RX65N Group, RX651 Group

Initial Settings Example

Introduction

This application note describes the settings that must be made after a reset of a RX65N Group, RX651 Group microcontroller, including clock settings, disabling of peripheral functions still running after a reset, and nonexistent port settings.

Target Devices

- RX65N Group, RX651 Group 177 and 176-pin versions, ROM capacity: 1.5 MB to 2 MB
- RX65N Group, RX651 Group 145 and 144-pin versions, ROM capacity: 512 KB to 2 MB
- RX65N Group, RX651 Group 100-pin versions, ROM capacity: 512 KB to 2 MB
- RX651 Group 64-pin versions, ROM capacity: 512 KB to 2 MB

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 makes settings to disable peripheral functions still running after a reset, nonexistent port settings, and clock settings. The description in this application note applies to the processing that occurs following power-on (cold start).

1.1 Project Description

This application note provides two projects: the project ‘r01an3034_rx65n_1m’ and the project ‘r01an3034_rx65n_2m’.

r01an3034_rx65n_1m is a project for Renesas Starter Kit + for RX65N. The ROM capacity of the device implemented in this RSK is 1 MB. r01an3034_rx65n_1m can also be used for a device of other ROM capacity 512 KB and 768KB.

r01an3034_rx65n_2m is a project for Renesas Starter Kit + for RX65N-2MB. The ROM capacity of the device implemented in this RSK is 2 MB. r01an3034_rx65n_2m can also be used for a device of other ROM capacity 1.5MB. This project corresponds to only a linear mode of the dual bank function. When using a dual mode, please refer to an application notebook "RX Family Firmware Update Sample Program with Dual Bank Function, and Flash Module and SCI Module Firmware Integration Technology" (R01AN3681).

1.2 Disabling Peripheral Functions Still Running After a Reset

Some peripheral functions start operating immediately after power-on, and some have the module stop function disabled. The processing covered under this item disables the following functions:

EXDMAC[*2], DMAC, DTC, standby RAM, RAM2[*1], and RAM0

Note that the above processing is not performed by the sample code. As necessary, overwrite the corresponding constants to execute the processing.

Note 1. RAM2 is function used only with device having RAM capacity of 640KB.

Note 2. EXDMAC is function used only with device having 100-pin and above.

1.3 Nonexistent Port Settings

The pins corresponding to nonexistent ports must be set to the output mode. The sample code of r01an3034_rx65n_1m contains initial setting values suitable for 144 pin products. The sample code of r01an3034_rx65n_2m contains initial setting values suitable for 176 pin products.

Overwrite the constants as necessary to accommodate the actual target device.

1.4 Clock Settings

1.4.1 Overview

The clock setting of r01an3034_rx65n_1m and r01an3034_rx65n_2m is same. The procedure for making clock settings is as follows:

1. Sub-clock settings
2. Main clock settings
3. HOCO clock settings
4. PLL clock settings
5. System clock switching

By making changes to the constants defined in r_init_clock.h, the sample code described in this application note can be used to change the various clock settings.

The sample code sets the PLL clock as the system clock and does not use a sub-clock. Overwrite the constants as necessary to match the clocks you wish to use.

1.4.2 Clock Specifications Assumed in Sample Code

The clock setting sample code of r01an3034_rx65n_1m and r01an3034_rx65n_2m is same. Table 1.1 lists the clock specifications assumed in sample code.

Table 1.1 Clock Specifications Assumed in Sample Code

Clock	Oscillation Frequency	Oscillation Stabilization Time	Remarks
Main clock oscillator	24 MHz	4.2 ms*2	Crystal
Sub-clock oscillator	32.768 kHz*1	1.3 s*2	For standard clock
PLL clock	240 MHz (main clock ×1/1 ×10)	—*3	
HOCO clock	20 MHz*1	—*3	

Note 1. Oscillation disabled by the sample code.

Note 2. The oscillator's stabilization time will differ due to factors such as the wiring pattern and oscillation constant of the system. To obtain the oscillation stabilization time, request an evaluation by the oscillator manufacturer of the system in which the oscillator will be used.

Note 3. See Electrical Characteristics in User's Manual: Hardware.

1.4.3 Clock Selection

By making changes to the constants defined in r_init_clock.h, the sample code described in this application note can be used to select settings such as the clock source of the system clock and whether each clock is oscillating or stopped. The clock selection procedure in r01an3034_rx65n_1m and r01an3034_rx65n_2m is same.

To determine which constants can be changed, see the listing of (user changeable) constants used by the sample code in Table 3.11 (1/3), Table 3.12 (2/3) and Table 3.13 (3/3).

Table 1.2 lists clock selection examples. The sample code sets the PLL clock as the system clock and does not use a sub-clock (No. 1).

Table 1.2 Clock Selection Examples

No.	1	2	3	4	5	6
System clock	PLL	PLL	HOCO	HOCO	Main clock	Main clock
PLL clock	Oscillating	Oscillating	Stopped	Stopped	Stopped	Stopped
Main clock	Oscillating	Oscillating	Stopped	Stopped	Oscillating	Oscillating
HOCO clock	Stopped	Stopped	Oscillating	Oscillating	Stopped	Stopped
Sub-clock	Stopped	Oscillating (using RTC)	Stopped	Oscillating (using RTC)	Stopped	Oscillating (using RTC)
Operating mode	High-speed operating mode	High-speed operating mode	High-speed operating mode	High-speed operating mode	Low-speed operating mode 1	Low-speed operating mode 1
ROM wait cycles*1	2 wait cycles	2 wait cycles	0 wait cycles	0 wait cycles	0 wait cycles	0 wait cycles
Constants						
SEL_SYSCLK	CLK_PLL	CLK_PLL	CLK_HOCO	CLK_HOCO	CLK_MAIN	CLK_MAIN
SEL_PLL	B_USE_PLL_MAIN	B_USE_PLL_MAIN	B_NOT_USE	B_NOT_USE	B_NOT_USE	B_NOT_USE
SEL_MAIN	B_USE	B_USE	B_NOT_USE	B_NOT_USE	B_USE	B_USE
SEL_HOCO	B_NOT_USE	B_NOT_USE	B_USE	B_USE	B_NOT_USE	B_NOT_USE
SEL_SUB*2	B_NOT_USE	B_NOT_USE	B_NOT_USE	B_NOT_USE	B_NOT_USE	B_NOT_USE
SEL_RTC*2	B_NOT_USE	B_USE	B_NOT_USE	B_USE	B_NOT_USE	B_USE
REG_OPCCR	OPCM_HIGH	OPCM_HIGH	OPCM_HIGH	OPCM_HIGH	OPCM_LOW_1	OPCM_LOW_1
REG_ROMWT	ROMWT_2WAIT	ROMWT_2WAIT	ROMWT_0WAIT	ROMWT_0WAIT	ROMWT_0WAIT	ROMWT_0WAIT

Note 1. Set the REG_ROMWT to ROMWT_2WAIT(2 wait) if the ICLK frequency is 100 MHz or higher. Do not set the REG_ROMWT to ROMWT_0WAIT(0 wait) if the ICLK frequency is 50 MHz or higher.

Note 2. Set SEL_SUB to B_USE (use) when the sub-clock is used as the system clock, and set SEL_RTC to B_USE when the sub-clock is used as the RTC count source. The sub-clock oscillates when either SEL_SUB or SEL_RTC, or both of them, are set to B_USE.

2. Operation Confirmation Conditions

The operation of the sample code referenced in this application note (No. 1 to 6 in Table 1.2) has been confirmed under the following conditions.

Table 2.1 lists the operation check conditions of r01an3034_rx65n_1m, and Table2.2 lists the operation check conditions of r01an3034_rx65n_2m.

Table 2.1 Operation Confirmation Conditions in r01an3034_rx65n_1m

Item		Contents
MCU used		R5F565N9ADFB (RX65N Group)
Operating frequency	PLL clock selected as system clock (No. 1 and 2 in Table 1.2)	Main clock: 24 MHz PLL: 240 MHz (main clock ×1/1 ×10) System clock (ICLK): 120 MHz (PLL ×1/2) Peripheral module clock A (PCLKA): 120 MHz (PLL ×1/2) Peripheral module clocks B to D (PCLKB to PCLKD): 60 MHz (PLL ×1/4) Flash interface clock (FCLK): 60 MHz (PLL ×1/4) External bus clock (BCLK): 60 MHz (PLL ×1/4)
	HOCO clock selected as system clock (No. 3 and 4 in Table 1.2)	HOCO: 20 MHz System clock (ICLK): 20 MHz (HOCO ×1/1) Peripheral module clock A (PCLKA): 20 MHz (HOCO ×1/1) Peripheral module clock B to D (PCLKB to PCLKD): 10 MHz (HOCO ×1/2) Flash interface clock (FCLK): 10 MHz (HOCO ×1/2) External bus clock (BCLK): 10 MHz (HOCO ×1/2)
	Main clock selected as system clock (No. 5 and 6 in Table 1.2)	Main clock: 24 MHz System clock (ICLK): 750 kHz (main clock ×1/32) Peripheral module clock A (PCLKA): 750 kHz (main clock ×1/32) Peripheral module clock B to D (PCLKB to PCLKD): 750 kHz (main clock ×1/32) Flash interface clock (FCLK): 750 kHz (main clock ×1/32) External bus clock (BCLK): 750 kHz (main clock ×1/32)
Operating voltage		3.3 V
Integrated development environment		Renesas Electronics e^2 studio Version: 2021-01
C compiler		Renesas Electronics C/C++ Compiler Package for RX Family V3.02
		Compiler option The integrated development environment default settings are used.
iodefine.h version		V 2.30
Endian		Little endian or big endian
Operating mode		Single-chip mode
Processor mode		Supervisor mode
Sample code version		Version 1.21
Board used		Renesas Starter Kit+ for RX65N (Product No. RTK500565NSxxxxxBE)

Table 2.2 Operation Confirmation Conditions in r01an3034_rx65n_2m

Item		Contents
MCU used		R5F565NEDDFC (RX65N Group)
Operating frequency	PLL clock selected as system clock (No. 1 and 2 in Table 1.2)	Main clock: 24 MHz PLL: 240 MHz (main clock ×1/1 ×10) System clock (ICLK): 120 MHz (PLL ×1/2) Peripheral module clock A (PCLKA): 120 MHz (PLL ×1/2) Peripheral module clocks B to D (PCLKB to PCLKD): 60 MHz (PLL ×1/4) Flash interface clock (FCLK): 60 MHz (PLL ×1/4) External bus clock (BCLK): 60 MHz (PLL ×1/4)
	HOCO clock selected as system clock (No. 3 and 4 in Table 1.2)	HOCO: 20 MHz System clock (ICLK): 20 MHz (HOCO ×1/1) Peripheral module clock A (PCLKA): 20 MHz (HOCO ×1/1) Peripheral module clock B to D (PCLKB to PCLKD): 10 MHz (HOCO ×1/2) Flash interface clock (FCLK): 10 MHz (HOCO ×1/2) External bus clock (BCLK): 10 MHz (HOCO ×1/2)
	Main clock selected as system clock (No. 5 and 6 in Table 1.2)	Main clock: 24 MHz System clock (ICLK): 750 kHz (main clock ×1/32) Peripheral module clock A (PCLKA): 750 kHz (main clock ×1/32) Peripheral module clock B to D (PCLKB to PCLKD): 750 kHz (main clock ×1/32) Flash interface clock (FCLK): 750 kHz (main clock ×1/32) External bus clock (BCLK): 750 kHz (main clock ×1/32)
Operating voltage		3.3 V
Integrated development environment		Renesas Electronics e^2 studio Version: 2021-01
C compiler		Renesas Electronics C/C++ Compiler Package for RX Family V3.02
		Compiler option The integrated development environment default settings are used.
iodefine.h version		V 2.2
Endian		Little endian or big endian
Operating mode		Single-chip mode
Processor mode		Supervisor mode
Sample code version		Version 1.11
Board used		Renesas Starter Kit+ for RX65N-2 MB (Product No. RTK50565N2SxxxxxBE)

3. Software

After disabling peripheral functions still running after a reset and making nonexistent port settings, the sample code makes clock settings.

3.1 Disabling Peripheral Functions Still Running After a Reset

The sample code disables peripheral functions still running after a reset.

Only the peripheral modules listed below are not in the module stop state after a reset is canceled. To transition a module to the module stop state, set the corresponding module stop bit to 1 (transition to module stop state). Putting modules into the module stop state can reduce the power consumption of the device.

In the sample code the value of the constant MSTP_STATE_<target module name> is 0 (MODULE_STOP_DISABLE), so the target module does not transition to the module stop state. To transition one or more modules to the module stop state on the target system, set the corresponding constant(s) to 1 (MODULE_STOP_ENABLE) in r_init_stop_module.h.

Table 3.1 lists the peripheral modules that are not in the module stop state after a reset.

Table 3.1 Peripheral Modules Not in Module Stop State After a Reset

Peripheral Module	Module Stop Setting Bit	Value After Reset	Setting When Not Using Module
EXDMAC*2	MSTPCRA.MSTPA29 bit	0 (module stop state canceled)	1 (transition to module stop state)
DMAC/DTC	MSTPCRA.MSTPA28 bit		
Standby RAM	MSTPCRC.MSTPC7 bit		
RAM2*1	MSTPCRC.MSTPC2 bit		
RAM0	MSTPCRC.MSTPC0 bit		

Note 1. RAM2 is function used only with device having RAM capacity of 640KB.

Note 2. EXDMAC is function used only with device having 100-pin and above.

3.2 Nonexistent Port Settings

3.2.1 Processing Overview

The bits in the PDR registers corresponding to nonexistent ports are set to 1 (output). When writing in byte units to PDR or PODR registers containing nonexistent ports after this function has been called, set the direction control bits corresponding to the nonexistent ports to 1 and the port output data storage bits corresponding to the nonexistent ports to 0.

Table 3.2, Table 3.3, Table 3.4 and Table 3.5 list the nonexistent ports setting corresponding to 177/176 pin, 144/145 pin, 100 pin and 64 pin device.

Table 3.2 Nonexistent Ports (177- and 176-pin)[*1]

Port Symbol	177- and 176-Pin Products	Pins
PORT0	P04, P06	2
PORT1	—	—
PORT2	—	—
PORT3	—	—
PORT4	—	—
PORT5	—	—
PORT6	—	—
PORT7	—	—
PORT8	—	—
PORT9	—	—
PORTA	—	—
PORTB	—	—
PORTC	—	—
PORTD	—	—
PORTE	—	—
PORTF	PF6, PF7	2
PORTG	—	—
PORTJ	PJ4, PJ6, PJ7	3

Note 1: Pins 177 and 176 correspond only to r01an3034_rx65n_2m.

Table 3.3 Nonexistent Ports (145- and 144-pin)

Port Symbol	145- and 144-Pin Products	Pins
PORT0	P04, P06	2
PORT1	P10, P11	2
PORT2	—	—
PORT3	—	—
PORT4	—	—
PORT5	P57	1
PORT6	—	—
PORT7	—	—
PORT8	P84, P85	2
PORT9	P94 to P97	4
PORTA	—	—
PORTB	—	—
PORTC	—	—
PORTD	—	—
PORTE	—	—
PORTF	PF0 to PF4, PF6, PF7	7
PORTG[*1]	PG0 to PG7	8
PORTJ	PJ0 to PJ2, PJ4, PJ6, PJ7	6

Note 1: PORTG correspond only to r01an3034_rx65n_2m.

Table 3.4 Nonexistent Ports (100-pin)

Port Symbol	100-Pin Products	Pins
PORT0	P00 to P04, P06	6
PORT1	P10, P11	2
PORT2	—	—
PORT3	—	—
PORT4	—	—
PORT5	P56, P57	2
PORT6	P60 to P67	8
PORT7	P70 to P77	8
PORT8	P80 to P87	8
PORT9	P90 to P97	8
PORTA	—	—
PORTB	—	—
PORTC	—	—
PORTD	—	—
PORTE	—	—
PORTF	PF0 to PF7	8
PORTG[*1]	PG0 to PG7	8
PORTJ	PJ0 to PJ2, PJ4 to PJ7	7

Note 1: PORTG correspond only to r01an3034_rx65n_2m.

Table 3.5 Nonexistent Ports (64-pin)

Port Symbol	64-Pin Products	Pins
PORT0[*1]	P00 to P04, P06, P07	7
PORT1	P10, P11, P14, P15	4
PORT2	P20 to P25	6
PORT3	P32, P33	2
PORT4	P44 to P47	4
PORT5	P50 to P52, P54 to P57	7
PORT6	P60 to P67	8
PORT7	P70 to P77	8
PORT8	P80 to P87	8
PORT9	P90 to P97	8
PORTA	PA0, PA3, PA5	3
PORTB	PB0 to PB4	5
PORTC	PC2, PC3	2
PORTD	PD0, PD1	2
PORTE	PE3 to PE5	3
PORTF	PF0 to PF7	8
PORTG[*2]	PG0 to PG7	8
PORTJ	PJ0 to PJ7	8

Note 1: P05 is not provided on TFBGA 64-pin.

Note 2: PORTG correspond only to r01an3034_rx65n_2m.

3.2.2 Pin Count Setting

The setting in the sample code of project r01an3034_rx65n_1m (PIN_SIZE=144) is for 144-pin products. The other pin counts supported by this project are 145, 100 and 64. If the pin count of the target device is other than 144, change the value of PIN_SIZE in r_init_port_initialize.h to match the target device.

The setting in the sample code of project r01an3034_rx65n_2m (PIN_SIZE=176) is for 176-pin products. The other pin counts supported by this project are 177, 145, 144, 100 and 64. If the pin count of the target device is other than 176, change the value of PIN_SIZE in r_init_port_initialize.h to match the target device.

3.3 Clock Settings

3.3.1 Clock Setting Procedure

Table 3.6 lists the steps in the clock setting procedure, the processing performed in each step, and the default settings of the sample code. Using the default settings, the sample code sets the PLL clock as the main clock and turns off the HOCO and sub-clock. The clock selection procedure in r01an3034_rx65n_1m and r01an3034_rx65n_2m is same.

Table 3.6 Clock Setting Procedure

Step	Processing	Details of Processing		Sample Code Settings
1	Sub-clock setting*2	Not used	Initializes the sub-clock control circuit.	The sub-clock is not used.
		Used	Initializes the sub-clock control circuit, sets the drive capacity, and sets in SOSCWTCR the waiting time until output of the sub-clock to the internal clock starts; then starts oscillation by the sub-clock. After this, waits for the clock oscillation stabilization waiting time*1 using hardware.	
2	Main clock setting*2	Not used	This setting is unnecessary.	The main clock is used.
		Used	Sets the main clock drive capacity and sets in MOSCWTCR the waiting time until output of the main clock to the internal clocks starts, then starts oscillation by the main clock. After this, waits for the clock oscillation stabilization waiting time*1 using hardware.	
3	HOCO clock setting*2	Not used	Turns off the HOCO power supply.	The HOCO is not used.
		Used	Sets the HOCO frequency, then starts oscillation by the HOCO clock. After this, waits for the clock oscillation stabilization waiting time*1 using hardware.	
4	PLL clock setting*2	Not used	Turns off the PLL power supply.	The PLL clock is used.
		Used	Sets the PLL input division ratio and frequency multiplication factor, then starts oscillation by the PLL clock. After this, waits for the clock oscillation stabilization waiting time*1 using hardware.	
5	Operating power control mode setting	Sets the operating power control mode according to the operating frequency and operating voltage used.		High-speed operating mode is selected.
6	Clock division ratio settings	Changes the clock division ratios.		• ICLK and PCLKA: ×1/2 • PCLKB to PCLKD, BCLK, and FCLK: ×1/4 • BCLK: Output stopped
7	System clock switching	Switches according to the system used.		Switches to PLL clock.

Note 1. Confirms that the appropriate bit in the oscillation stabilization flag register (OSCOVFSR) is set to 1.

Note 2. Change the values of the constants in r_init_clock.h as necessary to match the selection of the clocks you wish to use or not use.

3.4 Section Composition

Table 3.7 lists the section data changed in the sample code (r01an3034_rx65n_1m). Table 3.8 lists the section data changed in the sample code (r01an3034_rx65n_2m). For details on adding, changing, and deleting section, refer to the RX Family C/C++ Compiler Package User's Manual.

Table 3.7 Section Data Changed in the Sample Code (r01an3034_rx65n_1m)

Section Name	**Change**	**Address**	**Function**
End_of_RAM0	Addition	0003 FFFCh	Last address of On-chip RAM

Table 3.8 Section Data Changed in the Sample Code (r01an3034_rx65n_2m)

Section Name	**Change**	**Address**	**Function**
End_of_RAM0	Addition	0003 FFFCh	Last address of On-chip RAM
End_of_RAM2	Addition	0085 FFFCh	Last address of On-chip expansion RAM

3.5 File Composition

Table 3.9 lists the files used in the sample code. Files generated by the integrated development environment are not included in this table. The file structure is the same for r01an3034_rx65n_1m and r01an3034_rx65n_2m.

Table 3.9 Files Used in the Sample Code

File Name	**Outline**	**Remarks**
main.c	Main processing routine	
r_init_stop_module.c	Disable peripheral functions still running after a reset	
r_init_stop_module.h	Header file of r_init_stop_module.c	
r_init_port_initialize.c	Initial nonexistent port settings	
r_init_port_initialize.h	Header file of r_init_port_initialize.c	
r_init_clock.c	Initial clock settings	
r_init_clock.h	Header file of r_init_clock.c	

3.6 Option-Setting Memory

Table 3.10 lists the option-setting memory configured in the sample code. When necessary, set a value suited to the user system. The initial setting of OFS0, OFS1, and MDE is the same for r01an3034_rx65n_1m and r01an3034_rx65n_2m.

Table 3.10 Option-Setting Memory Configured in the Sample Code

Symbol	**Address**	**Setting Value**	**Contents**
OFS0	FE7F 5D04h to FE7F 5D07h	FFFF FFFFh	IWDT stopped after a reset WDT stopped after a reset
OFS1	FE7F 5D08h to FE7F 5D0Bh	FFFF FFFFh	Voltage monitor 0 reset disabled after a reset HOCO oscillation disabled after a reset
MDE	FE7F 5D00h to FE7F 5D03h	FFFF FFFFh	Little endian

3.7 Constants

Table 3.11 shows constants (user changeable) used by sample code (1/3). Table 3.12 shows constants (user changeable) used by sample code (2/3). Table 3.13 shows constants (user changeable) used by sample code (3/3). Table 3.14 shows constants (non user-changeable) used by sample code.

Table 3.15 shows constants for 177- and 176- pin products (PIN_SIZE = 177 or 176). Table 3.16 shows constants for 145- and 144- pin products (PIN_SIZE 145 or 144). Table 3.17 shows constants for 100-pin products (PIN_SIZE 100). Table 3.18 shows constants for 64-pin products (PIN_SIZE 64).

The constants of r01an3034_rx65n_1m and r01an3034_rx65n_2m are the same except for MSTP_STATE_RAM2 and DEF_PGPDR that are defined only for r01an3034_rx65n_2m.

Table 3.11 Constants (User Changeable) Used by Sample Code (1/3)

Constant Name	Setting Value	Contents
SEL_MAIN*1	B_USE	Main clock enable/disable selection B_USE: Used (main clock enabled) B_NOT_USE: Not used (main clock disabled)
MAIN_CLOCK_Hz*1	24,000,000 L	Main clock oscillator frequency (Hz)
REG_MOFCR*1	00h	Main clock oscillator drive capacity setting (setting value of MOFCR register)
REG_MOSCWTCR*1	53h	Setting value of main clock wait control register
SEL_SUB*1*2	B_NOT_USE	Sub-clock usage selection (used as system clock) B_USE: Used B_NOT_USE: Not used
SEL_RTC*1*2	B_NOT_USE	Sub-clock usage selection (used as RTC count source) B_USE: Used B_NOT_USE: Not used
SUB_CLOCK_Hz*1	32,768 L	Sub-clock oscillator frequency (Hz)
REG_SOSCWTCR*1	21h	Setting value of sub-clock wait control register
REG_RCR3*1	CL_STD	Sub-clock oscillator drive capacity selection CL_STD: Drive capacity for standard clock CL_LOW: Drive capacity for low clock
SEL_PLL*1	B_USE_PLL_MAIN	PLL clock enable/disable selection B_USE_PLL_MAIN: Used (Main clock) B_USE_PLL_HOCO: Used (HOCO) B_NOT_USE: Not used (PLL clock disabled)
REG_PLLCR*1	1300h	PLL input division ratio and frequency multiplication factor settings (setting value of PLLCR register)

Note 1. Change the settings values in r_init_clock.h to match the target system.

Note 2. The sub-clock oscillates when either SEL_SUB or SEL_RTC, or both of them, are set to B_USE (use).

Table 3.12 Constants (User Changeable) Used by Sample Code (2/3)

Constant Name	Setting Value	Contents
SEL_HOCO*1	B_NOT_USE	HOCO clock enable/disable selection B_USE: Used (HOCO clock enabled) B_NOT_USE: Not used (HOCO clock disabled)
REG_HOCOCR2*1	FREQ_20MHz	HOCO clock frequency selection FREQ_16MHz: 16 MHz FREQ_18MHz: 18 MHz FREQ_20MHz: 20 MHz
SEL_SYSCLK*1	CLK_PLL	System clock clock source selection CLK_PLL: PLL CLK_ HOCO: HOCO CLK_ MAIN: main clock CLK_SUB: sub-clock
REG_OPCCR*1	OPCM_HIGH	Operating power control mode selection*4 OPCM_HIGH: High-speed operating mode OPCM_LOW_1: Low-speed operating mode 1*2 OPCM_LOW_2: Low-speed operating mode 2*3

Note 1. Change the settings values in r_init_clock.h to match the target system.
Note 2. It is not possible to select low-speed operating mode 1 when the PLL clock is set to oscillate.
Note 3. It is not possible to select low-speed operating mode 2 when the PLL clock or HOCO is set to oscillate.
Note 4. The operating frequency range and operating voltage range differ depending on the operating mode. For details, see RX65N Group, RX651 Group User's Manual: Hardware.

Table 3.13 Constants (User Changeable) Used by Sample Code (3/3)

Constant Name	Setting Value	Contents
MSTP_STATE_EXDMAC*1	MODULE_STOP_ DISABLE	EXDMAC module stop state selection MODULE_STOP_DISABLE: Disable module stop MODULE_STOP_ENABLE: Transition to module stop
MSTP_STATE_DMACDTC*1	MODULE_STOP_ DISABLE	DMAC and DTC module stop state selection MODULE_STOP_DISABLE: Disable module stop MODULE_STOP_ENABLE: Transition to module stop
MSTP_STATE_STBYRAM*1	MODULE_STOP_ DISABLE	Standby RAM module stop state selection MODULE_STOP_DISABLE: Operating MODULE_STOP_ENABLE: Stopped
MSTP_STATE_ RAM2*1*2	MODULE_STOP_ DISABLE	RAM2 module stop state selection MODULE_STOP_DISABLE: Operating MODULE_STOP_ENABLE: Stopped
MSTP_STATE_ RAM0*1	MODULE_STOP_ DISABLE	RAM0 module stop state selection MODULE_STOP_DISABLE: Operating MODULE_STOP_ENABLE: Stopped
PIN_SIZE*3	144 *6 176 *7	Pin count of target device
REG_ROMWT*4*5	ROMWT_2WAIT	ROM wait cycle selection ROMWT_0WAIT: 0 wait cycles ROMWT_1WAIT: 1 wait cycle ROMWT_2WAIT: 2 wait cycles
SEL_ROM_CACHE*4	B_USE	ROM cache operation enable/disable B_USE: Used (operation enabled) B_NOT_USE: Not used (operation disabled)

Note 1. Change the settings values in r_init_stop_module.h to match the target system.
Note 2. This setting is only for r01an3034_rx65n_2m.
Note 3. Change the settings values in r_init_port_initialize.h to match the target system.
Note 4. Change the setting values in r_init_clock.h to match the target system.
Note 5. When ICLK is faster than 100 MHz, set it to 2 waits cycle.
When ICLK is faster than 50 MHz, do not set it to 0 wait.
Note 6. For initial setting of r01an3034_rx65n_1m.
Note 7. For initial setting of r01an3034_rx65n_2m.

Table 3.14 Constants (Non User-Changeable) Used by Sample Code

Constant Name	Setting Value	Contents
B_NOT_USE	0	Not used
B_USE	1	Used
B_USE_PLL_MAIN	2	Used the PLL clock (clock source : Main clock)
B_USE_PLL_HOCO	3	Used the PLL clock (clock source : HOCO)
CL_LOW	02h	Sub-clock: Drive capacity for low clock
CL_STD	0Ch	Sub-clock: Drive capacity for standard clock
FREQ_16MHz	00h	HOCO frequency: 16 MHz
FREQ_18MHz	01h	HOCO frequency: 18 MHz
FREQ_20MHz	02h	HOCO frequency: 20 MHz
CLK_PLL	0400h	Clock source: PLL
CLK_HOCO	0100h	Clock source: HOCO
CLK_SUB	0300h	Clock source: Sub-clock
CLK_MAIN	0200h	Clock source: Main clock
ROMWT_0WAIT	0	ROM wait cycles: 0 wait cycles
ROMWT_1WAIT	1	ROM wait cycles: 1 wait cycle
ROMWT_2WAIT	2	ROM wait cycles: 2 wait cycles
REG_SCKCR*1	21C2 1222h (PLL selected) 10C1 0111h (HOCO selected) 55C5 5555h (other than the above)	Internal clock division ratio and BCLK/SDCLK pin output control settings (setting value of SCKCR register)
OPCM_HIGH	00h	Operating power control mode: High-speed operating mode
OPCM_LOW_1	06h	Operating power control mode: Low-speed operating mode 1
OPCM_LOW_2	07h	Operating power control mode: Low-speed operating mode 2
SUB_CLOCK_CYCLE	(1,000,000,000L / SUB_CLOCK_Hz)	Sub-clock cycle (ns)
FOR_CMT0_TIME	121212L	Count cycle (ns) of timer for RTC software wait cycles (CMT0) = 1/LOCO (264 kHz) ×32 (LOCO = 264 kHz (max.), PCLKB ×1/32)
MODULE_STOP_ENABLE	1	Transition to module stop state
MODULE_STOP_DISABLE	0	Cancel module stop state

Note 1. The setting value differs depending on the clock source of the selected system clock.

Table 3.15 Constants for 177- and 176-Pin Products (PIN_SIZE=177 or PIN_SIZE=176)

Constant Name	Setting Value	Contents
DEF_P0PDR	0x50	Port P0 direction register setting value
DEF_P1PDR	0x00	Port P1 direction register setting value
DEF_P2PDR	0x00	Port P2 direction register setting value
DEF_P3PDR	0x00	Port P3 direction register setting value
DEF_P4PDR	0x00	Port P4 direction register setting value
DEF_P5PDR	0x00	Port P5 direction register setting value
DEF_P6PDR	0x00	Port P6 direction register setting value
DEF_P7PDR	0x00	Port P7 direction register setting value
DEF_P8PDR	0x00	Port P8 direction register setting value
DEF_P9PDR	0x00	Port P9 direction register setting value
DEF_PAPDR	0x00	Port PA direction register setting value
DEF_PBPDR	0x00	Port PB direction register setting value
DEF_PCPDR	0x00	Port PC direction register setting value
DEF_PDPDR	0x00	Port PD direction register setting value
DEF_PEPDR	0x00	Port PE direction register setting value
DEF_PFPDR	0xC0	Port PF direction register setting value
DEF_PGPDR	0x00	Port PG direction register setting value
DEF_PJPDR	0xD0	Port PJ direction register setting value

Note 1. The constants of pins 177 and 176 are defined only for r01an3034_rx65n_2m.

Table 3.16 Constants for 145- and 144-Pin Products (PIN_SIZE=145 or PIN_SIZE=144)

Constant Name	Setting Value	Contents
DEF_P0PDR	0x50	Port P0 direction register setting value
DEF_P1PDR	0x03	Port P1 direction register setting value
DEF_P2PDR	0x00	Port P2 direction register setting value
DEF_P3PDR	0x00	Port P3 direction register setting value
DEF_P4PDR	0x00	Port P4 direction register setting value
DEF_P5PDR	0x80	Port P5 direction register setting value
DEF_P6PDR	0x00	Port P6 direction register setting value
DEF_P7PDR	0x00	Port P7 direction register setting value
DEF_P8PDR	0x30	Port P8 direction register setting value
DEF_P9PDR	0xF0	Port P9 direction register setting value
DEF_PAPDR	0x00	Port PA direction register setting value
DEF_PBPDR	0x00	Port PB direction register setting value
DEF_PCPDR	0x00	Port PC direction register setting value
DEF_PDPDR	0x00	Port PD direction register setting value
DEF_PEPDR	0x00	Port PE direction register setting value
DEF_PFPDR	0xDF	Port PF direction register setting value
DEF_PGPDR[*1]	0xFF	Port PG direction register setting value
DEF_PJPDR	0xD7	Port PJ direction register setting value

Note 1. This setting is only for r01an3034_rx65n_2m.

Table 3.17 Constants for 100-Pin Products (PIN_SIZE=100)

Constant Name	Setting Value	Contents
DEF_P0PDR	0x5F	Port P0 direction register setting value
DEF_P1PDR	0x03	Port P1 direction register setting value
DEF_P2PDR	0x00	Port P2 direction register setting value
DEF_P3PDR	0x00	Port P3 direction register setting value
DEF_P4PDR	0x00	Port P4 direction register setting value
DEF_P5PDR	0xC0	Port P5 direction register setting value
DEF_P6PDR	0xFF	Port P6 direction register setting value
DEF_P7PDR	0xFF	Port P7 direction register setting value
DEF_P8PDR	0xFF	Port P8 direction register setting value
DEF_P9PDR	0xFF	Port P9 direction register setting value
DEF_PAPDR	0x00	Port PA direction register setting value
DEF_PBPDR	0x00	Port PB direction register setting value
DEF_PCPDR	0x00	Port PC direction register setting value
DEF_PDPDR	0x00	Port PD direction register setting value
DEF_PEPDR	0x00	Port PE direction register setting value
DEF_PFPDR	0xFF	Port PF direction register setting value
DEF_PGPDR*1	0xFF	Port PG direction register setting value
DEF_PJPDR	0xF7	Port PJ direction register setting value

Note 1. This setting is only for r01an3034_rx65n_2m.

Table 3.18 Constants for 64-Pin Products (PIN_SIZE=64)

Constant Name	Setting Value	Contents
DEF_P0PDR	0xDF*1	Port P0 direction register setting value
DEF_P1PDR	0x33	Port P1 direction register setting value
DEF_P2PDR	0x3F	Port P2 direction register setting value
DEF_P3PDR	0x0C	Port P3 direction register setting value
DEF_P4PDR	0xF0	Port P4 direction register setting value
DEF_P5PDR	0xF7	Port P5 direction register setting value
DEF_P6PDR	0xFF	Port P6 direction register setting value
DEF_P7PDR	0xFF	Port P7 direction register setting value
DEF_P8PDR	0xFF	Port P8 direction register setting value
DEF_P9PDR	0xFF	Port P9 direction register setting value
DEF_PAPDR	0x29	Port PA direction register setting value
DEF_PBPDR	0x1F	Port PB direction register setting value
DEF_PCPDR	0x0C	Port PC direction register setting value
DEF_PDPDR	0x03	Port PD direction register setting value
DEF_PEPDR	0x38	Port PE direction register setting value
DEF_PFPDR	0xFF	Port PF direction register setting value
DEF_PGPDR*2	0xFF	Port PG direction register setting value
DEF_PJPDR	0xFF	Port PJ direction register setting value

Note 1. For TFBGA 64-pin, P05 is nonexistent port, set the value to 0xFF.
Note 2. This setting is only for r01an3034_rx65n_2m.

3.8 Functions

Table 3.19 lists the functions. The function list is same for r01an3034_rx65n_1m and r01an3034_rx65n_2m.

Table 3.19 Functions

Function Name	Outline
main	Main processing routine
R_INIT_StopModule	Disable peripheral functions still running after a reset
R_INIT_Port_Initialize	Initial nonexistent port settings
R_INIT_Clock	Initial clock settings
CGC_oscillation_main	Main clock oscillation enable
CGC_oscillation_PLL	PLL clock oscillation enable
CGC_oscillation_HOCO	HOCO clock oscillation enable
CGC_oscillation_sub	Sub-clock oscillation enable
CGC_disable_subclk	Sub-clock disable
oscillation_subclk	Sub-clock oscillation enable
resetting_wtcr_subclk	Sub-clock wait control register resetting
init_rtc	Initialize RTC
cmt0_wait	Software wait cycles using CMT0
set_ad_conversion_time	A/D sequential conversion time setting

3.9 Function Specifications

The following tables list the sample code function specifications. The function specification is same for r01an3034_rx65n_1m and r01an3034_rx65n_2m.

main	
Outline	Main processing routine
Header	None
Declaration	void main(void)
Description	Calls the settings function for disabling peripheral functions still running after a reset, the initial nonexistent port settings function, and the initial clock settings function.
Arguments	None
Return Value	None

R_INIT_StopModule	
Outline	Disable peripheral functions still running after a reset
Header	r_init_stop_module.h
Declaration	void R_INIT_StopModule(void)
Description	Makes settings to transition to the module stop state.
Arguments	None
Return Value	None
Remarks	In the sample code, no transition to the module stop state occurs.

R_INIT_Port_Initialize	
Outline	Initial nonexistent port settings
Header	r_init_port_initialize.h
Declaration	void R_INIT_Port_Initialize(void)
Description	Makes initial settings to the port direction registers corresponding to the pins of nonexistent port.
Arguments	None
Return Value	None
Remarks	The setting in the sample code of r01an3034_rx65n_1m (PIN_SIZE=144) is for 144-pin products. The setting in the sample code of r01an3034_rx65n_2m (PIN_SIZE=176) is for 176-pin products. When writing in byte units to PDR or PODR registers containing nonexistent ports after this function has been called, set the direction control bits corresponding to the nonexistent ports to 1 and the port output data storage bits corresponding to the nonexistent ports to 0.

R_INIT_Clock	
Outline	Initial clock settings
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Makes initial clock settings and specifies the number of ROM wait cycles.
Arguments	None
Return Value	None
Remarks	In the sample code processing is selected that sets the PLL clock as the system clock, specifies two ROM wait cycles, and does not use a sub-clock. The function set_ad_conversion_time, which is called by R_INIT_Clock, must be called when the value of the PSW.I bit is 0 and the value of the ADCSR.ADST bit is 0. Therefore, clear the PSW.I bit to 0 (interrupts disabled) and the ADCSR.ADST bit to 0 before calling R_INIT_Clock.

CGC_oscillation_main	
Outline	Main clock oscillation enable
Header	r_init_clock.h
Declaration	void CGC_oscillation_main (void)
Description	Sets the drive capacity of the main clock and sets the MOSCWTCR register, then starts oscillation of the main clock. After this, waits for the main clock oscillation stabilization waiting time using hardware.
Arguments	None
Return Value	None

CGC_oscillation_PLL	
Outline	PLL clock oscillation enable
Header	r_init_clock.h
Declaration	void CGC_oscillation_PLL (void)
Description	Sets the PLL input division ratio and frequency multiplication factor, then starts oscillation of the PLL clock. After this, waits for the PLL clock oscillation stabilization waiting time using hardware.
Arguments	None
Return Value	None

CGC_oscillation_HOCO	
Outline	HOCO clock oscillation enable
Header	r_init_clock.h
Declaration	void CGC_oscillation_HOCO (void)
Description	Sets the HOCO frequency, then starts oscillation of the HOCO. After this, waits for the HOCO oscillation stabilization waiting time using hardware.
Arguments	None
Return Value	None

CGC_oscillation_sub	
Outline	Sub-clock oscillation enable
Header	r_init_clock.h
Declaration	void CGC_oscillation_sub (void)
Description	Makes settings for using the sub-clock as the system clock or as the RTC count source, or for both.
Arguments	None
Return Value	None

CGC_disable_subclk	
Outline	Sub-clock disable
Header	r_init_clock.h
Declaration	void CGC_disable_subclk (void)
Description	Makes settings for when the sub-clock is not used as the system clock or as the RTC count source.
Arguments	None
Return Value	None

oscillation_subclk	
Outline	Sub-clock oscillation enable
Header	None
Declaration	static void oscillation_subclk (void)
Description	Makes settings to start sub-clock oscillation.
Arguments	None
Return Value	None

resetting_wtcr_subclk	
Outline	Sub-clock wait control register resetting
Header	None
Declaration	static void resetting_wtcr_subclk (void)
Description	Resets the wait control register when returning from software standby mode. In this case the wait control register is set to the minimum value.
Arguments	None
Return Value	None

init_rtc	
Outline	Initialize RTC
Header	None
Declaration	static void init_rtc (void)
Description	Makes initial settings for the RTC (clock supply setting and RTC software reset).
Arguments	None
Return Value	None

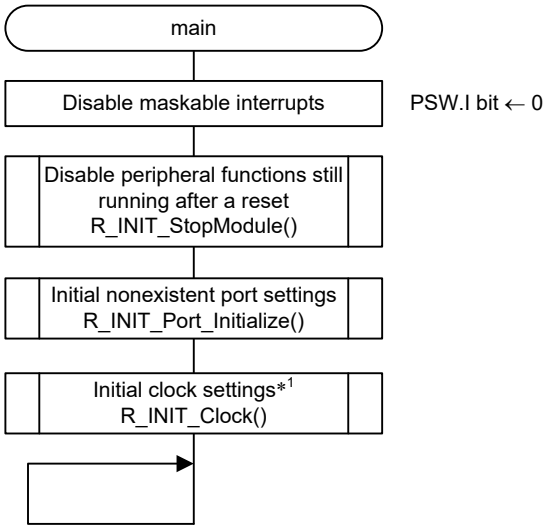
cmt0_wait		
Outline	Software wait cycles using CMT0	
Header	None	
Declaration	static void cmt0_wait (uint32_t cnt)	
Description	Used when waiting before writing to the RTC register.	
Arguments	uint32_t cnt	Wait time cnt = Wait time (ns) ÷ FOR_CMT0_TIME*1
Return Value	None	
Remarks	Note 1. The duration of FOR_CMT0_TIME is calculated based on LOCO = 264 kHz (max.). The actual wait time will differ depending on the LOCO frequency.	

set_ad_conversion_time	
Outline	A/D sequential conversion time setting
Header	None
Declaration	static void set_ad_conversion_time (void)
Description	Sets the sequential conversion time of S12AD unit 1 to medium speed.
Arguments	None
Return Value	None
Remarks	The ADSAM register, which is manipulated by this function, must be overwritten when the value of the PSW.I bit is 0 and the value of the ADCSR.ADST bit is 0. Therefore, clear the PSW.I bit to 0 (interrupts disabled) and the ADCSR.ADST bit to 0 before calling this function.

3.10 Flowcharts

3.10.1 Main Processing

Figure 3.1 shows the main processing. The flowcharts of r01an3034_rx65n_1m and r01an3034_rx65n_2m are the same except for the module stop of peripheral functions that are running after reset and the initial setting of non-existing port.

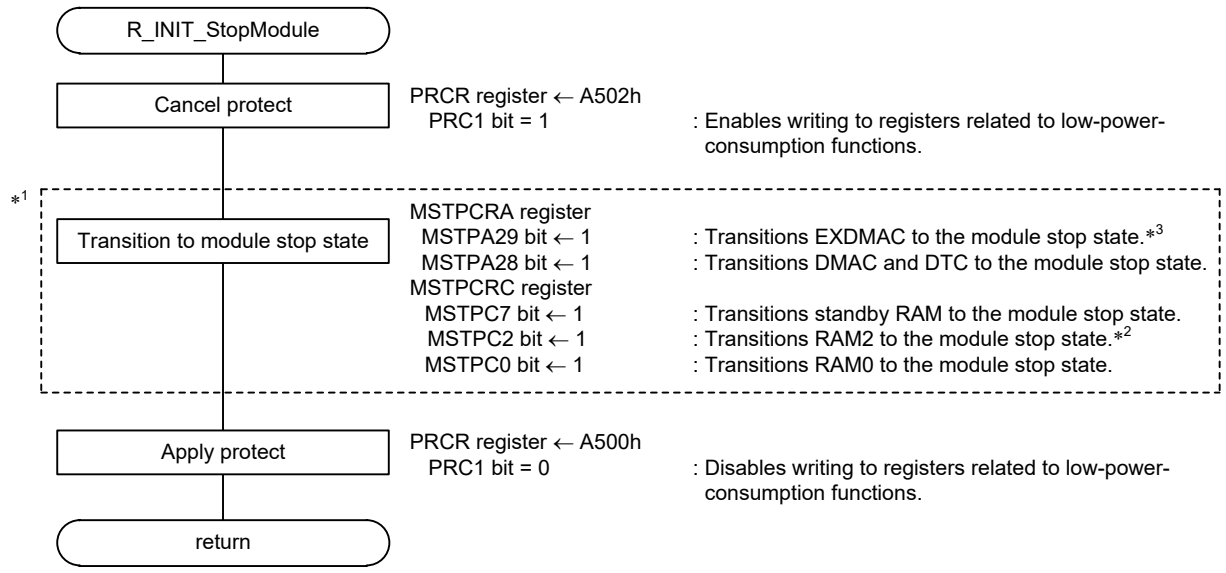


Note 1. Clear the PSW.I bit to 0 (interrupts disabled) and the ADCSR.ADST bit to 0 before calling this function. The default value of the ADCSR.ADST bit is 0, so the setting of ADCSR.ADST is not checked by the program accompanying this application note. If it is possible that the ADCSR.ADST may be manipulated before the R_INIT_Clock function is called, add appropriate processing to clear the ADCSR.ADST bit to 0.

Figure 3.1 Main Processing

3.10.2 Disable Peripheral Functions Still Running After a Reset

Figure 3.2 is a flowchart of the processing for disabling of peripheral functions still running after a reset.

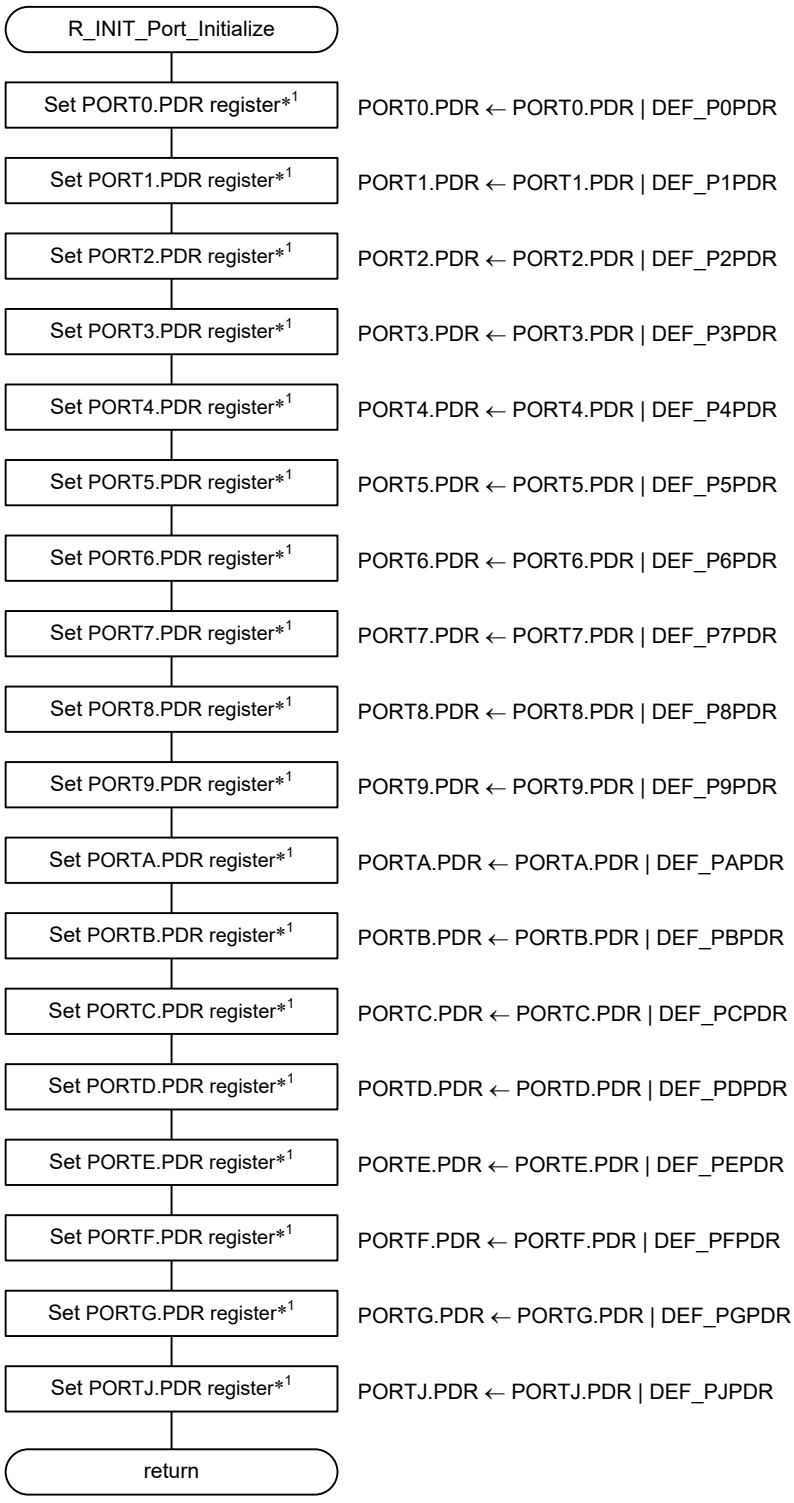


Note 1. In the sample code the module stop state is canceled. To transition to the module stop state, set the corresponding constant #define MSTP_STATE_<target module name> to 1.
Note 2. MSTPC2 bit (RAM2) setting is only for r01an3034_rx65n_2m.
Note 3. EXDMAC is function used only with device having 100-pin and above.

Figure 3.2 Disable Peripheral Functions Still Running After a Reset

3.10.3 Initial Nonexistent Port Settings

Figure 3.3 is a flowchart of the processing for making initial nonexistent port settings.



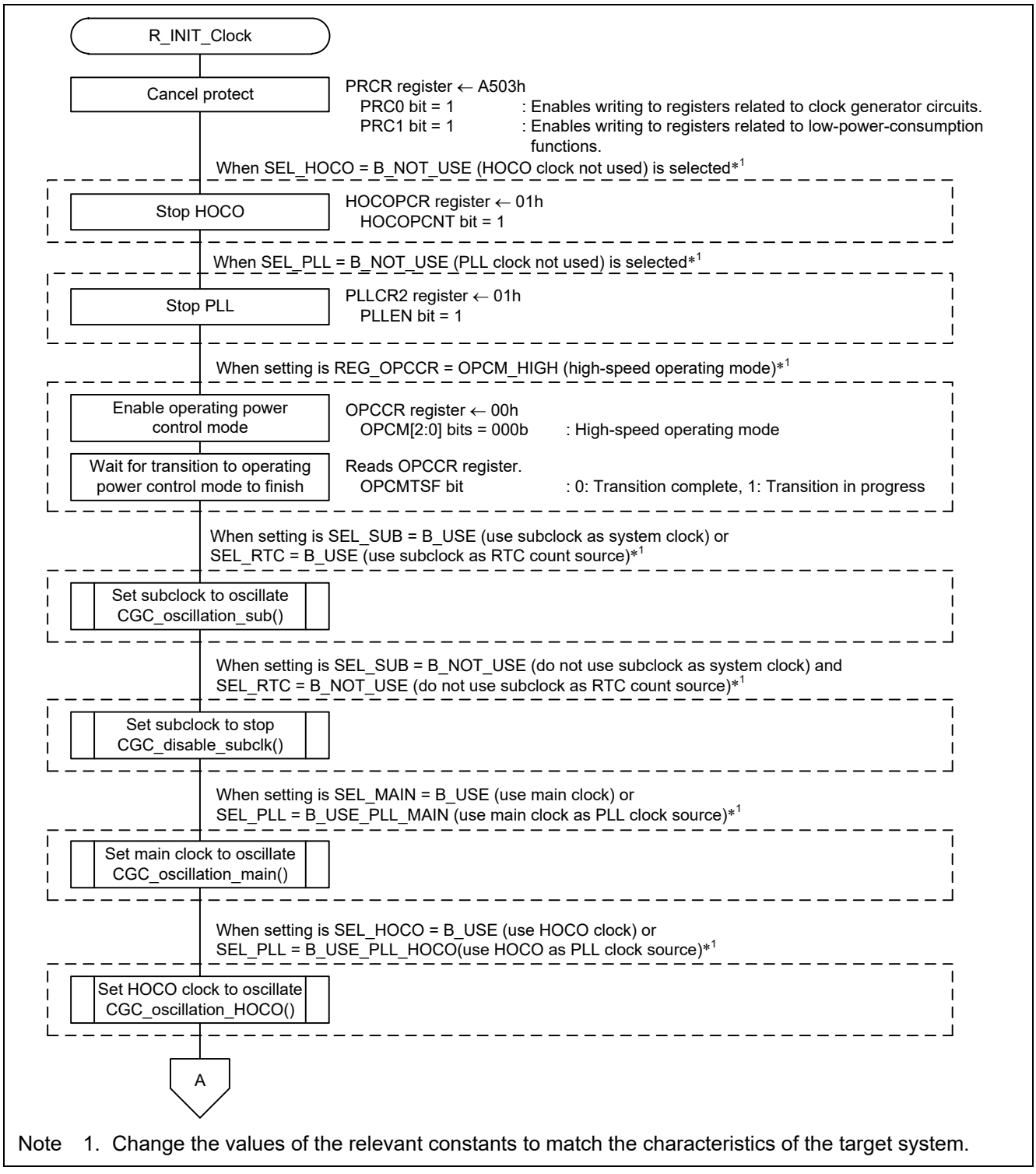
Note 1. No processing of settings is performed for registers in which all bits correspond to existing pins (omitted during compile).

Note 2. PORTG.PDR setting is only for r01an3034_rx65n_2m.

Figure 3.3 Initial Nonexistent Port Settings

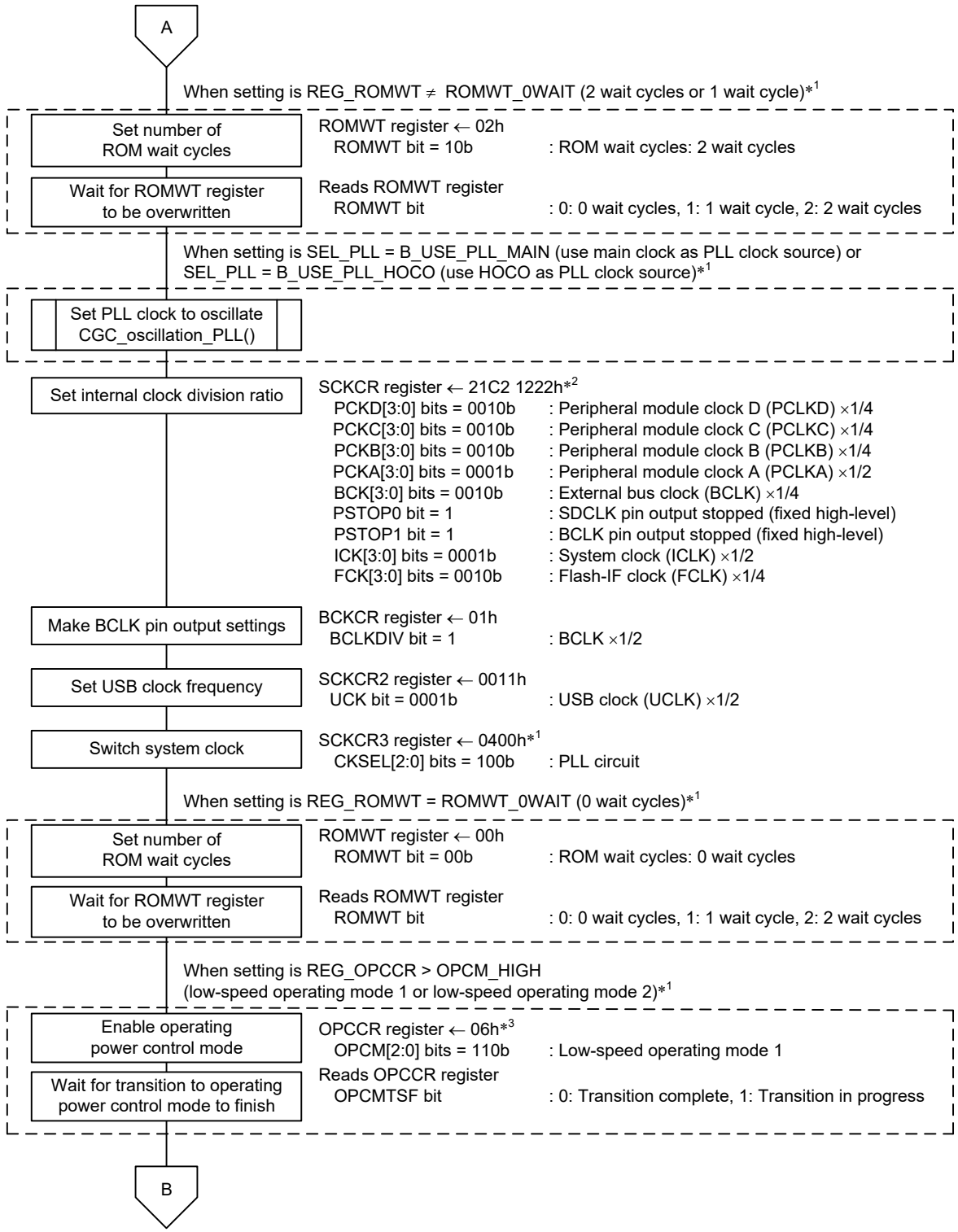
3.10.4 Initial Clock Settings

Figure 3.4, Figure 3.5, and Figure 3.6 are flowcharts of the processing for making initial clock settings (1/3), (2/3), and (3/3).



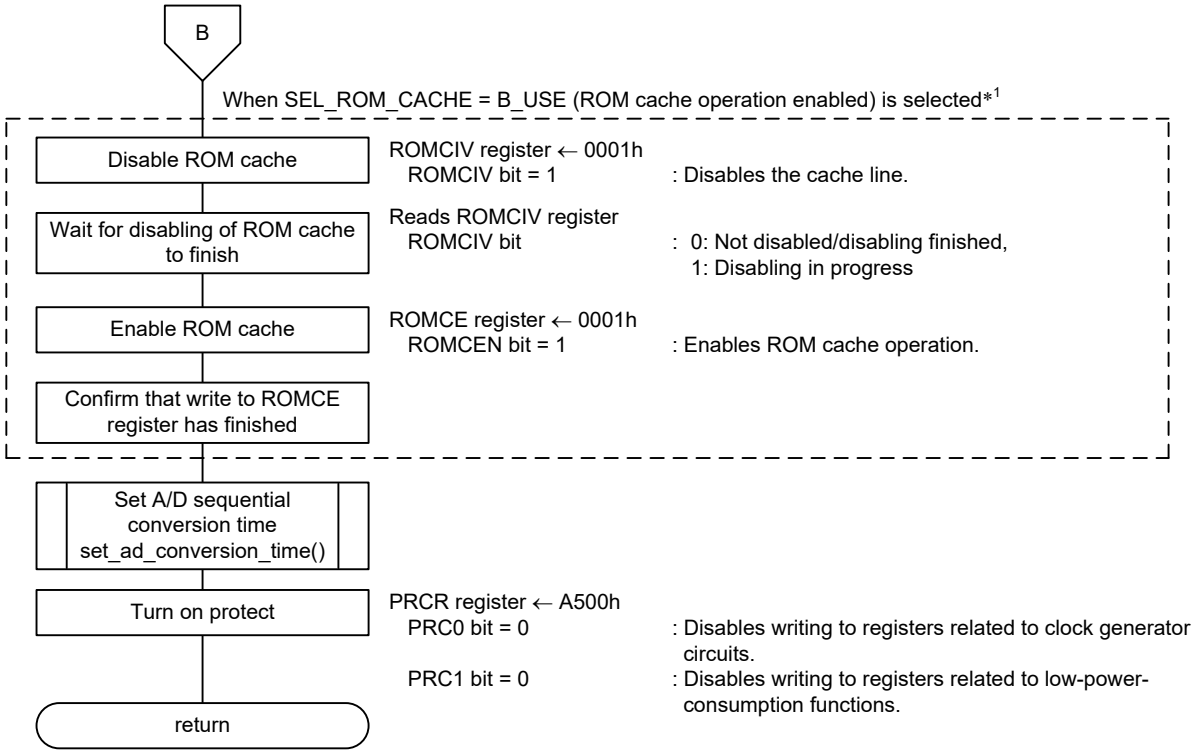
Note 1. Change the values of the relevant constants to match the characteristics of the target system.

Figure 3.4 Initial Clock Settings (1/3)



Note 1. Change the values of the relevant constants to match the characteristics of the target system.
Note 2. The setting values differ depending on the system clock selected by the constant.
Note 3. The setting values differ depending on the operating power control mode selected by the constant.

Figure 3.5 Initial Clock Settings (2/3)

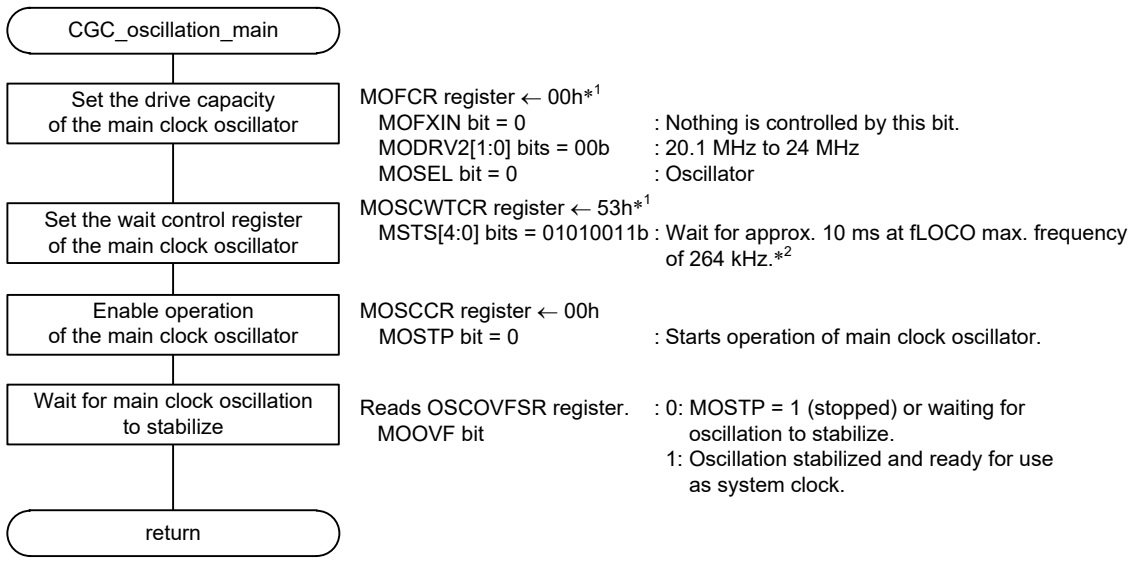


Note 1. Change the values of the relevant constants to match the characteristics of the target system.

Figure 3.6 Initial Clock Settings (3/3)

3.10.5 Main Clock Oscillation Enable

Figure 3.7 is a flowchart of the processing for starting oscillation of the main clock.

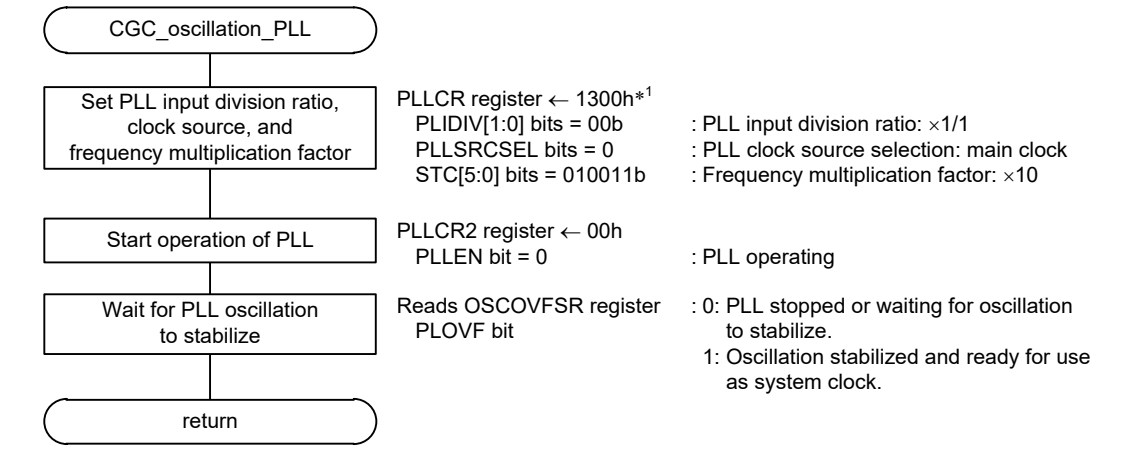


Note 1. Change the values of the relevant constants to match the characteristics of the target system.
Note 2. The sample code accompanying this application note uses the initial value of the register.

Figure 3.7 Main Clock Oscillation Enable

3.10.6 PLL Clock Oscillation Enable

Figure 3.8 is a flowchart of the processing for starting oscillation of the PLL clock.

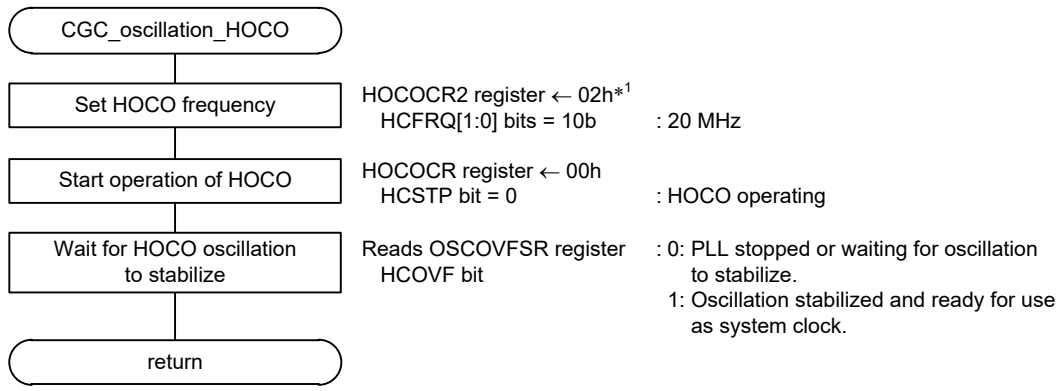


Note 1. Change the values of the relevant constants to match the characteristics of the target system.

Figure 3.8 PLL Clock Oscillation Enable

3.10.7 HOCO Clock Oscillation Enable

Figure 3.9 is a flowchart of the processing for starting oscillation of the HOCO clock.

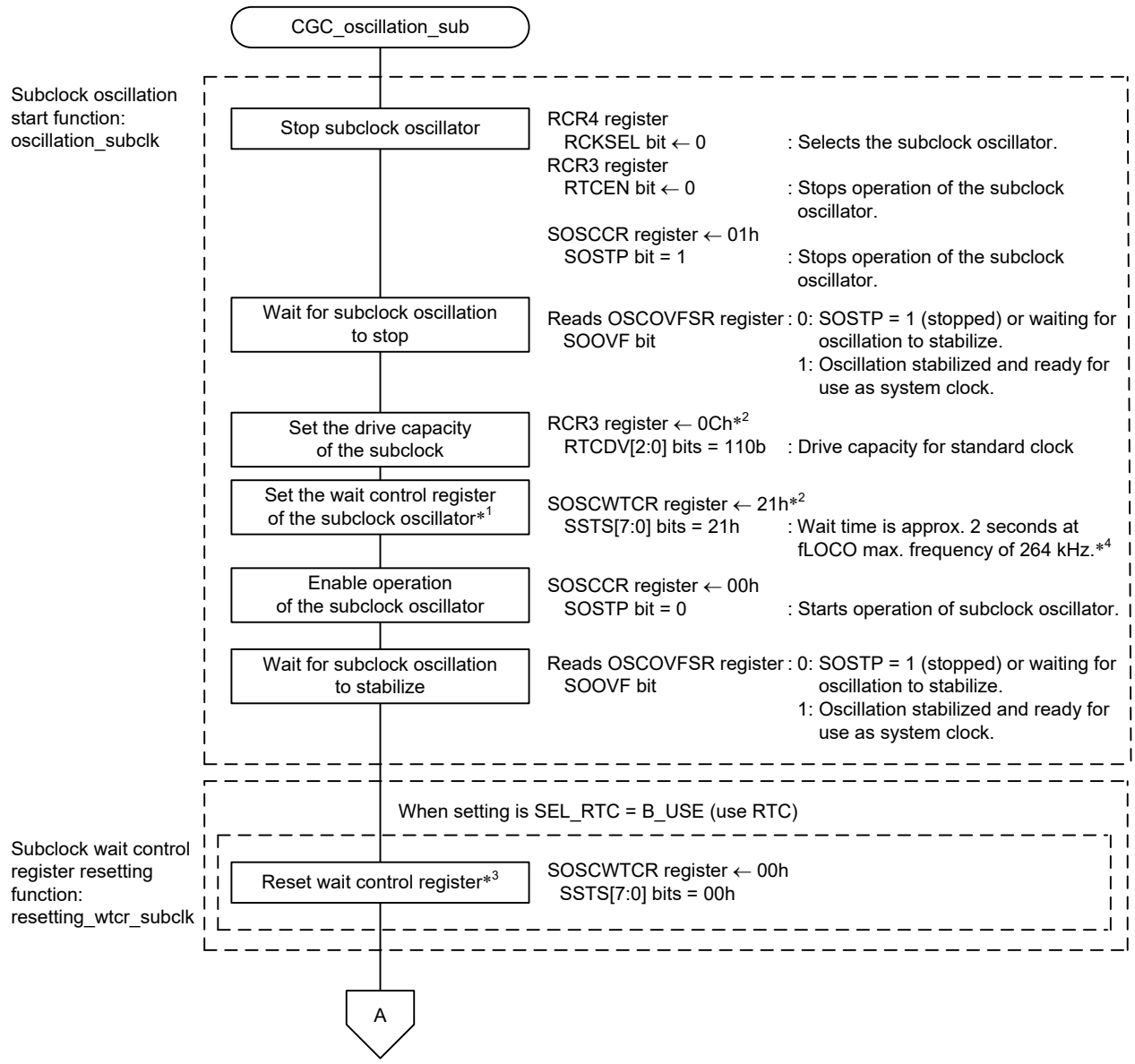


Note 1. Change the values of the relevant constants to match the characteristics of the target system.

Figure 3.9 HOCO Clock Oscillation Enable

3.10.8 Sub-clock Oscillation Enable

Figure 3.10 and Figure 3.11 are flowcharts of the processing for starting oscillation of the sub-clock.



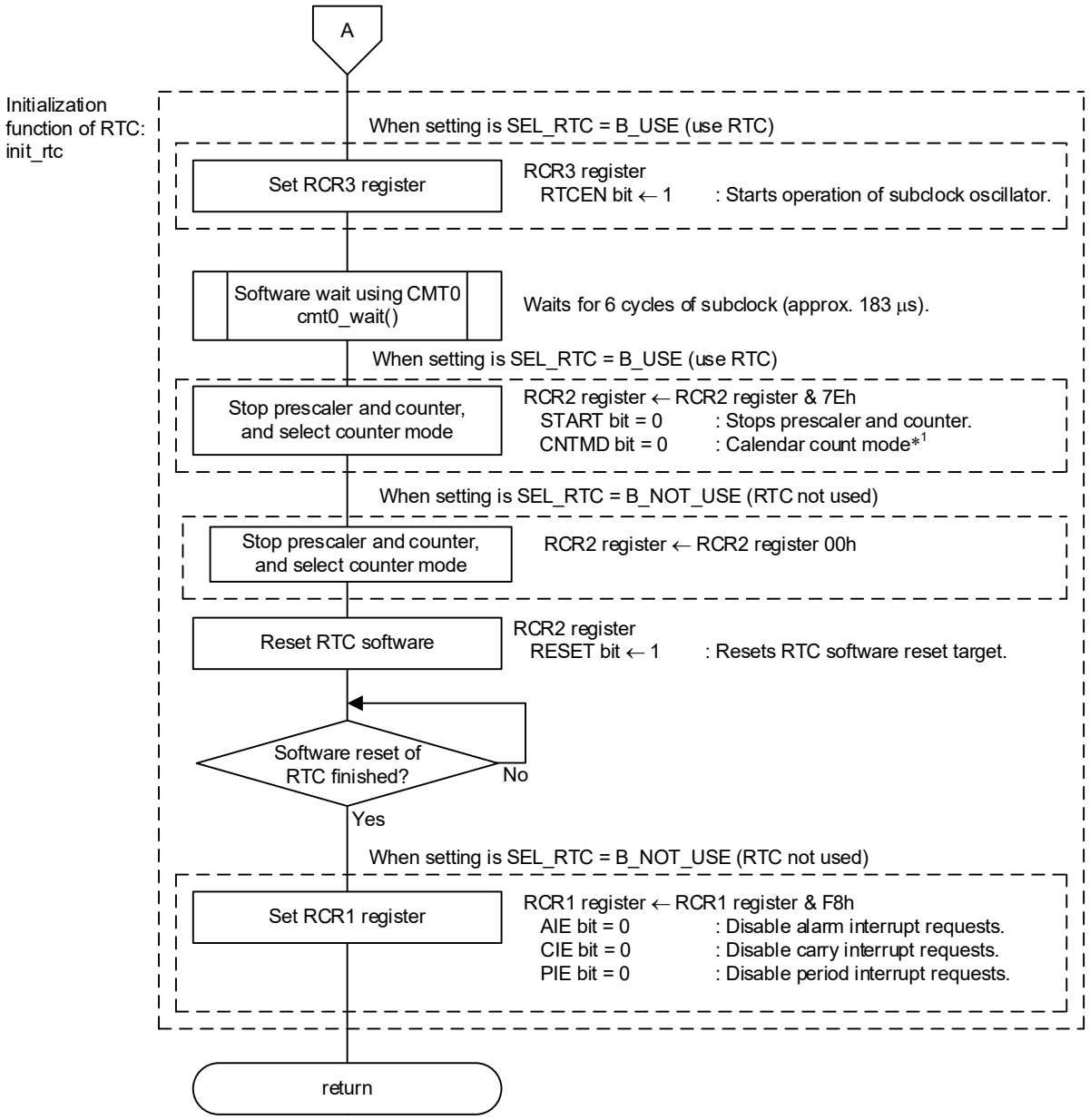
Note 1. Use the formulas in Sub-Clock Oscillator Wait Control Register (SOSCWTCR), in RX65N Group, RX651 Group User's Manual: Hardware to calculate the waiting time. Note that the sample code accompanying this application note uses the initial value of the register.

Note 2. Change the value of the constant to match the characteristics of the target system.

Note 3. When returning from software standby mode with the sub-clock selected as the system clock, supply of clock signals to the internals of the microcontroller starts after the time specified by bits SSTS4 to SSTS0 in SOSCWTCR has elapsed following generation of the return source. However, the sub-clock does not stop operating in software standby mode if value of the RTCEN bit in RCR3 is 1 (operates), in which case no oscillation stabilization time is needed for the sub-clock when returning from software standby mode. Therefore, bits SSTS7 to SSTS0 in SOSCWTCR are reset to 00h to minimize the sub-clock oscillation stabilization wait time.

Note 4. The sample code accompanying this application note uses the initial value of the register.

Figure 3.10 Sub-clock Oscillation Enable (1/2)



Note 1. The sample code accompanying this application note selects the calendar count mode. Change the values of the constants to match the characteristics of the target system.

Figure 3.11 Sub-clock Oscillation Enable (2/2)

3.10.9 Sub-clock Disable

Figure 3.12 is a flowchart of the processing for stopping the sub-clock.

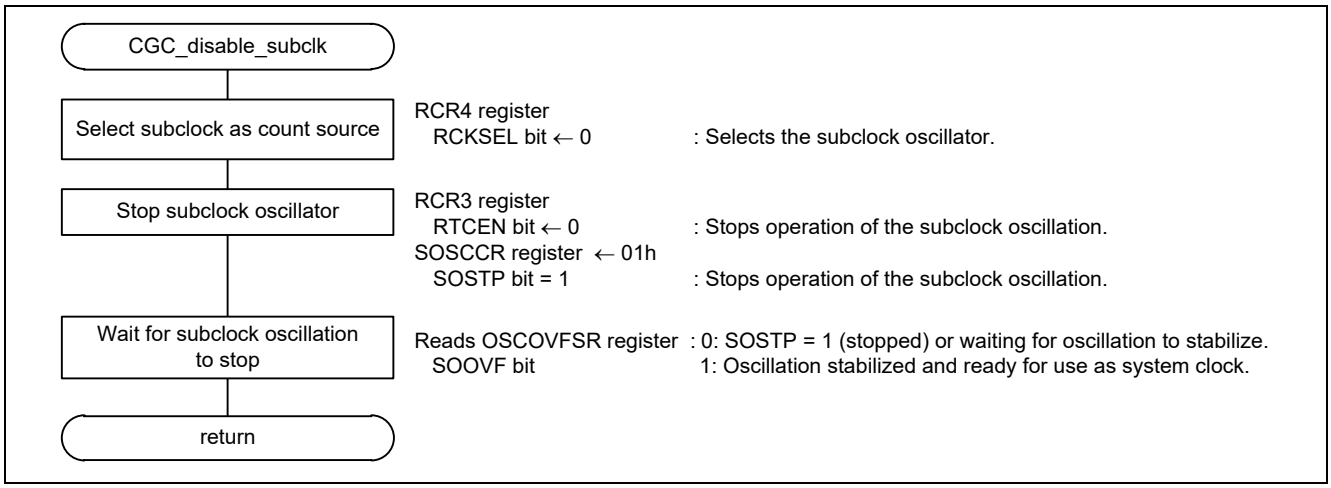


Figure 3.12 Sub-clock Disable

3.10.10 Software Wait Cycles Using CMT0

Figure 3.13 and Figure 3.14 are flowcharts of the processing for implementing a software wait using CMT0.

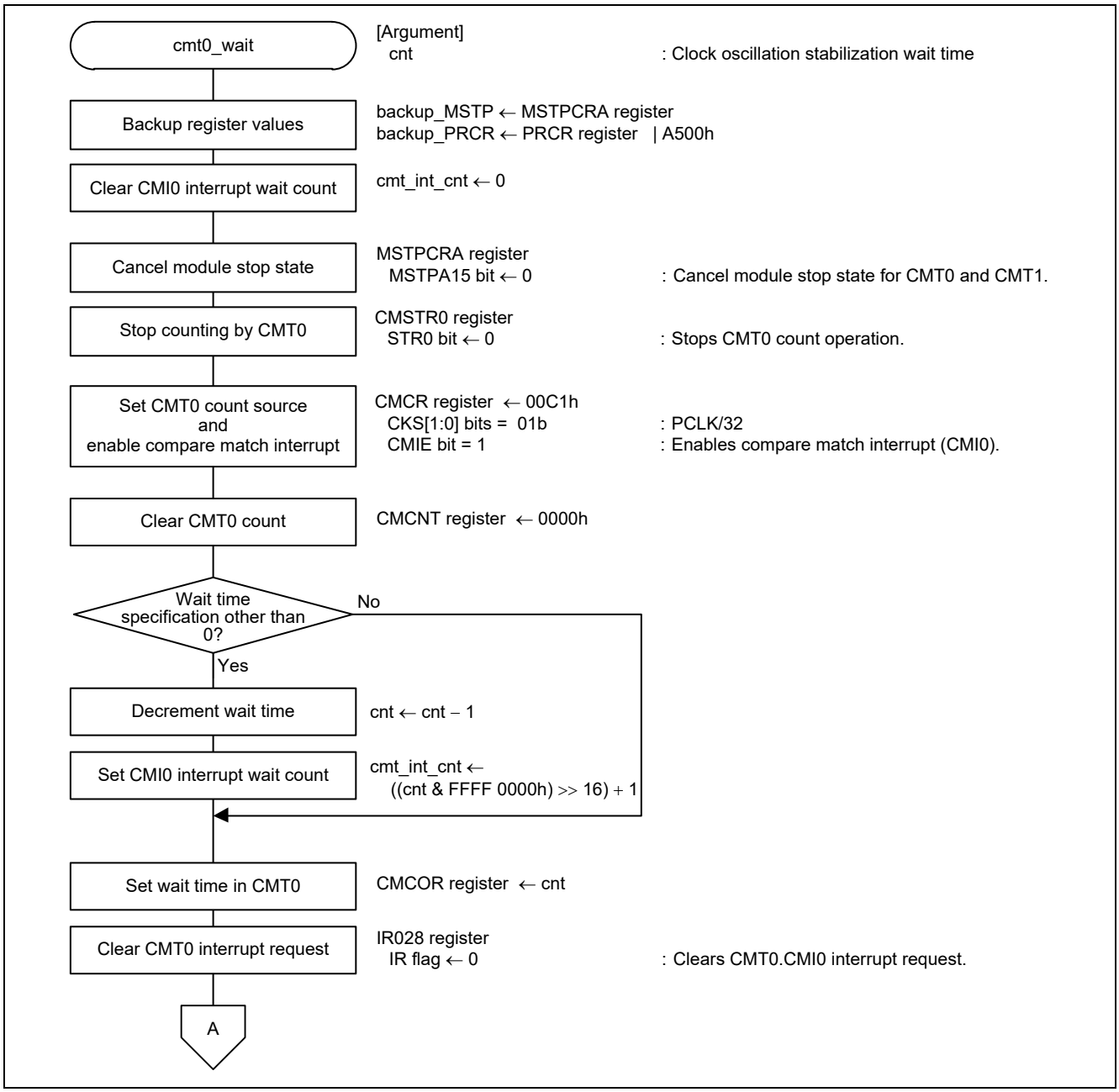
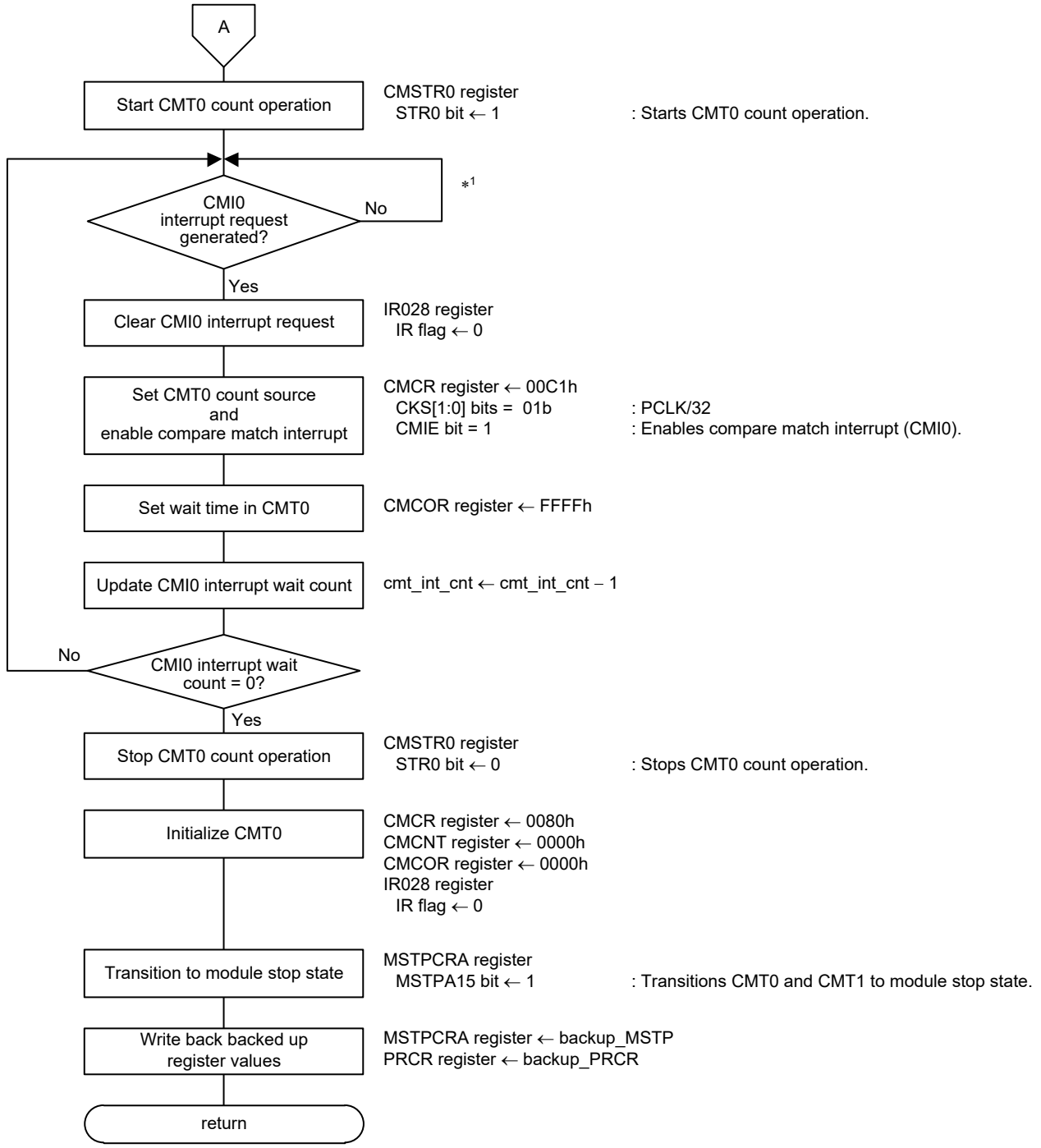


Figure 3.13 Software Wait Cycles Using CMT0 (1/2)



Note 1. When the watchdog timer (WDT) and independent watchdog timer (IWDT) are counting, use processing loop to refresh the WDT and IWDT.

Figure 3.14 Software Wait Cycles Using CMT0 (2/2)

3.10.11 A/D Sequential Conversion Time Settings

Figure 3.15 is a flowchart of the processing for making A/D sequential conversion time settings.

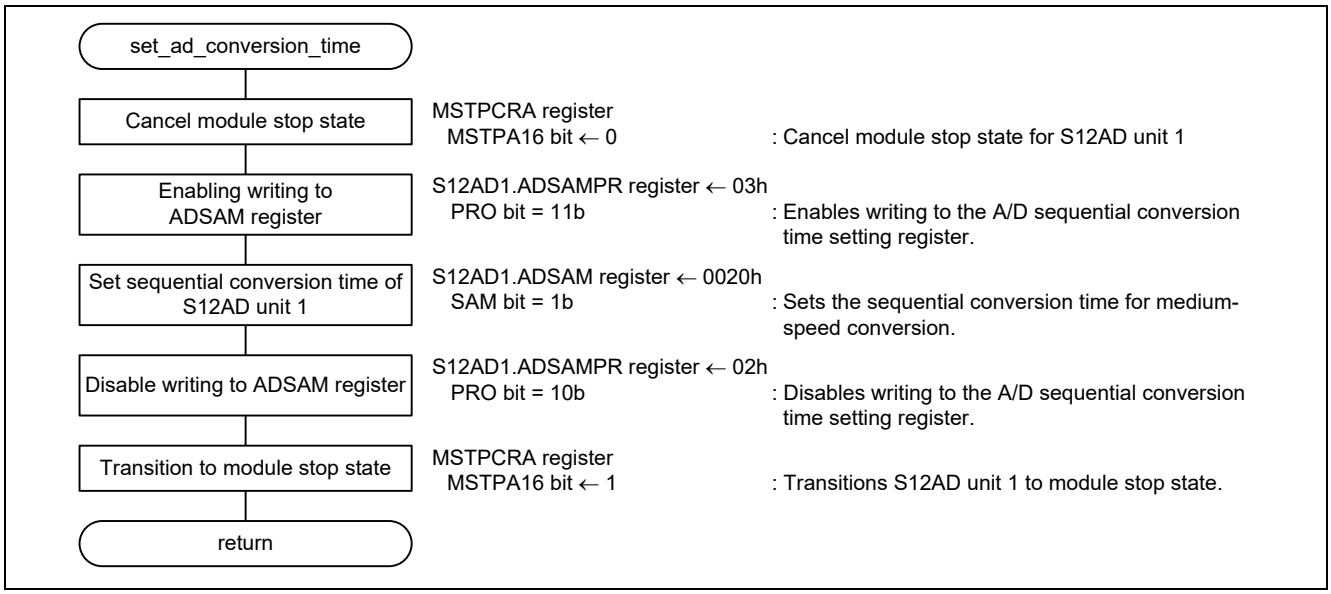


Figure 3.15 A/D Sequential Conversion Time Settings

4. Importing a Project

After importing the sample code, make sure to confirm build and debugger setting.

4.1 Importing a Project into e² studio

Follow the steps below to import your project into e² studio. Pictures may be different depending on the version of e² studio to be used.

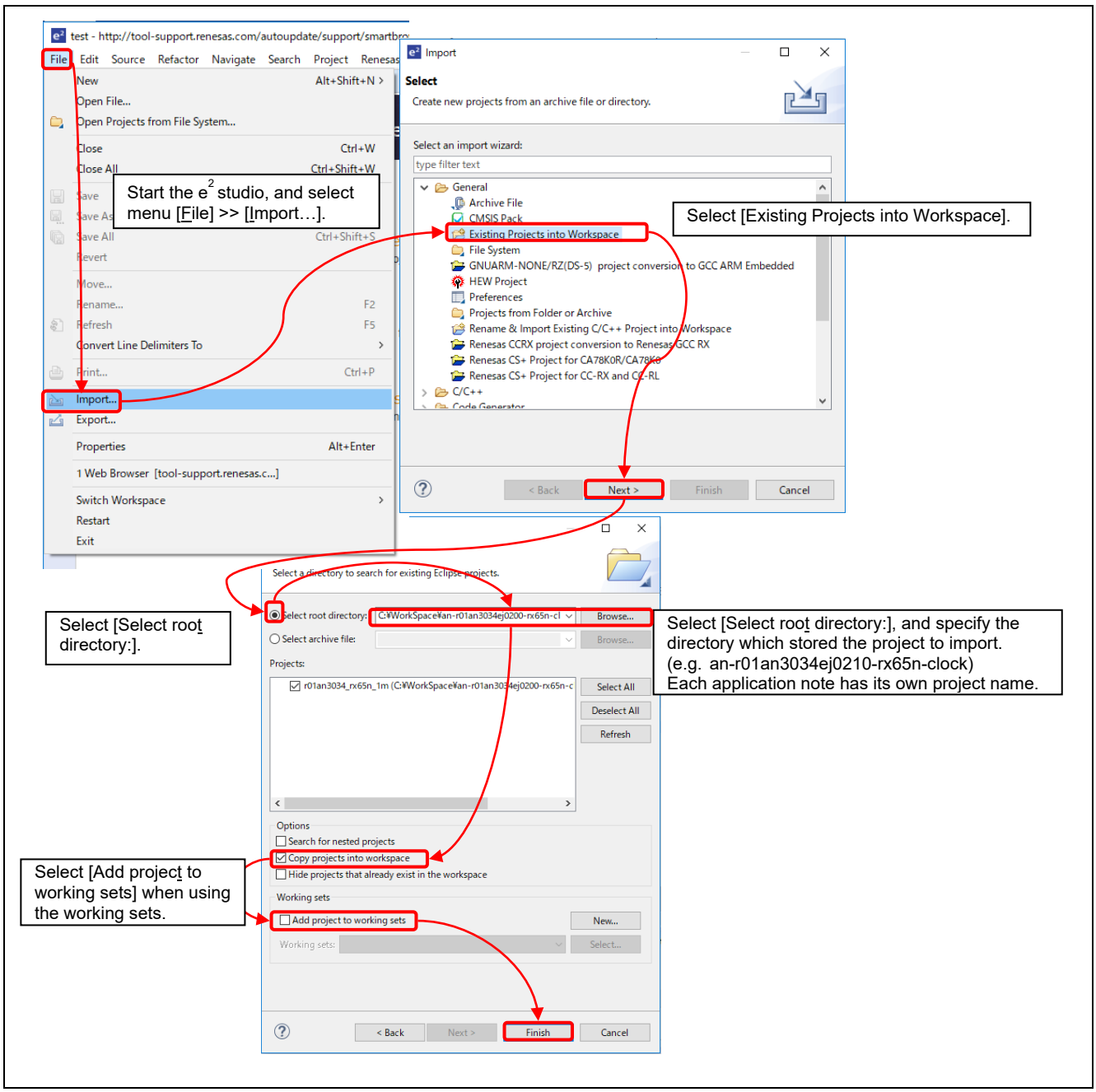


Figure 4.1 Importing a Project into e² studio

4.2 Importing a Project into CS+

Follow the steps below to import your project into CS+. Pictures may be different depending on the version of CS+ to be used.

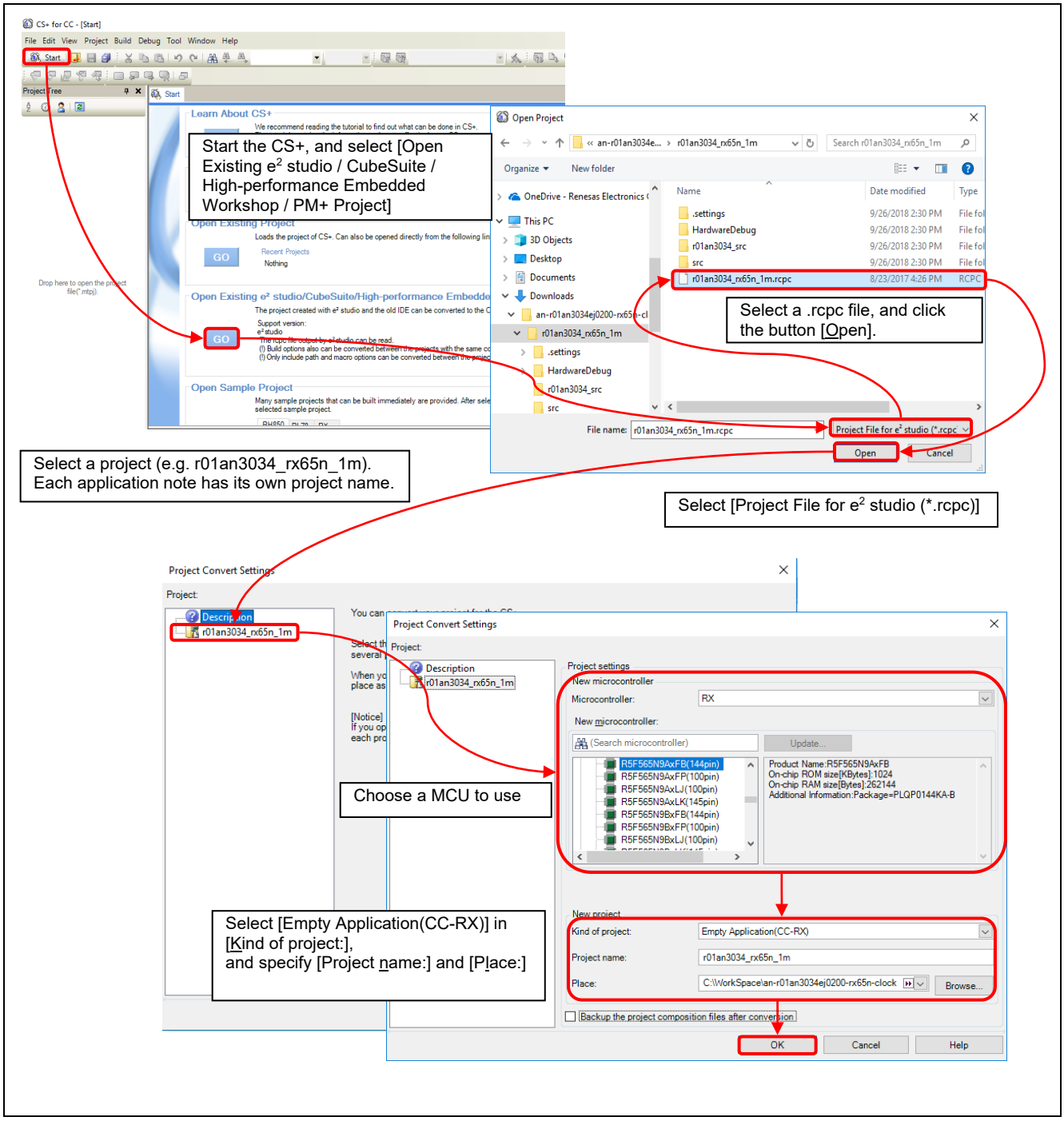


Figure 4.2 Importing a Project into CS+

5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

User's Manual: Hardware
RX65N Group, RX651 Group User's Manual: Hardware (R01UH0590)
(The latest version can be downloaded from the Renesas Electronics website.)

Technical Update/Technical News
(The latest version can be downloaded from the Renesas Electronics website.)

User's Manual: Development Tools
RX CC-RX Compiler User's Manual (R20UT3248)
(The latest version can be downloaded from the Renesas Electronics website.)

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.01	Oct.01.16	—	First edition issued
2.00	Oct.02.17	detail	The product specification of the ROM capacity of the 1.5 MB - 2 MB is added. Change of sample code name. r01an3034_rx65n → r01an3034_rx65n_1m
		5	Table 1.2 Clock Selection Examples, changed. Table 1.2 Clock Selection Examples, Note 1, changed.
		6	Table 2.1 Operation Confirmation Conditions in r01an3034_rx65n_1m, changed.
		12	3.4 Section Composition, added. Table 3.9 Option-Setting Memory Configured in the Sample Code, changed.
		13	Table 3.10 Constants (User Changeable) Used by Sample Code (1/3), changed.
		16	Table 3.13 Constants (Non User-Changeable) Used by Sample Code, changed.
		22	3.9 Function Specifications, changed. The item of the resetting_wtcr_subclk function was changed to a return value from remarks.
		26	Figure 3.4 Initial Clock Settings (1/3), changed.
		27	Figure 3.5 Initial Clock Settings (2/3), changed.
		28	Figure 3.6 Initial Clock Settings (3/3), changed.
		33	Figure 3.12 Sub-clock Disable, corrected.
		37	4. Importing a Project, changed.
		39	6. Reference Documents, changed.
		Program	The version of iodefine.h, changed. [r01an3034_rx65n_1m] The project name, changed. Added the following macro definition. - B_USE_PLL_MAIN - B_USE_PLL_HOCO Changed transition condition to CGC_oscillation_main function. Changed transition condition to CGC_oscillation_HOCO function. Changed transition condition to CGC_oscillation_PLL function. Added section setting of RAM 0. [r01an3034_rx65n_2m] Added a new project.
2.10	May.31.19	detail	The product support RX651 group 64 pin, added.
		1	Target Device, added.
		3	Section 1.2 Note for Disabling Peripheral Functions Still Running After a Reset, added.
		6,7	Table 2.1 and Table 2.2 Integrated development environment, C compiler, iodefine.h and Sample code version, changed. Spelling for “Nos” in operating frequency, changed. Table 2.1 Product No. for Board Used, changed.

Rev.	Date	Description	
		Page	Summary
2.10	May.31.19	8	Table 3.1 Note for Disabling Peripheral Functions Still Running After a Reset, added.
		10	Table 3.5 Nonexistent ports (64pin), added
		14	Section 3.7 Constants support 64 pin, added. Text description for Table 3.11 to Table 3.14 in paragraph, changed.
		19	Table 3.18 Constants for 64-Pin Products (PIN_SIZE=64), added.
		26	Figure 3.2 Disable Peripheral Functions Still Running After a Reset note, added.
		35	Figure 3.11 Sub-clock Oscillation Enable (2/2), changed.
		39	Importing a Project into e^2 studio and CS+, changed.
		41	Section 6 Reference Document for RX65N Group, RX651 Group User's Manual: Hardware, changed.
		Program	Bug fix of init_rtc function. [r01an3034_rx65n_1m and r01an3034_rx65n_2m] Fixed the bug that can not write the AIE bit of RCR1 register correctly in the init_rtc function. In the program before modification, an infinite loop may occur because the AIE bit of the RCR1 register can not be written correctly.
2.11	Feb. 1. 21	7	Table 2.1 Integrated development environment, C compiler, iodefine.h and Sample code version, changed.
		35	Figure 3.11 Subclock Oscillation Enable (2/2), changed.
		program	Technical update TN-RX*-A236B/E, supported.

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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