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.

RENESAS

H8/300L SLP Series

Transition to High-Speed Sleep Mode

Introduction

An IRQ0 interrupt is generated during program execution in the active mode, and the system enters a sleep mode (high speed) by executing a SLEEP instruction after the IRQ0 interrupt handling is completed.

Target Device

H8/38024

Contents

1. Specifications

1. This sample task shows an example of making transition to the high-speed sleep mode.

2. An IRQ0 interrupt is generated by the input from a switch on the $\overline{\text{IRQ0}}$ pin during program execution in the active mode. The system enters a sleep mode (high speed) by executing a SLEEP instruction after the IRQ0 interrupt handling is completed.

3. A Timer A interrupt is generated one second after the transition to the sleep mode. This causes the system to leave the sleep mode and return to the high-speed active mode.

4. The LED is turned when the program is started, and turned off when the system enters the sleep mode. The LED is turned on again upon transition to the active mode.

5. The LED is connected to the P92 output pin of port 9.

6. P92 is a large-current port.

7. Figure 1.1 shows an example of connecting a switch to the $\overline{\text{IRQ0}}$ pin.
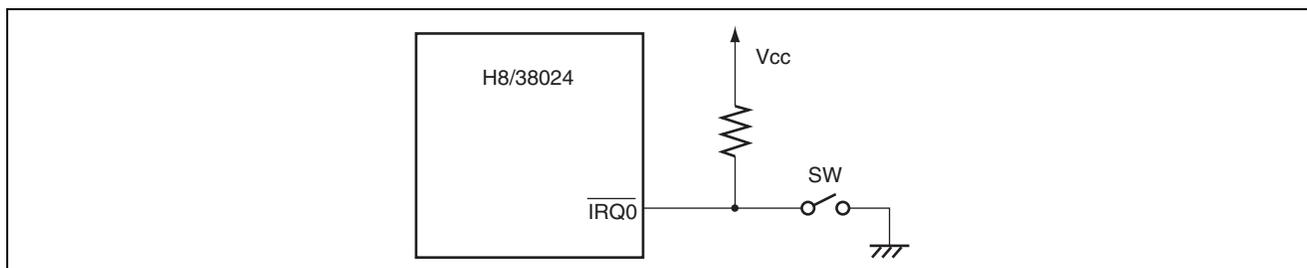
Figure 1.1   Example of Switch Connection for Transition to High-Speed Sleep Mode

2. Description of Functions

1. In this sample task, transition to the high-speed sleep mode, a power down mode, is made. Figure 2.1 shows a mode transition diagram to the high-speed sleep mode. The function of the high-speed sleep mode is described below.

   — When a SLEEP instruction is executed in the active mode while SSBY and LSON in SYSCR1 are both set to 0, and MSON and DTON in SYSCR2 are both set to 0, the system enters the high-speed sleep mode.

   — During a sleep mode, the CPU operation is halted but on-chip peripheral modules operate.

   — Sleep modes can be terminated by any interrupt (Timer A, Timer C, Timer F, Timer G, asynchronous event counter, IRQAEC, IRQ4, IRQ3, IRQ1, IRQ0, WKP7 to WKP0, SCI3 and A/D converter) or by $\overline{\text{RES}}$ pin input.

   — In the case of terminating the mode by an interrupt, the high-speed sleep mode is terminated upon generation of an interrupt and the interrupt exception handling starts. From the high-speed sleep mode, transition to the high-speed active mode is possible.

   — The high-speed sleep mode is not cleared if the I bit in CCR is set to 1 or acceptance of the interrupt is disabled by the interrupt enable register.

   — In the case of terminating the mode by the $\overline{\text{RES}}$ pin, when the $\overline{\text{RES}}$ pin is driven "Low", the system enters a reset state and the high-speed sleep mode thus ends.

   — In this sample task, the high-speed sleep mode is terminated by Timer A interrupt.
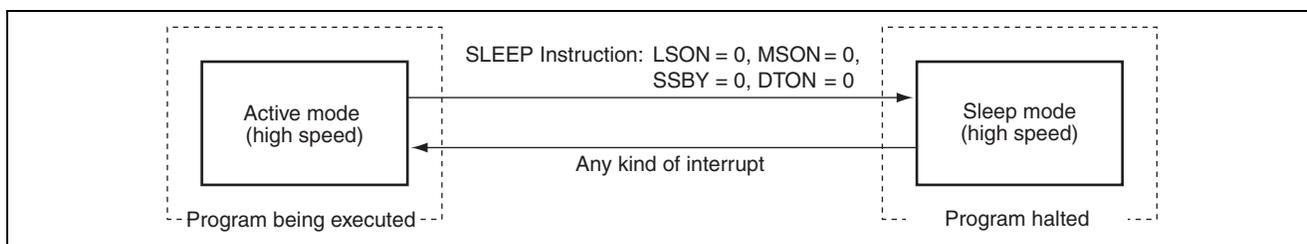
Figure 2.1 Mode Transition from/to High-Speed Sleep Mode

2. Table 2.1 shows the function assignments in this sample task. Transition to the high-speed sleep mode is carried out by assigning the functions as shown below.

Table 2.1   Assignment of Functions

Function	Assigned Function
PSW	A 5-bit up counter using the subclock (32.768 kHz)/4 as input.
SYSCR1	Controls power down modes.
SYSCR2	Controls power down modes.
TMA	Sets TCA overflow cycle to 1 sec.
TCA	1 sec. timer
PDR9	P92 output pin data storage
IEN0	Enables $\overline{IRQ0}$ pin interrupt requests.
IEG0	Selects $\overline{IRQ0}$ pin input edge.
IENTA	Enables Timer A interrupt requests.
IRRTA	Indicates whether or not a Timer A interrupt request has been generated.
IRRI0	Indicates whether or not an IRQ0 interrupt request has been generated.
P92	LED output
$\overline{IRQ0}$	Switch input

3. Principle of Operation

1. Figure 3.1 illustrates the operation of this sample task. Transition to the high-speed sleep mode is carried out through hardware and software processing as shown in the figure.
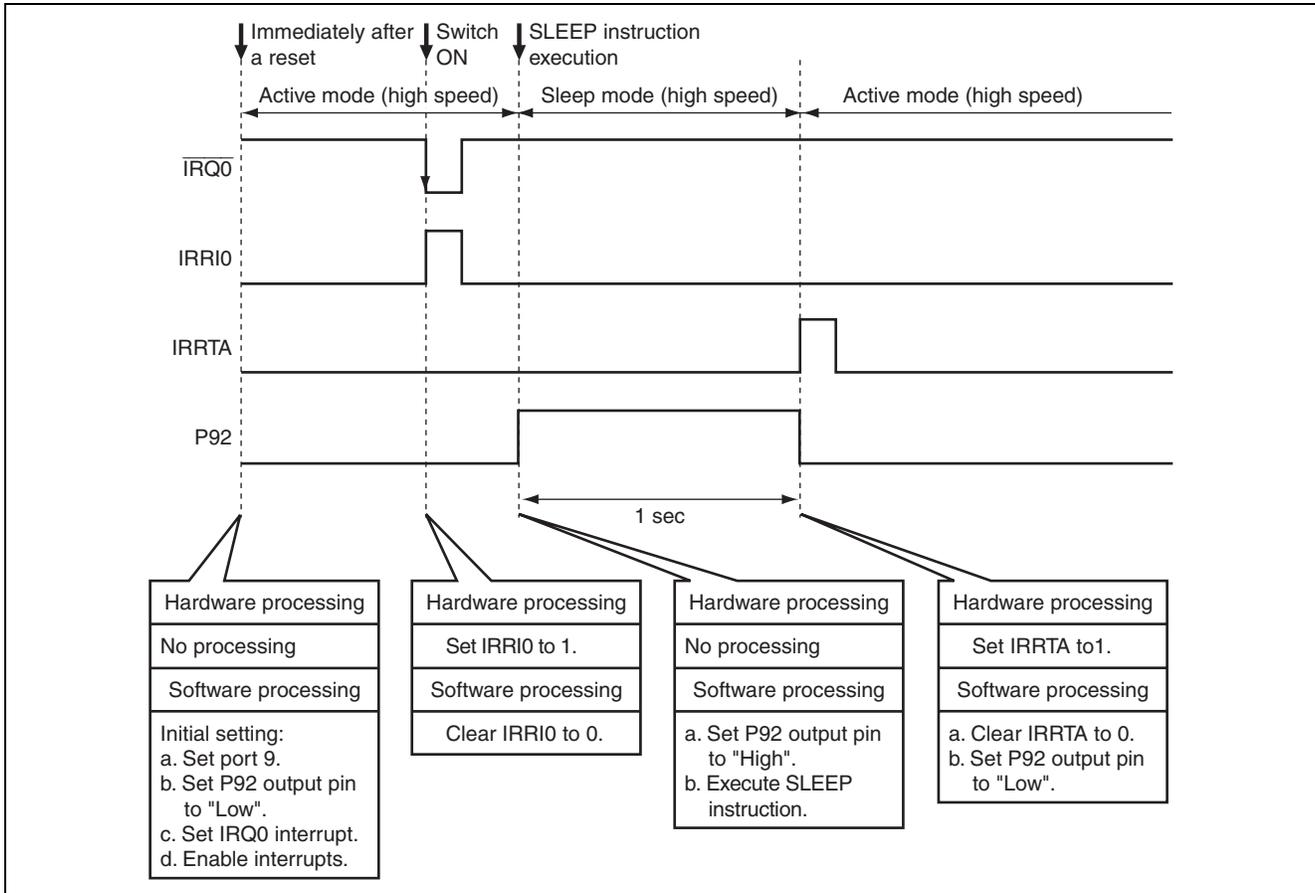
Figure 3.1   Operation Principle of Making Transition to High-Speed Sleep Mode

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 IRQ0 interrupt, port 9, and Timer A interrupt, enables interrupts, controls the LED, and makes transition to the high-speed sleep mode.
Switch ON	Irq0int	An IRQ0 interrupt handling routine which sets SWONF and disables IRQ0 interrupts.
Terminate high-speed sleep mode	taint	A Timer A interrupt handling routine which disables Timer A interrupt.

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	H'FFB0	H'18
		If TMA = H'18, Timer A function is set to clock time-base function, TCA input clock source to PSW and TCA overflow cycle to 1 sec.		
TCA		Timer Counter A	H'FFB1	H'00
		An 8-bit up-counter using clock input of 32.768 kHz / 32.		
PDR9	P92	Port Data Register 9 (Port Data Register 92)	H'FFDC	0
		If P92 = 0, the output level on pin P92 is Low.	Bit 2	
		If P92 = 1, the output level on pin P92 is High.		
SYSCR1	SSBY	System Control Register 1 (Software Standby)	H'FFF0	0
		If SSBY = 0, a transition is made to the sleep mode after a SLEEP instruction is executed in the active mode. A transition is made to the subsleep mode after a SLEEP instruction is executed in the subactive mode.	Bit 7	
	LSON	System Control Register 1 (Low speed ON flag)	H'FFF0	0
		If LSON = 0, the CPU operating clock is set to the system clock when the watch mode is terminated.	Bit 3	

Register		Function	Address	Setting
SYSCR2	DTON	System Control Register 2 (Direct Transition ON Flag) If DTON = 0, a transition is made to the standby, watch or sleep mode when a SLEEP instruction is executed in the active mode. If DTON = 1, A direct transition is made to the high-speed active mode (when SSBY = 0, MSON = 0, LSON = 0) or to the subactive mode (when SSBY = 1, MSON = 1, LSON = 1) when a SLEEP instruction is executed in the high-speed active mode.	H'FFF1 Bit 3	0
	MSON	System Control Register 2 (Medium Speed ON Flag) If MSON = 0, the system operates in the high-speed active mode after the standby, watch or sleep mode is terminated. The system operates in the high-speed sleep mode if a SLEEP instruction is executed in the active mode. If MSON = 1, the system operates in the high-speed active mode after the standby, watch or sleep mode is terminated. The system operates in the medium-speed sleep mode if a SLEEP instruction is executed in the active mode.	H'FFF1 Bit 2	0
IEGR	IEG0	IRQ Edge Select Register (IRQ0 Edge Select) If IEG0 = 0, falling edge is selected for edge detection of $\overline{IRQ0}$ pin input. If IEG0 = 1, rising edge is selected for edge detection of $\overline{IRQ0}$ pin input.	H'FFF2 Bit 0	0
IENR1	IENTA	Interrupt Enable Register 1 (Timer A Interrupt Enable) If IENTA = 0, Timer A interrupt requests are disabled. If IENTA = 1, Timer A interrupt requests are enabled.	H'FFF3 Bit 7	0
	IEN0	Interrupt Enable Register 1 (IRQ0 Interrupt Enable) If IEN0 = 0, $\overline{IRQ0}$ pin interrupt requests are disabled. If IEN0 = 1, $\overline{IRQ0}$ pin interrupt requests are enabled.	H'FFF3 Bit 0	1
IRR1	IRRTA	Interrupt Request Register 1 (Timer A Interrupt Request Flag) If IRRTA = 0, Timer A interrupt is not requested. If IRRTA = 1, Timer A interrupt has been requested.	H'FFF6 Bit 7	0
	IRRI0	Interrupt Request Register 1 (IRQ0 Interrupt Request Flag) If IRRI0 = 0, IRQ0 interrupt is not requested. If IRRI0 = 1, IRQ0 interrupt has been requested.	H'FFF6 Bit 0	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	SWONF	Flag to judge whether the switch input is on or off.	H'FB80 Bit 0	Main routine Switch ON

5. Flowchart

1. Main routine

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

2. IRQ0 interrupt handling routine

```
           ( Switch ON )
                 |
      [ Clear IRRI0 to 0. ]
                 |
      [ Set SWONF to 1. ]
                 |
   [ Clear IEN0 to 0 to disable IRQ0
     interrupts. ]
                 |
    ( End of interrupt handling )
```

3. Timer A interrupt handling routine

```
   ( Terminate high-speed sleep mode )
                 |
      [ Clear IRRTA to 0. ]
                 |
   [ Clear IENTA to 0 to disable Timer A
     interrupt. ]
                 |
    ( End of interrupt handling )
```
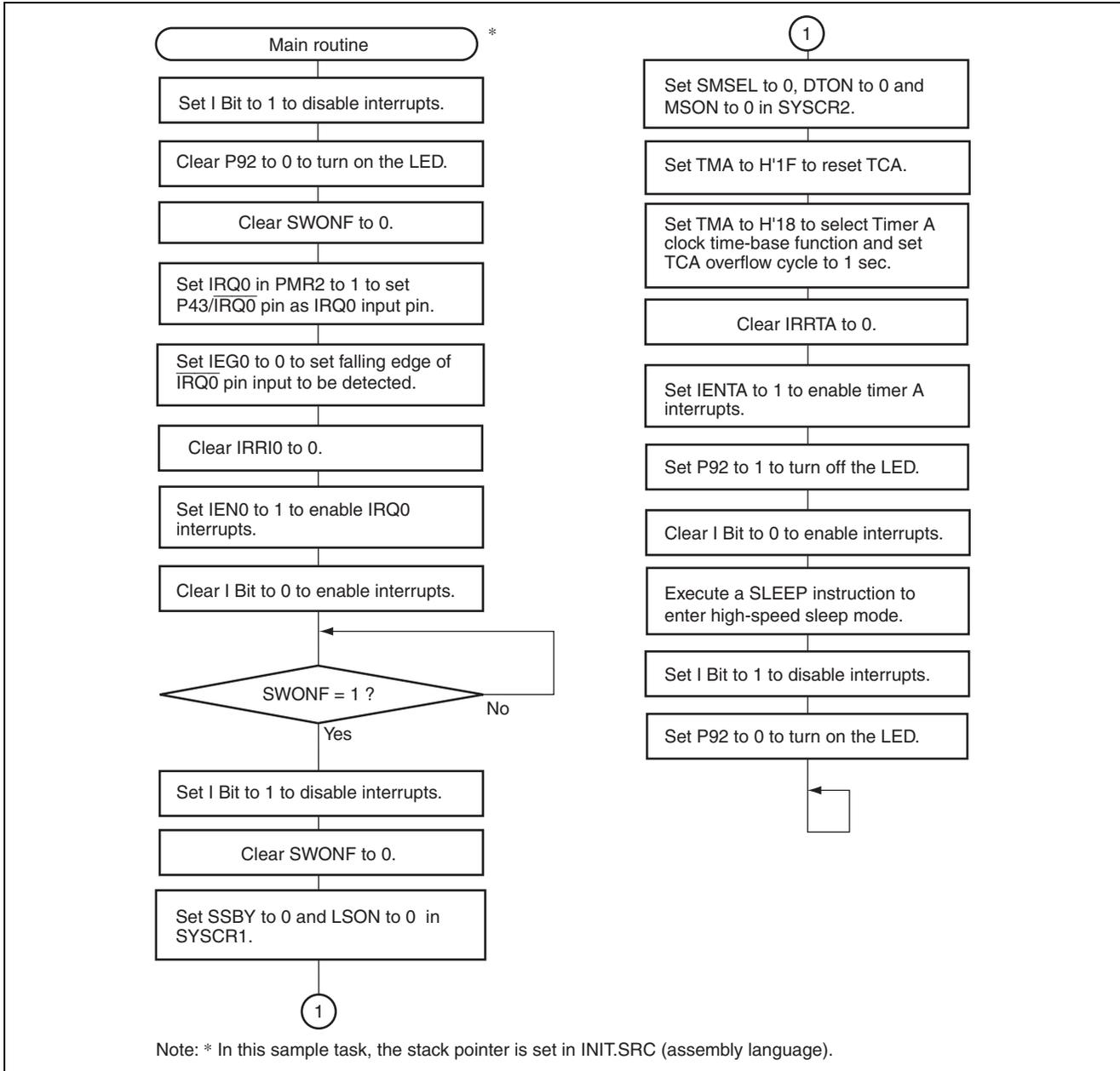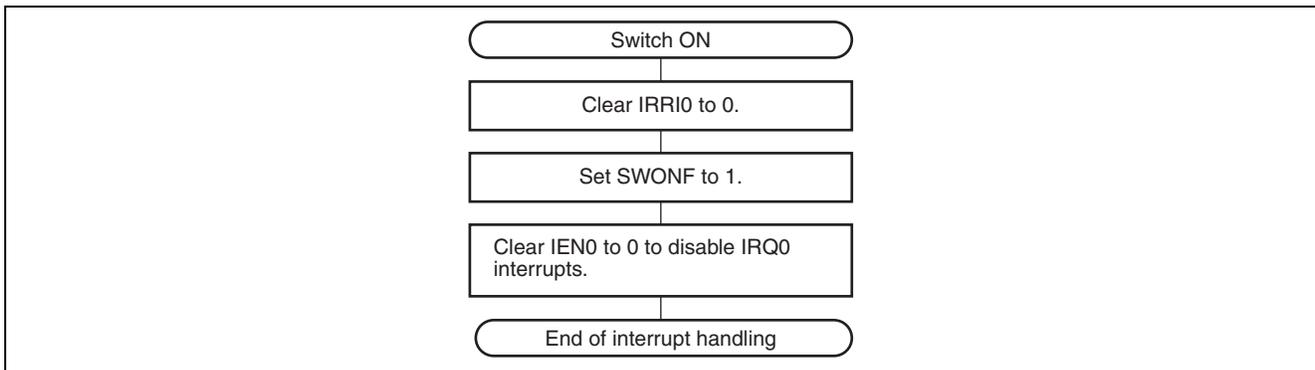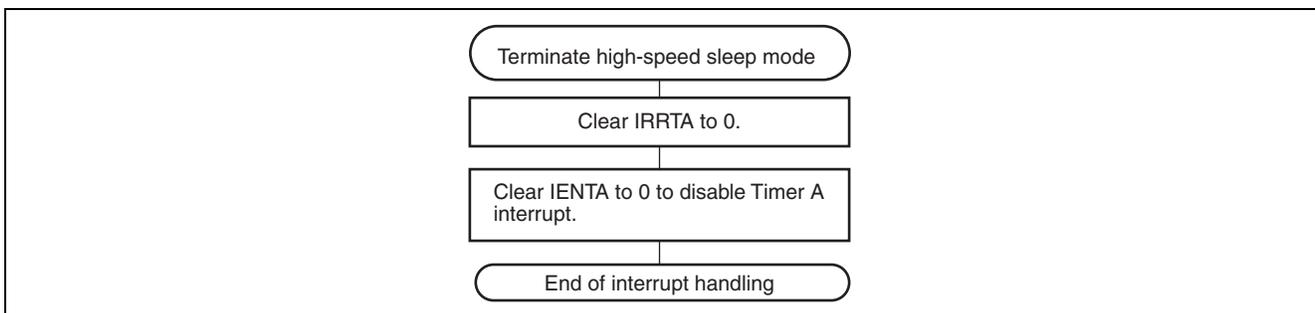
6. Program Listing

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
```

```c
/******************************************************************************************************/
/*                                                                                                    */
/*  H8/300L Super Low Power Series                                                                    */
/*      -H8/38024 Series-                                                                              */
/*  Application Note                                                                                   */
/*                                                                                                    */
/*  'Transition to Sleep (High-Speed) Mode'                                                            */
/*                                                                                                    */
/*  Function                                                                                           */
/*  : Power-Down Mode                                                                                  */
/*     Sleep (High-Speed) Mode                                                                         */
/*                                                                                                    */
/*  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    PMR2_BIT   (*(struct BIT *)0xFFC9)               /*Port Mode Register 2              */
#define    IRQ0       PMR2_BIT.b0                           /*P43/IRQ0 Select                   */
#define    PDR9_BIT   (*(struct BIT *)0xFFDC)               /*Port Data Register 9              */
#define    P92        PDR9_BIT.b2                           /*Port Data Register 92             */
#define    SYSCR1     *(volatile unsigned char *)0xFFF0     /*System Control Register 1         */
#define    SYSCR1_BIT (*(struct BIT *)0xFFF0)               /*System Control Register 1         */
#define    SSBY       SYSCR1_BIT.b7                         /*Software Standby                  */
#define    STS2       SYSCR1_BIT.b6                         /*Standby Timer Select 2            */
```

```
#define    STS1       SYSCR1_BIT.b5                  /*Standby Timer Select 1              */
#define    STS0       SYSCR1_BIT.b4                  /*Standby Timer Select 0              */
#define    LSON       SYSCR1_BIT.b3                  /*Low Speed On Flag                   */
#define    MA1        SYSCR1_BIT.b1                  /*Active Mode Clock Select 1          */
#define    MA0        SYSCR1_BIT.b0                  /*Active Mode Clock Select 0          */
#define    SYSCR2     *(volatile unsigned char *)0xFFF1   /*System Control Register 2      */
#define    SYSCR2_BIT (*(struct BIT *)0xFFF1)        /*System Control Register 2           */
#define    NESEL      SYSCR2_BIT.b4                  /*Noise Elimination Sampling          */
                                                     /*                 Frequency Select  */
#define    DTON       SYSCR2_BIT.b3                  /*Direct Transfer On Flag             */
#define    MSON       SYSCR2_BIT.b2                  /*Middle Speed On Flag                */
#define    SA1        SYSCR2_BIT.b1                  /*Subactive Mode Clock Select 1       */
#define    SA0        SYSCR2_BIT.b0                  /*Subactive Mode Clock Select 0       */
#define    IEGR1_BIT  (*(struct BIT *)0xFFF2)        /*Interrupt Edge Select Register 1    */
#define    IEG0       IEGR1_BIT.b0                   /*IRQ0 Edge Select                    */
#define    IENR1_BIT  (*(struct BIT *)0xFFF3)        /*Interrupt Enable Register 1         */
#define    IENTA      IENR1_BIT.b7                   /*Timer A Interrupt Enable            */
#define    IEN0       IENR1_BIT.b0                   /*IRQ0 Interrupt Enable               */
#define    IENR2_BIT  (*(struct BIT *)0xFFF4)        /*Interrupt Enable Register 2         */
#define    IENDT      IENR2_BIT.b7                   /*Direct Transfer Interrupt Enable    */
#define    IRR1_BIT   (*(struct BIT *)0xFFF6)        /*Interrupt Request Register 1        */
#define    IRRTA      IRR1_BIT.b7                    /*Timer A Interrupt Request Flag      */
#define    IRRI0      IRR1_BIT.b0                    /*IRQ0 Interrupt Request Flag         */
#define    IRR2_BIT   (*(struct BIT *)0xFFF7)        /*Interrupt Request Register 2        */
#define    IRRDT      IRR2_BIT.b7                    /*Direct Transfer Interrupt Request Flag */


#pragma interrupt   (irq0int)
#pragma interrupt   (taint)
/****************************************************************************************************/
/*  Function define                                                                               */
/****************************************************************************************************/
extern void INIT ( void );                           /*SP Set                              */
void       main ( void );
void       irq0int ( void );
void       taint ( void );


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

#define    USRF_BIT   (*(struct BIT *)&USRF)
#define    SWONF      USRF_BIT.b0                     /*Switch 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) = {
    irq0int                                             /* 0x0008 IRQ0 Interrupt Vector             */
};
#pragma section     V3                                  /* Vector Section Set                       */
void (*const VEC_TBL3[])(void) = {
    taint                                               /* 0x0016 timer A  Interrupt Vector         */
};

#pragma section                                         /*P                                         */
/****************************************************************************************************/
/*  Main Program                                                                                    */
/****************************************************************************************************/
void    main ( void )
{
    set_imask_ccr(1);                                   /*Interrupt Disable                         */

    P92 = 0;                                            /*Turn On LED                               */
    SWONF = 0;                                          /*Initialize SWONF                          */

    IRQ0 = 1;                                           /*Initialize Input Terminal IRQ0            */
    IEG0 = 0;                                           /*Set Rising Edge Of IEG0 Terminal Input    */
    IRRI0 = 0;                                          /*Clear IRRI0                               */
    IEN0 = 1;                                           /*IEN0 Interrupt Enable                     */

    set_imask_ccr(0);                                   /*Interrupt Enable                          */

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

    set_imask_ccr(1);                                   /*Interrupt Disable                         */

    SWONF = 0;                                          /*Clear SWONF                               */

    SYSCR1 = 0x07;                                      /* Set SYSCR1                               */
    SYSCR2 = 0xE0;                                      /* Set SYSCR2                               */

    TMA = 0x1F;                                         /*Initialize TCA                            */
    TMA = 0x18;                                         /*Initialize TCA Overflow Period            */
    IRRTA = 0;                                          /*Clear IRRTA                               */
    IENTA = 1;                                          /*Timer A Interrupt Enable                  */

    P92 = 1;                                            /*Turn Off LED                              */

    set_imask_ccr(0);                                   /*Interrupt Enable                          */
    sleep();                                            /*Transition to Sleep Mode                  */

    set_imask_ccr(1);                                   /*Interrupt Disable                         */
    P92 = 0;                                            /*Turn On LED                               */

    while(1){
```

```
        ;
    }
}


/*****************************************************************************************************/
/*  IRQ0 Interrupt                                                                                 */
/*****************************************************************************************************/
void   irq0int ( void )
{
    IRRI0 = 0;                                          /*Clear IRRI0                        */
    SWONF = 1;                                          /*Set SWONF                          */
    IEN0 = 0;                                           /*IEN0 Interrupt Disable             */
}


/*****************************************************************************************************/
/*  Timer A Interrupt                                                                              */
/*****************************************************************************************************/
void   taint ( void )
{
    IRRTA = 0;                                          /*Clear IRRTA                        */
    IENTA = 0;                                          /*Timer A Interrupt Disable          */
}
```

Link address specifications

Section Name	Address
CV1	H'0000
CV2	H'0008
CV3	H'0016
P	H'0100
B	H'FB80

Revision Record

Rev.	Date	Description	
		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.