RL78/G14, R8C/36M Group

Migration Guide from R8C to RL78:
Synchronous Serial Communication Unit (SSU)

Introduction

This application note describes how to provide the serial array unit (SAU) of the RL78/G14 with the communication modes equivalent to clock synchronous communication mode and 4-wire bus communication mode of the synchronous serial communication unit (SSU) incorporated in the clock synchronous serial interface of the R8C/36M group.

Target Device

RL78/G14, R8C/36M Group

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

Contents

RENESAS

1.   Migration Method from R8C Family to RL78 Family

The following sections describe how to implement the operations (clock synchronous communication mode and 4-wire bus communication mode) of the synchronous serial communication unit (SSU) of the R8C/36M group by using the serial array unit (SAU) of the RL78/G14.

Table 1.1 shows the SSU operation modes of the R8C/36M group. Table 1.2 shows the SAU operation modes of the RL78/G14.

Table 1.1 Operation Modes of Synchronous Serial Communication Unit (SSU) in R8C/36M Group (Summary)

Synchronous serial communication unit (SSU) in R8C/36M Group	
Operation Mode	Function
Clock synchronous communication mode	Uses the transfer clocks for transmission and reception.
4-wire bus communication mode	Uses four bus lines for communication: clock line, data input line, data output line, and chip-select line.

Table 1.2 Operation Modes of Serial Array Unit (SAU) in RL78/G14 (Summary)

Serial Array Unit (SAU) in RL78/G14	
Operation Mode	Function
3-wire serial I/O	Transmits and receives data in synchronization with the serial clock (SCK) output from the master channel.
UART	A start-stop synchronization function using two lines: the serial data transmission (TxD) and serial data reception (RxD) lines
Simplified I2C	A clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA)

The SSU of the R8C/36M group supports clock synchronous communication mode and 4-wire bus communication mode.

Each unit of the SAU of the RL78/G14 has multiple serial channels and uses a single channel or a combination of multiple channels to support the 3-wire serial I/O (CSI), UART communication, and simplified I2C.

Table 1.3 shows the relationship between the SSU and SAU.

Table 1.3 SSU-to-SAU Relationship

R8C/36M Synchronous serial communication unit (SSU)	RL78/G14 Serial Array Unit (SAU)
Operation Mode	Operation Mode
Clock synchronous communication mode	3-wire serial I/O communication (CSI)

4-wire bus communication mode
 	
• Selection of a slave for master transmission/reception
 	
    Uses the chip select line or controls output through the I/O port.
 	
• For slave transmission/reception
 	
    Uses the chip select line or controls input through the I/O port.
	3-wire serial I/O communication (CSI)
	
	• Selection of a slave for transmission/reception
	
	    Uses the pot I/O.
	
	• For slave transmission/reception
	
	    Uses the slave select input function or controls input through the port I/O.

The clock synchronous communication mode of the SSU of the R8C/36M group can be implemented by using 3-wire serial I/O (CSI) mode of the SAU of the RL78/G14.

Four-wire bus communication mode of the SSU of the R8C/36M group is implemented by using the four pins for the clock line, data input line, data output line, and chip select line. The chip select line is active low. When the R8C/36M group is a master device, the chip select line is used to select the slave device. When the R8C/36M group is a slave device, the chip select line is used to receive the signals from the master device.

The RL78/G14 uses 3-wire serial I/O (CSI) mode of the SAU and the SSI00 pin or one digital I/O port pin to provide 4-wire bus communication mode of the SSU of the R8C/36M group. The SSI00 pin is active low. The clock line, data input line, and data output line of the SSU are implemented by using the pertinent pins of 3-wire serial I/O (CSI) mode of the SAU of the RL78/G14. The chip select line of the SSU is implemented by using the slave select input function (SSI00 pin (chip select input)) of the RL78/G14 or the digital I/O port. When the RL78/G14 is a master device, the digital I/O port is used to select the slave device. When the RL78/G14 is a slave device, the SSI00 pin (chip select input) is used to receive the signals from the master device. If the serial clock is input while a high-level signal is input to the SSI00 pin, the SAU does not perform transmission or reception.

In this application note, chapter 2. Differences between RL78/G14 and the R8C/36M Group describe the differences between the modes of the SSU of the R8C/36M group (clock synchronous communication mode and 4-wire bus communication mode) and 3-wire serial I/O (CSI) mode of the SAU of the RL78/G14.

Since clock synchronous communication mode of the R8C/36M group can be implemented by using 3-wire serial I/O (CSI) mode of the SAU of the RL78/G14, only a migration example from 4-wire bus communication mode is given in the attached sample code.

If you are considering migration from clock synchronous communication mode, refer to the text of this application note and the following application notes.

- RL78/G13 Serial Array Unit for 3-Wire Serial I/O (Master Transmission/Reception) CC-RL (R01AN2547E)
- RL78/G13 Serial Array Unit for 3-Wire Serial I/O (Slave Transmission/Reception) CC-RL (R01AN2711E)

2.   Differences between RL78/G14 and the R8C/36M Group

This chapter describes the differences between RL78/G14 and the R8C/36M group.

2.1   Differences between Clock synchronous communication mode and 3-Wire Serial I/O Communication (CSI)

Table 2.1 shows the differences between clock synchronous communication mode of the R8C/36M group and 3-wire serial I/O (CSI) of the RL78/G14.

Table 2.1 Differences between Clock Synchronous Communication Mode And
3-Wire Serial I/O Communication (CSI)

Item	R8C/36M Group Synchronous Serial Communication Unit (SSU) Clock synchronous communication mode	RL78/G14 3-Wire Serial I/O Communication (CSI)
Transfer clock	Internal clock or external clock	Internal clock or external clock
Data length	8 to 16 bits	7/8 bits
Interrupt function	transmit-end, transmit-data-empty, receive-data-full, overrun error	• For transmission: transfer end interrupt and buffer empty interrupt • For reception: transfer end interrupt
Error detection	Overrun error	Overrun error
Selection of data phase	Not provided	Provided
Selection of clock phase	Not provided	Provided
Use in STOP mode	Not provided	Enabled (SNOOZE mode function) Note 1

Notes

1.   Enabled only when the internal clock is selected.

2.2 Differences between 4-Wire Bus Communication Mode And 3-Wire Serial I/O Communication (CSI)

Table 2.2 shows the differences between 4-wire bus communication mode of the R8C/36M group and 3-wire serial I/O (CSI) of the RL78/G14.

Table 2.2 Differences between 4-Wire Bus Communication Mode And
3-Wire Serial I/O Communication (CSI)

Item	R8C/36M Group Synchronous Serial Communication Unit (SSU) 4-wire bus communication mode	RL78/G14 3-Wire Serial I/O Communication (CSI)
Transfer clock	Internal clock or external clock	Internal clock or external clock
Data length	8 to 16 bits	7/8 bits
Interrupt function	transmit-end, transmit-data-empty, receive-data-full, overrun error, conflict error	• For transmission: transfer end interrupt and buffer empty interrupt • For reception: transfer end interrupt
Error detection	Overrun error	Overrun error
Multimaster error detection	Conflict error	Not provided
Selection of data phase	Not provided	Provided
Selection of clock phase / polarity	Provided	Provided
Use in STOP mode	Not provided	Enabled (SNOOZE mode function) Note 1
Slave select function	Provided (SCS: Chip-select I/O pin)	Provided (SSI00: Slave select input function)
Bidirectional mode (1 data I/O pin used for communication)	Provided	Not provided
Arbitration function	Provided	Not provided

Notes

1. Enabled only when the internal clock is selected.

2.3　　Comparison between Registers

Table 2.3, Table 2.4, and Table 2.5 compare the registers for the SSU functions of the R8C/36M group to the corresponding registers for the SAU of the RL78/G14.

Table 2.3　Comparison between Registers (1)

Setting Items	R8C/36M Group	RL78/G14
Enabling clock supply to peripheral hardware	MSTCR register 　　MSTIIC bit	PER0 register 　　Bits SAU0EN, SAU1EN
Communication mode select	SSUIICSR register 　　IICSEL bit SSMR2　register 　　SSUMS bit	SMRmn register 　　Bits MDmn1 and MDmn2
Data length select	SSBR register 　　BS0 to BS3 bits	SCRmn register 　　Bits DLSmn0 and DLSmn1
Bit rate	SSCRH register 　　CKS0 to CKS2 bits	SDRmn register ^{Note}
Transmit buffer	SSTDR register	SDRmn register ^{Note}
Selection of count source	f1 only	CKC register 　　Bits CSS and MCM0 SPSm register SMRmn register 　　Bits CKSmn and CCSmn
Communication status indication flag	−	SSRmn register 　　TSFmn bit
Output mode select of data output pin	SSMR2 register 　　SOOS bit	POMxx register
Clock polarity select	SSMR register 　　CPOS bit	−
Data phase select	−	SCRmn register 　　DAPmn bit
Clock phase select	SSMR register 　　CPHS bit	SCRmn register 　　CKPmn bit

Note　The lower 8 or 9 bits function as a transmit/receive buffer register.
　　　　The higher 7 bits are used as a register that sets the division ratio of the operation clock (f_{MCK}).

Remark

　　　　−: There are no corresponding registers.

　　　　i = 0, 1

　　　　m = Unit number (0, 1)

　　　　n = Channel number (0 to 3)

　　　　xx = 0, 1, 3, 5, 7

Table 2.4 Comparison between Registers (2)

Setting Items	R8C/36M Group	RL78/G14
Selection of MSB first/LSB first	SSMR register 　MLS bit	SCRmn register 　DIRmn bit
Transmit enable	SSER register 　TE bit	SCRmn register 　TXEmn bit SSm register SMRmn register 　STSmn bit
Transmit end flag	SSSR register 　TEND bit	−
Transmit data empty flag	SSSR register 　TDRE bit	SSRmn register 　BFFmn bit
Receive enable	SSSR register 　RE bit	SCRmn register 　RXEmn bit SSm register SMRmn register 　STSmn bit
Receive data register full flag	SSSR register 　RDRF bit	SSRmn register 　BFFmn bit
Selection of transmit interrupt (Transmit end or transmit data empty)	SSER register 　TEIE bit 　TIE bit	SMRmn register 　MDmn0 bit
Selection of receive operation after receiving 1 byte of data	SSCRH register 　RSSTP bit	−
Receive buffer	SSRDR register	SDRmn register Note
Overrun error flag	SSSR register 　ORER bit	SSRmn register 　OVFmn bit
Conflict error flag	SSSR register 　CE bit	−

Note　The lower 8 or 9 bits function as a transmit/receive buffer register.
　　　The higher 7 bits are used as a register that sets the division ratio of the operation clock (f_{MCK}).

Remark

　　−: There are no corresponding registers.

　　i = 0, 1

　　m = Unit number (0, 1)

　　n = Channel number (0 to 3)

　　xx = 0, 1, 3, 5, 7

Table 2.5 Comparison between Registers (3)

Setting Items	R8C/36M Group	RL78/G14

Pin select	SSUIICSR register
	    IICSEL bit
	SSMR2 register
	    SSUMS bit
	    CSS1bit, CSS0 bit
	    SCKS bit
	    BIDE bit
	SSCRH register
	    MSS bit
	SSER register
	    TE bit, RE bit
		PIM0, PIM1, PIM3, PIM5 registers
		POM0, POM1, POM3, POM5, POM7 registers
		PM0, PM1, PM3, PM5 to PM7 registers
		P0, P1, P3, P5 to P7 registers

Receive interrupt enable (receive data full and overrun error)	SSER register RIE bit	−
Conflict error interrupt enable	SSER register CEIE bit	−

Clear each error flag	SSSR register
	    CE bit
	    ORER bit
		SIRmn register

Operation stop trigger of channel n	−	STm register STmn bit
Indication of operation enable/stop status of channel n	−	SEm register SEmn bit
Serial output enable/stop of channel n	−	SOEm register SOEmn bit
Serial clock output	−	SOm register CKOmn bit

Serial data output	SSCRL register
	    SOLP bit, SOL bit
		SOm register
		    SOmn bit

Operation Enable setting in slave mode	SSMR2 register
	    CSS1 bit, CSS0 bit
		ISC register
		    SSIE00 bit

Remark

    −: There are no corresponding registers.

    m = Unit number (0, 1)

    n = Channel number (0 to 3)

3.   How to Migrate Synchronous Serial Communication Unit in This Sample Code

   In this sample program, the Synchronous serial communication unit operation of the R8C/36M group is realized with RL78/G14 by the method shown in Table 3.1. For detailed contents of the sample program, please refer to the following chapters.

Table 3.1 How to Migrate from R8C/36M group to RL78/G14 in This Sample Program

R8C/36M Group Synchronous serial communication unit (SSU)	RL78/G14 Serial array unit (SAU)
Operation Mode	Operation Mode

4-Wire Bus Communication Mode
 	
• Selection of a slave for master transmission/reception
 	
   Uses the chip select line.
 	
• Selection of a slave for slave transmission/reception
 	
   Uses the chip select line.
	3-Wire Serial I/O Communication
	
	• Selection of a slave for master transmission/reception
	
	   Uses the pot I/O.
	
	・Selection of a slave for slave transmission/reception
	
	   Uses the slave select input function.

Note   Since 4-wire bus communication mode of the R8C/36M group includes the functions of clock synchronous communication mode, only a migration example from 4-wire bus communication mode is given in the attached sample code.

   If you are considering migration from clock synchronous communication mode, refer to the text of this application note and the following application notes.

- •   RL78/G13 Serial Array Unit for 3-Wire Serial I/O (Master Transmission/Reception) CC-RL (R01AN2547E)
- •   RL78/G13 Serial Array Unit for 3-Wire Serial I/O (Slave Transmission/Reception) CC-RL (R01AN2711E)

4. Example of Migration from 4-Wire Bus Communication Mode (Master transmission/reception)

4.1 Specifications

To support the communication operations (master transmission/reception) in 4-wire bus communication mode of the SSU of the R8C/36M group by using the RL78/G14, 3-wire serial I/O communication (CSI) of the serial array unit (SAU) is used.

The RL78/G14, operated as the master, alternately transmits 0x05 and 0x50 to the slave, and receives data from the slave. The slave select function is implemented by the chip select signal (_CS signal) using the port I/O.

For data transmission/reception, it is also necessary to check in advance if the corresponding device connection target device is ready for communication (handshake operation). Therefore, the handshake signal (_BUSY signal) has been added to the serial communication signals.

Table 4.1 lists the peripheral functions to be used and their uses. Figure 4.1 presents an Overview of CSI Operation.

Figure 4.2 and Figure 4.3 show timing charts for explaining the CSI communication.

Table 4.1 Peripheral Functions to be Used and Their Uses

Peripheral Function	Use
Serial array unit 0 channel 0	CSI00 master transmission/reception
Timer array unit 0 channel 0	Interval timer operation

Caution: When the slave is busy, normal communication between the master and slave is impossible.
Through handshaking between the master and slave, make sure that the slave is not busy.
Then, perform communication between them.
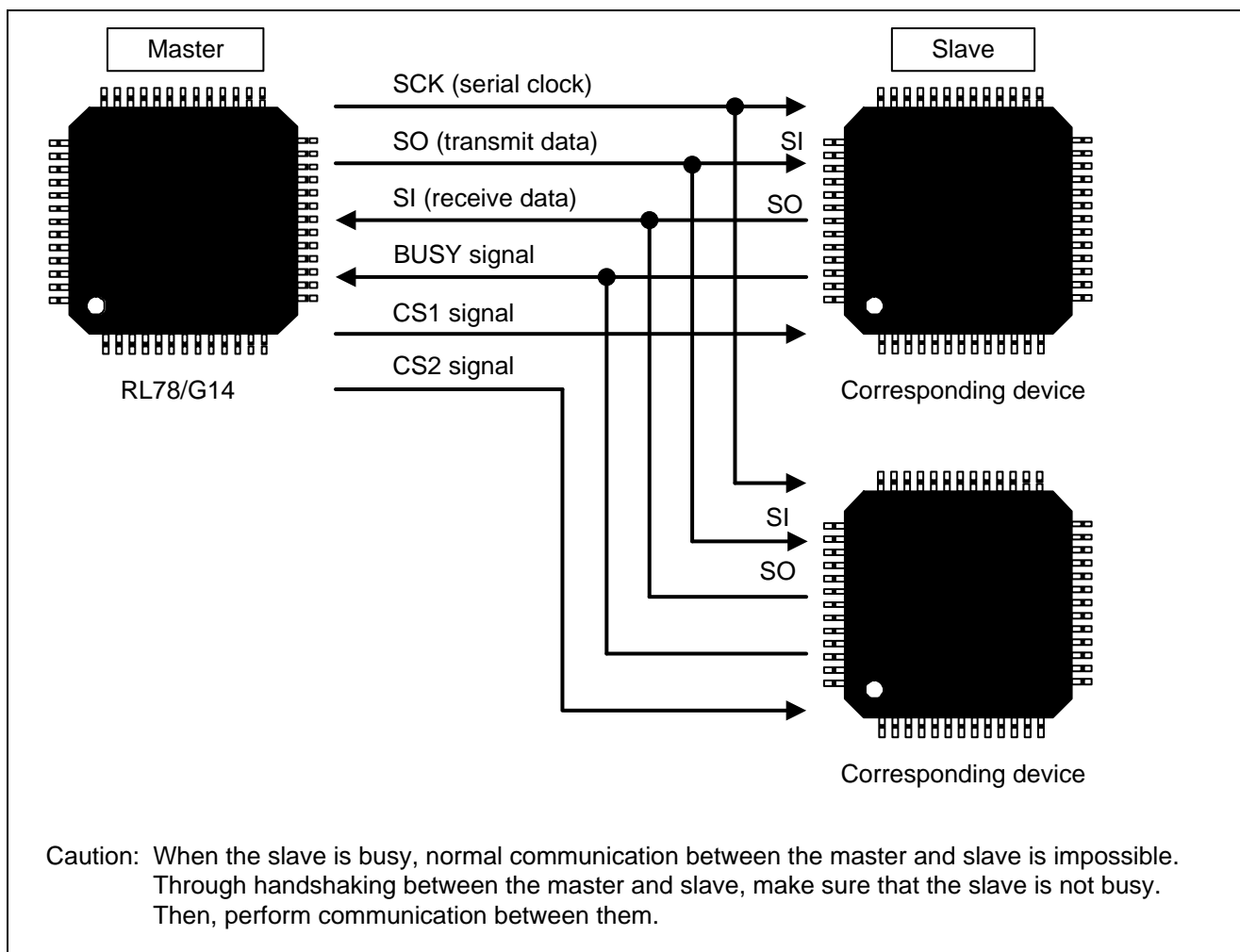
Figure 4.1 Overview of CSI Operation
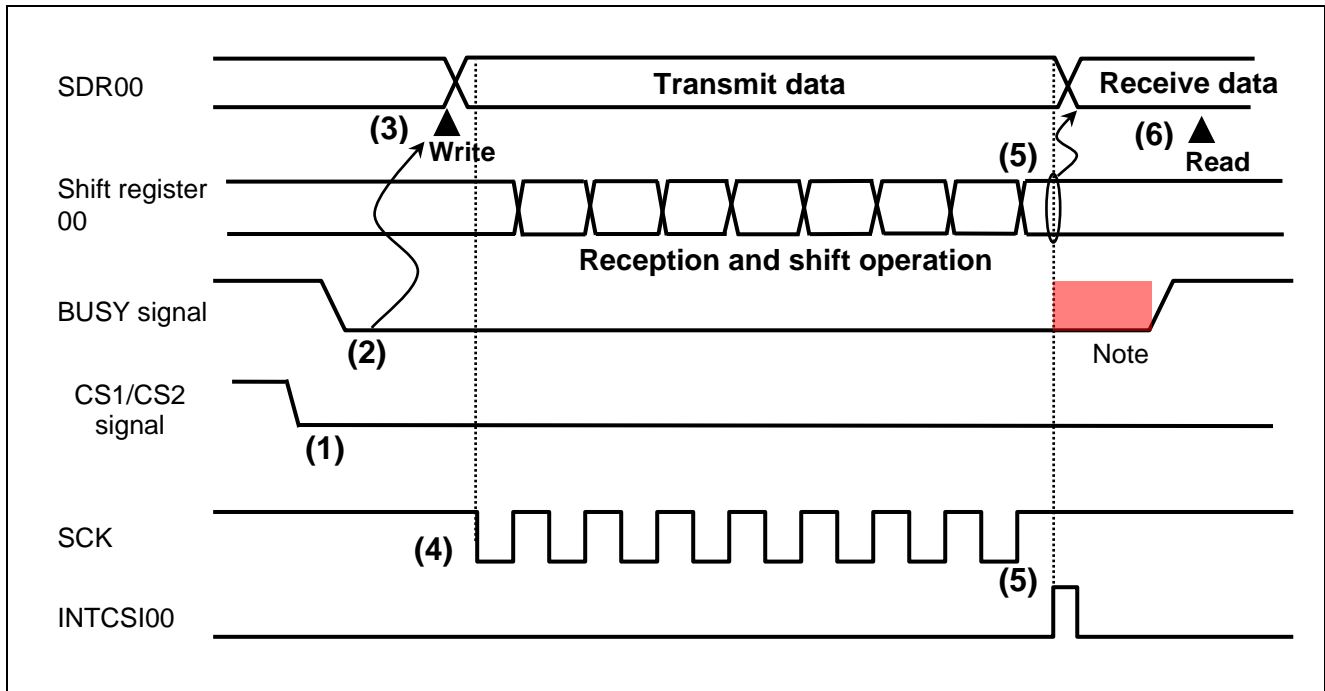
Figure 4.2 Handshake Operation and Communication

(1) [Software processing] Assert the _CS1 or_CS2 pin to select the target slave.

(2) [Software processing] Make sure that the slave is not busy.

(3) [Software processing] Write transmit data to the SDR00 register and then start CSI00 transmission/reception.

(4) [Hardware processing] Write data to the SDR00 register, output serial clock signals, and then enter the communication status.

(5) [Hardware processing] Transfer receive data from the shift register 00 to the SDR00 register and then generate a transfer end interrupt.

(6) [Software processing] Read the receive data from the SDR00 register.

Note: If the transmission/reception is restarted before the BUSY signal from the slave rises, the expected results may not be obtained. As an example of master operation to prevent this phenomenon, the timing chart (Figure 4.3) shows operation using the falling edge of the BUSY signal.
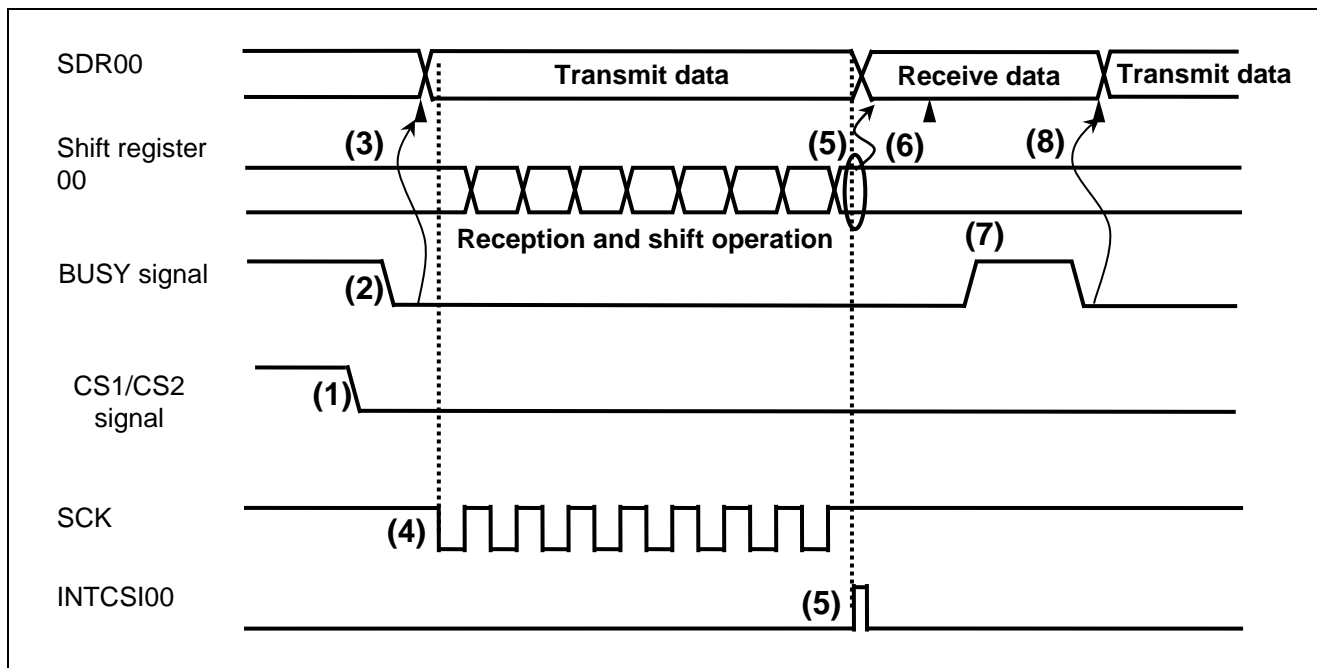
Figure 4.3 Example of BUSY Signal Edge Detection in the Master

・BUSY signal edge detection

In this example, the master starts communication upon detection of the falling edge of the BUSY signal from the slave.

(1) [Software processing in the master]
  Assert the _CS1 or_CS2 pin to select the target slave.
(2) [Software processing in the slave] Write the next transmit data to the master and make the BUSY signal fall.
(3) [Software processing in the master] Detect the falling edge of the BUSY signal and write transmit data to the SDR00 register.
(4) [Hardware processing in the master] Start transmission/reception and then output serial clock (SCK) signals.
(5) [Hardware processing in the master] After completion of the transfer, set the value of shift register 00 in the SDR00 register and then generate a transfer end interrupt (INTCSI00).
(6) [Software processing in the master] Read the receive data from the SDR00 register.
(7) [Software processing in the master] Wait until the falling edge of the BUSY signal is detected. (注)
(8) [Software processing in the master] Detect the falling edge of the BUSY signal and then write the transmit data to the SDR00 register.

Note: If the BUSY signal is held at the high level for a short period, the software may be unable to detect the edge. In this case, input the BUSY signal to an external interrupt pin (such as the INTP0 pin) so that the hardware detects the edge.

4.2 Operation Check Conditions

The sample code contained in this application note has been checked under the conditions listed in the table below.

Table 4.2 Operation Check Conditions

Item	Description
Microcontroller used	RL78/G14 (R5F104LEA)
Operating frequency	High-speed on-chip oscillator (HOCO) clock: 32 MHz CPU/peripheral hardware clock: 32 MHz
Operating voltage	5.0 V (Operation is possible over a voltage range of 2.9 V to 5.5 V.) LVD operation (V_{LVD}): Reset mode which uses 2.81 V (2.76 V to 2.87 V)
Integrated development environment (CS+)	CS+ V5.00.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.04.00 from Renesas Electronics Corp.
Integrated development environment (e2 studio)	e2 studio V5.4.0.018 from Renesas Electronics Corp.
C compiler (e2 studio)	CC-RL V1.04.00 from Renesas Electronics Corp.

4.3   Description of the Hardware

4.3.1   Hardware Configuration Example

Figure 4.4 shows an example of hardware configuration that is used for this application note.
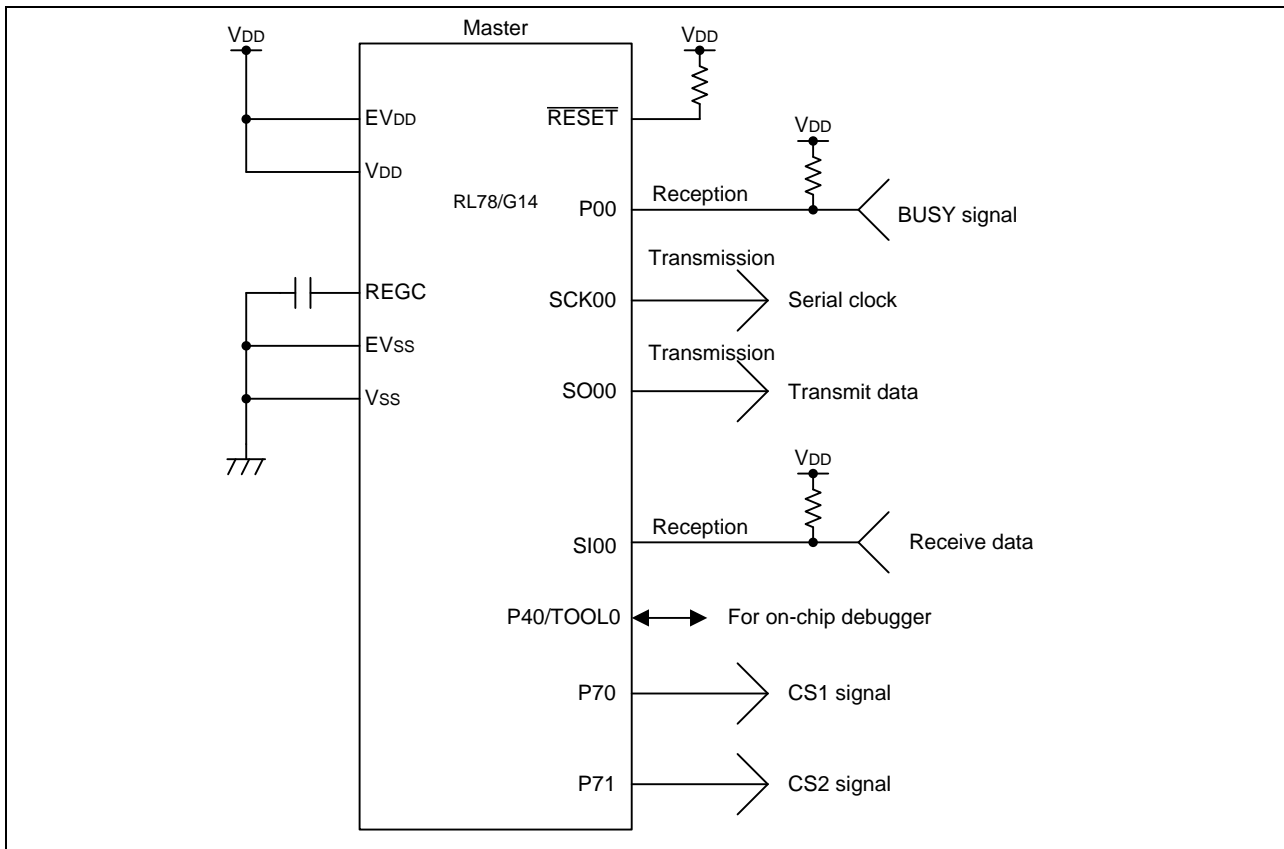
Figure 4.4 Hardware Configuration

Cautions:

1. The purpose of this circuit is only to provide the connection outline and the circuit is simplified accordingly. When designing and implementing an actual circuit, provide proper pin treatment and make sure that the hardware's electrical specifications are met (connect the input-only ports separately to VDD or VSS via a resistor).

2. Connect any pins whose name begins with EVSS to VSS and any pins whose name begins with EVDD to VDD, respectively.

3. VDD must be held at not lower than the reset release voltage (V_{LVD}) that is specified as LVD.

4.3.2 List of Pins to be Used

Table 4.3 lists the pins to be used and their functions.

Table 4.3 Pins to be Used and Their Functions

Pin Name	I/O	Description
P30/INTP3/RTC1HZ/SCK00/SCL00/TRJO0	Output	Serial clock output pin
P50/INTP1/SI00/RxD0/TOOLRxD/SDA00/TRGIOA	Input	Data reception pin
P51/INTP2/SO00/TxD0/TOOLTxD/TRGIOB	Output	Data transmission pin
P70/KR0/SCK21/SCL21	Output	_CS1 signal assert pin (slave 1 is enabled)
P71/KR1/SI21/SDA21	Output	_CS2 signal assert pin (slave 2 is enabled)
P00/TI00/TRGCLKA	Input	BUSY signal detection pin

4.4   Description of the Software

4.4.1   Operation Outline

The sample program covered in this application note transmits and receives data to and from the corresponding device (slave) via the CSI (master transmission/reception). It supplies clock signals to the slave, transmits data (0x05 or 0x50) to the slave, and receives data from the slave at intervals of about 10 ms. This communication is performed in full-duplex mode.

(1) Initialize SAU0.

< Conditions for setting >

- Use SAU0 channel 0 as the CSI.

- Set the serial clock frequency to about 312,500 Hz.

- Select the single transfer mode as the operation mode.

- Select type 1 as the phase between data and clock signals.

- Set data transfer order to the MSB first.

- The length of data should be 8 bits.

- A serial transfer end interrupt (INTCSI00) should occur in single transfer mode.

- Use the P30/SCK00 pin for clock output and set the initial output value to 1.

- Use the P51/SO00 pin for data output and set the initial output value to 1.

- Use the P50/SI00 pin for data input.

- Enable output for serial communication.

(2) Controlling the communication interval (10 ms) uses the interval timer function of the timer array unit (TAU) channel 0. The system starts the interval timer and then executes a HALT instruction. When the system is in HALT mode, it waits for the occurrence of a timer interrupt (INTTM00).

(3) When a timer count end interrupt occurs and moreover the system exits the HALT mode, the system checks whether communication is possible. If the communication is possible, the system transmits/receives data. If no communication is underway and moreover the slave is not busy, the system determines that communication is possible and transmits/receives data.

(4) When data transmission/reception is already completed or if communication is impossible, the system executes the HALT instruction again. Then, the system enters HALT mode to wait for the occurrence of a timer interrupt (INTTM00).

Caution: For information about timer array unit setup, refer to the RL78/G13 Timer Array Unit Interval Timer (R01AN2576E) Application Note.

4.5 List of Option Byte Settings

Table 4.4 summarizes the settings of the option bytes.

Table 4.4 Option Byte Settings

Address	Value	Description
000C0H/010C0H	01101110B	Disables the watchdog timer (Stops counting after the release from the reset state.)
000C1H/010C1H	01111111B	LVD reset mode, 2.81 V (2.76 V to 2.87 V)
000C2H/010C2H	11101000B	HS mode, HOCO: 32 MHz
000C3H/010C3H	10000100B	Enables the on-chip debugger.

4.5.1 List of Constants

Table 4.5 lists the constants that are used in this sample program.

Table 4.5 Constants for the Sample Program

Constant	Setting	Description
_0001_TAU_CH0_START_TRG_ON	0x0001U	Enables TAU0 channel 0 operation.
_0100_SAU_CH0_CLOCK_OUTPUT_1	0x0100U	Sets the serial clock output value for SAU0 channel 0.
_0001_SAU_CH0_DATA_OUTPUT_1	0x0001U	Sets the serial data output value for SAU0 channel 0.
_0001_SAU_CH0_OUTPUT_ENABLE	0x0001U	Enables output for SAU0 channel 0 serial communication.
_0001_SAU_CH0_START_TRG_ON	0x0001U	Starts SAU0 channel 0 operation.
_0001_SAU_OVERRUN_ERROR	0x0001U	Acquires the overrun error detection flag for SAU0 channel 0.
_FE_SLAVE_NO1	0xFEU	ANDed with P7 to select slave 1.
_FD_SLAVE_NO2	0xFDU	ANDed with P7 to select slave 2.

4.5.2 List of Variables

Table 4.6 lists the global variables that are used in this sample program.

Table 4.6 Global Variables for the Sample Program

Type	Variable Name	Contents	Function Used
unsigned char	g_tx_data	Serial transmit data	main()
unsigned char	g_rx_data	Serial receive data	main()
uint8_t	gp_csi00_rx_address	CSI00 receive buffer address	R_CSI00_Send_Receive() r_csi00_interrupt()
uint8_t	gp_csi00_tx_address	CSI00 transmit buffer address	R_CSI00_Send_Receive() r_csi00_interrupt()
uint16_t	g_csi00_tx_count	CSI00 transmit data size	R_CSI00_Send_Receive() r_csi00_interrupt()

4.5.3　　List of Functions

Table 4.7 summarizes the functions that are used in this sample program.

Table 4.7　Functions

Function Name	Outline
R_TAU0_Channel0_Start	Starts TAU0 channel 0 operation.
R_CSI00_Start	Starts CSI00 operation.
R_CSI00_Send_Receive	CSI00 data transmission/reception function
r_csi00_interrupt	CSI00 transfer end interrupt function
r_csi00_callback_sendend	CSI00 callback function upon transfer end

4.5.4　　Function Specifications

This section describes the specifications for the functions that are used in this sample program.

[Function Name] R_TAU0_Channel0_Start

Synopsis	TAU0 channel 0 operation start
Header	r_cg_macrodriver.h, r_cg_timer.h, and r_cg_userdefine.h
Declaration	void R_TAU0_Channel0_Start(void)
Explanation	This function releases a mask of TAU0 channel 0 count end interrupts and starts count operation.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_CSI00_Start

Synopsis	CSI00 operation start
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	void R_CSI00_Start(void)
Explanation	This function starts SAU0 channel 0 as CSI00 and sets it to a communication standby state.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_CSI00_Send_Receive

Synopsis	CSI00 data transmission/reception function
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	MD_STATUS R_CSI00_Send_Receive(uint8_t * const tx_buf, uint16_t tx_num, and uint8_t * const rx_buf)
Explanation	This function sets up CSI00 data transmission/reception.
Arguments	uint8_t * const tx_buf　　　　　　　　: [Transmit data buffer address] uint16_t tx_num　　　　　　　　　　　: [Transmit data buffer size] uint8_t * const rx_buf　　　　　　　　: [Receive data buffer address]
Return value	[MD_OK]: Transmission/reception setup completed [MD_ARGERROR]: Transmission/reception setup unsuccessful
Remarks	None

[Function Name] r_csi00_interrupt	
Synopsis	CSI00 transfer end interrupt function
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	static void __near r_csi00_interrupt(void)
Explanation	If there is data not transmitted, this function reads receive data and then starts transmitting the data not transmitted. Otherwise, this function reads receive data.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_csi00_callback_sendend	
Synopsis	CSI00 callback function upon transfer end
Header	r_cg_macrodriver.h, r_cg_serial.h, r_cg_userdefine.h
Declaration	static void __near r_csi00_interrupt(void)
Explanation	This function is executed upon CSI00 transfer end.
Arguments	None
Return value	None
Remarks	None

4.5.5   Flowcharts

4.5.5.1   Overall Flow

Figure 4.5 shows the overall flow of the sample program described in this application note.
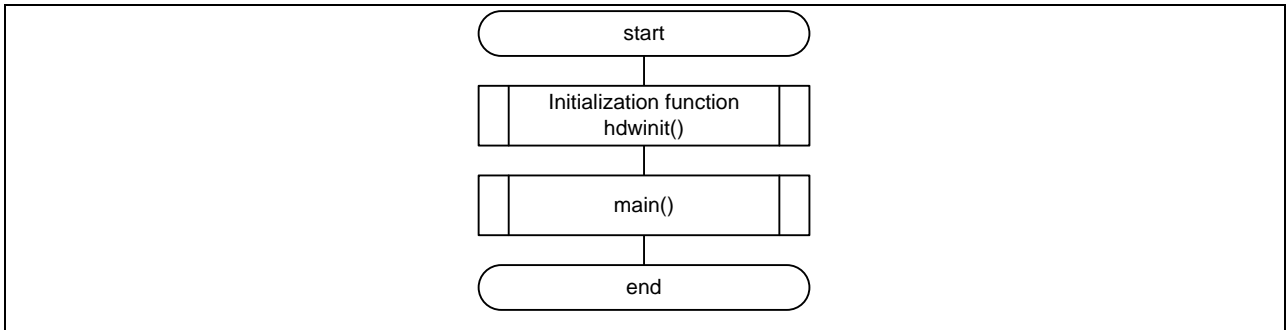
Figure 4.5 Overall Flow

4.5.5.2   Initialization Function

Figure 4.6 shows the flowchart for the initialization function.
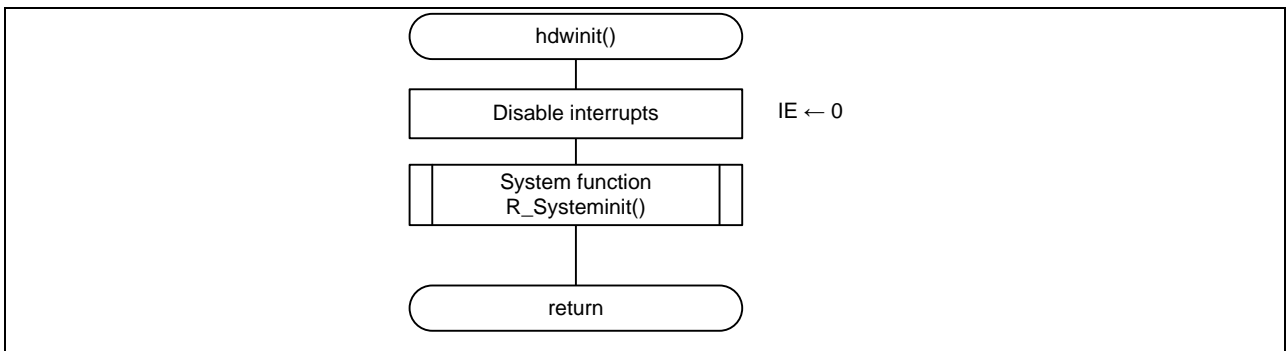
Figure 4.6 Initialization Function

4.5.5.3   System Function

Figure 4.7 shows the flowchart for the system function.
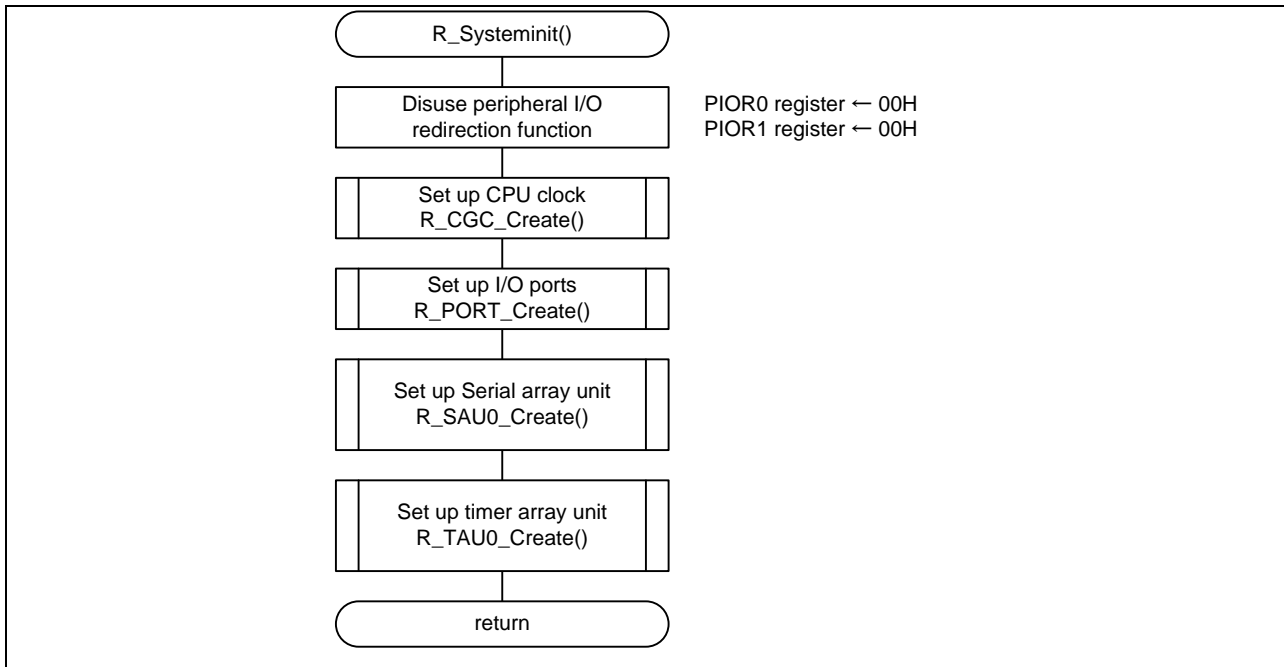
Figure 4.7   System Function

4.5.5.4   I/O Port Setup

Figure 4.8 shows the flowchart for I/O port setup.
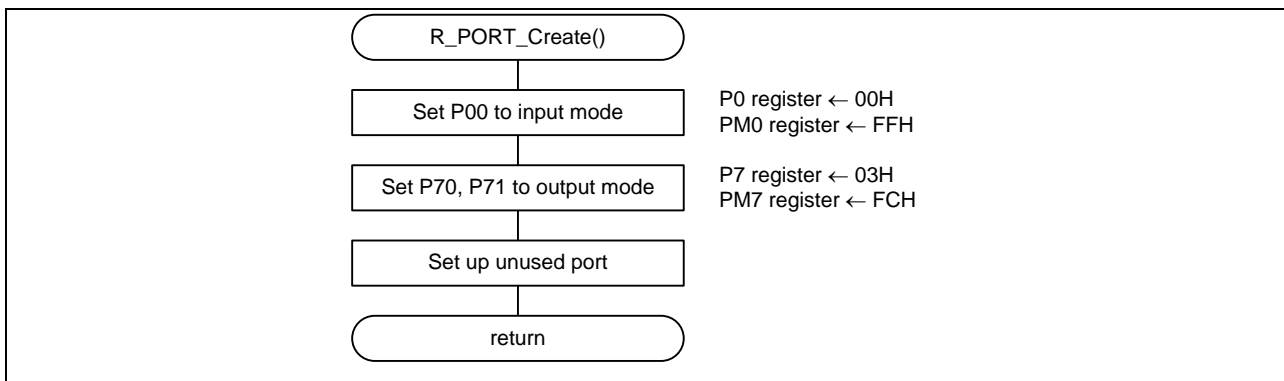
Figure 4.8   I/O Port Setup

Note: Refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN2575E) for the configuration of the unused ports.

Caution: Provide proper treatment for unused pins so that their electrical specifications are observed. Connect each of any unused input-only ports to V_{DD} or V_{SS} via a separate resistor.

Setting up the BUSY signal detection ports

- Port register 0 (P0)

- Port mode register 0 (PM0)
  Select an I/O mode and output latch for each port.

Symbol: P0

7	6	5	4	3	2	1	0
0	P06	P05	P04	P03	P02	P01	P00
0	x	x	x	x	x	x	0

Bit 0

P00	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Symbol: PM0

7	6	5	4	3	2	1	0
1	PM06	PM05	PM04	PM03	PM02	PM01	PM00
1	x	x	x	x	x	x	1

Bit 0

PM00	PM11 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Setting up the ports for _CS1 and _CS2 signals

- Port register 7 (P7)

- Port mode register 7 (PM7)
  Select an I/O mode and output latch for each port.

Symbol: P7

7	6	5	4	3	2	1	0
P77	P76	P75	P74	P73	P72	P71	P70
	x	x	x	x	x	**1**	**1**

Bit 1

P71	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	**Output 1**	**Input high level**

Bit 0

P70	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	**Output 1**	**Input high level**

Symbol: PM7

7	6	5	4	3	2	1	0
PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70
1	x	x	x	x	x	**0**	**0**

Bit 1

PM71	PM71 pin I/O mode selection
0	**Output mode (output buffer on)**
1	Input mode (output buffer off)

Bit 0

PM70	PM70 pin I/O mode selection
0	**Output mode (output buffer on)**
1	Input mode (output buffer off)

4.5.5.5   CPU Clock Setup

Figure 4.9 shows the flowchart for setting up the CPU clock.
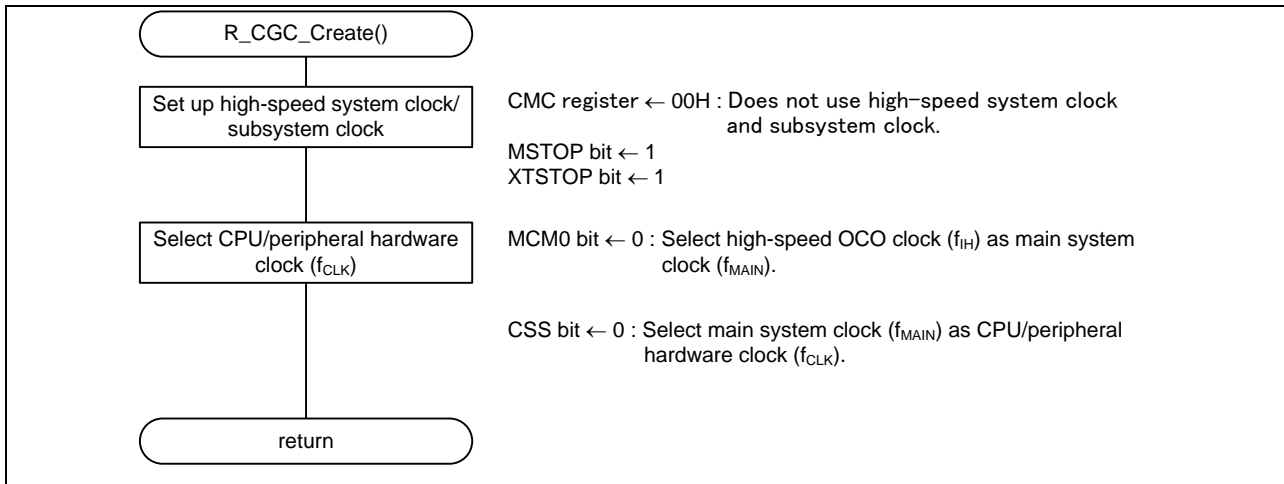
Figure 4.9 CPU Clock Setup

Caution: For details on the procedure for setting up the CPU clock (R_CGC_Create ()), refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN2575E).

4.5.5.6   SAU0 Setup

Figure 4.10 shows the flowchart for SAU0 setup.
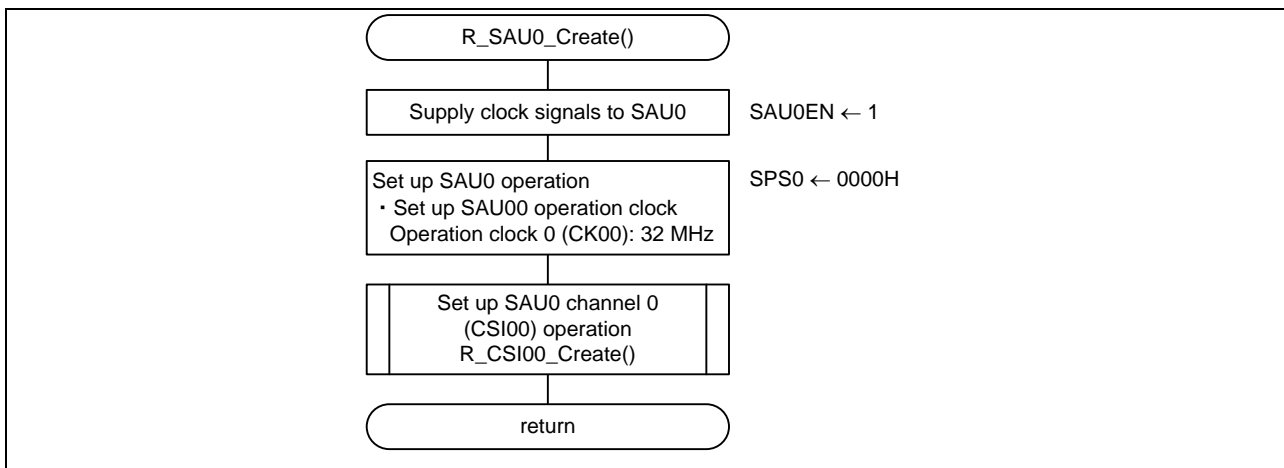
Figure 4.10 SAU0 Setup

Enabling supply of clock signals to the SAU

- Peripheral enable register 0 (PER0)
  Enable supply of clock signals to SAU0.

Symbol: PER0

7	6	5	4	3	2	1	0
RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
x	x	x	x	x	1	x	x

Bit 2

SAU0EN	Control of serial array unit 0 and input clock supply
0	Stops input clock supply.
1	Enables input clock supply.

Selecting a serial clock

- Serial clock select register 0 (SPS0)
  Select an operation clock for SAU0.

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Symbol: SPS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000
0	0	0	0	0	0	0	0	x	x	x	x	**0**	**0**	**0**	**0**

Bits 3 to 0

PRS 003	PRS 002	PRS 001	PRS 000		Selection of operation clock (CK00)				
					$f_{CLK} =$ 2 MHz	$f_{CLK} =$ 5 MHz	$f_{CLK} =$ 10 MHz	$f_{CLK} =$ 20 MHz	**$f_{CLK} =$ 32 MHz**
0	**0**	**0**	**0**	**f_{CLK}**	2 MHz	5 MHz	10 MHz	20 MHz	**32 MHz**
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1,25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	977 Hz

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

4.5.5.7   SAU0 Channel 0 (CSI00) Operation Setup

Figure 4.11 shows the flowchart for setting up SAU0 channel 0 (CSI00) operation.
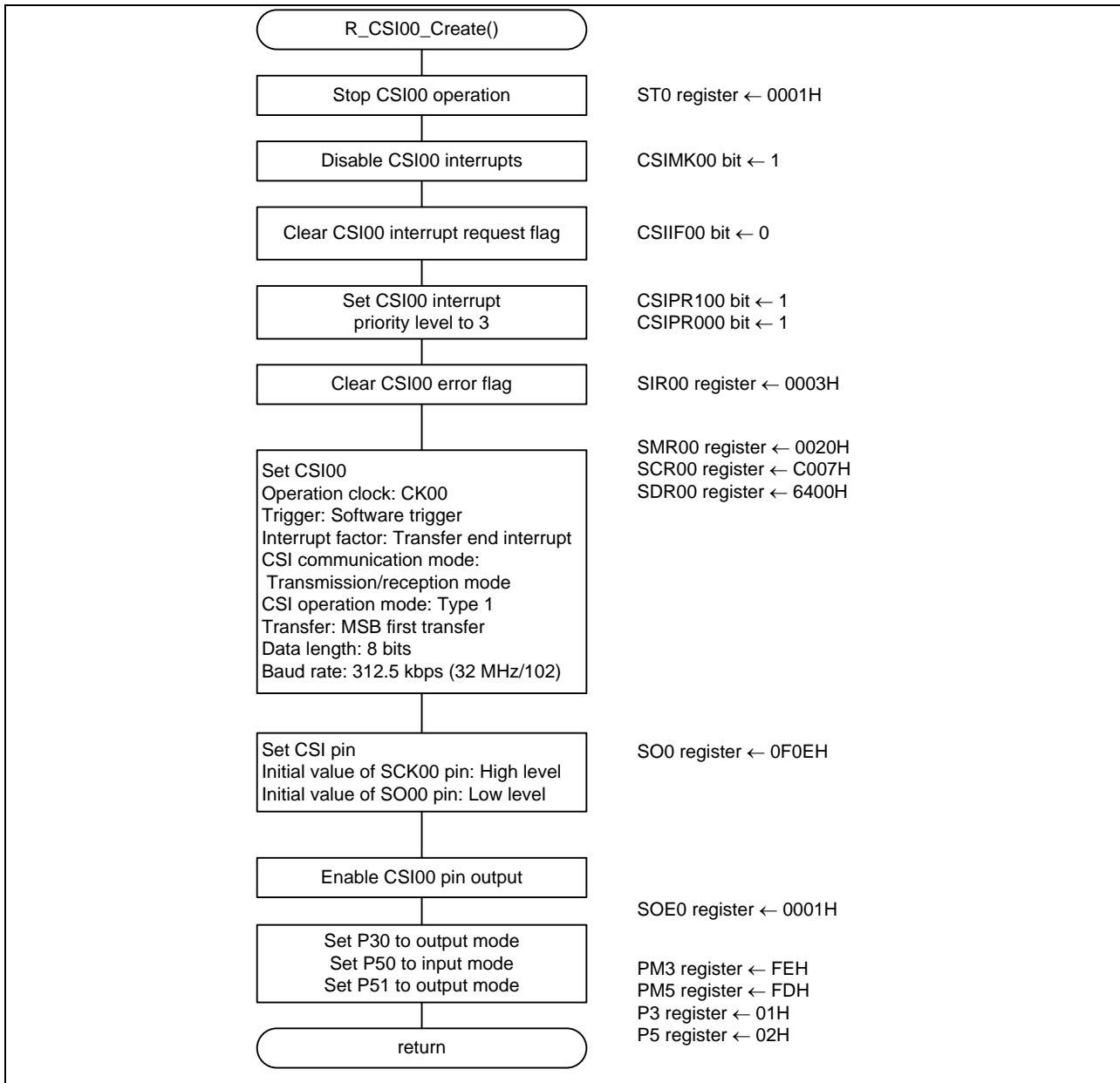

Figure 4.11   SAU0 Channel 0 (CSI00) Operation Setup

Stopping serial channel 0

- Serial channel stop register 0 (ST0)
  Stop communication/count operation of serial channel 0.

Symbol: ST0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	ST0 3	ST0 2	ST0 1	ST0 0
0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	1

Bit 0

ST00	Operation stop trigger of channel 0
0	No trigger operation
1	Clears the SE00 bit to 0 and stops the communication operation.

Setting a transfer end interrupt priority level

- Priority specification flag register 00H (PR00H)

- Priority specification flag register 10H (PR10H)
  Set the interrupt priority level.

Symbol: PR00H

7	6	5	4	3	2	1	0
SREPR00 TMPR001H	SRPR00 CSIPR001 IICPR001	STPR00 CSIPR000 IICPR000	1	1	SREPR02 TMPR011H	SRPR02 CSIPR021 IICPR021	STPR02 CSIPR020 IICPR020
x	x	1	x	x	x	x	x

Symbol: PR10H

7	6	5	4	3	2	1	0
SREPR10 TMPR101H	SRPR10 CSIPR101 IICPR101	STPR10 CSIPR100 IICPR100	1	1	SREPR12 TMPR111H	SRPR12 CSIPR121 IICPR121	STPR12 CSIPR120 IICPR120
x	x	1	x	x	x	x	x

Bit 5

CSIPR000	CSIPR100	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	1	Specify level 3 (low priority level)

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Clearing the CSI00 error flags

- Serial flag clear trigger register 00 (SIR00)
  Clear the SAU0 channel 0 error flags.

Symbol: SIR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	FEC T00	PEC T00	OVCT0 0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit 2

FECT00	Clear trigger of framing error flag of channel 0
0	Not cleared
1	**Clears the FEF00 bit of the SSR00 register to 0.**

Bit 1

PECT00	Clear trigger of parity error flag of channel 0
0	Not cleared
1	**Clears the PEF00 bit of the SSR00 register to 0.**

Bit 0

OVCT00	Clear trigger of overrun error flag of channel 0
0	Not cleared
1	**Clears the OVF00 bit of the SSR00 register to 0.**

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Setting up the SAU0 channel 0 operation mode

- Serial mode register 00 (SMR00)
  Select an operation clock (f_{MCK}).
  Specify whether to make the serial clock (f_{SCK}) input available.
  Set the start trigger and operation mode.
  Select an interrupt source.

Symbol: SMR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS 00	CCS 00	0	0	0	0	0	STS 00	0	SIS0 00	1	0	0	MD 002	MD 001	MD 000
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit 15

CKS00	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CK00 set by the SPS0 register
1	Operation clock CK01 set by the SPS0 register

Bit 14

CCS00	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKS00 bit
1	Clock input f_{SCK} from the SCK00 pin (slave transfer in CSI mode)

Bit 8

STS00	Selection of start trigger source
0	Only software trigger is valid
1	Valid edge of the RxDq pin (selected for UART reception)

Bits 2 and 1

MD002	MD001	Setting of operation mode of channel 0
0	0	CSI mode
0	1	UART mode
1	0	Simplified I²C mode
1	1	Setting prohibited

Bit 0

MD000	Selection of interrupt source of channel 0
0	Transfer end interrupt
1	Buffer empty interrupt

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Setting up the SAU0 channel 0 operation mode

- Serial communication operation setting register 00 (SCR00)
  Select an operation clock (fMCK).
  Specify whether to make the serial clock (fSCK) input available.
  Set up the start trigger and operation mode.
  Select an interrupt source.

Symbol: SCR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE 00	RXE 00	DAP 00	CKP 00	0	EOC 00	PTC 001	PTC 000	DIR 00	0	SLC 001	SLC 000	0	1	DLS 001	DLS 000
1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bits 15 and 14

TXE00	RXE00	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	**1**	**Transmission/reception**

Bits 13 and 12

DAP00	CKP00	Selection of data and clock phase in CSI mode	Type
0	**0**		1
0	1		2
1	0		3
1	1		4

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Symbol: SCR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE 00	RXE 00	DAP 00	CKP 00	0	EOC 00	PTC 001	PTC 000	DIR 00	0	SLC 001	SLC 000	0	1	DLS 001	DLS 000
1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit 7

DIR00	Selection of data transfer sequence in CSI and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.

Bits 1 and 0

DLS001	DLS000	Setting of data length in CSI and UART modes
0	0	9-bit data length (stored in bits 0 to 8 of the SDR00 register) (can be set in UART0 mode only.)
1	0	7-bit data length (stored in bits 0 to 6 of the SDR00 register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDR00 register)
Other than above		Setting prohibited

Selecting an operation clock frequency divisor

- Serial data register 00 (SDR00)
  Set the division ratio of the operation clock (f_{MCK}) frequency.

Symbol: SDR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 9

SDR00[15:9]							Transfer clock setting by dividing the operation clock (f_{MCK})
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
0	1	1	0	0	1	0	$f_{MCK}/102$
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Specifying the output values for the SCK00 and SO00 pins

- Serial output register 0 (SO0)
  Specify the output values for the serial data output pin and serial clock output pin.

Symbol: SO0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	CK O03	CK O02	CK O01	CK O00	0	0	0	0	SO 03	SO 02	SO 01	SO 00
0	0	0	0	x	x	x	1	0	0	0	0	x	x	x	0

Bit 8

CKO00	Serial clock output of channel 0
0	Serial clock output value is "0".
1	Serial clock output value is "1".

Bit 0

SO00	Serial data output of channel 0
0	Serial clock output value is "0".
1	Serial clock output value is "1".

Enabling output of serial communication operation

- Serial output enable register 0 (SOE0)
  Enable output of serial communication operation.

Symbol: SOE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SOE 03	SOE 02	SOE 01	SOE 00
0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	1

Bit 0

SOE00	Serial output enable/stop of channel 0
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Setting up the ports of the SCK00, SO00 and SI00 pins

- Port register 3 (P3)

- Port mode register 3 (PM3)

- Port register 5 (P5)

- Port mode register 5 (PM5)
  Select an input/output mode and output latch for each port.

Symbol: P3

7	6	5	4	3	2	1	0
P37	P36	P35	P34	P33	P32	P31	P30
x	x	x	x	x	x	x	1

Bit 0

P30	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Symbol: PM3

7	6	5	4	3	2	1	0
PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30
x	x	x	x	x	x	x	0

Bit 0

PM30	P10 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Symbol: P5

7	6	5	4	3	2	1	0
P57	P56	P55	P54	P53	P52	P51	P50
x	x	x	x	x	x	1	0

Bit 1

P51	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Bit 0

P50	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Symbol: PM5

7	6	5	4	3	2	1	0
PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50
x	x	x	x	x	x	0	1

Bit 1

PM51	P51 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Bit 0

PM50	P50 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

4.5.5.8   TAU0 Setup
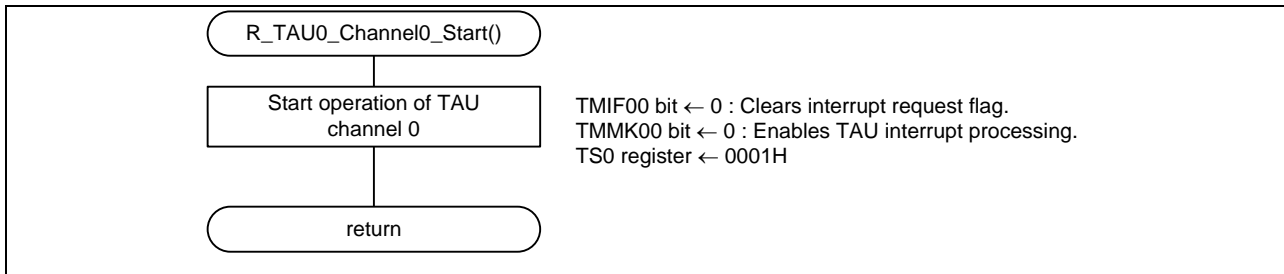
Figure 4.12 shows the flowchart for setting up TAU0.
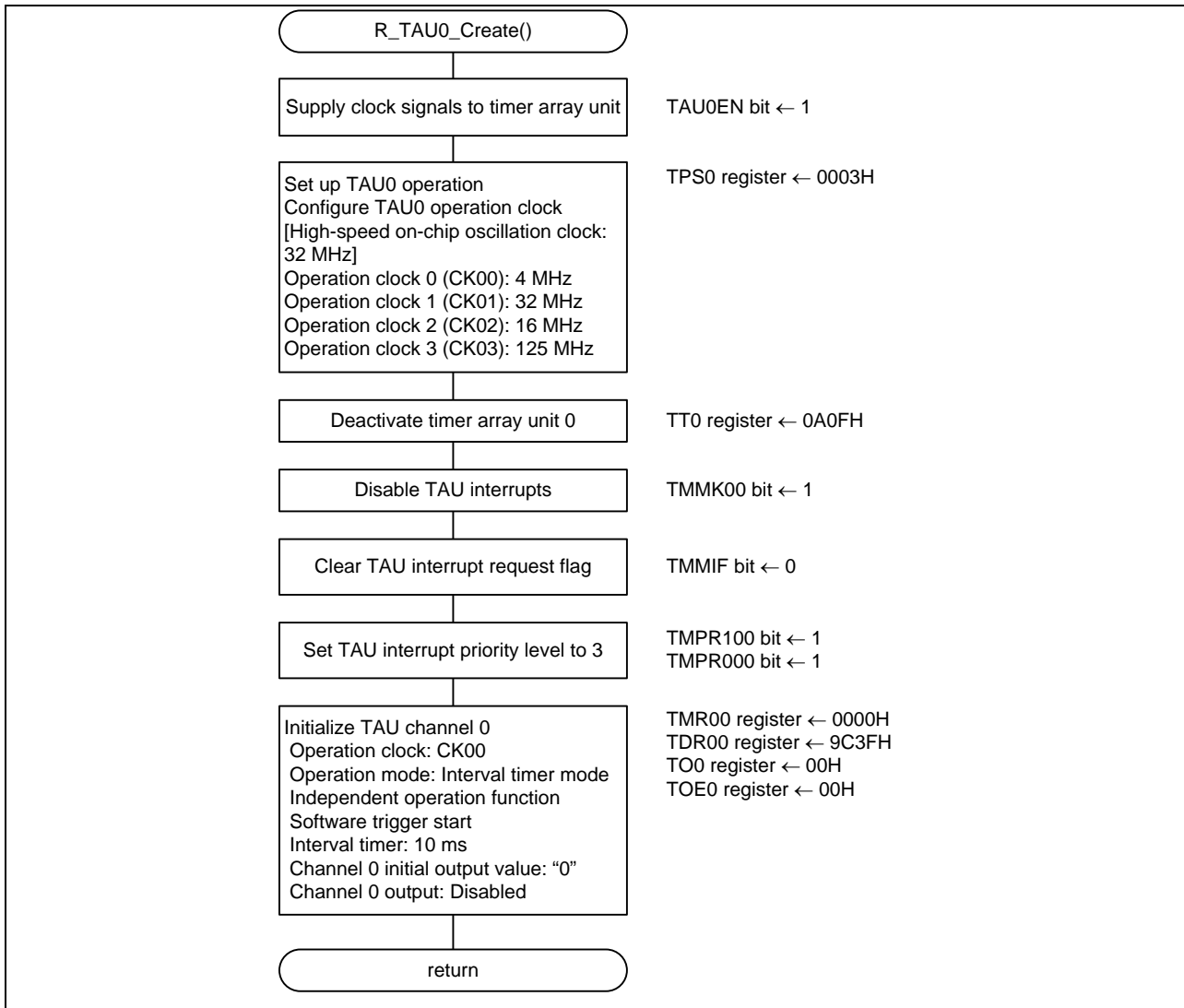
Figure 4.12 TAU0 Setup

Caution: For information about TAU0 setup (R_TAU0_Create()), refer to the section entitled "Flowcharts" in
RL78/G13 Timer Array Unit (Interval Timer) Application Note (R01AN2576E).

4.5.5.9   Main Processing

Figure 4.13 shows the flowchart for main processing.
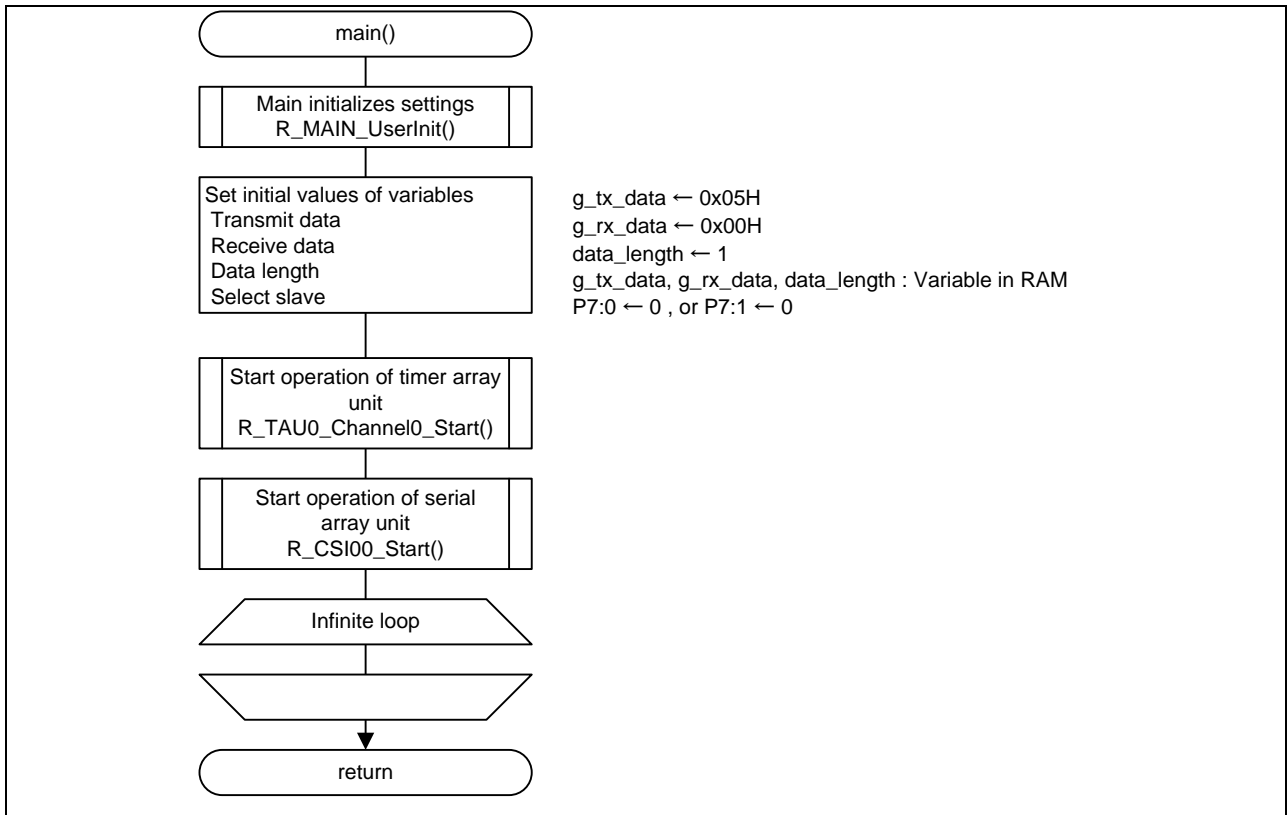
Figure 4.13 Main Processing

4.5.5.10   Main initializes setting

Figure 4.14 shows the flowchart for the main initializes settings.
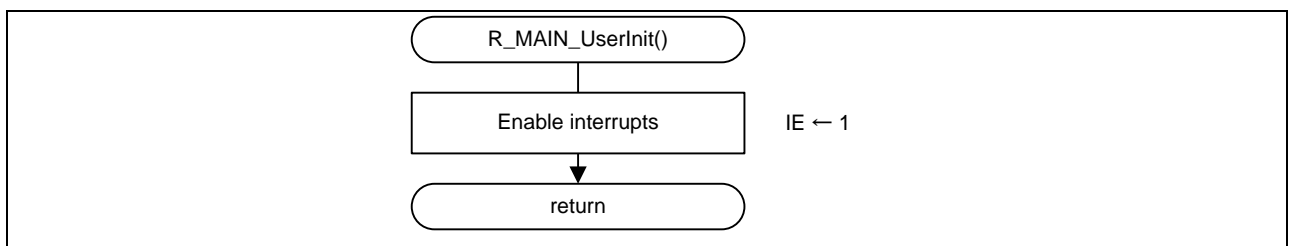
Figure 4.14 Main initializes setting

4.5.5.11   TAU0 Channel 0 Startup

Figure 4.15 shows the flowchart for starting the operation of TAU0 channel 0.

Figure 4.15 TAU0 Channel 0 Startup

Caution: For information about TAU0 setup (R_TAU0_Create()), refer to the section entitled "Flowcharts" in RL78/G13
         Timer Array Unit Interval Timer Application Note (R01AN2576E).

4.5.5.12   SAU0 Channel 0 Startup

Figure 4.16 shows the flowchart for starting the operation of SAU0 channel 0 (CSI00).
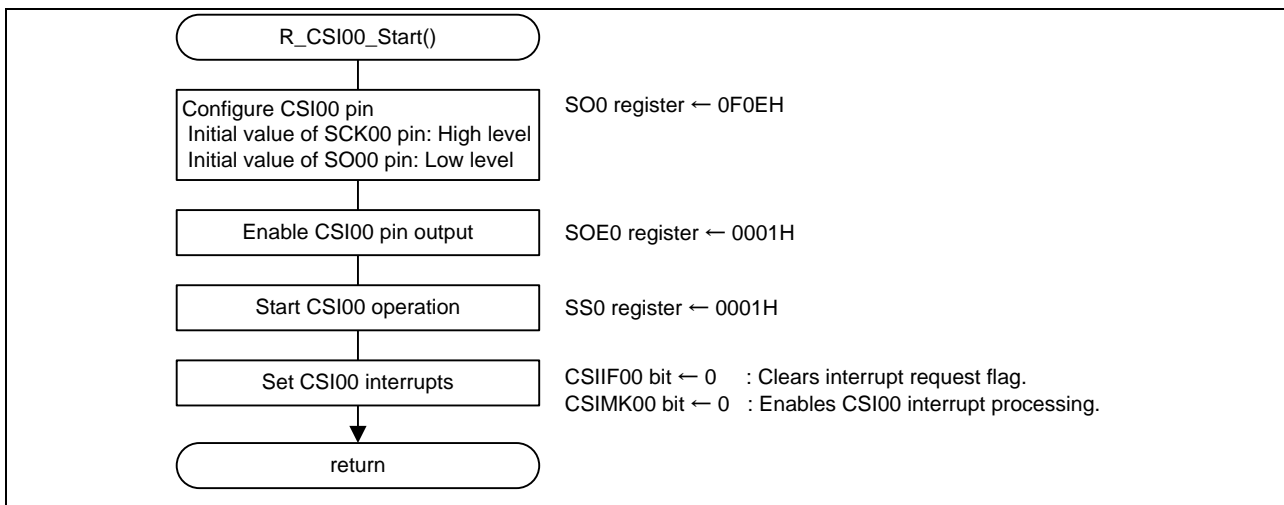
Figure 4.16 SAU0 Channel 0 (CSI00) Startup

Setting the transfer end interrupt

- Interrupt request flag register 0H (IF0H)
  Clear the interrupt request flag.
- Interrupt mask flag register 0H (MK0H)
  Enable interrupt processing.

Symbol: IF0H

7	6	5	4	3	2	1	0
SREIF0 TMIF01H	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	0	0	SREIF2 TMIF11H	SRIF2 CSIIF21 IICIF21	STIF2 CSIIF20 IICIF20
x	x	**0**	x	x	x	x	x

Bit 5

CSIIF00	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Symbol: MK0H

7	6	5	4	3	2	1	0
SREMK0 TMMK01H	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	1	1	SREMK2 TMMK11H	SRMK2 CSIMK21 IICMK21	STMK2 CSIMK20 IICMK20
x	x	**1**	x	x	x	x	x

Bit 5

CSIMK00	Interrupt processing control
0	Enables interrupt processing.
1	Disables interrupt processing.

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Enabling serial communication

- Serial channel start register 0 (SS0)
  Enable serial communication/count operation.

Symbol: SS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00
0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	1

Bit 0

SS00	Operation start trigger of channel 0
0	No trigger operation
1	Sets the SE00 bit to 1 and enters the communication wait status.

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Remark: When the SS0 register is read, 0000H is always read.

4.5.5.13   Infinite Loop in Main Processing

Figure 4.17 shows the flowchart for an infinite loop in the main processing.
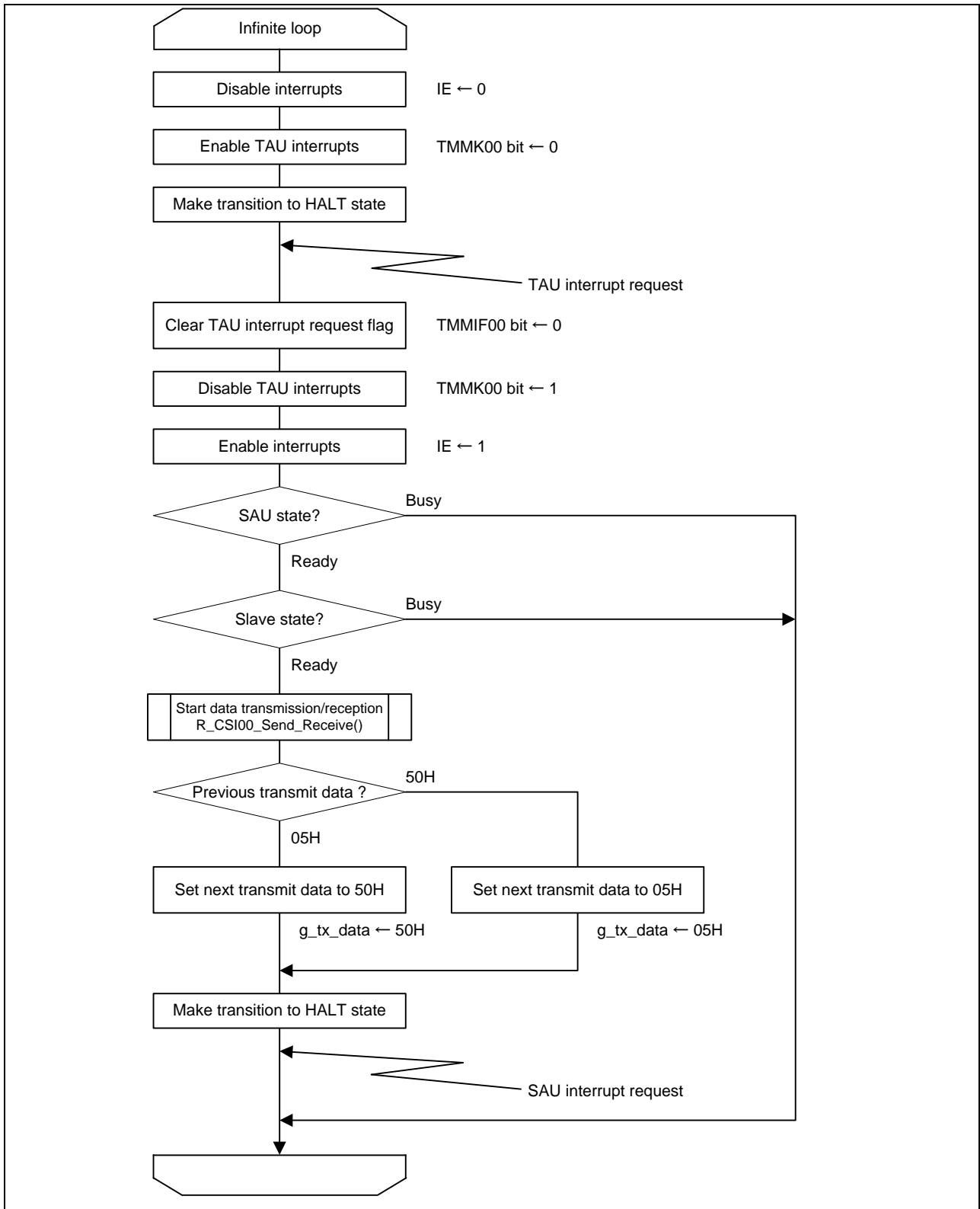
Figure 4.17 Infinite Loop in Main Processing

Confirming the communication state

- Serial status register 00 (SSR00)
  Indicate the communication status and error occurrence status of serial array unit channel 0.

Symbol: SSR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	TSF 00	BFF 00	0	0	FEF 00	PEF 00	OV F00
0	0	0	0	0	0	0	0	0	0/1	x	0	0	x	x	x

Bit 6

TSF00	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

4.5.5.14 CSI00 Data Transmission/Reception Start

Figure 4.18 shows the flowchart for starting CSI00 data transmission/reception.
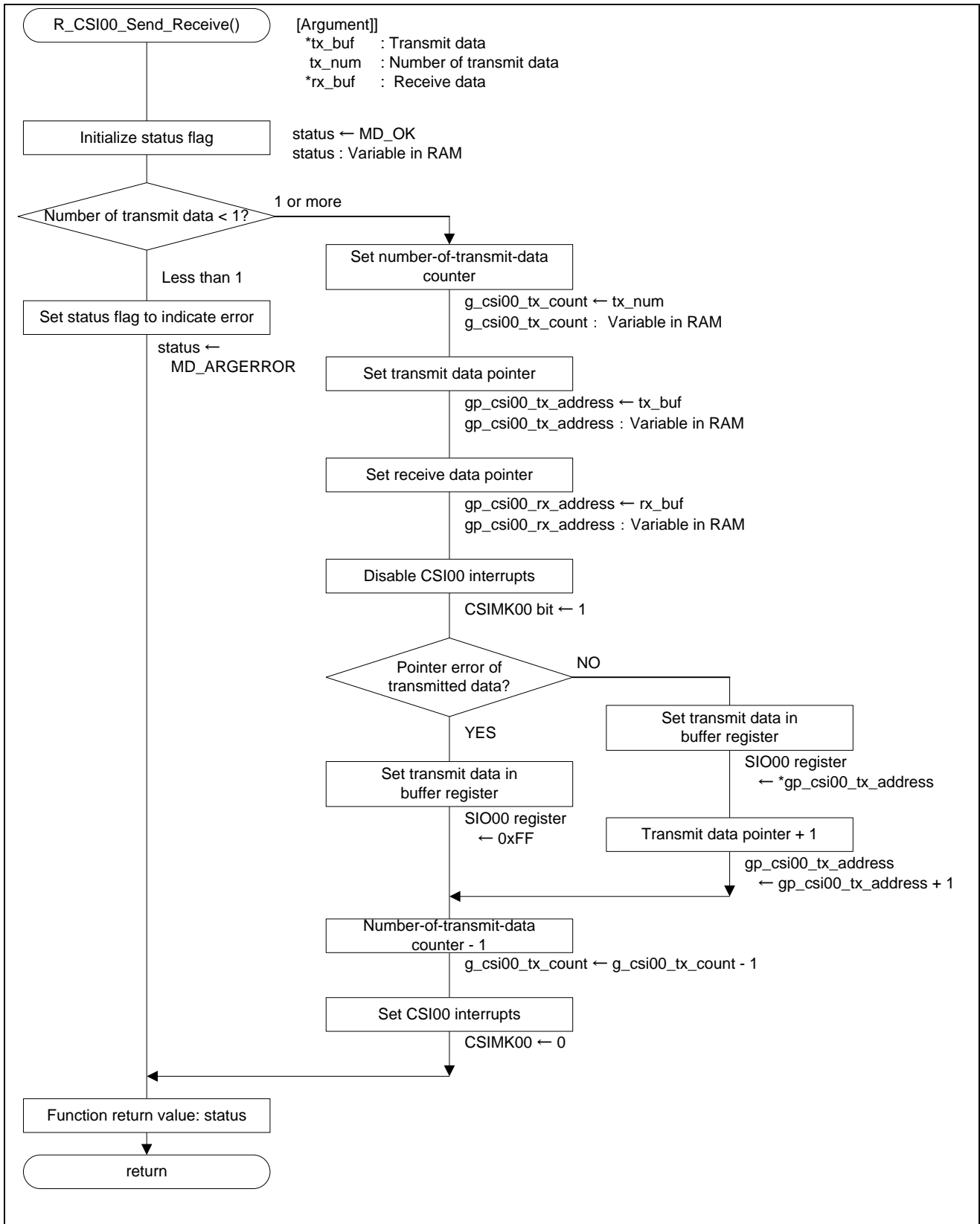
Figure 4.18   CSI00 Data Transmission/Reception Start

Setting transmit data

- Serial data register 00 (SDR00)
  Set transmit data and start transmitting the data.

Symbol: SDR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CSI00 data register (SIO00)
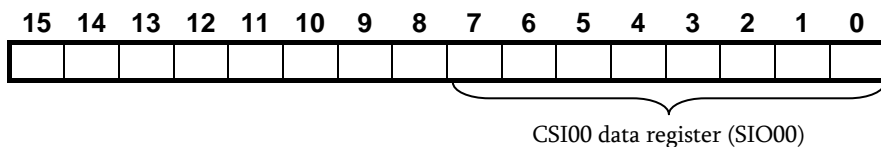
Write transmit data to the lower eight bits.

These eight bits should be accessed as the CSI00 register.

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

4.5.5.15   CSI00 Transfer End Interrupt Processing

Figure 4.19 and Figure 4.20 shows the flowchart for CSI00 transfer end interrupt processing.
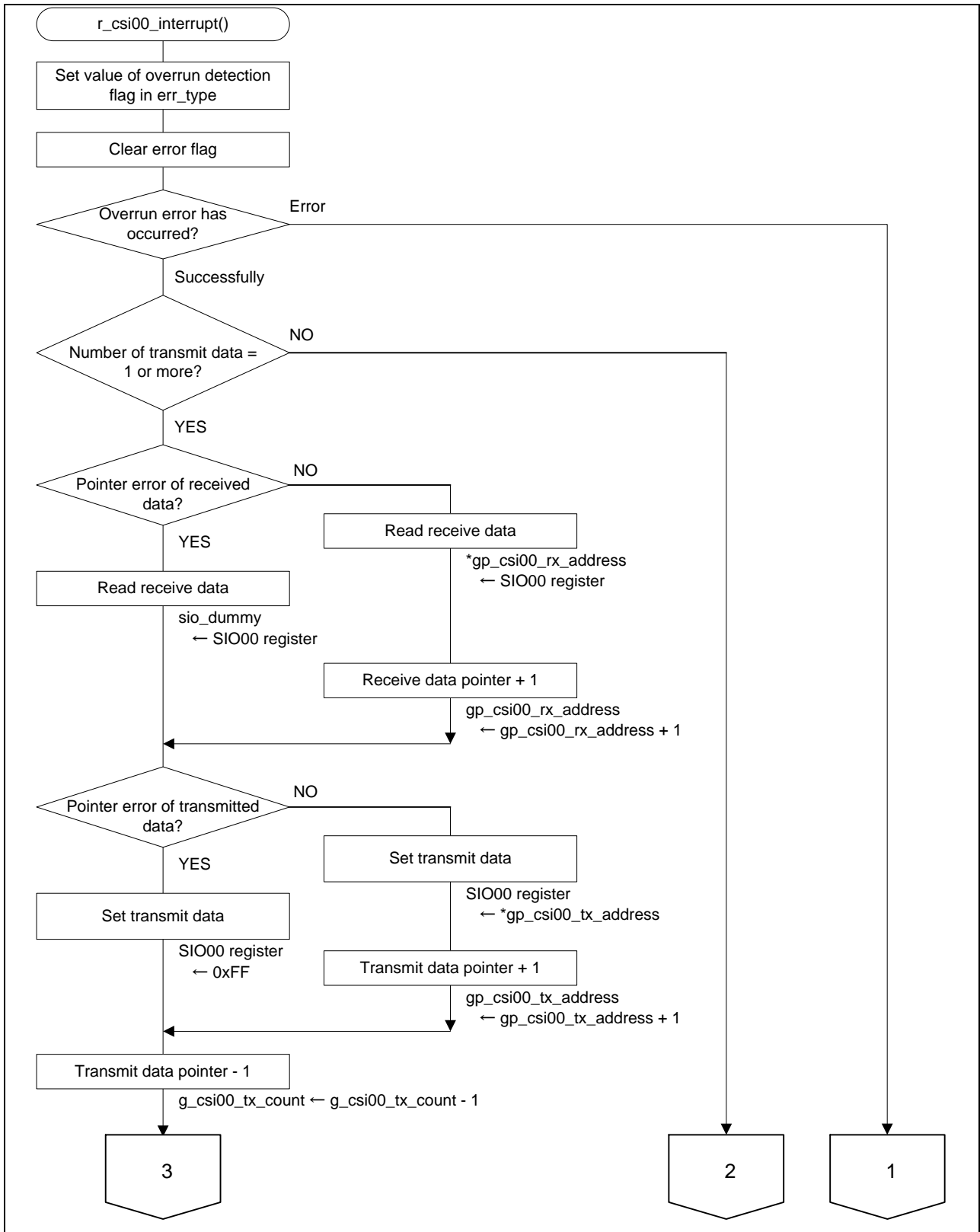
Figure 4.19   CSI00 Transfer End Interrupt Processing (1/2)
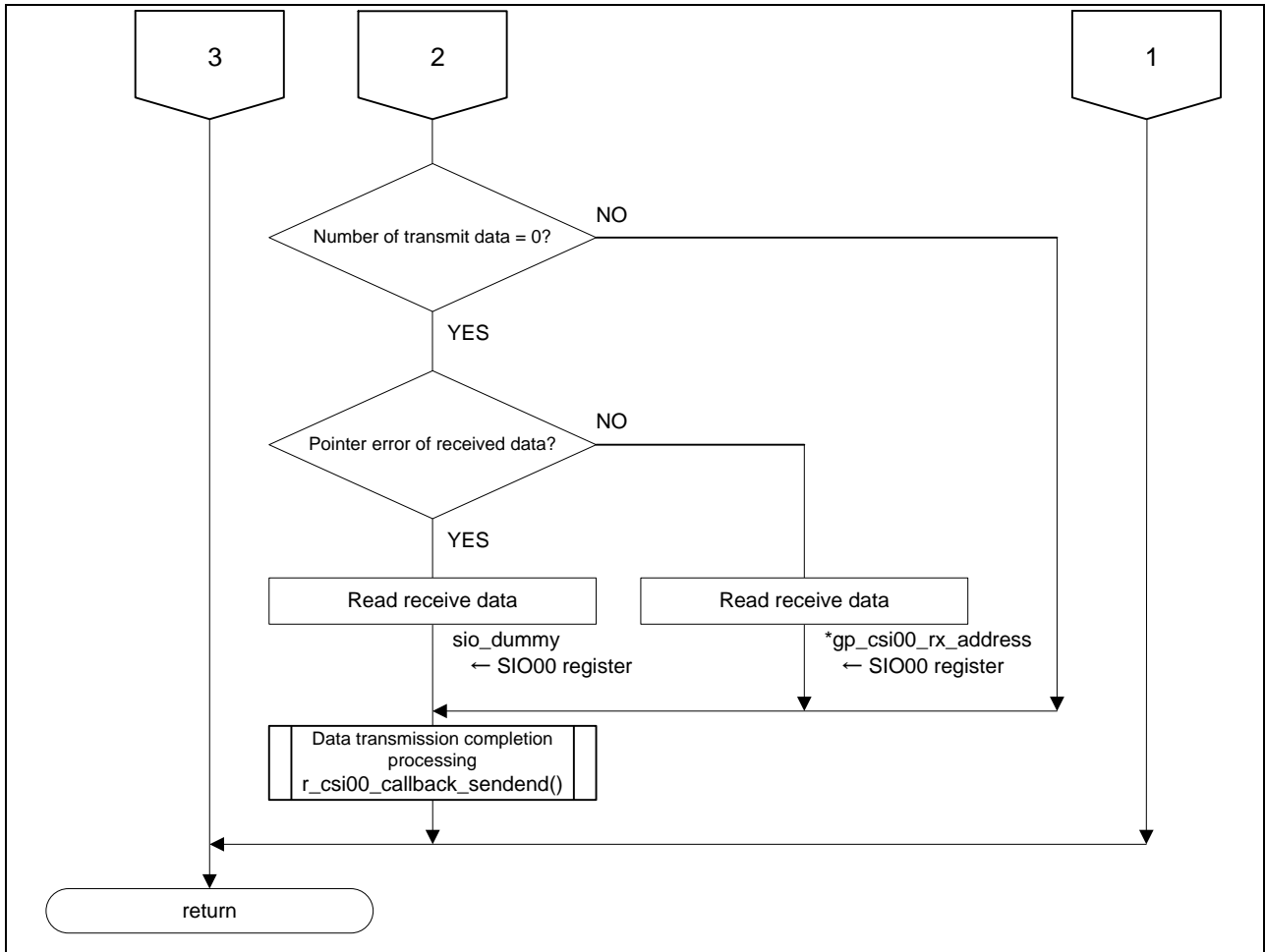
Figure 4.20   CSI00 Transfer End Interrupt Processing (2/2)

4.6   Sample Code

The sample code is available on the Renesas Electronics Website.

4.7   Related Application Note

The application note that is related to this application note is listed below for reference.

- RL78/G13 Initialization (R01AN2575E) Application Note
- RL78/G13 Timer Array Unit Interval Timer (R01AN2576E) Application Note
- RL78/G13 Serial Array Unit for 3-Wire Serial I/O (Master Transmission/Reception) (R01AN2547E) Application Note
- RL78/G13 Serial Array Unit for 3-Wire Serial I/O (SPI Master Transmission Reception) (R01AN2703E) Application Note

4.8   Documents for Reference

User's Manual:

RL78/G14 User's Manual: Hardware (R01UH0186)

R8C/36M Group User's Manual: Hardware (R01UH0259)

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

Technical Update/Technical News:

The latest information can be downloaded from the Renesas Electronics website.

Migration Guide

Migration to CubeSuite+ Integrated Development Environment for RL78 Family
(On-chip Debug) - Migration from R8C, M16C to RL78 (R20UT2150)

5.   Example of Migration from 4-Wire Bus Communication Mode (Slave transmission/reception)

5.1   Specifications

To support the communication operations (slave transmission/reception) in 4-wire bus communication mode of the SSU of the R8C/36M group by using the RL78/G14, 3-wire serial I/O communication (CSI) of the serial array unit (SAU) is used.

The RL78/G14, operated as the slave, transmits and receives data synchronously with the clock signal from the master. The slave select function is implemented by using the slave select input function.

For data transmission/reception, it is also necessary to check in advance if the corresponding device is ready for communication (handshake operation). Therefore, the handshake signal (_BUSY signal) has been added to the serial communication signals. The slave device notifies the master by using the _BUSY signal that it is ready for communication.

Table 5.1 lists the peripheral functions to be used and their uses. Figure 5.1 presents an overview of CSI operation.

Figure 5.2 to Figure 5.3 show timing charts for explaining the CSI communication.

Table 5.1 Peripheral Functions to be Used and Their Uses

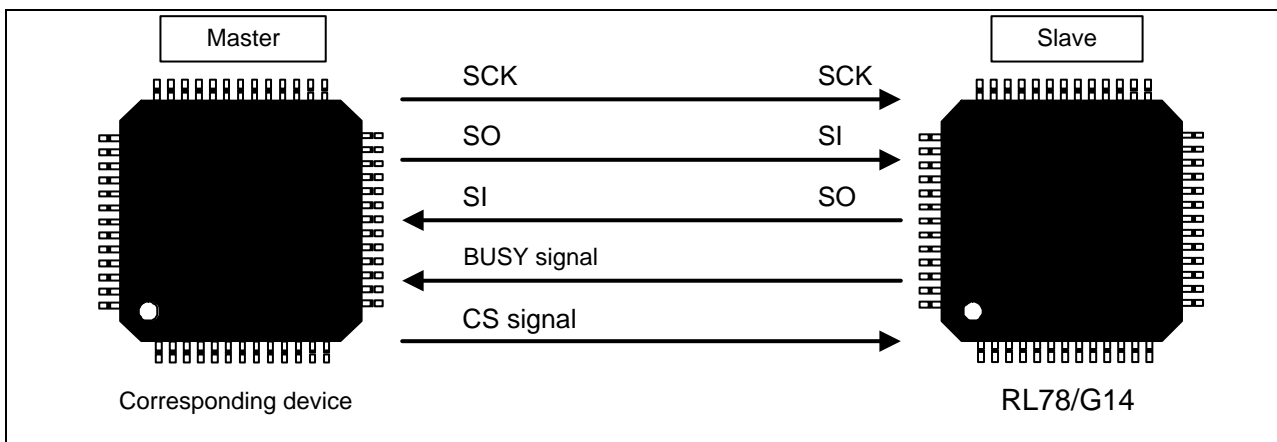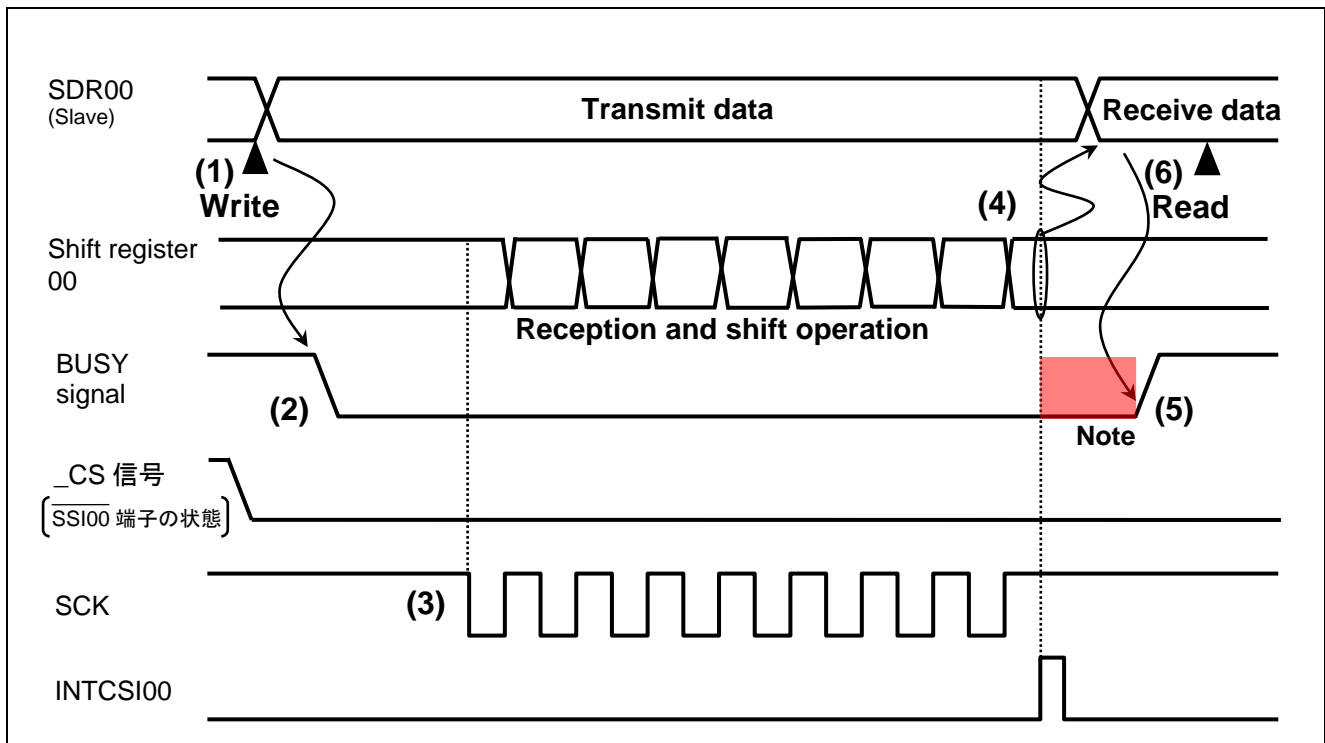| Peripheral Function	Use
Serial array unit 0 channel 0	CSI00 slave transmission/reception

Figure 5.1 Overview of CSI Operation

Figure 5.2 Handshake Operation and Communication

(1) [Software processing] Write the transmit data (slave to master) to the SDR00 register.

(2) [Software processing] Make the BUSY signal fall to notify the master that communication is possible.

(3) [Hardware processing] Input serial clock from the master and then enter the communication state.

(4) [Hardware processing] Transfer receive data from the shift register 00 to the SDR00 and then generate a transfer
    end interrupt.

(5) [Software processing] Raise the BUSY signal to notify the master that transfer is impossible.

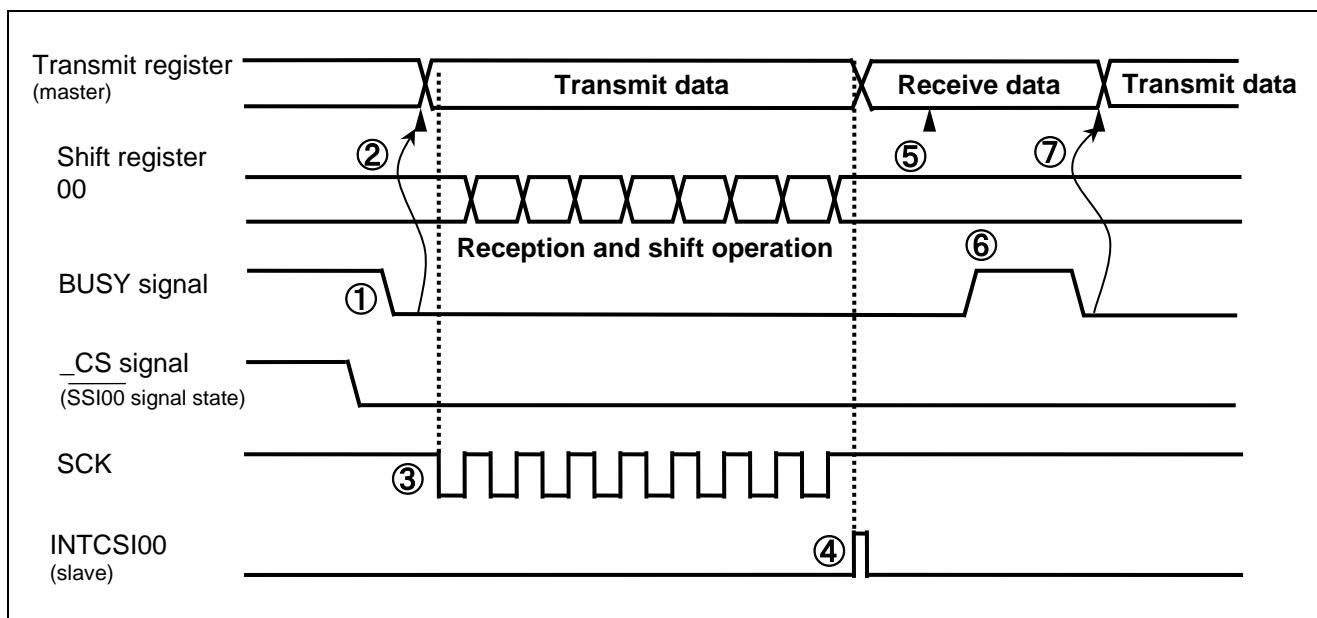(6) [Software processing] Read receive data from the SDR00 register.

Figure 5.3 Example of BUSY Signal Edge Detection in the Master

・BUSY signal edge detection

In this exapmle, the master starts communication upon detection of the falling edge of the BUSY signal from the slave.

(1) [Software processing in the slave]

Write the next transmit data (slave to master) and assert.

(2) [Software processing in the master]

Detect the falling edge of the BUSY signal and write the transmit data to the transmit register.

(3) [Hardware processing in the master]

Start transmission/reception and then output serial clock (SCK) signals.

(4) [Hardware processing in the slave]

After completion of the transfer, set the value of shift register 00 in the SDR00 register and then generate a transfer end interrupt (INTCSI00).

(5) [Software processing in the master]

Read the receive data and wait until the falling edge of the BUSY signal is detected. Note

(6) [Software processing in the slave]

Negate and read the receive data from the SDR00 register.

Write the next transmit data to the SDR00 register and assert.

(7) [Software processing in the master]

Detect the falling edge of the BUSY signal and then write the transmit data to the transmit register.

Note: If the BUSY signal is held at the high level for a short period, the software may be unable to detect the edge. In this case, input the BUSY signal to an external interrupt pin (such as the INTP0 pin) so that the hardware detects the edge.

5.2   Operation Check Conditions

The sample code contained in this application note has been checked under the conditions listed in the table below.

Table 5.2 Operation Check Conditions

Item	Description
Microcontroller used	RL78/G14 (R5F104LEA)
Operating frequency	High-speed on-chip oscillator (HOCO) clock: 32 MHz CPU/peripheral hardware clock: 32 MHz
Operating voltage	5.0 V (Operation is possible over a voltage range of 2.9 V to 5.5 V) LVD operation (V_{LVD}): Reset mode which uses 2.81 V (2.76 V to 2.87 V)
Integrated development environment (CS+)	CS+ for CC V5.00.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.04.00 from Renesas Electronics Corp.
Integrated development environment (e2 studio)	e2 studio V5.4.0.018 from Renesas Electronics Corp.
C compiler (e2 studio)	CC-RL V1.04.00 from Renesas Electronics Corp.

5.3　　Description of the Hardware

5.3.1　　Hardware Configuration Example

Figure 5.4 shows an example of hardware configuration that is used for this application note.
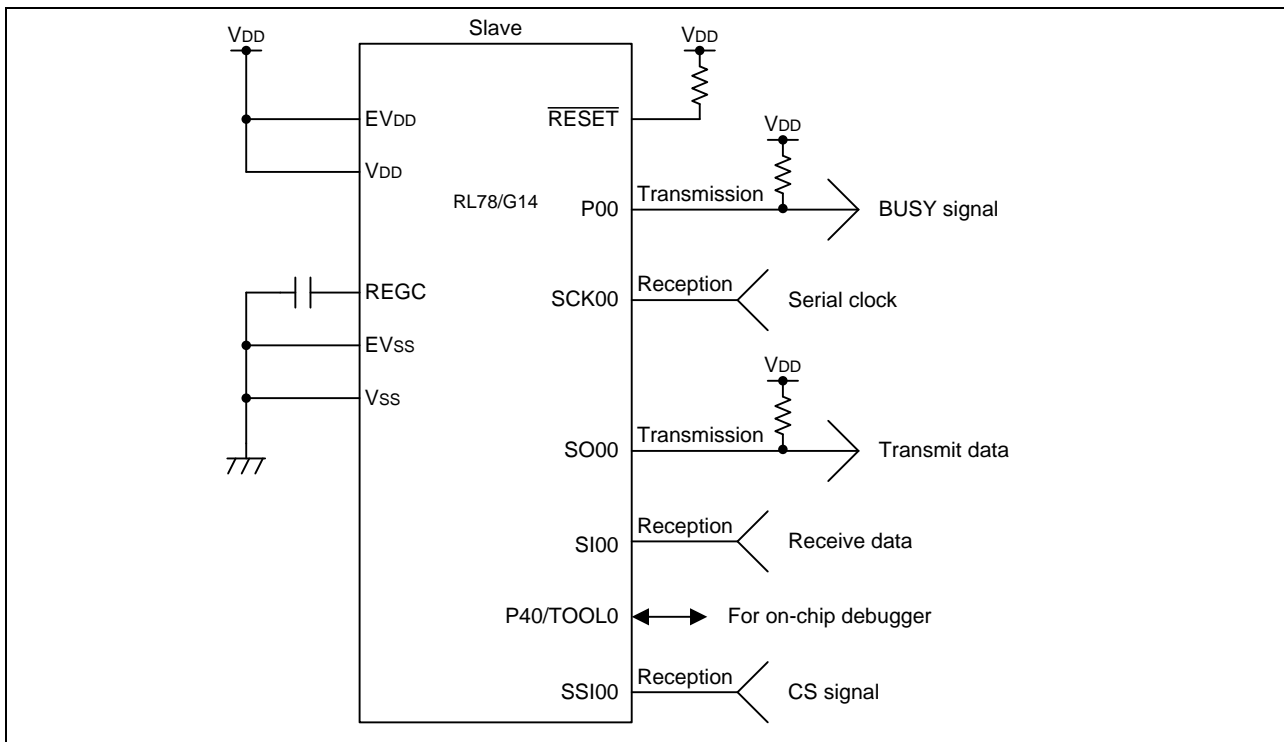

Figure 5.4 Hardware Configuration

Cautions:　1.　The purpose of this circuit is only to provide the connection outline and the circuit is simplified accordingly. When designing and implementing an actual circuit, provide proper pin treatment and make sure that the hardware's electrical specifications are met (connect the input-only ports separately to V_{DD} or V_{SS} via a resistor).

　　　　　　2.　Connect any pins whose name begins with EV_{SS} to V_{SS} and any pins whose name begins with EV_{DD} to V_{DD}, respectively.

　　　　　　3.　V_{DD} must be held at not lower than the reset release voltage (V_{LVD}) that is specified as LVD.

　　　　　　4.　Set the SO00 pin and BUSY signal to N-ch open-drain output. When more than one slave is connected, the SO00 pin and BUSY signal need to be set to N-ch open-drain output and pulled up.

5.3.2　　List of Pins to be Used

Table 5.3 lists the pins to be used and their functions.

Table 5.3 Pins to be Used and Their Functions

Pin Name	I/O	Description
P30/INTP3/RTC1HZ/SCK00/SCL00/TRJO0	Input	Serial clock input pin
P50/INTP1/SI00/RxD0/TOOLRxD/SDA00/TRGIOA	Input	Data reception pin
P51/INTP2/SO00/TxD0/TOOLTxD/TRGIOB	Output	Data transmission pin
P00/TI00/TRGCLKA	Output	BUSY signal pin
P62/SSI00	Input	Chip select input pin

5.4 Description of the Software

5.4.1 Operation Outline

The sample program covered in this application note transmits and receives data to and from the corresponding device (master) via the CSI (slave transmission/reception).

(1) Initialize SAU0.

<Conditions for setting>

- Use SAU0 channel 0 as the CSI.

- Select the single transfer mode as the operation mode.

- Select type 1 as the phase between data and clock signals.

- Set data transfer order to the MSB first.

- The length of data should be 8 bits.

- A serial transfer end interrupt (INTCSI00) should occur in single transfer mode.

- Use the P30/SCK00 pin for clock input.

- Use the P51/SO00 pin for data output.

- Use the P50/SI00 pin for data input.

- Enable output for serial communication.

(2) Write transmit data (slave → master) to the SDR00 register.

(3) Make sure that the SSI pin is at the low level.

(4) Set the BUSY signal to low level to notify the master that communication is possible.

(5) Execute a HALT instruction to enter HALT mode and then wait for the occurrence of a transfer end interrupt (INTCSI00).

(6) Update the receive data when a transfer end interrupt (INTCSI00) occurs. Then, output a BUSY state to the BUSY signal pin to cancel HALT mode.

(7) Repeats steps (3) to (6).

Note: The maximum transfer rate of the RL78/G14 is $f_{MCK}/6$ during slave communication.

5.4.2　　List of Option Byte Settings

Table 5.4 summarizes the settings of the option bytes.

Table 5.4 Option Byte Settings

Address	Value	Description
000C0H/010C0H	01101110B	Disables the watchdog timer. (Stops counting after the release from the reset state.)
000C1H/010C1H	01111111B	LVD reset mode, 2.81 V (2.76 V to 2.87 V)
000C2H/010C2H	11101000B	HS mode, HOCO: 32 MHz
000C3H/010C3H	10000100B	Enables the on-chip debugger.

5.4.3　　List of Constants

Table 5.5 lists the constants that are used in this sample program.

Table 5.5 Constants for the Sample Program

Constant	Setting	Description
_0001_SAU_CH0_DATA_OUTPUT_1	0x0001U	Serial data output setting for SAU0 channel 0
_0001_SAU_CH0_OUTPUT_ENABLE	0x0001U	Setting for enabling output through serial communication on SAU0 channel 0
_0001_SAU_CH0_START_TRG_ON	0x0001U	Setting for starting operation of SAU0 channel 0
_0001_SAU_OVERRUN_ERROR	0x0001U	Overrun error occurrence (SSR)
MD_STATUSBASE	0x00U	Communication status base value
MD_OK	MD_STATUSBASE+ 0x00U	Successful completion
MD_ERRORBASE	0x80U	Communication error status base value
MD_ARGERROR	MD_ERRORBASE+ 01U	Parameter error

5.4.4　　List of Variables

Table 5.6 list the global variables that are used in this sample program.

Table 5.6 Global Variables

Type	Variable Name	Contents	Function Used
unsigned char	g_tx_data	Serial transmit data	main()
unsigned char	g_rx_data	Serial receive data	main()
uint8_t	gp_csi00_rx_address	CSI00 receive buffer address	R_CSI00_Send_Receive() r_csi00_interrupt()
uint8_t	gp_csi00_tx_address	CSI00 transmit buffer address	R_CSI00_Send_Receive() r_csi00_interrupt()
uint16_t	g_csi00_tx_count	CSI00 transmit data size	R_CSI00_Send_Receive() r_csi00_interrupt()

5.4.5   List of Functions

Table 5.7 lists the global variable that is used by this sample program.

Table 5.7 Functions

Function Name	Outline
R_CSI00_Start	CSI00 operation start processing
R_CSI00_Send_Receive	CSI00 data transmission/reception function
r_csi00_interrupt	CSI00 transfer end interrupt function
r_csi00_callback_receiveend	CSI00 data reception completion processing

5.4.6   Function Specifications

This section describes the specifications for the functions that are used in the sample code.

[Function Name] R_CSI00_Start	
Synopsis	CSI00 operation start
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	void R_CSI00_Start(void)
Explanation	This function starts SAU0 channel 0 as CSI00 and sets it to a communication standby state.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_CSI00_Send_Receive	
Synopsis	CSI00 data transmission/reception function
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	MD_STATUS R_CSI00_Send_Receive(uint8_t *tx_buf, uint16_t tx_num, uint8_t *rx_buf)
Explanation	This function assigns the transmit/receive data buffer addresses and sizes that are specified by the arguments to global variables and then starts data transmission.
Arguments	uint8_t *tx_buf : [Transmit data buffer address] uint16_t tx_num : [Transmit data buffer size] uint8_t *rx_buf : [Receive data buffer address]
Return value	[MD_OK]: Transmission/reception setting has completed. [MD_ARGERROR]: Transmission/reception setting has failed.
Remarks	None

[Function Name] r_csi00_interrupt	
Synopsis	CSI00 transfer end interrupt function
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	static void __near r_csi00_interrupt(void)
Explanation	If there is data not transmitted, this function reads receive data and then starts transmitting the data not transmitted. Otherwise, this function reads receive data.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_csi00_callback_receiveend	
Synopsis	CSI00 data reception completion processing
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	static void r_csi00_callback_receiveend(void)
Explanation	If the reception of data is completed, set the BUSY signal to BUSY state.
Arguments	None
Return value	None
Remarks	None

5.4.7 Flowcharts

5.4.7.1 Overall Flow

Figure 5.5 shows the overall flow of the sample program described in this application note.
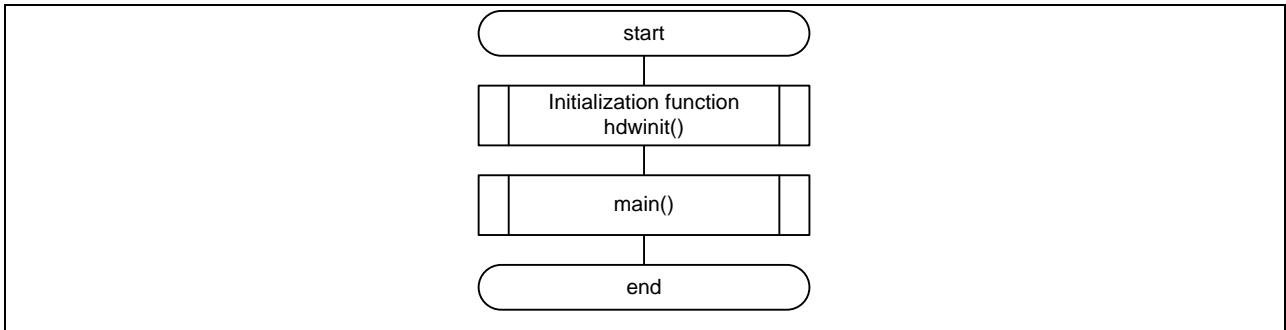

Figure 5.5   Overall Flow

5.4.7.2 Initialization Function

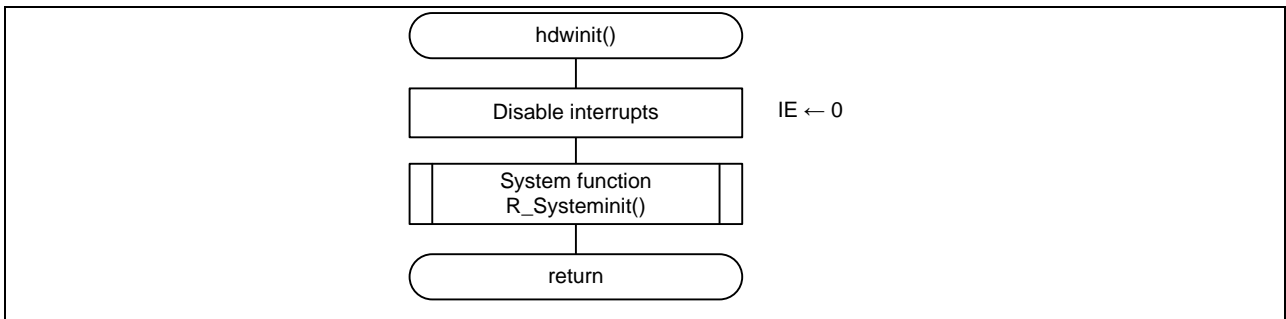Figure 5.6 shows the flowchart for the initialization function.

Figure 5.6 Initialization Function

5.4.7.3   System Function
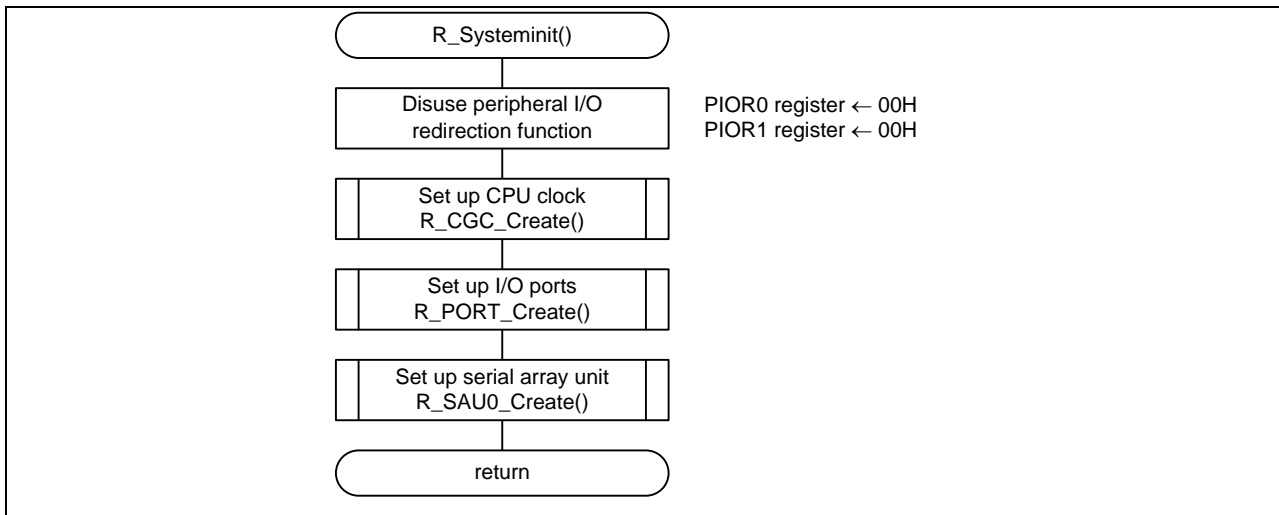
Figure 5.7 shows the flowchart for the system function.

Figure 5.7 System Function

5.4.7.4   I/O Port Setup

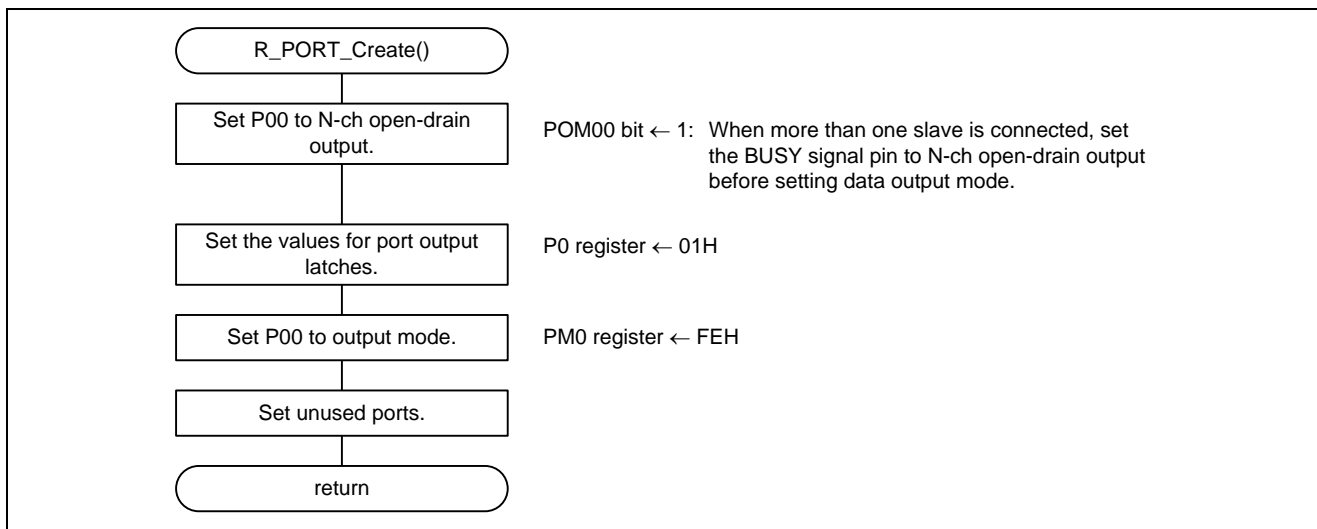Figure 5.8 shows the flowchart for I/O port setup.

Figure 5.8 I/O Port Setup

Note: Refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN2575EJ0100) for the configuration of the unused ports.

Caution: Provide proper treatment for unused pins so that their electrical specifications are observed. Connect each of any unused input-only ports to V_{DD} or V_{SS} via a separate resistor.

Setting up the BUSY signal output ports
- • Port register 0 (P0)

- • Port mode register 0 (PM0)
  Select an I/O mode and output latch for each port.

Symbol: P0

7	6	5	4	3	2	1	0
0	P06	P05	P04	P03	P02	P01	P00
0	x	x	x	x	x	x	**1**

Bit 1

P00	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	**Output 1**	**Input high level**

Symbol: PM0

7	6	5	4	3	2	1	0
1	PM06	PM05	PM04	PM03	PM02	PM01	PM00
1	x	x	x	x	x	x	**0**

Bit 1

PM00	PM00 pin I/O mode selection
0	**Output mode (output buffer on)**
1	Input mode (output buffer off)

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

5.4.7.5   CPU Clock Setup

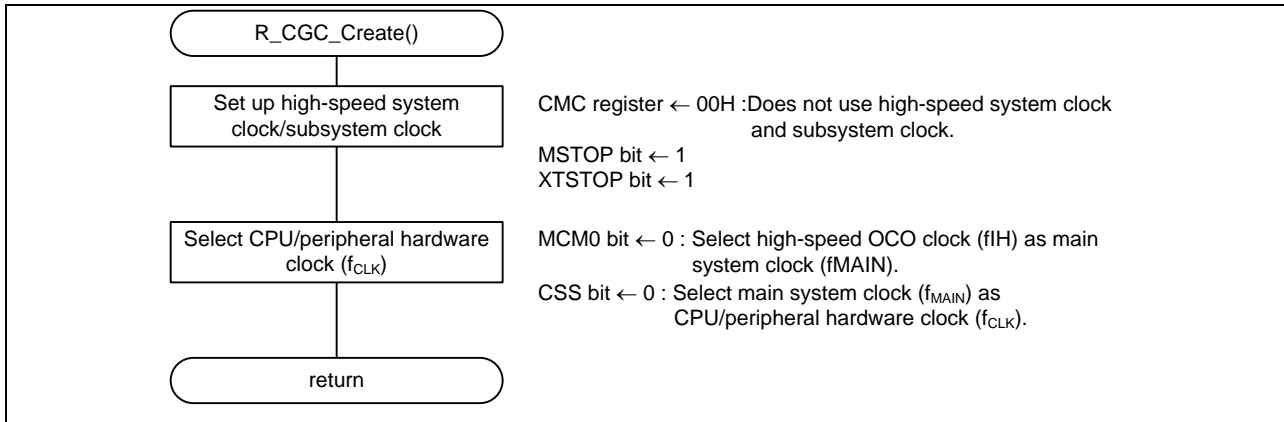Figure 5.9 shows the flowchart for setting up the CPU clock.

Figure 5.9 CPU Clock Setup

Caution: For details on the procedure for setting up the CPU clock (R_CGC_Create ()), refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN2575EJ0100).

5.4.7.6   SAU0 Setup

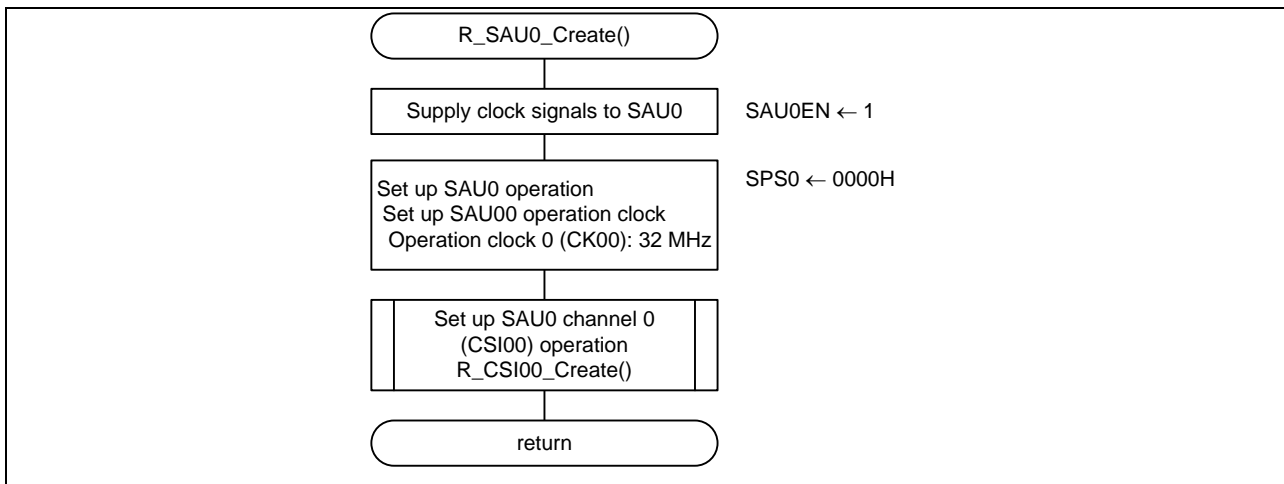Figure 5.10 shows the flowchart for SAU0 setup.

Figure 5.10 SAU0 Setup

Enabling supply of clock signals to the SAU

- Peripheral enable register 0 (PER0)
  Enable supply of clock signals to the SAU0.

Symbol: PER0

7	6	5	4	3	2	1	0
RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
x	x	x	x	x	1	x	x

Bit 2

SAU0EN	Control of serial array unit 0 and input clock supply
0	Stops input clock supply.
1	**Enables input clock supply.**

Select serial clock

- Serial clock selection register 0 (SPS0)
  Select an operation clock for the SAU0.

Symbol: SPS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PRS0 13	PRS0 12	PRS0 11	PRS0 10	PRS0 03	PRS0 02	PRS0 01	PRS0 00
0	0	0	0	0	0	0	0	x	x	x	x	**0**	**0**	**0**	**0**

Bits 3 to 0

PRS 003	PRS 002	PRS 001	PRS 000		Selection of operation clock (CK00)				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	**f_{CLK} = 32 MHz**
0	**0**	**0**	**0**	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	**32 MHz**
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1,25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^1$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	977 Hz

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

5.4.7.7   SAU0 Channel 0 (CSI00) Operation Setup

Figure 5.11 shows the flowchart for setting up SAU0 channel 0 (CSI00) operation.
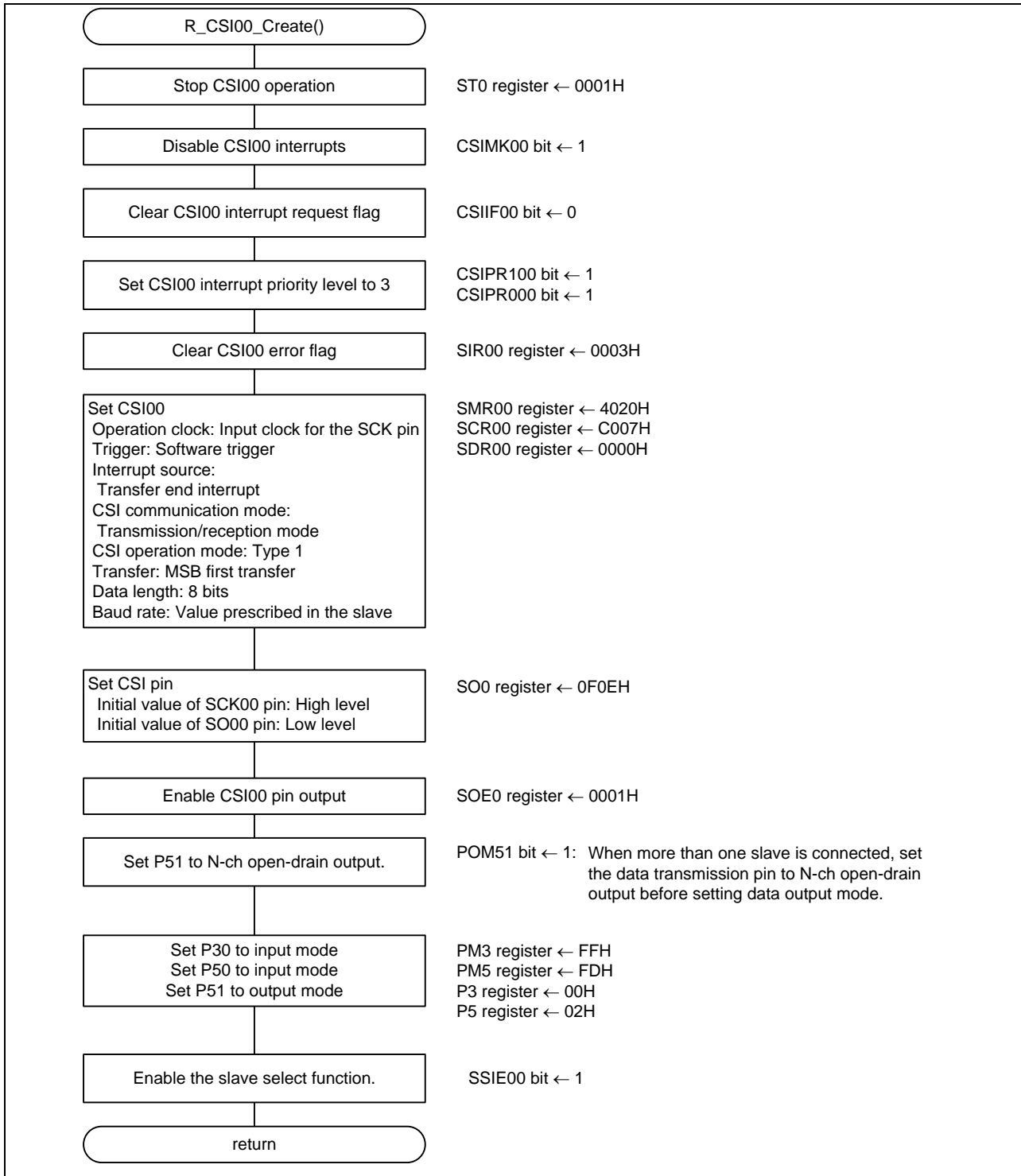

Figure 5.11 SAU0 Channel 0 (CSI00) Operation Setup

Stopping serial channel 0

- Serial channel stop register 0 (ST0)

    Stop communication/count operation of serial channel 0.

Symbol: ST0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	ST0 3	ST0 2	ST0 1	ST0 0
0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	1

Bit 0

ST00	Operation stop trigger of channel 0
0	No trigger operation
1	**Clears the SE00 bit to 0 and stops the communication operation.**

Setting a transfer end interrupt priority level

- Priority specification flag register 00H (PR00H)

- Priority specification flag register 10H (PR10H)

    Set the interrupt priority level.

Symbol: PR00H

7	6	5	4	3	2	1	0
SREPR00 TMPR001 H	SRPR00 CSIPR001 IICPR001	STPR00 **CSIPR000** IICPR000	1	1	SREPR02 TMPR011 H	SRPR02 CSIPR021 IICPR021	STPR02 CSIPR020 IICPR020
x	x	**1**	x	x	x	x	x

Symbol: PR10H

7	6	5	4	3	2	1	0
SREPR10 TMPR101 H	SRPR10 CSIPR101 IICPR101	STPR10 **CSIPR100** IICPR100	1	1	SREPR12 TMPR111H	SRPR12 CSIPR121 IICPR121	STPR12 CSIPR120 IICPR120
x	x	**1**	x	x	x	x	x

Bit 5

CSIPR000	CSIPR100	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	1	**Specify level 3 (low priority level)**

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Clearing the CSI00 error flags

- Serial flag clear trigger register 00 (SIR00)

  Clear the SAU0 channel 0 error flags.

Symbol: SIR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	FEC T00	PEC T00	OVCT 00
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit2

FECT00	Clear trigger of framing error of channel 0
0	Not cleared
1	**Clears the FEF00 bit of the SSR00 register to 0.**

Bit 1

PECT00	Clear trigger of parity error flag of channel 0
0	Not cleared
1	**Clears the PEF00 bit of the SSR00 register to 0.**

Bit 0

OVCT00	Clear trigger of overrun error flag of channel 0
0	Not cleared
1	**Clears the OVF00 bit of the SSR00 register to 0.**

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Setting up the SAU0 channel 0 operation mode

- Serial mode register 00 (SMR00)
  Select an operation clock (fMCK).
  Specify whether to make the serial clock (fSCK) input available.
  Set the start trigger and operation mode.
  Select an interrupt source.

Symbol: SMR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CK S00	CC S00	0	0	0	0	0	STS 00	0	SIS 000	1	0	0	MD 002	MD 001	MD 000
0	**1**	0	0	0	0	0	**0**	0	0	1	0	0	**0**	**0**	**0**

Bit 15

CKS00	Selection of operation clock (fMCK) of channel n
0	**Operation clock CK00 set by the SPS0 register**
1	Operation clock CK01 set by the SPS0 register

Bit 14

CCS00	Selection of transfer clock (fTCLK) of channel n
0	Divided operation clock fMCK specified by the CKS00 bit
1	**Clock input fSCK from the SCK00 pin (slave transfer in CSI mode)**

Bit 8

STS00	Selection of start trigger source
0	**Only software trigger is valid**
1	Valid edge of the RxDq pin (selected for UART reception)

Bits 2 and 1

MD002	MD001	Setting of operation mode of channel 0
0	**0**	**CSI mode**
0	1	UART mode
1	0	Simplified I²C mode
1	1	Setting prohibited

Bit 0

MD000	Selection of interrupt source of channel 0
0	**Transfer end interrupt**
1	Buffer empty interrupt

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Setting up the SAU0 channel 0 operation mode

- Serial communication operation setup register 00 (SCR00)
  Select an operation clock (f_{MCK}).
  Specify whether to make the serial clock (f_{SCK}) input available.
  Set up the start trigger and operation mode.
  Select an interrupt source.

Symbol: SCR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE 00	RXE 00	DAP 00	CKP 00	0	EOC 00	PTC 001	PTC 000	DIR 00	0	SLC 001	SLC 000	0	1	DLS 001	DLS 000
1	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**

Bits 15 and 14

TXE00	RXE00	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	**1**	**Transmission/reception**

Bits 13 and 12

DAP00	CKP00	Selection of data and clock phase in CSI mode	Type
0	**0**		**1**
0	1		2
1	0		3
1	1		4

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Symbol: SCR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE 00	RXE 00	DAP 00	CKP 00	0	EOC 00	PTC 001	PTC 000	DIR 00	0	SLC 001	SLC 000	0	1	DLS 001	DLS 000
1	**1**	**0**	**0**	0	0	0	0	**0**	0	0	0	0	1	**1**	**1**

Bit 7

DIR00	Selection of data transfer sequence in CSI and UART modes
0	**Inputs/outputs data with MSB first.**
1	Inputs/outputs data with LSB first.

Bits 1 and 0

DLS001	DLS000	Setting of data length in CSI and UART modes
0	0	9-bit data length (stored in bits 0 to 8 of the SDR00 register) (can be set in UART0 mode only.)
1	0	7-bit data length (stored in bits 0 to 6 of the SDR00 register)
1	**1**	**8-bit data length (stored in bits 0 to 7 of the SDR00 register)**
Other than above		Setting prohibited

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

SO00 pin output value setting

• Serial output register 0 (SO0)
  Set the output values of the serial data output pin and the serial clock output pin

Symbol: SO0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	CKO03	CKO02	CKO01	CKO00	0	0	0	0	SO03	SO02	SO01	SO00
0	0	0	0	x	x	x	**1**	0	0	0	0	x	x	x	**0**

Bit 8

CKO00	Serial clock output of channel 0
0	Serial clock output value is "0".
1	**Serial clock output value is "1".**

Bit 0

SO00	Serial data output of channel 0
0	**Serial data output value is "0"**
1	Serial data output value is "1"

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Setting for enabling output through serial communication

- Serial output enable register 0 (SOE0)
  Enable output of serial communication

Symbol: SOE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SO E03	SO E02	SO E01	SO E00
0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	**1**

Bit 0

SOE00	Serial output enable/stop of channel 0
0	Stops output by serial communication operation.
1	**Enables output by serial communication operation.**

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Setting up the ports of the SCK00, SO00 and SI00 pins

- Port register 3 (P3)
- Port mode register 3 (PM3)
- Port register 5 (P5)
- Port mode register 5 (PM5)
  Select an input/output mode and output latch for each port.

Symbol: P3

7	6	5	4	3	2	1	0
P37	P36	P35	P34	P33	P32	P31	P30
x	x	x	x	x	x	x	**0**

Bit 0

P30	**Output data control (in output mode)**	**Input data read (in input mode)**
0	**Output 0**	**Input low level**
1	Output 1	Input high level

Symbol: PM3

7	6	5	4	3	2	1	0
PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30
x	x	x	x	x	x	x	**1**

Bit 0

PM30	**P30 pin I/O mode selection**
0	Output mode (output buffer on)
1	**Input mode (output buffer off)**

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Symbol: P5

7	6	5	4	3	2	1	0
P57	P56	P55	P54	P53	P52	P51	P50
x	x	x	x	x	x	**1**	**0**

Bit 1

P51	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	**Output 1**	**Input high level**

Bit 0

P50	Output data control (in output mode)	Input data read (in input mode)
0	**Output 0**	**Input low level**
1	Output 1	Input high level

Symbol: PM5

7	6	5	4	3	2	1	0
PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50
x	x	x	x	x	x	**0**	**1**

Bit 0

PM51	P51 pin I/O mode selection
0	**Output mode (output buffer on)**
1	Input mode (output buffer off)

Bit 0

PM50	P50 pin I/O mode selection
0	Output mode (output buffer on)
1	**Input mode (output buffer off)**

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Enabling the slave select function

- Input switch control register (ISC)
  Enable input to the SSI00 pin.

Symbol: ISC

7	6	5	4	3	2	1	0
SSIE00	0	0	0	0	0	ISC1	ISC0
1	x	x	x	x	x	x	x

Bit 7

SSIE00	Channel 0 SSI00 input setting in CSI communication and slave mode
0	Disables SSI00 pin input.
1	**Enables SSI00 pin input.**

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

5.4.7.8   Main Processing

Figure 5.12 shows the flowchart for main processing.
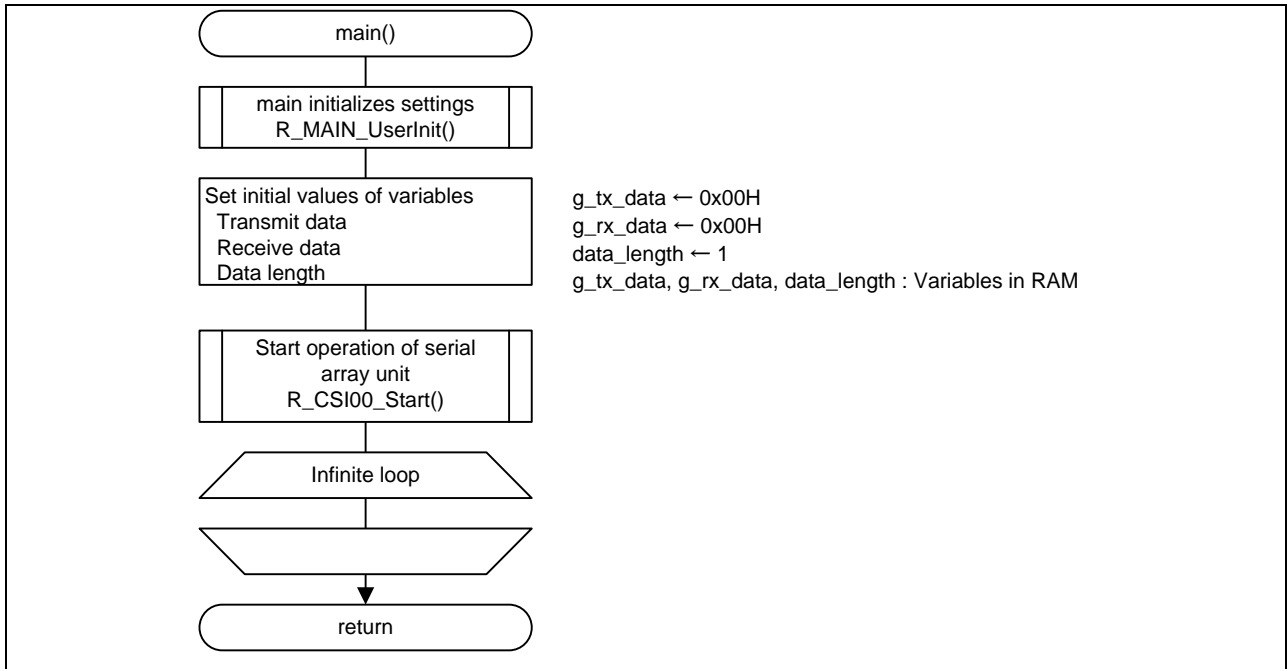
Figure 5.12   Main Processing

Caution: For infinite loop, please refer to the 5.4.7.11 Infinite Loop in Main Processing.

5.4.7.9   Main initializes settings

Figure 5.13 shows the flowchart for the main initializes settings.
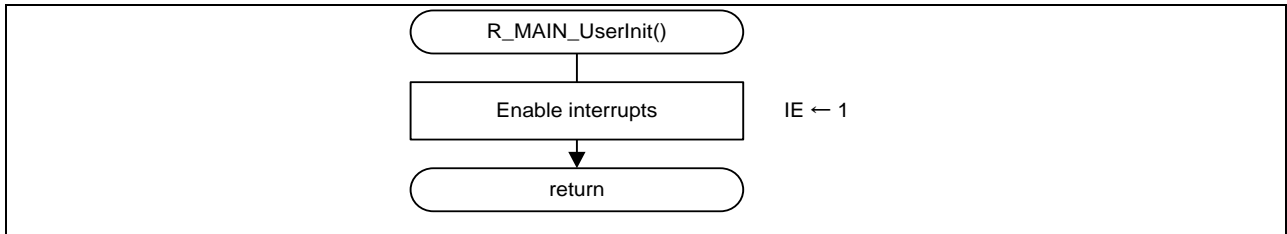
Figure 5.13 Main initializes settings

5.4.7.10   SAU0 Channel 0 (CSI00) Operation Start Processing

Figure 5.14 shows the flowchart for starting operation of SAU0 channel 0 (CSI00).
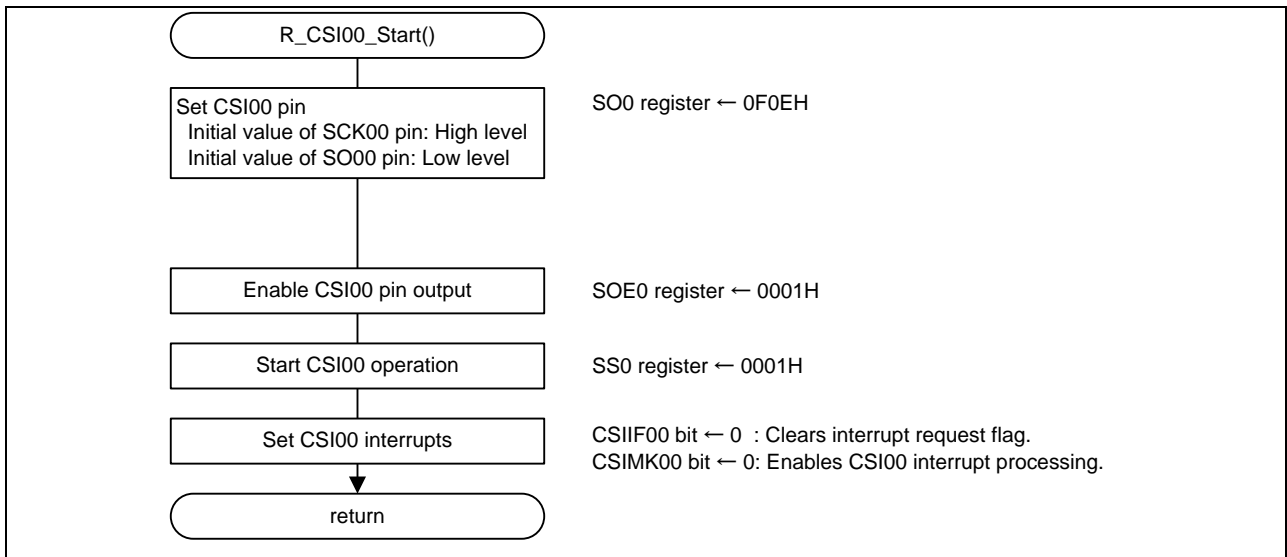
Figure 5.14 SAU0 Channel 0 (CSI00) Operation Start Processing

Setting the transfer end interrupt

- Interrupt request flag register 0H (IF0H)
  Clear the interrupt request flag.

- Interrupt mask flag register 0H (MK0H)
  Enable interrupt processing.

Symbol: IF0H

7	6	5	4	3	2	1	0
SREIF0 TMIF01H	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	0	0	SREIF2 TMIF11H	SRIF2 CSIIF21 IICIF21	STIF2 CSIIF20 IICIF20
x	x	**0**	0	0	x	x	x

Bit 5

CSIIF00	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Symbol: MK0H

7	6	5	4	3	2	1	0
SREMK0 TMMK01 H	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	0	0	SREMK2 TMMK11H	SRMK2 CSIMK21 IICMK21	STMK2 CSIMK20 IICMK20
x	x	**0**	0	0	x	x	x

Bit 5

CSIMK00	Interrupt processing control
0	Enables interrupt processing.
1	Disables interrupt processing.

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Enabling serial communication

- Serial channel start register 0 (SS0)
  Enable serial communication/count operation.

Symbol: SS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00
0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	1

Bit 0

SS00	Operation start trigger of channel 0
0	No trigger operation
1	Sets the SE00 bit to 1 and enters the communication wait status.

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

Remarks: When the SS0 register is read, 0000H is always read.

5.4.7.11   Infinite Loop in Main Routine

Figure 5.15 shows the flowchart for an infinite loop in the main routine.
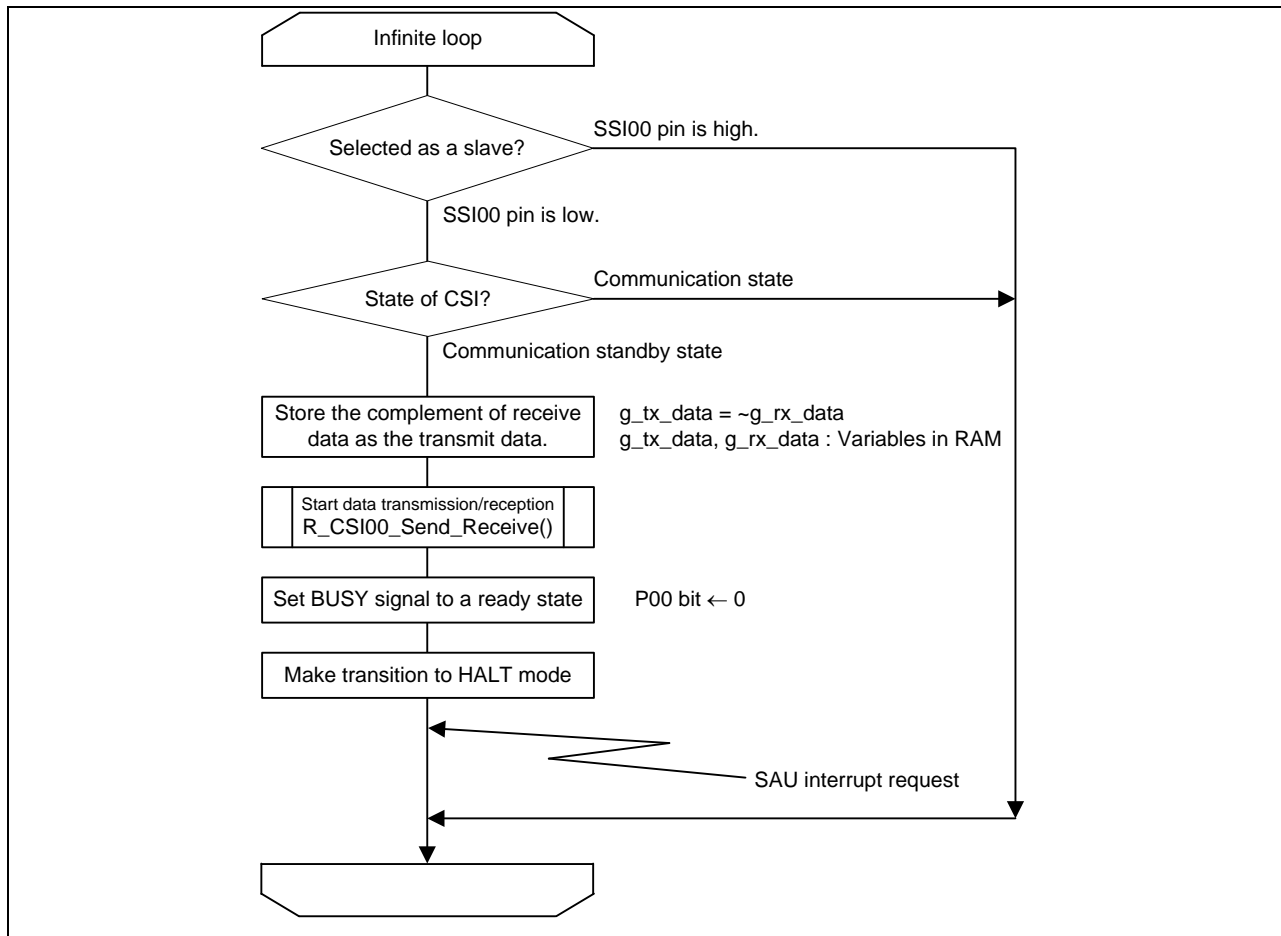
Figure 5.15 Infinite Loop in Main Processing

Confirming the communication state

•   Serial status register 00 (SSR00)
     Indicate the communication status and error occurrence status of serial array unit channel 0.

Symbol: SSR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	TSF 00	BFF 00	0	0	0	PEF 00	OV F00
0	0	0	0	0	0	0	0	0	**0/1**	x	0	0	0	x	x

Bit 6

TSF00	Communication status indication flag of channel 0
0	Communication is stopped or suspended.
1	Communication is in progress.

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

5.4.7.12   **CSI00 Data Transmission/Reception Start**

Figure 5.16 shows the flowchart for starting CSI00 data transmission/reception.
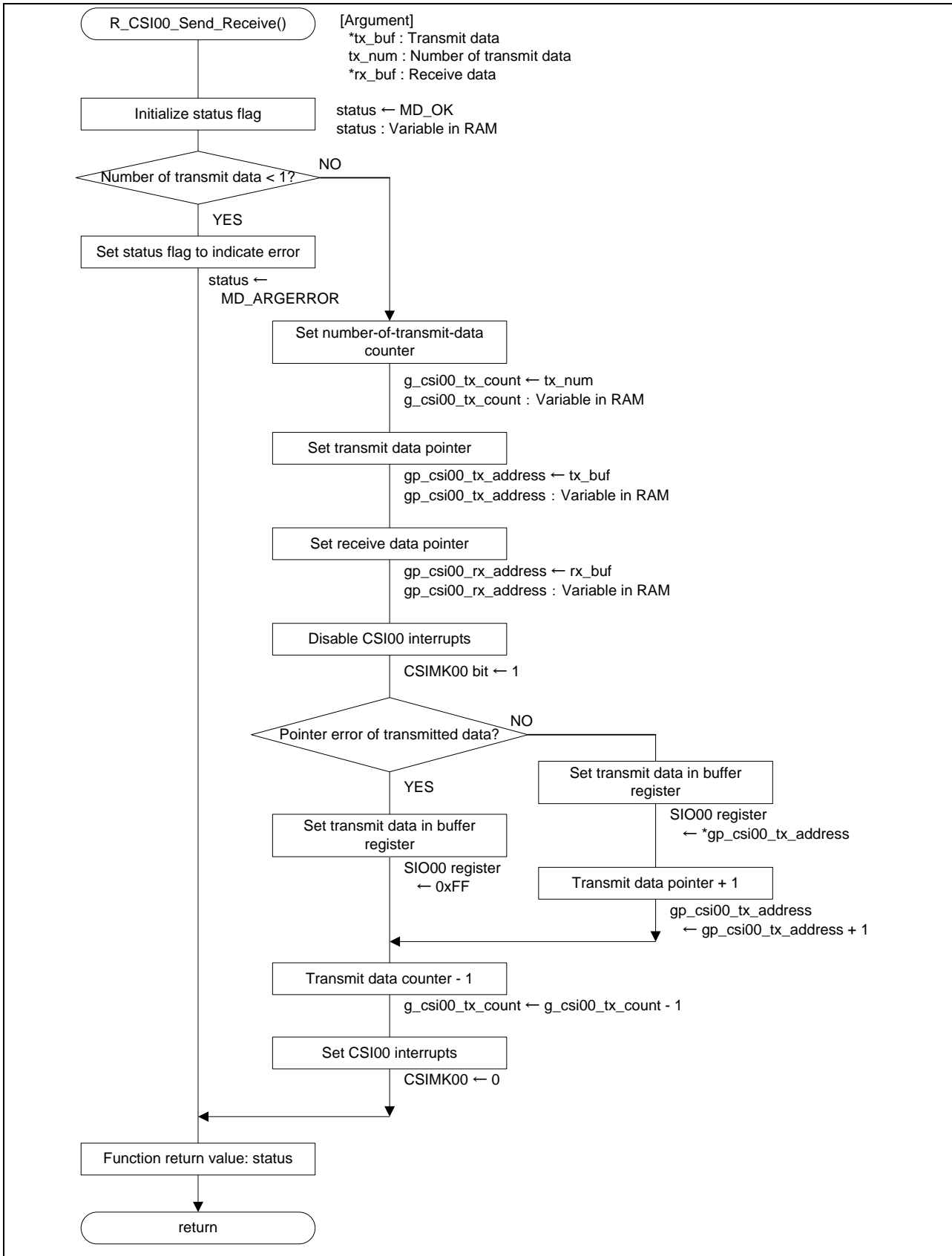
Figure 5.16   CSI00 Data Transmission/Reception Start

Setting transmit data

- Serial data register 00 (SDR00)
  Set transmit data and start transmitting the data.

Symbol: SDR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CSI00 data register (SIO00)
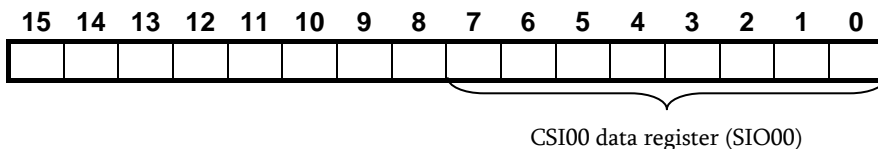
Write transmit data to the lower eight bits.

These eight bits should be accessed as the CSI00 register.

Caution: For details on the register setup procedures, refer to RL78/G14 User's Manual: Hardware.

5.4.7.13 CSI00 Transfer End Interrupt Processing

Figure 5.17, 18 shows the flowchart for CSI00 transfer end interrupt processing.
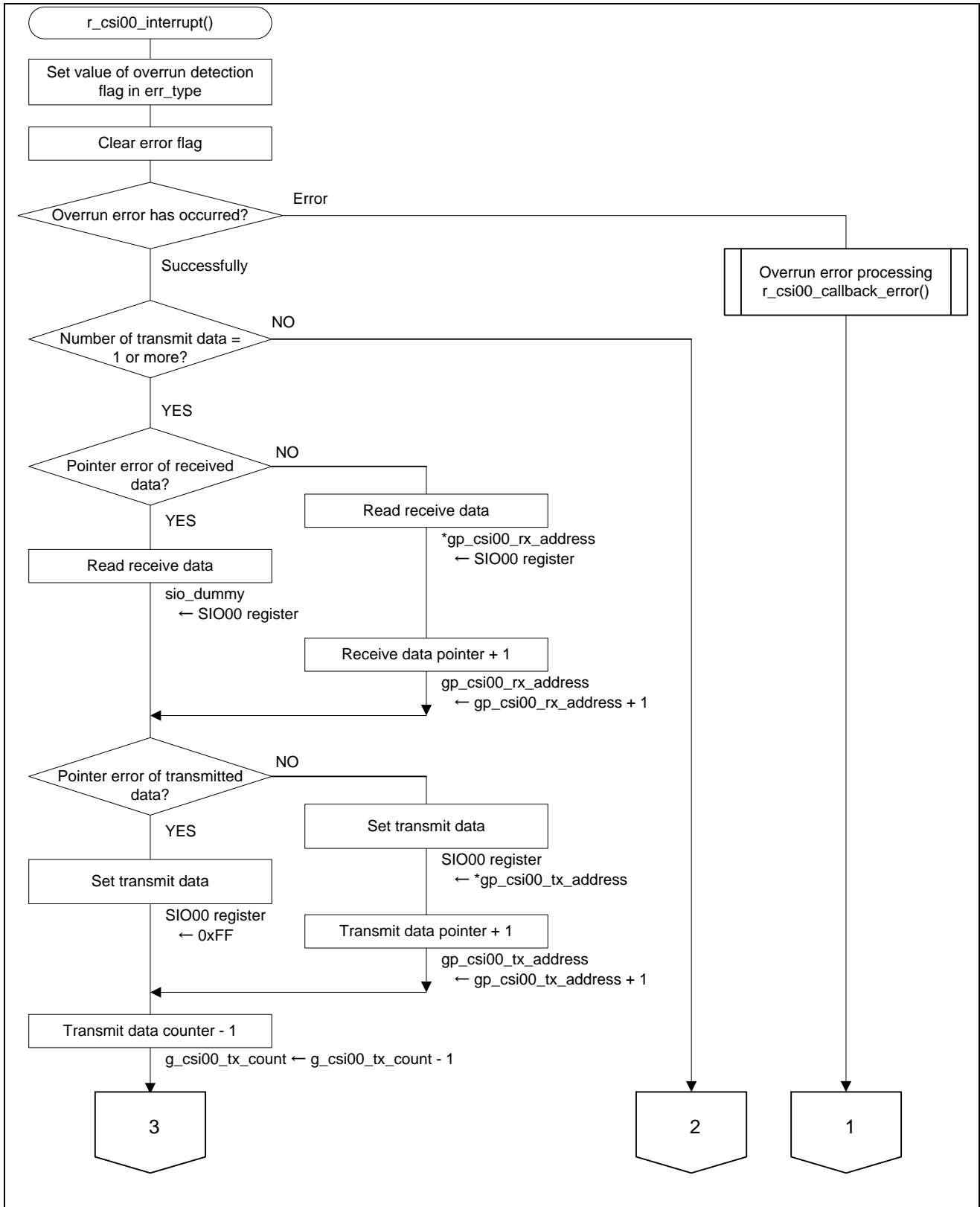
Figure 5.17   CSI00 Transfer End Interrupt Processing (1/2)
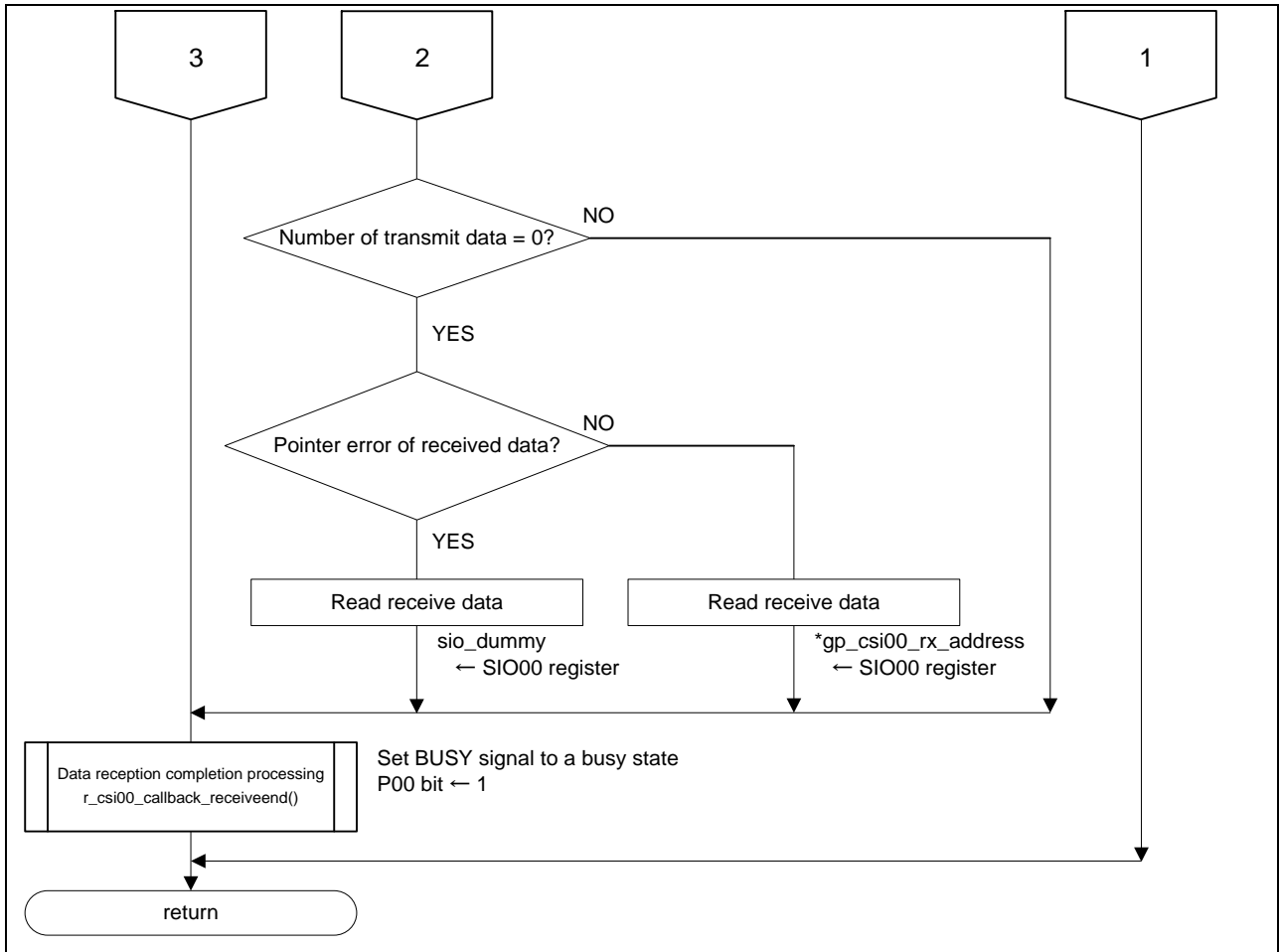
Figure 5.18   CSI00 Transfer End Interrupt Processing (2/2)

5.4.7.14   Data reception completion processing
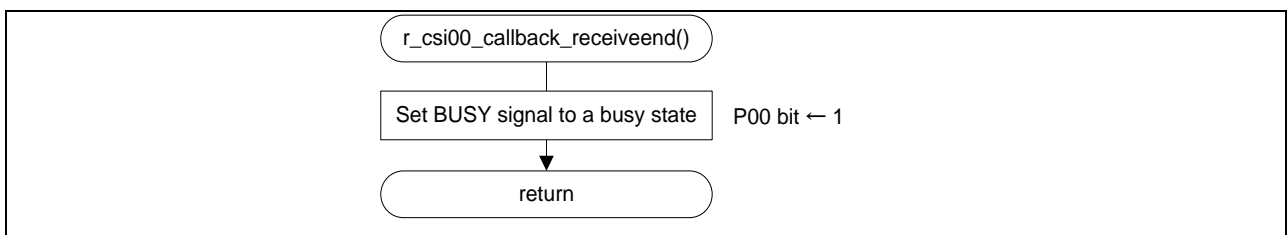
Figure 5.19 shows the flowchart for the data reception completion processing.
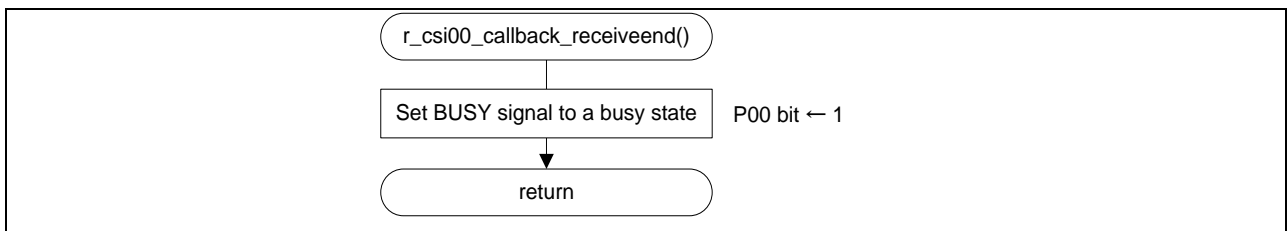
Figure 5.19   Data reception completion processing

5.5   Sample Code

The sample code is available on the Renesas Electronics Website.

5.6   Related Application Note

The application note that is related to this application note is listed below for reference.

- •   RL78/G13 Initialization (R01AN2575EJ0100) Application Note

- •   RL78/G13 Serial Array Unit for 3-Wire Serial I/O (Master Transmission/Reception) (R01AN2547EJ0100)
  Application Note

5.7   Documents for Reference

User's Manual:

RL78/G14 User's Manual: Hardware (R01UH0186)

R8C/36M Group User's Manual: Hardware (R01UH0259)

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

Technical Update/Technical News:

The latest information can be downloaded from the Renesas Electronics website.

Migration Guide

Migration to CubeSuite+ Integrated Development Environment for RL78 Family

(On-chip Debug) - Migration from R8C, M16C to RL78 (R20UT2150)

Website and Support

Renesas Electronics Website

http://japan.renesas.com/

Inquiries

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

All trademarks and registered trademarks are the property of their respective owners.

Revision History

| | | Description | |
Rev.	Date	Page	Summary
1.00	Apr. 19, 2018	-	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 claims 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, drawings, charts, programs, algorithms, and 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 reverse engineer 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, copying or reverse engineering.

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; 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.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; 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 any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

6. When using 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 dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of 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. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a 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. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using 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 comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

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 directly or indirectly controlled subsidiaries.

(Note 2)   "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)