RX630 Group

Exiting Software Standby Mode Using the RTCa

R01AN1065EJ0100
Rev. 1.00
June 14, 2013

Abstract

This document describes how to obtain the current time information while intermittently exiting software standby mode using the realtime clock (RTC) in the RX630 Group.

Products

RX630 Group, 176-Pin and 177-Pin Packages, ROM Capacities: 768 Kbytes to 2 Mbytes
RX630 Group, 144-Pin and 145-Pin Packages, ROM Capacities: 768 Kbytes to 2 Mbytes
RX630 Group, 100-Pin Package, ROM Capacities: 384 Kbytes to 2 Mbytes
RX630 Group, 80-Pin Package, ROM Capacities: 384 Kbytes and 512 Kbytes

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

This document describes how to obtain the time information while intermittently exiting software standby mode using the RTC.

After a reset, if the reset processing is a cold start, the sub-clock oscillator and RTC are initialized. If the reset processing is a warm start, the RTC data is retained so initialization is not performed.

Next, monitor the input level of the interrupt request pin. If the input level is low, the MCU enters software standby mode.

For the following periodic interrupts that occur every 0.5 seconds, the MCU exits software standby mode, time information is obtained, and the MCU enters software standby mode following the input level of the interrupt request pin.

- RTC count source: Sub-clock
- VBATT pin: Connect to the VCC pin
- Intermittent period: 0.5 seconds

Table 1.1 lists the Peripheral Functions and Their Applications and Figure 1.1 shows the Operation Overview.

In this document, operating states other than software standby mode are referred to as normal mode.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
RTCa	Clock function or exiting from software standby mode
IRQ2	External input for entering software standby mode

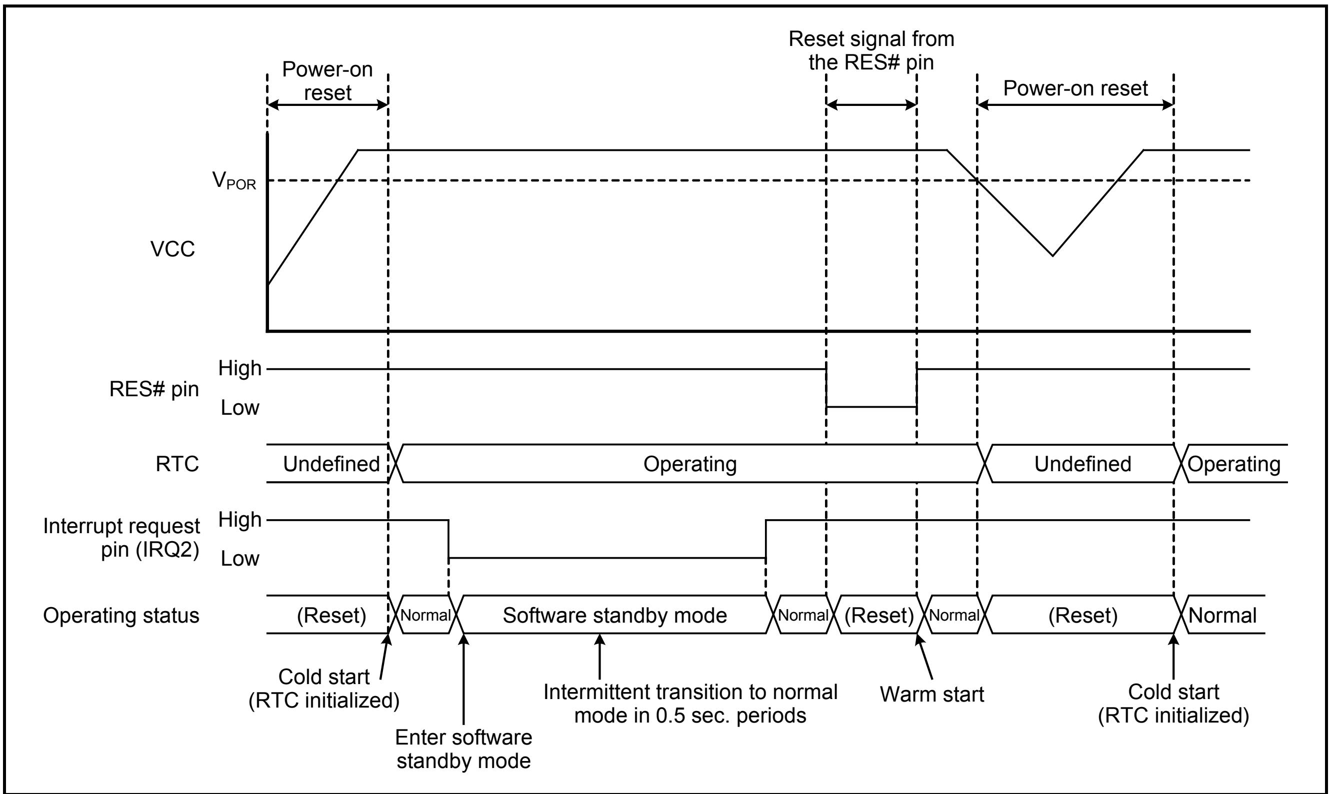


Figure 1.1 Operation Overview

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	R5F5630EDDFP (RX630 Group)
Operating frequencies	- Main clock: 12 MHz - Sub-clock: 32.768 kHz - PLL clock: 192 MHz (main clock divided by 1 and multiplied by 16) - LOCO clock: 125 kHz - System clock (ICLK): 96 MHz (PLL clock divided by 2) - Peripheral module clock B (PCLKB): 48 MHz (PLL clock divided by 4)
Operating voltage	3.3 V
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=rx600 -output=obj="$(CONFIGDIR)\$(FILELEAF).obj" -debug -nologo The integrated development environment default settings are used.
iodefine.h version	Version 1.50
Endian	Little endian
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Sample code version	Version 1.00
Device used	Renesas Starter Kit for RX630 (product part number: R0K505630C000BE)

3. Reference Application Note

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

- RX630 Group Initial Setting Rev. 1.00 (R01AN1004EJ0100)

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 the one 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 Hardware Configuration

Figure 4.1 shows a Connection Example.

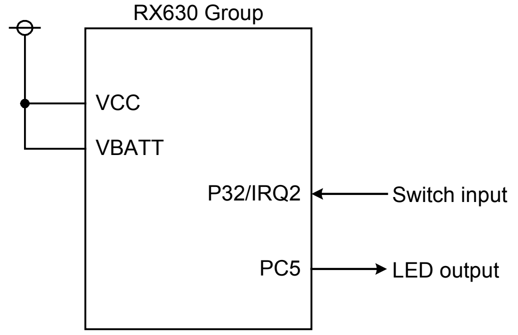


Figure 4.1 Connection Example

4.2 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

This table assumes the 100-pin package is used. When using packages with less than 100 pins, select the pins appropriate to the package used.

Table 4.1 Pins Used and Their Functions

Pin Name	I/O	Function
P32/IRQ2	Input	Switch input for entering modes
PC5	Output	LED output

5. Software

In normal mode, time information is updated intermittently and the level of the interrupt request pin is monitored. The RTC time information is stored in the time data storage area (global variable). The level of the interrupt request pin is monitored using the IR flag of the IRQ2 interrupt.

When the IR flag of the IRQ2 interrupt is 1, an RTC periodic interrupt (PRD interrupt) is enabled, and the MCU enters software standby mode. The MCU exits software standby mode by a PRD interrupt request, and the PRD interrupt is disabled.

After the MCU exits software standby mode, the RTC time information is updated, the level of the interrupt request pin is monitored, and the above processing is repeated.

Settings for the peripheral functions are listed below.

RTC

Count source: Sub-clock

Initial time setting: 00:00:00, Tuesday, January 1, 2013

Time mode: 24-hour mode

RTCOUT output: Disabled

Error adjustment: Not used

Time capture: Not used

Interrupts used: PRD interrupt is generated in 0.5 sec. periods; Carry interrupt (CUP)

IRQ2 input pin

Detection method: Low level is detected

Digital filter: Enabled (sampling clock: PCLKB/8)

Interrupt used: External pin interrupt (IRQ2 interrupt)

5.1 Operation Overview

(1) Initial setting
After a reset, if the RSTSR1.CWSF bit is 0 (cold start), the sub-clock oscillator and RTC are initialized, and the RSTSR1.CWSF bit is set to 1 (warm start). The LED is turned on, and the RTC time information is read. Also, the level of the interrupt request pin is monitored by the IR flag of the IRQ2 interrupt.

(2) Entering software standby mode
When the IR flag of the IRQ2 interrupt becomes 1, a PRD interrupt request is enabled, and the clock source for the system clock changes from the PLL clock to the LOCO clock. The LED is turned off, the WAIT instruction is executed, and the MCU enters software standby mode.

(3) Exiting software standby mode
The MCU exits software standby mode from the PRD interrupt request generated every 0.5 seconds. The LED is turned on in the PRD interrupt handling. The clock source for the system clock switches from the LOCO clock to the PLL clock, and the PRD interrupt request is disabled. After waiting 1/128th of a second, the RTC time information is read. If the IR flag of the IRQ2 interrupt is 1, perform the processing in step (3) again; if the IR flag is 0, read the RTC time information. Also, the level of the interrupt request pin is monitored by the IR flag of the IRQ2 interrupt.

(4) Warm start
After a reset, if the RSTSR1.CWSF bit is 1, the sub-clock oscillator and RTC are not initialized, and MCU operation continues.

Figure 5.1 shows the Timing Diagram.

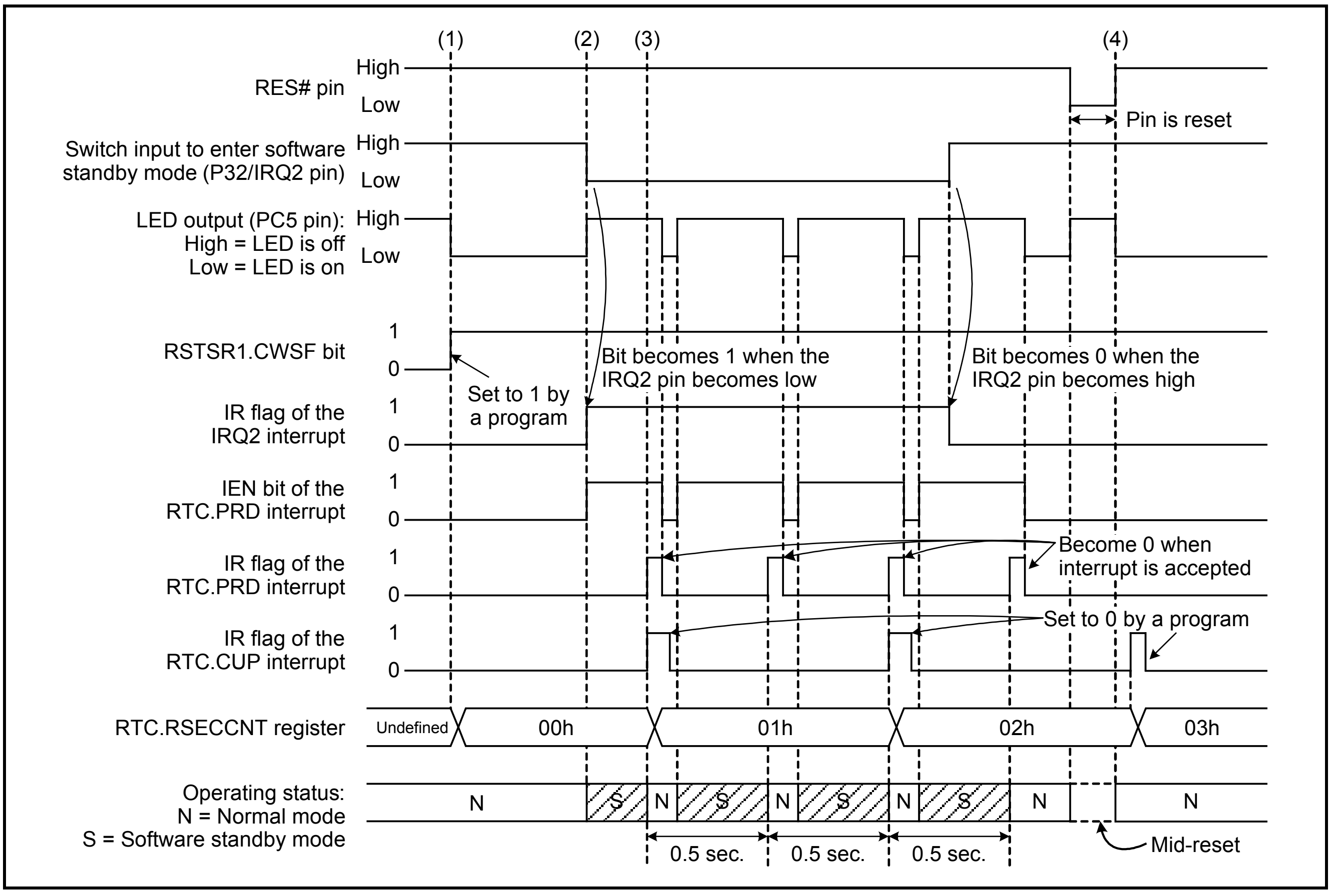


Figure 5.1 Timing Diagram

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
SW_STANDBY	IR(ICU,IRQ2)	IRQ2 interrupt status flag: Switch to enter software standby mode
SW_ON	1	Switch on
SW_OFF	0	Switch off
LED_RUN	PORTC.PODR.BIT.B5	PC5 output data storage bit: LED
LED_ON	0	LED on
LED_OFF	1	LED off
LOOP_COUNT	96,000,000L/128	Loop counter: Wait at least 1/128th of a sec. (ICLK = 96 MHz)

5.5 Structure/Union List

Figure 5.2 shows the Structure/Union Used in the Sample Code.

```
/* **** Time Data **** */
typedef struct
{
    uint8_t    second;        /* Second */
    uint8_t    minute;        /* Minute */
    uint8_t    hour;          /* Hour */
    uint8_t    dayweek;       /* Day of the week */
    uint8_t    day;           /* Day */
    uint8_t    month;         /* Month */
    uint16_t   year;          /* Year */
} time_bcd_t;
```

Figure 5.2 Structure/Union Used in the Sample Code

5.6 Variable

Table 5.4 lists the Global Variable.

Table 5.4 Global Variable

Type	Variable Name	Contents	Function Used
time_bcd_t	time	Time data storage area	rtc_time_read

5.7 Functions

Table 5.5 lists the Functions.

Table 5.5 Functions

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
irq_init	IRQ initialization
rtc_init	RTC initialization
rtc_time_read	RTC time information read
run_to_standby	Preparation to enter software standby mode
standby_to_run	Exit software standby mode
Excep_RTC_PRD	RTC period interrupt handling

5.8 Function Specifications

The following tables list the sample code function specifications.

main

Outline	Main processing
Header	None
Declaration	void main(void)
Description	After the initial setting, the LED is turned on, the RTC time information is read, and if the transition to software standby mode switch is on, the MCU enters software standby mode.
Arguments	None
Return Value	None

port_init

Outline	Port initialization
Header	None
Declaration	void port_init(void)
Description	Initializes the 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	Performs settings to enter the module-stop state.
Arguments	None
Return Value	None
Remark	Transition to the module-stop state is not performed in the sample code. For more information on this function, refer to the RX630 Group Initial Setting Rev. 1.00 application note.

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 176 pins.
Arguments	None
Return Value	None
Remark	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 and 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 more information on this function, refer to the RX630 Group Initial Setting Rev. 1.00 application note.

R_INIT_Clock

Outline	Clock initialization
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Determines if the reset processing is a cold start or a warm start. The sub-clock is only initialized when the reset processing is a cold start.
Arguments	None
Return Value	None
Remark	In the sample code, the PLL clock is selected as the system clock, and the count source of the RTC is selected as the sub-clock for processing (pattern D). For more information on this function, refer to the RX630 Group Initial Setting Rev. 1.00 application note.

peripheral_init

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

rtc_time_read

Outline	RTC time information read
Header	None
Declaration	void rtc_time_read(void)
Description	Reads the RTC time information and stores it in the time data storage area.
Arguments	None
Return Value	None

run_to_standby

Outline	Preparation to enter software standby mode
Header	None
Declaration	void run_to_standby(void)
Description	Performs processing before entering software standby mode.
Arguments	None
Return Value	None

standby_to_run

Outline	Exit software standby mode
Header	None
Declaration	void standby_to_run(void)
Description	Performs processing after exiting software standby mode.
Arguments	None
Return Value	None

Excep_RTC_PRD

Outline	RTC period interrupt handling
Header	None
Declaration	void Excep_RTC_PRD(void)
Description	Turns on the LED.
Arguments	None
Return Value	None

5.9 Flowcharts

5.9.1 Main Processing

Figure 5.3 shows the Main Processing.

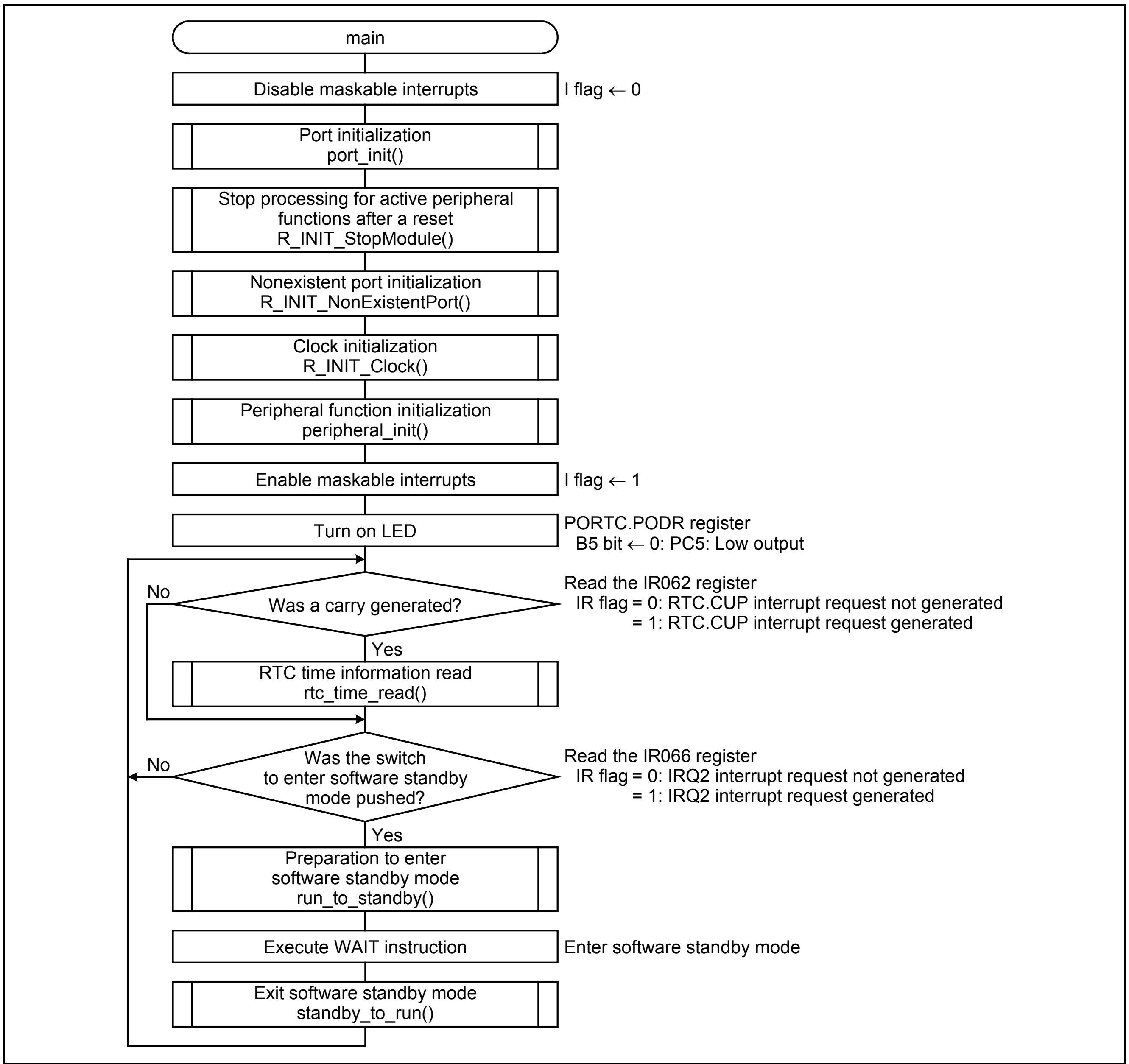


Figure 5.3 Main Processing

5.9.2 Port Initialization

Figure 5.4 shows Port Initialization.

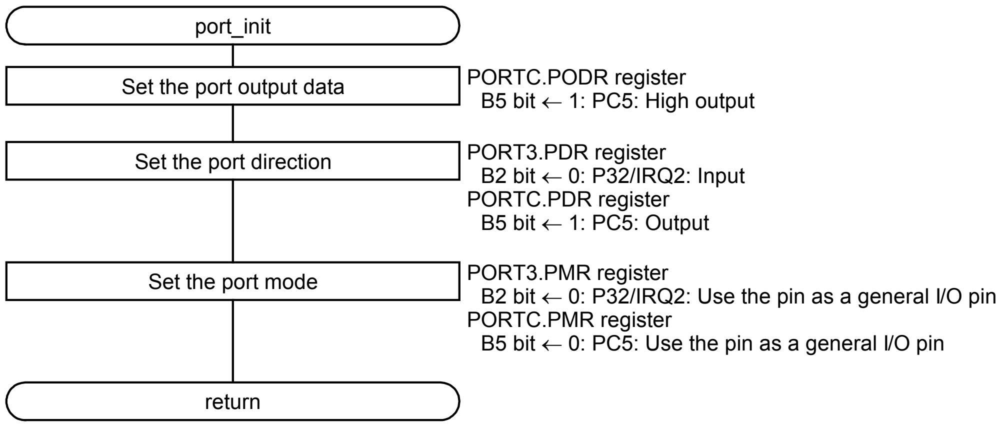
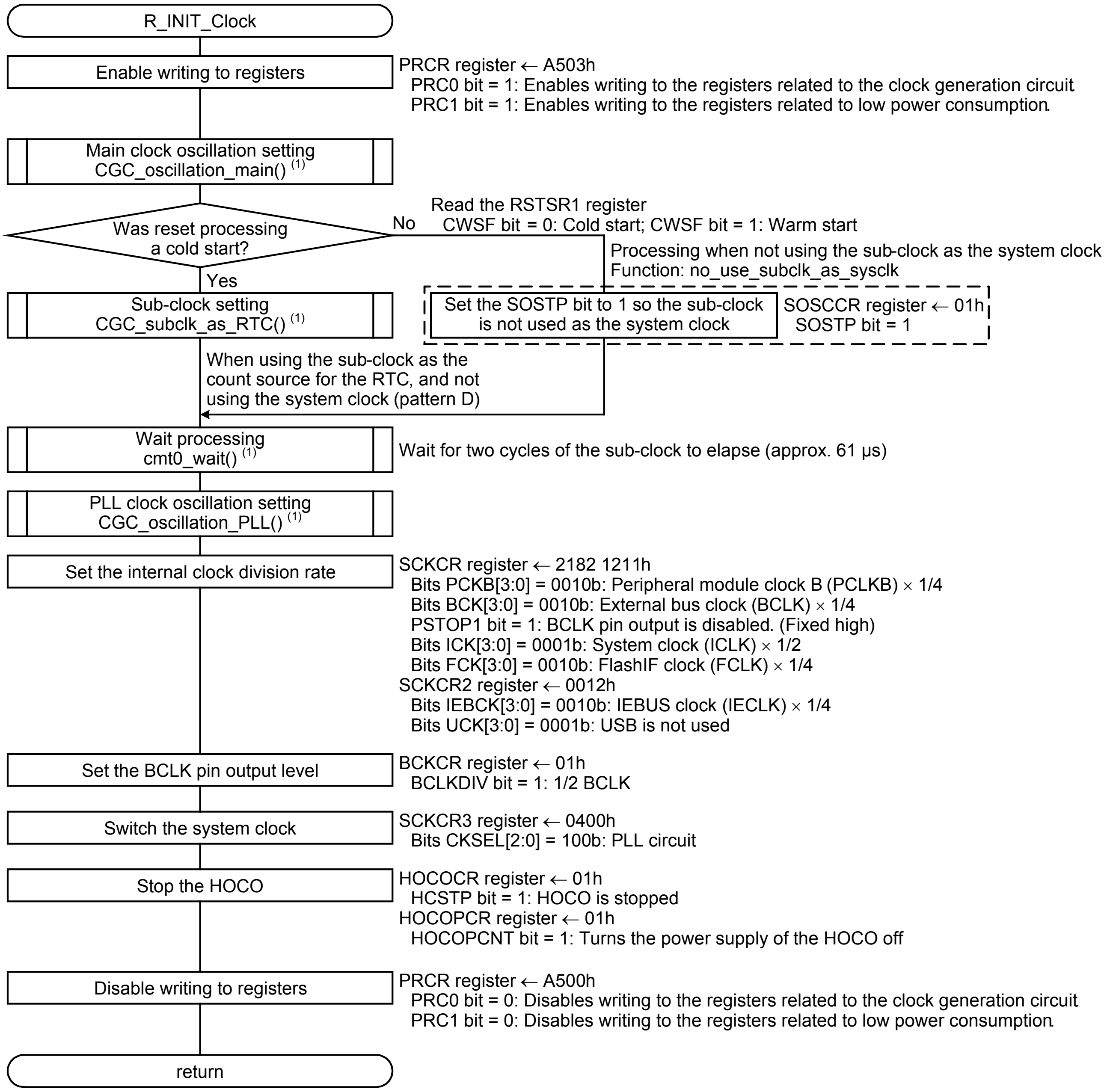


Figure 5.4 Port Initialization

5.9.3 Clock Initialization

Figure 5.5 shows Clock Initialization.

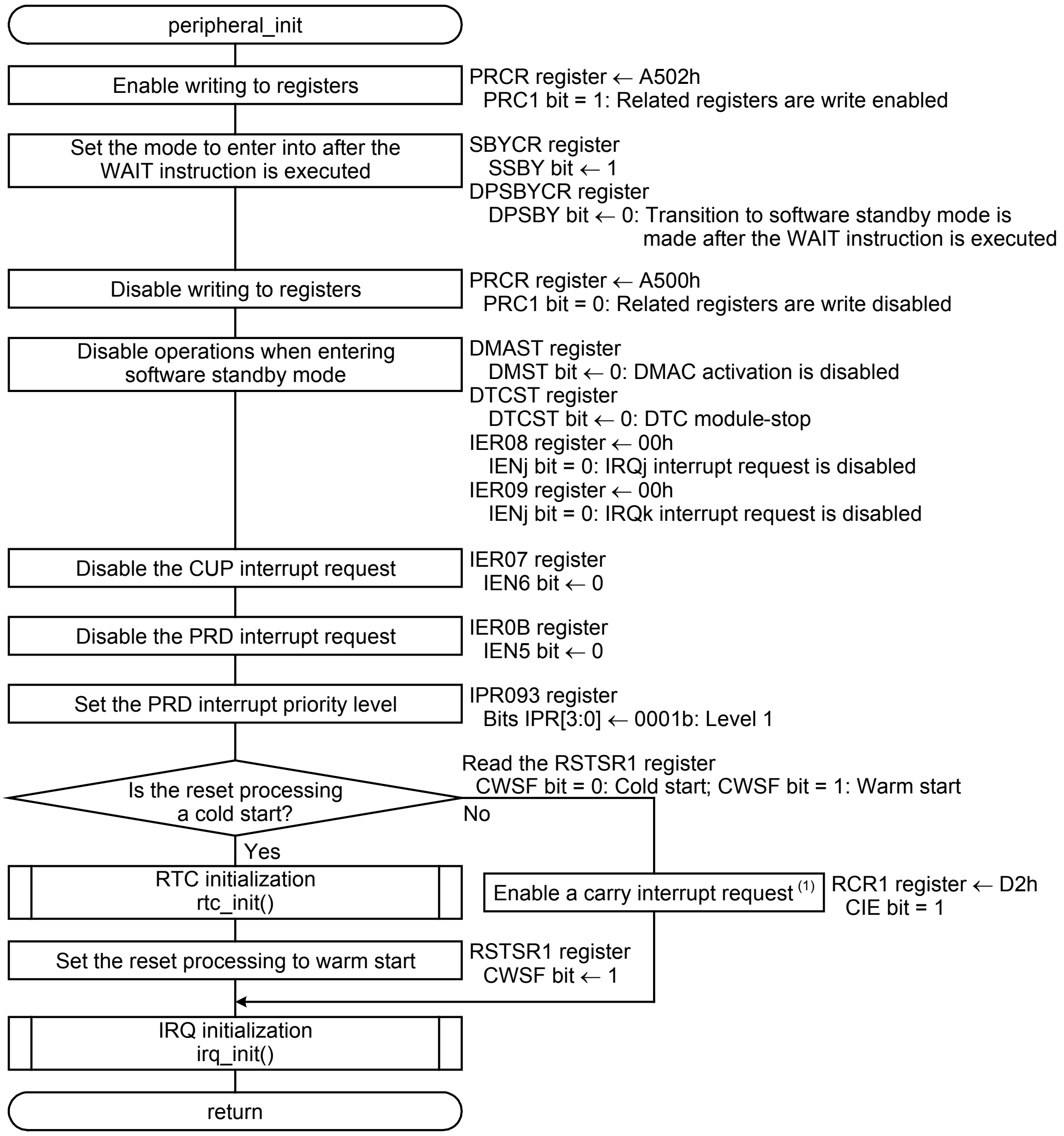


Note:
1. For more information on this function, refer to the RX630 Group Initial Setting Rev. 1.00 application note.

Figure 5.5 Clock Initialization

5.9.4 Peripheral Function Initialization

Figure 5.6 shows Peripheral Function Initialization.



j = 0 to 7; k = 8 to 15

Note:

1. After setting a value to the RCR1 register, read the register to confirm that it can be read.

Figure 5.6 Peripheral Function Initialization

5.9.5 IRQ Initialization

Figure 5.7 shows IRQ Initialization.

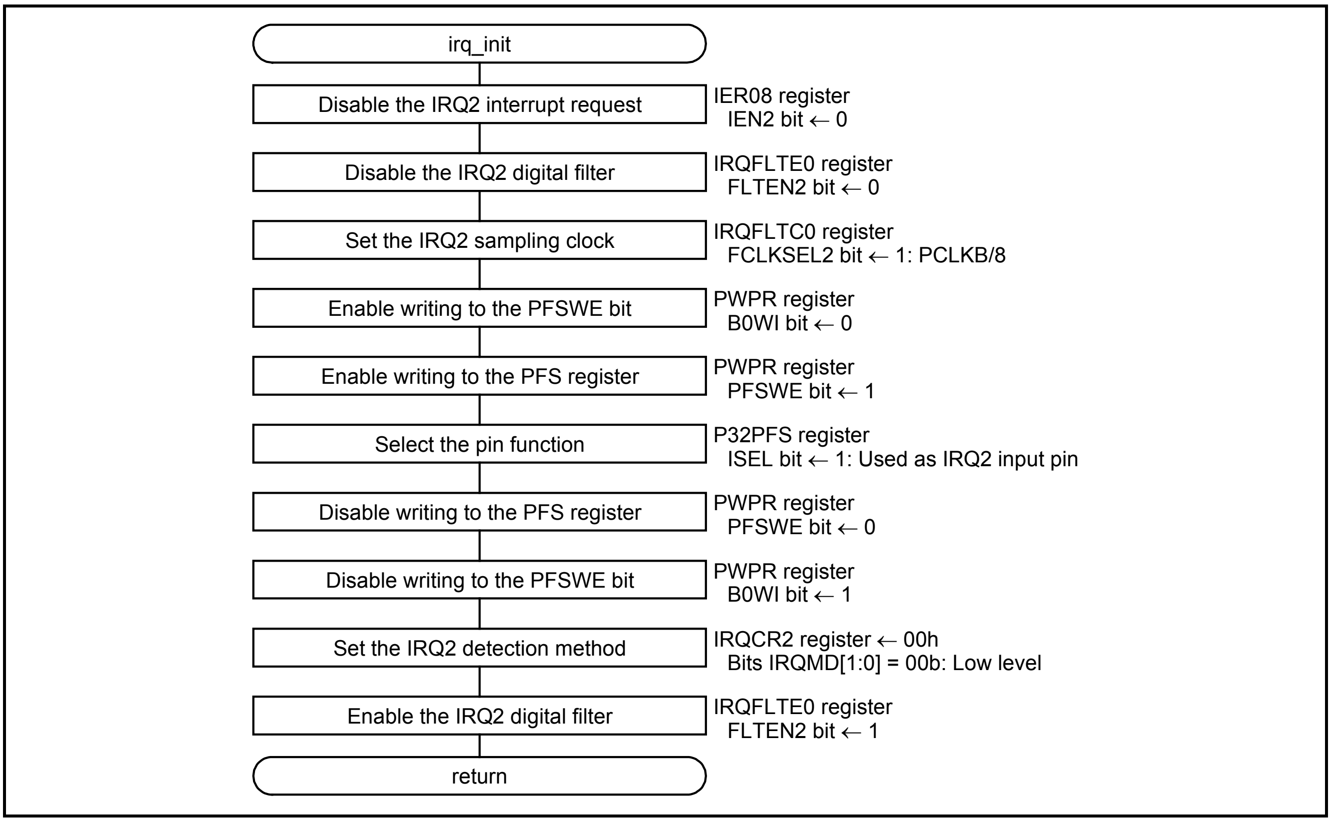
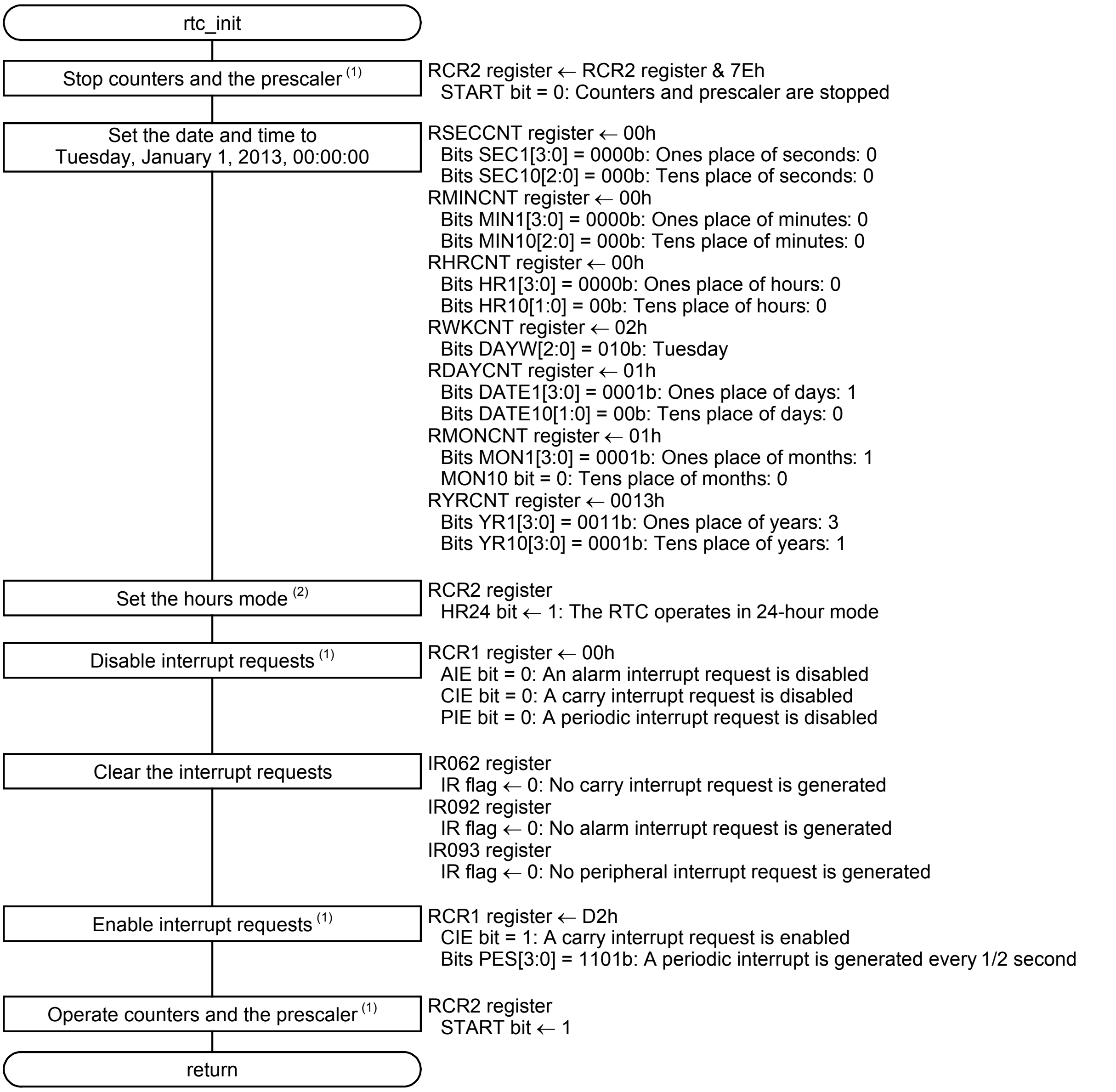


Figure 5.7 IRQ Initialization

5.9.6 RTC Initialization

Figure 5.8 shows RTC Initialization.



Notes:

1. After writing to the RCR1 register and the RCR2.START bit, read the register and the bit to confirm that they can be read.
2. After writing to the RCR2.HR24 bit, perform a dummy read three times.

Figure 5.8 RTC Initialization

5.9.7 RTC Time Information Read

Figure 5.9 shows the RTC Time Information Read.

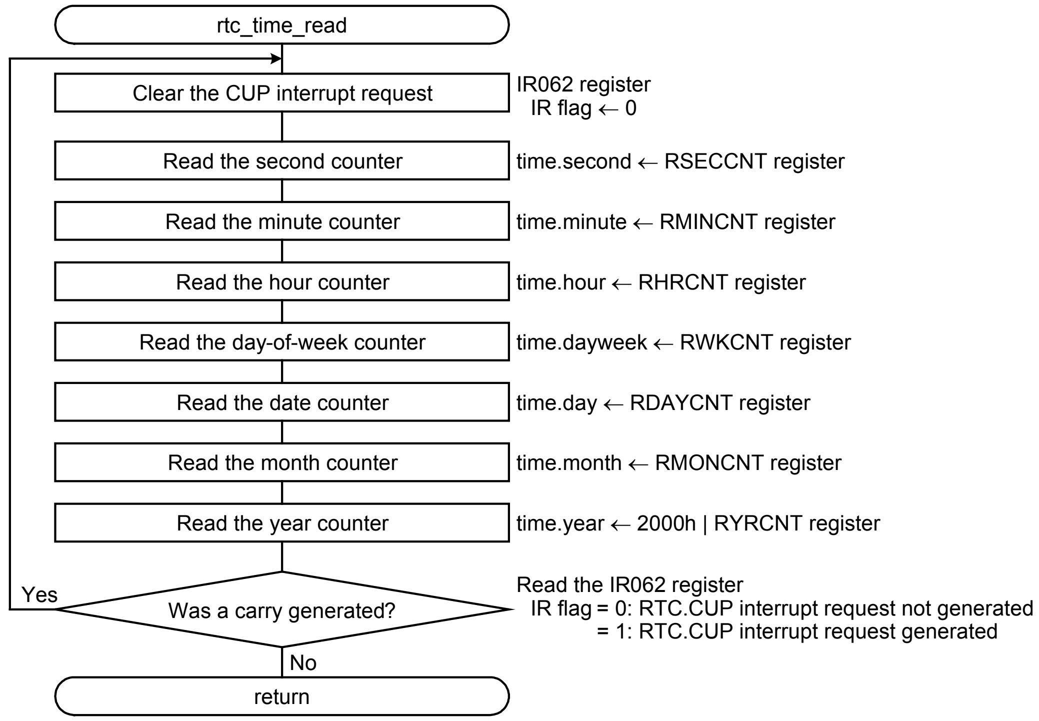
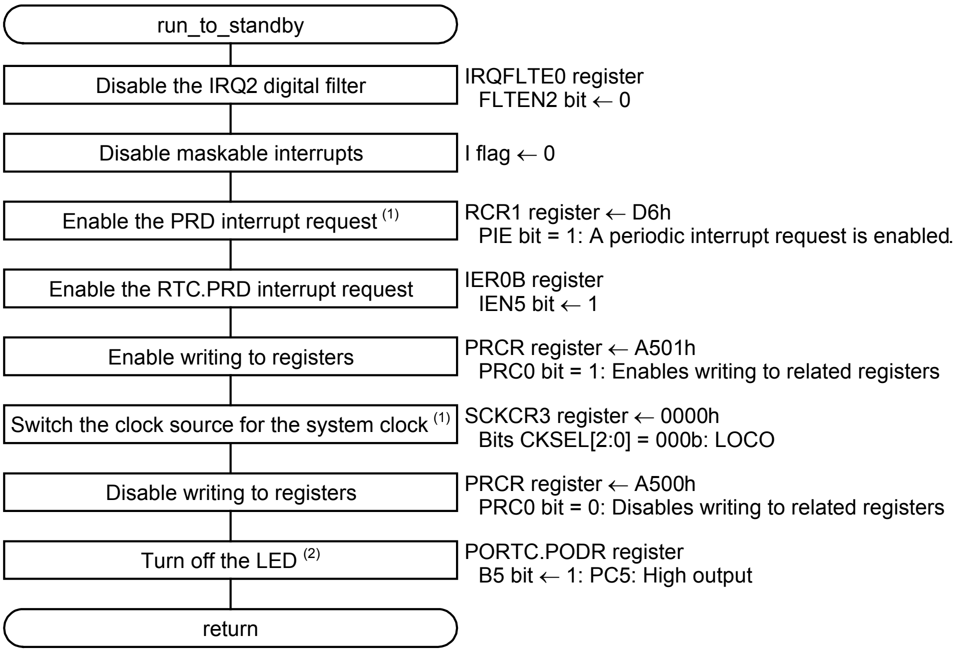


Figure 5.9 RTC Time Information Read

5.9.8 Preparation to Enter Software Standby Mode

Figure 5.10 shows Preparation to Enter Software Standby Mode.

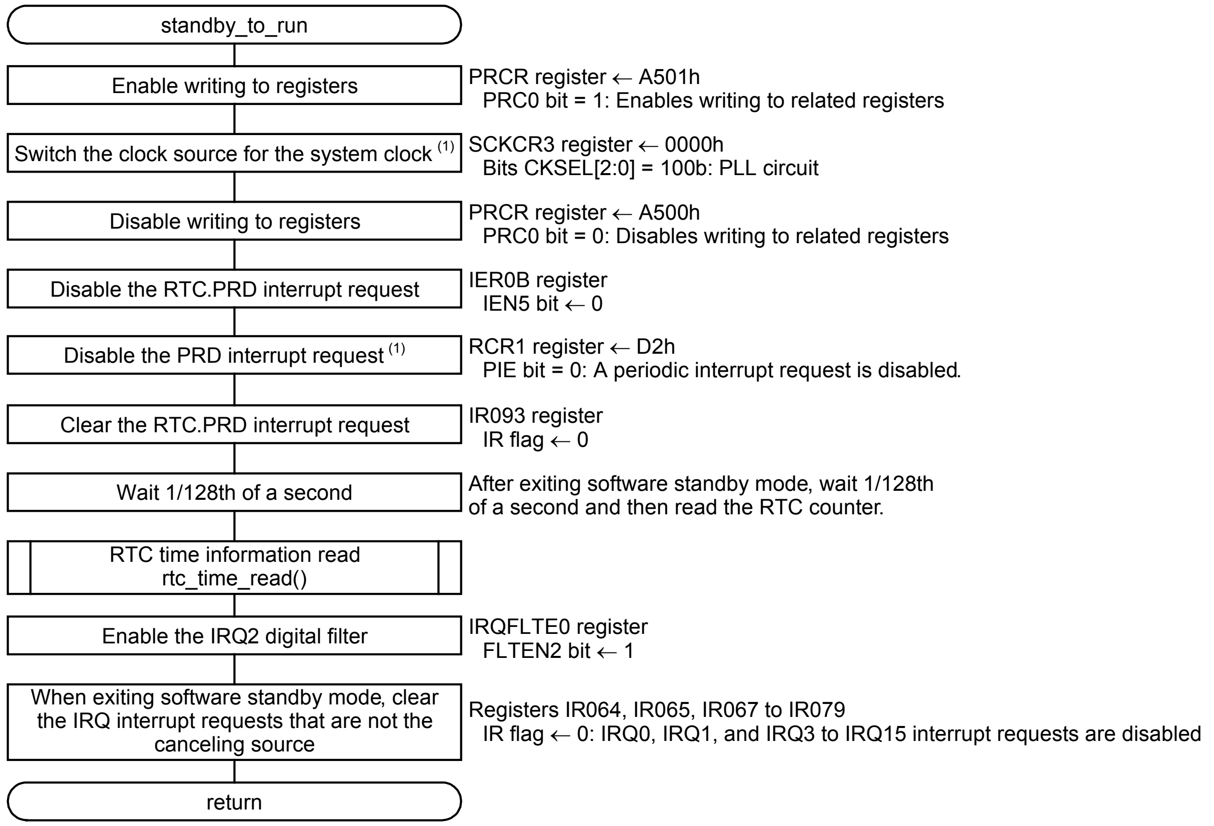


Notes:
1. After writing to the RCR1 register and the SCKCR3 register, read the registers to confirm that they can be read.
2. After setting this bit, read the register to confirm that it can be read.

Figure 5.10 Preparation to Enter Software Standby Mode

5.9.9 Exiting Software Standby Mode

Figure 5.11 shows Exiting Software Standby Mode.



Note:

1. After writing to the RCR1 register and the SCKCR3 register, read the registers to confirm that they can be read.

Figure 5.11 Exiting Software Standby Mode

5.9.10 RTC Period Interrupt Handling

Figure 5.12 shows RTC Period Interrupt Handling.

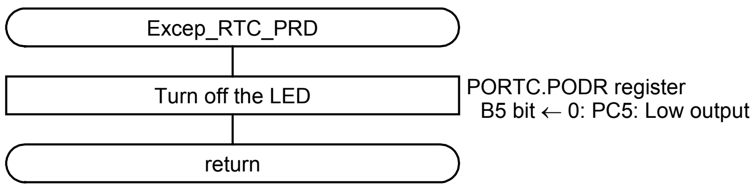


Figure 5.12 RTC Period Interrupt Handling

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

User's Manual: Hardware

RX630 Group User's Manual: Hardware Rev.1.50 (R01UH0040EJ)
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	RX630 Group Application Note Exiting Software Standby Mode Using the RTCa

Rev.	**Date**	**Description**	
		Page	**Summary**
1.00	June 14, 2013	—	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 manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord 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 one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

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.
2880 Scott Boulevard Santa Clara, CA 95050-2554, 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: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

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: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, 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.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.2