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.

H8/300H Tiny Series

Multi-Interrupt Operation with Internal Interrupts

Introduction

Both timer A and timer V interrupts are handled.

Target Device

H8/3664

Contents

1. Specifications

- Both timer A and timer V interrupts are handled.
- The priority level of the timer V interrupt request is set higher than that of the timer A interrupt request by software so that it is possible to accept timer V interrupt requests during the timer A interrupt handling.
- A timer A interrupt request is set to be generated every 32.768 ms by the interval timing function of timer A.
- A timer V interrupt request is set to be generated every 2.048 ms by the interval timing function of timer A.
- The LED is connected to the P74 output pin of port 7.

2. Description of Functions Used

In this sample task, multi-interrupt operation of the timer A and V interrupts are performed by internal interrupts.

The internal interrupts are described below.

- Each on-chip peripheral module has a flag to show the interrupt request status and an enable bit to enable or disable the interrupt. For timer A interrupt requests and direct transfer interrupt requests that are generated by execution of a SLEEP instruction, this function is included in interrupt request register 1 (IRR1) and interrupt enable register 1 (IENR1). When an on-chip peripheral module requests an interrupt, the corresponding interrupt request status flag is set to 1, requesting the CPU of an interrupt. These interrupts can be masked by writing 0 to clear the corresponding enable bit.
- All interrupts can be masked by setting the I bit in the condition code register (CCR) to 1.
- Interrupt operation is described as follows.
  1. If an interrupt occurs while the corresponding bit in the interrupt enable register is set to 1, an interrupt request signal is sent to the interrupt controller.
  2. On receiving the interrupt request signal, the corresponding interrupt request status flag is set to 1, requesting the CPU of an interrupt.
  3. When multiple interrupt requests are generated, the interrupt controller requests to the CPU for the interrupt handling with the highest priority at that time. 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, interrupt handling will begin. First, both the PC and CCR are pushed onto the stack. The PC value pushed onto 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 further interrupts.
  7. The CPU generates the vector address corresponding to the accepted interrupt, and transfers the address to PC as a start address of the interrupt handling routine. Then a program starts executing from the address indicated in PC.

- When disabling interrupts by clearing bits in IENR1 or IRR1, always do so while interrupts are masked (I bit is set to 1). If the above clear operations are performed while the I bit is cleared to 0, and as a result a conflict arises between the clear instruction and an interrupt request, exception handling for the interrupt will be executed after the clear instruction has been executed.
- The timer A and timer V interrupt cycles in this sample task are calculated by the following equations:

$$\text{Timer A interrupt cycle} = \frac{1}{\text{System clock}/2048} \times 256$$

$$32.768\ \text{ms} = \frac{1}{16\ \text{MHz}/2048} \times 256$$

$$\text{Timer V interrupt cycle} = 2 \times \frac{1}{\text{System clock}/64} \times 256$$

$$2.048\ \text{ms} = 2 \times \frac{1}{16\ \text{MHz}/64} \times 256$$

Table 1 lists the function allocation for this sample task. The functions listed in table 1 are allocated to perform multi-interrupt operation with internal interrupts.

Table 1 Function Allocation

Function	Description
OVIE	Enables interrupt requests by OVF in TSRW
IENTA	Enables timer A interrupt requests
OVF	Indicates whether or not a TCNTV overflow interrupt request is issued
IRRTA	Indicates whether or not a timer A interrupt request is issued
I bit in CCR	Enables or disables all interrupt requests
P74	LED output pin

3. Description of Operations

Figure 1 shows this sample task's principle of operation. The hardware and software processing shown in figure 1 performs multi-interrupt operation with internal interrupts.

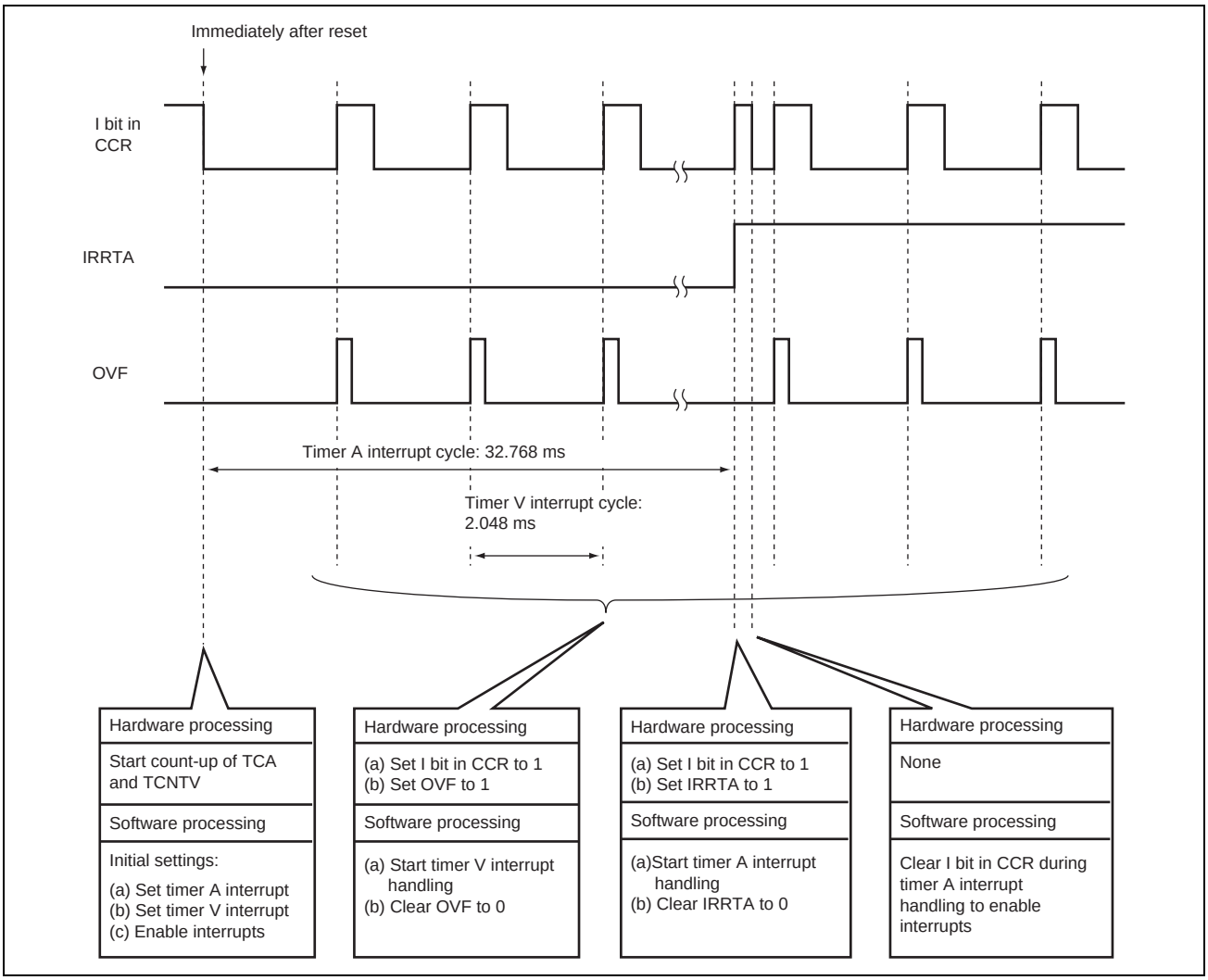


Figure 1 Operation Principle: Multi-Interrupt Operation with Internal Interrupts

4. Description of Software

4.1 Description of Modules

Table 2 describes the software used in this sample task.

Table 2 Description of Modules

Module Name	Label Name	Function
Main routine	main	Sets timer A interrupts, timer V interrupts, and port 7, and enables interrupts.
Count	taint	During the timer A interrupt handling routine, enables interrupts, increments the 16-bit counter, and ends TCNT when H'4000 is reached.
LED control	tvint	During the timer V interrupt handling routine, turns on/off the LED.

4.2 Description of Arguments

No arguments are used in this sample task.

4.3 Description of Internal Registers

Table 3 describes the internal registers used in this sample task.

Table 3 Description of Internal Registers

Register	Name	Function	Address	Setting
TCRV0	OVIE	Timer control register V0 (timer overflow interrupt enable): When OVIE is set to 1, TNCTV overflow interrupts are enabled.	H'FFA0 Bit 5	 0
		Timer control register V0 (clock select 2 to 0):	H'FFA0	
	CKS2	When CKS2 is cleared to 0 and CKS1, CKS0, and ICKS0 are all set to 1, TCNTV is incremented at the falling edge of system clock/64.	Bit 2	CKS2 = 0
	CKS1		Bit 1	CKS1 = 1
	CKS0		Bit 0	CKS0 = 1
TCSRV	OVF	Timer control/status register V (timer overflow flag): When OVF is cleared to 0, a TCNTV overflow interrupt is not requested. When OVF is set to 1, a TCNTV overflow interrupt is requested.	H'FFA1 Bit 5	 0
TNCTV		Timer counter V: 8-bit up-counter incremented by clock input of system clock/64	H'FFA4	H'00

Table 3 Description of Internal Registers (cont)

Register	Name	Function	Address	Setting
TCRV1	ICKS0	Timer control register V1 (internal clock select 0): Selects the TCNTV clock source, together with bits CK2 to CK0 in TCRV0.	H'FFA5 Bit 0	 0
TMA		Timer mode register A: When TMA is set to H'12, timer A is set to the interval timing function, the TCA input clock source to PSS, and the prescaler division ratio to division by 2048.	H'FFA6	H'12
TCA		Timer counter A: Incrementing the 8-bit counter using clock input of system clock/2048.	H'FFA7	H'00
PDR7	P74	Port data register 7 (port data register 74): When P74 is cleared to 0, the P74 pin output level is low. When P74 is set to 1, the P74 pin output level is high.	H'FFDA Bit 4	 0
PCR7	PCR74	Port control register 7 (port control register 74): When PCR74 is set to 1, the P74 pin functions as an output pin.	H'FFEA Bit 4	 1
IENR1	IENTA	Interrupt enable register 1 (timer A interrupt enable): When IENTA is set to 1, timer A interrupt requests are enabled.	H'FFF4 Bit 6	 1
IRR1	IRRTA	Interrupt request register 1 (timer A interrupt request flag): When IRRTA is cleared to 0, no timer A interrupt is requested. When IRRTA is set to 1, a timer A interrupt is requested.	H'FFF6 Bit 6	 0

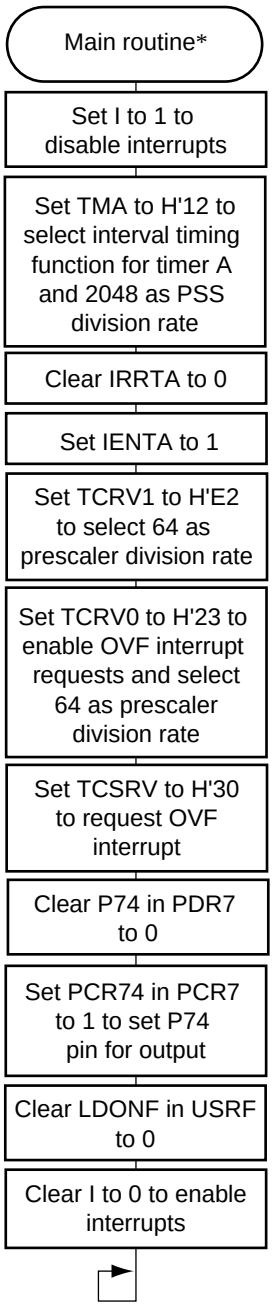
4.4 Description of RAM

Table 4 describes the RAM used in this sample task.

Table 4 Description of RAM

Label	Name	Function	Address	Used in
counter_sub		16-bit counter for counting up to H'4000 in an interrupt routine (timer A)	H'FB80	Interrupt routine Timer A
USRF	LDONF	Flag for judging on/off of the LED	H'FB82 Bit 0	LED control

5. Flowcharts



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

Figure 2 Flowchart for Main Routine

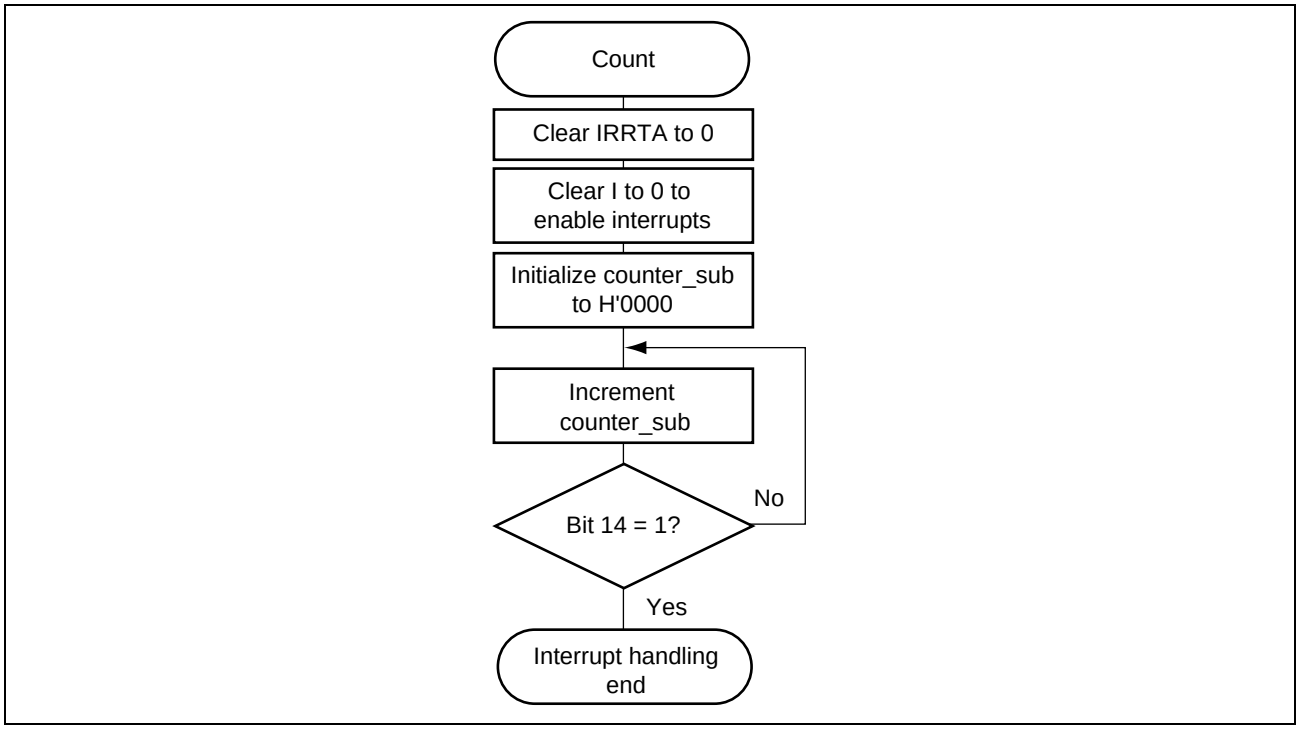


Figure 3 Flowchart for Timer A Interrupt Handling Routine

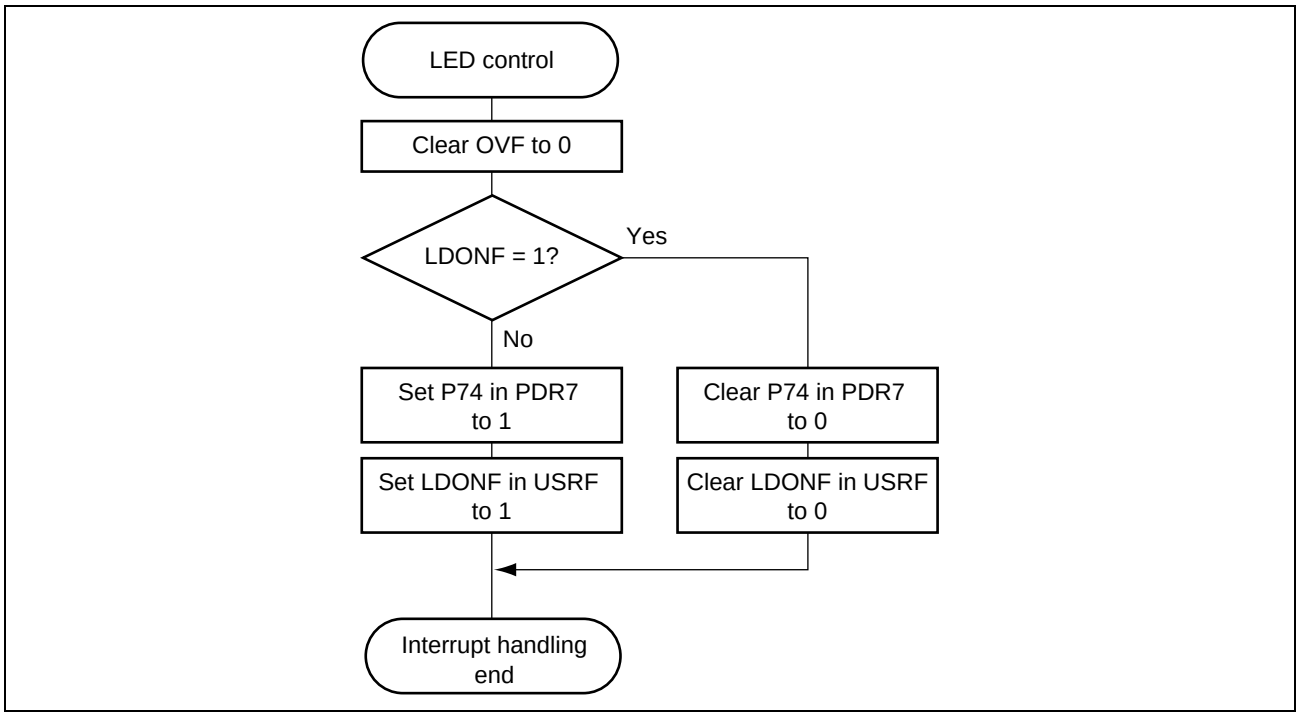


Figure 4 Flowchart for Timer V Interrupt Handling Routine

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
```

```
/************************************************************/
/*                                                          */
/*    H8/300H Tiny Series -H8/3664-                         */
/*    Application Note                                      */
/*                                                          */
/*    'Multiple Interrupt Operation by Internal             */
/*     Interrupt'                                           */
/*                                                          */
/*    Function                                              */
/*    : Internal Interrupt                                  */
/*                                                          */
/*    External Clock : 16MHz                                */
/*    Internal Clock : 16MHz                                */
/*    Sub Clock      : 32.768kHz                            */
/*                                                          */
/************************************************************/

#include    <machine.h>
```

```
/************************************************************/
/*  Symbol Defnition    */
/************************************************************/
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    TCRV0        *(volatile unsigned char *)0xFFA0                                        /*
Timer Contorol Register V0                          */
#define    TCSRV        *(volatile unsigned char *)0xFFA1                                        /*
Timer Control/Status Register V                     */
#define    TCSRV_BIT    (*(struct BIT *)0xFFA1)    /* Timer Control/Status Register V          */
#define    OVF          TCSRV_BIT.b5               /* Timer Overflow Flag                      */
#define    TCNTV        *(volatile unsigned char *)0xFFA4                                        /*
Timer Counter V         */
#define    TCRV1        *(volatile unsigned char *)0xFFA5                                        /*
Timer Contorol Register V1                          */
#define    TMA          *(volatile unsigned char *)0xFFA6                                        /*
Timer Mode Register A   */
#define    TCA          *(volatile unsigned char *)0xFFA7                                        /*
Timer Counter A         */
#define    PDR7_BIT     (*(struct BIT *)0xFFDA)    /* Port Data Register 7                     */
#define    P74          PDR7_BIT.b4                /* Port Data Register 7 bit4                */
#define    PCR7_BIT     (*(struct BIT *)0xFFEA)    /* Port Control Register 7                  */
#define    PCR74        PCR7_BIT.b4                /* Port Control Register 7 bit4             */
#define    IENR1_BIT    (*(struct BIT *)0xFFF4)    /* Interrupt Enable Register 1              */
#define    IENTA        IENR1_BIT.b6               /* Timer A Interrupt Enable                 */
#define    IEN0         IENR1_BIT.b0               /* IRQ0 Interrupt Enable                    */
#define    IRR1_BIT     (*(struct BIT *)0xFFF6)    /* Interrupt Request Register 1             */
#define    IRRTA        IRR1_BIT.b6                /* Timer A Interrupt Request Flag           */
#define    IRRI0        IRR1_BIT.b0                /* IRQ0 Interrupt Request Flag              */
#define    PMR1_BIT     (*(struct BIT *)0xFFE0)    /* Port Mode Register 1                     */
#define    IRQ0_SET     PMR1_BIT.b4                /* Port Mode Register 1 bit4                */
#define    TMOW         PMR1_BIT.b0                /* P10/TMOW Terminal Function Change        */
#define    IEGR1_BIT    (*(struct BIT *)0xFFF2)    /* Interrupt Edge Select Register 1         */
#define    IEG0         IEGR1_BIT.b0               /* IRQ0 Edge Select                         */
#pragma    interrupt    (taint)
#pragma    interrupt    (tvint)
```

```
/************************************************************/
/*     Function Definition                                  */
/************************************************************/
extern     void INIT     ( void );                /* SP Set                                  */
void main   ( void );
void taint  ( void );
void tvint  ( void );

/************************************************************/
/*    RAM define                                            */
/************************************************************/
unsigned int             counter_sub;
unsigned char            USRF;                    /* User Flag Erea                          */

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

/************************************************************/
/*    Vector Address                                        */
/************************************************************/
#pragma    section       V1                       /* VECTOR SECTOIN SET                      */
void (*const VEC_TBL1[])(void) = {
    INIT                                          /* 00 Reset                                */
};
#pragma    section       V2                       /* VECTOR SECTOIN SET                      */
void (*const VEC_TBL2[])(void) = {
    taint  /* 26 Timer A  Interrupt                */
};
#pragma    section       V3                       /* VECTOR SECTOIN SET                      */
void (*const VEC_TBL3[])(void) = {
    tvint                                         /* 2C Timer V  Interrupt                   */
};

#pragma    section                                /* P                                       */
```

```
/************************************************************/
/*    Main Program                                          */
/************************************************************/
void main   ( void )
{

    set_imask_ccr(1);                           /* Interrupt Disable                                    */

    TMA = 0x12;                                 /* Initialize TMA Function & TCA Input Clock Period     */

    IRRTA = 0;                                  /* Clear IRRTA                                          */

    IENTA = 1;                                  /* Timer A Interrupt Enable                             */

    TCRV1 = 0xE2;                               /* Initialize Time Control Register    V1               */
    TCRV0 = 0x23;                               /* Initialize TCRV0 Function & TCRV0 Input Clock Period */
    TCSRV = 0x30;                               /* Initialize Timer Control/Status Register V           */

    P74 = 0;                                    /* Clear P74                                            */
    PCR74 = 1;                                  /* Initialize P74 Output Terminal                       */

    LDONF = 0;                                  /* Clear LDONF                                          */

    set_imask_ccr(0);                           /* Interrupt Enable                                     */

    while(1){
           ;
    }
}

/************************************************************/
/*    Timer A Interrupt                                     */
/************************************************************/
void taint  ( void )
{

    IRRTA = 0;                                  /* Clear IRRTA                                          */

    set_imask_ccr(0);                           /* Interrupt Enable                                     */

    counter_sub = 0x0000;                       /* Initialize 16bit Counter                             */

    do{
    counter_sub++;      /* Increment 16bit Counter */
    }while((counter_sub & 0x4000) == 0);        /* 16bit Counter bit14 = 1 ?                            */

}
```

```
/************************************************************/
/*    Timer V Interrupt                                     */
/************************************************************/
void tvint  ( void )
{

    OVF = 0;

    if(LDONF == 1){                          /* LDONF = 1 ?                                    */
           P74 = 0;                          /* Turn off LED                                   */
           LDONF = 0;                        /* Clear LDONF                                    */
    }
           else{
                       P74 = 1;              /* urn on LED                                     */
                       LDONF = 1;            /* Set LDONF                                      */
           }

}
```

Link Address Setting:

Section Name	Address
CV1	H'0000
CV2	H'0026
CV3	H'002C
P	H'0100
B	H'FB80

Website and Support

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

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

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
2.00	Sep.01.06	All pages	Format has been changed from Hitachi version to Renesas version.

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.