RX Family, H8S Family

H8S to RX Migration Guide: Serial Communications Interface

R01AN4046EJ0100
Rev.1.00
Nov 13, 2017

Summary

This application note explains how to migrate from the SCI clocked synchronous mode on the H8S Family to the SCI clock synchronous mode on the RX Family, and from the SCI asynchronous mode on the H8S Family to the SCI asynchronous mode on the RX Family.

Target Devices

- RX Family
- H8S Family

An example of migrating from the H8S Family to the RX Family is presented, with the RX Family represented by the RX231 Group and the H8S Family represented by the H8S/2378 Group. When using this application note with other microcontrollers, appropriate changes should be made to match the specifications of the microcontroller used and thorough evaluation should be performed.

Devices on Which Operation Has Been Confirmed

- RX Family: RX231
- H8S Family: H8S/2378

Table Differences in Terminology between RX Family and H8S Family

Item	RX Family	H8S Family
Channel names	SCIn	Channel n
Peripheral function operating clocks	Peripheral module clocks (PCLKA, PCLKB, PCLKC, and PCLKD)	ϕ

Contents

1. Points of Difference between Serial Communications Interfaces

1.1 Points of Difference between Functions

Table 1.1 lists points of difference between the serial communications interface functions.

Table 1.2 lists the functions of the SCI channels on the RX231 Group.

Table 1.1 Points of Difference between Serial Communications Interfaces

Item		RX (RX231)	H8S (H8S/2378)
Channels		SCI0, SCI1, SCI5, SCI6, SCI8, SCI9, SCI12	Channel 0, Channel 1, Channel 2, Channel 3, Channel 4
Serial communication modes		• Asynchronous • Clock synchronous • Smart card interface • Simple I^2C bus • Simple SPI bus	• Asynchronous • Clock synchronous • Smart Card interface
Transfer speed		Bit rate specifiable by on-chip baud rate generator.	
Full-duplex communication		• Transmitter: Continuous transmission possible using double-buffer structure. • Receiver: Continuous reception possible using double-buffer structure.	
Data transfer		Selectable between LSB-first or MSB-first transfer*[1].	Selectable between LSB-first and MSB-first (excluding asynchronous 7-bit data transfer)
Interrupt sources		• Transmit end • Transmit data empty • Receive data full • Receive error • Completion of generation of start condition, restart condition, or stop condition (simple I^2C mode)	• Transmit end • Transmit data empty • Receive data full • Receive error
Low power consumption function		The module stop state can be specified for each channel.	
Asynchronous mode	Data length	7, 8, or 9 bits	7 or 8 bits
	Transmission stop bits	1 or 2 bits	
	Parity	Even parity, odd parity, or no parity	
	Receive error detection	Parity, overrun, and framing errors	
	Hardware flow control	The CTSn# and RTSn# pins can be used to control transmission and reception.	Not available.
	Start bit detection	Selectable between low level and falling edge.	Low level detection only
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.	
	Clock source	• An internal or external clock can be selected. • Transfer rate clock input from the TMR can be used (SCI5, SCI6, and SCI12).	• An internal or external clock can be selected. (excluding Smart Card interface)

Item		RX (RX231)	H8S (H8S/2378)
Asynchronous mode	Double-speed mode	Baud rate generator double-speed mode is selectable.	Not available.
	Multi-processor communication function	Serial communication among multiple processors	
	Noise cancellation	The signal paths from input on the RXDn pins incorporate on-chip digital noise filters.	Not available.
Clock synchronous mode	Data length	8 bits	
	Receive error detection	Overrun error	
	Hardware flow control	The CTSn# and RTSn# pins can be used to control transmission and reception.	Not available.
Smart card interface mode	Error processing	An error signal can be transmitted automatically when a parity error is detected during reception.	
		Data can be retransmitted automatically when an error signal is received during transmission.	
	Data type	Both direct convention and inverse convention are supported.	
IrDA		Supported.*2 In coordination with SCI5, IrDA communication waveforms conforming to IrDA specification version 1.0 can be transmitted and received.	IrDA communication waveforms conforming to IrDA specification version 1.0 can be generated (channel 0 only).
Simple I^2C mode	Communication format	I^2C bus format	Not available.
	Operating mode	Master (single-master operation only)	
	Transfer rate	Fast mode is supported.	
	Noise canceler	• The signal paths from input on the SSCLn and SSDAn pins incorporate on-chip digital noise filters. • The noise cancellation bandwidth is adjustable.	
Simple SPI mode	Data length	8 bits	Not available.
	Error detection	Overrun error	
	SS input pin function	Applying a high-level signal to the SSn# pin causes the output pins to enter the high-impedance state.	
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.	
Bit rate modulation function		On-chip baud rate generator output correction can reduce errors.	Not available.
Event link function (SCI5 only)		• Error (receive error, error signal detection) • Event output	Not available.
		Receive data full event output	
		Transmit data empty event output	
		Transmit end event output	

Item		RX (RX231)	H8S (H8S/2378)
Extended serial mode (SCI12 only)	Start frame transmission	• Output of the break field low width and generation of an interrupt on detection • Detection of bus collisions and the generation of interrupts on detection	Not available.
	Start frame reception	• Detection of the break field low width and generation of an interrupt on detection • Comparison of data in control fields 0 and 1 and generation of an interrupt when the two match • Two kinds of data for comparison (primary and secondary) can be set in control field 1. • A priority interrupt bit can be set in control field 1. • Support for handling of start frames that do not include a break field • Support for handling of start frames that do not include control field 0 • Function for measuring bit rates	
	I/O control functions	• Selectable polarity for TXDX12 and RXDX12 signals • Ability to enable digital filter function for RXDX12 • Half-duplex operation employing RXDX12 and TXDX12 signals multiplexed on the same pin • Selectable timing for the sampling of data received through RXDX12 • Signals received on RXDX12 can be passed through to SCIg when the extended serial mode control section is off.	
	Timer function	Usable as a reloading timer	

Note 1. Only MSB-first can be used in simple I^2C mode.
Note 2. On the RX231 Group the IrDA interface is a module separate from the SCI.

Table 1.2 Functions of SCI Channels on RX231 Group

	SCIg			SCIh
Item	SCI0, SCI1 SCI8, SCI9	SCI5	SCI6	SCI12
Asynchronous mode	○	○	○	○
Clock synchronous mode	○	○	○	○
Smart card interface mode	○	○	○	○
Simple I^2C mode	○	○	○	○
Simple SPI mode	○	○	○	○
Extended serial mode	—	—	—	○
TMR clock input	—	○	○	○
Event link function	—	○	—	—

1.2 Points of Difference between Registers

Table 1.3 lists the SCI registers of the RX231 Group and H8S/2378 Group.

Table 1.3 SCI registers of RX231 Group and H8S/2378 Group

RX (RX231)	H8S (H8S/2378)
Receive shift register (RSR)	Receive shift register (RSR)
Receive data register (RDR)	Receive data register (RDR)
Receive data registers H, L, HL (RDRH, RDRL, RDRHL)	Not available.
Transmit data register (TDR)	Transmit data register (TDR)
Transmit data registers H, L, and HL (TDRH, TDRL, and TDRHL)	Not available.
Transmit shift register (TSR)	Transmit shift register (TSR)
Serial mode register (SMR)	Serial mode register (SMR)
Serial control register (SCR)	Serial control register (SCR)
Serial status register (SSR)	Serial status register (SSR)
Smart card mode register (SCMR)	Smart card mode register (SCMR)
Bit rate register (BRR)	Bit rate register (BRR)
Supported.*[1] In coordination with SCI5, IrDA communication waveforms conforming to IrDA specification version 1.0 can be transmitted and received.	IrDA control register (IrCR)
Modulation duty register (MDDR)	Not available.
Serial extended mode register (SEMR)	Serial extended mode register (SEMR)
Noise filter setting register (SNFR)	Not available.
I^2C mode register 1 (SIMR1)	
I^2C mode register 2 (SIMR2)	
I^2C mode register 3 (SIMR3)	
I^2C status register (SISR)	
SPI mode register (SPMR)	
Extended serial module enable register (ESMER)	
Control register 0 (CR0)	
Control register 1 (CR1)	
Control register 2 (CR2)	
Control register 3 (CR3)	
Port control register (PCR)	
Interrupt control register (ICR)	
Status register (STR)	
Status clear register (STCR)	
Control field 0 data register (CF0DR)	
Control field 0 compare enable register (CF0CR)	
Control field 0 receive data register (CF0RR)	
Primary control field 1 data register (PCF1DR)	
Secondary control field 1 data register (SCF1DR)	
Control field 1 compare enable register (CF1CR)	
Control field 1 receive data register (CF1RR)	
Timer control register (TCR)	
Timer mode register (TMR)	
Timer prescaler register (TPRE)	
Timer count register (TCNT)	

Note 1. On the RX231 Group the IrDA interface is a module separate from the SCI.

Table 1.4 lists the points of difference between the SCI registers of the RX231 Group and H8S/2378 Group. Points of difference between registers with equivalent functions, among the registers listed in Table 1.3, are listed. Registers and bit functions not listed in Table 1.4 are identical on the RX231 Group and H8S/2378 Group.

Table 1.4 Points of Difference between SCI Registers of RX231 Group and H8S/2378 Group

Register Name		Bit Name		Description	
RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)
Receive data register H, L, HL (RDRH, RDRL, RDRHL)	Not available.			The RDRH and RDRL registers are 8-bit registers that store receive data. They are used when asynchronous mode and 9-bit data length are selected. They can also be accessed in 16-bit units as the RDRHL register.	—
Transmit data register H, L, HL (TDRH, TDRL, TDRHL)	Not available.			The TDRH and TDRL registers are 8-bit registers that store transmit data. They are used when asynchronous mode and 9-bit data length are selected. They can also be accessed in 16-bit units as the TDRHL register.	—
Serial mode register (SMR) (when SCMR.SMIF = 0)		CM	C/$\overline{A}$	Communication mode 0: Asynchronous mode 1: Clock synchronous mode or simple SPI mode	Communication mode 0: Asynchronous mode 1: Clock synchronous mode
		CHR		Character length*1 (Valid in asynchronous mode only) Selects in combination with the SCMR.CHR1 bit. CHR1 CHR 0 0 Transmit/receive in 9-bit data length 0 1 Transmit/receive in 9-bit data length 1 0 Transmit/receive in 8-bit data length 1 1 Transmit/receive in 7-bit data length*2	Character length*1 (Valid in asynchronous mode only) 0: Transmit/receive in 8-bit data length 1: Transmit/receive in 7-bit data length*2
		PM	O/$\overline{E}$	Parity mode (Only bit names differ.)	

Register Name		Bit Name		Description	
RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)
Serial mode register (SMR) (when SCMR.SMIF = 1)		PM	O/$\overline{E}$	Parity mode (Only bit names differ.)	
		BCP[1:0]	BCP1 BCP0	Base clock pulse Selects in combination with the SCMR.BCP2 bit. BCP2 BCP[1:0] 0 0 0 93 clock cycles (S = 93) 0 0 1 128 clock cycles (S = 128) 0 1 0 186 clock cycles (S = 186) 0 1 1 512 clock cycles (S = 512) 1 0 0 32 clock cycles (S = 32) 1 0 1 64 clock cycles (S = 64) 1 1 0 372 clock cycles (S = 372) 1 1 1 256 clock cycles (S = 256)	Base clock pulse 0 0 32 clock cycles (S = 32) 0 1 64 clock cycles (S = 64) 1 0 372 clock cycles (S = 372) 1 1 256 clock cycles (S = 256)
Serial control register (SCR) (when SCMR.SMIF = 0)		TEIE		TEI interrupt enable 0: TEI interrupt requests are disabled. 1: TEI interrupt requests are enabled. In simple I^2C mode, the TEI is allocated to the interrupt when issuance of a start, restart, or stop condition (STI) completes. In this case, the TEIE bit can be used to enable or disable STI interrupt requests.	TEI interrupt enable 0: TEI interrupt requests are disabled. 1: TEI interrupt requests are enabled.

Register Name		Bit Name		Description	
RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)
Serial control register (SCR) (when SCMR.SMIF = 0)		CKE[1:0]	CKE1 CKE0	Clock enable [Asynchronous mode] b1 b0 0 0: On-chip baud rate generator The SCKn pin can be used as an I/O port by means of I/O port settings. 0 1: On-chip baud rate generator A clock with the same frequency as the bit rate is output on the SCKn pin. 1 X: External clock or TMR clock When using an external clock, a clock with a frequency 16 times the bit rate should be input on the SCKn pin. When the SEMR.ABCS bit is set to 1, a clock with a frequency eight times the bit rate should be input. When using the TMR clock, the SCKn pin can be used as an I/O port by means of I/O port settings. (Note: The TMR clock can be used with SCI5, SCI6, and SCI12. It cannot be used with SCI0, SCI1, SCI8, or SCI9.) [Clock synchronous mode] b1 b0 0 X: Internal clock The SCKn pin functions as the clock output pin. 1 X: External clock The SCKn pin functions as the clock input pin. X: Don't care	Clock enable [Asynchronous mode] 0 0: On-chip baud rate generator The SCKn pin can be used as an I/O port. 0 1: On-chip baud rate generator A clock with the same frequency as the bit rate is output on the SCKn pin. 1 X: External clock A clock with a frequency 16 times the bit rate should be input on the SCKn pin. [Clock synchronous mode] 0 X: Internal clock The SCK pin functions as the clock output pin. 1 X: External clock The SCK pin functions as the clock input pin. X: Don't care

Register Name		Bit Name		Description	
RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)
Serial control register (SCR) (when SCMR.SMIF = 1)		CKE[1:0]	CKE1 CKE0	Clock enable [When SMR.GM = 0] b1 b0 0 0: Output disabled (The SCKn pin can be used as an I/O port by means of I/O port settings.) 0 1: Clock output 1 X: Reserved [When SMR.GM = 1] b1 b0 0 0: Output fixed low X 1: Clock output 1 0: Output fixed high X: Don't care	Clock enable [When SMR.GM = 0] 0 0: Output disabled (The SCKn pin can be used as an I/O port) 0 1: Clock output 1 X: Reserved [When SMR.GM = 1] 0 0: Output fixed low 0 1: Clock output 1 0: Output fixed high 1 1: Clock output X: Don't care
Serial status register (SSR) (when SCMR.SMIF = 0)		TDRE		Transmit data empty [Setting condition] • When data has been transferred from the TDR register to the TSR register [Clearing condition] • When transmit data has been written to the TDR register	Transmit data register empty [Setting conditions] • When the TE bit in SCR is cleared to 0 • When data has been transferred from TDR to TSR, and data writing to TDR has been enabled [Clearing conditions] • When 0 is written to TDRE after reading TDRE as 1 • When the DMAC or DTC has transmitted data to TDR in response to a TXI interrupt request
		RDRF		Receive data full [Setting condition] • When data has been received successfully, and transferred from the RSR register to the RDR register [Clearing condition] • When data has been read from the RDR register	Receive data register full [Setting condition] • When data has been received successfully, and transferred from the RSR to the RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF as 1 • When the DMAC or DTC has transferred data from RDR in response to an RXI interrupt

Register Name		Bit Name		Description	
RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)
Serial status register (SSR) (when SCMR.SMIF = 0)		TEND		Transmit end [Setting conditions] • When the SCR.TE bit is cleared to 0 (serial transmission disabled) • When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted [Clearing condition] • When transmit data has been written to the TDR register while the SCR.TE bit was set to 1	Transmit end [Setting conditions] • When the TE bit in SCR is cleared to 0 • When TDRE = 1 during transmission of the tail-end bit of a character being transmitted [Clearing conditions] • When 0 is written to the TDRE flag after reading TDRE as 1 • When the DMAC or DTC has written transmit data to TDR in response to a TXI interrupt request
Serial status register (SSR) (when SCMR.SMIF = 1)		TDRE		Transmit data empty [Setting condition] • When data has been transferred from the TDR register to the TSR register [Clearing condition] • When transmit data has been written to the TDR register	Transmit data register empty [Setting conditions] • When the TE bit in SCR is cleared to 0 • When data has been transferred from TDR to TSR, and data writing to TDR has been enabled [Clearing conditions] • When 0 is written to TDRE after reading TDRE as 1 • When the DMAC or DTC has transmitted data to TDR in response to a TXI interrupt request
		RDRF		Receive data full [Setting condition] • When data has been received successfully, and transferred from the RSR register to the RDR register [Clearing condition] • When data has been read from the RDR register	Receive data register full [Setting condition] • When data has been received successfully, and transferred from the RSR to the RDR [Clearing conditions] • When 0 is written to RDRF after reading RDRF as 1 • When the DMAC or DTC has transferred data from RDR in response to an RXI interrupt

Register Name		Bit Name		Description	
RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)
Serial status register (SSR) (when SCMR.SMIF = 1)		TEND		Transmit end [Setting conditions] • When the SCR.TE bit = 0 • When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag = 0, and the TDR register has not been updated [Clearing condition] • When transmit data has been written to the TDR register while the SCR.TE bit was set to 1	Transmit end [Setting conditions] • When TE = 0 and ERS = 0 in SCR • When ERS = 0 and TDRE = 1 after a specified period has elapsed following transmission of 1 byte of data [Clearing conditions] • When 0 is written to the TDRE flag after reading TDRE as 1 • When the DMAC or DTC has written transmit data to TDR in response to a TXI interrupt request
Smart card mode register (SCMR)		BCP2	Not available.	Base clock pulse 2 Selects in combination with the SMR.BCP[1:0] bits. BCP2 BCP[1:0] 0 0 0 93 clock cycles (S = 93) 0 0 1 128 clock cycles (S = 128) 0 1 0 186 clock cycles (S = 186) 0 1 1 512 clock cycles (S = 512) 1 0 0 32 clock cycles (S = 32) 1 0 1 64 clock cycles (S = 64) 1 1 0 372 clock cycles (S = 372) 1 1 1 256 clock cycles (S = 256)	—

Register Name		Bit Name		Description	
RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)
Smart card mode register (SCMR)		CHR1	Not available.	Character length 2*1 (Valid in asynchronous mode only) Selects in combination with the SMR.CHR bit. CHR1 CHR 0 0 Transmit/receive in 9-bit data length 0 1 Transmit/receive in 9-bit data length 1 0 Transmit/receive in 8-bit data length 1 1 Transmit/receive in 7-bit data length*2	—
		SDIR		Transmitted/received data transfer direction This bit can be used in the following modes: • Smart card interface mode • Asynchronous mode (multi-processor mode) • Clock synchronous mode • Simple SPI mode • This bit is set to 1 for operation in simple I^2C mode 0: LSB-first transmission/ reception 1: MSB-first transmission/ reception	Smart card data transfer direction 0: LSB-first transmission/ reception 1: MSB-first transmission/ reception The bit setting is valid only when the transmission/ reception data format is 8 bits. For 7-bit data, the setting is fixed at LSB-first.
		SMIF		Smart card interface mode select 0: Other than Smart card interface mode (asynchronous mode, clock synchronous mode, simple SPI mode, or simple I^2C mode) 1: Smart card interface mode	Smart Card interface mode select 0: Normal asynchronous or clock synchronous mode 1: Smart Card interface mode

Register Name		Bit Name		Description	
RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)
Serial extended mode register (SEMR)*3		RXDESEL	Not available.	Asynchronous Start bit edge detection select (Valid in asynchronous mode only) 0: The low level on the RXDn pin is detected as the start bit. 1: The falling edge on the RXDn pin is detected as the start bit.	—
		BGDM	Not available.	Baud rate generator double-speed mode select (Only valid when using the on-chip baud rate generator in asynchronous mode) 0: The baud rate generator outputs the clock with normal frequency. 1: The baud rate generator outputs the clock with doubled frequency.	—
		NFEN	Not available.	Digital noise filter function enable [Asynchronous mode] 0: Noise cancellation function for the RXDn input signal is disabled. 1: Noise cancellation function for the RXDn input signal is enabled. [Simple I^2C mode] 0: Noise cancellation function for the SSCLn and SSDAn input signals is disabled. 1: Noise cancellation function for the SSCLn and SSDAn input signals is enabled. The NFEN bit should be cleared to 0 in any mode other than above.	—
		BRME	Not available.	Bit rate modulation enable 0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	—

Register Name		Bit Name		Description	
RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)
Serial extended mode register (SEMR)*3		ACS0	ACS2 ACS1 ACS0	Asynchronous mode clock source select (Valid in asynchronous mode only) 0: External clock 1: Logical AND of two compare matches output from TMR (valid for SCI5, SCI6, and SCI12 only) The available compare match output varies per SCI channel.	Asynchronous clock source selection (Valid when CKE1 = 1 in asynchronous mode) Selects the clock source for the average transfer rate. The basic clock is set automatically by selecting the average transfer rate, regardless of the value of the ABCS bit. 000: External clock input 001: Selects 115.152 kbps, which is the average transfer rate exclusively for ϕ = 10.667 MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.) 010: Selects 460.606 kbps, which is the average transfer rate exclusively for ϕ = 10.667 MHz. (Operates on a basic clock with a frequency of 8 times the transfer rate.) 011: Selects 720 kbps, which is the average transfer rate exclusively for ϕ = 32 MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.) 100: Setting prohibited 101: Selects 115.196 kbps, which is the average transfer rate exclusively for ϕ = 16 MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.)

Register Name		Bit Name		Description	
RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)	RX (RX231)	H8S (H8S/2378)
Serial extended mode register (SEMR)*3		ACS0	ACS2 ACS1 ACS0		110: Selects 460.784 kbps, which is the average transfer rate exclusively for ϕ = 16 MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.) 111: Selects 720 kbps, which is the average transfer rate exclusively for ϕ = 16 MHz. (Operates on a basic clock with a frequency of 8 times the transfer rate.) The average transfer rate does not support frequencies other than 10.667 MHz, 16 MHz, or 32 MHz.

Note 1. Valid in asynchronous mode only. The data length is fixed at 8 bits in other than asynchronous mode.

Note 2. The format is fixed at LSB-first when the data length is 7 bits, and the MSB (bit 7) is not transmitted.

Note 3. Only available on channel 2 on the H8S/2378 Group.

2. Peripheral Functions Used

Table 2.1 lists the peripheral functions and modes used in the serial communications interface operation examples.

Table 2.1 Peripheral Functions and Modes Used in Serial Communications Interface Operation Examples

		RX (RX231)		H8S (H8S/2378)		
No.	Operation Example	Peripheral Function	Mode	Peripheral Function	Mode	Reference
1	Clock-synchronous serial communication (master transmit/receive operation)	SCI	Clock synchronous mode	SCI	Clock synchronous mode	3.1
2	Clock-synchronous serial communication (master transmit operation)					3.2
3	Clock-synchronous serial communication (slave receive operation)					3.3
4	Asynchronous serial communication (transmit operation)		Asynchronous mode		Asynchronous mode	4.1
5	Asynchronous serial communication (receive operation)					4.2

3. Points of Difference in Clock-Synchronous Serial Communication

This section describes points of difference in clock-synchronous serial communication on the RX231 Group and H8S/2378 Group.

Table 3.1 lists preconditions for clock-synchronous serial communication on the RX231 Group and H8S/2378 Group.

Table 3.1 Conditions for Clock-Synchronous Serial Communication

		Transmit/Receive Conditions	
Item		**RX (RX231)**	**H8S (H8S/2378)**
Peripheral function operating clock		PCLKB: 24 MHz	φ: 20 MHz
Communication speed		10 kbps	
Data format		LSB-first	
Hardware flow control		Not used	Function not available
Channels used	Master	SCI1	Channel 0
	Slave	SCI5	Channel 1
Pins used	Master	RXD1: P15	RXD0: P32
		TXD1: P26	TXD0: P30
		SCK1: P17	SCK0: P34
	Slave	RXD5: PA3	RXD1: P33
		TXD5: PA4	TXD1: P31
		SCK5: PA1	SCK1: P35

3.1 Points of Difference in Master Transmit/Receive Operation

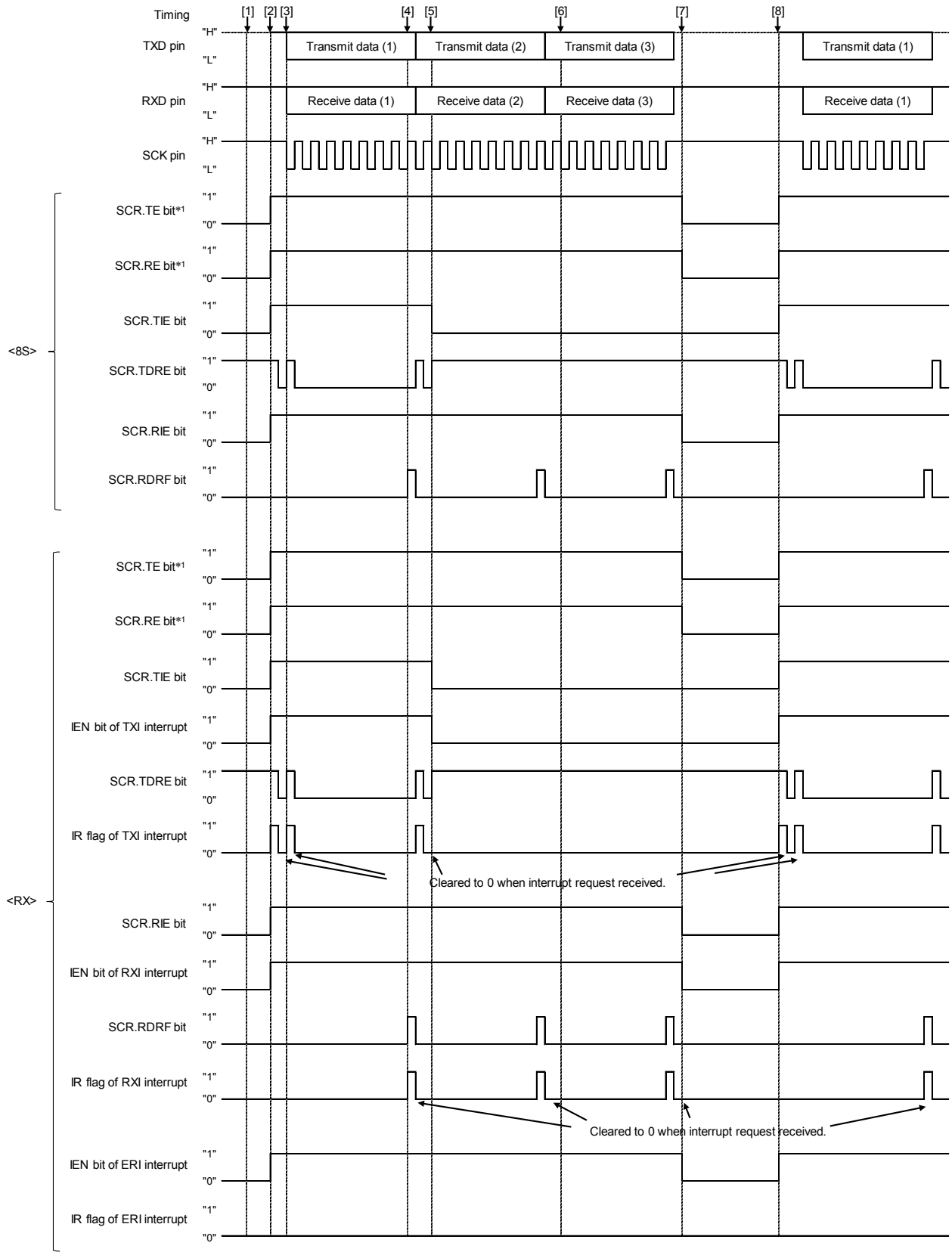
3.1.1 Points of Difference in Master Transmit/Receive Operation Timing

Points of difference in master transmit/receive operation in clock-synchronous mode are described below. The preconditions are that transmit, receive, and receive-error interrupts are used for transmit and receive processing, and that no receive errors are generated. For the points of difference when receive errors are generated, refer to 3.3.1, Points of Difference in Slave Receive Operation Timing.

Figure 3.1 shows points of difference in transmit/receive operation timing on the RX231 Group and H8S/2378 Group (for transmit/receive operation in 3-byte increments).

Table 3.2 lists points of difference in operation timing and processing details on the RX231 Group and H8S/2378 Group (for transmit/receive operation in 3-byte increments).

The numbers [1] to [8] in Figure 3.1 correspond to items [1] to [8] in Table 3.2.



Note 1. 1 or 0 should be written to bits SCR.TE and SCR.RE simultaneously.

Figure 3.1 Points of Difference in Transmit/Receive Operation Timing on RX231 Group and H8S/2378 Group (Transmit/Receive in 3-Byte Increments)

Table 3.2 Points of Difference in Operation Timing and Processing Details on RX231 Group and H8S/2378 Group (Transmit/Receive in 3-Byte Increments)

Timing	RX (RX231)	H8S (H8S/2378)
[1] Before start of transmission	When the pin function is set to TXDn, the TXD pin is in the high-impedance state until the SCR.TE bit is set to 1 (transmission enabled).*1	The TXD pin is set as a general I/O port until the SCR.TE bit is set to 1 (transmission enabled).
[2] Start of transmission/reception	The following bit settings are made to enable interrupts: SCR.TIE bit = 1 SCR.RIE bit = 1 TXI interrupt IEN bit = 1 RXI interrupt IEN bit = 1 ERI interrupt IEN bit = 1 Also, the following bit settings are made to enable transmission/reception: SCR.TE bit = 1 SCR.RE bit = 1 When the SCR.TE bit is set to 1, the IR flag of the transmit interrupt (TXI interrupt) is also set to 1. The first byte of transmit data is written in response to the transmit interrupt.	The following bit settings are made to enable interrupts: SCR.TIE bit = 1 SCR.RIE bit = 1 Also, the following bit settings are made to enable transmission/reception: SCR.TE bit = 1 SCR.RE bit = 1 When the SCR.TE bit is set to 1, a transmit interrupt (TXI interrupt) is generated. The first byte of transmit data is written in response to the transmit interrupt.
[3] Transfer of first byte of transmit data to transmit shift register	The transmit interrupt IR flag is set to 1, and a transmit interrupt is generated. The transmit interrupt handler then reads the second byte of data.	The SSR.TDRE bit is set to 1, and a transmit interrupt is generated. The receive interrupt handler then reads the second byte of data and clears the SSR.TDRE bit to 0.
[4] End of reception	After one byte of data is received, the received data is stored in the receive buffer and the receive interrupt (RXI interrupt) IR flag is set to 1. The value stored in the receive buffer is then read by the receive interrupt handler.	After one byte of data is received, the received data is stored in the receive buffer, the SSR.RDRF flag is set to 1, and a receive interrupt (RXI interrupt) is generated. The receive interrupt handler then reads the value stored in the receive buffer and clears the SSR.RDRF flag to 0.
[5] Transmit interrupt at write of final data	The following bit settings are made to disable the transmit interrupt: SCR.TIE bit = 0 TXI interrupt IEN bit = 0	The following bit settings are made to disable the transmit interrupt: SCR.TIE bit = 0
[6] After write of final data	No more transmit interrupts are generated.	

Timing	RX (RX231)	H8S (H8S/2378)
[7] Receive-end interrupt for final data	The receive interrupt handler reads the received data, and then makes the following bit settings to disable interrupts: SCR.RIE bit = 0 RXI interrupt IEN bit = 0 ERI interrupt IEN bit = 0 Also, the following bit settings are made to disable transmit and receive operation: SCR.TE bit = 0 SCR.RE bit = 0 When transmit and receive operation are disabled while the pin function is set to TXD, the TXD pin enters the high-impedance state.	The receive interrupt handler reads the received data, and then makes the following bit settings to disable interrupts: SCR.RIE bit = 0 Also, the following bit settings are made to disable transmit and receive operation: SCR.TE bit = 0 SCR.RE bit = 0 When transmit operation is disabled, the TXD pin becomes a general I/O port.
[8] Restart of transmission/reception	Processing is the same as that described in [2] Start of transmission/reception.	

Note 1. For details, refer to 9.4, Notes Regarding Transmit-Enable Bit.

3.1.2 Points of Difference in Setting Procedure for Master Transmit/Receive Operation

Table 3.3 lists the points of difference in the initial setting procedure for transmit/receive operation. The initial setting procedure shown for the H8S/2378 Group applies to the use of interrupt control mode 2.

Table 3.4 lists points of difference in transmit interrupt processing during transmit/receive operation.

Table 3.5 lists points of difference in receive interrupt processing during transmit/receive operation.

Table 3.3 Points of Difference in Initial Setting Procedure for Transmit/Receive Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Cancel the module stop state.*[1]	SYSTEM.PRCR.WORD = 0xA502; MSTP(SCI1) = 0; SYSTEM.PRCR.WORD = 0xA500;	MSTPCR.BIT._SCI0 = 0;
2	Disable transmit/receive interrupts.	SCI1.SCR.BYTE = 0x00;	SCI0.SCR.BYTE = 0x00;
3	Make I/O port function settings.*[2]	PORT1.PMR.BIT.B5 = 0; PORT2.PMR.BIT.B6 = 0; PORT1.PMR.BIT.B7 = 0; MPC.PWPR.BIT.B0WI = 0; MPC.PWPR.BIT.PFSWE = 1; MPC.P15PFS.BYTE = 0x0A; MPC.P26PFS.BYTE = 0x0A; MPC.P17PFS.BYTE = 0x0A; MPC.PWPR.BIT.PFSWE = 0; MPC.PWPR.BIT.B0WI = 1; PORT1.PMR.BIT.B5 = 1; PORT2.PODR.BIT.B6 = 1; PORT2.PDR.BIT.B6 = 1; PORT2.PMR.BIT.B6 = 1; PORT1.PMR.BIT.B7 = 1;	— (No processing)
4	Make transmit/receive mode, etc., settings.	SCI1.SCR.BIT.CKE = 00b; SCI1.SIMR1.BYTE = 0x00; SCI1.SPMR.BYTE = 0x00; SCI1.SMR.BYTE = 0x81; SCI1.SCMR.BYTE = 0xF2; SCI1.SEMR.BYTE = 0x00;	SCI0.SCR.BIT.CKE = 00b; SCI0.SMR.BYTE = 0x81; SCI0.SCMR.BYTE = 0xF2;
5	Make bit rate settings.*[3]	SCI1.BRR = 0x95;	SCI0.BRR = 0x7C;
6	Wait for 1-bit period.	— (No processing)	1-bit period wait processing
7	Make interrupt control mode setting.*[4]	— (No processing)	INTC.INTCR.BIT.INTM = 10b;
8	Make interrupt priority level setting.*[5]	IPR(SCI1,) = 0x01;	INTC.IPRI.BIT._SCI0 = 001b;
9	Clear interrupt requests.	IR(SCI1,TXI1) = 0; IR(SCI1,RXI1) = 0; IR(SCI1,ERI1) = 0;	— (No processing)
10	Enable peripheral function interrupt requests.	SCI1.SCR.BYTE |= 0xF0; /* *[6] */	SCI0.SCR.BYTE |= 0xF0; /* *[6] */
11	Enable transmit/receive operation.		

Step		RX (RX231)	H8S (H8S/2378)
12	Enable interrupt requests.*[7]	IEN(SCI1,TXI1) = 1; IEN(SCI1,RXI1) = 1; IEN(SCI1,ERI1) = 1; /* *[8] */	— (No processing)
13	Make processor interrupt priority level setting.	— (No processing)	set_imask_exr(0);
14	Enable maskable interrupts.	setpsw_i();	— (No processing)

Note 1. For information on the module stop function, refer to section 7, Module Stop Function.
Note 2. On the RX231 Group peripheral function pin settings are made in the MPC. For details, refer to 9.1, I/O Ports.
Note 3. For details on the bit rate, refer to section 5, Calculating the Bit Rate.
Note 4. The RX231 Group has no interrupt control mode. For details, refer to section 6, Points of Difference between Interrupts
Note 5. For details of the interrupt priority level setting method, refer to section 6, Points of Difference between Interrupts
Note 6. Bits SCR.TE and SCR.RE should be set to 1 (transmission enabled, reception enabled) simultaneously.
Note 7. The methods of enabling interrupt requests differ. For details, refer to section 6, Points of Difference between Interrupts
Note 8. The specifications of the receive-error interrupt differ between microcontrollers. For details, refer to User's Manual: Hardware.

Table 3.4 Points of Difference in Transmit Interrupt Processing During Transmit/Receive Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Read TDRE flag.	— (No processing)	if(SCI0.SSR.BIT.TDRE == 1) {
2	Write transmit data.	/* Writes transmit data to SCI1.TDR register. */	/* Writes transmit data to SCI0.TDR register. */
3	Clear TDRE flag.	— (No processing)	SCI0.SSR.BIT.TDRE = 0;
4	Confirm final data write.	if(Final data write finished?) {	if(Final data write finished?) {

5	Disable transmit interrupt (after final data write only).	IEN(SCI1,TXI1) = 0
		SCI1.SCR.BIT.TIE = 0;
		while(0 != SCI1.SCR.BIT.TIE)
		{
		}
			SCI0.SCR.BIT.TIE = 0;
			while(0 != SCI1.SCR.BIT.TIE)
			{
			}
			}

6	Clear interrupt request (after final data write only).	IR(SCI1,TXI1) = 0;
		while(0 != IR(SCI1,TXI1))
		{
		}
			—
			(No processing)

Table 3.5 Points of Difference in Receive Interrupt Processing During Transmit/Receive Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Read RDRF flag.	— (No processing)	if(SCI0.SSR.BIT.RDRF == 1) {
2	Read receive data.	/* Reads receive data from SCI1.RDR register. */	/* Reads receive data from SCI0.RDR register. */
3	Clear RDRF flag.	— (No processing)	SCI0.SSR.BIT.RDRF = 0;
4	Confirm reception of final data.	if(Final data received?) {	if(Final data received?) {

5	Disable transmit/receive operation and receive interrupt (after reception of final data only).	IEN(SCI1,RXI1) = 0;
		SCI1.SCR.BYTE &= 0x0B;*1
		while(0x00 != (SCI1.SCR.BYTE & 0xF4))
		{
		}
			SCI0.SCR.BYTE &= 0x0B;*1
			while(0x00 != (SCI0.SCR.BYTE & 0xF4))
			{
			}
			}

6	Clear interrupt request (after final data write only).	IR(SCI1,RXI1) = 0;
		while(0 != IR(SCI1,RXI1))
		{
		}
			—
			(No processing)

Note 1. Bits SCR.TE and SCR.RE should be set to 0 (transmission disabled, reception disabled) simultaneously.

3.2 Points of Difference in Master Transmit Operation

The points of difference in clock-synchronous master transmit operation are described below. The preconditions are that transmit and transmit-end interrupts are used for transmit processing

3.2.1 Points of Difference in Master Transmit Operation Timing

Figure 3.2 shows points of difference in transmit operation timing on the RX231 Group and H8S/2378 Group (for transmit operation in 3-byte increments).

Table 3.6 lists points of difference in operation timing and processing details on the RX231 Group and H8S/2378 Group (for transmit operation in 3-byte increments).

The numbers [1] to [8] in Figure 3.2 correspond to items [1] to [8] in Table 3.6.

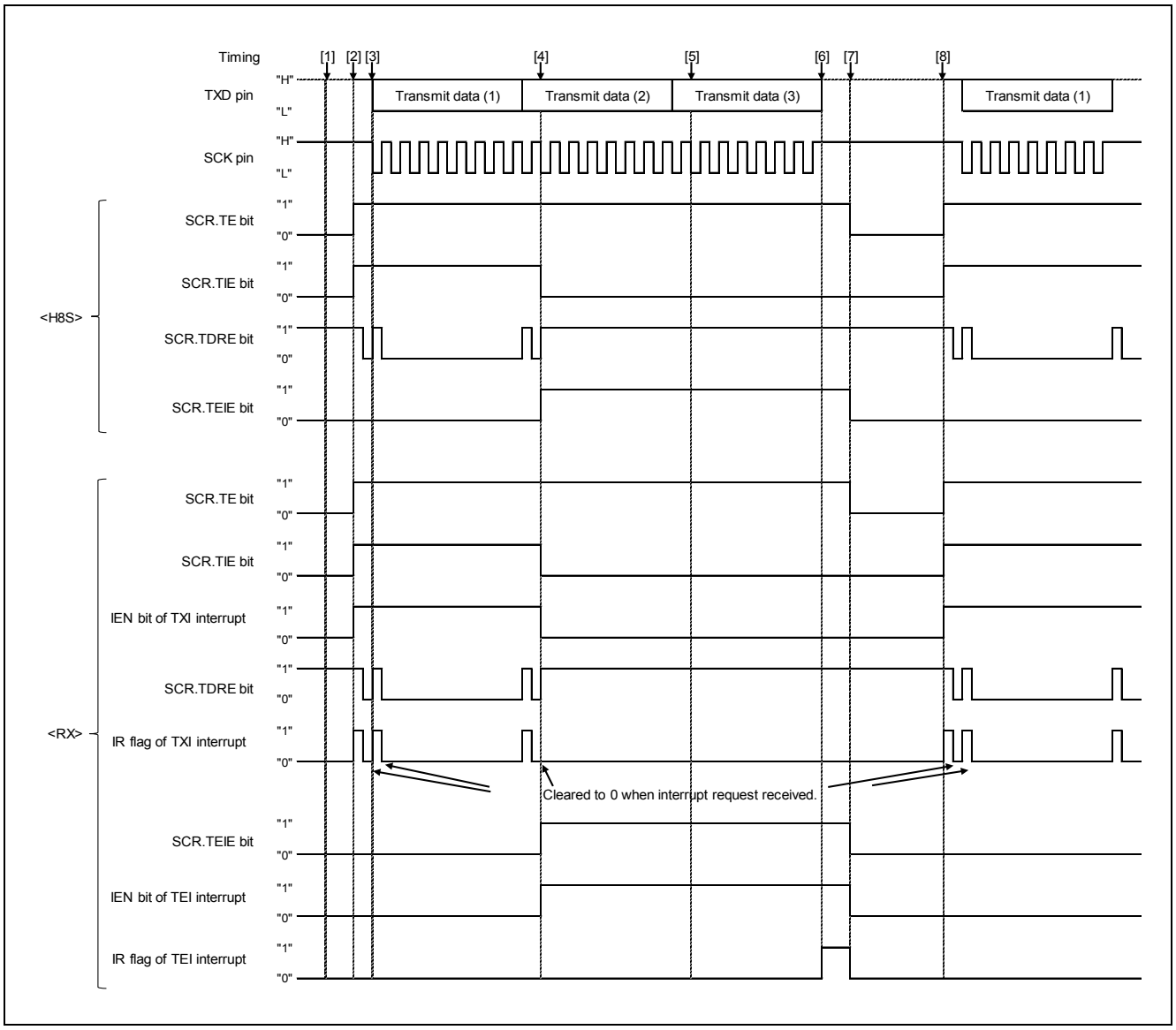


Figure 3.2 Points of Difference in Transmit Operation Timing on RX231 Group and H8S/2378 Group (Transmission in 3-Byte Increments)

Table 3.6 Points of Difference in Operation Timing and Processing Details on RX231 Group and H8S/2378 Group (Transmission in 3-Byte Increments)

Timing	RX (RX231)	H8S (H8S/2378)
[1] Before start of transmission	When the pin function is set to TXDn, the TXD pin is in the high-impedance state until the SCR.TE bit is set to 1 (transmission enabled).*1	The TXD pin is set as a general I/O port until the SCR.TE bit is set to 1 (transmission enabled).
[2] Start of transmission	The following bit settings are made to enable interrupts: SCR.TIE bit = 1 TXI interrupt IEN bit = 1 Also, the following bit settings are made to enable transmission: SCR.TE bit = 1 When the SCR.TE bit is set to 1, the IR flag of the transmit interrupt (TXI interrupt) is also set to 1. The first byte of transmit data is written in response to the transmit interrupt.	The following bit settings are made to enable interrupts: SCR.TIE bit = 1 Also, the following bit settings are made to enable transmission: SCR.TE bit = 1 When the SCR.TE bit is set to 1, a transmit interrupt (TXI interrupt) is generated. The first byte of transmit data is written in response to the transmit interrupt.
[3] Transfer of first byte of transmit data to transmit shift register	The transmit interrupt IR flag is set to 1, and a transmit interrupt is generated. The transmit interrupt handler then reads the second byte of data.	The SSR.TDRE flag is set to 1, and a transmit interrupt is generated. The receive interrupt handler then reads the second byte of data and clears the SSR.TDRE flag to 0.
[4] Transmit interrupt at write of final data	The following bit settings are made to disable the transmit interrupt: SCR.TIE bit = 0 TXI interrupt IEN bit = 0 In addition, the following bit settings are made to enable the transmit-end interrupt: SCR.TEIE bit = 1 TEI interrupt IEN bit = 1	The following bit settings are made to disable the transmit interrupt: SCR.TIE bit = 0 In addition, the following bit settings are made to enable the transmit-end interrupt: SCR.TEIE bit = 1
[5] After write of final data	No more transmit interrupts are generated.	
[6] End of transmission	A transmit-end interrupt is generated.	
[7] Transmit-end interrupt handler	The following bit settings are made to disable the transmit-end interrupt: SCR.TEIE bit = 0 Transmit-end interrupt IEN bit = 0 In addition, the following bit settings are made to disable transmission: SCR.TE bit = 0 When transmit and receive operation are disabled while the pin function is set to TXD, the TXD pin enters the high-impedance state.	The following bit settings are made to disable the transmit-end interrupt: SCR.TEIE bit = 0 In addition, the following bit settings are made to disable transmission: SCR.TE bit = 0 When transmit operation is disabled, the TXD pin becomes a general I/O port.
[8] Restart of transmission	Processing is the same as that described in [2] Start of transmission.	

Note 1. For details, refer to 9.4, Notes Regarding Transmit-Enable Bit.

3.2.2 Points of Difference in Setting Procedure for Master Transmit Operation

Table 3.7 lists the points of difference in the initial setting procedure for transmit operation. The initial setting procedure shown for the H8S/2378 Group applies to the use of interrupt control mode 2.

Table 3.8 lists points of difference in transmit interrupt processing during transmit operation.

Table 3.9 lists points of difference in transmit-end interrupt processing during transmit operation.

Table 3.7 Points of Difference in Initial Setting Procedure for Transmit Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Cancel the module stop state.*[1]	SYSTEM.PRCR.WORD = 0xA502; MSTP(SCI1) = 0; SYSTEM.PRCR.WORD = 0xA500;	MSTPCR.BIT._SCI0 = 0;
2	Disable transmit/receive interrupts.	SCI1.SCR.BYTE = 0x00;	SCI0.SCR.BYTE = 0x00;
3	Make I/O port function settings.*[2]	PORT2.PMR.BIT.B6 = 0; PORT1.PMR.BIT.B7 = 0; MPC.PWPR.BIT.B0WI = 0; MPC.PWPR.BIT.PFSWE = 1; MPC.P26PFS.BYTE = 0x0A; MPC.P17PFS.BYTE = 0x0A; MPC.PWPR.BIT.PFSWE = 0; MPC.PWPR.BIT.B0WI = 1; PORT2.PODR.BIT.B6 = 1; PORT2.PDR.BIT.B6 = 1; PORT2.PMR.BIT.B6 = 1; PORT1.PMR.BIT.B7 = 1;	— (No processing)
4	Make transmit mode, etc., settings.	SCI1.SCR.BIT.CKE = 00b; SCI1.SIMR1.BYTE = 0x00; SCI1.SPMR.BYTE = 0x00; SCI1.SMR.BYTE = 0x81; SCI1.SCMR.BYTE = 0xF2; SCI1.SEMR.BYTE = 0x00;	SCI0.SCR.BIT.CKE = 00b; SCI0.SMR.BYTE = 0x81; SCI0.SCMR.BYTE = 0xF2;
5	Make bit rate settings.*[3]	SCI1.BRR = 0x95;	SCI0.BRR = 0x7C;
6	Wait for 1-bit period.	— (No processing)	1-bit period wait processing
7	Make interrupt control mode setting.*[4]	— (No processing)	INTC.INTCR.BIT.INTM = 10b;
8	Make interrupt priority level setting.*[5]	IPR(SCI1,) = 0x01;	INTC.IPRI.BIT._SCI0 = 001b;
9	Clear interrupt requests.	IR(SCI1,TXI1) = 0; IR(SCI1,TEI1) = 0;	— (No processing)
10	Enable peripheral function interrupt requests.	SCI1.SCR.BYTE |= 0xA0;	SCI0.SCR.BYTE |= 0xA0;
11	Enable transmit operation.		
12	Enable interrupt requests.*[6]	IEN(SCI1,TXI1) = 1; IEN(SCI1,TEI1) = 1;	— (No processing)
13	Make processor interrupt priority level setting.	— (No processing)	set_imask_exr(0);
14	Enable maskable interrupts.	setpsw_i();	— (No processing)

Note 1. For information on the module stop function, refer to section 7, Module Stop Function.
Note 2. On the RX231 Group peripheral function pin settings are made in the MPC. For details, refer to 9.1, I/O Ports.
Note 3. For details on the bit rate, refer to section 5, Calculating the Bit Rate.
Note 4. The RX231 Group has no interrupt control mode. For details, refer to section 6, Points of Difference between Interrupts
Note 5. For details of the interrupt priority level setting method, refer to section 6, Points of Difference between Interrupts
Note 6. The methods of enabling interrupt requests differ. For details, refer to section 6, Points of Difference between Interrupts

Table 3.8 Points of Difference in Transmit Interrupt Processing During Transmit Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Read TDRE flag.	— (No processing)	if(SCI0.SSR.BIT.TDRE == 1) {
2	Write transmit data.	/* Writes transmit data to SCI1.TDR register. */	/* Writes transmit data to SCI0.TDR register. */
3	Clear TDRE flag.	— (No processing)	SCI0.SSR.BIT.TDRE = 0;
4	Confirm final data write.	if(Final data write finished?) {	if(Final data write finished?) {

5	Disable transmit interrupt (after final data write only).	IEN(SCI1,TXI1) = 0;
		SCI1.SCR.BIT.TIE = 0;
		while(0 != SCI1.SCR.BIT.TIE)
		{
		}
			SCI0.SCR.BIT.TIE = 0;
			while(0 != SCI0.SCR.BIT.TIE)
			{
			}

6	Clear interrupt request (after final data write only).	IR(SCI1,TXI1) = 0;
		while(0 != IR(SCI1,TXI1))
		{
		}
			—
			(No processing)

7	Enable transmit-end interrupt (after final data write only).	SCI1.SCR.BIT.TEIE = 1; IEN(SCI1,TEI1) = 1; }	SCI0.SCR.BIT.TEIE = 1; } }

Table 3.9 Points of Difference in Transmit-End Interrupt Processing During Transmit Operation

Step		RX (RX231)	H8S (H8S/2378)

1	Disable transmission and interrupts.	IEN(SCI1,TEI1) = 0;
		SCI1.SCR.BYTE &= 0x0B;
		while(0x00 != (SCI1.SCR.BYTE & 0xF4))
		{
		}
			SCI0.SCR.BYTE &= 0x0B;
			while(0x00 != (SCI0.SCR.BYTE & 0xF4))
			{
			}

2	Clear interrupt request.	IR(SCI1,TEI1) = 0;
		while(0 != IR(SCI1,TEI1))
		{
		}
			—
			(No processing)*[1]

Note 1. The transmit-end interrupt request is cleared when the SCR.TEIE is cleared to 0.

3.3 Points of Difference in Slave Receive Operation

The points of difference in clock-synchronous slave receive operation are described below.

In this example, the preconditions are that receive and receive-error interrupts are used for receive processing, that receive interrupts are delayed by other interrupts during data reception, and that overrun errors are generated.

3.3.1 Points of Difference in Slave Receive Operation Timing

Figure 3.3 shows points of difference in operation timing on the RX231 Group and H8S/2378 Group (for receive operation).

Table 3.10 lists points of difference in operation timing and processing details on the RX231 Group and H8S/2378 Group (for receive operation).

The numbers [1] to [6] in Figure 3.3 correspond to items [1] to [6] in Table 3.10.

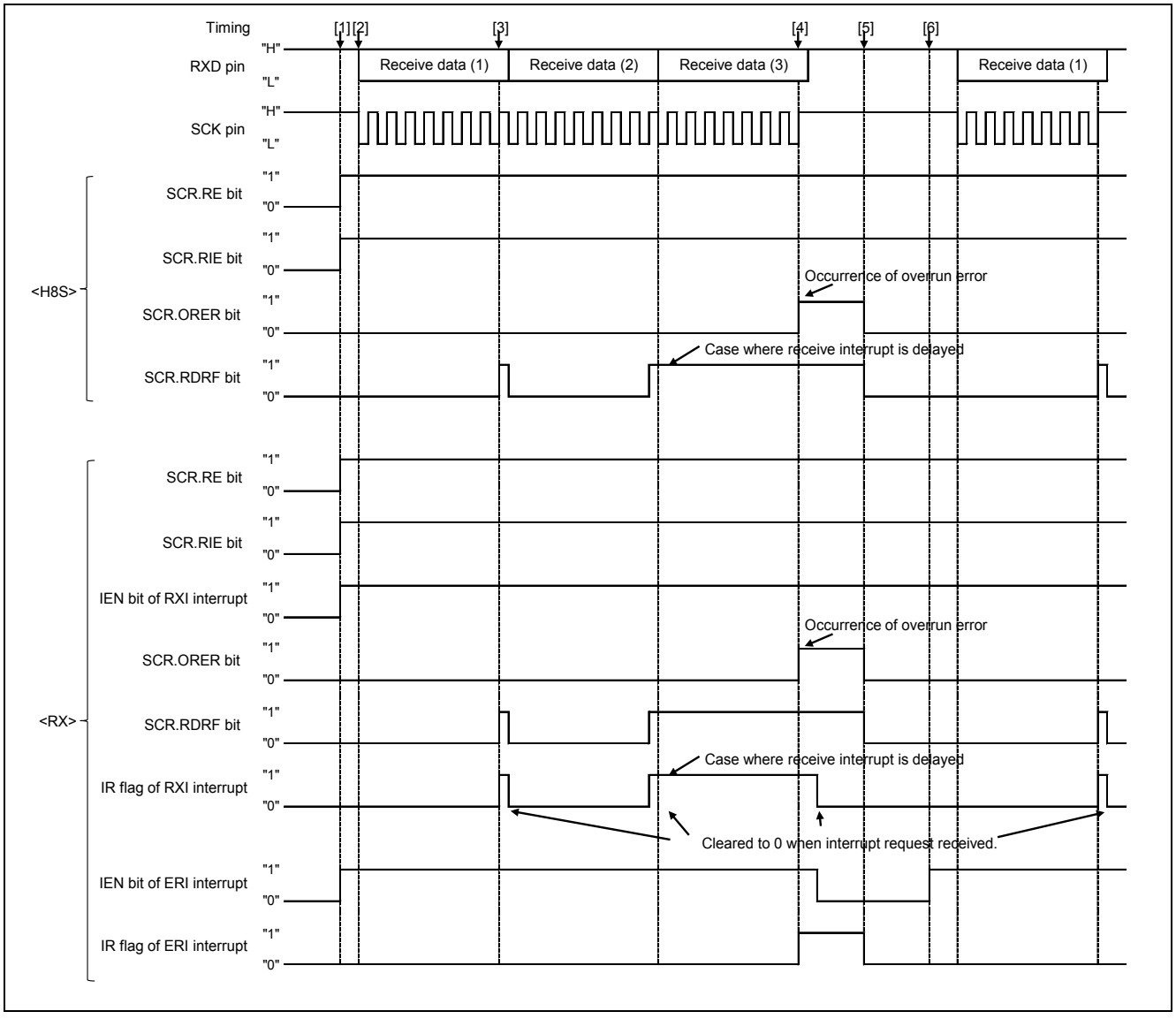


Figure 3.3 Points of Difference in Timing on RX231 Group and H8S/2378 Group (Reception)

Table 3.10 Points of Difference in Operation Timing and Processing Details on RX231 Group and H8S/2378 Group (Reception)

Timing	RX (RX231)	H8S (H8S/2378)
[1] Reception enable setting	The following bit settings are made to enable interrupts: SCR.RIE bit = 1 RXI interrupt IEN bit = 1 ERI interrupt IEN bit = 1 The following bit settings are made to enable reception: SCR.RE bit = 1	The following bit settings are made to enable interrupts: SCR.RIE bit = 1 The following bit settings are made to enable reception: SCR.RE bit = 1
[2] Start of reception	Receive operation starts when a clock is input on the SCK pin.	Receive operation starts when a clock is input on the SCK pin.
[3] End of reception	After one byte of data is received, the received data is stored in the receive buffer, the receive interrupt (RXI interrupt) IR flag is set to 1, and a receive interrupt is generated. The value stored in the receive buffer is then read by the receive interrupt handler.	After one byte of data is received, the received data is stored in the receive buffer, the SSR.RDRF flag is set to 1, and a receive interrupt (RXI interrupt) is generated. The receive interrupt handler then reads the value stored in the receive buffer and clears the SSR.RDRF flag to 0.
[4] Occurrence of receive error	When an overrun error occurs, the receive-error interrupt (ERI interrupt) IR flag is set to 1. The ERI interrupt handler performs receive error processing.	When an overrun error occurs, the SSR.ORER flag is set to 1, and an ERI interrupt is generated. The ERI interrupt handler performs receive error processing.
[5] Clearing of receive error flag [6] Reissuance of reception enable setting	After the overrun error flag in the SSR register is read, 0 is written to it to clear the error flag. When all the error flags are cleared, the ERI interrupt IR flag is cleared to 0 and a transition to the reception-enabled state occurs.	After the overrun error flag in the SSR register is read, 0 is written to it to clear the error flag. When all the error flags are cleared, a transition to the reception-enabled state occurs.

3.3.2 Points of Difference in Setting Procedure for Slave Receive Operation

Table 3.11 lists the points of difference in the initial setting procedure for receive operation. The initial setting procedure shown for the H8S/2378 Group applies to the use of interrupt control mode 2.

Table 3.12 lists points of difference in receive interrupt processing during receive operation.

Table 3.13 lists points of difference in ERI interrupt processing during receive operation.

Table 3.11 Points of Difference in Initial Setting Procedure for Receive Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Cancel the module stop state.*1	SYSTEM.PRCR.WORD = 0xA502; MSTP(SCI5) = 0; SYSTEM.PRCR.WORD = 0xA500;	MSTPCR.BIT._SCI1 = 0;
2	Disable transmit/receive interrupts.	SCI5.SCR.BYTE = 0x00;	SCI1.SCR.BYTE = 0x00;
3	Make I/O port function settings.*2	PORTA.PMR.BIT.B3 = 0; PORTA.PMR.BIT.B1 = 0; MPC.PWPR.BIT.B0WI = 0; MPC.PWPR.BIT.PFSWE = 1; MPC.PA3PFS.BYTE = 0x0A; MPC.PA1PFS.BYTE = 0x0A; MPC.PWPR.BIT.PFSWE = 0; MPC.PWPR.BIT.B0WI = 1; PORTA.PMR.BIT.B3 = 1; PORTA.PMR.BIT.B1 = 1;	— (No processing)
4	Make receive mode, etc., settings.	SCI5.SCR.BIT.CKE = 10b; SCI5.SIMR1.BYTE = 0x00; SCI5.SPMR.BYTE = 0x00; SCI5.SMR.BIT.CM= 1; SCI5.SCMR.BYTE = 0xF2; SCI5.SEMR.BYTE = 0x00;	SCI1.SCR.BIT.CKE = 10b; SCI1.SMR.BIT.CA = 1; SCI1.SCMR.BYTE = 0xF2;
5	Wait for 1-bit period.	— (No processing)	1-bit period wait processing
6	Make interrupt control mode setting.*3	— (No processing)	INTC.INTCR.BIT.INTM = 10b;
7	Make interrupt priority level setting.*4	IPR(SCI5,) = 0x01;	INTC.IPRI.BIT._SCI1 = 001b;
8	Clear interrupt requests.	IR(SCI5,RXI5) = 0;	— (No processing)
9	Enable peripheral function interrupt requests.	SCI5.SCR.BYTE |= 0x50;	SCI1.SCR.BYTE |= 0x50;
10	Enable transmit/receive operation.		
11	Enable interrupt requests.*5	IEN(SCI5,RXI5) = 1; IEN(SCI5,ERI5) = 1; /* *6 */	— (No processing)
12	Make processor interrupt priority level setting.	— (No processing)	set_imask_exr(0);
13	Enable maskable interrupts.	setpsw_i();	— (No processing)

Note 1. For information on the module stop function, refer to section 7, Module Stop Function.
Note 2. On the RX231 Group peripheral function pin settings are made in the MPC. For details, refer to 9.1, I/O Ports.
Note 3. The RX231 Group has no interrupt control mode. For details, refer to section 6, Points of Difference between Interrupts

Note 4. For details of the interrupt priority level setting method, refer to section 6, Points of Difference between Interrupts

Note 5. The methods of enabling interrupt requests differ. For details, refer to section 6, Points of Difference between Interrupts

Note 6. The specifications of the receive-error interrupt differ between microcontrollers. For details, refer to User's Manual: Hardware.

Table 3.12 Points of Difference in Receive Interrupt Processing During Receive Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Read RDRF flag.	— (No processing)	if(SCI1.SSR.BIT.RDRF == 1) {
2	Read receive data.	/* Reads receive data from SCI5.RDR register. */	/* Reads receive data from SCI1.RDR register. */
3	Clear RDRF flag.	— (No processing)	SCI1.SSR.BIT.RDRF = 0;
4	Confirm reception of final data.	if(Final data received?) {	if(Final data received?) {

5	Disable reception and interrupts (after reception of final data only).	IEN(SCI5,RXI5) = 0;
		IEN(SCI5,ERI5) = 0;
		SCI5.SCR.BYTE &= 0x0B;
		while(0x00 != (SCI5.SCR.BYTE & 0xF4))
		{
		}
			SCI1.SCR.BYTE &= 0x0B;
			while(0x00 != (SCI1.SCR.BYTE & 0xF4))
			{
			}
			}

6	Clear interrupt request.	IR(SCI5,RXI5) = 0;
		while(0 != IR(SCI5,RXI5))
		{
		}
		IR(SCI5,ERI5) = 0;
		while(0 != IR(SCI5,ERI5))
		{
		}
		}
			—
			(No processing)

Table 3.13 Points of Difference in ERI Interrupt Processing During Receive Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Read ORER flag.	dummy = SCI5.SSR.BIT.ORER; /* *[1] */	dummy = SCI1.SSR.BIT.ORER; /* *[1] */
2	Processing when receive error occurs	if(dummy == 1) {	if(dummy == 1) {
3	Perform dummy read of receive buffer.	dummy_data = SCI5.RDR;	— (No processing)
4	Error handler	/* Indicate processing when overrun error occurs. */	/* Indicate processing when overrun error occurs. */

5	Disable reception and interrupts.	IEN(SCI5,RXI5) = 0;
		IEN(SCI5,ERI5) = 0;
		SCI5.SCR.BYTE &= 0x0B;
		while(0x00 != (SCI5.SCR.BYTE & 0xF4))
		{
		}
			SCI1.SCR.BYTE &= 0x0B;
			while(0x00 != (SCI1.SCR.BYTE & 0xF4))
			{
			}

6	Clear ORER flag.	SCI5.SSR.BIT.ORER = 0;
		while(0 != SCI5.SSR.BIT.ORER)
		{
		}
			SCI1.SSR.BIT.ORER = 0;
			while(0 != SCI1.SSR.BIT.ORER)
			{
			}
			}

7	Clear interrupt request.	IR(SCI5,RXI5) = 0;
		while(0 != IR(SCI5,RXI5))
		{
		}
		IR(SCI5,ERI5) = 0;
		while(0 != IR(SCI5,ERI5))
		{
		}
		}
			—
			(No processing)

Note 1. Before clearing the overrun error (ORER) flag, first read it while its value is 1.

4. Points of Difference in Asynchronous Serial Communication

This section describes points of difference in asynchronous serial communication on the RX231 Group and H8S/2378 Group.

Table 4.1 lists preconditions for asynchronous serial communication on the RX231 Group and H8S/2378 Group.

Table 4.1 Conditions for Asynchronous Serial Communication

	Transmit/Receive Conditions	
Item	**RX (RX231)**	**H8S (H8S/2378)**
Peripheral function operating clock	PCLKB: 24 MHz	φ: 20 MHz
Communication speed	9,600 bps	
Data length	8 bits	
Parity	None	
Stop bit	1 bit	
Data format	LSB-first	
Hardware flow control	Not used	Function not available
Channels used	SCI5	Channel 0
Pins used	RXD5: PA3 TXD5: PA4	RXD0: P32 TXD0: P30

4.1 Points of Difference in Transmit Operation

The points of difference in transmit processing during asynchronous serial communication are described below.

The preconditions are that transmit and transmit-end interrupts are used.

4.1.1 Points of Difference in Transmit Operation Timing

Figure 4.1 shows points of difference in transmit operation timing on the RX231 Group and H8S/2378 Group (for transmit operation in 3-byte increments).

Table 4.2 lists points of difference in operation timing and processing details on the RX231 Group and H8S/2378 Group (for transmit operation in 3-byte increments).

The numbers [1] to [8] in Figure 4.1 correspond to items [1] to [8] in Table 4.2.

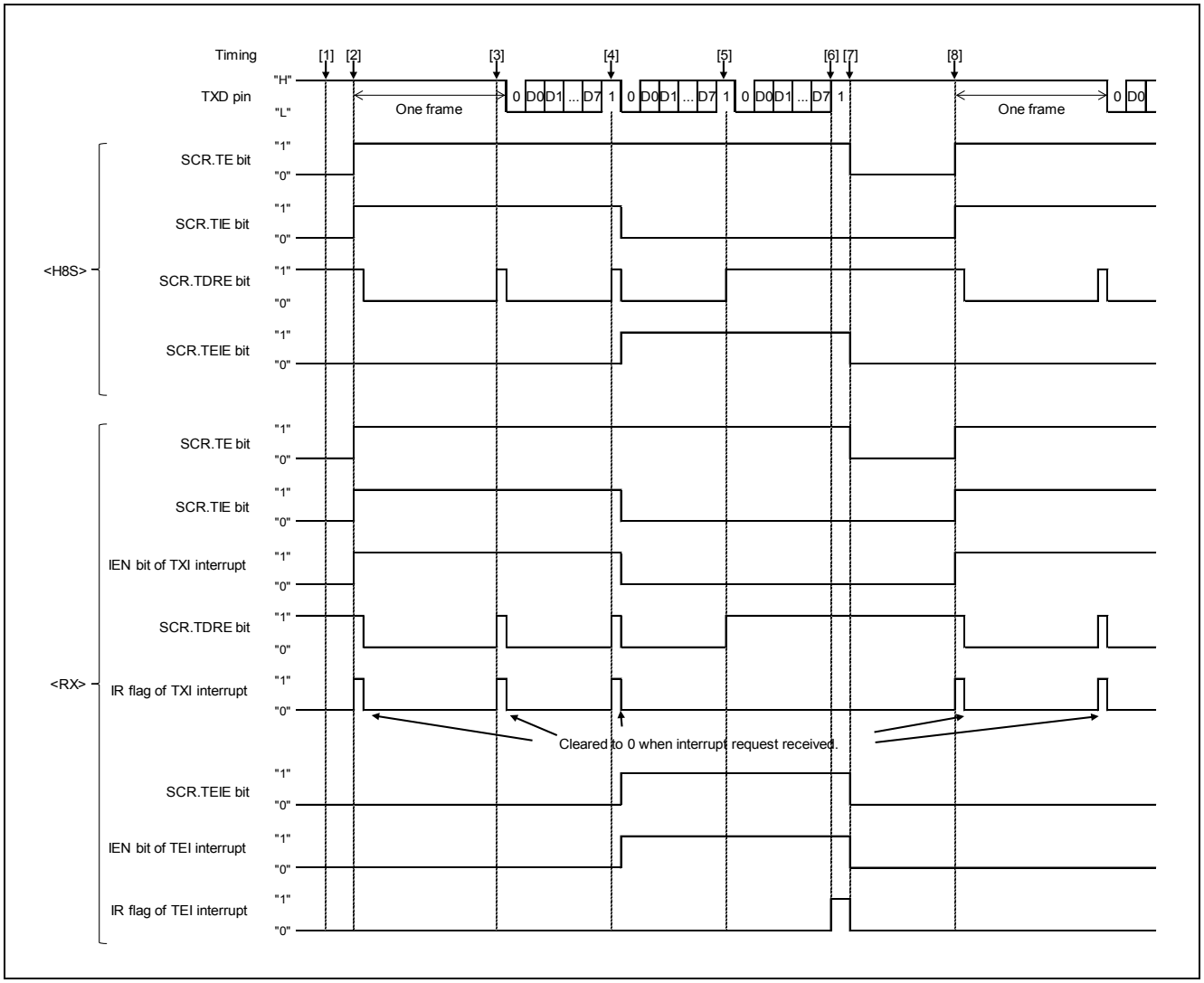


Figure 4.1 Points of Difference in Transmit Operation Timing on RX231 Group and H8S/2378 Group (Transmission in 3-Byte Increments)

Table 4.2 Points of Difference in Operation Timing and Processing Details on RX231 Group and H8S/2378 Group (Transmission in 3-Byte Increments)

Timing	RX (RX231)	H8S (H8S/2378)
[1] Before start of transmission	When the pin function is set to TXDn, the TXD pin is in the high-impedance state until the SCR.TE bit is set to 1 (transmission enabled).*1	The TXD pin is set as a general I/O port until the SCR.TE bit is set to 1 (transmission enabled).
[2] Start of transmission	The following bit settings are made to enable interrupts: SCR.TIE bit = 1 TXI interrupt IEN bit = 1 Also, the following bit settings are made to enable transmission: SCR.TE bit = 1 When the SCR.TE bit is set to 1, the IR flag of the transmit interrupt (TXI interrupt) is also set to 1. The first byte of transmit data is written in response to the transmit interrupt.	The following bit settings are made to enable interrupts: SCR.TIE bit = 1 Also, the following bit settings are made to enable transmission: SCR.TE bit = 1 When the SCR.TE bit is set to 1, a transmit interrupt (TXI interrupt) is generated. The first byte of transmit data is written in response to the transmit interrupt.
[3] Transfer of first byte of transmit data to transmit shift register	The transmit interrupt IR flag is set to 1, and a transmit interrupt is generated. The transmit interrupt handler then reads the second byte of data.	The SSR.TDRE bit is set to 1, and a transmit interrupt is generated. The receive interrupt handler then reads the second byte of data and clears the SSR.TDRE bit to 0.
[4] Transmit interrupt at write of final data	The following bit settings are made to disable the transmit interrupt: SCR.TIE bit = 0 TXI interrupt IEN bit = 0 In addition, the following bit settings are made to enable the transmit-end interrupt: SCR.TEIE bit = 1 TEI interrupt IEN bit = 1	The following bit settings are made to disable the transmit interrupt: SCR.TIE bit = 0 In addition, the following bit settings are made to enable the transmit-end interrupt: SCR.TEIE bit = 1
[5] After write of final data	No more transmit interrupts are generated.	
[6] End of transmission	A transmit-end interrupt is generated.	
[7] Transmit-end interrupt handler	The following bit settings are made to disable the transmit-end interrupt: SCR.TEIE bit = 0 Transmit-end interrupt IEN bit = 0 In addition, the following bit settings are made to disable transmission: SCR.TE bit = 0 When transmit and receive operation are disabled while the pin function is set to TXD, the TXD pin enters the high-impedance state.	The following bit settings are made to disable the transmit-end interrupt: SCR.TEIE bit = 0 In addition, the following bit settings are made to disable transmission: SCR.TE bit = 0 When transmit operation is disabled, the TXD pin becomes a general I/O port.
[8] Restart of transmission	Processing is the same as that described in [2] Start of transmission.	

Note 1. For details, refer to 9.4, Notes Regarding Transmit-Enable Bit.

4.1.2 Points of Difference in Transmit Operation Setting Procedure

Table 4.3 lists the points of difference in the initial setting procedure for transmit operation. The initial setting procedure shown for the H8S/2378 Group applies to the use of interrupt control mode 2.

Table 4.4 lists points of difference in transmit interrupt processing during transmit operation.

Table 4.5 lists points of difference in transmit-end interrupt processing during transmit operation.

Table 4.3 Points of Difference in Initial Setting Procedure for Transmit Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Cancel the module stop state.*1	SYSTEM.PRCR.WORD = 0xA502; MSTP(SCI5) = 0; SYSTEM.PRCR.WORD = 0xA500;	MSTPCR.BIT._SCI0 = 0;
2	Disable transmit/receive interrupts.	SCI5.SCR.BYTE = 0x00;	SCI0.SCR.BYTE = 0x00;
3	Make I/O port function settings.*2	PORTA.PMR.BIT.B4 = 0; MPC.PWPR.BIT.B0WI = 0; MPC.PWPR.BIT.PFSWE = 1; MPC.PA4PFS.BYTE = 0x0A; MPC.PWPR.BIT.PFSWE = 0; MPC.PWPR.BIT.B0WI = 1; PORTA.PMR.BIT.B4 = 1;	— (No processing)
4	Make transmit mode, etc., settings.	SCI5.SCR.BIT.CKE = 0; SCI5.SIMR1.BYTE = 0x00; SCI5.SPMR.BYTE = 0x00; SCI5.SMR.BYTE = 0x00; SCI5.SCMR.BYTE = 0xF2; SCI5.SEMR.BYTE = 0x00;	SCI0.SCR.BIT.CKE = 0; SCI0.SMR.BYTE = 0x01; SCI0.SCMR.BYTE = 0xF2;
5	Make bit rate settings.*3	SCI5.BRR = 0x4D;	SCI0.BRR = 0x0F;
6	Wait for 1-bit period.	— (No processing)	1-bit period wait processing
7	Make interrupt control mode setting.*4	— (No processing)	INTC.INTCR.BIT.INTM = 10b;
8	Make interrupt priority level setting.*5	IPR(SCI5,) = 0x01;	INTC.IPRI.BIT._SCI0 = 001b;
9	Clear interrupt requests.	IR(SCI5,TXI5) = 0;	— (No processing)
10	Enable peripheral function interrupt requests.	SCI5.SCR.BYTE |= 0xA0;	SCI0.SCR.BYTE |= 0xA0;
11	Enable transmit operation.		
12	Enable interrupt requests.*6	IEN(SCI5,TXI5) = 1;	— (No processing)
13	Make processor interrupt priority level setting.	— (No processing)	set_imask_exr(0);
14	Enable maskable interrupts.	setpsw_i();	— (No processing)

Note 1. For information on the module stop function, refer to section 7, Module Stop Function.

Note 2. On the RX231 Group peripheral function pin settings are made in the MPC. For details, refer to 9.1, I/O Ports.

Note 3. For details on the bit rate, refer to section 5, Calculating the Bit Rate.

Note 4. The RX231 Group has no interrupt control mode. For details, refer to section 6, Points of Difference between Interrupts

Note 5. For details of the interrupt priority level setting method, refer to section 6, Points of Difference between Interrupts

Note 6. The methods of enabling interrupt requests differ. For details, refer to section 6, Points of Difference between Interrupts

Table 4.4 Points of Difference in Transmit Interrupt Processing During Transmit Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Read TDRE flag.	— (No processing)	if(SCI0.SSR.BIT.TDRE == 1) {
2	Write transmit data.	/* Writes transmit data to SCI5.TDR register. */	/* Writes transmit data to SCI0.TDR register. */
3	Clear TDRE flag.	— (No processing)	SCI0.SSR.BIT.TDRE = 0;
4	Confirm final data write.	if(Final data write finished?) {	if(Final data write finished?) {

5	Disable transmit interrupt (after final data write only).	SCI5.SCR.BIT.TIE = 0
		IEN(SCI5,TXI5) = 0;
		while(0 != SCI5.SCR.BIT.TIE)
		{
		}
			SCI0.SCR.BIT.TIE = 0;
			while(0 != SCI0.SCR.BIT.TIE)
			{
			}

6	Clear interrupt request (after final data write only).	IR(SCI5,TXI5) = 0;
		while(0 != IR(SCI5,TXI5))
		{
		}
			—
			(No processing)

7	Enable transmit-end interrupt (after final data write only).	SCI5.SCR.BIT.TEIE = 1; IEN(SCI5,TEI5) = 1; }	SCI0.SCR.BIT.TEIE = 1; } }

Table 4.5 Points of Difference in Transmit-End Interrupt Processing During Transmit Operation

Step		RX (RX231)	H8S (H8S/2378)

1	Disable transmission and interrupts.	SCI5.SCR.BYTE &= 0x0B;
		while(0x00 != (SCI5.SCR.BYTE & 0xF4))
		{
		}
			SCI0.SCR.BYTE &= 0x0B;
			while(0x00 != (SCI0.SCR.BYTE & 0xF4))
			{
			}

2	Clear interrupt request.	IR(SCI5,TEI5) = 0;
		while(0 != SCI5,TEI5)
		{
		}
			—
			(No processing)*[1]

Note 1. The transmit-end interrupt request is cleared when the SCR.TEIE is cleared to 0.

4.2 Points of Difference in Receive Operation

The points of difference in receive processing during asynchronous serial communication are described below.

In the example of processing when a receive error is generated presented, the preconditions are that receive and receive-error interrupts are used.

4.2.1 Points of Difference in Receive Operation Timing

Figure 4.2 shows points of difference in receive operation timing on the RX231 Group and H8S/2378 Group (for receive operation in 3-byte increments).

Table 4.6 lists points of difference in operation timing and processing details on the RX231 Group and H8S/2378 Group (for receive operation in 3-byte increments).

The numbers [1] to [6] in Figure 4.2 correspond to items [1] to [6] in Table 4.6.

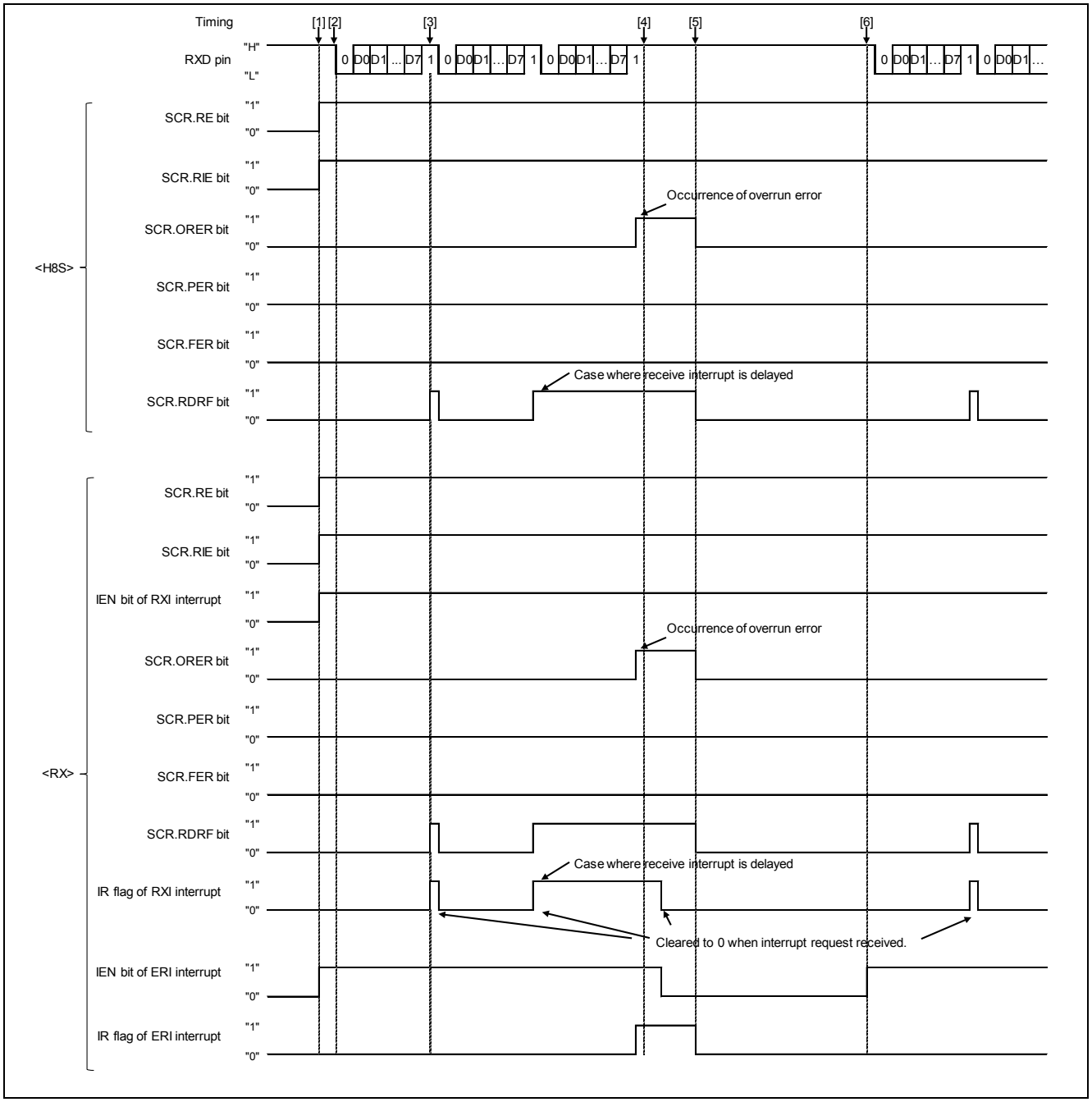


Figure 4.2 Points of Difference in Reception Timing on RX231 Group and H8S/2378 Group (Reception in 3-Byte Increments)

Table 4.6 Points of Difference in Operation Timing and Processing Details on RX231 Group and H8S/2378 Group (Reception in 3-Byte Increments)

Timing	RX (RX231)	H8S (H8S/2378)
[1] Reception enable setting	The following bit settings are made to enable interrupts: SCR.RIE bit = 1 RXI interrupt IEN bit = 1 ERI interrupt IEN bit = 1 The following bit settings are made to enable reception: SCR.RE bit = 1	The following bit settings are made to enable interrupts: SCR.RIE bit = 1 The following bit settings are made to enable reception: SCR.RE bit = 1
[2] Start of reception	When the start bit is detected, receive operation starts.	When the start bit is detected, receive operation starts.
[3] End of reception	After one byte of data is received, the received data is stored in the receive buffer, the receive interrupt (RXI interrupt) IR flag is set to 1, and a receive interrupt is generated. The value stored in the receive buffer is then read by the receive interrupt handler.	After one byte of data is received, the received data is stored in the receive buffer, the SSR.RDRF flag is set to 1, and a receive interrupt (RXI interrupt) is generated. The receive interrupt handler then reads the value stored in the receive buffer and clears the SSR.RDRF flag to 0.
[4] Occurrence of receive error	When an overrun error occurs, the receive-error interrupt (ERI interrupt) IR flag is set to 1. The ERI interrupt handler performs receive error processing.	When an overrun error occurs, the SSR.ORER flag is set to 1, and an ERI interrupt is generated. The ERI interrupt handler performs receive error processing.
[5] Clearing of receive error flag [6] Reissuance of reception enable setting	After the overrun error flag in the SSR register is read, 0 is written to it to clear the error flag. When all the error flags are cleared, the ERI interrupt IR flag is cleared to 0 and a transition to the reception-enabled state occurs.	After the overrun error flag in the SSR register is read, 0 is written to it to clear the error flag. When all the error flags are cleared, a transition to the reception-enabled state occurs.

4.2.2 Points of Difference in Receive Operation Setting Procedure

Table 4.7 lists the points of difference in the initial setting procedure for receive operation. The initial setting procedure shown for the H8S/2378 Group applies to the use of interrupt control mode 2.

Table 4.8 lists points of difference in receive interrupt processing during receive operation.

Table 4.9 lists points of difference in ERI interrupt processing during receive operation.

Table 4.7 Points of Difference in Initial Setting Procedure for Receive Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Cancel the module stop state.*1	SYSTEM.PRCR.WORD = 0xA502; MSTP(SCI5) = 0; SYSTEM.PRCR.WORD = 0xA500;	MSTPCR.BIT._SCI0 = 0;
2	Disable transmit/receive interrupts.	SCI5.SCR.BYTE = 0x00;	SCI0.SCR.BYTE = 0x00;
3	Make I/O port function settings.*2	PORTA.PMR.BIT.B3 = 0; MPC.PWPR.BIT.B0WI = 0; MPC.PWPR.BIT.PFSWE = 1; MPC.PA3PFS.BYTE = 0x0A; MPC.PWPR.BIT.PFSWE = 0; MPC.PWPR.BIT.B0WI = 1; PORTA.PMR.BIT.B3 = 1;	— (No processing)
4	Make receive mode, etc., settings.	SCI5.SCR.BIT.CKE = 0; SCI5.SIMR1.BYTE = 0x00; SCI5.SPMR.BYTE = 0x00; SCI5.SMR.BYTE = 0x00; SCI5.SCMR.BYTE = 0xF2; SCI5.SEMR.BYTE = 0x00;	SCI0.SCR.BIT.CKE = 0; SCI0.SMR.BYTE = 0x01; SCI0.SCMR.BYTE = 0xF2;
5	Make bit rate settings.*3	SCI5.BRR = 0x4D;	SCI0.BRR = 0x0F;
6	Wait for 1-bit period.	— (No processing)	1-bit period wait processing
7	Make interrupt control mode setting.*4	— (No processing)	INTC.INTCR.BIT.INTM = 10b;
8	Make interrupt priority level setting.*5	IPR(SCI5,) = 0x01;	INTC.IPRI.BIT._SCI0 = 001b;
9	Clear interrupt requests.	IR(SCI5,RXI5) = 0; IR(SCI5,ERI5) = 0;	— (No processing)
10	Enable peripheral function interrupt requests.	SCI5.SCR.BYTE |= 0x50;	SCI0.SCR.BYTE |= 0x50;
11	Enable transmit/receive operation.		
12	Enable interrupt requests.*6	IEN(SCI5,RXI5) = 1; IEN(SCI5,ERI5) = 1; /* *7 */	— (No processing)
13	Make processor interrupt priority level setting.	— (No processing)	set_imask_exr(0);
14	Enable maskable interrupts.	setpsw_i();	— (No processing)

Note 1. For information on the module stop function, refer to section 7, Module Stop Function.
Note 2. On the RX231 Group peripheral function pin settings are made in the MPC. For details, refer to 9.1, I/O Ports.
Note 3. For details on the bit rate, refer to section 5, Calculating the Bit Rate.
Note 4. The RX231 Group has no interrupt control mode. For details, refer to section 6, Points of Difference between Interrupts

Note 5. For details of the interrupt priority level setting method, refer to section 6, Points of Difference between Interrupts

Note 6. The methods of enabling interrupt requests differ. For details, refer to section 6, Points of Difference between Interrupts

Note 7. The specifications of the receive-error interrupt differ between microcontrollers. For details, refer to User's Manual: Hardware.

Table 4.8 Points of Difference in Receive Interrupt Processing During Receive Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Read RDRF flag.	— (No processing)	if(SCI0.SSR.BIT.RDRF == 1) {
2	Read receive data.	/* Reads receive data from SCI5.RDR register. */	/* Reads receive data from SCI0.RDR register. */
3	Clear RDRF flag.	— (No processing)	SCI0.SSR.BIT.RDRF = 0;
4	Confirm reception of final data.	if(Final data received?) {	if(Final data received?) {

5	Disable transmission/reception and interrupts (after reception of final data only).	SCI5.SCR.BYTE &= 0x0F;
		while(0x00 != (SCI5.SCR.BYTE & 0xF0))
		{
		}
			SCI0.SCR.BYTE &= 0x0F;
			while(0x00 != (SCI0.SCR.BYTE & 0xF0))
			{
			}
			}
			}

6	Clear interrupt request (after final data write only).	IR(SCI5,RXI5) = 0;
		while(0 != IR(SCI5,RXI5))
		{
		}
		}
			—
			(No processing)

Table 4.9 Points of Difference in ERI Interrupt Processing During Receive Operation

Step		RX (RX231)	H8S (H8S/2378)
1	Read ORER flag.	dummy = SCI5.SSR.BIT.ORER; /* *1 */	dummy = SCI0.SSR.BIT.ORER; /* *1 */

2	Processing when overrun error occurs	if(dummy == 1)
		{
		/* Indicate processing when overrun error occurs.*2 */
		}
			if(dummy == 1)
			{
			/* Indicate processing when overrun error occurs. */
			}

3	Read FER flag.	dummy = SCI5.SSR.BIT.FER; /* *1 */	dummy = SCI0.SSR.BIT.FER; /* *1 */

4	Processing when framing error occurs	if(dummy == 1)
		{
		/* Indicate processing when framing error occurs.*3 */
		}	

5	Read PER flag.	dummy = SCI5.SSR.BIT.PER; /* *1 */	dummy = SCI0.SSR.BIT.PER; /* *1 */

6	Processing when parity error occurs	if(dummy == 1)
		{
		/* Indicate processing when parity error occurs. */
		}	

7	Clear receive error.	SCI5.SSR.BYTE &= 0xC7;
		while(0 != (SCI5.SSR.BYTE & 0x38))
		{
		}
			SCI0.SSR.BYTE &= 0xC7;
			while(0 != (SCI0.SSR.BYTE & 0x38))
			{
			}

8	Clear interrupt request.	IR(SCI5,ERI5) = 0;
		while(0 != IR(SCI5,ERI5))
		{
		}
			—
			(No processing)

Note 1. Before clearing the overrun error (ORER) flag, framing error (FER) flag, or parity error (PER) flag, first read it while its value is 1.

Note 2. On the RX231 Group, read the RDR register as part of the overrun error handling.

Note 3. When a break is detected by means of a framing error, clear the SCR.RE bit to 0 and disable reception.

5. Calculating the Bit Rate

Table 5.1 shows the relationship between the BRR register setting value (N) and the bit rate (B). Note that the formulas listed in Table 5.1 are the same for the RX231 Group and H8S/2378 Group.

Table 5.1 Relationship between BRR Register Setting Value (N) and Bit Rate (B)

Mode	RX (RX231)	H8S (H8S/2378)
Clock-synchronous	$N = (PCLK \times 10^6 / (64 \times 2^{2n-1} \times B)) - 1$	$B = \phi \times 10^6 / (8 \times 2^{2n-1} \times (N+1))$
Asynchronous	[When BDGM = 0 and ABCS = 0 in SEMR] $N = (PCLK \times 10^6 / (64 \times 2^{2n-1} \times B)) - 1$	$B = \phi \times 10^6 / (64 \times 2^{2n-1} \times (N+1))$
	[When BDGM = 0 and ABCS = 1 in SEMR] [When BDGM = 1 and ABCS = 0 in SEMR] $N = (PCLK \times 10^6 / (32 \times 2^{2n-1} \times B)) - 1$	
	[When BDGM = 1 and ABCS = 1 in SEMR] $N = (PCLK \times 10^6 / (16 \times 2^{2n-1} \times B)) - 1$	

B: Bit rate [bps]

N: BRR register setting value ($0 \leq N \leq 255$)

PCLK or ϕ: Operating frequency [MHz]

n: Determined by the setting value of SMR. The relationship between the SMR setting value and n is shown in the Table below.
The relationship between the SMR setting value and n is the same on the RX231 Group and H8S/2378 Group.

SMR Setting Value		
CKS1	**CKS0**	**n**
0	0	0
0	1	1
1	0	2
1	1	3

Table 5.2 lists the functions of the SEMR.BDGM and SEMR.ABCS bits, which appear in Table 5.1.

Table 5.2 Functions of SEMR.BDGM and SEMR.ABCS Bits

Bit Name	Function
BDGM	Baud rate generator double-speed mode select bit (Only valid when using on-chip baud rate generator in asynchronous mode) 0: Baud rate generator outputs the clock with normal frequency. 1: Baud rate generator outputs the clock with doubled frequency.
ABCS	Asynchronous mode base clock select bit (Valid in asynchronous mode only) 0: Transfer rate with 16 base clock cycles as 1-bit period 1: Transfer rate with 8 base clock cycles as 1-bit period

6. Points of Difference between Interrupts

In contrast to the H8S/2378 Group, on the RX231 Group, in addition to interrupt enable and interrupt request bits in the registers of each peripheral function, there are interrupt enable and interrupt request bits in the registers of the interrupt controller for the peripheral functions.

Table 6.1 lists the points of difference between the interrupt-related resource of SCI0.

The IERm registers (m = 02h to 1Fh) and IRn registers (n = interrupt vector number) listed in Table 6.1 are interrupt controller registers.

For the interrupt sources corresponding to the bits in the IERm registers and the interrupt vector numbers, refer to the section describing the interrupt controller in User's Manual: Hardware.

Table 6.1 Points of Difference between SCI0 Interrupt-Related Resources

		RX (RX231)				H8S (H8S/2378)			
Item		**ERI0**	**RXI0**	**TXI0**	**TEI0**	**ERI0**	**RXI0**	**TXI0**	**TEI0**
Interrupt enable register (enable bit)	Peripheral function (SCI0)	SCR.RIE		SCR. TIE	SCR. TEIE	SCR.RIE		SCR. TIE	SCR. TEIE
	Interrupt controller	IER1A. IEN6	IER1A. IEN7	IER1B. IEN0	IER1B. IEN1	Not available.			
Interrupt request register (source flag)	Peripheral function (SCI0)	SSR. ORER SSR. FER SSR. PER	SSR. RDRF	SSR. TDRE	SSR. TEND	SSR. ORER SSR. FER SSR. PER	SSR. RDRF	SSR. TDRE	SSR. TEND
	Interrupt controller	IR214. IR	IR215. IR	IR216. IR	IR217. IR	Not available.			

Interrupts can be accepted on the RX231 Group when the following conditions are met:

- The I flag (PSW.I bit) is set to 1.
- The interrupt is enabled in the IER and IPR registers of the ICU.
- Interrupt requests are enabled by the corresponding peripheral function interrupt request enable bit.

Table 6.2 is a comparative listing of the interrupt generation conditions on the RX231 Group and H8S/2378 Group.

Table 6.2 Comparative Listing of Interrupt Generation Conditions on RX231 Group and H8S/2378 Group

Item	RX (RX231)	H8S (H8S/2378)
Interrupt enable bit (I bit)	Setting the I bit in the PSW register to 1 (enabled) enables acceptance of maskable interrupts.	In interrupt control mode 0, setting the I bit to 0 (enabled) in the CCR register enables acceptance of maskable interrupts. In interrupt control mode 2 the I bit in the CCR register is not used.
Processor interrupt priority level	Only interrupt requests with a higher priority level than that indicated by the IPL[3:0] bits in the PSW register are accepted.	In interrupt control mode 2 only interrupt requests with a higher priority level than that indicated by bits I2 to I0 in the EXR register are accepted. In interrupt control mode 0 the bits I2 to I0 in the EXR register is not used.
Interrupt priority level	Set in the IPR register.	In interrupt control mode 0 the default settings are used. In interrupt control mode 2 the IPR register settings are used.
Interrupt request flag	The interrupt controller manages all interrupt status flags for peripheral functions, external pins, NMI interrupts, etc.	The interrupt controller manages interrupt status flags for external interrupts, and interrupt status flags for internal interrupt sources are managed within each on-chip peripheral function.
Interrupt request enable	Set in the IER register for maskable interrupts and in the NMIER register for non-maskable interrupts.	IRQ interrupts are enabled by settings in the IER register.
Peripheral function interrupt enable	Interrupts can be enabled or disabled by each peripheral function.	

Table 6.3 lists points of difference in the enabling and priority levels of processor interrupts.

On the RX231 Group the processor interrupt priority level is 0 (lowest level) by default when the PSW.I bit is set to 1 (interrupt enabled), so maskable interrupts are enabled.

On H8S/2378 Group, in interrupt control mode 0, processor interrupt priority levels are not used when the CCR.I bit is cleared to 0 (interrupt enabled), so maskable interrupts are enabled.

On H8S/2378 Group, in interrupt control mode 2, the processor interrupt priority level is 7 (highest level) by default, so maskable interrupts are enabled by setting bits I2 to I0 in EXR.

Table 6.3 Points of Difference in Enabling and Priority Levels of Processor Interrupts

		H8S (H8S/2378)	
Item	**RX (RX231)**	**Interrupt Control Mode 0**	**Interrupt Control Mode 2**
Interrupt enable default value	PSW.I bit: 0 (interrupt mask)	CCR.I bit: 1 (interrupt mask)	Not used*[1]
Processor interrupt priority level default value	PSW.IPL[3:0] bits: 0000b (lowest level)	Not used*[1]	EXR bits I2 to I0: 111b (highest level)
Operation after a reset	Maskable interrupts are not accepted.		

Note 1. Don’t care.

Table 6.4 lists some of the embedded functions used for enabling interrupts.

Table 6.4 Embedded Functions Used for Enabling Interrupts (Partial Listing)

	Description		
		H8S (H8S/2378)	
Item	**RX (RX231)**	**Interrupt Control Mode 0**	**Interrupt Control Mode 2**
Processor interrupt enable setting	setpsw_i(); *[1]	set_imask_ccr(0); *[1]	Not used
Processor interrupt priority level setting (setting = 0)	set_ipl(0); *[1]	Not used	set_imask_exr(0); *[1]

Note 1. The file machine.h must be included.

For details, refer to the sections describing the interrupt controller (ICU), CPU, and peripheral functions used in User’s Manual: Hardware.

7. Module Stop Function

On the H8S/2378 Group and RX231 Group it is possible to halt the functioning of individual peripheral modules.

Power consumption can be reduced by transitioning unused peripheral modules to the module stop state. Modules not listed in Table 7.1 are in the module stop state after a reset.

Table 7.1 Modules that Operate under Initial Settings on RX231 Group and H8S/2378 Group

RX (RX231)	H8S (H8S/2378)
DMAC, DTC, RAM	EXDMAC, DMAC, DTC

When a module is in the module stop state, its registers cannot be read or written to.

Before using any module not listed in Table 7.1, it is necessary to cancel the module stop state and then make initial settings.

For details, refer to the section describing the low power consumption functions in User's Manual: Hardware.

8. Register Write Protection Function

On the RX231 Group it is possible to protect important registers from being overwritten if program runaway occurs. The protect register (PRCR) is used to specify the registers that are protected by this function.

Register protection can be enabled for the clock generation circuit–related registers, flash memory–related registers, operating mode–related registers, low power consumption function–related registers, low-power timer–related registers, LVD–related registers, and software reset register.

For details, refer to the section on the register write protection function in User's Manual: Hardware.

9. Key Points when Migrating from H8S to RX

Some points to keep in mind when migrating from the H8S/2378 Group to the RX231 Group are described below.

9.1 I/O Ports

On the RX231 Group it is necessary to make settings to the MPC to assign pins to peripheral function I/O signals.

To apply I/O control to a pin on the RX231 Group, make the following two settings:

- PFS register of MPC: Select the peripheral function to be assigned to the pin.
- PMR register of I/O port: Select whether to assign the pin to a general I/O port or a peripheral function.

Table 9.1 provides a comparative listing of I/O settings for peripheral function pins on the RX231 Group and H8S/2378 Group.

Table 9.1 Comparison of I/O settings for Peripheral Function Pins on RX231 Group and H8S/2378 Group

Function	RX (RX231)	H8S (H8S/2378)
Pin function selection	I/O pins for peripheral functions can be assigned from a selection of multiple pins by making settings in the PFS register.	Pins can be switched between general I/O port and peripheral function settings and pin functions selected through combinations of the MCU operating mode, the setting of the SYSCR.EXPE bit, the PFCR registers, the DDR registers, and the settings of the various peripheral functions.
General I/O port/peripheral function switching	Settings in the PMR register can be used to select whether specific pins are used as I/O ports or as peripheral functions.	

For details, refer to the sections describing the multi-function pin controller (MPC) and I/O ports in User's Manual: Hardware.

9.2 I/O Register Macros

The macro definitions listed below are contained in the I/O register definition file (iodefine.h) of the RX231 Group.

Using macro definitions can make program code easier to read.

Table 9.2 lists macro usage examples.

Table 9.2 Macro Usage Examples

Macro	Usage Example
IR(“module name”,”bit name”)	IR(MTU0,TGIA0) = 0; Clears the IR bit corresponding to TGIA0 of MTU0 to 0 (clear interrupt request).
DTCE(“module name”,”bit name”)	DTCE(MTU0,TGIA0) = 1; Sets the DTCE bit corresponding to TGIA0 of MTU0 to 1 (enable DTC start).
IEN(“module name”,”bit name”)	IEN(MTU0,TGIA0) = 1; Sets the IEN bit corresponding to TGIA0 of MTU0 to 1 (enable interrupt).
IPR(“module name”,”bit name”)	IPR(MTU0,TGIA0) = 0x02; Sets the IPR bits corresponding to TGIA0 of MTU0 to 2 (interrupt priority level 2).
MSTP(“module name”)	MSTP(MTU) = 0; Clears the module stop setting bit of MTU0 to 0 (cancel module stop state).
VECT(“module name”,”bit name”)	#pragma interrupt(Excep_MTU0_TGIA0(vect=VECT(MTU0,TGIA0))) Declares the interrupt function corresponding to TGIA0 of MTU0.

9.3 Embedded Functions

On the RX231 Group interrupt functions are provided to implement control register settings or special instructions. To use these embedded functions, include the file machine.h.

Table 9.3 lists (examples of) points of difference between control register settings and special instructions on the RX231 Group and H8S/2378 Group.

Table 9.3 Points of Difference between Control Register Settings and Special Instructions on RX231 Group and H8S/2378 Group (Example)

	Format	
Item	**RX (RX231)**	**H8S (H8S/2378)**
Set I flag to 1.	setpsw_i(); *1	set_imask_ccr(1); *1*2
Clear I flag to 0.	clrpsw_i(); *1	set_imask_ccr(0); *1*2
Expand to WAIT instruction.	wait(); *1	None
Expand to NOP instruction.	nop(); *1	nop(); *1

Note 1. It is necessary to include the file machine.h.

Note 2. I = 1 means enable interrupts on the RX231 Group, and I = 1 means mask interrupts on the H8S/2378 Group.

9.4 Notes Regarding Transmit-Enable Bit

When a pin's function is set to TXDn and the SCR.TE bit is cleared to 0 (serial transmit operation disabled), the pin's output enters the high-impedance state.

To prevent the TXDn line from entering the high-impedance state, do one of the following:

(1) Connect a pull-up resistor to the TXDn line.
(2) Change the pin's setting to "general I/O port, output" before clearing the SCR.TE bit to 0. Also, change the pin function to TXDn after setting the SCR.TE bit to 1.

10. Reference Documents

User's Manual: Hardware

H8S/2378 Group, H8S/2378R Group Hardware Manual Rev.7.00 (REJ09B0109-0700)
RX230 Group and RX231 Group User's Manual: Hardware Rev.1.10 (R01UH0496EJ0110)
(The latest versions can be downloaded from the Renesas Electronics website.)

Application Note

RX Family, M16C Family Migrating From the M16C Family to the RX Family: Clock Synchronous Serial Data Communications Rev.1.00 (R01AN1927EJ0100)
(The latest versions can be downloaded from the Renesas Electronics website.)

Technical Update/Technical News

(The latest versions can be downloaded from the Renesas Electronics website.)

User's Manual: Development Environment

CC-RX Compiler User's Manual Rev.1.04 (R20UT3248EJ0104)
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package Ver.6.01 User's Manual (REJ10B0161-0100)
(The latest versions 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

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Nov. 13, 2017	—	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit 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 Microprocessing unit or Microcontroller unit products 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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other disputes involving 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, including but not limited to, the product data, drawing, chart, program, algorithm, application examples.
3. 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 shall not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics products.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended 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 (traffic lights); large-scale communication equipment; key financial terminal systems; safety control 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 (space and undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability 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. When using the Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident 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 Renesas Electronics 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 ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of failure or malfunction of Renesas Electronics products, 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 by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, 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 investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall 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 shall not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
11. This document shall not be reprinted, 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.

(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.

(Rev.3.0-1 November 2016)

RENESAS

Renesas Electronics Corporation

http://www.renesas.com

SALES OFFICES

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
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

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 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

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 1207, 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 India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 6.0