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

Simultaneous Transmission/Reception in Synchronous Mode

Introduction

Four bytes of 8-bit data are simultaneously transmitted and received by serial data transfer in synchronous mode.

Target Device

H8/300H Tiny Series H8/3664

Contents

1. Specification 2
2. Description of Functions Used 3
3. Operational Description 6
4. Description of Software 7
5. Flowcharts 10
6. Program Listing 12

1. Specification

1. Four bytes of 8-bit data are simultaneously transmitted and received by serial data transfer in synchronous mode, as shown in figure 1.
2. Data is simultaneously transmitted and received in a 4 µs clock cycle with an internal clock used as the transmit/receive clock (serial clock).
3. The data length of the transmit/receive data is eight bits, and data is transmitted in the LSB-first format, starting from the least significant bit.

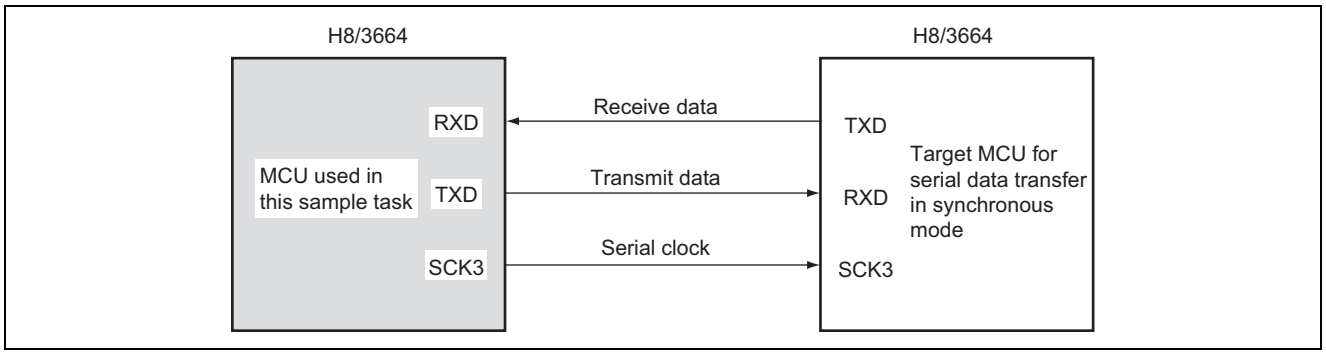
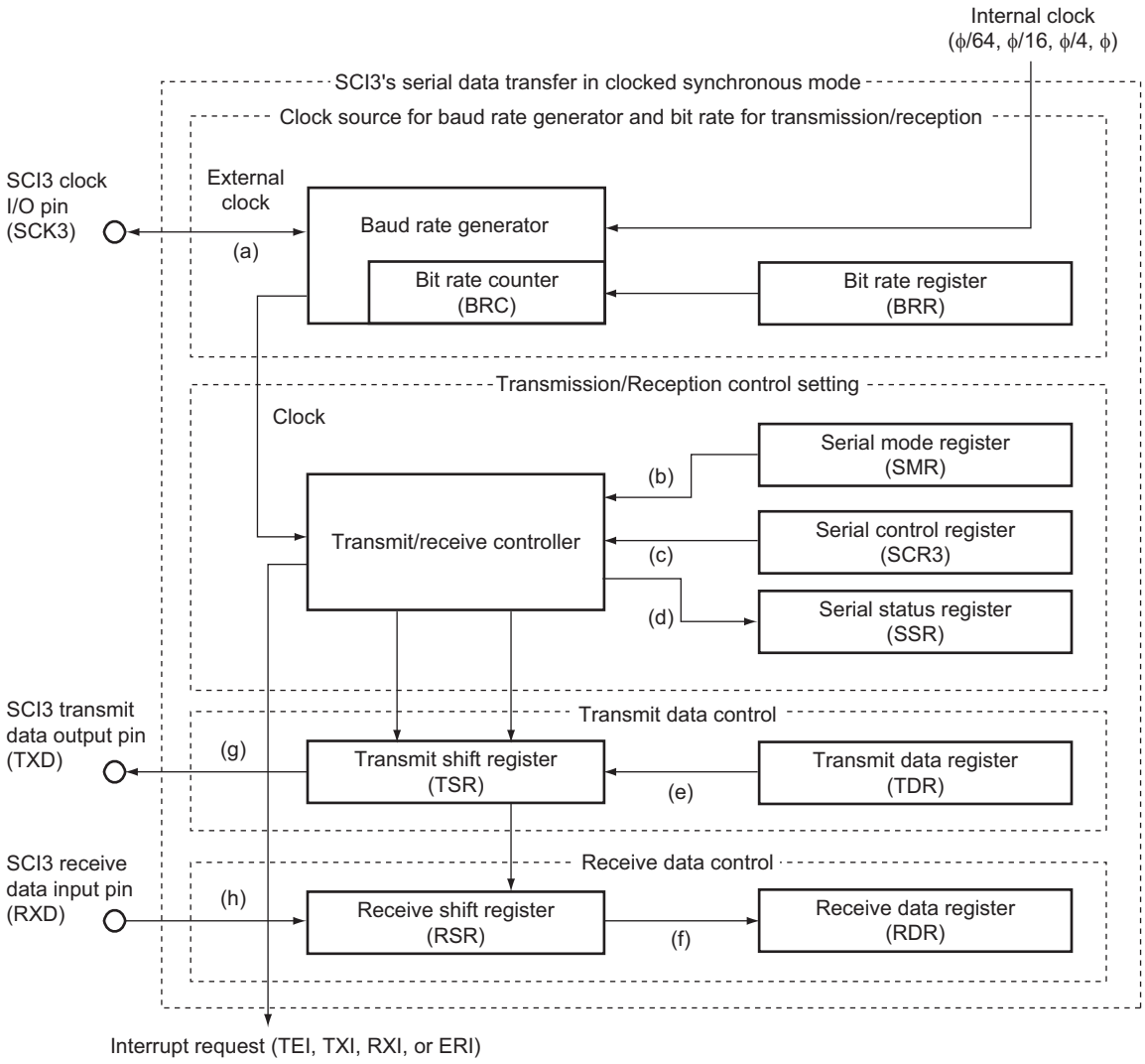


Figure 1 Simultaneous Serial Transmission/Reception in Synchronous Mode

2. Description of Functions Used

1. In this sample task, serial data is simultaneously transmitted and received in synchronous mode via the serial communication interface (SCI). Figure 2 is a block diagram of simultaneous serial transmission/reception in synchronous mode. The elements of the block diagram are described below.

- The system clock (ϕ) is a 16 MHz OSC clock that is used as a reference clock for operating the CPU and peripheral functions.
- Only overrun errors are detected as receive errors.
- In synchronous mode, the data length is eight bits.
- The receive shift register (RSR) is a register used to receive serial data. Serial data input to RSR from the RXD pin is set in the order in which it is received, starting from the LSB (bit 0), and converted to parallel data. When one byte of data is received, it is transferred to RDR automatically. RSR cannot be read from or written to directly by the CPU.
- The receive data register (RDR) is an 8-bit register that stores received serial data. When reception of one byte of data is finished, the received data is transferred from RSR to RDR, and the receive operation is completed. RSR is then enabled for reception. RSR and RDR are double-buffered, allowing consecutive receive operations. RDR is a read-only register, and cannot be written to by the CPU.
- The transmit shift register (TSR) is a register used to transmit serial data. Transmit data is first transferred from TDR to TSR, and serial data transmission is carried out by sending the data to the TXD pin in order, starting from the LSB (bit 0). When one byte of data is transmitted, the next byte of transmit data is automatically transferred from TDR to TSR, and transmission is started. Data transfer from TDR to TSR is not performed if no data has been written to TDR (if bit TDRE is set to 1). TSR cannot be read from or written to directly by the CPU.
- The transmit data register (TDR) is an 8-bit register that stores transmit data. When TSR is found to be empty, the transmit data written in TDR is transferred to TSR, and serial data transmission is started. Continuous transmission is possible by writing the next transmit data to TDR during TSR serial data transmission. TDR can be read from or written to by the CPU at any time.
- The serial mode register (SMR) is an 8-bit register used to set the serial data transfer format and to select the clock source for the baud rate generator.
- Serial control register 3 (SCR3) is an 8-bit register for selecting transmit or receive operation and the serial clock source.
- The serial status register (SSR) is an 8-bit register containing status flags that indicate the operational status of SCI3, and multiprocessor bits. Bits TDRE, RDRF, OER, PER, and FER can only be cleared to 0.
- The serial clock can be selected from a total of nine clocks: eight internal clocks and an external clock. When an internal clock is selected, the SCK3 pin functions as an output pin. When clock consecutive output mode is selected, the selected clock is consecutively output from the SCK3 pin. When an external clock is selected, the SCK3 pin functions as an input pin.
- In this sample task, the source of the serial clock (baud rate generator) is system clock/64 and the serial clock cycle is 4 μs.
- The SCI3 data transfer format uses a fixed 8-bit data length, and data is transmitted in the LSB-first format, starting from the least significant bit. Transmit data is output from one falling edge of the serial clock until the next falling edge. Receive data is latched at the rising edge of the serial clock.
- In this sample task, the data transfer format is set to an 8-bit data length, and 8-bit data is transmitted and received.
- The SCI3 clock (SCK3) pin is clock I/O pin of SCI3.
- The SCI3 receive data input (RXD) pin is the input pin for SCI3 receive data.
- The SCI3 transmit data output (TXD) pin is the output pin for SCI3 transmit data.



Notes: (a) Outputs clock source (φ/64) for baud rate generator set in SMR.
(b) Sets serial data transfer format and clock source for baud rate generator.
(c) Selects transmission or reception, and clock output pin in synchronous mode.
(d) Indicates operational status of SCI3 by status flags (transmit data register empty, receive data register full, and overrun error).
(e) When TSR is found to be empty, sends the transmit data written in TDR to TSR.
(f) When one byte of data is received, sends receive data in RSR to RDR.
(g) Transmits transmit data.
(h) Receives receive data.

Figure 2 Simultaneous Serial Transmission/Reception in Synchronous Mode

2. Table 1 lists the function allocation for this sample task. The functions listed in table 1 are allocated for simultaneous serial transmission/reception in clocked synchronous mode.

Table 1 Function Allocation

Function	Function Assignment
TSR	Transmits serial data
TDR	Stores transmit data
SMR	Sets the serial data transfer format and clock source for the baud rate generator
SSR	Status flags indicating the operational status of SCI3
SCR3	Selects transmit operation and sets the SCK3 pin to function as a clock output pin
SCK3	SCI3 clock output pin
TXD	SCI3 transmit data output pin
RXD	SCI3 receive data input pin
PMR1	Sets TXD output pin for SCI3 transmission

3. Operational Description

Figure 3 shows this sample task's principle of operation. The hardware and software processing shown in figure 3 performs simultaneous serial transmission/reception in clocked synchronous mode.

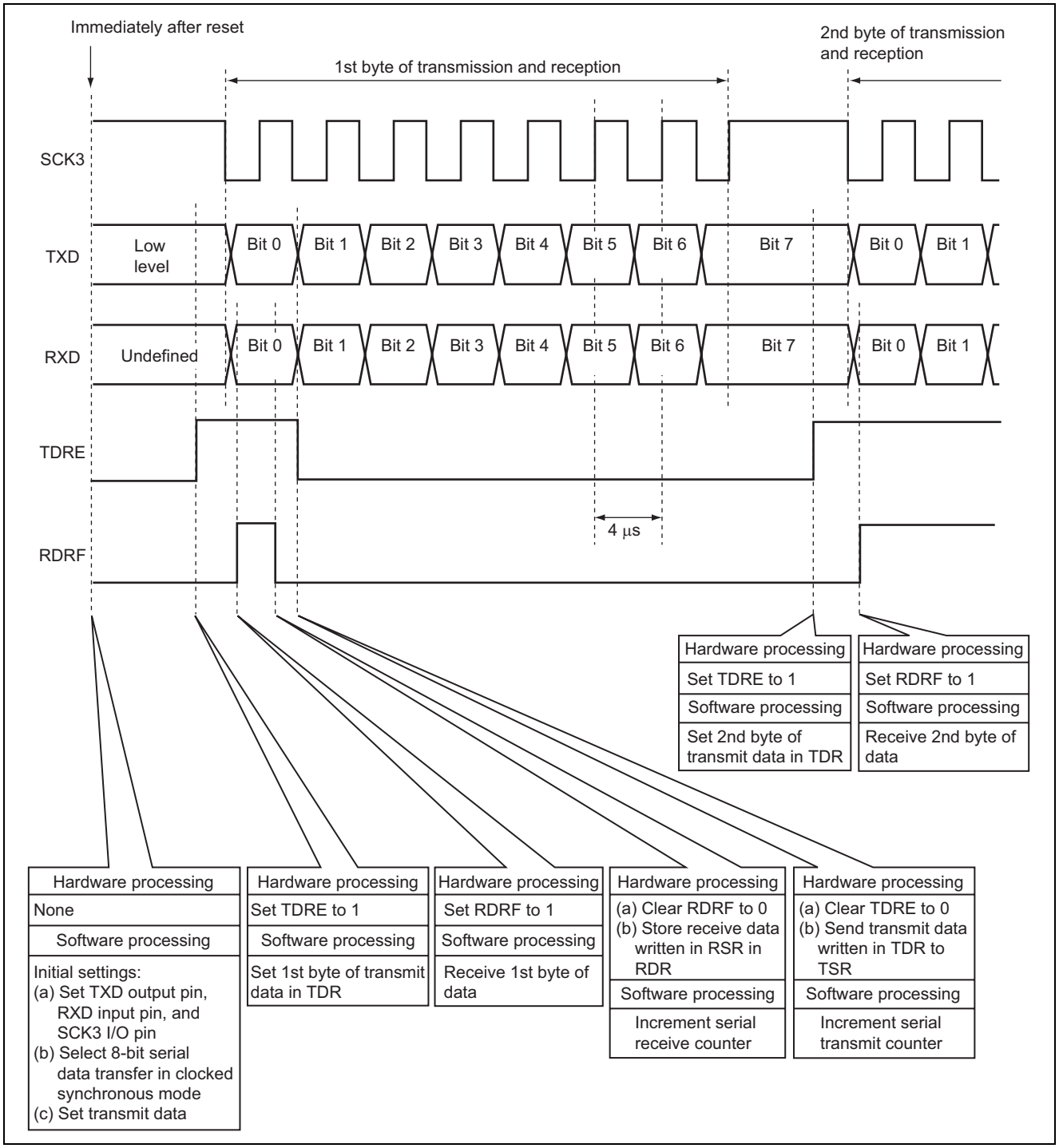


Figure 3 Operation Principle: Simultaneous Serial Transmission/Reception in Synchronous Mode

4. Description of Software

4.1 Description of Modules

Table 2 describes the software used in this sample task.

Table 2 Description of Module

Module Name	Label Name	Function
Main routine	main	Sets the transmit data, selects serial data transmission and reception in clocked synchronous mode, stores the receive data in RAM, and stops SCI3 after four bytes of data have been transmitted and received.

4.2 Description of Arguments

Table 3 describes the arguments used in this sample task.

Table 3 Description of Arguments

Argument Name	Function	Used in	Data Length	I/O
STD0 to STD3	Serial transmit data in synchronous mode	Main routine	1 byte	Output
SRD0 to SRD3	Serial receive data in synchronous mode	Main routine	1 byte	Input

4.3 Description of Internal Registers

Table 4 describes the internal registers used in this sample task.

Table 4 Description of Internal Registers

Register	Name	Functional Description	Address	Setting
SMR	COM	Serial mode register (communication mode): When COM is set to 1, the communication mode is set to clocked synchronous mode.	H'FFA8 Bit 7	1
	MP	Serial mode register (multiprocessor mode): This bit must be cleared to 0 (multiprocessor communication function disabled) in clocked synchronous mode.	H'FFA8 Bit 2	0
	CKS1 CKS0	Serial mode register (clock select 1 and 0): When CKS1 and CKS0 are both set to 1, the clock source for the baud rate generator is set to system clock/64.	H'FFA8 Bit 1 Bit 0	CKS1 = 1 CKS0 = 1
SCR3	TE	Serial control register 3 (transmit enable): When TE is set to 1, transmit operation is enabled.	H'FFAA Bit 5	0
	RE	Serial control register 3 (receive enable): When RE is set to 1, receive operation is enabled.	H'FFAA Bit 4	0
	CKE1 CKE0	Serial control register 3 (clock enable): When CKE1 and CKE0 are both cleared to 0, the clock source is set to an internal clock and the SCK3 pin functions as a clock output pin in synchronous mode.	H'FFAA Bit 1 Bit 0	CKE1 = 0 CKE0 = 0
TDR		Transmit data register: 8-bit register that stores the transmit data.	H'FFAB	—
SSR	TDRE	Serial status register (transmit data empty): When TDRE is cleared to 0, the transmit data written in TDR has not been sent to TSR. When TDRE is set to 1, the transmit data has not been written to TDR, or the transmit data written in TDR has been sent to TSR.	H'FFAC Bit 7	1
	RDRF	Serial status register (receive data full): When RDRF is cleared to 0, no receive data is stored in RDR. When RDRF is set to 1, receive data is stored in RDR.	H'FFAC Bit 6	1
	OER	Serial status register (overrun error): When OER is cleared to 0, reception is in progress or completed. When OER is set to 1, an overrun error has occurred during reception.	H'FFAC Bit 5	0
	TEND	Serial status register (transmit end): When TEND is cleared to 0, transmission is in progress. When TEND is set to 1, transmission has completed.	H'FFAC Bit 2	1
RDR		Receive data register: 8-bit register that stores the receive data.	H'FFAD	—
PMR1	PMR11	Port mode register 1 (P22/TXD pin function switch): When PMR11 is set to 1, the P22/TXD pin functions as the TXD output pin.	H'FFE0 Bit 1	1

4.4 Description of RAM

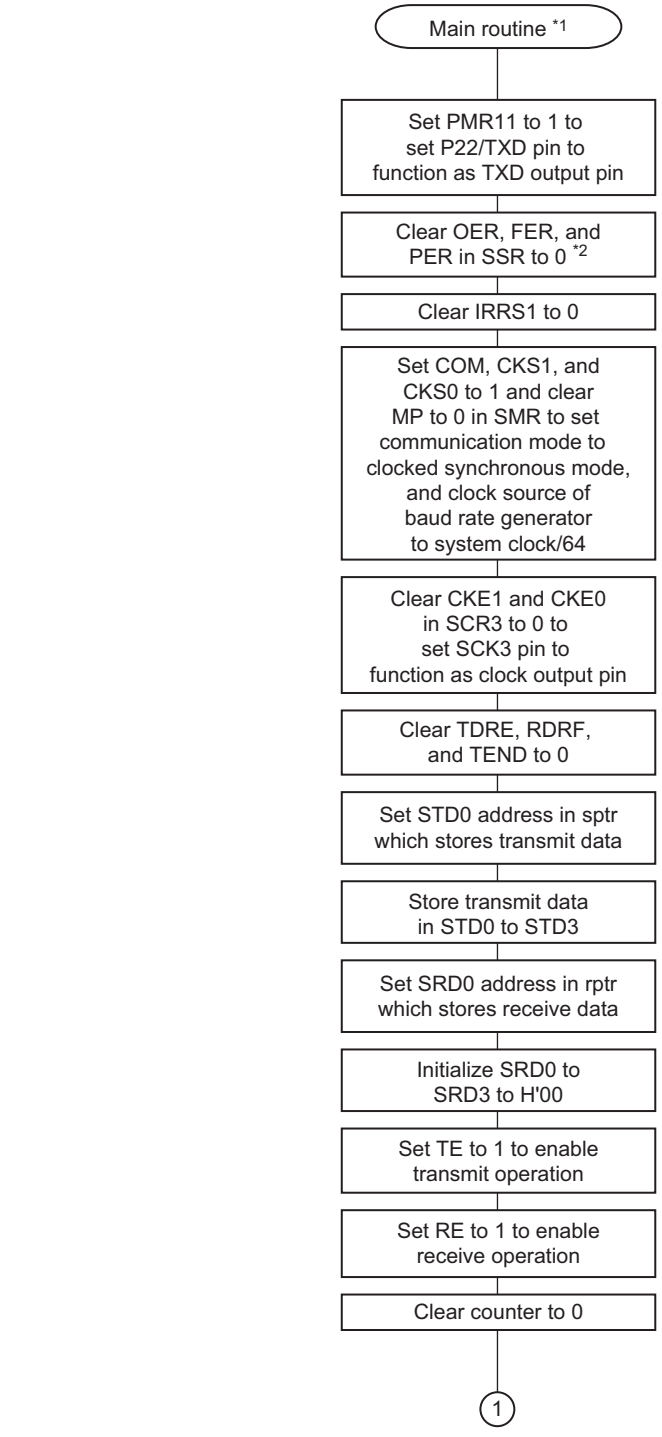
Table 5 describes the RAM used in this sample task.

Table 5 Description of RAM

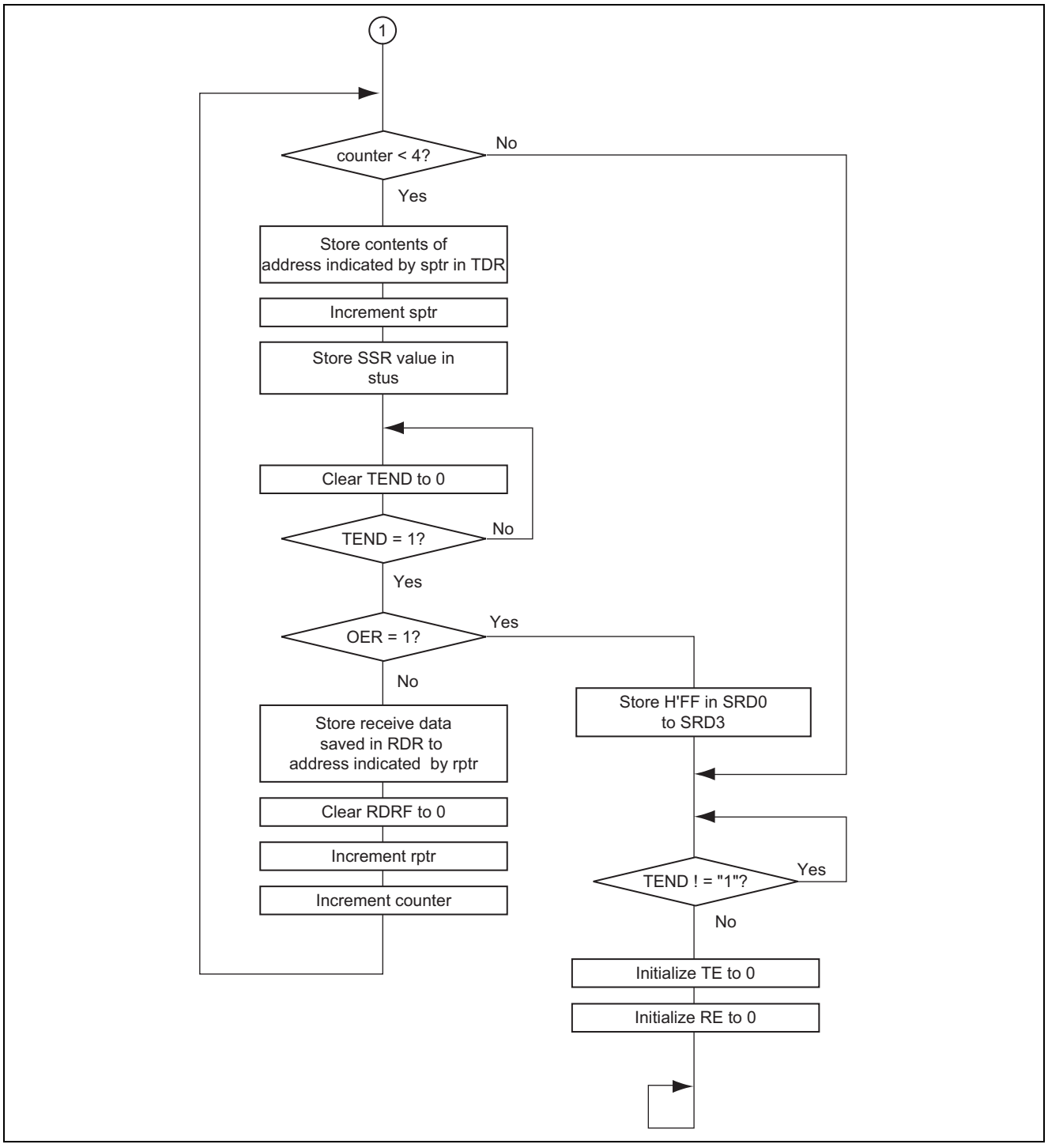
Label Name	Function	Address	Used in
STD0	Stores the first byte of transmit data in serial data transmission in clocked synchronous mode	H'FB80	Main routine
STD1	Stores the second byte of transmit data in serial data transmission in clocked synchronous mode	H'FB81	Main routine
STD2	Stores the third byte of transmit data in serial data transmission in clocked synchronous mode	H'FB82	Main routine
STD3	Stores the fourth byte of transmit data in serial data transmission in clocked synchronous mode	H'FB83	Main routine
SRD0	Receives the first byte of receive data in serial data reception in clocked synchronous mode	H'FB84	Main routine
SRD1	Receives the second byte of receive data in serial data reception in clocked synchronous mode	H'FB85	Main routine
SRD2	Receives the third byte of receive data in serial data reception in clocked synchronous mode	H'FB86	Main routine
SRD3	Receives the fourth byte of receive data in serial data reception in clocked synchronous mode	H'FB87	Main routine
counter	8-bit counter for counting four simultaneous serial transmit and receive operations in clocked synchronous mode	H'FB88	Main routine

5. Flowcharts

1. Main Routine



Notes: 1. In this sample task, the stack pointer is set in INIT.SRC (assembly).
2. In clocked synchronous mode (simultaneous transmission and reception), OER, FER, and PER must be cleared to 0.



5.1 Link Address Designation

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

6. Program Listing

INIT.SRC (Program listing)

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

```
/************************************************************/
/*                                                          */
/*  H8/300H Tiny Series -H8/3664-                           */
/*  Application Note                                        */
/*                                                          */
/*  'Synchronous Serial Data Simultaneous                   */
/*   Transmission/Reception'                                */
/*                                                          */
/*  Function                                                */
/*  : Serial Communication Interface                        */
/*     Synchronous Serial Interface                         */
/*     -Transmitting/Receiving                              */
/*                                                          */
/*  External Clock : 16MHz                                  */
/*  Internal Clock : 16MHz                                  */
/*  Sub Clock      : 32.768kHz                              */
/*                                                          */
/************************************************************/

#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    SMR_BIT (*(struct BIT *)0xFFA8)            /* Serial Mode Register           */
#define    COM      SMR_BIT.b7                        /* Communication Mode             */
#define    CHR      SMR_BIT.b6                        /* Character Length               */
#define    PE       SMR_BIT.b5                        /* Parity Enable                  */
#define    PM       SMR_BIT.b4                        /* Parity Mode                    */
#define    STOP     SMR_BIT.b3                        /* Stop Bit Length                */
```

```
#define    MP       SMR_BIT.b2                        /* Multiprocesor Mode              */
#define    CKS1     SMR_BIT.b1                        /* Clock Slsct 1                   */
#define    CKS0     SMR_BIT.b0                        /* Clock Slsct 0                   */
#define    BRR      *(volatile unsigned char *)0xFFA9  /* Bit Rate Register               */
#define    SCR3_BIT (*(struct BIT *)0xFFAA)           /* Serial Control Register 3       */
#define    TIE      SCR3_BIT.b7                       /* Transmit Interrupt Enable       */
#define    RIE      SCR3_BIT.b6                       /* Receive Interrupt Enable        */
#define    TE       SCR3_BIT.b5                       /* Transmit Enable                 */
#define    RE       SCR3_BIT.b4                       /* Receive Enable                  */
#define    MPIE     SCR3_BIT.b3                       /* Multiprocessor Interrupt Enable */
#define    TEIE     SCR3_BIT.b2                       /* Transmit End Interrupt Enable   */
#define    CKE1     SCR3_BIT.b1                       /* Clock Enable 1                  */
#define    CKE0     SCR3_BIT.b0                       /* Clock Enable 0                  */

#define    TDR      *(volatile unsigned char *)0xFFAB  /* Transmit Data Register          */
#define    SSR      *(volatile unsigned char *)0xFFAC  /* Serial Status Register          */
#define    SSR_BIT (*(struct BIT *)0xFFAC)            /* Serial Status Register          */
#define    TDRE     SSR_BIT.b7                        /* Transmit Data Register Empty    */
#define    RDRF     SSR_BIT.b6                        /* Receive Data Register Full      */
#define    OER      SSR_BIT.b5                        /* Overrun Erorr                   */
#define    FER      SSR_BIT.b4                        /* Framing Erorr                   */
#define    PER      SSR_BIT.b3                        /* Parity Erorr                    */
#define    TEND     SSR_BIT.b2                        /* Transmit End                    */
#define    MPBR     SSR_BIT.b1                        /* Multiprocessor Bit Receive      */
#define    MPBT     SSR_BIT.b0                        /* Multiprocessor Bit Transfer     */
#define    PMR1_BIT (*(struct BIT *)0xFFE0)           /* Port Mode Register 1            */
#define    PMR11    PMR1_BIT.b1                       /* Port Mode Register 1 Bit 1      */
#define    RDR      *(volatile unsigned char *)0xFFAD  /* Receive Data Register           */

/*********************************************************/
/*     Function Definition                               */
/*********************************************************/
extern  void    INIT( void );                          /* SP Set                          */
void    main    ( void );

/*********************************************************/
/*   RAM Allocation                                      */
/*********************************************************/
    unsigned char   STD[4];
    unsigned char   SRD[4];
    unsigned char   counter;

/*********************************************************/
/*  Vector Address                                       */
/*********************************************************/
#pragma section     V1                                 /* VECTOR SECTOIN SET              */
void (*const VEC_TBL1[])(void) = {
/* 0x00 - 0x0f */
    INIT                                               /* 00 Reset                        */
};

#pragma section                                        /* P                               */
```

```
/*********************************************************/
/*   Main Program                                        */
/*********************************************************/
void main ( void )
{
    unsigned char   stus;
    unsigned char   *sptr,*rptr;

    PMR11 = 1;                                  /* Initialize Output Port TXD                */

    OER = 0;                                    /* Clear OER                                 */
    FER = 0;                                    /* Clear FER                                 */
    PER = 0;                                    /* Clear PER                                 */

    COM = 1;                                    /* Initialize Communication Mode             */
    MP = 0;                                     /* Initialize Multiprocesor Mode             */

    CKS1 = 1;                                   /* Initialize Clock Select 1 φ/64            */
    CKS0 = 1;                                   /* Initialize Clock Select 0 φ/64            */

    CKE1 = 0;                                   /* Initialize Clock Enable 1    Output       */
    CKE0 = 0;                                   /* Initialize Clock Enable 0    Output       */

    TDRE = 0;                                   /* Clear TDRE                                */
    RDRF = 0;                                   /* Clear RDRF                                */
    TEND = 0;                                   /* Clear TEND                                */

    sptr = &STD[0];                             /* Initialize Serial Transmitting Data Address */

    STD[0] = 0x00;                              /* Set Serial Transfer Data 0                */
    STD[1] = 0x55;                              /* Set Serial Transfer Data 1                */
    STD[2] = 0xAA;                              /* Set Serial Transfer Data 2                */
    STD[3] = 0xFF;                              /* Set Serial Transfer Data 3                */

   rptr = &SRD[0];                              /* Initialize Serial Receiving Data Address    */

    SRD[0] = 0x00;                              /* Initialize Serial Receiving Data 0        */
    SRD[1] = 0x00;                              /* Initialize Serial Receiving Data 1        */
    SRD[2] = 0x00;                              /* Initialize Serial Receiving Data 2        */
    SRD[3] = 0x00;                              /* Initialize Serial Receiving Data 3        */

    TE = 1;                                     /* Start Serial Transmitting                 */

    RE = 1;                                     /* Start Serial Receiving                    */

    for(counter = 0 ; counter < 4 ; counter++){ /* Serial Transmitting Data Counter 4 Loop   */

        TDR = *sptr;                            /* Save Serial Transmitting Data             */

        sptr++;                                 /* Increment Serial Transmittimg Data Address  */

        stus = SSR;                             /* Serial Status Register read & save        */

        while((stus & 0x04) == 0){              /* End Serial Transmitting                   */
```

```
            stus = SSR;                         /* Serial Status Register read & save          */
            TEND = 0;
        }

        if ((stus & 0x20) != 0){                /* Overrun Erorr Flag = 1 ?                    */
            SRD[0] = 0xFF;                      /* Overrun Erorr 0                             */
            SRD[1] = 0xFF;                      /* Overrun Erorr 1                             */
            SRD[2] = 0xFF;                      /* Overrun Erorr 2                             */
            SRD[3] = 0xFF;                      /* Overrun Erorr 3                             */
            break;
        }
        else    {
            *rptr = RDR;                        /* Save Serial Receiving Data                  */
            RDRF = 0;
            rptr++;                             /* Increment Serial Receiving Data Address     */
        }

    }

    while(TEND != 1){                           /* End Serial Transmitting                     */
        ;
    }
    TE = 0;                                     /* Initialize Transmitting Enable              */
    RE = 0;                                     /* Initialize Receiving Enable                 */

    while(1){
       ;
    }
}
```

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.