# RENESAS

# RX Family Real-Time OS RI600V4 V1.06.00

# Release Notes

# Contents

1.		Packa	ged Tools	4
2.		User's	Manual	5
	2.1.	Errata o	n User's Manuals	5
3.		Target	Devices	6
4.		Opera	ting Environment	7
	4.1.	Hardwa	re Environment	7
	4.2.	Software	e Environment	7
	4.3.	Support	ed Tools	7
	4.4.	Caution	for Realtime OS Task Analyzer	8
5.		Installa	ation Notes	9
	5.1.	Commo	n 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+ e	environment	10
		5.2.1.	Cautions for Installation	. 10
		5.2.2.	Cautions for Uninstallation	. 11
	5.3.	In e² stu	dio environment	11
		5.3.1.	Cautions for Installation	. 11
		5.3.2.	Cautions for Uninstallation	. 11
6.		Chang	es from previous released versions	.12
	6.1.	Change	s 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.	Change	s in RI600V4 V1.05.00	14

		6.2.1.	Kernel	14
		6.2.2.	Sample programs for CS+	14
	6.3.	Change	es 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
7.		About	the latest plug-ins	23
8.		Cautio	ons	24
	8.1.	Commo	on cautions for CS+ and e <sup>2</sup> studio	24
		8.1.1.	Distinction of Version	24
		8.1.2.	Shift from a Previous Version	24
		8.1.3.	GUI Configurator	25
		8.1.4.	Timer Template File	26
		8.1.5.	How to Build Kernel Source Code	27
		8.1.6.	Stack Consumption	28
		8.1.7.	Caution when using RX610 Group	33
		8.1.8.	Caution when using global optimization of compile option	33
		8.1.9.	Caution about initializing OS data	34
		8.1.10.	Caution about building application	34
	8.2.	Caution	s for CS+	35
		8.2.1.	Enable Plug-ins	35
		8.2.2.	Create a CS+ Project	36
		8.2.3.	Sample Programs	38
		8.2.4.	Cautions for Realtime OS Resource Information Panel	39
		8.2.5.	Cautions for Realtime OS Task Analyzer	40
	8.3.	Caution	s for e <sup>2</sup> studio	44
9.		Restri	ctions	45
	9.1.	Commo	on restrictions of CS+ and e <sup>2</sup> studio	45
		9.1.1.	Coding	45
		9.1.2.	Realtime OS Resource Information Displaying Plug-in	45
		9.1.3.	Realtime OS Task Analyzer Plug-in	46
	9.2.	Restrict	ions of CS+	47



	9.2.1.	Realtime OS Build Tool Plug-in	47
9.3.	Restric	tions of e <sup>2</sup> studio	.48
	9.3.1.	Realtime OS Task Analyzer Plug-in	48
Revisio	n Histo	ry	49



# 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	A
R0R5RX00TCW01A	Evaluation License, Unlimited hosts	A
R0R5RX00TCW01K	Mass-production License, 3000 copies	А
R0R5RX00TCW01U	Mass-production License, Unlimited copies	A
R0R5RX00TCW01Z	Mass-production License, Unlimited copies, With source code	В

The following tools are provided.

Con	tents	Name	Version
	Real-Time OS RI600V4 Kernel Object		V1.06.00
		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
В	А	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	V1.06.00



# 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)
- Memory capacity: 2 GB or more recommended.
   Minimum requirement is 1 GB (32-bit / Windows® 7 or later) and 2 GB (64-bit / Windows 7 or later).
- Display: Resolution at least 1024 x 768 dots; at least 65,536 colors

# 4.2. Software Environment

The following OS are supported.

- Windows 7 (32bit, 64bit) : CS+, e<sup>2</sup> studio
- Windows 8.1 (32bit, 64bit) : CS+, e<sup>2</sup> studio
- Windows 10 (32bit, 64bit) : CS+, e<sup>2</sup> studio

Remark: It is recommended that the latest service pack is installed on any OS.

The following runtime libraries are required.

- .NET Framework 4.5.2
- Runtime library of Microsoft Visual C++ 2010 SP1

# 4.3. Supported Tools

The following tools are supported.

Tool Name	Manufacturer	Version	
Integrated development environment CS+ for CC	Renesas Electronics	V3.02.00 or later	
Integrated development environment e <sup>2</sup> studio	Renesas Electronics	V4.2 or later	
C/C++ Compiler CC-RX	Renesas Electronics	V1.02.01 or later (for RXv1)	
		V2.04.01 or later (for RXv2)	
		V3.01.00 or later (for RXv3) , We	
		recommend these version	

# 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 32bit and the installation drive is C:
   C:\Program Files\Common Files\Renesas Electronics CubeSuite+\
- 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, or CS+ Realtime OS RI600V4 Source Release, or CS+ Realtime OS RI600V4 Trial

# 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.

 Renesas Realtime OS RI600V4 Object Release, or Renesas Realtime OS RI600V4 Source Release, or Renesas Realtime OS RI600V4 Trial



# 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,	0.400	0.440
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

- 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+".
- 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 tabsThe tab selection area is divided into two columns, and adds the icon to each tab.
- (4) A part of messages is improvedA 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

# 6.1.5. Realtime OS Task Analyzer Plug-in

selected.

- 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+".
- 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+

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</ri_root>	V1.02.01 or later	RXv1 architecture	ri600lit.lib	For little endian
			RXv1 architecture	ri600lit.lib	For little endian
2	<ri_root>\library\rxv2</ri_root>	V2.01.00 or later	RXv2 architecture     RXv3 architecture	ri600big.lib	For big endian

#### (2) The kernel version information

The version change is as follows.

Item	Before	After
TKERNEL_PRVER,	0.440	0.450
T_RVER prver (the return value of "ref_ver" and "iref_ver")	0x140	0x150

# 6.2.2. Sample programs for CS+

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</ri_root>	V1.02.01 or later	RXv1 architecture	ri600lit.lib ri600bia.lib	For little endian
			RXv1 architecture	ri600lit.lib	For little endian
2	<ri_root>\library\rxv2</ri_root>	<ri_root>\library\rxv2 V2.01.00 or later • RXv2 architecture</ri_root>	ri600big.lib	For big endian	
3	<pre><ri_root>\library\rxv3 V3.01.00 or later</ri_root></pre>			ri600lit.lib	For little endian
		ri600lit_notrace.lib	For little endian (mode: Not tracing)		
		RXv2 architecture     RXv3 architecture	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,	0.450	0.400
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 described in a configuration file as follows.

In case of this version of CFG600 V1.0 6.00 or later and configuring by a configuration file which doesn't have the definitions in the table below, CFG600 applies the default values for each definition.

Definition	Description Definition		Possible	When	Definition
		Format	To Omit	Omitting	Range
system.max_regbank	Define maximum number of save register banks. (Depend on target device)	Numeric Value	Yes	0	0 ~ 256
task.regbank	Define whether to use register bank save function.	Symbol	Yes	OFF	ON or OFF
interrupt_vector.pragma_switch	Define switches to pass to "#prgma interrupt" directive. If you specify "REGBANK" keyword, "bank=xxx" is actually passed to "#prgma interrupt" directive. The "xxx" is a bank number which is assigned in an order to be specified "REGBANK" keyword.	Symbol	Yes	No switches are passed	E,F,S,H,NH, ACC,NOACC, REGBANK
interrupt_fvector.pragma_switch	As above.	Symbol	Yes	No switches are passed	S,ACC,H,NH, NOACC, REGBANK



In addition, the following table are summarized the main error / warning messages added by this change and how to handle them.

Туре	ID	Message	How to handle
Error	O1003	Unknown token> <max_regbank></max_regbank>	(When using CFG600 V1.03.00.002 or earlier)
		Unknown token> <regbank></regbank>	Delete the indicated definition. *1
		Unknown interrupt_vector.pragma_switch>	
		<regbank></regbank>	
		Unknown interrupt_fvector.pragma_switch>	
		<regbank></regbank>	
Error	O1037	system.max_regbank should be less than 256.	Change [system.max_regbank] to 256 (0x100)
		(Max.=0x100).	or less.
Error	O1038	Use register save bank is bigger than	Change register bank save function of either
		system.max_regbank.	[task], [interrupt_vector], or [interrupt_fvector]
			to unused or change the time management
			function to unused, and make the total number
			of use less than the value of
			[system.max_regbank]. *2
Warning	O4001	system.max_regbank is not defined	Define [system.max_regbank]. *3
Warning	O5001	system.context is MIN. sysetem.max_regbank	Change [system.max_regbank] to 0. *3
		change 0.	Change [system.context] to something other
			than [MIN]. *3

\*1 : CFG600 V1.03.00.002 or earlier does not support the definition for the register bank save function.

\*2 : If [system.max\_regbank] is 1 or more and [clock.timer] is other than [NOTIMER], the time management function also uses one save register bank. That is, in this case, the total number of save register bank usages of [task], [interrupt\_vector] and [interrupt\_fvector] needs to be smaller than the value of [system.max\_regbank]. \*3 : At the warning, the default value (system.max\_regbank = 0) is applied internally.

# 6.3.3. GUI Configurator

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.]	Select checkbox when using
	Checkbox	register bank save function
		throughout system, this activates
		following items in this table.
	[Max. Save register bank] Display	Maximum number of save register
	box / Modification button	banks are displayed. /You can
		modify this number to pressing the
		button.
[Task] View	[Register bank save function]	[Use] or [Not use] keyword is
	Column	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]	[regbank] Checkbox	Select checkbox to use register
Dialog		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]	[regbank] Checkbox	Select checkbox to use register
Dialog		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.

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 supportedWe added build function for the kernel library for the RXv3 architecture.
- (2) Tracing-Disabled Kernel Library is supportedWe added build function for the tracing-disabled kernel library.
- (3) Correct the error of loading a project fileWe 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 supportedWe 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

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
                                                    // Let FPSW RMbits=00 (round to nearest)
#define _ROUND 0x0000000
#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,
        Oxffffffff,
        Oxffffffff,
};
#pragma address __TMINFreg=0x00120060
                                                  // TMINF register
const unsigned long ___TMINFreg = 0xffffffff;
#pragma address __MDEreg=0x00120064
                                                  // MDE register (Single Chip Mode)
#ifdef __BIG
    const unsigned long __MDEreg = 0xfffffff8;
                                             // 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)
}
```

# 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	This version

# 8.1.2. Shift from a Previous Version

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



## 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.

5	·······
Interrupt Handler Definition	? ×
Relocatable Interrupt Vector Number         Vector Number       30         Address         Address	Description Language     High-Level Language (TA_HLNG)     Assembly Language (TA_ASM)     Interruption classification     Non-kernel Interrupt
#PRAGMA Extended Function Switch	<ul> <li>Use as fast interrupt.(fint)</li> <li>Support OS trace.(hook)</li> <li>Don't save ACC register.(no_acc)</li> </ul>
Register bank save function         regbank         Used       0         Remained       0	
ОК	Cancel

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.

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
	RX600 Series RX66T Group *1
	RX600 Series RX65N Group *1
	RX600 Series RX651 Group *1
	RX600 Series RX64M 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
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

Table 8-1 Timer templa	te file
------------------------	---------

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

# 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 was 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 was 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 was 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>&</sup>lt;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.

	Service call	The use size of	The use size of	Note
		User stack (a)	System stack (b), (c)	
Task Mar	agement Funct	lion		
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	

Table 8-2 Stack usage of service-call



	Service call	The use size of	The use size of	Note		
		User stack(a)	System stack (b), (c)			
Task Mar	Task Management Function					
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			
Task Dep	endent Synchro	onization Function				
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	irsm_tsk	0	48			
29	frsm_tsk	4	44			
30	ifrsm_tsk	0	48			
31	dly_tsk	4	44			
Semaphore						
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	The use size of	Note
		User stack(a)	System stack (b), (c)	
Eventflag				
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	
Data Que	ue			
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	
Mailbox				
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	The use size of	Note
		User stack(a)	System stack (b), (c)	
Mutex				
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	
Message	Buffer			
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	
Fixed-size	ed Memory Poo	bl		
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	
Variable Size Memory Pool				
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	The use size of	Note	
		User stack(a)	System stack (b), (c)		
Time Mar	Time Management Function				
99	set_tim	4	44		
100	iset_tim	0	48		
101	get_tim	4	44		
102	iget_tim	0	48		
Cyclic Ha	ndler				
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		
Alarm Ha	ndler				
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		
System S	tate Manageme	ent Function			
115	rot_rdq	4	44		
116	irot_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	
130	ivsta knl	0	48	initialized it uses it	



	Service call	The use size of User stack(a)	The use size of System stack (b), (c)	Note
System S	tate Manageme	ent Function		
131	vsys_dwn	4	44	
132	ivsys_dwn	0	48	
Interrupt I	Management Fi	unction		
133	chg_ims	4	44	
134	ichg_ims	0	48	
135	get_ims	4	4	
136	iget_ims	4	4	
137	Kernel	0	52	When a kernel interrupt handler ends,
	interrupt			48 bytes of the system stack is
	handler			consumed from just before generating
				of the interrupt.
System Configuration Management Function				
138	ref_ver	4	44	
139	iref_ver	0	48	
Object Reset Function				
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\_ROM" option After: There is not "-rom=DRI\_ROM=PRI\_ROM" option

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

After: Before: #pragma section C C\$DSEC #pragma section C C\$DSEC extern const struct { extern const struct { \_UBYTE \*rom\_s; \_UBYTE \*rom\_e; UBYTE \*rom s UBYTE \*rom\_e; UBYTE \*ram\_s; \_UBYTE \*ram\_s; \_DTBL[] = { DTBL[] = { } } ∫IbLj = { \_\_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("D"), \_\_secend("D"), \_\_sectop("R") }, \_\_sectop("D\_2"), \_\_secend("D\_2"), \_\_sectop("R\_2") }, \_\_sectop("D\_1"), \_\_secend("D\_1"), \_\_sectop("R\_1") } sectop("R\_1") }, \_\_sectop("D\_1"), \_\_secend("D\_1"), \_\_sectop("RI\_RAM") } \_\_sectop("DRI\_ROM"), \_\_secend("DRI\_ROM"), \_\_sectop("RRI\_RAM") } }; }: #pragma section C C\$BSEC #pragma section C C\$BSEC extern const struct { extern const struct { \_UBYTE \*b\_s; \_UBYTE \*b\_e; \_UBYTE \*b\_s; \_UBYTE \*b\_e; \_\_BTBL[] = { { \_\_sectop("B"), \_\_secend("B") }, { \_\_sectop("B\_2"), \_\_secend("B\_2") }, { \_\_sectop("B\_1"), \_\_secend("B\_1") } } } \_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)

Plug-in Manager					
Checked plug-ins are loaded at the CS+ start-up. These settings are enabled at the next start-up. * You can never uncheck a check box of the grayout plugin that is required by the CS+. Also, on the [Basic Function] tab, it is recommended that the checkboxes of the plug-in for the target microcontroller of the development are not cleared. Basic Function Additional Function					
Module Name	Description				
<ul> <li>Code Generator Plug-in</li> <li>Code Generator/PinView Plug-in</li> <li>Code Generator/PinView Plug-in</li> <li>Debug Console Plug-in</li> <li>Editor plug-in DLL</li> <li>Editor plug-in DLL</li> <li>IonPython Console Plug-in</li> <li>Pin Configurator Plug-in</li> <li>Pogram Analyzer Plug-in</li> </ul>	Plug-in to generate the device driver automatically. Plug-in to generate the device driver automatically DebugConsole plug-in to support using standard I/A SEditor DLL It is a console where the IronPython commands an Plug-in to define the device pin configuration. Plug-in to analyze program.				
Image: Start Star					
<					
	OK Cancel <u>H</u> elp				

#### 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.

Create Project				<b>— X</b>
Microcontroller:	RX			•
Using <u>m</u> icrocontroller:				
(Search microcontroller)		Update	]	
RX621     RX62G     RX62G     RX62N     RX62N     RX62T     RX630     R5F56307CxFN(8)     R5F56307CxFN(8)     R5F56307CxLA(10)     R5F56307CxLA(10)     R5F56307CxLA(10)	Dpin) Dpin) Opin)	Product Name:R5F On-chip ROM size[ On-chip RAM size[ Additional Informati	=56307CxFP (KBytes]:384 Bytes]:65536 on:Package=PLQP0	100KB-A
Kind of project:	Application(RI600	V4,CC-RX)		
Project name:	test			
Place:	C:\Users\guest\D	ocuments	•	Browse
	Make the proje	ct folder		
C:\Users\guest\Documents\tes	t\test.mtpj			
Pass the file composition of	an existing project	to the new project		
Project to be passed: (Input project file to be diverted.)			Browse	
Copy composition files in the	ne diverted project f	older to a new proje	ect folder.	
Create Cancel Help				

#### 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.

RX610_RI600V4 - CS+ for CC - [Property]	
File Edit View Project Build Debug T	Tool Window Help
🕅 Start 🚽 🍟 🔏 🖻 🖄 🔊	(*) # # # # # # # # # # # # # # # # # # #
i 💎 🖓 🖉 🧐 🥵 i 🗆 🗭 🗣 🔍 i	
Project Tree 4 ×	Property
2 @ 2 2	RI600V4 Property
E RX610 RI600V4 (Project)*	▲ Trace
R5F56107VxFP (Microcontroller)	Selection of trace mode Takes in trace chart by hardware trace mode
RI600V4 (Realtime OS)	
⊡])) File	
resetprq.c	
RX610_RI600V4.c	Selection of trace mode
	Selects the buffer holding trace and information to acquire by trace. Hardware trace mode : Hold the trace to debugging tool with trace memory
trcSW cmt.src	Ri600V4
trcLONG_cmt.src	Output 4 ×
	[FOL]
	All Messages -
4	🔜 Output 📴 Error List
F? Open F2 Rename F3 Find N F4 Rep	plac   F5 Go   F6 Build &   F7 Build P   F8 Ignore   F9 Set/D   F10 Step O   F11 Step In   F12 Jump to

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.

indure 8-5 Trace setting for E1/E20 Emulator	figure 8-5	Trace setting for E1/E20 Emulator
--	------------	-----------------------------------

RX610_RI600V4 - CS+ for CC - [Property]		
File Edit View Project Build Debug T	ool Window Help	
- <del> </del>		
Project Tree 4 X	Property	- X
2 🕜 🙎 🔳	S BX E1(ITAG) Property	<b>–</b> +-
RX610 RI600V4 (Project)*		
R5F56107VxFP (Microcontroller)	A Memory	1001
CC-RX (Build Tool)	Verifu on writing to memory	Yee
PI600V4 (Realtime OS)	A Access Memory While Burning	163
NIRV E1/ITAG) (Debug Teel)	Access by stopping execution	No
KA EL(JTAG) (Debug Tool)	Lindate the display during execution	Vae
Program Analyzer (Analyze I ool)	Update interval[me]	500
😑 🎒 File	A Begister	300
	PC display during the execution	No
resetprg.c	A Break	110
	Tupe of breakpoints to be preferentially used	Hardware break
	A Sustem	
sample cfg	Debug the program re-writing the on-chip PBOGBAM BOM	No
Realtime OS generated files	Debug the program re-writing the on-chip DATA ELASH	No
Realtime OS generated mes	Execute the specified routine immediately before execution of the user program.	No
trcsw_cmt.src	Execute the specified routine immediately after the user program stops	No
trcLONG_cmt.src	4 Trace	
	Usage of trace function	Trace
	Operation after trace memory is full	Overwrite trace memory and continue execution
	Trace data type	Data access
	Output timestamp	Yes
	Trace clock count source[MHz]	100.0000
	⊿ Timer	
	Use 64bit counter	No
	-	
	Trace	
	Connect Settings Debug Tool Settings Download File Settings	Hook Transaction Settings
	Output	4 <b>X</b>
	[EOF]	
	All Messages	•
< >	🙈 Output 🔠 Error List	
F7 Open He F2 Rename F3 Find Next F	Replace	Set/Dele FNB Step Over F77 Step In FNB Jump to F
		Big Disconnect

<sup>&</sup>lt;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.

RX610_RI600V4 - CS+ for CC - [Property]		• <b>X</b>
File Edit View Project Build Debug T	Tool Window Help	
		è- "
: 🕰 🔄 🛃 🚽 🖓 🖽 🗳		- <u>-</u>
- 🖓 🖓 🖉 🦃 🤻 💷 🗭 🗣 🦳 I		
Project Tree 4 ×	Property	- x
2 🕜 🙎 🔳	RY Simulator Property	<b>P</b> - +
RX610 RI600V4 (Project)*	The statistical property and the statistical statistic	
R5F56107VxFP (Microcontroller)	Size of Internal BOM/KButas] 1536	_
CC-RX (Build Tool)	Size of internal BM/KB/tes] 128	
RI600V4 (Realtime OS)	Endian	
	Endian of CPU Little-endian data	
Program Analyzer (Analyze Tool)		
🖃 📲 File	System clock (ICLK) frequency[MHz] 100	
dbsct.c	Peripheral Function Simulation	
- resetpro.c	Peripheral function simulation module     [2]	
RX610 RI600V4.c	A [0] CMT	
typedefine.h	Module Name CMT	
	4 [1] ICU	
Realtime OS generated files	Module Name ICU	
trcSW cmt.src	Setting Use	
trcLONG cmt.src	Peripheral clock rate 4	
	Internal ROM/RAM	
	Connect Settings Debug Tool Settings Download File Settings Hook Transaction Settings	•
	Output	<b>д X</b>
	[EOF]	
	All Messages /	-
- III	🔝 Output 🚰 Error List	
F? Open Hel F2 Rename F3 Find Next	F4 Replace F5 Go F6 Build & Do F7 Build Project F8 Ignore Bre F9 Set/Delet F80 Step Over F71 Step In F8	Jump to Fu
		CONNECT

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 $e^2$ 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.

# **Revision History**

		Description	
Rev.	Date	Page	Summary
1.00	Jul 1,2019	-	New Publication

### Notice

- 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.
- 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 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.
- 5. 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.

- 6. 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.
- 7. 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 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.
- 8. 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.
- 9. 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.
- 10. 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.
- 11. This document shall not be reprinted, 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.
- (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.4.0-1 November 2017)

### **Corporate Headquarters**

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan www.renesas.com

# 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: <u>www.renesas.com/contact/</u>.

# Trademarks

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

# **Mouser Electronics**

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

**Renesas Electronics:** 

R0R5RX00TCW011 R0R5RX00TCW01A R0R5RX00TCW01K R0R5RX00TCW01U R0R5RX00TCW01Z