

Contents

Chapter 1. Packaged Tools	3
Chapter 2. User's Manual	4
Chapter 3. Target Devices	5
Chapter 4. Operating Environment	6
4.1 Hardware Environment	6
4.2 Software Environment	6
4.3 Supported Tools	6
4.4 Cautions for Realtime OS Task Analyzer	6
Chapter 5. Installation Cautions	7
5.1 Cautions for Installation	7
5.1.1 Cautions for administrator privileges	7
5.1.2 Cautions for execution environment	7
5.1.3 Cautions for network drives	7
5.1.4 Cautions for installation folder name	7
5.1.5 Cautions for required files after installation	7
5.1.6 Cautions for modifying and repairing functions	7
5.1.7 Cautions for changing the installation folder	7
5.1.8 Cautions for version of installed tools	8
5.1.9 Cautions for starting installer	8
5.1.10 Cautions for installation order	8
5.2 Cautions for Uninstallation	8
5.2.1 Cautions for administrator privileges	8
5.2.2 Cautions for uninstallation folder name	8
5.2.3 Cautions for adding/repairing via other than the installer	8
Chapter 6. Key Word for Uninstallation	9
Chapter 7. Changes from Previous Version (V1.02.00)	10
7.1 Added or Updated Tools	10
7.2 Support RX100 Series	10
7.3 Support CubeSuite+ V2	10
7.4 Support CC-RX V2	10
7.5 Function Addition	10
7.5.1 Resource Information Displaying Plug-in	10
7.6 Cancel the Restriction	11
7.6.1 Kernel	11
7.6.2 Plug-ins for CubeSuite+	11
7.7 Kernel Version Information	12

Chapter 8. Changes by On-line Update..... 13

Chapter 9. Cautions..... 14

9.1 Shift from a Previous Version..... 14

9.2 Cautions in the Build 15

9.2.1 “nostuff” compiler option for the boot processing file.....15

9.2.2 ROM to RAM Mapped Section16

9.3 GUI Configurator 18

9.3.1 Summary.....18

9.3.2 [Interrupt Handler Definition] dialog box18

9.4 Timer Template File 19

9.5 How to Build Kernel Source Code..... 19

9.6 Stack Consumption 20

9.6.1 Stack consumption of base clock interrupt handler (*clocksz1, clocksz2, clocksz3*).....20

9.6.2 Stack consumption of service calls (*svcsz*)20

9.6.3 When the kernel library is built24

9.7 Cautions When Using RX610 Group 25

9.8 Cautions for Realtime OS Resource Information Panel..... 25

9.8.1 View after real-time OS is initialized.....25

9.8.2 Use programs with debug information generated.....25

9.9 Cautions for Realtime OS Task Analyzer..... 26

9.9.1 Change in the trace mode26

9.9.2 Select “Taking in trace chart by hardware trace mode” when using E1 emulator.....26

9.9.3 Select “Taking in trace chart by hardware trace mode” when using simulator26

9.10 Sample Programs..... 27

9.11 Cautions for RI600/4 Users..... 28

9.11.1 Delete the correspondence to the realtime OS trace for the High-performance Embedded Workshop28

9.11.2 Kernel Version information28

9.11.3 Cautions for CubeSuite+28

Chapter 10. Restrictions..... 29

10.1 Restrictions of CubeSuite+..... 29

10.1.1 Convert a High-performance Embedded Workshop project29

10.1.2 The path that contains the service call information file29

10.1.3 “Time Left” in “Realtime OS Resource Information Panel”30

10.1.4 Realtime OS Task Analyzer31

Chapter 11. Correction of Document 32

Chapter 1. Packaged Tools

Agreement type and contents are different according to the product.

Product Name	Agreement Type	Contents
R0R5RX00TCW011	Evaluation License, Limited 1 host	A
R0R5RX00TCW01A	Evaluation License, Unlimited hosts	A
R0R5RX00TCW01K	Mass-production License, 3000 copies	A
R0R5RX00TCW01U	Mass-production License, Unlimited copies	A
R0R5RX00TCW01Z	Mass-production License, Unlimited copies, With source code	B

The following tools are provided.

Contents	Name	Version	
B	A	Realtime OS RI600V4 Kernel Object	V1.02.01
		Command-line Configurator "cfg600"	V1.02.01
		Table Generation Utility "mkritbl"	V1.02.00
		Plug-ins for CubeSuite+	
		Realtime OS Build Tool Plug-in (Common)	V2.00.00
		Realtime OS Build Tool Plug-in (RI600V4)	V2.00.00
		Realtime OS Analysis Control Plug-in (Common)	V2.00.00
		Realtime OS Analysis Control Plug-in (μ ITRON4)	V2.00.00
		Realtime OS Analysis Control Plug-in (RI600V4)	V2.00.00
		Realtime OS Resource Information Displaying Plug-in (Common)	V2.00.00
		Realtime OS Resource Information Displaying Plug-in (μ ITRON4)	V2.00.00
		Realtime OS Task Analyzer Plug-in (Common)	V2.00.00
		Realtime OS Task Analyzer Plug-in (Panel)	V2.00.00
		Realtime OS Task Analyzer Plug-in (RI600V4)	V2.00.00
GUI Configurator "GUI600"	V1.01.00		
Realtime OS RI600V4 Kernel Source Code	V1.02.01		

Chapter 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	R20UT0751EJ0102
RI600V4 Real-Time Operating System User's Manual: Coding	R20UT0711EJ0102
RI600V4 Real-Time Operating System User's Manual: Debug	R20UT0775EJ0101
RI600V4 Real-Time Operating System User's Manual: Analysis	R20UT2185EJ0100
RI Series Real-Time Operating System User's Manual: Message	R20UT0756EJ0102

These PDF files are provided by this package or Renesas Electronics Home page.

Chapter 3. Target Devices

The following devices are supported by the product.

- RX600 series MCU
- RX200 series MCU
- RX100 series MCU

Chapter 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 or more (64-bit Windows® 7 or Windows® 8 requires 2 GB or more)
- Display: Resolution at least 1024 x 768; at least 65,536 colors

4.2 Software Environment

The following software environments are supported.

- Windows® XP (32bit)
- Windows Vista® (32bit, 64bit)
- Windows® 7 (32bit, 64bit)
- Windows® 8 (32bit, 64bit)
- .NET Framework 4
- Runtime library of Microsoft Visual C++ 2010 SP1
- Internet Explorer 6.0 or later

Remark: For 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 CubeSuite+	Renesas Electronics	V2.00.00 or later
C/C++ Compiler CC-RX	Renesas Electronics	V1.02.01 or later

4.4 Cautions for Realtime OS Task Analyzer

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

- Simulator
- Emulator which can be set as “Yes” to [Output timestamp]
For example, when E1 emulator is used on RX100 series, [Output timestamp] is fixed to “No”, so Realtime OS Task Analyzer cannot be used.
Note, [Output timestamp] is in [Trace] category on [Debug Tool Settings] tab of [Property] panel for using debug tool.

Chapter 5. Installation Cautions

This section provides cautions for installation and uninstallation.

5.1 Cautions for Installation

5.1.1 Cautions for administrator privileges

Windows® administrator privileges are required to install the software.

5.1.2 Cautions for execution environment

The Internet Explorer 6.0 (or later), the.NET Framework and the Visual C++ runtime libraries are required to run the installer.

5.1.3 Cautions for network drives

The software cannot be installed from a network drive.

It also cannot be installed to a network drive.

5.1.4 Cautions 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 backspace character (\) as the path separator for the installation folder. Do not use the forward slash (/).

5.1.5 Cautions 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+\

5.1.6 Cautions 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.

Clicking [Modify] from the Add or Remove Programs (Windows® XP), or Uninstall or change a program (Windows Vista® / Windows® 7 / Windows® 8) dialog boxes will cause an error.

5.1.7 Cautions for changing the installation folder

To change the folder that tools are installed to, you must first uninstall all tools, and then perform installation again.

To uninstall all tools, start the Integrated Uninstaller, and after deleting all the tools that are displayed.

5.1.8 Cautions for version of installed tools

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

5.1.9 Cautions 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.

5.1.10 Cautions for installation order

Before installing, please install CubeSuite+. Note that this package will be installed to the same folder that CubeSuite+ was installed to.

5.2 Cautions for Uninstallation**5.2.1 Cautions for administrator privileges**

Windows® administrator privileges are required to uninstall the software.

5.2.2 Cautions 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.

5.2.3 Cautions 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.

Chapter 6. Key Word for Uninstallation

There are two ways to uninstall this product.

- Use the integrated uninstaller (uninstalls CubeSuite+)
- Use separate uninstaller (uninstalls this product only)

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

- Add/Remove Programs (Windows® XP)
- Programs and Features (Windows Vista® / Windows® 7 / Windows® 8)

After the applet appears, delete from the following.

- CubeSuite+ Realtime OS Common Plugins
- CubeSuite+ Realtime OS RI600V4 Plugins
- CubeSuite+ Realtime OS RI600V4 Object Release, or CubeSuite+ Realtime OS RI600V4 Source Release

Chapter 7. Changes from Previous Version (V1.02.00)

7.1 Added or Updated Tools

Name	Before	After
Realtime OS RI600V4 Kernel Object	V1.02.00	V1.02.01
Command-line Configurator "cfg600"	V1.02.00	V1.02.01
Table Generation Utility "mkritbl"	V1.02.00	No change
Plug-ins for CubeSuite+		
Realtime OS Build Tool Plug-in (Common)	V1.03.00	V2.00.00
Realtime OS Build Tool Plug-in (RI600V4)	V1.01.00	V2.00.00
Realtime OS Analysis Control Plug-in (Common)	V1.01.01	V2.00.00
Realtime OS Analysis Control Plug-in (μ ITRON4)	V1.01.01	V2.00.00
Realtime OS Analysis Control Plug-in (RI600V4)	V1.00.01	V2.00.00
Realtime OS Resource Information Displaying Plug-in (Common)	V1.02.00	V2.00.00
Realtime OS Resource Information Displaying Plug-in (μ ITRON4)	V1.02.00	V2.00.00
Realtime OS Task Analyzer Plug-in (Common)	V1.00.00	V2.00.00
Realtime OS Task Analyzer Plug-in (Panel)	V1.00.00	V2.00.00
Realtime OS Task Analyzer Plug-in (RI600V4)	V1.00.00	V2.00.00
GUI Configurator "GUI600"	V1.01.00	No change
Realtime OS RI600V4 Kernel Source Code	V1.02.00	V1.02.01
RI Series Real-Time Operating System User's Manual: Start	Rev.1.02	No change
RI600V4 Real-Time Operating System User's Manual: Coding	Rev.1.02	No change
RI600V4 Real-Time Operating System User's Manual: Debug	Rev.1.01	No change
RI600V4 Real-Time Operating System User's Manual: Analysis	Rev.1.00	No change
RI Series Real-Time Operating System User's Manual: Message	Rev.1.02	No change

7.2 Support RX100 Series

The RX100 series is added to the target devices. And the sample program "RX111_RI600V4" is also added.

7.3 Support CubeSuite+ V2

The supported CubeSuite+ version is changed to V2.00.00.

7.4 Support CC-RX V2

The "CC-RX V2.00.00" is added to the supported tools.

7.5 Function Addition

7.5.1 Resource Information Displaying Plug-in

In [Task], [Cyclic Handler] and [Alarm Handler] of the Realtime OS Resource Information Panel, the function which opens the selected file by double click has been added..

7.6 Cancel the Restriction

The following restrictions have been canceled.

7.6.1 Kernel

(1) Problem with issuing the `unl_cpu` service call

Tool News URL : <http://tool-support.renesas.com/eng/toolnews/121216/tn4.htm>

(2) Problem with giving a value to the address of the reset vector

Tool News URL : <http://tool-support.renesas.com/eng/toolnews/121216/tn4.htm>

7.6.2 Plug-ins for CubeSuite+

(1) “Optimizes accesses to external variables” compiler option

When the following setting is performed by [Property panel] of “CC-RX (Build Tool)”, the error shown below may occur.

- Setting of [Property panel] of “CC-RX(Build Tool)”

Tab	Category	Item	Setting
Compiler Options	Optimization	Optimizes accesses to external variables	Yes(Optimizes the intermodule) (-map)

- Error

```
(O) : A3001 (F) Can't open file 'DefaultBuild\ritable.src'
```

(2) Realtime OS Task Analyzer : Error/warning

When the following error or warning occurs, applicable help is not displayed even if the help button or F1 key is pressed.

Number	Message
E1150000	The file < <i>filename</i> > does not exist.
E1150001	Reading trace data failed. The file type is invalid.
E1151001	Could not switch the trace of task analyzer on. The point trace is full.
W1151000	The setup of the property is not reflected in the load module. Build the program, it will be solved this warning.

(3) Realtime OS Task Analyzer : Extracting the analysis range

While extracting the analysis range, when operation which updates a display of the Realtime OS Task Analyzer is performed, the value mistaken in “CPU Usage Rate” may be displayed.

- When “The newest (Update)” is selected, run the program and the execution stops.
- Change from “Not update” to “The newest (Update)”.
- “Open trace data...”

(4) Realtime OS Task Analyzer : “Display User/Kernel Ratio” and “Display Total Ratio”

While extracting the analysis range, when “Display User/Kernel Ratio” and “Display Total Ratio” is changed, the value mistaken in “CPU Usage Rate” will be displayed.

7.7 Kernel Version Information

Item	Before	After
TKERNEL_PRVER, T_RVER.prver, which is returned by the ref_ver and iref_ver	0x0120	0x0121

Chapter 8. Changes by On-line Update

There is no updating after the release of this version.

Chapter 9. Cautions

This section describes cautions for RI600V4 V1.02.01.

9.1 Shift from a Previous Version

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

9.2 Cautions in the Build

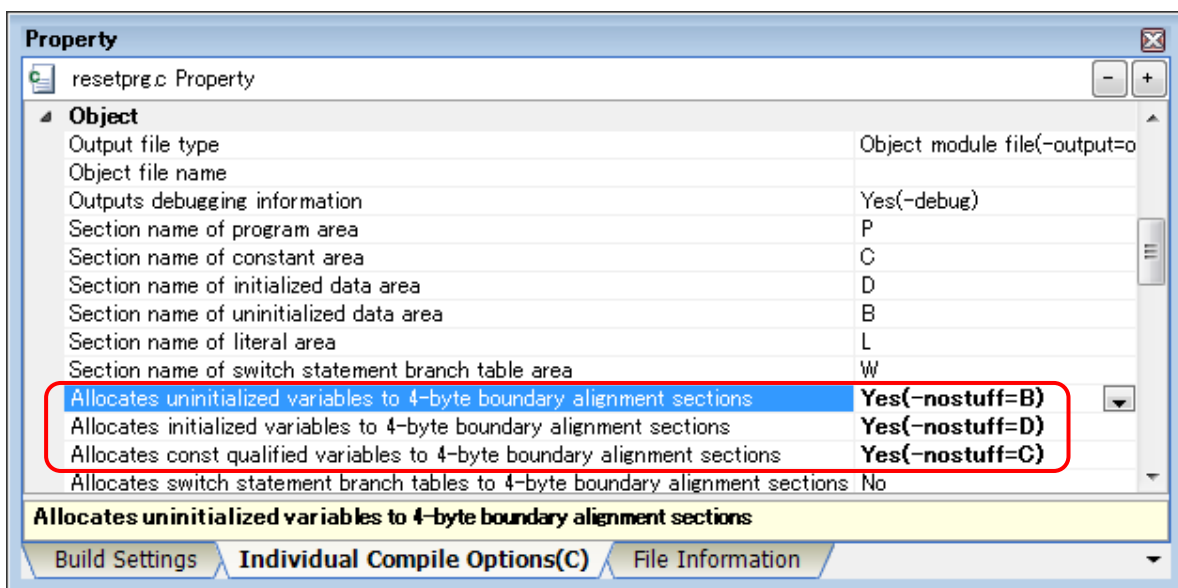
9.2.1 “nostuff” compiler option for the boot processing file

It is necessary to set “nostuff” option for the boot processing file (“resetprg.c” in the sample project) like a mention in “16.2.3 Compiler option for boot processing file” in “RI600V4 Real-Time Operating System User's Manual: Coding (R20UT0711EJ, Rev.1.02)”. If not, RI600V4 does not work correctly.

To set “nostuff” option only for the boot processing file, please set any of the following in the [Individual Compile Options] tab of [Property] panel for the boot processing file. To set “nostuff” option for all, please set any of the following in the [Compiler Options] tab of [Property] panel for [CC-RX (Build Tool)].

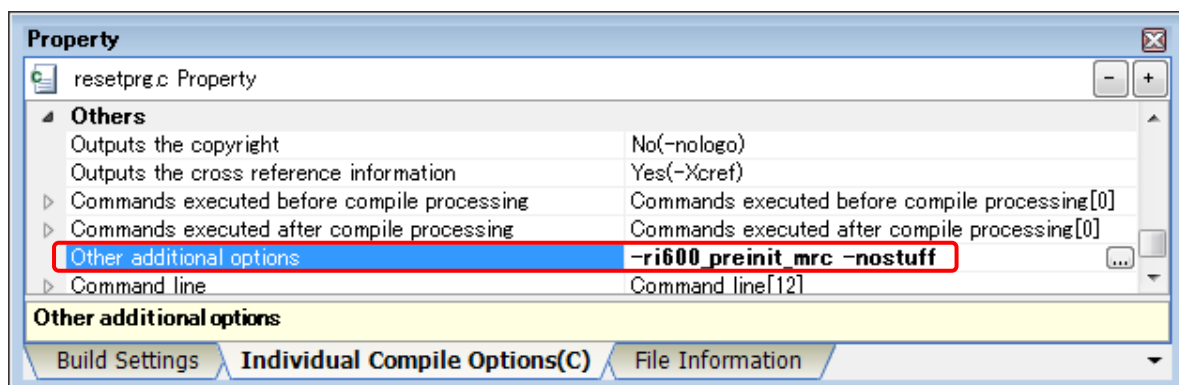
(1) Set in the [Object] category

Like the figure below, set “Yes” in [Allocates uninitialized variables to 4-byte boundary alignment sections], [Allocates initialized variables to 4-byte boundary alignment sections] and [Allocates const qualified variables to 4-byte boundary alignment sections].



(2) Set in the [Others] category

Like the figure below, add “-nostuff” to [Other additional options].



9.2.2 ROM to RAM Mapped Section

It is necessary to map initialized data section from ROM to RAM. It is also necessary to map “DRI_ROM” section in RI600V4 to “RRI_RAM” section in RAM. To that end, setting of section information file and setting of the linkage editor are required.

(1) Section information file

The section initialization function “_INITSCT()” called from Boot processing function (PowerON_Reset_PC()) is provided as a compiler standard library.

By the __INITSCT(), uninitialized data sections are initialized to zero, for initialized data sections, the initial values of the ROM area are copied to the RAM area.

The user needs to write the sections to be initialized to the tables for section initialization (DTBL and BTBL) into the section information file as user-won coding module. The section address operator is used to set the start and end addresses of the sections used by the _INITSCT function. Section names in the section initialization tables are declared, using C\$BSEC for uninitialized data areas, and C\$DSEC for initialized data areas.

Note, the section information file provided by RI600V4 as sample is “dbsct.c”, and setting to map from “DRI_ROM” section to “RRI_RAM” section has already done in this file.

```
#include "typedefine.h"
#pragma unpack

#pragma section C C$DSEC
extern const struct {
    _UBYTE *rom_s;          /* Start address of the initialized data section in ROM */
    _UBYTE *rom_e;          /* End address of the initialized data section in ROM */
    _UBYTE *ram_s;          /* Start address of the initialized data section in RAM */
} _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;           /* Start address of non-initialized data section */
    _UBYTE *b_e;           /* End address of non-initialized data section */
} _BTBL[] = {
    { __sectop("B"), __secend("B") },
    { __sectop("B_2"), __secend("B_2") },
    { __sectop("B_1"), __secend("B_1") }
};

#pragma section

/*
** CTBL prevents excessive output of L1100 messages when linking.
** Even if CTBL is deleted, the operation of the program does not change.
*/
_UBYTE * const _CTBL[] = {
    __sectop("C_1"), __sectop("C_2"), __sectop("C"),
    __sectop("W_1"), __sectop("W_2"), __sectop("W")
};

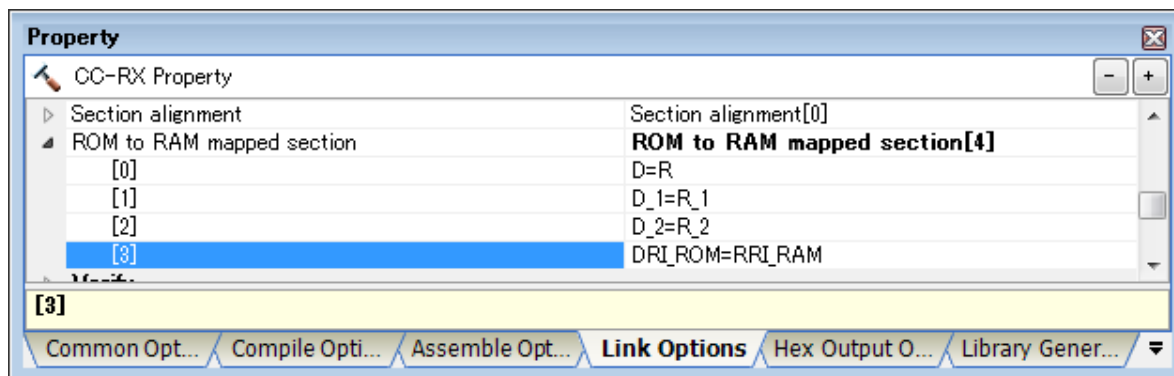
#pragma packoption
```


(2) Setting of linkage editor

About sections described in _DTBL[] of the section information file, it is necessary to perform setting to map sections placed on ROM to sections placed on RAM by using “rom” option of the linkage editor.

Set [Link Options] tab of [Property] panel for [CC-RX (Build Tool)] like the figure below.

Note, with the sample project provided by RI600V4, this has been already set.



9.3 GUI Configurator

9.3.1 Summary

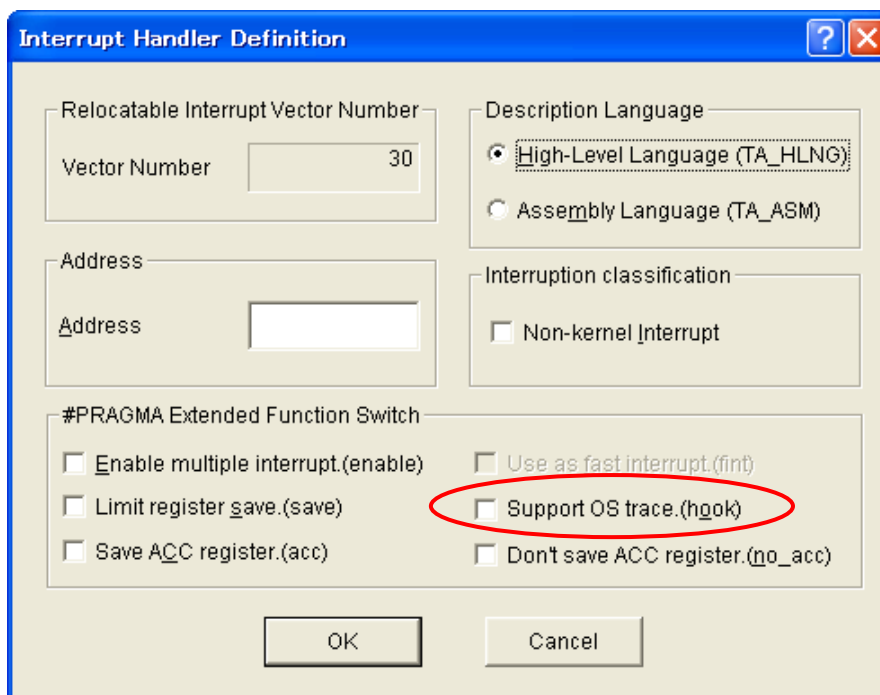
The GUI configurator is a tool that permits the user to generate a system configuration file by entering various kernel configuration information from GUI screen. Using the GUI configurator, it is possible to build the kernel without the need for learning how to write a system configuration file.

To start the GUI configurator, run "< installation folder >\bin600\Guiconfig_RI600.exe".

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

9.3.2 [Interrupt Handler Definition] dialog box

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



9.4 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.

Template File	Corresponded MCUs
rx610.tpl	RX600 series RX610 group
rx62t.tpl	RX600 series RX62T group
rx62n.tpl	RX600 series RX62N group RX600 series RX621 group
rx630.tpl	RX600 series RX630 group RX600 series RX631 group RX600 series RX63N group RX600 series RX63T group RX200 series RX21A group
rx210.tpl	RX200 series RX210 group RX200 series RX220 group RX100 series RX111 group

9.5 How to Build Kernel Source Code¹

Since the RI600V4 kernel is provided in the library form, it does not usually need to build the kernel.

The kernel source code is stored in "< installation folder >src600". To build the kernel, set current folder to this folder, and run "nmake.exe"². The environment variable settings are needed by compiler when building the kernel.

Example:

```
C:\Program Files\Renesas Electronics\CubeSuite+RI600V4\src600> nmake(RET)
```

The kernel library is generated to the following folders.

Kernel Library Name	Contents
product\big\debug\ri600big.lib	Big endian library with debugging information
product\big\release\ri600big.lib	Big endian library without debugging information
product\little\debug\ri600lit.lib	Little endian library with debugging information
product\little\release\ri600lit.lib	Little endian library without debugging information

Please copy "src600" folder to the writable folder if you don't have the write-access permission to the product installation folder. After the build, copy the generated library to the "lib600" folder under the product installation folder by the user who has write-access permission to the product installation folder.

¹ The source code is only attached to R0R5RX00TCW01Z.

² "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.

9.6 Stack Consumption

9.6.1 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* = 120
- *clocksz2* = 120
- *clocksz3* = 176

9.6.2 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 User stack(a)	The use size of System stack ((b),(c))	Note
Task Management Function				
1	act_tsk	0	44	
2	iact_tsk	0	48	
3	can_act	0	48	
4	ican_act	0	48	
5	sta_tsk	0	44	
6	ista_tsk	0	48	
7	ext_tsk	0	60	ext_tsk is called at the return from the task beginning function.
8	ter_tsk	0	104	
9	chg_pri	0	44	
10	ichg_pri	0	52	
11	get_pri	0	48	
12	iget_pri	0	48	
13	ref_tsk	0	48	
14	iref_tsk	0	48	
15	ref_tst	0	48	
16	iref_tst	0	48	
Task Dependent Synchronization Function				
17	slp_tsk	0	44	
18	tslp_tsk	0	44	
19	wup_tsk	0	44	
20	iwup_tsk	0	52	
21	can_wup	0	48	
22	ican_wup	0	48	
23	rel_wai	0	100	
24	irel_wai	0	108	
25	sus_tsk	0	44	
26	isus_tsk	0	48	
27	rsm_tsk	0	44	
28	irmsm_tsk	0	48	
29	frsm_tsk	0	44	
30	ifrsn_tsk	0	48	
31	dly_tsk	0	44	
Semaphore				
32	sig_sem	0	44	
33	isig_sem	0	52	
34	wai_sem	0	44	
35	pol_sem	0	48	
36	ipol_sem	0	48	
37	twai_sem	0	44	
38	ref_sem	0	48	
39	iref_sem	0	48	
Eventflag				
40	set_flg	0	52	
41	iset_flg	0	68	
42	clr_flg	0	48	
43	iclr_flg	0	48	

	Service call	The use size of User stack(a)	The use size of System stack ((b),(c))	Note
44	wai_flg	0	44	
45	pol_flg	0	48	
46	ipol_flg	0	48	
47	twai_flg	0	44	
48	ref_flg	0	48	
49	iref_flg	0	48	
Data Queue				
50	snd_dtq	0	44	
51	psnd_dtq	0	44	
52	ipsnd_dtq	0	52	
53	tsnd_dtq	0	44	
54	fsnd_dtq	0	44	
55	ifsnd_dtq	0	48	
56	rcv_dtq	0	44	
57	prcv_dtq	0	44	
58	iprcv_dtq	0	56	
59	trcv_dtq	0	44	
60	ref_dtq	0	48	
61	iref_dtq	0	48	
Mailbox				
62	snd_mbx	0	44	
63	isnd_mbx	0	52	
64	rcv_mbx	0	44	
65	prcv_mbx	0	48	
66	iprcv_mbx	0	48	
67	trcv_mbx	0	44	
68	ref_mbx	0	48	
69	iref_mbx	0	48	
Mutex				
70	loc_mtx	0	44	
71	ploc_mtx	0	44	
72	tlloc_mtx	0	44	
73	unl_mtx	0	52	
74	ref_mtx	0	48	
Message Buffer				
75	snd_mbf	0	44	
76	psnd_mbf	0	44	
77	ipsnd_mbf	0	56	
78	tsnd_mbf	0	44	
79	rcv_mbf	0	56	
80	prcv_mbf	0	56	
81	trcv_mbf	0	56	
82	ref_mbf	0	48	
83	iref_mbf	0	48	
Fixed-sized Memory Pool				
84	get_mpf	0	48	
85	pget_mpf	0	48	
86	ipget_mpf	0	48	

	Service call	The use size of User stack(a)	The use size of System stack ((b),(c))	Note
87	tget_mpf	0	48	
88	rel_mpf	16	44	
89	irel_mpf	0	56	
90	ref_mpf	0	48	
91	iref_mpf	0	48	
Variable Size Memory Pool				
92	get_mpl	24	80	
93	pget_mpl	0	88	
94	ipget_mpl	0	88	
95	tget_mpl	24	80	
96	rel_mpl	0	92	
97	ref_mpl	0	48	
98	iref_mpl	0	48	
Time Management Function				
99	set_tim	0	48	
100	iset_tim	0	48	
101	get_tim	0	48	
102	iget_tim	0	48	
Cyclic Handler				
103	sta_cyc	0	48	
104	ista_cyc	0	48	
105	stp_cyc	0	48	
106	istp_cyc	0	48	
107	ref_cyc	0	48	
108	iref_cyc	0	48	
Alarm Handler				
109	sta_alm	0	48	
110	ista_alm	0	48	
111	stp_alm	0	48	
112	istp_alm	0	48	
113	ref_alm	0	48	
114	iref_alm	0	48	
System State Management Function				
115	rot_rdq	0	44	
116	irotd_rdq	0	48	
117	get_tid	0	48	
118	iget_tid	0	48	
119	loc_cpu	0	48	
120	iloc_cpu	0	48	
121	unl_cpu	0	44	
122	iunl_cpu	0	48	
123	dis_dsp	0	48	
124	ena_dsp	0	44	
125	sns_ctx	0	48	
126	sns_loc	0	48	
127	sns_dsp	0	48	
128	sns_dpn	0	48	
129	vsta_knl	0	52	After the system stack pointer is initialized,

	Service call	The use size of User stack(a)	The use size of System stack ((b),(c))	Note
130	ivsta_knl	0	52	it uses it.
131	vsys_dwn	0	44	
132	ivsys_dwn	0	44	
Interrupt Management Function				
133	chg_ims	0	48	
134	ichg_ims	0	44	
135	get_ims	4	4	
136	iget_ims	4	4	
137	Kernel interrupt handler	0	48	When a kernel interrupt handler ends, 48 bytes of the system stack is consumed from just before generating of the interrupt.
System Configuration Management Function				
138	ref_ver	0	48	
139	iref_ver	0	48	
Object Reset Function				
140	vrst_dtq	0	44	
141	vrst_mbx	0	44	
142	vrst_mbf	0	44	
143	vrst_mpf	0	44	
144	vrst_mpl	0	60	

9.6.3 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.

9.7 Cautions 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

9.8 Cautions for Realtime OS Resource Information Panel

9.8.1 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.

9.8.2 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".

9.9 Cautions for Realtime OS Task Analyzer

9.9.1 Change in the trace mode

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

9.9.2 Select “Taking in trace chart by hardware trace mode” when using E1 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 value³
Ex: The setting of the sample program for RX610 is 100.000.

9.9.3 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 [Clock] category : Suitable value³
Ex. The setting of the sample program for RX610 is 100.000.
- [CMT] in [Peripheral Function Simulation] category : Use
- [ICU] in [Peripheral Function Simulation] category : Use
- [Peripheral clock rate] in [Peripheral Function Simulation] category : Suitable value³
Ex: The setting of the sample program for RX610 is 4.

And do not assign the following combination to [Trace] category in [Debug Tool Settings] tab.

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

³ The Realtime OS Task Analyzer displays various kinds of time information based on this value.

9.10 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 Realtime 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
};
```

9.11 Cautions for RI600/4 Users

The RI600V4 is the Real-Time Operating System product which has been changed from the RI600/4 to be able to use on the CubeSuite+. This section explains the changes from RI600/4 V.1.01 Release 00 to RI600V4 V1.02.01.

9.11.1 Delete the correspondence to the realtime OS trace for the High-performance Embedded Workshop

In the RI600/4, “H” needed to be specified as “interrupt_vector[].pragma_switch” and “interrupt_fvector[].pragma_switch” in the system configuration file to trace interrupt handlers.

In the RI600V4, “pragma_switch=H” is ignored.

9.11.2 Kernel Version information

Item	Before (RI600/4)	After (RI600V4)
TKERNEL_PRVER, T_RVER.prver, which is returned by the ref_ver and iref_ver	0x0110	0x0121

9.11.3 Cautions for CubeSuite+

When you convert a High-performance Embedded Workshop project for RI600/4 into the CubeSuite+ project, you should run a rebuild.

Chapter 10. Restrictions

This section describes restrictions on RI600V4 V1.02.01

10.1 Restrictions of CubeSuite+

There is the following restriction when using CubeSuite+ V2.00.00.

10.1.1 Convert a High-performance Embedded Workshop project.

(1) Description

When a High-performance Embedded Workshop project for the RI600/4 is converted into a CubeSuite+ project, the following settings are not reflected in a CubeSuite+ 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 [Mkritel] category, the "Search directory of MRC files" settings.

(2) Workaround

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

10.1.2 The path that contains the service call information file

(1) Description

When the following setting is performed by [Property panel] of the system configuration file, the load module generated by build operation may be inaccurate. In that case, a part of service call(s) returns E_NOSPT error.

- Setting of [Property panel] of the system configuration file

Tab	Category	Item	Setting
System Configuration File Related Information	Service Call Information File	The path that contains the service call information file.	String containing a blank

(2) Workaround

Please set the path which does not contain a blank to [The path that contains the service call information file.].

Or please copy required service call information files to the build mode folder before build operation.

10.1.3 "Time Left" in "Realtime OS Resource Information Panel"

(1) Description

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

(2) Workaround

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

10.1.4 Realtime OS Task Analyzer

(1) Description

When the Realtime OS Task Analyzer is used in “Taking in trace chart by software trace mode”, the illegal writing of 0 in long-word (4-bytes) to address 0x00008007 is performed at the start-up of the RI600V4 (vsta_knl).

(2) Workaround

Please avoid by one method of the followings. Note, the method (b) cannot avoid this restriction when address 0x00008007 for the used MCU is not RAM.

(a) Add patch code

Please add the following source file to a project.

```

        .SECTION          PRI_KERNEL, CODE
        .GLB      __RI_trcSW_init
__RI_trcSW_init:
        .GLB      __RI_BUFCB
MOV.L    #__RI_BUFCB, R1
MOV.L    #1, 8[R1]

        MOV.L    #0FFFFFFFH, 4[R1]
        .GLB      __RI_TRCBUF
MOV.L    #__RI_TRCBUF, [R1]

        .GLB      __RIUSR_trcSW_init_tmr
BSR      __RIUSR_trcSW_init_tmr

MOV.L    #__RI_BUFCB, R1
MOV.L    #0, 8[R1]

        .GLB      __RI_trcinf_SW
        .GLB      __RI_TRCVALIDPTN
MOV.L    #__RI_TRCVALIDPTN, R1
MOV.L    #__RI_trcinf_SW, R2

        RTS

        .END

```

When linked, the following warning occurs, but there is no problem.

```

W0561320:Duplicate symbol "__RI_trcSW_init" in "DefaultBuild\swtrc_patch.obj"
W0561320:Duplicate symbol "__RI_trcSW_init" in "<RI600V4 library>(trace_SW)"

```

(b) Arrangement of sections

Please do not arrange a section effective in the range of address 0x0000800A from 0x00008007.

Chapter 11. Correction of Document

There is no correction of document.

All trademarks and registered trademarks are the property of their respective owners.

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
 2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
 3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
 4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
 5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
 6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
 7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
 9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
 11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.
(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.

2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited

1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.

7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.

Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited

Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.

80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.

11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141