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.

RENESAS

H8SX Family

Using the DMAC to Drive Continuous SCI 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 (reception) of data enables continuous reception with no CPU intervention.

Target Device

H8SX/1653

Contents

1. Specification

Clock synchronous transfer is used to receive 128 bytes of data. Using the DMAC to handle the transfer (reception) of data enables continuous 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 transmitting side, the state of P13 pin is polled. When the P13 pin is a high level, the transmitting side judges the transfer is enabled at the receiving side, after which operations for the clock synchronous transmission of 128 bytes of data proceed.
- After a power-on reset of the receiving 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 reception of 128 bytes of data proceed in synchronization with the transmitting-side clock input on pin SCK.

In this sample task, the DMAC module is interrupt-activated to handle consecutive reception of the 128 bytes of data.

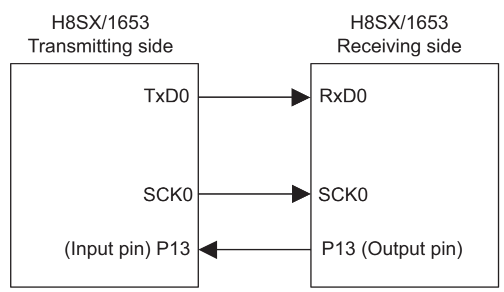


Figure 1 Clock Synchronous Serial Reception

Table 1 Format for Clock Synchronous Serial Reception

Format	Setting
Pϕ	32 MHz
Serial communications mode	Clock synchronous
Clock source	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.

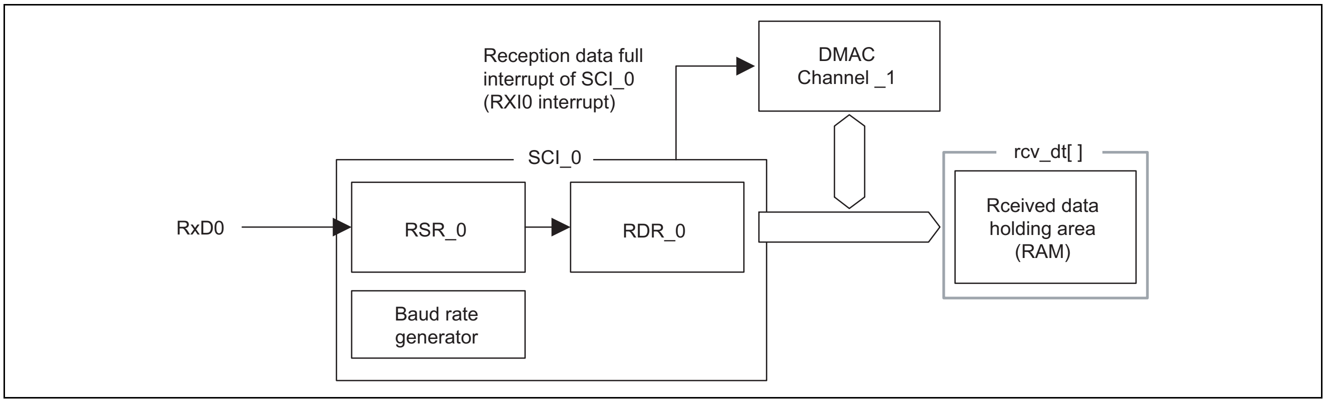


Figure 2 Functions of the H8SX/1653

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

1. SCI_0
   Receives data with timing provided by the clock synchronous communications.
   - 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 channel 1
   - 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 reception. Figure 3 is a block diagram of SCI_0.

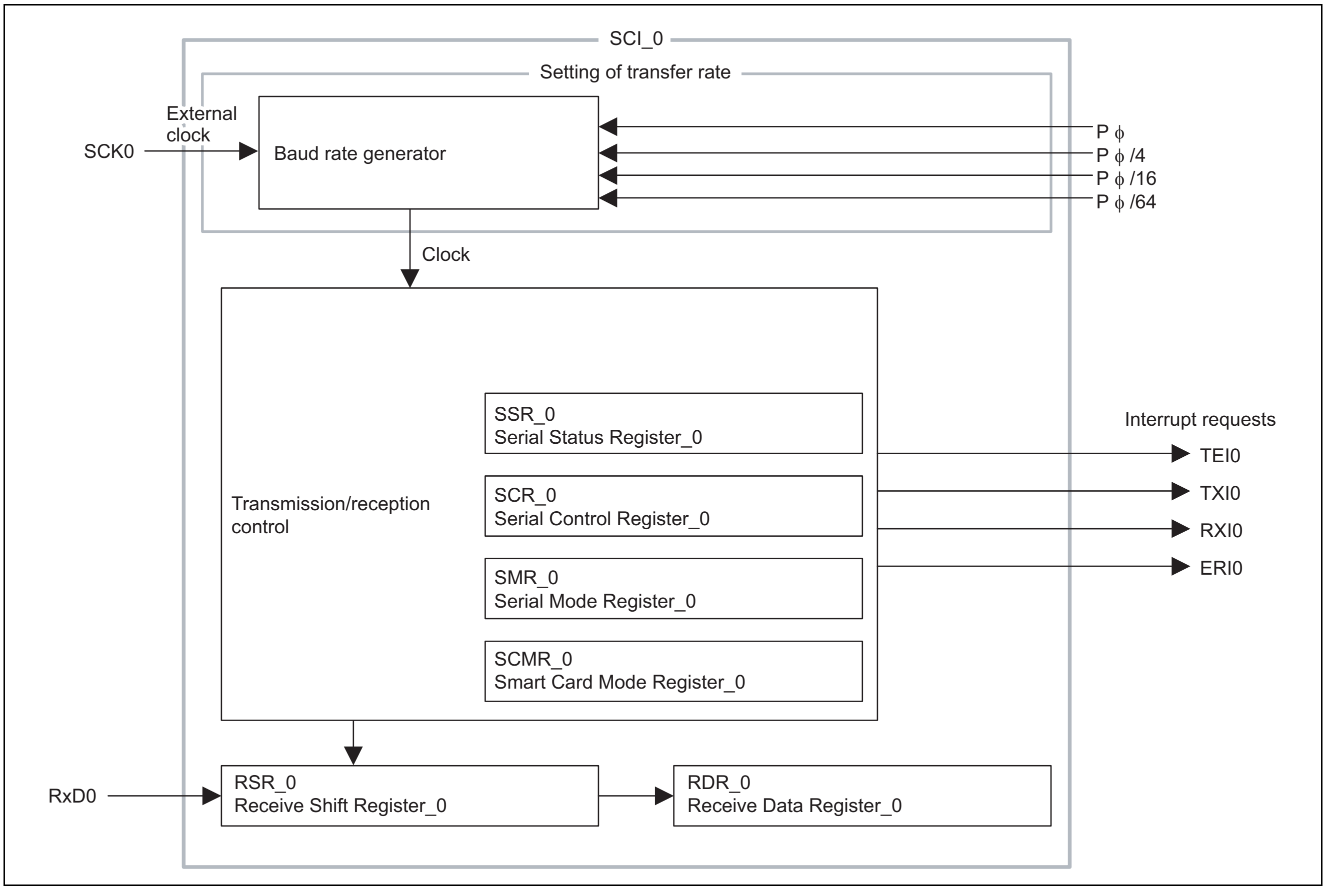


Figure 3 Block Diagram of SCI_0

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.

- 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 Channel 1 of the DMAC

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

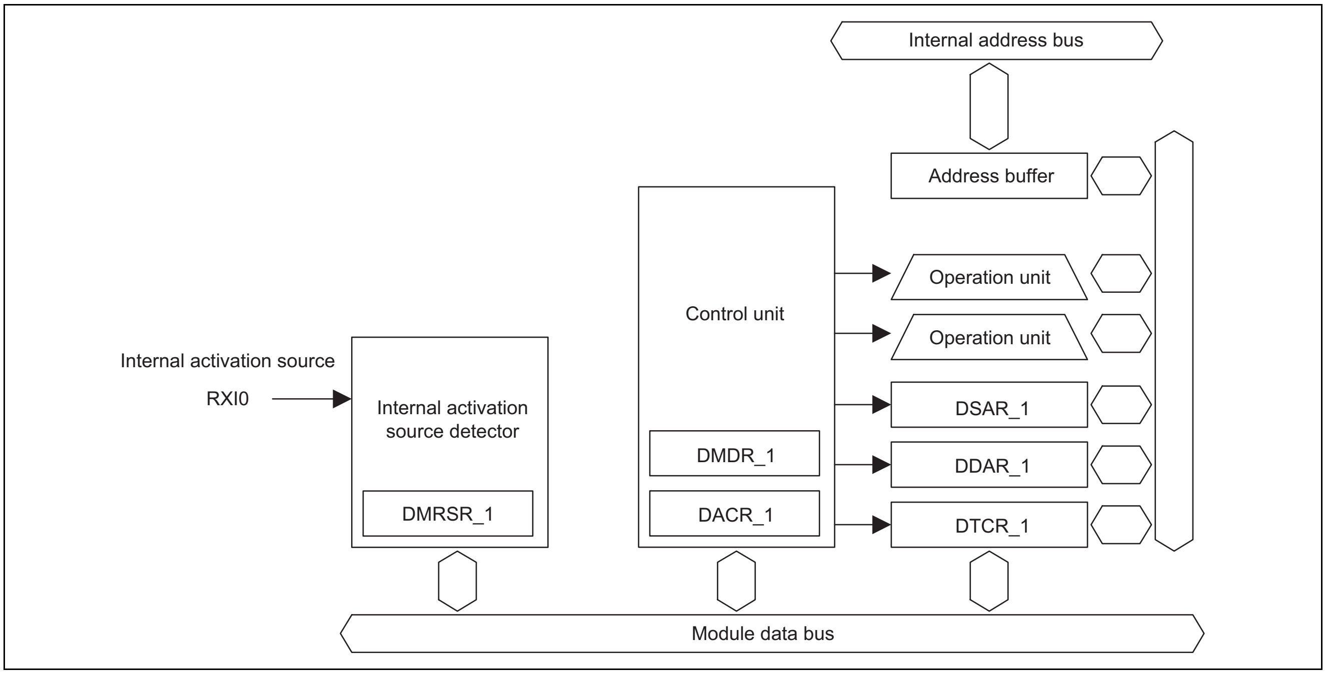


Figure 4 Block Diagram of the DMAC

The description with reference to figure 4 is stated below.

- DMA source address register _1 (DSAR_1)
  DSAR_1 is a 32-bit readable/writable register and specifies the source address for the transfer. The 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 _1 (DDAR_1)
  DDAR_1 is a 32-bit readable/writable register and specifies the destination address for the transfer. The 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 _1 (DTCR_1)
  DTCR_1 is a 32-bit readable/writable register and specifies 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, the register is set for 128 bytes of data, and the byte is selected as the unit of data access. One is subtracted from the value on DMAC operation, to indicate the amount still to be transferred.

- DMA mode control register _1 (DMDR_1)
  DMDR_1 controls DMAC operation.

- DMA address control register_1 (DACR_1)
  DACR_1 sets the operating mode and transfer method.

- DMA module request select register_1 (DMRSR_1)
  DMRSR_1 sets 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 transmitting and receiving sides.

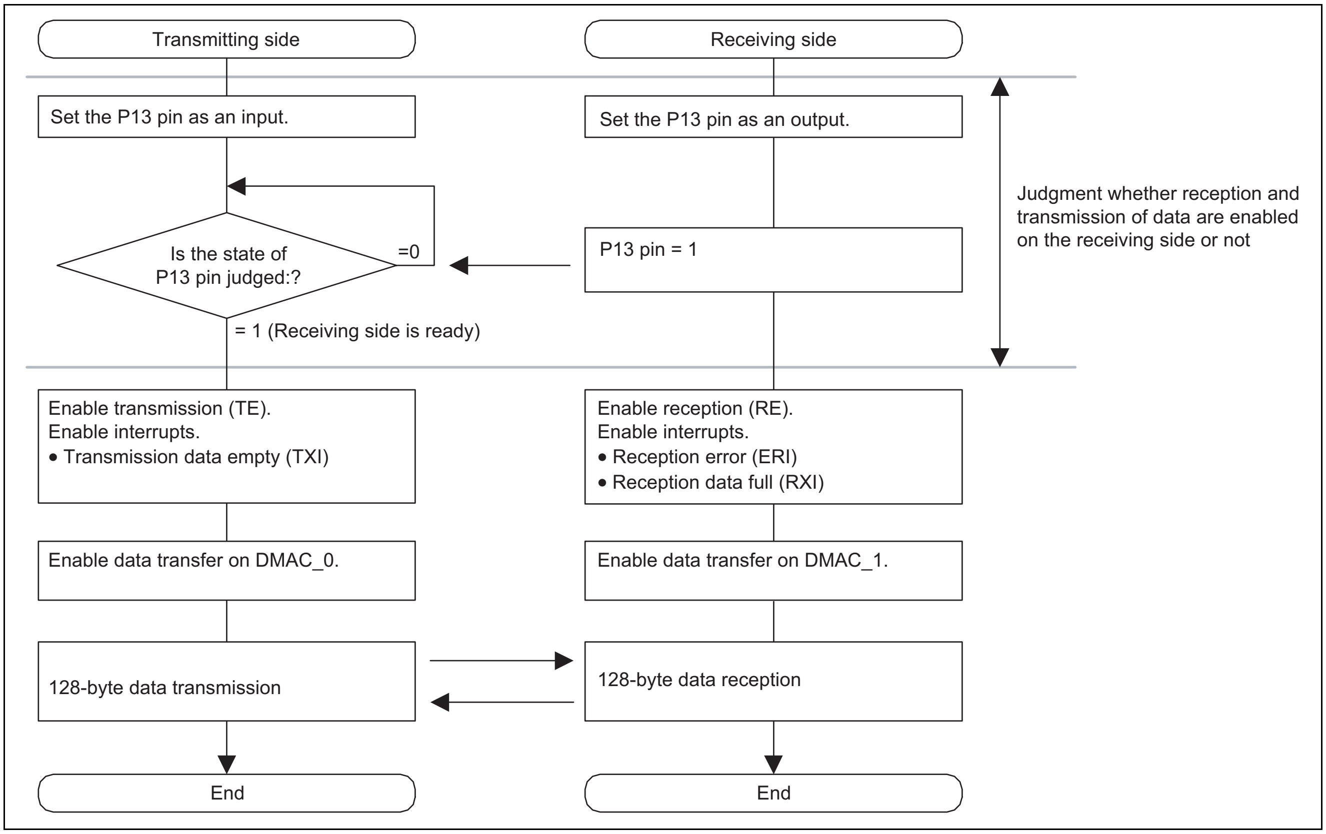


Figure 5 Outline of Operation

4.2 Reception

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

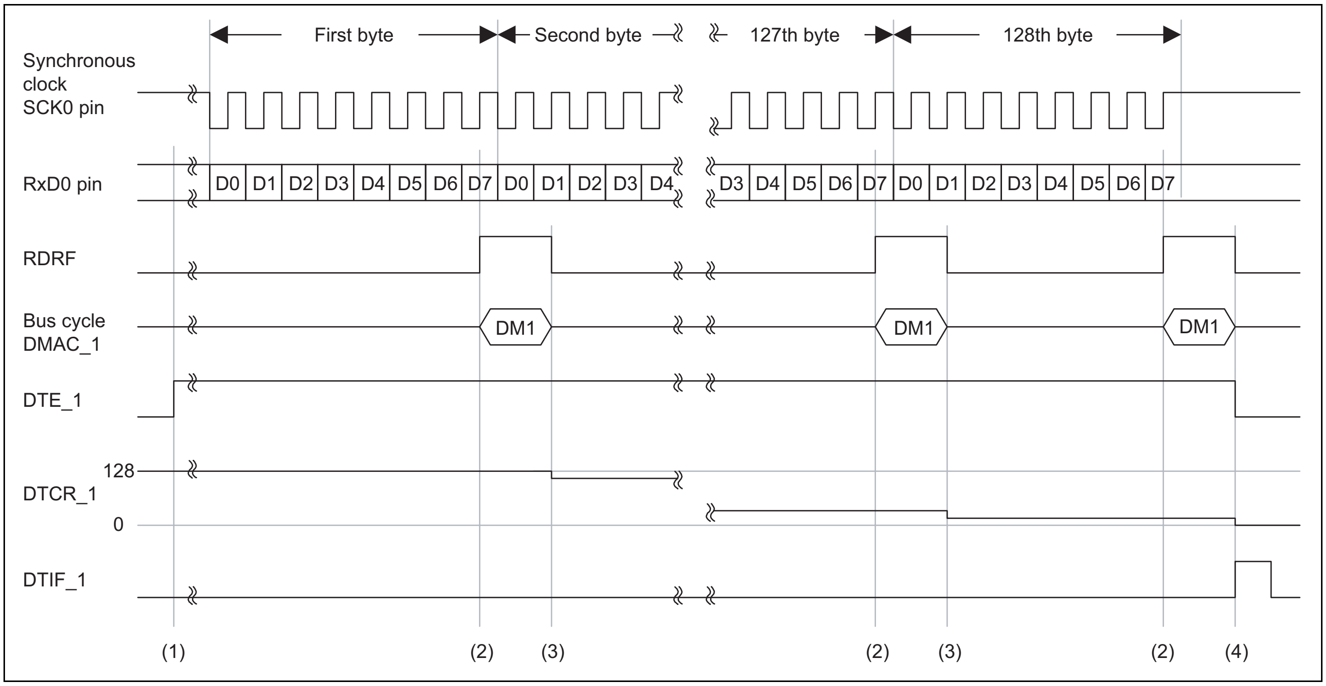


Figure 6 Timing of Reception

Table 3 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. Set clock source 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 4 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 5 Section Setting

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

Table 6 Vector Table for Interrupt Exception Handling

Exception Handling Source		Vector No.	Vector Address	Function to Interrupt Destination
Reset		0	H'000000	init
DMAC_1	DMTEND1	129	H'000204	dmtend1_int
SCI_0	ERI0	144	H'000240	eri0_int

5.2 List of Functions

Table 7 lists the functions used in this sample task. Figure 7 shows the structure of hierarchy.

Table 7 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	Main routine Selects clock synchronous SCI. Calls the DMAC1_rcv_init function. Outputs a high-level signal on pin P13. Makes settings for transmission and reception of 128 bytes of data.
DMAC1_rcv_init	DMAC_1 initialization Selects RXI0-interrupt-triggered processing of transfer from the area where received data are to be stored.
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.

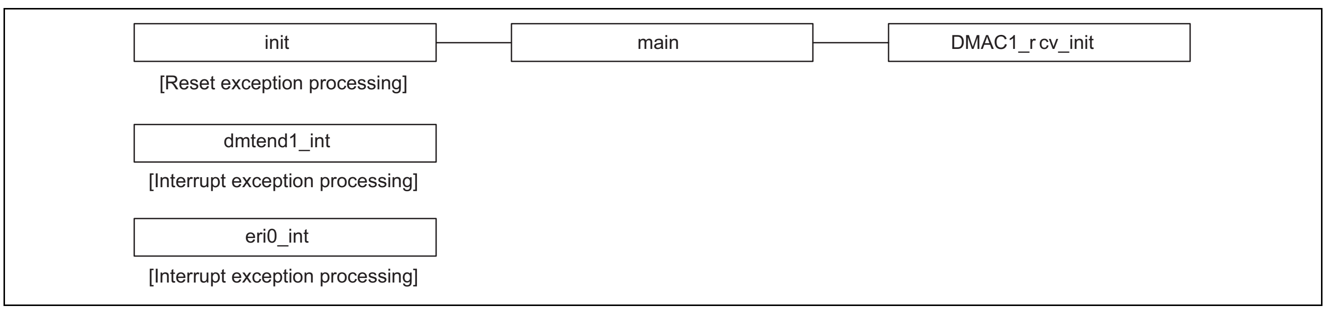


Figure 7 Hierarchy of Calls in the User Program

5.3 RAM Usage

Table 8 RAM Usage

Type	Variable Name	Description	Used in
unsigned char	errbuf	Reception error buffer The contents of SSR_0 are stored when an overrun error occurs.	main, eri0_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 Symbolic Constants

Table 9 Symbolic Constants

Constant Name	Setting	Description
NUM	128	Sets the number of data for reception.

5.5 Description of Functions

5.5.1 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 10 9 8	MDS3 MDS2 MDS1 MDS0	Undefined* Undefined* Undefined* Undefined*	R R R R	Mode Select 3 to 0 These bits indicate the operating mode selected by the mode pins (MD2 to MD0) (see table 10). 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.

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

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

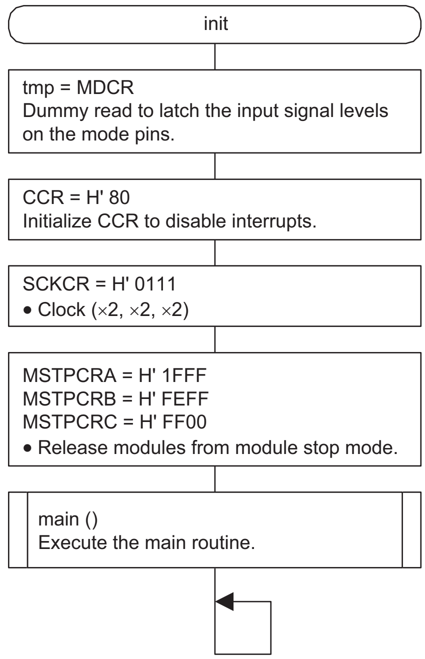
- Module stop control register B (MSTPCRB)   Number of bits: 16   Address: H'FFFDCA

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


init
tmp = MDCR
Dummy read to latch the input signal levels on the mode pins.
CCR = H' 80
Initialize CCR to disable interrupts.
SCKCR = H' 0111
• Clock (×2, ×2, ×2)
MSTPCRA = H' 1FFF
MSTPCRB = H' FEFF
MSTPCRC = H' FF00
• Release modules from module stop mode.
main ()
Execute the main routine.

5.5.2 main Function

1. Functional overview
   Main routine: Sets the clock synchronous SCI, calls the DMAC1_rcv_init function, 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: 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 (RxD0) pin input buffer is disabled. Input signal is fixed to the high level. 1: P21 (RxD0) pin input buffer is enabled. The pin state reflects the peripheral modules.
0	P20ICR	1	R/W	0: P20 (SCK0) pin input buffer is disabled. Input signal is fixed to the high level. 1: P21 (SCK0) pin input buffer is enabled. The pin state reflects the peripheral modules.

- 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 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 0	CKS1 CKS0	0 0	R/W R/W	Clock Select 1, 0 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 0	CKE1 CKE0	1 0	R/W	Clock Enable 1 and 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

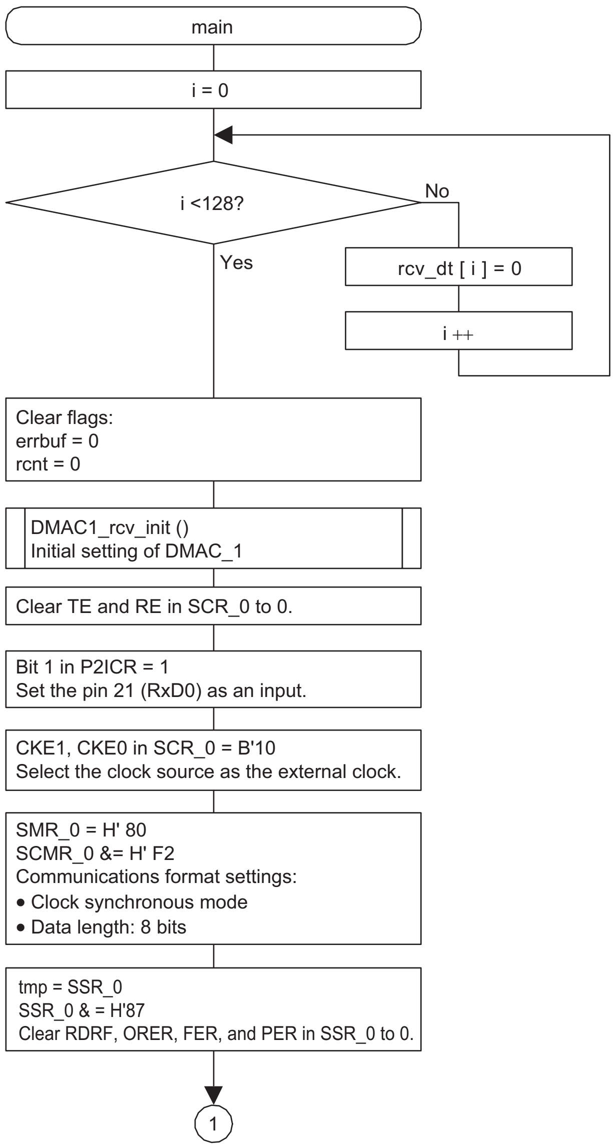
Bit	Bit Name	Setting	R/W	Description
7	TDRE	Undefined	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains data for transmission. [Setting conditions] • Clearing of the TE bit in SCR to 0 • Transfer of data from TDR to TSR [Clearing conditions] • Writing of 0 to TDRE after having read TDRE = 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

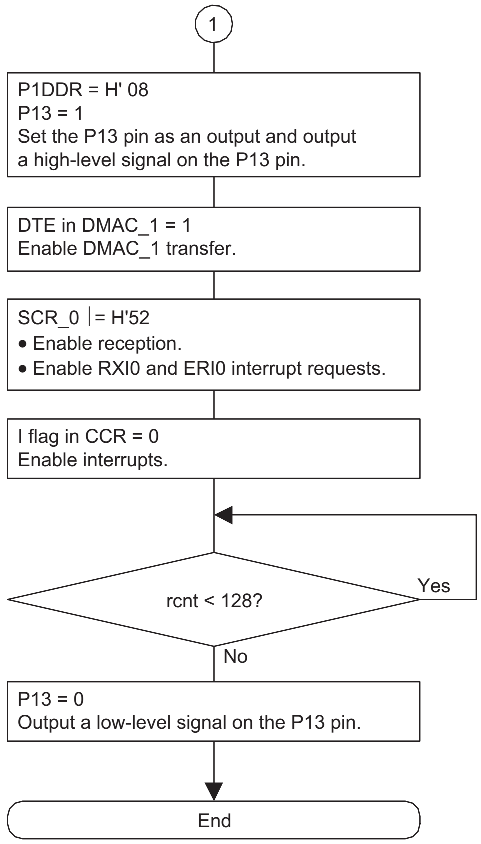
Note: * Only 0 can be written here, to clear the flag.

- 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




1
P1DDR = H' 08
P13 = 1
Set the P13 pin as an output and output
a high-level signal on the P13 pin.
DTE in DMAC_1 = 1
Enable DMAC_1 transfer.
SCR_0 | = H'52
• Enable reception.
• Enable RXI0 and ERI0 interrupt requests.
I flag in CCR = 0
Enable interrupts.
rcnt < 128?
Yes
No
P13 = 0
Output a low-level signal on the P13 pin.
End

5.5.3 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 of 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 14	DTSZ1 DTSZ0	0 0	R/W R/W	Data Access Size 1 and 0 00: Data access size for transfer is in bytes (8 bits).
13 12	MDS1 MDS0	0 0	R/W R/W	Transfer Mode Select 1 and 0 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 6	DTF1 DTF0	1 0	R/W R/W	Data Transfer Factor 1 and 0 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

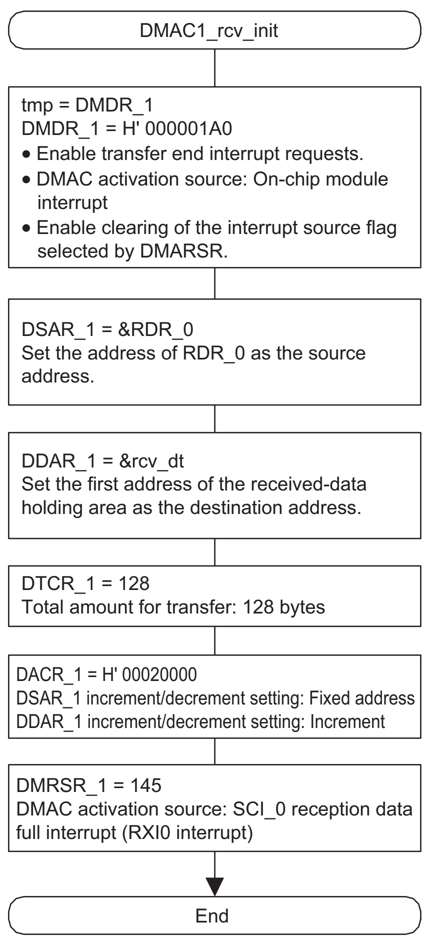
Bit	Bit Name	Setting	R/W	Description
31	AMS	0	R/W	Address Mode Select 0: Dual address mode 1: Single address mode
21	SAT1	0	R/W	Source Address Update Mode 1 and 0
20	SAT0	0	R/W	00: Source address is fixed.
17	DAT1	1	R/W	Destination Address Update Mode 1 and 0
16	DAT0	0	R/W	10: Increments the source address.

- 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



5.5.4 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.

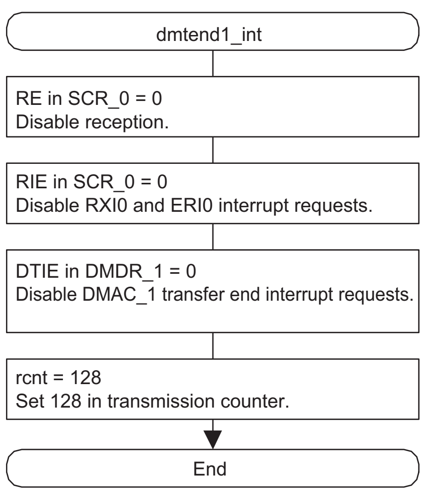
- Serial control register_0 (SCR_0) Number of bits: 8 Address: H'FFFF82

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



5.5.5 eri0_int Function

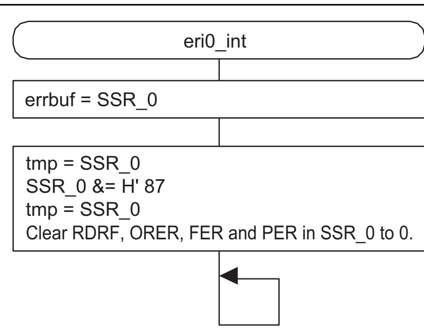
1. Functional overview
   Handler for SCI_0 reception error interrupts (ERI0 interrupt). Writes error data to RAM and initializes SSR_0.
2. Argument
   None
3. Return value
   None
4. Description of internal registers used
   The internal registers used in this sample task are described below. The settings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



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

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