

Thank you for using the CS+ integrated development environment.

This document describes the restrictions and points for caution. Read this document before using the product.

## Contents

<b>Chapter 1. Target Devices .....</b>	<b>2</b>
<b>Chapter 2. User's Manuals .....</b>	<b>3</b>
<b>Chapter 3. Keywords When Uninstalling the Product.....</b>	<b>4</b>
<b>Chapter 4. Changes .....</b>	<b>5</b>
4.1 Changes to the CC-RH compiler .....	5
4.1.1 <i>Improvements to the feature for checking source code against MISRA-C:2012 rules [Professional edition].....</i>	<i>5</i>
4.1.2 <i>Synchronization function when control registers are updated [Professional edition] .....</i>	<i>6</i>
4.1.3 <i>Enhanced optimization.....</i>	<i>8</i>
4.1.4 <i>Measurement of times for CAN communications.....</i>	<i>11</i>
4.1.5 <i>Change to the output of the initial values for array type variables.....</i>	<i>11</i>
4.1.6 <i>Addition of special symbols.....</i>	<i>11</i>
4.1.7 <i>Change to the specification of link map files.....</i>	<i>12</i>
4.1.8 <i>Specification of the type of end record of Motorola S-type files.....</i>	<i>12</i>
4.1.9 <i>Addition of numbers to messages when using the evaluation version.....</i>	<i>12</i>
4.1.10 <i>Change to the messages for linkage errors.....</i>	<i>13</i>
4.1.11 <i>Improvement to the method of authenticating licenses.....</i>	<i>13</i>
4.1.12 <i>Rectified point for caution.....</i>	<i>13</i>
4.1.13 <i>Other changes and improvements.....</i>	<i>13</i>

---

## Chapter 1. Target Devices

The target devices supported by the CC-RH compiler are listed on the Website.

Please see the URL below.

CS+ Product Page:

<http://www.renesas.com/cs+>

## Chapter 2. User's Manuals

Please read the following user's manuals along with this document.

Manual Name	Document Number
CC-RH Compiler User's Manual	R20UT3516EJ0103
CS+ Integrated Development Environment User's Manual: CC-RH Build Tool Operation	R20UT3283EJ0104

---

## Chapter 3. Keywords When Uninstalling the Product

There are two ways to uninstall this product.

- Use the integrated uninstaller from Renesas (uninstalls all CS+ components)
- Use the Windows uninstaller (only uninstalls this product)

To use the Windows uninstaller, select "CS+ CC-RH V1.06.00" from "Programs and Features" of the control panel.

## Chapter 4. Changes

This chapter describes changes to the CC-RH compiler.

### 4.1 Changes to the CC-RH compiler

This section describes changes to the CC-RH compiler from V1.05.00 to V1.06.00. Note that the features which are only available to users holding a registered license for the Professional edition are indicated as **[Professional edition]**.

#### 4.1.1 Improvements to the feature for checking source code against MISRA-C:2012 rules **[Professional edition]**

The following rule numbers have been added to those which can be designated as arguments of the -Xmisra2012 option, which selects checking by the compiler of source code against the specified MISRA-C:2012 rules.

The V1.06.00 compiler supports Amendment 1 of MISRA-C:2012.

[Mandatory rules] **12.5, 21.13**

[Required rules] **13.2, 13.5, 21.15, 21.16**

[Advisory rules] **17.5, 17.8**

The following are the numbers of MISRA-C:2012 rules against which each revision of compilers can check source code for compliance.

<i>Rule classification (number of rules in the standard)</i>	<i>V1.03.00</i>	<i>V1.04.00</i>	<i>V1.05.00</i>	<i>V1.06.00</i>
Mandatory rules (16)	3	3	4	<b>6</b>
Required rules (108)	31	58	76	<b>80</b>
Advisory rules (32)	7	21	23	<b>25</b>
Total number of rules (156)	41	82	103	<b>111</b>

#### 4.1.2 Synchronization function when control registers are updated [Professional edition]

When control registers are consecutively updated by the store instruction of RH850, synchronization processing must be inserted following each update of a control register so that the order of updating matches that in the code in the source file. However, synchronization processing is not required if the control registers being consecutively updated are in the same group.

The professional edition of CC-RH supports the insertion of synchronization processing into positions where the updating of control registers and the updating is consecutive.

The following steps enable this function.

1. Specify the address range and group information on control registers with a directive from the language extensions.

```
#pragma register_group start address, end address[, id="group ID"]
```

2. Specify the required compiler option.

```
-store_reg[=mode]
```

Specify one of the following as the argument "*mode*" of the compiler option.

- ***list***

This option allows the compiler to detect updating of the control registers defined as #pragma register\_group and display the addresses of the write instructions through the standard error output, except where the next instruction will clearly be for updating of a control register in the same group, in which case the compiler does not display the addresses.

- ***list\_all***

This option allows the compiler to detect updating of the control registers defined as #pragma register\_group and display the addresses of the write instructions through the standard error output. The addresses are displayed regardless of whether the next instruction will clearly be for updating of a control register in the same group or not.

- ***sync***

This option allows the compiler to detect updating of the control registers defined as #pragma register\_group and inserts synchronization processing after updating of these registers, except where the next instruction will clearly be for updating of a control register in the same group, in which case the compiler does not insert synchronization processing.

- ***ignore***

#pragma register\_group is ignored and warnings are not output.

The following shows an example of the use and effects of this function.

[Example of the code in a C source file "scr.c"]

```

1: #pragma register_group 0xfedf0000, 0xfedffff, id="CPU"
2: #pragma register_group 0xfef00000, 0xfef0ffff, id="0"
3:
4: #define REG1 (*(volatile unsigned char*)0xfedf0000) /* Control register of group "CPU" */
5: #define REG2 (*(volatile unsigned char*)0xfedf0001) /* Control register of group "CPU" */
6: #define REG3 (*(volatile unsigned short*)0xfef00000) /* Control register of group "0" */
7:
8: void func(void) {
9:     REG1 = 0;
10:    REG2 = 1;
11:    REG3 = 2;
12: }

```

Control registers are consecutively updated in the function *func* in the order REG1 (group "CPU"), REG2 (group "CPU"), and REG3 (group "0").

The compiler detects the following items and applies processing according to the argument "*mode*" of the compiler option.

- ✓ Since REG1 and REG2 belong to **the same group**, synchronization processing is **not required** after updating of REG1.
- ✓ Since REG2 and REG3 belong to **different groups**, synchronization processing is **required** after updating of REG2.
- ✓ Since processing after updating of REG3 is not clear, synchronization processing is **required** after updating of REG3.
- When the `-store_reg=list` option is specified

The following messages are displayed through the standard error output.

```

src.c(10): M0536601:control register is written. (id=CPU, 0xfedf0001)
src.c(11): M0536601:control register is written. (id=0, 0xfef00000)

```

- When the `-store_reg=list_all` option is specified

The following messages are displayed through the standard error output.

```

src.c(9): M0536601:control register is written. (id=CPU, 0xfedf0000)
src.c(10): M0536601:control register is written. (id=CPU, 0xfedf0001)
src.c(11): M0536601:control register is written. (id=0, 0xfef00000)

```

- When the `-store_reg=syncp` option is specified

Synchronization processing is inserted in the output code.

As synchronization processing, the compiler outputs the combination of a load instruction for the same control register and the `syncp` instruction.

```

_func:
    .stack _func = 0
    movhi 0x0000FEDF, r0, r2
    st.b r0, 0x00000000[r2]          ; REG0 is updated.
    ; Synchronization processing is not inserted because the next
    ; instruction is for updating of a control register in the same group.
    movhi 0x0000FEDF, r0, r2
    mov 0x00000001, r5
    st.b r5, 0x00000001[r2]        : REG1 is updated.
    ld.bu 0x00000001[r2], r10      ; Loading from the same control register
    syncp                          ; Synchronization instruction
    movhi 0x0000FEE0, r0, r2
    mov 0x00000002, r5             ; REG2 is updated.
    st.h r5, 0x00000000[r2]
    ld.hu 0x00000002[r2], r10     ; Loading from the same control register
    syncp ;                          ; Synchronization instruction
    jmp [r31]

```

#### 4.1.3 Enhanced optimization

For V1.06.00, optimization has been further enhanced on points (1) and (2), listed and described below.

##### (1) Handling of switch statements

###### <Example of source code>

```

void sub(int);
void func(int key) {
    switch(key & 0x3){
    case 0:
        sub(0); break;
    case 1:
        sub(1); break;
    case 2:
        sub(2); break;
    case 3:
        sub(3); break;
    default:
        sub(4); break;
    }
}

```

Since the result of `(key & 0x3)` at the third line will be 0, 1, 2, or 3, optimization in V1.06.00 takes the fact that the condition for the block `default` will never be satisfied into account.

## &lt;Code generated by V1.05.00&gt;

```

_func:
.stack _func = 4
prepare 0x00000001, 0x00000000
andi 0x00000003, r6, r2
cmp 0x00000003, r2
bh9 .BB.LABEL.1_6
.BB.LABEL.1_1: ; entry
shl 0x00000001, r2
jmp #.SWITCH.LABEL.1_7[r2]
.SWITCH.LABEL.1_7:
br9 .BB.LABEL.1_2
br9 .BB.LABEL.1_3
br9 .BB.LABEL.1_4
br9 .BB.LABEL.1_5
.SWITCH.LABEL.1_7.END:
.BB.LABEL.1_2: ; bb
mov 0x00000000, r6
jarl _sub, r31
dispose 0x00000000, 0x00000001, [r31]
.BB.LABEL.1_3: ; bb2
mov 0x00000001, r6
jarl _sub, r31
dispose 0x00000000, 0x00000001, [r31]
.BB.LABEL.1_4: ; bb3
mov 0x00000002, r6
jarl _sub, r31
dispose 0x00000000, 0x00000001, [r31]
.BB.LABEL.1_5: ; bb4
mov 0x00000003, r6
jarl _sub, r31
dispose 0x00000000, 0x00000001, [r31]
.BB.LABEL.1_6: ; bb5
mov 0x00000004, r6
jarl _sub, r31
dispose 0x00000000, 0x00000001, [r31]

```

## &lt;Code generated by V1.06.00&gt;

```

_func:
.stack _func = 4
prepare 0x00000001, 0x00000000
andi 0x00000003, r6, r2
.BB.LABEL.1_1: ; entry
shl 0x00000001, r2
jmp #.SWITCH.LABEL.1_6[r2]
.SWITCH.LABEL.1_6:
br9 .BB.LABEL.1_2
br9 .BB.LABEL.1_3
br9 .BB.LABEL.1_4
br9 .BB.LABEL.1_5
.SWITCH.LABEL.1_6.END:
.BB.LABEL.1_2: ; bb
mov 0x00000000, r6
jarl _sub, r31
dispose 0x00000000, 0x00000001, [r31]
.BB.LABEL.1_3: ; bb2
mov 0x00000001, r6
jarl _sub, r31
dispose 0x00000000, 0x00000001, [r31]
.BB.LABEL.1_4: ; bb3
mov 0x00000002, r6
jarl _sub, r31
dispose 0x00000000, 0x00000001, [r31]
.BB.LABEL.1_5: ; bb4
mov 0x00000003, r6
jarl _sub, r31
dispose 0x00000000, 0x00000001, [r31]

```

## (2) Alias analysis

**<Example of source code>**

```

struct tag1 {
    char member1;
    int member2;
    long long member3;
} StructArray[2];

struct tag2 {
    short index0;
    short index1;
    short index2;
};

void func(struct tag2 *p) {
    StructArray[p->index1].member1 = 1;
    StructArray[p->index1].member2 = 2;
    StructArray[p->index1].member3 = 3;
}

```

Although the address of StructArray[p->index1] would be calculated three times in V1.05.00, it is only calculated once in V1.06.00.

This function is only valid when the -Xalias=ansi option is specified.

**<Code generated by V1.05.00>**

```

_func:
    .stack_func = 0
    ld.h 0x00000002[r6], r2
    shl 0x00000004, r2
    mov #_StructArray, r5
    add r5, r2
    mov 0x00000001, r7
    st.b r7, 0x00000000[r2]
    ld.h 0x00000002[r6], r2
    shl 0x00000004, r2
    add r5, r2
    mov 0x00000002, r7
    st.w r7, 0x00000004[r2]
    ld.h 0x00000002[r6], r2
    shl 0x00000004, r2
    add r2, r5
    mov 0x00000003, r2
    st.w r2, 0x00000008[r5]
    st.w r0, 0x0000000C[r5]
    jmp [r31]

```

**<Code generated by V1.06.00>**

```

_func:
    .stack_func = 0
    ld.h 0x00000002[r6], r2
    shl 0x00000004, r2
    mov #_StructArray, r5
    add r2, r5
    mov 0x00000001, r2
    st.b r2, 0x00000000[r5]
    mov 0x00000002, r2
    st.w r2, 0x00000004[r5]
    mov 0x00000003, r2
    st.w r2, 0x00000008[r5]
    st.w r0, 0x0000000C[r5]
    jmp [r31]

```

#### 4.1.4 Measurement of times for CAN communications

When CS+ is used with the E2 emulator to measure times for CAN communications, the `-insert_dbtag_with_label` option was added. When this option is specified, DBTAG, which is one of the software trace instructions, is generated on the line specified in the editor of CS+.

This option is used by CS+ in measuring times for CAN communications; it is not specified by the user when running a build.

#### 4.1.5 Change to the output of the initial values for array type variables

The output of the initial values for array type variables in assembly source code has been changed so that the values are collectively output on one line. This change reduces build times where it is applicable.

##### <Example of source code>

```
float flt[4] = {1,2,3,4};
```

##### <Code generated by V1.05.00>

```
_flt:
    .dw 0x3F800000 ; float value: 1
    .dw 0x40000000 ; float value: 2
    .dw 0x40400000 ; float value: 3
    .dw 0x40800000 ; float value: 4
```

##### <Code generated by V1.06.00>

```
_flt:
    .dw 0x3F800000,0x40000000,0x40400000,0x40800000 ; float 1,2,3,4
```

#### 4.1.6 Addition of special symbols

A system for reference by C source code generated by the linker at the start and end of a section was added by the inclusion of special replacement symbols.

- Reserved symbols with the values of the addresses where sections start:  
Instances of the character '.' or '@' in symbol names for sections are replaced with '\_', and the string '\_\_S' is added to the beginning.
- Reserved symbols with the value of the first address after the end of a section:  
Instances of the character '.' or '@' in symbol names for sections are replaced with '\_', and the string '\_\_E' is added to the beginning.

For example, the name of the reserved symbol with the value of the start address of the .text section will be '\_\_S\_text'. When the name is referred to by the C source code, it will be '\_S\_text', from which one of the leading underscores has been removed.

##### <Example of source code>

```
extern unsigned long _S_text;

unsigned long* func(void) {
    return &_amp;S_text; /* The address where the .text section starts is acquired. */
}
```

#### 4.1.7 Change to the specification of link map files

The ATTRIBUTE column which is the relocation attribute was added to “Mapping List” of the link map file.

When **-show=relocation\_attribute** is specified, the relocation attribute corresponding to the section is categorized into the four types TEXT, CONST, DATA, and BSS and output. An example of output of the relocation attribute is shown below.

*** Mapping List ***					
SECTION	START	END	SIZE	ALIGN	ATTRIBUTE
.text	00000100	0000013b	3c	2	TEXT
.data	000f0400	000f0403	4	4	DATA
.bss	000f0404	000f040b	8	4	BSS

In addition, output of indications of the types of errors when errors in linkage occur to “Error information” in the link map file has been added.

#### 4.1.8 Specification of the type of end record of Motorola S-type files

A linker option **-end\_record** for specifying the types of end records of Motorola S-type files has been added. In versions earlier than V1.06.00, the end record was output to suit the address of the entry point. In V1.06.00, Motorola S-type files can be generated with specified types of end record.

**-end\_record=record**

For the argument “record”, **S7**, **S8**, or **S9** can be specified.

#### 4.1.9 Addition of numbers to messages when using the evaluation version

Numbers W0561016 and W0561017 were added to the messages that may be output during building by using the evaluation version. This enables control by using **-change\_message** such that the message is handled as an error when the evaluation version is in use.

**W0561016**:The evaluation version is valid for the remaining \*\*\* days

**W0561017**:The evaluation period has expired

#### 4.1.10 Change to the messages for linkage errors

The file name is output in messages for the linkage error "F0563102".

- Versions earlier than V1.06.00

F0563102:Section contents overlap in absolute section "*section name*"

- V1.06.00 and later versions

F0563102:Section contents overlap in absolute section <*section name*> in <*file name*>.

#### 4.1.11 Improvement to the method of authenticating licenses

The way licenses are authenticated was improved to reduce build times.

With this improvement, when a license for the professional edition has not been registered and the code includes #pragma extended language directives for the professional edition, the compiler operates as follows.

- Versions earlier than V1.06.00

- The compiler outputs a warning and ignores the option.

- V1.06.00 and later versions

- When the syntax of the #pragma extended language directive is correct, the compiler outputs a warning but still ignores the option.
- When the syntax of the #pragma extended language directive is not correct, the compiler outputs an error message.

#### 4.1.12 Rectified point for caution

The point for caution on the following item no longer applies. For details, refer to Tool News.

- Point for caution on the #pragma pmodule extended language (No.15)

#### 4.1.13 Other changes and improvements

Other major changes and improvements are described below.

- (a) Correction of a compiler error

The generation of compiler error code "F0530800" in response to specification of the -Xmerge\_files option has been corrected.

- (b) Improved prevention of internal errors

A problem with an internal error during building has been rectified.

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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other disputes 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, drawing, chart, program, algorithm, 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 otherwise misappropriate 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, copy or otherwise misappropriation of Renesas Electronics products.
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; and 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.  
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 (space and 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 third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. When using the 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 radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident 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 Renesas Electronics 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 ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of failure or malfunction of Renesas Electronics products, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, 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 investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use 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 not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
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.  
(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.

(Rev.3.0-1 November 2016)



### 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.

2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130

#### Renesas Electronics Canada Limited

9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3  
Tel: +1-905-237-2004

#### Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: +44-1628-585-100, Fax: +44-1628-585-900

#### Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

#### Renesas Electronics (China) Co., Ltd.

Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

#### Renesas Electronics (Shanghai) Co., Ltd.

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

#### Renesas Electronics Hong Kong Limited

Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2265-6688, Fax: +852-2886-9022

#### Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei 10543, 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 1207, 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 India Pvt. Ltd.

No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India  
Tel: +91-80-67208700, Fax: +91-80-67208777

#### Renesas Electronics Korea Co., Ltd.

12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141