RX24T Group

Initial Settings Example

R01AN2837EJ0210
Rev.2.10
Aug. 20. 2020

Introduction

This application note describes the initial settings that must be performed after a reset of RX24T Group microcontroller. The initial settings for RX24T group include clock settings, option to disable running peripheral modules (modules that have default enabled setting after reset), and nonexistent port settings.

Target Device

- RX24T Group 64-pin version, ROM capacity: 128 KB, 256 KB
- RX24T Group 80-pin version, ROM capacity: 128 KB, 256 KB
- RX24T Group 100-pin, 16Kbytes RAM capacity version, ROM capacity: 128 KB, 256 KB
- RX24T Group 100-pin, 32Kbytes RAM capacity version, ROM capacity: 256 KB, 384 KB, 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

The sample code performing settings after a reset is to be able to disable running peripheral modules (modules that have default enabled setting after a reset), execute nonexistent port settings, and clock settings. The description in this application note applies to the processing that is required following to power-on reset (cold start).

1.1 Disabling Peripheral Modules Running After a Reset

Some peripheral modules start operating immediately after power-on reset due to default enabled setting, and others have the default disabled setting. The processing covered under this item disables the following default enabled peripheral modules:

DTC and RAM0

Note that the above processing is not performed by the sample code. As per necessity, it is required to overwrite the corresponding constants, to execute the module stop process.

1.2 Nonexistent Port Settings

It is necessary to set the bits in the port direction registers corresponding to nonexistent ports to a predetermined value. The sample code contains initial port direction register setting values that are suitable for 100-pin, 16Kbytes RAM capacity products. Overwrite the constants as necessary for the actual target device.

1.3 Clock Settings

1.3.1 Overview

The procedure for making clock settings is as follows:

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

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

The sample code sets the PLL clock as the system clock. Overwrite the constants as necessary to match the clock frequency requirements, according to applications.

1.3.2 Oscillation Stabilization Times in Sample Code

Table 1.1 lists the oscillation stabilization times in the sample code. The oscillation stabilization times have been calculated to match the specification of oscillators listed in the table.

Table 1.1 Oscillation Stabilization Times in Sample Code

Clock	Oscillation Frequency	Oscillation Stabilization Time	Remarks
Main clock oscillator	20.0 MHz	8.192 ms*1	Crystal
PLL clock	80.0 MHz	50 μs*2	
HOCO clock	32.0 MHz	4.438 μs*2*3	
	64.0 MHz	4.219 μs*2*3	

Note 1. The oscillation stabilization time of each oscillator will differ depending on conditions such as the wiring pattern of the actual system, the oscillation constant, etc. To determine the appropriate oscillation stabilization time, ask the oscillator manufacturer to evaluate the actual target system.

Note 2. See Electrical Characteristics in RX24T Group User's Manual: Hardware.

Note 3. These values are calculated based on the value of HOCOWTCR.HSTS bits corresponding with HOCO oscillation frequency.

1.3.3 Clock Selection

By making changes to the constants defined in r_init_clock.h, the sample code described in this application note can be used to select settings such as the clock source of the system clock and whether each clock is oscillating or stopped. To determine which constants can be changed, see 3.6 Constants. Table 1.2 lists clock selection examples.

Table 1.2 Clock Selection Examples

No.	1	2	3	4
System clock	PLL (Main)*4	Main clock	PLL (HOCO)*4	HOCO
PLL clock	Oscillating	Stopped	Oscillating	Stopped
Main clock	Oscillating	Oscillating	Stopped	Stopped
HOCO clock	Stopped	Stopped	Oscillating	Oscillating
Operating power control mode	High-speed operating mode	High-speed operating mode	High-speed operating mode	High-speed operating mode
Constants				
SEL_SYSCLK	CLK_PLL	CLK_MAIN	CLK_PLL	CLK_HOCO
SEL_PLL	B_USE_PLL_MAIN	B_NOT_USE	B_USE_PLL_HOCO	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_USE
SEL_HOCO_FREQ	————	————	HOCO_32MHz*3	HOCO_64MHz*3
REG_OPCCR	OPCM_HIGH	OPCM_HIGH	OPCM_HIGH	OPCM_HIGH
REG_MEMWAIT *1*2	MEMWAIT_ON_ICLK_80MHZ_AND_LESS	MEMWAIT_OFF	MEMWAIT_ON_ICLK_80MHZ_AND_LESS	MEMWAIT_ON_ICLK_64MHZ_AND_LESS

Note 1. Selecting no wait cycles is prohibited when a clock of frequency higher than 32 MHz is selected as the system clock (ICLK). In this case, select MEMWAIT_ON_ICLK_64MHZ_AND_LESS or MEMWAIT_ON_ICLK_80MHZ_AND_LESS as the setting for REG_MEMWAIT.

Note 2. Selecting no wait cycles or wait cycles when ICLK ≤ 64 MHz is prohibited when a clock of frequency higher than 64 MHz is selected as the system clock (ICLK). In this case, select MEMWAIT_ON_ICLK_80MHZ_AND_LESS as the setting for REG_MEMWAIT.

Note 3. Selecting HOCO oscillator by setting B_USE for SEL_HOCO. The frequency of HOCO is selected by setting SEL_HOCO_FREQ to HOCO_32MHz or HOCO_64MHz. The HOCO stabilization wait time setting is performed by assigning HOCO_WAIT constant equal to HOCO_WAIT_142CYCLE corresponding to HOCO_32MHz and HOCO_WAIT_270CYCLE corresponding to HOCO_64MHz.

Note 4. PLL (Main) and PLL (HOCO) indicate about the input clock to PLL.

2. Operation Confirmation Conditions

The operation of the sample code referenced in this application note has been confirmed under the following conditions.

Table 2.1 Operation Confirmation Conditions

Item		Contents
MCU used		R5F524TAADFP (RX24T Group)
Operating frequency	PLL (Main) clock selected as system clock	Main clock: 20.0 MHz PLL: 80.0 MHz (main clock ×1/2 ×8) HOCO: Stopped LOCO: 4 MHz System clock (ICLK): 80.0 MHz (PLL ×1/1) Peripheral module clock A (PCLKA): 80.0 MHz (PLL ×1/1) Peripheral module clock B (PCLKB): 40.0 MHz (PLL ×1/2) Peripheral module clock D (PCLKD): 40.0 MHz (PLL ×1/2) Flash interface clock (FCLK): 20.0 MHz (PLL ×1/4)
	Main clock selected as system clock	Main clock: 20.0 MHz PLL: Stopped HOCO: Stopped LOCO: 4 MHz System clock (ICLK): 20.0 MHz (main clock ×1/1) Peripheral module clock A (PCLKA): 20.0 MHz (main clock ×1/1) Peripheral module clock B (PCLKB): 20.0 MHz (main clock ×1/1) Peripheral module clock D (PCLKD): 20.0 MHz (main clock ×1/1) Flash interface clock (FCLK): 20.0 MHz (main clock ×1/1)
	PLL (HOCO) clock selected as system clock	Main clock: Stopped PLL: 80.0 MHz (HOCO clock ×1/4 ×10) HOCO: 32 MHz LOCO: 4 MHz System clock (ICLK): 80.0 MHz (PLL ×1/1) Peripheral module clock A (PCLKA): 80.0 MHz (PLL ×1/1) Peripheral module clock B (PCLKB): 40.0 MHz (PLL ×1/2) Peripheral module clock D (PCLKD): 40.0 MHz (PLL ×1/2) Flash interface clock (FCLK): 20.0 MHz (PLL ×1/4)
	HOCO clock selected as system clock	Main clock: Stopped PLL: Stopped HOCO: 64 MHz LOCO: 4 MHz System clock (ICLK): 64.0 MHz (HOCO×1/1) Peripheral module clock A (PCLKA): 64.0 MHz (HOCO clock ×1/1) Peripheral module clock B (PCLKB): 32.0 MHz (HOCO clock ×1/2) Peripheral module clock D (PCLKD): 32.0 MHz (HOCO clock ×1/2) Flash interface clock (FCLK): 32.0 MHz (HOCO clock ×1/2)
Operating voltage		3.3 V
Integrated development environment		Renesas Electronics e² studio 2020-04
C compiler		Renesas Electronics C/C++ Compiler Package for RX Family V3.02.00
		Compiler option The integrated development environment default settings are used.
iodefine.h version		V1.0H
Endian		Little endian, big endian
Operating mode		Single-chip mode

Item	Contents
Processor mode	Supervisor mode
Sample code version	Version 2.10
Board used	Renesas Starter Kit for RX24T (product No.: RTK500524TC01000BR)

Notes:

If the same version of the toolchain (C compiler) specified in the original project is not in the import destination, the toolchain will not be selected and an error will occur.
Check the selected status of the toolchain on the project configuration dialog.

For the setting method, refer to FAQ 3000404.

FAQ 3000404 :Program ""make"" not found in PATH’ error when attempting to build an imported project (e² studio)"

3. Software

After performing disable peripheral modules (modules having default enabled setting after a reset) settings and nonexistent port settings, the sample code executes the clock settings.

3.1 Disabling Peripheral Modules Running After a Reset

The sample code includes function to disable peripheral modules that have default enabled setting after a reset.

Only the peripheral modules listed below are not in the module stop state after a reset. The transition of module to the module stop state, is performed by setting the corresponding module stop bit to 1. Putting modules into the module stop state reduces the power consumption of the device.

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

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

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

Peripheral Module	Module Stop Setting Bit	Value After Reset	Setting When Not Using Module
DTC	MSTPCRA.MSTPA28 bit	0 (module stop state canceled)	1 (transition to module stop state)
RAM0	MSTPCRC.MSTPC0 bit		

3.2 Nonexistent Port Settings

3.2.1 Processing Overview

The sample code sets nonexistent port pins direction control bits to values that are specified by user's manual. The initial setting for a nonexistent port pin must not change in the user program, so writing in byte unit to PDR or PODR registers must be done carefully. The writing to PDR and PODR registers in byte unit must ensure to set the direction control bits to values defined in user's manual and the port output data storage bits to 0, corresponding to the nonexistent ports.

3.2.2 Pin Count and RAM Capacity Setting

The setting in the sample code (RAM_SIZE = 16, PIN_SIZE = 100) is for 100-pin, 16Kbytes RAM capacity products. The pin counts supported by this application note are 64, 80 and 100. If the pin count of the target device is other than 100 and RAM capacity is different from 16 Kbytes, change the value of PIN_SIZE and RAM_SIZE in r_init_port_initialize.h to match the target device.

3.3 Clock Settings

3.3.1 Clock Setting Procedure

Table 3.2 lists the steps in the clock setting procedure, the processing performed in each step, and the settings of the sample code. Using the settings, the sample code sets the PLL clock as the main clock.

Table 3.2 Clock Setting Procedure

Step	Processing	Details of Processing		Remarks
1	Main clock setting*1	Used (default)	Sets the main clock drive capacity and sets MOSCWTCR register to select the oscillation stabilization wait time, then enables operation of main clock oscillation. After this, waits for the oscillation stabilization time to elapse.	
		Not used	No setting performed to main clock. The registers related to main clock contain the value same as after reset.	
2.	HOCO clock setting*1	Used	HOCOCR2 register is set to 32 MHz/64 MHz Oscillation mode. HOCOWTCR register is set for 142 wait cycles, corresponding to 32 MHz HOCO oscillation and 270 wait cycles, corresponding to 64 MHz HOCO oscillation. Then enables the operation of HOCO clock oscillator and wait for the HOCO clock oscillation stabilization time to elapse.	The HOCO clock is stopped.
		Not used (default)	No setting performed to HOCO clock. The registers related to HOCO clock contain the value same as after reset.	
3	PLL clock setting*1	Used with main clock as input (default)	Sets the PLL input division ratio and frequency multiplication factor, then starts oscillation by the PLL clock. After this, waits for the oscillation stabilization time to elapse.	
		Used with HOCO clock as input	Sets the PLL input division ratio and frequency multiplication factor, then starts oscillation by the PLL clock. After this, waits for the oscillation stabilization time to elapse.	
		Not used	No setting performed to PLL clock. The registers related to PLL clock contain the value same as after reset.	
4	Operating power control mode setting	Sets the operating power control mode according to the operating frequency and operating voltage used.		High-speed operating mode is selected.
5	Memory wait cycles setting	Specifies whether or not wait cycles are inserted for ROM. Selecting no wait cycles is prohibited when ×1/1 is selected by bits SCKCR.ICK[3:0] and a clock frequency of 32 MHz or higher is selected by bits SCKCR3.CKSEL[2:0]. Selecting no wait cycles or wait cycles when ICLK ≤ 64 MHz is prohibited when ×1/1 is selected by bits SCKCR.ICK[3:0] and a clock frequency of higher than 64 MHz is selected by bits SCKCR3.CKSEL[2:0].		Wait cycles are enabled. (ICLK ≤ 80 MHz)
6	Clock division ratio settings	Changes the clock division ratios.		• ICLK, PCLKA: ×1/1 • PCLKB, PCLKD: ×1/2 • FCLK: ×1/4
7	System clock switching	Switches according to the system used.		Switches to PLL clock.

Note 1. Change the values of the constants in r_init_clock.h as necessary to match the selection of the clocks you wish to use or not use.

3.4 File Composition

Table 3.3 lists the files used in the sample code. Files generated by the integrated development environment are not included in this table.

Table 3.3 Files Used in the Sample Code

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

3.5 Option-Setting Memory

Table 3.4 lists the option-setting memory configured in the sample code. When necessary, set a value suited to the user system.

Table 3.4 Option-Setting Memory Configured in the Sample Code

Symbol	Address	Setting Value	Contents
OFS0	FFFF FF8Fh to FFFF FF8Ch	FFFF FFFFh	IWDT stopped after a reset
OFS1	FFFF FF8Bh to FFFF FF88h	FFFF FFFFh	Voltage monitor 0 reset disabled after a reset HOCO oscillation disabled after a reset
MDE	FFFF FF83h to FFFF FF80h	FFFF FFFFh	Little endian

3.6 Constants

Table 3.5 lists the user changeable constants used in the sample code, Table 3.6 lists the fixed constants (not changeable by user), Table 3.7 lists the constants specific to 100-pin, 32Kbytes RAM capacity products (RAM_SIZE = 32, PIN_SIZE = 100), Table 3.8 lists the constants specific to 100-pin, 16Kbytes capacity products (RAM_SIZE = 16, PIN_SIZE = 100), Table 3.9 lists the constants specific to 80-pin, 16Kbytes capacity products (RAM_SIZE = 16, PIN_SIZE = 80), Table 3.10 lists the constants specific to 64-pin, 16Kbytes capacity products (RAM_SIZE = 16, PIN_SIZE = 64).

Table 3.5 User Changeable Constants Used in Sample Code

Constant Name	Setting Value	Contents
SEL_MAIN*1	B_USE	Main clock enable/disable selection B_USE: Used (main clock enabled) B_NOT_USE: Not used (main clock disabled)
REG_MOFCR*1	20h	Main clock oscillator drive capacity setting (setting value of MOFCR register)
REG_MOSCWTCR*1	06h	Setting value of main clock wait control register
SEL_PLL*1	B_USE_PLL_MAIN	PLL clock enable/disable selection B_USE_PLL_MAIN: Used (PLL clock enabled, using main clock) B_USE_PLL_HOCO: Used (PLL clock enabled, using HOCO clock) B_NOT_USE: Not used (PLL clock disabled)
REG_PLLCR*1	0F01h	PLL input division ratio and frequency multiplication factor settings (setting value of PLLCR register)
SEL_HOCO*1	B_NOT_USE	HOCO clock enable/disable selection B_USE: Used (HOCO clock enabled) B_NOT_USE: Not used (HOCO clock disabled)
SEL_HOCO_FREQ	HOCO_32MHz	Selecting frequency for HOCO clock HOCO_32MHz (selecting 32 MHz) HOCO_64MHz (selecting 64 MHz)
SEL_SYSCLK*1	CLK_PLL	System clock source selection CLK_MAIN: Main clock CLK_HOCO: HOCO clock CLK_PLL: PLL clock
REG_SCKCR*1	2001 0101h (PLL selected)	Internal clock frequency setting value (SCKCR register setting value) 0000 0000h (Main selected) 10010101 (HOCO selected)
REG_OPCCR*1	OPCM_HIGH	Operating power control mode selection*4 OPCM_HIGH: High-speed operating mode OPCM_MID: Mid-speed operating mode
REG_MEMWAIT*1	MEMWAIT_ON_ICLK_80MHZ_AND_LESS (PLL selected)	Memory wait cycle selection MEMWAIT_ON_ICLK_80MHZ_AND_LESS: Wait cycles enabled (ICLK ≤ 80 MHz) MEMWAIT_ON_ICLK_64MHZ_AND_LESS: Wait cycles enabled (ICLK ≤ 64 MHz) MEMWAIT_OFF: Wait cycles disabled (ICLK ≤ 32 MHz)
MSTP_STATE_DTC*2	MODULE_STOP_DISABLE	DTC module stop state selection MODULE_STOP_DISABLE: Disable module stop MODULE_STOP_ENABLE: Transition to module stop

MSTP_STATE_RAM0 *2	MODULE_STOP_DISABLE	RAM0 module stop state selection MODULE_STOP_DISABLE: Operating MODULE_STOP_ENABLE: Stopped
PIN_SIZE*3	100	Pin count of target device
RAM_SIZE*3	16	RAM capacity of target device

Note 1. Change the setting values in r_init_clock.h to match the target system.
Note 2. Change the setting values in r_init_stop_module.h to match the target system.
Note 3. Change the setting values in r_init_port_initialize.h to match the target system.
Note 4. The operating frequency range and operating voltage range differ depending on the operating mode. For details, see RX24T Group User's Manual: Hardware.

Table 3.6 Fixed Constants (Not Changeable by user) Used in Sample Code

Constant Name	Setting Value	Contents
B_NOT_USE	0	Not used
B_USE	1	Used
B_USE_PLL_MAIN	2	Use PLL, main clock as PLL input
B_USE_PLL_HOCO	3	Use PLL, HOCO clock as PLL input
CLK_HOCO	0100h	Clock source: HOCO
CLK_MAIN	0200h	Clock source: Main clock
CLK_PLL	0400h	Clock source: PLL
HOCO_32MHz	00h	Select HOCO 32 MHz
HOCO_64MHz	03h	Select HOCO 64 MHz
HOCO_WAIT_142CYCLE	05h	HOCO stabilization wait time (HOCO 32 MHz)
HOCO_WAIT_270CYCLE	06h	HOCO stabilization wait time (HOCO 64 MHz)
OPCM_MID	02h	Operating power control mode: Mid-speed operating mode
OPCM_HIGH	00h	Operating power control mode: High-speed operating mode
MEMWAIT_OFF	00h	Memory wait cycles disabled
MEMWAIT_ON_ICLK_64MHZ_AND_LESS	01h	Memory wait cycles enabled (ICLK ≤ 64 MHz)
MEMWAIT_ON_ICLK_80MHZ_AND_LESS	02h	Memory wait cycles enabled (ICLK ≤ 80 MHz)
MODULE_STOP_ENABLE	1	Transition to module stop state
MODULE_STOP_DISABLE	0	Cancel module stop state

Table 3.7 Constants for 100-Pin, 32Kbytes RAM Capacity Products (RAM_SIZE = 32, PIN_SIZE=100)

Constant Name	Setting Value	Contents
DEF_P0PDR	00h	Port P0 direction register setting value
DEF_P1PDR	FCh	Port P1 direction register setting value
DEF_P2PDR	E0h	Port P2 direction register setting value
DEF_P3PDR	30h	Port P3 direction register setting value
DEF_P4PDR	00h	Port P4 direction register setting value
DEF_P5PDR	00h	Port P5 direction register setting value
DEF_P6PDR	00h	Port P6 direction register setting value
DEF_P7PDR	00h	Port P7 direction register setting value
DEF_P8PDR	18h	Port P8 direction register setting value
DEF_P9PDR	00h	Port P9 direction register setting value
DEF_PAPDR	C0h	Port PA direction register setting value
DEF_PBPDR	00h	Port PB direction register setting value
DEF_PDPDR	00h	Port PD direction register setting value
DEF_PEPDR	40h	Port PE direction register setting value

Table 3.8 Constants for 100-Pin, 16Kbytes RAM Capacity Products (RAM_SIZE = 16, PIN_SIZE = 100)

Constant Name	Setting Value	Contents
DEF_P0PDR	00h	Port P0 direction register setting value
DEF_P1PDR	00h	Port P1 direction register setting value
DEF_P2PDR	00h	Port P2 direction register setting value
DEF_P3PDR	00h	Port P3 direction register setting value
DEF_P4PDR	00h	Port P4 direction register setting value
DEF_P5PDR	00h	Port P5 direction register setting value
DEF_P6PDR	00h	Port P6 direction register setting value
DEF_P7PDR	00h	Port P7 direction register setting value
DEF_P8PDR	00h	Port P8 direction register setting value
DEF_P9PDR	00h	Port P9 direction register setting value
DEF_PAPDR	00h	Port PA direction register setting value
DEF_PBPDR	00h	Port PB direction register setting value
DEF_PDPDR	00h	Port PD direction register setting value
DEF_PEPDR	00h	Port PE direction register setting value

Table 3.9 Constants for 80-Pin, 16Kbytes RAM Capacity Products (RAM_SIZE = 16, PIN_SIZE = 80)

Constant Name	Setting Value	Contents
DEF_P0PDR	00h	Port P0 direction register setting value
DEF_P1PDR	00h	Port P1 direction register setting value
DEF_P2PDR	00h	Port P2 direction register setting value
DEF_P3PDR	0Ch	Port P3 direction register setting value
DEF_P4PDR	00h	Port P4 direction register setting value
DEF_P5PDR	00h	Port P5 direction register setting value
DEF_P6PDR	3Bh	Port P6 direction register setting value
DEF_P7PDR	00h	Port P7 direction register setting value
DEF_P8PDR	07h	Port P8 direction register setting value
DEF_P9PDR	00h	Port P9 direction register setting value
DEF_PAPDR	17h	Port PA direction register setting value
DEF_PBPDR	80h	Port PB direction register setting value
DEF_PDPDR	03h	Port PD direction register setting value
DEF_PEPDR	23h	Port PE direction register setting value

Table 3.10 Constants for 64-Pin, 16Kbytes RAM Capacity Products (RAM_SIZE = 16, PIN_SIZE = 64)

Constant Name	Setting Value	Contents
DEF_P0PDR	00h	Port P0 direction register setting value
DEF_P1PDR	01h	Port P1 direction register setting value
DEF_P2PDR	01h	Port P2 direction register setting value
DEF_P3PDR	0Ch	Port P3 direction register setting value
DEF_P4PDR	88h	Port P4 direction register setting value
DEF_P5PDR	20h	Port P5 direction register setting value
DEF_P6PDR	3Fh	Port P6 direction register setting value
DEF_P7PDR	00h	Port P7 direction register setting value
DEF_P8PDR	07h	Port P8 direction register setting value
DEF_P9PDR	00h	Port P9 direction register setting value
DEF_PAPDR	3Fh	Port PA direction register setting value
DEF_PBPDR	81h	Port PB direction register setting value
DEF_PDPDR	07h	Port PD direction register setting value
DEF_PEPDR	3Bh	Port PE direction register setting value

3.7 Functions

Table 3.11 lists the functions.

Table 3.11 Functions

Function Name	Outline
main	Main processing routine
R_INIT_StopModule	Disable peripheral modules running after a reset
R_INIT_Port_Initialize	Initial nonexistent port settings
R_INIT_Clock	Initial clock settings
cgc_oscillation_main	Main clock oscillation enable
cgc_oscillation_pll	PLL clock oscillation enable
cgc_oscillation_hoco	HOCO clock oscillation enable

3.8 Function Specifications

The following tables list the sample code function specifications.

main	
Outline	Main processing routine
Header	None
Declaration	void main(void)
Description	Calls the function to disable the peripheral module stop after a reset, the initial nonexistent port settings function, and the initial clock setting function.
Arguments	None
Return Value	None

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

R_INIT_Port_Initialize	
Outline	Initial nonexistent port settings
Header	r_init_port_initialize.h
Declaration	void R_INIT_Port_Initialize(void)
Description	Makes initial settings to the port direction registers corresponding to the pins of nonexistent port.
Arguments	None
Return Value	None
Remarks	The setting in the sample code (RAM_SIZE = 16, PIN_SIZE = 100) is for 100-pin, 16Kbytes RAM capacity products. When writing in byte units to PDR or PODR registers containing nonexistent ports after this function has been called, set the direction control bits corresponding to the nonexistent ports to values defined in user's manual and the port output data storage bits corresponding to the nonexistent ports to 0.

R_INIT_Clock	
Outline	Initial clock settings
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Makes initial clock settings.
Arguments	None
Return Value	None
Remarks	In the sample code processing is selected that sets the PLL clock as the system clock.

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

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

cgc_oscillation_hoco	
Outline	HOCO clock oscillation enable
Header	r_init_clock.h
Declaration	static void cgc_oscillation_hoco (void)
Description	Sets the HOCO frequency and sets the HOCOWTCR register, then starts oscillation of the HOCO clock. After this, waits for the HOCO oscillation stabilization waiting time to elapse.
Arguments	None
Return Value	None

3.9 Flowcharts

3.9.1 Main Processing

Figure 3.1 shows the main processing.

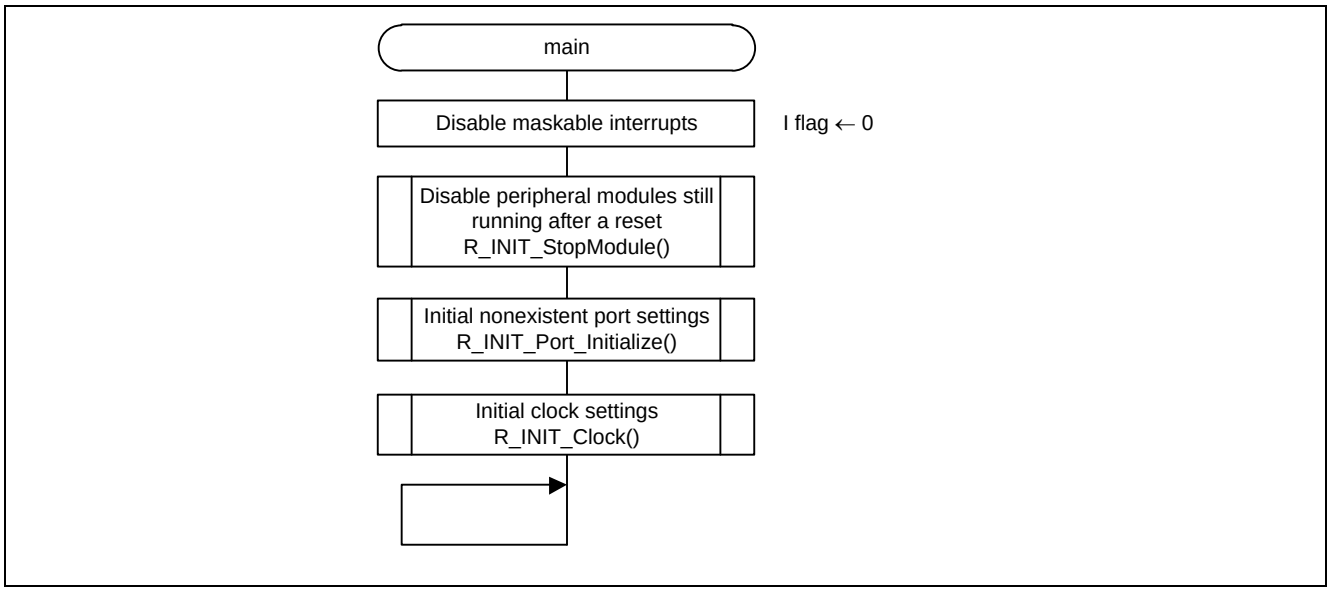
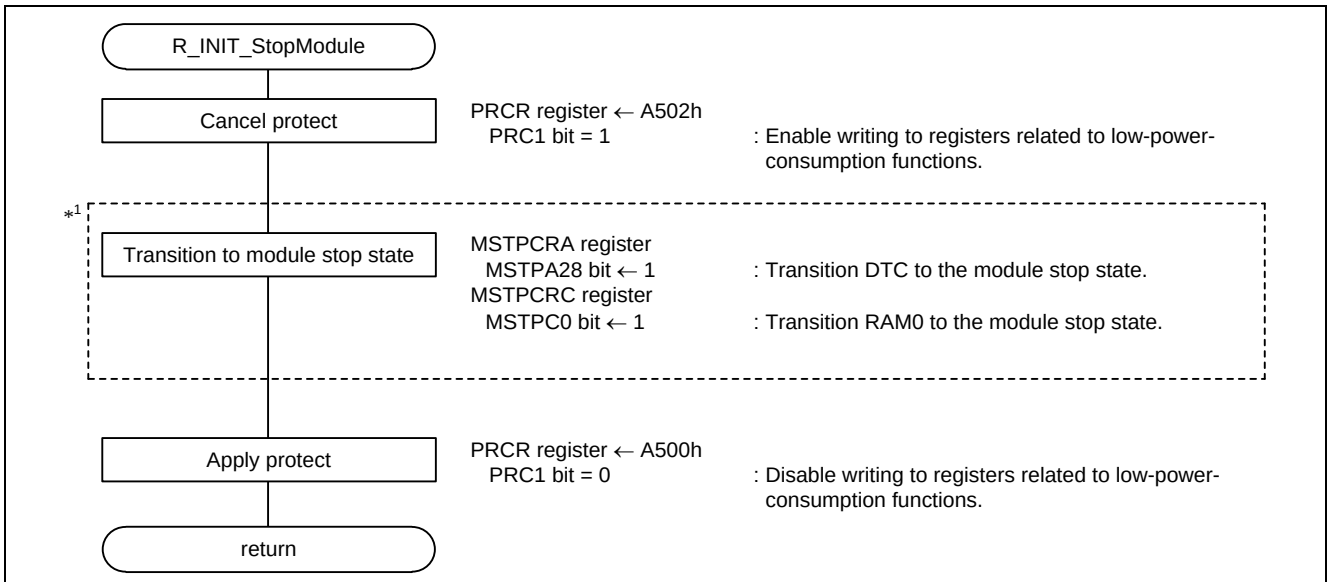


Figure 3.1 Main Processing

3.9.2 Disable Peripheral Module Running After a Reset

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

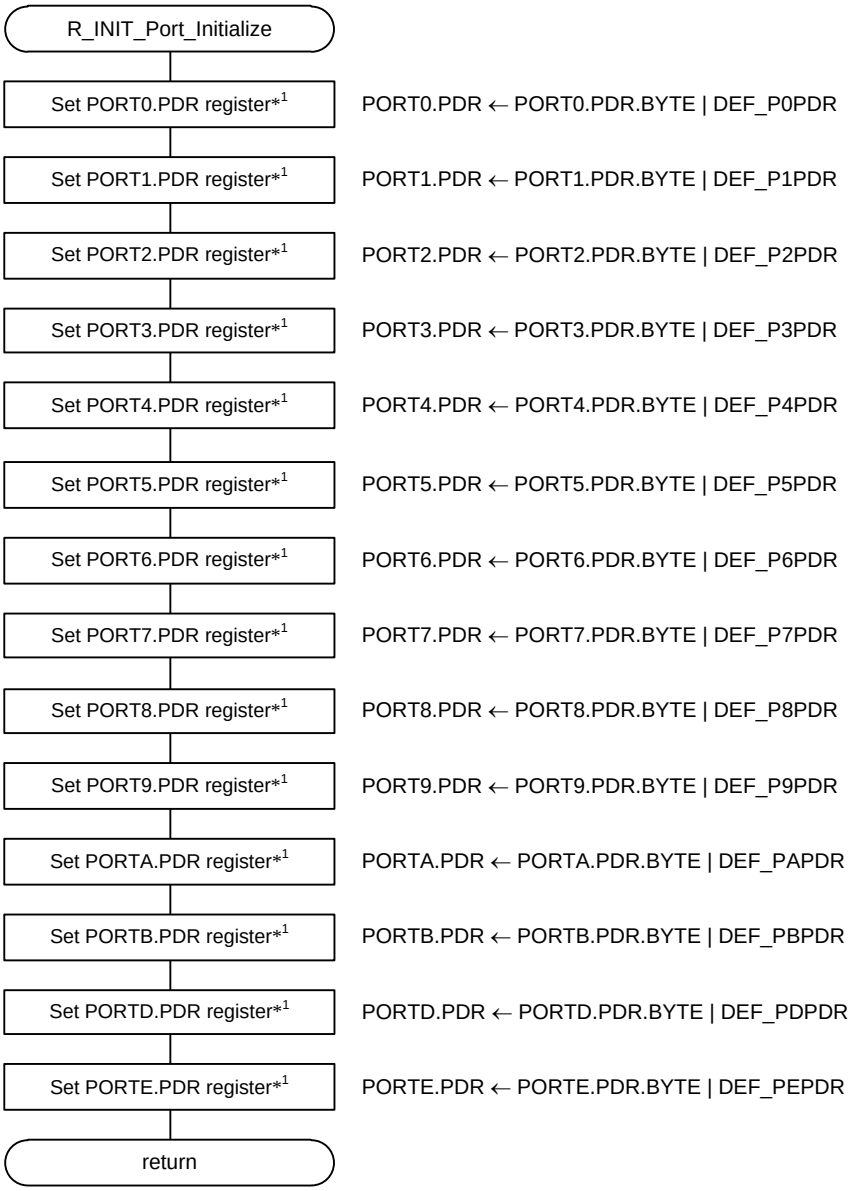


Note 1. In the sample code the module stop state is canceled. To transition to the module stop state, set the corresponding constant #define MSTP_STATE_<target module name> to 1.

Figure 3.2 Disable Peripheral Modules Still Running After a Reset

3.9.3 Initial Nonexistent Port Settings

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

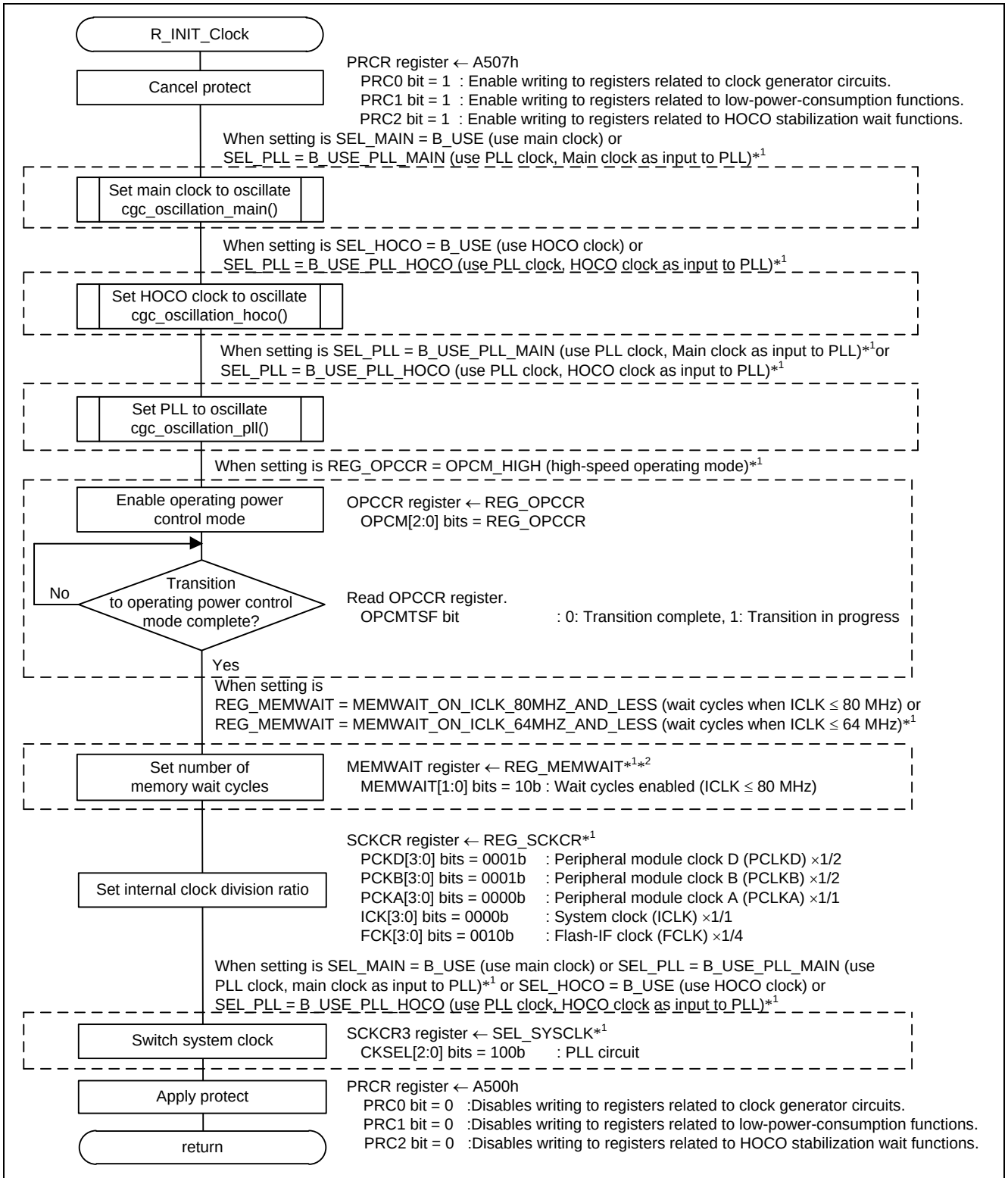


Note 1. Setting processing is not performed for registers where all pins exist.
(This processing is omitted when the source code is compiled.)

Figure 3.3 Initial Nonexistent Port Settings

3.9.4 Initial Clock Settings

Figure 3.4 is a flowchart of the processing for making initial clock settings.

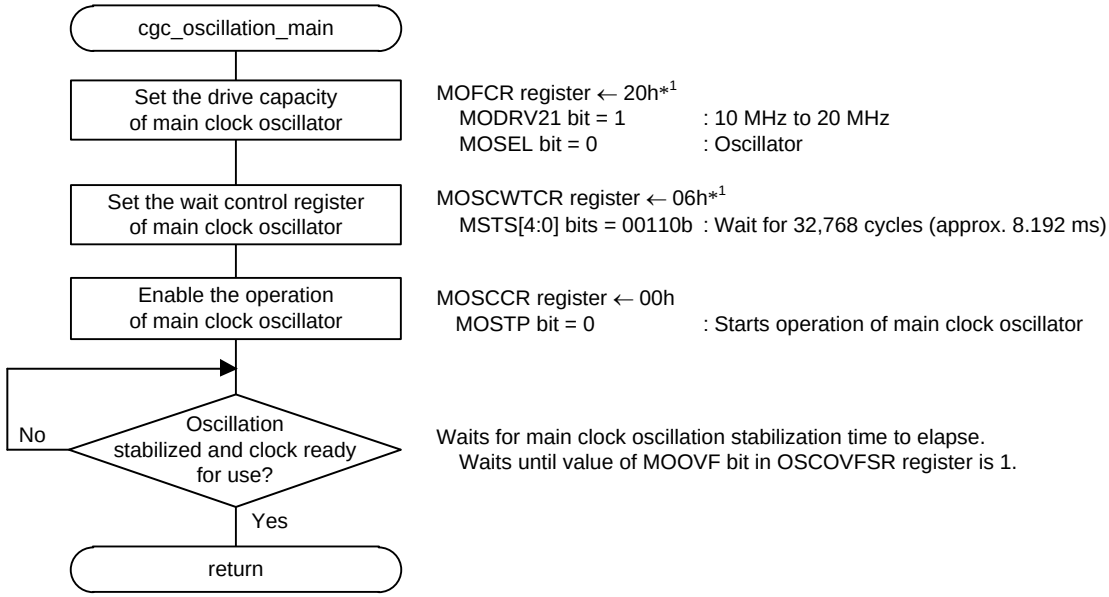


Note 1. The setting values differ depending on the value of constants.
2. After writing to the MEMWAIT register, while loop is used to confirm the value written to MEMWAIT register and to implement the wait process.

Figure 3.4 Initial Clock Settings

3.9.5 Main Clock Oscillation Enable

Figure 3.5 is a flowchart of the processing for starting oscillation of the main clock.

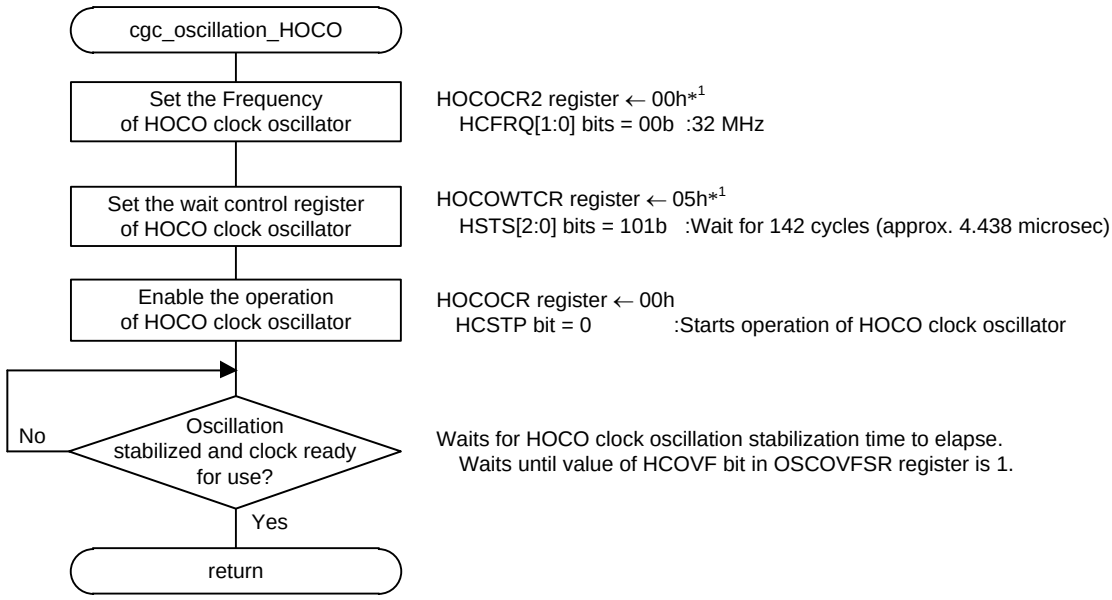


Note 1. Change the values of the relevant constants to match the characteristics of the target system.

Figure 3.5 Main Clock Oscillation Enable

3.9.6 HOCO Clock Oscillation Enable

Figure 3.6 is a flowchart of the processing for starting oscillation of the HOCO clock.

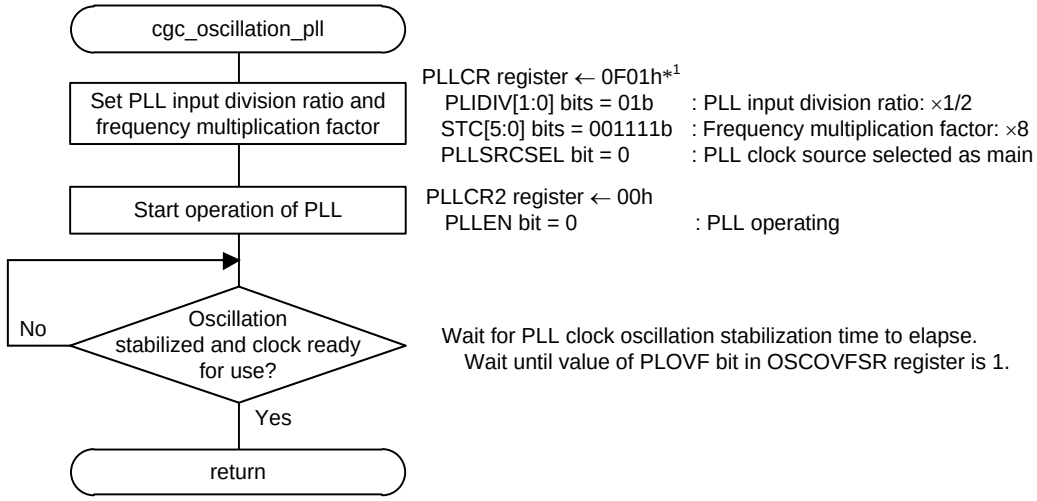


Note 1. Change the values of the relevant constants to match the characteristics of the target system.

Figure 3.6 HOCO Clock Oscillation Enable

3.9.7 PLL Clock Oscillation Enable

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



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

Figure 3.7 PLL Clock Oscillation Enable

4. Importing a Project

4.1 Importing a Project into e² studio

Follow the steps below to import your project into e² studio.

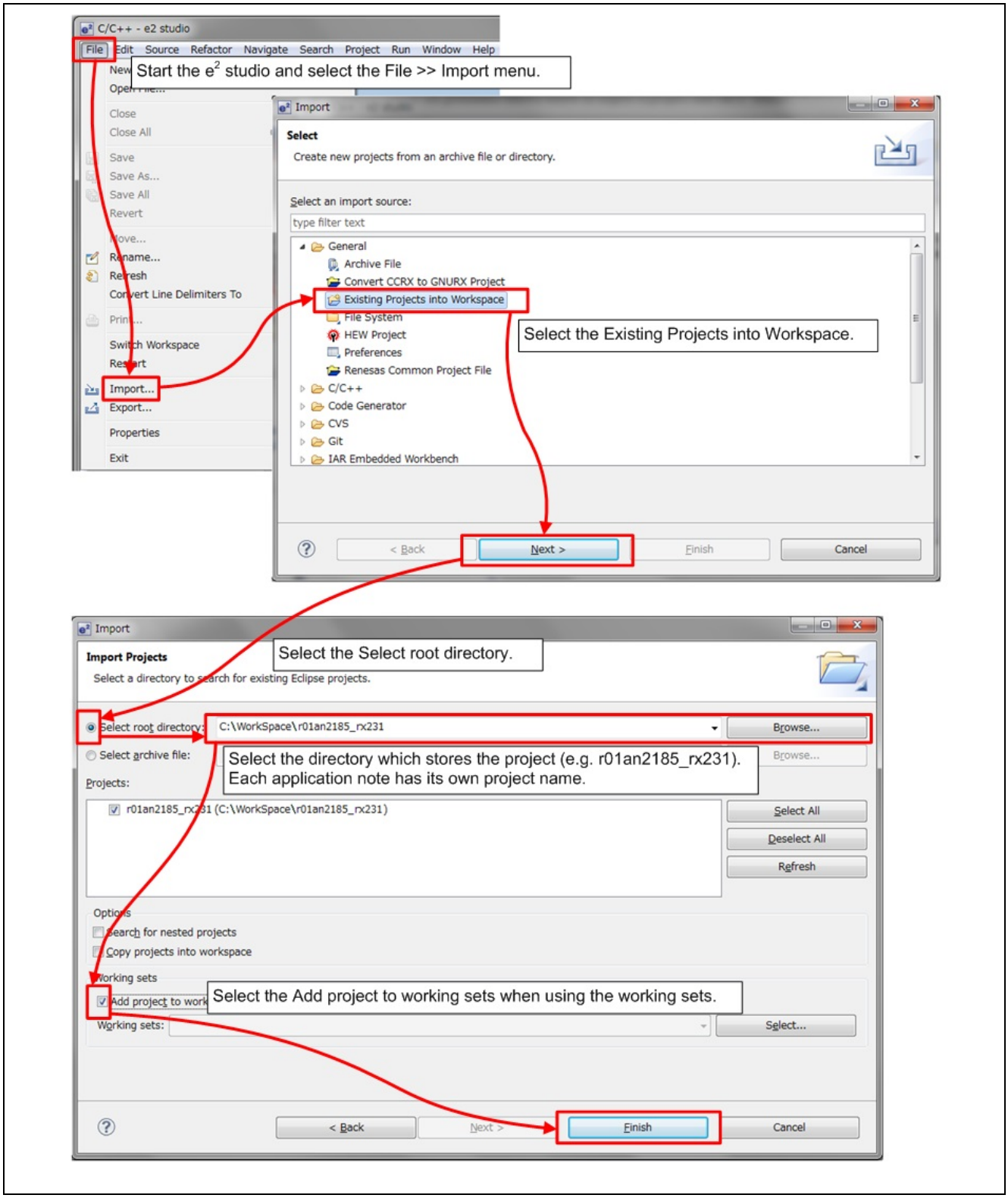


Figure 4.1 Importing a Project into e² studio

4.2 Importing a Project into CS+

Follow the steps below to import your project into CS+.

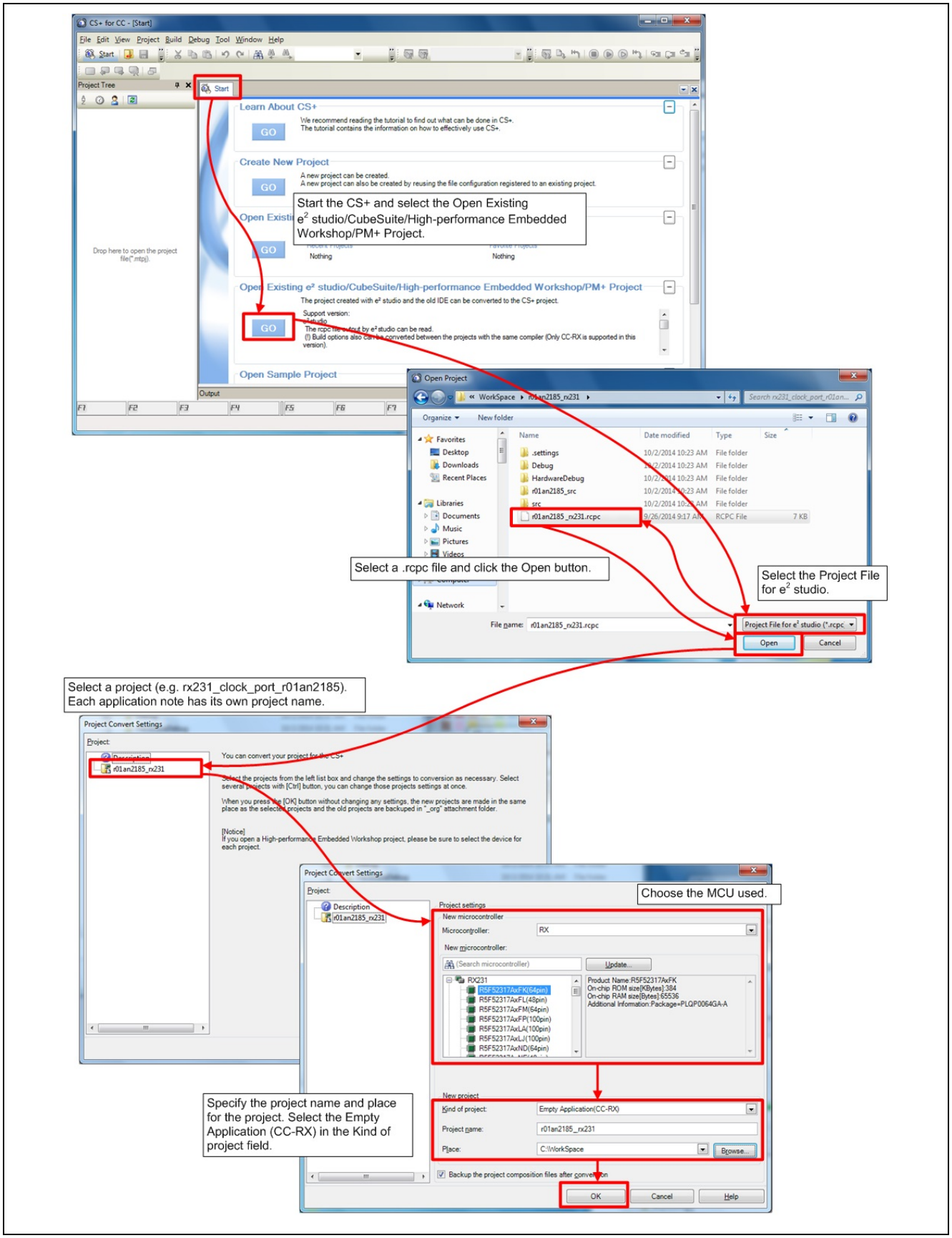


Figure 4.2 Importing a Project into CS+

5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

User's Manual: Hardware
RX24T Group User's Manual: Hardware (R01UH0576)
(The latest version can be downloaded from the Renesas Electronics website.)

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

User's Manual: Development Tools
[CS+][e^2 studio] RX C/C++ Compiler CC-RX User's Manual (R20UT3248)
(The latest version can be downloaded from the Renesas Electronics website.)

Website and Support

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

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

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Jan. 21, 2016	—	First edition issued
2.00	Feb. 28, 2017	—	512 KB, 64-pin package supported HOCO clock operation added
2.10	Aug.20.2020	—	Update the toolchain version. Changed the sample code settings to (RAM_SIZE = 16, PIN_SIZE = 100) 100-pin, 16Kbyte RAM capacity products.

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 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.
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; 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.
6. 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.
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. 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.
8. 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.
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 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.
10. 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.
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.

(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.4.0-1 November 2017)

Corporate Headquarters

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

Trademarks

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/.