RX610 Group

Vector Table Address Switching

R01AN0203EJ0100
Rev.1.00
Dec 09, 2010

Abstract

This application note presents a sample program showing the method for switching the relocatable vector table address and interrupt usage.

Target Device

RX610 Group

Introduction

This application note was compiled using as a basis RX610 Group MCUs, on which operation has been confirmed.

The program can be used with other RX Family MCUs that have the same I/O registers (peripheral device control registers) as the RX610 Group. Check the latest version of the manual for any additions and modifications to the functions or address map used by this application note. Careful evaluation is recommended before using this application note.

The program works with an endian specification of big or little and with left or right specified as the bit order.

Contents

1. Overview

RX610 MCUs have a function that allows a relocatable vector table to be assigned to a user-defined address. The address assigned to the relocatable vector table can be changed by setting the interrupt table register (INTB).

In this example, both the ROM and RAM areas are used to store relocatable vector tables and interrupt handlers, and switching between the vector table in ROM and the vector table in RAM is accomplished by setting the interrupt table register.

Note: The program in this application note makes changes to the interrupt table register and enables/disables interrupts. In order to make changes to the interrupt table register and enable or disable interrupts, it is necessary to first set the processor mode to supervisor mode. When the MCU is in user mode, writes to the interrupt table register and to the interrupt enable bit (I) in the processor status word (PSW) register are ignored.

Note that the initial value of the interrupt table register is undefined, so interrupts must be disabled when initial settings are performed.

It is also recommended that after the initial settings of the interrupt table register have been changed, subsequent changes be made with interrupts disabled.

2. Operation

2.1 Interrupt Table Register Switching

In this example, both the ROM and RAM areas are used to store relocatable vector tables and interrupt handlers. The interrupt handler in ROM is used for the vector table in ROM, and interrupt handler in RAM is used for the vector table in RAM. Switching between the vector tables is accomplished by changing the interrupt table register (INTB).

Figure 1 shows an example of interrupt table register switching. In figure 1 the address of the vector table in ROM is H'FFFF8100 and the address of the vector table in RAM is H'00010000.

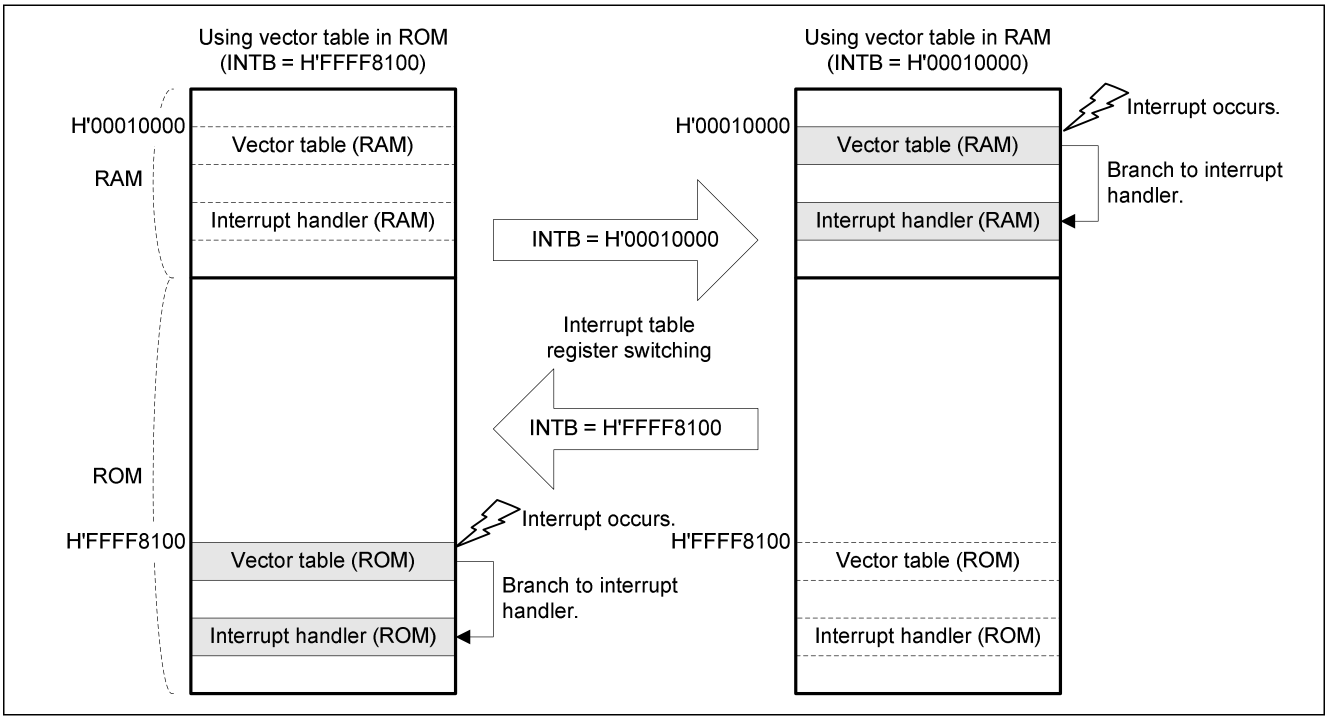


Figure 1 Interrupt Table Register Switching Example

2.2 Operation Overview

Figure 2 shows an overview of the operation of the sample program.

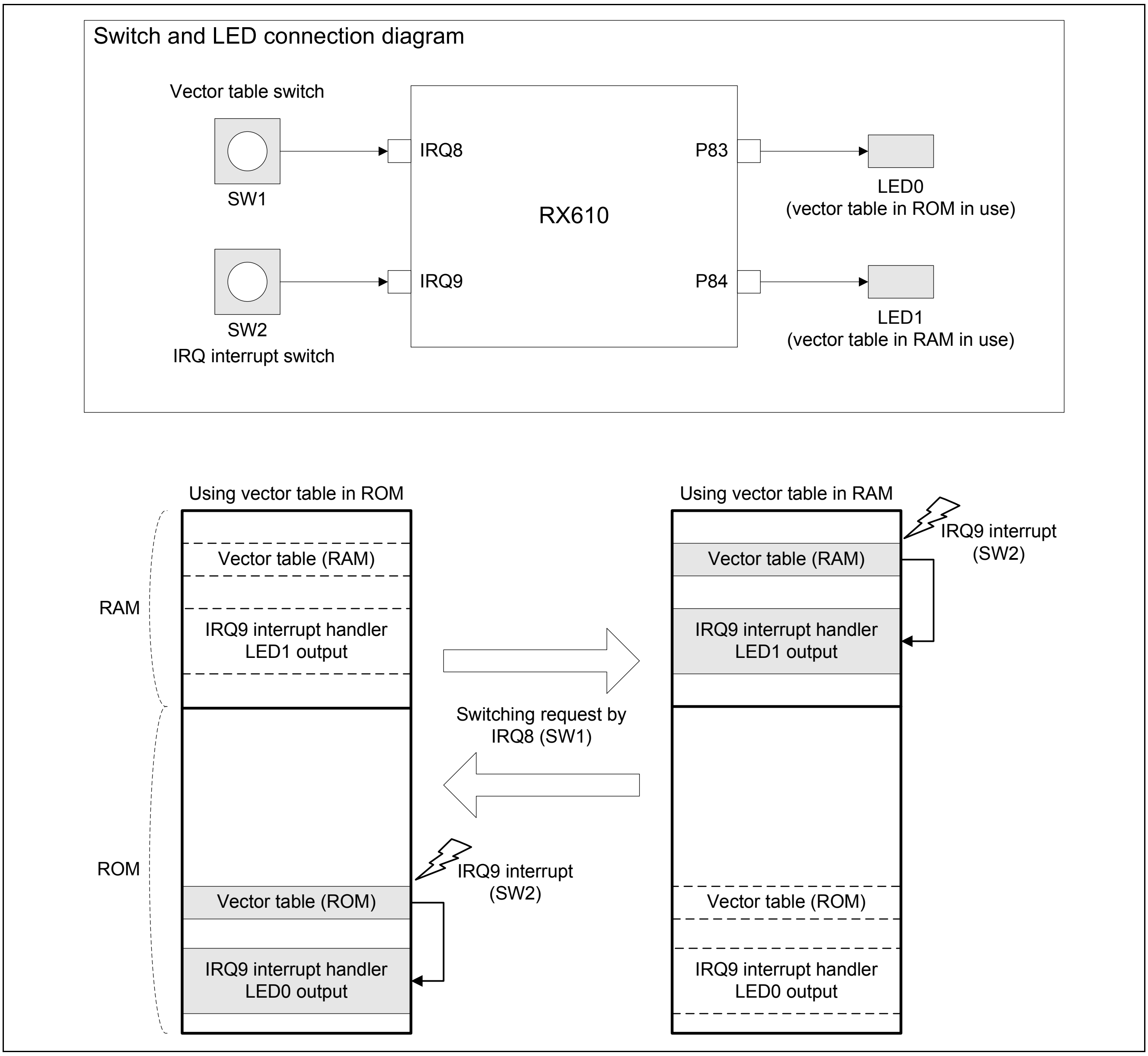


Figure 2 Operation Overview

In the sample program, IRQ8 and IRQ9 are used as interrupt sources. The IRQ8 pin is connected to SW1 and the IRQ9 pin to SW2 on the evaluation board.

Pressing SW1 generates an IRQ8 interrupt. When an IRQ8 interrupt occurs, a request to switch the interrupt table register is issued. In the main process block, the switching request is accepted and the interrupt table register is switched, causing the vector table and interrupt handler to be switched as well.

Pressing SW2 generates an IRQ9 interrupt. When an IRQ9 interrupt occurs, the interrupt handler in ROM or the interrupt handler in RAM is executed, according to the setting of the interrupt table register. When the interrupt handler in ROM is executed, the output to LED0 on the evaluation board is inverted, and the output to LED1 is inverted when the interrupt handler in RAM is executed.

2.3 Operation Sequence

Figure 3 shows the operation sequence of the main process of the sample program. Figure 4 shows the operation sequence of the interrupt handler.

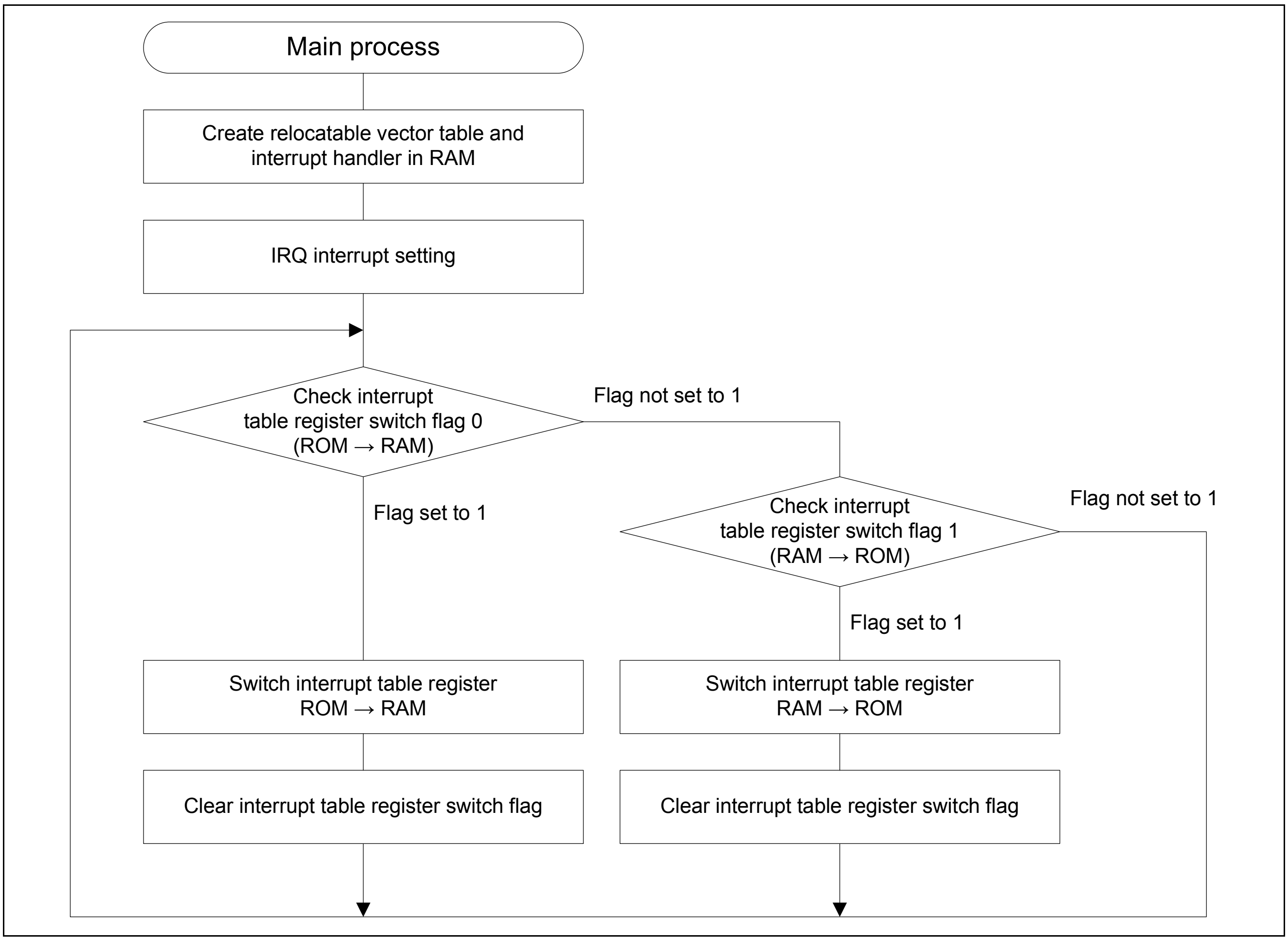

Main process
Create relocatable vector table and interrupt handler in RAM
IRQ interrupt setting
Check interrupt table register switch flag 0 (ROM → RAM)
Flag not set to 1
Flag set to 1
Check interrupt table register switch flag 1 (RAM → ROM)
Flag not set to 1
Flag set to 1
Switch interrupt table register ROM → RAM
Switch interrupt table register RAM → ROM
Clear interrupt table register switch flag
Clear interrupt table register switch flag

Figure 3 Main Process Operation Sequence

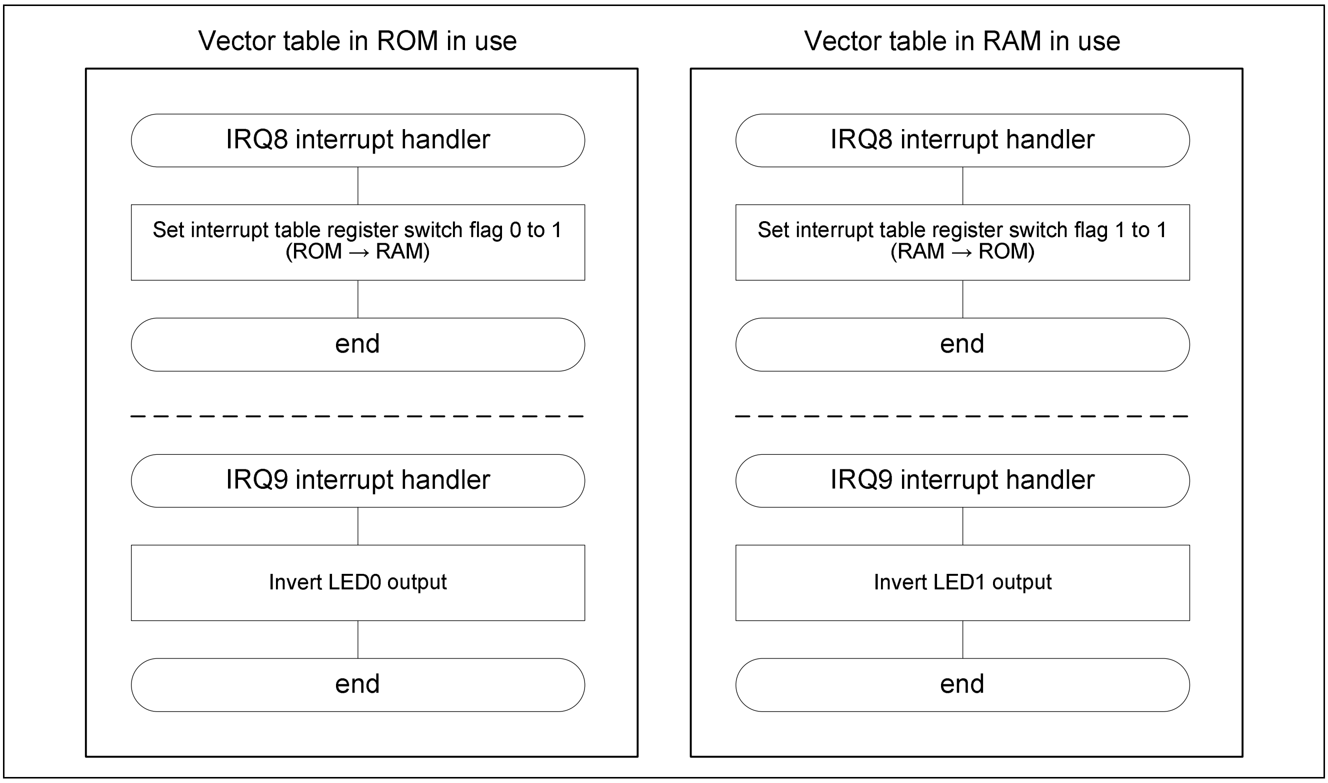

Vector table in ROM in use
IRQ8 interrupt handler
Set interrupt table register switch flag 0 to 1 (ROM → RAM)
end
IRQ9 interrupt handler
Invert LED0 output
end
Vector table in RAM in use
IRQ8 interrupt handler
Set interrupt table register switch flag 1 to 1 (RAM → ROM)
end
IRQ9 interrupt handler
Invert LED1 output
end

Figure 4 Interrupt Handler Operation Sequence

The sample program uses interrupt table register switch flags. There are two interrupt table register switch flags: one requests a change from the vector table register in ROM to the one in RAM and the other from the vector table register in RAM to the one in ROM.

The main process executes a test loop that repeatedly checks the status of the interrupt table register switch flags. When a flag is set, it changes the interrupt table register and clears the flag.

The IRQ8 interrupt handlers set the respective interrupt table register switch flags. The IRQ8 interrupt handler in ROM sets interrupt table register switch flag 0 (ROM → RAM). The IRQ8 interrupt handler in RAM sets interrupt table register switch flag 1 (RAM → ROM).

The IRQ9 interrupt handlers manipulate the output to the respective LEDs, indicating whether the interrupt handler in ROM or the interrupt handler in RAM has been executed. The IRQ9 interrupt handler in ROM inverts the output to LED0. The IRQ9 interrupt handler in RAM inverts the output to LED1.

3. Program Description

3.1 File Structure

Table 1 shows the file structure. In addition to the files listed in table 1, some files generated automatically by HEW are used as well.

Table 1 File Structure

File Name	Description
resetprg.c *[1]	Initial settings process
main.c	Main process
rom_intprg.c	ROM area interrupt handler
ram_intprg.c	RAM area interrupt handler
ram_vecttbl.c	RAM area relocatable vector table definition
dbsct.c*[2]	Definition of sections to be copied from ROM area to RAM area
vvect.h	IRQ8 and IRQ9 interrupt handler function declarations
vect.h*[3]	Interrupt handler function declarations for interrupts other than IRQ8 and IRQ9

Notes: 1. This file is generated automatically by HEW, but it is necessary to comment out the processor mode switching portion of the PowerON_Reset_PC function. For details, see 3.3.1, PowerON_Reset_PC Function.

2. This file is generated automatically by HEW, but it is necessary to the add RAM area relocatable vector table and interrupt handler sections.

3. This file is generated automatically by HEW, but it is necessary to comment out the interrupt handler function declarations for IRQ8 and IRQ9.

3.2 Variable Structure

Table 2 lists the variables used.

Table 2 List of Variables

Variable Name	Type	Description
chg_intb_flag0	unsigned char	• Interrupt table register switch flag 0 (ROM → RAM) Set to 1 by the ROM area IRQ8 interrupt handler. Cleared to 0 when the interrupt table register in RAM is switched to.
chg_intb_flag1	unsigned char	• Interrupt table register switch flag 1 (RAM → ROM) Set to 1 by the RAM area IRQ8 interrupt handler. Cleared to 0 when the interrupt table register in ROM is switched to.

3.3 Function Structure

The specifications of the functions are described below.

Table 3 List of Functions

Function Name	File Name	Description
PowerON_Reset_PC *	resetprg.c	Initial settings after reset function
main	main.c	Main function
init	main.c	MCU initial settings function, operating clock setting
changeINTB	main.c	Changes interrupt table register (INTB)
set_IRQ	main.c	IRQ8 and IRQ9 interrupt settings
Rom_Int_IRQ8	rom_intprg.c	IRQ8 interrupt handler function (ROM area)
Rom_Int_IRQ9	rom_intprg.c	IRQ9 interrupt handler function (ROM area)
Ram_Int_IRQ8	ram_intprg.c	IRQ8 interrupt handler function (RAM area)
Ram_Int_IRQ9	ram_intprg.c	IRQ9 interrupt handler function (RAM area)

Note: * This function is generated automatically by HEW, but it is necessary to comment out the processor mode switching portion. For details, see 3.3.1, PowerON_Reset_PC Function.

3.3.1 PowerON_Reset_PC Function

(1) Functional overview

The PowerON_Reset_PC function is called after a reset. It uses embedded functions and a standard library function to make settings to CPU registers. During this process, the _INITSCT function copies the relocatable vector table and interrupt handler functions from a predetermined ROM area to a RAM area. Then the main function is called.

Note: This function is generated automatically by HEW. In its initial form, it switches the processor mode to user mode. However, writes to the interrupt enable bit (I) in the processor status word (PSW) register are ignored in user mode, making it impossible to change the enabled/disabled state of interrupts. The processor mode switching portion of the PowerON_Reset_PC function must therefore be commented out for use with the sample program.

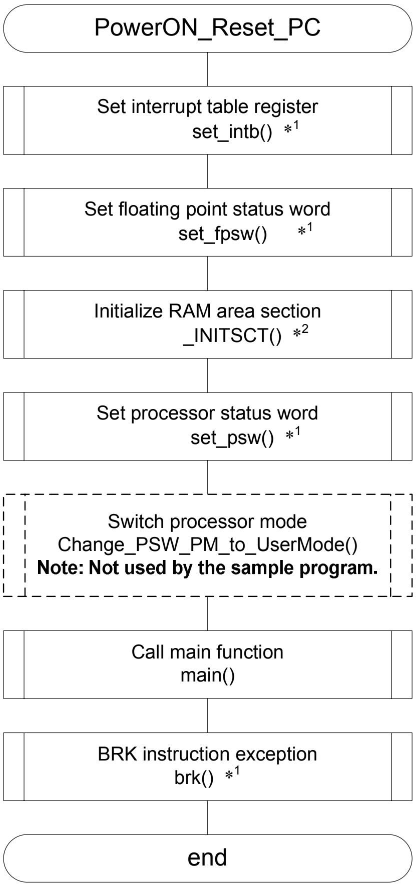
(2) Arguments

None

(3) Return values

None

(4) Flowchart


PowerON_Reset_PC
Set interrupt table register
set_intb() *1
Set floating point status word
set_fpsw() *1
Initialize RAM area section
_INITSCT() *2
Set processor status word
set_psw() *1
Switch processor mode
Change_PSW_PM_to_UserMode()
Note: Not used by the sample program.
Call main function
main()
BRK instruction exception
brk() *1
end

Notes: 1. Embedded function (set_intb, set_fpsw, set_psw, and brk functions)
2. Standard library function (_INITSCT function)
For details, see “Development Environment Manual” in 5, Reference Documents.

Figure 5 Flowchart (PowerON_Reset_PC Function)

3.3.2 main Function

(1) Functional overview

The main function calls init and set_IRQ to make interrupt settings. Then it starts a test loop to receive interrupts. The test loop checks the interrupt table register switch flags. When a flag is set by the IRQ8 interrupt, the changINTB function is called to switch the vector table address.

(2) Arguments

None

(3) Return values

None

(4) Flowchart

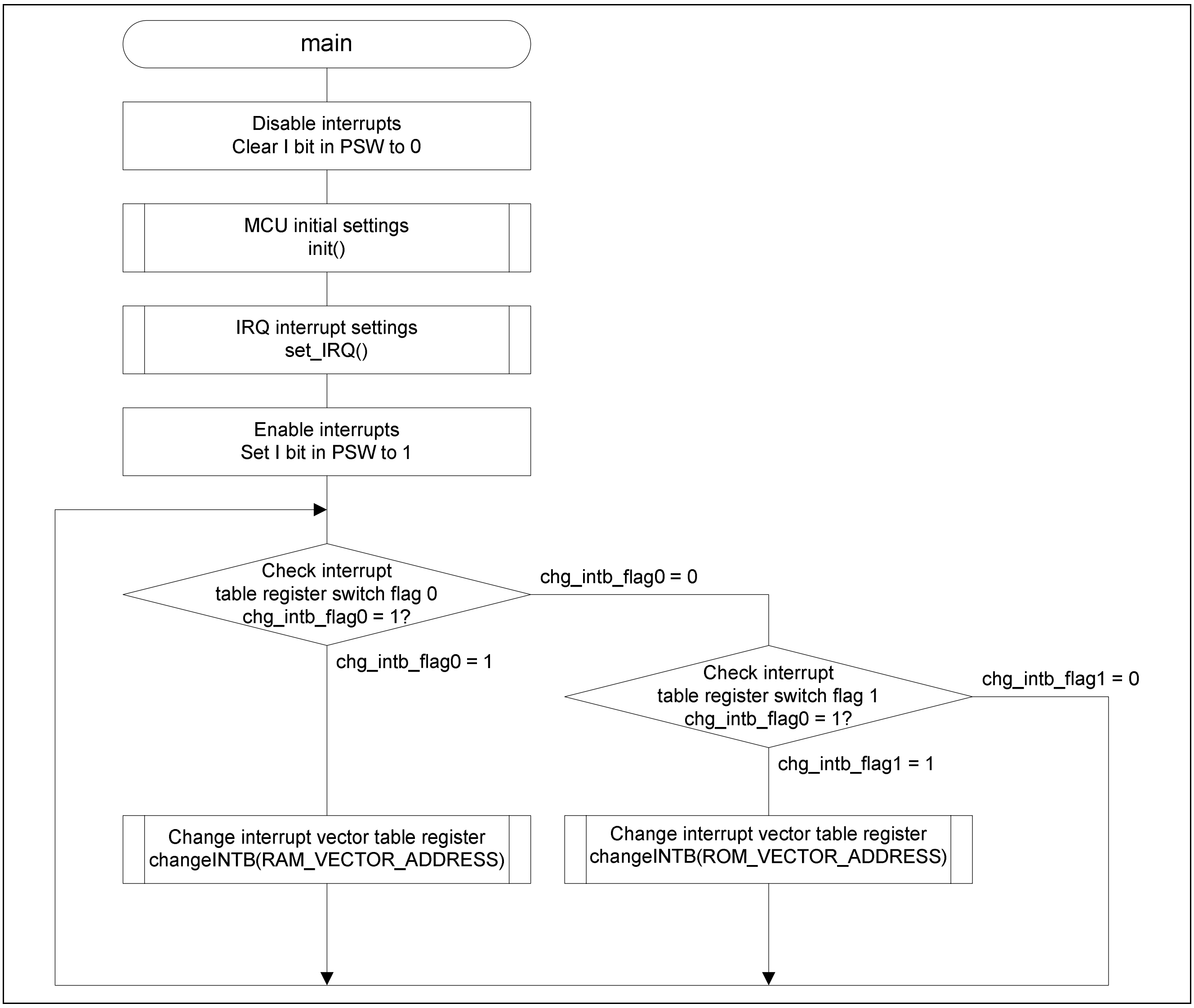


Figure 6 Flowchart (main Function)

3.3.3 init Function

(1) Functional overview

The init function performs initial program settings, setting the operating clock, setting the output ports, and initializing the interrupt table register switch flags.

(2) Arguments

None

(3) Return values

None

(4) Flowchart

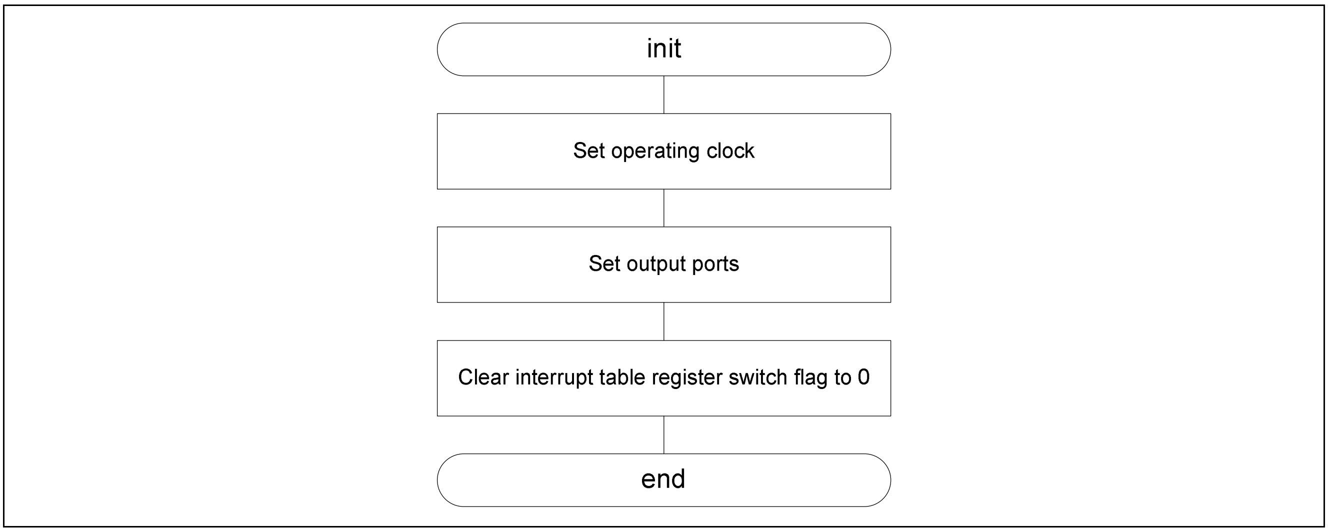


Figure 7 Flowchart (init Function)

3.3.4 changeINTB Function

(1) Functional overview

The changeINTB function changes the interrupt table register (INTB). Note that in the sample program the INTB change takes place while interrupts are disabled.

The changeINTB function transitions to the interrupts-disabled state and changes the INTB. Then it clears LED output, clears the interrupt table register switch flag, and re-enables interrupts. An embedded function is used to change the INTB.

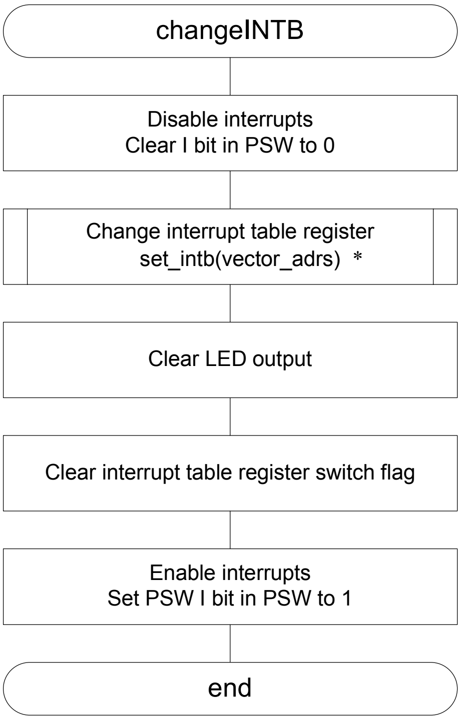
(2) Arguments

Argument	Type	Description
vector_adrs	unsigned long	Interrupt table address

(3) Return values

None

(4) Flowchart



Note: * Embedded function (set_intb function)
For details, see “Development Environment Manual” in 5, Reference Documents.

Figure 8 Flowchart (changeINTB Function)

3.3.5 set_IRQ Function

(1) Functional overview

The set_IRQ function makes input port and IRQ interrupt settings, and it enables IRQ interrupts.

(2) Arguments

None

(3) Return values

None

(4) Flowchart

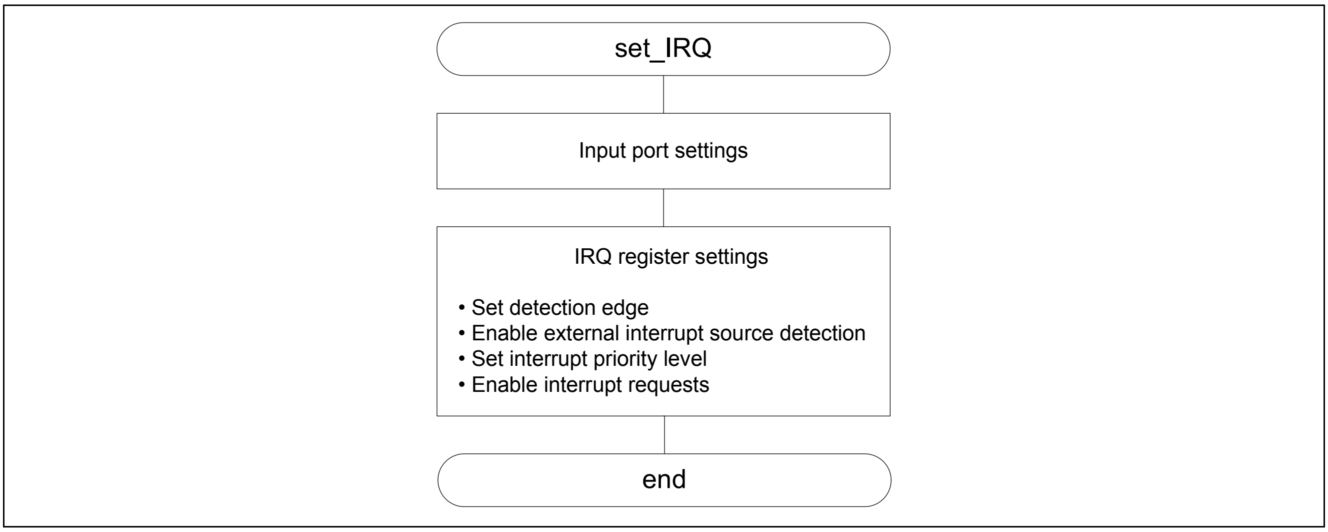

set_IRQ
Input port settings
IRQ register settings
• Set detection edge
• Enable external interrupt source detection
• Set interrupt priority level
• Enable interrupt requests
end

Figure 9 Flowchart (set_IRQ Function)

3.3.6 Rom_Int_IRQ8 Function

(1) Functional overview

The Rom_Int_IRQ8 function is the IRQ8 interrupt handler function that indicates the relocatable vector table in the ROM area is in use. It sets interrupt table register switch flag 0 to 1 and returns control to the main function.

(2) Arguments

None

(3) Return values

None

(4) Flowchart

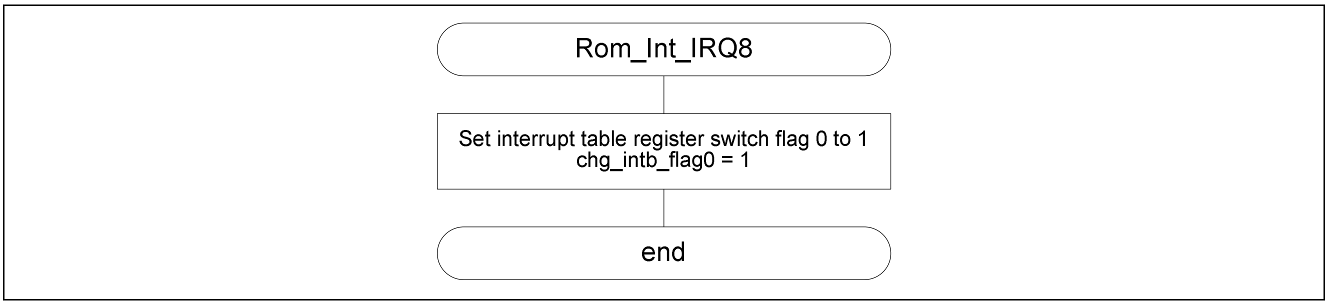


Figure 10 Flowchart (Rom_Int_IRQ8 Function)

3.3.7 Rom_Int_IRQ9 Function

(1) Functional overview

The Rom_Int_IRQ9 function is the IRQ9 interrupt handler function that indicates the relocatable vector table in the ROM area is in use. It inverts output to LED0 and returns control to the main function.

(2) Arguments

None

(3) Return values

None

(4) Flowchart

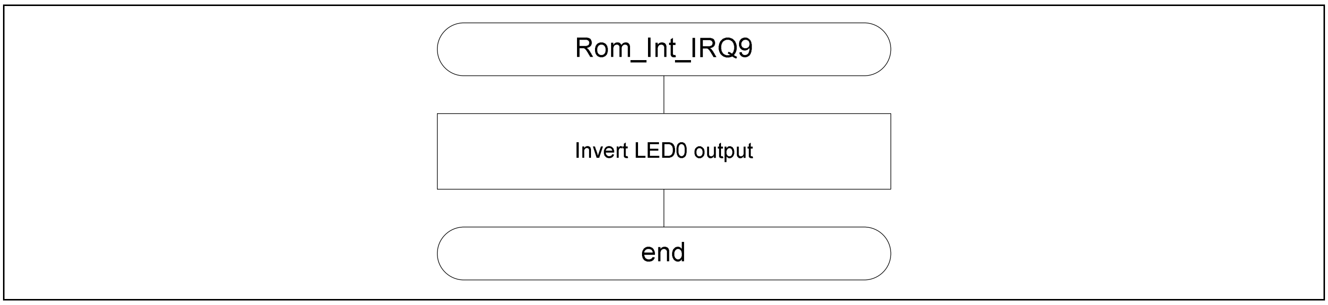


Figure 11 Flowchart (Rom_Int_IRQ9 Function)

3.3.8 Ram_Int_IRQ8 Function

(1) Functional overview

The Ram_Int_IRQ8 function is the IRQ8 interrupt handler function that indicates the relocatable vector table in the RAM area is in use. It sets interrupt table register switch flag 1 to 1 and returns control to the main function.

(2) Arguments

None

(3) Return values

None

(4) Flowchart

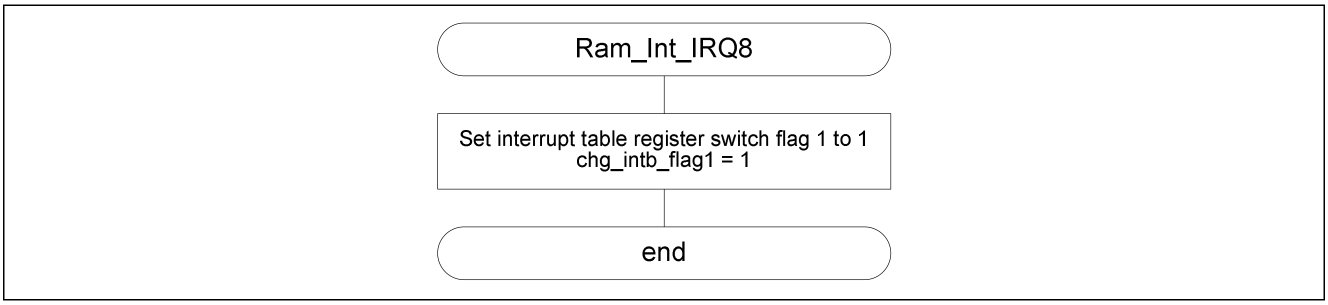


Figure 12 Flowchart (Ram_Int_IRQ8 Function)

3.3.9 Ram_Int_IRQ9 Function

(1) Functional overview

The Ram_Int_IRQ9 function is the IRQ9 interrupt handler function that indicates the relocatable vector table in the RAM area is in use. Inverts output to LED1 and returns control to the main function.

(2) Arguments

None

(3) Return values

None

(4) Flowchart

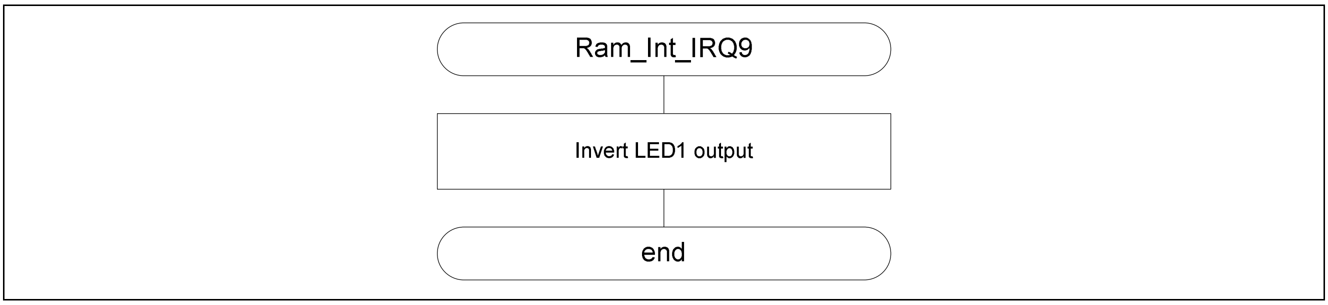


Figure 13 Flowchart (Ram_Int_IRQ9 Function)

4. Operation Confirmation Environment

Table 4 shows the environment on which the operation of the example program has been confirmed.

Table 4 Operation Confirmation Environment

Item	Name
Device	RX610 (R5F56108VNFP)
Board	Evaluation board
Power supply voltage	5.0 V (CPU operating voltage: 3.3 V)
Input clock	12.5 MHz (ICLK = 100 MHz, PCLK = 50 MHz, BCLK = 25 MHz)
Operating temperature	Room temperature
HEW	Version 4.07.00.007
Toolchain	RX Standard Toolchain (V.1.0.0.0) RX Family C/C++ Compile Driver V.1.00.00.001 RX Family C/C++ Compiler V.1.00.00.001 RX Family Assembler V.1.00.00.001 Optimizing Linkage Editor V.10.00.00.001 RX Family C/C++ Standard Library Generator V.1.00.00.001
Debugger	RX E20 SYSTEM V.1.00.00.000

5. Reference Documents

- Hardware Manual
  RX610 Group Hardware Manual
  (The latest version can be downloaded from the Renesas Electronics Web site.)

- Development Environment Manual
  RX Family C/C++ Compiler Package User's Manual
  (The latest version can be downloaded from the Renesas Electronics Web site.)

- Technical Updates
  (The latest information can be downloaded from the Renesas Electronics Web site.)

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Dec.09.10	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "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.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. 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.
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.

RENESAS

Renesas Electronics Corporation

http://www.renesas.com

SALES OFFICES

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: +44-1628-585-100, Fax: +44-1628-585-900

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.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

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

Colophon 1.0