RL78/G11

UART Reception in STOP Mode Using
Middle-speed On-chip Oscillator IAR

Introduction

This application note explains how to receive data in UART communication during STOP mode using the RL78/G11 middle-speed on-chip oscillator. STOP mode is released when the INTP (external interrupt) pin detects the start bit. Operations return from STOP mode and UART data is received simultaneously.

The middle-speed on-chip oscillator does not provide frequency accurate enough for UART communications and baud rate corrections are necessary. A detailed description on how to correct the baud rate is provided in the following document: "RL78/G11 UART communication with middle speed on chip oscillator IAR (R01AN3621EJ)."

Target Device

RL78/G11

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

Contents

1.   Specification

The RL78/G11 middle-speed on-chip oscillator can be used to receive data in UART communications during STOP mode. When the INTP (external interrupt) pin detects the start bit during STOP mode, STOP mode is released. Operations return from STOP mode and UART data is received simultaneously.

The serial array unit (SAU) is only used for receiving UART communication.

The timer array unit (TAU) is used to correct the baud rate. The external interrupt edge detection 11 (INTP11) interrupt is used to return the CPU from STOP mode

An LED lamp indicates whether the data from the targeted communication device was successfully received or an error occurred.

The middle-speed on-chip oscillator does not provide frequency accurate enough for UART communications and baud rate corrections are necessary. A detailed description on how to correct the baud rate is provided in the following document: "RL78/G11 UART communication with middle speed on chip oscillator IAR (R01AN3621EJ)."

Note: When using the middle-speed on-chip oscillator, UART data cannot be received during STOP mode if the SNOOZE mode is in use. To do, please use the high-speed on-chip oscillator.

Table 1.1 shows peripheral functions and usages. Figure 1.1 shows an outline of the application.

Table 1.1   Peripheral functions and usages

Peripheral functions	Usage
Serial array unit (SAU)	UART receive (UART0)
External interrupt (INTP)	Start bit detection
Timer array Unit (TAU)	Middle-speed on-chip oscillator frequency measurement
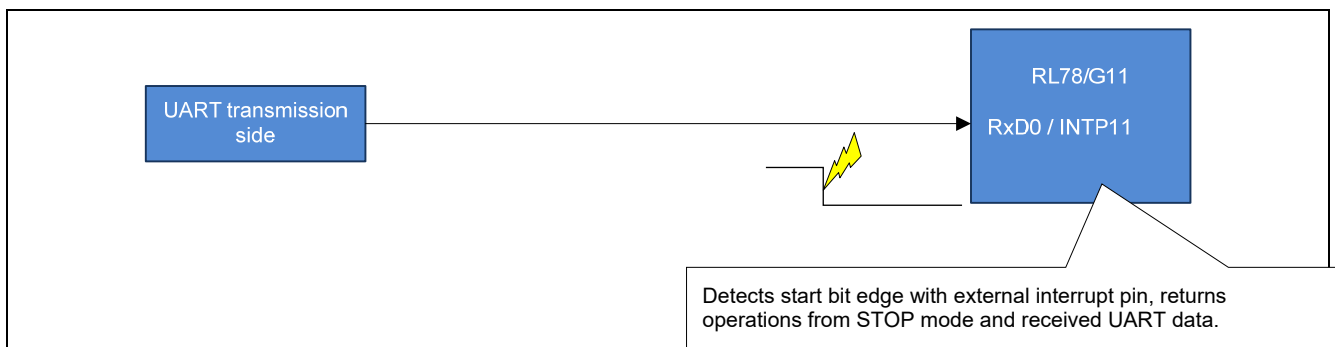
Figure 1.1   Application Outline

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 Confirmation Operations

Item	Description
MCU	RL78/G11 (R5F1056A)
Operating frequency	● Middle-speed on-chip oscillator clock (f_{IM}): 4MHz (UART communication) ● High-speed on-chip oscillator clock (f_{IH}): 24MHz (with MOCO frequency correction)
Operating voltage	3.0V (operating range1.6V to 3.6V) LVD operations (V_{LVD}): Reset mode 2.81V (2.76V to 2.87V)
Integrated development environment	IAR Embedded Workbench for Renesas RL78 V2.21.2
C compiler	IAR C/C++ Compiler for Renesas RL78 V2.21.1.1833

3.   Related Application Notes

Applications notes related to this document are shown below.

- RL78/G13 Initialization CC-RL (R01AN2575EJ)
- RL78/G11 Serial Array Unit (UART Communication) IAR (R01AN3646EJ)
- RL78/G11 UART communication with middle speed on chip oscillator IAR (R01AN3621EJ)

4.　Hardware Explanation

4.1　Hardware Structure Example

Figure 4.1shows the hardware used in this application note.
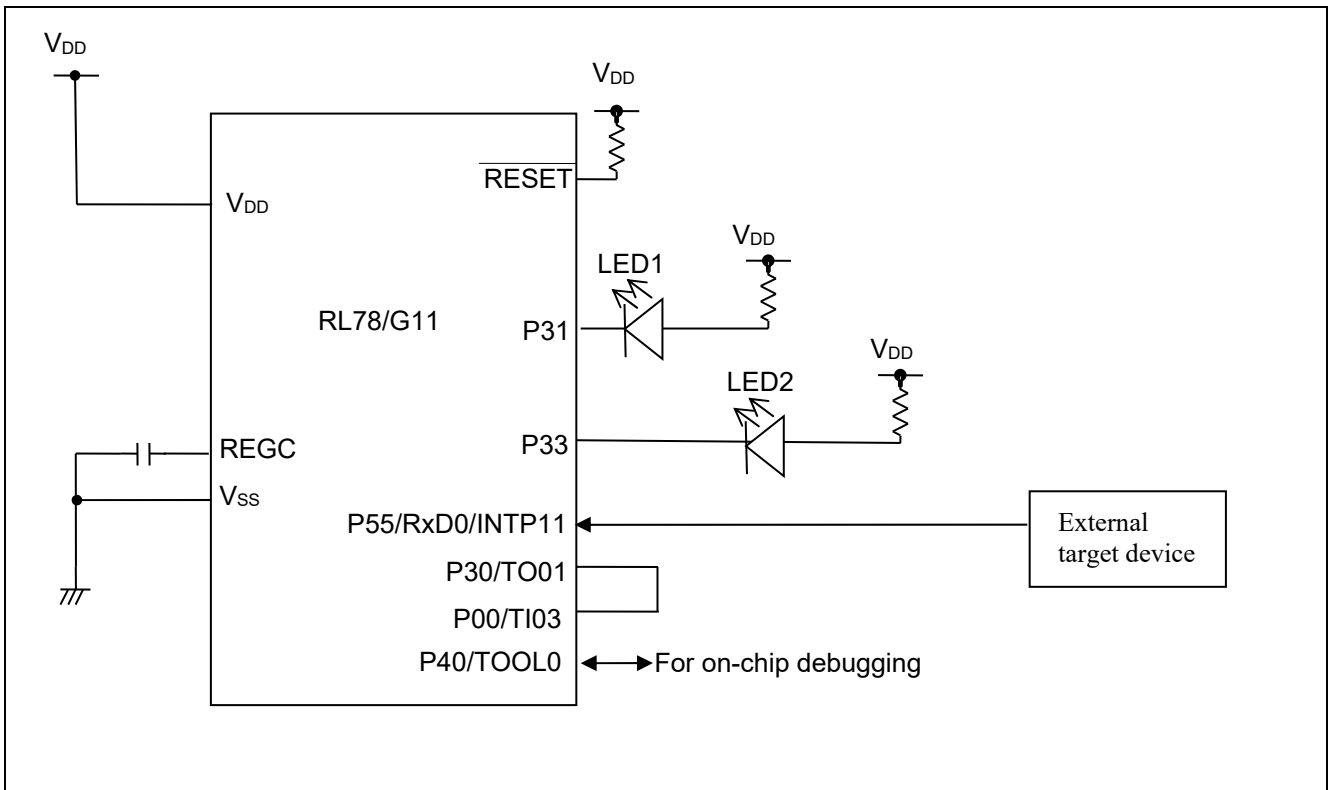
Figure 4.1　Hardware Configuration

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 EVSS, connect the pin to V_{SS}, if it starts with EVDD, connect it to V_{DD}.
　　3 Make V_{DD} higher than the RESET release voltage (V_{LVD}) set in LVD.

4.2   Pin List

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

Table 4.1   List of Pins and Functions

Pin Name	Input/Output	Function
P31	Output	LED1 control
P33	Output	LED2 control
P30/TO01	Output	Waveform for MOCO accuracy measurement
P00/TI03	Input	Waveform for MOCO accuracy measurement
P55/RxD0/INTP11	Input	RxD0: UART0 receive input pin INTP11: Interrupt edge detection

5. Software Explanation

5.1 Operation Outline

This sample code receives UART data while the device is in the wait state with STOP mode. STOP mode is released when the INTP (external interrupt) pin detects the start bit. Operations return from STOP mode and UART data is received simultaneously. The middle-speed on-chip oscillator does not provide frequency accurate enough for UART communications and baud rate corrections are necessary. A detailed description on how to correct the baud rate is provided in the following document: "RL78/G11 UART communication with middle speed on chip oscillator IAR (R01AN3621EJ)."

LEDs are controlled to indicate whether the UART receive data is normal or abnormal.

Table 5.1 shows the correspondence of receive data and the LED state.

Table 5.1 Correspondence of Receive Data and Corresponding LED State

UART Receive Data	LED

Normal data	During pin input edge detection interrupt:
	    LED1/2: On
	After UART0 reception is complete
	    LED1/2: Off

Abnormal data (Framing error)	LED1: Blinking LED2: Off
Abnormal data (Parity error)	LED1: Off LED2: Blinking
Abnormal data (Overrun error)	LED1: Blinking LED2: Blinking

(1) Port initialization

Set P31 and P33 to High output to turn off LED1 and LED2.

(2) SAU initialization

<Setting conditions>

● SAU0 channel 0: use as UART

● 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) TAU initialization

<Channel 1 setting conditions>

- Operating clock: middle-speed on-chip oscillator (MOCO) 4MHz
- Use 16-bit timers
- Software trigger start
- MOCO valid edge: falling edge
- Interval timer mode/square waveform output
- Timer interrupt is not generated when count starts.
- Positive logic output

<Channel 3 setting condition>

- Operating clock: high-speed on-chip oscillator (HOCO) 24MHz
- Use 16-bit timers
- TI03 pin valid edge: set to start trigger and capture trigger
- TI03 pin valid edge: rising edge
- Input pulse width measurement mode
- Positive logic output

(4) Interrupt initialization

<Setting conditions>

- Pin input edge detection (INTP11): rising edge

(5)   main processing initialization

<Setting conditions>

- UART0 reception status initialization
  - Set variable g_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.
- Baud rate setting
  - Set CKC register MCM1 bit to "0" (high-speed on-chip oscillator clock)
  - Wait until high-speed on-chip oscillator clock setting changes.
  - TAU0 channel 3 starts operating.
    - Set IF1L register TMIF03 bit to "0" (interrupt request signal not generated).
    - Set MK1L register TMMK03 bit t to "0" (interrupt processing enabled).
    - Set TS0 register TS03 bit to "1" (count operation enabled state).
  - Set TIS0 register bits TIS02 to TIS00 to "011B" (middle-speed on-chip oscillator clock (f_{IM}).
  - Set TMR01 register to "1000H.
    - Set bits CKS011 to CKS010 to "00B" (operating clock CKm0 set in timer clock selection register m(TPSm).
    - Set CS01 bit to "0" (valid edge of input signal from TImn pin).
    - SPLIT01 to "0" (operates as 16-bit timer).
    - Set bits STS012 to STS010 to "000B" (only software trigger start is valid; other trigger sources are unselected).
    - Set bits CIS011 to CIS010 to "00B" (falling edge).
    - Set bits MD013 to MD011 to "000B" (interval timer mode).
    - Set MD013 bit to "0" (timer interrupt is not generated when count starts (no change to timer output).
  - Set TDR01 register to "1295H".
  - TAU0 channel 1 starts operating.
    - Set IF1L register TMIF01 bit to "0" (interrupt request signal not generated).
    - Set MK1L register TMMK01 bit to "0" (interrupt processing enabled)
    - Set TOE0 register TOE01 bit to "1" (timer output enabled).
    - Set TS0 register TS01 bit to "1" (count operation enabled state)
  - Shift to HALT mode, wait until TAU0 channel 3 interrupt is generated twice.
  - Set CSC register HIOSTOP bit to "1" (high-speed on-chip oscillator stopped).

- ■ Stop TAU0 channel 1 after HALT mode release
  - ◆ Set TT0 register TT01 bit "1" (operation stop (stop trigger generated)).
  - ◆ Set TOE0 register TOE01 bit to "0" (timer output disabled).
  - ◆ Set MK1L register TMMK01 bit to "1" (interrupt processing disabled).
  - ◆ Set IF1L register TMIF01 bit to "0" (interrupt request signal not generated).
- ■ Stop TAU0 channel 3
  - ◆ Set TT0 register TT02 bit to "1" (operation stop (stop trigger generated)).
  - ◆ Set MK1L register TMMK02 bit to "1" (interrupt processing disabled).
  - ◆ Set IF1L register TMIF02 bit to "0" (interrupt request signal not generated)
- ■ Set SDR01 register to baud rate.
  - ◆ SDR01 = (2 * ((23232 / (g_tau0_ch3_width >> 8)) -1)) << 8)
- ■ Set CKC register MCM1 bit to "1" (middle-speed on-chip oscillator clock).
- ■ Wait until middle-speed on-chip oscillator clock setting changes.
- ● UART0 operation start
  - ■ Set SS0 register SS01 bit to "1" (set SEmn bit to 1, shifts to communication wait state).
  - ■ 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 SRMK0 bit to "0" (interrupt processing enabled).
  - ■ Set MK0H register SREMK0 bit to "0" (interrupt processing enabled).
- ● INTP11 operation start
  - ■ Set IF1H register PIF11 bit to "0" (interrupt request signal not generated).
  - ■ Set MK1H register PMK11 bit to "0" (interrupt processing enabled).

(6)   Shift to STOP mode

(7)   Set the following and return from STOP mode after detecting INTP11 (pin input edge detection).
- ● Set MK1H register PMK11 bit to "1" (interrupt processing disabled).
- ● Set IF1H register PIF11 bit to "0" (interrupt request signal not generated).
- ● Set P31 and P33 to Low output, turn on LED1 and LED2.

(8)   Wait until UART0 reception is complete, and then set the following:
- ● Set variable g_uart0rxerr to "0" (normal UART0 completion).
- ● Set variable g_uart0rxend to "1" (normal UART0 reception completion).
- ● Set P31 and P33 to High output, turn off LED1 and LED2.

(9)   When reception data is abnormal

● Framing error: set P31 to Low output/High output, LED1 blinks.

● Parity error: set P33 to Low output/High output, LED2 blinks.

● Overrun error: set P31 and P33 to Low output/High output, LED1/LED2 both blink.

● Repeat step (9).

(10) UART0 reception status initialization

● Set variable g_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.

(11) Set variable g_uart0rxend to "0" (UART0 not received).

(12) INTP11 Operation start

● Set IF1H register PIF11 bit to "0" (interrupt request signal not generated).

● Set MK1H register PMK11 bit to "0" (interrupt processing enabled).

(13) Repeat steps (6) to (12).
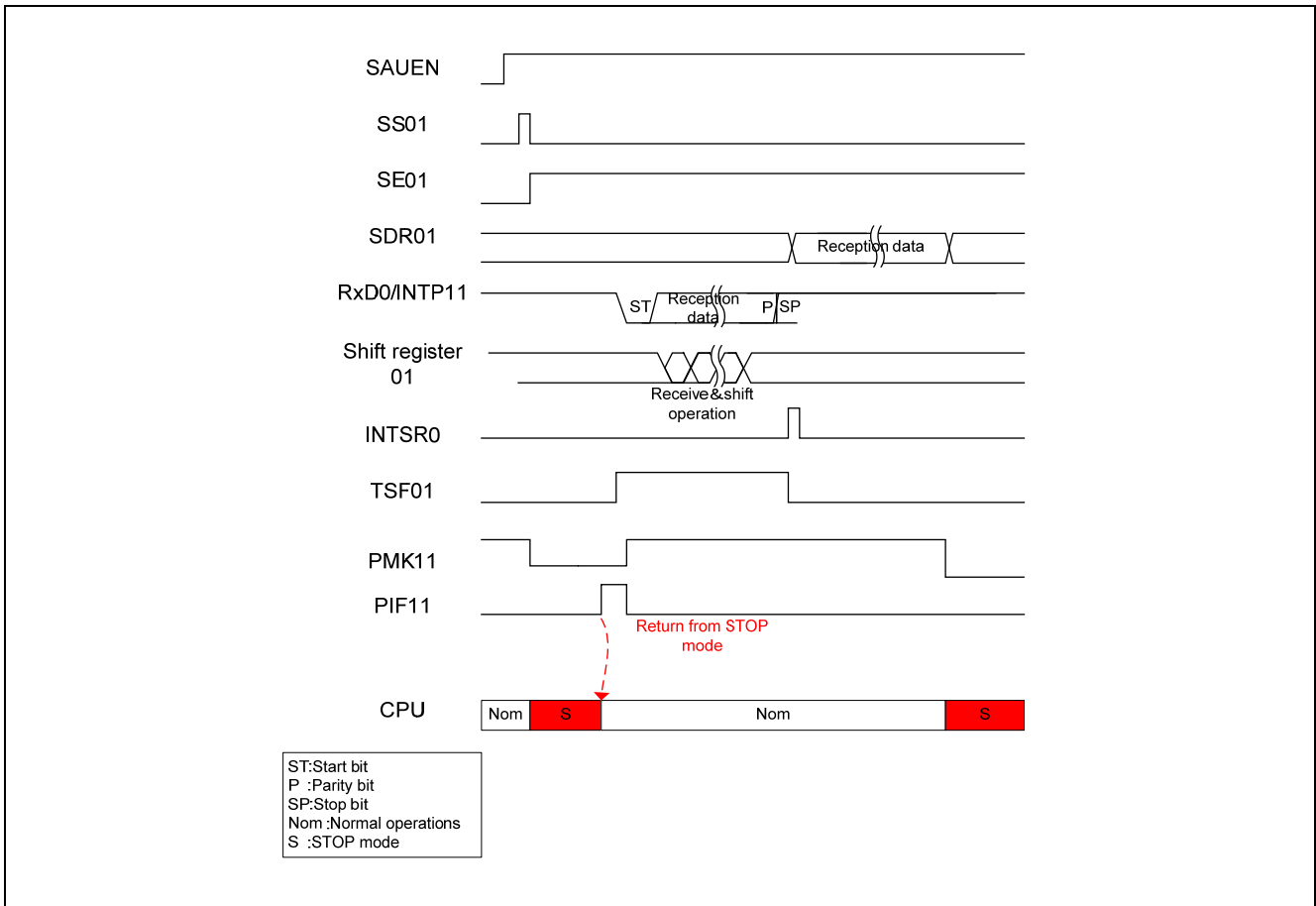
Figure 5.1   Timing Chart

5.2   Correction Processing

The following provides a detailed explanation of correction processing.

(1)   Generating waveforms for measurement

As shown in Figure 5.2, the middle-speed on-chip oscillator (MOCO) frequency is divided by N to generate a waveform for measurement. Increasing the value of N provides a more accurate measurement. However, the final count results from TAU0 channel 3 must be within the 16-bit range. Table 5.2 shows the corresponding measurements (count) based on combinations of Max, Typ, and Min values for MOCO (4MHz ± 12%) and HOCO (24MHz ± 1%).
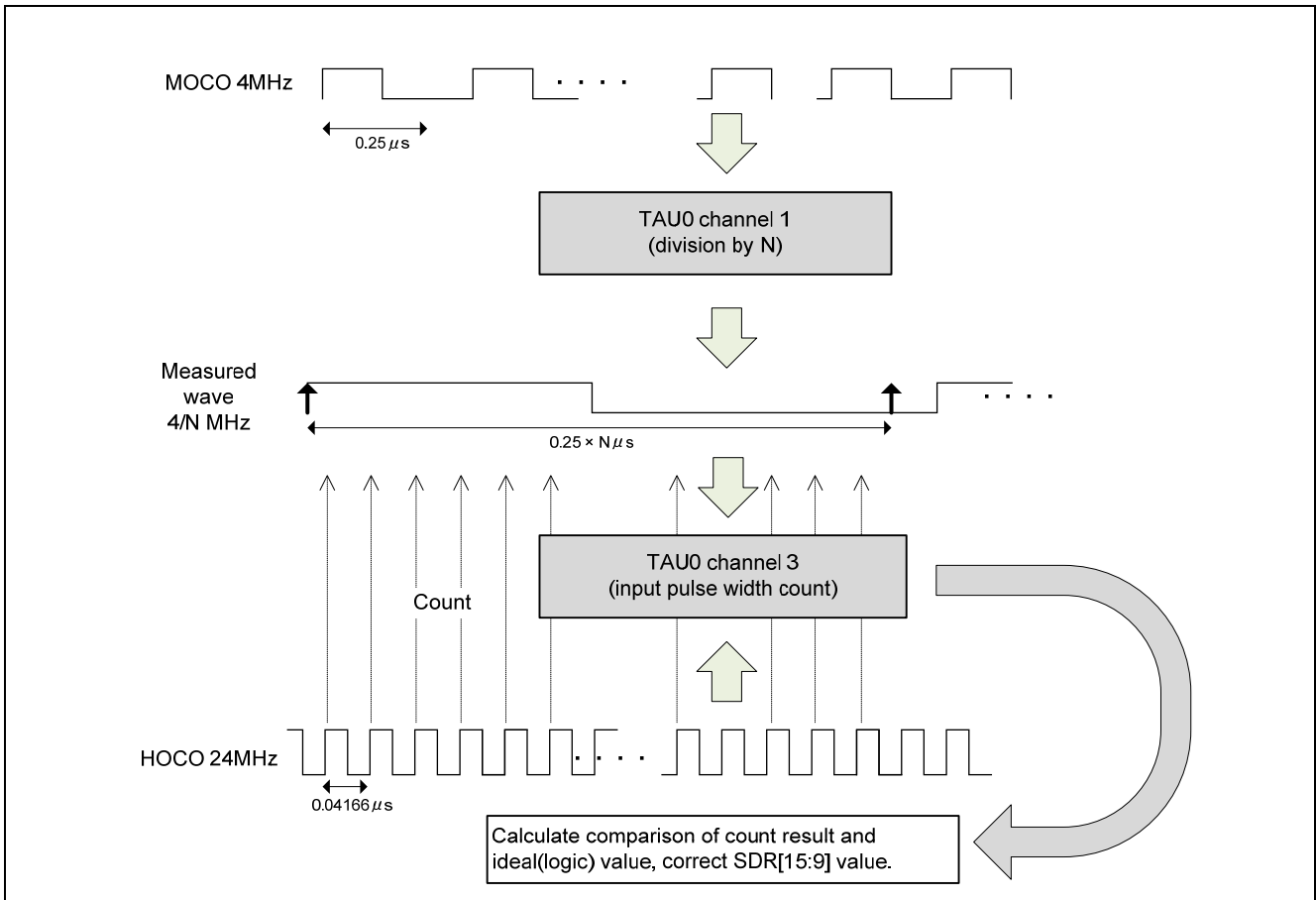
Figure 5.2   Correction Processing

Table 5.2   Measurement Results of MOCO Waveform Divided by N

		MOCO (4 MHz±12%)		
		min: 3.52 MHz	Typ: 4.00 MHz	Max: 4.42 MHz
HOCO (24 MHz±1%)	Max: 24.24 MHz	(24.24 / 3.52) N = 6.886N	(24.24 / 4.00) N = 6.060N	(24.24 / 4.42) N = 5.484N
	Typ: 24.00 MHz	(24.00 / 3.52) N = 6.818N	(24.00 / 4.00) N = 6.000N	(24.00 / 4.42) N = 5.430N
	min: 23.76 MHz	(23.76 / 3.52) N = 6.750N	(23.76 / 4.00) N = 5.940N	(23.76 / 4.42) N = 5.376N

As Table 5.2 shows, the measurement results produce the largest count value when MOCO is 3.52MHz and HOCO is 24.24MHz. Therefore, N can be calculated using the following expressions.

$(24.24 / 3.52)\ N < 65535$

$N < 65535 * 3.52 / 24.24$

$N < 9516.633$

$\therefore$ N (Max) = 9516

The TDR01 register should be set to a half cycle of the waveform to be measured. Therefore, set the value as follows and confirm the r_tau0_baudrate_correction function.

Value to set to TDR01 $= N/2-1$

$= 4757\ (0x1295)$

(2)   Baud rate correction

In this application, UART is MOCO driven, with the frequency set to: f_{CLK} = 4MHz or f_{MCK} = CK00 = f_{CLK} / 2 = 2MHz. The target baud rate is 9600bps; the values for the upper 7 bits of the SDR register (SDR[15:9]) are calculated by the following expressions.

$$SDR01[15:9] + 1 = f_{MCK} \ / \ (2 * 9600)$$
$$= 2 \ MHz / (2 * 9600)$$

In addition, the following expression is used to reflect the accuracy of the MOCO frequency.

$$SDR[15:9] + 1 = 2 \ MHz * (\textit{ideal count value / capture value of CH3}) / (2 * 9600)$$
$$= 2 \ MHz * (6.000N / \textit{capture value of CH3}) / (2 * 9600)$$
$$= 2 \ MHz * (6 * 9516 / \textit{capture value of CH3}) / (2 * 9600)$$
$$= 5947500 / (\textit{capture value of CH3})$$

To simplify the calculation, the right side is divided by 256, and the final expression is as shown below. The sample program states this expression in the r_tau0_baudrate_correction function. Note that the left side value SDR[15:9] is described here as the local variable k.

$$= 23232 / (\textit{capture value of CH3} \ \text{for the upper 8 bits})$$

5.3 UART How to Receive UART Data in STOP Mode

Generally, when MOCO (middle-speed on-chip oscillator) is selected as the system clock, the UART receive operation is disabled if the program is stopped due to SNOOZE mode.

However, the UART receive operation can be enabled when MOCO is selected as the system clock by skipping SNOOZE mode and, instead, returning directly from STOP mode. To do so, you will need to release STOP mode by connecting the external interrupt pin (INTPn) to the RxD pin and detecting the start bit with the external interrupt pin.

Figure 5.3 shows the flow for enabling the UART receive operation in STOP mode.
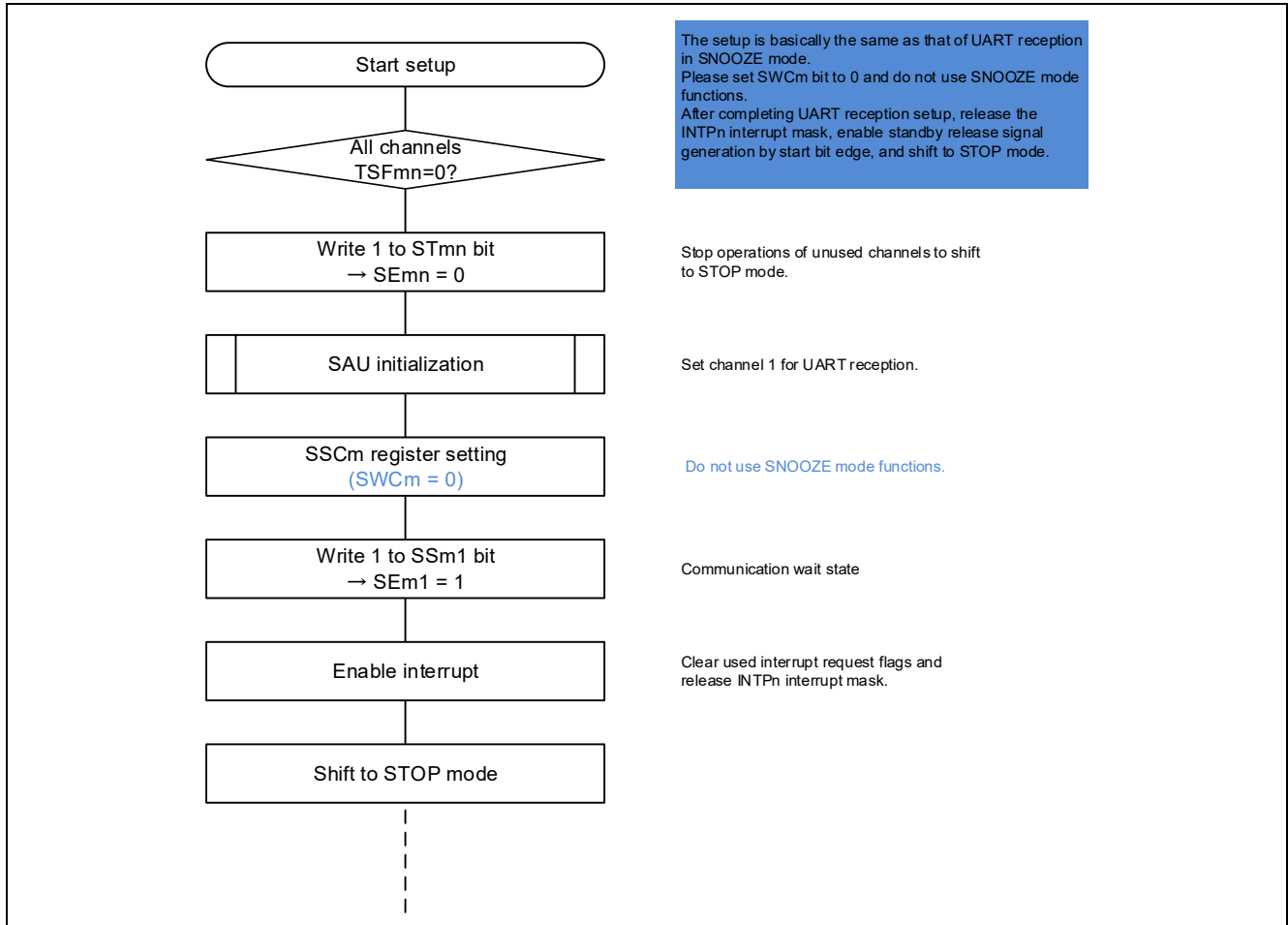
Figure 5.3   How to Receive UART Data in STOP Mode

5.4 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 disabled (stop count after reset release)
000C1H/010C1H	0111 1111B	LVD reset mode, 2.81V(2.76 to 2.87V)
000C2H/010C2H	1110 0000B	HS mode, HOCO: 24MHz
000C3H/010C3H	1000 0100B	Enables the on-chip debugging

5.5   Constants

Table 5.4 lists the constants used in the sample program.

Table 5.4   Constants

Constant Name	Setting	Description
WAIT_MAX	240000	MAX wait value for LED blinking

5.6   Global 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_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
uint16_t	g_uart0_rx_length	Receive data number	R_UART0_Receive r_uart0_interrupt_receive
uint8_t	g_valid_measure	Number of interrupts generated for TAU0 channel 3	r_tau0_baudrate_correction r_tau0_channel3_interrupt
MDSTATUS	g_uart0rxend	UART receive processing complete flag	main r_uart0_callback_receiveend
uint8_t	g_uart0rxerr	UART error receive	main r_uart0_callback_receiveend r_uart0_callback_error
uint32_t	g_tau0_ch3_width	Measurement result of one square wave cycle	r_tau0_baudrate_correction r_tau0_channel3_interrupt
uint32_t	g_wait_count	Wait counter for LED blinking	main r_processing_pause

5.7   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
R_TAU0_Create	TAU0 initialization function
R_SAU0_Create	SAU0 initialization function
R_UART0_Create	UART0 initialization function
R_INTC_Create	INTP 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_tau0_baudrate_correction	TAU0 Baud rate setting function
R_INTC11_Start	INTP11 operation start function
R_TAU0_Channel3_Start	TAU0 channel 3 operation start function
R_TAU0_Channel1_Start	TAU0 channel 1 operation start function
R_TAU0_Channel1_Stop	TAU0 channel 1 operation stop function
R_TAU0_Channel3_Stop	TAU0 channel 3 operation stop function
r_processing_pause	Wait function for LED blinking
r_tau0_channel3_interrupt	TAU0 channel 3 count complete interrupt function
r_intc11_interrupt	INTP11 interrupt processing function
r_uart0_interrupt_receive	UART0 reception complete processing 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

5.8   Function Specifications

The following shows function specifications for the sample code.

Function name: hdwinit

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

Function name: R_TAU0_Create

Outline	TAU0 initialization function	
Header	r_cg_tau.h	
Declaration	void R_TAU0_Create (void)	
Description	Initial setup of TAU0 channels.	
Argument	None	
Return value	None	
Additional notes	None	

Function name: R_SAU0_Create

Outline	SAU0 initialization function
Header	r_cg_sau.h,intrinsics.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: R_INTC_Create

Outline	INTP initialization function
Header	r_cg_intp.h
Declaration	void R_INTC_Create (void)
Description	Initial setup of INTP.
Argument	None
Return value	None
Additional notes	None

Function name: main

Outline	Main function
Header	intrinsics.h
Declaration	void main(void)
Description	After initial main setup, this function waits for a UART reception transmission in the STOP state. UART reception process is executed when the UART start bit is detected by INTP11. The function returns to the STOP state after the UART reception transmission. If a reception error is detected, the error is judged and the error type is indicated by an LED.
Argument	None
Return value	None
Additional notes	None

Function name: R_MAIN_UserInit

Outline	Main function initialization function
Header	intrinsics.h
Declaration	void R_MAIN_UserInit(void)
Description	After UART0 and TAU0 are initialized, this function starts INTP11 operations and 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 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_tau0_baudrate_correction

Outline	TAU0 baud rate initialization function
Header	r_cg_tau.h
Declaration	void r_tau0_baudrate_correction (void)
Description	This function executes the baud rate correction. TAU0 channel 1 and channel 3 are started. The pulse width of the square waveform generated by channel 1 is measured by channel 3. The results are back calculated and the clock division modified to make the UART baud rate as close to its ideal value as possible.
Argument	None
Return value	None
Additional notes	None

Function name: R_INTC11_Start

Outline	INTP11 operation start function
Header	r_cg_intp.h
Declaration	void R_INTC11_Start (void)
Description	Sets INTP11 to operation enabled state.
Argument	None
Return value	None
Additional notes	None

Function name: R_TAU0_Channel3_Start

Outline	TAU0 channel 3 operation start function
Header	r_cg_tau.h
Declaration	void R_TAU0_Channel3_Start (void)
Description	Sets TAU0 channel 3 to operation enabled state.
Argument	None
Return value	None
Additional notes	None

Function name: R_TAU0_Channel1_Start

Outline	TAU0 channel 1 operation start function
Header	r_cg_tau.h
Declaration	void R_TAU0_Channel1_Start (void)
Description	Sets TAU0 channel 1 to operation enabled state.
Argument	None
Return value	None
Additional notes	None

Function name: R_TAU0_Channel1_Stop

Outline	TAU0 channel 2 operation stop function
Header	r_cg_tau.h
Declaration	void R_TAU0_Channel1_Stop (void)
Description	Sets TAU0 channel 1 to operation disabled state.
Argument	None
Return value	None
Additional notes	None

Function name: R_TAU0_Channel3_Stop

Outline	TAU0 channel 3 operation stop function
Header	r_cg_tau.h
Declaration	void R_TAU0_Channel3_Stop (void)
Description	Sets TAU0 channel 3 to operation disabled state.
Argument	None
Return value	None
Additional notes	None

Function name: r_processing_pause

Outline	Wait function for LED blinking
Header	intrinsics.h
Declaration	void r_processing_pause (void)
Description	Executes wait for LED blinking
Argument	None
Return value	None
Additional notes	None

Function name: r_tau0_channel3_interrupt

Outline	TAU0 channel 3 count complete interrupt function
Header	r_cg_tau.h
Declaration	__interrupt void r_tau0_channel3_interrupt(void)
Description	Stores result of pulse width measured by TAU0 channel 3 in global variable.
Argument	None
Return value	None
Additional notes	None

Function name: r_intc11_interrupt

Outline	INTP11 interrupt processing function
Header	r_cg_intp.h
Declaration	__interrupt void r_intc11_interrupt(void)
Description	Disable INTP11 interrupt.
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 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	Acknowledge reception completion and 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 void r_uart0_interrupt_error(void)
Description	Stores reception data in RAM, responds based on error detected in r_uart0_callback_error function.
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	

5.9   Flowcharts

Figure5.1 shows the entire flow of the sample code.
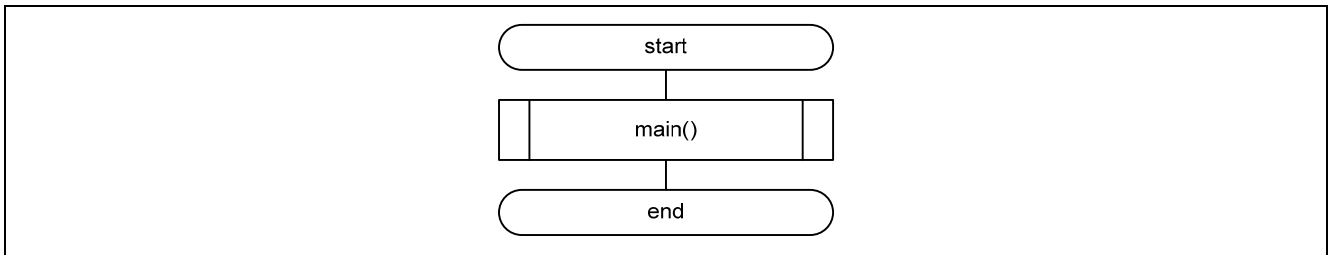
Figure 5.1   Entire Flow

5.9.1   Main processing

Figure5.2 shows the flowchart for main processing (1/2).
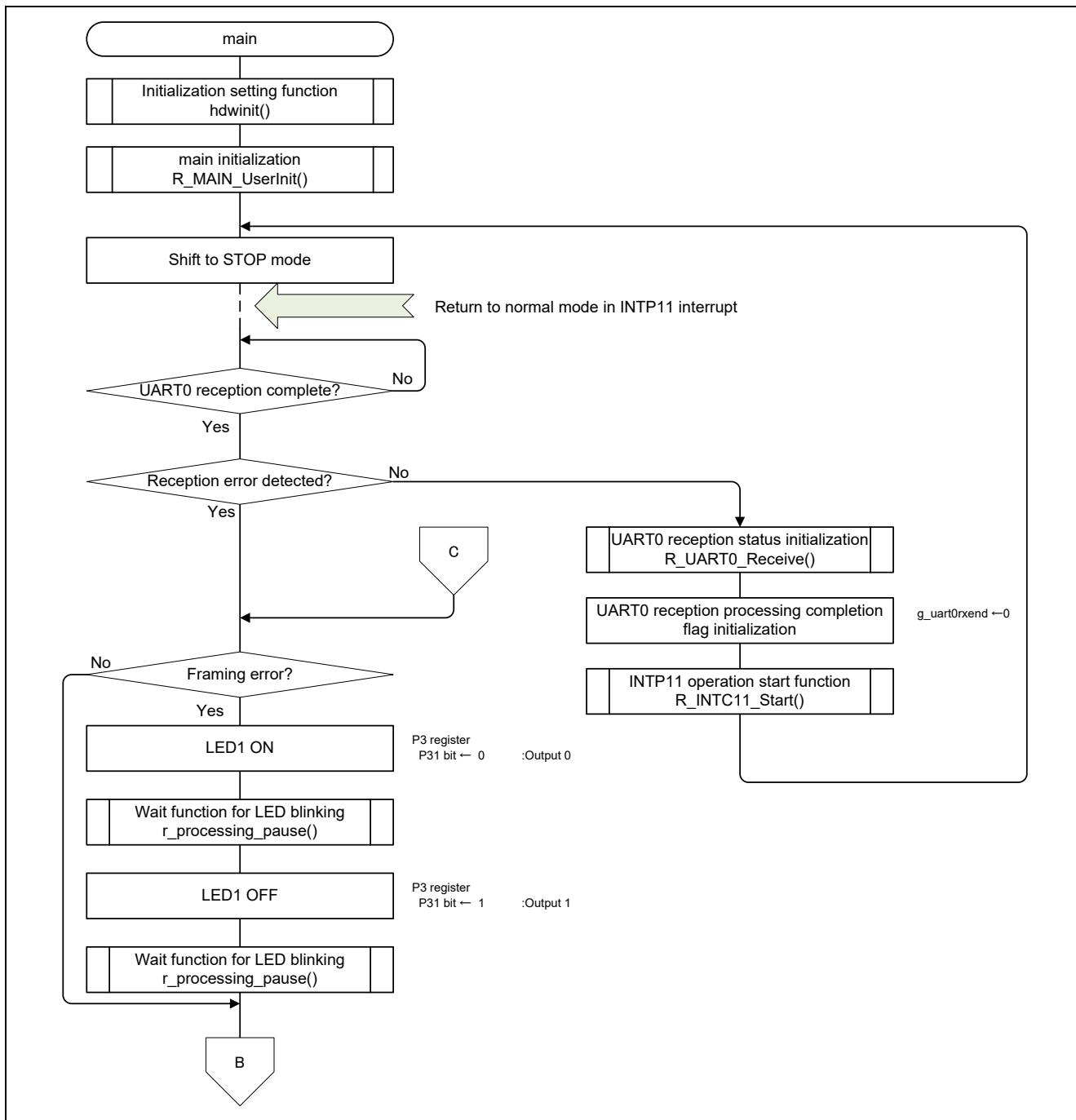
Figure 5.2   Main Processing (1/2)

Figure5.3 shows the flowchart for main processing (2/2).

B

No ← Parity error? Yes

LED2 ON — P3 register
P33 bit ← 0 　:Output 0

Wait function for LED blinking
r_processing_pause()

LED2 OFF — P3 register
P33 bit ← 1 　:Output 1

Wait function for LED blinking
r_processing_pause()

No ← Over run error? Yes

LED1 / LED2 ON — P3 register
P33, p31 bit ← 00B :Output 0

Wait function for LED blinking
r_processing_pause()

LED1 / LED2 OFF — P3 register
P33, p31 bit ← 11B :Output 1

Wait function for LED blinking
r_processing_pause()

C

Figure 5.3　Main Processing (2/2)

5.9.2   Initialization

Figure5.4Figure 5. shows the flowchart of the initialization process.
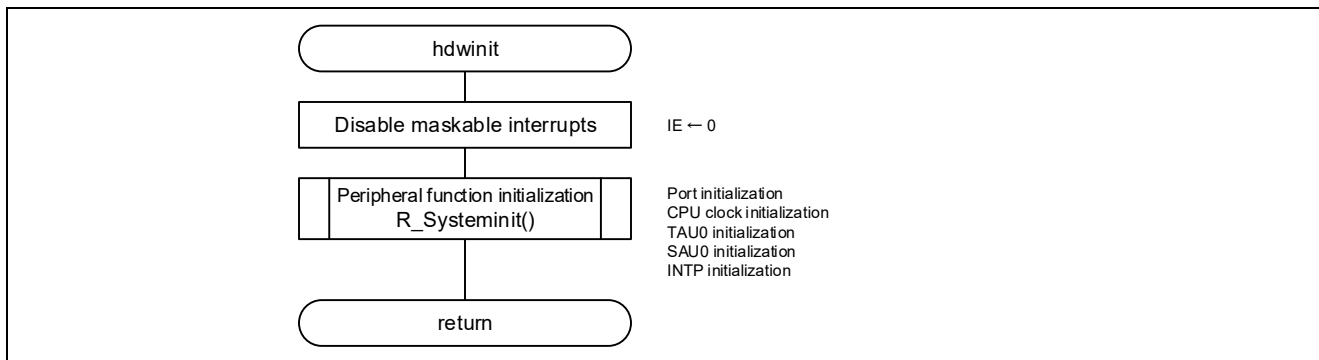
Figure 5.4   Initialization

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

5.9.2   Peripheral function initialization

Figure5.5 shows the flowchart for the peripheral function initialization.
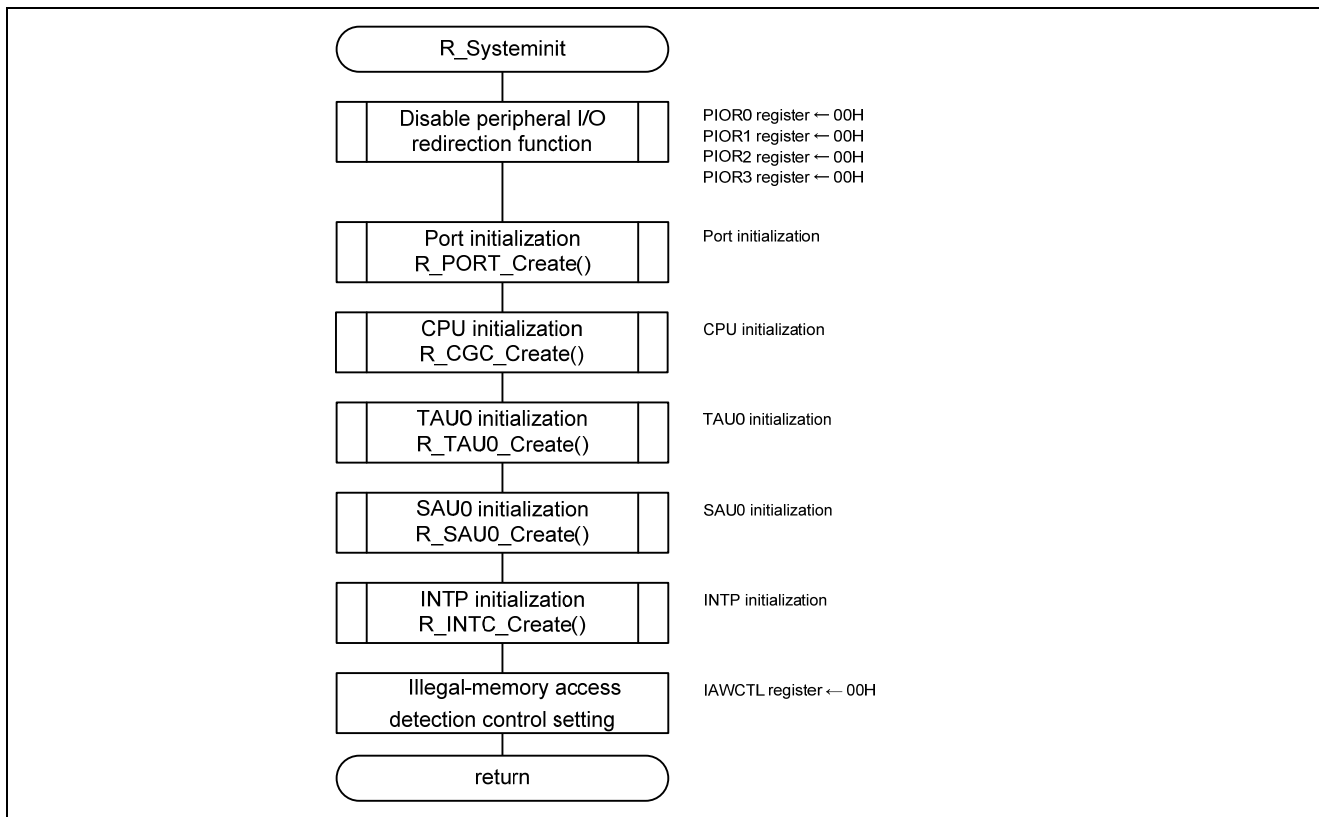
Figure 5.5   Peripheral Function Initialization

5.9.3   Port initialization

Figure5.6 shows the flowchart for port initialization.

PMC3 regisrer F5H
  PMC33, PMC31 bits ← 0B
P3レジスタ 0AH
  P33, P31 bits ← 1B   :Output 1
PM3レジスタ F5H
  PM33, P31 bits ← 00B   :Output mode
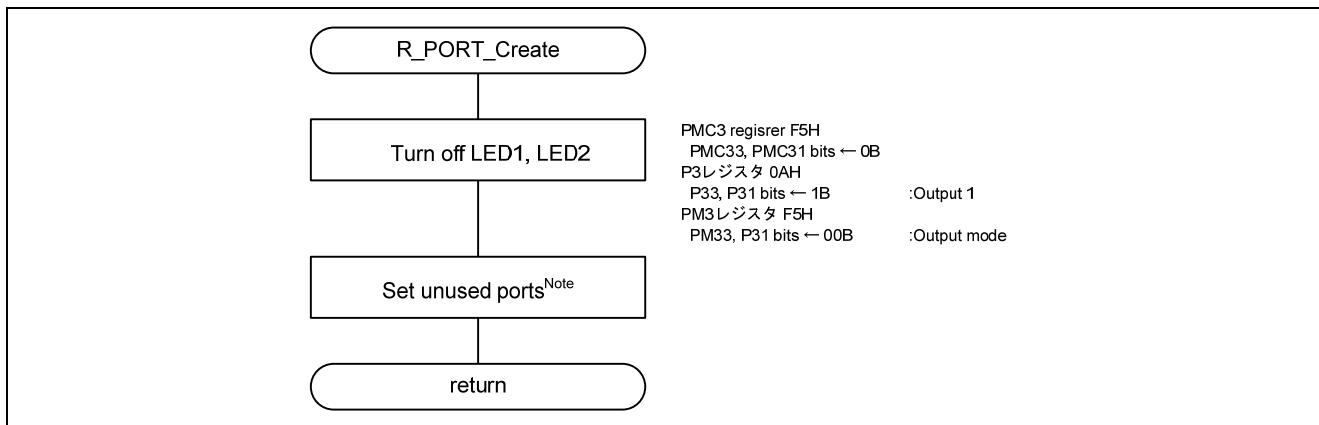
Figure 5.6 Port Initialization

Note   Refer to the initialization flowchart in the RL78/G13 Initialization (R01AN2575EJ) 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.

5.9.4   CPU clock initialization

Figure5.7 shows the flowchart for CPU clock initialization.

MOCODIV register ← 00H   : Set 4MHz clock to
                                                middle speed OCO
MIOEN bit ← 1   : Middle speed on chip oscillator operate

CMC register ← 00H   : High speed system clock
MSTOP bit ← 1 : X1 oscillation stop
SELLOSC bit ← 1 : Select low speed OCO as Sub system clock

OSMC register ← 10H : Provide fIL to 8-bit interval timer
CSS bit ← 0   : Select Main system clock (fMAIN) to
                            CPU/ peripheral hardware clock (fCLK)
MCM0 bit ← 0 : Select main on chip oscillator clock (fOCO) to
                            main system clock (fMAIN)
MCM1 bit ← 1 : Select middle speed main system clock (fMOCO)
                             to main on chip oscillator clock (fOCO)

Figure 5.7 CPU Clock Initialization

5.9.5   TAU0 initialization

Figure5.8 shows the flowchart for TAU0 initialization (1/2).

Figure 5.8   TAU0 Initialization (1/2)

Figure5.9 shows the flowchart for TAU0 initialization (2/2).

A

Channel 1 initialization(generate square waveform generated):
・Operation clock :　　CK00 (24MHz)
・Operate as 16-bit timer
・Software trigger start
・TI01 pin valid edge: falling
・Interval timer mode
・Timer interrupt not generated when count starts
・Channel 1 : master channel output mode
・Channel 1 : positive logic output
・Set channel 1 output to 0
・Channel 1 output enabled

TMR01 register ← 0000H
　CKS011・CKS010 bit ← 00B　:Operation clock CKm0 specified by
　　　　　　　　　　　　　　　　　　　timer clock selection register m(TPSm)
　CCS01 bit ← 0　　　　　　　　　:Operation clock(fMCK) specified by bits CKSmn0 and CKSmn1
　SPLIT01 bit ← 0　　　　　　　　:Operate as 16-bit timer
　STS012・STS010 bit ← 000B :Only software trigger start valid(Other trigger sources unselected)
　CIS011・CIS010 bit ← 00B　:Falling edge
　MD013 - MD011 bit ← 000B　:Interval timer mode
　MD010 bit ← 0　　　　　　　　　:Timer interrupt not generated when count starts
　　　　　　　　　　　　　　　　　　　　(Timer output does not change either)
TDR01 register ← 1295H
TOM0 register
　TOM01 bit ← 0　　　　　　　　　:Master channel output mode
TOL0 register
　TOL01 bit ← 0　　　　　　　　　:Positive logic output(active high)
TO0 register
　TO01 bit ← 0　　　　　　　　　　:Timer output value is 0
TOE0 register
　TOE01 bit ← 1　　　　　　　　　:Enable timer output

Channel 3 initialization(input pulse measurement):
・Operation clock :　　CK00 (24MHz)
・Operate as 16-bit timer
・Set TI03 pin valid edge as start trigger and capture trigger
・TI03 pin valid edge: rising
・Input pulse width measurement mode
・Timer interrupt not generated when count starts
・Channel 3 : master channel output mode
・Channel 3 : positive logic output
・Set channel 3 output to 0
・Channel 3 output enabled

TMR03 register ← 0144H
　CKS031・CKS030 bit ← 00B　:Operation clock CKM0 specified by timer clock
　　　　　　　　　　　　　　　　　　　selection register m(TPSm)
　CCS03 bit ← 0　　　　　　　　　:Operation clock(fMCK) specified by bits CKSmn0 and CKSmn1
　SPLIT03 bit ← 0　　　　　　　　:Operate as 16-bit timer
　STS032・STS030 bit ← 001B :Use valid edge of TImn pin input for both
　　　　　　　　　　　　　　　　　　　start trigger and capture tirigger
　CIS031・CIS030 bit ← 01B　:Rising edge
　MD033 - MD031 bit ← 010B　:Capture mode(input pulse width measurement)
　MD030 bit ← 0　　　　　　　　　:Timer interrupt not generated
TOM0 register
　TOM03 bit ← 0　　　　　　　　　:Master channel output mode
TOL0 register
　TOL03 bit ← 0　　　　　　　　　:Positive logic output(active high)
TO0 register
　TO03 bit ← 0　　　　　　　　　　:Timer output value is 0
TOE0 register
　TOE03 bit ← 1　　　　　　　　　:Enable timer output

TI03 pin noise filter invalid

NFEN1 register
　TNFEN03 bit ← 0 :Noise filter OFF

P00・P30 settings

POM3 register
　POM30 bit ← 0　　　　　　　　　　:Normal output mode
PMC3 register:
　PMC30 bit ← 0　　　　　　　　　　:Digital I/O
P3 register
　P30 bit ← 0　　　　　　　　　　　:Output 0
PM3 register
　PM30 bit ← 0　　　　　　　　　　:Output mode
PMC0 register
　PMC00 bit ← 1　　　　　　　　　　:Digital I/O
PM0 register
　PM00 bit ← 1　　　　　　　　　　:Input mode

return

Figure 5.9　TAU0 Initialization (2/2)

TAU0 Reset Release

・Peripheral reset control register 0 (PRR0)

Release TAU from the reset state.

Symbol: PRR0

7	6	5	4	3	2	1	0
0	IICA1RES	ADCRES	IICA0RES	0	SAU0RES	0	TAU0RES
x	x	x	x	x	x	x	**0**

Bit 0

TAU0RES	Reset control of timer array unit 0
0	**Reset control of timer array unit 0**
1	Reset state of timer array unit 0.

TAU0 Clock Supply

・Peripheral reset control register 0 (PER0)

Clock supply for TAU0.

Symbol: PER0

7	6	5	4	3	2	1	0
RTCWEN	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN
x	x	x	x	x	x	x	**1**

Bit 0

TAU0EN	Input clock supply control for timer array unit 0
0	Stops input clock supply
1	**Enables input clock supply**

TAU0 Operation Clock Selection

・Timer clock selection register 0 (TPS0)
  Set CK00 / CK01 = f_{CLK}

Symbol: TPS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	PRS 031	PRS 030	0	0	PRS 021	PRS 020	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000
x	x	x	x	x	x	x	x	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Bit 7- 0   (n = 0, 1)

PRS 0n3	PRS 0n2	PRS 0n1	PRS 0n0		Operation Clock (CK01/CK00) 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

TAU0 All Channel Operation Stop

・Timer channel stop register 0 (TT0)
  Set all TAU0 channels to operation stop state.

Symbol: TT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	TTH 03	0	TTH 01	0	0	0	0	0	TT0 3	TT0 2	TT0 1	TT0 0
x	x	x	x	**1**	x	**1**	x	x	x	x	x	**1**	**1**	**1**	**1**

Bit 11

TTH 03	Trigger to stop operations of higher 8-bit timer when channel 3 is in 8-bit timer mode
0	No trigger operation
1	**TEHm3 bit is cleared to 0 and the count operation is stopped.**

Bit 9

TTH 01	Trigger to stop operations of higher 8-bit timer when channel 1 is in 8-bit timer mode
0	No trigger operation
1	**TEHm1 bit is cleared to 0 and the count operation is stopped.**

Bit 3 - 0 (n = 0 - 3)

TT 0n	Trigger to stop operations for channel n
0	TEmn bit is cleared to 0 and the count operation is stopped.
1	**Operations are stopped (stop trigger generated)**

TAU0 All Interrupts Disable

・Interrupt request flag register (MK0H/MK1L)
   Disable all TAU0 interrupts.
・Interrupt request flag register (IF0H/IF1L)
   Clear TAU interrupt request flags.

Symbol: MK0H

7	6	5	4	3	2	1	0
STMK1 CSIMK10 IICMK10	TMMK00	SREMK0	1	1	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	PMK6
x	**1**	x	x	x	x	x	x

Bit 6

TMMK00	Interrupt processing control
0	Interrupt processing enabled
1	**Interrupt processing disabled**

Symbol: MK1L

7	6	5	4	3	2	1	0
TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	IICAMK0	SREMK1	SRMK1 CSIMK11 IICMK11
1	**1**	**1**	**1**	**1**	x	x	x

Bit 7 - 3 (n = 1 - 3)(m = 1, 3)

TMMK0n TMMK0mH	Interrupt processing control
0	Enables interrupt processing
1	**Disables interrupt processing**

Symbol: IF0H

7	6	5	4	3	2	1	0
ST1IF CSIIF10 IICIF10	TMIF00	SREIF0	0	0	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	PIF6
x	**0**	x	x	x	x	x	x

Bit 6

TMIF00	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state

Symbol: IF1L

7	6	5	4	3	2	1	0
TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	IICAIF0	SREIF1	SRIF1 CSIIF11 IICIF11
0	**0**	**0**	**0**	**0**	x	x	x

Bit 7 - 3 (n = 1 - 3)(m = 1, 3)

TMIF0n TMIF0mH	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state

TAU0 Interrupt Priority Level Setting

・Priority level flag registers (PR11L, PR01L)
  Set channels 1 and 3 to level 2.

Symbol: PR11L

7	6	5	4	3	2	1	0
TMPR103	TMPR102	TMPR101	TMPR103H	TMPR101H	IICAPR10	SREPR11	SRPR11 CSIPR111 IICPR111
1	x	**1**	x	x	x	x	x

Symbol: PR01L

7	6	5	4	3	2	1	0
TMPR003	TMPR002	TMPR001	TMPR003H	TMPR001H	IICAPR00	SREPR01	SRPR01 CSIPR011 IICPR011
0	x	**0**	x	x	x	x	x

Bit 7, 5 (n = 1, 3)

TMPR10n	TMPR00n	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)

TAU0 Channel 1 Initialization

・Timer mode register 01 (TMR01)
　Set TAU0 channel 1 as follows:
　　・Operation clock: CK00(24MHz)
　　・Operate as 16-bit timer
　　・Software trigger start
　　・TI01 pin valid edge: falling
　　・Interval timer mode

Symbol: TMR01

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS 011	CKS 010	0	CCS 01	SPLIT 01	STS 012	STS 011	STS 010	CIS 011	CIS 010	0	0	MD 013	MD 012	MD 011	MD 010
0	**0**	x	**0**	**0**	**0**	**0**	**0**	**0**	**0**	x	x	**0**	**0**	**0**	**0**

Bit 15 – 14

CKS 011	CKS 010	Selection of operation clock (f_{MCK}) for channel n
0	**0**	**Operation clock CKm0 set in timer clock selection register m (TPSm)**
0	1	Operation clock CKm2 set in timer clock selection register m (TPSm)
1	0	Operation clock CKm1 set in timer clock selection register m (TPSm)
1	1	Operation clock CKm3 set in timer clock selection register m (TPSm)

Bit 12

CCS01	Selection of channel n operation clock (f_{TCLK})
0	**Operation clock (f_{MCK}) set in bits CKSmn0 and CKSmn1**
1	Valid edge of input signal from TImn pin

Bit 11

SPLIT01	Selection of 8-bit/16-bit timer operation for channels 1 and 3
0	**Operates as 16-bit timer**
1	Operates as 8-bit timer

Bit 10 – 8

STS 012	STS 011	STS 010	Setting start or capture trigger of channel n
0	**0**	**0**	**Only software trigger start is valid (other trigger sources are unselected)**
0	0	1	Valid edge of TImn pin input is used as both the start trigger and capture trigger
0	1	0	Both edges of TImn pin input are used as the start trigger and capture trigger
1	0	1	Interrupt signal of master channel is used (when using slave channel with simultaneous channel operation function)
Other than above			Setting prohibited

Bit 7 – 6

CIS 011	CIS 010	Selection of TImn pin input valid edge
0	**0**	**Falling edge**
0	1	Rising edge
1	0	Both edges (when low-level width is measured)
1	1	Both edges (when high-level width is measured)

Bit 3 – 1

MD 013	MD 012	MD 011	Channel n operation mode setting	Corresponding function	TCR count operation
0	**0**	**0**	**Interval timer mode**	Interval timer / **Square wave output** / Divider function / PWM output (master)	**Counting down**
0	1	0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	Event counter mode	External event counter	Counting down
1	0	0	One-count mode	Delay counter / One-shot pulse output / PWM output (slave)	Counting down
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Counting up
Other than above			Setting prohibited		

Bit 0

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MD 010	Setting of count start and interrupt
• Interval timer mode (0, 0, 0)	**0**	**Timer interrupt is not generated when counting is started (timer output does not change either).**
• Capture mode (0, 1, 0)	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change either).
• One-count mode Note2 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation Note 3. At that time, interrupt is not generated.
• Capture and one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change either). Start trigger is invalid during counting operation. At that time, interrupt is not generated.

TAU0 Channel 1 Compare Value Setting

・Timer data register 01 (TDR01)
   Set TAU0 channel 1 compare value to 1295H.

Symbol: TDR01

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

TAU0 Channel 1 Output Mode Setting

・Timer output mode register 0 (TOM0)
   Set the master channel mode output mode.

Symbol: TOM0

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	TOM 03	TOM 02	TOM 01	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**0**	x

Bit 1

TOM 01	Control of channel n timer output mode
0	**Master channel output mode**
1	Slave channel output mode

TAU0 Channel 1 Output Level Control Setting

・Timer output level register 0 (TOL0)
   Set positive logic output.

Symbol: TOL0

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	TOL 03	TOL 02	TOL 01	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**0**	x

Bit 1

TOL 01	Control of channel n timer output level
0	**Positive logic output (active-high)**
1	Negative logic output (active-low)

TAU0 Channel 1 Timer Output Setting

・Timer output register 0 (TO0)

Set timer output value to 0.

Symbol: TO0

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	TO0 3	TO0 2	TO0 1	TO0 0.
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**0**	x

Bit 1

TO0 1	Timer output of channel n
0	**Timer output value is 0.**
1	Timer output value is 1.

TAU0 Channel 1 Timer Output Enable

・Timer output enable register 0 (TOE0)

Enable timer output.

Symbol: TOE0

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	TOE0 3	TOE0 2	TOE0 1	TOE0 0.
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x

Bit 1

TOE0 1	Enable/Disable timer output of channel n
0	Disables timer output.
1	**Enables timer output**

TAU0 Channel 3 Initialization

・Timer mode register 03 (TMR03)
Set TAU0 channel 3 as follows:

・Operation clock: CK00(24MHz)

・Single channel operations

・TI03 pin valid edge used as start trigger and capture trigger

・TI03 pin valid edge: rising

・Input pulse width measurement mode

・Timer interrupt not generated when count star count starts

Symbol: TMR03

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS 031	CKS 030	0	CCS 03	SPLIT 03	STS 032	STS 031	STS 030	CIS 031	CIS 030	0	0	MD 033	MD 032	MD 031	MD 030
0	**0**	x	**0**	**0**	**0**	**0**	**1**	**0**	**1**	x	x	**0**	**1**	**0**	**0**

Bit 15 – 14

CKS 031	CKS 030	Selection of channel n operation clock (f_{MCK})
0	**0**	**Operation clock CKm0 set in timer clock selection register m (TPSm)**
0	1	Operation clock CKm2 set in timer clock selection register m (TPSm)
1	0	Operation clock CKm1 set in timer clock selection register m (TPSm)
1	1	Operation clock CKm3 set in timer clock selection register m (TPSm)

Bit 12

CCS03	Selection of channel n operation clock (f_{TCLK})
0	**Operation clock (f_{MCK}) set in bits CKSmn0 and CKSmn1**
1	Valid edge of input signal from TImn pin

Bit 11

SPLIT03	Selection of 8 or 16-bit timer operation for channels 1 and 3
0	**Operation as 16-bit timer**
1	Operation as 8-bit timer

Bit 10 – 8

STS 032	STS 031	STS 030	Setting start trigger and capture trigger for channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected)
0	**0**	**1**	**Valid edge of TImn pin input is used as both the start trigger and capture trigger**
0	1	0	Both edges of TImn pin input are used as the start trigger and capture trigger
1	0	1	Interrupt signal of master channel is used (when using slave channel with simultaneous channel operation function)
Other than above			Setting prohibited

Bit 7 – 6

CIS 031	CIS 030	Selection of TImn pin input valid edge
0	0	Falling edge
0	**1**	**Rising edge**
1	0	Both edges (when low-level width is measured)
1	1	Both edges (when high-level width is measured)

Bit 3 – 1

MD 033	MD 032	MD 031	Channel n operation mode setting	Corresponding function	TCR count operation
0	0	0	Interval timer mode	Interval timer / Square wave output / Divider function / PWM output (master)	Counting down
0	**1**	**0**	**Capture mode**	**Input pulse interval measurement**	**Counting up**
0	1	1	Event counter mode	External event counter	Counting down
1	0	0	One-count mode	Delay counter / One-shot pulse output / PWM output (slave)	Counting down
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Counting up
Other than above			Setting prohibited		

Bit 0

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MD 030	Setting of count start and interrupt
• Interval timer mode (0, 0, 0) • **Capture mode (0, 1, 0)**	**0**	**Timer interrupt is not generated when counting is started (timer output does not change, either).**
	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change either).
• One-count mode Note2 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation Note 3. At that time, interrupt is not generated.
• Capture and one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change either). Start trigger is invalid during counting operation. At that time, interrupt is not generated.

TAU0 Channel 3 Output Mode Setting

・Timer output mode register 0 (TOM0)
  Set master channel mode output mode.

Symbol: TOM0

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	TOM 03	TOM 02	TOM 01	0
x	x	x	x	x	x	x	x	x	x	x	x	**0**	x	x	x

Bit 3

TOM 03	Control of channel n timer output mode
0	**Master channel output mode**
1	Slave channel output mode

TAU0 Channel 3 Output Level Control Setting

・Timer output level register 0 (TOL0)
  Set positive logic output.

Symbol: TOL0

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	TOL 03	TOL 02	TOL 01	0
x	x	x	x	x	x	x	x	x	x	x	x	**0**	x	x	x

Bit 3

TOL 03	Control of channel n timer output level
0	**Positive logic output (active-high)**
1	Negative logic output (active-low)

TAU0 Channel 3 Timer Output Setting

・Timer output register 0 (TO0)

  Set timer output value to 0.

Symbol: TO0

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	TO0 3	TO0 2	TO0 1	TO0 0.
x	x	x	x	x	x	x	x	x	x	x	x	**0**	x	x	x

Bit 3

TO0 3	Timer output of channel n
0	**Timer output value is 0.**
1	Timer output value is 1.

TAU0 Channel 3 Timer Output Enable Setting

・Timer output enable register 0 (TOE0)

  Enable timer output.

Symbol: TOE0

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	TOE0 3	TOE0 2	TOE0 1	TOE0 0.
x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x	x

Bit 3

TOE0 3	Enable/Disable timer output of channel n
0	Disables timer output
1	**Enables timer output**

TAU0 TI03 Pin Noise Filter Disable

・Noise filter enable register 1 (NFEN1)
  Set noise filter to OFF.

Symbol: NFEN1

7	6	5	4	3	2	1	0
0	0	0	0	TNFEN03	TNFEN02	TNFEN01	TNFEN00
x	x	x	x	**0**	x	x	x

Bit 3

TNFEN03	Enable/disable use of noise filter for TI03 pin input signal
0	**Noise filter OFF**
1	Noise filter ON

TAU0 P00/P30 Settings

- Port output mode register (POM3)
  Set to normal mode.

- Port mode control register (PMC3/PMC0)
  Set to digital input/output.

- Port register (P3)
  Set to low level.

- Port mode register (PM3/PM0)
  Set PM30 to output mode and PM00 to input mode

Symbol: POM3

7	6	5	4	3	2	1	0
0	0	0	0	POM33	POM32	POM31	POM30
x	x	x	x	x	x	x	**0**

Bit 0

POM30	Selection of Pmn pin output mode (m = 0, 2-5; n = 0-6)
0	**Normal output mode**
1	N-ch open drain output (V_{DD} withstand voltage)

Symbol: PMC0

7	6	5	4	3	2	1	0
1	1	1	1	1	1	PMC01	PMC00
x	x	x	x	x	x	x	**0**

Bit 0

PMC00	Selection of Pmn pin digital input/output or analog input (m = 0-3; n = 0-7)
0	**Digital input/output (alternate function other than analog input)**
1	Analog input

Symbol: PMC3

7	6	5	4	3	2	1	0
1	1	1	1	PMC33	PMC32	PMC31	PMC30
x	x	x	x	x	x	x	**0**

Bit 0

PMC30	Selection of Pmn pin digital input/output or analog input (m = 0-3; n = 0-7)
0	**Digital input/output (alternate function other than analog input)**
1	Analog input

Symbol: P3

7	6	5	4	3	2	1	0
0	0	0	0	P33	P32	P31	P30
x	x	x	x	x	x	x	**0**

Bit 0

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

Symbol: PM0

7	6	5	4	3	2	1	0
1	1	1	1	1	1	PM01	PM00
x	x	x	x	x	x	x	**1**

Bit 0

PM00	Selection of Pmn pin input mode (m = 0, 2-5; n = 0-6)
0	Output mode (functions as output port (output buffer ON))
1	**Input mode (functions as input port (output buffer OFF))**

Symbol: PM3

7	6	5	4	3	2	1	0
1	1	1	1	PM33	PM32	PM31	PM30
x	x	x	x	x	x	x	**0**

Bit 0

PM30	Selection of Pmn pin input mode (m = 0, 2-5; n = 0-6)
0	**Output mode (functions as output port (output buffer ON))**
1	Input mode (functions as input port (output buffer OFF))

5.9.6   SAU0 initialization

Figure5.10 shows the flowchart for SAU0 initialization.
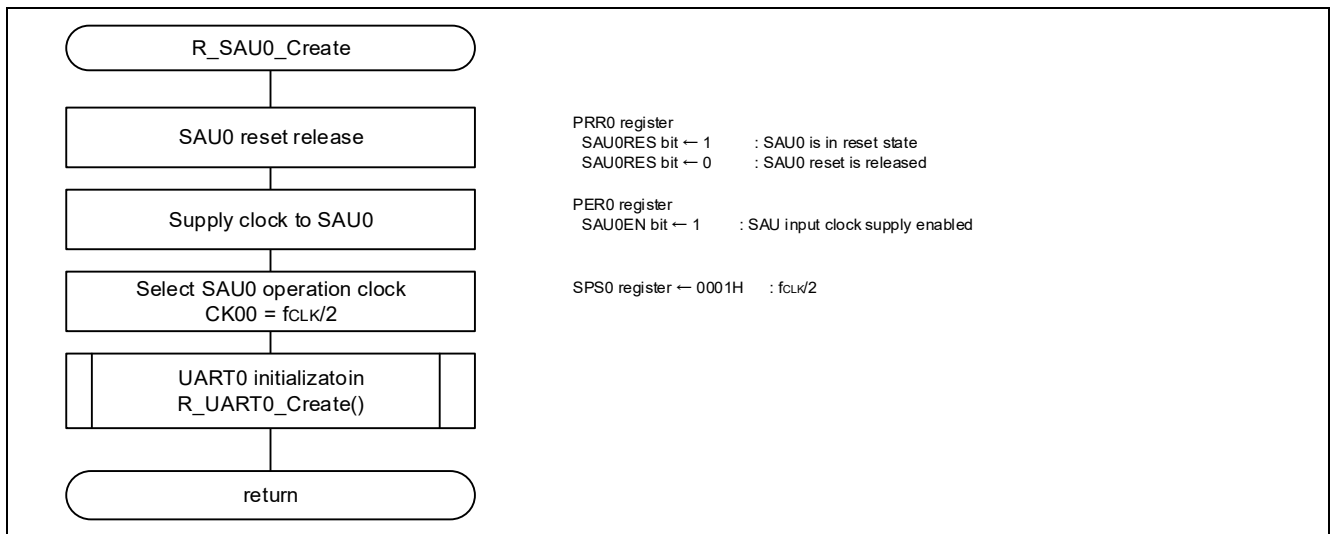
Figure 5.10   SAU0 Initialization

SAU0 Reset Release

・Peripheral reset control register 0 (PRR0)

  Release SAU0 from reset state.

Symbol: PRR0

7	6	5	4	3	2	1	0
0	IICA1RES	ADCRES	IICA0RES	0	SAU0RES	0	TAU0RES
x	x	x	x	x	**0**	x	x

Bit 2

SAU0RES	Control of serial array unit reset release
0	**Reset release of serial array unit**
1	Reset state of serial array unit

SAU0 Clock Supply

・Peripheral enable register 0 (PER0)

  Enable clock supply to SAU0.

Symbol: PER0

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

Bit 2

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

SAU0 Operation Clock Selection

・Serial clock selection register 0 (TPS0)
  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
−	−	−	−	−	−	−	−	x	x	x	x	**0**	**0**	**0**	**1**

Bit 3 - 0

PRS 003	PRS 002	PRS 001	PRS 000		Operation Clock (CK00) 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

5.9.7　　UART0 initialization

Figure5.11 shows the flowchart for UART0 initialization.
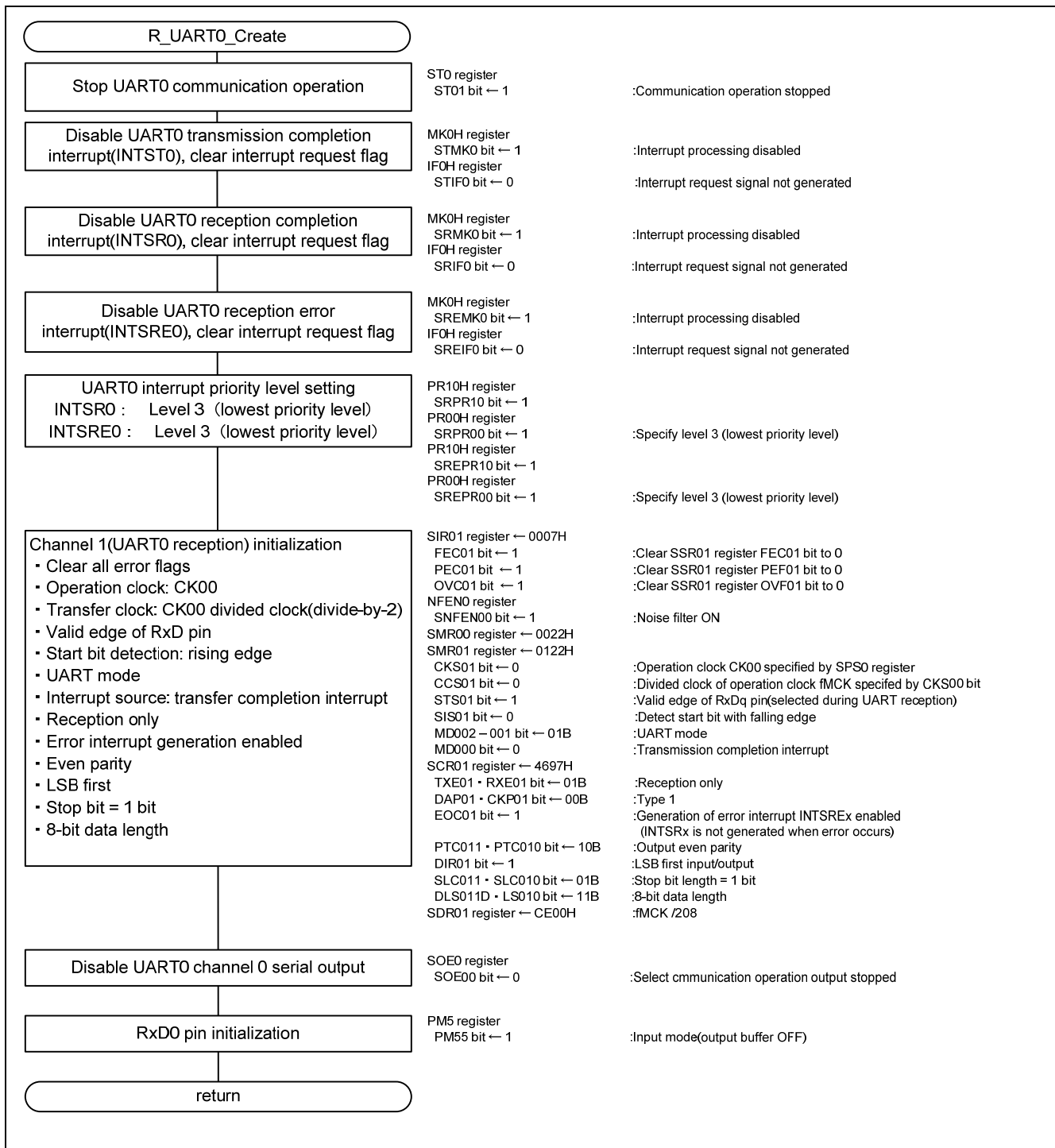
Figure 5.11　UART0 Initialization

UART0 Communication Operation Stop

・Serial channel stop register 0 (ST0)
 Set UART0 to communication operation stop.

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**	x

Bit 1

ST01	Trigger to start operations of channel n
0	No trigger operation
1	**SEmn bit is cleared to 0 and the communication operation is stopped.**

UART0 Transmission Completion Interrupt/Reception Error Interrupt

・Interrupt request flag register (MK0H)
Disable UART0 transmission completion interrupt/reception error interrupt.

・Interrupt request flag register (IF0H)
Clear UART0 transmission completion interrupt/reception error interrupt flag.

Symbol: MK0H

7	6	5	4	3	2	1	0
STMK1 CSIMK10 IICMK10	TMMK00	SREMK0	1	1	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	PMK6
x	x	**1**	x	x	**1**	**1**	x

Bit 5

SREMK0	Control of interrupt processing
0	Enables interrupt processing
1	**Disables interrupt processing**

Bit 2

SRMK0	Control of interrupt processing
0	Enables interrupt processing
1	**Disables interrupt processing**

Bit 1

STMK0	Control of interrupt processing
0	Enables interrupt processing
1	**Disables interrupt processing**

Symbol: IF0H

7	6	5	4	3	2	1	0
ST1IF CSIIF10 IICIF10	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 is not generated**
1	Interrupt request signal is generated and goes to interrupt request state.

Bit 2

SRIF0	Interrupt request flag
0	**Interrupt request signal is not generated**
1	Interrupt request signal is generated and goes to interrupt request state.

Bit 1

STIF0	Interrupt request flag
0	**Interrupt request signal is not generated**
1	Interrupt request signal is generated and goes to interrupt request state.

UART0 Interrupt Priority Level Settings

・Priority level flag registers (PR10L, PR00L)
  Set reception/reception error interrupt to level 3 (lowest priority level).

Symbol: PR00H

7	6	5	4	3	2	1	0
STPR01 CSIPR010 IICPR010	TMPR000	SREPR00	1	1	SRPR00 CSIPR001 IICPR001	STPR00 CSIPR000 IICPR000	PPR06
X	x	**1**	x	x	**1**	x	x

Symbol: PR10H

7	6	5	4	3	2	1	0
STPR11 CSIPR110 IICPR110	TMPR100	SREPR10	1	1	SRPR10 CSIPR101 IICPR101	STPR10 CSIPR100 IICPR100	PPR16
x	x	**1**	x	x	**1**	x	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)**

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	FECT 01	PECT 01	OVCT 01
—	—	—	—	—	—	—	—	—	—	—	—	—	**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 Noise Filter ON

・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	SNFEN10	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 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
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**	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 detected with falling edge.** **The input communication data is captured as is.**
1	Start bit 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^{2}C 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	Transmission/reception

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 0 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	DLS 010	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 Output Enable Setting

- Serial output enable register 0 (SOE0)
  Disable 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	0

Bit 0

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

RxD0 Pin Initialization

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

  Symbol: PM5

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

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)**

5.9.8   INTP initialization
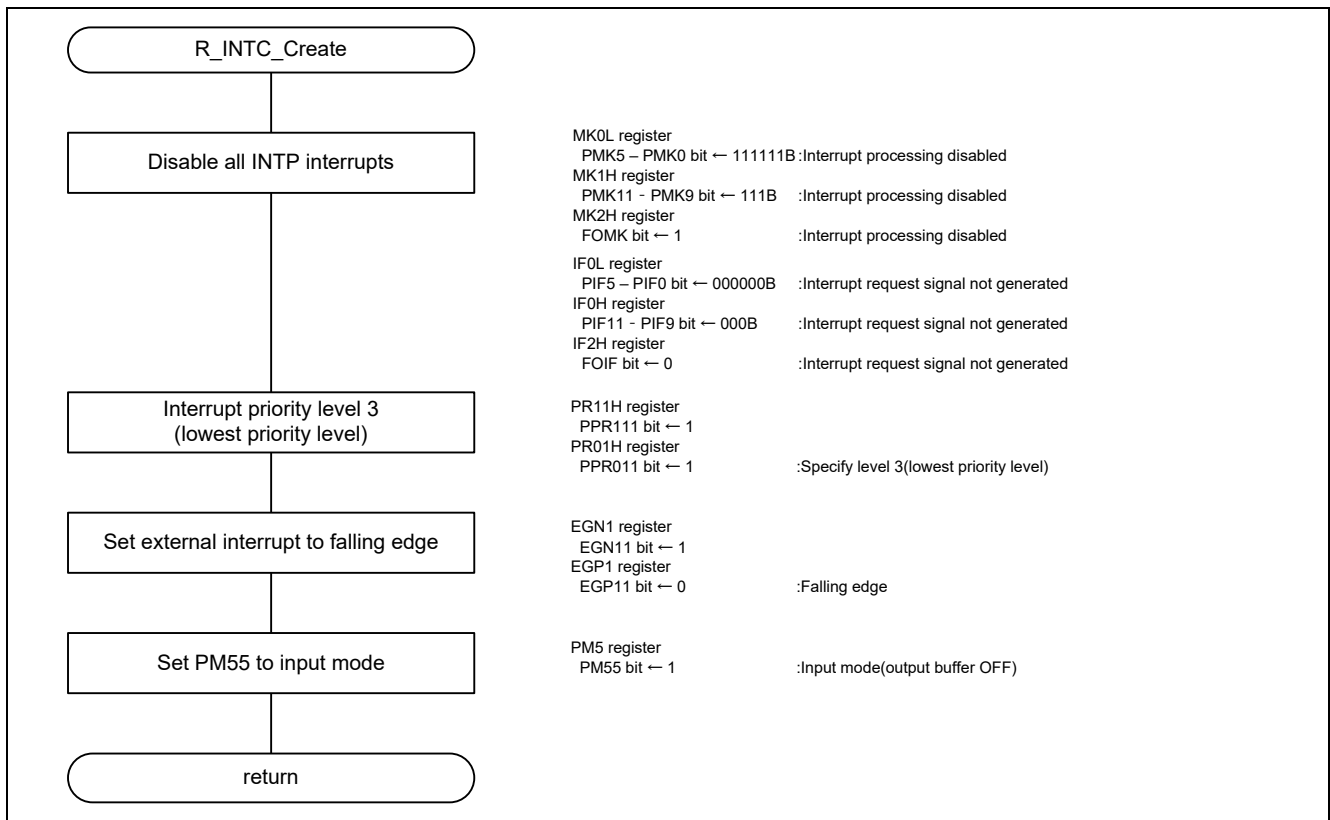
Figure5.12 shows the INTP initialization flowchart.

```
        ┌──────────────────────────┐
        │       R_INTC_Create       │
        └──────────────────────────┘
                     │
        ┌──────────────────────────┐    MK0L register
        │  Disable all INTP interrupts │      PMK5 – PMK0 bit ← 111111B :Interrupt processing disabled
        └──────────────────────────┘    MK1H register
                                          PMK11 - PMK9 bit ← 111B    :Interrupt processing disabled
                                         MK2H register
                                          FOMK bit ← 1               :Interrupt processing disabled

                                         IF0L register
                                          PIF5 – PIF0 bit ← 000000B  :Interrupt request signal not generated
                                         IF0H register
                                          PIF11 - PIF9 bit ← 000B    :Interrupt request signal not generated
                                         IF2H register
                                          FOIF bit ← 0               :Interrupt request signal not generated

        ┌──────────────────────────┐    PR11H register
        │   Interrupt priority level 3 │      PPR111 bit ← 1
        │    (lowest priority level)    │    PR01H register
        └──────────────────────────┘      PPR011 bit ← 1            :Specify level 3(lowest priority level)

        ┌──────────────────────────┐    EGN1 register
        │ Set external interrupt to falling edge │   EGN11 bit ← 1
        └──────────────────────────┘    EGP1 register
                                          EGP11 bit ← 0              :Falling edge

        ┌──────────────────────────┐    PM5 register
        │   Set PM55 to input mode   │      PM55 bit ← 1             :Input mode(output buffer OFF)
        └──────────────────────────┘
                     │
        ┌──────────────────────────┐
        │          return           │
        └──────────────────────────┘
```

Figure 5.12   INTP Initialization

INTP All Interrupts Disable

・Interrupt request flag register (MK0L/MK1H)
  Disable all INTP interrupts.
・Interrupt request flag register (IF0L/IF1H)
  Clear all INTP interrupt request flags.

Symbol: MK0L

7	6	5	4	3	2	1	0
PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK
1	**1**	**1**	**1**	**1**	**1**	x	x

Bit 7 – 2 (n = 0 - 5)

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

Symbol: MK1H

7	6	5	4	3	2	1	0
PMK11	PMK10	PNK9	PMK8	PMK7	KRMK	TMKAMK	ADMK
1	**1**	**1**	x	x	x	x	x

Bit 7 – 5(n = 9 - 11)

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

Symbol: IF0L

7	6	5	4	3	2	1	0
PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF
0	**0**	**0**	**0**	**0**	**0**	x	x

Bit 7 - 2 (n = 0 - 5)

PIFn	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state

Symbol: IF1H

7	6	5	4	3	2	1	0
PIF11	PIF10	PIF9	PIF8	PIF7	KRIF	TMKAIF	ADIF
0	**0**	**0**	x	x	x	x	**0**

Bit 7 – 5(n = 9 - 11)

PIFn	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state

INTP Interrupt Priority Level 3 (lowest priority level)
・Priority level flag registers (PR11H, PR01H)
  Set interrupt to level 3 (lowest priority level)

Symbol: PR11H

7	6	5	4	3	2	1	0
PPR111	PPR110	PPR19	PPR18	PPR17	KRPR1	TMKAPR1	ADPR1
1	x	x	x	x	x	x	X

Symbol: PR01H

7	6	5	4	3	2	1	0
PPR011	PPR010	PPR09	PPR08	PPR07	KRPR0	TMKAPR0	ADPR0
1	x	x	x	x	x	x	x

Bit 7

PPR011	PPR111	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)**

External Interrupt Falling Edge
・External interrupt falling edge enable register (EGN1)
  External interrupt rising edge enable register (EGP1)
  Set to rising/falling edges.

Symbol: EGN1

7	6	5	4	3	2	1	0
0	0	0	0	EGN11	EGN10	EGN9	EGN8
x	x	x	x	**1**	x	X	X

Symbol: EGP1

7	6	5	4	3	2	1	0
0	0	0	0	EGP11	EGP10	EGP9	EGP8
x	x	x	x	**0**	x	x	x

Bit 3

EGP11	EGN11	INTPn pin valid edge selection (n = 0-11)
0	0	Edge detection disabled
0	**1**	**Falling edge**
1	0	Rising edge
1	1	Both rising and falling edges

PM55 Input Mode Setting

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

Symbol: PM5

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

Bit 5

PM55	I/O mode selection for Pmn pin
0	Output mode (functions as output port, output buffer ON)
1	**Input mode (functions as input port, output buffer OFF)**

5.9.9   Main initialization

Figure5.13 shows the flowchart for main initialization.
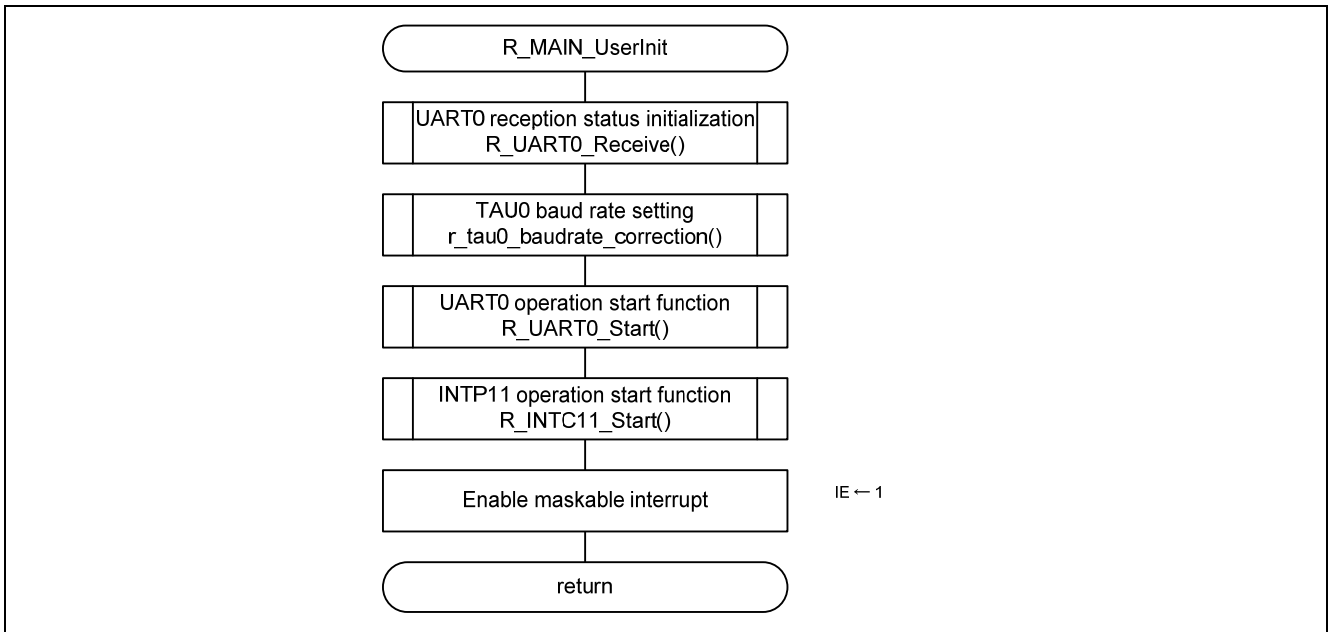
Figure 5.13   Main Initialization

5.9.10   UART0 reception status initialization

Figure5.14 shows the flowchart for UART0 reception status initialization.
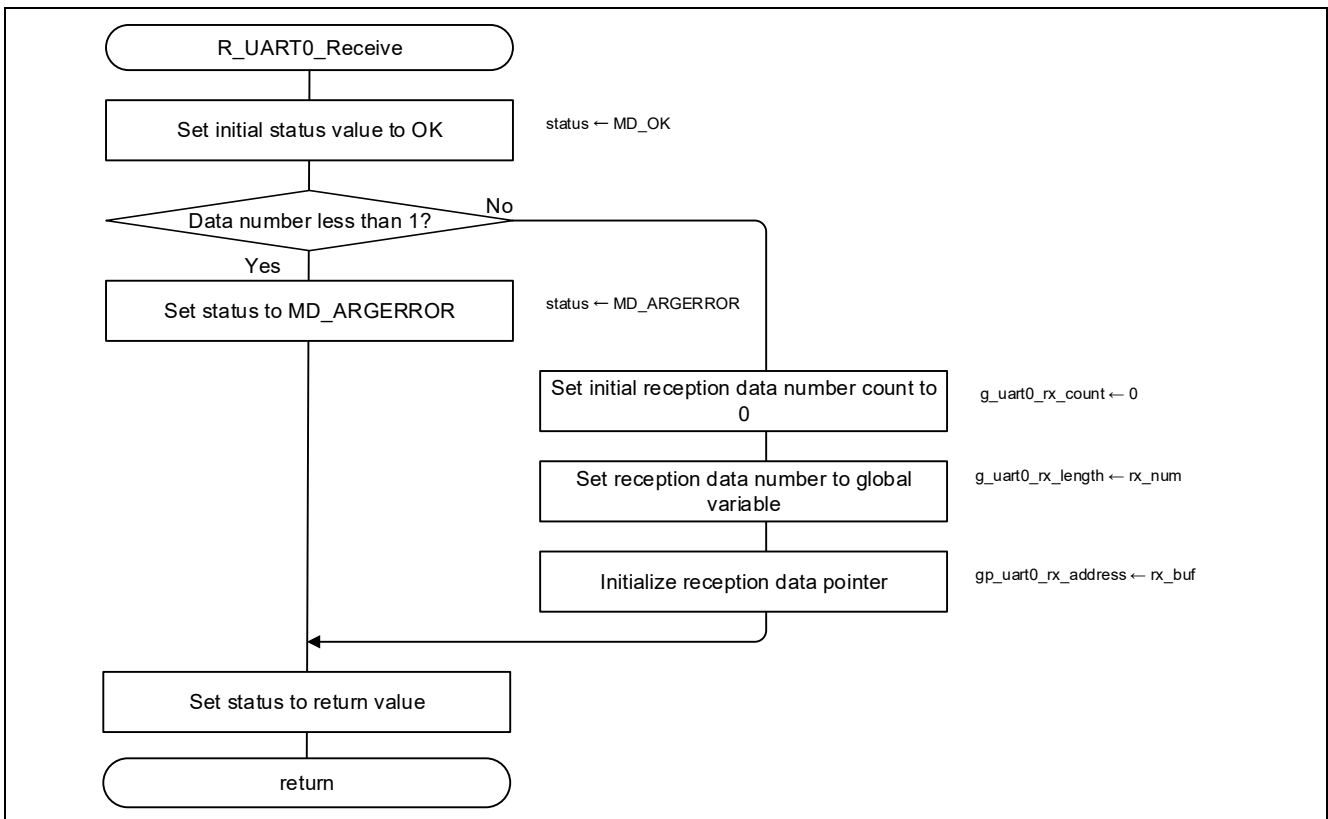
Figure 5.14   UART0 Reception Status Initialization

5.9.11   UART0 operation start function

Figure5.15 shows the flowchart for the UART0 operation start function.

```
                        R_UART0_Start

                     Enable UART0 operation                 SS0 register
                                                              SS01 bit ← 1  : Shift to communication wait state

              Clear UART0 reception completion/reception     IF0H register
                   error interrupt request flag                SRIF0 bit  ← 0  : Interrupt request signal is not generated
                                                               SREIF0 bit ← 0  : Interrupt request signal is not generated

              Enable UART0 reception completion/reception     MK0H register
                   error interrupt request                     SRMK0 bit  ← 0  : Interrupt processing enabled
                                                               SREMK0 bit ← 0  : Interrupt processing enabled

                            return
```

Figure 5.15   UART0 Operation Start Function

UART0 Operation Enable

・Serial channel start register 0 (SS0)
  Enable 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	SS0 1	SS0 0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x

Bit 1

SS0 1	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 Reception Complete/Reception Error Interrupt Enable

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

Symbol: IF0H

7	6	5	4	3	2	1	0
ST1IF CSIIF10 IICF10	TMIF00	SREIF0	0	0	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	PIF6
x	x	**0**	x	x	**0**	x	x

Bit 5

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

Bit 2

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

Symbol: MK0H

7	6	5	4	3	2	1	0
ST1MK CSIMK10 IICMK10	TMMK00	SREMK0	1	1	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	PMK6
x	x	**0**	x	x	**0**	x	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

5.9.12 TAU0 baud rate setting

Figure5.16 shows the flowchart for TAU0 baud rate setting (1/2).
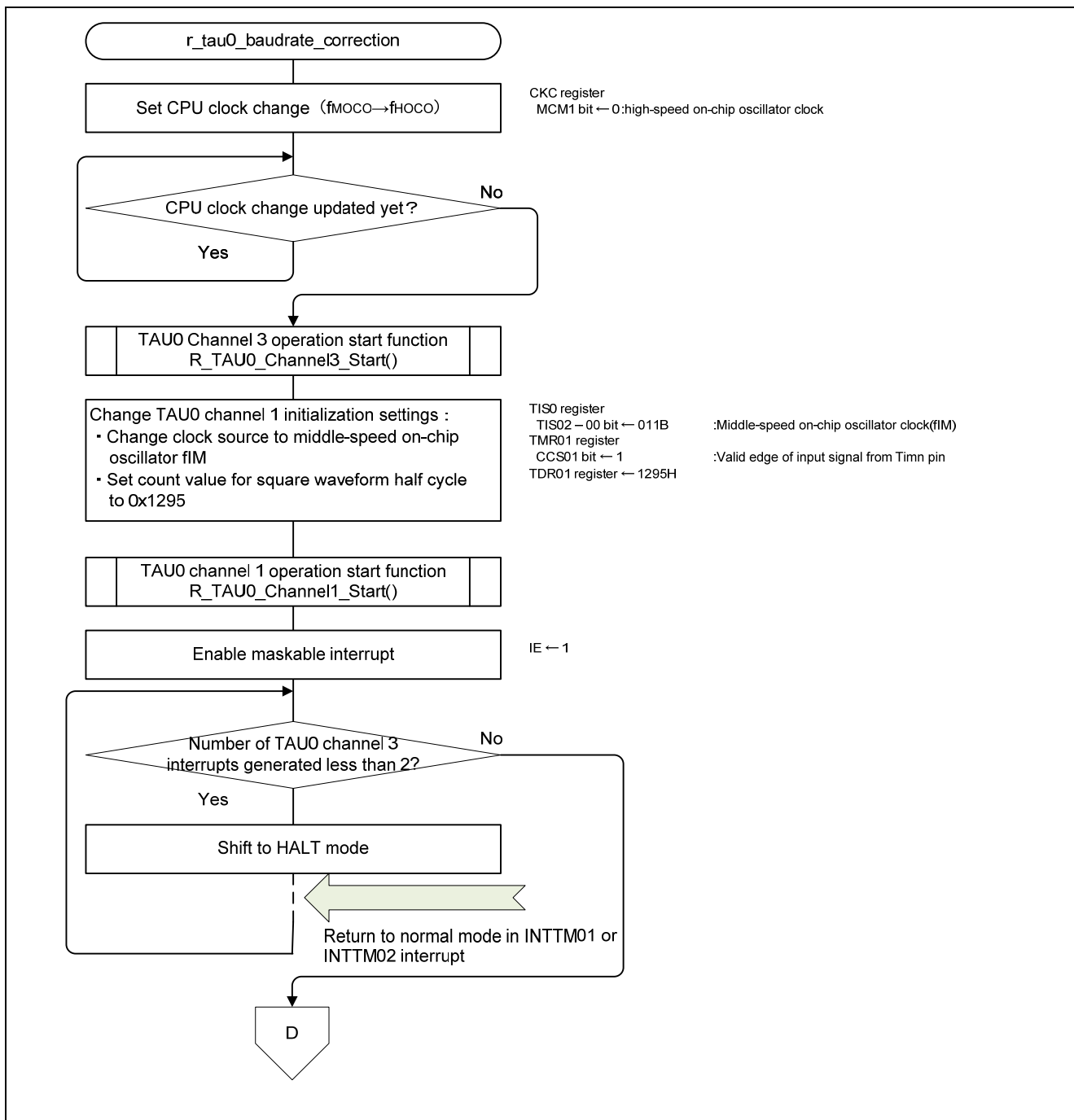
Figure 5.16 TAU0 Baud Rate Setting (1/2)

Figure5.17 shows the flowchart for TAU0 baud rate setting (2/2).
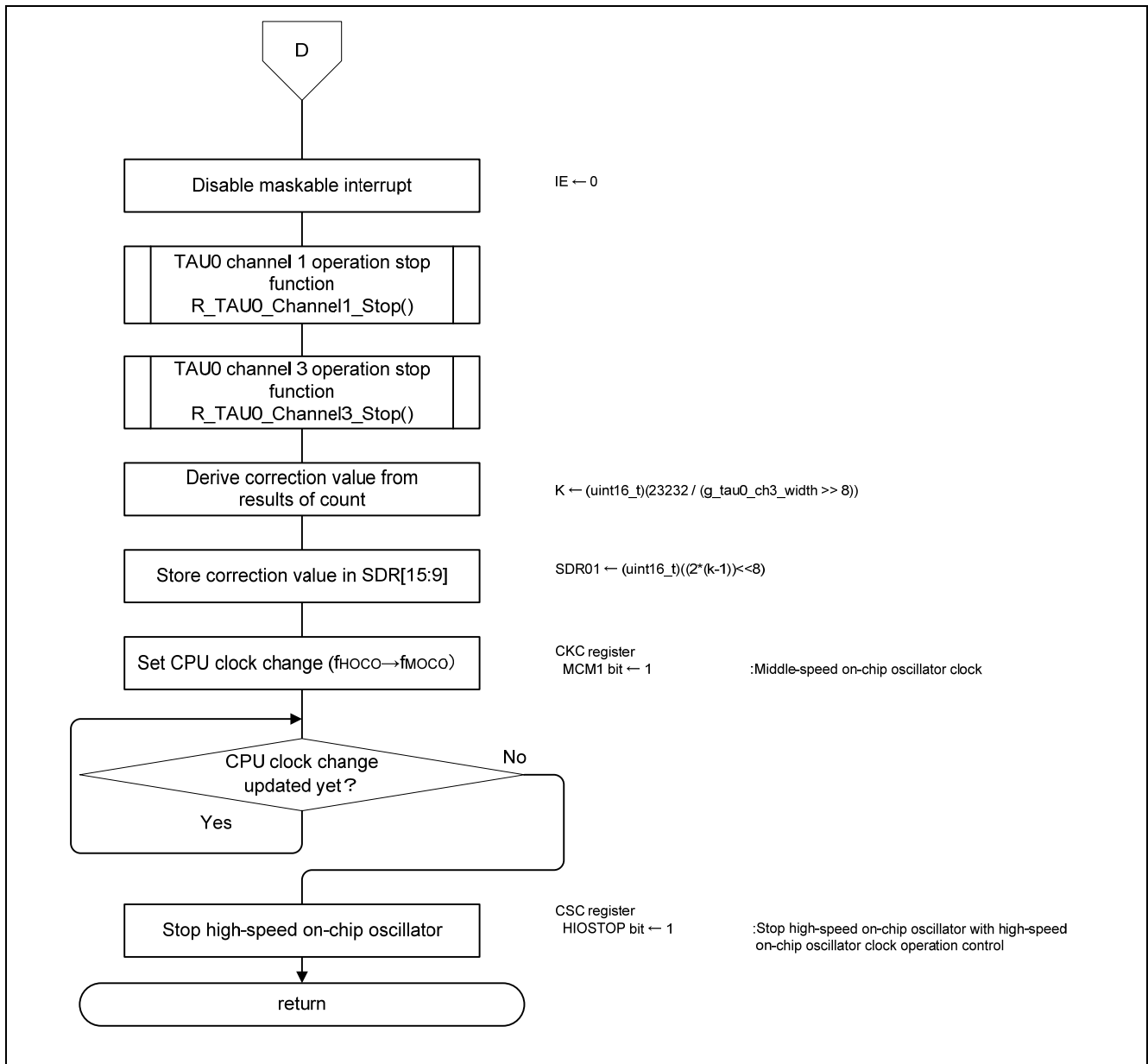
Figure 5.17   TAU0 Baud Rate Setting (2/2)

CPU Clock Change Setting

・System clock control register (CKC)
    Change setting from high-speed to middle-speed on-chip oscillator clock.

Symbol: CKC

7	6	5	4	3	2	1	0
CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
x	x	x	x	x	x	x	**1**

Bit 0

MCM1	Operation control of main on-chip oscillator clock (f_{OCO})
0	High-speed on-chip oscillator clock
1	**Middle-speed on-chip oscillator clock**

TAU0 Channel 1 Timer Input Setting

・Timer input selection register 0 (TIS0)
    Set to middle-speed on-chip oscillator clock.

Symbol: TIS0

7	6	5	4	3	2	1	0
0	TIS06	TIS05	TIS04	0	TIS02	TIS01	TIS00
x	x	x	x	x	**0**	**1**	**1**

Bit 2 - 0

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	Event input signal from ELC
0	1	0	Input signal of timer input pin (TI01)
0	**1**	**1**	**Middle-speed on-chip oscillator clock(f_{IM})**
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

TAU0 Channel 1 Count Clock Setting

・Timer mode register 01 (TMR01)
　Set to valid edge of input signal.

Symbol: TMR01

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS 011	CKS 010	0	CCS 01	SPLIT 01	STS 012	STS 011	STS 010	CIS 011	CIS 010	0	0	MD 013	MD 012	MD 011	MD 010
x	x	x	**1**	x	x	x	x	x	x	x	x	x	x	x	x

Bit 12

CCS01	Selection of count clock (f_{TCLK}) for channel n
0	Operation clock (f_{MCK}) specified by CKSmn0 and CKSmn1 bits
1	**Valid edge of input signal from TImn pin.**

TAU0 Channel 1 Compare Value Setting

・Timer data register 01 (TDR01)
　Set TAU0 channel 1 compare value to 1295H.

Symbol: TDR01

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

5.9.13   INTP11 operation start function

Figure5.18 shows the flowchart for INTP11 operation start function.
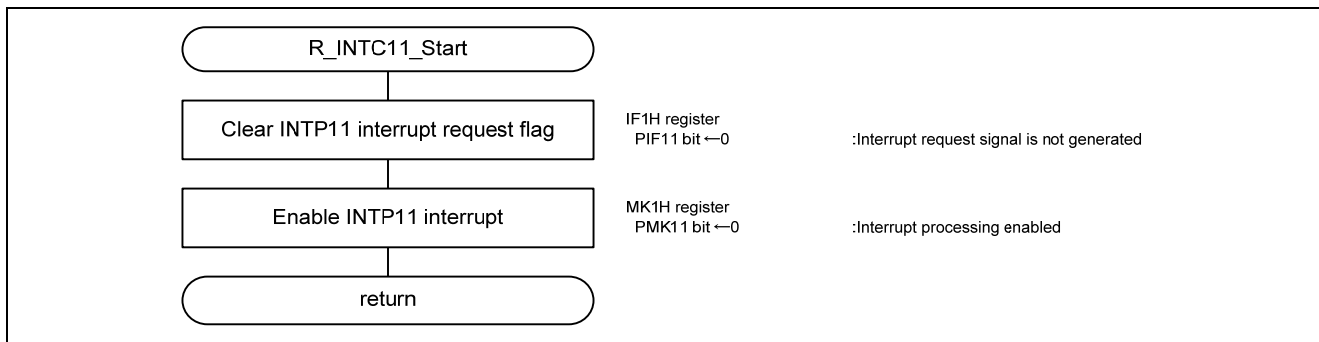
Figure 5.18   INTP11 Operation Start Function

INTP11 Interrupt Enable
 ・Interrupt request flag register (IF1H)
   Clear interrupt request flag.
 ・Interrupt request flag register (MK1H)
   Set to interrupt enabled.

Symbol: IF1H

7	6	5	4	3	2	1	0
PIF11	PIF10	PIF9	PIF8	PIF7	KRIF	TMKAIF	ADIF
0	x	x	x	x	x	x	x

Bit 7

PIF11	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state.

Symbol: MK1H

7	6	5	4	3	2	1	0
PMK11	PMK10	PMK9	PMK8	PMK7	KRMK	TMKAMK	ADMK
0	x	x	x	x	x	x	x

Bit 7

PMK11	Interrupt processing control
0	**Enables interrupt processing**
1	Disables interrupt processing

5.9.14   TAU0 channel 3 operation start function

Figure5.19 shows the flowchart for TAU0 channel 3 operation start function.
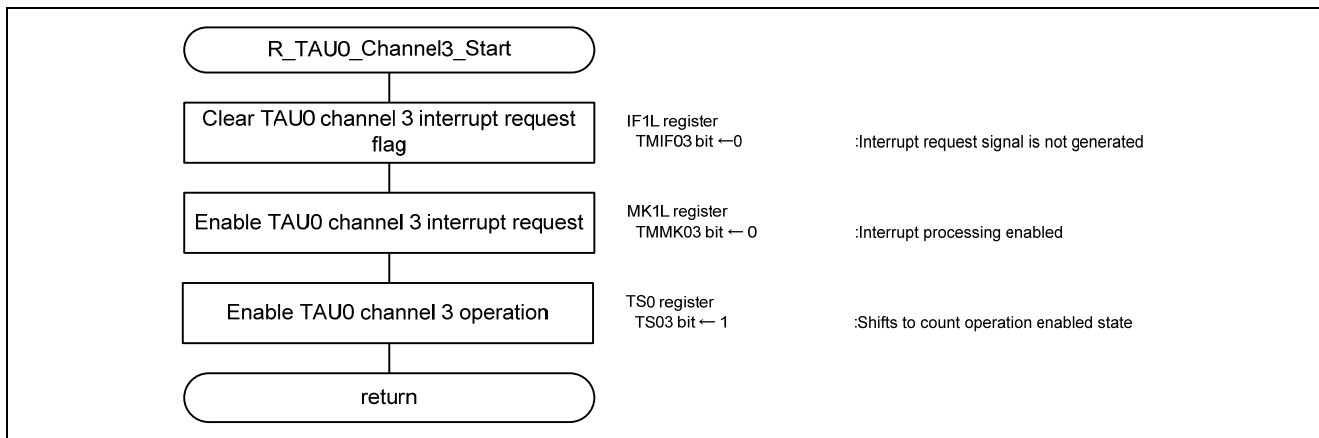
Figure 5.19   TAU0 Channel 3 Operation Start Function

TAU0 Channel 3 Interrupt Enable
・Interrupt request flag register (IF1L)
  Clear interrupt request flag.
・Interrupt request flag register (MK1L)
  Set to interrupt enabled.

Symbol: IF1L

7	6	5	4	3	2	1	0
TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	IICAIF0	SREIF1	SRIF1 CSIIF11 IICIF11
0	x	x	x	x	x	x	x

Bit 7

TMIF03	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state.

Symbol: MK1L

7	6	5	4	3	2	1	0
TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	IICAMK0	SREMK1	SRMK1 CSIMK11 IICMK11
0	x	x	x	x	x	x	x

Bit 7

TMMK03	Interrupt processing control
0	**Enables interrupt processing**
1	Disables interrupt processing

TAU0 Channel 3 Count Operation Enable

・Timer channel start register 0 (TS0)
  Set to count enabled state.

Symbol: TS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	TSH 03	0	TSH 01	0	0	0	0	0	TS03	TS02	TS01	TS00
x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x	x

Bit 3

TS03	Operation enable (start) trigger for channel n
0	No trigger operation
1	**TEmn bit is set to 1, goes to count operation enabled state.**

5.9.15   TAU0 channel 1 operation start function

Figure5.20 shows the flowchart for the TAU0 channel 1 operation start function.
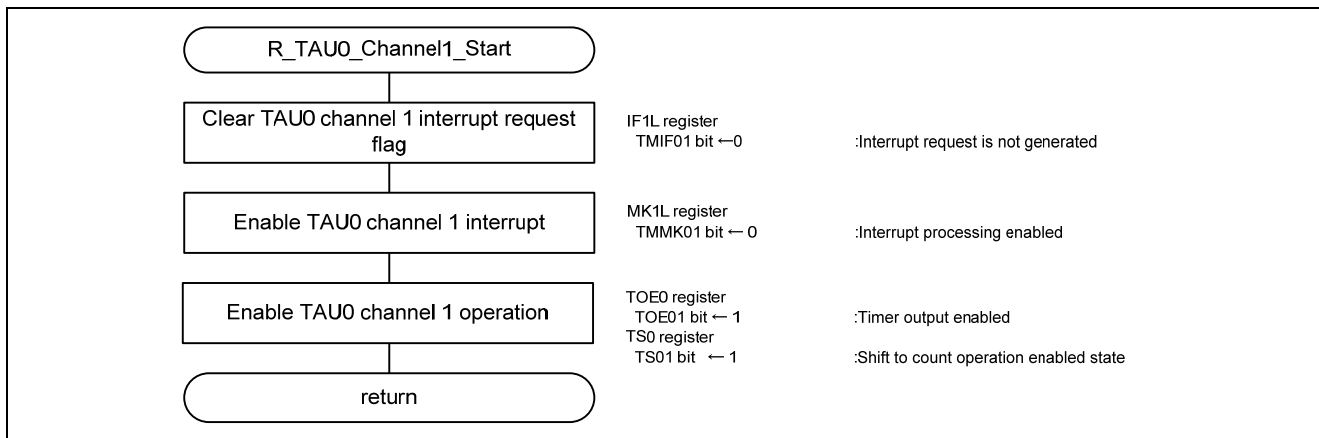
Figure 5.20   TAU0 Channel 1 Operation Start Function

TAU0 Channel 1 Interrupt Enable
・Interrupt request flag register (IF1L)
  Clear interrupt request flag.
・Interrupt request flag register (MK1L)
  Set to interrupt enabled.

Symbol: IF1L

7	6	5	4	3	2	1	0
TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	IICAIF0	SREIF1	SRIF1 CSIIF11 IICIF11
x	x	**0**	x	x	x	x	x

Bit 5

TMIF01	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state

Symbol: MK1L

7	6	5	4	3	2	1	0
TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	IICAMK0	SREMK1	SRMK1 CSIMK11 IICMK11
x	x	**0**	x	x	x	x	x

Bit 5

TMMK01	Interrupt processing control
0	**Enables interrupt processing**
1	Disables interrupt processing

TAU0 Channel 1 Timer Output Enable

・Timer output enable register 0 (TOE0)
Set to timer output enabled.

Symbol: TOE0

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	TOE0 3	TOE0 2	TOE0 1	TOE0 0.
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x

Bit 1

TOE0 1	Timer output enable/disable of channel n
0	Disables timer output
1	**Enables timer output**

TAU0 Channel 1 Operation Enable

・Timer channel start register 0 (TS0)
Set to count enabled state.

Symbol: TS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	TSH 03	0	TSH 01	0	0	0	0	0	TS03	TS02	TS01	TS00
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x

Bit 1

TS01	Operation enable (start) trigger for channel n
0	No trigger operation
1	**TEmn bit is set to 1, goes to count operation enabled state.**

5.9.16   TAU0 channel 1 operation stop function

Figure5.21 shows the flowchart for the TAU0 channel 1 operation stop function.
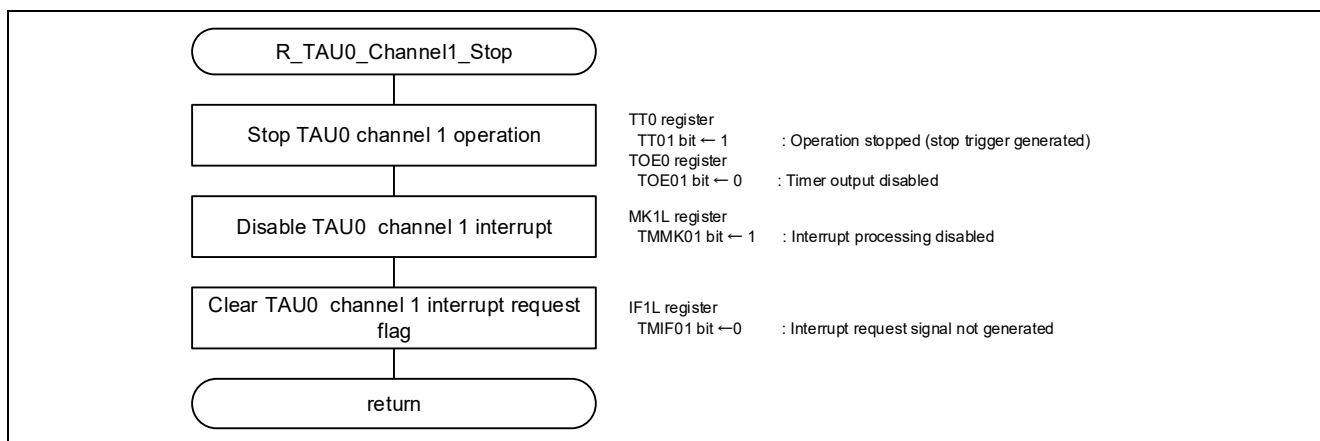
Figure 5.21   TAU0 Channel 1 Operation Stop Function

TAU0 Channel 1 Operation Stop

・Timer channel stop register 0 (TT0)

Set to TAU0 operation stop.

Symbol: TT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	TTH 03	0	TTH 01	0	0	0	0	0	TT0 3	TT0 2	TT0 1	TT0 0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x

Bit 1

TT 01	Operation stop trigger for channel n
0	Clears TEmn bit to 0 and goes to count operation stop state.
1	**Operation stops (stop trigger generated)**

TAU0 Channel 1 Timer Output Disable

・Time output enable register 0 (TOE0)
   Set timer output to disabled.

Symbol: TOE0

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	TOE0 3	TOE0 2	TOE0 1	TOE0 0.
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**0**	x

Bit 1

TOE0 1	Timer output enable/disable of channel n
0	**Timer output is disabled**
1	Timer output is enabled

TAU0 Channel 1 Interrupt Disable

・Interrupt request flag register (MK1L)
   Set to interrupt disabled.
・Interrupt request flag register (IF1L)
   Clear interrupt request flag.

Symbol: MK1L

7	6	5	4	3	2	1	0
TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	IICAMK0	SREMK1	SRMK1 CSIMK11 IICMK11
x	x	**1**	x	x	x	x	x

Bit 5

TMMK01	Interrupt request flag
0	Enables interrupt processing
1	**Disables interrupt processing**

Symbol: IF1L

7	6	5	4	3	2	1	0
TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	IICAIF0	SREIF1	SRIF1 CSIIF11 IICIF11
x	x	**0**	x	x	x	x	x

Bit 5

TMIF01	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state.

RENESAS

5.9.17   TAU0 channel 3 operation stop function

Figure5.22 shows the flowchart for TAU0 channel 3 operation stop function.
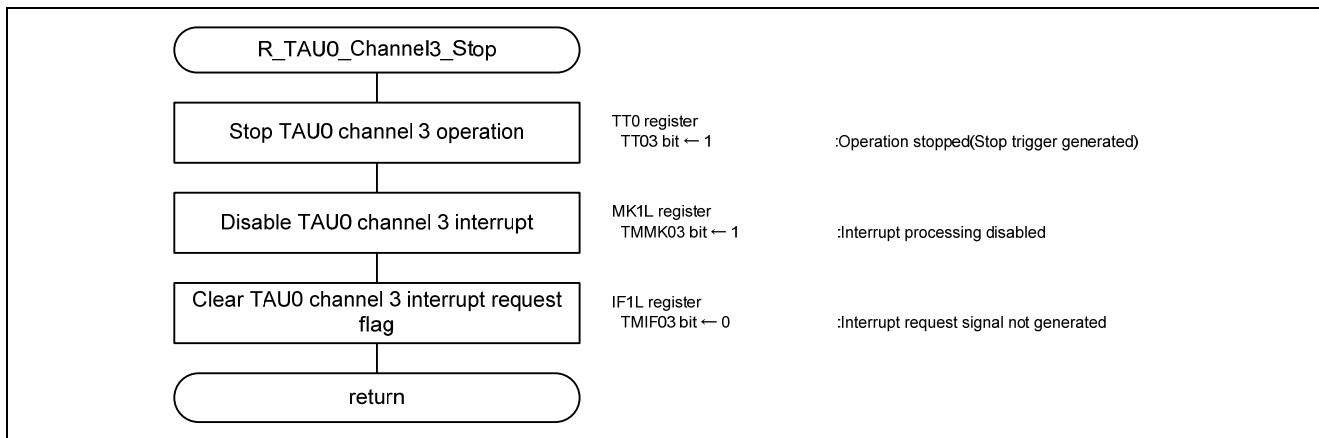
Figure 5.22   TAU0 Channel 3 Operation Stop Function

TAU0 Channel 3 Operation Disable

・Timer channel stop register 0 (TT0)
   Set to TAU0 operation stop.

Symbol: TT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	TTH 03	0	TTH 01	0	0	0	0	0	TT0 3	TT0 2	TT0 1	TT0 0
x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x	x

Bit 3

TT 03	Operation stop trigger for channel n
0	Clears TEmn bit to 0 and goes to count operation stop state.
1	**Operation stopped (stop trigger generated)**

TAU0 Channel 3 Interrupt Disable
・Interrupt request flag register (MK1L)
  Set to interrupt disabled.
・Interrupt request flag register (IF1L)
  Clear interrupt request flag.

Symbol: MK1L

7	6	5	4	3	2	1	0
TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	IICAMK0	SREMK1	SRMK1 CSIMK11 IICMK11
1	x	x	x	x	x	x	x

Bit 7

TMMK03	Interrupt processing control
0	Enables interrupt processing
1	**Disables interrupt processing**

Symbol: IF1L

7	6	5	4	3	2	1	0
TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	IICAIF0	SREIF1	SRIF1 CSIIF11 IICIF11
0	x	x	x	x	x	x	x

Bit 7

TMIF03	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state.

5.9.18   Wait function for LED blinking

Figure5.24 shows the flowchart for the wait function for LED blinking.
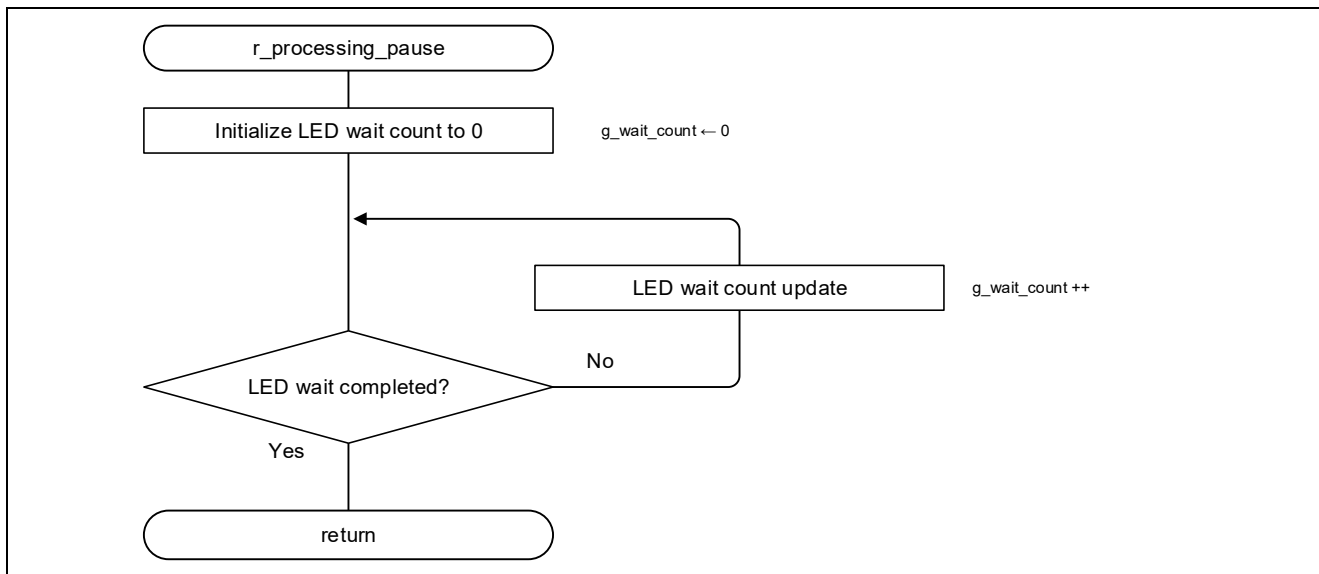
Figure 5.24   Wait Function for LED Blinking

5.9.19   TAU0 channel 3 count complete interrupt function

Figure5.25 shows the flowchart for the TAU0 channel 3 count complete interrupt function.
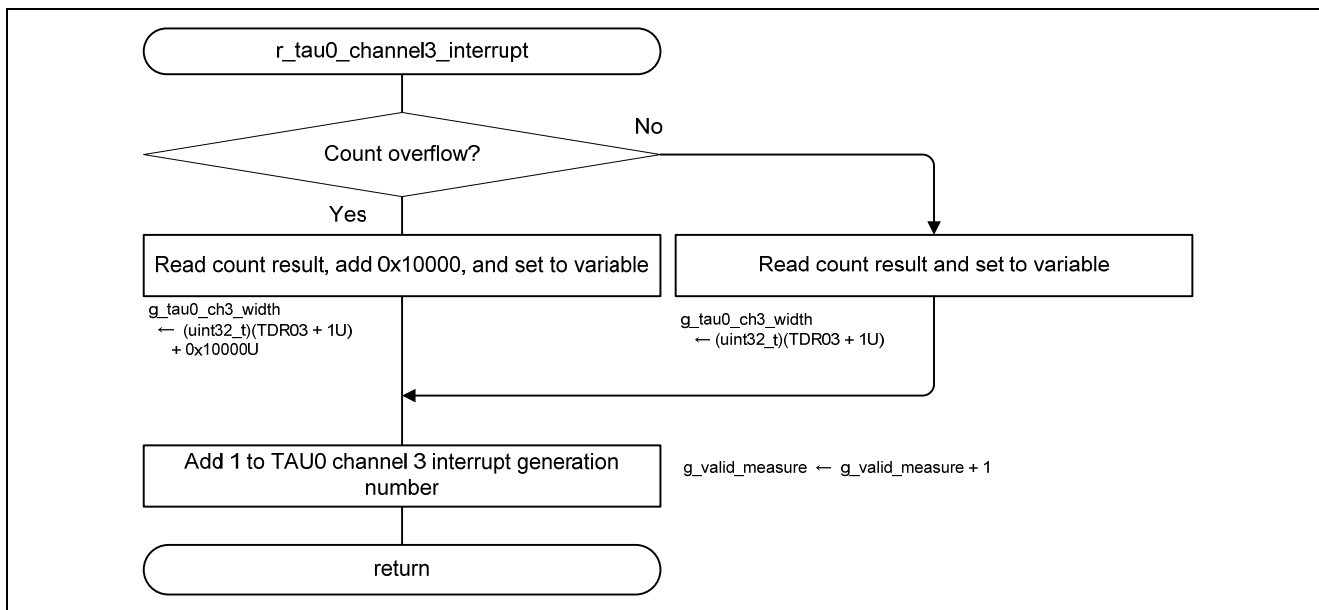
Figure 5.25   TAU0 Channel 3 Count Complete Interrupt Function

5.9.20   INTP11 interrupt processing

Figure5.26 shows the flowchart for INTP11 interrupt processing.
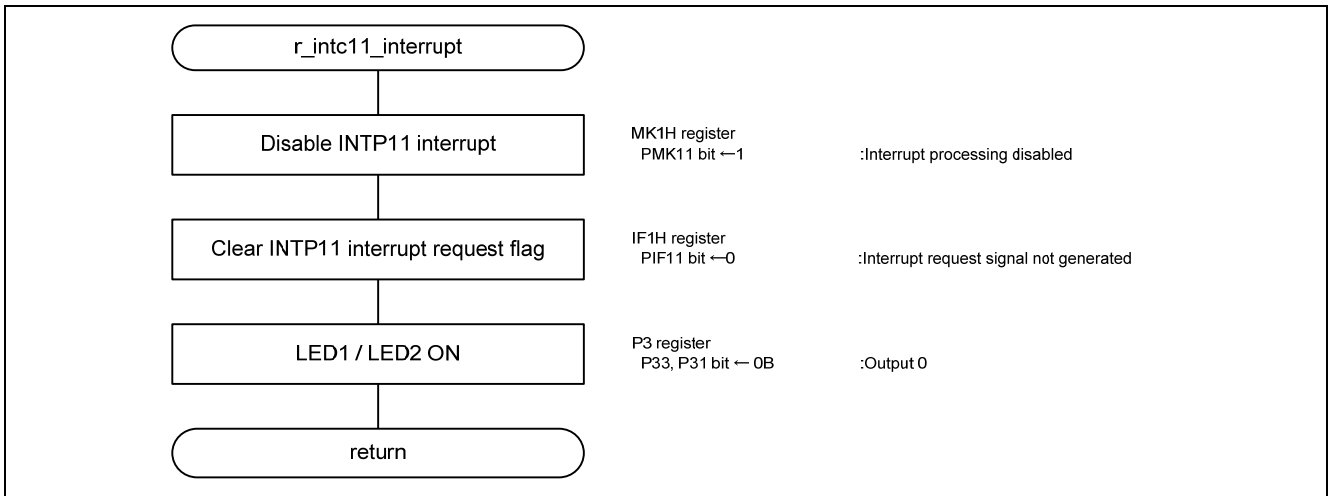
Figure 5.26   INTP11 Interrupt Processing

INTP11 Interrupt Disable

・Interrupt request flag register (MK1H)
    Set INTP interrupt to disabled.
・Interrupt request flag register (IF1H)
    Clear INTP interrupt request flag.

Symbol: MK1H

7	6	5	4	3	2	1	0
PMK11	PMK10	PMK9	PMK8	PMK7	KRMK	TMKAMK	ADMK
1	x	x	x	x	x	x	x

Bit 7

PMK11	Interrupt request flag
0	Enables interrupt processing
1	**Disables interrupt processing**

Symbol: IF1H

7	6	5	4	3	2	1	0
PIF11	PIF10	PIF9	PIF8	PIF7	KRIF	TMKAIF	ADIF
0	x	x	x	x	x	x	x

Bit 7

PIF11	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state.

5.9.21   UART0 reception complete interrupt processing

Figure5.27 shows the flowchart for UART0 reception complete interrupt processing.
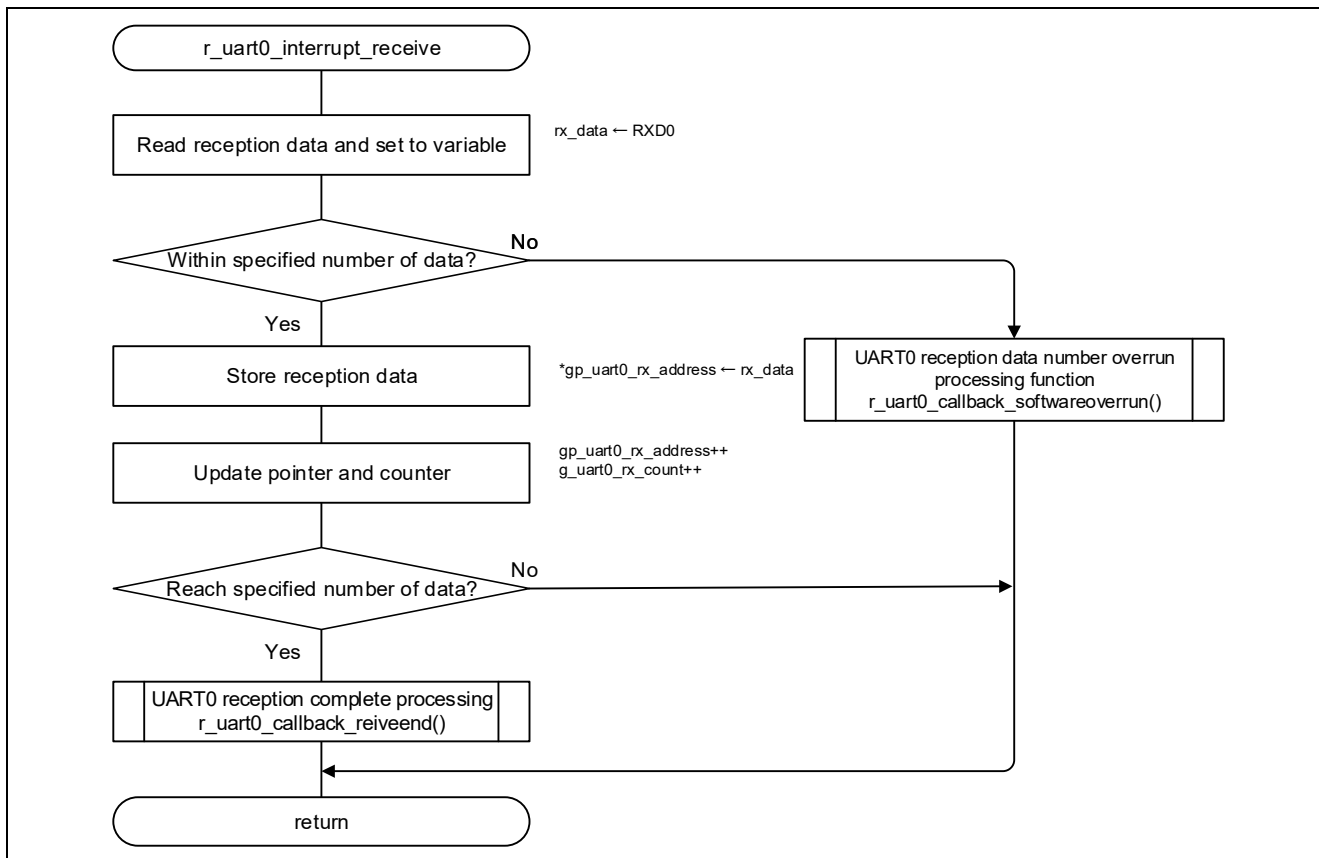
Figure 5.27   UART0 Reception Complete Interrupt Processing

5.9.22   UART0 receive data number overrun processing function

Figure5.28 shows the flowchart for the UART0 receive data number overrun processing function.
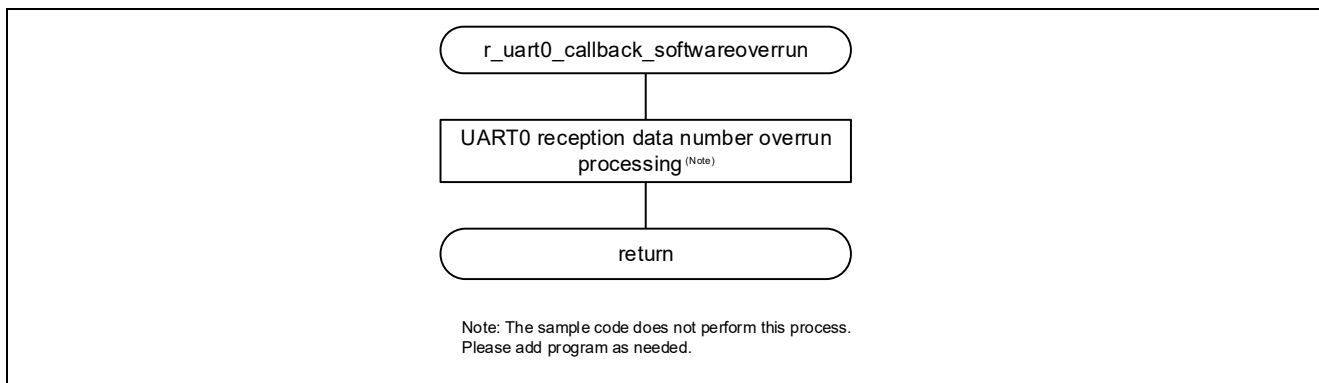
Figure 5.28   UART0 Receive Data Number Overrun Processing Function

5.9.23　UART0 reception complete processing

Figure5.29 shows the flowchart for UART0 reception complete processing.

```
        ┌──────────────────────────────┐
        │ r_uart0_callback_receiveend  │
        └──────────────────────────────┘
                      │
        ┌──────────────────────────────┐
        │ Clear UART0 reception error flag │      g_uart0rxerr ← 0
        └──────────────────────────────┘
                      │
        ┌──────────────────────────────┐
        │   Uart0 reception complete   │        g_uart0rxend ← 1
        └──────────────────────────────┘
                      │
        ┌──────────────────────────────┐        P3 register
        │       LED1 / LED2 OFF        │          P33, P31 bit ← 1B:Output 1
        └──────────────────────────────┘
                      │
        ┌──────────────────────────────┐
        │           return             │
        └──────────────────────────────┘
```
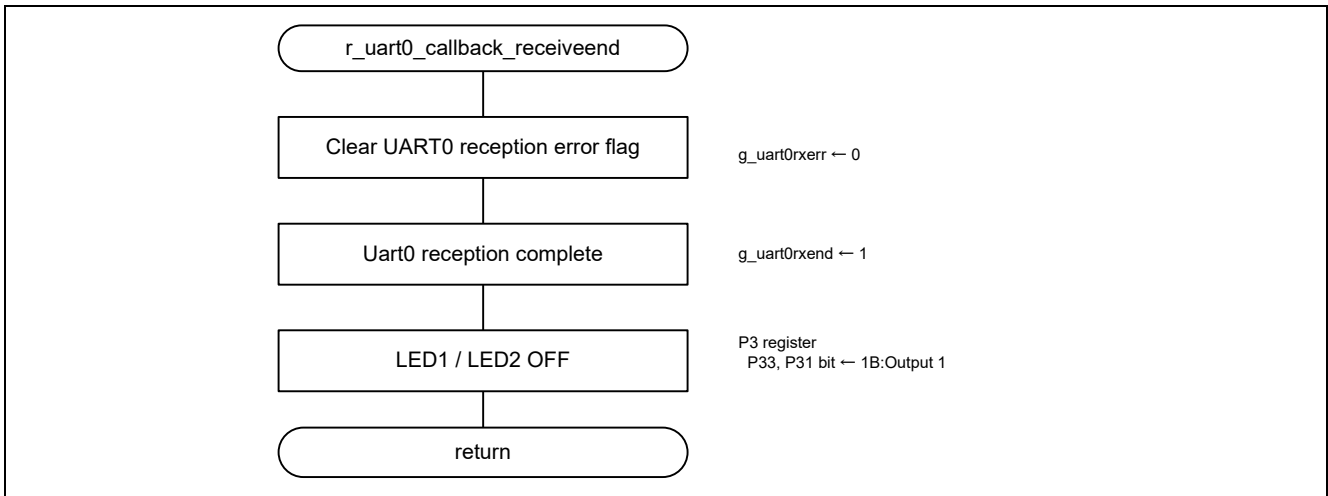
Figure 5.29　UART0 Reception Complete Processing

5.9.24　UART0 error interrupt function

Figure5.30 shows the flowchart for the UART0 error interrupt function.

```
        ┌──────────────────────────────┐
        │    r_uart0_interrupt_error    │
        └──────────────────────────────┘
                      │
        ┌──────────────────────────────┐
        │ Read reception data and set to variable │   *gp_uart0_rx_address ← RXD0
        └──────────────────────────────┘
                      │
        ┌──────────────────────────────┐
        │       Read error flag        │   err_type ← (uint8_t)(SSR01 & 0x0007U)
        └──────────────────────────────┘
                      │
        ┌──────────────────────────────┐
        │       Clear error flag       │   SIR01 ← (uint16_t)err_type
        └──────────────────────────────┘
                      │
       ┌┤──────────────────────────────┤┐
        │  UART0 reception error processing │
        │     r_uart0_callback_error()   │
       └┤──────────────────────────────┤┘
                      │
        ┌──────────────────────────────┐
        │           return             │
        └──────────────────────────────┘
```
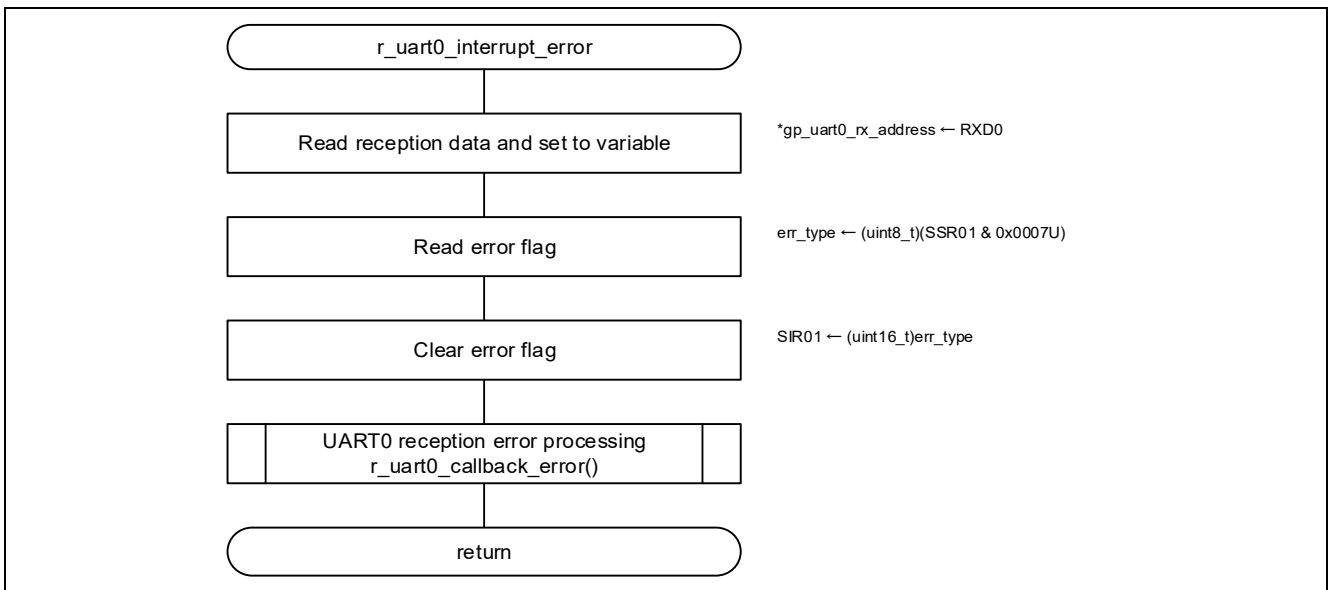
Figure 5.30　UART0 Error Interrupt Function

5.9.25   UART0 reception error processing

Figure5.31 shows the flowchart for UART0 reception error processing.
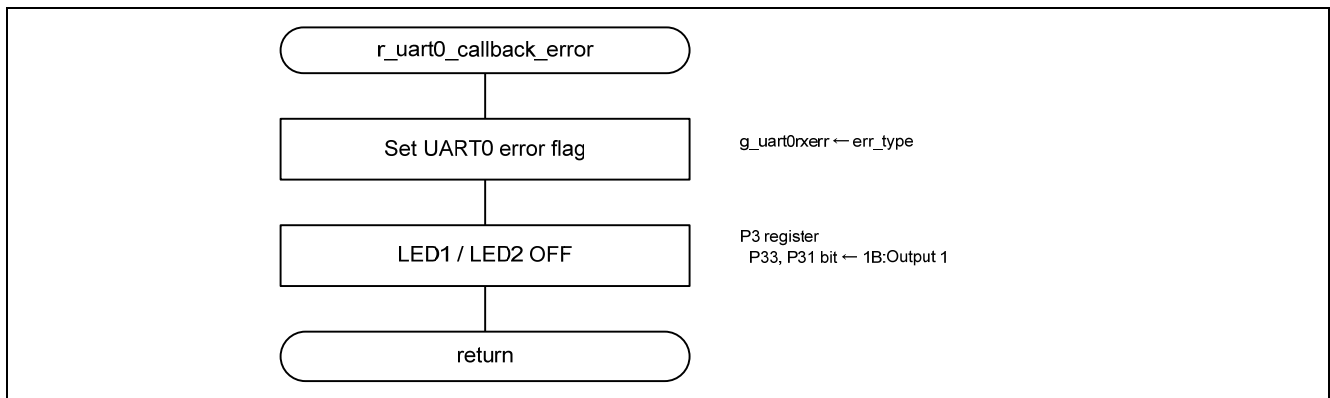
Figure 5.31   UART0 Reception Error Processing

6. Sample Code

Download the sample code from the Renesas Electronics homepage.

7.   Reference Documents

RL78/G11 User's Manual: Hardware (R01UH0637E)

RL78 family User's Manual: Software (R01US0015E)

(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	Feb. 15, 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 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

Renesas Electronics Corporation

http://www.renesas.com