

Contents

Chapter 1. Changes	2
1.1 Improvements to the feature for checking source code against	2
1.2 Addition of a compiler option: <code>-avoid_cross_boundary_prefetch</code>	2
1.3 Addition of a predefined macro for the assembler: <code>__FPU</code>	2
1.4 Specification of the type of end record of Motorola S-type files	2
1.5 Change to the specification of link map files	2
1.6 Change to the specifications of link map files and the output of messages when linkage errors occur	3
1.7 Change to the specifications of messages when the compiler is operating as the evaluation version	3
1.8 Improved license acquisition times	3
1.9 Enhanced optimization	4
1.10 Other improvements	7
Chapter 2. Points for Caution	8
2.1 Note on a case of the W0523041 message [C/C++ Compiler]	8
2.2 Note on using MVTC or POPC instructions [Assembler]	8
2.3 Note on the delete option for linkage [Optimizing linkage editor]	8
Chapter 3. Restrictions	9
3.1 Restriction on usage of <code>math.h</code> functions (<code>frexp</code> , <code>ldexp</code> , <code>scalbn</code> and <code>remquo</code>) in C++ language (including EC++)	9
3.2 Restriction of PIC/PID function (<code>pic</code> and <code>pid</code> options)	11
3.3 Eliminated options (for the C/C++ compiler)	11
3.4 Notes on C/C++ source-level debugging (for the C/C++ compiler)	11
3.5 Note on using sections that include the address <code>0xffffffff</code> (in assembler)	12
3.6 Note on using -form and -output at the same time (in the linkage editor)	12
3.7 Note on using function names that begin with __builtin (for the C/C++ compiler)	12
3.8 Note on using #pragma interrupt with functions for which save_acc is enabled and that have dummy arguments (for the C/C++ compiler)	12
3.9 Restriction of <code>-merge_files</code>	13
Chapter 4. Standard Libraries Included	14
4.1 Library files	14
4.2 Using the library files	15

Chapter 1. Changes

This section describes changes on CC-RX from V2.06.00 to V2.07.00.

The features indicated as **[Professional]** can only be used if the compiler is registered under the professional license.

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

The following rule numbers have been added to the arguments for `-misra2012` option, which allows the compiler to check the source code against MISRA-C:2012 rules.

12.5, 13.1, 13.2, 13.5, 17.5, 17.8, 21.13, 21.15, 21.16

According to Amendment 1 of MISRA-C:2012, the `getenv` function is not checked even if rule 21.8 is specified.

This does not matter for CC-RX, since it does not support the `getenv` function regardless of whether checking of the source code for this is enabled or disabled among the MISRA-C:2012 rules.

1.2 Addition of a compiler option: `-avoid_cross_boundary_prefetch`

When the following string handling library functions are expanded for string manipulation instructions while this option is specified, access to memory is arranged to avoid the crossing of 4-byte boundaries.

`memchr()`, `strlen()`, `strcpy()`, `strncpy()`, `strcmp()`, `strncmp()`, `strcat()`, and `strncat()`

When library functions for string handling are expanded to string manipulation instructions, the `-library=intrinsic` option must be specified at the same time. When `-library=function` is specified, specification of the new option does not take effect.

1.3 Addition of a predefined macro for the assembler: `__FPU`

The macro `__FPU` is usable in assembler-language source code.

When the `-fpu` assembler option is in effect, the value of `__FPU` will be 1.

1.4 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.

1.5 Change to the specification of link map files

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

When **-show=relocation_attribute** is specified, the relocation attribute corresponding to the section is output.

1.6 Change to the specifications of link map files and the output of messages when linkage errors occur

The following error codes and the corresponding error messages are output to the link map file.

In addition to section names, file names are output in the error messages.

F0563100, F0563120, F0563121, F0563122, F0563123, and F0563124

This change simplifies identification of the reasons for linkage errors.

1.7 Change to the specifications of messages when the compiler is operating as the evaluation version

Messages that are output during operation of the evaluation version are handled as warnings.

If no license of the required type can be obtained, the compiler operates as the evaluation version; however, such warnings can easily prevent the creation of load modules in that case.

1.8 Improved license acquisition times

The timing of the acquisition of the professional and standard editions of the license has been changed to reduce the effect on build times. We have also partially adjusted the interpretation of #pragma directives for the professional edition.

- Versions earlier than V2.07.00

When a #pragma directive for the professional edition is used without the required license, the compiler outputs a warning and ignores the #pragma.

- V2.07.00 and later versions

When a #pragma directive for the professional edition is used without the required license and the syntax is correct, the compiler outputs a warning and ignores the #pragma. When the syntax is not correct, the compiler outputs an error message.

1.9 Enhanced optimization

For V2.07.00, optimization has been further enhanced on points (a) and (b), listed and described below.

- (a) Improved handling of code in switch statements that is not generated from `default` blocks
Optimization has been enhanced so that the code is not generated from `default` blocks for which the condition is never satisfied.

```
< source code >
void sub(int);
void func(int key) {
    switch(key & 0x3){ /* The result of " key & 0x03" is 0, 1, 2, or 3. */
        case 0:
            sub(0);
            break;
        case 1:
            sub(1);
            break;
        case 2:
            sub(2);
            break;
        case 3:
            sub(3);
            break;
        default:
            sub(4); /* The assembly code for this is not generated. */
            break;
    }
}
```

```
<V2.06.00 generate code >
_func:
    AND #03H, R1
    BEQ L15
L11: ; entry
    CMP #01H, R1
    BEQ L16
L12: ; entry
    CMP #02H, R1
    BEQ L17
L13: ; entry
    CMP #03H, R1
    BEQ L18
L14: ; bb5
    MOV.L #00000004H, R1
    BRA _sub
L15: ; bb
    MOV.L #00000000H, R1
    BRA _sub
L16: ; bb2
    MOV.L #00000001H, R1
    BRA _sub
L17: ; bb3
    MOV.L #00000002H, R1
    BRA _sub
L18: ; bb4
    MOV.L #00000003H, R1
    BRA _sub
```

```
<V2.07.00 generate code >
_func:
    AND #03H, R1
    BEQ L14
L11: ; entry
    CMP #01H, R1
    BEQ L15
L12: ; entry
    CMP #02H, R1
    BNE L16
L13: ; bb3
    MOV.L #00000002H, R1
    BRA _sub
L14: ; bb
    MOV.L #00000000H, R1
    BRA _sub
L15: ; bb2
    MOV.L #00000001H, R1
    BRA _sub
L16: ; bb4
    MOV.L #00000003H, R1
    BRA _sub
```

(b) Improved precision of alias analysis when the `-alias=ansi` option is specified

The precision of alias analysis has been improved.

```
< 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;
}
```

```
<V2.06.00 generate code >
_func:
    MOV.W 02H[R1], R2
    MOV.L #_StructArray, R3
    SHLL #04H, R2
    MOV.L #00000001H, R14
    MOV.B R14, [R3,R2]
    MOV.W 02H[R1], R2
    SHLL #04H, R2
    ADD R3, R2
    MOV.L #00000002H, 04H[R2]
    MOV.W 02H[R1], R1
    SHLL #04H, R1
    ADD R1, R3
    MOV.L #00000000H, 0CH[R3]
    MOV.L #00000003H, 08H[R3]
    RTS
```

```
<V2.07.00 generate code >
_func:
    MOV.W 02H[R1], R1
    MOV.L #_StructArray, R2
    SHLL #04H, R1
    MOV.L #00000001H, R14
    MOV.B R14, [R2,R1]
    ADD R1, R2
    MOV.L #00000002H, 04H[R2]
    MOV.L #00000000H, 0CH[R2]
    MOV.L #00000003H, 08H[R2]
    RTS
```

1.10 Other improvements

Other major improvements are described below.

(a) Reduced assembly times during compilation

In the assembly source code output by the compiler, the output of the initial values for array type variables in assembly source code has been changed as shown below. This may improve assembly times during compilation.

```
<V2.06.00 generate code >
```

```
.lword 00000000H  
.lword 00000000H  
.word 0001H  
.word 0001H  
.word 0000H  
.word 0000H  
.word 0000H  
.word 0001H  
.word 0000H  
.word 0000H
```

```
<V2.07.00 generate code >
```

```
.lword 00000000H, 00000000H  
.word 0001H, 0001H, 0000H, 0000H  
.word 0000H, 0001H, 0000H, 0000H
```

(b) Correction of a compiler error

The generation of compiler error code "F0530800" in response to specification of the `-merge_files` option has been corrected.

(c) Improved prevention of internal errors

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

Chapter 2. Points for Caution

This section describes points for caution regarding CC-RX.

2.1 Note on a case of the W0523041 message [C/C++ Compiler]

When the `int_to_short` option is specified and a file including a C standard header is compiled as C++ or EC++, the compiler may show the W0523041 message. In this case, simply ignore the message because there are no problems.

[NOTE]

In compilation of C++ or EC++, the `int_to_short` option will be invalid.

Data that are shared between C and C++ (EC++) program must be declared as the long or short type rather than as the int type.

2.2 Note on using MVTC or POPC instructions [Assembler]

In the assembly language, the program counter (PC) cannot be specified for MVTC or POPC instructions.

2.3 Note on the delete option for linkage [Optimizing linkage editor]

When a function symbol is removed by the delete option, its following function in the source program is not allowed to have a breakpoint at its function name on the editor in your debugging. If you would like to set a breakpoint via the Label window at the function entrance, set the breakpoint via the Label window or at the program code of the function.

Chapter 3. Restrictions

This chapter describes restrictions on CC-RX.

3.1 Restriction on usage of math.h functions (frexp, ldexp, scalbn and remquo) in C++ language (including EC++)

An object is generated which will be an infinite-loop at execution when the actual argument of some function (frexp, ldexp, scalbn or remquo) of math.h is int-type, at compiling C++ or EC++ program.

Conditions:

This problem occurs when both (1) and (2) are satisfied.

(1) This program is in C++ or the lang=cpp option is effective.

(2) math.h is included and any of the following functions is called.

- (a) frexp(double, long*) with 'int *' type second argument (except when the first argument is float-type and the dbl_size=8 option is effective).
- (b) ldexp(double, long) with 'int *' type second argument (except when the first argument is float-type and the dbl_size=8 option is effective).
- (c) scalbn(double, long) with 'int *' type second argument (except when the first argument is float-type and the dbl_size=8 option is effective).
- (d) remquo(double, double, long*) with 'int *' type third argument (except when the both the first and second arguments are float-type and the dbl_size=8 option is effective).

Examples:

file.cpp:

```
// Example of compiling C++ source that generates an infinity-loop
#include <math.h>
double d1,d2;
int i;
void func(void)
{
    d2 = frexp(d1, &i);
}
```

Command Line:

```
ccrx -cpu=rx600 -output=src file.cpp
```

file.src: Example of the generated assembly program

```
_func:
    ; ... (Omitted)
    ; Calling substitute function of frexp
    BSR __$frexp__tm__2_f__FZ1ZPi_Q2_21_Real_type__tm__4_Z1Z5_Type
    ; ... (Omitted)

__$frexp__tm__2_f__FZ1ZPi_Q2_21_Real_type__tm__4_Z1Z5_Type:
L11:
    BRA L11 ; Calls itself ==> infinity-loop
```

Countermeasures:

Select one of the following ways to avoid the problem.

- (1) Compile the program with the lang=c or lang=c99 option.
- (2) Change int or int * into long or long *.
- (3) Append the following declarations to each function that is being used.

```
/* For the frexp function */
static inline double frexp(double x, int *y)
{ long v = *y; double d = frexp(x,&v); *y = v; return (d); }
/* For the ldexp function */
static inline double ldexp(double x, int y)
{ long v = y; double d = ldexp(x,v); return (d); }
/* For the scalbn function */
static inline double scalbn(double x, int y)
{ long v = y; double d = scalbn(x,v); return (d); }
/* For the remquo function */
static inline double remquo(double x, double y, int *z)
{ long v = *z; double d = remquo(x,y,&v); *z = v; return (d); }
```

Example of (2):

Change in file.cpp:

```
#include <math.h>
double d1,d2;
int i;
void func(void)
{
    long x = i; /* Accept as long type temporary */
    d2 = frexp(d1, &x); /* Call with long type argument */
    i = x; /* Set the result for variable 'i' */
}
```

Example of (3):

Change in file.cpp:

```
#include <math.h>
/* Append declaration */
static inline double frexp(double x, int *y)
{ long v = *y; double d = frexp(x,&v); *y = v; return (d); }
double d1,d2;
int i;
void func(void)
{
    d2 = frexp(d1, &i);
}
```

3.2 Restriction of PIC/PID function (pic and pid options)

When a standard library is created by the library generator (lbgrx) with the pic or pid option specified, the following warning may appear once or more.

```
W0591301:"-pic" option ignored (When the pic option has been specified)
```

```
W0591301:"-pid" option ignored (When the pid option has been specified)
```

Despite the warning, the created standard library has no problems.

3.3 Eliminated options (for the C/C++ compiler)

(a) `-file_inline`, `-file_inline_path`

Specifying these options has no effect and the compiler will output a warning. Instead of **`-file_inline`** or **`-file_inline_path`**, write **`#include`** in the source code. In case of C and C99, `merge_files` can be used instead.

(b) `-enable_register`

This option is simply ignored and does not affect the generated code.

3.4 Notes on C/C++ source-level debugging (for the C/C++ compiler)

(1) Even when **`-debug`** is specified, you may not be able to set a breakpoint or stop stepped execution on lines that

contain a dynamic initialization expression for a global variable (in C++),

are the first lines of functions that begin with a loop statement (e.g. **`do`** or **`while`**) and do not have an **`auto`** variable or of functions for which **`#pragma inline_asm`** has been specified, or

contain the control section and body of a loop statement (e.g. **`for`**, **`while`**, or **`do`**) written as a single line.

(2) The values of members of union type and of dummy variables that are to be passed via registers may be displayed incorrectly (e.g. in the [Watch] window).

3.5 Note on using sections that include the address 0xffffffff (in assembler)

If two or more **.section** directives in the assembly source code contain **.org** directives, the sections have the same name, and the sections overlap at 0xffffffff, the assembler outputs an internal error message (C0554098).

Example)

```
.section SS,ROMDATA
.org 0fffffffh
.byte 1
.byte 2 ; 0xffffffff
.section SS,ROMDATA
.org 0fffffffh
.byte 3; ; 0xffffffff
.end
```

3.6 Note on using **-form** and **-output** at the same time (in the linkage editor)

When **-form=rel** and **-output=<filename>** are specified for the linkage editor (**rlink**) at the same time, the filename extension given as **<filename>** is ignored and replaced with **.rel**.

Example)

```
rlink -form=relocate -output=DefaultBuild\lib_test.lib
```

The filename specified for output, **test.lib**, is changed to **test.rel**.

3.7 Note on using function names that begin with **_builtin** (for the C/C++ compiler)

Declaration of a function with a name that begins with **_builtin** and for which the definition is in **machine.h** in the **include** directory may lead to an internal error. In general, do not use any names that begin with an underscore (**_**) in your source code, since such names are reserved.

3.8 Note on using **#pragma interrupt** with functions for which **save_acc** is enabled and that have dummy arguments (for the C/C++ compiler)

When **#pragma interrupt** is specified for a function and the **save_acc** flag is enabled (including where this is done by using the **-save_acc** compiler option), the compiler may not output code that reflects the correct values of dummy arguments which are passed via R4. Note: In general, we do not recommend defining arguments for functions with the **#pragma interrupt** specification.

3.9 Restriction of `-merge_files`

Under certain conditions, compilation with `-merge_files` or `-whole_program` specified as the translation unit of code that includes union-type variables will produce error code F0530800 or warning code W0530811.

[Conditions]

If all the following conditions are satisfied, error code F0530800 or warning code W0530811 will be produced.

- (1) `-merge_files` or `-whole_program` is specified.
- (2) A union-type external variable having two or more members has been initialized outside any function, and, other than the member that have been initialized, a member has an alignment and size larger than the other member or members.
- (3) The variable described in (2) above is declared as extern for reference by either of the following.
 - (3-1) Source files other than the definition of external variable described in (2) exists.
 - (3-2) Header files included directly or indirectly by the source files other than the definition of external variable described in (2) exists.

[Workarounds]

Take any of the following steps.

- (1) Specify neither of the options in condition (1).
- (2) Initialize the union-type external variable described in condition (2) within a function.
- (3) Refer to the variables corresponding to condition (2) only in the source file that includes the definition of the external variable.

Chapter 4. Standard Libraries Included

This chapter describes restrictions on standard libraries included in RX Family C/C++ Compiler.

This compiler package includes four library files (*.lib) for the RX600. You can use any of the library files if they correspond to the options that you wish to specify. Using these files shortens the time required for building.

4.1 Library files

Table 4.1 shows the standard library files and compiler options.

Note:

The compiler options you specify should be the same as the microcontroller options defined for each of the library files listed in table 4.1. Otherwise these library files are not usable, so specify your compiler options in the library generator to generate your own library file.

Library File	Purposes	Optimize ^{*2} Options	Microcontroller Options ^{*1 *2}		
			-endian	-cpu -rtti -exception -noexception	Others ^{*3}
rx600lq.lib	For the RX600 Optimization type:Speed Little endian	-speed -goptimize	-endian=little	-cpu=rx600	-round=nearest -denormalize=off -dbl_size=4 -unsigned_char -unsigned_bitfield -bit_order=right -unpack -fint_register=0 -branch=24
rx600ls.lib	For the RX600 Optimization type:Size Little endian	-size -goptimize			
rx600bq.lib	For the RX600 Optimization type: Speed Big endian	-speed -goptimize	-endian=big	-rtti=on -exception	
rx600bs.lib	For the RX600 Optimization type: Size Big endian	-size -goptimize			

Table 4.1 Library Files

*Notes:

*1 For details on microcontroller options, please see the “Microcontroller Options” columns of the “(1) Compile Options” of “section A.1.3 Options”, in the Integrated Development Environment User’s Manual:RX Build.

*2 These option selections are same from the each default of them.

4.2 Using the library files

The library files included in the compiler package must be linked in either of the ways given in sections 4.2.1 and 4.2.2.

4.2.1 Using the library files

When the e² studio has been installed in C:\Renesas\e2_studio, the library files are stored in the following location:

```
C:\Program Files\Renesas\RX\V2_7_0\lib
```

("V2.07.00" indicates the version and revision number of the compiler package.)

4.2.2 Directory specifying a library file in the optimizing linkage editor

Copy the library file(s) included in the package (stored in the location given in section 4.2.1) into a desired directory. Then specify one of the copied library files for the Library option and start the linkage processing.

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