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

Using the DMAC to Drive Continuous SCI Transmission and Reception in Clock Synchronous Mode

Introduction

Clock synchronous transfer is used to transmit and receive 128 bytes of data. Using the DMAC to handle the transfer (transmission and reception) of data enables continuous transmission and reception with no CPU intervention.

Target Device

H8SX/1653

Contents

1. Specification

Clock synchronous transfer is used to transmit and receive 128 bytes of data. Using the DMAC to handle the transfer (transmission and reception) of data enables continuous transmission and reception with no CPU intervention.

- An example of connection for this sample task is shown in figure 1.
- Table 1 shows the communications format.
- After a power-on reset of the master side, the SCI and DMAC modules are set up and the state of pin P13 is polled. When the P13 pin is at the high level, the master side judges that transfer has been enabled at the slave side, after which operations for the clock synchronous transmission and reception of 128 bytes of data proceed.
- After a power-on reset of the slave side, the SCI and DMAC functions are set up. The same side outputs a high-level signal on pin P13, after which operations for the clock synchronous transmission and reception of 128 bytes of data proceed in synchronization with the master-side clock input on the SCK pin.

In this sample task, the DMAC modules on each side are interrupt-activated to continuously handle transmission and reception of the 128 bytes of data.

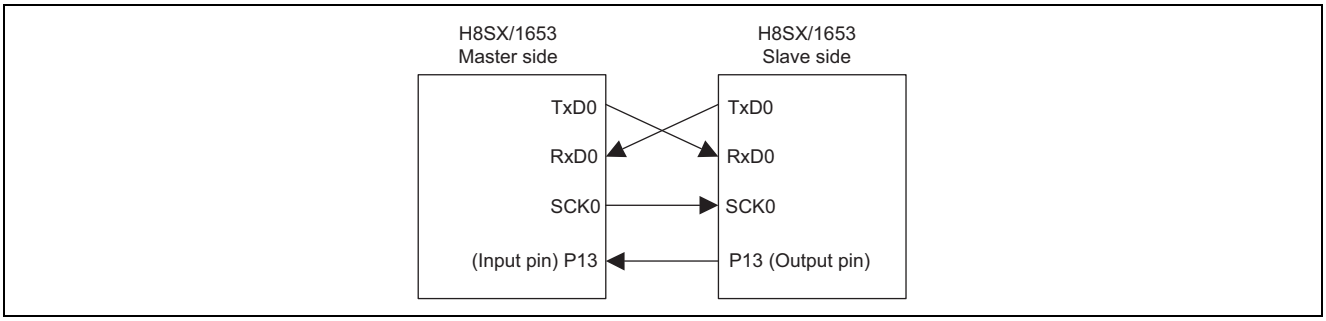


Figure 1 Clock Synchronous Serial Transmission and Reception

Table 1 Format for Clock Synchronous Serial Transmission and Reception

Format	Setting
Pϕ	32 MHz
Serial communications mode	Clock synchronous
Clock source	Master side: internal clock Slave side: external clock
Transfer rate	250 kbps
Data length	8 bits
Serial/parallel conversion format	LSB first

2. Applicable Conditions

Table 2 Applicable Conditions

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

3. Description of Modules Used

3.1 Description in Outline

Peripheral modules of the H8SX/1653 which are used in this sample task are shown in Figure 2.

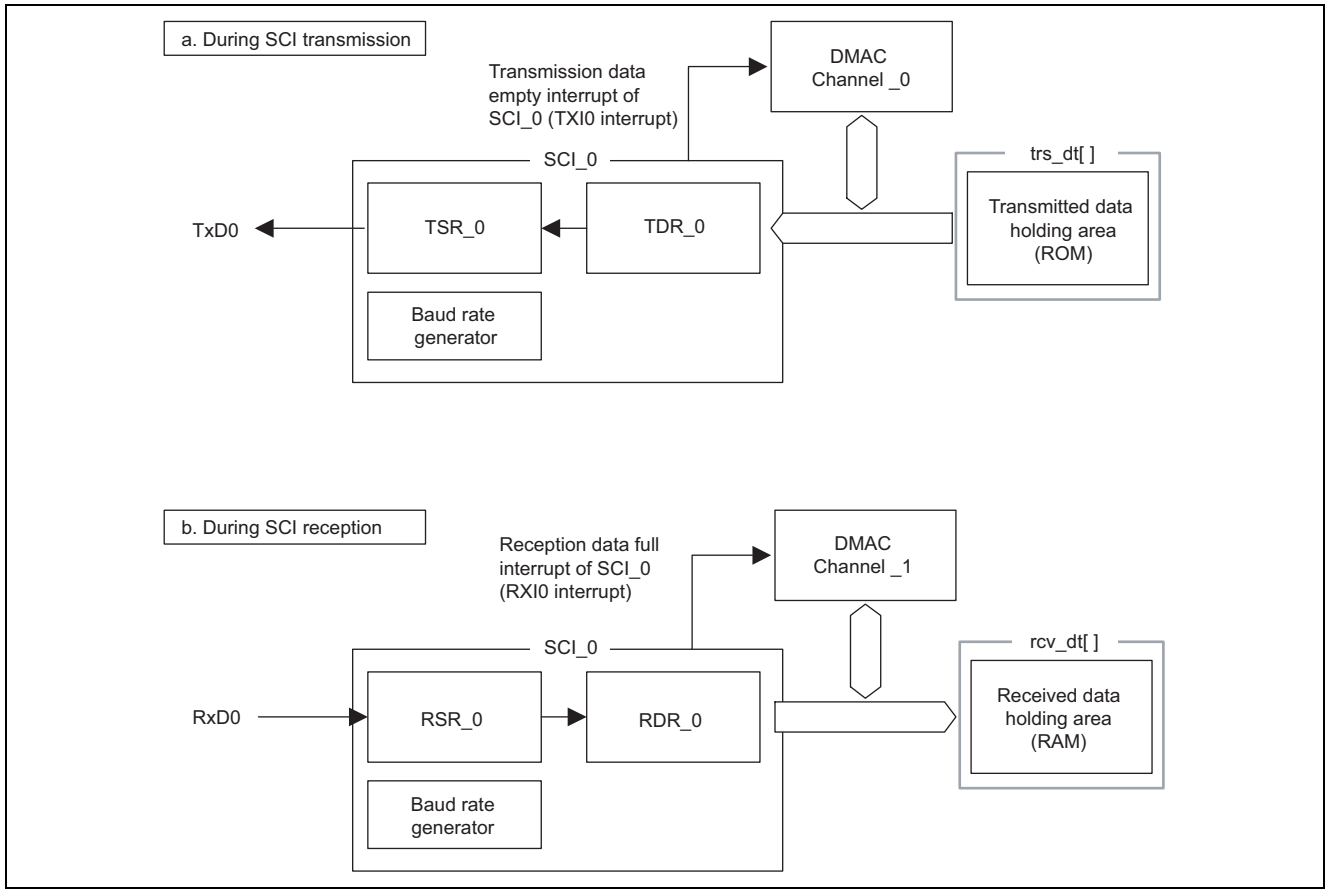


Figure 2 Functions of the H8SX/1653

The description which concerns the blocks shown in figure 2 is stated below.

1. SCI_0
   Transmits and receives data with timing provided by the clock synchronous communications.
   a. During SCI transmission
      - When TSR_0 is not full, data for transmission are written to TDR_0, transferred to TSR_0, and then output on TxD0 pin.
      - When the data are transferred from TDR_0 to TSR_0, a transmission data empty interrupt (TXI0 interrupt) from SCI_0 is generated.
   b. During SCI reception
      - After one frame of data has been received via the RxD0 pin, the received data are transferred from RSR_0 to RDR_0.
      - Once the data have been successfully received and then transferred from RSR_0 to RDR_0, a reception data full interrupt (RXI interrupt) from SCI_0 is generated.

2. DMAC channels 0 and 1
   a. During SCI transmission
      - Channel 0 is activated by the transmission data empty interrupt (TXI0 interrupt) from SCI_0 and transfers data from the area where data for transmission are stored to the TDR_0 register.
   b. During SCI reception
      - Channel 1 is activated by the received data full interrupt (RXI0 interrupt) from SCI_0 and transfers data from RDR_0 to the area where received data is to be stored.

3.2 Description of SCI_0

In this sample task, SCI_0 is used for clock synchronous serial data transmission and reception. Figure 3 is a block diagram of SCI_0.

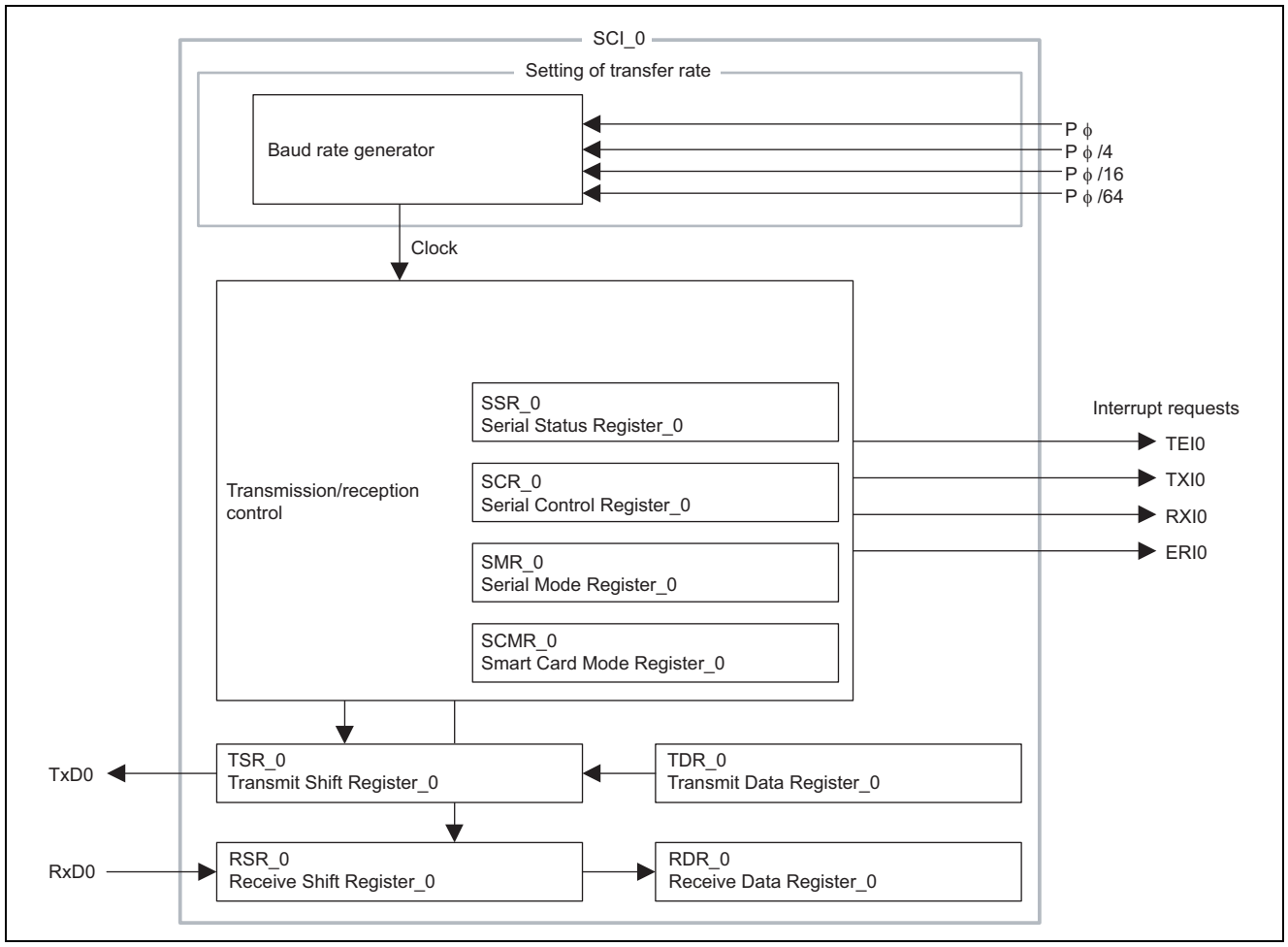


Figure 3 Block Diagram of SCI_0

The description which concerns the blocks shown in figure 3 is stated below.

- On-chip peripheral clock Pϕ
  This is the base clock for the operation of on-chip peripheral functions and is generated by a clock oscillator.
- Receive shift register_0 (RSR_0)
  RSR_0 is used to receive serial data. Serial data on RSR_0 are input via the RxD0 pin. When one frame of data has been received, the data bits are automatically transferred to the receive data register (RDR_0). RSR_0 is not accessible by the CPU.
- Receive data register_0 (RDR_0)
  RDR_0 is an 8-bit register and used to store received data. After RSR_0 has received one frame, the data bits are automatically transferred from RSR_0 to RDR_0. Since RSR_0 and RDR_0 function as a double buffer, continuous reception is possible. RDR_0 is for reception only, and so is seen as a read-only register by the CPU.
- Transmit shift register_0 (TSR_0)
  TSR_0 is used to transmit serial data. In transmission, data are transferred from the transmit data register (TDR_0) to TSR_0, and then output on the TxD0 pin. TSR_0 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. When SCI_0 detects that TSR_0 is empty, data that have been written to TDR_0 are automatically transferred to TSR_0. Since TDR_0 and TSR_0 function as a double buffer, if the next data for transmission has already been written to TDR_0 when one frame of data is transmitted, the written data are transferred to TSR_0. This allows continual transmission. Although TDR_0 can be read from or written to by the CPU at all times, only write data for transmission data after having confirmed setting of the TDRE bit in the serial status register (SSR_0) to 1.
- Serial mode register_0 (SMR_0)
  SMR_0 is an 8-bit register and used to select the format of serial data communications and the clock source for the on-chip baud-rate generator.
- Serial control register_0 (SCR_0)
  SCR_0 is used to control transmission, reception, and interrupts, and to select the clock source for transmission and reception.
- Serial status register_0 (SSR_0)
  SSR_0 consists of status flags for SCI_0 and multiprocessor bits for transmission and reception. TDRE, RDRF, ORER, PER, and FER can only be cleared.
- Smart card mode register_0 (SCMR_0)
  SCMR_0 is used to select the smart-card or normal interface mode for SCMR_0, and to set up the format for the smart-card mode. For this task, the setting in SCMR_0 selects the normal asynchronous or clock synchronous mode.
- Bit rate register_0 (BRR_0)
  BRR_0 is an 8-bit register that is used to adjust the bit rate.

3.3 Channels 0 and 1 of the DMAC

In this sample task, DMAC channel 0 is activated by the TXI0 interrupt of SCI_0 and DMAC channel 1 is activated by the RXI0 interrupt of SCI_0. A block diagram of the DMAC is given in figure 4.

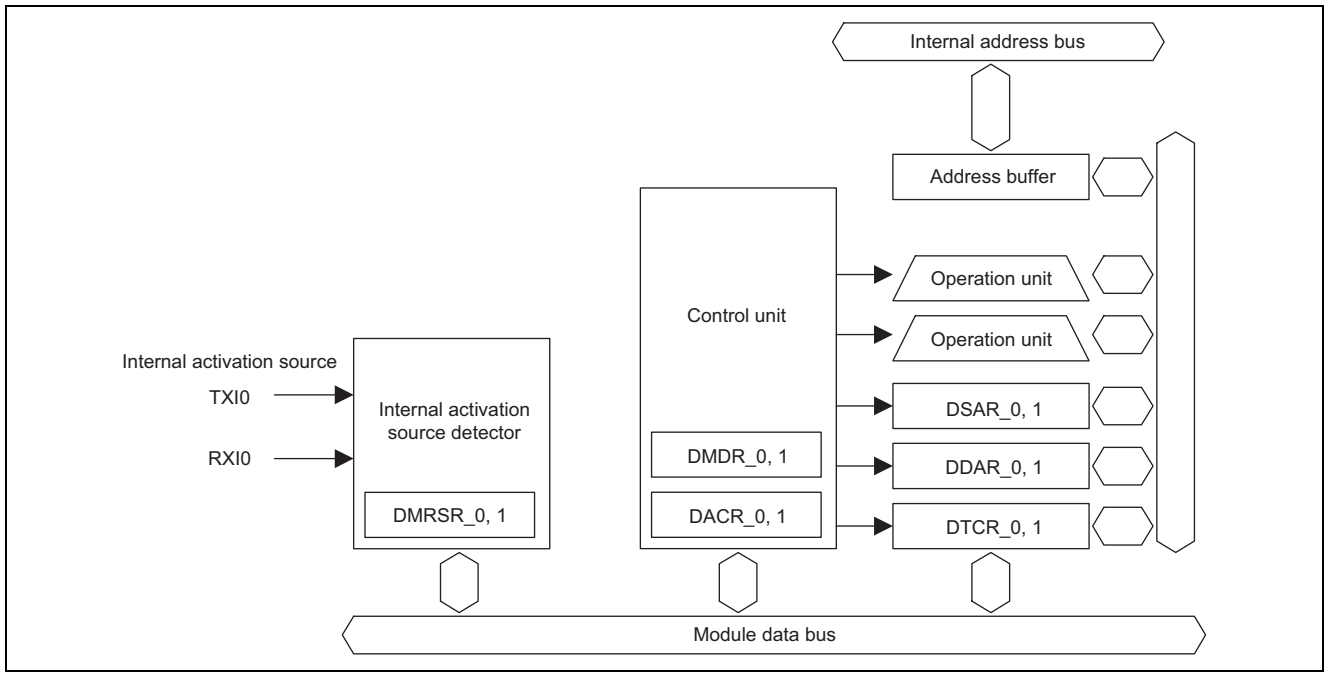


Figure 4 Block Diagram of the DMAC

The description with reference to figure 4 is stated below.

- DMA source address register _0 (DSAR_0)
- DMA source address register _1 (DSAR_1)
  DSARs are 32-bit readable/writable registers and specify the source address for the transfer. Each register is equipped with an address-updating function, so the source address is updated to that for the next transfer each time a transfer operation takes place.

- DMA destination address register _0 (DDAR_0)
- DMA destination address register _1 (DDAR_1)
  DDARs are 32-bit readable/writable registers and specify the destination address for the transfer. Each register is equipped with an address-updating function, so the destination address is updated to that for the next transfer each time a transfer operation takes place.

- DMA transfer count register _0 (DTCR_0)
- DMA transfer count register _1 (DTCR_1)
  DTCRs are 32-bit readable/writable registers and specify the amount of data to be transferred (total size for transfer). After each data transfer operation, the value is reduced by the amount that corresponds to the transferred amount of data. In this sample task, both are set for 128 bytes of data, and the byte is selected as the unit of data access. One is subtracted from the value on each DMAC operation, to indicate the amount still to be transferred.

- DMA mode control register _0 (DMDR_0)
- DMA mode control register _1 (DMDR_1)
  DMDRs control DMAC operation.

- DMA address control register_0 (DACR_0)
- DMA address control register_1 (DACR_1)
  DACRs set the operating mode and transfer method.

- DMA module request select register_0 (DMRSR_0)
- DMA module request select register_1 (DMRSR_1)
  DMRSRs set the activation source.

4. Principles of Operation

4.1 Outline

An outline of operation for this sample task is given in figure 5. 128-byte blocks of data are simultaneously transferred in both directions between the master and slave sides.

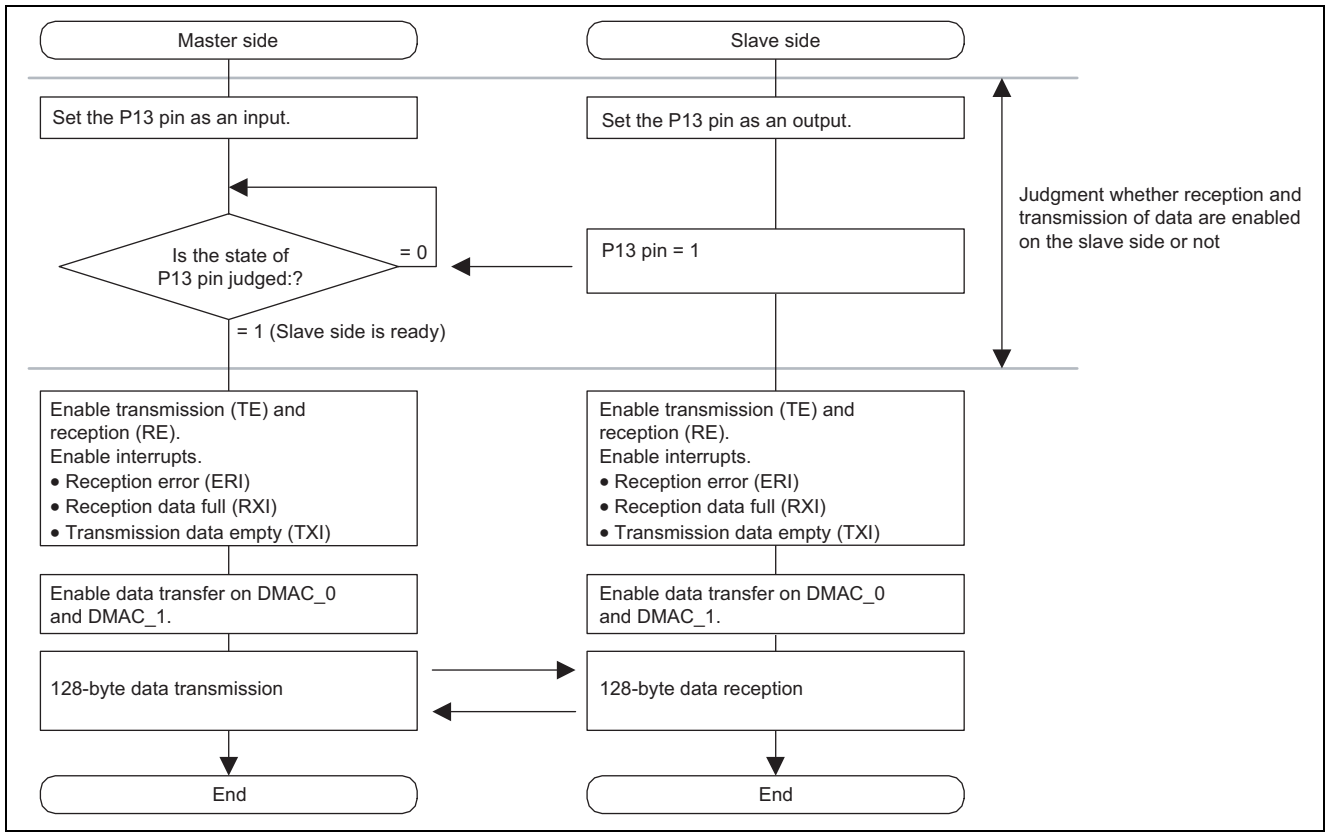


Figure 5 Outline of Operation

4.2 Transmission

4.2.1 Transmission Start

The timing of transmission start operations is illustrated in figure 6. Table 3 is a list of the hardware and software processing at the numbered points in figure 6.

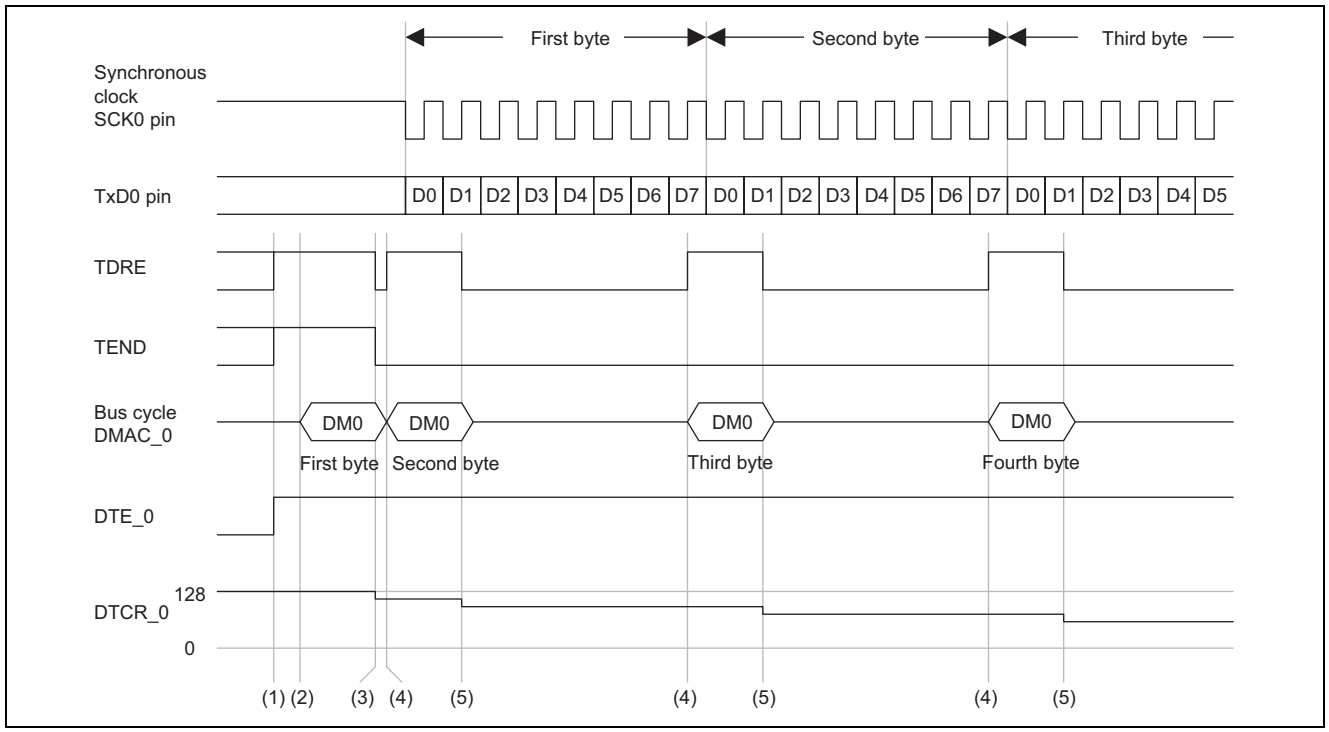


Figure 6 Timing of Transmission Start

Table 3 Processing

	Hardware Processing	Software Processing
(1)	Power-on reset	Initial settings*
(2)	a. Activate DMAC_0, and transfers data for transmission from the transmitted data holding area to TDR_0.	No processing
(3)	a. Clear TDRE to 0. b. DTCR_0 counts down. c. Transfer the contents of TDR_0 to TSR_0.	No processing
(4)	a. Set TDRE to 1. b. Activate DMAC_0 by TXI interrupt, and transfers data for transmission from the transmitted data holding area to TDR_0. c. Output the contents of data of TSR_0 on pin TxD0.	No processing
(5)	a. Clear TDRE to 0. b. DTCR_0 counts down. c. Transfer the contents of TDR_0 to TSR_0.	No processing

Notes: *Initial settings

DMAC_0 settings

a. Source for activation: TXI0 interrupt. The flag (TDRE) fro the TXI0 interrupt source is cleared on completion of the DMA transfer.
b. Source address: First address of the area where the data for transmission are stored. Incrementation is selected as the address incrementation or decrementation setting.
c. Destination address: Address of TDR_0. Fixed address is selected as the address incrementation or decrementation setting.
d. Total amount of transfer: 128 bytes
e. DMA data transfer is enabled (DTE_0 = 1).

SCR_0 settings

a. Clock synchronous mode. When Pϕ = 32 MHz, set the transfer rate of the master side to 250 kbps. Set clock source of the slave side to external clock.
b. TXI0 interrupt requests are enabled.
c. Set SCI_0 to enable transmit operations.

4.2.2 Transmission End

The timing of end of transmission operations is illustrated in figure 7. Table 4 is a list of the hardware and software processing at the numbered points in figure 7.

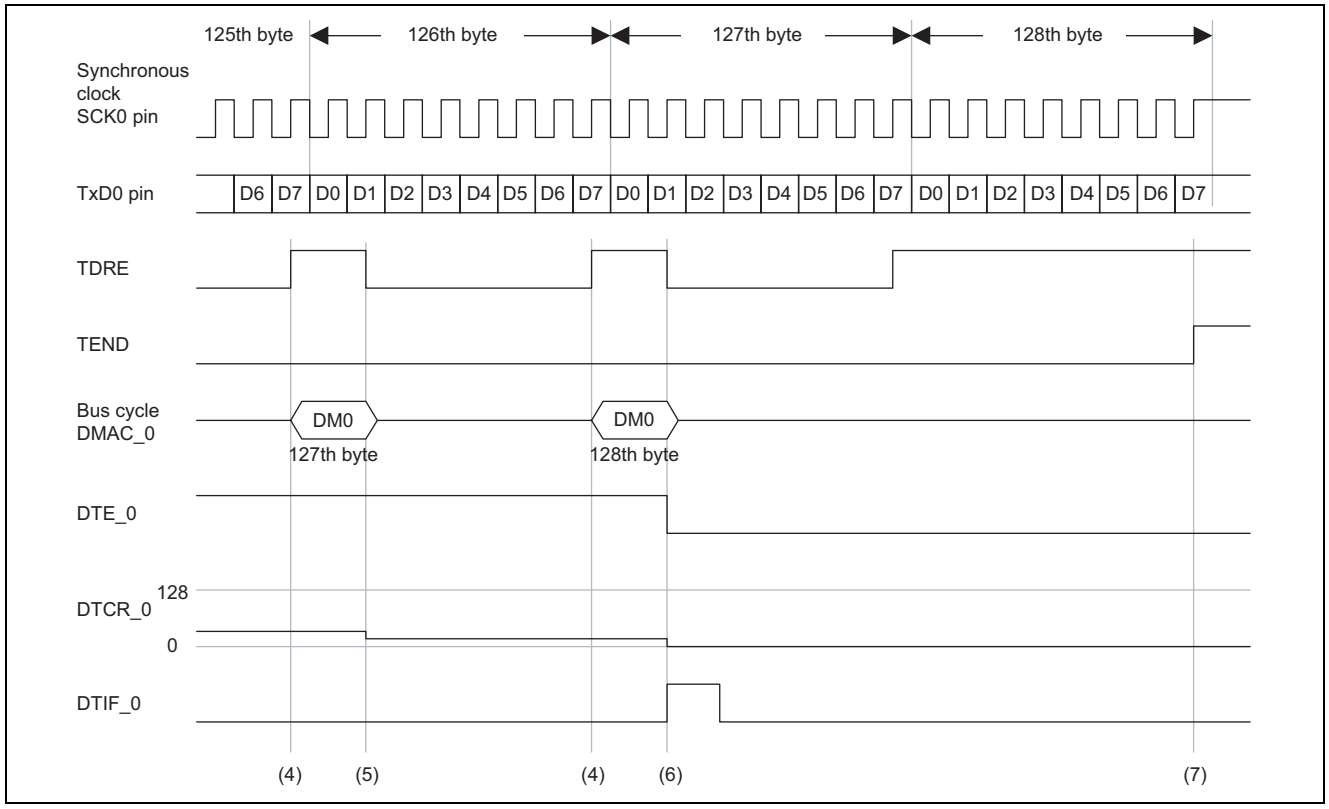


Figure 7 Timing of Transmission End

Table 4 Processing

	Hardware Processing	Software Processing
(4)	a. Set TDRE to 1. b. Activate DMAC_0 by TXI0 interrupt, and transfers data for transmission from the transmitted data holding area to TDR_0. c. Output the contents of TSR_0 on pin TxD0.	No processing
(5)	a. Clear TDRE to 0. b. DTCR_0 counts down. c. Transfer the contents of TDR_0 to TSR_0.	No processing
(6)	a. Clear TDRE to 0. b. DTCR_0 (DTCR_0 = 0) counts down. c. Transfer the contents of TDR_0 to TSR_0.	DMAC_0 transfer end interrupt a. Disable interrupt request. b. Disable DMAC_0 transfer end interrupt requests.
(7)	a. Set TEND to 1.	TEI0 Interrupt a. Stop SCI_0 transmission operations (TE = 0). b. Disable TEI0 interrupt requests.

4.3 Reception

The timing of reception operation is illustrated in figure 8. Table 5 is a list of the hardware and software processing at the numbered points in figure 8.

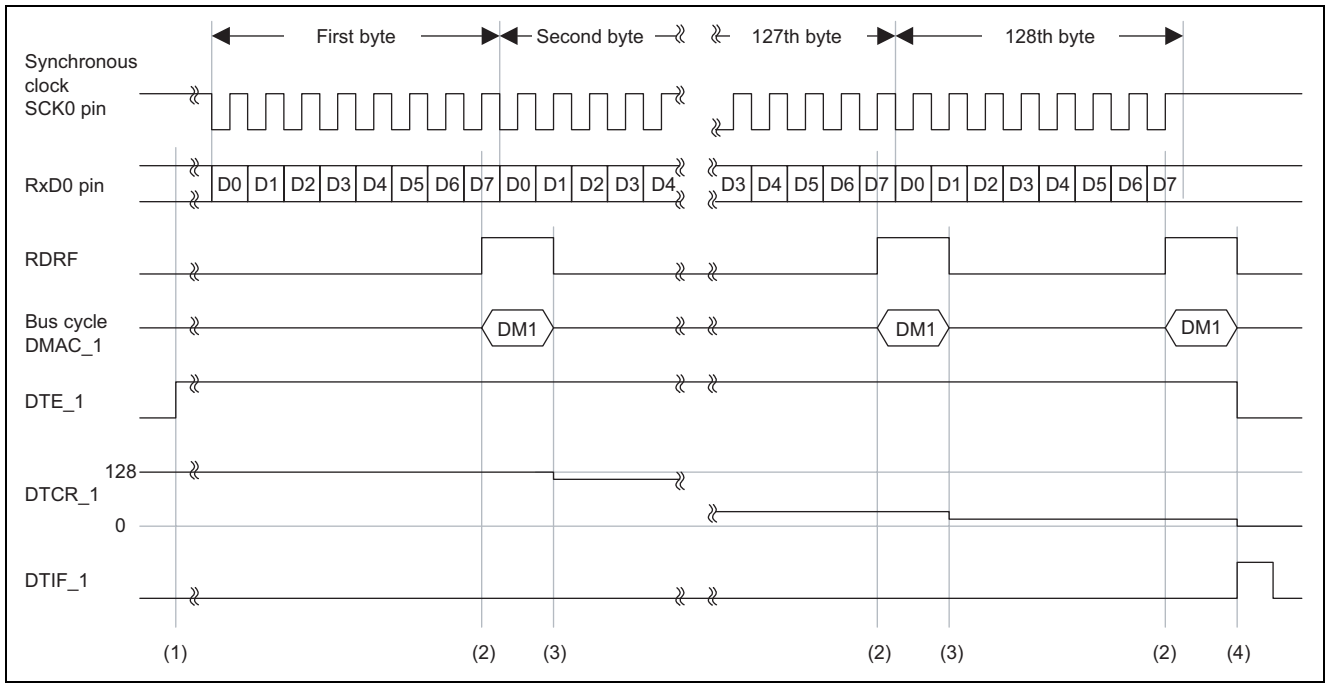


Figure 8 Timing of Reception

Table 5 Processing

	Hardware Processing	Software Processing
(1)	Power-on reset	Initial settings*
(2)	a. Set RDRF to 1. b. End transmission normally and transfer the received data from RSR_0 to RDR_0. c. Activate DMAC_1 and transfer the received data from RDR_0 to received data holding area.	No processing
(3)	a. Clear RDRF0 to 0. b. DTCR_1 counts down.	No processing
(4)	a. DTCR_0 counts down (DTCR_1 = 0).	DMAC_1 transfer end interrupt a. Disable reception of SCI_0 (RE = 0). b. Disable interrupt requests of RXI0 and ERI0. c. Disable DMAC_1 transfer end interrupt request.

Notes: *Initial settings

DMAC_1 settings

a. Source for activation: RXI0 interrupt. The flag (RDRF) for the RXI0 interrupt source is cleared on completion of the DMA transfer.
b. Source address: Address of RDR_0. Fixed address is selected as the address incrementation or decrementation.
c. Destination address: First address of the area where the received data are to be stored. Incrementation is selected as the address incrementation or decrementation setting.
d. Total amount of transfer: 128 bytes
e. DMA data transfer is enabled. (DTE_1 = 1).

SCR_0 settings

a. Clock synchronous mode. When Pϕ = 32 MHz, set the transfer rate of the master side to 250 kbps. Set clock source of the slave side to external clock.
b. RXI0 interrupt requests are enabled.
c. Set SCI_0 reception operations enabled.

5. Description of Software

5.1 Operating Environment

Table 6 Operating Environment

Item	Details
Development tool	High-performance Embedded Workshop Ver.4.01.01
C/C++ compiler	H8S, H8/300 Series C/C++ Compiler Ver.6.01.02 (manufactured by Renesas Technology)
Compiler options	-cpu = h8sxa:24:md, -code = machinecode, -optimize = 1, -regparam = 3, -speed = (register, shift, struct, expression)

Table 7 Section Setting

Address	Section Name	Description
H'001000	P	Program area
	C	Constant area
H'FF2000	B	Non-initialized data area (RAM area)

Table 8 Vector Table for Interrupt Exception Handling

Exception Handling Source		Vector No.	Vector Address	Function to Interrupt Destination
Reset		0	H'000000	init
DMAC_0	DMTEND0	128	H'000200	dmtend0_int
DMAC_1	DMTEND1	129	H'000204	dmtend1_int
SCI_0	ERI0	144	H'000240	eri0_int
SCI_0	TEI0	147	H'00024C	tei0_int

5.2 List of Functions

Table 9 lists the functions used in this sample task. Figure 9 shows the structure of hierarchy.

Table 9 List of Functions

Function Name	Description
init	Initialization routine: Takes the chip out of module stop mode, performs clock settings, and calls the main function.
main	Master side (MASTER) main routine Selects clock synchronous SCI, calls functions DMAC0_trs_init and DMAC1_rcv_init functions. Judges input of a high-level signal on the P13 pin. Makes settings for transmission and reception of 128 bytes of data.
	Slave side (SLAVE) main routine Selects clock synchronous SCI, calls the DMAC0_trs_init and DMAC1_rcv_init functions. Outputs a high-level signal on the P13 pin. Makes settings for transmission and reception of 128 bytes of data.
DMAC0_trs_init	DMAC_0 initialization Selects TXI0-interrupt-triggered processing of transfer from the area where data for transmission are stored to TDR_0.
DMAC1_rcv_init	DMAC_1 initialization Selects RXI0-interrupt-triggered processing of transfer from the area where received data are to be stored.
dmtend0_int	DMAC_0 transfer end interrupt Sets TEI0 interrupt request enabled, TXI0 interrupt and DMAC_0 transfer end interrupt requests disabled.
dmtend1_int	DMAC_1 transfer end interrupt Sets SCI_0 reception, RXI0 and ERI0 interrupt requests, and DMAC_1 transfer end interrupt requests disabled.
eri0_int	Reception error interrupt Writes error data to RAM and initializes SSR_0.
tei0_int	Transmission end interrupt Sets SCI_0 transmission and TEI0 interrupt requests disabled.

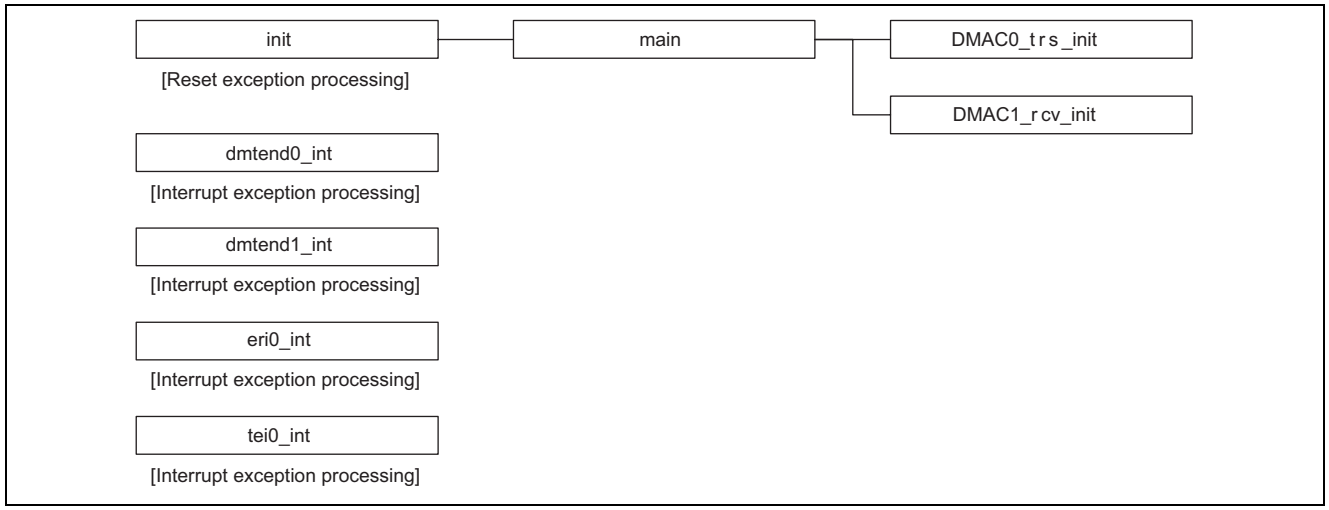


Figure 9 Hierarchy of Calls in the User Program

5.3 RAM Usage

Table 10 RAM Usage

Type	Variable Name	Description	Used in
unsigned char	endflg	Transmission end flag 0: Transmission in progress 1: Transmission ended	main, tei0_int
unsigned char	errbuf	Reception error buffer The contents of SSR_0 are stored when an overrun error occurs.	main, eri0_int
unsigned char	tcnt	Transmission counter	main, dmtend0_int
unsigned char	rcnt	Reception counter	main, dmtend1_int
unsigned char	rcv_dt[128]	Received data holding area (RAM)	main, DMAC1_rcv_int

5.4 Constant

Table 11 Constants

Type	Variable Name	Setting	Description	Used in
unsigned char	trs_dt[128]	H'00, H'01, H'02, … ……H'7E, H'7F	Transmitted data holding area (ROM)	main, DMAC0_trs_init

5.5 Macro Definitions

Table 12 Macro Definitions

Identifier	Description	Used in
MASTER	Generates the master-side program.	main
SLAVE	Generates the slave-side program.	main

5.6 Symbolic Constants

Table 13 Symbolic Constants

Constant Name	Setting	Description
NUM	128	Sets the number of data 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, makes clock settings, and calls the main function.
2. Argument
   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. This 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 the mode pins (MD2 to MD0) (see table 14). When MDCR is read, the signal levels input on pins MD2 to MD0 are latched into these bits. The latches are released by a reset.
9	MDS1	Undefined*	R	
8	MDS0	Undefined*	R	

Note: * Determined by the settings on pins MD3 to MD0.

Table 14 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 signal,
8	ICK0	1	R/W	which is provided to the CPU, DMAC, and DTC. 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
4	PCK0	1	R/W	clock. 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, MSTPCRB, and MSTPCRC control module stop mode. Setting a bit to 1 places the corresponding module in 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	All-Module-Clock-Stop Mode Enable This bit enables/disables all-module-clock-stop mode for reducing current consumption by stopping the bus controller and I/O port operation when the CPU executes the SLEEP instruction after 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	0	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, TMR_2)
8	MSTPA8	1	R/W	8-bit timer unit (TMR_1, 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)

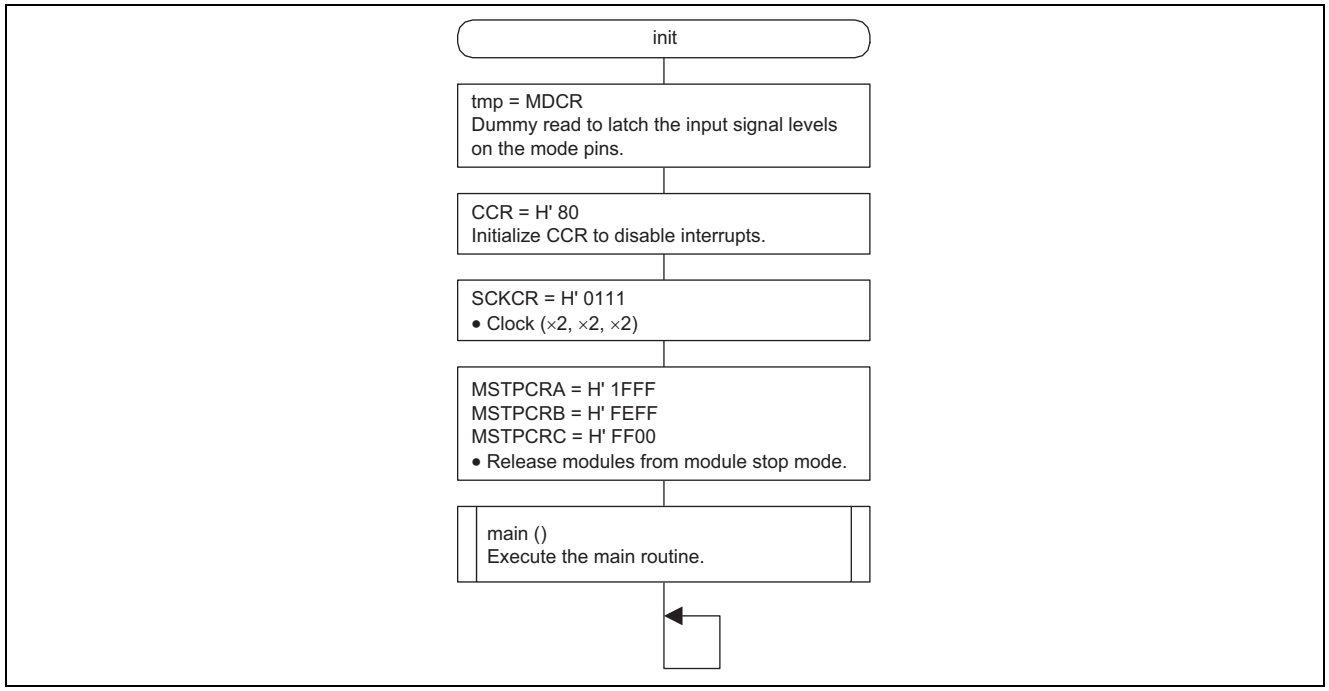
- Module stop control register B (MSTPCRB) Number of bits: 16 Address: H'FFFDCA

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	MSTPC15	1	R/W	Serial communications interface_5 (SCI_5), (IrDA)
14	MSTPC14	1	R/W	Serial communications interface_6 (SCI_6)
13	MSTPC13	1	R/W	8-bit timer unit (TMR_4, TMR_5)
12	MSTPC12	1	R/W	8-bit timer unit (TMR_6, TMR_7)
11	MSTPC11	1	R/W	Universal serial bus interface (USB)
10	MSTPC10	1	R/W	Cyclic redundancy check module
4	MSTPC4	0	R/W	On-chip RAM_4 (H'FF2000 to H'FF3FFF)
3	MSTPC3	0	R/W	On-chip RAM_3 (H'FF4000 to H'FF5FFF)
2	MSTPC2	0	R/W	On-chip RAM_2 (H'FF6000 to H'FF7FFF)
1	MSTPC1	0	R/W	On-chip RAM_1 (H'FF8000 to H'FF9FFF)
0	MSTPC0	0	R/W	On-chip RAM_0 (H'FFA000 to H'FFBFFF)

5. Flowchart



5.7.2 Master-Side (MASTER) main Function

1. Functional overview
   Main routine: Sets the clock synchronous SCI, calls functions DMAC0_trs_init and DMAC1_rcv_init, judges input of high-level signal on pin P13, and sets transmission and reception of 128 byte-data.
2. Argument
   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.

- Port 1 data direction register (P1DDR) Number of bits: 8 Address: Address H'FFFB80

Bit	Bit Name	Setting	R/W	Description
3	P13DDR	0	W	0: Sets pin P13 as an input. 1: Sets pin P13 as an output.

- 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: P21 pin input buffer is disabled. Input signal is fixed to the high level. 1: P21 pin input buffer is enabled. The pin state reflects the peripheral modules.

- DMA mode control register_0 (DMDR_0) Number of bits: 32 Address: H'FFFC14

Bit	Bit Name	Setting	R/W	Description
31	DTE	1	R/W	Data Transfer Enable 0: Disables data transfer. 1: Enables data transfer.

- DMA mode control register_1 (DMDR_1) Number of bits: 32 Address: H'FFFC34

Bit	Bit Name	Setting	R/W	Description
31	DTE	1	R/W	Data Transfer Enable 0: Disables data transfer. 1: Enables data transfer.

- Port 1 register (PORT1) Number of bits: 8 Address: H'FFFF40

Bit	Bit Name	Setting	R/W	Description
3	P13PORT	Undefined	R	0: Pin P13 is set to a low level. 1: Pin P13 is set to a high level.

- Serial mode control register_0 (SMR_0) Number of bits: 8 Address: H'FFFF80

Bit	Bit Name	Setting	R/W	Description
7	C/$\overline{A}$	1	R/W	Communication Mode 0: Asynchronous 1: Clock synchronous
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. When Pϕ = 32 MHz, CKS1 and CKS2 in SMR_0 = B'00, and BRR_0 = 31 will set the bit rate to 250 kbps.

  Setting: 31

- Serial control register_0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
7	TIE	0/1	R/W	Transmit Interrupt Enable 0: Disables TXI interrupt requests. 1: Enables TXI interrupt requests.
6	RIE	0/1	R/W	Receive Interrupt Enable 0: Disables RXI and ERI interrupt requests. 1: Enables RXI and ERI interrupt requests.
5	TE	0/1	R/W	Transmit Enable 0: Disables transmission. 1: Enables transmission.
4	RE	0/1	R/W	Receive Enable 0: Disables reception. 1: Enables reception.
2	TEIE	0/1	R/W	Transmit End Interrupt Enable 0: Disables TEI interrupt requests. 1: Enables TEI interrupt requests.
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0		Selects the clock source. When in clock synchronous mode; 0X: Internal clock. SCK pin is set as a clock output pin. 1X: External clock. SCK pin is set as a clock input pin.

Note X: Don't care.

- Serial status register_0 (SSR_0) Number of bits: 8 Address: H'FFFF84

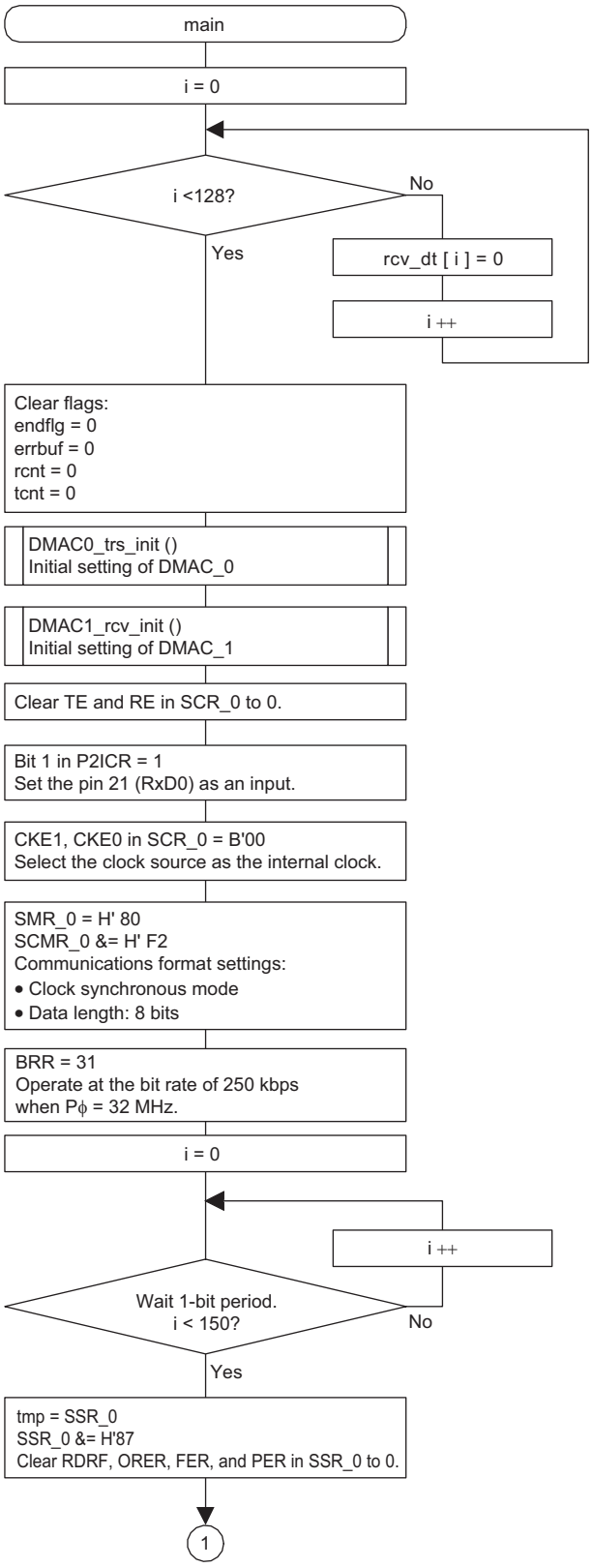
Bit	Bit Name	Setting	R/W	Description
7	TDRE	Undefined	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains data for transmission. [Setting conditions] • Clearing of the TE bit in SCR to 0 • Transfer of data from TDR to TSR [Clearing conditions] • Writing of 0 to TDRE after having read TDRE = 1 (when using an interrupt and having the CPU clear it, be sure to read the flag after having written 0 to it). • Generation of a TXI interrupt request allowing DMAC to write transmit data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether RDR holds received data. [Setting condition] • The normal end of serial reception and the transfer of received data from RSR to RDR [Clearing conditions] • Writing of 0 to RDRF after having read RDRF = 1 (when using an interrupt and having the CPU clear it, be sure to read the flag after having written 0 to it). • Generation of an RXI interrupt request allowing DMAC or DTC to read data from RDR. The RDRF flag is not affected and retains its previous value even though the RE bit in SCR is cleared to 0. Note that when the next reception is completed while the RDRF flag is being set to 1, an overrun error occurs and the received data are lost.
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • Occurrence of an overrun error during reception [Clearing condition] • Writing of 0 to ORER after having read ORER = 1 (when using an interrupt and having the CPU clear it, be sure to read the flag after having written 0 to it).
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 of 0 to TDRE after having read TDRE = 1 • Generation of a TXI interrupt request allowing its value DMAC to write data to TDR

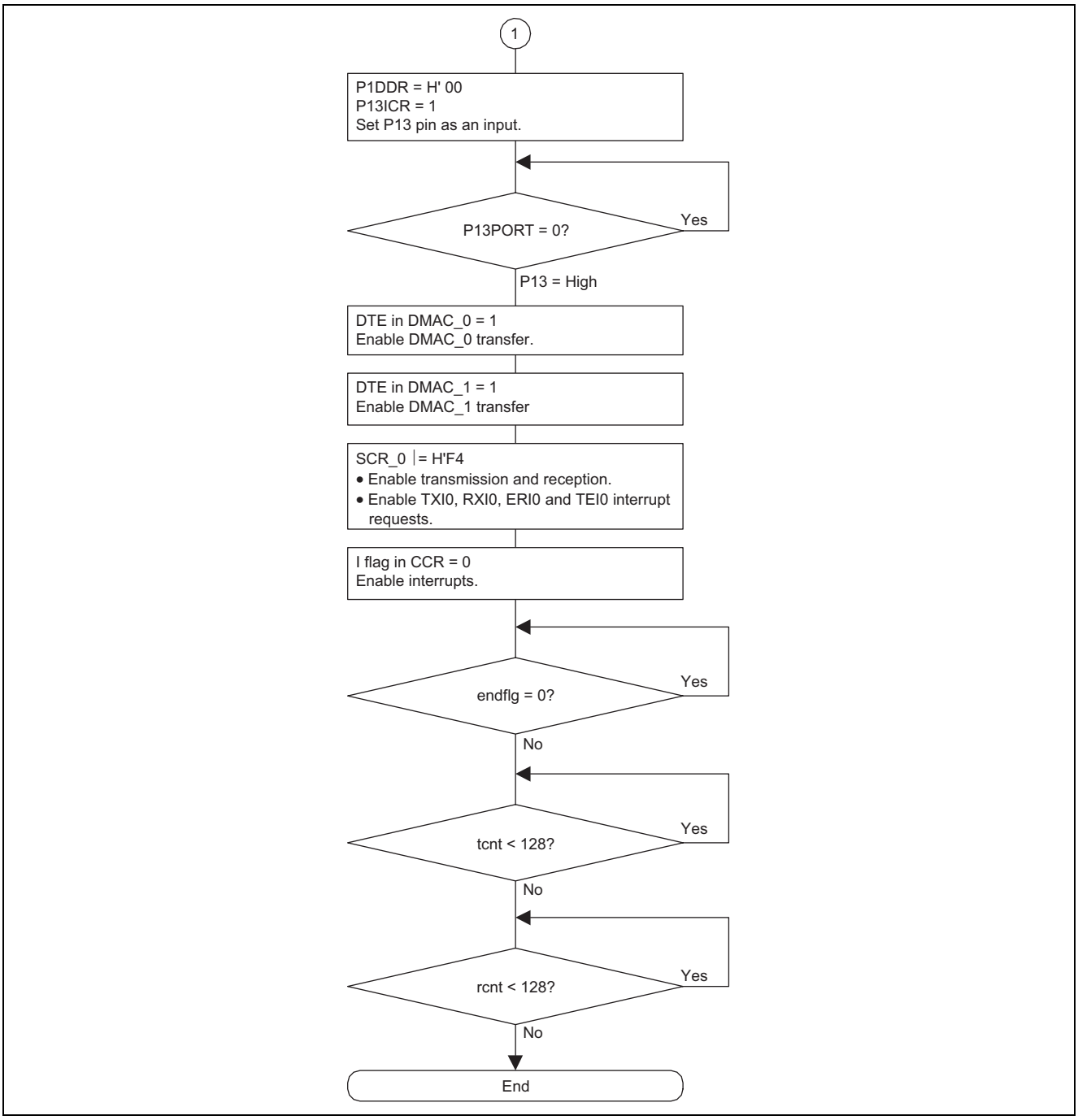
Note: * Only 0 can be written here, to clear the flag.

- 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: Operation is in the normal asynchronous or clock synchronous mode. 1: Operation is in smart card interface mode.

5. Flowchart


main
i = 0
i <128?
No
Yes
rcv_dt [i] = 0
i ++
Clear flags:
endflg = 0
errbuf = 0
rcnt = 0
tcnt = 0
DMAC0_trs_init ()
Initial setting of DMAC_0
DMAC1_rcv_init ()
Initial setting of DMAC_1
Clear TE and RE in SCR_0 to 0.
Bit 1 in P2ICR = 1
Set the pin 21 (RxD0) as an input.
CKE1, CKE0 in SCR_0 = B'00
Select the clock source as the internal clock.
SMR_0 = H' 80
SCMR_0 &= H' F2
Communications format settings:
• Clock synchronous mode
• Data length: 8 bits
BRR = 31
Operate at the bit rate of 250 kbps
when Pϕ = 32 MHz.
i = 0
i ++
Wait 1-bit period.
i < 150?
No
Yes
tmp = SSR_0
SSR_0 &= H'87
Clear RDRF, ORER, FER, and PER in SSR_0 to 0.
1


1
P1DDR = H' 00
P13ICR = 1
Set P13 pin as an input.
P13PORT = 0?
Yes
P13 = High
DTE in DMAC_0 = 1
Enable DMAC_0 transfer.
DTE in DMAC_1 = 1
Enable DMAC_1 transfer
SCR_0 |= H'F4
• Enable transmission and reception.
• Enable TXI0, RXI0, ERI0 and TEI0 interrupt requests.
I flag in CCR = 0
Enable interrupts.
endflg = 0?
Yes
No
tcnt < 128?
Yes
No
rcnt < 128?
Yes
No
End

5.7.3 Slave-Side (SLAVE) main Function

1. Functional overview
   Main routine: Sets the clock synchronous SCI, calls functions DMAC0_trs_init and DMAC1_rcv_init, outputs high level signal on pin P13, and sets transmission and reception of 128-byte data.
2. Argument
   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.

- Port 1 data direction register (P1DDR) Number of bits: 8 Address: H'FFFB80

Bit	Bit Name	Setting	R/W	Description
3	P13DDR	1	W	0: Sets pin P13 as an input. 1: Sets pin P13 as an output.

- 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: P21 pin (RxD0) input buffer is disabled. Input signal is fixed to the high level. 1: P21 pin (RxD0) input buffer is enabled. The pin state reflects the peripheral modules.
0	P20ICR	1	R/W	0: P20 pin (SCK0) input buffer is disabled. Input signal is fixed to the high level. 1: P20 pin (SCK0) input buffer is enabled. The pin state reflects the peripheral modules.

- DMA mode control register_0 (DMDR_0) Number of bits: 32 Address: H'FFFC14

Bit	Bit Name	Setting	R/W	Description
31	DTE	0	R/W	Data Transfer Enable 0: Disables data transfer. 1: Enables data transfer.

- DMA mode control register_1 (DMDR_1) Number of bits: 32 Address: H'FFFC34

Bit	Bit Name	Setting	R/W	Description
31	DTE	0	R/W	Data Transfer Enable 0: Disables data transfer. 1: Enables data transfer.

- Port 1 data register (P1DR) Number of bits: 8 Address: H'FFFF50

Bit	Bit Name	Setting	R/W	Description
3	P13DR	0/1	R/W	0: Pin P13 is set to a low level. 1: Pin P13 is set to a high level.

- Serial mode control register_0 (SMR_0) Number of bits: 8 Address: H'FFFF80

Bit	Bit Name	Setting	R/W	Description
7	C/$\overline{A}$	1	R/W	Communication Mode 0: Asynchronous 1: Clock synchronous
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).

- Serial control register_0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
7	TIE	0/1	R/W	Transmit Interrupt Enable 0: Disables TXI interrupt requests. 1: Enables TXI interrupt requests.
6	RIE	0/1	R/W	Receive Interrupt Enable 0: Disables RXI and ERI interrupt requests. 1: Enables RXI and ERI interrupt requests.
5	TE	0/1	R/W	Transmit Enable 0: Disables transmission. 1: Enables transmission.
4	RE	0/1	R/W	Receive Enable 0: Disables reception. 1: Enables reception.
2	TEIE	0/1	R/W	Transmit End Interrupt Enable 0: Disables TEI interrupt requests. 1: Enables TEI interrupt requests.
1	CKE1	1	R/W	Clock Enable 1 and 0
0	CKE0	0		Selects the clock source. When in clock synchronous mode; 0X: Internal clock. SCK pin is set as a clock output pin. 1X: External clock. SCK pin is set as a clock input pin.

Note X: Don't care.

- Serial status register_0 (SSR_0) Number of bits: 8 Address: H'FFFF84

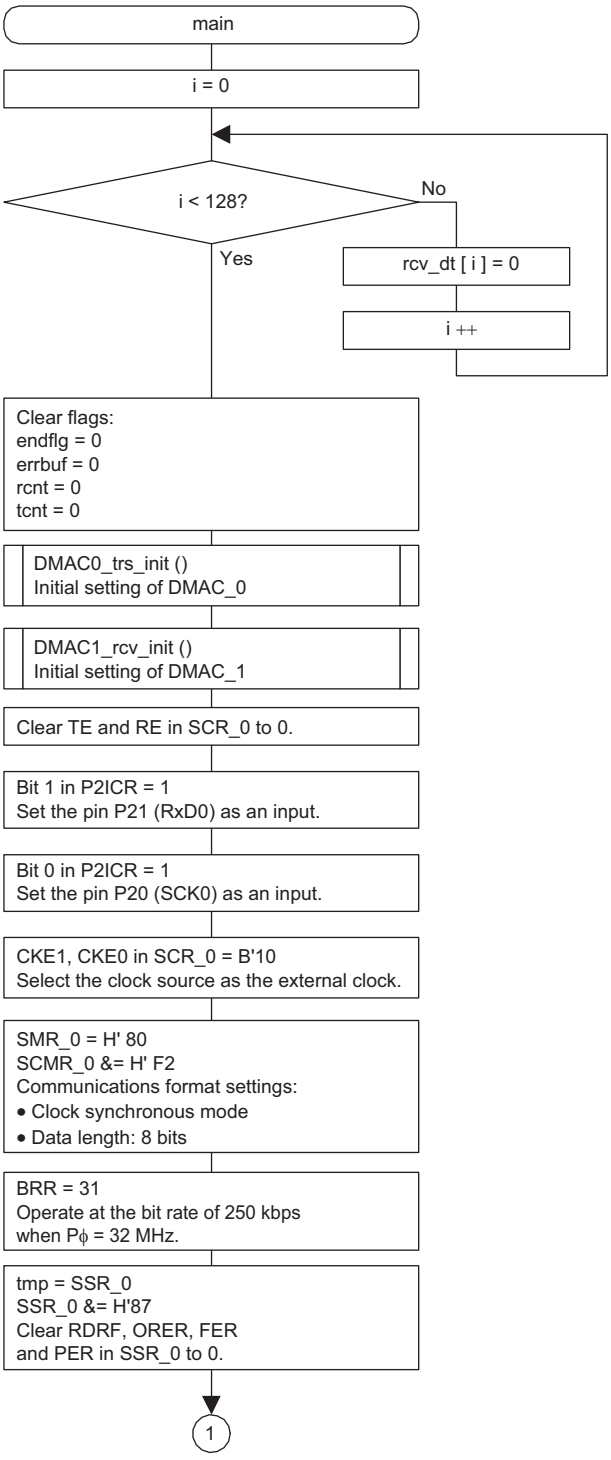
Bit	Bit Name	Setting	R/W	Description
7	TDRE	Undefined	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains data for transmission. [Setting conditions] • Clearing of the TE bit in SCR to 0 • Transferring of data from TDR to TSR [Clearing conditions] • Writing of 0 to TDRE after having read TDRE = 1 (when using an interrupt and having the CPU clear it, be sure to read the flag after having written 0) • Generation of a TXI interrupt request allowing DMAC to write transmit data to TDR
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether RDR holds received data. [Setting condition] • The normal end of serial reception and the transfer of received data from RSR to RDR [Clearing conditions] • Writing of 0 to RDRF after having read RDRF = 1 (when using an interrupt and having the CPU clear it, be sure to read the flag after having written 0) • Generation of an RXI interrupt request allowing DMAC or DTC to read data from RDR. The RDRF flag is not affected and retains its previous value even though the RE bit in SCR is cleared to 0. Note that when the next reception is completed while the RDRF flag is being set to 1, an overrun error occurs and the received data are lost.
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • Occurrence of an overrun error during reception [Clearing condition] • Writing of 0 to ORER after having read ORER = 1 (when using an interrupt and having the CPU clear it, be sure to read the flag after having written 0)
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 of 0 to TDRE after having read TDRE = 1 • Generation of a TXI interrupt request allowing its value DMAC to write data to TDR

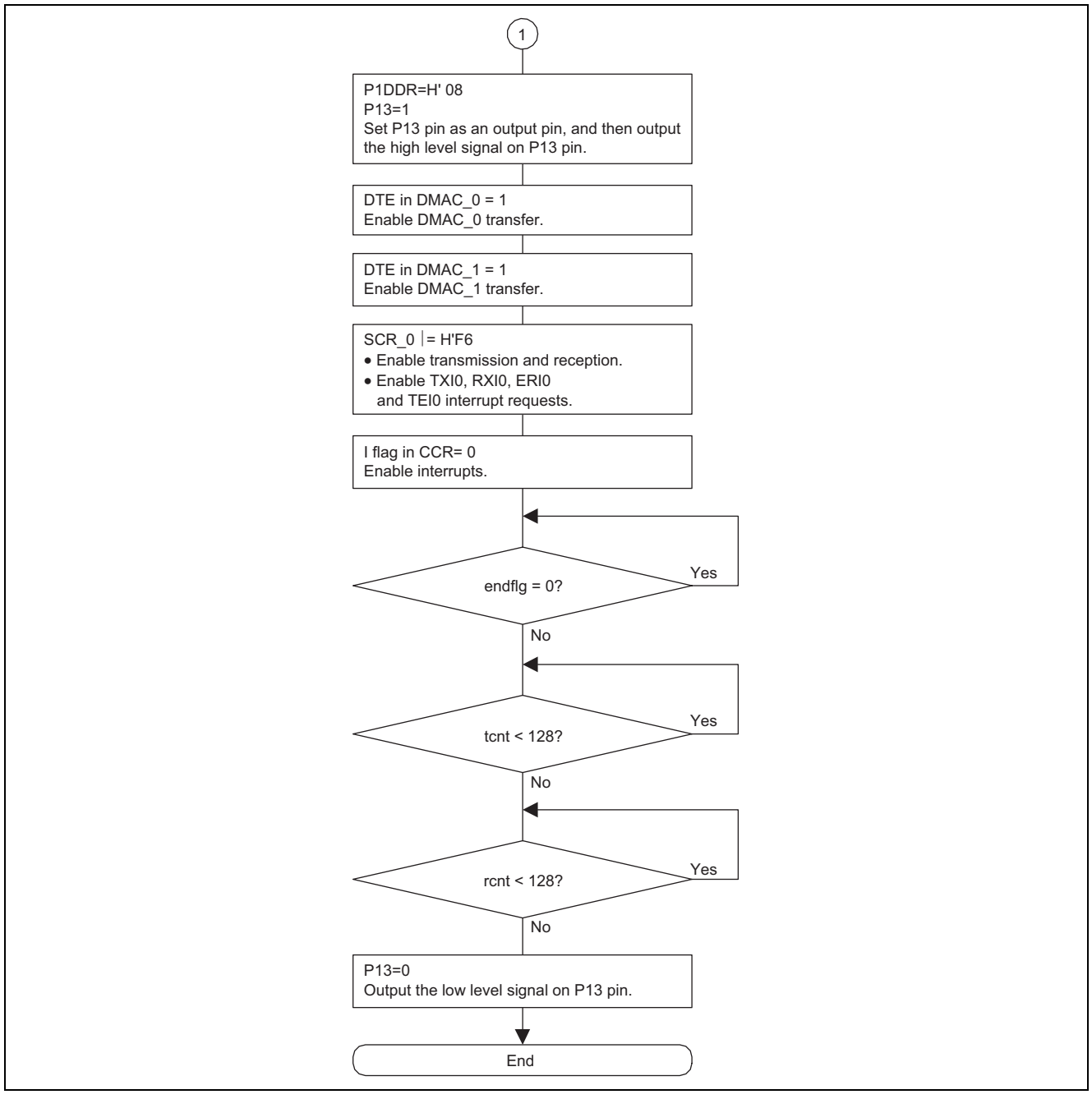
Note: * Only 0 can be written here, to clear the flag.

- 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: Operation is in the normal asynchronous or clock synchronous mode. 1: Operation is in smart card interface mode.

5. Flowchart


main
i = 0
i < 128?
No
Yes
rcv_dt [i] = 0
i ++
Clear flags:
endflg = 0
errbuf = 0
rcnt = 0
tcnt = 0
DMAC0_trs_init ()
Initial setting of DMAC_0
DMAC1_rcv_init ()
Initial setting of DMAC_1
Clear TE and RE in SCR_0 to 0.
Bit 1 in P2ICR = 1
Set the pin P21 (RxD0) as an input.
Bit 0 in P2ICR = 1
Set the pin P20 (SCK0) as an input.
CKE1, CKE0 in SCR_0 = B'10
Select the clock source as the external clock.
SMR_0 = H' 80
SCMR_0 &= H' F2
Communications format settings:
• Clock synchronous mode
• Data length: 8 bits
BRR = 31
Operate at the bit rate of 250 kbps
when Pϕ = 32 MHz.
tmp = SSR_0
SSR_0 &= H'87
Clear RDRF, ORER, FER
and PER in SSR_0 to 0.
1


1
P1DDR=H' 08
P13=1
Set P13 pin as an output pin, and then output the high level signal on P13 pin.
DTE in DMAC_0 = 1
Enable DMAC_0 transfer.
DTE in DMAC_1 = 1
Enable DMAC_1 transfer.
SCR_0 |= H'F6
• Enable transmission and reception.
• Enable TXI0, RXI0, ERI0 and TEI0 interrupt requests.
I flag in CCR= 0
Enable interrupts.
endflg = 0?
Yes
No
tcnt < 128?
Yes
No
rcnt < 128?
Yes
No
P13=0
Output the low level signal on P13 pin.
End

5.7.4 DMAC0_trs_init Function

1. Functional overview
   DMAC_0 initialization. Sets the transfer processing by TXI0 interrupts to TDR_0 from transmitted data holding area.
2. Argument
   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.

- DMA source address register_0 (DSAR_0) Number of bits: 32 Address: H'FFFC00
  Function: DSAR_0 specifies the source address for the transfer.
  Setting: &trs_dt

- DMA destination address register_0 (DDAR_0) Number of bits: 32 Address: H'FFFC04
  Function: DDAR_0 specifies the destination address for the transfer.
  Setting: &TDR_0

- DMA transfer count register_0 (DTCR_0) Number of bits: 32 Address: H'FFFC0C
  Function: DTCR_0 sets the amount of data to be transferred (total amount for transfer).
  Setting: 128

- DMA mode control register_0 (DMDR_0) Number of bits: 32 Address: H'FFFC14

Bit	Bit Name	Setting	R/W	Description
31	DTE	0	R/W	Data Transfer Enable 0: Disables data transfer. 1: Enables data transfer.
26	NRD	0	R/W	Next Request Delay 0: Starts accepting the next transfer request after completion of the current transfer. 1: Starts accepting the next transfer request one cycle after completion of the current round of transfer.
17	ESIF	0	R/(W)*	Transfer Escape Interrupt Flag 0: A transfer escape end interrupt request has not been issued. 1: A transfer escape end interrupt request has been issued.
16	DTIF	0	R/(W)*	Data Transfer Interrupt Flag 0: A transfer end interrupt request by the transfer counter has not been issued. 1: A transfer end interrupt request by the transfer counter has been issued.
15	DTSZ1	0	R/W	Data Access Size 1 and 0
14	DTSZ0		R/W	00: Data access size for transfer is in bytes (8 bits).
13	MDS1	0	R/W	Transfer Mode Select 1 and 0
12	MDS0		R/W	00: Sets the normal transfer mode.

Bit	Bit Name	Setting	R/W	Description
9	ESIE	0	R/W	Transfer Escape interrupt Enable 0: Disables transfer escape interrupt requests. 1: Enables transfer escape interrupt requests.
8	DTIE	1	R/W	Data Transfer End Interrupt Enable 0: Disables transfer end interrupt requests. 1: Enables transfer end interrupt requests.
7	DTF1	1	R/W	Data Transfer Factor 1 and 0
6	DTF0	0	R/W	10: DMAC activation source is an on-chip module interrupt.
5	DTA	1	R/W	Data Transfer Acknowledge When DTF 1 and DTF 0 = H'10, which selects execution of DMA transfer in response to an internal module interrupt, this bit enables or disables clearing of the source flag selected by DMRSR. 0: Source flag for the internal module interrupt is not cleared. 1: Source flag for the internal module interrupt is cleared.

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

- DMA address control register_0 (DACR_0) Number of bits: 32 Address: H'FFFC18

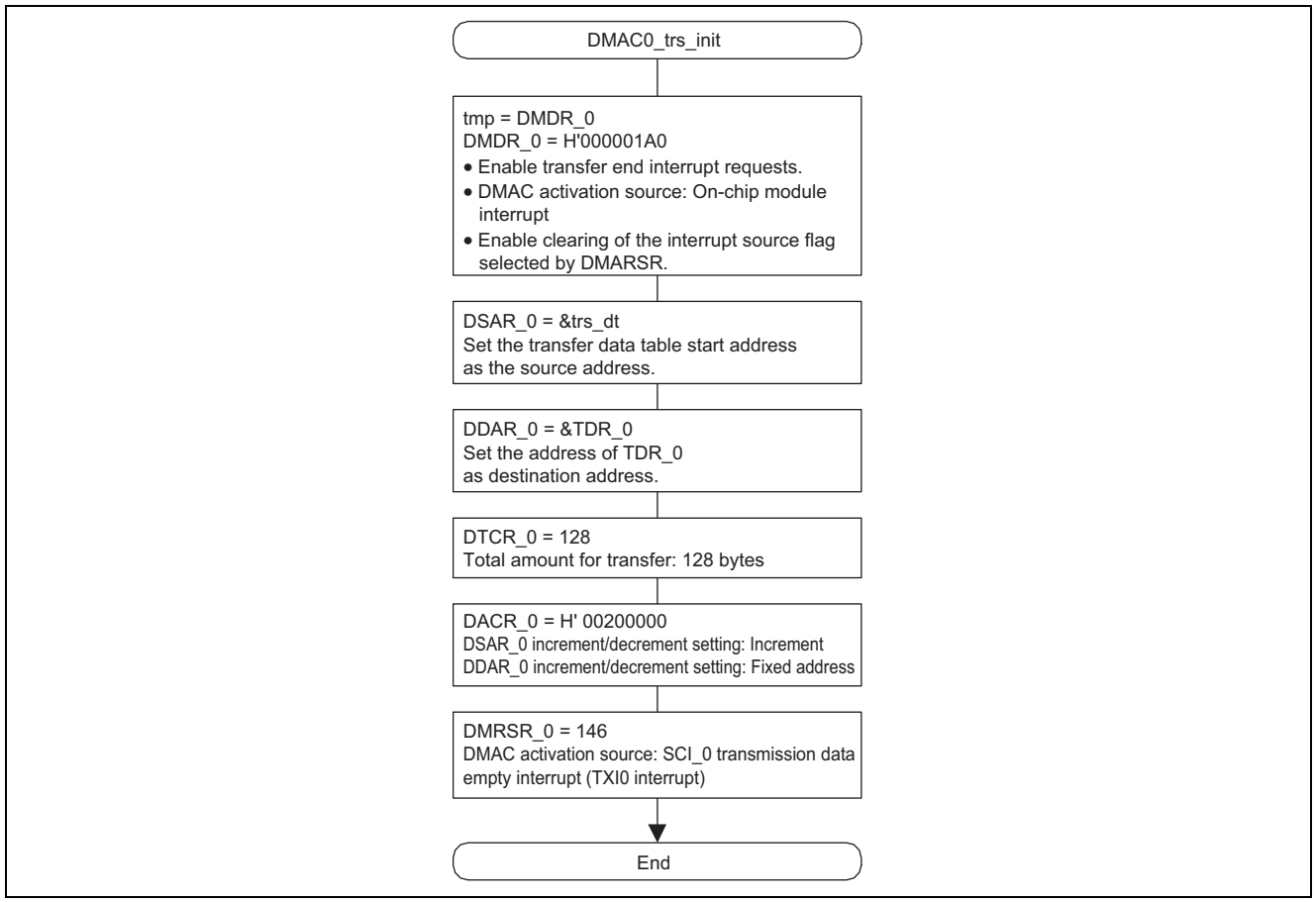
Bit	Bit Name	Setting	R/W	Description
31	AMS	0	R/W	Address Mode Select 0: Dual address mode 1: Single address mode
21	SAT1	1	R/W	Source Address Update Mode 1 and 0
20	SAT0	0	R/W	10: Increment the source address.
17	DAT1	0	R/W	Destination Address Update Mode 1 and 0
16	DAT0	0	R/W	00: Destination address is fixed.

- DMA module request select register_0 (DMRSR_0) Number of bits: 8 Address: H'FFFD20

  Function: DMRSR_0 specifies the source of on-chip module interrupts. The setting 146 corresponds to DMAC activation by SCI_0 transmission data empty interrupts (TXI0 interrupts).

  Setting: 146

5. Flowchart


DMAC0_trs_init
tmp = DMDR_0
DMDR_0 = H'000001A0
• Enable transfer end interrupt requests.
• DMAC activation source: On-chip module interrupt
• Enable clearing of the interrupt source flag selected by DMARSR.
DSAR_0 = &trs_dt
Set the transfer data table start address as the source address.
DDAR_0 = &TDR_0
Set the address of TDR_0 as destination address.
DTCR_0 = 128
Total amount for transfer: 128 bytes
DACR_0 = H' 00200000
DSAR_0 increment/decrement setting: Increment
DDAR_0 increment/decrement setting: Fixed address
DMRSR_0 = 146
DMAC activation source: SCI_0 transmission data empty interrupt (TXI0 interrupt)
End

5.7.5 DMAC1_rcv_init Function

1. Functional overview
   DMAC_1 initialization. Sets the transfer processing by RXI0 interrupts from RDR_0 to received data holding area.
2. Argument
   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.

- DMA source address register_1 (DSAR_1) Number of bits: 32 Address: H'FFFC20
  Function: DSAR_1 specifies the source address for the transfer.
  Setting: &RDR_0

- DMA destination address register_1 (DDAR_1) Number of bits: 32 Address: H'FFFC24
  Function: DDAR_1 specifies the destination address for the transfer.
  Setting: &rcv_dt

- DMA transfer count register_1 (DTCR_1) Number of bits: 32 Address: H'FFFC2C
  Function: DTCR_1 sets the amount of data to be transferred (total amount for transfer).
  Setting: 128

- DMA mode control register_1 (DMDR_1) Number of bits: 32 Address: H'FFFC34

Bit	Bit Name	Setting	R/W	Description
31	DTE	0	R/W	Data Transfer Enable 0: Disables data transfer. 1: Enables data transfer.
26	NRD	0	R/W	Next Request Delay 0: Starts accepting the next transfer request after completion of the current transfer. 1: Starts accepting the next transfer request one cycle after completion of the current round of transfer.
17	ESIF	0	R/(W)*	Transfer Escape Interrupt Flag 0: A transfer escape end interrupt request has not been issued. 1: A transfer escape end interrupt request has been issued.
16	DTIF	0	R/(W)*	Data Transfer Interrupt Flag 0: A transfer end interrupt request by the transfer counter has not been issued. 1: A transfer end interrupt request by the transfer counter has been issued.
15	DTSZ1	0	R/W	Data Access Size 1 and 0
14	DTSZ0	0	R/W	00: Data access size for transfer is in bytes (8 bits).
13	MDS1	0	R/W	Transfer Mode Select 1 and 0
12	MDS0	0	R/W	00: Sets the normal transfer mode.
9	ESIE	0	R/W	Transfer Escape Interrupt Enable 0: Disables transfer escape interrupt requests. 1: Enables transfer escape interrupt requests.
8	DTIE	1	R/W	Data Transfer End Interrupt Enable 0: Disables transfer end interrupt requests. 1: Enables transfer end interrupt requests.
7	DTF1	1	R/W	Data Transfer Factor 1 and 0
6	DTF0	0	R/W	10: DMAC activation source is an on-chip module interrupt.
5	DTA	1	R/W	Data Transfer Acknowledge DMA transfer in response to an internal module interrupt, this bit enables or disables clearing of the source flag selected by DMRSR. 0: Source flag for the internal module interrupt is not cleared. 1: Source flag for the internal module interrupt is cleared.

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

- DMA address control register_1 (DACR_1) Number of bits: 32 Address: H'FFFC38

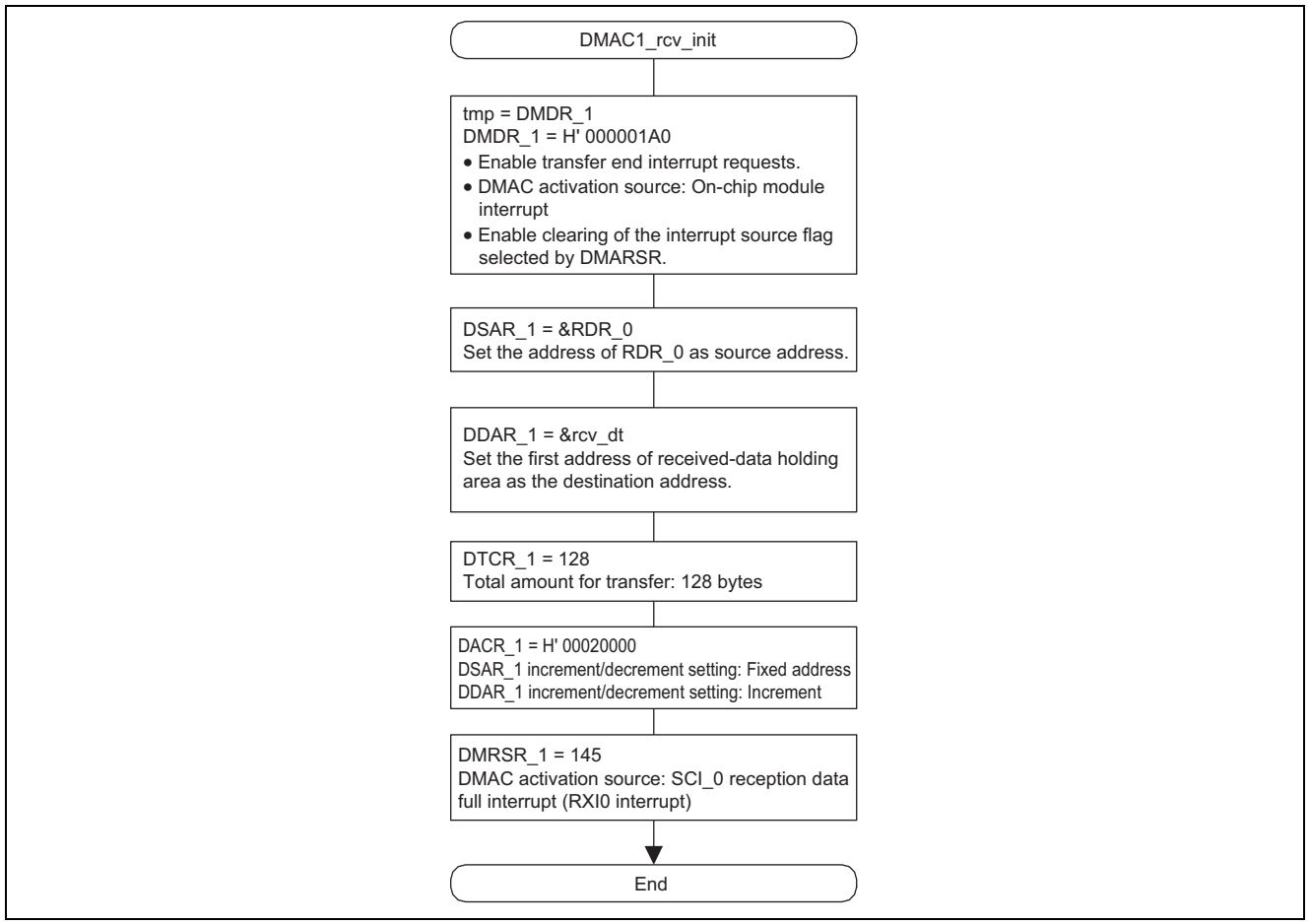
Bit	Bit Name	Setting	R/W	Description
31	AMS	0	R/W	Address Mode Select 0: Dual address mode 1: Single address mode
21	SAT1	0	R/W	Source Address Update Mode 1 and 0
20	SAT0	0	R/W	00: Source address is fixed.
17	DAT1	1	R/W	Destination Address Update Mode 1 and 0
16	DAT0	0	R/W	10: Increments the source address.

- DMA module request select register_1 (DMRSR_1) Number of bits: 8 Address: H'FFFD21

  Function: DMRSR_1 specifies the source of on-chip module interrupts. The setting 145 corresponds to DMAC activation by SCI_0 received data full interrupts (RXI0 interrupts).

  Setting: 145

5. Flowchart



5.7.6 dmtend0_int Function

1. Functional overview
   Handler for the DMAC_0 transfer end interrupt. Sets TEI0 interrupt requests enabled, TXI0 interrupt requests disabled, and DMAC_0 transfer end interrupt requests disabled.
2. Argument
   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.

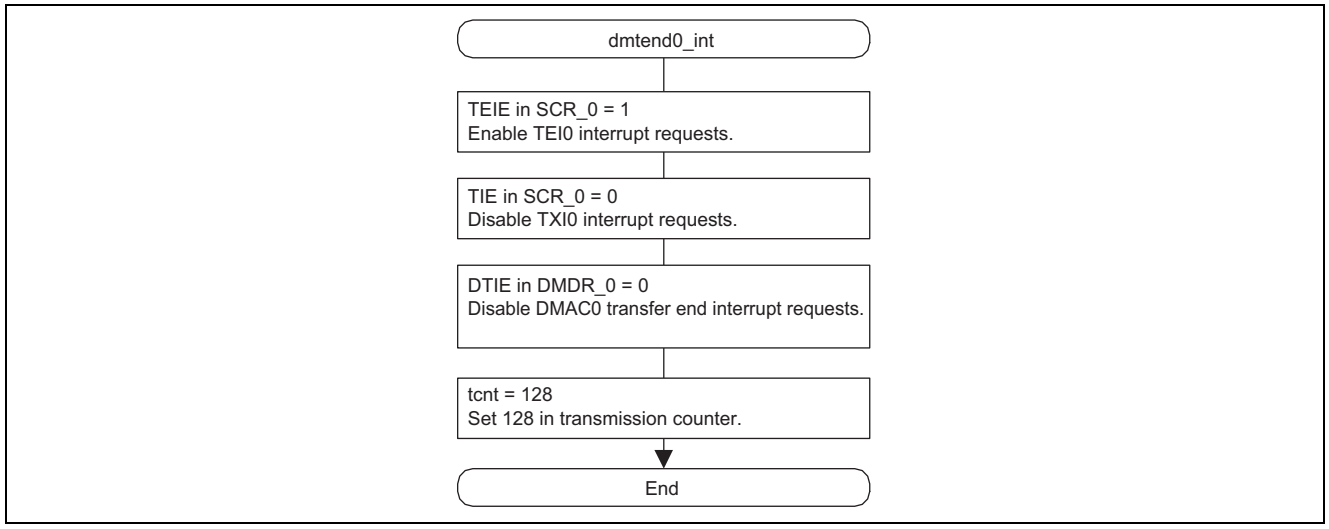
- Serial control register_0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable 0: Disables TXI interrupt requests. 1: Enables TXI interrupt requests.
2	TEIE	1	R/W	Transmit End Interrupt Enable 0: Disables TEI interrupt requests. 1: Enables TEI interrupt requests.

- DMA mode control register_0 (DMDR_0) Number of bits: 32 Address: H'FFFC14

Bit	Bit Name	Setting	R/W	Description
8	DTIE	0	R/W	Data Transfer End Interrupt Enable 0: Disables transfer end interrupt requests. 1: Enables transfer end interrupt requests.

5. Flowchart



5.7.7 dmtend1_int Function

1. Functional overview
   Handler for the DMAC_1 transfer end interrupt. Sets SCI_0 reception disabled, RXI0 and ERI0 interrupt requests, and DMAC_1 transfer end interrupt requests disabled.
2. Argument
   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.

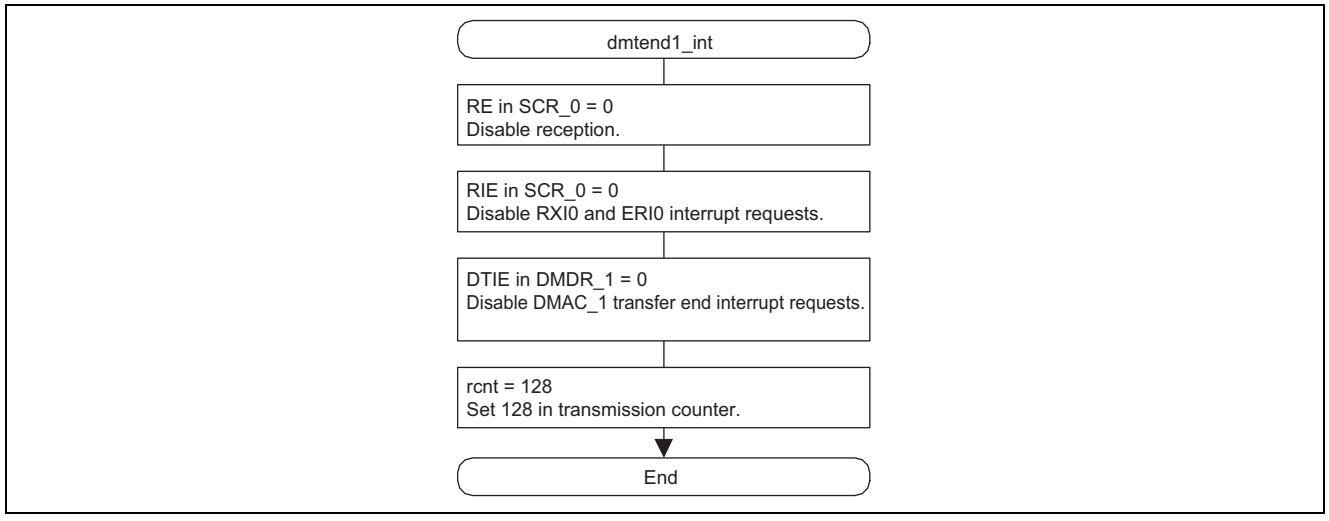
- Serial control register_0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable 0: Disables RXI and ERI interrupt requests. 1: Enables RXI and ERI interrupt requests.
4	RE	0	R/W	Receive Enable 0: Disables reception. 1: Enables reception.

- DMA mode control register _1 (DMDR_1) Number of bits: 32 Address: H'FFFC34

Bit	Bit Name	Setting	R/W	Description
8	DTIE	0	R/W	Data Transfer End Interrupt Enable 0: Disables transfer end interrupt requests. 1: Enables transfer end interrupt requests.

5. Flowchart



5.7.8 eri0_int Function

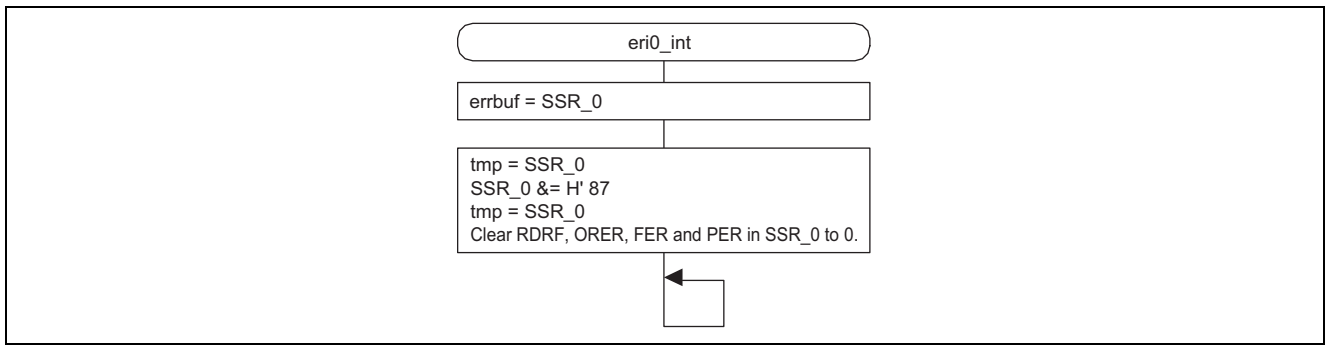
1. Functional overview
   Handler for SCI_0 reception error interrupts (ERI0 interrupt). Writes error data to RAM and initializes SSR_0.
2. Argument
   None
3. Return value
   None
4. Description of internal registers used
   The internal registers used in this sample task are described below. The settings shown in these tables are the values used in this sample task and differ from the initial values.

- 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 RDR holds received data. [Setting condition] • The normal end of serial reception and the transfer of received data from RSR to RDR [Clearing conditions] • Writing of 0 to RDRF after having read RDRF = 1 (when using an interrupt and clearing by CPU, be sure to read a flag after having written 0.) • Generation of an RXI interrupt request allowing DMAC or DTC to read data from RDR. The RDRF flag is not affected and retains its previous value even though the RE bit in SCR is cleared to 0. Note that when the next reception is completed while the RDRF flag is being set to 1, an overrun error occurs and the received data are lost.
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • Occurrence of an overrun error during reception [Clearing condition] • Writing of 0 to ORER after having read ORER = 1 (when using an interrupt and clearing by CPU, be sure to read a flag after having written 0.)

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

5. Flowchart



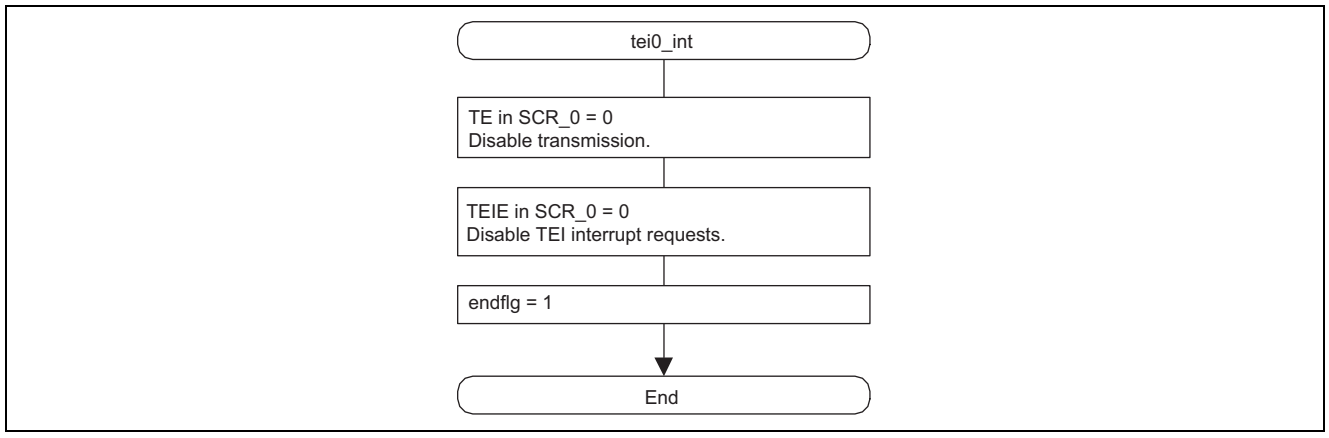
5.7.9 tei0_int Function

1. Functional overview
   Handler for SCI_0 transmission end interrupts (TEI0 interrupt). Sets to SCI_0 transmission and TEI0 interrupt requests disabled.
2. Argument
   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.

- Serial control register_0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
5	TE	0	R/W	Transmit Enable 0: Disables transmission. 1: Enables transmission.
2	TEIE	0	R/W	Transmit End Interrupt Enable 0: Disables TEI interrupt requests. 1: Enables TEI interrupt requests.

5. Flowchart



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.