RX13T Group

Initial Settings Example

Summary

This application note describes the settings that must be made after a reset of a RX13T Group microcontroller, including clock settings, disabling of peripheral functions still running after a reset, and nonexistent port settings.

Target Devices

- RX13T Group 48-pin version, ROM capacity: 64 KB and 128 KB
- RX13T Group 32-pin version, ROM capacity: 64 KB and 128 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 makes settings to disable peripheral functions still running after a reset, nonexistent port settings, and clock settings. The description in this application note applies to the processing that occurs following power-on (cold start).

1.1 Disabling Peripheral Functions Still Running After a Reset

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

DTC and RAM0

Note that the above processing is disabled in the sample code. If necessary, overwrite the corresponding constants to enable the processing.

1.2 Nonexistent Port Settings

It is necessary to set the direction control bits for nonexistent ports as specified in 17.4, Initialization of Port Direction Register (PDR), in RX13T Group User's Manual: Hardware. The sample code described in this application note makes initial settings appropriate for 48-pin products. Overwrite the constants as necessary to accommodate the actual target device.

1.3 Clock Selection

1.3.1 Overview

The procedure for making clock settings is as follows:

1. Main clock settings
2. PLL clock settings
3. HOCO 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 HOCO clock as the system clock. Overwrite the constants as necessary to change the clocks you wish to use.

1.3.2 Clock Specifications Assumed in Sample Code

Table 1.1 lists the clock specifications assumed in sample code.

Table 1.1 Clock Specifications Assumed in Sample Code

Clock	Oscillation Frequency	Oscillation Stabilization Time	Remarks
Main clock*1	8 MHz	4.2 ms*2	Crystal Not installed on RX13T CPU card (RTK0EMXA10C00000BJ)
PLL clock*1	32 MHz	—*3	—
HOCO clock	32 MHz	—*3	—

Notes: 1. Oscillation disabled by the sample code.
2. The actual oscillation stabilization time of the oscillator may differ due to conditions such as the system's wiring pattern and the oscillation constant. To determine the correct oscillation stabilization time, request an evaluation of the system you are actually using from the oscillator manufacturer.
3. Refer to the "Electrical Characteristics" section in RX13T Group User's Manual: Hardware.

1.3.3 Clock Selection Examples

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. Table 1.2 lists clock selection examples.

Table 1.2 Clock Selection Examples

No.		**1**[*1]	**2**	**3**
System clock		HOCO	PLL	Main clock
PLL clock		Stopped	Oscillating	Stopped
Main clock		Stopped	Oscillating	Oscillating
HOCO clock		Oscillating	Stopped	Stopped
Operating power control mode		High-speed operating mode	High-speed operating mode	Mid-speed operating mode
Constants[*2]	SEL_SYSCLK	CLK_HOCO	CLK_PLL	CLK_MAIN
	SEL_PLL	B_NOT_USE	B_USE	B_NOT_USE
	SEL_MAIN	B_NOT_USE	B_USE	B_USE
	SEL_HOCO	B_USE	B_NOT_USE	B_NOT_USE
	SEL_OPCM	OPCM_HIGH	OPCM_HIGH	OPCM_MID

Notes: 1. Default setting in sample code
2. For details of constants, refer to the description of the constants used by the sample code.

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		R5F513T5ADFL (RX13T Group)
Operating frequency	HOCO clock selected as system clock (No. 1 in Table 1.2)	• HOCO: 32 MHz • System clock (ICLK): 32 MHz (HOCO × 1/1) • Peripheral module clock B (PCLKB): 32 MHz (HOCO × 1/1) • Peripheral module clock D (PCLKD): 32 MHz (HOCO × 1/1) • FlashIF clock (FCLK): 32 MHz (HOCO × 1/1)
	PLL clock selected as system clock (No. 2 in Table 1.2)	• Main clock: 8 MHz • PLL: 32 MHz (main clock × 1/2 × 8) • System clock (ICLK): 32 MHz (PLL × 1/1) • Peripheral module clock B (PCLKB): 32 MHz (PLL × 1/1) • Peripheral module clock D (PCLKD): 32 MHz (PLL × 1/1) • FlashIF clock (FCLK): 32 MHz (PLL × 1/1)
	Main clock selected as system clock (No. 3 in Table 1.2)	• Main clock: 8 MHz • System clock (ICLK): 8 MHz (main clock × 1/1) • Peripheral module clock B (PCLKB): 8 MHz (main clock × 1/1) • Peripheral module clock D (PCLKD): 8 MHz (main clock × 1/1) • FlashIF clock (FCLK): 8 MHz (main clock × 1/1)
Operating voltage		3.3 V
Integrated development environment		Renesas Electronics e^2 studio Version: 7.5.0
C compiler		Renesas Electronics C/C++ Compiler Package for RX Family V.3.01
		Compiler option The integrated development environment default settings are used.
iodefine.h version		Version 1.00A
Endian order		Little endian or big endian
Operating mode		Single-chip mode
Processor mode		Supervisor mode
Sample code version		Version 1.00
Board used		RX13T CPU card (Product No. RTK0EMXA10C00000BJ)

3. Software

3.1 Disabling Peripheral Functions Still Running After a Reset

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

To transition a module to the module stop state after a reset is canceled, set the corresponding module stop bit to 1 (transition to module stop state). Putting modules into the module stop state can reduce the power consumption of the device.

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

Table 3.1 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	MATPCRC.MSTPC0 bit	0 (RAM0 Operating)	1 (RAM0 Stopped)

3.2 Nonexistent Port Settings

3.2.1 Processing Overview

The sample code sets the bits in the PDR registers corresponding to nonexistent ports to 0 (input) or 1 (output). The setting values are as described in 17.4, Initialization of Port Direction Register (PDR), in RX13T Group User's Manual: Hardware. When writing in byte units to PDR registers containing nonexistent ports after this function has been called, set the direction control bits corresponding to the nonexistent ports as described in 17.4, Initialization of Port Direction Register (PDR), in RX13T Group User's Manual: Hardware. In addition, when writing in byte units to the PODR registers, set the output data storage bits corresponding to nonexistent ports to 0.

3.2.2 Pin Count Setting

The setting in the sample code (PIN_SIZE=48) is for 48-pin products. The other pin count supported by the sample code is 32. When using a product with a pin count other than 48, change the value of PIN_SIZE in r_init_port_initialize.h to match the target device.

Table 3.2 Nonexistent Ports

Port Symbol	48-Pin Products	Pins	32-Pin Products	Pins
PORT1	P12 to P17	6	P10, P12 to P17	7
PORT2	P20, P21, P25 to P27	5	P20 to P27	8
PORT3	P30 to P35	6	P30 to P35	6
PORT4	—	—	P45 to P47	3
PORT7	P77	1	P70, P77	2
PORT9	P90 to P92, P95 to P97	6	P90 to P92, P95 to P97	6
PORTA	PA0, PA1, PA4 to PA7	6	PA0 to PA7	8
PORTB	—	—	PB4, PB5	2
PORTD	PD0 to PD2, PD7	4	PD0 to PD7	8
PORTE	PE0, PE1, PE3 to PE7	7	PE0, PE1, PE3 to PE7	7

3.3 Clock Settings

3.3.1 Clock Setting Procedure

Table 3.3 lists the steps in the clock setting procedure, the processing performed in each step, and the settings made by the sample code. The sample code turns on the HOCO clock and turns off the main clock and PLL clock.

Table 3.3 Clock Setting Procedure

Step	Processing	Details of Processing		Sample Code Settings
1	Main clock setting*2	Not used	No settings required.	The main clock is not used.
		Used	Sets the main clock drive capacity and sets in MOSCWTCR the waiting time until output of the main clock to the internal clocks starts, then starts oscillation by the main clock. After this, waits for the clock oscillation stabilization waiting time*1.	
2	HOCO clock setting*2	Not used	No settings required.	The HOCO clock is used.
		Used	Starts oscillation by the HOCO clock. After this, waits for the clock oscillation stabilization waiting time*1.	
3	PLL clock setting*2	Not used	No settings required.	The PLL clock is not used.
		Used	Sets the PLL input division ratio and frequency multiplication factor, then starts oscillation by the PLL clock. After this, waits for the clock oscillation stabilization waiting time*1.	
4	Clock division ratio settings	Changes the internal clock division ratios.		ICLK: × 1/1 PCLKB: × 1/1 PCLKD: × 1/1 FCLK: × 1/1
5	System clock switching	Switches according to the system used.		Switches to HOCO clock.

Notes: 1. Confirms that the appropriate bit in the oscillation stabilization flag register is set to 1.
2. Change the values of the constants in r_init_clock.h as necessary to match the selection of the clocks you wish to use or not use.

3.4 Section Composition

Table 3.4 lists section information changed in the sample code. For instructions for adding, changing, and deleting sections, refer to the latest version of RX Family: CC-RX Compiler User's Manual.

Table 3.4 Changes to Section Information in Sample Code

Section Name	Change	Address	Description
End_of_RAM	Added	0000 2FFCh	On-chip RAM end address

3.5 File Composition

Table 3.5 lists the files used in the sample code. Files generated by the integrated development environment are not included in this table.

Table 3.5 Files Used in the Sample Code

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

3.6 Option-Setting Memory

Table 3.6 lists the option-setting memory configured in the sample code. When necessary, set a value suited to the user system.

Table 3.6 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

3.7 Constants

Table 3.7 to Table 3.10 list the constants used by the sample code.

Table 3.7 Constants (User Changeable) Used by Sample Code

Constant Name	Setting Value	Description
SEL_MAIN*1	B_NOT_USE	Main clock enable/disable selection B_USE: Used B_NOT_USE: Not used
MAIN_CLOCK_HZ*1	8000000L	Main clock oscillator frequency (Hz)
REG_MOFCR*1	00h	Main clock oscillator drive capacity setting (setting value of MOFCR register)
REG_MOSCWTCR*1	06h	Main clock oscillator stabilization wait time setting (setting value of MOSCWTCR register)
SEL_HOCO*1	B_USE	HOCO clock enable/disable selection B_USE: Used B_NOT_USE: Not used
SEL_PLL*1	B_NOT_USE	PLL clock enable/disable selection B_USE: Used B_NOT_USE: Not used
REG_PLLCR*1	0F01h	PLL input division ratio and frequency multiplication factor settings
SEL_SYSCLK*1	CLK_HOCO	Clock source selection for the system clock CLK_PLL: PLL clock CLK_MAIN: Main clock CLK_HOCO: HOCO clock
REG_SCKCR*1	0000 0000h	Internal clock division ratio settings
SEL_OPCM*1	OPCM_HIGH	Operating power control mode selection*4 OPCM_HIGH: High-speed operating mode OPCM_MID: Mid-speed operating mode
MSTP_STATE_DTC*2	MODULE_STOP_DISABLE	DTC module stop state selection MODULE_STOP_DISABLE: Cancel MODULE_STOP_ENABLE: Transition
MSTP_STATE_RAM0*2	MODULE_STOP_DISABLE	On-chip RAM module stop state selection MODULE_STOP_DISABLE: Operating MODULE_STOP_ENABLE: Stopped
PIN_SIZE*3	48	Pin count of product used

Notes: 1. Change the settings values in r_init_clock.h to match the target system.
2. Change the settings values in r_init_stop_module.h to match the target system.
3. Change the settings values in r_init_port_initialize.h to match the target system.
4. The operating frequency range and operating voltage range differ depending on the operating mode. For details, refer to RX13T Group User's Manual: Hardware.

Table 3.8 Constants (Non User Changeable) Used by Sample Code

Constant Name	Setting Value	Description
B_NOT_USE	0	Not used
B_USE	1	Used
CLK_PLL	0400h	Clock source: PLL
CLK_MAIN	0200h	Clock source: Main clock
CLK_HOCO	0100h	Clock source: HOCO
OPCM_MID	02h	Operating power control mode: Mid-speed operating mode
OPCM_HIGH	00h	Operating power control mode: High-speed operating mode
OPCM_DEFAULT	OPCM_MID	Operating mode after reset cancellation
MODULE_STOP_ENABLE	1	Transition to module stop state
MODULE_STOP_DISABLE	0	Module stop state canceled

Table 3.9 Constants for 48-Pin Products (PIN_SIZE=48)

Constant Name	Setting Value	Description
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_P7PDR	00h	Port P7 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

Table 3.10 Constants for 32-Pin Products (PIN_SIZE=32)

Constant Name	Setting Value	Description
DEF_P1PDR	01h	Port P1 direction register setting value
DEF_P2PDR	1Ch	Port P2 direction register setting value
DEF_P3PDR	00h	Port P3 direction register setting value
DEF_P4PDR	E0h	Port P4 direction register setting value
DEF_P7PDR	01h	Port P7 direction register setting value
DEF_P9PDR	00h	Port P9 direction register setting value
DEF_PAPDR	00h	Port PA direction register setting value
DEF_PBPDR	30h	Port PB direction register setting value
DEF_PDPDR	78h	Port PD direction register setting value

3.8 Functions

Table 3.11 lists the functions of the sample code.

Table 3.11 Functions

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

3.9 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 halt peripheral functions still running after a reset, the initial nonexistent port settings function, and the initial clock settings function.
Arguments	None
Return Value	None

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

R_INIT_Port_Initialize	
Outline	Initial nonexistent port settings
Header	r_init_port_initialize.h
Declaration	void R_INIT_Port_Initialize (void)
Description	Makes initial settings to the port direction registers corresponding to the pins of nonexistent port.
Arguments	None
Return Value	None
Remarks	The setting in the sample code (PIN_SIZE=48) is for 48-pin products. When writing in byte units to PDR or PODR registers containing nonexistent ports after this function has been called, set the direction control bits corresponding to the nonexistent ports as described in 17.4, Initialization of Port Direction Register (PDR), in RX13T Group User's Manual: Hardware. Also, set the port output data storage bits corresponding to ports set to output to 0.

R_INIT_Clock	
Outline	Initial clock settings
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Makes initial clock settings and sets the power control mode.
Arguments	None
Return Value	None
Remarks	The sample code processing sets the HOCO as the system clock and selects high-speed operating mode as the power control mode.

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.
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.
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	Starts oscillation of the HOCO. After this, waits for the HOCO oscillation stabilization waiting time.
Arguments	None
Return Value	None

3.10 Flowcharts

3.10.1 Main Processing

Figure 3.1 shows the main processing.

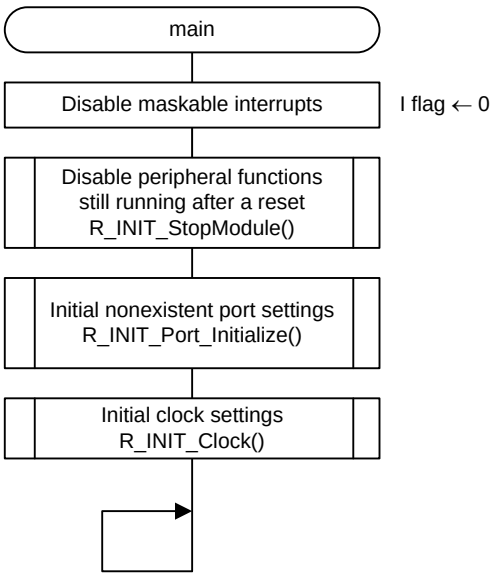
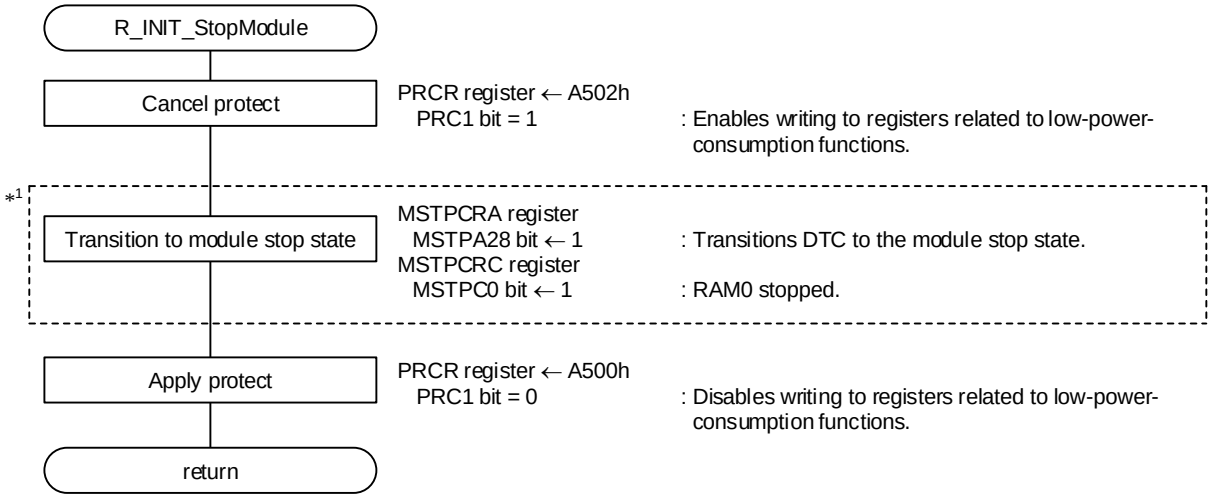


Figure 3.1 Main Processing

3.10.2 Disable Peripheral Functions Still Running After a Reset

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



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

Figure 3.2 Disable Peripheral Functions Still Running After a Reset

3.10.3 Initial Nonexistent Port Settings

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

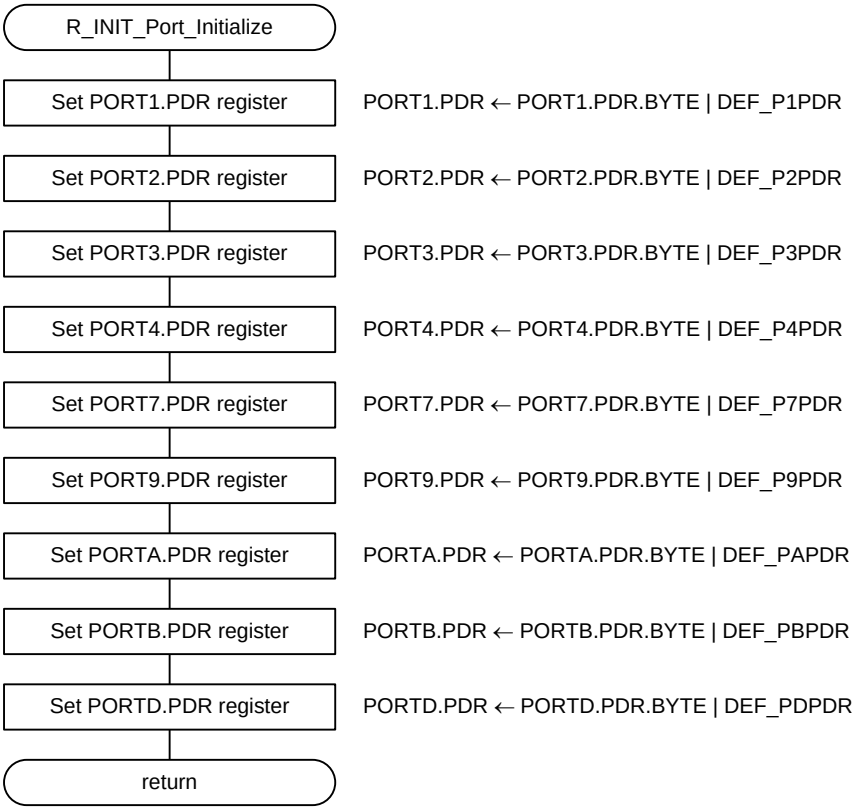
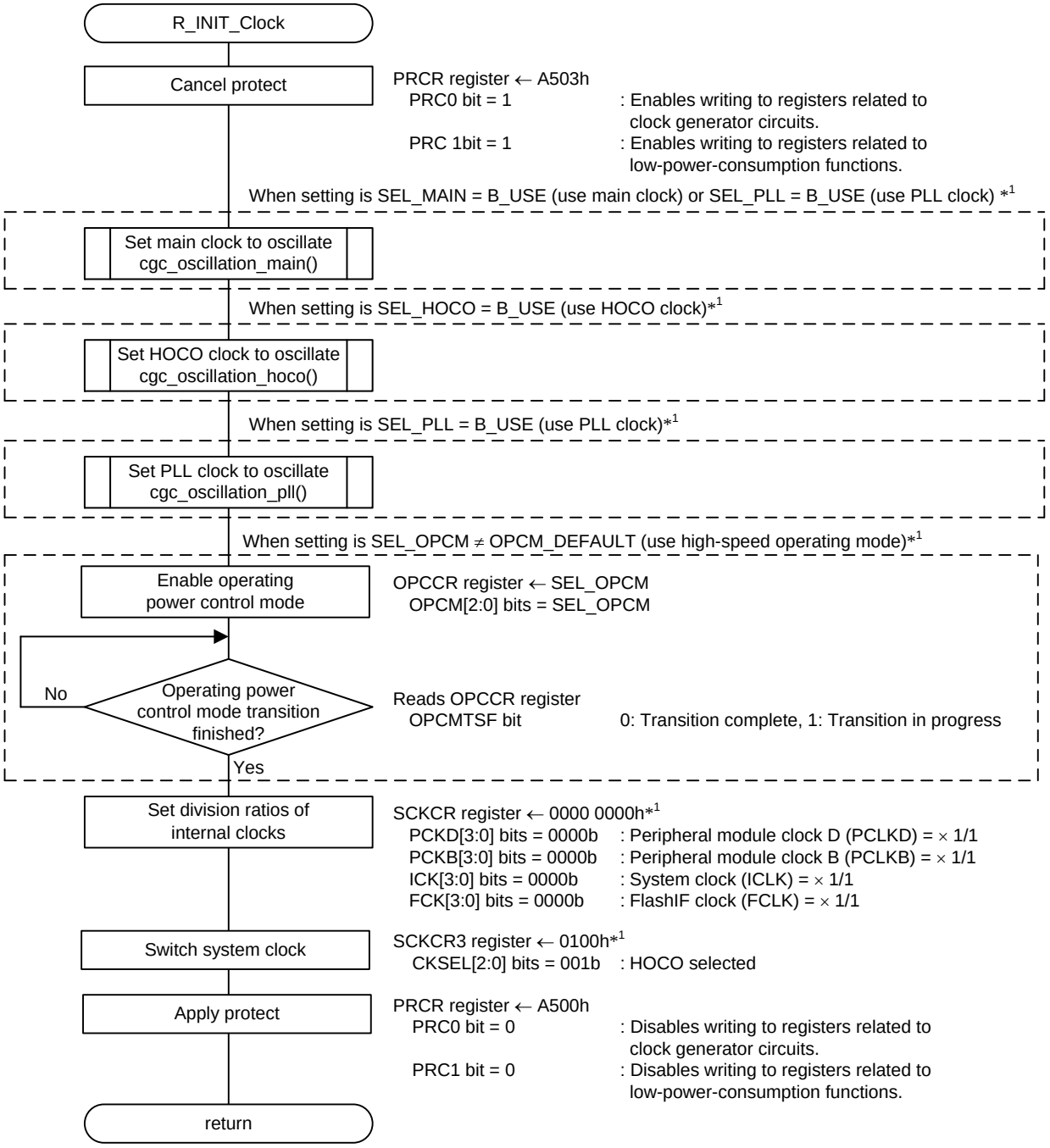


Figure 3.3 Initial Nonexistent Port Settings

3.10.4 Initial Clock Settings

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

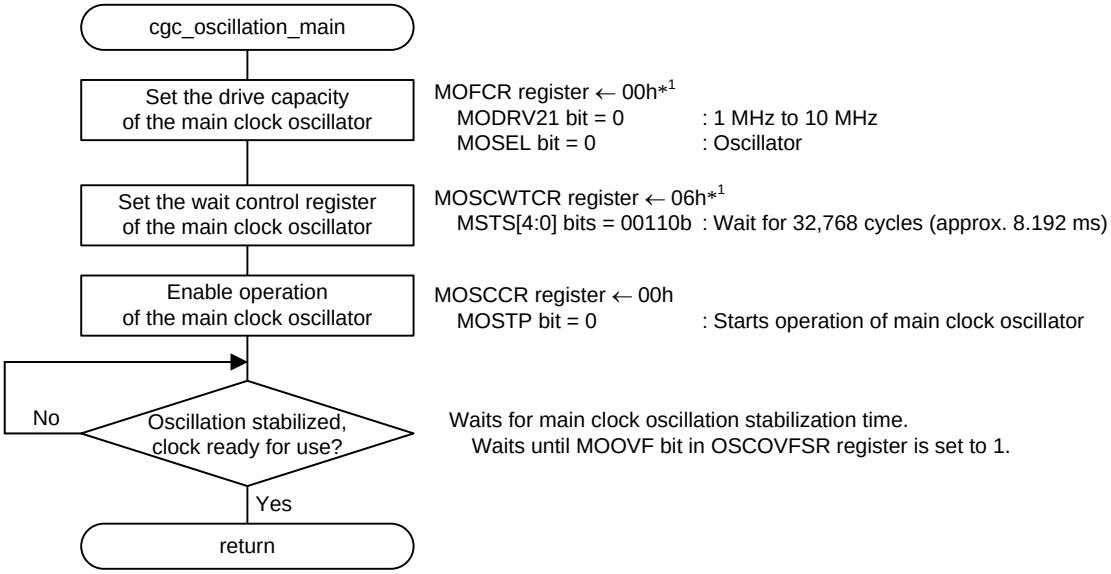


Note: 1. Change the values of the relevant constants to match the characteristics of the target system.

Figure 3.4 Initial Clock Settings

3.10.5 Main Clock Oscillation Enable

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

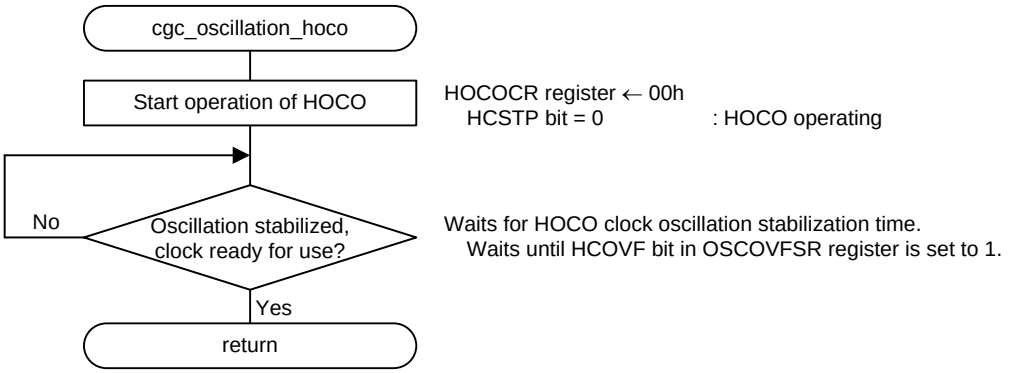


Note: 1. Change the values of the relevant constants to match the characteristics of the target system.

Figure 3.5 Main Clock Oscillation Enable

3.10.6 HOCO Clock Oscillation Enable

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

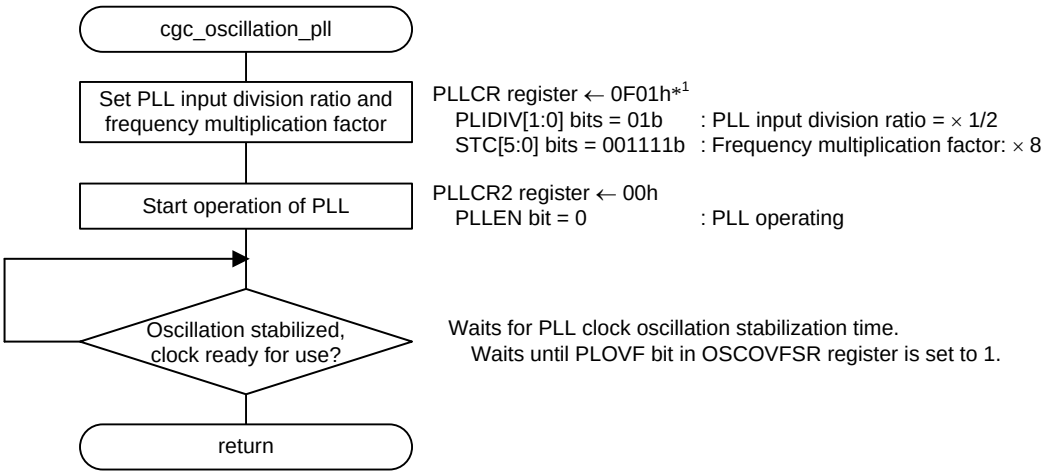


Note: 1. Change the values of the relevant constants to match the characteristics of the target system.

Figure 3.6 HOCO Clock Oscillation Enable

3.10.7 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

The sample code is provided in e^2 studio project format. The procedures for importing the project into e^2 studio and into CS+ are presented below. After importing the project, check the build and debug settings.

4.1 Importing a Project in the e^2 studio

Follow the steps below to import your project into e^2 studio. Pictures may be different depending on the version of e^2 studio to be used.

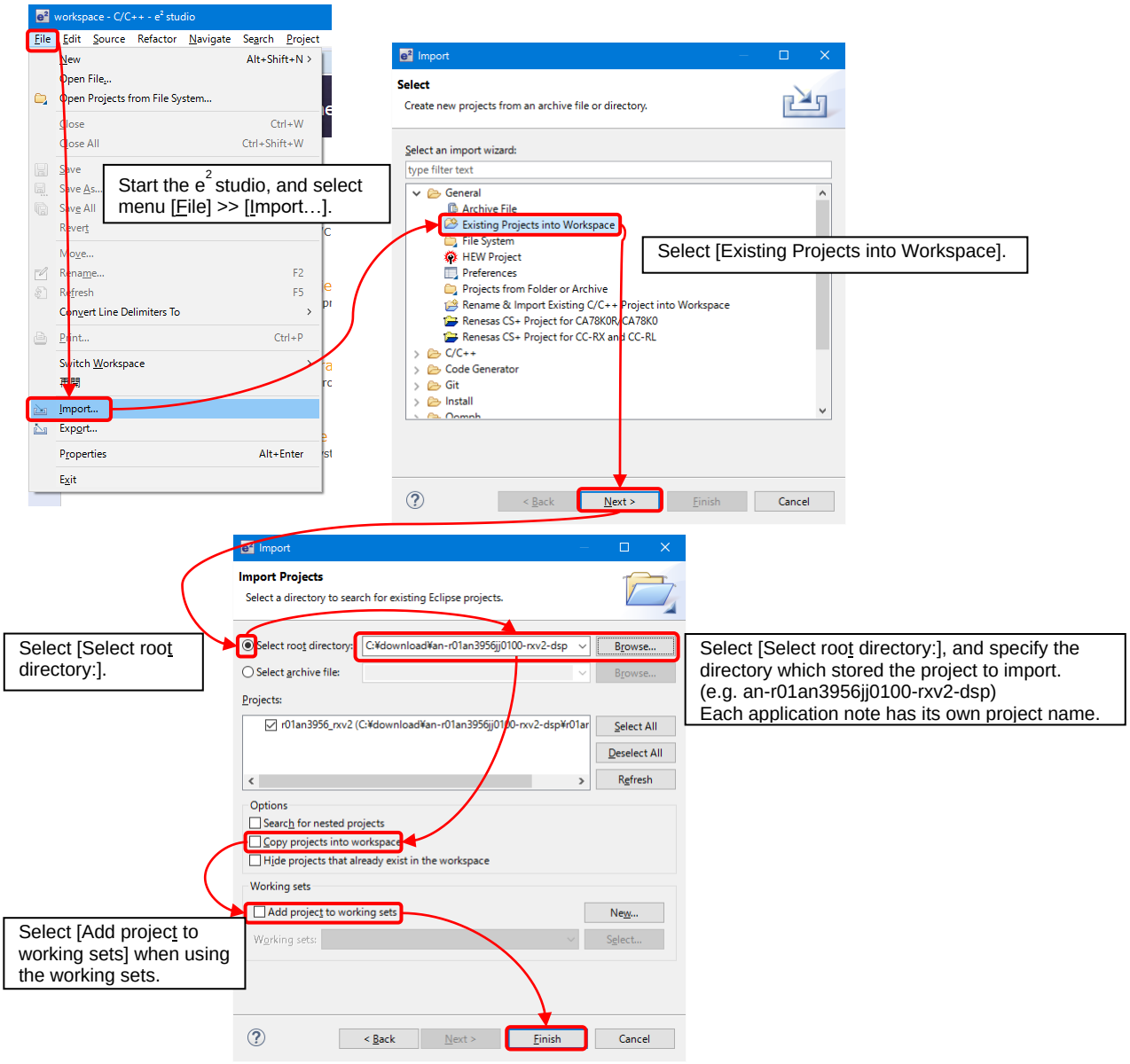


Figure 4.1 Importing a Project in the e^2 studio

4.2 Importing a Project in CS+

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

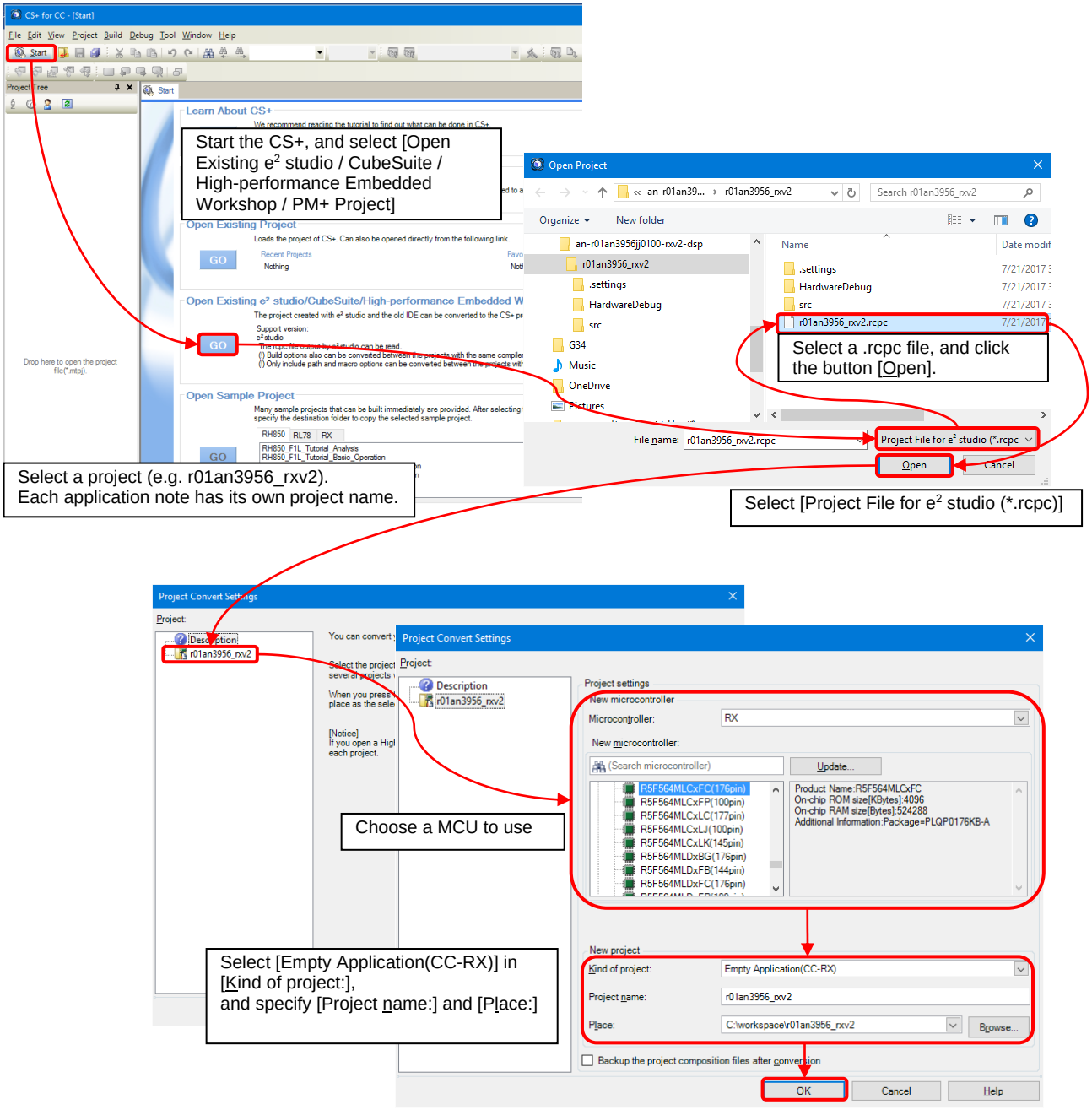


Figure 4.2 Importing a Project in CS+

5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

User's Manual: Hardware
RX13T Group User's Manual: Hardware (R01UH0822)
(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 CC-RX Compiler User's Manual (R20UT3248)
(The latest version can be downloaded from the Renesas Electronics website.)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Nov. 01, 2019	—	First edition issued

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

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

Contact information

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