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

## Using External Interrupt to Start Incrementing Counter

### Introduction

An IRQ0 interrupt is generated when the switch connected to the $\overline{\text{IRQ0}}$ pin is turned on, and incrementing of the 16-bit counter set in the 16-bit general register, counter_sub, starts.

### Target Device

H8/3664

### Contents

## 1. Specifications

- An IRQ0 interrupt is generated when the switch connected to the $\overline{\text{IRQ0}}$ pin is turned on, and incrementing of the 16-bit counter set in the 16-bit general register, counter_sub, starts.
- An IRQ0 interrupt is requested when the falling edge of the $\overline{\text{IRQ0}}$ pin input is detected.
- The LED is turned on or off whenever the 16-bit counter set in R1 overflows.
- The LED is connected to the P74 output pin of port 7.

Figure 1 shows an example of connecting a switch to the $\overline{\text{IRQ0}}$ input pin.

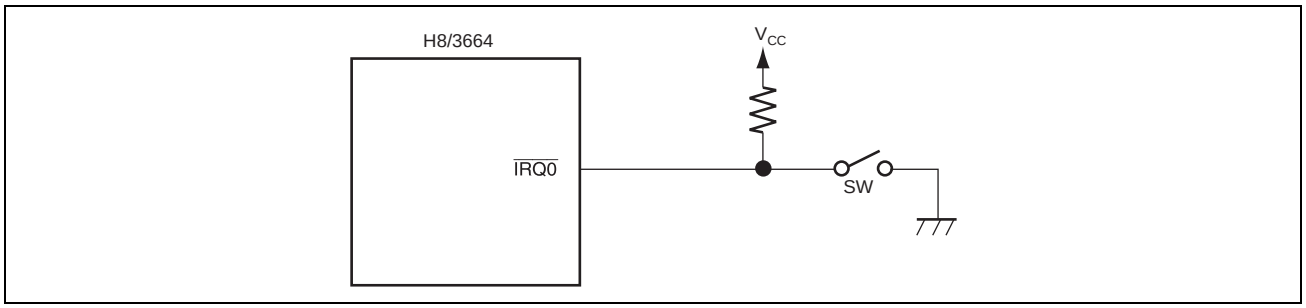



**Figure 1 Example of Connecting Switch to $\overline{\text{IRQ0}}$ Input Pin**

## 2. Description of Functions Used

In this sample task, the counter starts incrementing when an IRQ0 external interrupt occurs.

The external interrupts are described below.

- There are six external interrupts, NMI, IRQ3 to IRQ0, and WKP.
- An NMI interrupt is requested by input signal to pin $\overline{\text{NMI}}$. This interrupt is detected by sensing either the rising edge or falling edge, according to the setting of the NMIEG bit in interrupt edge select register 1 (IEGR1).
- NMI is the highest-priority interrupt, and can always be accepted regardless of the I bit setting in the condition code register (CCR).
- IRQ3 to IRQ0 interrupts are requested by input signals to pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$. These four interrupts are detected individually by sensing either the rising edge or falling edge, according to the settings of the IEG3 to IEG0 bits in IEGR1.
- Pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ are also used as I/O port pins. To use these pins as $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ input pins, set the IRQ3 to IRQ0 bits in port mode register 1 (PMR1) to 1.
- When pins $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ are designated for interrupt input by PMR1 and the designated signal edge is input, the corresponding bit among the IRRI3 to IRRI0 bits in interrupt flag register 1 (IRR1) is set to 1, requesting the CPU of an interrupt.
- IRQ3 to IRQ0 interrupts can be disabled by clearing the IEN3 to IEN0 bits in interrupt enable register 1 (IENR1) to 0.
- WKP5 to WKP0 interrupts are requested by input signals to pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$. These interrupts are detected individually by either rising edge sensing or falling edge sensing, depending on the settings of the WPEG5 to WPEG0 bits in interrupt edge select register 2 (IEGR2).
- Pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ are also used as I/O pins for port 5. To use these pins as $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ input pins, set the WKP5 to WKP0 bits in port mode register 5 (PMR5) to 1.

- When pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ are designated for interrupt input by PMR5 and the designated signal edge is input, the corresponding bit among the IWPF5 to IWPF0 bits in the wakeup interrupt flag register (IWPR) is set to 1, requesting the CPU of an interrupt.
- WKP interrupts can be disabled by clearing the IENWP bit in IENR1 to 0.
- All interrupts can be masked by setting the I bit in 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.

Table 1 lists the function allocation for this sample task. The functions listed in table 1 are allocated to start incrementing the counter when an external interrupt occurs.

**Table 1 Function Allocation**

| Function | Description |
|---|---|
| IRRI0 | Indicates whether or not an IRQ0 interrupt is requested |
| IEN0 | Enables $\overline{\text{IRQ0}}$ pin interrupt requests |
| IEG0 | Selects input edge of the $\overline{\text{IRQ0}}$ pin |
| $\overline{\text{IRQ0}}$ | Switch input pin |
| PCR7 | Sets P74 output pin function |
| PDR7 | Stores P74 output pin data |
| P74 | LED output pin |

## 3. Description of Operations

Figure 2 shows this sample task's principle of operation. The hardware and software processing shown in figure 2 applies external interrupts to start incrementing the counter.

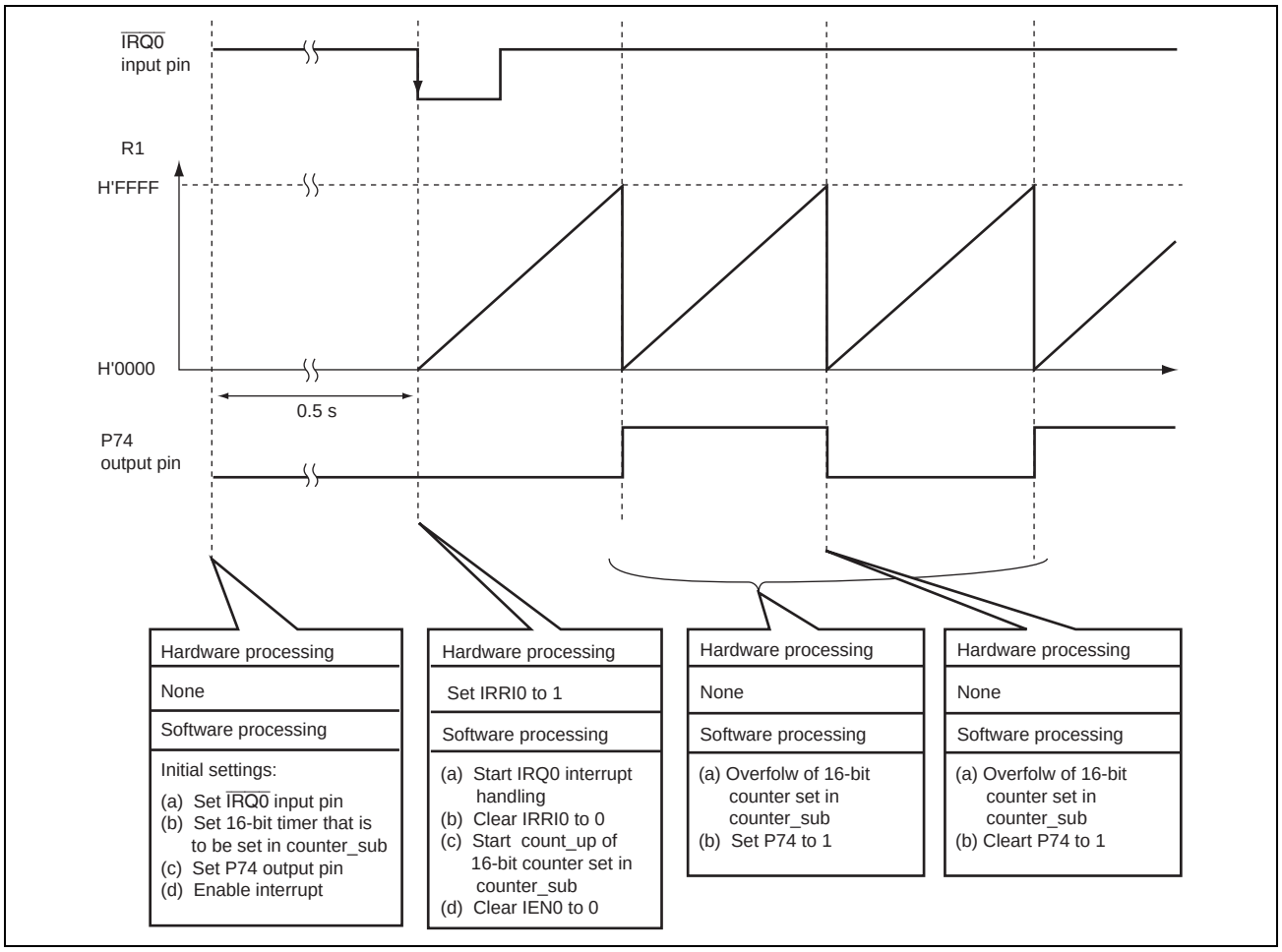



**Figure 2 Operation Principle: Using External Interrupts to Start Incrementing Counter**

# 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 the IRQ0 interrupt and LED output pin, enables interrupts, increments the 16-bit counter, and performs LED output. |
| Switch on | IRQ0 | During the IRQ0 interrupt handling routine, sets SWONF to 1. |

## 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 |
|---|---|---|---|---|
| PDR7 | P74 | Port data register 7 (port data register 74):<br>When P74 is cleared to 0, the P74 pin output level is low.<br>When P74 is set to 1, the P74 pin output level is high. | H'FFDA<br>Bit 4 | <br>0 |
| PCR7 | PCR74 | Port control register 7 (port control register 74):<br>When PCR74 is set to 1, the P74 pin functions as an output pin. | H'FFEA<br>Bit 4 | <br>1 |
| IEGR1 | IEG0 | Interrupt edge select register 1 (IRQ0 edge select):<br>When IEG0 is cleared to 0, the falling edge of the $\overline{\text{IRQ0}}$ pin input is detected.<br>When IEG0 is set to 1, the rising edge of the $\overline{\text{IRQ0}}$ pin input is detected. | H'FFF2<br>Bit 0 | <br>0 |
| IENR1 | IEN0 | Interrupt enable register 1 (IRQ0 interrupt enable):<br>When IEN0 is set to 1, interrupt requests from the $\overline{\text{IRQ0}}$ pin are enabled. | H'FFF4<br>Bit 0 | <br>1 |
| IRR1 | IRRI0 | Interrupt flag register 1 (IRQ0 interrupt request flag):<br>When IRRI0 is cleared to 0, no IRQ0 interrupt is requested.<br>When IRRI0 is set to 1, an IRQ0 interrupt is requested. | H'FFF6<br>Bit 0 | <br>0 |

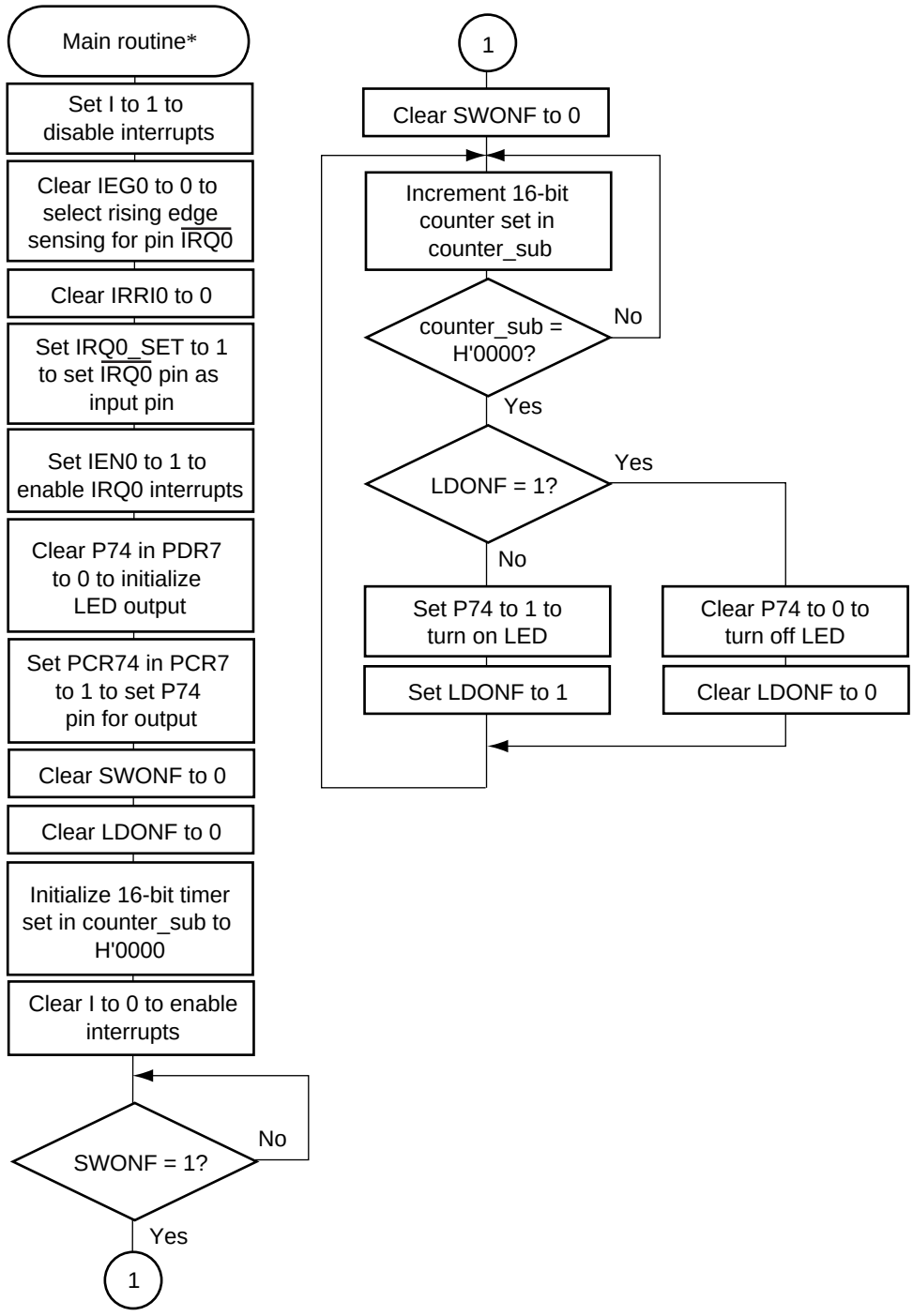
## 4.4 Description of RAM

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 up-counter that turns on or off the LED when it overflows | H'FB80 | Main routine |
| USRF | SWONF | Flag for judging on/off of the switch | H'FB82<br>Bit 0 | Main routine<br>Switch on |
| | LDONF | Flag for judging on/off of the LED | H'FB82<br>Bit 1 | Main routine |

## 5. Flowcharts




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

**Figure 3 Flowchart for Main Routine**

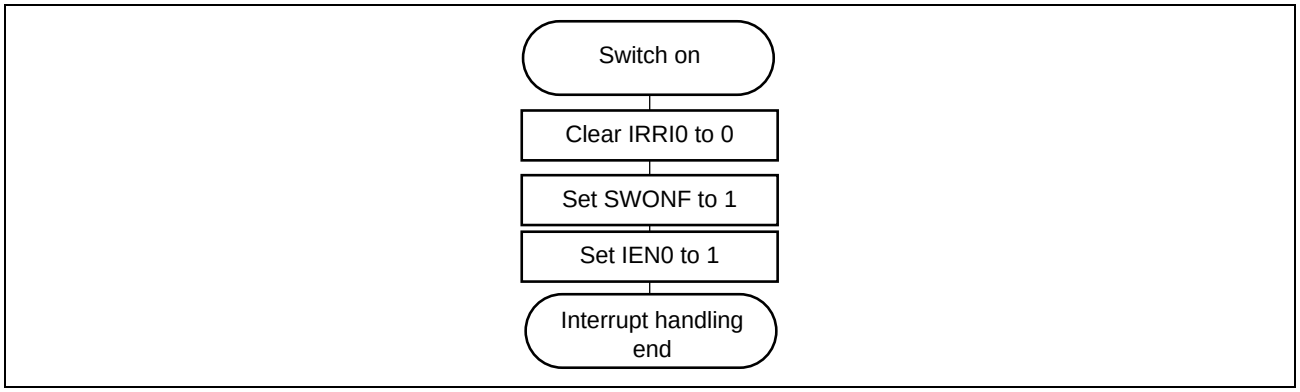



**Figure 4 Flowchart for IRQ0 Interrupt Handling Routine**

## 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                                      */
/*                                                          */
/*    'Count Start by External Interrupt'                   */
/*                                                          */
/*    Function                                              */
/*    : External Interrupt                                  */
/*                                                          */
/*    External Clock : 16MHz                                */
/*    Internal Clock : 16MHz                                */
/*    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        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 bit4       */
#define        IEGR1_BIT   (*(struct BIT *)0xFFF2)     /* Interrupt Edge Select Register 1 */
#define        IEG0        IEGR1_BIT.b0                /* IRQ0 Edge Select                 */
#define        IENR1_BIT   (*(struct BIT *)0xFFF4)     /* Interrupt Enable Register 1      */
#define        IEN0        IENR1_BIT.b0                /* IRQ0 Interrupt Enable            */
#define        IRR1_BIT    (*(struct BIT *)0xFFF6)     /* Interrupt Request Register 1     */
#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        */

#pragma        interrupt    (IRQ0)
/************************************************************/
/*     Function Definition                                  */
/************************************************************/
extern    void    INIT ( void );                     /* SP Set                           */
void      main   ( void );
void      IRQ0   ( void );
void      wait   ( void );
```

```
/************************************************************/
/*    RAM define                                            */
/************************************************************/

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

#define        USRF_BIT    (*(struct BIT *)&USRF)
#define        SWONF        USRF_BIT.b0 /* Switch On Flag                          */
#define        LDONF        USRF_BIT.b1 /* 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) = {
    IRQ0                                /* IRQ0  Interrupt                         */
};
#pragma    section                       /* P                                       */
/************************************************************/
/*    Main Program                                          */
/************************************************************/
void    main ( void )
{

    set_imask_ccr(1);                   /* Interrupt Disable                       */

    IEG0 = 1;                           /* Initialize IRQ0 Terminal Input Edge     */

    IRRI0 = 0;                          /* Initialize IRRI0                        */

    IRQ0_SET = 1;                       /* Initialize Input TerminalIRQ0           */

    IEN0 = 1;                           /* IRQ0 Interrupt Request Enable           */

    P74 = 0;                            /* Initialzie LED Output                   */
```

```
    PCR74 = 1;                          /* Initialize P74 Input-Output Terminal Function */

    SWONF = 0;                          /* Initialize SWONF                              */
    LDONF = 0;                          /* Initialize LDONF                              */

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

    set_imask_ccr(0);                   /* Interrupt Enable                              */

    while(SWONF != 1){                  /* SWONF = 0 ?                                   */
        ;
    }

    SWONF = 0;                          /* Clear SWONF                                   */

    while(1){
        do{
            counter_sub++;              /* Increment 16bit Counter                       */
        }while(counter_sub != 0x0000);  /* 16bit Counter = H'0000 ?                      */

        if(LDONF == 1){                 /* LDONF = 1 ?                                   */
            P74 = 0;                    /* Turn Off LED                                  */
            LDONF = 0;                  /* Clear LDONF                                   */
        }
            else{
                P74 = 1;                /* Turn On LED                                   */
                LDONF = 1;              /* Set LDONF                                     */
            }
    }

}
```

```
/************************************************************/
/*    IRQ0 Interrupt                                        */
/************************************************************/
void    IRQ0 ( void )
{

    IRRI0 = 0;                          /* Clear    IRRI0                          */

    SWONF = 1;                          /* Set SWONF                               */

    IEN0 = 0;                           /* IRQ0 Interrupt Disable                  */

}
```

**Link Address Setting:**

| Section Name | Address |
|---|---|
| CV1 | H'0000 |
| CV2 | H'001C |
| 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.