High-Performance Embedded Workshop 3 User’s Manual Simulator/Debugger Part.



Debug sessions:
DefaultSession
SimSessionSH-1

Current session: SimSessionSH-1

Figure 4.5 Debug Sessions Dialog Box

Section 5 Debugging

5.1 Preparation

The basic functions of the simulator/debugger will be described in this section using a sample program.

NOTE
Note that the contents of usage examples (figures) in this section will differ depending on the compiler version.

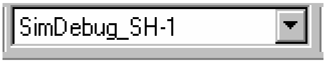
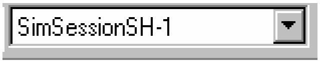
5.1.1 Sample Program

The HEW demonstration program is used for the sample program and is written in C language It first sorts ten random data in the ascending order, and then in the descending order. The sample program:

(1) Generates random data for sorting using the main function.
(2) Inputs the array which stores the random data that is generated by the main function, then sorts the data in the ascending order using the sort function.
(3) Inputs the array generated by the sort function, and sorts the data in the descending order using the change function.
(4) Displays the random data and the sorted data using the printf function.

5.1.2 Creating the Sample Program

Create the HEW demonstration program by referring to sections 1 to 4, and note the following.

- Specify [Demonstration] for the [Project Type] in section 2.1, Creating a New Workspace.
- Specify [SH-1] for the [CPU Series:].
- Specify [SH-1 Simulator] for the [Target:].
- Specify [SimDebug_SH-1] for the configuration  SimDebug_SH-1 on the toolbar before building the project.
- Specify [SimSession_SH-1] for the session  SimSessionSH-1 on the toolbar.

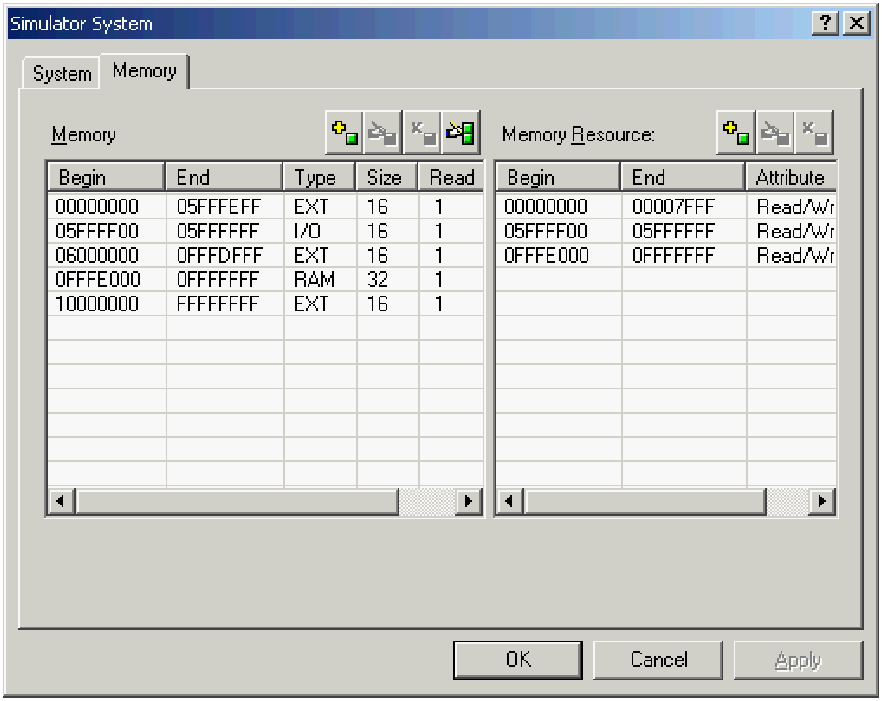
Since section 5 explains the debugging function, [Demonstration] has not been optimized. Do not change this setting.

5.2 Settings for Debugging

5.2.1 Allocating the Memory Resource

The allocation of the memory resource is necessary to run the application being developed. When using the demonstration project, the memory resource is allocated automatically, so check the setting.

- Select [Simulator->Memory Resource...] from the [Option] menu, and display the allocation of the current memory resource.



Begin	End	Type	Size	Read
00000000	05FFFEFF	EXT	16	1
05FFFF00	05FFFFFF	I/O	16	1
06000000	0FFFDFFF	EXT	16	1
0FFFE000	0FFFFFFF	RAM	32	1
10000000	FFFFFFFF	EXT	16	1

Begin	End	Attribute
00000000	00007FFF	Read/Wr
05FFFF00	05FFFFFF	Read/Wr
0FFFE000	0FFFFFFF	Read/Wr

Figure 5.1 Simulator System Dialog Box (Memory page)

The program area is allocated to the addresses H'00000000 to H'00007FFF. The stack area is allocated to the addresses H'0FFFE000 to H'0FFFFFFF, which can be read from or written to.

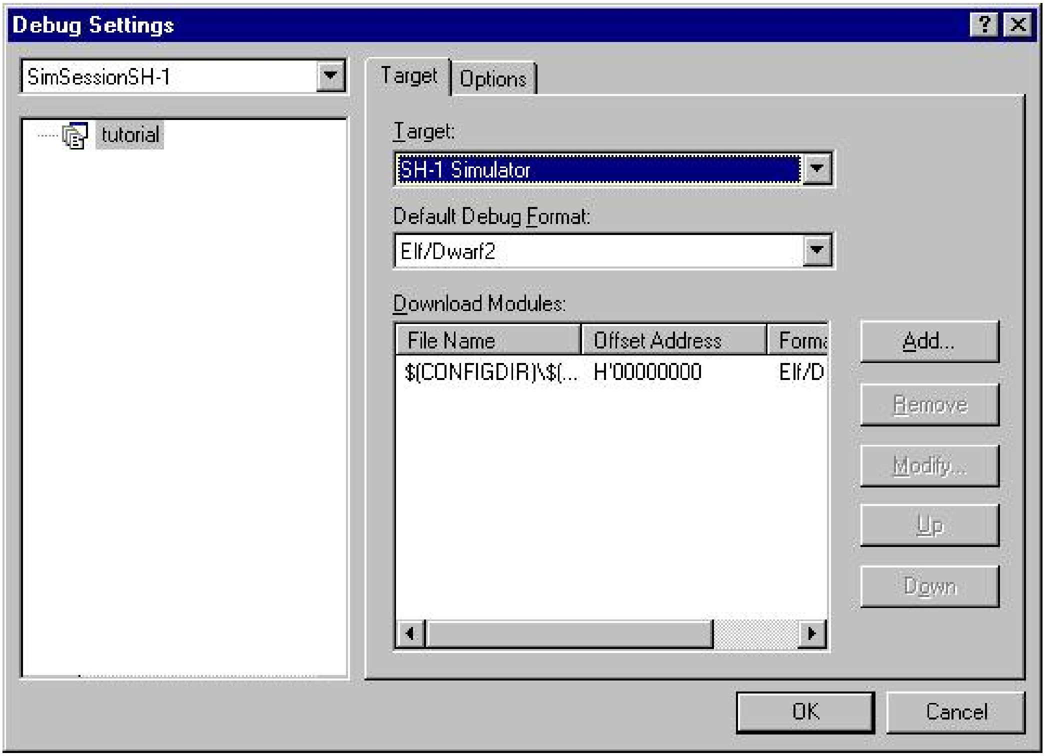
- Close the dialog box by clicking [OK].

The memory resource can also be referred to or modified by using the [Simulator] page on the [SuperH RISC engine Standard Toolchain] dialog box. Changes made in either of the dialog boxes are reflected.

5.2.2 Downloading the Sample Program

When using the demonstration project, the sample program to be downloaded is automatically set, so check the settings.

- Open the [Debug Setting] dialog box by selecting [Debug Settings...] on the [Option] menu.



Target: SH-1 Simulator

Default Debug Format: Elf/Dwarf2

Download Modules:

File Name	Offset Address	Form
$(CONFIGDIR)\$(...	H'00000000	Elf/D

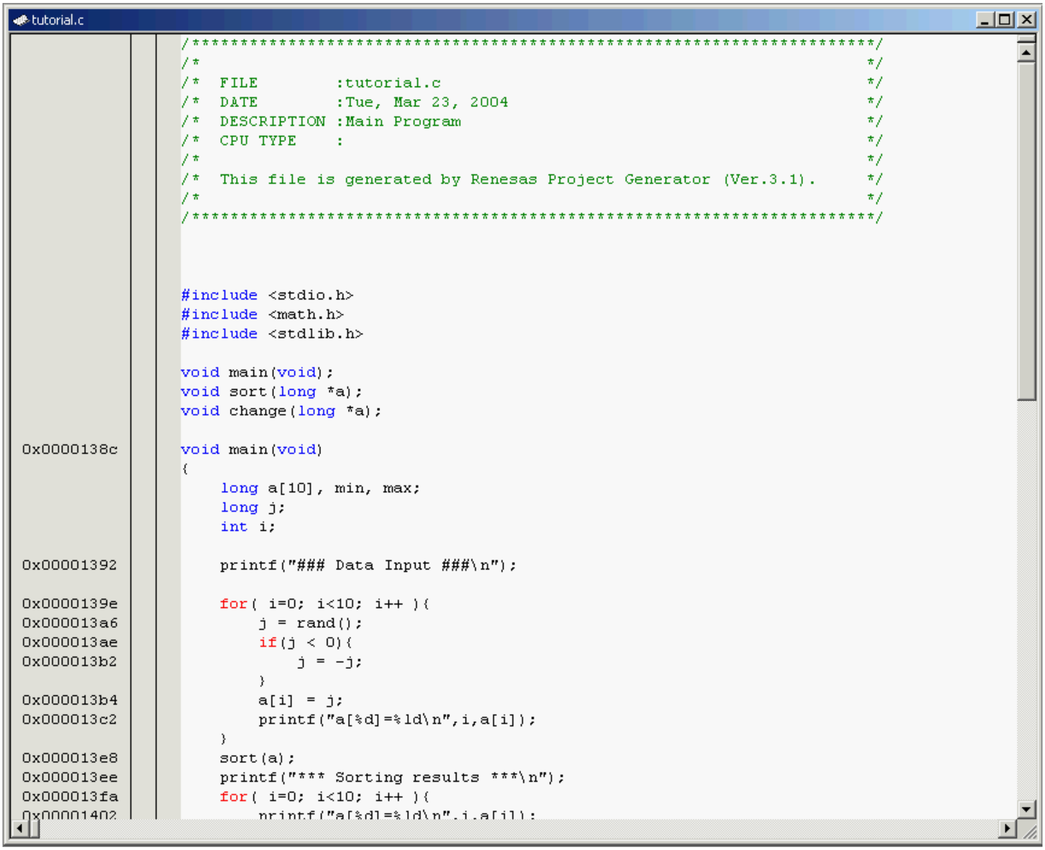
Figure 5.2 Debug Settings Dialog Box

- Files to be downloaded is set in [Download Modules].
- Close the [Debug Settings] dialog box by clicking the [OK] button.
- Download the sample program by selecting [Download Modules->All Download Modules] from the [Debug] menu.

5.2.3 Displaying the Source Program

The source-level debugging is supported by the HEW. Refer to section 3.1, Editing and Creating a Source Program File, and display the source file ("tutorial.c") in the [Source] window.

- Open the [Source] window by double-clicking tutorial.c on the [Workspace] window.



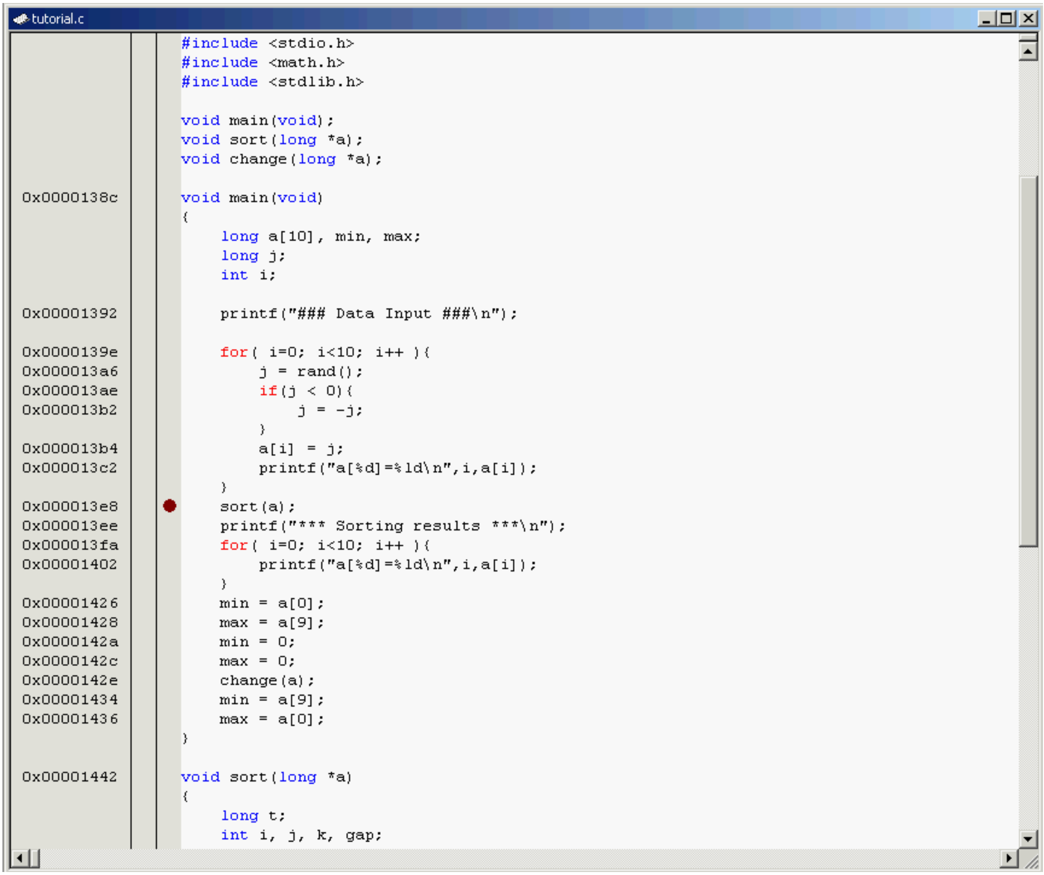
```
            /************************************************************************/
            /*                                                                      */
            /*  FILE        :tutorial.c                                             */
            /*  DATE        :Tue, Mar 23, 2004                                      */
            /*  DESCRIPTION :Main Program                                           */
            /*  CPU TYPE    :                                                       */
            /*                                                                      */
            /*  This file is generated by Renesas Project Generator (Ver.3.1).     */
            /*                                                                      */
            /************************************************************************/



            #include <stdio.h>
            #include <math.h>
            #include <stdlib.h>

            void main(void);
            void sort(long *a);
            void change(long *a);

0x0000138c  void main(void)
            {
                long a[10], min, max;
                long j;
                int i;

0x00001392      printf("### Data Input ###\n");

0x0000139e      for( i=0; i<10; i++ ){
0x000013a6          j = rand();
0x000013ae          if(j < 0){
0x000013b2              j = -j;
                    }
0x000013b4          a[i] = j;
0x000013c2          printf("a[%d]=%ld\n",i,a[i]);
                }
0x000013e8      sort(a);
0x000013ee      printf("*** Sorting results ***\n");
0x000013fa      for( i=0; i<10; i++ ){
0x00001402          printf("a[%d]=%ld\n",i,a[i]);
```

Figure 5.3 Source Window (Displaying the Source Program)

5.2.4 Setting a PC Breakpoint

Breakpoints can be set easily by the [Source] window. To set a breakpoint on a line that includes the sort function call:

- Place the cursor in the line that includes the sort function call and click the right mouse button to launch the pop-up menu, and select [Toggle Breakpoint] from the pop-up menu.



```
                  #include <stdio.h>
                  #include <math.h>
                  #include <stdlib.h>

                  void main(void);
                  void sort(long *a);
                  void change(long *a);

0x0000138c        void main(void)
                  {
                      long a[10], min, max;
                      long j;
                      int i;

0x00001392            printf("### Data Input ###\n");

0x0000139e            for( i=0; i<10; i++ ){
0x000013a6                j = rand();
0x000013ae                if(j < 0){
0x000013b2                    j = -j;
                          }
0x000013b4                a[i] = j;
0x000013c2                printf("a[%d]=%ld\n",i,a[i]);
                      }
0x000013e8      ●     sort(a);
0x000013ee            printf("*** Sorting results ***\n");
0x000013fa            for( i=0; i<10; i++ ){
0x00001402                printf("a[%d]=%ld\n",i,a[i]);
                      }
0x00001426            min = a[0];
0x00001428            max = a[9];
0x0000142a            min = 0;
0x0000142c            max = 0;
0x0000142e            change(a);
0x00001434            min = a[9];
0x00001436            max = a[0];
                  }

0x00001442        void sort(long *a)
                  {
                      long t;
                      int i, j, k, gap;
```

Figure 5.4 Source Window (Setting the Breakpoint)

A [•] is displayed at the line that includes the sort function call, indicating that the PC breakpoint is set at the address.

5.2.5 Setting the Profiler

- Open the [Profile] window by selecting [Profile] from the [View->Performance] menu.

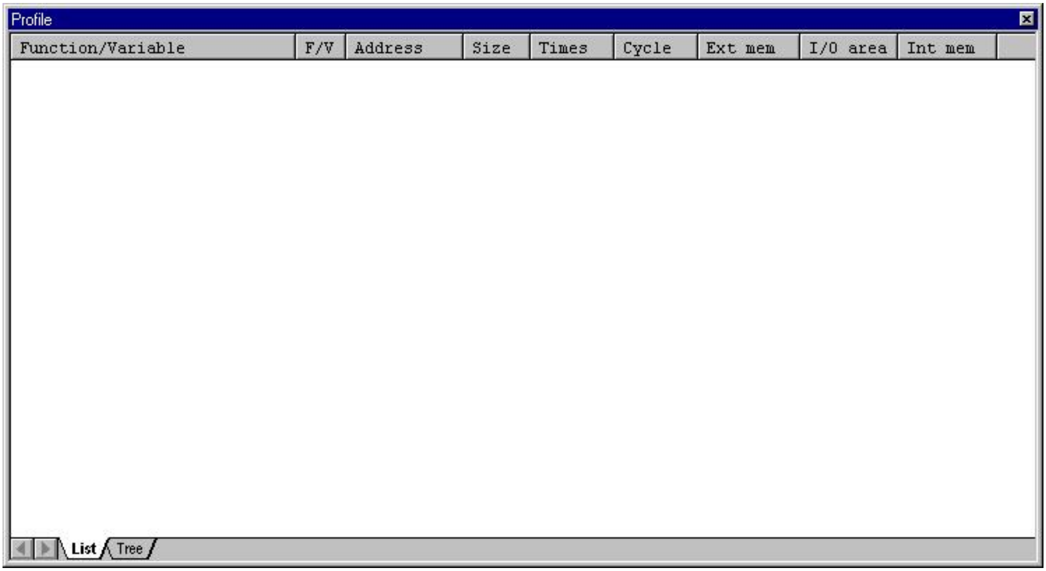


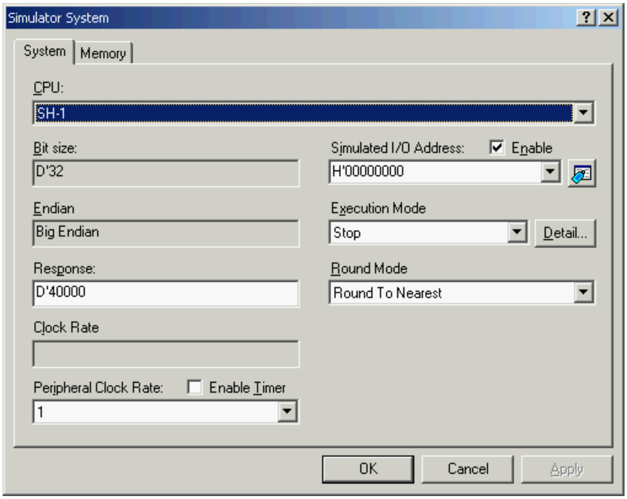
Figure 5.5 Profile Window

- Open the pop-up menu by right clicking the mouse on the [Profile] window, and select [Enable Profiler] to enable acquisition of the profile information.

5.2.6 Setting the Simulated I/O

When the demonstration project is used, the simulated I/O is automatically set, so check the setting.

- Open the [Simulator System] dialog box by selecting [Simulator->System] from the [Option] menu.



CPU: SH-1

Bit size: D'32

Simulated I/O Address: ☑ Enable

H'00000000

Endian: Big Endian

Execution Mode: Stop

Response: D'40000

Round Mode: Round To Nearest

Clock Rate

Peripheral Clock Rate: ☐ Enable Timer

1

Figure 5.6 Simulator System Dialog Box (System page)

- Confirm that [Enable] in [System Call Address] is checked.
- Click the [OK] button to enable the Simulated I/O
- Select [Simulated I/O] from the [View->CPU] menu and open the [Simulated I/O] window. The Simulated I/O will not be enabled if the [Simulated I/O] window is not open.

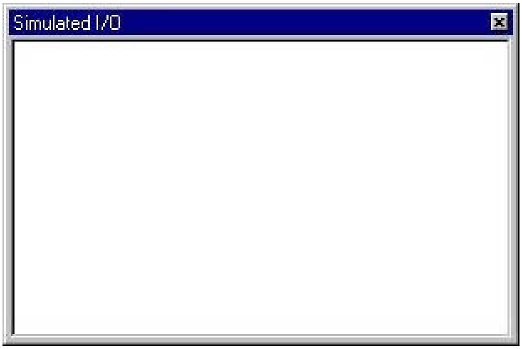
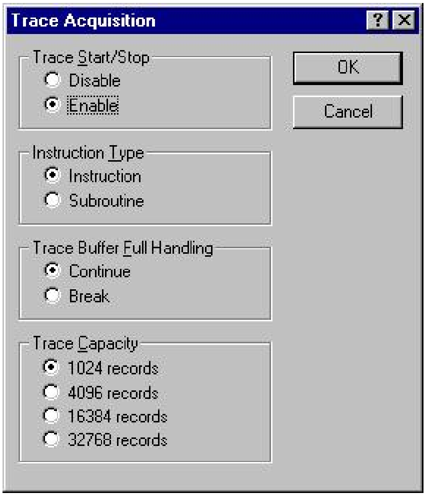


Figure 5.7 Simulated I/O Window

5.2.7 Setting the Trace Information Acquisition Conditions

- Select [Trace] from the [View->Code] menu and open the [Trace] window. Open the pop-up menu by right clicking the mouse on the [Trace] window, and select [Acquisition...] from the pop-up menu.

The [Trace Acquisition] dialog box below will be displayed.



Trace Acquisition

Trace Start/Stop
○ Disable
● Enable

Instruction Type
● Instruction
○ Subroutine

Trace Buffer Full Handling
● Continue
○ Break

Trace Capacity
● 1024 records
○ 4096 records
○ 16384 records
○ 32768 records

Figure 5.8 Trace Acquisition Dialog Box

- Set [Trace start/Stop] to [Enable] in the [Trace Acquisition] dialog box, and click the [OK] button to enable the acquisition of the trace information.

5.2.8 Setting the Stack Pointer and Program Counter

To execute the program, the program counter must be set from the location of the reset vector. In the reset vector of the sample program, the PC value H'800 is written, and the SP value H'0FFFFFF0 is written.

- Select [Reset CPU] from the [Debug] menu, or click the [Reset CPU] button on the toolbar.

Set the program counter to H'800, and the stack pointer to H'0FFFFFF0 from the reset vector.

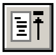


Figure 5.9 Reset CPU Button

5.3 Start Debugging

5.3.1 Executing a Program

- Select [Go] from the [Debug] menu, or click the [Go] button on the toolbar.

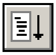
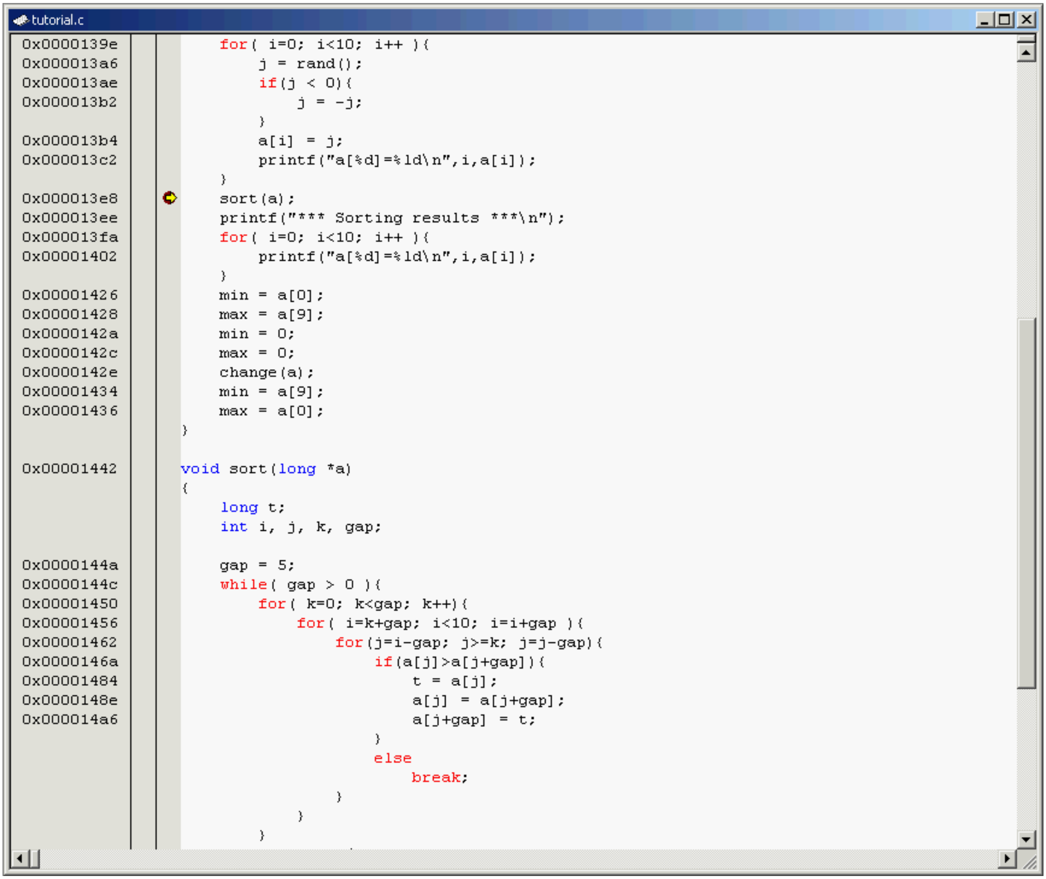


Figure 5.10 Go Button

The program halts where a breakpoint is set. An arrow is displayed in the [Source] window, indicating the location the execution has stopped. As the termination cause, [PC Breakpoint] is displayed in the [Output] window.

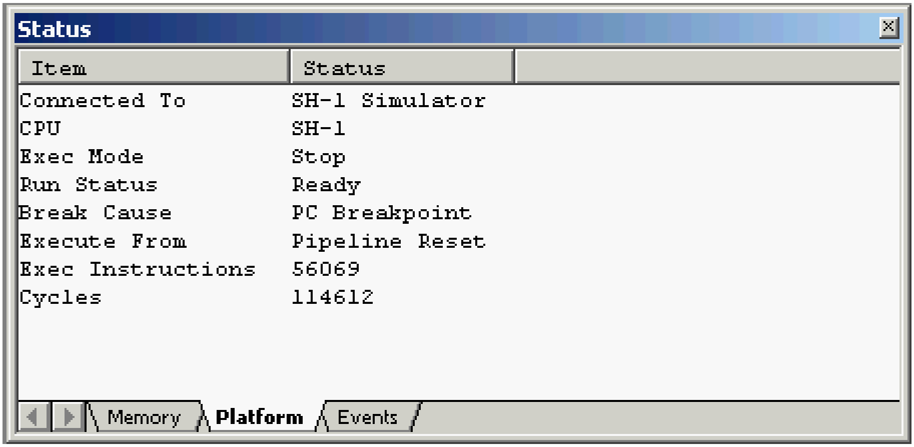


```
0x0000139e        for( i=0; i<10; i++ ){
0x000013a6            j = rand();
0x000013ae            if(j < 0){
0x000013b2                j = -j;
                      }
0x000013b4            a[i] = j;
0x000013c2            printf("a[%d]=%ld\n",i,a[i]);
                  }
0x000013e8   ➡    sort(a);
0x000013ee        printf("*** Sorting results ***\n");
0x000013fa        for( i=0; i<10; i++ ){
0x00001402            printf("a[%d]=%ld\n",i,a[i]);
                  }
0x00001426        min = a[0];
0x00001428        max = a[9];
0x0000142a        min = 0;
0x0000142c        max = 0;
0x0000142e        change(a);
0x00001434        min = a[9];
0x00001436        max = a[0];
              }

0x00001442    void sort(long *a)
              {
                  long t;
                  int i, j, k, gap;

0x0000144a        gap = 5;
0x0000144c        while( gap > 0 ){
0x00001450            for( k=0; k<gap; k++){
0x00001456                for( i=k+gap; i<10; i=i+gap ){
0x00001462                    for(j=i-gap; j>=k; j=j-gap){
0x0000146a                        if(a[j]>a[j+gap]){
0x00001484                            t = a[j];
0x0000148e                            a[j] = a[j+gap];
0x000014a6                            a[j+gap] = t;
                                  }
                                  else
                                      break;
                              }
                          }
                      }
```

Figure 5.11 Source Window (Break Status)

The termination cause can be displayed in the [Status] window.

- Select [Status] from the [View->CPU] menu to open the [Status] window, and select the [Platform] sheet in the [Status] window.



Item	Status
Connected To	SH-1 Simulator
CPU	SH-1
Exec Mode	Stop
Run Status	Ready
Break Cause	PC Breakpoint
Execute From	Pipeline Reset
Exec Instructions	56069
Cycles	114612

Figure 5.12 Status Window

The above status window indicates that:

(1) The cause of break is a PC breakpoint

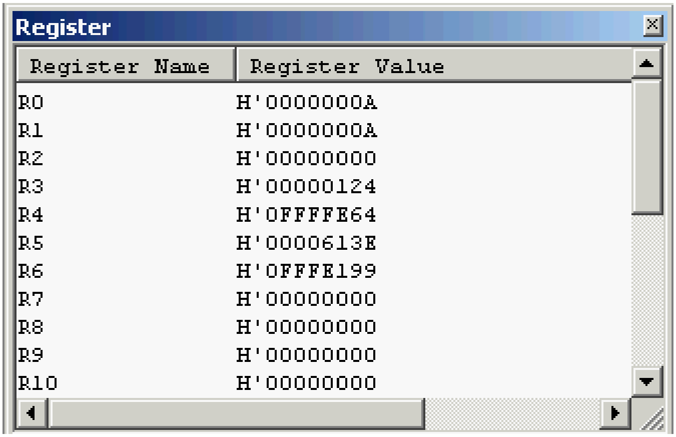
(2) Execution is performed from the pipeline reset

(3) The number of executed instructions by the GO command is 56,069

(4) The executed number of cycles from the pipeline reset is 114,612

Register values can be checked in the [Register] window.

- Select [Registers] from the [View->CPU] menu.



Register Name	Register Value
R0	H'0000000A
R1	H'0000000A
R2	H'00000000
R3	H'00000124
R4	H'0FFFFE64
R5	H'0000613E
R6	H'0FFFE199
R7	H'00000000
R8	H'00000000
R9	H'00000000
R10	H'00000000

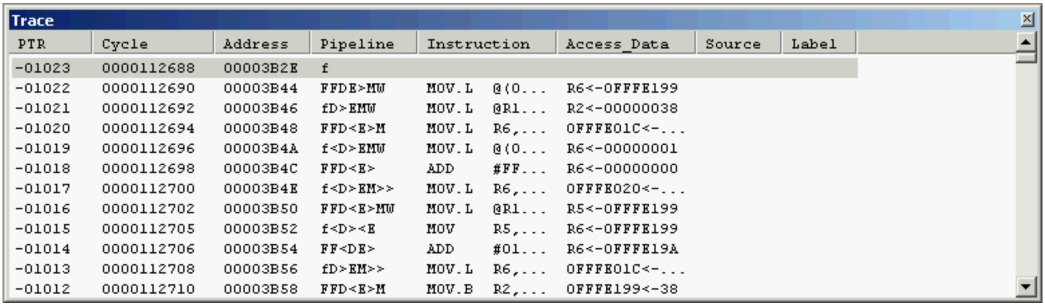
Figure 5.13 Register Window

Register values when the program is terminated can be checked.

5.3.2 Using the Trace Buffer

The trace buffer can be used to clarify the history of instruction execution.

- Select [Trace] from the [View->Code] menu and open the [Trace] window. Scroll up to the very top of the window.

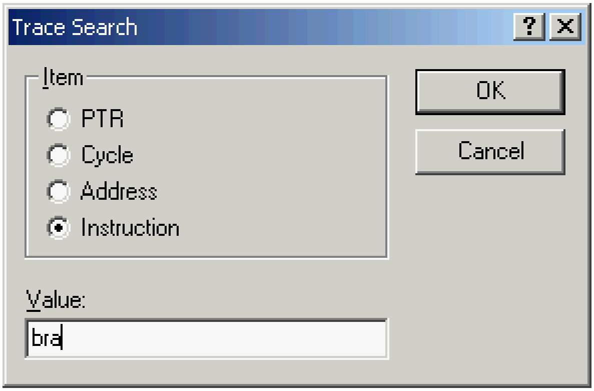


PTR	Cycle	Address	Pipeline	Instruction	Access_Data	Source	Label
-01023	0000112688	00003B2E	f				
-01022	0000112690	00003B44	FFDE>MW	MOV.L @(0...	R6<-0FFFE199		
-01021	0000112692	00003B46	fD>EMW	MOV.L @R1...	R2<-00000038		
-01020	0000112694	00003B48	FFD<E>M	MOV.L R6,...	0FFFE01C<-...		
-01019	0000112696	00003B4A	f<D>EMW	MOV.L @(0...	R6<-00000001		
-01018	0000112698	00003B4C	FFD<E>	ADD #FF...	R6<-00000000		
-01017	0000112700	00003B4E	f<D>EM>>	MOV.L R6,...	0FFFE020<-...		
-01016	0000112702	00003B50	FFD<E>MW	MOV.L @R1...	R5<-0FFFE199		
-01015	0000112705	00003B52	f<D><E	MOV R5,...	R6<-0FFFE199		
-01014	0000112706	00003B54	FF<DE>	ADD #01...	R6<-0FFFE19A		
-01013	0000112708	00003B56	fD>EM>>	MOV.L R6,...	0FFFE01C<-...		
-01012	0000112710	00003B58	FFD<E>M	MOV.B R2,...	0FFFE199<-38		

Figure 5.14 Trace Window (Trace Information Display)

5.3.3 Performing Trace Search

Click the right mouse button on the [Trace] window to launch the pop-up menu, and select [Find...] to open the [Trace Search] dialog box.



Trace Search

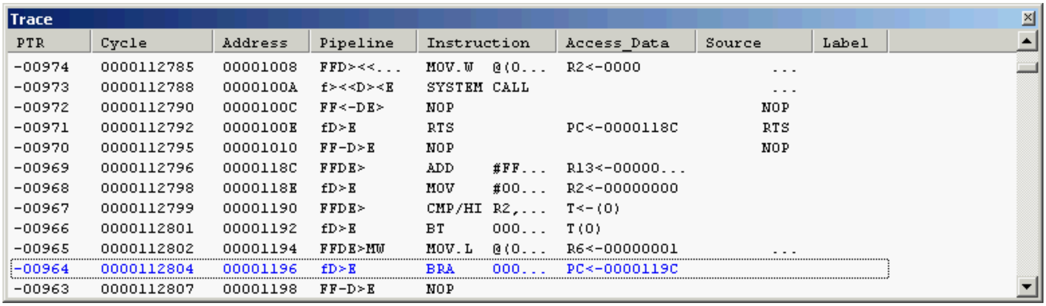
Item

○ PTR
○ Cycle
○ Address
● Instruction

Value: bra

Figure 5.15 Trace Search Dialog Box

Setting the item to be searched to [Item] and the contents to be searched to [Value] and clicking the [OK] button begins the trace search. When the searched item is found, the first line is highlighted. To continue searching the same contents [Value], click the right mouse button in the [Trace] window to display the pop-up menu, and select [Find Next] from the pop-up menu. The next searched line is highlighted.



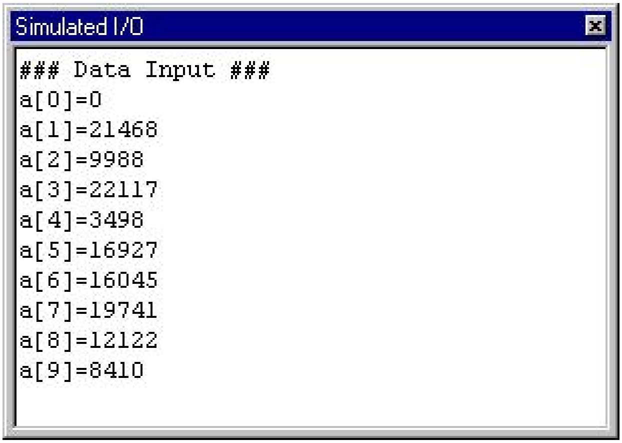
Trace

```
PTR     Cycle       Address   Pipeline  Instruction      Access_Data      Source      Label
-00974  0000112785  00001008  FFD><<... MOV.W  @(0...    R2<-0000               ...
-00973  0000112788  0000100A  f><<D><E  SYSTEM CALL                             ...
-00972  0000112790  0000100C  FF<-DE>   NOP                                    NOP
-00971  0000112792  0000100E  fD>E      RTS              PC<-0000118C          RTS
-00970  0000112795  00001010  FF-D>E    NOP                                    NOP
-00969  0000112796  0000118C  FFDE>     ADD    #FF...    R13<-00000...
-00968  0000112798  0000118E  fD>E      MOV    #00...    R2<-00000000
-00967  0000112799  00001190  FFDE>     CMP/HI R2,...    T<-(0)
-00966  0000112801  00001192  fD>E      BT     000...    T(0)
-00965  0000112802  00001194  FFDE>MW   MOV.L  @(0...    R6<-00000001           ...
-00964  0000112804  00001196  fD>E      BRA    000...    PC<-0000119C
-00963  0000112807  00001198  FF-D>E    NOP
```

Figure 5.16 Trace Window (Searched Result)

5.3.4 Checking Simulated I/O

Random data that is displayed by the printf function can be checked in the [Simulated I/O] window.



Simulated I/O

```
### Data Input ###
a[0]=0
a[1]=21468
a[2]=9988
a[3]=22117
a[4]=3498
a[5]=16927
a[6]=16045
a[7]=19741
a[8]=12122
a[9]=8410
```

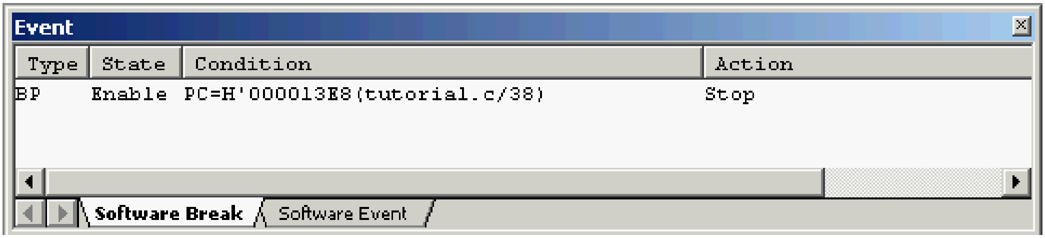
Figure 5.17 Simulated I/O Window

- Do not close the [Simulated I/O] window.

5.3.5 Checking the Breakpoints

A list of all the breakpoints that are set in the program can be checked in the [Event] window.

- Select [Eventpoints] from the [View -> Code] menu.



Event

Type	State	Condition	Action
BP	Enable	PC=H'000013E8(tutorial.c/38)	Stop

Figure 5.18 Event Window

A breakpoint can be set, a new breakpoint can be defined, and a breakpoint can be deleted using the [Event] window.

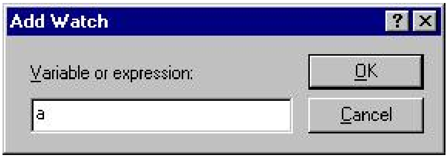
- Close the [Event] window.

5.3.6 Watching Variables

It is possible to watch the values of variables used in your program and to verify that they change in the way that you expected. For example, set a watch on the long-type array “a” declared at the beginning of the program, by using the following procedure:

- Select [Watch] from the [View -> Symbol] menu and open the [Watch] window. And click the right mouse button on the [Watch] window and choose [Add Watch...] from the pop-up menu.

The following dialog box will be displayed.

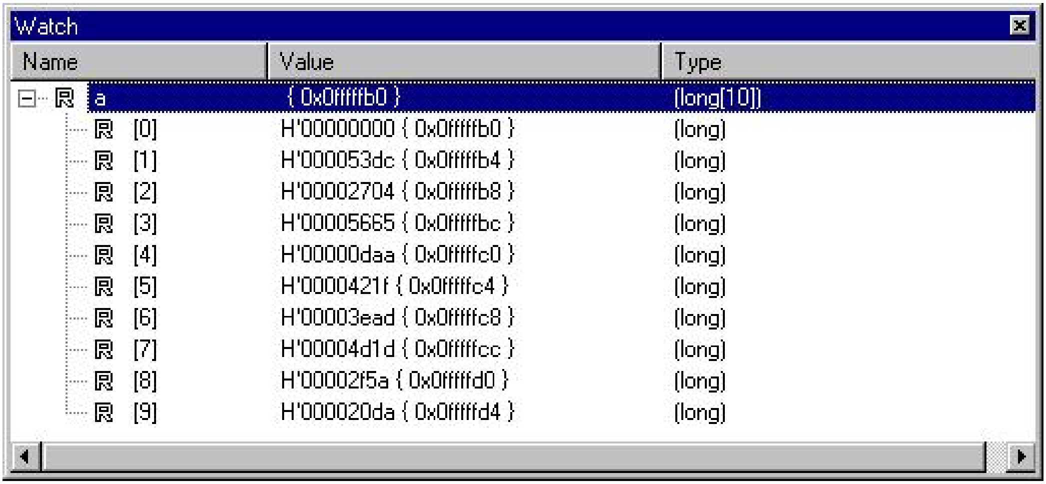


Variable or expression: a

Figure 5.19   Add Watch Dialog Box

- Type array a and click the **[**OK**]** button.

The [Watch] window will show the long-type array a.
You can double-click the + symbol to the left of array “a” in the [Watch] window to expand the variable and show the individual elements in the array.



Name	Value	Type
a	{ 0x0ffffb0 }	(long[10])
[0]	H'00000000 { 0x0ffffb0 }	(long)
[1]	H'000053dc { 0x0ffffb4 }	(long)
[2]	H'00002704 { 0x0ffffb8 }	(long)
[3]	H'00005665 { 0x0ffffbc }	(long)
[4]	H'00000daa { 0x0ffffc0 }	(long)
[5]	H'0000421f { 0x0ffffc4 }	(long)
[6]	H'00003ead { 0x0ffffc8 }	(long)
[7]	H'00004d1d { 0x0ffffcc }	(long)
[8]	H'00002f5a { 0x0ffffd0 }	(long)
[9]	H'000020da { 0x0ffffd4 }	(long)

Figure 5.20 Watch Window

- Close the [Watch] window.

5.3.7 Executing the Program in Single Steps

The simulator/debugger has various stepping menus that are useful in debugging the program.

Menu	Description
Step In	Executes each statement (includes statements within the function)
Step Over	Executes a function call in a single step
Step Out	Steps out of a function, and stops at the next statement of the program that called the function
Step...	Executes the specified number of steps at the specified speed

[Step In]: Enters the called function and stops at the statement at the start of the called function.

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

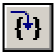
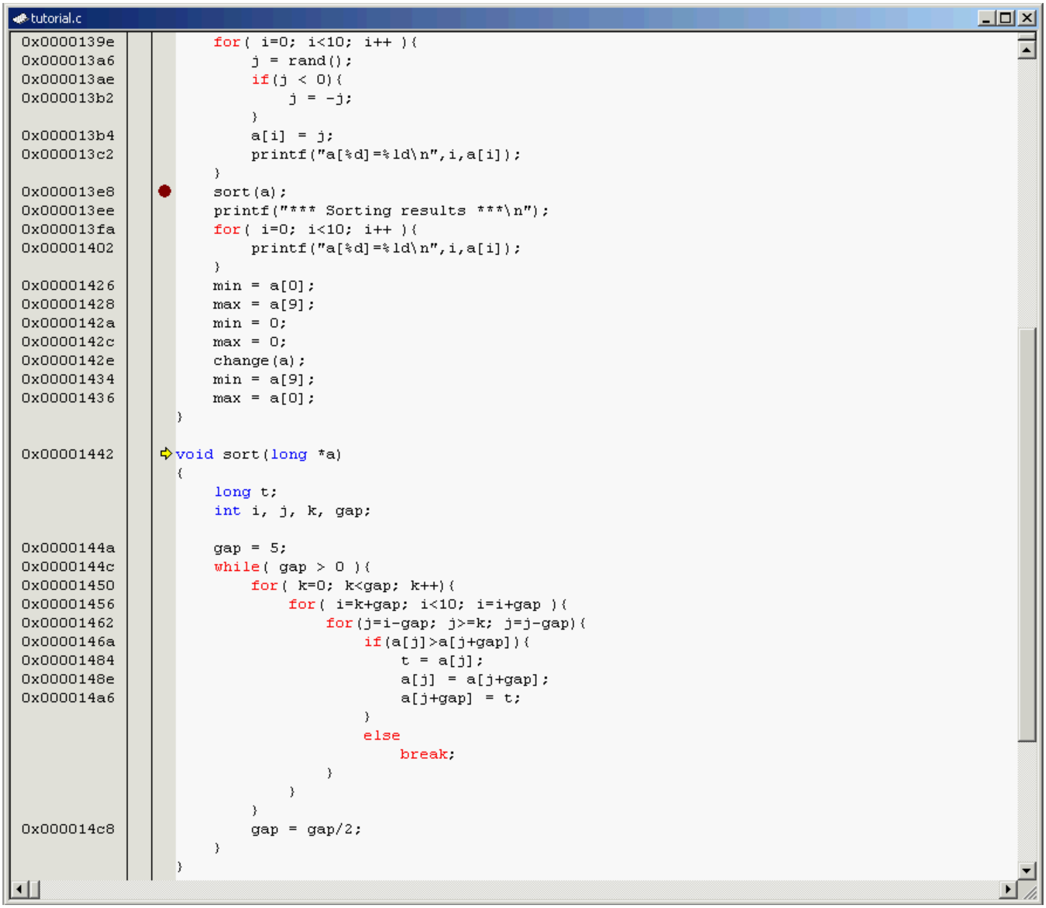


Figure 5.21 Step In Button



```
0x0000139e         for( i=0; i<10; i++ ){
0x000013a6             j = rand();
0x000013ae             if(j < 0){
0x000013b2                 j = -j;
                       }
0x000013b4             a[i] = j;
0x000013c2             printf("a[%d]=%ld\n",i,a[i]);
                   }
0x000013e8  ●      sort(a);
0x000013ee         printf("*** Sorting results ***\n");
0x000013fa         for( i=0; i<10; i++ ){
0x00001402             printf("a[%d]=%ld\n",i,a[i]);
                   }
0x00001426         min = a[0];
0x00001428         max = a[9];
0x0000142a         min = 0;
0x0000142c         max = 0;
0x0000142e         change(a);
0x00001434         min = a[9];
0x00001436         max = a[0];
               }

0x00001442  ⇨  void sort(long *a)
               {
                   long t;
                   int i, j, k, gap;

0x0000144a         gap = 5;
0x0000144c         while( gap > 0 ){
0x00001450             for( k=0; k<gap; k++){
0x00001456                 for( i=k+gap; i<10; i=i+gap ){
0x00001462                     for(j=i-gap; j>=k; j=j-gap){
0x0000146a                         if(a[j]>a[j+gap]){
0x00001484                             t = a[j];
0x0000148e                             a[j] = a[j+gap];
0x000014a6                             a[j+gap] = t;
                                   }
                                   else
                                       break;
                               }
                           }
                       }
0x000014c8             gap = gap/2;
                   }
               }
```

Figure 5.22 Source Window (Step In)

- The PC location display (=>) in the [Source] window moves to the statement at the start of the sort function.

[Step Out]: Steps out of the called function and stops at the next statement in the called program.

- Select [Step Out] from the [Debug] menu to exit the sort function, or click the [Step Out] button on the toolbar.

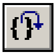
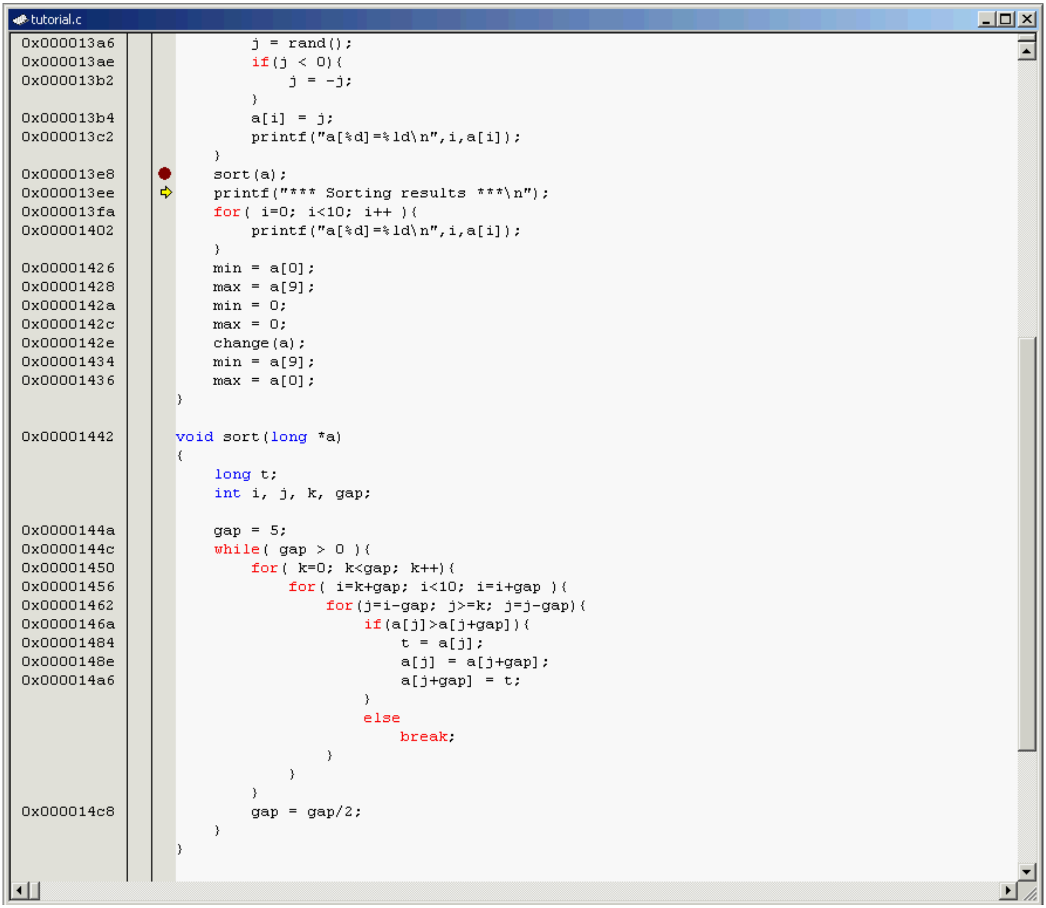


Figure 5.23 Step Out Button



```
0x000013a6            j = rand();
0x000013ae            if(j < 0){
0x000013b2                j = -j;
                      }
0x000013b4            a[i] = j;
0x000013c2            printf("a[%d]=%ld\n",i,a[i]);
                  }
0x000013e8    ●   sort(a);
0x000013ee    ⇨   printf("*** Sorting results ***\n");
0x000013fa        for( i=0; i<10; i++ ){
0x00001402            printf("a[%d]=%ld\n",i,a[i]);
                  }
0x00001426        min = a[0];
0x00001428        max = a[9];
0x0000142a        min = 0;
0x0000142c        max = 0;
0x0000142e        change(a);
0x00001434        min = a[9];
0x00001436        max = a[0];
              }

0x00001442    void sort(long *a)
              {
                  long t;
                  int i, j, k, gap;

0x0000144a        gap = 5;
0x0000144c        while( gap > 0 ){
0x00001450            for( k=0; k<gap; k++){
0x00001456                for( i=k+gap; i<10; i=i+gap ){
0x00001462                    for(j=i-gap; j>=k; j=j-gap){
0x0000146a                        if(a[j]>a[j+gap]){
0x00001484                            t = a[j];
0x0000148e                            a[j] = a[j+gap];
0x000014a6                            a[j+gap] = t;
                                  }
                                  else
                                      break;
                              }
                          }
                      }
0x000014c8            gap = gap/2;
                  }
              }
```

Figure 5.24 Source Window (Step Out)

[Step Over]: Executes a function call in a single step, and stops at the next statement in the main program.

Select [Step Over] from the [Debug] menu or click the [Step Over] button on the toolbar to step over the statements in the printf function.

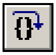
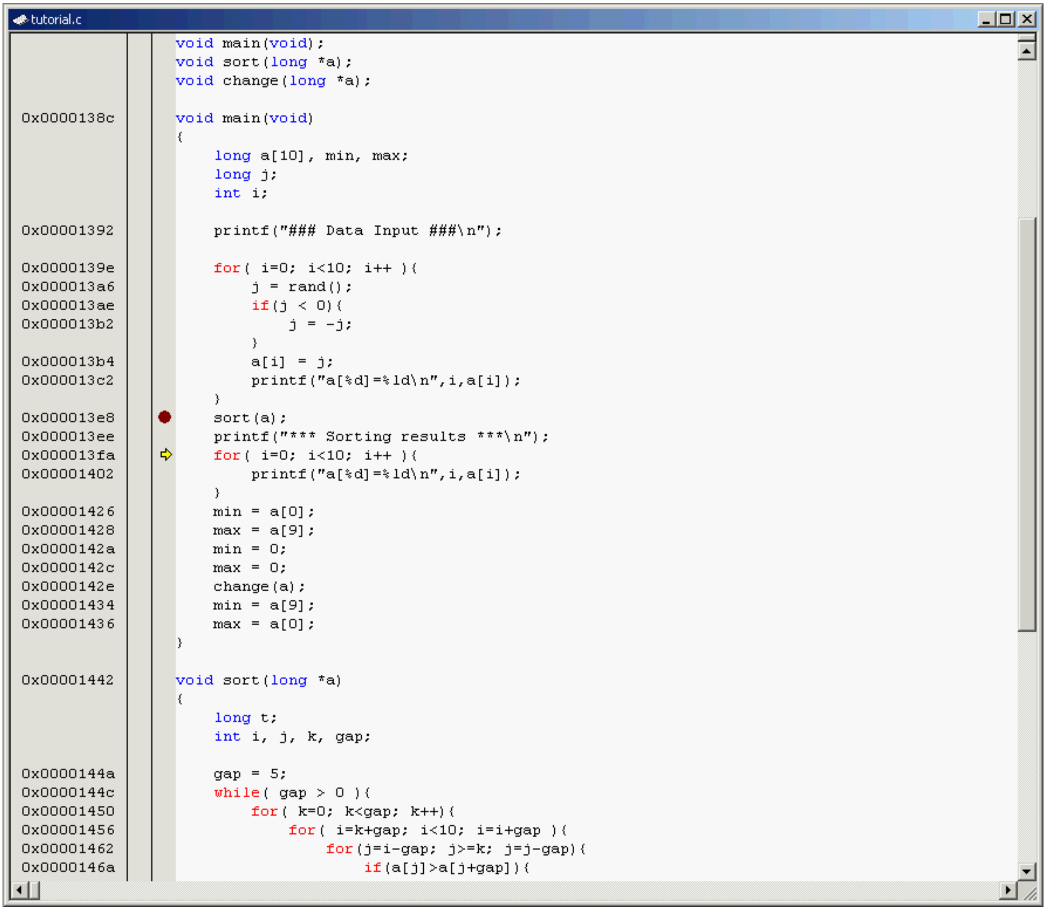


Figure 5.25 Step Over Button



```
                   void main(void);
                   void sort(long *a);
                   void change(long *a);

0x0000138c         void main(void)
                   {
                       long a[10], min, max;
                       long j;
                       int i;

0x00001392             printf("### Data Input ###\n");

0x0000139e             for( i=0; i<10; i++ ){
0x000013a6                 j = rand();
0x000013ae                 if(j < 0){
0x000013b2                     j = -j;
                           }
0x000013b4                 a[i] = j;
0x000013c2                 printf("a[%d]=%ld\n",i,a[i]);
                       }
0x000013e8       ●     sort(a);
0x000013ee             printf("*** Sorting results ***\n");
0x000013fa       ⇨     for( i=0; i<10; i++ ){
0x00001402                 printf("a[%d]=%ld\n",i,a[i]);
                       }
0x00001426             min = a[0];
0x00001428             max = a[9];
0x0000142a             min = 0;
0x0000142c             max = 0;
0x0000142e             change(a);
0x00001434             min = a[9];
0x00001436             max = a[0];
                   }

0x00001442         void sort(long *a)
                   {
                       long t;
                       int i, j, k, gap;

0x0000144a             gap = 5;
0x0000144c             while( gap > 0 ){
0x00001450                 for( k=0; k<gap; k++){
0x00001456                     for( i=k+gap; i<10; i=i+gap ){
0x00001462                         for(j=i-gap; j>=k; j=j-gap){
0x0000146a                             if(a[j]>a[j+gap]){
```

Figure 5.26 Source Window (Step Over)

When the printf function has been executed, *** Sorting results *** will be displayed in the [Simulated I/O] window.

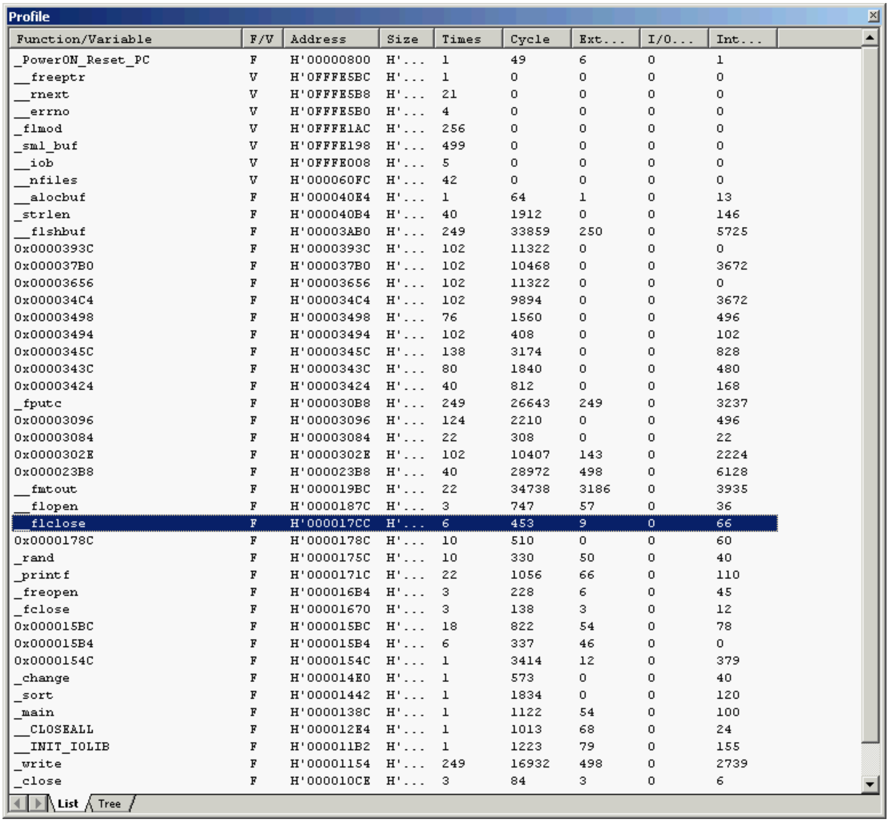
5.3.8 Checking Profile Information

The profile information can be checked in the [Profile] window.

- Clicking the [Go] button and continuing execution from the current PC executes the SLEEP instruction and stops.

[List] Sheet: Displays the profile information as a list.

- Open the [Profile] window by selecting [Profile] from the [View->Performance] menu. The [List] sheet will be displayed.



Function/Variable	F/V	Address	Size	Times	Cycle	Ext...	I/O...	Int...
_PowerON_Reset_PC	F	H'00000800	H'...	1	49	6	0	1
__freeptr	V	H'0FFFE5BC	H'...	1	0	0	0	0
__rnext	V	H'0FFFE5B8	H'...	21	0	0	0	0
__errno	V	H'0FFFE5B0	H'...	4	0	0	0	0
_flmod	V	H'0FFFE1AC	H'...	256	0	0	0	0
_sml_buf	V	H'0FFFE198	H'...	499	0	0	0	0
__iob	V	H'0FFFE008	H'...	5	0	0	0	0
__nfiles	V	H'000060FC	H'...	42	0	0	0	0
__alocbuf	F	H'000040E4	H'...	1	64	1	0	13
_strlen	F	H'000040B4	H'...	40	1912	0	0	146
__flshbuf	F	H'00003AB0	H'...	249	33859	250	0	5725
0x0000393C	F	H'0000393C	H'...	102	11322	0	0	0
0x000037B0	F	H'000037B0	H'...	102	10468	0	0	3672
0x00003656	F	H'00003656	H'...	102	11322	0	0	0
0x000034C4	F	H'000034C4	H'...	102	9894	0	0	3672
0x00003498	F	H'00003498	H'...	76	1560	0	0	496
0x00003494	F	H'00003494	H'...	102	408	0	0	102
0x0000345C	F	H'0000345C	H'...	138	3174	0	0	828
0x0000343C	F	H'0000343C	H'...	80	1840	0	0	480
0x00003424	F	H'00003424	H'...	40	812	0	0	168
_fputc	F	H'000030B8	H'...	249	26643	249	0	3237
0x00003096	F	H'00003096	H'...	124	2210	0	0	496
0x00003084	F	H'00003084	H'...	22	308	0	0	22
0x0000302E	F	H'0000302E	H'...	102	10407	143	0	2224
0x000023B8	F	H'000023B8	H'...	40	28972	498	0	6128
__fmtout	F	H'000019BC	H'...	22	34738	3186	0	3935
__flopen	F	H'0000187C	H'...	3	747	57	0	36
__flclose	F	H'000017CC	H'...	6	453	9	0	66
0x0000178C	F	H'0000178C	H'...	10	510	0	0	60
_rand	F	H'0000175C	H'...	10	330	50	0	40
_printf	F	H'0000171C	H'...	22	1056	66	0	110
_freopen	F	H'000016B4	H'...	3	228	6	0	45
_fclose	F	H'00001670	H'...	3	138	3	0	12
0x000015BC	F	H'000015BC	H'...	18	822	54	0	78
0x000015B4	F	H'000015B4	H'...	6	337	46	0	0
0x0000154C	F	H'0000154C	H'...	1	3414	12	0	379
_change	F	H'000014E0	H'...	1	573	0	0	40
_sort	F	H'00001442	H'...	1	1834	0	0	120
_main	F	H'0000138C	H'...	1	1122	54	0	100
__CLOSEALL	F	H'000012E4	H'...	1	1013	68	0	24
__INIT_IOLIB	F	H'000011B2	H'...	1	1223	79	0	155
_write	F	H'00001154	H'...	249	16932	498	0	2739
_close	F	H'000010CE	H'...	3	84	3	0	6

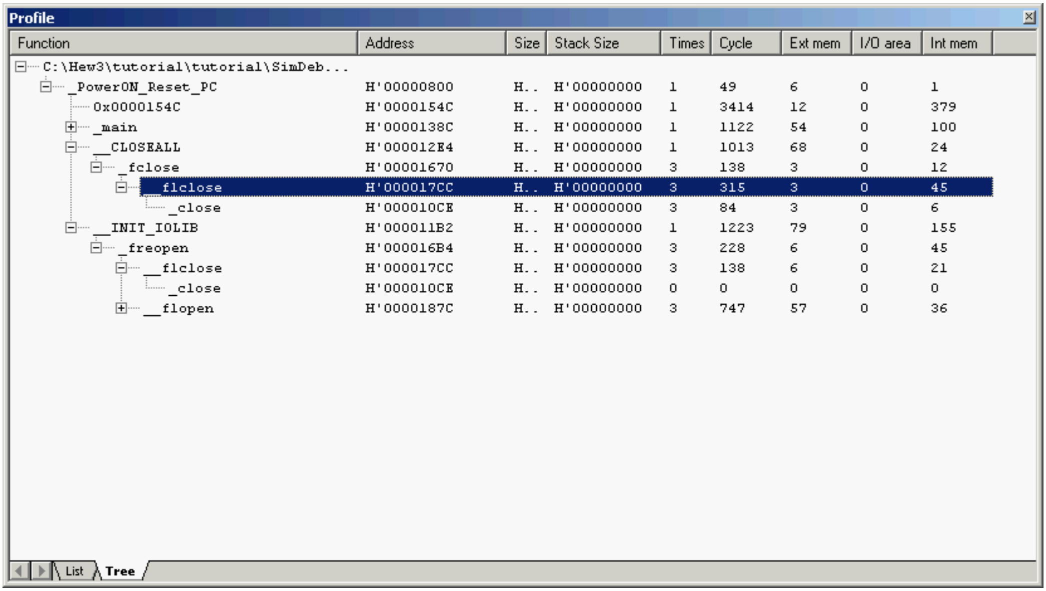
Figure 5.27 Profile Window (List Sheet)

In above figure, it can be found that the __flclose function was called six times, the execution cycle was 453, the external memory was accessed six times, and the internal memory was accessed 66 times.

It is possible to search for the critical path, such as a function that is called or accesses the memory many times, for the program performance.

[Tree] Sheet: Displays the profile information as a tree diagram.

- Select the [Tree] sheet. Double-clicking the function name in the [Profile] window expands or minimizes the tree structure.



Function	Address	Size	Stack Size	Times	Cycle	Ext mem	I/O area	Int mem
C:\Hew3\tutorial\tutorial\SimDeb...								
_PowerON_Reset_PC	H'00000800	H..	H'00000000	1	49	6	0	1
0x0000154C	H'0000154C	H..	H'00000000	1	3414	12	0	379
_main	H'0000138C	H..	H'00000000	1	1122	54	0	100
__CLOSEALL	H'000012E4	H..	H'00000000	1	1013	68	0	24
_fclose	H'00001670	H..	H'00000000	3	138	3	0	12
flclose	H'000017CC	H..	H'00000000	3	315	3	0	45
_close	H'000010CE	H..	H'00000000	3	84	3	0	6
__INIT_IOLIB	H'000011B2	H..	H'00000000	1	1223	79	0	155
_freopen	H'000016B4	H..	H'00000000	3	228	6	0	45
__flclose	H'000017CC	H..	H'00000000	3	138	6	0	21
_close	H'000010CE	H..	H'00000000	0	0	0	0	0
__flopen	H'0000187C	H..	H'00000000	3	747	57	0	36

Figure 5.28 Profile Window (Tree Sheet)

In above figure, it can be found that the __flclose function was called three times from the _fclose function, the execution cycle was 309, the external memory was accessed three times, and the internal memory was accessed 45 times.

[Profile-Chart] Window: Displays the relation of calls for a specific function.

- Select the __flclose function on the [Profile] window. Open the pop-up menu by right clicking the mouse on the [Profile] window, and select [View Profile-Chart] to display the [Profile-Chart] window.

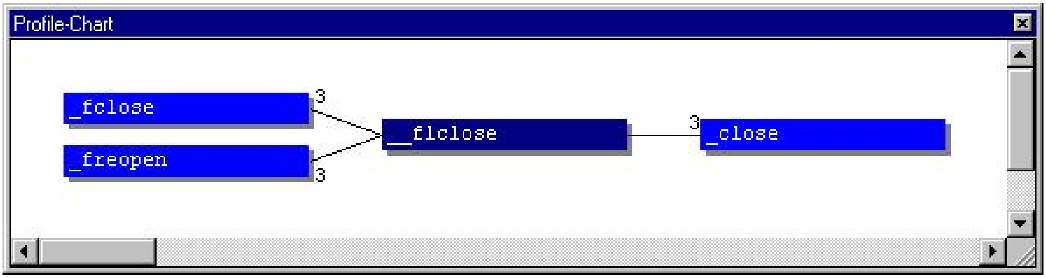


Figure 5.29 Profile-Chart Window

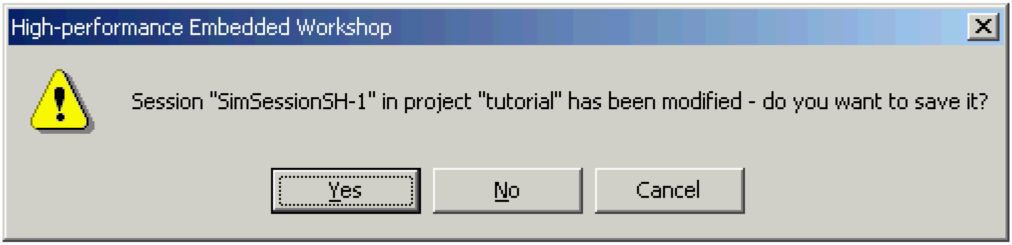
In above figure, it can be found that the __flclose function was called three times from the _fclose and _freopen functions, and the _close function was called three times.

This is the end of the tutorial using the simulator/debugger.

Section 6 Exiting from the HEW

Selecting [File->Exit] will close the HEW. Depending on the setting, the message box (figure 6.1) which asks you whether to save the session or not will be launched. Not only the setting like this regarding exiting from the HEW but also setting regarding initiation of the HEW can to specified via [Tools->Options…]. For details, refer to the High-Performance Embedded Workshop 3 User’s Manual.

The other tools launched from the HEW will not be closed when the HEW is closed. Close the tools by yourself.



High-performance Embedded Workshop

Session "SimSessionSH-1" in project "tutorial" has been modified - do you want to save it?

Figure 6.1 Confirmation Dialog Box for Saving Session on Exiting from the HEW

SuperH RISC engine
High-Performance Embedded Workshop 3
Tutorial

Publication Date: Rev.2.00, June 25, 2004
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Technical Documentation & Information Department
Renesas Kodaira Semiconductor Co., Ltd.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500 Fax: <1> (408) 382-7501

Renesas Technology Europe Limited.
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, United Kingdom
Tel: <44> (1628) 585 100, Fax: <44> (1628) 585 900

Renesas Technology Europe GmbH
Dornacher Str. 3, D-85622 Feldkirchen, Germany
Tel: <49> (89) 380 70 0, Fax: <49> (89) 929 30 11

Renesas Technology Hong Kong Ltd.
7/F., North Tower, World Finance Centre, Harbour City, Canton Road, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2375-6836

Renesas Technology Taiwan Co., Ltd.
FL 10, #99, Fu-Hsing N. Rd., Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
26/F., Ruijin Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1, Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Colophon 1.0

SuperH RISC engine
High-Performance Embedded Workshop 3
Tutorial

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan