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.

SH7145 Group

SCI Break Detection

Introduction

This application note discusses break detection during asynchronous communication using the SH7145F's SCI (Serial Communication Interface) module.

Target Device

SH7145F

Contents

1. Specifications

As shown in figure 1, break detection is performed using channel 0 (ch0) of the SH7145F's SCI. Break detection is only possible in asynchronous communication. In the SH7145, break detection is not performed by hardware, and so is executed by software.

Break detection is performed by monitoring the level of the RxD pin when a framing error occurs: if the RxD pin is at low level, it is regarded that a break has occurred. In this sample task, break detection is performed after receiving three bytes of data. Confirmation of the RxD pin level at the time of a framing error is performed three times at 5 msec intervals, and a break is detected if the level is low all three times. When a break is detected, the RE bit of SCR_0 is cleared to 0 to terminate the reception by the SCI. The RxD pin state is checked using a compare-match timer interrupt.

The SCI communication format is 19,200 bps, 8 bits, one stop bit, and no parity.

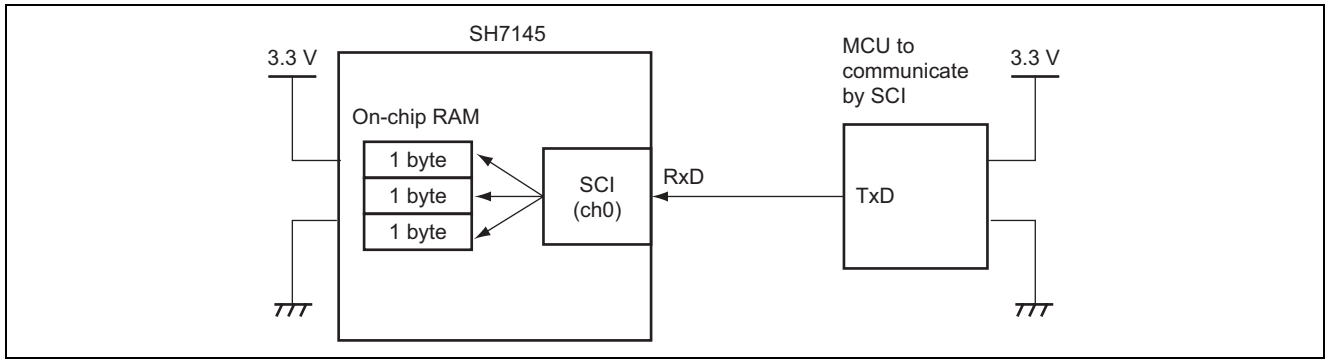


Figure 1 SH7145 SCI Reception Connection Diagram

Table 1 Asynchronous Serial Reception Format

Item	Setting
Bit rate	19200 bps
Data length	8 bits
Parity bit	none
Stop bit	1 bit
Serial/parallel conversion format	LSB first

2. Description of Functions

In this sample task, the SCI (Serial Communication Interface) and CMT (Compare-Match Timer) are used.

2.1 Serial Communication Interface (SCI)

In this sample task, the SCI is used to perform asynchronous serial data communication. Figure 2 is the block diagram of SCI module channel 0 (ch0); below, functions are explained referring to figure 2.

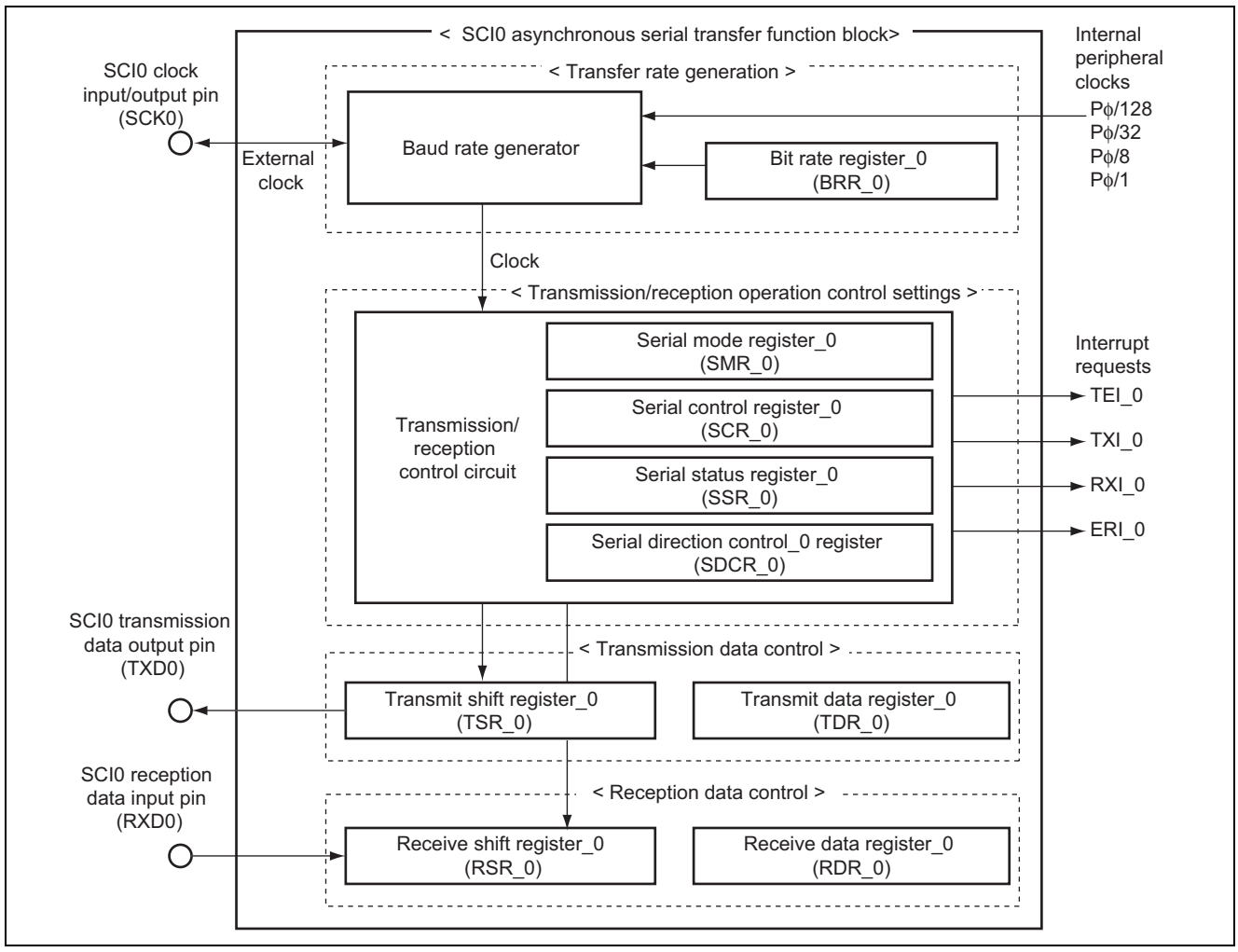


Figure 2 Block Diagram of SCI (ch0)

- In asynchronous mode, serial data communication is performed with synchronization in character units, allowing serial communication with a dedicated asynchronous communication LSI conforming to the Universal Asynchronous Receiver/Transmitter (UART), Asynchronous Communication Interface Adapter (ACIA) or other standard. Further, in asynchronous mode, a function (multiprocessor communication function) is provided for serial communications with multiple processors.
- The internal peripheral clock Pϕ is a reference clock used to drive the on-chip peripheral functions, and is generated by the clock pulse generator.
- The receive shift register (RSR_0) is a register used to receive serial data. Serial data is input from the RxD0 pin, and when data for one frame has been received, the data in RSR_0 is automatically transferred to the receive data register (RDR_0). RSR_0 cannot be accessed from the CPU.
- The receive data register (RDR_0) is an 8-bit register used to store received data. Upon receiving data for one frame, the data is automatically transferred from RSR_0. RSR_0 and RDR_0 have a double-buffered structure, so that continuous receive operation is possible. Since RDR_0 is a receive-only register, only reading is possible from the CPU.
- The transmit shift register (TSR_0) is a register used to transmit serial data. During transmission, data is transmitted from the transmit data register (TDR_0) to TSR_0, and the transmission data is output from the TxD0 pin. TSR_0 cannot be directly accessed from the CPU.
- The transmit data register (TDR_0) is an 8-bit register used to store data for transmission. When TSR_0 is detected to be empty, data written to TDR_0 is automatically transferred to TSR_0. TDR_0 and TSR_0 have a double-buffered structure, so that when data for one frame has been transmitted and the next data is written to TDR_0, the data is transferred to TSR_0. Continuous transmission is thus possible. TDR can always be read and written by the CPU, but writing should be performed after confirming that the TDRE bit of the serial status register (SSR_0) is 1.
- The serial mode register (SMR_0) is an 8-bit register used to select the serial data communication format and the clock source for the internal baud rate generator.
- The serial control register (SCR_0) is a register used to control transmission/reception and interrupts and select the transmission/reception clock source.
- The serial status register (SSR_0) consists of SCI0 status flags and transmission/reception multiprocessor bits. TDRE, RDRF, ORER, PER, and FER can only be cleared.
- The serial direction control register (SDCR_0) is used to select LSB-first or MSB-first. In 8-bit length communications, either LSB-first or MSB-first can be selected, but in 7-bit communications, LSB-first should be selected.
- The bit rate register (BRR_0) is an 8-bit register used to adjust the bit rate. In the SCI, a baud rate generator is provided independently for each channel, so that different bit rates can be set. For the relationship between the setting values and the execution rate and other details, please refer to the hardware manual.

Table 2 shows the assignment of functions in this sample task.

Table 2 Assignment of Functions

Element	Classification	Description
TXD0	Pin	Channel 0 transmission data output pin
RXD0	Pin	Channel 0 reception data input pin
SMR_0	SCI0	Sets communication format to asynchronous mode.
SCR_0	SCI0	Enables reception.
SSR_0	SCI0	Status flags indicating the operation state of SCI0
SDCR_0	SCI0	Set to select LSB-first.
BRR_0	SCI0	Sets the communication bit rate.
TSR_0	SCI0	Register for serial data transmission
TDR_0	SCI0	Register for storing data for transmission
RSR_0	SCI0	Register for receiving serial data
RDR_0	SCI0	Register for storing received data

2.2 Compare-Match Timer (CMT)

The CMT generates an interrupt at given intervals. Figure 3 is a block diagram of the CMT module channel 0 (ch0); below, functions are explained referring to figure 3.

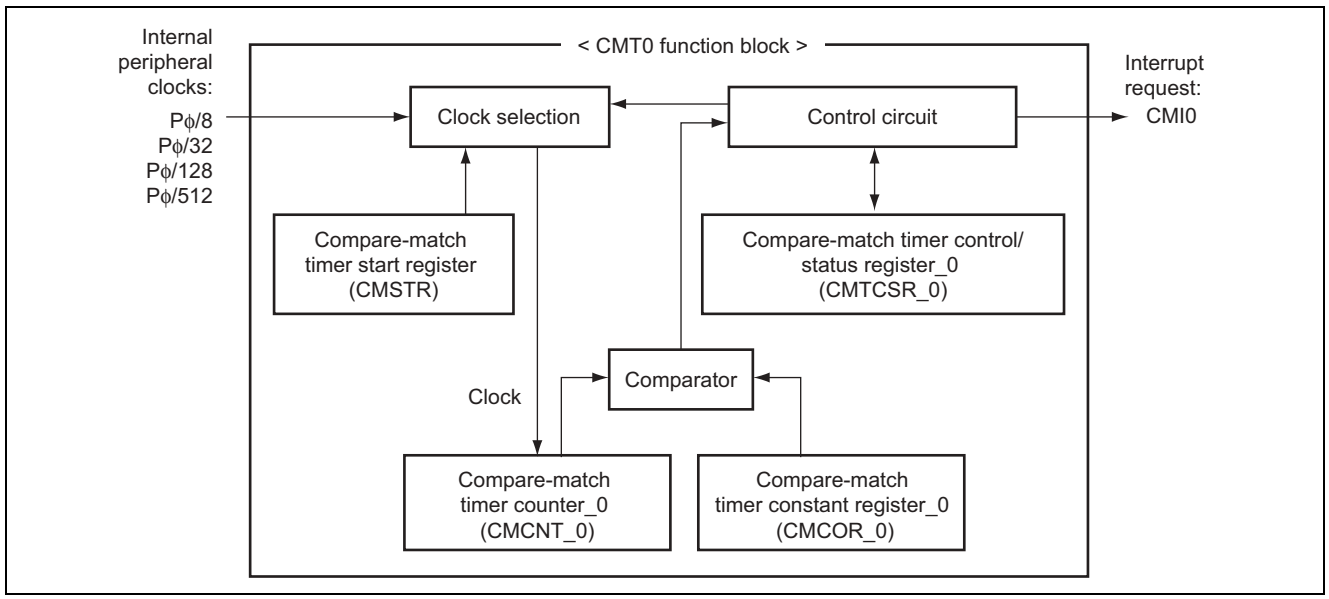


Figure 3 CMT (ch0) Block Diagram

- The CMT has a 16-bit counter, and can generate an interrupt at given intervals.
- A clock signal obtained by dividing the internal peripheral clock Pϕ can be selected. The counter is incremeted by the selected clock.
- The compare-match timer start register (CMSTR) starts or stops counting.
- The compare-match timer control/status register (CMCSR_0) indicates compare-match occurrence, sets up interrupts, and selects the count-up clock.
- The compare-match timer counter (CMCNT_0) is an up-counter used to generate interrupt requests.
- The compare-match timer constant register (CMCOR_0) sets the compare-match interval.

3. Description of Operation

Figure 4 shows the operation during reception in asynchronous mode in this sample task. As an explanation of figure 4, table 3 describes the software and hardware processing.

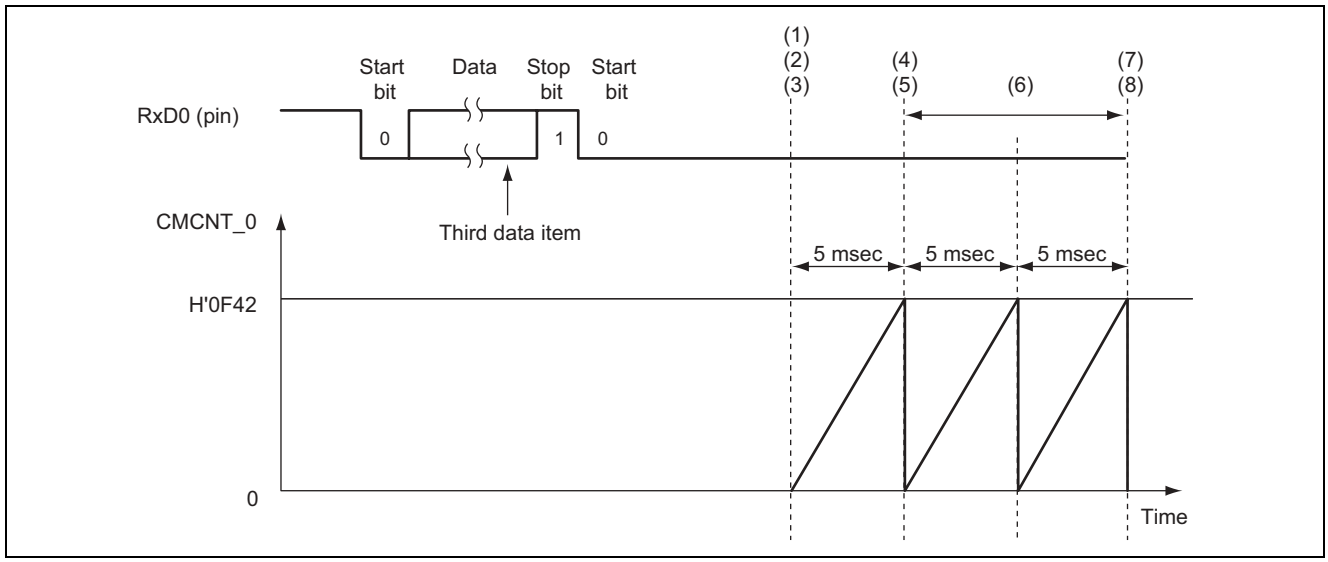


Figure 4 Operation during Data Reception

Table 3 Description of Processing

	Software Processing	Hardware Processing
(1)	Framing error processing	Set the FER bit in SSR_0 to 1.
(2)	Set the STR0 bit in CMSTR to 1.	Start counting by CMT0.
(3)	—	Set the CMF flag (generate a compare-match interrupt).
(4)	Clear the CMF flag to 0.	Start counting by CMT0.
(5)	Read RxD pin level.	—
(6)	Repeat steps (3) through (5) twice.	Repeat steps (3) through (5) twice.
(7)	Clear the STR0 bit in CMSTR to 0.	Stop operation of CMT0.
(8)	Clear the RE bit in SCR0 to 0.	Stop reception by SCI0.

4. Description of Software

4.1 Modules

Table 4 describes the modules of this sample task.

Table 4 Description of Modules

Module Name	Label Name	Functions
Main routine	main	Calls various modules.
SCI routine	init_sci	Initializes SCI0 and CMT0.
Receive routine	rcv_sci	Receives serial data.
Error processing routine	err_int	Performs reception error processing.
CMT0 interrupt routine	cmt_int	Reads RxD pin level at 5 msec intervals.

4.2 Internal Registers

Tables 5 through 7 describe the internal registers used in this sample task. The settings are the values used in this sample task and are different from their initial values.

Table 5 Description of Internal Registers (1)

Register Name	Bit	Bit Name	Setting	Function
MSTCR1				Module standby control register 1
	0	MSTP16	0	SCI0 standby control bit When MSTP16 = 0, cancels the standby state.
MSTCR2				Module standby control register 2
	12	MSTP12	0	CMT standby control bit When MSTP12 = 0, cancels the standby state of the CMT.
SCR_0			H'10	Serial control register_0 Controls transmission/reception and interrupts and selects transmit/receive clock source.
	7	TIE	0	Transmit interrupt enable Set to 1 to enable TXI interrupt requests.
	6	RIE	0	Receive interrupt enable Set to 1 to enable RXI and ERI interrupt requests.
	5	TE	0	Transmit enable Set to 1 to enable transmission.
	4	RE	1	Receive enable Set to 1 to enable reception.
	3	MPIE	0	Multiprocessor interrupt enable (valid in asynchronous mode when MP = 1 in SMR) In this sample task, this bit is invalid because of MP = 0.
	2	TEIE	0	Transmit end interrupt enable Set to 1 to enable TEI interrupt requests.
	1 0	CKE1 CKE0	0 0	Clock enable 1, 0 These bits select the clock source and SCK pin function. In this sample task, the clock source is an internal clock and the SCK pin is not used.

Table 6 Description of Internal Registers (2)

Register Name	Bit	Bit Name	Setting	Function
SMR_0			H'00	Serial mode register_0 Selects communication format and clock source for the internal baud rate generator.
	7	C/$\overline{A}$	0	Communication mode When C/$\overline{A}$ = 0, SCI0 operates in asynchronous mode.
	6	CHR	0	Character length (only valid in asynchronous mode) When CHR = 0, communication data length is 8 bits.
	5	PE	0	Parity enable (only valid in asynchronous mode) When PE= 0, communication is performed with no parity.
	4	O/$\overline{E}$	0	Parity mode (valid when PE = 1 in asynchronous mode) In this sample task, PE = 0 and so this bit is invalid.
	3	STOP	0	Stop bit length (only valid in asynchronous mode) When STOP = 0, one stop bit is used.
	2	MP	0	Multiprocessor mode (only valid in asynchronous mode) When MP = 0, multiprocessor communication function is disabled.
	1	CKS1	0	Clock select 1, 0
	0	CKS0	0	When CKS1 = CKS0 = 0, the clock source for the internal baud rate generator is set to Pϕ.
BRR_0			H'40	Bit rate register_0 An 8-bit register used to adjust the bit rate.
SDCR_0			H'F2	Serial direction control register_0 The DIR bit (bit 3) is used to select LSB/MSB-first; in this sample task, set to DIR = 0 (LSB-first).
SSR_0			H'xx	Serial status register_0 Consists of SCI0 status flags and communication multiprocessor bits; only 0 can be written to the status flags for flag clearing.
	7	TDRE	*	Transmit data register empty (status flag)
	6	RDRF	*	Receive data register full (status flag)
	5	ORER	*	Overrun error (status flag)
	4	FER	*	Framing error (status flag)
	3	PER	*	Parity error (status flag)
	2	TEND	*	Transmit end (status flag)
	1	MPB	0	Multiprocessor bit
	0	MPBT	0	Multiprocessor bit transfer
PACRL2				Port A control register L2
	0	PA0MD	1	PA0 mode bit Specifies the function of PA0, a multiplexed pin of Port A (RxD0).

Note: * Only clearing is possible; setting to 1 is done by hardware.

Table 7 Description of Internal Registers (3)

Register Name	Bit	Bit Name	Setting	Function
MSTCR2				Module standby control register 2
	12	MSTP12	0	CMT standby control bit When MSTP12 = 0, cancels the standby state of the CMT.
CMSTR			H'01	Compare-match timer start register
	15 to 2	—	0	Reserved
	1	STR1	0	Count start 1 When STR1 = 0, the CMCNT_1 stops counting.
	0	STR0	1	Count start 0 When STR0 = 1, the CMCNT_0 starts counting.
CMCSR_0				Compare-match timer control/status register_0
	15 to 8	—	0	Reserved
	7	CMF	*	Compare-match flag When CMF=1, CMCNT and CMCOR values match.
	6	CMIE	1	Compare-match interrupt enable Enables or disables compare-match interrupts. When CMIE=1, compare-match interrupts are enabled.
	5 to 2	—	0	Reserved
	1	CKS1	0	CMCNT_0 input clock select 1, 0
	0	CKS0	1	In this sample task, Pϕ/32 is selected.
CMCNT_0			—	Compare-match timer counter_0 Up-counter used to generate interrupt requests.
CMCOR_0			H'0F42	Compare-match timer constant register_0 Specify the interval of compare-match with CMCNT_0.
IPRG			H'00F0	Interrupt priority register G Sets priority levels for interrupt sources.
	7 to 4	IPR7	1	These bits set the priority level of CMT0 (0-15).
		IPR6	1	
		IPR5	1	
		IPR4	1	

Note: * Only clearing is possible; setting to 1 is done by hardware.

4.3 Description of Arguments

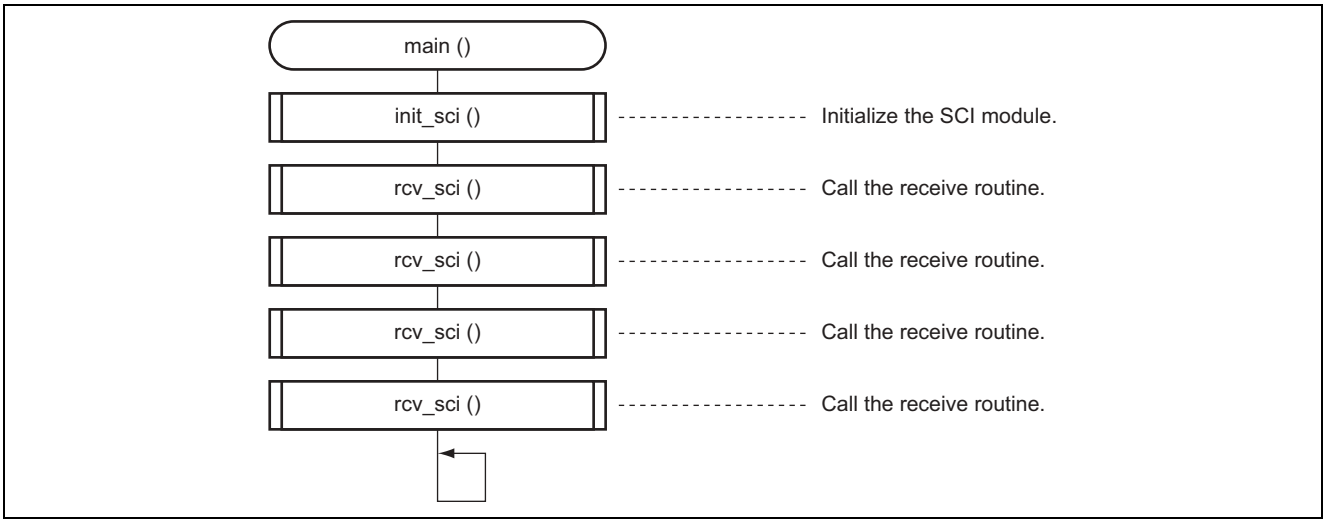
Arguments used in this sample task are described in table 8.

Table 8 Description of Arguments

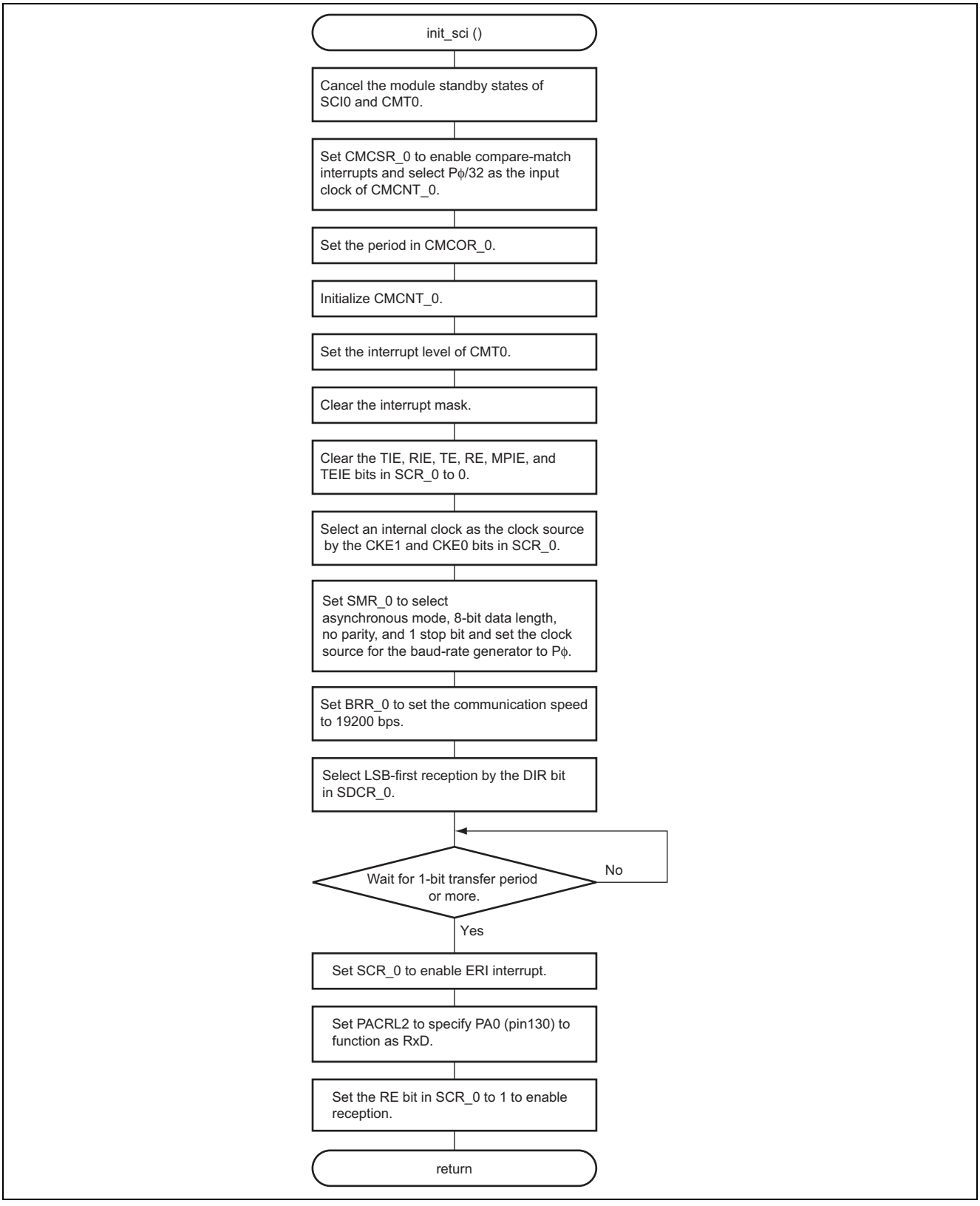
Argument Name	Function	Used in
Rev_data[0 to 2]	Stores data received by SCI0	Receive routine
T_count	Counts the number of CMT0 interrupts	CMT0 interrupt routine, error processing routine
Rd_count	Number of times level is low when RxD pin level is read	CMT0 interrupt routine, error processing routine

5. Flowchart

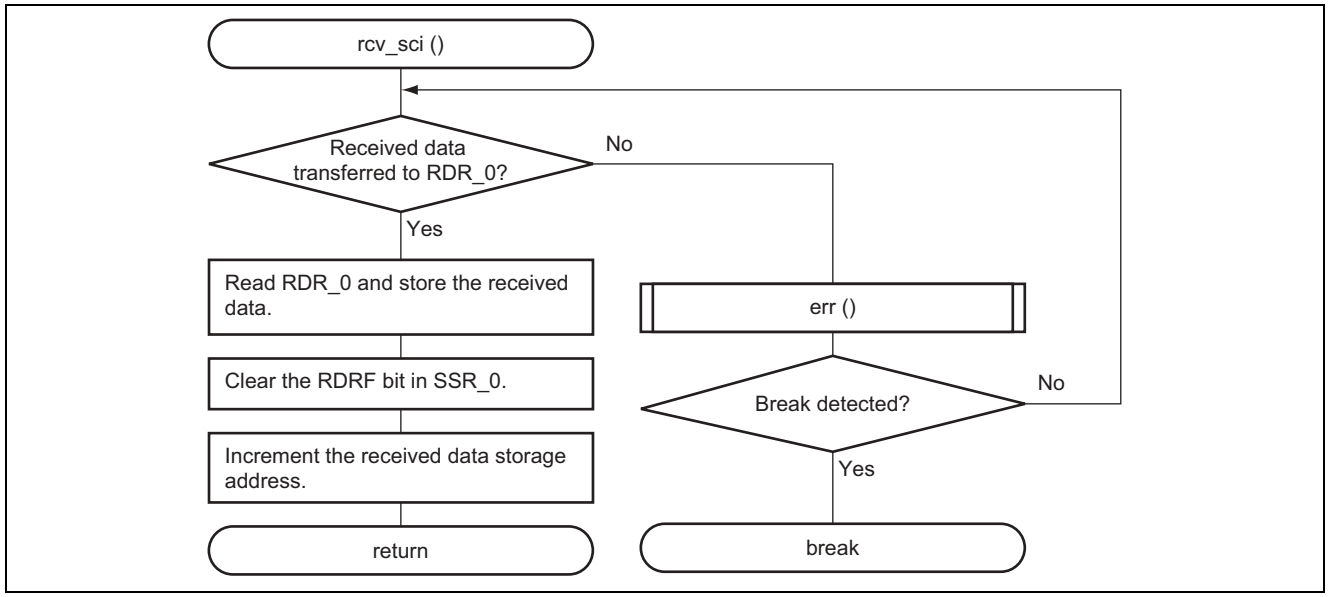
5.1 Main Routine



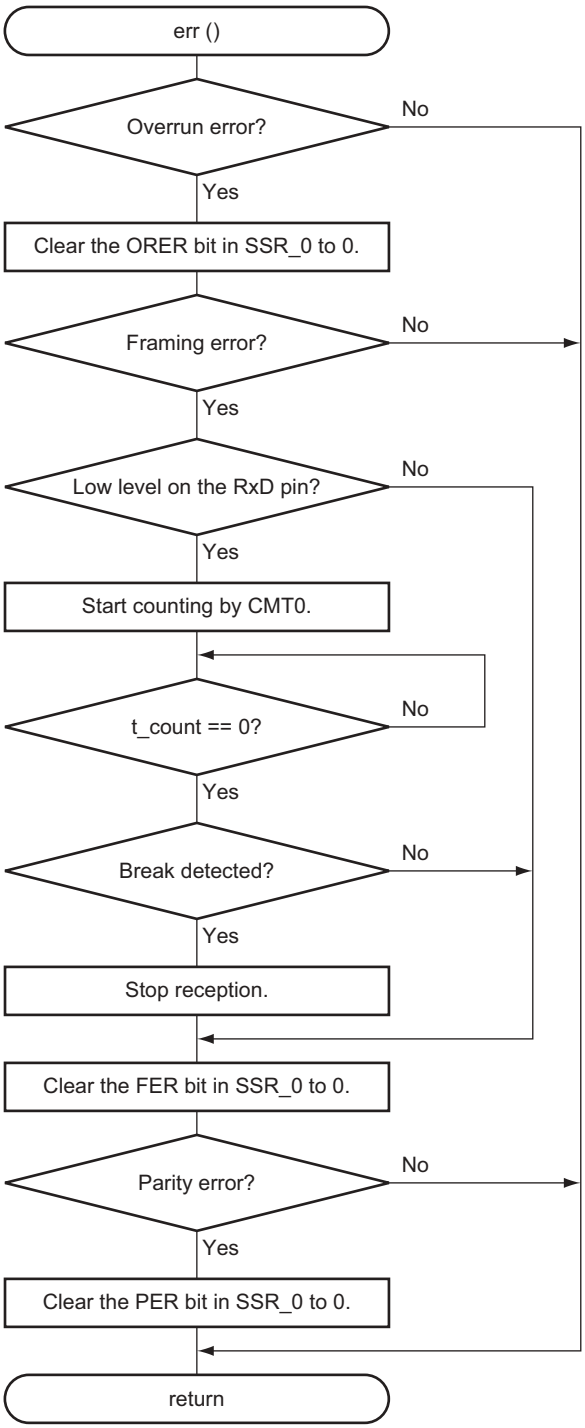
5.2 SCI Routine



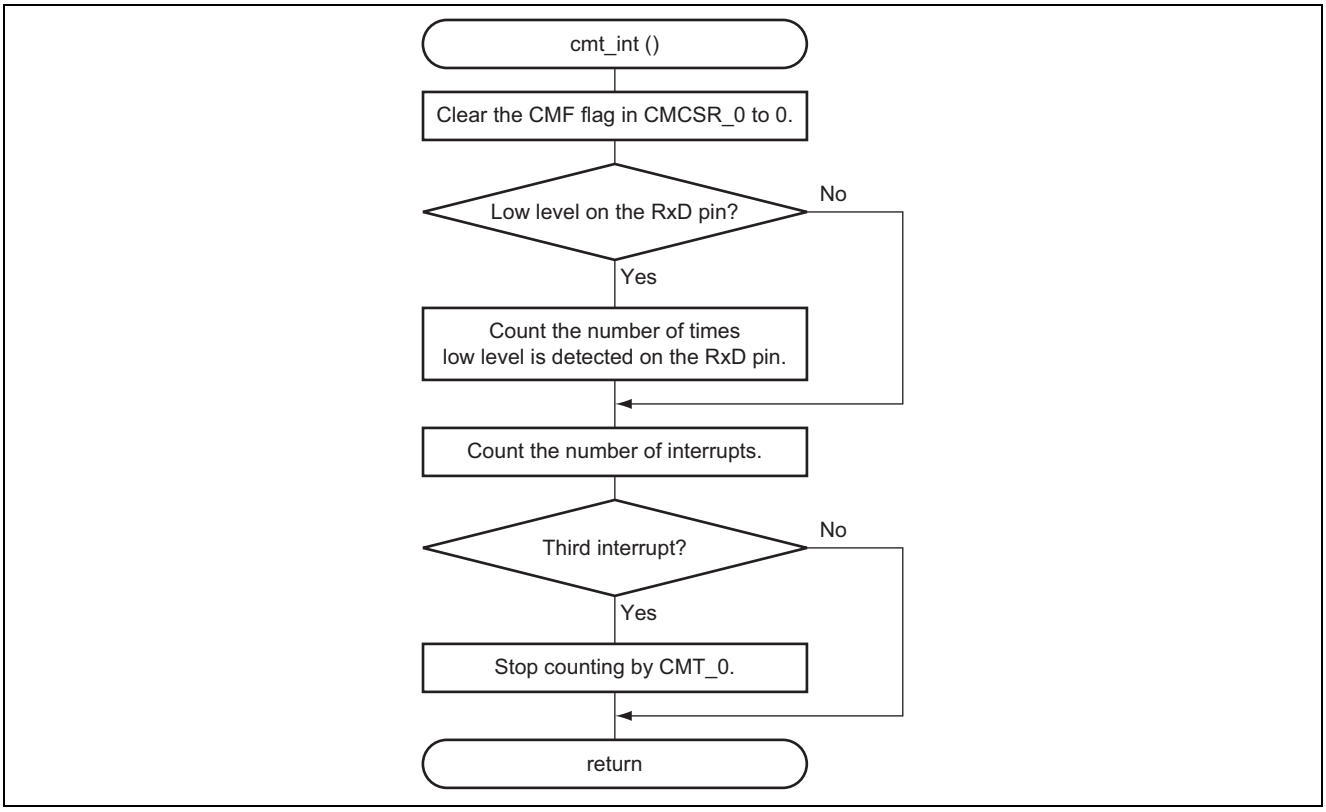
5.3 Receive Routine



5.4 Error Processing Routine



5.5 CMT0 Interrupt Routine



6. Program Listing

```
/*****************************************************************************************************/
/* SH7145F    Application Note                                                                       */
/*                                                                                                   */
/* Function                                                                                          */
/*    :SCI0                                                                                          */
/*    :Asynchronous Receive Mode(break)                                                              */
/*                                                                                                   */
/* External input clock         :12.5MHz                                                             */
/* Internal CPU clock           :50MHz                                                               */
/* Internal peripheral clock    :25MHz                                                               */
/*                                                                                                   */
/* Written        :2003/12      Rev.1.0                                                              */
/*****************************************************************************************************/

#include "iodefine.h"
#include <machine.h>

/*****************************************************************************************************/
/*  Symbol Definition                                                                                */
/*****************************************************************************************************/
#define COUNT 3

/*****************************************************************************************************/
/*  Function Define                                                                                  */
/*****************************************************************************************************/
void main(void);

void init_sci(void);
unsigned char rcv_sci(unsigned char);

void err(void);
void cmt_int(void);
void dummy_f(void);

/*****************************************************************************************************/
/*  RAM Allocation Definition                                                                        */
/*****************************************************************************************************/
unsigned char T_count;                          /* Timer interrupt count                    */
unsigned char Rd_Count;                         /* RxD low level count                      */

volatile unsigned char Rev_data[COUNT];         /* Receive data                             */
```

```
/************************************************************************************************/
/*  Main Program                                                                                */
/************************************************************************************************/
void main( void )
{
    unsigned char i = 0;

    init_sci();                                 /* SCI initialize routine                   */

    i = rcv_sci(i);
    i = rcv_sci(i);
    i = rcv_sci(i);
    i = rcv_sci(i);

    while(1);
}

/************************************************************************************************/
/*  Function   : init_sci                                                                       */
/*  Operation  : Initialize SCI0                                                                */
/*  Asynchronous Receive Mode                                                                   */
/*      -data length  : 8bit                                                                    */
/*      -stop bit     : 1bit                                                                    */
/*      -parity bit   : Non-parity bit                                                          */
/************************************************************************************************/
void init_sci(void)
{
    unsigned long i;

    T_count = 0;
    Rd_count = 0;

    P_STBY.MSTCR1.BIT.MSTP16 = 0;               /* Disable SCI0 standby mode                */
    P_STBY.MSTCR2.BIT.MSTP12 = 0;               /* Disable CMT standby mode                 */

    P_CMT.CMCSR_0.WORD = 0x0041;                /* Initialize CMCSR_0                       */
            // [15-8]  = 0
            // [7]CMF  = 0
            // [6]CMIE = 1    CMT0 interrupt enable
            // [5-2]   = 0
            // [1]CKS1 = 0
            // [0]CKS0 = 1    count clock = P phi/32
    P_CMT.CMCOR_0 = 0x0F42;                     /* Set CMCOR_0 (5msec)                      */
    P_CMT.CMCNT_0 = 0;                          /* Initialize CMCNT_0                       */
    P_INTC.IPRG.BIT.CMT0 = 0xF;                 /* Set CMT0 interrupt level                 */
    set_imask(0);                               /* Clear interrupt mask level               */
```

```
    /* Initialize SCI asynchronous mode */
    P_SCI0.SCR_0.BYTE &= 0x03 ;                    /* Clear TIE,RIE,TE,RE,MPIE,TEIE bit        */
    P_SCI0.SCR_0.BIT.CKE = 0;                      /* Clock:internal,SCK:output                */
    P_SCI0.SMR_0.BYTE = 0x00;                      /* 8bit,No parity,1stop bit                 */
            // CA     = 0;                        /* Asynchronous mode                        */
            // CHR    = 0;                        /* Data length 8 bits                       */
            // PE     = 0;                        /* Non-parity                               */
            // OE     = 0;                        /* (=0)even parity                          */
            // STOP   = 0;                        /* 1 stop bit                               */
            // CKS    = 0;                        /* Clock source = P phi(25MHz)              */
    P_SCI0.BRR_0 = 40;                             /* 19200bps@25MHz                           */
    P_SCI0.SDCR_0.BIT.DIR = 0;                     /* LSB first                                */

    for( i=0; i < 0x0400 ; i++);                   /* Wait 1 bit                               */

    /* Initialize SCI0 PORT */
    P_PORTA.PACRL2.BIT.PA0MD = 1;                  /* Set RXD0(PA1:130pin@SH7145)              */

    P_SCI0.SCR_0.BIT.RE = 1;                       /* RE=1,receive enable                      */
}


/*****************************************************************************************************/
/*  Function    : rcv_sci                                                                            */
/*  Operation   : Receive serial data(SCI0)                                                          */
/*  Asynchronous Receive Mode                                                                        */
/*****************************************************************************************************/
unsigned char rcv_sci(unsigned char rev_count )
{
    while(P_SCI0.SSR_0.BIT.RDRF == 0){             /* Wait until RDRF flag high level          */
        err();                                     /* Error judging                            */

    /* Judging break detect                                                                         */
        if(P_SCI0.SCR_0.BIT.RE == 0)               /* Break detect                             */
            break;
    }

    Rev_data[rev_count] = P_SCI0.RDR_0;            /* Store receive data                       */

    P_SCI0.SSR_0.BIT.RDRF = 0;                     /* Clear RDRF flag                          */

    rev_count++ ;                                  /* Storing address increment                */

    return(rev_count);
}
```

```
/*****************************************************************************************/
/*  Function   : err                                                                     */
/*  Operation  : error judging(SCI0)                                                     */
/*  Asynchronous Receive Mode                                                            */
/*****************************************************************************************/
void err(void)
{
    if(P_SCI0.SSR_0.BIT.ORER == 1){                 /* Overrun error                    */
        P_SCI0.SSR_0.BIT.ORER = 0;                  /* Clear ORER flag                  */
    }

    if(P_SCI0.SSR_0.BIT.FER == 1){                  /* Framing error                    */

        if(P_PORTA.PADRL.BIT.PA0DR == 0){
            P_CMT.CMSTR.BIT.STR = 1;                /* CMT0 count start                 */

            while(T_count != 0);                    /* Wait until T_count = 0           */

            if(Rd_count == 3){                      /* Break detect (Rd_count==3)       */
                P_SCI0.SCR_0.BIT.RE = 0;            /* RE=0,receive disable             */
            }
        }

        P_SCI0.SSR_0.BIT.FER = 0;                   /* Clear FER flag                   */
    }

    if(P_SCI0.SSR_0.BIT.PER == 1){                  /* Parity error                     */
        P_SCI0.SSR_0.BIT.PER = 0;                   /* Clear PER flag                   */
    }
}
/*****************************************************************************************/
/*  Interruption Program                                                                 */
/*****************************************************************************************/
/*****************************************************************************************/
/*  CMT0 Interruption Program                                                            */
/*****************************************************************************************/
#pragma interrupt(cmt_int)
void cmt_int(void)
{
    P_CMT.CMCSR_0.BIT.CMF = 0;                      /* Clear CMF flag                   */

    /* Read RxD-pin level                                                               */
    if(P_PORTA.PADRL.BIT.PA0DR == 0){               /* RxD-pin level is low             */
        Rd_count++;                                 /* Count                            */
    }

    T_count--;                                      /* Count interrupt times            */

    /* Judging interrupt times                                                          */
    if(T_count == 0){                               /* 3 times interrupt                */
        P_CMT.CMSTR.BIT.STR = 0;                    /* CMT_0 count stop                 */
    }
}
/*****************************************************************************************/
/*  Other Interruption Program                                                           */
/*****************************************************************************************/
#pragma interrupt(dummy_f)
void dummy_f(void)
{
    /* Other Interrupt */
}
```

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Sep.16.04	—	First edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.