

Thank you very much for using the RI600V4 Real-time OS for RX Family.

This document describes cautions and restrictions for using this product. Please read this document carefully before using the product.

The main changes in this version are as follows. For details, refer to 6.4 Changes in RI600V4 V1.06.01.

- Fixed a memory corruption bug in RI600V4 V1.06.00

## Contents

1.	Packaged Tools .....	4
2.	User's Manual .....	5
2.1.	Errata on User's Manuals .....	5
3.	Target Devices .....	6
4.	Operating Environment .....	7
4.1.	Hardware Environment .....	7
4.2.	Software Environment .....	7
4.3.	Supported Tools .....	7
4.4.	Caution for Realtime OS Task Analyzer .....	8
5.	Installation Notes .....	9
5.1.	Common in CS+ and e <sup>2</sup> studio environments .....	9
5.1.1.	Cautions for Installation .....	9
5.1.2.	Cautions for Uninstallation .....	10
5.2.	In CS+ environment .....	10
5.2.1.	Cautions for Installation .....	10
5.2.2.	Cautions for Uninstallation .....	11
5.3.	In e <sup>2</sup> studio environment .....	11
5.3.1.	Cautions for Installation .....	11
5.3.2.	Cautions for Uninstallation .....	11
6.	Changes from previous released versions .....	12

6.1.	Changes in RI600V4 V1.04.00 .....	12
6.1.1.	Kernel .....	12
6.1.2.	Configurator .....	12
6.1.3.	Realtime OS Build Tool Plug-in .....	12
6.1.4.	Realtime OS Resource Information Displaying Plug-in .....	13
6.1.5.	Realtime OS Task Analyzer Plug-in .....	13
6.1.6.	Sample programs for CS+ .....	13
6.1.7.	System down routine .....	13
6.2.	Changes in RI600V4 V1.05.00 .....	14
6.2.1.	Kernel .....	14
6.2.2.	Sample programs for CS+ .....	14
6.3.	Changes in RI600V4 V1.06.00 .....	15
6.3.1.	Kernel .....	15
6.3.2.	Configurator .....	16
6.3.3.	GUI Configurator .....	18
6.3.4.	Realtime OS Build Tool Plug-in .....	19
6.3.5.	Realtime OS Analysis Control Plug-in .....	19
6.3.6.	Realtime OS Resource Information Displaying Plug-in .....	20
6.3.7.	Realtime OS Task Analyzer Plug-in .....	20
6.3.8.	Sample programs for CS+ .....	21
6.4.	Changes in RI600V4 V1.06.01 .....	23
6.4.1.	Kernel .....	23
7.	About the latest plug-ins .....	24
8.	Cautions .....	25
8.1.	Common cautions for CS+ and e <sup>2</sup> studio .....	25
8.1.1.	Distinction of Version .....	25
8.1.2.	Shift from a Previous Version .....	25
8.1.3.	GUI Configurator .....	26
8.1.4.	Timer Template File .....	27
8.1.5.	How to Build Kernel Source Code .....	29
8.1.6.	Stack Consumption .....	30
8.1.7.	Caution when using RX610 Group .....	35
8.1.8.	Caution when using global optimization of compile option .....	35
8.1.9.	Caution about initializing OS data .....	36
8.1.10.	Caution about building application .....	36
8.2.	Cautions for CS+ .....	37
8.2.1.	Enable Plug-ins .....	37
8.2.2.	Create a CS+ Project .....	38

---

8.2.3.	Sample Programs .....	40
8.2.4.	Cautions for Realtime OS Resource Information Panel .....	41
8.2.5.	Cautions for Realtime OS Task Analyzer .....	42
8.3.	Cautions for e <sup>2</sup> studio .....	46
9.	Restrictions .....	47
9.1.	Common restrictions of CS+ and e <sup>2</sup> studio .....	47
9.1.1.	Coding.....	47
9.1.2.	Realtime OS Resource Information Displaying Plug-in .....	47
9.1.3.	Realtime OS Task Analyzer Plug-in .....	48
9.2.	Restrictions of CS+.....	49
9.2.1.	Realtime OS Build Tool Plug-in .....	49
9.3.	Restrictions of e <sup>2</sup> studio.....	50
9.3.1.	Realtime OS Task Analyzer Plug-in .....	50
9.3.2.	Realtime OS Resource Information Displaying Plug-in .....	50
	Revision History .....	51

# 1. Packaged Tools

Depending on the model name, RI600V4 has different forms of contract and offer as follows.

- Trial

Product Name	Agreement Type
RTRRX0000TR01ERRZZ	Trial License, Limited 1 host

The provided tools are “Real-Time OS RI600V4 Kernel Object for trial (with restriction)” and “Command-line Configurator “cfg600”. These tools can be downloaded from a website of the Renesas Electronics.

And if you use the plug-in for “CS+ for CC” or e<sup>2</sup> studio, please download from a website of the Renesas Electronics.

- Evaluation License, Mass-production License

Product Name	Agreement Type	Contents
R0R5RX00TCW011	Evaluation License, Limited 1 host, No source code	A
R0R5RX00TCW01A	Evaluation License, Unlimited hosts, No source code	A
R0R5RX00TCW01K	Mass-production License, 3000 copies, No source code	A
R0R5RX00TCW01U	Mass-production License, Unlimited copies, No source code	A
R0R5RX00TCW01Z	Mass-production License, Unlimited copies, With source code	B

The following tools are provided.

Contents		Name	Version
B	A	Real-Time OS RI600V4 Kernel Object	V1.06.01
		Command-line Configurator “cfg600”	V1.06.00
		Plug-ins for CS+ for CC	
		Realtime OS Build Tool Plug-in (Common)	V3.02.01.01
		Realtime OS Build Tool Plug-in (RI600V4)	V3.01.00.00
		Realtime OS Analysis Control Plug-in (Common)	V3.00.00.03
		Realtime OS Analysis Control Plug-in (μITRON4)	V3.03.00.00
		Realtime OS Analysis Control Plug-in (RI600V4)	V3.01.00.00
		Realtime OS Resource Information Displaying Plug-in (Common)	V3.01.00.01
		Realtime OS Resource Information Displaying Plug-in (μITRON4)	V3.02.00.00
		Realtime OS Task Analyzer Plug-in (Common)	V3.00.02.01
		Realtime OS Task Analyzer Plug-in (Panel)	V3.01.00.08
		Realtime OS Task Analyzer Plug-in (RI600V4)	V3.02.00.00
		GUI Configurator “GUI600”	V1.06.00
		Real-Time OS RI600V4 Kernel Source Code	V1.06.01

## 2. User's Manual

The following user's manuals are included with this version. Please read these manuals together with this document.

Manual Name	Document Number
RI Series Real-Time Operating System User's Manual: Start	R20UT0751EJ0106
RI600V4 Real-Time Operating System User's Manual: Coding	R20UT0711EJ0104
RI600V4 Real-Time Operating System User's Manual: Debug	R20UT0775EJ0101
RI600V4 Real-Time Operating System User's Manual: Analysis	R20UT2185EJ0101
RI Series Real-Time Operating System User's Manual: Message	R20UT0756EJ0105

These PDF files are provided by this package. You can read them using the Windows Start Menu after installing this package.

These PDF files are also provided by Renesas Electronics Home page. If you use the Trial License, please download from a website of the Renesas Electronics.

### 2.1. Errata on User's Manuals

Information of errata on user's manuals is as follows.

Manual Name	Document Number	Page	wrong	right
RI Series Real-Time Operating System User's Manual: Start	R20UT0751EJ0106	Page 31 Table 1-1	EZ Emulator supports RI600V4.	EZ Emulator doesn't support RI600V4.
RI Series Real-Time Operating System User's Manual: Start	R20UT0751EJ0106	Page 32 Table 1-3	EZ Emulator supports RI600V4.	EZ Emulator doesn't support RI600V4.

### 3. Target Devices

The following devices are supported by the product.

- RX700 Series MCU
- RX600 Series MCU
- RX200 Series MCU
- RX100 Series MCU

## 4. Operating Environment

Below is described the operating environment for using the product.

### 4.1. Hardware Environment

- Processor: At least 1GHz (supported for hyper threading/multicore CPU)
- Main memory: 2 GB or more recommended
- Display: Resolution at least 1,024 x 768 dots; at least 65,536 colors

### 4.2. Software Environment

The following software environments are supported.

- Windows 8.1 (64bit) : CS+, e<sup>2</sup> studio
- Windows 10 (64bit) : CS+, e<sup>2</sup> studio
- Microsoft .NET Framework 4.6.2
- Microsoft Visual C++ 2015 Update 3 runtime libraries (x86)

Remark: or any of these, we recommend having the latest service pack installed.

### 4.3. Supported Tools

The following tools are supported.

Tool Name	Manufacturer	Version
Integrated development environment CS+ for CC	Renesas Electronics	V8.07.00 or later
Integrated development environment e <sup>2</sup> studio 64 bit	Renesas Electronics	2022-01 or later
Integrated development environment e <sup>2</sup> studio 32 bit	Renesas Electronics	V7.8.0
C/C++ Compiler CC-RX	Renesas Electronics	V1.02.01 or later (for RXv1) <sup>(Note)</sup> V2.04.01 or later (for RXv2) <sup>(Note)</sup> V3.01.00 or later (for RXv3) <sup>(Note)</sup>

Note: When using CS+, the kernel library is automatically selected according to the CC-RX version.  
For detail, refer to 6.3.1(1)The RXv3 kernel library.

## 4.4. Caution for Realtime OS Task Analyzer

To use Realtime OS Task Analyzer in “Taking in trace chart by hardware trace mode”, one of the following debug tool is required.

- Simulator
- Emulator with trace function

In CS+ for CC, Emulator which can be set as “Yes” to [Output timestamp] has trace function.

Please set “Data access” to [Trace data type].

Note, [Output timestamp] and [Trace data type] are in [Trace] category on [Debug Tool Settings] tab of [Property] panel for using debug tool.

When E1 emulator is used on RX100 Series, [Output timestamp] is fixed to “No”, so Realtime OS Task Analyzer cannot be used.



## 5. Installation Notes

This section provides cautions for installation and uninstallation.

### 5.1. Common in CS+ and e<sup>2</sup> studio environments

#### 5.1.1. Cautions for Installation

##### 5.1.1.1. Caution for administrator privileges

Windows® administrator privileges are required to install the software.

##### 5.1.1.2. Caution for execution environment

The .NET Framework and the Visual C++ runtime libraries are required to run the installer.

##### 5.1.1.3. Caution for network drives

The software cannot be installed from a network drive.

It also cannot be installed to a network drive.

##### 5.1.1.4. Caution for installation folder name

The available characters for specifying the installation folder are the same as for Windows®.

The 11 characters / \* : < > ? | " \ ; , cannot be used. Folder names also cannot start or end with a space.

Specify folders as absolute paths. Do not use relative paths.

Use the backslash character (\) as the path separator for the installation folder. Do not use the forward slash (/).

##### 5.1.1.5. Caution for modifying and repairing functions

To modify or repair the function of a tool that has already been installed, have the tool's installer package on hand, and run the installation program. The program maintenance program will start; select Modify or Repair.

Uninstall or change a program dialog boxes will cause an error.

##### 5.1.1.6. Caution for version of installed tools

If the newer version tool is already installed, the older version tool may not be installed.

##### 5.1.1.7. Caution for starting installer

If the installer is started on a non-Japanese version of Windows®, then if the path contains multi-byte characters it will cause an error, and the installer will not start.

##### 5.1.1.8. Caution for migration from a trial to a product version

Manually uninstall a trial in advance. Then install a product version.

If uninstallation is not performed, there is no bad effect immediately, but some files may not be deleted from the installation folder or the Windows® start menu when uninstalling the migrated product version.

## 5.1.2. Cautions for Uninstallation

### 5.1.2.1. Caution for administrator privileges

Windows® administrator privileges are required to uninstall the software.

### 5.1.2.2. Caution for uninstallation folder name

Depending on the order in which tools are uninstalled, the folders may not be completely deleted. If this happens, remove any remaining folders via Explorer or the like.

### 5.1.2.3. Caution for adding/repairing via other than the installer

If you added or modified files to the folders in which tools and manuals were installed using other means than the installers, they cannot be deleted during uninstallation.

## 5.2. In CS+ environment

### 5.2.1. Cautions for Installation

#### 5.2.1.1. Caution for required files after installation

The following folder is created after installation. Do not delete it, because it contains files that are necessary for the tools to run.

- If Windows® is 64bit and the installation drive is C:  
C:\Program Files (x86)\Common Files\Renesas Electronics CubeSuite+

#### 5.2.1.2. Enable Plug-ins

Plug-ins of this product may be disabled immediately after installation of this product. Please enable Plug-ins of this product. For details, refer to "8.2.1 Enable Plug-ins".

## 5.2.2. Cautions for Uninstallation

### 5.2.2.1. Key Word for Uninstallation

There are two ways to uninstall this product.

- Use the integrated uninstaller (uninstalls CS+ for CC)
- Use separate uninstaller (uninstalls this product only)

To use the separate uninstaller, select the following from the Control Panel:

- Programs and Features

After the applet appears, delete the followings.

- CS+ Realtime OS Common Plugins
- CS+ Realtime OS RI600V4 Plugins
- CS+ Realtime OS RI600V4 Object Release (In case of no source code, mass-production / evaluation license)
- CS+ Realtime OS RI600V4 Source Release (In case of with source code, mass-production license)
- CS+ Realtime OS RI600V4 Trial (In case of trial license)

## 5.3. In e<sup>2</sup> studio environment

### 5.3.1. Cautions for Installation

No caution.

### 5.3.2. Cautions for Uninstallation

#### 5.3.2.1. Key Word for Uninstallation

Select the following from the Control Panel:

- Programs and Features

After the applet appears, delete from the following.

- CS+ Realtime OS RI600V4 Object Release (In case of no source code, mass-production / evaluation license)
- CS+ Realtime OS RI600V4 Source Release (In case of with source code, mass-production license)
- CS+ Realtime OS RI600V4 Trial (In case of trial license)

## 6. Changes from previous released versions

This section provides changes in each release version of this product.

### 6.1. Changes in RI600V4 V1.04.00

#### 6.1.1. Kernel

(1) Change of “the service call” call method

For supporting to the “e<sup>2</sup> studio” environment, the “Table Generation Utility” is abolished, and how to call a service call is changed into the function call from the table jump. But how to use the service call is not change.

- It is necessary to change build environment. See “8.1.10 Caution about building application”.
- Stack Consumption of service call is changed. See “8.1.6 Stack Consumption”.

(2) The kernel version information

The version change is as follows.

Item	Before	After
TKERNEL_PRVER, T_RVER prver (the return value of “ref_ver” and “iref_ver”)	0x130	0x140

#### 6.1.2. Configurator

(1) The following fault is corrected

In the “kernel\_id.h”, the turn of description which is the following #pragma-description and the prototype declaration of the function is reversed.

- #pragma task
- #pragma cychandler
- #pragma almhandler

#### 6.1.3. Realtime OS Build Tool Plug-in

(1) The “CS+ for CC” tools are supported

The “CS+ for CC” tools are supported. In addition, this plug-in of this version does not operate on the “CubeSuite+”.

(2) The help can be opened from [Realtime OS] tab and [System Configuration File Related Information] tab

#### 6.1.4. Realtime OS Resource Information Displaying Plug-in

- (1) The “CS+ for CC” tools are supported  
The “CS+ for CC” tools are supported. In addition, this plug-in of this version does not operate on the “CubeSuite+”.
- (2) The waiting factor which are showed by the “ID” are changed to “name”  
The waiting factor which are showed by the “ID” are changed to “name”. It became intelligible the waiting factor.
- (3) It became intelligible the tabs  
The tab selection area is divided into two columns, and adds the icon to each tab.
- (4) A part of messages is improved  
A part of messages, for example error message, is improved.
- (5) The following restriction is canceled.  
The resource information panel does not get focus even if a display menu or a display button on toolbar is selected.

#### 6.1.5. Realtime OS Task Analyzer Plug-in

- (1) The “CS+ for CC” tools are supported  
The “CS+ for CC” tools are supported. In addition, this plug-in of this version does not operate on the “CubeSuite+”.
- (2) The following restriction is canceled.  
In some cases, dragging and dropping of objects such as the names of tasks and interrupt handlers does not work. This problem can be solved by sorting the objects (in any way), after which dragging and dropping will work again.

#### 6.1.6. Sample programs for CS+

- (1) Add new sample programs using FIT (Firmware Integration Technology) modules.  
Add new sample programs using FIT modules. For details, refer to “10.1 Sample programs using Firmware Integration Technology”.

#### 6.1.7. System down routine

- (1) Error no longer occurred in system down routine.  
In the system down routine `_RI_sys_dwn__` on RI600V4 V1.04.00 or higher, the error of type=-3 (Unsupported service call) is no longer occurred by changing internal process.

## 6.2. Changes in RI600V4 V1.05.00

The following changes are applied on only CS+, not e<sup>2</sup> studio.

### 6.2.1. Kernel

- (1) The RXv3 architecture is Supported.

To support the RXv3 architecture, we have changed that the RXv2 architecture's library is linked when using the RXv3 architecture. The RXv3 architecture is compatible with the RXv2 architecture.

It should be noted that Table 2-1 Kernel libraries in 2.6.3. Kernel library of "RI600V4 Real-Time Operating System User's Manual: Coding" (R20UT0711EJ0104) shall be replaced with the below table.

	Folder	Compiler version corresponding to the library	Corresponding CPU core	File name	Description
1	<ri_root>\library\rxv1	V1.02.01 or later	- RXv1 architecture	ri600lit.lib	For little endian
				ri600big.lib	For big endian
2	<ri_root>\library\rxv2	V2.01.00 or later	- RXv1 architecture	ri600lit.lib	For little endian
			- RXv2 architecture - <b>RXv3 architecture</b>	ri600big.lib	For big endian

- (2) The kernel version information

The version change is as follows.

Item	Before	After
TKERNEL_PRVER, T_RVER prver (the return value of "ref_ver" and "iref_ver")	0x140	0x150

### 6.2.2. Sample programs for CS+

- (1) Add new sample programs for RX66T

Because of supporting the RXv3 architecture, we have added new sample programs for CS+ for RX66T.

### 6.3. Changes in RI600V4 V1.06.00

The following changes are applied on both CS+ and e<sup>2</sup> studio.

#### 6.3.1. Kernel

(1) The RXv3 kernel library

We add the RXv3 kernel library when using the RXv3 architecture.

It should be noted that Table 2-1 Kernel libraries in 2.6.3. Kernel library of “RI600V4 Real-Time Operating System User’s Manual: Coding” (R20UT0711EJ0104) shall be replaced with the below table.

	Folder	Compiler version corresponding to the library	Corresponding CPU core	File name	Description
1	<ri_root>\library\rxv1	V1.02.01 or later	- RXv1 architecture	ri600lit.lib	For little endian
				ri600big.lib	For big endian
2	<ri_root>\library\rxv2	V2.01.00 or later	- RXv1 architecture - RXv2 architecture	ri600lit.lib	For little endian
				ri600big.lib	For big endian
3	<ri_root>\library\rxv3	V3.01.00 or later	- RXv1 architecture - RXv2 architecture - RXv3 architecture	ri600lit.lib	For little endian
				ri600lit_notrace.lib	For little endian (mode: Not tracing)
				ri600big.lib	For big endian
				ri600big_notrace.lib	For big endian (mode: Not tracing)

(2) The performance improvement in case of not using the Task Analyzer

You can get the application with higher performance if you build the source code with the CC-RX V3.00.00 or later and setting to “Not tracing” mode which is provide by the Realtime OS Task Analyzer Plug-in, compared to application built with other tracing modes.

In this case, a kernel library incorporated into the higher performance application is “ri600big\_notrace.lib” or “ri600lit\_notrace.lib”, is called as tracing-disabled kernel library. Their libraries are omitted tracing process from the general kernel library.

(3) Register Bank Save Function is supported

We support register bank save function which is optional function of the RXv3 architecture. This function enable to the fast collective saving and restoring of register values by using dedicated save register banks. RI600V4 provides this function for tasks, interrupt handlers, and fixed interrupt handlers. Note restrictions of using this function, please check the [“Usage of Register Bank Save Function”](#) entry.

(4) The kernel version information

The version change is as follows.

Item	Before	After
TKERNEL_PRVER, T_RVER prver (the return value of “ref_ver” and “iref_ver”)	0x150	0x160

### 6.3.2. Configurator

(1) Register Bank Save Function is supported

Because of supporting register bank save function, we added and changed configuration definition in the configurator.

In case of CFG600 V1.06.00 (included in RI600V4 V1.06.00) or later and configuring by a configuration file which doesn't have the definitions in the table below, CFG600 applies the values indicated in “When omitting”.

Definition	Description	Definition format	When omitting	Definition range
system.max_regbank	Define maximum number of save register banks. (Maximum number is depend on target device.)	Numeric value	0 *1	0 ~ 256
task.regbank	Define whether to use register bank save function on a task. The bank number is assigned in an order of task ID.	Symbol	OFF *1	ON or OFF
interrupt_vector.pragma_switch	Define switches to pass to “#prgma interrupt” directive. When you specify “REGBANK” keyword, an interrupt handler is passed “bank=xxx”, xxx means a bank number, to “#prgma interrupt” directive and uses register bank save function. A bank number is assigned in an order to be specified “REGBANK” keyword.	Symbol	No switches are passed *1	E,F,S,H,NH, ACC,NOACC, REGBANK
interrupt_fvector.pragma_switch	As above.	Symbol	No switches are passed *1	S,ACC,H,NH, NOACC, REGBANK



In addition, the following table indicates the error / warning messages added newly.

Type	ID	Message	Condition	How to handle
Error	O1037	system.max_regbank should be less than 256. (Max.=0x100).	[system.max_regbank] is larger than 256 (0x100).	Change [system.max_regbank] to 256 (0x100) or less.
Error	O1038	Use register save bank is bigger than system.max_regbank.	The sum of save register banks used in [task], [interrupt_vector], [interrupt_fvector], and the time management function is larger than [system.max_regbank].	Change register bank save function of either [task], [interrupt_vector], or [interrupt_fvector] to unused or change the time management function to unused in order to make the sum of used- save register banks is [system.max_regbank] or less. *2
Warning	O5001	system.context is MIN. sysetem.max_regbank change 0.	[system.context] is specified in MIN and [sysetem.max_regbank] is not 0.	When [system.context] is specified in MIN, change [system.max_regbank] to 0. *3
Warning	O5002	interrupt_vector[xx] is kernel reserved. Change to another vector number.	The xx is specified in vector number of [interrupt_vector] reserved by the kernel (1 ~ 8).	Change vector number of [interrupt_vector] to something other than number reserved by the kernel (1 ~ 8).

\*1 : The O1003 warning is occurred when omitting the definition, but it's no problem you ignore this warning.

\*2 : When [system.max\_regbank] is 1 or more and [clock.timer] is other than [NOTIMER], the time management function also uses one save register bank.

Therefore, in this case, the sum of save register bank usages of [task], [interrupt\_vector] and [interrupt\_fvector] needs to be less than [system.max\_regbank].

\*3 : When ignoring the warning, the configurator internally applies to system.max\_regbank = 0.

### 6.3.3. GUI Configurator

(1) Register Bank Save Function is supported

Because of supporting register bank save function, we added items of GUI Configurator function as follows.

Note that GUI Configurator V1.06.00 can't read input files created by GUI Configurator V1.01.00.004 or earlier.

View	Item	Description
[System Definition] View	[Use register bank save function.] Checkbox	Select checkbox when using register bank save function throughout system, this activates following items in this table.
	[Max. Save register bank] Display box / Modification button	Maximum number of save register banks are displayed. /You can modify this number to pressing the button.
[Task] View	[Register bank save function] Column	[Use] or [Not use] keyword is displayed whether you use register bank save function by task.
[Creation of Task] Dialog	[Use] Checkbox	Select checkbox to use register bank save function by task.
	[Used] Display box	Total number you are currently using save register banks for tasks, interrupt handlers, fixed interrupt handlers, and timer management function is displayed.
	[Remained] Display box	Total number you can currently use save register banks is displayed.
[Interrupt] View	[Switch] Column	If you use register bank save function, [regbank] keyword is displayed by handler.
	[Used] Display box	Same as [Used] Display box on [Creation of Task] Dialog.
	[Remained] Display box	Same as [Remained] Display box on [Creation of Task] Dialog.
[Interrupt Handler Definition] Dialog	[regbank] Checkbox	Select checkbox to use register bank save function by interrupt handler.
	[Used] Display box	Total number you are currently using save register banks is displayed.
	[Remained] Display box	Total number you can currently

		use save register banks is displayed.
[Fixed Interrupt Handler Definition] Dialog	[regbank] Checkbox	Select checkbox to use register bank save function by fixed interrupt handler.
	[Used] Display box	Total number you are currently using save register banks is displayed.
	[Remained] Display box	Total number you can currently use save register banks is displayed.

(2) Reserved vectors Clearly displayed on [Interrupt] View

We changed that [Reserved] keyword is displayed on a [Address] column of vectors used by RI600V4 kernel. You cannot open a handler definition dialog of the reserved vectors and cannot set these information.

### 6.3.4. Realtime OS Build Tool Plug-in

(1) The kernel library for the RXv3 architecture is supported

We added build function for the kernel library for the RXv3 architecture.

(2) Tracing-Disabled Kernel Library is supported

We added build function for the tracing-disabled kernel library.

(3) Correct the error of loading a project file

We corrected the raising exception in case of that you load a project registered plural cfg files.

### 6.3.5. Realtime OS Analysis Control Plug-in

(1) Register Bank Save Function is supported

We supported processing when using the register bank save function.

### 6.3.6. Realtime OS Resource Information Displaying Plug-in

- (1) Register Bank Save Function is supported

Because of supporting register bank save function, we added following entries in Realtime OS Resource Information panel.

Tab	Entry	Description
[System]	[Max,Save register bank]	Maximum number of save register banks are displayed.
[Task]	[Register bank save function]	The usage status of register bank save function is displayed.

- (2) Correct the error display of task information on Realtime OS Resource Information panel

We correct the error which an incorrect value is displayed in [Current PC] and [Current task SP] on the [Task] tab of switching source when a task started initially is switched to another task. In case of an incorrect value, the "-" keyword is displayed on Realtime OS Resource Information panel.

### 6.3.7. Realtime OS Task Analyzer Plug-in

- (1) Tracing-Disabled Kernel Library is supported

This plug-in is supported operation when using tracing-disabled kernel library.

### 6.3.8. Sample programs for CS+

- (1) Add new sample programs for RX72T and RX72M with using Register Bank Save Function

Because of supporting register bank save function, we added new sample programs for CS+ for RX72T and RX72M with using this function.

Also, additional describes are coded in the [resetprg.c]: the definition of Option-Setting Memory (OFSM) and the Initialization function which must be called to use Arithmetic Unit for Trigonometric Functions (TFU).

The [resetprg.c] of sample programs for RX72T

```
void PowerON_Reset_PC(void)
{
    // Initialize clock
    IO_PRCR = PRCR_KEY|PRCR_PRC0;    // Cancel register-write-protection
    IO_SCKCR3 = 0x200;                // Clock source = Main clock(24MHz)
    IO_PRCR = PRCR_KEY;              // Set register-write-protection

    #ifdef _ROZ                        // Initialize FPSW
    #define _ROUND 0x00000001          // Let FPSW RMbits=01 (round to zero)
    #else
    #define _ROUND 0x00000000          // Let FPSW RMbits=00 (round to nearest)
    #endif
    #ifdef _DOFF
    #define _DENOM 0x00000100          // Let FPSW DNbit=1 (denormal as zero)
    #else
    #define _DENOM 0x00000000          // Let FPSW DNbit=0 (denormal as is)
    #endif

    set_fpsw(FPSW_init | _ROUND | _DENOM);

    #ifdef __TFU
    __init_tfu();                    // Use TFU
    #endif

    __INITSCT();

    (Omission)
}
```

- (2) Correct existing sample programs because of supporting Register Bank Save Function

We added [system.max\_regbank] and [task.regbank] definitions in [sample.cfg] of existing sample programs (for RX111, RX210, RX610, RX64M, and RX66T) which were contained in previous RI600V4. It's because of supporting register bank save function.

- (3) Add the definition of OFSM for sample programs for RX64M and RX 66T

We defined OFSM in the [resetprg.c] of sample programs for RX64M and RX 66T. This definition contributes not to display the warning about OFSM definition (W1811334) occurring when downloading sample programs to a emulator in RI600V4 V1.05.00 or earlier.

The [resetprg.c] of sample programs for RX66T

```
(Omission)

// Clock
#define IO_SCKCR3 (*(volatile UH __evenaccess *)0x00080026UL) /**< SCKCR3 Address */

////////////////////////////////////
// Option-Setting Memory (OFSM) for RX66T
////////////////////////////////////
#pragma address __SPCCreg=0x00120040 // SPCC register
const unsigned long __SPCCreg = 0xffffffff;

#pragma address __TMEFreg=0x00120048 // TMEF register
const unsigned long __TMEFreg = 0xffffffff;

#pragma address __OSISreg=0x00120050 // OSIC register (ID codes)
const unsigned long __OSISreg[4] = {
    0xffffffff,
    0xffffffff,
    0xffffffff,
    0xffffffff,
};

#pragma address __TMINreg=0x00120060 // TMINF register
const unsigned long __TMINreg = 0xffffffff;

#pragma address __MDEreg=0x00120064 // MDE register (Single Chip Mode)
#ifdef __BIG
    const unsigned long __MDEreg = 0xffffffff8; // big
#else
    const unsigned long __MDEreg = 0xffffffff; // little
#endif

#pragma address __OFS0reg=0x00120068 // OFS0 register
const unsigned long __OFS0reg = 0xffffffff;

#pragma address __OFS1reg=0x0012006c // OFS1 register
const unsigned long __OFS1reg = 0xffffffff;

#pragma address __RCPreg=0x0012007c // RCP register
const unsigned long __RCPreg = 0xffffffff;

#pragma section ResetPRG // output PowerON_Reset to PResetPRG section

////////////////////////////////////
// Boot processing
////////////////////////////////////
#pragma entry PowerON_Reset_PC

void PowerON_Reset_PC(void)
{
(Omission)
}

```

## 6.4. Changes in RI600V4 V1.06.01

The following changes are applied on both CS+ and e<sup>2</sup> studio.

### 6.4.1. Kernel

(1) Bug fix for memory corruption

We fixed the bug of memory corruption when using the RXv3 kernel libraries <sup>(Note)</sup> of V1.06.00. For detail of this bug, please refer to the following.

[\[Notes\] RI600V4 Real-Time OS for the RX Family](#) (R20TS0746JJ0100)

(2) The kernel version information

The version change is as follows.

Item	Before	After
TKERNEL_PRVER, T_RVER prver (the return value of "ref_ver" and "iref_ver")	0x160	0x161

Note: The libraries in the folder "<ri\_root>\library\rxv3"

## 7. About the latest plug-ins

We may update plug-ins without notice. Please update to the latest plug-ins by using CS+ update feature.



## 8. Cautions

### 8.1. Common cautions for CS+ and e<sup>2</sup> studio

#### 8.1.1. Distinction of Version

By referring to this variable, the version of the kernel is distinguishable.

```
const UW _RI600V4_VERSION = <Value>;
```

The version of the kernel is denoted by form of "X,YY,ZZ,aa". The bit31-24of \_RI600V4\_VERSION expresses "X", the bit 23-16 expresses "YY", the bit 15-8 expresses "ZZ", and the bit 7-0 expresses "aa".

The actual versions are as follows.

Kernel version (Product version)	_RI600V4_VERSION	Note
V1.01.00 (V1.01.00)	(Not defined)	The past version
V1.02.00 (V1.02.00)	(Not defined)	The past version
V1.02.01 (1.02.01)	(Not defined)	The past version
V1.02.02 (1.02.02)	(Not defined)	The past version
V1.03.00.03 (V1.03.00)	0x01030003	The past version
V1.04.00.00 (1.04.00)	0x01040000	The past version
V1.05.00.00 (1.05.00)	0x01050000	The past version
V1.06.00.00 (1.06.00)	0x01060000	The past version
V1.06.01.00 (1.06.01)	0x01060100	This version

#### 8.1.2. Shift from a Previous Version

When you shift from a previous version, please be sure to re-build after clean.

### 8.1.3. GUI Configurator

#### 8.1.3.1. Summary

The GUI configurator is a tool that permits the user to generate a system configuration file by entering various kernel configuration information from GUI screen. Using the GUI configurator, it is possible to build the kernel without the need for learning how to write a system configuration file.

Check online help for details on how to use the GUI configurator.

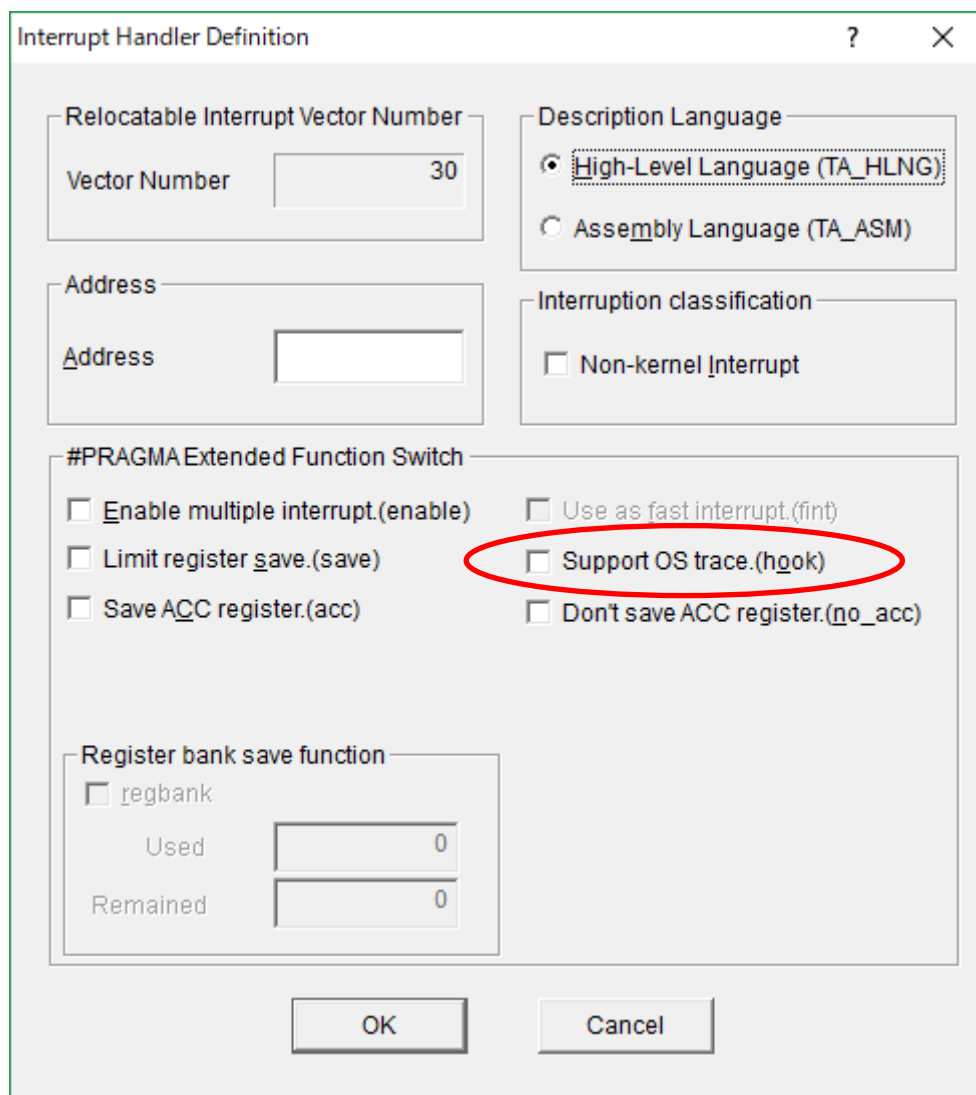
To start the GUI configurator, run "GUI Configurator" at the Windows Start Menu.

- The full path of the GUI configurator's [executable](#) file  
 "< installation folder >\bin600\Guiconfig\_RI600.exe

#### 8.1.3.2. [Interrupt Handler Definition] dialog box

The setting of the [Support OS trace (hook)] check box in the [Interrupt Handler Definition] dialog box is ignored.

figure 8-1 Definition of Interrupt handler



### 8.1.4. Timer Template File

The relation between timer template file provided by RI600V4 and corresponded MCUs is shown as follows.

The timer template file is specified to “clock.template” in the system configuration file.

Please check the latest information of the timer template file on the product website of RI600V4.

Table 8-1 Timer template file

Template File	Corresponded MCUs
rx610.tpl	RX600 Series RX610 Group
rx62t.tpl	RX600 Series RX62T Group
rx62n.tpl	RX600 Series RX62G Group RX600 Series RX62N Group RX600 Series RX621 Group
rx630.tpl	RX700 Series RX72T Group *1 RX700 Series RX71M Group *1 RX700 Series RX72N Group *1 RX700 Series RX72M Group *1 RX600 Series RX66T Group *1 RX600 Series RX65N Group *1 RX600 Series RX651 Group *1 RX600 Series RX64M Group *1 RX600 Series RX671 Group *1 RX600 Series RX66N Group *1 RX600 Series RX630 Group RX600 Series RX63N Group RX600 Series RX631 Group RX600 Series RX634 Group RX600 Series RX63T Group RX200 Series RX21A Group RX200 Series RX230 Group RX200 Series RX231 Group RX200 Series RX23T Group RX200 Series RX24T Group RX200 Series RX24U Group RX200 Series RX23W Group *1 RX200 Series RX23E-A Group *1
rx210.tpl	RX200 Series RX210 Group RX200 Series RX220 Group RX100 Series RX110 Group *1 RX100 Series RX111 Group *1 RX100 Series RX113 Group *1 RX100 Series RX130 Group *1

	RX100 Series RX13T Group *1
	RX100 Series RX140 Group *1

\*1 Don't specify "CMT2" and "CMT3" as "clock.timer" in a system configuration file.

It's because that the device doesn't have CMT2 and CMT3 or the device is disenable to CMT2 and CMT3 (software configurable interrupt) in default.

### 8.1.5. How to Build Kernel Source Code

Since the RI600V4 kernel is provided in the library form, it does not usually need to build the kernel. The source code is only attached to "Mass-production License, Unlimited copies, With source code" (R0R5RX00TCW01Z) which is one of agreement types.

The kernel source code is stored in "< installation folder >src600". To build the kernel, set current folder to this folder, and run "nmake.exe"<sup>1</sup> as follows. The libraries will be generated under "< installation folder >library".

- Command to generate libraries at "< installation folder >library\rxv1"

```
> nmake release_install(RET)
```

Note: The libraries attached to this product were built by using CC-RX V1.02.01.

- Command to generate libraries at "< installation folder >library\rxv2"

```
> nmake -f make_rxv2.mak release_install(RET)
```

Note: The libraries attached to this product were built by using CC-RX V2.01.00.

- Command to generate libraries at "< installation folder >library\rxv3"

```
> nmake -f make_rxv3.mak release_install(RET)
```

```
> nmake -f make_rxv3.mak release_notrace_install(RET)
```

Note: The libraries attached to this product (V1.06.01) were built by using CC-RX V3.01.00.

Please copy the installation folder to the writable folder if you don't have the write-access permission to the installation folder. After the build, copy the generated libraries to the "library\rxv1", "library\rxv2", or "library\rxv3" folder under the installation folder by the user who has write-access permission to the product installation folder.

---

<sup>1</sup> "nmake.exe" is a tool to build the project provided by Microsoft Corporation in United States. "nmake.exe" is included in Microsoft Visual Studio 2008 etc.

## 8.1.6. Stack Consumption

### 8.1.6.1. Stack consumption of base clock interrupt handler (*clocksz1*, *clocksz2*, *clocksz3*)

The value of *clocksz1*, *clocksz2* and *clocksz3* described in appendix D.4 of “RI600V4 Real-Time Operating System User's Manual: Coding” are as follows.

- *clocksz1*=136
- *clocksz2*=136
- *clocksz3*=204

### 8.1.6.2. Stack consumption of service calls (*svcsz*)

In the service call, the stack is used as follows.

(1) Called from the task context

The stack in the task context execution is a user stack. The service call is using following.

- (a) User stack (Former call stack)
- (b) System stack

(2) Called from the non-task context

The stack in the non-task context execution is a system stack. The service call is using following.

- (c) System stack (Former call stack)

The use size of former stack ((a), (c)) which the service call uses is displayed by Call Walker.

Moreover, to calculate consumption of the system stack described in appendix D.4 of “RI600V4 Real-Time Operating System User's Manual: Coding”, the size of (b) and (c) is needed. (Appendix D.4 has described as *svcsz*. ) The size of (a), (b) and (c) of each service call is shown as follows.

Table 8-2 Stack usage of service-call

	Service call	The use size of User stack (a)	The use size of System stack (b), (c)	Note
<b>Task Management Function</b>				
1	act_tsk	4	44	
2	iact_tsk	0	52	
3	can_act	4	44	
4	ican_act	0	48	
5	sta_tsk	4	44	
6	ista_tsk	0	48	
7	ext_tsk	0	60	ext_tsk is called at the return from the task entry function.
8	ter_tsk	4	116	
9	chg_pri	4	44	

	Service call	The use size of User stack(a)	The use size of System stack (b), (c)	Note
<b>Task Management Function</b>				
10	ichg_pri	0	60	
11	get_pri	4	44	
12	iget_pri	0	48	
13	ref_tsk	4	44	
14	iref_tsk	0	48	
15	ref_tst	4	44	
16	iref_tst	0	48	
<b>Task Dependent Synchronization Function</b>				
17	slp_tsk	4	44	
18	tslp_tsk	4	44	
19	wup_tsk	4	44	
20	iwup_tsk	0	52	
21	can_wup	4	44	
22	ican_wup	0	48	
23	rel_wai	4	112	
24	irel_wai	0	132	
25	sus_tsk	4	44	
26	isus_tsk	0	48	
27	rsm_tsk	4	44	
28	irms_tsk	0	48	
29	frsm_tsk	4	44	
30	ifrsn_tsk	0	48	
31	dly_tsk	4	44	
<b>Semaphore</b>				
32	sig_sem	4	44	
33	isig_sem	0	60	
34	wai_sem	4	44	
35	pol_sem	4	44	
36	ipol_sem	0	48	
37	twai_sem	4	44	
38	ref_sem	4	44	
39	iref_sem	0	48	

	Service call	The use size of User stack(a)	The use size of System stack (b), (c)	Note
<b>Eventflag</b>				
40	set_flg	4	48	
41	iset_flg	0	68	
42	clr_flg	4	44	
43	iclr_flg	0	48	
44	wai_flg	4	48	
45	pol_flg	4	44	
46	ipol_flg	0	48	
47	twai_flg	4	48	
48	ref_flg	4	44	
49	iref_flg	0	48	
<b>Data Queue</b>				
50	snd_dtq	4	44	
51	psnd_dtq	4	44	
52	ipsnd_dtq	0	60	
53	tsnd_dtq	4	44	
54	fsnd_dtq	4	44	
55	ifsnd_dtq	0	60	
56	rcv_dtq	4	44	
57	prcv_dtq	4	44	
58	iprcv_dtq	0	64	
59	trcv_dtq	4	44	
60	ref_dtq	4	44	
61	iref_dtq	0	48	
<b>Mailbox</b>				
62	snd_mbx	4	44	
63	isnd_mbx	0	60	
64	rcv_mbx	4	44	
65	prcv_mbx	4	44	
66	iprcv_mbx	0	48	
67	trcv_mbx	4	44	
68	ref_mbx	4	44	
69	iref_mbx	0	48	



	Service call	The use size of User stack(a)	The use size of System stack (b), (c)	Note
<b>Mutex</b>				
70	loc_mtx	4	44	
71	ploc_mtx	4	44	
72	tloc_mtx	4	44	
73	unl_mtx	4	52	
74	ref_mtx	4	44	
<b>Message Buffer</b>				
75	snd_mbf	4	44	
76	psnd_mbf	4	44	
77	ipsnd_mbf	0	64	
78	tsnd_mbf	4	44	
79	rcv_mbf	4	56	
80	prcv_mbf	4	56	
81	trcv_mbf	4	56	
82	ref_mbf	4	44	
83	iref_mbf	0	48	
<b>Fixed-sized Memory Pool</b>				
84	get_mpf	4	48	
85	pget_mpf	4	44	
86	ipget_mpf	0	48	
87	tget_mpf	4	48	
88	rel_mpf	20	44	
89	irel_mpf	0	64	
90	ref_mpf	4	44	
91	iref_mpf	0	48	
<b>Variable Size Memory Pool</b>				
92	get_mpl	28	88	
93	pget_mpl	4	104	
94	ipget_mpl	0	108	
95	tget_mpl	28	88	
96	rel_mpl	4	104	
97	ref_mpl	4	44	
98	iref_mpl	0	48	

	Service call	The use size of User stack(a)	The use size of System stack (b), (c)	Note
<b>Time Management Function</b>				
99	set_tim	4	44	
100	iset_tim	0	48	
101	get_tim	4	44	
102	iget_tim	0	48	
<b>Cyclic Handler</b>				
103	sta_cyc	4	44	
104	ista_cyc	0	48	
105	stp_cyc	4	44	
106	istp_cyc	0	48	
107	ref_cyc	4	44	
108	iref_cyc	0	48	
<b>Alarm Handler</b>				
109	sta_alm	4	44	
110	ista_alm	0	48	
111	stp_alm	4	44	
112	istp_alm	0	48	
113	ref_alm	4	44	
114	iref_alm	0	48	
<b>System State Management Function</b>				
115	rot_rdq	4	44	
116	irotd_rdq	0	48	
117	get_tid	4	44	
118	iget_tid	0	48	
119	loc_cpu	4	44	
120	iloc_cpu	0	48	
121	unl_cpu	4	44	
122	iunl_cpu	0	48	
123	dis_dsp	4	44	
124	ena_dsp	4	44	
125	sns_ctx	0	48	
126	sns_loc	0	48	
127	sns_dsp	0	48	
128	sns_dpn	0	48	
129	vsta_knl	4	40	After the system stack pointer is initialized, it uses it.
130	ivsta_knl	0	48	

	Service call	The use size of User stack(a)	The use size of System stack (b), (c)	Note
<b>System State Management Function</b>				
131	vsys_dwn	4	44	
132	ivsys_dwn	0	48	
<b>Interrupt Management Function</b>				
133	chg_ims	4	44	
134	ichg_ims	0	48	
135	get_ims	4	4	
136	iget_ims	4	4	
137	Kernel interrupt handler	0	52	When a kernel interrupt handler ends, 48 bytes of the system stack is consumed from just before generating of the interrupt.
<b>System Configuration Management Function</b>				
138	ref_ver	4	44	
139	iref_ver	0	48	
<b>Object Reset Function</b>				
140	vrst_dtq	4	44	
141	vrst_mbx	4	44	
142	vrst_mbf	4	44	
143	vrst_mpf	4	44	
144	vrst_mpl	4	72	

### 8.1.6.3. When the kernel library is built

Please note that the stack consumption might change when a version and/or an optional setting of the compiler are changed and the kernel library is built.

### 8.1.7. Caution when using RX610 Group

The value specified as follows should be less than 8 because the PSW.IPL is configured in 3-bit widths.

- Interrupt mask specified in chg\_ims and ichg\_ims
- "system.system\_IPL" in the system configuration file
- "clock.IPL" in the system configuration file

### 8.1.8. Caution when using global optimization of compile option

It is not able to specify global optimization (-ip\_optimize, -merge\_files, -whole\_program) to the program embedded RI600V4.

### 8.1.9. Caution about initializing OS data

In RI600V4 V1.04.00, the way to initialize OS data is changed. So, change the following settings to a user program.

- Linker option

Before: There is "-rom=DRI\_ROM=PRI\_RAM" option

After: There is not "-rom=DRI\_ROM=PRI\_RAM" option

- Initializing setting of B, R section (dbsct.c in sample program)

Before:

```
#pragma section C C$DSEC
extern const struct {
  _UBYTE *rom_s;
  _UBYTE *rom_e;
  _UBYTE *ram_s;
} _DTBL[] = {
  { __sectop("D"), __secend("D"), __sectop("R") },
  { __sectop("D_2"), __secend("D_2"), __sectop("R_2") },
  { __sectop("D_1"), __secend("D_1"), __sectop("R_1") },
  { __sectop("DRI_ROM"), __secend("DRI_ROM"), __sectop("RRI_RAM") }
};

#pragma section C C$BSEC
extern const struct {
  _UBYTE *b_s;
  _UBYTE *b_e;
} _BTBL[] = {
  { __sectop("B"), __secend("B") },
  { __sectop("B_2"), __secend("B_2") },
  { __sectop("B_1"), __secend("B_1") }
};
```

After:

```
#pragma section C C$DSEC
extern const struct {
  _UBYTE *rom_s;
  _UBYTE *rom_e;
  _UBYTE *ram_s;
} _DTBL[] = {
  { __sectop("D"), __secend("D"), __sectop("R") },
  { __sectop("D_2"), __secend("D_2"), __sectop("R_2") },
  { __sectop("D_1"), __secend("D_1"), __sectop("R_1") }
};

#pragma section C C$BSEC
extern const struct {
  _UBYTE *b_s;
  _UBYTE *b_e;
} _BTBL[] = {
  { __sectop("B"), __secend("B") },
  { __sectop("B_2"), __secend("B_2") },
  { __sectop("B_1"), __secend("B_1") },
  { __sectop("BRI_RAM"), __secend("BRI_RAM") }
};
```

### 8.1.10. Caution about building application

In RI600V4 V1.04.00, "mkritbl.exe" (table generation utility) is removed. Because the way to call service calls is changed from table jumping to function calling.

Please try the following procedure, if your application was built before RI600V4 V1.04.00. Because it is necessary to change "ritable.src" from output by "mkritbl.exe" to output by "cfg600.exe" (configurator).

- If you built applications as CubeSuite+ project:
  - Load and build project to CS+ V3.00.00
- If you built applications by another way (e.g. makefile):
  - Delete to start "mkritbl.exe".
  - Change linking "ritable.src" from output by "mkritbl.exe" to output by "cfg600.exe".

In RI600V4 V1.04.00 or later, "mrc file" is not needed, but it is outputted. Please ignore "mrc file". Because it does not affect behavior of application.

## 8.2. Cautions for CS+

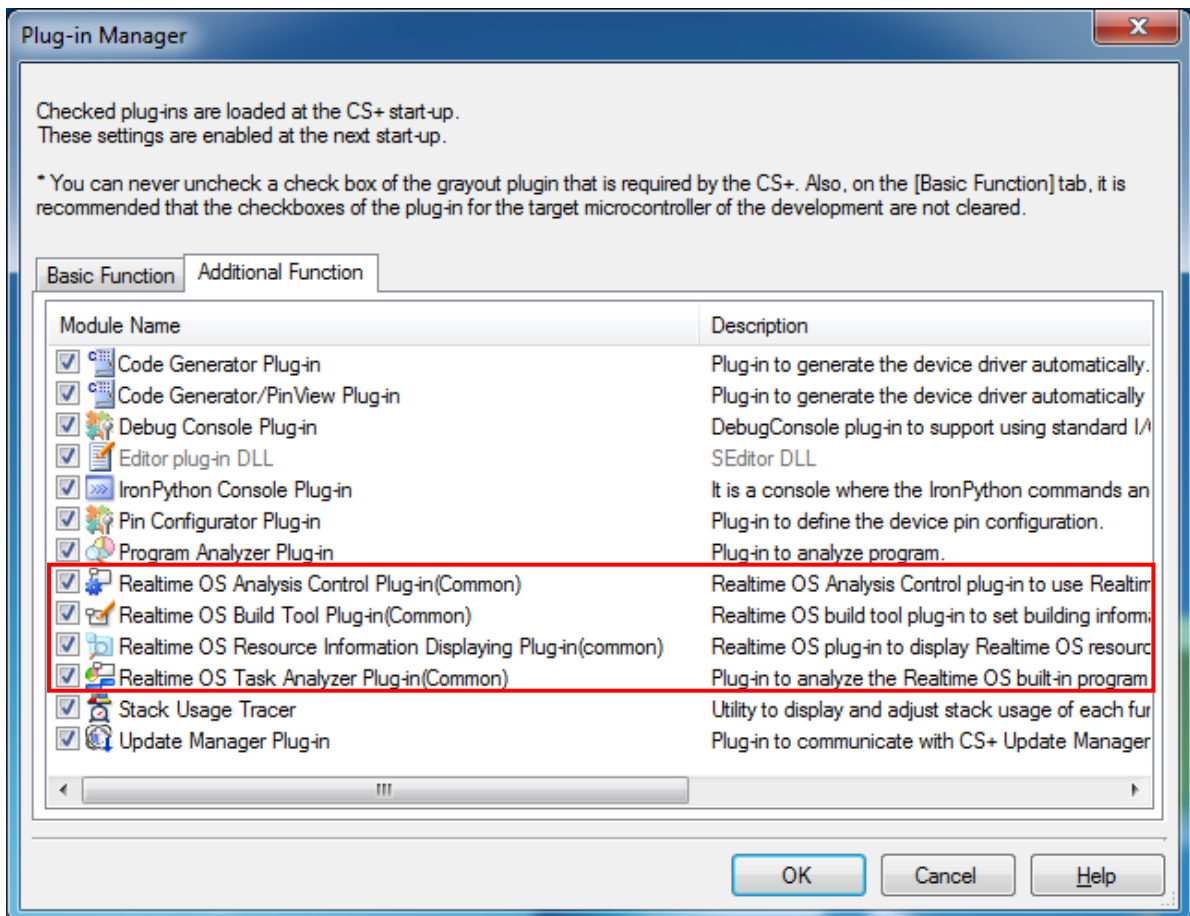
### 8.2.1. Enable Plug-ins

Plug-ins of this product may be disabled immediately after installation of this product. If plug-ins are disabled, the problem of being unable to build arises.

Please enable following Plug-ins by [Additional Function] tab in [Plug-in Manager] dialog box of the CS+ for CC

- Realtime OS Analysis Control Plug-in(Common)
- Realtime OS Build Tool Plug-in(Common)
- Realtime OS Resource Information Displaying Plug-in(common)
- Realtime OS Task Analyzer Plug-in(Common)

figure 8-2 Plug-in Manager



## 8.2.2. Create a CS+ Project

To create a project which uses this product, there are the following two methods.

- Divert the sample project attached to this product.
- Create a new project

### 8.2.2.1. Divert the sample project attached to this product

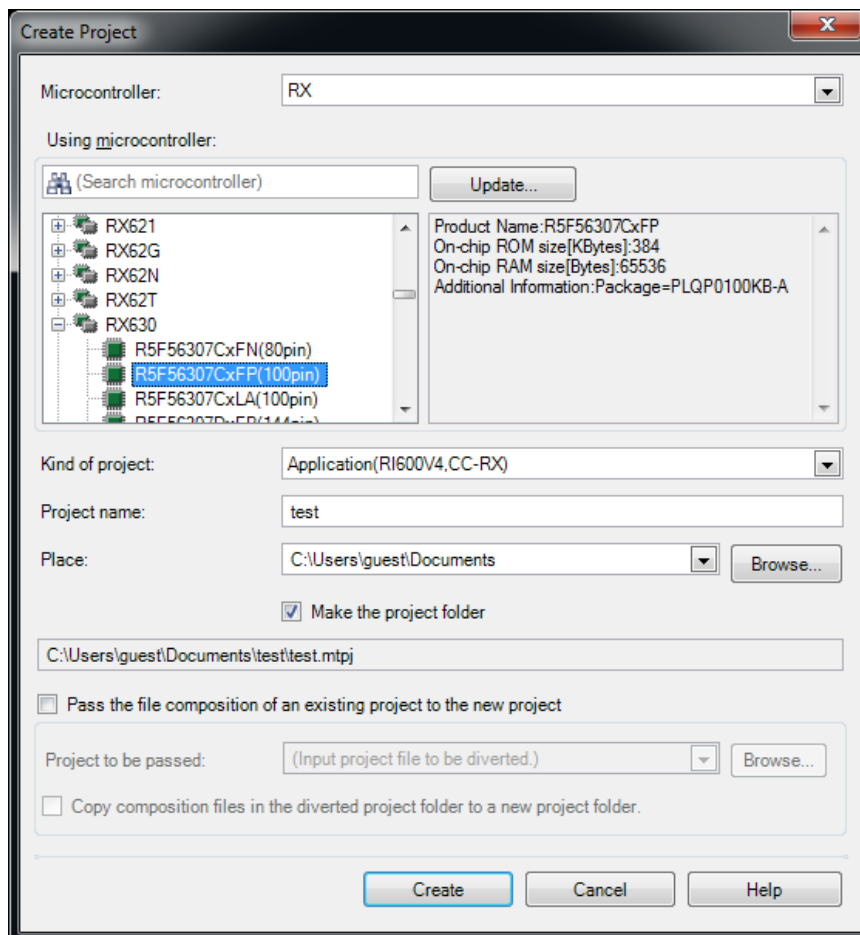
Select [RX] tab in [Open Sample Project] area of [Start] panel of the CS+, and choose the project named “RX???\_RI600V4”.

### 8.2.2.2. Create a new project

(1) Create a project

Press [Go] button in [Create New Project] area of [Start] panel of the CS+, then [Create Project] dialog box will be opened.

figure 8-3 Create Project



- [Microcontroller] : Select “RX”
- [Kind of project] : Select “Application(RI600V4,CC-RX)”

Press [Create] button, then a project will be generated.

## (2) Register files

No files are registered immediately after project creation. Please register the following files according to "CHAPTER 2 SYSTEM BUILDING" in "RI600V4 User's Manual: Coding".

- Processing programs, such as tasks and handlers (refer to section 2.2 in "RI600V4 User's Manual: Coding")
- System configuration file (refer to section 2.3 in "RI600V4 User's Manual: Coding")
- User-own coding module (refer to section 2.4 in "RI600V4 User's Manual: Coding")

## (3) Build options

Please set up suitable build options according to "2.5 Creating Load Module" and "2.6 Build Options" in "RI600V4 User's Manual: Coding".

### 8.2.3. Sample Programs

The provided sample programs are set up to use the Realtime OS Task Analyzer by “Taking in trace chart by hardware trace mode”.

When the trace mode is changed to “Taking in trace chart by software trace mode” or “Taking in long-statistics by software trace mode”, the following description should be added to the system configuration file. For details, refer to “15.3 User-Own Coding Module for Software Trace Mode” in “RI600V4 Real-Time Operating System User’s Manual : Coding”. In addition, these descriptions are commented out in the system configuration file at the time of shipment.

(1) “Taking in trace chart by software trace mode”

```
interrupt_vector[29]{           // CMT CH1
  os_int           = NO;
  entry_address    = _RIUSR_trcSW_interrupt();           // in trcSW_cmt.src
};
```

(2) Taking in long-statistics by software trace mode”

```
interrupt_vector[29]{           // CMT CH1
  os_int           = NO;
  entry_address    = _RIUSR_trcLONG_interrupt();         // in trcLONG_cmt.src
};
```



## 8.2.4. Cautions for Realtime OS Resource Information Panel

### 8.2.4.1. View after Real-Time OS is initialized

View the Realtime OS Resource Information Panel after the Real-Time OS has been initialized. Before the Real-Time OS has been initialized, the information in the Realtime OS Resource Information Panel is undefined.

### 8.2.4.2. Use programs with debug information generated

When using the Realtime OS Resource Information Panel, download a program for which debug information has been generated. Downloading a program without debug information and viewing it in the Realtime OS Resource Information Panel will cause an error.

To generate debug information, under Build Tool, under the Link Options properties, set "Generate debug information" to "Yes".

## 8.2.5. Cautions for Realtime OS Task Analyzer

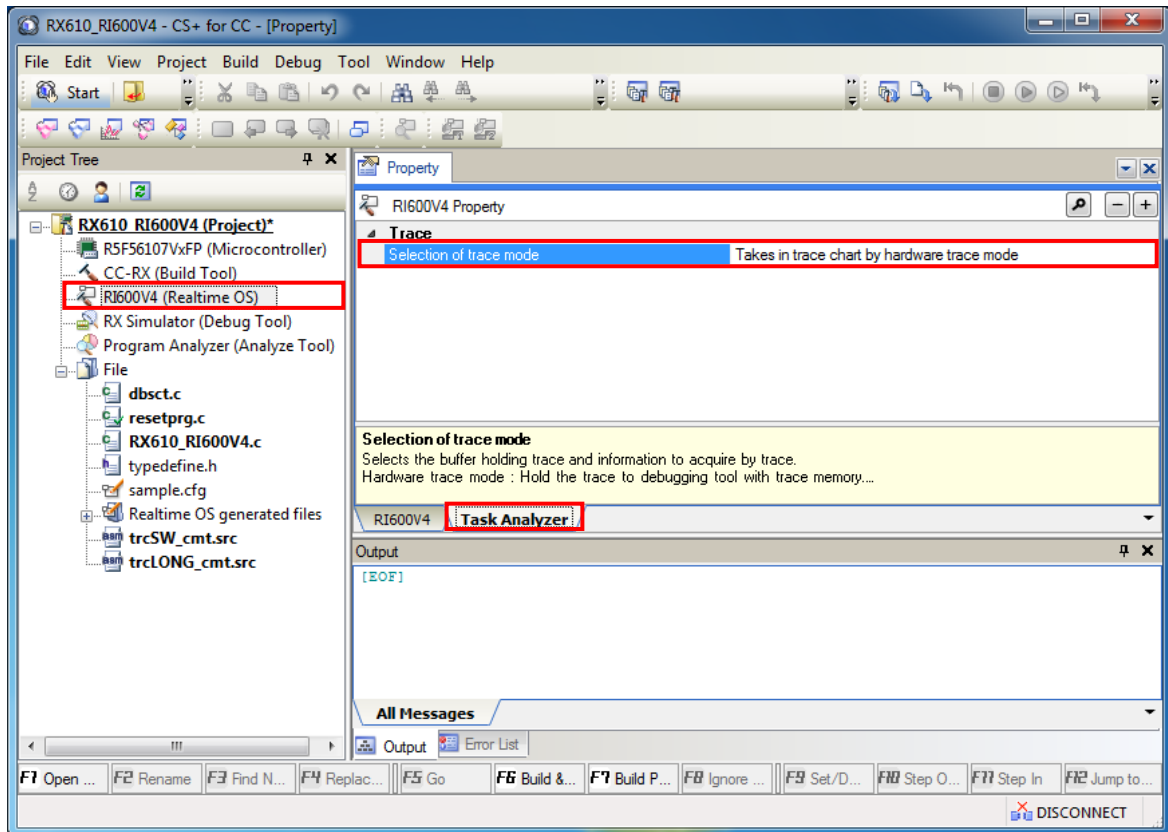
### 8.2.5.1. Change in the trace mode

Selects the “Selection of trace mode” in the “Task Analyzer” tab of the property of the RI600V4.

When the trace mode is changed, you should build the program.

The monitor which uses it for every trace mode is different, the correct monitor is embedded by building the program.

figure 8-4 Plug-in Manager



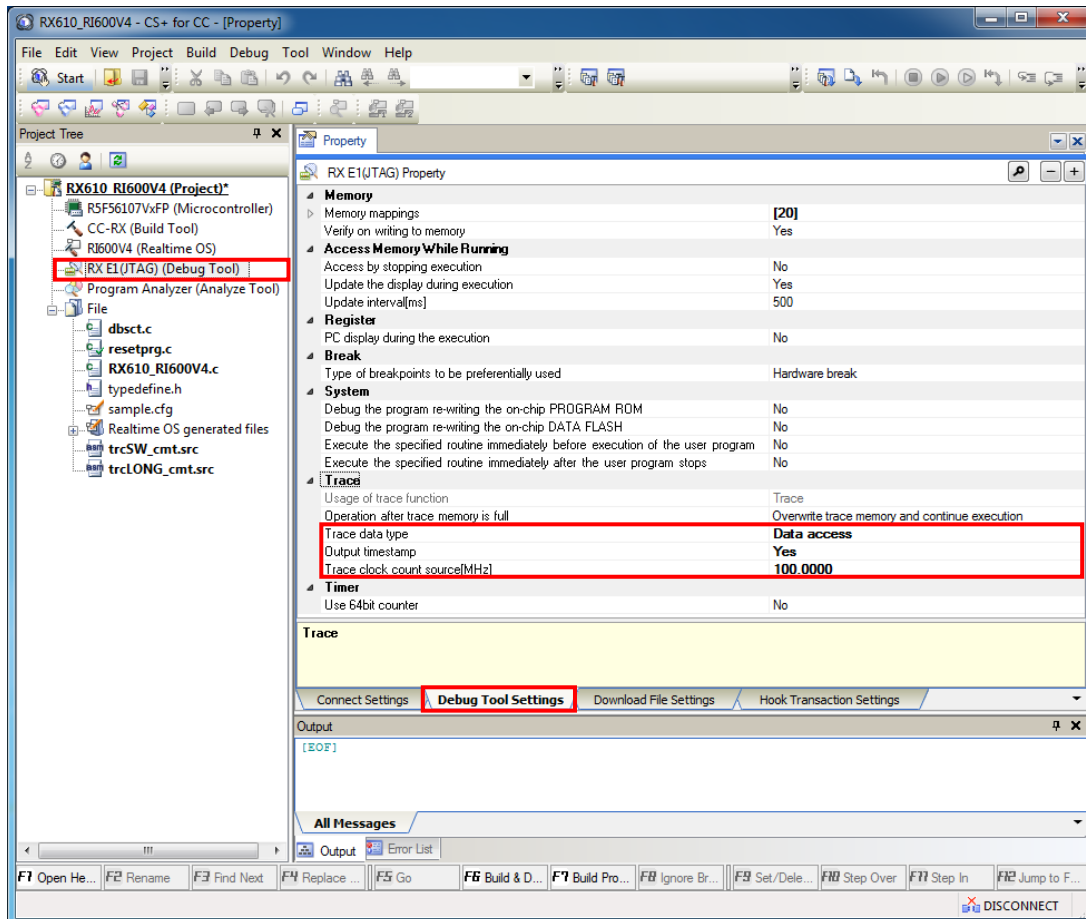
**8.2.5.2. Select “Taking in trace chart by hardware trace mode” when using E1/E20 emulator**

Please set up [Trace] category of [Debug Tool Settings] tab in the debug tool's property panel as follows.

- [Trace data type] : Data access
- [Output timestamp] : Yes
- [Trace clock count source [MHz] ] : Suitable value <sup>1</sup>

Ex: The setting of the sample program for RX610 is 100.000.

figure 8-5 Trace setting for E1/E20 Emulator



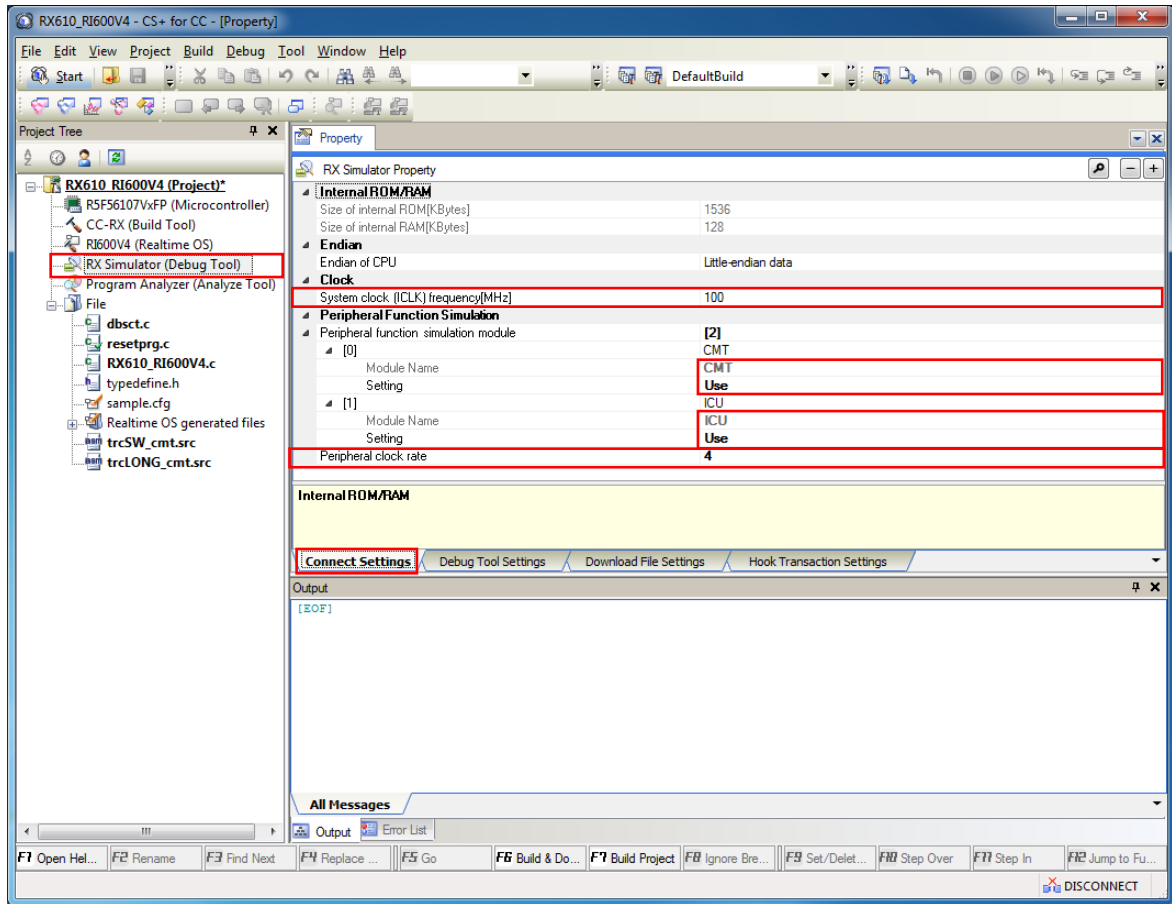
<sup>1</sup> The Realtime OS Task Analyzer displays various kinds of time information based on this value.

### 8.2.5.3. Select “Taking in trace chart by hardware trace mode” when using simulator

Please set up [Connect Settings] tab in the debug tool's property panel as follows.

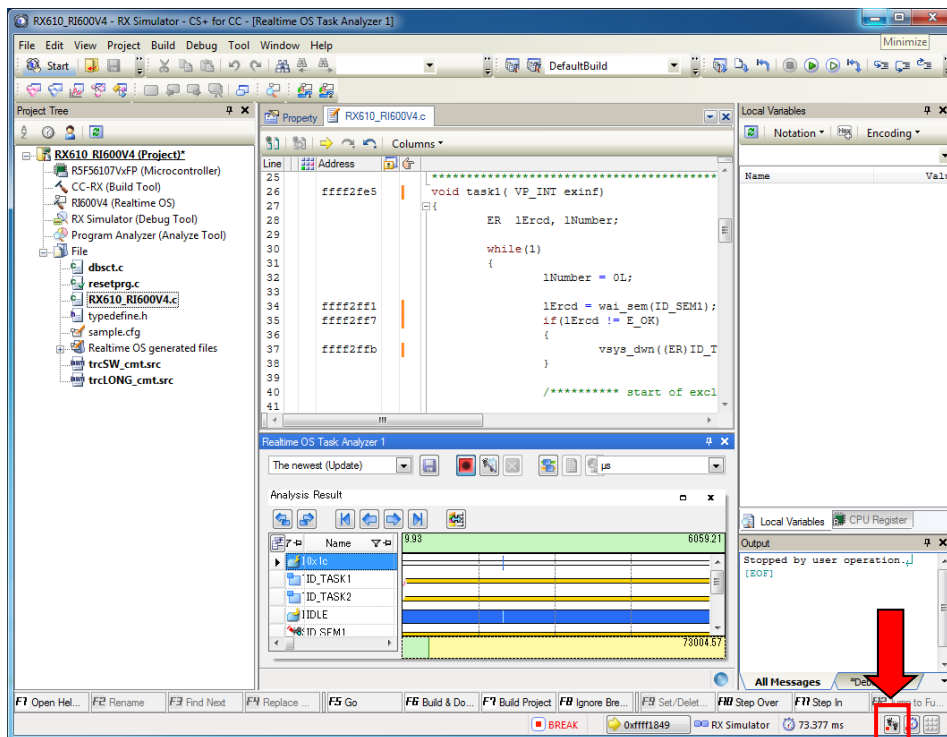
- [System clock (ICLK) frequency [MHz] ] in the [Clock] category  
Sets a suitable value to the [System clock (ICLK) frequency [MHz] ] in the [Clock] category in the “Connect Settings” tab of the Property Panel for the Debug tool  
Ex. The setting of the sample program for RX610 is 100.000.
- [CMT] in [Peripheral Function Simulation] category  
Sets “Use” to the [CMT] in the [Peripheral Function Simulation] category in the “Connect Settings” tab of the Property Panel for the Debug tool.
- [ICU] in [Peripheral Function Simulation] category  
Sets “Use” to the [ICU] in the [Peripheral Function Simulation] category in the “Connect Settings” tab of the Property Panel for the Debug tool.
- [Peripheral clock rate] in the [Peripheral Function Simulation] category  
Sets a suitable value to the [Peripheral clock rate] in the [Peripheral Function Simulation] category in the “Connect Settings” tab of the Property Panel for the Debug tool.  
Ex: The setting of the sample program for RX610 is 4.

Figure 8-6 “Connect Settings” tab of the Property Panel for the Debug tool



And the trace function of the debugger turns on (the icon of a footprint mark)

Figure 8-7 Trace switch



**8.2.5.4. Setting for the debugging tool**

Do not assign the following combination to the [Trace] category in the [Debug Tool Settings] tab in the property of the debugging tool.

- [Clear trace memory before running] : No
- [Accumulate trace time] : Yes

**8.2.5.5. Time stamp for the Software trace**

The timer of the kernel is used to acquire timestamps in Software tracing. Since the timer of the kernel uses OS timer interrupts, handling of timer interrupts is suspended while interrupts are disabled. Correct display of time is not possible while interrupts are disabled (e.g. by a task) for a period of 1 ms or longer (but the occurrence of interrupts or tasks will still be displayed in the correct order).

## 8.3. Cautions for e<sup>2</sup> studio

Refer to the e<sup>2</sup> studio Help.

## 9. Restrictions

### 9.1. Common restrictions of CS+ and e<sup>2</sup> studio

#### 9.1.1. Coding

(1) **Arrangement of section**

Please arrange tasks, handlers, and coding module in the range of 24 bits of program counter relative from "PRI\_KERNEL" which is the kernel module, ( $PRI\_KERNEL + dsp:24, -8388608 \leq dsp \leq 8388607$ , "dsp" means displacement of program counter relative).

(2) **Usage of Register Bank Save Function**

There are the following restrictions when using register bank save function.

- In case of using register bank save function and time management function of configuration settings, RI600V4 always uses one save register bank for time management function.
- In case of enabling the "MIN" context settings, you cannot use register bank save function. It is because of preventing from performance degradation by using the "MIN" settings together.
- We don't support RI600V4 operations if you use register bank save function and that a target device doesn't have this function, and also you execute configuration with more than the number of save register banks on a target device. Please check a user's manual of RX series for information of whether it has register bank save function and number of save register banks.

(3) **Double-precision floating-point**

In RI600V4 V1.06.00, we don't support using the double-precision floating-point.

#### 9.1.2. Realtime OS Resource Information Displaying Plug-in

(1) **Effect of resetting the display of waiting tasks (child nodes) on the display of the [Task] tabbed page**

Resetting the display of waiting tasks also resets the display of other tasks in the [Task] tabbed page. However, the information being displayed will be correct.

(2) **"Time Left" in "Realtime OS Resource Information Panel"**

The value displayed on the following items may become larger TIC\_NUME than the original value at the maximum.

- "Time Left" item in [Task] tab
- "Time Left" item in [Cyclic Handler] tab
- "Time Left" item in [Alarm Handler] tab

The original value can be calculated by the following formulas.

- When (The value displayed on "Time Left") > TIC\_NUME  
The original value = (The value displayed on "Time Left") – TIC\_NUME
- When (The value displayed on "Time Left") ≤ TIC\_NUME  
The original value = 0

### 9.1.3. Realtime OS Task Analyzer Plug-in

- (1) **In some cases, no correct result may display when filtering operation is performed in a CPU usage rate column**

The result filtered correctly may not be displayed by the kind of filtering. For example, when “displays 80% or more” as CPU usage rate is specified, all column is displayed without being filtered.



## 9.2. Restrictions of CS+

### 9.2.1. Realtime OS Build Tool Plug-in

(1) **Multiple build modes**

Do not use multiple build modes for the following reasons.

- The configurator options are common to all build modes. Even if multiple build modes are used, the same configurator options are applied.
- Every time the build mode is changed, the path to the kernel\_id.h file is added to [Additional include paths] of the build tool. Although the build-setting plug-in sets the correct path in [System include paths], the IDE adds the old path prior to the change of the build mode to [Additional include paths]. In the process of building, the build tool refers to the old path set by the IDE. This means that editing the configuration file to change the build mode before editing kernel\_id.h, for example, will not be reflected in building.

(2) **Utilizing existing projects**

If you choose to recycle as the basis of a new project an existing project that does not contain any files such as sit.s which are generated by the configurator, and you select copy processing for the files you will be reusing, the missing files such as sit.s that are supposed to be grayed out in the project tree will be deleted from the project tree.

(3) **Convert a High-performance Embedded Workshop project.**

When a High-performance Embedded Workshop project for the RI600/4 is converted into a CS+ project, the following settings are not reflected in a CS+ project.

- On the RX Standard Toolchain [RI600/4] tab, in the [Configuration] category, the [Miscellaneous options] settings except for the "[v] Displays the command option descriptions and detailed information on the version" option.
- On the RX Standard Toolchain [RI600/4] tab, in the [Configuration] category, the [User-defined options] settings.
- On the RX Standard Toolchain [RI600/4] tab, in the [Mkritbl] category, the "Search directory of MRC files" settings.

Please perform a setup required by [Property panel] of the system configuration file.

## 9.3. Restrictions of e<sup>2</sup> studio

### 9.3.1. Realtime OS Task Analyzer Plug-in

#### 9.3.1.1. Create a CS+ Project

The following is combination of debug tool and trace mode. Part of them has restriction.

Debug tool	Hardware trace mode	Software trace mode
E1	Restriction (1)	Restriction (3)
E2	Restriction (1)	Restriction (3)
E2 Lite	Restriction (1)	Restriction (3)
E20	Restriction (1)	Restriction (3)
Simulator	Restriction (2)	Restriction (3)

(1) **Using Emulator and Hardware trace, it cannot use time stamp**

If the debug tool is E1, E2, E2 Lite or E20, RTOS analyzer cannot use. Because it displays illegal information.

(2) **Using Simulator and Hardware trace, it cannot use time stamp**

If the debug tool is the simulator, time stamp cannot use. Because the simulator does not output time stamp. You can trace to see the trace chart, but the time between events is not right (the time is all one). If you analyze time information, please use the software trace mode.

(3) **Using Software trace, it may be not able to get trace data, if the trace buffer size is large.**

If the trace buffer size is large, when the program stops and get the trace data, e<sup>2</sup> studio does not return about 1 minutes. When e<sup>2</sup> studio returns, it is not able to get the trace data.

If this issue occurs, please specify a lower size to the trace buffer size.

#### 9.3.1.2. Restriction of using the task analyzer tool on the e<sup>2</sup> studio 64 bit

You cannot use the task analyzer tool on the e<sup>2</sup> studio 64 bit. If you want to use this tool, please use it on the e<sup>2</sup> studio 32 bit.

### 9.3.2. Realtime OS Resource Information Displaying Plug-in

#### 9.3.2.1. Restriction of using the resource information tool on the e<sup>2</sup> studio 64 bit

You cannot use the resource information tool on the e<sup>2</sup> studio 64 bit. If you want to use this tool, please use it on the e<sup>2</sup> studio 32 bit.

**Revision History**

Rev.	Date	Description	
		Page	Summary
1.00	Mar.25.22	-	New Publication

# General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

## 1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

## 2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

## 3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

## 4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

## 5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

## 6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.).

## 7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

## 8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

## Notice

1. 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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

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

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall 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. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,  
Koto-ku, Tokyo 135-0061, Japan  
[www.renesas.com](http://www.renesas.com)

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

## Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:  
[www.renesas.com/contact/](http://www.renesas.com/contact/)