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

Interrupt-Period Setup Using Auto-Reload Timer Function

Introduction

Generate a Timer B1 interrupt every 160 µs using the auto reload timer function of Timer B1.

Target Device

H8/300H Tiny Series H8/3687

Contents

1. Specification

1. Generate a Timer B1 interrupt every 160 µs using the auto reload timer function of Timer B1.
2. Timer B1 interrupt events are counted during the interrupt sequence, and each time a count of 250 is reached the LED is either illuminated or extinguished.

2. Description of Functions Used

1. In this task example, a Timer B1 interrupt can be generated every 160 µs by using the auto reload timer function of Timer B1.
   A. Figure 1 shows the block diagram of the Timer B1 auto reload function which is described as follows:
      - The system clock (ϕ) is a 10 MHz and is used as the reference clock that drives the CPU and peripheral functions.
      - Prescaler S (PSS) is a 13-bit counter to which the system clock ϕ is applied and is incremented by one on each clock cycle.
      - Timer Mode Register B1 (TMB1) is an 8-bit read/write register used to select the auto reload function and select the input clock.
      - Timer Counter B1 (TCB1) is an 8-bit readable up counter that is incremented by means of the applied internal clock and/or external events. The applied input clock can be selected from a total of eight types of clock derived from the system clock divided by 8192, 2048, 512, 256, 64, 16 and 4, and by an external clock. In this task example, a clock, derived by dividing the system clock by 64, is selected as the input clock applied to TCB1.
      - Timer Control Register B1 (TLB1) is an 8-bit write-only register used to set the reload value for TCB1. In this task example, setting TLB1 to H'E7 ensures that TCB1 will overflow in 160 µs.
      - Timer B1 Interrupt Request Flag (IRRTB1) is set to 1 by a TCB1 overflow event. If, provided that IRRTB1 has been set to 1, Timer B1 Interrupt Enable (IENTB1) of the Interrupt Enable Register (IENR1) is set to 1, and the I bit of the Condition Code register (CCR) is cleared to 0, the Timer B1 interrupt sequence will start on reception of the Timer B1 interrupt.

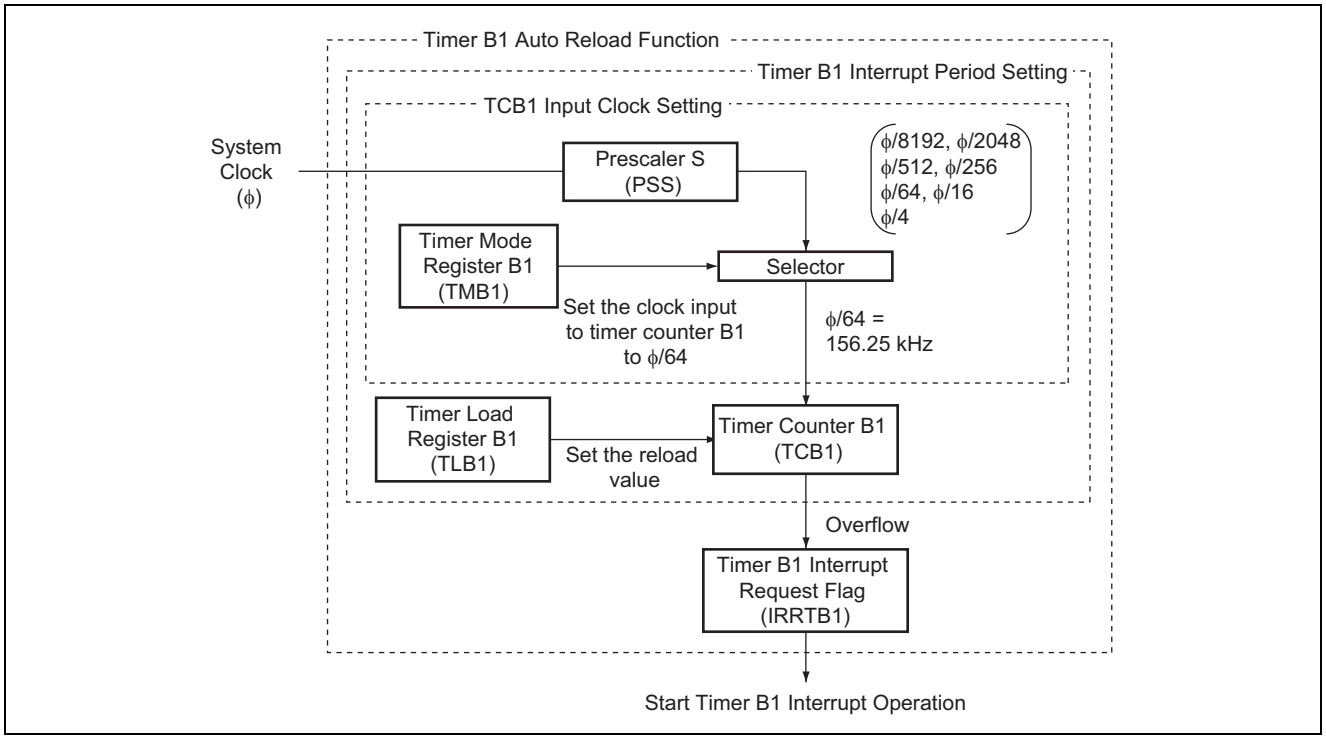


Figure 1 Timer B1 Auto Reload Function Block Diagram

B. The following is brief description of how to set the interrupt period using the Timer B1 auto reload function.
   - The interrupt period using the Timer B1 auto-reload function is set by the following expression:

     Timer B1 Interrupt Period (s) = (TCB1 input clock period (s)) × (256 – (reload setting value))

   - The Timer B1 interrupt period is set by respectively setting the TCB1 input clock period, as defined by the aforementioned expression, in TMB1 and the reload setting value in TLB1.

2. Table 1 lists the function assignments applicable to this task example. The interrupt period is set using the Timer B1 auto reload function.

Table 1 Function Assignment

Function	Function Assignment
PSS	This is a 13-bit counter to which the system clock is applied
TCB1	This is an 8-bit counter to which the system clock divided by 64 is input
TMB1	This register sets the clock input to timer counter B1 to ϕ/64
TLB1	This sets the reload value for TCB1
IRRTB1	This flag reflects the presence of the Timer B1 interrupt request
PCR7	This sets the P73 output pin function
PDR7	This stores the P73 output pin data
P73	This is the LED output

3. Operational Description

Figure 2 shows the principle of operation by way of waveform diagrams. As shown in figure 2, setting the interrupt period by means of the Timer B1 auto reload function is facilitated by both hardware and software.

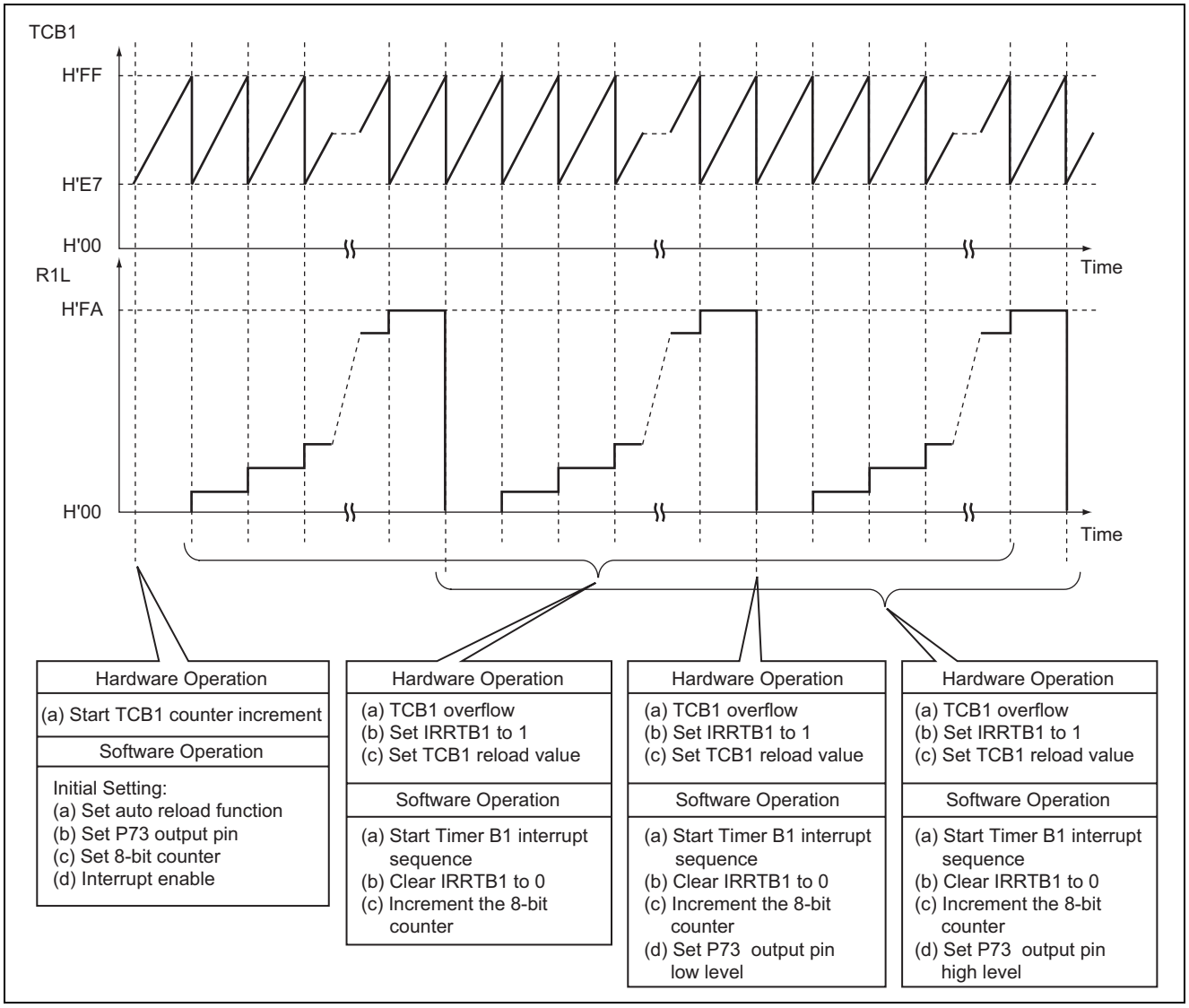


Figure 2 Operating Principle of Interrupt Period Setup by the Timer B1 Auto Reload Function

4. Software Description

4.1 Module Description

The modules applicable to this task example are listed in table 2.

Table 2 Module Description

Module Name	Label Name	Function
Main Routine	main	Initializes the stack pointer, sets the auto-reload function, facilitates port 73 setup, sets the 8-bit counter, enables the interrupts, and controls on/off illumination of the LED.
Count Up	TB1INT	Increments the 8-bit counter and sets CTEDF to 1 when the 8-bit counter counts up to H'FA.

4.2 Argument Description

There are no arguments used in this task example.

4.3 Description of Applicable Internal Registers

Table 3 lists the internal registers used in this task example.

Table 3 Description of Applicable Internal Registers

Register Name		Functional Description	Address	Setting
IRR2	IRRTB1	Interrupt Request Register 1 (Timer B1 Interrupt Request Flag): When IRRTB1 is 0, a Timer B1 interrupt is not requested When IRRTB1 is 1, a Timer B1 interrupt is requested	H'FFF7 Bit 5	0
IENR2	IENTB1	Interrupt Enable Register 1 (Timer B1 Interrupt Enable): When IENTB1 is 1, the Timer B1 interrupt is enabled	H'FFF5 Bit 5	1
TMB1		Timer Mode Register B1: When TMB1 is H'FC, the Timer B1 function is set as the auto reload function and the clock, derived from the system clock divided by 64, is set as the TCB1 input clock	H'FFB2	H'FC
TCB1		Timer Counter B1: This is an 8-bit up counter to which the input clock, derived from the system clock divided by 64, is applied	H'FFB3	H'00
TLB1		Timer Load Register B1: When TLB1 is H'E7, TCB1 starts to count from H'E7, and when the register overflows, TCB1 is reloaded with the value H'E7	H'FFB3	H'E7
PDR7	P73	Port Data Register 7 (Port Data Register 73): When P73 is set to 0, the output level of pin P73 is low When P73 is set to 1, the output level of pin P73 is high	H'FFDA Bit 3	0
PCR7	PCR73	Port Control Register 7 (Port control Register 73): When PCR73 is set to 1, pin P73 is set as an output pin.	H'FFEA Bit 3	1

4.4 Description of RAM Used

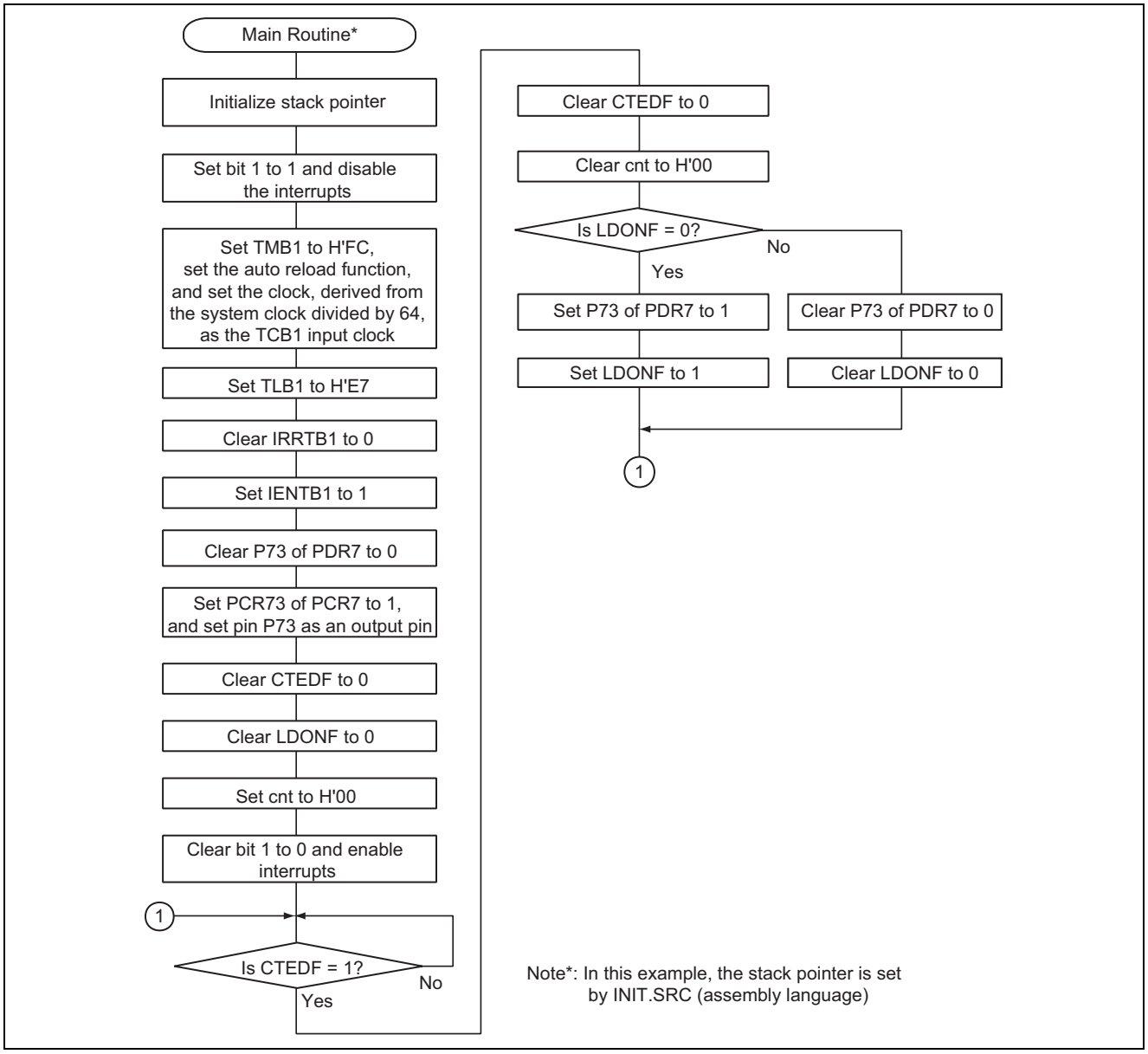
Table 4 lists and describes the RAM used in this task example.

Table 4 Description of Applicable RAM

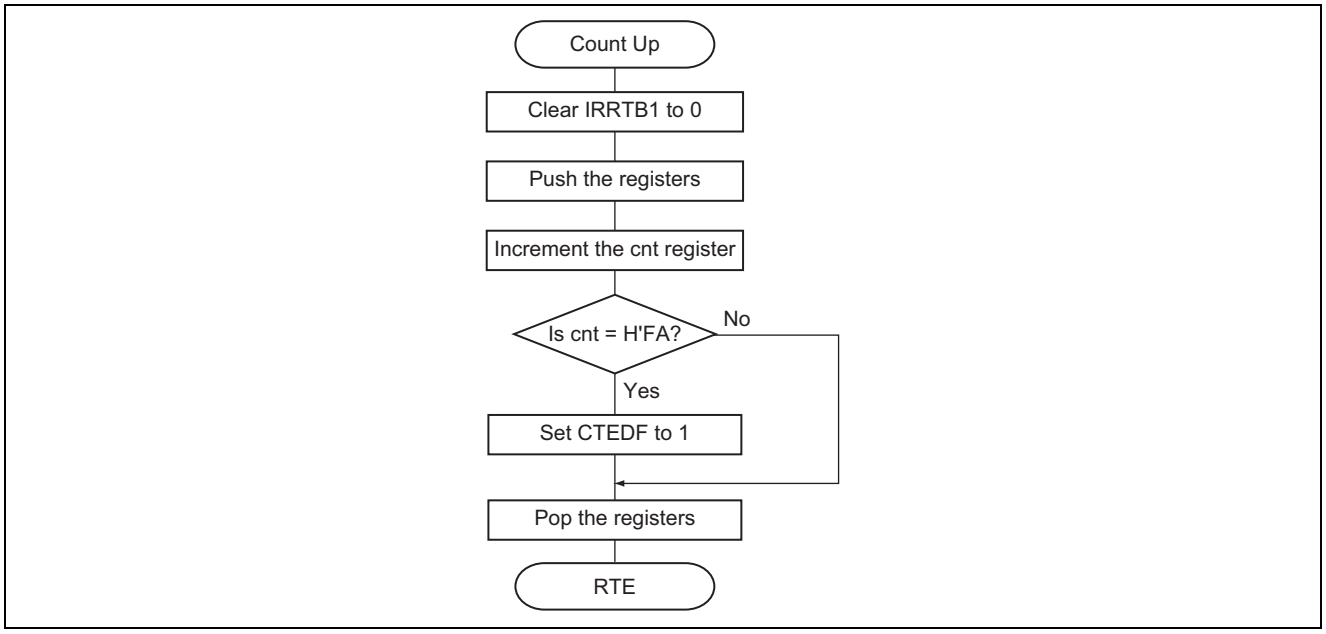
Label Name		Function	Address	Used in
USRF	CTEDF	Flag that determines whether the count value in the 8-bit counter is H'FA	H'FB80 Bit 0	Main routine Count up
	LDONF	Flag that determines the on/off status of the LED	H'FB80 Bit 1	Main routine
cnt		An 8-bit counter	H'FB81	Count up

5. Flowcharts

1. Main Routine



2. Timer B1 Interrupt Service Routine



5.1 Link Address Designation

Section Name	Address
CV1	H'0000
P	H'0100
V	H'FB80

6. Program Listing

INIT.SRC (Program List)

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

```
/************************************************************/
/*                                                          */
/*  H8/300HN Series -H8/3687-                               */
/*  Application Note                                        */
/*                                                          */
/*'Interrupt Period Setting by Auto Reload Timer Function'*/
/*                                                          */
/*  Function                                                */
/*  :Timer B1 Auto Reload                                   */
/*                                                          */
/*  External Clock : 10MHz                                  */
/*  Internal Clock : 10MHz                                  */
/*  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  TMB  *(volatile unsigned char *)0xF760          /* Timer Mode Register B1         */
#define  TCB1 *(volatile unsigned char *)0xF761          /* Timer Counter B1               */
#define  TLB1 *(volatile unsigned char *)0xF761          /* Timer Load Register            */

#define  IEGR1_BIT    (*(struct BIT *)0xFFF2)            /*Interrupt Edge Select Register 1*/
#define  IEG3 IEGR1_BIT.b3                               /* IRQ3 Edge Select               */
#define  IEG2 IEGR1_BIT.b2                               /* IRQ2 Edge Select               */
#define  IEG1 IEGR1_BIT.b1                               /* IRQ1 Edge Select               */
#define  IEG  IEGR1_BIT.b0                               /* IRQ0 Edge Select               */
#define  IENR1    *(volatile unsigned char *)0xFFF4      /* Interrupt Enable Register 1    */
#define  IENR1_BIT    (*(struct BIT *)0xFFF4             /* Interrupt Enable Register 1    */
#define  IEN3 IENR1_BIT.b3                               /* IRQ3 Interrupt Enable          */
#define  IEN2 IENR1_BIT.b2                               /* IRQ2 Interrupt Enable          */
#define  IEN1 IENR1_BIT.b1                               /* IRQ1 Interrupt Enable          */
#define  IEN0 IENR1_BIT.b0                               /* IRQ0 Interrupt Enable          */
#define  IENR2    *(volatile unsigned char *)0xFFF5      /* Interrupt Enable Register 1    */
#define  IENR2_BIT    (*(struct BIT *)0xFFF5)            /* Interrupt Enable Register 1    */
#define  IENTB1   IENR2_BIT.b5                           /* IRQ3 Interrupt Enable          */
#define  IRR1 *(volatile unsigned char *)0xFFF6          /* Interrupt Flag Register 1      */
#define  IRR1_BIT (*(struct BIT *)0xFFF6)                /* Interrupt Flag Register 1      */
#define  IRRI3    IRR1_BIT.b3                            /* IRQ3 Interrupt Request Flag    */
#define  IRRI2    IRR1_BIT.b2                            /* IRQ2 Interrupt Request Flag    */
#define  IRRI1    IRR1_BIT.b1                            /* IRQ1 Interrupt Request Flag    */
#define  IRRI0    IRR1_BIT.b0                            /* IRQ0 Interrupt Request Flag    */
#define  IRR2 *(volatile unsigned char *)0xFFF7          /* Interrupt Flag Register 2      */
#define  IRR2_BIT (*(struct BIT *)0xFFF7)                /* Interrupt Flag Register 2      */
#define  IRRTB1   IRR2_BIT.b5                            /* TMB1 Interrupt Request Flag    */
#define  PDR7 *(volatile unsigned char *)0xFFDA          /* Port7 Data Register            */
#define  PDR7_BIT (*(struct BIT *)0xFFDA)                /* Port7 Data Register            */
#define  P7B3 PDR7_BIT.b3                                /* Port Data Register 73          */
#define  PCR7_BIT (*(struct BIT *)0xFFDA)                /* Port7 Control Register         */
#define  PCR73    PDR7_BIT.b3                            /* Port73 Control Register        */

#pragma  interrupt    (TB1INT)
```

```
/*********************************************************/
/*  Function Definitions                                 */
/*********************************************************/
extern   void INIT ( void );                              /* SP Set                       */
void     main     ( void );
void     TB1INT   ( void );

/*********************************************************/
/*  RAM define                                           */
/*********************************************************/
unsigned char    USRF;                                    /* User Flag Erea               */
unsigned char    USRF_1;                                  /* User Flag Erea               */
unsigned char    cnt;                                     /* Counter                      */

extern void _INITSCT();

/*********************************************************/
/*  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) = {
    TB1INT                                                /* 29 Timer B1 Interrupt        */
};
#pragma  section                                          /* P                            */
```

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

    set_imask_ccr(1);                  /* Interrupt Disable                          */

    TMB1 = 0xFC;                       /* Initialize Timer B1 Function & Input Clock */
    TLB1 = 0xE7;                       /* Initialize TCB1 reload Value               */
    IRRTB1 = 0;                        /* Interrupt Flag Initialize                  */
    IENTB1 = 1;                        /* Interrupt Enable Flag Set                  */

    P7B3 = 0;
    PCR73 = 1;                         /* Initialize P73 Output Terminal Function    */

    USRF = 0x00;                       /* Initialize User Flag Erea                  */
    USRF_1 = 0x00;                     /* Initialize User Flag Erea                  */
    cnt = 0x00;                        /* Initialize 8 bit counter                   */

    set_imask_ccr(0);                  /* Interrupt Enable                           */

    while(1)
    {

        while(USRF == 0x01)            /* CTEDF = "1"?                               */
        {
              USRF = 0x00;             /* Clear CTEDF                                */

              cnt = 0x00;              /* Initialize 8 bit counter                   */

        if (USRF_1 == 0x00)            /* LDONF = "1"?                               */
        {
               P7B3 = 1;               /* Turn on LED                                */
               USRF_1 = 0x01;          /* Set LDONF                                  */
           }
          else
           {
               P7B3 = 0;               /* Turn off LED                               */
               USRF_1 = 0x00;          /* Clear LDONF                                */
           }
       }
    }
}
```

```
/*********************************************************/
/*  TimerB1 Interrupt                                    */
/*********************************************************/
void TB1INT( void )
{
    IRRTB1 = 0;                         /* Clear IRRTB1                              */
    cnt = cnt+1;                        /* Increment 8 bit Counter                   */

    if(cnt == 0xFA)                     /* 8 bit Counter = H'FA                      */
     {
      USRF = 0x01;                      /* Set CTEDF                                 */
     }
}
```

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Feb.26.03	—	First edition issued
2.00	Jul.22.05	—	Second 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.