RL78/I1D

R01AN2584EJ0100
Rev.1.00
July 1. 2015

UART communication with middle speed on chip oscillator (Baud rate correction with 32.768kHz sub-clock)

Introduction

This application note explains how to execute UART transmission/reception using the RL78/I1D middle-speed on-chip oscillator. A 32.768kHz sub-clock with frequency accuracy sufficient for a real-time clock is used to periodically measure the oscillation period of the middle-speed on-chip oscillator. UART communication using the middle-speed on-chip oscillator is enabled through corrections to the baud rate based on measurement results.

Target Device

RL78/I1D

When using this application note for other microcomputers, please change it according to the corresponding specification and evaluate thoroughly before use.

Contents

1. Specification

The RL78/I1D middle-speed on-chip oscillator can be used for UART transmission/reception by using the 32.768kHz sub-clock to correct the baud rate. However, the RL78/I1D middle-speed on-chip oscillator does not provide frequency accurate enough for UART communications. A 32.768kHz sub-clock with frequency accuracy sufficient for a real-time clock is used to periodically measure the oscillation period of the middle-speed on-chip oscillator. The results are used to detect the oscillation period difference and correct the baud rate accordingly.

The serial array unit (SAU) is used to carry out UART communication (both transmission and reception). In UART communication, ACII characters sent by the targeted communication device are decoded and the corresponding process executed.

The 8-bit interval timer generates an interrupt every 1000ms, creating the frequency measurement period. The frequency measurement circuit generates the middle-speed on-chip oscillator frequency.

Table 1.1 Peripheral functions and usages

Peripheral functions	Usage
Serial Array Unit (SAU)	UART transmission/reception (UART0)
8-bit interval timer	Generates correction period for middle-speed on-chip oscillator
Frequency measurement circuit	Measures oscillation accuracy of middle-speed on-chip oscillator

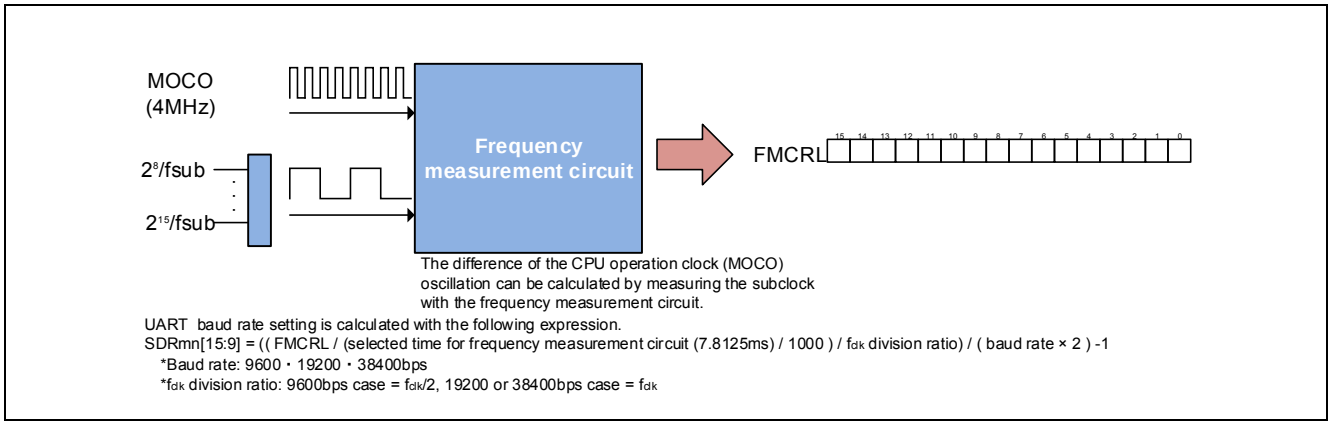

Figure 1.1 Baud Rate Correction

2. Conditions for Confirming Operations

Operations for the sample code discussed in this application note were confirmed under the following conditions.

Table 2.1 Conditions for Confirming Operations

Item	Description
MCU	RL78/I1D (R5F117GCG)
Operating frequency	Middle-speed on-chip oscillator clock (f_{IM}): 4MHz (UART communication)
Operating voltage	3.3V (operating range1.6V to 3.6V) LVD operations (V_{LVI}): LVD off
Integrated development environment (CS+)	Made by Renesas Electronics Corp. CS+ V3.00.00
C compiler (CS+)	Made by Renesas Electronics Corp. CA78K0R V1.71
Integrated development environment (e2studio)	Made by Renesas Electronics Corp. e2studio V3.1.2.10
C compiler (e2studio)	Made by Renesas Electronics Corp. KPIT GNURL78-ELF Toolchain V14.0.3
Integrated development environment (IAR)	Made by IAR Systems Corporation IAR Embedded Workbench for Renesas RL78 V1.40.5
C compiler (IAR)	Made by IAR Systems Corporation IAR C/C++ Compiler for Renesas RL78 V1.40.5
Board	RL78/I1D CPU board (RTE5117GC0TGB00000R)

Notes: The RL78/I1D CPU board (RTE5117GC0TGB00000R) LEDs are connected to N-ch open drain ports (P60, P61). Setting P60 and P61 to Hi-Z to turn off the LEDs may cause through current flow. Operations for this application note were evaluated assuming no through current. Therefore, actual measured supply current during HALT mode is 73.5[uA].

3. Related Application Notes

Application notes related to this document are shown below. Please refer to these as needed.

- RL78/G13 Initialization [for CubeSuite+, IAR, and e2 studio] (R01AN0451EJ)
- RL78/G13 Serial Array Unit (UART Communication) (R01AN0459EJ)
- RL78/I1D UART communication with middle speed on chip oscillator (R01AN2326EJ)

4. Hardware Explanation

4.1 Hardware Structure Example

Figure 4.1 shows the hardware used in this application note.

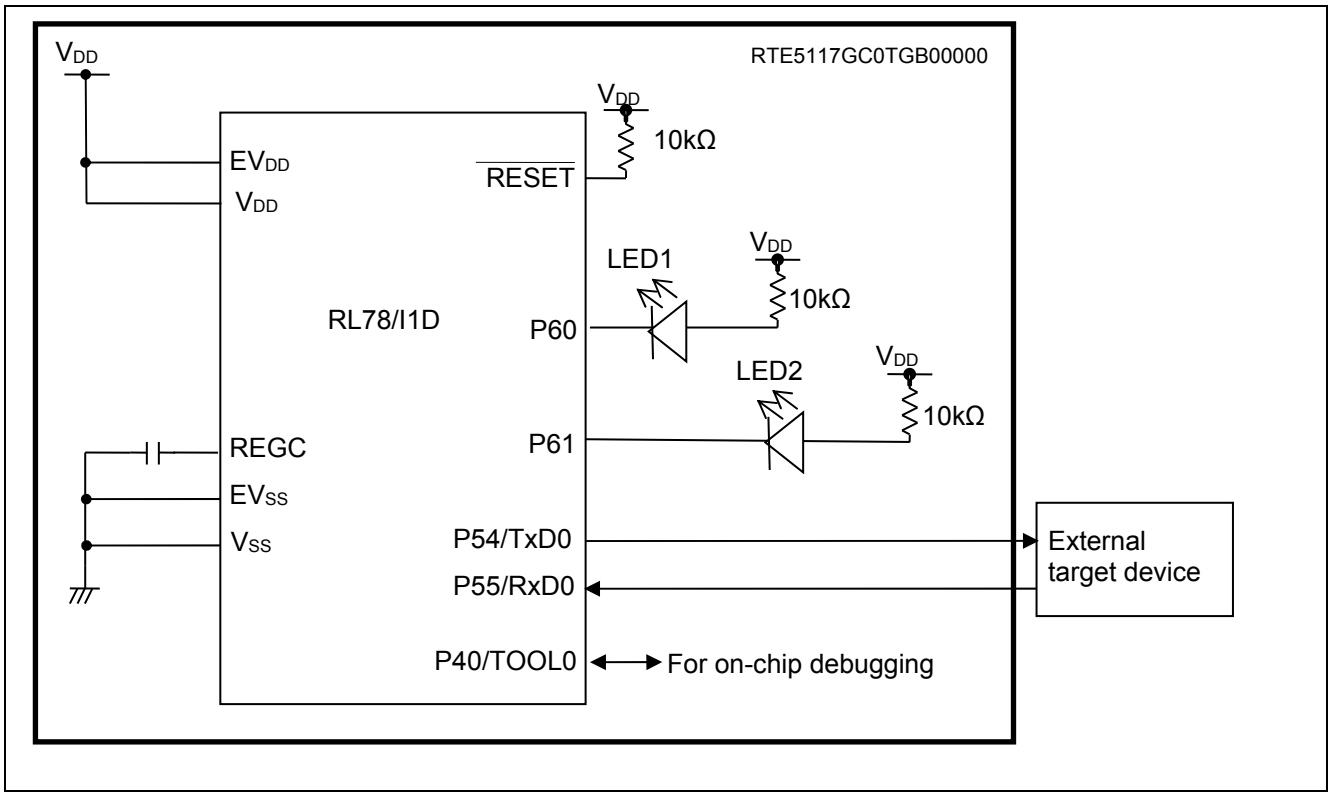

Figure 4.1 Hardware Configuration

Note: 1 This simplified circuit diagram was created to show an overview of connections only.
When actually designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.
(Connect each input-only port to V_{DD} or V_{SS} through a resistor.)

2 If a pin name starts with EV_{SS}, connect the pin to V_{SS}, if it starts with EV_{DD}, connect it to V_{DD}.

3 Make V_{DD} higher than the RESET release voltage (V_{LVI}) set in LVD.

4 P60 and P61 are N-ch open drain output ports and need to be set to Hi-Z to turn off the LEDs. This may cause through current to flow through the two ports.

4.2 Pin List

Table 4.1 provides a list of the pins used in this application note and their functions.

Table 4.1 List of Pins and Functions

Pin Name	Input/Output	Function
P60	Output	LED1 control
P61	Output	LED2 control
P54/TxD0	Output	TxD0: UART0 transmission output pin
P55/RxD0	Input	RxD0: UART0 receive input pin

5. Software Explanation

This RL78/1D sample code uses the compiler code generation function. When a generated function needs to be edited, CS+ or e2studio versions can be used to change the code generation properties. By setting the code generation mode to "Do nothing if file exists", an existing file in the project will not be updated even if code generation is executed. Note that if code generation is executed after setting the mode to "Merge file" or "Overwrite file", the file existing in the project will be updated, but the RL78/1D sample code will not operate properly.

Note that the IAR Embedded Workbench does not include the code generation plugin. The user needs to generate the code with the Applilet3 code generation tool (available at the following URL) and embed the code in the IAR project. (http://am.renesas.com/products/tools/coding_tools/coding_assistance/applilet/downloads.jsp#)

Note that the RL78/1D sample code will not operate properly when the source file is rewritten with a downloaded IAR sample code that has been updated with Applilet3.

Figure 5.1 and Figure 5.2 show code generation property setting screens.

- CS+

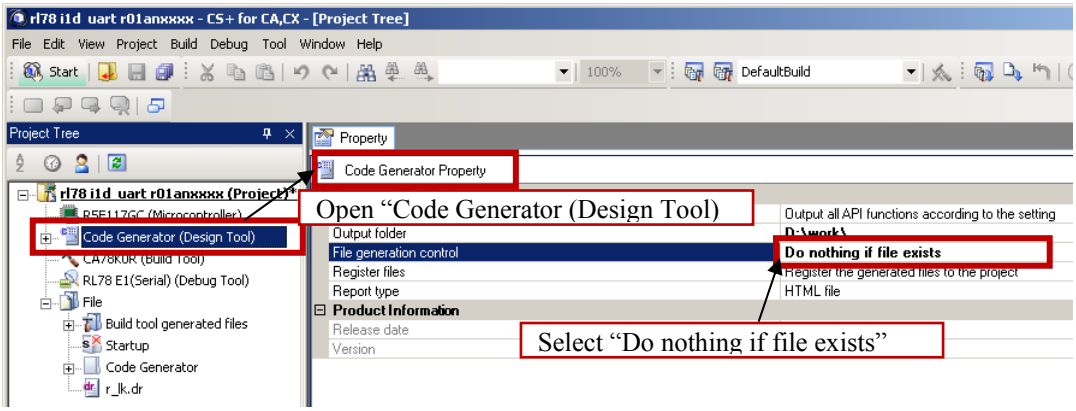

Figure 5.1 Code Generation Property Setting Screen (CS+)

- e2studio

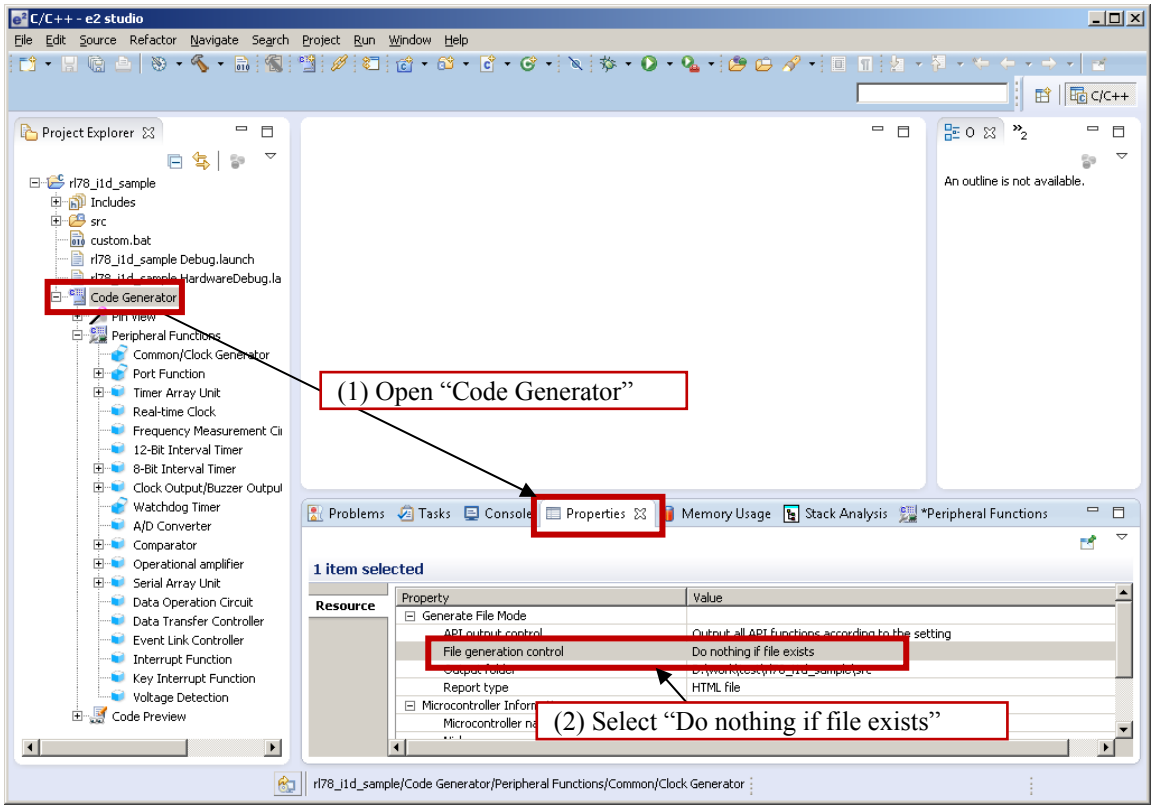

Figure 5.2 Code Generation Property Setting Screen (e2studio)

5.1 Operation Outline

This sample code transmits corresponding data to the target device in response to received data from the device. When an error occurs, the sample code transmits data to the device based on the type of error. Table 5.1 and Table 5.2 show the data corresponding to reception data and errors, respectively.

Table 5.1 Receive Data and Corresponding Transmission Data

Receive Data	Response (transmission) 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 5.2 Error type and Corresponding Transmission Data

Occurred error	Response (transmission) 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) **Port initialization**

- Set P60 and P61 to Hi-Z output to turn off LED1 and LED2

(2) **SAU initialization**

<Setting conditions>

- SAU0 channel 0 and 1: use as UART
- Data output pin: P54/TxD0
- Data input pin: P55/RxD0
- Data length: 8-bit
- Data transfer direction setting: LSB first
- Parity setting: even
- Receive data level setting: standard
- Baud rate: 9600bps
- Reception complete interrupt (INSTR0), error interrupt (INTSRE0): enabled
- Interrupt priority level of INTSR0, INTST0, INTSRE0: low (level 3)

(3) **8-bit interval timer initialization**

- Use auto interval timer operation
- Interval time setting: 1000ms
- Use 8-bit interval timer compare match interval (INTIT00)
- Interrupt priority level: low (level 3)

(4) **Frequency measurement circuit initialization**

- Frequency count clock: f_{lm}
- Frequency measurement time: 2^8/f_{SXR} (7.813ms)
- Use frequency measurement complete interrupt (INTFM)
- Interrupt priority level: low (level 3)

(5) **main process initialization**

<Setting conditions>

- UART0 receive status initialization
  - Set variable md_status to “0” (OK).
  - Set variable g_uart0_rx_count to “0” (receive count value = 0).
  - Set variable g_uart0_rx_length to “1” (receive data number = 1).
  - Set variable gp_uart0_rx_address to receive data pointer.

- UART0 operation start
  - Set SO0 register SO00 bit to “1” (set serial data output value to 1).
  - Set SOE0 register SOE00 bit to “1” (enable output in serial communication operation).
  - Set SS0 register SS01 bit and SS00 bit to “11B” (sets SEmn bit to 1, shifts to communication wait state).
  - Set IF0H register STIF0 bit to “0” (interrupt request signal not generated).
  - Set IF0H register SRIF0 bit to “0” (interrupt request signal not generated).
  - Set IF0H register SREIF0 bit to “0” (interrupt request signal not generated).
  - Set MK0H register STMK0 bit to “0” (interrupt processing enabled).
  - Set MK0H register SRMK0 bit to “0” (interrupt processing enabled).
  - Set MK0H register SREMK0 bit to “0” (interrupt processing enabled).
- 8-bit interval timer operation start
  - Set IF2L register ITIF00 bit to “0” (interrupt request signal not generated).
  - Set MK2L register ITMK00 bit to “0” (interrupt processing enabled).
  - Set TRTCR0 register TSTART00 bit to “1” (start count).

(6) **Shift to HALT mode**

(7) **Shift from HALT to normal mode with the 8-bit interval timer compare match interrupt (INTIT00)**

- Set P60 to Low output, turn on LED1.
- Set variable g_uart0afteradjustment to “1” (after baud rate correction).
- Set PER2 register FMCEN bit to “1” (frequency measurement circuit input clock supply).
- Start frequency measurement circuit
  - Set IF1L register FMIF bit to “0” (interrupt request signal not generated).
  - Set MK1L register FMMK bit to “0” (interrupt processing enabled).
  - Set FMCTL register FMS bit to “1” (frequency measurement circuit operation)
- Wait until frequency measurement complete interrupt (INTFM)
- After the frequency is measured, read frequency measurement count register (FMCRL) and set value to variable g_fmcrl.
- Stop frequency measurement circuit
  - Set MK1L register FMMK bit to “1” (interrupt processing disabled).
  - Set IF1L register FMIF IF2 bit to “0” (interrupt request signal not generated).
  - Set FMCTL register FMS bit to “0” (frequency measurement circuit stopped).
- Set PER2 register FMCEN bit to “0” (frequency measurement circuit input clock supply stopped).

- UART0 stop
  - Set MK0H register STMK0 bit to “1” (interrupt processing disabled).
  - Set MK0H register SRMK0 bit to “1” (interrupt processing disabled).
  - Set MK0H register SREMK0 bit to “1” (interrupt processing disabled).
  - Set ST0 register ST01 and ST00 bits to “11B” (clear SEmn bit to 0, stop communication operation).
  - Set SOE0 register SOE00 bit to “1” (enable output in serial communication operations).
  - Set IF0H register STIF0 bit to “0” (interrupt request signal not generated).
  - Set IF0H register SRIF0 bit to “0” (interrupt request signal not generated).
  - Set IF0H register SREIF0 bit to “0” (interrupt request signal not generated).
- Set baud rates to SDR01/SDR00 registers.
  - g_sdr = (uint16_t)((g_fmcrl / (SUB_CLK_ FSXR)) / 1 / (g_baudrate * 2) -1) << 8
  - SDR01 = g_sdr
  - SDR00 = g_sdr
- UART0 operation start
  - Set SO0 register SO00 bit to “1” (sets serial data output value to “1”).
  - Set SOE0 register SOE00 bit to “1” (enable output in serial communication operation).
  - Set SS0 register SS01 and SS00 bits to “11B” (sets SEmn to 1 and shifts to communication wait state).
  - Set IF0H register STIF0 bit to “0” (interrupt request signal not generated).
  - Set IF0H register RIF0 bit to “0” (interrupt request signal not generated).
  - Set IF0H register SREIF0 bit to “0” (interrupt request signal not generated).
  - Set MK0H register STMK0 bit to “0” (interrupt processing disabled).
  - Set MK0H register SRMK0 bit to “0” (interrupt processing disabled).
  - Set MK0H register SREMK0 bit to “0” (interrupt processing disabled).
- Set P60 to Hi-Z output, turn off LED1.

(8) **Return from HALT mode with UART reception transmission complete interrupt (INTSR0)**

- Stop 8-bit interval timer.
  - Set MK2L register ITMK00 bit to “1” (interrupt processing disabled).
  - Set IF2L register ITIF00 bit to “0” (interrupt request signal not generated).
  - Set TRTCR0 register TSTART00 bit to “0” (stop count).
- Disable UART reception interrupt.
  - Set MK0H register SRMK0 bit to “1” (interrupt processing disabled).
  - Set MK0H register SREMK0 bit to “1” (interrupt processing disabled).

- UART reception errors
  - Framing error
    - Send “FE¥r¥n” message.
    - Wait until message transfer is complete (set g_uart0txend to “1”).
  - Parity error
    - Send “PE¥r¥n” message.
    - Wait until message transfer is complete (set g_uart0txend to “1”).
  - Overrun error
    - Send “OE¥r¥n” message.
    - Waits until message transfer is complete (set g_uart0txend to “1”).
- Normal UART reception
  - When received data is “T,” send “OK¥r¥n” message.
  - When received data is “t,” send “ok¥r¥n” message.
  - In all other cases, send “UC¥r¥n” message.
  - In all above cases, after message is sent, wait until message transfer complete (set g_uart0txend to “1”).
- UART0 reception status initialization
  - Set variable md_status to “0” (OK).
  - Set variable g_uart0_rx_count to “0” (receive count value = 0).
  - Set variable g_uart0_rx_length to “1” (receive data number = 1).
  - Set variable gp_uart0_rx_address to receive data pointer.
- Enable UART reception interrupt
  - Set MK0H register SRMK0 bit to “0” (interrupt processing enabled).
  - Set MK0H register SREMK0 bit to “0” (interrupt processing enabled).
- 8-bit interval timer start
  - Set IF2L register ITIF00 bit to “0” (interrupt request signal not generated).
  - Set MK2L register ITMK00 bit to “0” (interrupt processing enabled).
  - Set TRTCR0 register TSTART00 bit to “1” (start count).

(9) **Repeat step 7 or 8.**

Figure 5.3 shows the Timing of Return from HALT Mode (8-bit interval timer compare match interrupt) and Figure 5.4 shows the Timing of Return from HALT Mode (UART reception transfer complete).

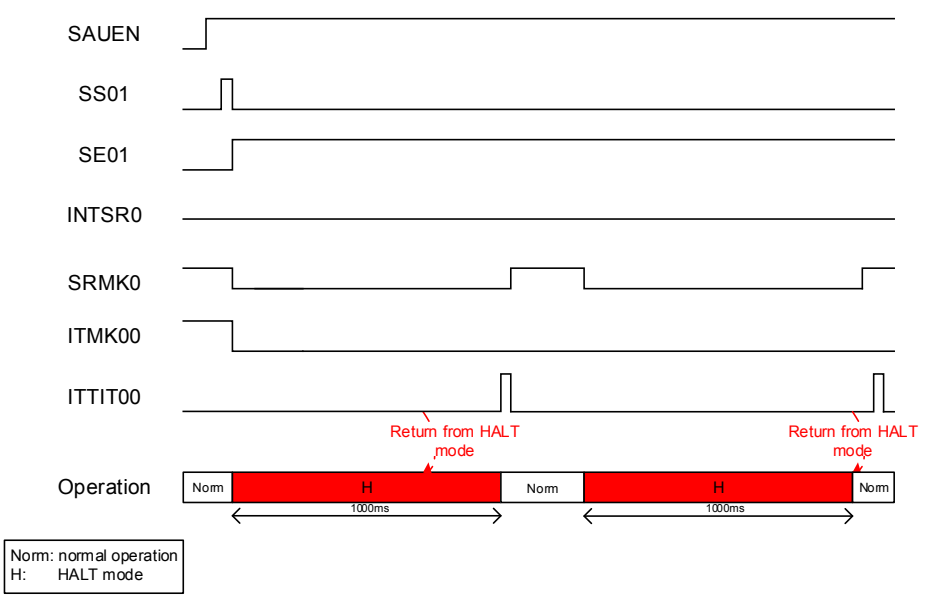

Figure 5.3 Timing of Return from HALT Mode (8-bit interval timer compare match interrupt)

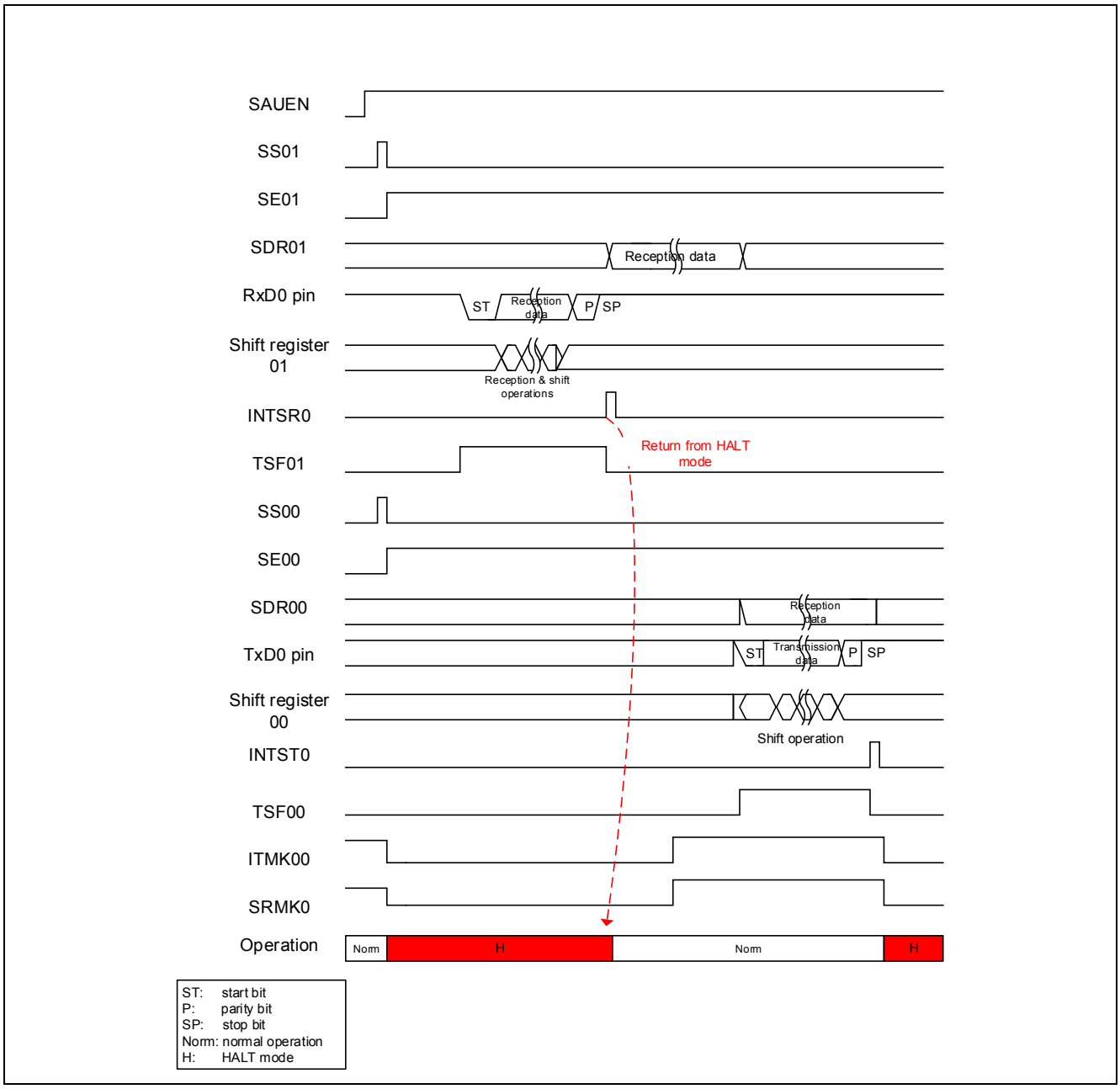

Figure 5.4 Timing of Return from HALT Mode (UART reception transfer complete)

5.2 Middle-speed On-chip Oscillator Temperature Characteristics

Changes in temperature cause both the middle-speed on-chip oscillator frequency and the UART baud rate to change. Large discrepancies in the baud rate can cause UART communication errors. To prevent such errors during operations, baud rate correction targeting 0.1 %/°C + α is essential.

Figure 5.5 shows the middle-speed on-chip oscillator temperature characteristics.

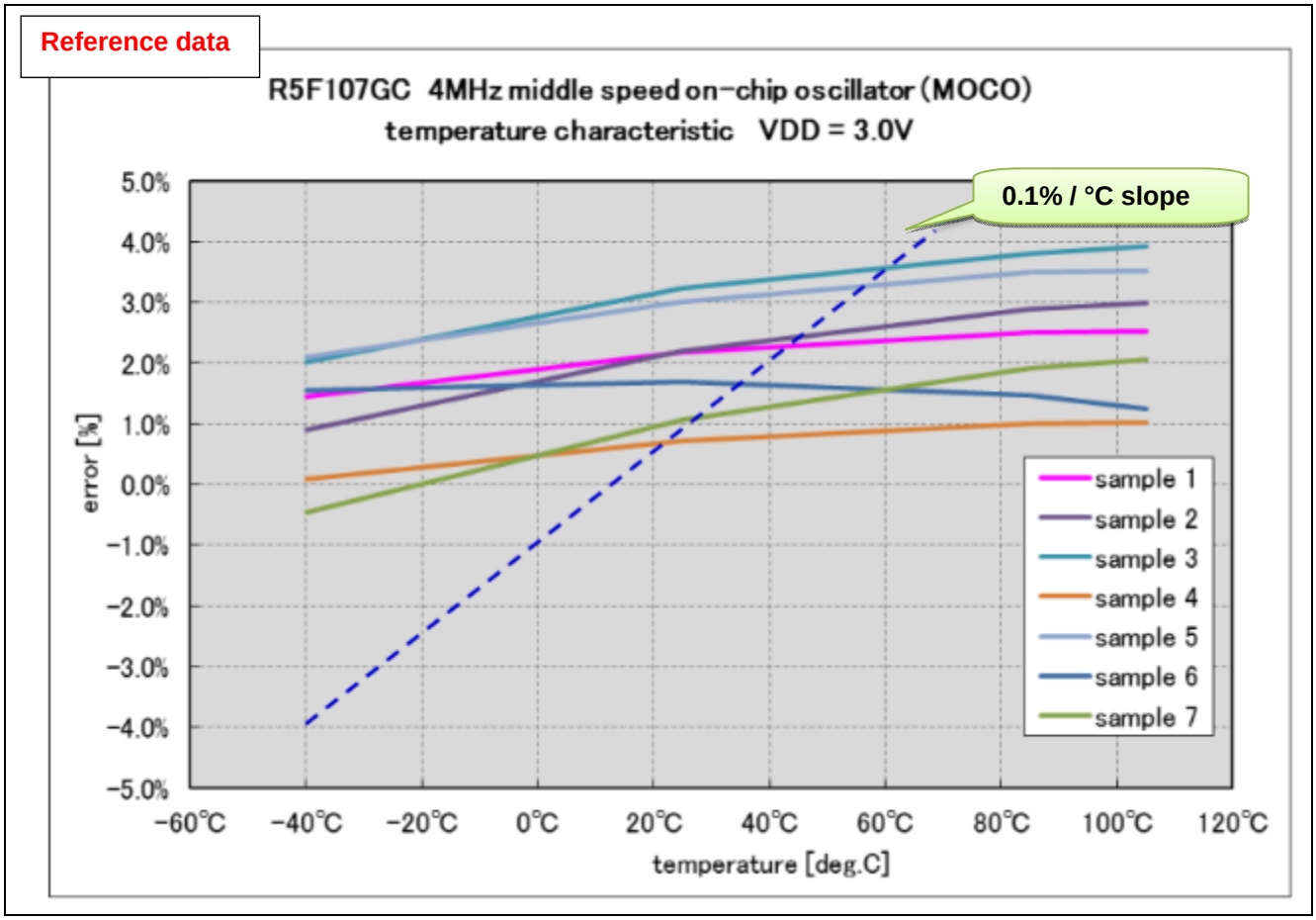

Figure 5.5 Middle-speed On-chip Oscillator Temperature Characteristics

5.3 Optional Byte Settings

Table 5.3 shows the list of settings for optional bytes.

Table 5.3 Optional Byte Settings

Address	Setting	Description
000C0H/010C0H	1110 1111B	Watchdog timer operation prohibited (stop count after reset release)
000C1H/010C1H	1111 1111B	LVD off
000C2H/010C2H	1010 1011B	LS mode, HOCO: 4MHz
000C3H/010C3H	0000 0100B	On-chip debugging prohibited

5.4 Constants

Table 5.4 lists the constants used in the sample program.

Table 5.4 Constants

Constant Name	Setting	Description
MessageOK[4]	"OK¥r¥n"	Message returned when “T” is received.
Messageok[4]	"ok¥r¥n"	Message returned when “t” is received.
MessageUC[4]	"UC¥r¥n"	Message returned when data other than “T” or “t” is received.
MessageFE[4]	"FE¥r¥n"	Message returned in response to framing error.
MessagePE[4]	"PE¥r¥n"	Message returned in response to parity error.
MessageOE[4]	"OE¥r¥n"	Message returned in response to overrun error.
Chg_Baudrate[3]	9600, 19200, 38400	Baud rate value storage (9600bps / 19200bps / 38400bps) Change baud rate by setting Chg_Baudrate[n](n=0-2) (in r_cg_userdefine.h) to g_baudrate. *Values 0 to 2 represent the following baud rates: 0: 9600 1: 19200 2: 38400 Use the following to set 38400bps. #define BAUDRATE_SET Chg_Baudrate[2]
BAUDRATE_SET	Chg_Baudrate[0] to Chg_Baudrate[2]	Acceptable baud rate values *Program will not run properly if value other than 0 to 2 is set.

5.5 Variables

Table 5.5 lists the global variables used in the sample program.

Table 5.5 Global Variables

Type	Variable Name	Contents	Function Used
uint8_t	g_uart0rxbuf	Receive data buffer	main R_MAIN_UserInit
uint8_t	gp_uart0_tx_address	Receive data pointer	R_UART0_Send, r_uart0_interrupt_send
uint16_t	g_uart0_tx_count	Receive 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
MDSTATUS	g_uart0txend	UART transmission processing complete flag	main r_uart0_callback_sendend
uint8_t	g_uart0afteradjustment	Baud rate correction complete state flag	main r_it8bit0_channel0_interrupt
uint8_t	g_uart0rxerr	UART error receive	main r_uart0_callback_receiveend r_uart0_callback_error
uint16_t	g_fmcrl	Frequency measurement circuit result	r_it8bit0_channel0_interrupt
uint16_t	g_sdr	Baud rate correction result	r_it8bit0_channel0_interrupt

5.6 Functions

Table 5.6 shows the list of functions used in the sample code.

Table 5.6 Functions

Function Name	Outline
hdwinit	Initialization function
R_Systeminit	Peripheral function initialization function
R_PORT_Create	Port initialization function
R_CGC_Create	CPU clock initialization function
R_FMC_Create	Frequency measurement circuit initialization
R_IT8Bit0_Channel0_Create	8-bit interval timer initialization function
R_SAU0_Create	SAU0 initialization function
R_UART0_Create	UART0 initialization function
main	Main processing function
R_MAIN_UserInit	Main user initialization function
R_UART0_Receive	UART0 reception status initialization function
R_UART0_Start	UART0 operation start function
R_IT8Bit0_Channel0_Start	8-bit interval timer operation start function
R_IT8Bit0_Channel0_Stop	8-bit interval timer operation stop function
R_UART0_Send	UART0 data transmission function
r_it8bit0_channel0_interrupt	8-bit interval timer interrupt function
R_UART0_Stop	UART0 operation stop function
R_FMC_Start	Frequency measurement circuit operation start function
R_FMC_Stop	Frequency measurement circuit operation stop function
r_uart0_interrupt_receive	UART0 reception complete interrupt function
r_uart0_callback_softwareoverrun	UART0 receive data number overrun processing function
r_uart0_callback_receiveend	UART0 reception complete processing function
r_uart0_interrupt_error	UART0 error interrupt function
r_uart0_callback_error	UART0 reception error processing function
r_uart0_interrupt_send	UART0 transmission complete interrupt function
r_uart0_callback_sendend	UART0 transmission complete processing function

5.7 Function Specifications

The following shows function specifications for the sample code.

Function name: hdwinit

Outline	Initialization function
Header	None
Declaration	void hdwinit(void)
Description	Initial setup of peripheral functions.
Argument	None
Return value	None
Additional notes	None

Function name: R_Systeminit

Outline	Peripheral function initialization function
Header	None
Declaration	void R_Systeminit(void)
Description	Initial setup of peripheral functions described in this application note.
Argument	None
Return value	None
Additional notes	None

Function name: R_PORT_Create

Outline	Port initialization function
Header	r_cg_port.h
Declaration	void R_PORT_Create(void)
Description	Initializes ports.
Argument	None
Return value	None
Additional notes	None

Function name: R_CGC_Create

Outline	CPU clock initialization function
Header	r_cg_cgc.h
Declaration	void R_CGC_Create(void)
Description	Initial setup of CPU clock.
Argument	None
Return value	None
Additional notes	None

Function name: R_FMC_Create

Outline	Frequency measurement circuit initialization function
Header	r_cg_fmc.h
Declaration	void R_FMC_Create(void)
Description	Initial setup of frequency measurement circuit.
Argument	None
Return value	None
Additional notes	None

Function name: R_IT8Bit0_Channel0_Create

Outline	8-bit interval timer initialization function
Header	r_cg_it8bit.h
Declaration	void R_IT8Bit0_Channel0_Create (void)
Description	Initial setup of 8-bit interval timer
Argument	None
Return value	None
Additional notes	None

Function name: R_SAU0_Create

Outline	SAU0 initialization function
Header	r_cg_sau.h
Declaration	void R_ SAU0_Create(void)
Description	Initial setup of SAU0.
Argument	None
Return value	None
Additional notes	None

Function name: R_UART0_Create

Outline	UART0 initialization function
Header	r_cg_sau.h
Declaration	void R_UART0_Start (void)
Description	Initial setup of UART0.
Argument	None
Return value	None
Additional notes	None

Function name: main

Outline	Main function
Header	None
Declaration	void main(void)
Description	After initial main setup function is executed, this function waits for a UART reception transmission interrupt in the HALT state. When the UART reception complete interrupt is generated, the MCU responds by proceeding on to the UART transmission processing. If the UART reception is not carried out due to release of HALT mode, the program does not proceed to the UART transmission processing, but returns to the HALT state again.
Argument	None
Return value	None
Additional notes	None

Function name: R_MAIN_UserInit

Outline	Main function initialization function
Header	None
Declaration	void R_MAIN_UserInit(void)
Description	This function first enables UART0 and 8-bit interval timer operations, and then enables interrupts with the EI instruction.
Argument	None
Return value	None
Additional notes	None

Function name: R_UART0_Receive

Outline	UART0 reception status initialization function	
Header	r_cg_sau.h	
Declaration	MD_STATUS R_UART0_Receive(uint8_t * const rx_buf, uint16_t rx_num)	
Description	Initial setup of UART0 reception status	
Argument	uint8_t* const rx_buf	: Address of receive data buffer
	uint16_t* const rx_num	: Size of receive data buffer
Return value	MD_OK: reception setup completed	
	MD_ARGERROR: reception setup failure	
Additional notes	None	

Function name: R_UART0_Start

Outline	UART0 operation start function
Header	r_cg_sau.h
Declaration	void R_UART0_Start (void)
Description	Sets UART0 to operation enabled state.
Argument	None
Return value	None
Additional notes	None

Function name: R_IT8Bit0_Channel0_Start

Outline	8-bit interval timer operation start function
Header	r_cg_it8bit.h
Declaration	void R_IT8Bit0_Channel0_Start (void)
Description	Sets 8-bit interval timer to operation enabled state.
Argument	None
Return value	None
Additional notes	None

Function name: R_IT8Bit0_Channel0_Stop

Outline	8-bit interval timer operation stop function
Header	r_cg_it8bit.h
Declaration	void R_IT8Bit0_Channel0_Stop (void)
Description	Sets 8-bit interval timer to operation disabled state.
Argument	None
Return value	None
Additional notes	None

Function name: R_UART0_Send

Outline	UART0 data transmission function	
Header	r_cg_sau.h	
Declaration	MD_STATUS R_UART0_Send(uint8_t * const tx_buf, uint16_t tx_num);	
Description	Sets up and starts UART0 transmission.	
Argument	uint8_t* const tx_buf	Address of transmission data buffer
	uint16_t* const tx_num	Size of transmission data buffer
Return value	[MD_OK]: transmission setup complete [MD_ARGERROR]: transmission setup failure	
Additional notes	None	

Function name: r_it8bit0_channel0_interrupt

Outline	8-bit interval timer interrupt function
Header	r_cg_it8bit.h
Declaration	__interrupt static void r_it8bit0_channel0_interrupt(void)
Description	This function performs periodic baud rate corrections (every second in this application). First, UART communication is disabled. Next, the frequency measurement circuit is started up. Based on the measurement results of the frequency measurement circuit, the clock division value is then adjusted to bring the UART baud rate as close to the ideal value as possible. UART communication is subsequently enabled and the program returns to the UART communication wait state.
Argument	None
Return value	None
Additional notes	None

Function name: R_UART0_Stop

Outline	UART0 operation stop function
Header	r_cg_sau.h
Declaration	void R_UART0_Stop (void)
Description	Sets UART0 to operation disabled state.
Argument	None
Return value	None
Additional notes	None

Function name: R_FMC_Start

Outline	Frequency measurement circuit operation start function
Header	r_cg_fmc.h
Declaration	void R_FMC_Start (void)
Description	Sets frequency measurement circuit to operation enabled state.
Argument	None
Return value	None
Additional notes	None

Function name: R_FMC_Stop

Outline	Frequency measurement circuit operation stop function
Header	r_cg_fmc.h
Declaration	void R_FMC_Stop (void)
Description	Sets frequency measurement circuit to operation disabled state.
Argument	None
Return value	None
Additional notes	None

Function name: r_uart0_interrupt_receive

Outline	UART0 reception complete processing function
Header	r_cg_sau.h
Declaration	__interrupt static void r_uart0_interrupt_receive(void)
Description	Stores received data in RAM and updates address and number of reception data.
Argument	None
Return value	None
Additional notes	None

Function name: r_uart0_callback_softwareoverrun

Outline	UART0 receive data number overrun processing function
Header	r_cg_sau.h
Declaration	static void r_uart0_callback_softwareoverrun (uint16_t rx_data)
Description	This function is called when the specified number of reception data is overrun.
Argument	uint16_t rx_data
Return value	None
Additional notes	This sample code does not perform the overrun processing. The user may add an overrun program if necessary.

Function name: r_uart0_callback_receiveend

Outline	UART0 reception complete processing function
Header	r_cg_sau.h
Declaration	static void r_uart0_callback_receiveend(void)
Description	Clears reception error flag.
Argument	None
Return value	None
Additional notes	None

Function name: r_uart0_interrupt_error

Outline	UART0 error interrupt function
Header	r_cg_sau.h
Declaration	__interrupt static void r_uart0_interrupt_error(void)
Description	Stores reception data in RAM, sets detected error to r_uart0_callback_error for response to target device.
Argument	None
Return value	None
Additional notes	None

Function name: r_uart0_callback_error

Outline	UART0 reception error processing function	
Header	r_cg_sau.h	
Declaration	static void r_uart0_callback_error(uint8_t err_type)	
Description	Sets up data transmission flag corresponding to error.	
Argument	err_type	Error type
Return value	None	
Additional notes	None	

Function name: r_uart0_interrupt_send

Outline	UART0 transmission complete interrupt function
Header	r_cg_sau.h
Declaration	__interrupt static void r_uart0_interrupt_send(void)
Description	Transmits data and then updates pointer and counter. If no data remains, starts transmission complete processing.
Argument	None
Return value	None
Additional notes	None

Function name: r_uart0_callback_sendend

Outline	UART0 transmission complete processing function
Header	r_cg_sau.h
Declaration	static void r_uart0_callback_receiveend(void)
Description	Notifies transmission completion.
Argument	None
Return value	None
Additional notes	None

5.8 Flowcharts

Figure 5.6 shows the entire flow of the sample code.

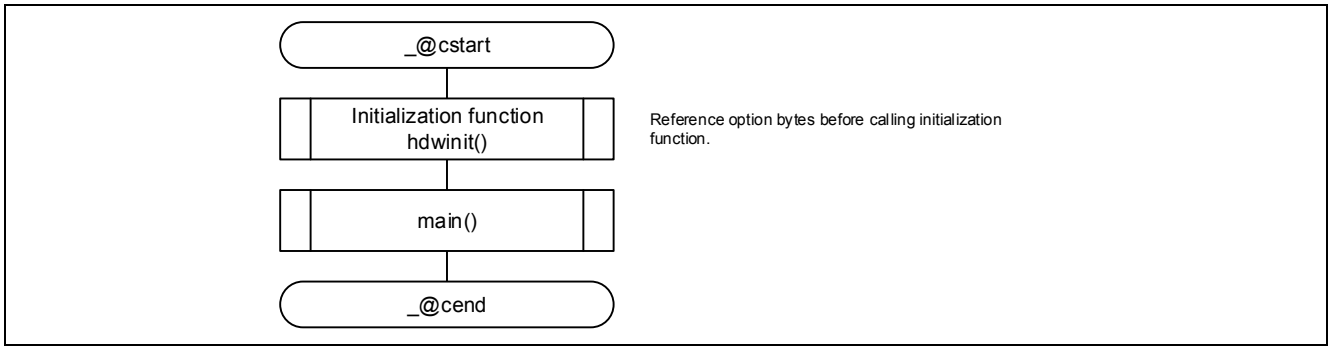

Figure 5.6 Entire Flow

5.8.1 Initialization

Figure 5.7 shows the flowchart of the initialization process.

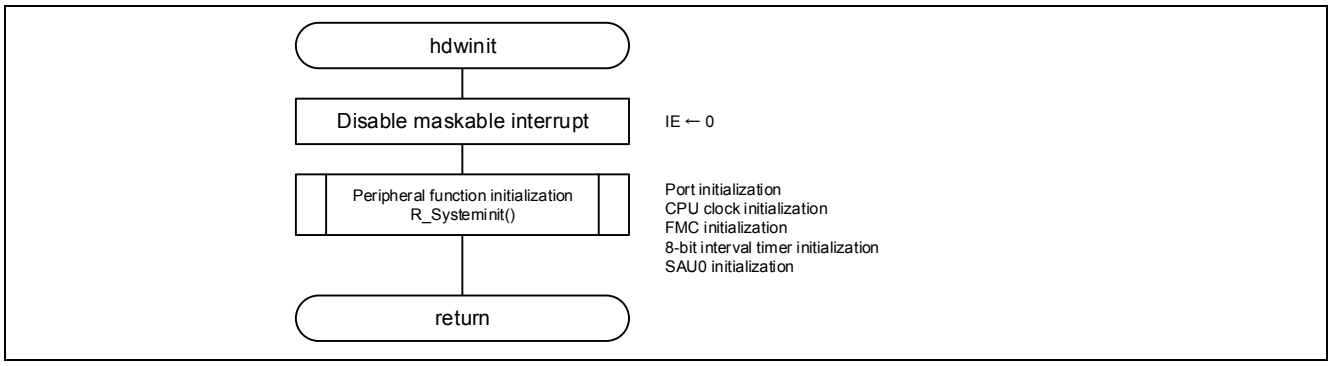

Figure 5.7 Initialization

Note: Initialization in the IAR sample code is executed in the __low_level_init function.

5.8.2 Peripheral function initialization

Figure 5.8 shows the flowchart for peripheral function initialization.

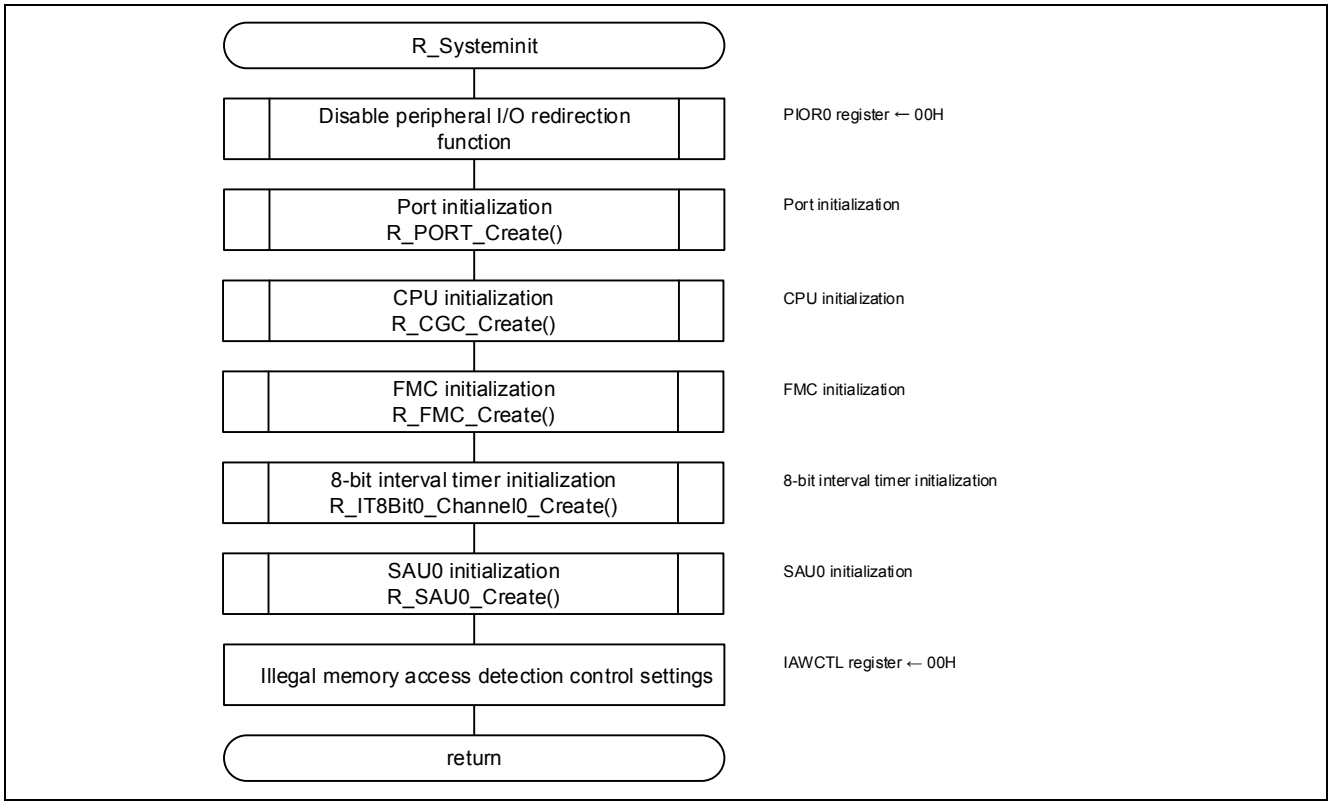

Figure 5.8 Peripheral Function Initialization

5.8.3 Port initialization

Figure 5.9 shows the flowchart for port initialization.

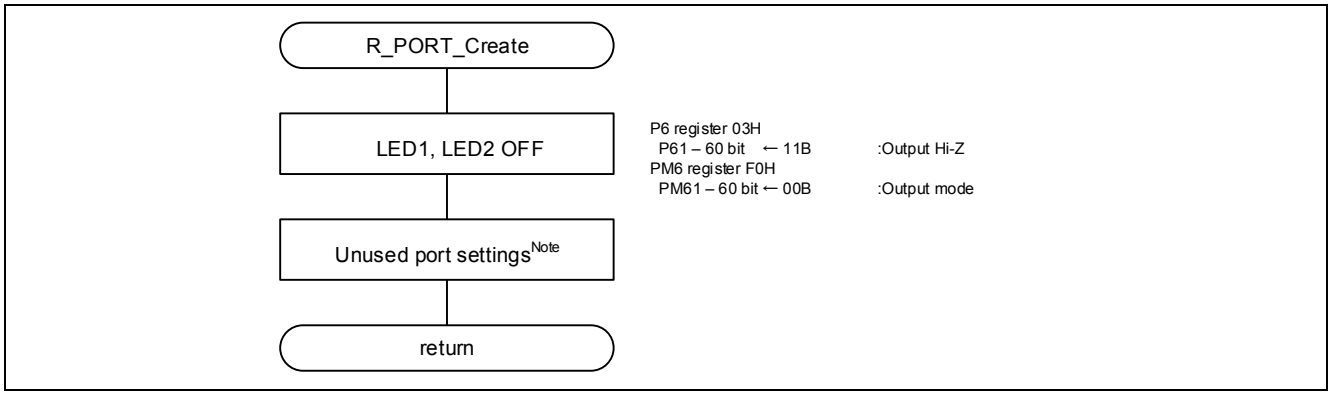

Figure 5.9 Port Initialization

Note Refer to the initialization flowchart in the RL78/G13 Initialization (R01AN0451J) Application Note for details on how to set unused ports.

Note When designing circuits, always make sure unused ports are properly processed and all electrical characteristics are met. Also make sure each unused input-only port is connected to V_{DD} or V_{SS} through a resister.

5.8.4 CPU clock initialization

Figure 5.10 shows the flowchart for CPU clock initialization.

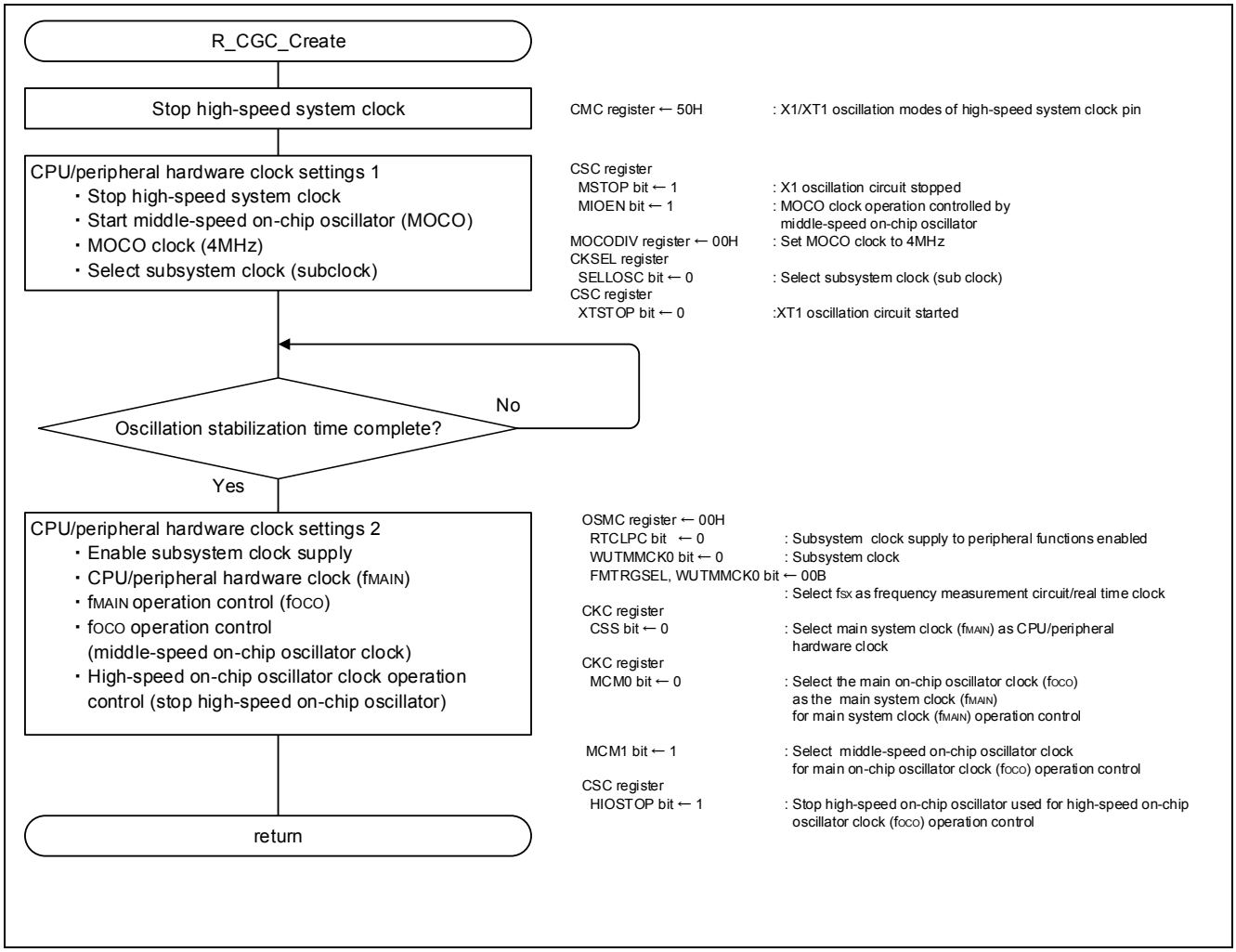

Figure 5.10 CPU Clock Initialization

5.8.5 Frequency measurement circuit initialization

Figure 5.11 shows the flowchart for frequency measurement circuit initialization.

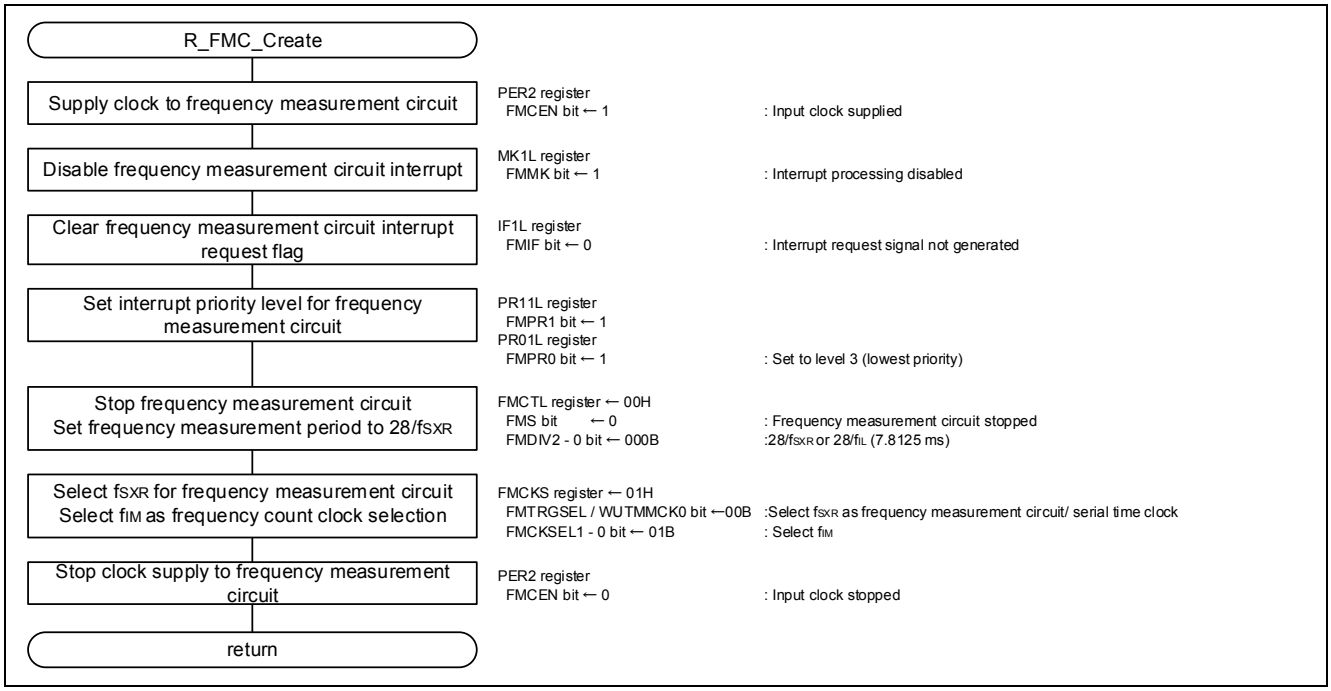

Figure 5.11 Frequency Measurement Circuit Initialization

Frequency Measurement Circuit Clock Supply

・Peripheral enable register 2 (PER2)
Supply clock to frequency measurement circuit.

Symbol: PER2

7	6	5	4	3	2	1	0
TMKAEN	FMCEN	DOCEN	0	0	0	0	0
x	**1**	x	x	x	x	x	x

Bit 6

FMCEN	Input clock supply control for frequency measurement circuit
0	Stops input clock supply
1	**Enables input clock supply**

Frequency Measurement Circuit Interrupt Disable

・Interrupt request flag register (MK1L)
Disable frequency measurement circuit interrupt.
・Interrupt request flag register (IF1L)
Clear frequency measurement circuit interrupt request flag.

Symbol: MK1L

7	6	5	4	3	2	1	0
0	0	TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	FMMK
x	x	x	x	x	x	x	**1**

Bit 0

FMMK	Interrupt processing control
0	Enables interrupt processing
1	**Disables interrupt processing**

Symbol: IF1L

7	6	5	4	3	2	1	0
0	0	TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	FMIF
x	x	x	x	x	x	x	**0**

Bit 0

FMIF	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

Frequency Measurement Circuit Interrupt Priority Level Setting

・Priority level flag registers (PR11L, PR01L)
Set interrupt to level 3 (lowest priority level).

Symbol: PR11L

7	6	5	4	3	2	1	0
0	0	TMPR103	TMPR102	TMPR101	TMPR103H	TMPR101H	FMPR1
x	x	x	x	x	x	x	**1**

Symbol: PR01L

7	6	5	4	3	2	1	0
0	0	TMPR003	TMPR002	TMPR001	TMPR003H	TMPR001H	FMPR0
x	x	x	x	x	x	x	**1**

Bit 0

FMPR1	FMPR0	Selection of priority level
0	0	Set to level 0 (highest priority level)
0	1	Set to level 1
1	0	Set to level 2
1	**1**	**Set to level 3 (lowest priority level)**

Frequency Measurement Circuit Stop and Time Setting

・Frequency measurement control register (FMCTL)
Set frequency measurement circuit to stop state.
Set frequency measurement time to $2^8/f_{SXR}$.

Symbol: FMCTL

7	6	5	4	3	2	1	0
FMS	0	0	0	0	FMDIV2	FMDIV1	FMDIV0
0	x	x	x	x	**0**	**0**	**0**

Bit 7

FMS	Enable frequency measurement circuit operation
0	**Enables frequency measurement circuit operation**
1	Disables frequency measurement circuit operation

Bit 2 - 0

FMDIV2	FMDIV1	FMDIV0	Frequency measurement period setting
0	**0**	**0**	**$2^8/f_{SXR}$ or $2^8/f_{IL}$ (7.8125 ms)**
0	0	1	$2^9/f_{SXR}$ or $2^9/f_{IL}$ (15.625 ms)
0	1	0	$2^{10}/f_{SXR}$ or $2^{10}/f_{IL}$ (31.25 ms)
0	1	1	$2^{11}/f_{SXR}$ or $2^{11}/f_{IL}$ (62.5 ms)
1	0	0	$2^{12}/f_{SXR}$ or $2^{12}/f_{IL}$ (0.125 s)
1	0	1	$2^{13}/f_{SXR}$ or $2^{13}/f_{IL}$ (0.25 s)
1	1	0	$2^{14}/f_{SXR}$ or $2^{14}/f_{IL}$ (0.5 s)
1	1	1	$2_{15}/f_{SXR}$ or $2^{15}/f_{IL}$ (1s)

Frequency Measurement Circuit Count Operation and Frequency Count Clock Settings

・Frequency measurement clock selection register (FMCKS)
Select f_{SXR} for frequency measurement clock.
Set f_{IM} for frequency count clock selection.

Symbol: FMCKS

7	6	5	4	3	2	1	0
0	0	0	FMTRGSEL	0	0	FMCKSEL1	FMCKSEL0
x	x	x	**0**	x	x	**0**	**1**

Bit 4

FMTRGSEL	WUTMMCK0	Selection of frequency measurement circuit count operation/stop trigger clock/real-time clock operation clock
0	**0**	**Select f_{SXR} for frequency measurement circuit/real-time clock**
0	1	Select f_{IL} for real-time clock (constant-period interrupt function)
1	0	Disable setting
1	1	Select f_{IL} for frequency measurement circuit

Bit 1 - 0

FMCKSEL1	FMCKSEL0	Selection of frequency count clock
0	0	Select f_{MX}
0	**1**	**Select f_{IM}**
1	x	Select f_{IH}

Frequency Measurement Circuit Clock Supply Stop

・Peripheral enable register (PER2)
Stop supply of clock to frequency measurement circuit.

Symbol: PER2

7	6	5	4	3	2	1	0
TMKAEN	FMCEN	DOCEN	0	0	0	0	0
x	**0**	x	x	x	x	x	x

Bit 6

FMCEN	Frequency measurement circuit input clock supply control
0	**Stops input clock supply**
1	Enables input clock supply

5.8.6 8-bit interval timer initialization

Figure 5.12 shows the flowchart for 8-bit interval timer initialization.

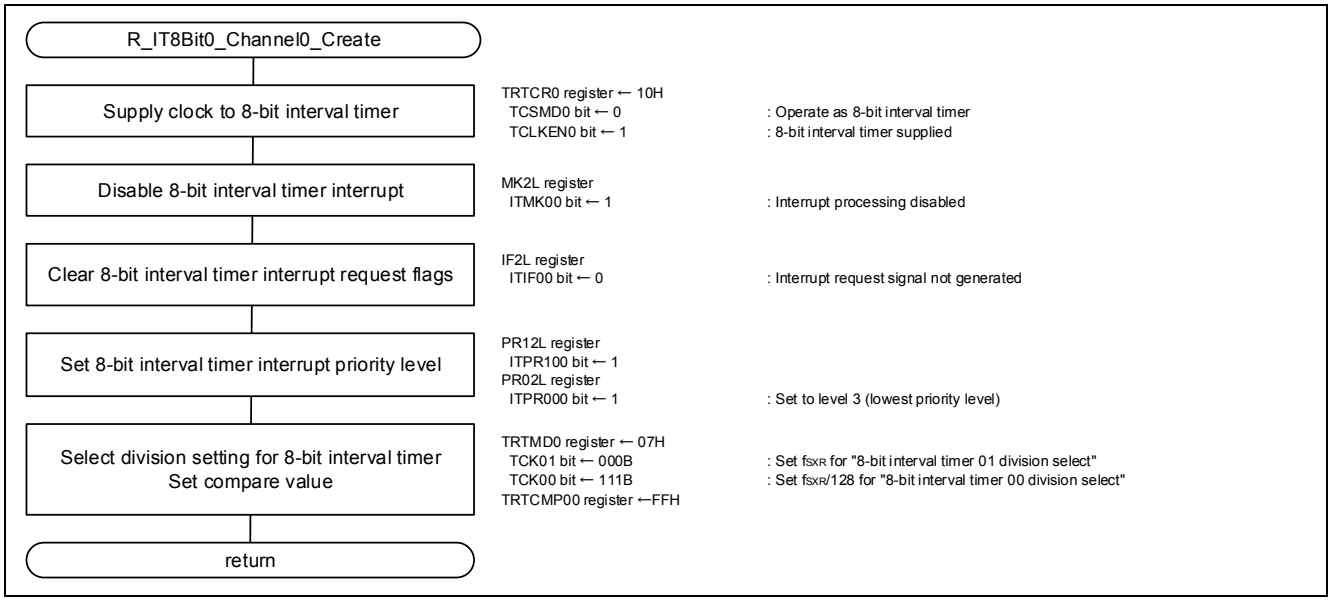

Figure 5.12 8-bit Interval Timer Initialization

8-bit Interval Timer Clock Supply

- 8-bit interval timer clock register 0 (TRTCR0)
  Set to operate as 8-bit counter.
  Supply clock to 8-bit interval timer

Symbol: TRTCR0

7	6	5	4	3	2	1	0
TCSMD0	0	0	TCLKEN0	0	TSTART01	0	TSTART00
0	x	x	**1**	x	**0**	x	**0**

Bit 7

TCSMD0	Mode selection
0	**Operate as 8-bit counter**
1	Operate as 16-bit counter (channel 0 and channel 1)

Bit 4

TCLKEN0	8-bit interval timer clock enable
0	Stops clock
1	**Enables clock supply**

Bit 2

TSTART01	8-bit interval timer 1 count start
0	**Stops count**
1	Starts count

Bit 0

TSTART00	8-bit interval timer 0 count start
0	**Stops count**
1	Starts count

8-bit Interval Timer Interrupt Disable

- Interrupt request flag register (MK2L)
  Disable 8-bit interval timer interrupt
- Interrupt request flag register (IF2L)
  Clear 8-bit interval timer interrupt request flag.

Symbol: MK2L

7	6	5	4	3	2	1	0
FLMK	0	0	0	ITMK11	ITMK10	ITMK01	ITMK00
x	x	x	x	x	x	x	**1**

Bit 0

ITMK00	Interrupt processing control
0	Enables interrupt processing
1	**Disables interrupt processing**

Symbol: IF2L

7	6	5	4	3	2	1	0
FLIF	0	0	0	ITIF11	ITIF10	ITIF01	ITIF00
x	x	x	x	x	x	x	**0**

Bit 0

ITIF00	Interrupt request flag
0	**Interrupt request signal is not generated**
1	Interrupt request signal is generated, interrupt request state

8-bit Interval Timer Interrupt Priority Level Setting

・Priority flag register (PR12L, PR02L)
Set 8-bit interval timer interrupt to level 3 (lowest priority level).

Symbol: PR12L

7	6	5	4	3	2	1	0
FLPR1	0	0	0	ITPR111	ITPR110	ITPR101	ITPR100
x	x	x	x	x	x	x	**1**

Symbol: PR02L

7	6	5	4	3	2	1	0
FLPR0	0	0	0	ITPR011	ITPR010	ITPR001	ITPR000
x	x	x	x	x	x	x	**1**

Bit 0

ITPR100	ITPR000	Priority level selection
0	0	Set to level 0 (highest priority level)
0	1	Set to level 1
1	0	Set to level 2
1	**1**	**Set to level 3 (lowest priority level)**

8-bit Interval Timer Division Ratio Setting

・8-bit interval timer division register 0 (TRTMD0)
Set frequency measurement circuit to stop.
Set frequency measurement period TCK01 to f_{SXR} and TCK00 to f_{SXR}/128.

Symbol: TRTMD0

7	6	5	4	3	2	1	0
x	TCK01			x	TCK00		
x	**0**	**0**	**0**	x	**1**	**1**	**1**

Bit 6 - 4

TCK01			8-bit interval timer 1 division selection
Bit 6	Bit 5	Bit 4	
0	**0**	**0**	**f_{SXR} or $2^8/f_{IL}$**
0	0	1	$f_{SXR}/2$ or $f_{IL}/2$
0	1	0	$f_{SXR}/4$ or $f_{IL}/4$
0	1	1	$f_{SXR}/8$ or $f_{IL}/8$
1	0	0	$f_{SXR}/16$ or $f_{IL}/16$
1	0	1	$f_{SXR}/32$ or $f_{IL}/32$
1	1	0	$f_{SXR}/64$ or $f_{IL}/64$
1	1	1	$f_{SXR}/128$ or $f_{IL}/128$

Bit 2 – 0

TCK00			8-bit interval timer 0 division selection
Bit 2	Bit 1	Bit 0	
0	0	0	f_{SXR} or $2^8/f_{IL}$
0	0	1	$f_{SXR}/2$ or $f_{IL}/2$
0	1	0	$f_{SXR}/4$ or $f_{IL}/4$
0	1	1	$f_{SXR}/8$ or $f_{IL}/8$
1	0	0	$f_{SXR}/16$ or $f_{IL}/16$
1	0	1	$f_{SXR}/32$ or $f_{IL}/32$
1	1	0	$f_{SXR}/64$ or $f_{IL}/64$
1	**1**	**1**	**$f_{SXR}/128$ or $f_{IL}/128$**

8-bit Interval Timer Channel 1 Compare Value Setting

・8- bit interval timer compare register 0 (TRTCMP0)
Set 8- bit interval timer compare to FFH.

Symbol: TRTCMP0

8	7	6	5	4	3	2	1	0
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

5.8.7 SAU0 Initialization

Figure 5.13 shows the flowchart for SAU0 initialization.

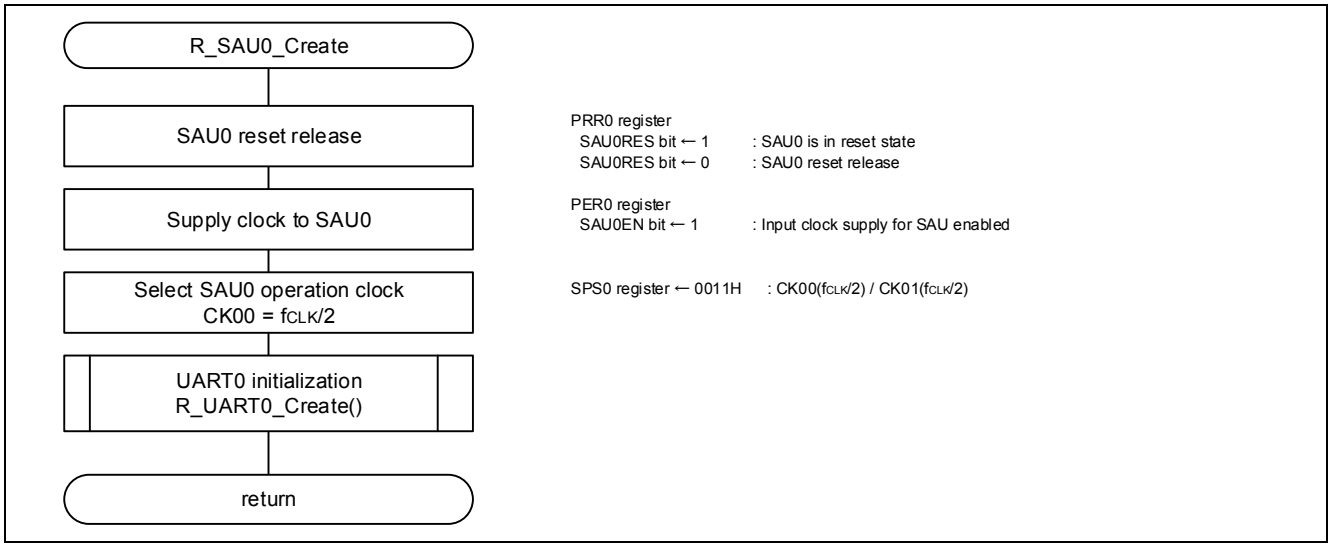

Figure 5.13 SAU0 Initialization

SAU0 Reset Release

・Peripheral reset control register 0 (PRR0)
Release SAU0 from the reset state.

Symbol: PRR0

7	6	5	4	3	2	1	0
0	0	ADCRES	0	0	SAU0RES	0	TAU0RES
x	x	x	x	x	**0**	x	x

Bit 2

SAU0RES	Serial Array Unit reset control
0	**Serial array unit reset release**
1	Serial array unit reset state

SAU0 Input Clock Supply

・Peripheral enable register 0（PER0）
Enable clock supply to SAU0.

Symbol: PER0

7	6	5	4	3	2	1	0
RTCEN	0	ADCEN	0	0	SAU0EN	0	TAU0EN
x	x	x	x	x	**1**	x	x

Bit 2

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

SAU0 Operation Clock Selection

・Serial clock selection register 0 (SPS0)
Set to 12MHz.

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

Bit 7 - 4, 3 - 0 (n = 1, 0)

PRS 0n3	PRS 0n2	PRS 0n1	PRS 0n0	Operation Clock (CK0n) Selection					
					f_{CLK}= 2MHz	f_{CLK}= 5MHz	f_{CLK}= 10MHz	f_{CLK}= 20MHz	f_{CLK}= 24MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	**0**	**0**	**1**	**f_{CLK}/2**	1 MHz	2.5 MHz	5 MHz	10 MHz	**12 MHz**
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977Hz	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	732 Hz

5.8.8 UART0 Initialization

Figure 5.14 shows the flowchart for UART0 initialization (1/2).

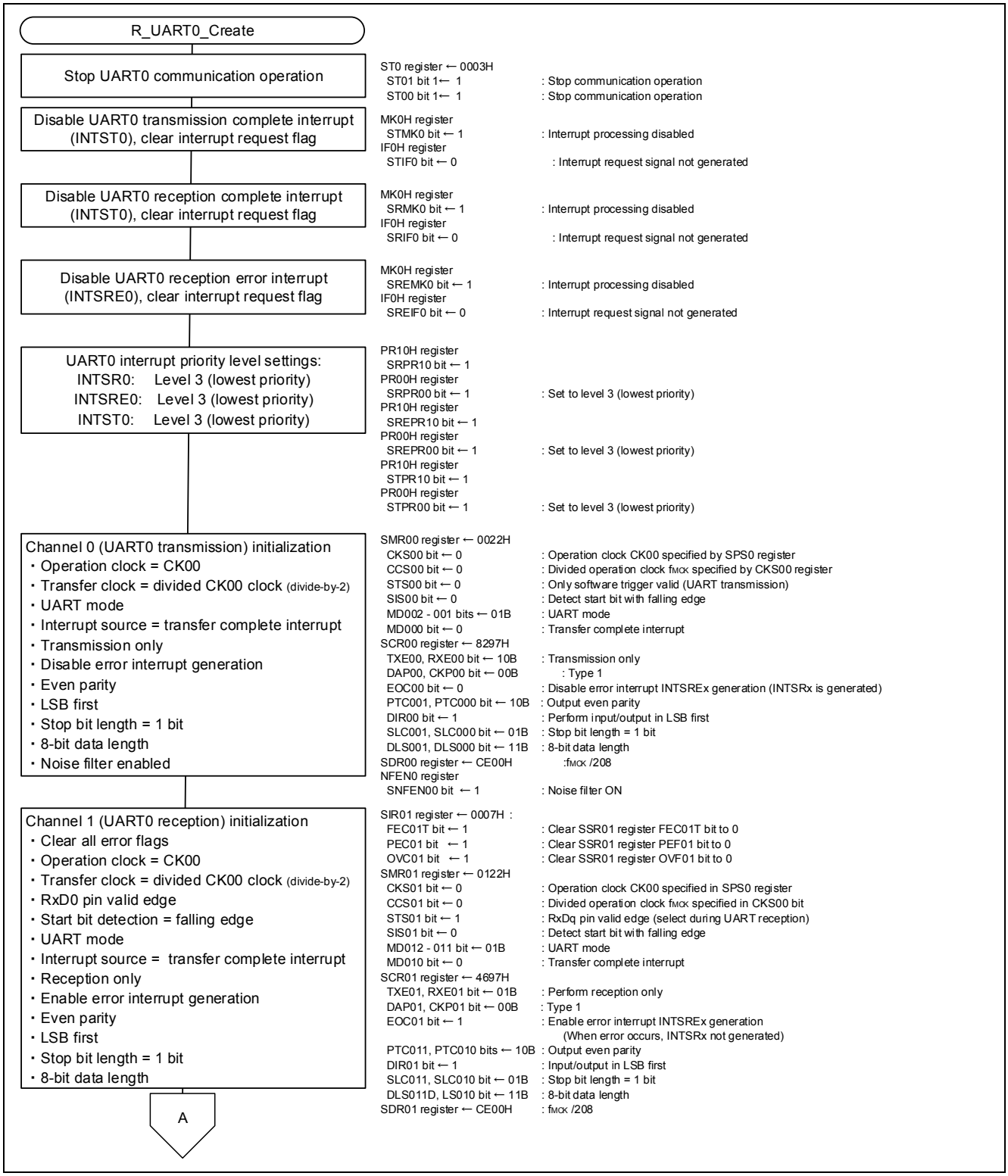

Figure 5.14 UART0 Initialization (1/2)

Figure 5.15 shows the flowchart for UART0 initialization (2/2).

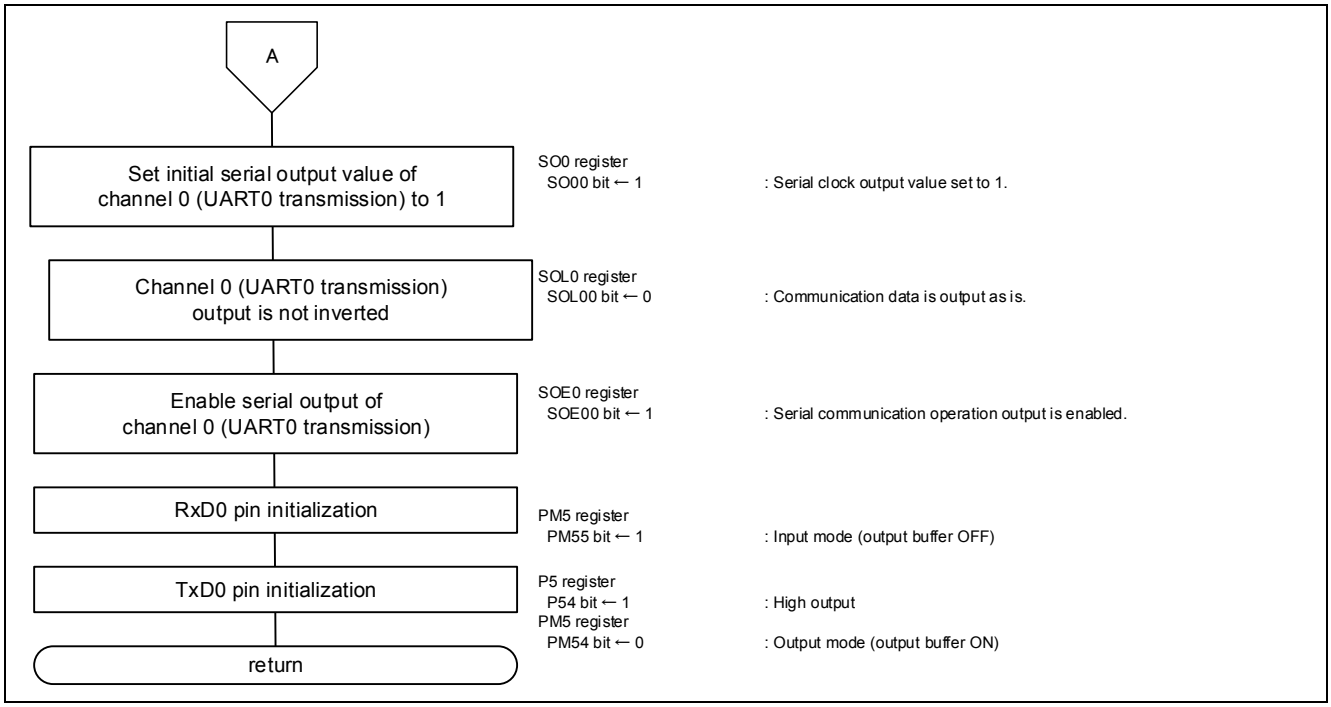

Figure 5.15 UART0 Initialization (2/2)

UART0 Communication Operation Stop

・Serial channel stop register 0 (ST0)

Set communication operation to stop state.

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	0	0	ST01	ST00
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	**1**

Bit 1

ST01	Channel n operation start trigger
0	No trigger operation
1	**SEmn bit is cleared to 0, communication operation is stopped.**

Bit 0

ST00	Channel n operation start trigger
0	No trigger operation
1	**SEmn bit is cleared to 0, communication operation is stopped.**

UART0 Transmission&Reception / Reception Error Interrupt Complete Disable

- Interrupt request flag register (MK0H)
  Disable UART0 transmission/reception/ reception error interrupts.
- Interrupt request flag register (IF0H)
  Clear UART0 UART0 transmission/reception/ reception error interrupt flags.

Symbol: MK0H

7	6	5	4	3	2	1	0
RTITMK	TMMK00	SREMK0	1	1	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	PMK6
x	x	**1**	x	x	**1**	**1**	x

Bit 5

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

Bit 2

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

Bit 1

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

Symbol: IF0H

7	6	5	4	3	2	1	0
RTITIF	TMIF00	SREIF0	0	0	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	PIF6
x	x	**0**	x	x	**0**	**0**	x

Bit 5

SREIF0	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

Bit 2

SRIF0	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

Bit 1

STIF0	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

UART0 Transmission/Reception/Reception Error Interrupt Priority Level Setting

・Priority level flag registers (PR10H, PR00H)
Set transmission/reception/reception error interrupts to level 3 (lowest priority level) .

Symbol: PR00H

7	6	5	4	3	2	1	0
RTITPR0	TMPR000	SREPR00	1	1	SRPR00 CSIPR001 IICPR001	STPR00 CSIPR000 IICPR000	PPR06
x	x	**1**	x	x	**1**	**1**	x

Symbol: PR10H

7	6	5	4	3	2	1	0
RTITPR1	TMPR100	SREPR10	1	1	SRPR10 CSIPR101 IICPR101	STPR10 CSIPR100 IICPR100	PPR16
x	x	**1**	x	x	**1**	**1**	x

Bit 5

SREPR00	SREPR10	Selection of priority level
0	0	Set to level 0 (highest priority level)
0	1	Set to level 1
1	0	Set to level 2
1	**1**	**Set to level 3 (lowest priority level)**

Bit 2

SRPR00	SRPR10	Selection of priority level
0	0	Set to level 0 (highest priority level)
0	1	Set to level 1
1	0	Set to level 2
1	**1**	**Set to level 3 (lowest priority level)**

Bit 1

STPR00	STPR10	Selection of priority level
0	0	Set to level 0 (highest priority level)
0	1	Set to level 1
1	0	Set to level 2
1	**1**	**Set to level 3 (lowest priority level)**

UART0 channel 0 Initialization

- Serial mode register 00 (SMR00)
  Set UART0 channel 0 as follows:
  - Operation clock: CK00
  - Transfer clock: CK00 divided clock (divide by 2)
  - Only software trigger is valid edge
  - Start bit detection: falling edge
  - UART mode
  - Interrupt source: transfer complete interrupt

Symbol: SMR00

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

Bit 15

CKS00	Selection of operation clock (f_{MCK}) for channel n
0	**Operation clock CKm0 set by SPSm register**
1	Operation clock CKm1 set in SPSm register

Bit 14

CCS00	Selection of transfer clock (f_{TCLK}) for channel n
0	**Divided operation clock f_{MCK} specified by CKSmn bit**
1	Input clock f_{SCK} from SCKp pin (slave transfer in CSI mode)

Bit 8

STS00	Selection of start trigger source
0	**Only software trigger is valid (selected for CSI, UART transmission, and simplified I2C)**
1	Valid edge of the RxDq pin (selected for UART reception)

Bit 6

SIS00	Controls inversion of receive data level of channel n in UART mode
0	**Start bit is detected with falling edge.** **The input communication data is captured as is.**
1	Start bit is detected with rising edge. The input communication data is inverted and captured.

Bit 2 - 1

MD002	MD001	Setting of operation mode for channel n
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 n
0	**Transfer complete interrupt**
1	Buffer empty interrupt Occurs when data is transferred from the SDRmn register to the shift register

UART0 Channel 0 Serial Communication Operation

- Serial communication operation setting register 00 (SCR00)
  Set UART0 channel 0 as follows:
  - Operation mode: transmission only
  - Clock phase: type 1
  - Data transmission order: LSB first
  - Data length: 8-bit data

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

Bit 15 - 14

TXE00	RXE00	Setting of operation mode of channel n
0	0	Communication disabled
0	1	Reception only
1	**0**	**Transmission only**
1	1	Transmission/reception

Bit 13 - 12

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

Bit 10

EOC00	Mask control of error interrupt signal (INTSREx (x = 0-3))
0	**Disables generation of error interrupt INTSREx ((INTSRx is generated)**
1	Enables generation of error interrupt INTSREx (INTSRx is not generated if an error occurs).

Bit 9 - 8

PTC 001	PTC 000	Setting of parity bit setting in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs parity	No parity judgment
1	**0**	**Outputs even parity.**	Judged as even parity.
1	1	Outputs odd parity.	Judged as odd parity.

Bit 7

DIR00	Setting of data transfer order in CSI and UART modes
0	Inputs/Outputs data with MSB first.
1	**Inputs/Outputs data with LSB first.**

Bit 5 - 4

SLC001	SLC000	Setting of stop bit in UART mode
0	0	No stop bit
0	**1**	**Stop bit length = 1 bit**
1	0	Stop bit length = 2 bits (mn = 00, 02, 10, 12 only)
1	1	Setting prohibited

Bit 1 - 0

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

UART0 Channel 0 Baud Rate Setting

Serial data register 00 (SDR00)

- Set transfer clock to 9600bps.

(9600bps = f_{MCK} ÷ 208 = 2MHz ÷ 208)

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**	x	x	x	x	x	x	x	x	x

Bit 15 - 9

SDR00[15:9]							Transfer clock setting based on operation clock (f_{MCK}) division
0	0	0	0	0	0	0	f_{MCK} /2
0	0	0	0	0	0	1	f_{MCK} /4
. .	. .	. .	. .	. .	. .	. .	. .
1	**1**	**0**	**0**	**1**	**1**	**1**	**f_{MCK}/208 (= f_{MCK}/{(103+1)×2})**

UART0 Channel 1 Noise Filter Setting

- Noise filter enable register 0 (NFEN0)

  Set noise filter to ON.

Symbol: NFEN0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SNFEN00
x	x	x	x	x	x	x	**1**

Bit 0

SNFEN00	Enable/Disable use of noise filter for RxD0 pin
0	Noise filter OFF
1	**Noise filter ON**

UART0 Channel 1 Error Flag Setting

- Serial flag clear trigger register 01 (SIR01)
  Clear all error flags.

Symbol: SIR01

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	FECT01	PECT01	OVCT01
—	—	—	—	—	—	—	—	—	—	—	—	—	**1**	**1**	**1**

Bit 2

FECT01	Clear trigger for framing error flag of channel n
0	Do not clear
1	**Clear SSRmn register FEFmn bit to 0.**

Bit 1

PECT01	Clear trigger for parity error flag of channel n
0	Do not clear
1	**Clear SSRmn register PEFmn bit to 0.**

Bit 0

OVCT01	Clear trigger for overrun error flag of channel n
0	Do not clear
1	**Clear SSRmn register OVFmn bit to 0.**

UART0 Channel 1 Initialization

- Serial mode register 01 (SMR01)
  Set UART0 channel 1 as follows:
  - Operation clock: CK00
  - Transfer clock: CK00 divided clock (divided-by-2)
  - RxD0 pin valid edge
  - Start bit detection: falling edge
  - UART mode
  - Interrupt source: transfer complete interrupt

Symbol: SMR01

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS01	CCS01	0	0	0	0	0	STS01	0	SIS01	1	0	0	MD012	MD011	MD010
0	**0**	x	x	x	x	x	**1**	x	**0**	x	x	x	**0**	**1**	**0**

Bit 15

CKS01	Selection of operation clock (f_{MCK}) for channel n
0	**Operation clock CKm0 set by SPSm register**
1	Operation clock CKm1 set in SPSm register

Bit 14

CCS01	Selection of transfer clock (f_{TCLK}) for channel n
0	**Divided operation clock f_{MCK} specified by CKSmn bit**
1	Input clock f_{SCK} from SCKp pin (slave transfer in CSI mode)

Bit 8

STS01	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I2C)
1	**Valid edge of the RxDq pin (selected for UART reception)**

Bit 6

SIS01	Control of inversion of receive data level of channel n in UART mode
0	**Start bit is detected with falling edge.** **The input communication data is captured as is.**
1	Start bit is detected with rising edge. The input communication data is inverted and captured.

Bit 2 - 1

MD012	MD011	Setting of operation mode for channel n
0	0	CSI mode
0	**1**	**UART mode**
1	0	Simplified I^2C mode
1	1	Setting prohibited

Bit 0

MD010	Selection of interrupt source of channel n
0	**Transfer complete interrupt**
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)

UART0 Channel 1 Serial Communication Operation Settings

- Serial communication operation setting register 01 (SCR01)
  Set UART0 channel 1 as follows:
  - Operation mode: reception only
  - Clock phase: type 1
  - Data transmission order: LSB first
  - Data length: 8-bit data

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

Bit 15 - 14

TXE01	RXE01	Setting of operation mode of channel n
0	0	Disables communication
0	**1**	**Reception only**
1	0	Transmission only
1	1	Setting of operation mode of channel n

Bit 13 - 12

DAP01	CKP01	Selection of data and clock phase in CSI mode
0	**0**	**Type 1**
0	1	Type 2
1	0	Type 3
1	1	Type 4

Bit 10

EOC01	Mask control of error interrupt signal (INTSREx (x = 0-3))
0	Disables generation of error interrupt INTSREx (INTSRx is generated)
1	**Enables generation of error interrupt INTSREx (INTSRx is not generated if an error occurs).**

Bit 9 - 8

PTC 011	PTC 010	Setting of parity bit setting in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs parity	No parity judgment
1	**0**	Outputs even parity.	**Judged as even parity.**
1	1	Outputs odd parity.	Judged as odd parity.

Bit 7

DIR01	Setting of data transfer order in CSI and UART modes
0	Inputs/Outputs data with MSB first.
1	**Inputs/Outputs data with LSB first.**

Bit 5 - 4

SLC011	SLC010	Setting of stop bit in UART mode
0	0	No stop bit
0	**1**	**Stop bit length = 1 bit**
1	0	Stop bit length = 2 bits (mn = 00, 02, 10, 12 only)
1	1	Setting prohibited

Bit 1 - 0

DLS011	DLS010	Setting of data length in CSI and UART modes
0	1	9-bit data length (stored in bits 0 to 8 of the SDRmn register) (settable in UART mode only)
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	**1**	**8-bit data length (stored in bits 0 to 7 of the SDRmn register)**
Other than above		Setting prohibited

UART0 Channel 1 Baud Rate Setting

・Serial data register 01 (SDR01)
Set transfer clock to 9600bps.
(9600bps = f_{MCK} ÷ 208 = 2MHz ÷ 208)

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**	x	x	x	x	x	x	x	x	x

Bit 15 - 9

SDR01[15:9]							Transfer clock setting based on operation clock (f_{MCK}) division
0	0	0	0	0	0	0	f_{MCK}/2
0	0	0	0	0	0	1	f_{MCK}/4
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
1	**1**	**0**	**0**	**1**	**1**	**1**	**f_{MCK}/208 (= f_{MCK}/{(103+1)×2})**

UART0 Serial Data Output Value Setting

・Serial output register 0 (SO0)
Set serial data output value to 2.

Symbol: SO0

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

Bit 0

SO0	Channel n serial data output
0	Set serial data output value to “0”
1	**Set serial data output value to “1”**

UART0 Transmission Data Setting

・Serial output level register 0 (SOL0)
Set communication data to be output as is.

Symbol: SOL0

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	0	0	SOL00
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**0**

Bit 0

SOL00	Inversion selection for transmission data level of channel n in UART mode
0	**The communication data is output as is.**
1	The communication data is inverted and then output.

UART0 Serial Output Enable

・Serial output enable register 0 (SOE0)
Enable output in 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	0	0	SOE01	SOE00
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**

Bit 0

SOE00	Serial output enable/stop for channel n
0	Stops output in serial communication operation
1	**Enables output in serial communication operation**

RxD0 Pin Initialization

・Port mode register (PM5)
Set P55 to input mode.

Symbol: PM5

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

Bit 5

PM55	Selection of input/output mode for Pmn pin
0	Output mode (functions as output port (output buffer ON))
1	**Input mode (functions as input port (output buffer OFF))**

TxD Pin Initialization

・Port register (P5)
Set P54 to High output.
・Port mode register (PM5)
Set PM54 to output mode.

Symbol: P5

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

Bit 4

P54	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	**0**	x	x	x	x

Bit 4

PM54	Selection of Pmn pin input/output mode
0	**Output mode (functions as output port (output buffer ON))**
1	Input mode (functions as input port (output buffer OFF))

5.8.9 Main Processing

Figure 5.16 shows the flowchart for main processing (1/2).

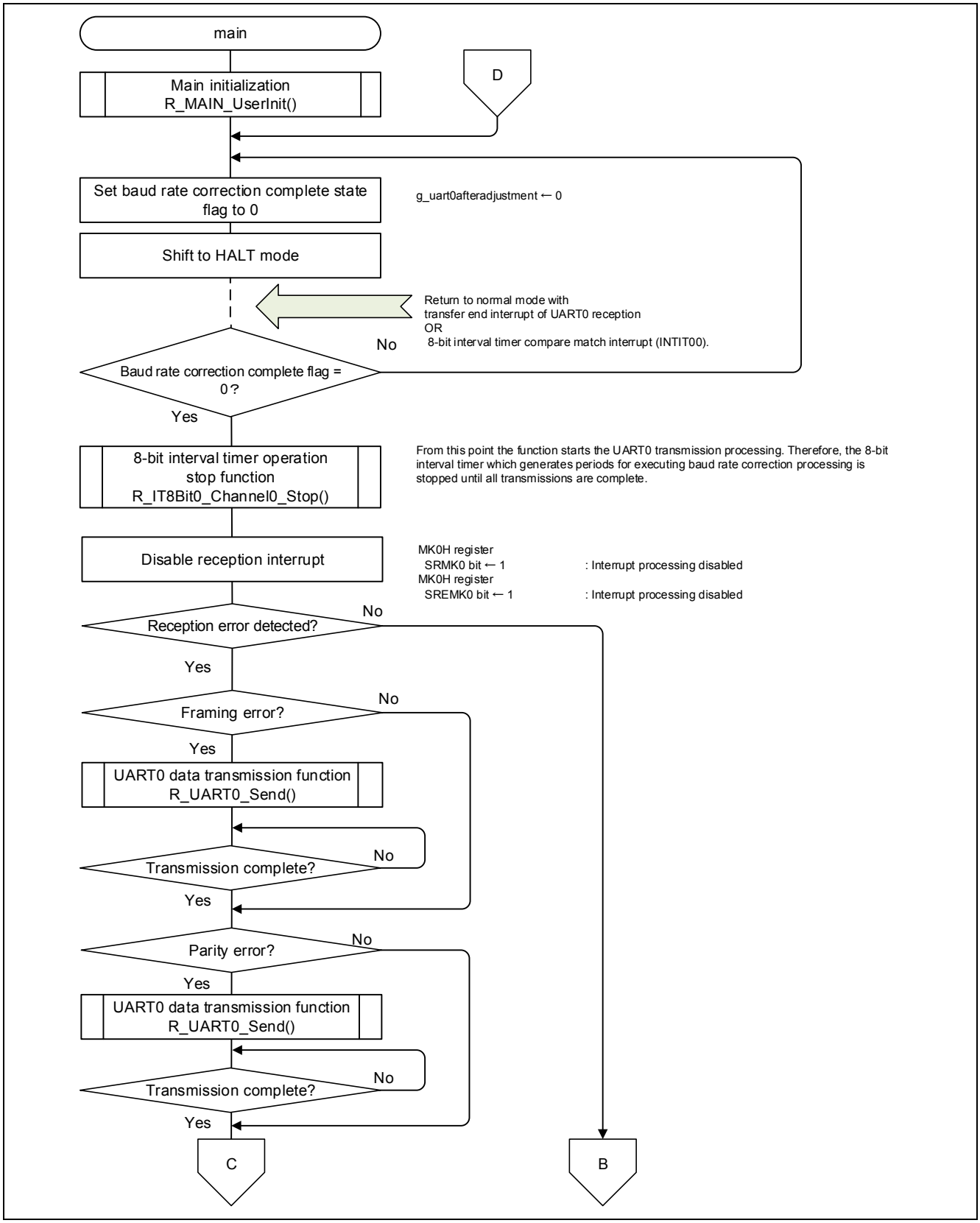

Figure 5.16 Main Processing (1/2)

Figure 5.17 shows the flowchart for main processing (2/2).

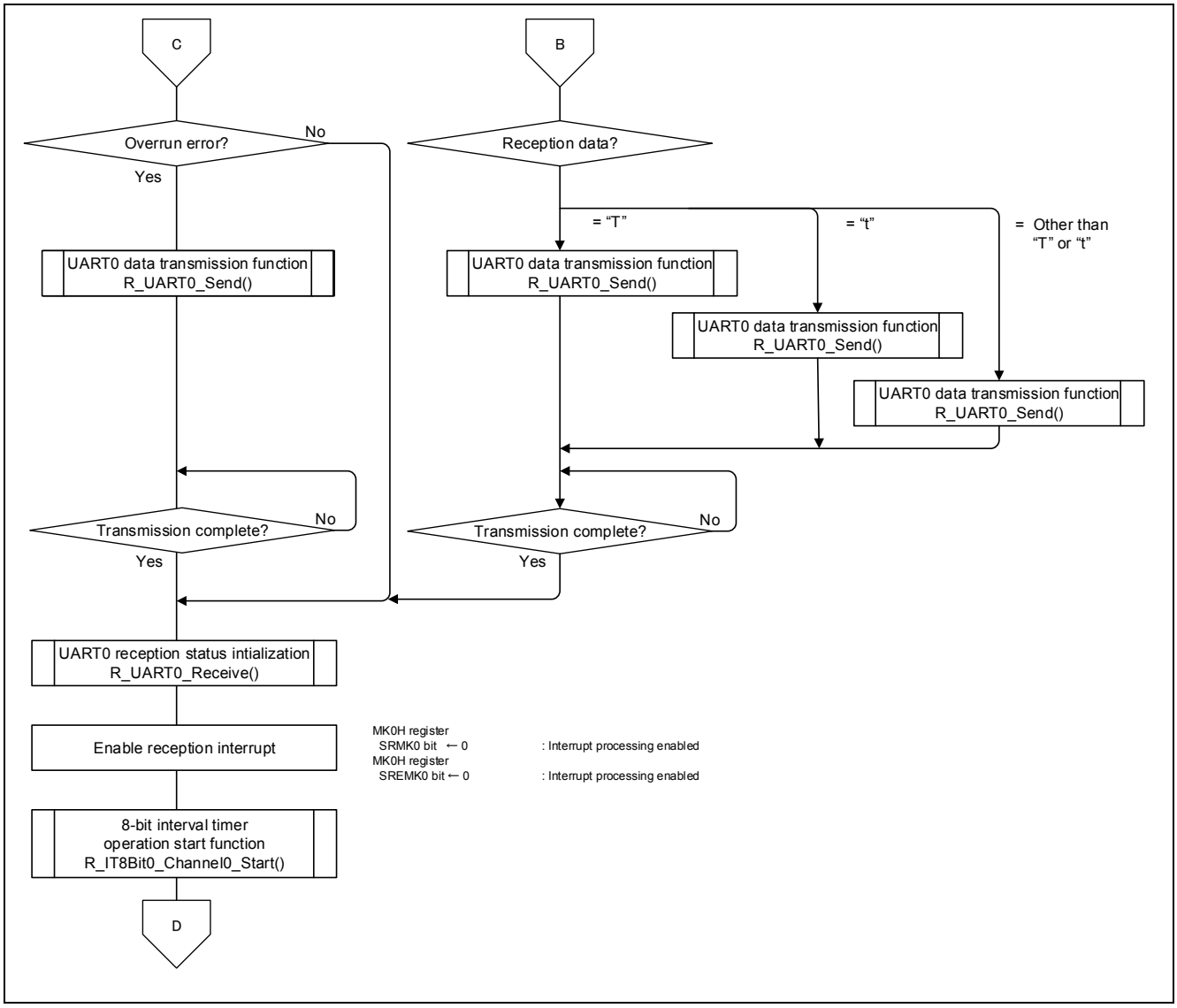

Figure 5.17 Main Processing (2/2)

5.8.10 main initialization

Figure 5.18 shows the flowchart for main initialization.

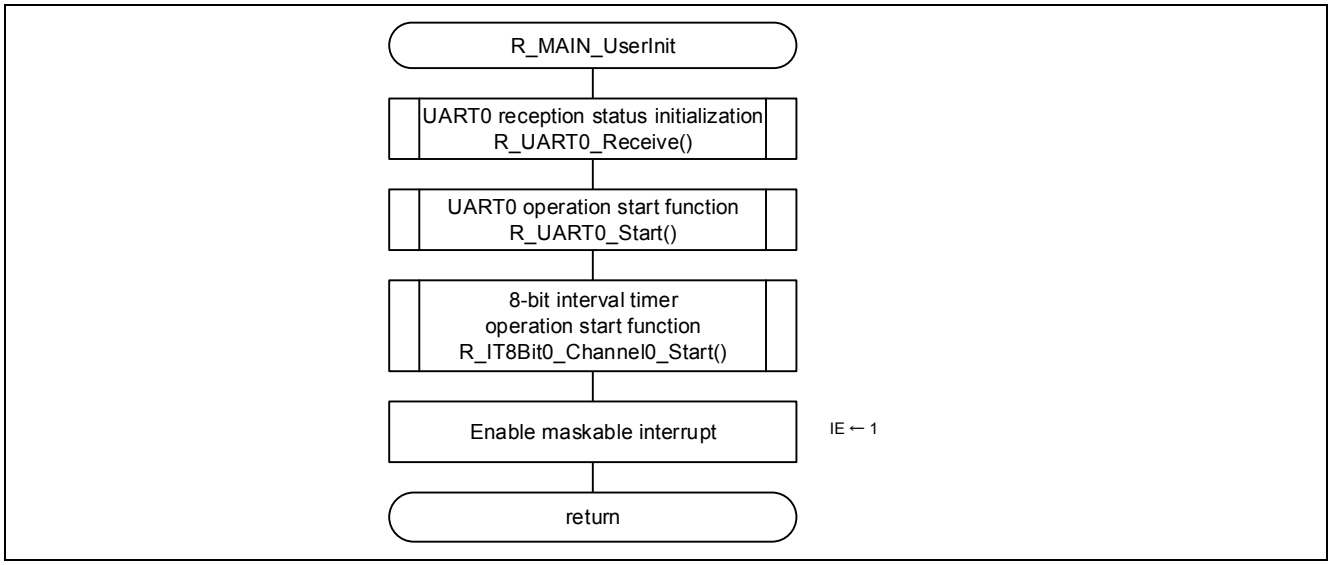

Figure 5.18 main Initialization

5.8.11 UART0 reception status initialization

Figure 5.19 shows the flowchart for UART0 reception status initialization.

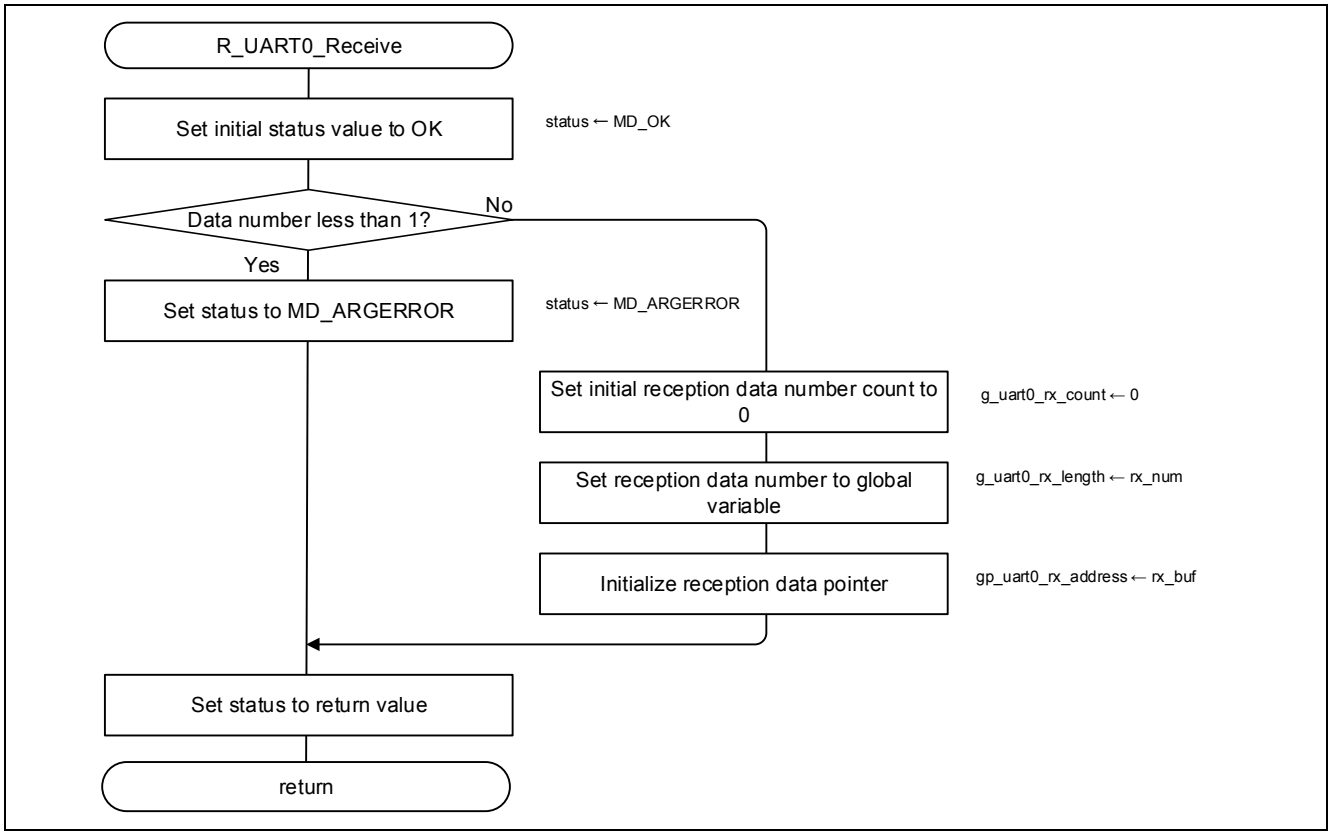

Figure 5.19 UART0 Reception Status Initialization

5.8.12 UART0 operation start function

Figure 5.20 shows the flowchart for the UART0 operation start function.

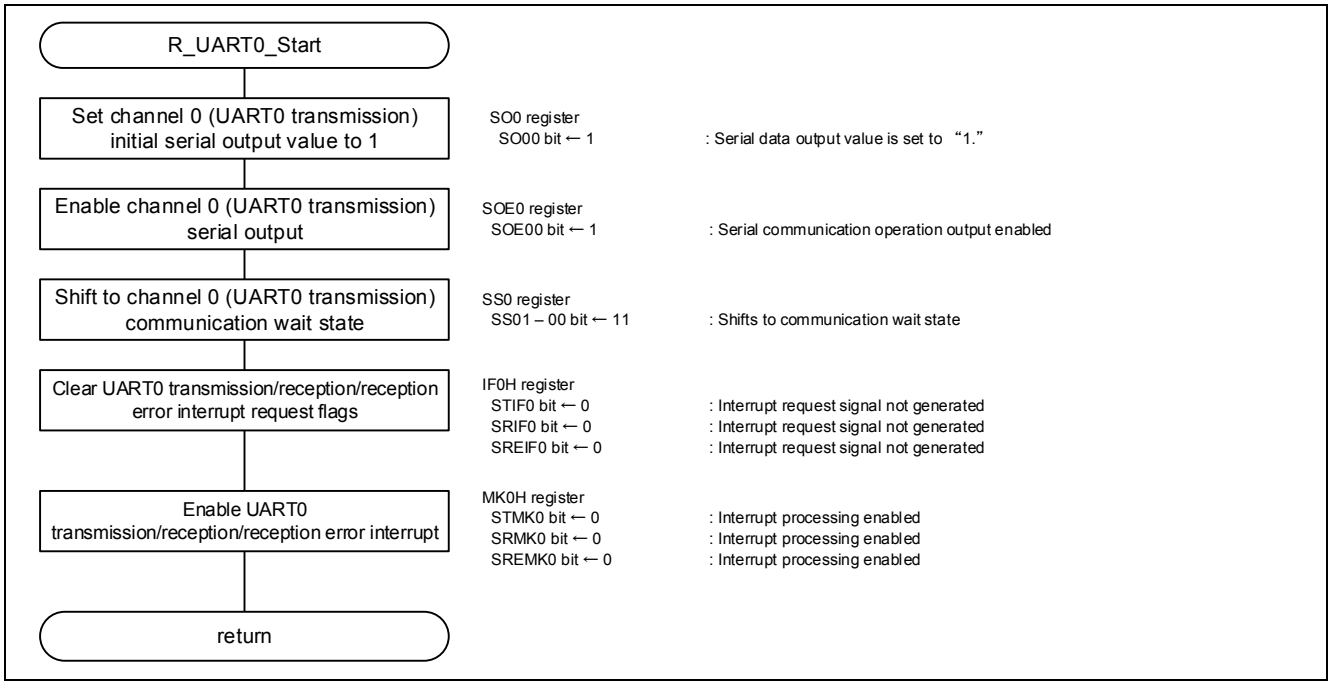

Figure 5.20 UART0 Operation Start Function

UART0 Serial Data Output Value Setting

- Serial output register 0 (SO0)
  Set serial data output value to 1.

Symbol: SO0

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

Bit 0

SO0	Channel n serial data output
0	Set serial data output value to "0"
1	**Set serial data output value set to "1"**

UART0 Serial Output Enable

・Serial output enable register 0 (SOE0)
Enable output in serial communication operations.

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	0	0	SOE01	SOE00
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**

Bit 0

SOE00	Serial output enable/disable for channel n
0	Disables output in serial communication operations
1	**Enables output in serial communication operations**

UART0 Communication Wait State Shift

・Serial channel start register 0 (SS0)
Set to communication wait state.

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	0	0	SS01	SS00
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	**1**

Bit 1

SS01	Trigger to start operations for channel n
0	No trigger operation
1	**SEmn bit is set to 1 and goes to communication wait state.**

Bit 0

SS00	Trigger to start operations for channel n
0	No trigger operation
1	**SEmn bit is set to 1 and goes to communication wait state.**

UART0 Transmission/Reception/Reception Error Interrupt Enable

- Interrupt request flag register (IF0H)
  Clear all UART0 interrupt request flags.
- Interrupt request flag register (MK0H)
  Enable all UART0 interrupts.

Symbol: IF0H

7	6	5	4	3	2	1	0
RTITIF	TMIF00	SREIF0	0	0	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	PIF6
x	x	**0**	x	x	**0**	**0**	x

Bit 5

SREIF0	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

Bit 2

SRIF0	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

Bit 1

STIF0	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

Symbol: MK0H

7	6	5	4	3	2	1	0
RTITMK	TMMK00	SREMK0	1	1	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	PMK6
x	x	**0**	x	x	**0**	**0**	x

Bit 5

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

Bit 2

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

Bit1

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

5.8.13 8-bit interval timer operation start function

Figure 5.21 shows the flowchart for the 8-bit interval timer operation start function.

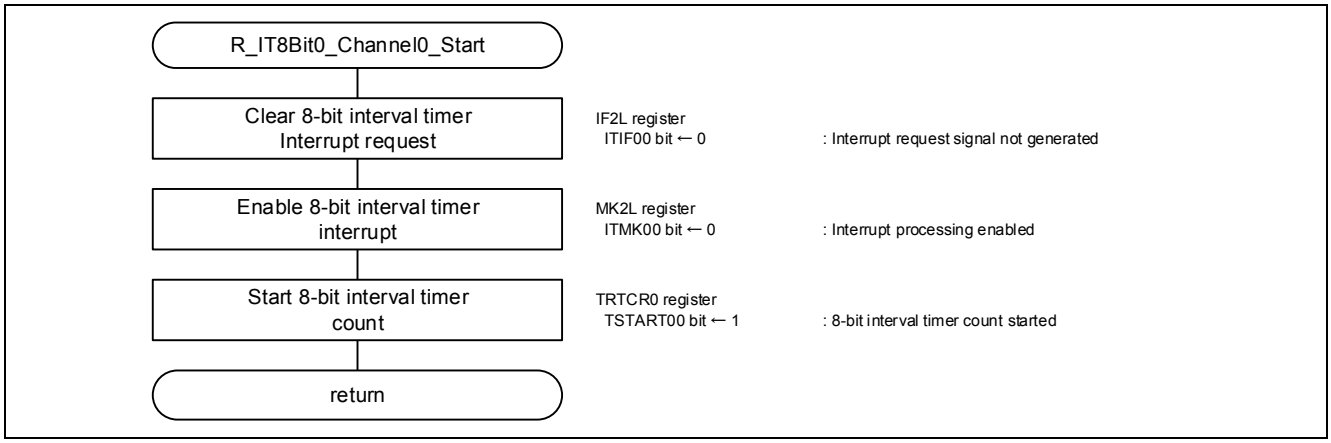

Figure 5.21 8-bit Interval Time Operation Start Function

8-bit Interval Timer Interrupt Enable

- Interrupt request flag register (IF2L)
  Clear 8-bit interval timer interrupt request flag.
- Interrupt request flag register (MK2L)
  Enable 8-bit interval timer interrupt.

Symbol: IF2L

7	6	5	4	3	2	1	0
FLIF	0	0	0	ITIF11	ITIF10	ITIF01	ITIF00
x	x	x	x	x	x	x	**0**

Bit 0

ITIF00	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

Symbol: MK2L

7	6	5	4	3	2	1	0
FLMK	0	0	0	ITMK11	ITMK10	ITMK01	ITMK00
x	x	x	x	x	x	x	**0**

Bit 0

ITMK00	Interrupt processing control
0	**Enables interrupt processing**
1	Disables interrupt processing

8-bit Interval Timer Count Start

- 8-bit interval timer control register 0 (TRTCR0)
  Start 8-bit interval timer count.

Symbol: TRTCR0

7	6	5	4	3	2	1	0
TCSMD0	0	0	TCLKEN0	0	TSTART01	0	TSTART00
x	x	x	x	x	x	x	**1**

Bit 0

TSTART00	8-bit interval timer 0 count start
0	Stops count
1	**Starts count**

5.8.14 8-bit interval timer operation stop function

Figure 5.22 shows the flowchart for the 8-bit interval timer operation stop function.

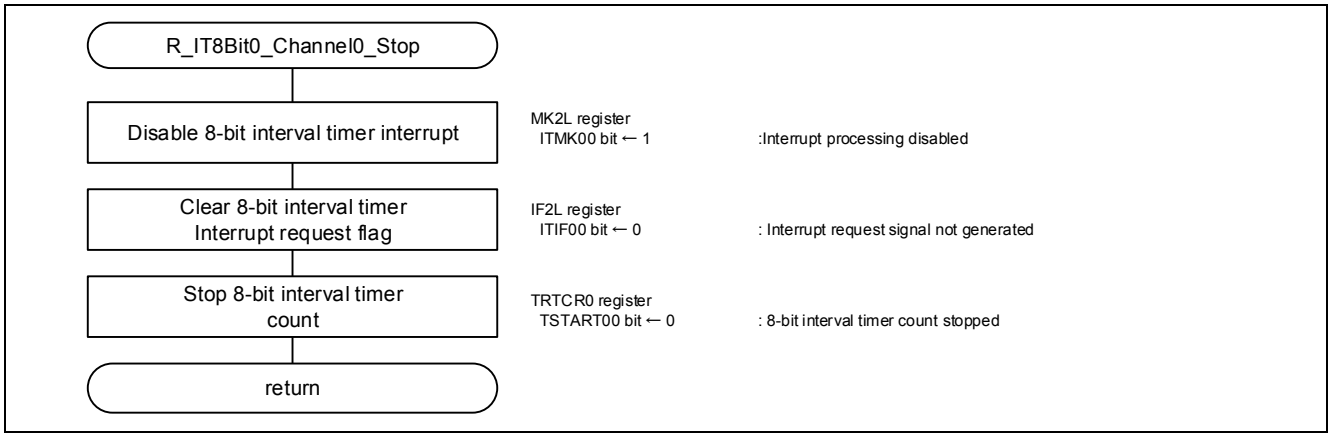

Figure 5.22 8-bit Interval Timer Operation Stop Function

8-bit Interval Timer Interrupt Disable

- Interrupt request flag register (MK2L)
  Disable 8-bit interval timer interrupt.
- Interrupt request flag register (IF2L)
  Clear 8-bit interval timer interrupt request flag.

Symbol: MK2L

7	6	5	4	3	2	1	0
FLMK	0	0	0	ITMK11	ITMK10	ITMK01	ITMK00
x	x	x	x	x	x	x	**1**

Bit 0

ITMK00	Interrupt processing control
0	Enables interrupt processing
1	**Disables interrupt processing**

Symbol: IF2L

7	6	5	4	3	2	1	0
FLIF	0	0	0	ITIF11	ITIF10	ITIF01	ITIF00
x	x	x	x	x	x	x	**0**

Bit 0

ITIF00	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

8-bit Interval Timer Count Stop

- 8-bit interval timer control register 0 (TRTCR0)
  Stop 8-bit interval timer count.

Symbol: TRTCR0

7	6	5	4	3	2	1	0
TCSMD0	0	0	TCLKEN0	0	TSTART01	0	TSTART00
x	x	x	x	x	x	x	**0**

Bit 0

TSTART00	8-bit interval timer 0 count start
0	**Stops count**
1	Enables count

5.8.15 UART0 data transmission function

Figure 5.23 shows the flowchart for the UART0 data transmission function.

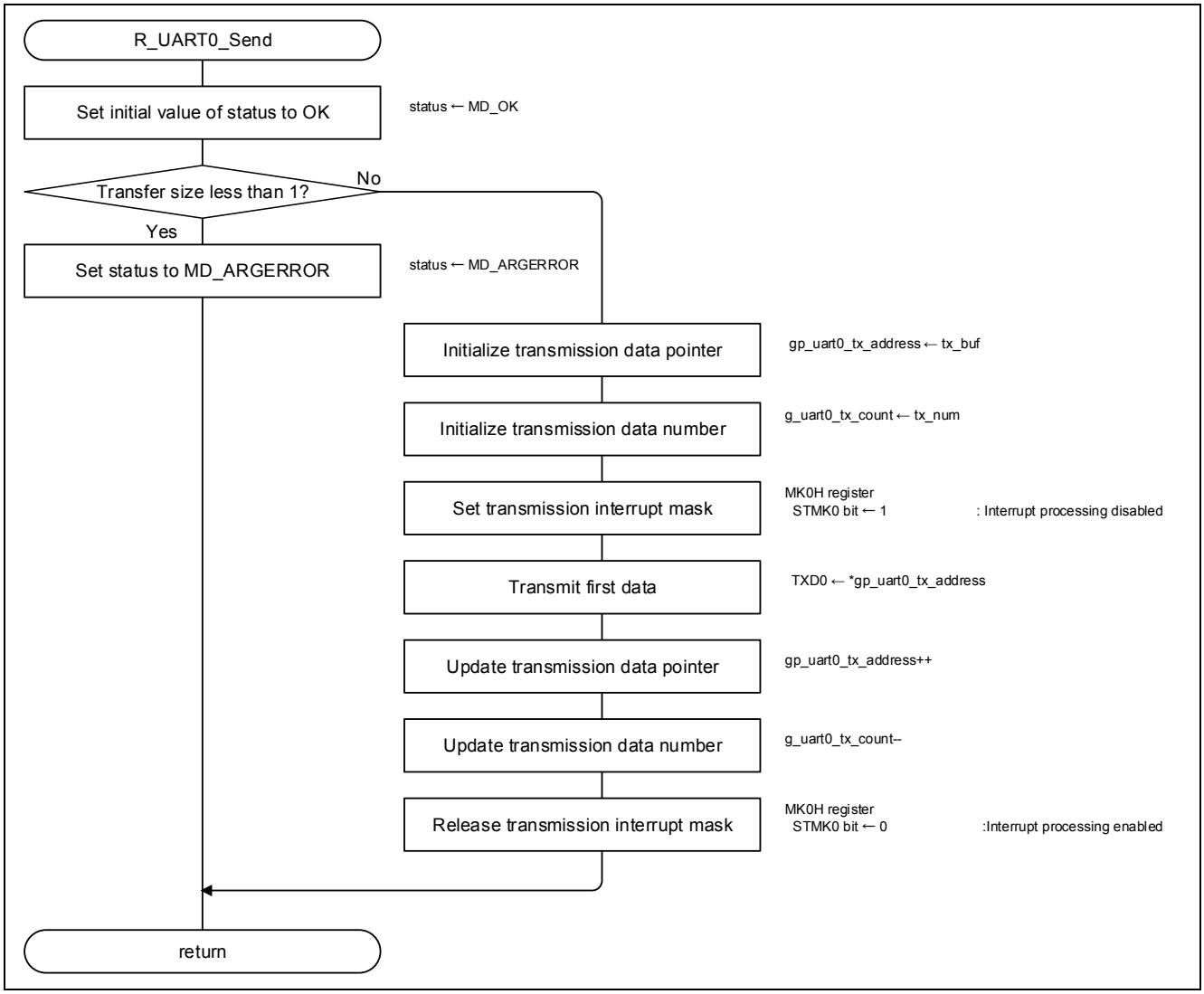

Figure 5.23 UART0 Data Transmission Function

5.8.16 8-bit interval timer interrupt function

Figure 5.24 shows the flowchart for the 8-bit interval timer interrupt function.

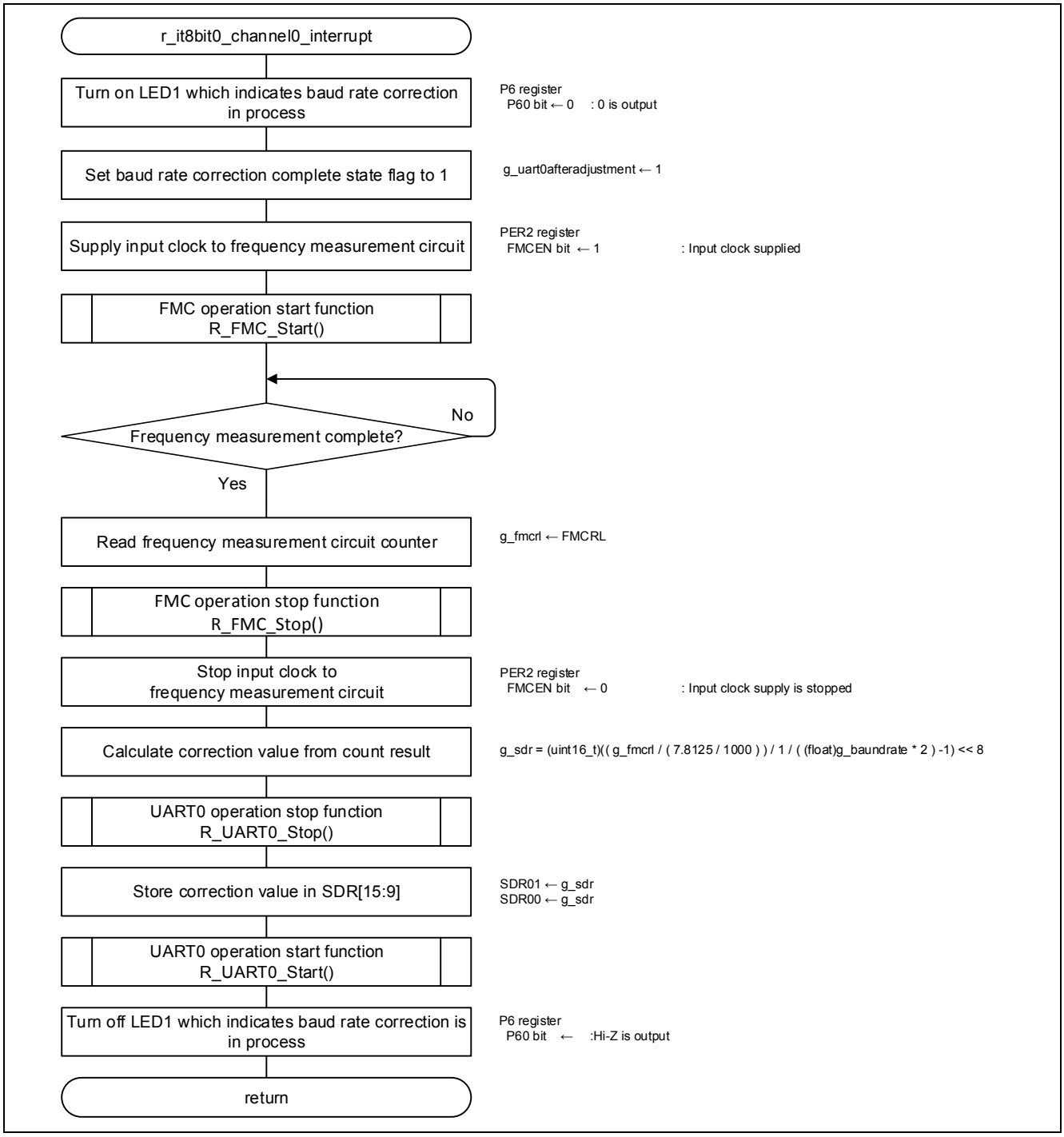

Figure 5.24 8-bit Interval Timer Interrupt Function

Frequency Measurement Circuit Clock Supply Setting

・Peripheral enable register 2 (PER2)

Supply input clock to the frequency measurement circuit during frequency measurement.

Stop input clock supply to the frequency measurement circuit after frequency measurement is completed.

Symbol: PER2

7	6	5	4	3	2	1	0
TMKAEN	FMCEN	DOCEN	0	0	0	0	0
x	**0**	x	x	x	x	x	x

Bit 6

FMCEN	Input clock supply control for frequency measurement circuit
0	**Stops input clock supply**
1	Enables input clock supply

Port Settings

・Port register (P6)

Set P60 to Low during baud rate correction.

Set P60 to Hi-Z after baud rate correction is completed.

Symbol: P6

7	6	5	4	3	2	1	0
0	0	0	0	P63	P62	P61	P60
x	x	x	x	x	x	x	**1**

Bit 4

P60	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	**Output 1**	Input high level

5.8.17 UART0 operation stop function

Figure 5.25 shows the flowchart for the UART0 operation stop function.

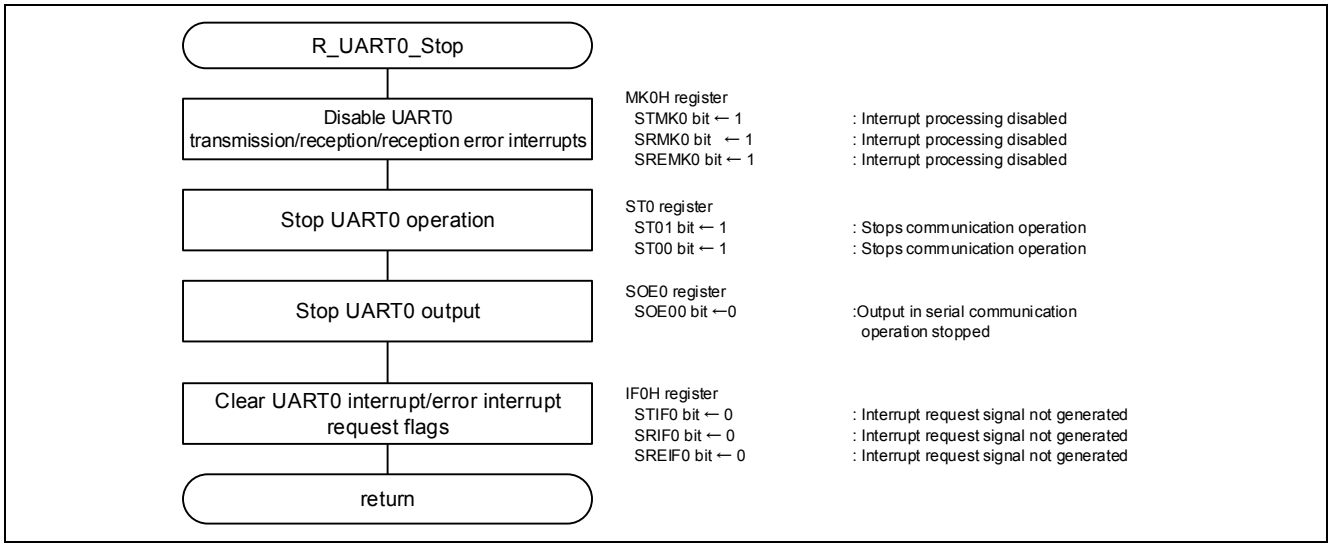

Figure 5.25 UART0 Operation Stop Function

UART0 Interrupt Disable

- Interrupt request flag register (MK0H)
  Disable UART0 interrupt.

Symbol: MK0H

7	6	5	4	3	2	1	0
RTITMK	TMMK00	SREMK0	1	1	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	PMK6
x	x	**1**	x	x	**1**	**1**	x

Bit 5

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

Bit 2

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

Bit 1

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

UART0 Reset Release

・Serial channel stop register 0 (ST0)
Stop communication operation.

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	0	0	ST01	ST00
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	**1**

Bit 1

ST01	Channel n operation start trigger
0	No trigger operation
1	**SEmn bit is cleared to 0, communication operation is stopped.**

Bit 0

ST00	Channel n operation start trigger
0	No trigger operation
1	**SEmn bit is cleared to 0, communication operation is stopped.**

UART0 Output Stop

・Serial output enable register 0(SOE0)
Stop output in 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	0	0	SOE01	SOE00
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**0**

Bit 0

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

UART0 Interrupt Request Flag Clear

- Interrupt request flag register (IF0H)
  Clear all UART0 interrupt request flags.

Symbol: IF0H

7	6	5	4	3	2	1	0
RTITIF	TMIF00	SREIF0	0	0	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	PIF6
x	x	**0**	x	x	**0**	**0**	x

Bit 5

SREIF0	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

Bit 2

SRIF0	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

Bit 1

STIF0	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

5.8.18 Frequency measurement circuit operation start function

Figure 5.26 shows the flowchart for the frequency measurement circuit operation start function.

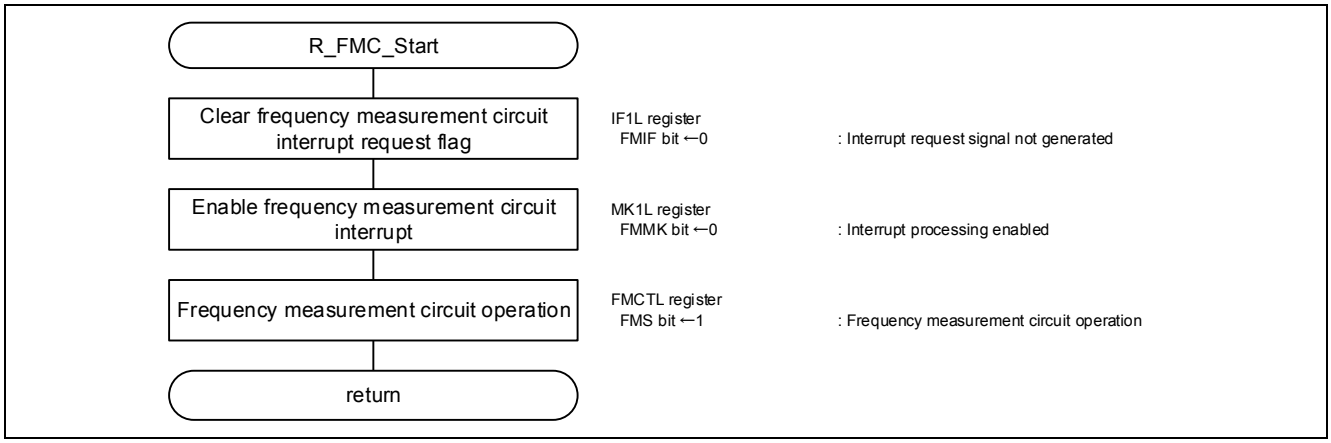

Figure 5.26 Frequency Measurement Circuit Operation Start Function

Frequncy Measurement Circuit Interrupt Enable

- Interrupt request flag register (IF1L)
  Clear frequency measurement circuit interrupt request flag.
- Interrupt request flag register (MK1L)
  Enable frequency measurement circuit interrupt.

Symbol: IF1L

7	6	5	4	3	2	1	0
0	0	TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	FMIF
x	x	x	x	x	x	x	**0**

Bit 0

FMIF	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

Symbol: MK1L

7	6	5	4	3	2	1	0
0	0	TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	FMMK
x	x	x	x	x	x	x	**0**

Bit 0

FMMK	Interrupt processing control
0	**Enables interrupt processing**
1	Disables interrupt processing

Frequency Measurement Circuit Operation

- Frequency measurement control register (FMCTL)
  Set to frequency measurement circuit operation start.

Symbol: FMCTL

7	6	5	4	3	2	1	0
FMS	0	0	0	0	FMDIV2	FMDIV1	FMDIV0
1	x	x	x	x	x	x	x

Bit 7

FMS	Frequency measurement circuit operation enable
0	Stops frequency measurement circuit operation
1	**Enables frequency measurement circuit operation**

5.8.19 Frequency measurement circuit operation stop function

Figure 5.27 shows the flowchart for the frequency measurement circuit operation stop function.

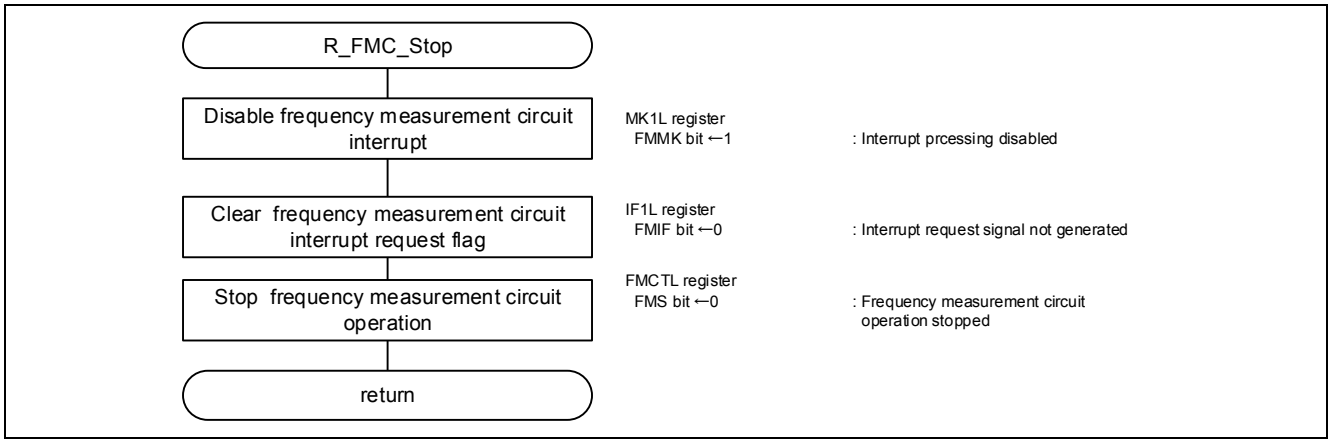

Figure 5.27 Frequency Measurement Circuit Operation Stop Function

Frequency Measurement Circuit Interrupt Disable

- Interrupt request flag register (MK1L)
  Disable frequency measurement circuit interrupt.
- Interrupt request flag register (IF1L)
  Clear frequency measurement circuit interrupt request flag.

Symbol: MK1L

7	6	5	4	3	2	1	0
0	0	TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	FMMK
x	x	x	x	x	x	x	**1**

Bit 0

FMMK	Interrupt processing control
0	Enables interrupt processing
1	**Disables interrupt processing**

Symbol: IF1L

7	6	5	4	3	2	1	0
0	0	TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	FMIF
x	x	x	x	x	x	x	**0**

Bit 0

FMIF	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, interrupt request state

Frequency Measurement Circuit Stop

- Frequency measurement control register (FMCTL)
  Set to frequency measurement circuit operation stop.

Symbol: FMCTL

7	6	5	4	3	2	1	0
FMS	0	0	0	0	FMDIV2	FMDIV1	FMDIV0
0	x	x	x	x	x	x	x

Bit 7

FMS	Frequency measurement circuit operation enable
0	**Stops frequency measurement circuit operation**
1	Enables frequency measurement circuit operation

5.8.20 UART0 reception complete interrupt processing

Figure 5.28 shows the flowchart for UART0 reception complete interrupt processing.

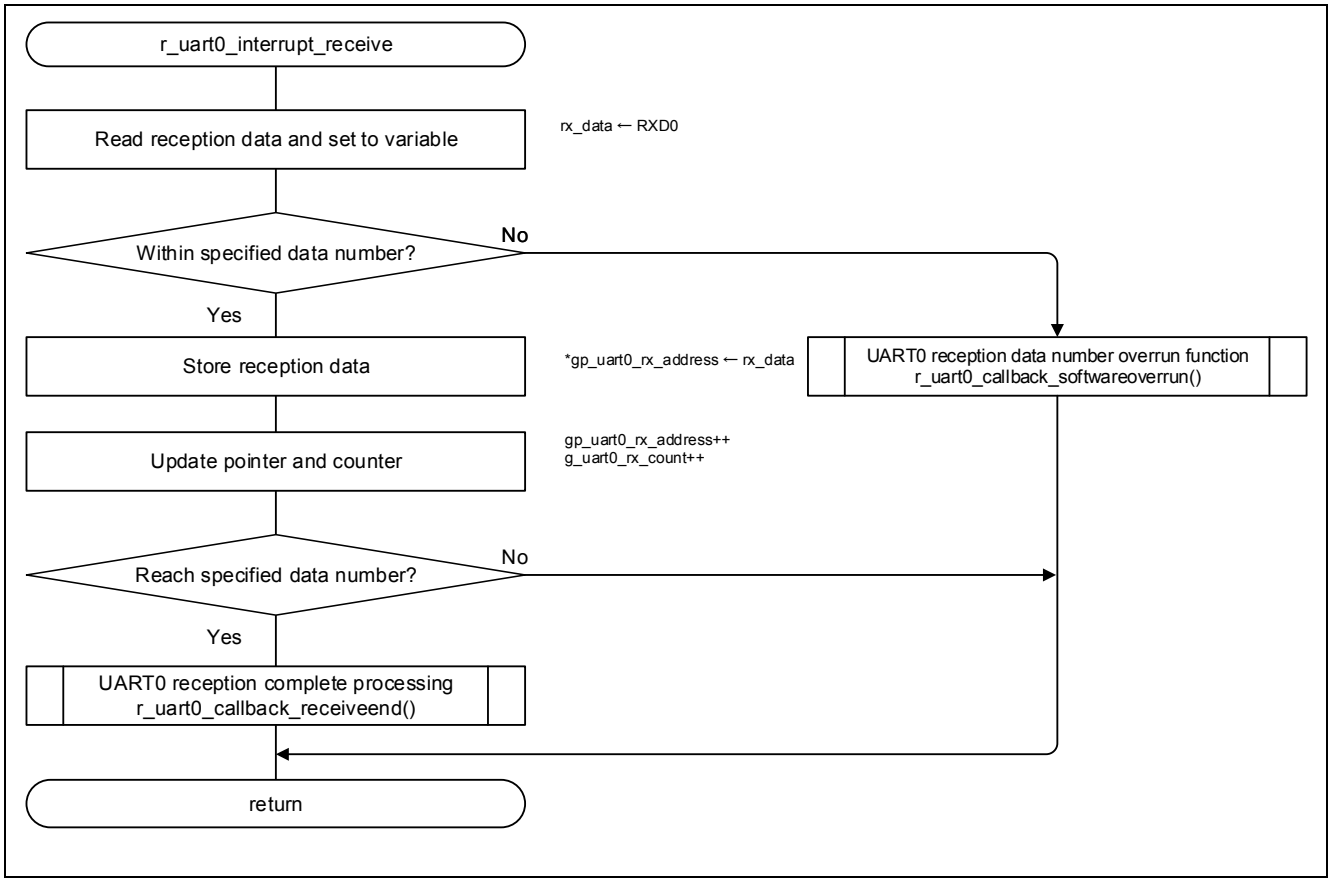

Figure 5.28 UART0 Reception Complete Interrupt Processing

5.8.21 UART0 receive data number overrun processing function

Figure 5.29 shows the flowchart of the UART0 receive data number overrun processing function.

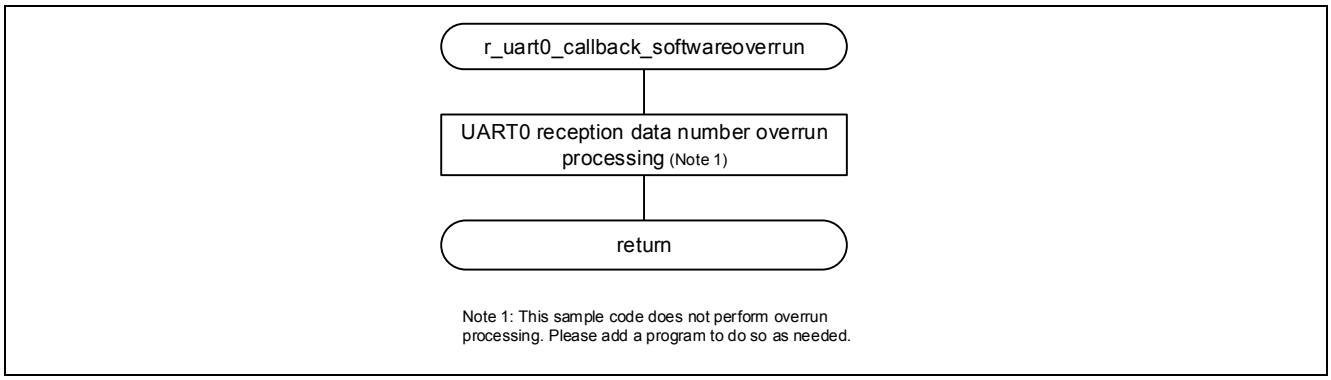

Figure 5.29 UART0 Receive Data Number Overrun Processing Function

5.8.22 UART0 reception complete processing

Figure 5.30 shows the flowchart for UART0 reception complete processing.

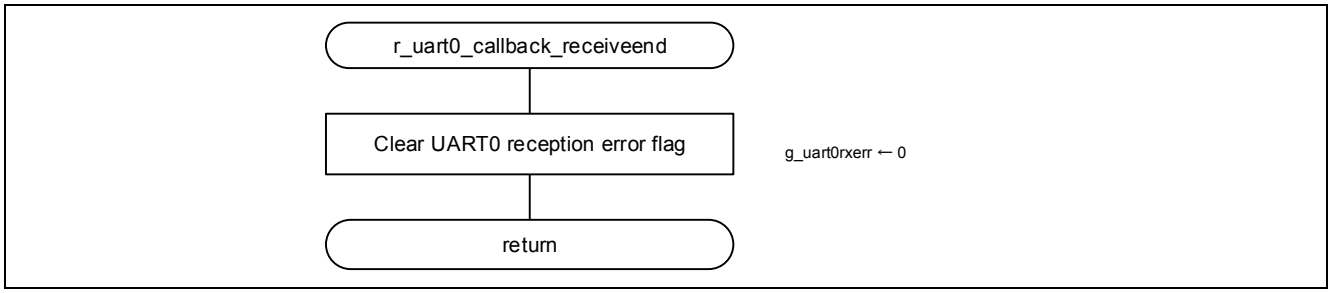

Figure 5.30 UART0 Reception Complete Processing

5.8.23 UART0 error interrupt function

Figure 5.31 shows the flowchart for the UART0 error interrupt function.

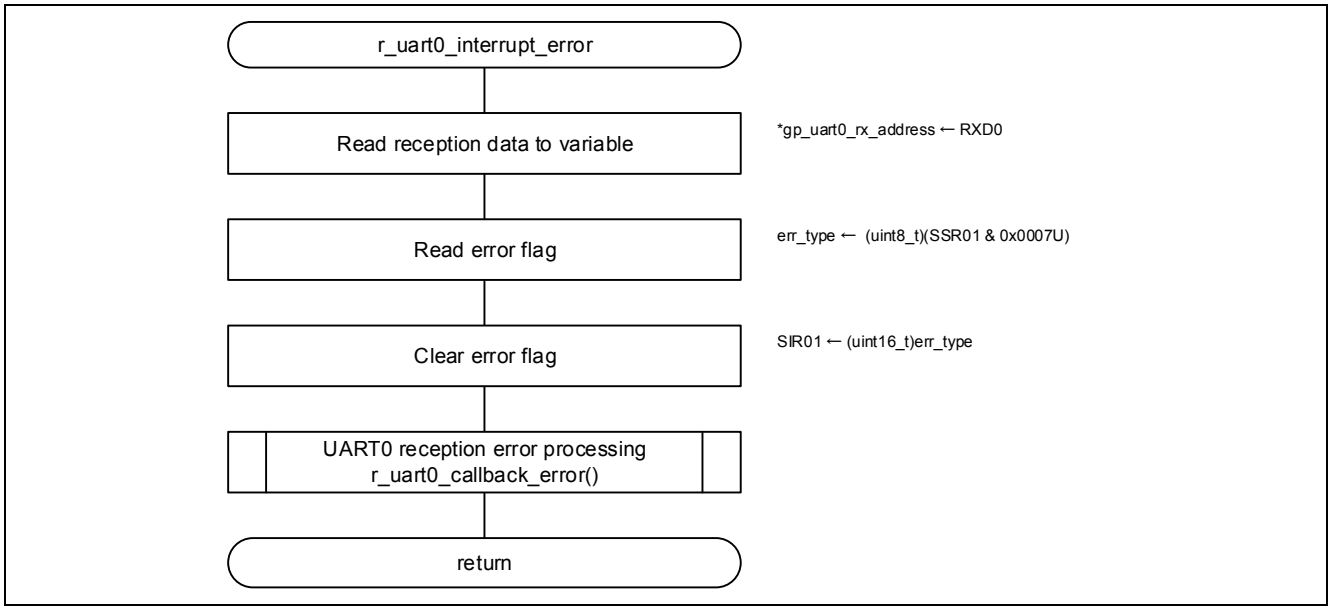

Figure 5.31 UART0 Error Interrupt Function

5.8.24 UART0 reception error processing

Figure 5.32 shows the flowchart for UART0 reception error processing.

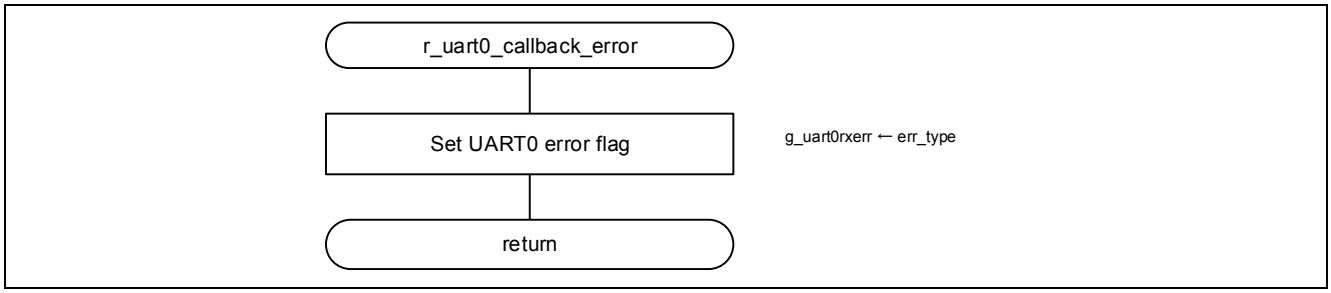

Figure 5.32 UART0 Reception Error Processing

5.8.25 UART0 transmission complete interrupt function

Figure 5.33 shows the flowchart for the UART0 transmission complete interrupt function.

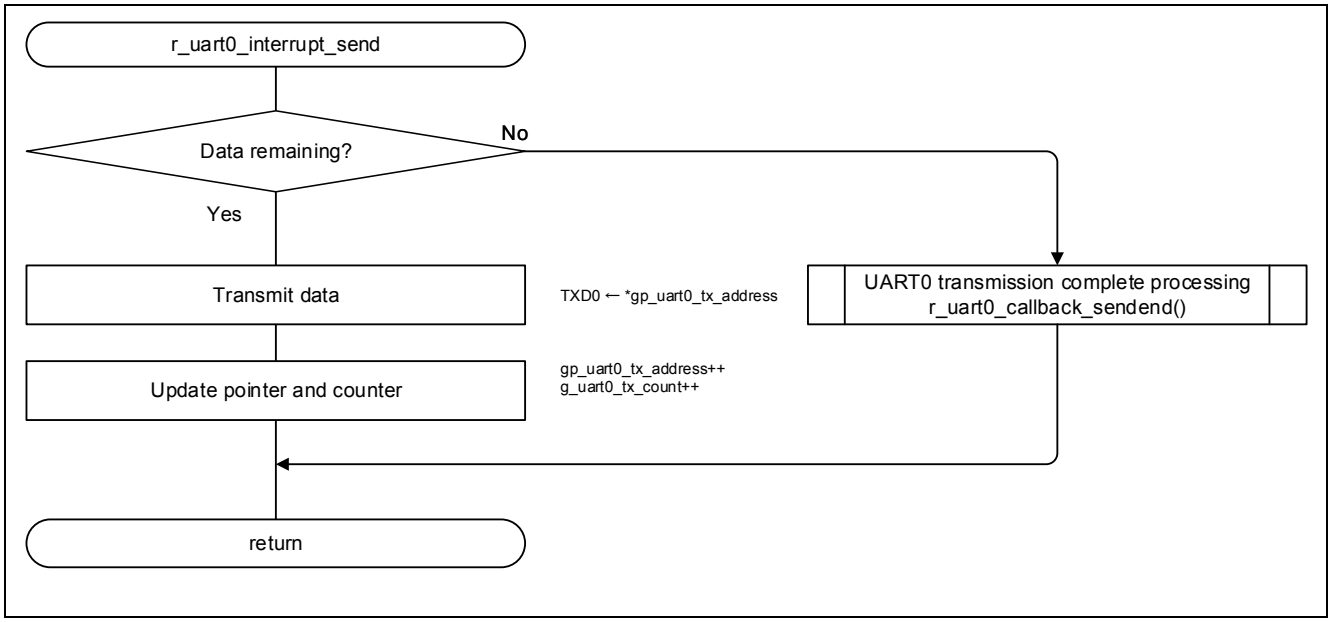

Figure 5.33 UART0 Transmission Complete Interrupt Function

5.8.26 UART0 transmission complete processing

Figure 5.34 shows the flowchart for UART0 transmission complete processing.

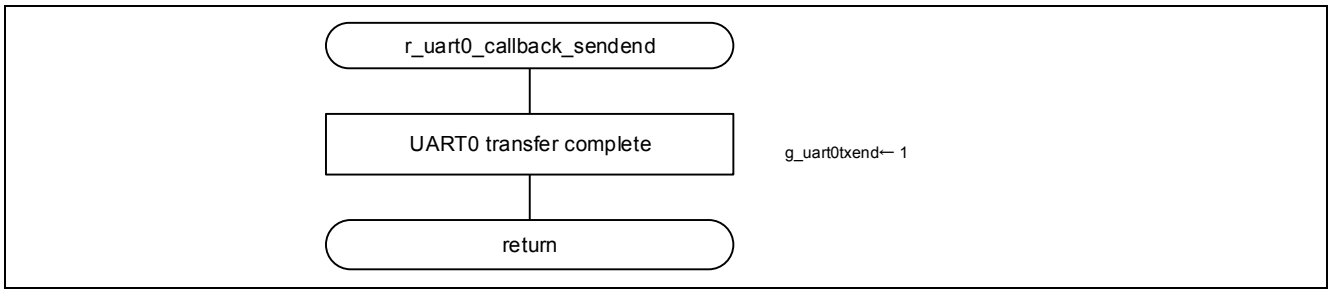

Figure 5.34 UART0 Transmission Complete Processing

6. Sample Code

Download the sample code from the Renesas Electronics homepage.

7. Reference Documents

RL78/I1D User's Manual: Hardware (R01UH0474EJ)

RL78 family User's Manual: Software (R01US0015EJ)

(Please make sure you obtain the latest version from the Renesas Electronics homepage.)

Technical Updates

Obtain the latest information from the Renesas Electronics homepage.

Website and Support

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

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

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	July 1, 2015	-	First edition issued

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

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

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

2. Processing at Power-on

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

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

RENESAS

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

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

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

Renesas Electronics Canada Limited
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, HALII 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 5.0