RL78/I1D

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

Introduction

This application note explains how to receive data in UART communication during STOP mode using the RL78/I1D 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/I1D UART communication with middle speed on chip oscillator (R01AN2326EJ).”

Target Device

RL78/I1D

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/I1D 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 2 (INTP2) 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/I1D UART communication with middle speed on chip oscillator (R01AN2326EJ).”

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

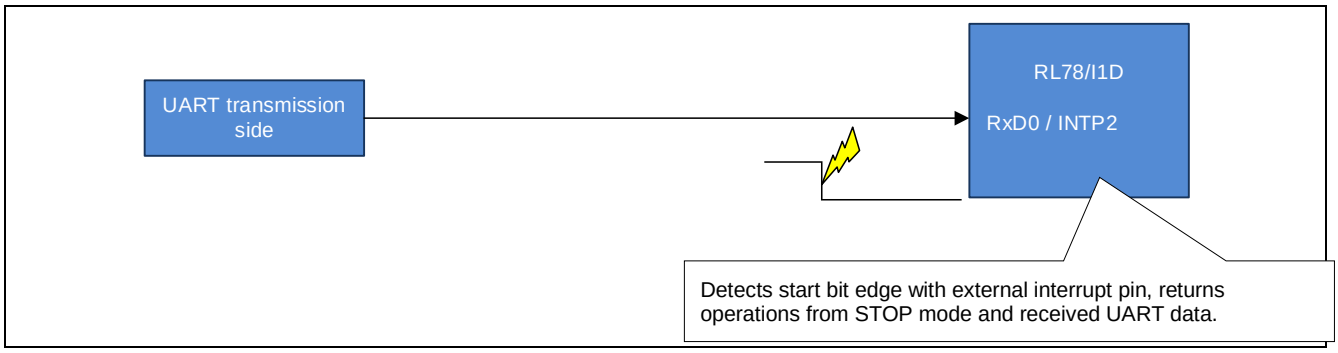

Figure 1.1 Application Outline

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/I1D (R5F117GCG)
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	3.3V (operating range1.6V to 3.6V) LVD operations (V_{LVI}): LVD off
Integrated development environment (CS+)	Made by Renesas Electronics Corp. CS+ V3.00.00
C compiler (CS+)	Made by Renesas Electronics Corp. CA78K0R V1.71
Integrated development environment (e2studio)	Made by Renesas Electronics Corp. e2studio V3.1.2.10
C compiler (e2studio)	Made by Renesas Electronics Corp. KPIT GNURL78-ELF Toolchain V14.0.3
Board	RL78/I1D CPU board (RTE5117GC0TGB00000R)

Notes: The RL78/I1D CPU board (RTE5117GC0TGB00000R) LEDs are connected to N-ch open drain ports (P60, P61). Setting P60 and P61 to Hi-Z to turn off the LEDs may cause through current to flow. Operations for this application note were evaluated assuming no through current. Therefore, actual measured supply current during STOP mode is 0.4[uA].

3. Related Application Notes

Applications notes related to this document are shown below.

- RL78/G13 Initialization [for CubeSuite+, IAR, and e2 studio] (R01AN0451EJ)
- RL78/G13 Serial Array Unit (UART Communication) (R01AN0459EJ)
- RL78/I1D UART communication with middle speed on chip oscillator (R01AN2326EJ)
- RL78/I1D UART communication with middle speed on chip oscillator (Baud rate correction with 32.768kHz sub-clock) (R01AN2584EJ)

4. Hardware Explanation

4.1 Hardware Structure Example

Figure 4.1shows the hardware used in this application note.

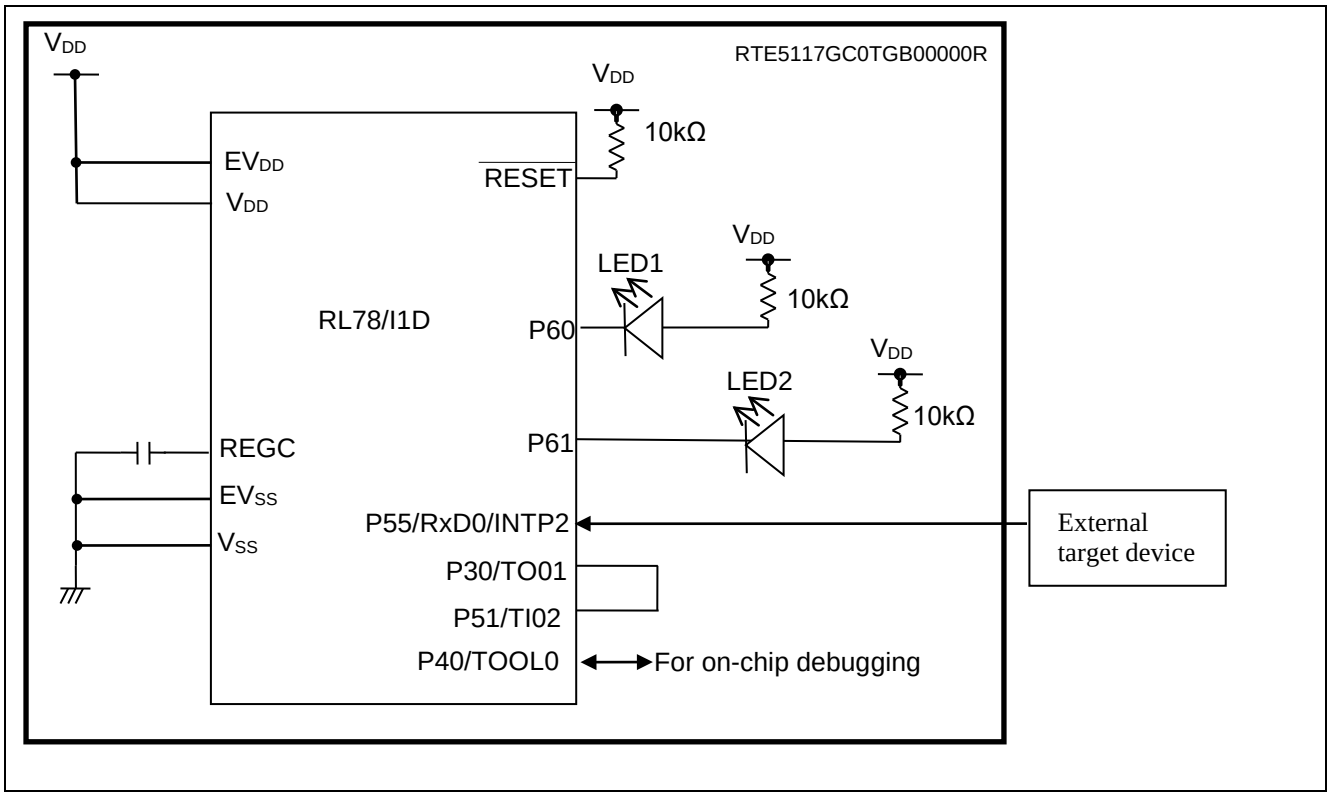

Figure 4.1 Hardware Configuration

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_{LVI}) set in LVD.

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

4.2 Pin List

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

Table 4.1 List of Pins and Functions

Pin Name	Input/Output	Function
P60	Output	LED1 control
P61	Output	LED2 control
P30/TO01	Output	Waveform for MOCO accuracy measurement
P51/TI02	Input	Waveform for MOCO accuracy measurement
P55/RxD0/INTP2	Input	RxD0: UART0 receive input pin INTP2: Interrupt edge detection

5. Software Explanation

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

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

- CS+

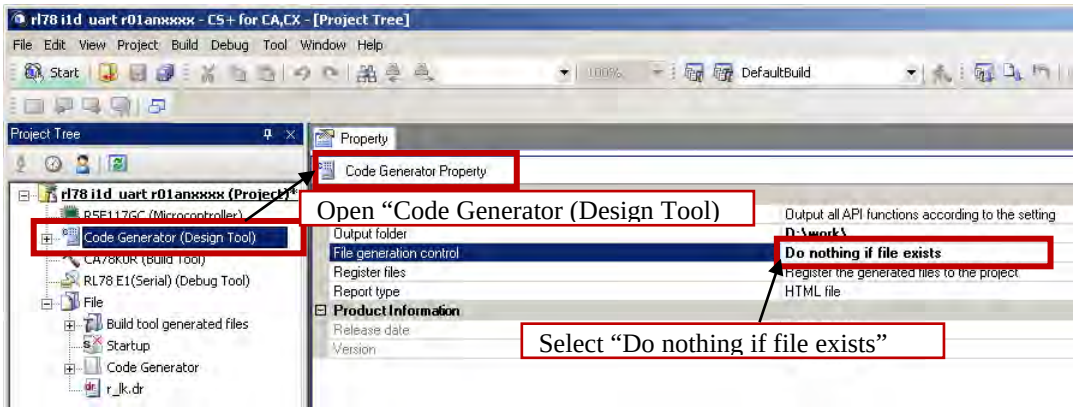

Figure 5.1 Code Generation Property Setting Screen (CS+)

- e2studio

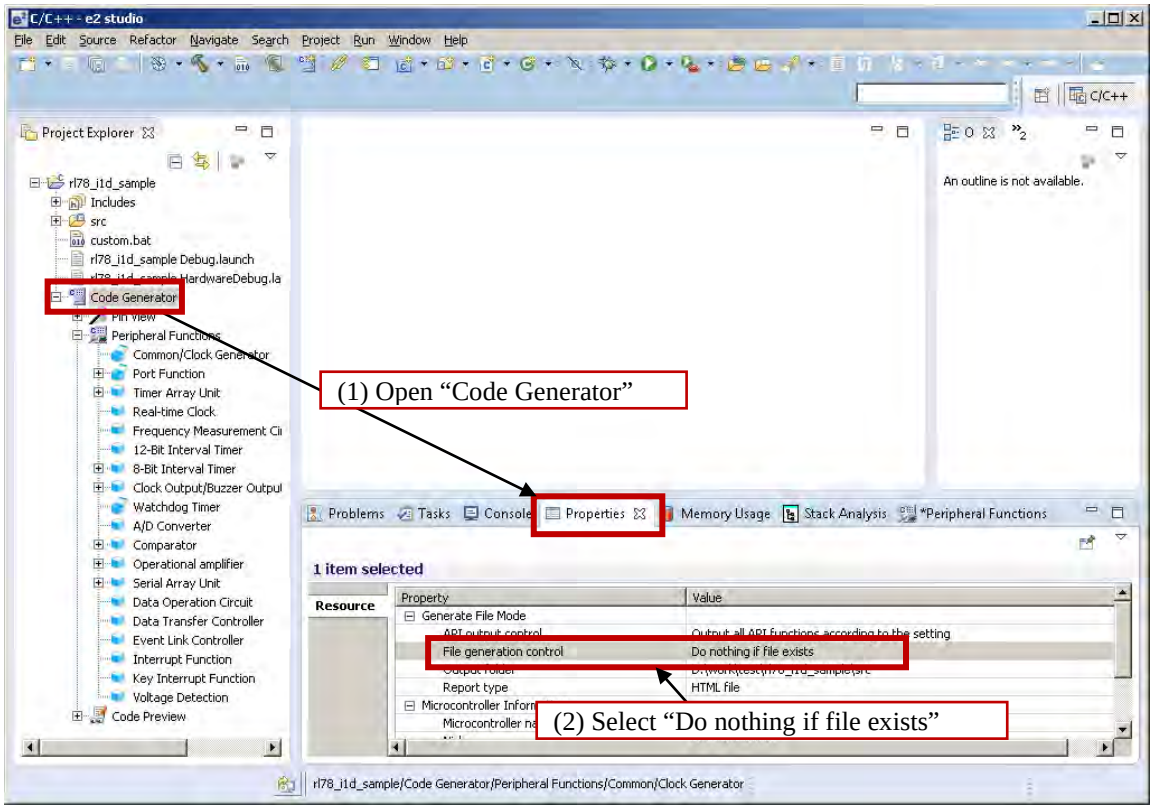

Figure 5.2 Code Generation Property Setting Screen (e2studio)

5.1 Operation Outline

This sample code 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/I1D UART communication with middle speed on chip oscillator (R01AN2326EJ)."

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 P60 and P61 to Hi-Z 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 2 setting condition>

- Operating clock: high-speed on-chip oscillator (HOCO) 24MHz
- Single channel operations
- TI02 pin valid edge: set to start trigger and capture trigger
- TI02 pin valid edge: rising edge
- Input pulse width measurement mode
- Positive logic output

(4) Interrupt initialization

<Setting conditions>

- Pin input edge detection (INTP2): rising edge

(5) main processing initialization

<Setting conditions>

- UART0 reception status initialization
  - Set variable md_status to “0” (OK).
  - Set variable g_uart0_rx_count to “0” (receive count value = 0).
  - Set variable g_uart0_rx_length to “1” (receive data number = 1).
  - Set variable gp_uart0_rx_address to receive data pointer.
- 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 2 starts operating.
    - Set IF1L register TMIF02 bit to “0” (interrupt request signal not generated).
    - Set MK1L register TMMK02 bit t to “0” (interrupt processing enabled).
    - Set TS0 register TS02 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 2 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 2
  - 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_ch2_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).
- INTP2 operation start
  - Set IF0L register PIF2 bit to “0” (interrupt request signal not generated).
  - Set MK0L register PMK2 bit to “0” (interrupt processing enabled).

(6) Shift to STOP mode

(7) Set the following and return from STOP mode after detecting INTP2 (pin input edge detection).

- Set MK0L register PMK2 bit to “1” (interrupt processing disabled).
- Set IF0L register PIF2 bit to “0” (interrupt request signal not generated).
- Set P60 and P61 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 P60 and P61 to Hi-Z output, turn off LED1 and LED2.

(9) When reception data is abnormal

- Framing error: set P60 to Low output/Hi-Z output, LED1 blinks.
- Parity error: set P61 to Low output/Hi-Z output, LED2 blinks.
- Overrun error: set P60 and P61 to Low output/Hi-Z output, LED1/LED2 both blink.
- Repeat step (9).

(10) UART0 reception status initialization

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

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

(12) INTP2 Operation start

- Set IF0L register PIF2 bit to “0” (interrupt request signal not generated).
- Set MK0L register PMK2 bit to “0” (interrupt processing enabled).

(13) Repeat steps (6) to (12).

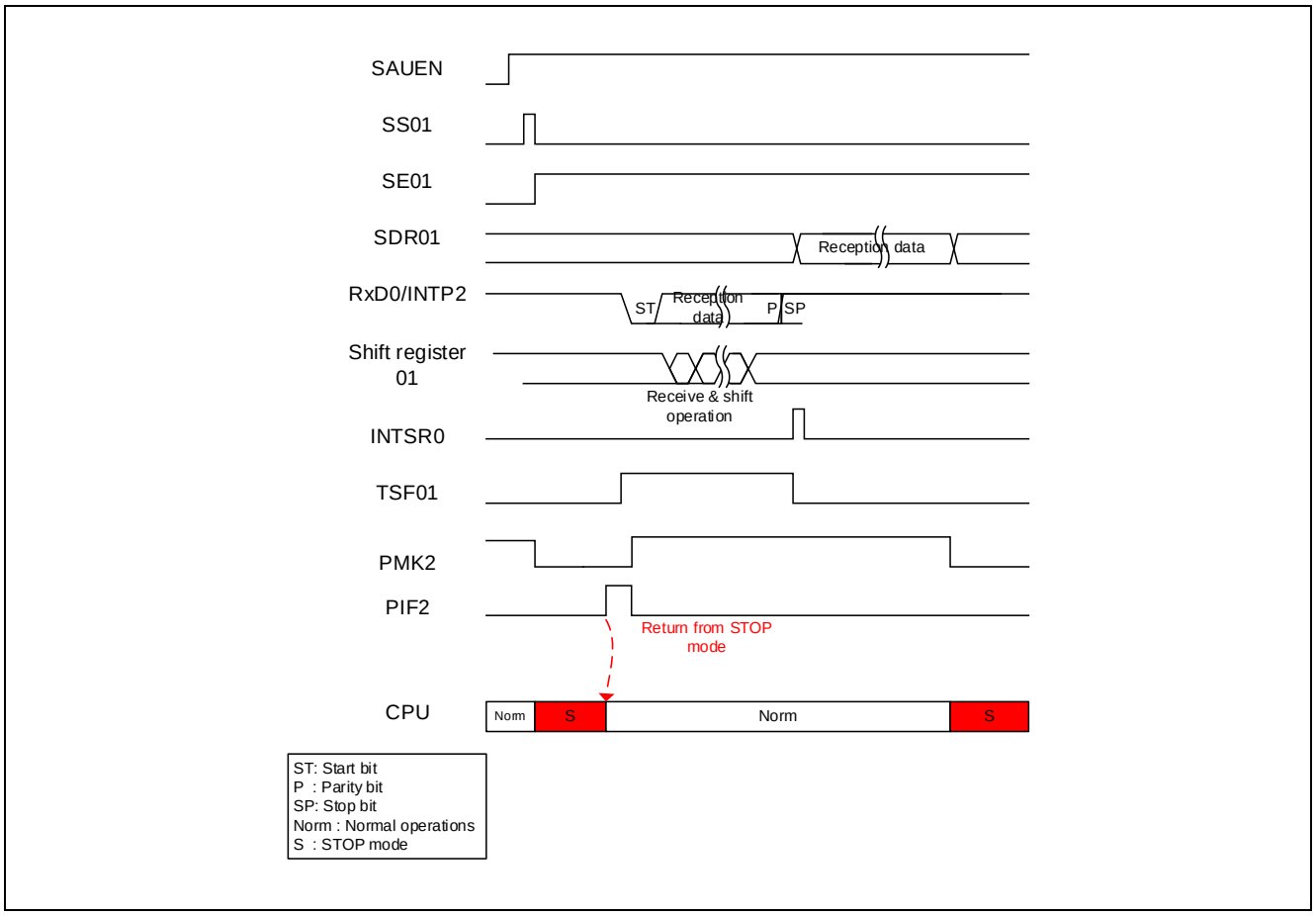

Figure 5.3 Timing Chart

5.2 Correction Processing

The following provides a detailed explanation of correction processing.

(1) Generating waveforms for measurement

As shown in Figure 5.4, 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 2 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%).

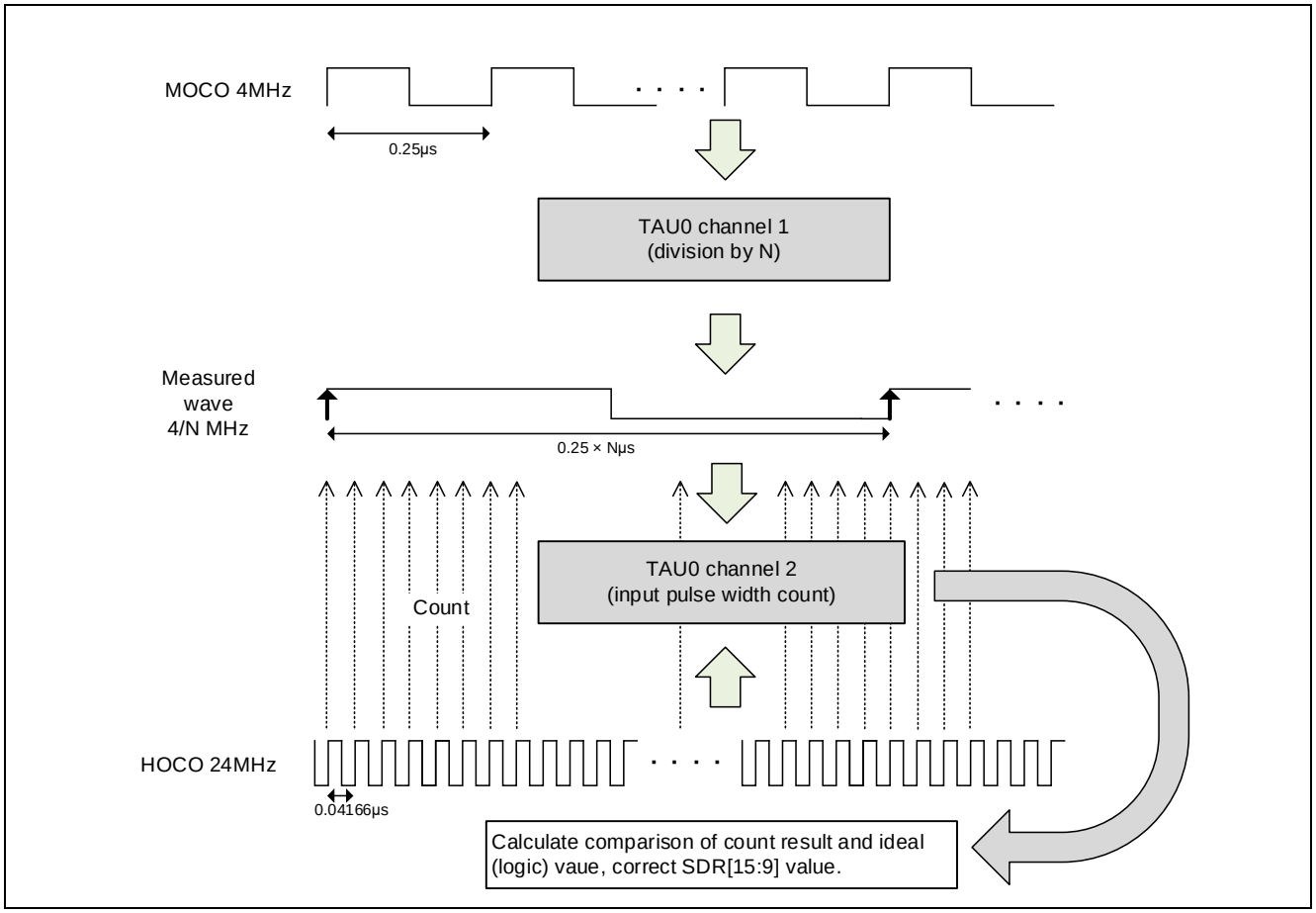

Figure 5.4 Correction Processing

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

∴ 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 * (*ideal count value / capture value of CH2*) / (2 * 9600)

= 2 MHz * (6.000N / *capture value of CH2*) / (2 * 9600)

= 2 MHz * (6 * 9516 / *capture value of CH2*) / (2 * 9600)

= 5947500 / (*capture value of CH2*)

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 / (*capture value of CH2* for the upper 8 bits)

5.3 Middle-speed On-chip Oscillator Temperature Characteristics

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

As the objective of this application note is to explain how to receive data in UART communications during STOP mode, baud rate correction can only be processed once after reset. For more details about baud rate correction, please refer to the following documents: UART communication with middle speed on chip oscillator (R01AN2326JJ) and UART communications with middle speed on chip oscillator (Baud rate correction with 36.768kHzsub-clock) (R01AN2584JJ))

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

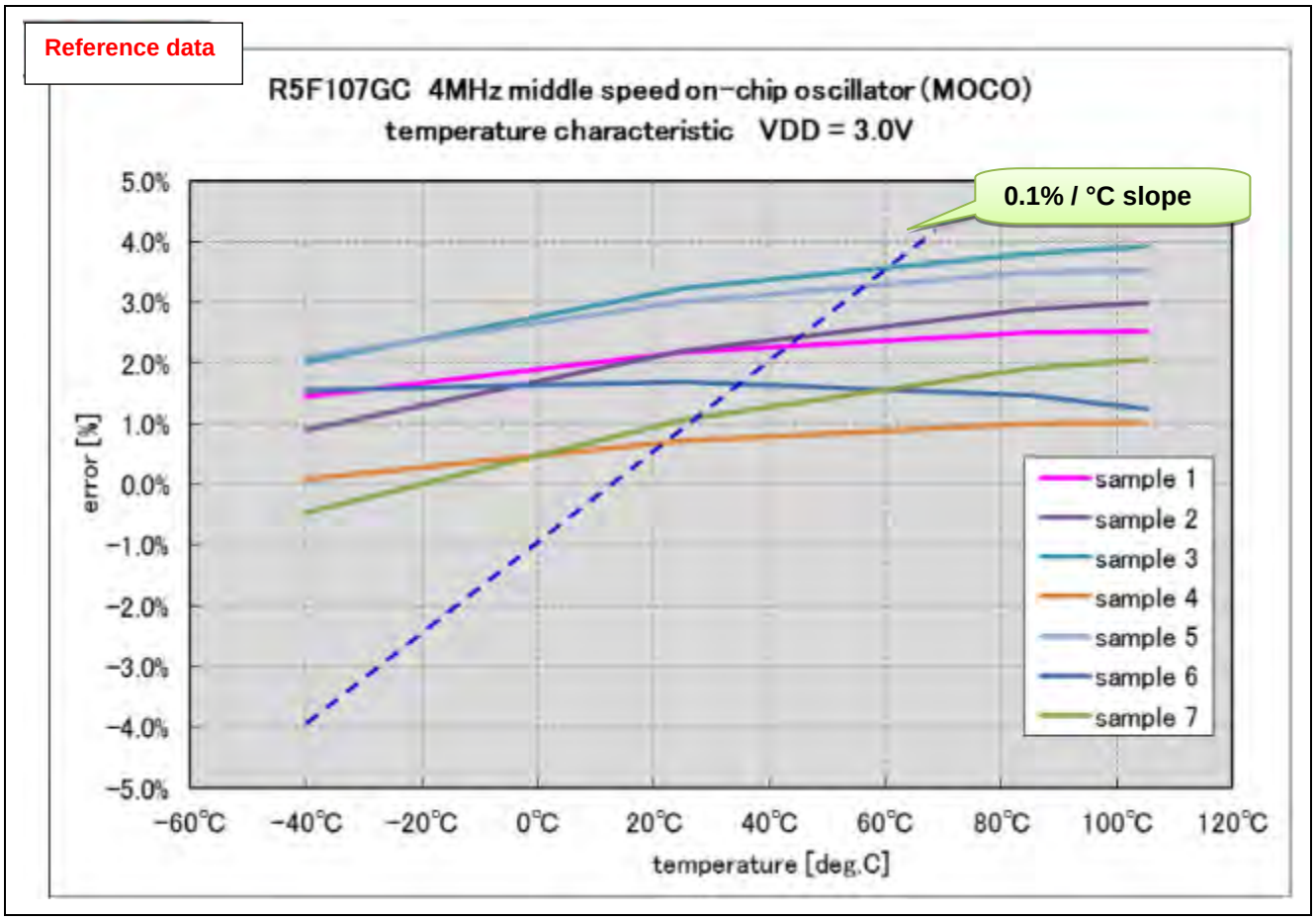

Figure 5.5 Middle-speed On-chip Oscillator Temperature Characteristics

5.4 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 5.6 shows the flow for enabling the UART receive operation in STOP mode.

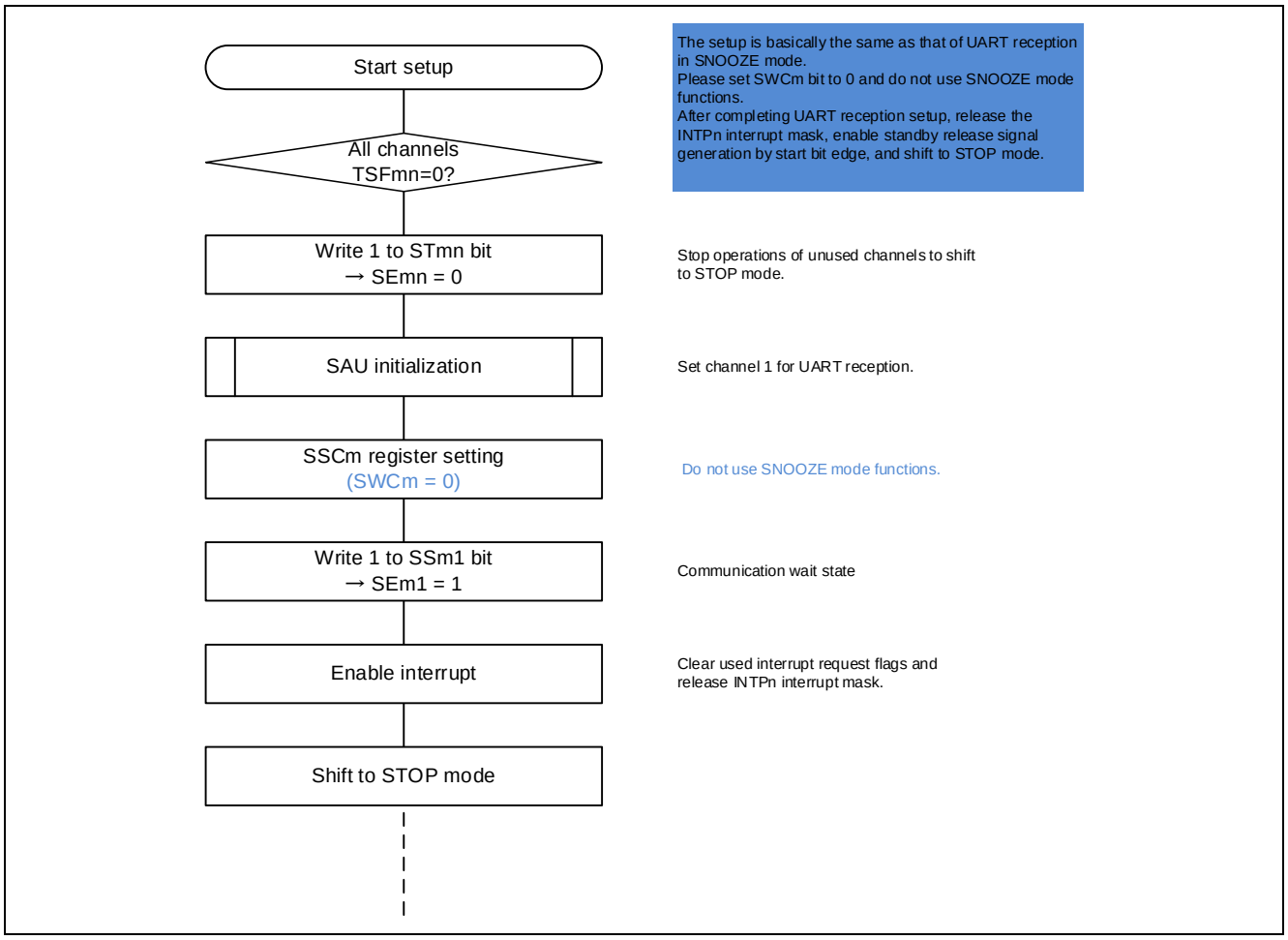

Figure 5.6 How to Receive UART Data in STOP Mode

5.5 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	1111 1111B	LVD off
000C2H/010C2H	1110 0000B	HS mode, HOCO: 24MHz
000C3H/010C3H	0000 0100B	On-chip debugging disabled

5.6 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

5.7 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 2	R_TAU0_Baudrate_Correction r_tau0_channel2_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_ch2_width	Measurement result of one square wave cycle	R_TAU0_Baudrate_Correction r_tau0_channel2_interrupt
uint32_t	g_wait_count	Wait counter for LED blinking	main R_Processing_pause

5.8 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_INTC2_Start	INTP2 operation start function
R_TAU0_Channel2_Start	TAU0 channel 2 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_Channel2_Stop	TAU0 channel 2 operation stop function
R_Processing_pause	Wait function for LED blinking
r_tau0_channel2_interrupt	TAU0 channel 2 count complete interrupt function
r_intc2_interrupt	INTP2 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

5.9 Function Specifications

The following shows function specifications for the sample code.

Function name: hdwinit

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

Function name: R_Systeminit

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

Function name: R_PORT_Create

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

Function name: R_CGC_Create

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

Function name: R_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
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	None
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 INTP2. 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	None
Declaration	void R_MAIN_UserInit(void)
Description	After UART0 and TAU0 are initialized, this function starts INTP2 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 2 are started. The pulse width of the square waveform generated by channel 1 is measured by channel 2. 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_INTC2_Start

Outline	INTP2 operation start function
Header	r_cg_intp.h
Declaration	void R_INTC2_Start (void)
Description	Sets INTP2 to operation enabled state.
Argument	None
Return value	None
Additional notes	None

Function name: R_TAU0_Channel2_Start

Outline	TAU0 channel 2 operation start function
Header	r_cg_tau.h
Declaration	void R_TAU0_Channel2_Start (void)
Description	Sets TAU0 channel 2 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_Channel2_Stop

Outline	TAU0 channel 2 operation stop function
Header	r_cg_tau.h
Declaration	void R_TAU0_Channel2_Stop (void)
Description	Sets TAU0 channel 12 to operation disabled state.
Argument	None
Return value	None
Additional notes	None

Function name: R_Processing_pause

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

Function name: r_tau0_channel2_interrupt

Outline	TAU0 channel 2 count complete interrupt function
Header	r_cg_tau.h
Declaration	__interrupt static void r_tau0_channel1_interrupt(void)
Description	Stores result of pulse width measured by TAU0 channel 2 in global variable.
Argument	None
Return value	None
Additional notes	None

Function name: r_intc2_interrupt

Outline	INTP2 interrupt processing function
Header	r_cg_intp.h
Declaration	__interrupt static void r_intc2_interrupt (void)
Description	Disable INTP2 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 static void r_uart0_interrupt_receive(void)
Description	Stores received data in RAM and updates address and number of reception data.
Argument	None
Return value	None
Additional notes	None

Function name: r_uart0_callback_softwareoverrun

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

Function name: r_uart0_callback_receiveend

Outline	UART0 reception complete processing function
Header	r_cg_sau.h
Declaration	static void r_uart0_callback_receiveend(void)
Description	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 static 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	

5.10 Flowcharts

Figure 5.7 shows the entire flow of the sample code.

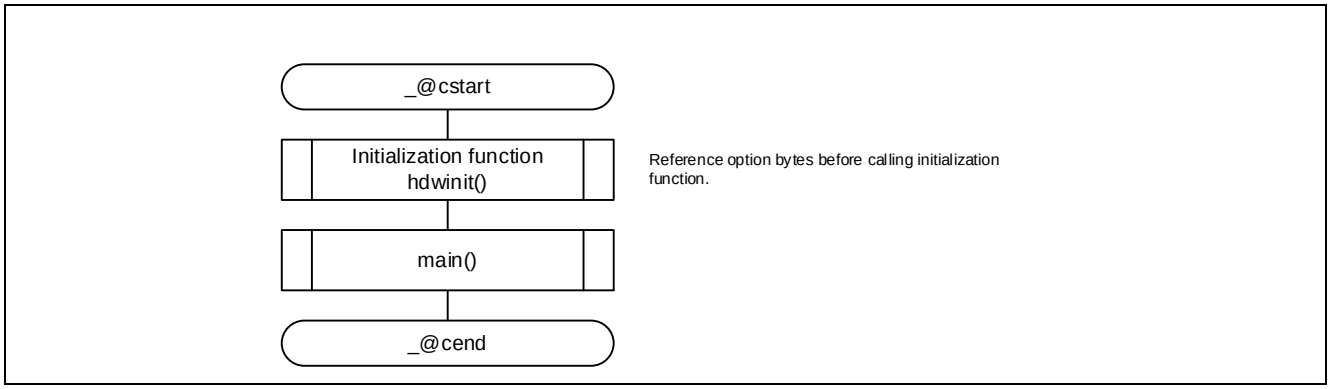

Figure 5.7 Entire Flow

5.10.1 Initialization

Figure 5.8 shows the flowchart of the initialization process.

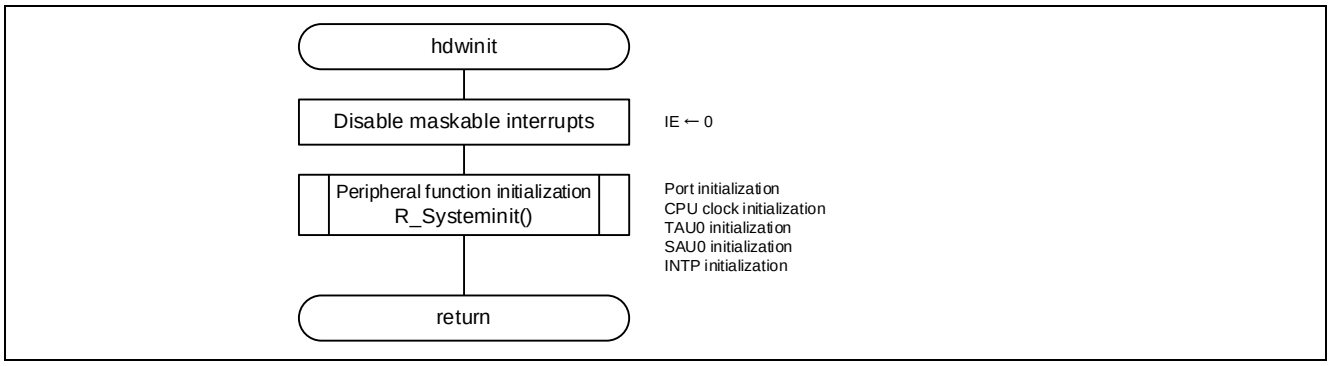

Figure 5.8 Initialization

5.10.2 Peripheral function initialization

Figure 5.9 shows the flowchart for the peripheral function initialization.

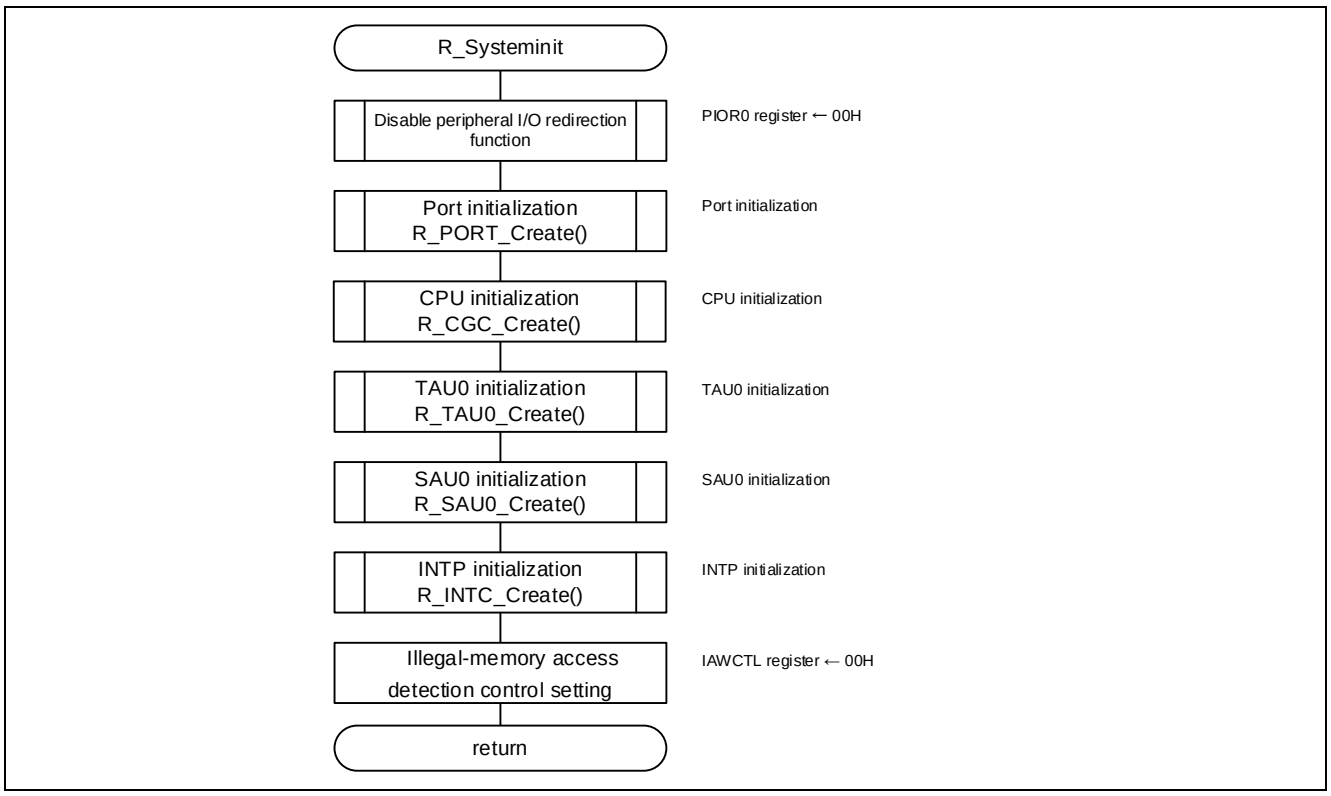

Figure 5.9 Peripheral Function Initialization

5.10.3 Port initialization

Figure 5.10 shows the flowchart for port initialization.

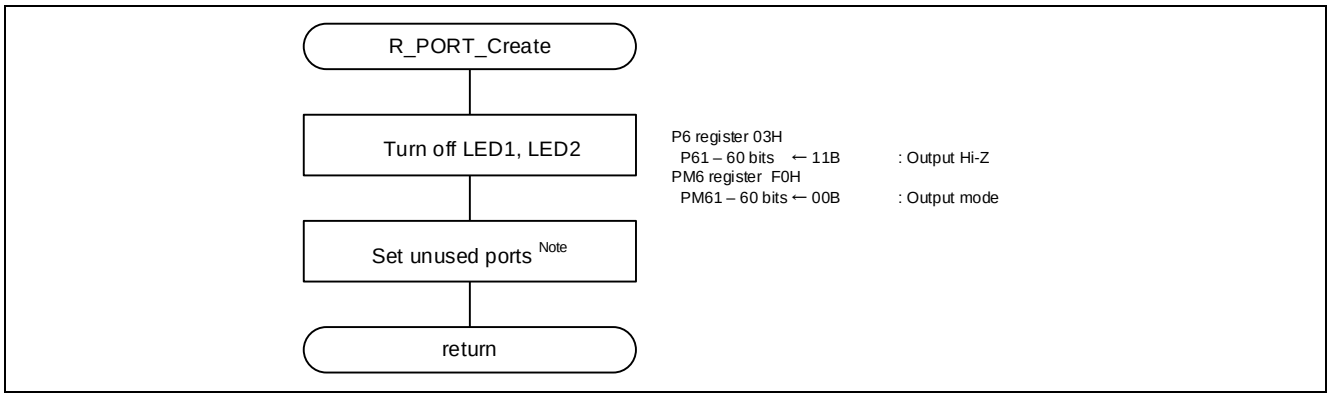

Figure 5.10 Port Initialization

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

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

5.10.4 CPU clock initialization

Figure 5.11 shows the flowchart for CPU clock initialization.

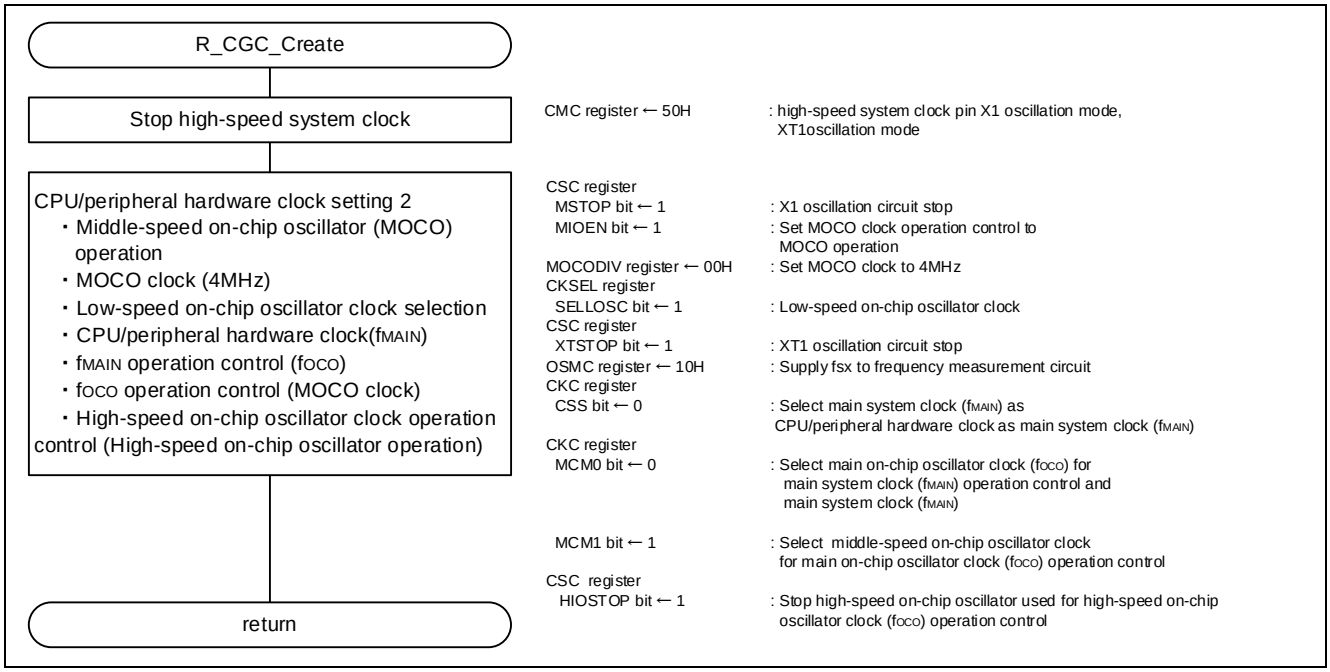

Figure 5.11 CPU Clock Initialization

5.10.5 TAU0 initialization

Figure 5.12 shows the flowchart for TAU0 initialization (1/2).

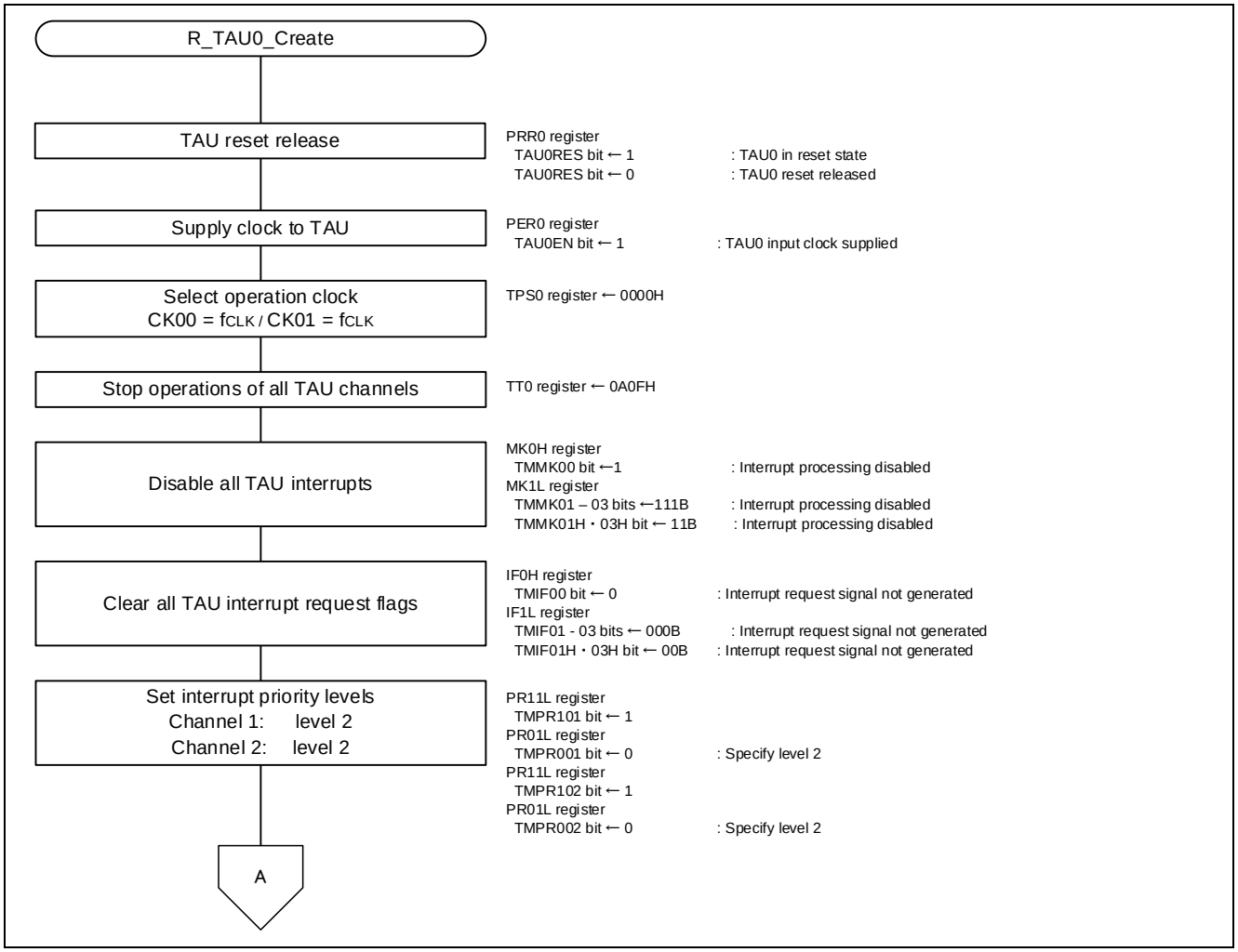

Figure 5.12 TAU0 Initialization (1/2)

Figure 5.13 shows the flowchart for TAU0 initialization (2/2).

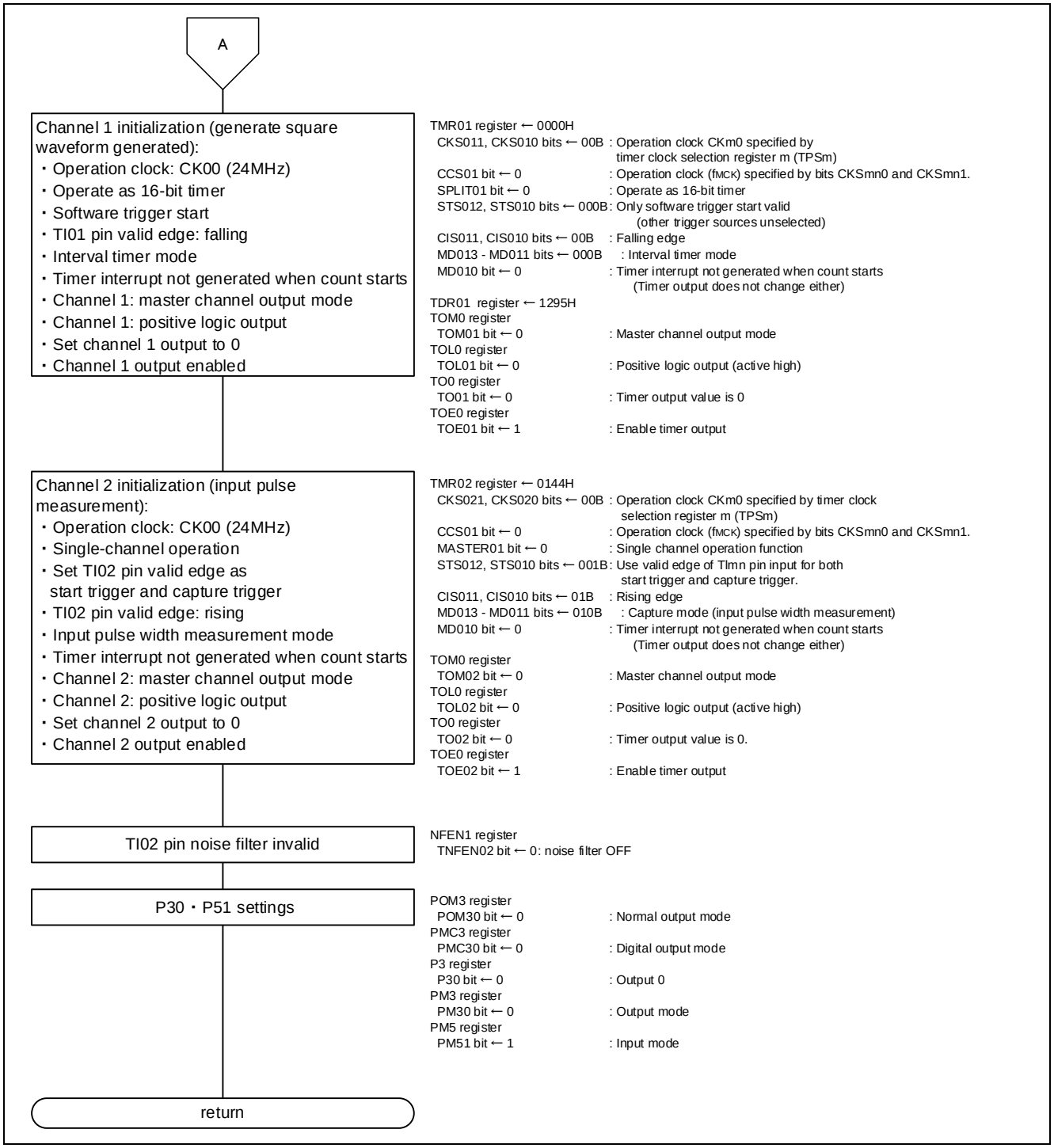

Figure 5.13 TAU0 Initialization (2/2)

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	0	ADCRES	0	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	0	ADCEN	0	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	TTH03	0	TTH01	0	0	0	0	0	TT03	TT02	TT01	TT00
x	x	x	x	**1**	x	**1**	x	x	x	x	x	**1**	**1**	**1**	**1**

Bit 11

TTH03	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

TTH01	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)

TT0n	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
RTITMK	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
0	0	TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	FMMK
x	x	**1**	**1**	**1**	**1**	**1**	x

Bit 5 - 1 (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
RTITIF	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
0	0	TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	FMIF
x	x	**0**	**0**	**0**	**0**	**0**	x

Bit 5 - 1 (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 2 to level 2.

Symbol: PR11L

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

Symbol: PR01L

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

Bit 4 - 3 (n = 1 - 2)

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	TO03	TO02	TO01	TO00.
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**0**	x

Bit 1

TO01	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	TOE03	TOE02	TOE01	TOE00.
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x

Bit 1

TOE01	Enable/Disable timer output of channel n
0	Disables timer output.
1	**Enables timer output**

TAU0 Channel 2 Initialization

- Timer mode register 02 (TMR02)
  Set TAU0 channel 2 as follows:
  - Operation clock: CK00(24MHz)
  - Single channel operations
  - TI02 pin valid edge used as start trigger and capture trigger
  - TI02 pin valid edge: rising
  - Input pulse width measumrent mode
  - Timer interrupt not generated when count star count starts

Symbol: TMR02

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS 021	CKS 020	0	CCS 02	MASTER 02	STS 022	STS 021	STS 020	CIS 021	CIS 020	0	0	MD 023	MD 022	MD 021	MD 020
0	**0**	x	**0**	**0**	**0**	**0**	**1**	**0**	**1**	x	x	**0**	**1**	**0**	**0**

Bit 15 – 14

CKS 021	CKS 020	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

CCS02	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

MASTER 02	Selection of single -channel operation/multi-channel operation (slave/master) for channel n
0	**Single-channel operation function**, or use slave channel with simultaneous channel operation function
1	Use master channel with simultaneous channel operation function

Bit 10 – 8

STS 022	**STS 021**	**STS 020**	**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 021	CIS 020	Selection of Tlmn 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 023	MD 022	MD 021	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 020	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 2 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	TOM03	TOM02	TOM01	0
x	x	x	x	x	x	x	x	x	x	x	x	x	**0**	x	x

Bit 2

TOM02	Control of channel n timer output mode
0	**Master channel output mode**
1	Slave channel output mode

TAU0 Channel 2 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	TOL03	TOL02	TOL01	0
x	x	x	x	x	x	x	x	x	x	x	x	x	**0**	x	x

Bit 2

TOL02	Control of channel n timer output level
0	**Positive logic output (active-high)**
1	Negative logic output (active-low)

TAU0 Channel 2 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	TO03	TO02	TO01	TO00.
x	x	x	x	x	x	x	x	x	x	x	x	x	**0**	x	x

Bit 2

TO02	Timer output of channel n
0	**Timer output value is 0.**
1	Timer output value is 1.

TAU0 Channel 2 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	TOE03	TOE02	TOE01	TOE00.
x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x

Bit 2

TOE02	Enable/Disable timer output of channel n
0	Disables timer output
1	**Enables timer output**

TAU0 TI02 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	x	**0**	x	x

Bit 2

TNFEN02	Enable/disable use of noise filter for TI02 pin input signal
0	**Noise filter OFF**
1	Noise filter ON

TAU0 P30/P51 Settings

- Port output mode register (POM3)
  Set to normal mode.
- Port mode control register (PMC3)
  Set to digital input/output.
- Port register (P3)
  Set to low level.
- Port mode register (PM3/PM5)
  Set PM30 to output mode and PM51 to input mode

Symbol: POM3

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	POM30
x	x	x	x	x	x	x	**0**

Bit 0

POM30	Selection of Pmn pin output mode (m = 3, 5; n = 0-6)
0	**Normal output mode**
1	N-ch open drain output (V_{DD} withstand voltage)

Symbol: PMC3

7	6	5	4	3	2	1	0
1	1	1	1	1	1	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: 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 = 0-6, 12, 13; n = 0-7)
0	**Output mode (functions as output port (output buffer ON))**
1	Input mode (functions as input port (output buffer OFF))

Symbol: PM5

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

Bit 1

PM51	Selection of Pmn pin input/output mode m = 0-6, 12, 13; n = 0-7)
0	Output mode (functions as output port (output buffer ON))
1	**Input mode (functions as input port (output buffer OFF))**

5.10.6 SAU0 initialization

Figure 5.14 shows the flowchart for SAU0 initialization.

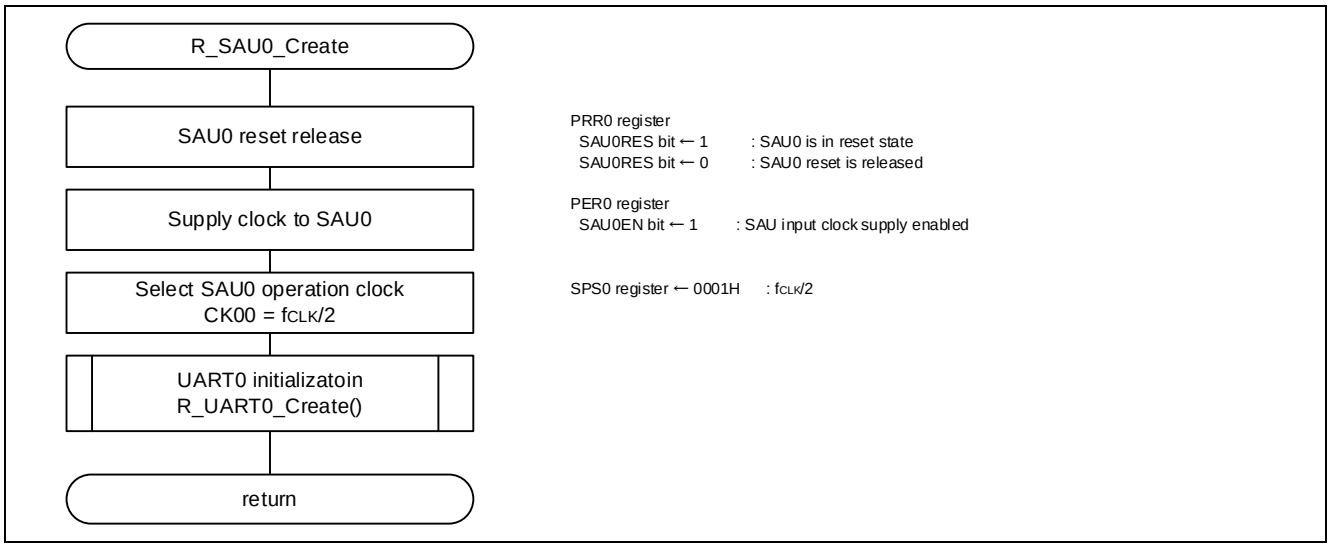

Figure 5.14 SAU0 Initialization

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	0	ADCRES	0	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	0	ADCEN	0	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

5.10.7 UART0 initialization

Figure 5.15 shows the flowchart for UART0 initialization.

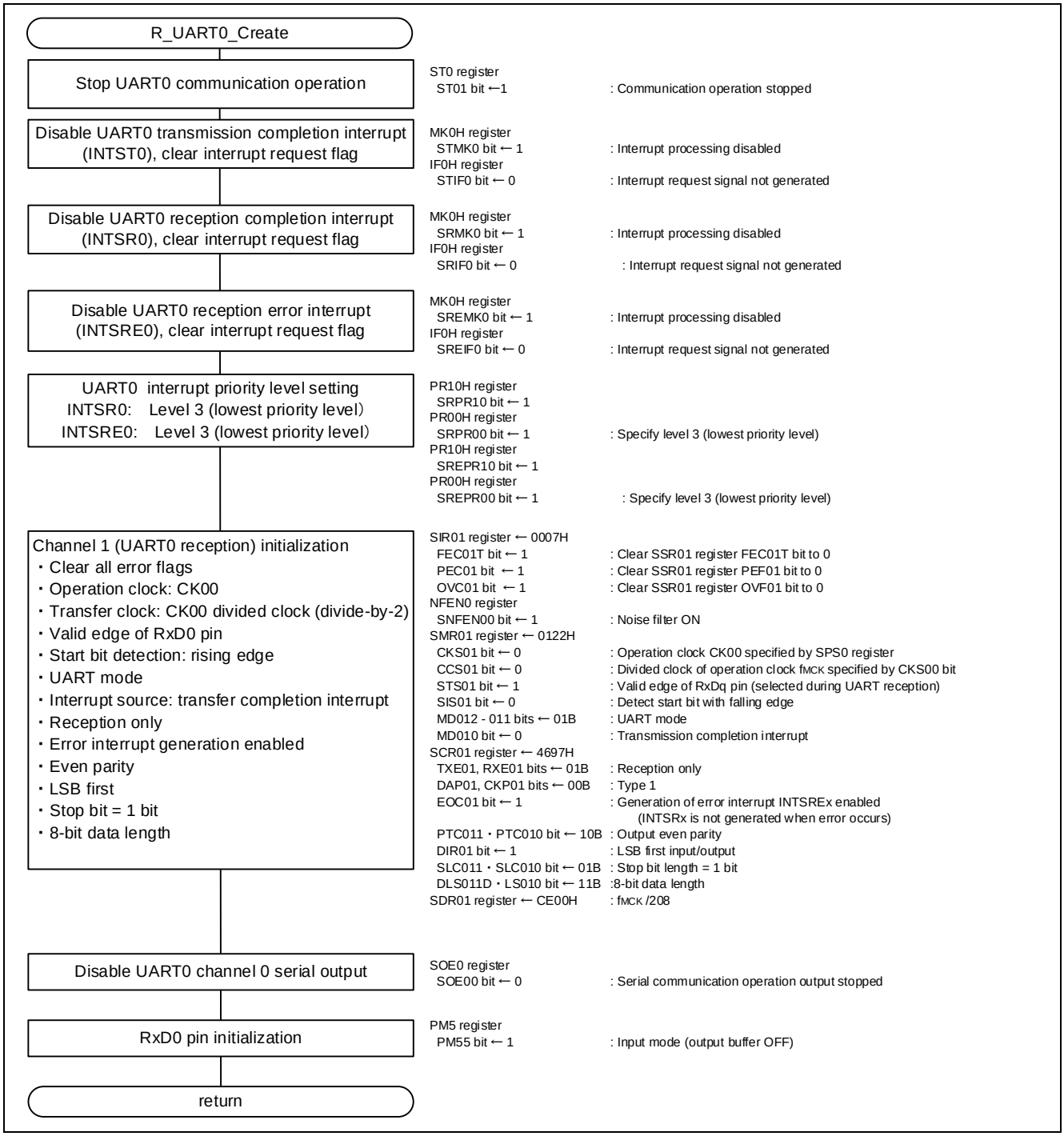

Figure 5.15 UART0 Initialization

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
RTITMK	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
RTITIF	TMIF00	SREIF0	0	0	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	PIF6
x	x	**0**	x	x	**0**	**0**	x

Bit 5

SREIF0	Interrupt request flag
0	**Interrupt request signal 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
RTITPR0	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
RTITPR1	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	FECT01	PECT01	OVCT01
—	—	—	—	—	—	—	—	—	—	—	—	—	**1**	**1**	**1**

Bit 2

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

Bit 1

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

Bit 0

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

UART0 channel 1 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	0	0	SNFEN00
x	x	x	x	x	x	x	**1**

Bit 0

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

UART0 Channel 1 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^2C mode
1	1	Setting prohibited

Bit 0

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

UART0 Channel 1 Serial Communication Operation Settings

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

Symbol: SCR01

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE01	RXE01	DAP01	CKP01	0	EOC01	PTC011	PTC010	DIR01	0	SLC011	SLC010	0	1	DLS011	DLS010
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

PTC011	**PTC010**	**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
PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50
x	x	**1**	x	x	x	x	x

Bit 5

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

5.10.8 INTP initialization

Figure 5.16 shows the INTP initialization flowchart.

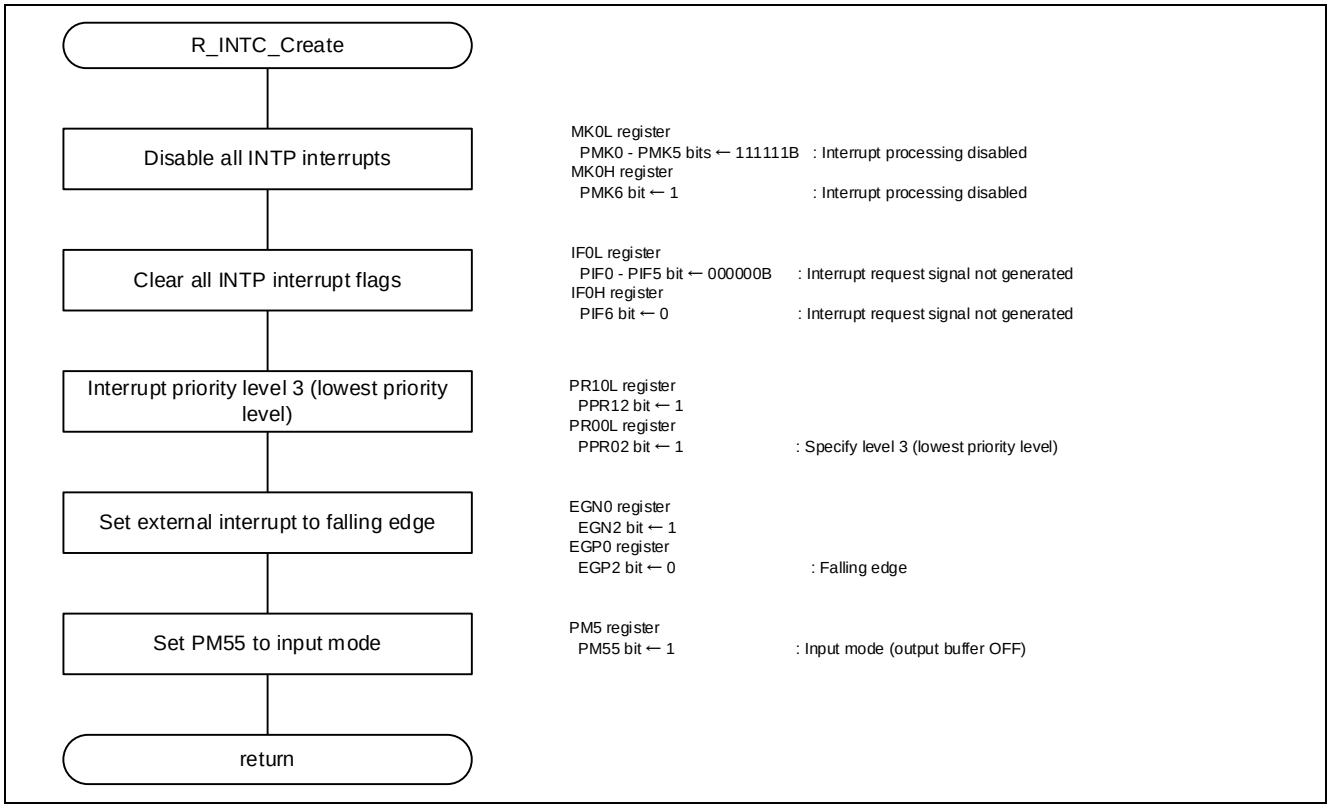

Figure 5.16 INTP Initialization

INTP All Interrupts Disable

- Interrupt request flag register (MK0L/MK0H)
  Disable all INTP interrupts.
- Interrupt request flag register (IF0L/IIF0H)
  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: MK0H

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

Bit 0

PMK6	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: IF0H

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

Bit 0

PIF6	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 (PR10L, PR00L)
Set interrupt to level 3 (lowest priority level)

Symbol: PR10L

7	6	5	4	3	2	1	0
PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1
x	x	x	**1**	x	x	x	x

Symbol: PR00L

7	6	5	4	3	2	1	0
PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0
x	x	x	**1**	x	x	x	x

Bit 4

PPR02	PPR12	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 (EGN0)
External interrupt rising edge enable register (EGP0)
Set to rising/falling edges.

Symbol: EGN0

7	6	5	4	3	2	1	0
0	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0
x	x	x	x	x	**1**	x	x

Symbol: EGP0

7	6	5	4	3	2	1	0
0	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0
x	x	x	x	x	**0**	x	x

Bit 2

EGP2	EGN2	INTPn pin valid edge selection (n = 0-6)
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
PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50
x	x	**1**	x	x	x	x	x

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)**

5.10.9 Main processing

Figure 5.17 shows the flowchart for main processing (1/2).

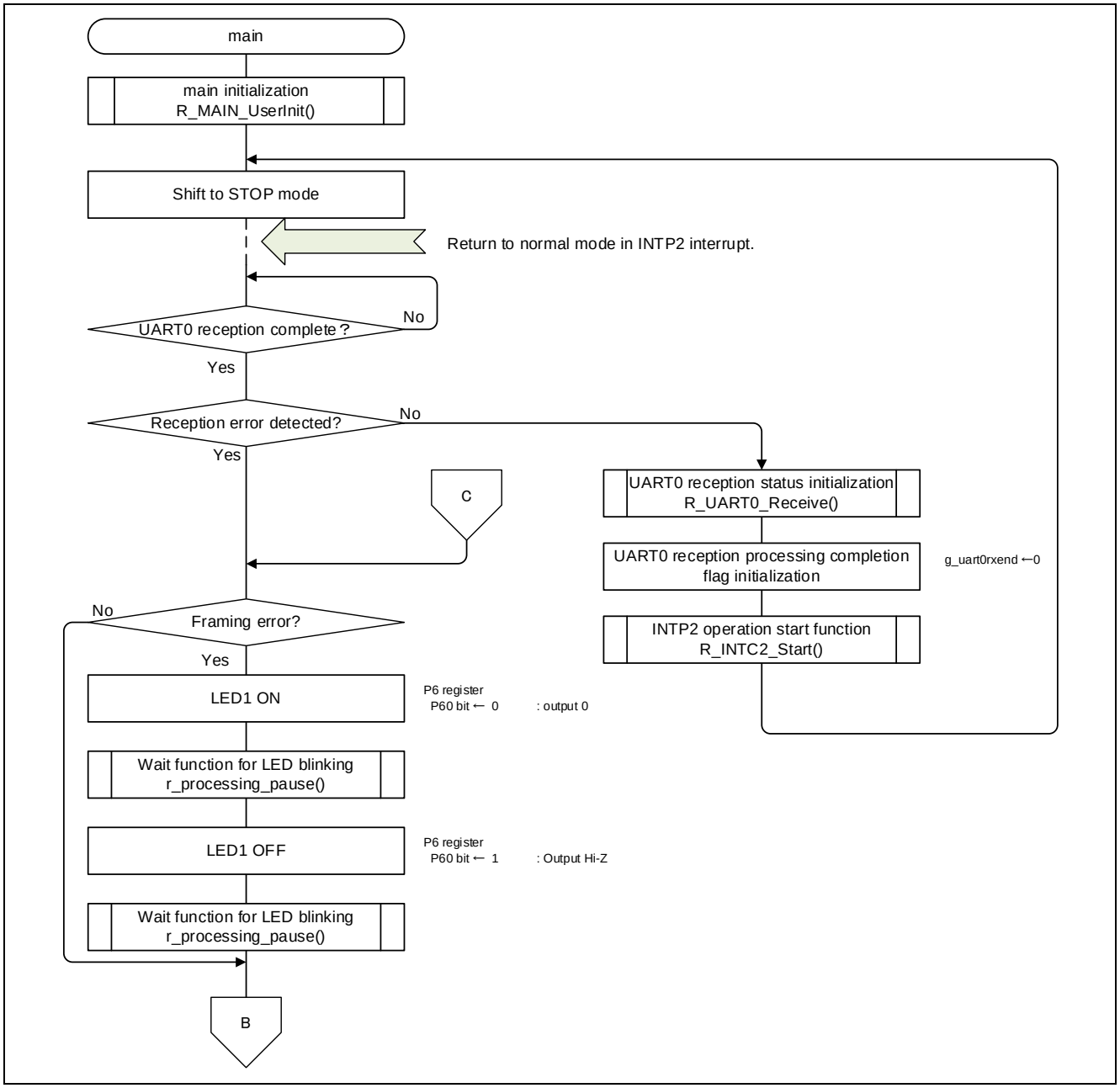

Figure 5.17 Main Processing (1/2)

Figure 5.18 shows the flowchart for main processing (2/2).

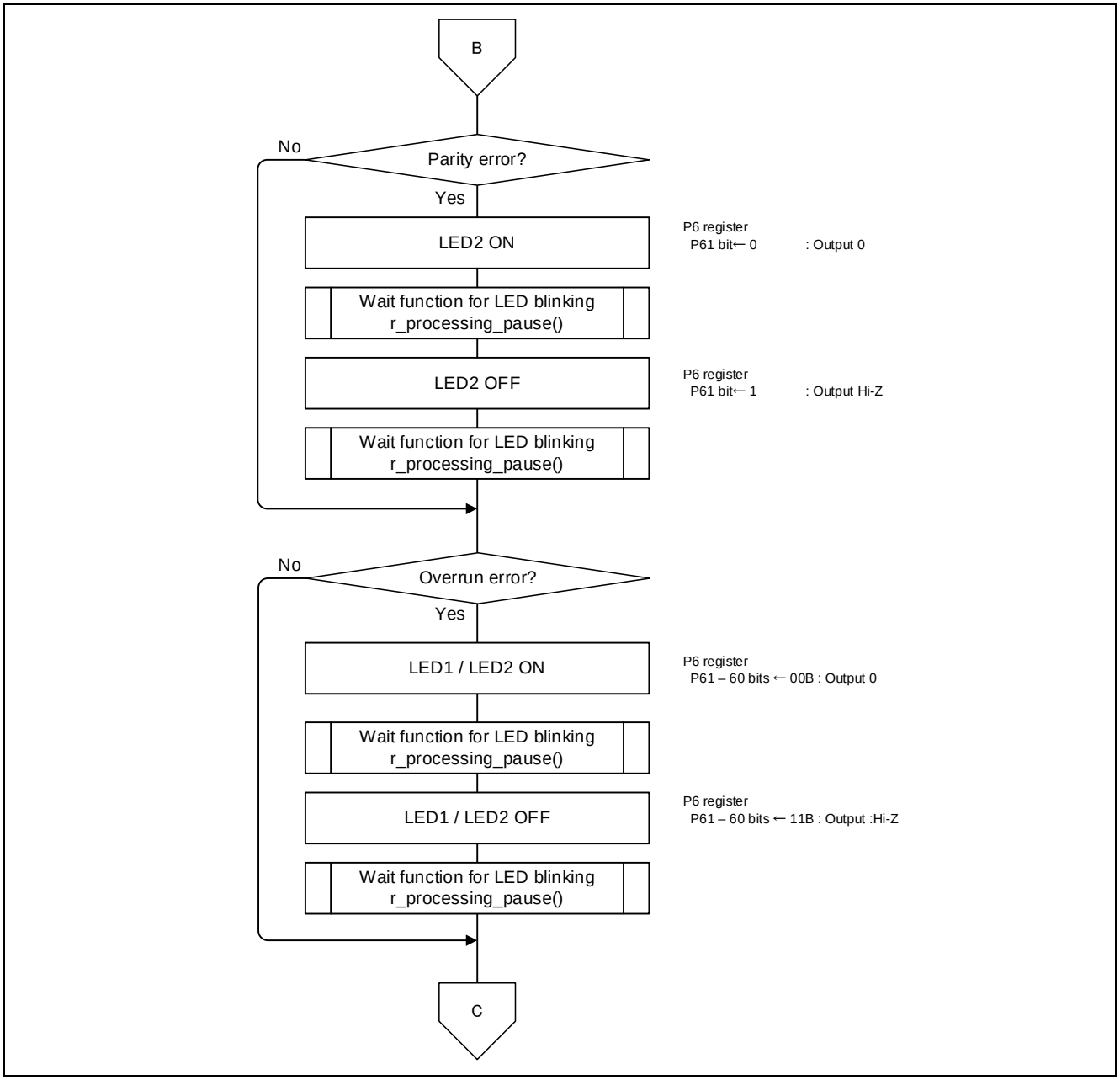

Figure 5.18 Main Processing (2/2)

5.10.10 Main initialization

Figure 5.19 shows the flowchart for main initialization.

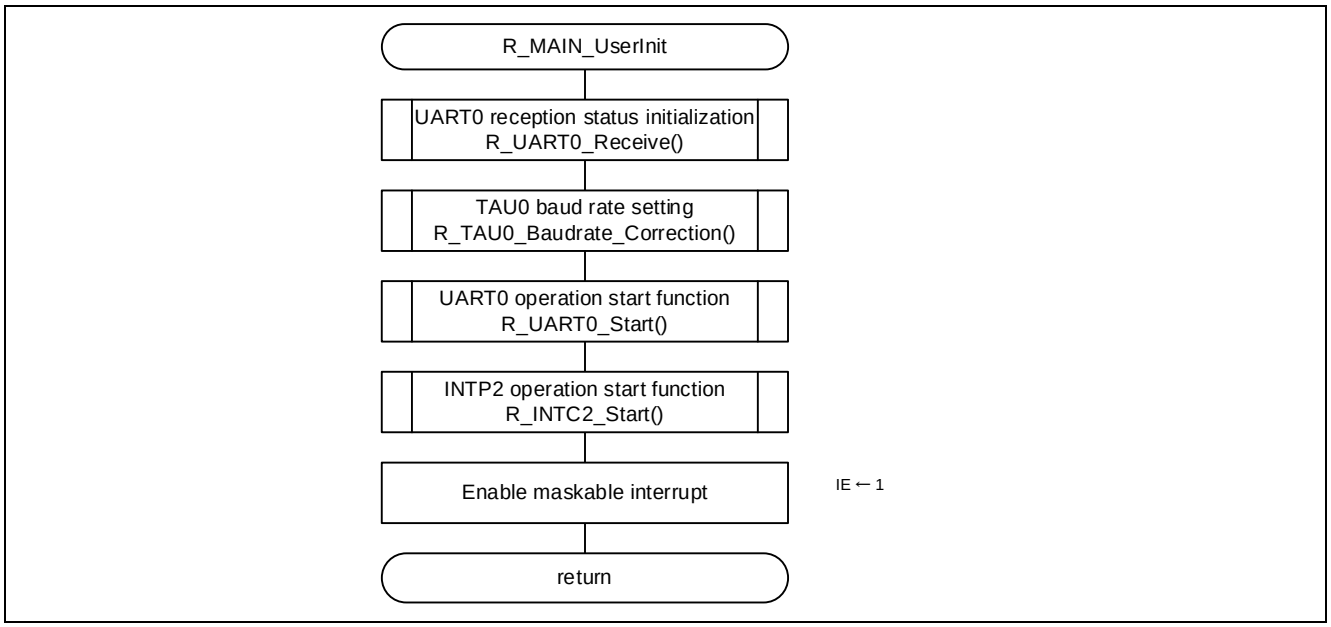

Figure 5.19 Main Initialization

5.10.11 UART0 reception status initialization

Figure 5.20 shows the flowchart for UART0 reception status initialization.

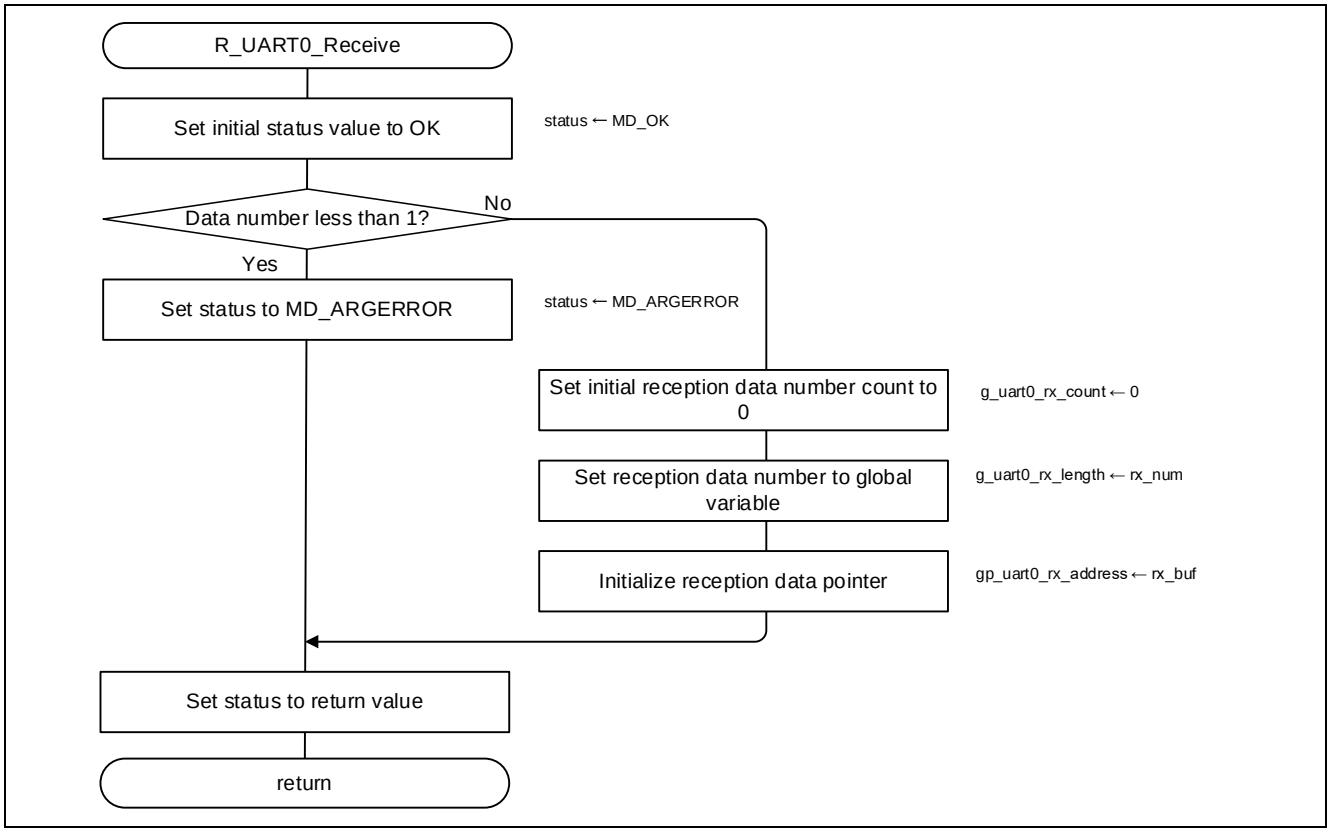

Figure 5.20 UART0 Reception Status Initialization

5.10.12 UART0 operation start function

Figure 5.21 shows the flowchart for the UART0 operation start function.

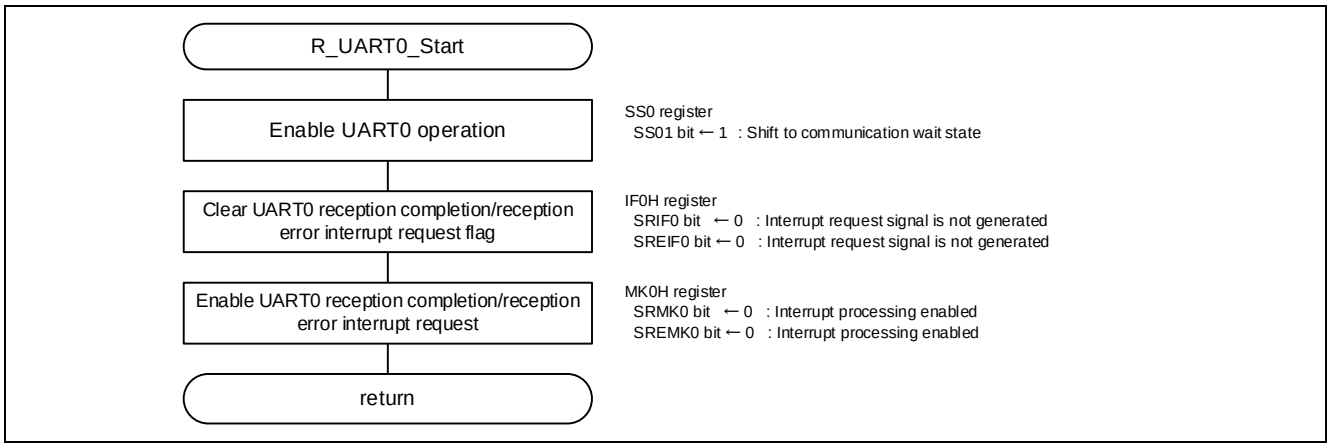

Figure 5.21 UART0 Operation Start Function

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

Bit 1

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

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
RTITIF	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
RTITMK	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

5.10.13 TAU0 baud rate setting

Figure 5.22 shows the flowchart for TAU0 baud rate setting (1/2).

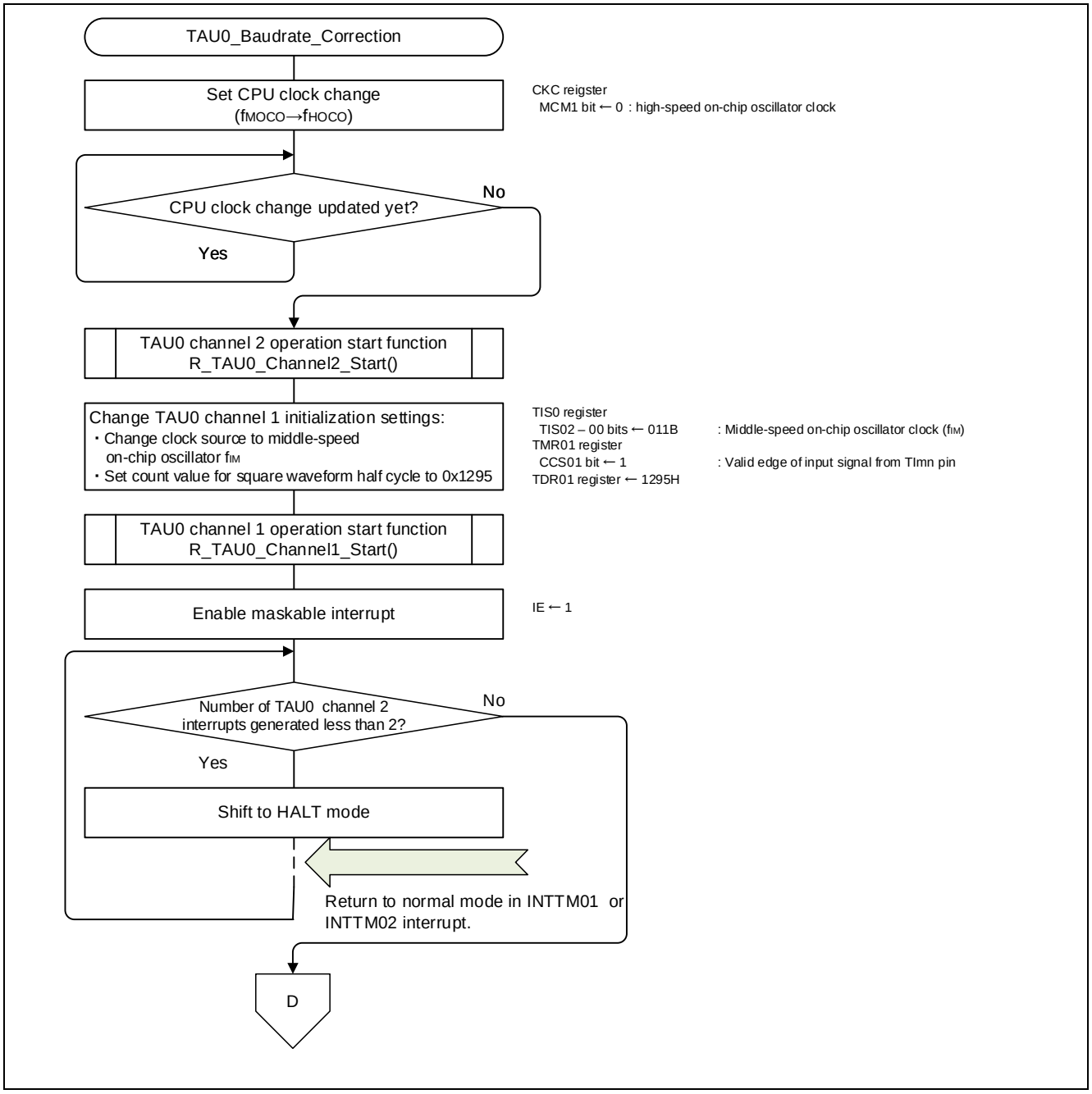

Figure 5.22 TAU0 Baud Rate Setting (1/2)

Figure 5.23 shows the flowchart for TAU0 baud rate setting (2/2).

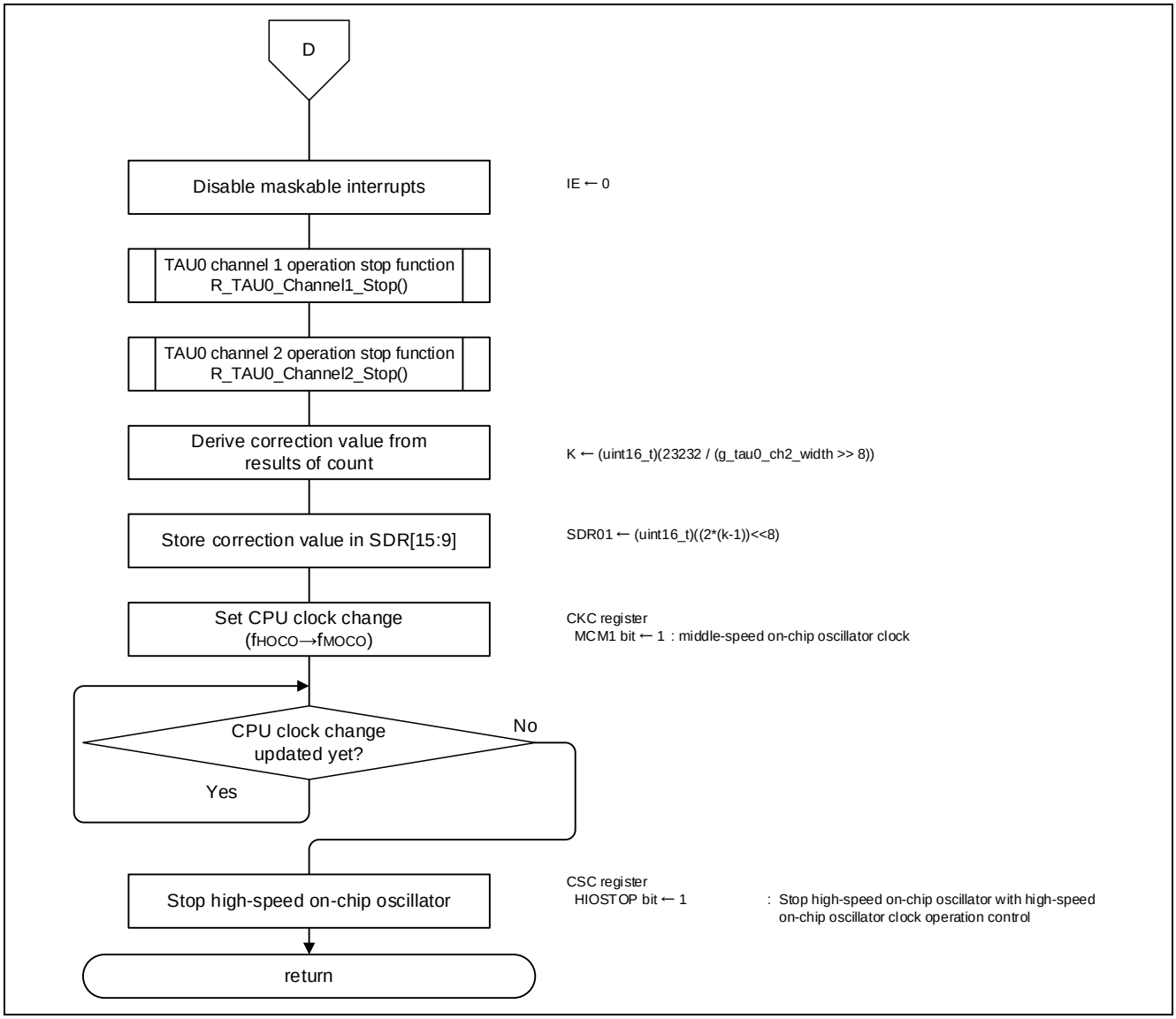

Figure 5.23 TAU0 Baud Rate Setting (2/2)

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	0	0	TIS04	0	TIS02	TIS01	TIS00
x	x	x	x	x	**0**	**1**	**1**

Bit 2 - 1

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 1296H.

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

5.10.14 INTP2 operation start function

Figure 5.24 shows the flowchart for INTP2 operation start function.

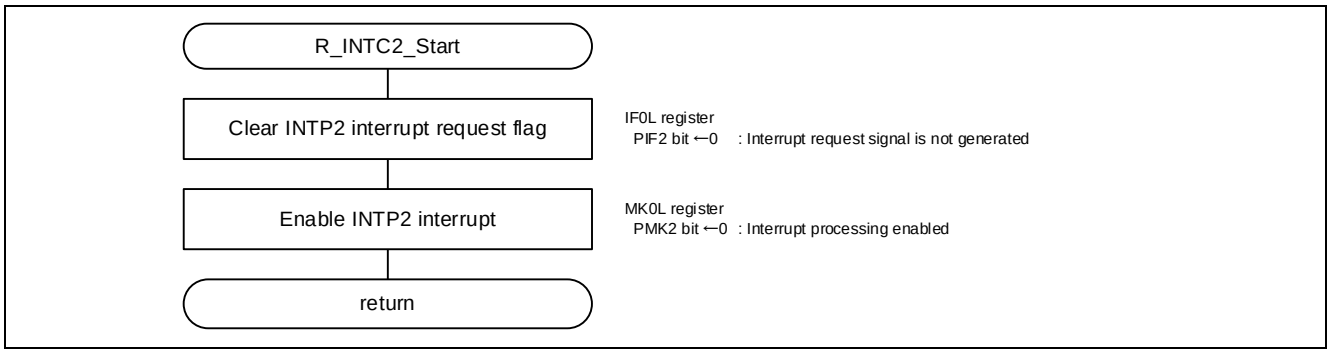

Figure 5.24 INTP2 Operation Start Function

INTP2 Interrupt Enable

- Interrupt request flag register (IF0L)
  Clear interrupt request flag.
- Interrupt request flag register (MK0L)
  Set to interrupt enabled.

Symbol: IF0L

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

Bit 4

PIF2	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state.

Symbol: MK0L

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

Bit 4

PMK2	Interrupt processing control
0	**Enables interrupt processing**
1	Disables interrupt processing

5.10.15 TAU0 channel 2 operation start function

Figure 5.25 shows the flowchart for TAU0 channel 2 operation start function.

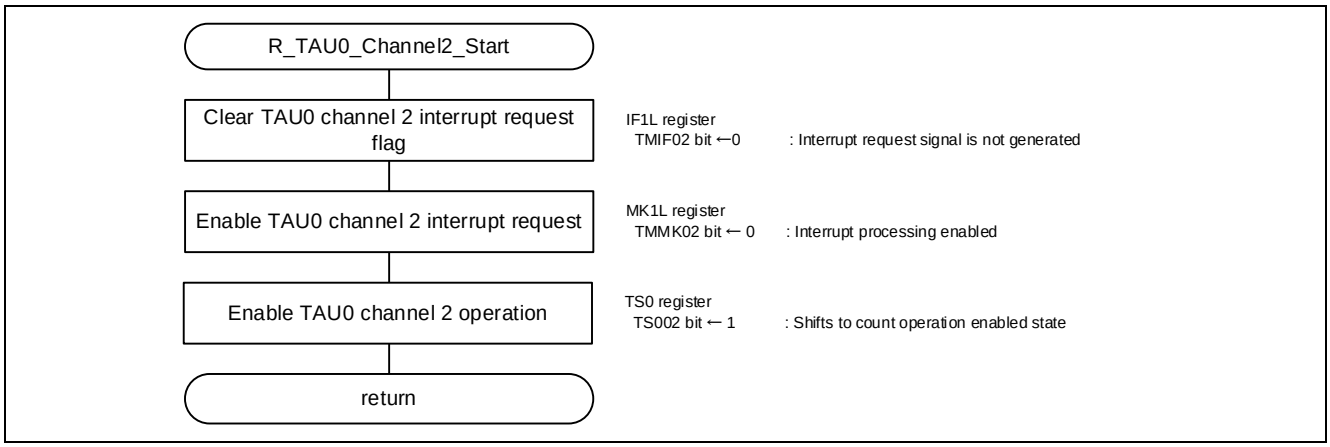

Figure 5.25 TAU0 Channel 2 Operation Start Function

TAU0 Channel 2 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
0	0	TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	FMIF
x	x	x	**0**	x	x	x	x

Bit 4

TMIF02	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
0	0	TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	FMMK
x	x	x	**0**	x	x	x	x

Bit 4

TMMK02	Interrupt processing control
0	**Enables interrupt processing**
1	Disables interrupt processing

TAU0 Channel 2 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	TSH03	0	TSH01	0	0	0	0	0	TS03	TS02	TS01	TS00
x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x

Bit 2

TS02	Operation enable (start) trigger for channel n
0	No trigger operation
1	**TEmn bit is set to 1, goes to count operation enabled state.**

5.10.16 TAU0 channel 1 operation start function

Figure 5.26 shows the flowchart for the TAU0 channel 1 operation start function.

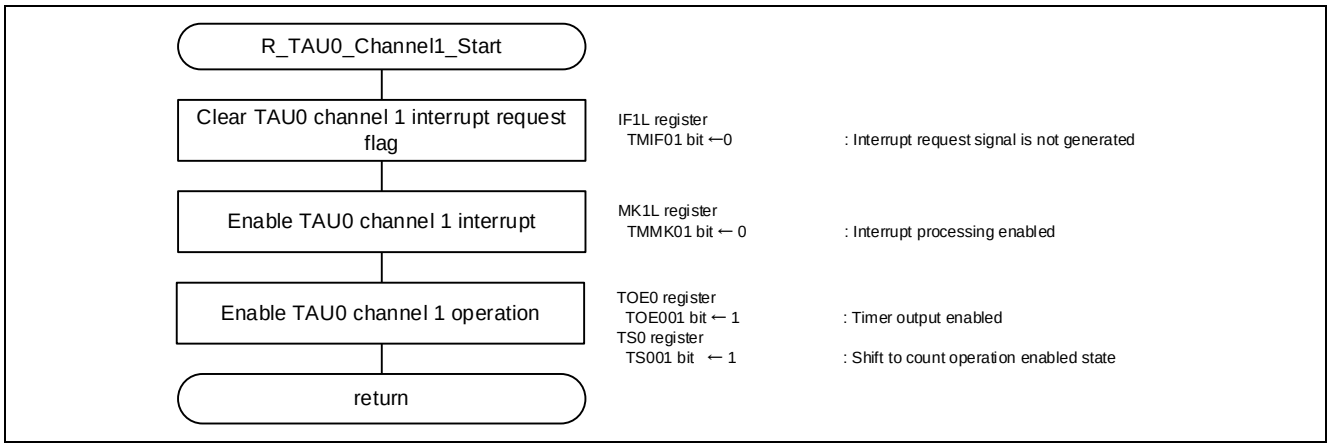

Figure 5.26 TAU0 Channel 1 Operation Start Function

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
0	0	TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	FMIF
x	x	x	x	**0**	x	x	x

Bit 3

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
0	0	TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	FMMK
x	x	x	x	**0**	x	x	x

Bit 3

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	TOE03	TOE02	TOE01	TOE00.
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x

Bit 1

TOE01	Timer output enable/disable of channel n
0	Disables timer output
1	**Enables timer output**

TAU0 Channel 2 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	TSH03	0	TSH01	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.**

5.10.17 TAU0 channel 1 operation stop function

Figure 5.27 shows the flowchart for the TAU0 channel 1 operation stop function.

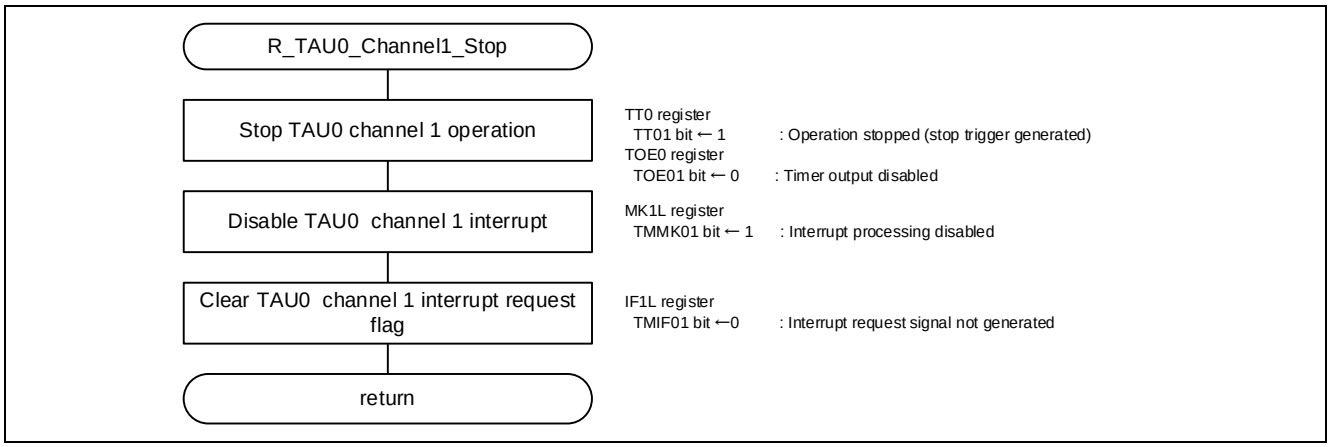

Figure 5.27 TAU0 Channel 1 Operation Stop Function

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	TTH03	0	TTH01	0	0	0	0	0	TT03	TT02	TT01	TT00
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x

Bit 1

TT01	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	TOE03	TOE02	TOE01	TOE00.
x	x	x	x	x	x	x	x	x	x	x	x	x	x	**0**	x

Bit 1

TOE01	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
0	0	TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	FMMK
x	x	x	x	**1**	x	x	x

Bit 3

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

Symbol: IF1L

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

Bit 3

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

5.10.18 TAU0 channel 2 operation stop function

Figure 5.28 shows the flowchart for TAU0 channel 2 operation stop function.

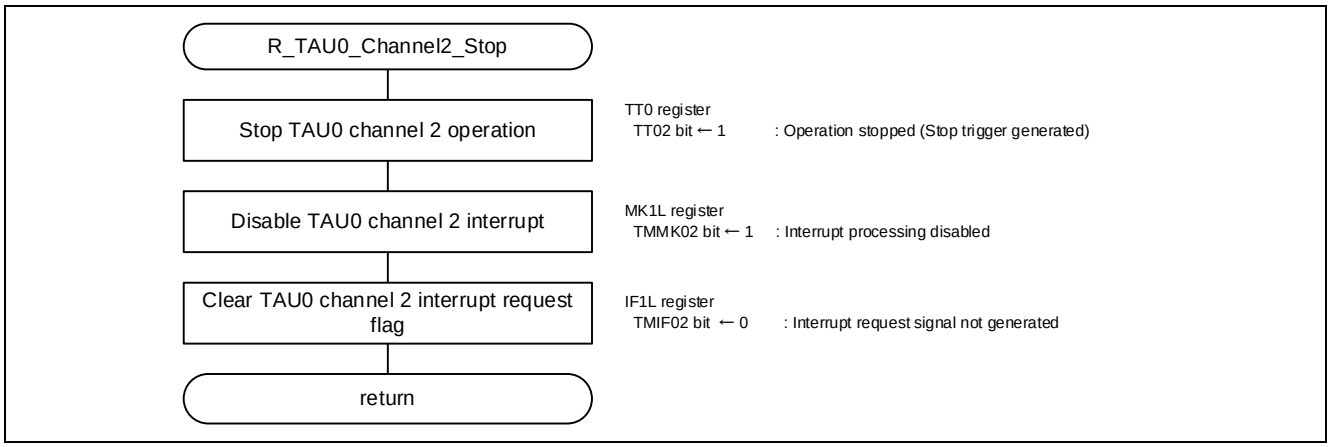

Figure 5.28 TAU0 Channel 2 Operation Stop Function

TAU0 Channel 2 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	TTH03	0	TTH01	0	0	0	0	0	TT03	TT02	TT01	TT00
x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x

Bit 2

TT02	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 2 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
0	0	TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	FMMK
x	x	x	**1**	x	x	x	x

Bit 4

TMMK02	Interrupt processing control
0	Enables interrupt processing
1	**Disables interrupt processing**

Symbol: IF1L

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

Bit 4

TMIF02	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state.

5.10.19 Wait function for LED blinking

Figure 5.29 shows the flowchart for the wait function for LED blinking.

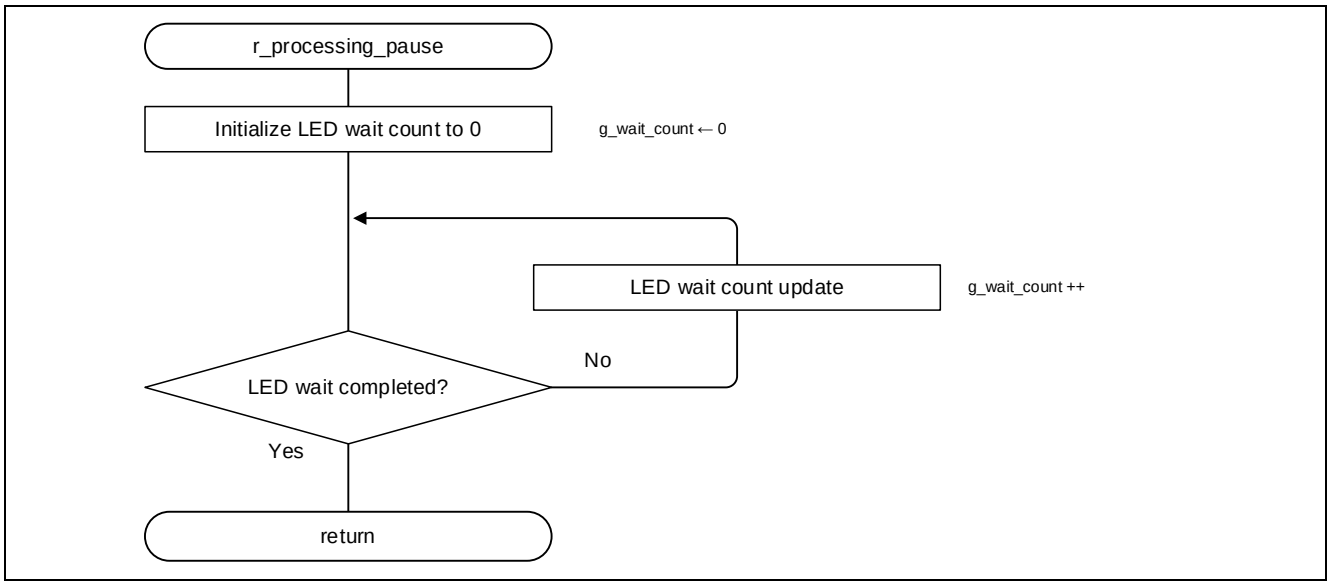

Figure 5.29 Wait Function for LED Blinking

5.10.20 TAU0 channel 2 count complete interrupt function

Figure 5.30 shows the flowchart for the TAU0 channel 2 count complete interrupt function.

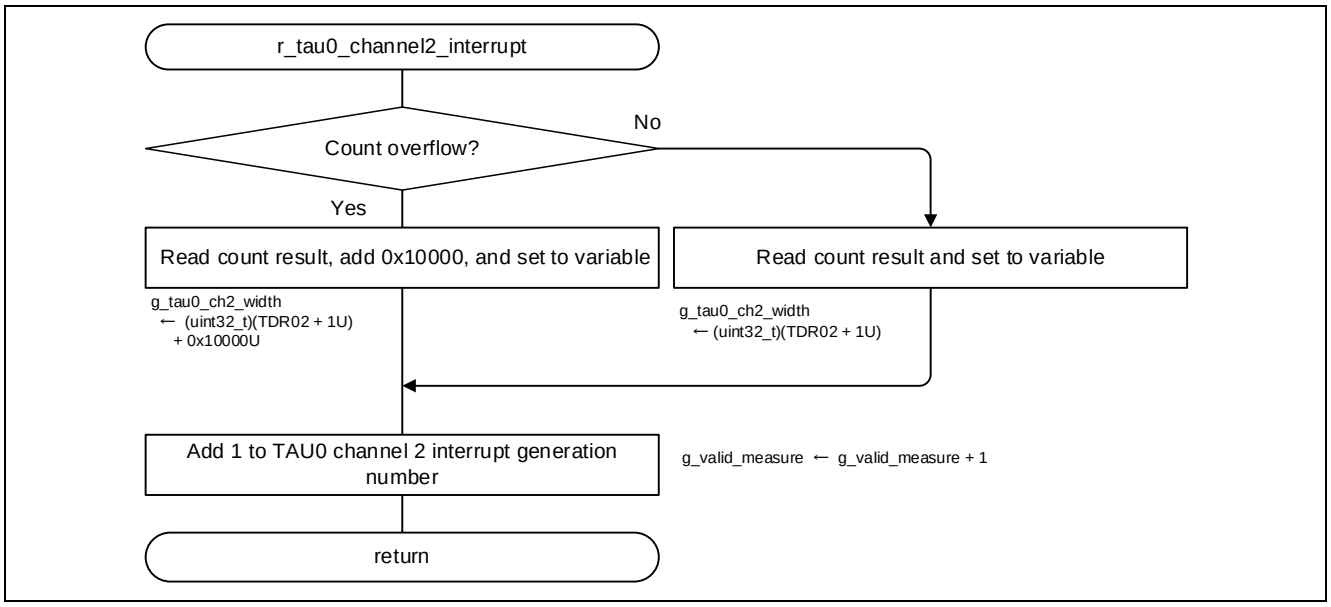

Figure 5.30 TAU0 Channel 2 Count Complete Interrupt Function

5.10.21 INTP2 interrupt processing

Figure 5.31 shows the flowchart for INTP2 interrupt processing.

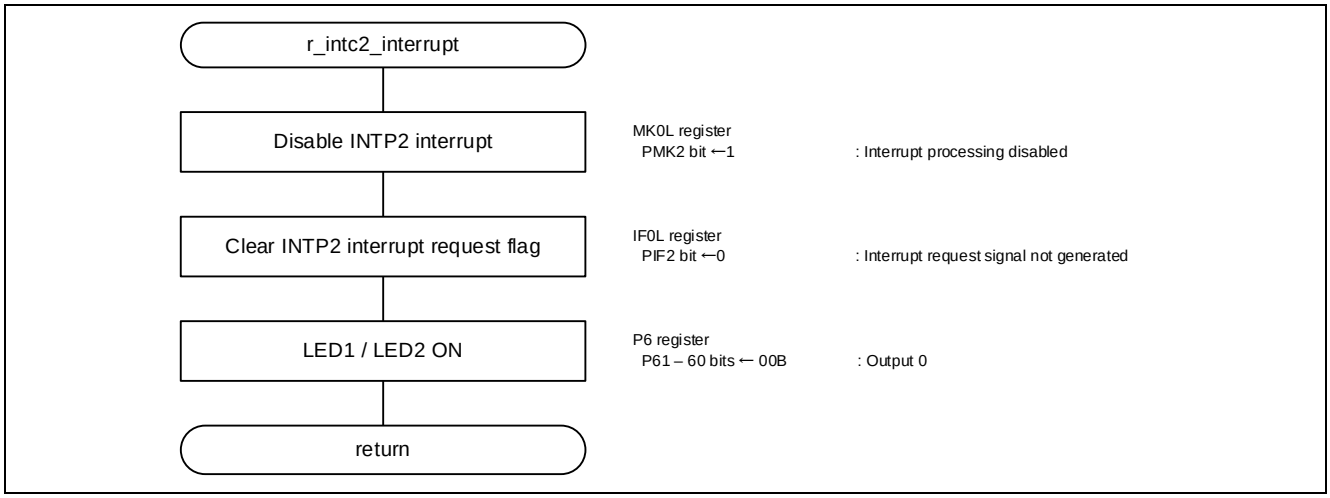

Figure 5.31 INTP2 Interrupt Processing

INTP2 Interrupt Disable

- Interrupt request flag register (MK0L)
  Set INTP interrupt to disabled.
- Interrupt request flag register (IF0L)
  Clear INTP interrupt request flag.

Symbol: MK0L

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

Bit 4

PMK2	Interrupt request flag
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
x	x	x	**0**	x	x	x	x

Bit 4

PIF2	Interrupt request flag
0	**Interrupt request signal not generated**
1	Interrupt request signal generated, goes to interrupt request state.

5.10.22 UART0 reception complete interrupt processing

Figure 5.32 shows the flowchart for UART0 reception complete interrupt processing.

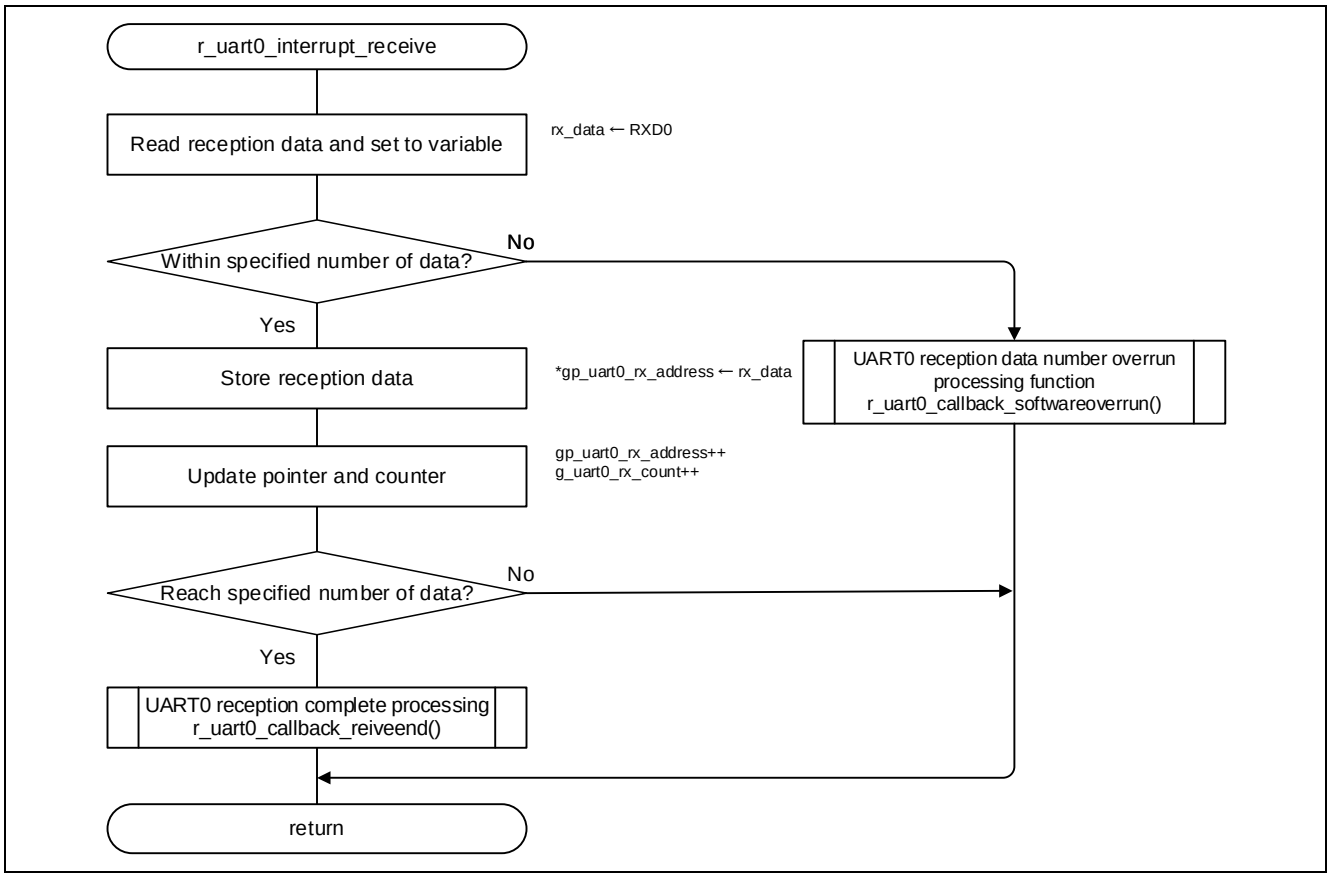

Figure 5.32 UART0 Reception Complete Interrupt Processing

5.10.23 UART0 receive data number overrun processing function

Figure 5.33 shows the flowchart for the UART0 receive data number overrun processing function.

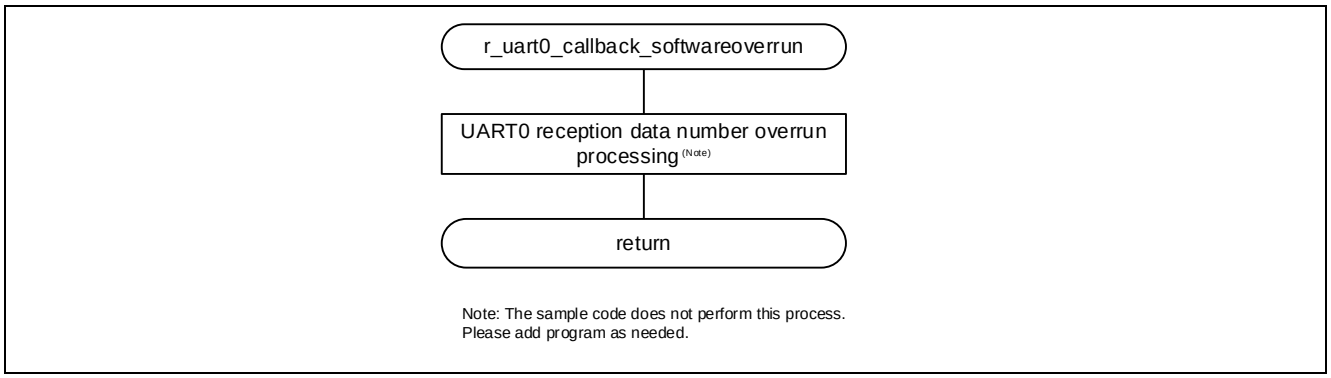

Figure 5.33 UART0 Receive Data Number Overrun Processing Function

5.10.24 UART0 reception complete processing

Figure 5.34 shows the flowchart for UART0 reception complete processing.

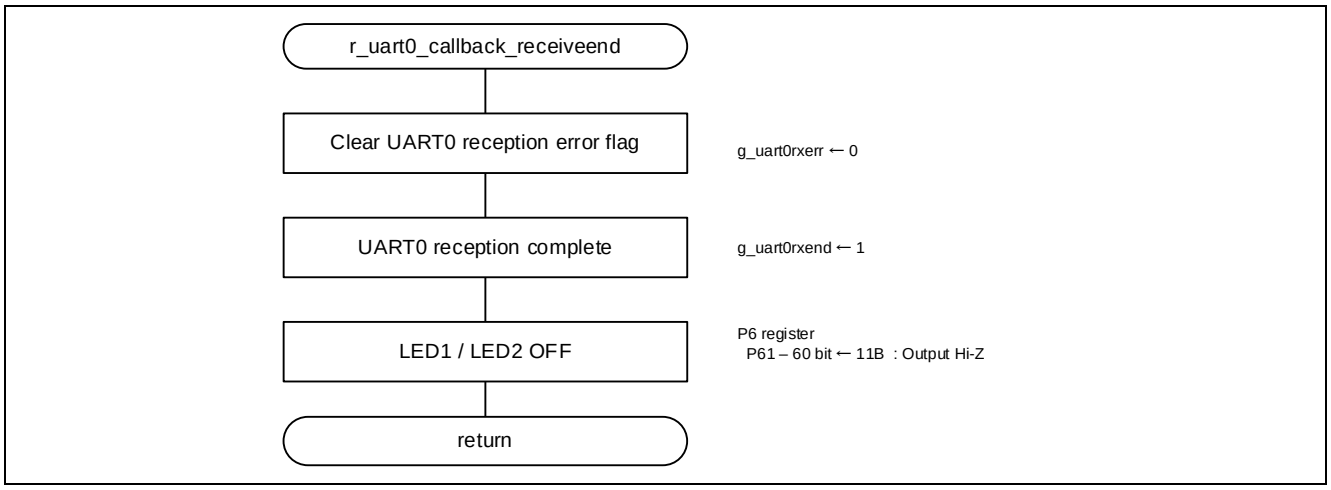

Figure 5.34 UART0 Reception Complete Processing

5.10.25 UART0 error interrupt function

Figure 5.35 shows the flowchart for the UART0 error interrupt function.

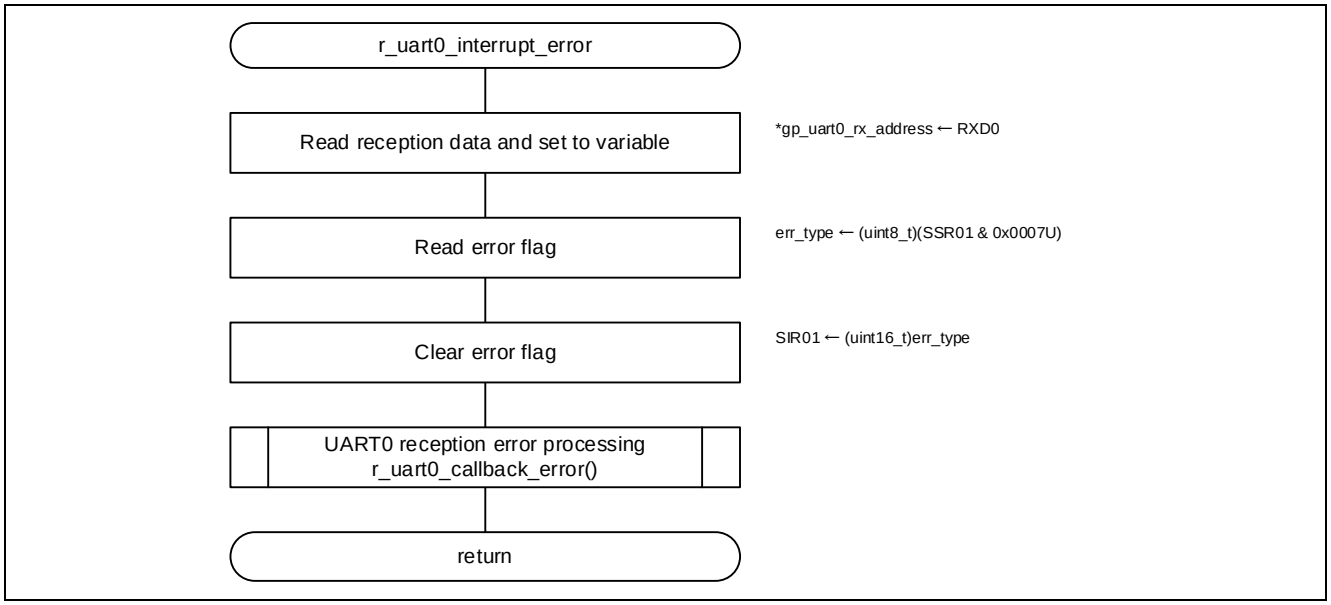

Figure 5.35 UART0 Error Interrupt Function

5.10.26 UART0 reception error processing

Figure 5.36 shows the flowchart for UART0 reception error processing.

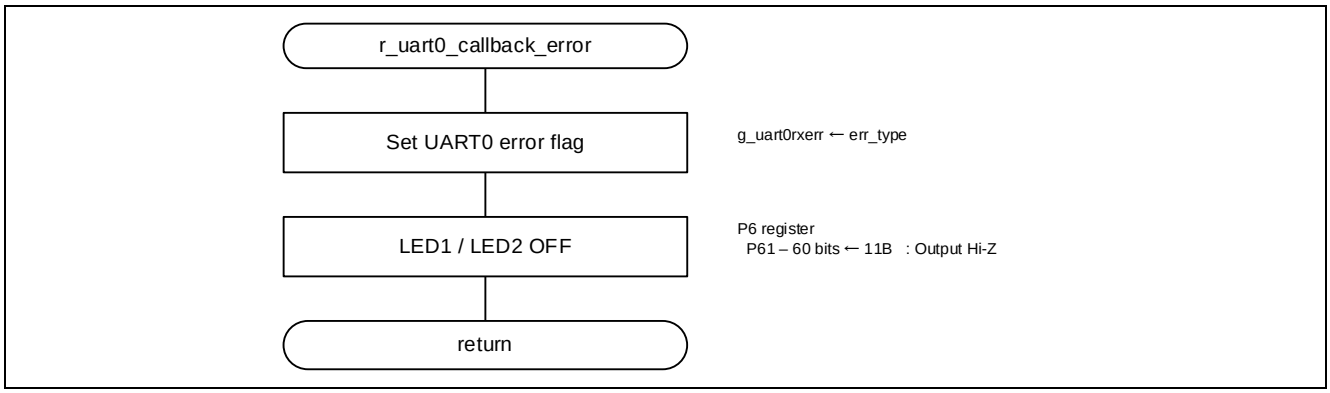

Figure 5.36 UART0 Reception Error Processing

6. Sample Code

Download the sample code from the Renesas Electronics homepage.

7. Reference Documents

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

RL78 family User's Manual: Software (R01US0015EJ)

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

Technical Updates

Obtain the latest information from the Renesas Electronics homepage.

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Jul. 01, 2015	-	First edition issued
1.01	Jan. 12, 2017	23	Corrected the Parameter
1.02	Sep. 02, 2022		IAR Compiler unsupported

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. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.