To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

H8/300L SLP Series

Transition to Medium-Speed Active Mode

Introduction

Transition from the high-speed active mode to medium-speed active mode is performed through hardware and software processing.

Target Device

H8/38024

Contents

1. Specifications

1. This sample task shows an example of making transition to the medium-speed active mode.
2. The system enters the watch mode by executing a SLEEP instruction in the high-speed active mode.
3. The system enters the medium-speed active mode when a Timer A interrupt request is generated in the watch mode.
4. A Timer A interrupt request is generated every 0.5 sec by the clock time-base function. The LED is turned on and off in Timer A interrupt handling. After Timer A interrupt handling is completed, the operating mode again changes to the watch mode by execution of SLEEP instruction. The LED is turned on and off alternately every 0.5 sec.
5. The LED is connected to P92 output pin of port 9.
6. P92 is a large-current port.
7. A direct transition from the medium-speed active mode to the high-speed active mode takes place when the 60th Timer A interrupt request have been generated, that is, 30 sec. of time has elapsed.

2. Description of Functions

1. In this sample task, the operating mode is changed to the medium-speed active mode, a power down mode. Figure 2.1 shows a mode transition diagram to the medium-speed active mode. The function of the medium-speed active mode is described below.
   - The mode changes to the medium-speed active mode if LSON in SYSCR1 is 0 and MSON in SYSCR2 is 1 when an interrupt (IRQ1, IRQ0, or WKP7 to WKP0) is generated in the standby mode, an interrupt (Timer A, Timer F, Timer G, IRQ0, or WKP7 to WKP0) is generated in the watch mode, or any kind of interrupt is generated in the sleep mode (medium speed).
   - The mode does not change to the medium-speed active mode if the I bit in OCR is 1 or an acceptance of interrupts is disabled by the interrupt enable register.
   - The medium-speed active mode is terminated by a SLEEP instruction or by $\overline{\text{RES}}$ pin input.
   - When a SLEEP instruction is executed while SSBY is set to 1and LSON is set to 0 in SYSCR1, and TMA3 in TMA is set to 0, the medium-speed active mode is terminated and changes to the standby mode. When a SLEEP instruction is executed while SSBY in SYSCR1 is set to 1 and TMA3 in TMA is set to 1, the mode changes to the watch mode. When a SLEEP instruction is executed while SSBY is set to 0 and LSON is set to 0 in SYSCR1, the mode changes to the high-speed sleep mode if MSON in SYSCR2 is 0, or to the medium-speed sleep mode if MSON is 1. The mode changes to the high-speed active mode or to the subactive mode by direct transition.
   - 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 medium-speed active mode thus ends.
   - During the medium-speed active mode, the system operates at the clock frequency set by MA1 and MA0 in SYSCR1.
   - The CPU operates in three modes to execute programs, namely, the high-speed active mode, medium-speed active mode and subactive mode. A direct transition is a transition between these three operation modes which is made without stopping the program execution. A direct transition is made by setting DTON in SYSCR2 to 1 and executing a SLEEP instruction. After a direct transition, direct transition interrupt exception handling starts. If direct transition interrupts are disabled by the interrupt enable register 2, the mode changes to the sleep mode or watch mode instead. If the direct transition is attempted while the I bit in CCR is set to 1, the mode changes to the sleep mode or watch mode, and the mode cannot be terminated by an interrupt.
   - A direct transition from the medium-speed active mode to the high-speed active mode takes place when a SLEEP instruction is executed in the medium-speed active mode while SSBY is set to 0 and LSON is set to 0 in SYSCR1, and MSON is set to 0 and DTON is set to 1 in SYSCR2. The mode changes directly from the medium-speed active mode to the high-speed active mode via the sleep mode.
   - When changing to the medium-speed active mode by terminating the watch mode by a Timer A interrupt, the waiting time for the CPU and its peripheral functions till the clock stabilizes is set by STS2 to STS0 in SYSCR1. This setting must be made so that the waiting time will be longer than 10 ms in accordance with the operating frequency.

— In this sample task the waiting time is set to 1.638 ms.
— The operating clock in the medium-speed active mode is set by MA1 and MA0 in SYSCR1. In this sample task, the operating clock in the medium-speed active mode is set to $\phi_{osc}/128$, which makes 78.125 kHz.

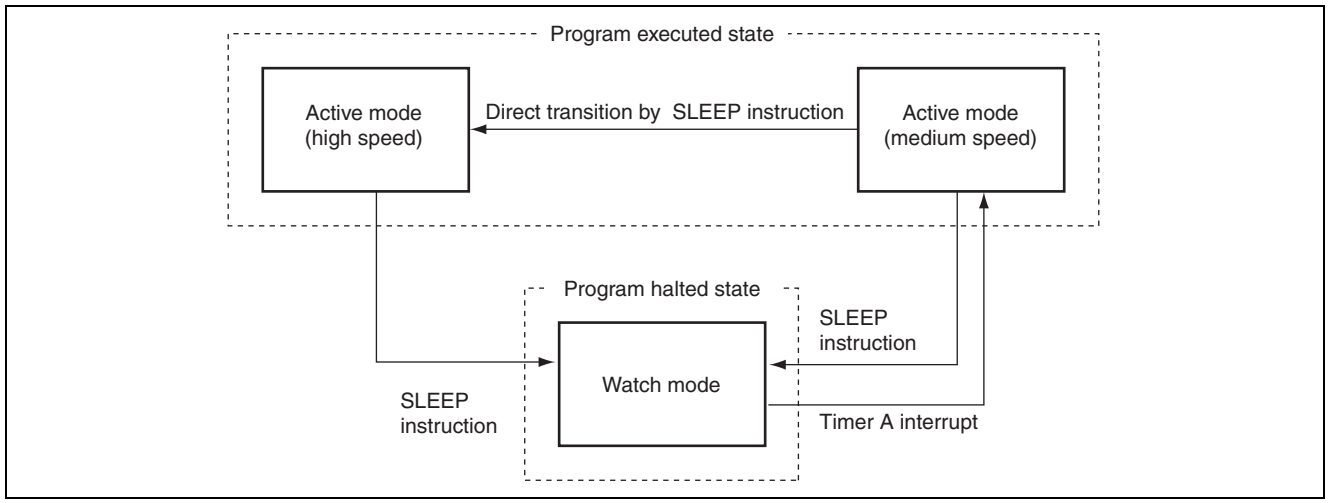


Figure 2.1 Mode Transition to/from Medium-Speed Active Mode

2. Table 2.1 shows the assignment of functions in this sample task. Transition to the medium-speed active mode is performed by assigning the functions as shown in table 2.1.

Table 2.1 Function Assignment

Function	Assignment
SYSCR1	Controls power down modes.
SYSCR2	Controls power down modes.
PDR9	P92 output pin data storage
P92	LED output
TMA	Sets Timer A clock time-base function and TCA overflow period.
TCA	An 8-bit up-counter which overflows every 0.5 sec. by the clock time-base function
IRRTA	Indicates whether or not a Timer A interrupt has been requested.
IENTA	Enables or disables Timer A interrupt requests.

3. Principle of Operation

1. Figure 3.1 illustrates the operation of this sample task. Transition to the medium-speed active mode is made through hardware and software processing as shown in the figure.

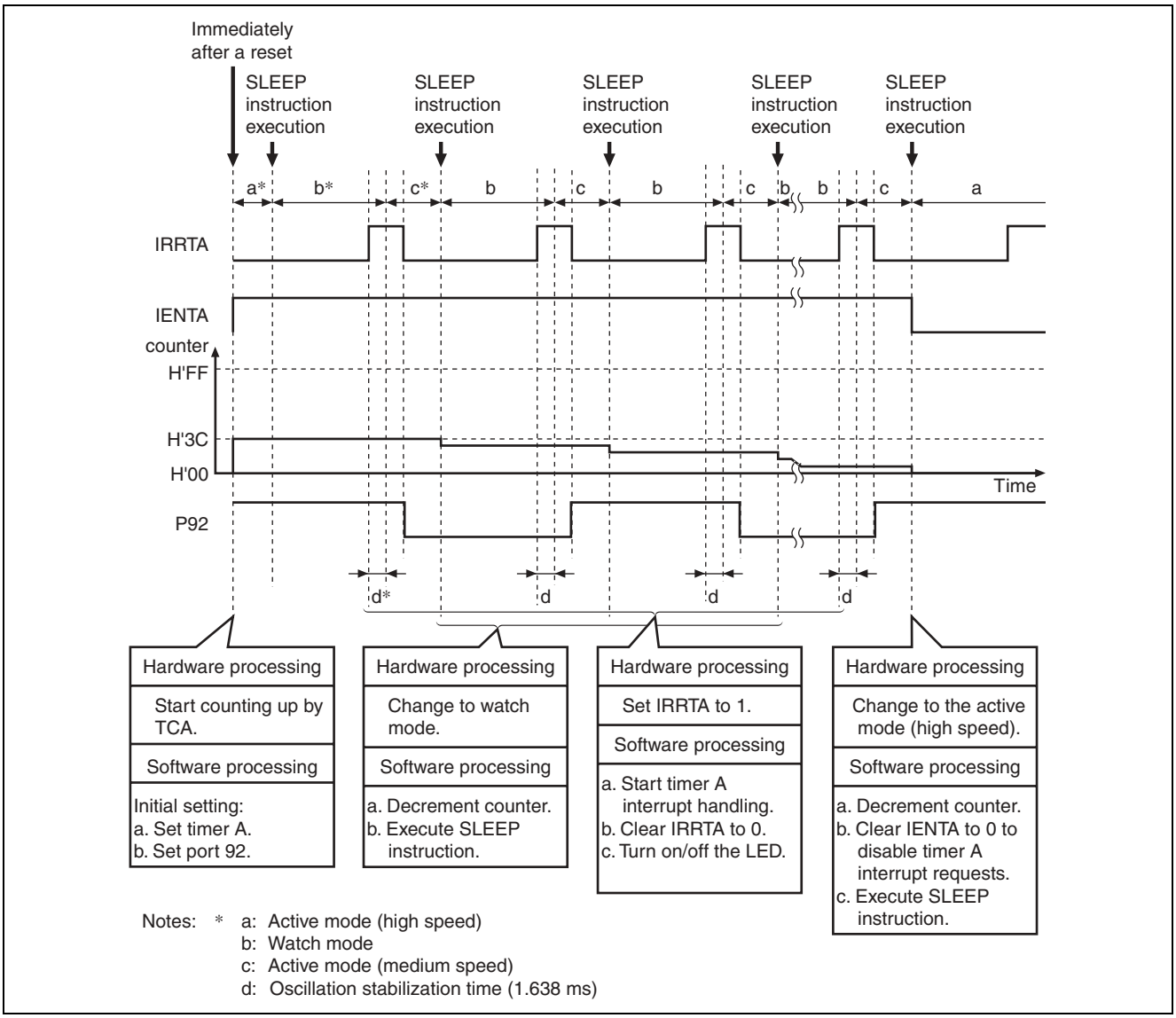


Figure 3.1 Operation Principle of Making Transition to Medium-Speed Active Mode

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 Timer A interrupts and port 9, enables interrupts, and makes transitions to the watch mode, decrements the 8-bit Timer A counter value, and disables Timer A interrupts.
LED control	taint	A Timer A interrupt handling routine which controls the LED.
Direct transition	dtint	A direct transition interrupt handling routine which clears the direct transition interrupt request flag.

4.2 Arguments

This sample task does not use arguments.

4.3 Internal Registers

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

Table 4.2 Description of Internal Registers

Register		Function	Address	Setting
TMA		Timer Mode Register A If TMA = H'19, Timer A function is set to the clock time-based function and TCA overflow period is set to 0.5 sec.	H'FFB0	H'19
TCA		Timer Counter A An 8-bit up counter which overflows every 0.5 sec by the clock time-base function and uses PSW output clock as input.	H'FFB1	H'00
PDR9	P92	Port Data Register 9 (Port Data Register 92) If P92 = 0, the output level on P92 pin is "Low". If P92 = 1, the output level on P92 pin is "High".	H'FFDC Bit 2	1
SYSCR1	SSBY	System Control Register 1 (Software Standby) 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 a SLEEP instruction is executed in the subactive mode. If SSBY = 1, a transition is made to the standby mode or watch 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.	H'FFF0 Bit 7	1
	STS2 STS1 STS0	System Control Register 1 (Standby Timer Select 2, 1, 0) If STS2 = 0, STS1 = 0, and STS0 = 0, oscillation stabilization time after the termination of watch mode is set to 1.638 ms.	H'FFF0 Bit 6 Bit 5 Bit 4	STS2 = 0 STS1 = 0 STS0 = 0

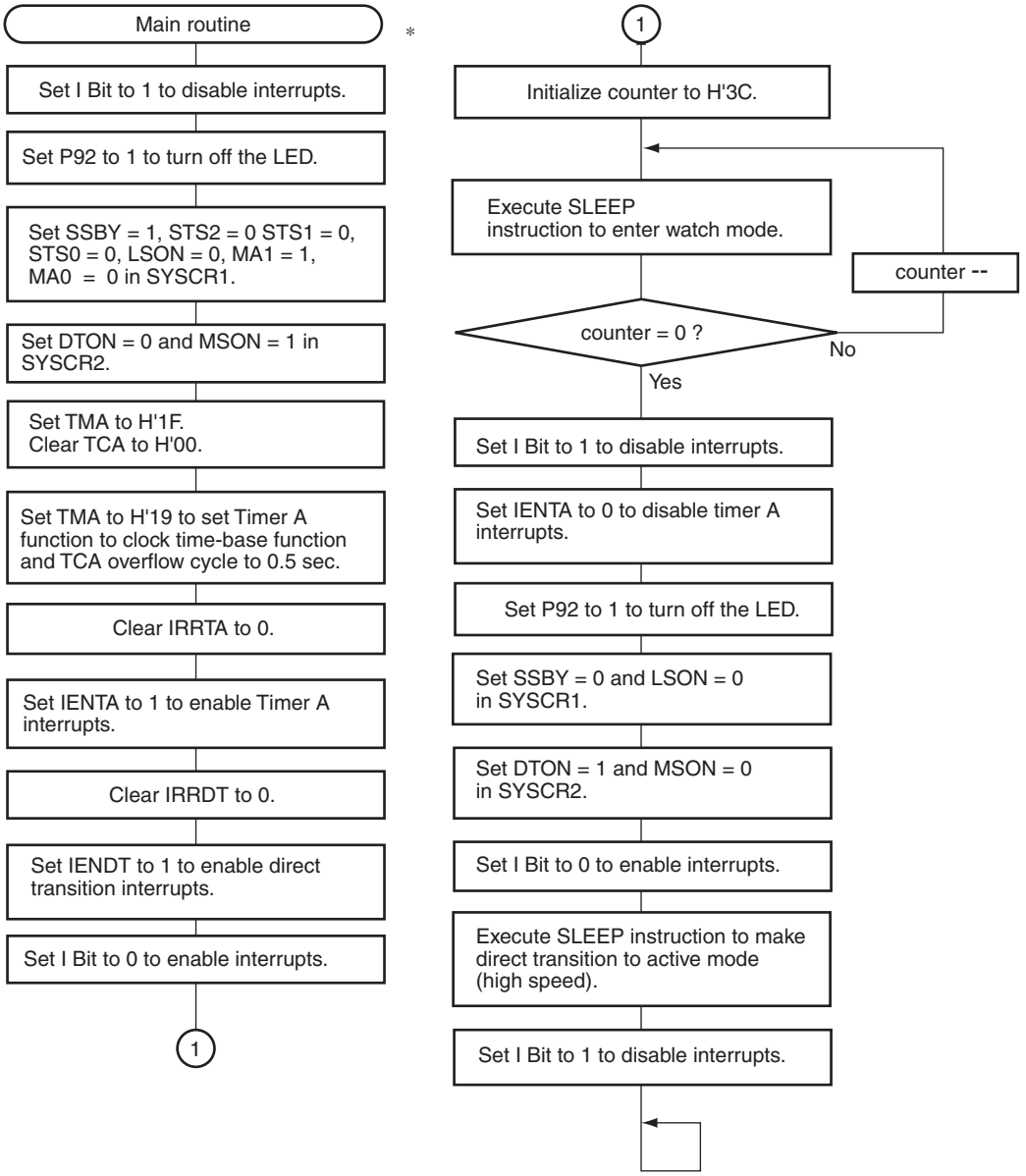
Register		Function	Address	Setting
SYSCR1	LSON	System Control Register 1 (Low Speed ON Flag) If LSON = 0, the CPU operating clock is set to the system clock after the watch mode is terminated. If LSON = 1, the CPU operating clock is set to the subsystem clock after the watch mode is terminated.	H'FFF0 Bit 3	0
	MA1 MA0	System Control Register 1 (Medium-speed active mode clock select 1, 0) If MA1 = 1 and MA0 = 1, the operating clock in the medium-speed active mode is set to ϕ_{OSC}/128.	H'FFF0 Bit 1 Bit 0	MA1 = 1 MA0 = 1
SYSCR2	DTON	System Control Register 2 (Direct Transfer 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. a transition is made to the watch or subsleep mode when a SLEEP instruction is executed in the subactive mode. If DTON = 1, a direct transition is made to the high-speed active mode (if SSBY = 0, MSON = 0 and LSON = 0) or to the subactive mode (if SSBY = 1, TMA3 = 1 and LSON = 1) when a SLEEP instruction is executed in the medium-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 medium-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
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	1
IENR2	IENDT	Interrupt Enable Register 2 (Direct Transition Interrupt Enable) If IENDT = 0, interrupt requests by direct transition are disabled. If IENDT = 1, interrupt requests by direct transition are enabled.	H'FFF4 Bit 7	1
IRR1	IRRTA	Interrupt Request Register 1 (Timer A Interrupt Request Flag) If IRRTA = 0, a Timer A interrupt is not requested. If IRRTA = 1, a Timer A interrupt has been requested.	H'FFF6 Bit 7	0
IRR2	IRRDT	Interrupt Request Register 2 (Direct Transition Interrupt Request Flag) If IRRDT = 0, a direct transition interrupt is not requested. If IRRDT = 1, a direct transition interrupt has been requested.	H'FFF7 Bit 7	0

4.4 Description of RAM

This sample task does not use RAM.

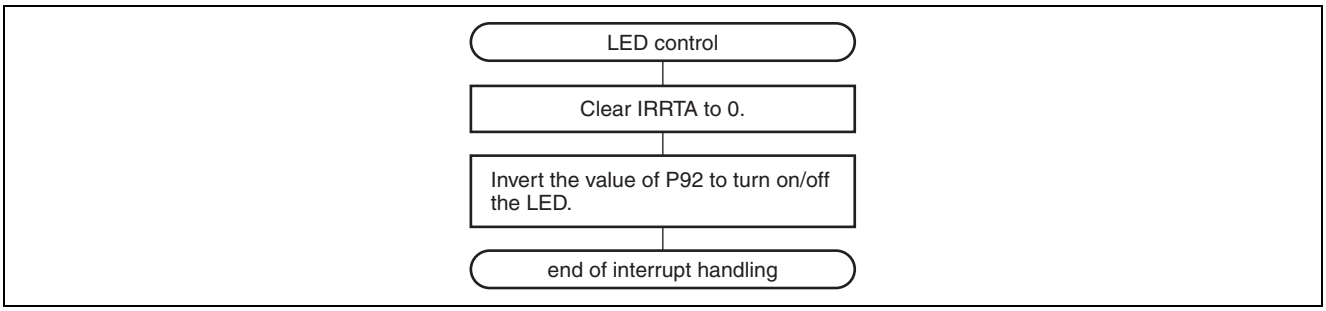
5. Flowchart

1. Main routine

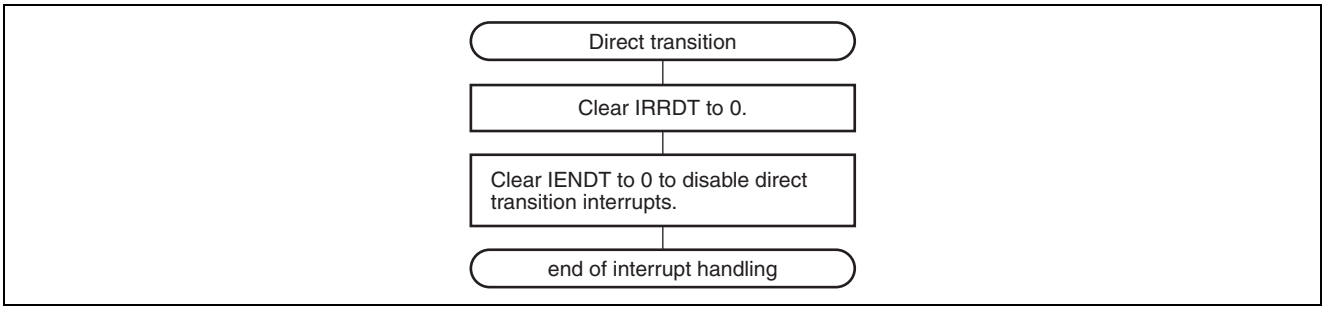


Note: * In this sample task, the stack pointer is set in INIT.SRC (assembly language).

2. Timer A interrupt handling routine



3. Direct transition interrupt handling routine



6. Program Listing

INIT.SRC (Program listing)

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

```
/************************************************************************************************************************/
/*                                                                                                                      */
/*  H8/300L Super Low Power Series                                                                                      */
/*      -H8/38024 Series-                                                                                               */
/*  Application Note                                                                                                    */
/*                                                                                                                      */
/*  'Transition to Subactive Mode'                                                                                      */
/*                                                                                                                      */
/*  Function                                                                                                            */
/*  : Power-Down Mode                                                                                                   */
/*     Subactive 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                             /* Port Mode Register 2 bit0            */
#define     PDR9_BIT    (*(struct BIT *)0xFFDC)                 /* Port Data Register 9                 */
#define     P92         PDR9_BIT.b2                             /* Port Data Register 92                */
#define     PMRB_BIT    (*(struct BIT *)0xFFEE)                 /* Port Mode Register B                 */
#define     IRQ1        PMRB_BIT.b3                             /* Port Mode Register B bit3            */
#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     IEGR_BIT    (*(struct BIT *)0xFFF2)             /* Interrupt Edge Select Register 1         */
#define     IEG1        IEGR_BIT.b1                         /* IRQ1 Edge Select                         */
#define     IEG0        IEGR_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     IEN1        IENR1_BIT.b1                        /* IRQ1 Interrupt Request Enable            */
#define     IEN0        IENR1_BIT.b0                        /* IRQ0 Interrupt Request Enable            */
#define     IENR2_BIT   (*(struct BIT *)0xFFF4)             /* Interrupt Enable Register 1              */
#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     IRRI1       IRR1_BIT.b1                         /* IRQ1 Interrupt Request Flag              */
#define     IRRI0       IRR1_BIT.b0                         /* IRQ0 Interrupt Request Flag              */
#define     IRR2_BIT    (*(struct BIT *)0xFFF7)             /* Interrupt Request Register 1             */
#define     IRRDT       IRR2_BIT.b7                         /* Direct Transfer Interrupt Request Flag   */

#pragma interrupt   (dtint)
#pragma interrupt   (irq0int)
#pragma interrupt   (irq1int)
#pragma interrupt   (taint)
/************************************************************************************************************/
/*  Function define                                                                                         */
/************************************************************************************************************/
extern void INIT ( void );                                  /* SP Set                                   */
void        main ( void );
void        dtint ( void );
void        irq0int ( void );
void        irq1int ( void );
void        taint ( void );

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

#define     USRF_BIT    (*(struct BIT *)&USRF)
#define     SWONF       USRF_BIT.b1                         /* Switch On Flag                           */
#define     LDONF       USRF_BIT.b0                         /* LED On Flag                              */
```

```
/*****************************************************************************************************************/
/*  Vector Address                                                                                               */
/*****************************************************************************************************************/
#pragma section     V1                                          /* Vector Section Set                        */
void (*const VEC_TBL1[])(void) = {
    INIT                                                        /* 0x0000 Reset Vector                       */
};
#pragma section     V2                                          /* Vector Section Set                        */
void (*const VEC_TBL2[])(void) = {
    irq0int                                                     /* 0x0008 IRQ0 Interrupt Vector              */
};
#pragma section     V3                                          /* Vector Section Set                        */
void (*const VEC_TBL3[])(void) = {
    irq1int                                                     /* 0x000A IRQ1 Interrupt Vector              */
};
#pragma section     V4                                          /* Vector Section Set                        */
void (*const VEC_TBL4[])(void) = {
    taint                                                       /* 0x0016 timer A Interrupt Vector           */
};
#pragma section     V5                                          /* Vector Section Set                        */
void (*const VEC_TBL5[])(void) = {
    dtint                                                       /* 0x0028 Sleep Interrupt Vector             */
};

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

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

    P92 = 1;                                                    /* Initialize P92                            */

    IRQ1 = 1;                                                   /* Initialize IRQ1 Terminal Input            */
    IRQ0 = 1;                                                   /* Initialize IRQ0 Terminal Input            */

    TMA = 0x1F;                                                 /* Reset PSW & TCA                           */
    TMA = 0x19;                                                 /* Set TMA3                                  */
    SYSCR1 = 0x8F;                                              /* Set SYSCR1                                */
    SYSCR2 = 0xE0;                                              /* Set SYSCR2                                */

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

    IEG1 = 0;                                                   /* Initialize IRQ1 Terminal Input Edge       */
    IRRI1 = 0;                                                  /* Clear IRRI1                               */
    IEN1 = 0;                                                   /* IRQ1 Interrupt Disable                    */

    IRRTA = 0;                                                  /* Clear IRRTA                               */
    IENTA = 0;                                                  /* Timer A Interrupt Disable                 */

    IRRDT = 0;                                                  /* Clear IRRDT                               */
    IENDT = 1;                                                  /* Direct Transfer Interrupt Enable          */
```

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

    TMA = 0x1F;                                         /* Reset PSW & TCA                  */
    TMA = 0x19;                                         /* Initialize Timer A Function      */

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

    IRRI1 = 0;                                          /* Clear IRRI1                      */
    IEN1 = 1;                                           /* IRQ1 Interrupt Enable            */

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

    P92 = 1;                                            /* Turn off LED                     */

    SYSCR1 = 0x87;                                      /* Set SYSCR1                       */
    SYSCR2 = 0xE8;                                      /* Set SYSCR2                       */

    set_imask_ccr(0);                                   /* Interrupt Enable                 */
    sleep();                                            /* Transition to Active Mode        */
    set_imask_ccr(1);                                   /* Interrupt Disable                */

    while(1){
        ;
    }
}

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

/*****************************************************************************************************************/
/*  IRQ1 Interrupt                                                                                               */
/*****************************************************************************************************************/
void    irq1int ( void )
{
    IRRI1 = 0;                                          /* Clear IRRI1                      */
    SWONF = 1;                                          /* Set SWONF                        */
    IENTA = 0;                                          /* Timer A Interrupt Disable        */
    IEN1 = 0;                                           /* IEN1 Interrupt Disable           */
}

/*****************************************************************************************************************/
/*  Timer A Interrupt                                                                                            */
/*****************************************************************************************************************/
void    taint ( void )
{
    IRRTA = 0;                                          /* Clear IRRTA                      */

    set_imask_ccr(0);                                   /* Interrupt Enable                 */
```

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

/*****************************************************************************************************************/
/*  Direct Transfer Interrupt                                                                                    */
/*****************************************************************************************************************/
void    dtint ( void )
{
    IRRDT = 0;                                          /* Clear IRRDT                      */
    IENDT = 0;                                          /* Direct Transfer Interrupt Enable */
}
```

Link address specifications

Section Name	Address
CV1	H'0000
CV2	H'0016
CV3	H'0028
P	H'0100

Revision Record

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

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.