# RX24U Group, RX24T Group

## Example of Setting Three-Phase Complementary PWM Output Duty Ratio to 0% or 100% Using MTU


### Summary

The RX24U Group and RX24T Group each incorporate a multi-function timer pulse unit 3 (MTU3d) module that can be used to generate pulse width modulation (PWM) waveforms.

This application note describes a method of producing three-phase complementary PWM output with a duty ratio of 0% or 100%.


### Target Devices

RX24U Group and RX24T Group

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

## Contents

# 1. Introduction

## 1.1 Generating Complementary PWM Waveforms Using the MTU

It is possible to use the complementary PWM mode of the MTU to generate complementary PWM waveforms with a dead time setting.

Figure 1.1 shows an example of complementary PWM waveform generation using the MTU. A PWM waveform is generated by compare matches between the compare register or temporary register and the counter register. During the Ta interval the temporary register operates as a register for updating the compare register, and the compare register generates PWM waveforms by compare matches with the red lines. In the Tb1 and Tb2 intervals the temporary register is compared with the blue lines, the compare register is compared with the red lines, and PWM waveforms are generated by the compare matches of each. The compare matches that generate the PWM waveforms are designated as *a*, *b*, *c*, *d*, *a'*, and *b'*. Of these, *a* and *a'* are compare matches that turn the negative-phase output off, *b* and *b'* are compare matches that turn the positive-phase output on, *c* is a compare match that turns the positive-phase output off, and *d* is a compare match that turns the negative-phase output on.

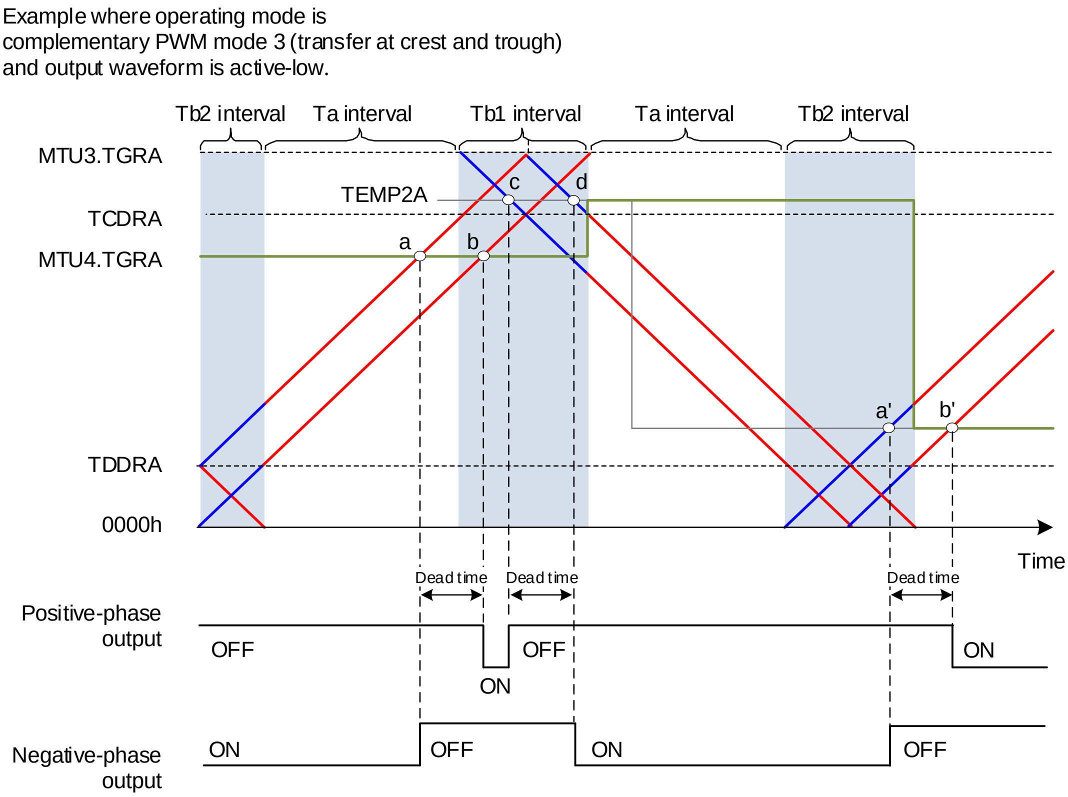



**Figure 1.1 Example of Complementary PWM Waveform Generation Using MTU**

For details of complementary PWM mode, refer to 20.3.8, Complementary PWM Mode, in RX24U Group User's Manual: Hardware or RX24T Group User's Manual: Hardware.

# 2. Specifications

## 2.1 Overview of Sample Code

The sample code uses the MTU's complementary PWM mode 2 to output a waveform with a positive-phase duty ratio of 0% in the first cycle, a waveform with a positive-phase duty ratio of 100% in the second cycle, a waveform with a duty ratio of 0% in the third cycle, and continues to repeat the same operations thereafter.

The MTU settings used in the sample code are as follows:

- Complementary PWM mode 2 is used.
- Channels MTU3 and MTU4 are used.
- The PWM output signal is active-low.
- The dead time is set to 30 µs.
- PWM carrier period is set to 1 ms.
- The same duty ratio settings are used for all three phases.

Table 2.1 lists the peripheral functions used and their applications.

**Table 2.1 Peripheral Functions Used and Their Applications**

| Peripheral Function | Application |
|---|---|
| MTU | Generates three-phase complementary PWM waveforms. |
| POE | Sets the PWM output pins to the Hi-Z state when making MPC settings. |
| MPC | Changes the setting of the pins used from general I/O port to peripheral function I/O port. |

Figure 2.1 shows the three-phase complementary PWM output produced by the sample code.

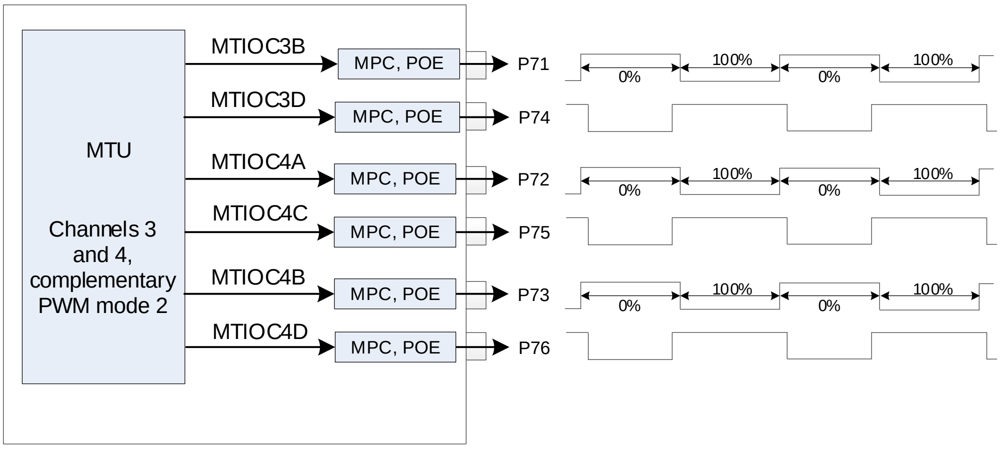



**Figure 2.1 Three-Phase Complementary PWM Output Produced by Sample Code**

## 3. Operation Confirmation Conditions

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

**Table 3.1 Operation Confirmation Conditions**

| Item | Description |
|---|---|
| MCU used | R5F524TEADFP (RX24T Group) |
| Operating frequency | Main clock: 20.0 MHz<br>PLL: 80.0 MHz (main clock ×1/2 ×8)<br>HOCO: Stopped<br>LOCO: 4 MHz<br>System clock (ICLK): 80.0 MHz (PLL ×1/1)<br>Peripheral module clock A (PCLKA): 80.0 MHz (PLL ×1/1)<br>Peripheral module clock B (PCLKB): 40.0 MHz (PLL ×1/2)<br>Peripheral module clock D (PCLKD): 40.0 MHz (PLL ×1/2)<br>FlashIF clock (FCLK): 20.0 MHz (PLL ×1/4) |
| Operating voltage | 3.3 V |
| Integrated development environment | Renesas Electronics<br>$e^2$ studio Version 7.6.0 |
| C compiler | Renesas Electronics<br>C/C++ Compiler Package for RX Family V3.01.00 |
| | Compiler option<br>The integrated development environment default settings are used. |
| iodefine.h version | V1.0H |
| Endian | Little endian, big endian |
| Operating mode | Single-chip mode |
| Processor mode | Supervisor mode |
| Sample code version | Version 1.00 |
| Board used | Renesas Starter Kit for RX24T (product No.: RTK500524TSxxxxxBE) |

# 4. Hardware

## 4.1 Pins Used

Table 4.1 lists the pins used and their functions.

**Table 4.1 Pins Used and Their Functions**

| Pin Name | I/O | Function |
|---|---|---|
| P71/MTIOC3B | Output | Positive-phase complementary PWM output pin 1 |
| P72/MTIOC4A | Output | Positive-phase complementary PWM output pin 2 |
| P73/MTIOC4B | Output | Positive-phase complementary PWM output pin 3 |
| P74/MTIOC3D | Output | Negative-phase complementary PWM output pin 1 |
| P75/MTIOC4C | Output | Negative-phase complementary PWM output pin 2 |
| P76/MTIOC4D | Output | Negative-phase complementary PWM output pin 3 |

# 5. Software

## 5.1 Sample Code Operation

Figure 5.1 shows the operation of the sample code for a single phase. The MTU3.TGRA compare match interrupt (TGIA3) is set to generate an interrupt to change the duty ratio setting, and the buffer register is overwritten in the TGIA3 interrupt handling routine in order to update the compare register to the MTU3.TGRA value (to set the duty ratio to 0% in the next cycle) or to 0000h (to set the duty ratio to 100% in the next cycle). The buffer register value is transferred to the temporary register in the Ta interval, and the temporary register value is transferred to the counter register at the end of the Tb2 interval. The initial output is turned off for both the positive and negative phases, according to the settings of the TOCR1A.OLSP and TOCR1A.OLSN bits, and the negative-phase output turns on when the dead time elapses following the start of count operation. Thereafter, steps 1 to 5 below take place repeatedly.

1. At *c* a compare match between the compare register and counter register occurs and the positive-phase output turns off, but the waveform does not change because the output goes from off to off.
2. At *a* and *d* compare matches occur that turn the negative-phase output on and off, respectively, but the waveform does not change because when turn-on and turn-off compare matches occur at the same time they are both ignored.
3. At *b*, which occurs in the same interval as turn-off compare match *c*, the waveform does not change because compare match *b* is ignored.
4. At *a’* and *b’* compare matches occur between the temporary register and counter register, causing the negative-phase output to turn off and the positive-phase output to turn on.
5. At *c’* and *d’* compare matches occur between the compare register and counter register, causing the positive-phase output to turn off and the negative-phase output to turn on.

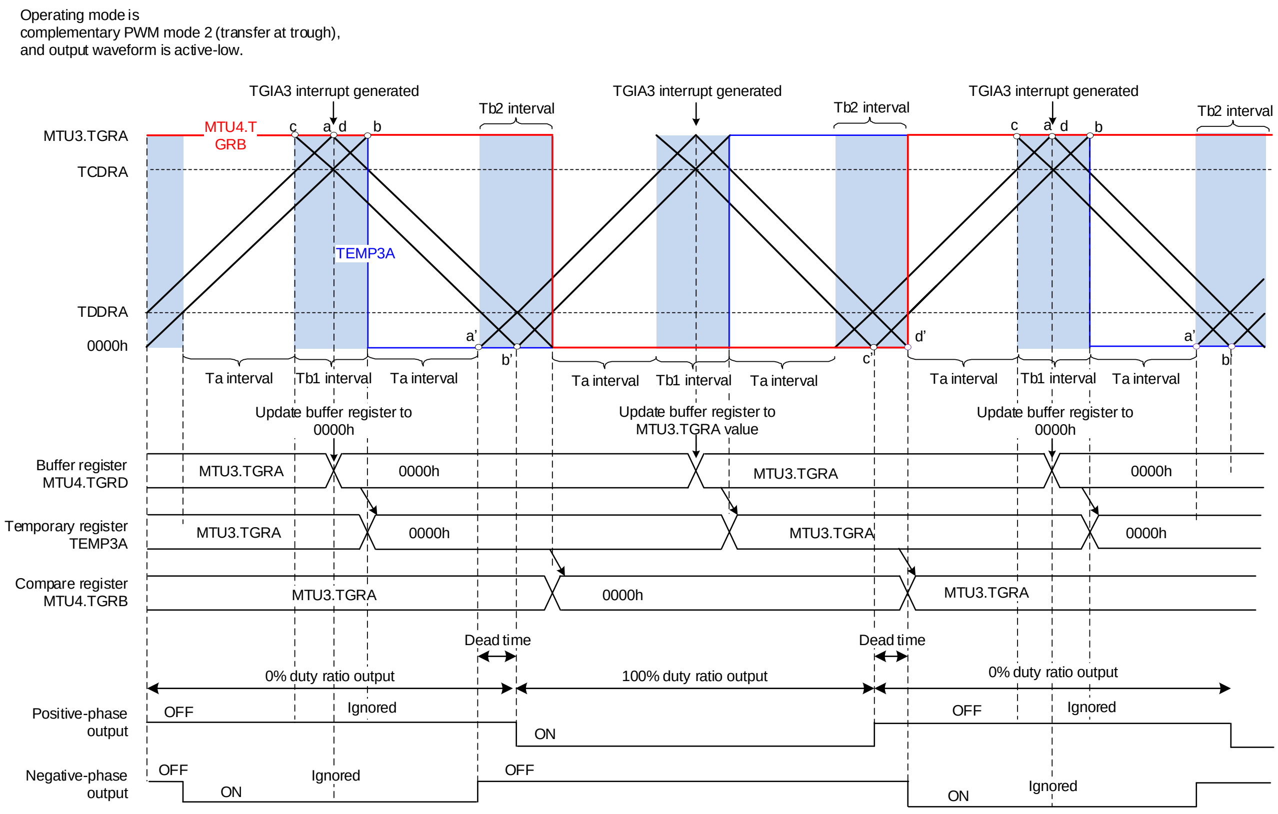



**Figure 5.1 Sample Code Operation**

## 5.2 File Composition

Table 5.1 lists the files used in the sample code. Files generated by the integrated development environment are not included in this table.

**Table 5.1 Files Used in Sample Code**

| File Name | Outline | Remarks |
|---|---|---|
| main.c | Main processing routine<br>Initial MTU settings<br>Initial port settings<br>TGIA3 interrupt handling routine | |
| r_init_stop_module.c | RX24T Group initial settings | For details, refer to RX24T Group Initial Settings Example (R01AN2837). |
| r_init_stop_module.h | | |
| r_init_port_initialize.c | | |
| r_init_port_initialize.h | | |
| r_init_clock.c | | |
| r_init_clock.h | | |

## 5.3 Option-Setting Memory

Table 5.2 lists the option-setting memory settings in the sample code. If necessary, change the values to match your system.

**Table 5.2 Option-Setting Memory Settings in Sample Code**

| Symbol | Address | Setting Value | Description |
|---|---|---|---|
| OFS0 | FFFF FF8Fh to FFFF FF8Ch | FFFF FFFFh | IWDT stops after a reset. |
| OFS1 | FFFF FF8Bh to FFFF FF88h | FFFF FFFFh | Voltage monitor 0 reset disabled after a reset.<br>HOCO oscillation disabled after a reset. |
| MDE | FFFF FF83h to FFFF FF80h | FFFF FFFFh | Little endian |

## 5.4 Variables

Table 5.3 lists the variables used in the sample code.

**Table 5.3 Variables**

| Type | Name | Description | Function Using Variable |
|---|---|---|---|
| uint8_t | s_duty_update_flag | Duty ratio update flag<br>0: Update buffer register to 0000h.<br>1: Update buffer register to MTU3.TGRA value. | Excep_MTU3_TGIA3 |

## 5.5 Functions

Table 5.4 lists the functions.

**Table 5.4 Functions**

| Function Name | Outline |
|---|---|
| main | Main processing routine |
| mtu_init | Initial MTU settings |
| mtu_port_init | Initial port settings |
| Excep_MTU3_TGIA3 | TGIA3 interrupt handling routine<br>(Updates the buffer register value each cycle.) |

## 5.6 Function Specifications

The sample code function specifications are listed below.

| main | |
|---|---|
| **Outline** | Main processing routine |
| **Header** | None |
| **Declaration** | void main(void) |
| **Description** | Calls the following functions, then starts count operation on MTU3 and MTU4.<br>• Disable peripheral modules running after a reset<br>• Initial nonexistent port settings<br>• Initial clock settings<br>• MTU initialization<br>• Port settings for MTU functions |
| **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** | For details, refer to RX24T Group Initial Settings Example (R01AN2837). |

| 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 pins of nonexistent ports. |
| **Arguments** | None |
| **Return Value** | None |
| **Remarks** | For details, refer to RX24T Group Initial Settings Example (R01AN2837). |

| 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** | For details, refer to RX24T Group Initial Settings Example (R01AN2837). |

| mtu_init | |
|---|---|
| **Outline** | MTU function initialization |
| **Header** | None |
| **Declaration** | static void mtu_init (void) |
| **Description** | Makes MTU settings needed to generate PWM waveforms using complementary PWM mode 2 with buffer operation.<br>Sets the PWM carrier period to 1 ms and dead time to 30 µs.<br>Sets the compare registers and buffer registers to the MTU3.TGRA value to generate 0% duty ratio on the first cycle. |
| **Arguments** | None |
| **Return Value** | None |

| mtu_port_init | |
|---|---|
| **Outline** | Port settings for MTU functions |
| **Header** | None |
| **Declaration** | static void mtu_port_init (void) |
| **Description** | Initializes port pins P71 to P76 for use by MTU functions.<br>Makes Hi-Z settings using the POE before and after port pin initialization for MTU functions. |
| **Arguments** | None |
| **Return Value** | None |

| Excep_MTU3_TGIA3 | |
|---|---|
| **Outline** | TGIA3 interrupt handling routine |
| **Header** | None |
| **Declaration** | void Excep_MTU3_TGIA3(void) |
| **Description** | Updates the buffer registers to 0000h or the MTU3.TGRA value. |
| **Arguments** | None |
| **Return Value** | None |

## 5.7 Flowcharts

### 5.7.1 Main Processing Routine

Figure 5.2 shows the main processing routine.

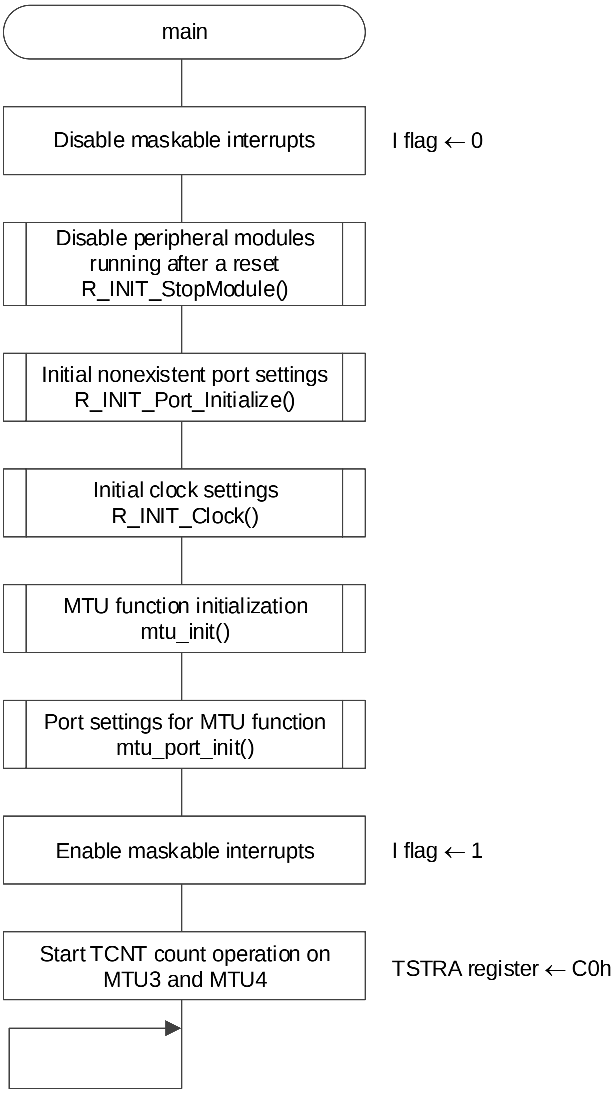



**Figure 5.2 Main Processing Routine**

### 5.7.2 MTU Initialization Function

Figure 5.3 shows the processing of the MTU initialization function.

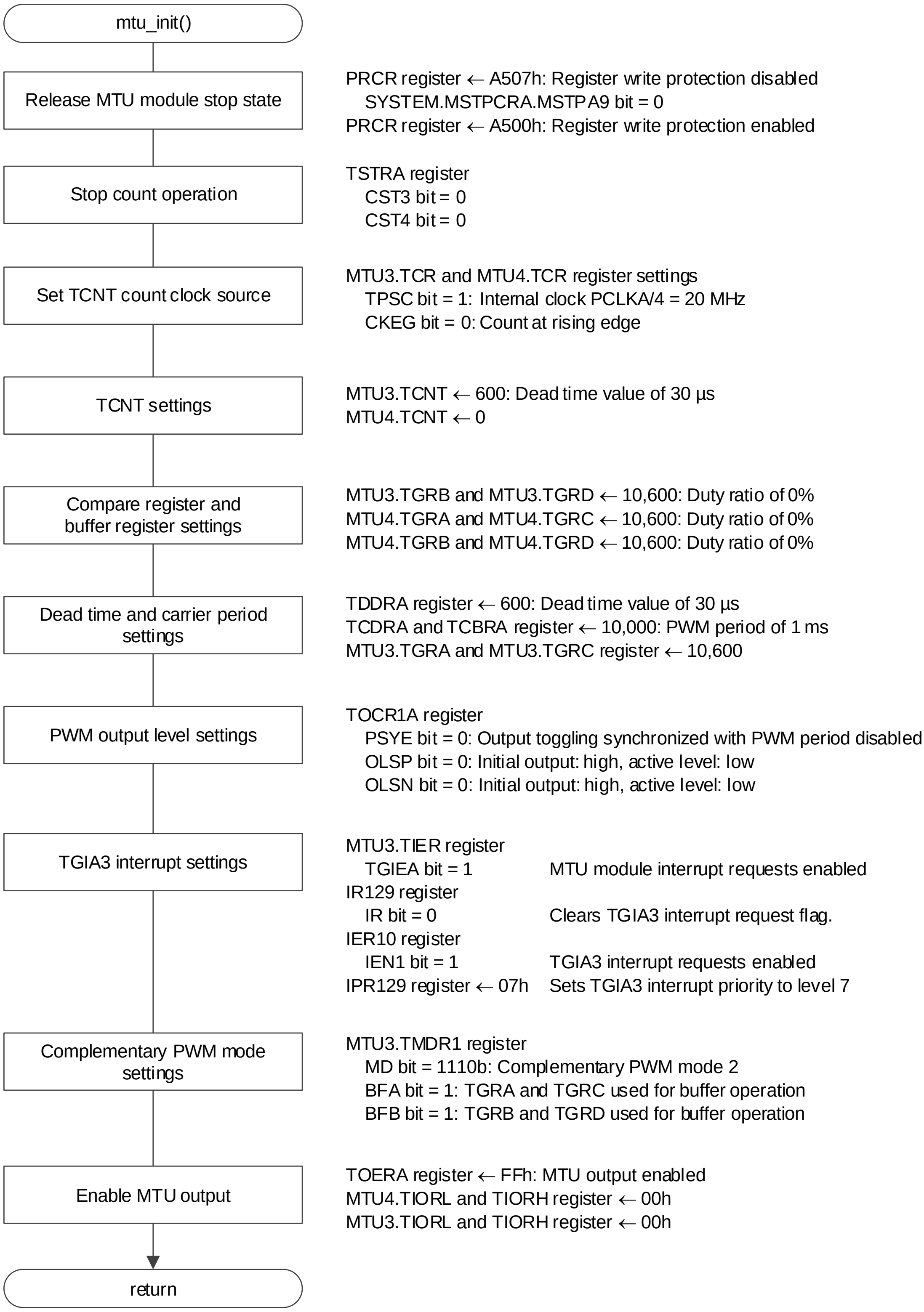



**Figure 5.3 MTU Initialization Function**

### 5.7.3 Port Initialization Function

Figure 5.4 shows the processing of the port initialization function.

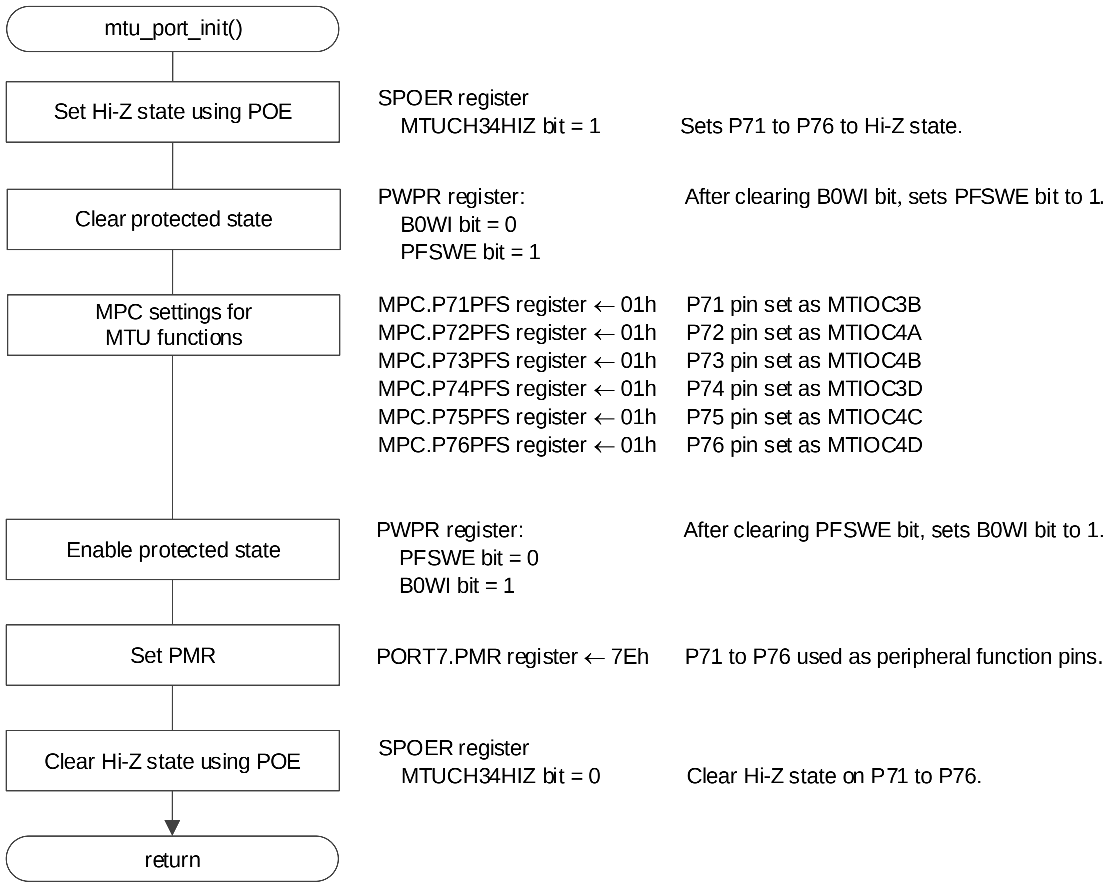
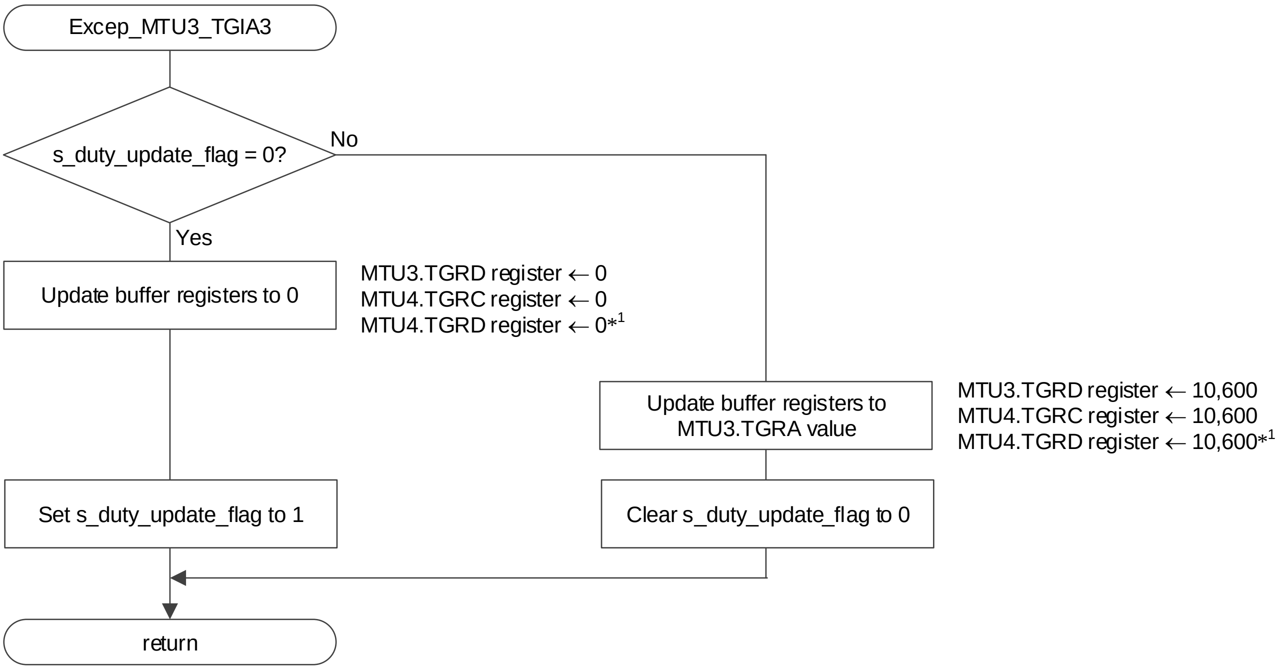



**Figure 5.4 Port Initialization Function**

### 5.7.4 TGIA3 Interrupt Handling Routine

Figure 5.5 shows the processing of the TGIA3 interrupt handling routine.




Note: 1. It is necessary to write to MTU4.TGRD at the end of the updating process in order to enable data transfer from the buffer registers to the temporary registers.

**Figure 5.5 TGIA3 Interrupt Handling Routine**

## 6. Applying This Application Note to The RX24U Group

In order to use the sample code with the RX24U Group it is necessary to create a project for RX24U in $e^2$ studio and replace the files from RX24T Group Initial Settings Example with the files from RX24U Group Initial Settings Example. For details, refer to 5.2, File Composition, in this application note and RX24U Group Initial Settings Example (R01AN3425).

# 7. Importing a Project

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

## 7.1 Importing a Project into 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.

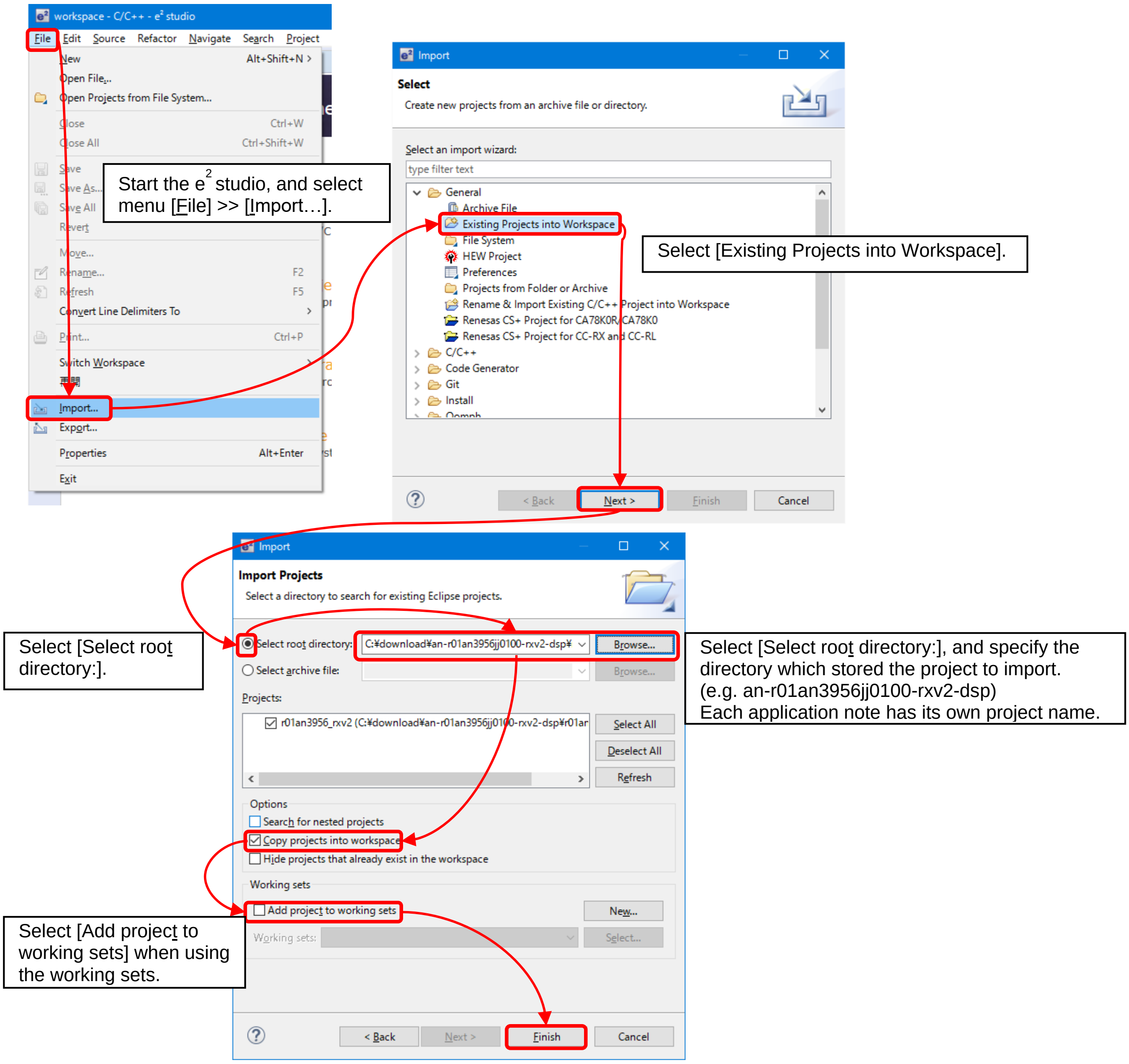



**Figure 7.1 Importing a Project into e$^2$ studio**

## 7.2 Importing a Project into CS+

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

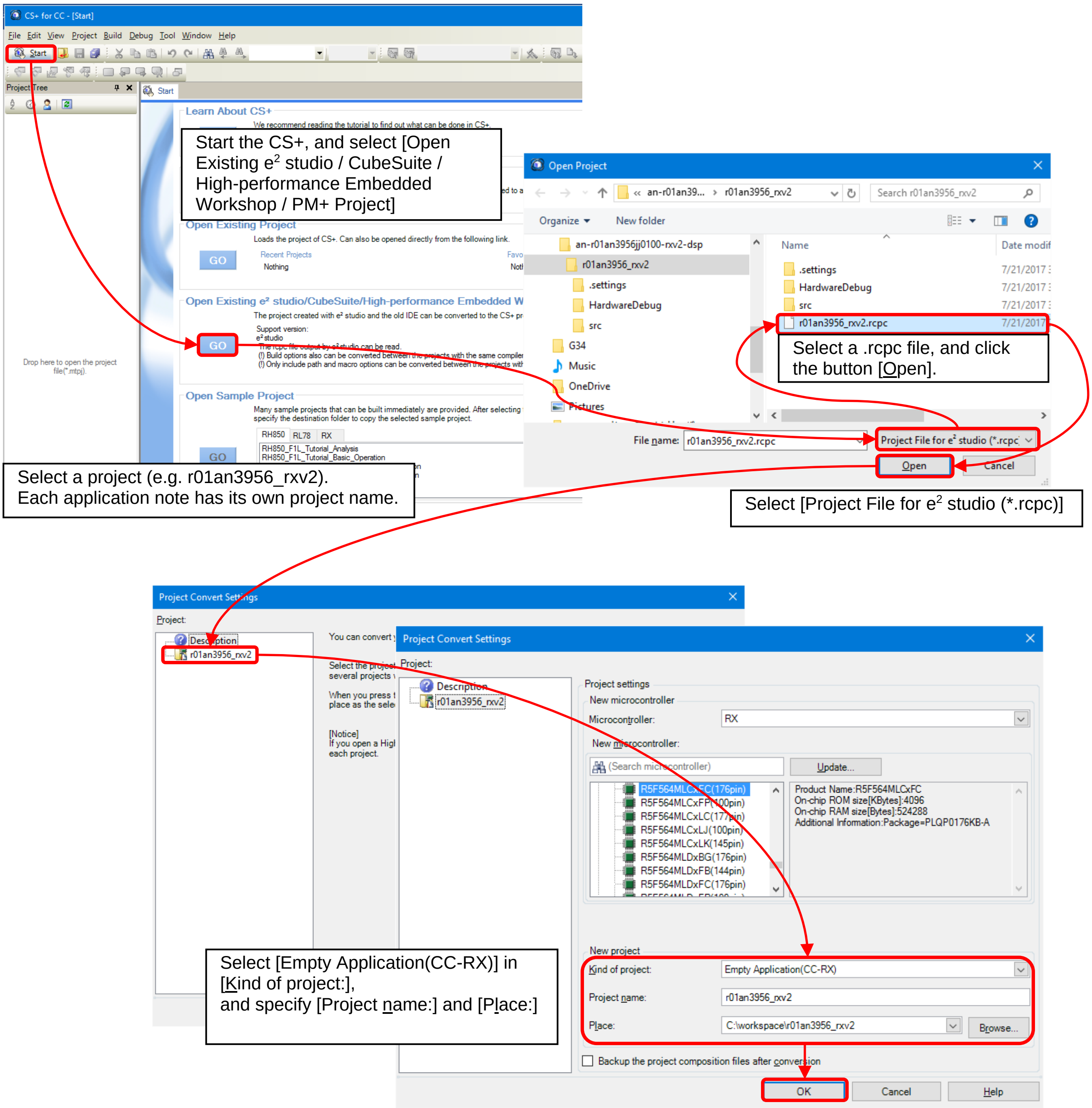



**Figure 7.2 Importing a Project into CS+**

## 8. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

## 9. Reference Documents

User's Manual: Hardware

RX24U Group User's Manual: Hardware (R01UH0658)
(The latest version can be downloaded from the Renesas Electronics website.)

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

Application Note
RX24U Group Initial Settings Example (R01AN3425)
(The latest version can be downloaded from the Renesas Electronics website.)

RX24T Group Initial Settings Example (R01AN2837)
(The latest version can be downloaded from the Renesas Electronics website.)

## Revision History

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.00 | Dec.10.19 | — | 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.

## Corporate Headquarters

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


## 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/.