RX231 Group

Initial Setting

R01AN2185EJ0110
Rev.1.10
Mar 31, 2017

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

RX231 Group ROM size: 128 KB to 512 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 DMAC, 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 Battery Backup Function

The battery backup function is enabled after the microcontroller is powered on. In the sample code the battery backup function is enabled when the RTC is used and disabled when the RTC is not used.

1.3 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 100-pin products. Change the values according to the product used.

1.4 Setting Clocks

1.4.1 Overview

Clocks are configured in the following steps:

1. Sub-clock setting (including the associated RTC settings and battery backup function)
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 PLL clock is used as the system clock, and the sub-clock, RTC, and battery backup function are not used. Change the constant to select the required clock setting.

1.4.2 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	54 MHz	50 μs*3	
HOCO clock	32 MHz*1	56 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.

1.4.3 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 4.5 and Table 4.6 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		PLL	Main clock	HOCO	Sub-clock
PLL clock		Oscillating	Stopped	Stopped	Stopped
Main clock		Oscillating	Oscillating	Stopped	Stopped
HOCO clock		Stopped	Stopped	Oscillating	Stopped
Sub-clock		Stopped*1	Stopped*1	Stopped*1	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_PLL	CLK_MAIN	CLK_HOCO	CLK_SUB
	SEL_PLL	B_USE	B_NOT_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*1	B_NOT_USE*1	B_NOT_USE*1	B_ USE
	SEL_OPCM	OPCM_HIGH	OPCM_HIGH	OPCM_HIGH	OPCM_LOW
	REG_MEMWAIT*2	MEMWAIT_ON	MEMWAIT_OFF	MEMWAIT_OFF	MEMWAIT_OFF

Notes: 1. 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 the sub-clock for the system clock or the RTC, refer to Table 1.3.

2. When divided by 1 is selected by the SCKCR.ICK[3:0] bits and a clock with a frequency higher than 32 MHz is selected by the SCKCR3.CKSEL[2:0] bits, select MEMWAIT_ON for the REG_MEMWAIT constant since selecting “no wait cycles” is not allowed.

Table 1.3 Examples of the Sub-Clock and RTC Selections

	Sub-Clock	System Clock*2		RTC and Battery Backup Function	
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 and RCR3.RTCEN. When the sub-clock is used as the system clock, it is controlled by the SOSCCR.SOSTP bit, and when the sub-clock is used as the RTC count source, it is controlled by the RCR3.RTCEN bit. 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 has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions (R01AN2185EJ0110)

Item		Contents
MCU used		R5F52318BDFP (RX231 Group)
Operating frequencies	When the PLL clock is selected as the system clock	• Main clock: 8 MHz • Sub-clock: 32.768 kHz (stopped when the sub-clock is not used) • PLL: 54 MHz (main clock divided by 2 and multiplied by 13.5) • LOCO: 4 MHz • HOCO: Stopped • System clock (ICLK): 54 MHz (PLL divided by 1) • Peripheral module clock A (PCLKA): 54 MHz (PLL divided by 1) • Peripheral module clock B (PCLKB): 27 MHz (PLL divided by 2) • Peripheral module clock D (PCLKD): 54 MHz (PLL divided by 1) • External bus clock (BCLK): 27 MHz (PLL divided by 2) • FlashIF clock (FCLK): 27 MHz (PLL divided by 2)
Operating voltage		3.3 V
Integrated development environment		Renesas Electronics Corporation e^2 studio Version 5.3.0.23
C compiler		Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.2.06.00
		Compile options The default setting is used in the integrated development environment.
iodefine.h version		V1.00h
Endian		Little endian
Operating mode		Single-chip mode
Processor mode		Supervisor mode
Sample code version		Version 1.10

Table 2.2 Operation Confirmation Conditions (R01AN2185EJ0100)

Item		Contents
MCU used		R5F52318ADFP (RX231 Group)
Operating frequencies	When the PLL clock is selected as the system clock	• Main clock: 8 MHz • Sub-clock: 32.768 kHz (stopped when the sub-clock is not used) • PLL: 54 MHz (main clock divided by 2 and multiplied by 13.5) • LOCO: 4 MHz • HOCO: Stopped • System clock (ICLK): 54 MHz (PLL divided by 1) • Peripheral module clock A (PCLKA): 54 MHz (PLL divided by 1) • Peripheral module clock B (PCLKB): 27 MHz (PLL divided by 2) • Peripheral module clock D (PCLKD): 54 MHz (PLL divided by 1) • External bus clock (BCLK): 27 MHz (PLL divided by 2) • FlashIF clock (FCLK): 27 MHz (PLL divided by 2)
	When the main clock is selected as the system clock	• Main clock: 8 MHz • Sub-clock: 32.768 kHz (stopped when the sub-clock is not used) • PLL: Stopped • LOCO: 4 MHz • HOCO: Stopped • System clock (ICLK): 8 MHz (main clock divided by 1) • Peripheral module clock A (PCLKA): 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) • External bus clock (BCLK): 8 MHz (main clock divided by 1) • FlashIF clock (FCLK): 8 MHz (main clock divided by 1)
	When the HOCO clock is selected as the system clock	• Main clock: Stopped • Sub-clock: 32.768 kHz (stopped when the sub-clock is not used) • PLL: Stopped • LOCO: 4 MHz • HOCO: 32 MHz • System clock (ICLK): 32 MHz (HOCO divided by 1) • Peripheral module clock A (PCLKA): 32 MHz (HOCO divided by 1) • Peripheral module clock B (PCLKB): 32 MHz (HOCO divided by 1) • Peripheral module clock D (PCLKD): 32 MHz (HOCO divided by 1) • External bus clock (BCLK): 32 MHz (HOCO divided by 1) • FlashIF clock (FCLK): 32 MHz (HOCO divided by 1)
Operating voltage		3.3 V
Integrated development environment		Renesas Electronics Corporation e^2 studio Version 4.0.0.26
C compiler		Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.2.03.00
		Compile options The default setting is used in the integrated development environment.
iodefine.h version		V1.00c
Endian		Little endian
Operating mode		Single-chip mode
Processor mode		Supervisor mode
Sample code version		Version 1.00

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 Rev. 1.00 (R01AN1852EJ).

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
DMAC/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.

4.2.2 Selecting the Number of Pins

The number of pins in the sample code is set for the 100-pin package (PIN_SIZE=100). This application note covers 100-pin, 64-pin, and 48-pin packages. When using products other than the 100 pin-package, change PIN_SIZE in r_init_port_initialize.h to the number of pins on the package used.

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 main clock and PLL are operating, and the HOCO 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 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 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 not 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	Memory wait cycle setting	The wait cycles of the ROM are set. Do not select “no wait cycles” when divided by 1 is selected by the SCKCR.ICK[3:0] bits and a clock of frequency higher than 32 MHz is selected by the SCKCR3.CKSEL[2:0] bits.		Wait cycles set
7	Clock division ratio setting	The clock division ratio is changed.		• ICLK, PCLKA, PCLKD: Divided by 1 • PCLKB, FCLK, BCLK: Divided by 2 • BCLK: Output is stopped.
8	System clock switching	The system clock is switched according to the user system.		Switched to the PLL.

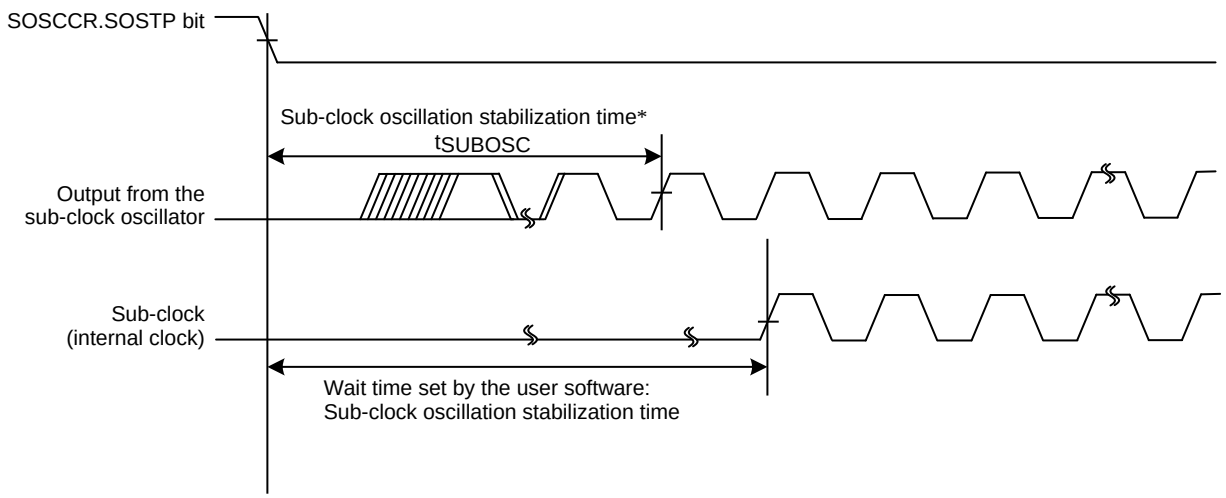
Notes: 1. When selecting each clock usage, change the appropriate constant in r_init_clock.h as required.
2. Refer to 4.3.2 Sub-Clock Oscillation Stabilization Time for details on the sub-clock oscillation stabilization time.

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.



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 File Composition

Table 4.3 lists the files used in the sample code. Files generated by the integrated development environment should not be listed in this table.

Table 4.3 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	

4.5 Option-Setting Memory

Table 4.4 lists the option-setting memory configured in the sample code. When necessary, set a value suited to the user system.

Table 4.4 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.
MDE	FFFF FF83h to FFFF FF80h	FFFF FFFFh	Little endian

4.6 Constants

Table 4.5 and Table 4.6 list the constants used in the sample code, which can be changed by users. Table 4.7 lists the constants used in the sample code, which cannot be changed by users. Table 4.8 lists the Constants when a 100-Pin Package is Used (PIN_SIZE = 100), Table 4.9 lists the Constants when a 64-Pin Package is Used (PIN_SIZE = 64), Table 4.10 lists the Constants when a 48-Pin Package is Used (PIN_SIZE = 48).

Table 4.5 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_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	B_NOT_USE	Selection of the HOCO clock operation: • B_USE: Used (HOCO clock oscillating) • B_NOT_USE: Not used (HOCO clock stopped)
REG_HOCOCR2	FREQ_32MHz	Selection of the HOCO clock frequency • FREQ_32MHz: 32 MHz • FREQ_54MHz: 54 MHz
SEL_PLL	B_ USE	Selection of the PLL clock operation: • B_USE: Used (PLL clock oscillating) • B_NOT_USE: Not used (PLL clock stopped)
REG_PLLCR	1A01h	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 (used as RTC counter source, battery backup function enabled) • B_NOT_USE: Not used (RTC not used, battery backup function disabled)
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_RCR3*1	CL_LOW	Selection of the sub-clock oscillator drive capability: • CL_LOW: Drive capacity for low CL • CL_STD: Drive capacity for standard CL
SEL_CNTMD*1	CNTMD_CAL	Selection of the real-time clock count mode • CNTMD_CAL: Calendar count mode • CNTMD_BIN: Binary count mode

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

Constant Name	Setting Value	Contents
SEL_SYSCLK*1	CLK_PLL	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*6	• 1081 0100h (when PLL is selected) • 0080 0000h (other than above)	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_LOW: Low-speed operating mode*4
REG_MEMWAIT	MEMWAIT_ON	Selection of whether the memory wait cycles are set • MEMWAIT_ON: Wait cycles set • MEMWAIT_OFF: No wait cycles
MSTP_STATE_DMACDTC*2	MODULE_STOP_ DISABLE	Selection of the module-stop state for DMAC/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	100	Number of pins on the product used

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 and all clock sources other than the sub-clock are stopped. Note that in the sample code accompanying this application note only LOCO is stopped when low-speed operating mode is selected. Therefore, select the stopped state for the other clock sources.
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. The setting values differ depending on the clock source selected for the system clock.

Table 4.7 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	02h	Drive capacity for low CL
CL_STD	0Ch	Drive capacity for standard CL
CNTMD_CAL	0	RTC: Calendar count mode
CNTMD_BIN	1	RTC: Binary count mode
CLK_MAIN	0200h	Clock source: Main clock
CLK_PLL	0400h	Clock source: PLL
CLK_HOCO	0100h	Clock source: HOCO
CLK_SUB	0300h	Clock source: Sub-clock
SUB_CLOCK_CYCLE	1000000L/SUB_CLOCK_Hz	Sub-clock cycle (μs)
LOCO_CLOCK_kHz	4560L	LOCO 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_MID	02h	Operating power control mode: Middle-speed operating mode
OPCM_HIGH	00h	Operating power control mode: High-speed operating mode
OPCM_LOW	FFh	Operating power control mode: Low-speed operating mode
OPCM_DEFAULT	OPCM_MID	Operating mode after reset cancellation
MEMWAIT_OFF	00h	No memory wait cycles
MEMWAIT_ON	01h	Memory wait cycles are set
MODULE_STOP_ENABLE	1	Transition to the module stop-state is made
MODULE_STOP_DISABLE	0	Module stop-state is canceled

Table 4.8 Constants when a 100-Pin Package is Used (PIN_SIZE = 100)

Constant Name	Setting Value	Contents
DEF_P0PDR	57h	Setting value for the port P0 direction register
DEF_P1PDR	03h	Setting value for the port P1 direction register
DEF_P2PDR	00h	Setting value for the port P2 direction register
DEF_P3PDR	00h	Setting value for the port P3 direction register
DEF_P4PDR	00h	Setting value for the port P4 direction register
DEF_P5PDR	C0h	Setting value for the port P5 direction register
DEF_PAPDR	00h	Setting value for the port PA direction register
DEF_PBPDR	00h	Setting value for the port PB direction register
DEF_PCPDR	00h	Setting value for the port PC direction register
DEF_PDPDR	00h	Setting value for the port PD direction register
DEF_PEPDR	00h	Setting value for the port PE direction register
DEF_PJPDR	F7h	Setting value for the port PJ direction register

Table 4.9 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	1Ch	Setting value for the port P3 direction register
DEF_P4PDR	A0h	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_PJPDR	FFh	Setting value for the port PJ direction register

Table 4.10 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	B8h	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_PJPDR	FFh	Setting value for the port PJ direction register

4.7 Functions

Table 4.11 lists the functions used in the sample code.

Table 4.11 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
enable_rtc	Initialization when using the 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
R_DELAY	Inline function to specify the number of loops
R_DELAY_us	Function to specify the execution time

4.8 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 100-pin package (PIN_SIZE=100). After this function is called, when writing in byte units to the PDR registers or PODR registers which have nonexistent ports, set the 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.

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 PLL as the system clock without using the sub-clock and RTC.

cgc_oscillation_main	
Outline	Main clock oscillation setting
Header	r_init_clock.h
Declaration	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	void cgc_oscillation_hoco(void)
Description	Enables the HOCO 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	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 oscillation stabilization time.
Arguments	None
Return Value	None

cgc_oscillation_sub	
Outline	Sub-clock oscillation setting
Header	r_init_clock.h
Declaration	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	void cgc_disable_subclk(void)
Description	Configures the setting when the sub-clock is not used as the system clock or 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

enable_rtc	
Outline	Initialization when using the RTC
Header	None
Declaration	static void enable_rtc(void)
Description	Initializes the settings when using 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	uint32_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 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

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 five cycles).
Arguments	loop_cnt: The number of loops
Return Value	None

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

4.9 Flowcharts

4.9.1 Main Processing

Figure 4.2 shows the main processing.

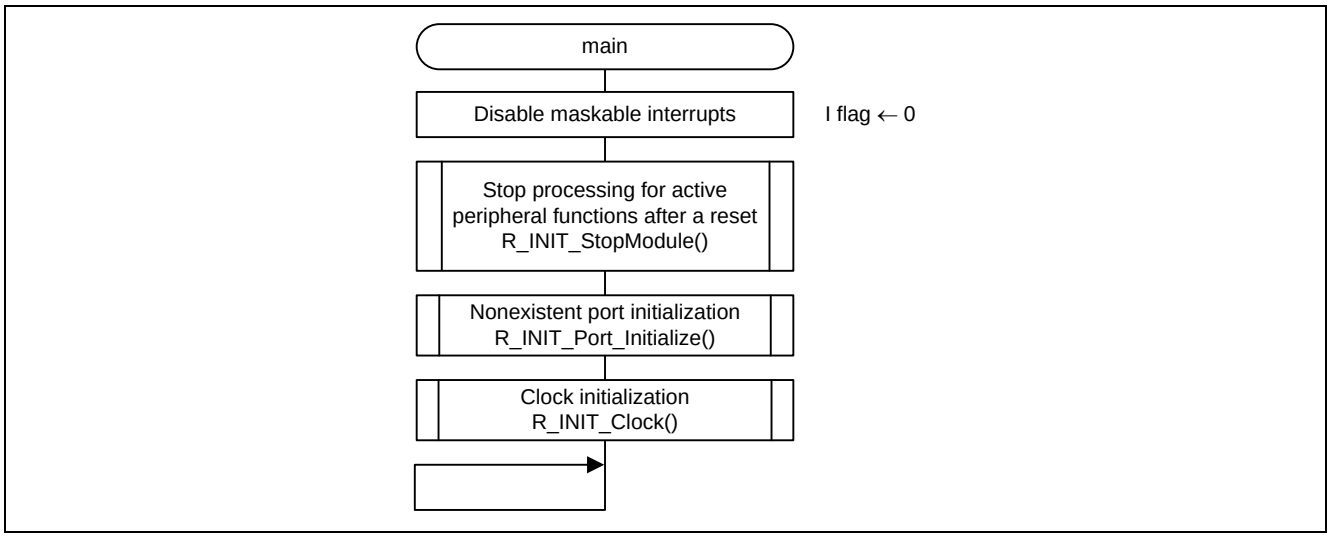
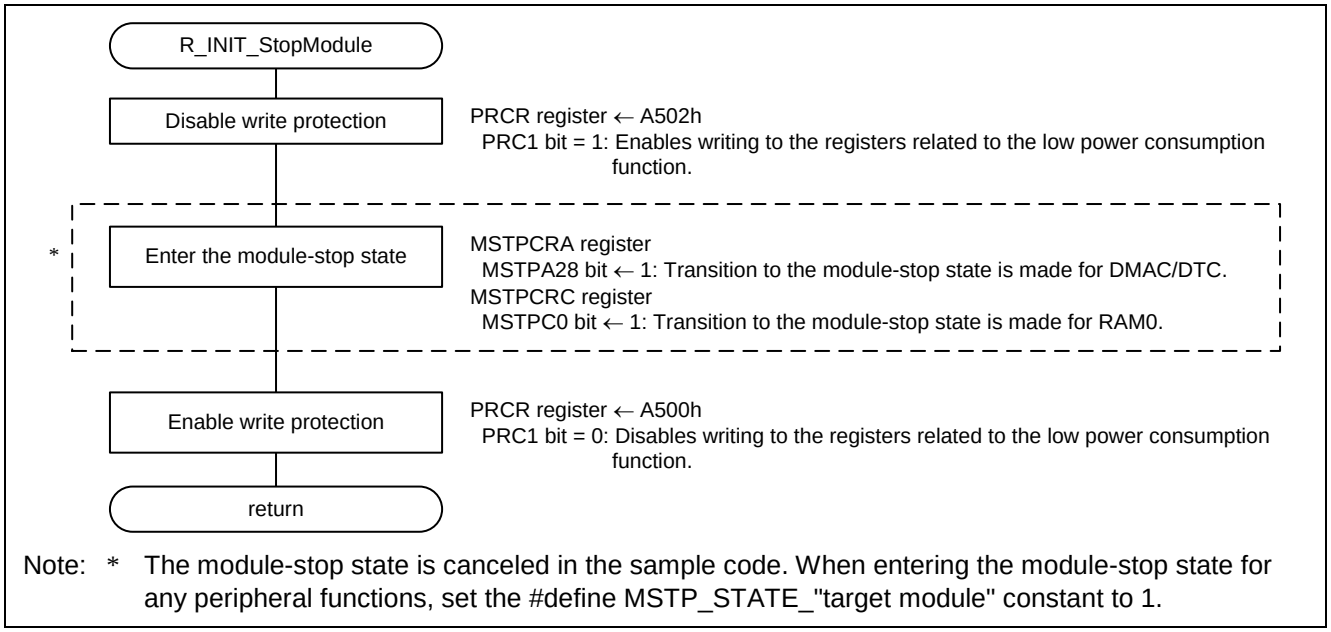


Figure 4.2 Main Processing

4.9.2 Stop Processing for Active Peripheral Functions after a Reset

Figure 4.3 shows the 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 Functions after a Reset

4.9.3 Nonexistent Port Initialization

Figure 4.4 shows the nonexistent port initialization.

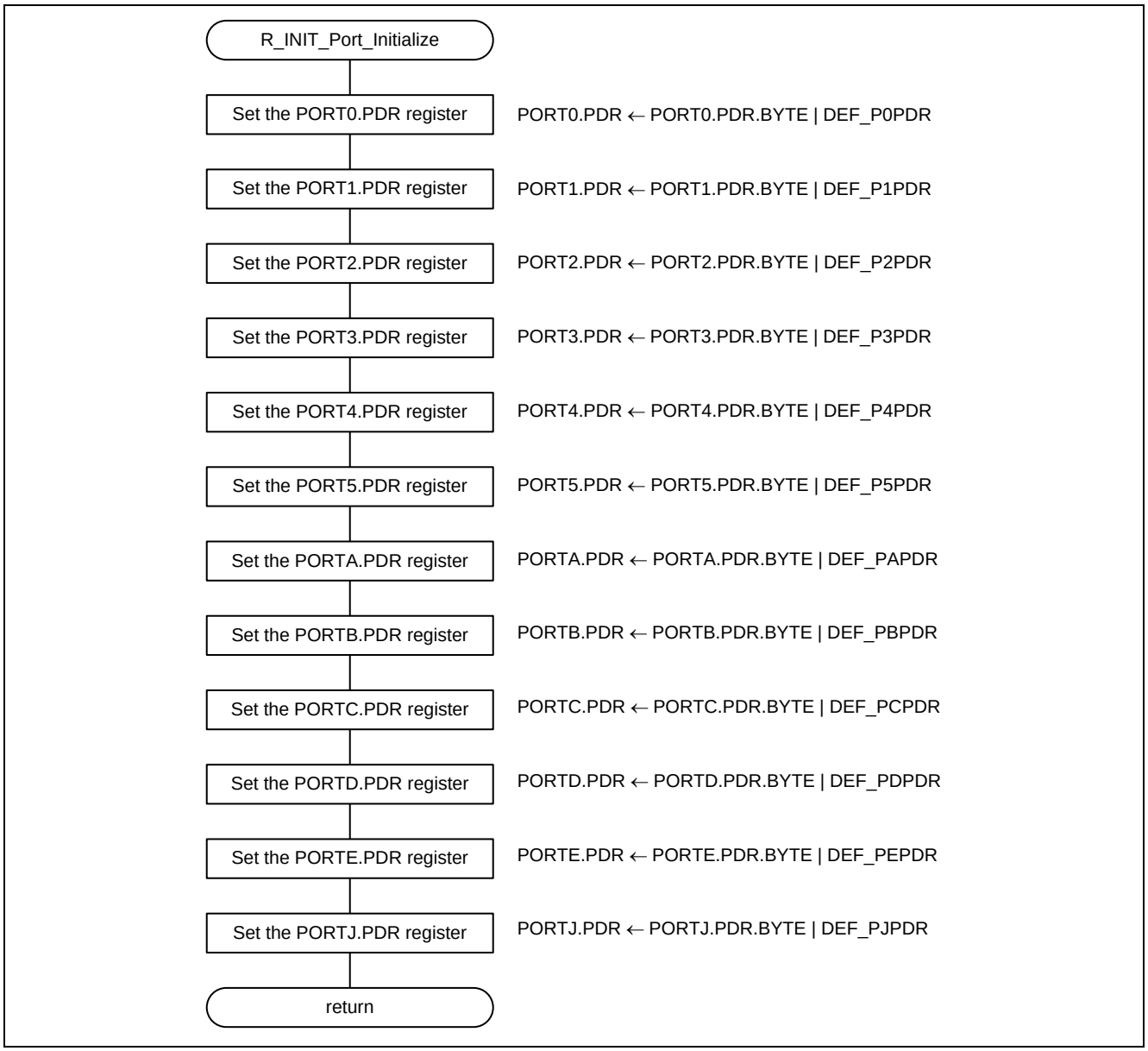
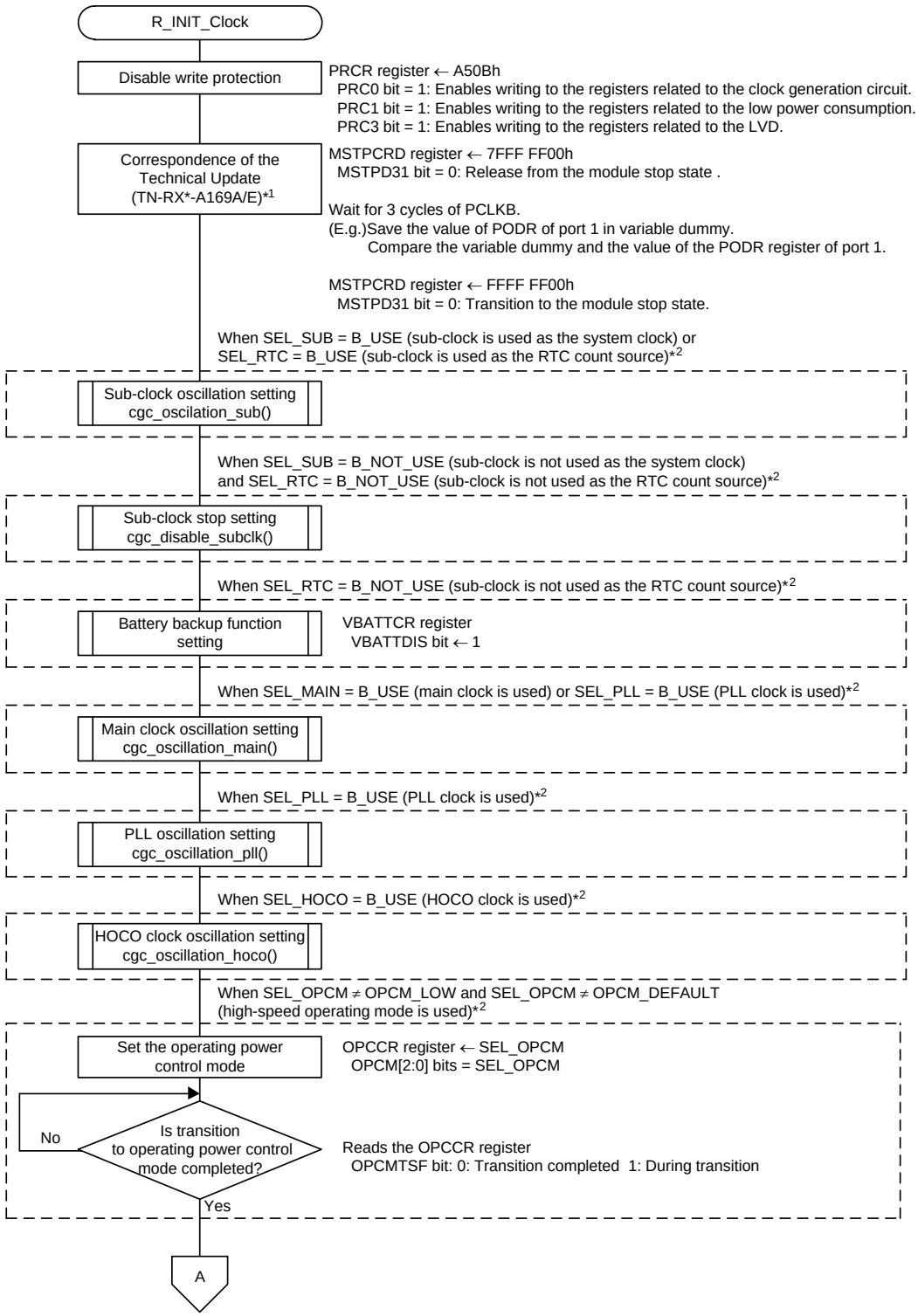


Figure 4.4 Nonexistent Port Initialization

4.9.4 Clock Initialization

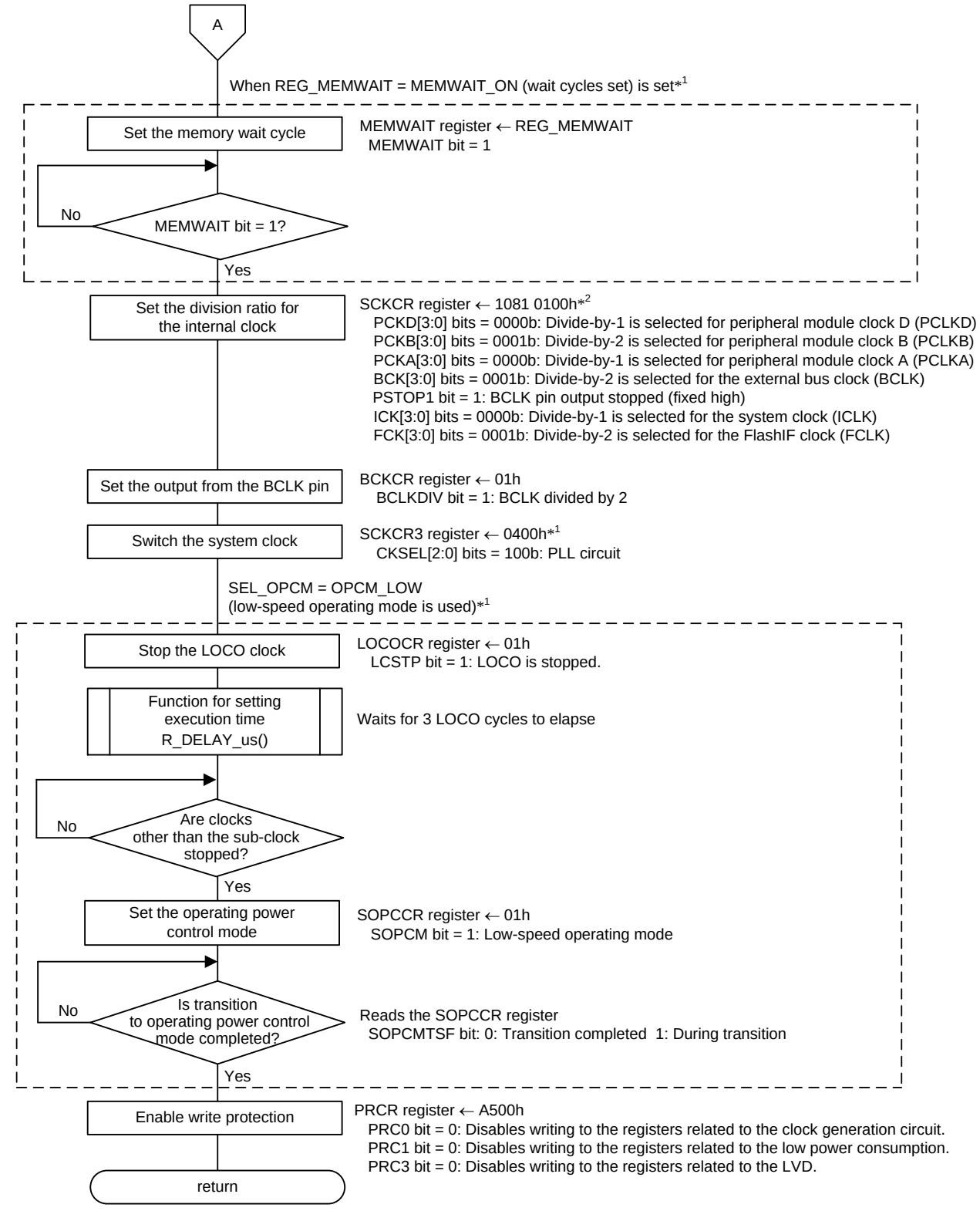
Figures 4.5 and 4.6 show the clock initialization.



Note: 1. Handling of TECHNICAL UPDATE is implemented in an automatic generated file depending on the version of the tool to use. When using, please check it. Even if processing is executed in duplicate, there is no problem in operation.

2. Change the constant value according to the user system.

Figure 4.5 Clock Initialization (1/2)

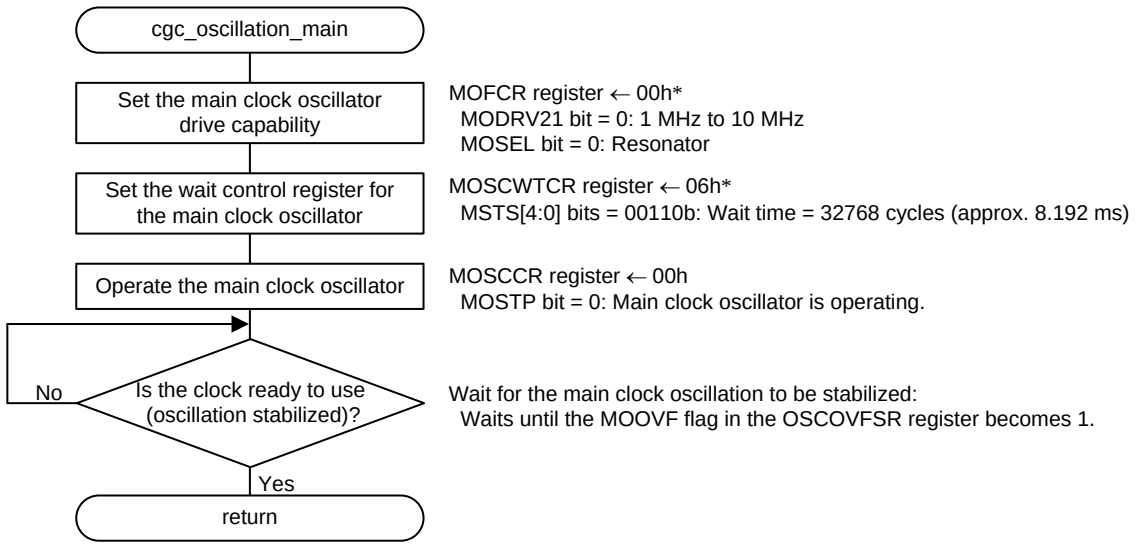


Notes: 1. Change the constant value according to the user system.
2. The setting value varies depending on the system clock selected by the constant.

Figure 4.6 Clock Initialization (2/2)

4.9.5 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

4.9.6 HOCO Clock Oscillation Setting

Figure 4.8 shows the HOCO clock oscillation setting.

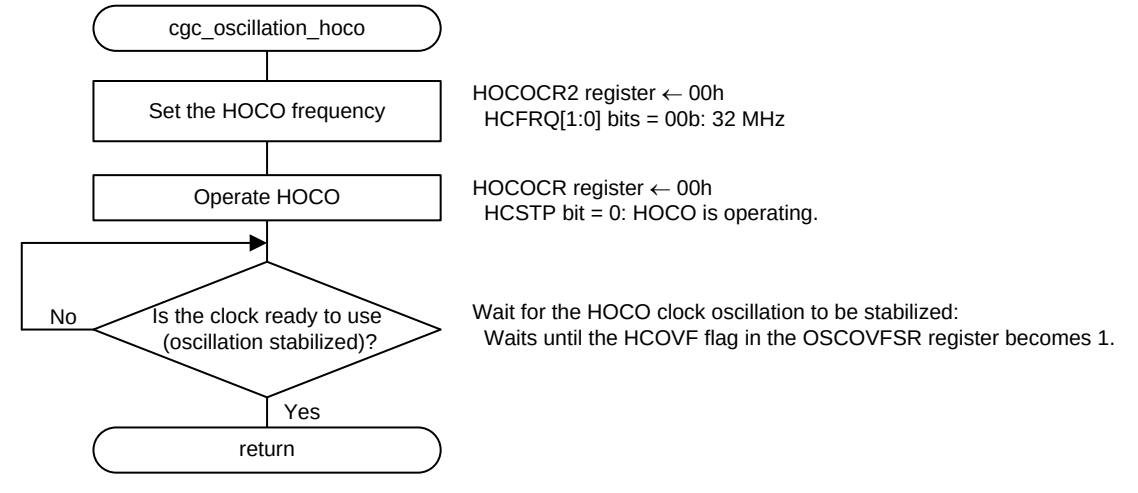
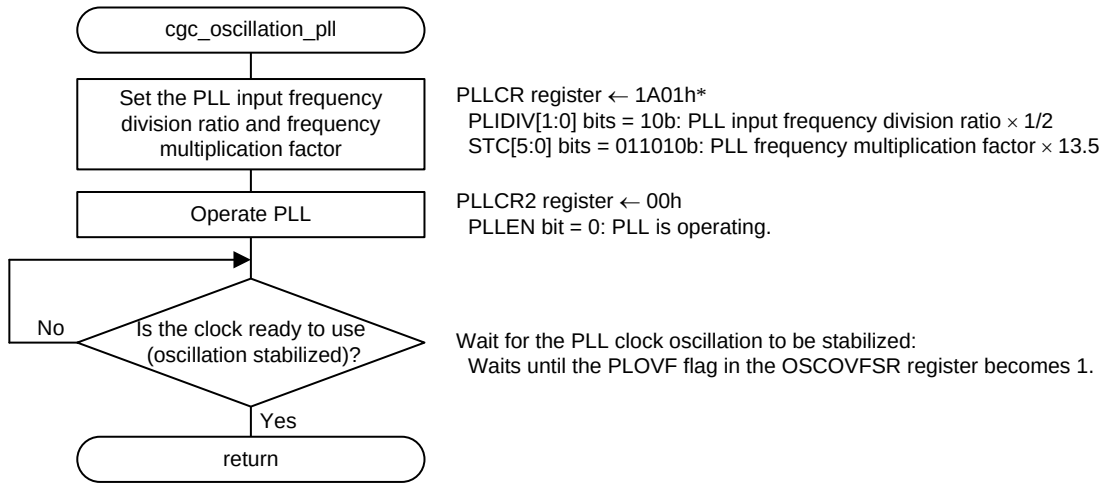


Figure 4.8 HOCO Clock Oscillation Setting

4.9.7 PLL Clock Oscillation Setting

Figure 4.9 shows the PLL clock oscillation setting.

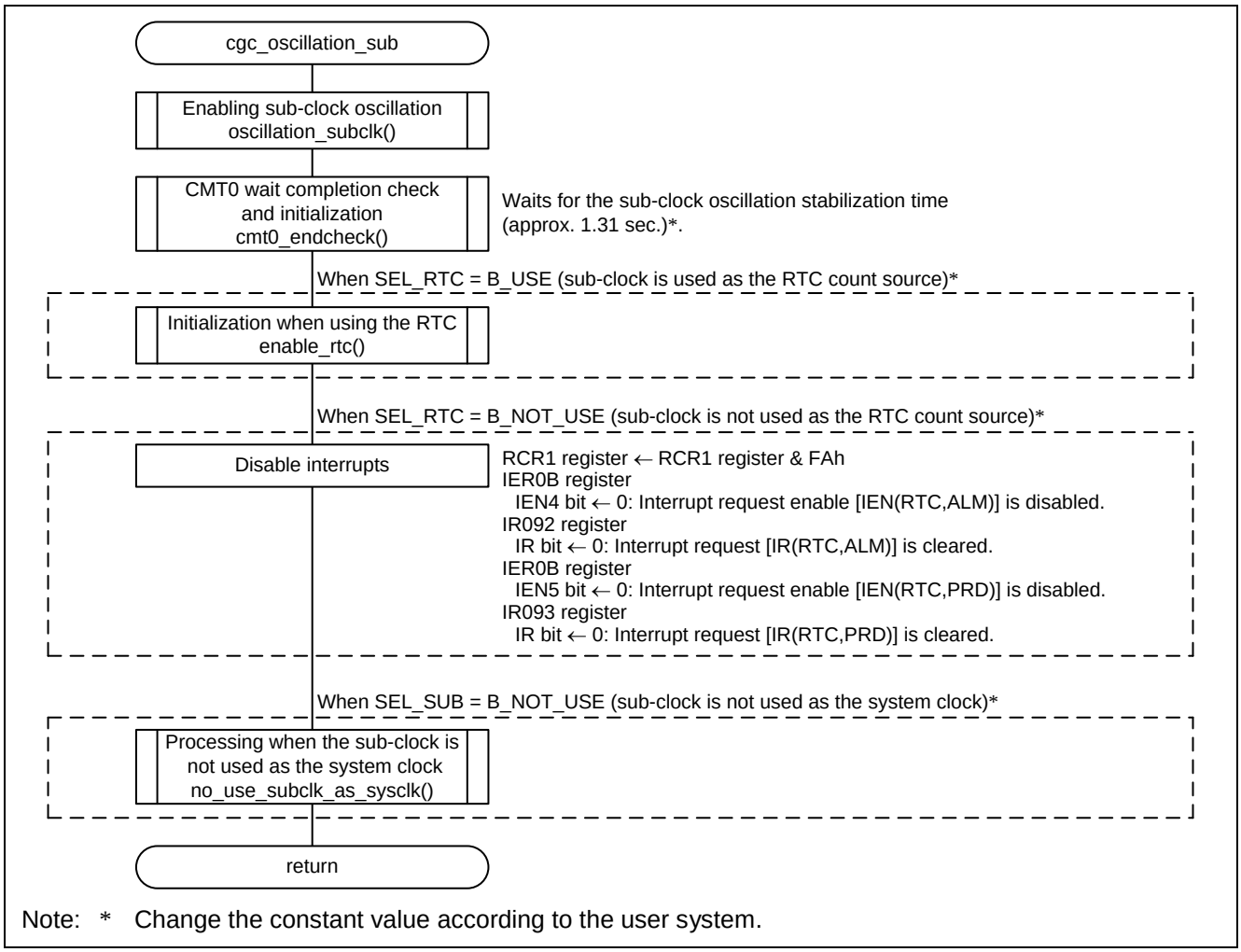


Note: * Change the constant value according to the user system.

Figure 4.9 PLL Clock Oscillation Setting

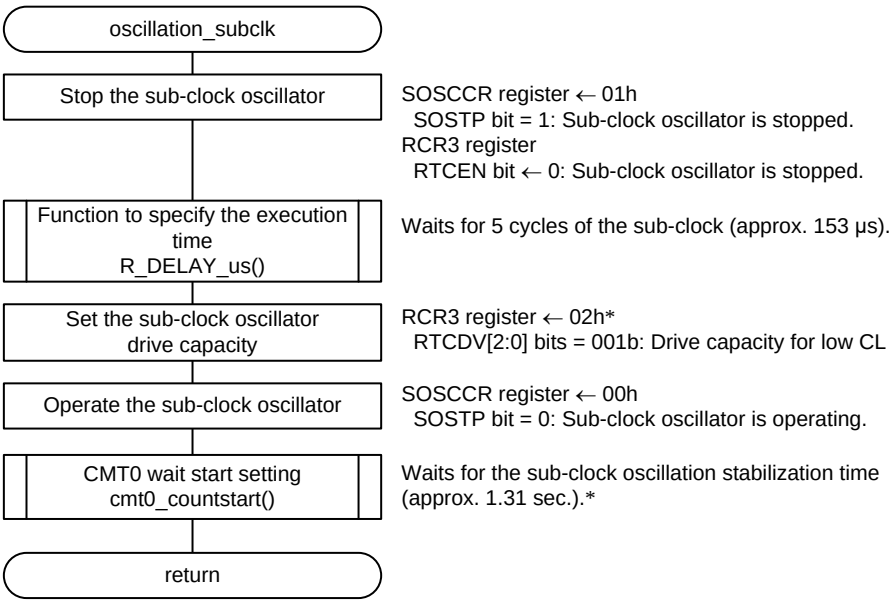
4.9.8 Sub-Clock Oscillation Setting

Figures 4.10 to 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

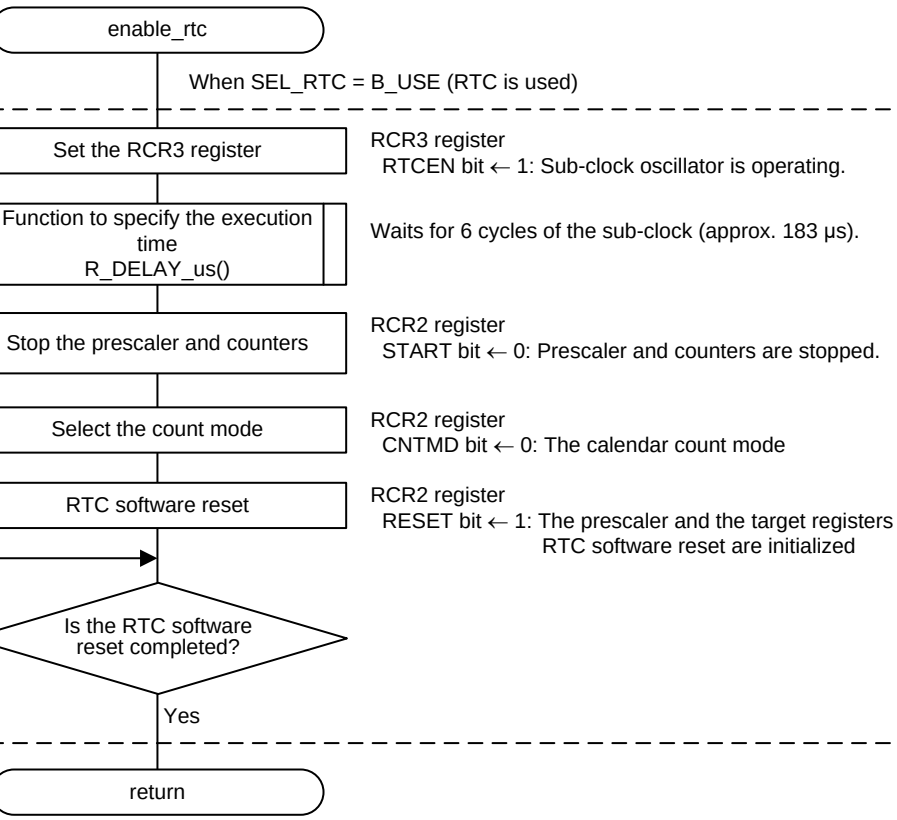


Figure 4.12 Initialization when Using the RTC

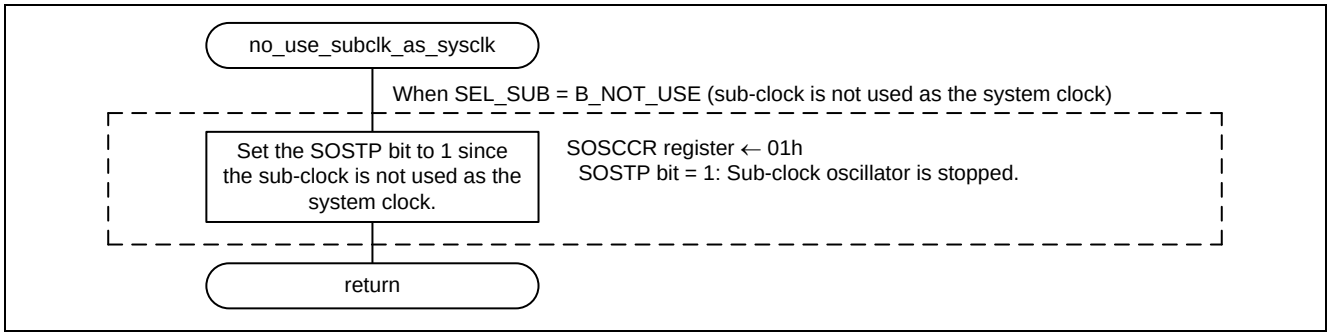


Figure 4.13 Processing when the Sub-Clock is not Used as the System Clock

4.9.9 Sub-Clock Stop Setting

Figure 4.14 shows the sub-clock stop setting.

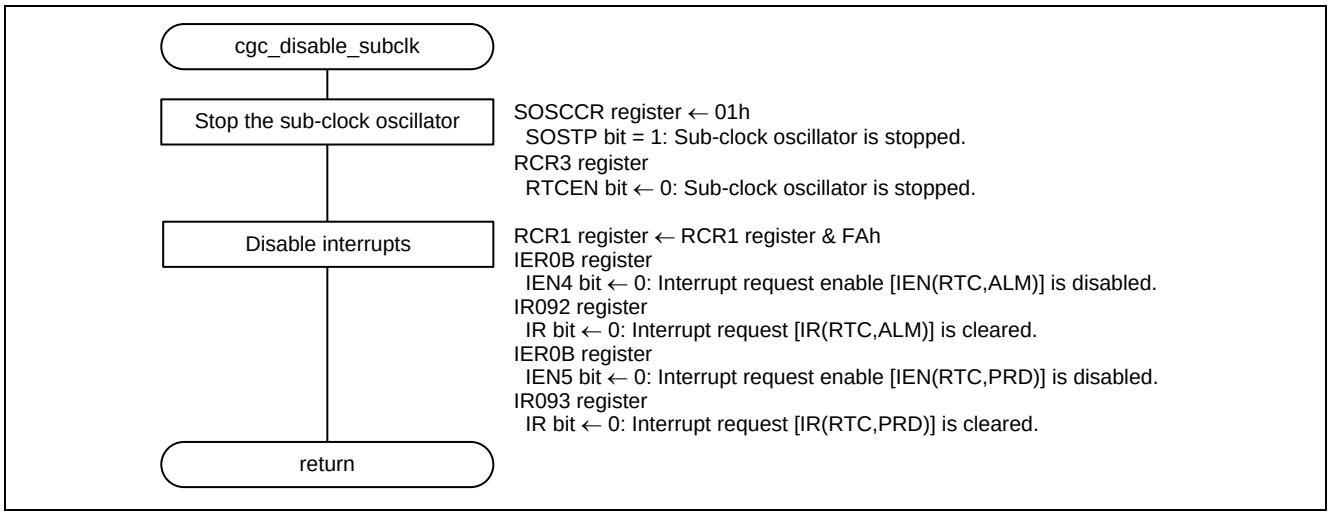


Figure 4.14 Sub-Clock Stop Setting

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

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

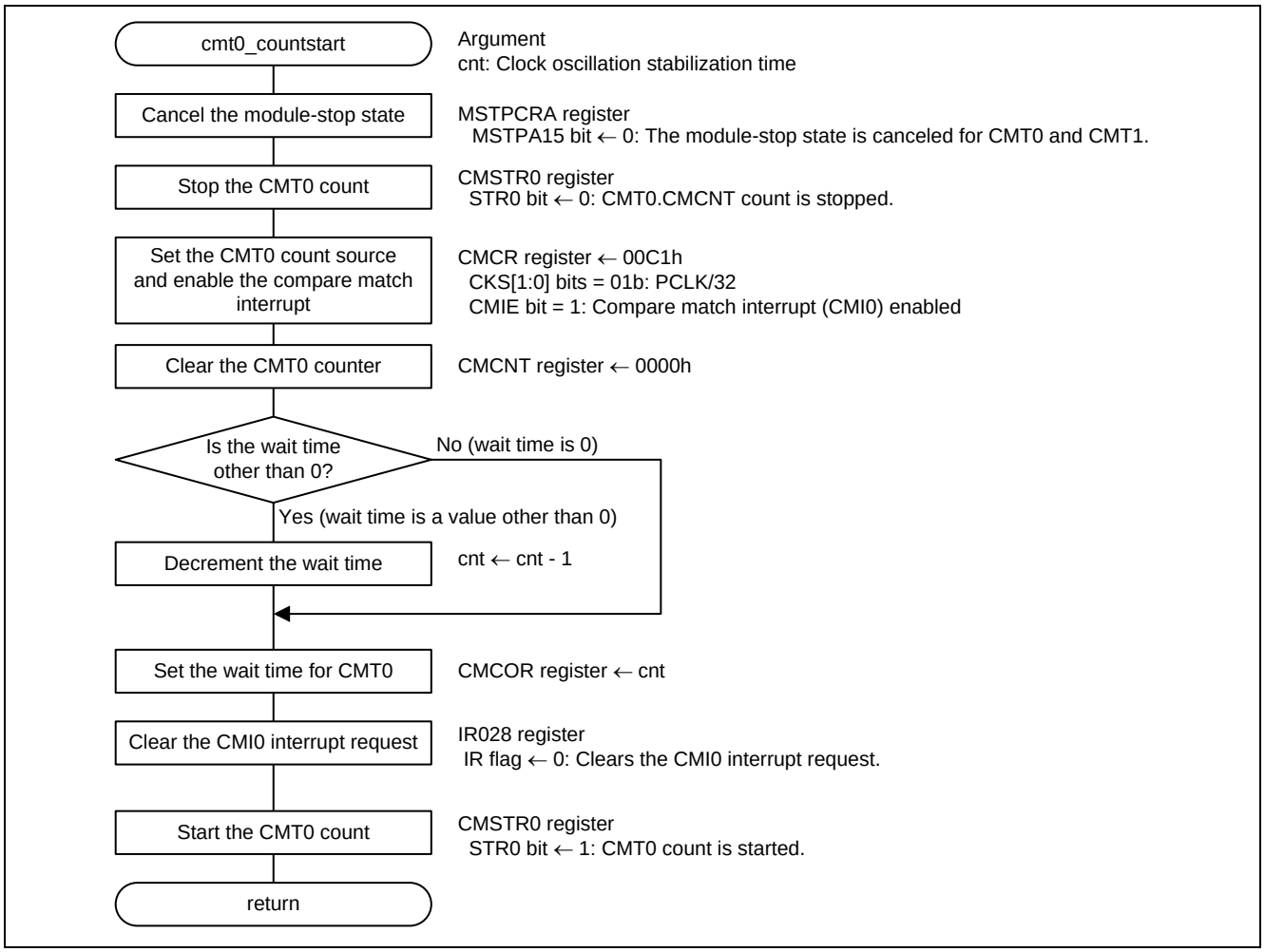
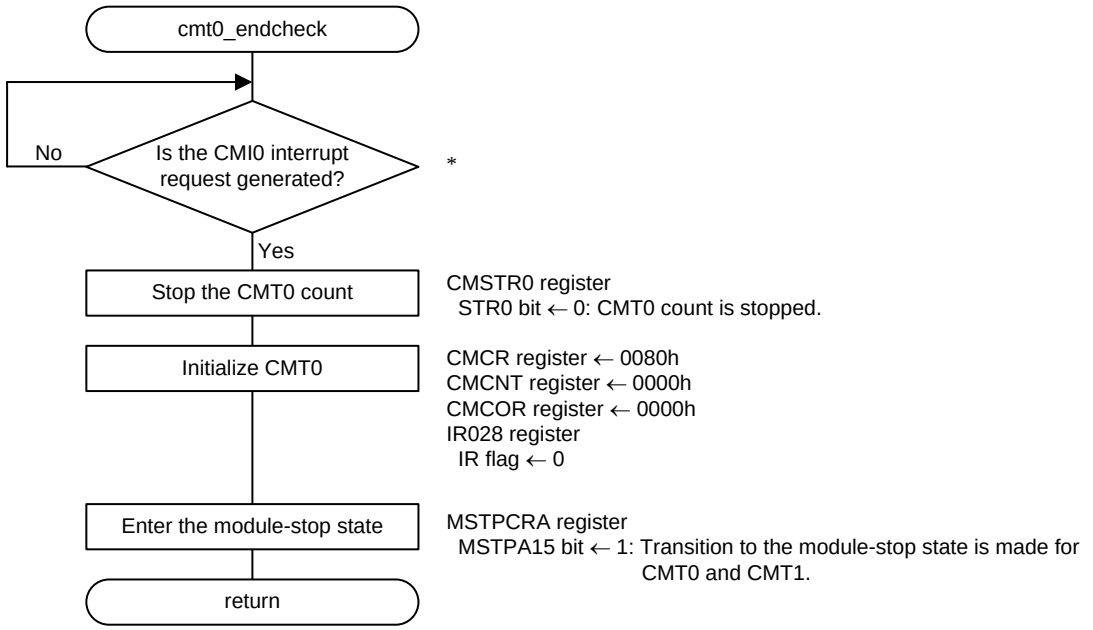


Figure 4.15 CMT0 Wait Start Setting



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

5. Importing a Project

5.1 Importing a Project in the e² studio

When using the e² studio, follow the procedure shown below to import a project into the e² studio.

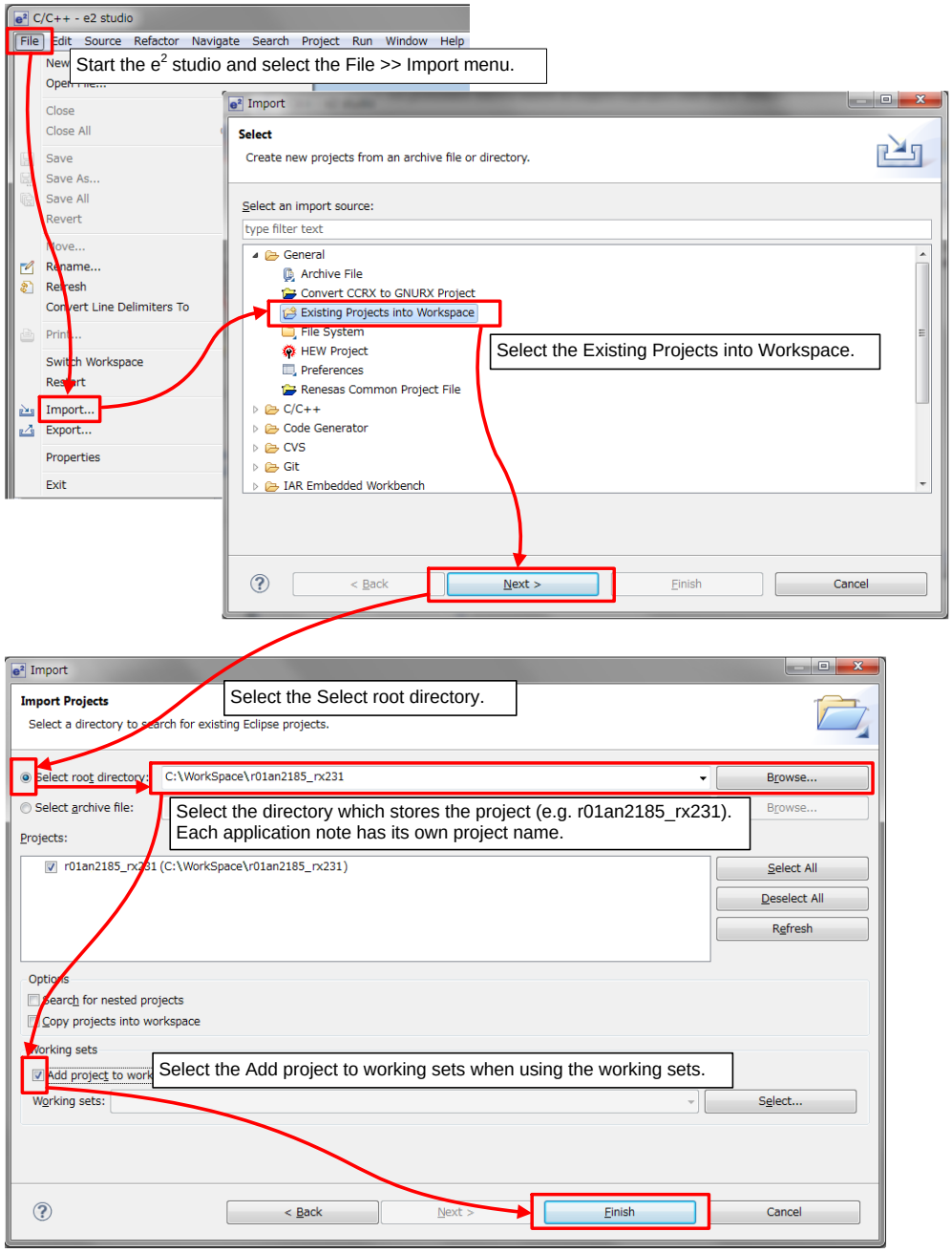


Figure 5.1 Importing a Project in the e² studio

5.2 Importing a Project in CS+

When usin CS+, follow the procedure shown below to import a project into CS+.

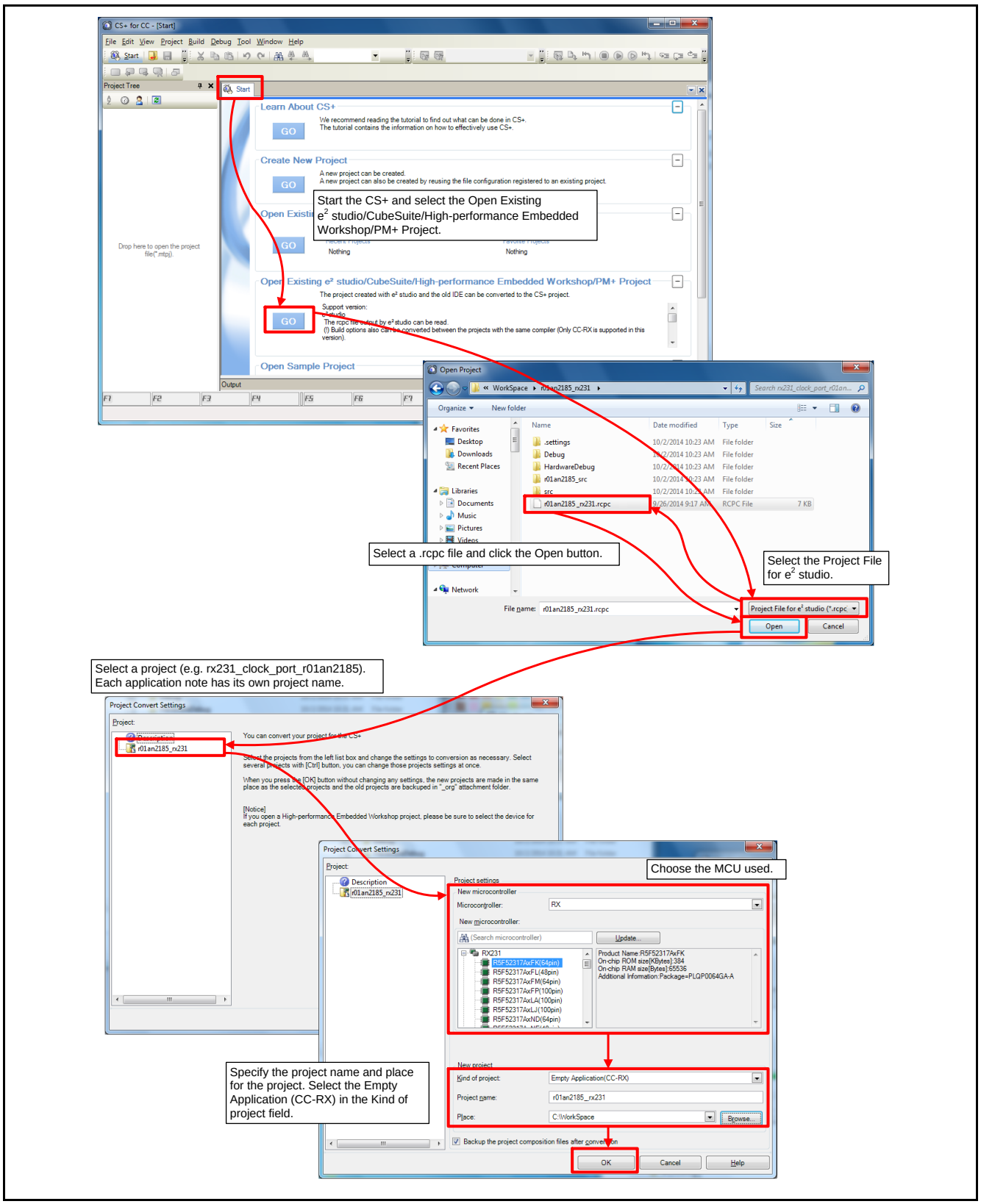


Figure 5.2 Importing a Project in CS+

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

User's Manual: Hardware

RX230 Group, RX231 Group User's Manual: Hardware (R01UH0496)

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.

Consistency with Technical Updates

This application note reflects the contents of the following technical updates:

- TN-RX*-A169A/E

Website and Support

Renesas Electronics Website
http://www.renesas.com/

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

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Sep. 16, 2015	–	First edition issued
1.10	Mar. 31, 2017	5	Added operation confirmation condition of R01AN 2185JJ 0110.
		6	The board used of Table 2.2 is deleted.
		22	Correspondence of TECHNICAL UPDATE is added to figure 4.5.
		33	Revision and Region No. on a user's manual Hardware,changed.
			Title on Compiler CC-RX user’s manual,changed.
			Region No. on Compiler CC-RX user’s manual,changed.
		34	Added chapter of Consistency with Technical Updates.
		Program	Correspondence of a technical update(TN-RX*-A169A/E) is added.
			Changed the version of iodefine.h.

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. Handling of Unused Pins

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

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

2. Processing at Power-on

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

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 disputes 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, drawing, chart, program, algorithm, 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 not alter, modify, copy, or otherwise misappropriate 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, copy or otherwise misappropriation of Renesas Electronics products.
5. 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; and 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.

   Renesas Electronics products are neither intended nor 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 damages (space and 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 third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. When using the 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 radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident arising out of the use of Renesas Electronics products beyond such specified ranges.
7. 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. Further, Renesas Electronics products are not subject to radiation resistance design. Please ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of 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 by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use 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.
9. 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 not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.3.0-1 November 2016)

RENESAS

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

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

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

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

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

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

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

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

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 6.0