RL78/G1F

Door Lock Check Using Accelerometer

R01AN4303EJ0100
Rev.1.00
2018.09.28

Introduction

This application note describes how to check whether the door is locked or unlocked by using the accelerometer.

The application system described here uses the serial array unit, real-time clock, external interrupts, and buzzer output controller of the RL78/G1F.

Target Device

RL78/G1F

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

Contents

1. Specifications

In this application system, the accelerometer is used to detect the position of the knob of the thumb-turn installed on the door to determine whether the door is locked or unlocked.

The door lock check device enters lock mode after power on and waits for change in the acceleration speed. When it detects change in the acceleration speed indicating that the door is currently not locked (= unlocked), it enters unlock mode from lock mode and turns on the LED. Also, if the door is not locked within 10 seconds after unlocked, the device makes a buzzer sound. If the door is then locked, the device turns off the LED and stops a buzzer sound output.

In addition, the door lock check device described here supports the snooze function. When the snooze button installed on the device is pressed, the device enters hold mode to hold the LED turned off with a buzzer sound output stopped for one minute. Even in hold mode, if the device detects change in accelerometer, it immediately enters either lock mode or unlock mode. If the device detects no change in accelerometer for one minute after entering hold mode, it enters either lock mode or unlock mode according to the latest change in accelerometer obtained.

Figure 1.1 shows the schematic system configuration.

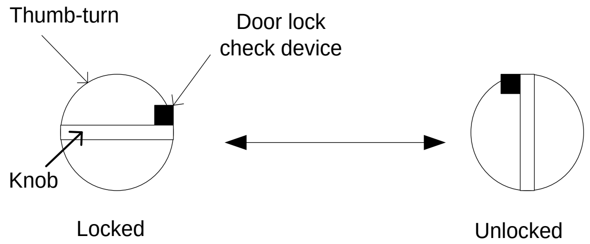

Figure 1.1 Schematic System Configuration

1.1 Accelerometer

This application system uses the accelerometer that is capable of measuring the acceleration speeds of three axes (x, y, and z). When actually preparing the circuit, the circuit should be designed so that the electrical characteristics of the accelerometer are satisfied.

2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	RL78/G1F
Operating frequencies	High-speed on-chip oscillator (HOCO) clock: 8MHz CPU/peripheral hardware clock: 8MHz Subsystem clock (XT1): 32.768kHz
Operating voltage	3.0V (operating range 1.8 V to 5.5V) LVD operation (V_{LVD}): Reset mode which uses 1.88 V (1.84 V to 1.91 V)
Integrated development environment (CS+)	CS+ for CC V6.00.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.05.00 from Renesas Electronics Corp.
Integrated development environment (e^2 studio)	e^2 studio V5.4.0.018 from Renesas Electronics Corp.
C compiler (e^2 studio)	CC-RL V1.05.00 from Renesas Electronics Corp.

3. Related Application Notes

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

RL78/G13 Initialization CC-RL (R01AN2575EJ) Application Note

RL78/G13 Real-Time Clock CC-RL (R01AN2590) Application Note

RL78/G13 Serial Array Unit for 3-Wire Serial I/O (Master Transmission/Reception) (R01AN2547EJ) Application Note

4. Description of the Hardware

4.1 Hardware Configuration Example

Figure 4.1 shows an example of hardware configuration that is used for this application note.

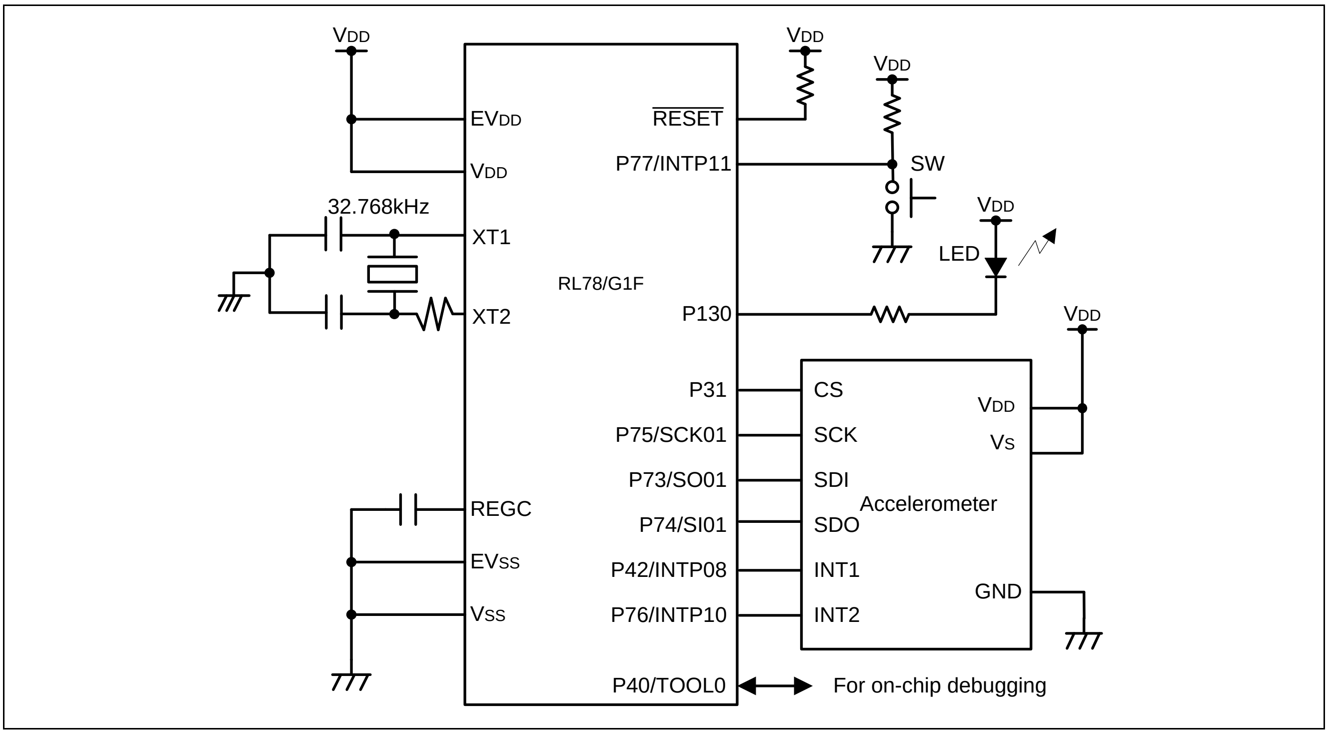

Figure 4.1 Hardware Configuration

Cautions:

1. The purpose of this circuit is only to provide the connection outline and the circuit is simplified accordingly. When designing and implementing an actual circuit, provide proper pin treatment and make sure that the hardware's electrical specifications are met (connect the input-only ports separately to VDD or VSS via a resistor).
2. Connect any pins whose name begins with EVSS to VSS and any pins whose name begins with EVDD to VDD, respectively.
3. VDD must be held at not lower than the reset release voltage (V_{LVD}) that is specified as LVD.

4.2 List of Pins to be Used

Table 4.1 lists the pins to be used and their functions.

Table 4.1 Pins to be Used and Their Functions

Pin Name	I/O	Description
P31	Output	Chip select output
P75/SCK01	Output	Serial clock output
P74/SI01	Input	Data reception
P73/SO01	Output	Data transmission
P140/PCLBUZ0	Output	Buzzer output
P42/INTP8	Input	Interrupt signal input from a sensor
P76/INTP10	Input	Interrupt signal input from a sensor
P77/INTP11	Input	Snooze button input
P130	Output	LED output port
P123/XT1	-	Subsystem clock resonator connection
P124/XT2/EXCLKS	-	Subsystem clock resonator connection

5. Description of the Software

5.1 Operation Outline

In this application system, the RL78/G1F is used for the door lock check device. When power is turned on, the initial setting is made for the RL78/G1F (specifically, the serial array unit, real-time clock, buzzer output controller, and external interrupts). Then the initial setting is made for the accelerometer through SPI communication using the port function and 3-wire serial I/O (CSI) communication of the serial array unit (SAU) of the RL78/G1F.

Figure 5.1 shows the position of the thumb-turn and accelerator while the door is locked.

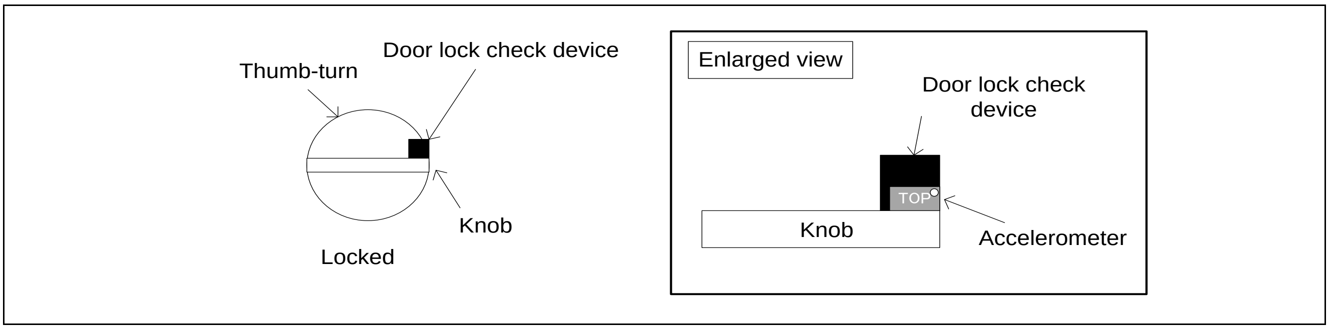

Figure 5.1 Position of Knob and Accelerometer while Door is Locked

In this application system, change in the x-axis of the accelerometer is detected. Therefore, the accelerometer needs to be installed so that no output is provided from the x-axis of the accelerometer while the door is locked, whereas output is provided while the door is unlocked. Also, note that it is assumed that power is turned on while the door is locked.

When no change is detected in the x-axis of the accelerometer (the position is confirmed) after the initial setting is complete for the accelerometer, lock mode is entered. After that, either lock mode, unlock mode, or hold mode is entered according to the accelerometer interrupt detected.

- Lock mode (operation status: CLOSE_DOOR)

  This mode is entered upon an interrupt from the accelerometer. Here the LED is turned off, buzzer output is stopped, and the RL78/G1F is placed in STOP mode.

- Unlock mode (operation status: OPEN_DOOR)

  This mode is entered upon an interrupt from the accelerometer and various interrupts from the RL78/G1F. Here the LED is turned on, and then a buzzer sound is made when 10 seconds elapse after unlock mode was entered.

- Hold mode (operation status: SW_WAIT)

  This mode is entered when the snooze button is pressed. When the snooze button is pressed, the LED is held turned off with a buzzer sound output stopped for one minute. Even in hold mode, if change in accelerometer is detected, either lock mode or unlock mode is immediately entered. If no change in accelerometer is detected for one minute after hold mode was entered, either lock mode or unlock mode is entered according to the latest change in accelerometer obtained.

Table 5.1 lists the required peripheral functions of the RL78/G1F and their use, and Figure 5.2 shows the state transition of the system.

Table 5.1 Required Peripheral Functions and Their Use

Peripheral Function	Description
Serial array unit 0 channel 1	Communication control of accelerometer
P31	Communication control of accelerometer
INTP8	INT1 interrupt detection of accelerometer（no x-axis output provided）
INTP10	INT2 interrupt detection of accelerometer（x-axis output provided）
INTP11	Snooze button input
PCLBUZ0	Buzzer output
Real-time clock (constant-period interrupt function)	Creation of 10 seconds after unlocking
Real-time clock (alarm interrupt function)	Creation of 1 minute after snooze button input
XT1	Subsystem clock resonator connection
XT2	Subsystem clock resonator connection

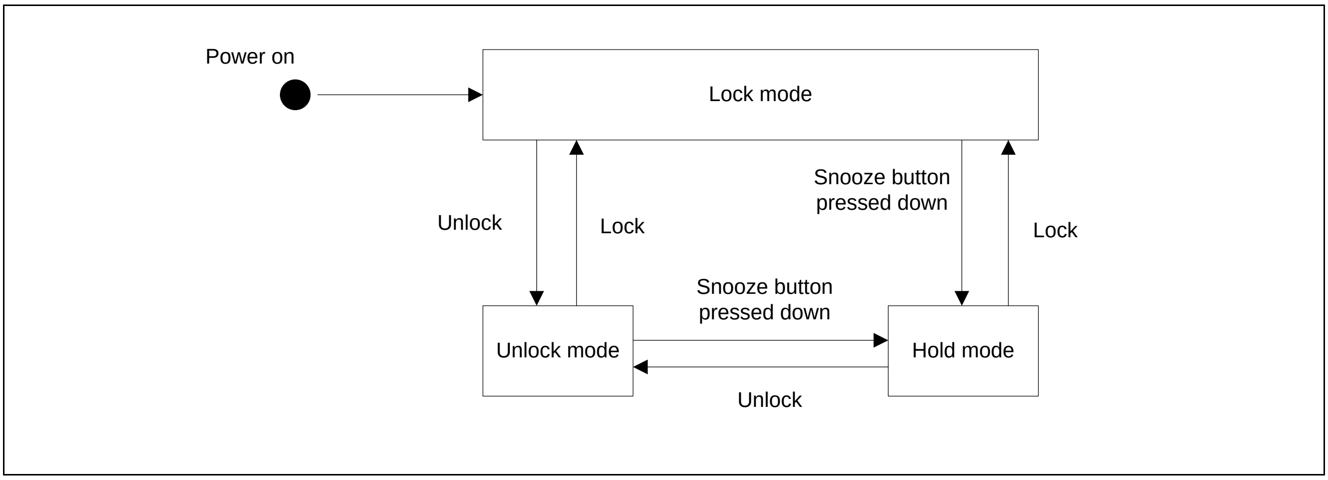

Figure 5.2 State Transition of System

① Makes the initial setting for the real-time clock (RTC).

- Selects the subsystem clock (f_{SUB}) as the RTC operation clock.
- Disables the RTC1HZ pin output.
- Sets a 24-hour system.
- Enables the real-time clock constant-period interrupt with the interrupt period of 1 second.
- Initializes the current time: 00:00:00, Wednesday, 1/1/2020
- Enables the constant-period interrupt and alarm interrupt.

② Makes the initial setting for the buzzer output controller.

- Selects f_{SUB}/2 as the operation clock of the buzzer output controller.

③ Makes the initial setting for the serial array unit 0 (SAU0).

- Uses SAU0 channel 1 as the CSI.
- Sets the serial clock to approximately 76800 bps (Hz).
- Sets single transfer mode as the operation mode.
- Selects type 1 phase for data and clock.
- Selects the MSB-first data transfer sequence.
- Sets the 8-bit data length.
- Enables the serial transfer end interrupt (INTCSI00) in single transfer mode.
- Uses the P75/SCK01 pin for clock output.
- Uses the P73/SO01 pin for data output.
- Uses the P74/SI01 pin for data input.
- Enables output through serial communication.

④ Makes the initial setting for the external interrupts.

- Uses the P00/INTP8 pin.
- Selects a rising edge as the valid edge for the INTP8 pin.
- Uses the P01/INTP10 pin.
- Selects a rising edge as the valid edge for the INTP10 pin.
- Uses the P11/INTP11 pin.
- Selects a falling edge as the valid edge for the INTP11pin.

⑤ Disables the maskable interrupt.

⑥ Starts the RTC and disable the constant-period interrupt and alarm interrupt.

⑦ Starts the SAU0.

⑧ Enables the external interrupts.

⑨ Enables the maskable interrupt.

⑩ Makes the initial setting for the accelerometer by using the SAU0 and P31 when the initial setting is complete for the device peripheral functions.

- For the initial setting for the accelerometer, refer to the appropriate manual and make sure to set the accelerometer properly. In this application system, the accelerometer is set through the SPI communication using the CSI and port function.
- In this application system, only the x-axis is used among three axes (x, y, and z).
- Change in the x-axis is detected by using the interrupt function of the accelerometer. The state in which no output is provided from the x-axis of the accelerometer is defined as the locked state, whereas the state in which output is provided from the x-axis is defined as the unlocked state.
- In the locked state, a high-level signal is output from the INT1 pin of the accelerometer.
- In the unlocked state, a high-level signal is output from the INT2 pin of the accelerometer.
- Figure 5.3 shows the output waveforms from the INT1 and INT2 pins of the accelerometer. The INT1 and INT2 pins become low by reading the interrupt source from the accelerometer through the SPI communication.

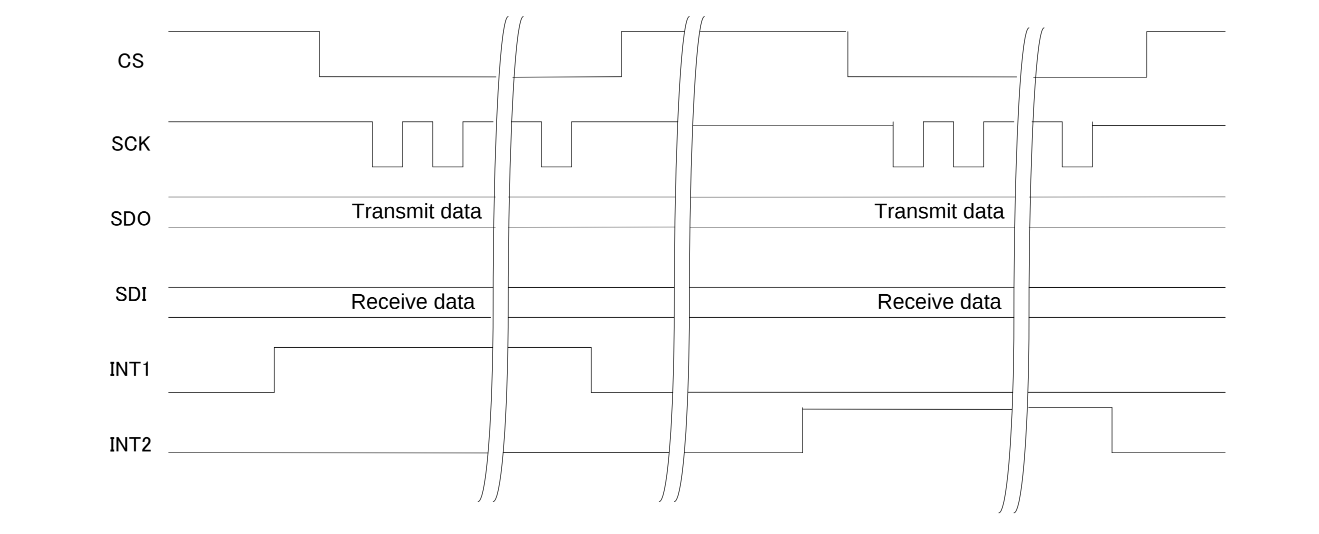

Figure 5.3 Output Waveforms from Accelerometer

- The x-axis output from the accelerometer is detected for 5 seconds.
- Table 5.2 shows the register names of the accelerometer, addresses, and the setting values.

Table 5.2 Settings for Accelerometer

Register Name	Address	Setting Value
DATA_FORMAT	0x31	0x0B
THRESH_ACT	0x24	0x10
THRESH_INACT	0x25	0x10
ACT_INACT_CTL	0x27	0x44
TIME_INACT	0x26	0x05
INT_ENABLE	0x2E	0x18
INT_MAP	0x2F	0x10
BW_RATE	0x2C	0x09
POWER_CTL	0x2D	0x38

⑪ Waits for a high-level output (locked state) from the INT1 pin of the accelerometer. When the locked state is entered, changes the status to lock mode.

⑫ The RL78/G1F enters STOP mode.

⑬ The RL78/G1F returns from STOP mode by each interrupt source and enters either unlock mode, lock mode, or hold mode. Operation in each mode is described below.

- Unlock mode (operation status: OPEN_DOOR)

  If the 10-second count start flag is off, goes to ⑫.

  If the 10-second count start flag is on, performs the following operation and then goes to ⑫.

  - Clears the accelerometer interrupt source.
  - Obtains (not uses) the acceleration speed value of the accelerometer.
  - Disables the alarm interrupt.
  - Enables the RTC constant-period interrupt.
  - Initializes the counter used for creating 10 seconds after unlocking.
  - Turns on the LED.

- Lock mode (operation status: CLOSE_DOOR)

  Clears the accelerometer interrupt source, then performs the following operations, and goes to ⑫.

  - Disables the RTC constant-period interrupt.
  - Disables the alarm interrupt.
  - Turns off the LED.
  - Disables buzzer output.

- Hold mode (operation status: SW_WAIT)

  If the 1-minute count start flag is off, goes to ⑫.

  If the 1-minute count start flag is on, performs the following operations and then goes to ⑫.

  - Turns off the 1-minute count start flag.
  - Obtains the current time from the RTC.
  - Sets the current time that was obtained from the RTC + 1 minute as the time of alarm.
  - Disables the alarm interrupt.
  - Disables buzzer output.

The following operations are performed to unlock the door.

- When the accelerometer detects change in the acceleration speed of the x-axis, the output voltage of the INT2 pin changes.
- The RL78/G1F is informed of the voltage change of the INT2 pin of the accelerometer by the external interrupt (INTP10), and returns from STOP mode.
- Changes the status to unlock mode in the external interrupt processing. Also turns on the 10-second count start flag.
- Determines the status in the main processing, and performs the following operations.
- If the 10-second count start flag is off, enters STOP mode.
- If the 10-second count start flag is on, performs the following operations.
  - Turns off the 10-second count start flag.
  - Clears the interrupt request from the accelerometer.
  - Obtains the data of three axes (x, y, and z) of the accelerometer.
  - Switches the P130 output and turns on the LED.
  - Enables the RTC constant-period interrupt (interrupt period of 1 second), and enters STOP mode.
- Increments the 10-second counter every second in the constant-period interrupt processing.
- Makes a buzzer sound when 10 seconds elapse in the constant-period interrupt processing.

The following operations are performed to lock the door.

- When the accelerometer detects change in the acceleration speed of the x-axis, the output voltage of the INT1 pin changes.
- The RL78/G1F is informed of the voltage change of the INT1 pin of the accelerometer by the external interrupt (INTP8), and returns from STOP mode.
- Changes the status to lock mode in the external interrupt processing.
- Determines the status in the main processing, and performs the following operations.
- Clears the interrupt request from the accelerometer.
- Disables the RTC constant-period interrupt.
- Disables the RTC alarm interrupt.
- Changes the status to lock mode.
- Initializes the 10-second counter.
- Switches the P130 output and turns off the LED.
- Stops buzzer output.

The following operations are performed when the snooze button is pressed.

- The RL78/G1F is informed of the level change of the snooze button by the external interrupt (INTP11), and returns from STOP mode.
- Reduces noise by the 12-bit interval timer in the external interrupt processing.
- If the current status is not hold mode (is unlock or lock mode), backs up the status.
- Changes the status to hold mode and turns on the 1-minute count start flag.
- Determines the status in the main processing, and performs the following operations.
- Checks the 1-minute count start flag, and enters STOP mode if the flag is off.
- Checks the 1-minute count start flag, and performs the following operations if the flag is on.

  - Turns off the 1-minute count start flag.

  - Reads the RTC time.

  - Sets the time read from the RTC + approximately 1 minute as the time of RTC alarm interrupt.

  - Disables the RTC constant-period interrupt.

  - Enables the RTC alarm interrupt.
- When the alarm interrupt occurs, disables the alarm interrupt and turns on the 1-mimute count start flag. Then returns the status from the backup status. If the status is unlock mode, turns on the 10-second count start flag.

5.2 Option Byte Settings

Table 5.3 shows the option byte settings.

Table 5.3 Option Byte Settings

Address	Setting Value	Contents
000C0H/010C0H	01101110B	Watchdog timer operation is stopped (count is stopped after reset)
000C1H/010C1H	00111111B	LVD reset mode, 1.84 V (1.84 V to 1.88 V)
000C2H/010C2H	10101010B	LS mode, HOCO: 8 MHz
000C3H/010C3H	10000100B	Enables the on-chip debugger.

5.3 List of Constants

Table 5.4 lists the constants that are used in this sample program.

Table 5.4 Constants for the Sample Program

Constant	Setting	Description
OPEN_DOOR	0	Operation status setting (unlock mode)
CLOSE_DOOR	1	Operation status setting (lock mode)
SW_WAIT	2	Operation status setting (hold mode)
OFF	0	Flag setting (off)
ON	1	Flag setting (on)

5.4 List of Variables

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

Table 5.5 Global Variables for the Sample Program

Type	Variable Name	Contents	Function Used
uint8_t	g_operation_status	Holds operation status.	main, r_rtc_callback_alarm, r_intc8_interrupt, r_intc10_interrupt, r_intc11_interrupt
uint8_t	g_status_backup	Holds backup of operation status.	r_rtc_callback_alarm, r_intc8_interrupt, r_intc10_interrupt,
uint8_t	g_wait_enable_flag	1-minute count start flag	main, r_rtc_callback_alarm, r_intc11_interrupt
uint8_t	g_receive_end	Transmit end flag	R_SENSOR_Transmit_Check, R_SENSOR_write, R_SENSOR_read, r_csi01_callback_receiveend
uint8_t	g_tx_data[6]	Transmit data array	R_SENSOR_write, R_SENSOR_read
uint8_t	g_rx_data[6]	Receive data array	R_SENSOR_Init, R_SENSOR_write, R_SENSOR_read
uint8_t	g_open_flag	10-second count start flag	main, r_cg_intp_user.c, r_rtc_callback_alarm
uint8_t	g_10s_timer_count	10-second counter	main, r_rtc_callback_constperiod,
uint8_t	g_data_length	Holds the number of transmit and receive data bytes.	R_SENSOR_write, R_SENSOR_read
int16_t	g_temp_x	Holds the x-axis acceleration speed (integer).	R_SENSOR_Init
int16_t	g_temp_y	Holds the y-axis acceleration speed (integer).	R_SENSOR_Init
int16_t	g_temp_z	Holds the z-axis acceleration speed (integer).	R_SENSOR_Init
float	g_xx	Holds the x-axis acceleration speed (real number).	R_SENSOR_Init
float	g_yy	Holds the y-axis acceleration speed (real number).	R_SENSOR_Init
float	g_zz	Holds the z-axis acceleration speed (real number).	R_SENSOR_Init

5.5 List of Functions

Table 5.6 summarizes the functions that are used in this sample program.

Table 5.6 Functions

Function Name	Outline
main	Main function
R_MAIN_UserInit	Main user initialization function
R_INTC8_Start	Enables INTP8 external interrupt.
R_INTC10_Start	Enables INTP10 external interrupt.
R_INTC11_Start	Enables INTP11 external interrupt.
R_IT_Start	Starts 12-bit interval timer operation.
R_IT_Stop	Stops 12-bit interval timer operation.
R_RTC_Start	Starts real-time clock operation.
R_CSI01_Start	Starts CSI01 operation.
R_PCLBUZ0_Start	Starts buzzer output controller operation.
R_PCLBUZ0_Stop	Stops buzzer output controller operation.
R_RTC_Get_CounterValue	Reads real-time clock counter value.
R_RTC_Set_AlarmOn	Enables real-time clock alarm interrupt.
R_RTC_Set_AlarmOff	Disables real-time clock alarm interrupt.
R_RTC_Set_AlarmValue	Sets real-time clock alarm conditions.
R_RTC_Set_ConstPeriodInterruptOn	Enables real-time clock constant-period interrupt.
R_RTC_Set_ConstPeriodInterruptOff	Disables real-time clock constant-period interrupt.
R_SENSOR_Init	Initializes accelerometer.
R_SENSOR_Transmit_Check	Checks if data transmission/reception of accelerometer is complete.
R_SENSOR_write	Writes data to accelerometer.
R_SENSOR_read	Reads data from accelerometer.
r_rtc_interrupt	Processes real-time clock interrupt.
r_rtc_callback_constperiod	Callback function of real-time clock constant-period interrupt
r_rtc_callback_alarm	Callback function of real-time clock alarm interrupt
r_csi01_interrupt	Processes CSI01 interrupt.
r_csi01_callback_receiveend	Callback function of CSI01 data reception
r_intc8_interrupt	Processes INTP8 external interrupt.
r_intc10_interrupt	Processes INTP10 external interrupt.
r_intc11_interrupt	Processes INTP11 external interrupt.

5.6 Function Specifications

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

[Function Name] main

Synopsis	Main function
Header	r_cg_macrodriver.h, r_cg_cgc.h, r_cg_port.h, r_cg_rtc.h, r_cg_it.h, r_cg_pclbuz.h, r_cg_sau.h, r_cg_intp.h, r_SENSOR.h, r_cg_userdefine.h
Declaration	-
Explanation	After executing the main user initialization function, enables interrupts and initializes the accelerometer. After initializing the accelerometer, enters STOP mode and waits for interrupts to be generated. When an interrupt is generated from the accelerometer, controls the LED and buzzer.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_MAIN_UserInit

Synopsis	Main user initialization function
Header	r_cg_macrodriver.h, r_cg_cgc.h, r_cg_port.h, r_cg_rtc.h, r_cg_it.h, r_cg_pclbuz.h, r_cg_sau.h, r_cg_intp.h, r_SENSOR.h, r_cg_userdefine.h
Declaration	static void R_MAIN_UserInit(void);
Explanation	Disables interrupts by the DI instruction. Then starts the real-time clock operation and disables the constant-period interrupt and alarm interrupt.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_INTC8_Start

Synopsis	Enables INTP8 external interrupt.
Header	r_cg_macrodriver.h, r_cg_intp.h, r_cg_userdefine.h
Declaration	void R_INTC8_Start(void);
Explanation	Enables the INTP8 external interrupt.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_INTC10_Start

Synopsis	Enables INTP10 external interrupt.
Header	r_cg_macrodriver.h, r_cg_intp.h, r_cg_userdefine.h
Declaration	void R_INTC10_Start(void);
Explanation	Enables the INTP10 external interrupt.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_INTC11_Start

Synopsis	Enables INTP11 external interrupt.
Header	r_cg_macrodriver.h, r_cg_intp.h, r_cg_userdefine.h
Declaration	void R_INTC11_Start(void);
Explanation	Enables the INTP11 external interrupt.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_IT_Start

Synopsis	Starts 12-bit interval timer operation.
Header	r_cg_macrodriver.h, r_cg_it.h, r_cg_userdefine.h
Declaration	void R_IT_Start(void);
Explanation	Starts 12-bit interval timer operation.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_IT_Stop

Synopsis	Stops 12-bit interval timer operation.
Header	r_cg_macrodriver.h, r_cg_it.h, r_cg_userdefine.h
Declaration	void R_IT_Stop (void);
Explanation	Stops 12-bit interval timer operation.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_RTC_Start

Synopsis	Starts real-time clock operation.
Header	r_cg_macrodriver.h, r_cg_rtc.h, r_cg_userdefine.h
Declaration	void R_RTC_Start (void);
Explanation	Starts real-time clock operation.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_CSI01_Start

Synopsis	Starts CSI01 operation.
Header	r_cg_macrodriver.h, r_cg_rtc.h, r_cg_userdefine.h
Declaration	void R_CSI01_Start (void);
Explanation	Starts CSI01 data transmission/reception.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_PCLBUZ0_Start

Synopsis	Starts buzzer output controller operation.
Header	r_cg_macrodriver.h, r_cg_pclbuz.h, r_cg_userdefine.h
Declaration	void R_PCLBUZ0_Start(void);
Explanation	Starts buzzer output controller operation.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_PCLBUZ0_Stop

Synopsis	Stops buzzer output controller operation.
Header	r_cg_macrodriver.h, r_cg_pclbuz.h, r_cg_userdefine.h
Declaration	void R_PCLBUZ0_Stop(void);
Explanation	Stops buzzer output controller operation.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_RTC_Get_CounterValue

Synopsis	Reads real-time clock counter value.
Header	r_cg_macrodriver.h, r_cg_rtc.h, r_cg_userdefine.h
Declaration	void R_RTC_Get_CounterValue (rtc_counter_value_t * const counter_read_val) ;
Explanation	Reads the real-time clock counter value.
Arguments	rtc_counter_value_t const counter_read_val: [count register value that was read]
Return value	None
Remarks	None

[Function Name] R_RTC_Set_AlarmOn

Synopsis	Starts alarm interrupt processing.
Header	r_cg_macrodriver.h, r_cg_rtc.h, r_cg_userdefine.h
Declaration	void R_RTC_Set_AlarmOn (void);
Explanation	Starts alarm interrupt processing.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_RTC_Set_AlarmOff

Synopsis	Stops alarm interrupt processing.
Header	r_cg_macrodriver.h, r_cg_rtc.h, r_cg_userdefine.h
Declaration	void R_RTC_Set_AlarmOff (void);
Explanation	Stops alarm interrupt processing.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_RTC_Set_AlarmValue

Synopsis	Sets alarm conditions (day of week, hour, and minute).
Header	r_cg_macrodriver.h, r_cg_rtc.h, r_cg_userdefine.h
Declaration	void R_RTC_Set_AlarmValue (rtc_alarm_value_t alarm_val) ;
Explanation	Sets the conditions for alarm generation (day of the week, hour, and minute).
Arguments	rtc_alarm_value_t alarm_val :[time of alarm]
Return value	None
Remarks	None

[Function Name] R_RTC_Set_ConstPeriodInterruptOn

Synopsis	Sets the generation period of constant-period interrupt and enables the constant-period interrupt.
Header	r_cg_macrodriver.h, r_cg_rtc.h, r_cg_userdefine.h
Declaration	void R_RTC_Set_ConstPeriodInterruptOn (rtc_int_period_t period) ;
Explanation	Sets the generation period of constant-period interrupt and enables the constant-period interrupt.
Arguments	rtc_int_period_t period :[period of constant period]
Return value	None
Remarks	None

[Function Name] R_RTC_Set_ConstPeriodInterruptOff

Synopsis	Disables constant-period interrupt.
Header	r_cg_macrodriver.h, r_cg_rtc.h, r_cg_userdefine.h
Declaration	void R_RTC_Set_ConstPeriodInterruptOff (void);
Explanation	Disables the constant-period interrupt.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_SENSOR_Init

Synopsis	Initializes accelerometer.
Header	r_cg_macrodriver.h, r_SENSOR.h, r_cg_userdefine.h
Declaration	void R_SENSOR_Init (void);
Explanation	Initializes the accelerometer.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_SENSOR_Transmit_Check

Synopsis	Waits for completion of data transmission/reception of the accelerometer.
Header	r_cg_macrodriver.h, r_cg_userdefine.h
Declaration	void R_SENSOR_Transmit_Check (void);
Explanation	Checks if data transmission/reception of the accelerometer is complete.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_SENSOR_write

Synopsis	Writes data to accelerometer.
Header	r_cg_macrodriver.h, r_cg_sau.h, r_cg_userdefine.h
Declaration	void R_SENSOR_write (void);
Explanation	Writes data to the accelerometer.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_SENSOR_read

Synopsis	Reads data from accelerometer.
Header	r_cg_macrodriver.h, r_cg_sau.h, r_cg_userdefine.h
Declaration	void R_SENSOR_read (void);
Explanation	Reads data from the accelerometer.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_rtc_interrupt

Synopsis	Processes real-time clock interrupt.
Header	r_cg_macrodriver.h, r_cg_rtc.h, r_cg_userdefine.h
Declaration	void r_rtc_interrupt (void);
Explanation	Performs the appropriate process according to the real-time clock constant-period interrupt or alarm interrupt.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_rtc_callback_constperiod

Synopsis	Processes real-time clock constant-period interrupt.
Header	r_cg_macrodriver.h, r_cg_rtc.h, r_cg_userdefine.h
Declaration	void r_rtc_callback_constperiod (void);
Explanation	Increments the 10-second counter each time the real-time clock constant-period interrupt is generated every second. When the counter reaches 10, initializes the counter and starts buzzer output.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_rtc_callback_alarm

Synopsis	Processes real-time clock alarm interrupt.
Header	r_cg_macrodriver.h, r_cg_rtc.h, r_cg_userdefine.h
Declaration	void r_rtc_callback_constperiod (void);
Explanation	When the real-time clock alarm interrupt is generated, disables the alarm interrupt and writes the backup operation status value as the operation status.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_csi01_interrupt

Synopsis	Processes CSI01 transfer end interrupt.
Header	r_cg_macrodriver.h, r_cg_sau.h, r_cg_userdefine.h
Declaration	void r_csi01_interrupt (void);
Explanation	If there is data to be transmitted, reads received data and starts transmitting data to be transmitted. If there is no data to be transmitted, reads received data.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_csi01_callback_receiveend

Synopsis	CSI01 transfer end function
Header	r_cg_macrodriver.h, r_cg_sau.h, r_cg_userdefine.h
Declaration	void r_csi01_callback_receiveend (void);
Explanation	When the number of received CSI01 data bytes reaches the specified number, turns off the receive end flag.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_intc8_interrupt

Synopsis	Processes INTP8 external interrupt.
Header	r_cg_macrodriver.h, r_cg_intp.h, r_cg_userdefine.h
Declaration	void r_intc8_interrupt (void);
Explanation	Detects the rising edge of the INT1 pin of the accelerometer.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_intc10_interrupt

Synopsis	Processes INTP10 external interrupt.
Header	r_cg_macrodriver.h, r_cg_intp.h, r_cg_userdefine.h
Declaration	void r_intc10_interrupt (void);
Explanation	Detects the rising edge of the INT2 pin of the accelerometer.
Arguments	None
Return value	None
Remarks	None

[Function Name] r_intc11_interrupt

Synopsis	Processes INTP11 external interrupt.
Header	r_cg_macrodriver.h, r_cg_intp.h, r_cg_userdefine.h
Declaration	void r_intc11_interrupt (void);
Explanation	Detects the falling edge of the snooze button input.
Arguments	None
Return value	None
Remarks	None

5.7 Flowcharts

5.7.1 Overall Flow

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

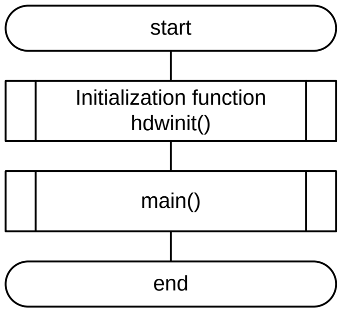

Figure 5.4 Overall Flow

5.7.2 Initialization Function

Figure 5.5 shows the flowchart for the initialization function.

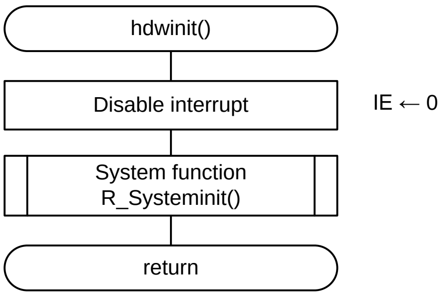

Figure 5.5 Initialization Function

5.7.3 System Function

Figure 5.6 shows the flowchart for the system function.

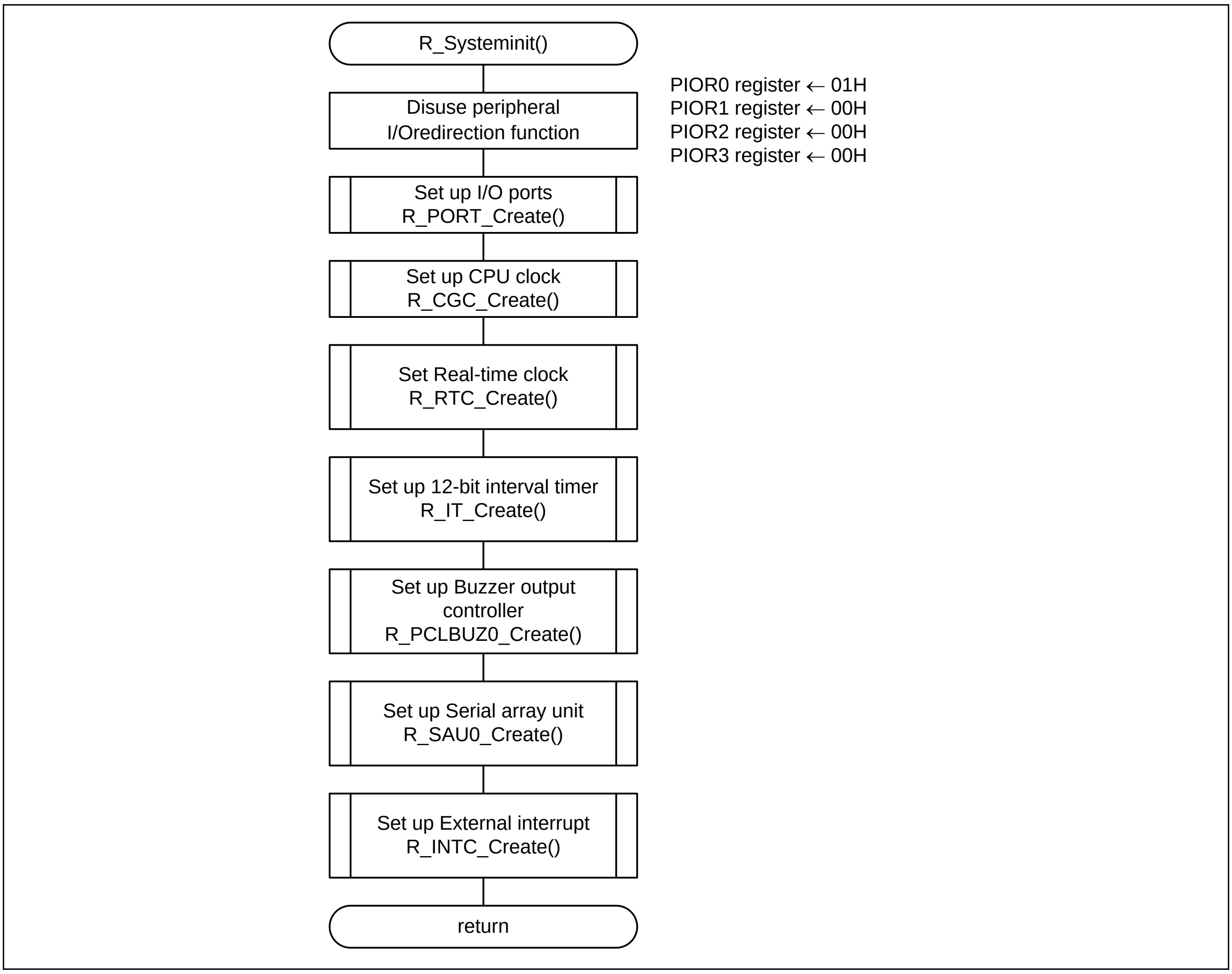

Figure 5.6 System Function

5.7.4 I/O Port Setup

Figure 5.7 shows the flowchart for I/O port setup.

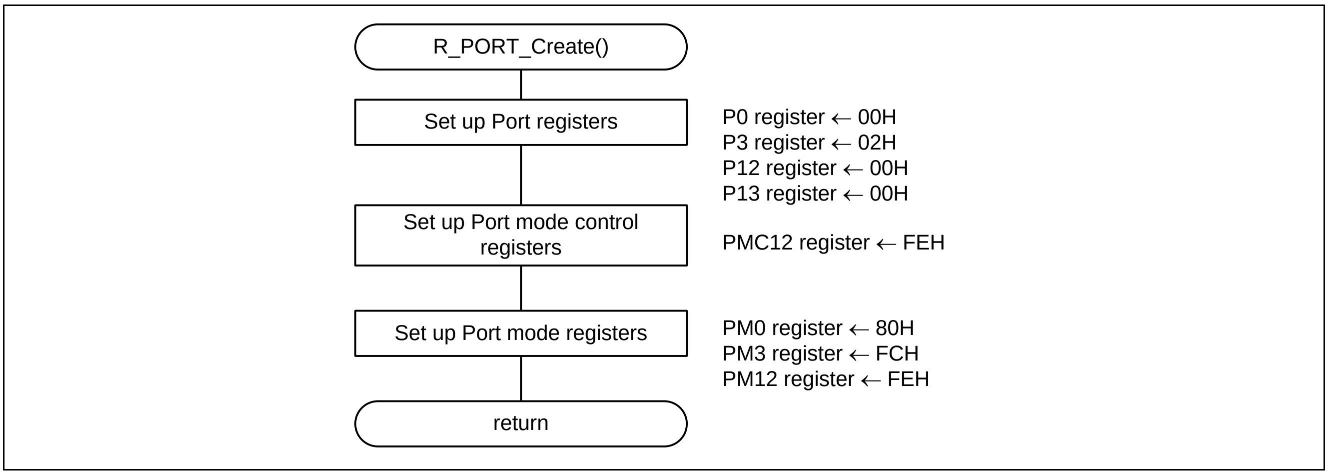

Figure 5.7 I/O Port Setup

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

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

5.7.5 CPU Clock Setup

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

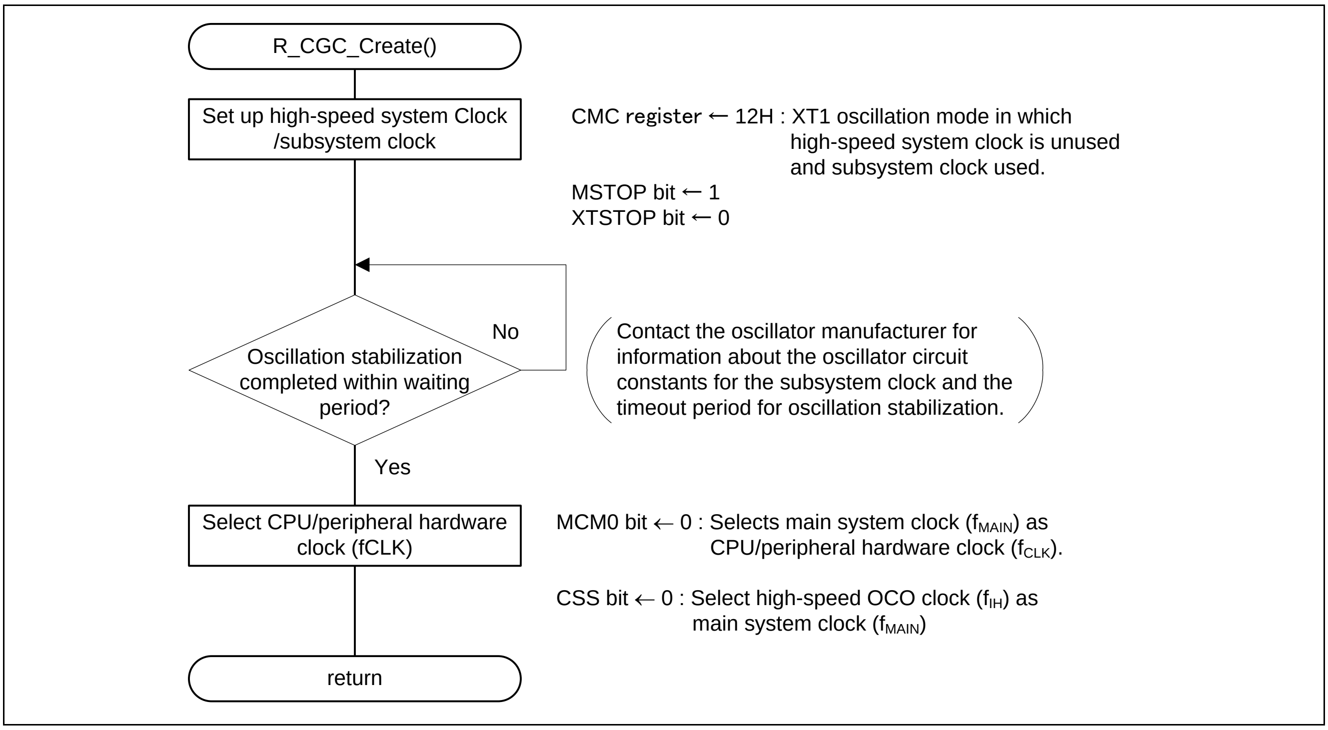

Figure 5.8 CPU Clock Setup

Remarks: CPU clock setup (R_CGC_Create()) determines whether stabilization of subsystem clock oscillation is completed within a waiting period (about 1 second). This period is specified with constant CGC_SUBWAITTIME in r_cg_cgc.h.

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

5.7.6 RTC Initial Setting

Figure 5.9 shows the flowchart for the RTC initial setting.

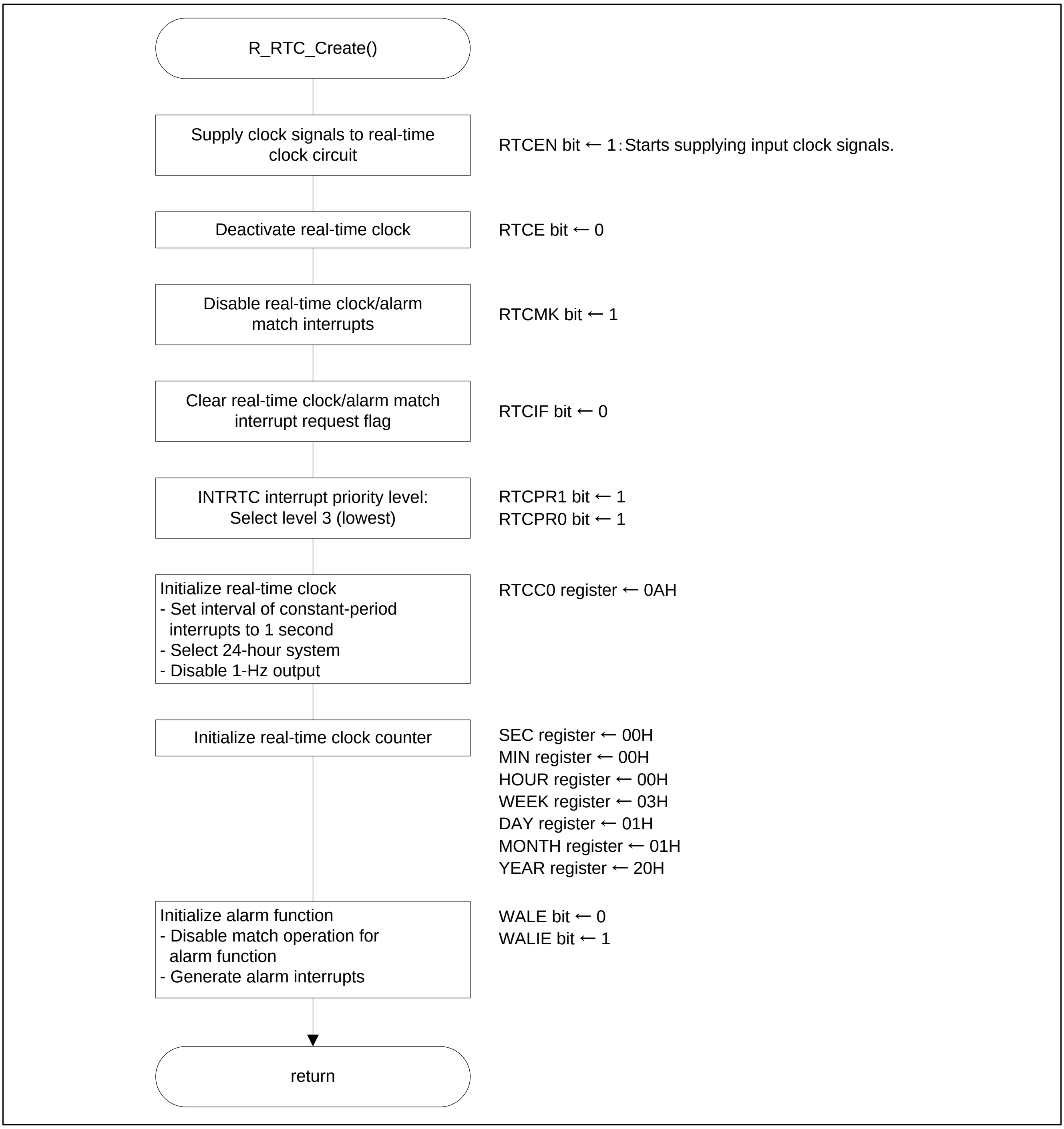

Figure 5.9 RTC Initial Setting

Note: For details on the procedure for the initial setting of the RTC (R_RTC_Create ()), refer to the section entitled "Flowcharts" in RL78/G13 Real-Time Clock Application Note (R01AN2590E).

5.7.7 12-Bit Interval Timer Initial Setting

Figure 5.10 shows the flowchart for the initial setting of the 12-bit interval timer.

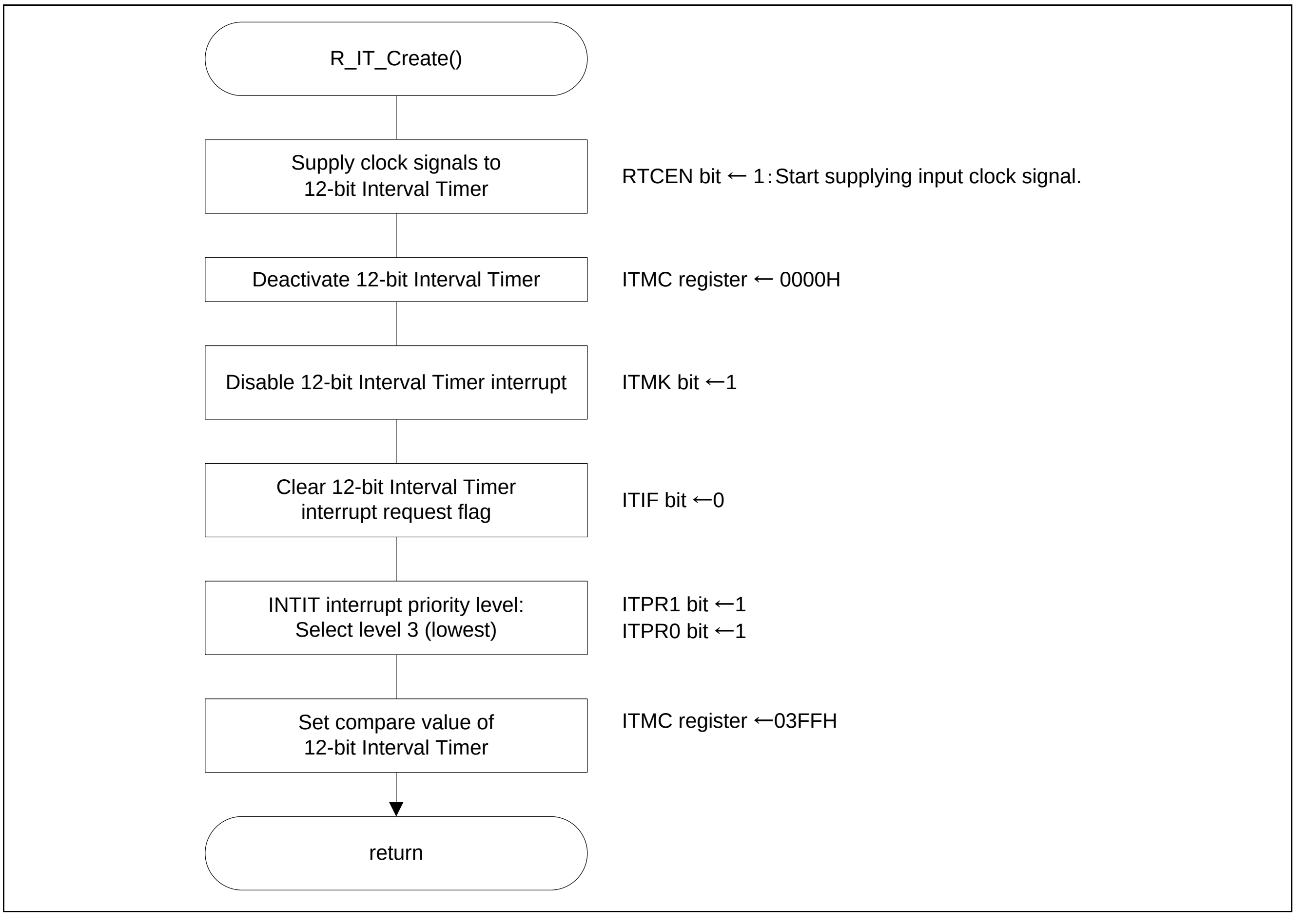

Figure 5.10 12-Bit Interval Timer Initial Setting

Enabling clock supply to RTC

- Peripheral enable register 0 (PER0)

  Enable clock supply to the 12-bit interval timer.

  Symbol: PER0

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

Bit 7

RTCEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply.
1	**Enables input clock supply.**

Controlling 12-bit interval timer operation

- 12-bit interval timer control register (ITMC)

  Set the interval period of the 12-bit interval timer to 31.25 ms.

  Symbol: ITMC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RINTE	0	0	0	ITCMP11-ITCMP0											
0	0	0	0	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Bit 15

RINTE	12-bit interval timer operation control
0	**Count operation stopped (count clear)**
1	Count operation started

Bit 11 to 0

ITCMP11-ITCMP0
Set 12-bit interval timer compare value

Note: For details of register settings, refer to the RL78/G1F User's Manual: Hardware.

5.7.8 Initial Setting of Buzzer Output Controller

Figure 5.11 shows the flowchart for the initial setting of the buzzer output controller.

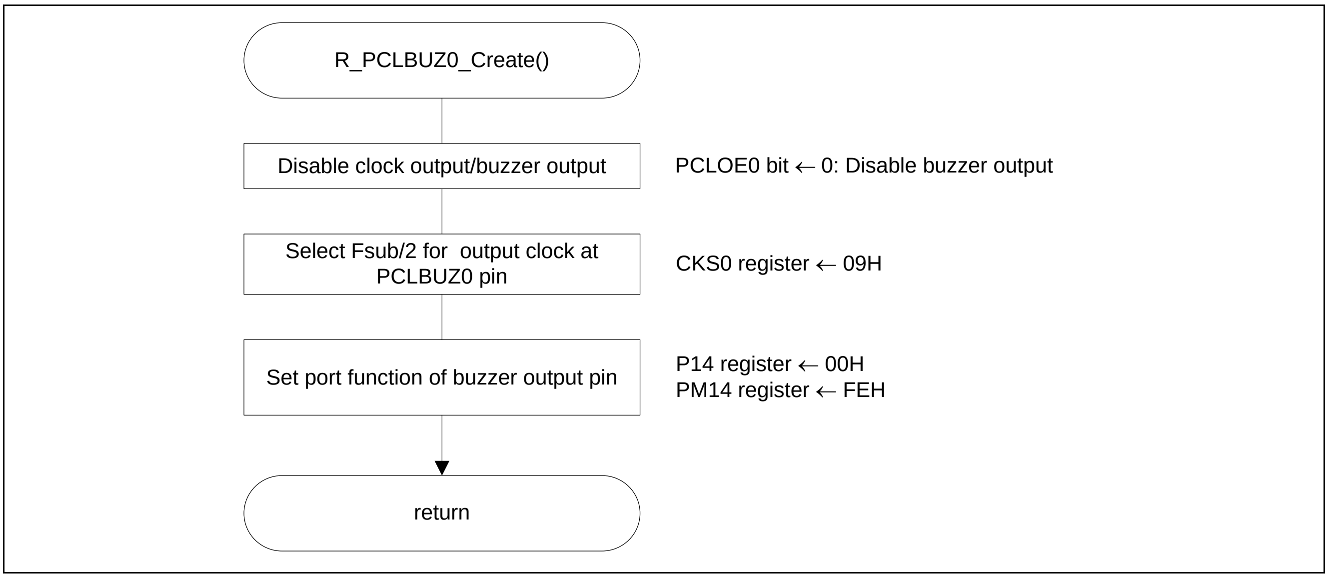

Figure 5.11 Initial Setting of Buzzer Output Controller

Controlling buzzer output

- Clock output select register 0 (CKS0)

Symbol: CKS0

7	6	5	4	3	2	1	0
PCLOE0	0	0	0	CSEL0	CCS02	CCS01	CCS00
0	0	0	0	**1**	**0**	**0**	**1**

Bit 7

PCLOE0	PCLBUZ0 pin output enable/disable specification
0	**Output disable (default)**
1	Output enable

Bit 3 to 0

CSEL0	CCS02	CCS01	CCS00	PCLBUZ0 pin output clock selection
0	0	0	0	f_{MAIN}
0	0	0	1	$f_{MAIN}/2$
:	:	:	:	:
1	0	0	0	f_{sub}
1	**0**	**0**	**1**	**$f_{sub}/2$ (16.384 kHz)**
:	:	:	:	:
1	1	1	1	$f_{sub}/2^7$(256Hz)

Setting buzzer output pin

- Port register (P14)

Set the port output data value.

Symbol: P14

7	6	5	4	3	2	1	0
P147	P146	0	0	0	0	P141	P140
1	**1**	0	0	0	0	**1**	**0**

Bit 0

P140	Output data control (in output mode)	Input data read (in input mode)
0	**Output 0**	**Input low level**
1	Output 1	Input high level

- Port mode register (PM14)

Set the pin I/O mode.

Symbol: PM14

7	6	5	4	3	2	1	0
PM147	PM146	0	0	0	0	PM141	PM140
1	**1**	0	0	0	0	**1**	**0**

Bit 0

PM140	P140 pin I/O mode selection
0	**Output mode (output buffer on)**
1	Input mode (output buffer off)

5.7.9 SAU0 Setup

Figure 5.12 shows the flowchart for SAU0 setup.

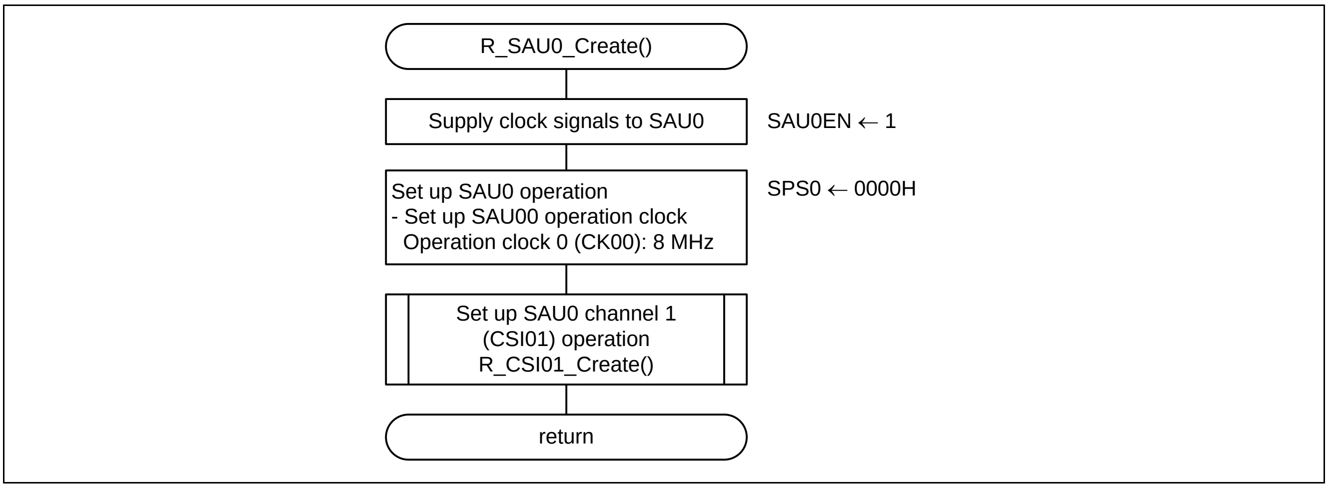

Figure 5.12 SAU0 Setup

Note1: For details on the procedure for the initial setting of the SAU0 (R_SAU0_Create ()), refer to the section entitled "Flowcharts" in RL78/G13 Serial Array Unit for 3-Wire Serial I/O (Master Transmission/Reception) Application Note (R01AN2547E).

Note 2: For details of register settings, refer to the RL78/G1F User’s Manual: Hardware.

5.7.10 SAU0 Channel 1 (CSI01) Operation Setup

Figure 5.13 shows the flowchart for setting up SAU0 channel 1 (CSI01) operation.

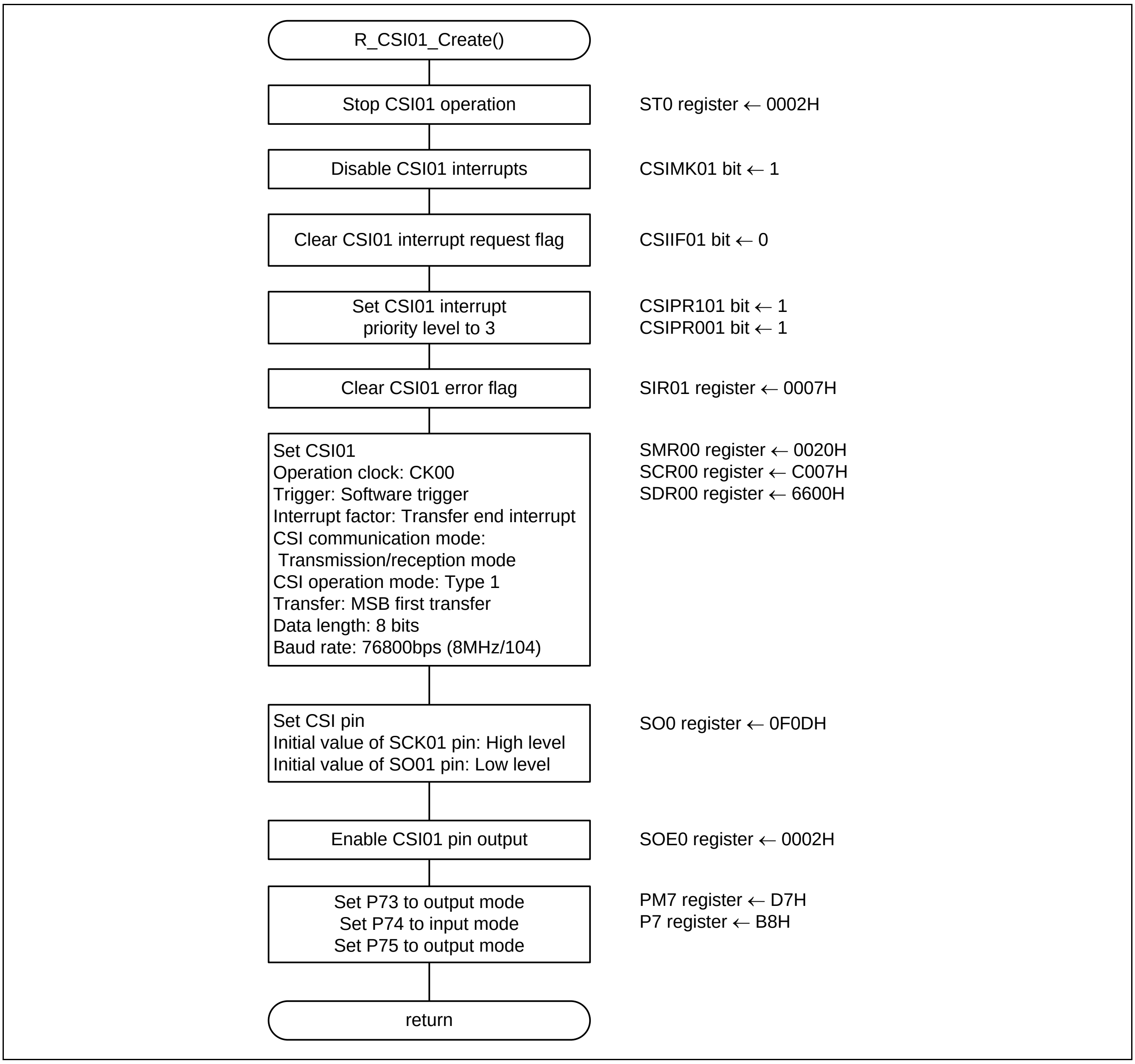

Figure 5.13 SAU0 Channel 1 (CSI01) Operation Setup

Note1: For details on the procedure for the initial setting of the SAU0 (R_SAU0_Create ()), refer to the section entitled "Flowcharts" in RL78/G13 RL78/G13 Serial Array Unit for 3-Wire Serial I/O (Master Transmission/Reception) Application Note (R01AN2547E).

Note 2: For details of register settings, refer to the RL78/G1F User’s Manual: Hardware.

5.7.11 Initial Setting of External Interrupts

Figure 5.14 shows the flowchart for the initial setting of external interrupts.

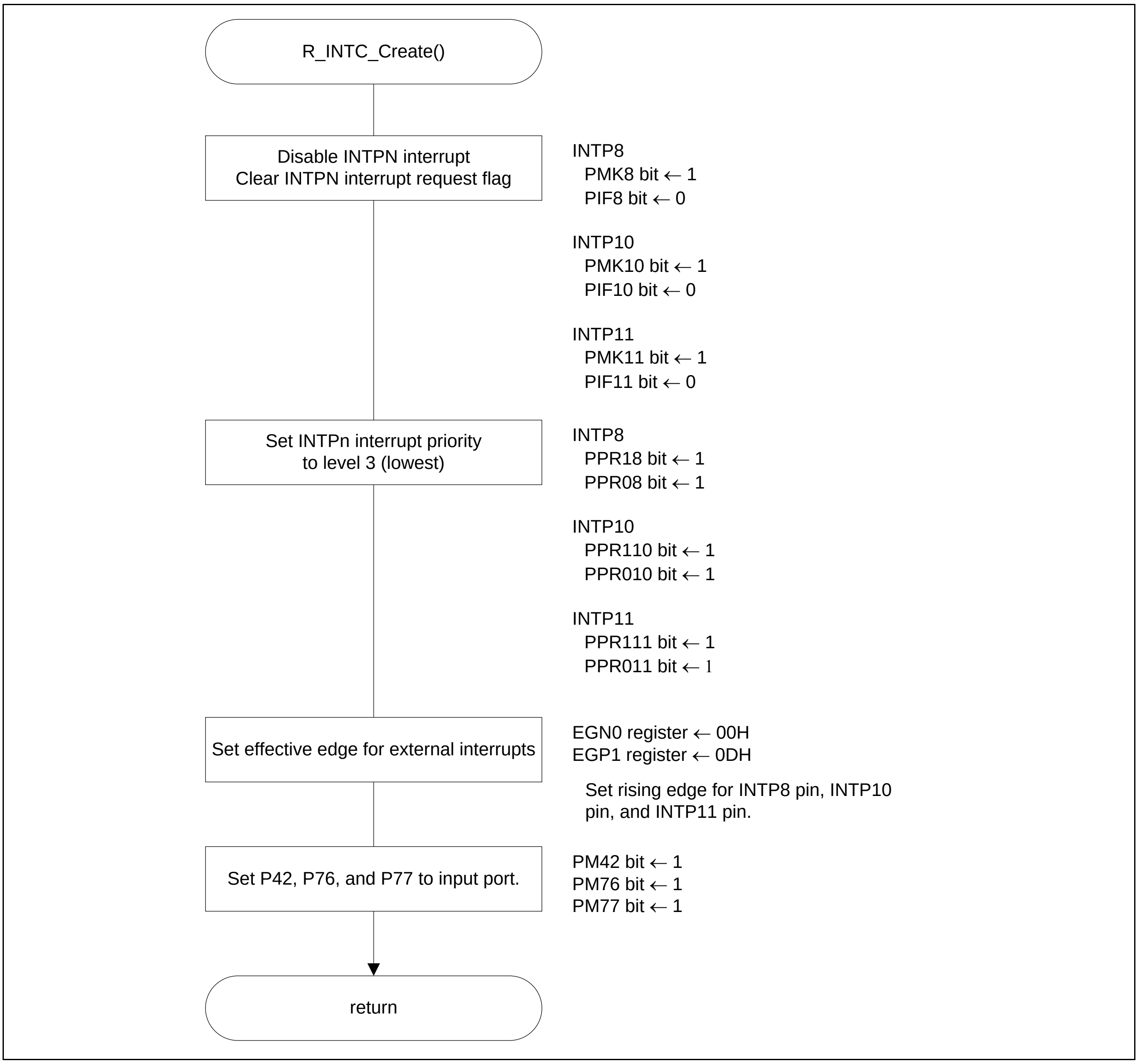

Figure 5.14 Initial Setting of External Interrupts

5.7.12 Main Processing

Figure 5.15 shows the flowchart for main processing.

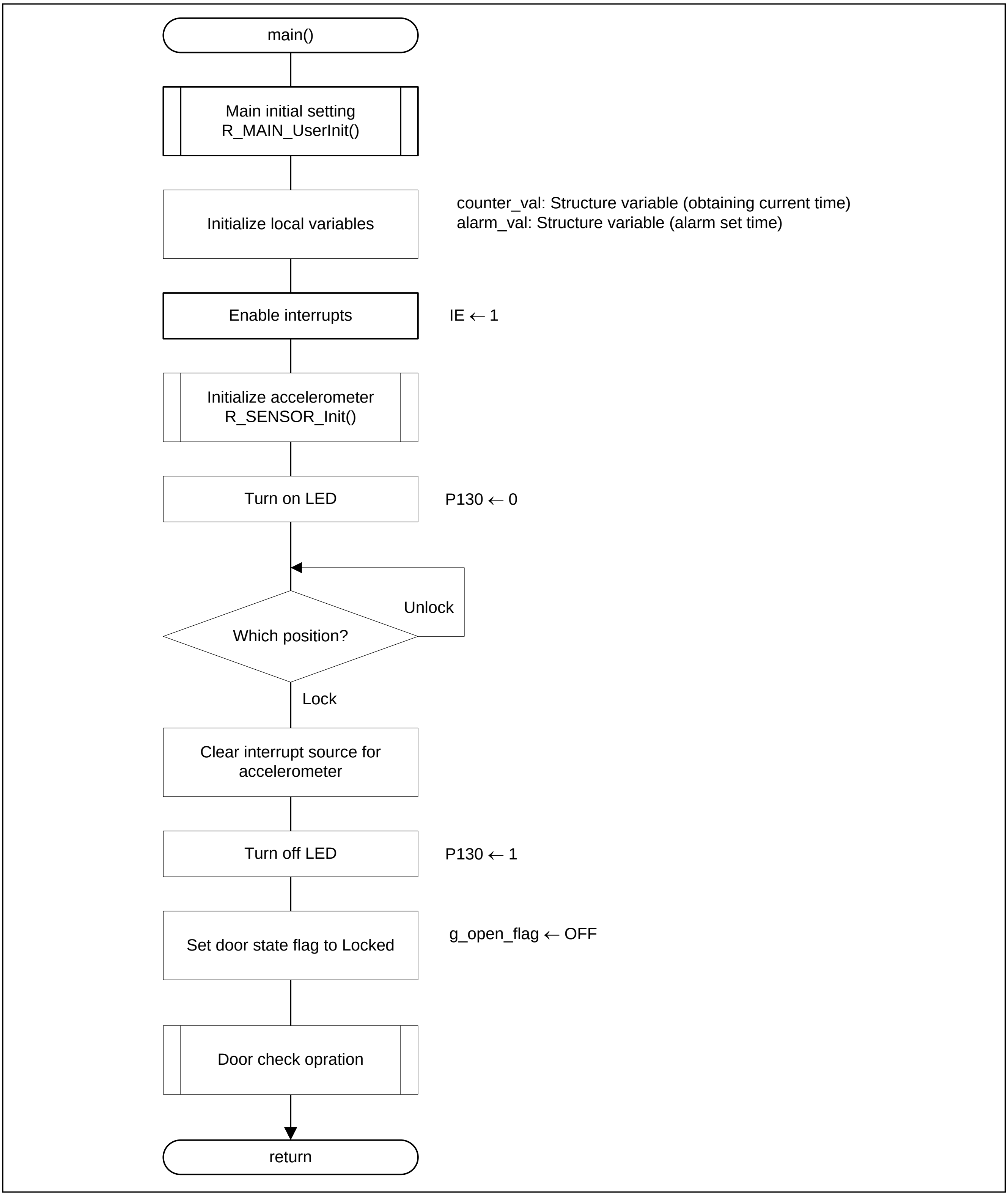

Figure 5.15 Main Processing

5.7.13 Main Initialization Setting

Figure 5.16 shows the flowchart for the main initialization settings.

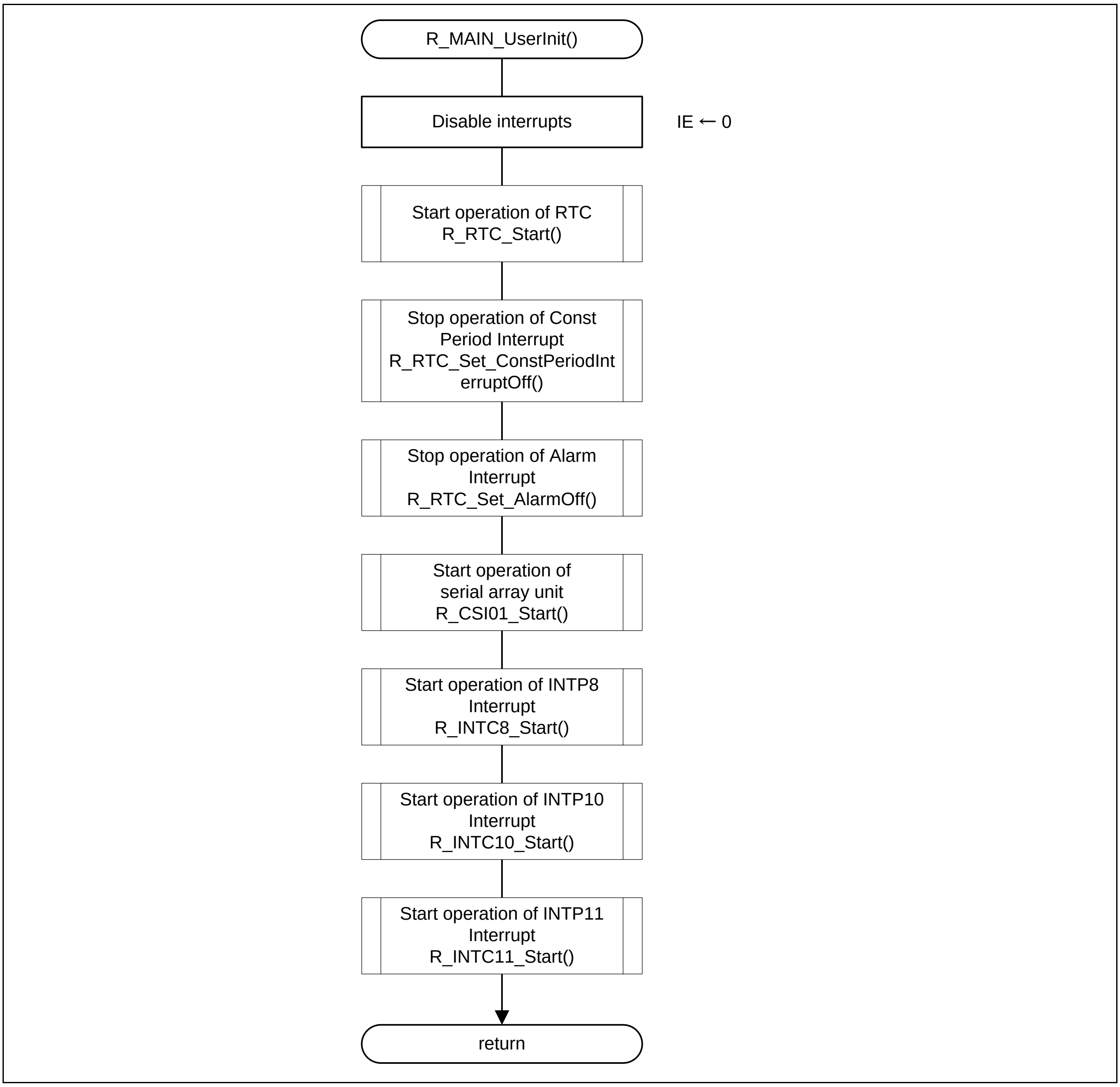

Figure 5.16 Main initialization setting

5.7.14 RTC Operation Startup

Figure 5.17 shows the flowchart for starting the RTC operation.

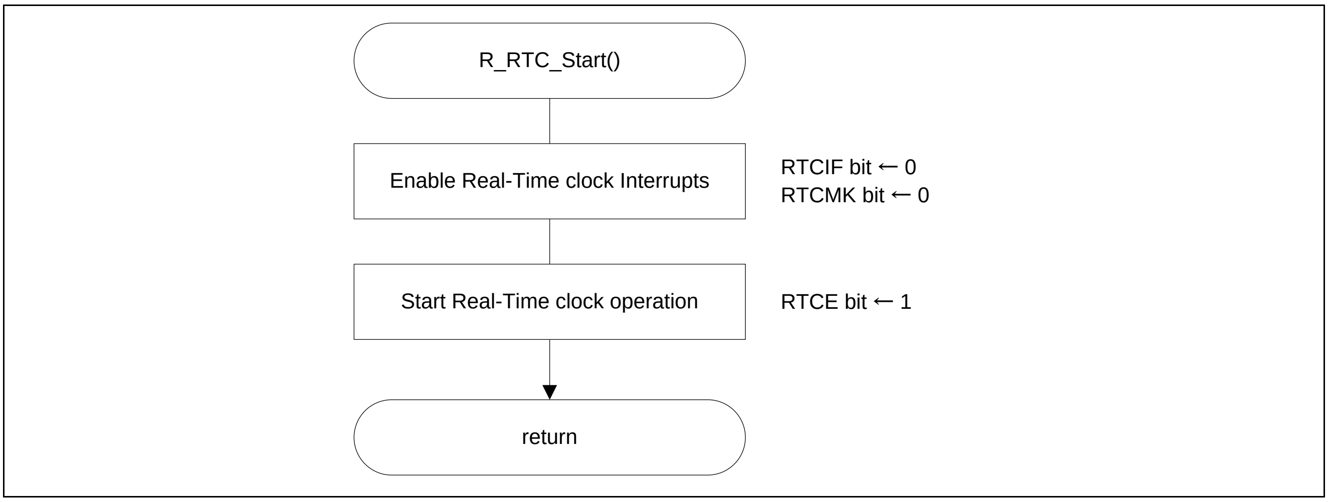

Figure 5.17 RTC Operation Startup

5.7.15 Disabling Constant-Period Interrupt

Figure 5.18 shows the flowchart for disabling the RTC constant-period interrupt.

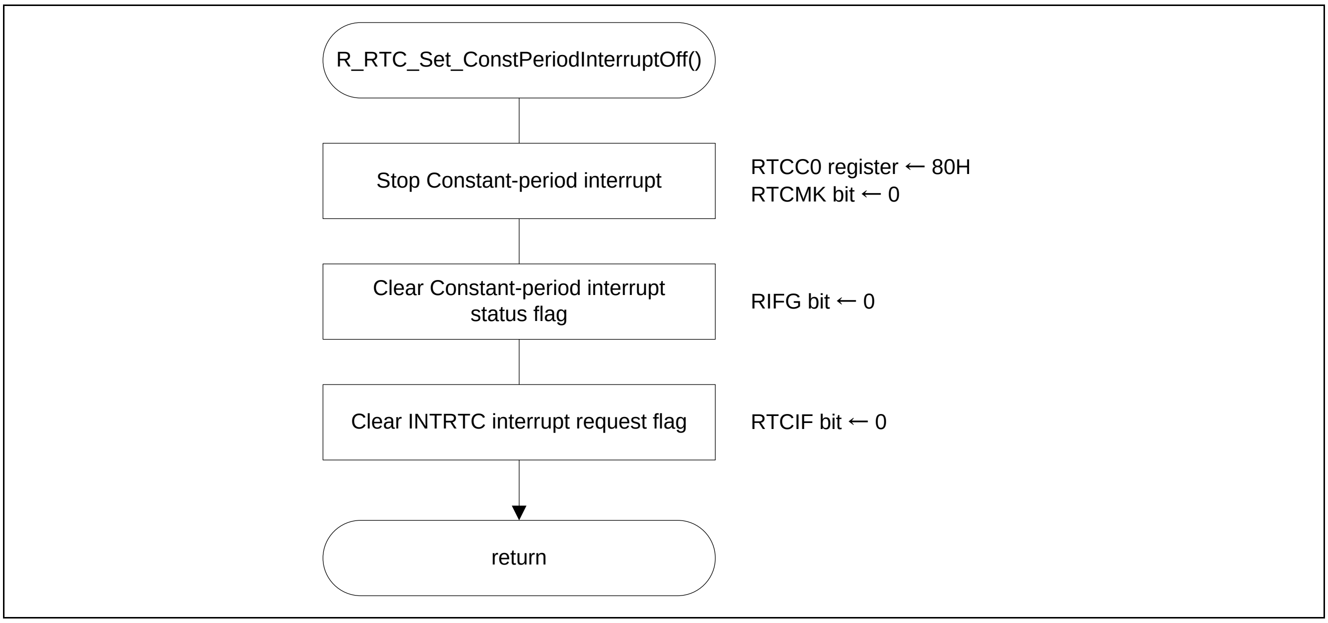

Figure 5.18 Disabling RTC Constant-Period Interrupt

5.7.16 Disabling Alarm Interrupt

Figure 5.19 shows the flowchart for disabling the alarm interrupt.

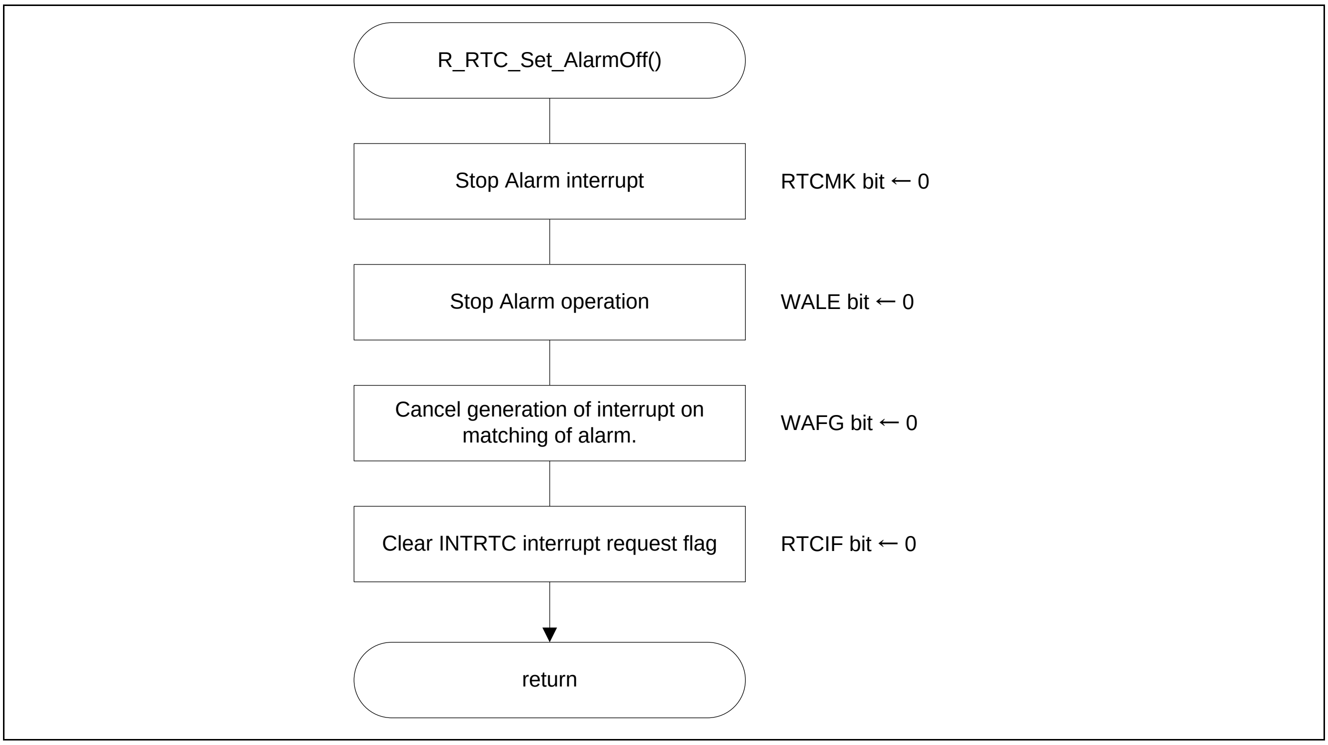

Figure 5.19 Disabling Alarm Interrupt

5.7.17 SAU0 Channel 1 (CSI01) Startup

Figure 5.20 shows the flowchart for starting the operation of SAU0 channel 1 (CSI01).

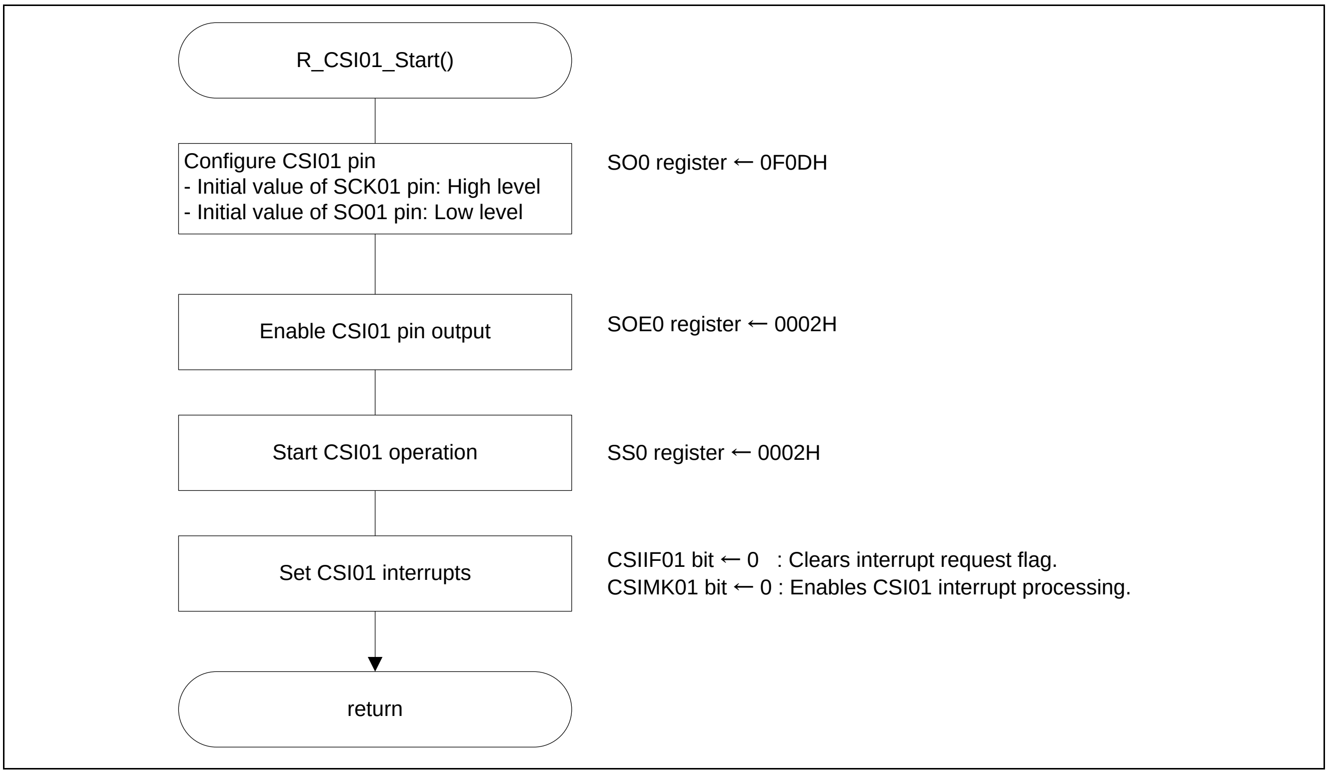

Figure 5.20 SAU0 Channel 1 (CSI01) Startup

5.7.18 INTP8 External Interrupt Enabling Function

Figure 5.21 shows the flowchart for the INTP8 external interrupt enabling function.

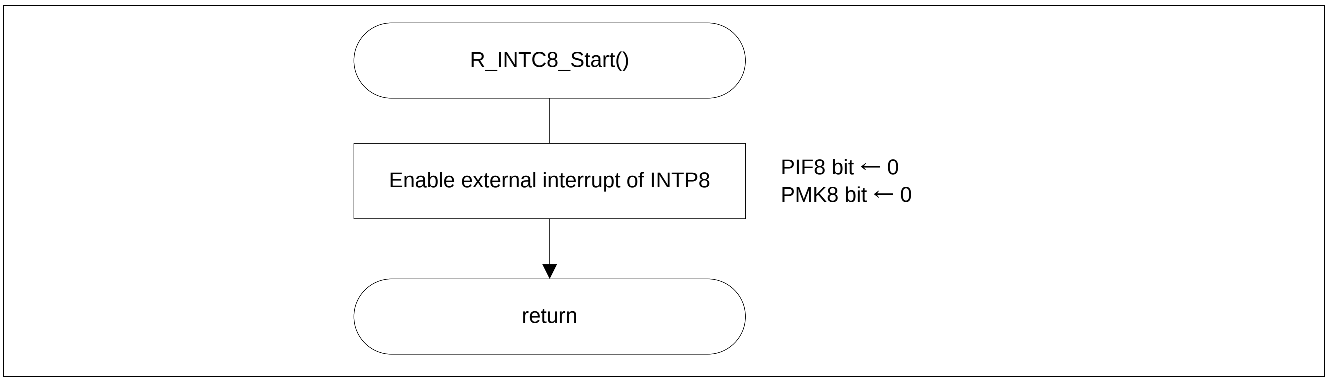

Figure 5.21 INTP8 External Interrupt Enabling Function

5.7.19 INTP10 External Interrupt Enabling Function

Figure 5.22 shows the flowchart for the INTP10 external interrupt enabling function.

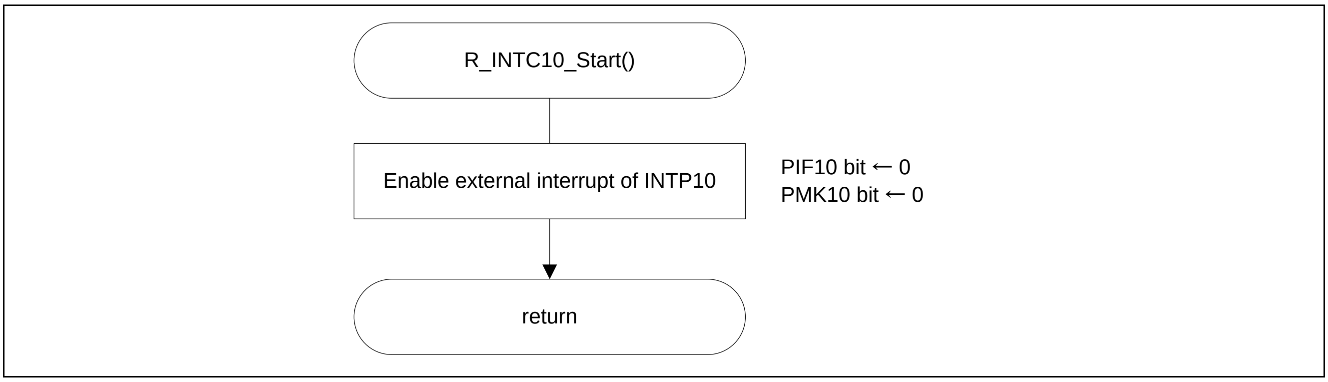

Figure 5.22 INTP10 External Interrupt Enabling Function

5.7.20 INTP11 External Interrupt Enabling Function

Figure 5.23 shows the flowchart for the INTP11 external interrupt enabling function.

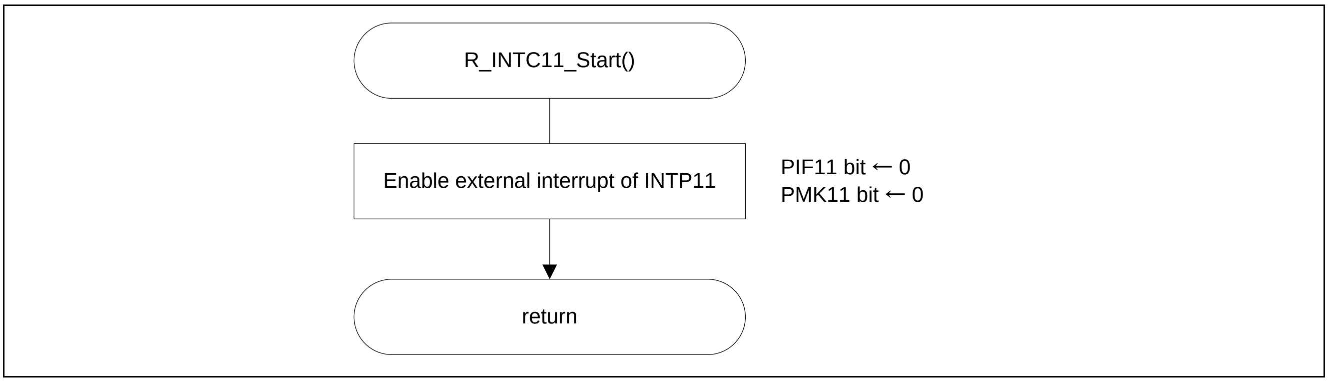

Figure 5.23 INTP11 External Interrupt Enabling Function

5.7.21 Accelerometer Initialization Function

Figure 5.24 shows the flowchart for initializing the accelerometer.

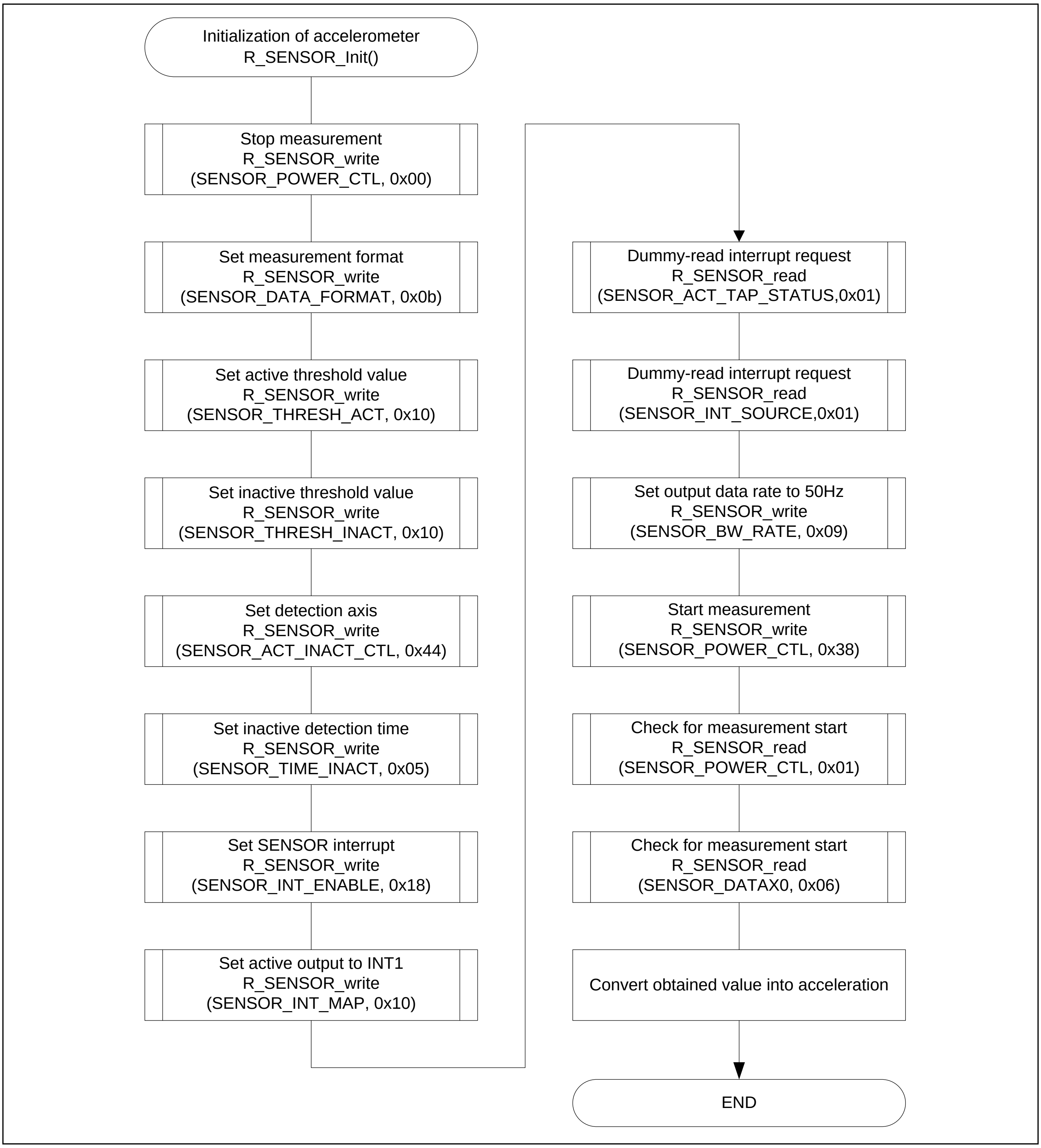

Figure 5.24 Accelerometer Initialization

5.7.22 Accelerometer Data Write

Figure 5.25 shows the flowchart for the accelerometer data write.

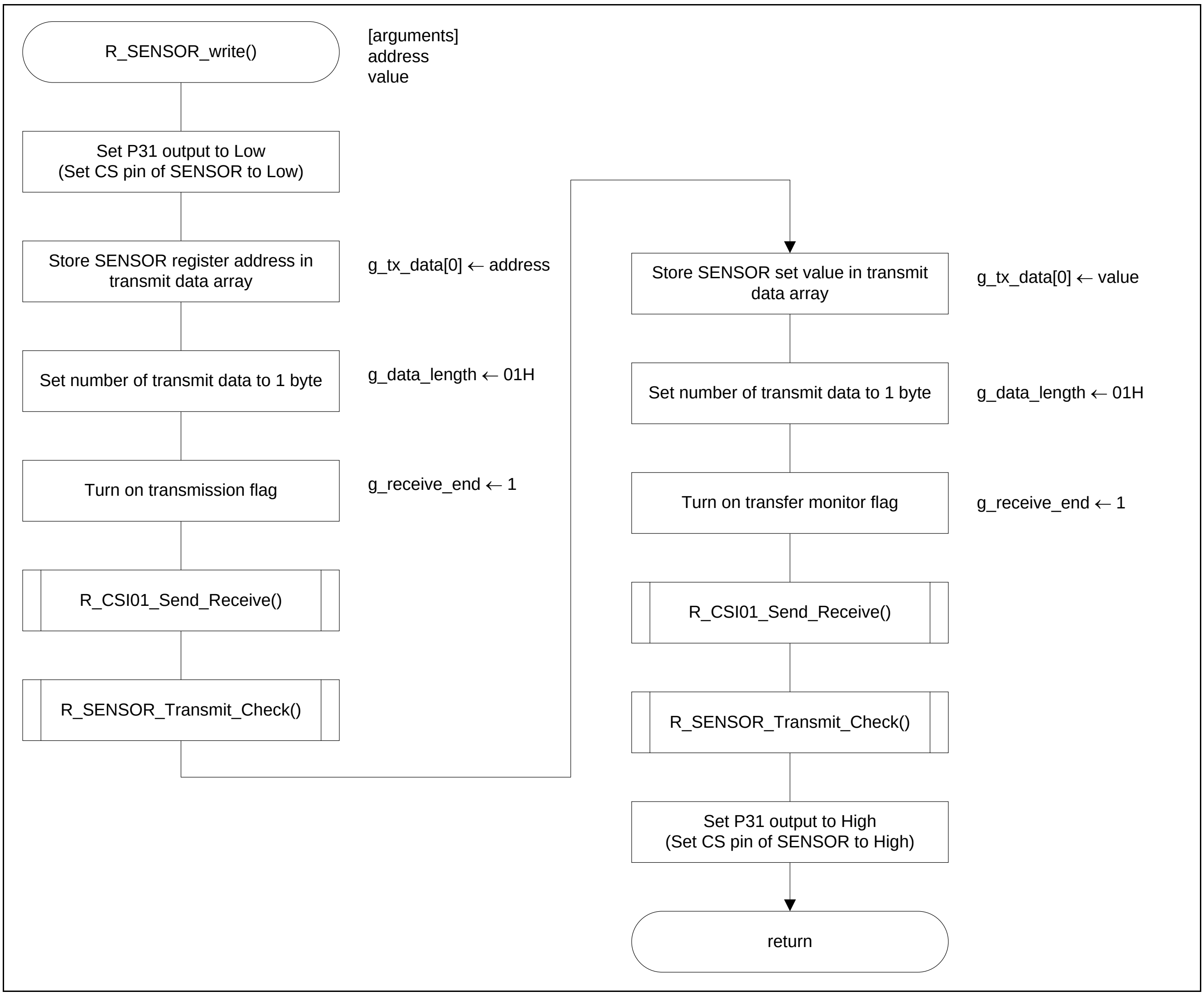

Figure 5.25 Accelerometer Data Write

5.7.23 Accelerometer Data Read

Figure 5.26 shows the flowchart for the accelerometer data read.

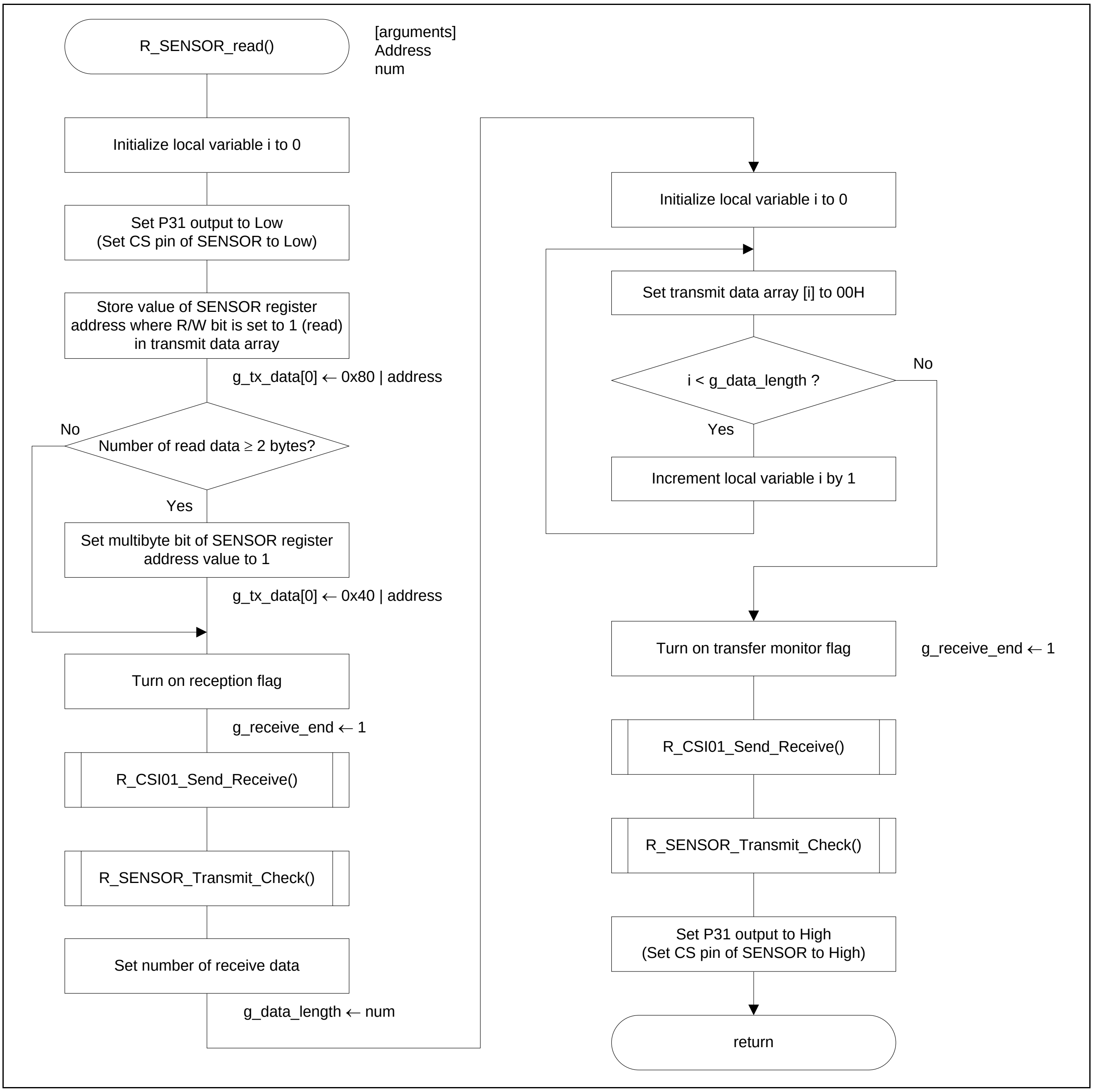

Figure 5.26 Accelerometer Data Read

5.7.24 CSI01 Data Transfer Start

Figure 5.27 shows the flowchart for starting the CSI01 data transfer.

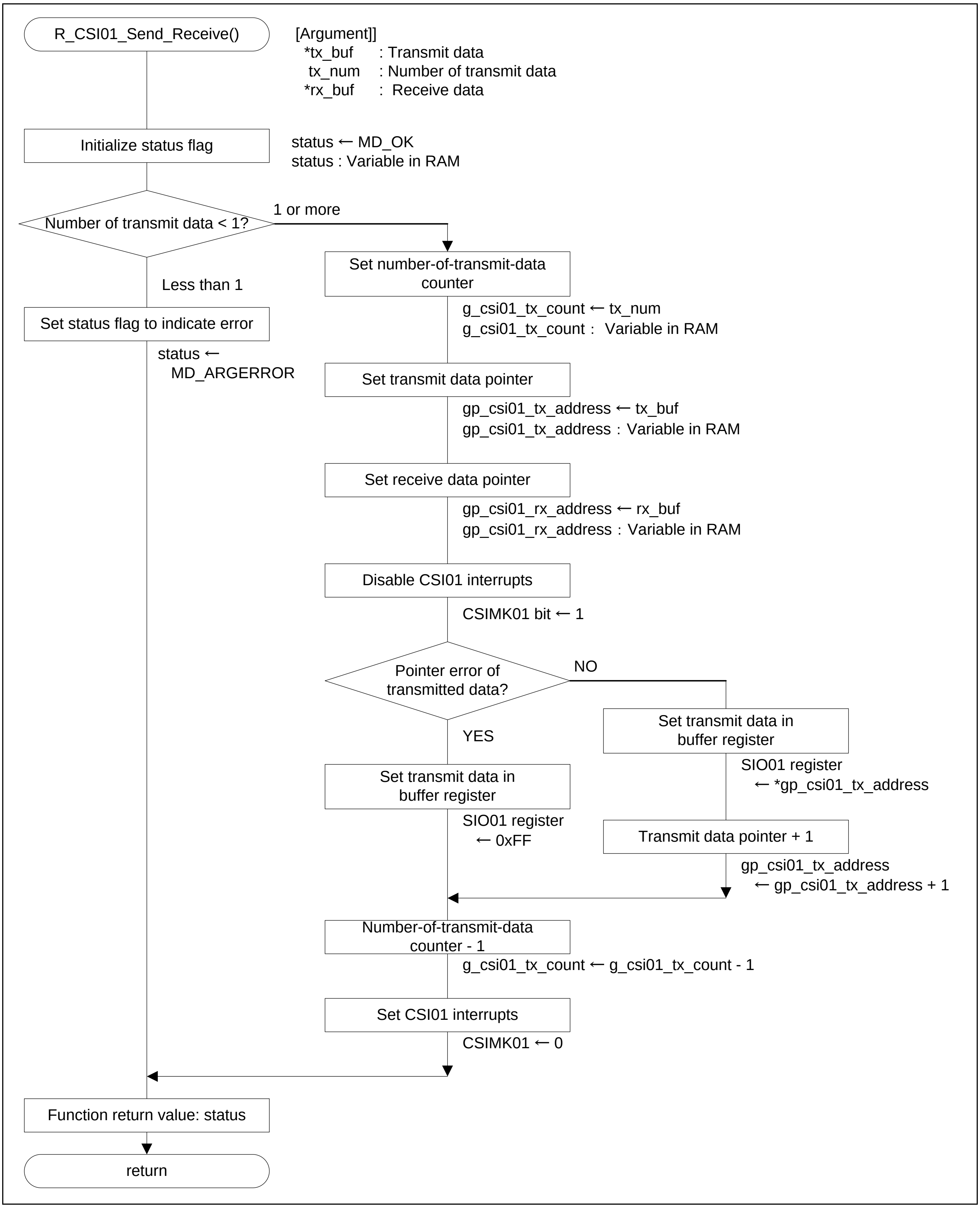

Figure 5.27 CSI01 Data Transfer Start

5.7.25 Waiting for Accelerometer Data Transfer End

Figure 5.28 shows the flowchart for waiting for the accelerometer data transfer end.

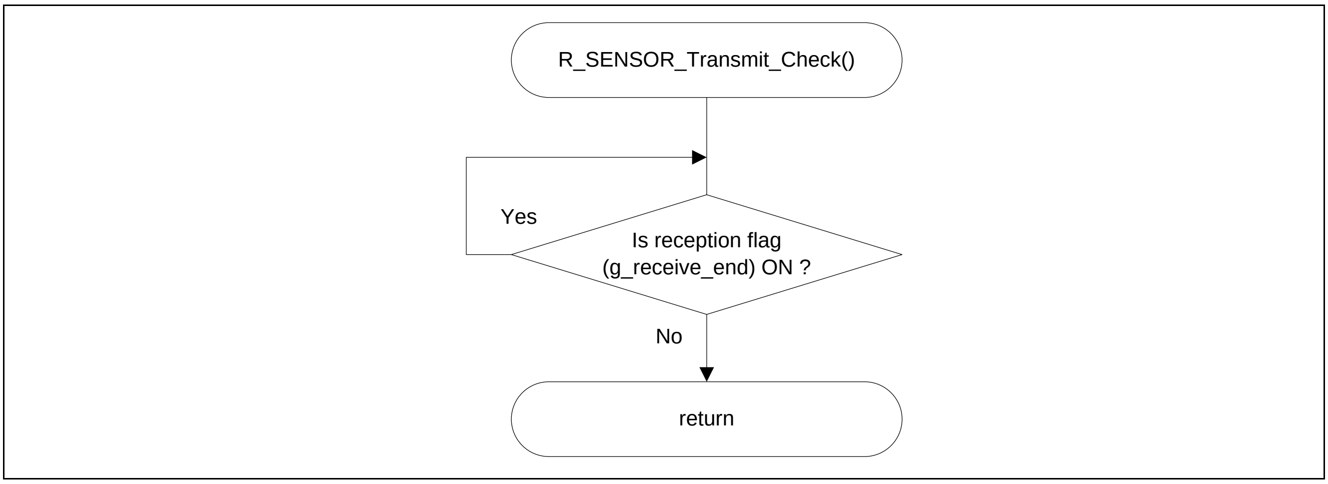

Figure 5.28 Waiting for Accelerometer Data Transfer End

5.7.26 Door Check Operation

Figure 5.29 to Figure 5.32 show the flowcharts for the door check operation.

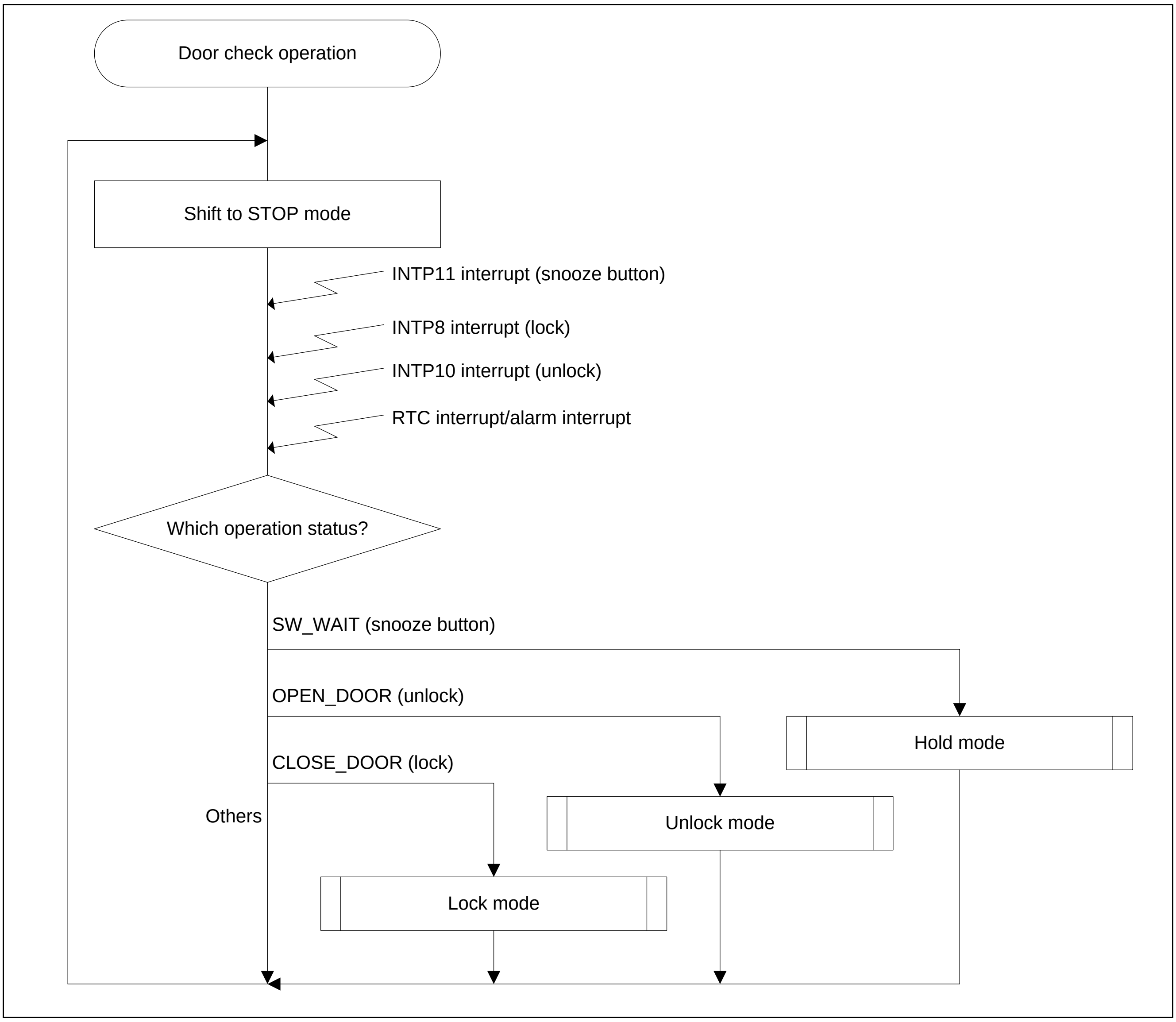

Figure 5.29 Door Check Operation (1/4)

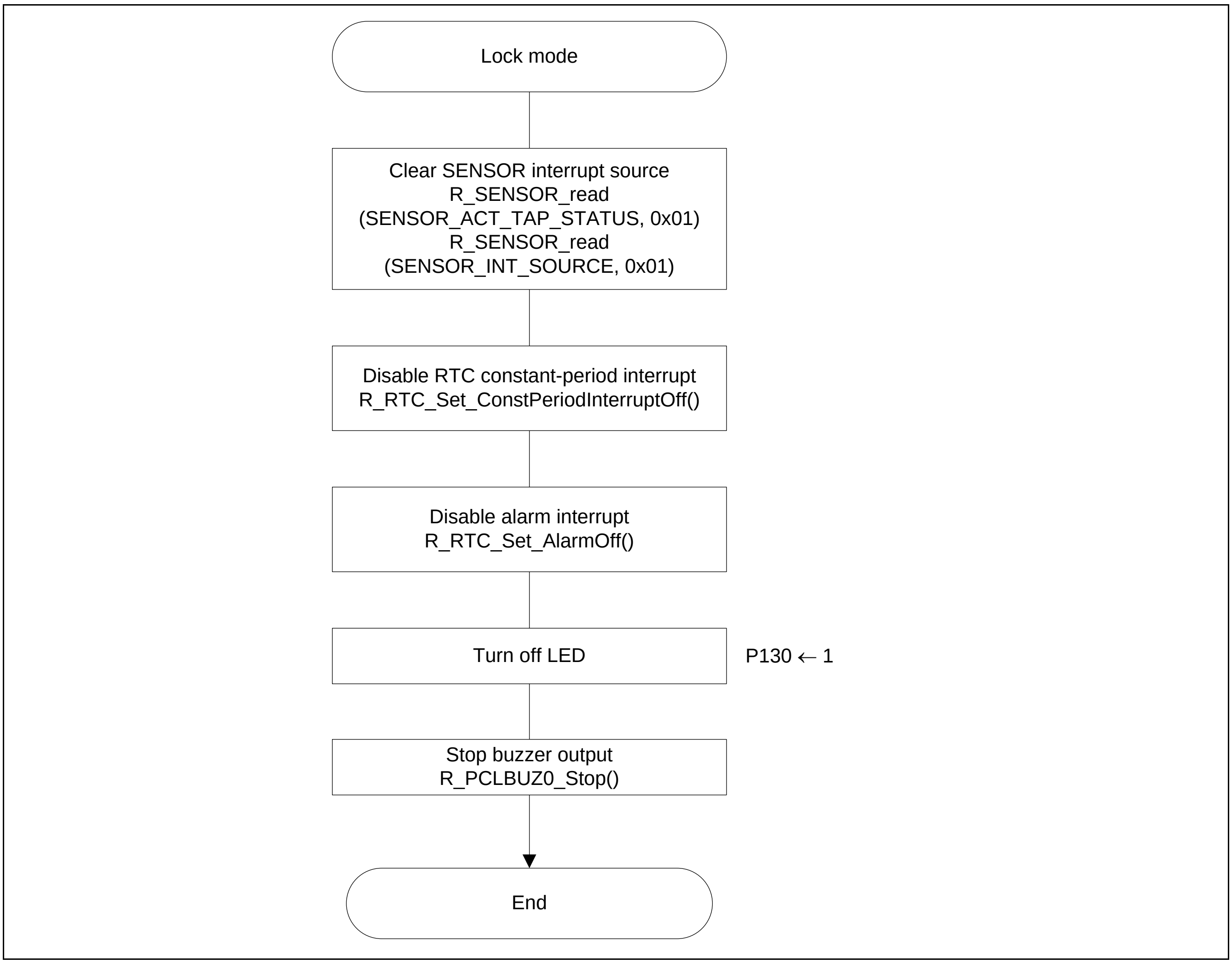

Figure 5.30 Door Check Operation (2/4)

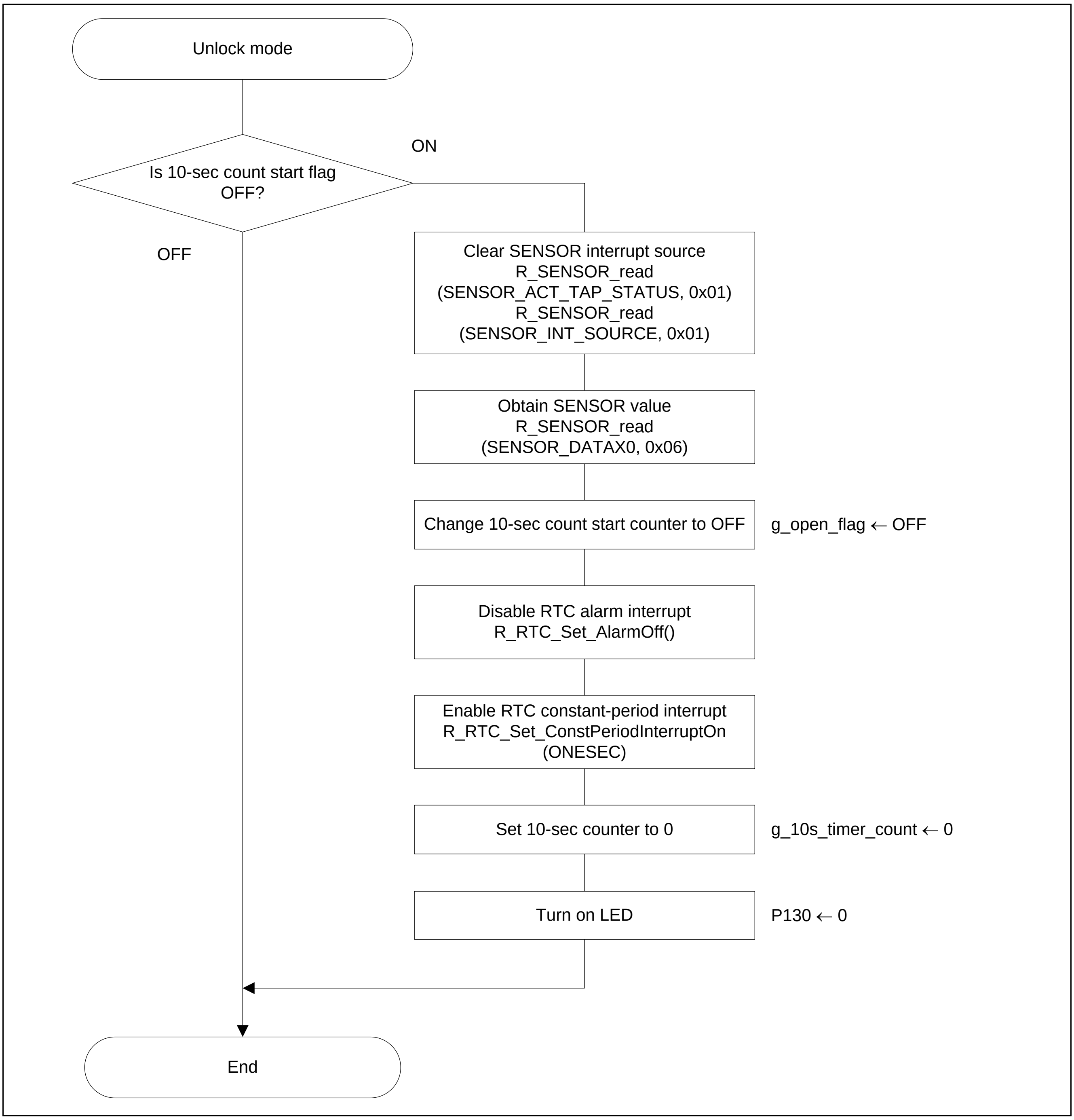

Figure 5.31 Door Check Operation (3/4)

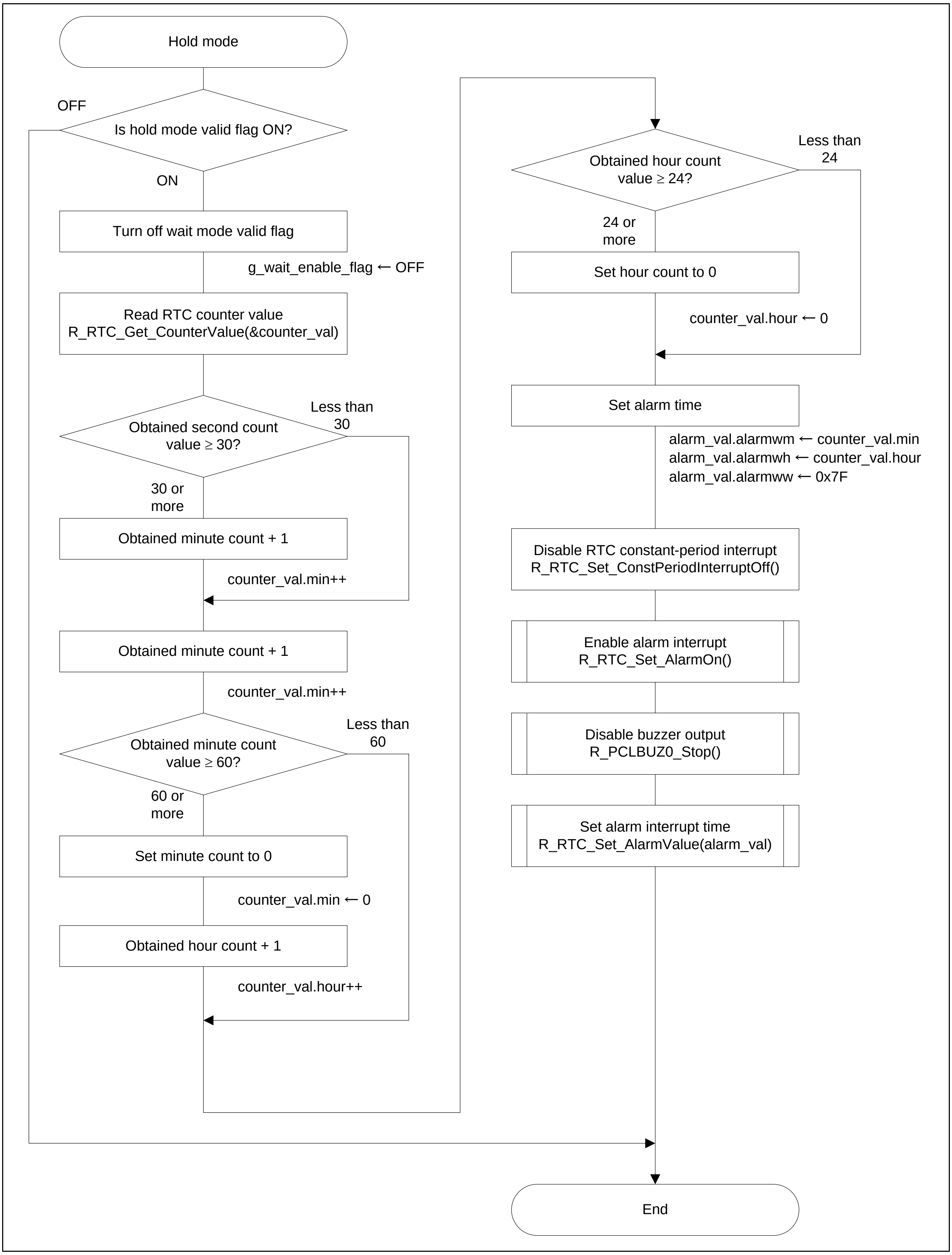

Figure 5.32 Door Check Operation (4/4)

5.7.27 CSI01 Transfer End Interrupt Processing

Figure 5.33 and Figure 5.34 show the flowcharts for the CSI01 transfer end interrupt processing.

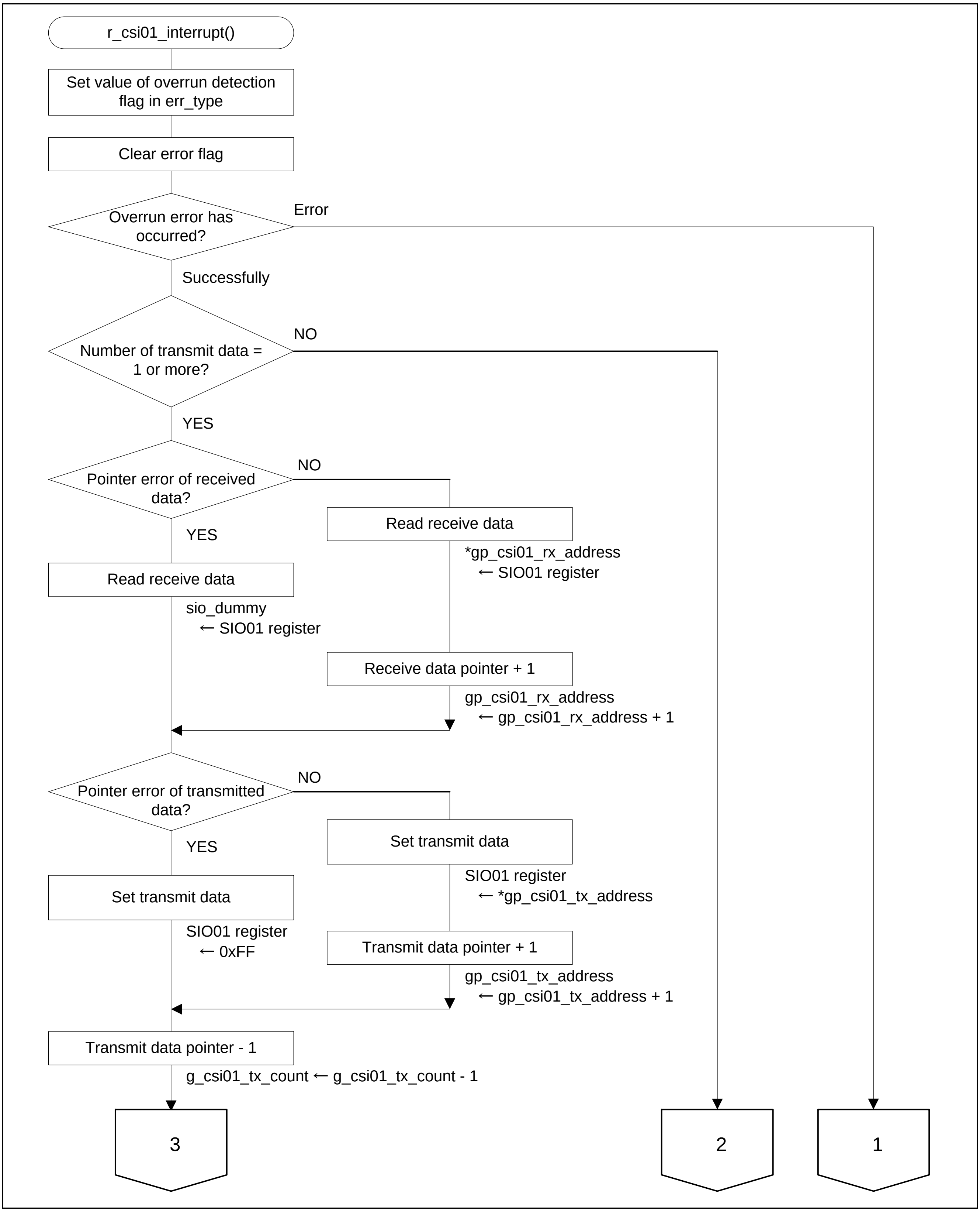

Figure 5.33 CSI01 Transfer End Interrupt Processing (1/2)

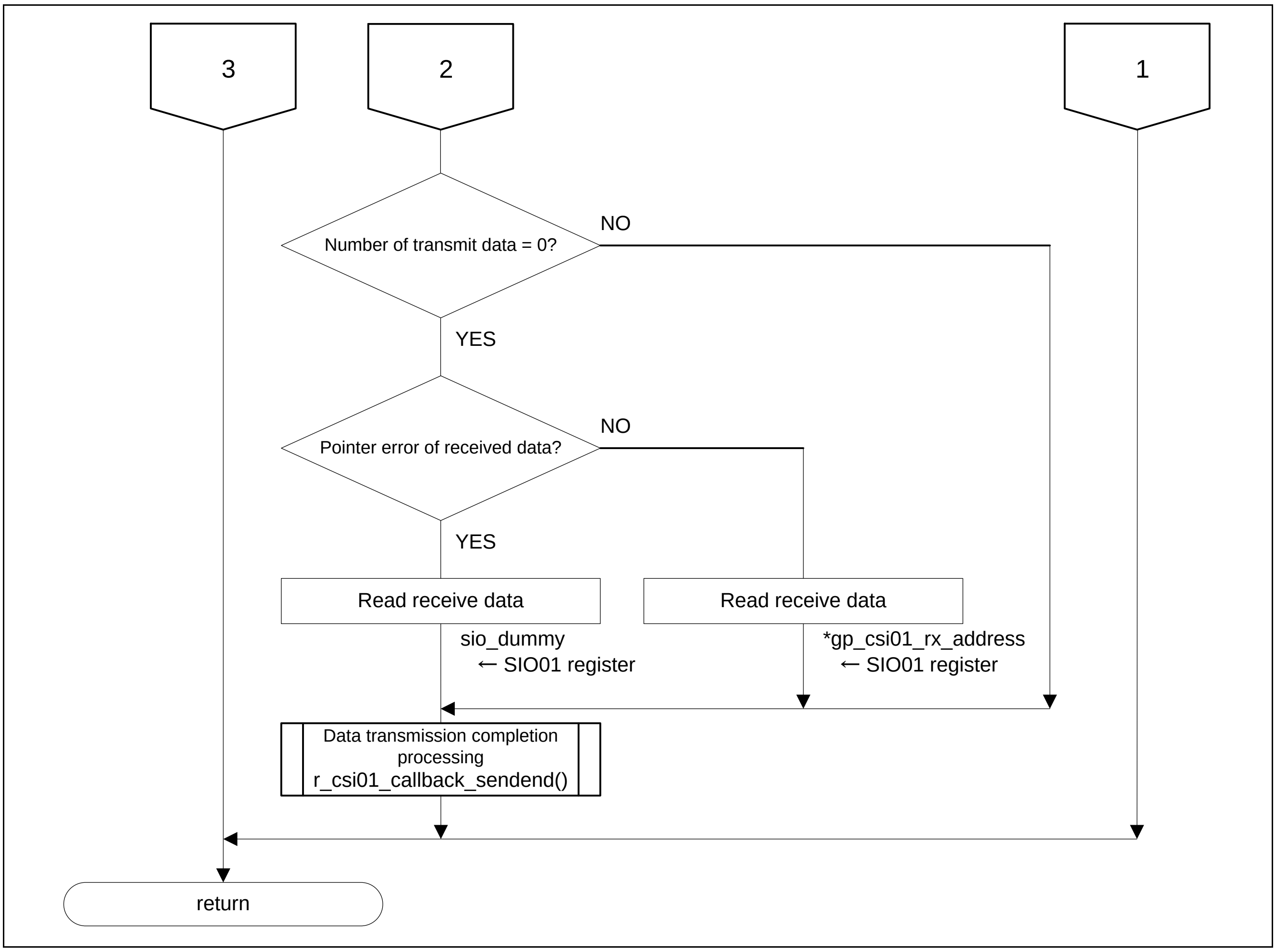

Figure 5.34 CSI01 Transfer End Interrupt Processing (2/2)

5.7.28 Data Reception End Processing

Figure 5.35 shows the flowchart for the data reception end processing.

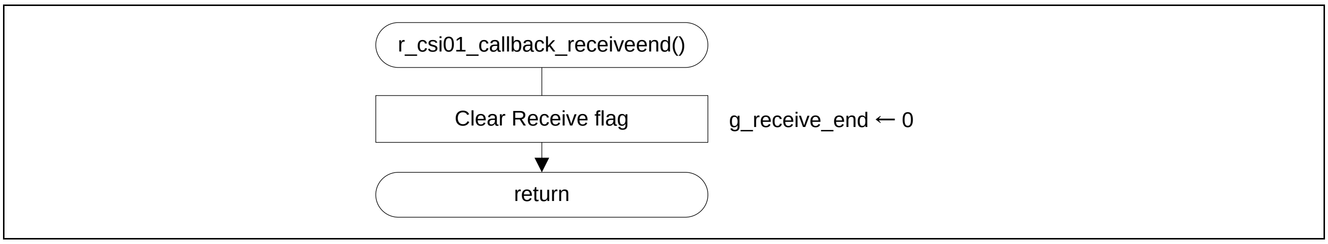

Figure 5.35 Data Reception End Processing

5.7.29 Enabling Alarm Interrupt

Figure 5.36 shows the flowchart for enabling the alarm interrupt.

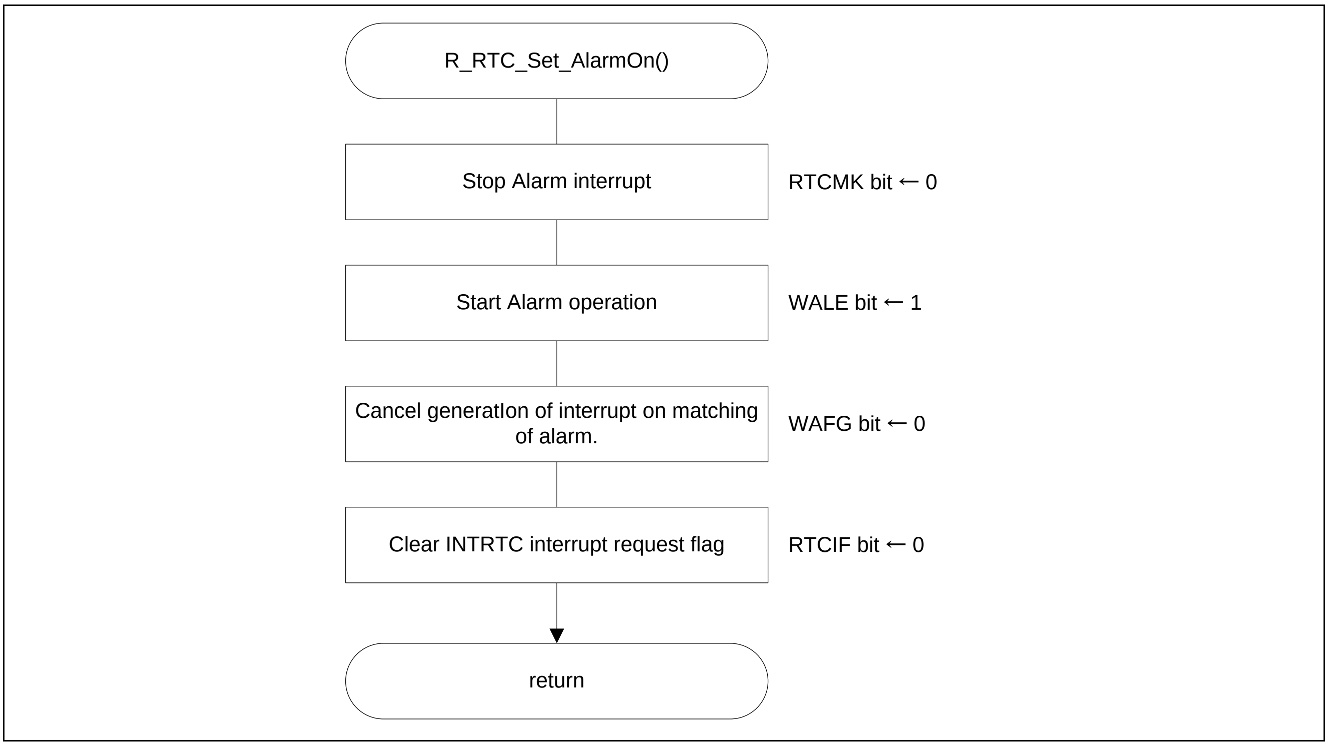

Figure 5.36 Enabling Alarm Interrupt

5.7.30 Enabling Constant-Period Interrupt

Figure 5.37 shows the flowchart for enabling the RTC constant-period interrupt.

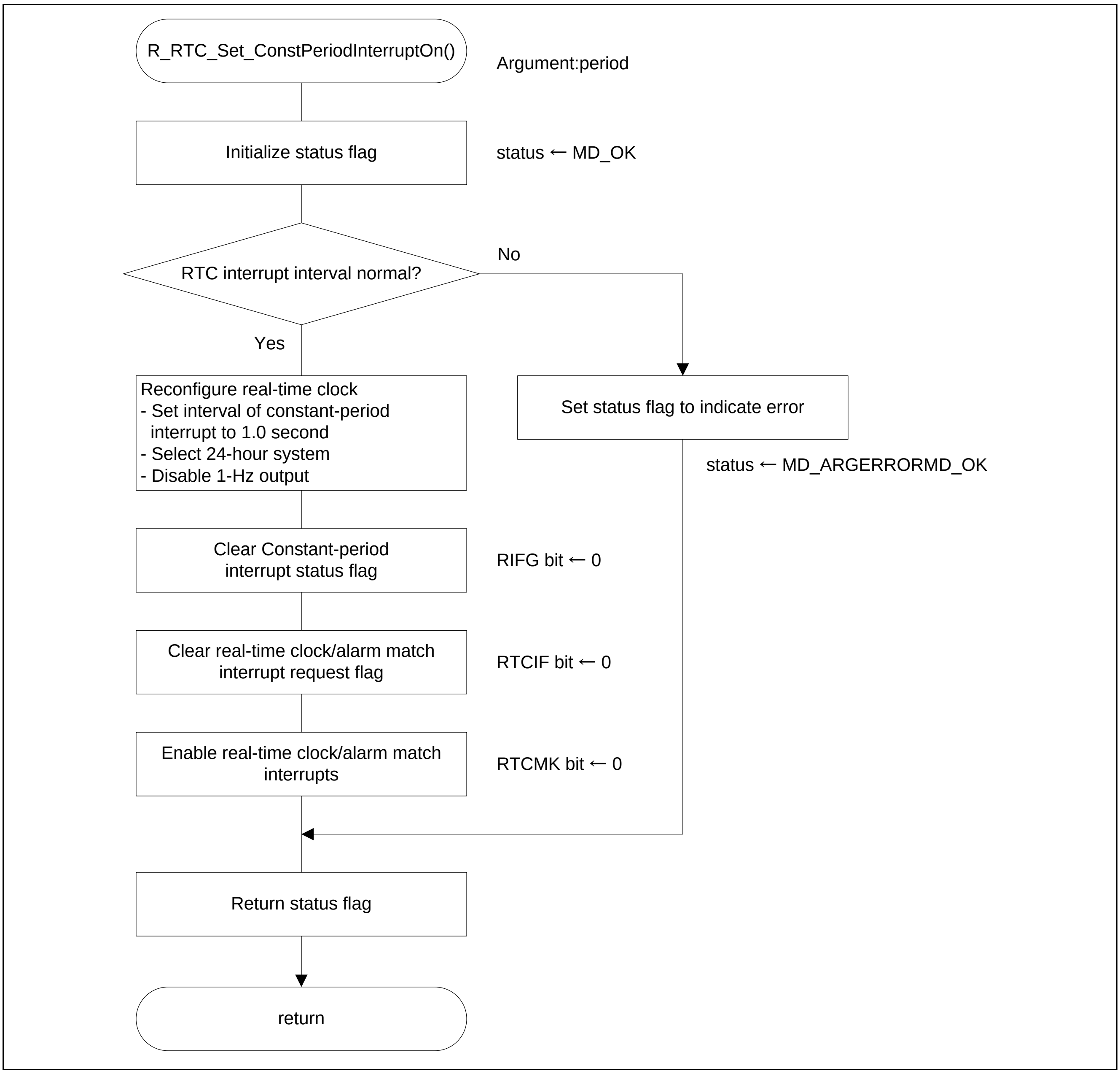

Figure 5.37 Enabling RTC Constant-Period Interrupt

5.7.31 INTP8 External Interrupt Function

Figure 5.38 shows the flowchart for INTP8 external interrupt function.

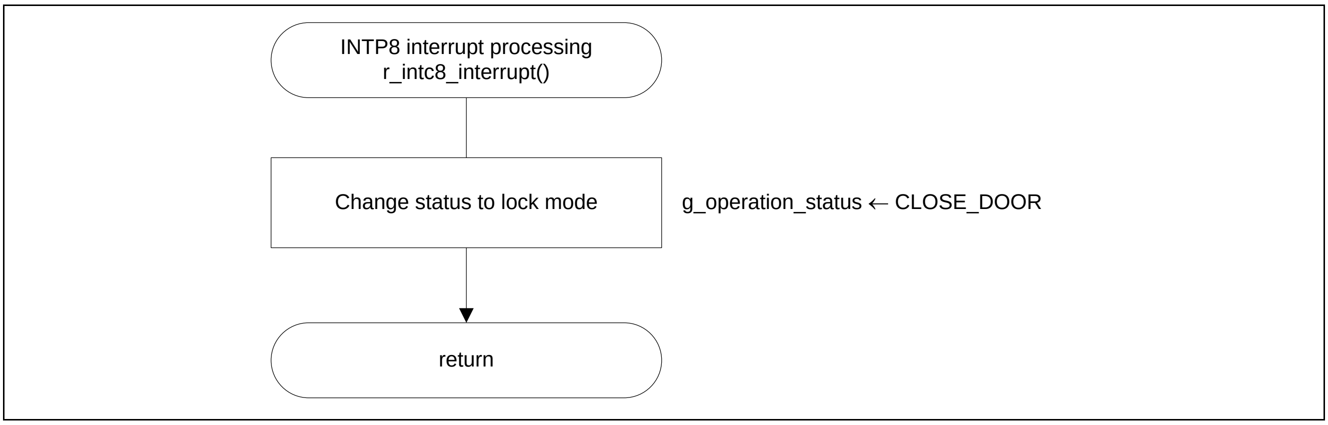

Figure 5.38 INTP8 External Interrupt Function

5.7.32 INTP10 External Interrupt Function

Figure 5.39 shows the flowchart for INTP10 external interrupt function.

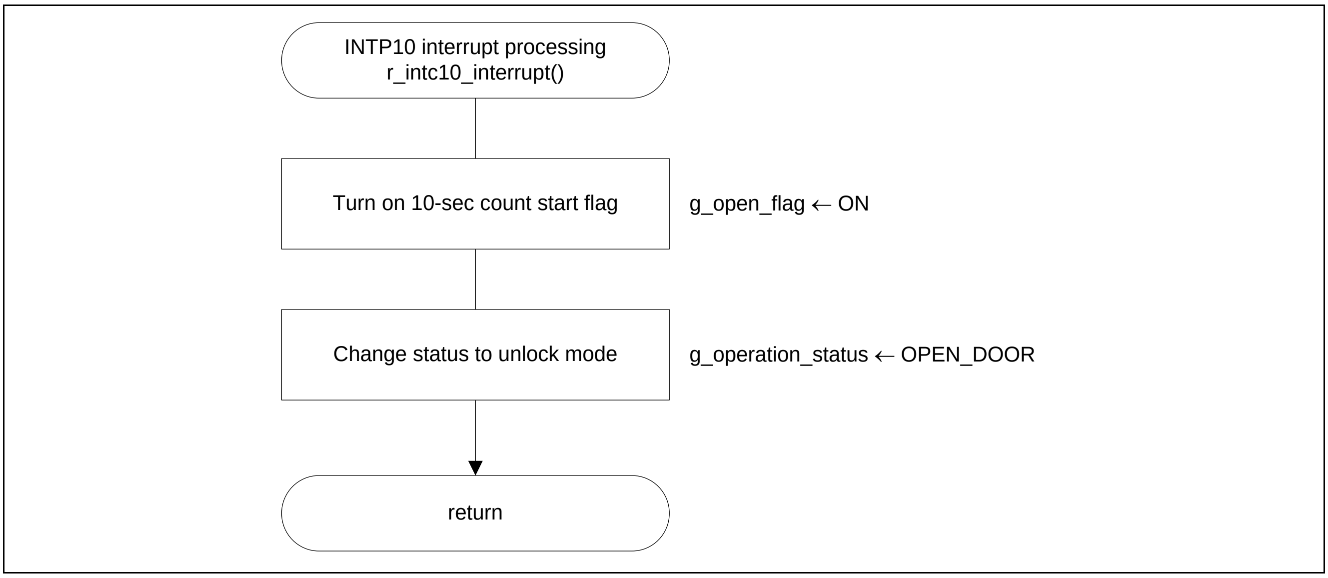

Figure 5.39 INTP10 External Interrupt Function

5.7.33 INTP11 External Interrupt Function

Figure 5.40 shows the flowchart for INTP11 external interrupt function.

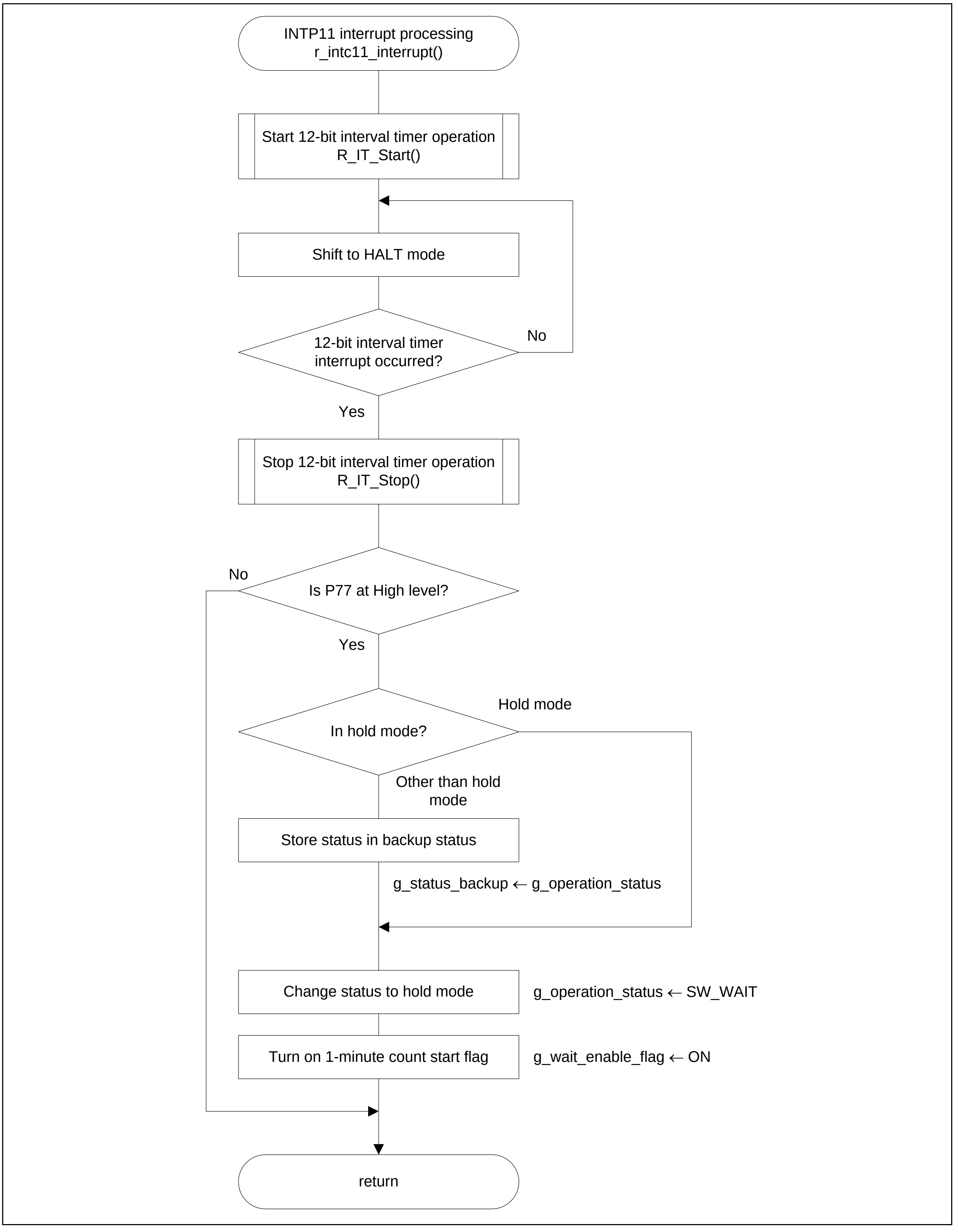

Figure 5.40 INTP11 External Interrupt Function

5.7.34 RTC Interrupt Processing

Figure 5.41 shows the flowchart for RTC interrupt processing. The constant-period interrupt and alarm interrupt are generated by the same interrupt source.

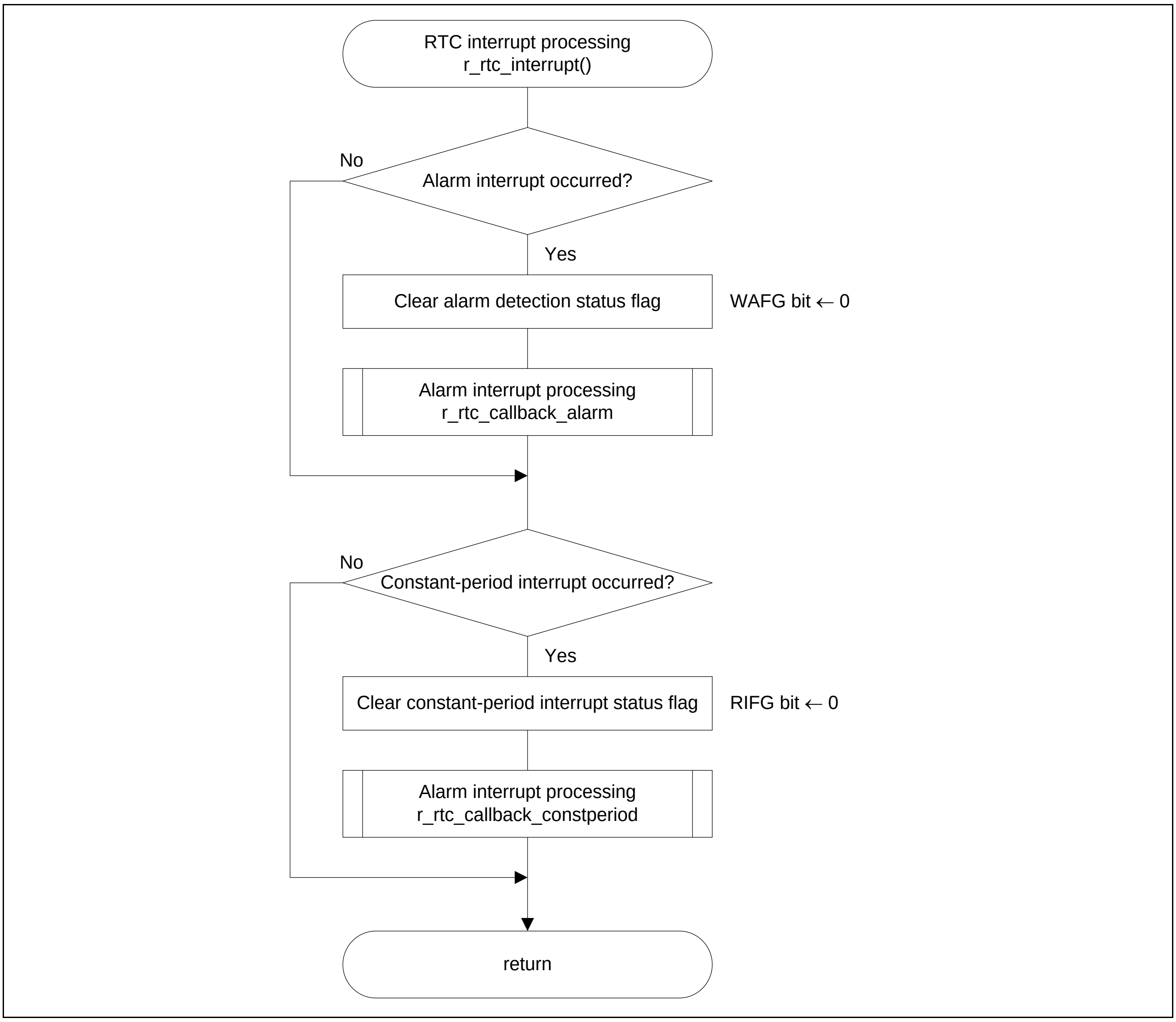

Figure 5.41 RTC Interrupt Processing

5.7.35 Alarm Interrupt Processing

Figure 5.42 shows the flowchart for alarm interrupt processing.

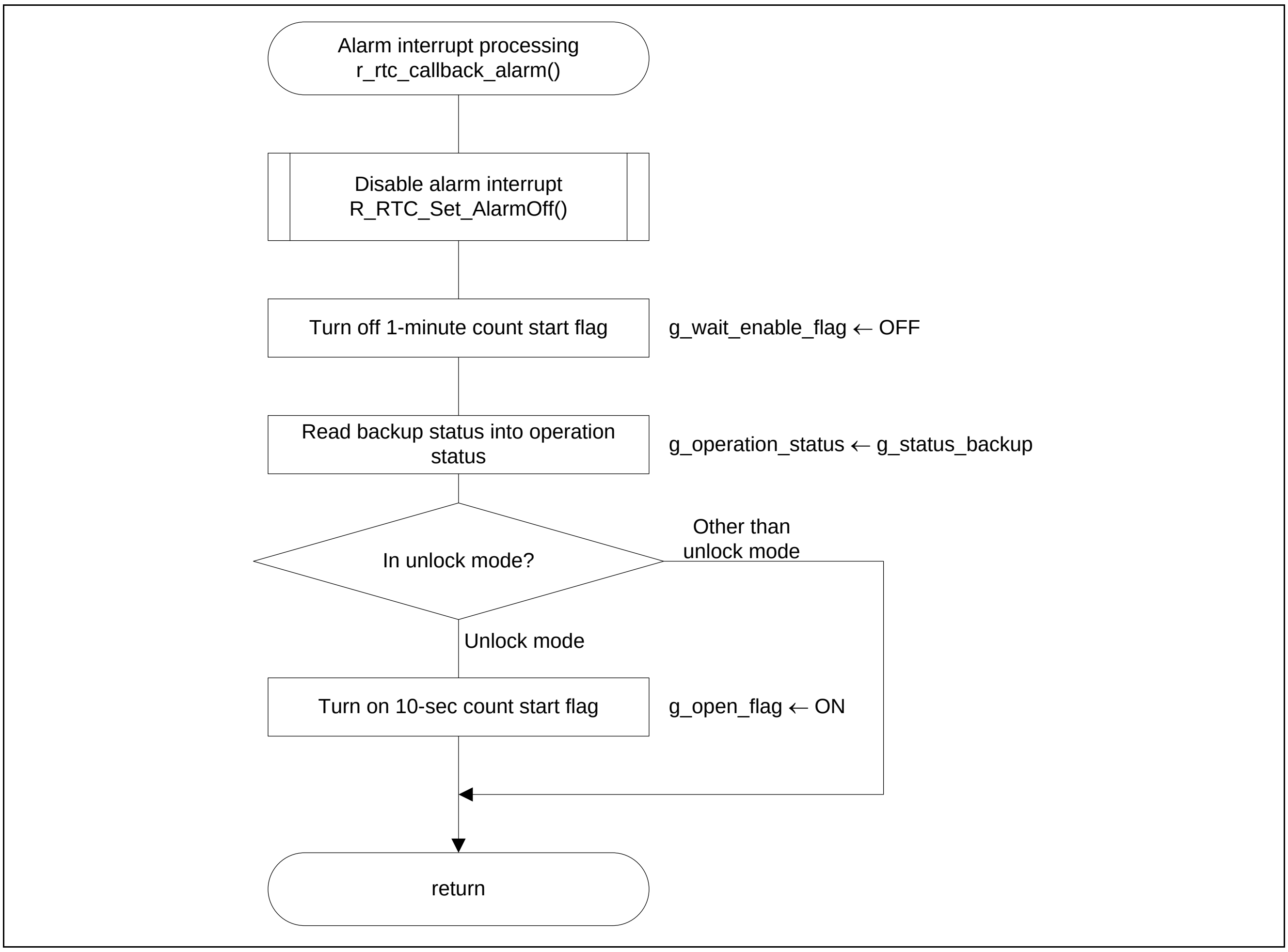

Figure 5.42 Alarm Interrupt Processing

5.7.36 Constant-Period Interrupt Processing

Figure 5.43 shows the flowchart for constant-period interrupt processing.

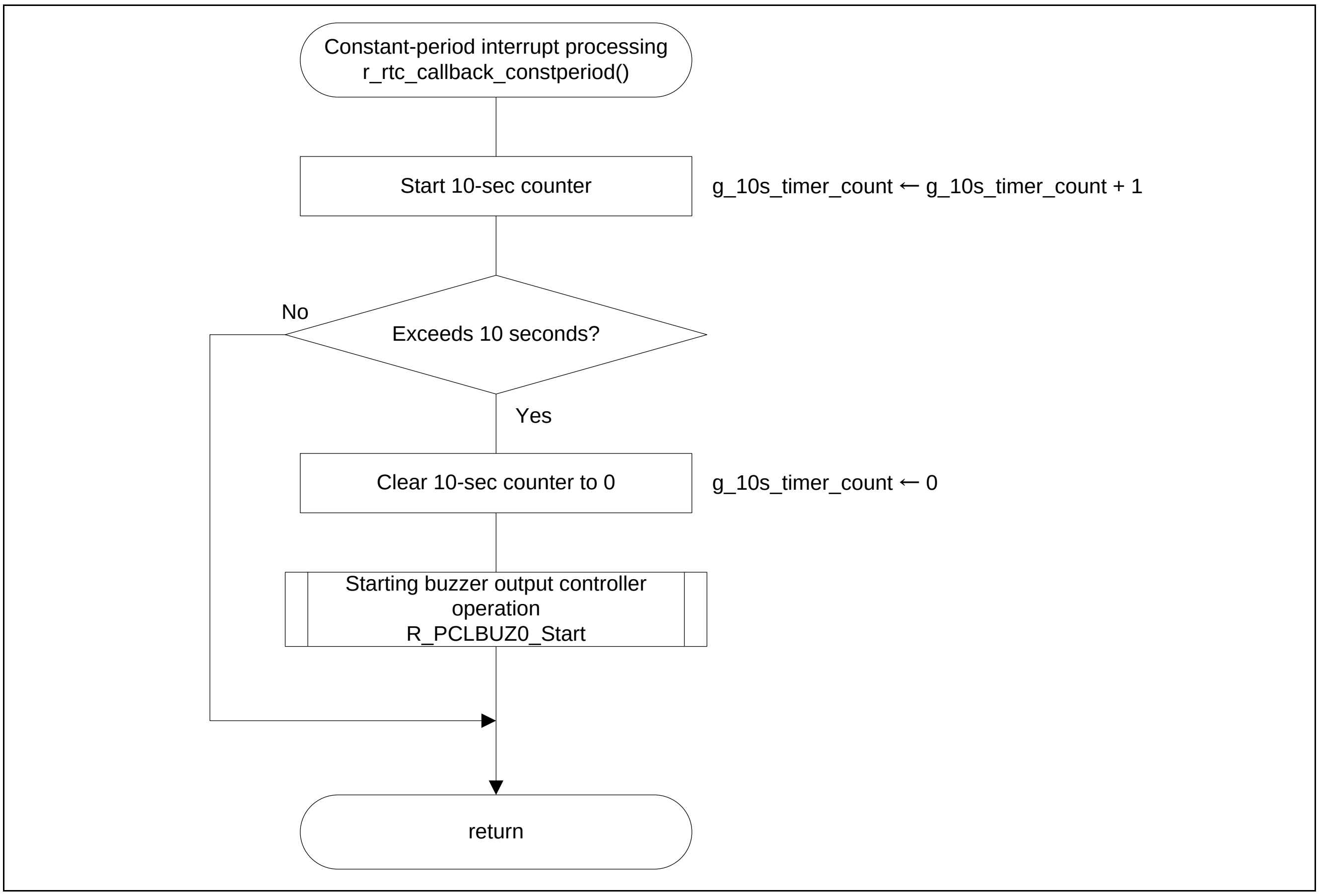

Figure 5.43 Constant-Period Interrupt Processing

6. Sample Code

The sample code is available on the Renesas Electronics Website.

7. Documents for Reference

RL78/G1F User's Manual: Hardware (R01UH0516E)

RL78 Family User's Manual: Software (R01US0015E)

(The latest versions of the documents are available on the Renesas Electronics Website.)

Website and Support

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

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 28, 2018	-	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

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

1. Handling of Unused Pins

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

   ¾ The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

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

   ¾ The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   ¾ The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

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

   ¾ When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

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

   ¾ The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

RENESAS

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

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

Renesas Electronics Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.2