RX210, RX21A, and RX220 Groups

Clock Synchronous SCIc Communication Using DMACA

R01AN1200EJ0101
Rev. 1.01
July 1, 2014

Abstract

This application note describes how to perform clock synchronous communication using the serial communications interface (SCI) with the DMA controller (DMAC) in the RX210, RX21A, and RX220 Groups.

Products

RX210, RX21A, and RX220 Groups

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1 Specifications

This document describes performing clock synchronous serial communication using the SCI.

Transmit data is prestored in the RAM's transmit data storage area and transferred using the DMAC. Receive data is stored in the RAM's receive data storage area using the DMAC.

Serial communication start when a falling edge is detected on the IRQ1 interrupt request pin.

- Bit rate: 38400 bps
- Communication formats: 8-bit length, LSB first
- Clock input/output: Clock output (master)
- Transmit and receive operations: Simultaneous transmit and receive operations

Table 1.1 lists the Peripheral Functions and Their Applications, and Figure 1.1 shows the Block Diagram.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
SCIc channel 1 (SCI1)	Clock synchronous serial communication
DMACA channel 0 (DMAC0)	Transfer SCI1 receive data to the RAM
DMACA channel 1 (DMAC1)	Transfer RAM transmit data to SCI1
IRQ1	Start trigger for serial communication

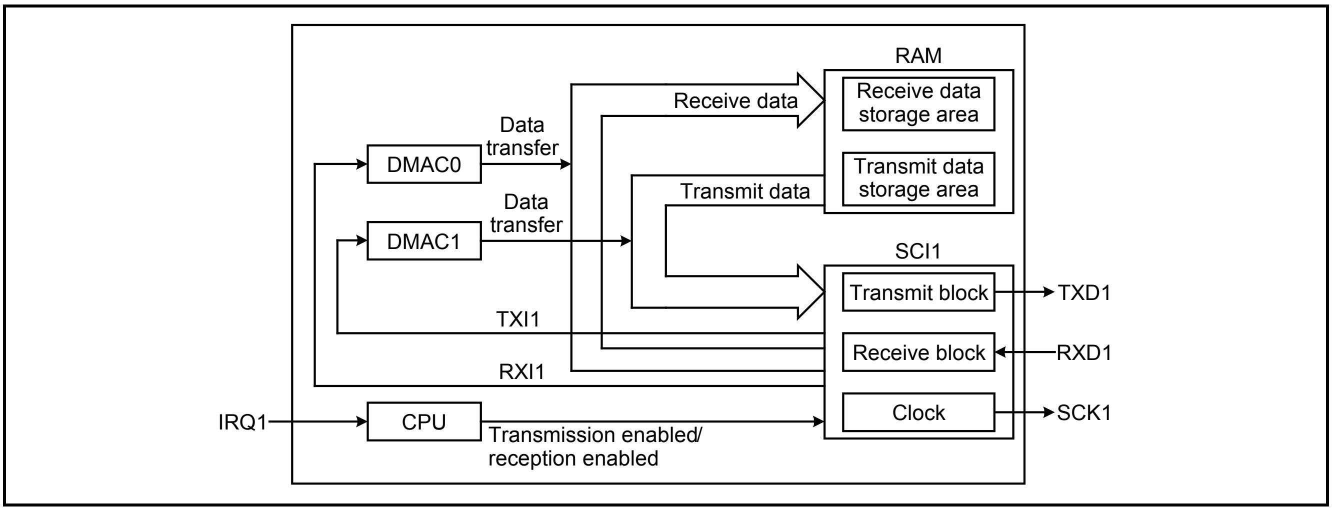


Figure 1.1 Block Diagram

2 Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	R5F52108ADFP (RX210 Group)
Operating frequencies	- Main clock: 20 MHz - PLL: 100 MHz (main clock divided by 2 and multiplied by 10) - System clock (ICLK): 50 MHz (PLL divided by 2) - Peripheral module clock B (PCLKB): 25 MHz (PLL divided by 4)
Operating voltage	5.0 V
Integrated development environment	Renesas Electronics Corporation High-performance Embedded Workshop Version 4.09.01
C compiler	Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.1.02 Release 01
	-cpu=rx200 -output=obj="$(CONFIGDIR)\$(FILELEAF).obj" -debug -nologo The integrated development environment default settings are used.
iodefine.h version	Version 1.2A
Endian	Little endian
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Sample code version	Version 1.00
Board used	Renesas Starter Kit for RX210 (product part number: R0K505210C000BE)

3 Reference Application Notes

For additional information associated with this document, refer to the following application notes.

- RX210 Group Initial Setting Rev. 2.00 (R01AN1002EJ)
- RX21A Group Initial Setting Rev. 1.10 (R01AN1486EJ)
- RX220 Group Initial Setting Rev. 1.10 (R01AN1494EJ)

The initial setting functions in the reference application notes are used in the sample code in this application note. The revision numbers of the reference application notes are current as of when this application note was made. However the latest version is always recommended. Visit the Renesas Electronics Corporation website to check and download the latest version.

4 Hardware

4.1 Hardware Configuration

Figure 4.1 shows a Connection Example.

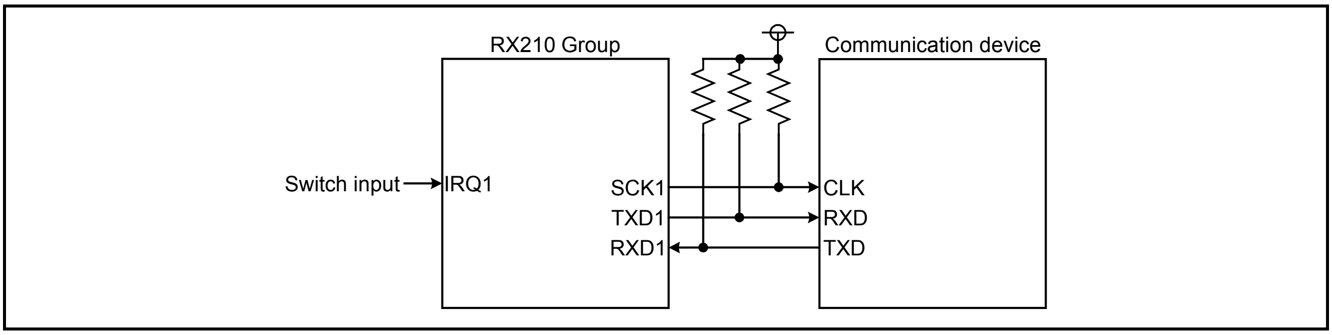


Figure 4.1 Connection Example

4.2 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

This document assumes the product used is a 100-pin package. For products with less than 100 pins, select pins appropriate for the package used.

Table 4.1 Pins Used and Their Functions

Pin Name	I/O	Function
P31/IRQ1	Input	Switch input to start transmission and reception
P17/SCK1	Output	Clock output from SCI1
P15/RXD1	Input	Receive data input from SCI1
P16/TXD1	Output	Transmit data output from SCI1

5 Software

In the sample code, DMAC is used for automatically processing SCI1 data transmission and reception. SCI1 data transmission and reception are started by pressing a switch.

When data transmission is enabled, a TXI1 interrupt request is generated, and this becomes the DMAC1 transfer request. DMAC1 is used to transfer transmit data from the transmit data storage area to TDR register, and then SCI1 transmits the data.

When data reception is completed, an RXI1 interrupt request is generated, and this becomes the DMAC0 transfer request. DMAC0 is used to transfer receive data to the receive data storage area.

After transmit data has been transferred 256 times, a DMAC1 interrupt occurs. At this point, the TXI1 interrupt is disabled and the TEI1 interrupt is enabled.

After receive data has been transferred 256 times, a DMAC0 interrupt occurs. At this point, the RXI1 interrupt is disabled and the receive end flag becomes 1. If the transmit end flag is 1 at this time, SCI1 transmission and reception are disabled.

After 256 bytes of data have been transmitted and received, a TEI1 interrupt occurs. At this point, the TEI1 interrupt is disabled and the transmit end flag becomes 1. If the receive end flag is 1 at this time, SCI1 transmission and reception are disabled.

Settings for the peripheral functions used are listed below.

SCI1

- Communication mode: Clock synchronous mode
- SCK1 pin: Outputs the internal clock (master)
- Clock source: PCLKB
- Transfer rate: 38400 bps (BRR register setting value = (PCLKB ÷ (8 × 0.5 × 38400 bps)) -1)
- Transmit operation: Enabled
- Receive operation: Enabled
- Data transfer direction: LSB first
- Interrupts: Transmit end interrupt (TEI1) used, transmit data empty interrupt (TXI1) used, receive data full interrupt (RXI1) used, receive error interrupt (ERI1) used

DMAC0

- Activation source: RXI1 interrupt request
  : The IR flag for the RXI1 interrupt is set to 0 when data transfer starts
- Transfer source address: SCI1.RDR register
- Transfer source address update mode: Fixed address
- Transfer destination address: RAM (start address of the receive data storage area)
- Transfer destination address update mode: Increment
- Transfer mode: Normal transfer
- Data transfer size: 8-bit
- Number of transfer operations: 256
- Interrupt: Transfer end interrupt (DMAC0I) used

DMAC1

- Activation source: TXI1 interrupt request
  : The IR flag for the TXI1 interrupt is set to 0 when data transfer starts
- Transfer source address: RAM (start address of the transmit data storage area)
- Transfer source address update mode: Increment
- Transfer destination address: SCI1.TDR register
- Transfer destination address update mode: Fixed address
- Transfer mode: Normal transfer
- Data transfer size: 8-bit
- Number of transfer operations: 256
- Interrupt: Transfer end interrupt (DMAC1I) used

IRQ1 input pin

- Detection method: Falling edge
- Digital filter: Disabled
- Interrupts: Not used

5.1 Operation Overview

5.1.1 Transmit Operation

1. Initialization
   After initialization, wait for input from a switch to start transmission/reception.

2. Detecting input from a switch to start transmission/reception
   When input from a switch to start transmission/reception is detected, set the IR flag for the IRQ1 interrupt to 0. Read the transmit end flag and receive end flag. If transmission and reception have both ended, then set the transmit end flag to 0 (transmitting). Set the transfer source address and the number of transfer operations for DMAC1, and enable DMA transfer.
   Set the SCI1.SCR.TIE, RIE, TE, RE, and TEIE bits to 1 at the same time to enable a transmission and reception. By setting the SCI1.SCR.TIE and TE bits to 1 at the same time, the IR flag for the TXI1 interrupt becomes 1.

3. Starting data transfer
   After the TXI1 interrupt is enabled, DMAC1 is activated and the IR flag for the TXI1 interrupt becomes 0. The first byte of transmit data is transferred from the RAM's transmit data storage area to the SCI1.TDR register.

4. Starting data transmission
   The data is transferred from the SCI1.TDR register to the SCI1.TSR register, the IR flag for the TXI1 interrupt becomes 1, and the first byte of transmit data is output from the TXD1 pin. DMAC1 is activated by a TXI1 interrupt request, and the second byte of transmit data is transferred.

5. DMAC1I interrupt
   After 256 data transfers have ended, a DMAC1I interrupt request occurs. In the DMAC1I interrupt handling, disable the TXI1 interrupt and enable the TEI1 interrupt.

6. TEI1 interrupt
   When the 256th byte is transmitted, the SCI1.TDR register is not updated, so a TEI1 interrupt request occurs. In the TEI1 interrupt handling, disable the TEI1 interrupt and set the transmit end flag to 1 (transmission ended). If the receive end flag is 1 at this time, transmit and receive operations are disabled. Then operation is repeated from step 2 above.

Figure 5.1 shows the Timing Diagram of the Transmit Operation.

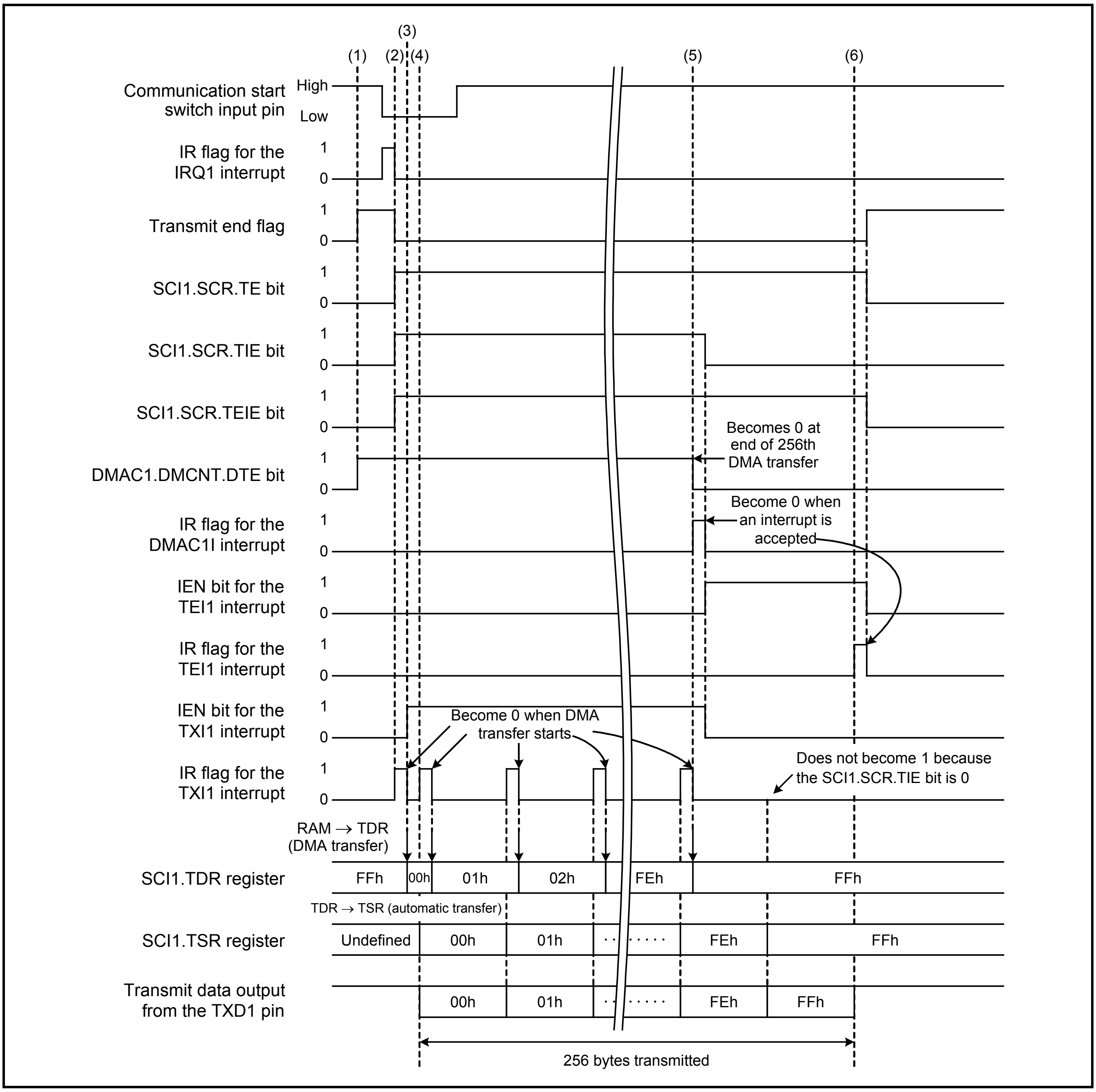

(3)
(1)
(2)
(4)
(5)
(6)
Communication start switch input pin
High
Low
IR flag for the IRQ1 interrupt
Transmit end flag
SCI1.SCR.TE bit
SCI1.SCR.TIE bit
SCI1.SCR.TEIE bit
DMAC1.DMCNT.DTE bit
IR flag for the DMAC1I interrupt
IEN bit for the TEI1 interrupt
IR flag for the TEI1 interrupt
IEN bit for the TXI1 interrupt
IR flag for the TXI1 interrupt
1
0
Becomes 0 at end of 256th DMA transfer
Become 0 when an interrupt is accepted
Become 0 when DMA transfer starts
Does not become 1 because the SCI1.SCR.TIE bit is 0
RAM → TDR (DMA transfer)
SCI1.TDR register
FFh
00h
01h
02h
FEh
FFh
TDR → TSR (automatic transfer)
SCI1.TSR register
Undefined
00h
01h
FEh
FFh
Transmit data output from the TXD1 pin
00h
01h
FEh
FFh
256 bytes transmitted

Figure 5.1 Timing Diagram of the Transmit Operation

5.1.2 Receive Operation

1. Initialization
   After initialization, wait for input from a switch to start transmission/reception.

2. Detecting input from a switch to start transmission/reception
   When input from a switch to start transmission/reception is detected, set the IR flag for the IRQ1 interrupt to 0. Read the transmit end flag and receive end flag. If transmission and reception have both ended, then set the receive end flag to 0 (receiving). Set the transfer destination address and the number of transfer operations for DMAC0, and enable DMA transfer.
   Set the SCI1.SCR.TIE, RIE, TE, RE, and TEIE bits to 1 at the same time to enable a transmission and reception, and enable the RXI1 interrupt.

3. Data reception end
   After the first byte of data is received, transfer the data from the SCI1.RSR register to the SCI1.RDR register, and the IR flag for the RXI1 interrupt becomes 1.

4. Starting data transfer
   DMAC0 is activated by the RXI1 interrupt request, and the IR flag for the RXI1 interrupt becomes 0. Then transfer the first byte of receive data from the SCI1.RDR register to the RAM's receive data storage area.

5. DMAC0I interrupt
   When the last bit of the 256th byte is transferred, a DMAC0I interrupt request is generated. In the DMAC0I interrupt handling, disable the RXI1 interrupt and set the receive end flag to 1 (reception ended). If the receive end flag is 1 (transmission ended) at this time, transmit and receive operations are disabled. Then operation is repeated from step 2 above.

Figure 5.2 shows the Timing Diagram of the Receive Operation.

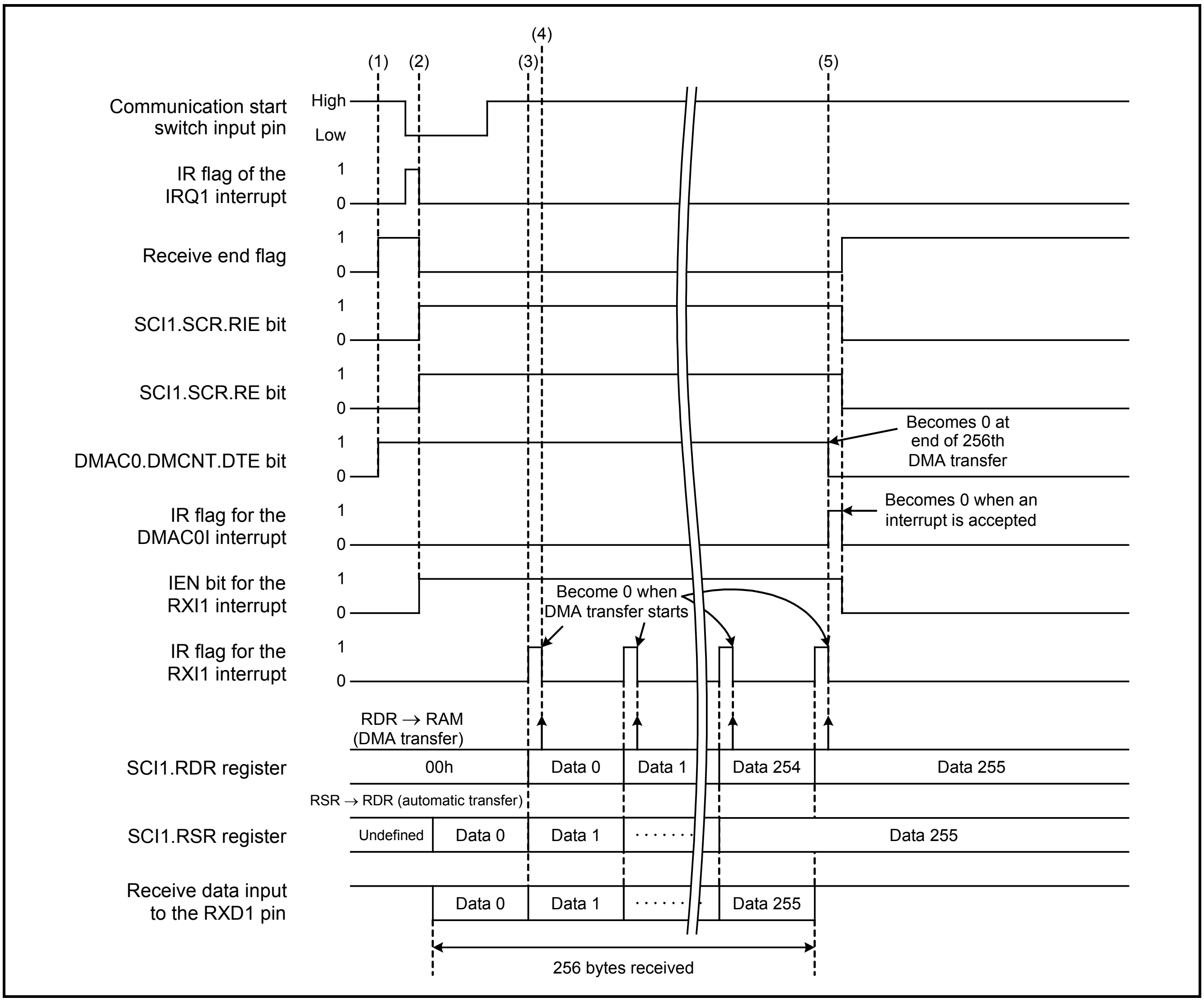

(1)
(2)
(3)
(4)
(5)
Communication start switch input pin
High
Low
IR flag of the IRQ1 interrupt
Receive end flag
SCI1.SCR.RIE bit
SCI1.SCR.RE bit
DMAC0.DMCNT.DTE bit
IR flag for the DMAC0I interrupt
IEN bit for the RXI1 interrupt
IR flag for the RXI1 interrupt
1
0
Becomes 0 at end of 256th DMA transfer
Becomes 0 when an interrupt is accepted
Become 0 when DMA transfer starts
RDR → RAM (DMA transfer)
SCI1.RDR register
00h
Data 0
Data 1
Data 254
Data 255
RSR → RDR (automatic transfer)
SCI1.RSR register
Undefined
Data 0
Data 1
Data 255
Receive data input to the RXD1 pin
Data 0
Data 1
Data 255
256 bytes received

Figure 5.2 Timing Diagram of the Receive Operation

Notes on Implementing the Sample Code Into the User System

Note that the following may occur when implementing the sample code accompanying this application note into the user system.

- When using the code in slave mode, if user defined interrupt handling makes the interrupts defined in the sample wait for a lengthy amount of time, the sample code may operate erroneously.
- When using an external clock source as the synchronous clock, after the DMAC updates the TDR register, wait for at least five cycles of the PCLK before inputting the transmit clock. After the TDR register has been updated, inputting the transmit clock within four cycles of the PCLK may cause the sample code to operate erroneously.

5.2 File Composition

Table 5.1 lists the Files Used in the Sample Code. Files generated by the integrated development environment are not included in this table.

Table 5.1 Files Used in the Sample Code

File Name	Outline
main.c	Main processing
r_init_stop_module.c	Stop processing for active peripheral functions after a reset
r_init_stop_module.h	Header file for r_init_stop_module.c
r_init_non_existent_port_init.c	Nonexistent port initialization
r_init_non_existent_port_init.h	Header file for r_init_non_existent_port_init.c
r_init_clock.c	Clock initialization
r_init_clock.h	Header file for r_init_clock.c

5.3 Option-Setting Memory

Table 5.2 lists the Option-Setting Memory Configured in the Sample Code. When necessary, set a value suited to the user system.

Table 5.2 Option-Setting Memory Configured in the Sample Code

Symbol	Address	Setting Value	Contents
OFS0	FFFF FF8Fh to FFFF FF8Ch	FFFF FFFFh	The IWDT is stopped after a reset. The WDT is stopped after a reset.
OFS1	FFFF FF8Bh to FFFF FF88h	FFFF FFFFh	The voltage monitor 0 reset is disabled after a reset. HOCO oscillation is disabled after a reset.
MDES	FFFF FF83h to FFFF FF80h	FFFF FFFFh	Little endian

5.4 Constants

Table 5.3 lists the Constants Used in the Sample Code.

Table 5.3 Constants Used in the Sample Code

Constant Name	Setting Value	Contents
MASTER	00h	SCI1.SCR.CKE[1:0] bit setting: Internal clock when in master mode
SLAVE	02h	SCI1.SCR.CKE[1:0] bit setting: External clock when in slave mode
SCI_CLK	MASTER	SCI1.SCR.CKE[1:0] bit setting: Master mode selected
BUF_SIZE	256	Size of the transmit data and receive data storage area
DMAC_CNT	BUF_SIZE	Number of DMAC transfers

5.5 Variables

Table 5.4 lists the Global Variables.

Table 5.4 Global Variables

Type	Variable Name	Contents	Function Used
unsigned char	rcv_end_flag	Receive end flag 0: Receiving 1: Reception ended	main, Excep_DMAC_DMAC0I, Excep_SCI1_TEI1
unsigned char	trn_end_flag	Transmit end flag 0: Transmitting 1: Transmission ended	main, Excep_DMAC_DMAC0I, Excep_SCI1_TEI1
unsigned char	rcvbuf[BUF_SIZE]	Receive data storage area	dmac0_init, sci1_start
unsigned char	trnbuf[BUF_SIZE]	Transmit data storage area	main, dmac1_init, sci1_start

5.6 Functions

Table 5.5 lists the Functions.

Table 5.5 Functions

Function Name	Outline
main	Main processing
port_init	Port initialization
R_INIT_StopModule	Stop processing for active peripheral functions after a reset
R_INIT_NonExistentPort	Nonexistent port initialization
R_INIT_Clock	Clock initialization
peripheral_init	Peripheral function initialization
sci1_init	SCI1 initialization
dmac0_init	DMAC0 initialization
dmac1_init	DMAC1 initialization
irq_init	IRQ initialization
sci1_start	SCI1 transmission/reception start processing
Excep_DMAC_DMAC0I	DMAC0 transfer end interrupt handling
Excep_DMAC_DMAC1I	DMAC1 transfer end interrupt handling
Excep_SCI1_TEI1	SCI1 transmit end interrupt handling
Excep_SCI1_ERI1	SCI1 receive error interrupt handling

5.7 Function Specifications

The following tables list the sample code function specifications.

main	
Overview	Main processing
Header	None
Declaration	void main(void)
Description	After initialization, SCI1 transmission/reception starts when the transmit/receive start switch input is detected.
Arguments	None
Return values	None

port_init	
Overview	Port initialization
Header	None
Declaration	void port_init(void)
Description	Ports are initialized.
Arguments	None
Return values	None

R_INIT_StopModule	
Overview	Stop processing for active peripheral functions after a reset
Header	r_init_stop_module.h
Declaration	void R_INIT_StopModule(void)
Description	Performs settings to enter the module stop state.
Arguments	None
Return values	None
Remarks	Transition to the module stop state is not performed in the sample code. For details on this functions, refer to the Initial Setting application note for the product used.

R_INIT_NonExistentPort	
Overview	Nonexistent port initialization
Header	non_existent_port_init.h
Declaration	void R_INIT_NonExistentPort (void)
Description	Initialize port direction registers for ports that do not exist in products with less than 100 pins.
Arguments	None
Return values	None
Remarks	The number of pins in the sample code is set for the 100-pin package (PIN_SIZE=100). After this function is called, when writing in byte units to the PDR and PODR registers which have nonexistent ports, set the corresponding bits for nonexistent ports as follows: set the I/O select bits in the PDR registers to 1 and set the output data store bits in the PODR registers to 0. For details on this functions, refer to the Initial Setting application note for the product used.

R_INIT_Clock	
Overview	Clock initialization
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Initialize clocks.
Arguments	None
Return values	None
Remarks	In the sample code, processing is performed so the system clock is used as the PLL clock, and the sub-clock is not used. For details on this functions, refer to the Initial Setting application note for the product used.

peripheral_init	
Overview	Peripheral function initialization
Header	None
Declaration	void peripheral_init (void)
Description	Initializes peripherals functions that are used.
Arguments	None
Return values	None

sci1_init	
Overview	SCI1 initialization
Header	None
Declaration	void sci1_init(void)
Description	Initializes SCI1.
Arguments	None
Return values	None

dmac0_init	
Overview	DMAC0 initialization
Header	None
Declaration	void dmac0_init(void)
Description	Initializes DMAC0.
Arguments	None
Return values	None

dmac1_init	
Overview	DMAC1 initialization
Header	None
Declaration	void dmac1_init(void)
Description	Initializes DMAC1.
Arguments	None
Return values	None

irq_init	
Overview	IRQ initialization
Header	None
Declaration	void irq_init(void)
Description	Initializes IRQ1.
Arguments	None
Return values	None

sci1_start	
Overview	SCI1 transmission/reception start processing
Header	None
Declaration	void sci1_start(void)
Description	Starts SCI1 transmission/reception.
Arguments	None
Return values	None

Excep_DMAC_DMAC0I	
Overview	DMAC0 transfer end interrupt handling
Header	None
Declaration	void Excep_DMAC_DMAC0I(void)
Description	The RXI1 interrupt is disabled, and the receive end flag is set. If the transmit end flag is 1, SCI1 transmission and reception are disabled.
Arguments	None
Return values	None

Excep_DMAC_DMAC1I	
Overview	DMAC1 transfer end interrupt handling
Header	None
Declaration	void Excep_DMAC_DMAC1I(void)
Description	Disables the TXI1 interrupt and enables the TEI1 interrupt.
Arguments	None
Return values	None

Excep_SCI1_TEI1	
Overview	SCI1 transmit end interrupt handling
Header	None
Declaration	void Excep_SCI1_TEI1(void)
Description	Disables the TEI1 interrupt and sets the transmit end flag. If the receive end flag is 1, SCI1 transmission and reception are disabled.
Arguments	None
Return values	None

Excep_SCI1_ERI1	
Overview	SCI1 receive error interrupt handling
Header	None
Declaration	void Excep_SCI1_ERI1(void)
Description	Performs SCI1 reception error processing.
Arguments	None
Return values	None
Remarks	SCI1 reception error processing is not performed in the sample code (infinite loop). Add processing to the user code as needed.

5.8 Flowcharts

5.8.1 Main Processing

Figure 5.3 shows the Main Processing.

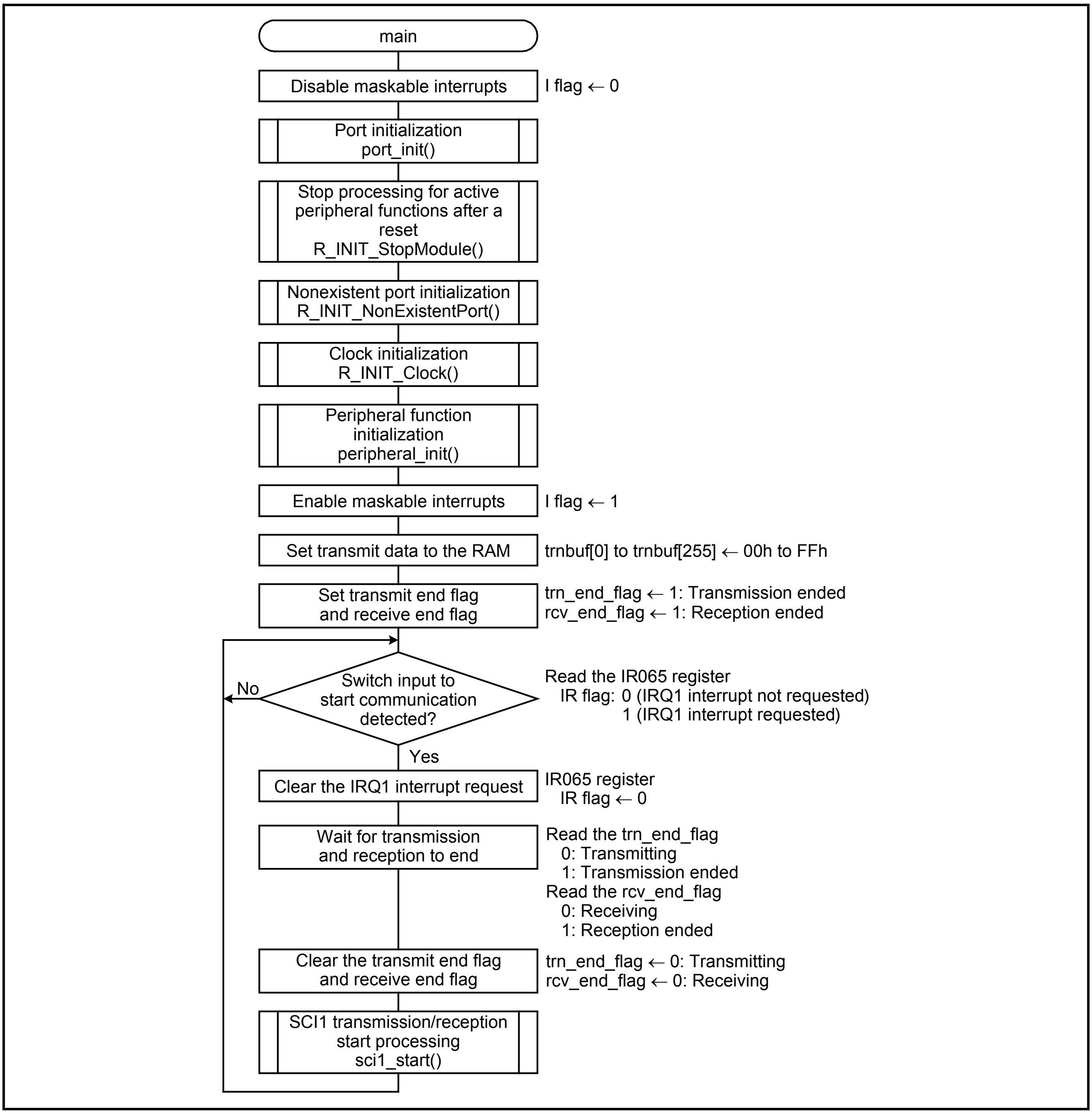

main
Disable maskable interrupts
I flag ← 0
Port initialization
port_init()
Stop processing for active peripheral functions after a reset
R_INIT_StopModule()
Nonexistent port initialization
R_INIT_NonExistentPort()
Clock initialization
R_INIT_Clock()
Peripheral function initialization
peripheral_init()
Enable maskable interrupts
I flag ← 1
Set transmit data to the RAM
trnbuf[0] to trnbuf[255] ← 00h to FFh
Set transmit end flag and receive end flag
trn_end_flag ← 1: Transmission ended
rcv_end_flag ← 1: Reception ended
Switch input to start communication detected?
No
Yes
Read the IR065 register
IR flag: 0 (IRQ1 interrupt not requested)
1 (IRQ1 interrupt requested)
Clear the IRQ1 interrupt request
IR065 register
IR flag ← 0
Wait for transmission and reception to end
Read the trn_end_flag
0: Transmitting
1: Transmission ended
Read the rcv_end_flag
0: Receiving
1: Reception ended
Clear the transmit end flag and receive end flag
trn_end_flag ← 0: Transmitting
rcv_end_flag ← 0: Receiving
SCI1 transmission/reception start processing
sci1_start()

Figure 5.3 Main Processing

5.8.2 Port Initialization

Figure 5.4 shows the Port Initialization.

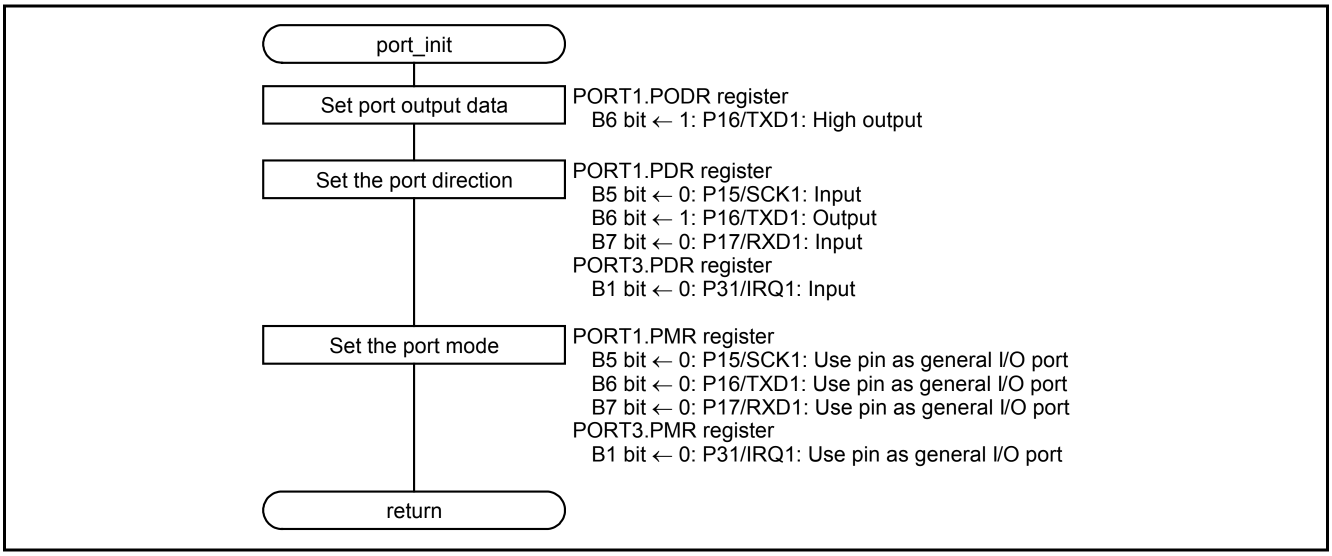

port_init
Set port output data
PORT1.PODR register
B6 bit ← 1: P16/TXD1: High output
Set the port direction
PORT1.PDR register
B5 bit ← 0: P15/SCK1: Input
B6 bit ← 1: P16/TXD1: Output
B7 bit ← 0: P17/RXD1: Input
PORT3.PDR register
B1 bit ← 0: P31/IRQ1: Input
Set the port mode
PORT1.PMR register
B5 bit ← 0: P15/SCK1: Use pin as general I/O port
B6 bit ← 0: P16/TXD1: Use pin as general I/O port
B7 bit ← 0: P17/RXD1: Use pin as general I/O port
PORT3.PMR register
B1 bit ← 0: P31/IRQ1: Use pin as general I/O port
return

Figure 5.4 Port Initialization

5.8.3 Peripheral Function Initialization

Figure 5.5 shows the Peripheral Function Initialization.

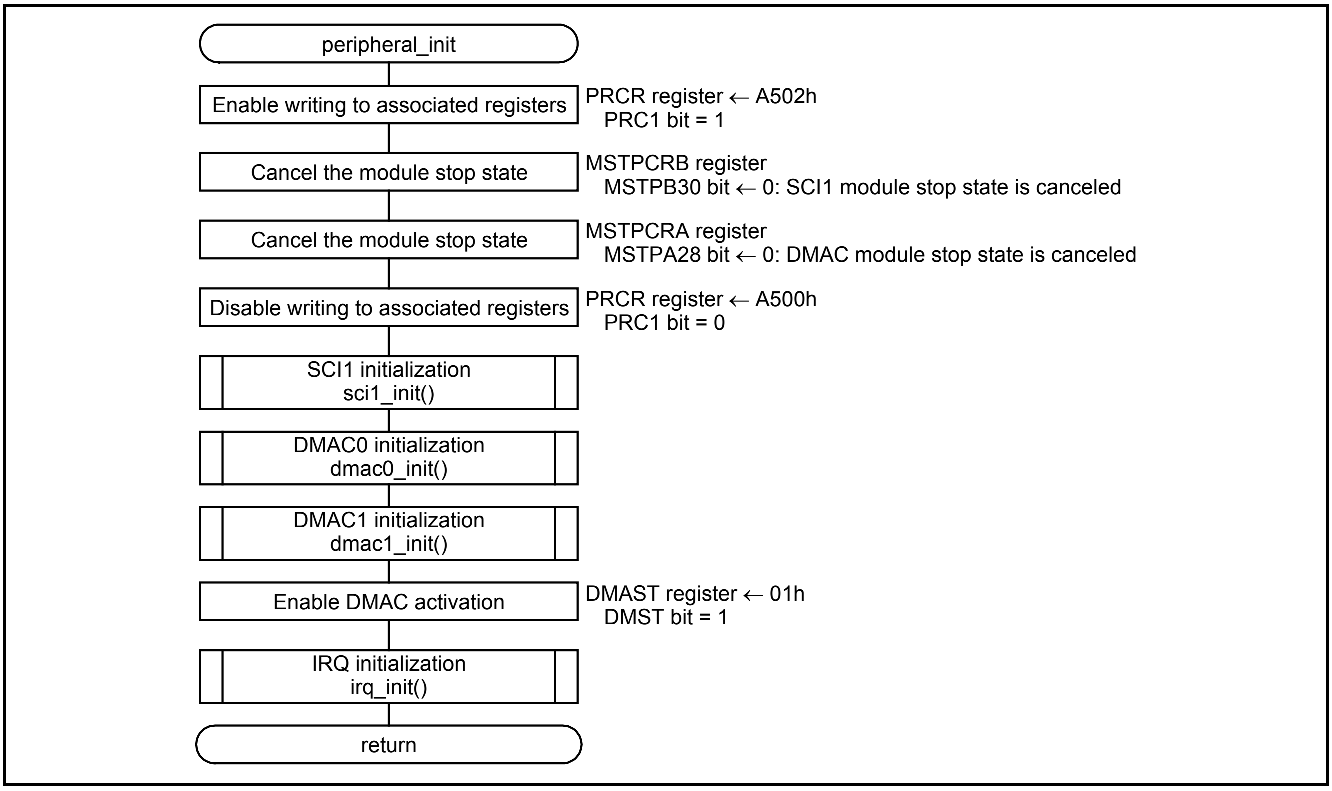
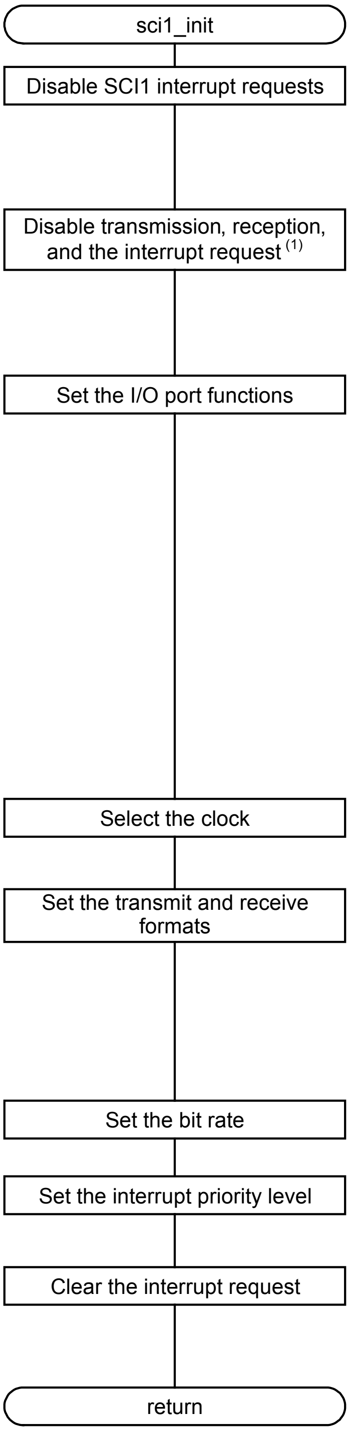


Figure 5.5 Peripheral Function Initialization

5.8.4 SCI1 Initialization

Figure 5.6 and shows the SCI1 initialization.


sci1_init
Disable SCI1 interrupt requests
Disable transmission, reception, and the interrupt request (1)
Set the I/O port functions
Select the clock
Set the transmit and receive formats
Set the bit rate
Set the interrupt priority level
Clear the interrupt request
return

IER1B register
  IEN2 bit ← 0: SCI1.ERI1 interrupt request is disabled
  IEN3 bit ← 0: SCI1.RXI1 interrupt request is disabled
  IEN4 bit ← 0: SCI1.TXI1 interrupt request is disabled
  IEN5 bit ← 0: SCI1.TEI1 interrupt request is disabled

SCI1.SCR register ← 00h
  TEIE bit = 0: TEI interrupt request is disabled
  RE bit = 0: Serial reception is disabled
  TE bit = 0: Serial transmission is disabled
  RIE bit = 0: RXI and ERI interrupt requests are disabled
  TIE bit = 0: TXI interrupt request is disabled

PWPR register
  B0WI bit ← 0: Writing to the PFSWE bit is enabled
PWPR register
  PFSWE bit ← 1: Writing to the PFS register is enabled
P15PFS register ← 0Ah
  PSEL[3:0] bits = 1010b: P15 pin functions as RXD1
P16PFS register ← 0Ah
  PSEL[3:0] bits = 1010b: P16 pin functions as TXD1
P17PFS register ← 0Ah
  PSEL[3:0] bits = 1010b: P17 pin functions as SCK1
PWPR register
  PFSWE bit ← 0: Writing to the PFS register is disabled
PWPR register
  B0WI bit ← 1: Writing to the PFSWE bit is disabled
PORT1.PMR register
  B5 bit ← 1: P15/RXD1 pin used as I/O port for peripheral functions
  B7 bit ← 1: P17/SCK1 pin used as I/O port for peripheral functions

SCI1.SCR register
  Bits CKE[1:0] ← SCK_CLK:Set the internal clock when the MCU is the master, and set the external clock when the MCU is the slave

SCI1.SMR register ← 80h
  Bits CKS[1:0] = 00b: PCLK
  CM bit = 1: Clock synchronous mode
SCI1.SCMR register ← F2h
  SMIF bit = 0: Serial communications interface mode
  SINV bit = 0: TDR contents are transmitted as they are.
    Receive data is stored as it is in RDR.
  SDIR bit = 0: Transfer with LSB-first

SCI1.BRR register ← 162: $161.76 = \left(\frac{25MHz}{8\times 2^{-1}\times 38400bps}\right)-1$

IPR218 register
  Bits IPR[3:0] ← 0001b: Interrupt priority level for SCI1.RXI1, TXI1, and TEI1 is level 1

IR219 register
  IR flag ← 0: SCI1.RXI1 interrupt request not generated
IR220 register
  IR flag ← 0: SCI1.TXI1 interrupt request not generated

Note:
1. After writing to the SCR register, read the register to confirm that the value was written.

Figure 5.6 SCI1 Initialization

5.8.5 DMAC0 Initialization

Figure 5.7 shows the DMAC0 Initialization.

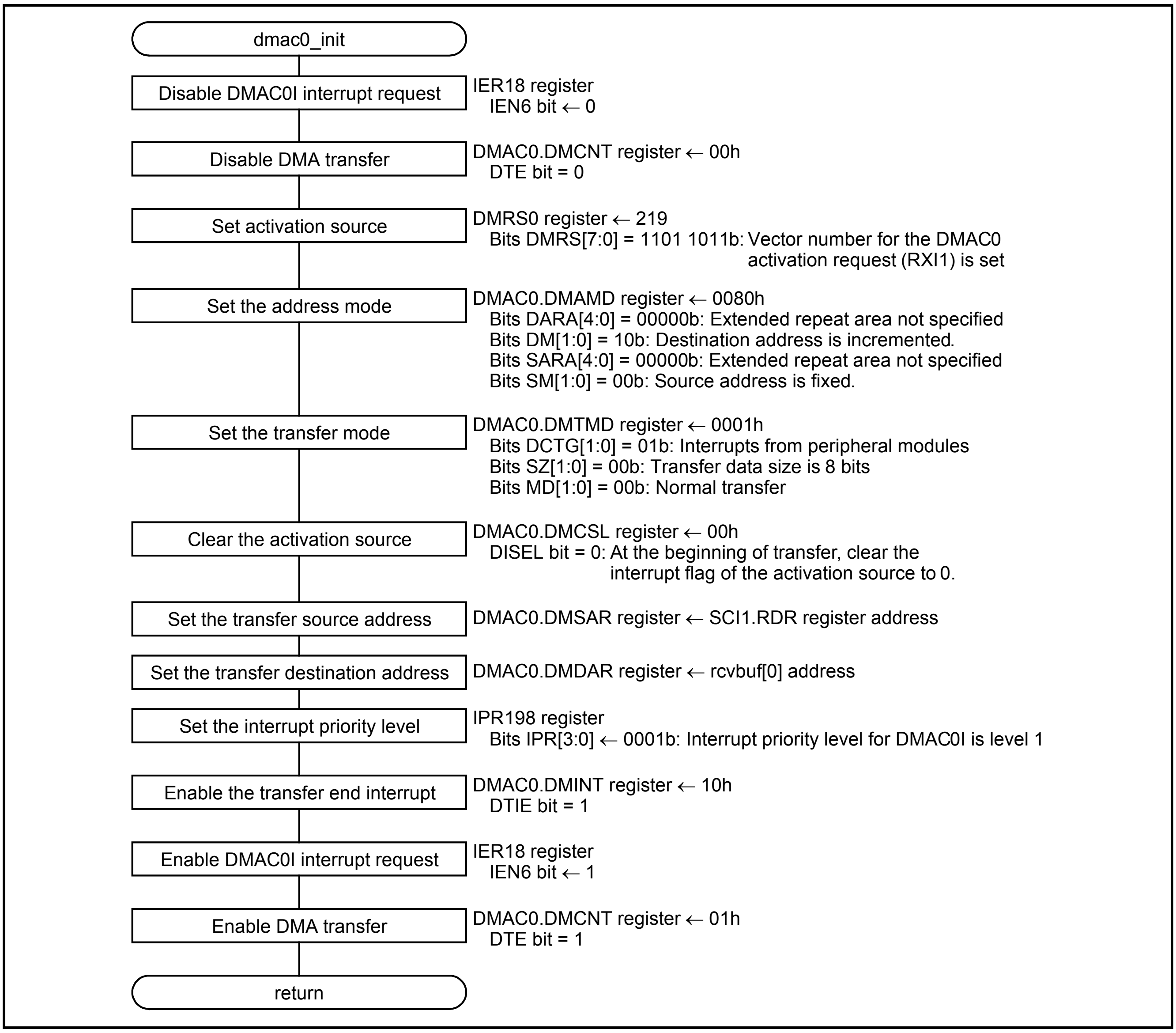

dmac0_init
Disable DMAC0I interrupt request
IER18 register
IEN6 bit ← 0
Disable DMA transfer
DMAC0.DMCNT register ← 00h
DTE bit = 0
Set activation source
DMRS0 register ← 219
Bits DMRS[7:0] = 1101 1011b: Vector number for the DMAC0 activation request (RXI1) is set
Set the address mode
DMAC0.DMAMD register ← 0080h
Bits DARA[4:0] = 00000b: Extended repeat area not specified
Bits DM[1:0] = 10b: Destination address is incremented.
Bits SARA[4:0] = 00000b: Extended repeat area not specified
Bits SM[1:0] = 00b: Source address is fixed.
Set the transfer mode
DMAC0.DMTMD register ← 0001h
Bits DCTG[1:0] = 01b: Interrupts from peripheral modules
Bits SZ[1:0] = 00b: Transfer data size is 8 bits
Bits MD[1:0] = 00b: Normal transfer
Clear the activation source
DMAC0.DMCSL register ← 00h
DISEL bit = 0: At the beginning of transfer, clear the interrupt flag of the activation source to 0.
Set the transfer source address
DMAC0.DMSAR register ← SCI1.RDR register address
Set the transfer destination address
DMAC0.DMDAR register ← rcvbuf[0] address
Set the interrupt priority level
IPR198 register
Bits IPR[3:0] ← 0001b: Interrupt priority level for DMAC0I is level 1
Enable the transfer end interrupt
DMAC0.DMINT register ← 10h
DTIE bit = 1
Enable DMAC0I interrupt request
IER18 register
IEN6 bit ← 1
Enable DMA transfer
DMAC0.DMCNT register ← 01h
DTE bit = 1
return

Figure 5.7 DMAC0 Initialization

5.8.6 DMAC1 Initialization

Figure 5.8 shows the DMAC1 Initialization.

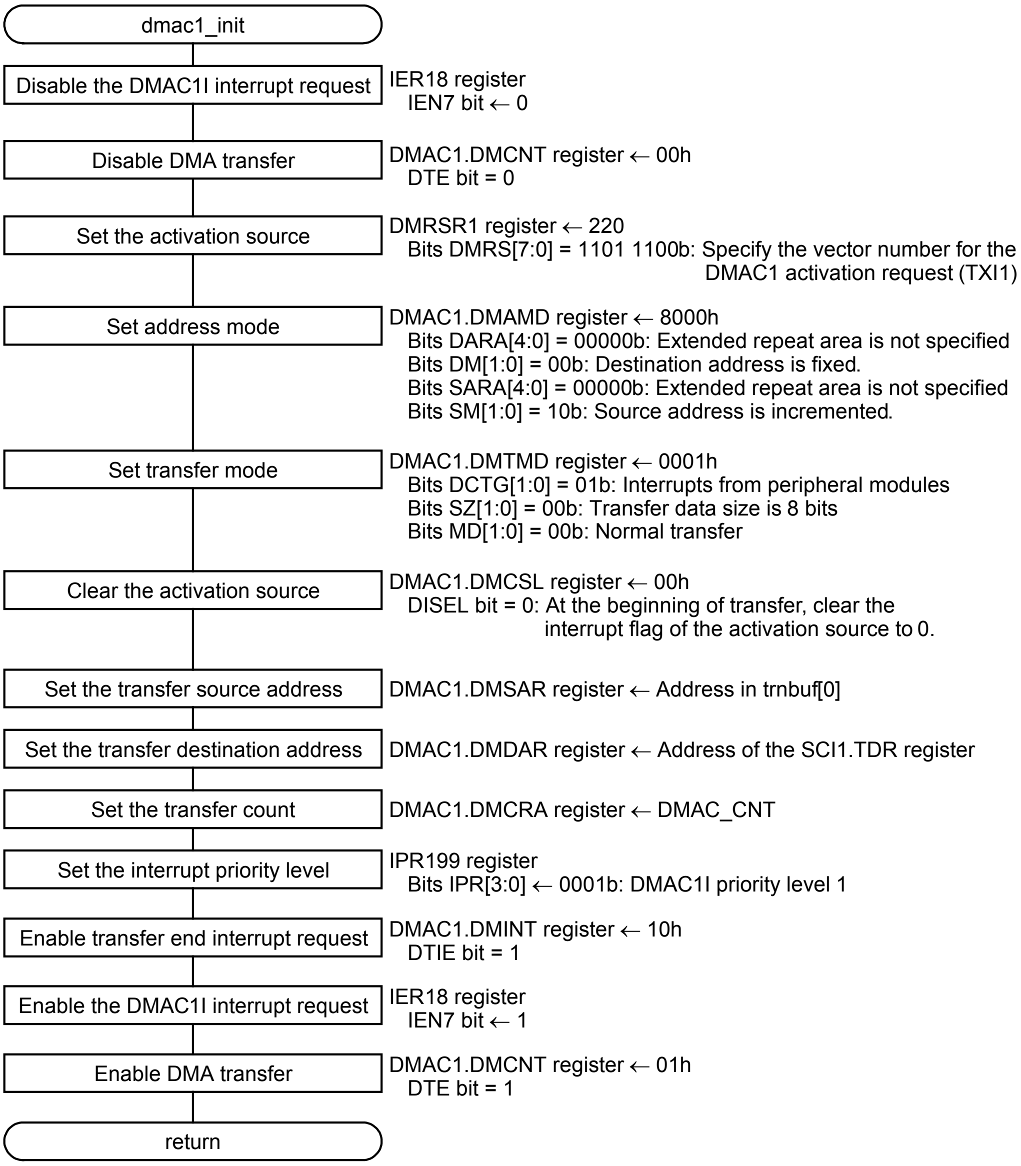

dmac1_init
Disable the DMAC1I interrupt request
IER18 register
IEN7 bit ← 0
Disable DMA transfer
DMAC1.DMCNT register ← 00h
DTE bit = 0
Set the activation source
DMRSR1 register ← 220
Bits DMRS[7:0] = 1101 1100b: Specify the vector number for the DMAC1 activation request (TXI1)
Set address mode
DMAC1.DMAMD register ← 8000h
Bits DARA[4:0] = 00000b: Extended repeat area is not specified
Bits DM[1:0] = 00b: Destination address is fixed.
Bits SARA[4:0] = 00000b: Extended repeat area is not specified
Bits SM[1:0] = 10b: Source address is incremented.
Set transfer mode
DMAC1.DMTMD register ← 0001h
Bits DCTG[1:0] = 01b: Interrupts from peripheral modules
Bits SZ[1:0] = 00b: Transfer data size is 8 bits
Bits MD[1:0] = 00b: Normal transfer
Clear the activation source
DMAC1.DMCSL register ← 00h
DISEL bit = 0: At the beginning of transfer, clear the interrupt flag of the activation source to 0.
Set the transfer source address
DMAC1.DMSAR register ← Address in trnbuf[0]
Set the transfer destination address
DMAC1.DMDAR register ← Address of the SCI1.TDR register
Set the transfer count
DMAC1.DMCRA register ← DMAC_CNT
Set the interrupt priority level
IPR199 register
Bits IPR[3:0] ← 0001b: DMAC1I priority level 1
Enable transfer end interrupt request
DMAC1.DMINT register ← 10h
DTIE bit = 1
Enable the DMAC1I interrupt request
IER18 register
IEN7 bit ← 1
Enable DMA transfer
DMAC1.DMCNT register ← 01h
DTE bit = 1
return

Figure 5.8 DMAC1 Initialization

5.8.7 IRQ Initialization

Figure 5.9 shows the IRQ Initialization.

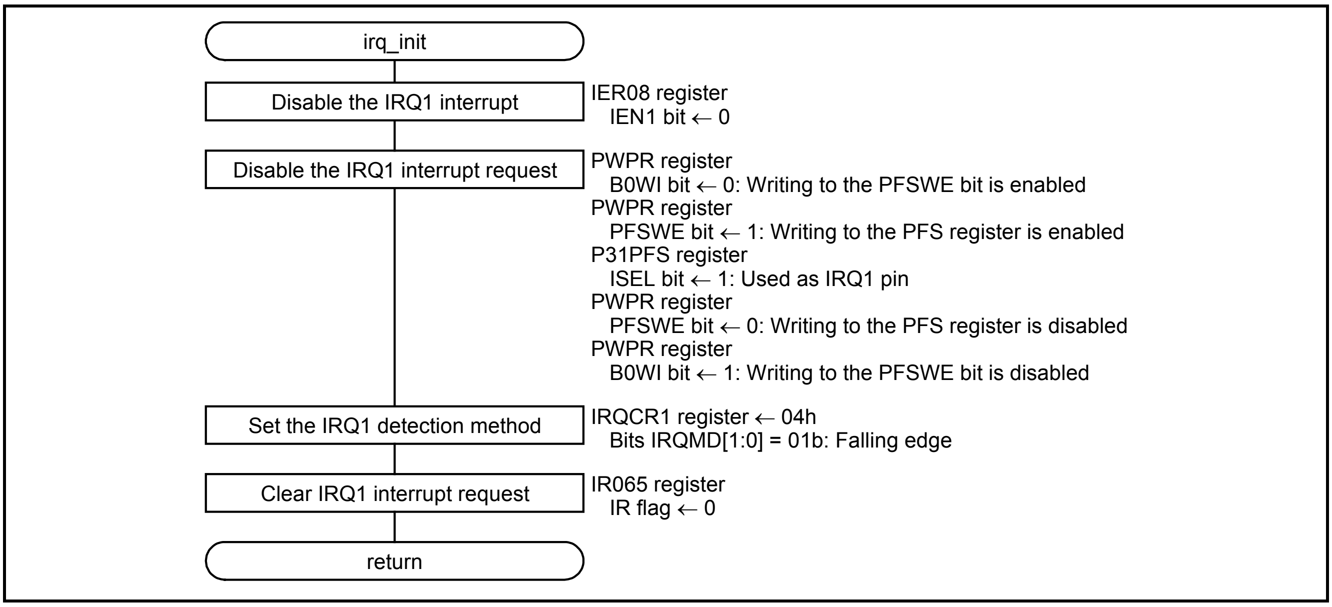

irq_init
Disable the IRQ1 interrupt
IER08 register
IEN1 bit ← 0
Disable the IRQ1 interrupt request
PWPR register
B0WI bit ← 0: Writing to the PFSWE bit is enabled
PWPR register
PFSWE bit ← 1: Writing to the PFS register is enabled
P31PFS register
ISEL bit ← 1: Used as IRQ1 pin
PWPR register
PFSWE bit ← 0: Writing to the PFS register is disabled
PWPR register
B0WI bit ← 1: Writing to the PFSWE bit is disabled
Set the IRQ1 detection method
IRQCR1 register ← 04h
Bits IRQMD[1:0] = 01b: Falling edge
Clear IRQ1 interrupt request
IR065 register
IR flag ← 0
return

Figure 5.9 IRQ Initialization

5.8.8 SCI1 Transmission/Reception Start Processing

Figure 5.10 shows the SCI1 Transmission/Reception Start Processing.

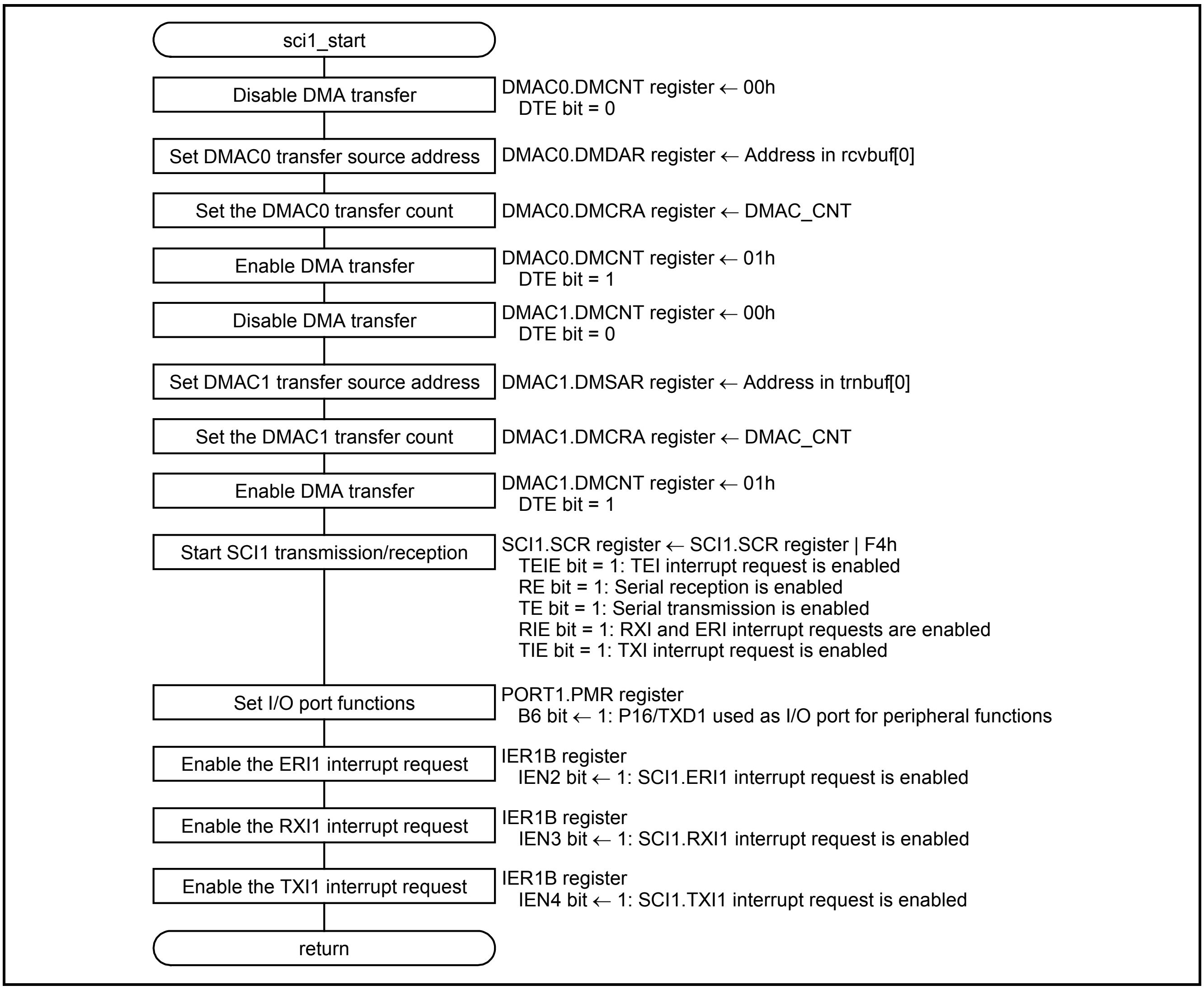
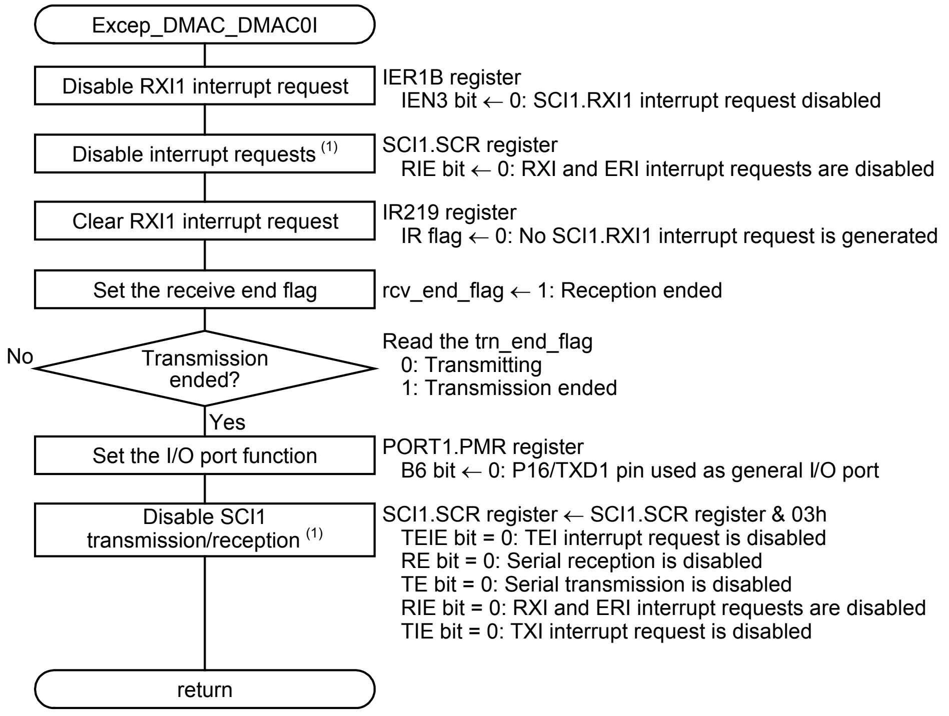

sci1_start
Disable DMA transfer
DMAC0.DMCNT register ← 00h
DTE bit = 0
Set DMAC0 transfer source address
DMAC0.DMDAR register ← Address in rcvbuf[0]
Set the DMAC0 transfer count
DMAC0.DMCRA register ← DMAC_CNT
Enable DMA transfer
DMAC0.DMCNT register ← 01h
DTE bit = 1
Disable DMA transfer
DMAC1.DMCNT register ← 00h
DTE bit = 0
Set DMAC1 transfer source address
DMAC1.DMSAR register ← Address in trnbuf[0]
Set the DMAC1 transfer count
DMAC1.DMCRA register ← DMAC_CNT
Enable DMA transfer
DMAC1.DMCNT register ← 01h
DTE bit = 1
Start SCI1 transmission/reception
SCI1.SCR register ← SCI1.SCR register | F4h
TEIE bit = 1: TEI interrupt request is enabled
RE bit = 1: Serial reception is enabled
TE bit = 1: Serial transmission is enabled
RIE bit = 1: RXI and ERI interrupt requests are enabled
TIE bit = 1: TXI interrupt request is enabled
Set I/O port functions
PORT1.PMR register
B6 bit ← 1: P16/TXD1 used as I/O port for peripheral functions
Enable the ERI1 interrupt request
IER1B register
IEN2 bit ← 1: SCI1.ERI1 interrupt request is enabled
Enable the RXI1 interrupt request
IER1B register
IEN3 bit ← 1: SCI1.RXI1 interrupt request is enabled
Enable the TXI1 interrupt request
IER1B register
IEN4 bit ← 1: SCI1.TXI1 interrupt request is enabled
return

Figure 5.10 SCI1 Transmission/Reception Start Processing

5.8.9 DMAC0 Transfer End Interrupt Handling

Figure 5.11 shows the DMAC0 Transfer End Interrupt Handling.

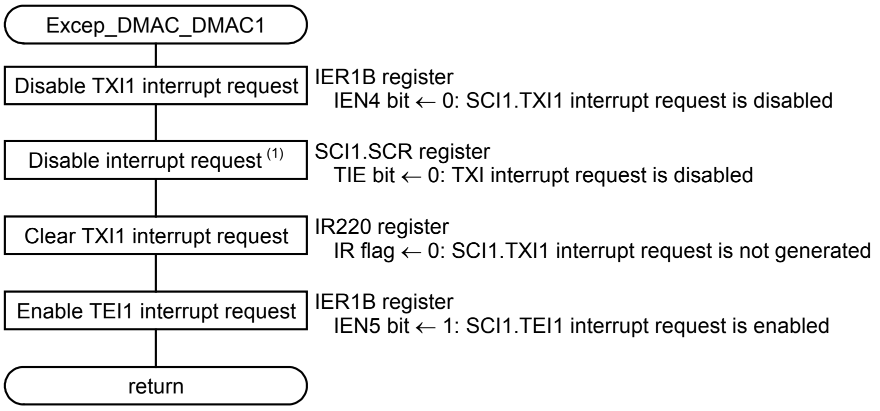

Excep_DMAC_DMAC0I
Disable RXI1 interrupt request
IER1B register
IEN3 bit ← 0: SCI1.RXI1 interrupt request disabled
Disable interrupt requests (1)
SCI1.SCR register
RIE bit ← 0: RXI and ERI interrupt requests are disabled
Clear RXI1 interrupt request
IR219 register
IR flag ← 0: No SCI1.RXI1 interrupt request is generated
Set the receive end flag
rcv_end_flag ← 1: Reception ended
No
Transmission ended?
Read the trn_end_flag
0: Transmitting
1: Transmission ended
Yes
Set the I/O port function
PORT1.PMR register
B6 bit ← 0: P16/TXD1 pin used as general I/O port
Disable SCI1 transmission/reception (1)
SCI1.SCR register ← SCI1.SCR register & 03h
TEIE bit = 0: TEI interrupt request is disabled
RE bit = 0: Serial reception is disabled
TE bit = 0: Serial transmission is disabled
RIE bit = 0: RXI and ERI interrupt requests are disabled
TIE bit = 0: TXI interrupt request is disabled
return

Note:
1. After writing to the SCR register, read the register and confirm that the read value is the written value.

Figure 5.11 DMAC0 Transfer End Interrupt Handling

5.8.10 DMAC1 Transfer End Interrupt Handling

Figure 5.12 shows the DMAC1 Transfer End Interrupt Handling.

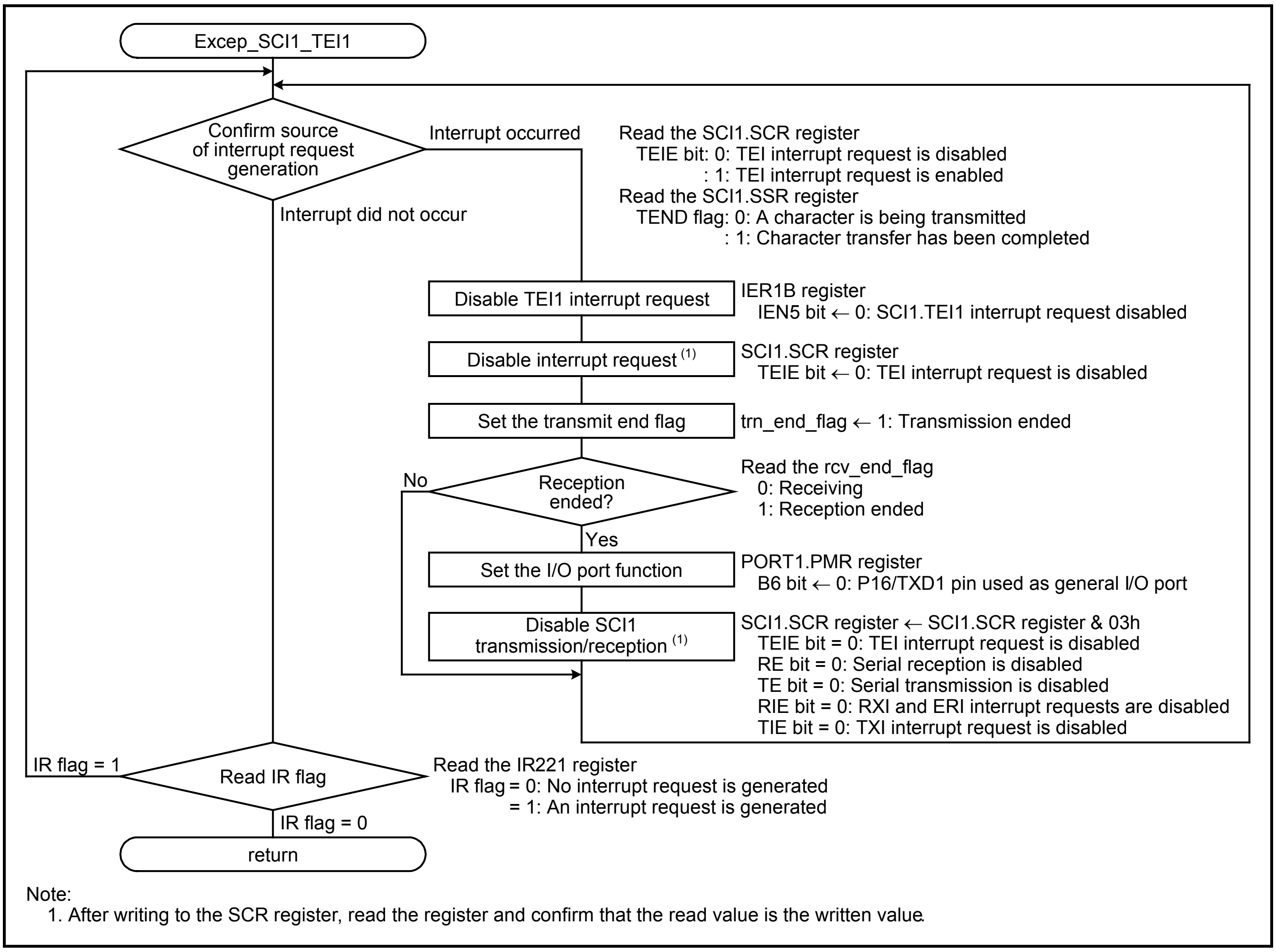

Excep_DMAC_DMAC1
Disable TXI1 interrupt request
IER1B register
IEN4 bit ← 0: SCI1.TXI1 interrupt request is disabled
Disable interrupt request (1)
SCI1.SCR register
TIE bit ← 0: TXI interrupt request is disabled
Clear TXI1 interrupt request
IR220 register
IR flag ← 0: SCI1.TXI1 interrupt request is not generated
Enable TEI1 interrupt request
IER1B register
IEN5 bit ← 1: SCI1.TEI1 interrupt request is enabled
return

Note:
1. After setting the TIE bit, read the TIE bit and confirm it is the value set.

Figure 5.12 DMAC1 Transfer End Interrupt Handling

5.8.11 SCI1 Transmit End Interrupt Handling

Figure 5.13 shows the SCI1 Transmit End Interrupt Handling.


Excep_SCI1_TEI1
Confirm source of interrupt request generation
Interrupt occurred
Read the SCI1.SCR register
TEIE bit: 0: TEI interrupt request is disabled
: 1: TEI interrupt request is enabled
Read the SCI1.SSR register
TEND flag: 0: A character is being transmitted
: 1: Character transfer has been completed
Interrupt did not occur
Disable TEI1 interrupt request
IER1B register
IEN5 bit ← 0: SCI1.TEI1 interrupt request disabled
Disable interrupt request (1)
SCI1.SCR register
TEIE bit ← 0: TEI interrupt request is disabled
Set the transmit end flag
trn_end_flag ← 1: Transmission ended
No
Reception ended?
Read the rcv_end_flag
0: Receiving
1: Reception ended
Yes
Set the I/O port function
PORT1.PMR register
B6 bit ← 0: P16/TXD1 pin used as general I/O port
Disable SCI1 transmission/reception (1)
SCI1.SCR register ← SCI1.SCR register & 03h
TEIE bit = 0: TEI interrupt request is disabled
RE bit = 0: Serial reception is disabled
TE bit = 0: Serial transmission is disabled
RIE bit = 0: RXI and ERI interrupt requests are disabled
TIE bit = 0: TXI interrupt request is disabled
IR flag = 1
Read IR flag
Read the IR221 register
IR flag = 0: No interrupt request is generated
= 1: An interrupt request is generated
IR flag = 0
return

Note:
1. After writing to the SCR register, read the register and confirm that the read value is the written value.

Figure 5.13 SCI1 Transmit End Interrupt Handling

5.8.12 SCI1 Receive Error Interrupt Handling

Figure 5.14 shows the SCI1 Receive Error Interrupt Handling.

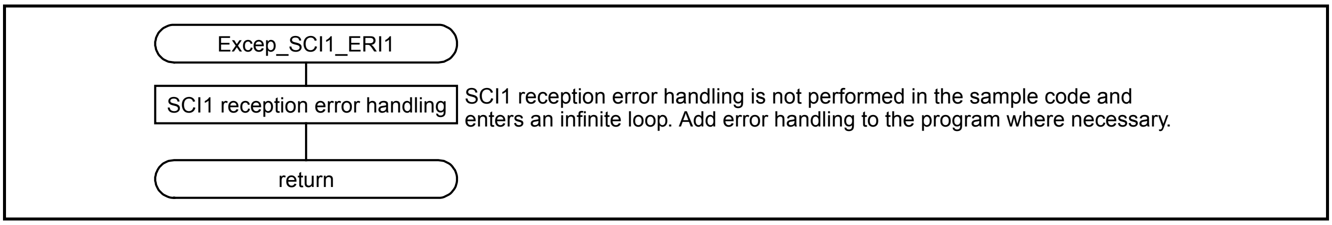

Excep_SCI1_ERI1
SCI1 reception error handling
SCI1 reception error handling is not performed in the sample code and enters an infinite loop. Add error handling to the program where necessary.
return

Figure 5.14 SCI1 Receive Error Interrupt Handling

6 Applying This Application Note to the RX21A or RX220 Group

The sample code accompanying this application note has been confirmed to operate with the RX210 Group. To make the sample code operate with the RX21A or RX220 Group, use this application note in conjunction with the Initial Setting application note for each group.

For details on using this application note with the RX21A and RX220 Groups, refer to "5. Applying the RX210 Group Application Note to the RX21A Group" in the RX21A Group Initial Setting application note, and "4. Applying the RX210 Group Application Note to the RX220 Group" in the RX220 Group Initial Setting application note.

Note: • When using the RX21A Group, SCI0 and SCI12 are not available. Use SCI1, SCI5, SCI6, SCI8, or SCI9.
When using the RX220 Group, SCI0, SCI8, and SCI12 are not available. Use SCI1, SCI5, SCI6, or SCI9.

7 Sample Code

Sample code can be downloaded from the Renesas Electronics website.

8 Reference Documents

User’s Manual: Hardware
RX210 Group User's Manual: Hardware Rev.1.50 (R01UH0037EJ)
RX21A Group User’s Manual: Hardware Rev.1.00 (R01UH0251EJ)
RX220 Group User’s Manual: Hardware Rev.1.10 (R01UH0292EJ)
The latest versions can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

User’s Manual: Development Tools
RX Family C/C++ Compiler Package V.1.01 User’s Manual Rev.1.00 (R20UT0570EJ)
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website
http://www.renesas.com

Inquiries
http://www.renesas.com/contact/

REVISION HISTORY	RX210, RX21A, and RX220 Groups Application Note Clock Synchronous SCIc Communication Using the DMACA

Rev.	Date	Description	
		Page	Summary
1.00	Apr. 1, 2013	—	First edition issued
1.01	July 1, 2014	1	Products: Added the RX21A and RX220 Groups.
		4	3. Reference Application Notes: Added the Initial Setting application notes for the RX21A and RX220 Groups.
		13,14	Modified the description of reference application note in the following functions: R_INIT_StopModule, R_INIT_NonExistentPort, and R_INIT_Clock.
		25	6. Applying This Application Note to the RX21A or RX220 Group: Added.
		26	8. Reference Documents: Added the User’s Manual: Hardware for the RX21A and RX220 Groups.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. 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.
2. 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.
3. 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.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "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 etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). 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 for which it is not intended. 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 which the product is not intended by Renesas Electronics.
6. 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.
7. 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 systems manufactured by you.
8. 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.
9. 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. You should not use Renesas Electronics products or 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. When exporting the Renesas Electronics 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.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
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.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0