RX Family

Usage Examples for Phase Counting Modes Using MTU3/GPTW

Introduction

This application note describes how to use phase counting modes by using the MTU3d and GPTW.

The RX66T-group microcomputer has on-chip multi-function timer pulse unit 3 (MTU3d) and general PWM timer (GPTW), thus supporting phase counting modes that count the number of edges for two pulse signals with a phase difference.

This application note is applicable to RX-family devices that have on-chip MTU3 and GPTW. If you use this application note for a microcomputer that is not of the RX66T group, you need to revise the content according to the specifications of the target microcomputer and evaluate operation adequately.

Applicable devices

RX-family devices that have on-chip MTU3 and GPTW

Device used for operational verification

RX66T group

Hereinafter, the multi-function timer pulse unit 3 is referred to as "MTU".

Contents

1. Specifications of the MTU and GPTW

This application note covers the phase counting modes for the MTU and GPTW.

Two pulse signals with a phase difference are input to the MTCLKA (MTCLKC and GTIOCnA) and MTCLKB (MTCLKD and GTIOCnB) pins, and then the number of edges are counted (by incrementing or decrementing).

The following describes the functionality of phase counting modes.

Table 1.1 Functionality of Phase Counting Modes

Item	MTU	GPTW
Channel	MTU1 and MTU2	GPT*n* (*n* = 0 to 9)
Functionality	A timer independently working in 16-bit phase counting mode can be set for each channel. A timer working in conjunction with two channels in 32-bit phase counting mode can be set (LWA = 1). Long words can be accessed. Timers can be cascade-connected (LWA = 0). Long words cannot be accessed.	A timer in 32-bit phase counting mode can be set for each MTU independently.
Mode	Modes 1 to 5	Modes 1 to 5
Configuration register	Mode setting: TMDR1 Mode expansion: TCR2.PCB[1:0]	Count-up setting: GTUPSR Count-down setting: GTDNSR
Input pin	A-phase: MTCLKA and MTCLKC B-phase: MTCLKB and MTCLKD	A-phase: GTCIOCnA B-phase: GTCIOCnB

1.1 Types of Phase Counting Modes and Settings in Smart Configurator

The following table shows the types of phase counting modes that can be set for the MTU and GPTW of an RX66T. For details about how to specify the settings in Smart Configurator for MTU or GPTW, click the [MTU] or [GPTW] link in the "Mode" column of the table.

Table 1.2 Types of Phase Counting Modes (1/3)

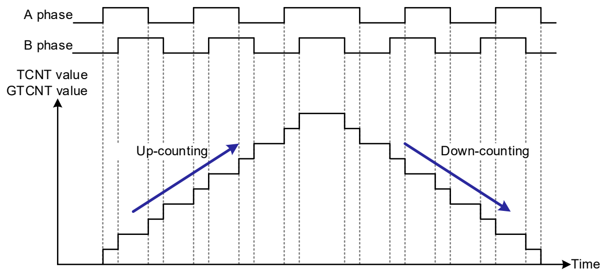
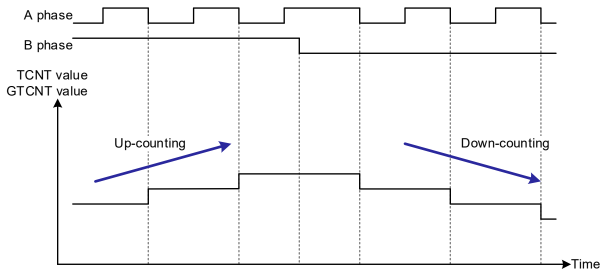
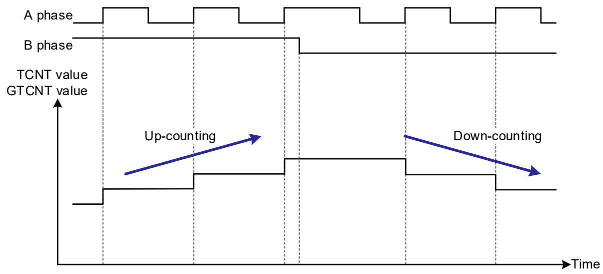
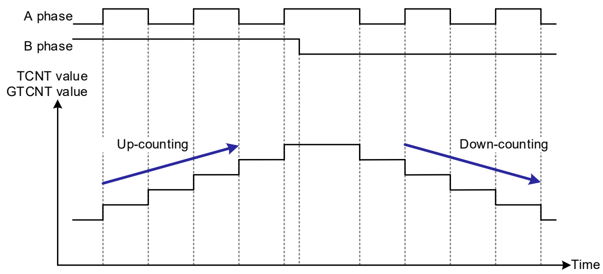
Mode	Operational conditions			Waveform
	Operation	**A-phase**	**B-phase**	
Mode 1 • MTU • GPTW	Up-counting	High	Rising	
		Low	Falling	
		Rising	Low	
		Falling	High	
	Down-counting	High	Falling	
		Low	Rising	
		Rising	High	
		Falling	Low	
Mode 2-1 • MTU • GPTW	Up-counting	Falling	High	
	Down-counting	Falling	Low	
Mode 2-2 • MTU • GPTW	Up-counting	Rising	High	
	Down-counting	Rising	Low	
Mode 2-3 • MTU • GPTW	Up-counting	Falling	High	
		Rising		
	Down-counting	Falling	Low	
		Rising		

Table 1.3 Types of Phase Counting Modes (2/3)

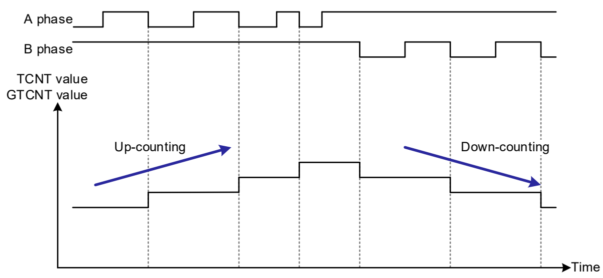
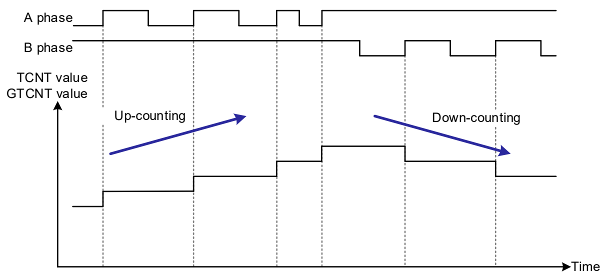
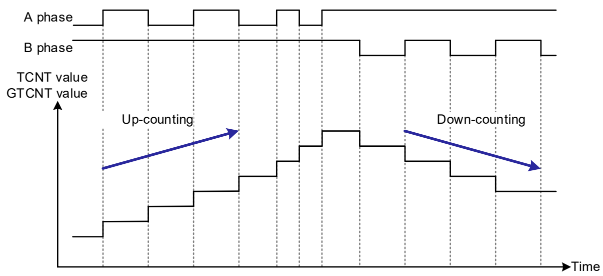
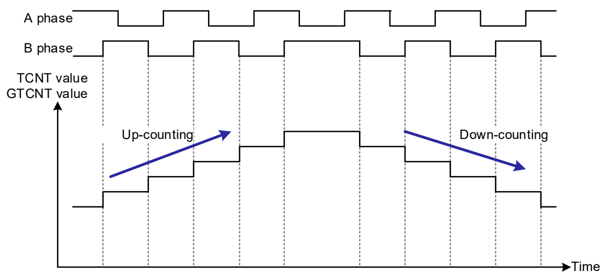
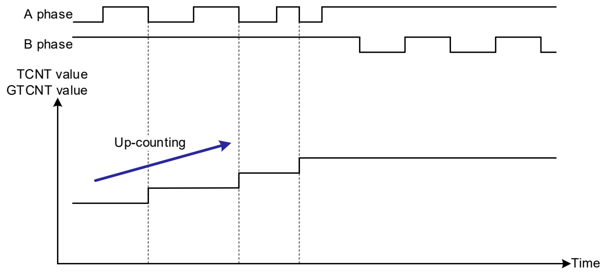
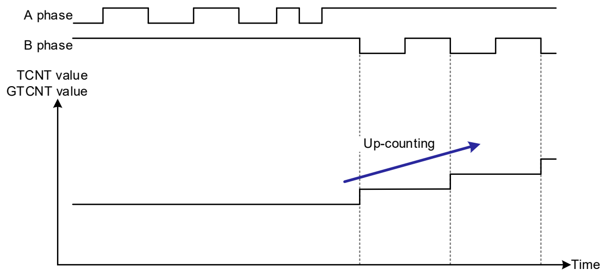
Mode	Operational conditions			Waveform
	Operation	**A-phase**	**B-phase**	
Mode 3-1 • MTU • GPTW	Up-counting	Falling	High	
	Down-counting	High	Falling	
Mode 3-2 • MTU • GPTW	Up-counting	Rising	High	
	Down-counting	High	Rising	
Mode 3-3 • MTU • GPTW	Up-counting	Falling	High	
		Rising		
	Down-counting	High	Falling	
			Rising	
Mode 4 • MTU • GPTW	Up-counting	High	Rising	
		Low	Falling	
	Down-counting	High	Falling	
		Low	Rising	

Table 1.4 Types of Phase Counting Modes (3/3)

Mode	Operational conditions			Waveform
	Operation	A-phase	B-phase	
Mode 5-1 • MTU • GPTW	Up-counting	Falling	High	
			Low	
	Down-counting	Down-counting not performed		
Mode 5-2 • MTU • GPTW	Up-counting	High	Falling	
		Low		
	Down-counting	Down-counting not performed		

1.1.1 Specifying the Settings in Smart Configurator (for MTU)

This section describes how to specify the phase counting mode settings for the MTU in Smart Configurator. The same settings apply to all types of phase counting modes (16-bit phase counting mode, 32-bit phase counting mode, and cascade connection mode)

Table 1.5 Phase Counting Mode Settings (for MTU) (1/4)

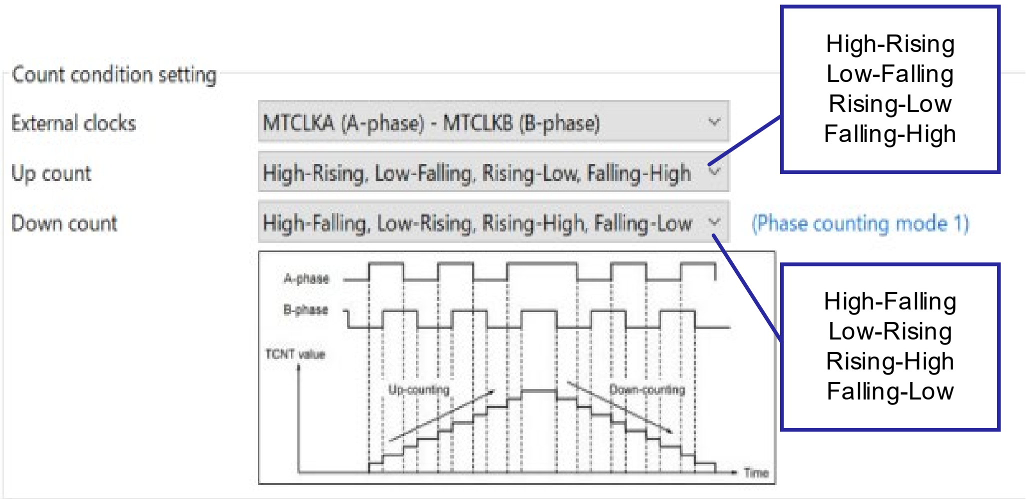
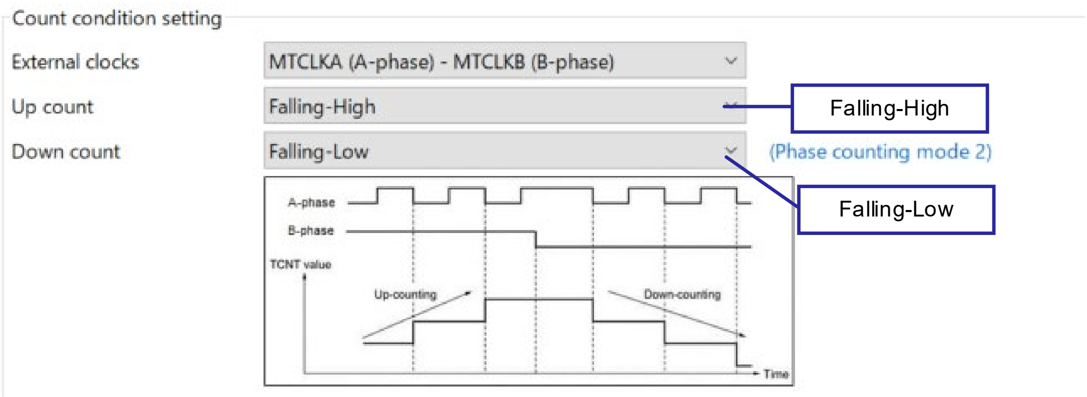
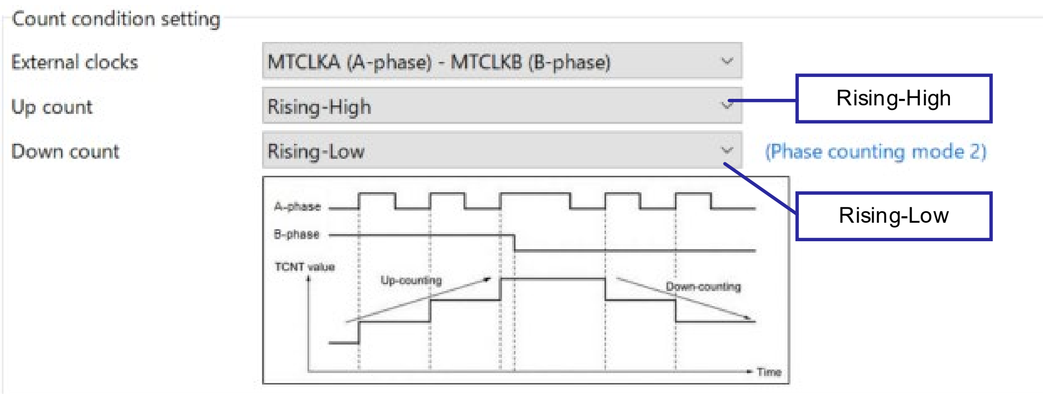
Mode	Waveform
Mode 1	
Mode 2-1	
Mode 2-2	

Table 1.6 Phase Counting Mode Settings (for MTU) (2/4)

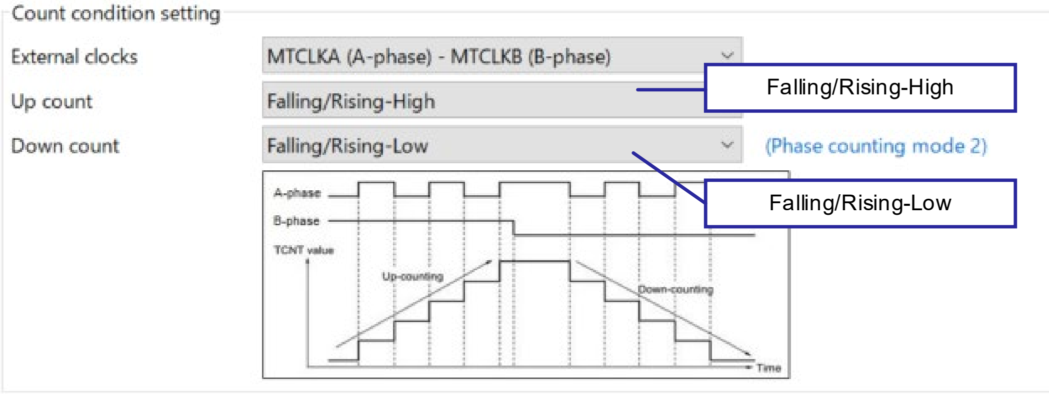
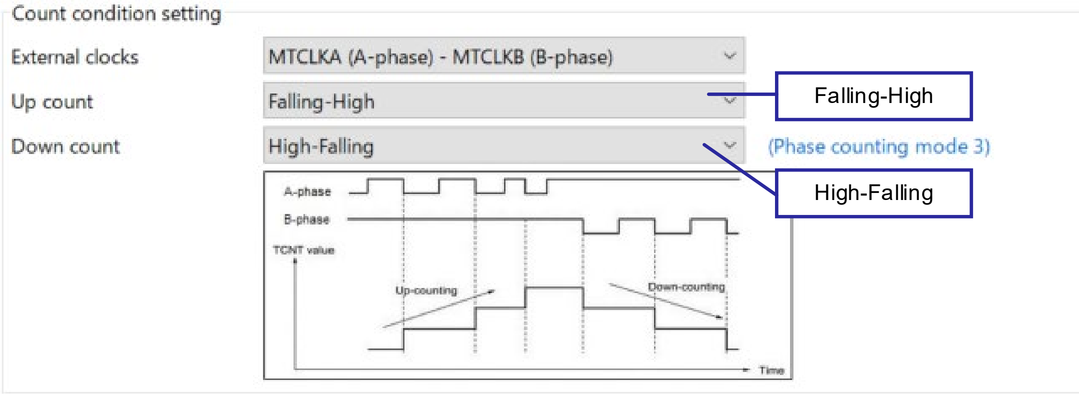
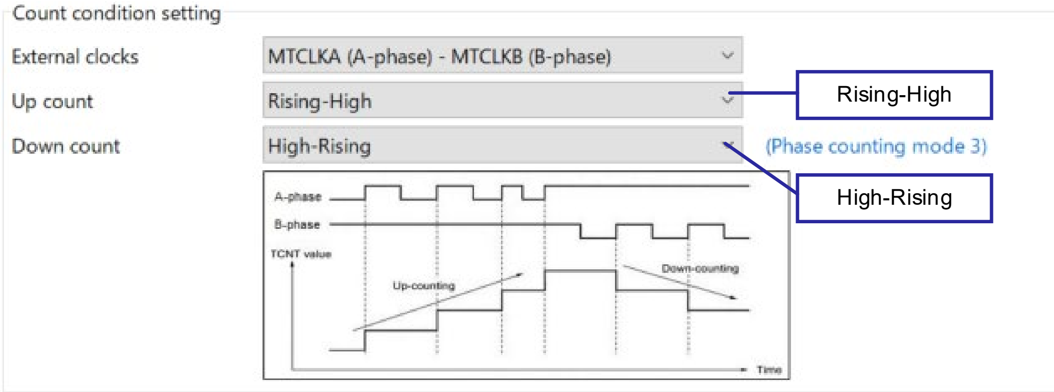
<table>
<tr><th>Mode</th><th>Waveform</th></tr>
<tr><td>Mode 2-3</td><td>
</td></tr>
<tr><td>Mode 3-1</td><td>
</td></tr>
<tr><td>Mode 3-2</td><td>
</td></tr>
</table>

Table 1.7 Phase Counting Mode Settings (for MTU) (3/4)

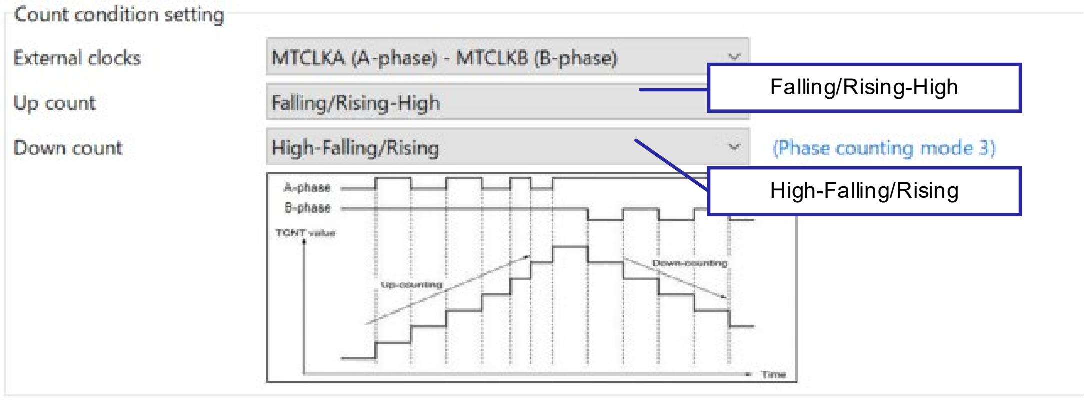
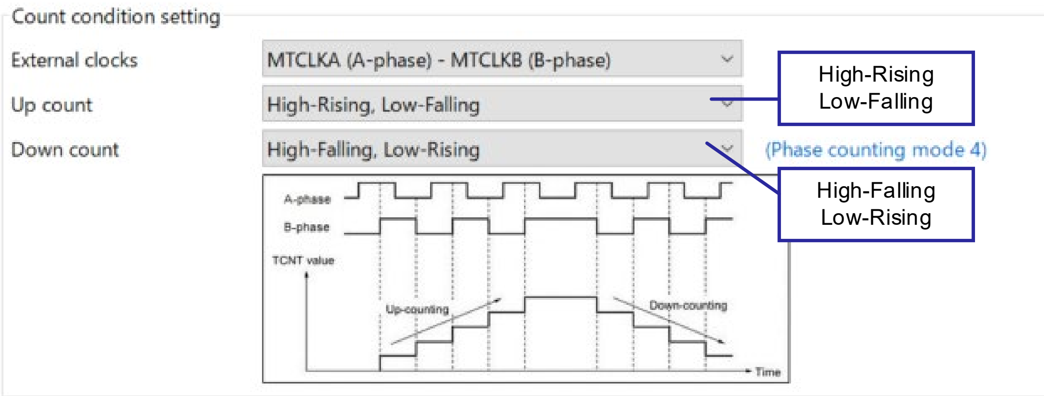
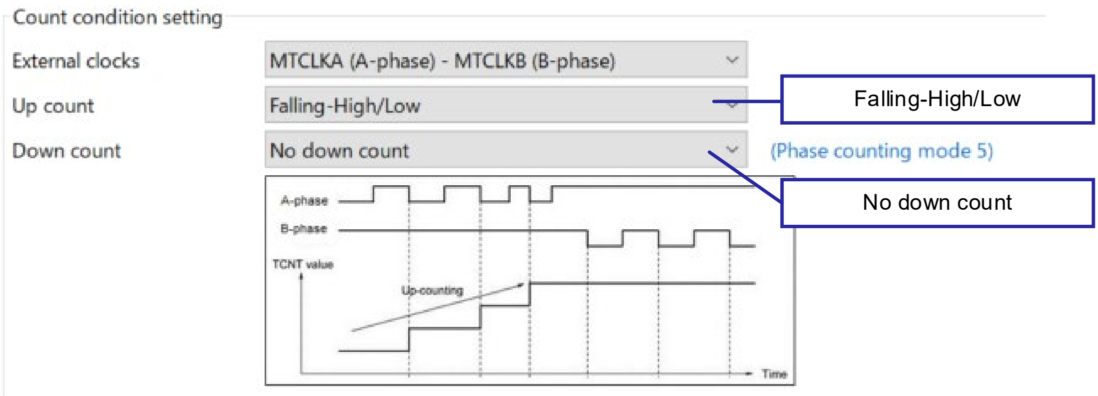
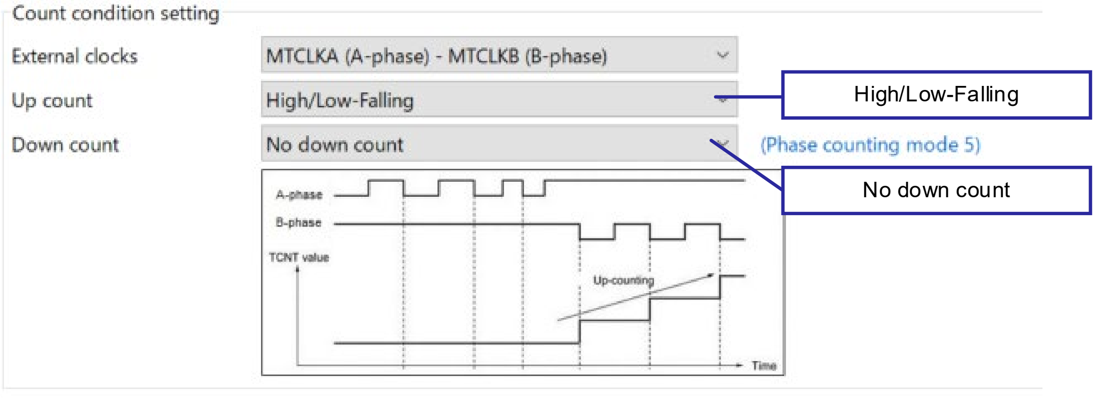
Mode	Waveform
Mode 3-3	
Mode 4	
Mode 5-1	

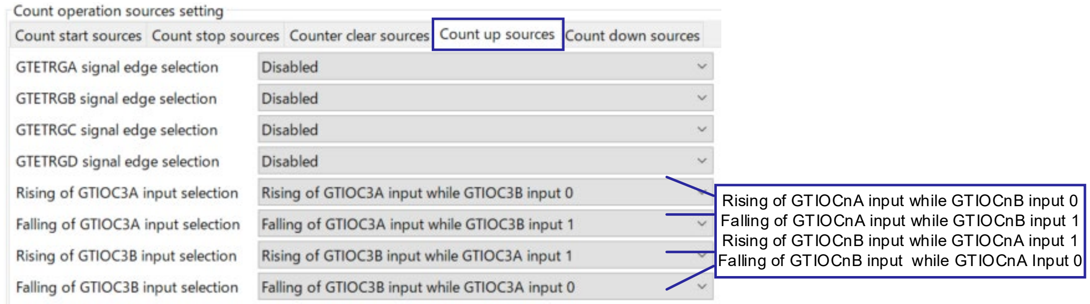
Table 1.8 Phase Counting Mode Settings (for MTU) (4/4)

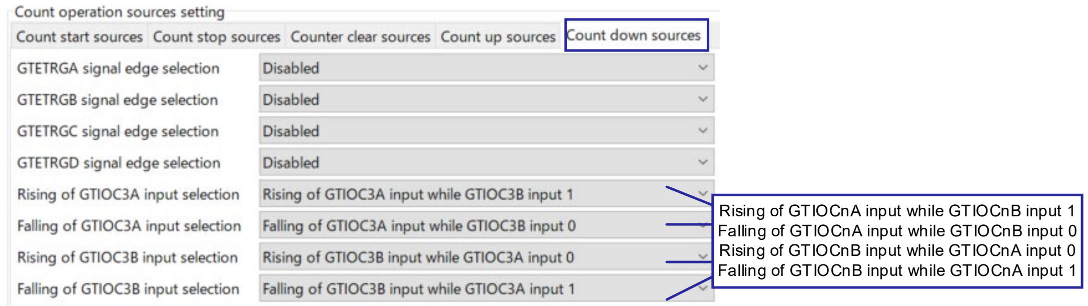
Mode	Waveform
Mode 5-2	

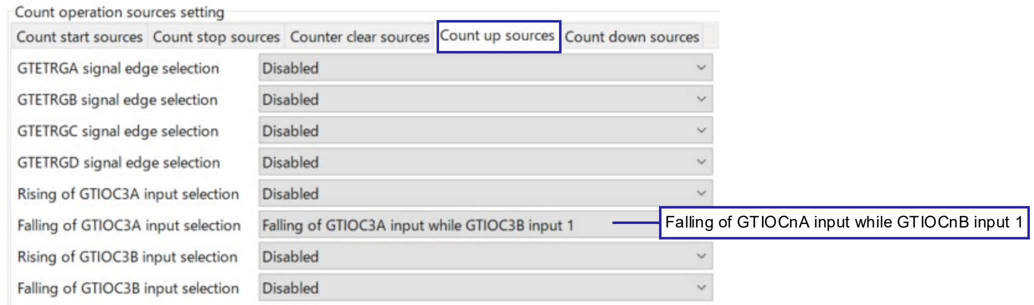
1.1.2 Specifying the Settings in Smart Configurator (for GPTW)

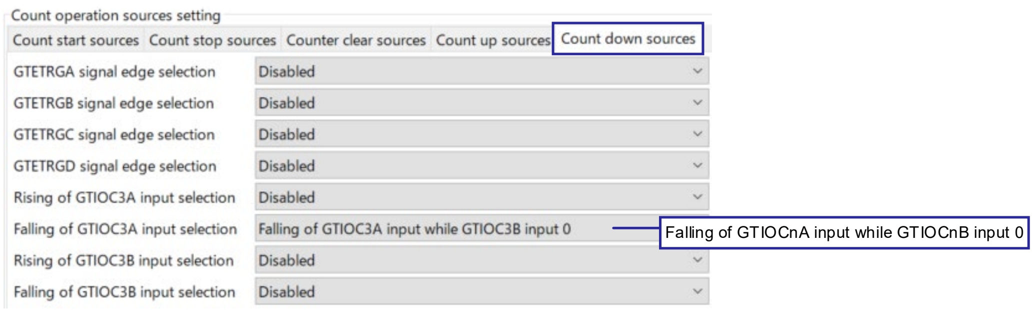
This section describes how to specify the phase counting mode settings for the GPTW in Smart Configurator.

Table 1.9 Phase Counting Mode Settings (for GPTW) (1/5)

<table>
<tr><th>Mode</th><th>Waveform</th></tr>
<tr><td>Mode 1</td><td>


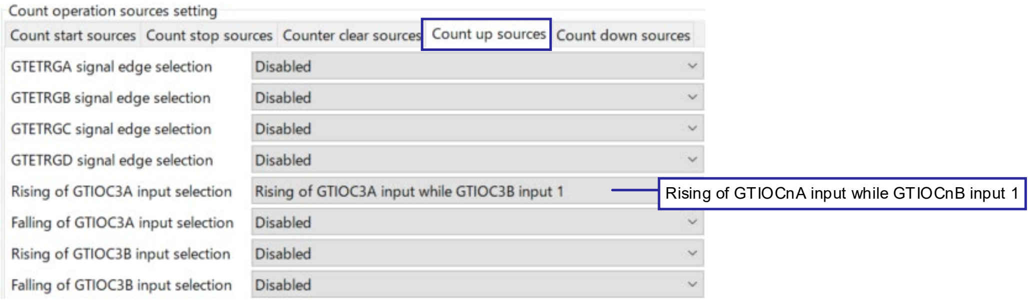


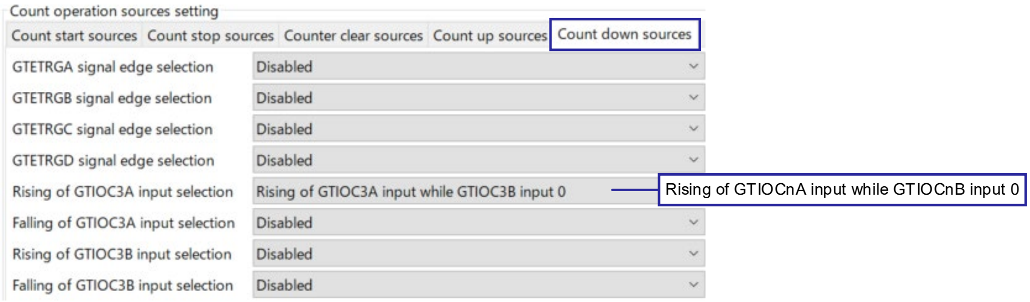
</td></tr>
<tr><td>Mode 2-1</td><td>


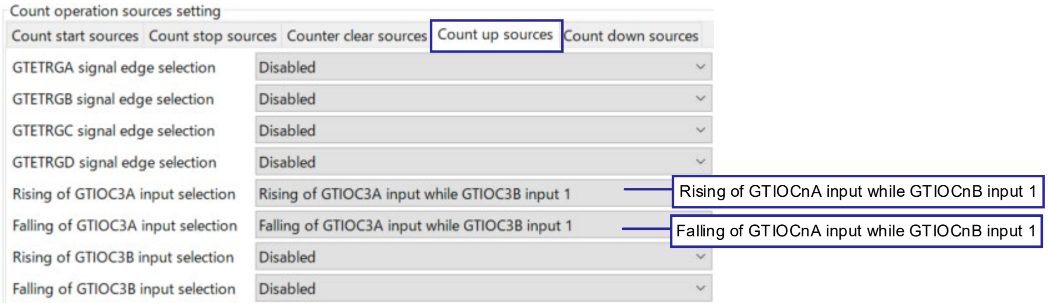


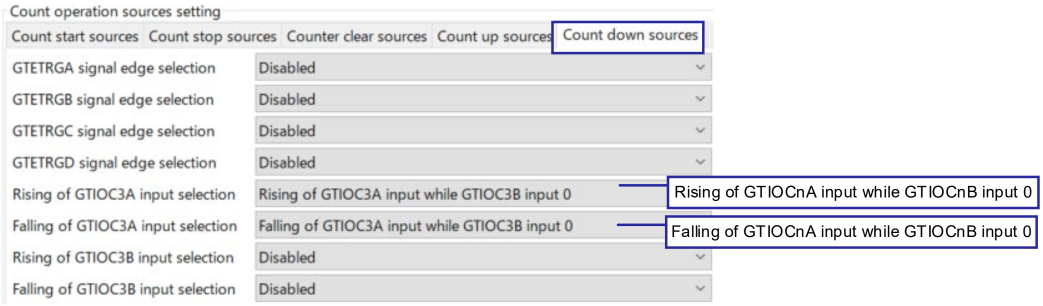
</td></tr>
</table>

Table 1.10 Phase Counting Mode Settings (for GPTW) (2/5)

<table>
<tr><th>Mode</th><th>Waveform</th></tr>
<tr><td>Mode 2-2</td><td>


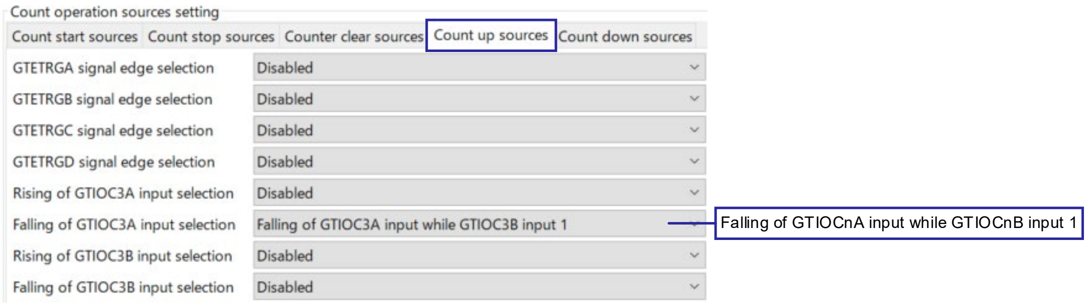


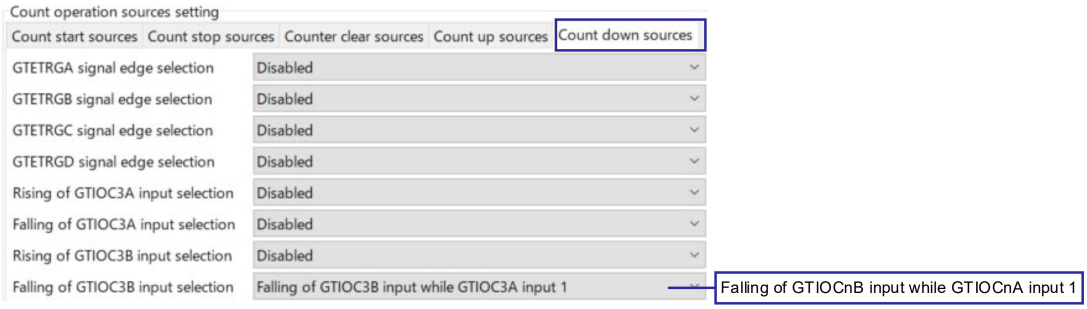
</td></tr>
<tr><td>Mode 2-3</td><td>


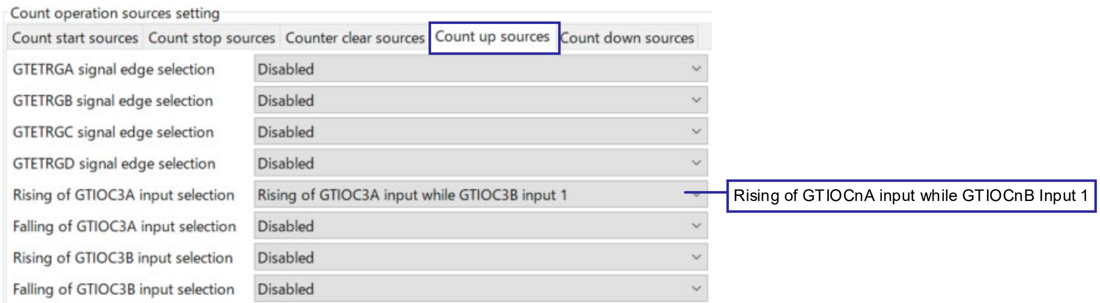


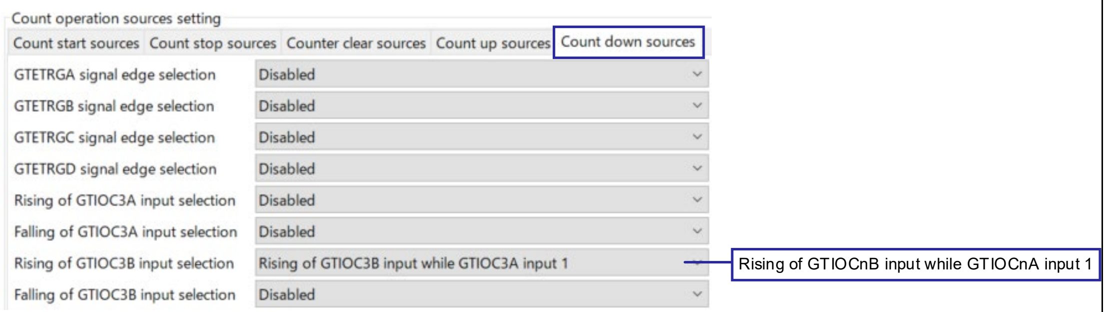
</td></tr>
</table>

Table 1.11 Phase Counting Mode Settings (for GPTW) (3/5)

<table>
<tr><th>Mode</th><th>Waveform</th></tr>
<tr><td>Mode 3-1</td><td>




</td></tr>
<tr><td>Mode 3-2</td><td>




</td></tr>
</table>

Table 1.12 Phase Counting Mode Settings (for GPTW) (4/5)

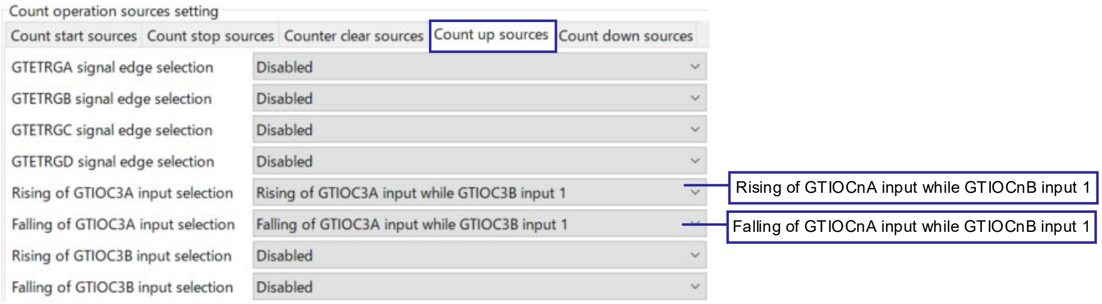
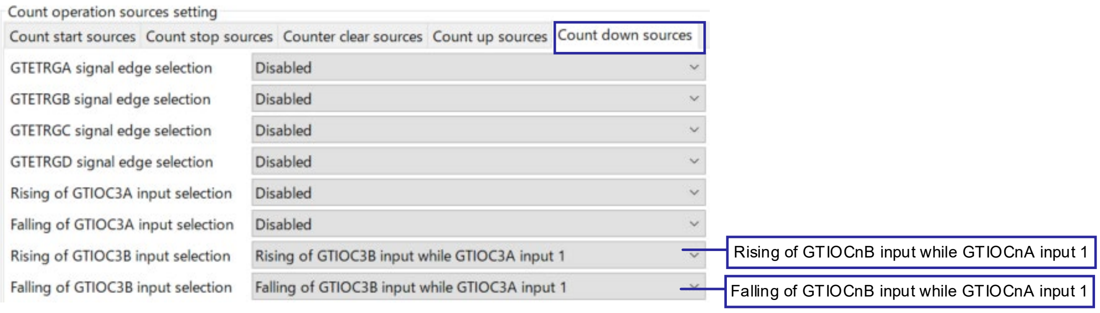
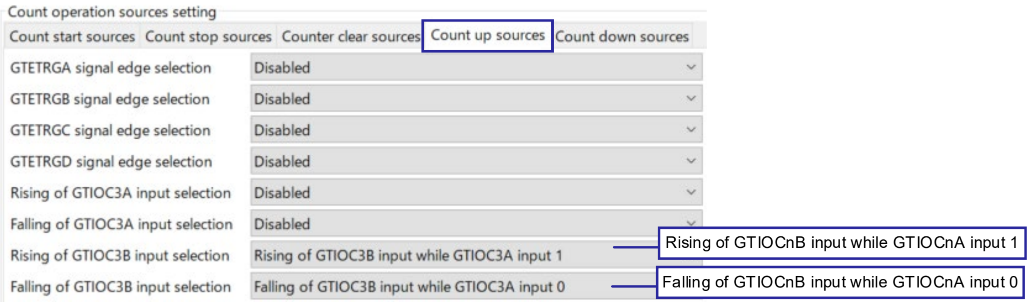
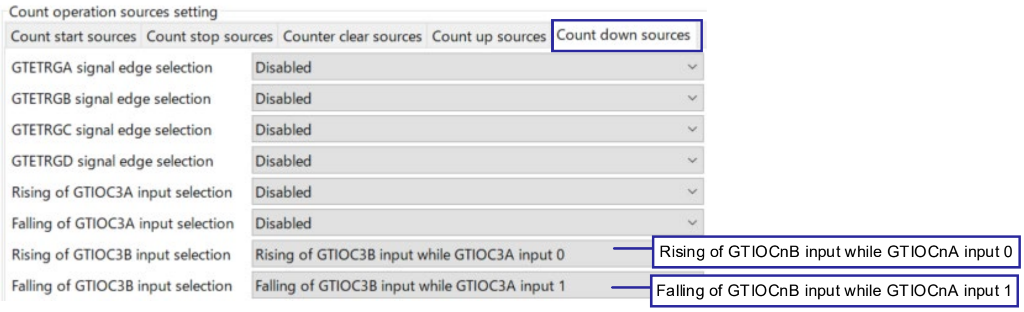
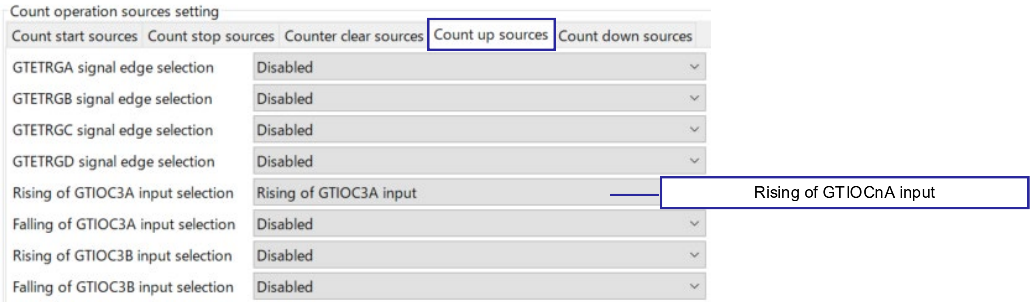
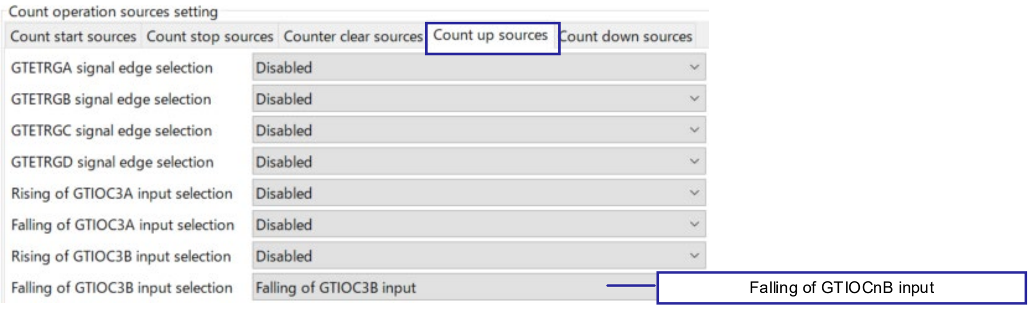
Mode	Waveform
Mode 3-3	 
Mode 4	 

Table 1.13 Phase Counting Mode Settings (for GPTW) (5/5)

<table>
<tr><th>Mode</th><th>Waveform</th></tr>
<tr><td>Mode 5-1</td><td>


</td></tr>
<tr><td>Mode 5-2</td><td>


</td></tr>
</table>

2. Conditions Under Which Operation Was Verified

For the sample code in this application note, operation was verified under the conditions in the following table.

Table 2.1 Environment Used for Verifying Operation

Item	Description
MCU used	R5F566TEADFP (mounted on the Renesas Starter Kit for RX66T)
Operating frequency	Main clock: 8 MHz PLL: 160 MHz (main clock x 1/1 x 20) HOCO: Inactive LOCO: Inactive System clock (ICLK): 160 MHz (PLL x 1/1) Peripheral module clock A (PCLKA): 80 MHz (PLL x 1/2) Peripheral module clock B (PCLKB): 40 MHz (PLL x 1/4) Peripheral module clock C (PCLKC): 160 MHz (PLL x 1/1) Peripheral module clock D (PCLKD): 40 MHz (PLL x 1/4) FlashIF clock (FCLK): 40 MHz (PLL x 1/4)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics e² studio Version 2022-04
C compiler[Note]	Renesas Electronics C/C++ Compiler Package for RX Family V3.04.00
	Compiler option The default settings of the integrated development environment apply.
iodefine.h version	V1.00
Endian	Little endian
Operation mode	Single-chip mode
Processor mode	Supervisor mode
Sample code version	V1.00
Board used	Renesas Starter Kit for RX66T (Product number: RTK50566T0C*xxxxx*BE)
Emulator	E2-Lite

Note: If the import-destination project does not have the same version of toolchain (for the C compiler) as that specified in the import-source project, no toolchain will be selected and an error will occur. Check whether a toolchain is selected in the project settings window.

For details about how to specify the settings, refer to "FAQ 3000404".

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

3. MTU Sample Code

3.1 General

3.1.1 Sample Code List

This application note provides the following types of sample code available with Smart Configurator.

These can be downloaded from the Renesas Electronics website.

Table 3.1 MTU Sample Code List

Name	Conditions where the sample code can be used	Refer to
16-bit Phase Counting Mode r01an6387_rx66t_mtu3_16_phase_cnt.zip	16-bit phase counting mode The MTIOC1A pin is used for Z-phase.	3.2
32-Bit Phase Counting Mode (MTU1.TMDR3.LWA = 1) r01an6387_rx66t_mtu3_32_phase_cnt.zip	32-bit phase counting mode The MTIOC1A pin is used for Z-phase.	3.3
32-Bit Phase Counting Mode (MTU1.TMDR3.LWA = 0) r01an6387_rx66t_mtu3_32_phase_cnt_cas.zip	32-bit phase counting mode (using cascade-connected timers) The MTIOC1A pin is used for Z-phase.	3.4

3.1.2 Folder Structure

The following shows the main folders for the sample code.

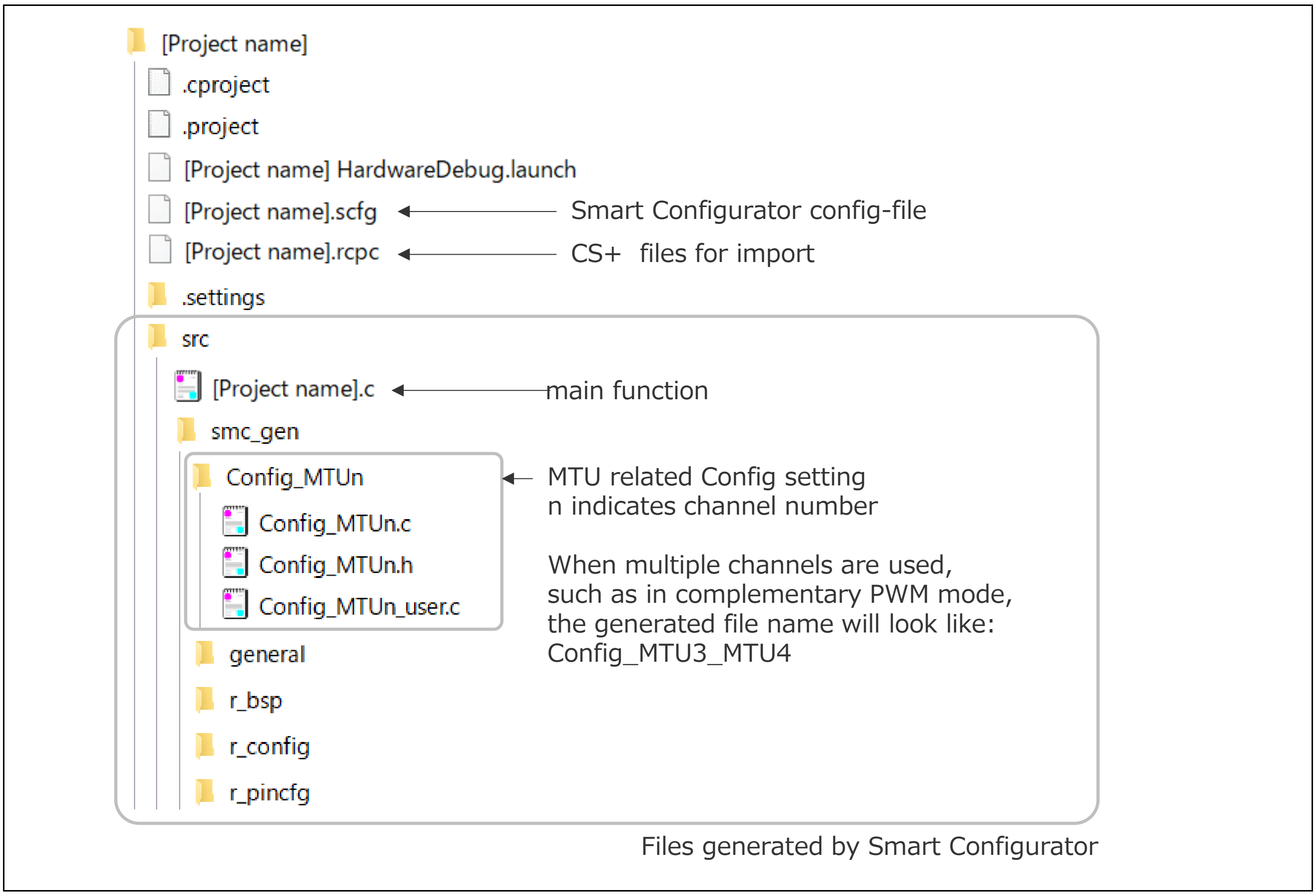


Figure 3.1 MTU Folder Structure

3.1.3 File Structure

The following shows the main files for the sample code.

Table 3.2 MTU File Structure

File name	Description
[Project-name].c	main function This is the main function. Smart Configurator generates this function with empty content. Add necessary processing according to the type of sample code.
Config_MTU*n*.c[Note]	R_Config_MTU*n*_Create function This function specifies the initial settings for the MTU. Smart Configurator generates an initial setting function according to the settings specified in Smart Configurator. Smart Configurator generates processing that invokes this function. This function is invoked by the R_SystemInit function that is run before the main function is run.
	R_Config_MTU*n*_Start function This function starts the counting for the MTU. Smart Configurator generates this function. In the sample code, this function is invoked by the main function.
	R_Config_MTU*n*_Stop function This function stops the counting for the MTU. Smart Configurator generates this function. This function is not used in the sample code.
Config_MTU*n*_user.c[Note]	r_Config_MTU*n*_Create_UserInit function This is a user function that specifies the initial settings for the MTU. Smart Configurator generates this function with empty content. Add necessary processing according to the type of sample code. This function is invoked at the end of the R_Config_MTU*n*_Create function that is generated by Smart Configurator.
	r_Config_MTU*n*_*interrupt-name*_interrupt function This is an interrupt handler function. Smart Configurator generates this function with empty content. Add necessary processing according to the type of sample code.
Config_MTU*n*.h[Note]	This is a header file that defines MTU-related functions. This file is included by the r_smc_entry.h file that is generated by Smart Configurator. To use MTU-related functions, include the r_smc_entry.h file.

Note: *n* indicates the channel number.

3.1.4 Adding a Component

In the sample code, Smart Configurator is used to add the MTU as follows.

Table 3.3 Adding a Component

Item	Description
Component	Refer to the section for the relevant type of sample code. ((1) in the following figure)
Configuration name	In the sample code, the initial value is used.
Operation	Refer to the section for the relevant type of sample code. ((2) in the following figure)
Resource	Refer to the section for the relevant type of sample code. ((3) in the following figure)

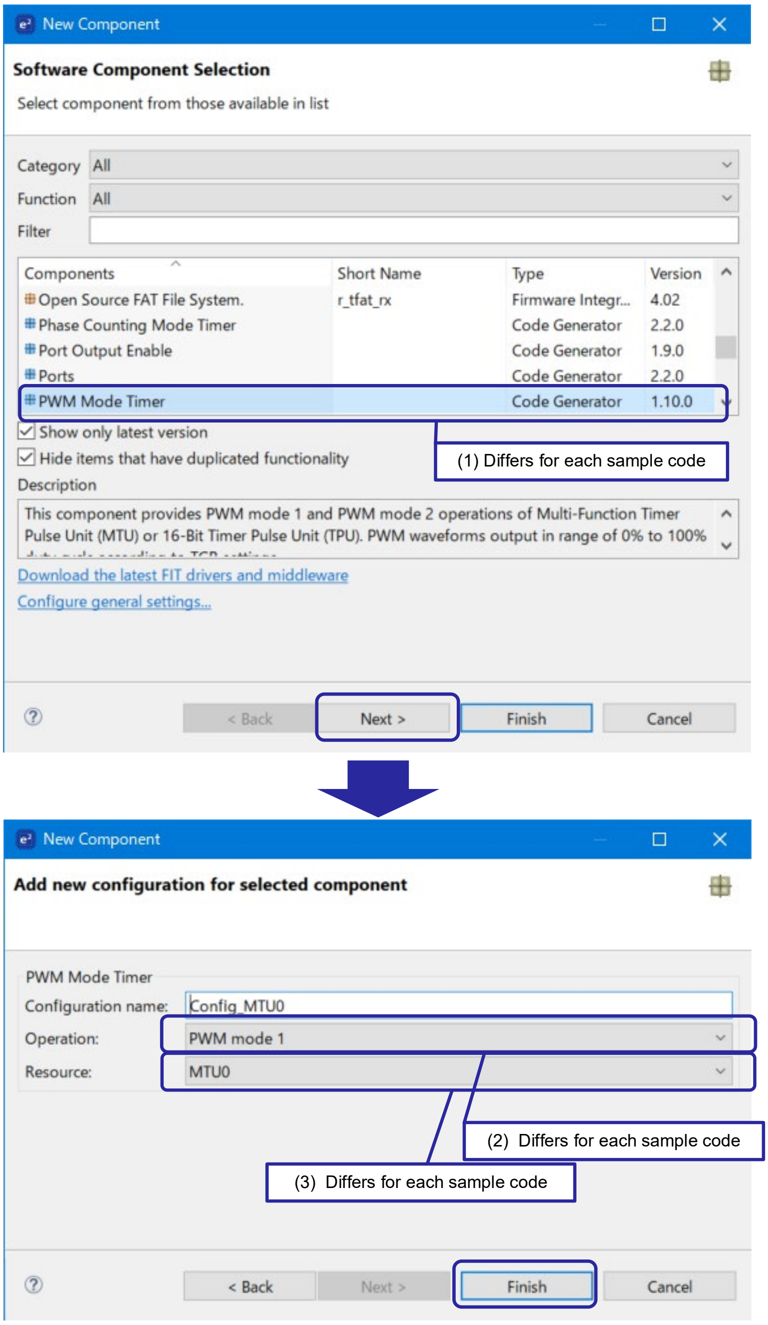


Figure 3.2 Adding a Component

3.1.5 Pin Configuration

Figure 3.3 shows an example of configuring pins by using Smart Configurator.

Before you configure pins, make sure that the MTU has been configured. For details about how to configure the MTU, refer to the "Smart Configurator Settings" section appropriate for the relevant type of sample code.

Pin configuration is performed inside the R_Config_MTU*n*_Create function that is generated by Smart Configurator.

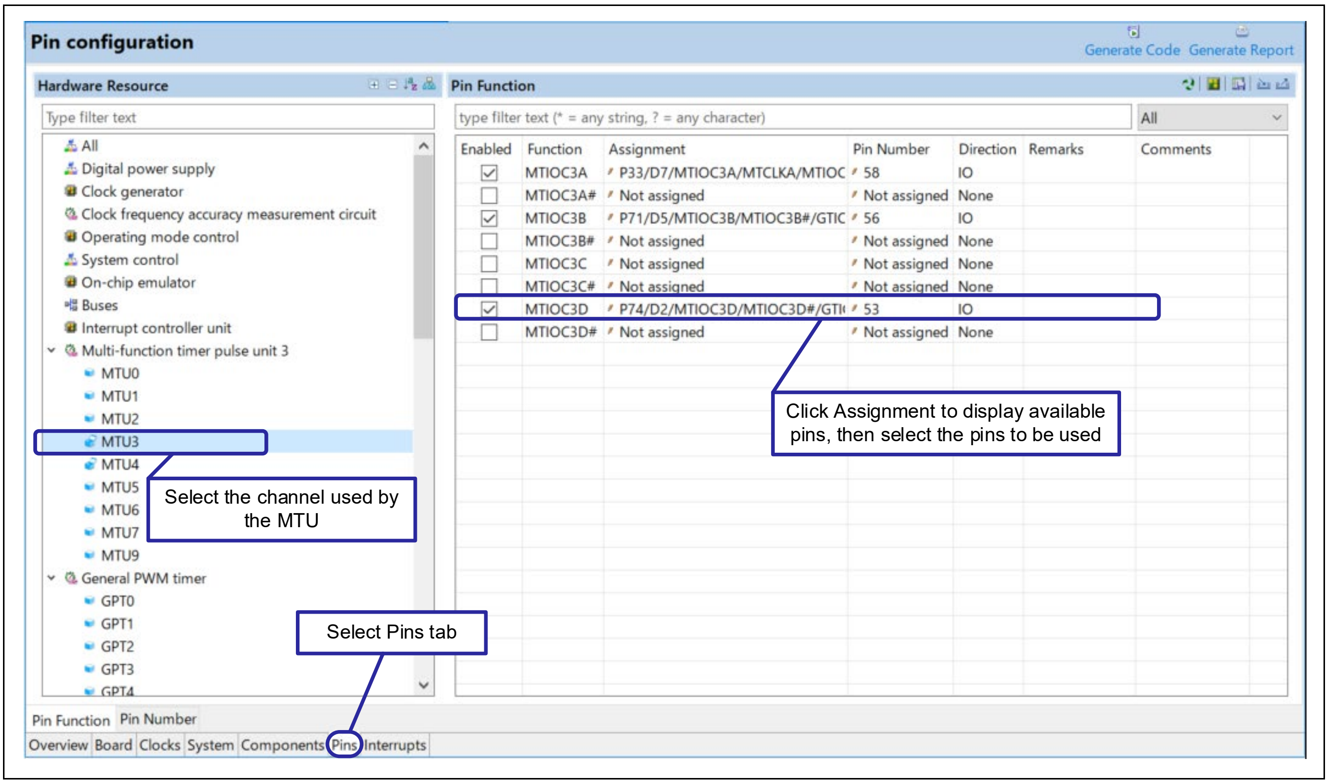


Figure 3.3 Pin Configuration

3.1.6 Interrupt Configuration

Figure 3.4 shows an example of setting an interrupt by using Smart Configurator. For details about software configurable interrupt A, refer to section 14.4.5.1, Software Configurable Interrupt A in the RX66T Group User's Manual: Hardware.

Before you configure interrupts, make sure that the MTU has been configured. For details about how to configure the MTU, refer to the "Smart Configurator Settings" section appropriate for the relevant type of sample code.

Interrupt configuration is performed inside the R_Config_MTU*n*_Create, R_Config_MTU*n*_Start, and R_Config_MTU*n*_Stop functions that are generated by Smart Configurator.

Interrupt handler functions are created with names in the "r_Config_MTU*n*_*interrupt-name*_interrupt" format in the Config_MTU*n*_user.c file that is generated by Smart Configurator.

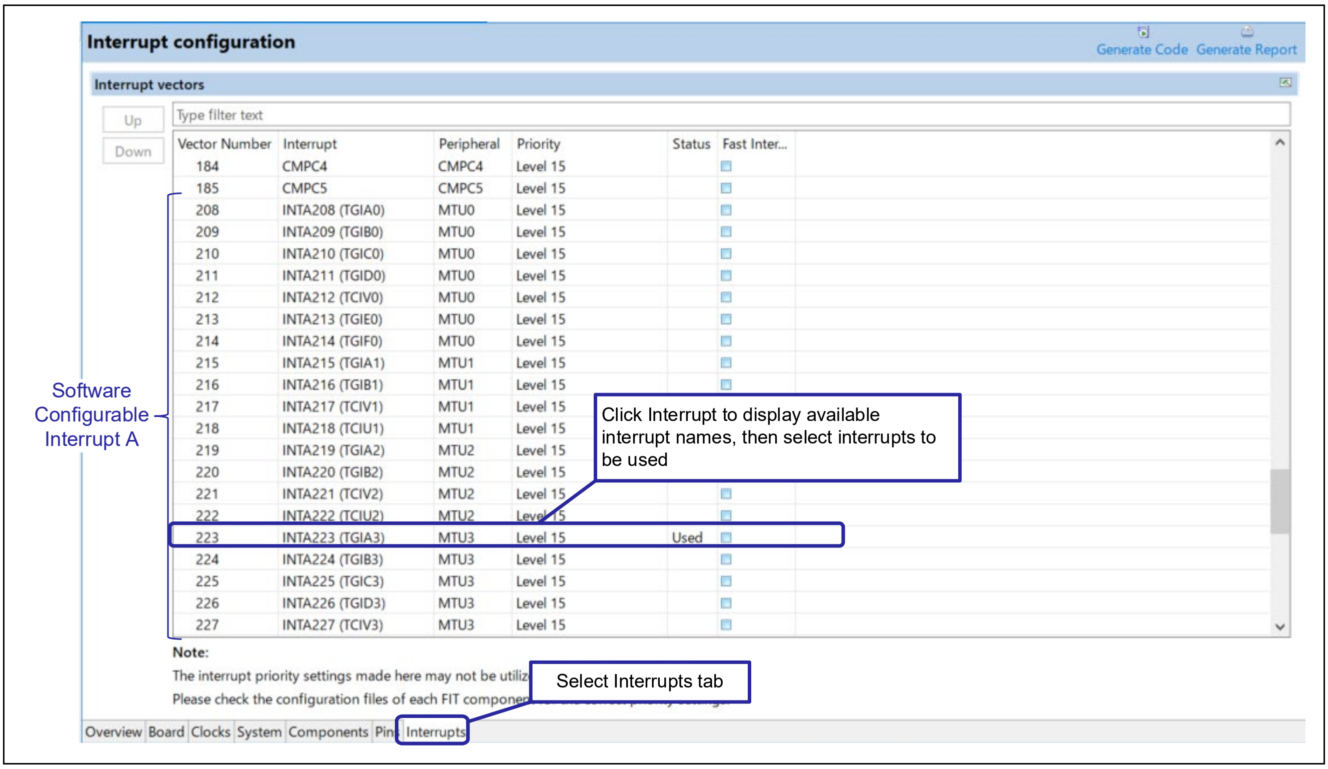


Figure 3.4 Interrupt Configuration

3.2 16-bit Phase Counting Mode

- Applicable sample code file name: r01an6387_rx66t_mtu3_16_phase_cnt.zip

3.2.1 Overview

This section describes how to use the 16-bit phase counting mode for the MTU.

This sample code uses phase counting mode 1, in which A-phase and B-phase signals from a 2-phase encoder are input to the MTCLKA and MTCLKB pins, and the number of pulses is counted.

The sample code also inputs the Z-phase signal into MTIOC1A, and clears MTU1.TCNT upon detecting a rising edge.

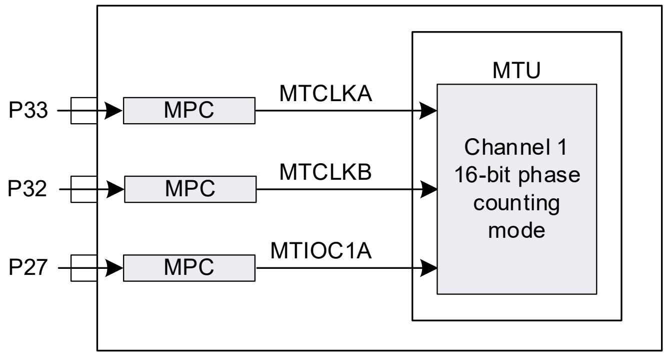
The following describes the MTU settings used in the sample code.

- MTU1 (channel 1)
  - 16-bit phase counting mode is used.
  - External clocks MTCLKA (for A-phase) and MTCLKB (for B-phase) are used.
  - For [Up count], select: [High-Rising, Low-Falling, Rising-Low, Falling-High]
  - For [Down count], select: [High-Falling, Low-Rising, Rising-High, Falling-Low]
  - MTU1.TGRA is used as an input capture register.
    - Input capture with MTU1.TGRA is used as the clearing source for the timer counter.
  - Input capture is performed at a rising edge of the MTIOC1A pin input.

These settings can be specified by using Smart Configurator.

For details about how to specify the settings, refer to section 3.2.3.

The following shows an overview of the hardware configuration for this sample code.



MTU1: 16-bit phase counting mode
MPC: Sets pins to be used from general purpose I/O port to peripheral function I/O port

Figure 3.5 Hardware Configuration for the Sample Code

3.2.2 Operation Details

This section describes the operation of this sample code.

The TCNT counter performs up-counting or down-counting according to the waveforms of input A-phase and B-phase signals, based on the edge detection settings for the count-up and count-down sources in phase counting mode 1.

When the sample code detects a rising edge at MTIOC1A, it uses TGRA to perform input capture, which triggers clearing the MTU1.TCNT counter.

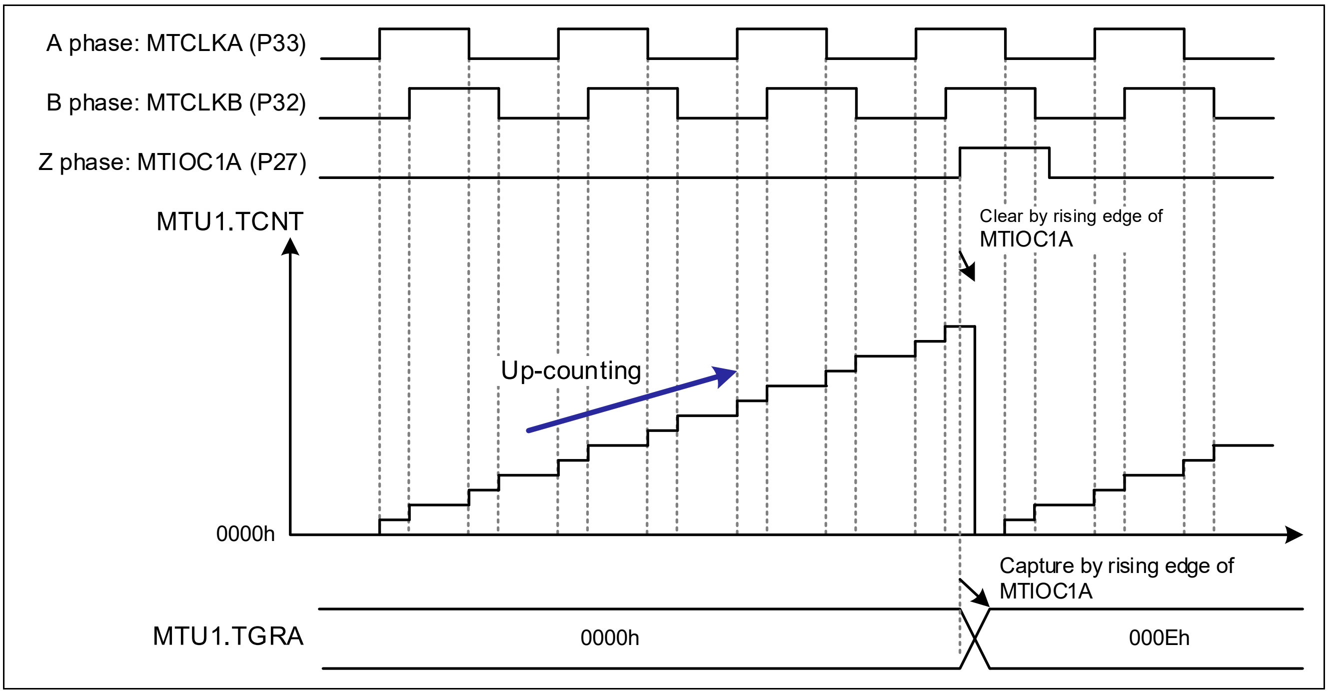


Figure 3.6 Operation of the Sample Code

3.2.3 Smart Configurator Settings

In the sample code, Smart Configurator is used to add the MTU as follows. For details about how to add a component, refer to section 3.1.4, Adding a Component.

Table 3.4 Adding a Component

Item	Description
Component	Phase counting mode timer
Configuration name	Config_MTU1
Operation	16-bit phase counting mode
Resource	MTU1

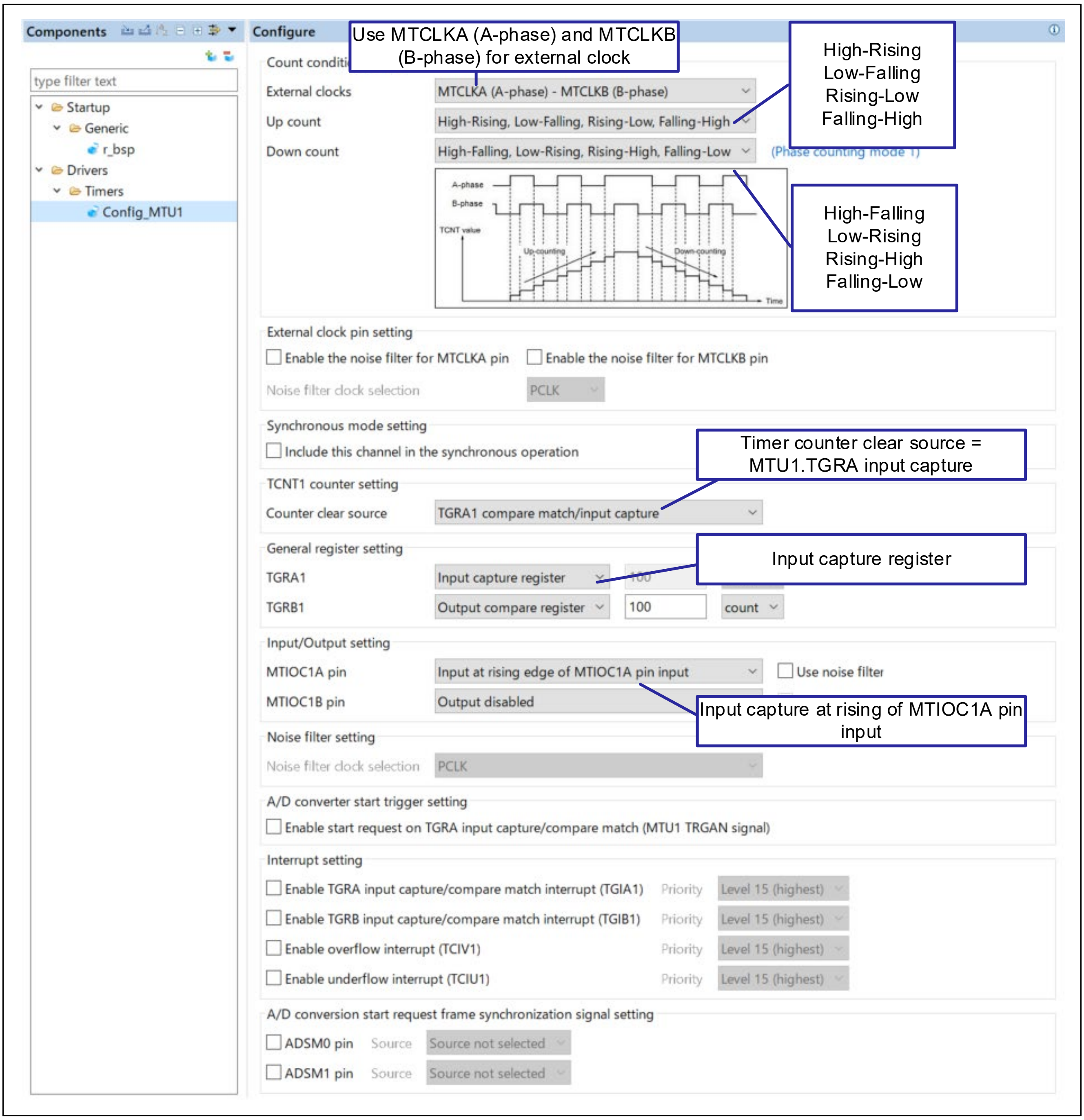


Figure 3.7 MTU1 Settings

3.2.4 Flowchart

The following shows the processing that was added to the main function after code was generated by Smart Configurator.

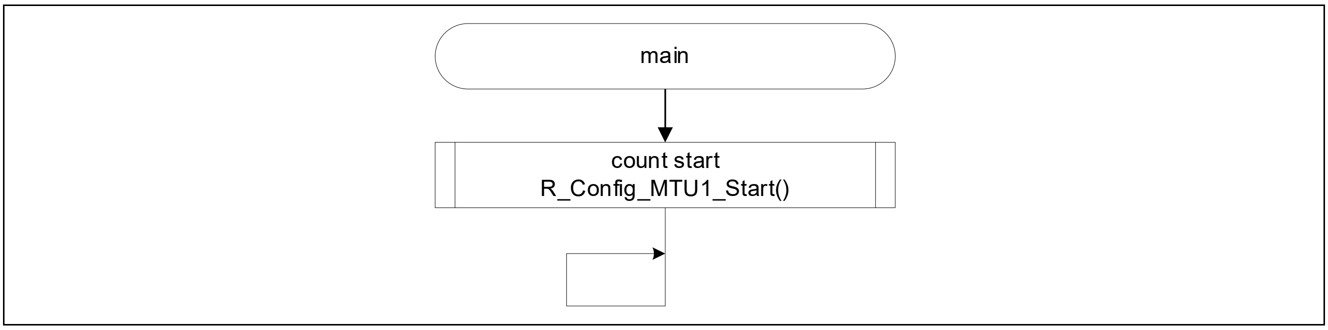


Figure 3.8 main Function

3.2.5 Precautions

3.2.5.1 External Clock Input Pins Connected in 16-Bit Phase Counting Mode

In 16-bit phase counting mode, MTU1 and MTU2 can operate independently. In 16-bit phase counting mode, the external clock input pins that can be selected for MTU1 and for MTU2 are different. Inverting input pins of the external clock can also be selected. The following table shows the combinations of external clock input pins that can be selected.

Table 3.5 Clock Input Pins Available in 16-Bit Phase Counting Mode

Channel	External clock pin	
	A-phase	B-phase
MTU1	MTCLKA	MTCLKB
	MTCLKA#	MTCLKB#
	MTCLKA#	MTCLKB
	MTCLKA	MTCLKB#
MTU2	MTCLKA	MTCLKB
	MTCLKC	MTCLKD
	MTCLKA#	MTCLKB#
	MTCLKC#	MTCLKD#
	MTCLKA#	MTCLKB
	MTCLKA	MTCLKB#
	MTCLKC#	MTCLKD
	MTCLKC	MTCLKD#

For details, refer to section 22.3.6, Phase Counting Mode in the RX66T Group User's Manual: Hardware.

3.2.5.2 Times When Counting Starts and the Counter Is Cleared in Phase Counting Modes

For details about the times when counting starts in phase counting modes, refer to (1), TCNT Count Timing, in section 22.5.1, Input/Output Timing, in the RX66T Group User's Manual: Hardware.

For details about the times when the counter is cleared, refer to (4), Timing for Counter Clearing by Compare Match/Input Capture in section 22.5.1, Input/Output Timing in the RX66T Group User's Manual: Hardware.

3.2.5.3 Specifications Applying When MTIOC1B Is Not Used

This sample code uses only MTIOC1A. Because MTIOC1B is unused, it can be used for the other function or as an I/O port.

By using the TGR register as a compare match register, you can output any value of your choice by using the compare match function.

3.3 32-Bit Phase Counting Mode (MTU1.TMDR3.LWA = 1)

- Applicable sample code file name: r01an6387_rx66t_mtu3_32_phase_cnt.zip

3.3.1 Overview

This section describes how to use the 32-bit phase counting mode for the MTU.

This sample code uses phase counting mode 1, in which A-phase and B-phase signals of a 2-phase encoder are input to the MTCLKA and MTCLKB pins and the number of pulses is counted.

The sample code also inputs the Z-phase signal into MTIOC1A, and clears TCNTLW upon detecting a rising edge.

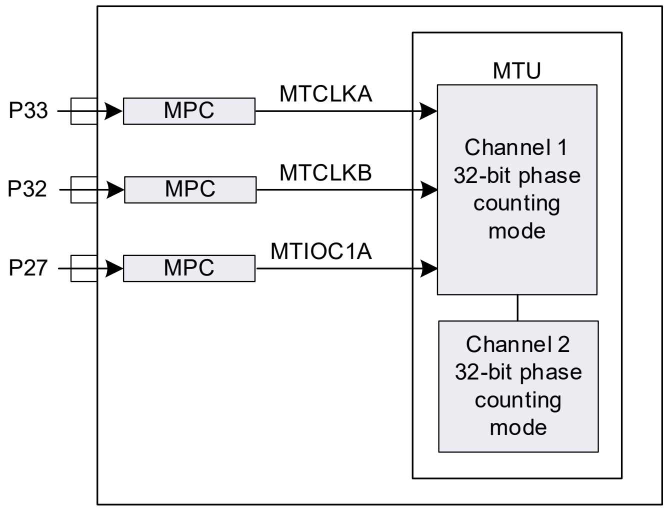
The following describes the MTU settings used in the sample code.

- MTU1 (channel 1) and MTU2 (channel 2)
  - 32-bit phase counting mode with cascade-connected timers is used.
  - External clocks MTCLKA (for A-phase) and MTCLKB (for B-phase) are used.
  - For [Up count], select: [High-Rising, Low-Falling, Rising-Low, Falling-High]
  - For [Down count], select: [High-Falling, Low-Rising, Rising-High, Falling-Low]
  - MTU1.TGRALW is used as an input capture register.
    - Input capture with MTU1.TGRALW is used as the clearing source for the timer counter.
  - Input capture is performed at a rising edge of the MTIOC1A pin input.

These settings can be specified by using Smart Configurator.

For details about how to specify the settings, refer to section 3.3.3.

The following shows an overview of the hardware configuration for this sample code.



MTU1, MTU2: 32-bit phase counting mode
MPC: Sets pins to be used from general purpose I/O port to peripheral function I/O port

Figure 3.9 Hardware Configuration for the Sample Code

3.3.2 Operation Details

This section describes the operation of this sample code.

The TCNT counter performs up-counting or down-counting according to the waveforms of input A-phase and B-phase signals, based on the edge detection settings for the count-up and count-down sources in phase counting mode 1.

When the sample code detects a rising edge at MTIOC1A, it uses TGRALW to perform input capture, which triggers clearing the TCNTLW counter.

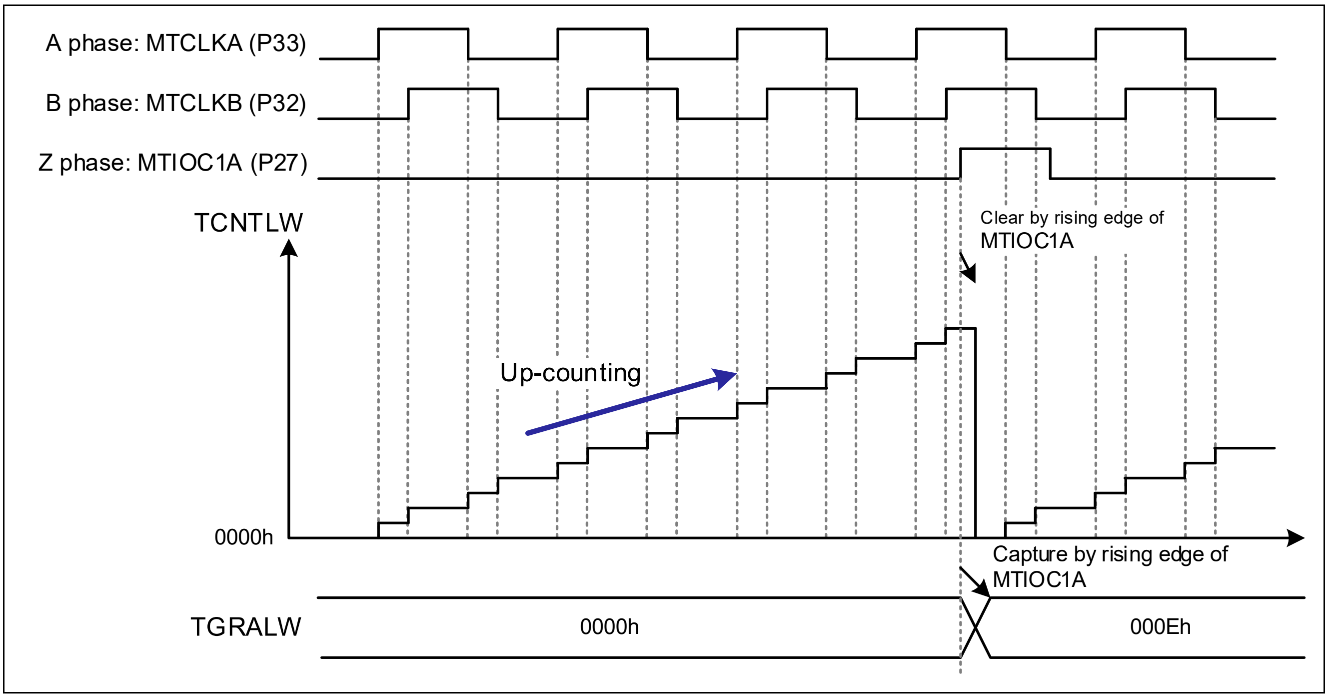


Figure 3.10 Operation of the Sample Code

3.3.3 Smart Configurator Settings

In the sample code, Smart Configurator is used to add the MTU as follows. For details about how to add a component, refer to section 3.1.4, Adding a Component.

Table 3.6 Adding a Component

Item	Description
Component	Phase counting mode timer
Configuration name	Config_MTU1_MTU2
Operation	32-bit phase counting mode with cascade-connected timers
Resource	MTU1_MTU2

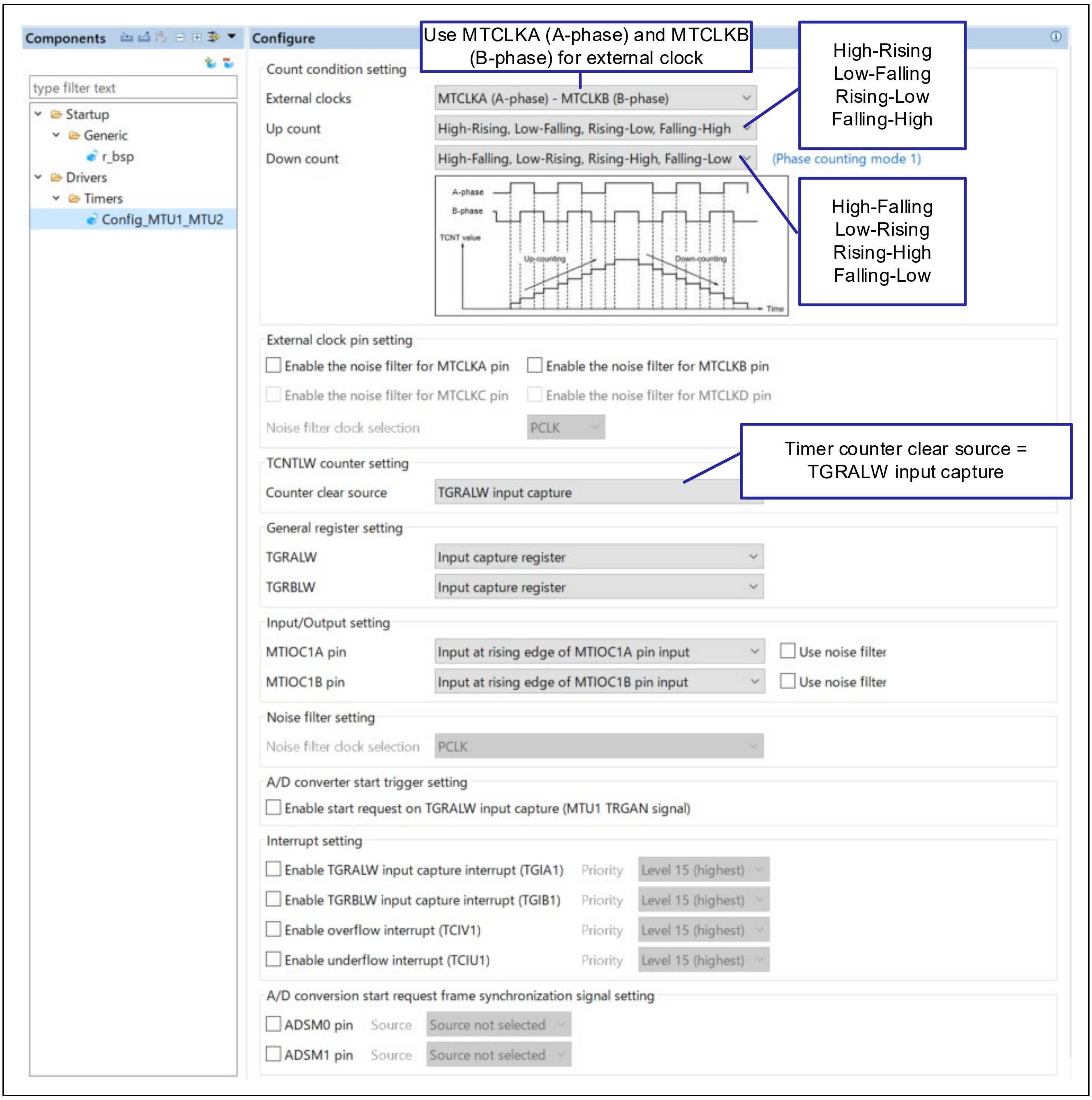


Figure 3.11 MTU1_MTU2 Settings

If you specify the settings as shown in Figure 3.11, the MTIOC1B function, which is unused, is assigned to the pin. Therefore, clear the check box to cancel the assignment as follows.

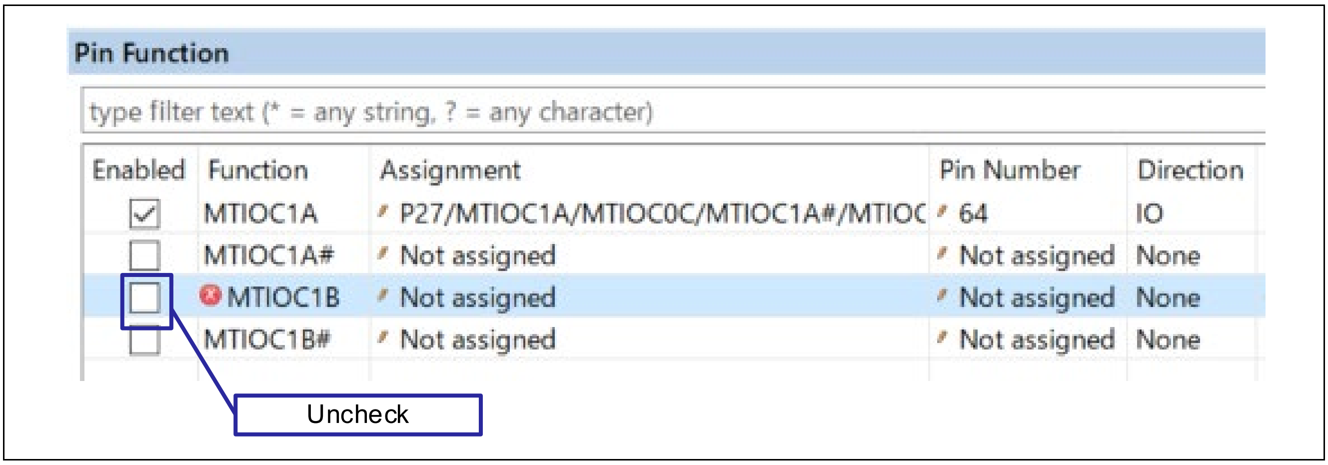


Figure 3.12 Pin Function Settings

If you specify the settings in Figure 3.12, a configuration error occurs in the phase counting mode timer as follows. However, this error is not harmful in this sample code.

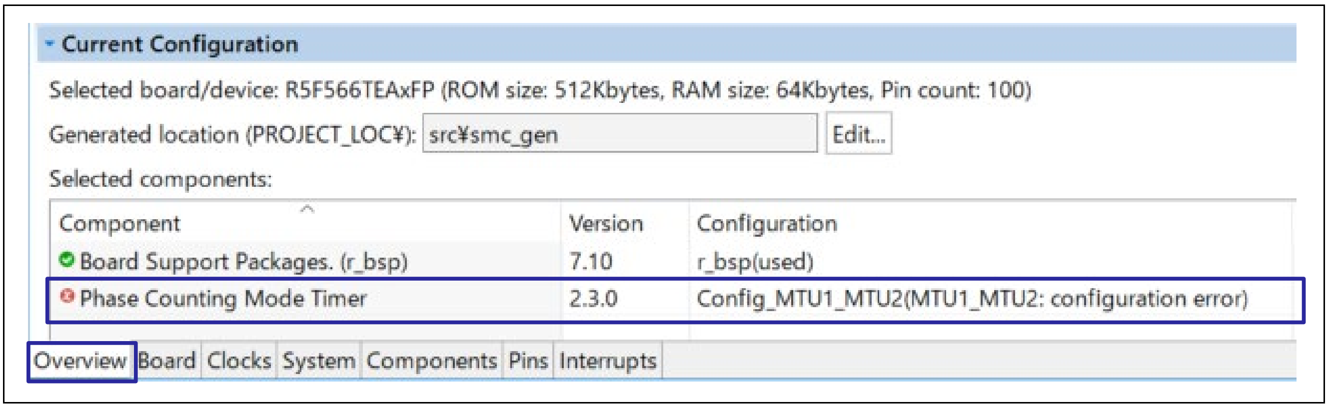


Figure 3.13 Smart Configurator Window

3.3.4 Flowchart

The following shows the processing that was added to the main function after code was generated by Smart Configurator.

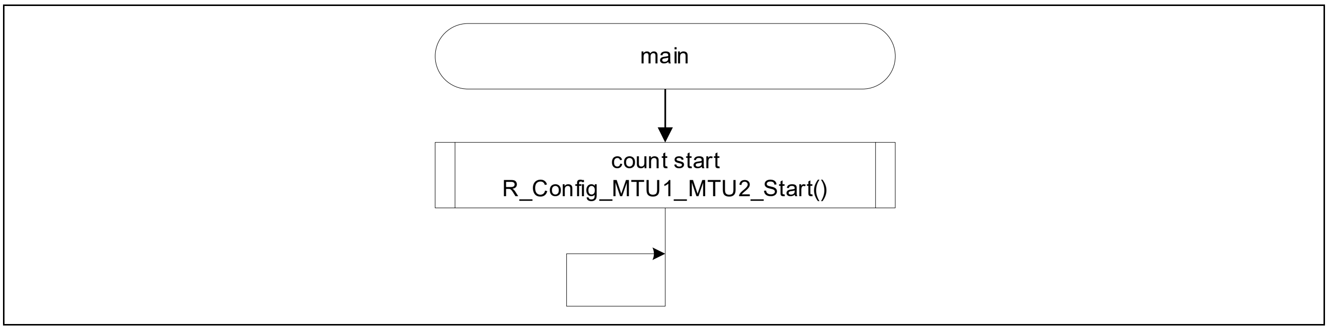


Figure 3.14 main Function

3.3.5 Related Operation

3.3.5.1 To Use the TGRALW or TGRBLW Register as an Output Compare Register

In the Smart Configurator environment for e[2] studio Version 2022-04, normally, the TGRALW and TGRBLW registers can be configured as input capture registers only.

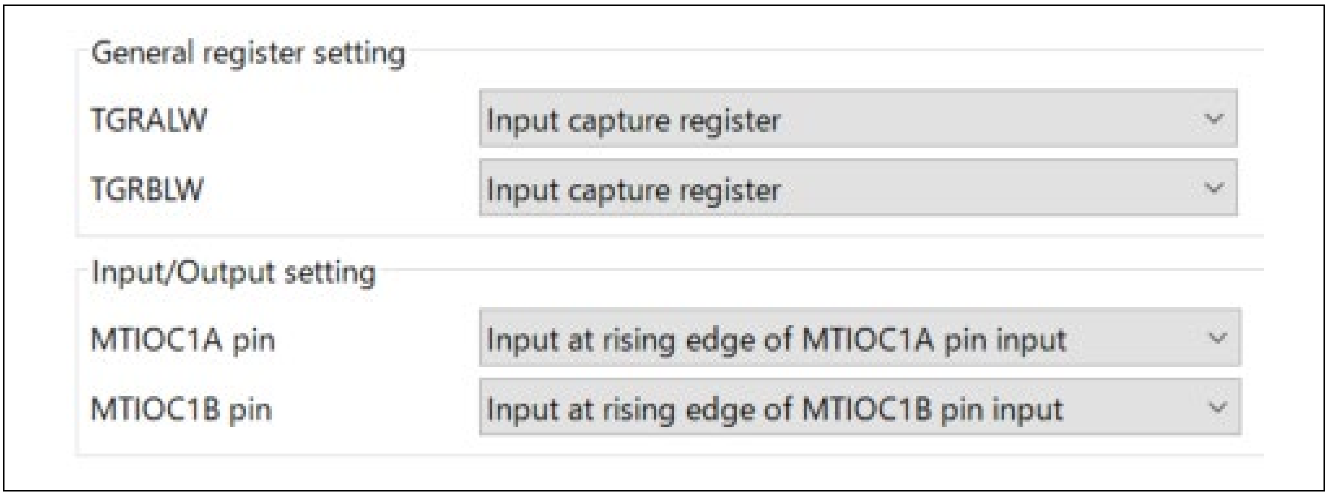


Figure 3.15 Smart Configurator Settings

However, you can also use TGRALW or TGRBLW as an output compare register to output any value of your choice from MTIOC1A or MTIOC1B by using the compare match function.

The following describes how to specify the settings so that the TGRBLW register is used as an output compare register and, when a compare match occurs, the output level of the MTIOC1B pin (initially set for low-level output) is toggled.

After code is generated, add the following code to the R_Config_MTU1_MTU2_Create_UserInit function in the Config_MTU1_MTU2_user.c file. Make sure that you select the MTIOC1B check box that you cleared in Figure 3.12.

```
void R_Config_MTU1_MTU2_Create_UserInit(void)
{
    /* Start user code for user init. Do not edit comment generated here */

    MTU1.TGRBLW = 0x00000032U;
    MTU1.TIOR.BYTE |= 0x30U;

    /* End user code. Do not edit comment generated here */
}
```

In TGRBLW, set any compare value of your choice. For details about the value to be set for TIOR, refer to section 22.5.1, Input/Output Timing, and section 22.2.6, Timer I/O Control Register (TIOR) in the RX66T Group User's Manual: Hardware.

3.3.5.2 To Use the MTIOC1A or MTIOC1B Register as a General-Purpose I/O Port

You can also use the MTIOC1A or MTIOC1B register as a general-purpose I/O port.

The following describes how to use MTIOC1B as PA4 (general-purpose I/O port).

In Smart Configurator, specify the following settings:

- In the Configure pane of the Components tab, specify the MTIOC1B settings as shown in Figure 3.15, and then specify the pin settings as shown in Figure 3.12.
- Register "PORT" in the component selection window, and then specify the PA4 settings.

3.3.6 Precautions

3.3.6.1 Precautions in the case of "LWA = 1"

If the TMDR3.LWA bit is set to 1b, MTU1 and MTU2 are cascade-connected to operate as a single 32-bit timer. This timer is controlled by using the MTU1.TCR, MTU1.TCR2, MTU1.TIOR, and MTU1.TMDR1 registers. The settings of the MTU2.TCR, MTU2.TCR2, MTU2.TIOR, and MTU2.TMDR1 registers do not take effect. These settings cannot be used to access 16-bit MTU1 and MTU2 registers (TCNT, TGRA, and TGRB registers). The input capture and compare match functions using MTU2 are also disabled. Therefore, linkage with ELC triggered by MTU2 is unavailable.

A cascade connection of MTU1 and MTU2 enabled by setting the LWA bit to 1b can be used in phase counting mode only. It cannot be used in normal mode, PWM1 mode, or PWM2 mode.

Before you set the LWA bit to 1b, initialize the TCNT, TGRA, and TGRB registers of MTU1 and MTU2.

3.3.6.2 External Clock Input Pins Connected in 32-Bit Phase Counting Mode

The following table shows the combinations of external clock input pins that can be selected in 32-bit phase counting mode (LWA = 1).

Table 3.7 Clock Input Pins Available in 32-Bit Phase Counting Mode

Channel	External clock pin	
	A-phase	B-phase
MTU1_MTU2	MTCLKA	MTCLKB
	MTCLKC	MTCLKD
	MTCLKA#	MTCLKB#
	MTCLKC#	MTCLKD#
	MTCLKA#	MTCLKB
	MTCLKA	MTCLKB#
	MTCLKC#	MTCLKD
	MTCLKC	MTCLKD#

For details, refer to section 22.3.6, Phase Counting Mode in the RX66T Group User's Manual: Hardware.

3.3.6.3 Times When Counting Starts and the Counter Is Cleared in Phase Counting Modes

For details about the times when counting starts in phase counting modes, refer to (1), TCNT Count Timing in section 22.5.1, Input/Output Timing in the RX66T Group User's Manual: Hardware.

For details about the times when the counter is cleared, refer to (4), Timing for Counter Clearing by Compare Match/Input Capture in section 22.5.1, Input/Output Timing in the RX66T Group User's Manual: Hardware.

3.3.6.4 Specifications Applying When MTIOC1A/MTIOC1B and MTIOC2A/MTIOC2B Are Not Used

This sample code uses only MTIOC1A. The pin assigned to MTIOC1B, which is not used, can also be used for another function or port.

By using the TGRB register as a compare match register, you can output any value of your choice by using the compare match function.

3.4 32-Bit Phase Counting Mode (MTU1.TMDR3.LWA = 0)

- Applicable sample code file name: r01an6387_rx66t_mtu3_32_phase_cnt_cas.zip

3.4.1 Overview

This section describes how to use the 32-bit phase counting mode for the MTU.

This sample code uses MTU1 as a free-running counter and MTU2 for 16-bit phase counting mode. MTU2 uses phase counting mode 1, in which A-phase and B-phase signals from a 2-phase encoder are input to the MTCLKA and MTCLKB pins, and the number of pulses is counted. MTU1 is set to start counting when overflow or underflow occurs in MTU2.TCNT.

The sample code also inputs the Z-phase signal into MTIOC2A, and clears MTU1.TCNT and MTU2.TCNT upon detecting a rising edge.

The following describes the MTU settings used in the sample code.

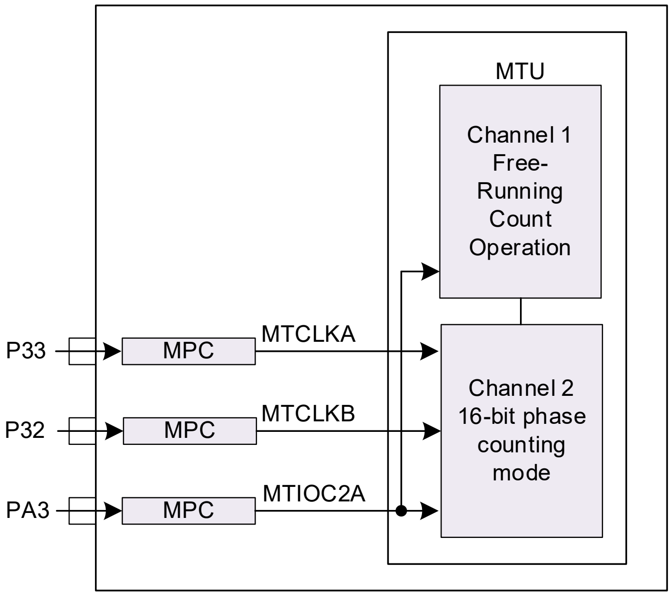
- MTU1 (channel 1)
  - — A normal mode timer is used.
  - — The timer clock counter operates based on an MTU2 overflow and underflow.
  - — MTU1.TGRA is used as an input capture register.
    - Input capture with MTU1.TGRA is used as the clearing source for the timer counter.
  - — Input capture is performed at a rising edge of the MTIOC1A pin input.

- MTU2 (channel 2)
  - — 16-bit phase counting mode is used.
  - — External clocks MTCLKA (for A-phase) and MTCLKB (for B-phase) are used.
  - — For [Up count], select:
[High-Rising, Low-Falling, Rising-Low, Falling-High]
  - — For [Down count], select:
[High-Falling, Low-Rising, Rising-High, Falling-Low]
  - — MTU2.TGRA is used as an input capture register.
    - Input capture with MTU1.TGRA is used as the clearing source for the timer counter.
  - — Input capture is performed at a rising edge of the MTIOC2A pin input.

These settings can be specified by using Smart Configurator.

For details about how to specify the settings, refer to section 3.4.3.

The following shows an overview of the hardware configuration for this sample code.



MTU1: Free-Running Count Operation
MTU2: 16-bit phase counting mode
MPC: Sets pins to be used from general purpose I/O port to peripheral function I/O port

Figure 3.16 Hardware Configuration for the Sample Code

3.4.2 Operation Details

This section describes the operation of this sample code.

The MTU2.TCNT counter performs up-counting or down-counting according to the waveforms of input A-phase and B-phase signals, based on the edge detection settings for the count-up and count-down sources in phase counting mode 1. Operation of MTU1.TCNT switches between up-counting and down-counting, depending on whether overflow or underflow occurs in MTU2.TCNT.

The input capture control register (TICCR) is set so that MTU1.TGRA and MTU2.TGRA perform input capture when detecting a rising edge of MTIOC2A, and the MTU1.TCNT and MTU2.TCNT counters are cleared by input capture performed by MTU2.TGRA.

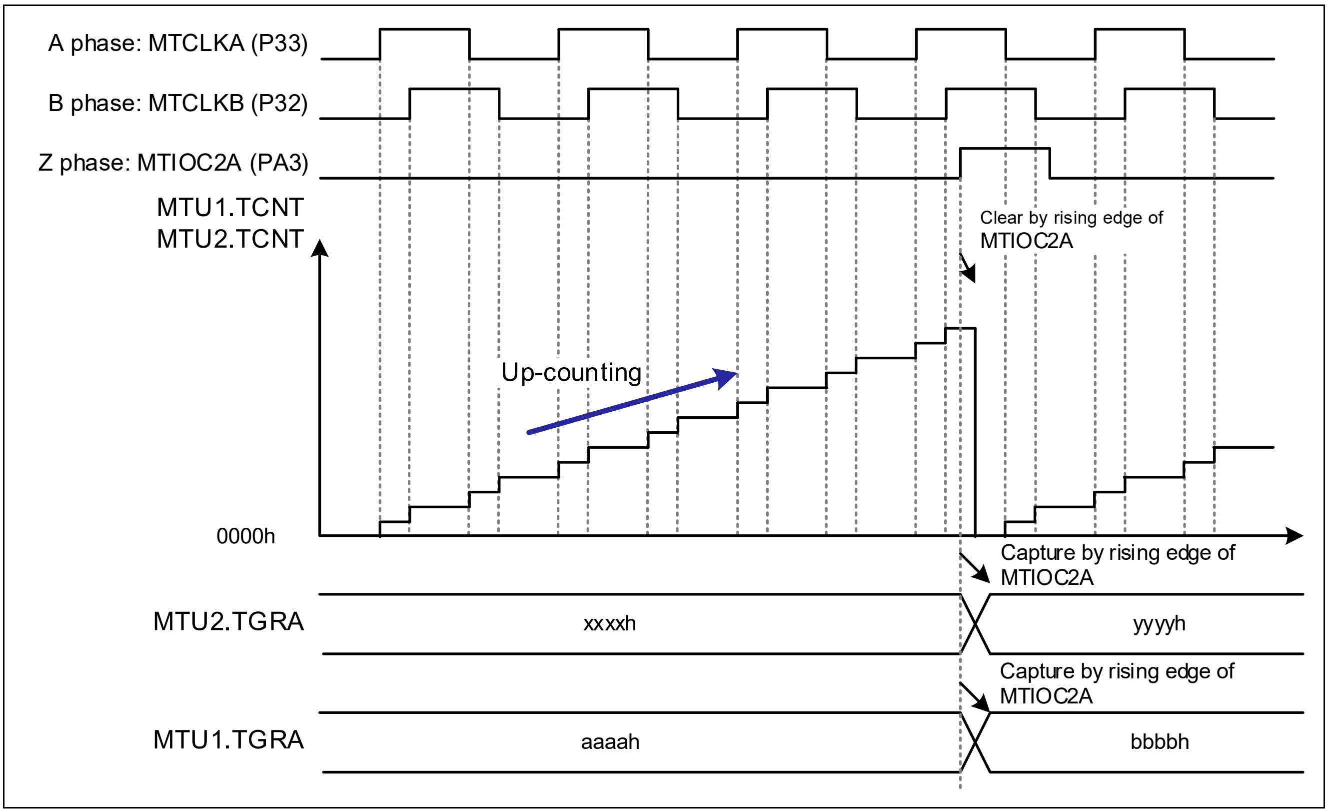


Figure 3.17 Operation of the Sample Code

3.4.3 Smart Configurator Settings

In the sample code, Smart Configurator is used to add the MTU as follows. For details about how to add a component, refer to section 3.1.4, Adding a Component.

Table 3.8 Adding a Component

Item	Description
Component	Normal mode timer
Configuration name	Config_MTU1
Operation	Two pins
Resource	MTU1

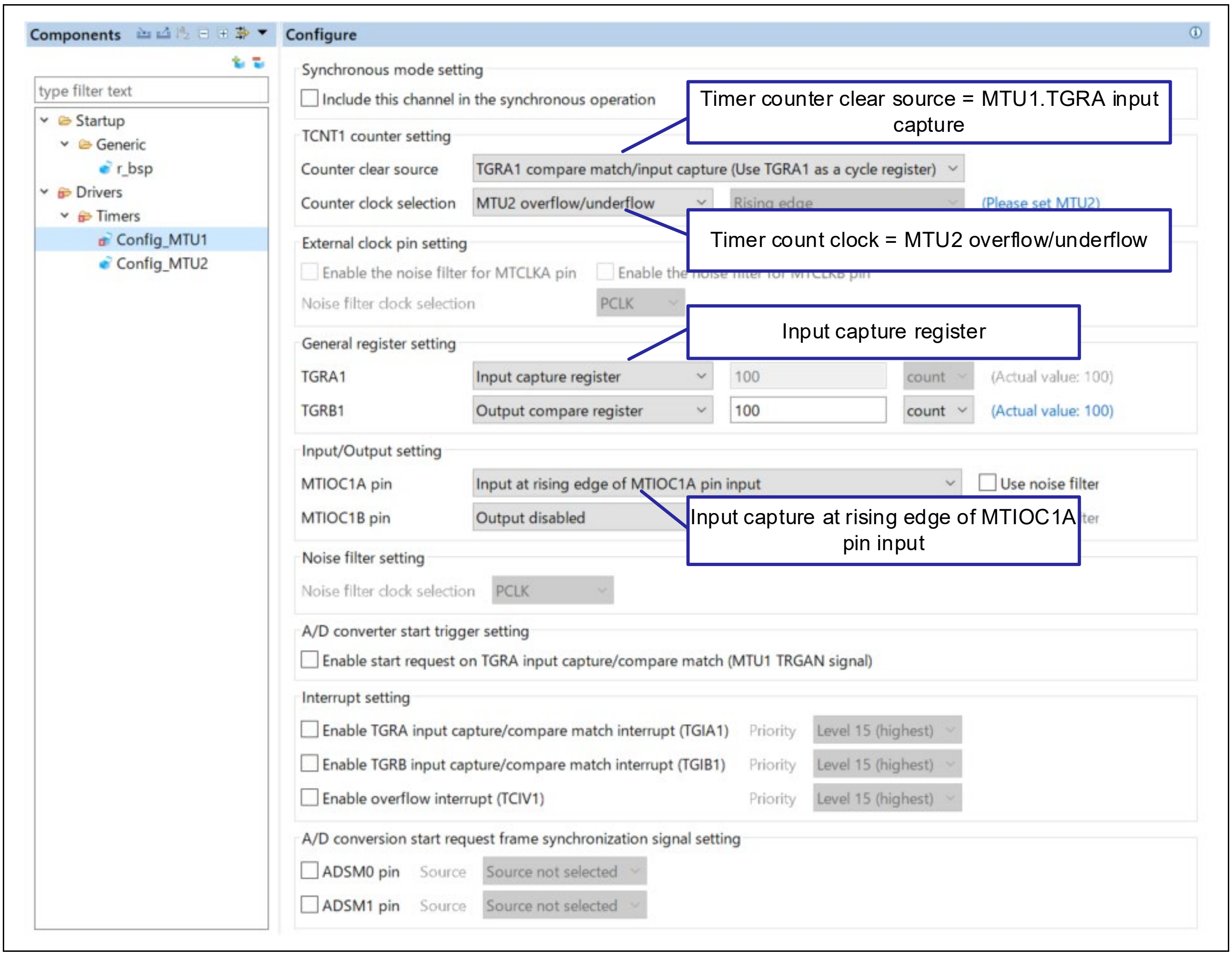


Figure 3.18 MTU1 Settings

Table 3.9 Adding a Component

Item	Description
Component	Phase counting mode timer
Configuration name	Config_MTU2
Operation	16-bit phase counting mode
Resource	MTU2

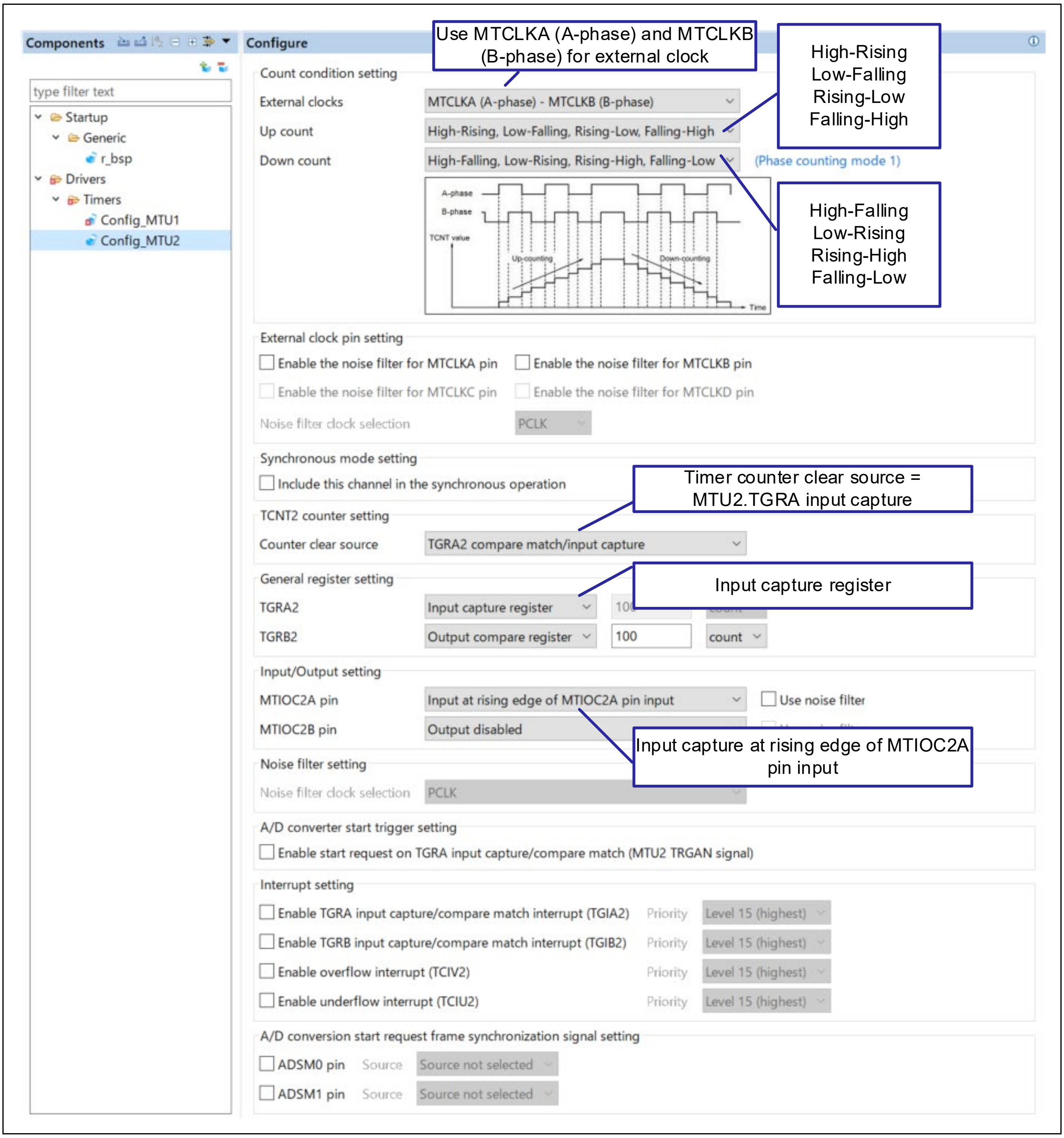


Figure 3.19 MTU2 Settings

If you specify the settings as shown in Figure 3.18, you must specify the MTIOC1A function and pin settings. Specify them as follows.

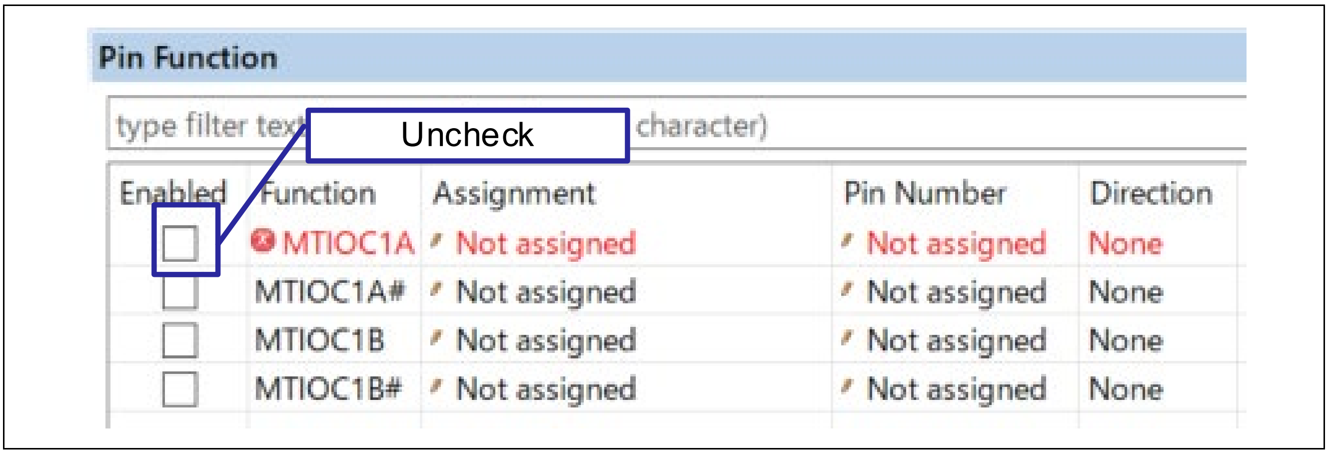


Figure 3.20 Pin Function Settings

If you specify the settings in Figure 3.20, a configuration error occurs in the normal mode timer as follows. However, this error is not harmful in this sample code.

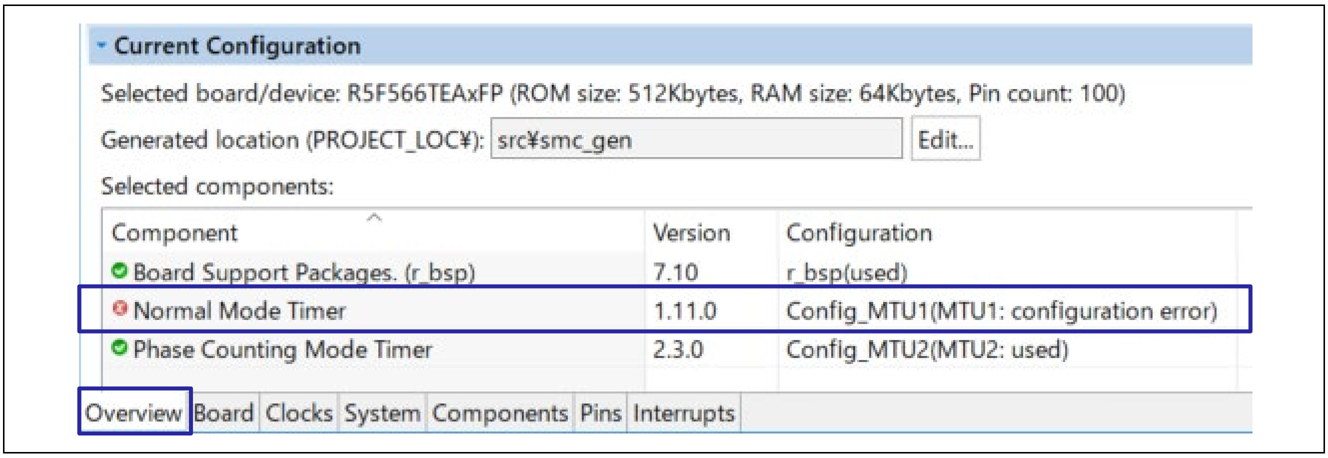


Figure 3.21 Smart Configurator Window

3.4.4 Flowchart

The following shows the processing that was added to the main function after code was generated by Smart Configurator.

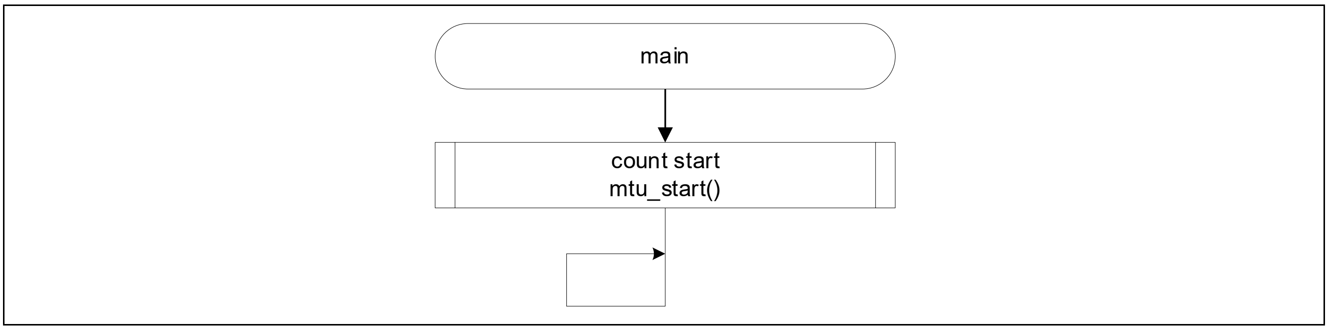


Figure 3.22 main Function

The count start function starts the counting for MTU1 and MTU2.

This function is a new function created after code is generated by Smart Configurator.

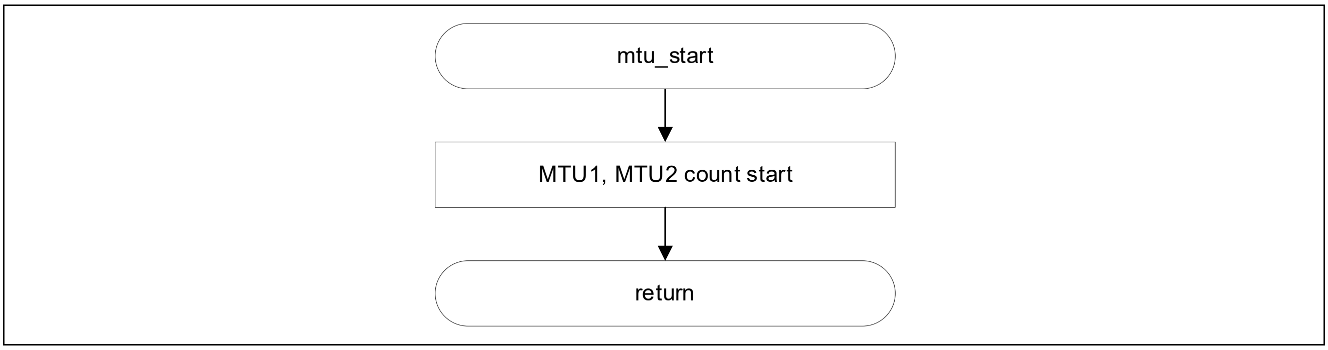


Figure 3.23 Count Start Function

The R_Config_MTU1_Create_UserInit user initialization function that is run before the main function is used to add MTIOC2A as a trigger to perform input capture with MTU1.TGRA. This function is invoked from the inside of the R_Config_MTU1_Create function.

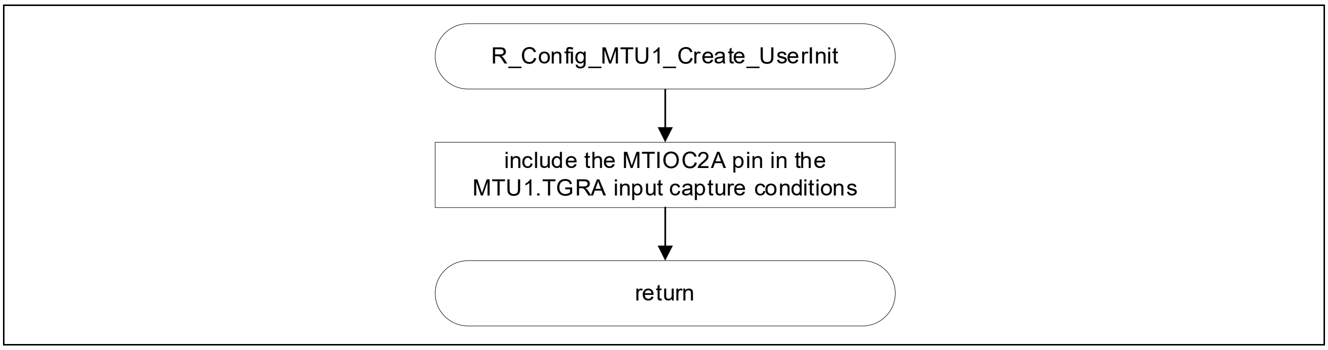


Figure 3.24 User Initialization Function

3.4.5 Precautions

3.4.5.1 Precautions in the case of "LWA = 0"

If the TMDR3.LWA bit is set to 0b, MTU1 and MTU2 operate as independent 16-bit timers. Therefore, the TCNTLW, TGRALW, and TGRBLW registers cannot be accessed.

3.4.5.2 External Clock Input Pins Connected in 32-Bit Phase Counting Mode

In 32-bit phase counting mode (LWA = 0), MTU2 is set in 16-bit phase counting mode. For details about the external clock input pin combinations that can be selected for MTU2, refer to section 3.2.5.1, External Clock Input Pins Connected in 16-Bit Phase Counting Mode.

3.4.5.3 Times When Counting Starts and the Counter Is Cleared in Phase Counting Modes

For details about the times when counting starts in phase counting modes, refer to (1), TCNT Count Timing in section 22.5.1, Input/Output Timing, in the RX66T Group User's Manual: Hardware.

For details about the times when the counter is cleared, refer to (4), Timing for Counter Clearing by Compare Match/Input Capture in section 22.5.1, Input/Output Timing in the RX66T Group User's Manual: Hardware.

3.4.5.4 Simultaneous Input Capture with Cascade-Connected MTU1.TCNT and MTU2.TCNT

In this sample code, input capture with MTU1.TCNT and input capture with MTU2.TCNT are performed simultaneously at a rising edge of MTIOC2A. The input capture control register (TICCR) is used to add MTIOC2A as a trigger to perform input capture with MTU1.TCNT.

For details, refer to section 22.3.4, Cascaded Operation and section 22.6.21, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection in the RX66T Group User's Manual: Hardware.

3.4.5.5 Specifications Applying When MTIOC1A/MTIOC1B and MTIOC2A/MTIOC2B Are Not Used

This sample code uses only MTIOC2A. The pins assigned to unused MTIOC1A/MTIOC1B and MTIOC2B can be used for other functions or ports.

By using the TGR register as a compare match register, you can output any value of your choice by using the compare match function.

4. GPTW Sample Code

4.1 General

4.1.1 Sample Code List

This application note provides the following types of sample code available with Smart Configurator.

These can be downloaded from the Renesas Electronics website.

Table 4.1 GPTW Sample Code List

Name	Description	Refer to
Phase counting mode r01an6387_rx66t_gptw_phase_cnt.zip	• Phase counting mode 1 • External trigger input is used for Z-phase.	4.2

4.1.2 Folder Structure

The following shows the main folders for the sample code.

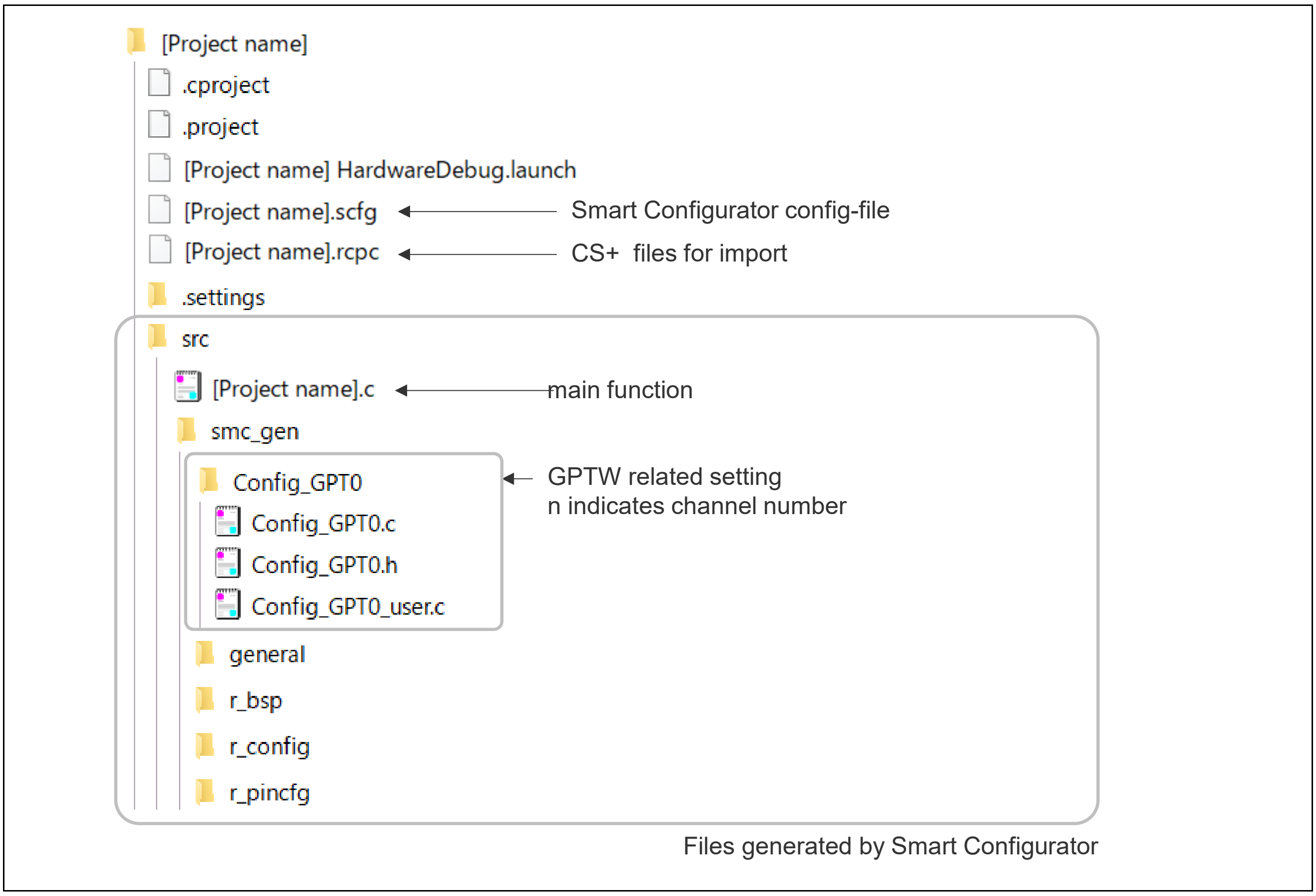


Figure 4.1 GPTW Folder Structure

4.1.3 File Structure

The following shows the main files for the sample code.

Table 4.2 GPTW File Structure

File name	Description
[Project-name].c	main function This is the main function. Smart Configurator generates this function with empty content. Add necessary processing according to the type of sample code.
Config_GPT*n*.c Note	R_Config_GPT*n*_Create function This function specifies the initial settings for the GPTW. Smart Configurator generates an initial setting function according to the settings specified in Smart Configurator. Smart Configurator generates processing that invokes this function. This function is invoked by the R_SystemInit function that is run before the main function is run.
	R_Config_GPT*n*_Start function This function starts the counting for the GPTW. Smart Configurator generates this function. In the sample code, this function is invoked by the main function.
	R_Config_GPT*n*_Stop function This function stops the counting for the GPTW. Smart Configurator generates this function. This function is not used in the sample code.
Config_GPT*n*_user.c Note	r_Config_GPT*n*_Create_UserInit function This is a user function that specifies the initial settings for the GPTW. Smart Configurator generates this function with empty content. Add necessary processing according to the type of sample code. This function is invoked at the end of the R_Config_GPT*n*_Create function that is generated by Smart Configurator.
	r_Config_GPT*n*_*interrupt-name*_interrupt function This is an interrupt handler function. Smart Configurator generates this function with empty content. Add necessary processing according to the type of sample code.
Config_GPT*n*.h Note	This is a header file that defines GPTW-related functions. This file is included by the r_smc_entry.h file that is generated by Smart Configurator. To use GPTW-related functions, include the r_smc_entry.h file.

Note: *n* indicates the channel number.

4.1.4 Adding a Component

In the sample code, Smart Configurator is used to add the GPTW as follows.

Table 4.3 Adding a Component

Item	Description
Component	General PWM Timer ((1) in the following figure)
Configuration name	In the sample code, the initial value is used.
Work mode	Refer to the section for the relevant type of sample code. ((2) in the following figure)
Resource	Refer to the section for the relevant type of sample code. ((3) in the following figure)

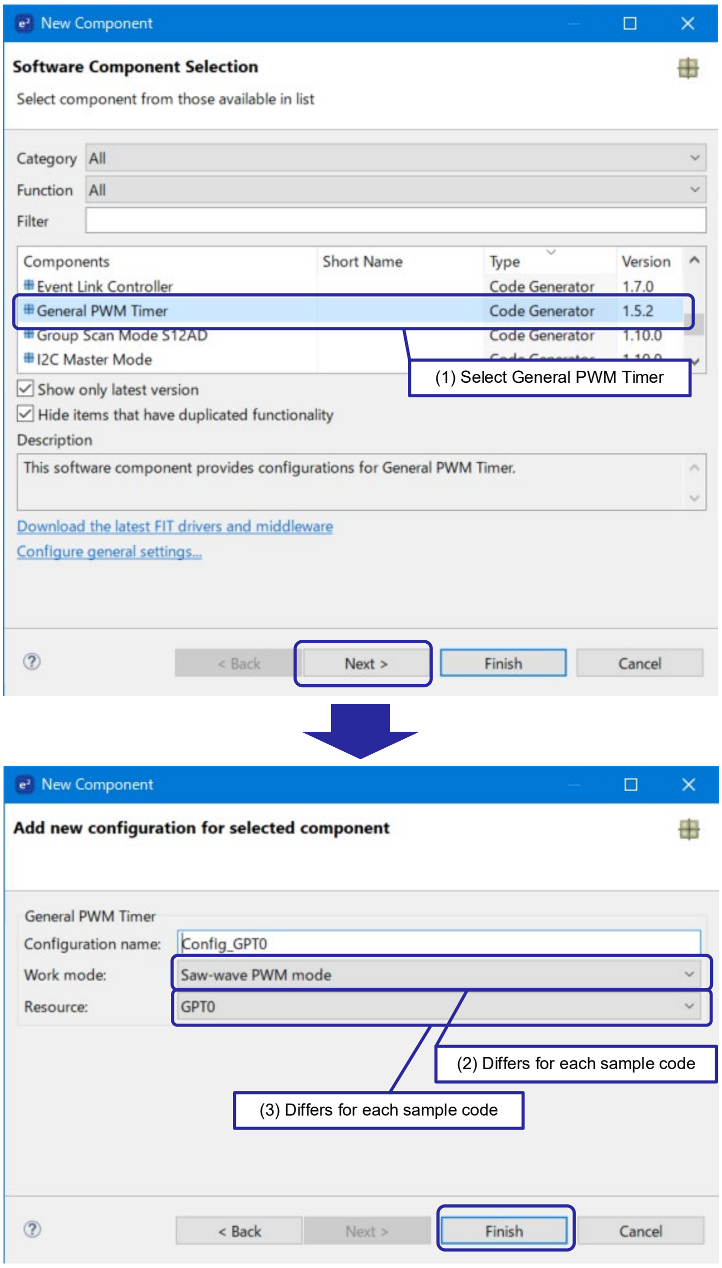


Figure 4.2 Adding a Component

4.1.5 Pin Configuration

Figure 4.3 shows an example of configuring pins by using Smart Configurator.

Before you configure pins, make sure that the GPTW has been configured. For details about how to configure the GPTW, refer to the "Smart Configurator Settings" section appropriate for the relevant type of sample code.

Pin configuration is performed inside the R_Config_GPT*n*_Create function that is generated by Smart Configurator.

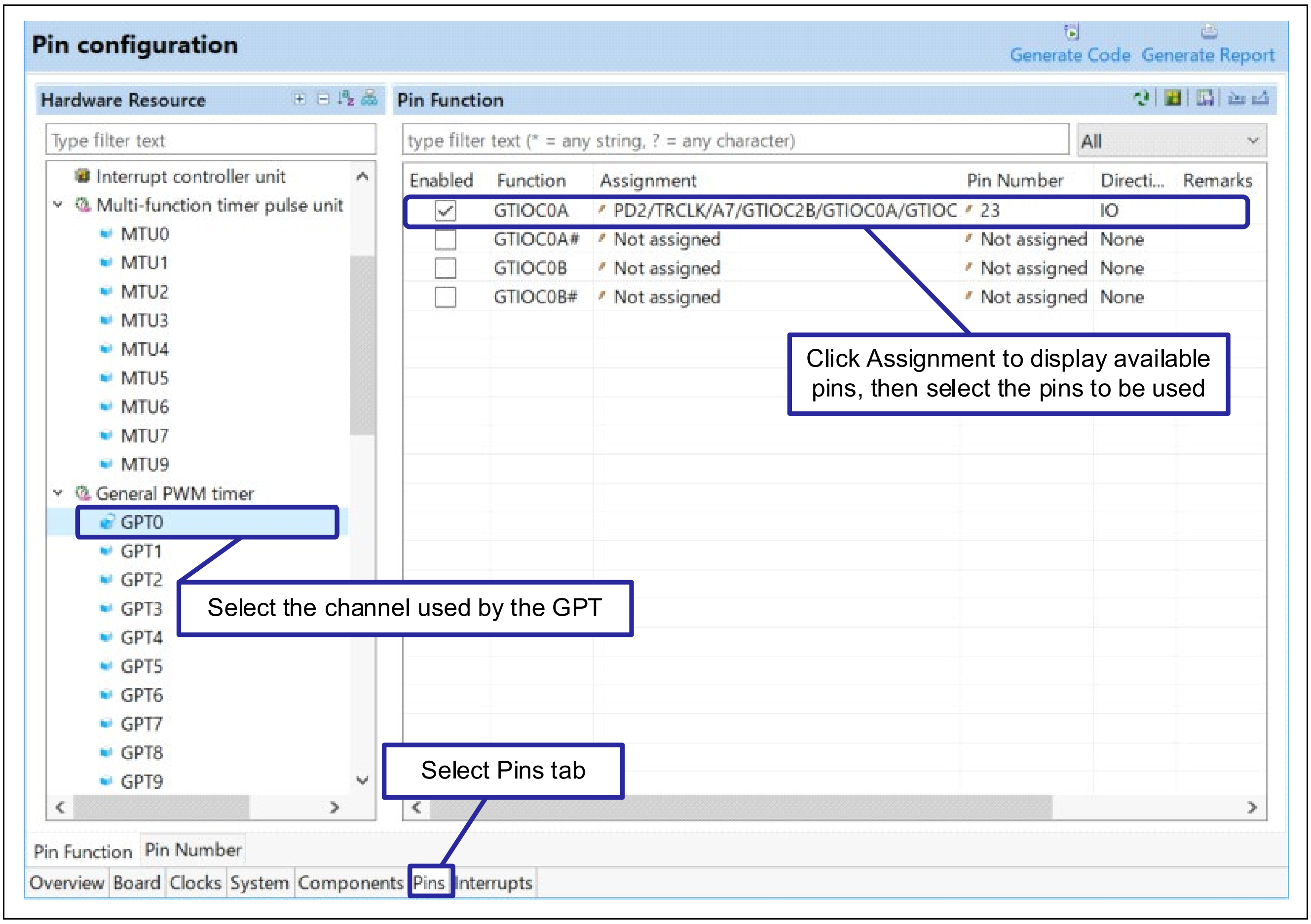


Figure 4.3 Pin Configuration

4.1.6 Interrupt Configuration

Figure 4.4 shows an example of setting an interrupt by using Smart Configurator. For details about software configurable interrupt A, refer to section 14.4.5.1, Software Configurable Interrupt A in the RX66T Group User's Manual: Hardware.

Before you configure interrupts, make sure that the GPTW has been configured. For details about how to configure the GPTW, refer to the "Smart Configurator Settings" section appropriate for the relevant type of sample code.

Interrupt configuration is performed inside the R_Config_GPT*n*_Create, R_Config_GPT*n*_Start, and R_Config_GPT*n*_Stop functions that are generated by Smart Configurator.

Interrupt handler functions are created with names in the "r_Config_GPT*n*_*interrupt-name*_interrupt" format in the Config_GPT*n*_user.c file that is generated by Smart Configurator.

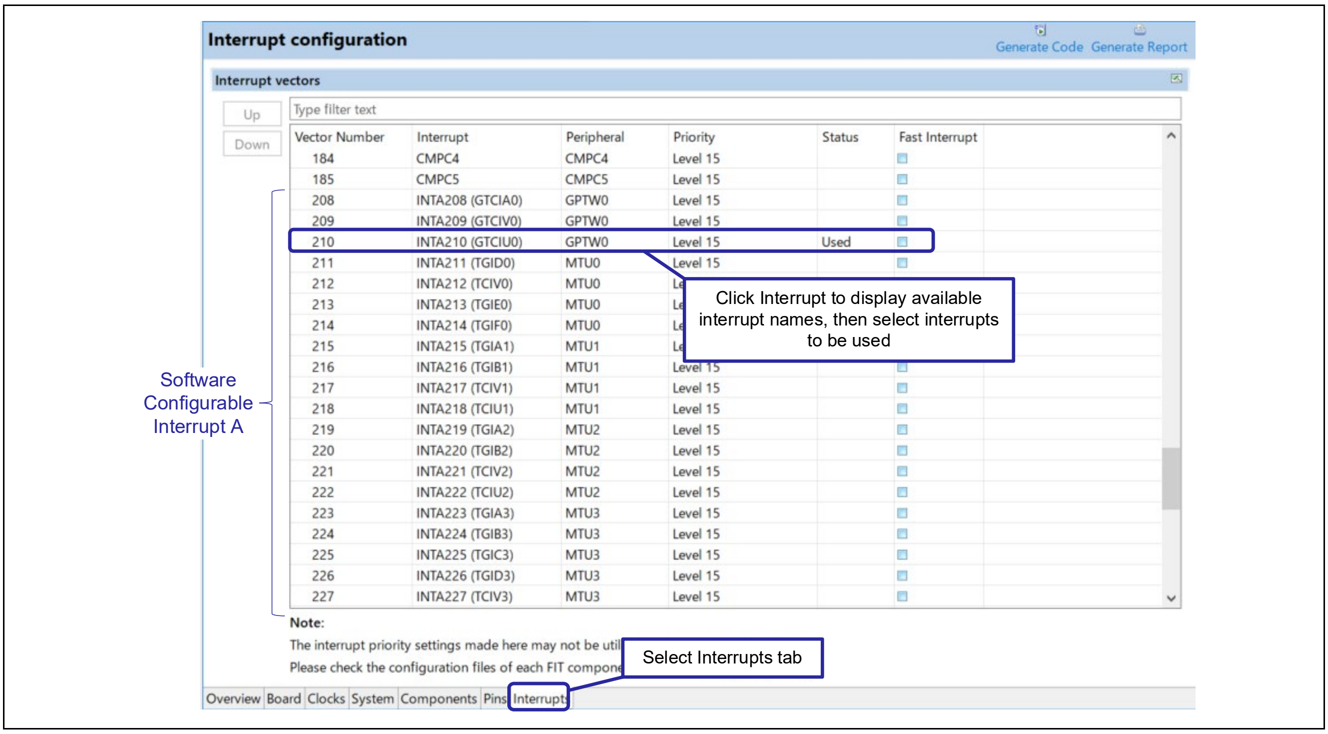


Figure 4.4 Interrupt Configuration

In the initial settings in the Interrupts tab of Smart Configurator, only GTCIE0, GTCIF0, and GDTE0 are selected as interrupts for GPTW. To use the interrupts that were selected in the Components tab, they must also be selected in the Interrupts tab. The following figure shows an example when there is a missing selection and an error message that is output.

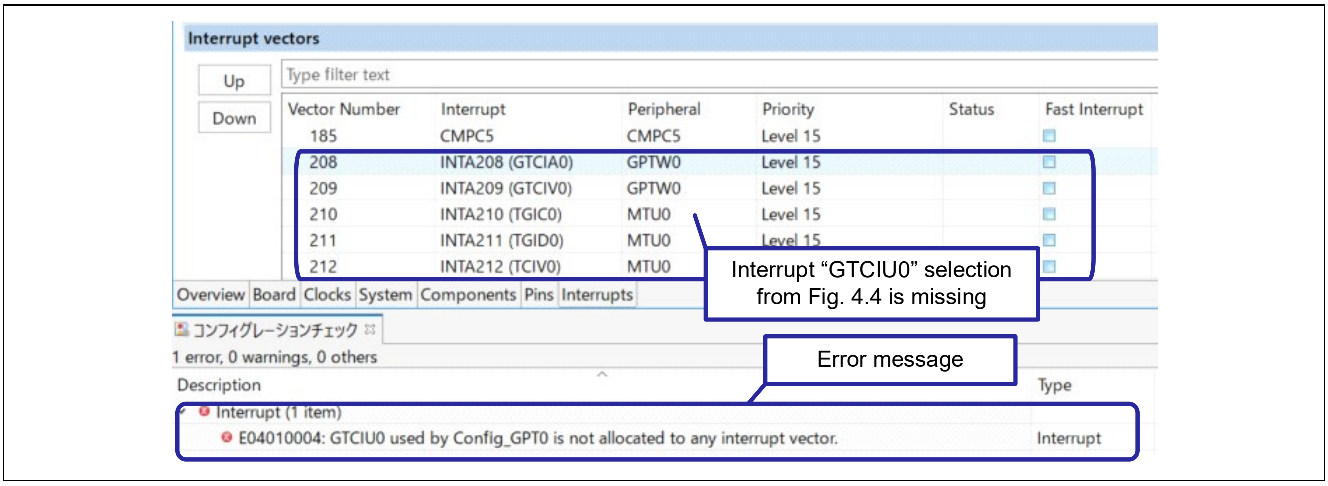


Figure 4.5 Interrupt Configuration (with a Missing Selection)

4.2 Phase counting mode

- Applicable sample code file name: r01an6387_rx66t_gptw_phase_cnt.zip

4.2.1 Overview

This section describes how to use the phase counting mode for GPTW.

This sample code uses phase counting mode 1, in which A-phase and B-phase signals from a 2-phase encoder are input to the GTIOC3A and GTIOC3B pins, and the number of pulses is counted.

The Z-phase signal is input via POEG to the external trigger input pin (GTETRGA). When a rising edge is detected, GPTW3.GTCNT is cleared.

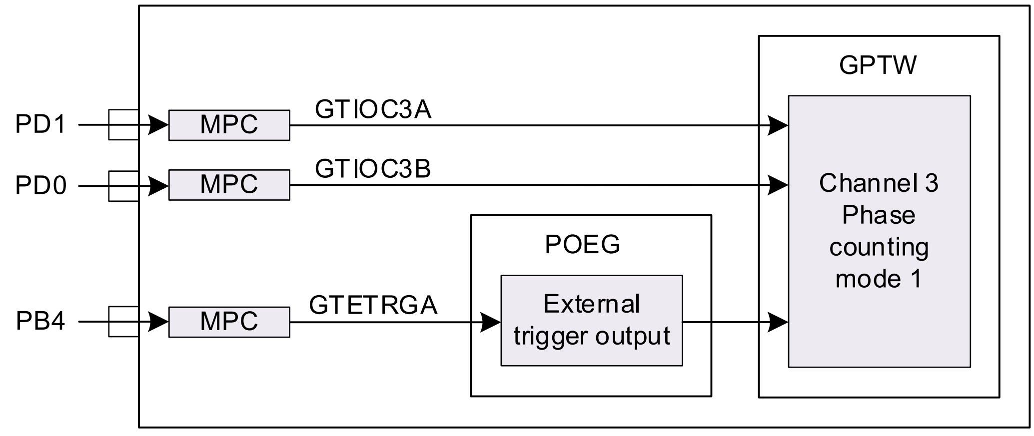
The following describes the GPTW and POEG settings used in the sample code.

- GPTW3 (channel 3)
  - Sawtooth-wave PWM mode is used.
  - The frequency of the timer count clock is 80 MHz (PCLKC / 2).
  - The timer operates at intervals of 0x20000 clock cycles.
    - The timer value changes in the up-counting direction.
    - The initial value of the counter is 0.
  - The GTIOC3A pin is used for an input pin.
  - The GTIOC3B pin is used for an input pin.
  - The counter is cleared when a rising edge of the GTETRGA pin input is detected.
  - Triggers of a count-up operation are as follows:
    Rising of GTIOC3A input while GTIOC3B input 0
    Falling of GTIOC3A input while GTIOC3B input 1
    Rising of GTIOC3B input while GTIOC3A input 1
    Falling of GTIOC3B input while GTIOC3A input 0
  - Triggers of a count-down operation are as follows:
    Rising of GTIOC3A input while GTIOC3B input 1
    Falling of GTIOC3A input while GTIOC3B input 0
    Rising of GTIOC3B input while GTIOC3A input 0
    Falling of GTIOC3B input while GTIOC3A input 1
- POEG
  - The GTETRGA pin settings are enabled.

These settings can be specified by using Smart Configurator.

For details about how to specify the settings, refer to section 4.2.3.

The following shows an overview of the hardware configuration for this sample code.



GPTW3: Phase counting mode 1

POEG: Outputs GTETRGA pin as external trigger to GPTW

MPC: Sets pins to be used from general purpose I/O port to peripheral function I/O port

Figure 4.6 Hardware Configuration for the Sample Code

4.2.2 Operation Details

This section describes the operation of this sample code. The GTCNT counter performs up-counting or down-counting according to the waveforms of input A-phase and B-phase signals, based on the edge detection settings for the count-up and count-down sources in phase counting mode 1.

The sample code clears the counter when detecting a rising edge at the GTETRGA external trigger input pin.

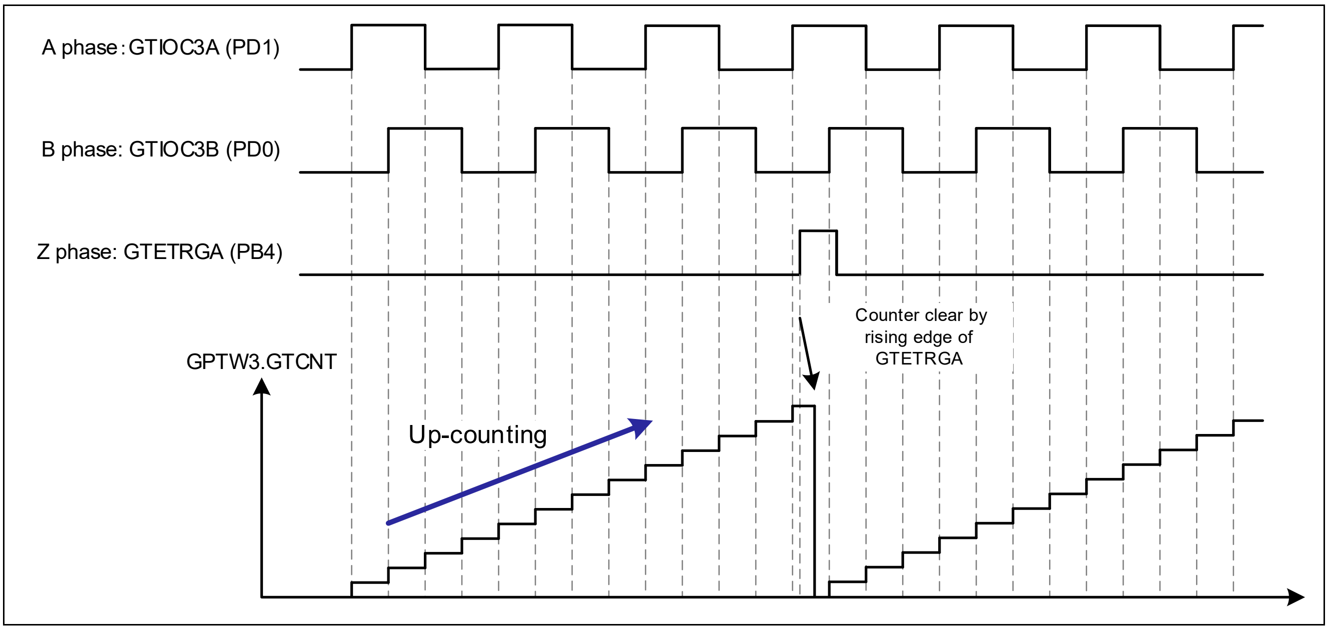


Figure 4.7 Operation of the Sample Code

4.2.3 Smart Configurator Settings

In the sample code, Smart Configurator is used to add the GPTW as follows. For details about how to add the GPTW component, refer to section 4.1.4, Adding a Component.

Table 4.4 Adding a Component (GPTW3)

Item	Description
Component	General PWM timer
Configuration name	Config_GPT3
Work mode	Sawtooth-wave PWM mode 1
Resource	GPT3

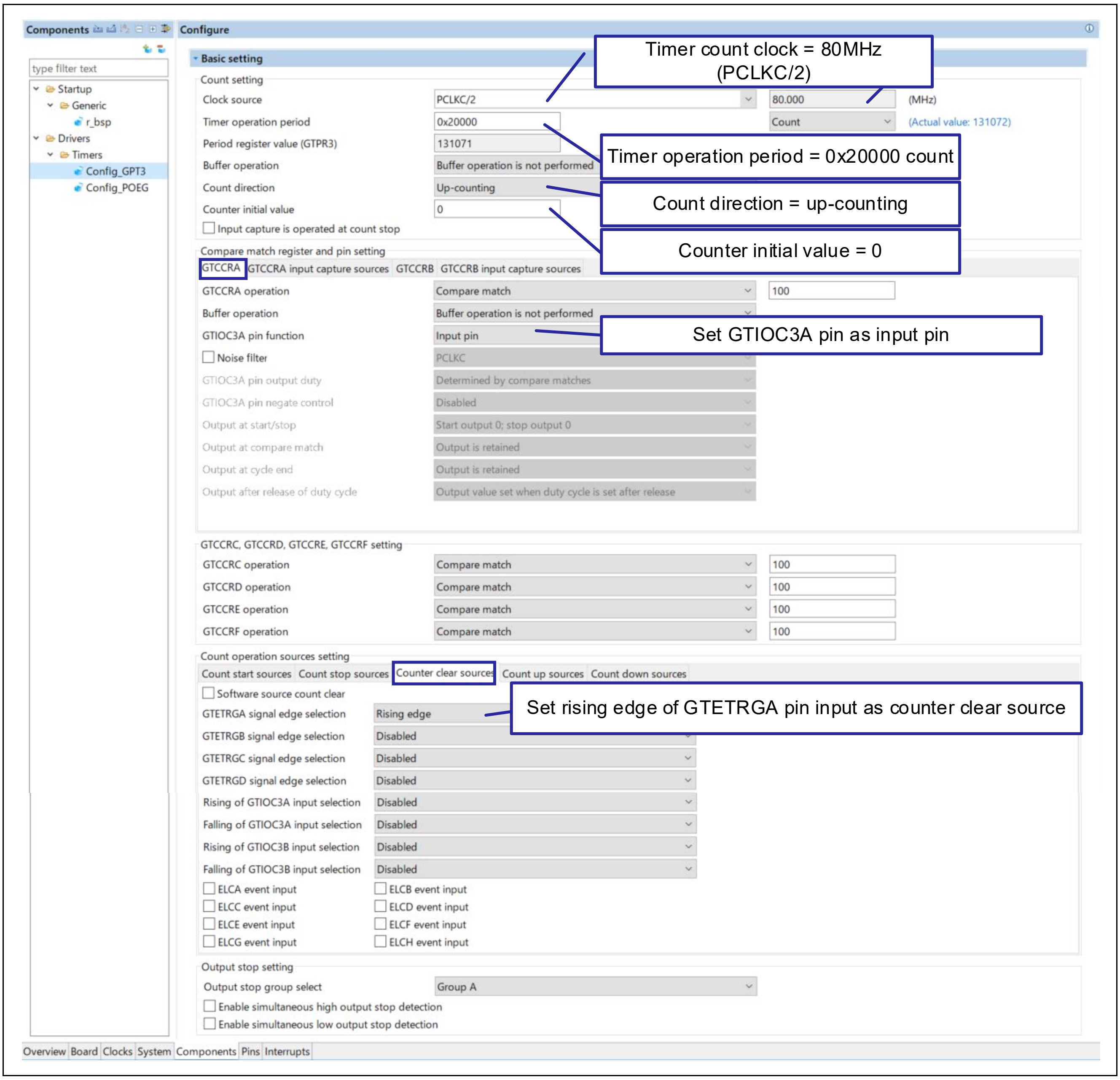


Figure 4.8 GPT3 Settings (1/4)

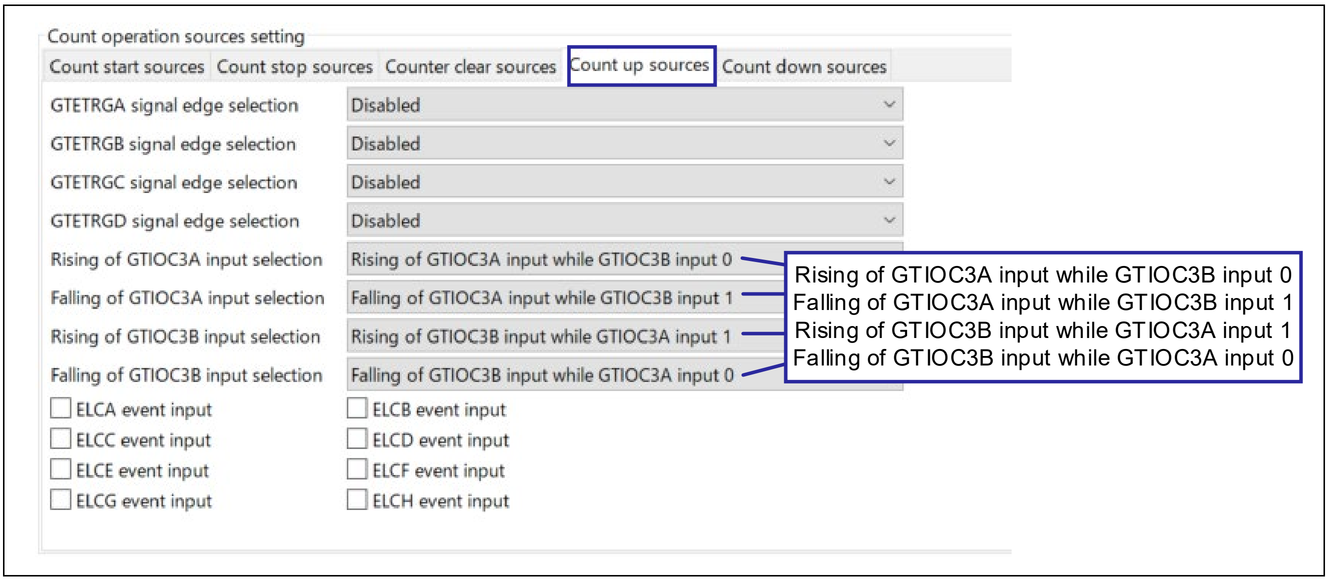


Figure 4.9 GPT3 Settings (2/4)

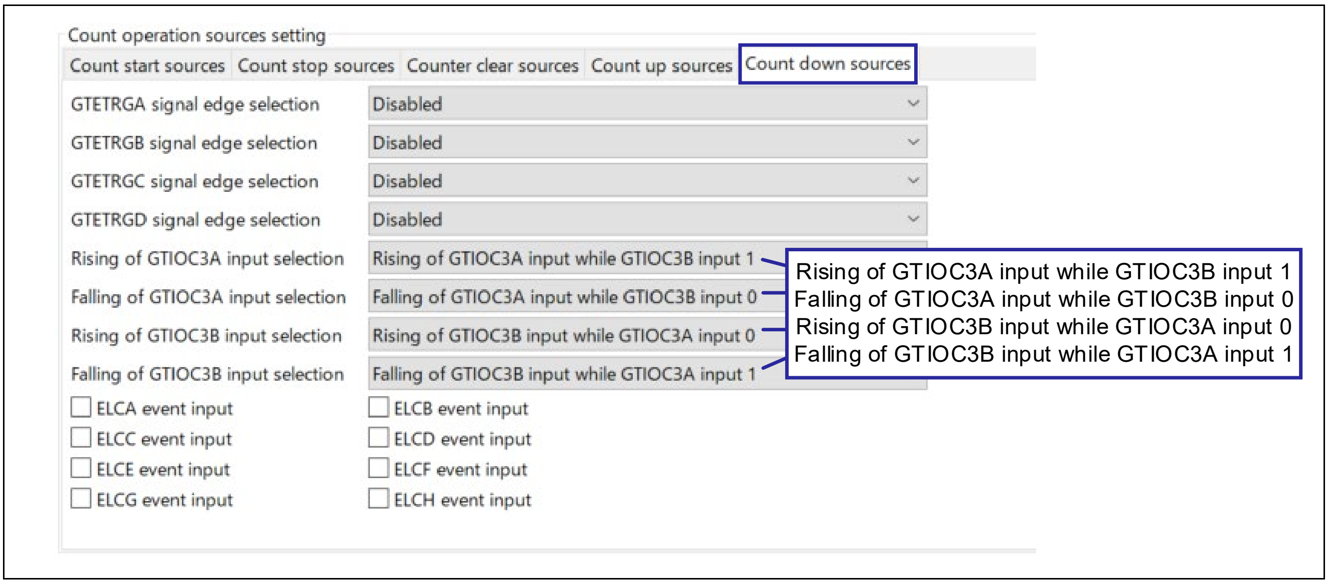


Figure 4.10 GPT3 Settings (3/4)

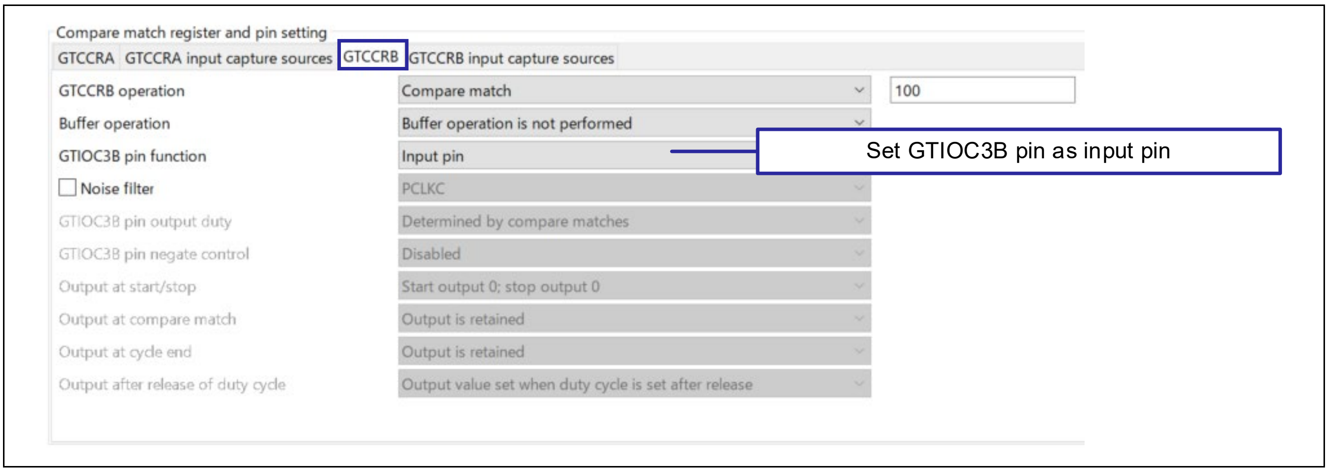


Figure 4.11 GPT3 Settings (4/4)

In order to use the external trigger input pin, add the POEG component as follows.

Table 4.5 Adding a Component (POEG)

Item	Description
Component	Port Output Enable
Configuration name	Config_POEG
Resource	POEG

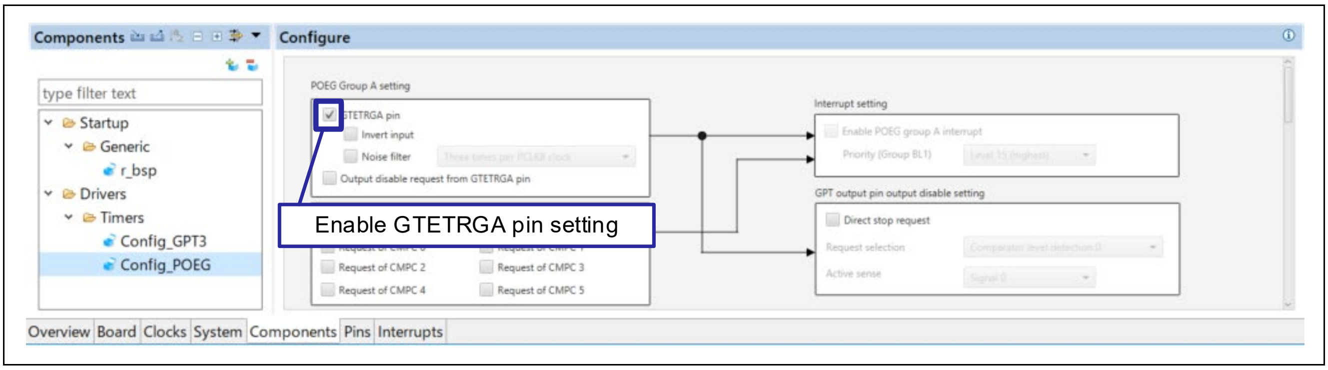


Figure 4.12 POEG Settings

4.2.4 Flowchart

The following shows the processing that was added to the main function after code was generated by Smart Configurator.

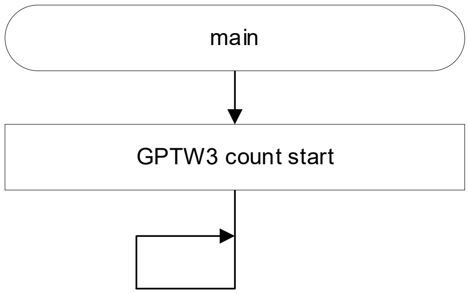


Figure 4.13 main Function

4.2.5 Precautions

4.2.5.1 Starting the Counting in Phase Counting Mode

In this sample code, the counting is started by an event counting operation (up-counting or down-counting operation triggered by a hardware source). Therefore, the CST bit of the GTCR general PWM timer control register is set to 1b in the main function.

The "Software source count start" check box in Smart Configurator is not used because code that controls the GTSTR general PWM timer software start register is generated in the R_Config_GPT*n*_Start function.

For details, refer to (4), Event Count Operation (In Up-Counting by Hardware Source) and (5), Event Count Operation (In Down-Counting by Hardware Source) in section 24.3.1.1, Counter Operation in the RX66T Group User's Manual: Hardware.

4.2.5.2 When the Counter Is Cleared by a Hardware Source

In this sample code, the external trigger input as a counter-clearing hardware source is used for event count operation. Therefore, the GTCNT counter is cleared in synchronization with PCLKC after GPTW outputs a clearing source.

The counter clearing timing differs depending on the hardware source and clock used.

For details, refer to the following in the RX66T Group User's Manual: Hardware:
• (6), Counter Clearing Operation, in section 24.3.1.1, Counter Operation
• Figure 24.74, Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTETRGA Pin (During Counting Triggered by Hardware Source), in section 24.3.7.3, Hardware Clear Operation

4.2.5.3 Order of Priority in Events

If a count-up or count-down operation triggered by the hardware source set by the GTUPSR or GTDNSR register and a counter clearing operation triggered by the hardware source set by the GTCSR register collide with each other, the counter clearing operation takes priority.

For details, refer to (1), GTCNT Counter in section 24.10.5, Order of Priority in Events in the RX66T Group User's Manual: Hardware.

5. How to Import a Project

The sample code is provided in the form of an e² studio project. This chapter describes how to import a project into e² studio and CS+. After the import is complete, confirm the build and debugger settings.

5.1 Import into e² studio

If you use the sample code with e² studio, use the following procedure to import the project into e² studio.

Note that the screenshots in the following procedure might be slightly different from the screens actually displayed, depending on the version of e² studio you are using.

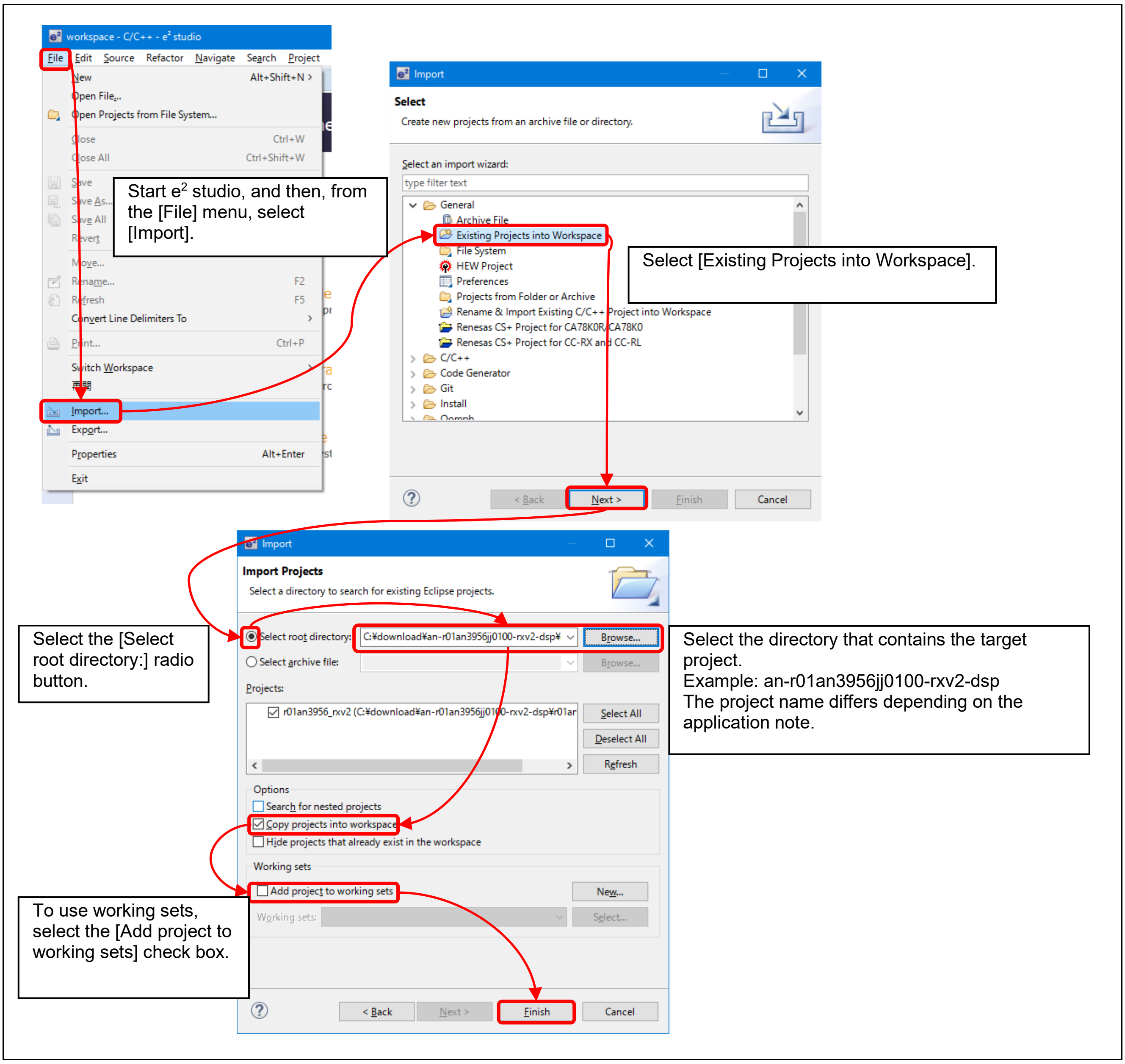


Figure 5.1 How to Import a Project into e² studio

5.2 Import into CS+

If you use the sample code with CS+, use the following procedure to import the project into CS+.

Note that the screenshots in the following procedure might be slightly different from the screens actually displayed, depending on the version of CS+ you are using.

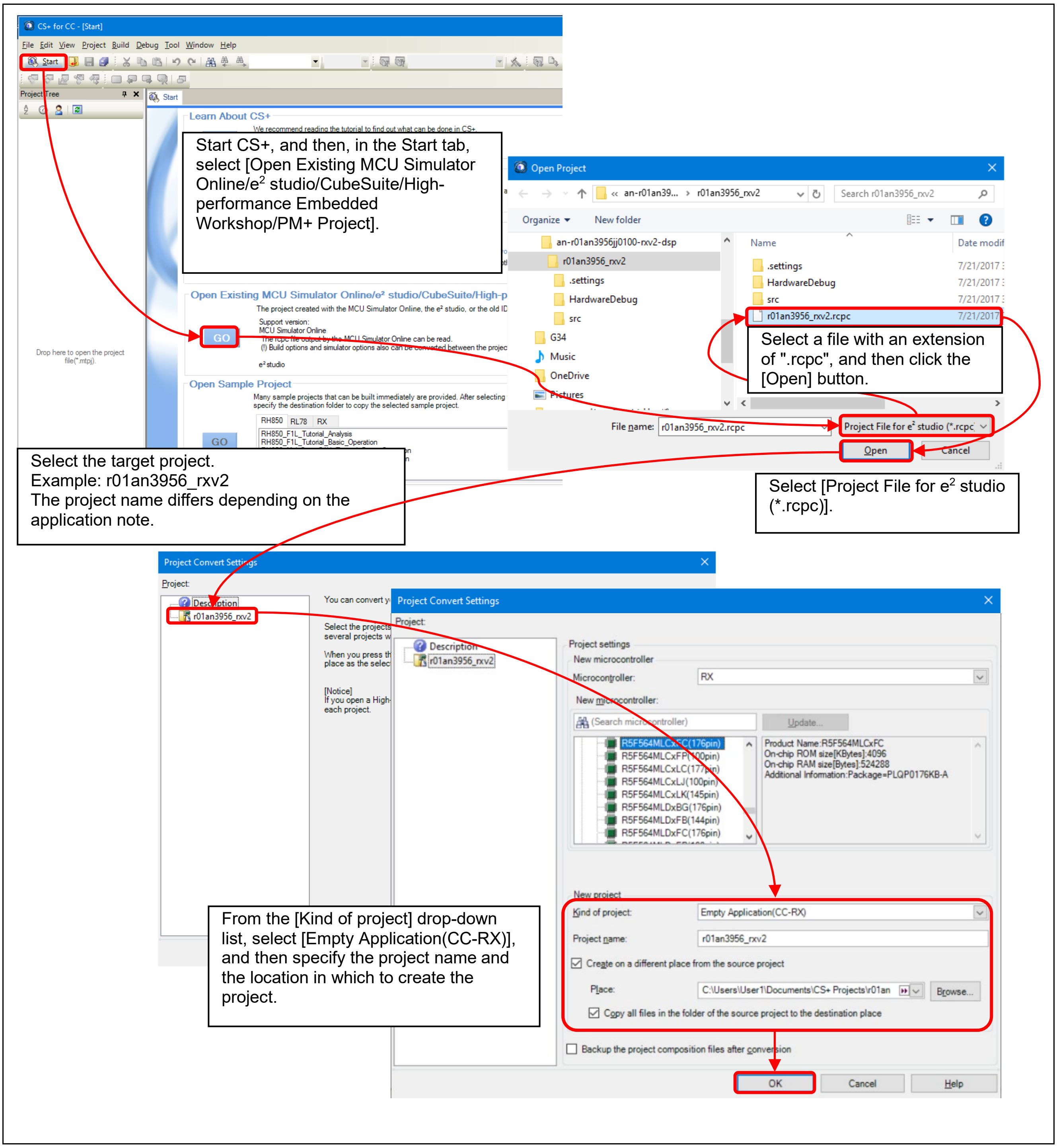


Figure 5.2 How to Import a Project into CS+

6. Reference Documents

- User's Manual: Hardware
  RX66T Group User's Manual: Hardware (R01UH0749)
  (The latest version is available at the Renesas Electronics website.)
- Technical Updates/Technical News
  (The latest version is available at the Renesas Electronics website.)
- User’s Manual: Development Environment
  RX Family CC-RX Compiler User's Manual (R20UT3248)
  (The latest version is available at the Renesas Electronics website.)
- User’s Manual: Development Environment
  RX66T Group Renesas Starter Kit User's Manual (R20UT4150)
  (The latest version is available at the Renesas Electronics website.)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	June 29, 2022	—	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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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