RX63N Group, RX631 Group

R01AN1920EJ0100
Rev. 1.00
Sep. 1, 2014

Transition to Low Power Consumption Mode Using Low Power Consumption Function

Abstract

This document describes setting examples of transition to each low power consumption mode for the RX63N Group, RX631 Group.

Products

RX63N Group, 177-pin and 176-pin packages	ROM capacities: 768 Kbytes to 2 Mbytes
RX63N Group, 145-pin and 144-Pin packages	ROM capacities: 768 Kbytes to 2 Mbytes
RX63N Group, 100-pin package	ROM capacities: 768 Kbytes to 2 Mbytes
RX631 Group, 177-pin and 176-pin packages	ROM capacities: 256 Kbytes to 2 Mbytes
RX631 Group, 145-pin and 144-pin packages	ROM capacities: 256 Kbytes to 2 Mbytes
RX631 Group, 100-pin package	ROM capacities: 256 Kbytes to 2 Mbytes

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

Selectable triggers can be used to enter and exit low power consumption modes.

The MCU enters a low power consumption mode when a trigger to place the MCU in a low power consumption mode (transition trigger) occurs. The MCU exits a low power consumption mode when a trigger to release the MCU from the low power consumption mode (release trigger) occurs.

Low power consumption modes: Selectable from sleep mode, software standby mode, or deep software standby mode
Transition and release triggers: Selectable from IRQ9 or the RTC
Clock source: Selectable *1
Operating power control mode: Selectable *1
Sleep mode return clock source switching function: Used *2

Note 1. Selected by r_init_clock.h. In this document, the PLL clock is selected as a clock source.
Refer to the “RX63N Group, RX631 Group Initial Setting Rev. 1.10” application note for details.

Note 2. After returning to the main clock, the clock source is switched to the PLL clock.

Table 1.1 lists the Peripheral Functions and Their Applications, and Figure 1.1 shows the Block Diagram.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
Low power consumption function	Reduces power consumption.
External pin interrupt (IRQ)	Enters a low power consumption mode. Exits a low power consumption mode.
Realtime clock (RTC)	Enters a low power consumption mode. Exits a low power consumption mode.

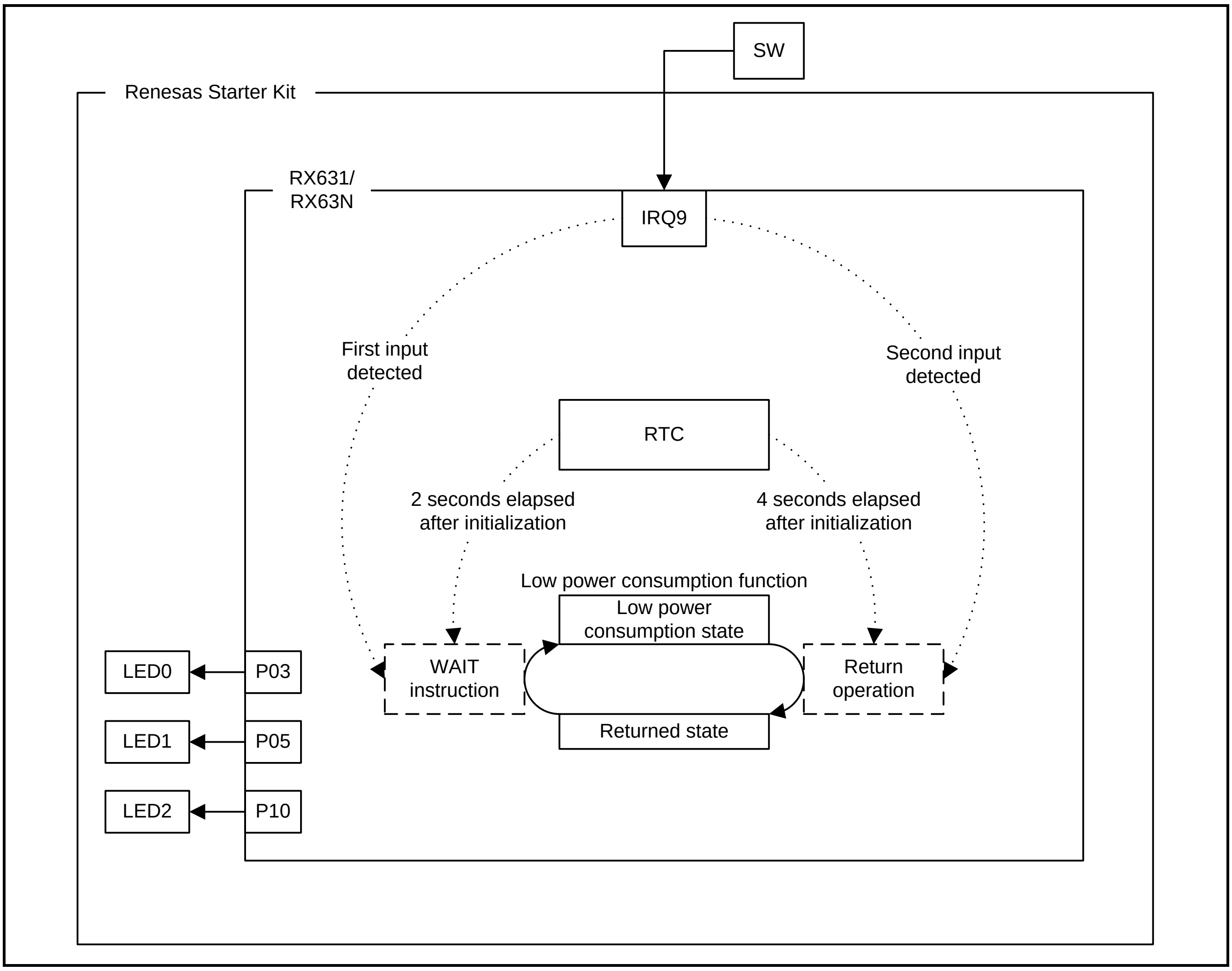


Figure 1.1 Block Diagram

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	R5F563NBDDFC (RX63N Group)
Operating frequencies	Main clock: 12 MHz PLL: 192 MHz (main clock divided by 1 and multiplied by 16) HOCO stopped System clock (ICLK): 96 MHz (PLL divided by 2) Peripheral module clock B (PCLKB): 48 MHz (PLL divided by 4)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance Embedded Workshop Version 4.09.01
C compiler	Renesas Electronics C/C++ Compiler Package for RX Family V.1.02 Release 01
	Compile options -cpu=rx600 -output=obj="$(CONFIGDIR)¥$(FILELEAF).obj" -debug -nologo (Default settings of the integrated development environment are used.)
iodefine.h version	Version 1.6A
Endian	Little endian
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Sample code version	Version 1.00
Board used	Renesas Starter Kit+ for RX63N (part number: R0K50563NC000BE)

3. Reference Application Note

For additional information associated with this document, refer to the following application note:

RX63N Group, RX631 Group Initial Setting Rev 1.10 (R01AN1245EJ)

The initial setting functions in the reference application note are used in the sample code in this application note. The revision number of the reference application note is current as of the publication of this application note. However, the latest version is always recommended. Visit the Renesas Electronics Corporation website to check for and download the latest version.

4. Hardware

4.1 Hardware Configuration

When the IRQ interrupt is used to exit deep software standby mode, only pins IRQ0-DS to IRQ15-DS can be used. In this document, the IRQ9-DS pin where an external switch is connected is used.

Figure 4.1 shows the Example of Connecting an External Switch to the IRQ9-DS Pin.

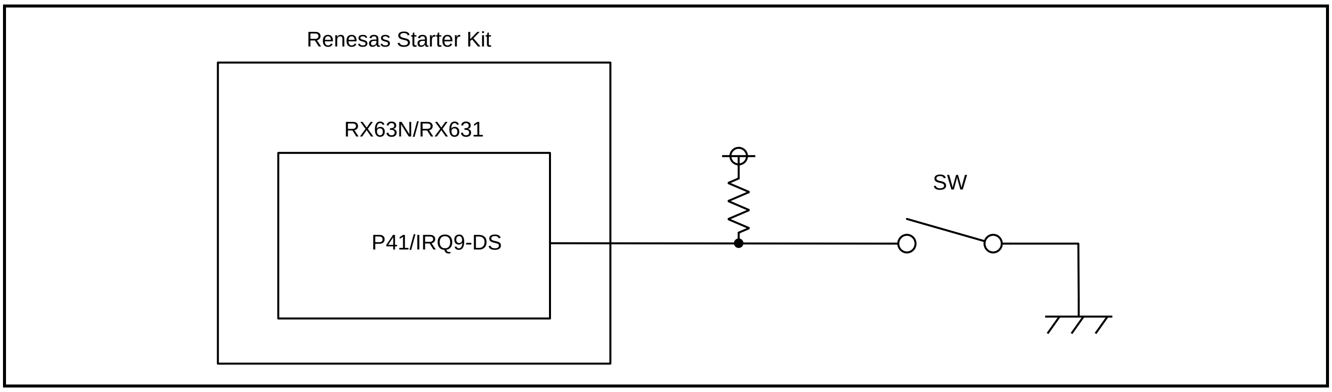


Figure 4.1 Example of Connecting an External Switch to the IRQ9-DS Pin

4.2 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

These pins are used in the 176-pin package. When using the product with less than 176 pins, select pins according to the product used.

Table 4.1 Pins Used and Their Functions

Pin Name	I/O	Function
P41/IRQ9-DS	Input	SW input (transition to and release from a low power mode)
P03	Output	LED0 output (turned on after initialization)
P05	Output	LED1 output (turned on before transition to a low power consumption mode)
P10	Output	LED2 output (turned on after release form a low power consumption mode)

5. Software

The MCU enters a low power consumption mode when a trigger to place the MCU in a low power consumption mode (transition trigger) occurs. The MCU exits a low power consumption mode when a trigger to release the MCU from low power consumption mode (release trigger) occurs in the low power consumption mode.

In the sample code, the low power consumption mode, and transition and release triggers can be selected.

The low power consumption mode is selectable from sleep mode, software standby mode, or deep software standby mode.

Transition and release triggers are selectable from the IRQ or RTC.

The settings of the peripheral functions used are as follows.

IRQ

Detection: Falling edge of the IRQ9 pin

Digital filter: Disabled

Interrupt priority: Level 15

RTC

Default time: Wed, Jan 1, 2014, 00:00:00

Hours mode: 24-hour mode

Interrupt: Periodic interrupt (PRD) is used (generated every 2 seconds).

Interrupt priority: Level 15

5.1 Operation Overview

5.1.1 Sleep Mode

After initialization, LED0 is turned on, and the MCU waits for a transition trigger. When a transition trigger occurs, LED0 is turned off, LED1 is turned on, and the MCU enters sleep mode. When a release trigger occurs in sleep mode, the MCU exits sleep mode, LED1 is turned off, and LED2 is turned on.

Figure 5.1 shows the Operation Overview in Sleep Mode.

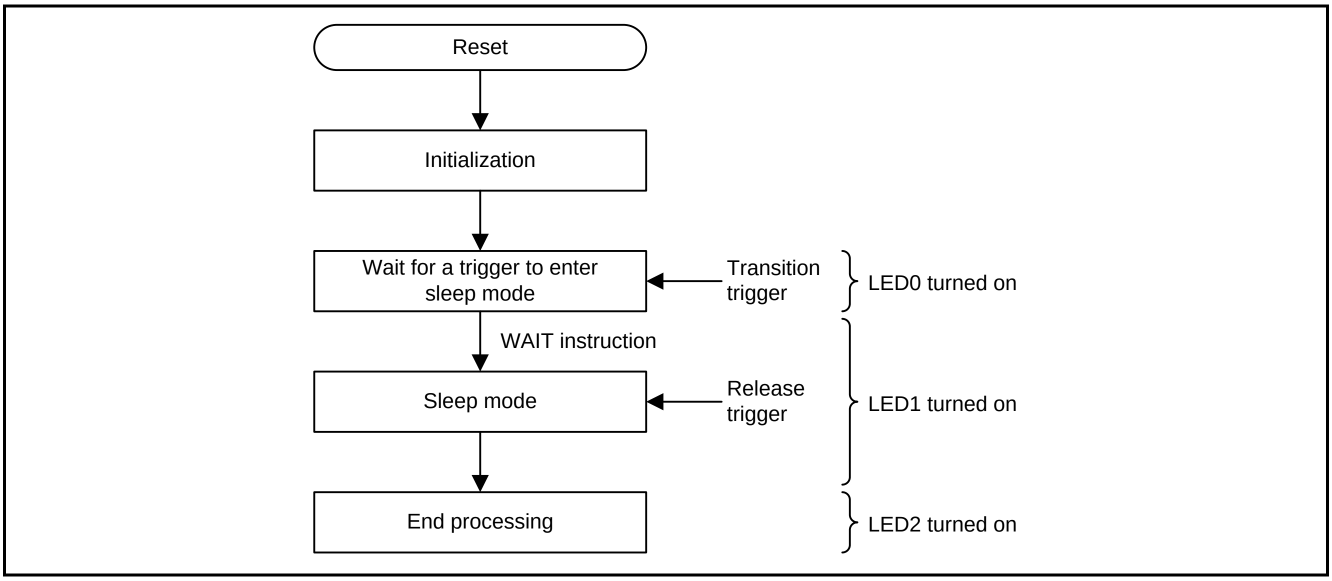


Figure 5.1 Operation Overview in Sleep Mode

5.1.2 Software Standby Mode

After initialization, LED0 is turned on, and the MCU waits for a transition trigger. When a transition trigger occurs, LED0 is turned off, LED1 is turned on, and the MCU enters software standby mode. When a release trigger occurs in software standby mode, the MCU exits software standby mode, LED1 is turned off, and LED2 is turned on.

Figure 5.2 shows the Operation Overview in Software Standby Mode.

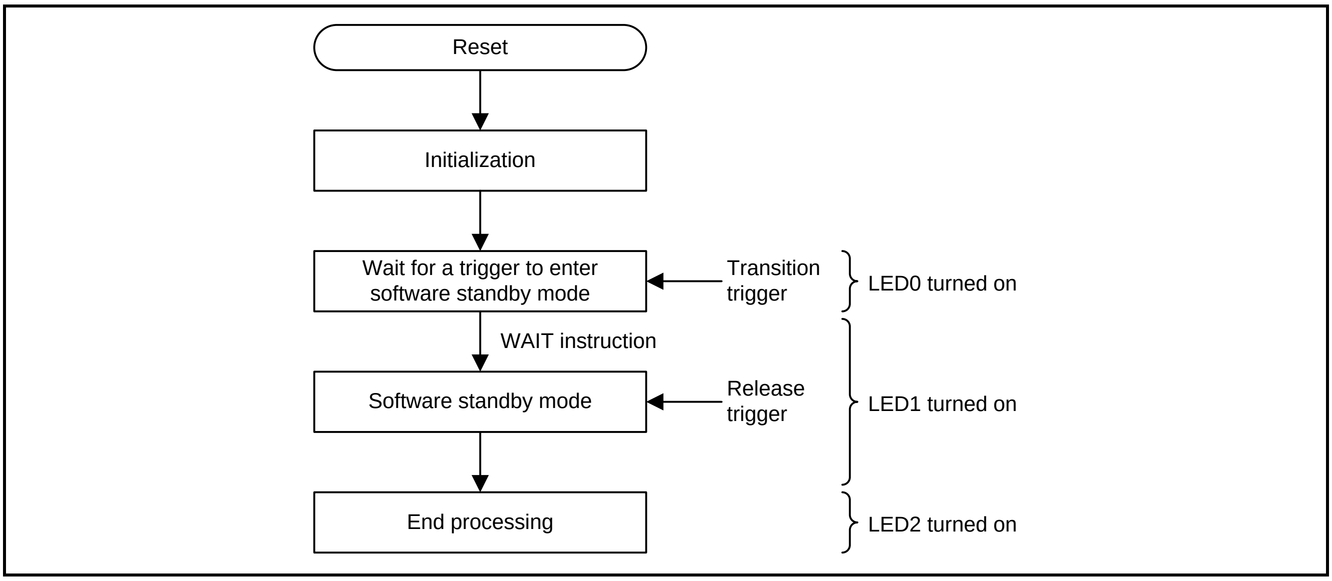
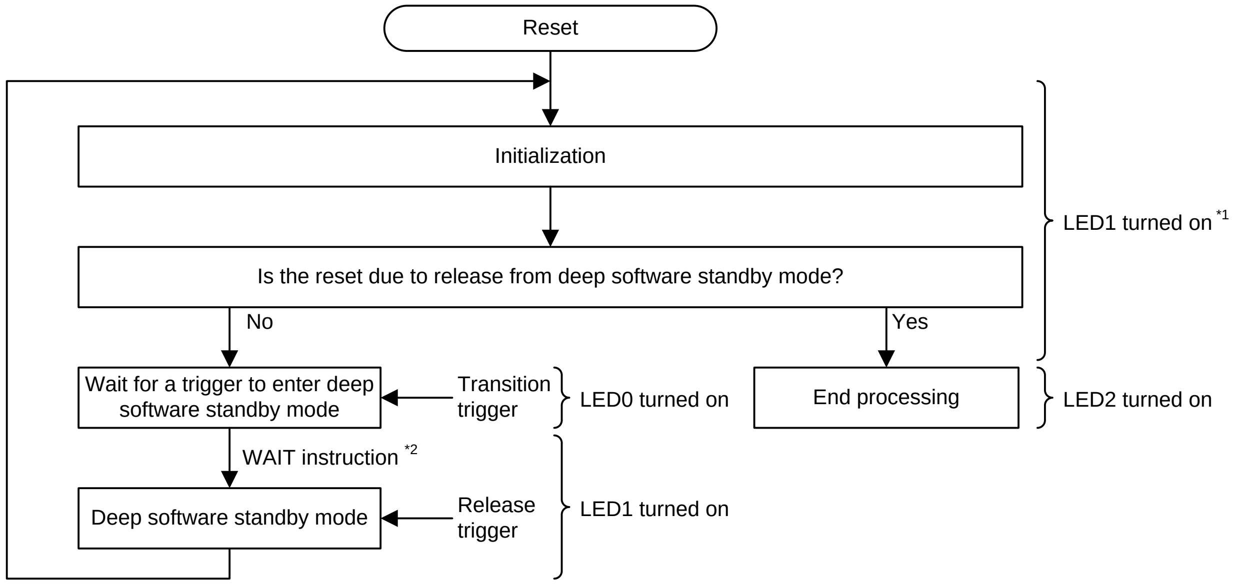


Figure 5.2 Operation Overview in Software Standby Mode

5.1.3 Deep Software Standby Mode

After initialization, LED0 is turned on, and the MCU waits for a transition trigger. When a transition trigger occurs, LED0 is turned off, LED1 is turned on, and the MCU enters deep software standby mode. When a release trigger occurs in deep software standby mode, the MCU exits deep software standby mode, LED1 is turned off, and LED2 is turned on.

Figure 5.3 shows the Operation Overview in Deep Software Standby Mode.



Note 1. LED1 is turned on only for reset due to release from deep software standby mode (deep software standby reset).
Note 2. The MCU enters deep software standby mode via software standby mode. If a request to exit software standby mode has been generated when the WAIT instruction is executed, the MCU does not enter deep software standby mode, and the subsequent instruction is executed. In this case, the deep software standby reset does not occur.

Figure 5.3 Operation Overview in Deep Software Standby Mode

5.1.4 Entering and Exiting a Low Power Consumption Mode by the IRQ

When the IRQ is selected as transition and release triggers, the MCU enters and exits a low power consumption mode by generating IRQ interrupt requests.

When an IRQ interrupt request is generated while the MCU waits for a transition trigger, the IRQ interrupt request becomes a transition trigger. When an IRQ interrupt request is generated in a low power consumption mode, the IRQ interrupt request becomes a release trigger.

Figure 5.4 shows the Timing to Enter and Exit a Low Power Consumption Mode by the IRQ.

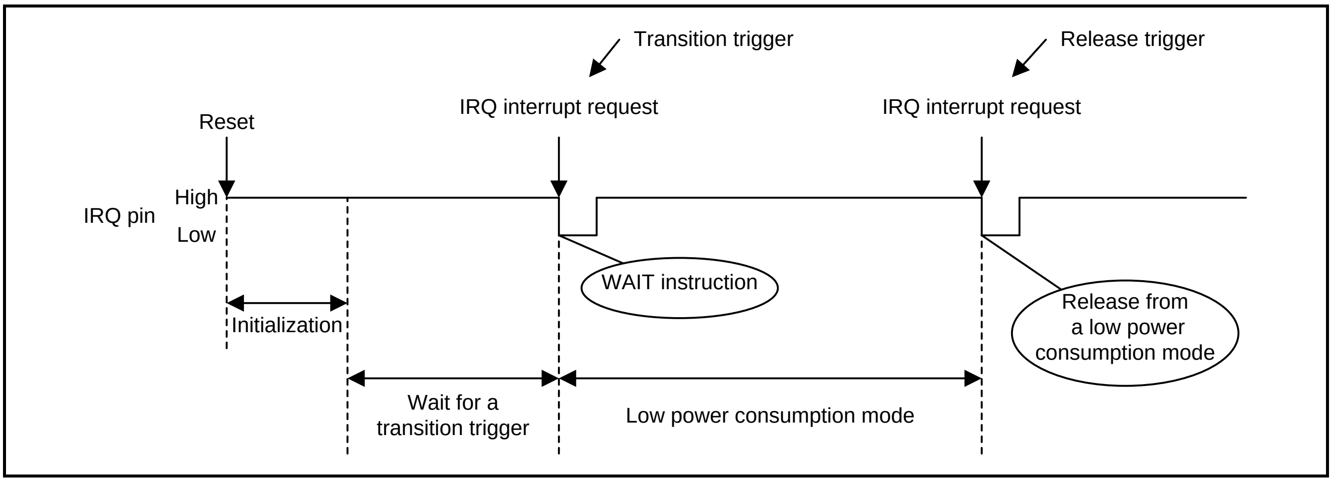


Figure 5.4 Timing to Enter and Exit a Low Power Consumption Mode by the IRQ

5.1.5 Entering and Exiting a Low Power Consumption Mode by the RTC

When the RTC is selected as the transition and release triggers, the MCU enters and exits a low power consumption mode by an RTC.PRD interrupt request that is generated every 2 seconds.

When an RTC.PRD interrupt request is generated while the MCU waits for a transition trigger, the RTC.PRD interrupt request becomes a transition trigger. When an RTC.PRD interrupt request is generated in a low power consumption mode, the RTC.PRD interrupt request becomes a release trigger.

Figure 5.5 shows the Timing to Enter and Exit a Low Power Consumption Mode by the RTC.

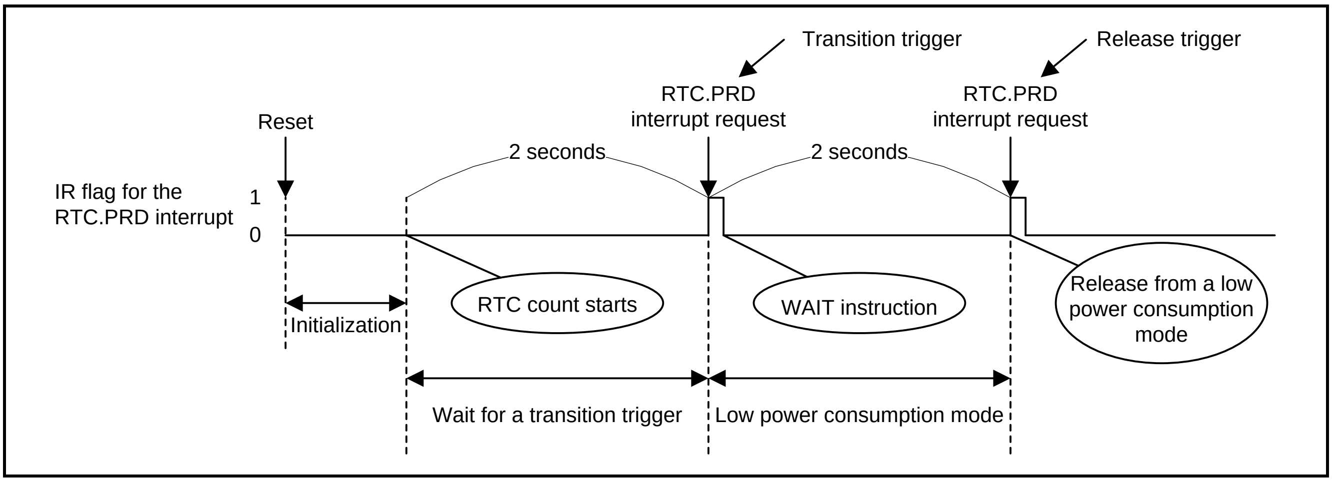


Figure 5.5 Timing to Enter and Exit a Low Power Consumption Mode by the RTC

5.2 File Composition

Table 5.1 lists the Files Used in the Sample Code. Files generated by the integrated development environment are not included in this table.

Table 5.1 Files Used in the Sample Code

File Name	Outline	Remarks
main.c	Main processing	
r_init_stop_module.c	Stop of peripheral functions that are operating after a reset	
r_init_stop_module.h	Header file for r_init_stop_module.c	
r_init_non_existent_port.c	Initial setting for non-existent ports	
r_init_non_existent_port.h	Header file for r_init_non_existent_port.c	
r_init_clock.c	Clock initialization	
r_init_clock.h	Header file for r_init_clock.c	Sub-clock setting pattern D is selected.

5.3 Option-Setting Memory

Table 5.2 lists the Option-Setting Memory Configured in the Sample Code. When necessary, set a value suited to the user system.

Table 5.2 Option-Setting Memory Configured in the Sample Code

Symbol	Address	Setting Value	Contents
OFS0	FFFF FF8Fh to FFFF FF8Ch	FFFF FFFFh	After a reset, the IWDT stops. After a reset, the WDT stops.
OFS1	FFFF FF8Bh to FFFF FF88h	FFFF FFFFh	After a reset, voltage monitoring 0 reset is disabled. After a reset, the HOCO oscillation is disabled.
MDES	FFFF FF83h to FFFF FF80h	FFFF FFFFh	Little endian

5.4 Constants

Table 5.3 lists the Constants Used in the Sample Code.

Table 5.3 Constants Used in the Sample Code

Constant Name	Setting Value	Contents
L_IRQ	0	Transition and release triggers: IRQ
L_RTC	1	Transition and release triggers: RTC
L_SOURCE	L_IRQ	Selected transition and release triggers: IRQ
L_SLEEP	0	Low power consumption mode: Sleep mode
L_SOFT_STANDBY	1	Low power consumption mode: Software standby mode
L_DEEP_STANDBY	2	Low power consumption mode: Deep software standby mode
L_MODE	L_SLEEP	Selected low power consumption mode: Sleep mode
L_LP_DISABLE	0	Transition trigger: None
L_LP_ENABLE	1	Transition trigger: Generated

5.5 Variables

Table 5.4 lists the static Variables.

Table 5.4 static Variables

Type	Variable Name	Contents	Function Used
static uint8_t	initial_end	Initialization end flag 0: Initialization in progress 1: Initialization ended	sleep_mode software_standby_mode deep_standby_mode Excep_ICU_IRQ9 Excep_RTC_PRD
static uint8_t	enable_low_power	Low power consumption mode transition trigger flag	

5.6 Functions

Table 5.5 lists the Functions Used in the Sample Code.

Table 5.5 Functions Used in the Sample Code

Function Name	Outline
main	Main processing
port_init	Port initialization
R_INIT_StopModule	Stop of peripheral functions that are operating after a reset
R_INIT_NonExistentPort	Initial setting for non-existent ports
R_INIT_Clock	Clock initialization
CGC_oscillation_main	Main clock oscillation setting
CGC_oscillation_PLL	PLL clock oscillation setting
peripheral_init	Peripheral function initialization
sleep_mode	Transition to sleep mode
software_standby_mode	Transition to software standby mode
deep_standby_mode	Transition to deep software standby mode
irq_init	IRQ initialization
rtc_init	RTC initialization
Excep_ICU_IRQ9	IRQ9 interrupt handling
Excep_RTC_PRD	RTC.PRD interrupt handling

5.7 Function Specifications

The following tables list the sample code function specifications.

main	
Outline	Main processing
Header	None
Declaration	void main(void)
Description	After initialization, the MCU enters a low power consumption mode.
Arguments	None
Return Value	None

port_init	
Outline	Port initialization
Header	None
Declaration	void port_init(void)
Description	Initializes ports.
Arguments	None
Return Value	None

R_INIT_StopModule	
Outline	Stop of peripheral functions that are operating after a reset
Header	r_init_stop_module.h
Declaration	void R_INIT_StopModule(void)
Description	Performs the setting to enter the module stop state.
Arguments	None
Return Value	None
Remarks	In the sample code, transition to the module stop state is not performed. Refer to the "RX63N Group, RX631 Group Initial Setting Rev. 1.10" application note for details on this function.

R_INIT_NonExistentPort	
Outline	Initial setting for non-existent ports
Header	r_init_non_existent_port.h
Declaration	void R_INIT_NonExistentPort(void)
Description	Initializes the port direction register corresponding to non-existent ports for products with less than 100 pins.
Arguments	None
Return Value	None
Remarks	In the sample code, 176-pin package is set (PIN_SIZE = 176). After calling this function, when writing in bytes to registers PDR and PODR that include non-existent ports, set the I/O select bits port for non-existent ports to 1 and set the port output data store bits for non-existent ports to 0. Refer to the "RX63N Group, RX631 Group Initial Setting Rev. 1.10" application note for details on this function.

R_INIT_Clock

Outline	Clock initialization
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Initializes clocks.
Arguments	None
Return Value	None
Remarks	In the sample code, the PLL is used for the system clock, and sub-clock oscillation is used for RTC operation. Refer to the “RX63N Group, RX631 Group Initial Setting Rev. 1.10” application note for details on this function. Sub-clock setting pattern D is selected.

CGC_oscillation_main

Outline	Main clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_main(void)
Description	Oscillates the main clock after setting the MOSCWTCR register, and then waits the main clock oscillation stabilization wait time by software.
Arguments	None
Return Value	None
Remarks	Refer to the “RX63N Group, RX631 Group Initial Setting Rev. 1.10” application note for details on this function.

CGC_oscillation_PLL

Outline	PLL oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_PLL(void)
Description	Oscillates the PLL clock after setting the PLL input division ratio and frequency multiplier and setting the PLLWTCR register, and then waits the PLL clock oscillation stabilization wait time.
Arguments	None
Return Value	None
Remarks	Refer to the “RX63N Group, RX631 Group Initial Setting Rev. 1.10” application note for details on this function.

peripheral_init

Outline	Peripheral function initialization
Header	None
Declaration	void peripheral_init(void)
Description	Initializes the peripheral functions used.
Arguments	None
Return Value	None

sleep_mode	
Outline	Transition to sleep mode
Header	None
Declaration	void sleep_mode(void)
Description	Performs the setting to enter sleep mode.
Arguments	None
Return Value	None

software_standby_mode	
Outline	Transition to software standby mode
Header	None
Declaration	void software_standby_mode(void)
Description	Performs the setting to enter software standby mode.
Arguments	None
Return Value	None

deep_standby_mode	
Outline	Transition to deep software standby mode
Header	None
Declaration	void deep_standby_mode(void)
Description	Performs the setting to enter software standby mode.
Arguments	None
Return Value	None

irq_init	
Outline	IRQ initialization
Header	None
Declaration	void irq_init(void)
Description	Initializes the IRQ.
Arguments	None
Return Value	None

rtc_init	
Outline	RTC initialization
Header	None
Declaration	void rtc_init(void)
Description	Initializes the RTC.
Arguments	None
Return Value	None

Excep_ICU_IRQ9	
Outline	IRQ9 interrupt handling
Header	None
Declaration	static void Excep_ICU_IRQ9(void)
Description	Performs the IRQ9 interrupt handling.
Arguments	None
Return Value	None

Excep_RTC_PRD	
Outline	RTC.PRD interrupt handling
Header	None
Declaration	static void Excep_RTC_PRD(void)
Description	Performs the RTC.PRD interrupt handling.
Arguments	None
Return Value	None

5.8 Flowcharts

5.8.1 Main Processing

Figure 5.6 shows the Main Processing.

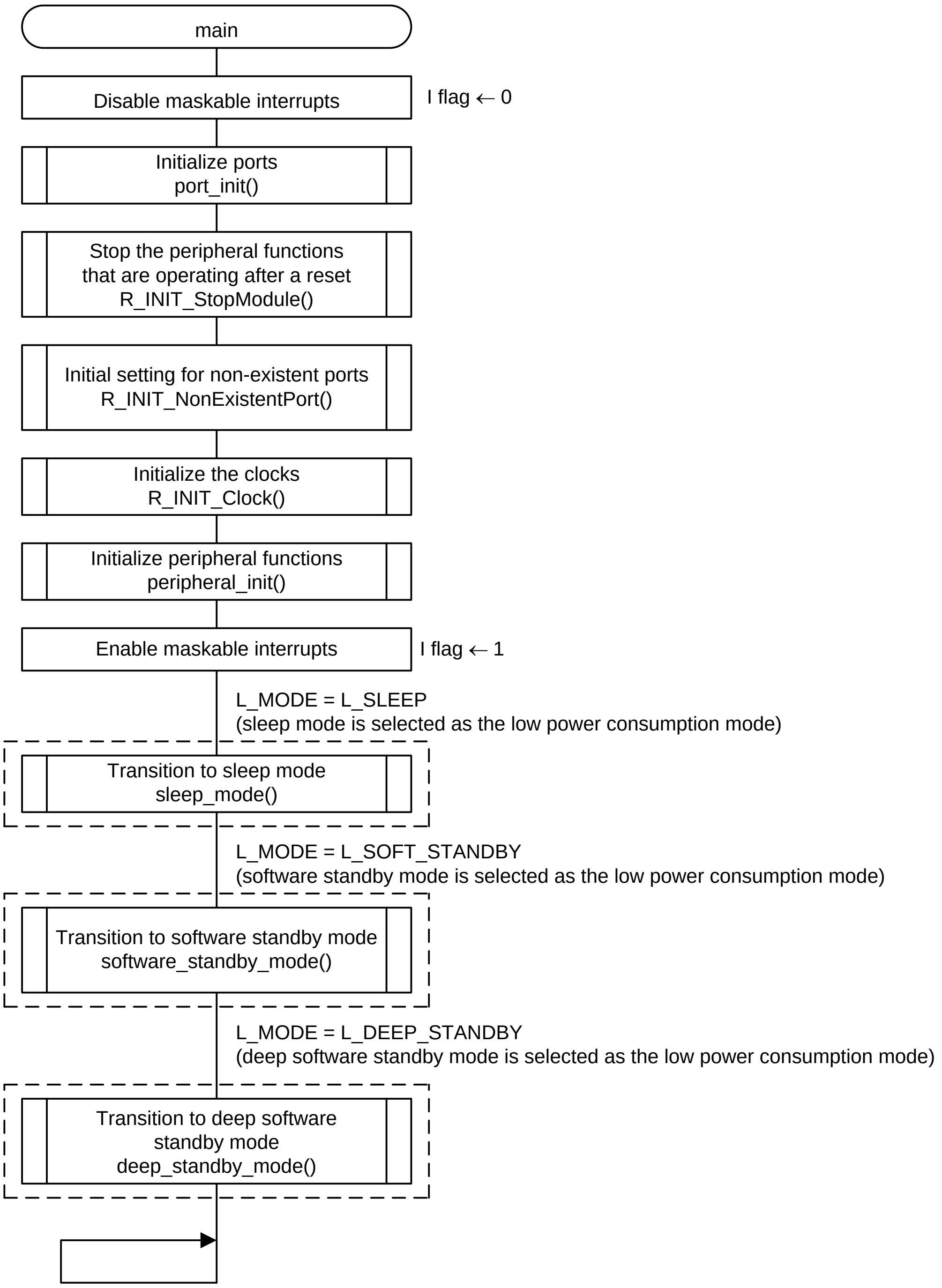


Figure 5.6 Main Processing

5.8.2 Port Initialization

Figure 5.7 shows the Port Initialization.

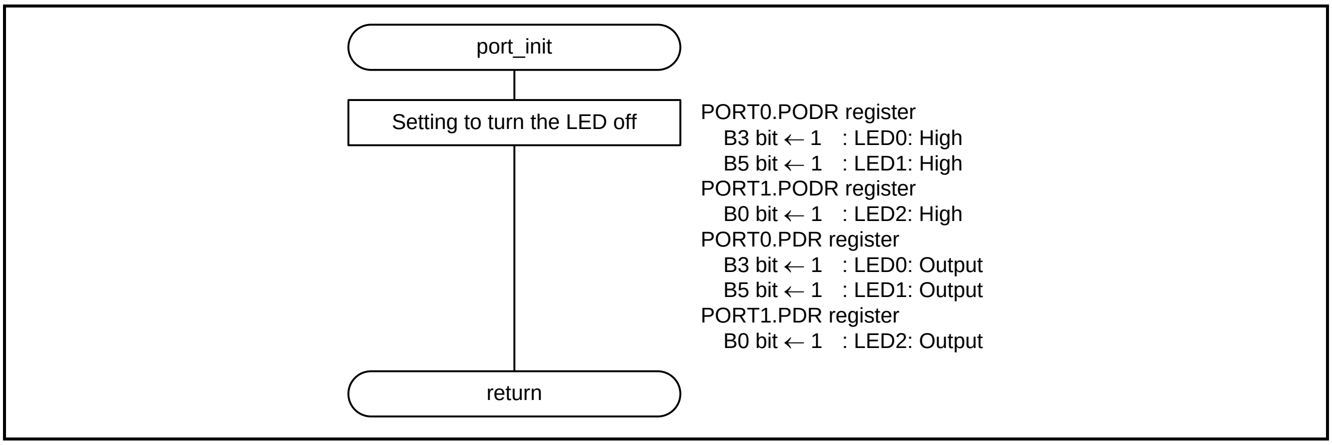


Figure 5.7 Port Initialization

5.8.3 Peripheral Function Initialization

Figure 5.8 shows the Peripheral Function Initialization.

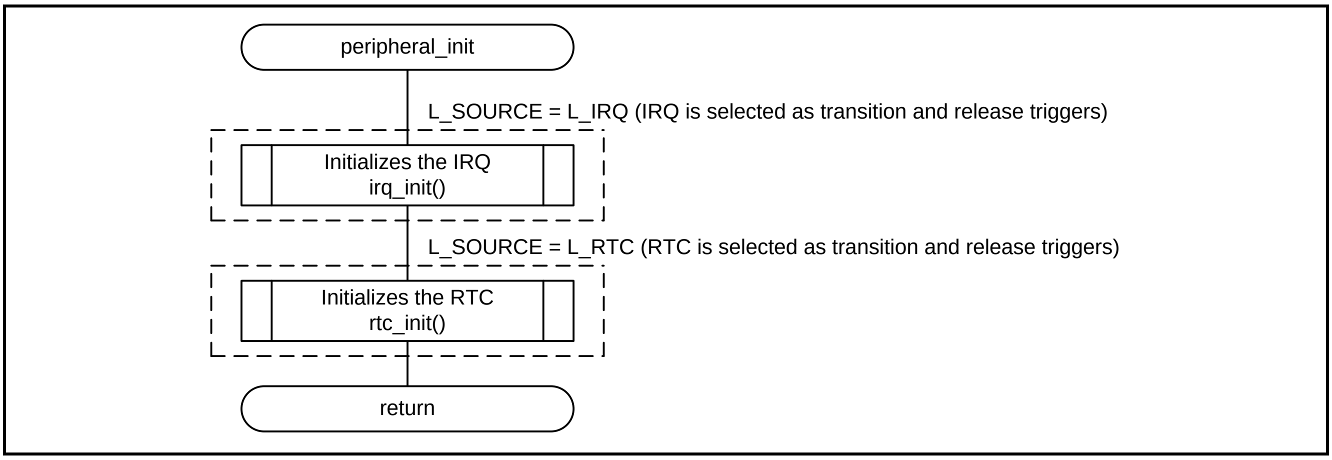
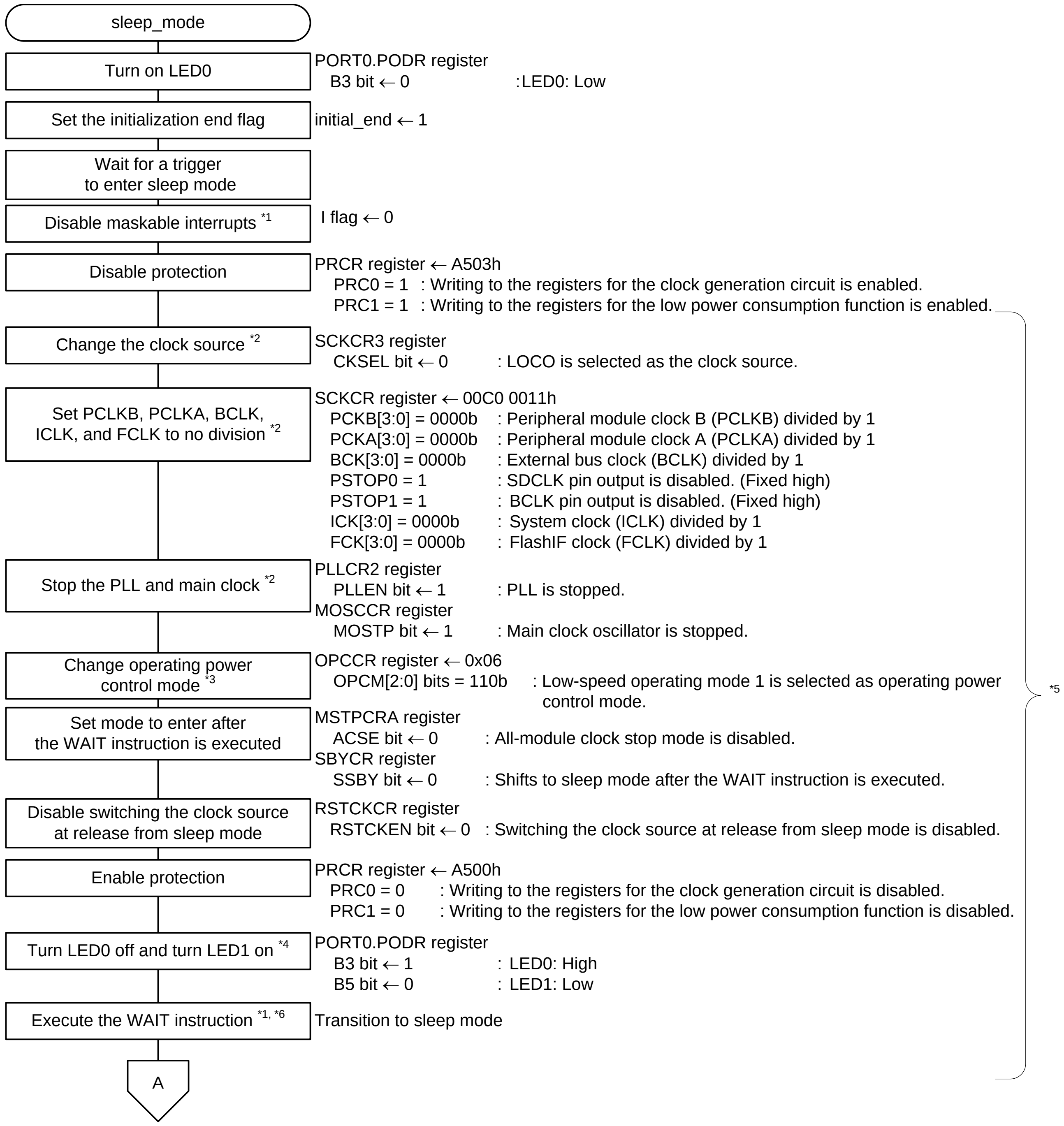


Figure 5.8 Peripheral Function Initialization

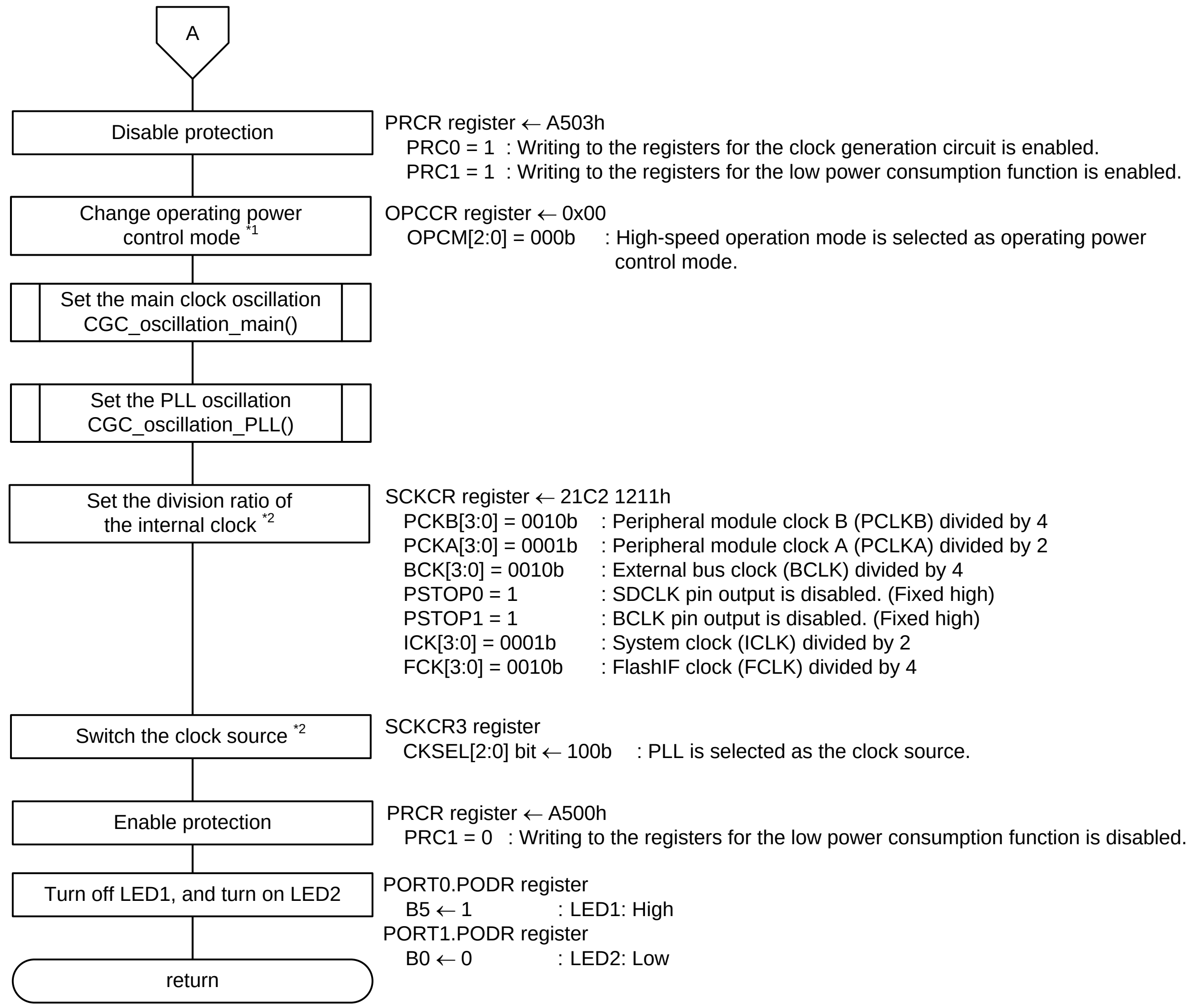
5.8.4 Transition to Sleep Mode

Figure 5.9 and Figure 5.10 show the Transition to Sleep Mode.



Note 1. If a return request is generated between when a trigger to enter sleep mode is detected and when maskable interrupts are disabled, the WAIT instruction is not executed.
Note 2. Read the written register to confirm that the written value can be read.
Note 3. Read the OPCCR.OPCMTSF bit to confirm that the transition has been completed.
Note 4. Read the register written before the WAIT instruction is executed and confirm that the written value can be read.
Note 5. Since the clock source is switched to the LOCO, the peripheral clock frequency changes.
Note 6. When a return request is generated after maskable interrupts are disabled, the MCU does not enter sleep mode even if the WAIT instruction is executed.

Figure 5.9 Transition to Sleep Mode (1/2)

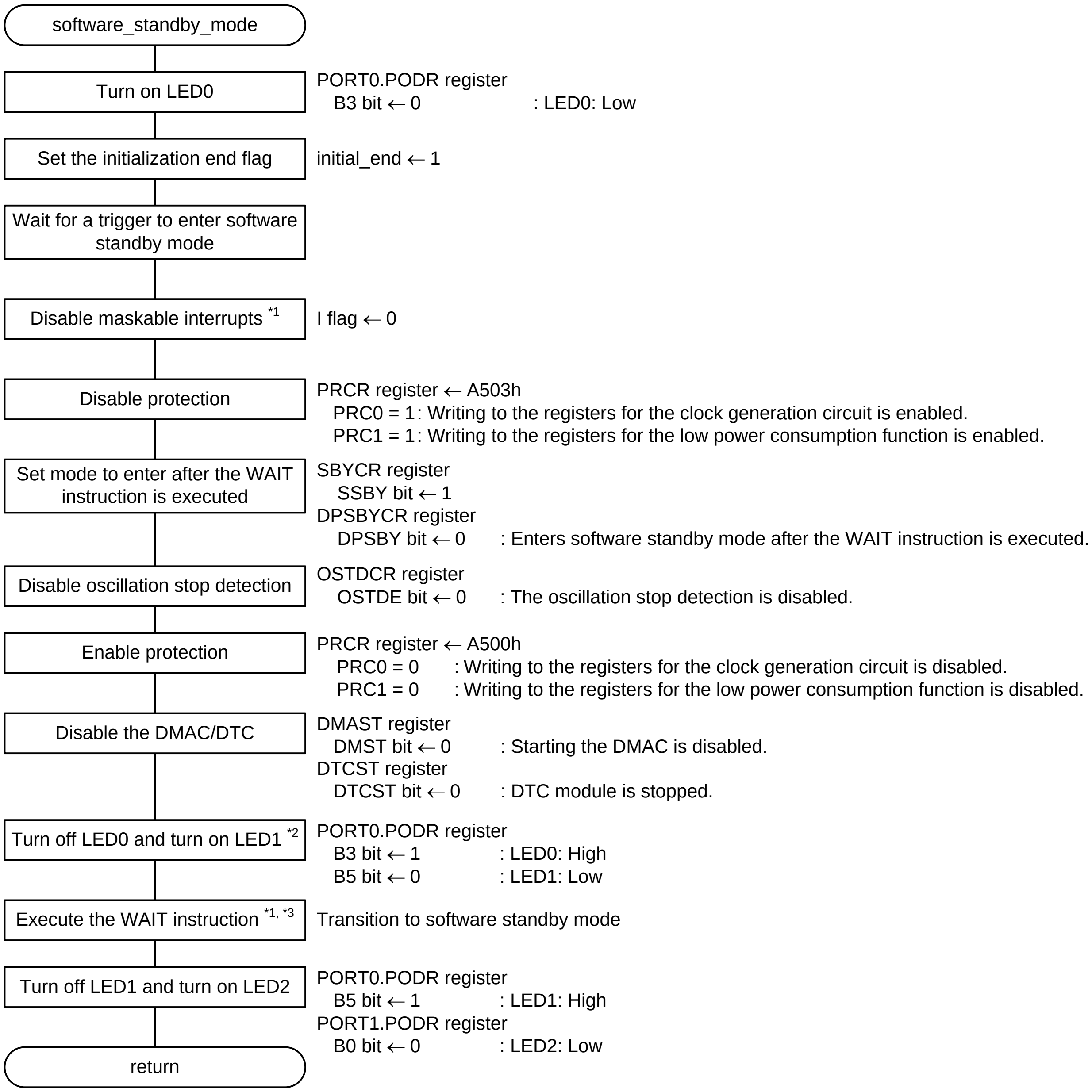


Note 1. Read the OPCCR.OPCMTSF bit to confirm that transition has been completed.
Note 2. Read the written register to confirm that the written value can be read.

Figure 5.10 Transition to Sleep Mode (2/2)

5.8.5 Transition to Software Standby Mode

Figure 5.11 shows the Transition to Software Standby Mode.



Note 1. If a return request is generated between when a trigger to enter software standby mode is detected and when maskable interrupts are disabled, the WAIT instruction is not executed.
Note 2. To confirm that the written value can be read, read the register written before the WAIT instruction is executed.
Note 3. When a return request is generated after maskable interrupts are disabled, the MCU does not enter software standby mode even if the WAIT instruction is executed.

Figure 5.11 Transition to Software Standby Mode

5.8.6 Transition to Deep Software Standby Mode

Figure 5.12 to Figure 5.14 show the Transition to Deep Software Standby Mode.

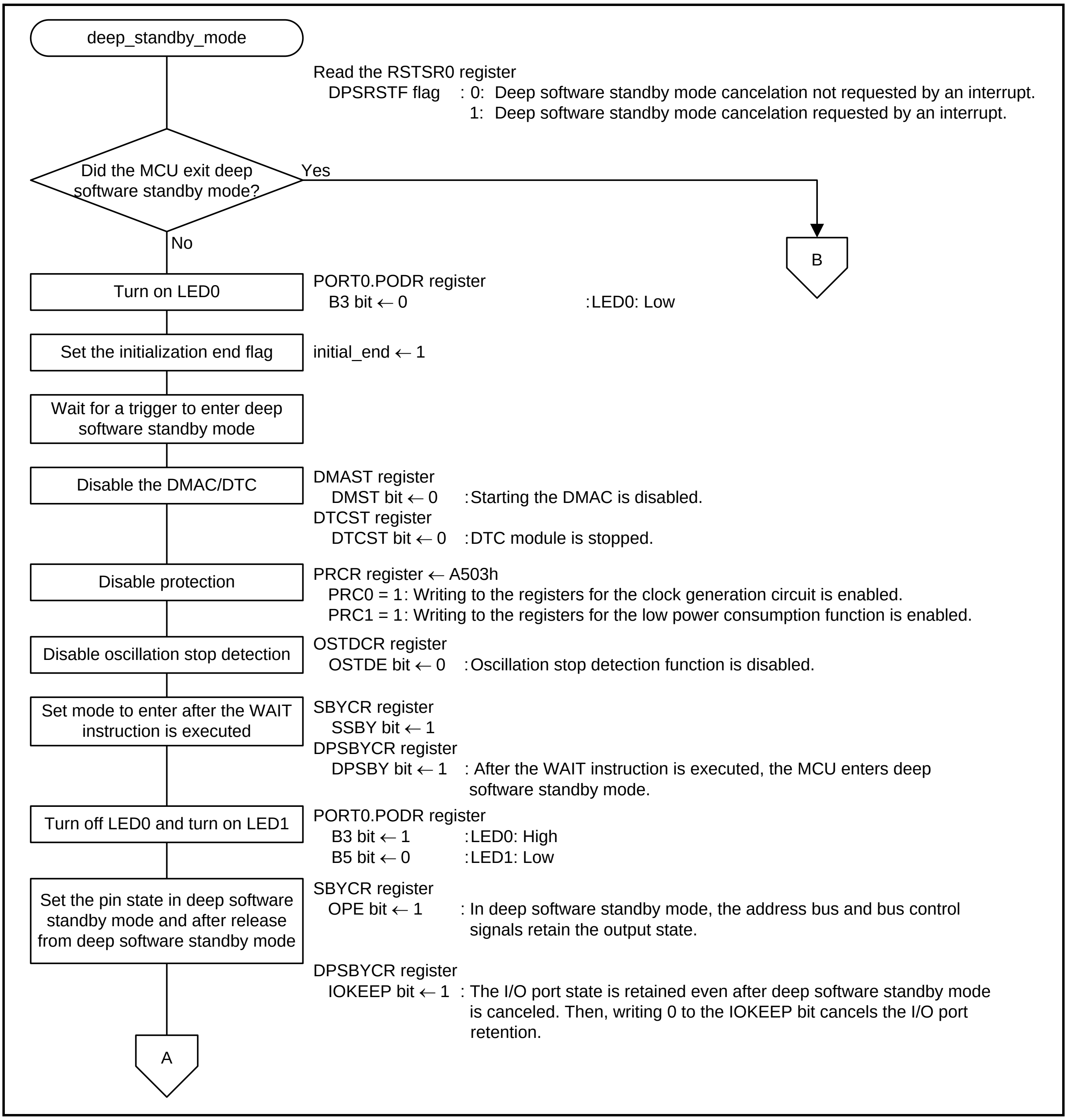
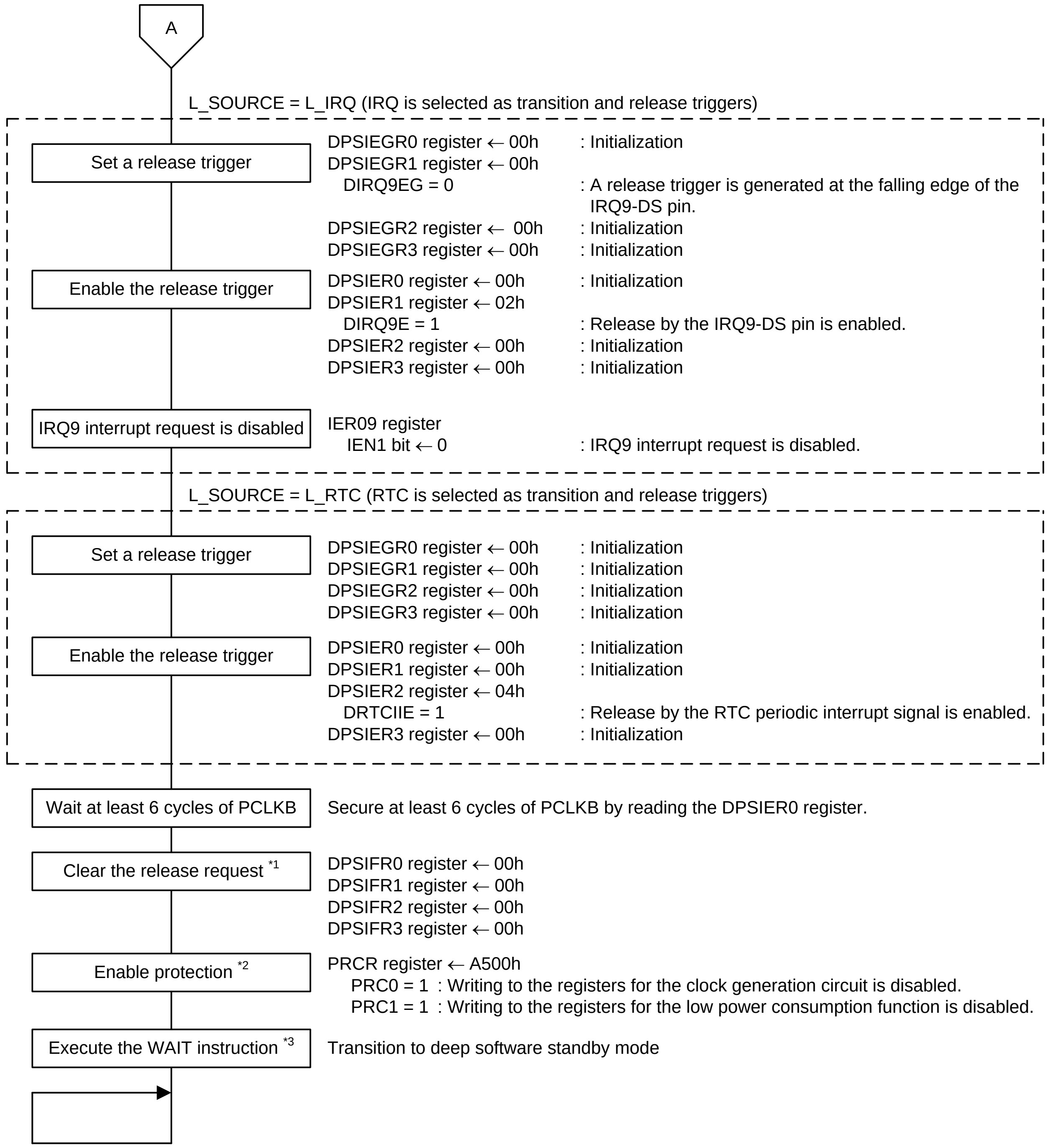
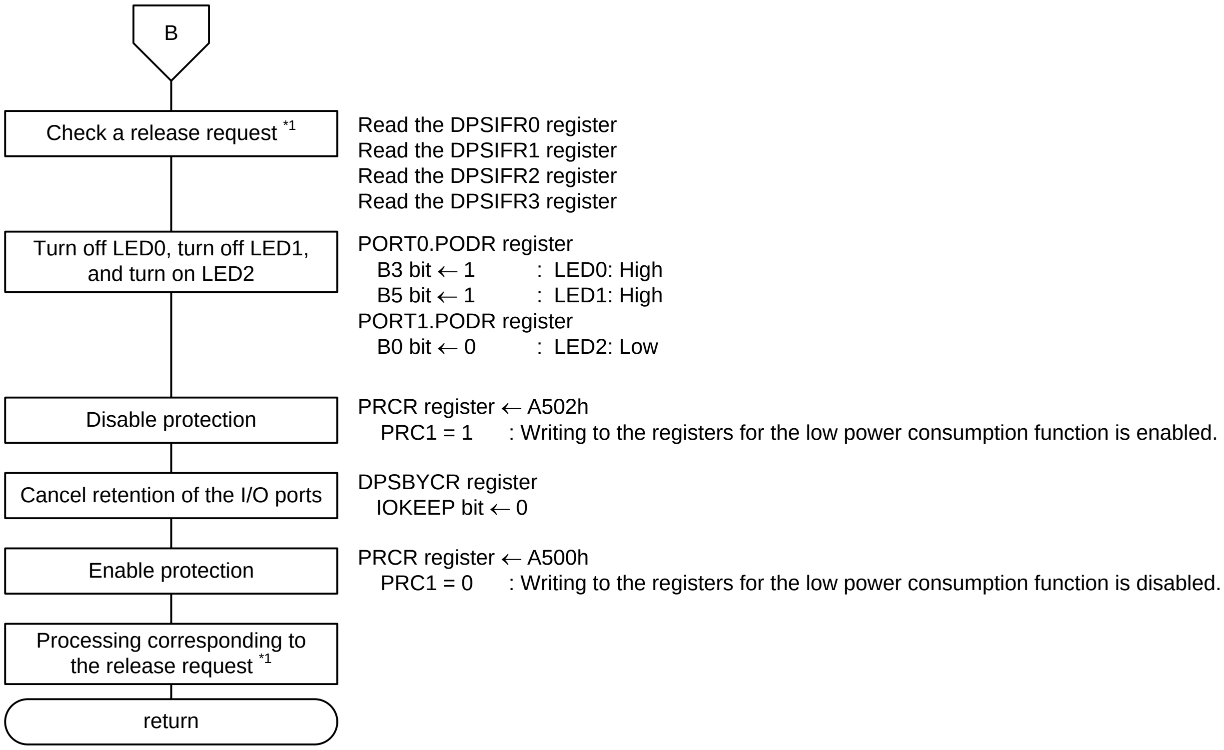


Figure 5.12 Transition to Deep Software Standby Mode (1/3)



Note 1. Read registers DRSIFR0, DPSIFR1, DPSIFR2, DPSIFR3 and then write 0.
Note 2. To confirm that the written value can be read, read the register written before the WAIT instruction is executed .
Note 3. The MCU enters deep software standby mode via software standby mode. If a request to exit software standby mode has been generated when the WAIT instruction is executed, the MCU does not enter deep software standby mode, and the subsequent instruction is executed. In this case, a deep software standby reset does not occur.

Figure 5.13 Transition to Deep Software Standby Mode (2/3)



Note 1: In the sample code, processing corresponding to a release request is not performed. Add a program as needed.

Figure 5.14 Transition to Deep Software Standby Mode (3/3)

5.8.7 IRQ Initialization

Figure 5.15 shows the IRQ Initialization.

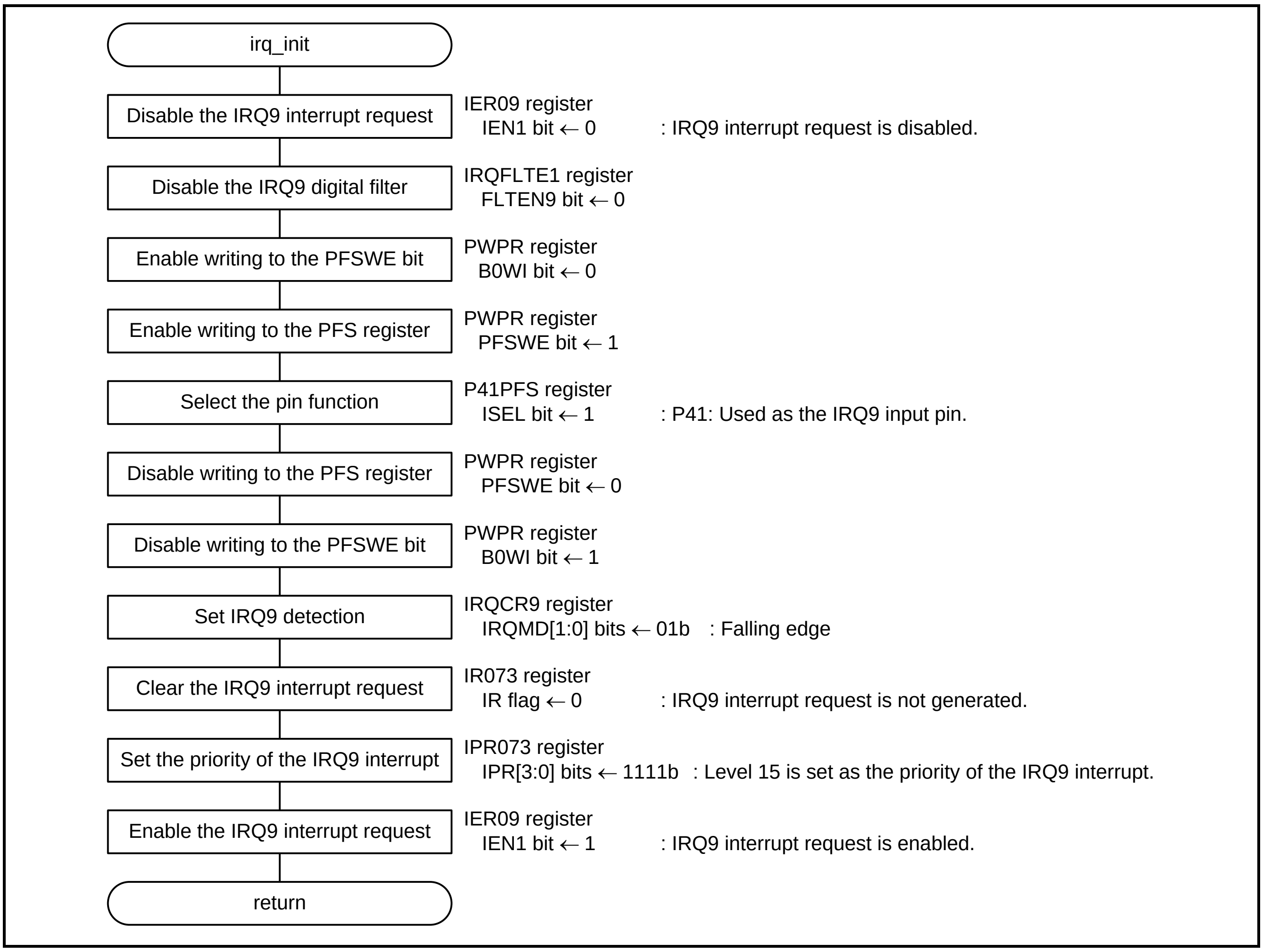
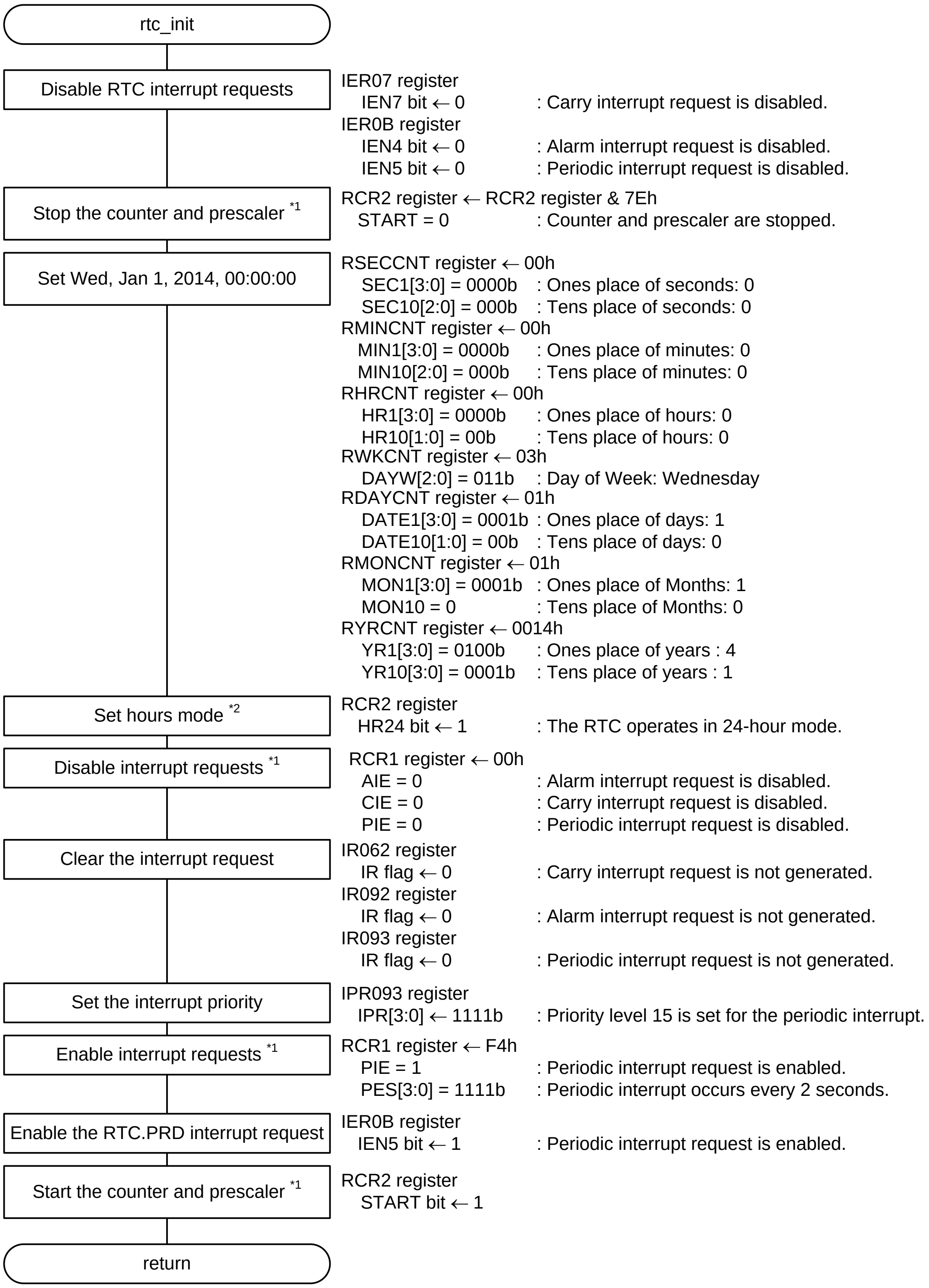


Figure 5.15 IRQ Initialization

5.8.8 RTC Initialization

Figure 5.16 shows the RTC Initialization.



Note 1. After writing to the RCR1 register and RCR2.START bit, confirm that the written values can be read.
Note 2. After writing to the RCR2.HR24 bit, perform three dummy read operations.

Figure 5.16 RTC Initialization

5.8.9 IRQ9 Interrupt Handling

Figure 5.17 shows the IRQ9 Interrupt Handling.

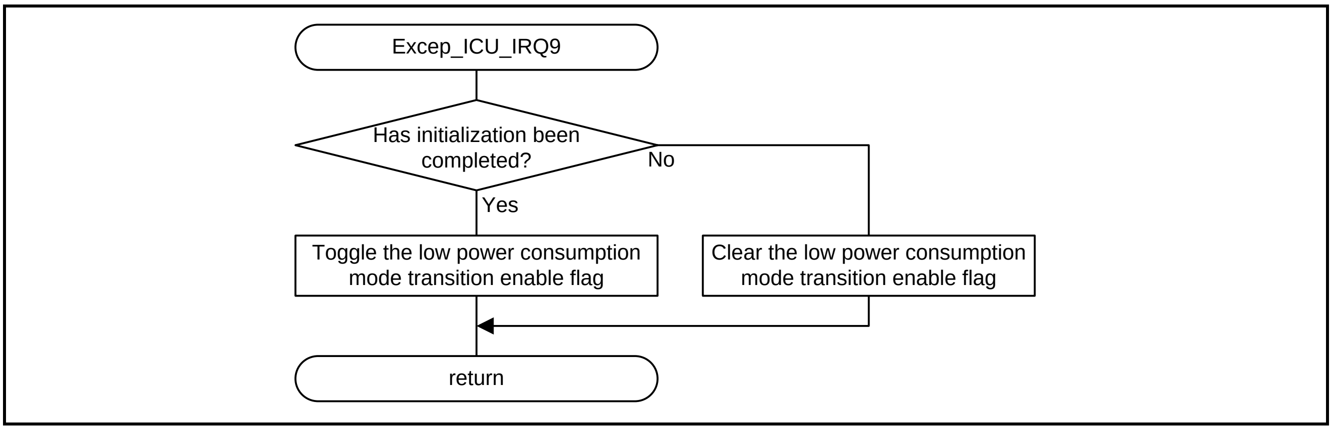


Figure 5.17 IRQ9 Interrupt Handling

5.8.10 RTC.PRD Interrupt Handling

Figure 5.18 shows the RTC.PRD Interrupt Handling.

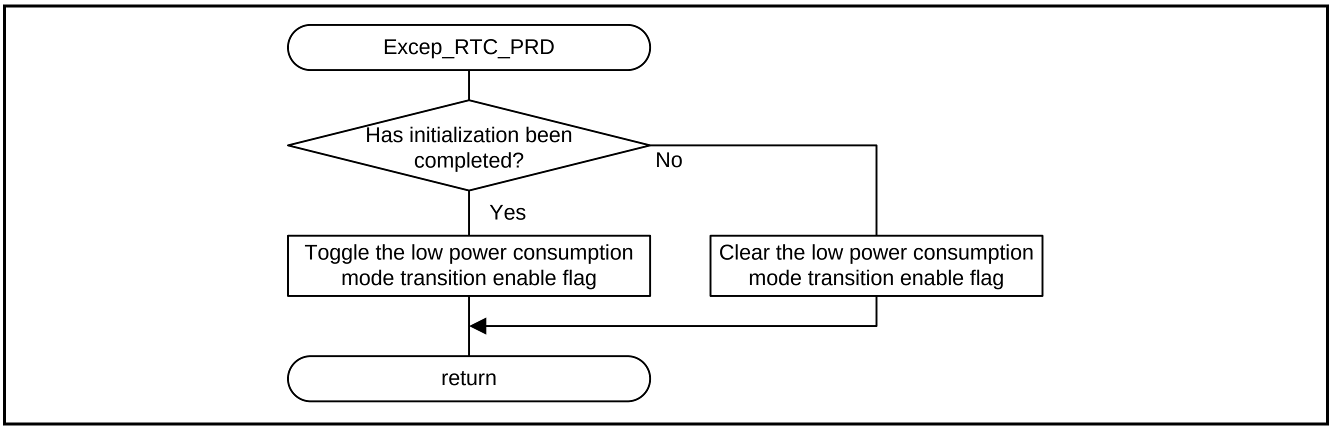


Figure 5.18 RTC.PRD Interrupt Handling

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

User's Manual: Hardware

RX63N Group, RX631 Group User's Manual: Hardware Rev.1.70 (R01UH0041EJ)
The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

User's Manual: Development Tools

RX Family C/C++ Compiler Package V.1.01 User's Manual Rev.1.00 (R20UT0570EJ)
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website

http://www.renesas.com

Inquiries

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

REVISION HISTORY	RX63N Group, RX631 Group Application Note Transition to Low Power Consumption Mode Using Low Power Consumption Function

Rev.	Date	Description	
		Page	**Summary**
1.00	Sep. 1, 2014	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

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

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

2. Processing at Power-on

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

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

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

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

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

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

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

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

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

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 906, 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 Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0