

---

# H8S/2200 Series

## Usage of PC Break Controller

---

### Introduction

This application uses the address break generation function of the PC break controller, and when a preset condition is satisfied, LEDs connected to the H8S/2238 are lit on to confirm it.

### Target Device

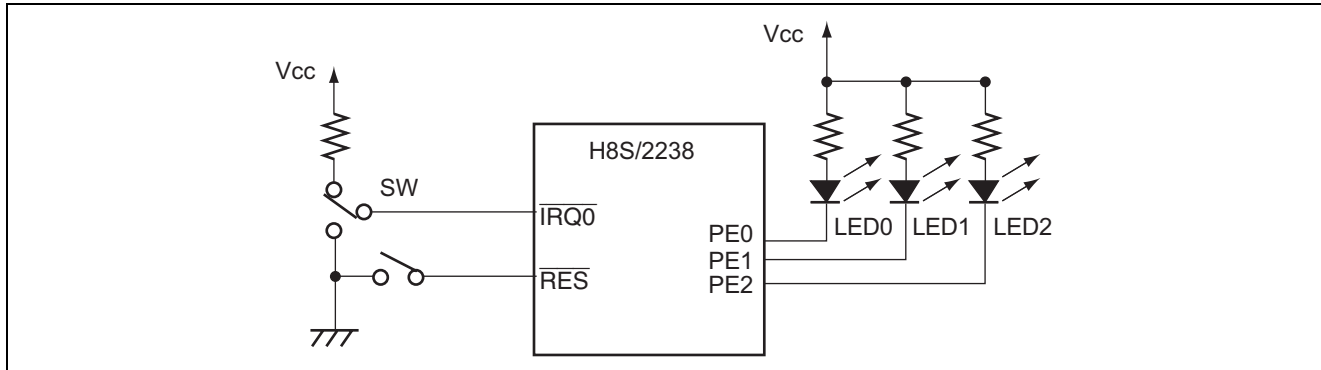
H8S/2238B, H8S/2238R

### Contents

1. Specifications .....	2
2. Description of Functions .....	3
3. Principle of Operation .....	4
4. Description of Software .....	6
5. Flowchart.....	8

### 1. Specifications

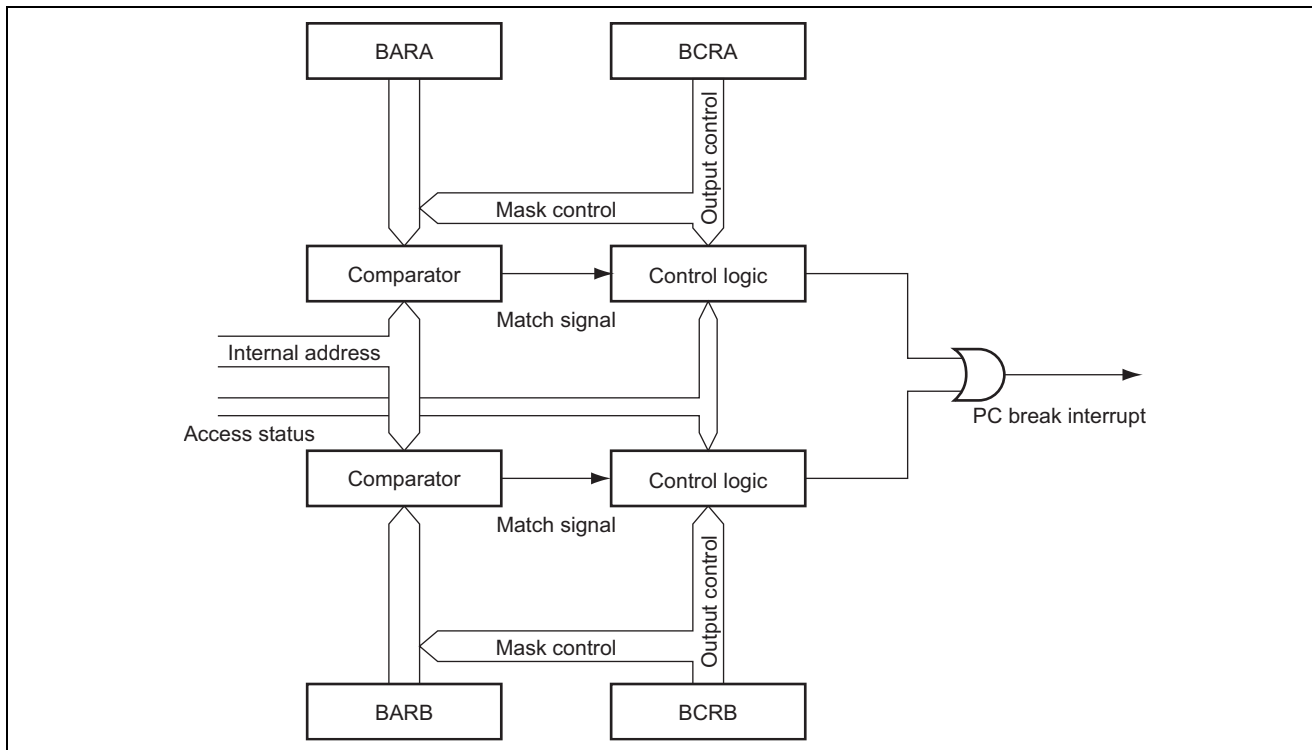
1. As shown in figure 1, an address break is generated by the PC break controller and it is confirmed by lighting certain LEDs on.
2. When power is supplied or the reset switch (pin 59) is pressed, LED0 is lit on, and a break address is set.
3. By inputting a low level signal to the IRQ0 pin (pin 38), processing shifts to the "IRQ0 interrupt" routine, FLAG = 1 is set, and LED1 is lit on.
4. The processing to light LED2 on is branched by setting FLAG to 1, then FLAG is reset to 0. Because an address break is set to the instruction which lights on LED2, when this address is executed, processing shifts to the "pbc routine", and the "PBCtlset" is transmitted to the PC monitor for display.



**Figure 1 Usage Example of PC Break Controller Function**

### 2. Description of Functions

- Figure 2 shows a block diagram of the PC break controller, and the following is the description of the PC break controller registers:
  - The break address register A (BARA) is a 32-bit readable/writable register to set a break address for channel A.
  - The break control register A (BCRA) controls the PC break for channel A.
  - The break address register B (BARB) is a break address register for channel B. The bit configuration is the same as that of BARA.
  - The break control register B (BCRB) is a break control register for channel B. The bit configuration is the same as that of BCRA.



**Figure 2 Block Diagram of PC Break Controller**

- Table 1 shows the assignment of functions used in this sample task.

**Table 1 Assignment of Functions**

Elements	Description
BARA	Sets a PC break address within the range specifiable with break address bits 23 to 0 (H'0010DC is set in these bits).
BDRA	Sets break address mask to validate all bits from 23 to 0, selects instruction fetch as the break condition, and enables PC break interrupt requests.
BARB	PC break address register for channel B (unused).
BCRB	Register for control of PC break set with BARB (unused).

### 3. Principle of Operation

Figure 3 describes the PC break controller interrupt operations of this sample task. Software processing and indication on the monitor are shown in the figure.

1. When the power is turned on or on a system reset, the main routine is executed, and LED0 is lit on by setting the 0 bit of PEDR (port E register) to "0". H'001078 is set within the program as a break address. In this sample task, an address for the location where the first byte of an instruction exists is specified for an instruction fetch break.
2. By inputting a low level signal to the IRQ0 pin (pin 38), IRQ0 interrupt occurs, processing shifts to the "irq0int" routine, the 1 bit of PEDR is set to "0" to light LED1 on. Also, FLAG = 1 is set, and IRQ0 interrupt is reset to return to the main routine.
3. The operation to light LED2 (where bit 2 of PEDR is set to "0") on is branched by setting FLAG = 1, and FLAG = 0 is reset. Because an address break is set to where FLAG = 0, an address break occurs by executing this address.
4. When an address break occurs, processing shifts to the "pbc routine", and "PBCtlret" is displayed on the PC indicating the end of operation.

Note: A break address value can be found by setting the HEW as follows and executing a build to compile the C program file:

Operation on HEW:

Select "Option > Renesas H8S/2200 Standard Toolchain" from the menu and make the following settings for "C/C++ tag":

Category: Select "List".

Generate list file: Select this checkbox.

Contents: Settings in the "Contents" are enabled. Select the "Object list" checkbox.

By executing a build after the above settings are made, a file is created in the following folder, from which you can obtain the address.

Folder Location:

The relative addresses for the C program are output to "xxxx.lst" in the "main > Debug" folder.

(Note: xxxx is a file name of C program that is set as desired.)

The C program start address is a value indicated at the START part of section P in "main map" in the "Debug" folder. A break address is calculated by adding the start address to the relative address of "xxxx.lst". In this sample task, the break address is H'0010DC. ("main map" is created by default.)

(Start address in "main map") + (Relative address in "xxxx.lst") = Break address value

### (1) Initial settings

Setting of port E and the address break in the main routine

Initial settings	
PEDDR=0xFF;	(Set port E to output)
PEDR=0xFE;	(Light LED bit 0)
:	
BARA=0x00106E;	(Set the break address)
BCRA=0x01;	(Enable interrupts by the break address)

### (2) Execution and state of the main routine

The main routine waits for the change of FLAG state.

Address	Instructions
:	
L69	
00001070	MOV.B @_FLAG:32,R0L
00001076	BEQ L69:8
:	

Note: The Address indicates the total of the relative and start addresses.

### (3) IRQ interrupt operation

The IRQ interrupt causes the program to branch to the irq0int routine, FLAG is set to 1, LED1 is lit, and the program returns to the main routine.

Address	Instructions
:	
00001078	MOV.B #-5,R0L ←*
0000107A	MOV.B R0L,@16776973:8
:	

\*: The address set in BARA and the execution address match.

### (4) Operation of pbc routine by the address break interrupt

In the main routine, by branching to the LED2 lighting routine, the address moves to 1078 and an interrupt occurs. The interrupt causes the program to move to the pbc routine, PBCtlset is sent to the PC, and it is displayed on the monitor.

void irq0int(void)	
Address	Instructions
:	
000010F8	MOV.W @SP,R0
000010FA	EXTU.L ERO
000010FC	MOV.B @(_STD2:32, ERO),R0L
00001104	MOV.B R0L,@16777099:8
:	

Display on the monitor screen

PBCtlset
----------

**Figure 3 Operation of PC Break Controller**

## 4. Description of Software

### 4.1 Modules

Table2 describes the modules used in this sample task.

**Table 2 Description of Modules**

Module	Label	Function
Main routine	main	Sets a break address, sets the 0 bit of port E to 0 to light LED0 on, and waits for FLAG = 1.
IRQ0 interrupt routine	irq0int	By inputting a low level signal to the IRQ0 pin, this routine is executed to set FLAG = 1. Also, LED1 is lit on .
PC break controller interrupt routine	pbc	When FLAG = 1 is set, processing branches to that of LED1, an address break interrupt is executed, and this routine is called to light LED2 on. Then, "PBctlset" is sent to the PC monitor to be displayed.

### 4.2 Arguments

This sample program does not use arguments.

### 4.3 Internal Registers

The internal registers used in this sample task are described in table 3.

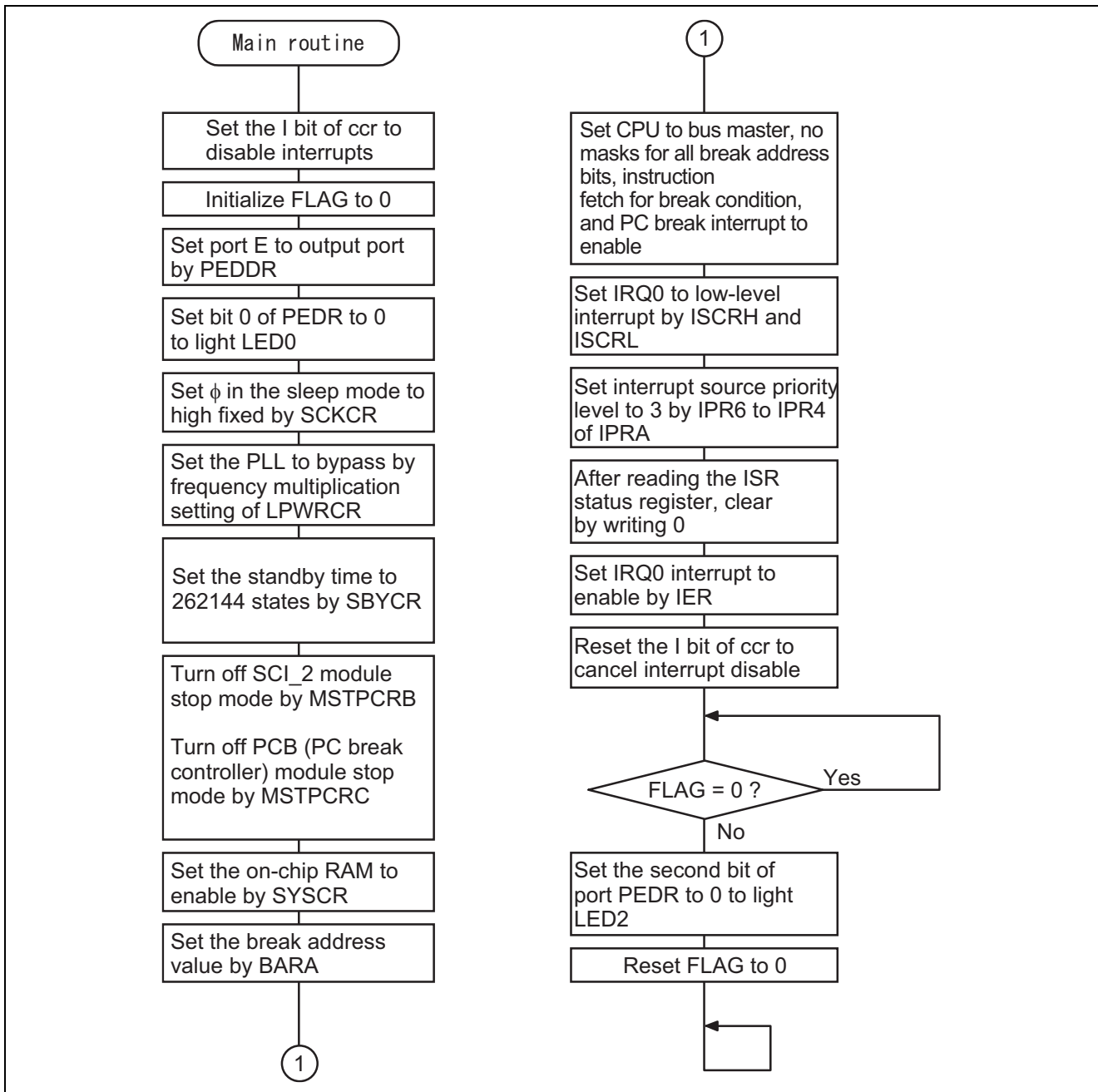
**Table 3 Description of Internal Registers**

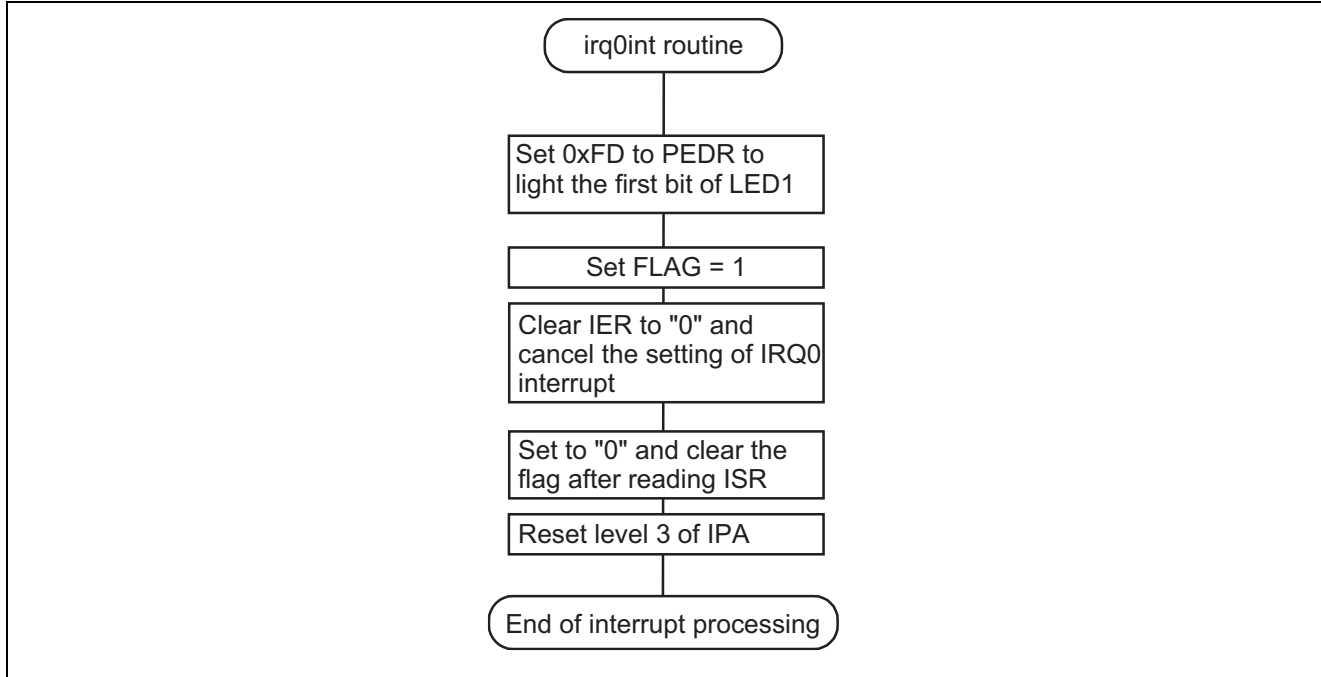
Register	Function	Address	Setting	
BARA	—	Break Address Register A (Reserved) The read value is undefined. These bits cannot be modified.	H'FFFE00 Bits 31 to 24	Undefined
	BAA23 to BAA0	Break Address Register A (Break Addresses 23 to 0) Sets a break address for channel A.	H'FFFE00 Bits 23 to 0	0010DC
BARB	—	Break Address Register B (Reserved) The read value is undefined. These bits cannot be modified.	H'FFFE04 Bits 31 to 24	Undefined
	BAB23 to BAB0	Break Address Register B (Break Addresses 23 to 0) Sets a break address for channel B.	H'FFFE04 Bits 23 to 0	000000
BCRA	CMFA	Break Control Register A (Condition Match Flag A) CMFA = 0 indicates that the break condition set for channel A is not satisfied or 0 has been written to this bit after 1 was read from it. CMFA = 1 indicates that the break condition set for channel A is satisfied.	H'FFFE08 Bit 7	0
	CDA	Break Control Register A (CPU Cycle/DTC Cycle Select A) When CDA = 0, the CPU is selected as the bus master. When CDA = 1, the CPU or DTC/DMAC is selected as the bus master. (DMAC is only supported by the H8S/2239 series)	H'FFFE08 Bit 6	0

Register	Function	Address	Setting	
BCRA	BAMRA2 to BAMRA0	Break Control Register A (Break Address Mask Register A2 to A0)	H'FFFE08 Bits 5 to 3	000
	BAMRA0	When BAMRA2 to BAMRA0 = 000, bits BAA23 to BAA0 of the break address set in BARA is valid. When BAMRA2 to BAMRA0 = 001 to 111, lower bits are masked. (For details, refer to the hardware manual.)		
CSELA1, CSELA0	CSELA1, CSELA0	Break Control Register A (Break Condition Select)	H'FFFE08 Bits 2 and 1	00
		When CSELA1 and CSELA0 = 00, instruction fetch is selected. When CSELA1 and CSELA0 = 01, a data read cycle is selected. (Description for CSELA1 and CSELA0 = 10 or 11 is omitted.)		
BIEA	Break Control Register A (Break Interrupt Enable)	H'FFFE08 Bit 0	1	
	When BIEA = 0, PC break interrupt requests are disabled for channel A. When BIEA = 1, PC break interrupt requests are enabled for channel A.			
BCRB	CMFB	Break Control Register B (Condition Match Flag B)	H'FFFE09 Bit 7	0
		Same as BCRA		
CDB	Break Control Register B (CPU Cycle/DTC Cycle Select B)	H'FFFE09 Bit 6	0	
	Same as BCRA			
BAMRB2 to BAMRB0	BAMRB2 to BAMRB0	Break Control Register B (Break Address Mask Register B2 to B0)	H'FFFE09 Bits 5 to 3	000
		Same as BCRA		
CSELB1, CSELB0	CSELB1, CSELB0	Break Control Register B (Break Condition Select)	H'FFFE09 Bits 2 and 1	00
		Same as BCRA		
BIEB	Break Control Register B (Break Interrupt Enable)	H'FFFE09 Bit 0	0	
	Same as BCRA			

5. Flowchart

5.1 Main routine



**5.2 IRQ0 interrupt routine**



#### 5.4 Link Addresses

Section Name	Address
CV1	H'00000000
CV2	H'00000040
CV3	H'0000006C
P, D	H'00001000
B	H'00FFB000

## Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Mar.16.04	—	First edition issued
2.00	Jul.30.04		Second edition issued
		2	1. Specifications
		4-5	3. Principle of Operation
		6	4.1 Modules
		8-10	5. Flowchart

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.  
The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.  
Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.  
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.