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.

SH7145F

Synchronous Serial Data Transmission/Reception

Summary

The SH7144 series is a single-chip microprocessor based on the SH-2 RISC (Reduced Instruction Set Computer) CPU core and integrating a number of peripheral functions.

This application note describes synchronous serial data transmission/reception using the SCI (Serial Communication Interface) module of the SH7145F. It is intended to be used as reference by users designing software applications.

The program examples contained in this application note have been tested. However, operation should be confirmed before using them in an actual application.

Device for Which Operation Has Been Confirmed

SH7145F

Contents

1. Specifications 2
2. Functions Used 3
3. Operation 6
4. Software 8
5. Flowcharts 11
6. Program Listing 14

1. Specifications

As shown in figure 1, synchronous serial data transmission and reception is performed using channel 1 (ch1) of the SCI module of the SH7145F. In this task example 3 bytes of serial data are transmitted from the SH7145, and serial data sent from the MCU, the transfer target, is received.

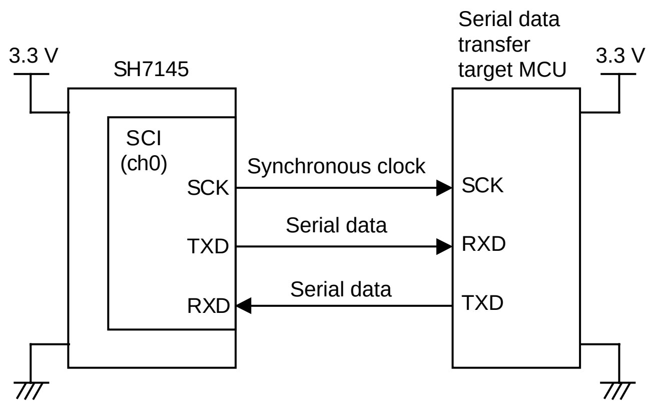


Figure 1 Synchronous Serial Data Transmission/Reception by SH7145

2. Functions Used

In this task example the SCI (Serial Communication Interface) is used to perform synchronous serial data transmission. Figure 2 shows a block diagram of channel 0 (ch0) of the SCI module. The functions of the elements shown in figure 2 are described below.

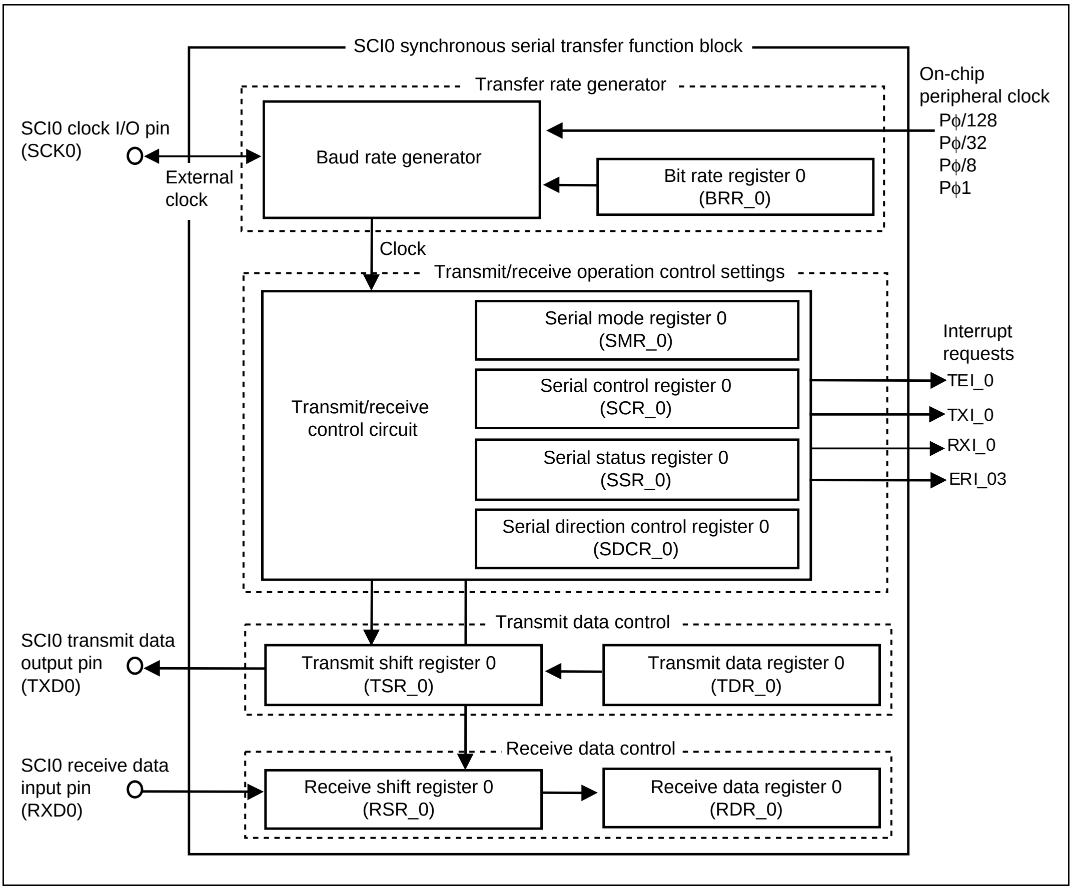


Figure 2 SCI (ch0) Block Diagram

- Synchronous Mode

  Data is transmitted and received in synchronization with clock pulses. The transfer data length is fixed at 8 bits.

- On-Chip Peripheral Clock Pϕ

  This is the reference clock for operation of on-chip peripheral functions. The clock signal is generated by a clock oscillator.

- Receive Shift Register (RSR_0)

  This register is used to receive serial data. Serial data is input to RSR_0 from the RxD_0 pin. When one frame of data has been received, it is automatically transferred to the receive data register (RDR_0). RSR_0 cannot be accessed by the CPU.

- Receive Data Register (RDR_0)

  Received data is stored in this 8-bit register. When one frame of data has been received, it is automatically transferred from RSR_0. RSR_0 and RDR_0 are in a double-buffer configuration, allowing continuous reception of data. RDR_0 is a receive-only register, so it can only be read by the CPU.

- Transmit Shift Register (TSR_0)

  This register is used to transmit serial data. In order to transmit data, the data is first transferred from the transmit data register (TDR_0) to TSR_0. Then the transmit data is output from the TxD_0 pin. TSR_0 cannot be accessed directly by the CPU.

- Transmit Data Register (TDR_0)

  Data to be transmitted is stored in this 8-bit register. When it is detected that TDR_0 is empty, data that has been written to TDR_0 is automatically transferred to TSR_0. TDR_0 and TSR_0 are in a double-buffer configuration. This allows data to be transferred to TSR_0 after one frame of data has been transmitted and the next frame of data is still being written to TDR_0, making possible continuous transmission of data. It is always possible to read or write to the TDR from the CPU, but before writing to the TDR it should be confirmed that the value of the TDRE bit in the serial status register (SSR_0) is 1.

- Serial Mode Register (SMR_0)

  This 8-bit register is used to select the serial data communication format and the clock source for the on-chip baud rate generator.

- Serial Control Register (SCR_0)

  This register is used for transmit and receive control, interrupt control, and to select the transmit and receive clock source.

- Serial Status Register (SSR_0)

  This register comprises the SCI0 status flag and the transmit and receive multiprocessor bits. TDRE, RDRF, ORER, PER, and FER can be cleared only.

- Serial Direction Control Register (SDCR_0)

  This register is used to select whether the LSB or MSB is first. For 8-bit communication either LSB-first or MSB-first may be selected, but LSB-first should be used for 7-bit communication.

- Bit Rate Register (BRR_0)

  This 8-bit register is used to adjust the bit rate. The SCI has independent baud rate generators for the individual channels, allowing different bit rates to be set for each. See the hardware manual for details on setting values, execution rate relationships, etc.

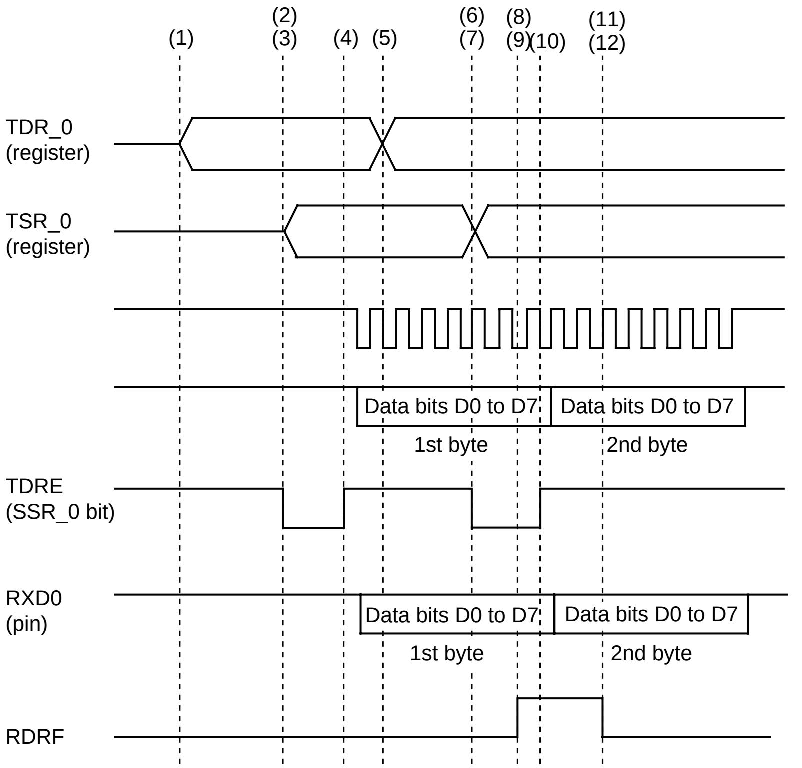
Table 1 shows the function allocations for the task example.

Table 1 Function Allocations

Function	Classification	Function Allocation
SCK0	Pin	Channel 0 synchronous clock I/O
TXD0	Pin	Channel 0 transmit data output pin
RXD0	Pin	Channel 0 transmit data input pin
SMR_0	SCI0	Sets communication format to synchronous mode
SCR_0	SCI0	Enables transmit operation
SSR_0	SCI0	Status flag showing SCI0 operation status
SDCR_0	SCI0	Specifies LSB-first
BRR_0	SCI0	Sets communication bit rate
TSR_0	SCI0	Register for transmitting serial data
TDR_0	SCI0	Register for storing transmit data

3. Operation

Figure 3 shows the operation of synchronous mode data transmission in the task example. To help explain figure 3, table 2 lists the software and hardware processing that is performed.



Note: To perform continuous reception, read data transferred to RDR until reception of next data is complete.

Figure 3 Data Transmission Operation

Table 2 Processing

	Software Processing	Hardware Processing
(1)	Write transmit data to TDR_0	—
(2)	Clear TDRE flag in SSR_0 to 0	—
(3)	—	Transfer data from TDR_0 to TSR_0
(4)	—	Set TDRE flag in SSR_0 to 0 and output transmit data from pin TXD0
(5)	Confirm TDRE is 0 and send next transmit data to TDR_0	—
(6)	Clear TDRE flag in SSR_0 to 0	
(7)	—	Transfer data from TDR_0 to TSR_0
(8)	—	RSR_0 receives serial data and transfers it to RDR_0
(9)	—	Set RDRF flag to 1
(10)	—	When output of previous data finishes, set TDRE flag in SSR_0 to 1 and output transfer data from pin TXD0
(11)	Read data from RDR_3	—
(12)	Clear RDRF flag in SSR_3 to 0	—
(13)	Repeat	Repeat

4. Software

(1) Module Descriptions

Table 3 lists the modules used in the task example.

Table 3 Module Descriptions

Module	Label	Function
Main routine	main	Calls modules
SCI routine	init_sci	Initial settings of SCI0
Transmit routine	trans_sci	Transmits serial data
Receive routine	rcv_sci	Receives serial data
Error handling	err_int	Handles receive errors

(2) Argument Descriptions

Table 4 lists the arguments used in the task example.

Table 4 Argument Descriptions

Argument	Function	Module
trans_data	Transmits data from SCI_0	Transfer routine
Rev_data[0–2]	Stores SCI_0 receive data	Receive routine

(3) On-Chip Register Descriptions

Table 5 lists the on-chip registers used in the task example. The set values shown are the values used in the task example and differ from the initial settings.

Table 5 On-Chip Register Descriptions

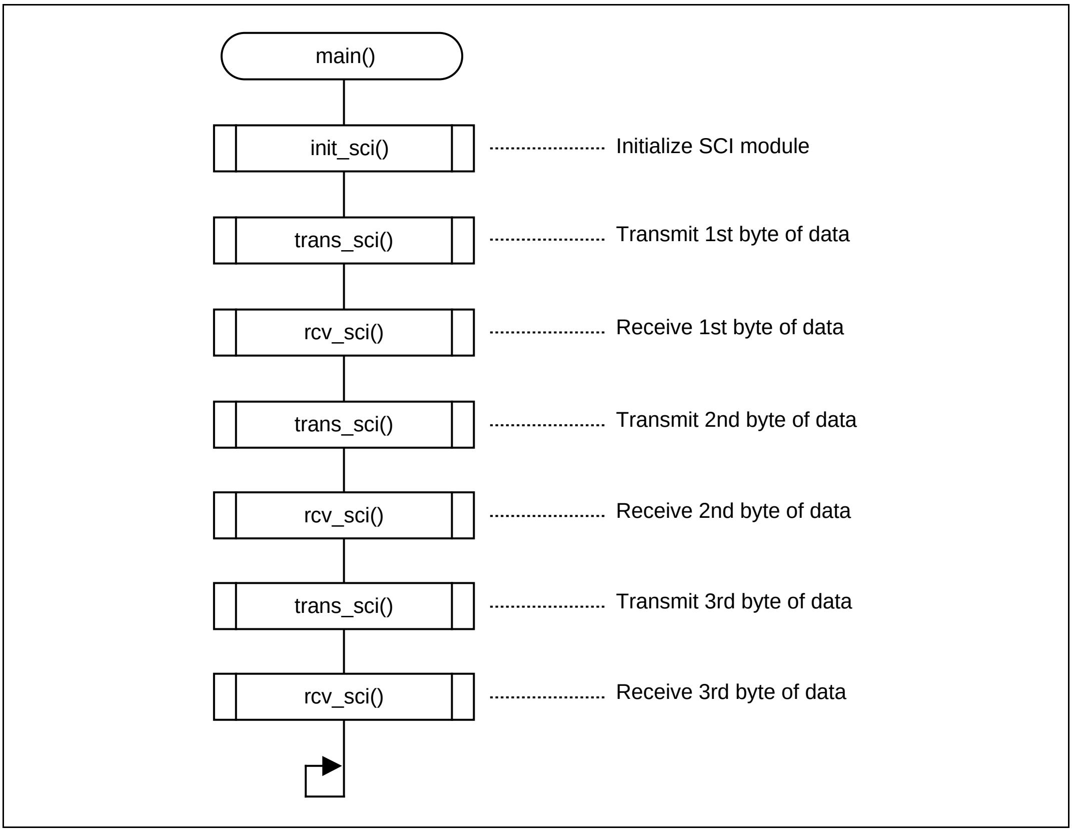
Register	Bit	Set Value	Function
MSTCR1	MSTP16	0	Module standby control register 1 SCI0 standby control bit Standby cancelled when MSTP16 = 0
SCR_0		H'70	Serial control register 0 (SCI_0) Transmit and receive control, interrupt control, transmit and receive clock source control
	TIE	0	Transmit interrupt enable TXI interrupt requests enabled when set to 1
	RIE	1	Receive interrupt enable RXI and ERI interrupt requests enabled when set to 1
	TE	1	Transmit enable Transmit operations enabled when set to 1
	RE	1	Receive enable Receive operations enabled when set to 1
	MPIE	0	Multiprocessor interrupt enable (In asynchronous mode, enabled when MP = 1 in SMR) In the task example, disabled because MP = 0
	TEIE	0	Transmit end interrupt enable TEI interrupt requests enabled when set to 1
	CKE1 CKE2	0 0	Clock enable 1, 0 Selects clock source and SCK pin function In the task example, clock source is on-chip clock and SCK pin is not used
SMR_0		H'00	Serial mode register 0 Selects communication format and the clock source for on-chip baud rate generator
	C/A	1	Communication mode Synchronous mode when cleared to 0
	CHR	0	Character length (enabled in asynchronous mode only) 8-bit transmission and reception when 0
	PE	0	Parity enable (enabled in asynchronous mode only) No-parity transmission and reception when 0
	O/E	0	Parity mode (enabled in asynchronous mode when PE = 1) (In this example PE = 0 and this bit is disabled)
	STOP	0	Stop bit length (enabled in asynchronous mode only) 1-stop-bit transmission and reception when 0

Register	Bit	Set Value	Function
SMR_0	MP	0	Multiprocessor mode (enabled in asynchronous mode only) Multiprocessor communication disabled when 0
	CKS1 CKS2	0 0	Clock select 1, 0 When value is 00, Pϕ clock selected using on-chip baud rate generator as clock source
BRR_0		H'40	Bit rate register 0 8-bit register for adjusting bit rate
SDCR_0		H'F2	Serial direction control register 0 DIR bit (bit 3) selects LSB-first or MSB-first In task example, DIR = 0 (LSB-first)
SSR_0		H'xx	Serial status register 0 Comprises SCI1 status flag and transmit and receive multiprocessor bits Only 0 may be written to the status flag, to clear it
	TDRE	*	Transmit data register empty (status flag)
	RDRF	*	Receive data register full (status flag)
	ORER	*	Overrun error (status flag)
	FER	*	Framing error (status flag)
	PER	*	Parity error (status flag)
	TEND	*	Transmit end (status flag)
	MPB	0	Multiprocessor bit
	MPBT	0	Multiprocessor bit transfer
PACRL2	PA2MD1 PA2MD0	0 1	Port A control register L2 Function setting for port A multiplex pin (SCK0)
	PA1MD1 PA1MD0	0 1	Port A control register L2 Function setting for port A multiplex pin (TXD0)
	PA0MD1 PA0MD0	0 1	Port A control register L2 Function setting for port A multiplex pin (RXD0)

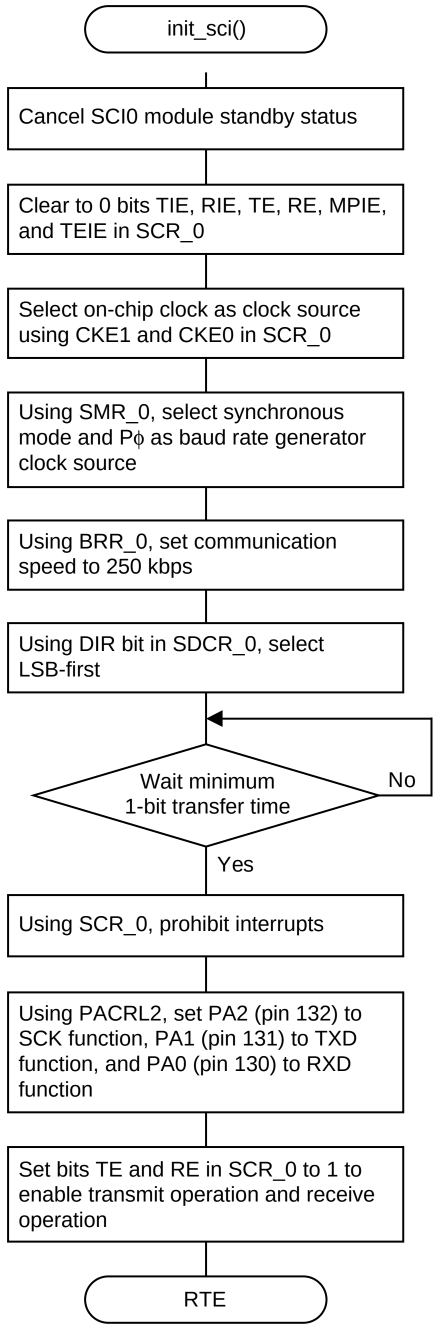
*: Can only be cleared to 0. Setting to 1 is performed by hardware.

5. Flowcharts

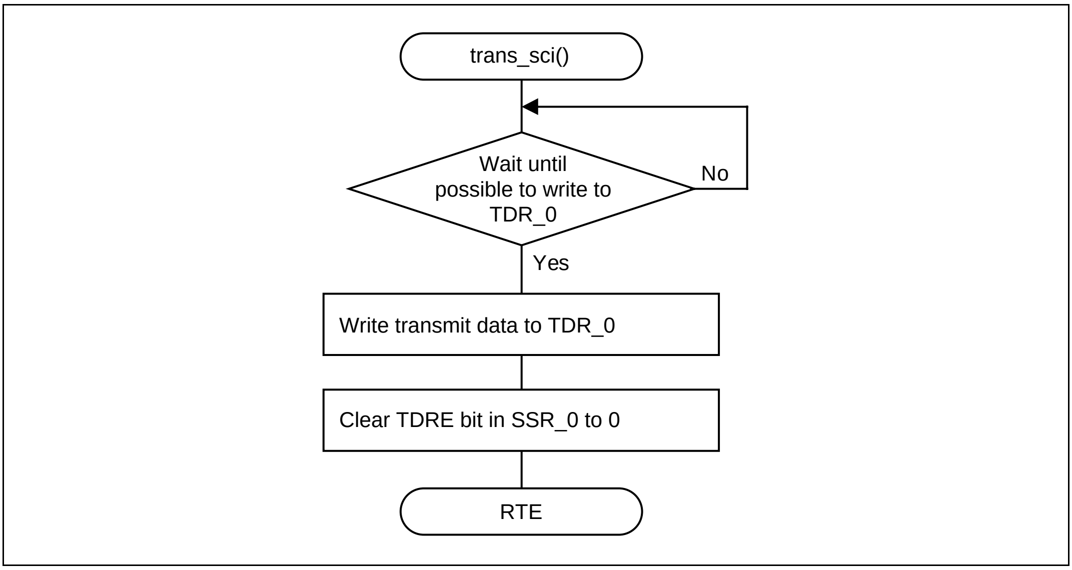
(1) Main Routine



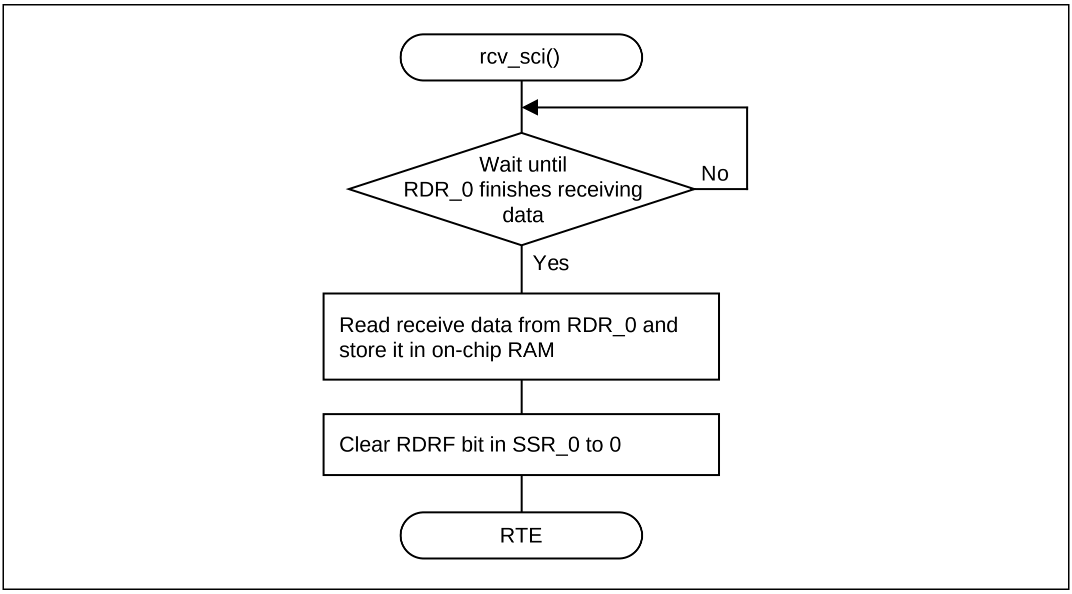
(2) SCI1 Initialize Routine



(3) Data Transfer Routine



(4) Data Receive Routine



6. Program Listing

```
/**************************************************************/
/* SH7145F Application Note                                   */
/*                                                            */
/* Function                                                   */
/*   :SCI0                                                    */
/*                                                            */
/* External input clock       :12.5MHz                        */
/* Internal CPU clock         :50MHz                          */
/* Internal peripheral clock  :25MHz                          */
/*                                                            */
/* Written                    :2003/9     Rev.1.0             */
/**************************************************************/

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

/*------------ Symbol Definition ----------------------------------------------*/

#define COUNT 3

#define TRANS_DATA_1 0x61
#define TRANS_DATA_2 0x62
#define TRANS_DATA_3 0x63

/*------------ Function Definition --------------------------------------------*/
void main(void);

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

void err_int(void);
void dummy_f(void);

/*------------ RAM allocation Definition --------------------------------------*/

volatile unsigned char Rev_data[COUNT];

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

   init_sci();                 /* Initialize SCI      */

   trans_sci(TRANS_DATA_1);
   i = rcv_sci(i);
```

```
   trans_sci(TRANS_DATA_2);
   i = rcv_sci(i);

   trans_sci(TRANS_DATA_3);
   i = rcv_sci(i);

   while(1);                    /* LOOP                   */
}
/***************************************************************
  Function  : init_sci
  Operation : Initialize serial (sci0, sci1)
       sci0 : Synchronous transmit operation
***************************************************************/
void init_sci(void)
{
   unsigned long i;

   P_STBY.MSTCR1.BIT.MSTP16 = 0;   /* disable SCI0 standby mode      */

   /* Initialize SCI       */
   P_SCI0.SCR_0.BYTE &= 0x03;      /* clear TIE,RIE,TE,RE,MPIE,TEIE  */
   P_SCI0.SCR_0.BIT.CKE = 0;       /* clock:internal,SCK:output      */
   P_SCI0.SMR_0.BYTE = 0x80;       /* Clock synchronous mode         */
            //   CA = 1;           /* Clock synchronous mode         */

   P_SCI0.BRR_0 = 24;              /* 250kbps@25MHz(Peripheral)      */
   P_SCI0.SDCR_0.BIT.DIR = 0;      /* LSB first send                 */

   for( i=0; i < 0x0300 ; i++);    /* Wait 1bit                      */

   /* Initialize SCI0 PORT         */
   P_PORTA.PACRL2.BIT.PA2MD = 1;   /* set SCK0(PA2:132pin@SH7145)    */
   P_PORTA.PACRL2.BIT.PA1MD = 1;   /* set TXD0(PA1:131pin@SH7145)    */
   P_PORTA.PACRL2.BIT.PA0MD = 1;   /* set RXD0(PA0:130pin@SH7145)    */

   P_PORTA.PAIORL.BIT.PA2IOR = 1;   /* SCK0 Output                   */

   P_SCI0.SCR_0.BYTE = 0x30;        /* TE=RE=1,Transmit Receive Enable(SCI0) */

}

/**************************************************************************/
/* Function       : trans_sci                                             */
/* Operation      : Write 1 character to serial output                    */
/* Argument       : c                                                     */
/* Value returned : None                                                  */
/**************************************************************************/
void trans_sci(char c )
{
   while(!(P_SCI0.SSR_0.BYTE & 0x80)){   /*Wait until data can be written to TDR*/
```

```
      ;
   }

   P_SCI0.TDR_0 = (unsigned char)c;      /* Write data to TDR          */
   P_SCI0.SSR_0.BYTE &= 0x7F;            /* Clear flag, transmit        */
}

/************************************************************************/
/* Function       : rcv_sci                                            */
/* Operation      : Store receive data in on-chip RAM                  */
/* Argument       : rev_count                                          */
/* Value returned : rev_count                                          */
/************************************************************************/
unsigned char rcv_sci(unsigned char rev_count)
{
   while(P_SCI0.SSR_0.BIT.RDRF == 0);   /* Wait until reception is finished  */

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

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

   rev_count++ ;                        /* Increment storage address         */

   return(rev_count);
}

/**************************************
          Interrupt handling
**************************************/
#pragma interrupt(err_int)
void err_int(void)
{
   if(P_SCI0.SSR_0.BIT.ORER == 1){      /* Overrun error           */
      P_SCI0.SSR_0.BIT.ORER = 0;        /* ORER flag clear         */
   }
}

#pragma interrupt(dummy_f)
void dummy_f(void)
{
   /*   Other Interrupt   */
}
```

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.