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

Multiprocessor Communication

Introduction

Data H'B8 is transmitted to receiving station A and data H'DE is transmitted to receiving station B by the multiprocessor communication function.

Target Device

H8/300H Tiny Series H8/3664

Contents

1. Specification

1. Data H'B8 is transmitted to receiving station A and data H'DE is transmitted to receiving station B by the multiprocessor communication function, as shown in figure 1.
2. The data transfer format set for transmit data is a data length of eight bits, an odd parity, and a stop bit length of one bit.
3. The bit rate for transmission is 31250 (bit/s). A break is output when data transmission ends.

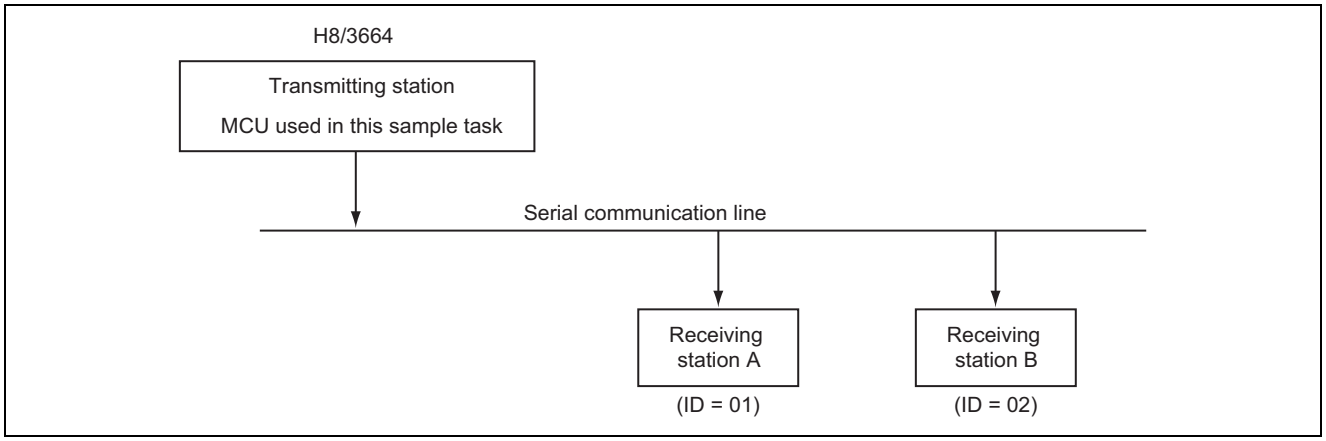


Figure 1 Multiprocessor Communication

2. Description of Functions Used

1. In this sample task, transmission by multiprocessor communication is performed via the serial communication interface (SCI). Figure 3 is a block diagram of multiprocessor communication. The elements of the block diagram are described below.

- Use of the multiprocessor communication function enables data transfer between a number of processors sharing communication lines by asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data.
- When multiprocessor communication is performed, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles; an ID transmission cycle that specifies the receiving station, and a data transmission cycle.
- The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle; if the multiprocessor bit is 0, the cycle is a data transmission cycle.
- The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added.
- When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose IDs do not match continue to skip data until data with a 1 multiprocessor bit is again received.
- There is a choice of four data transfer formats. When the multiprocessor format is selected, the parity bit setting is rendered invalid.
- Separate transmission and reception units are provided, enabling transmission and reception to be carried out simultaneously. The transmission and reception units are both double-buffered, allowing continuous transmission and reception.
- Any desired bit rate can be selected with the on-chip baud rate generator.
- An internal or external clock can be selected as the transmit/receive clock source.
- There are six interrupt sources: transmit end, transmit data empty, receive data full, overrun error, framing error, and parity error.
- 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. SMR can be read from or written to by the CPU at any time.
- Serial control register 3 (SCR3) is an 8-bit register for selecting transmit or receive operation, the asynchronous mode clock output, to enable or disable interrupt requests, and the transmit/receive clock source. SCR3 can be read from or written to by the CPU at any time.

- Figure 2 shows serial data with the multiprocessor bit and table 1 shows multiprocessor formats.

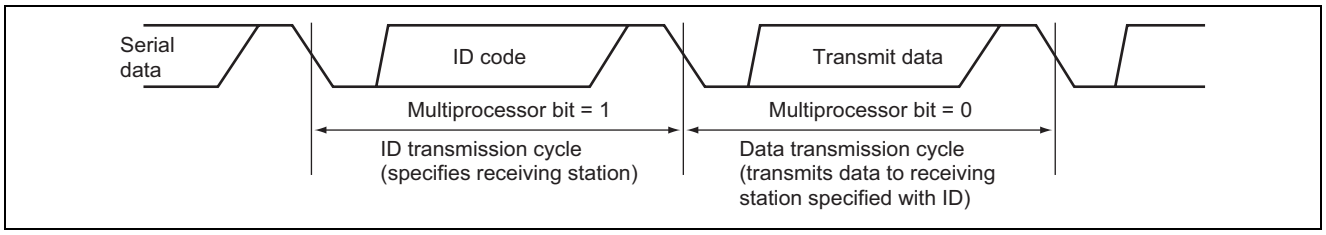
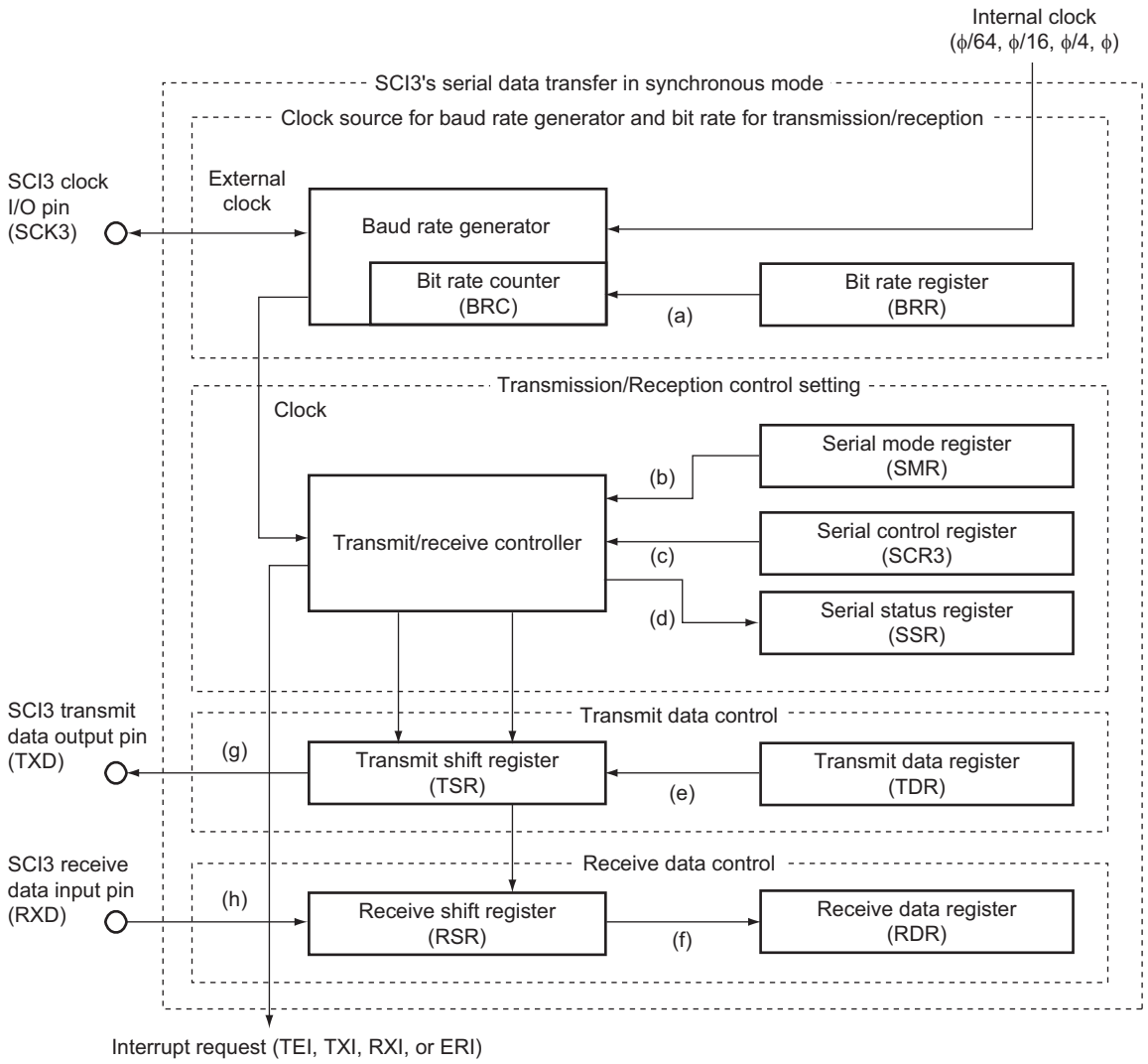


Figure 2 Serial Data with Multiprocessor Bit

Table 1 Multiprocessor Formats

MSR				
CHR	**PE**	**MP**	**STOP**	**Serial Communication Format and Frame Length**
0	*	1	0	1 2 3 4 5 6 7 8 9 10 11 12 S | 8-bit data | MPB | STOP
0	*	1	1	1 2 3 4 5 6 7 8 9 10 11 12 S | 8-bit data | MPB | STOP | STOP
1	*	1	0	1 2 3 4 5 6 7 8 9 10 11 12 S | 7-bit data | MPB | STOP
1	*	1	1	1 2 3 4 5 6 7 8 9 10 11 12 S | 7-bit data | MPB | STOP | STOP

Legend:
*: Don't care
S: Start bit
STOP: Stop bit
MPB: Multiprocessor bit



Notes: (a) Sets bit rate for transmission/reception according to clock source for baud rate generator set in SMR. In this sample task, transmission bit rate is set to 31250 (bit/s).
(b) Sets serial data transfer format and clock source for baud rate generator. In this sample task, communication mode is set to asynchronous mode, data transfer format is set to data length of eight bits, parity bit addition and checking are disabled, multiprocessor communication function is enabled, stop bit length is one bit, and clock source for baud rate generator is set to φ.
(c) Selects transmission/reception and clock output in asynchronous mode, and enables or disables interrupt requests. In this sample task, communication mode is set to asynchronous mode, clock source is set to internal clock, and SCK3 pin is set to function as clock output pin. Interrupts requested by transmit data register empty and receive data register full are disabled.
(d) Indicates the operational status of SCI3 by status flags (transmit data register empty, receive data register full, overrun error, framing error, parity error, and transmit end).
(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) Transmit data.
(h) Receive data.

Figure 3 Multiprocessor Communication Function

- The serial status register (SSR) is an 8-bit register containing status flags that indicate the operational status of SCI3, and multiprocessor bits. SSR can be read from or written to by the CPU at any time. Bits TDRE, RDRF, OER, PER, and FER can only be cleared to 0. However, in order to clear these bits by writing 0, 1 must first be read. Bits TEND and MPBR are read-only bits, and cannot be modified.
- The bit rate register (BRR) is an 8-bit register that designates the transmit/receive bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 of the serial mode register (SMR). BRR can be read from or written to by the CPU at any time.
- Table 2 shows examples of BRR settings in asynchronous mode. The values shown are for active (high-speed) mode and with an OSC of 16 MHz.

Table 2 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate (bit/s)	n	N	Error (%)
110	3	70	0.03
150	2	207	0.16
300	2	103	0.16
600	1	207	0.16
1200	1	103	0.16
2400	0	207	0.16
4800	0	103	0.16
9600	0	51	0.16
19200	0	25	0.16
31250	0	15	0.00
38400	0	12	0.16

Notes: 1. The BRR setting must be such that the error is within 1%.

2. The value set in BRR is given by the following equation:

$$N = \frac{OSC}{64 \times 2^{2n} \times B} \times 10^6 - 1$$

where

B: Bit rate (bit/s)

N: BRR setting ($0 \leq N \leq 255$)

OSC: Value of ϕOSC (MHz) = 16 MHz

n: Value set in bits CKS1 and CKS0 in SMR ($0 \leq n \leq 3$)
(The relation between n and the clock is shown in table 3.)

Table 3 Relation between n and Clock

n	Clock	SMR Setting	
		CKS1	CKS0
0	ϕ	0	0
1	ϕ/4	0	1
2	ϕ/16	1	0
3	ϕ/64	1	1

3. The bit rate error is given by the following equation (rounded off to two decimals):

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

4. When OSC is 16 MHz, the maximum bit rate (in asynchronous mode) is 500000 (bit/s). In this case, n = 0 and N = 0.

- In asynchronous mode, serial communication is performed with synchronization provided character by character. A start bit indicating the start of communication and one or two stop bits indicating the end of communication are added to each character before it is sent.
- SCI3 has separate transmission and reception units, allowing full-duplex communication. As the transmission and reception units are both double-buffered, data can be written during transmission and read during reception, making continuous transmission and reception possible.
- In asynchronous communication, the communication line is normally in the mark state (high level). SCI3 monitors the communication line and when it detects a space (low level), identifies this as a start bit and begins serial data communication.
- One transfer data character consists of a start bit (low level), followed by transmit/receive data (LSB-first format, starting from the least significant bit), a parity bit (high or low level), and finally one or two stop bits (high level).
- In asynchronous mode, synchronization is performed by the falling edge of the start bit during reception. The data is sampled on the 8th pulse of a clock with a frequency 16 times the bit period, so that the transfer data is latched at the center of each bit.
- The SCI3 clock (SCK3) pin is the SCI3 clock I/O pin.
- 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.
- SCI3 can generate six kinds of interrupts: transmit end, transmit data empty, receive data full, and three receive error interrupts (overrun error, framing error, and parity error). These interrupts have the same vector address.
- Each interrupt request can be enabled or disabled by means of bits TIE and RIE in SCR3.
- When bit TDRE is set to 1 in SSR, a TXI interrupt is requested. When bit TEND is set to 1 in SSR, a TEI interrupt is requested. These two interrupts are generated during transmission.
- The initial value of bit TDRE in SSR is 1. Therefore, if the transmit data empty interrupt request (TXI) is enabled by setting bit TIE to 1 in SCR3 before transmit data is transferred to TDR, a TXI interrupt will be requested even if the transmit data is not ready.
- The initial value of bit TEND in SSR is 1. Therefore, if the transmit end interrupt request (TEI) is enabled by setting bit TEIE to 1 in SCR3 before transmit data is transferred to TDR, a TEI interrupt will be requested even if the transmit data has not been sent.
- Effective use of these interrupt requests can be made by having processing that transfers transmit data to TDR carried out in the interrupt service routine. To prevent the generation of these interrupt requests (TXI and TEI), on the other hand, the enable bits for these interrupt requests (bits TIE and TEIE) should be set to 1 after transmit data has been transferred to TDR.
- When bit RDRF is set to 1 in SSR, an RXI interrupt is requested, and if any of bits OER, PER, and FER is set to 1, an ERI interrupt is requested. These two interrupt requests are generated during reception.

2. Table 4 lists the function allocation for this sample task. The functions listed in table 4 are allocated for multiprocessor communication.

Table 4 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
BRR	Sets bit rate of transmission/reception
PMR1	Sets TXD output pin
SCK3	SCI3 clock output pin
TXD	SCI3 transmit data output pin

3. Operational Description

Figure 4 shows this sample task's principle of operation. The hardware and software processing shown in figure 4 performs transmission by the multiprocessor communication function.

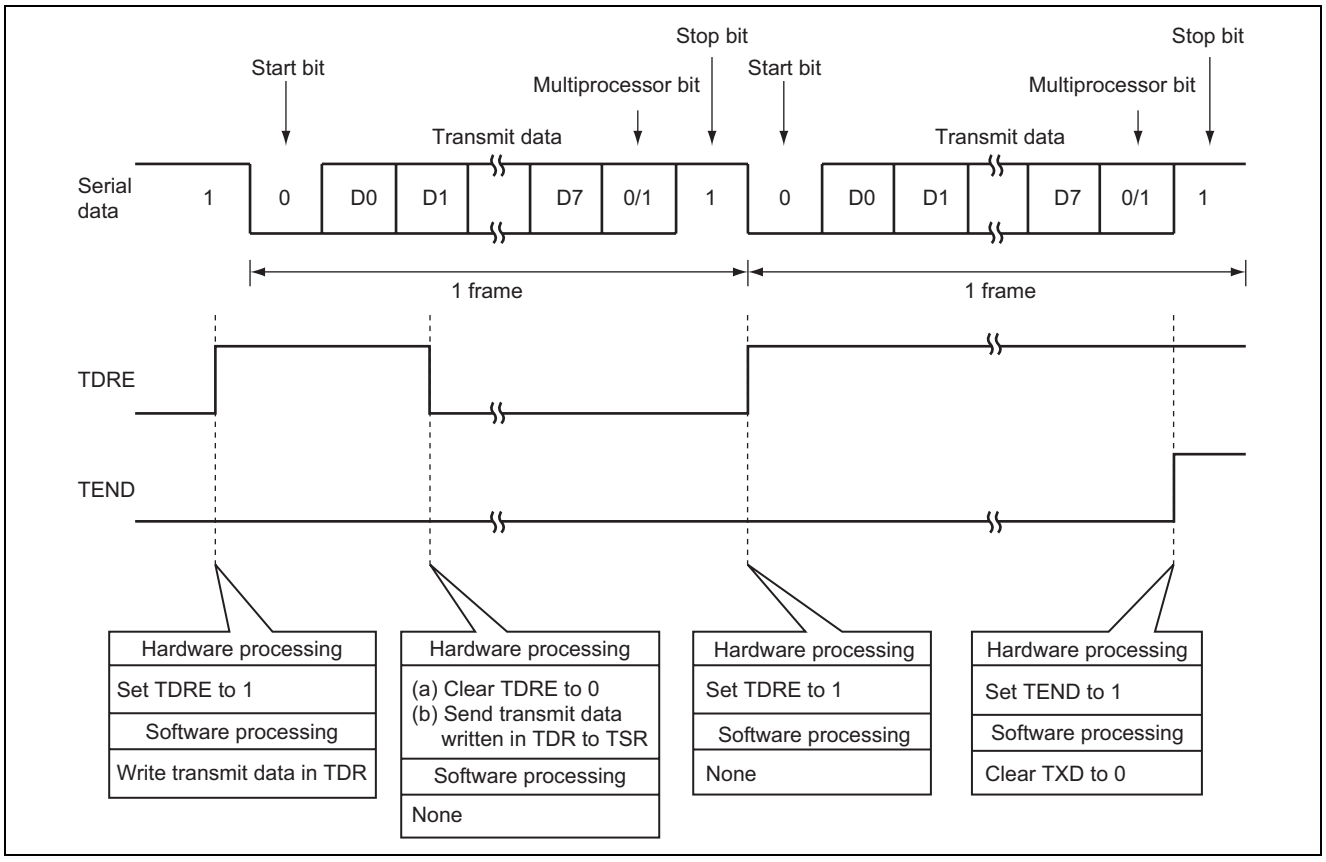


Figure 4 Operation Principle: Transmission by Multiprocessor Communication

4. Description of Software

4.1 Description of Modules

Table 5 describes the software used in this sample task.

Table 5 Description of Module

Module Name	Label Name	Function
Main routine	main	Sets the transmit data, selects multiprocessor communication, and stops SCI3 after four bytes of data have been transmitted.

4.2 Description of Arguments

Table 6 describes the argument used in this sample task.

Table 6 Description of Argument

Argument Name	Function	Used in	Data Length	I/O
STD0 to STD3	Serial transmit data in asynchronous mode	Main routine	1 byte	Output

4.3 Description of Internal Registers

Table 7 describes the internal registers used in this sample task.

Table 7 Description of Internal Registers

Register Name		Functional Description	Address	Setting
SMR	COM	Serial mode register (communication mode): When COM is cleared to 0, the communication mode is set to asynchronous mode.	H'FFA8 Bit 7	0
	CHR	Serial mode register (character length): When CHR is cleared to 0, the data length in asynchronous mode is set to 8 bits.	H'FFA8 Bit 6	0
	PE	Serial mode register (parity enable): When PE is cleared to 0, parity bit addition and checking are disabled at transmission in asynchronous mode.	H'FFA8 Bit 5	0
	STOP	Serial mode register (stop bit length): When STOP is cleared to 0, the stop bit length in asynchronous mode is set to 1 bit.	H'FFA8 Bit 3	0
BRR		Bit rate register: When BRR is set to H'0F, the transmit bit rate that is in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SMR is set to 31250 (bit/s).	H'FFA9	H'0F

Table 7 Description of Internal Registers (cont)

Register Name		Functional Description	Address	Setting
SCR3	TE	Serial control register 3 (transmit enable): When TE is cleared to 0, transmit operation is disabled (TXD pin is the transmit data pin).	H'FFAA Bit 5	0
	CKE1 CKE0	Serial control register 3 (clock enable): When CKE1 is cleared to 0 and CKE0 is set to 1, the clock source is set to an internal clock and the SCK3 pin functions as a clock output pin in asynchronous mode.	H'FFAA Bit 1 Bit 0	CKE1 = 0 CKE0 = 1
TDR		Transmit data register: 8-bit register that stores the transmit data.	H'FFAB	—
SSR	TDRE	Serial status register (transmit data register 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
	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
	MPBR	Serial status register (multiprocessor bit receive): When MPBR is cleared to 0, data in which the multiprocessor bit is 0 has been received. When MPBR is set to 1, data in which the multiprocessor bit is 1 has been received.	H'FFAC Bit 1	0
	MPBT	Serial status register (multiprocessor bit transfer): When MPBT is cleared to 0, the multiprocessor bit value is 0. When MPBT is set to 1, the multiprocessor bit value is 1.	H'FFAC Bit 0	0
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

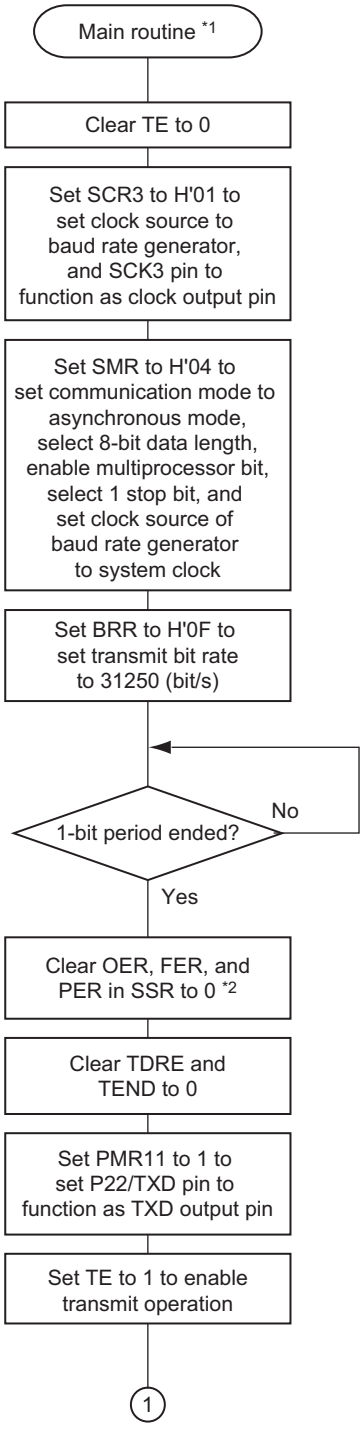
Table 8 describes the RAM used in this sample task.

Table 8 Description of RAM

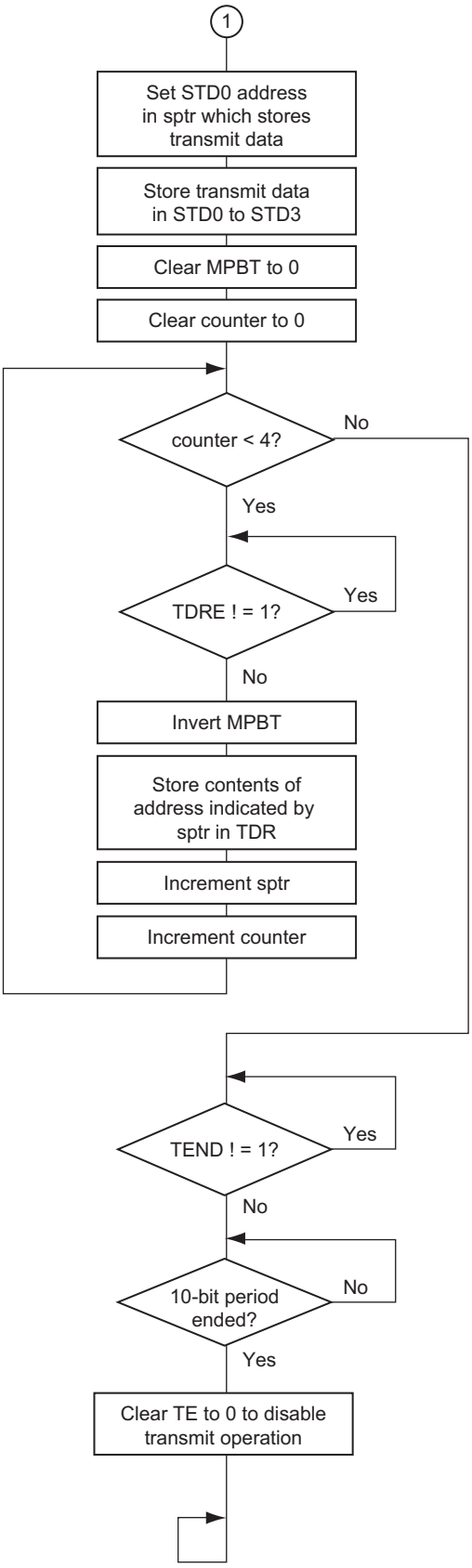
Label Name	Function	Address	Used in
STD0	Stores the first byte of transmit data in serial data transmission in asynchronous mode	H'FB80	Main routine
STD1	Stores the second byte of transmit data in serial data transmission in asynchronous mode	H'FB81	Main routine
STD2	Stores the third byte of transmit data in serial data transmission in asynchronous mode	H'FB82	Main routine
STD3	Stores the fourth byte of transmit data in serial data transmission in asynchronous mode	H'FB83	Main routine
counter	8-bit counter for counting four transmit operations in serial data transmission in asynchronous mode	H'FB84	Main routine

5. Flowchart

1. Main Routine



Notes: 1. In this sample task, the stack pointer is set in INIT.SRC (assembly).
2. In asynchronous mode (multiprocessor communication), OER, FER, and PER must be cleared to 0.


1
Set STD0 address in sptr which stores transmit data
Store transmit data in STD0 to STD3
Clear MPBT to 0
Clear counter to 0
counter < 4?
No
Yes
TDRE ! = 1?
Yes
No
Invert MPBT
Store contents of address indicated by sptr in TDR
Increment sptr
Increment counter
TEND ! = 1?
Yes
No
10-bit period ended?
No
Yes
Clear TE to 0 to disable transmit operation

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                                        */
/*                                                          */
/*  'MultiProcessor Communications'                         */
/*                                                          */
/*  Function                                                */
/*  : Serial Communication Interface                        */
/*     -Multi-Processor Communication                       */
/*                                                          */
/*  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      *(volatile unsigned char *)0xFFA8  /* Serial Mode Register            */
#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 Select 1                  */
#define    CKS0     SMR_BIT.b0                         /* Clock Select 0                  */
#define    BRR      *(volatile unsigned char *)0xFFA9  /* Bit Rate Register               */
#define    SCR3     *(volatile unsigned char *)0xFFAA  /* Serial Control Register 3       */
#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 bit1       */
#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   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   *sptr;

    TE = 0;                                     /* Clear Serial Transmitting                  */

    SCR3 = 0x01;                                /* Initialize Serial Control Register 3       */

    SMR = 0x04;                                 /* Initialize Serial Mode Register            */

    BRR = 0x0F;                                 /* Initialize Bit Rate  Register              */

    for(counter = 0 ; counter < 1 ; counter++){ /* dummy Wait                                 */
        ;
    }

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

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

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

    TE = 1;                                     /* Start Serial Transmitting                  */

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

    STD[0] = 0x01;                              /* Set Serial Transfer Data 0                 */
    STD[1] = 0xB8;                              /* Set Serial Transfer Data 1                 */
    STD[2] = 0x02;                              /* Set Serial Transfer Data 2                 */
    STD[3] = 0xDE;                              /* Set Serial Transfer Data 3                 */

    MPBT = 0;                                   /* Clear Multiprocessor Bit Transfer          */

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

        while(TDRE != 1){                       /* End Serial Transmitting                    */
            ;
        }

        MPBT = ~MPBT;                           /* Initialize Multiprocessor Bit Transfer     */

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

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

    }
```

```
    while(TEND != 1){                           /* End Serial Transmitting          */
        ;
    }

    for(counter = 0 ; counter < 10 ; counter++){    /* dummy Wait                   */
        ;
    }

    TE = 0;                                     /* Initialize Transmitting 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.