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

WKP Interrupt

Introduction

By turning on a switch input connected to the $\overline{\text{WKP}}$ pin, a WKP interrupt is generated, and counting-up of a 16-bit counter which is set in a two-byte variable (counter_sub) starts.

Target Device

H8/3664

Contents

1. Specifications

1. By turning on a switch input connected to the $\overline{\text{WKP}}$ pin, a WKP interrupt is generated, and counting-up of a 16-bit counter which is set in a two-byte variable (counter_sub) starts.
2. The WKP interrupt is requested by detection of the falling edge of the input to the $\overline{\text{WKP}}$ pin.
3. During WKP interrupt handling, counting-up of the 16-bit counter value set in counter_sub starts.
4. Each time the 16-bit counter set in counter_sub overflows, an LED is lit or extinguished.
5. The LED is assumed to be connected to the P74 output pin of port 7.
6. Figure 1.1 shows an example of connection of a switch to the $\overline{\text{WKP}}$ input pin.

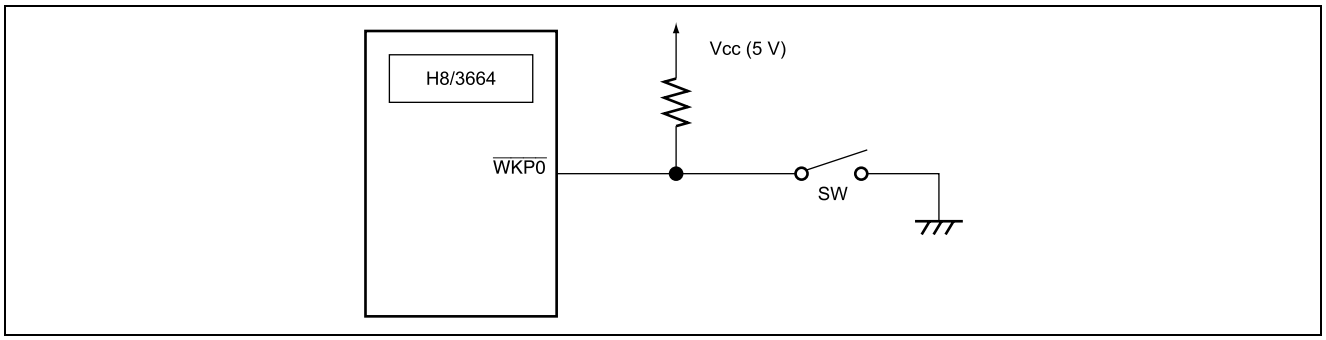


Figure 1.1 LED lighting/extinction operation

2. Description of Functions

1. In this task example, the counter is started by a WKP interrupt.
   Below, the WKP interrupt is explained.
   - The WKP interrupt is requested by an input signal to pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$. Rising or falling edge sensing for the WKP interrupt can be specified using the interrupt edge select register 2 (IEGR2) bits WPEG5 to WPEG0.
   - Pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ are also used for port 5. When using these pins as $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ input pins, the WKP5 to WKP0 bits of the port mode register 5 (PMR5) are set to 1.
   - When, with the pin functions for pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ selected using the port mode register 5 (PMR5), the specified edge is input, the corresponding bits IWPF5 to IWPF0 of the wakeup interrupt flag register (IWPR) are set to "1", and an interrupt request is generated.
   - Acceptance of an interrupt request can be prohibited by clearing IENWP in the interrupt enable register 1 (IENR1) to "0".
   - By setting the I bit of the condition code register (CCR) to 1, all interrupts can be prohibited.
   - Interrupt operation is described below.
     a. With the corresponding bits of the interrupt enable register set to 1, when an interrupt request occurs, an interrupt request signal is sent to the interrupt controller.
     b. When the interrupt request signal is sent to the interrupt controller, the interrupt request flag is set.
     c. Among interrupts for which the interrupt request flag is set to 1, the highest-priority interrupt request is selected according to the order of priority, and the others are held.
     d. The CCR I bit is referenced, and if the I bit is cleared to 0 then interrupt requests are accepted, but if the I bit is set to 1 then interrupt requests are held.
     e. When an interrupt is accepted, after the instruction being executed at that time ends, the program counter (PC) and CCR are saved to the stack area. The PC saved on the stack indicates the start address for execution on return.
     f. The CCR I bit is set to 1. As a result, all interrupts are masked.
     g. A vector address is generated for the accepted interrupt, and execution of the interrupt processing routine is begun from the address indicated by the contents of the vector address. When interrupts are disabled by clearing the interrupt enable register 1, or when the interrupt flag register 1 is cleared, interrupts should always be masked (I=1). If such operations are performed while I=0, if there is contention between execution of the operation instruction and the occurrence of the interrupt, exception handling for the interrupt which has occurred is executed at the end of execution of the operation instruction.

2. Table 2.1 indicates function allocations in this task example.
   The function allocations are indicated in Table 1, and operations to light and extinguish the LED connected to the I/O port are performed.

Table 2.1 Function Allocation

Function	Function allocation
WPEG0	Sets detection edge direction for $\overline{\text{WKP}}$ pin input
PCR7	Sets P74 output pin function
PDR7	Stores data of P74 output pin
P74	Output pin for LED output

3. Description of Operations

Figure 3.1 explains the operation. Through the hardware and software processing shown in the figure, after a WKP interrupt is generated, the LED connected to the I/O port is lit and extinguished.

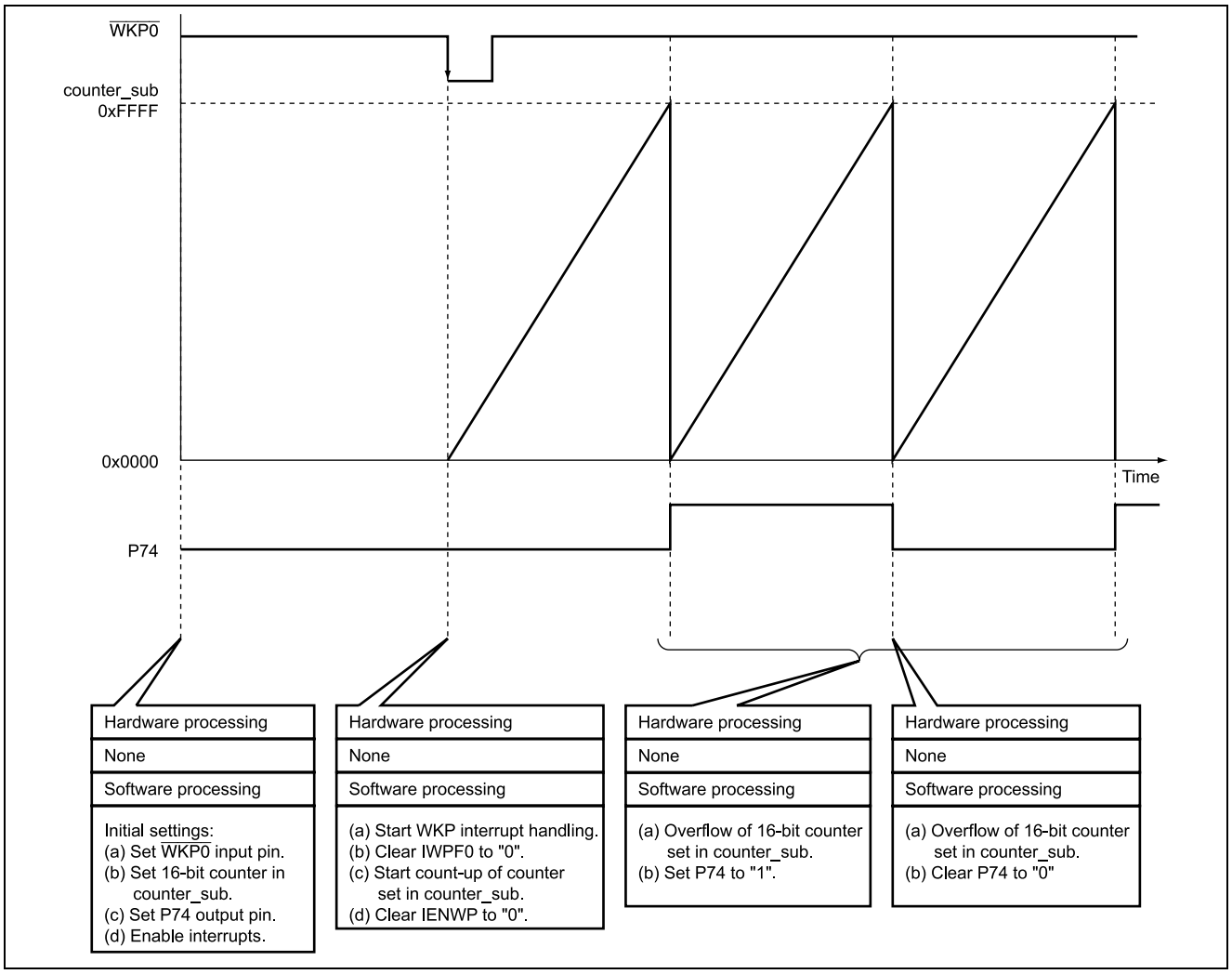


Figure 3.1 Explanation of operation to light and extinguish LED connected to the I/O port

4. Description of software

4.1 Description of Modules

Table 4.1 explains the modules in this task example.

Table 4.1 Description of Modules

Module name	Label name	Function
Main routine	main	Sets direction of input edge for WKP interrupt, sets LED output pin, increments 16-bit counter, and performs LED output
Switch on	WKP	In the WKP interrupt routine, sets SWO NF to 1

4.2 Description of Arguments

This sample task uses no arguments.

4.3 Description of Internal Registers

The internal registers used in this sample task are described in table 4.2.

Table 4.2 Description of Internal Registers

Register	Name	Function	Address	Setting
PDR7	P74	Port data register 7 (port data register 74) P74 = 0: The pin P74 output level is low P74 = 1: The pin P74 output level is high	H'FFDA Bit 4	0
PCR7	PCR74	Port control register 7 (port control register 74) PCR74 = 1: The I/O pin P74 functions as an output pin	H'FFEA Bit 4	1
IEGR2	WPEG0	Interrupt edge select register 2 (WKP0 edge select) WPEG0 = 0: Detects the falling edge of the $\overline{\text{WKP0}}$ pin input	H'FFF3 Bit 0	0
IENR1	IENWP	Interrupt enable register 1 (WKP0 interrupt enable) IENWP = 1: Enables $\overline{\text{WKP0}}$ pin input interrupt requests	H'FFF4 Bit 5	1
IWPR	IWPF0	Interrupt flag register (WKP0 interrupt request flag) IWPF0 = 0: A WKP0 interrupt is requested IWPF0 = 1: A WKP0 interrupt is not requested	H'FFF8 Bit 0	0
PMR5	WKP0	Selects general I/O port/WKP0 pin function WKP0 = 0: Selects general I/O output port WKP0 = 1: selects $\overline{\text{WKP0}}$ input pin	H'FFE1 Bit 0	1

4.4 Description of RAM Used

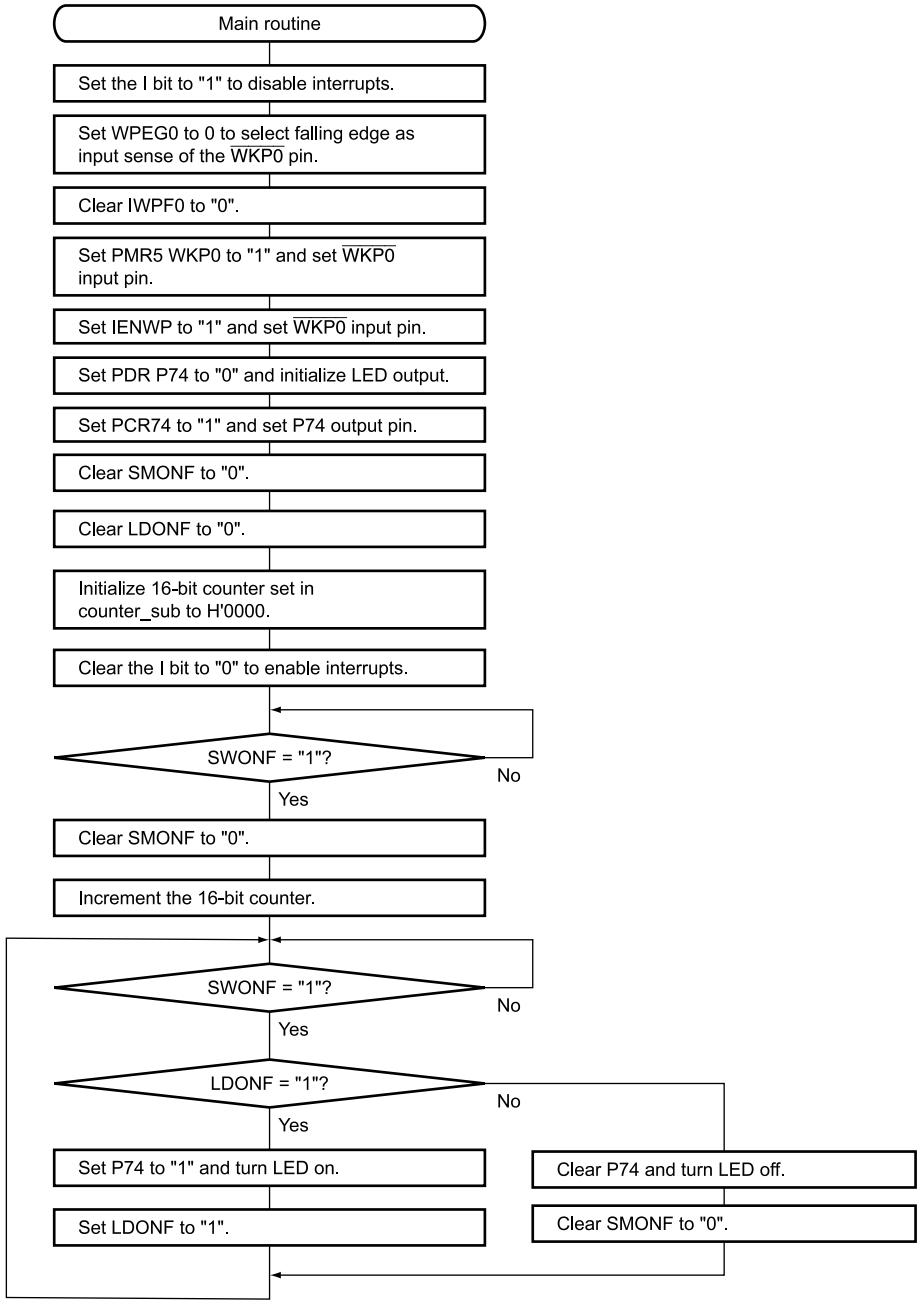
Table 4.3 describes the RAM used in this sample task.

Table 4.3 Description of RAM

Label Name		Function	Address	Used in
Counter_sub		16-bit up-counter which lights and extinguishes the LED each time overflow occurs	H'FB80	Main routine
USRF	SWONF	Flag to judge whether switch input is on or off	H'FB82 Bit 0	Main routine Switch on
	LDONF	Flag to judge whether LED is on or off	H'FB82 Bit 1	Main routine

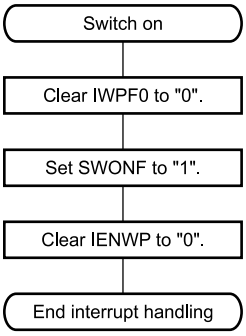
5. Flowcharts

1. Main routine



Note: * In this example, the stack pointer is set using INIT.SRC (assembly language).

2. WKP0 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                                                                                                                           */
/*                                                                                                                                              */
/*   'Wake Up Interrupt function                                                                                                                */
/*                                                                                                                                              */
/      Function                                                                                                                                  */
/*   : Wake Up                                                                                                                                  */
/*                                                                                                                                              */
/*   External Clock : 16MHz                                                                                                                     */
/*   Internal Clock : 16MHz                                                                                                                     */
/*   Sub Clock      : 32.768kHz                                                                                                                 */
/*                                                                                                                                              */
/************************************************************************************************************************************************/

#include    <C:\ch38\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   IEGR2          *(volatile unsigned char *)0xFFF3                    /* Interrupt Select Edge Register 2         */
#define   IEGR2_BIT           (*(struct BIT *)0xFFF3)                         /* Interrupt Select Edge Register 2         */
#define   WPEG0               IEGR2_BIT.b0                                    /* Wake Up 0 Edge Select                    */
#define   INER1               *(volatile unsigned char *)0xFFF4               /* Interrupt Enable Register 1              */
#define   INER1_BIT      (*(struct BIT *)0xFFF4)                              /* Interrupt Enable Register 1              */
#define   IENWP               INER1_BIT.b5                                    /* Wake Up Interrupt Enbable                */
#define   IWPR           *(volatile unsigned char *)0xFFF8                    /* Interrupt Flag Register                  */
#define   IWPR_BIT       (*(struct BIT *)0xFFF8)                              /* Interrupt Flag Register                  */
#define   IWPF0               IWPR_BIT.b0                                     /* Wake Up0 Interuput Request Flag          */
#define   PMR5           *(volatile unsigned char *)0xFFE1                    /* Port Mode Register 5                     */
#define   PMR5_BIT       (*(struct BIT *)0xFFE1)                              /* Port Mode Register 5                     */
#define   WKP0           PMR5_BIT.b0                                          /* WKP0 Function Select Bit                 */
```

```
#define   PCR7          *(volatile unsigned char *)0xFFEA                    /* Port Control Register 7                */
#define   PCR7_BIT      (*(struct BIT *)0xFFEA)                              /* Port Control Register 7                */
#define   PCR74              PCR7_BIT.b4                                     /* Port Control Register Bit 4            */
#define   PDR7          *(volatile unsigned char *)0xFFDA                    /* Port Data Register 7                   */
#define   PDR7_BIT      (*(struct BIT *)0xFFDA)                              /* Port Data Register 7                   */
#define   P74           PDR7_BIT.b4                                          /* Port Data Register Bit 4               */

#pragma       interrupt    (WKP_0)
/************************************************************************************************************************/
/*   RAM define                                                                                                         */
/************************************************************************************************************************/
unsigned int        counter_sub;                                            /* 16 Bit Up Counter                      */
unsigned char       USRF;                                                   /* User Flag Erea                         */
#define             USRF_BIT    (*(struct BIT *)&USRF)
#define             SWONF         USRF_BIT.b0                               /* Swich ON/OFF Judgment Flag             */
#define             LDONF         USRF_BIT.b1                               /* LED ON/OFF Judgment Flag               */

extern void _INITSCT();
/************************************************************************************************************************/
/*   Function definition                                                                                                */
/************************************************************************************************************************/
extern  void   INIT( void );                                                /* SP Set                                 */
extern  void   WKP_0( void );                                               /* Wake Up 0 Interrupt Routine            */
void    main   ( void );
/************************************************************************************************************************/
/*   Vector Address                                                                                                     */
/************************************************************************************************************************/
#pragma section     V1                                                      /* VECTOR SECTOIN SET                     */
void (*const VEC_TBL1[])(void) = {                                          /* 0x00 - 0x0f                            */
     INIT                                                                   /* 00 Reset                               */
};

#pragma section     V2                                                      /* VECTOR SECTOIN SET                     */
void (*const VEC_TBL2[])(void) = {                                          /* 0x24 - 0x25                            */
     WKP_0                                                                  /* 18 WKP0                                */
};

#pragma section                                                             /* P                                      */
/************************************************************************************************************************/
/*   Main Program                                                                                                       */
/************************************************************************************************************************/
void main ( void )
{
     _INITSCT();

     set_imask_ccr(1);                                                      /* Interrupt Disable                      */

     WPEG0 = 0;                                                             /* WKP0 Falling Edge Interrupt Select     */

     WKP0 = 1;                                                              /* WKP0 Input Pin Select                  */

     IENWP = 1;                                                             /* WKP0 Interrupt Enable                  */

     P74 = 0;                                                               /* Port74 "0" Output                      */

     PCR7 = 0x10;                                                           /* Port74 Output                          */
     SWONF = 0;                                                             /* Swich ON/OFF Judgment Flag Clear       */

     LDONF = 0;                                                             /* LED ON/OFF Judgment Flag Clear         */
```

```
    counter_sub = 0x0000;                                   /* 16 Bit Counter Clear                 */

    set_imask_ccr(0);                                       /* Interrupt Enable                     */

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

    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                            */
            }
        }
}

/*****************************************************************************************************************************************/
/*   Wake Up Interrupt                                                                                                                   */
/*****************************************************************************************************************************************/
void WKP_0( void )
{
    IWPF0 = 0;                                              /* Clear IWPF0 to 0                     */

    SWONF = 1;                                              /* Set SWONF to 1                       */

    IENWP = 0;                                              /* Clear IENWP to 0                     */
}
```

Link address specifications

Section Name	Address
CV1	H'0000
CV2	H'0016
P	H'0100

Revision Record

		Description	
Rev.	Date	Page	Summary
1.00	Sep.29.03	—	First edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corporation 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 Corporation 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 Corporation or a third party.
2. Renesas Technology Corporation 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 Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation 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 Corporation 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 Corporation by various means, including the Renesas Technology Corporation 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 Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation 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 Corporation or an authorized Renesas Technology Corporation 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 Corporation 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 Corporation for further details on these materials or the products contained therein.