

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 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 1<sup>st</sup>, 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

## Serial Data Reception in Asynchronous Mode

---

### Introduction

Four bytes of 8-bit data are received by serial data transfer in asynchronous mode.

### Target Device

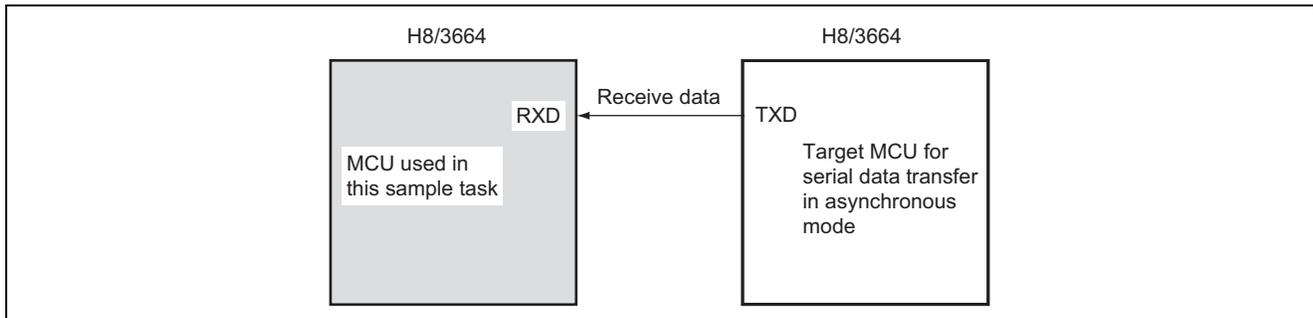
H8/300H Tiny Series H8/3664

### Contents

|                                       |    |
|---------------------------------------|----|
| 1. Specification.....                 | 2  |
| 2. Description of Functions Used..... | 3  |
| 3. Operational Description .....      | 8  |
| 4. Description of Software.....       | 9  |
| 5. Flowchart.....                     | 12 |
| 6. Program Listing.....               | 13 |

### 1. Specification

1. Four bytes of 8-bit data are received by serial data transfer in asynchronous mode, 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). SCI3 stops after four bytes of data have been received.



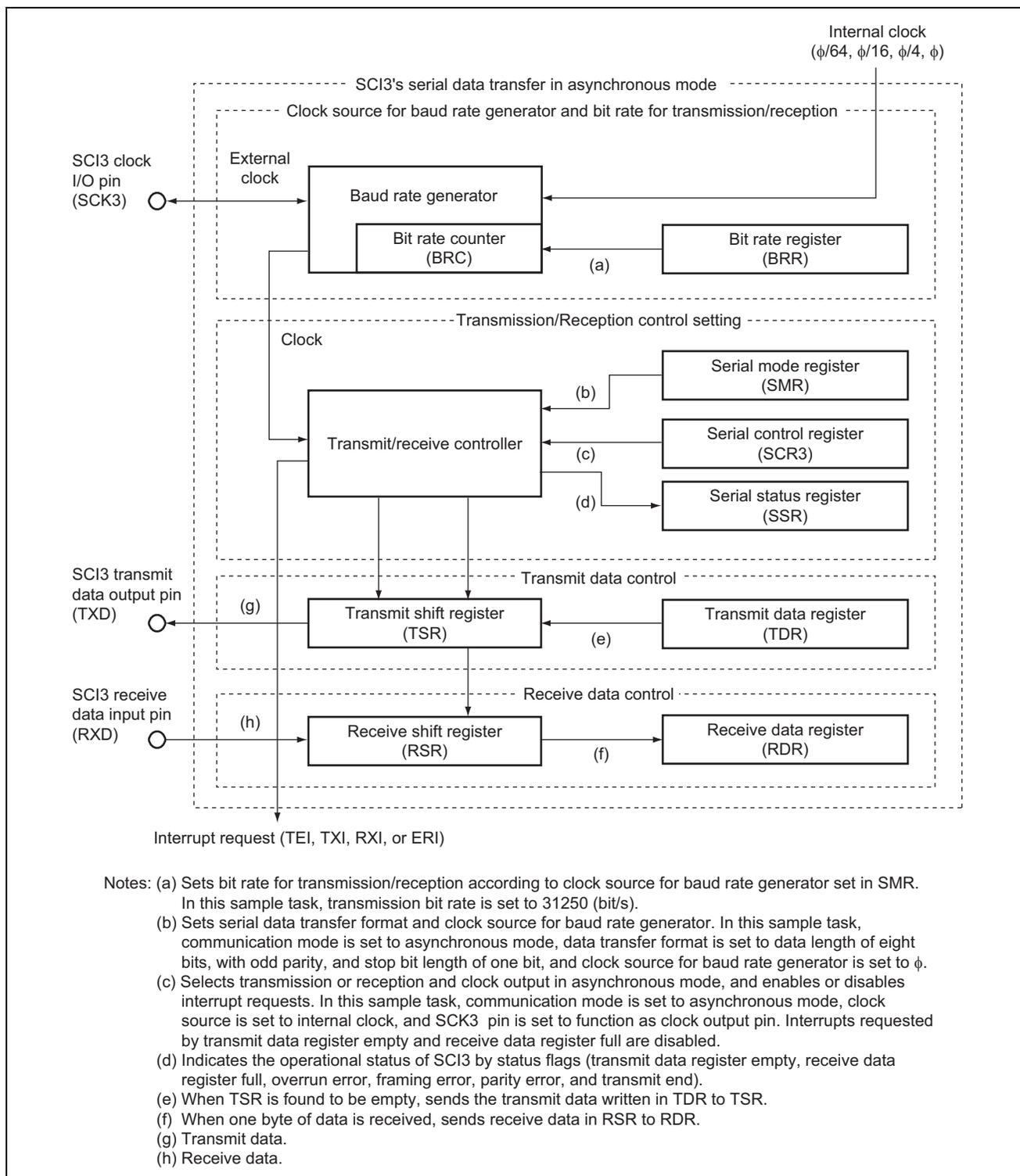
**Figure 1 Serial Data Reception in Asynchronous Mode**

## 2. Description of Functions Used

1. In this sample task, serial data is received in asynchronous mode via the serial communication interface (SCI).

Figure 2 is a block diagram of serial data reception in asynchronous mode. The elements of the block diagram are described below.

- In asynchronous mode, serial data communication is performed asynchronously, with synchronization provided character by character.
- In this mode, serial data can be exchanged with standard asynchronous communication LSIs such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA).
- A multiprocessor communication function is also provided, enabling serial data communication among processors.
- There is a choice of 12 data transfer formats.
- 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 Serial Data Reception in Asynchronous Mode**

- 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 1 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 1 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:

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

where

B: Bit rate (bit/s)

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

OSC: Value of  $\phi\text{OSC}$  (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 2.)

**Table 2 Relation between n and Clock**

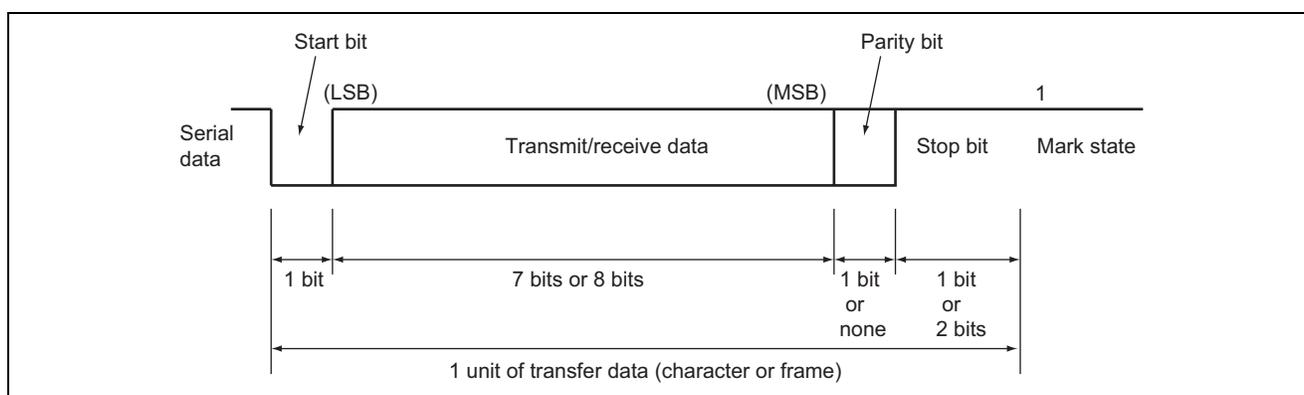
| n | Clock     | SMR Setting |      |
|---|-----------|-------------|------|
|   |           | CKS1        | CKS0 |
| 0 | $\phi$    | 0           | 0    |
| 1 | $\phi/4$  | 0           | 1    |
| 2 | $\phi/16$ | 1           | 0    |
| 3 | $\phi/64$ | 1           | 1    |

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.
- Figure 3 shows the general data transfer format in asynchronous communication. 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.



**Figure 3 Data Format in Asynchronous Communication**

- 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 3 lists the function allocation for this sample task. The functions listed in table 3 are allocated for serial data reception in asynchronous mode.

**Table 3 Function Allocation**

| <b>Function</b> | <b>Function Assignment</b>  |
|-----------------|---|
| RSR             | Receives serial data  |
| RDR             | Stores receive 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   |
| SCK3            | I/O port  |
| RXD             | SCI3 receive data input pin   |

### 3. Operational Description

Figure 4 shows this sample task's principle of operation. The hardware and software processing shown in figure 4 performs serial data reception in asynchronous mode.

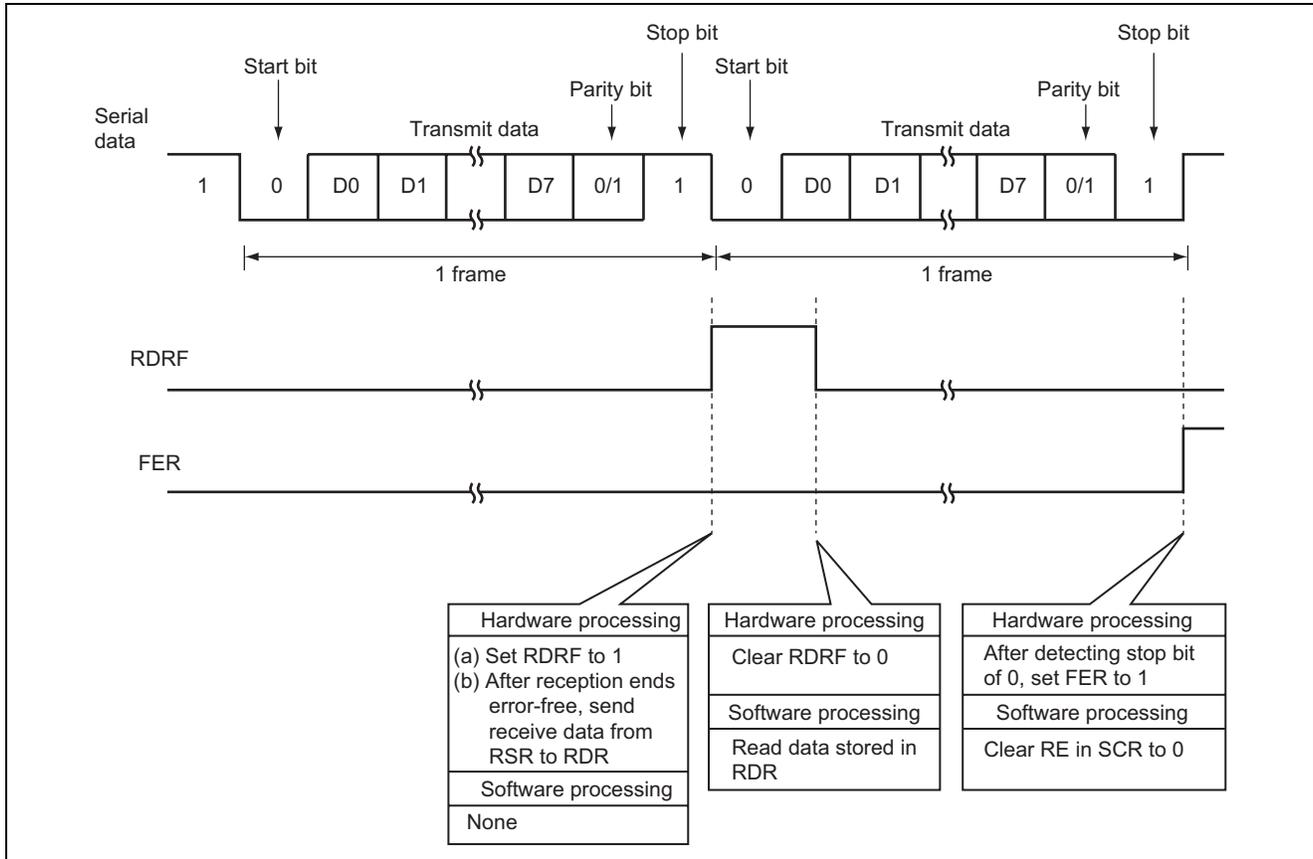


Figure 4 Operation Principle: Serial Data Reception in Asynchronous Mode

## 4. Description of Software

### 4.1 Description of Modules

Table 4 describes the software used in this sample task.

**Table 4 Description of Module**

| Module Name  | Label Name | Function   |
|--------------|------------|--|
| Main routine | main       | Selects serial data reception in asynchronous mode, stores H'FF in SRD0 to SRD3 when a receive error occurs, and stops SCI3 after four bytes of data have been received. |

### 4.2 Description of Arguments

Table 5 describes the argument used in this sample task.

**Table 5 Description of Argument**

| Argument Name | Function                                 | Used in      | Data Length | I/O   |
|---------------|--|--------------|-------------|-------|
| SRD0 to SRD3  | Serial receive data in asynchronous mode | Main routine | 1 byte      | Input |

### 4.3 Description of Internal Registers

Table 6 describes the internal registers used in this sample task.

**Table 6 Description of Internal Registers**

| Register Name | Functional Description   | Address         | Setting  |
|---------------|--|-----------------|----------|
| SMR           | COM<br>Serial mode register (communication mode):<br>When COM is cleared to 0, the communication mode is set to asynchronous mode.                     | H'FFA8<br>Bit 7 | 0        |
|               | CHR<br>Serial mode register (character length):<br>When CHR is cleared to 0, the data length in asynchronous mode is set to 8 bits.                    | H'FFA8<br>Bit 6 | 0        |
|               | PE<br>Serial mode register (parity enable):<br>When PE is set to 1, parity bit addition and checking are enabled at transmission in asynchronous mode. | H'FFA8<br>Bit 5 | 1        |
|               | PM<br>Serial mode register (parity mode):When PM is set to 1, odd parity is to be used for parity addition and checking.                               | H'FFA8<br>Bit 4 | 1        |
|               | STOP<br>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<br>Bit 3 | 0        |
|               | MP<br>Serial mode register (multiprocessor mode):<br>When MP is cleared to 0, the multiprocessor communication function is disabled.                   | H'FFA8<br>Bit 2 | 0        |
|               | CKS1<br>Serial mode register (clock select 1 and 0):   | H'FFA8          | CKS1 = 0 |
|               | CKS0<br>When CKS1 and CKS0 are both cleared to 0, the clock source for the baud rate generator is set to the system clock.                             | Bit 1<br>Bit 0  | CKS0 = 0 |

**Table 6 Description of Internal Registers (cont)**

| Register Name | Functional Description   | Address  | Setting         |          |
|---------------|--|--|-----------------|----------|
| BRR           | Bit rate register:<br>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            |          |
| SCR3          | RE   | Serial control register 3 (receive enable):<br>When RE is cleared to 0, receive operation is disabled.<br>When RE is set to 1, receive operation is enabled.                           | H'FFAA<br>Bit 4 | 0        |
|               | CKE1   | Serial control register 3 (clock enable):  | H'FFAA          | CKE1 = 0 |
|               | CKE0   | When CKE1 and CKE0 are both cleared to 0, the clock source is set to an internal clock and the SCK3 pin functions as an I/O port in asynchronous mode.                                 | Bit 1<br>Bit 0  | CKE0 = 0 |
| RDR           | Receive data register:<br>8-bit register that stores the receive data.   | H'FFAD   | —               |          |
| SSR           | RDRF   | Serial status register (receive data register full):<br>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<br>Bit 6 | 1        |
|               | OER  | Serial status register (overflow error):<br>When OER is cleared to 0, reception is in progress or completed.<br>When OER is set to 1, an overflow error has occurred during reception. | H'FFAC<br>Bit 5 | 0        |
|               | FER  | Serial status register (framing error):<br>When FER is cleared to 0, reception is in progress or completed.<br>When FER is set to 1, a framing error has occurred during reception.    | H'FFAC<br>Bit 4 | 0        |
|               | PER  | Serial status register (parity error):<br>When PER is cleared to 0, reception is in progress or completed.<br>When PER is set to 1, a parity error has occurred during reception.      | H'FFAC<br>Bit 3 | 0        |

#### 4.4 Description of RAM

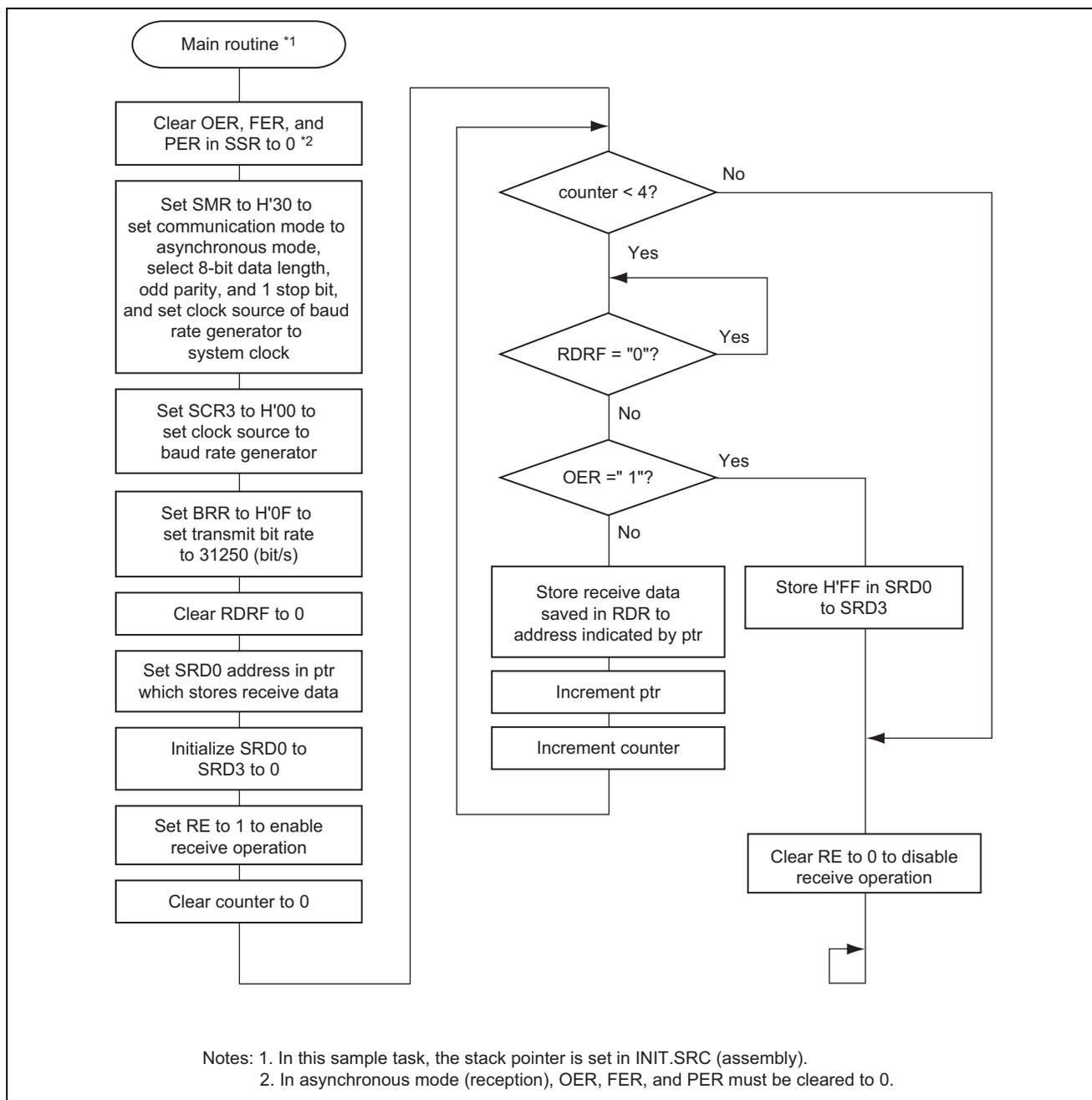
Table 7 describes the RAM used in this sample task.

**Table 7 Description of RAM**

| <b>Label Name</b> | <b>Function</b>  | <b>Address</b> | <b>Used in</b> |
|-------------------|--|----------------|----------------|
| SRD0              | Receives the first byte of receive data in serial data reception in asynchronous mode            | H'FB80         | Main routine   |
| SRD1              | Receives the second byte of receive data in serial data reception in asynchronous mode           | H'FB81         | Main routine   |
| SRD2              | Receives the third byte of receive data in serial data reception in asynchronous mode            | H'FB82         | Main routine   |
| SRD3              | Receives the fourth byte of receive data in serial data reception in asynchronous mode           | H'FB83         | Main routine   |
| counter           | 8-bit counter for counting four receive operations in serial data reception in asynchronous mode | H'FB84         | Main routine   |

### 5. Flowchart

#### 1. Main Routine



#### 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                    */
/*                                     */
/* 'Asynchronous Serial Data Reception' */
/*                                     */
/* Function                            */
/* : Serial Communication Interface     */
/*   Asynchronous Serial Interface     */
/*   -Receiving                        */
/*                                     */
/* External Clock : 16MHz               */
/* Internal Clock : 16MHz               */
/* 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    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                      /* Multiprocessor 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 Error               */
#define    FER      SSR_BIT.b4                       /* Framing Error               */
#define    PER      SSR_BIT.b3                       /* Parity Error                */
#define    TEND     SSR_BIT.b2                       /* Transmit End                 */
#define    MPBR     SSR_BIT.b1                       /* Multiprocessor Bit Receive  */
#define    MPBT     SSR_BIT.b0                       /* Multiprocessor Bit Transfer */
#define    RDR      *(volatile unsigned char *)0xFFAD /* Receive data Register       */

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

/*****
/*   RAM Allocation
*****/
    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    *ptr;

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

    SMR = 0x30;        /* Initialize Serial Mode Register */

    SCR3 = 0x00;       /* Initialize Serial Control Register 3 */

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

    RDRF = 0;          /* Clear RDRF        */

    ptr = &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 */

    RE = 1;            /* Start Serial Receiving */

    counter = 0;       /* Clear counter      */

    while (counter < 4){ /* Serial Receiving Data Counter 4 Loop */

        while(RDRF == 0){ /* End Serial Receiving */
            ;
        }
    }
}

```

```
if (OER == 1){                                /* Overrun Errorr Flag = 1 ?          */
    SRD[0] = 0xFF;                            /* Overrun Errorr 0                  */
    SRD[1] = 0xFF;                            /* Overrun Errorr 1                  */
    SRD[2] = 0xFF;                            /* Overrun Errorr 2                  */
    SRD[3] = 0xFF;                            /* Overrun Errorr 3                  */
    break;
}
else {
    *ptr = RDR;                               /* Save Serial Receiving Data        */
    ptr++;                                    /* Increment Serial Receiving Data Address */
    counter++;                               /* Increment counter                  */
}
}

RE = 0;                                       /* Initialize Receiving Enable        */

while(1){
    ;
}
}
```

### Revision Record

| Rev. | Date      | Description |                       |
|------|-----------|-------------|-----------------------|
|      |           | 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.