RX140 Group

Initial setting

Introduction

This document describes settings required after a reset such as clock settings, stop processing for active peripheral functions after a reset, and nonexistent port initialization according to usage conditions selected in the header files.

Target Device

- RX140 Group 80-pin version, ROM capacity: 128 KB, 256 KB
- RX140 Group 64-pin version, ROM capacity: 64 KB, 128 KB, 256 KB
- RX140 Group 48-pin version, ROM capacity: 64 KB, 128 KB, 256 KB
- RX140 Group 32-pin version, ROM capacity: 64 KB

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

Contents

1. Specifications

In the sample code, peripheral functions operating after a reset are stopped, and nonexistent port and clock settings are configured. The application note assumes processing at power-on (cold start).

1.1 Stopping Peripheral Functions Operating after a Reset

Some peripheral functions operate at power-on, and the module-stop function is disabled for others. These include the DTC and RAM0. Although the sample code includes processing for stopping these peripheral functions, it is not executed in the sample code. Change the constant as required to execute processing.

1.2 Configuring Nonexistent Ports

Port direction registers which have nonexistent ports need to be specified with determined values. In the sample code, initial values are set for port direction registers in 64-pin package and 64 KB of code flash memory. Change the values according to the product used.

1.3 Setting clocks

1.3.1 Overview

Clocks are configured in the following steps:

1. Sub-clock setting (including the associated RTC settings)
2. Main clock setting
3. PLL clock setting
4. HOCO clock setting
5. System clock switching

In this application note, the clock settings are switched by changing the constants defined in r_init_clock.h.

In the sample code, the HOCO clock is used as the system clock, and the main clock, the PLL clock, the sub-clock, and RTC are not used. Change the constant to select the required clock setting.

1.3.2 Clock Specifications Used in the Sample Code

Table 1.1 lists the clock specifications used in the sample code. Values such as the oscillation stabilization time are calculated using values listed in Table 1.1.

Table 1.1 Clock Specifications Used in the Sample Code

Clock	Oscillation Frequency	Oscillation Stabilization Time	Remarks
Crystal/ceramic resonator for the main clock	8 MHz	4.2 ms*2	Crystal used
Crystal for the sub-clock	32.768 kHz*1	1.3 sec.*2	For low CL
PLL clock	48 MHz	82μs*3	
HOCO clock	32 MHz*1*4	5 μs*3	

Notes: 1. The clock is disabled in the sample code.
2. The oscillation stabilization time of a crystal/ceramic resonator differs depending on the wiring pattern, conditions of oscillation parameters, and other settings in the user system. Contact the crystal/ceramic resonator manufacturer to evaluate the user system and provide an appropriate oscillation stabilization time.
3. Refer to the Electrical Characteristics in the User's Manual: Hardware.
4. Refer to the Option-Setting Memory in the User's Manual: Hardware.

1.3.3 Selecting Clocks

In the sample code, users can select the system clock source, whether clocks are oscillating or stopped, and other settings by changing the constants defined in r_init_clock.h. Refer to Table 4.6 and Table 4.7 for constants that can be changed.

Table 1.2 lists examples of clock selections and Table 1.3 lists examples of the sub-clock and RTC selections.

Table 1.2 Examples of Clock Selections

No.		1	2	3	4
System clock		Main clock	PLL	HOCO clock	Sub-clock
PLL clock		Stopped	Oscillating	Stopped	Stopped
Main clock		Oscillating	Oscillating	Stopped	Stopped
HOCO clock		Stopped	Stopped	Oscillating	Stopped
Sub-clock		Stopped*	Stopped*	Stopped*	Oscillating
Operating power control mode		High-speed operating mode	High-speed operating mode	High-speed operating mode	Low-speed operating mode
Constants	SEL_SYSCLK	CLK_MAIN	CLK_PLL	CLK_HOCO	CLK_SUB
	SEL_PLL	B_NOT_USE	B_USE	B_NOT_USE	B_NOT_USE
	SEL_MAIN	B_USE	B_USE	B_NOT_USE	B_NOT_USE
	SEL_HOCO	B_NOT_USE	B_NOT_USE	B_USE	B_NOT_USE
	SEL_SUB	B_NOT_USE*	B_NOT_USE*	B_NOT_USE*	B_USE
	SEL_OPCM	OPCM_HIGH	OPCM_HIGH	OPCM_HIGH	OPCM_LOW

Note: * When not using the sub-clock for the system clock, clock frequency accuracy measurement circuit (CAC), or the realtime clock (RTC), set the value of the SEL_SUB constant to B_NOT_USE. When using any of the above, refer to Table 1.3.

Table 1.3 Examples of the Sub-Clock and RTC Selections

	Sub-Clock	System Clock*2		RTC	
Sub-Clock Usage	Crystal	Used/ Not Used	Value in SEL_SUB*1	Used/ Not Used	Value in SEL_RTC*1
Not used	None	–	B_NOT_USE	–	B_NOT_USE
System clock	Used	Used	B_USE	Not used	B_NOT_USE
RTC	Used	Not used	B_NOT_USE	Used	B_USE
System clock and RTC	Used	Used	B_USE	Used	B_USE

Notes: 1. When setting B_USE to either or both the SEL_SUB and SEL_RTC constants, the sub-clock oscillates.

2. The sub-clock oscillation is controlled by bits SOSCCR.SOSTP. Therefore the initial setting for the sub-clock differs depending on whether the sub-clock is used as the system clock or not. Also the sub-clock starts oscillating at power-on. Thus processing to stop the sub-clock is performed even when the sub-clock is not used.

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		R5F51403ADFM (RX140 Group) R5F51406BDFN (RX140 Group)
Operating frequencies	When the main clock is selected as the system clock	• Main clock: 8 MHz • System clock (ICLK): 8 MHz (main clock divided by 1) • Peripheral module clock B (PCLKB): 8 MHz (main clock divided by 1) • Peripheral module clock D (PCLKD): 8 MHz (main clock divided by 1) • FlashIF clock (FCLK): 8 MHz (main clock divided by 1)
	When the PLL clock is selected as the system clock	• Main clock: 8 MHz • PLL clock: 48 MHz (main clock divided by 1 and multiplied by 6) • System clock (ICLK): 48 MHz (PLL clock divided by 1) • Peripheral module clock B (PCLKB): 24 MHz (PLL clock divided by 2) • Peripheral module clock D (PCLKD): 48 MHz (PLL clock divided by 1) • FlashIF clock (FCLK): 48 MHz (PLL clock divided by 1)
	When the HOCO clock is selected as the system clock (default setting of sample code)	• HOCO clock: 32 MHz • System clock (ICLK): 32 MHz (HOCO clock divided by 1) • Peripheral module clock B (PCLKB): 32 MHz (HOCO clock divided by 1) • Peripheral module clock D (PCLKD): 32 MHz (HOCO clock divided by 1) • FlashIF clock (FCLK): 32 MHz (HOCO clock divided by 1)
Operating voltage		3.3 V
Integrated development environment		Renesas Electronics Corporation e^2 studio Version 2022-01
C compiler		Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.3.04.00
		Compile options The default setting is used in the integrated development environment.
iodefine.h version		V1.10A
Endian		Little endian
Operating mode		Single-chip mode
Processor mode		Supervisor mode
Sample code version		Version 1.10
Board used		Target Board for RX140 (product No.: RTK5RX1400CxxxxxBJ) Renesas Starter Kit for RX140

3. Reference Application Note

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

RX Family Coding Example of Wait Processing by Software (R01AN1852).

The wait function in the reference application note is used in the sample code accompanying this application note. The revision number of the reference application note is 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. Software

In the sample code, peripheral functions operating after a reset are stopped, nonexistent ports are configured, and then clock settings are configured.

4.1 Stop Processing for Active Peripheral Functions after a Reset

Peripheral functions that are operating after a reset are stopped in this processing.

The module-stop state is canceled after a reset only for modules listed in Table 4.1. To enter the module-stop state, set the module stop bit to 1 (transition to the module-stop state is made). Power consumption can be reduced by entering the module-stop state.

In the sample code, the MSTP_STATE_“target module” constant is set to 0 (MODULE_STOP_DISABLE), so the target module does not enter the module-stop state. When the system requires a module to enter the module-stop state, set the constant in r_init_stop_module.h to 1 (MODULE_STOP_ENABLE).

Table 4.1 lists the peripheral modules whose module-stop states are canceled after a reset.

Table 4.1 Peripheral Modules whose Module-Stop States are Canceled after a Reset

Peripheral Module	Module Stop Bit	Value after a Reset	Value when not Using the Module
DTC	MSTPCRA.MSTPA28 bit	0	1
RAM0	MSTPCRC.MSTPC0 bit	(module-stop state is canceled)	(transition to the module-stop state is made)

4.2 Nonexistent Port Initialization

4.2.1 Overview

Bits corresponding to the nonexistent ports in the PRD register are set to 1. After the nonexistent port initialization function is called, when writing in byte units to the PDR registers or PODR registers which have nonexistent ports, set the I/O select bits for nonexistent ports to 1, and set the output data store bits for nonexistent ports to 0.

Refer to 18.4, Initialization of the Port Direction Register (PDR), in User’s Manual: Hardware when making settings to the PDR registers.

4.2.2 Selecting the Number of Pins and Code Flash Memory Capacity

The number of pins in the sample code is set for the 64-pin package (PIN_SIZE = 64). This application note covers 80-pin, 64-pin, 48-pin and 32-pin packages. When using products other than the 64 pin-package, change PIN_SIZE in r_init_port_initialize.h to the number of pins on the package used.

Furthermore, it is set to use the code flash memory 64 KB product. When using products with the code flash memory of 128 KB or more, change value of CODE_FLASH_MEMORY_SIZE in r_init_port_initialize.h to SIZE_NOT_64KB.

4.3 Clock Settings

4.3.1 Clock Setting Procedure

Table 4.2 lists the clock setting procedure with each processing and setting in the sample code. In the sample code, the HOCO are operating, and the main clock , PLL clock and sub-clock are stopped.

Table 4.2 Clock Setting Procedure

Step	Processing	Details		Setting in the Sample Code
1	Sub-clock setting*1	Not used	The sub-clock control circuit is initialized.	Sub-clock is not used.
		Used	The sub-clock control circuit is initialized and the sub-clock oscillation is enabled. Then wait for the oscillation stabilization time*2 by software is processed.	
2	Main clock setting *1	Not used	No setting is required.	Main clock is not used.
		Used	The main clock drive capability is set, the MOSCWTCR register is set with a wait time until the main clock output is provided to the internal clock, and then the main clock oscillation is enabled. Then wait for the oscillation stabilization time is processed.	
3	PLL clock setting*1	Not used	No setting is required.	PLL clock is not used.
		Used	The PLL input frequency division ratio and frequency multiplication factor are set, and PLL clock oscillation is enabled. Then wait for the oscillation stabilization time is processed.	
4	HOCO clock setting*1	Not used	No setting is required.	HOCO clock is used.
		Used	The HOCO clock oscillation is enabled. Then wait for the oscillation stabilization time is processed.	
5	Operating power control mode setting	The operating power control mode is set according to the operating frequency and operating voltage in the user system.		High-speed operating mode is set.
6	Clock division ratio setting	The clock division ratio is changed.		ICLK, PCLKD, PCLKB, FCLK: No frequency division
7	System clock switching	The system clock is switched according to the user system.		Switched to the HOCO clock.
8	CLKOUT setting*3	Not used	No settings required.	The CLKOUT is not used.
		Used	Selects the clock source output on the CLKOUT pin and sets the clock division ratio. After this, enables output on the CLKOUT pin.	

Notes: 1. When selecting each clock usage, change the appropriate constant in r_init_clock.h as required.
2. Refer to 4.3.2 Sub-Clock Oscillation Stabilization Time for details on the sub-clock oscillation stabilization time.
3. The sample code only makes the CLKOUT settings. To actually output this clock, refer to section 18, I/O Ports, and section 19, Multi-Function Pin Controller (MPC), in RX140 Group User's Manual: Hardware, and make settings appropriate for your system.

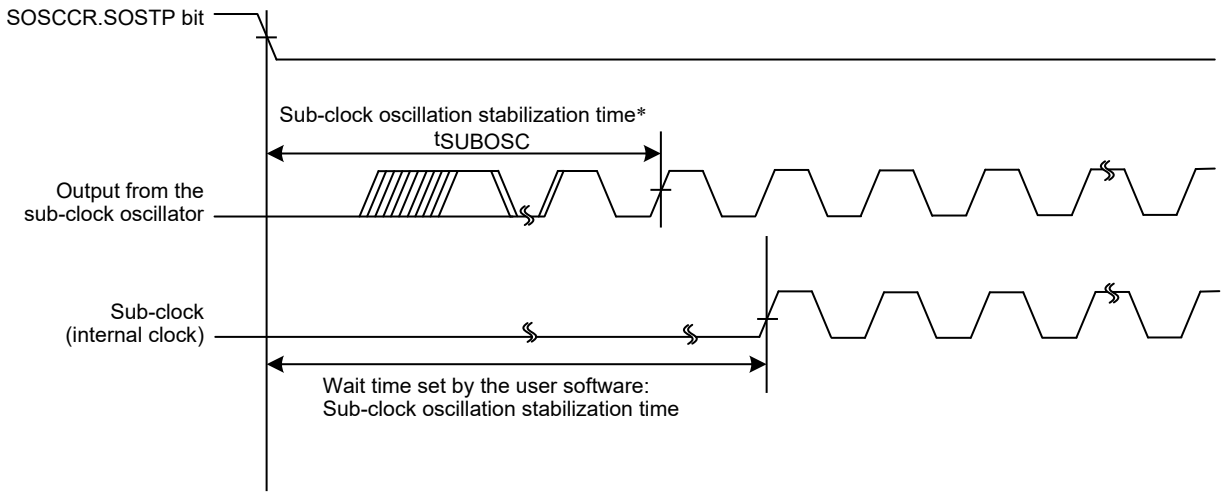
4.3.2 Sub-Clock Oscillation Stabilization Time

This section describes the sub-clock oscillation stabilization time shown in Figure 4.1.

The sub-clock oscillation stabilization time (tSUBOSC) is set to the sub-clock oscillation stabilization time recommended by the crystal/ceramic resonator manufacturer. The wait time by software is set to a value greater than or equal to tSUBOSC.

tSUBOSC used in the sample code is 1.3 seconds, thus the wait time by software is 1.31 seconds here.

Sub-Clock Oscillation Stabilization Time



Note: * Contact the crystal/ceramic resonator manufacturer to determine the oscillation stabilization time of a crystal/ceramic resonator for the user system. The oscillation stabilization time is not a condition for MCU operation, but for a crystal/ceramic resonator to start oscillation.

Figure 4.1 Sub-Clock Oscillation Stabilization Time

4.4 Section Composition

Table 4.3 lists the section data changed in the sample code. For details on adding, changing, and deleting section, refer to the RX Family C/C++ Compiler Package User's Manual.

Table 4.3 Section Data Changed in the Sample Code

Section Name	Change	Address	Function
End_of_RAM	Addition	0000 3FFCh*	Last address of internal RAM

Notes: The capacity of the internal RAM is different every product. Change the address according to the product to be used.

4.5 File Composition

Table 4.4 lists the files used in the sample code. Files generated by the integrated development environment should not be listed in this table.

Table 4.4 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_port_initialize.c	Nonexistent port initialization	
r_init_port_initialize.h	Header file for r_init_port_initialize.c	
r_init_clock.c	Clock initialization	
r_init_clock.h	Header file for r_init_clock.c	
r_delay.c	Wait processing by software	
r_delay.h	Header file for r_delay.c	

4.6 Option-Setting Memory

Table 4.5 lists the option-setting memory configured in the sample code. When necessary, set a value suited to the user system.

Table 4.5 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.
OFS1	FFFF FF8Bh to FFFF FF88h	FFFF FFFFh	Fast startup time at power-on is disabled. The voltage monitor 0 reset is disabled after a reset. HOCO oscillation is disabled after a reset. The frequency of HOCO is 32 MHz.
MDE	FFFF FF83h to FFFF FF80h	FFFF FFFFh	Little endian

4.7 Constants

Table 4.6 and Table 4.7 list the constants used in the sample code, which can be changed by users. Table 4.8 lists the constants used in the sample code, which cannot be changed by users. Table 4.9 lists the constants when a 80-pin package is used (PIN_SIZE = 80), Table 4.10 lists the constants when a 64-pin package is used (PIN_SIZE = 64), Table 4.11 lists the constants when a 48-pin package is used (PIN_SIZE = 48), Table 4.12 lists the constants when a 32-pin package is used (PIN_SIZE = 32).

Table 4.6 Constants Used in the Sample Code (1/2)
(Users can change the constants listed in this table.)

Constant Name	Setting Value	Contents
SEL_MAIN*1	B_NOT_USE	Selection of the main clock operation: B_USE: Used (main clock oscillating) B_NOT_USE: Not used (main clock stopped)
MAIN_CLOCK_HZ*1	8,000,000L	Oscillation frequency of a crystal/ceramic resonator for the main clock (Hz)
REG_MOFCR*1	00h	Setting for the drive capability of the main clock oscillator (setting value in the MOFCR register)
REG_MOSCWTCR*1	06h	Setting value in the main clock wait control register
SEL_HOCO*1	B_USE	Selection of the HOCO clock operation: B_USE: Used (HOCO clock oscillating) B_NOT_USE: Not used (HOCO clock stopped)
SEL_PLL*1	B_NOT_USE	Selection of the PLL clock operation: B_USE: Used (PLL clock oscillating) B_NOT_USE: Not used (PLL clock stopped)
REG_PLLCR*1	1701h	PLL input frequency division ratio and frequency multiplication factor settings (setting value in the PLLCR register)
SEL_SUB*1*2	B_NOT_USE	Selection of the sub-clock usage for the system clock: B_USE: Used B_NOT_USE: Not used
SEL_RTC*1*2	B_NOT_USE	Selection of the sub-clock usage for the RTC count source: B_USE: Used B_NOT_USE: Not used
SUB_CLOCK_HZ*1	32,768 L	Oscillation frequency of a crystal for the sub-clock (Hz)
WAIT_TIME_FOR_SUB_OSCILLATION*1	1,310,000,000L	Sub-clock oscillation stabilization time (ns)
REG_SOMCR*1	CL_LOW	Selection of the sub-clock oscillator drive capability: CL_LOW: Low-drive output for the low CL CL_MID: Middle-drive output for the low CL CL_HIGH: High-drive output for the low CL CL_STD: Drive capacity for standard CL

Notes: 1. Change the setting value in r_init_clock.h according to the user system.
2. The sub-clock operation is set to be oscillating by setting B_USE (sub-clock used) to either of the SEL_SUB constant or SEL_RTC constant, or both.

Table 4.7 Constants Used in the Sample Code (2/2)
(Users can change the constants listed in this table.)

Constant Name	Setting Value	Contents
SEL_CKOUT*1	CKOUT_NOT_USE	CKOCR setting values CKOUT_USE: CLKOUT pin output enabled CKOUT_NOT_USE: CLKOUT pin output disabled (fixed low)
CKO_CLK*1	CKO_LOCO	Clock output source selection CKO_LOCO: LOCO CKO_HOCO: HOCO CKO_MAIN: main clock CKO_SUB: sub-clock CKO_PLL: PLL CKO_CTSU: CTSU internal clock*6
CKO_DIV*1	0h	Clock output divisor selection 0:1/1 1:1/2 2:1/4 3:1/8 4:1/16
SEL_CNTMD*1	CNTMD_CAL	Selection of the real-time clock count mode CNTMD_CAL: Calendar count mode CNTMD_BIN: Binary count mode
SEL_SYSCLK*1	CLK_HOCO	Clock source selection for the system clock CLK_HOCO: HOCO clock CLK_MAIN: Main clock CLK_SUB: Sub-clock CLK_PLL: PLL clock
REG_SCKCR*1	0000 0000h	Setting for the internal clock division ratio (setting value in the SCKCR register)
SEL_OPCM*1	OPCM_HIGH	Selection of the operating power control mode*5 OPCM_HIGH: High-speed operating mode OPCM_MID: Middle-speed operating mode OPCM_MID2: Middle-speed operating mode2 OPCM_LOW: Low-speed operating mode*4
REG_MEMWAIT*1	MEMWAIT_OFF	Selection of whether the memory wait cycles are set MEMWAIT_ON: Wait cycles set MEMWAIT_OFF: No wait cycles
MSTP_STATE_DTC*2	MODULE_STOP_DISABLE	Selection of the module-stop state for DTC MODULE_STOP_DISABLE: Module-stop state canceled MODULE_STOP_ENABLE: Entering the module-stop state
MSTP_STATE_RAM0*2	MODULE_STOP_DISABLE	Selection of the module-stop state for RAM0 MODULE_STOP_DISABLE: Operating MODULE_STOP_ENABLE: Stopped
PIN_SIZE*3	64	Number of pins on the product used
CODE_FLASH_MEMORY_SIZE	SIZE_64KB	Select a product with code flash memory of 64 KB or 128 KB or more SIZE_64KB: Use 64 KB product SIZE_NOT_64KB: Use 128KB or more product

Notes: 1. Change the setting value in r_init_clock.h according to the user system.
2. Change the setting value in r_init_stop_module.h according to the user system.
3. Change the setting value in r_init_port_initialize.h according to the user system.
4. Low-speed operating mode can be selected only when the sub-clock is used as the system clock.
5. The ranges of the operating frequency and operating voltage differ depending on operating modes. Refer to the User's Manual: Hardware for details.
6. CTSU internal clock output setting is required separately. Refer to the User's Manual: Hardware for details.

Table 4.8 Constants Used in the Sample Code
(Users cannot change the constants listed in this table.)

Constant Name	Setting Value	Contents
B_NOT_USE	0	Not used
B_USE	1	Used
CL_LOW	03h	Low-drive output for the low CL
CL_MID	02h	Middle-drive output for the low CL
CL_HIGH	01h	High-drive output for the low CL
CL_STD	00h	Drive capacity for standard CL
CNTMD_CAL	0	RTC: Calendar count mode
CNTMD_BIN	1	RTC: Binary count mode
CLK_HOCO	0100h	Clock source: HOCO clock
CLK_MAIN	0200h	Clock source: Main clock
CLK_SUB	0300h	Clock source: Sub-clock
CLK_PLL	0400h	Clock source: PLL clock
CKOUT_USE	0	CLKOUT used LOCO selected, division ratio x 1/1, CLKOUT pin output enabled
CKOUT_NOT_USE	1	CLKOUT not used LOCO selected, division ratio x 1/1, CLKOUT pin output disabled
CKO_LOCO	0h	CLKOUT clock source: LOCO
CKO_HOCO	1h	CLKOUT clock source: HOCO
CKO_MAIN	2h	CLKOUT clock source: MAIN
CKO_SUB	3h	CLKOUT clock source: SUB
CKO_PLL	4h	CLKOUT clock source: PLL
CKO_CTSU	8h	CLKOUT clock source: CTSU internal clock
SUB_CLOCK_CYCLE	(1000000L/SUB_CLOCK_HZ)	Sub-clock cycle (μs)
LOCO_CLOCK_KHZ	(4560L)	LOCO clock frequency (kHz)
FOR_CMT0_TIME	(7018*8)	Counter cycle (ns) of the sub-clock oscillation stabilization wait timer (CMT0) (LOCO = 4.56 MHz (max.) × 1/8, PCLK × 1/32)
OPCM_HIGH	00h	Operating power control mode: High-speed operating mode
OPCM_MID	02h	Operating power control mode: Middle-speed operating mode
OPCM_MID2	04h	Operating power control mode: Middle-speed operating mode2
OPCM_LOW	FFh	Operating power control mode: Low-speed operating mode
OPCM_DEFAULT	OPCM_MID	Operating mode after reset cancellation
MEMWAIT_ON	0001h	Memory wait cycles are set
MEMWAIT_OFF	0000h	No memory wait cycles
MODULE_STOP_ENABLE	1	Transition to the module stop-state is made
MODULE_STOP_DISABLE	0	Module stop-state is canceled
SIZE_64KB	1	Use 64 KB product
SIZE_NOT_64KB	0	Use 128KB or more product

Table 4.9 Constants when a 80-Pin Package is Used (PIN_SIZE = 80)

Constant Name	Setting Value	Contents
DEF_P0PDR	07h	Setting value for the port P0 direction register
DEF_P1PDR	03h	Setting value for the port P1 direction register
DEF_P2PDR	3Ch	Setting value for the port P2 direction register
DEF_P3PDR	08h	Setting value for the port P3 direction register
DEF_P4PDR	00h	Setting value for the port P4 direction register
DEF_P5PDR	CFh	Setting value for the port P5 direction register
DEF_PAPDR	80h	Setting value for the port PA direction register
DEF_PBPDR	00h	Setting value for the port PB direction register
DEF_PCPDR	03h	Setting value for the port PC direction register
DEF_PDPDR	F8h	Setting value for the port PD direction register
DEF_PEPDR	C0h	Setting value for the port PE direction register
DEF_PGPDR	7Fh	Setting value for the port PG direction register
DEF_PHPDR	30h	Setting value for the port PH direction register
DEF_PJPDR	3Dh	Setting value for the port PJ direction register

Table 4.10 Constants when a 64-Pin Package is Used (PIN_SIZE = 64)

Constant Name	Setting Value	Contents
DEF_P0PDR	D7h	Setting value for the port P0 direction register
DEF_P1PDR	0Fh	Setting value for the port P1 direction register
DEF_P2PDR	3Fh	Setting value for the port P2 direction register
DEF_P3PDR	18h	Setting value for the port P3 direction register
DEF_P4PDR	00h	Setting value for the port P4 direction register
DEF_P5PDR	CFh	Setting value for the port P5 direction register
DEF_PAPDR	A4h	Setting value for the port PA direction register
DEF_PBPDR	14h	Setting value for the port PB direction register
DEF_PCPDR	03h	Setting value for the port PC direction register
DEF_PDPDR	FFh	Setting value for the port PD direction register
DEF_PEPDR	C0h	Setting value for the port PE direction register
DEF_PEPDR	7Fh	Setting value for the port PG direction register
DEF_PHPDR*1	30h (128KB or more product) F0h (64 KB product)	Setting value for the port PH direction register
DEF_PJPDR	3Fh	Setting value for the port PJ direction register

Table 4.11 Constants when a 48-Pin Package is Used (PIN_SIZE = 48)

Constant Name	Setting Value	Contents
DEF_P0PDR	FFh	Setting value for the port P0 direction register
DEF_P1PDR	0Fh	Setting value for the port P1 direction register
DEF_P2PDR	3Fh	Setting value for the port P2 direction register
DEF_P3PDR	1Ch	Setting value for the port P3 direction register
DEF_P4PDR	18h	Setting value for the port P4 direction register
DEF_P5PDR	FFh	Setting value for the port P5 direction register
DEF_PAPDR	A5h	Setting value for the port PA direction register
DEF_PBPDR	D4h	Setting value for the port PB direction register
DEF_PCPDR	0Fh	Setting value for the port PC direction register
DEF_PDPDR	FFh	Setting value for the port PD direction register
DEF_PEPDR	E1h	Setting value for the port PE direction register
DEF_PGPDR	7Fh	Setting value for the port PG direction register
DEF_PHPDR	F0h	Setting value for the port PH direction register
DEF_PJPDR	3Fh	Setting value for the port PJ direction register

Table 4.12 Constants when a 32-Pin Package is Used (PIN_SIZE = 32)

Constant Name	Setting Value	Contents
DEF_P0PDR	FFh	Setting value for the port P0 direction register
DEF_P1PDR	3Fh	Setting value for the port P1 direction register
DEF_P2PDR	3Fh	Setting value for the port P2 direction register
DEF_P3PDR	9Ch	Setting value for the port P3 direction register
DEF_P4PDR	F8h	Setting value for the port P4 direction register
DEF_P5PDR	FFh	Setting value for the port P5 direction register
DEF_PAPDR	E5h	Setting value for the port PA direction register
DEF_PBPDR	FEh	Setting value for the port PB direction register
DEF_PCPDR	0Fh	Setting value for the port PC direction register
DEF_PDPDR	FFh	Setting value for the port PD direction register
DEF_PEPDR	E1h	Setting value for the port PE direction register
DEF_PGPDR	7Fh	Setting value for the port PG direction register
DEF_PHPDR	FFh	Setting value for the port PH direction register
DEF_PJPDR	3Fh	Setting value for the port PJ direction register

4.8 Function

Table 4.13 lists the functions used in the sample code.

Table 4.13 Functions Used in the Sample Code

Function Name	Outline
main	Main processing
R_INIT_StopModule	Stop processing for active peripheral functions after a reset
R_INIT_Port_Initialize	Nonexistent port initialization
R_INIT_Clock	Clock initialization
cgc_oscillation_main	Main clock oscillation setting
cgc_oscillation_hoco	HOCO clock oscillation setting
cgc_oscillation_pll	PLL clock oscillation setting
cgc_oscillation_sub	Sub-clock oscillation setting
cgc_disable_subclk	Sub-clock stop setting
oscillation_subclk	Enabling sub-clock oscillation
init_rtc	Initialization RTC
no_use_subclk_as_sysclk	Setting when the sub-clock is not used as the system clock
cmt0_countstart	CMT0 wait start setting (wait for sub-clock oscillation stabilization)
cmt0_endcheck	CMT0 wait (wait for sub-clock oscillation stabilization) completion check and initialization
enable_clkout	CKOUT settings
R_DELAY	Inline function to specify the number of loops
R_DELAY_Us	Function to specify the execution time

4.9 Function Specifications

The following tables list the sample code function specifications.

main	
Outline	Main processing
Header	None
Declaration	void main(void)
Description	Calls the following functions: Stop processing for active peripheral functions after a reset, nonexistent port initialization, and clock initialization.
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.

R_INIT_Port_Initialize	
Outline	Nonexistent port initialization
Header	r_init_port_initialize.h
Declaration	void R_INIT_Port_Initialize(void)
Description	Initializes port direction registers for ports that do not exist in products.
Arguments	None
Return Value	None
Remarks	The number of pins in the sample code is set for the 64-pin (PIN_SIZE=64) package and the 64 KB(CODE_FLASH_MEMORY_SIZE=SIZE_64KB) product. After this function is called, when writing in byte units to the PDR registers or PODR registers which have nonexistent ports, set the I/O select bits for nonexistent ports in the PDR registers to 1, and set the output data store bits for nonexistent ports in the PODR registers to 0. For products with code flash memory of 64 KB or less, "00h" can be read even if the value is set in the PDR register of PORTD. For details, refer to the user's manual hardware edition.

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 which uses the HOCO clock as the system clock without using the PLL clock, the sub-clock, and RTC.

cgc_oscillation_main	
Outline	Main clock oscillation setting
Header	r_init_clock.h
Declaration	static void cgc_oscillation_main(void)
Description	Sets the main clock drive capability, sets the MOSCWTCR register, and enables main clock oscillation. Then waits for the main clock oscillation stabilization time.
Arguments	None
Return Value	None

cgc_oscillation_hoco	
Outline	HOCO clock oscillation setting
Header	r_init_clock.h
Declaration	static void cgc_oscillation_hoco(void)
Description	Enables the HOCO clock oscillation. Then waits for the HOCO clock oscillation stabilization time.
Arguments	None
Return Value	None

cgc_oscillation_pll	
Outline	PLL clock oscillation setting
Header	r_init_clock.h
Declaration	static void cgc_oscillation_pll(void)
Description	Sets the PLL input frequency division ratio and frequency multiplication factor, and enables PLL clock oscillation. Then waits for the PLL clock oscillation stabilization time.
Arguments	None
Return Value	None

cgc_oscillation_sub	
Outline	Sub-clock oscillation setting
Header	r_init_clock.h
Declaration	static void cgc_oscillation_sub(void)
Description	Configures the setting when the sub-clock is used as either the system clock or the RTC count source, or both.
Arguments	None
Return Value	None

cgc_disable_subclk	
Outline	Sub-clock stop setting
Header	r_init_clock.h
Declaration	static void cgc_disable_subclk(void)
Description	Makes settings for when the sub-clock is not used as both the system clock and the RTC count source.
Arguments	None
Return Value	None

oscillation_subclk	
Outline	Enabling the sub-clock oscillation
Header	None
Declaration	static void oscillation_subclk(void)
Description	Configures settings for sub-clock oscillation.
Arguments	None
Return Value	None

init_rtc	
Outline	Initialization RTC
Header	None
Declaration	static void init_rtc(void)
Description	Initializes the RTC. (setting for clock provision and RTC software reset)
Arguments	None
Return Value	None

no_use_subclk_as_sysclk	
Outline	Processing when the sub-clock is not used as the system clock
Header	None
Declaration	static void no_use_subclk_as_sysclk(void)
Description	Stops the sub-clock as the system clock when the sub-clock is used only as the RTC count source.
Arguments	None
Return Value	None

cmt0_countstart	
Outline	CMT0 wait start setting (wait for sub-clock oscillation stabilization)
Header	None
Declaration	static void cmt0_countstart(uint16_t cnt)
Description	When using the sub-clock oscillator, waits for the sub-clock oscillation stabilization time with CMT0. When starting to wait for the oscillation stabilization, CMT0 count starts.
Arguments	uint16_t cnt: Oscillation stabilization time cnt = oscillation stabilization time (ns)*1 ÷ FOR_CMT0_TIME*2
Return Value	None
Remarks	1. The oscillation stabilization time varies depending on the crystal/ceramic resonator. Set the value referring to 4.3.2 Sub-Clock Oscillation Stabilization Time. 2. The value of FOR_CMT0_TIME is calculated with 4.56 MHz (max.) of LOCO. The actual wait time may differ depending on the LOCO clock frequency.

cmt0_endcheck	
Outline	CMT0 wait (wait for sub-clock oscillation stabilization) completion check and initialization
Header	None
Declaration	static void cmt0_endcheck(void)
Description	When using the sub-clock oscillator, checks whether the wait processing for the sub-clock oscillation stabilization is completed. If completed, initializes CMT0.
Arguments	None
Return Value	None

enable_clkout	
Outline	CLKOUT settings.
Header	None
Declaration	static void enable_clkout (void)
Description	Configure the setting when the CLKOUT is used.
Arguments	None
Return Value	None

R_DELAY	
Outline	Inline function to specify the number of loops
Header	r_delay.h
Declaration	static void R_DELAY(unsigned long loop_cnt)
Description	Wait processing which performs loops for the specified number of times (a loop is fixed at 5 cycles).
Arguments	loop_cnt: The number of loops
Return Value	None
Remarks	For details on this function, refer to the application note RX Family: Coding Example of Wait Processing by Software, rev. 1.00.

R_DELAY_Us	
Outline	Function to specify the execution time
Header	r_delay.h
Declaration	void R_DELAY_Us(unsigned long us, unsigned long kHz)
Description	Calculates the number of loops based on the execution time (μs) and the system clock (ICLK) frequency, and calls the inline function to specify the number of loops.
Arguments	us: Execution time kHz: System clock (ICLK) frequency when the function is called.
Return Value	None
Remarks	For details on this function, refer to the application note RX Family: Coding Example of Wait Processing by Software, rev. 1.00.

4.10 Flowcharts

4.10.1 Main Processing

Figure 4.2 shows the main processing.

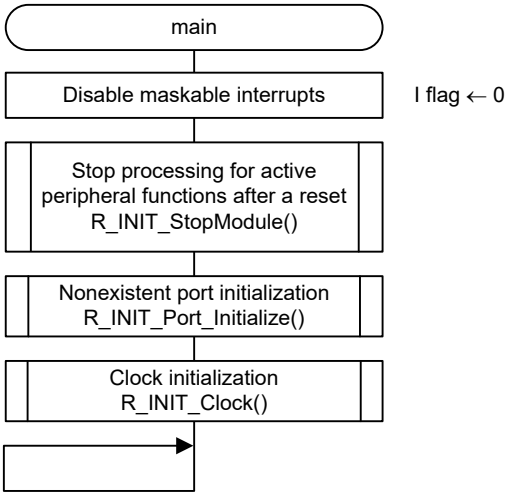
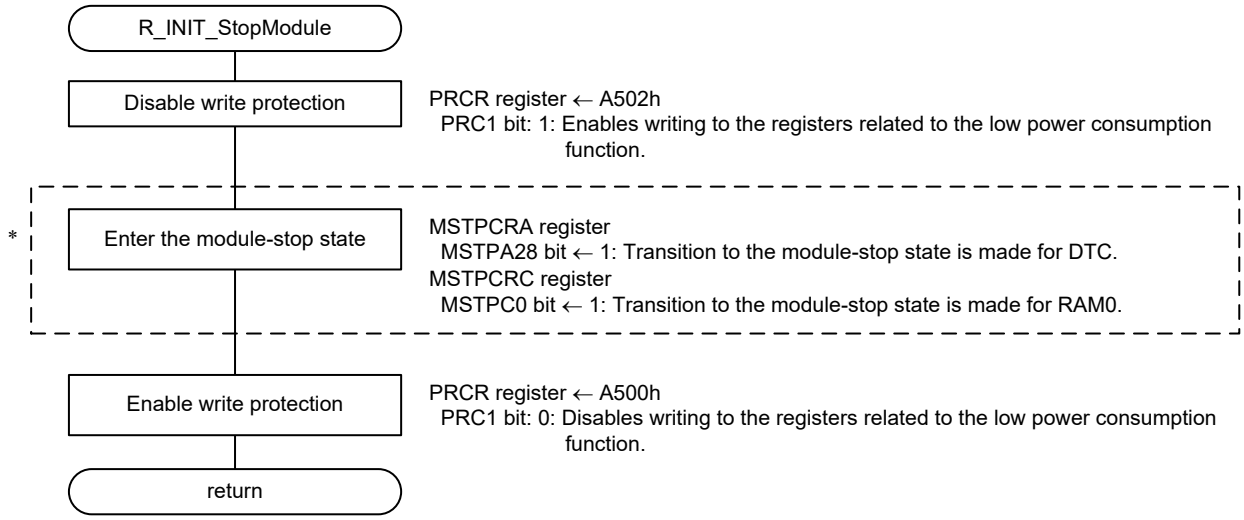


Figure 4.2 Main Processing

4.10.2 Stop Processing for Active Peripheral Functions after Reset

Figure 4.3 shows stop processing for active peripheral functions after a reset.



Note: * The module-stop state is canceled in the sample code. When entering the module-stop state for any peripheral functions, set the #define MSTP_STATE_"target module" constant to 1

Figure 4.3 Stop Processing for Active Peripheral Function after a Reset

4.10.3 Nonexistent Port Initialization

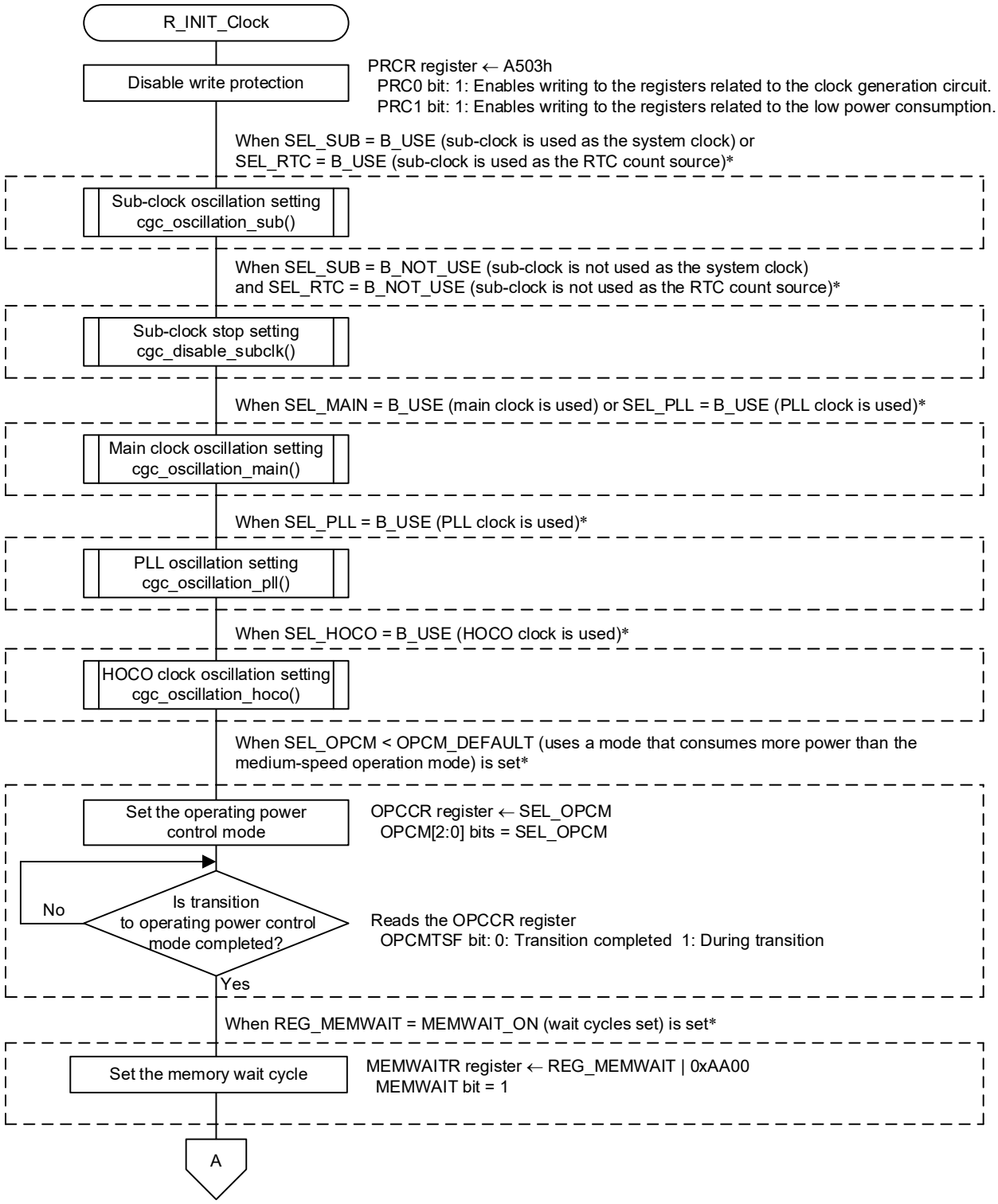
Figure 4.4 shows the nonexistent port initialization.

Note: * For products with code flash memory of 64 KB or less, "00h" can be read even if the value is set in the PDR register of PORTD. For details, refer to the user's manual hardware edition.

Figure 4.4 Nonexistent Port Initialization

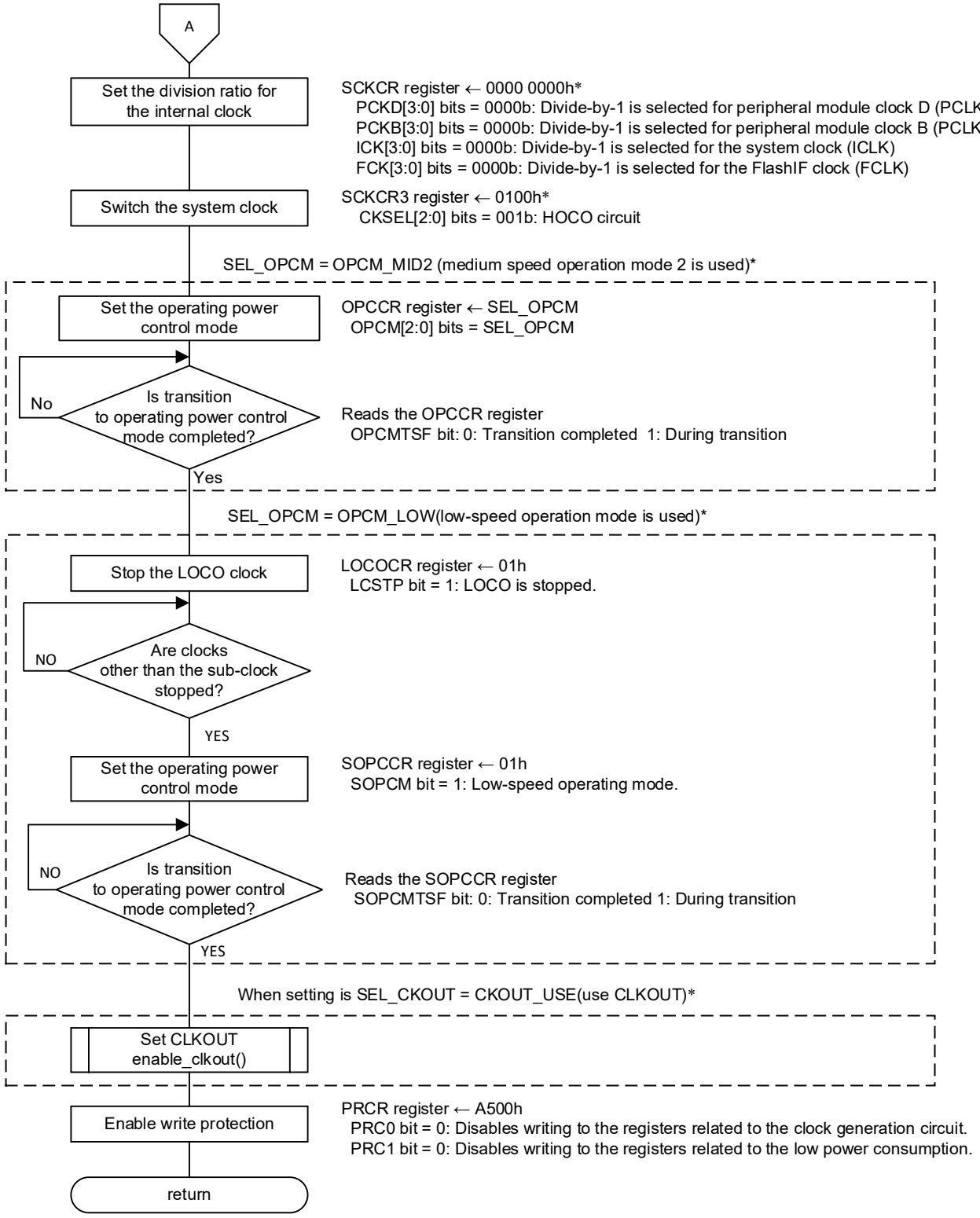
4.10.4 Clock Initialization

Figure 4.5 and Figure 4.6 show the clock initialization.



Note: * Change the constant value according to the user system.

Figure 4.5 Clock Initialization (1/2)

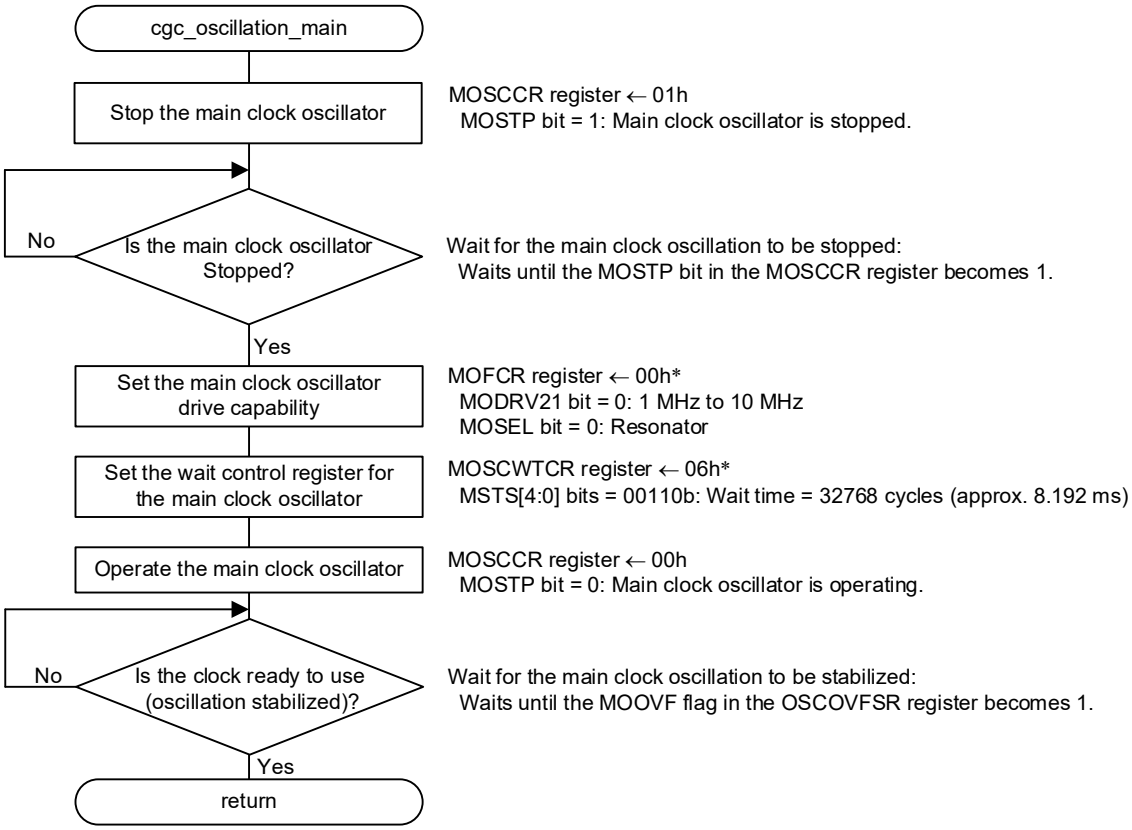


Note: * Change the constant value according to the user system.

Figure 4.6 Clock Initialization (2/2)

4.10.5 Main Clock Oscillation Setting

Figure 4.7 shows the main clock oscillation setting.

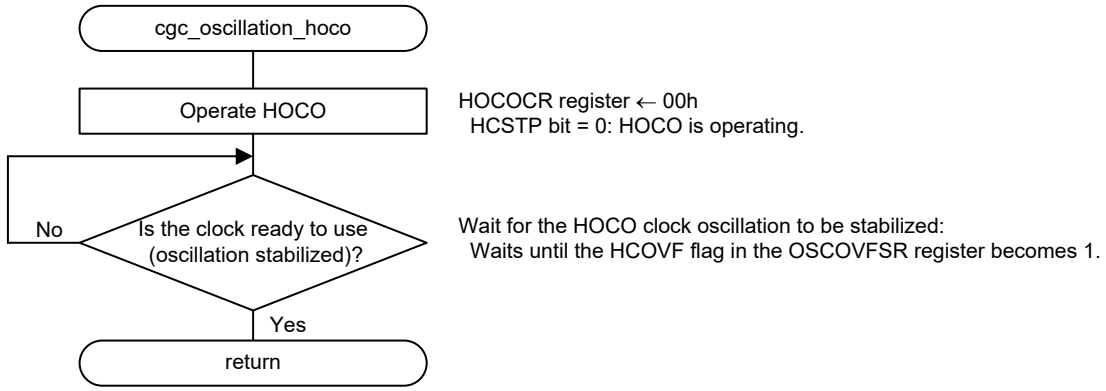


Note: * Change the constant value according to the user system.

Figure 4.7 Main Clock Oscillation Setting

4.10.6 HOCO Clock Oscillation Setting

Figure 4.8 shows the HOCO clock oscillation setting.

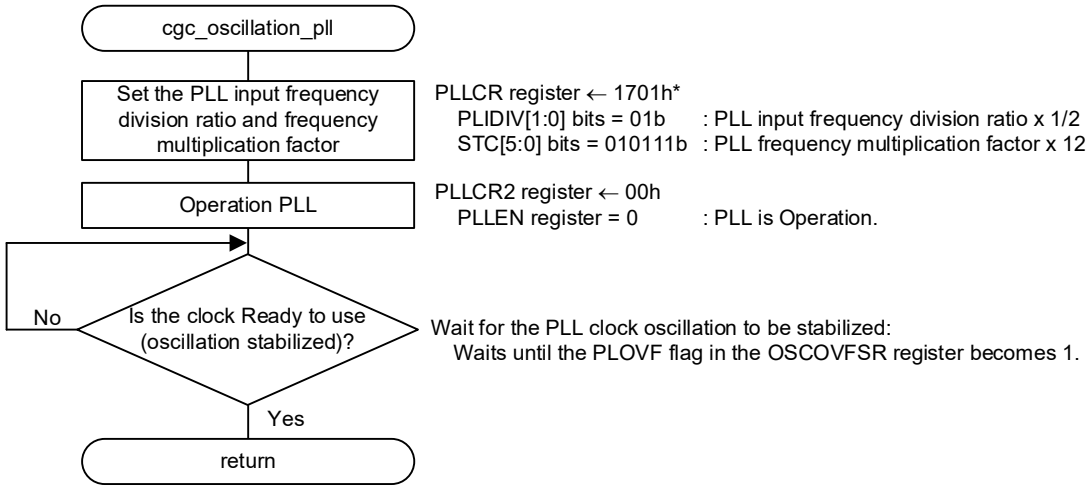


Note: * Change the constant value according to the user system.

Figure 4.8 HOCO Clock Oscillation Setting

4.10.7 PLL Clock Oscillation Setting

Figure 4.9 shows the PLL clock oscillation setting.

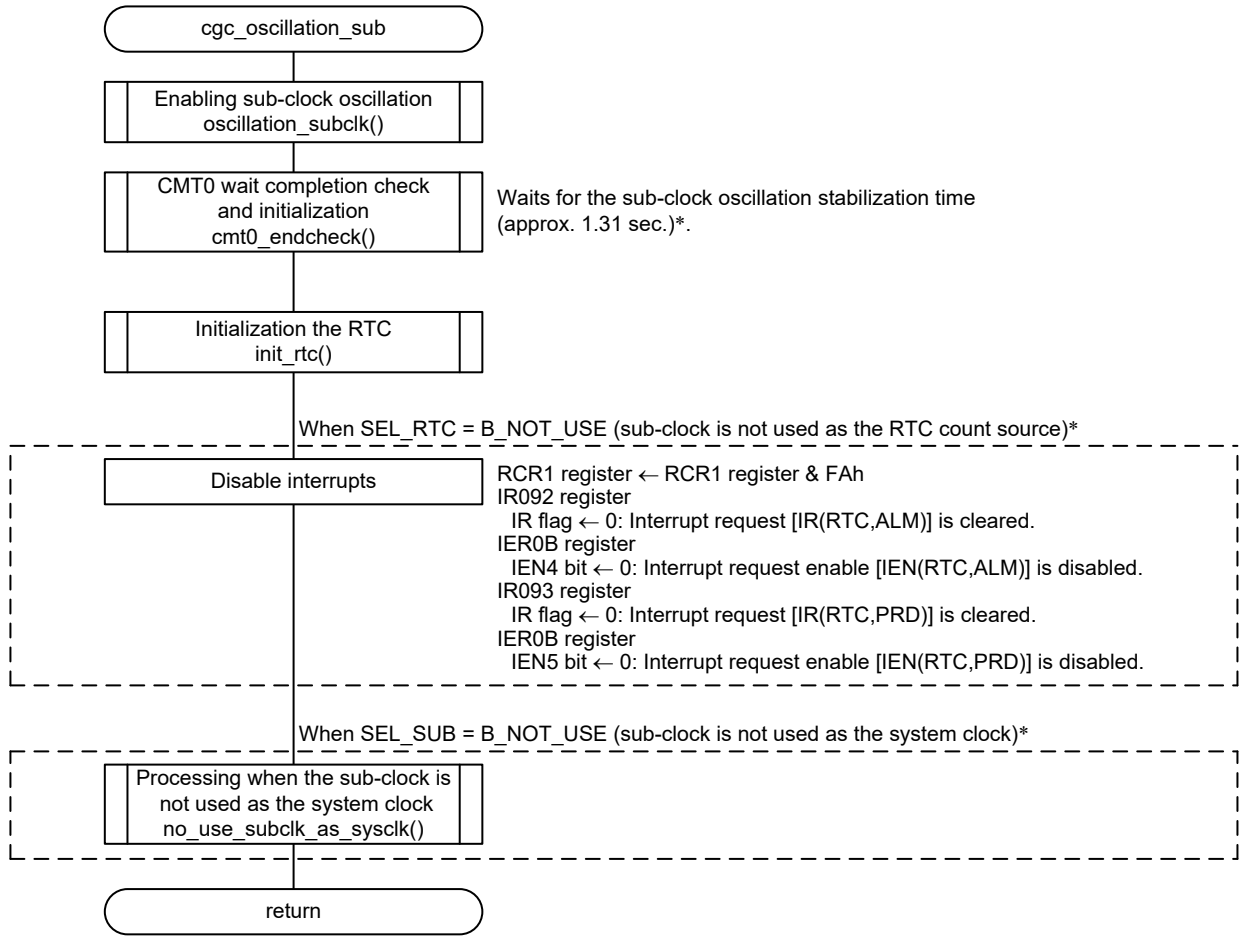


Note: * Change the constant value according to the user system.

Figure 4.9 PLL Clock Oscillation Setting

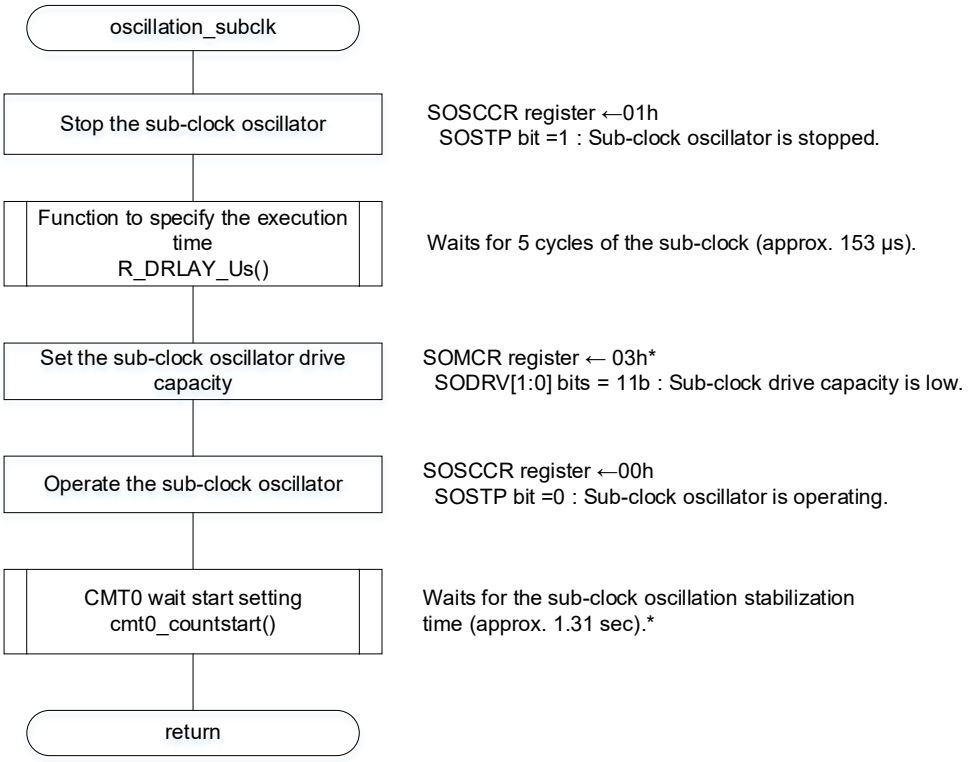
4.10.8 Sub-Clock Oscillation Setting

Figure 4.10 to Figure 4.13 show the sub-clock oscillation setting.



Note: * Change the constant value according to the user system.

Figure 4.10 Sub-Clock Oscillation Setting



Note: * Change the constant value according to the user system.

Figure 4.11 Enabling Sub-Clock Oscillation

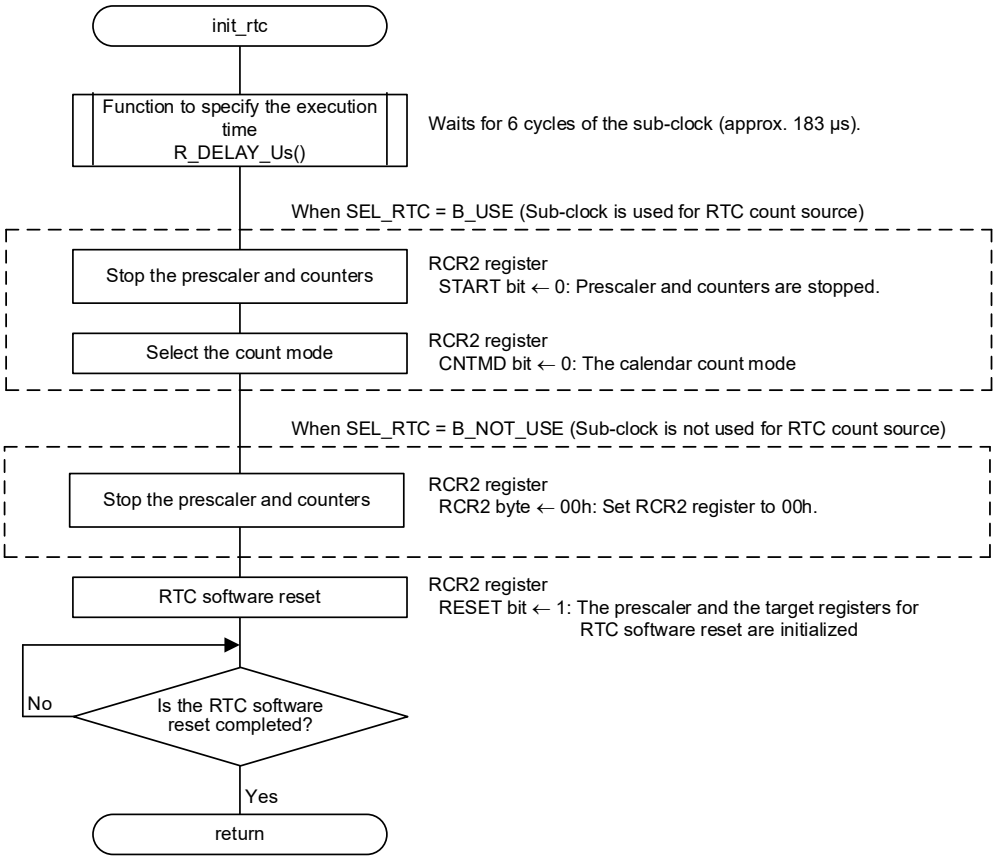


Figure 4.12 Initialization when Using the RTC

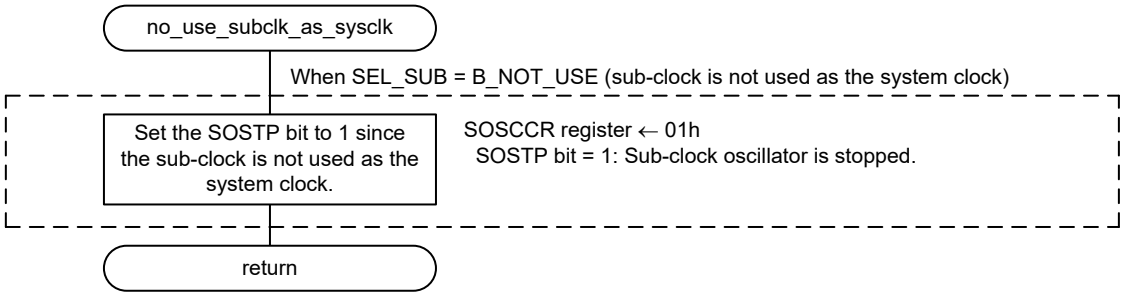


Figure 4.13 Processing when the Sub-Clock is not Used as the System Clock

4.10.9 Sub-Clock Stop Setting

Figure 4.14 shows the sub-clock stop setting.

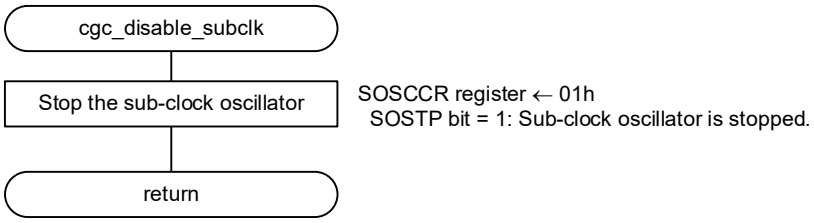


Figure 4.14 Sub-Clock Stop Setting

4.10.10 CMT0 Wait Start Setting, and CMT0 Wait Completion Check and Initialization

Figure 4.15 and Figure 4.16 show the CMT0 wait start setting, and cmt0 wait completion check and initialization.

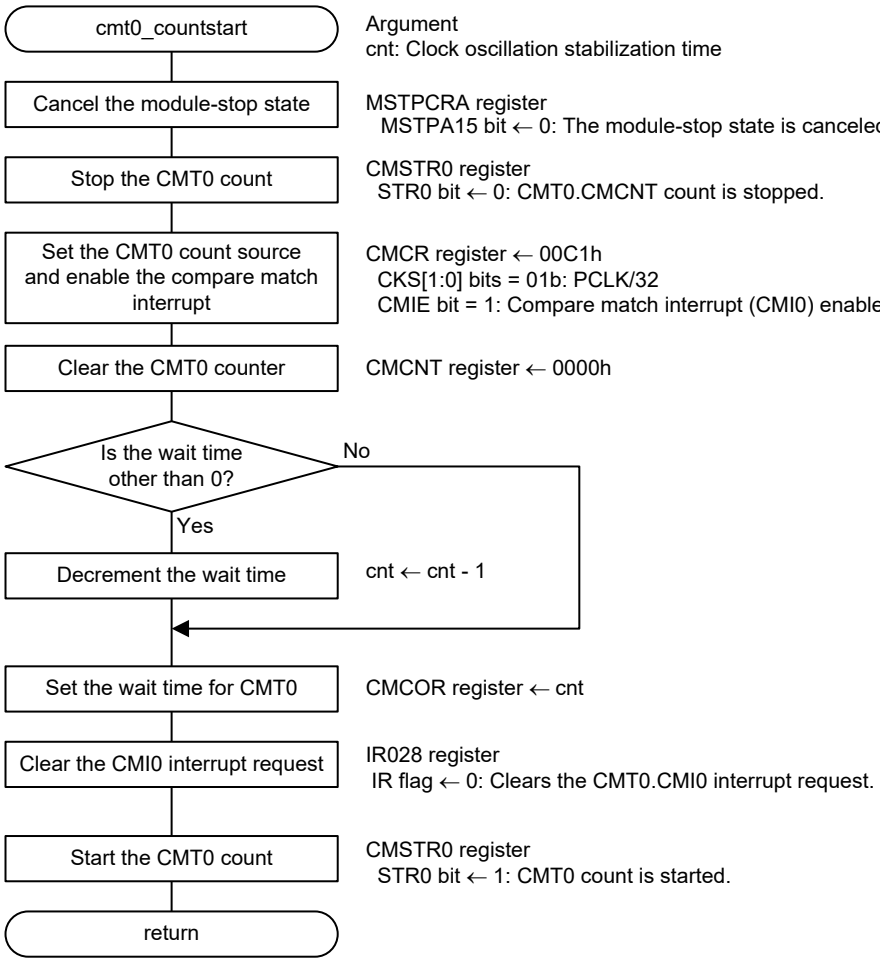
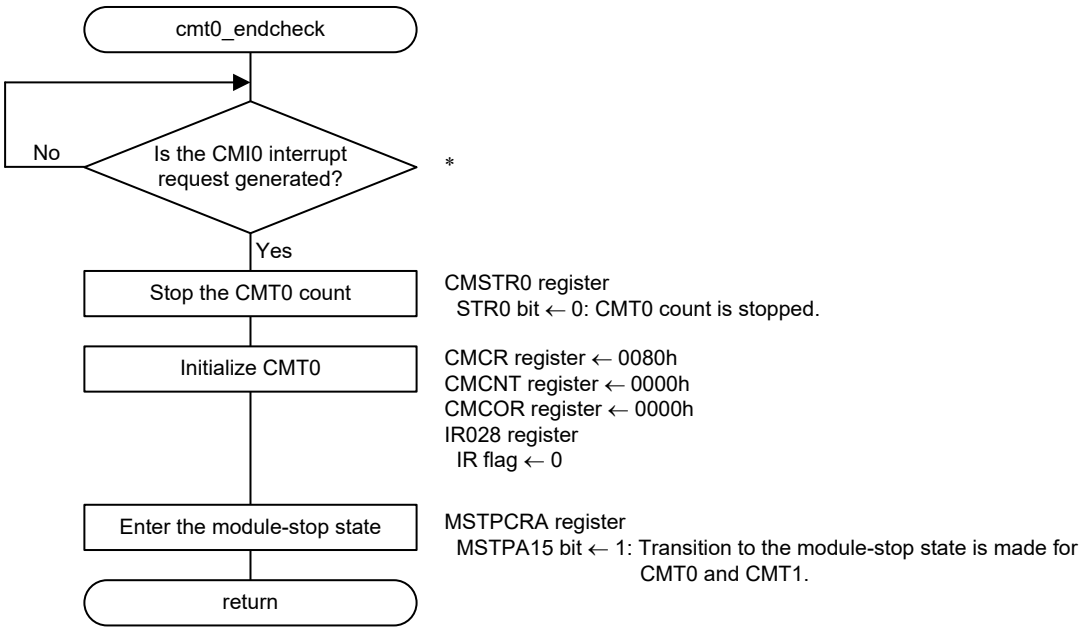


Figure 4.15 CMT0 Wait Start Setting

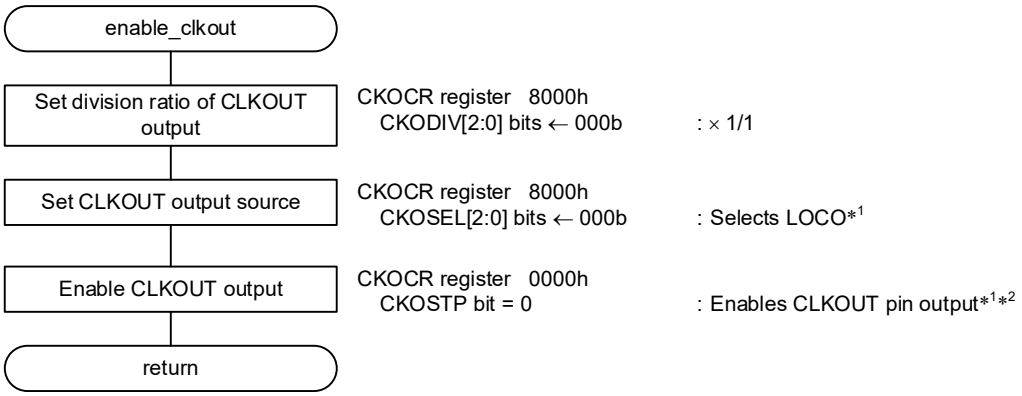


Note: * When the counter of the independent watchdog timer (IWDT) is operating, refresh the counter in this loop as required.

Figure 4.16 CMT0 Wait Completion Check and Initialization

4.10.11 CLKOUT Settings

Figure 4.17 is a flowchart of the processing for making CLKOUT settings.



Notes: 1. The initial settings example only makes operation settings. To actually output this clock, it is necessary to also make settings to the pin function control register and port mode register of the corresponding pin. Refer to section 18, I/O Ports, section 19, Multi-Function Pin Controller (MPC) , and section 32, Capacitive Touch Sensing Unit (CTSU2SL, CTSU2L), in RX140 Group User's Manual: Hardware, and make settings appropriate for your system.

2. Overwriting CKOSTP while the clock is oscillating may cause glitches in the output.

Figure 4.17 CLKOUT Settings

5. Importing a Project

5.1 Importing a Project in the 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.

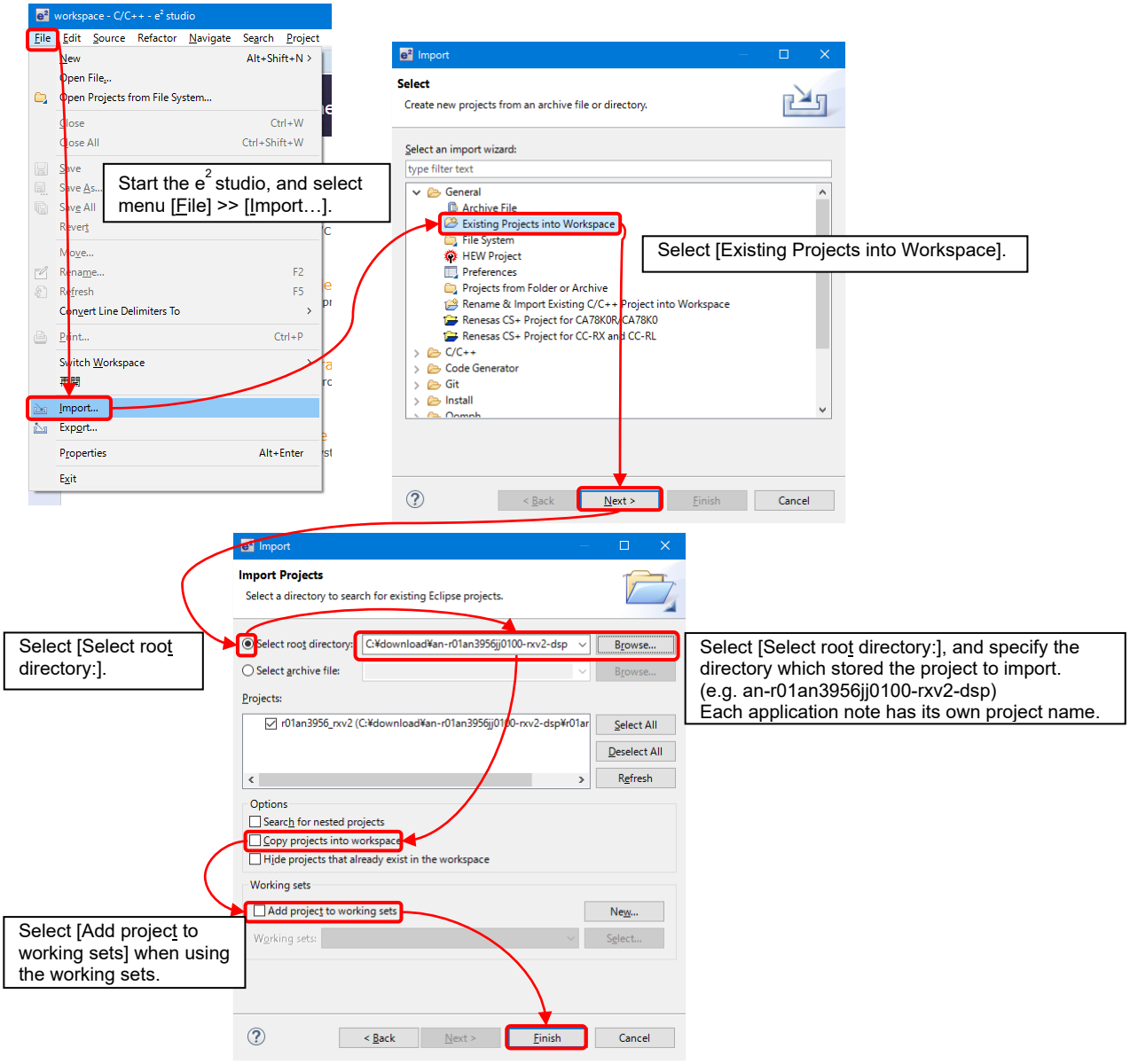


Figure 5.1 Importing a Project in the e² studio

5.2 Importing a Project in CS+

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

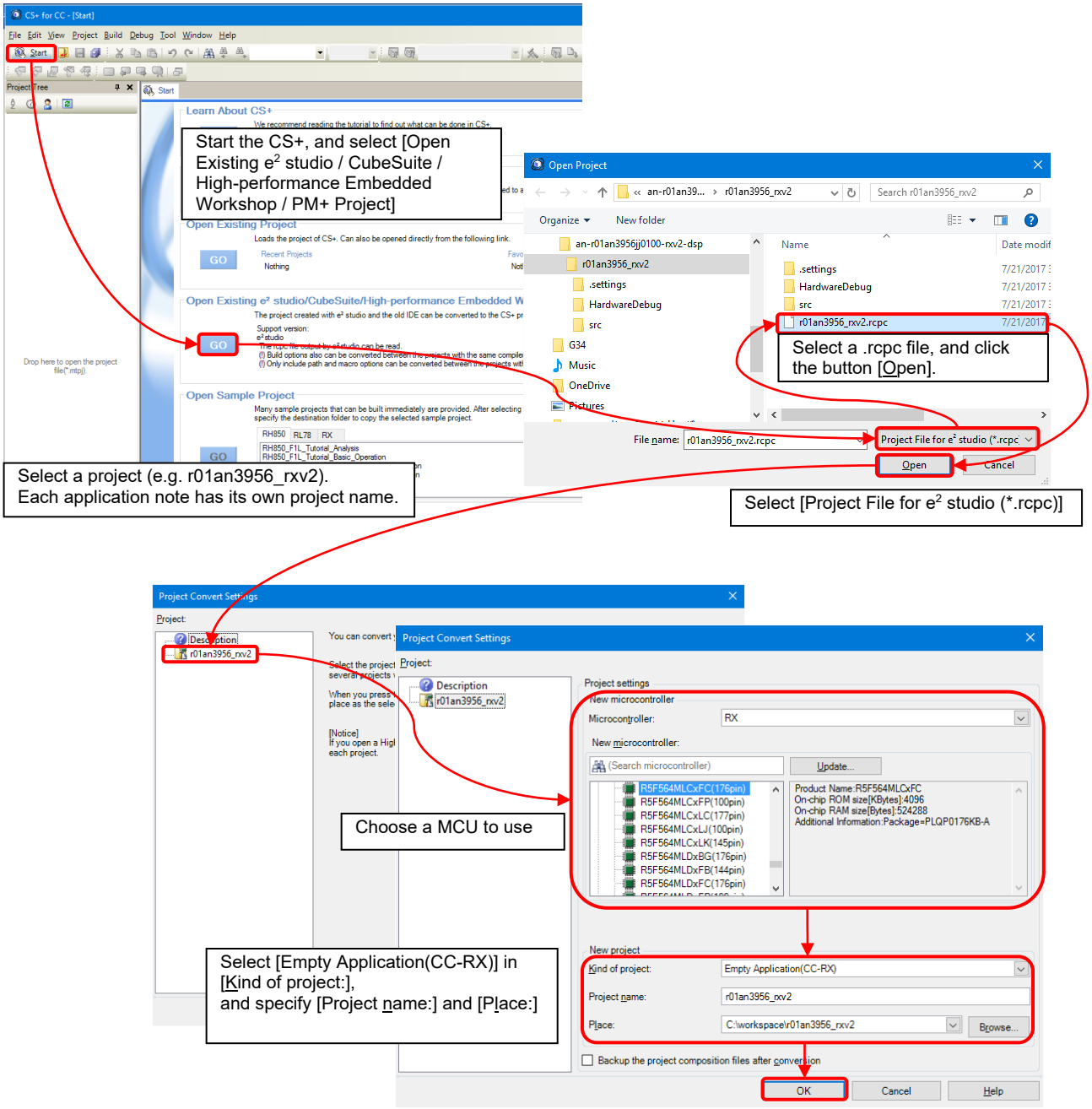


Figure 5.2 Importing a Project in CS+

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

User's Manual: Hardware
RX140 Group User's Manual: Hardware (R01UH0905)
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 Compiler CC-RX User's Manual (R20UT3248)
The latest version can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 31, 2021	–	First edition issued
1.10	Mar. 4, 2022	4	Description of Configuring Nonexistent Ports, added
		7	Table 2.1 Integrated development environment, C compiler, iodefine.h version, and Sample code version., changed. Used MCU, added
		8	Description about how to select the code flash memory capacity, added
		13	Table 4.7 CTSU internal clock options to CKO_CLK content, added. Table 4.7 Constant” CODE_FLASH_MEMORY_SIZE”, added
		14	Notes for” CKO_CTSU”, added
		15	Table 4.8 Constant” CKO_CTSU”, added. Table 4.8 Constant” SIZE_64KB”, added. Table 4.8 Constant” SIZE_NOT_64KB”, added.
		16	Table 4.9 Constant setting value for 80-Pin Package (PIN_SIZE=80), changed. Table 4.10 Constant setting value for 64-Pin Package (PIN_SIZE=64), changed
		19	Description in the remarks column of the function "R_INIT_Port_Initialize", added. "ROM capacity" in the remarks of the function "R_INIT_Port_Initialize" to "code flash memory”, changed
		24	"ROM capacity" in Note to "code flash memory", changed
		34	One sentence to Note 1, added

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/.