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

Using an Infrared Transceiver to Transmit and Receive Character Data

Introduction

Character reception/transmission via a physical layer is performed with a microcomputer to which an infrared light transceiver is connected using an IrDA communication port in a personal computer.

Target Device

H8/38024

Contents

1. Specifications 2
2. Description of Functions 4
3. Principles of Operation 5
4. Description of Software 6
5. Flowchart 9
6. Program Listing 14

1. Specifications

1. Figure 1.1 shows a hardware structure for data reception/transmission using an infrared light transceiver.
2. In this sample task, a character transmitted from an IrDA port in a personal computer is received by the infrared light transceiver, ASCII code of the received character is incremented, and the character is returned to the personal computer via the infrared light transceiver. On the monitor of the personal computer, a character corresponding to the incremented ASCII code is displayed.
3. In this sample task, the operating voltage (Vcc) and the analog power supply voltage (AVcc) of the H8/38024 are 3.3 V, the OSC clock frequency is 10 MHz, and the watch clock frequency is 32.768 kHz.

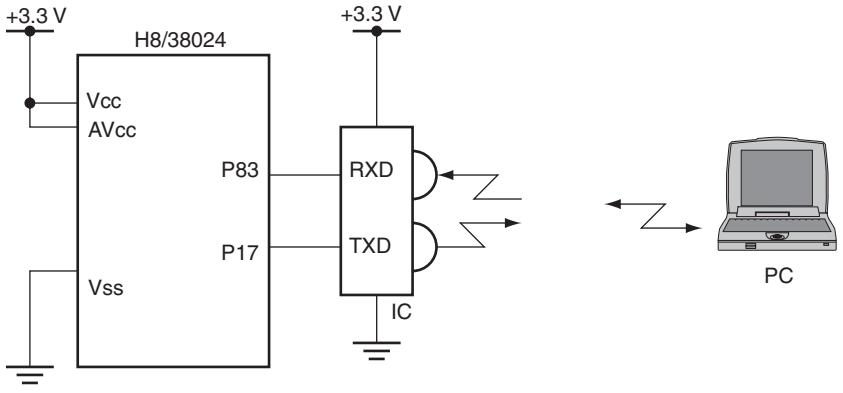


Figure 1.1 Hardware Structure

4. The infrared light transceiver used in this sample task is manufactured by ROHM (type name: RPM851A).

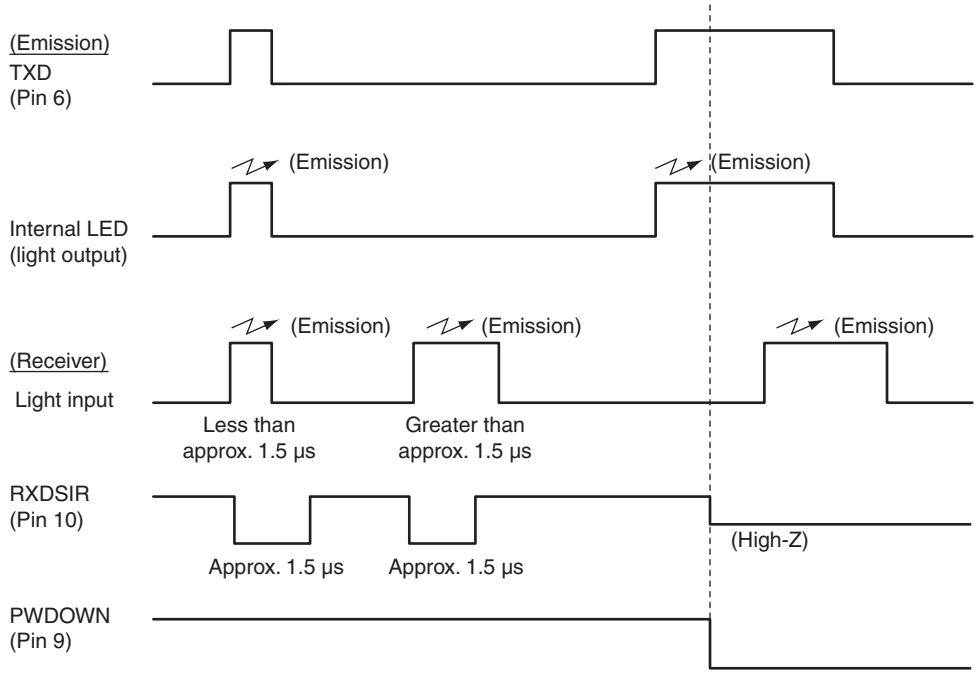


Figure 1.2 Example of Infrared Light Transceiver Operation Timing Chart

A. Features of the RPM851A are as follows.
   a. Conforms to IrDA Ver. 1.0.
   b. Designed for saving current during standby mode (typ. 220 µA)
   c. Most appropriate for using with a battery through a power down control function
   d. Power supply voltage range: 2.7 V to 5.5 V
   e. Package allowing surface mounting on both top and side surfaces

B. Figure 1.2 shows an example of infrared light transceiver operation timing chart.
C. Figure 1.3 shows the block diagram of the infrared light transceiver and an example of the application circuit.

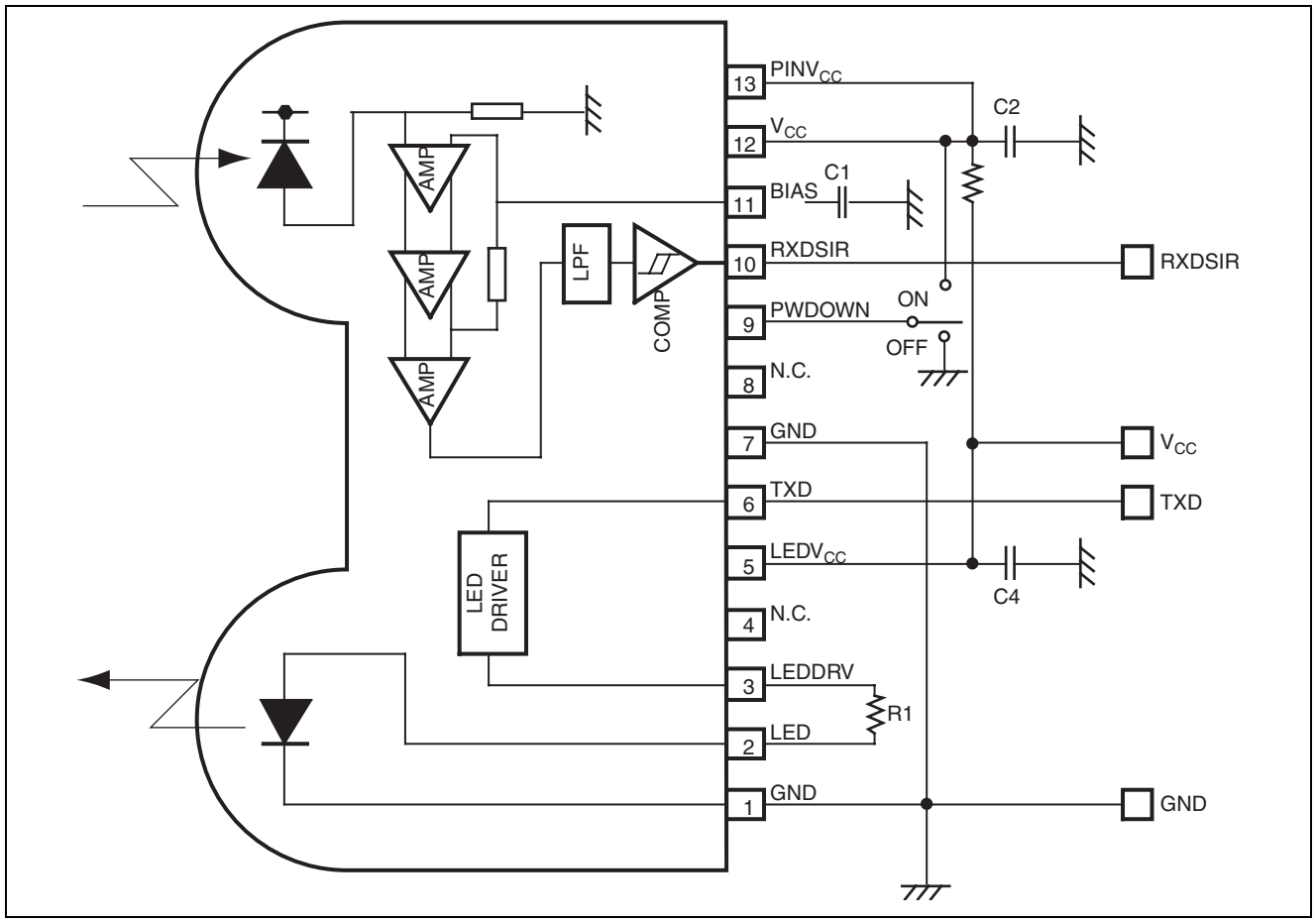


Figure 1.3 Block Diagram of Infrared Light Transceiver and Example of Application Circuit

5. This sample task operates as follows.
   A. Communication via a physical layer is performed by a terminal software in a personal computer using a waveform of IrDA 1.0.
   B. For example, a character '1' corresponding to ASCII code 'H31' is input from the keyboard of the personal computer.
   C. A modulated signal is transmitted from the IrDA port of the personal computer in LSB first.
   D. The infrared light transceiver connected to a microcomputer receives the signal and captures data 'H31'.
   E. The microcomputer increments the data to 'H32', modulates it, and immediately returns it via the infrared light transceiver.
   F. The signal received by the IrDA port of the personal computer is demodulated, and 'H32' is captured. '2' is displayed on the monitor since the ASCII code 'H32' corresponds to the character '2'.

2. Description of Functions

1. Figure 2.1 shows a block diagram of functions of the H8/38024 in this sample task; table 2.1 shows function allocations.

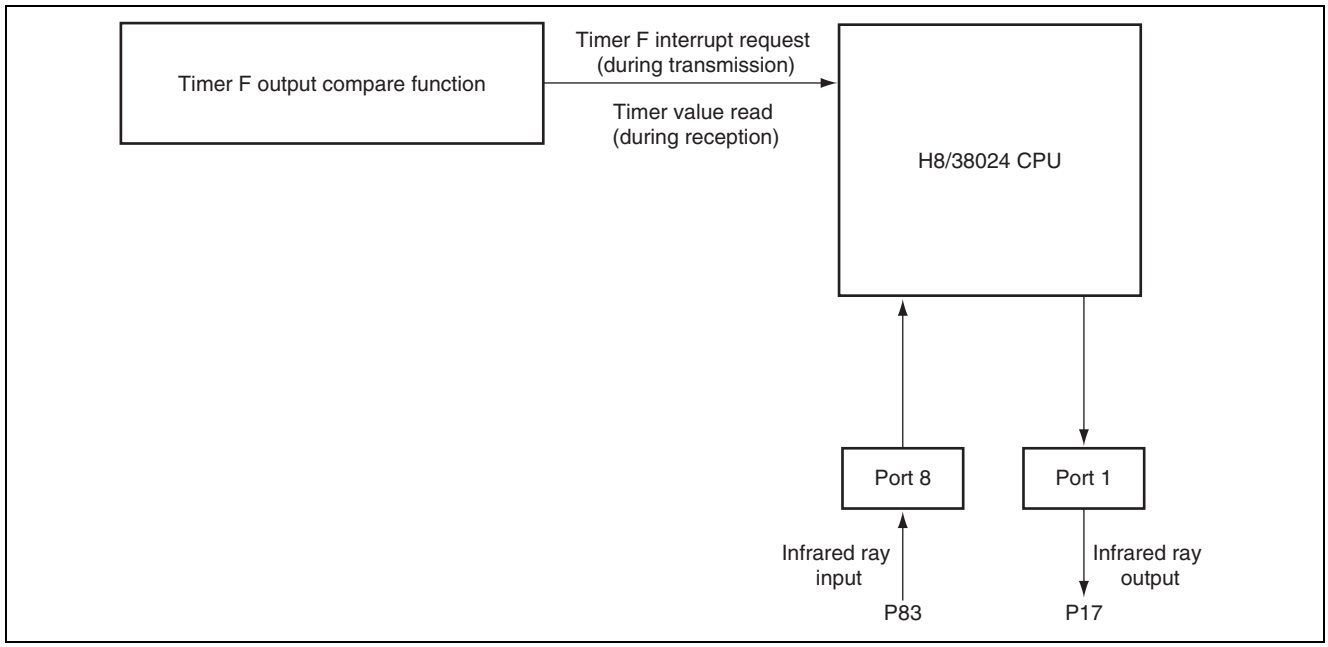


Figure 2.1 Block Diagram of Functions

Table 2.1 Function Allocation

Function	Function Allocation
Timer F	Outputs toggle signal using a compare-match function. Changes the output frequency by setting a value to the output compare register (OCRFL).
Port 1	Transmits infrared ray data from P17 output pin of port 1
Port 7	Receives infrared ray data from P83 input pin of port 8

3. Principles of Operation

1. Figure 3.1 shows the principle of operation when infrared ray communication is performed using the timer F. Transmission is performed through a timer F compare-match interrupt, and reception is performed by using a TCFL value without using an interrupt.

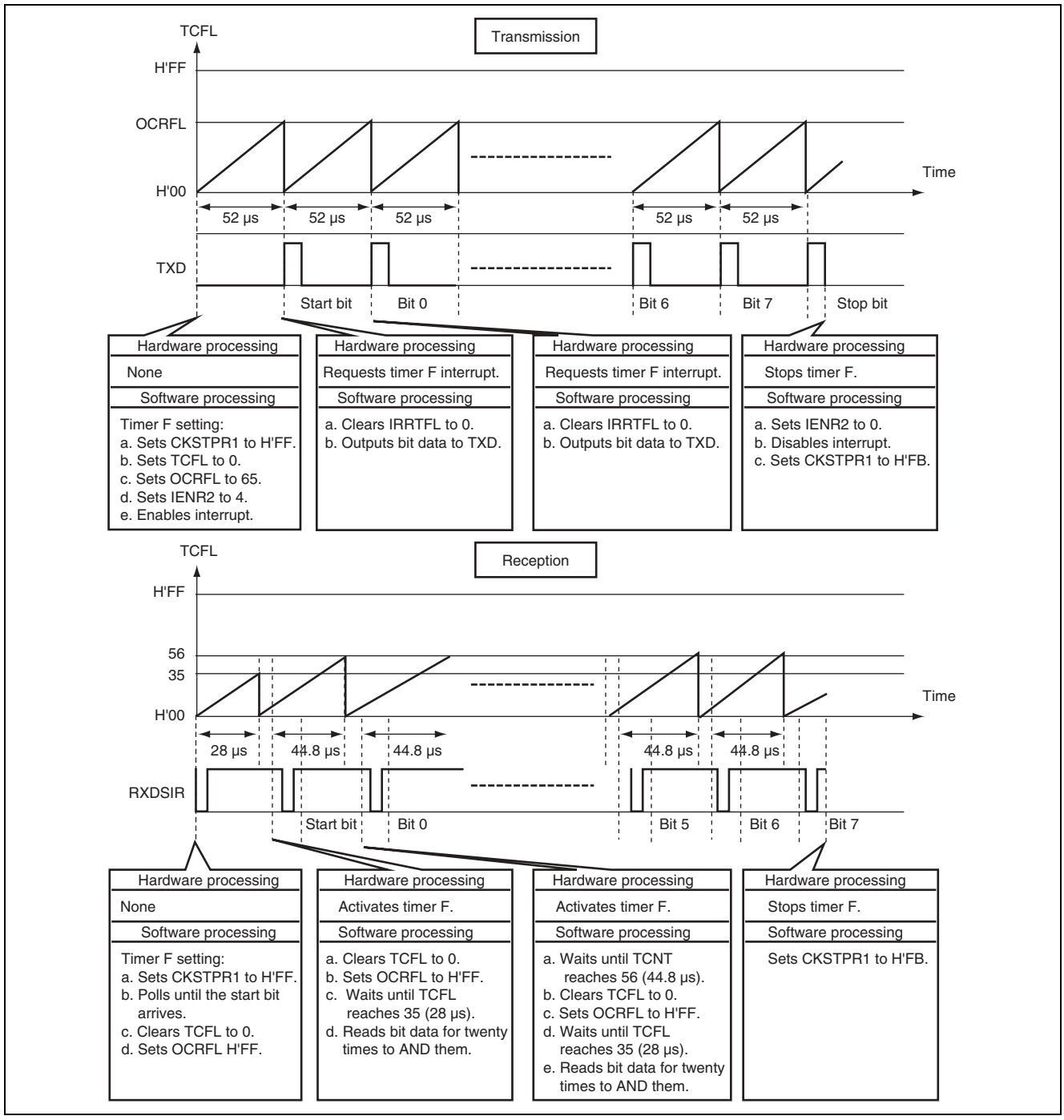


Figure 3.1 Principle of Operation of Infrared Ray Communication Using Timer F

4. Description of Software

4.1 Modules

Table 4.1 describes the modules used in this sample task.

Table 4.1 Description of Modules

Module	Label	Function
Main routine	main	Initializes, and calls infrared ray data reception processing routine and infrared ray data transmission processing routine alternately
Infrared ray data reception processing routine	irda_rcv	Receives data through an infrared ray
Infrared ray data transmission processing routine	irda_snd	Transmits data through an infrared ray
Timer F interrupt processing routine	tmrw	This is used as a 52-µs timer

4.2 Arguments

This sample task does not use arguments.

4.3 Internal Registers

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

Table 4.2 Description of Internal Registers

Register		Function	Address	Setting
TCRF		Timer control register F Switches over 16-bit/8-bit mode, selects from four internal clocks and external events, and sets the TMOFH/TMOFL pin output level.	H'FFB6	H'66 (initial setting)
	TOLH	Toggle output level H Sets the TMOFH pin output level. When TOLH = 0: Low.	Bit 7	0
	CKSH2	Clock select H When CKSH2 = 1, CKSH1 = 1, and CKSH0 = 0, counts with the internal clock ϕ/4.	Bit 6	1
	CKSH1		Bit 5	1
	CKSH0		Bit 4	0
	TOLL	Toggle output level L Sets the TMOFL pin output level. When TOLL = 0: Low.	Bit 3	0
	CKSL2	Clock select L When CKSL2 = 1, CKSL1 = 1, and CKSL0 = 0, counts with the internal clock ϕ/4.	Bit 2	1
	CKSL1		Bit 1	1
	CKSL0		Bit 0	0

Register		Function	Address	Setting
TCSRF		Timer control/status register F Selects counter clear, sets overflow flag, sets compare-match flag, enables/disables overflow interrupt request.	H'FFB7	H'01
	OVFH	Timer overflow flag H Status flag which indicates that the TCFH overflow (H'FF → H'00) has occurred.	Bit 7	0
	CMFH	Compare-match flag H Status flag which indicates that TCFH and OCRFH match.	Bit 6	0
	OVIEH	Timer overflow interrupt enable H When OVIEH = 0, disables TCFH overflow interrupt requests.	Bit 5	0
	CCLRH	Counter clear H When CCLRH = 1, enables TCF clear when compare-match occurs.	Bit 4	0
	OVFL	Timer overflow flag L Status flag which indicates that TCFL overflow (H'FF → H'00) has occurred.	Bit 3	0
	CMFL	Compare-match flag L Status flag which indicates that TCFL and OCRFL match.	Bit 2	0
	OVIEL	Timer overflow interrupt enable L When OVIEL = 0, disables TCFL overflow interrupt requests.	Bit 1	0
	CCLRL	Counter clear L When CCLRL = 1, enables TCFL clear when compare-match occurs.	Bit 0	0
TCFL		8-bit timer counter An 8-bit readable/writable up-counter	H'FFB9	H'00
OCRFL		16-bit output compare register Generates interrupts when a value matches with TCFL	H'FFBB	255/65
CKSTPR1		Clock stop register 1 Controls module standby mode. When TFCKSTP = 0 (CKSTPR1 = H'FB): sets timer F to module standby mode. When TFCKSTP = 1 (CKSTPR1 = H'FF): cancels timer F module standby mode.	H'FFFA	H'FB/H'FF
IENR2		Interrupt enable register 2	H'FFF4	H'04/H'00
	IENTFL	When IENTFL = 0, enables timer FL interrupt requests	Bit 2	1/0
IRR2		Interrupt request register 2	H'FFF7	H'00
	IRRTFL	This register can be cleared when 0 is written to IRRTFL while IRRTFL = 1. This register is set to 1 when TCFL and OCRFL match in 8-bit timer mode.	Bit 2	1/0
PCR1		Port control register 1 Selects, for each bit, the pin I/O to use as the port 1 general-purpose I/O port. When PCR1 = H'80, pin P17 functions as a general-purpose output pin, and other pins function as general-purpose input pins.	H'FFE4	H'80

Register	Function	Address	Setting
PDR1	Port data register 1 General-purpose I/O port data register for port 1.	H'FFD4	H'00
PCR8	Port control register 8 Selects, for each bit, the pin I/O used as a port 8 general-purpose I/O port. When PCR8 = H'F7, the pin P83 functions as a general-purpose input pin, and other pins function as general-purpose output pins.	H'FFEB	H'F7
PDR8	Port data register 8 Port 8 general-purpose I/O port data register.	H'FFDB	H'00 (initial setting)

4.4 Description of RAM

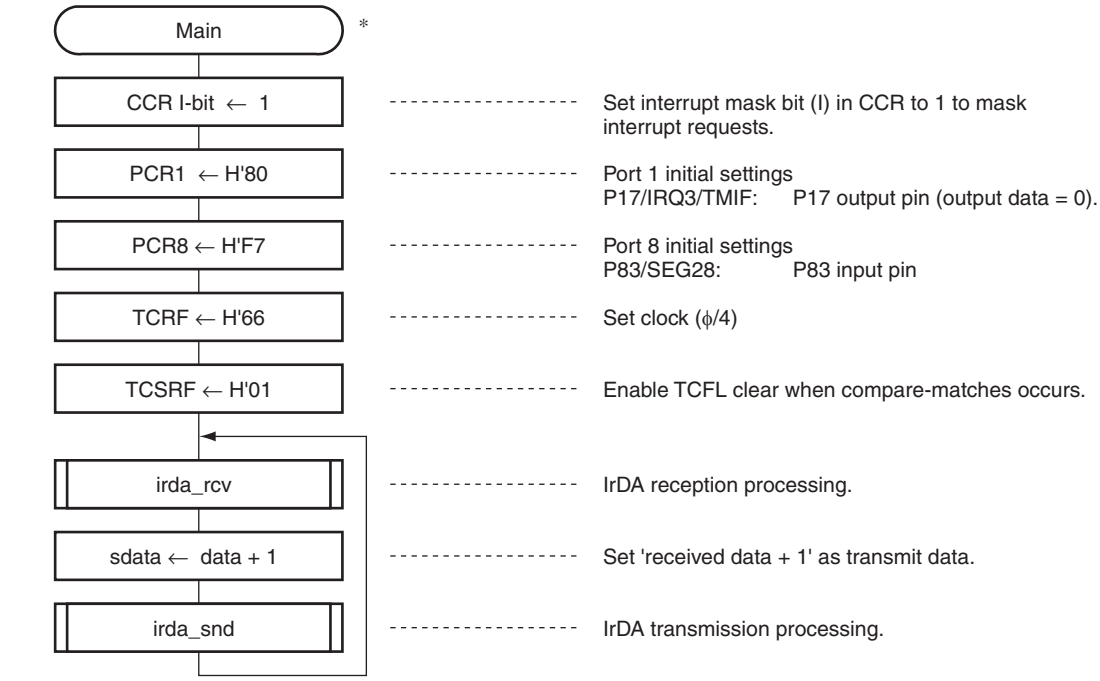
Table 4.3 describes the RAM used in this sample task.

Table 4.3 Description of RAM

Label	Function	Address	Used in
sdata	Transmit data (1 byte)	H'FB86	main, tmrw
rdata	Receive data (1 byte)	H'FB87	main, tmrw
bitdata	Store 10-bit transmit data (10 bytes)	H'FB88	main, tmrw
i	Store loop counter (2 bytes)	H'FB80	input_key
j	Store loop counter (2 bytes)	H'FB82	input_key
bit	Store bit data during reception (1 byte)	H'FB92	main
indata	Store input data (1 byte)	H'FB93	input_key
Wtimeup	Timer F timeup flag (2 bytes)	H'FB84	input_key
bitpos	For bit on/off determination (1 byte)	H'FB94	input_key
dummy	Dummy (1 byte)	H'FB95	

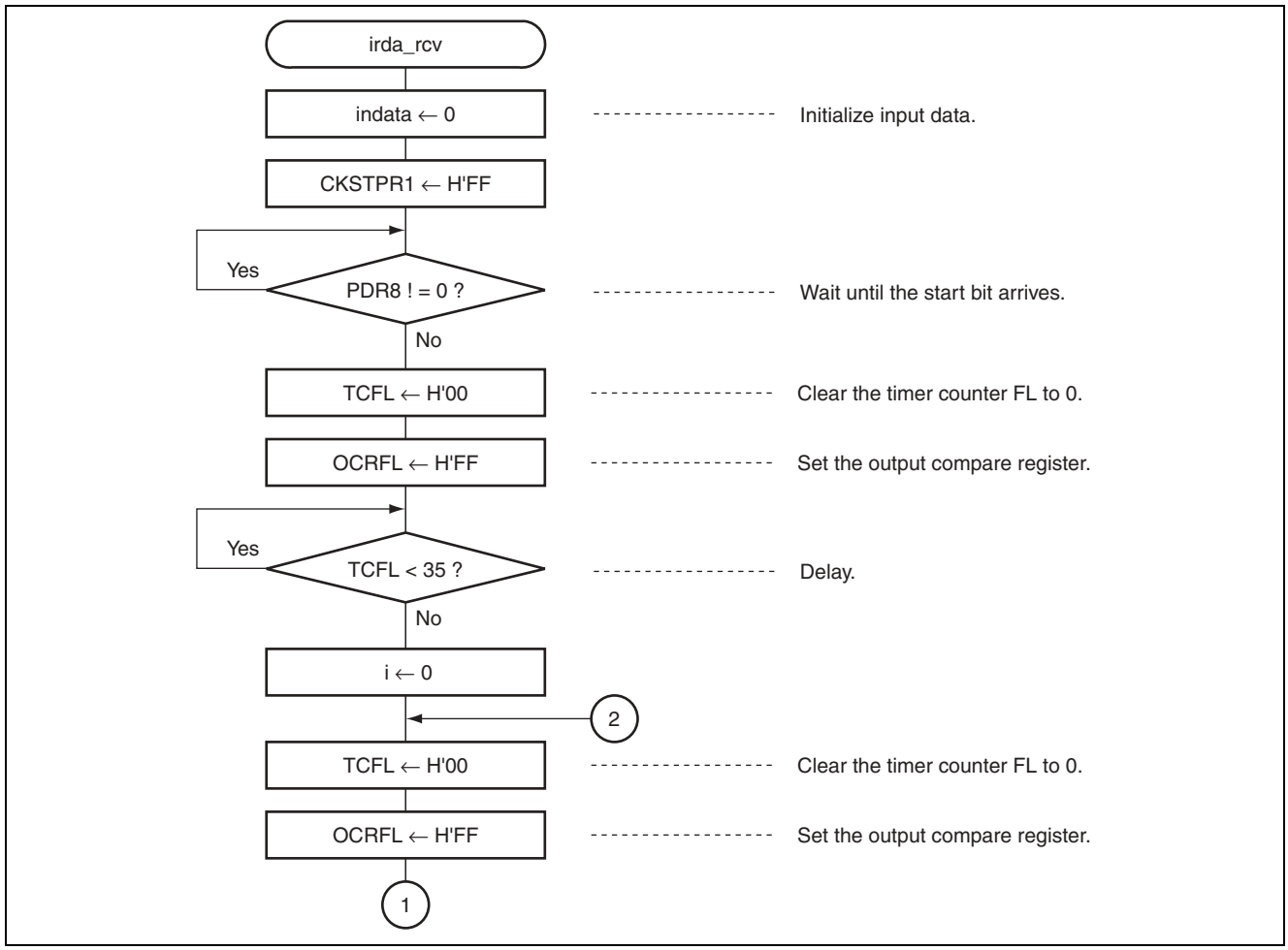
5. Flowchart

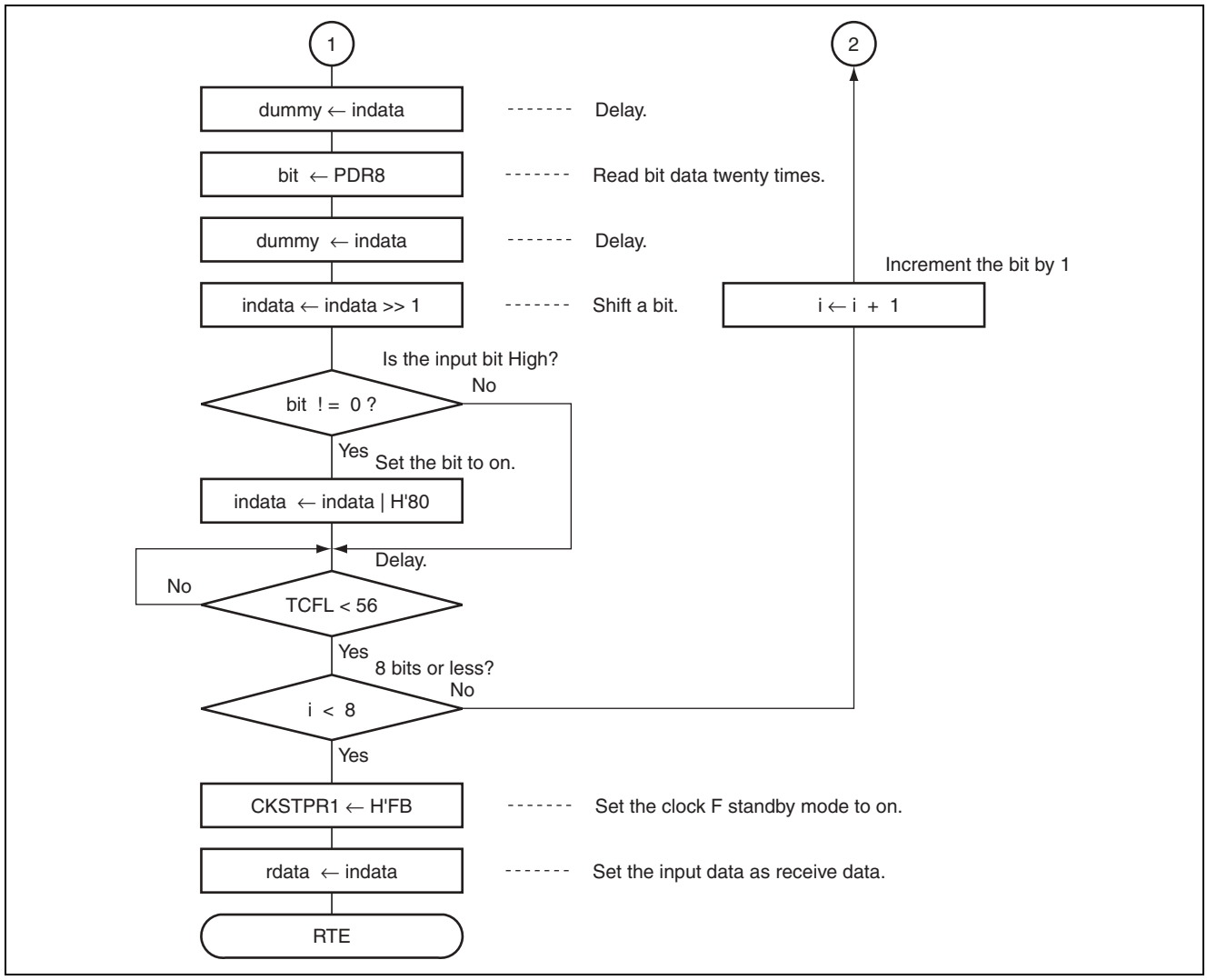
1. Main routine



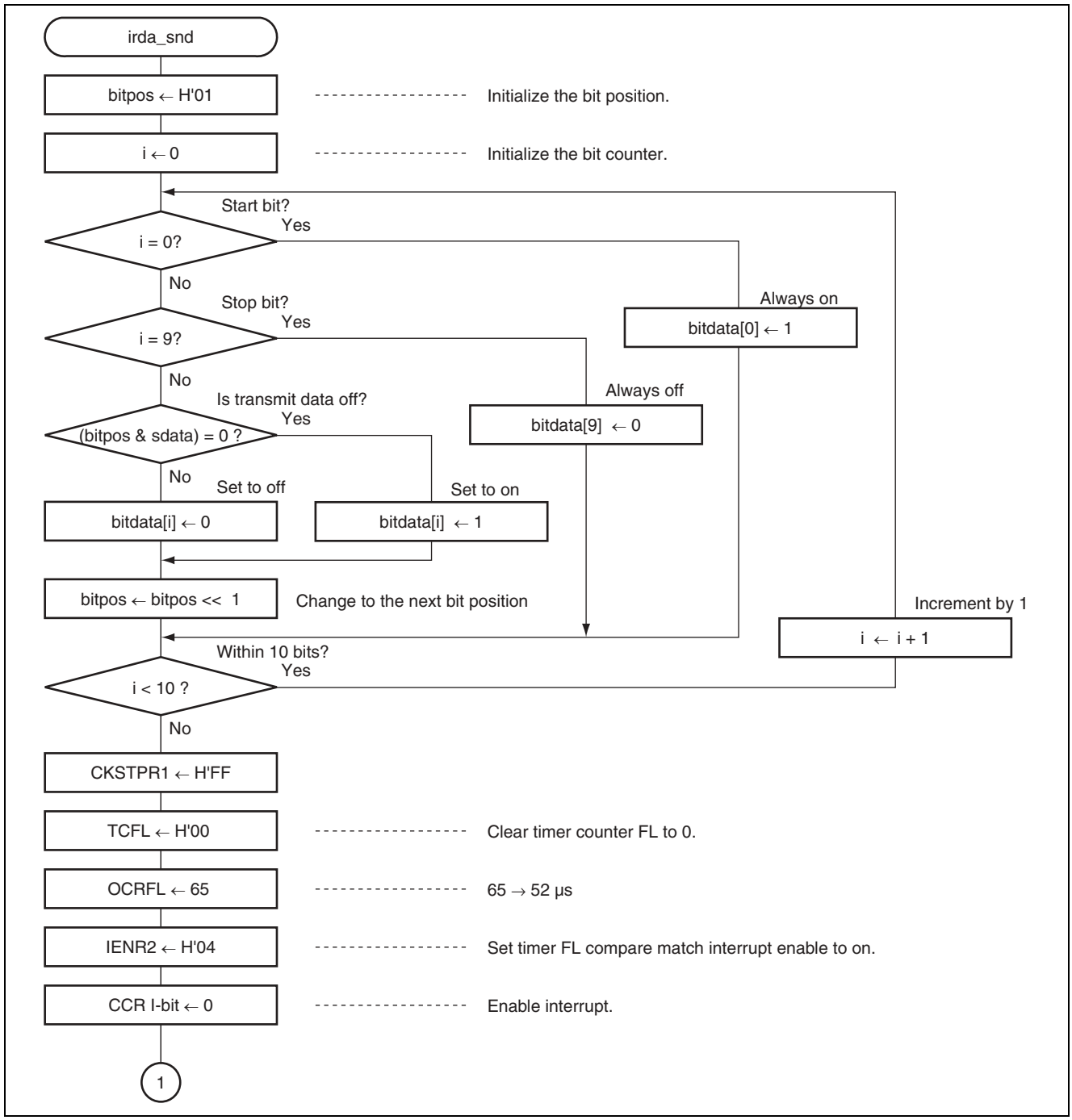
Note: * In this sample task, the stack pointer is set in INIT.SRC (assembly language).

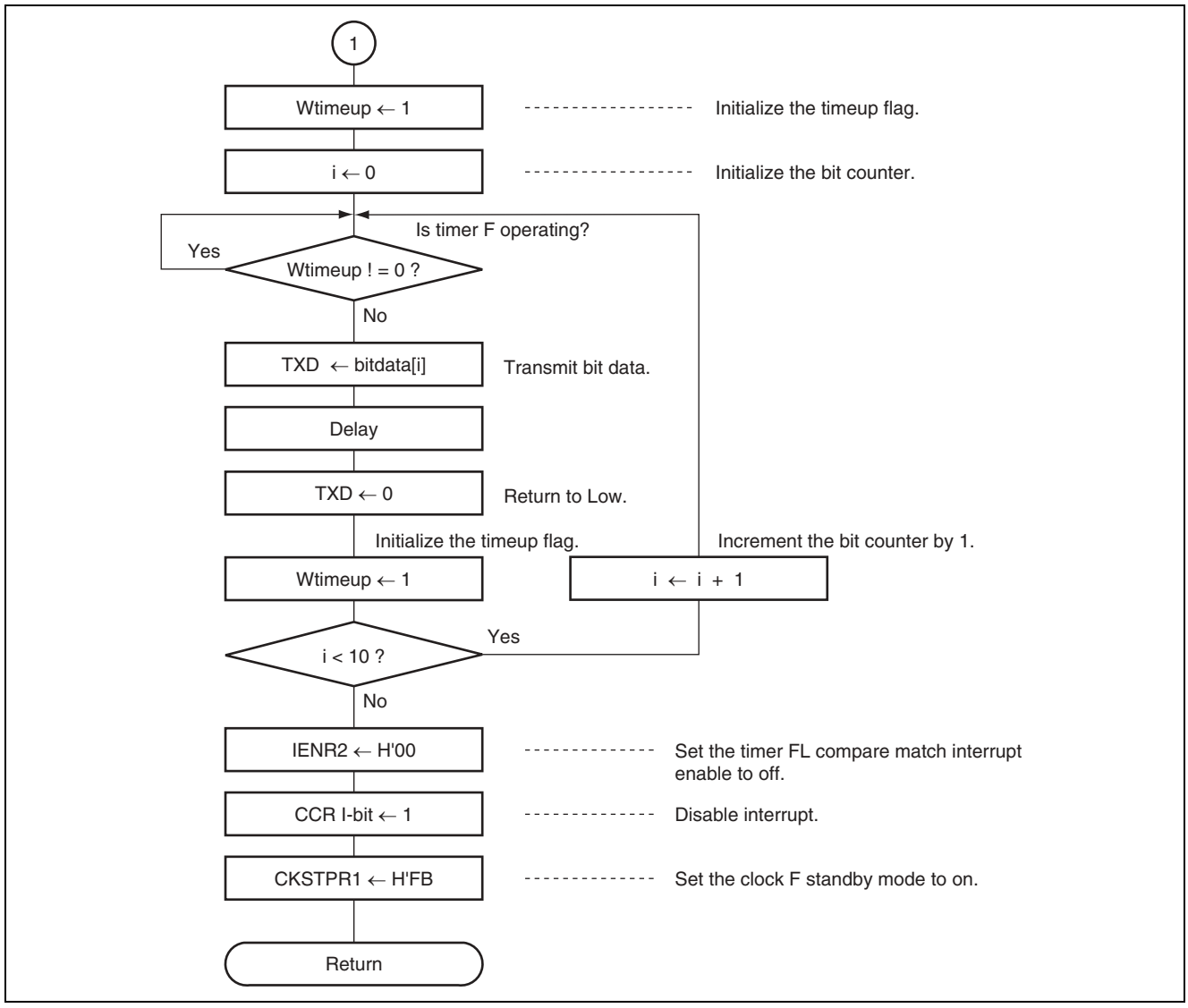
2. Infrared ray data reception processing routine (irda_rcv)



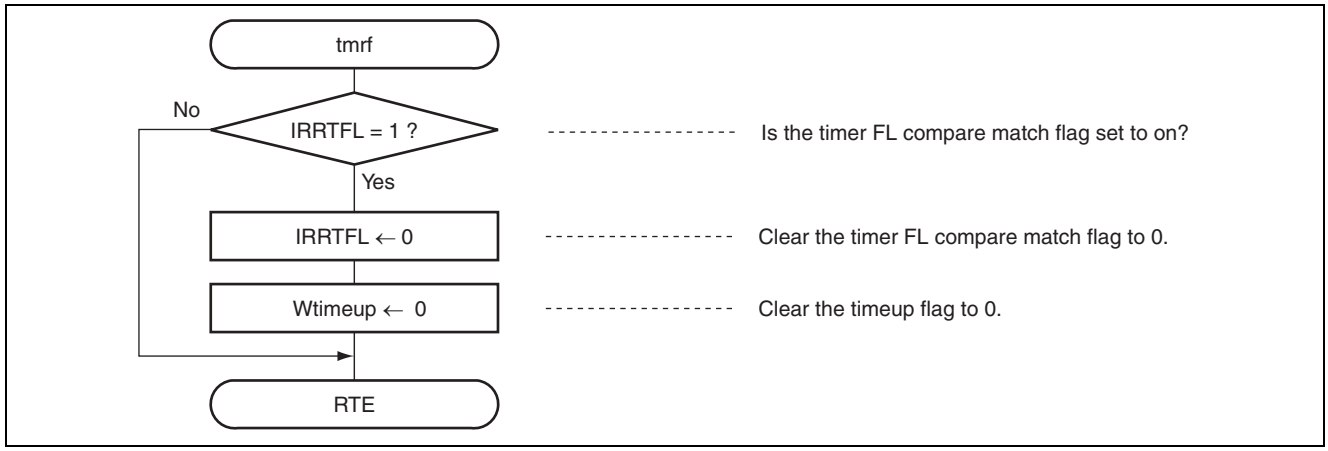

1
2
dummy ← indata
Delay.
bit ← PDR8
Read bit data twenty times.
dummy ← indata
Delay.
Increment the bit by 1
indata ← indata >> 1
Shift a bit.
i ← i + 1
Is the input bit High?
No
bit != 0 ?
Yes
Set the bit to on.
indata ← indata | H'80
Delay.
No
TCFL < 56
Yes
8 bits or less?
No
i < 8
Yes
CKSTPR1 ← H'FB
Set the clock F standby mode to on.
rdata ← indata
Set the input data as receive data.
RTE

3. Infrared ray data transmission processing routine (irda_snd)





4. Timer F interrupt processing routine (tmrw)



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                                                                                                    */
/*                                                                                                                      */
/*  ' Using an Infrared Transceiver to Transmit and Receive Character Data '                                            */
/*                                                                                                                      */
/*  Function                                                                                                            */
/*  : IrDA                                                                                                              */
/*                                                                                                                      */
/*  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    PDR1         *(volatile unsigned char *)0xFFD4        /* Port Data Register 1                  */
#define    PDR1_BIT     (*(struct BIT *)0xFFD4)
#define    TXD          PDR1_BIT.b7                              /* Transfer Data                         */
#define    PCR1         *(volatile unsigned char *)0xFFE4        /* Port Control Register 1               */

#define    PDR8         *(volatile unsigned char *)0xFFDB        /* Port Data Register 8                  */
#define    PDR8_BIT     (*(struct BIT *)0xFFDB)
#define    RXD          PDR8_BIT.b1                              /* Receive Data                          */
#define    PCR8         *(volatile unsigned char *)0xFFEB        /* Port Control Register 8               */
```

```
#define    TCRF         *(volatile unsigned char *)0xFFB6        /* Timer Control Register F                */
#define    TCSRF        *(volatile unsigned char *)0xFFB7        /* Timer Control Status Register F         */
#define    TCSRF_BIT    (*(struct BIT *)0xFFB7)
#define    CMFL         TCSRF_BIT.b2                             /* Compare-Match Flag L                    */
#define    TCF          *(volatile unsigned int  *)0xFFB8        /* 16 bit Timer Counter F                  */
#define    TCFH         *(volatile unsigned char *)0xFFB8        /* 8 bit Timer Counter F(HIGH)             */
#define    TCFL         *(volatile unsigned char *)0xFFB9        /* 8 bit Timer Counter F(LOW)              */
#define    OCRF         *(volatile unsigned int  *)0xFFBA        /* 16 bit Output Compare Register          */
#define    OCRFH        *(volatile unsigned char *)0xFFBA        /* 8 bit Output Compare Register F(HIGH)   */
#define    OCRFL        *(volatile unsigned char *)0xFFBB        /* 8 bit Output Compare Register F(LOW)    */
#define    CKSTPR1      *(volatile unsigned char *)0xFFFA        /* Clock Stop Register 1                   */
#define    IENR2        *(volatile unsigned char *)0xFFF4        /* Interrupt Enable Register 2             */
#define    IRR2         *(volatile unsigned char *)0xFFF7        /* Interrupt Request Register 2            */
#define    IRR2_BIT     (*(struct BIT *)0xFFF7)
#define    IRRTFL       IRR2_BIT.b2                              /* Timer FL Interrupt Enable               */

#pragma interrupt (tmrf)
/************************************************************************************************************/
/* Function Define                                                                                          */
/************************************************************************************************************/
extern void INIT( void );                                       /* Stack Pointer Set                       */
void main(void);                                                /* Main Routine                            */
void irda_rcv( void );
void irda_snd( void );
void tmrf(void);                                                /* Timer A Interrupt Routine               */

/************************************************************************************************************/
/*  RAM Define                                                                                              */
/************************************************************************************************************/
volatile unsigned char   sdata;                                 /* Send Data                               */
volatile unsigned char   rdata;                                 /* Receive Data                            */
unsigned char bitdata[10];                                      /* Bit Data (send)                         */
int i,j;                                                        /* Loop Counter                            */
unsigned char bit, indata;                                      /* Input Data                              */
volatile int             Wtimeup;                               /* F Timer Time Up                         */
unsigned char bitpos;                                           /* Bit Position                            */
char dummy;

/************************************************************************************************************/
/*  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) = {
     tmrf                                                       /* 0x001C Timer F Interrupt Vector         */
};
#pragma section                                                 /* P                                       */
```

```
/*****************************************************************************************************************/
/* Main Program                                                                                                  */
/*****************************************************************************************************************/
void main( void )
{
     set_imask_ccr(1);                                    /* CCR I-bit = 1                          */

     PCR1 = 0x80;                                         /* Initialize for IrDA                    */
     PCR8 = 0xF7;

     TCRF = 0x66;                                         /* Set Internal Clock: phi/4              */
     TCSRF = 0x01;                                        /* Enable TCF Clear                       */

     while(1){
          irda_rcv();                                     /* Receive Routine                        */
          sdata = rdata + 1;
          irda_snd();                                     /* Send Routine                           */
     }

}

/*****************************************************************************************************************/
/*  Receive Routine                                                                                              */
/*****************************************************************************************************************/
void irda_rcv( void )
{
     indata = 0;                                          /* Input Data Initialize                  */

      CKSTPR1 = 0xFF;                                     /* Clock F STAND-BY-MODE OFF              */

     while(PDR8);
     TCFL = 0x00;                                         /* Clear Timer Counter FL to 0            */
     OCRFL = 0x0FF;                                       /* Set Output Compare Register            */
     while(TCFL < 35);                                    /* 28us                                   */

     for(i = 0;i<8;i++){
          TCFL = 0x00;                                    /* Clear Timer Counter FL to 0            */
          OCRFL = 0x0FF;                                  /* Set Output Compare Register            */
          dummy = indata;                                 /* Dummy Wait(8 cycle)                    */
          bit = PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8
               & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8;
          dummy = indata;                                 /* Dummy Wait(8 cycle)                    */
          indata >> = 1;                                  /* Shift Input Data                       */
          if(bit) {                                       /* If Input is High level                 */
               indata | = 0x80;                           /* Then Data Set                          */
          }
          while(TCFL < 56);                               /* 44.8us                                 */
     }
     CKSTPR1 = 0xFB;                                      /* Clock F STAND-BY-MODE ON               */

     rdata = indata;
}
```

```
/*****************************************************************************************************************/
/*  Send Routine                                                                                                 */
/*****************************************************************************************************************/
void irda_snd( void )
{
                                                        /* Set Send Data                                        */
     bitpos = 0x01;                                     /* Set Bit Position (0 Bit)                             */
     for(i = 0; i < 10; i++) {
          if(i == 0) {
               bitdata[0] = 1;                          /* Set Start Bit                                        */
          } else if(i == 9) {
               bitdata[9] = 0;                          /* Set Stop Bit                                         */
          } else {
               if((bitpos & sdata) == 0) {              /* Set Data Bit                                         */
                    bitdata[i] = 1;
               } else {
                    bitdata[i] = 0;
               }
               bitpos <<= 1;                            /* Shift Bit Position                                   */
          }
     }

      CKSTPR1 = 0xFF;                                   /* Clock F STAND-BY-MODE OFF                            */
     TCFL = 0x00;                                       /* Clear Timer Counter FL to 0                          */
     OCRFL = 65;                                        /* Set Output Compare Register                          */
     IENR2 = 0x04;                                      /* FL Interrupt Enable                                  */
     set_imask_ccr(0);                                  /* CCR I-bit = 0                                        */

     Wtimeup = 1;
     for(i = 0; i < 10; i++) {
          while(Wtimeup);                               /* Wait 52us                                            */
          TXD = bitdata[i];                             /* TXD ON                                               */
          for(j = 0; j < 1; j++);
          TXD = 0;                                      /* TXD OFF                                              */
          Wtimeup = 1;
     }

     IENR2 = 0x00;                                      /* FH Interrupt Disable                                 */
     set_imask_ccr(1);                                  /* CCR I-bit = 1                                        */
     CKSTPR1 = 0xFB;                                    /* Clock F STAND-BY-MODE ON                             */

}

/*****************************************************************************************************************/
/*  Timer F Interrupt (every 52us)                                                                               */
/*****************************************************************************************************************/
void tmrf(void)
{
     if ( IRRTFL == 1 ) {
          IRRTFL = 0;                                   /* Clear Compare Match Flag A                           */
          Wtimeup = 0;                                  /* Set Time Up                                          */
     }
}
```

Link address specifications

Section Name	Address
CV1	H'0000
CV2	H'001C
P	H'0100
B	H'FB80

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.