## 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.
- 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. 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; anticrime 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 majorityowned subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.



# M16C/60 Series PC7501 Emulator

## C0 Coverage Facility

#### Summary

The full-spec emulator PC7501 for the M16C/60 series incorporates the C0 coverage facility. This document explains how to use the C0 coverage facility while using the PC7501 emulator.

Explained in this document is for the case where the user system incorporating an M16C/60 series microcomputer and the PC7501 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 M16C/60 series microcomputers.

#### Table of Contents

1.	Overview	2
2.	Facilities Used	2
3.	Preparing the Software	2
3.1	Getting Started	2
3.2	Installing the Software Supplied with the PC7501 Emulator	2
3.3	Auto Update Utility	2
4.	Operational Description	3
4.1	Starting the High-performance Embedded Workshop	3
4.2	Opening a Workspace	4
4.3	Downloading the Target Program	8
4.4	Measuring the C0 Coverage	9
5.	Frequently Asked Questions	. 18
5.1	Why do the measurement result in source units and that in the coverage window differ?	. 18
6.	Related Documents	. 20



#### 1. Overview

The coverage facility refers to the one that measures the extent of a test covered. The coverage rate of a test program measured while it was run is used to determine the validity of the test program. The coverage facility comes in several types such as C0 coverage and C1 coverage. Incorporated in the PC7501 is the C0 coverage. Also known as instruction coverage, the C0 coverage allows measuring the coverage of whether instructions were executed. On the other hand, the C1 coverage, also referred to as branch coverage, allows measuring the coverage of whether a branch occurred on true or false side of a conditional branch instruction.

The PC7501 can not only display the result of a C0 coverage measurement in the C0 coverage window, but also permits you to verify instruction execution or non-execution in source line units using the editor window columns, as well as verify instruction execution or non-execution in address units in the memory window. The coverage measurement area can be any area comprised of 1 to 32 blocks (up to 8 Mbytes) beginning with the 256-kbyte boundary. There is another thing to be noted for the PC7501's C0 coverage facility that its measurement result includes prefetches.

#### 2. Facilities Used

In this document, the method of C0 coverage measurement is explained using the sample program included in the CD-ROM supplied with the PC7501 emulator or a downloaded package from the Renesas website.

The tool versions used are listed below.

M16C R8C PC7501 Emulator Debugger V.1.03 Release 00 for the M16C series

#### 3. Preparing the Software

#### 3.1 Getting Started

Install the software included in the CD-ROM that is supplied with the PC7501 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 PC7501 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 PC7501 Emulator

Execute M16cPc7501 Debugger.exe present in the CD-ROM supplied with the PC7501 emulator.

For details on how to install, see the PC7501 emulator setup guide published at the Renesas website. During installation work, follow the instructions displayed on the screen. The installation procedure is omitted here.

#### 3.3 Auto Update Utility

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 tool is available.



#### 4. Operational Description

This section describes the method for using the C0 coverage facility 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 PC7501 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  $\rightarrow$  High-performance Embedded Workshop  $\rightarrow$  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.

🖗 High-performance Embedded Workshop	- ð 🗙
File Edit View Project Build Debug Setup Tools Test Window Help	
Welcome!     Image: Create a new project workspace     OK   Cancel   Image: Create a new project workspace:     Administration     Image: Create a new project workspace:     Image: Create a new project workspace	
× 9J 9T AJ AT 82 8 T 2⁄ № ₩ ?	
	×
Ready INS	NUM

Select the "Browse Another Project Workspace" radio button in the Welcome! dialog box and click the OK button.

Welcome!		? 🛛
👔 🤉 C <u>C</u> reate a r	new project workspace	OK Cancel
Den are	ecent project workspace:	Administration
Browse to	another project workspace	



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

Open Work	space		? 🛛
Look jn: 🔀	) Tutorial	• 🔁 (	* 💷 *
Tutorial			
File <u>n</u> ame:	Tutorial	(	Select
Files of <u>t</u> ype:	HEW Workspaces (*.hws)		Cancel

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\E8\M16C\Tutorial

Examples of directory:

C:\hew3\Tools\Renesas\DebugComp\Platform\E8\M16C\Tutorial

C:\hew2\Tools\Renesas\DebugComp\Platform\E8\M16C\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.

High-pe	erformance Embedded Workshop 🛛 🔀
<u>.</u>	The Workspace you are about to open was created with an earlier version of HEW. The data files for the workspace, projects and sessions will be updated. Once updated this workspace cannot be opened by an older version of HEW. Backup versions of your old files will be created in the workspace and project directories with the prefix 'old_version_xxx'. Do you wish to continue ?



(4) When the workspace is opened, you are ready to use the High-performance Embedded Workshop. Change sessions to connect the PC7501 emulator. To do it, change "DefaultSession" in the toolbar to "SessionM16C\_R8C\_PC7501\_Emulator."

Before changing sessions here, be sure that the interface select switch on the back panel of the PC7501 emulator main body is set to the USB side, and then turn the power for the PC7501 emulator on.



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





(5) The Init (M16C R8C PC7501 Emulator) dialog box will be displayed. Select the "USB" radio button on the MCU tab of the dialog box and click OK.

Init (M16C R8C PC7501 Emulator)	
MCU Debugging Information Emulator Script	
MCU: M16C62P.mcu	
CLPT CLAN CUSE	B
Serial No.: 5AS1567B Target	sk -
Debug Option	Automatically selected
<ul> <li>Enable the Address Match Interrupt Break Function.</li> <li>Debug the program using the CPU Rewrite Mode.</li> </ul>	
OK Cancel He	

- If your emulator is connected via LPT or LAN instead of USB, visit the Renesas website and see the PC7501 user's manual.
- (6) The MCU Setting dialog box will be displayed. Select "Single-Chip Mode" for Processor Mode on the MCU tab of the dialog box and click OK.

MCU Setting	MCU Status
MCU: M16C/62P	NMI*: H
Processor Mode Single-Chip Mode External Data Bus Width: 16-bit Memory Space Expansion: Normal Mode I PM13 (b3 of 000005H) is '1'. PM10 (b0 of 000005H) is '1'.	HOLD*: H RDY*: H CNVss: NC BYTE NC
Debug Option Disable Internal Flash ROM (for 10MHz or below frequency).	ow operating



(7) A connection of the PC7501 emulator will be completed, by which the High-performance Embedded Workshop screen becomes ready to operate. When this connection is complete, a message "Connected" is displayed on the Debug tab of the output window.



#### 4.3 Downloading the Target Program

First, download the target program.

(1) In the example here, we'll download the program "Tutorial.x30 - 00000000" appearing beneath Download Modules in Workspace. Double-click it to download.



When the program is successfully downloaded, its icon is marked by a down arrow.





## 4.4 Measuring the C0 Coverage

To view the results of coverage measurements, open the coverage window.

(1) From the View menu, choose Code and then Coverage.

🖗 Tutorial	- High-performance Embedded	W	/orkshop - [ncrt0.a30]
🧼 File 🛛 Edit	View Project Build Debug Setu	Р	Tools Test Window Help
🗋 🗅 😅 🖬	Differences		▲ 44 €
0 15 1	Мир Мар		7 9 <del>7</del> 9 <del>1</del> 1 1 1 1 1 1
	Command Line Ctrl+L		
Tutor ⊡∵ri⊋ 1	TCL Toolkit Chrl+Shift+K		
- <del>-</del>	Workspace Alt+K		ine Sour A. S. Source
	Dutput Alt+0		27 ; Intern 28 ;
	Status Bar Alt+A		29
	💭 Disassembly Ctrl+D		30 31
	CPU	F	32 start: 33 ;
	Symbol	Þ	34 ; after 35 ;
	<u>G</u> raphic	٠	. 36 F43C8 ⊄>
	🛃 Script		38 F43D0
	Gode	C	Coverage
	Break	٠	Stack Trace Ctrl+K
	Irace	Þ	43 .else
	RTOS	۲	45 .endif
		L	

(2) The coverage window will open. The window shows the function names of the target program under the heading "Function," as well as the start addresses under "Start," the end addresses under "End" and the C0 coverage measurement result under "Coverage." At this point in time, the values under "Coverage" are 0.00% because the program is not run yet.

In the explanation in this document, the Auto Display Update icon is "enabled" (the icon is depressed when enabled).

Function	Start	End	Coverage
change	OF41BE	OF4219	0.00 %
init	OF4014	OF4100	0.00 %
sort	OF4102	OF41BD	0.00 %
abort	OF4376	OF4376	0.00 %
main	OF421A	OF4223	0.00 %
tutorial	OF4224	OF4375	0.00 %



• Description of the icons

## 멶

Source selection: Selects the source file for which the results of coverage measurements are displayed.

When you select the Source Select icon, the Select Source Files dialog box is displayed, allowing you to display only a selected source file in the coverage window.

Select Source File		
C Select all files	le name	
Tutorial.c		•
Select a source file:		
	OK	Cancel



the icon is depressed when enabled

Automatic display update: Automatically updates the displayed coverage measurement result when the target program stops.

## ۲

Display update: Updates the displayed coverage measurement result.

## ѷ

Initialization: Initializes the coverage measurement result.



(3) Double-click any function line displayed in the coverage window. The function you've selected will be displayed in the editor window.



(4) We'll now try stopping the program immediately before a call to the sort function to see how the C0 coverage will be measured.

Double-click in the Address Match Breakpoint column at line No. 40 of the source file "Tutorial.c."



Since we've set the program to break immediately before the source address "F4276" of the sort function is executed, the functions change and abort present ahead of it are not executed.



(5) Run the target program to measure the C0 coverage. Select the Run After Reset icon to run the program.



The program can also be run from the Debug menu.



(6) While a C0 coverage measurement is in progress, the "Coverage" values in the coverage window are shown as "- %."

Function	Start	End	Coverage
change	OF41BE	OF4219	- %
init	OF4014	OF4100	- %
sort	OF4102	OF41BD	- 8
abort	OF4376	OF4376	- 8
main	OF421A	OF4223	- 8
tutorial	OF4224	OF4375	- 8



(7) The program will stop at the breakpoint, showing the result of the C0 coverage measurement in the coverage window.

🖻 🙀 Tuto	vial				<b>6 6</b>			
	Assembly source fil	e		Line	Sour	Α.,	S.,	Source
	≚] ncrt0.a30	201			6 F4259			j = -j;
	C source file			3	7			}
	≚] sort.c			3	8 F4262			a[i] = j;
	≚] Tutorial.c				9			
🗄 🔂 🛙	Download modules				0 F4276	•	⇔	sort(a);
	🕽 Tutorial.x30 - 0	0000000		4	1 F427E			change (a);
ė 🔂 🕻	Dependencies			- 4	2			
	(=S)		020		3 F4286			p sam->s0=a[0];
Projects	Templates 4	Navidation	Test 1		54 545 5 5 S			
Projects	🛃 Templates 🧹	Navigation	Test	4	4 F4298			p_sam->s1=a[1];
Projects	률 Templates 🔤	Navigation	Test	4	5 F42AE			p_sam->s1=a[1]; p_sam->s2=a[2];
		Navigation	Test		5 F42AE 6 F42C6			p_sam->s1=a[1]; p_sam->s2=a[2]; p_sam->s3=a[3];
Projects		Navigation	Test		5 F42AE 6 F42C6 7 F42DE			p_sam->s1=a[1]; p_sam->s2=a[2]; p_sam->s3=a[3]; p_sam->s4=a[4];
		Navigation	Test		5 F42AE 6 F42C6 7 F42DE 8 F42F6			p_sam->s1=a[1]; p_sam->s2=a[2]; p_sam->s3=a[3]; p_sam->s4=a[4]; p_sam->s5=a[5];
		End	Coverage		5 F42AE 6 F42C6 7 F42DE 8 F42F6 9 F430E			<pre>p_sam-&gt;s1=a[1]; p_sam-&gt;s2=a[2]; p_sam-&gt;s3=a[3]; p_sam-&gt;s4=a[4]; p_sam-&gt;s5=a[5]; p_sam-&gt;s6=a[6];</pre>
* 🗿 💊 🗙	<u> </u>				5 F42AE 6 F42C6 7 F42DE 8 F42F6 9 F43OE 0 F4326			<pre>p_sam-&gt;s1=a[1]; p_sam-&gt;s2=a[2]; p_sam-&gt;s3=a[3]; p_sam-&gt;s4=a[4]; p_sam-&gt;s5=a[5]; p_sam-&gt;s6=a[6]; p_sam-&gt;s7=a[7];</pre>
Function	Start	End	Coverage		5 F42AE 6 F42C6 7 F42DE 8 F42F6 9 F430E 0 F4326 1 F433E			<pre>p_sam-&gt;s1=a[1]; p_sam-&gt;s2=a[2]; p_sam-&gt;s3=a[3]; p_sam-&gt;s4=a[4]; p_sam-&gt;s5=a[5]; p_sam-&gt;s6=a[6]; p_sam-&gt;s6=a[6];</pre>
Function change init	Start OF41BE	End OF4219	Coverage 0.00 %		5 F42AE 6 F42C6 7 F42DE 8 F42F6 9 F430E 0 F4326 1 F433E 2 F4356			<pre>p_sam-&gt;s1=a[1]; p_sam-&gt;s2=a[2]; p_sam-&gt;s3=a[3]; p_sam-&gt;s4=a[4]; p_sam-&gt;s5=a[5]; p_sam-&gt;s6=a[6]; p_sam-&gt;s8=a[8]; p_sam-&gt;s8=a[8]; p_sam-&gt;s9=a[9];</pre>
Function change init sort	Start OF41BE OF4014	End 0F4219 0F4100	Coverage 0.00 % 100.00 %		5 F42AE 6 F42C6 7 F42DE 8 F42F6 9 F43OE 0 F4326 1 F433E 2 F4356 3 F436E			<pre>p_sam-&gt;s1=a[1]; p_sam-&gt;s2=a[2]; p_sam-&gt;s3=a[3]; p_sam-&gt;s4=a[4]; p_sam-&gt;s5=a[5]; p_sam-&gt;s6=a[6]; p_sam-&gt;s6=a[6];</pre>
Function change	Start OF41BE OF4014 OF4102	End 0F4219 0F4100 0F418D	Coverage 0.00 % 100.00 % 1.06 %		5 F42AE 6 F42C6 7 F42DE 8 F42F6 9 F430E 0 F4326 1 F433E 2 F4356			<pre>p_sam-&gt;s1=a[1]; p_sam-&gt;s2=a[2]; p_sam-&gt;s3=a[3]; p_sam-&gt;s4=a[4]; p_sam-&gt;s5=a[5]; p_sam-&gt;s6=a[6]; p_sam-&gt;s8=a[8]; p_sam-&gt;s8=a[8]; p_sam-&gt;s9=a[9];</pre>

In this case, the function change to be called after the function sort and the function abort used for interrupts were not executed. Therefore, as you can see, their measured values are shown as "0.00%."

(8) The result of C0 coverage can also be verified in the editor window. Choose "Set Display Columns" from the Edit menu.





(9) The Entire Column State of Editor dialog box will be displayed. Select the "Coverage" and "Coverage-ASM" check boxes in the dialog box and click OK.



Coverage: Permits you to verify C0 coverage in C source units.

Coverage-ASM: Permits you to verify C0 coverage in assembly units.

(10) The "Coverage" column will be added in the editor window, allowing you to verify C0 coverage visually. The yellow part of this column represents executed instructions, and the gray part represents unexecuted instructions.



In mixed mode or disassembly mode too, it is possible to verify C0 coverage visually in the same way.



(11) What's more, it is possible to verify C0 coverage visually in address units using the memory window. Choose CPU and then Memory from the View menu.

👂 Tutorial -	High-performance Embedde	d W	orkshop - [Tutor	ial.c]
🎾 File Edit	View Project Build Debug Set	up	Tools Test Windo	w Help
🗅 😅 🖬	Differences			<b>→</b> M §
011	MAP Map		T EL EF EL E	₹ ₽} ₽ ₽
	Command Line Ctrl+L			
Tutori 🚰 ⊡ T 🖓 ⊡	🍇 TCL Toolkit Ctrl+Shift+K			
	Workspace Alt+k	2		Line So 36 F4
	🔊 Output Alt+C		37	
	Status Bar Alt+A		38 F4	
		202		39
	Disassembly Ctrl+D	)		40 F4
		Þ	R1 Registers	Ctrl+R 4
Project:	<u>S</u> ymbol	Þ	Memory	Ctrl+M 4
	<u>G</u> raphic	٠	<u>1/0</u> <u>I</u> O	Ctrl+I 4
8 🌆 🧆	🔀 Script		🙀 Stat <u>u</u> s	Ctrl+U 4
Function	⊆ode	٠	💯 RamMonitor	4
change	Break		0.00 %	50 F4
init	5. e.g.v.	22	0.00 %	51 F4 52 F4
sort	Trace	F	1.06 %	52 F4 53 F4
abort main	<u>R</u> TOS	۲	0.00 %	54 F4
tutorial	N#4774 N#4375		74 76 8	

(12) The Display Start Address dialog box will be displayed. Set "H'F4270" for Display Start Address in this dialog box and click OK.

Display Address		? 🛛
Display Address:	F4270	- 🔊
Scroll Start Address:	000000	- 🔊
Scroll End Address:	OFFFF	• 🔊
ОК	Cancel	



(13) The memory window will open. Right-click in the window and from the ensuing menu, choose Coverage and then Enable.

1 11 100		16 10	±10 {	<u>S</u> et		.16 .3	2 2	2	
Address	+0	+1	+2	Eill		+A	+B	- +C	+1
OF4270	04	C9	1B	<u>M</u> ove	1	FD	02	41	01
OF4280	14	04	FD	Compare		в5	FC	75	21
OF4290	C9	24	77	Test	8	FA	73	в5	F
OF42AO	77	E5	75	Save Memory contents		75	2 F	1A	04
OF42BO	FA	73	в5 –	Dave Memory concents		75	2F	1C	04
OF42CO	77	E5	75	Search		в5	FC	77	44
OF42DO	77	E5	75	Search Next	3	75	2 F	22	04
OF42EO	FA	73	в5 –			75	2 F	24	04
OF42FO	77	E5	75	<u>A</u> ddress	1	в5	FC	77	4
OF4300	77	E5	75	Sgroll Area		75	2 F	2A	04
OF4310	FA	73	в5	Register	•	75	2 F	2C	04
OF4320	77	E5	75	Followed Stack Pointer		в5	FC	77	4
OF4330	77	E5	75		8	75	2 F	32	04
OF4340	FA	73	в5	Set Start Up Symbol		75	2 F	34	04
OF4350	77	E5	75	Refresh		в5	FC	77	44
OF4360	77	E5	75	Lock Refresh	1	75	2 F	ЗA	04
OF4370	FA	D9	Ов _	Fock Kenesii		C6	41	7D	E2
				Data Length	•				
				Radix	•				
nort0.a30	0	Tutori		Code		-			
		ruton		Layout					
						-			
				<u>C</u> olumn					
				Co <u>v</u> erage	·	Enat	ole		
				<u>S</u> ave					
				Load					

(14) The memory window will be color-coded. Shown in the blue part are the executed instructions, and those in the gray part are the unexecuted instructions.

<b>I</b> 11 mm		<u>16 10</u>	± <u>10</u>	8	2 d	be dis	க	ão da	f.	.d	.16 .3	2 2	2			
Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
OF4270	04	C9	1B	FE	FE	CB	75	C1	14	04	FD	02	41	OF	75	C1
OF4280	14	04	FD	BE	41	OF	73	в4	FA	73	в5	FC	75	2 F	14	04
OF4290	C9	24	77	E5	75	2 F	16	04	73	в4	FA	73	в5	FC	С9	44
OF42A0	77	E5	75	2 F	18	04	с9	24	77	E5	75	2 F	1A	04	73	в4
OF42BO	FA	73	в5	FC	77	44	08	00	77	E5	75	2 F	1C	04	С9	24
OF42CO	77	E5	75	2 F	1E	04	73	в4	FA	73	в5	FC	77	44	OC	00
OF42DO	77	E5	75	2 F	20	04	с9	24	77	E5	75	2 F	22	04	73	в4
OF42E0	FA	73	в5	FC	77	44	10	00	77	E5	75	2 F	24	04	C9	24
OF42FO	77	E5	75	2 F	26	04	73	в4	FA	73	в5	FC	77	44	14	00
OF4300	77	E5	75	2 F	28	04	C9	24	77	E5	75	2 F	2A	04	73	в4
OF4310	FA	73	в5	FC	77	44	18	00	77	E5	75	2 F	2C	04	С9	24
OF4320	77	E5	75	2 F	2 E	04	73	в4	FA	73	в5	FC	77	44	1C	00
OF4330	77	E5	75	2 F	30	04	С9	24	77	E5	75	2 F	32	04	73	в4
OF4340	FA	73	в5	FC	77	44	20	00	77	E5	75	2 F	34	04	С9	24
OF4350	77	E5	75	2 F	36	04	73	в4	FA	73	в5	FC	77	44	24	00
OF4360	77	E5	75	2 F	38	04	с9	24	77	E5	75	2 F	3A	04	D9	OB
OF4370	FA	D9	OB	FC	7D	F2	FЗ	в4	7D	<b>E</b> 2	C6	41	7D	<b>E</b> 2	6D	4E
OF4380	73	FO	10	04	73	F2	12	04	FD	AC	43	OF	7D	в4	77	40



(15) Try verifying the content of the internal RAM following the same procedure. Right-click in the memory window and from the ensuing menu, choose Start Address. The PC7501's coverage facility permits you to measure the coverage of data areas (whether addresses accessed), not just the coverage of instruction execution.

1 11 111	<b></b> <u>16</u>	<u>S</u> et	族	.f	Q
Address	+0	<u>E</u> ill		+8	+
OF4270	04	<u>M</u> ove		14	0
OF4280	14	Compare		FA	7
OF4290	С9	Test		73	в
OF42AO	77	Save Memory contents		77	E
OF42BO	FA	2-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1		77	Ε
OF42CO	77	Search		FA	7
OF42DO	77	Search Next		77	E
OF42EO	FA			77	E
OF42FO	77	<u>A</u> ddress		FA	7
OF4300	77	Scroll Area		77	E
OF4310	FA	Register	- 61	77	E
OF4320	77	Followed Stack Pointer	- 11	FA	7
OF4330	77			77	Ε
OF4340	FA	Set Start <u>U</u> p Symbol	- 41	77	E

(16) The Display Start Address dialog box will be displayed. Set "H'400" for Address in this dialog box and click OK.



(17) The window permits you to verify the variables in the internal RAM that were accessed and those that were not accessed.

<b>1</b> 11 m		<u>16</u> 10	± <u>10</u>	8	2 d	bc 🗟	あ	ão da	<b>1</b> .	.d	.16 .3	2 2	2			
Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000400	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000410	DF	D5	F6	OF	C6	41	00	00	7E	16	00	00	81	27	00	00
000420	6B	44	00	00	4B	79	00	00	FB	15	00	00	<b>E</b> 2	59	00	00
000430	FB	1C	00	00	54	3 F	00	00	F6	OF	00	00	00	00	00	00
000440	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000450	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000460	00	00	00	00	6C	0A	00	00	00	03	00	00	E3	55	73	AF
000470	81	48	08	01	68	D4	3A	56	CF	DA	F4	FB	7B	5A	FF	FE
000480	AB	47	27	2 D	00	DE	C5	81	F4	вО	D8	C1	7D	BD	EB	63
000490	9A	01	55	C1	BE	9E	8 E	8A	6E	27	DF	EB	7C	ЗF	ΕO	F5
0004A0	A1	66	FO	7A	83	24	26	28	4F	99	AC	8 F	DF	E8	F4	E5
0004B0	88	С3	8 E	08	A2	C2	1C	62	F7	ЗD	CD	FF	ЗF	5F	E7	AF
0004c0	83	6E	43	24	31	75	37	71	46	BF	FF	EE	8A	70	8 D	C7
0004D0	A8	60	OF	61	18	17	C5	48	42	2 F	7D	57	1D	F4	7в	В6
0004E0	C9	1A	52	04	81	19	4D	15	54	AB	в6	4F	2 F	D5	ЗD	FF
0004F0	4D	54	C1	00	00	D6	DA	23	BF	4F	DE	FD	C8	43	00	OF
000500	05	1E	F8	DG	8 E	16	в6	74	в1	в8	AA	A6	FЗ	ЗD	E5	FF
000510	11	CF	C3	1F	56	8C	14	05	8 E	7F	BE	45	74	ED	39	F4

These coverage facilities allow you not only to verify the validity of a test program, but also measure the usage rate, etc. of the RAM space in actual program operation.



#### 5. Frequently Asked Questions

5.1 Why do the measurement result in source units and that in the coverage window differ?

Run the sample program once up until it ends and see the result of C0 coverage measurement. It will look like the one shown below.

Function	Start	End	Coverage
change	OF41BE	OF4219	100.00 %
init	OF4014	OF4100	100.00 %
sort	OF4102	OF41BD	100.00 %
abort	OF4376	OF4376	100.00 %
main	OF421A	OF4223	100.00 %
tutorial	OF4224	OF4375	98.22 %

Taking a look at the C0 coverage of the function tutorial in C source units, you'll find that the function appears to have been executed in whole.

23       void tutorial (void)         24       {         25       F4224       {         26       long j;       int i;         27       int i;       struct Sample far *p_sam;         29       g       g         30       F4227       p_sam= &st         31       F4231       init (p_sam);         32       for ( i=0; i<10; i++ )(         34       F4248       j = rand();         35       F4254       if(j < 0) (         36       F4259       j = -j;         37       }       j         38       F4262       a[i] = j;         39        j         40       F4276       sort(a);         41       F4286       p_sam->s0=a[0];         42       g       g         43       F4286       p_sam->s1=a[1];         44       F4298       p_sam->s3=a[3];         45       F420E       p_sam->s3=a[3];         46       F4276       p_sam->s6=a[6];         50       F4326       p_sam->s6=a[6];         51       F433E       p_sam->s9=a[9];         53       F436E       p_sam = NULL;	ne	Sour	C.,	A.,	S., Source
25       F4224       {         26       long j; int i; struct Sample far *p_sam;         29       p_sam= &st init (p_sam);         30       F4227       p_sam= &st init (p_sam);         31       F4231       init (p_sam);         32       for ( i=0; i<10; i++ )( j = rand(); if (j < 0) { j = -j;         33       F4248       j = rand();         34       F4259       if (j < 0) { j = -j;         35       F4262       a[i] = j;         39       >       sort(a);         40       F4276       change(a);         42       g_sam->s0=a[0];         43       F4286       p_sam->s1=a[1];         44       F4298       p_sam->s2=a[2];         45       F4226       p_sam->s5=a[5];         46       F4266       p_sam->s5=a[6];         47       F420E       p_sam->s5=a[6];         48       F4266       p_sam->s5=a[6];         49       F430E       p_sam->s6=a[6];         50       F4326       p_sam->s9=a[9];         51       F438E       p_sam = NULL;         54       F4374       >         55       F4376          56       void abort(void)	23				
<pre>26 long j; 27 int i; 28 struct Sample far *p_sam; 29 30 F4227 p_sam= &amp;st 31 F4231 init(p_sam); 32 f423D for(i=0; i&lt;10; i++)( 34 F4248 j = rand(); 35 F4254 if(j &lt; 0)( 36 F4259 j = -j; 37 } 38 F4262 a[i] = j; 39 } 40 F4276 sort(a); 41 F427E change(a); 42 p_sam-&gt;s0=a[0]; 42 p_sam-&gt;s1=a[1]; 45 F42AE p_sam-&gt;s1=a[1]; 45 F42AE p_sam-&gt;s1=a[1]; 46 F42C6 p_sam-&gt;s1=a[1]; 48 F42F6 p_sam-&gt;s1=a[1]; 48 F42F6 p_sam-&gt;s1=a[1]; 48 F42F6 p_sam-&gt;s1=a[1]; 50 F4326 p_sam-&gt;s1=a[3]; 47 F42DE p_sam-&gt;s1=a[1]; 51 F433E p_sam-&gt;s1=a[3]; 52 F4356 p_sam-&gt;s1=a[3]; 53 F436E p_sam-&gt;s1=a[3]; 54 F4374 } 55 f4376 ( 56 void abort(void) 57 F4376 ( 58 void abort(void) 57 F4376 ( 50 str(a); 51 str(a); 51 str(a); 51 str(a); 52 str(a); 53 str(a); 54 str(a); 55 str(a); 55 str(a); 55 str(a); 56 str(a); 57 str(a); 50 str(a); 50 str(a); 50 str(a); 50 str(a); 50 str(a); 51 str(a); 52 str(a); 53 str(a); 54 str(a); 55 str(a); 55 str(a); 55 str(a); 56 str(a); 57 str(a); 57</pre>	24				void tutorial (void)
<pre>27 int i; 28 int i; 29 sam= &amp;st 30 F4227 p_sam= &amp;st 31 F4231 init(p_sam); 32 int (p_sam); 33 F423D for (i=0; i&lt;10; i++)( 34 F4248 j = rand(); 35 F4254 if(j &lt; 0)( 36 F4259 j = -j; 37 } 38 F4262 a[i] = j; 39 int i; 40 F4276 sort(a); 41 F427E change(a); 42 int f427E p_sam-&gt;s0=a[0]; 44 F4298 p_sam-&gt;s1=a[1]; 45 F42AE p_sam-&gt;s1=a[1]; 45 F42AE p_sam-&gt;s1=a[3]; 47 F42DE p_sam-&gt;s3=a[3]; 47 F42DE p_sam-&gt;s6=a[6]; 50 F4326 p_sam-&gt;s8=a[6]; 51 F433E p_sam-&gt;s8=a[6]; 52 F4356 p_sam = NULL; 54 F4374 } 55 F4376 ( 56 void abort(void) 57 F4376 ( 58 void abort(void)</pre>	25	F4224			{
28       struct Sample far *p_sam;         29       p_sam= &st         31       F4231         33       F4231         34       F4231         35       F4248         36       F4248         37       for(i=0; i<10; i++)(	26				long j;
29 $4$ $p_sam = 4st;$ 30       F4227 $init(p_sam);$ 31       F4231 $init(p_sam);$ 32 $for(i=0; i<10; i++)(i)$ 34       F4248 $j = rand();$ 35       F4254 $if(j < 0)(i)$ 36       F4259 $j = -j;$ 37 $3$ $3$ 38       F4262 $a[i] = j;$ 39 $4$ $p_sam - s0=a[0];$ 40       F4276 $sort(a);$ 41       F4276 $p_sam - s0=a[0];$ 42 $p_sam - s0=a[0];$ $p_sam - s0=a[0];$ 44       F4298 $p_sam - s0=a[0];$ 45       F4264 $p_sam - s0=a[0];$ 46       F4266 $p_sam - s0=a[0];$ 47       F4266 $p_sam - s0=a[0];$ 48       F4266 $p_sam - s0=a[6];$ 50       F430E $p_sam - s0=a[6];$ 51       F4376 $p_sam - s0=a[6];$ 52       F4366 $p_sam - s0=a[6];$ 53       F436E $p_sam = NULL;$ 54       F4374       )	27				int i;
30 $F4227$ $p_sam = \xist;$ 31 $F4231$ init $(p_sam);$ 32 $for (i=0; i<10; i++) ($ 34 $F4248$ $j = rand();$ 35 $F4254$ if $(j < 0) ($ 36 $F4259$ $j = -j;$ 37 $3$ $a[i] = j;$ 38 $F4262$ $a[i] = j;$ 39 $a[i] = j;$ $3$ 40 $F4276$ $sort(a);$ 41 $F4276$ $sort(a);$ 42 $p_sam - s0 = a[0];$ 43 $F4286$ $p_sam - s0 = a[0];$ 44 $F4298$ $p_sam - ss0 = a[0];$ 45 $F4216$ $p_sam - ss0 = a[0];$ 46 $F4226$ $p_sam - ss0 = a[0];$ 47 $F420E$ $p_sam - ss = a[3];$ 48 $F4276$ $p_sam - ss = a[6];$ 50 $F4326$ $p_sam - s9 = a[9];$ 51 $F433E$ $p_sam - s9 = a[9];$ 52 $F436$ $p_sam = NULL;$ 54 $F4374$ $\gamma$ 55 $a[s]$ $void abort(vo$	28				struct Sample far *p sam;
<pre>31 F4231 init(p_sam); 32 33 F423D init(p_sam); 34 F4248 if(j &lt; 0); i++); 35 F4254 if(j &lt; 0); if(j &lt; 0); 36 F4259 if(j &lt; 0); if(j &lt; 0); 36 F4259 init(p_sam); 37</pre>	29				
32       a       for ( i=0; i<10; i++ ) (	30	F4227			p_sam= &st
33       F423D       for(i=0; i<10; i++)(	31	F4231			init(p_sam);
34       F4248       j = rand();         35       F4254       if(j < 0) {	32				22
35       F4254       if $(j < 0)$ {         36       F4259 $j = -j;$ 37       )         38       F4262 $a[i] = j;$ 39       )       >         40       F4276       sort(a);         41       F4276       change(a);         42        p_sam->s0=a[0];         43       F4286       p_sam->s1=a[1];         44       F4298       p_sam->s1=a[1];         45       F4266       p_sam->s2=a[2];         46       F4266       p_sam->s3=a[3];         47       F42DE       p_sam->s5=a[5];         48       F42F6       p_sam->s5=a[5];         49       F430E       p_sam->s6=a[6];         50       F4326       p_sam->s8=a[8];         52       F4356       p_sam->s8=a[9];         53       F436E       p_sam = NULL;         54       F4374       >         55        void abort (void)         57       F4376       {	33	F423D			<pre>for ( i=0; i&lt;10; i++ ) {</pre>
36       F4259 $j = -j;$ 37 $j$ $j = -j;$ 38       F4262 $a[i] = j;$ 39 $j$ $a[i] = j;$ 39 $f4276$ $p_sam > s0=a[0];$ 41       F4264 $p_sam > s1=a[1];$ 45 $p_sam > s1=a[1];$ $p_sam > s1=a[1];$ 46       F4266 $p_sam > s3=a[3];$ 47       F4266 $p_sam > s6=a[6];$ 50       F4326 $p_sam > s0=a[9];$ 51       F4356 $p_sam > s0=a[9];$ 53       F4366 $p_sam = NULL;$ 54       F43	34	F4248			j = rand();
37       ,         38       F4262       a[i] = j;         39       ,         40       F4276       sort(a);         41       F4276       change(a);         42       ,       p_sam->s0=a[0];         44       F4298       p_sam->s1=a[1];         45       F42AE       p_sam->s1=a[1];         45       F42AE       p_sam->s2=a[2];         46       F42C6       p_sam->s3=a[3];         47       F42DE       p_sam->s3=a[3];         47       F42DE       p_sam->s5=a[5];         49       F430E       p_sam->s6=a[6];         50       F4326       p_sam->s6=a[6];         51       F4356       p_sam->s9=a[9];         53       F436E       p_sam = NULL;         54       F4374       )         55       56       void abort(void)         57       F4376       {	35	F4254			<b>if</b> (j < 0){
38       F4262 $a[1] = j;$ 39       a       a[1] = j;         39       a       sort(a);         40       F4276       change(a);         41       F4276       p_sam->s0=a[0];         42       a       p_sam->s1=a[1];         43       F4286       p_sam->s1=a[1];         44       F4298       p_sam->s2=a[2];         45       F424E       p_sam->s3=a[3];         47       F42DE       p_sam->s4=a[4];         48       F42F6       p_sam->s5=a[5];         49       F430E       p_sam->s6=a[6];         50       F4326       p_sam->s6=a[6];         51       F4356       p_sam->s9=a[9];         53       F436E       p_sam = NULL;         54       F4374       >         55       a       a         56       a       a	36	F4259			j = -j;
39	37				}
40       F4276       sort(a);         41       F427E       change(a);         42       g_sam->s0=a[0];         44       F4298       p_sam->s1=a[1];         45       F42AE       p_sam->s2=a[2];         46       F42C6       p_sam->s3=a[3];         47       F42DE       p_sam->s5=a[5];         48       F42F6       p_sam->s6=a[6];         50       F4326       p_sam->s8=a[8];         52       F4356       p_sam = NULL;         54       F4374       >         55       F4376       (	38	F4262			a[i] = j;
41       F427E       change(a);         42       p_sam->s0=a[0];         44       F4298       p_sam->s1=a[1];         45       F42AE       p_sam->s2=a[2];         46       F42C6       p_sam->s3=a[3];         47       F42DE       p_sam->s5=a[5];         48       F42F6       p_sam->s6=a[6];         50       F4326       p_sam->s8=a[8];         52       F4356       p_sam->s9=a[9];         53       F4376       p_sam         56       void abort (void)         57       F4376       (					3
42       43       F4286 $p_sam ->s0=a[0];$ 44       F4298 $p_sam ->s1=a[1];$ 45       F42AE $p_sam ->s2=a[2];$ 46       F42C6 $p_sam ->s3=a[3];$ 47       F42DE $p_sam ->s4=a[4];$ 48       F42F6 $p_sam ->s5=a[5];$ 49       F430E $p_sam ->s6=a[6];$ 50       F4326 $p_sam ->s8=a[8];$ 52       F4356 $p_sam ->s9=a[9];$ 53       F4376 $p_sam ->s0=a[9];$ 56 $p_sam ->s0=a[9];$ $p_sam ->s0=a[9];$ 56 $p_sam ->s0=a[9];$ $p_sam ->s0=a[9];$ 56 $p_sam ->s0=a[9];$ $p_sam ->s0=a[9];$ 57       F4376 $($					
43       F4286 $p_sam > s0=a[0];$ 44       F4298 $p_sam > s1=a[1];$ 45       F42AE $p_sam > s1=a[1];$ 46       F42C6 $p_sam > s3=a[3];$ 47       F42DE $p_sam > s4=a[4];$ 48       F42F6 $p_sam > s5=a[5];$ 49       F430E $p_sam > s6=a[6];$ 50       F4326 $p_sam > s8=a[8];$ 52       F4356 $p_sam > s9=a[9];$ 53       F4374       )         55 $sam > sam = NULL;$ 56 $sam > sam = null;$		F427E			change (a) ;
44 $F4298$ $p_sam->s1=a[1];$ 45 $F42AE$ $p_sam->s2=a[2];$ 46 $F42C6$ $p_sam->s3=a[3];$ 47 $F42DE$ $p_sam->s4=a[4];$ 48 $F42F6$ $p_sam->s5=a[5];$ 49 $F430E$ $p_sam->s6=a[6];$ 50 $F4326$ $p_sam->s8=a[8];$ 51 $F4356$ $p_sam->s9=a[9];$ 53 $F436E$ $p_sam = NULL;$ 54 $F4374$ )         55 $F4376$ (					
45     F42AE     p_sam->s2=a[2];       46     F42C6     p_sam->s3=a[3];       47     F42DE     p_sam->s4=a[4];       48     F42F6     p_sam->s5=a[5];       49     F430E     p_sam->s6=a[6];       50     F4326     p_sam->s8=a[8];       52     F4356     p_sam->s9=a[9];       53     F4376     p_sam       56     void abort(void)       57     F4376     (					
46     F42C6     p_sam->s3=a[3];       47     F42DE     p_sam->s4=a[4];       48     F42F6     p_sam->s5=a[5];       49     F430E     p_sam->s6=a[6];       50     F4326     p_sam->s7=a[7];       51     F4356     p_sam->s9=a[8];       52     F4356     p_sam = NULL;       54     F4374     >       55     void abort (void)       57     F4376					
47       F42DE       p_sam->s4=a[4];         48       F42F6       p_sam->s5=a[5];         49       F430E       p_sam->s6=a[6];         50       F4326       p_sam->s7=a[7];         51       F4356       p_sam->s9=a[8];         52       F436E       p_sam = NULL;         54       F4374       >         56       void abort(void)       57         57       F4376       (					
48     F42F6     p_sam->s5=a[5];       49     F430E     p_sam->s6=a[6];       50     F4326     p_sam->s7=a[7];       51     F433E     p_sam->s8=a[8];       52     F4356     p_sam->s9=a[9];       53     F4374     )       55     56     void abort (void)       57     F4376     (					
49     F430E     p_sam->s6=a[6];       50     F4326     p_sam->s7=a[7];       51     F433E     p_sam->s8=a[8];       52     F4356     p_sam->s9=a[9];       53     F436E     p_sam = NULL;       54     F4374     >       55     56     void abort (void)       57     F4376     (					
50     F4326     p_sam->s7=a[7];       51     F433E     p_sam->s8=a[8];       52     F4356     p_sam->s9=a[9];       53     F436E     p_sam = NULL;       54     F4374     >       55     void abort(void)       57     F4376					
51     F433E     p_sam->s8=a[8];       52     F4356     p_sam->s9=a[9];       53     F436E     p_sam = NULL;       54     F4374     )       55     void abort(void)       57     F4376     (					
52     F4356     p_sam->s9=a[9];       53     F436E     p_sam = NULL;       54     F4374     )       55     void abort(void)       57     F4376       58					
53     F436E     p_sam = NULL;       54     F4374     )       55     56     void abort (void)       57     F4376     (					
54     F4374     )       55     56     void abort (void)       57     F4376     (       58     6     6					
55 56 57 F4376 ( 58					
56         void abort (void)           57         F4376         (           58          (		14374			3
57 F4376 ( 58					moid short (moid)
58		F4276	-		
		14376			
	58 59	F4376			)
	60				



However, if the C0 coverage of the function tutorial is verified in assembly units, you'll find that part of the function was unexecuted.

Line	C., .	A	S., Disass	Obj code	Label	Mixed
23						
24						void tutorial (void)
25						C
			F4224	7CF2OA	_tutorial	ENTER #OAH
26					New C	long j;
27						int i;
28						struct Sample far *p_sam;
29						
30						p_sam= &st
			F4227	75CBFA3C04		MOV.W:G #043CH,-6H[FB]
			F422C	75CBFC0000		MOV.W:G #0000H,-4H[FB]
31	_					init(p_sam);
			F4231	754BFC		PUSH.W:G -4H[FB]
			F4234	754BFA		PUSH.W:G -6H[FB]
			F4237	FD14400F 7DB4		JSR.A init F4014H
32	_		F423B	7064		ADD.B:Q #4H,SP
33						<pre>for( i=0; i&lt;10; i++ ){</pre>
55	-		F423D	D90BFE		MOV.W:Q #OH,-2H[FB]
			F4240	778BFE0A00		CMP.U:G #000AH,-2H[FB]
			F4245	7DCA2F		JGE F4276H
34						j = rand();
			F4248	FD78430F		JSR.A rand F4378H
			F424C	7CF3		EXTS.W RO
			F424E	732BF8		MOV.W:G R2,-8H[FB]
			F4251	730BF6		MOV.W:G RO,-AH[FB]
35						if(j < 0) (
			F4254	7EBBCF		BTST:G 7,-7H[FB]
			F4257	6AOA		JEQ F4262H
36						j = -j; NOT.W:G -8H[FB] Executed up until I
			F4259	757BF8		
			F425C	755BF6		NEG.W -AH[FB] in assembly units
	-	Χ	F425F	77EBF8		ADCF.W -8H[FB]
37						}
38	-		F4262	73B4FE		a[i] = j; MOV.W:G -2H[FB],AO
			F4265	FQ14		MOV.W:G -2H[FB],AO SHL W #2H AO
			F4267	73BCF61404		MOV.W:G -AH[FB],0414H[A0]
			F426C	73BCF81604		MOV.U:G -8H[FB],0416H[A0]
			F4271	C91BFE		ADD.U:Q #1H, -2H[FB]
			F4274	FECB		JMP.B F4240H

The reason why such a phenomenon occurs is because a prefetch is involved. Instructions are loaded into the cache beforehand no matter whether the next data is required. Once prefetched, when in assembly units, instructions are executed after the JEQ instruction up until the next instruction. When this is verified in C source units, instructions are assumed to have been executed up until "j = -j," so that it appears on the surface that all instructions were executed.

If you want to verify C0 coverage at the source level, we recommend verifying the result in mixed mode or disassembly mode.



#### 6. Related Documents

The PC7501 emulator and the HEW have numerous other convenient facilities and features not discussed in this document. Detailed specifications, 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.

[PC7501 emulator related documents]

- M16C R8C PC7501 Emulator Debugger User's Manual (for the M16C/60, M16C/30, M16C/Tiny and R8C/Tiny series)
- M16C R8C PC7501 Emulator Debugger Release Notes (for the M16C/60, M16C/30, M16C/Tiny and R8C/Tiny series)
- PC7501 Setup Guide (M16C Family Emulator)
- PC7501 User's Manual (M16C Family Emulator)

[High-performance Embedded Workshop related documents]

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

[CPU related documents]

- M16C/60 Series Hardware Manual
- M16C/60, M16C/20, M16C/Tiny Series Software Manual

[M16C/60 series C compiler related documents]

- M3T-NC30WA C Compiler Guidebook (C compiler package for the R8C/Tiny, M16C/60, M16C/30, M16C/20, M16C/10 and M16C/Tiny series)
- M3T-NC30WA Assembler User's Manual (C compiler package for the R8C/Tiny, M16C/60, M16C/30, M16C/20, M16C/10 and M16C/Tiny series)

To see more information on the PC7501 emulator, please visit the Renesas websites given below.

Japan site:	http://japan.renesas.com/pc7501
Global site:	http://www.renesas.com/pc7501



### Home Page and Where to Contact for Support

Renesas Technology home page <u>http://www.renesas.com/</u>

#### Where to contact

http://www.renesas.com/inquiry

#### **Revision Record**

		Contents of revision					
Rev.	Issue date	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 life

Renesas 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.

© 2009. Renesas Technology Corp., All rights reserved.