RL78/G14, R8C/36M Group

Migration Guide from R8C to RL78: Serial Interface

R01AN3990EJ0100
Rev.1.00
Dec 22, 2017

Introduction

This application note explains how to achieve each mode (clock synchronous serial I/O mode and clock asynchronous serial I/O mode) of the serial interface (UART0) in the R8C/36M group using the serial array unit (SAU) in RL78/G14.

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

1. Migration Method from R8C Family to RL78 Family

This application note explains how to achieve each mode (clock synchronous serial I/O mode and clock asynchronous serial I/O mode) of the serial interface (UART0) in the R8C/36M group using the serial array unit (SAU) in RL78/G14.

Table 1.1 shows the operation modes of the serial interface in the R8C/36M group, and Table 1.2 shows the operation modes of the serial array unit (SAU) in RL78/G14.

Table 1.1 Operation Modes of Serial Interface in R8C/36M Group (Summary)

Serial Interface in R8C/36M Group	
Operation Mode	Function
Clock synchronous serial I/O mode	Transmits and receives data using a transfer clock.
Clock asynchronous serial I/O mode (UART mode)	Transmits and receives data after setting the desired bit rate and transfer data format.

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 I^{2}C	A clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA)

In the R8C/36C group, UART0 supports clock synchronous serial I/O mode and clock asynchronous serial I/O mode (UART mode).

In RL78/G14, one SAU is provided with multiple serial channels that are used alone or in combination for 3-wire serial I/O (CSI), UART, and simplified I^{2}C communications.

Table 1.3 shows the correspondence between the serial I/O modes in the R8C/36M group and the serial communications in RL78/G14.

Table 1.3 Serial Interface Correspondence

R8C/36M Serial Interface (UART0)	RL78/G14 Serial Array Unit (SAU)
Clock synchronous serial I/O mode	3-wire serial I/O communication (CSI)
Clock asynchronous serial I/O mode (UART mode)	UART communication
(No corresponding functions)	Simplified I^{2}C

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 Serial I/O Mode and 3-Wire Serial I/O Communication (CSI)

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

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

Item	R8C/36M Group Clock Synchronous Serial I/O 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 bits	7/8 bits
Interrupt function	• Transmission end interrupt and transmit buffer empty interrupt • Reception end interrupt	• For transmission: transfer end interrupt and buffer empty interrupt • For reception: transfer end interrupt
Error detection	Overrun error	Overrun error
Selection of clock polarity/phase	Provided	Provided
Selection of data phase	Not provided	Provided
Use in STOP mode	Enabled (when external clock is selected)	Enabled (SNOOZE mode function) Note 1
Slave select function	Not provided	Provided Note 2

Notes

1. Enabled only when the internal clock is selected.
2. Only for channel 0 of unit 0 of the serial array unit (SAU)

2.2 Differences between Clock Asynchronous Serial I/O Mode and UART Communication

Table 2.2 shows the differences between clock asynchronous serial I/O mode (UART mode) in the R8C/36M group and UART communication in RL78/G14.

Table 2.2 Differences between Clock Asynchronous Serial I/O Mode (UART Mode) and UART Communication

Item	R8C/36M Group Clock Asynchronous Serial I/O Mode (UART Mode)	RL78/G14 UART Communication
Transfer clock	Internal or external clock	Internal clock
Data length	7/8/9 bits	7/8/9 bits
Setting of transmit/receive data level and selection of data level inversion	Not provided	Provided
Interrupt function	• Transmit buffer empty interrupt/ transmit end interrupt • Receive end interrupt	• For transmission: transfer end interrupt/buffer empty interrupt • For reception: transfer end interrupt • Error interrupt caused by framing error, parity error, or overrun error
Error detection	• Overrun error • Framing error • Parity error • Error sum flag	• Overrun error • Framing error • Parity error
Selection of data phase	Not provided	Provided
Parity bit	The following selectable: • Parity enabled/disabled • Even parity • Odd parity	The following selectable: • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity
Use in STOP mode	Enabled (when the external clock is selected)	Enabled (SNOOZE mode function)

2.3 Comparison between Registers

Table 2.3, Table 2.4, and Table 2.5 show the comparison between the UART0 registers in the R8C/36M group and the SAU registers in RL78/G14.

Table 2.3 Comparison between Registers (1)

Setting Items	R8C/36M Group	RL78/G14
Selection of serial I/O mode and data length	The SMD0 to SMD2 bits in the UiMR register	The MDmn1 and MDmn2 bits in the SMRmn register The DLSmn0 and DLSmn1 bits in the SCRmn register
Selection of internal/external clock	The CKDIR bit in the UiMR register	–
Selection of stop bit length	The STPS bit in the UiMR register	The SLCmn0 and SLCmn1 bits in the SCRmn register
Parity bit control	The PRY and PRYE bits in the UiMR register	The PTCmn0 and PTCmn1 bits in the SCRmn register
Bit rate	The UiBRG register	The SDRmn register Note
Transmit buffer	The UiTB register	The SDRmn register Note
Selection of count source	The CLK0 and CLK1 bits in the UiC0 register	The CSS and MCM0 bits in the CKC register The SPSm register The CKSmn and CCSmn bits in the SMRmn register
Transmit register empty flag	The TXEPT bit in the UiC0 register	The TSFmn bit in the SSRmn register
Selection of port output mode	The NCH bit in the UiC0 register	The POMxx registers
Selection of clock polarity	The CKPOL bit in the UiC0 register	The CKPmn and DAPmn bits in the SCRmn register

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 transfer format	The UFORM bit in the UiC0 register	The DIRmn bit in the SCRmn register
Enabling transmission	The TE bit in the UiC1 register	The TXEmn bit in the SCRmn register The SSm register The STSmn bit in the SMRmn register
Transmit buffer empty flag	The TI bit in the UiC1 register	The BFFmn bit in the SSRmn register
Enabling reception	The RE bit in the UiC1 register	The RXEmn bit in the SCRmn register The SSm register The STSmn bit in the SMRmn register
Receive complete flag	The RI bit in the UiC1 register	–
Selection of transmit interrupt source	The UiIRS bit in the UiC1 register	The MDmn0 bit in the SMRmn register
Enabling continuous receive mode	The UiRRM bit in the UiC1 register	–
Receive buffer	The UiRB register	The SDRmn register Note
Overrun error flag	The OER bit in the UiRB register	The OVFmn bit in the SSRmn register
Framing error flag	The FER bit in the UiRB register	The FEFmn bit in the SSRmn register
Parity error flag	The PER bit in the UiRB register	The PEFmn bit in the SSRmn register
Error sum flag	The SUM bit in the UiRB register	–

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 selection	The TXD0SEL0, RXD0SEL0, and CLK0SEL0 bits in the U0SR register	The PIM0, PIM1, PIM3, and PIM5 registers The POM0, POM1, POM3, POM5, and POM7 registers The PM0, PM1, PM3, PM5, and PM7 registers The P0, P1, P3, P5, and P7 registers
Control of receive data level inversion on channel n	–	The SISmn0 bit in the SMRmn register
Mask control of error interrupt signal	–	The EOCmn bit in the SCRmn register
Clearing of error flags	–	The SIRmn register
Operation stop trigger of channel n	–	The STmn bit in the STm register
Indication of operation enable/stop status of channel n	–	The SEmn bit in the SEm register
Enabling or stopping serial output of channel n	–	The SOEmn bit in the SOEm register
Serial clock output of channel n	–	The CKOmn bit in the SOm register
Serial data output of channel n	–	The SOmn bit in the SOm register
Selection of transmit data level inversion on channel n	–	The SOLm register
Enabling or disabling generation of transfer end interrupt	–	The SSECm bit in the SSCm register
Enabling or disabling startup of UART0/UART2 reception in STOP mode	–	The SWCm bit in the SSCm register
Input switching control (LIN-bus communication operation)	–	The ISC register
Enabling use of the noise filter	–	The NFEN0 register

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

3. How to Migrate Serial Interface in This Sample Code

In this sample program, the serial interface operation of the R8C/36M group is realized with RL78/G14 by the method shown in

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	RL78/G14
Clock synchronous serial I/O mode	3-wire serial I/O communication (CSI)
Clock asynchronous serial I/O mode (UART mode)	UART communication

4. Example of Migration from Clock Synchronous Serial I/O Mode

4.1 Specifications

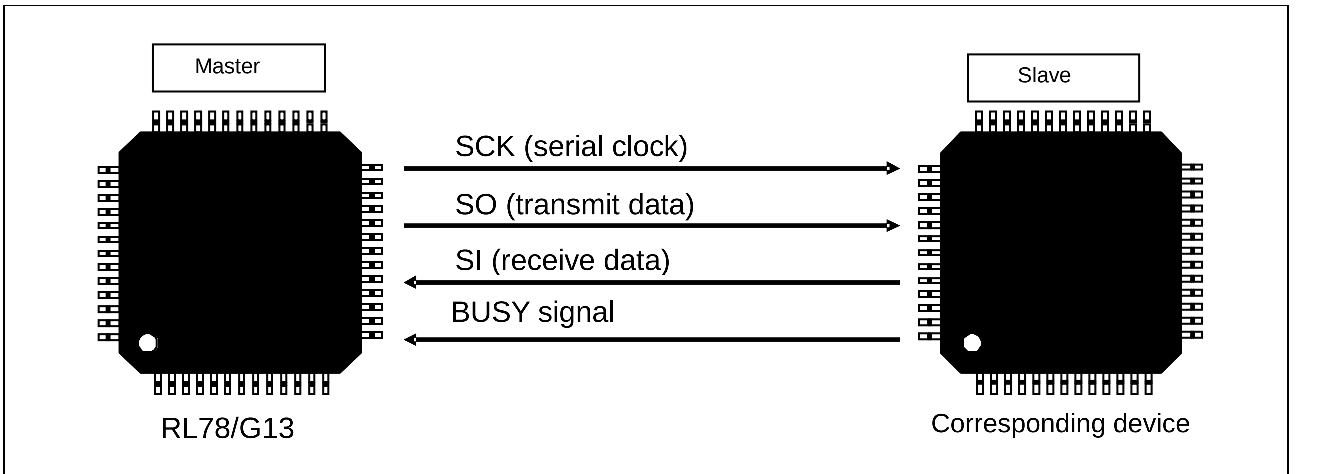
The serial array unit (SAU) described in this application note performs master transmission and reception by 3-wire serial I/O communication (CSI). As the CSI master, this unit supplies clock signals to the slave, transmits data 0x05 and 0x50 alternately to the slave, and receives data from the slave.

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

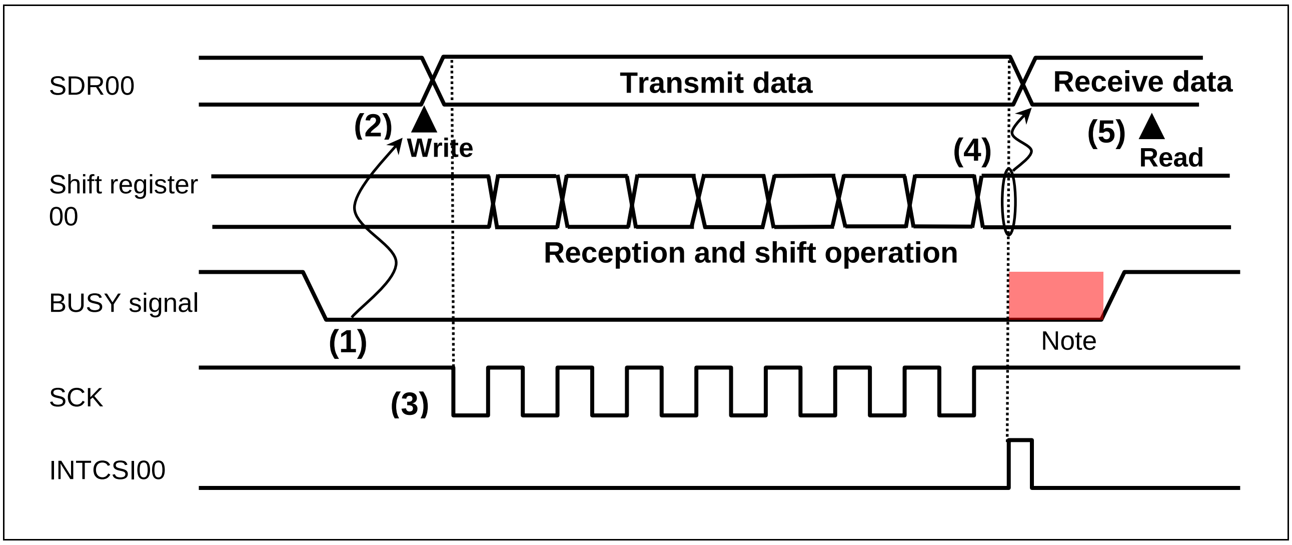


Figure 4.2 Handshake Operation and Communication

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

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

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

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

(5) [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.

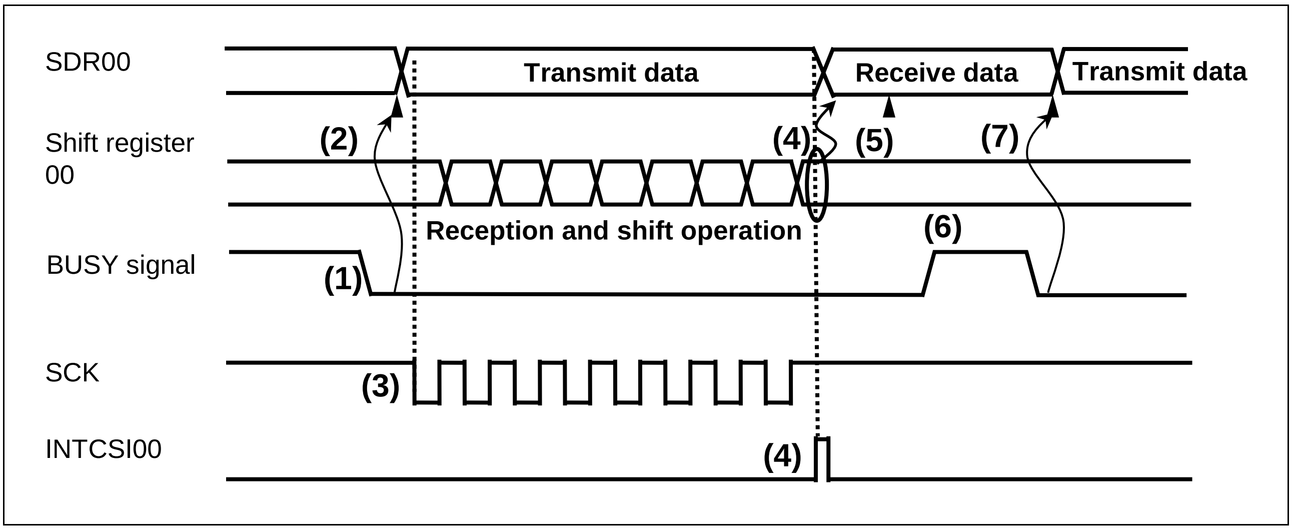


Figure 4.3 Example of BUSY Signal Edge Detection in the Master

- 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 slave] Write the next transmit data to the master and make the BUSY signal fall.

(2) [Software processing in the master] Detect the falling edge of the BUSY signal and write transmit data to the SDR00 register.

(3) [Hardware processing in the master] Start transmission/reception and then output serial clock (SCK) signals.

(4) [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).

(5) [Software processing in the master] Read the receive data from the SDR00 register.

(6) [Software processing in the master] Wait until the falling edge of the BUSY signal is detected. Note

(7) [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 (e^2 studio)	e^2 studio V5.4.0.018 from Renesas Electronics Corp.
C compiler (e^2 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.

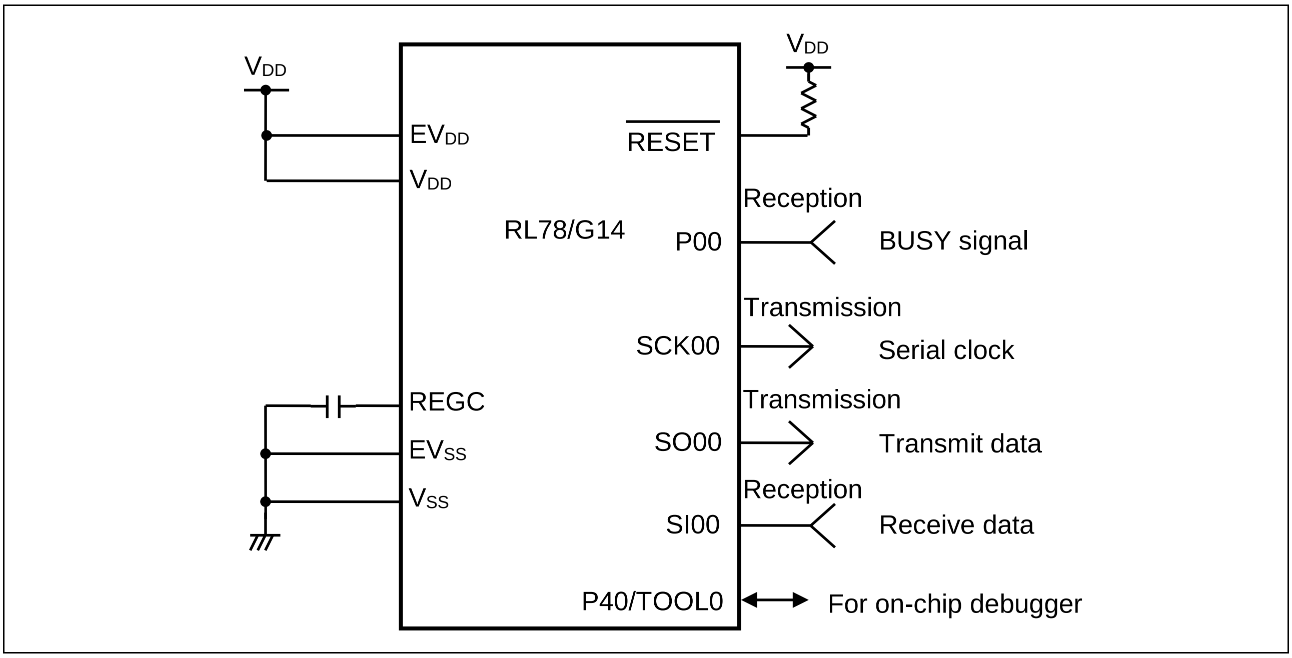


Figure 4.4 Hardware Configuration

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 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.

4.3.2 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
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.

4.4.2 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.

4.4.3 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.

4.4.4 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()

4.4.5 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

4.4.6 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

4.4.7 Flowcharts

4.4.7.1 Overall Flow

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

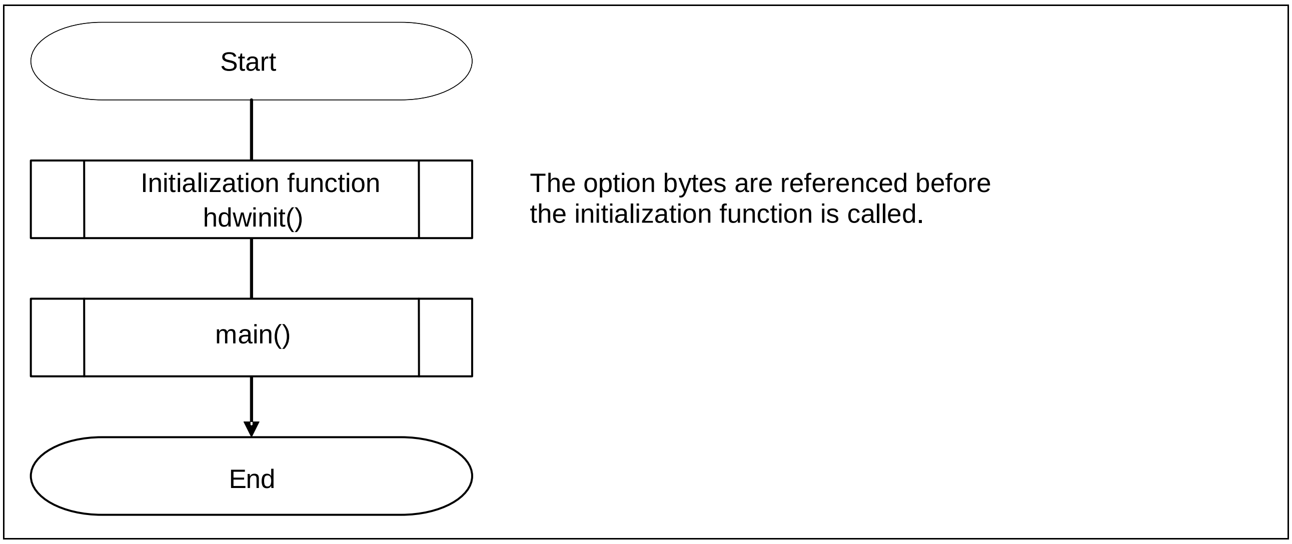

Start
Initialization function
hdwinit()
main()
End
The option bytes are referenced before
the initialization function is called.

Figure 4.5 Overall Flow

4.4.7.2 Initialization Function

Figure 4.6 shows the flowchart for the initialization function.

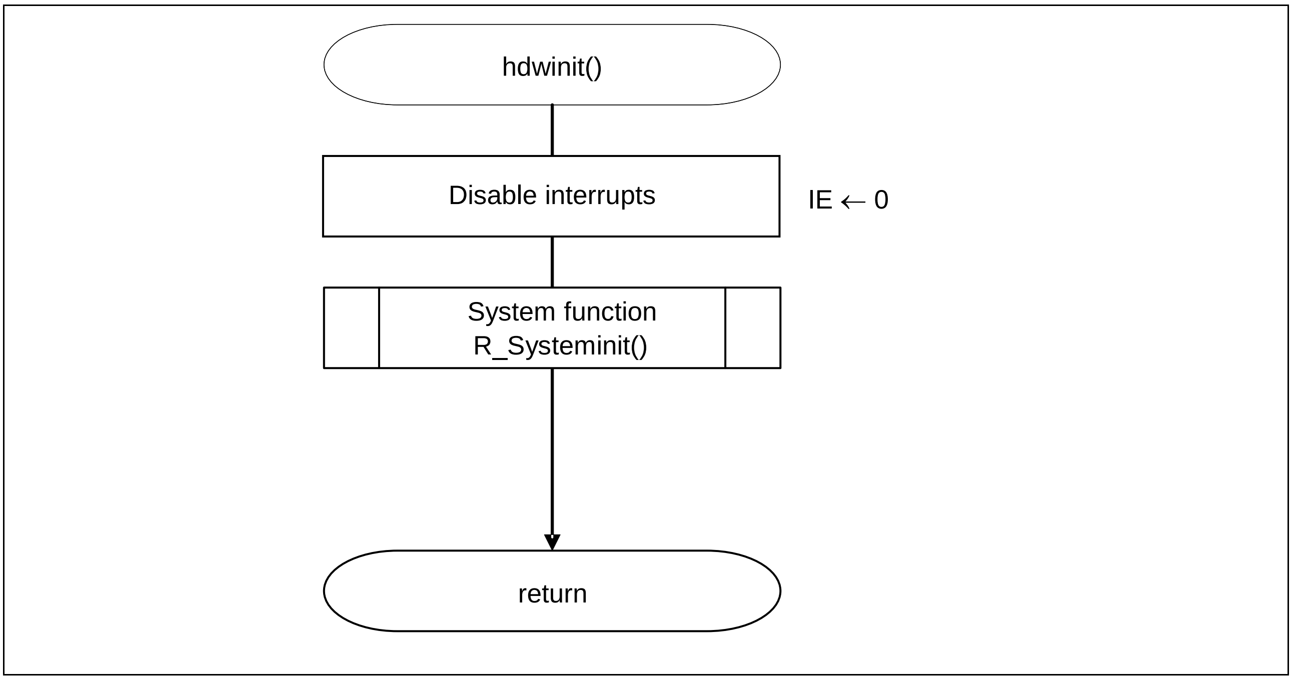


Figure 4.6 Initialization Function

4.4.7.3 System Function

Figure 4.7 shows the flowchart for the system function.

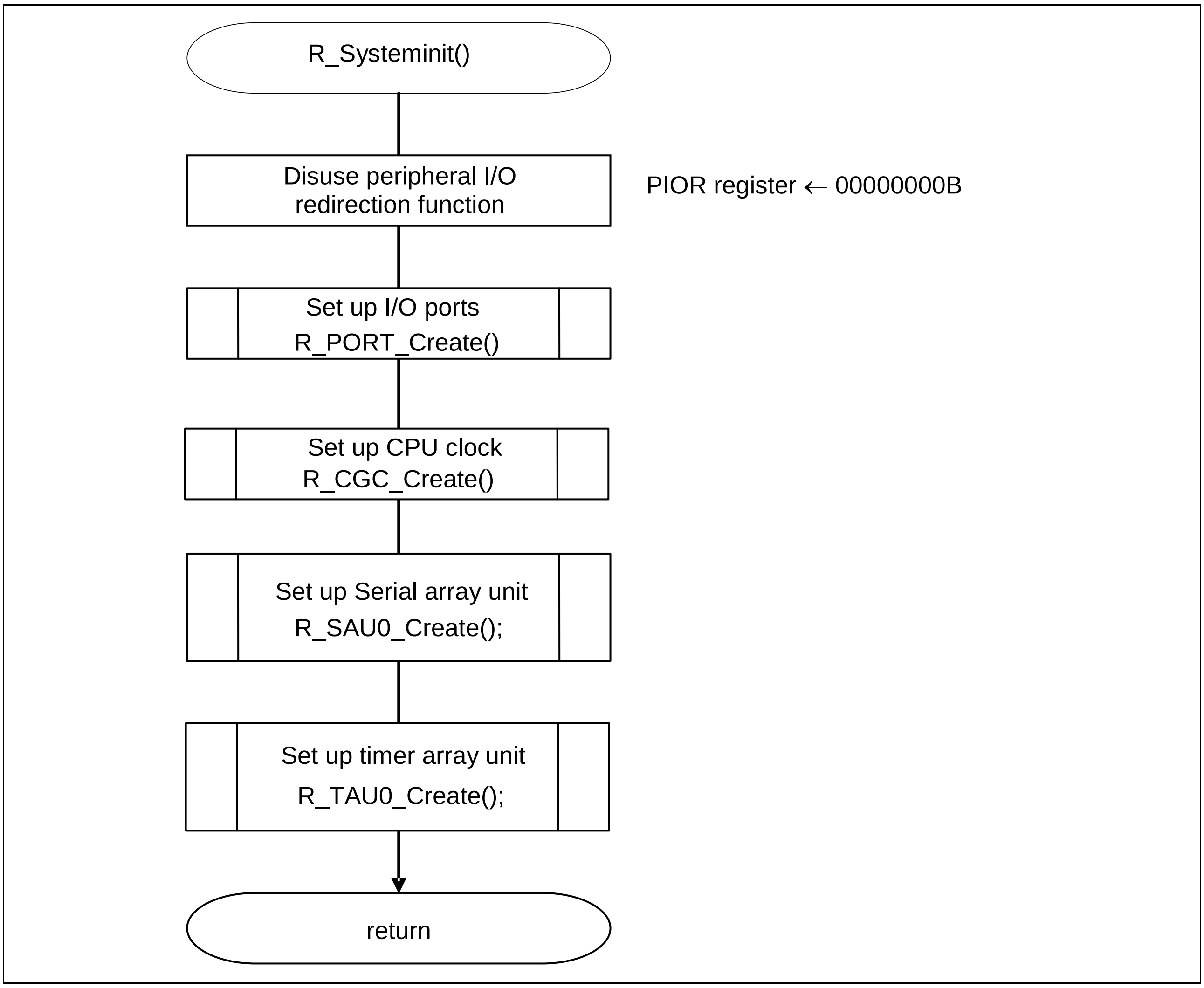


Figure 4.7 System Function

4.4.7.4 I/O Port Setup

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

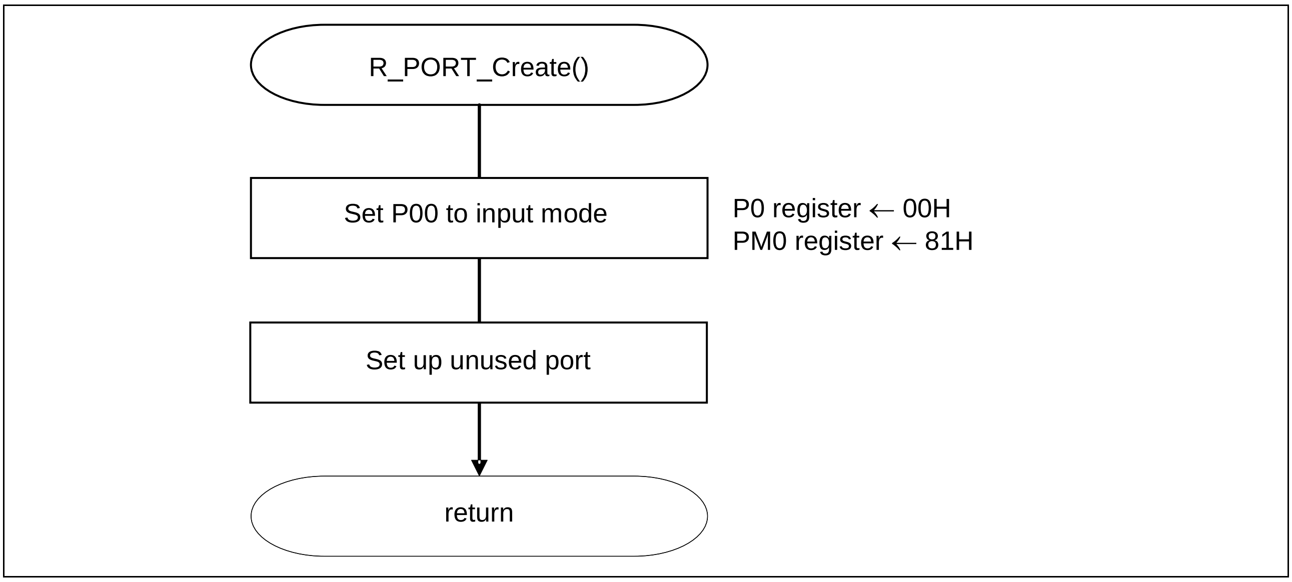


Figure 4.8 I/O Port Setup

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.

4.4.7.5 CPU Clock Setup

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

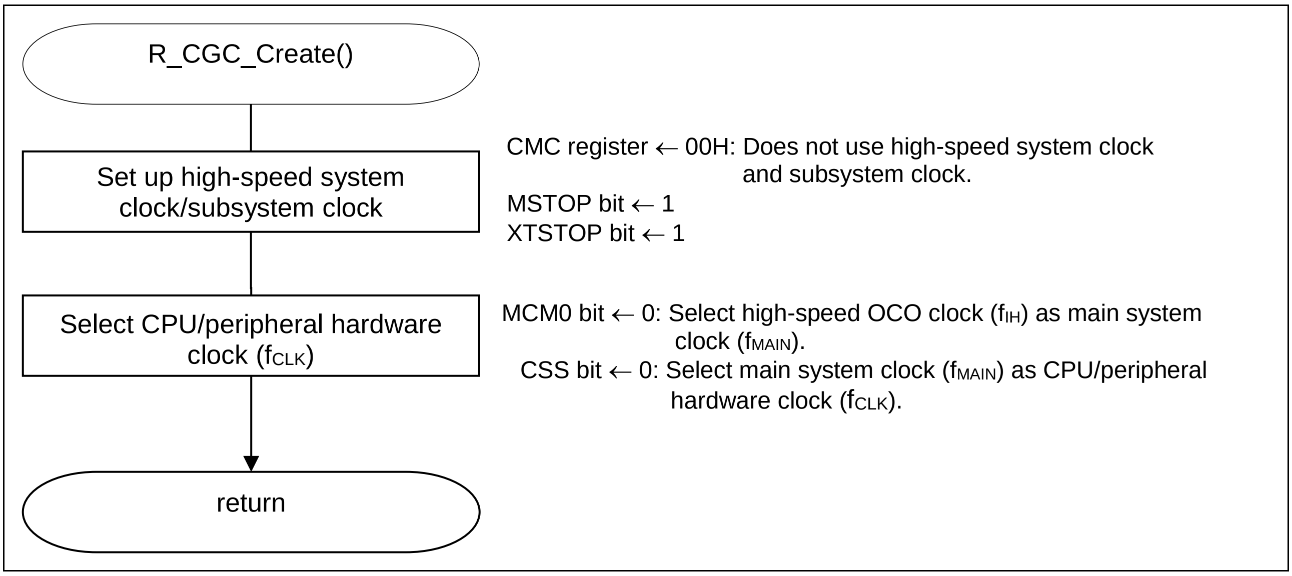

R_CGC_Create()
Set up high-speed system clock/subsystem clock
CMC register ← 00H: Does not use high-speed system clock and subsystem clock.
MSTOP bit ← 1
XTSTOP bit ← 1
Select CPU/peripheral hardware clock (fCLK)
MCM0 bit ← 0: Select high-speed OCO clock (fIH) as main system clock (fMAIN).
CSS bit ← 0: Select main system clock (fMAIN) as CPU/peripheral hardware clock (fCLK).
return

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).

4.4.7.6 SAU0 Setup

Figure 4.10 shows the flowchart for SAU0 setup.

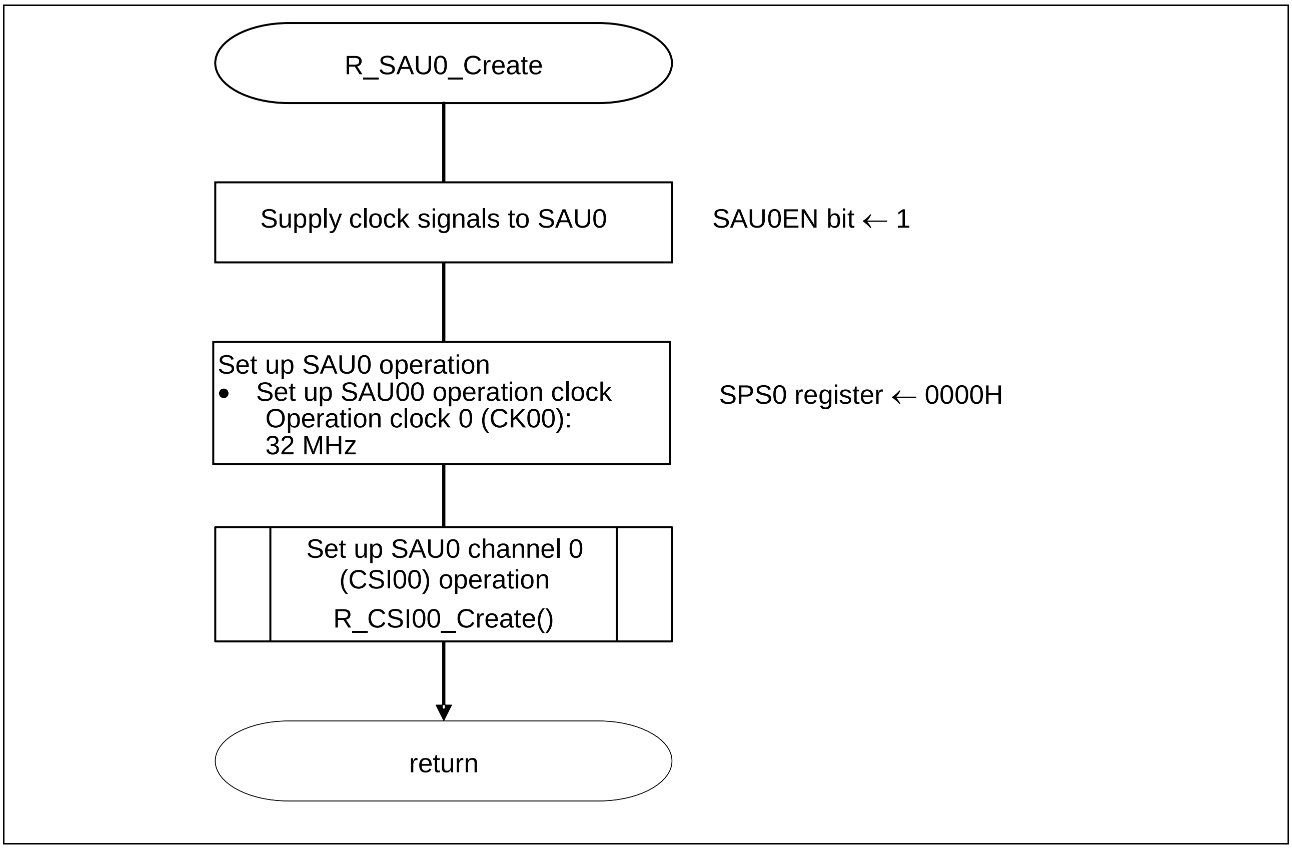


Figure 4.10 SAU0 Setup

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.**

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

Selecting a serial clock

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

4.4.7.7 SAU0 Channel 0 (CSI00) Operation Setup

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

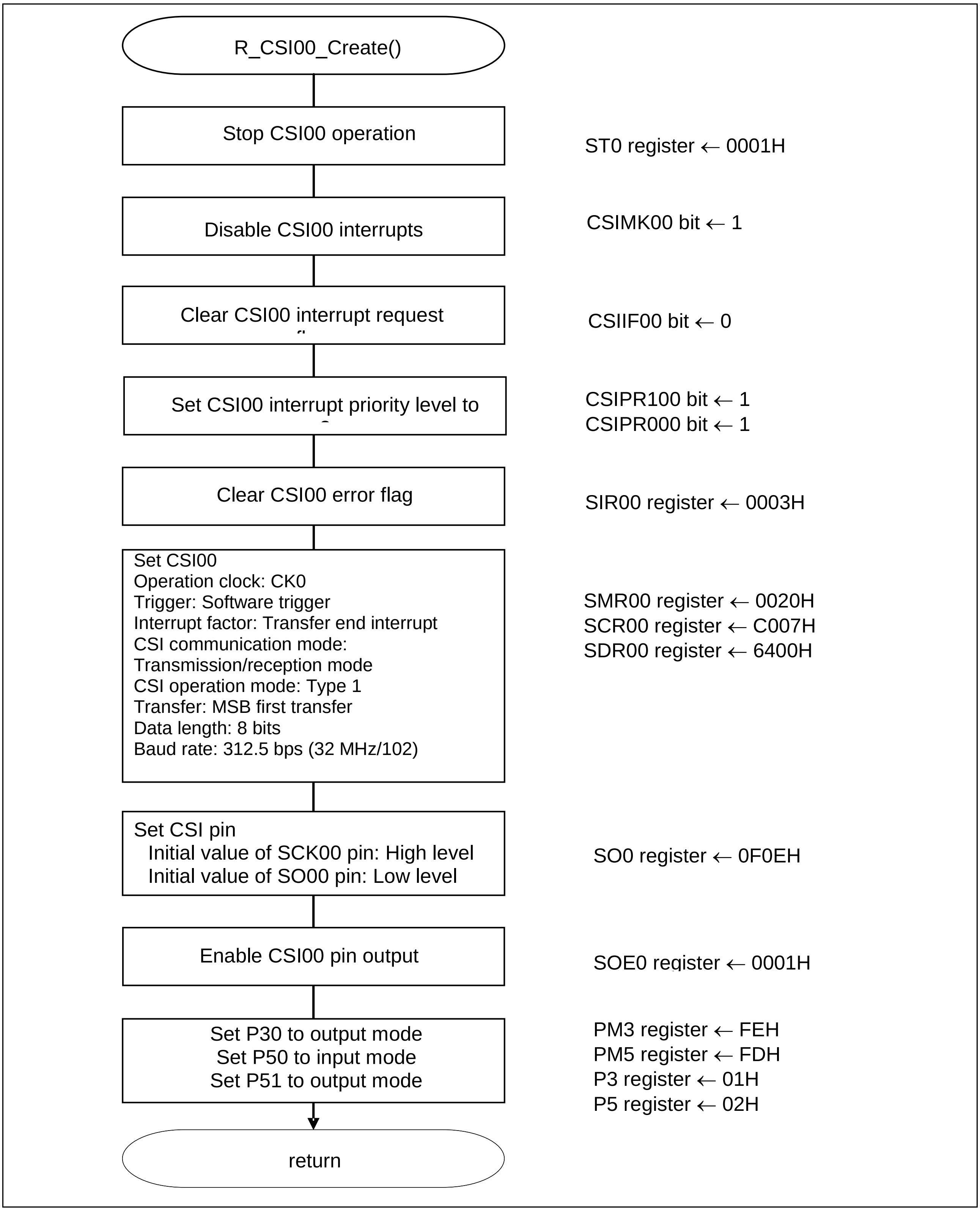


Figure 4.11 SAU0 Channel 0 (CSI00) Operation Setup

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	ST03	ST02	ST01	ST00
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	OVCT 00
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
CK S00	CC S00	0	0	0	0	0	STS 00	0	SIS 000	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 $\overline{\text{SCK00}}$ 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 I^2C 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.

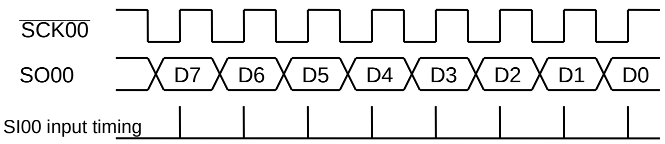
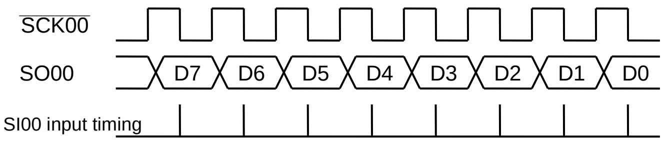
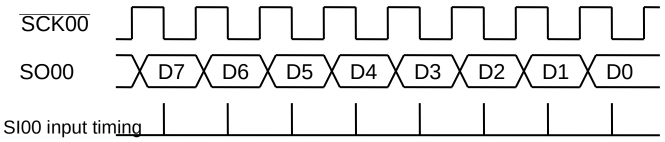
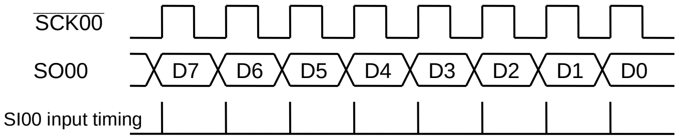
Symbol: SCR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE00	RXE00	DAP00	CKP00	0	EOC00	PTC001	PTC000	DIR00	0	SLC001	SLC000	0	1	DLS001	DLS000
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**	 SCK00 SO00 D7 D6 D5 D4 D3 D2 D1 D0 SI00 input timing	**1**
0	1	 SCK00 SO00 D7 D6 D5 D4 D3 D2 D1 D0 SI00 input timing	2
1	0	 SCK00 SO00 D7 D6 D5 D4 D3 D2 D1 D0 SI00 input timing	3
1	1	 SCK00 SO00 D7 D6 D5 D4 D3 D2 D1 D0 SI00 input timing	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	RX E00	DA P00	CK P00	0	EO C00	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	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 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	SOE03	SOE02	SOE01	SOE00
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.

4.4.7.8 TAU0 Setup

Figure 4.12 shows the flowchart for setting up TAU0.

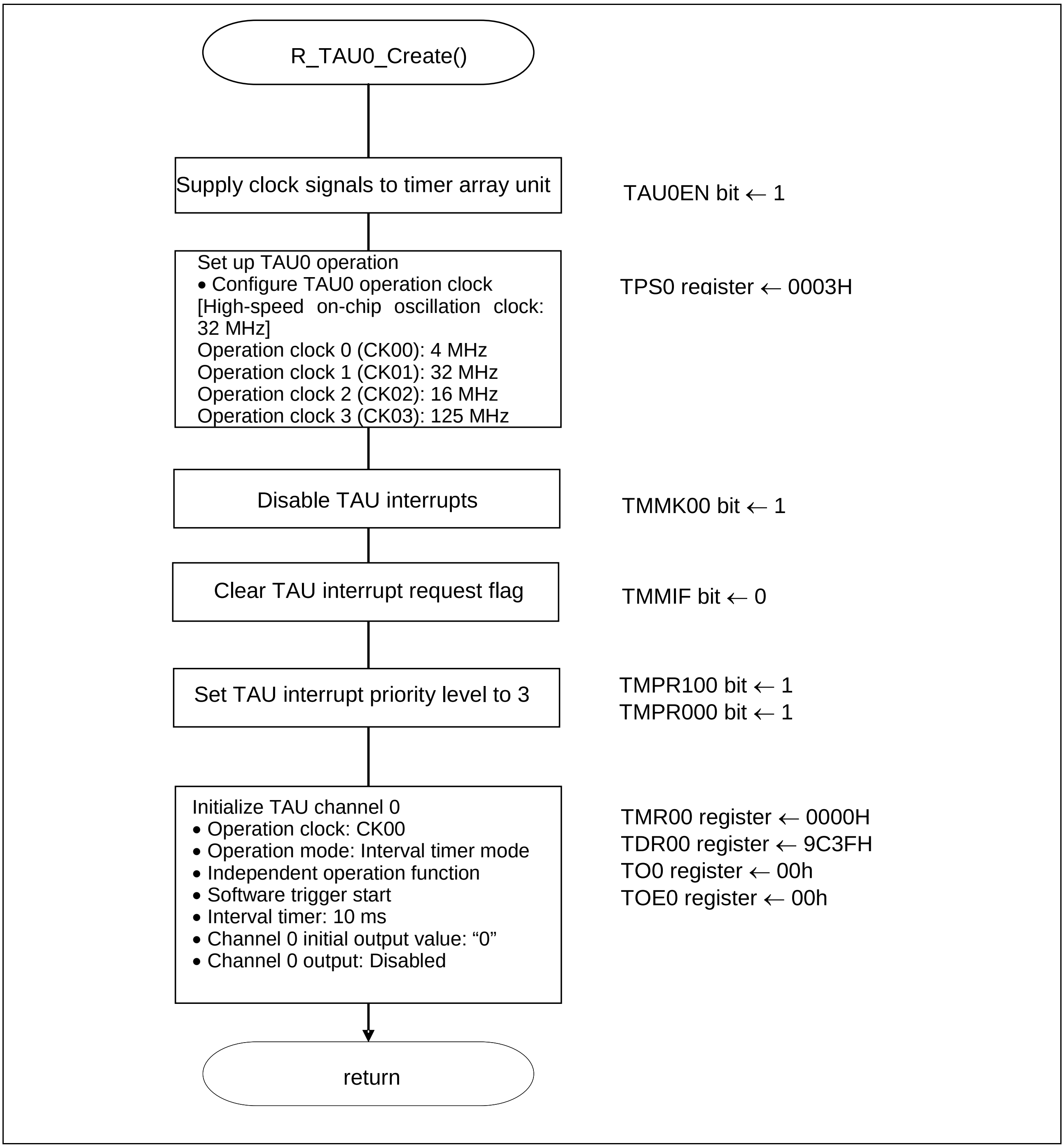

R_TAU0_Create()
Supply clock signals to timer array unit
TAU0EN bit ← 1
Set up TAU0 operation
• Configure TAU0 operation clock
[High-speed on-chip oscillation clock: 32 MHz]
Operation clock 0 (CK00): 4 MHz
Operation clock 1 (CK01): 32 MHz
Operation clock 2 (CK02): 16 MHz
Operation clock 3 (CK03): 125 MHz
TPS0 register ← 0003H
Disable TAU interrupts
TMMK00 bit ← 1
Clear TAU interrupt request flag
TMMIF bit ← 0
Set TAU interrupt priority level to 3
TMPR100 bit ← 1
TMPR000 bit ← 1
Initialize TAU channel 0
• Operation clock: CK00
• Operation mode: Interval timer mode
• Independent operation function
• Software trigger start
• Interval timer: 10 ms
• Channel 0 initial output value: “0”
• Channel 0 output: Disabled
TMR00 register ← 0000H
TDR00 register ← 9C3FH
TO0 register ← 00h
TOE0 register ← 00h
return

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).

4.4.7.9 Main Processing

Figure 4.13 shows the flowchart for main processing.

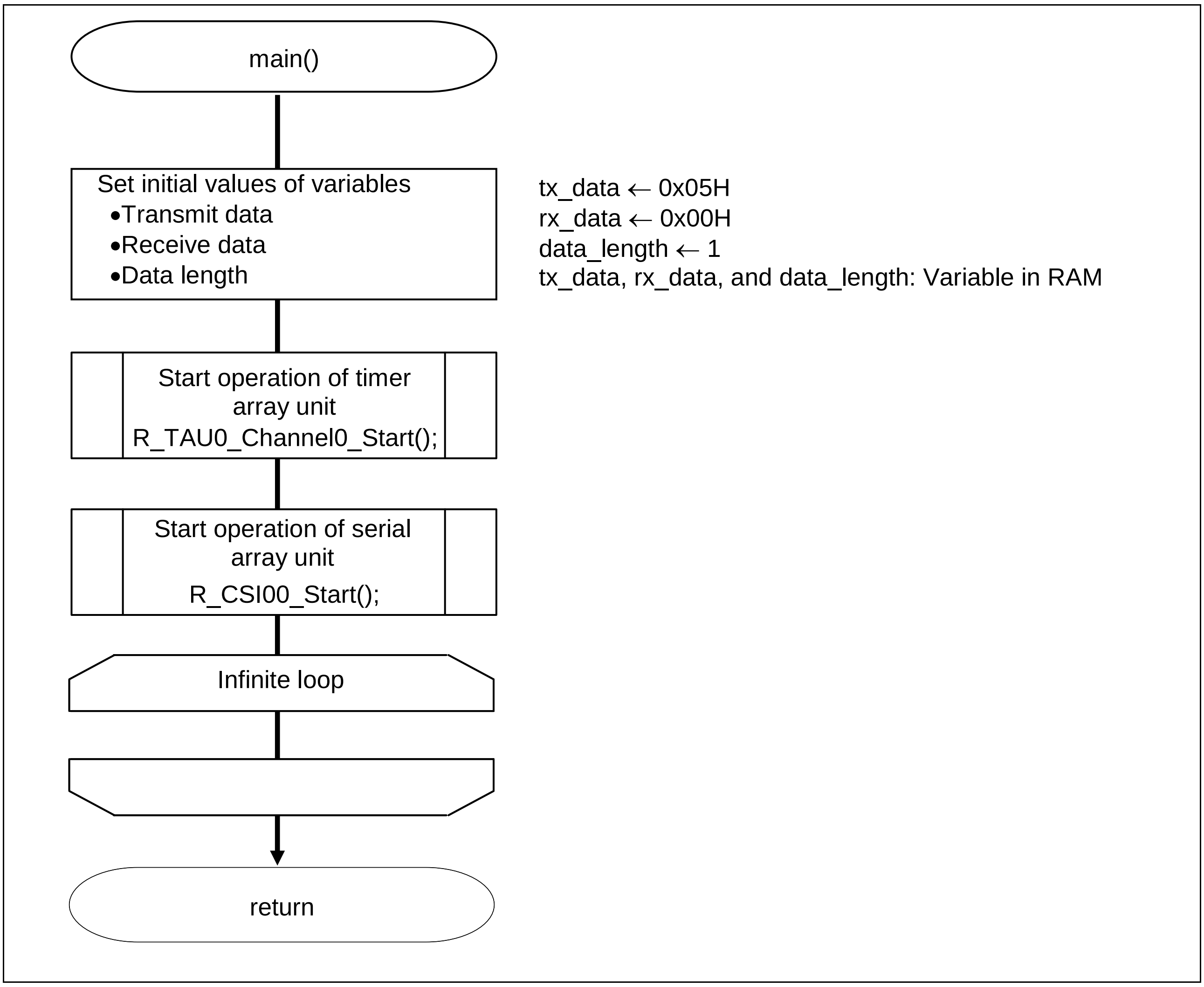

main()
Set initial values of variables
•Transmit data
•Receive data
•Data length
tx_data ← 0x05H
rx_data ← 0x00H
data_length ← 1
tx_data, rx_data, and data_length: Variable in RAM
Start operation of timer array unit
R_TAU0_Channel0_Start();
Start operation of serial array unit
R_CSI00_Start();
Infinite loop
return

Figure 4.13 Main Processing

4.4.7.10 TAU0 Channel 0 Startup

Figure 4.14 shows the flowchart for starting the operation of TAU0 channel 0.

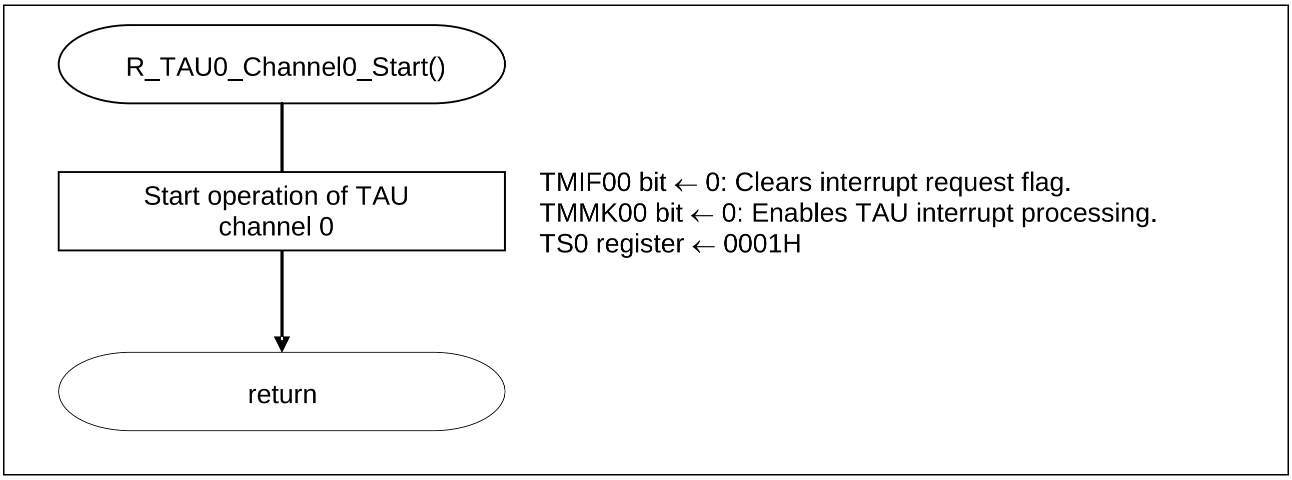

R_TAU0_Channel0_Start()
Start operation of TAU channel 0
TMIF00 bit ← 0: Clears interrupt request flag.
TMMK00 bit ← 0: Enables TAU interrupt processing.
TS0 register ← 0001H
return

Figure 4.14 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).

4.4.7.11 SAU0 Channel 0 Startup

Figure 4.15 shows the flowchart for starting the operation of SAU0 channel 0 (CSI00).

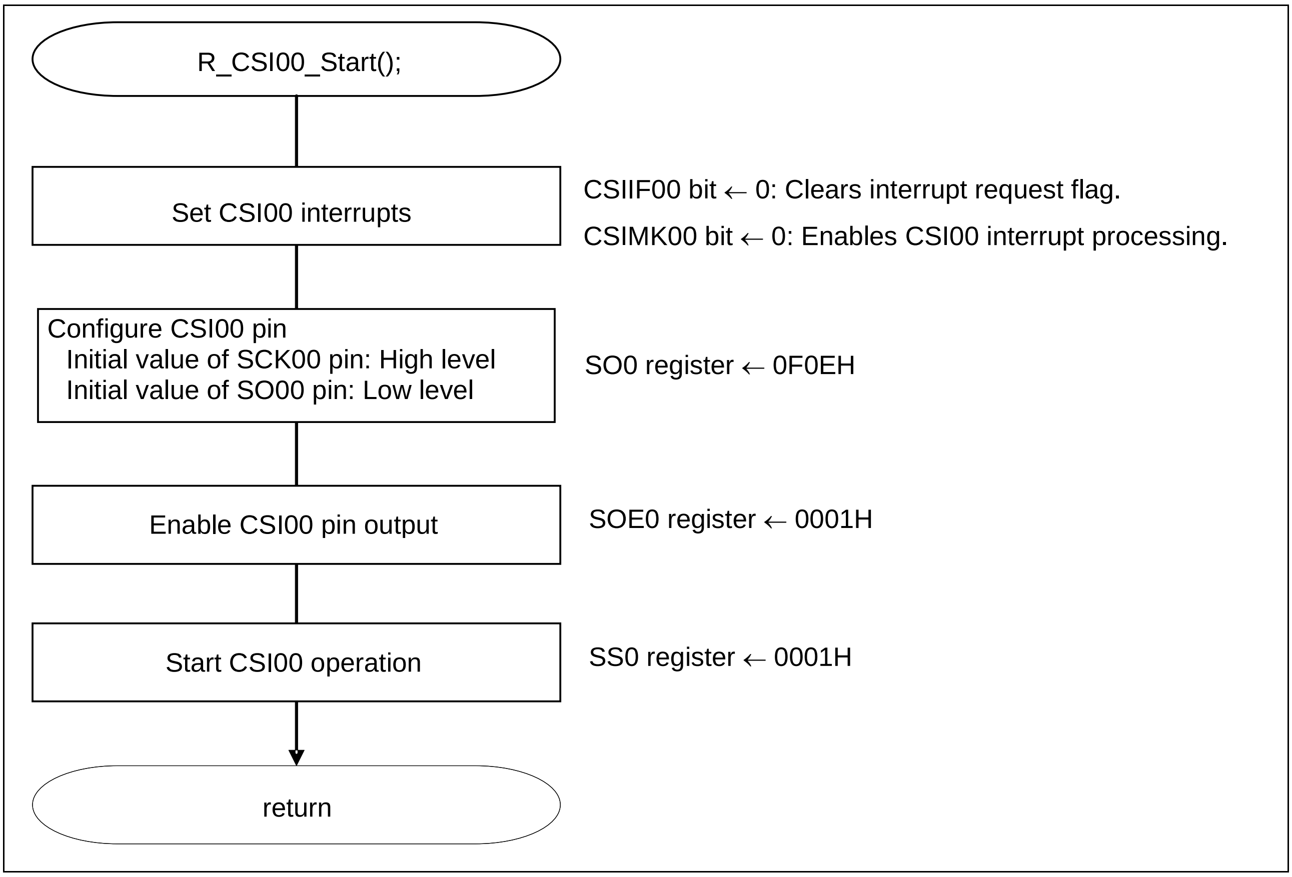

R_CSI00_Start();
Set CSI00 interrupts
CSIIF00 bit ← 0: Clears interrupt request flag.
CSIMK00 bit ← 0: Enables CSI00 interrupt processing.
Configure CSI00 pin
Initial value of SCK00 pin: High level
Initial value of SO00 pin: Low level
SO0 register ← 0F0EH
Enable CSI00 pin output
SOE0 register ← 0001H
Start CSI00 operation
SS0 register ← 0001H
return

Figure 4.15 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.

4.4.7.12 Infinite Loop in Main Processing

Figure 4.16 shows the flowchart for an infinite loop in the main processing.

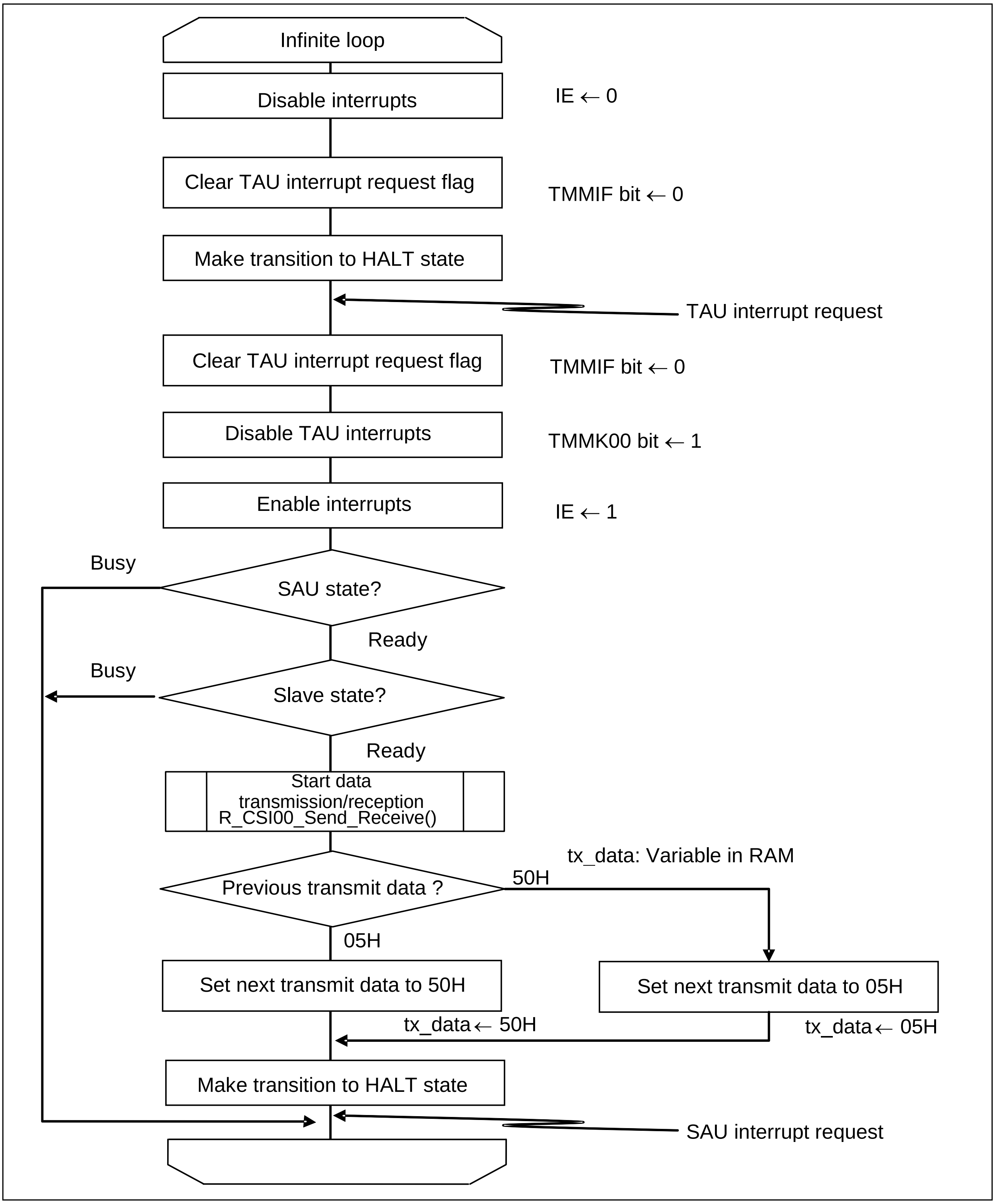


Figure 4.16 Infinite Loop in Main Processing

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.

4.4.7.13 CSI00 Data Transmission/Reception Start

Figure 4.17 shows the flowchart for starting CSI00 data transmission/reception.

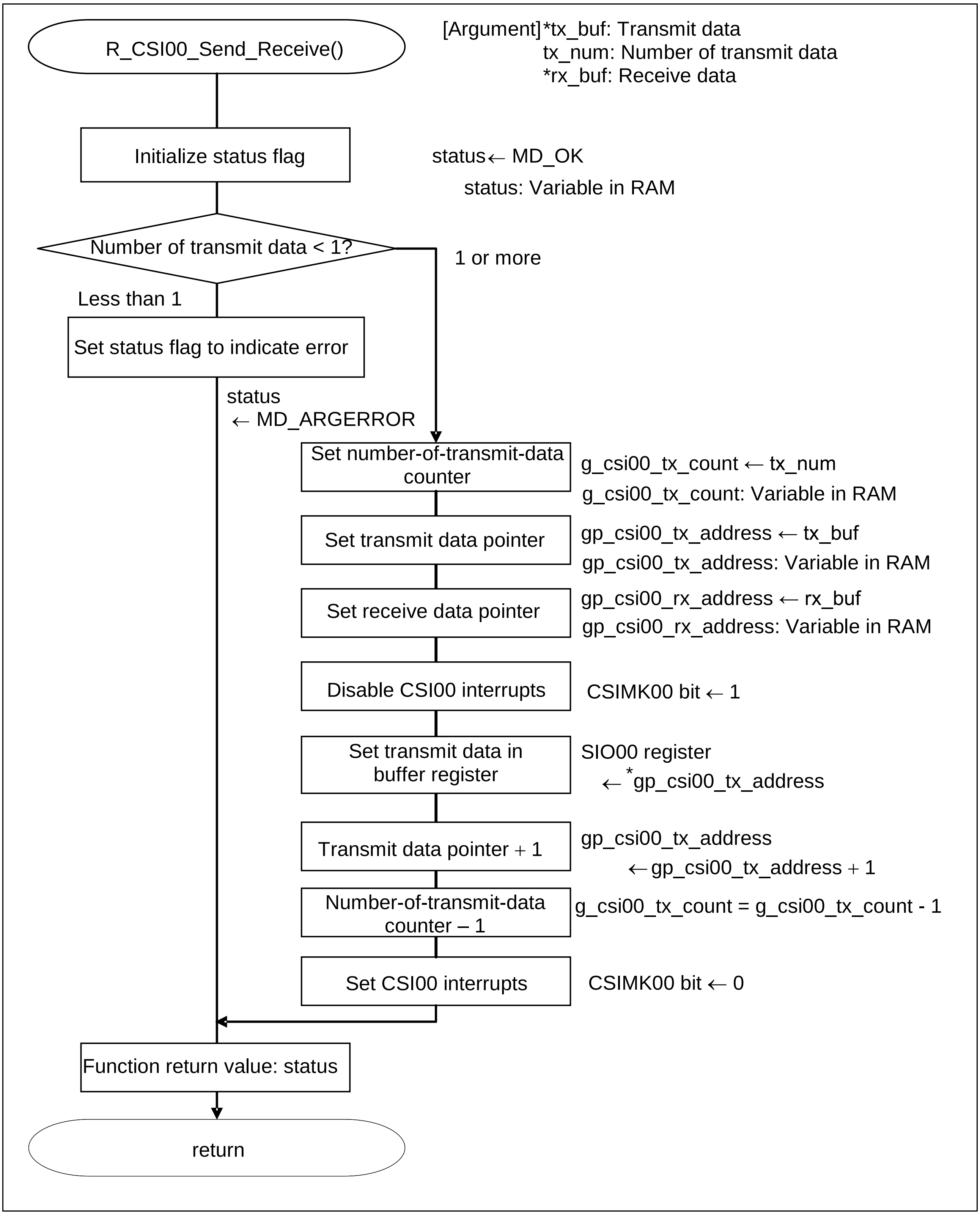
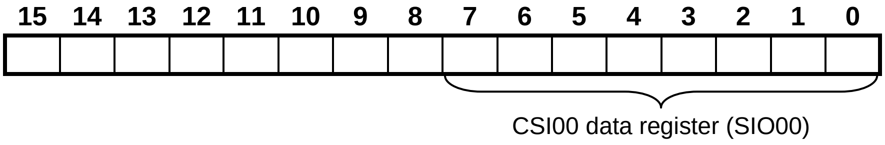

R_CSI00_Send_Receive()
[Argument]*tx_buf: Transmit data
tx_num: Number of transmit data
*rx_buf: Receive data
Initialize status flag
status← MD_OK
status: Variable in RAM
Number of transmit data < 1?
1 or more
Less than 1
Set status flag to indicate error
status
← MD_ARGERROR
Set number-of-transmit-data counter
g_csi00_tx_count ← tx_num
g_csi00_tx_count: Variable in RAM
Set transmit data pointer
gp_csi00_tx_address ← tx_buf
gp_csi00_tx_address: Variable in RAM
Set receive data pointer
gp_csi00_rx_address ← rx_buf
gp_csi00_rx_address: Variable in RAM
Disable CSI00 interrupts
CSIMK00 bit ← 1
Set transmit data in buffer register
SIO00 register
← *gp_csi00_tx_address
Transmit data pointer + 1
gp_csi00_tx_address
←gp_csi00_tx_address + 1
Number-of-transmit-data counter – 1
g_csi00_tx_count = g_csi00_tx_count - 1
Set CSI00 interrupts
CSIMK00 bit ← 0
Function return value: status
return

Figure 4.17 CSI00 Data Transmission/Reception Start

Setting transmit data

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

Symbol: SDR00



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.

4.4.7.14 CSI00 Transfer End Interrupt Processing

Figure 4.18 shows the flowchart for CSI00 transfer end interrupt processing.

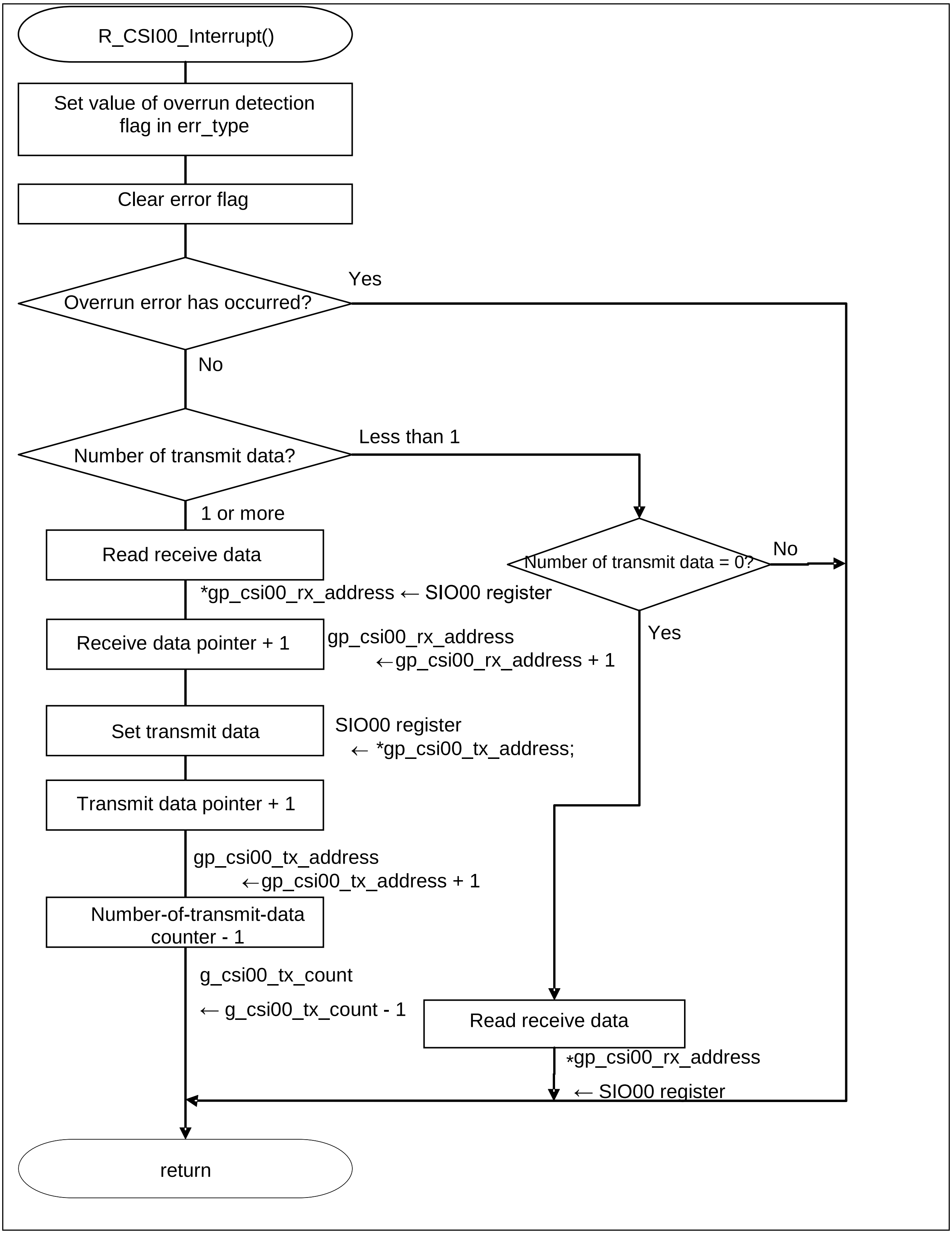

R_CSI00_Interrupt()
Set value of overrun detection flag in err_type
Clear error flag
Overrun error has occurred?
Yes
No
Number of transmit data?
Less than 1
1 or more
Read receive data
*gp_csi00_rx_address ← SIO00 register
Receive data pointer + 1
gp_csi00_rx_address ←gp_csi00_rx_address + 1
Set transmit data
SIO00 register ← *gp_csi00_tx_address;
Transmit data pointer + 1
gp_csi00_tx_address ←gp_csi00_tx_address + 1
Number-of-transmit-data counter - 1
g_csi00_tx_count ← g_csi00_tx_count - 1
Number of transmit data = 0?
No
Yes
Read receive data
*gp_csi00_rx_address ← SIO00 register
return

Figure 4.18 CSI00 Transfer End Interrupt Processing

4.4.8 Sample Code

The sample code is available on the Renesas Electronics Website.

4.4.9 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 (Slave Transmission/Reception) (R01AN2711E) Application Note

4.4.10 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.

5. Example of Migration from Clock Asynchronous Serial I/O (UART) Mode

5.1 Specifications

In this application note, UART communication is performed through the serial array unit (SAU). ASCII characters transmitted from the device on the opposite side are analyzed to make responses.

Table 5.1 shows the peripheral function to be used and its use. Figure 5.1 and Figure 5.2 illustrate UART communication operation.

Table 5.1 Peripheral Function to be Used and its Use

Peripheral Function	Use
Serial array unit 0	Perform UART communication using the TxD0 pin (transmission) and the RxD0 pin (reception).

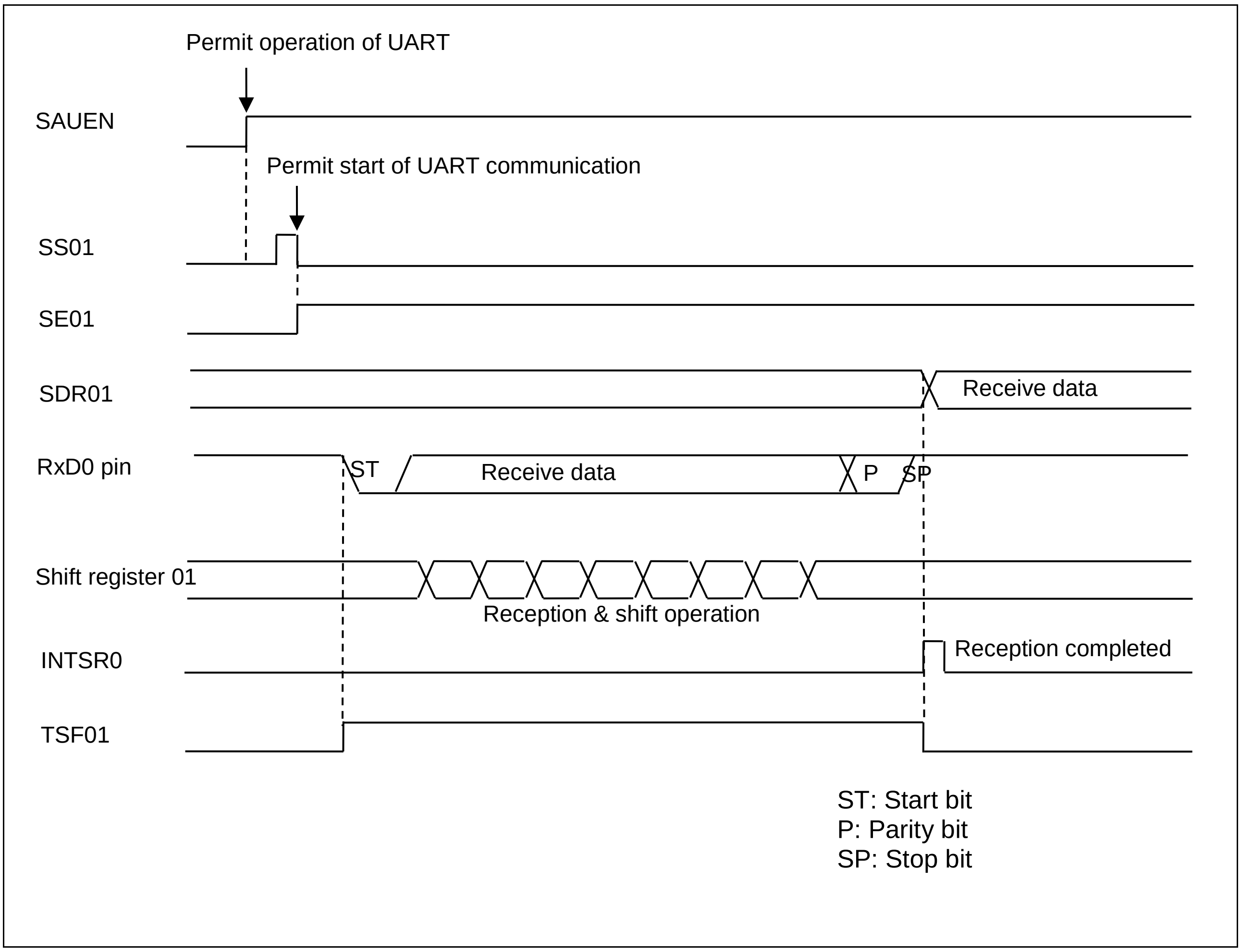


Figure 5.1 UART Reception Timing Chart (1/2)

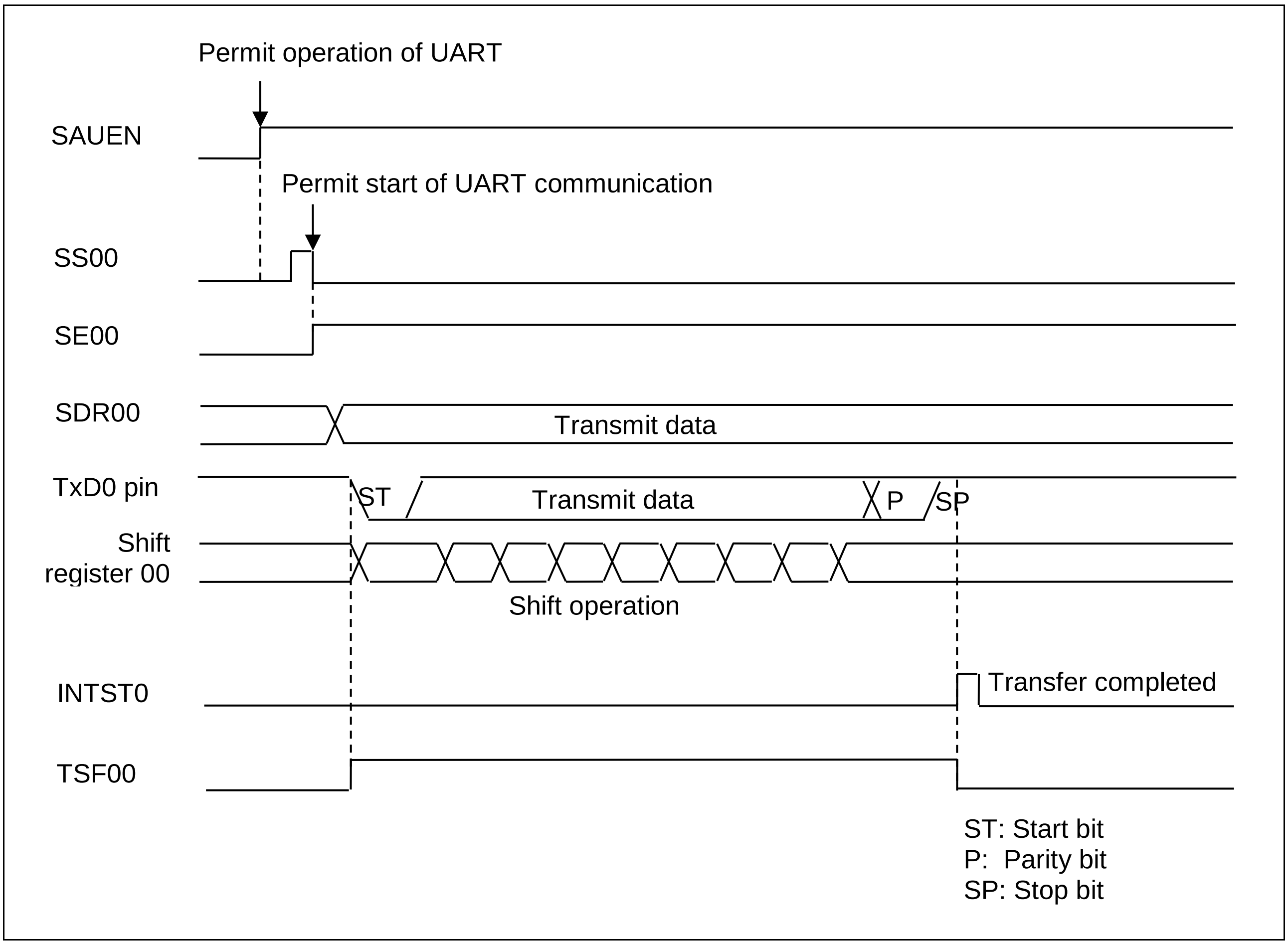


Figure 5.2 UART Transmission Timing Chart (2/2)

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 (can run on a voltage range of 2.9 V to 5.5 V.) LVD operation (V_{LVD}): Reset mode 2.81 V (2.76 V to 2.87 V)
Integrated development environment (CS+)	Renesas Electronics Corporation CS+ for CC V5.00.00
C compiler (CS+)	Renesas Electronics Corporation CC-RL V1.04.00
Integrated development environment (e^2 studio)	Renesas Electronics Corporation e^2 studio V5.4.0.018
C compiler (e^2 studio)	Renesas Electronics Corporation CC-RL V1.04.00

5.3 Description of the Hardware

5.3.1 Hardware Configuration Example

Figure 5.3 shows an example of hardware configuration that is used for this application note.

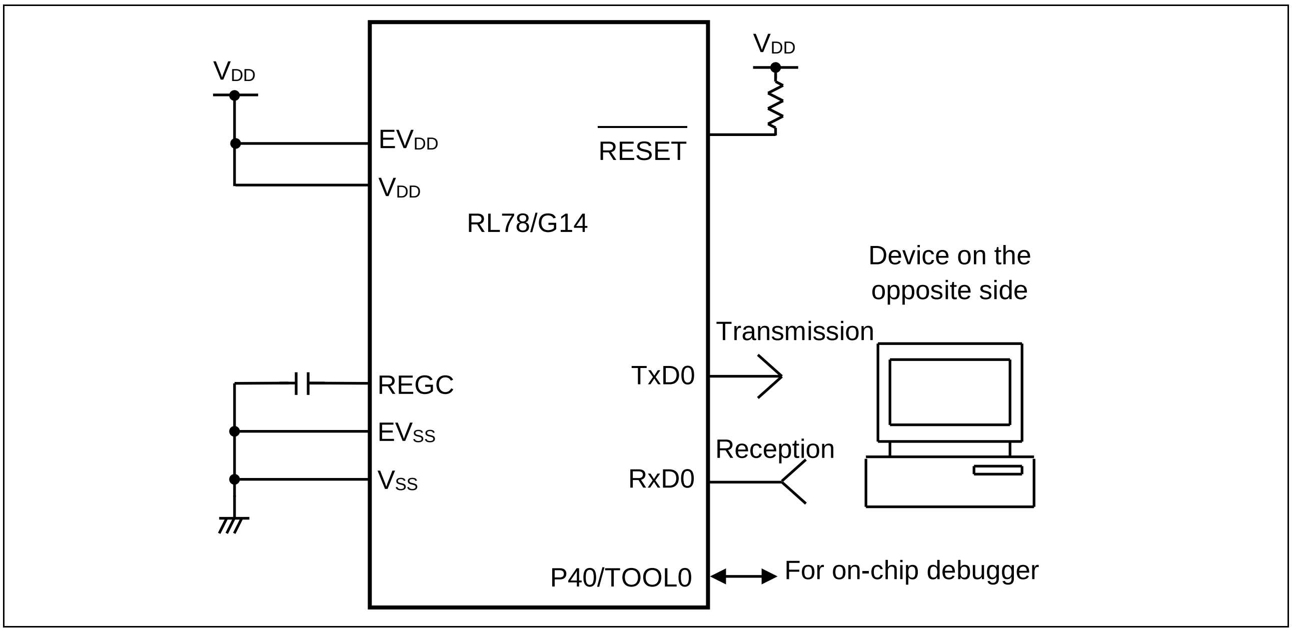


Figure 5.3 Hardware Configuration

Caution:

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.

5.3.2 List of Pins to be Used

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

Table 5.3 Pins to be Used and their Functions

Pin Name	I/O	Description
P51/SO00/TxD0/TOOLTxD	**Output**	Data transmission pin
P50/SI00/RxD0/TOOLRxD/SDA00	**Input**	Data reception pin

5.4 Description of the Software

5.4.1 Operation Outline

This sample code transmits, to the device on the opposite side, the data corresponding to that received from the device. If an error occurs, it transmits to the device the data corresponding to the error. Table 5.4 and Table 5.5 show the correspondence between transmit data and receive data.

Table 5.4 Correspondence between Receive Data and Transmit Data

Receive Data	Response (Transmit) Data
T (54H)	O (4FH), K (4BH), ”CR” (0DH), ”LF” (0AH)
t (74H)	o (6FH), k (6BH), ”CR” (0DH), ”LF” (0AH)
Other than above	U (55H), C (43H), ”CR” (0DH), ”LF” (0AH)

Table 5.5 Correspondence between Error and Transmit Data

Error	Response (Transmit) Data
Parity error	P (50H), E (45H), ”CR” (0DH), ”LF” (0AH)
Framing error	F (46H), E (45H), ”CR” (0DH), ”LF” (0AH)
Overrun error	O (4FH), E (45H), ”CR” (0DH), ”LF” (0AH)

(1) Perform initial setting of UART.

<UART Setting Conditions>

- Use SAU0 channels 0 and 1 as UART.
- Use the P12/TxD0 pin and the P11/RxD0 pin for data output and data input, respectively.
- The data length is 8 bits.
- Set the data transfer direction to LSB first.
- Use even parity as the parity setting.
- Set the receive data level to standard.
- Set the transfer rate to 9600 bps.
- Use reception end interrupt (INTSR0), transmission end interrupt (INTST0), and error interrupt (INTSRE0).
- Select interrupt priority level 2 or 1 for INTSR0 and for INTSRE0. Select the low interrupt priority level (level 3) for INTST0.

(2) After the system is made to enter a UART communication wait state by using the serial channel start register, a HALT instruction is executed. Processing is performed in response to reception end interrupt (INTSR0) and error interrupt (INTSRE0).

- When an INTSR0 occurs, the received data is taken in and the data corresponding to the received data is transmitted. When an INTSRE0 occurs, error handling is performed to transmit the data corresponding to the error.
- After data transmission, a HALT instruction is executed again to wait for reception end interrupt (INTSR0) and error interrupt (INTSRE0).

5.4.2 List of Option Byte Settings

Table 5.6 summarizes the settings of the option bytes.

Table 5.6 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 5.7 lists the constants that are used in this sample program.

Table 5.7 Constants for the Sample Program

Constant	Setting	Description
g_messageOK[4]	**"OK\r\n"**	Response message to reception of “T”.
g_messageok[4]	**"ok\r\n"**	Response message to reception of “t”.
g_messageUC[4]	**"UC\r\n"**	Response message to reception of characters other than “T” or “t”.
g_messageFE[4]	**"FE\r\n"**	Response message to a framing error.
g_messagePE[4]	**"PE\r\n"**	Response message to a parity error.
g_messageOE[4]	**"OE\r\n"**	Response message to an overrun error.

5.4.4 List of Variables

Table 5.8 lists the global variable that is used by this sample program.

Table 5.8 Global Variable

Type	Variable Name	Contents	Function Used
uint8_t	g_uart0_rx_buffer	Receive data buffer	main()
uint8_t	gp_uart0_tx_address	Transmit data pointer	R_UART0_Send(), R_UART0_Interrupt_Send()
uint16_t	g_uart0_tx_count	Transmit data number counter	R_UART0_Send(), R_UART0_Interrupt_Send()
uint8_t	gp_uart0_rx_address	Receive data pointer	R_UART0_Receive(), R_UART0_Interrupt_Receive(), R_UART0_Interrupt_Error()
uint16_t	g_uart0_rx_ count	Receive data number counter	R_UART0_Receive(), R_UART0_Interrupt_Receive()
uint16_t	g_uart0_rx_length	Receive data number	R_UART0_Receive(), R_UART0_Interrupt_Receive()
MD_STATUS	g_uart0_tx_end	Transmit status	main(), r_uart0_callback_sendend()
unit8_t	g_uart0_rx_error	Receive error status	main(), r_uart0_callback_receiveend(), r_uart0_callback_error()

5.4.5 List of Functions

Table 5.9 lists the functions that are used in this sample program.

Table 5.9 Functions

Function Name	Outline
R_UART0_Start	UART0 operation start
R_UART0_Receive	UART0 reception status initialization function
R_UART0_Send	UART0 data transmission function
r_uart0_interrupt_receive	UART0 reception end interrupt handling
r_uart0_callback_receiveend	UART0 receive data classification function
r_uart0_interrupt_error	UART0 error interrupt handling
r_uart0_callback_error	UART0 reception error classification function
r_uart0_interrupt_send	UART0 transmission end interrupt handling
r_uart0_callback_sendend	UART0 transmission end processing function
r_uart0_callback_softwareoverrun	UART0 overflow data receive function

5.4.6 Function Specifications

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

[Function Name]R_ UART0_Start

Synopsis UART0 operation start

Header r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h

Declaration void R_ UART0_Start(void)

Explanation Starts operation of channel 0 of serial array units 0 and 1 to make the system enter a communication wait state.

Arguments None

Return value None

Remarks None

[Function Name] R_UART0_Receive

Synopsis UART0 reception status initialization function

Header r_cg_macrodriver.h, r_cg_serial.h, r_cg_userdefine.h

Declaration MD_STATUS R_UART0_Receive(uint8_t *rx_buf, uint16_t rx_num)

Explanation Makes initial setting for UART0 reception.

Arguments	uint8_t *rx_buf	: [Receive data buffer address]
	uint16_t rx_num	: [Receive data buffer size]

Return value [MD_OK]: Reception setting is completed
[MD_ARGERROR]: Reception setting failed

Remarks None

[Function Name] R_UART0_Send

Synopsis	UART0 data transmission function	
Header	r_cg_macrodriver.h, r_cg_serial.h, r_cg_userdefine.h	
Declaration	MD_STATUS R_UART0_Send(uint8_t* tx_buf, uint16_t tx_num)	
Explanation	Makes initial setting for UART0 transmission, and starts data transmission.	
Arguments	uint8_t *tx_buf	: [Transmit data buffer address]
	uint16_t tx_num	: [Transmit data buffer size]
Return value	[MD_OK]: Transmission setting is completed	
	[MD_ARGERROR]: Transmission setting failed	
Remarks	None	

[Function Name] r_uart0_interrupt_receive

Synopsis	UART0 reception end interrupt handling
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	static void __near r_uart0_interrupt_receive(void)
Explanation	Makes a response (data transmission) corresponding to received data.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_uart0_interrupt_erro

Synopsis	UART error interrupt function
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	static void __near r_uart0_interrupt_error(void)
Explanation	Transmits the data corresponding to a detected error.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_uart0_callback_receiveend

Synopsis	UART0 receive data classification function
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	static void r_uart0_callback_receiveend(void)
Explanation	Clears the reception error flag.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_uart0_callback_error

Synopsis	UART0 reception error classification function	
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h	
Declaration	static void r_uart0_callback_error(uint8_t err_type)	
Explanation	Makes flag setting for transmission of the data corresponding to an error.	
Arguments	err_type	: Error type
Return value	None	
Remarks	None	

[Function Name] r_uart0_interrupt_send

Synopsis	UART0 transmission end interrupt handling
Header	r_cg_macrodriver.h, r_cg_serial.h, and r_cg_userdefine.h
Declaration	static void __near r_uart0_interrupt_send(void)
Explanation	Transmits a specified number of pieces of data.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_uart0_callback_sendend

Synopsis	UART0 transmission end processing function
Header	r_cg_macrodriver.h, r_cg_serial.h, r_cg_userdefine.h
Declaration	static void r_uart0_callback_sendend(void)
Explanation	Makes transmission end flag setting.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_uart0_callback_softwareoverrun

Synopsis	UART0 overflow data receive function
Header	r_cg_macrodriver.h, r_cg_serial.h, r_cg_userdefine.h
Declaration	static void r_uart0_callback_softwareoverrun(void)
Explanation	Executes when detected overflow of data by software.
Arguments	None
Return value	None
Remarks	Unused function

5.4.7 Flowcharts

5.4.7.1 Overall Flow

Figure 5.4 shows the overall flow of the sample program described in this application note.

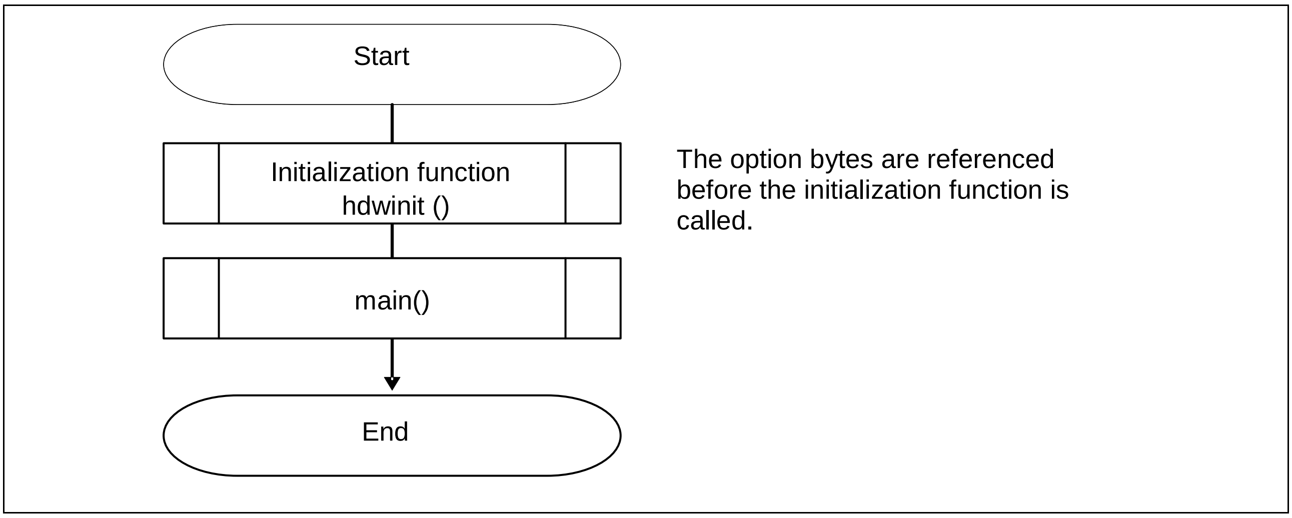

Start
Initialization function
hdwinit ()
main()
End
The option bytes are referenced before the initialization function is called.

Figure 5.4 Overall Flow

Note: Startup routine is executed before and after the initialization function.

5.4.7.2 Initialization Function

Figure 5.5 shows the flowchart for the initialization function.

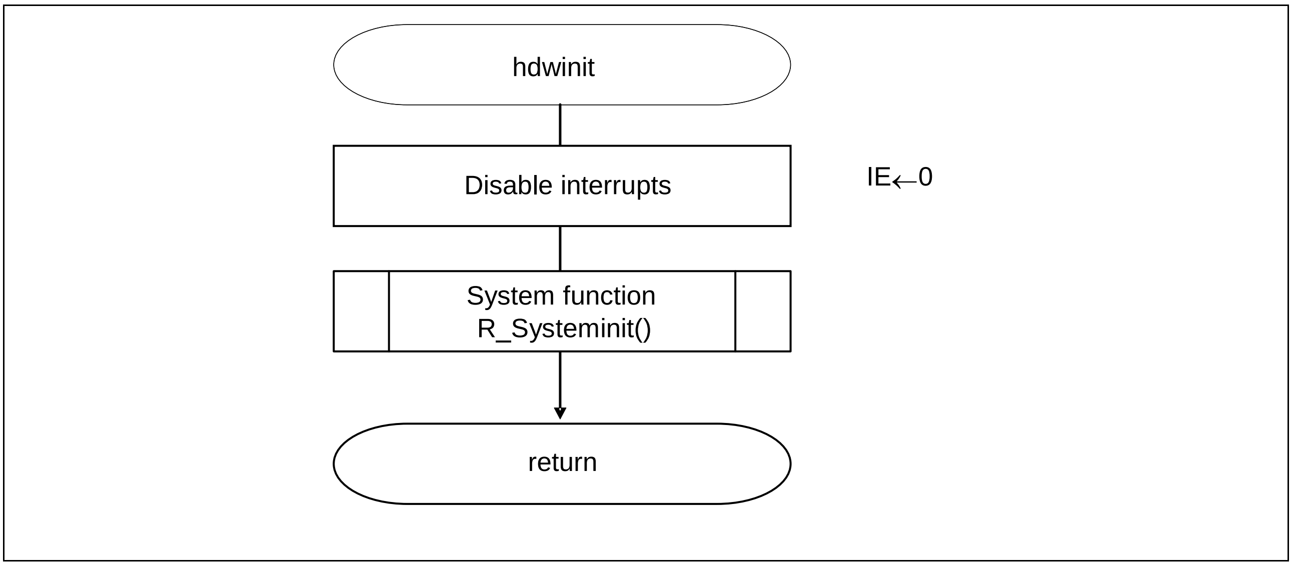


Figure 5.5 Initialization Function

5.4.7.3 System Function

Figure 5.6 shows the flowchart for the system function.

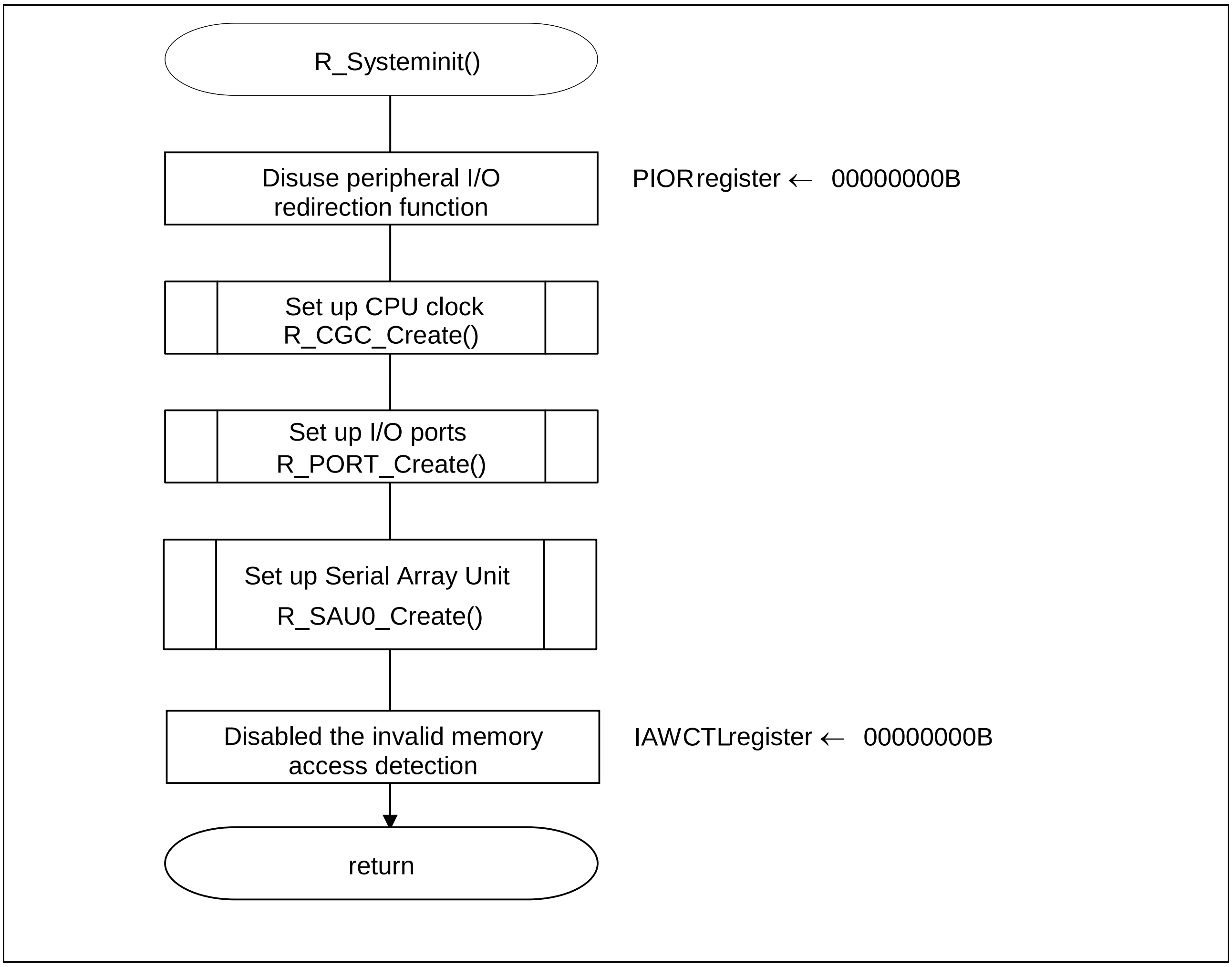

R_Systeminit()
Disuse peripheral I/O redirection function
PIOR register ← 00000000B
Set up CPU clock R_CGC_Create()
Set up I/O ports R_PORT_Create()
Set up Serial Array Unit R_SAU0_Create()
Disabled the invalid memory access detection
IAWCTLregister ← 00000000B
return

Figure 5.6 System Function

5.4.7.4 I/O Port Setup

Figure 5.7 shows the flowchart for setting up the I/O ports.

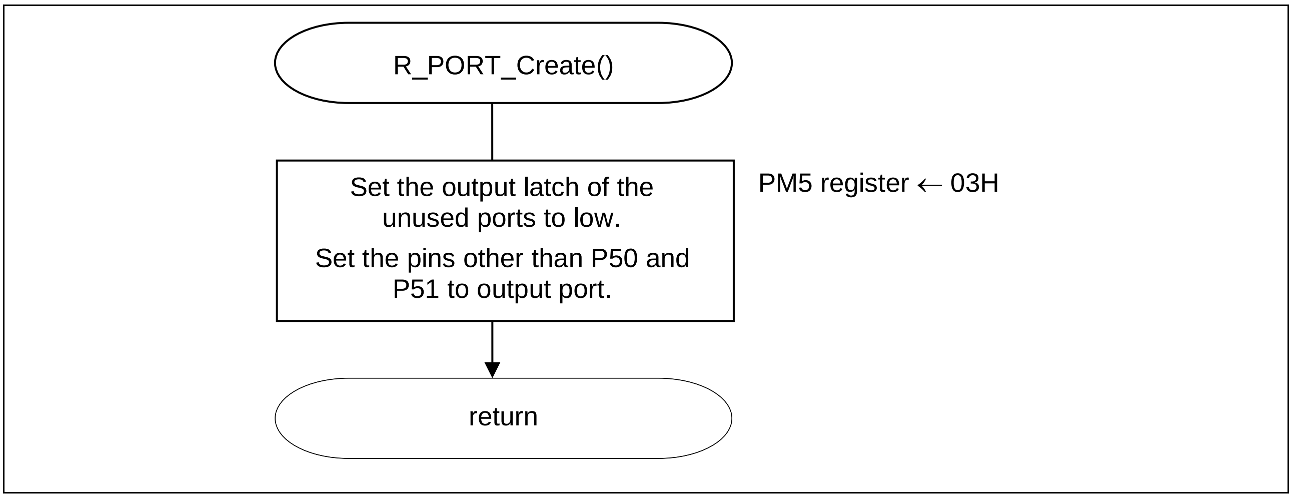

R_PORT_Create()
Set the output latch of the unused ports to low.
Set the pins other than P50 and P51 to output port.
PM5 register ← 03H
return

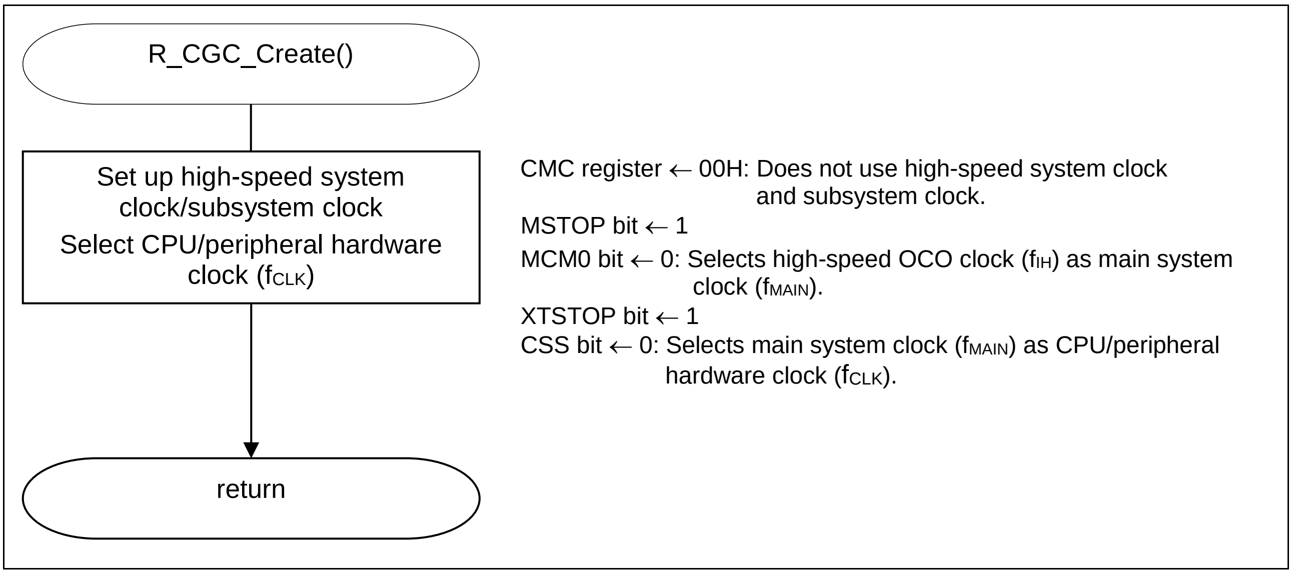
Figure 5.7 I/O Port Setup

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.

5.4.7.5 CPU Clock Setup

Figure 5.8 shows the flowchart for setting up the CPU clock.


R_CGC_Create()
Set up high-speed system clock/subsystem clock
Select CPU/peripheral hardware clock (f_{CLK})
return

CMC register ← 00H: Does not use high-speed system clock and subsystem clock.

MSTOP bit ← 1

MCM0 bit ← 0: Selects high-speed OCO clock (f_{IH}) as main system clock (f_{MAIN}).

XTSTOP bit ← 1

CSS bit ← 0: Selects main system clock (f_{MAIN}) as CPU/peripheral hardware clock (f_{CLK}).

Figure 5.8 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).

5.4.7.6 Serial Array Unit Setup

Figure 5.9 shows the flowchart for setting up the serial array unit.

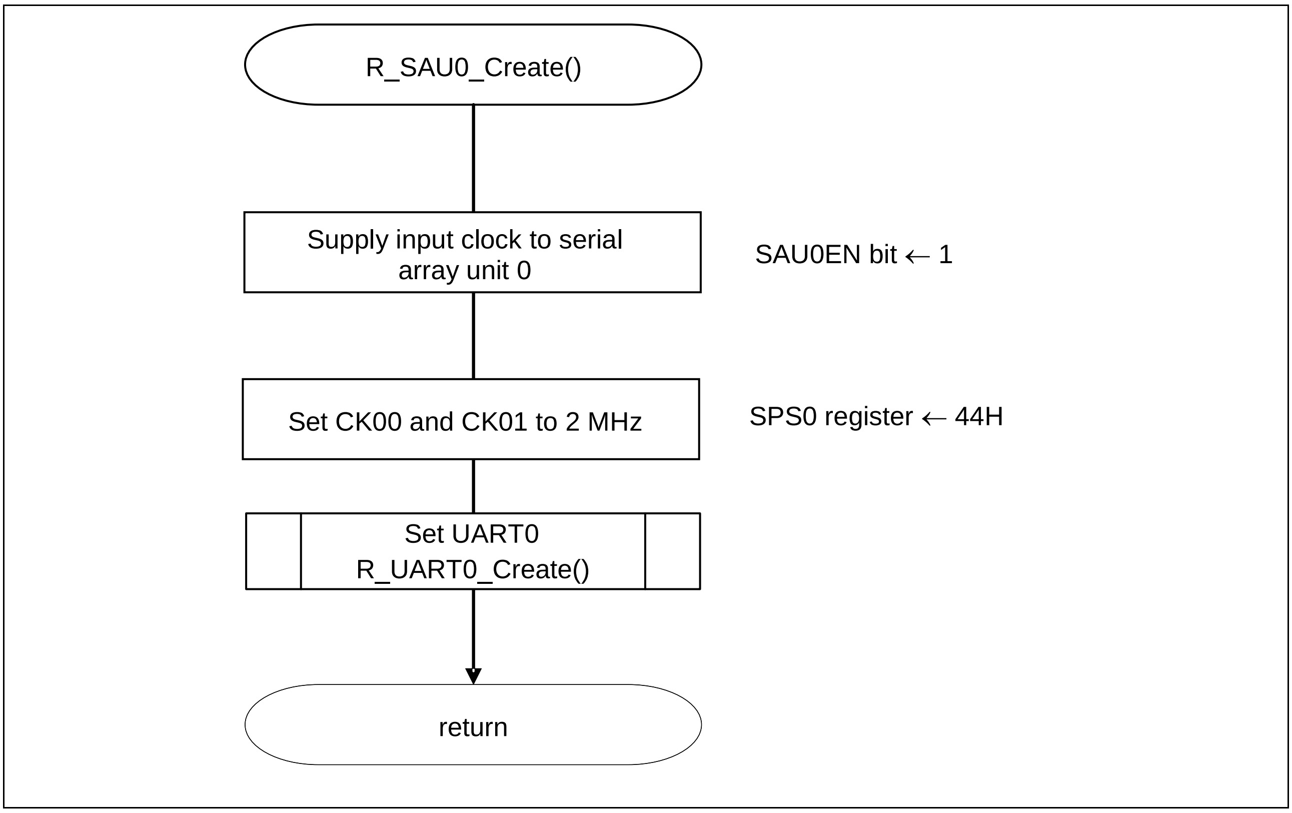


Figure 5.9 Serial Array Unit Setup

Start supplying clock to the SAU

- Peripheral enable register 0 (PER0)
  Clock supply

Symbol: PER0

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

Bit 2

SAU0EN	Input clock control for serial array unit 0
0	Stops supply of input clock.
1	**Starts supply of input clock.**

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

Select serial clock

- Serial clock select register 0 (SPS0)
  Operation clock setting

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	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**

Bits 7 to 0

PRS 0n3	PRS 0n2	PRS 0n1	PRS 0n0	Operation clock (CK00) selection (n = 0, 1)					
					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**	**312.5 kHz**	**625 kHz**	**1.25 MHz**	**2 MHz**
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.62 kHz	39.1 kHz	78.1 kHz	156.2 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	9.76 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz	31.25 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.76 kHz	15.6 kHz
Other than above.				Setting prohibited.					

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

5.4.7.7 UART0 Setup

Figure 5.10, Figure 5.11, and Figure 5.12 show the flowcharts for setting up UART0.

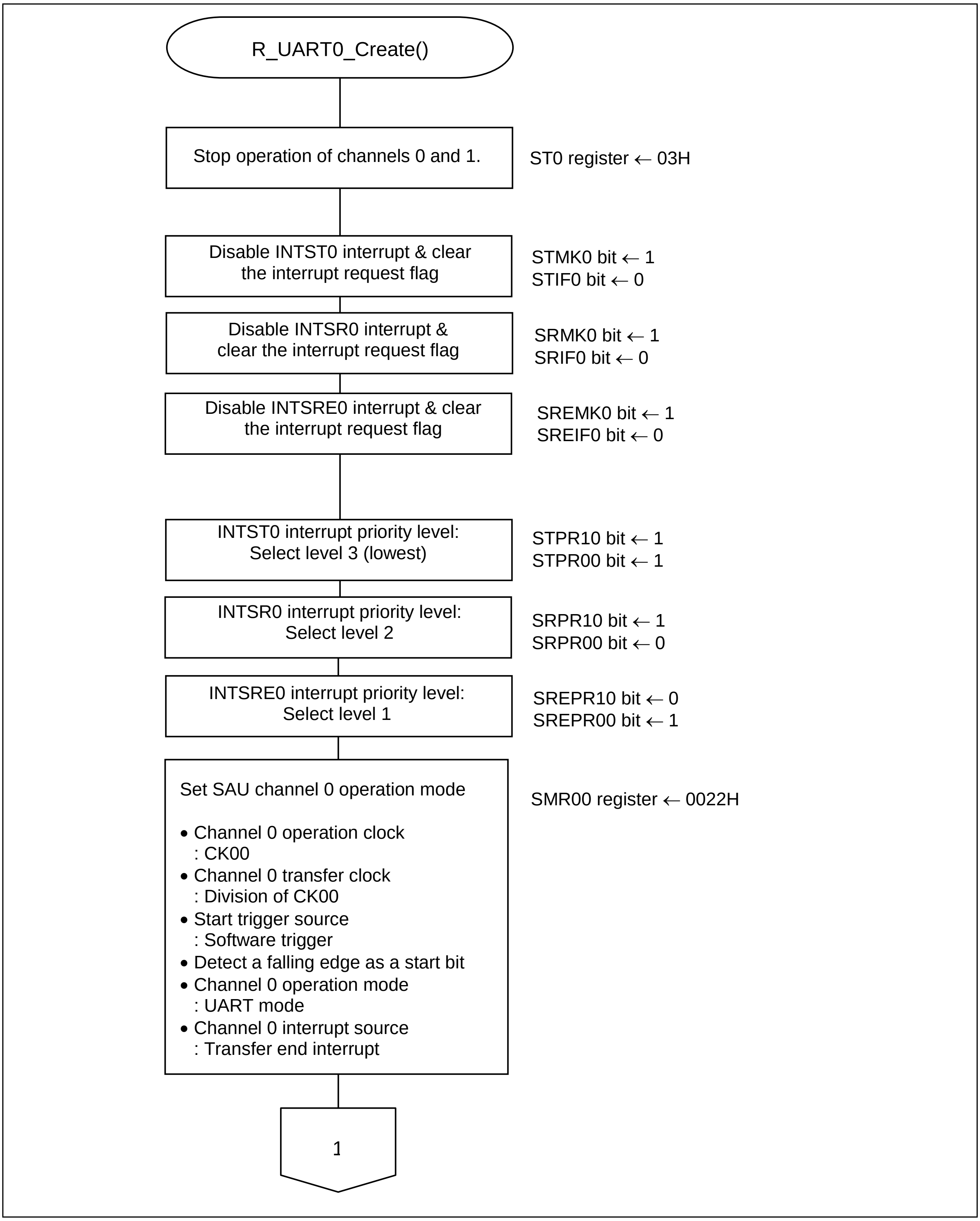

R_UART0_Create()
Stop operation of channels 0 and 1.
ST0 register ← 03H
Disable INTST0 interrupt & clear the interrupt request flag
STMK0 bit ← 1
STIF0 bit ← 0
Disable INTSR0 interrupt & clear the interrupt request flag
SRMK0 bit ← 1
SRIF0 bit ← 0
Disable INTSRE0 interrupt & clear the interrupt request flag
SREMK0 bit ← 1
SREIF0 bit ← 0
INTST0 interrupt priority level: Select level 3 (lowest)
STPR10 bit ← 1
STPR00 bit ← 1
INTSR0 interrupt priority level: Select level 2
SRPR10 bit ← 1
SRPR00 bit ← 0
INTSRE0 interrupt priority level: Select level 1
SREPR10 bit ← 0
SREPR00 bit ← 1
Set SAU channel 0 operation mode
• Channel 0 operation clock : CK00
• Channel 0 transfer clock : Division of CK00
• Start trigger source : Software trigger
• Detect a falling edge as a start bit
• Channel 0 operation mode : UART mode
• Channel 0 interrupt source : Transfer end interrupt
SMR00 register ← 0022H
1

Figure 5.10 UART0 Setup (1/3)

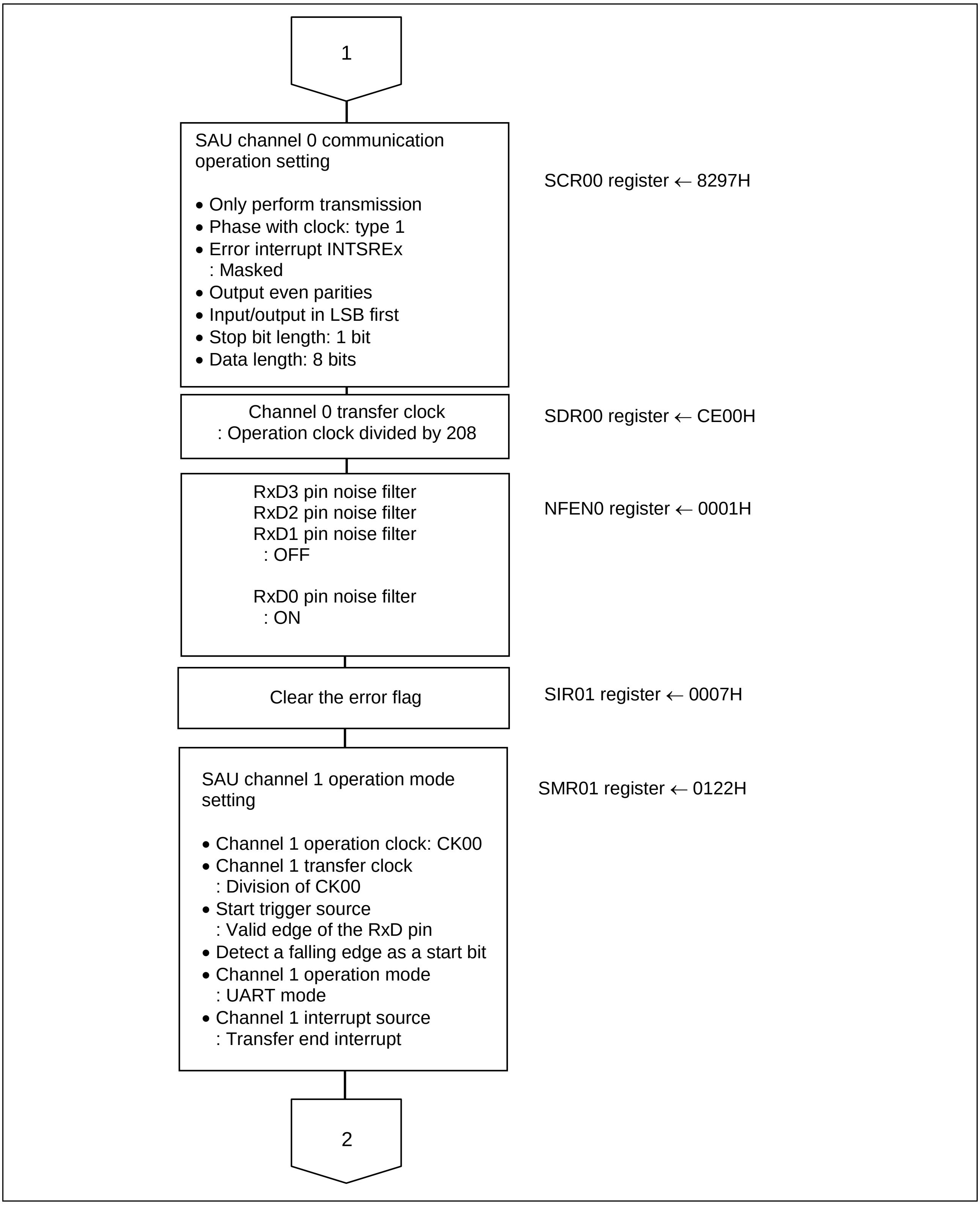

1
SAU channel 0 communication operation setting
• Only perform transmission
• Phase with clock: type 1
• Error interrupt INTSREx : Masked
• Output even parities
• Input/output in LSB first
• Stop bit length: 1 bit
• Data length: 8 bits
SCR00 register ← 8297H
Channel 0 transfer clock : Operation clock divided by 208
SDR00 register ← CE00H
RxD3 pin noise filter
RxD2 pin noise filter
RxD1 pin noise filter
: OFF
RxD0 pin noise filter
: ON
NFEN0 register ← 0001H
Clear the error flag
SIR01 register ← 0007H
SAU channel 1 operation mode setting
• Channel 1 operation clock: CK00
• Channel 1 transfer clock : Division of CK00
• Start trigger source : Valid edge of the RxD pin
• Detect a falling edge as a start bit
• Channel 1 operation mode : UART mode
• Channel 1 interrupt source : Transfer end interrupt
SMR01 register ← 0122H
2

Figure 5.11 UART0 Setup (2/3)

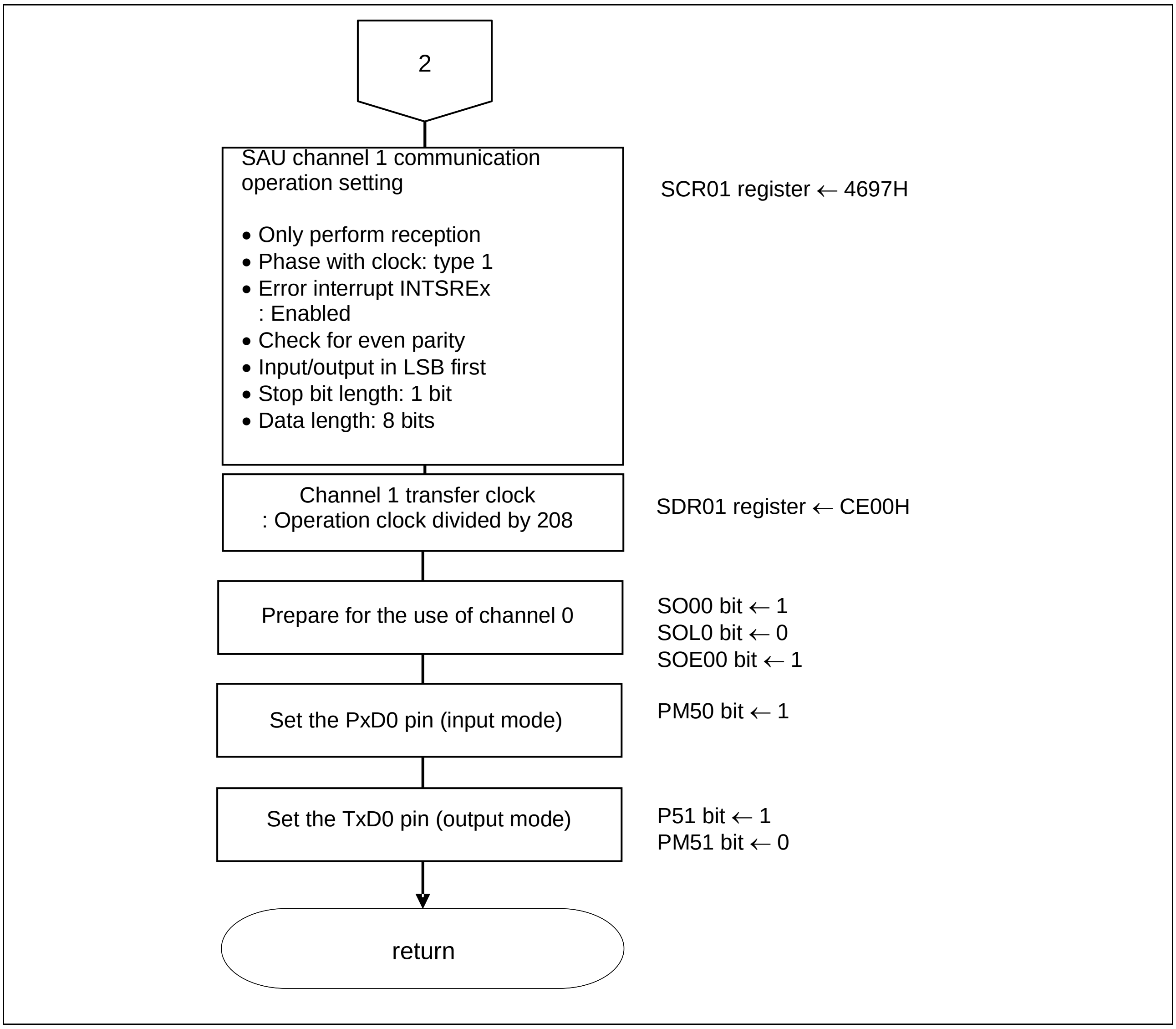

2
SAU channel 1 communication operation setting
• Only perform reception
• Phase with clock: type 1
• Error interrupt INTSREx : Enabled
• Check for even parity
• Input/output in LSB first
• Stop bit length: 1 bit
• Data length: 8 bits
SCR01 register ← 4697H
Channel 1 transfer clock : Operation clock divided by 208
SDR01 register ← CE00H
Prepare for the use of channel 0
SO00 bit ← 1
SOL0 bit ← 0
SOE00 bit ← 1
Set the PxD0 pin (input mode)
PM50 bit ← 1
Set the TxD0 pin (output mode)
P51 bit ← 1
PM51 bit ← 0
return

Figure 5.12 UART0 Setup (3/3)

Transmission channel operation mode setting

- Serial mode register 00 (SMR00)
  Interrupt source
  Operation mode
  Transfer clock selection
  fMCK selection

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	0	0	0	1	0	0	MD 002	MD 001	MD 000
0	**0**	0	0	0	0	0	0	0	0	1	0	0	**0**	**1**	**0**

Bit 15

CKS00	Channel 0 operation clock (f_{MCK}) selection
0	**Prescaler output clock CK00 configured by the SPS0 register**
1	Prescaler output clock CK01 configured by the SPS0 register

Bit 14

CCS00	Channel 0 transfer clock (TCLK) selection
0	**Clock obtained by dividing the operation clock f_{MCK} specified by the CKS00 bit.**
1	Clock input from the SCK pin.

Bits 2 and 1

MD002	MD001	Channel 0 operation mode setting
0	0	CSI mode
0	**1**	**UART mode**
1	0	Simplified I^{2}C mode
1	1	Setting prohibited

Bit 0

MD000	Channel 0 interrupt source selection
0	**Transfer end interrupt**
1	Buffer empty interrupt

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

Transmission channel communication operation setting

- Serial communication operation setting register 00 (SCR00)
  Data length setting, data transfer order, error interrupt signal mask availability, and operation mode

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	**0**	**0**	**0**	0	**0**	**1**	**0**	**1**	0	**0**	**1**	0	1	**1**	**1**

Bits 15 and 14

TXE00	RXE00	Channel 0 operation mode setting
0	0	Communication prohibited
0	1	Reception Only
1	**0**	**Transmission only**
1	1	Both transmission and reception

Bit 10

EOC00	Error interrupt signal (INTSREx (x = 0, 1)) mask availability selection
0	**Error interrupt INTSREx is masked**
1	Generation of error interrupt INTSREx is enabled

Bits 9 and 8

PTC001	PTC000	Parity bit setting in UART mode	
		Transmission	Reception
0	0	No parity bit is output	Data is received without parity
0	1	0 parity is output	No parity check is made
1	**0**	**Even parity is output**	Check is made for even parity
1	1	Odd parity is output	Check is made for odd parity

Bit 7

DIR00	Selection of data transfer order in CSI and UART modes
0	Input and output in MSB first
1	**Input and output in LSB first**

Bits 5 and 4

SLC001	SLC000	Stop bit setting in UART mode
0	0	No stop bit
0	**1**	**Stop bit length = 1 bit**
1	0	Stop bit length = 2 bits
1	1	Setting prohibited

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	**0**	**0**	**0**	0	**0**	**1**	**0**	**1**	0	**0**	**1**	0	1	**1**	**1**

Bits 1 and 0

DLS001	DLS000	Data length setting in CSI mode
0	1	9-bit data length
1	0	7-bit data length
1	**1**	**8-bit data length**
Others		Setting prohibited

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

Transmission channel transfer clock setting

- Serial data register 00 (SDR00)
  Transfer clock frequency: fMCK/208 (≈ 9600 Hz)

Symbol: SDR00

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

Bits 15 to 9

SDR00[15:9]							Transfer clock setting by dividing 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
. .	. .	. .	. .	. .	. .	. .	. .
1	**1**	**0**	**0**	**1**	**1**	**1**	**f_{MCK} /208**
. .	. .	. .	. .	. .	. .	. .	. .
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.

Reception channel operation mode setting

- Serial mode register 01 (SMR01)
  Interrupt source
  Operation mode
  Transfer clock selection
  fMCK selection

Symbol: SMR01

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS 01	CCS 01	0	0	0	0	0	STS 01	0	SIS 010	1	0	0	MD 012	MD 011	MD 010
0	**0**	0	0	0	0	0	**1**	0	**0**	1	0	0	**0**	**1**	**0**

Bit 15

CKS01	Channel 1 operation clock (f_{MCK}) selection
0	**Prescaler output clock CK00 configured by the SPS0 register**
1	Prescaler output clock CK01 configured by the SPS0 register

Bit 14

CCS01	Channel 1 transfer clock (TCLK) selection
0	**Clock obtained by dividing the operation clock f_{MCK} specified by the CKS01 bit**
1	Clock input from the SCK pin

Bit 8

STS01	Start trigger source selection
0	Only software trigger is valid
1	**Valid edge of the RxD pin (selected during UART reception)**

Bit 6

SIS010	Control of receive data level inversion on channel 1 in UART mode
0	**Falling edge is detected as a start bit**
1	Rising edge is detected as a start bit

Bits 2 and 1

MD012	MD011	Channel 1 operation mode setting
0	0	CSI mode
0	**1**	**UART mode**
1	0	Simplified I^{2}C mode
1	1	Setting prohibited

Bit 0

MD010	Channel 1 interrupt source selection
0	**Transfer end interrupt**
1	Buffer empty interrupt

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

Reception channel communication operation setting

- Serial communication operation setting register 01 (SCR01)
  Data length setting, data transfer order, error interrupt signal mask availability, and operation mode

Symbol: SCR01

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE 01	RXE 01	DAP 01	CKP 01	0	EOC 01	PTC 011	PTC 010	DIR 01	0	SLC 011	SLC 010	0	1	DLS 011	DLS 010
0	**1**	**0**	**0**	0	**1**	**1**	**0**	**1**	0	**0**	**1**	0	1	**1**	**1**

Bits 15 and 14

TXE01	RXE01	Channel 1 operation mode setting
0	0	Communication prohibited
0	**1**	**Reception only**
1	0	Transmission only
1	1	Both transmission and reception

For UART reception, wait for 4 f_{CLK} clock cycles or more before setting SS01 to 1, after setting the RXE01 bit of the SCR01 register to 1.

Bit 10

EOC01	Error interrupt signal (INTSRE1) mask availability selection
0	Error interrupt INTSRE1 is masked
1	**Generation of error interrupt INTSRE1 is enabled**

Bits 9 and 8

PTC011	PTC010	Parity bit setting in UART mode	
		Transmission	Reception
0	0	No parity bit is output	Data is received without parity
0	1	0 parity is output	No parity check is made
1	**0**	Even parity is output	**Check is made for even parity**
1	1	Odd parity is output	Check is made for odd parity

Bit 7

DIR01	Selection of data transfer order in CSI and UART modes
0	Input and output in MSB first
1	**Input and output in LSB first**

Bits 5 and 4

SLC011	SLC010	Stop bit setting in UART mode
0	0	No stop bit
0	**1**	**Stop bit length = 1 bit**
1	0	Stop bit length = 2 bits
1	1	Setting prohibited

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

Symbol: SCR01

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE 01	RXE 01	DAP 01	CKP 01	0	EOC 01	PTC 011	PTC 010	DIR 01	0	SLC 011	SLC 010	0	1	DLS 011	DLS 010
0	**1**	**0**	**0**	0	**1**	**1**	**0**	**1**	0	**0**	**1**	0	1	**1**	**1**

Bits 1 and 0

DLS011	DLS010	Data length setting in CSI mode
0	1	9-bit data length
1	0	7-bit data length
1	**1**	**8-bit data length**
others		Setting prohibited

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

Reception transfer clock setting

- Serial data register 01 (SDR01)
  Transfer clock frequency: fMCK/208 (≈ 9600 Hz)

Symbol: SDR01

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

Bits 15 to 9

SDR01[15:9]							Transfer clock setting by dividing 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
·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·
1	**1**	**0**	**0**	**1**	**1**	**1**	**f_{MCK} /208**
·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·
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.

Initial output level setting

- Serial output register 0 (SO0)
  Initial output: 1

Symbol: SO0

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

Bit 0

SO00	Channel 0 serial data output
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.

Enabling of data output on target channel

- Serial output enable register 0 (SOE0)
  Output enable

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	Channel 0 serial output enable/stop
0	Serial communication output is stopped
1	**Serial communication output is enabled**

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

Port setting

- Port register 5 (P5)
- Port mode register 5 (PM5)
  Port setting for each of transmit data and receive data.

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)**
0	0 is output
1	**1 is output**

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

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

Bit 0

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

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

5.4.7.8 Main Function

Figure 5.13, Figure 5.14 and Figure 5.15 show the flowchart for the main function.

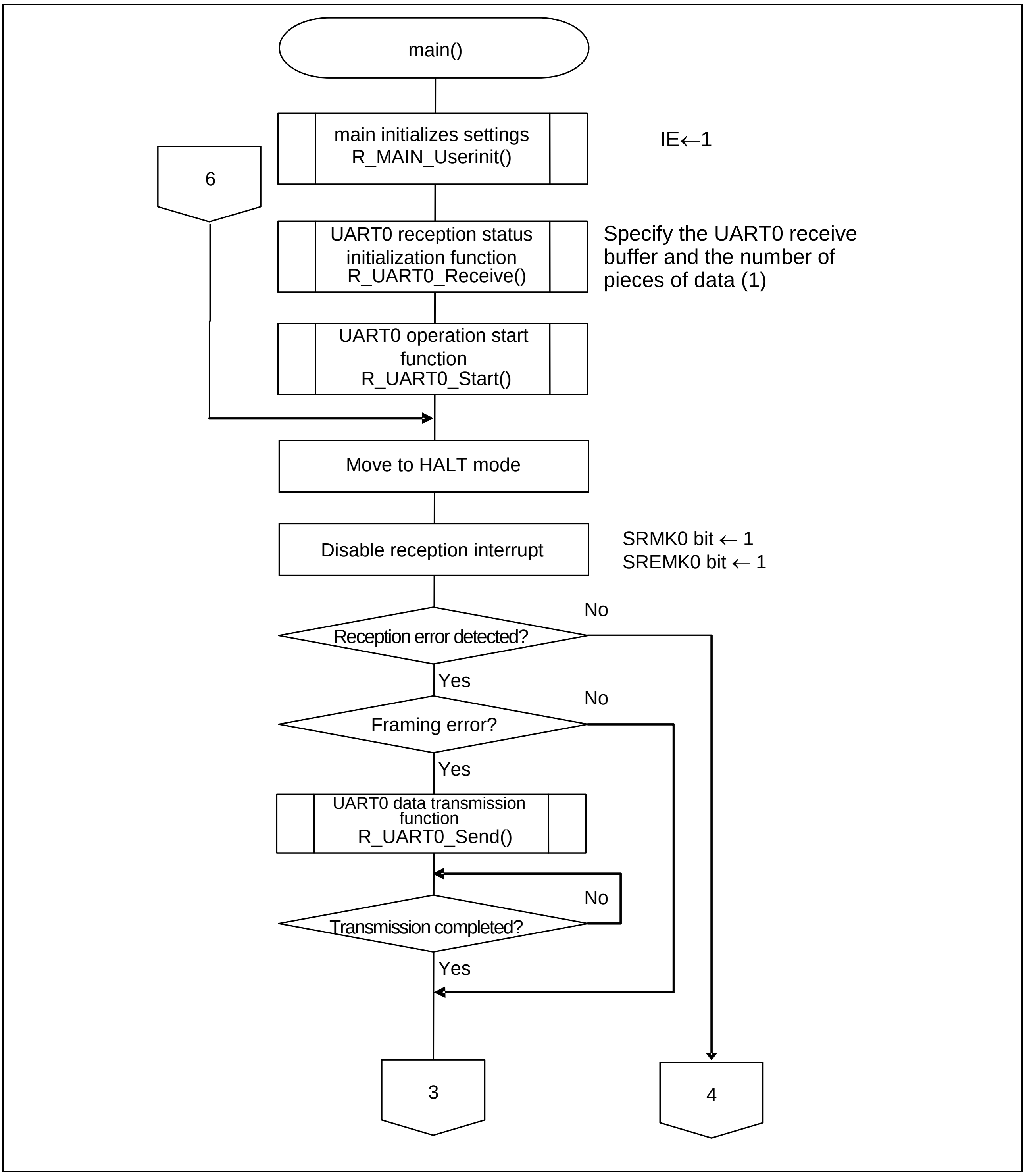


Figure 5.13 Main Function (1/3)

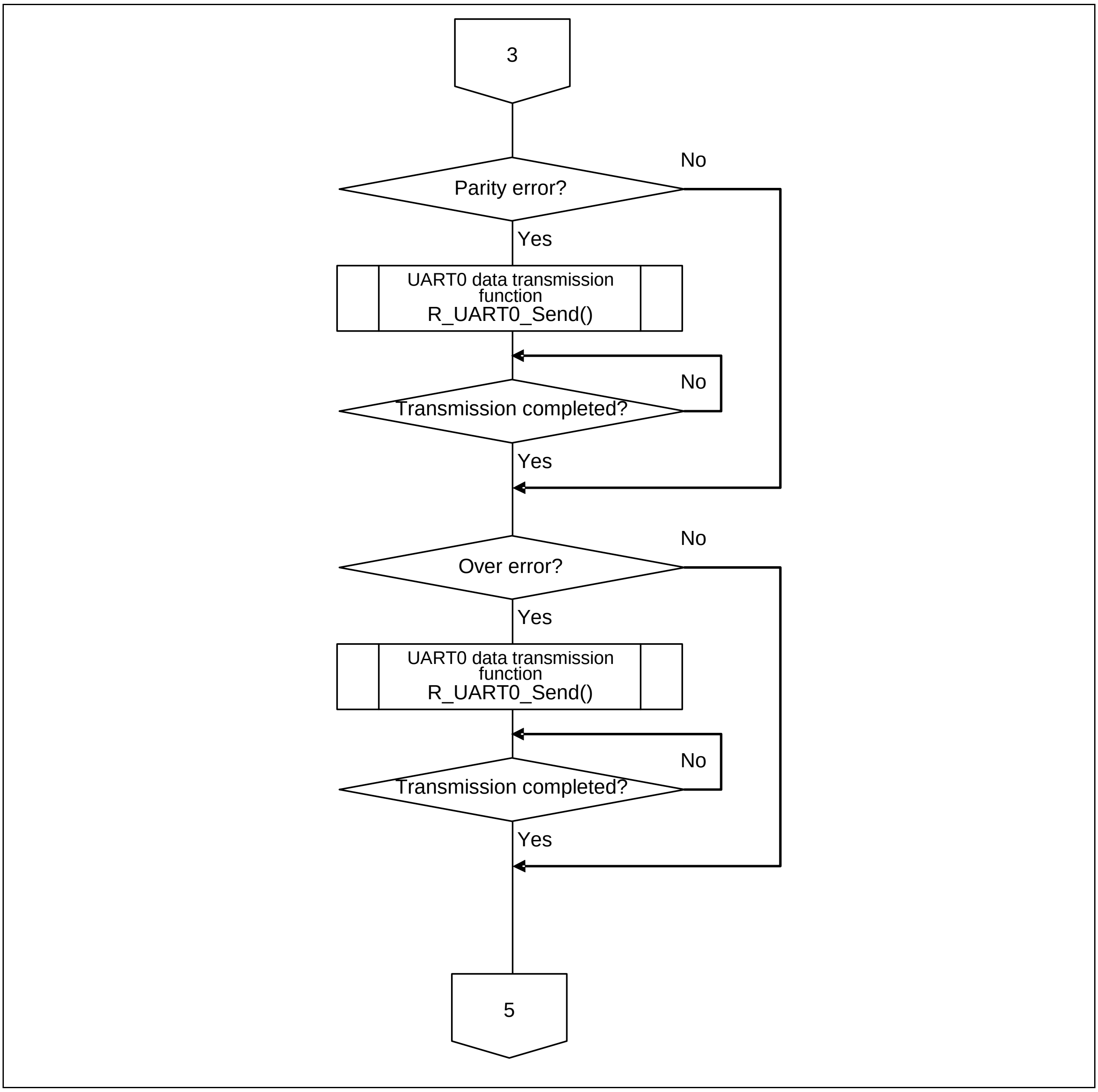


Figure 5.14 Main Function (2/3)

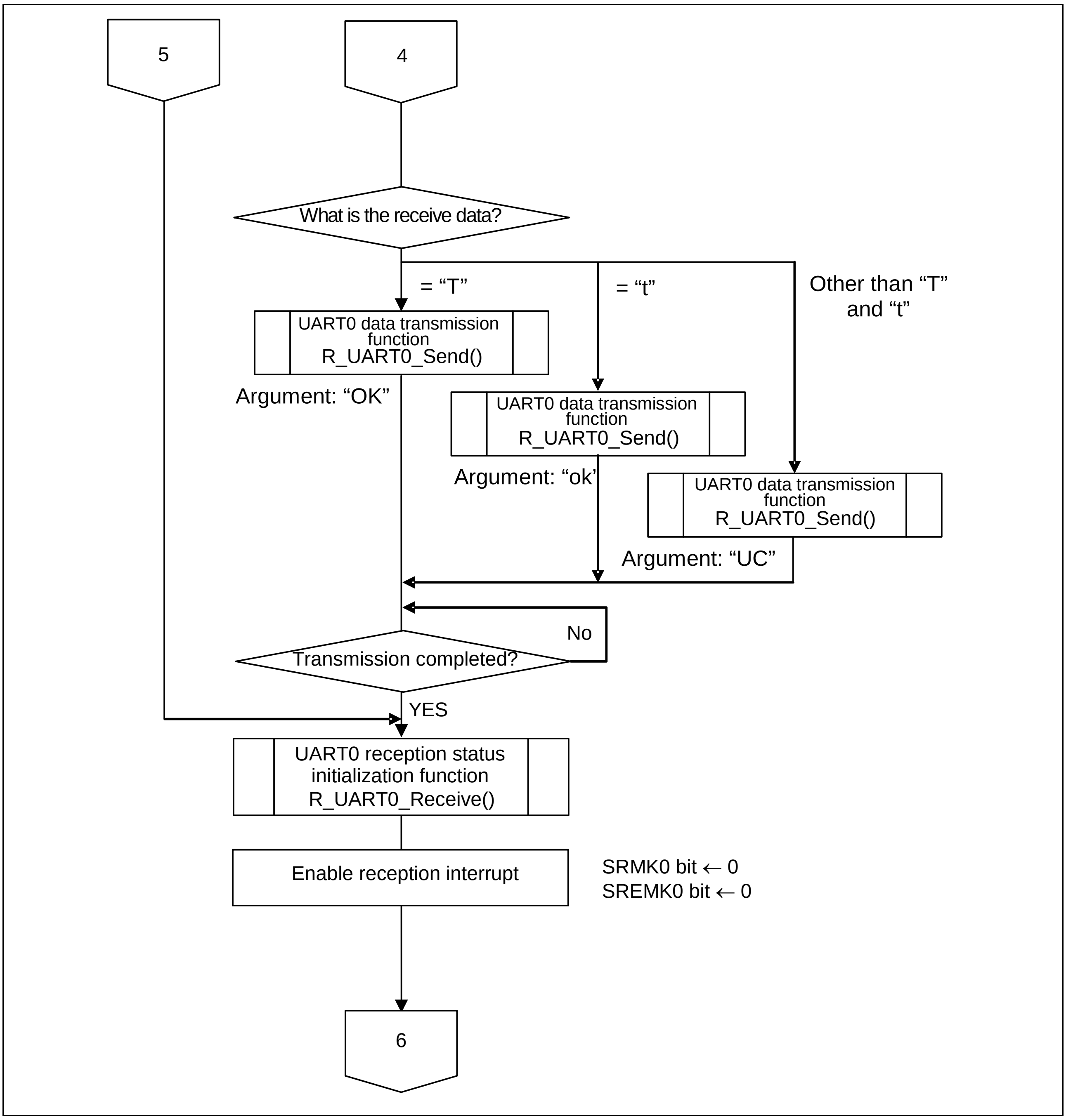

5
4
What is the receive data?
= "T"
= "t"
Other than "T" and "t"
UART0 data transmission function
R_UART0_Send()
Argument: "OK"
UART0 data transmission function
R_UART0_Send()
Argument: "ok"
UART0 data transmission function
R_UART0_Send()
Argument: "UC"
No
Transmission completed?
YES
UART0 reception status initialization function
R_UART0_Receive()
Enable reception interrupt
SRMK0 bit ← 0
SREMK0 bit ← 0
6

Figure 5.15 Main Function (3/3)

5.4.7.9 Main initializes settings

Figure 5.16 shows the flowchart for the main initializes settings.

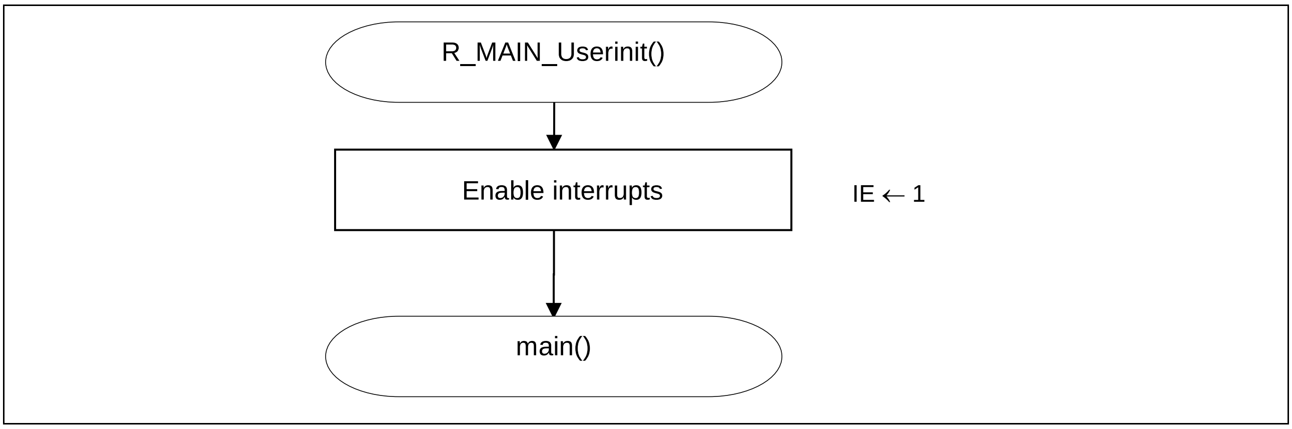


Figure 5.16 Main initializes settings

5.4.7.10 UART0 Reception Status Initialization Function

Figure 5.17 shows the flowchart for the UART0 reception status initialization function.

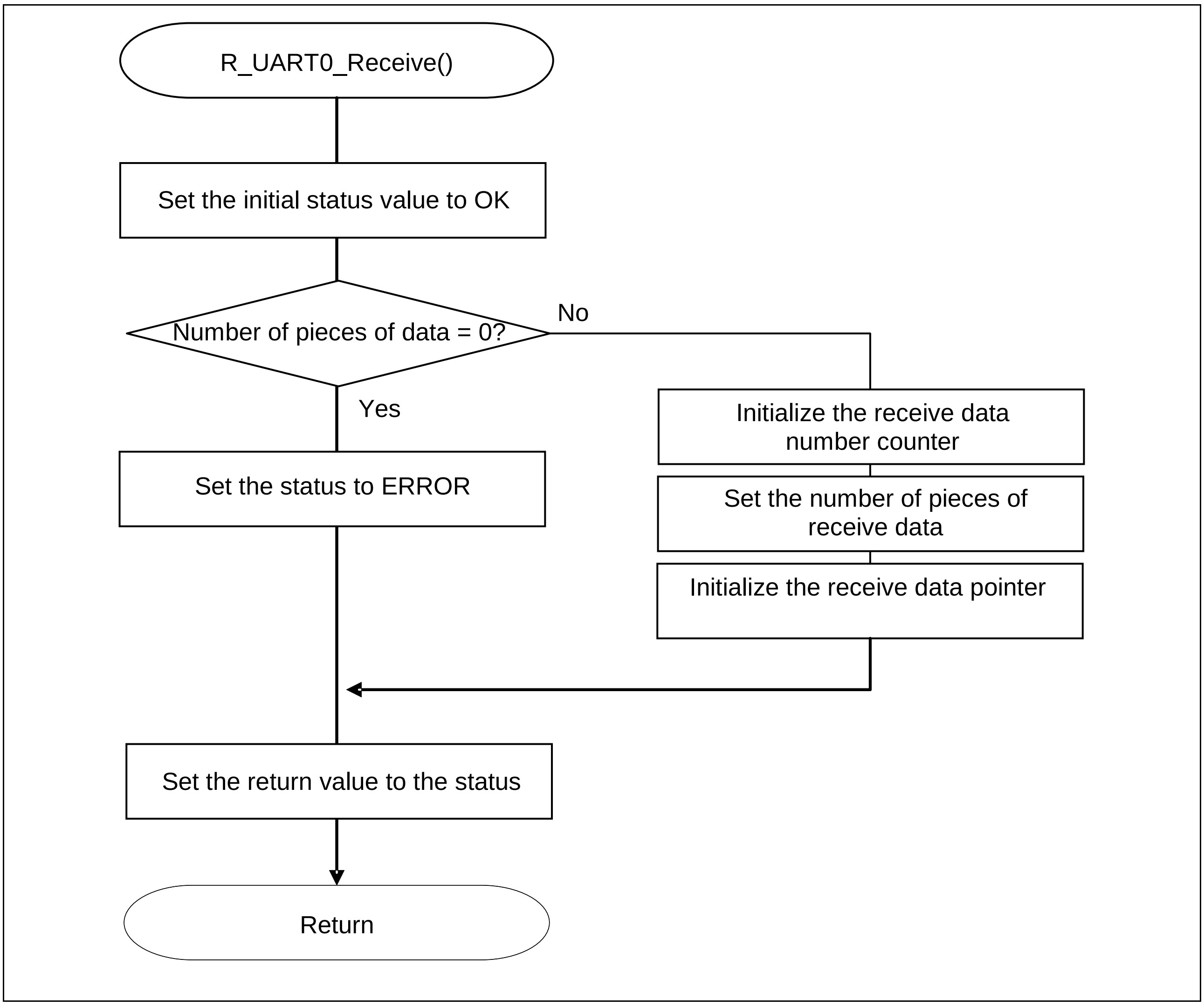

R_UART0_Receive()
Set the initial status value to OK
Number of pieces of data = 0?
No
Yes
Set the status to ERROR
Initialize the receive data number counter
Set the number of pieces of receive data
Initialize the receive data pointer
Set the return value to the status
Return

Figure 5.17 UART0 Reception Status Initialization Function

5.4.7.11 UART0 Operation Start Function

Figure 5.18 shows the flowchart for the UART0 operation start function.

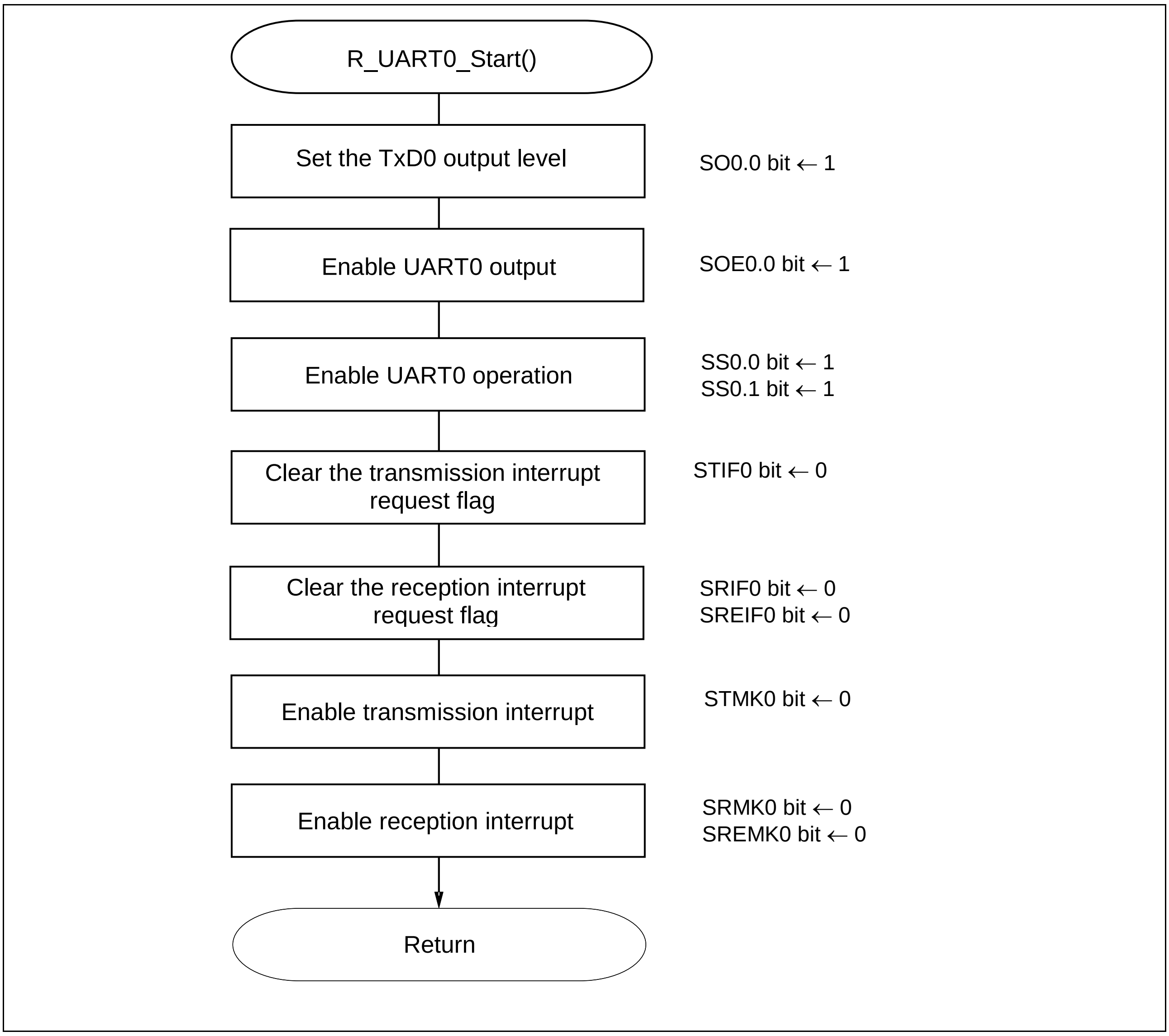


Figure 5.18 UART0 Operation Start Function

Interrupt setting

- Interrupt request flag register (IF0H)
  Clear the interrupt request flag
- Interrupt mask flag register (MK0H)
  Cancel interrupt mask

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
0	**0**	**0**	0	0	x	x	x

Bit 7

SREIF0	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

Bit 6

SRIF0	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

Bit 5

STIF0	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

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

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
0	**0**	**0**	1	1	x	x	x

Bit 7

SREMK0	Interrupt processing control
0	**Enables interrupt processing.**
1	Disables interrupt processing.

Bit 6

SRMK0	Interrupt processing control
0	**Enables interrupt processing.**
1	Disables interrupt processing.

Bit 5

STMK0	Interrupt processing control
0	**Enables interrupt processing.**
1	Disables interrupt processing.

Transition to communication wait state

- Serial channel start register 0 (SS0)
  Operation start

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[Note]	x	**1**[Note]	**1**

Bits 3 to 0

SS0n	Channel n operation start trigger
0	Trigger operation is not performed
1	**SE0n is set to 1, and a communication wait state is entered.**

Note For UART reception, wait for 4 f_{CLK} clock cycles or more before setting SS0n to 1, after setting the RXE0n bit of the SCR0n register to 1.

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

5.4.7.12 INTSR0 Interrupt Service Routine

Figure 5.19 shows the flowchart for the INTSR0 interrupt service routine.

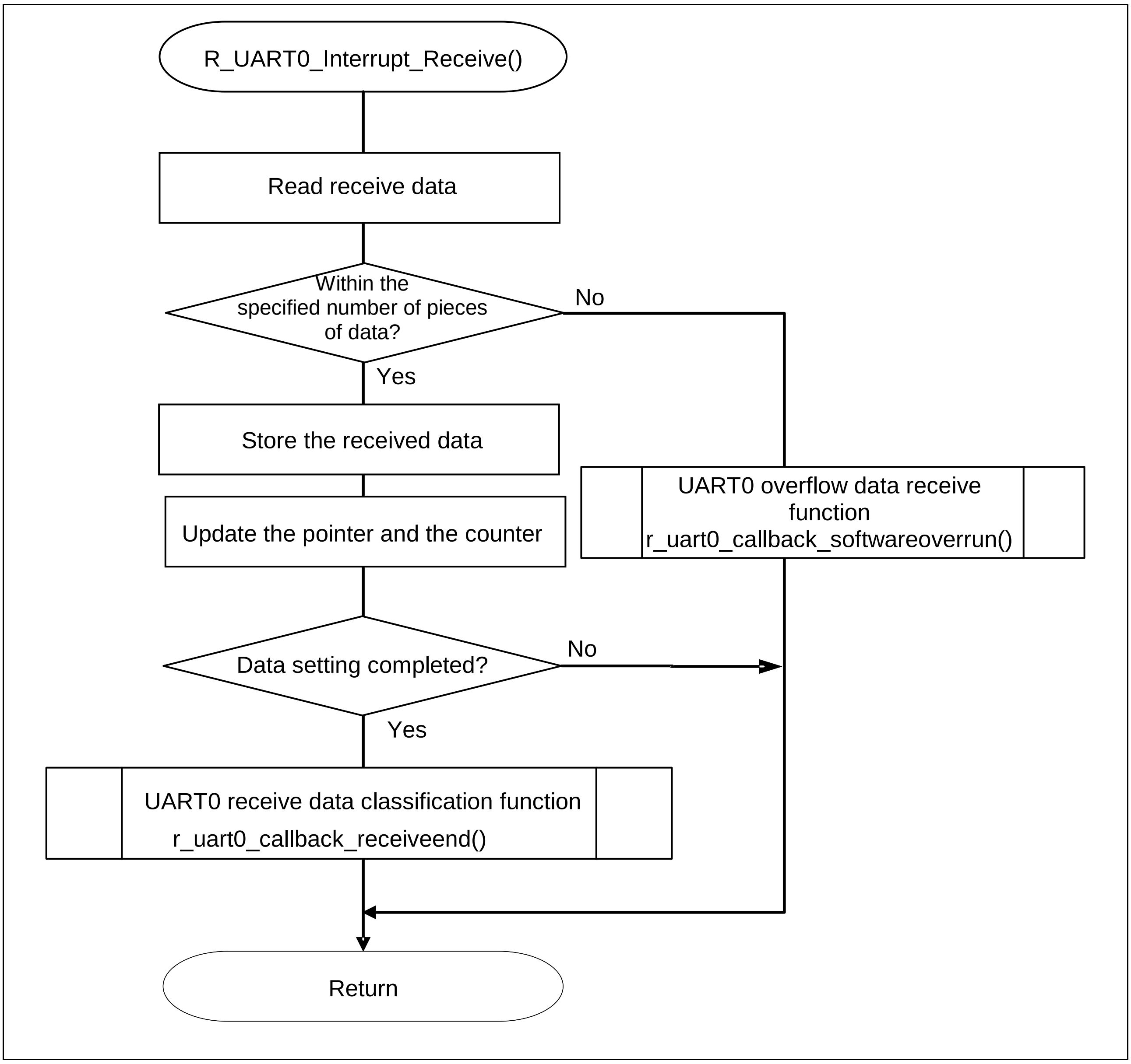

R_UART0_Interrupt_Receive()
Read receive data
Within the specified number of pieces of data?
No
Yes
Store the received data
UART0 overflow data receive function r_uart0_callback_softwareoverrun()
Update the pointer and the counter
Data setting completed?
No
Yes
UART0 receive data classification function r_uart0_callback_receiveend()
Return

Figure 5.19 INTSR0 Interrupt Service Routine

5.4.7.13 UART0 Receive Data Classification Function

Figure 5.20 shows the flowchart for the UART0 receive data classification function.

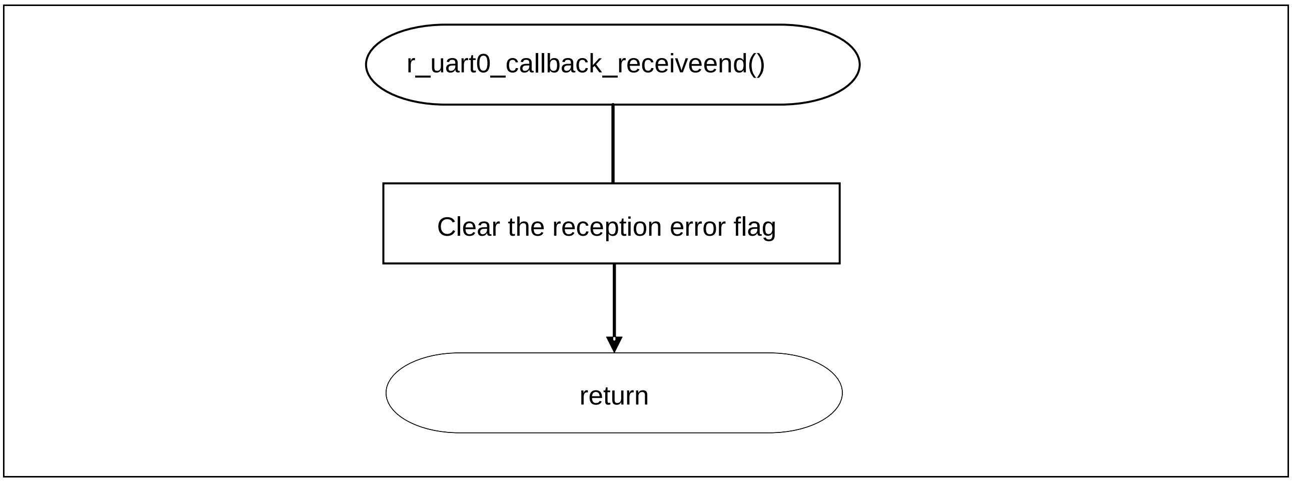


Figure 5.20 UART0 Receive Data Classification Function

5.4.7.14 UART0 Data Transmission Function

Figure 5.21 shows the flowchart for the UART0 data transmission function.

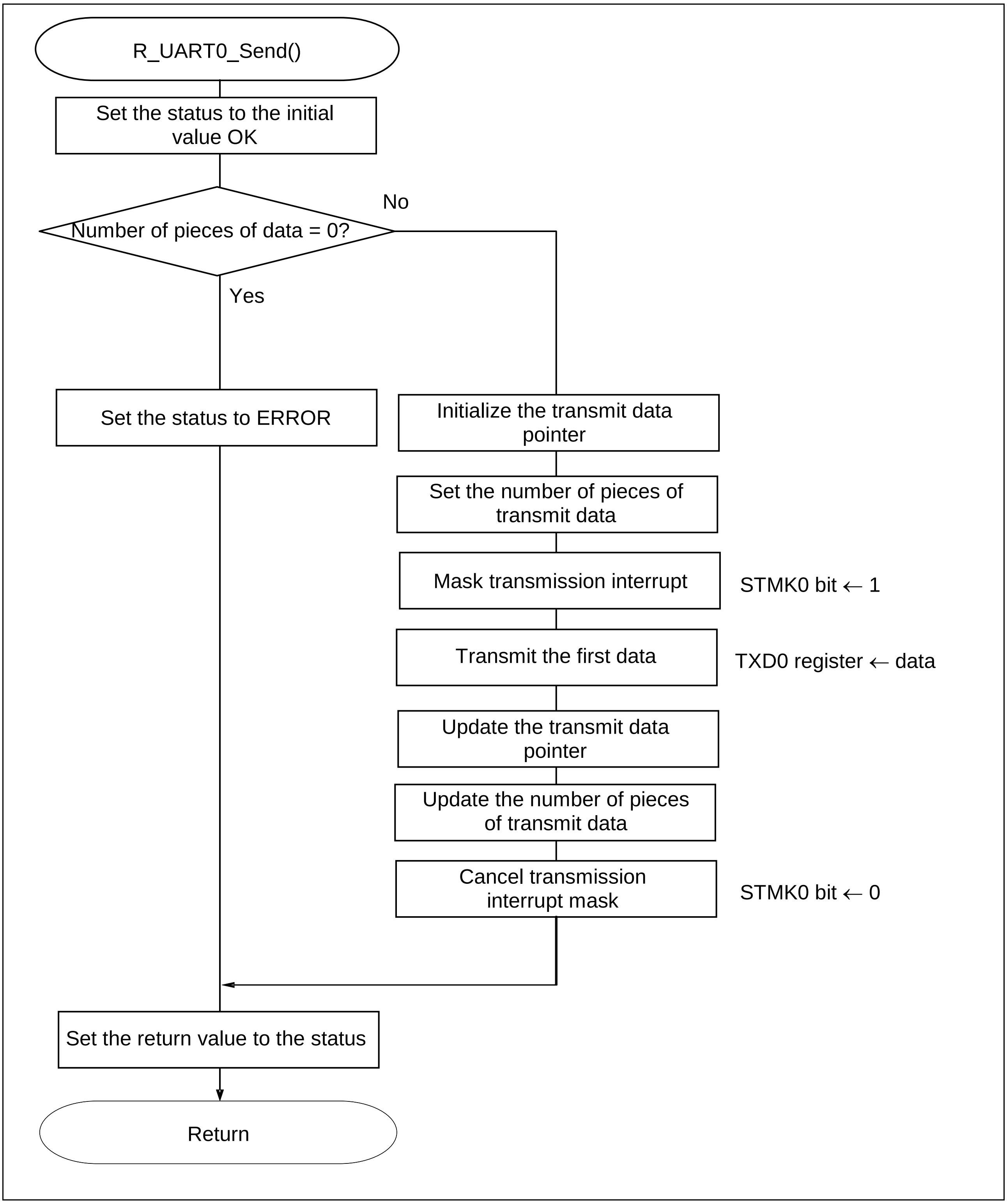

R_UART0_Send()
Set the status to the initial value OK
Number of pieces of data = 0?
No
Yes
Set the status to ERROR
Initialize the transmit data pointer
Set the number of pieces of transmit data
Mask transmission interrupt
STMK0 bit ← 1
Transmit the first data
TXD0 register ← data
Update the transmit data pointer
Update the number of pieces of transmit data
Cancel transmission interrupt mask
STMK0 bit ← 0
Set the return value to the status
Return

Figure 5.21 UART0 Data Transmission Function

5.4.7.15 UART0 Reception Error Interrupt Function

Figure 5.22 shows the flowchart for the UART0 reception error interrupt function.

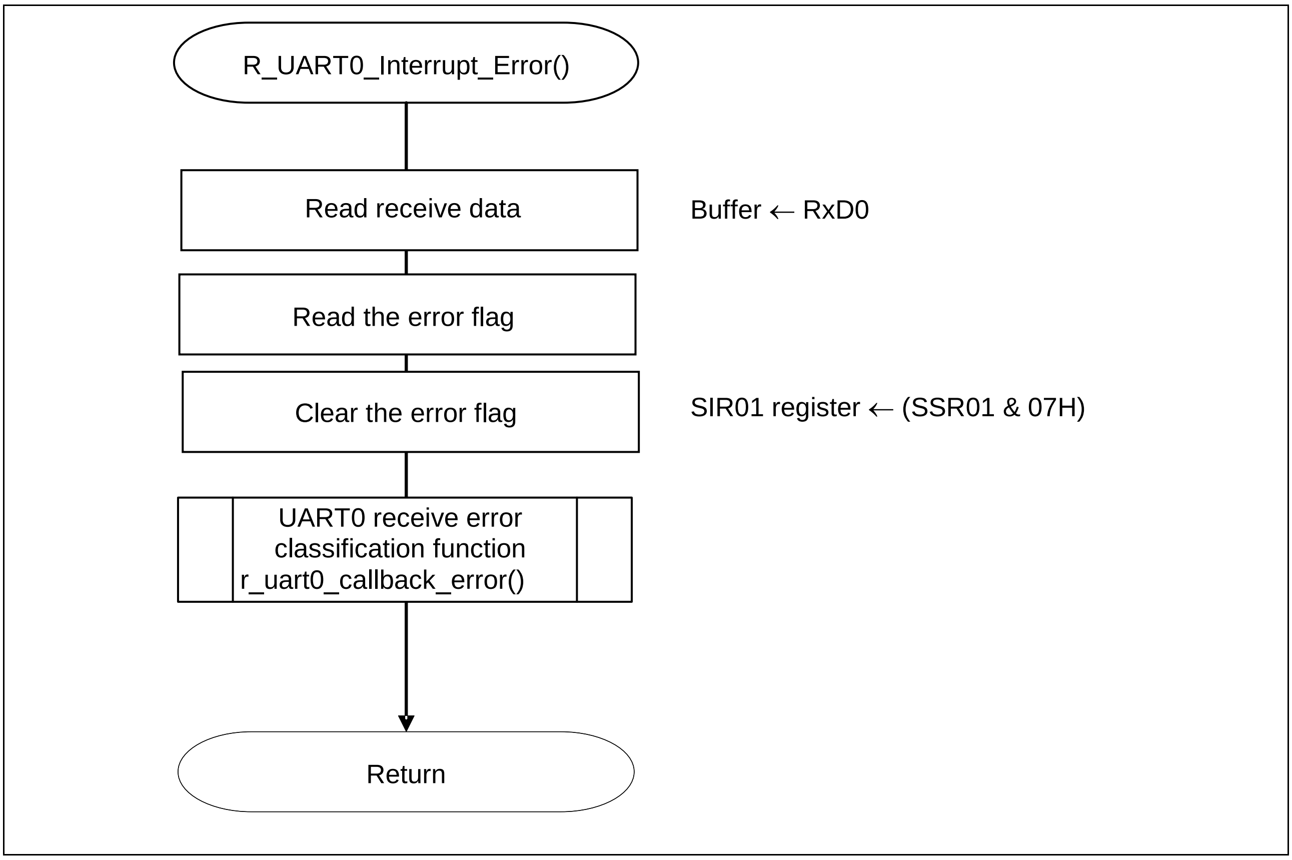

R_UART0_Interrupt_Error()
Read receive data
Buffer ← RxD0
Read the error flag
Clear the error flag
SIR01 register ← (SSR01 & 07H)
UART0 receive error classification function r_uart0_callback_error()
Return

Figure 5.22 UART0 Reception Error Interrupt Function

5.4.7.16 UART0 Reception Error Classification Function

Figure 5.23 shows the flowchart for the UART0 reception error classification function.

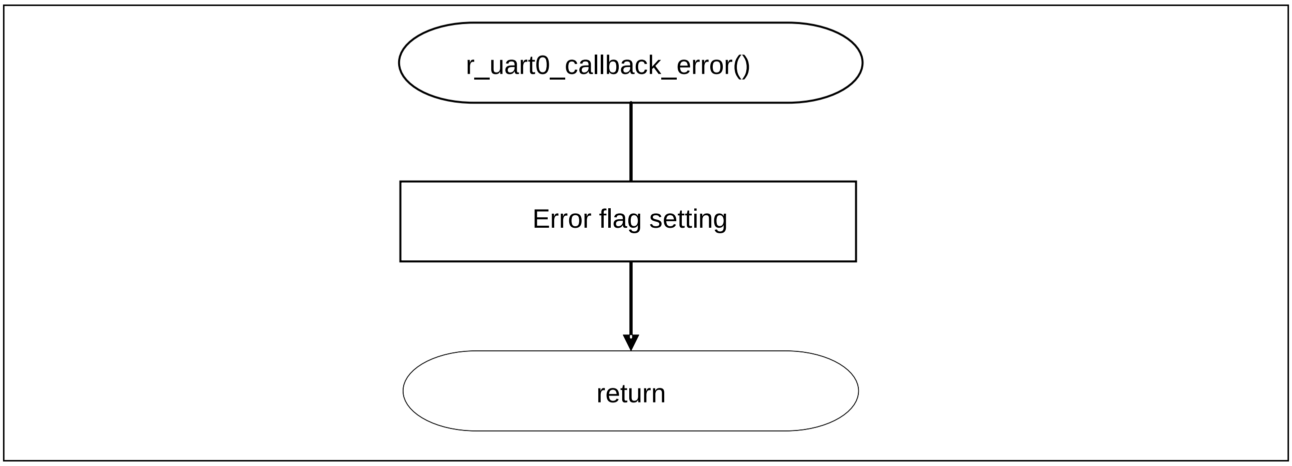


Figure 5.23 UART0 Reception Error Classification Function

5.4.7.17 INTST0 Interrupt Service Routine

Figure 5.24 shows the flowchart for the INTST0 interrupt service routine.

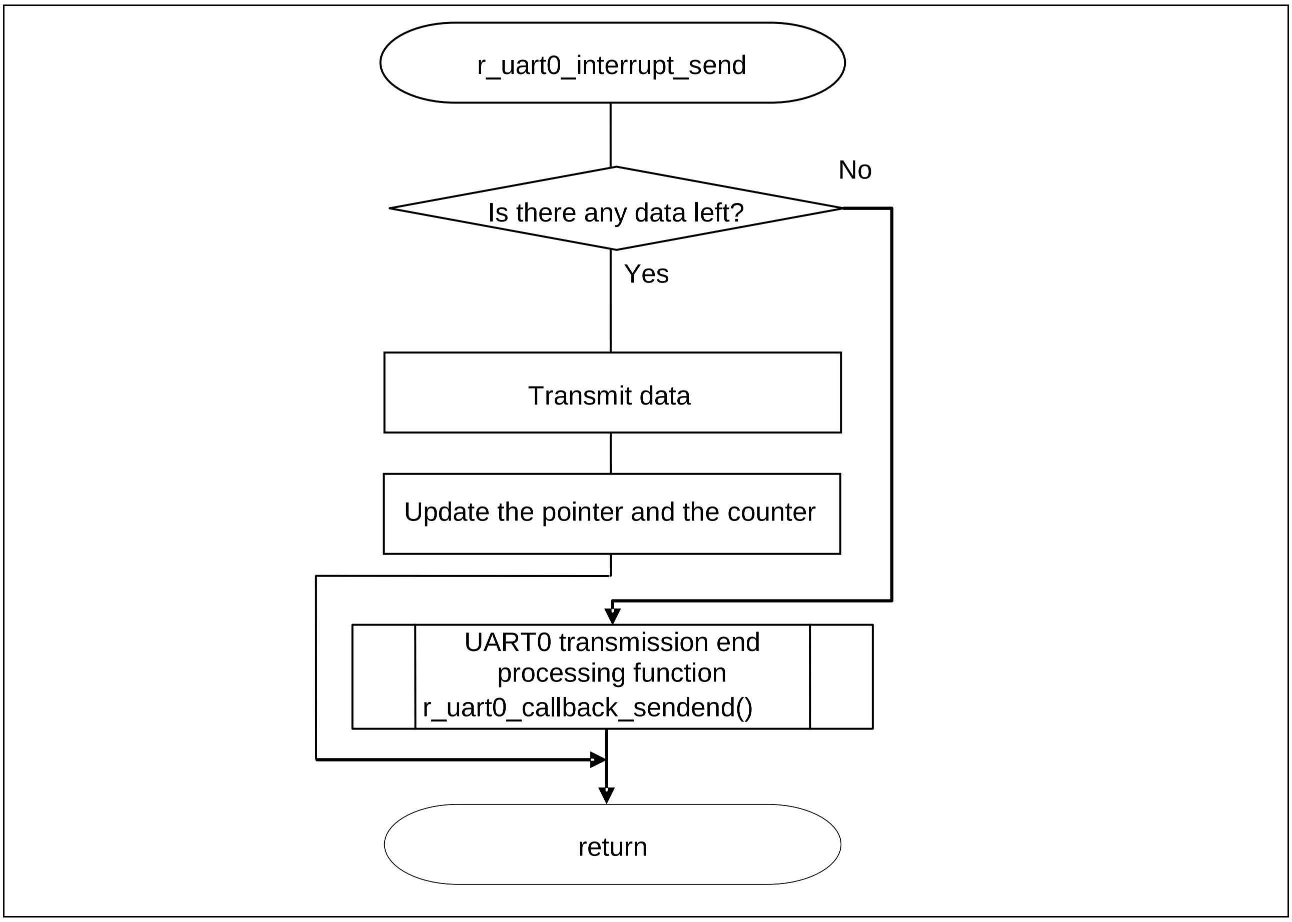

r_uart0_interrupt_send
No
Is there any data left?
Yes
Transmit data
Update the pointer and the counter
UART0 transmission end processing function r_uart0_callback_sendend()
return

Figure 5.24 INTST0 Interrupt Service Routine

5.4.7.18 UART0 Transmission End Processing Function

Figure 5.25 shows the flowchart for the UART0 transmission end processing function.

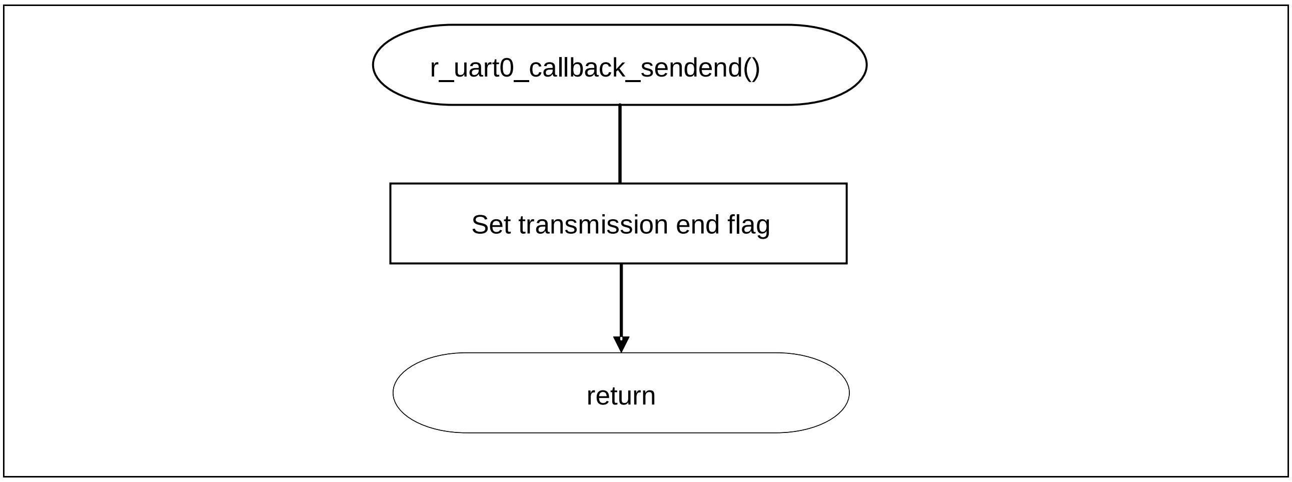


Figure 5.25 UART0 Transmission End Processing Function

5.4.8 Sample Code

The sample code is available on the Renesas Electronics Website.

5.4.9 Related Application Note

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

- RL78/G13 Initialization (R01AN2575E) Application Note

5.4.10 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.

Website and Support

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

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec. 22, 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

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

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

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

Renesas Electronics Canada Limited
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