To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

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

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

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.

H8/300L SLP Series

Handling Multiple Internally Generated Interrupts

Introduction

Multiple-interrupt handling is implemented using internal interrupts generated by Timer A and FH.

Target Device

H8/38024

Contents

1. Specifications

1. Multiple-interrupt handling is implemented using internal interrupts generated by Timer A and FH.
2. By accepting a Timer FH interrupt request in Timer A interrupt handling, the priority of Timer FH interrupt request is set higher than that of Timer A interrupt request by software.
3. The LED is made to blink in the Timer FH interrupt handling.
4. Using the interval function, a Timer A interrupt request is set to be generated every 26.214 ms.
5. Using the interval function, a Timer FH interrupt request is set to be generated every 1.638 ms.
6. The LED is connected to the P92 output pin of port 9.
7. P92 pin is a large-current port.

2. Description of Functions

1. In this sample task, multiple-interrupt handling is implemented using internal interrupts generated by Timer A and FH.
   A. Internal interrupts are described below.
      - Each on-chip peripheral module has flags to indicate its interrupt request statuses and enable bits to enable or disable the interrupts. For timer A interrupt requests and direct transition interrupt requests generated by execution of a SLEEP instruction, their interrupt request status flags and enable bits are provided in IRR1 and IENR1. When an on-chip peripheral module generates an interrupt request, the corresponding interrupt request status flag is set to 1 and the interrupt request is issued to the CPU. These interrupts can be masked by clearing the corresponding enable bits.
      - All interrupts can be masked by setting the I bit in the CCR to 1.
      - Interrupt operation is described below.
        (1) If an interrupt occurs with its corresponding bit in the interrupt enable register set to 1, an interrupt request signal is sent to the interrupt controller.
        (2) On receiving the interrupt request signal, the corresponding interrupt request flag is set.
        (3) If multiple interrupt requests are generated with their corresponding interrupt request flags set to 1, the interrupt with the highest priority at that time is requested. Other interrupt requests are held pending.
        (4) The CPU checks the I bit setting in CCR. If the I bit is cleared to 0, the interrupt request is accepted. If the I bit is set to 1, the interrupt request is held pending.
        (5) If the CPU accepts the interrupt, after processing of the current instruction is completed, both the program counter (PC) and CCR are saved to the stack. This PC value saved in the stack is the address of the first instruction to be executed upon return from interrupt handling.
        (6) The I bit in CCR is set to 1 to mask all other interrupts.
        (7) The CPU generates the vector address corresponding to the accepted interrupt, and the interrupt handling routine starts execution from the address indicated in that address.
      - Disabling of interrupts by clearing bits in IENR1 and clearing of bits in IRR1 must be done while interrupts are masked (I bit is set to 1).
      - If the above clearing operations are performed while the I bit is 0, a conflict may arise between the clearing instruction and an interrupt request. In such situation, exception handling for the interrupt will be executed after the execution of the clearing instruction has been completed.
      - The timer A and timer FH interrupt periods in this sample task are calculated by the following equations:

$$\text{Timer A interrupt period} = \frac{1}{\text{System clock} / 512} \times 256$$

$$26.214\ \text{ms} = \frac{1}{5\ \text{MHz} / 512} \times 256$$

$$\text{Timer FH interrupt period} = \frac{1}{\text{System clock}/32} \times 256$$

$$1.638\ \text{ms} = \frac{1}{5\ \text{MHz} / 32} \times 256$$

2. Table 2.1 shows the function assignments in this sample task. Multiple-interrupt handling for internal interrupts is performed by assigning the functions as shown in table 2.1.

Table 2.1 Assignment of Functions

Pin/Register	Assigned Function
PSS	A 13-bit up counter using the system clock (5 MHz) as input.
TMA	Selects PSS and sets the divide-by-ratio for the prescaler.
TCA	An 8-bit up counter using the system clock (5 MHz)/512 as input.
TCRF	Selects the output level of TCFH and selects the input clock.
TCSRF	Enables TCFH overflow interrupts and selects TCFH clearing by compare-match FH.
TCFH	An 8-bit up counter using the system clock /32 as input.
OCRFH	Timer FH interrupt period
IENTA	Enables Timer A interrupt requests.
IENTFH	Enables Timer FH interrupt requests.
IRRTA	Indicates whether or not a Timer A interrupt request has been issued.
IRRTFH	Indicates whether or not a Timer FH interrupt request has been issued.
Bit I of CCR	Enables/disables all interrupt requests.
P92	LED output

3. Principle of Operation

1. Figure 3.1 illustrates the operation of this sample task. Multiple-interrupt handling for internally generated interrupts is implemented through hardware and software processing as shown in the figure.

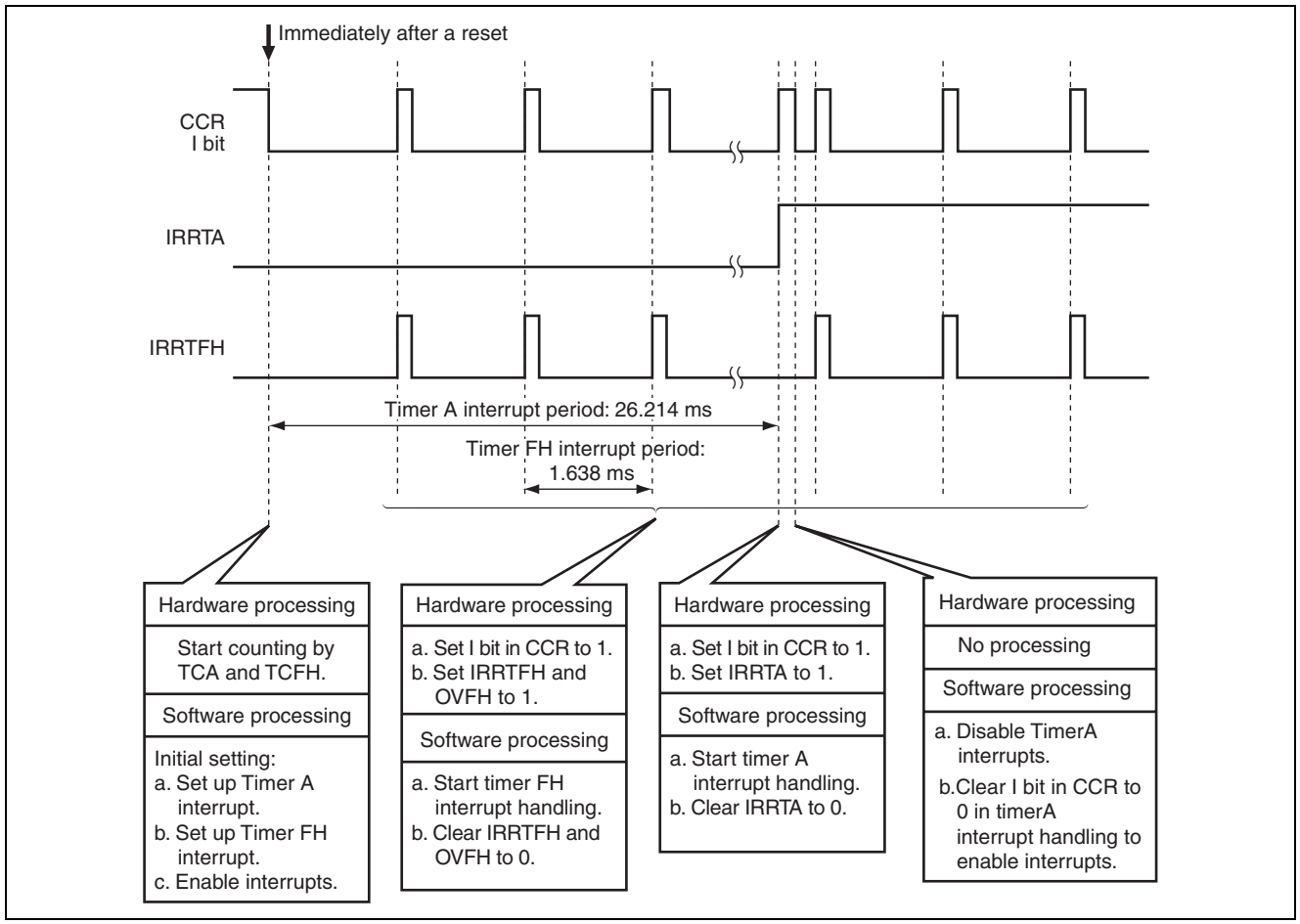


Figure 3.1 Operation Principle of Multiple-Interrupt Handling for Internal Interrupts

4. Description of Software

4.1 Modules

The modules used in this sample task are shown in table 4.1.

Table 4.1 Description of Modules

Module	Label	Function
Main Routine	main	Makes settings for Timer A interrupts, Timer FH interrupts, and port 9 and enables interrupts.
Count	taint	A Timer A interrupt handling routine that enables interrupts, increments the 16-bit counter, and ends when the counter value reached 5000.
LED Control	tfint	A Timer FH interrupt handling routine that makes the LED blink.

4.2 Arguments

This sample task does not use arguments.

4.3 Internal registers

Table 4.2 shows the internal registers involved in this sample task.

Table 4.2 Description of Internal Registers

Register		Function	Address	Setting
TMA		Timer Mode Register A If TMA = H'13, timer A functions as an interval timer, the TCA input clock source is PSS, and the divide-by-ratio for the prescaler is set to 512.	H'FFB0	H'13
TCA		Timer Counter A An 8-bit up counter using clock input of system clock/512	H'FFB1	H'00
TCRF	TOLH	Timer Control Register F (Toggle Output Level H) If TOLH = 0, TMOFH pin output level is high. If TOLH = 1, TMOFH pin output level is low.	H'FFB6 Bit 7	0
	CKSH2 CKSH1 CKSH0	Timer Control Register F (Clock Select H) If CKSH2 = 1, CKSH1 = 0 and CKSH0 = 0, TCFH is incremented by system clock/32.	H'FFB6 Bit 6 Bit 5 Bit 4	CKSH2 = 1 CKSH1 = 0 CKSH0 = 0
TCSRF	OVFH	Timer Control/Status Register F (Timer Overflow Flag H) If OVFH = 0, TCFH has not overflowed. If OVFH = 1, TCFH has overflowed.	H'FFB7 Bit 7	0
	CMFH	Timer Control/Status Register F (Compare-match Flag H) If CMFH = 0, Compare-match FH has not occurred. If CMFH = 1, Compare-match FH has occurred.	H'FFB7 Bit 6	0

Register		Function	Address	Setting
TCSRF	OVIEH	Timer Control/Status Register F (Timer Overflow Interrupt Enable H) If OVIEH = 0, interrupt request by overflow of TCFH is disabled. If OVIEH = 1, interrupt request by overflow of TCFH is enabled.	H'FFB7 Bit 5	0
	CCLRH	Timer Control/Status Register F (Counter Clear H) If CCLRH = 0, clearing of TCFH by compare-match FH is disabled. If CCLRH = 1, clearing of TCFH by compare-match FH is enabled.	H'FFB7 Bit 4	1
TCFH		Timer Counter FH An 8-bit up counter using system clock/32 as input	H'FFB8	H'00
OCRFH		Output Compare Register FH If OCRFH = H'80, compare-match FH signal is generated when the counter value of TCFH has reached H'80.	H'FFBA	H'FF
PDR9	P92	Port Data Register 9 (Port Data Register 92) If P92 = 0, the P92 pin output level is low. If P92 = 1, the P92 pin output level is high.	H'FFDC Bit 2	0
IENR1	IENTA	Interrupt Enable Register 1 (Timer A Interrupt Enable) If IENTA = 0, Timer A interrupt request is disabled. If IENTA = 1, Timer A interrupt request is enabled.	H'FFF3 Bit 7	1
IENR2	IENTFH	Interrupt Enable Register 2 (Timer FH Interrupt Enable) If IENTFH = 0, Timer FH interrupt request is disabled. If IENTFH = 1, Timer FH interrupt request is enabled.	H'FFF4 Bit 3	1
IRR1	IRRTA	Interrupt Request Register 1 (Timer A Interrupt Request Flag) If IRRTA = 0, Timer A interrupt request is not issued. If IRRTA = 1, Timer A interrupt request has been issued.	H'FFF6 Bit 7	0
IRR2	IRRTFH	Interrupt Request Register 2 (Timer FH Interrupt Request Flag) If IRRTFH = 0, Timer FH interrupt request is not issued. If IRRTFH = 1, Timer FH interrupt request has been issued.	H'FFF7 Bit 3	0

4.4 Description of RAM

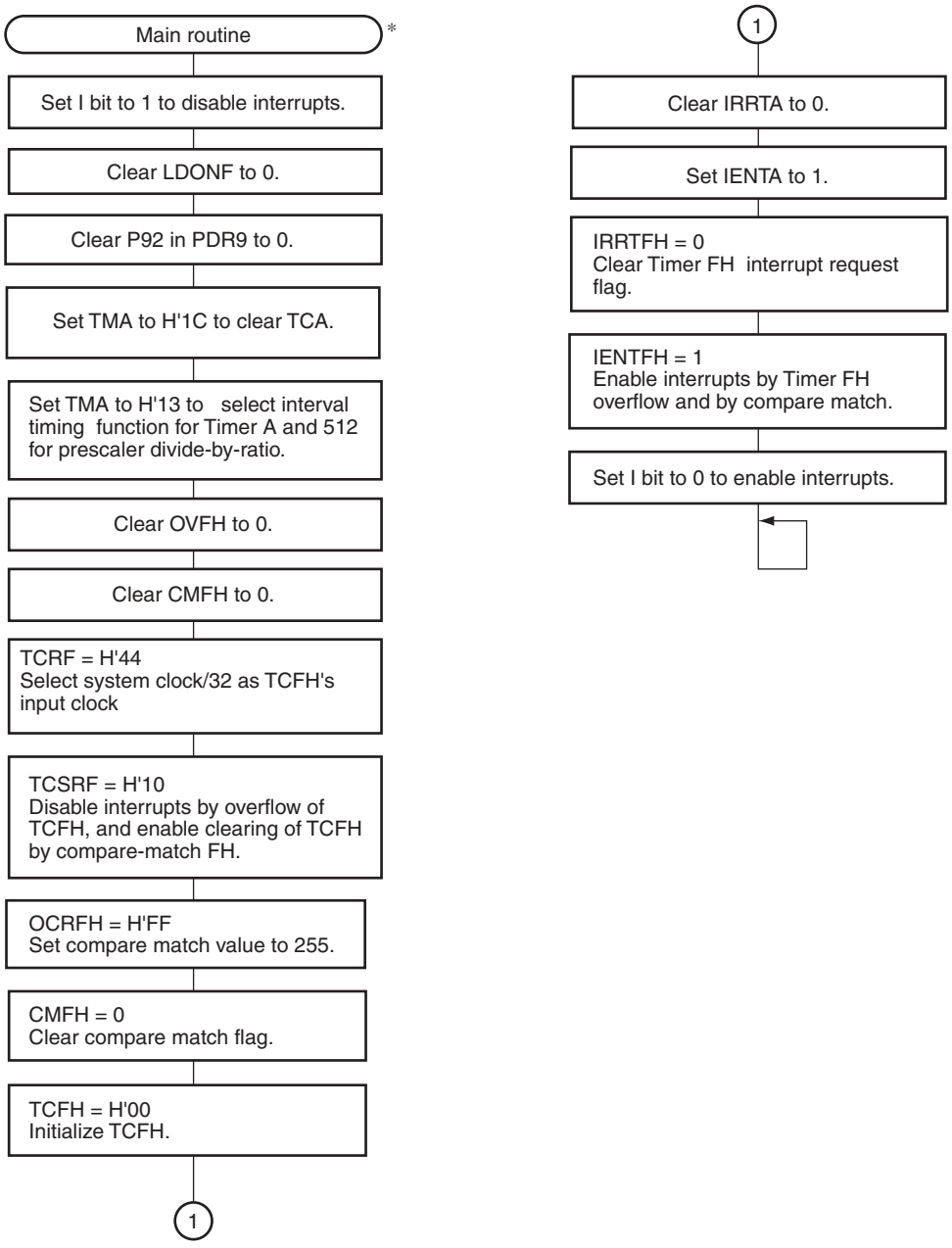
Table 4.3 describes the RAM area used in this sample task.

Table 4.3 Description of RAM

Label		Function	Address	Used in
USRF	LDONF	Flag to judge ON/OFF of the LED	H'FB82 Bit 0	LED control

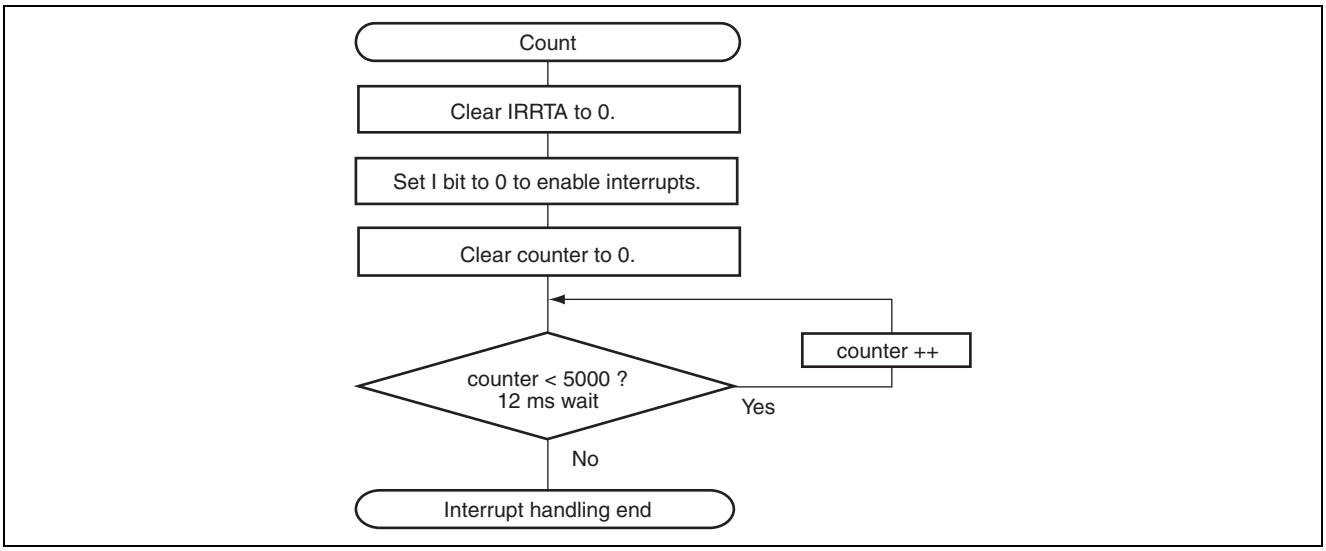
5. Flowchart

1. Main routine

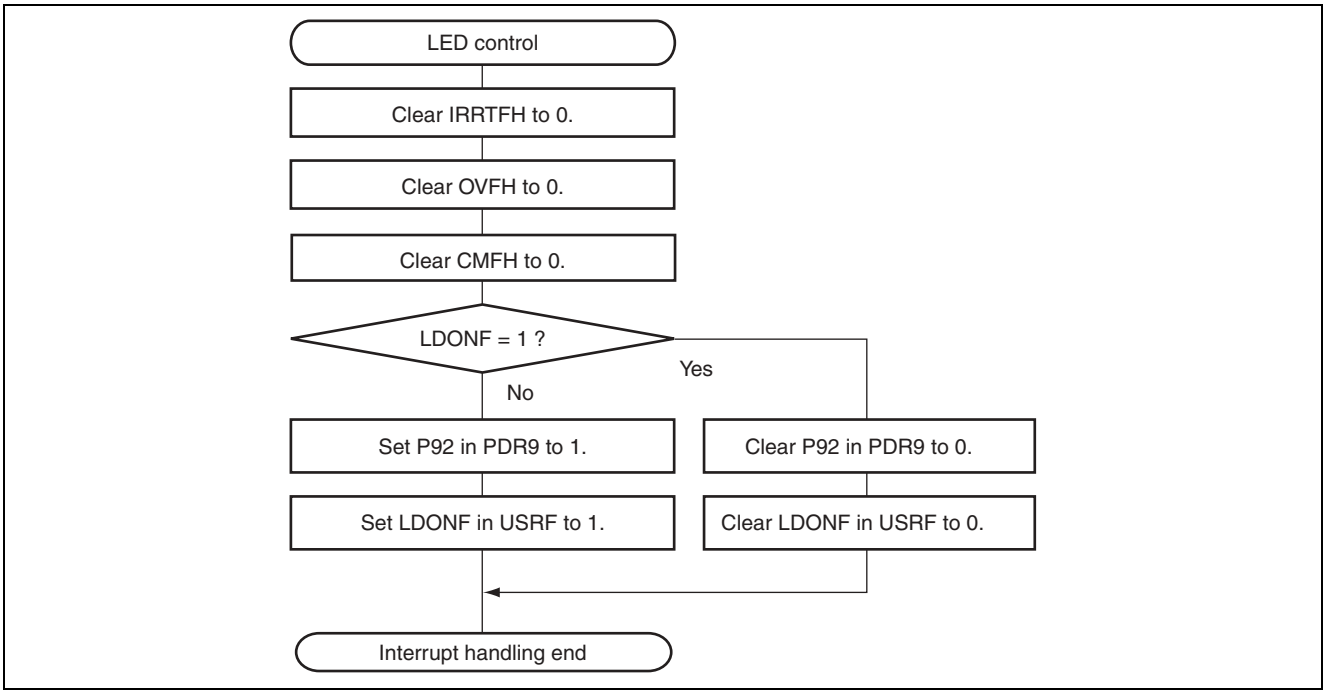


Note: * In this sample task, the stack pointer is set in INIT.SRC (assembly language).

2. Timer A interrupt handling routine



3. Timer FH interrupt handling routine



6. Program Listing

INIT.SRC (Program listing)

```
    .EXPORT   _INIT
    .IMPORT   _main
;
    .SECTION  P,CODE
_INIT:
    MOV.W     #H'FF80,R7
    LDC.B     #B'10000000,CCR
    JMP       @_main
;
    .END
```

```
/************************************************************************************************************************/
/*                                                                                                                      */
/*  H8/300L Super Low Power Series                                                                                      */
/*      -H8/38024 Series-                                                                                               */
/*  Application Note                                                                                                    */
/*                                                                                                                      */
/*  'Multiple Interrupt Operation by Internal                                                                           */
/*   Interrupt'                                                                                                         */
/*                                                                                                                      */
/*  Function                                                                                                            */
/*  : Internal Interrupt                                                                                                */
/*                                                                                                                      */
/*  External Clock : 10MHz                                                                                              */
/*  Internal Clock : 5MHz                                                                                               */
/*  Sub Clock      : 32.768kHz                                                                                          */
/*                                                                                                                      */
/************************************************************************************************************************/

#include    <machine.h>

/************************************************************************************************************************/
/*  Symbol Definition                                                                                                   */
/************************************************************************************************************************/
struct BIT {
    unsigned char   b7:1;       /* bit7 */
    unsigned char   b6:1;       /* bit6 */
    unsigned char   b5:1;       /* bit5 */
    unsigned char   b4:1;       /* bit4 */
    unsigned char   b3:1;       /* bit3 */
    unsigned char   b2:1;       /* bit2 */
    unsigned char   b1:1;       /* bit1 */
    unsigned char   b0:1;       /* bit0 */
};

#define     TMA         *(volatile unsigned char *)0xFFB0       /* Timer Mode Register A                */
#define     TCA         *(volatile unsigned char *)0xFFB1       /* Timer Counter A                      */
#define     TCRF        *(volatile unsigned char *)0xFFB6       /* Timer Control Register F             */
#define     TCRF_BIT    (*(struct BIT *)0xFFB6)                 /* Timer Control Register F             */
#define     TOLH        TCRF_BIT.b7                             /* Toggle Output Level F                */
#define     CKSH2       TCRF_BIT.b6                             /* Clock Select H2                      */
```

```
#define     CKSH1       TCRF_BIT.b5                         /* Clock Select H1                          */
#define     CKSH0       TCRF_BIT.b4                         /* Clock Select H0                          */
#define     TCSRF       *(volatile unsigned char *)0xFFB7   /* Timer Control Status Register F          */
#define     TCSRF_BIT   (*(struct BIT *)0xFFB7)             /* Timer Control Status Register F          */
#define     OVFH        TCSRF_BIT.b7                        /* Timer Overflow Flag H                    */
#define     CMFH        TCSRF_BIT.b6                        /* Compare-match Flag H                     */
#define     OVIEH       TCSRF_BIT.b5                        /* Timer Overflow Interrupt Enable          */
#define     CCLRH       TCSRF_BIT.b4                        /* Output Select 3                          */
#define     TCFH        *(volatile unsigned char *)0xFFB8   /* Timer Counter FH                         */
#define     OCRFH       *(volatile unsigned char *)0xFFBA   /* Output Compare Register FH               */
#define     IENR1_BIT   (*(struct BIT *)0xFFF3)             /* Interrupt Enable Register 1              */
#define     IENTA       IENR1_BIT.b7                        /* Timer A Interrupt Enable                 */
#define     IENR2_BIT   (*(struct BIT *)0xFFF4)             /* Interrupt Enable Register 2              */
#define     IENTFH      IENR2_BIT.b3                        /* Timer FH Interrupt Enable                */
#define     IRR1_BIT    (*(struct BIT *)0xFFF6)             /* Interrupt Request Register 1             */
#define     IRRTA       IRR1_BIT.b7                         /* Timer A Interrupt Request Flag           */
#define     IRR2_BIT    (*(struct BIT *)0xFFF7)             /* Interrupt Request Register 2             */
#define     IRRTFH      IRR2_BIT.b3                         /* Timer FH Interrupt Request Flag          */
#define     PDR9_BIT    (*(struct BIT *)0xFFDC)             /* Port Data Register 9                     */
#define     P92         PDR9_BIT.b2                         /* Port Data Register 9 bit2                */

#pragma interrupt   (taint)
#pragma interrupt   (tfint)
/************************************************************************************************************/
/*  Function define                                                                                         */
/************************************************************************************************************/
extern void INIT ( void );                                  /* SP Set                                   */
void        main ( void );
void        taint ( void );
void        tfint ( void );

/************************************************************************************************************/
/*  RAM define                                                                                              */
/************************************************************************************************************/
unsigned char   USRF;                                       /* User Flag Area                           */

#define  USRF_BIT   (*(struct BIT *)&USRF)
#define  LDONF      USRF_BIT.b0                             /* LED On Flag                              */

/************************************************************************************************************/
/*  Vector Address                                                                                          */
/************************************************************************************************************/
#pragma section     V1                                      /* Vector Section Set                       */
void (*const VEC_TBL1[])(void) = {
    INIT                                                    /* 0x0000 Reset Vector                      */
};
#pragma section     V2                                      /* Vector Section Set                       */
void (*const VEC_TBL2[])(void) = {
    taint                                                   /* 0x016 Timer A  Interrupt Vector          */
};
#pragma section     V3                                      /* Vector Section Set                       */
void (*const VEC_TBL3[])(void) = {
    tfint                                                   /* 0x01E Timer F Interrupt Vector           */
};
```

```
#pragma section                                         /* P                                    */
/*****************************************************************************************************************/
/*  Main Program                                                                                                 */
/*****************************************************************************************************************/
void    main ( void )
{
    int tmp;

    set_imask_ccr(1);                                   /* Interrupt Disable                    */

    LDONF = 0;                                          /* Clear LDONF                          */
    P92   = 0;                                          /* Clear P92                            */

    TMA  = 0x1C;                                        /* TCA Clear                            */
    TMA  = 0x13;                                        /* Initialize TMA Function &            */
                                                        /*                   TCA Input Clock Period */

    tmp  = OVFH;
    tmp  = 0;
    OVFH = tmp;                                         /* Clear OVFH                           */
    tmp  = CMFH;
    tmp  = 0;
    CMFH = tmp;                                         /* Clear CMFH                           */

    TCRF  = 0x44;                                       /* Initialize Clock Select              */
    TCSRF = 0x10;                                       /* Initialize Overflow Interrupt        */

    OCRFH = 0xFF;                                       /* Initialize Compare-match FH Value    */
    CMFH  = 0;                                          /* Clear Compare-match Flag FH          */
    TCFH  = 0;                                          /* Compare-match FH Interrupt Enable    */

    IRRTA = 0;                                          /* Clear IRRTA                          */
    IENTA = 1;                                          /* Timer A Interrupt Enable             */

    IRRTFH = 0;                                         /* Clear IRRTFH                         */
    IENTFH = 1;                                         /* Timer FH Interrupt Enable            */

    set_imask_ccr(0);                                   /* Interrupt Enable                     */

    while(1){
        ;
    }
}
```

```
/************************************************************************************************************************/
/*  Timer A Interrupt                                                                                                   */
/************************************************************************************************************************/
void    taint ( void )
{
    unsigned short    counter;

    IRRTA = 0;                                          /* Clear IRRTA                          */

    set_imask_ccr(0);                                   /* Interrupt Enable                     */

    for(counter = 0; counter < 5000; counter++);        /* dummy wait                           */
}

/************************************************************************************************************************/
/*  Timer F Interrupt                                                                                                   */
/************************************************************************************************************************/
void    tfint ( void )
{
    int tmp;

    IRRTFH = 0;

    tmp = OVFH;
    tmp = 0;
    OVFH = tmp;                                         /* Clear OVFH                           */

    tmp = CMFH;
    tmp = 0;
    CMFH = tmp;                                         /* Clear CMFH                           */

    if(LDONF == 1){                                     /* LDONF = 1 ?                          */
        P92 = 0;                                        /* Turn off LED                         */
        LDONF = 0;                                      /* Clear LDONF                          */
    }
    else{
        P92 = 1;                                        /* urn on LED                           */
        LDONF = 1;                                      /* Set LDONF                            */
    }
}
```

Link address specifications

Section Name	Address
CV1	H'0000
CV2	H'0016
CV3	H'001E
P	H'0100
B	H'FB80

Revision Record

		Description	
Rev.	Date	Page	Summary
1.00	Dec.19.03	—	First edition issued

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.