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.

H8SX Family

Asynchronous Transfer of Data with Appended CRC Codes via an SCI Interface

Introduction

The SCI module and CRC calculator applied in the asynchronous transfer of four-byte data blocks. In transmission, a two-byte CRC is appended to every byte of transmitted data. In reception, for each byte of received data, the corresponding two-byte CRC code is used to check for CRC errors in the data block.

Target Device

H8SX/1653

Contents

1. Specification

The SCI module and CRC calculator are applied to transmit and receive four-byte sequences of data by asynchronous transfer.

- An example of the connection for this application is given as figure 1.
- Details of the format for data transfer are given in table 1.
- On the transmitting side, a two-byte CRC code is appended to every one byte of data for transmission. Thus, for every four-byte unit of data for transmission, a total of 12 bytes is transmitted, including the appended CRC codes.
- On the receiving side, a total of 12 bytes is received, including the appended CRC codes. The data block is checked for CRC errors by using the corresponding two-byte CRC code to check every byte of received data.
- In this example, asynchronous transfer is performed through CPU processing under software control.

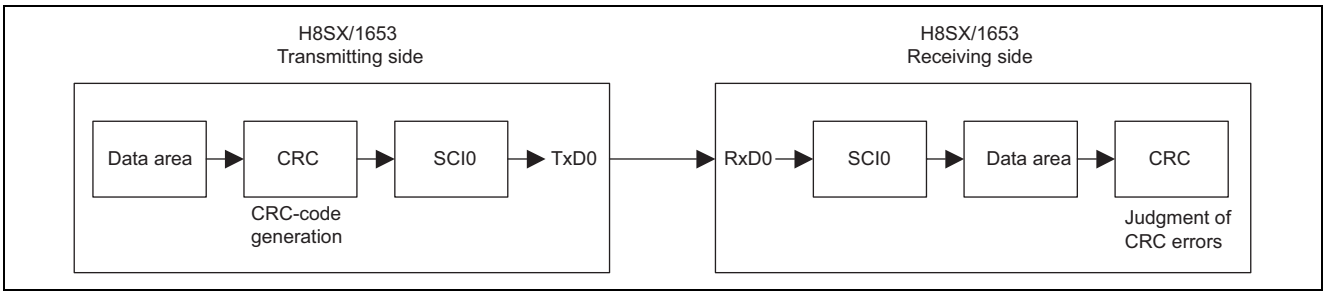


Figure 1 Asynchronous Transfer of Data with Appended CRC Codes via an SCI

Table 1 Asynchronous Serial Transmission and Reception Format

Format	Setting
Pϕ	32 MHz
Serial transfer mode	Asynchronous
Clock source	Internal baud rate generator
Transfer rate	38,400 bps
Data length	8 bits
Parity bit	None
Stop bit	One
Serial–parallel conversion format	LSB first
Error detection	Polynomial for CRC code generation: $X^{16} + X^{12} + X^5 + 1$

2. Applicable Conditions

Table 2 Applicable Conditions

Item	Setting	
Operating frequency	Input clock	: 16 MHz
	System clock (Iϕ)	: 32 MHz (input clock frequency × 2)
	Peripheral module clock (Pϕ)	: 32 MHz (input clock frequency × 2)
	External bus clock (Bϕ)	: 32 MHz (input clock frequency × 2)
Operating mode	Mode 6 (MD2 = 1, MD1 = 1, and MD0 = 0, MD_CLK = 0)	

3. Description of Modules Used

3.1 Description of SCI_0

In this example SCI_0 is applied for asynchronous serial data transmission. Figure 2 is a block diagram of SCI_0.

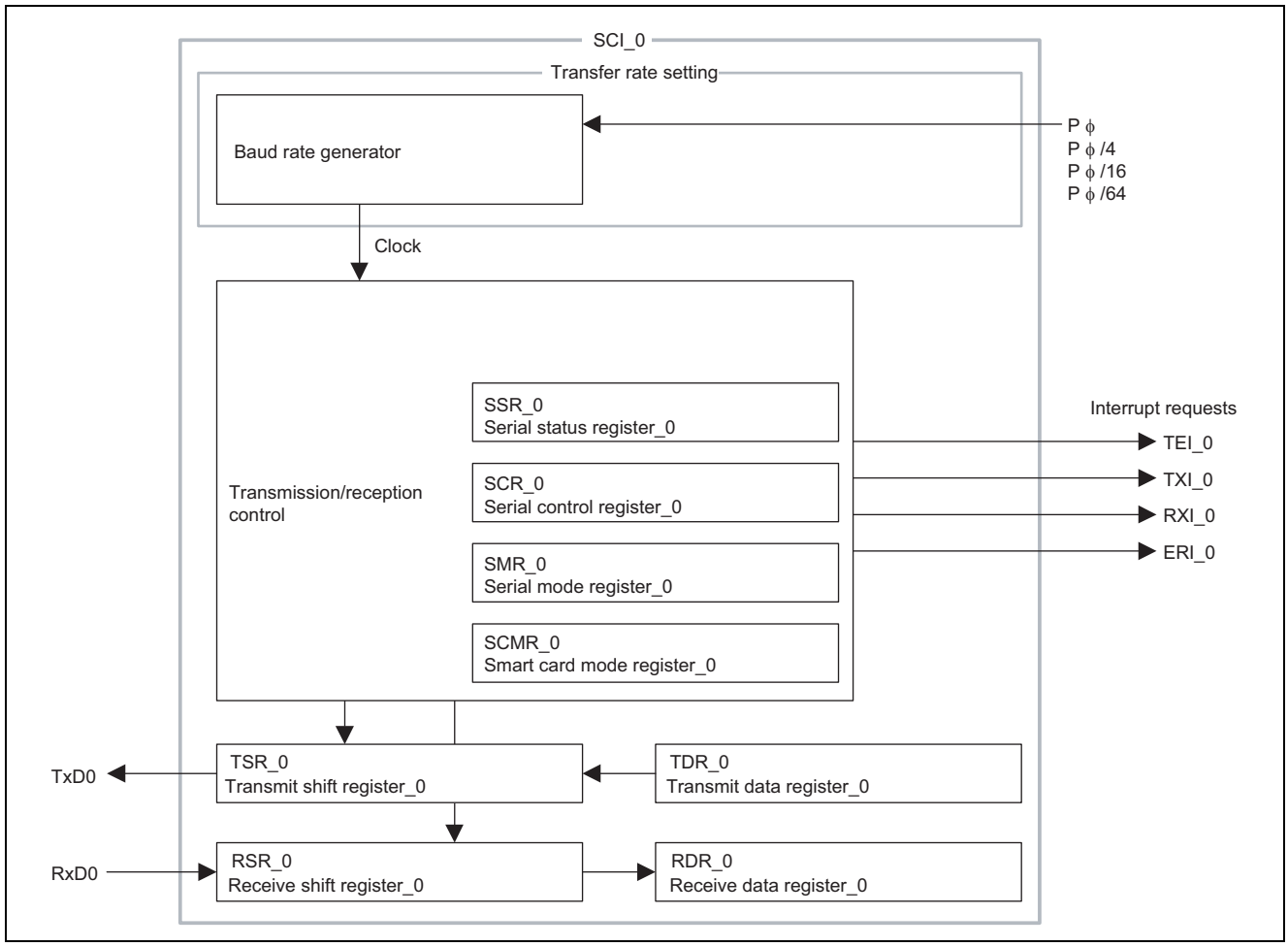


Figure 2 Block Diagram of SCI_0

Explanations of the functional elements shown in figure 2 are given below.

- Internal peripheral clock (Pϕ)
  This base clock, which drives operation of the internal peripheral modules, is generated by a clock oscillator.
- Receive shift register_0 (RSR_0)
  RSR_0 is for the reception of serial data. Serial data received on the RxD0 pin are input to RSR_0 and, when one frame of data has been received, the data are automatically transferred to the receive data register (RDR_0). RSR_0 is not accessible from the CPU.
- Receive data register 0 (RDR_0)
  RDR_0 is an 8-bit register and used to store received data. When one frame of data has been received, it is automatically transferred from RSR_0 to here. RSR_0 and RDR_0 form a buffered structure and thus allow continuous reception operations. Since RDR_0 is a register especially for reception, it is read-only from the CPU.
- Transmit shift register 0 (TSR_0)
  TSR_0 is for the transmission of serial data. On every round of transmission, data for transmission are transferred from the transmit data register (TDR_0) to TSR_0, which then outputs the data via pin TxD0. This register is not directly accessible from the CPU.
- Transmit data register 0 (TDR_0)
  TDR_0 is an 8-bit register and used to store data for transmission.
  After detection of TSR_0 having become empty, data that have been written to TDR_0 are automatically transferred to TSR_0. Furthermore, TDR_0 and TSR_0 form a buffered structure, so each time the data for one frame is transferred to TSR_0, data for the next can be written to TDR_0, allowing continuous transmission of transferred data from TSR_0. TDR_0 is always readable and writable from the CPU; however, only proceed with writing after checking that the TDRE bit of the serial status register (SSR_0) is one.
- Serial mode register 0 (SMR_0)
  SMR_0 is an 8-bit register and used to select the format for serial data transfer and the source of the clock signal for the internal baud-rate generator.
- Serial control register 0 (SCR_0)
  SCR_0 is an 8-bit register and used to control transfer and interrupts, and to select the source of the clock signal that provides timing for the transfer.
- Serial status register 0 (SSR_0)
  SSR_0 consists of the status flags for SCI_0 and the multiprocessor bit for transfer. For bits TDRE, RDRF, ORER, PER, and FER, the only effective write operation is clearing.
- Smart card mode register 0 (SCMR_0)
  SCMR_0 is used to select the smart-card interface and the format for the interface. In this example, this register is used to make the “normal asynchronous or clock-synchronous mode” setting.
- Bit rate register 0 (BRR_0)
  BRR_0 is an 8-bit register that is used to adjust the bit rate.

3.2 CRC Calculator

An outline of the CRC calculator is given below.

- A CRC code is generated for any 8-bit unit of data.
- CRC calculation for each 8-bit unit is executed as a parallel operation.
- Any of three generating polynomials is selectable.
- The generation of CRC codes for use in either LSB-first or MSB-first transfer is selectable.

Figure 3 is a block diagram of the CRC calculator.

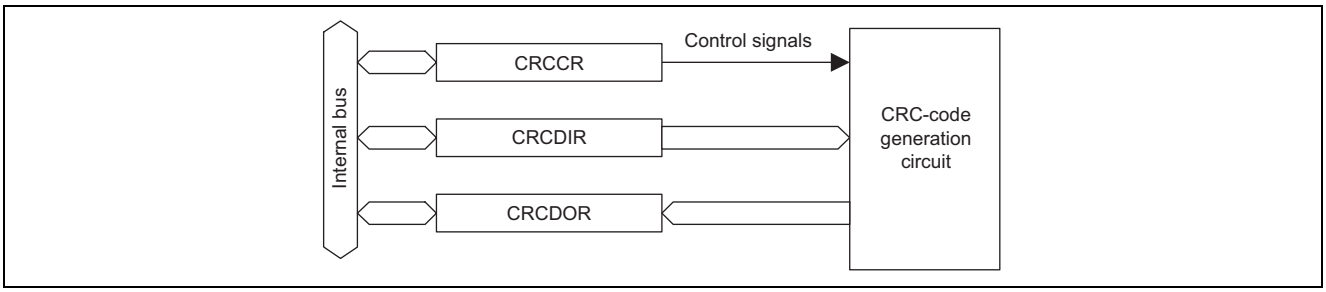


Figure 3 Block Diagram of the CRC Calculator

Explanations of the functional elements shown in figure 3 are given below.

- CRC control register (CRCCR)
  CRCCR is used to initialize the CRC calculator, switch the calculation, and select the generating polynomial.
- CRC data input register (CRCDIR)
  CRCDIR is an 8-bit readable/writable register. When a byte for the CRC calculator is written to the CRCDIR, the result can be acquired from the CRCDOR.
- CRC data output register (CRCDOR)
  CRCDOR is a 16-bit readable/writable register. Once the CRCDOR has been cleared, when a byte for the CRC calculator is written to the CRCDIR, the result can be acquired from the CRCDOR. When a target byte for CRC calculation and the appended CRC code are written to the CRCDIR, the result will be H'0000 if there is no CRC error. In cases where CRCCR bits 1 and 0 have been set to G1 = 0 and G0 = 1, the result is acquired in the lower-order byte.

4. Principles of Operation

4.1 Data Transmission with the CRC Calculator

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer. Figure 4 illustrates an example of usage to generate a CRC code for use in LSB-first transfer of H'F0 as the data. In the figure, bits G1 and G0 of the CRCCR have the value B'11, selecting $X^{16} + X^{12} + X^5 + 1$ as the generating polynomial.

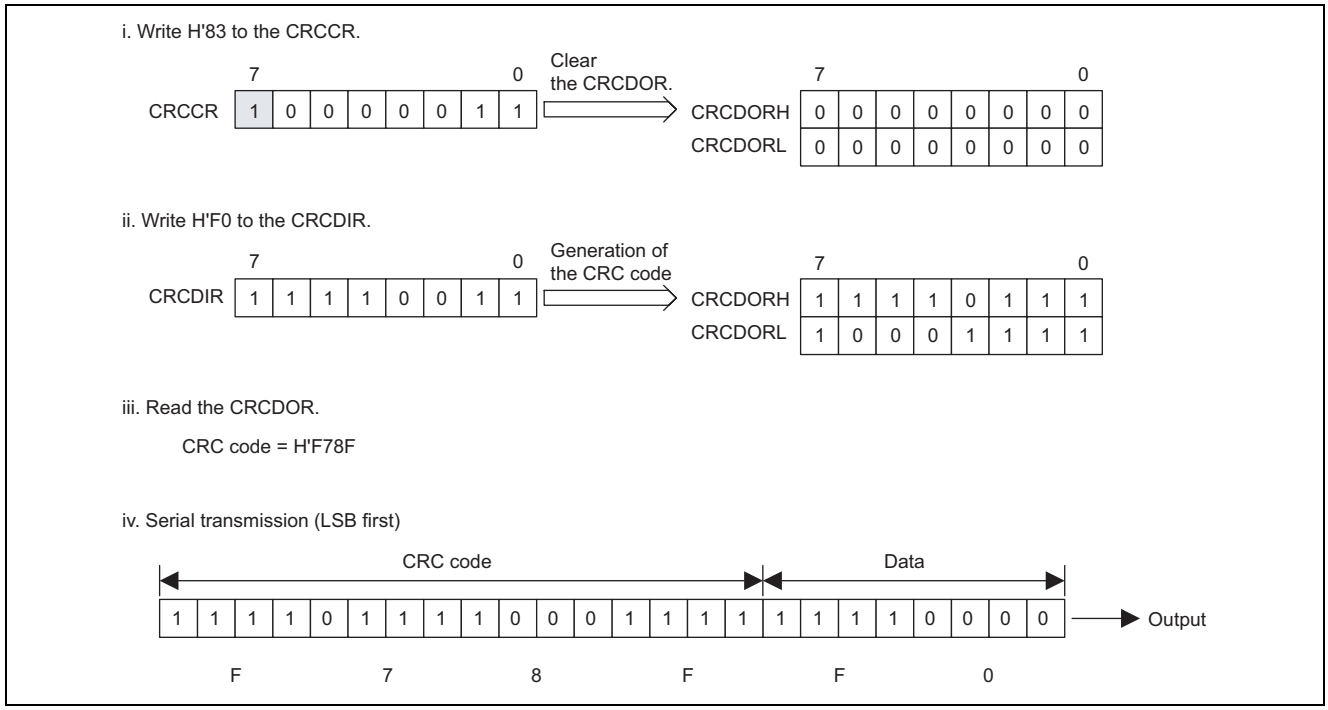


Figure 4 LSB-First Data Transmission

4.2 Data Reception with the CRC Calculator

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer. Figure 5 illustrates an example of usage to generate a CRC code for use in LSB-first transfer of H'F0 as the data. In the figure, bits G1 and G0 of the CRCCR have the value B'11, selecting $X^{16} + X^{12} + X^5 + 1$ as the generating polynomial.

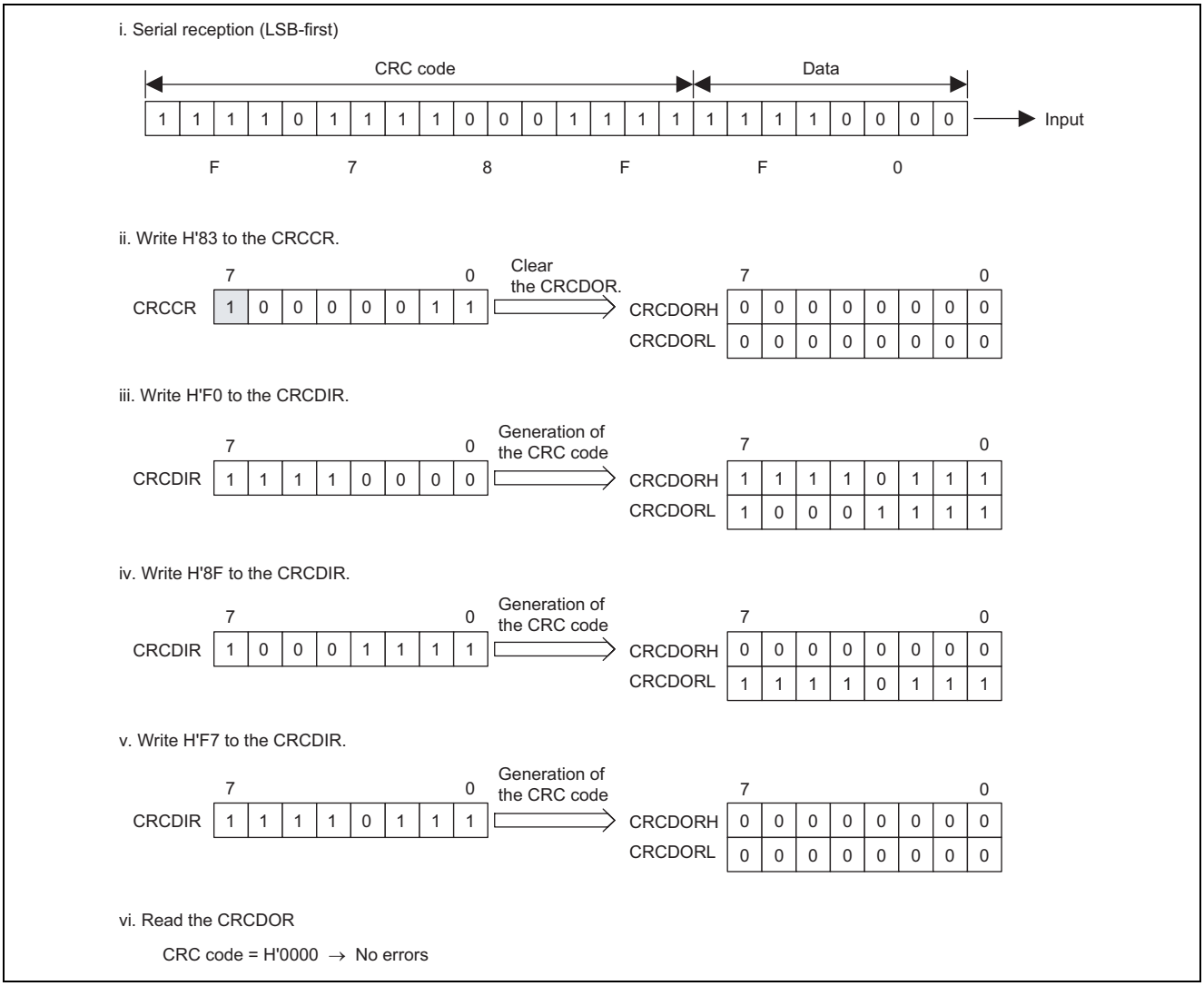


Figure 5 LSB-First Data Reception

5. Description of Software

5.1 List of Functions

The functions of this sample application are listed in table 3. The hierarchical structure of calls on the respective sides is illustrated in figures 6 and 7.

Table 3 List of Functions

Function Name	Description
init	Initialization routine Sets the CCR and configures the clocks, releases the required modules from the module stop mode, and calls the main function.
main	Main routine Transmitting side: Makes settings for asynchronous serial transfer; handles processing to transmit 12 bytes (data and appended CRC codes) Receiving side: Makes settings for asynchronous serial transfer; handles processing to receive 12 bytes (data and appended CRC codes), and to check the data block for errors
sci0_init	Initialization for asynchronous serial transfer
sci0_rcv1byte	Receives one byte through asynchronous serial transfer
sci0_trs1byte	Transmits one byte through asynchronous serial transfer

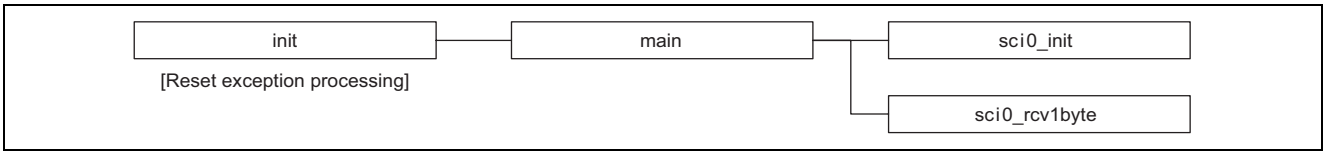


Figure 6 Hierarchical Structure on the Transmitting Side

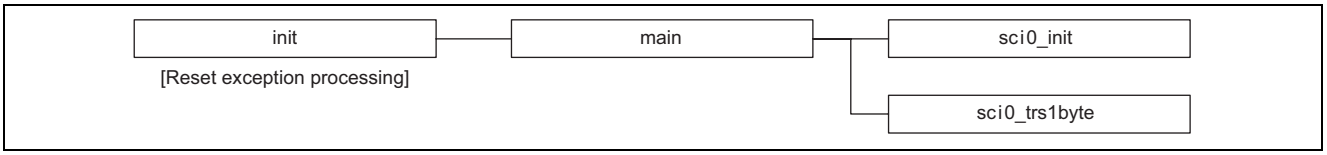


Figure 7 Hierarchical Structure on the Receiving Side

5.2 Vector Table

The vector table for exception processing in this example is shown in table 4.

Table 4 Vector Table for Exception Processing

Source for Exception Handling	Vector Number	Address in Vector Table	Destination Function
Reset	0	H'000000	init

5.3 RAM Usage

Table 5 RAM Usage

Type	Variable Name	Description	Used in
unsigned char	errbuf[4]	CRC error buffer Storage of results of CRC calculation for error detection	Receiving-side main
unsigned char	rcv_dt[4][3]	RAM area for storage of received data with appended CRC codes	Receiving-side main

5.4 Data Table

Table 6 Data Table

Type	Variable Name	Description	Used in
unsigned char	trs_dt[4]	ROM area for storage of data for transmission; contains four bytes of data: H'41, H'42, H'43, H'44	Transmitting-side main

5.5 Macro Definitions

Table 7 Macro Definitions

Identifier	Description	Used in
TRANSMIT	Generate the transmitting-side program.	Transmitting-side main
RECEIVE	Generate the receiving-side program.	Receiving-side main

5.6 Symbolic Constant

Table 8 Symbolic Constant

Constant Name	Setting	Description
NUM	4	Setting of the number of bytes in the data blocks for transmission and reception

5.7 Description of Functions

5.7.1 init Function

1. Functional overview
   Initialization routine, which releases the required modules from module-stop mode, configures the clocks, and calls the main function.
2. Arguments
   None
3. Return value
   None
4. Description of internal registers used
   The internal registers used in this sample task are described below. The settings shown in these tables are the values used in this sample task and differ from the initial values.

- Mode control register (MDCR) Number of bits: 16 Address: H'FFFDC0

Bit	Bit Name	Setting	R/W	Description
15	MDS7	Undefined*	R	Indicates the value set by a mode pin (MD3). When MDCR is read, the input level on the MD3 pin is latched. The latching is released by a reset.
11	MDS3	Undefined*	R	Mode Select 3 to 0
10	MDS2	Undefined*	R	These bits indicate the operating mode selected by mode pins (MD2 to MD0; see table 9). When MDCR is read, the signal levels input on pins MD2 to MD0 are latched into these bits. The latching is released by a reset.
9	MDS1	Undefined*	R	
8	MDS0	Undefined*	R	

Note: * Values are determined by the settings on pins MD0 to MD3.

Table 9 Settings of Bits MDS3 to MDS0

MCU Operating Mode	Mode Pins			MDCR			
	MD2	MD1	MD0	MDS3	MDS2	MDS1	MDS0
2	0	1	0	1	1	0	0
4	1	0	0	0	0	1	0
5	1	0	1	0	0	0	1
6	1	1	0	0	1	0	1
7	1	1	1	0	1	0	0

- System clock control register (SCKCR) Number of bits: 16 Address: H'FFFDC4

Bit	Bit Name	Setting	R/W	Description
10	ICK2	0	R/W	System Clock (Iϕ) Select
9	ICK1	0	R/W	These bits select the frequency of the system clock, which is provided to the CPU, DMAC, and DTC.
8	ICK0	1	R/W	001: Input clock × 2
6	PCK2	0	R/W	Peripheral Module Clock (Pϕ) Select
5	PCK1	0	R/W	These bits select the frequency of the peripheral module clock.
4	PCK0	1	R/W	001: Input clock × 2
2	BCK2	0	R/W	External Bus Clock (Bϕ) Select
1	BCK1	0	R/W	These bits select the frequency of the external bus clock.
0	BCK0	1	R/W	001: Input clock × 2

- MSTPCRA, B and C control module stop mode. Setting a bit to 1 makes the corresponding module enter the module stop mode, while clearing the bit to 0 releases the module from module stop mode.

- Module stop control register A (MSTPCRA) Number of bits: 16 Address: H'FFFDC8

Bit	Bit Name	Setting	R/W	Description
15	ACSE	0	R/W	Enables/disables all-module-clock-stop mode for reducing current consumption by stopping operation of the bus controller and I/O ports when the CPU executes the SLEEP instruction after the module stop mode has been set for all of the on-chip peripheral modules controlled by MSTPCR. 0: All-module-clock-stop mode disabled 1: All-module-clock-stop mode enabled
13	MSTPA13	1	R/W	DMA controller (DMAC)
12	MSTPA12	1	R/W	Data transfer controller (DTC)
9	MSTPA9	1	R/W	8-bit timer unit (TMR_3 and TMR_2)
8	MSTPA8	1	R/W	8-bit timer unit (TMR_1 and TMR_0)
5	MSTPA5	1	R/W	D/A converter (channels 1 and 0)
3	MSTPA3	1	R/W	A/D converter (unit 0)
0	MSTPA0	1	R/W	16-bit timer pulse unit (TPU channels 5 to 0)

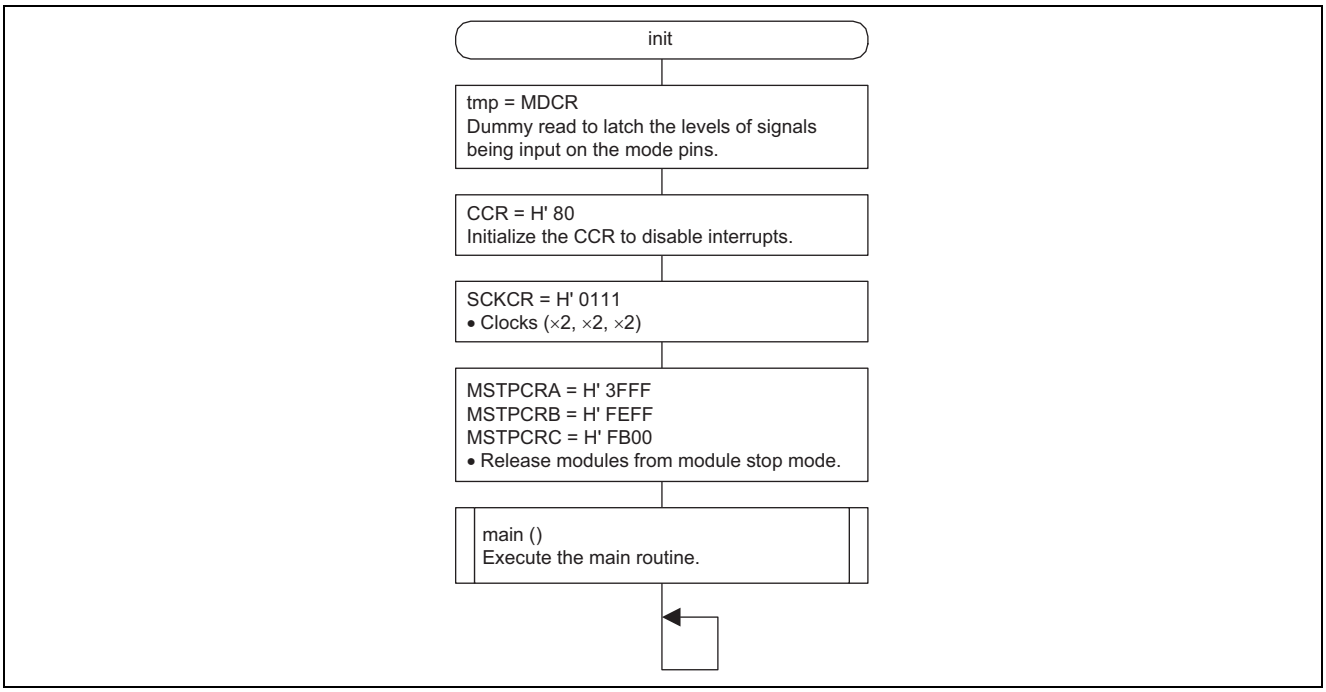
- Module stop control register B (MSTPCRB) Number of bits: 16 Address: H'FFFDCA

Bit	Bit Name	Setting	R/W	Description
15	MSTPB15	1	R/W	Programmable pulse generator (PPG)
12	MSTPB12	1	R/W	Serial communications interface_4 (SCI_4)
10	MSTPB10	1	R/W	Serial communications interface_2 (SCI_2)
9	MSTPB9	1	R/W	Serial communications interface_1 (SCI_1)
8	MSTPB8	0	R/W	Serial communications interface_0 (SCI_0)
7	MSTPB7	1	R/W	I^2C bus interface 1 (IIC_1)
6	MSTPB6	1	R/W	I^2C bus interface 0 (IIC_0)

- Module stop control register C (MSTPCRC) Number of bits: 16 Address: H'FFFDCC

Bit	Bit Name	Setting	R/W	Description
15	MSTPC	1	R/W	Serial communications interface_5 (SCI_5) , (IrDA)
14	MSTPC	1	R/W	Serial communications interface_6 (SCI_6)
13	MSTPC	1	R/W	8-bit timer unit (TMR_4, TMR_5)
12	MSTPC	1	R/W	8-bit timer unit (TMR_6, TMR_7)
11	MSTPC	1	R/W	Universal serial bus interface (USB)
10	MSTPC	0	R/W	Cyclic redundancy check module
4	MSTPC	0	R/W	On-chip RAM_4 (H'FF2000 to H'FF3FFF)
3	MSTPC	0	R/W	On-chip RAM_3 (H'FF4000 to H'FF5FFF)
2	MSTPC	0	R/W	On-chip RAM_2 (H'FF6000 to H'FF7FFF)
1	MSTPC	0	R/W	On-chip RAM_1 (H'FF8000 to H'FF9FFF)
0	MSTPC	0	R/W	On-chip RAM_0 (H'FFA000 to H'FFBFFF)

5. Flowchart


init
tmp = MDCR
Dummy read to latch the levels of signals being input on the mode pins.
CCR = H' 80
Initialize the CCR to disable interrupts.
SCKCR = H' 0111
• Clocks (×2, ×2, ×2)
MSTPCRA = H' 3FFF
MSTPCRB = H' FEFF
MSTPCRC = H' FB00
• Release modules from module stop mode.
main ()
Execute the main routine.

5.7.2 main Function on the Transmitting Side (TRANSMIT)

1. Functional overview
   The main routine makes settings for asynchronous serial transfer and handles processing for the transmission of 12 bytes (data and appended CRC codes).
2. Arguments
   None
3. Return value
   None
4. Description of internal registers used
   The internal registers used in this sample task are described below. The settings shown in these tables are the values used in this sample task and differ from the initial values.

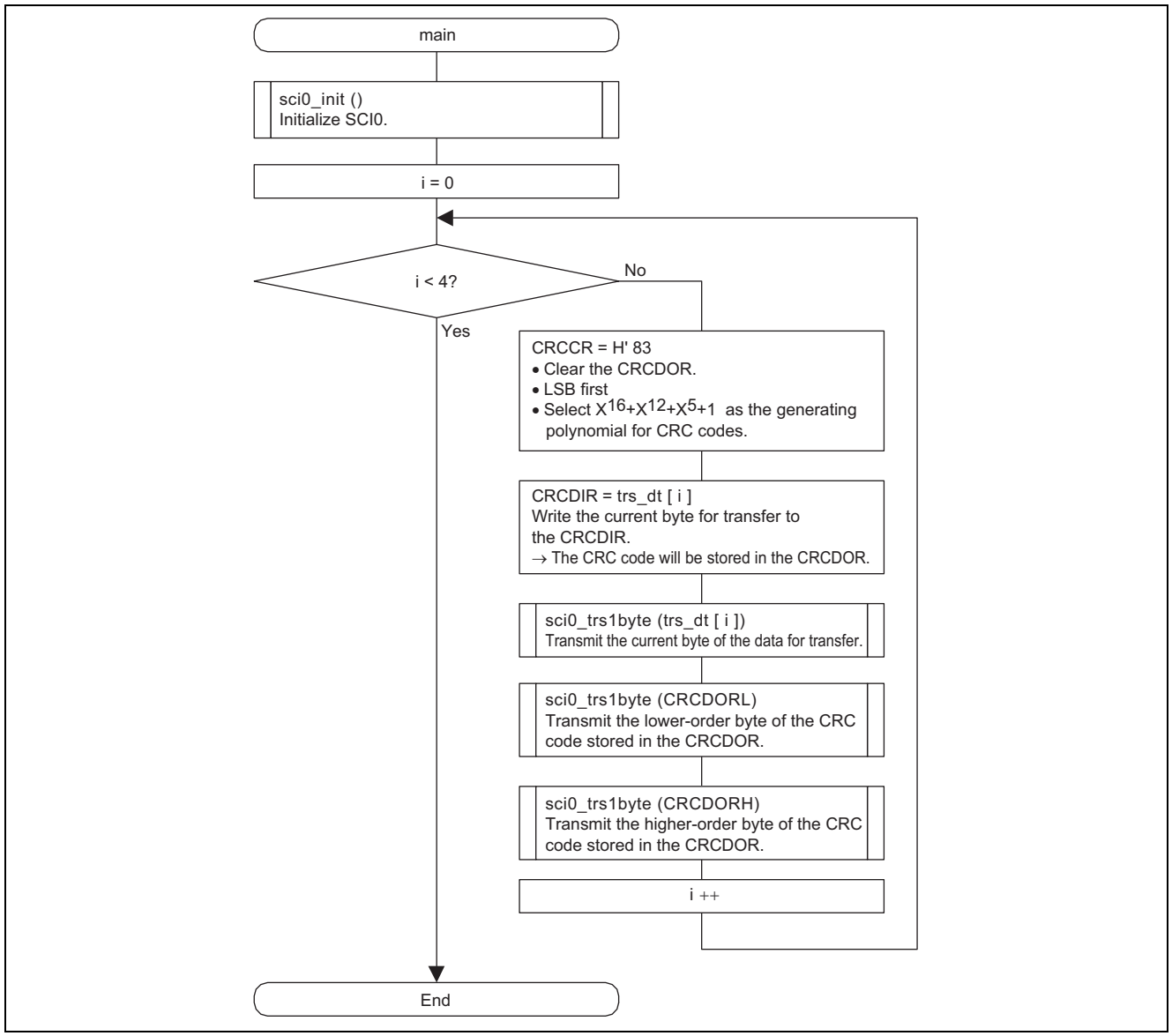
- CRC control register (CRCCR) Number of bits: 8 Address: H'FFEA4C

Bit	Bit Name	Setting	R/W	Description
7	DORCLR	1	W	CRCDOR Clear Setting this bit to 1 clears CRCDOR to H'0000.
2	LMS	0	R/W	RC Calculation Selects CRC code generation for LSB-first or MSB-first communications. 0: Performs CRC operation for LSB-first communications. The lower-order byte (bits 7 to 0) is transmitted first when the CRCDOR contents (the CRC code) are divided into two bytes for transmission. 1: Performs CRC operation for MSB-first communications. The higher-order byte (bits 15 to 8) is transmitted first when the CRCDOR contents (the CRC code) are divided into two bytes for transmission.
1 0	G1 G0	1 1	R/W	CRC Generating Polynomial Select Selects the polynomial 00: Reserved 01: $X^8 + X^2 + X + 1$ 10: $X^{16} + X^{15} + X^2 + 1$ 11: $X^{16} + X^{12} + X^5 + 1$

- CRC data input register (CRCDIR) Number of bits: 8 Address: H'FFEA4D
  Function: CRCDIR is an 8-bit readable and writable register. Once the target byte for CRC calculation has been written to CRCDIR, the result becomes available in CRCDOR.
  Setting: trs_dt[i]

- CRC data output register (CRCDOR) Number of bits: 16 Address: H'FFEA4E
  Function: CRCDOR is a 16-bit readable and writable register. When CRCDOR has been cleared, the result of CRC calculation can be acquired from CRCDOR after the target byte for CRC calculation has been written to CRCDIR. When the result of CRC calculation for the target byte is present and the CRC data that was appended to the byte is written to CRCDIR, the result in CRCDOR will be H'0000 if there are no errors. When bits 1 and 0 of the CRCCR are set to G1 = 0 and G0 = 1, results are acquired from the lower-order byte.
  Setting: Undefined

5. Flowchart


main
sci0_init ()
Initialize SCI0.
i = 0
i < 4?
No
Yes
CRCCR = H' 83
• Clear the CRCDOR.
• LSB first
• Select X16+X12+X5+1 as the generating polynomial for CRC codes.
CRCDIR = trs_dt [i]
Write the current byte for transfer to the CRCDIR.
→ The CRC code will be stored in the CRCDOR.
sci0_trs1byte (trs_dt [i])
Transmit the current byte of the data for transfer.
sci0_trs1byte (CRCDORL)
Transmit the lower-order byte of the CRC code stored in the CRCDOR.
sci0_trs1byte (CRCDORH)
Transmit the higher-order byte of the CRC code stored in the CRCDOR.
i ++
End

5.7.3 main Function on the Receiving Side (RECEIVE)

1. Functional outline
   The main routine makes settings for asynchronous serial transfer and implements the reception of the 12 bytes (data and appended CRC codes) and checking of the data block for CRC errors.
2. Arguments
   None
3. Return value
   None
4. Description of internal registers used
   The internal registers used in this sample task are described below. The settings shown in these tables are the values used in this sample task and differ from the initial values.

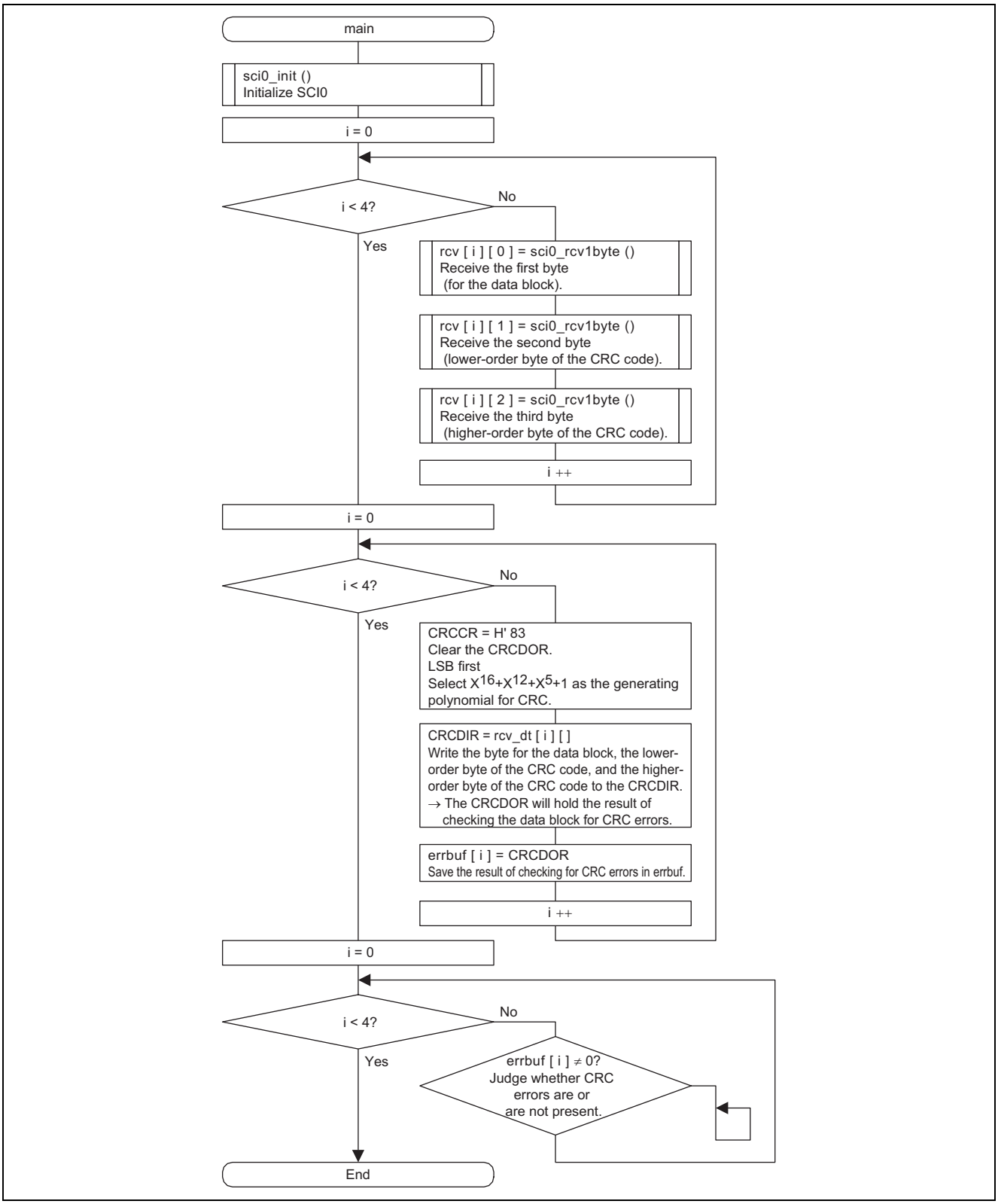
- CRC control register (CRCCR) Number of bits: 8 Address: H'FFEA4C

Bit	Bit Name	Setting	R/W	Description
7	DORCLR	1	W	CRCDOR Clear Setting this bit to 1 clears CRCDOR to H'0000.
2	LMS	0	R/W	Switch CRC Calculation Selects CRC code generation for LSB-first or MSB-first communications. 0: Performs CRC operation for LSB-first communications. The lower-order byte (bits 7 to 0) is transmitted first when the CRCDOR contents (the CRC code) are divided into two bytes for transmission. 1: Performs CRC operation for MSB-first communications. The higher-order byte (bits 15 to 8) is transmitted first when the CRCDOR contents (the CRC code) are divided into two bytes for transmission.
1	G1	1	R/W	CRC Generating Polynomial Select
0	G0	1		Selects the polynomial. 00: Reserved 01: $X^8 + X^2 + X + 1$ 10: $X^{16} + X^{15} + X^2 + 1$ 11: $X^{16} + X^{12} + X^5 + 1$

- CRC data input register (CRCDIR) Number of bits: 8 Address: H'FFEA4D
  Function: CRCDIR is an 8-bit readable and writable register. Once the target byte for CRC calculation has been written to CRCDIR, the result becomes available in CRCDOR.
  Setting: rcv_dt[i][]

- CRC data output register (CRCDOR) Number of bits: 16 Address: H'FFEA4E
  Function: CRCDOR is a 16-bit readable and writable register. When CRCDOR has been cleared, the result of CRC calculation can be acquired from CRCDOR after the target byte for CRC calculation has been written to CRCDIR. When the result of CRC calculation for the target byte is present and the CRC data that was appended to the byte is written to CRCDIR, the result in CRCDOR will be H'0000 if there are no errors. When bits 1 and 0 of the CRCCR are set to G1 = 0 and G0 = 1, results are acquired from the lower-order byte.
  Setting: Undefined

5. Flowchart


main
sci0_init ()
Initialize SCI0
i = 0
i < 4?
No
Yes
rcv [i] [0] = sci0_rcv1byte ()
Receive the first byte
(for the data block).
rcv [i] [1] = sci0_rcv1byte ()
Receive the second byte
(lower-order byte of the CRC code).
rcv [i] [2] = sci0_rcv1byte ()
Receive the third byte
(higher-order byte of the CRC code).
i ++
i = 0
i < 4?
No
Yes
CRCCR = H' 83
Clear the CRCDOR.
LSB first
Select X16+X12+X5+1 as the generating polynomial for CRC.
CRCDIR = rcv_dt [i] [ ]
Write the byte for the data block, the lower-order byte of the CRC code, and the higher-order byte of the CRC code to the CRCDIR.
→ The CRCDOR will hold the result of checking the data block for CRC errors.
errbuf [i] = CRCDOR
Save the result of checking for CRC errors in errbuf.
i ++
i = 0
i < 4?
No
Yes
errbuf [i] ≠ 0?
Judge whether CRC errors are or are not present.
End

5.7.4 sci0_init Function

1. Functional overview
   Initialization for asynchronous serial transfer
2. Arguments
   None
3. Return value
   None
4. Description of internal register used
   The internal registers used in this sample task are described below. The settings shown in these tables are the values used in this sample task and differ from the initial values.

- Port 2 input buffer control register (P2ICR) Number of bits: 8 Address: H'FFFB91

Bit	Bit Name	Setting	R/W	Description
1	P21ICR	1	R/W	0: Input buffer for pin P21 (RxD0) is disabled. The input signal is fixed to the high level. 1: Input buffer for pin P21 (RxD0) is enabled. The input signal reflects the state of the pin on the peripheral-module side.

- Serial mode register 0 (SMR_0) Number of bits: 8 Address: H'FFFF80

Bit	Bit Name	Setting	R/W	Description
7	C/$\overline{A}$	0	R/W	Communications Mode 0: Asynchronous 1: Clock synchronous
6	CHR	0	R/W	Character Length (only valid in asynchronous mode) 0: Transmission is with a data length of eight bits. 1: Transmission is with a data length of seven bits.
5	PE	0	R/W	Parity Enable 0: No parity bit 1: Include parity bit
3	STOP	0	R/W	Stop Bit Length Selects the stop bit length in transmission. 0: One stop bit 1: Two stop bits In reception, only the first stop bit is checked regardless of the value of this bit. If the second stop bit is 0, it is treated as the start bit of the next frame to be transmitted.
1	CKS1	0	R/W	Clock Select 1, 0
0	CKS0	0	R/W	These bits select the clock source for the baud rate generator. 00: Pϕ clock (n = 0) For the relation between the settings of these bits and the baud rate, see section 14.3.9, Bit Rate Register (BRR) in the hardware manual. n is the decimal display of the value of n in BRR (see section 14.3.9, Bit Rate Register in the hardware manual).

- Bit rate register 0 (BRR_0) Number of bits: 8 Address: H'FFFF81

  Function: BRR_0 is used to adjust the bit rate. For Pϕ: 32 MHz, CKS1 and 0 in SMR_0 = B'00, and BRR_0 = 25, the bit rate will be set to 38,400 bps.

  Setting: 25

- Serial control register 0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable 0: TXI interrupt requests disabled 1: TXI interrupt requests enabled
6	RIE	0	R/W	Receive Interrupt Enable 0: RXI and ERI interrupt requests disabled 1: RXI and ERI interrupt requests enabled
5	TE	0	R/W	Transmit Enable 0: Transmission disabled 1: Transmission enabled
4	RE	0	R/W	Receive Enable 0: Reception disabled 1: Reception enabled
2	TEIE	0	R/W	Transmit End Interrupt Enable 0: TEI interrupt requests disabled 1: TEI interrupt requests enabled
1 0	CKE1 CKE0	0 0	R/W	Clock Enable 1, 0 Selection of the clock source 00: Internal baud-rate generator 1x: TMR clock input or the average transfer rate generator Legend

Note X: Don't care.

- Serial status register 0 (SSR_0) Number of bits: 8 Address: H'FFFF84

Bit	Bit Name	Setting	R/W	Description
7	TDRE	Undefined	R/(W)*	Transmit Data Register Empty Indicates whether or not the TDR contains data for transmission. [Setting conditions] • Clearing of the TE bit in SCR to 0 • Transfer of latest data from TDR to TSR [Clearing conditions] • Writing 0 to TDRE after having read it as 1 (when interrupts are in use and TDRE is cleared by the CPU, be sure to read the flag after writing 0 to the bit). • Generation of a TXI interrupt causing the DMAC to transfer data for transmission to the TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether received data stored in the RDR is valid or invalid. [Setting condition] • Normal end of serial reception and transfer of received data from RSR to RDR [Clearing conditions] • Writing 0 to RDRF after having read it as 1 (when interrupts are in use and RDRF is cleared by the CPU, be sure to read the flag after writing 0 to RDRF). • When an RXI interrupt has caused the DMAC to transfer received data from the RDR, the RDRF flag is not affected and retains its previous value although the RE bit in SCR has been cleared to 0. Note that completion of the next round of serial reception while the RDRF flag remains set to one constitutes an overrun error, and the received data will be lost.
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • Occurrence of an overrun error during reception [Clearing condition] • Writing 0 to the bit after having read it as one (when interrupts are in use and ORER is cleared by the CPU, be sure to read the flag after 0 zero to the bit).
4	FER	0	R/(W)*	Framing Error [Setting condition] • Occurrence of a framing error during reception [Clearing condition] • Writing 0 to FER after having read it as one (when interrupts are in use and FER is cleared by the CPU, be sure to read the flag after writing zero to the bit).

Note: * Only 0 can be written here, to clear the flag.

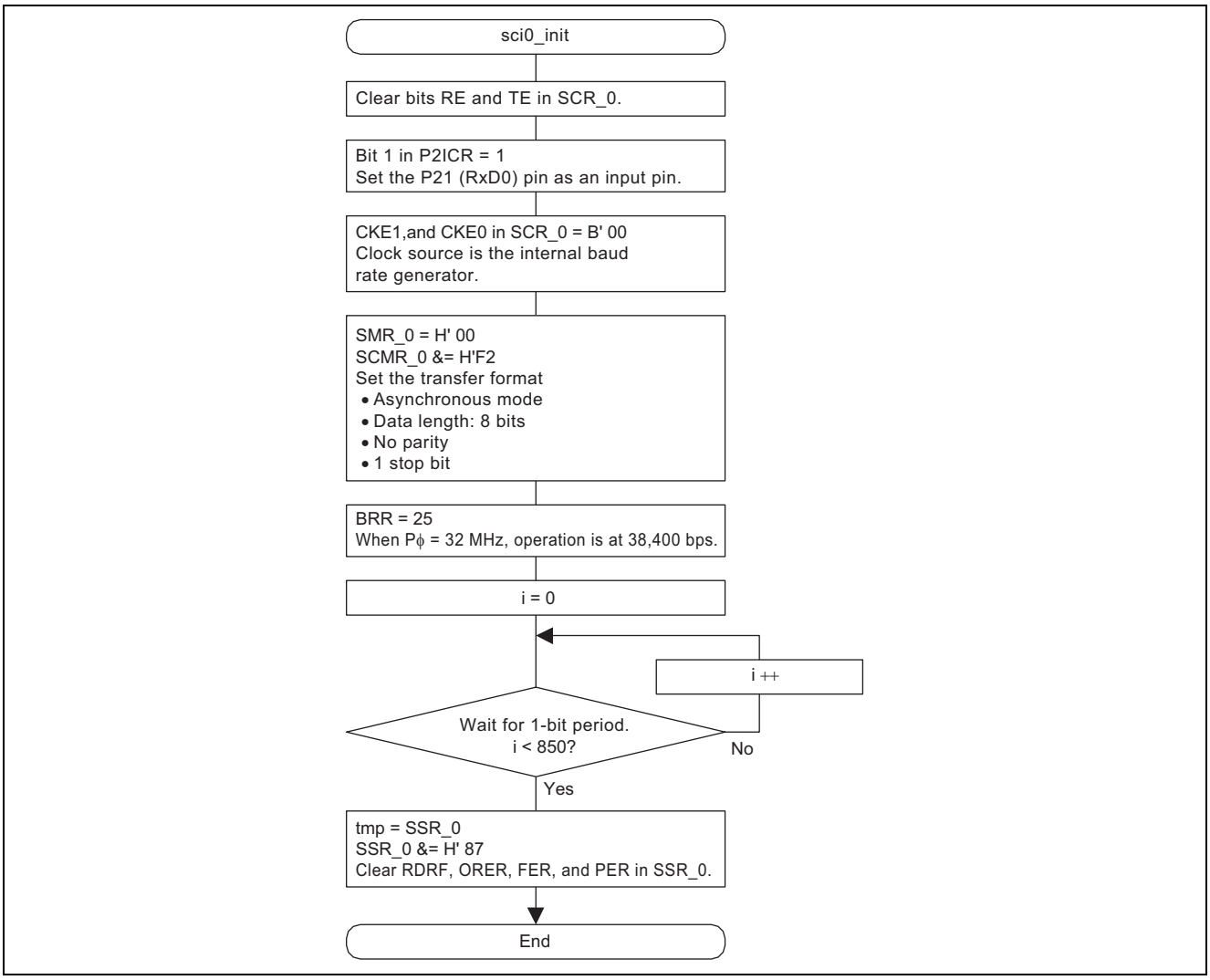
Bit	Bit Name	Setting	R/W	Description
3	PER	0	R/(W)*	Parity Error [Setting condition] • Occurrence of a parity error during reception [Clearing condition] • Writing 0 to PER after having read it as 1 (when interrupts are in use and PER is cleared by the CPU, be sure to read the flag after writing zero to the bit).
2	TEND	Undefined	R	Transmit End [Setting conditions] • Clearing of the TE bit in SCR to 0 • TDRE = 1 on transmission of the last bit of a character [Clearing conditions] • Writing 0 to the TDRE flag after having read the TDRE bit as 1 • Generation of a TXI interrupt causing the DMAC to write data for transmission to the TDR

Note: * Only 0 can be written here, to clear the flag.

- Smart card mode register 0 (SCMR_0) Number of bits: 8 Address: H'FFFF86

Bit	Bit Name	Setting	R/W	Description
0	SMIF	0	R/W	Smart Card Interface Mode Select 0: Normal asynchronous or clock synchronous operation 1: Operation in smart-card interface mode

5. Flowchart


sci0_init
Clear bits RE and TE in SCR_0.
Bit 1 in P2ICR = 1
Set the P21 (RxD0) pin as an input pin.
CKE1,and CKE0 in SCR_0 = B' 00
Clock source is the internal baud rate generator.
SMR_0 = H' 00
SCMR_0 &= H'F2
Set the transfer format
• Asynchronous mode
• Data length: 8 bits
• No parity
• 1 stop bit
BRR = 25
When Pϕ = 32 MHz, operation is at 38,400 bps.
i = 0
i ++
Wait for 1-bit period.
i < 850?
No
Yes
tmp = SSR_0
SSR_0 &= H' 87
Clear RDRF, ORER, FER, and PER in SSR_0.
End

5.7.5 Function sci0_rcv1byte

1. Functional overview
   Receives one byte of asynchronous serial data
2. Arguments
   None
3. Return value

Type	Description
unsigned char	One byte of received data

4. Description of internal registers used
   The internal registers used in this sample task are described below. The settings shown in these tables are the values used in this sample task and differ from the initial values.

- Serial control register 0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
4	RE	0/1	R/W	Receive Enable 0: Reception disabled 1: Reception enabled

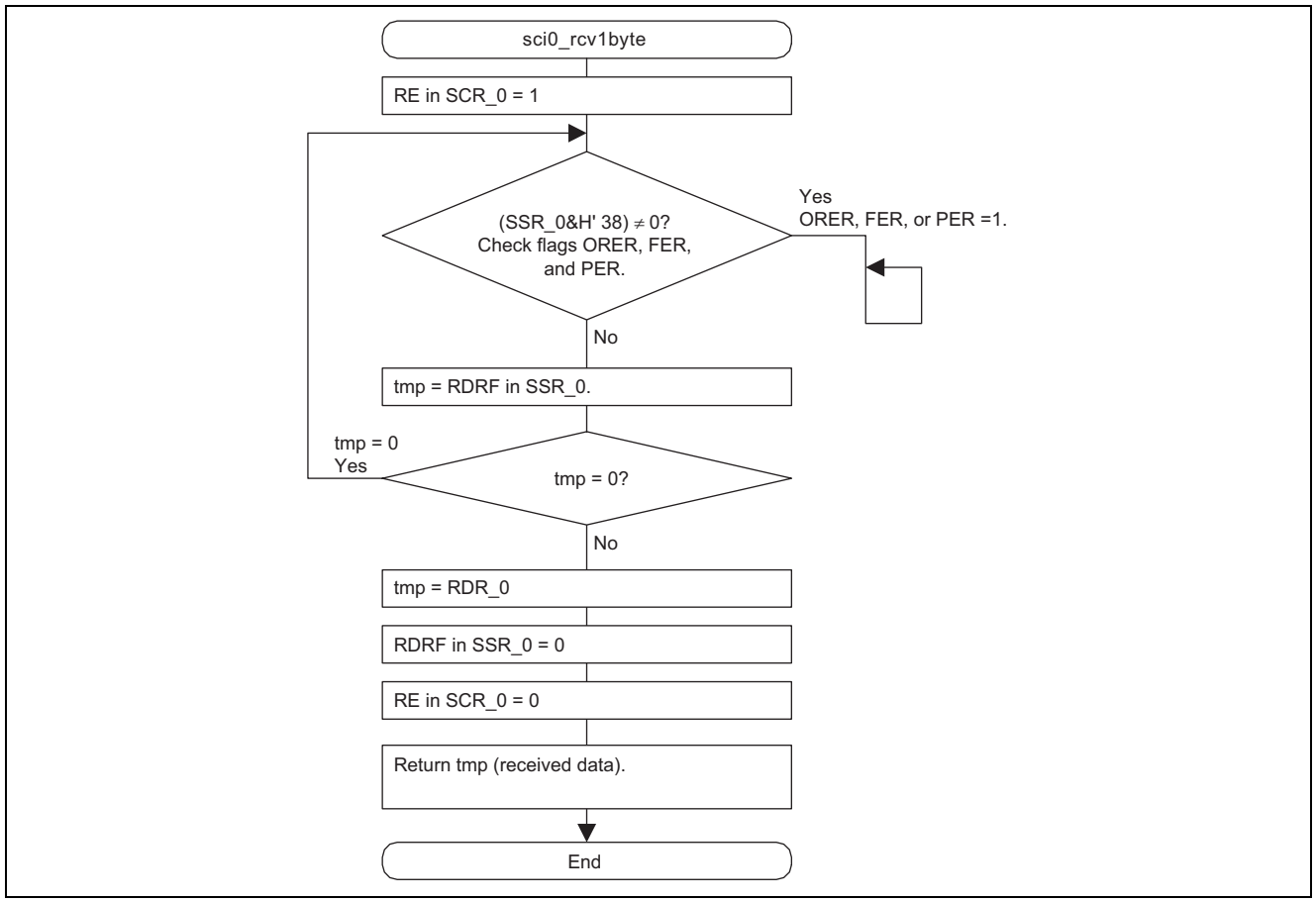
- Serial status register 0 (SSR_0) Number of bits: 8 Address: H'FFFF84

Bit	Bit Name	Setting	R/W	Description
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether received data stored in the RDR are valid or invalid. [Setting condition] • Normal end of serial reception and transfer of received data from RSR to RDR [Clearing conditions] • Writing 0 to RDRF after having read it as 1 (when interrupts are in use and the bit is cleared by the CPU, be sure to read the flag after writing 0 to the bit). • When an RXI interrupt has caused the DMAC to transfer received data from the RDR, the RDRF flag is not affected and retains its previous value although the RE bit in SCR has been cleared to 0. Note that completion of the next round of serial reception while the RDRF flag remains set to 1 constitutes an overrun error, and the received data will be lost.
5	ORER	Undefined	R/(W)*	Overrun Error [Setting condition] • Occurrence of an overrun error during reception [Clearing condition] • Writing 0 to ORER after having read it as 1 (when interrupts are in use and the ORER is cleared by the CPU, be sure to read the flag after writing 0 to the bit).
4	FER	Undefined	R/(W)*	Framing Error [Setting condition] • Occurrence of a framing error during reception [Clearing condition] • Writing 0 to FER after having read it as one (when interrupts are in use and FER is cleared by the CPU, be sure to read the flag after writing 0 to the bit).
3	PER	Undefined	R/(W)*	Parity Error [Setting condition] • Occurrence of a parity error during reception [Clearing condition] • Writing 0 to PER after having read it as 1 (when interrupts are in use and PER is cleared by the CPU, be sure to read the flag after writing 0 to the bit).

Note: * Only 0 can be written here, to clear the flag.

- Receive data register 0 (RDR_0) Number of bits: 8 Address: H'FFFF85
  Function: RDR_0 is an 8-bit register for the storage of received data.
  Setting: Undefined

5. Flowchart


sci0_rcv1byte
RE in SCR_0 = 1
(SSR_0&H' 38) ≠ 0?
Check flags ORER, FER, and PER.
Yes
ORER, FER, or PER =1.
No
tmp = RDRF in SSR_0.
tmp = 0
Yes
tmp = 0?
No
tmp = RDR_0
RDRF in SSR_0 = 0
RE in SCR_0 = 0
Return tmp (received data).
End

5.7.6 Function sci0_trs1byte

1. Functional overview
   Transmits one byte of asynchronous serial data
2. Arguments

Type	Variable Name	Description
unsigned char	tdt	One byte of data for transmission

3. Return value
   None
4. Description of internal registers used
   The internal registers used in this sample task are described below. The settings shown in these tables are the values used in this sample task and differ from the initial values.

- Serial control register 0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
5	TE	0/1	R/W	Transmit Enable 0: Transmission disabled 1: Transmission enabled

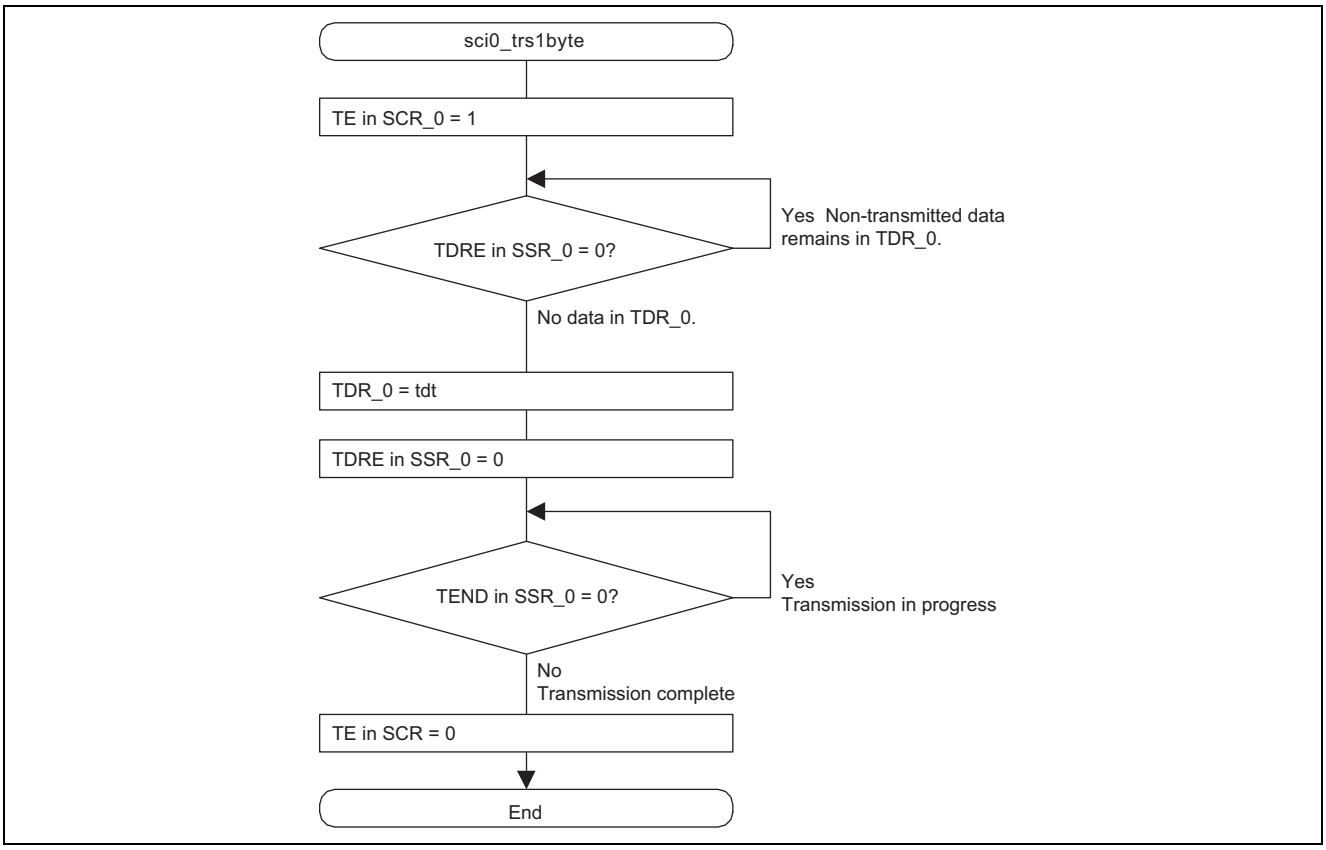
- Transmit data register 0 (TDR_0) Number of bits: 8 Address: H'FFFE8B
  Function: TDR_0 is an 8-bit register is for the storage of data for transmission.
  Setting: Undefined

- Serial status register 0 (SSR_0) Number of bits: 8 Address: H'FFFF84

Bit	Bit Name	Setting	R/W	Description
7	TDRE	0	R/(W)*	Transmit Data Register Empty Indicates whether or not the TDR contains data for transmission. [Setting conditions] • Clearing of the TE bit in SCR to 0 • Transfer of latest data from TDR to TSR [Clearing conditions] • Writing 0 to TDRE after having read it as 1 (when interrupts are in use and TDRE is cleared by the CPU, be sure to read the flag after writing 0 to the bit). • Generation of a TXI interrupt causing the DMAC to transfer data for transmission to the TDR
2	TEND	Undefined	R	Transmit End [Setting conditions] • Clearing of the TE bit in SCR to 0 • TDRE = 1 on transmission of the last bit of a character [Clearing conditions] • Writing 0 to the TDRE flag after having read the TDRE bit as 1 • Generation of a TXI interrupt causing the DMAC to write data for transmission to the TDR

Note: * Only 0 can be written here, to clear the flag.

5. Flowchart


sci0_trs1byte
TE in SCR_0 = 1
TDRE in SSR_0 = 0?
Yes Non-transmitted data remains in TDR_0.
No data in TDR_0.
TDR_0 = tdt
TDRE in SSR_0 = 0
TEND in SSR_0 = 0?
Yes
Transmission in progress
No
Transmission complete
TE in SCR = 0
End

6. Documents for Reference (Note)

- Hardware Manual
  H8SX/1653 Group Hardware Manual
  The most up-to-date version of this document is available on the Renesas Technology Website.

- Technical News/Technical Update
  The most up-to-date information is available on the Renesas Technology Website.

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Aug.23.07	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.