RX210, RX21A, and RX220 Groups

R01AN1482EJ0101
Rev. 1.01
July 1, 2014

Transition to Low Power Consumption Modes Using the Low Power Consumption Function

Abstract

This document describes transition to low power consumption modes using the low power consumption function in the RX210, RX21A, and RX220 Groups.

Products

- RX210, RX21A, and RX220 Groups

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

The sample code performs processing to enter and exit a low power consumption mode by specifying a source for transition and exit. The MCU enters and exits a low power consumption mode when the specified source occurs.

- Low power consumption modes:
  Selectable from sleep mode, software standby mode, and deep software standby mode
- Source to enter or exit low power consumption mode: Selectable from IRQ1, LVD, and RTC
- Clock source: Selectable (1)
- Operating power control mode: Selectable (1)
- Sleep mode return clock source switching function: Not used

Note:

1. The clock source and operating power control mode are selected in r_init_clock.h. This application note uses HOCO as the clock source and middle-speed operating mode 1A for the operating power control mode. Refer to RX210 Group Initial Setting Rev. 2.00 application note for details.

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 low power consumption mode. Exits low power consumption mode.
Voltage detection circuit (LVD)	Enters low power consumption mode. Exits low power consumption mode.
Realtime clock (RTC)	Enters low power consumption mode. Exits low power consumption mode.

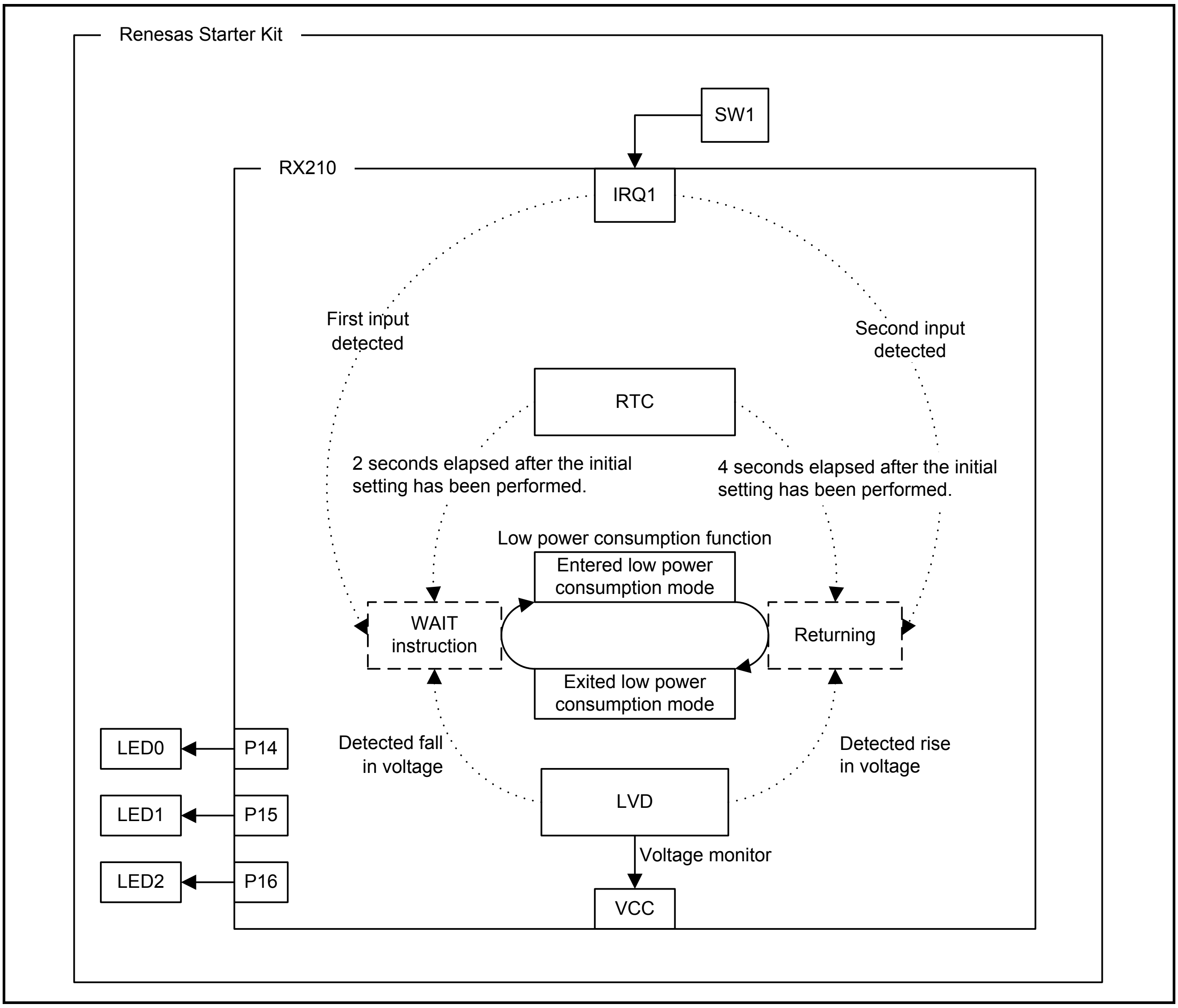


Figure 1.1 Block Diagram

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	R5F52108ADFP (RX210 Group)
Operating frequencies	- HOCO: 32 MHz - Sub-clock: 32.768 kHz - System clock (ICLK): 32 MHz (HOCO divided by 1) (1) - Peripheral module clock B (PCLKB): 16 MHz (HOCO divided by 2)
Operating voltage	1.8 to 5.0 V (1)
Integrated development environment	Renesas Electronics Corporation High-performance Embedded Workshop Version 4.09.01
C compiler	Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.1.02 Release 01
	Compile options -cpu=rx200 -output=obj="$(CONFIGDIR)\$(FILELEAF).obj" -debug -nologo (The default setting is used in the integrated development environment.)
iodefine.h version	Version 1.2A
Endian	Little endian
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Sample code version	Version 1.00
Board used	Renesas Starter Kit for RX210 (product part no.: R0K505210C000BE)

Note:

1. In this application note, middle-speed operating mode 1A is selected for the operating power control mode. Therefore when the operating voltage used is less than 1.8 V, set ICLK to less than or equal to 20 MHz.

3. Reference Application Notes

For additional information associated with this document, refer to the following application notes.

- RX210 Group Initial Setting Rev. 2.00 (R01AN1002EJ)
- RX21A Group Initial Setting Rev. 1.10 (R01AN1486EJ)
- RX220 Group Initial Setting Rev. 1.10 (R01AN1494EJ)

The initial setting functions in the reference application notes are used in the sample code in this application note. The revision numbers of the reference application notes are current as of when this application note was made. However the latest version is always recommended. Visit the Renesas Electronics Corporation website to check and download the latest version.

4. Hardware

4.1 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

The pins described here are for 100-pin products. When the product with less than 100-pin is used, select appropriate pins for the product used.

Table 4.1 Pins Used and Their Functions

Pin Name	I/O	Function
P31/IRQ1	Input	SW1 input (for entering or exiting low power consumption mode)
P14	Output	LED0 output (turned on after the initial setting has been performed)
P15	Output	LED1 output (turned on before entering low power consumption mode)
P16	Output	LED2 output (turned on after exiting low power consumption mode)

5. Software

When the source for entering a low power consumption mode occurs, the MCU enters a low power consumption mode. When in a low power consumption mode, if the source for exit occurs, the MCU exits the mode it is in.

Sleep mode, software standby mode, or deep software standby mode can be selected as the low power consumption mode. The IRQ, LVD, or RTC can be used for the source to enter or exit a low power consumption mode.

Settings for the peripheral functions are as follows:

IRQ

- Detection method: Falling edge on the IRQ1 pin
- Digital filter: Disabled
- Interrupt priority level: Level 15

LVD

- Condition for LVD1 detection: VCC passed upward through Vdet1 (4.15 V)
- Condition for LVD2 detection: VCC passed downward through Vdet2 (3.40 V)
- Processing when LVD1 is detected: Voltage monitoring 1 interrupt (maskable)
- Processing when LVD2 is detected: Voltage monitoring 2 interrupt (maskable)
- Digital filter: Disabled
- Interrupt priority level: Level 15

RTC

- Initial date and time setting: 2013-01-01 (Tue.) 00:00:00
- Time mode: 24-hour mode
- Interrupt: Periodic interrupt (PRD) is used and generated every 2 seconds.
- Interrupt priority level: Level 15

5.1 Operation Overview

5.1.1 Sleep Mode

After the initial setting has been performed, turns on LED0 and waits until the source for transition to sleep mode occurs. When the source occurs, turns off LED0, turns on LED1, and enters sleep mode. When the source for exit occurs during sleep mode, exits sleep mode, turns off LED1, and turns on LED2.

Figure 5.1 shows the Operation Overview when Entering and Exiting Sleep Mode.

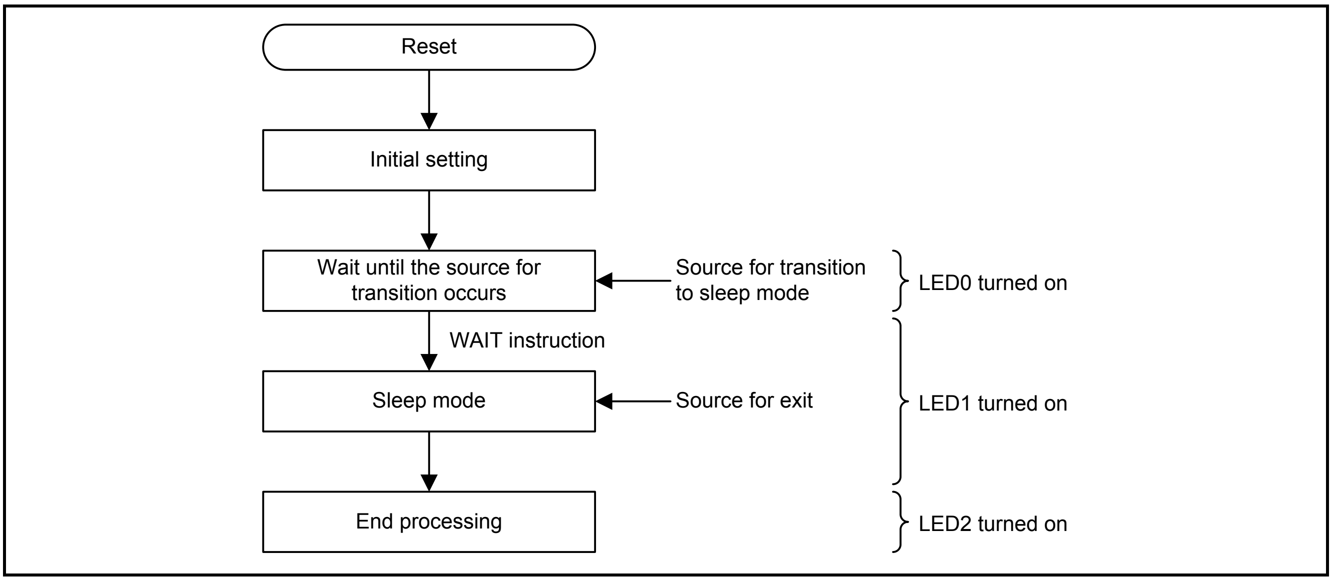


Figure 5.1 Operation Overview when Entering and Exiting Sleep Mode

5.1.2 Software Standby Mode

After the initial setting has been performed, turns on LED0 and waits until the source for transition to software standby mode occurs. When the source occurs, turns off LED0, turns on LED1, and enters software standby mode. When the source for exit occurs during software standby mode, exits software standby mode, turns off LED1, and turns on LED2.

Figure 5.2 shows the Operation Overview when Entering and Exiting Software Standby Mode.

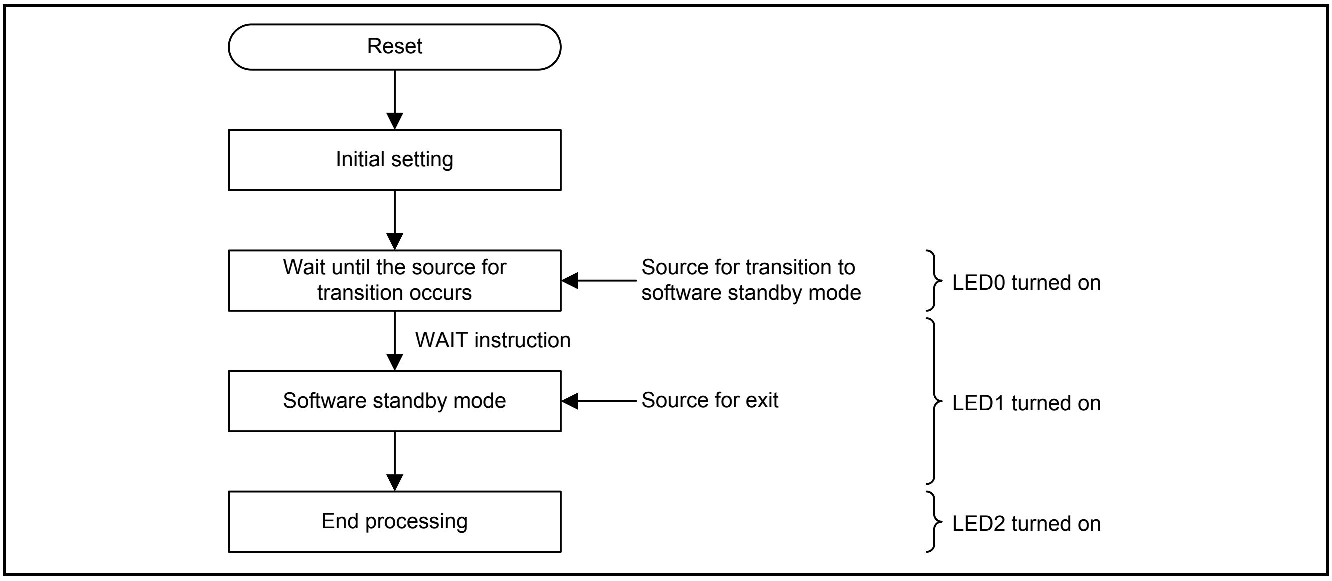
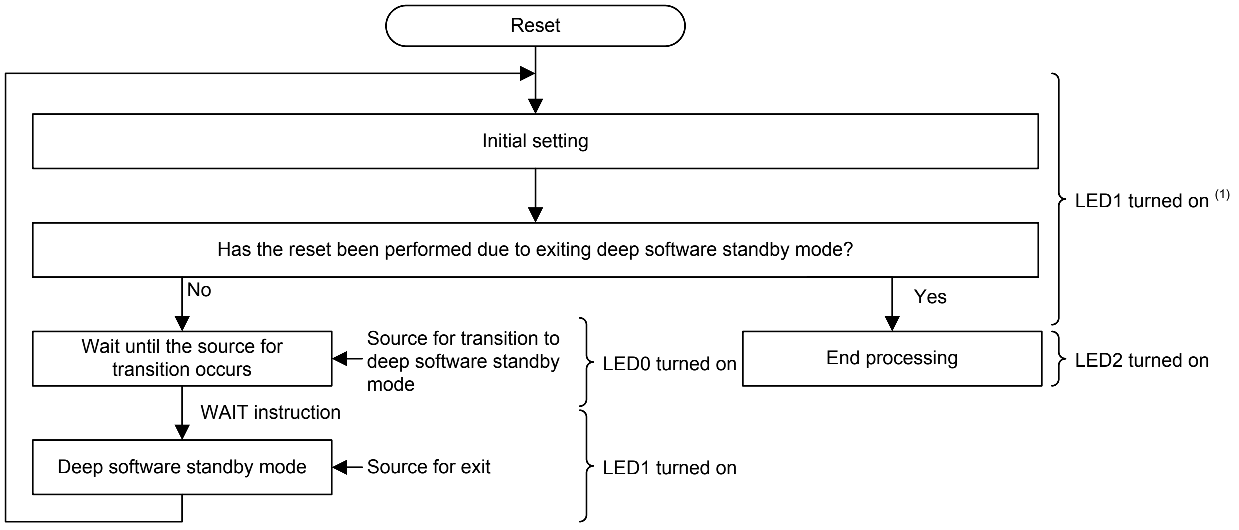


Figure 5.2 Operation Overview when Entering and Exiting Software Standby Mode

5.1.3 Deep Software Standby Mode

After the initial setting has been performed, turns on LED0 and waits until the source for transition to deep software standby mode occurs. When the source occurs, turns off LED0, turns on LED1, and enters deep software standby mode. When the source for exit occurs during deep software standby mode, exits deep software standby mode, and performs a reset. After the reset, performs the initial setting and turns on LED2.

Figure 5.3 shows the Operation Overview when Entering and Exiting Deep Software Standby Mode.



Note:

1. LED1 is turned on only when the reset is executed due to exiting deep software standby mode (deep software standby reset).

Figure 5.3 Operation Overview when Entering and Exiting Deep Software Standby Mode

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

When using the IRQ for the source to enter and exit a low power consumption mode, the MCU enters or exits a low power consumption mode by the IRQ interrupt request generation.

The IRQ interrupt request generated during a wait period for transition to a low power consumption mode becomes the source for transition, and the IRQ interrupt request generated during the low power consumption mode becomes the source for exit.

Figure 5.4 shows the Timing for Entering and Exiting a Low Power Consumption Mode Using the IRQ.

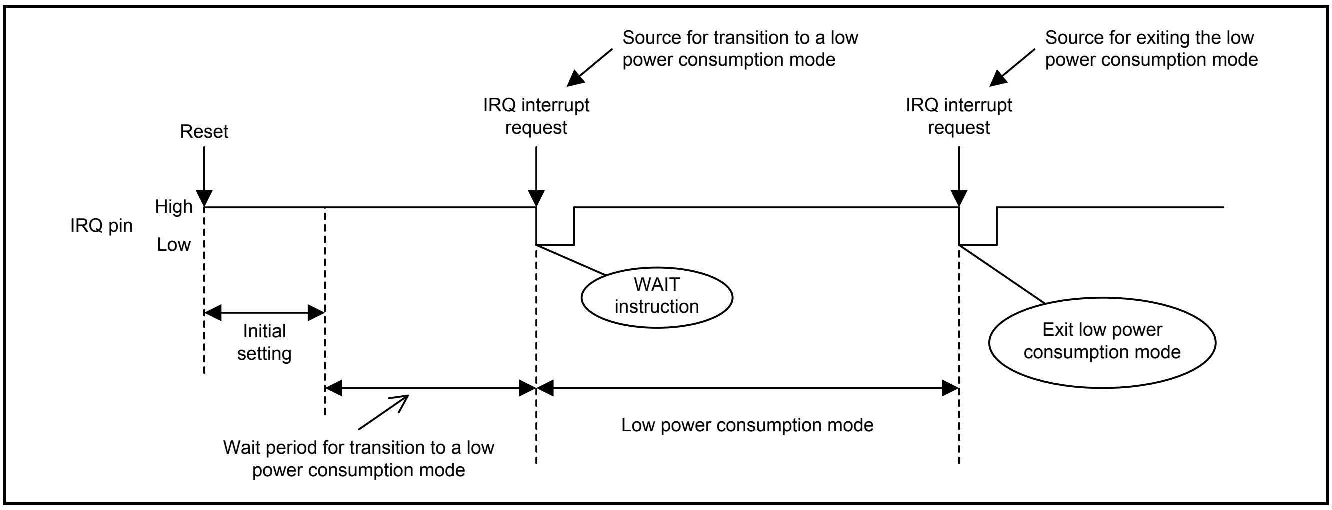


Figure 5.4 Timing for Entering and Exiting a Low Power Consumption Mode Using the IRQ

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

When using the RTC for the source to enter and exit a low power consumption mode, the MCU enters or exits a low power consumption mode by the RTC.PRD interrupt request generated every 2 seconds.

The RTC.PRD interrupt request generated during a wait period for transition to a low power consumption mode becomes the source for transition, and the RTC.PRD interrupt request generated during the low power consumption mode becomes the source for exit.

Figure 5.5 shows the Timing of Entering and Exiting a Low Power Consumption Mode Using the RTC.

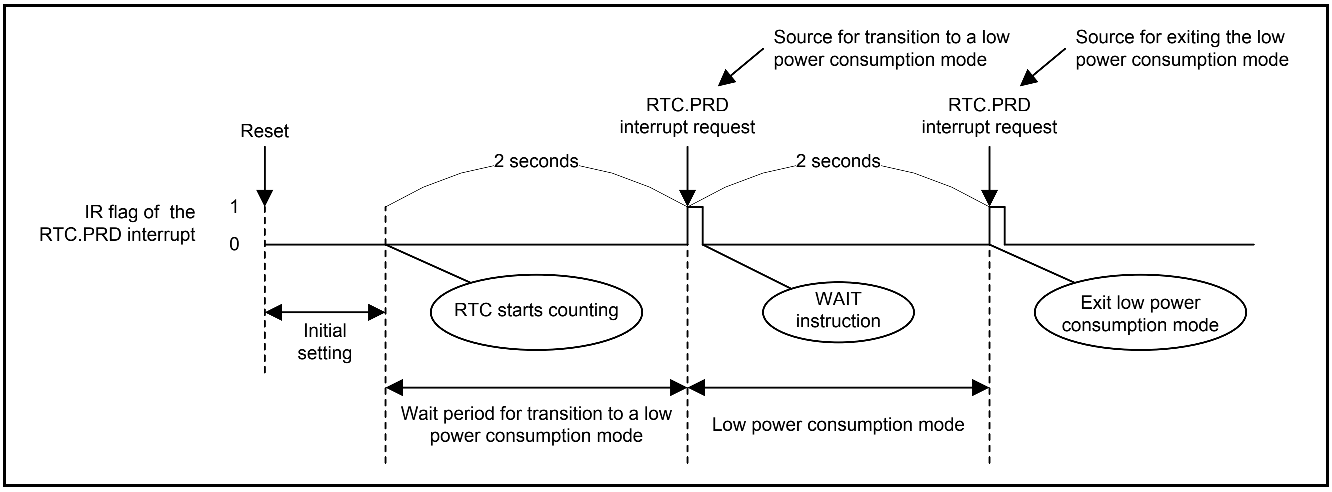


Figure 5.5 Timing of Entering and Exiting a Low Power Consumption Mode Using the RTC

5.1.6 Entering and Exiting a Low Power Consumption Mode Using the LVD

When using the LVD for the source to enter and exit a low power consumption mode, the MCU enters a low power consumption mode by the LVD2 interrupt request generation and exits the low power consumption mode by the LVD1 interrupt request generation. The LVD2 interrupt request is generated when ‘VCC < Vdet2’ is detected. The LVD1 interrupt request is generated when ‘VCC ≥ Vdet1’ is detected.

Figure 5.6 shows the Timing of Entering and Exiting a Low Power Consumption Mode Using the LVD.

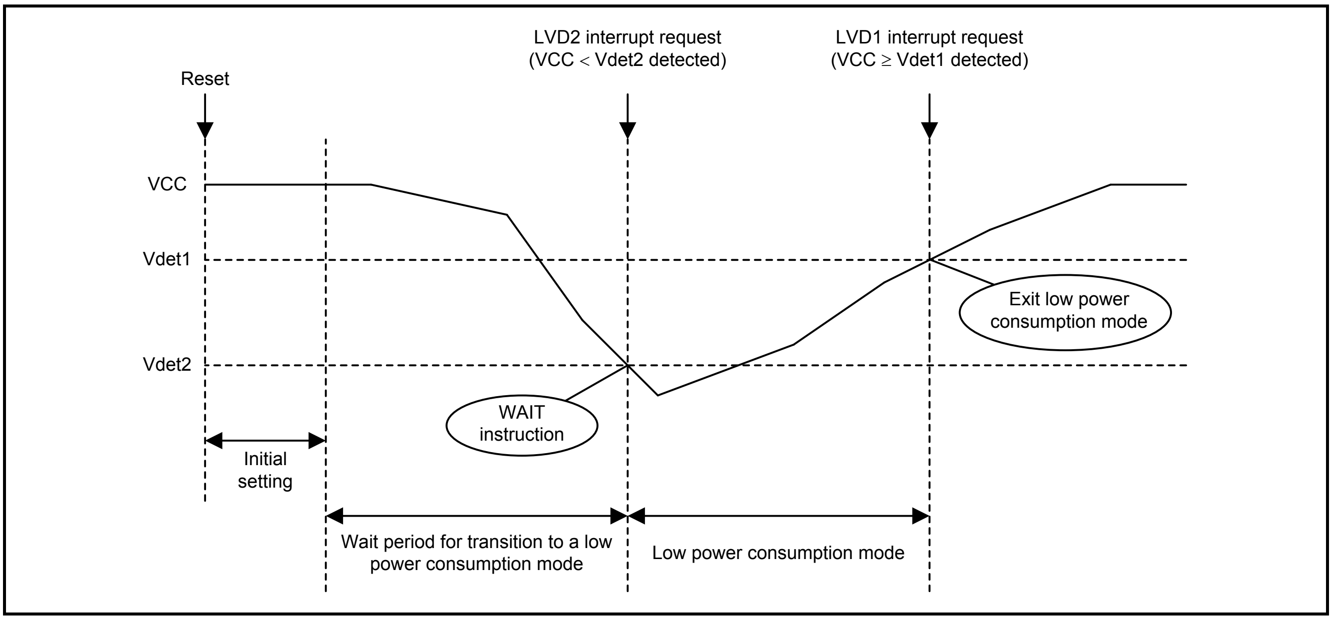


Figure 5.6 Timing of Entering and Exiting a Low Power Consumption Mode Using the LVD

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 processing for active peripheral functions after a reset	
r_init_stop_module.h	Header file for r_init_stop_module.c	
r_init_non_existent_port.c	Nonexistent port initialization	
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	

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	The IWDT is stopped after a reset. The WDT is stopped after a reset.
OFS1	FFFF FF8Bh to FFFF FF88h	FFFF FFFFh	The voltage monitor 0 reset is disabled after a reset. HOCO oscillation is disabled after a reset.
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	Source to enter and exit a low power consumption mode: IRQ
L_LVD	1	Source to enter and exit a low power consumption mode: LVD
L_RTC	2	Source to enter and exit a low power consumption mode: RTC
L_SOURCE	L_IRQ	Selection of the source to enter and exit a low power consumption mode: 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	Selection of low power consumption mode: Sleep mode
WAIT_tdEA	480	td(E-A) wait time (max. 15 μs) Wait time ÷ ICLK (32 MHz) cycles = 15 ÷ 0.03125 = 480

5.5 Variables

Table 5.4 lists the Global Variables.

Table 5.4 Global Variables

Type	Variable Name	Contents	Function Used
uint8_t	initial_end	Initial setting end flag 0: Processing 1: Completed	sleep_mode software_standby_mode deep_standby_mode Excep_ICU_IRQ1 Excep_LVD_LVD2 Excep_RTC_PRD
uint8_t	enable_low_power	Enable flag for transition to a low power consumption mode 0: Transition disabled 1: Transition enabled	

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 processing for active peripheral functions after a reset
R_INIT_NonExistentPort	Nonexistent port initialization
R_INIT_Clock	Clock initialization
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
lvd_init	LVD initialization
rtc_init	RTC initialization
Excep_ICU_IRQ1	IRQ1 interrupt handling
Excep_LVD_LVD1	LVD1 interrupt handling
Excep_LVD_LVD2	LVD2 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	Enters a low power consumption mode after the initial setting has been performed.
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 processing for active peripheral functions after a reset
Header	r_init_stop_module.h
Declaration	void R_INIT_StopModule(void)
Description	Configures the setting to enter the module-stop state.
Arguments	None
Return Value	None
Remarks	Transition to the module-stop state is not performed in the sample code. For details on this function, refer to the Initial Setting application note for the product used.

R_INIT_NonExistentPort	
Outline	Nonexistent port initialization
Header	r_init_non_existent_port.h
Declaration	void R_INIT_NonExistentPort(void)
Description	Initializes port direction registers for ports that do not exist in products with less than 100 pins.
Arguments	None
Return Value	None
Remarks	The number of pins in the sample code is set for the 100-pin package (PIN_SIZE=100). After this function is called, when writing in byte units to the PDR registers or PODR registers which have nonexistent ports, set the corresponding bits for nonexistent ports as follows: set the I/O select bits in the PDR registers to 1 and set the output data store bits in the PODR registers to 0. For details on this function, refer to the Initial Setting application note for the product used.

R_INIT_Clock	
Outline	Clock initialization
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Initializes the clock.
Arguments	None
Return Value	None
Remarks	The sample code selects processing with the following settings: - System clock: HOCO - Operating power control mode: Middle-speed operating mode 1A - Main clock and PLL: Not used For details on this function, refer to the Initial Setting application note for the product used.

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

sleep_mode	
Outline	Transition to sleep mode
Header	None
Declaration	void sleep_mode(void)
Description	Configures settings 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	Configures settings 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	Configures settings to enter deep software standby mode.
Arguments	None
Return Value	None

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

lvd_init	
Outline	LVD initialization
Header	None
Declaration	void lvd_init(void)
Description	Performs the LVD initialization.
Arguments	None
Return Value	None

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

Excep_ICU_IRQ1	
Outline	IRQ1 interrupt handling
Header	None
Declaration	void Excep_ICU_IRQ1(void)
Description	Performs the IRQ1 interrupt handling.
Arguments	None
Return Value	None

Excep_LVD_LVD1	
Outline	LVD1 interrupt handling
Header	None
Declaration	void Excep_LVD_LVD1(void)
Description	Performs the LVD1 interrupt handling.
Arguments	None
Return Value	None

Excep_LVD_LVD2	
Outline	LVD2 interrupt handling
Header	None
Declaration	void Excep_LVD_LVD2(void)
Description	Performs the LVD2 interrupt handling.
Arguments	None
Return Value	None

Excep_RTC_PRD	
Outline	RTC.PRD interrupt handling
Header	None
Declaration	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 5.7 shows the Main Processing.

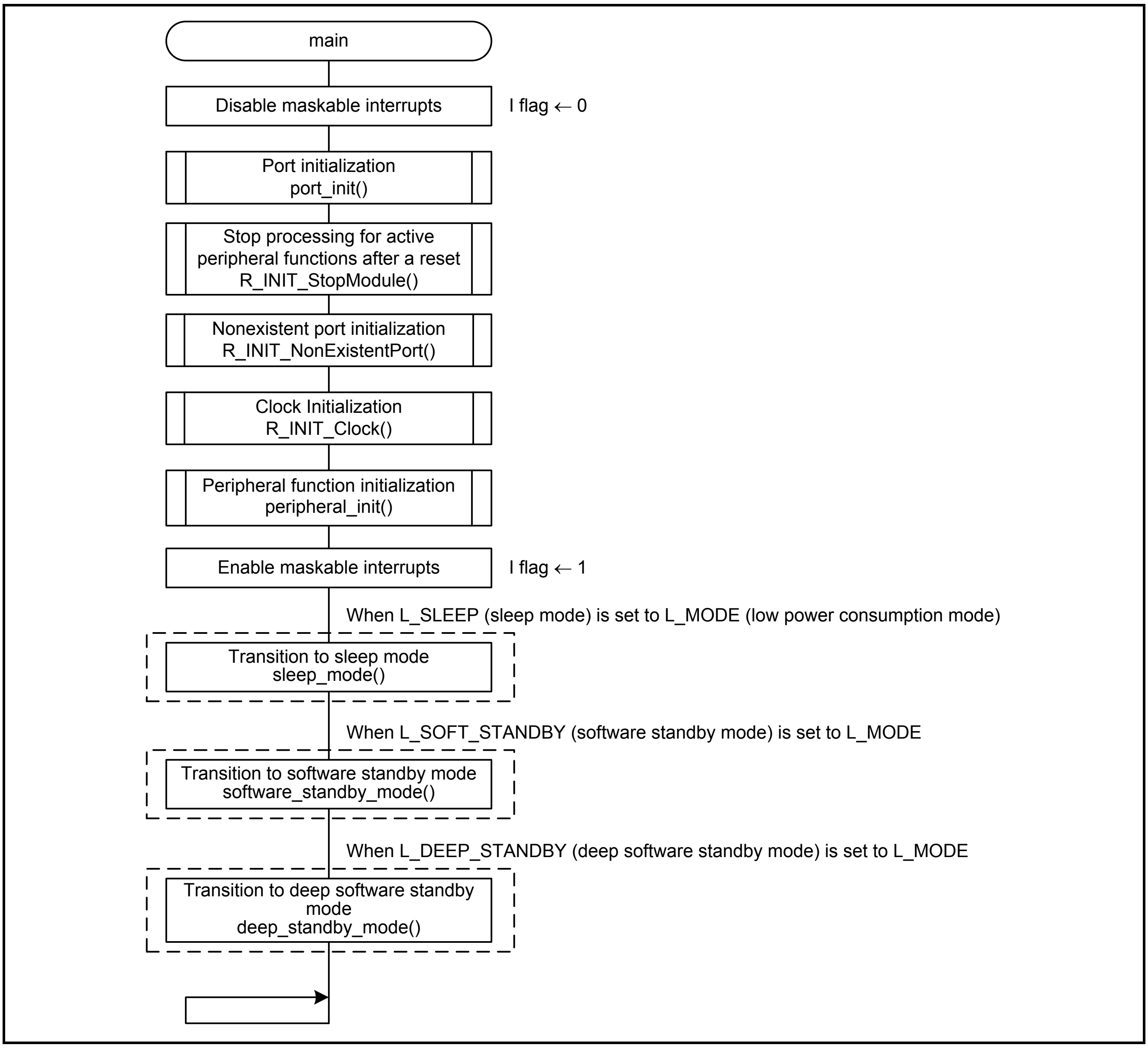


Figure 5.7 Main Processing

5.8.2 Port Initialization

Figure 5.8 shows the Port Initialization.

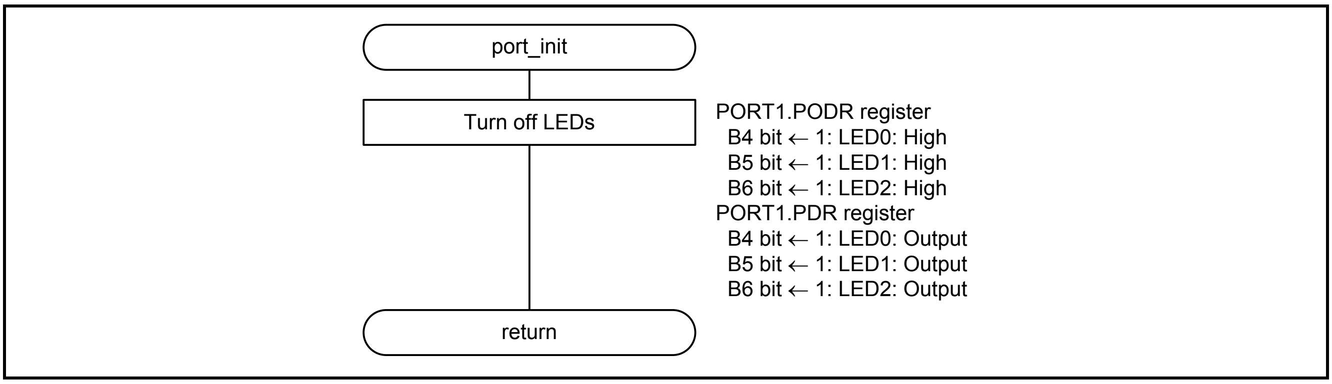


Figure 5.8 Port Initialization

5.8.3 Peripheral Function Initialization

Figure 5.9 shows the Peripheral Function Initialization.

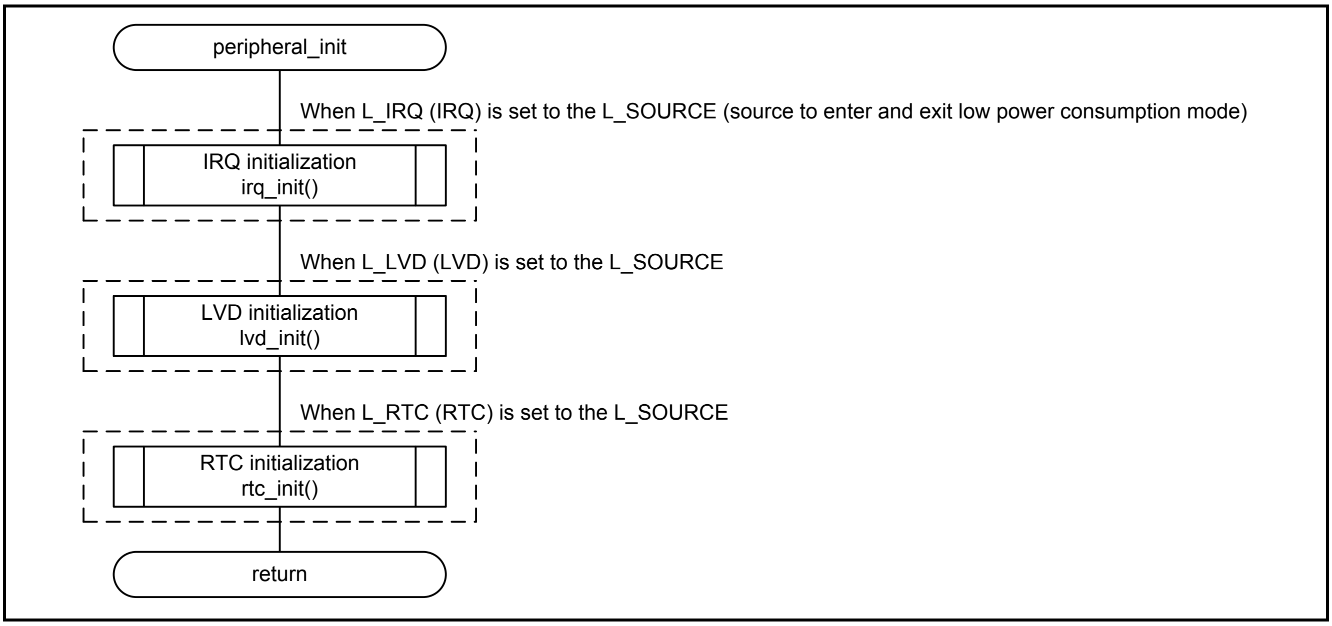
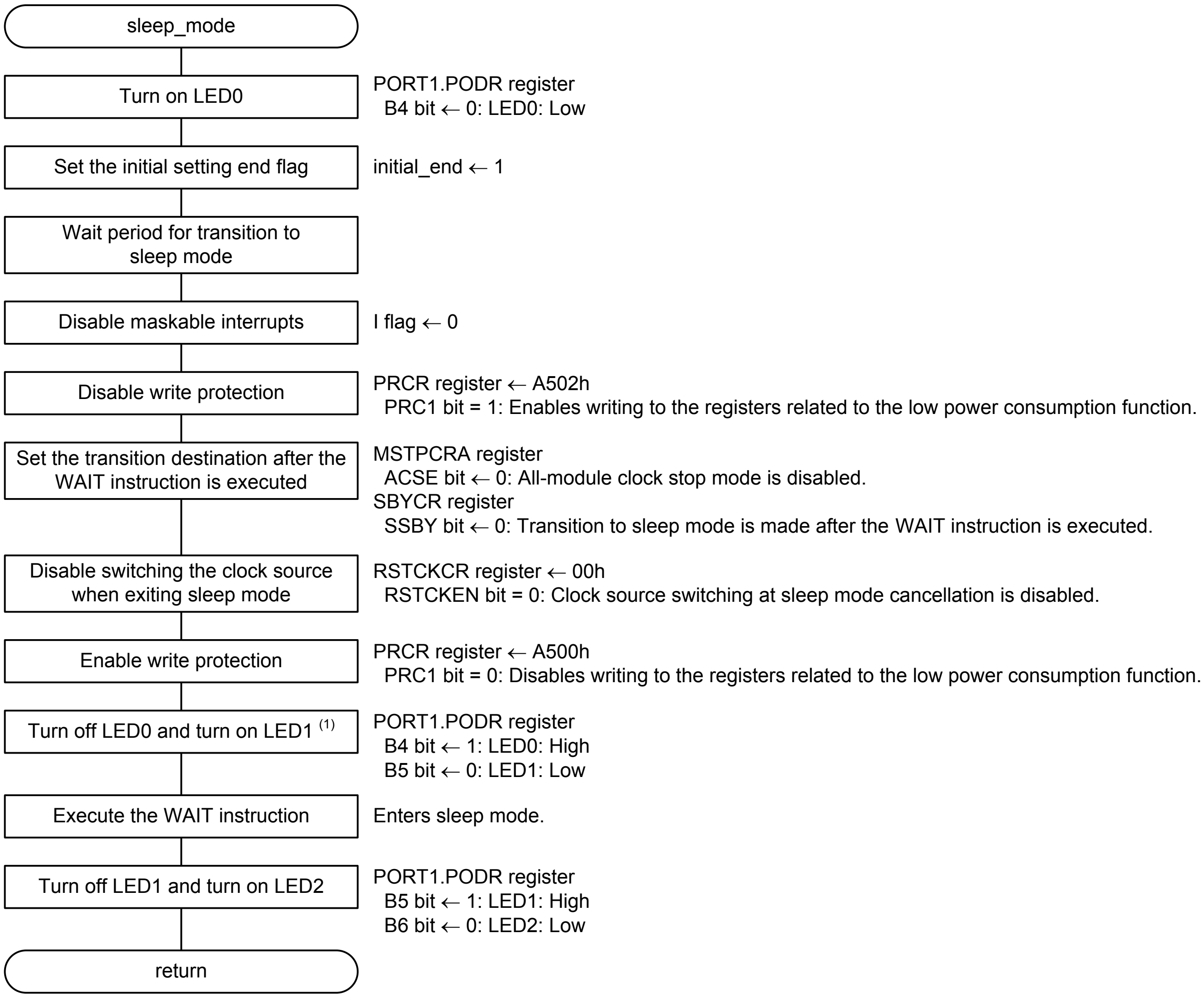


Figure 5.9 Peripheral Function Initialization

5.8.4 Transition to Sleep Mode

Figure 5.10 shows the Transition to Sleep Mode.



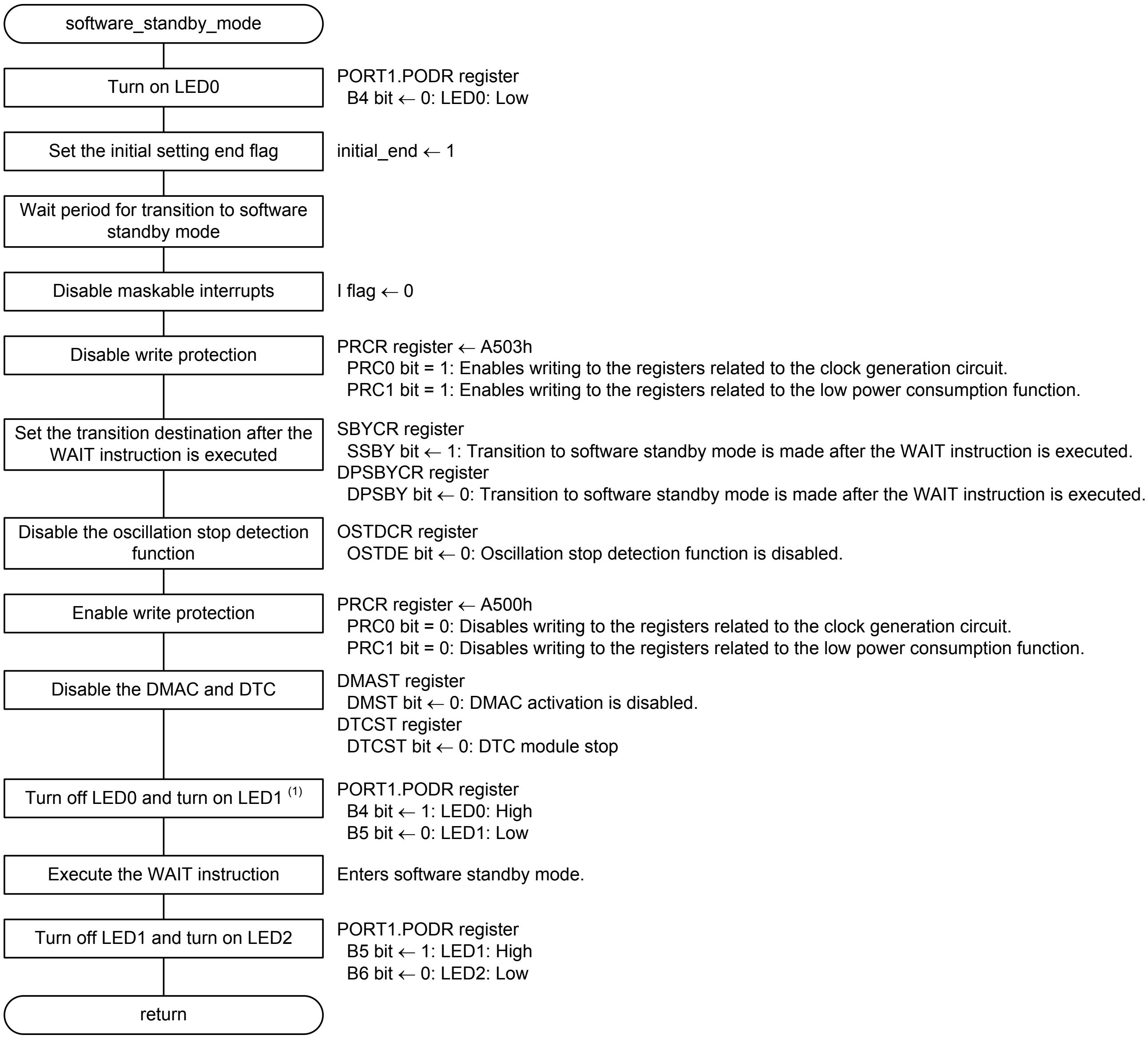
Note:

1. Read the register written immediately before the WAIT instruction is executed to confirm the written value can be read correctly.

Figure 5.10 Transition to Sleep Mode

5.8.5 Transition to Software Standby Mode

Figure 5.11 shows the Transition to Software Standby Mode.



Note:
1. Read the register written immediately before the WAIT instruction is executed to confirm the written value can be read correctly.

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.

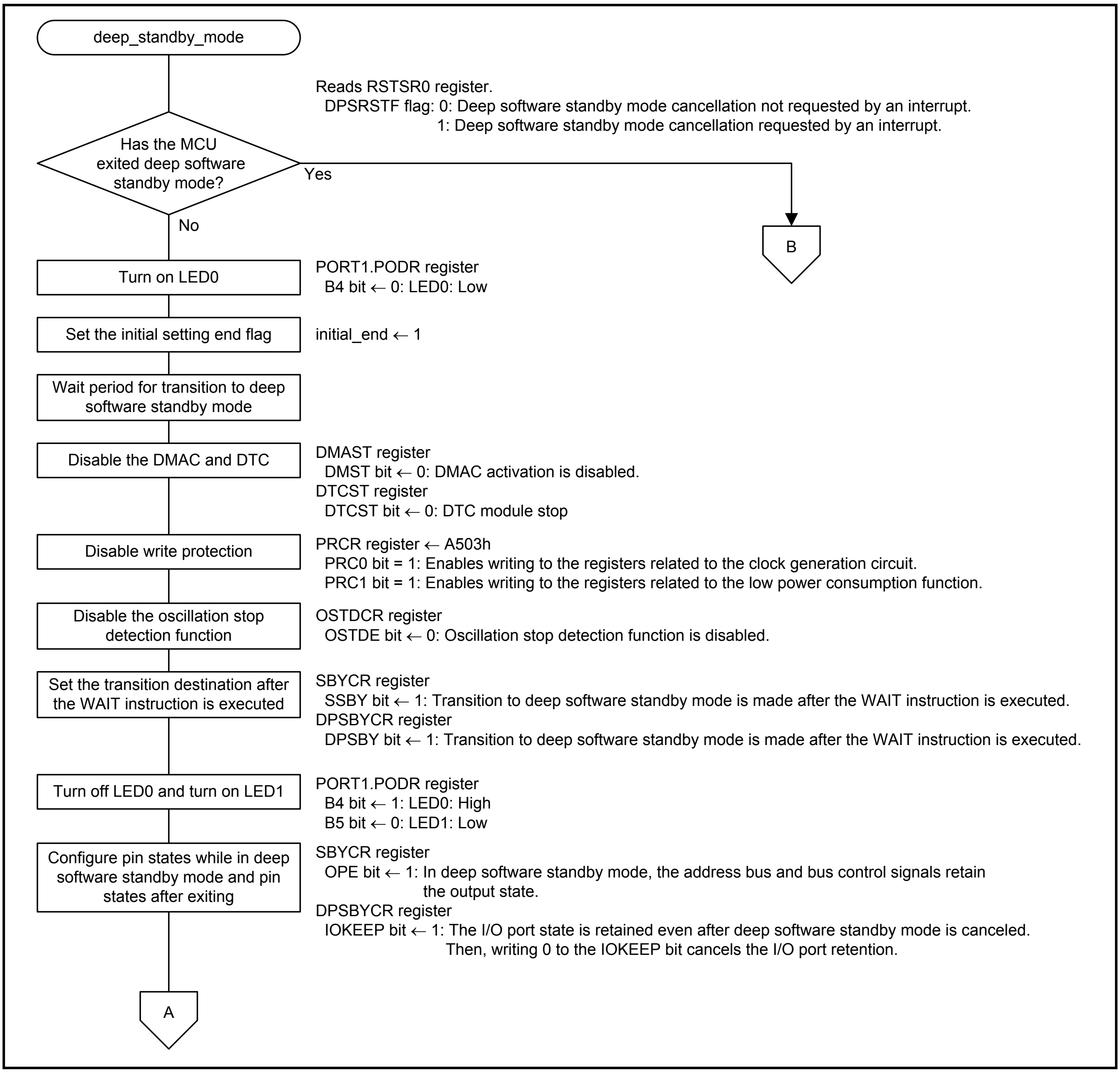
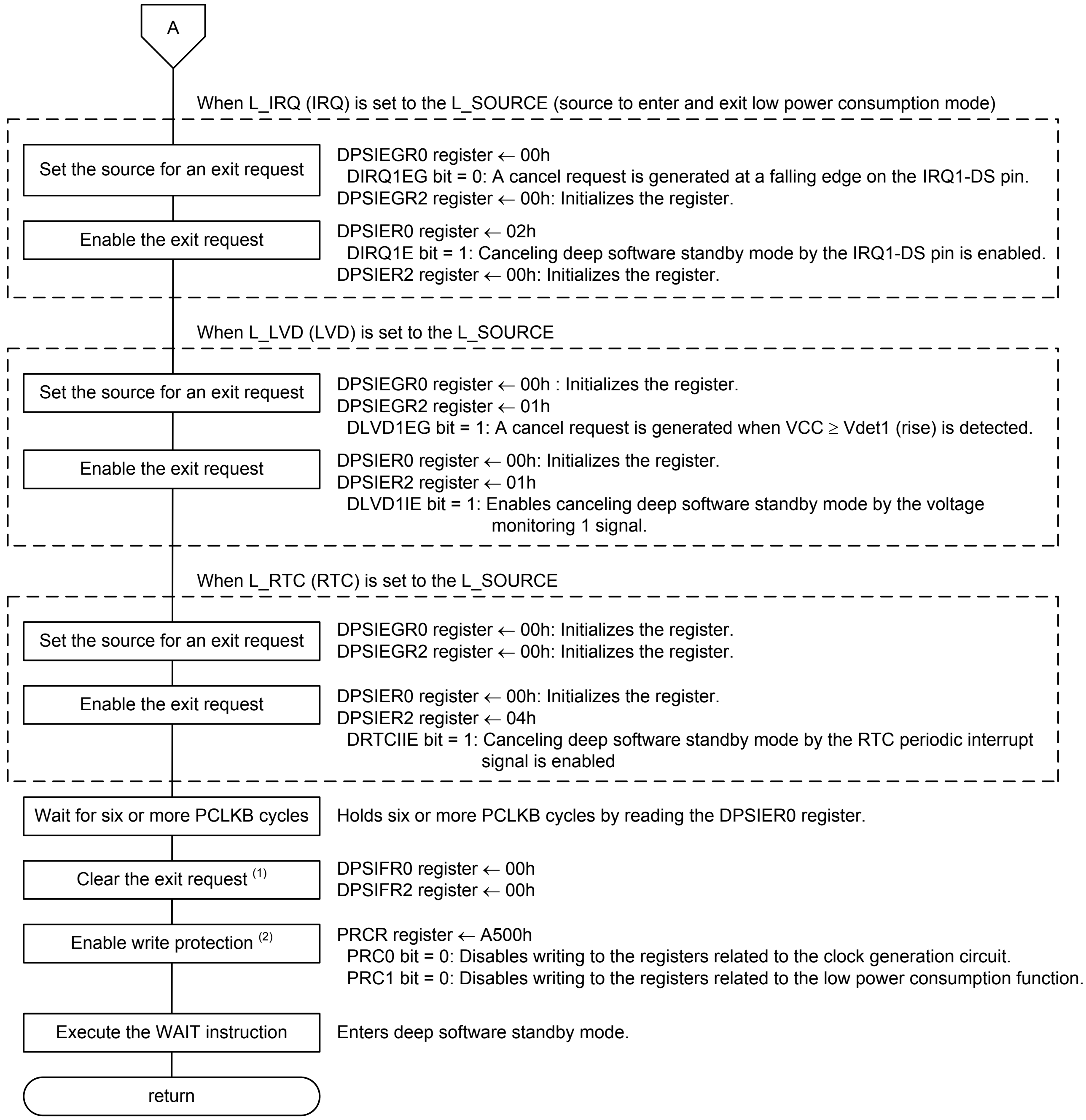
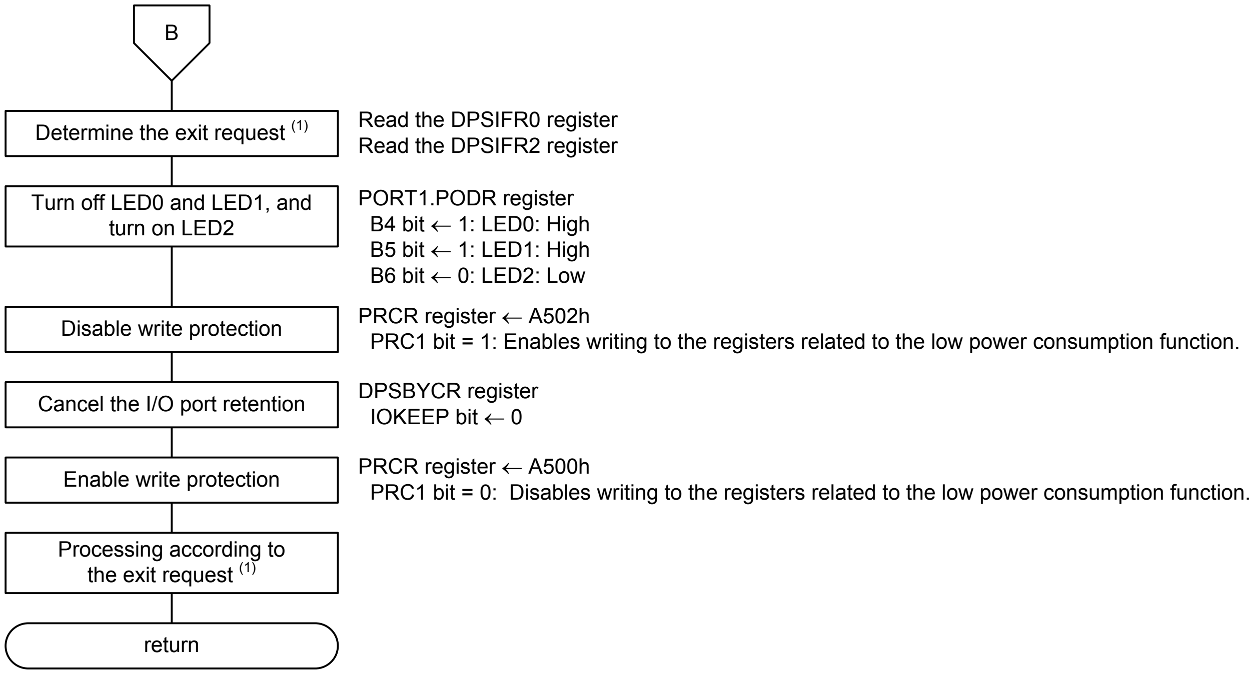


Figure 5.12 Transition to Deep Software Standby Mode (1/3)



Notes:
1. After reading registers DPSIFR0 and DPSIFR2, write 00h to these registers.
2. Read the register written immediately before the WAIT instruction is executed to confirm the written value can be read correctly.

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



Note:

1. Processing for exit requests is not performed in the sample code. Add a program as required.

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

5.8.7 IRQ Initialization

Figure 5.15 shows the IRQ Initialization.

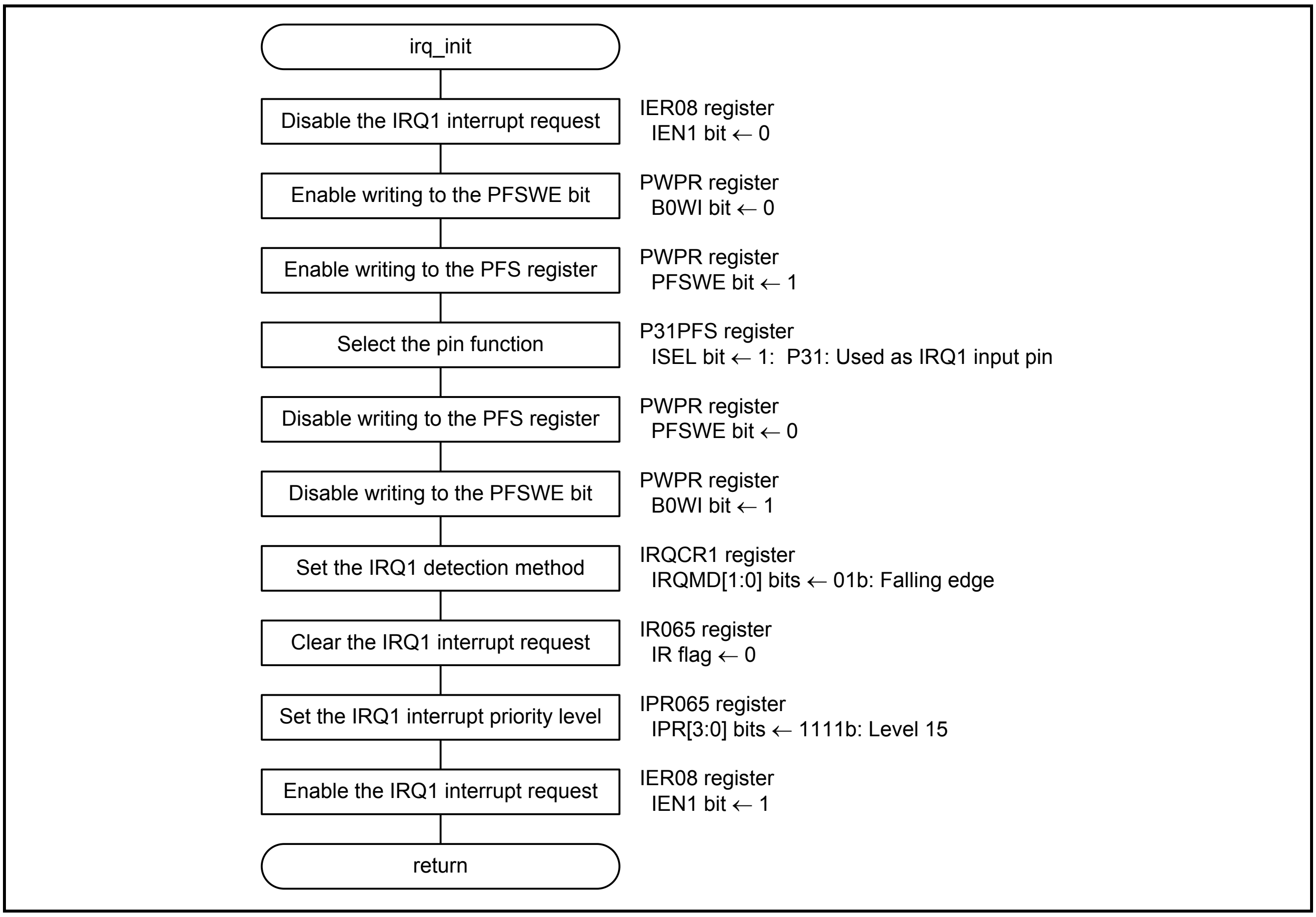


Figure 5.15 IRQ Initialization

5.8.8 LVD Initialization

Figure 5.16 and Figure 5.17 show the LVD Initialization.

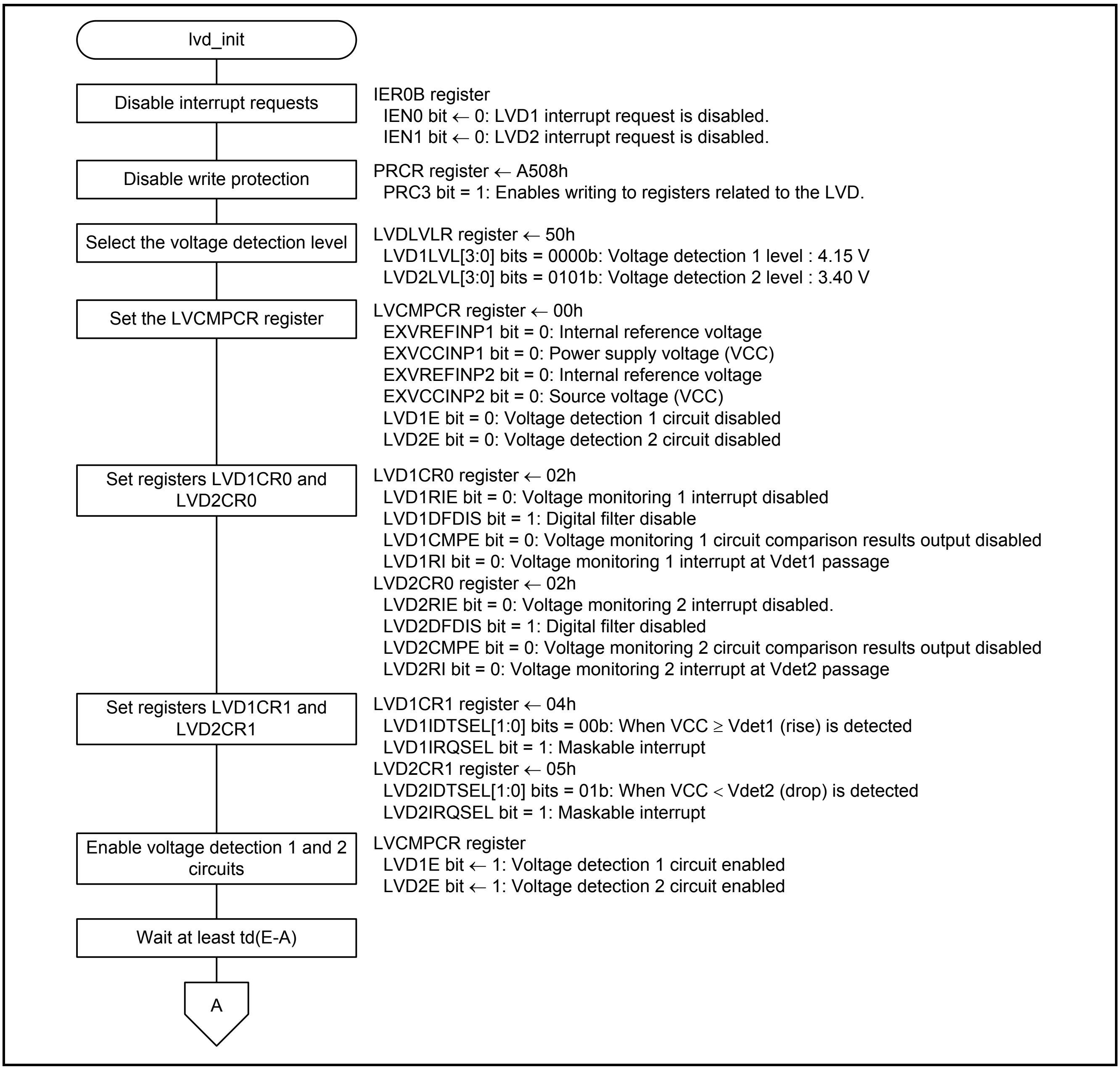


Figure 5.16 LVD Initialization (1/2)

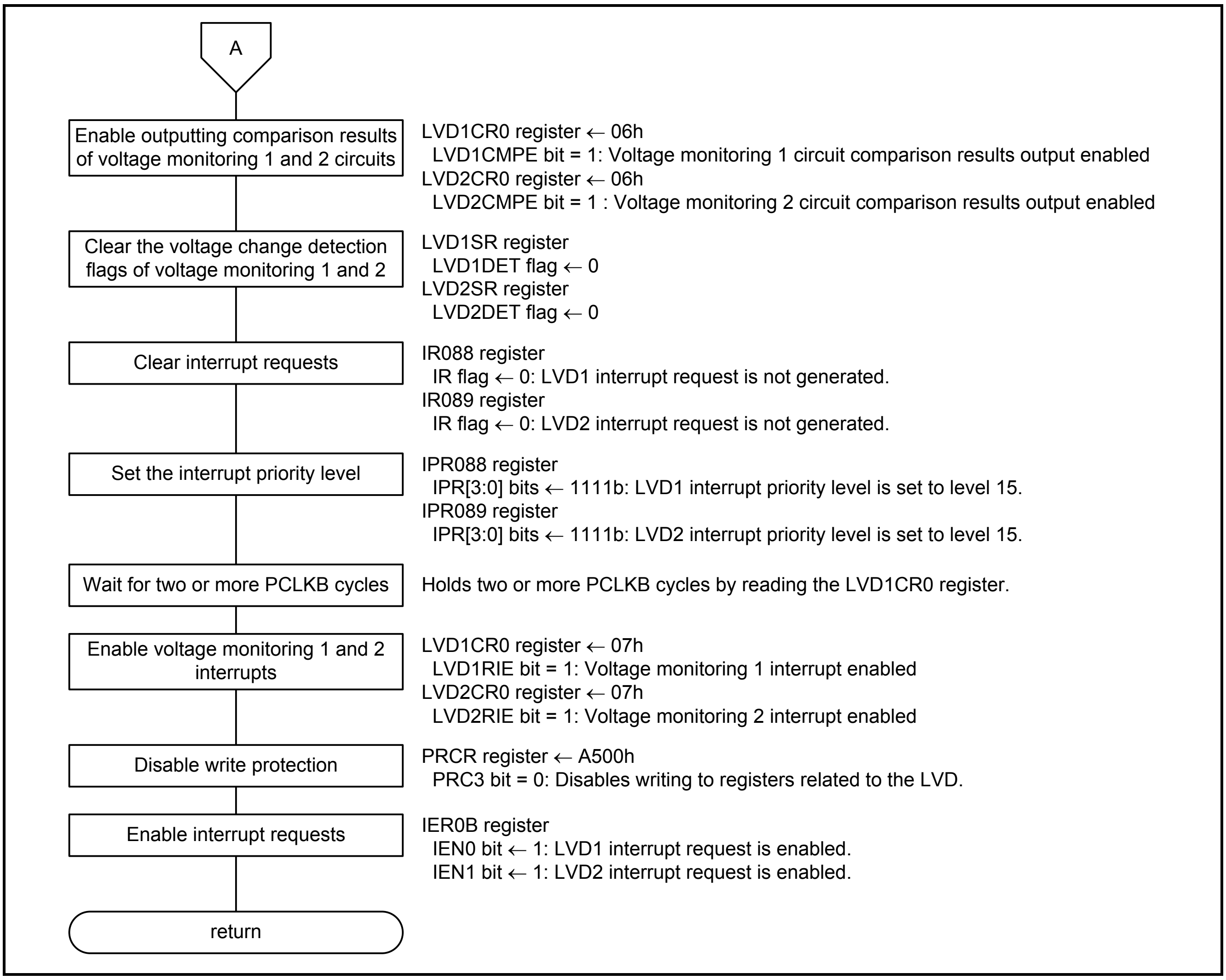
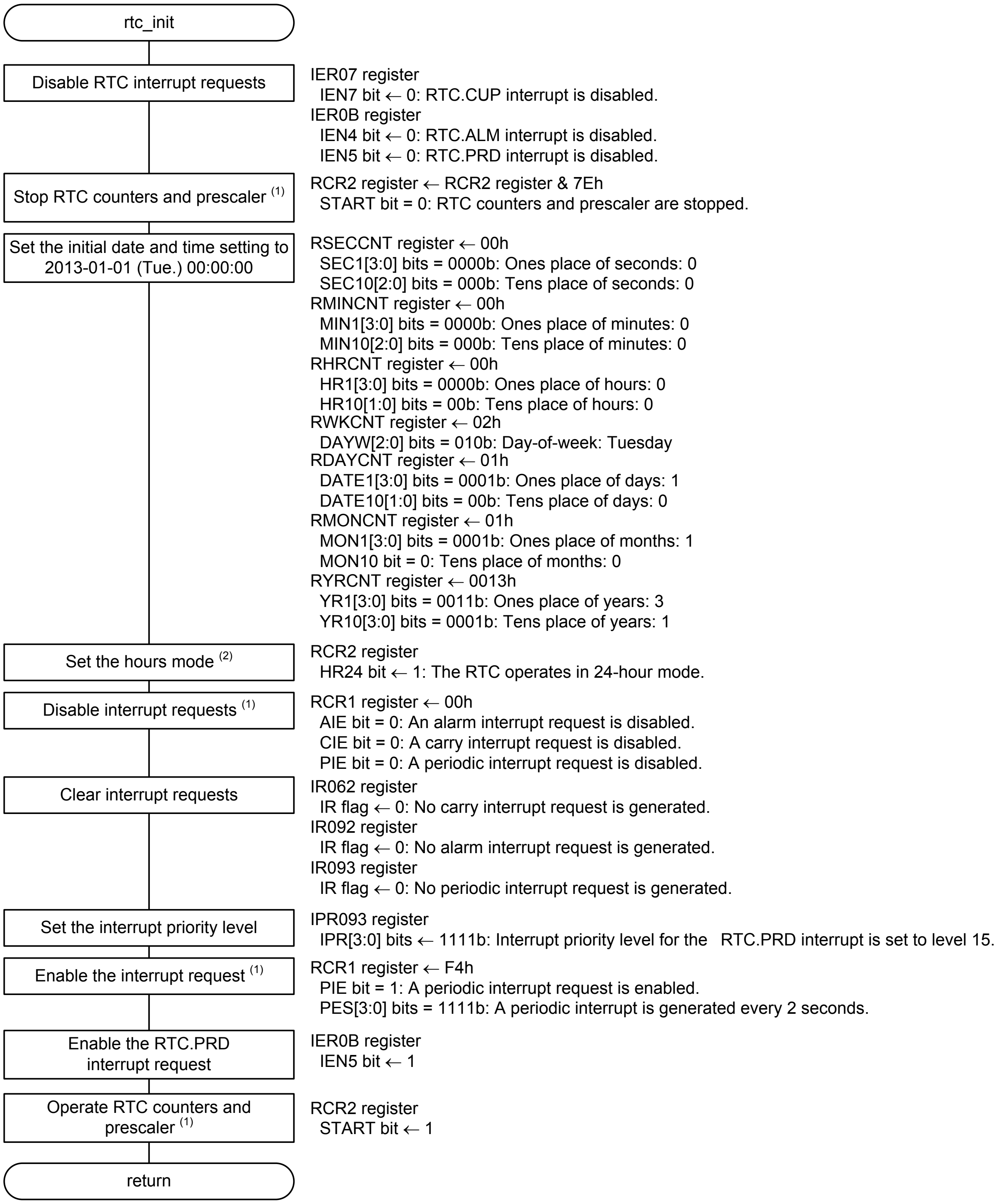


Figure 5.17 LVD Initialization (2/2)

5.8.9 RTC Initialization

Figure 5.18 shows the RTC Initialization.



Notes:
1. After writing to the RCR1 register and RCR2.START bit, confirm that the written value can be read correctly.
2. After writing to the RCR2.HR24 bit, dummy read the register three times.

Figure 5.18 RTC Initialization

5.8.10 IRQ1 Interrupt Handling

Figure 5.19 shows the IRQ1 Interrupt Handling.

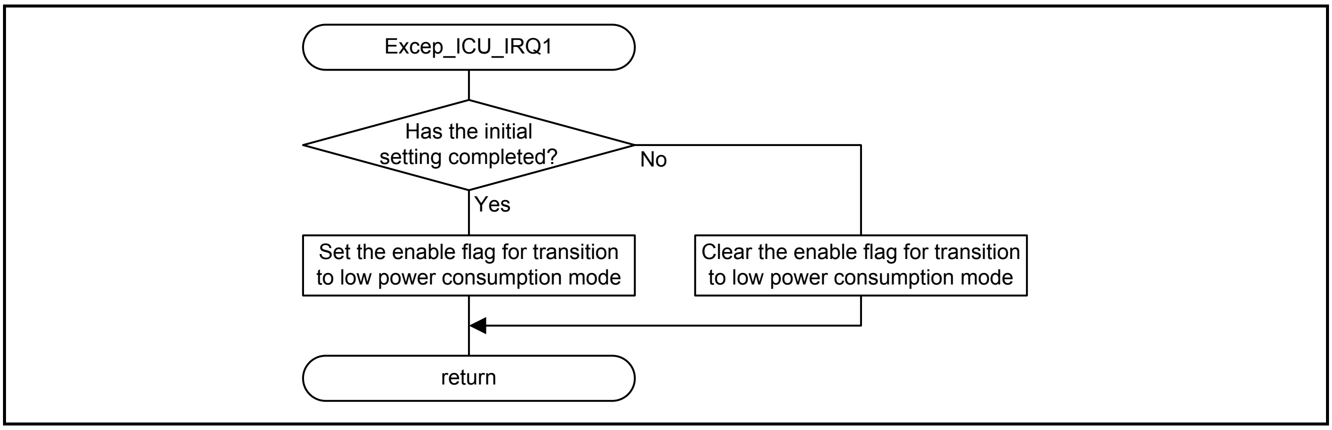


Figure 5.19 IRQ1 Interrupt Handling

5.8.11 LVD1 Interrupt Handling

Figure 5.20 shows the LVD1 Interrupt Handling.

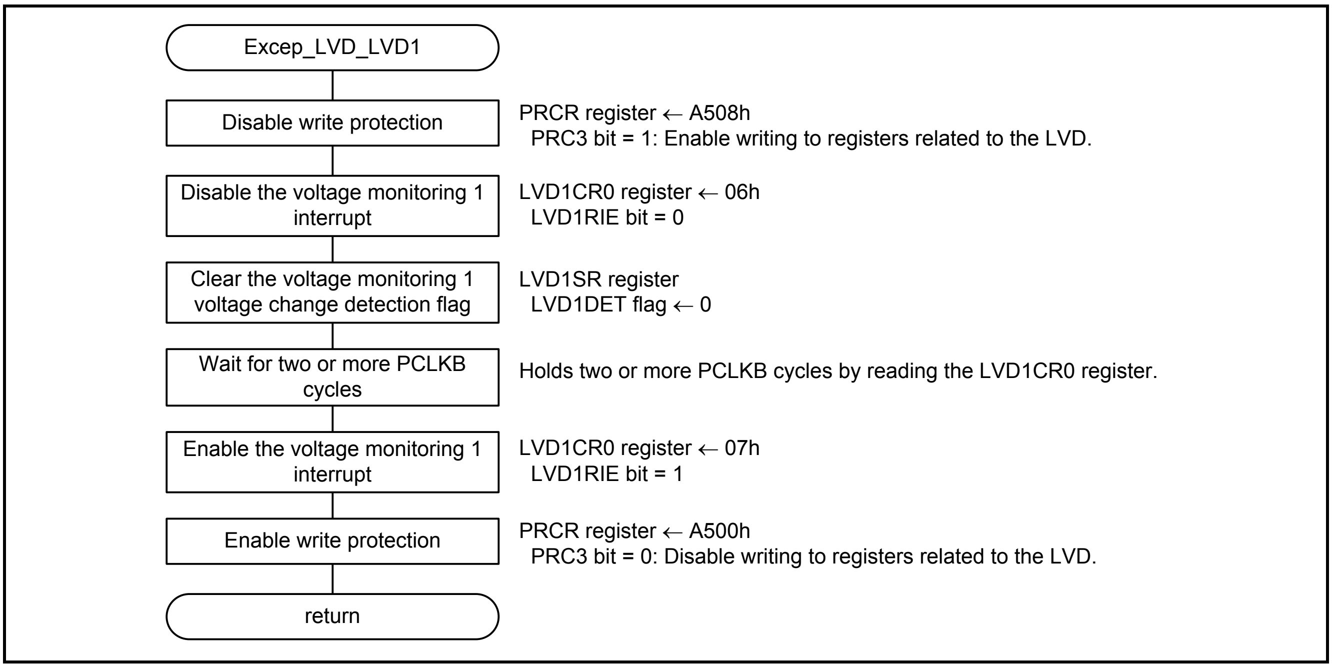


Figure 5.20 LVD1 Interrupt Handling

5.8.12 LVD2 Interrupt Handling

Figure 5.21 shows the LVD2 Interrupt Handling.

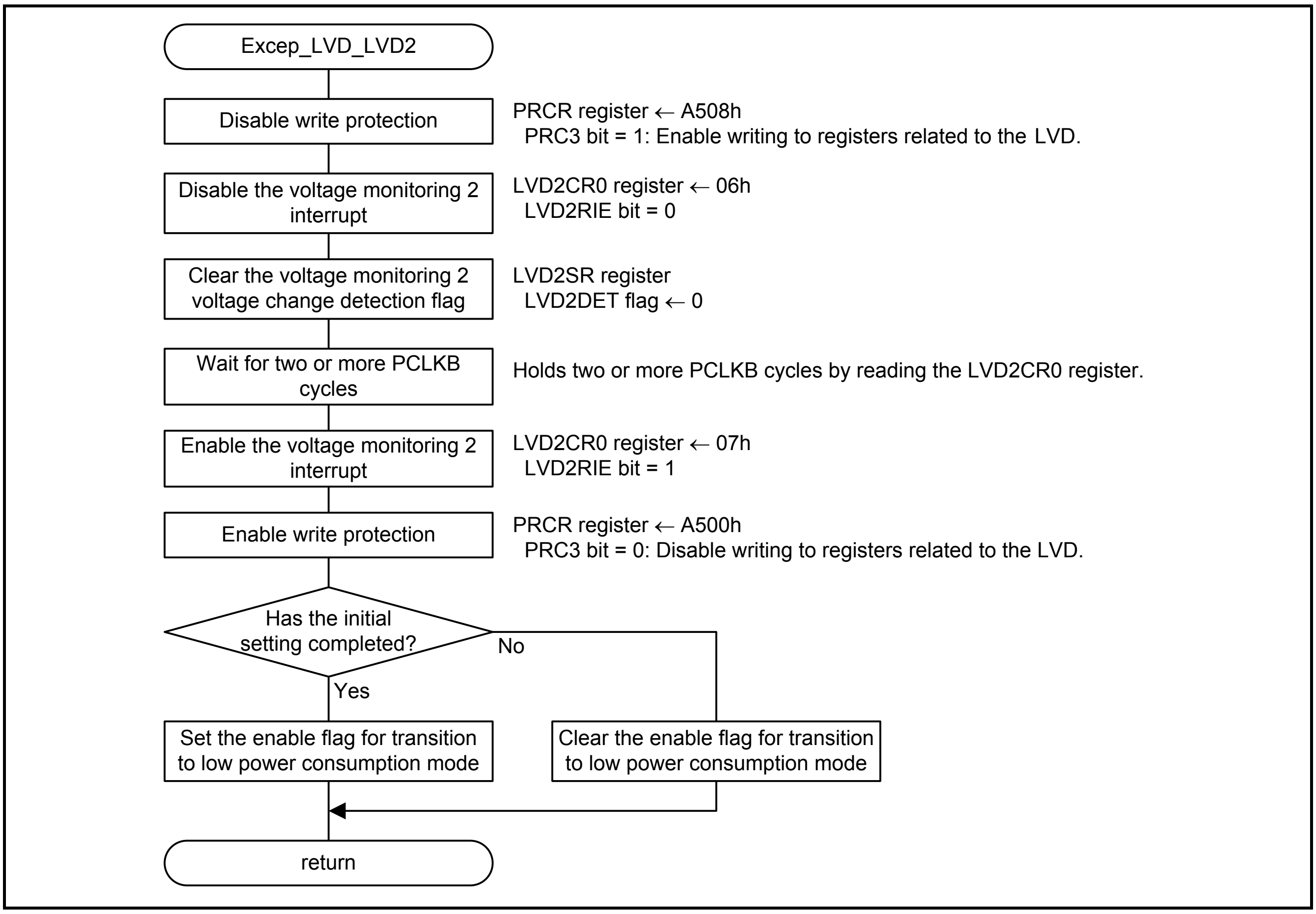


Figure 5.21 LVD2 Interrupt Handling

5.8.13 RTC.PRD Interrupt Handling

Figure 5.22 shows the RTC.PRD Interrupt Handling.

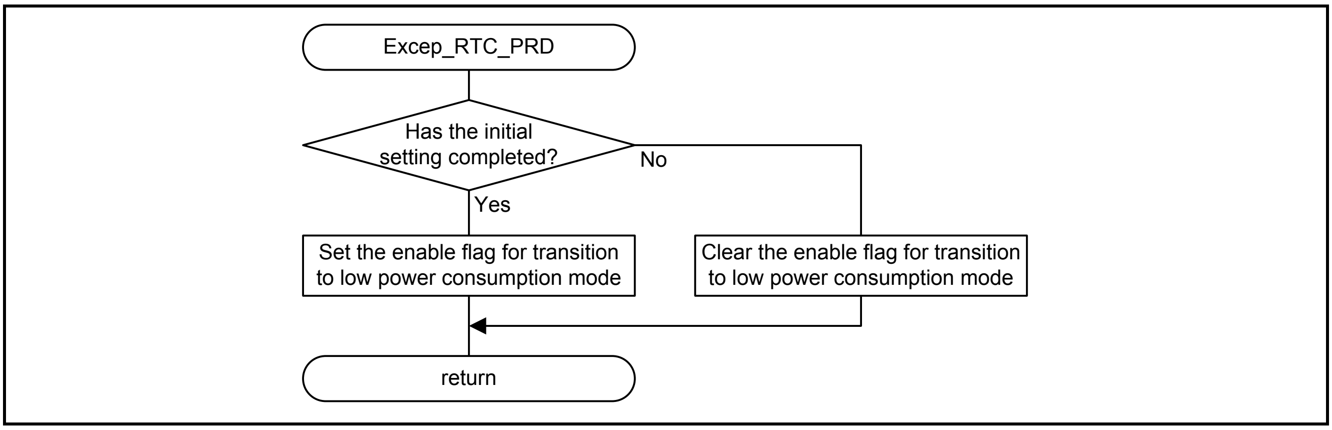


Figure 5.22 RTC.PRD Interrupt Handling

6. Applying This Application Note to the RX21A or RX220 Group

The sample code accompanying this application note has been confirmed to operate with the RX210 Group. To make the sample code operate with the RX21A or RX220 Group, use this application note in conjunction with the Initial Setting application note for each group.

For details on using this application note with the RX21A and RX220 Groups, refer to “5. Applying the RX210 Group Application Note to the RX21A Group” in the RX21A Group Initial Setting application note, and “4. Applying the RX210 Group Application Note to the RX220 Group” in the RX220 Group Initial Setting application note.

Note: • The r_init_clock.h file will be overwritten when applying the Initial Setting for the product used. Make the settings in the overwritten file be same as the original settings in the r_init_clock.h file accompanying this application note.

7. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

8. Reference Documents

User's Manual: Hardware

RX210 Group User's Manual: Hardware Rev.1.50 (R01UH0037EJ)
RX21A Group User's Manual: Hardware Rev.1.00 (R01UH0251EJ)
RX220 Group User's Manual: Hardware Rev.1.10 (R01UH0292EJ)
The latest versions 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	RX210, RX21A, and RX220 Groups Application Note Transition to Low Power Consumption Modes Using the Low Power Consumption Function

Rev.	Date	Description	
		Page	**Summary**
1.00	Aug. 1, 2013	—	First edition issued
1.01	July 1, 2014	1	Products: Added the RX21A and RX220 Groups.
		5	3. Reference Application Notes: Added the Initial Setting application notes for the RX21A and RX220 Groups.
		14	Modified the description of reference application note in the following functions: R_INIT_StopModule, R_INIT_NonExistentPort, and R_INIT_Clock.
		30	6. Applying This Application Note to the RX21A or RX220 Group: Added.
		31	8. Reference Documents: Added the User's Manual: Hardware for the RX21A and RX220 Groups.

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.

RENESAS

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

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

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

Renesas Electronics Canada Limited
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