RX Family

PWM Output Methods Using MTU3/GPTW

Introduction

This application note describes the method of PWM output using the MTU3d and GPTW.

RX66T Group microcontrollers (MCUs) are equipped with the Multi-Function Timer Pulse Unit 3 (MTU3d) and the General-Purpose PWM Timer (GPTW), which can generate pulse-width modulation (PWM) waveforms.

The descriptions in this application note target RX Family devices equipped with the MTU3 and the GPTW. When using this application note with Renesas MCUs other than the RX66T Group, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Target Device

RX Family devices equipped with the MTU3 and GPTW

Confirmed Devices

RX66T Group

The Multi-Function Timer Pulse Unit 3 is referred to as “MTU” throughout this document.

Contents

1. MTU and GPTW Specifications

The following table lists the main functions related to PWM output from the MTU and GPTW.

Table 1.1 PWM Output Functions

Item		MTU	GPTW
Basic Functions	No. of channels	9 channels	10 channels
	Counter accuracy	16-bit Note 1	32-bit
	Max. operating frequency	Same as CPU frequency	Same as CPU frequency
	Synchronous operation	✓	✓
Common Functions	PWM output	✓	✓
	Complementary PWM output	✓	✓
	PWM output protection function (POE)	✓	✓
	Laterally asymmetric triangle-wave PWM	✓	✓
	Duty cycle 0% or 100% output	✓	✓
GPTW Proprietary Functions	Sawtooth-wave complementary PWM	—	✓
	Left/right dead time control	—	✓

Note 1: 32-bit is available with cascaded connection, but only 16-bit is available during PWM output.

The MTU supports different functions for each channel. The functions of each channel are listed below.

Table 1.2 Functions (by channel)

Item	MTU									GPTW Note 1
	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5	MTU6	MTU7	MTU9	
Count clocks	14	11	12	11	11	10	11	11	14	13
Synchronous operation	✓	✓	✓	✓	✓	-	✓	✓	✓	✓
No. of I/O pins	4	2	2	4	4	3	4	4	4	2
Buffer operation	✓	-	-	✓	✓	-	✓	✓	✓	✓
PWM mode 1Note 2	2	1	1	2	2	-	2	2	2	-
PWM mode 2Note 2	4	2	2	-	-	-	-	-	4	-
Complementary PWM Mode 1/2/3Note 2	-	-	-	2	4	-	2	4	-	-
Reset-synchronized PWM modeNote 2	-	-	-	2	4	-	2	4	-	-
Sawtooth-wave PWM modeNote 2	-	-	-	-	-	-	-	-	-	2
Sawtooth-wave one-shot pulse modeNote 2	-	-	-	-	-	-	-	-	-	2
Triangle-wave PWM mode 1/2/3Note 2	-	-	-	-	-	-	-	-	-	2
Automatic dead time setting function	-	-	-	-	-	-	-	-	-	✓

Note 1: Indicates the function for 1 channel. The GPTW has 10 channels for the same function.

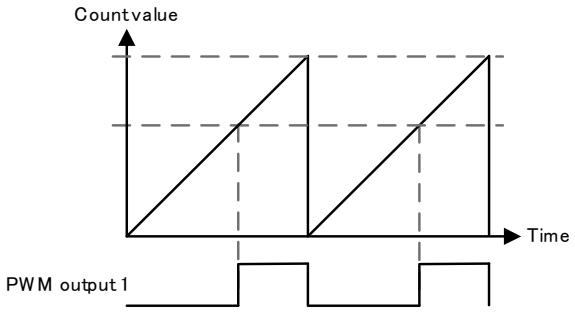
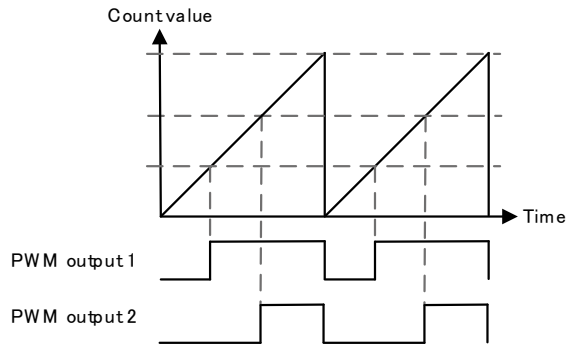
Note 2: Number indicates number of PWM output pins. The positive-phase and negative-phase each count as one pin.

The following sections provide PWM function details of MTU and GPTW and the differences in specifications.

1.1 Operating Mode and Output Waveform

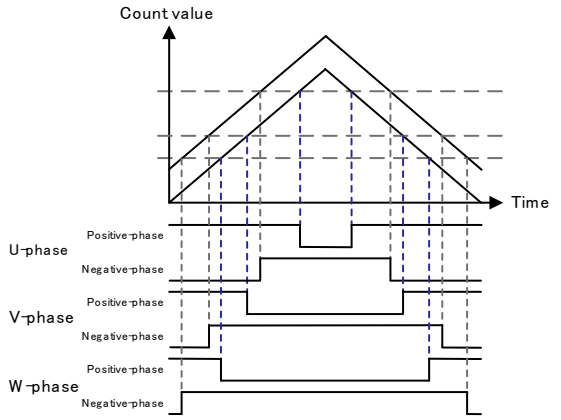
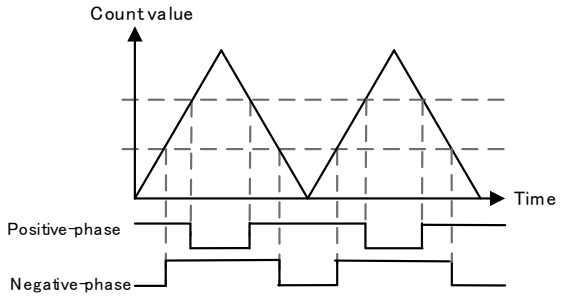
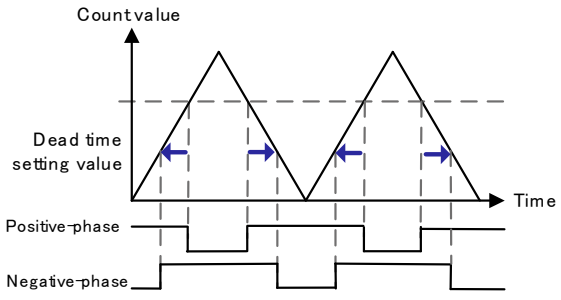
This section describes the relationship between the modes and output waveforms of the MTU and GPTW. The PWM output modes and output waveforms for MTU and GPTW are listed in the table below.

Table 1.3 PWM Output

Output Waveform	Timer	Description
	MTU	• PWM mode 1 — Up to 14 phases of PWM waveforms can be output — One period register and one duty register are used per PWM output — Synchronous operation can be enabled by synchronizing two or more phases
	GPTW	• Sawtooth-wave PWM mode — Up to 20 phases of PWM waveforms can be output — One period register and the same number of duty registers as PWM outputs are used per two PWM outputs — Synchronous operation can be enabled by synchronizing three or more phases
	MTU	• PWM mode 2 — Up to 12 phases of PWM waveforms can be output — One period register and the same number of duty registers as PWM outputs are used per multiple PWM outputs — Synchronous operation can be enabled by synchronizing three or more phases
	GPTW	• Sawtooth-wave PWM mode — Up to 20 phases of PWM waveforms can be output — One period register and the same number of duty registers as PWM outputs are used per two PWM outputs — Synchronous operation can be enabled by synchronizing three or more phases

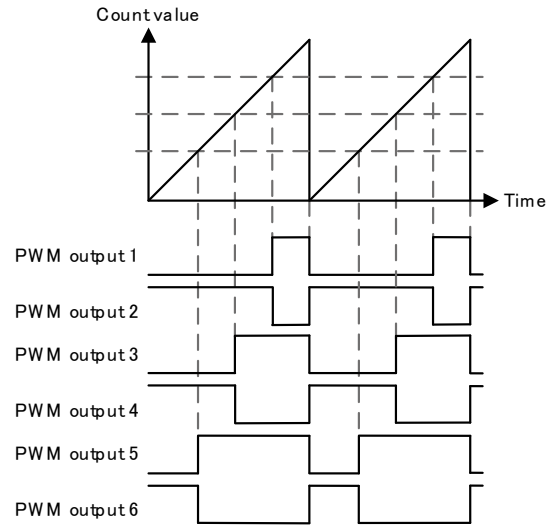
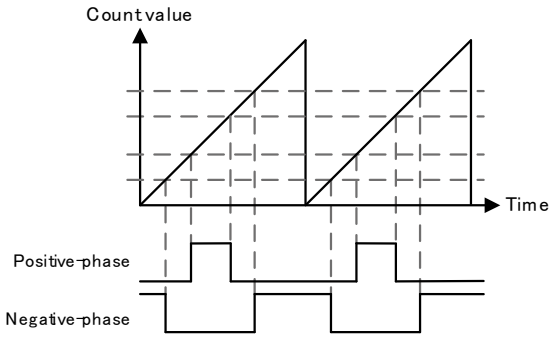
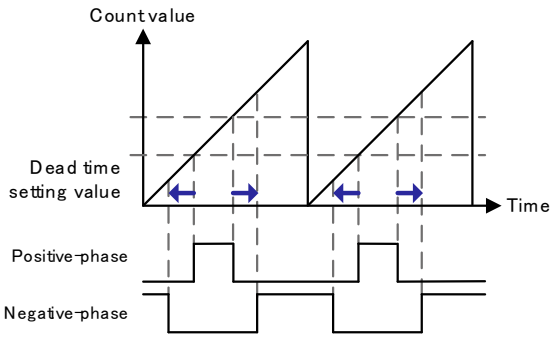
The following describes the modes and output waveforms of the MTU and GPTW complementary PWM output (triangle-waves).

Table 1.4 Complementary PWM Modes (triangle-waves)

Output Waveform	Timer	Description
	MTU	• Complementary PWM modes 1/2/3 — Maximum of 2 channels in 3-phase complementary — Duty cycle can be changed by changing one compare value per 1 phase — Automatic dead time setting function
	GPTW	• Triangle-wave PWM mode 1/2/3 — Maximum of 10 channels in single-phase complementary — Duty cycle can be changed by changing one compare value each in positive-phase and negative-phase — Dead time is the difference between each positive-phase and negative-phase compare value — Synchronous operation can be used to synchronize 2 or more channels (Example: 3 channels for 3-phase complementary + 1 channel for single phase, 2 channels for 5-phase complementary)
		• Triangle-wave PWM mode 1/2/3 + Automatic dead time setting function — Maximum of 10 channels in single-phase complementary — Positive-phase duty cycle can be changed by changing positive-phase compare value — Negative-phase duty cycle can be changed by setting the left and right dead times — Synchronous operation can be used to synchronize 2 or more channels (Example: 3 channels for 3-phase complementary + 1 channel for single phase, 2 channels for 5-phase complementary)

The following describes the modes and output waveforms of the MTU and GPTW complementary PWM output (sawtooth waves).

Table 1.5 Complementary PWM Mode (sawtooth waves)

Output Waveform	Timer	Description
	MTU	• Reset-synchronized PWM mode — Maximum of 2 channels in 3-phase complementary
	GPTW	• Sawtooth-wave PWM mode + Synchronous operation — Maximum of 10 channels in single-phase complementary — The same compare value is set to the compare values set in each positive-phase and negative-phase — GTIOCnA pin output setting: Initial output is low, high output at GTCCRA register compare match, and low output at the end of the cycle GTIOCnB pin output setting: Initial output is high, low output at GTCCRB register compare match, and high output at the end of the cycle
	GPTW	• Sawtooth-wave one-shot pulse mode — Maximum of 10 channels in single-phase complementary — Set both compare values for both positive-phase and negative-phase using double buffers — Dead time is the difference between each positive-phase and negative-phase compare value — Synchronous operation can be used to synchronize 2 or more channels (Example: 3 channels for 3-phase complementary + 1 channel for single phase, 2 channels for 5-phase complementary)
	GPTW	• Sawtooth-wave one-shot pulse mode + Automatic dead time setting function — Maximum of 10 channels in single-phase complementary — Negative-phase duty cycle can be changed by setting the left and right dead times — Synchronous operation can be used to synchronize 2 or more channels (Example: 3 channels for 3-phase complementary + 1 channel for single phase, 2 channels for 5-phase complementary)

1.2 Buffer Functions

The MTU and GPTW are equipped with buffer functions. Buffer structure and transfer timing differ according to mode.

The GPTW supports a function that disables buffer transfer at a buffer register write. For details, refer to RX66T Group User's Manual: Hardware, section 24.8.2 Disabling of Buffer Operation.

Buffer structure and transfer timing for the MTU in PWM modes 1/2 and the reset-synchronized PWM mode are as follows. Transfer timing indicates the timing from buffer register to register.

For details, refer to RX66T Group User's Manual: Hardware, section 22.3.3 Buffer Operation.

Table 1.6 MTU PWM Modes 1/2

Channel	Register	Buffer Register	Transfer Timing
MTU0, MTU9	TGRA	TGRC	In PWM modes 1 or 2 • Compare match • Counter clear
	TGRB	TGRD	
	TGRE	TGRF	
MTU3, MTU4, MTU6, MTU7	TGRA	TGRC	In PWM mode 1 • Compare match • Counter clear
	TGRB	TGRD	

Buffer structure and transfer timing for the MTU in reset-synchronized PWM mode are as follows. Transfer timing indicates the timing from buffer register to register.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.15 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode.

Table 1.7 MTU Reset-synchronized PWM Mode

Channel	Register	Buffer Register	Transfer Timing
MTU3, MTU6	TGRA	TGRC	• The end of the cycle (MTU3.TGRA, MTU6.TGRA compare match)
	TGRB	TGRD	

Buffer structure and transfer timing for the MTU in complementary PWM mode (single buffer) are as follows. Transfer timing for MTU3.TGRA, MTU6.TGRA, TCDRA and TCDRB registers is the timing of data transfer from buffer register to register. For all other registers, it is the timing of data transfer from temporary register to register.

For details, refer to RX66T Group User's Manual: Hardware, section 22.3.8 Complementary PWM Mode, (b) Register Operation.

Table 1.8 MTU Complementary PWM Mode 1/2/3

Channel	Register	Temporary Register	Buffer Register	Transfer Timing
MTU3	TGRA	-	TGRC	Complementary PWM mode 1 • Crest Complementary PWM mode 2 • Trough Complementary PWM mode 3 • Crest and trough
	TGRB	TEMP1A	TGRD	
MTU4	TGRA	TEMP2A	TGRC	
	TGRB	TEMP3A	TGRD	
MTU6	TGRA	-	TGRC	
	TGRB	TEMP4A	TGRD	
MTU7	TGRA	TEMP5A	TGRC	
	TGRB	TEMP6A	TGRD	
MTU	TCDRA	-	TCBRA	
	TCDRB	-	TCBRB	

Buffer structure and transfer timing for the MTU in complementary PWM mode (double buffer) are as follows. Transfer timing for the MTU3.TGRA, MTU6.TGRA, TCDRA and TCDRB registers is the timing of data transfer from buffer register to register. For all other registers, confirm the timing in the table below.

When setting buffer register A (MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD), also set buffer register B (MTU3.TGRE, MTU4.TGRE, MTU4.TGRF, MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) at the same time.

Buffer register A is the compare value at down-counting; buffer register B is the compare value at up-counting.

For details, refer to RX66T Group User's Manual: Hardware, section 22.3.8 Complementary PWM Mode, (s) Double Buffer Function in Complementary PWM Mode.

Table 1.9 MTU Complementary PWM Mode 3 (using double buffer)

Channel	Register	Temporary Register	Buffer Register	Double Buffer Register	Transfer Timing
MTU3	TGRB	TEMP1A	TGRD	-	When writing to MTU4.TGRD or MTU7.TGRD, the values of TGRC, TGRD, TGRE, and TGRF are transferred to the corresponding temporary registers. Transfer timing from TEMP1A, TEMP2A, TEMP3A, TEMP4A, TEMP5A, and TEMP6A to the corresponding register is at the crest. Transfer timing from TEMP1B, TEMP2B, TEMP3B, TEMP4B, TEMP5B, and TEMP6B to the corresponding register is at the trough.
		TEMP1B	-	TGRE	
MTU4	TGRA	TEMP2A	TGRC	-	
		TEMP2B	-	TGRE	
	TGRB	TEMP3A	TGRD	-	
		TEMP3B	-	TGRF	
MTU6	TGRB	TEMP4A	TGRD	-	
		TEMP4B	-	TGRE	
MTU7	TGRA	TEMP5A	TGRC	-	
		TEMP5B	-	TGRE	
	TGRB	TEMP6A	TGRD	-	
		TEMP6B	-	TGRF	
MTU3	TGRA	-	TGRC	-	Crest and trough
MTU6	TGRA	-	TGRC	-	
MTU	TCDRA	-	TCBRA	-	
	TCDRB	-	TCBRB	-	

Buffer structure and transfer timing for the GPTW in the sawtooth-wave PWM mode and triangle-wave PWM mode 1/2 are as follows.

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.2 Buffer Operation.

Table 1.10 GPTW Sawtooth-wave PWM Mode and Triangle-wave PWM Mode 1/2

Register	Buffer Register	Double Buffer Register	Transfer Timing
GTPR	GTPBR	GTPDBR	Sawtooth-wave PWM mode • Overflow/underflow • Counter clear — hardware — software Triangle-wave PWM mode 1/2 • Trough
GTCCRA	GTCCRC	GTCCRD	Sawtooth-wave PWM mode • Overflow/underflow • Counter clear — hardware — software • Forcible buffer transfer Triangle-wave PWM mode 1 • Trough • Forcible buffer transfer Triangle-wave PWM mode 2 • Crest and trough • Forcible buffer transfer
GTCCRB	GTCCRE	GTCCRF	
GTDVU[Note]	GTDBU	-	Triangle-wave PWM mode 1/2 • Counter clear — Clear GTCNT counter • Trough
GTDVD[Note]	GTDBD	-	

Note: Cannot be used in the Sawtooth-wave PWM mode.

Buffer structure and transfer timing for the GPTW in sawtooth-wave one-shot pulse mode are as follows.

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.3 PWM Output Operating Mode, (2) Sawtooth-Wave One-Shot Pulse Mode.

Table 1.11 GPTW Sawtooth-Wave One-Shot Pulse Mode

Register	Buffer Register	Temporary Register	Double Buffer Register	Transfer Timing
GTPR	GTPBR	-	GTPDBR	• Overflow/underflow • Counter clear — hardware — software
GTCCRA	GTCCRC	-	-	At the end of the cycle (overflow/underflow): transfer from GTCCRC to GTCCRA, from GTCCRE to GTCCRB, from GTCCRD to temporary register A, and from GTCCRF to temporary register B At the GTCCRA compare match, transfer from temporary register A to GTCCRA; at the GTCCRB compare match, transfer from temporary register B to GTCCRB
	-	Temporary register A	GTCCRD	
GTCCRB	GTCCRE	-	-	
	-	Temporary register B	GTCCRF	
GTDVU	GTDBU	-	-	• Counter clear — Clear GTCNT counter • Overflow/underflow
GTDVD	GTDBD	-	-	

Buffer structure and transfer timing for the GPTW in triangle-wave PWM mode 3 are as follows.

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.3 PWM Output Operating Mode, (5) Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough).

Table1.12 GPTW Triangle-Wave PWM Mode 3

Register	Buffer Register	Temporary Register	Double Buffer Register	Transfer Timing
GTPR	GTPBR	-	GTPDBR	• Trough
GTCCRA	GTCCRC	-	-	Trough: Transfer from GTCCRC to GTCCRA, from GTCCRE to GTCCRB, from GTCCRD to temporary register A, and from GTCCRF to temporary register B Crest: Transfer from temporary register A to GTCCRA, and from temporary register B to GTCCRB.
	-	Temporary register A	GTCCRD	
GTCCRB	GTCCRE	-	-	
	-	Temporary register B	GTCCRF	
GTDVU	GTDBU	-	-	• Counter clear — Clear GTCNT counter • Trough
GTDVD	GTDBD	-	-	

Double buffer structure and transfer timing for the MTU and GPTW are shown in the figure below.

The example shows cases in which the next compare value is set in the up-counting period dedicated buffer (up dedicated buffer) or the down-counting period dedicated buffer (down dedicated buffer) when a crest interrupt is generated. Although the structures differ, in both cases, laterally asymmetric complementary PWM is realized in one interrupt processing per one carrier cycle.

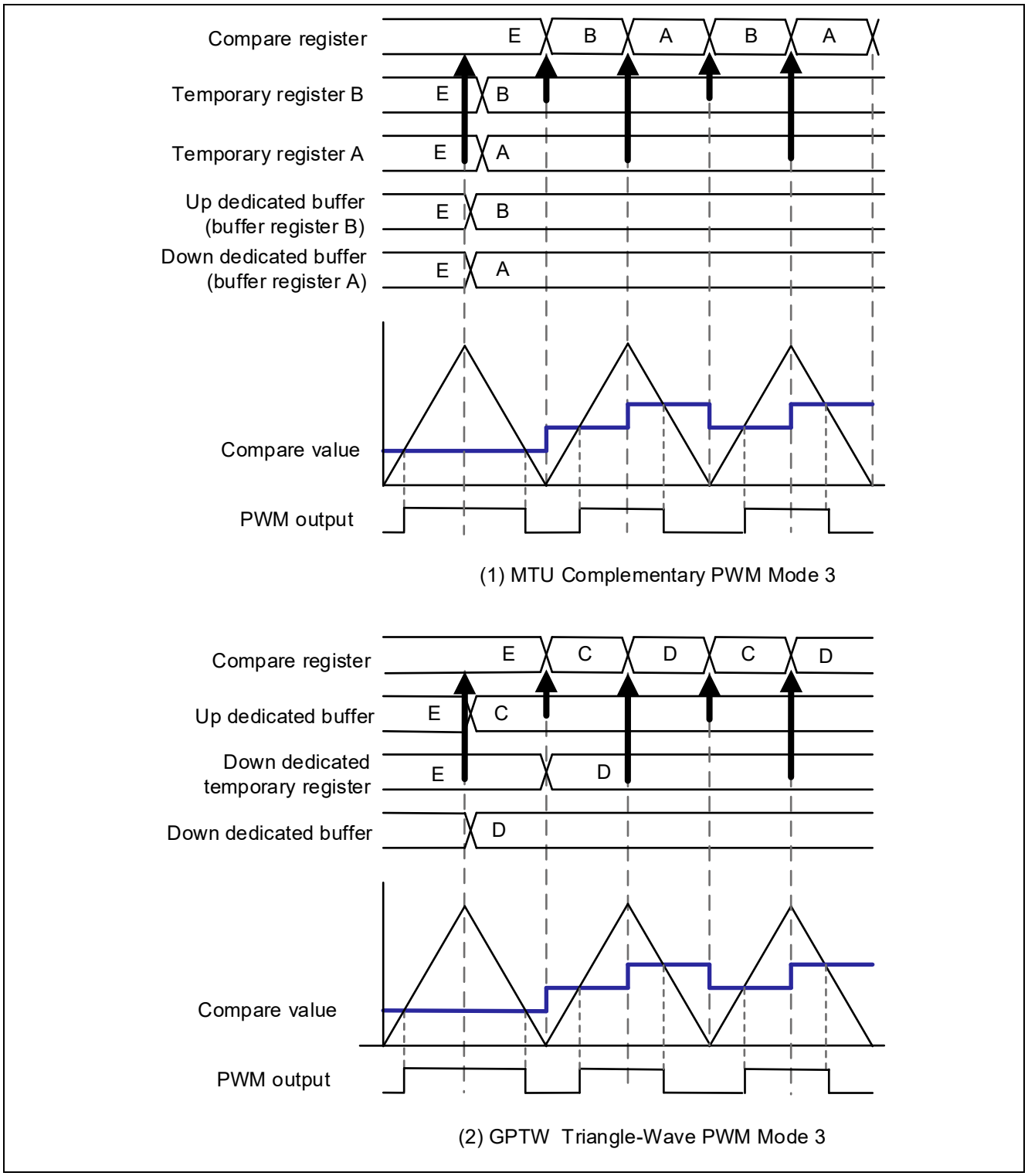
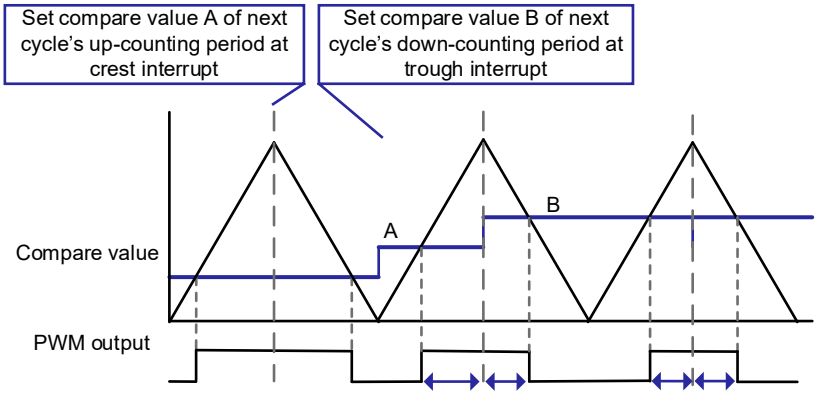


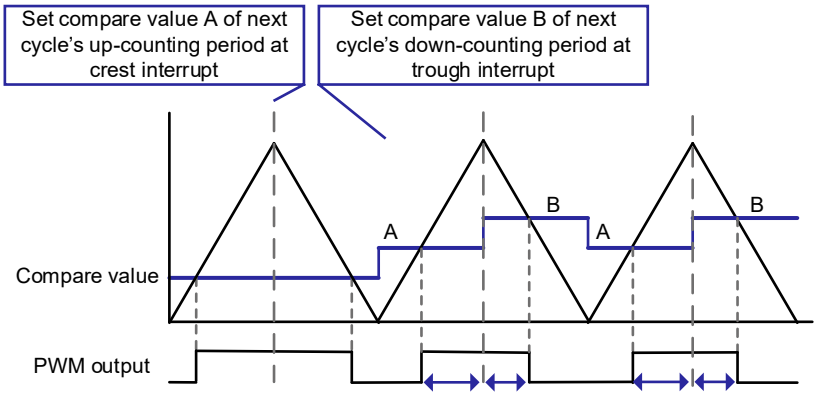
Figure 1.1 Double Buffer Structure

1.3 Laterally Asymmetric Complementary PWM Output

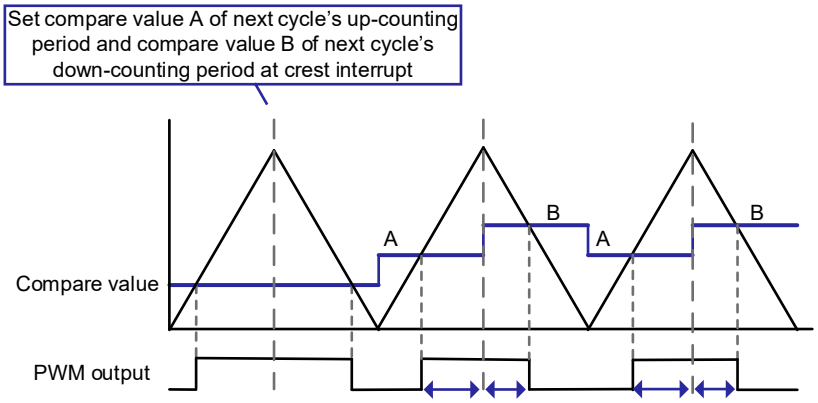
The MTU and GPTW are capable of laterally asymmetric complementary PWM output. To realize laterally asymmetric complementary PWM output, set the compare register at the timing indicated in the figure below.



(1) MTU Complementary PWM Mode 3 (single buffer)



(2) GPTW Triangle-Wave PWM Mode 2 (single buffer)



(3) MTU Complementary PWM Mode 3 (double buffer)
GPTW Triangle-Wave PWM Mode 2 (double buffer)
GPTW Triangle-Wave PWM Mode 3
GPTW Sawtooth-Wave One-Shot Pulse Mode[Note]

Note: In sawtooth-wave one-shot pulse mode, set compare value at the end of the cycle (overflow/underflow).

Figure 1.2 Laterally Asymmetric Complementary PWM Output

1.4 Automatic Dead Time Setting Function

The automatic dead time settings for the MTU and GPTW are as shown below. For the MTU, the same value is set for up-counting and down-counting, but two different values can be set for the GPTW.

For details regarding the MTU, refer to RX66T Group User's Manual: Hardware, section 22.3.8 Complementary PWM Mode, (e) Dead Time Setting.

For details regarding the GPTW, refer to RX66T Group User's Manual: Hardware, section 24.3.4 Automatic Dead Time Setting Function.

Table1.13 Automatic Dead Time Setting Function

Item	MTU	GPTW
Usable modes	• Complementary PWM mode 1/2/3	• Sawtooth-wave one-shot pulse mode • Triangle-wave PWM mode 1/2/3
Setting register	One register shared for counting up and down	One up-counting dedicated register One down-counting dedicated register
Change setting during operation	Prohibited	Enabled
Output protection function	-	✓

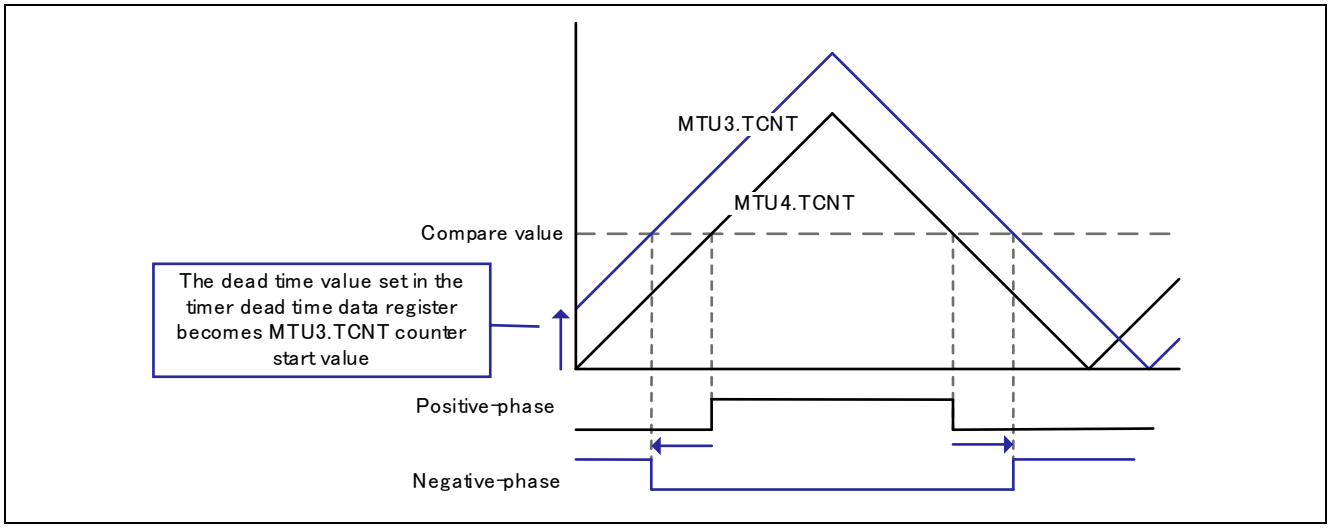


Figure 1.3 MTU Automatic Dead Time Setting (MTU3, MTU4)

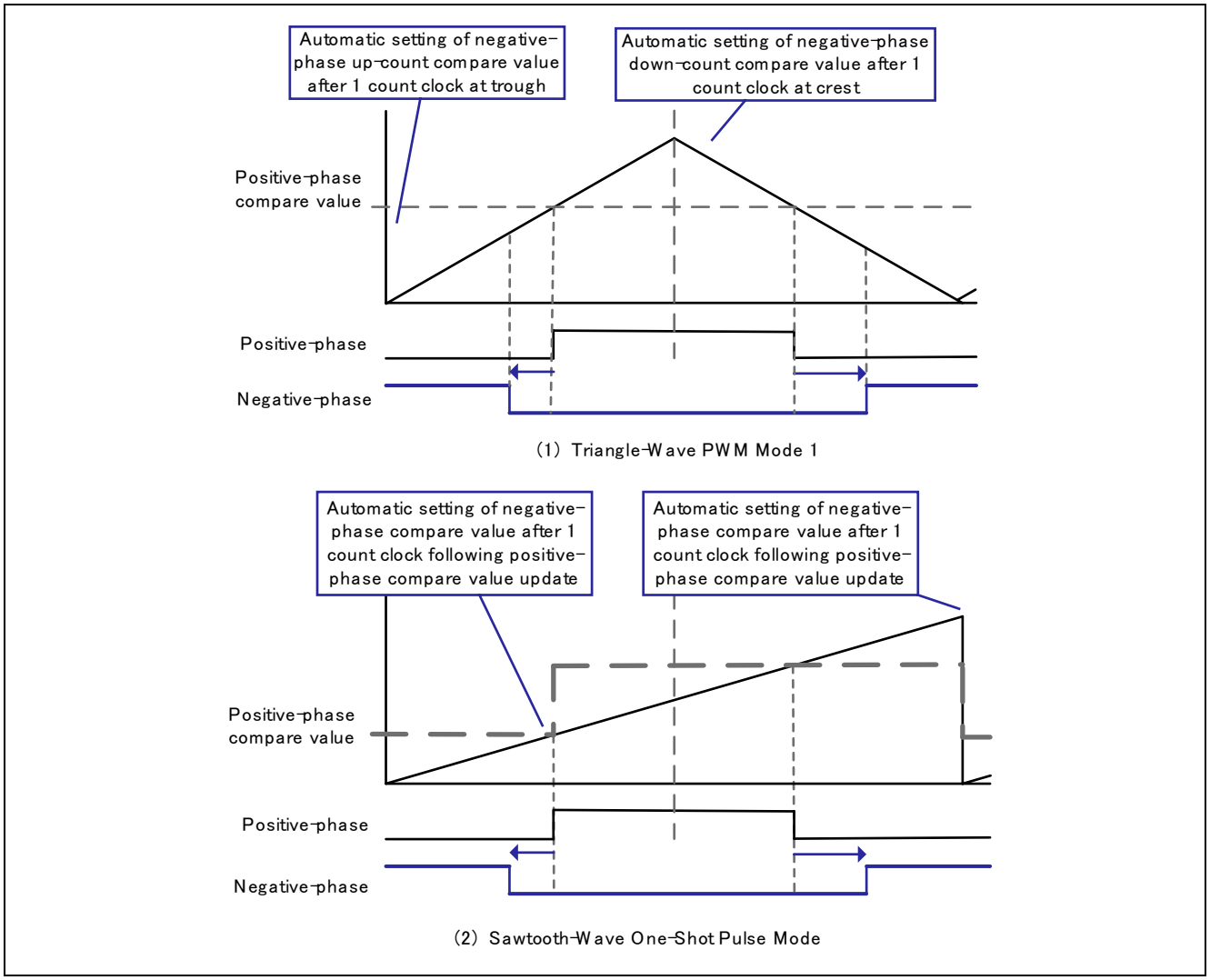


Figure 1.4 GPTW Automatic Dead Time Setting

1.5 Duty Cycle 0%/100% Output Function

The following describes setting methods for MTU and GPTW to duty cycles 0% and 100%.

Table1.14 Setting Methods for Duty Cycles 0% and 100%

Setting Method	MTU	GPTW
Compare register setting MTU: TGR register GPTW: GTCCR register	• PWM mode 1/2 100%: compare value = period register value 0%: compare value > period register value • Complementary PWM mode 1/2/3 100%: compare value = 0000h 0%: compare value = period register value The above settings may not be sufficient in some cases. If necessary, use in combination with TIOR.	Depending on the mode, the GTCCR register can be used with GTIOR to realize duty cycle 0%/100% output. For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F).
Bit setting of pin output duty setting GPTW: settings of GTUDDTYC.OADTY[1:0] and OBDTY[1:0]	-	For details, refer to RX66T Group User's Manual: Hardware, section 24.3.6 Duty Cycle 0%/100% Output Function.
I/O control register setting MTU: TIOR register GPTW: GTIOR register	For details, refer to RX66T Group User's Manual: Hardware, section 22.2.6 Timer I/O Control Register (TIOR).	For details, refer to RX66T Group User's Manual: Hardware, section 24.2.14 General PWM Timer I/O Control Register (GTIOR).

1.6 Relationship between Period and Period Register

The following describes the relationship between the period and period register in the MTU and GPTW.

Table1.15 Period and Period Register Relationship

Relational Expression	MTU	GPTW
Period = period register value + 1	• PWM mode 1/2 • Reset-synchronized PWM mode	• Sawtooth-wave PWM mode • Sawtooth-wave one-shot pulse
Period = period register value x 2	• Complementary PWM mode 1/2/3	• Triangle-wave PWM mode 1/2/3

2. Operation Confirmation Conditions

The sample codes included in this application note have been confirmed under the following operating conditions.

Table 2.1 Operation Confirmation Environments

Item	Description
MCU	R5F566TEADFP (included in Renesas Starter Kit for RX66T)
Operating frequency	Main clock: 8MHz PLL: 160MHz (Main clock x 1/1 x 20) HOCO: Stopped LOCO: Stopped System clock (ICLK): 160MHz (PLL x 1/1) Peripheral module clock A (PCLKA): 80MHz (PLL x 1/2) Peripheral module clock B (PCLKB): 40MHz (PLL x 1/4) Peripheral module clock C (PCLKC): 160MHz (PLL x 1/1) Peripheral module clock D (PCLKD): 40MHz (PLL x 1/4) FlashIF clock (FCLK): 40MHz (PLL x 1/4)
Operating voltage	3.3V
Integrated development environment (IDE)	Renesas Electronics e² studio Version 2021-07
C compiler[Note]	Renesas Electronics C/C++ Compiler Package for RX Family V3.03.00
	Compiler option The integrated development environment default settings are used.
iodefine.h version	V1.00
Endian	Little endian
Operation mode	Single-chip mode
Processor mode	Supervisor mode
Sample code version	V1.00
Board	Renesas Starter Kit for RX66T (Product number: RTK50566T0CxxxxxBE)
Emulator	E2-Lite

Note: Import the same version of the toolchain (C compiler) as specified in the original project. If the same toolchain is not located in the import destination, the toolchain cannot be selected, and an error will occur. Check the toolchain selection status on the project settings screen.

Refer to FAQ 3000404 for setting methods.

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

3. MCU Sample Codes

3.1 Common

3.1.1 Sample Code List

This application note provides the following sample codes created with the Smart Configurator. If the duty cycle is not indicated, waveform output is repeated in duty cycles 20%→40%→60%→80%→20%→・・・.

Sample code can be downloaded from the Renesas Electronics website.

Table 3.1 MTU Sample Code List (1/2)

Name	Description	Ref.
PWM Mode 1 Compare Match r01an5995_rx66t_mtu3_pwm1_cmp.zip	Use PWM mode 1 Buffer transfer when TGRA compare match occurs, and PWM waveform output	3.2
PWM Mode 1 Count Clear r01an5995_rx66t_mtu3_pwm1_cntclear.zip	Use PWM mode 1 Buffer transfer when counter clear occurs, and PWM waveform output	3.3
PWM Mode 1 Without Buffer Register r01an5995_rx66t_mtu3_pwm1.zip	Use PWM mode 1 PWM waveform output without using buffer	3.4
PWM Mode 2 Compare Match r01an5995_rx66t_mtu3_pwm2_cmp.zip	Use PWM mode 2 Buffer transfer when TGRA compare match occurs、and 2-phase PWM waveform output	3.5
PWM Mode 2 Count Clear r01an5995_rx66t_mtu3_pwm2_cntclear.zip	Use PWM mode 2 Buffer transfer when counter clear occurs, and 2-phase PWM waveform output	3.6
PWM Mode 2 Without Buffer Register r01an5995_rx66t_mtu3_pwm2.zip	Use PWM mode 2 2-phase PWM waveform output without using buffer	3.7
Reset-Synchronized PWM Mode r01an5995_rx66t_mtu3_reset_sync_pwm.zip	Use reset-synchronized PWM mode 3-phase complementary PWM waveform output without dead time Each phase outputs 25%, 50%, and 75% fixed duty cycle	3.8
Complementary PWM Mode with Double Buffer r01an5995_rx66t_mtu3_complementary_pwm_dblbuf.zip	Use complementary PWM mode 3 (transfer at crest and trough) Use double buffer to generate laterally asymmetric duty cycle and output 3-phase complementary PWM waveforms with dead time Output duty cycles: 20%→40%→60%→80%→60%→…	3.9
Complementary PWM Mode Without Double Buffer r01an5995_rx66t_mtu3_complementary_pwm.zip	Use complementary PWM mode 3 (transfer at crest and trough) Generate symmetric duty cycle without using double buffer, and output 3-phase complementary PWM waveforms with dead time Output duty cycles: 20%→40%→60%→80%→60%→…	3.10
PWM Mode 1 Duty Cycles 0% to 100% (modify compare register at compare match) r01an5995_rx66t_mtu3_pwm1_50to100.zip	Use PWM mode 1 PWM waveform output including duty cycles 0% and 100% Compare register modify at compare match Output duty cycles: 50%→80%→100%→80%→50%→0%→…	3.11

Table 3.2 MTU Sample Code List (2/2)

Name	Description	Ref.
PWM Mode 1 Duty Cycles 0% to 100% (compare register modify at counter clear) r01an5995_rx66t_mtu3_pwm1_50to100_rwcc.zip	Use PWM mode 1 PWM waveform output including duty cycles 0% and 100% Compare register modify at counter clear Output duty cycles: 50%→80%→100%→80%→50%→0%→…	3.12
PWM Mode 1 Duty Cycles 0% and 100% r01an5995_rx66t_mtu3_pwm1_0to100.zip	Use PWM mode 1 PWM waveform output, alternating between duty cycles 0% and 100% Output duty cycles: 0%→100%→0%→100%→…	3.13
Complementary PWM Mode Duty 0% to 100% r01an5995_rx66t_mtu3_complementary_pwm_50to100.zip	Use complementary PWM mode 2 (transfer at trough) PWM waveform output including duty cycles 0% and 100% Output duty cycles: 50%→80%→100%→80%→50%→0%→…	3.14
Complementary PWM Mode Duty Cycles 0% and 100% r01an5995_rx66t_mtu3_complementary_pwm_0to100.zip	Use complementary PWM mode 2 (transfer at trough) PWM waveform output, alternating between duty cycles 0% and 100% Output duty cycles: 0%→100%→0%→100% →…	3.15

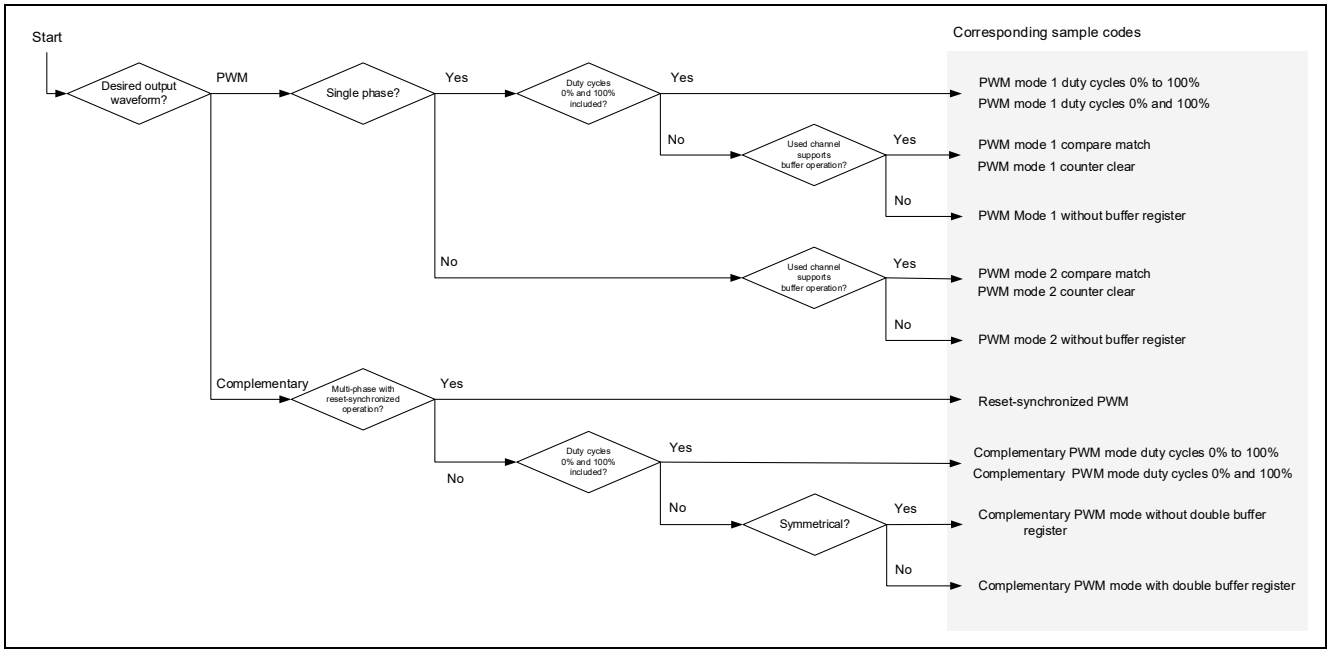


Figure 3.1 Sample Codes According to Purpose

3.1.2 Folder Structure

The main folder structure of a sample code is as follows.

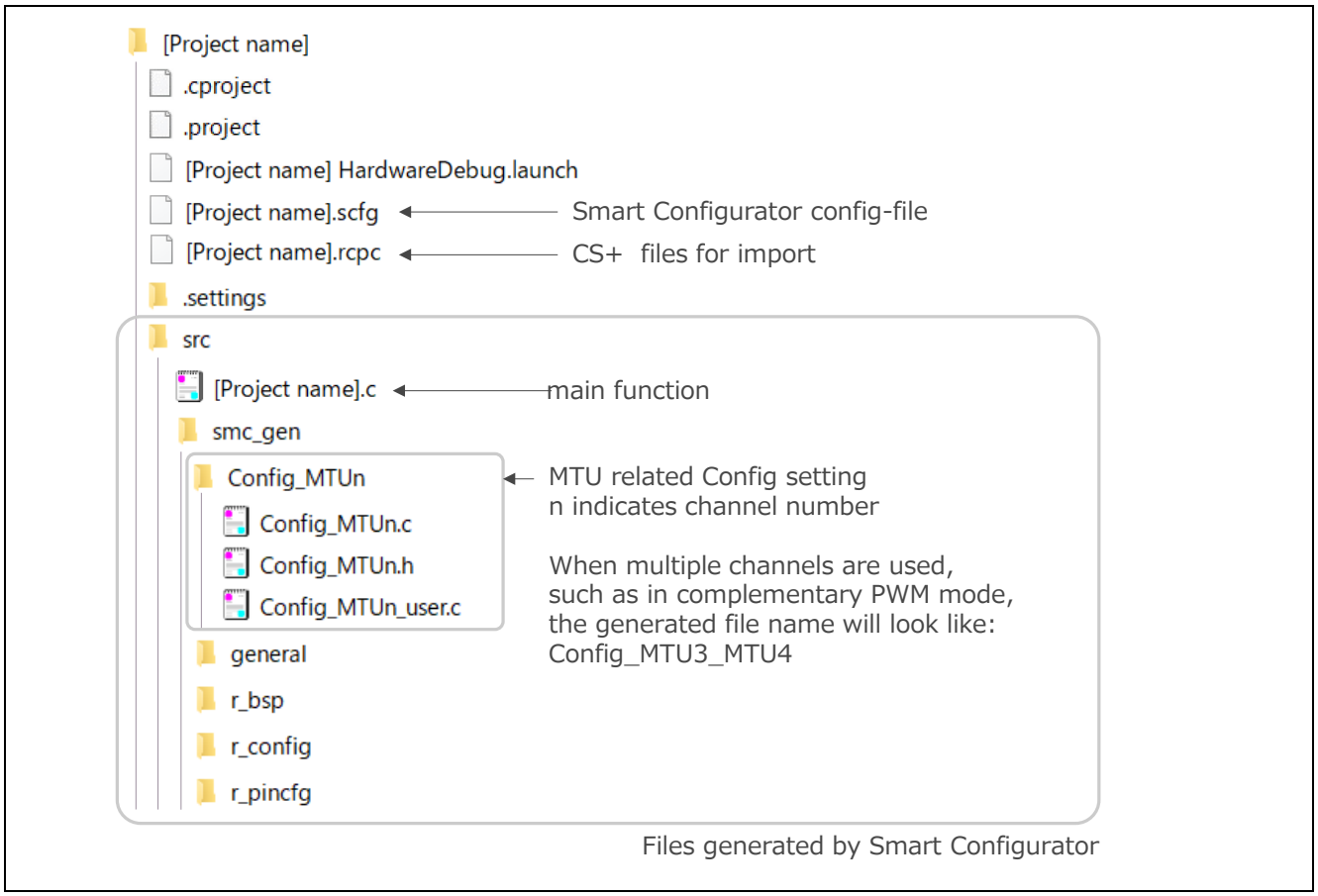


Figure 3.2 MTU Folder Structure

3.1.3 File Structure

The main file structure of a sample code is as follows.

Table 3.3 MTU File Structure

File Name	Description
[Project name].c	main function This is the main function. The Smart Configurator generates an empty function. The necessary processing for each sample code is described here.
Config_MTUn.c*	R_Config_MTUn_Create function This is the MTU's initialization function. The initialization function based on the settings in the Smart Configurator is generated by the Smart Configurator. The call for this function is generated by the Smart Configurator. This function is called in the R_SystemInit function executed before the main function.
	R_Config_MTUn_Start function This is the MTU's count start function. This function is generated by the Smart Configurator. In the sample codes, this function is called from the main function.
	R_Config_MTUn_Stop function This is the MTU's count stop function. This function is generated by the Smart Configurator. This function is not used in the sample codes.
Config_MTUn_user.c*	r_Config_MTUn_Create_UserInit function This is the MTU's user initialization function. The Smart Configurator generates an empty function. The necessary processing for each sample code is described here. This is the last function to be called in the R_Config_MTUn_Create function generated by the Smart Configurator.
	r_Config_MTUn_[interrupt name]_interrupt function This is the interrupt handler function. The Smart Configurator generates an empty function. The necessary processing for each sample code is described here.
Config_MTUn.h*	This is the header file that defines MTU related functions. This file is included in the r_smc_entry.h file generated by the Smart Configurator. To use MTU related functions, be sure to include the r_smc_entry.h file.

*: n indicates channel number

3.1.4 Adding Components

The sample codes use the Smart Configurator to add the MTU as described below.

Table 3.4 Adding Components

Item	Description
Component	Reference the section for each sample code ((1) in figure below)
Configuration Name	Sample codes use the default setting name
Operation	Reference the section for each sample code ((2) in figure below)
Resource	Reference the section for each sample code ((3) in figure below)

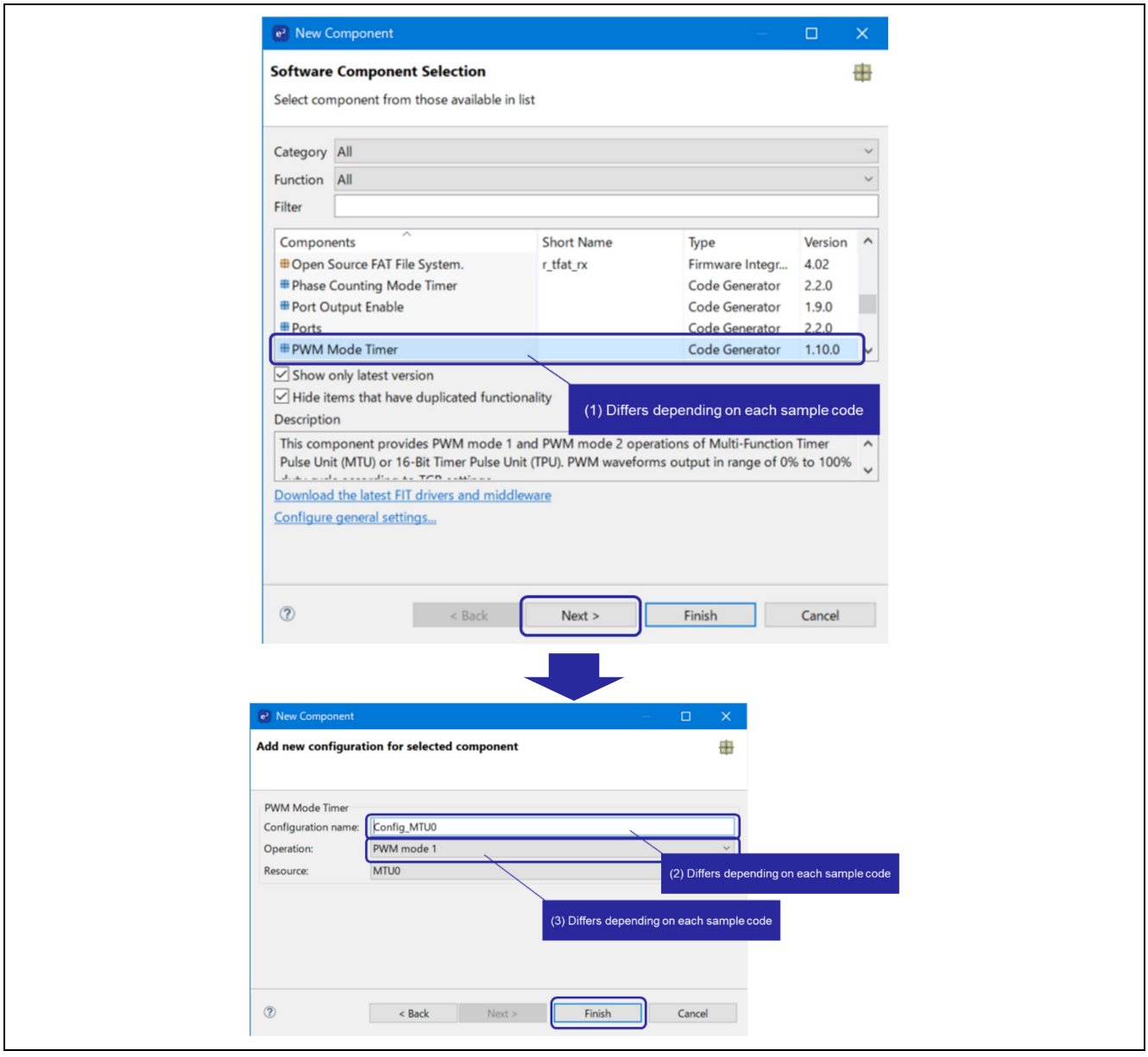


Figure 3.3 Adding Components

3.1.5 Pin Settings

Figure 3.4 shows an example of pin settings using the Smart Configurator.

Configure the pins after setting the MTU. For MTU settings, refer to “Smart Configurator Settings” for each sample code.

Pin settings are carried out in the R_Config_MTUn_Create function generated by the Smart Configurator.

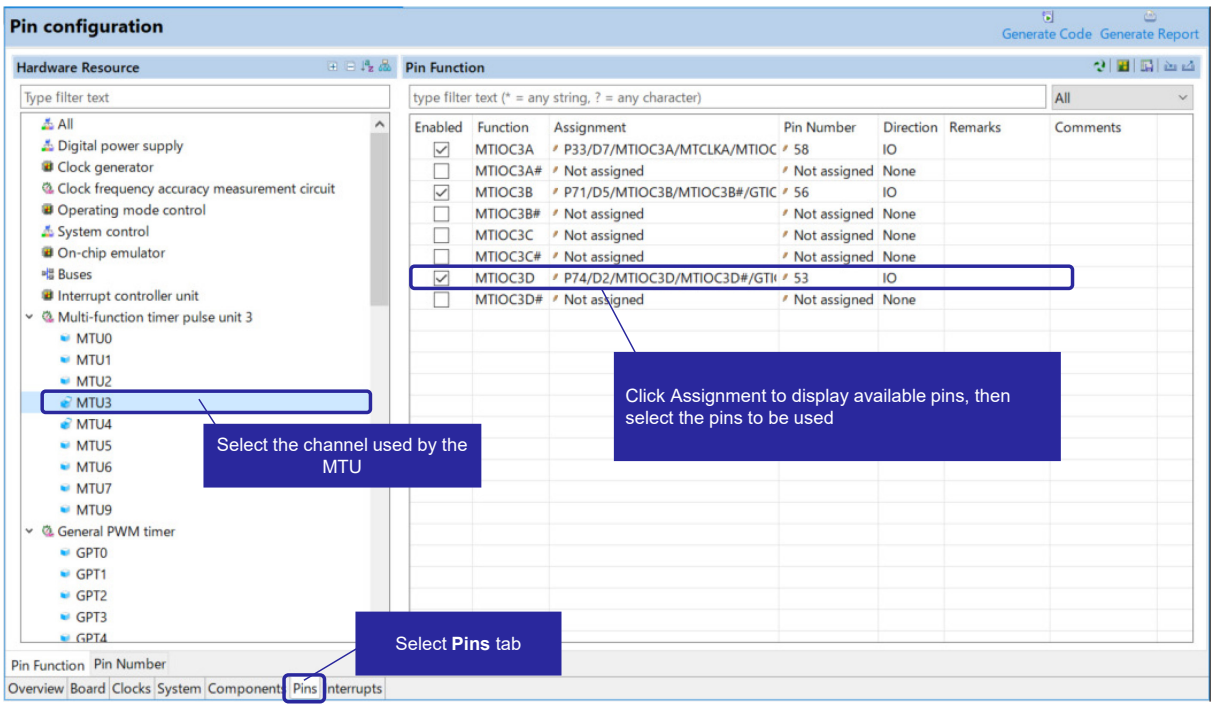


Figure 3.4 Pin Settings

3.1.6 Interrupt Settings

Figure 3.5 shows an example of interrupt settings using the Smart Configurator. For details on Software Configurable Interrupt A, refer to Renesas RX66T Group User's Manual Hardware, section 14.4.5.1 Software Configurable Interrupt A.

Configure interrupts after setting the MTU settings. For MTU settings, refer to "Smart Configurator Settings" for each sample code.

Interrupt settings can be configured in the R_Config_MTUn_Create function, R_Config_MTUn_Start function, and R_Config_MTUn_Stop function, all of which are generated by the Smart Configurator.

The interrupt handler function is created with the name r_Config_MTUn_[interrupt name]_interrupt in the Config_MTUn_user.c file generated by the Smart Configurator.

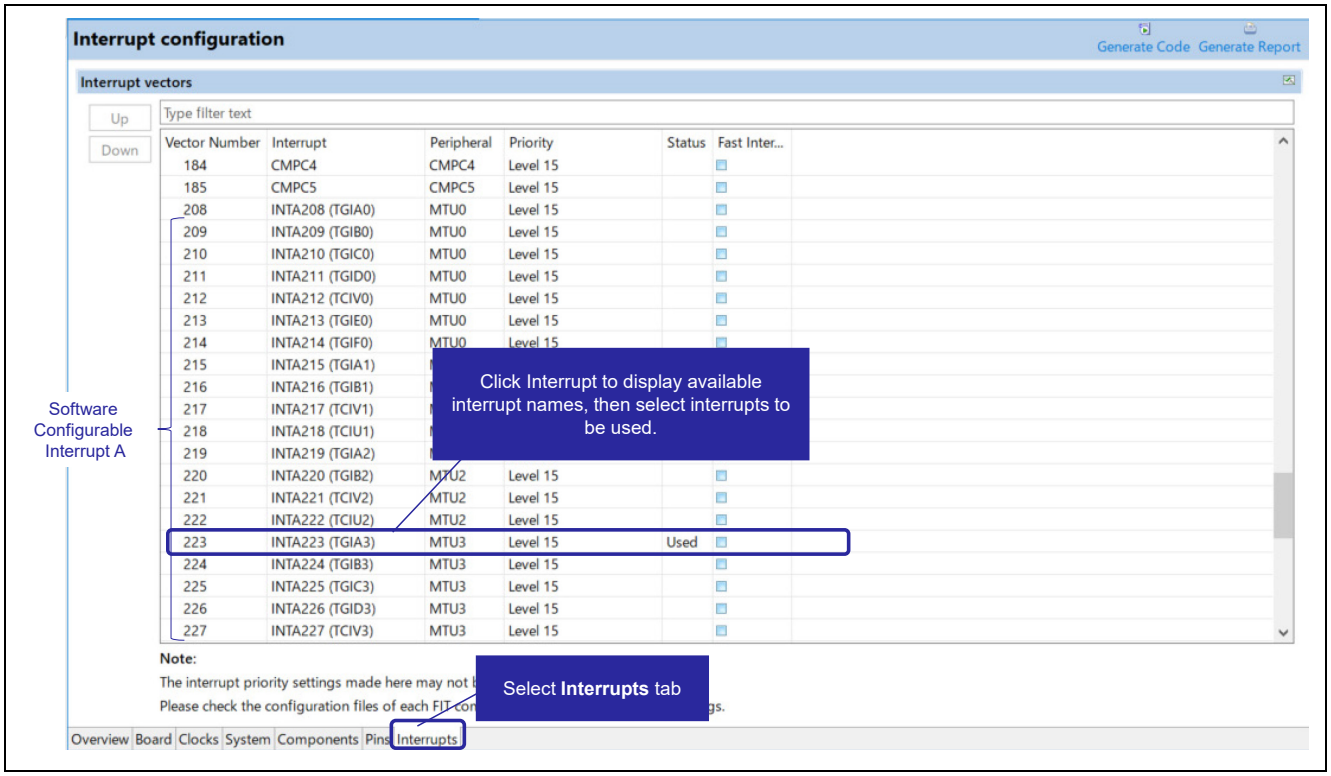


Figure 3.5 Interrupt Settings

3.2 PWM Mode 1 Compare Match

- Target sample code file name: r01an5995_rx66t_mtu3_pwm1_cmp.zip

3.2.1 Overview

The MTU PWM mode can be used to output PWM waveforms of duty cycles 0% to 100% according to the TGR register setting.

This sample code describes a sample code that uses PWM mode 1 and repeats waveform output of duty cycles 20% → 40% → 60% → 80% → 20% →···. The duty cycle is changed using buffer register TGRC to transfer the TGRC value to duty register TGRA when a TGRA compare match occurs.

The following list provides the MTU settings used in the sample code.

- Use PWM mode 1
- Use channel 0
- Initial output value = low
- Carrier period = 1ms
- Timer count clock = 40MHz (PCLKC/4)
- Use TGRB as period register
  - Timer counter clear source = TGRB compare match
  - Low output at TGRB compare match
- Use TGRA as duty register
  - High output at TGRA compare match
- Use buffer register
  - Use TGRC as buffer register of TGRA
  - Buffer transfer when TGRA compare match occurs
- Duty changes at each cycle
  - Change duty cycle at TGBR compare match interrupt
  - Refer to Figure 3.7 for details on the timing for duty cycle changes

Set in Smart Configurator.
For Setting Methods,
refer to section 3.2.3.

PWM mode 1 output for this sample code is shown below.

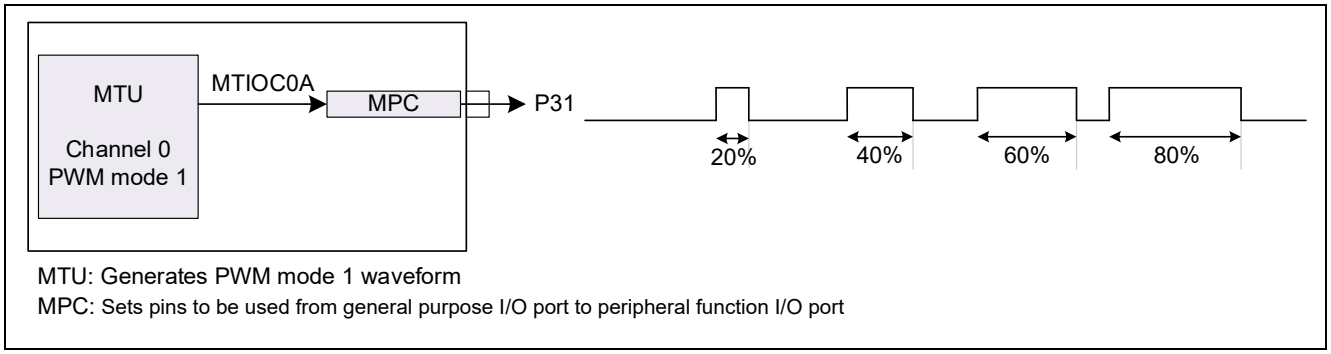
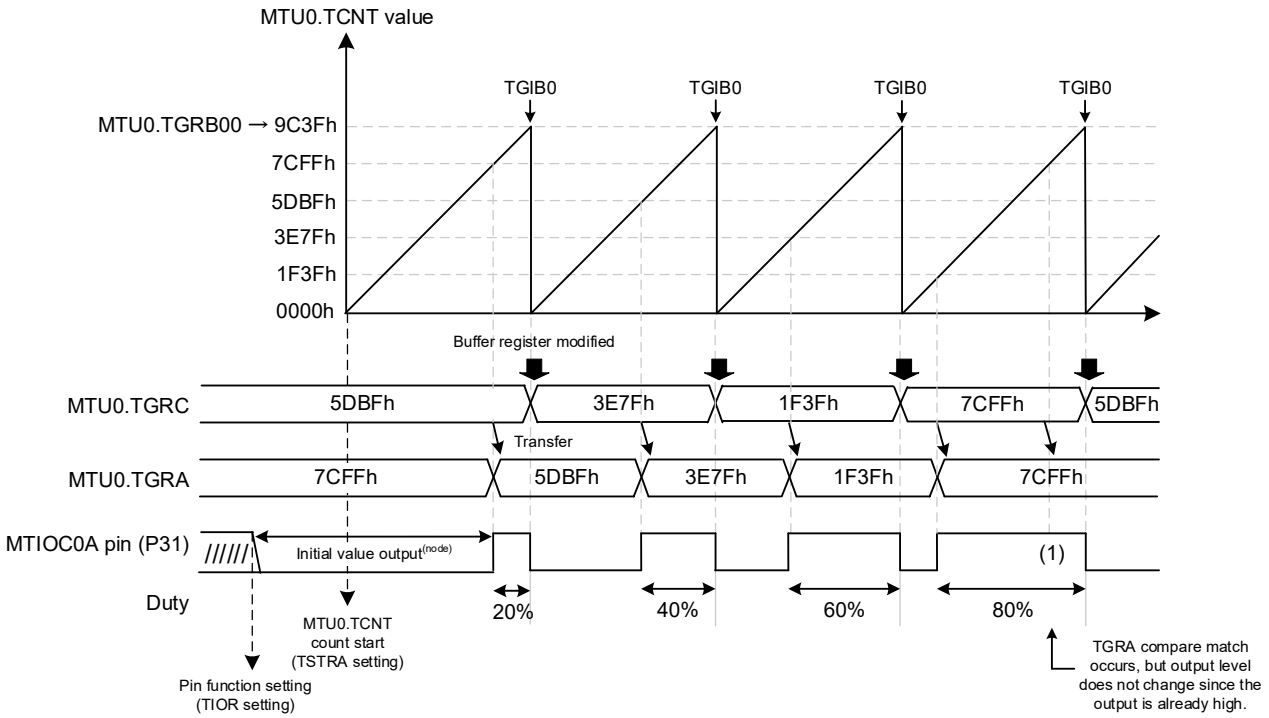


Figure 3.6 PWM Mode 1 Output

3.2.2 Operation Details

The sample code operations are shown in Figure 3.7. The settings of the duty cycle are changed with each period by modifying the value of buffer register TGRC at the TGRB compare match interrupt (TGIB0) of period register TGRB. The TGRC value is transferred to duty register TGRA when a TGRA compare match occurs.

When switching from duty cycle 80% to 20%, two TGRA compare matches occur in the same cycle, but the waveform does not change because the second compare match occurs during high output ((1) in figure below).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.7 Sample Code Operations

3.2.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table 3.5 Adding Components

Item	Description
Component	PWM Mode Timer
Configuration name	Config_MTU0
Operation	PWM Mode 1
Resource	MTU0

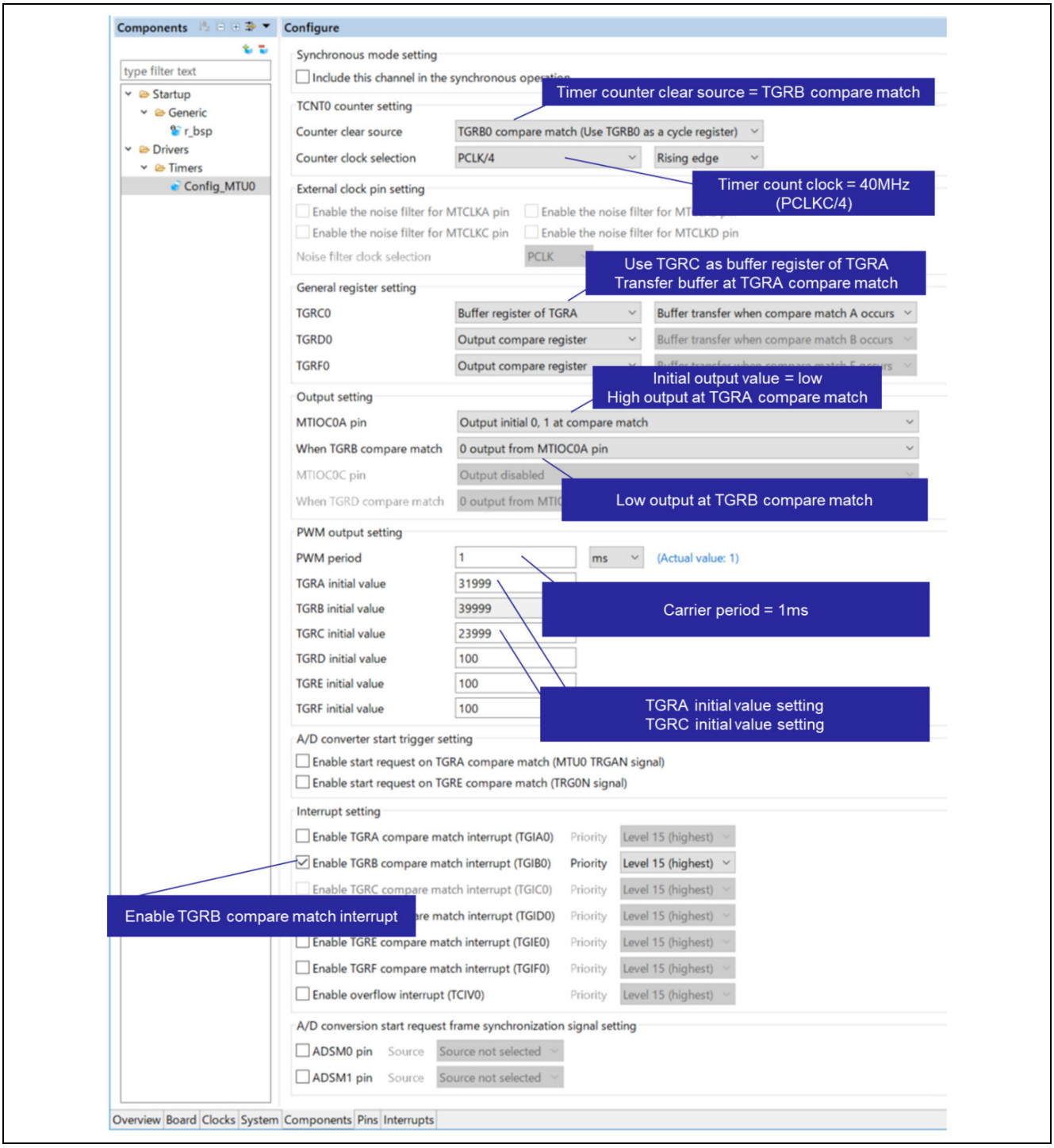


Figure 3.8 MTU0 Setting

3.2.4 Flowcharts

The following shows the main function processing and the processing when a TGIB0 interrupt occurs, both of which were added after code generation by the Smart Configurator.

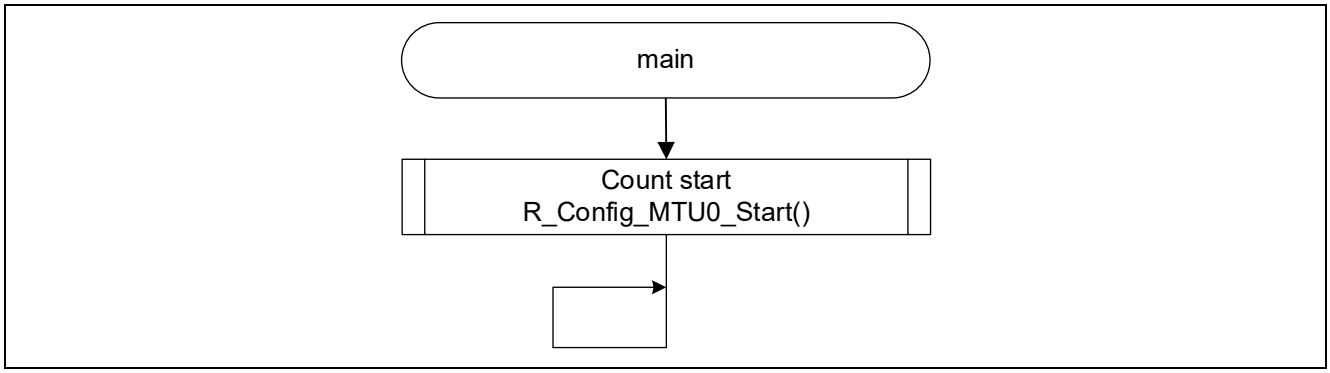


Figure 3.9 main Function

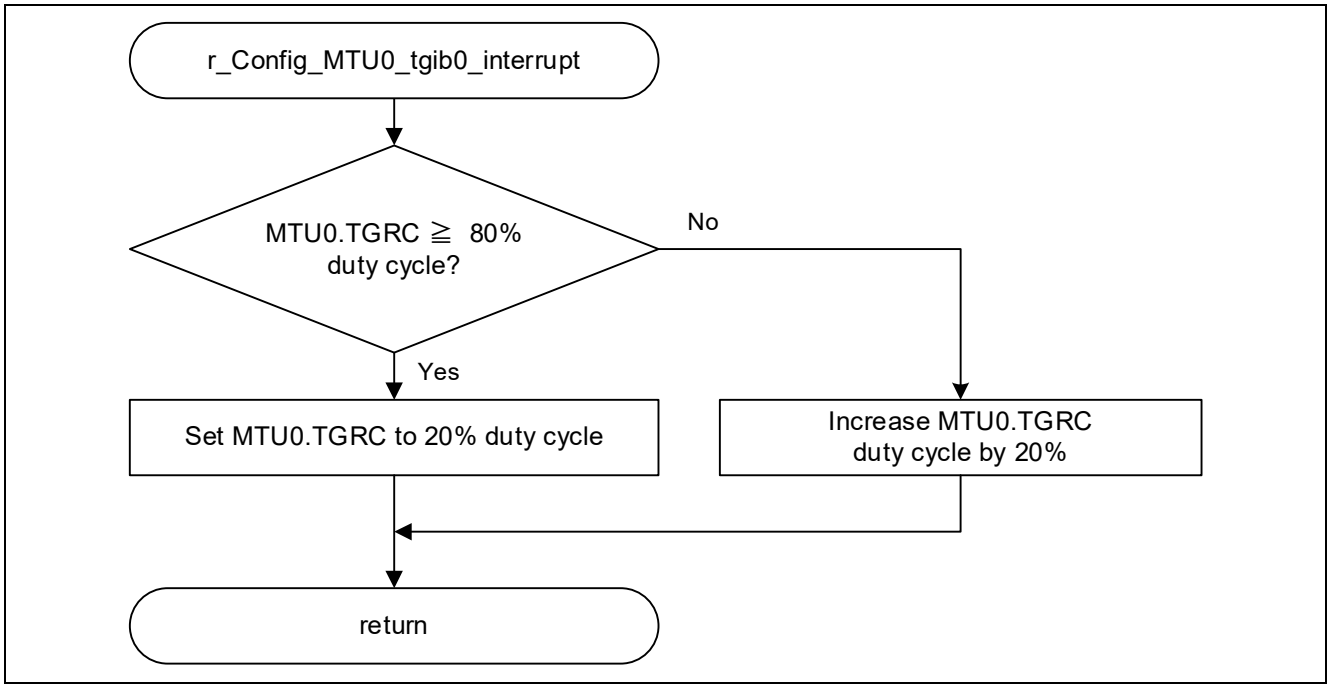


Figure 3.10 TGIB0 Interrupt Handler Function

3.2.5 Usage Notes

3.2.5.1 Contention between Buffer Register Modify Operation and Compare Match

When the buffer transfer timing is set at compare match, the data in the buffer register before the modification is transferred to the duty register if a compare match occurs in the buffer register write cycle.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.7 Contention between Buffer Register Write Operation and Compare Match.

3.2.5.2 Buffer Register Modify Delay

In this sample code, the transfer from buffer register TGRC to TGRA is executed when a TGRA compare match occurs. Modification of the TGRC value must be completed before the next compare match occurs.

If the buffer register modification is delayed, the TGRA value may not be updated by the time the TGRA compare match occurs in the next cycle. In such a case, consider transferring the buffer when counter clear occurs.

Refer to 3.3 PWM Mode 1 Count Clear for a sample code of PWM mode 1 that performs buffer transfer when counter clear occurs.

3.2.5.3 Setting a Value Greater than the Duty Register

In this sample code, the transfer from buffer register TGRC to TGRA is performed when a TGRA compare match occurs.
If a value greater than the value currently set in TGRA is set in TGRA after a TGRA compare match occurs, two TGRA compare matches may occur in one cycle.
For details, refer to (1) in Figure 3.7.

3.3 PWM Mode 1 Count Clear

- Target sample code file name: r01an5995_rx66t_mtu3_pwm1_cntclear.zip

3.3.1 Overview

The MTU PWM mode can be used to output PWM waveforms of duty cycles 0% to 100% according to the TGR register setting.

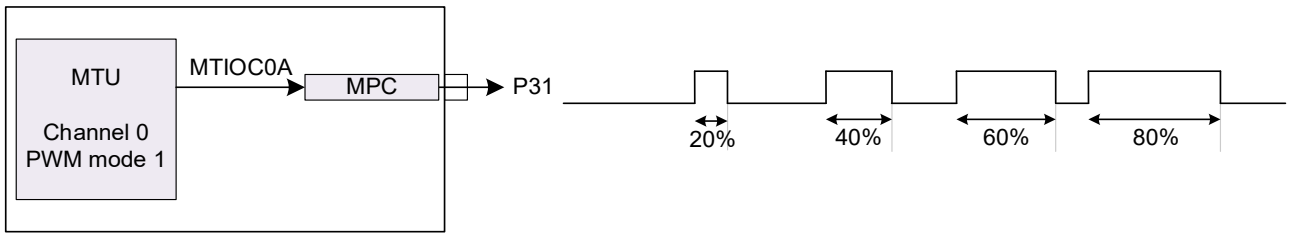
This sample code describes a sample code that uses PWM mode 1 and repeats waveform output of duty cycles 20% → 40% → 60% → 80% → 20% →···. The duty cycle is changed using buffer register TGRC to transfer the TGRC value to duty register TGRA when counter clear occurs.

The following list provides the MTU settings used in the sample code.

- Use PWM mode 1
- Use channel 0
- Initial output value = low
- Carrier period = 1ms
- Timer count clock = 40MHz (PCLKC/4)
- Use TGRB as period register
  - Timer counter clear source = TGRB compare match
  - Low output at TGRB compare match
- Use TGRA as duty register
  - High output at TGRA compare match
- Use buffer register
  - Use TGRC as buffer register of TGRA
  - Buffer transfer when counter clear occurs
- Duty changes at each cycle
  - Change duty cycle at TGBR compare match interrupt
  - Refer to Figure 3.12 for details on timing for duty cycle changes

Set in Smart Configurator.
For setting methods,
refer to section 3.3.3.

PWM mode 1 output for this sample code is shown below.

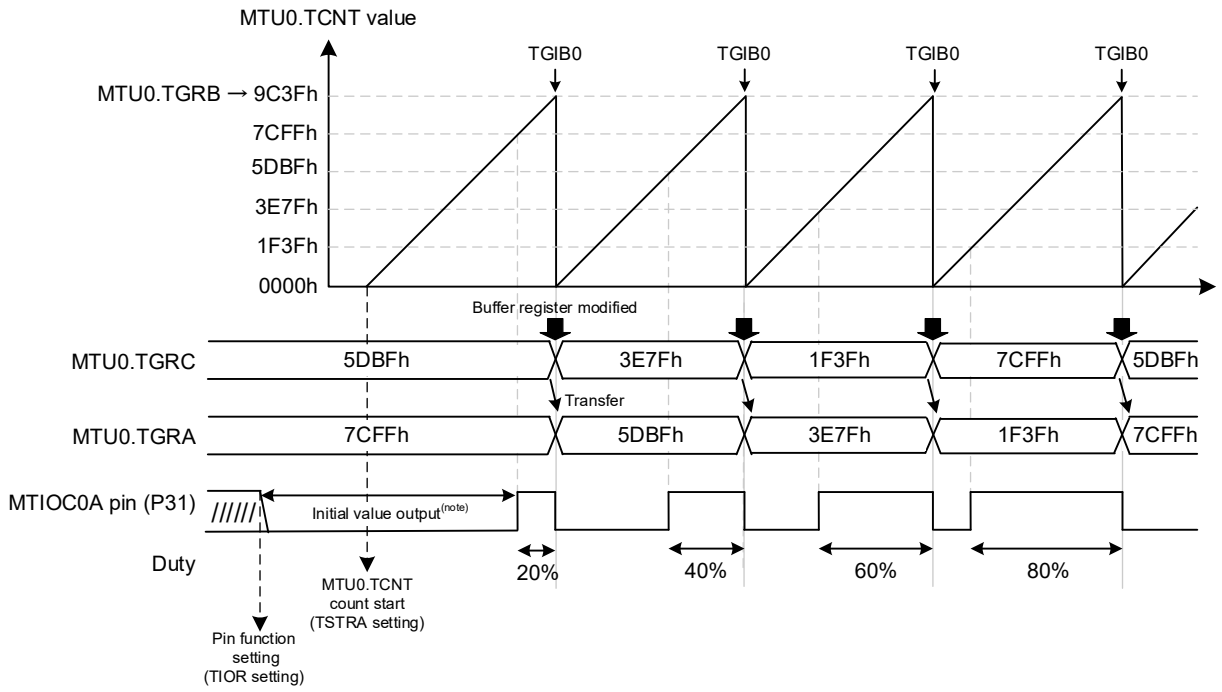


MTU: Generates PWM mode 1 waveform
MPC: Sets pins to be used from general purpose I/O port to peripheral function I/O port

Figure 3.11 PWM Mode 1 Output

3.3.2 Operation Details

The sample code operations are shown in Figure 3.12. The settings of the duty cycle are changed with each period by modifying the value of buffer register TGRC at the TGRB compare match interrupt (TGIB0) of period register TGRB. The TGRC value is transferred to duty register TGRA when a counter clear occurs.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.12 Sample Code Operations

3.3.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table 3.6 Adding Components

Item	Description
Component	PWM Mode Timer
Configuration name	Config_MTU0
Operation	PWM Mode 1
Resource	MTU0

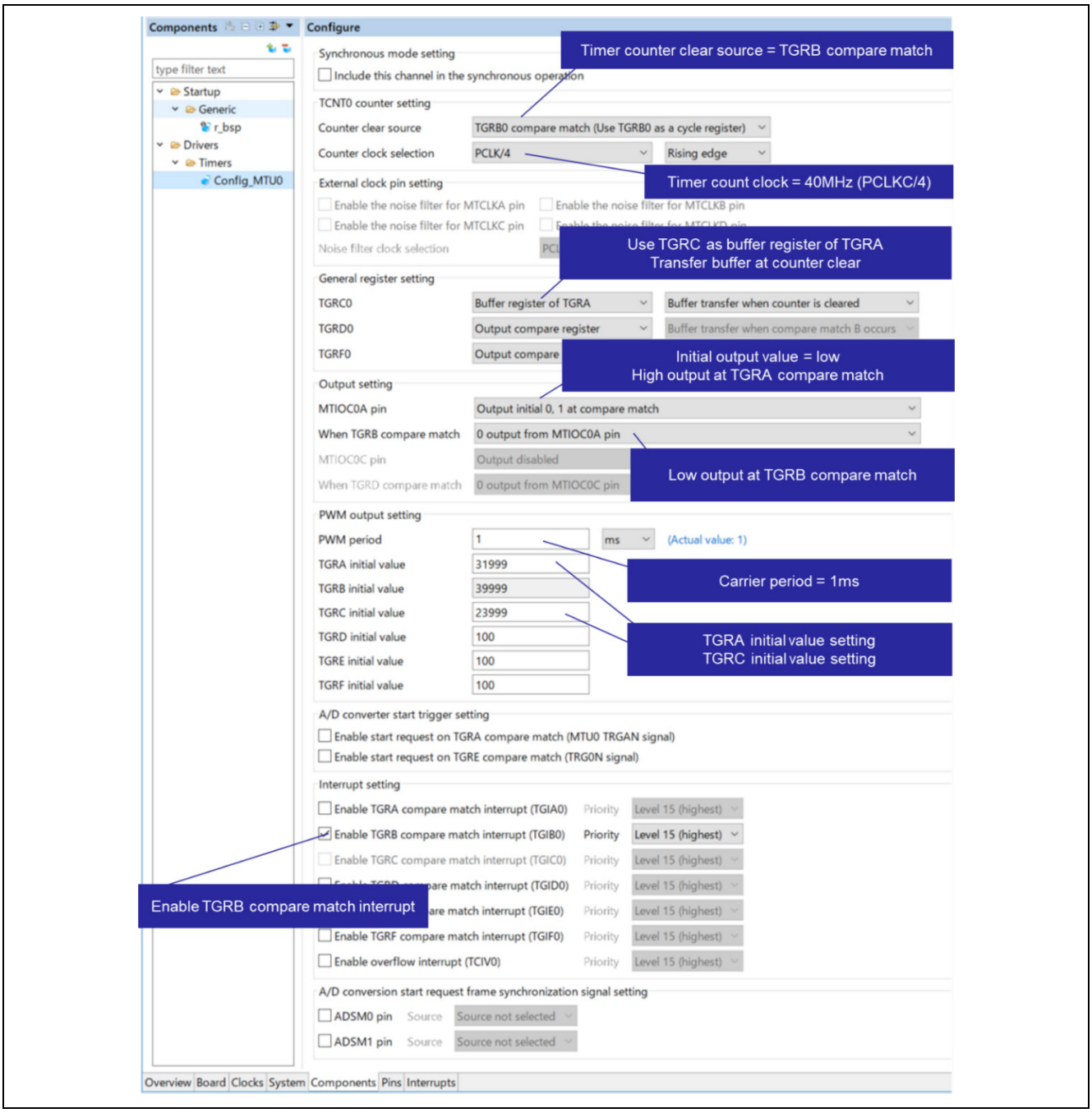


Figure 3.13 MTU0 Settings

3.3.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

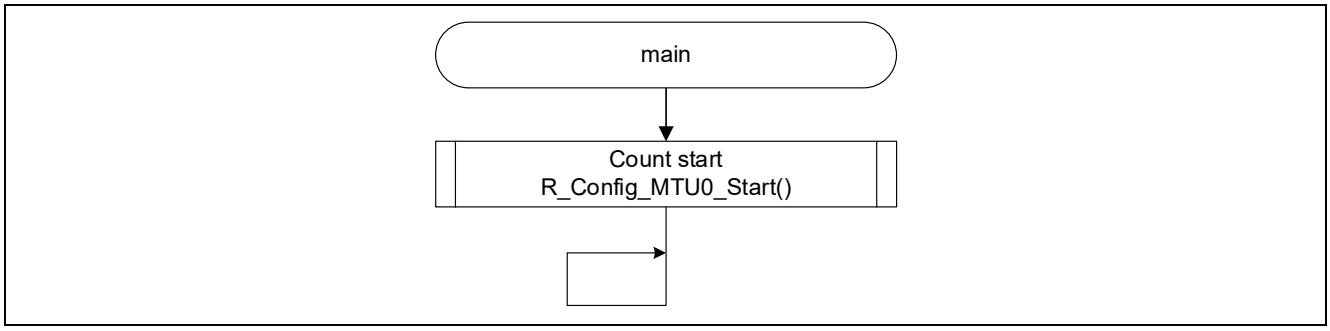


Figure 3.14 main Function

The TGIB0 interrupt handler function changes the value according to the current value of the TGRC register.

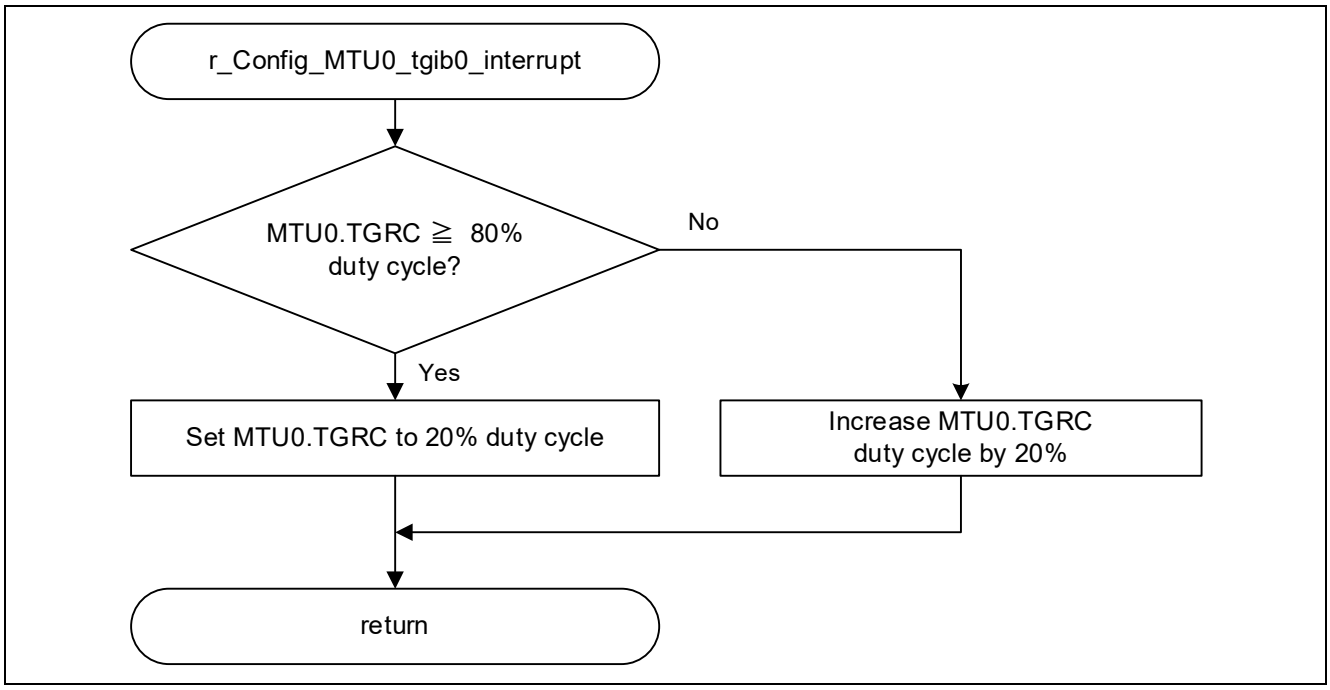


Figure 3.15 TGIB0 Interrupt Handler Function

3.3.5 Usage Notes

3.3.5.1 Contention between Buffer Register Modify Operation and Counter Clear

If the buffer transfer timing is set at counter clear, the data in the buffer register before the modification is transferred to the duty register if a counter clear occurs in the buffer register write cycle.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.8 Contention between Buffer Register Write and TCNT Clear Operations.

3.3.5.2 Buffer Register Modify Delay

In this sample code, the transfer from buffer register TGRC to TGRA is executed when a TGRA counter clear occurs. Modification of the TGRC value must be completed before the next counter clear occurs.

If the buffer register TGRC modification is delayed, the desired duty cycle cannot be output.

3.4 PWM Mode 1 Without Buffer Register

- Target sample code file name: r01an5995_rx66t_mtu3_pwm1.zip

3.4.1 Overview

The MTU PWM mode can be used to output PWM waveforms of duty cycles 0% to 100% according to the TGR register setting.

This sample code describes a sample code that uses PWM mode 1 and repeats waveform output of duty cycles 20% → 40% → 60% → 80% → 20% →···. The duty cycle is changed by updating the value of duty register TGRA without using the buffer register.

The following list provides the MTU settings used in the sample code.

- Use PWM mode 1
- Use channel 1
- Initial output value = low
- Carrier period = 1ms
- Timer count clock = 40MHz (PCLKC/4)
- Use TGRB as period register
  - Timer counter clear source = TGRB compare match
  - Low output at TGRB compare match
- Use TGRA as duty register
  - High output at TGRA compare match
- Duty changes at each cycle
  - Change duty cycle at TGRA compare match interrupt
  - Refer to Figure 3.17 for details on timing for duty cycle changes

Set in Smart Configurator.
For setting methods,
refer to section 3.4.3.

PWM mode 1 output for this sample code is shown below.

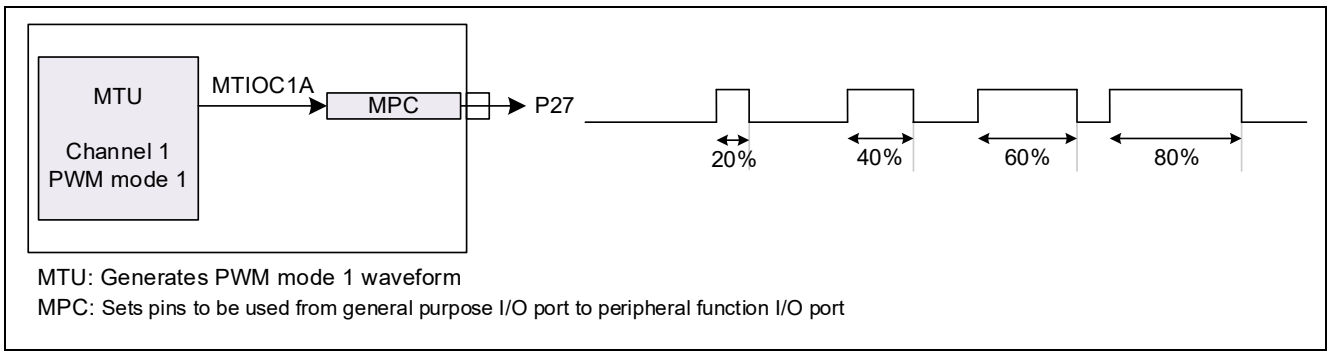
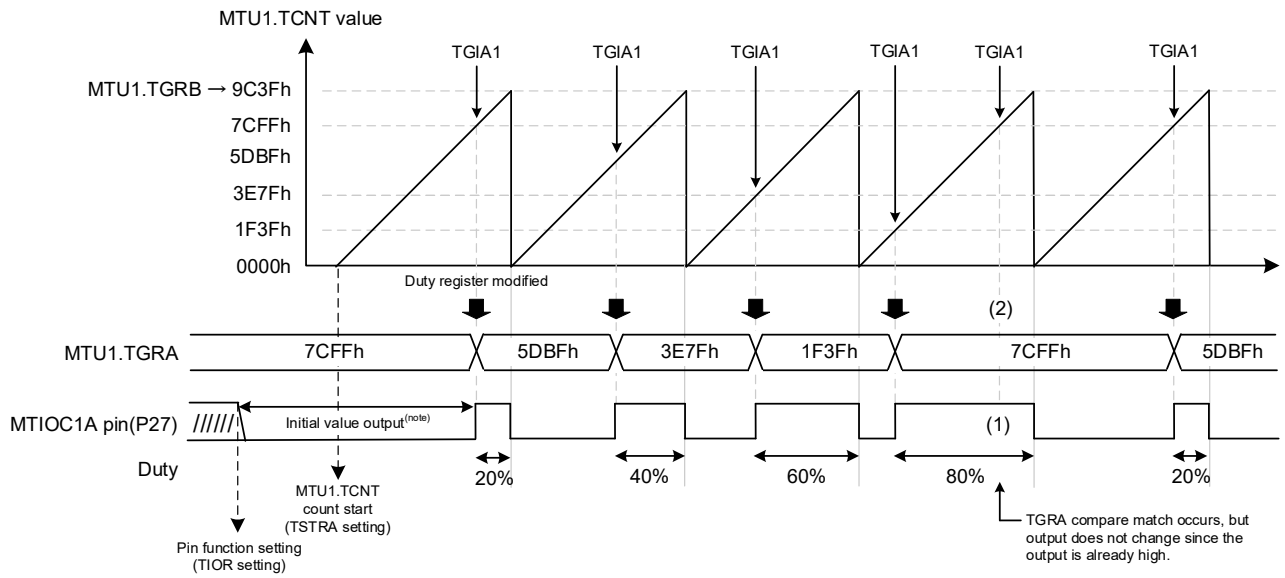


Figure 3.16 PWM Mode 1 Output

3.4.2 Operation Details

The sample code operations are shown in Figure 3.17. The settings of the duty cycle are changed with each period by modifying the TGRA value at the compare match interrupt (TGIA1) of duty register TGRA.

When switching from duty cycle 80% to 20%, two TGRA compare matches occur in the same cycle, but the waveform does not change because the second compare match occurs during high output ((1) in figure below). The duty cycle setting is changed with each period, so the timing for modifying the TGRA value needs to be adjusted ((2) in figure below).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.17 Sample Code Operations

3.4.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table 3.7 Adding Components

Item	Description
Component	PWM Mode Timer
Configuration name	Config_MTU1
Operation	PWM Mode 1
Resource	MTU1

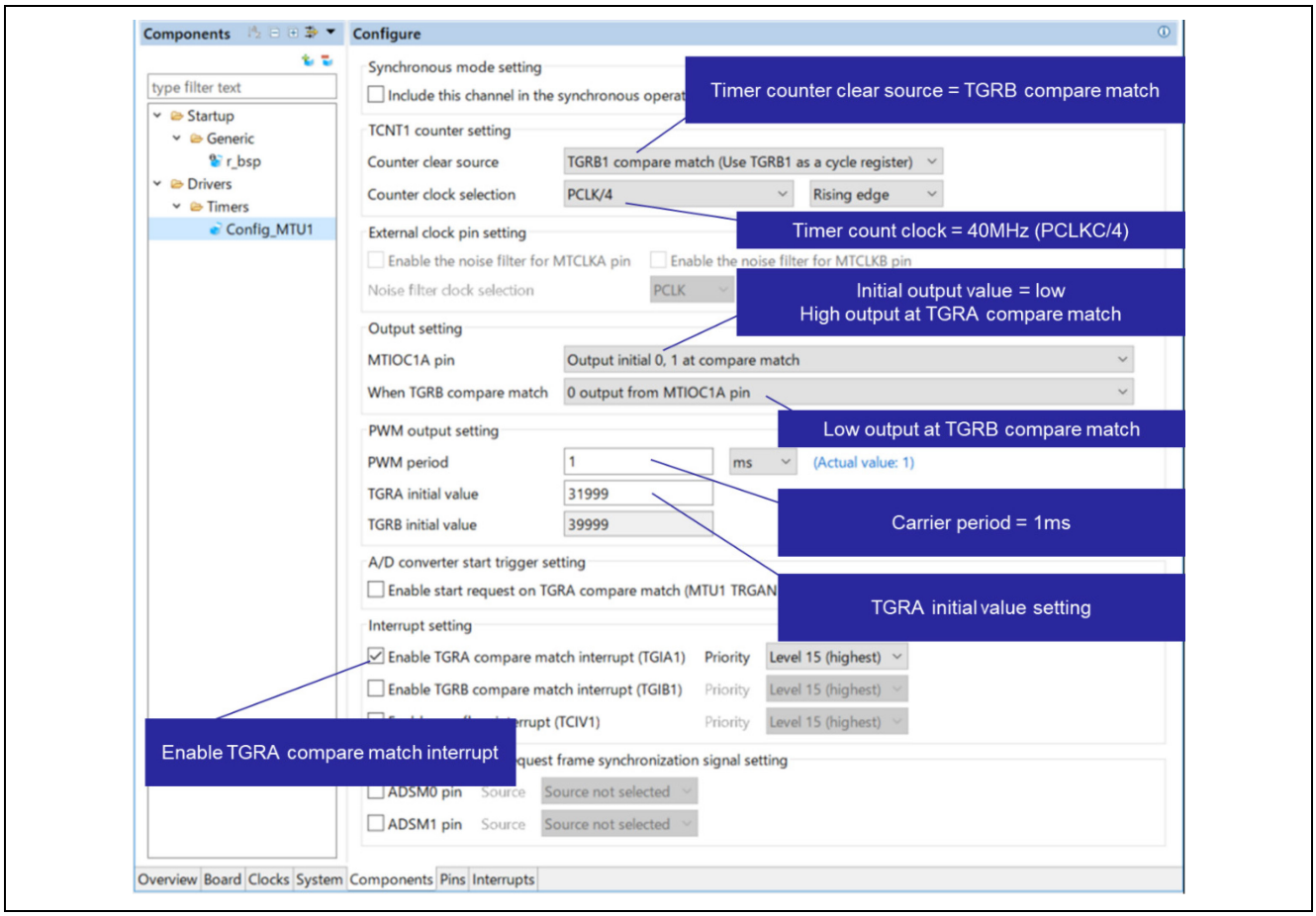


Figure 3.18 MTU1 Setting

3.4.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

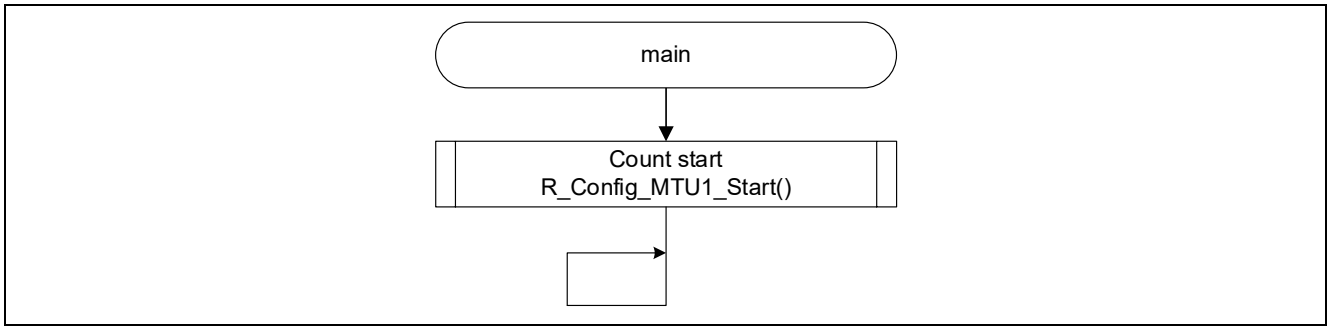


Figure 3.19 main Function

The user initialization function R_Config_MTU1_Create_UserInit, which is executed before the main function, initializes variables. This function is called from within the R_Config_MTU1_Create function.

This function initializes the following variable used in this sample code.

- s_duty_prv: variable for retaining the value of the previous TGRA register

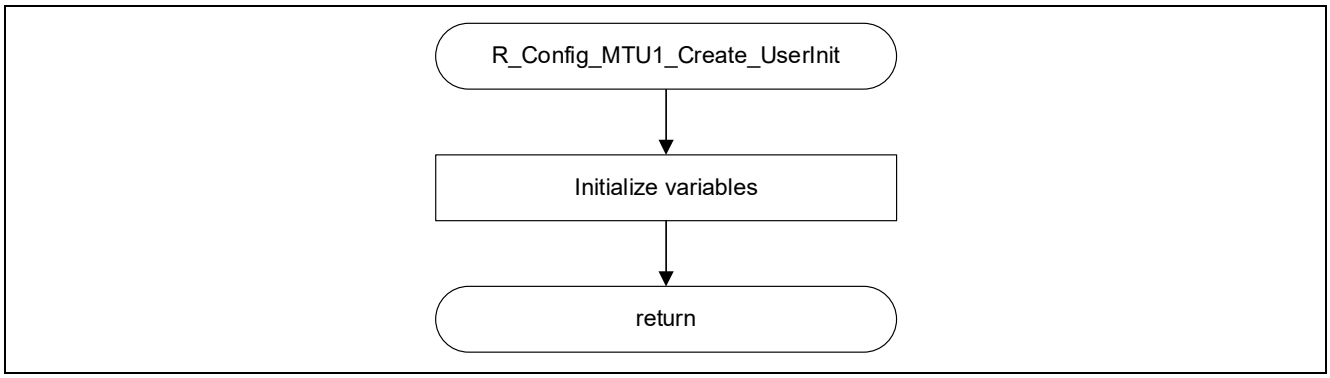


Figure 3.20 User Initialization Function

The TGIA1 interrupt handler function changes the value of the TGRA register according to the current value of the TGRA register and the value set in the previous TGRA register.

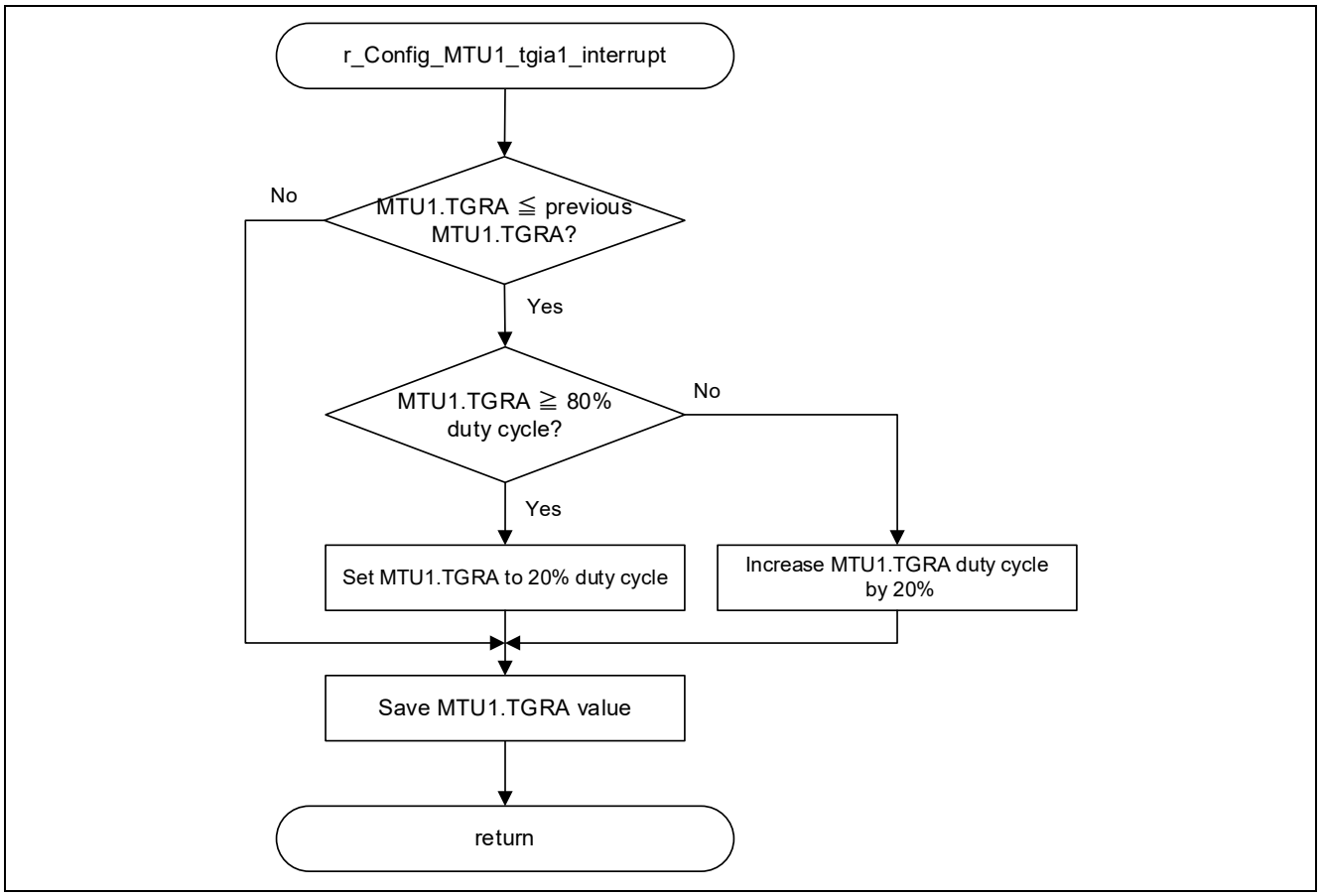


Figure 3.21 TGIA1 Interrupt Handler Function

3.4.5 Usage Notes

3.4.5.1 Contention between TGR Register Write and Compare Match

If a compare match occurs in the TGR register write cycle, a write to the TGR register write is performed and a compare match signal is also generated.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.6 Contention between TGR Write Operation and Compare Match.

3.4.5.2 Setting a Value Greater than the Duty Register

In this sample code, duty register TGRA is modified when the MTU1.TGRA compare match interrupt (TGIA1) is generated.

If a value greater than the value currently set in TGRA is set in TGRA after a TGRA compare match occurs, two TGRA compare matches may occur in one cycle.

For details, refer to (1) in Figure 3.17.

3.5 PWM Mode 2 Compare Match

- Target sample code file name: r01an5995_rx66t_mtu3_pwm2_cmp.zip

3.5.1 Overview

The MTU PWM mode can be used to output PWM waveforms of duty cycles 0% to 100% according to the TGR register setting.

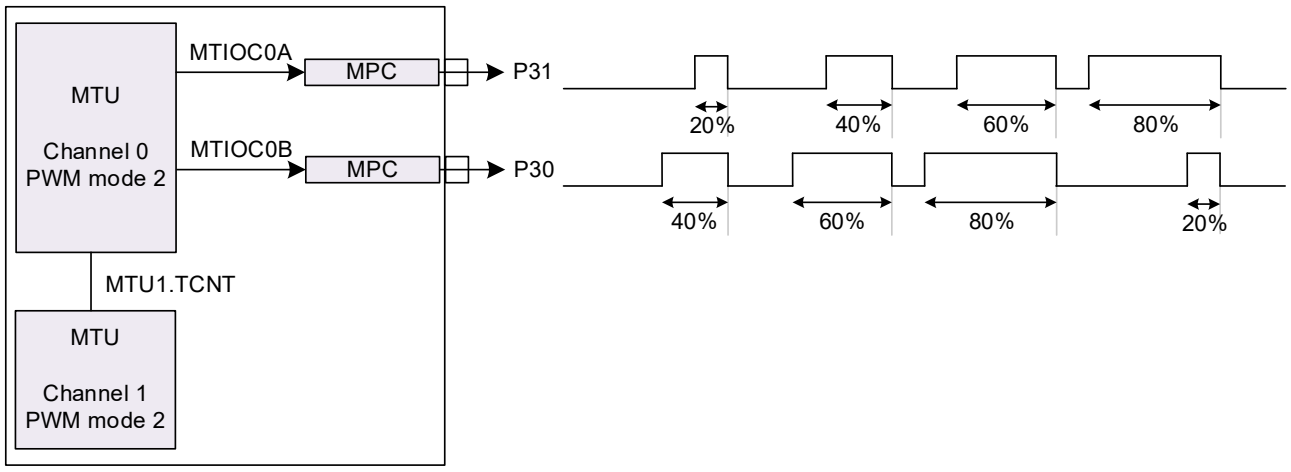
This sample code describes a sample code that uses PWM mode 2 and repeats waveform output of duty cycles 20% → 40% → 60% → 80% → 20% →⋯. The duty cycle is changed using buffer registers TGRC and TGRD to transfer the values of TGRC and TGRD to duty registers TGRA and TGRB when a compare match occurs between duty registers TGRA and TGRB. Since the TGRA to TGRD registers of MTU channel 0 are used for the duty register and buffer register, the period register uses the TGRA register of channel 1.

The following list provides the MTU settings used in the sample code.

- Use PWM mode 2
- Use channels 0 and 1
- Initial output value = low
- Carrier period = 1ms
- Timer count clock = 40MHz (PCLKC/4)
- Use MTU1.TGRA as period register
  - Timer counter clear source = MTU1.TGRA compare match
  - Low output at MTU1.TGRA compare match
- Use MTU0.TGRA as duty register
  - High output at MTU0.TGRA compare match
- Use MTU0.TGRB as duty register
  - High output at MTU0.TGRB compare match
- Use buffer register
  - Use MTU0.TGRC as buffer register of MTU0.TGRA
  - Use MTU0.TGRD as buffer register of MTU0.TGRB
  - Buffer transfer when compare match occurs
- Duty changes at each cycle
  - Change duty cycle at MTU1.TGRA compare match interrupt
  - Refer to Figure 3.23 for details on the timing for duty cycle changes

Set in Smart Configurator.
For Setting Methods,
refer to section 3.5.3.

PWM mode 2 output for this sample code is shown below.



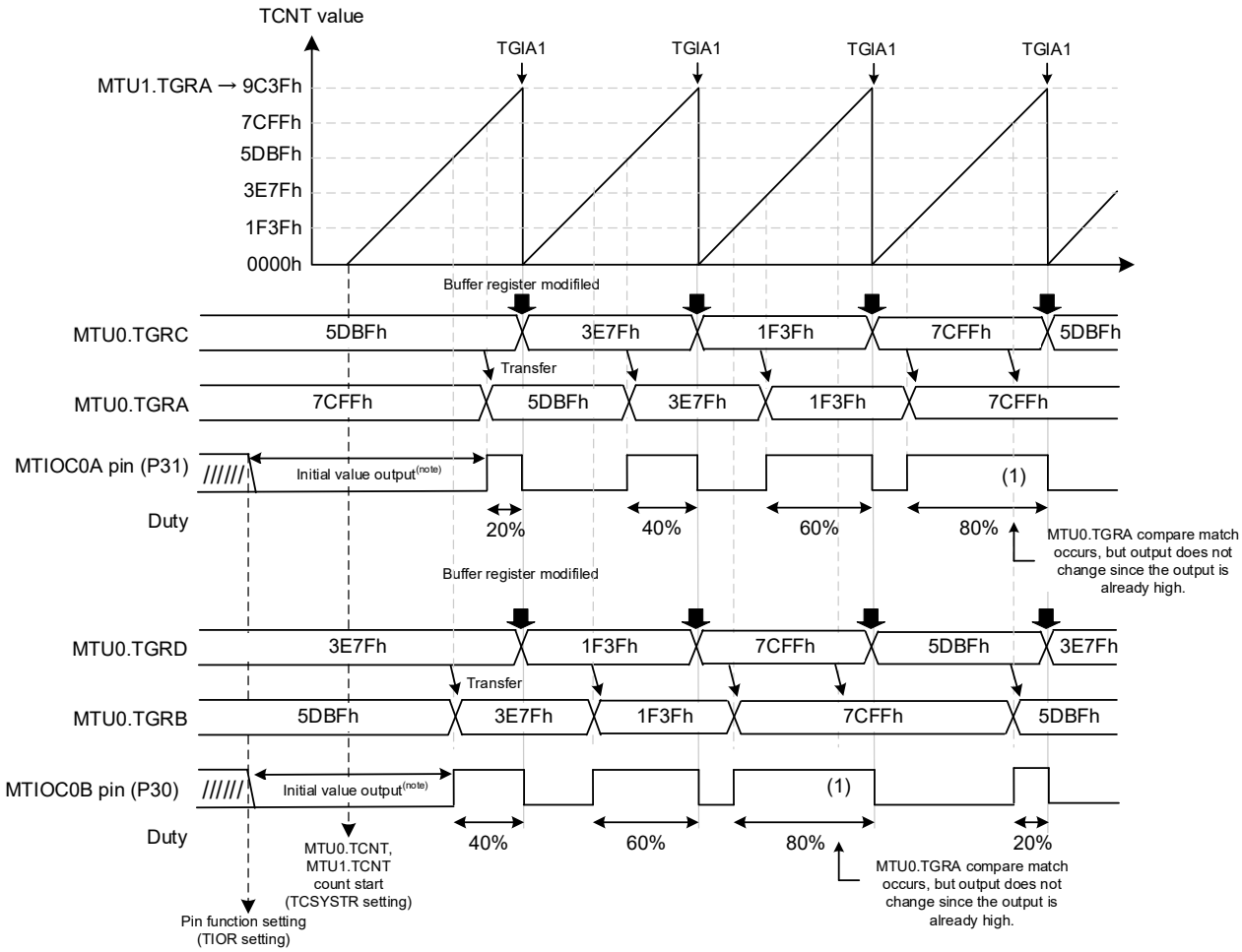
MTU: Generates PWM mode 2 waveform
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

Figure 3.22 PWM Mode 2 Output

3.5.2 Operation Details

The sample code operations are shown in Figure 3.23. The settings of the duty cycle are changed with each period by modifying the values of buffer registers MTU0.TGRC and MTU0.TGRD at the compare match interrupt (TGIA1) of period register MTU1.TGRA. The MTU0.TGRC and MTU0.TGRD values are transferred to duty registers MTU0.TGRA and MTU0.TGRB when a compare match occurs.

When switching from duty cycle 80% to 20%, two TGRA compare matches occur in the same cycle, but the waveform does not change because the second compare match occurs during high output ((1) in figure below).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.23 Sample Code Operations

3.5.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table 3.8 Adding Components

Item	Description	
Component	PWM Mode Timer	
Configuration name	Config_MTU0	Config_MTU1
Operation	PWM Mode 2	
Resource	MTU0	MTU1

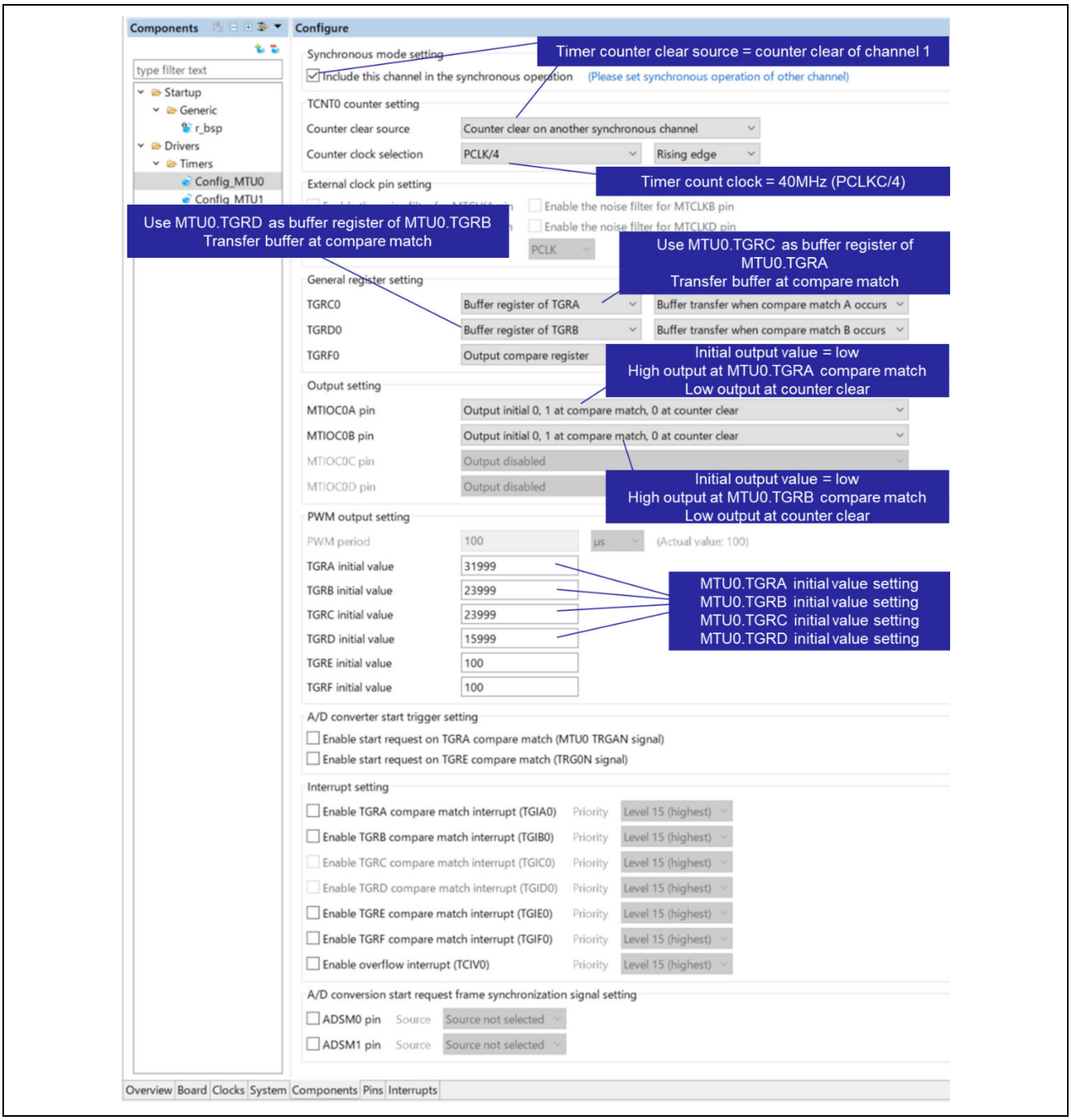


Figure 3.24 MTU0 Settings

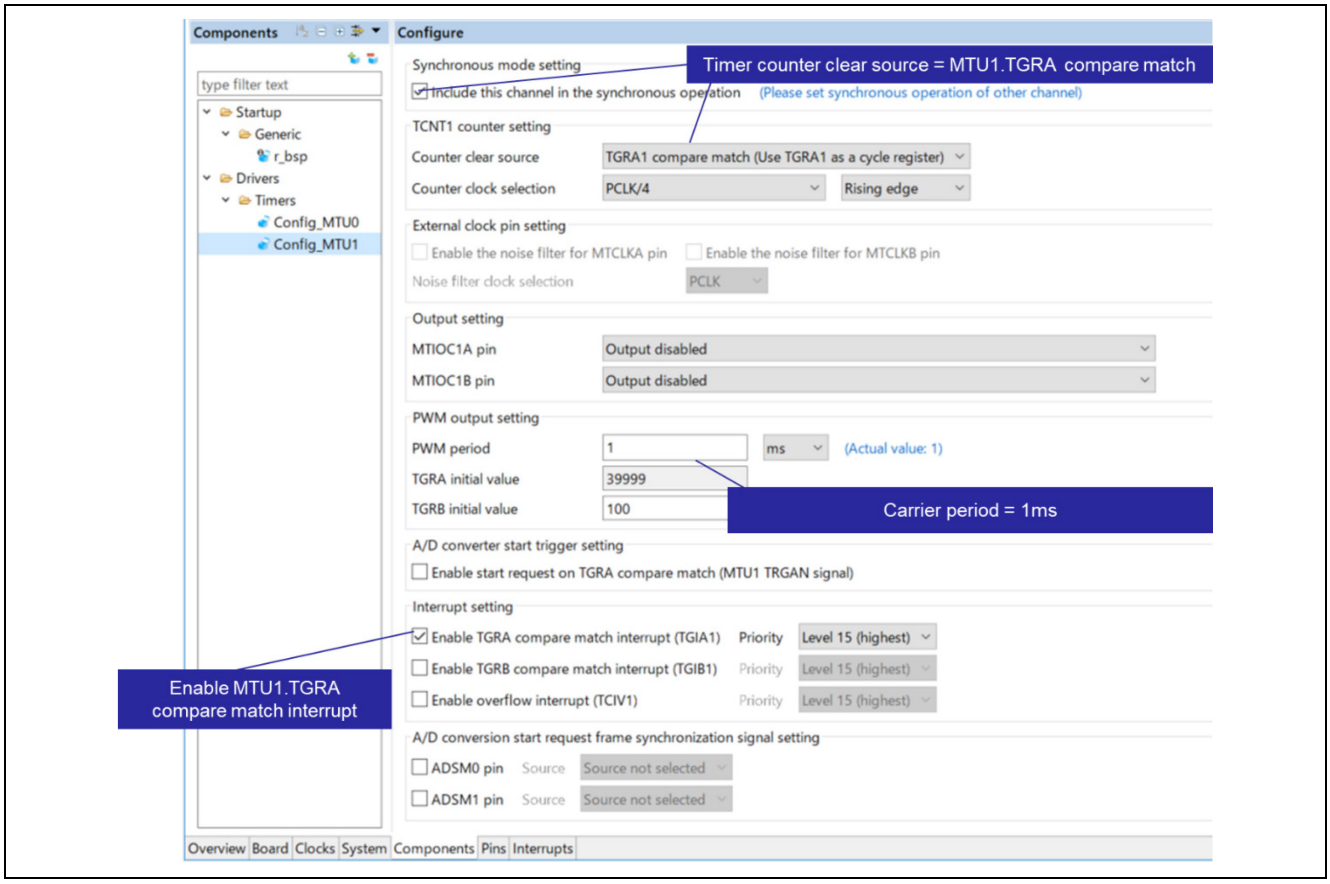


Figure 3.25 MTU1 Settings

3.5.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

In the main function, count start function mtu0_mtu1_start is read and counting is started.

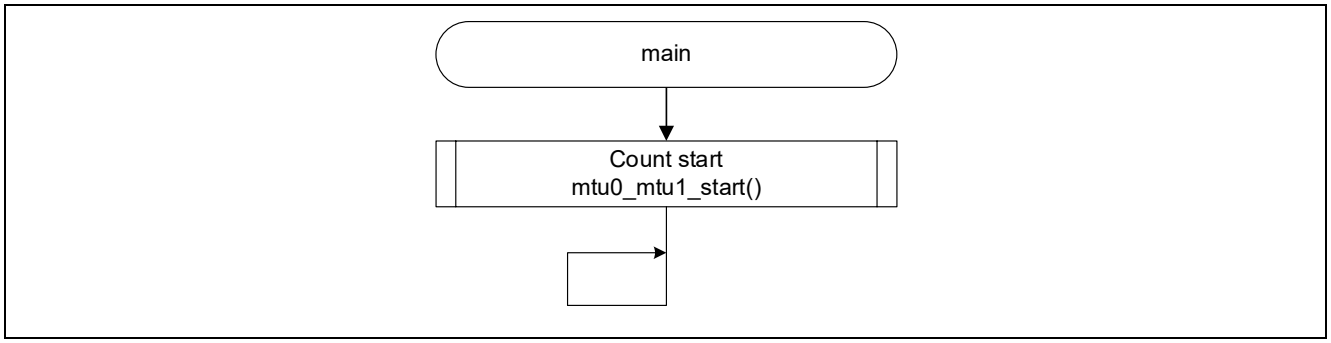


Figure 3.26 main Function

In the count start function, the MTU0 and MTU1 counting is started after the TGIA1 interrupt is enabled

This function is newly created after code generation by the Smart Configurator.

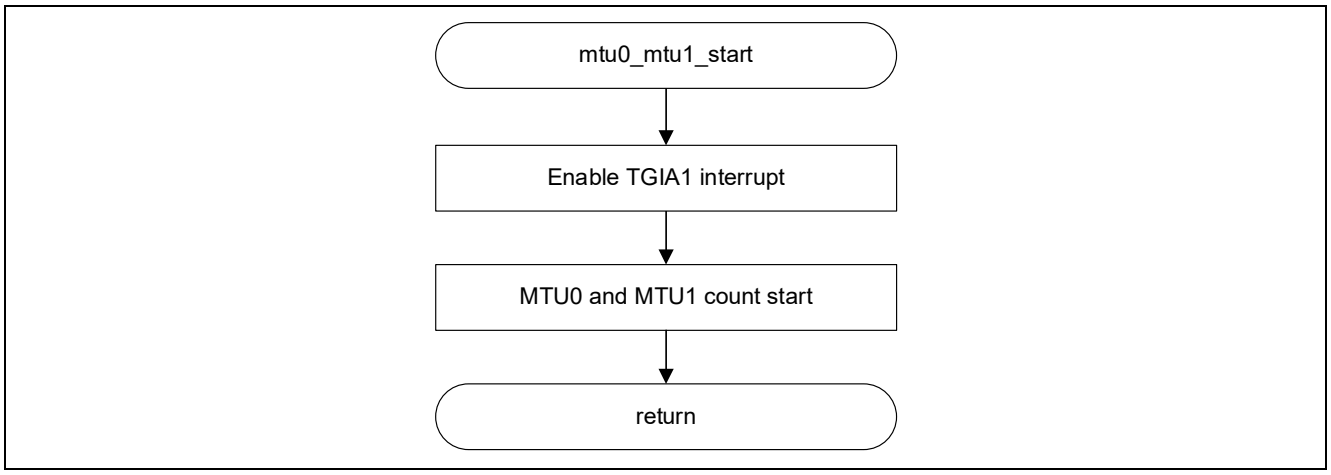


Figure 3.27 Count Start Function

The TGIA1 interrupt handler function changes the value according to the current values of the MTU0.TGRC and MTU0.TGRD registers.

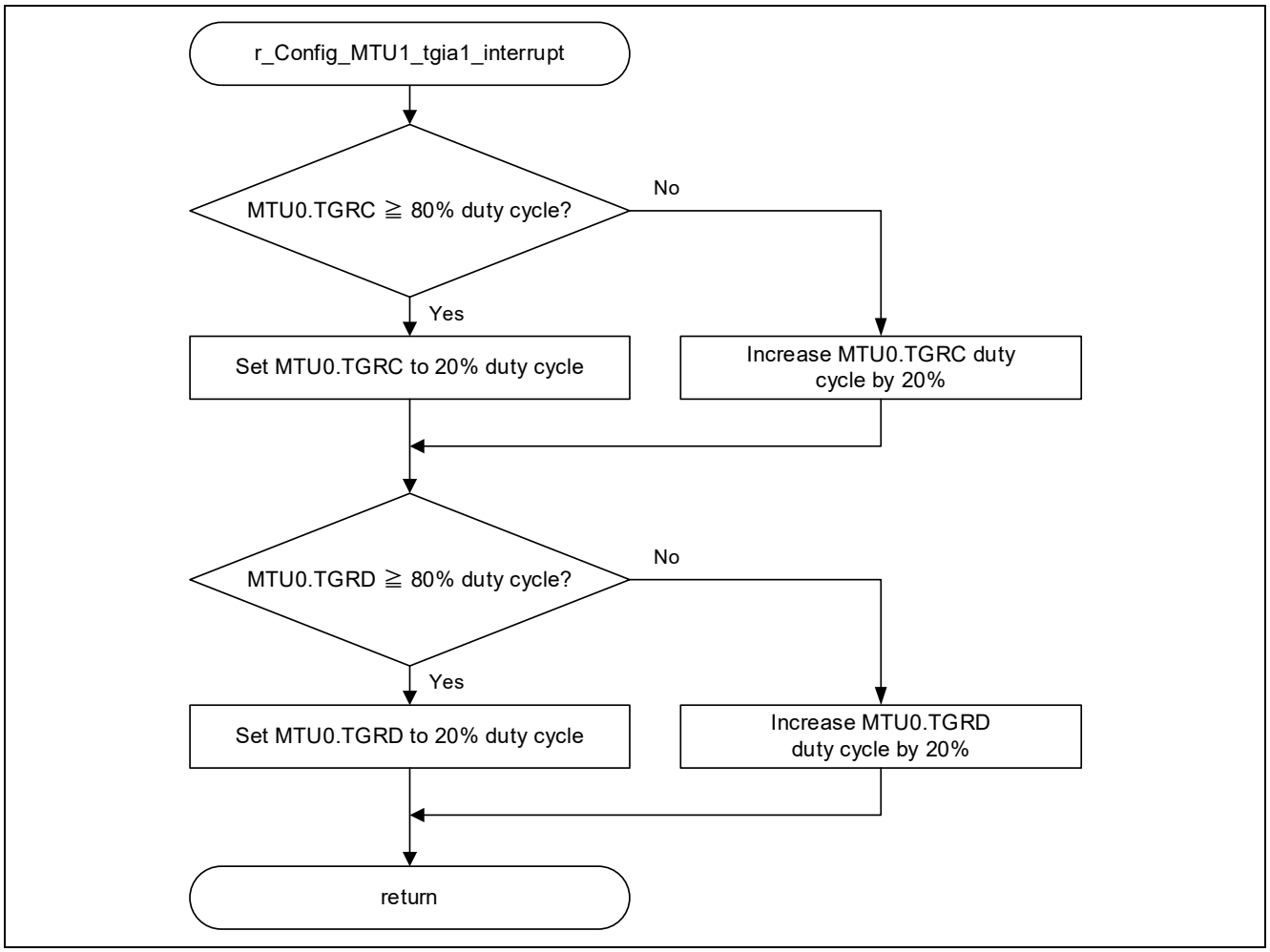


Figure 3.28 TGIA1 Interrupt Handler Function

3.5.5 Usage Notes

3.5.5.1 Counting Starts for Multiple Channels

In this sample code, the SCH0 and SCH1 bits of timer counter synchronous start register TCSYSTR are set at the same time in the mtu0_mtu1_start function to start MTU0 and MTU1 counting at the same time.

When using the R_Config_MTU0_Start and R_Config_MTU1_Start functions generated by the Smart Configurator, the counting starts timing may not be the same because each of the functions are read. It is possible to start the MTU0 and MTU1 counting at the same time by setting the CST0 and CST1 bits of the TSTRA timer start register at the same time.

Refer to RX66T Group User's Manual: Hardware, section 22.2.17 Timer Start Registers (TSTRA, TSTRB, TSTR) and section 22.2.19 Timer Counter Synchronous Start Register (TCSYSTR).

3.5.5.2 Contention between Buffer Register Modify Operation and Compare Match

When the buffer transfer timing is set at compare match, the data in the buffer register before the modification is transferred to the duty register if a compare match occurs in the buffer register write cycle.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.7 Contention between Buffer Register Write Operation and Compare Match.

3.5.5.3 Buffer Register Modify Delay

In this sample code, the transfer from buffer register TGRC (TGRD) to TGRA (TGRB) is executed when a TGRA (TGRB) compare match occurs. Modification of the TGRC (TGRD) value must be completed before the next compare match occurs.

If the buffer register modification is delayed, the TGRA (TGRB) value may not be updated by the time the TGRA (TGRB) compare match occurs in the next cycle. In such a case, consider transferring the buffer when the counter clear occurs.

Refer to section 3.6 PWM Mode 2 Count Clear for a sample code of PWM Mode 2 that performs buffer transfer when counter clear occurs.

3.5.5.4 Setting a Value Greater than the Duty Register

In this sample code, the transfer from buffer register TGRC to TGRA is performed when a TGRA compare match occurs.

If a value greater than the value currently set in TGRA is set in TGRA after a TGRA compare match occurs, two TGRA compare matches may occur in one cycle.

For details, refer to (1) in Figure 3.23.

3.6 PWM Mode 2 Count Clear

- Target sample code file name: r01an5995_rx66t_mtu3_pwm2_cntclear.zip

3.6.1 Overview

The MTU PWM mode can be used to output PWM waveforms with duty cycles 0% to 100% according to the TGR register setting.

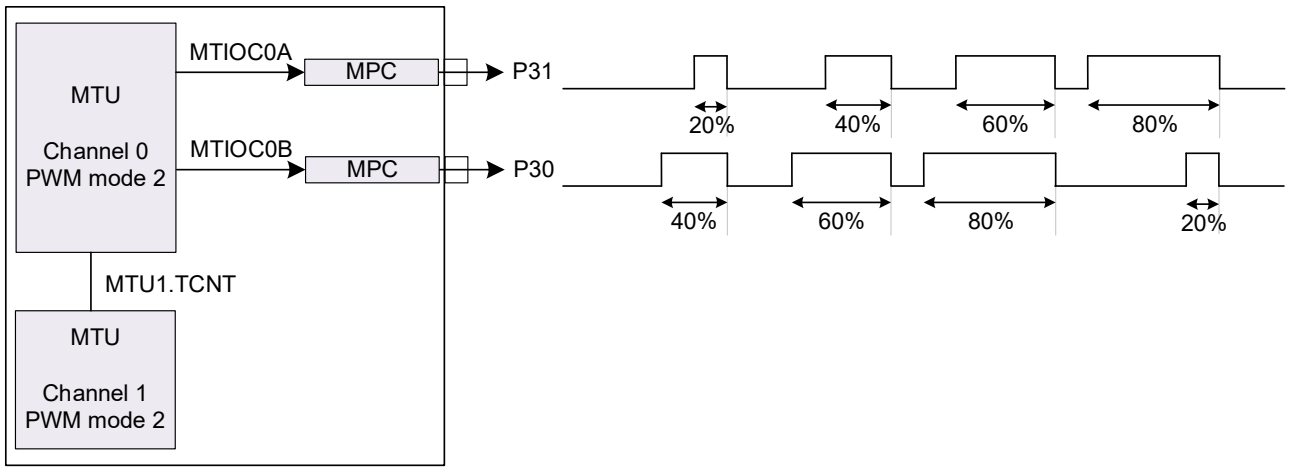
This sample code describes a sample code that uses PWM mode 2 and repeats waveform output with duty cycles 20% → 40% → 60% → 80% → 20% →⋯. The duty cycle is changed by using buffer registers TGRC and TGRD to transfer the values of TGRC and TGRD to duty registers TGRA and TGRB when a counter clear occurs. Since the TGRA to TGRD registers of MTU channel 0 are used for the duty register and buffer register, the period register uses the TGRA register of channel 1.

The following list provides the MTU settings used in the sample code.

- Use PWM mode 2
- Use channels 0 and 1
- Initial output value = low
- Carrier period = 1ms
- Timer count clock = 40MHz (PCLKC/4)
- Use MTU1.TGRA as period register
  - Timer counter clear source = MTU1.TGRA compare match
  - Low output at MTU1.TGRA compare match
- Use MTU0.TGRA as duty register
  - High output at MTU0.TGRA compare match
- Use MTU0.TGRB as duty register
  - High output at MTU0.TGRB compare match
- Use buffer register
  - Use MTU0.TGRC as buffer register of MTU0.TGRA
  - Use MTU0.TGRD as buffer register of MTU0.TGRB
  - Buffer transfer when counter clear occurs
- Duty changes at each cycle
  - Change duty cycle at MTU1.TGRA compare match interrupt
  - Refer to Figure 3.30 for details on timing for duty cycle changes

Set in Smart Configurator.
For Setting Methods,
refer to section 3.6.3.

PWM mode 2 output for this sample code is shown below.

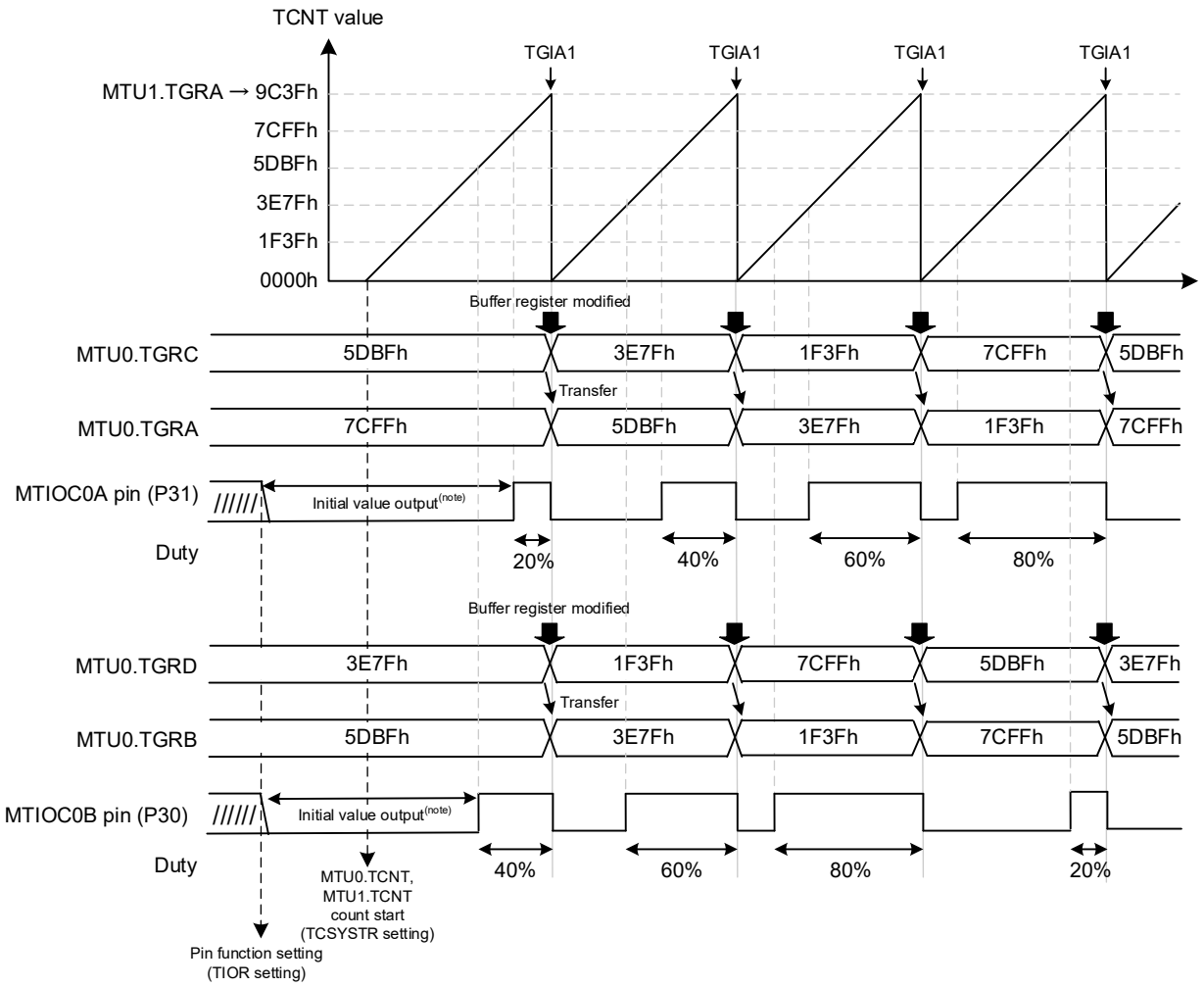


MTU: Generates PWM mode 2 waveform
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

Figure 3.29 PWM Mode 2 Output

3.6.2 Operation Details

The sample code operations are shown in Figure 3.30. The settings of the duty cycle are changed with each period by modifying the value of buffer registers MTU0.TGRC and MTU0.TGRD at the compare match interrupt (TGIA1) of period register MTU1.TGRA. The MTU0.TGRC and MTU0.TGRD values are transferred to duty registers MTU0.TGRA and MTU0.TGRB when a counter clear occurs.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.30 Sample Code Operations

3.6.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table 3.9 Adding Components

Item	Description	
Component	PWM Mode Timer	
Configuration name	Config_MTU0	Config_MTU1
Operation	PWM Mode 2	
Resource	MTU0	MTU1

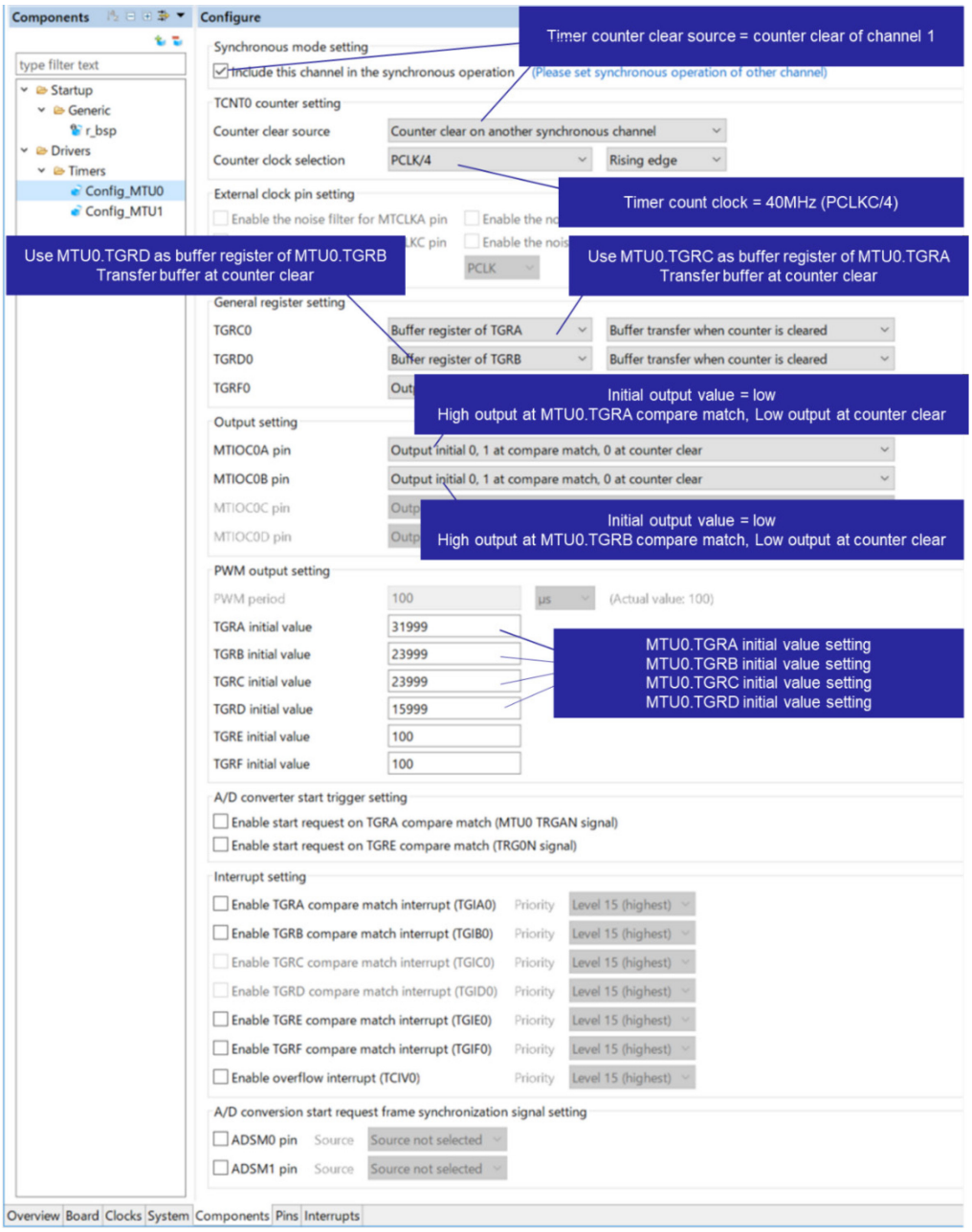


Figure 3.31 MTU0 Setting

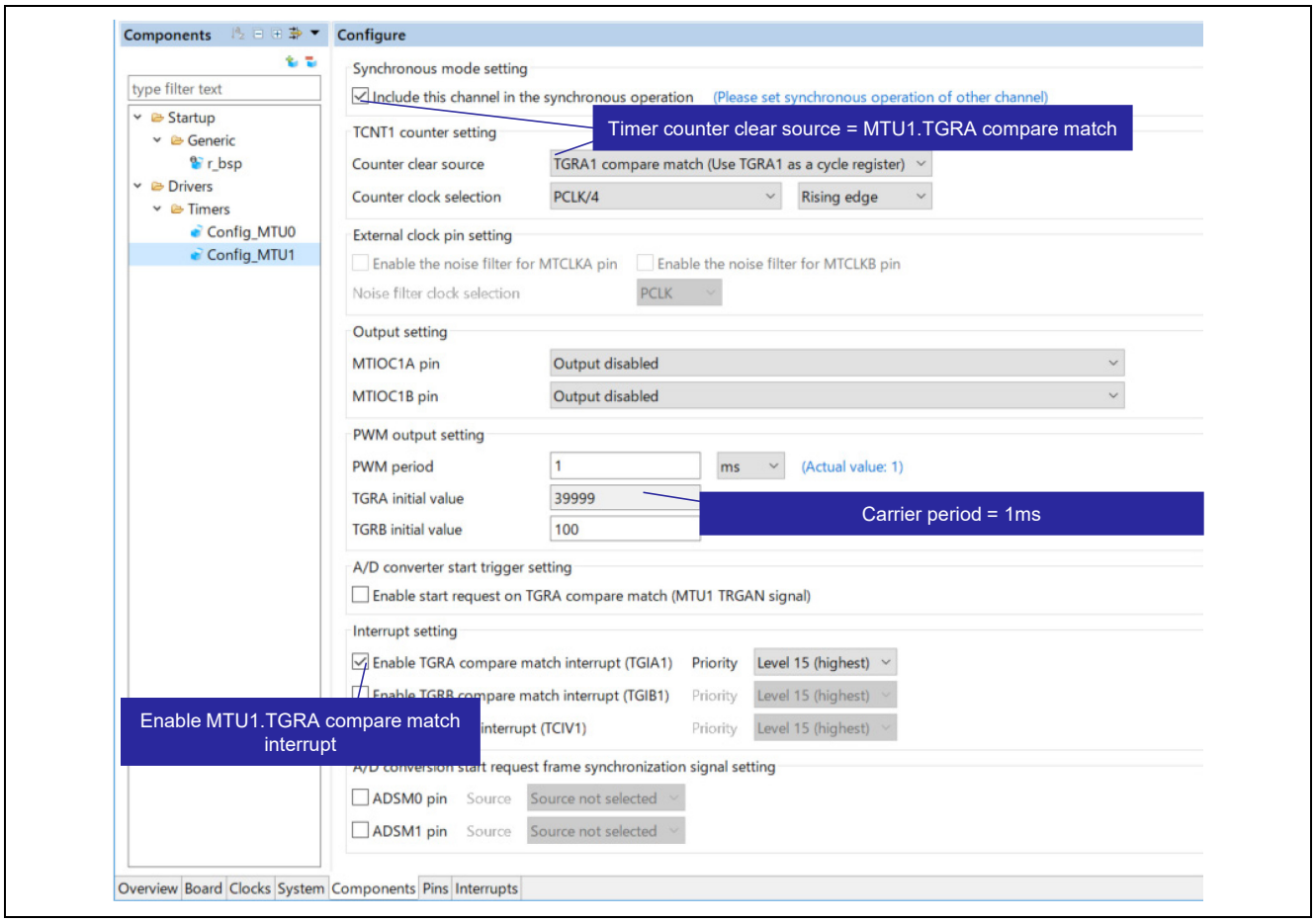


Figure 3.32 MTU1 Setting

3.6.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

In the main function, count start function mtu0_mtu1_start is read, and counting is started.

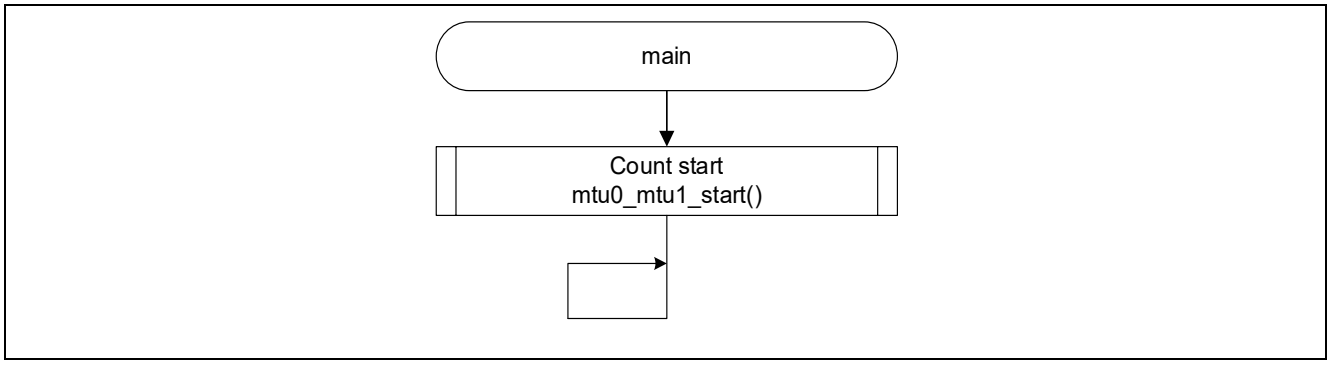


Figure 3.33 main Function

In the count start function, the MTU0 and MTU1 counting is started after the TGIA1 interrupt is enabled.

This function is newly created after code generation by the Smart Configurator.

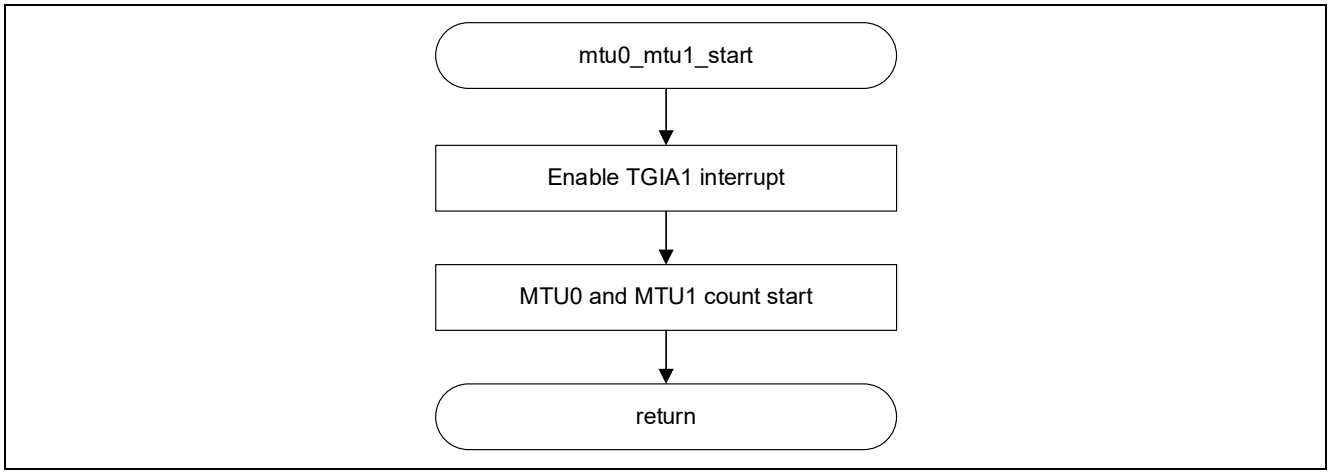


Figure 3.34 Count Start Function

The TGIA1 interrupt handler function changes the value according to the current values of the MTU0.TGRA and MTU0.TGRB registers.

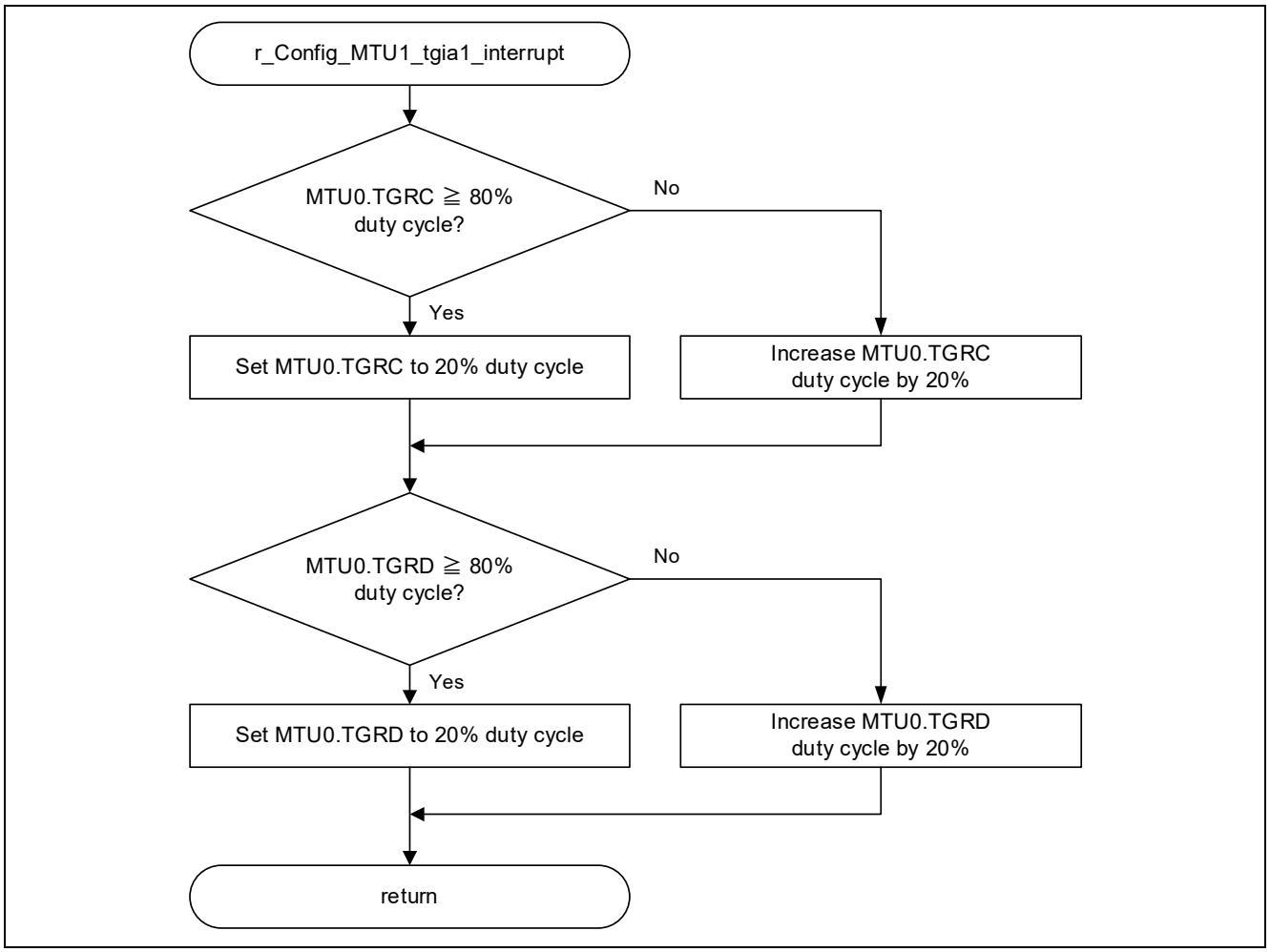


Figure 3.35 TGIA1 Interrupt Handler Function

3.6.5 Usage Notes

3.6.5.1 Counting Starts for Multiple Channels

In this sample code, the SCH0 and SCH1 bits of timer counter synchronous start register TCSYSTR are set at the same time in the mtu0_mtu1_start function to start the MTU0 and MTU1 counting at the same time.

When using the R_Config_MTU0_Start and R_Config_MTU1_Start function generated by the Smart Configurator, the counting start timing may not be the same because each of the functions are read. It is possible to start the MTU0 and MTU1 counting at the same time by setting the CST0 and CST1 bits of the TSTRA timer start register at the same time.

Refer to RX66T Group User's Manual: Hardware, section 22.2.17 Timer Start Registers (TSTRA, TSTRB, TSTR) and section 22.2.19 Timer Counter Synchronous Start Register (TCSYSTR).

3.6.5.2 Contention between Buffer Register Modify Operation and Counter Clear

If the buffer transfer timing is set at counter clear, the data in the buffer register before the modification is transferred to the duty register if a counter clear occurs in the buffer register write cycle.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.8 Contention between Buffer Register Write and TCNT Clear Operations.

3.6.5.3 Buffer Register Modify Delay

In this sample code, the transfer from buffer register TGRC (TGRD) to TGRA (TGRB) is executed when a counter clear occurs. Modification of the TGRC (TGRD) value must be completed before the next counter clear occurs.

If buffer register TGRC (TGRD) modification is delayed, the expected duty cycle cannot be output.

3.7 PWM Mode 2 Without Buffer Register

- Target sample code file name: r01an5995_rx66t_mtu3_pwm2.zip

3.7.1 Overview

The MTU PWM mode can be used to output PWM waveforms with duty cycles 0% to 100% according to the TGR register setting.

This sample code describes a sample code that uses PWM mode 2 and repeats waveform output with duty cycles 20% → 40% → 60% → 80% → 20% →⋯. The duty cycle is changed by updating the values of duty registers TGRA and TGRB without using the buffer register. Since the TGRA and TGRB registers of MTU channel 1 are used for the duty register and buffer register, the period register uses the TGRA register of channel 0.

The following list provides the MTU settings used in the sample code.

- Use PWM mode 2
- Use channels 0 and 1
- Initial output value = low
- Carrier period = 1ms
- Timer count clock = 40MHz (PCLKC/4)
- Use MTU0.TGRA as period register
  - Timer counter clear source = MTU0.TGRA compare match
  - Low output at MTU0.TGRA compare match
- Use MTU1.TGRA as duty register
  - High output at MTU1.TGRA compare match
- Use MTU1.TGRB as duty register
  - High output at MTU1.TGRB compare match
- Duty changes at each cycle
  - Change MTU1.TGRA duty cycle at MTU1.TGRA compare match interrupt
  - Change MTU1.TGRB duty cycle at MTU1.TGRB compare match interrupt
  - Refer to Figure 3.37 for details on timing for duty cycle changes

Set in Smart Configurator.
For setting methods,
refer to section 3.7.3.

PWM mode 2 output for this sample code is shown below.

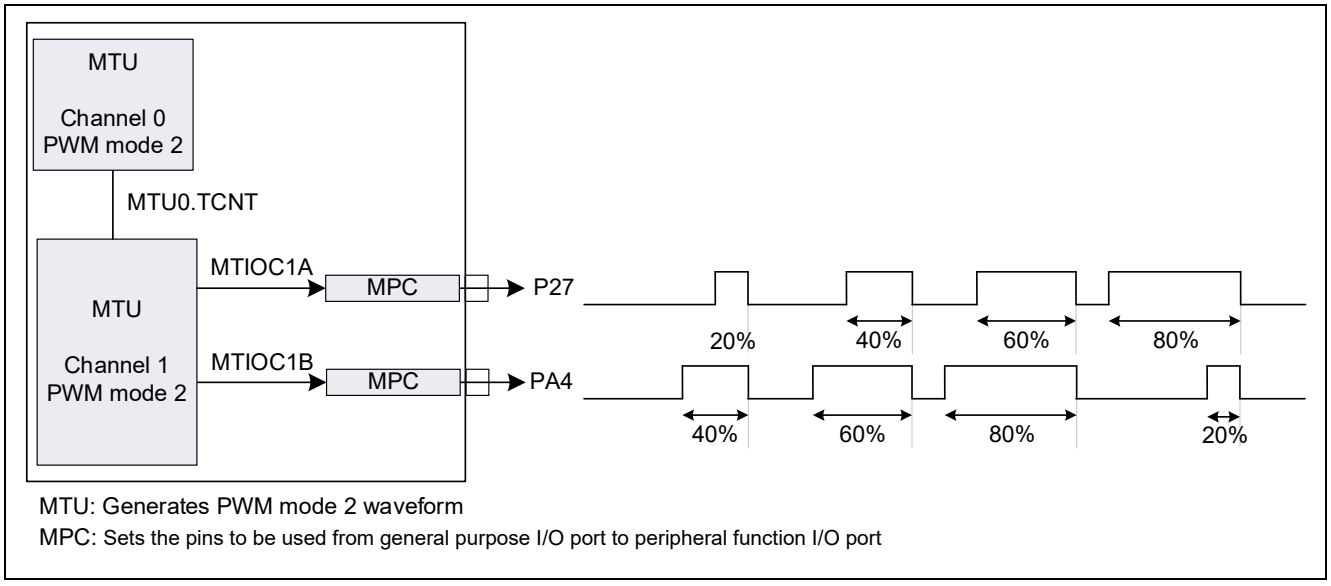
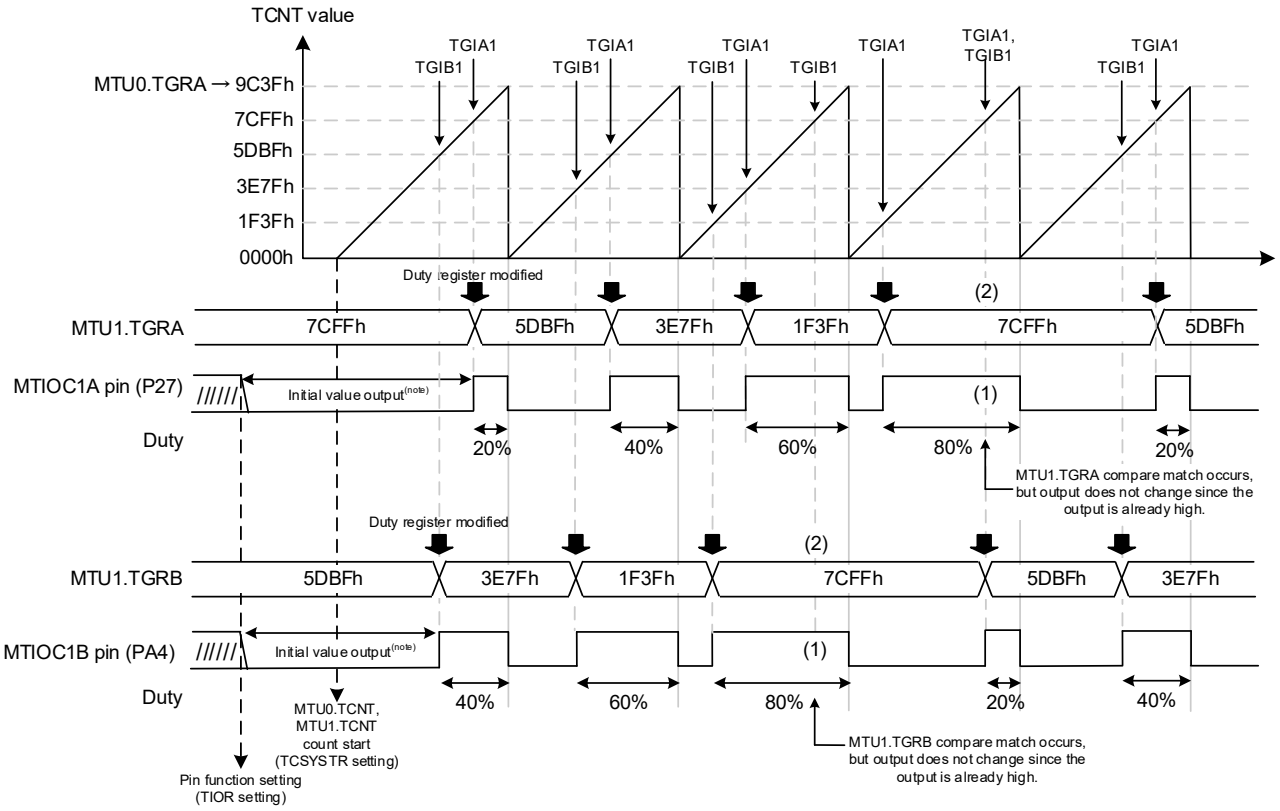


Figure 3.36 PWM Mode 2 Output

3.7.2 Operation Details

The sample code operations are shown in Figure 3.37. The settings of the duty cycle are changed with each period by modifying the value of duty register MTU1.TGRA at the MTU1.TGRA compare match interrupt (TGIA1) and duty register MTU1.TGRB at the MTU1.TGRB compare match interrupt (TGIB1)

When switching from duty cycle 80% to 20%, two compare matches occur in the same cycle, but the waveform does not change because the second compare match occurs during high output ((1) in figure below). The duty cycle setting is changed with each period, so the timing for modifying the MTU1.TGRA and MTU1.TGRB values needs to be adjusted ((2) in figure below).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.37 Sample Code Operation

3.7.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table 3.10 Adding Components

Item	Description	
Component	PWM Mode Timer	
Configuration name	Config_MTU0	Config_MTU1
Operation	PWM Mode 2	
Resource	MTU0	MTU1

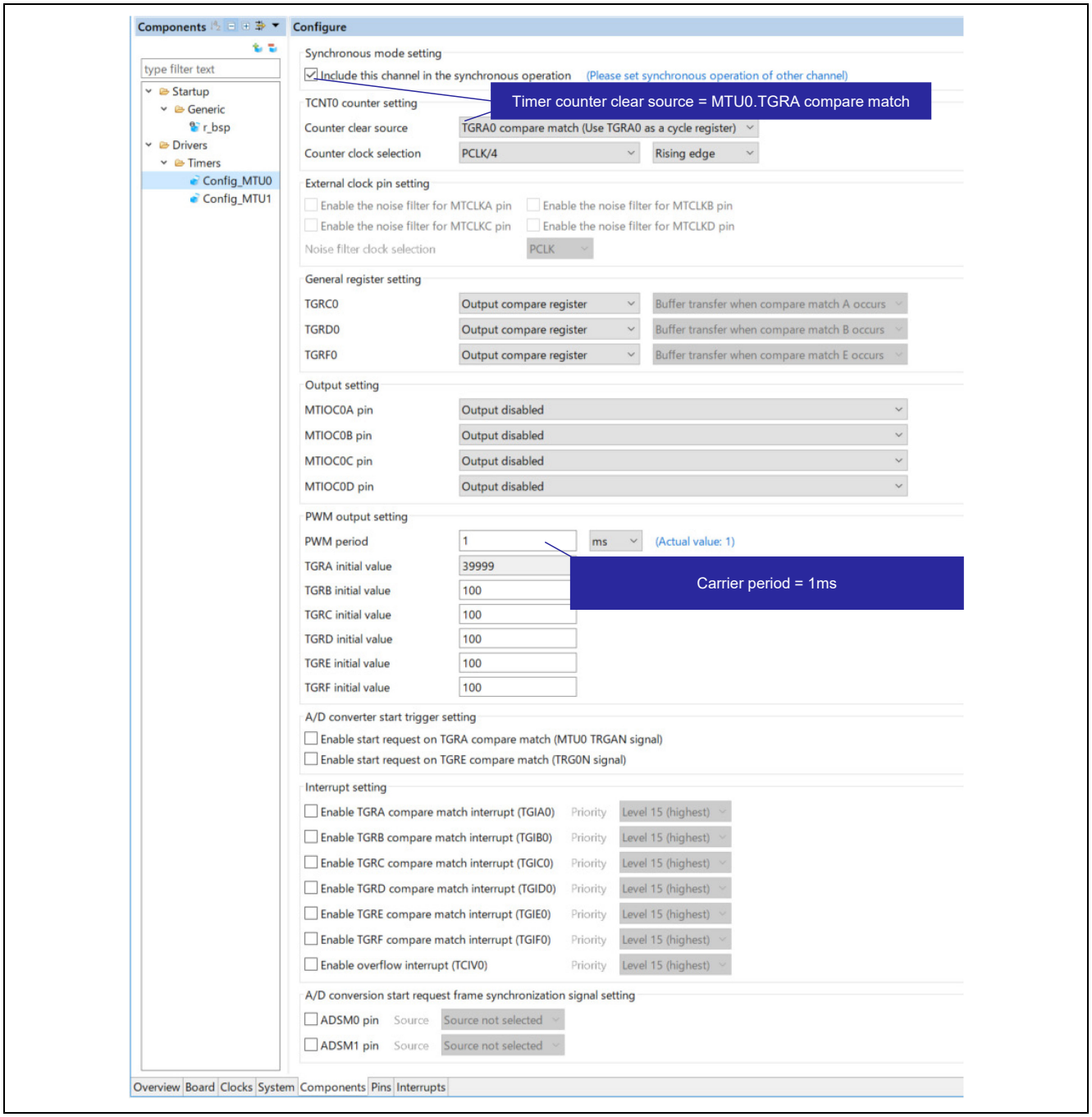


Figure 3.38 MTU0 Settings

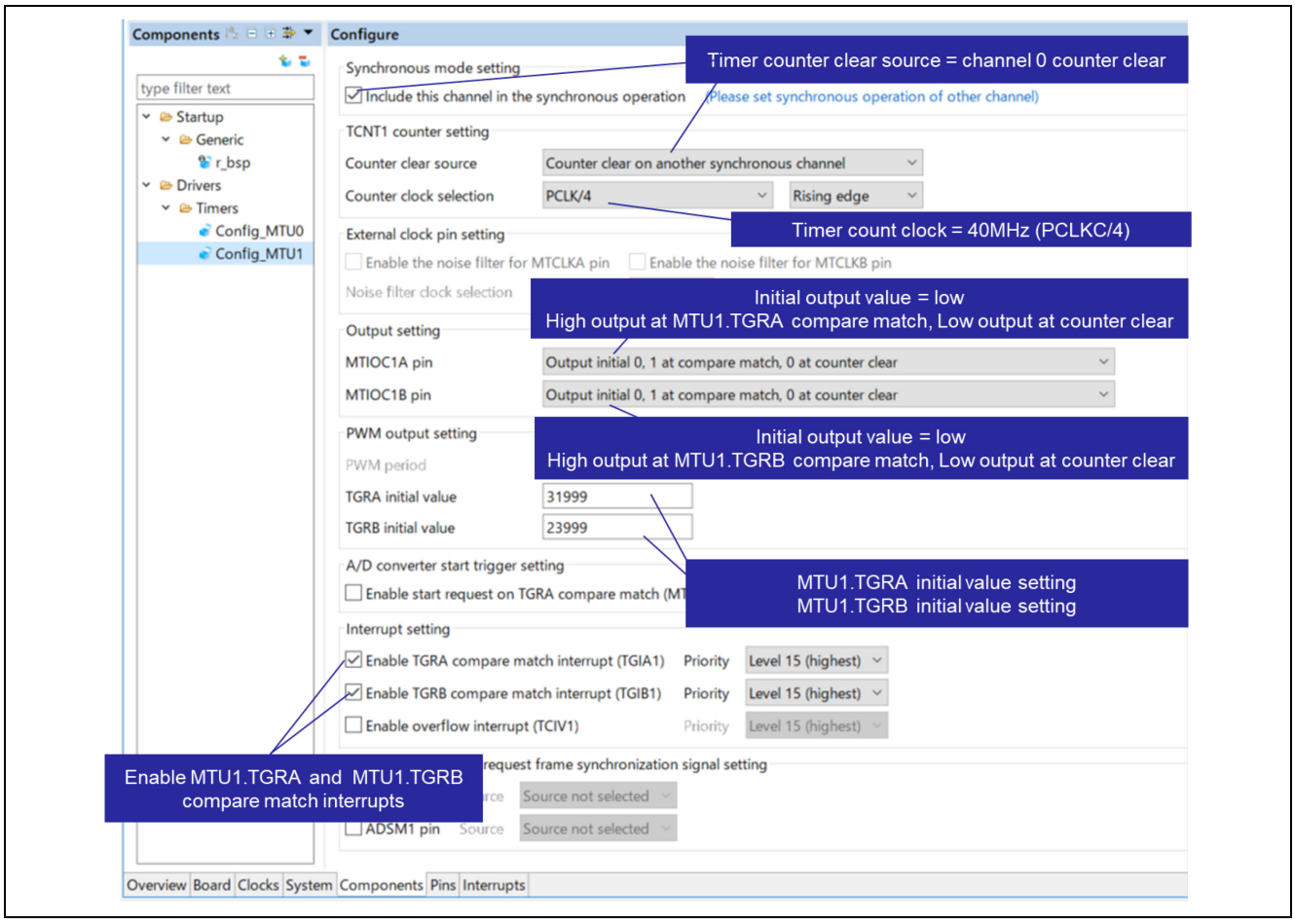


Figure 3.39 MTU1 Settings

3.7.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

In the main function, count start function mtu0_mtu1_start is read, and counting is started.

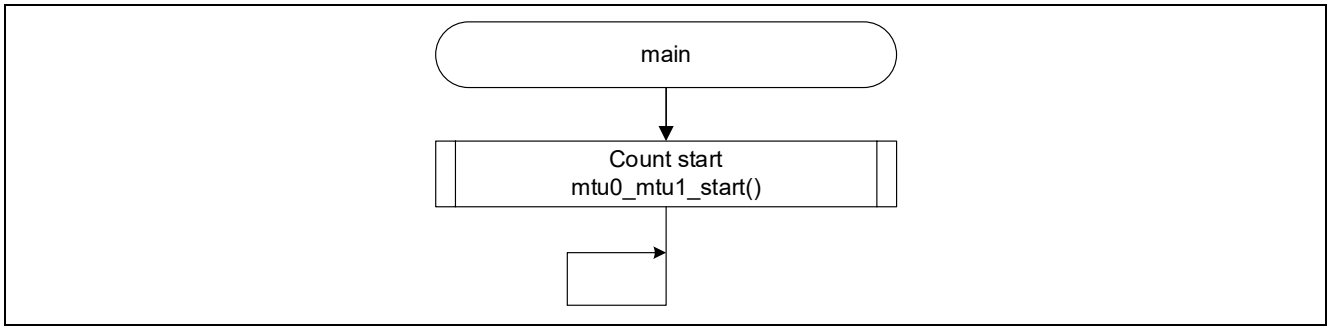


Figure 3.40 main function

In the count start function, the MTU0 and MTU1 counting is started after the TGIA1 and the TGIB1 interrupts are enabled.

This function is newly created after code generation by the Smart Configurator.

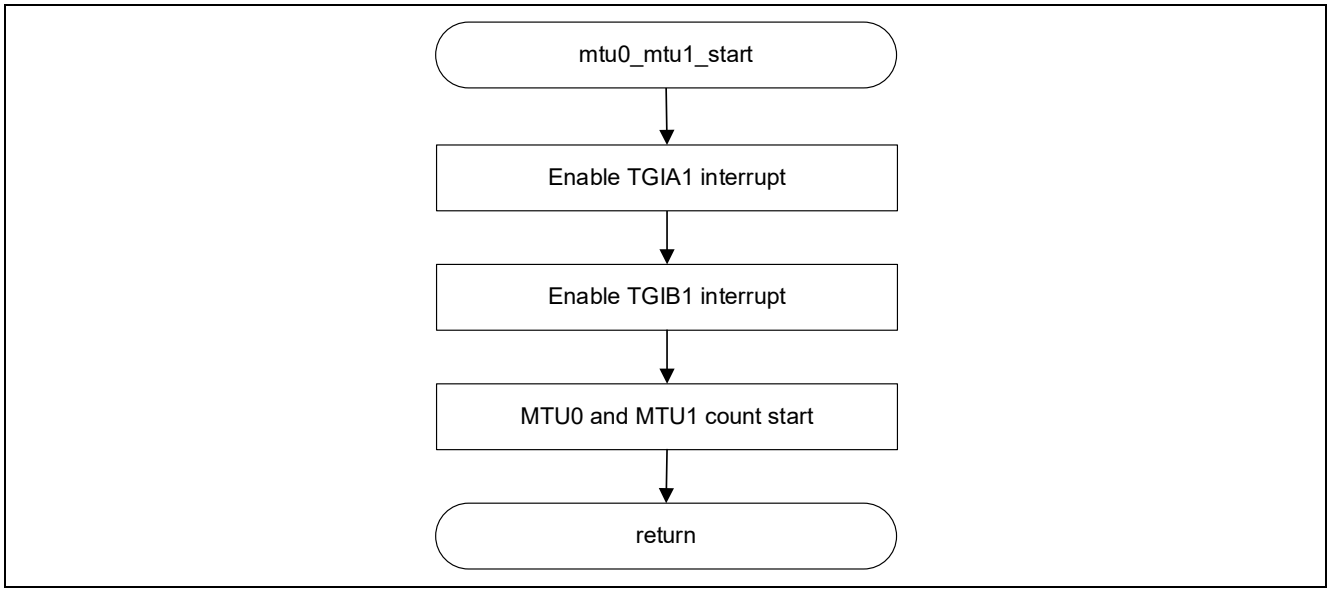


Figure 3.41 Count Start Function

The user initialization function R_Config_MTU1_Create_UserInit, which is executed before the main function, initializes variables. This function is called from within the R_Config_MTU1_Create function.

This function initializes the following variables used in this sample code.

- s_duty_prv_a: variable for retaining the value of the previous MTU1.TGRA register
- s_duty_prv_b: variable for retaining the value of the previous MTU1.TGRB register

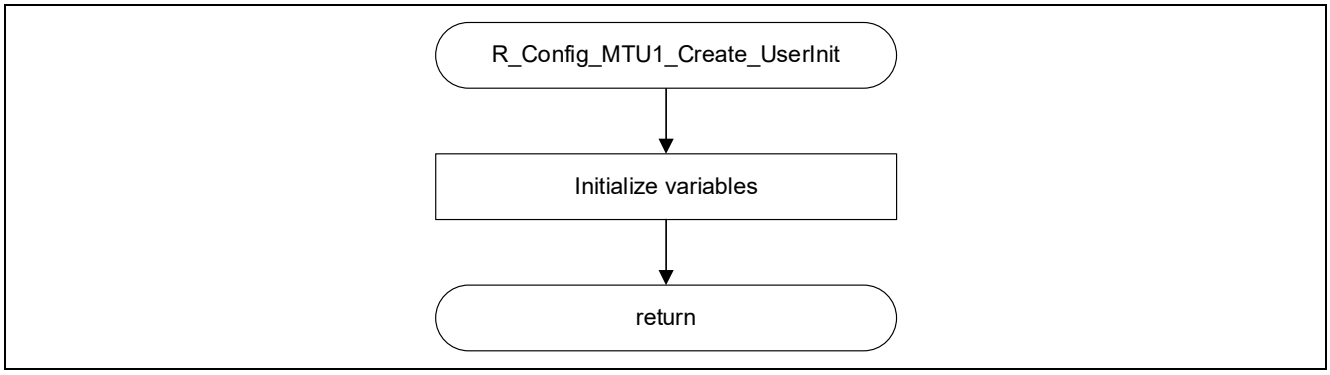


Figure 3.42 User Initialization Function

The TGIA1 interrupt handler function changes the value of the MTU1.TGRA register according to the current value of the MTU1.TGRA register and the value set in the previous MTU1.TGRA register.

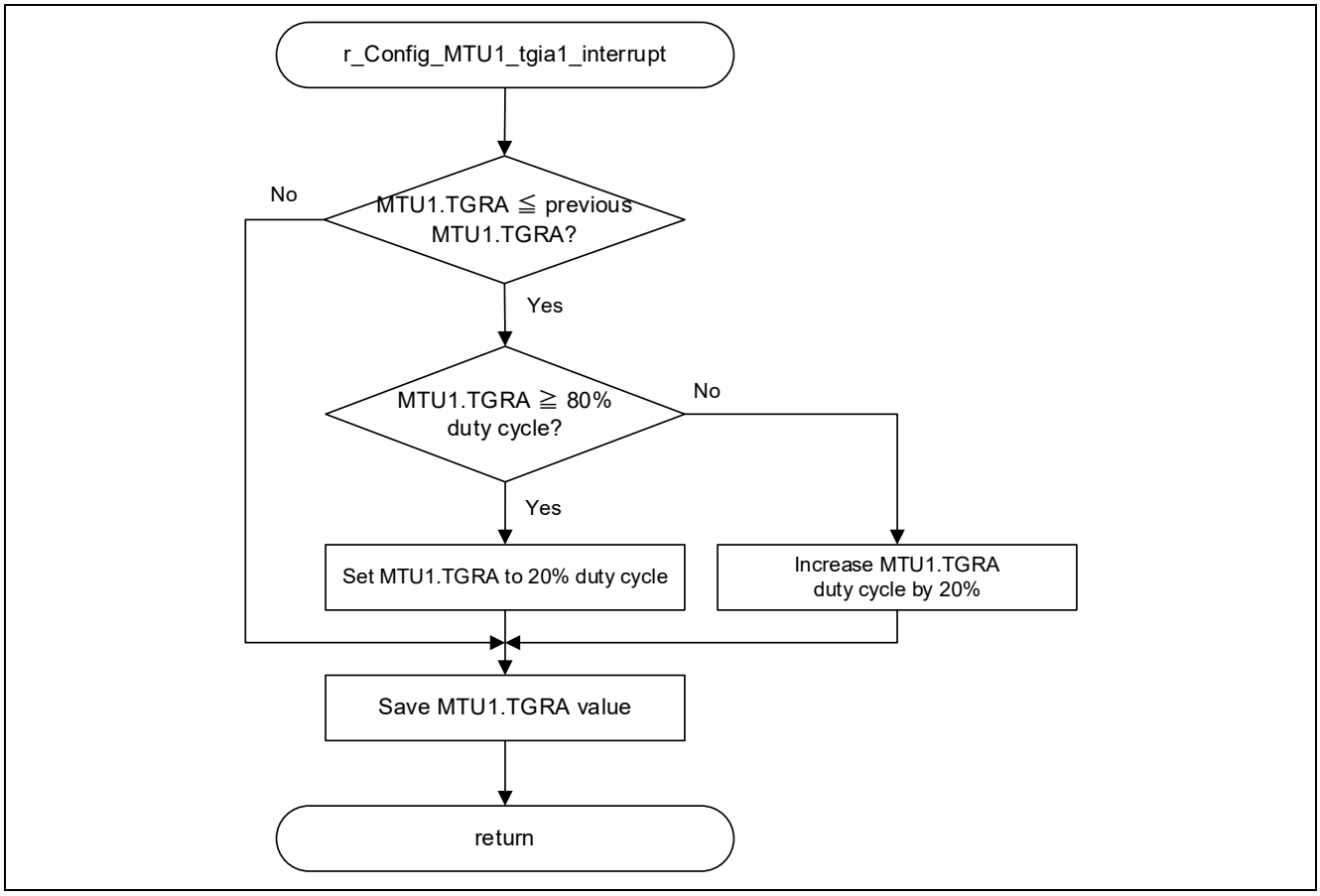


Figure 3.43 TGIA1 Interrupt Handler Function

The TGIB1 interrupt handler function changes the value of the MTU1.TGRB register according to the current MTU1.TGRB register and the value of the previous MTU1.TGRB register.

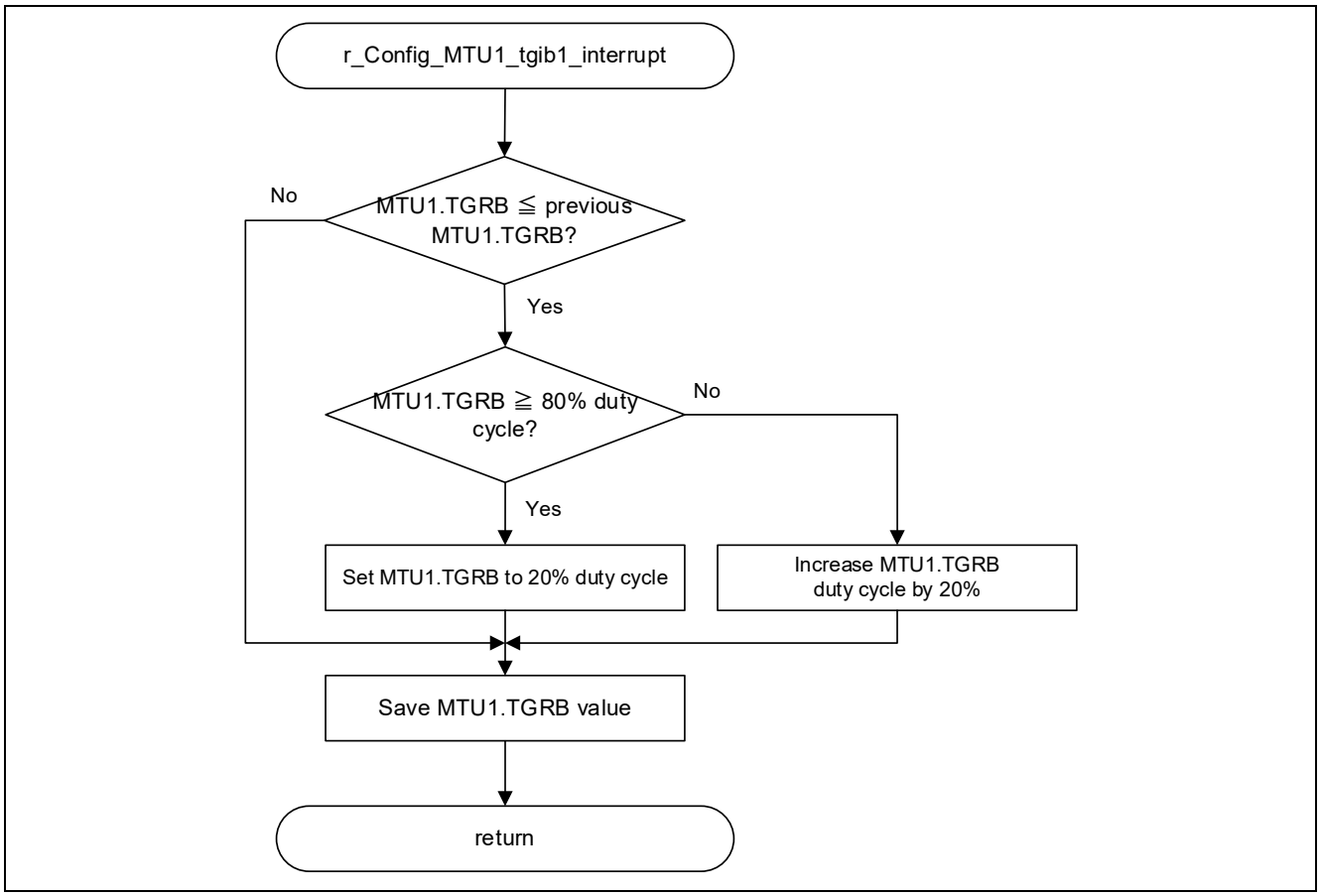


Figure 3.44 TGIB1 Interrupt Handler Function

3.7.5 Usage Notes

3.7.5.1 Counting Starts for Multiple Channels

In this sample code, the SCH0 and SCH1 bits of timer counter synchronous start register TCSYSTR are set at the same time in the mtu0_mtu1_start function to start the MTU0 and MTU1 counting at the same time.

When using the R_Config_MTU0_Start and R_Config_MTU1_Start functions generated by the Smart Configurator, the counting start timing may not be the same because each of the functions are read. It is possible to start the MTU0 and MTU1 counting at the same time by setting the CST0 and CST1 bits of the TSTRA timer start register at the same time.

Refer to RX66T Group User's Manual: Hardware, section 22.2.17 Timer Start Registers (TSTRA, TSTRB, TSTR) and section 22.2.19 Timer Counter Synchronous Start Register (TCSYSTR).

3.7.5.2 Contention between TGR Register Write and Compare Match

If a compare match occurs in the TGR register write cycle, a write to the TGR register write is performed and a compare match signal is also generated.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.6 Contention between TGR Write Operation and Compare Match.

3.7.5.3 Setting a Value Greater than the Duty Register

In this sample, duty register TGRA is modified when the MTU1.TGRA compare match interrupt (TGIA1) is generated; duty register TGRB is modified when the MTU1.TGRB compare match interrupt (TGIB1) is generated.

If a value greater than the value currently set in TGRA (TGRB) is set in TGRA (TGRB) after a TGRA (TGRB) compare match occurs, two TGRA (TGRB) compare matches may occur in one cycle.

For details, refer to (1) in Figure 3.37.

3.8 Reset-Synchronized PWM Mode

- Target sample code file name: r01an5995_rx66t_mtu3_reset_sync_pwm.zip

3.8.1 Overview

In the MTU's reset-synchronized PWM mode, 6 phases of positive and negative PWM waveforms (12 phases in total) that share a common wave transition point can be output by combining MTU3 and MTU4 with MTU6 and MTU7.

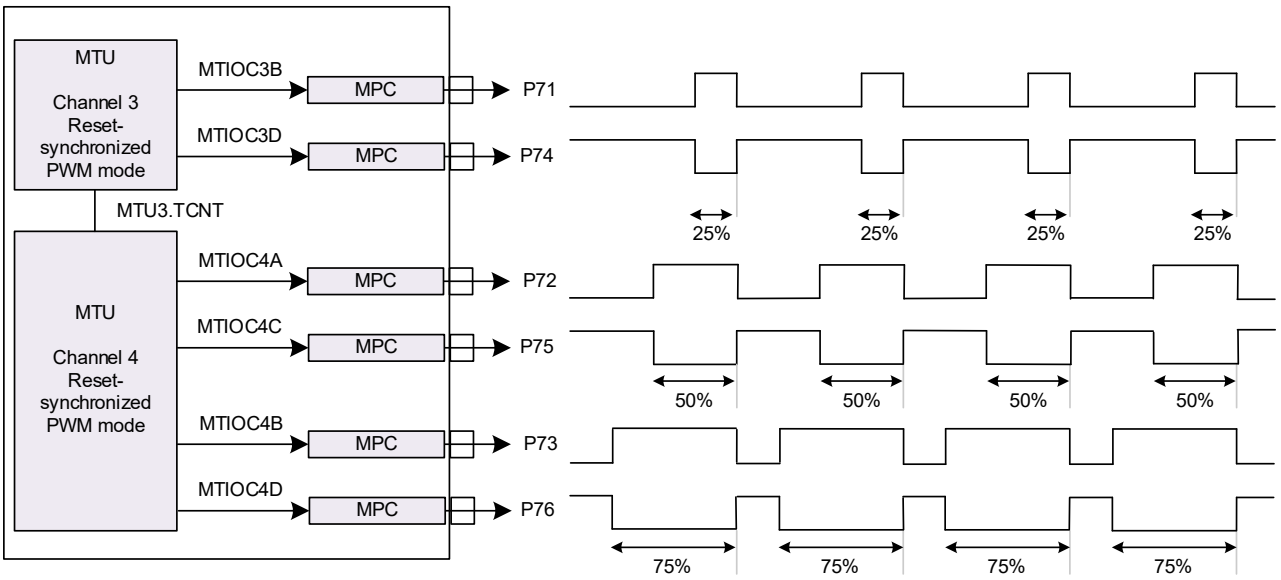
This sample code uses the reset-synchronized PWM mode with MTU3 and MTU4, and repeatedly outputs PWM waveforms (positive and negative phases) for 3 phases each, for a total of 6 phases, with a constant duty cycle.

The following list provides the MTU settings used in the sample code.

- Use channels 3 and 4
- Carrier period = 1ms
- Timer count clock = 40MHz (PCLKC/4)
- Use MTU3.TGRA as period register
  - Timer counter clear source = MTU3.TGRA compare match
- Use MTU3.TGRB as duty register
  - MTIOC3B pin (positive) initial output is low; high output at MTU3.TGRB compare match
  - MTIOC3D pin (negative) initial output is high; low output at MTU3.TGRB compare match
- Use MTU4.TGRA as duty register
  - MTIOC4A pin (positive) initial output is low; high output at MTU4.TGRA compare match
  - MTIOC4C pin (negative) initial output is high; low output at MTU4.TGRA compare match
- Use MTU4.TGRB as duty register
  - MTIOC4B pin (positive) initial output is low; high output at MTU4.TGRB compare match
  - MTIOC4D pin (negative) initial output is high; low output at MTU4.TGRB compare match
- Brushless DC motor control is invalid
- Use reset-synchronized PWM mode

Set in Smart Configurator.
For setting methods,
refer to section 3.8.3.

Reset-synchronized PWM mode output for this sample code is shown below.

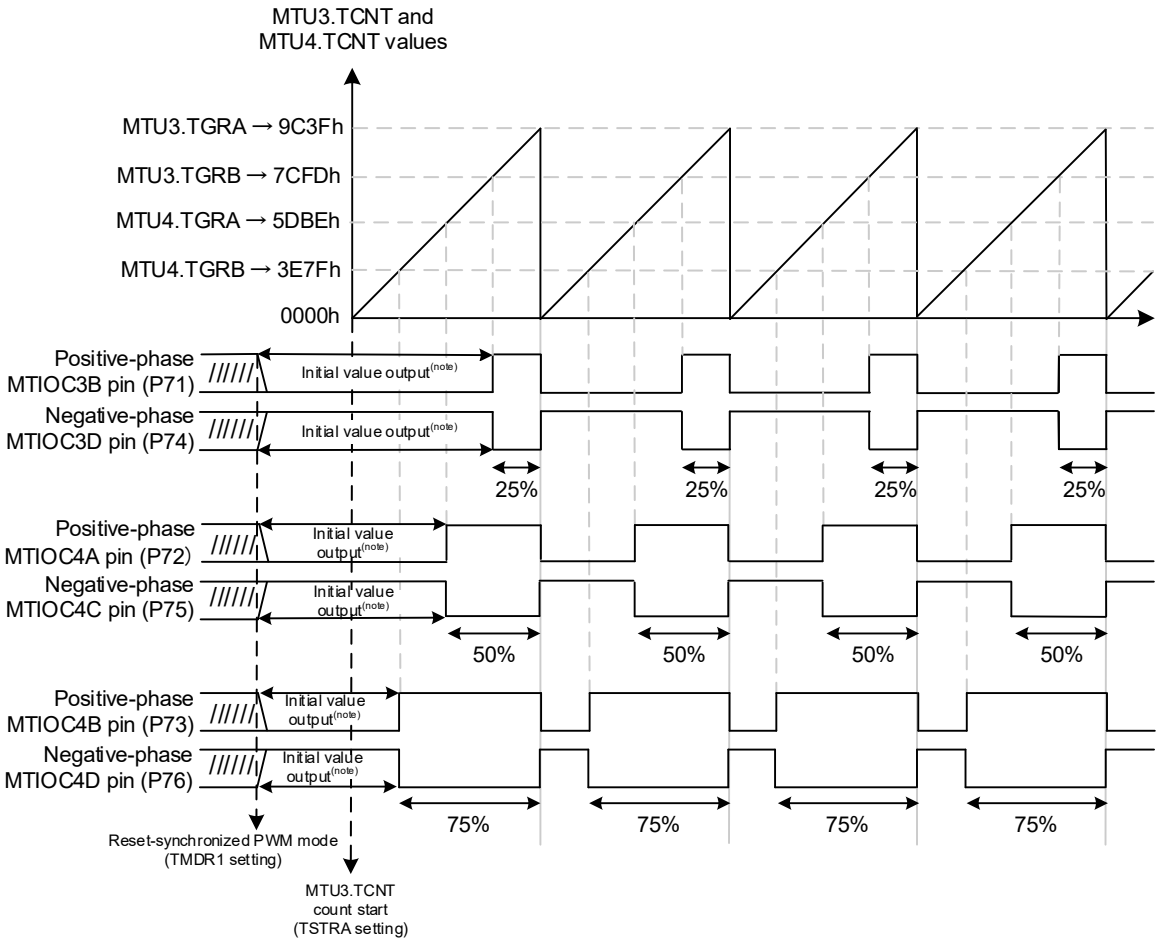


MTU: Generates reset-synchronized PWM waveform
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

Figure 3.45 Reset Synchronous PWM Mode Output

3.8.2 Operation Details

The sample code operations are shown in Figure 3.46. MTU3.TCNT and MTU4.TCNT operate as up-counters using the reset-synchronized PWM mode. The counters are cleared when a compare match occurs between MTU3.TCNT and MTU3.TGRA, and then begin increments from 0000h. Output from the PWM output pin toggles every time a compare match of MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB and a counter clear occurs.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.46 Sample Code Operations

3.8.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below.

As there is no reset-synchronized PWM mode component, the code is generated using the normal mode timer component.

For details on how to add components, refer to section 3.1.4 Adding Components.

Table 3.11 Adding Components

Item	Description	
Component	Normal Mode Timer Note	
Configuration name	Config_MTU3	Config_MTU4
Input capture/output compare pins	2 pins	4 pins
Resource	MTU3	MTU4

Note: After the code is generated using the normal mode timer component, set the reset-synchronized PWM mode in MTU3 User Initialization Function R_Config_MTU3_Create_UserInit. For details on register settings, refer to Figure 3.50.

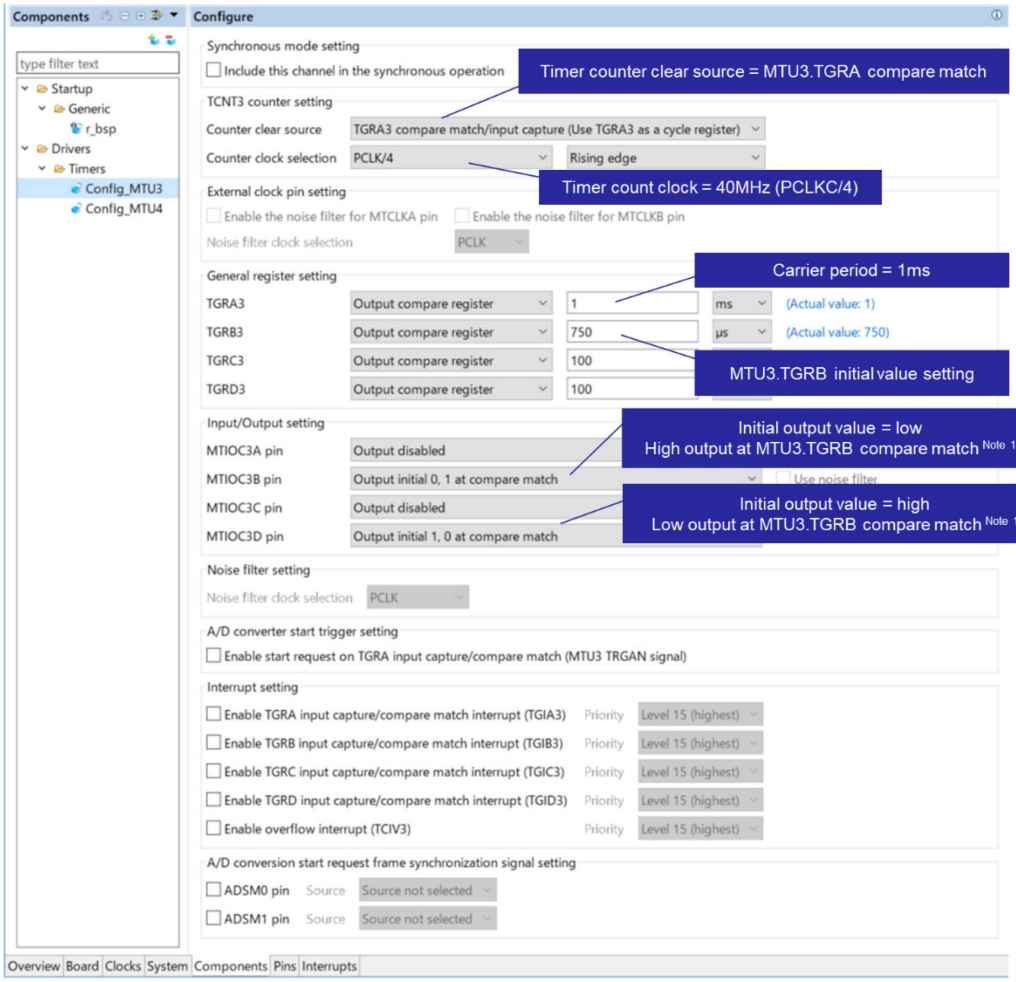


Figure 3.47 MTU3 Settings

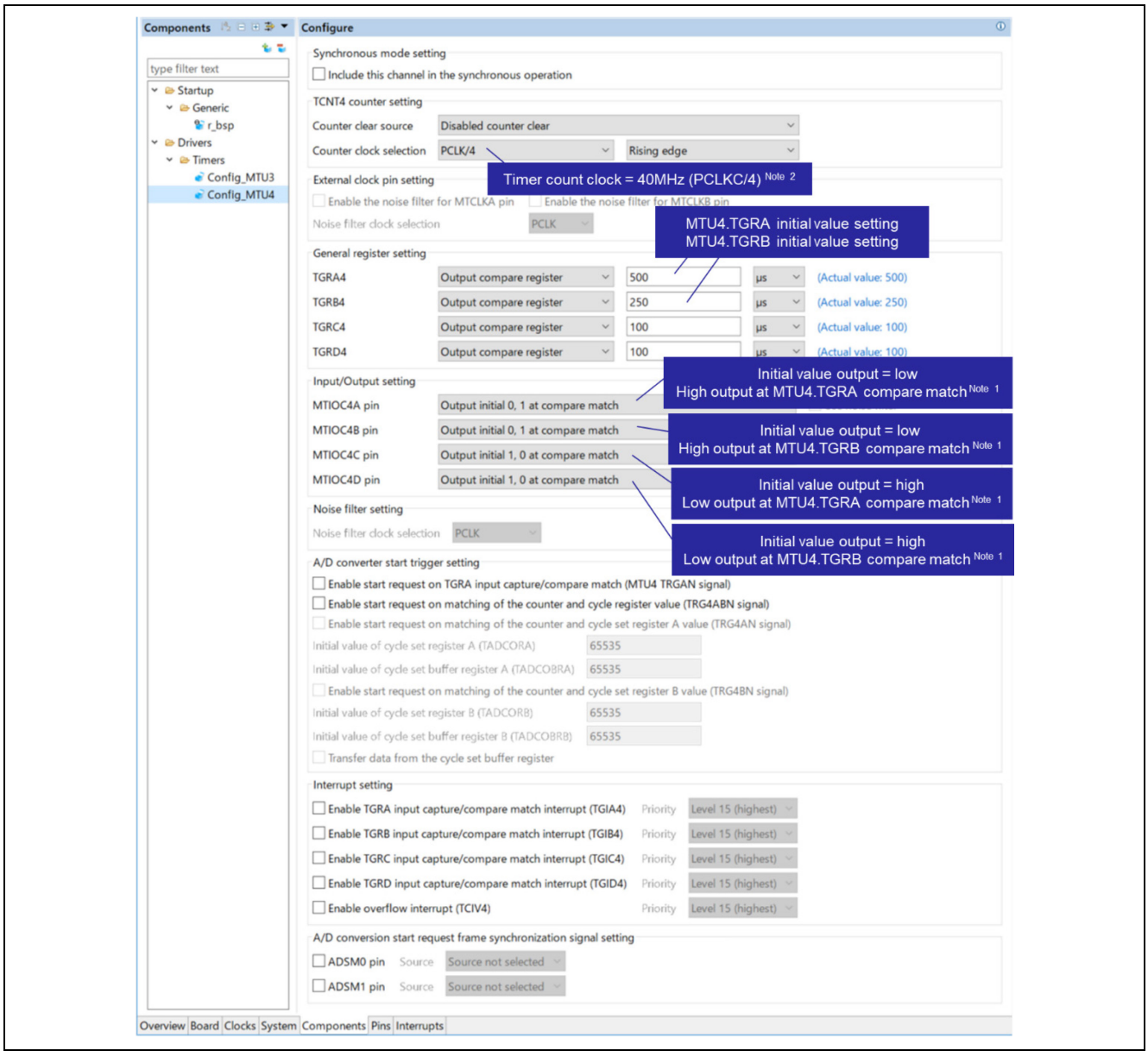


Figure 3.48 MTU4 Settings

Note 1: The code generated using the normal mode timer component outputs the TIOR register setting. The PWM output level for the reset-synchronized PWM mode is set in the TOCR1A register, not the TIOR register. As a result, initialize the TIOR register that is not necessary and set the TOCR1A register in user initialization functions R_Config_MTU3_Create_UserInit and R_Config_MTU4_Create_UserInit, respectively. Refer to Figure 3.50 for details on the R_Config_MTU3_Create_UserInit function, and Figure 3.51 for details on the R_Config_MTU4_Create_UserInit function.

Note 2: This setting causes the generated code to output the MTU4 pre-scaler selection bit (TCR.TPSC) setting. The reset-synchronized PWM mode counter clock is enabled in the MTU3 settings but not necessary for the MTU4 settings. As a result, initialize the TCR register that is not necessary in the user initialization function R_Config_MTU4_Create_UserInit. Refer to Figure 3.51 for details on the R_Config_MTU4_Create_UserInit function.

3.8.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

Counting is started in the main function. In reset-synchronized PWM mode, the start of MTU3.TCNT counting also starts the MTU4.TCNT counting.

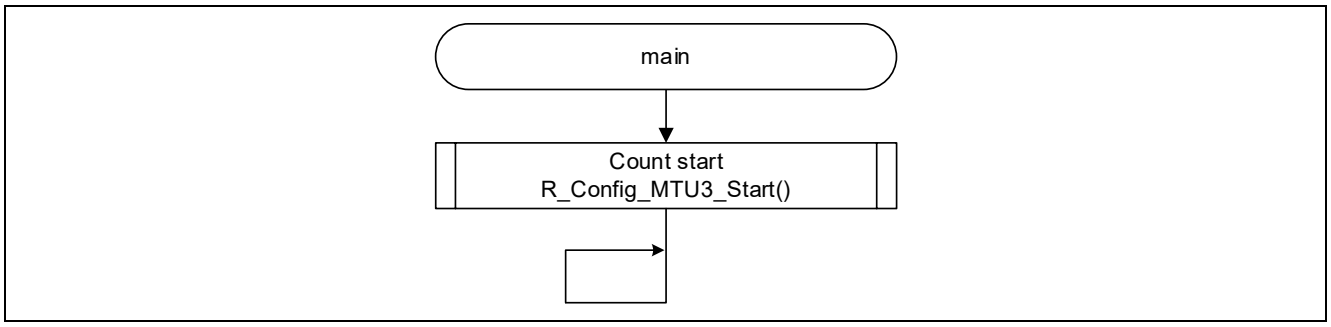


Figure 3.49 main Function

The user initialization function R_Config_MTU3_Create_UserInit, which is called by the R_Config_MTU3_Create function executed before the main function, sets the brushless DC motor control, PWM output level, and timer mode, and initializes the TIOR register that is not necessary in the reset-synchronized PWM mode.

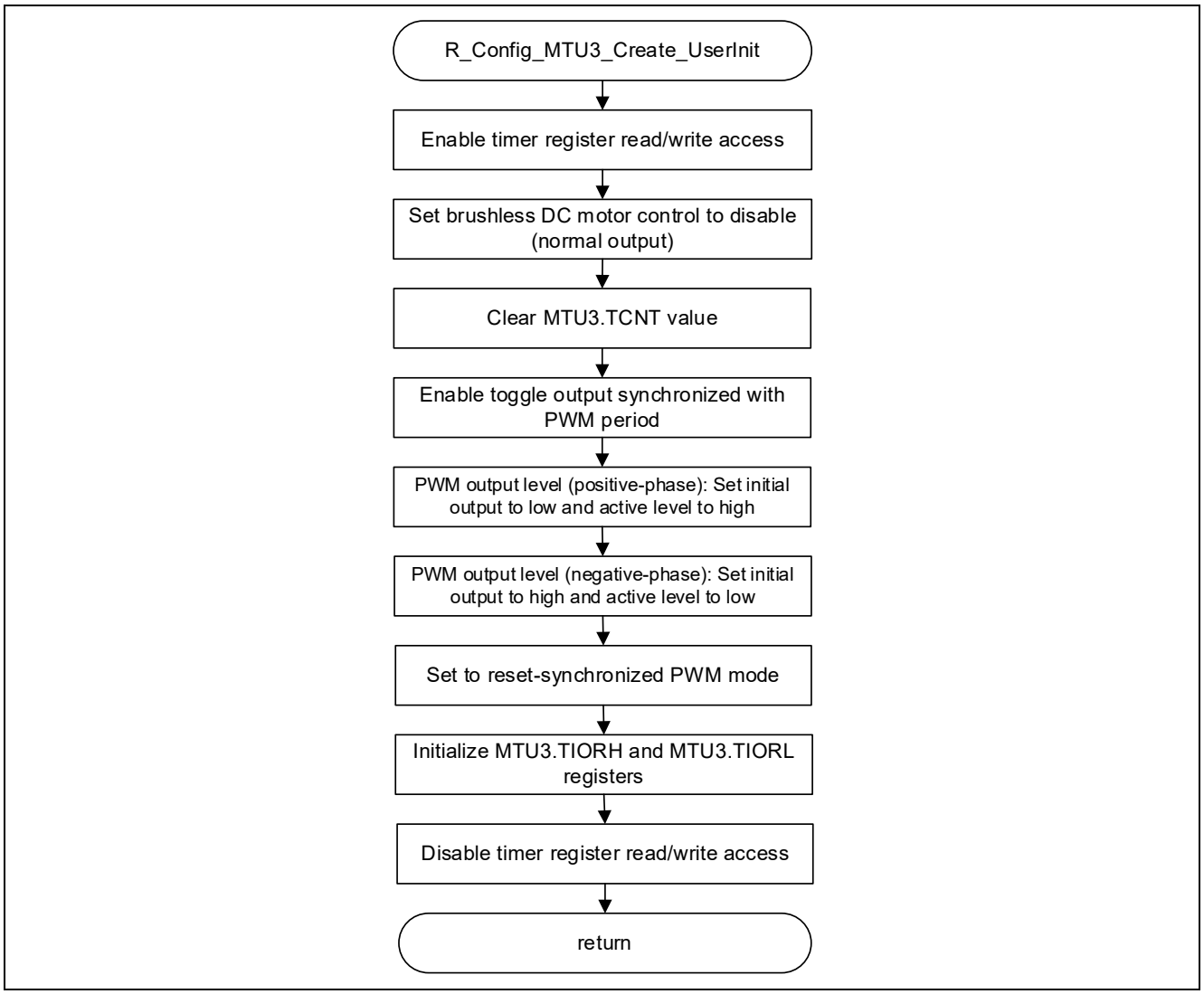


Figure 3.50 MTU3 User Initialization Function

The user initialization function R_Config_MTU4_Create_UserInit, which is called in the R_Config_MTU4_Create function executed before the main function, initializes the TIOR register that is not necessary in reset-synchronized PWM mode.

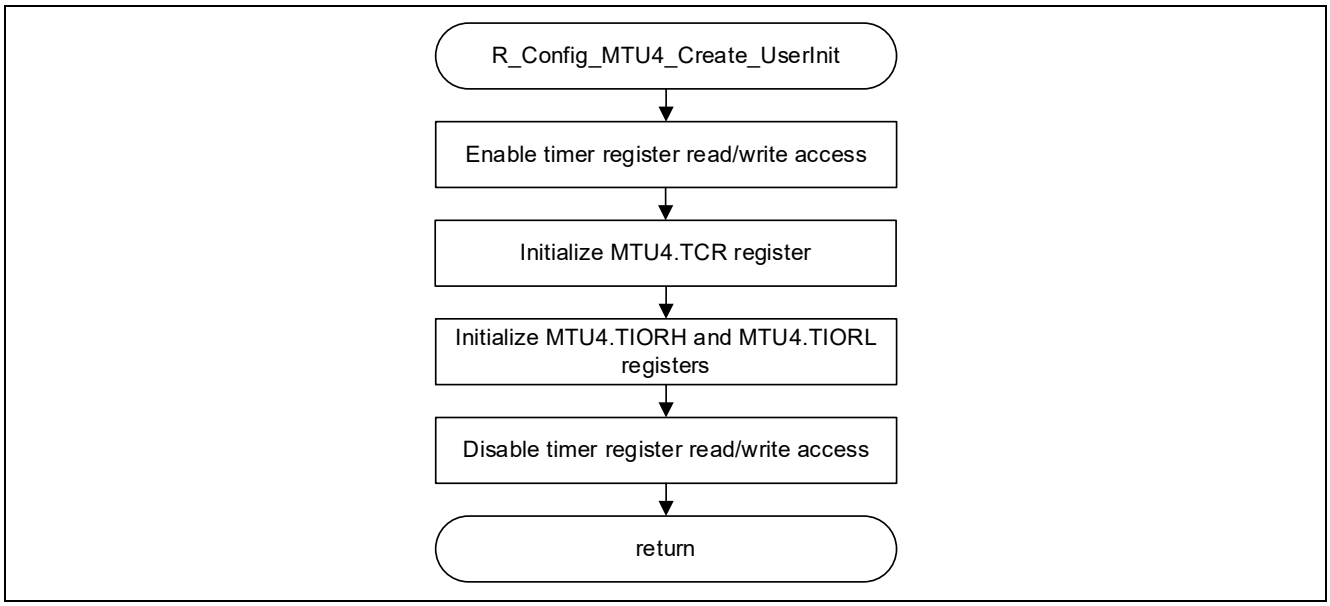


Figure 3.51 MTU4 User Initialization Function

3.8.5 Usage Notes

3.8.5.1 Count Start When Using 4 Channels

In this mode, six phases of positive and negative PWM waveforms (12 phases in total) that share a common wave transition point can be output by combining MTU3 and MTU4 with MTU6 and MTU7.

When the MTU3 and MTU6 counting starts using the R_Config_MTU3_Start and R_Config_MTU6_Start functions generated by the Smart Configurator, the counting start timing may not be the same because each of the functions are read. It is possible to start the MTU3 and MTU6 counting at the same time by setting the SCH3 and SCH6 bits of the timer counter synchronous start register TCSYSTR at the same time.

Refer to RX66T Group User's Manual: Hardware, section 22.2.19 Timer Counter Synchronous Start Register (TCSYSTR).

3.8.5.2 Components Used by Smart Configurator

The Smart Configurator does not have reset-synchronized PWM mode components.

In this sample code, the timer mode setting must be changed to reset-synchronized PWM mode in MTU3 user initialization function R_Config_MTU3_Create_UserInit for the code generated using the normal mode timer component. For details on the R_Config_MTU3_Create_UserInit function, refer to Figure 3.50.

Also, when using the complementary PWM mode component, the period register value is automatically fixed according to the timer operation period setting, and codes to implement up-counting and down-counting are generated because the counter clear source cannot be specified. As a result, the complementary PWM mode component cannot be used to create a code that generates reset-synchronized PWM waveforms.

3.9 Complementary PWM Mode with Double Buffer

- Target sample code file name: r01an5995_rx66t_mtu3_complementary_pwm_dblbuf.zip

3.9.1 Overview

Using the MTU's complementary PWM mode enables output of 3-phase complementary PWM waveforms with dead time.

This sample code describes a sample code that repeats the following output waveforms with dead time in complementary PWM mode 3 (transfer at crest and trough). Each duty cycle generates laterally asymmetric PWM waveforms using the double buffer.

- U-phase duty switching: 20% → 40% → 60% → 80% → 60% → 40% → ⋯
- V-phase duty switching: 40% → 60% → 80% → 60% → 40% → 20% → ⋯
- W-phase duty switching: 60% → 80% → 60% → 40% → 20% → 40%→ ⋯

The duty cycle is changed by transferring from buffer registers A and B to temporary registers A and B, and from temporary register A to the compare register, when a TCNT counter overflow occurs; and from temporary register B to the compare register when an underflow occurs.

The following list provides the MTU settings used in the sample code.

- Use complementary PWM mode 3 (transfer at crest and trough)
- Use channels 3 and 4
- Carrier period = 1ms
- Dead time= 30µs
- Timer count clock = 40MHz (PCLKC/4)
  Set MTU3.TGRA to MTU3.TCNT upper limit value
  (1/2 carrier period + dead time)
  — MTIOC3A pin toggle output setting
- Set buffer transfer timing
  — Transfers data at the crest and trough of the count
- Initial output value is high, active level is low
- Use MTU3.TGRB as U-phase duty register
  — Positive-phase:
  Low output at up-counting compare match
  High output at down-counting compare match
  — Negative-phase:
  High output at up-counting compare match
  Low output at down-counting compare match
- Use MTU4.TGRA as V-phase duty register
  — Positive-phase:
  Low output at up-counting compare match
  High output at down-counting compare match
  — Negative-phase:
  High output at up-counting compare match
  Low output at down-counting compare match
- Use MTU4.TGRB as W-phase duty register
  — Positive-phase:
  Low output at up-counting compare match
  High output at down-counting compare match
  — Negative-phase:
  High output at up-counting compare match
  Low output at down-counting compare match
- Use double buffer register
  — Use MTU3.TGRD and MTU3.TGRE as
  buffer registers of MTU3.TGRB
  — Use MTU4.TGRC and MTU4.TGRE as
  buffer registers of MTU4.TGRA
  — Use MTU4.TGRD and MTU4.TGRF as
  buffer registers of MTU4.TGRB
  — Refer to Figure 3.54 for details on initial value of buffer register
  for laterally asymmetric PWM output waveform
- Duty changes at each cycle
  — Change duty cycle at MTU3.TGRA compare match interrupt
  — Refer to Figure 3.54 for details on the timing for duty cycle changes

Set in Smart Configurator.
For setting methods,
refer to section 3.9.3.

Complementary PWM mode output for this sample code is shown below.

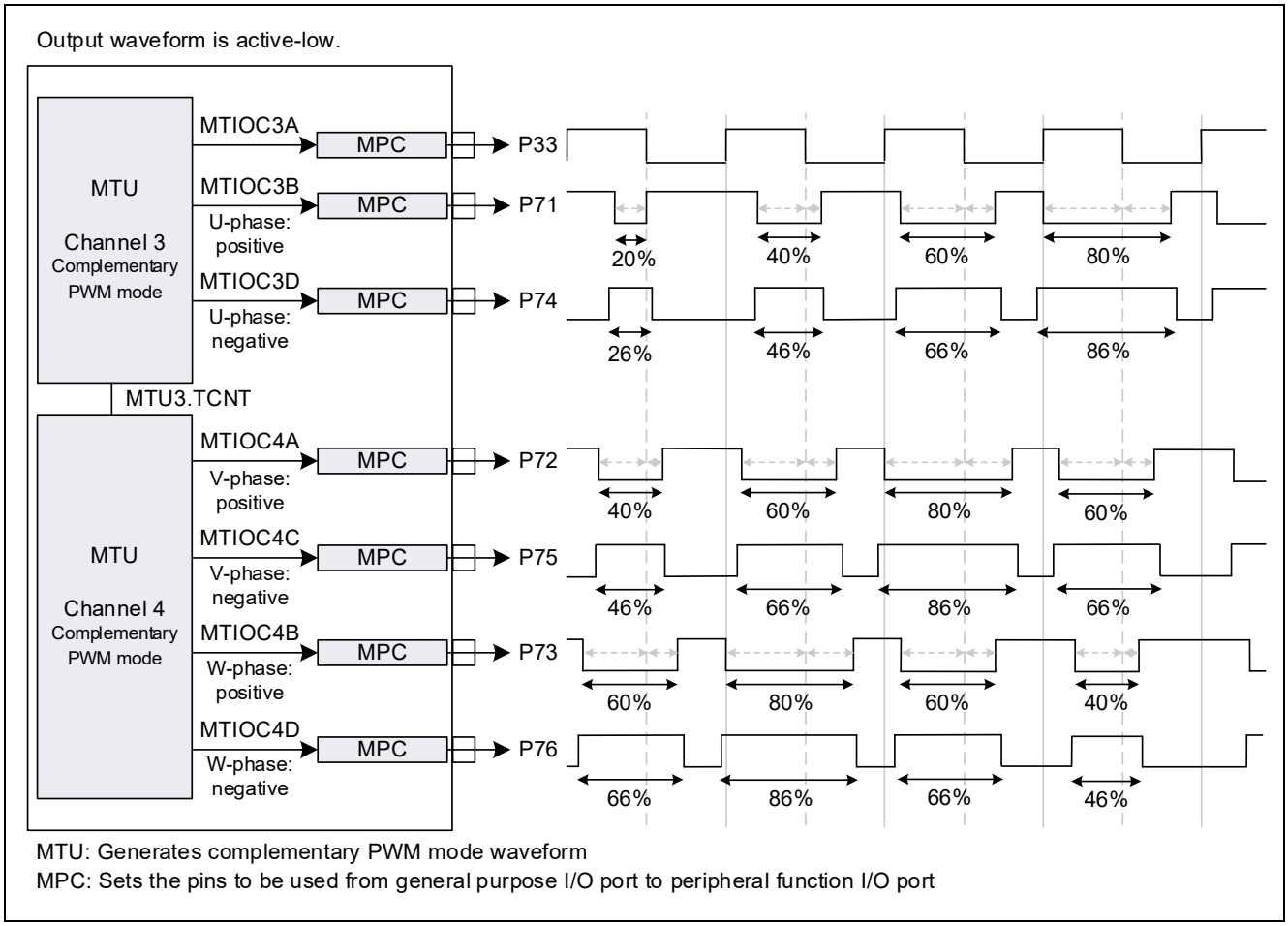


Figure 3.52 Complementary PWM Mode Output

3.9.2 Operation Details

The sample code operations are shown in Figure 3.54. The settings of the duty cycle are changed with each period by modifying the values of buffer register A (MTU3.TGRD, MTU4.TGRC and MTU4.TGRD) and buffer register B (MTU3.TGRE, MTU4.TGRE and MTU4.TGRF) at the compare match interrupt (TGIA3) of the MTU3.TRGA register, which is set to the upper limit value of MTU3.TCNT ((1) in Figure 3.54).

Writing MTU4.TGRD at the end of the buffer register modification enables data transfer from buffer registers A and B to temporary registers A and B. When MTU4.TGRD is written in the Ta interval, the data written to the buffer register is immediately transferred to the temporary register, but in this sample code, MTU4.TGRD is written in the Tb1 interval, so it is transferred to the temporary register after the Tb1 interval ends ((2) in Figure 3.54).

Since this sample code uses complementary PWM mode 3 for crest and trough transfers, the compare register updates the data by transferring from temporary register A at the end of the Tb1 interval, ((3) in Figure 3.54), and from temporary register B at the end of the Tb2 interval ((4) in Figure 3.54).

Initial output is OFF for both positive-phase output and negative-phase output according to the settings of TOCR2A bits OLS1P, OLS1N, OLS2P, OLS2N, OLS3P, and OLS3N. After setting the complementary PWM mode in MTU3.TMDR1, output continues until MTU4.TCNT is greater than the value of the TDDRA register ((5) in Figure 3.54).

After the above operations, the following steps 1 to 3 are repeated.

1. A compare match between the compare register and the counter register occurs in the Ta interval, the negative-phase output turns OFF, and then the positive-phase output turns ON. ((6) in Figure 3.54).
2. The compare register and temporary register A are enabled in the Tb1 interval. A compare match occurs in the U-phase, the positive-phase output turns OFF, and then the negative-phase output turns ON. As a compare match does not occur in the V-phase or W-phase, the waveform does not change. ((7) in Figure 3.54).
3. The compare register and temporary register B are enabled in period Tb2. As a compare match does not occur between the counter register and the compare register in the U-phase and V-phase, the waveform does not change. A compare match occurs in the W-phase, the negative-phase output turns OFF and the positive-phase output turns ON. ((8) in Figure 3.54).

- Laterally Asymmetric PWM Waveform Output
  The duty cycle in each period generates a different duty cycle for the up-counting period and down-counting periods.

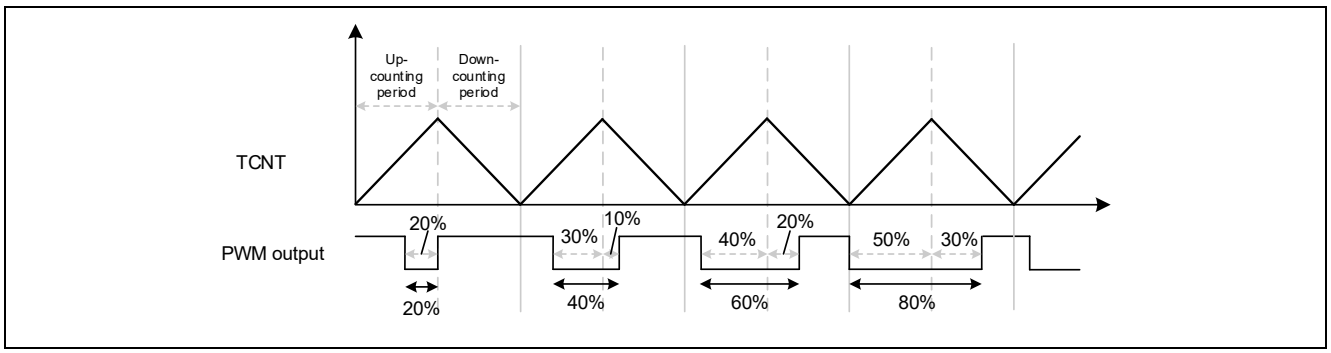
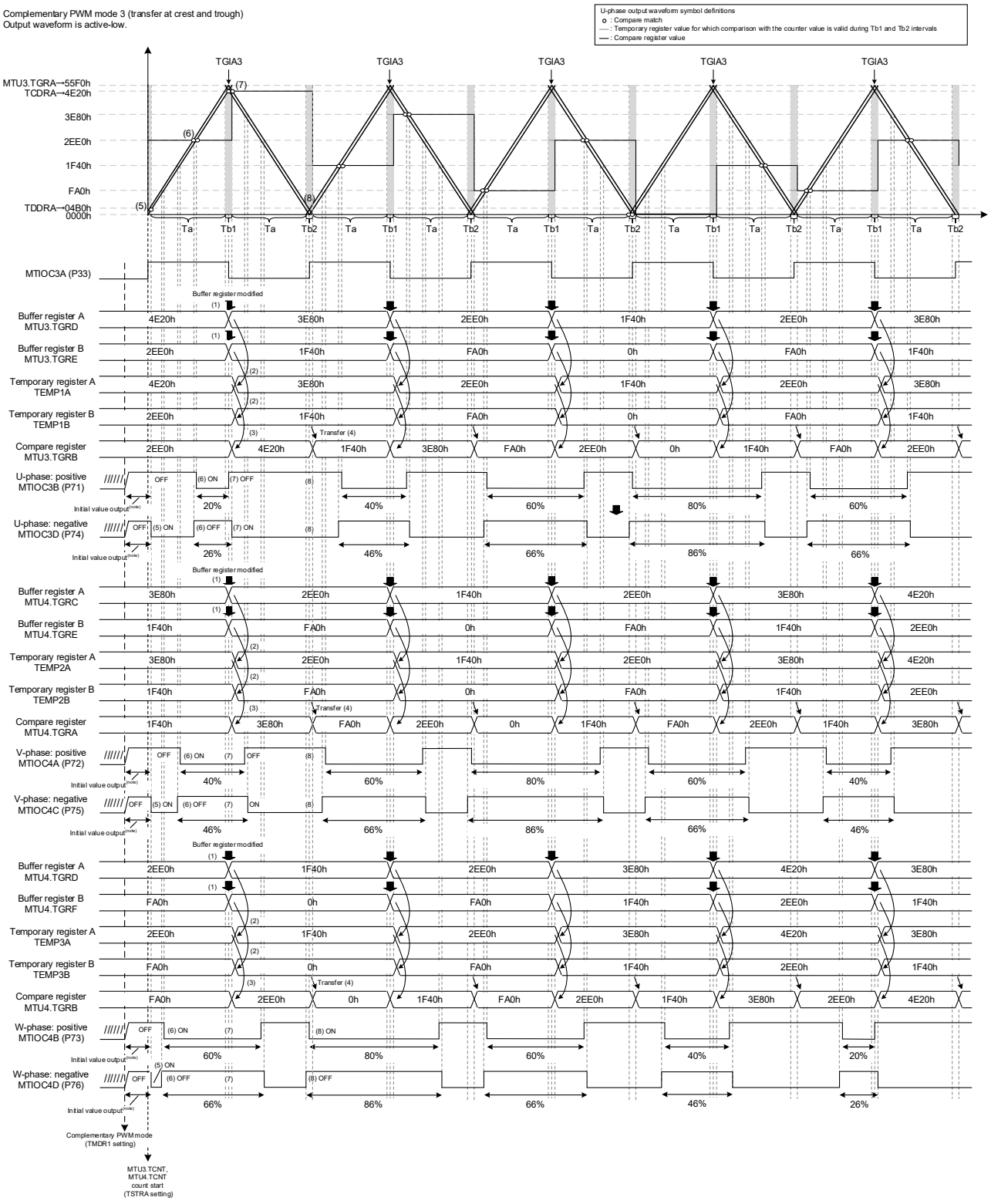


Figure 3.53 Laterally Asymmetric PWM Output Waveform



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.54 Sample Code Operations

3.9.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table 3.12 Adding Components

Item	Description
Component	Complementary PWM Mode Timer
Configuration name	Config_MTU3_MTU4
Operation	Complementary PWM Mode 3 (transfer at crest and trough)
Resource	MTU3_MTU4

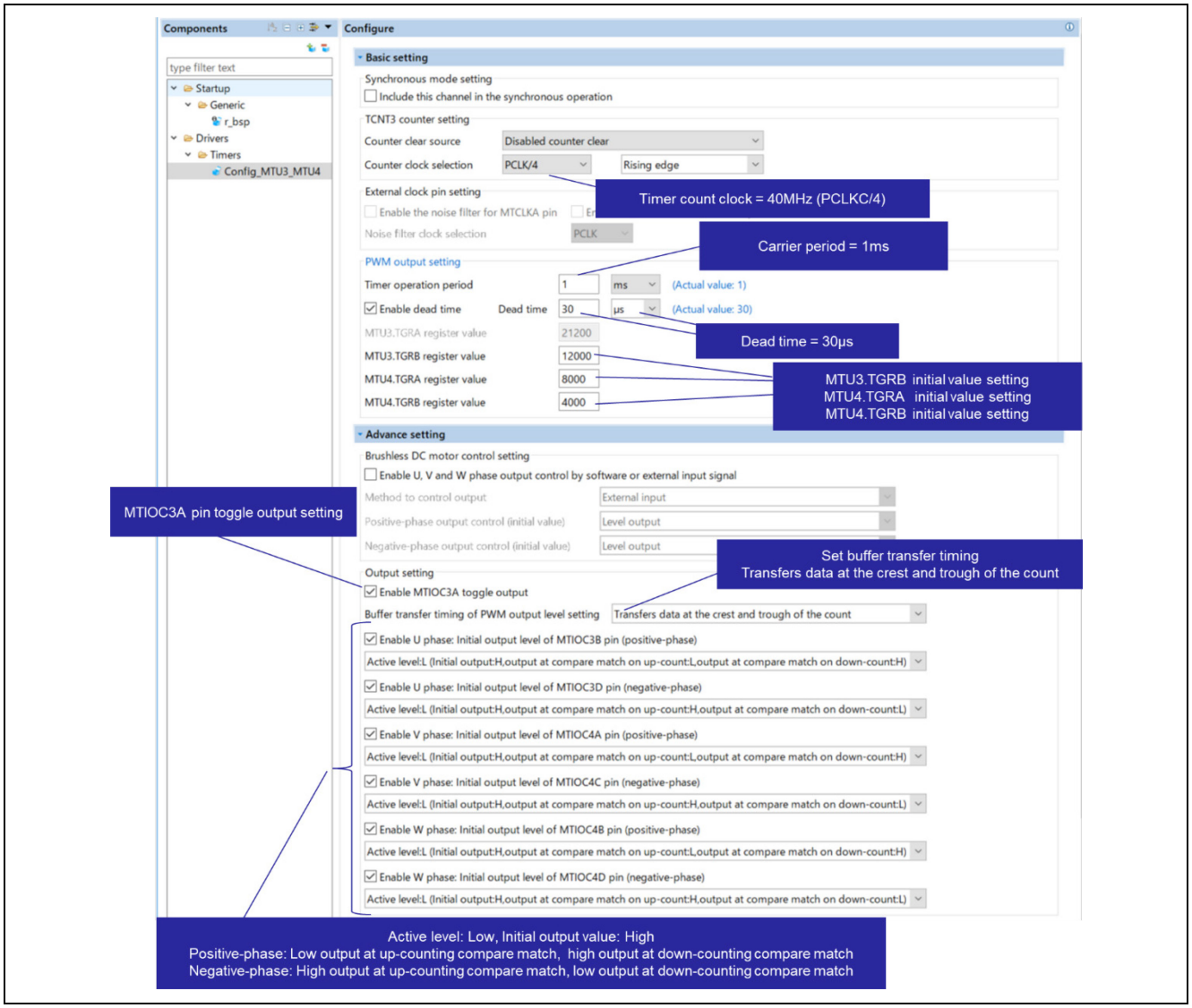


Figure 3.55 MTU3 and MTU4 Settings (1/2)

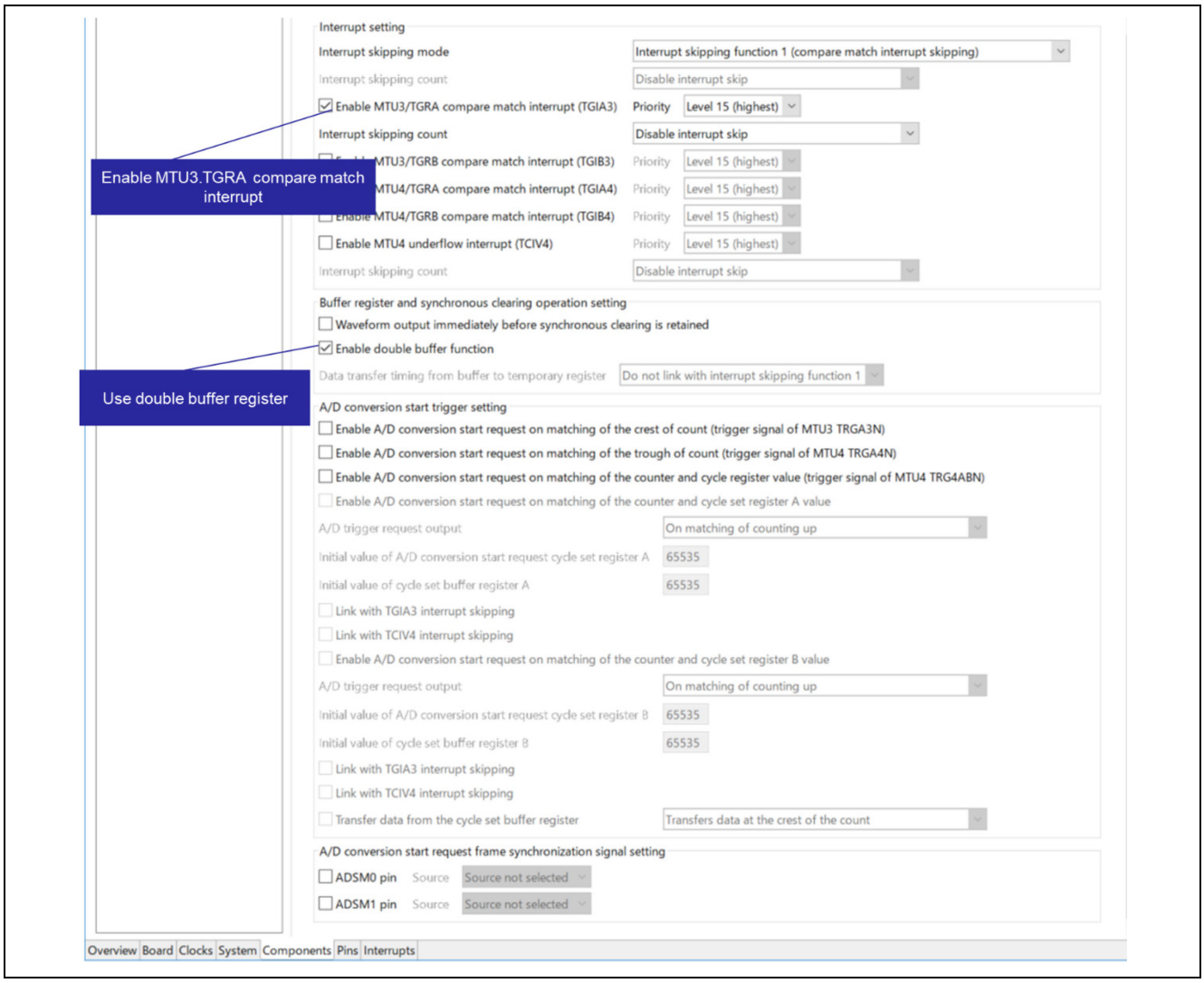


Figure 3.56 MTU3 and MTU4 Settings (2/2)

3.9.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

MTU3.TCNT and MTU4.TCNT counting is started in the main function.

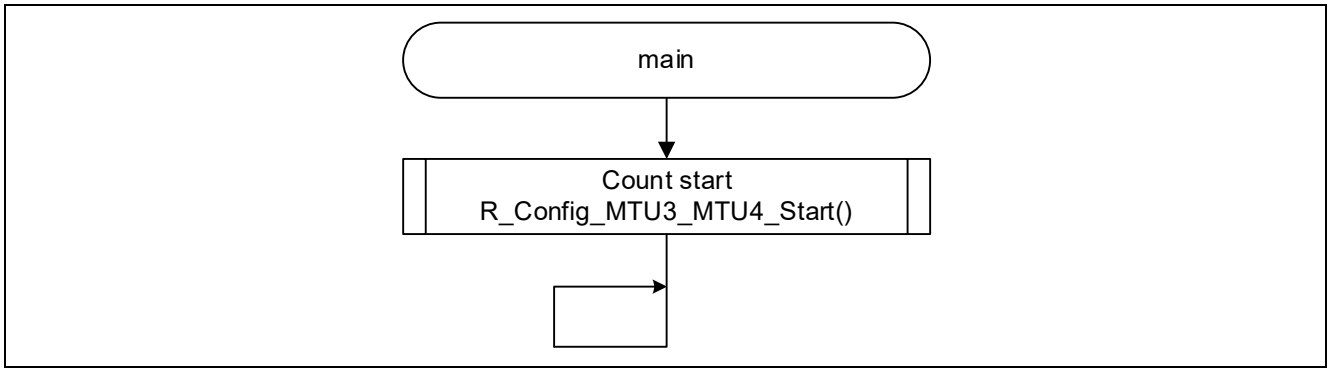


Figure 3.57 main Function

The user initialization function R_Config_MTU3_MTU4_Create_UserInit, which is executed before the main function, sets the initial value of the buffer register and initializes the variables. This function is called from within the R_Config_MTU3_MTU4_Create function.

This function initializes the following variables used in this sample code.

- s_u_ucduty_prv: variable for retaining the value of the previous MTU3.TGRD register
- s_v_ucduty_prv: variable for retaining the value of the previous MTU4.TGRC register
- s_w_ucduty_prv: variable for retaining the value of the previous MTU4.TGRD register

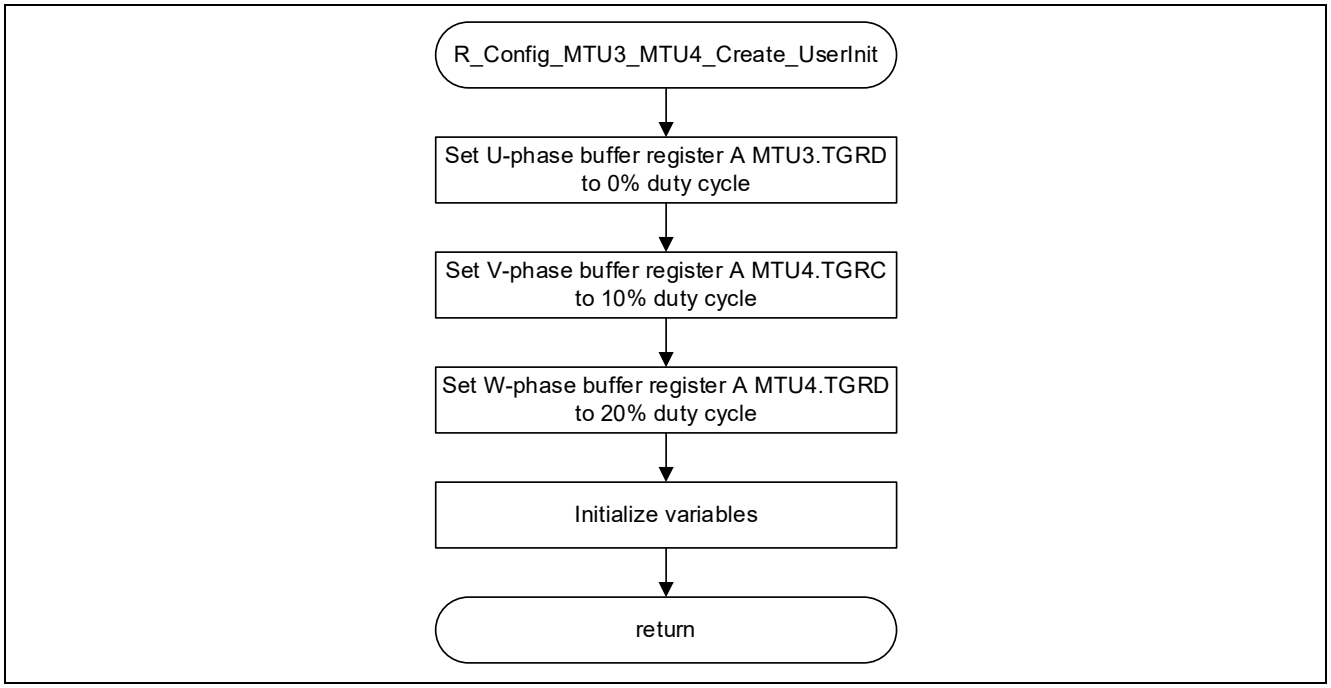


Figure 3.58 User Initialization Function

The TGIA3 interrupt handler function changes the values of buffer register A and buffer register B (MTU3.TGRE, MTU4.TGRE and MTU4.TGRF) according to the current values of buffer register A (MTU3.TGRD, MTU4.TGRC and MTU4.TGRD) and the values set in the previous buffer register A.

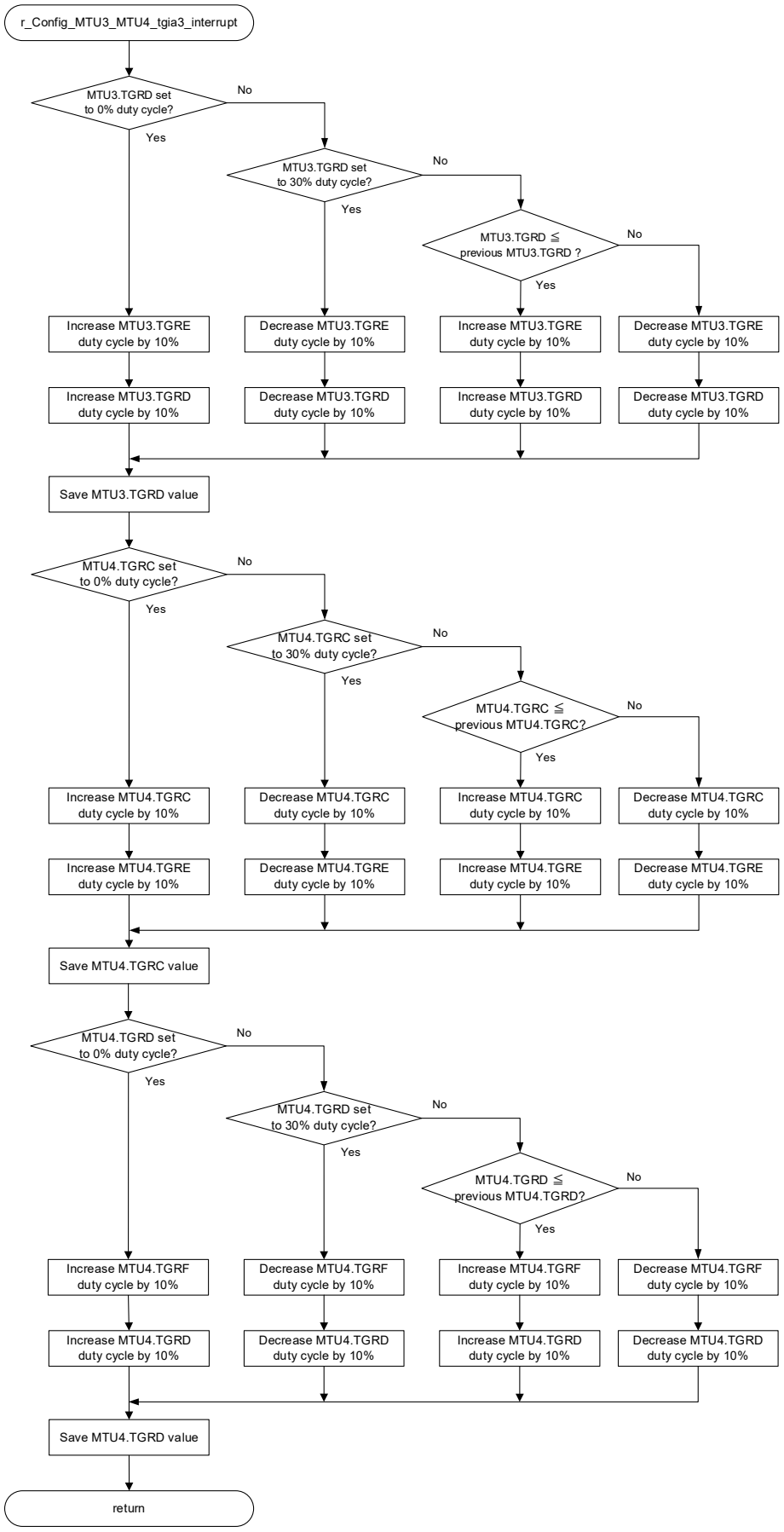


Figure 3.59 TGIA3 Interrupt Handler Function

3.9.5 Usage Notes

3.9.5.1 Pin Settings

The MTIOC3C and MTIOC6C pins cannot be used as the timer input/output pin in the complementary PWM mode. Set them to I/O ports.

For details, refer to RX66T Group User's Manual: Hardware, section 22.3.8 Complementary PWM Mode, Table 22.74 Output Pins for Complementary PWM Mode.

3.9.5.2 Buffer Register Value Updating

When modifying the data in the buffer register, be sure to modify to MTU4.TGRD (MTU7.TGRD) last. Data is transferred from the buffer register to all five temporary registers simultaneously after the write to MTU4.TGRD (MTU7.TGRD).

Even when not updating all five registers or when not updating the data in MTU4.TGRD (MTU7.TGRD), be sure to write to MTU4.TGRD (MTU7.TGRD) after writing data to the registers to be updated. In this case, the data to be written to MTU4.TGRD (MTU7.TGRD) should be the same as the data prior to the write operation.

3.9.5.3 Buffer Operation Settings

In complementary PWM mode, use the buffer operation to modify the PWM period setting registers (MTU3.TGRA and MTU6.TGRA), timer period data registers (TCDRA and TCDRB), and duty setting registers (MTU3.TGRB, MTU4.TGRA, MTU4.TGRB, MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB).

The MTIOC4C (MTIOC7C) and MTIOC4D (MTIOC7D) pins cannot output waveforms when buffer operation bits MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) and MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) are set to 1. Set MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) and MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) bits to 0.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.14 Buffer Operation Setting in Complementary PWM Mode.

3.9.5.4 Output Level Settings

When MTU3 and MTU4 (or MTU6 and MTU7) are in the complementary PWM mode, PWM waveform output level is determined by the TOCR1A.OLSP, TOCR1A.OLSN, TOCR1B.OLSP, and TOCR1B.OLSN bits, and the TOCR2A.OLSnP, TOCR2A.OLSnN, TOCR2B.OLSnP, and TOCR2B.OLSnN (n = 0 to 3) bits. Set the TIOR register to 00h.

If the TDERA.TDER (TDERB.TDER) bit is set to 0 (dead time is not generated) in the complementary PWM mode, the negative-phase output is the inverted level of the positive-phase output according to the settings of the TOCR1A.OLSP (TOCR1B.OLSP) and TOCR2A.OLSnP (TOCR2B.OLSnP) (n = 0 to 3) bits, and does not depend on the settings of the TOCR1A.OLSN (TOCR1B.OLSN) and TOCR2A.OLSnN (TOCR2B.OLSnN) (n = 0 to 3) bits.

When dead time is not generated, if only the output of the negative-phase side is enabled and the output of the positive-phase side is prohibited in the TOER register, the negative-phase side does not be output.

For details, refer to RX66T Group User's Manual: Hardware, section 22.2.22 Timer Output Control Registers 1 (TOCR1A, TOCR1B) and section 22.2.23 Timer Output Control Registers 2 (TOCR2A, TOCR2B).

3.9.5.5 TGR Register Initial Values

The values of buffer register A and buffer register B in the code generated by the Smart Configurator are set to the same value as the compare register. Set the initial values of buffer registers A and B by adding them to the code in user initialization function R_Config_MTU3_MTU4_Create_UserInit.

3.9.5.6 Laterally Asymmetric PWM Output Methods

When the value of buffer register A is set to the value of buffer register B, the PWM output is symmetrical. To output laterally asymmetric PWM waveforms, set buffer register B to a value different from that of buffer register A.

For details, refer to RX66T Group User's Manual: Hardware, section 22.3.8 Complementary PWM Mode, in (2) Outline of Complementary PWM Mode Operation, paragraph (s) Double Buffer Function in Complementary PWM Mode.

3.10 Complementary PWM Mode Without Double Buffer

- Target sample code file name: r01an5995_rx66t_mtu3_complementary_pwm.zip

3.10.1 Overview

Using the MTU's complementary PWM mode enables output of 3-phase complementary PWM waveforms with dead time.

This sample code describes a sample code that repeats the following output waveforms with dead time in complementary PWM mode 3 (transfer at crest and trough). Each duty cycle generates symmetric PWM waveforms using the buffer (not a double buffer).

- U-phase duty switching: 20% → 40% → 60% → 80% → 60% → 40% → ⋯
- V-phase duty switching: 40% → 60% → 80% → 60% → 40% → 20% → ⋯
- W-phase duty switching: 60% → 80% → 60% → 40% → 20% → 40%→ ⋯

The duty cycle is changed by transferring from the buffer register to the temporary register, and from the temporary register to the compare register, when a TCNT counter overflow occurs; and from the temporary register to the compare register when an underflow occurs.

The following list provides the MTU settings used in the sample code.

- Use complementary PWM mode 3 (transfer at crest and trough)
- Use channels 3 and 4
- Carrier period = 1ms
- Dead time = 30µs
- Timer count clock = 40MHz (PCLKC/4)
- Set MTU3.TGRA to MTU3.TCNT upper limit value
  (1/2 carrier period + dead time)
  — MTIOC3A pin toggle output setting
- Set buffer transfer timing
  — Transfers at the crest and trough of the count
- Initial output value is high, active level is low
- Use MTU3.TGRB as U-phase duty register
  — Positive-phase:
  Low output at up-counting compare match
  High output at down-counting compare match
  — Negative-phase:
  High output at up-counting compare match
  Low output at down-counting compare match
- Use MTU4.TGRA as V-phase duty register
  — Positive-phase:
  Low output at up-counting compare match
  High output at down-counting compare match
  — Negative-phase:
  High output at up-counting compare match
  Low output at down-counting compare match
- Use MTU4.TGRB as W-phase duty register
  — Positive-phase:
  Low output at up-counting compare match
  High output at down-counting compare match
  — Negative-phase:
  High output at up-counting compare match
  Low output at down-counting compare match
- Use buffer register
  — Use MTU3.TGRD as buffer register of MTU3.TGRB
  — Use MTU4.TGRC as buffer register of MTU4.TGRA
  — Use MTU4.TGRD as buffer register of MTU4.TGRB
  — Refer to Figure 3.62 details on initial value of buffer register
- Duty changes at each cycle
  — Change duty cycle at MTU3.TGRA compare match interrupt
  — Refer to Figure 3.62 for details on the timing for duty cycle changes

Set in Smart Configurator.
For setting methods,
refer to section 3.10.3.

Complementary PWM mode output for this sample code is shown below.

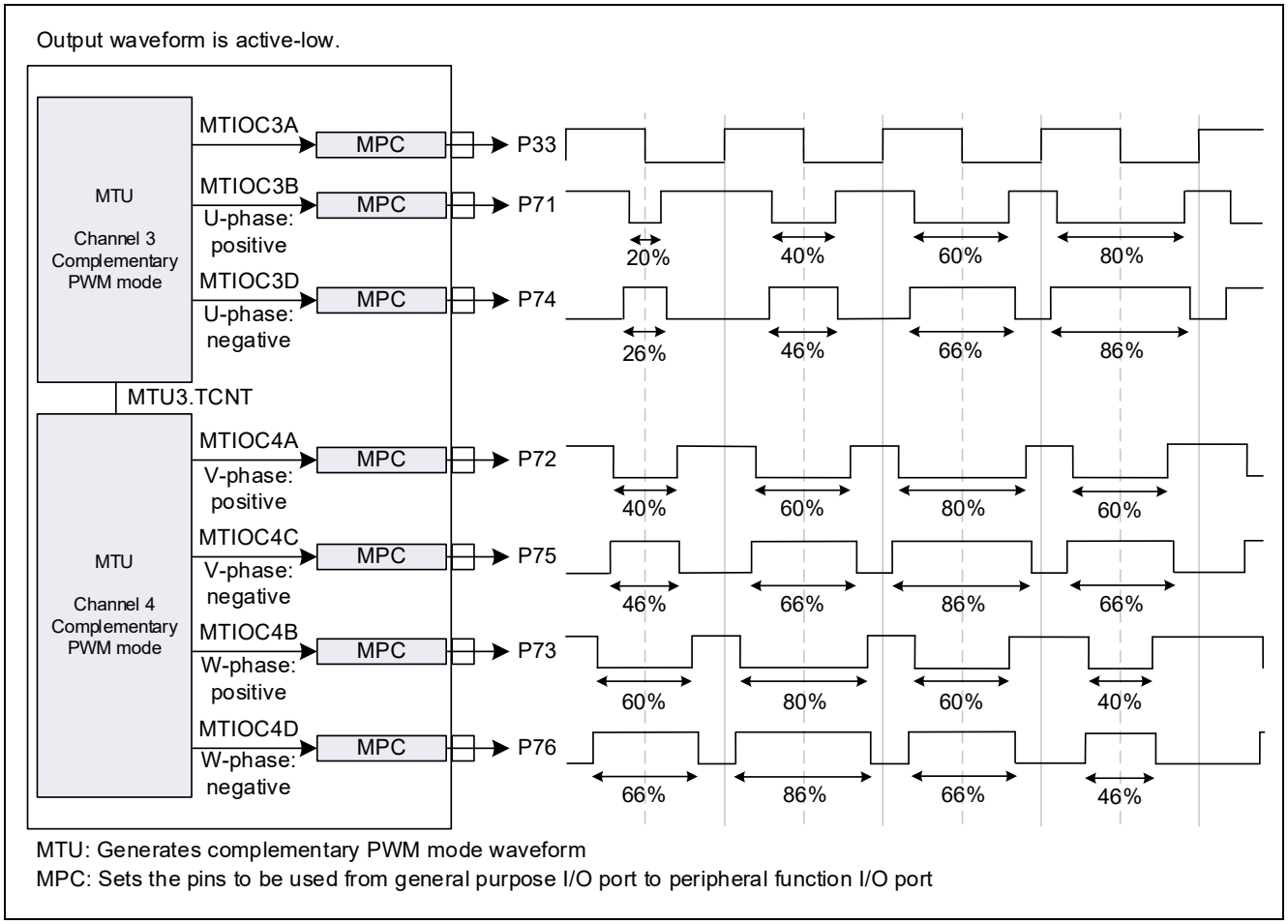


Figure 3.60 Complementary PWM Mode Output

3.10.2 Operation Details

The sample code operations are shown in Figure 3.62. The settings of the duty cycle are changed with each period by modifying the values of the buffer register (MTU3.TGRD, MTU4.TGRC and MTU4.TGRD) at the compare match interrupt (TGIA3) of the MTU3.TGRA register, which is set to the upper limit value of MTU3.TCNT ((1) in Figure 3.62).

Writing MTU4.TGRD at the end of the buffer register modification enables data transfer from the buffer register to the temporary register. When MTU4.TGRD is written in the Ta interval, the data written to the buffer register is immediately transferred to the temporary register, but in this sample code, MTU4.TGRD is written in the Tb1 interval, so the data is transferred to the temporary register after the Tb1 interval ends ((2) in Figure 3.62).

Since this sample code uses complementary PWM mode 3 for crest and trough transfers, the compare register updates the data by transferring from the temporary register at the end of the Tb1 interval ((3) in Figure 3.62) and at the end of the Tb2 interval ((4) in Figure 3.62).

Initial output is OFF for both positive-phase output and negative-phase output according to the settings of TOCR2A bits OLS1P, OLS1N, OLS2P, OLS2N, OLS3P, and OLS3N. After setting the complementary PWM mode in MTU3.TMDR1, output continues until MTU4.TCNT is greater than the value of the TDDRA register ((5) in Figure 3.62).

After the above operations, the following steps 1 to 3 are repeated.

1. A compare match between the compare register and the counter register occurs in the Ta interval, the negative-phase output turns OFF, and then the positive-phase output turns ON. ((6) in Figure 3.62).
2. The compare register and temporary register are enabled in the Tb1 interval, but because a compare match with the counter register does not occur, the waveform does not change. ((7) in Figure 3.62).
3. The compare register and temporary register are enabled in the Tb2 interval, but because a compare match with the counter register does not occur, the waveform does not change. ((8) in Figure 3.62).

- Symmetrical PWM Output Waveform
  The duty cycle in each period generates the 1/2 duty cycle for the up-counting period and down-counting periods.

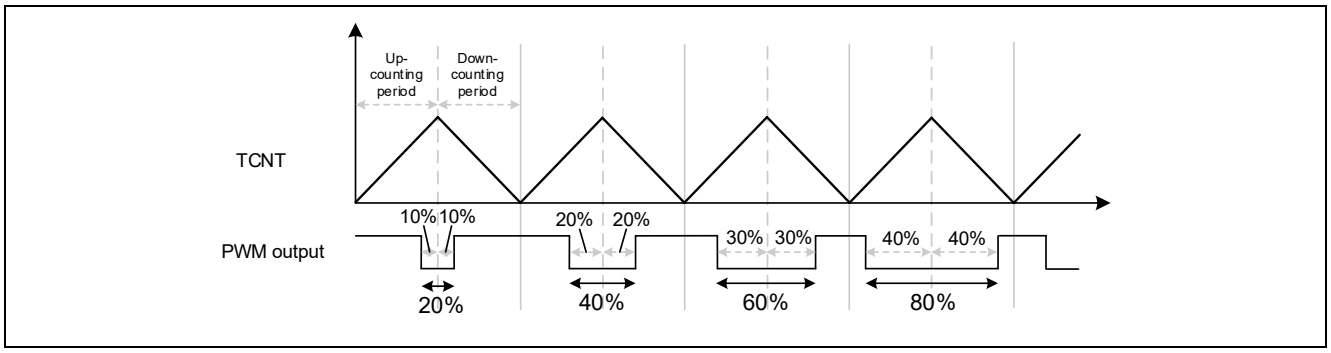
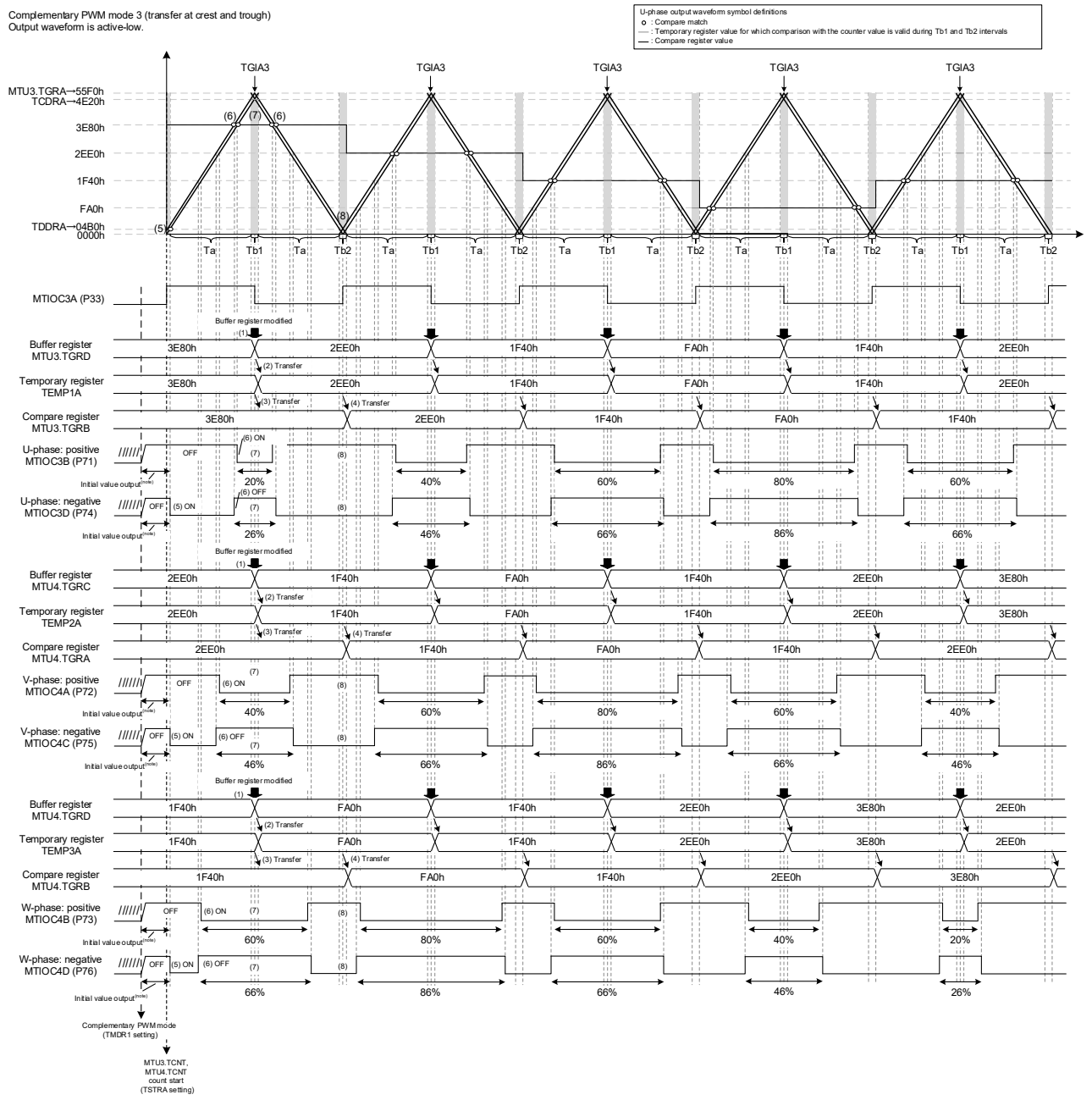


Figure 3.61 Symmetrical PWM Output Waveform



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.62 Sample Code Operations

3.10.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table3.13 Adding Components

Item	Description
Component	Complementary PWM Mode Timer
Configuration name	Config_MTU3_MTU4
Operation	Complementary PWM Mode 3 (transfer at crest and trough)
Resource	MTU3_MTU4

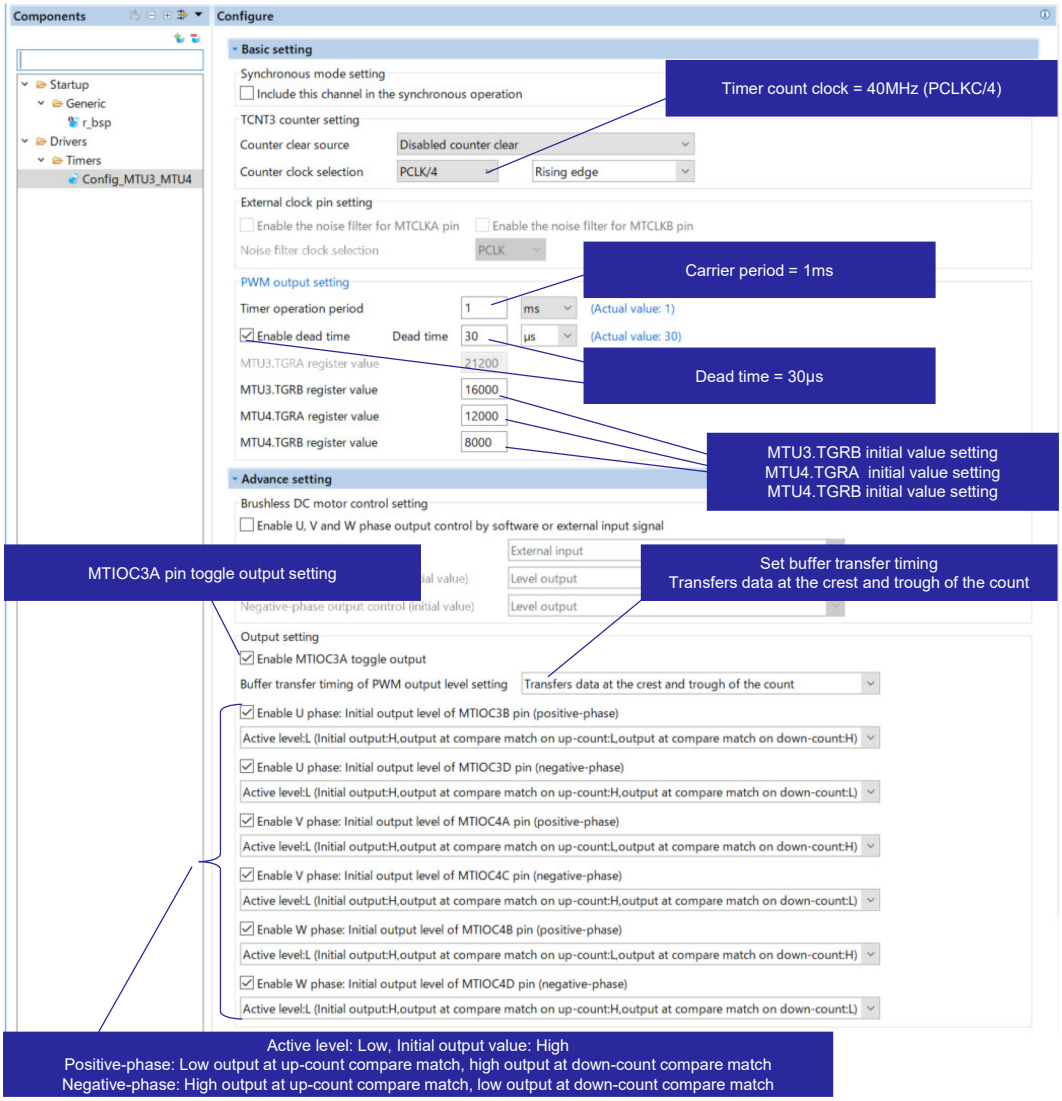


Figure 3.63 MTU3 and MTU4 Settings (1/2)

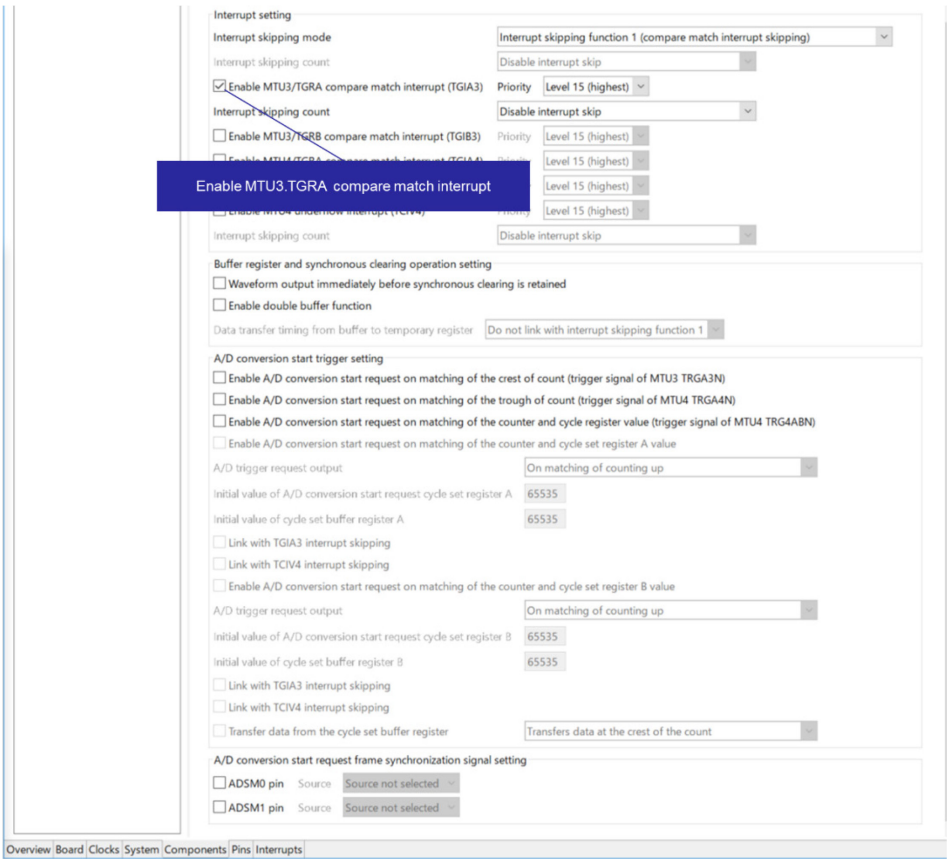


Figure 3.64 MTU3 and MTU4 Settings (2/2)

3.10.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

MTU3.TCNT and MTU4.TCNT counting is started in the main function.

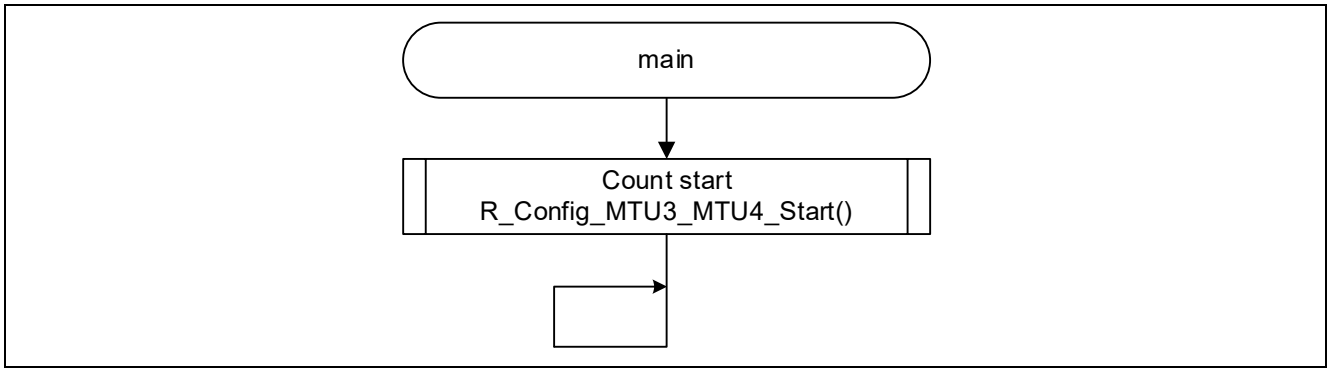


Figure 3.65 main Function

The user initialization function R_Config_MTU3_MTU4_Create_UserInit, which is executed before the main function, initializes the variables. This function is called from within the R_Config_MTU3_MTU4_Create function.

This function initializes the following variables used in this sample code.

- s_u_ucduty_prv: variable for retaining the value of the previous MTU3.TGRD register
- s_v_ucduty_prv: variable for retaining the value of the previous MTU4.TGRC register
- s_w_ucduty_prv: variable for retaining the value of the previous MTU4.TGRD register

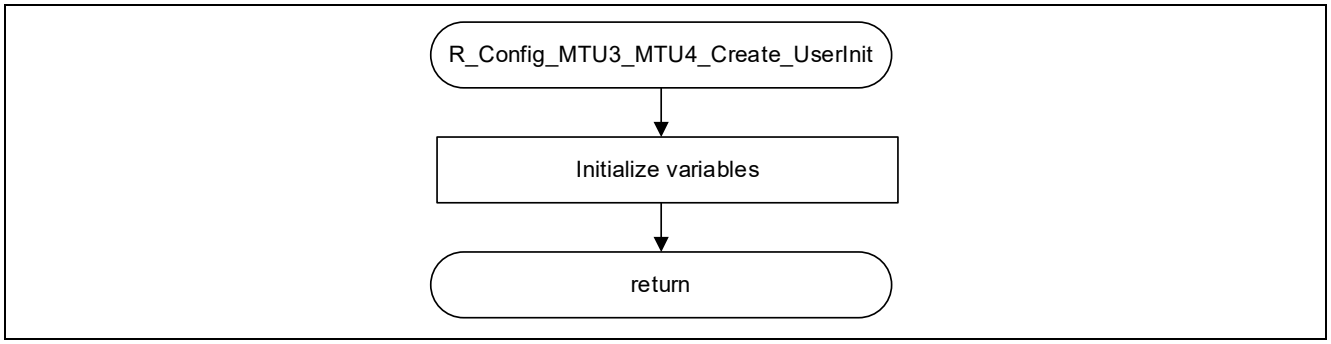


Figure 3.66 User Initialization Function

The TGIA3 interrupt handler function changes the values of the buffer registers according to the current value of the buffer registers (MTU3.TGRD, MTU4.TGRC and MTU4.TGRD) and the values set in the previous buffer registers.

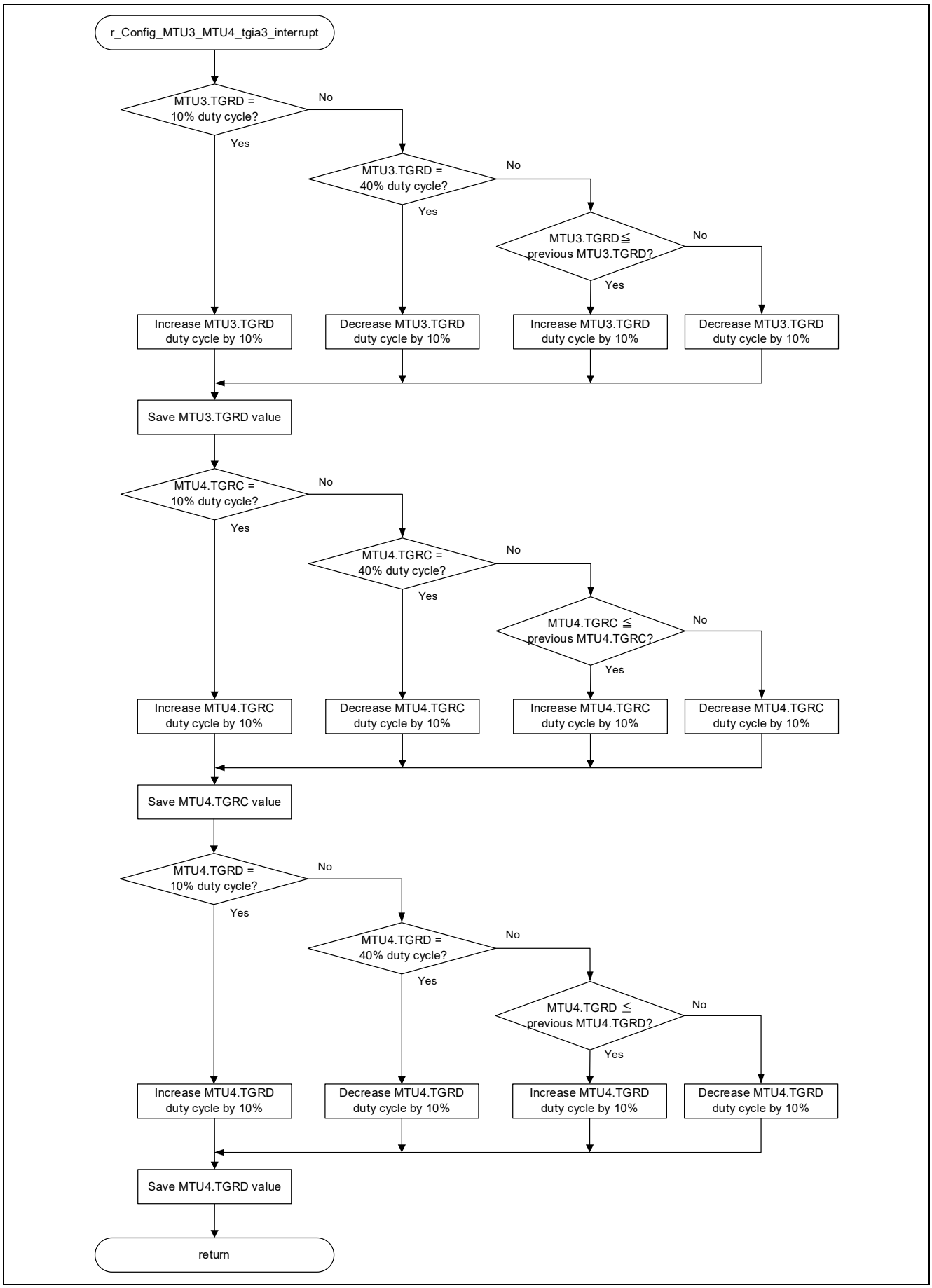


Figure 3.67 TGIA3 Interrupt Handler Function

3.10.5 Usage Notes

3.10.5.1 Pin Settings

The MTIOC3C and MTIOC6C pins cannot be used as the timer input/output pin in the complementary PWM mode. Set them to I/O ports.

For details, refer to RX66T Group User's Manual: Hardware, section 22.3.8 Complementary PWM Mode, Table 22.74 Output Pins for Complementary PWM Mode.

3.10.5.2 Buffer Register Value Updating

When modifying the data in the buffer register, be sure to modify to MTU4.TGRD (MTU7.TGRD) last. Data is transferred from the buffer register to all five temporary registers simultaneously after the write to MTU4.TGRD (MTU7.TGRD).

Even when not updating all five registers or when not updating the data in MTU4.TGRD (MTU7.TGRD), be sure to write to MTU4.TGRD (MTU7.TGRD) after writing data to the registers to be updated. In this case, the data to be written to MTU4.TGRD (MTU7.TGRD) should be the same as the data prior to the write operation.

3.10.5.3 Buffer Operation Settings

In complementary PWM mode, use the buffer operation to modify the PWM period setting registers (MTU3.TGRA and MTU6.TGRA), timer period data registers (TCDRA and TCDRB), and duty setting registers (MTU3.TGRB, MTU4.TGRA, MTU4.TGRB, MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB).

The MTIOC4C (MTIOC7C) and MTIOC4D (MTIOC7D) pins cannot output waveforms when buffer operation bits MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) and MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) are set to 1. Set MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) and MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) bits to 0.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.14 Buffer Operation Setting in Complementary PWM Mode.

3.10.5.4 Output Level Settings

When MTU3 and MTU4 (or MTU6 and MTU7) are in the complementary PWM mode, PWM waveform output level is determined by the TOCR1A.OLSP, TOCR1A.OLSN, TOCR1B.OLSP, and TOCR1B.OLSN bits, and the TOCR2A.OLSnP, TOCR2A.OLSnN, TOCR2B.OLSnP, and TOCR2B.OLSnN (n = 0 to 3) bits. Set the TIOR register to 00h.

If the TDERA.TDER (TDERB.TDER) bit is set to 0 (dead time is not generated) in the complementary PWM mode, the negative-phase output is the inverted level of the positive-phase output according to the settings of the TOCR1A.OLSP (TOCR1B.OLSP) and TOCR2A.OLSnP (TOCR2B.OLSnP) (n = 0 to 3) bits, and does not depend on the settings of the TOCR1A.OLSN (TOCR1B.OLSN) and TOCR2A.OLSnN (TOCR2B.OLSnN) (n = 0 to 3) bits.

When dead time is not generated, if only the output of the negative-phase side is enabled and the output of the positive-phase side is prohibited in the TOER register, the negative-phase side does not be output.

For details, refer to RX66T Group User's Manual: Hardware, section 22.2.22 Timer Output Control Registers 1 (TOCR1A, TOCR1B) and section 22.2.23 Timer Output Control Registers 2 (TOCR2A, TOCR2B).

3.11 PWM Mode 1 Duty Cycles 0% to 100% (modify compare register at compare match)

- Target sample code file name: 01an5995_rx66t_mtu3_pwm1_50to100.zip

3.11.1 Overview

The MTU PWM mode can be used to output PWM waveforms with duty cycles 0% to 100% according to the TGR register setting.

This sample code describes a sample code that uses PWM mode 1 and repeats the following waveform output, including duty cycles 0% and 100%.

- Duty switching: 50% → 80% → 100% → 80% → 50% → 0% → ・・・

The basic operation is to make changes to the duty cycle by transferring the TGRC value to the TGRA when a duty register TGRA compare match occurs using buffer register TGRC. When switching from 0% to 50%, the process to directly modify the TGRA register is performed when a period register TGRB compare match occurs.

The following list provides the MTU settings used in the sample code.

- Use PWM mode 1
- Use channel 0
- Initial output value = low
- Carrier period = 1ms
- Timer count clock = 40MHz (PCLKC/4)
- Use TGRB as period register
  - Timer counter clear source = TGRB compare match
  - Low output at TGRB compare match
- Use TGRA as duty register
  - High output at TGRA compare match
- Use buffer register
  - Use TGRC as buffer register of TGRA
  - Buffer transfer when TGRA compare match occurs
- Duty changes at each cycle
  - Change duty cycle at TGRA compare match interrupt
  - Refer to Figure 3.69 for details on the timing for duty cycle changes

Set in Smart Configurator.
For Setting Methods,
refer to section 3.11.3.

PWM mode 1 output for this sample code is shown below.

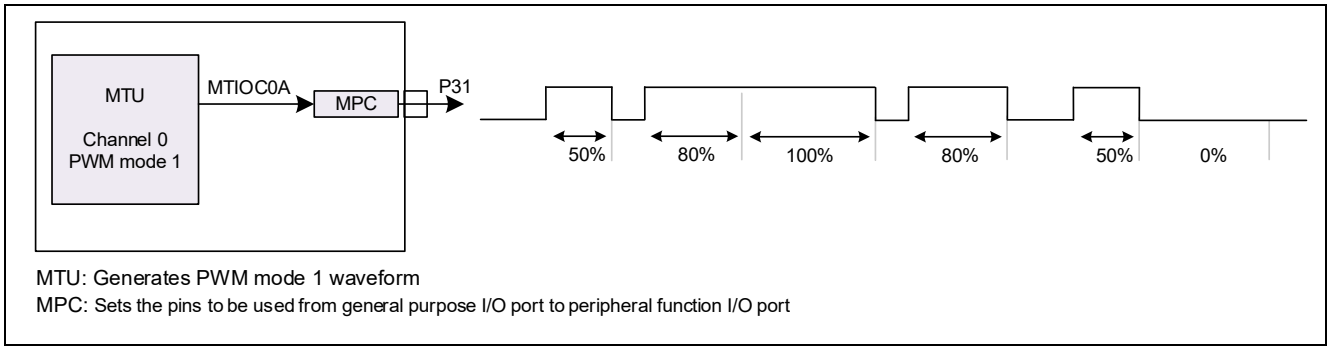


Figure 3.68 PWM Mode 1 Output

3.11.2 Operation Details

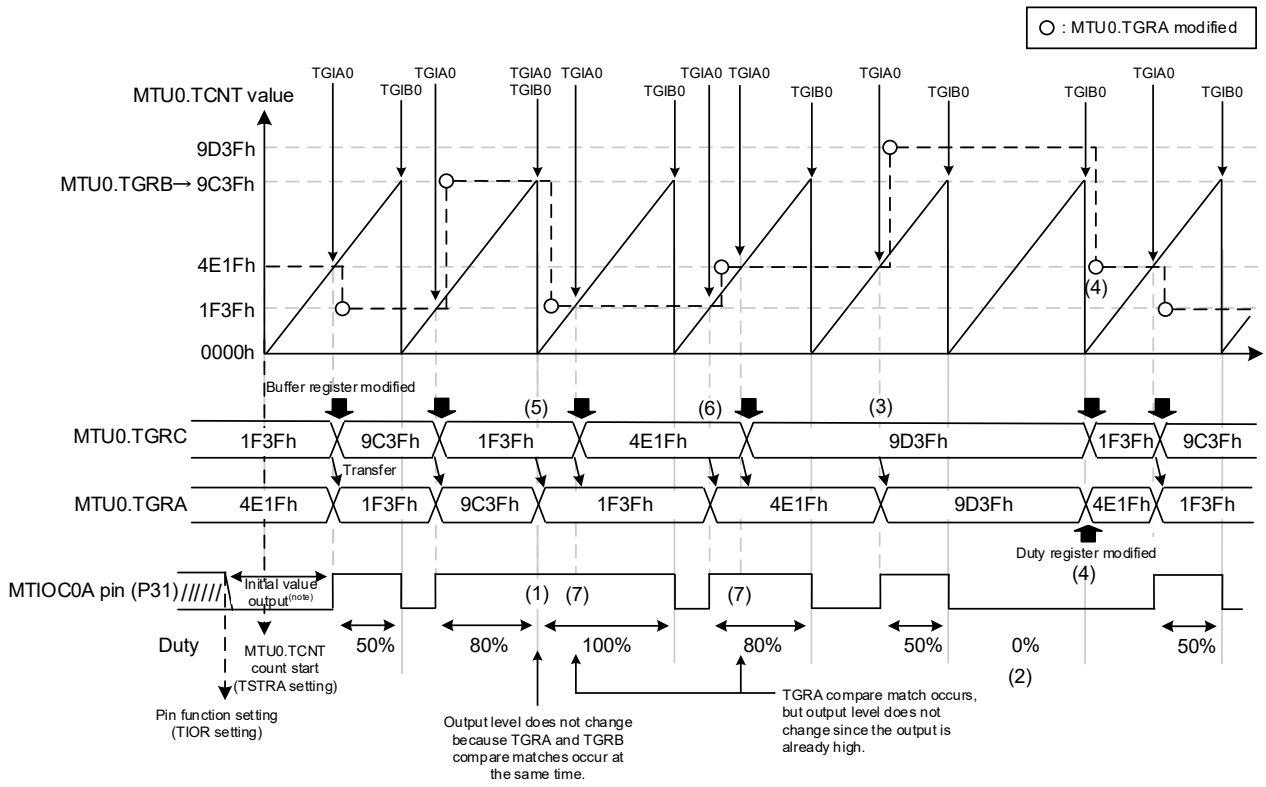
The sample code operations are shown in Figure 3.69. The basic operation is to make changes to the settings of the duty cycle by modifying the buffer register TGRC value at the compare match interrupt (TGIA0) of duty register TGRA and transferring the value of TGRC to duty register TGRA when a TGRA compare match occurs.

- 100% Duty Cycle Output
  When outputting 100% duty cycle, set the TGRA to the same value as the TGRB.
  When the TGRA value is the same as the TGRB value, both compare matches occur at the same time. Because output does not change when two compare matches occur at the same time, high is retained and 100% duty cycle is output ((1) in Figure 3.69).

- 0% Duty Cycle Output
  When outputting 0% duty cycle, set the TGRA to a value greater than the TGRB.
  When the TGRA value is greater than the TGRB, low is retained and 0% duty cycle is output because the TGRA compare match does not occur ((2) in Figure 3.69).
  To continue low for one cycle period and output 0% duty cycle, the TGRC value is not modified by the TGIA0 interrupt in which a value greater than the TGRB is transferred to the TGRA. The timing for modifying the TGRC value when TGIA0 occurs needs to be adjusted ((3) in Figure 3.69).

- Switching from 0% duty cycle
  When 0% duty cycle is output, a TGRA compare match does not occur because the TGRA is greater than the TGRB. As a result, TGIA0 does not occur and there is no timing for modifying the buffer register TGRC value or transferring the value of the TGRC to the TGRA. When switching from duty cycle 0% to 50%, the TGRC and TGRA registers are directly modified when TGIB0 occurs ((4) in Figure 3.69).

- TGRA compare match occurring twice in one cycle
  When switching from duty cycle 100% to 80%, two compare matches will occur, including the TGRA compare match that occurred at the same time as the TGRB compare match, so the timing for modifying the TGRC value must be adjusted ((5) in Figure 3.69).
  To switch from duty cycle 80% to 50%, set the TGRA to a value greater than the current value.
  This will cause two compare matches to occur in one cycle, so the timing for modifying the TGRC value must be adjusted ((6) in Figure 3.69).
  The waveform does not change because the second TGRA compare match occurs during high output ((7) in Figure 3.69).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.69 Sample Code Operations

3.11.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table3.14 Adding Components

Item	Description
Component	PWM Mode Timer
Configuration name	Config_MTU0
Operation	PWM Mode 1
Resource	MTU0

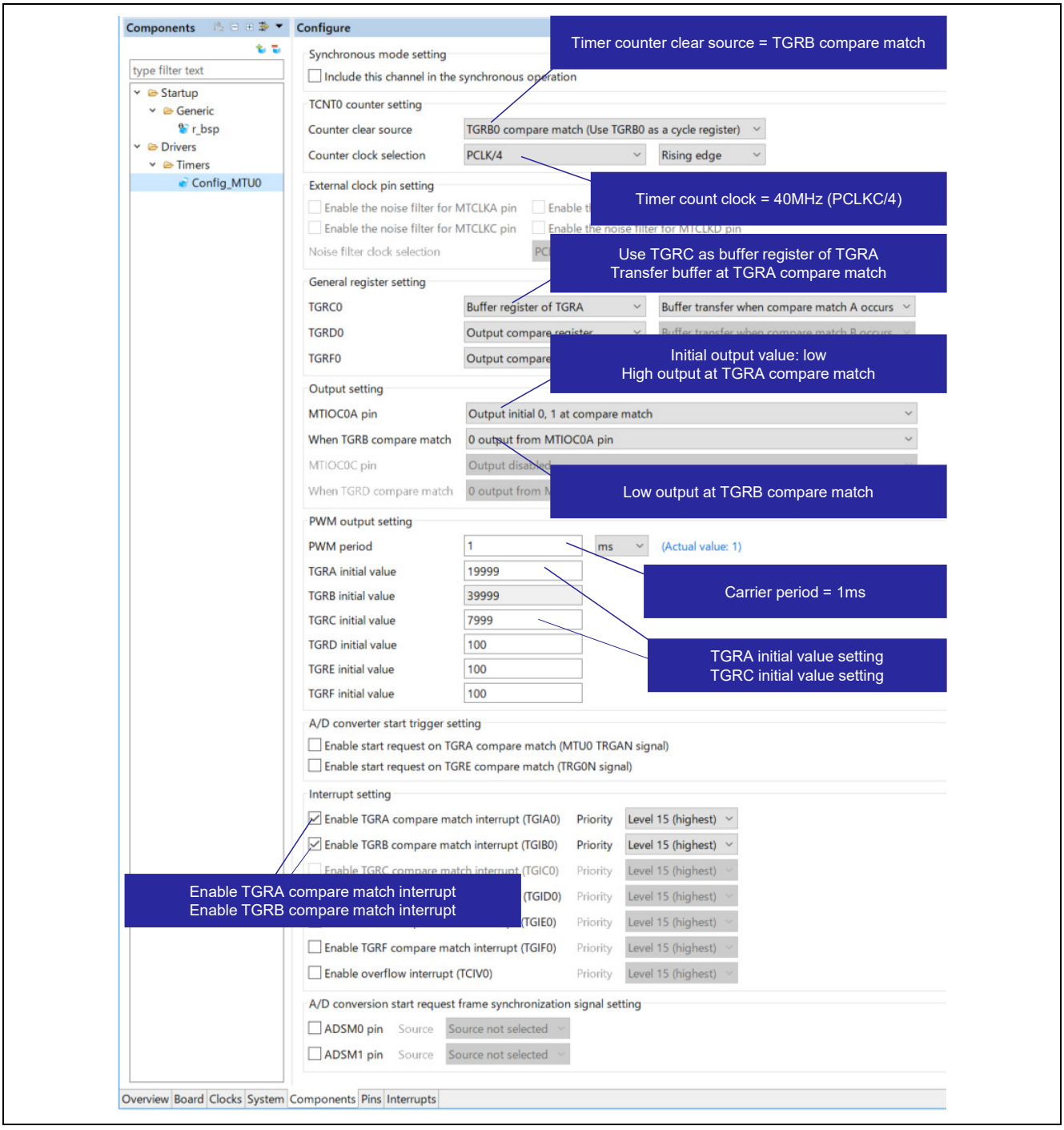


Figure 3.70 MTU0 Settings

3.11.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

Counting is started in the main function.

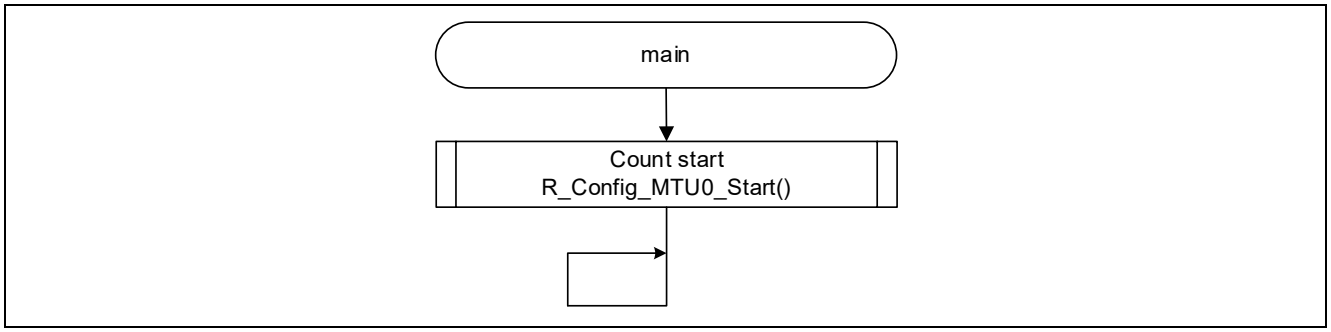


Figure 3.71 main Function

The user initialization function R_Config_MTU0_Create_UserInit, which is executed before the main function, initializes variables. This function is called from within the R_Config_MTU0_Create function.

This function initializes the following variables used in this sample code.

- s_duty_list_counter: counter variable for reading from the duty cycle list
- s_wait_cnt: wait variable for managing the number of TGIB0 occurrences during 0% duty cycle output
- s_duty_prv: variable for retaining the value of the previous TGRA register

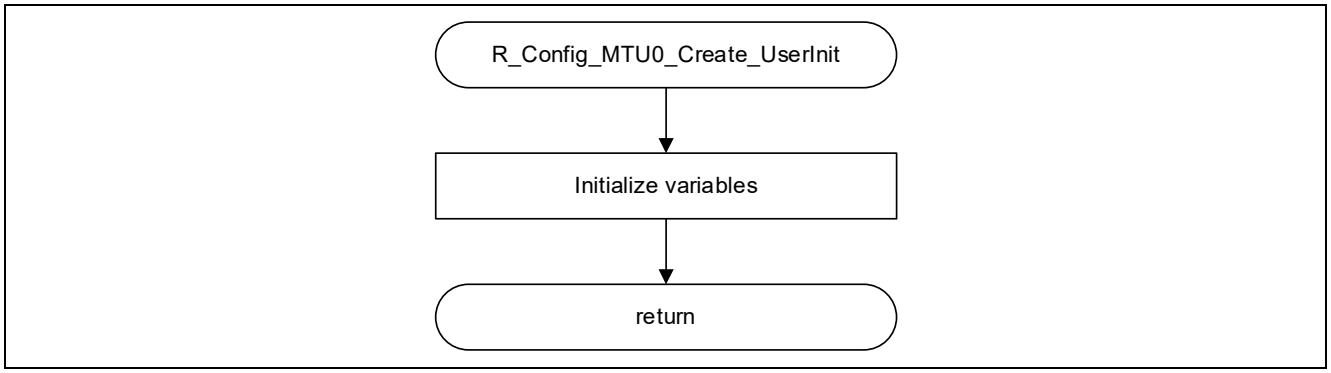


Figure 3.72 User Initialization Function

The TGIA0 interrupt handler function changes the value of the TGRC register according to the current value of the TGRA register and the value set in the previous TGRA register.

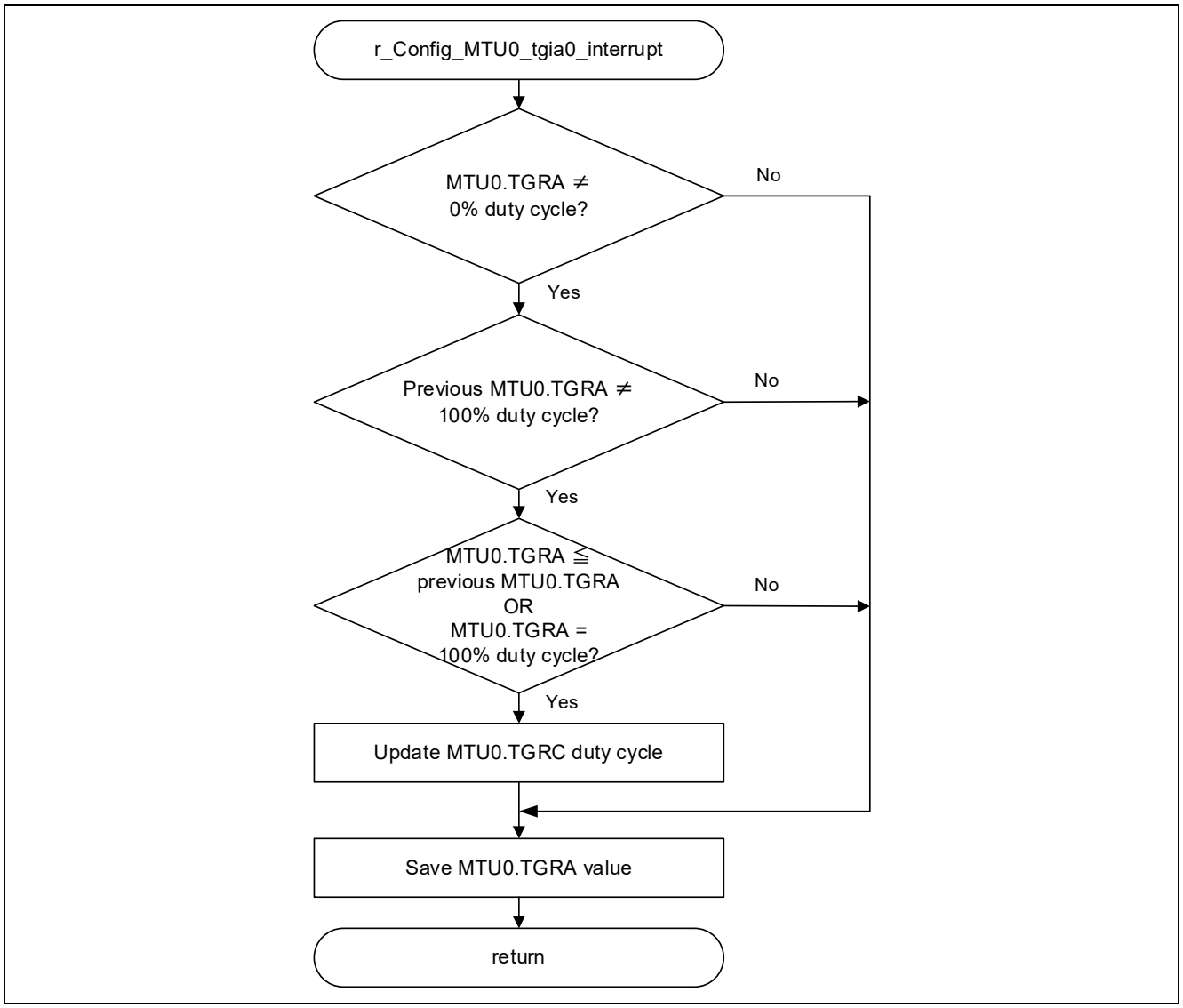


Figure 3.73 TGIA0 Interrupt Handler Function

The TGIB0 interrupt handler function sets the duty cycle directly to the TGRA and TGRC when switching from 0% duty cycle. If the TGRA is modified when the first TGIB0 occurs after 0% duty cycle is transferred to the TGRA, 0% is not output, so the duty cycle needs to be set when the second TGIB0 occurs, and the wait count needs to be managed.

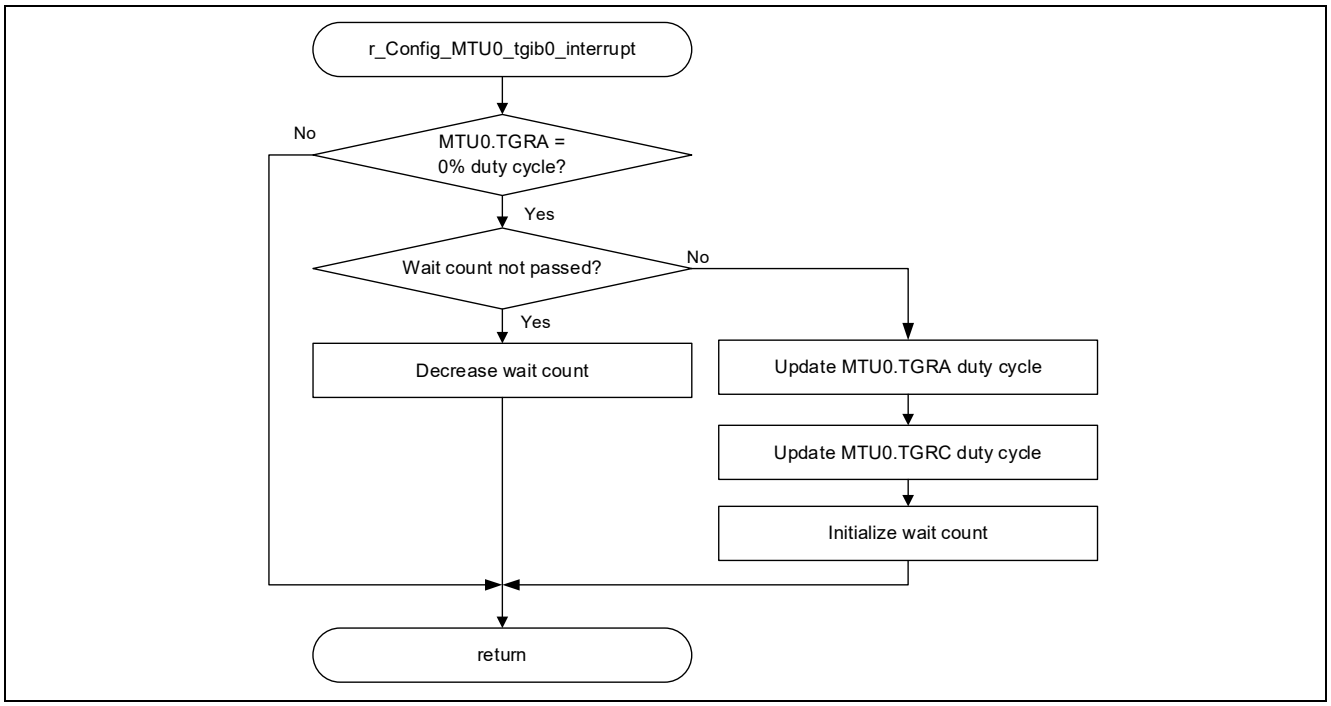


Figure 3.74 TGIB0 Interrupt Handler Function

3.11.5 Related Operations

3.11.5.1 Duty Cycles 0% to 100% (without buffer)

The following shows an example of PWM waveform output with duty cycles 0% to 100%, as shown in Figure 3.69, without using a buffer.

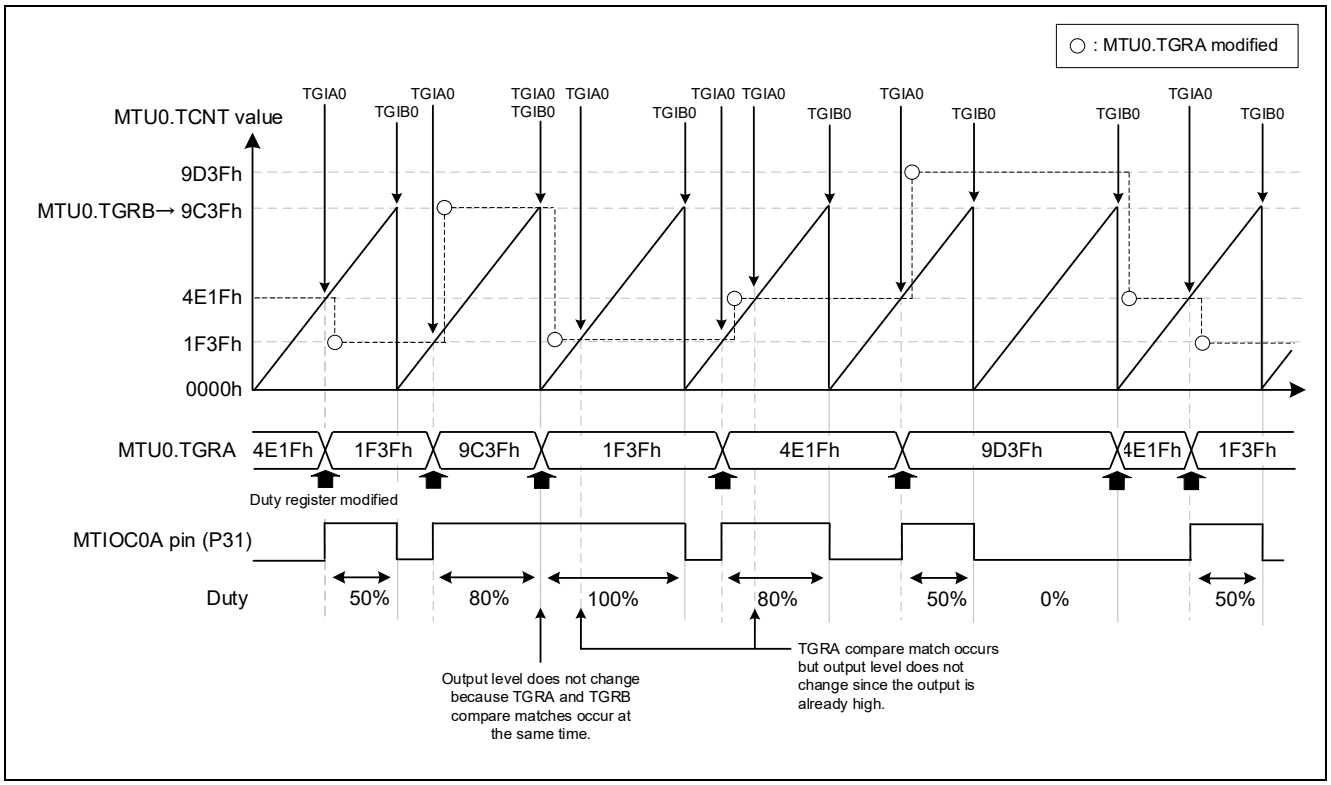


Figure 3.75 PWM Output with Duty Cycles 0% to 100% (without buffer)

3.11.6 Usage Notes

3.11.6.1 Buffer Register Modify Delay

In this sample code, the transfer from buffer register TGRC to TGRA is executed when a TGRA compare match occurs. Modification of the TGRC value must be completed before the next TGRA compare match occurs.

If the buffer register TGRC modification is delayed, the expected duty cycle cannot be output.

3.11.6.2 Setting a Value Greater than the Duty Register

In this sample code, buffer register TGRC is modified when the MTU0.TGRA compare match interrupt (TGIA0) is generated, and the value is transferred from the TGRC to the TGRA when a TGRA compare match occurs. If a value greater than the value currently set in the TGRA is set in the TGRA after a TGRA compare match occurs, two TGRA compare matches may occur in one cycle.

For details, refer to (6) in Figure 3.69.

3.11.6.3 Switching from 0% Duty Cycle

In this sample code, buffer register TGRC is modified when the MTU0.TGRA compare match interrupt (TGIA0) is generated, and the value is transferred from the TGRC to the TGRA when a TGRA compare match occurs. If a value greater than the TGRB is set in the TGRA and 0% duty cycle is output, the TGRA compare match does not occur because the counter is cleared before the TGRA value is reached.

When switching from 0% duty cycle, the value needs to be directly set in the TGRA register.

For details, refer to (4) in Figure 3.69.

3.12 PWM Mode 1 Duty Cycles 0% to 100% (compare register modify at counter clear)

- Target sample code file name: r01an5995_rx66t_mtu3_pwm1_50to100_rwcc.zip

3.12.1 Overview

The MTU PWM mode can be used to output PWM waveforms with duty cycles 0% to 100% according to the TGR register setting.

This sample code describes a sample code that uses PWM mode 1 and repeats the following waveform output, including duty cycles 0% and 100%.

- Duty switching: 50% → 80% → 100% → 80% → 50% → 0% → ・・・

The basic operation is to make changes to the duty cycle by transferring the TGRC value to the TGRA when a duty register TGRA compare match occurs using buffer register TGRC. When switching from 0% to 50%, the TGRA register is directly modified when a compare match occurs in period register TGRB. The TGRC is changed when a period register TGRB compare match occurs.

The following list provides the MTU settings used in the sample code.

- Use PWM mode 1
- Use channel 0
- Initial output value = low
- Carrier period = 1ms
- Timer count clock = 40MHz (PCLKC/4)
- Use TGRB as period register
  - Timer counter clear source = TGRB compare match
  - Low output at TGRB compare match
- Use TGRA as duty register
  - High output at TGRA compare match
- Use buffer register
  - Use TGRC as buffer register of TGRA
  - Buffer transfer when TGRA compare match occurs
- Duty changes at each cycle
  - Change duty cycle at TGRB compare match interrupt
  - Refer to Figure 3.77 for details on the timing for duty cycle changes

Set in Smart Configurator.
For Setting Methods,
refer to section 3.12.3.

PWM mode 1 output for this sample code is shown below.

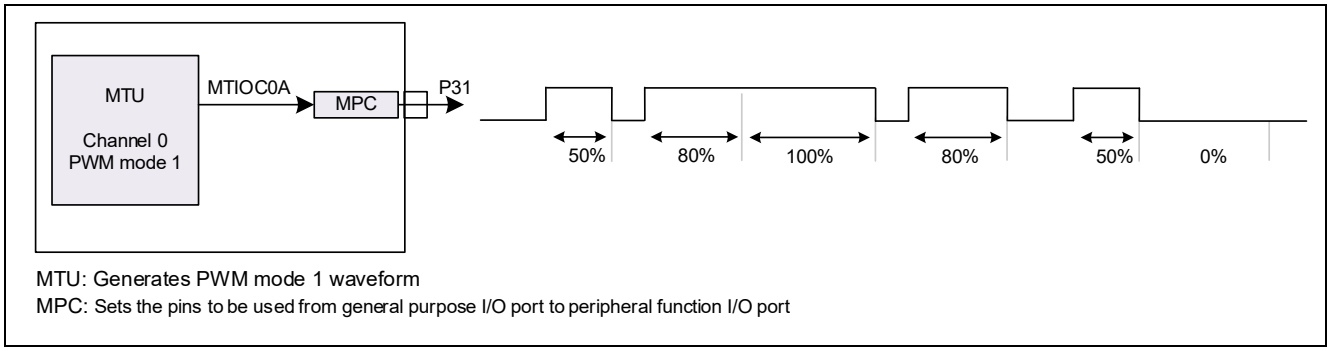
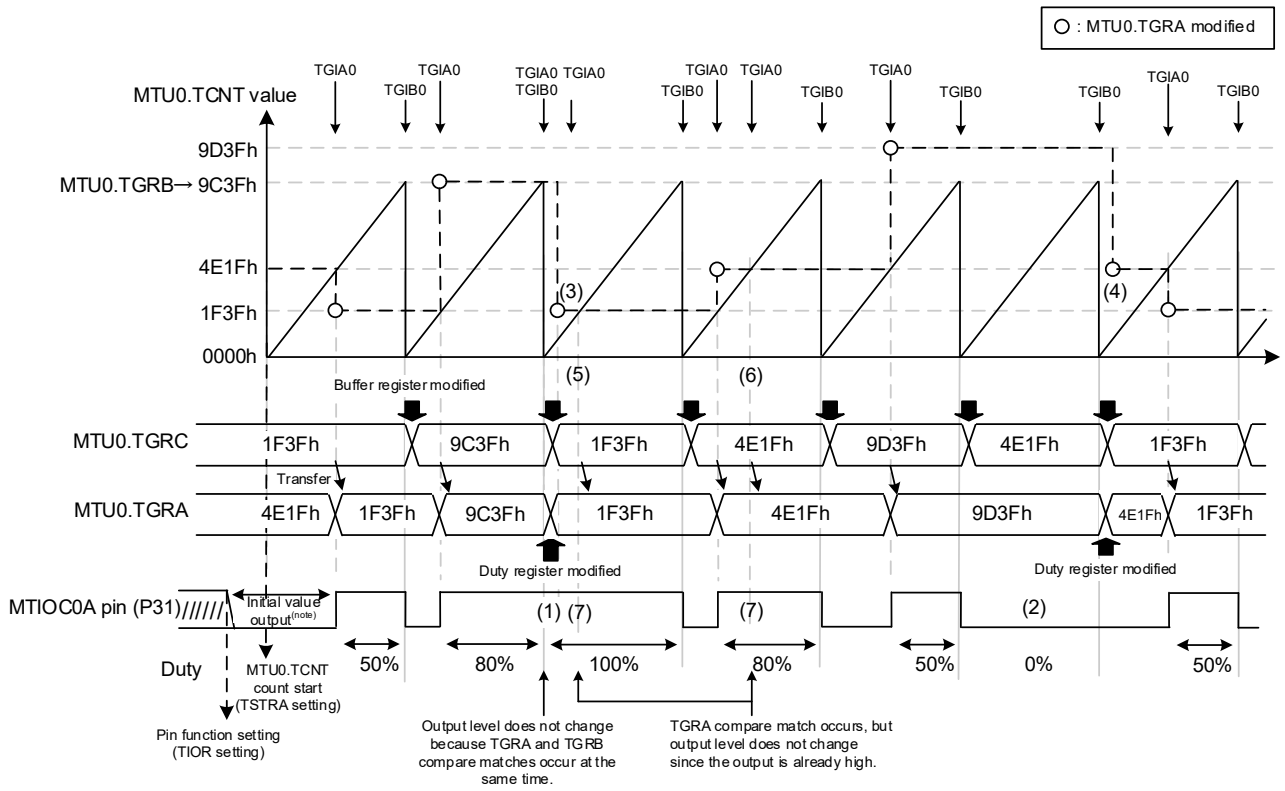


Figure 3.76 PWM Mode 1 Output

3.12.2 Operation Details

The sample code operations are shown in Figure 3.77. The basic operation is to make changes to the settings of the duty cycle with each period by modifying buffer register TGRC value at the compare match interrupt (TGIB0) of period register TGRB and transferring the value of TGRC to duty register TGRA when a TGRA compare match occurs

- 100% Duty Cycle Output
  When outputting 100% duty cycle, set the TGRA to the same value as the TGRB.
  When the TGRA value is the same as the TGRB value, both compare matches occur at the same time. Because output does not change when two compare matches occur at the same time, high is retained and 100% duty cycle is output ((1) in Figure 3.77).
- 0% Duty Cycle Output
  When outputting 0% duty cycle, set the TGRA to a value greater than the TGRB.
  When the TGRA value is higher than the TGRB, low is retained and 0% duty cycle is output because the TGRA compare match does not occur ((2) in Figure 3.77).
- Switching from 100% duty cycle
  When 100% duty cycle is output, TGRA and TGRB compare matches occur at the same time. As a result, there is no timing for modifying the TGRC to a new value. When switching from duty cycle 100% to 0%, the values of the TGRA register is directly modified when TGIB0 occurs ((3) in Figure 3.77).
- Switching from 0% duty cycle
  When 0% duty cycle is output, a TGRA compare match does not occur because the TGRA is greater than the TGRB. As a result, there is no timing for transferring the value of the TGRC to the TGRA. When switching from duty cycle 0% to 50%, the values of the TGRC and TGRA registers are directly modified when TGIB0 occurs ((4) in Figure 3.77).
- TGRA compare match occurring twice in one cycle
  When switching from duty cycle 100% to 80%, two compare matches will occur, including the TGRA compare match that occurred at the same time as the TGRB compare match ((5) in Figure 3.77).
  To switch from duty cycle 80% to 50%, set the TGRA to a value greater than the current value.
  This will cause two TGRA compare matches to occur in one cycle ((6) in Figure 3.77).
  The waveform does not change because the second TGRA compare match occurs during high output ((7) in Figure 3.77).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.77 Sample Code Operations

3.12.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table3.15 Adding Components

Item	Description
Component	PWM Mode Timer
Configuration name	Config_MTU0
Operation	PWM Mode 1
Resource	MTU0

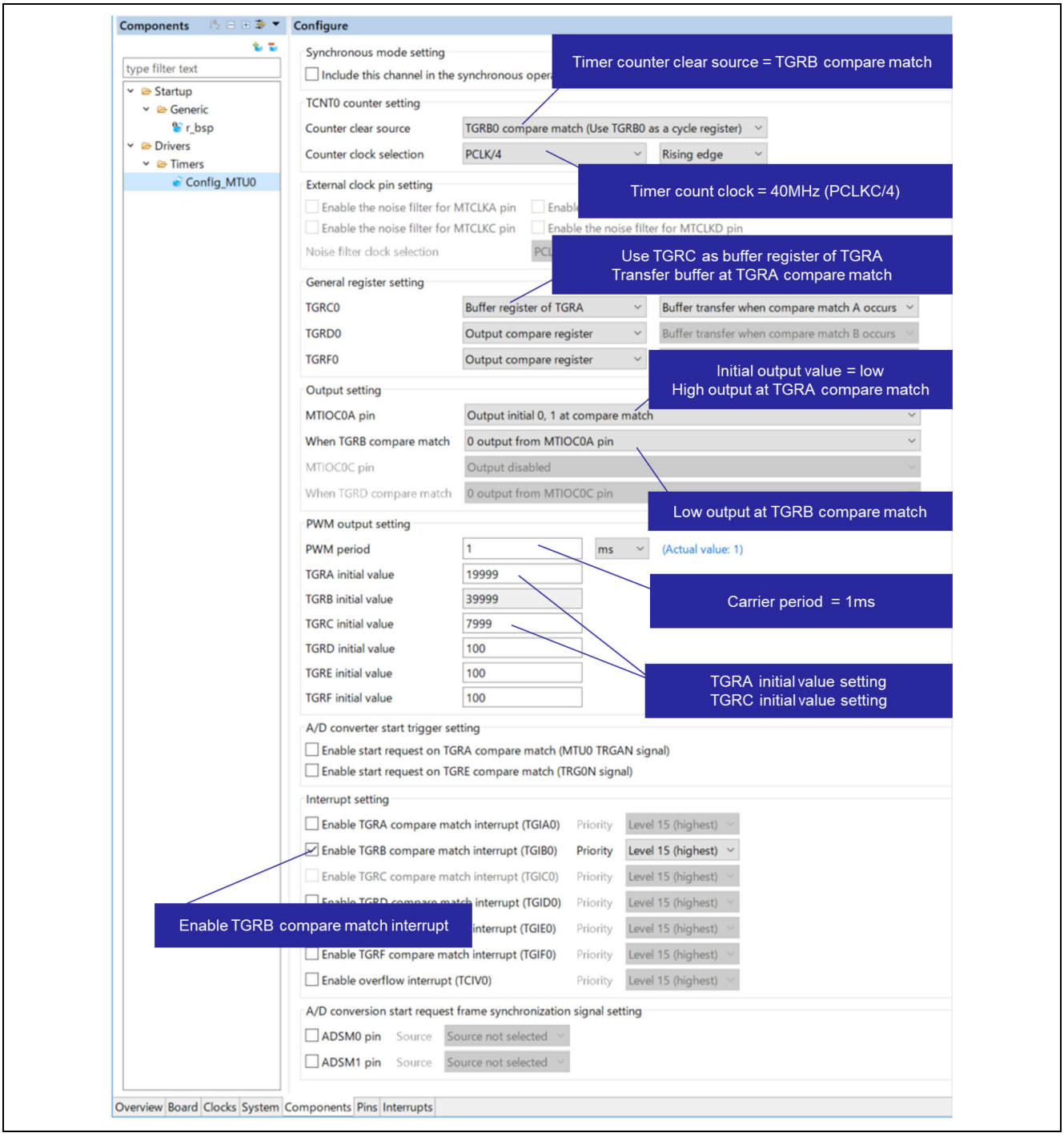


Figure 3.78 MTU0 Settings

3.12.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

Counting is started in the main function.

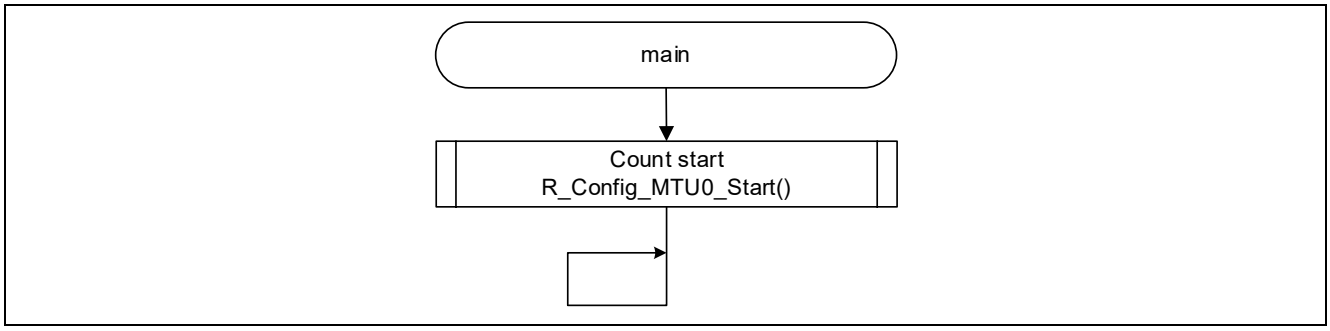


Figure 3.79 main Function

The user initialization function R_Config_MTU0_Create_UserInit, which is executed before the main function, initializes variables. This function is called from within the R_Config_MTU0_Create function.

This function initializes the following variables used in this sample code.

- s_duty_list_counter: counter variable for reading from the duty cycle list
- s_duty_prv: variable for retaining the value of the previous TGRA register

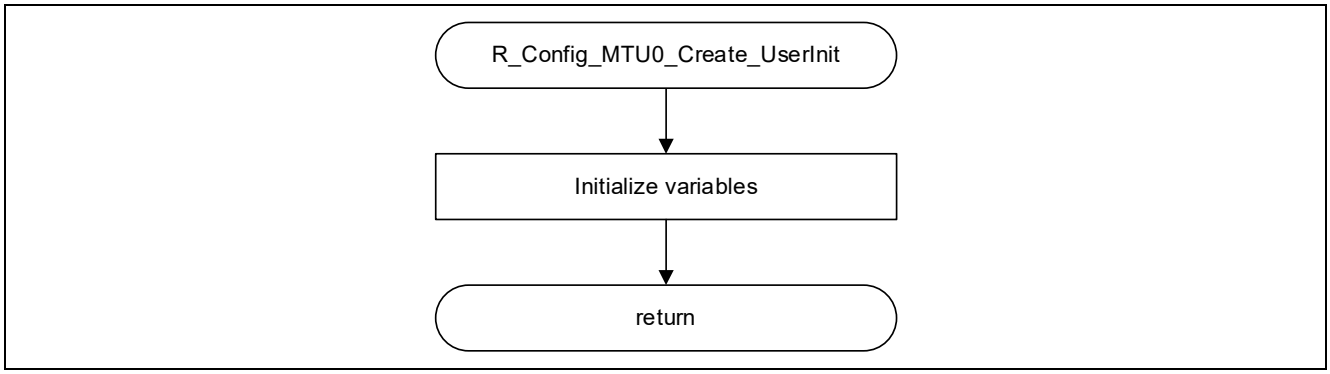


Figure 3.80 User Initialization Function

The TGIB0 interrupt handler function directly sets the value of the TGRA register when the duty cycle is switched from 100% or 0%. The value of the TGRC register is changed each time.

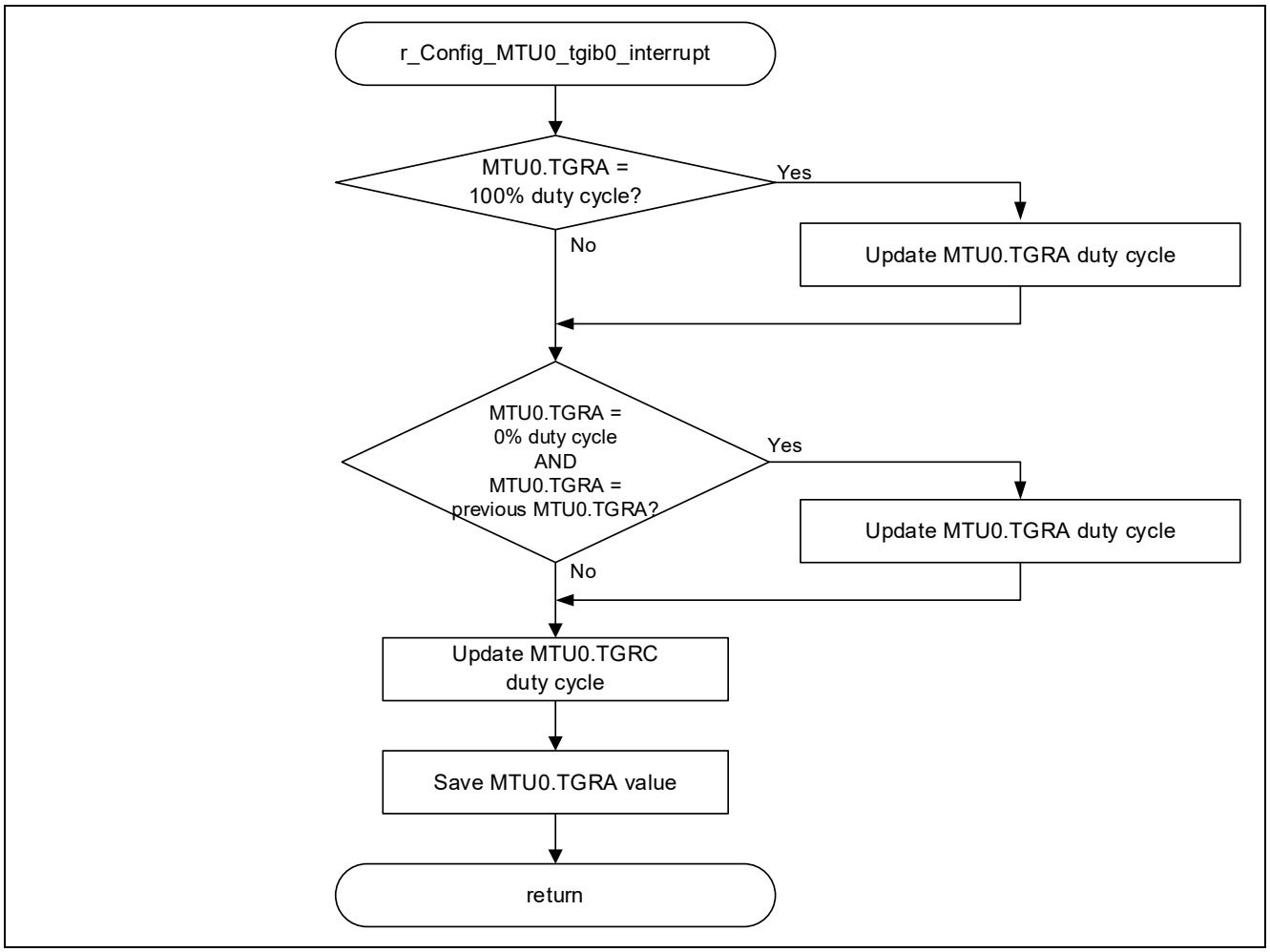


Figure 3.81 TGIB0 Interrupt Handler Function

3.12.5 Related Operations

3.12.5.1 Duty Cycles 0% to 100% (without buffer)

The following shows an example of PWM waveform output with duty cycles 0% to 100%, as shown in Figure 3.69, without using a buffer.

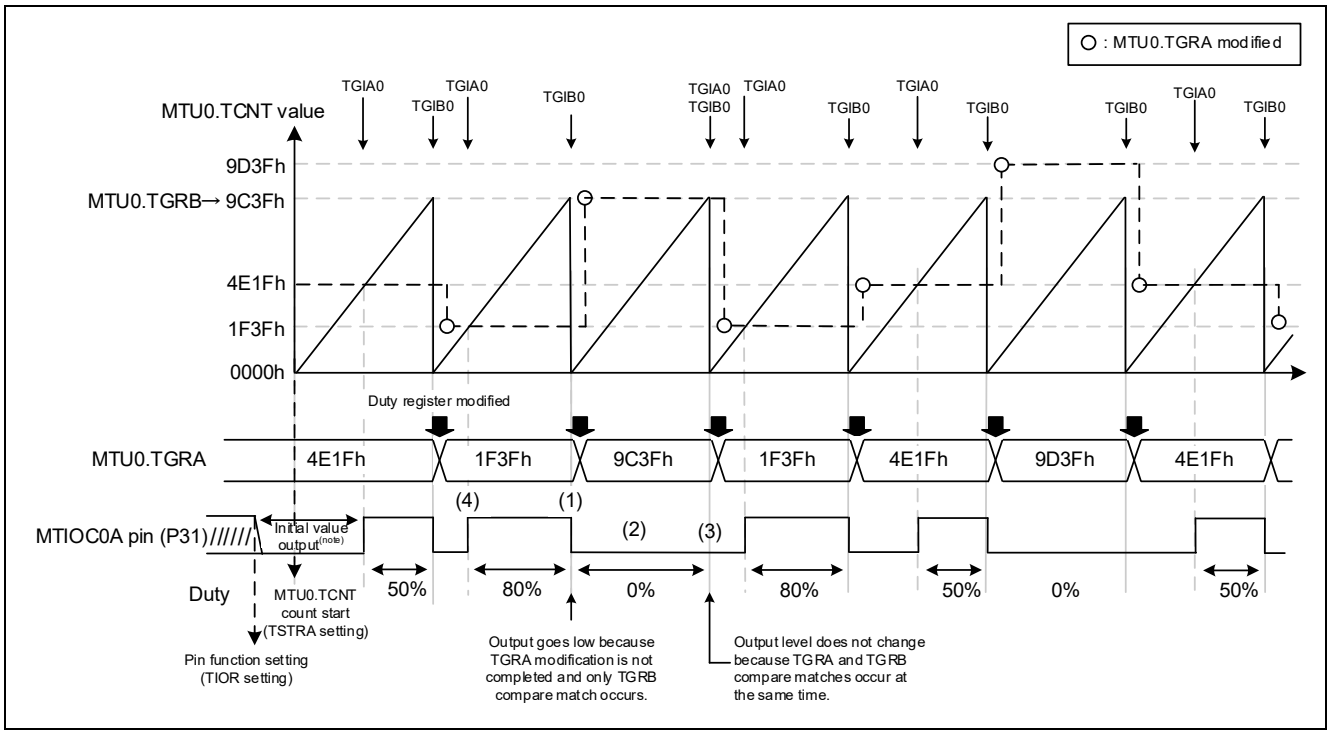


Figure 3.82 PWM Output with Duty Cycles 0% to 100% (without buffer)

The following are the differences of outputting PWM waveforms with duty cycles 0% to 100% when using and not using a buffer.

- 100% Duty Cycle Output
  When outputting 100% duty cycle, set the TGRA to the same value as the TGRB.
  The TGRA value is changed when TGIB0 occurs, in the timing shown in (1) in Figure 3.82, and output is changed to low because only the TGRB compare match occurs. As a result, even though 100% is desired the output, 0% is output ((2) in Figure 3.82). Also, because output does not change when the TGRA and TGRB compare matches occur at the same time, low is retained ((3) in Figure 3.82).

Figure 3.83 shows how to modify the TGRA at the timing indicated by (4) to avoid the conditions of (2) in Figure 3.82.

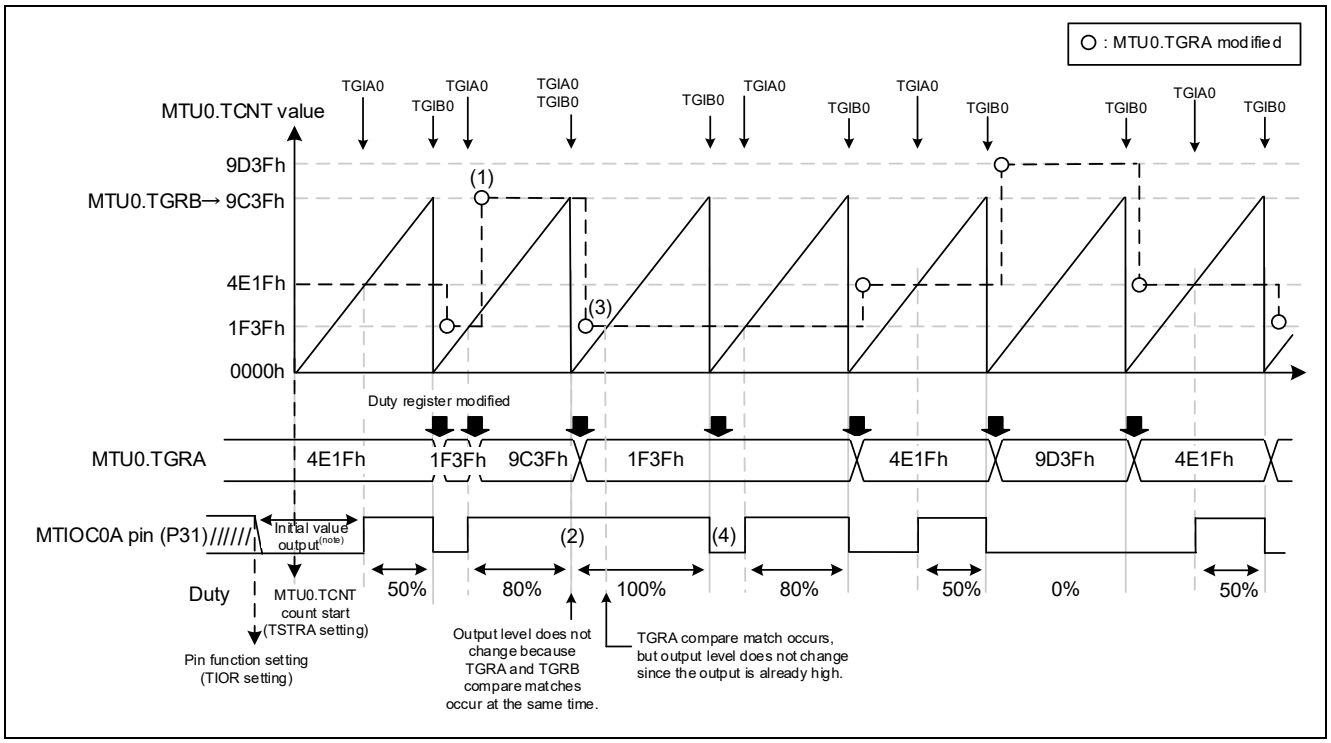


Figure 3.83 PWM Output with Duty Cycles 0% to 100% (without buffer)

- 100% Duty Cycle Output
  When outputting 100% duty cycle, set the TGRA to the same value as the TGRB.
  The TGRA value is modified when TGIA0 occurs ((1) in Figure 3.83).
  When the TGRA value is the same as the TGRB value, both compare matches occur at the same time. Because output does not change when two compare matches occur at the same time, high is retained and 100% duty cycle is output ((2) in Figure 3.83).
  The TGRA is changed to 80% when TGIB0 occurs ((3) Figure 3.83).

- Switching from 100% duty cycle
  The pin outputs low when the 100% duty cycle TGRB compare match occurs ((4) Figure 3.83). At this time, the TGRA remains at 80% again.

- TGRA compare match occurring twice in one cycle
  When switching from duty cycle 100% to 80%, two compare matches will occur, including the TGRA compare match that occurred at the same time as the TGRB compare match ((3) in Figure 3.83).
  The waveform does not change because the second TGRA compare match occurs during high output.

3.12.6 Usage Notes

3.12.6.1 Contention between Buffer Register Modification and Compare Match

If the buffer transfer timing is set at compare match, the data in the buffer register before the modification is transferred to the duty register if a compare match occurs.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.7 Contention between Buffer Register Write Operation and Compare Match.

3.12.6.2 Buffer Register Modify Delay

In this sample code, the transfer from buffer register TGRC to TGRA is executed when a TGRA compare match occurs. Modification of the TGRC value must be completed before the next compare match occurs.

If the buffer register modification is delayed, the TGRA value may not be updated by the time the TGRA compare match occurs in the next cycle. In that case, consider transferring the buffer when counter clear occurs.

3.12.6.3 Setting a Value Greater than the Duty Register

In this sample code, buffer register TGRC is modified when the MTU0.TGRB compare match interrupt (TGIB0) is generated, and the value is transferred from the TGRC to the TGRA when a TGRA compare match occurs. If a value greater than the value currently set in the TGRA is set in the TGRA after a TGRA compare match occurs, two TGRA compare matches may occur in one cycle.

For details, refer to (7) in Figure 3.69.

3.12.6.4 Switching from 0% Duty Cycle

In this sample code, buffer register TGRC is modified when the MTU0.TGRB compare match interrupt (TGIB0) is generated, and the value is transferred from the TGRC to the TGRA when a TGRA compare match occurs. If a value greater than the TGRB is set in the TGRA, when 0% duty cycle is output, the TGRA compare match does not occur because the counter is cleared before the TGRA value is reached.

When switching from 0% duty cycle, the value needs to be directly set in the TGRA register.

For details, refer to (4) in Figure 3.69.

3.13 PWM Mode 1 Duty Cycles 0% and 100%

- Target sample code file name: r01an5995_rx66t_mtu3_pwm1_0to100.zip

3.13.1 Overview

The MTU PWM mode can be used to output PWM waveforms with duty cycles 0% to 100% according to the TGR register setting.

This sample code describes a sample code that uses PWM mode 1 and repeats waveform output alternating between duty cycles 0% and 100%.

Waveform output cannot be alternated between duty cycles 0% and 100% with each cycle using duty register TGRA, therefore, by switching output for high and low when a period register TGRB compare match occurs, waveforms with duty cycles 0% and 100% are output. Users should be aware of potential conflicts with other processes, interrupt delays, etc., and evaluate the product thoroughly before use if the sample code procedure has been adapted to the user system because the TIOR register setting is modified during timer operations.

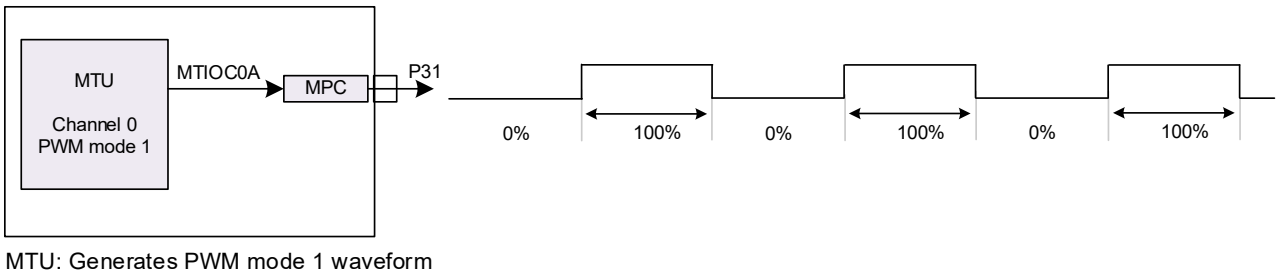
Note that alternating between duty cycles 0% and 100% can be realized easily by setting the output to toggle when a TGRB compare match occurs. In this sample code, the TIOR register setting is modified to allow for combinations with duty cycles other than 0% or 100%.

The following list provides the MTU settings used in the sample code.

- Use PWM mode 1
- Use channel 0
- Initial output value = low
- Carrier period = 1ms
- Timer count clock = 40MHz (PCLKC/4)
- Use TGRB as period register
  - Timer counter clear source = TGRB compare match
  - High output at TGRB compare match
- Use TGRA as duty register
  - High output at TGRA compare match
- Use buffer register
  - Use TGRC as buffer register of TGRA
  - Buffer transfer when TGRA compare match occurs
- Duty changes at each cycle
  - Set TGRB compare match output at TGRB compare match interrupt
  - Refer to Figure 3.85 for details on the timing for duty cycle changes.

Set in Smart Configurator.
For Setting Methods,
refer to section 3.13.3.

PWM mode 1 output for this sample code is shown below.



MTU: Generates PWM mode 1 waveform
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

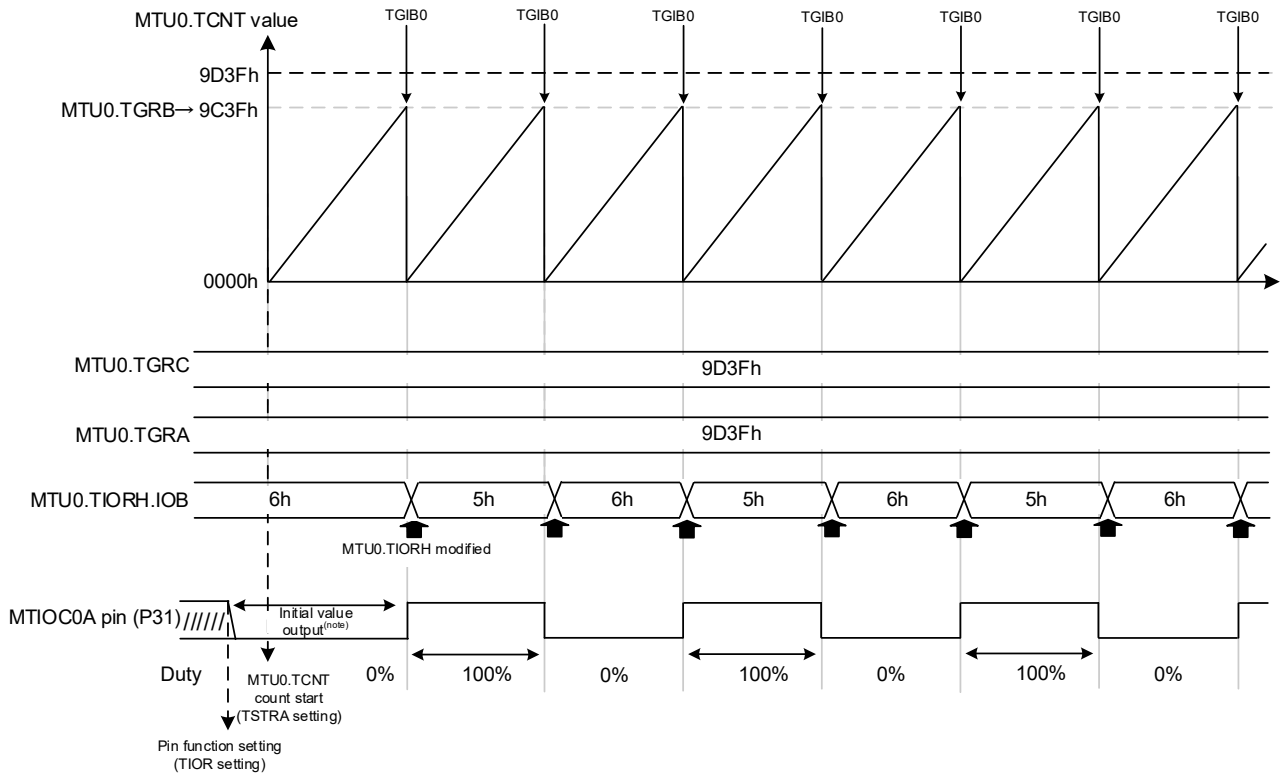
Figure 3.84 PWM Mode 1 Output

3.13.2 Operation Details

The sample code operations are shown in Figure 3.85. To ensure the output waveforms are not changed by a compare match of duty register TGRA, the TGRA and TGRC are set to a value greater than that of period register TGRB.

Duty cycle is alternated between 0% and 100% by modifying the value of the TIOR register as indicated below at a TGRB compare match interrupt (TGIB0).

- MTU0.TIORH.IOB = 5h: Low output at TGRB compare match
- MTU0.TIORH.IOB = 6h; High output at TGRB compare match



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.85 Sample Code Operations

3.13.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table3.16 Adding Components

Item	Description
Component	PWM Mode Timer
Configuration name	Config_MTU0
Operation	PWM Mode 1
Resource	MTU0

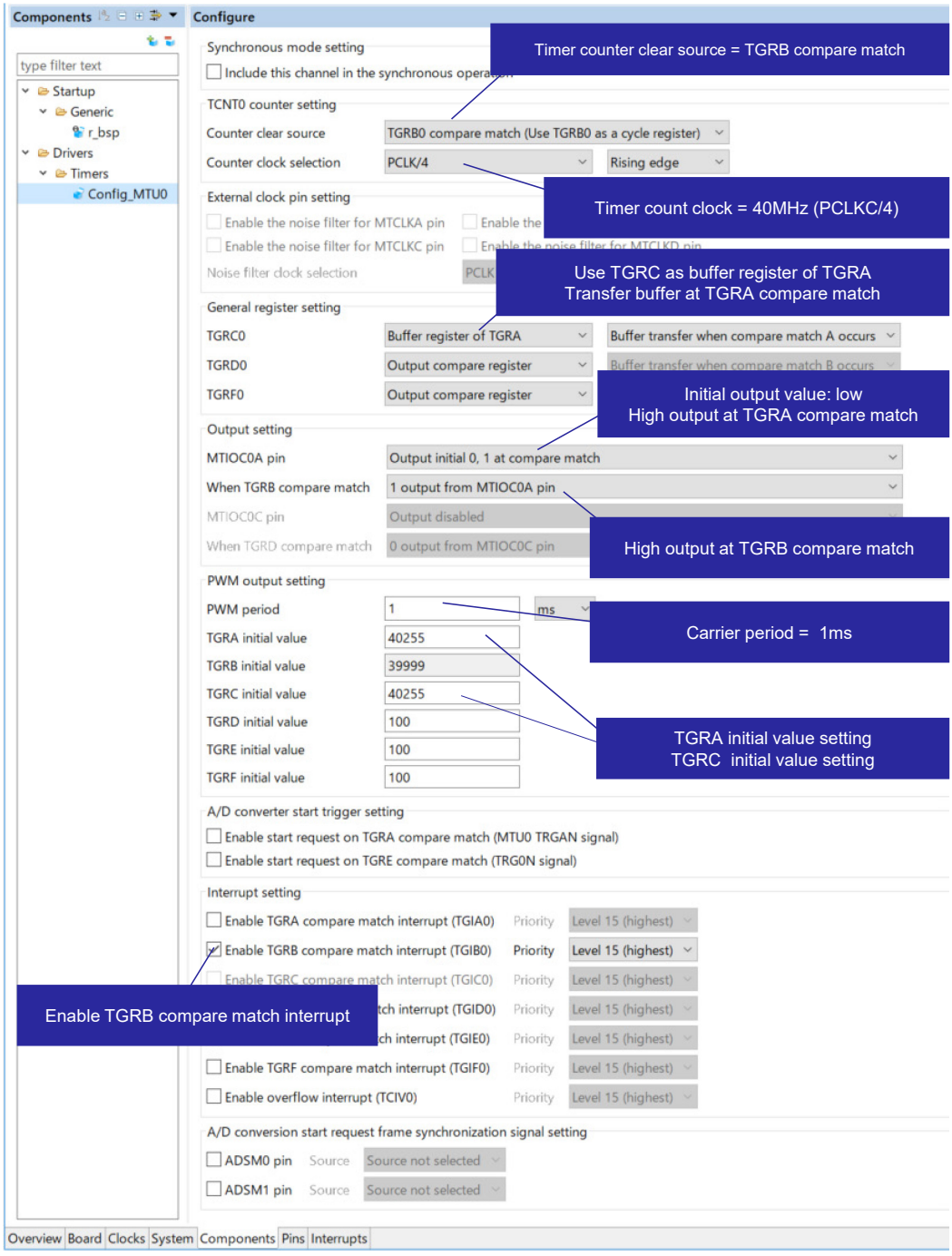


Figure 3.86 MTU0 Settings

3.13.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

Counting is started in the main function.

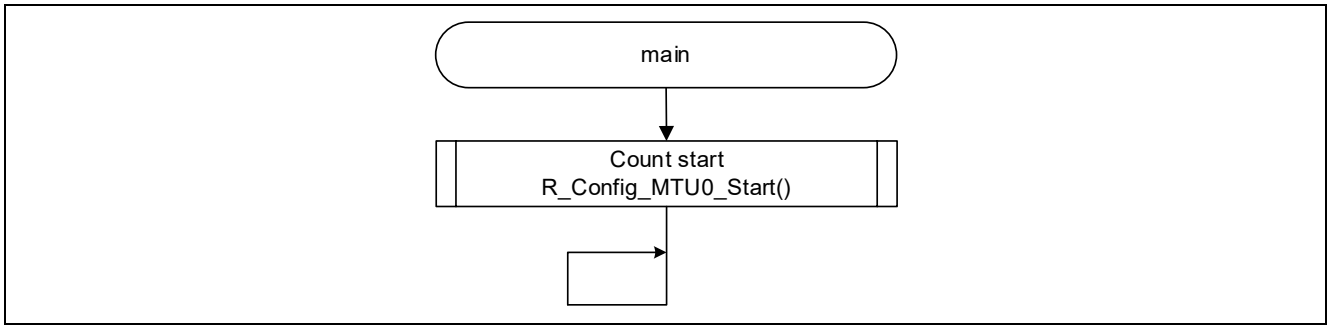


Figure 3.87 main Function

The TGIB0 interrupt handler function changes the value of the TIOR register and changes the output setting when a TGRB compare match occurs.

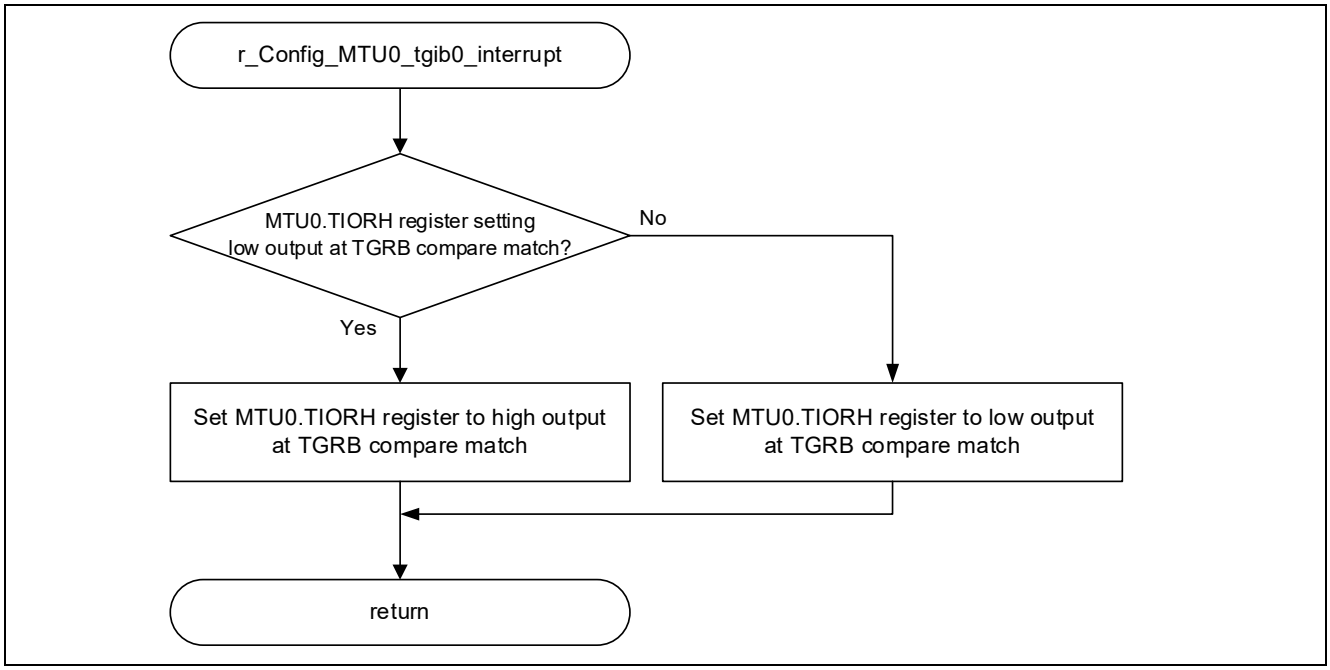


Figure 3.88 TGIB0 Interrupt Handler Function

3.13.5 Related Operations

3.13.5.1 Duty Cycles 0% and 100% (without buffer)

The following shows an example of PWM waveform output with duty cycles 0% and 100%, as shown in Figure 3.85, without using a buffer. Since the buffer register and duty register are not used to switch the duty cycle, the procedure is the same as using a buffer.

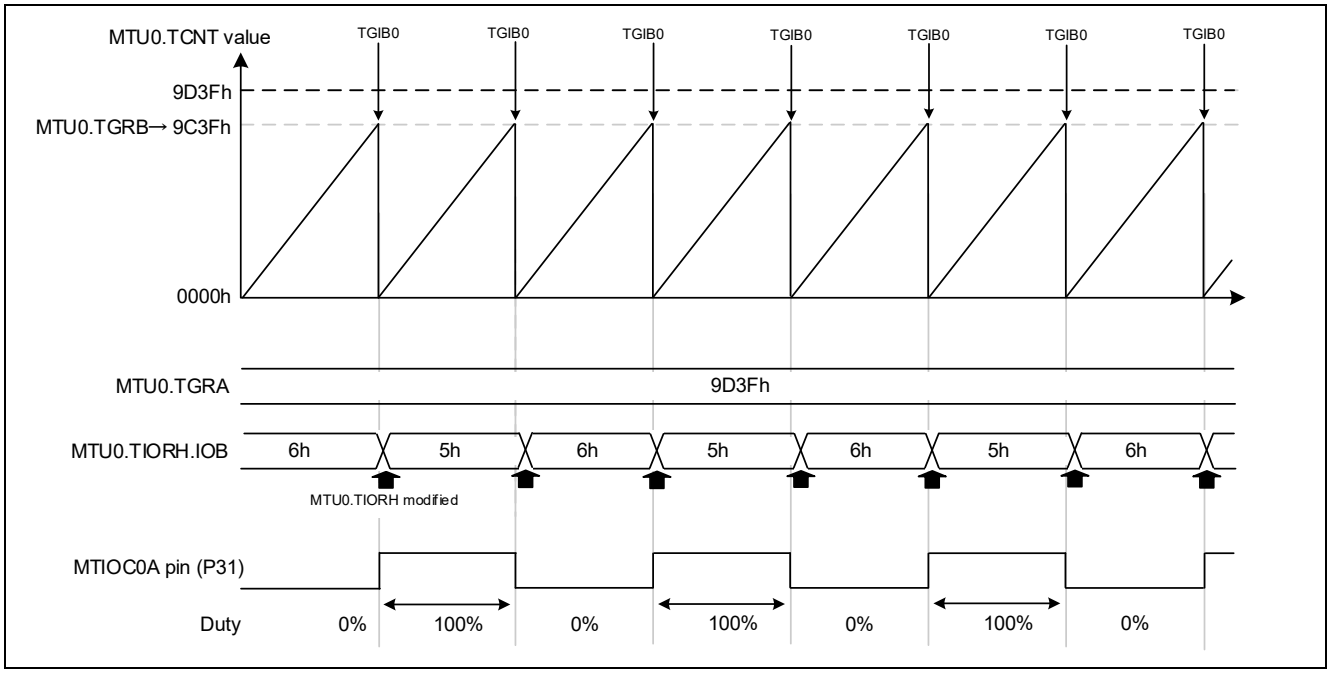
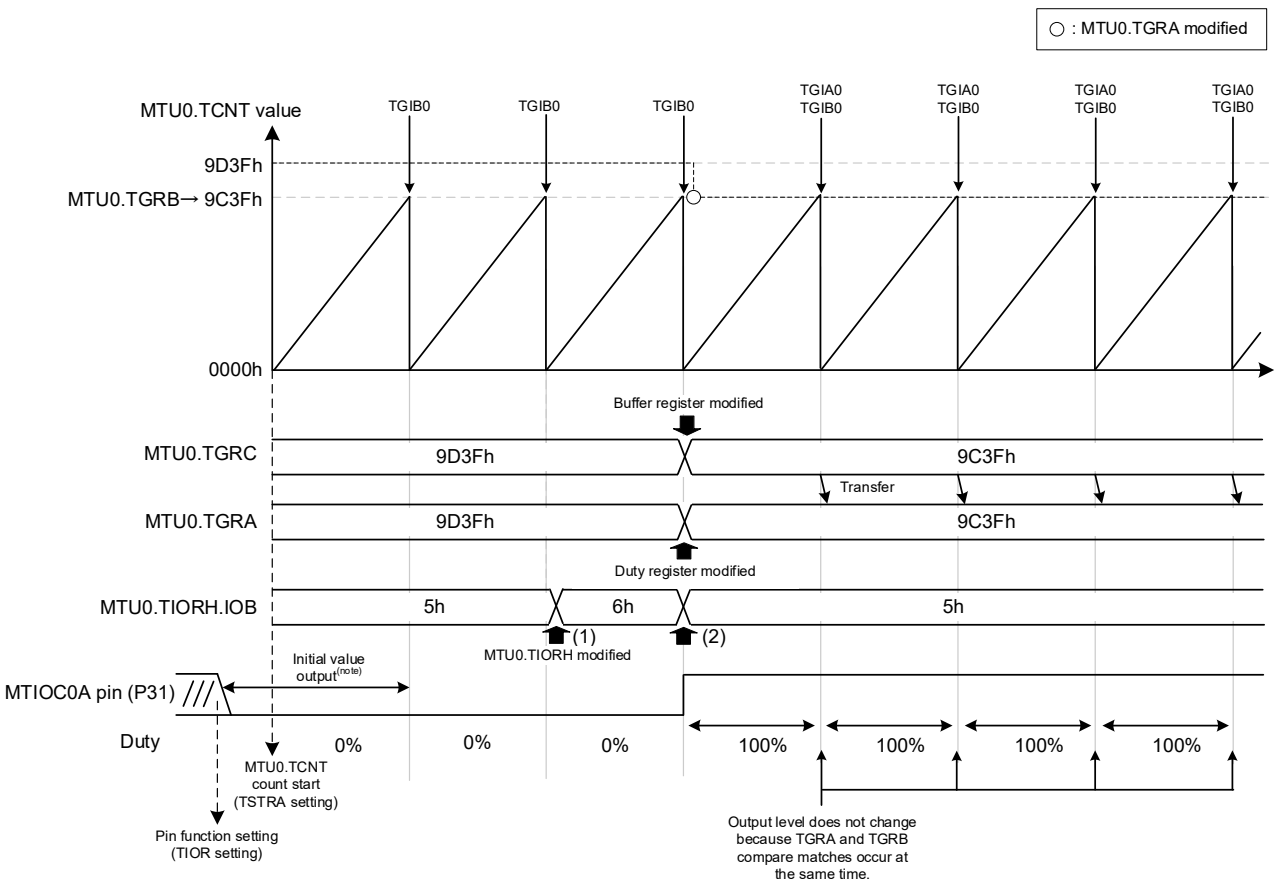


Figure 3.89 PWM Output with Duty Cycles 0% and 100% (without buffer)

3.13.5.2 Change Duty Cycle to 100% After Holding 0% at Initial Value Low

The following shows an example of operations when the duty cycle is switched to 100% after outputting 0% for several cycles.

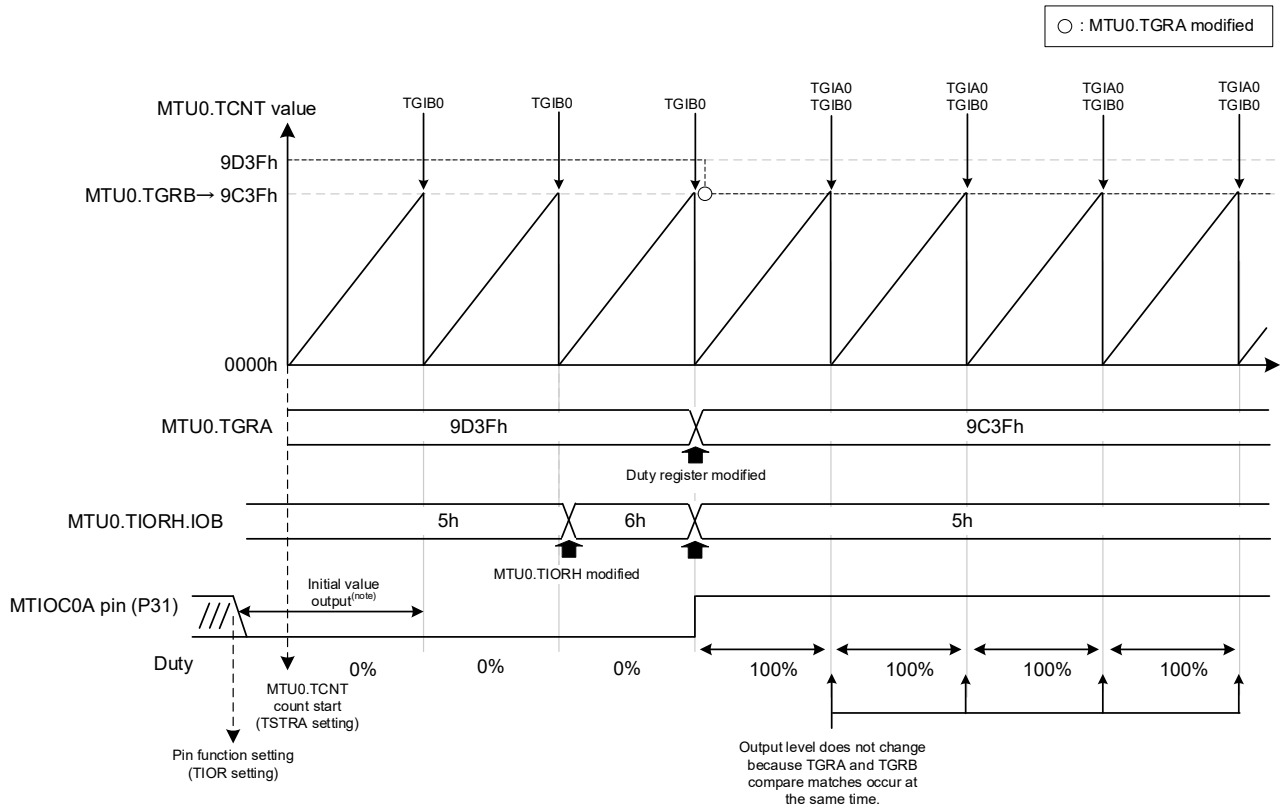
The MITOC0A pin outputs an initial value low and 0% duty cycle after operations start when TIORH.IOA is set to 2h and TIORH.IOB is set to 5h. Then, when TIORH.IOB is set to 6h ((1) in figure below), the pin outputs high at a TGRB compare match; at the next TGIB0 occurs, when TGRA = TGRC = TGRB and TIORH.IOB is set to 5h ((2) in figure below), the duty cycle goes to 100%.



Note: The waveform in the sample code starts outputting initial values when the PMR register is set.

Figure 3.90 Change Duty Cycle to 100% After Holding 0% (with buffer)

The following shows an example of operations when the duty cycle is switched to 100% after outputting 0% for several cycles, as shown Figure 3.90, when not using a buffer. The procedure is the same as using a buffer.



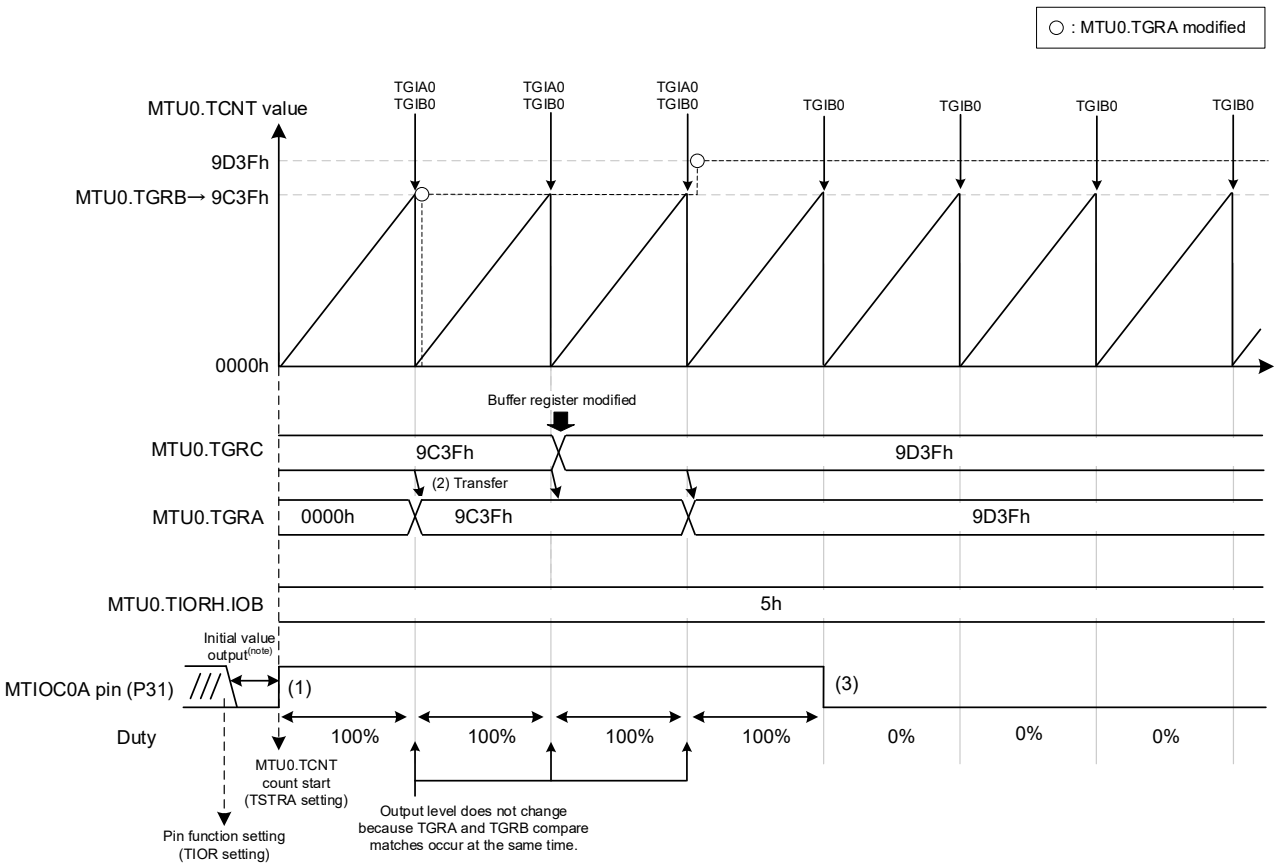
Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.91 Change Duty Cycle to 100% After Holding 0% (without buffer)

3.13.5.3 Change Duty Cycle to 0% After Holding 100% at Initial Value Low

The following shows an example of operations when the duty cycle is switched to 0% after outputting at 100% for several cycles.

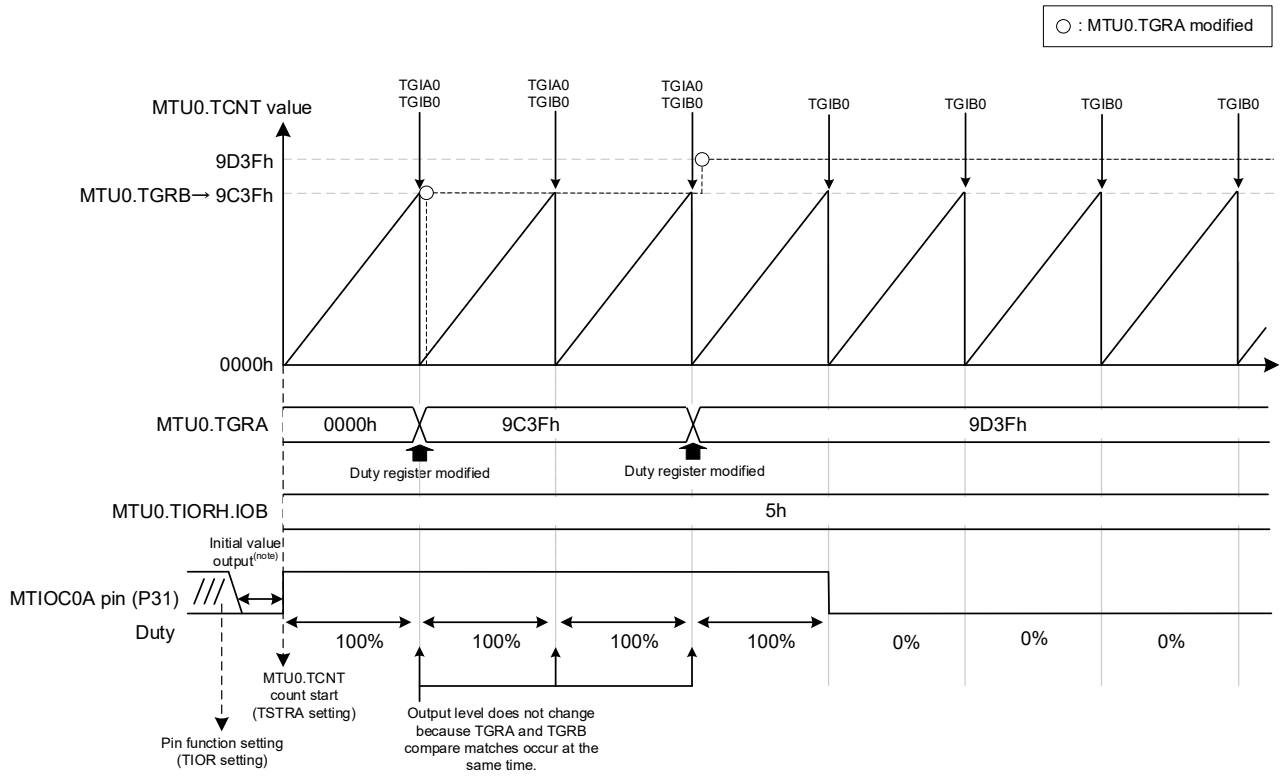
The MITOC0A pin is initially set to low when TIORH.IOA is set to 2h and TIORH.IOB is set to 5h. With the initial TGRA value of 0000h, when TCNT counts up from 0000h to 0001h, a compare match occurs and high is output, resulting in 100% duty output ((1) in figure below). At count clear, the same value as the TGRB is sent from the TGRC to the TRGA ((2) in figure below). When the TGRC is set to a value greater than the TGRB at any timing, low is output after 2 cycles and 0% duty cycle is output ((3) in figure below).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.92 Change Duty Cycle to 0% After Holding 100% (with buffer)

The following is an example of operations when the duty cycle is switched to 0% after outputting 100% for several cycles, as shown in Figure 3.92, without using a buffer. The procedure is the same as using a buffer, but after a value greater than the TGRB is set to the TGRA, 0% duty cycle is output in the next cycle.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

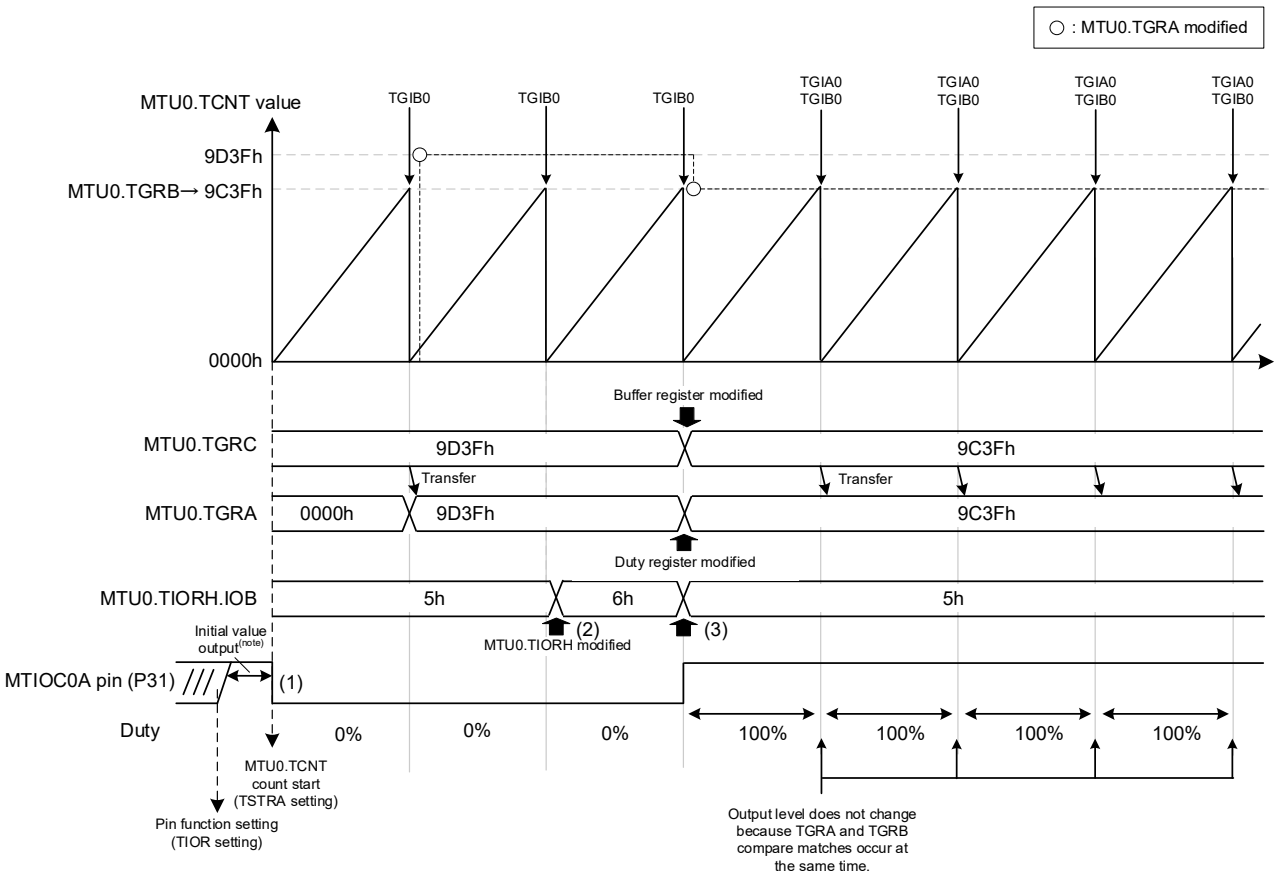
Figure 3.93 Change Duty Cycle to 0% After Holding 100% (without buffer)

3.13.5.4 Change Duty Cycle to 100% After Holding 0% at Initial Value High

The following shows an example of operations when the duty cycle is switched to 0% after outputting 0% for several cycles.

The MITOC0A pin is initially set to low when TIORH.IOA is set to 5h and TIORH.IOB is set to 5h. With the initial TGRA value of 0000h, when TCNT counts up from 0000h to 0001h, a compare match occurs and low is output, resulting in 0% duty output ((1) in figure below). Then, when TIORH.IOB is set to 6h ((2) in figure below), the pin outputs high at a TGRB compare match; at the next TGIB0 occurrence, when TGRA = TGRC = TGRB and TIORH.IOB is set to 5h ((3) in figure below), the duty cycle goes to 100%.

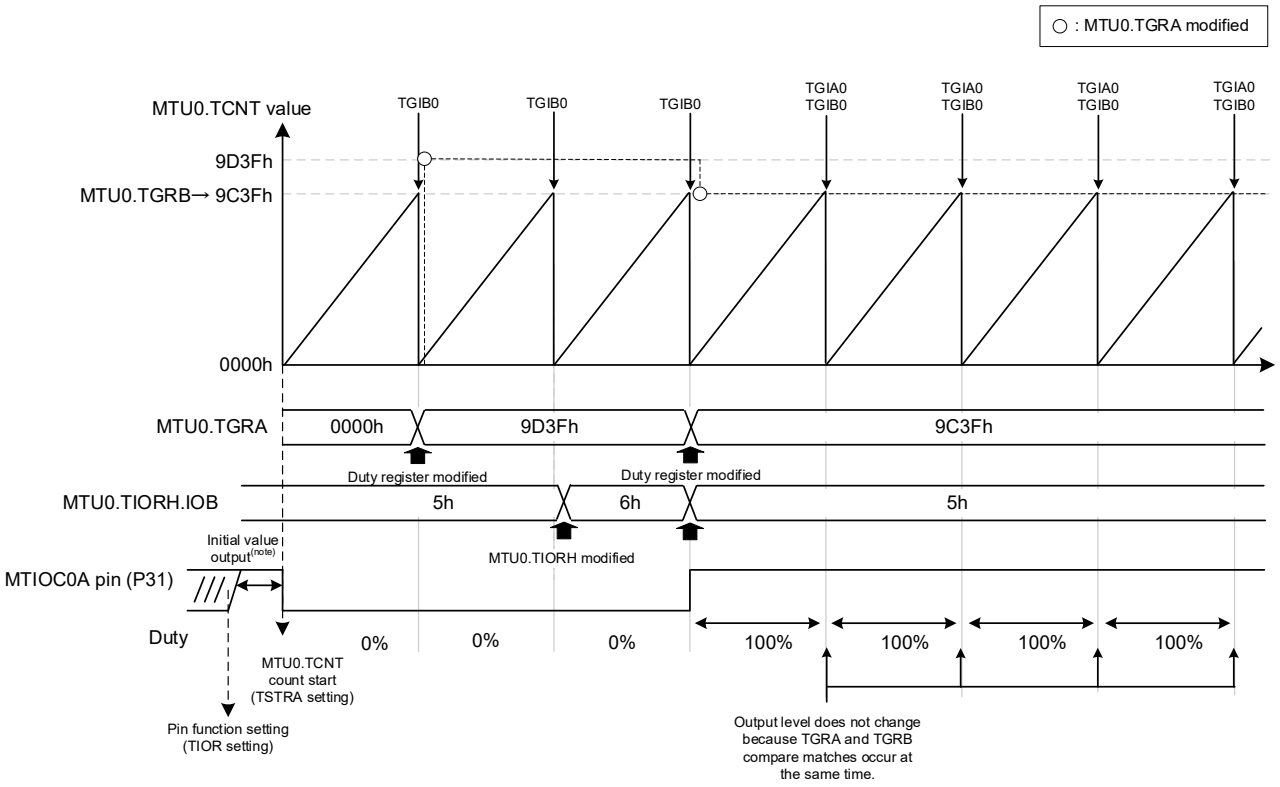
Also, active-high PWM can be output when TIORH.IOA is set to 2h at the timing indicated by (2).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.94 Change Duty Cycle to 100% After Holding 0% (with buffer)

The following shows an example of operations when the duty cycle is switched to 100% after outputting 0% for several cycles, as shown in Figure 3.94, when not using a buffer. The procedure is the same as using a buffer.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

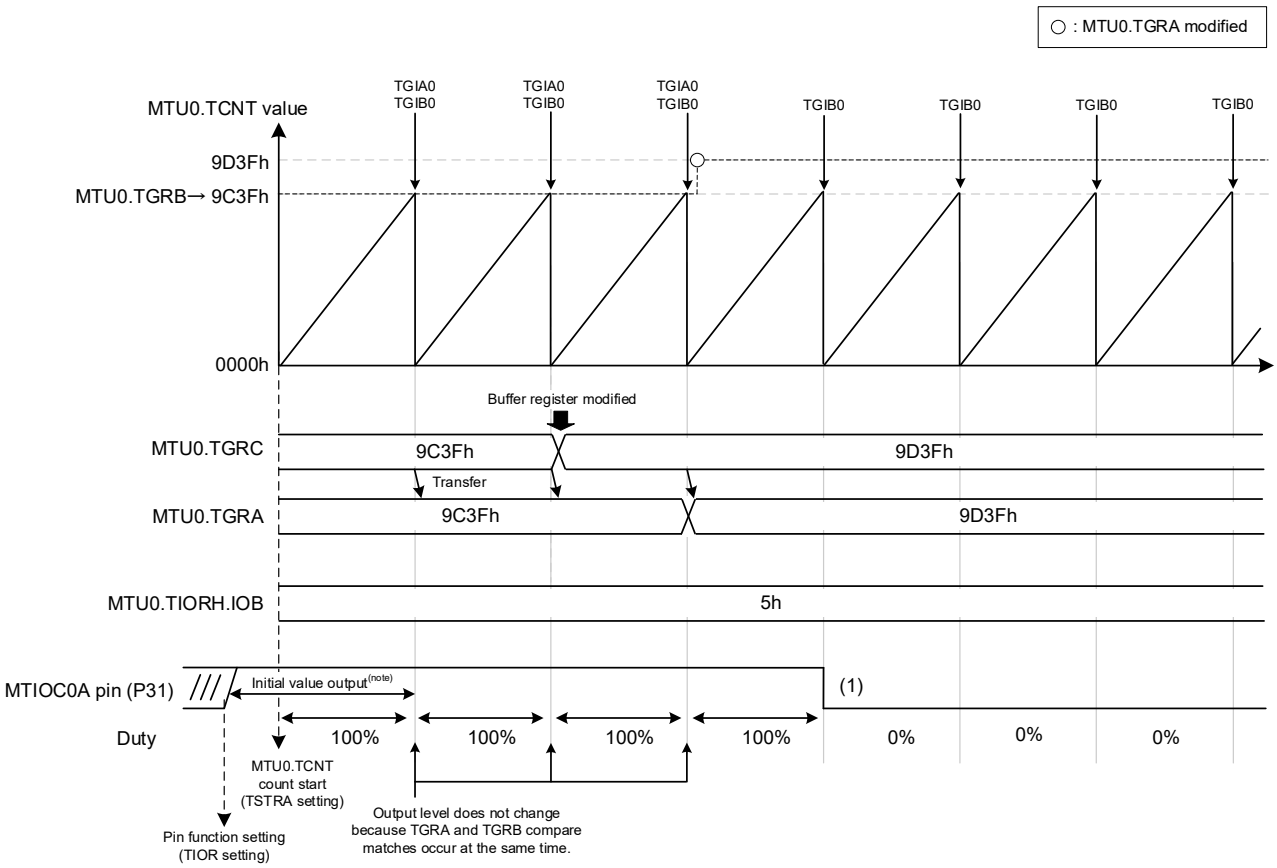
Figure 3.95 Change Duty Cycle to 100% After Holding 0% (without buffer)

3.13.5.5 Change Duty Cycle to 0% After Holding 100% at Initial Value High

The following shows an example of operations when the duty cycle is switched to 0% after outputting at 100% for several cycles.

The MITOC0A pin outputs an initial value high and outputs 100% duty cycle after operations start when TIORH.IOA is set to 6h and TIORH.IOB is set to 5h. When the TGRC is set to a value greater than the TGRB at any timing, low is output after 2 cycles and 0% duty cycle is output ((1) in figure below).

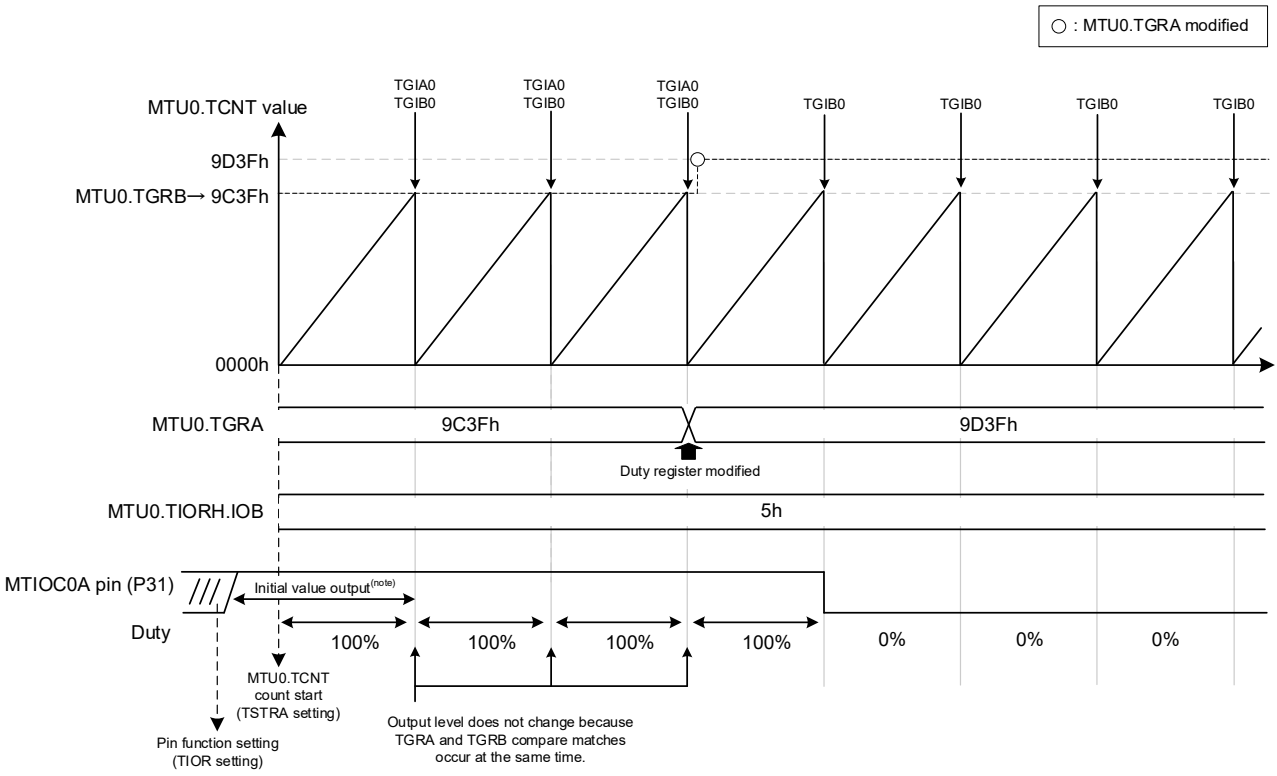
Also, active-high PWM can be output when TIORH.IOA is set to 2h at any timing after operations start.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.96 Change Duty Cycle to 0% After Holding 100% (with buffer)

The following shows an example of operations when the duty cycle is switched to 0% after outputting 100% for several cycles, as shown in Figure 3.96, when not using a buffer. The procedure is the same as using a buffer, but after a value greater than the TGRB is set to the TGRA, 0% duty cycle is output in the next cycle.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.97 Change Duty Cycle to 0% After Holding 100% (without buffer)

3.13.6 Usage Notes

In this sample code, the process to modify the TIOR register is performed when the MTU0.TGRB compare match interrupt (TGIB0) is generated. If the same process has been adapted to the user system, users should be aware of potential conflicts with other processes, interrupt delays, etc., and evaluate the product thoroughly before use.

3.13.6.1 TIOR Register Modify Delay

If the TIOR register modification could not be completed before the next TGRB compare match occurs, the desired duty cannot be output

3.13.6.2 Contention between TIOR Register Modify Operation and Compare Match

If a MTU0.TGRB compare match interrupt (TGIB0) is generated and the TIOR register is modified during the TIOR register modify operation, the desired duty cannot be output.

3.13.6.3 Usage Notes for Modifying TIOR Registers of MTU3, MTU4, MTU6, MTU7

When modifying the TIOR registers of MTU3, MTU4, MTU6 or MTU7 in PWM mode 1, set RWE in registers TRWERA and TRWERB to 1.

For details, refer to RX66T Group User's Manual: Hardware, section 22.2.20 Timer Read/Write Enable Registers (TRWERA, TRWERB).

3.13.6.4 Contention between Buffer Register Modify Operation and Compare Match

When the buffer transfer timing is set at compare match, the data in the buffer register before the modification is transferred to the duty register if a compare match occurs in the buffer register write cycle.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.7 Contention between Buffer Register Write Operation and Compare Match.

3.13.6.5 Switching from 0% duty cycle

When the TGRA is set to a value greater than the TGRB and 0% duty cycle is output, a TGRA compare match does not occur because the counter is cleared before the TGRA value is reached.

If buffer register TGRC is set to transfer a value to TGRA when a TGRA compare match occurs and the duty register is changed to switch from 0% duty, the value must be directly set in the TGRA register.

For details, refer to Figure 3.90.

3.13.6.6 Output for 1 Cycle When Duty Register is Set to 0

When the duty register is set to 0, waveforms are output for one cycle after a period register compare match occurs, and then a duty register compare match will occur.

When repeatedly switching between duty cycles 0% and 100% with each cycle, consider switching by changing the TIOR register, as done in the sample code.

3.13.6.7 Waveform does not Change When Duty Register is Set to Same Value as Period Register

When the duty register is set to same value as the period register, a compare match occurs for both registers at the same time and waveform does not change.

When repeatedly switching between duty cycles 0% and 100% with each cycle, consider switching by changing the TIOR register, as done in the sample code.

3.14 Complementary PWM Mode Duty 0% to 100%

- Target sample code file name: r01an5995_rx66t_mtu3_complementary_pwm_50to100.zip

3.14.1 Overview

The MTU complementary PWM mode can be used to output 3-phase complementary PWM waveforms with dead time.

This sample code describes a sample code that repeats the following output waveforms with dead time in complementary PWM mode 2 (transfer at trough), including duty cycles 0% and 100%.

- Duty switching: 50% → 80% → 100% → 80% → 50% → 0% → ・・・

The basic operation is to make changes to the duty cycle by transferring from the buffer register to the temporary register when a TCNT counter overflow occurs and from the temporary register to the compare register when an underflow occurs. The buffer register is modified when a period register MTU3.TGRA compare match occurs.

The following list provides the MTU settings used in the sample code.

- Use complementary PWM mode 2 (transfer at trough)
- Use channels 3 and 4
- Carrier period = 1ms
- Dead time= 50µs
- Timer count clock = 40MHz (PCLKC/4)
- Set MTU3.TGRA to MTU3.TCNT upper limit value
  (1/2 carrier period + dead time)
  — MTIOC3A pin toggle output setting
- Set buffer transfer timing
  — Transfer at trough of the count
- Initial output value is low, active level is high
- Use MTU3.TGRB as U-phase duty register
  — Positive-phase:
  High output at up-counting compare match
  Low output at down-counting compare match
  — Negative-phase:
  — Low output at up-counting compare match
  High output at down-counting compare match
- Use MTU4.TGRA as V-phase duty register
  — Positive-phase:
  High output at up-counting compare match
  Low output at down-counting compare match
  — Negative-phase:
  — Low output at up-counting compare match
  High output at down-counting compare match
- Use MTU4.TGRB as W-phase duty register
  — Positive-phase:
  — High output at up-counting compare match
  — Low output at down-counting compare match
  — Negative-phase:
  — Low output at up-counting compare match
  — High output at down-counting compare match
- Use buffer register
  — Use MTU3.TGRD as buffer register of MTU3.TGRB
  — Use MTU4.TGRC as buffer register of MTU4.TGRA
  — Use MTU4.TGRD as buffer register of MTU4.TGRB
  — Refer to Figure 3.100 for buffer register initial values
- Change duty cycle for each period
  — Change duty cycle at MTU3.TGRA compare match interrupt
  — Refer to Figure 3.100 for details on timing for duty cycle changes

Set in Smart Configurator.
For Setting Methods,
refer to section 3.14.3.

Complementary PWM mode output for this sample code is shown below.

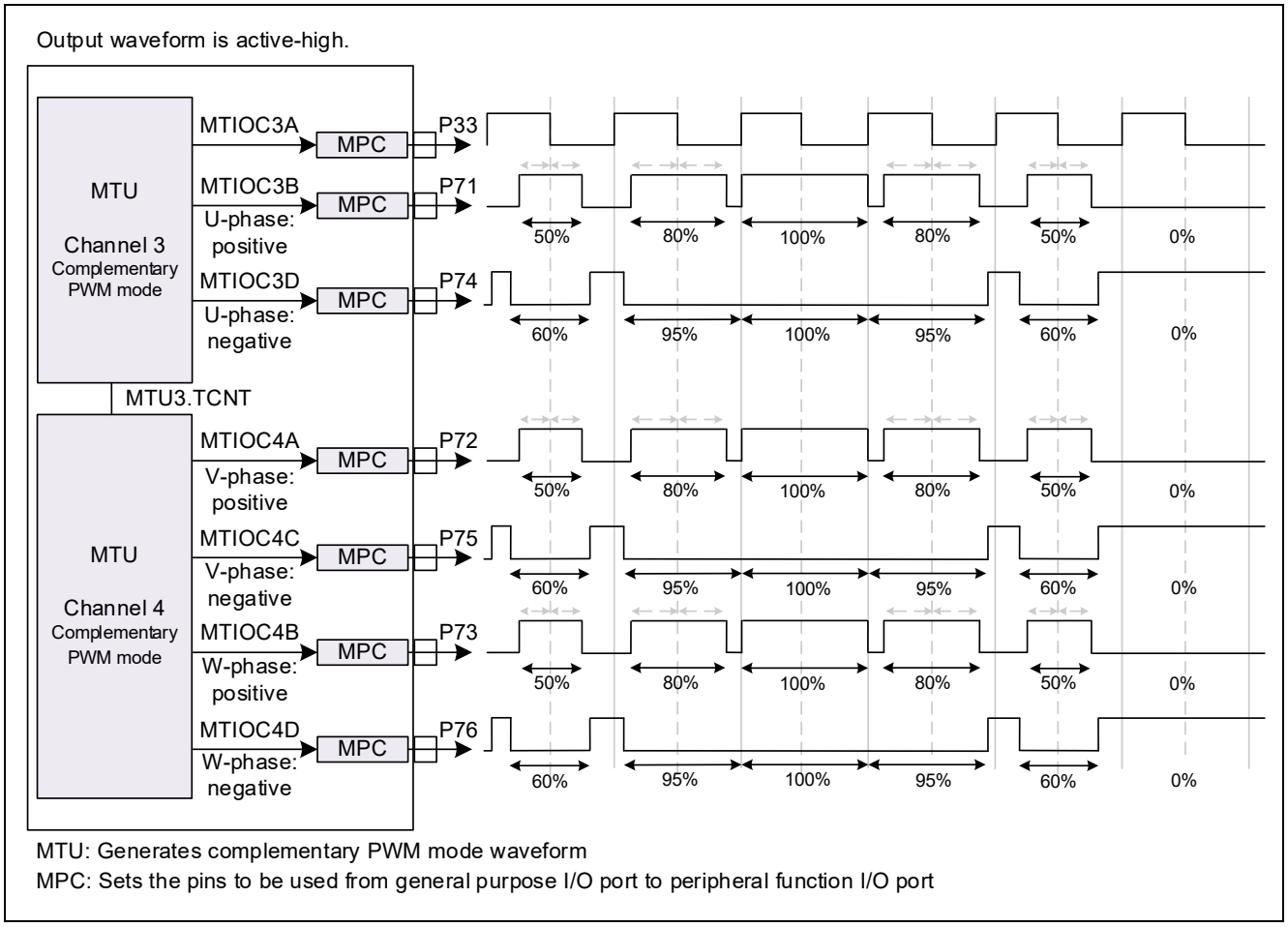


Figure 3.98 Complementary PWM Mode Output

3.14.2 Operation Details

The sample code operations are shown in Figure 3.100. The settings of the duty cycle are changed with each period by modifying the values of the buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD) at the compare match interrupt (TGIA3) of the MTU3.TGRA register, which is set to the upper limit value of MTU3.TCNT ((1) in Figure 3.100).

Writing MTU4.TGRD at the end of the buffer register modification enables data transfer from the buffer register to the temporary register. When MTU4.TGRD is written in the Ta interval, the data written to the buffer register is immediately transferred to the temporary register, but in this sample code, MTU4.TGRD is written in the Tb1 interval, so the data is transferred to the temporary register after the Tb1 interval, ends ((2) in Figure 3.100).

Since this sample code uses complementary PWM mode 2 for trough transfers, the compare register updates the data by transferring from the temporary register at the end of the Tb2 interval ((3) in Figure 3.100).

Initial output is OFF for both positive-phase output and negative-phase output according to the settings of TOCR2A bits OLS1P, OLS1N, OLS2P, OLS2N, OLS3P, and OLS3N. After setting the complementary PWM mode in MTU3.TMDR1, output continues until MTU4.TCNT is greater than the value of the TDDRA register ((4) in Figure 3.100).

The duty cycle in each period generates the 1/2 duty cycle for the up-counting period and down-counting periods. The duty cycles generated in the up-counting and down-counting periods are shown in Figure 3.99.

- 100% Duty Cycle Output
  When outputting 100% duty cycle, set 0000h at the buffer register modify timing one cycle before in order to make the compare register value 0000h.
  The compare and temporary registers are enabled in the Tb2 interval and the positive-phase turns ON when a compare match occurs between the temporary and counter registers ((5) in Figure 3.100). Because the negative-phase does not turn OFF when both compare matches occur, low output is retained from the previous cycle ((6) in Figure 3.100).

- 0% Duty Cycle Output
  When outputting 0% duty cycle, set the buffer register to the same value as MTU3.TGRA at the buffer modify timing one cycle before in order to make the compare register the same value as MTU3.TGRA. The following compare matches (a, b, c and d) occur, but the waveform does not change. The positive-phase wave retains low output and the negative-phase wave retains high output ((7) in Figure 3.100).
  — For compare match c, a compare match occurs between the compare register and the counter register, and the positive-phase turns OFF, but the waveform does not change since the phase goes from OFF to OFF.
  — For compare matches a and d, compare matches that turn the negative-phase output ON/OFF occur at the same time. However, when compare matches that turn output ON/OFF for the same phase occur at the same time, both compare matches are ignored and the waveform does not change.
  — For compare match b, since compare match c, which turns output OFF, occurs in the same section, b is ignored, and the waveform does not change.

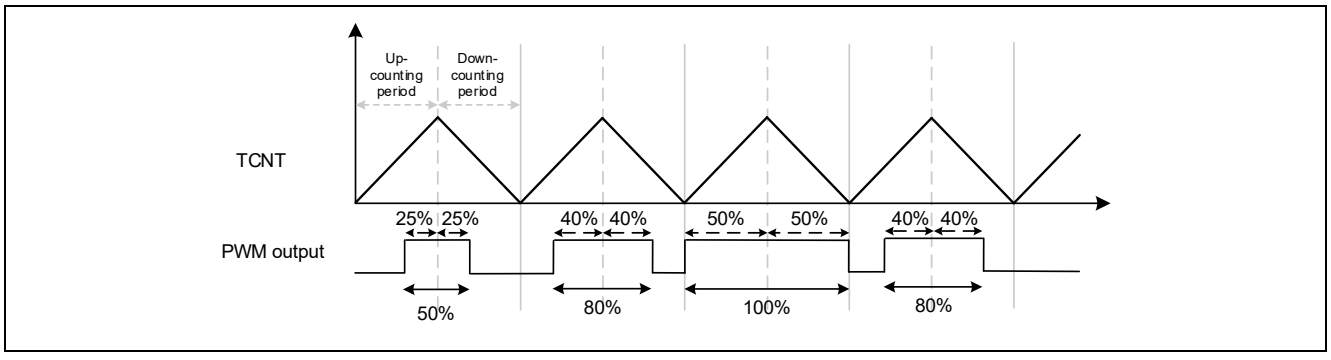
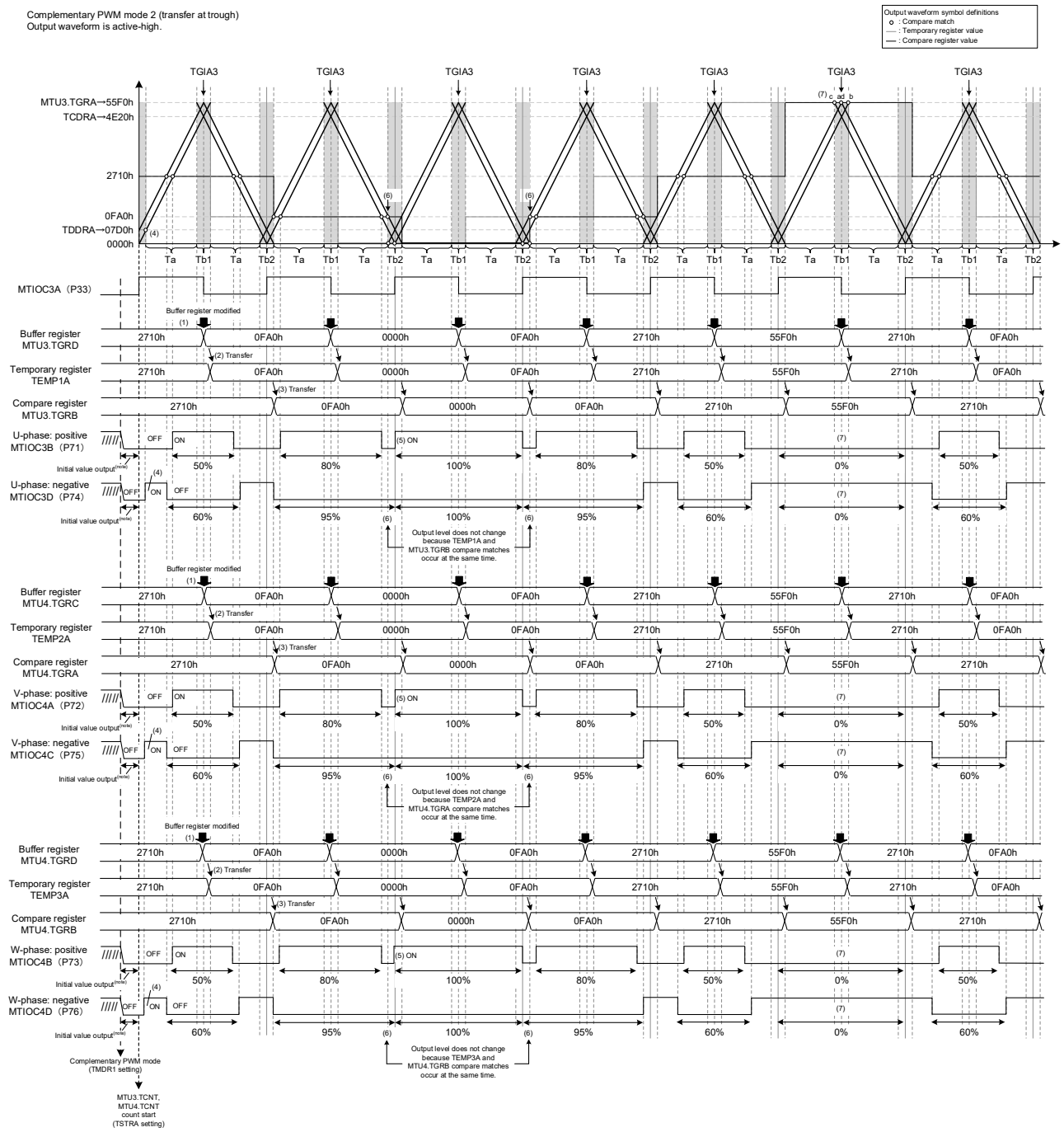


Figure 3.99 Symmetric PWM Output Waveform



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.100 Sample Code Operations

3.14.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table3.17 Adding Components

Item	Description
Component	Complementary PWM Mode Timer
Configuration name	Config_MTU3_MTU4
Operation	Complementary PWM Mode 2 (transfer at trough)
Resource	MTU3_MTU4

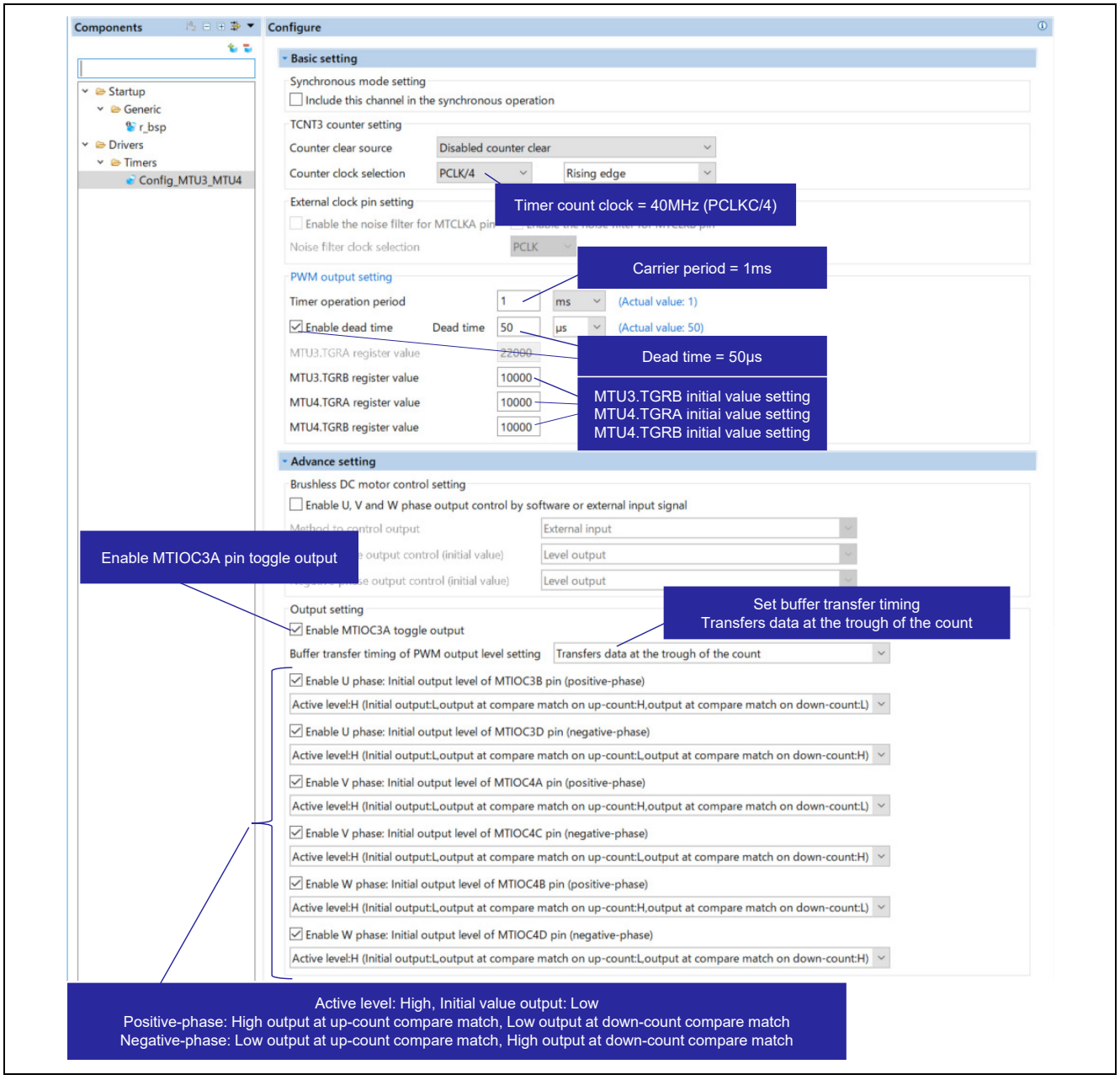


Figure 3.101 MTU3 and MTU4 Settings (1/2)

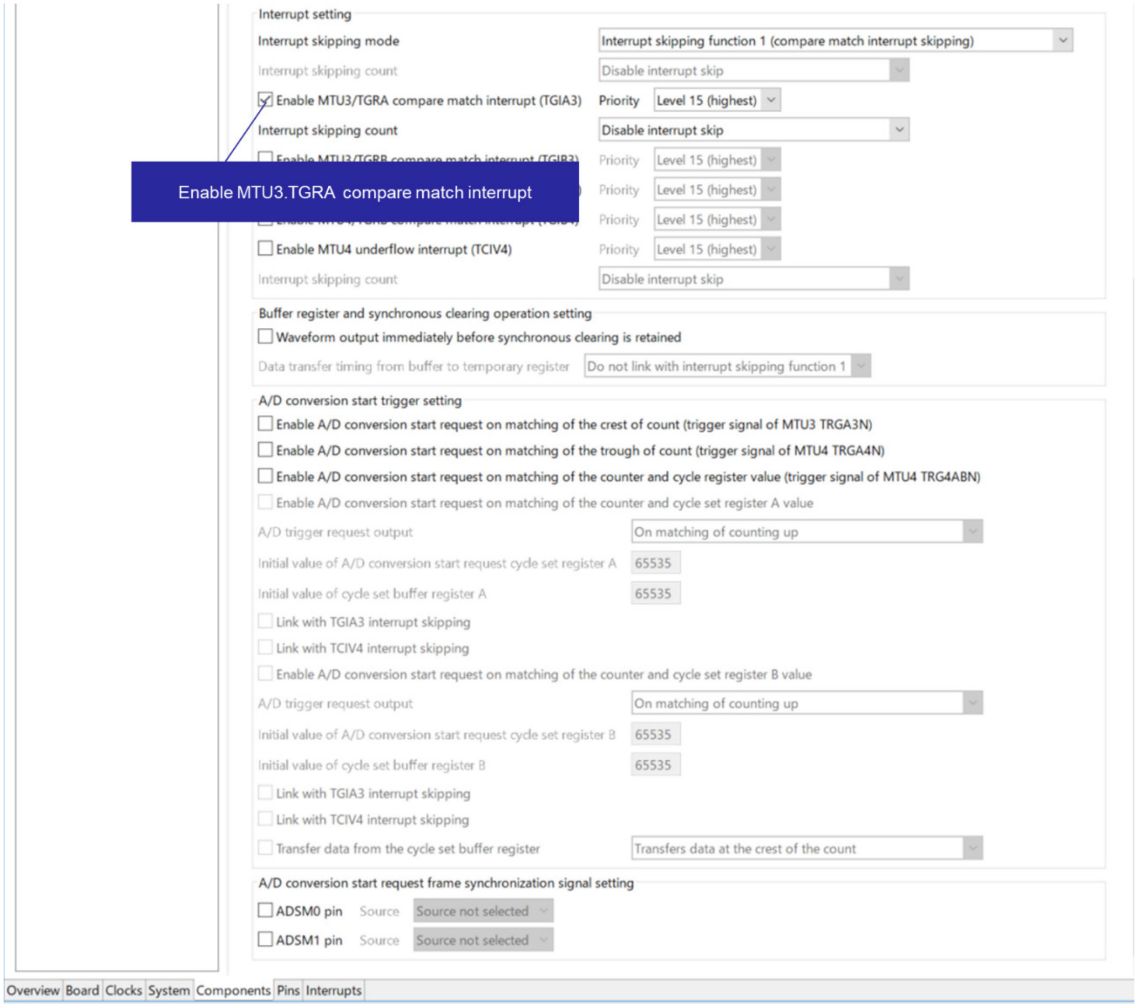


Figure 3.102 MTU3 and MTU4 Settings (2/2)

3.14.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

MTU3.TCNT and MTU4.TCNT counting is started in the main function.

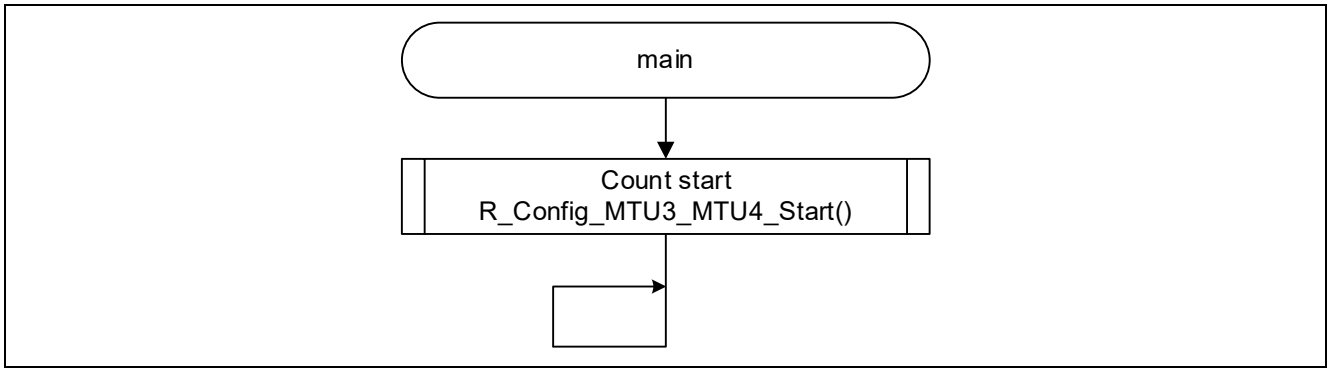


Figure 3.103 main Function

User initialization function R_Config_MTU0_Create_UserInit, which is executed before the main function, initializes the variables. This function is called from within the R_Config_MTU0_Create function.

This function initializes the following variable used in this sample code.

- s_duty_list_counter: counter variable for reading from the duty cycle list

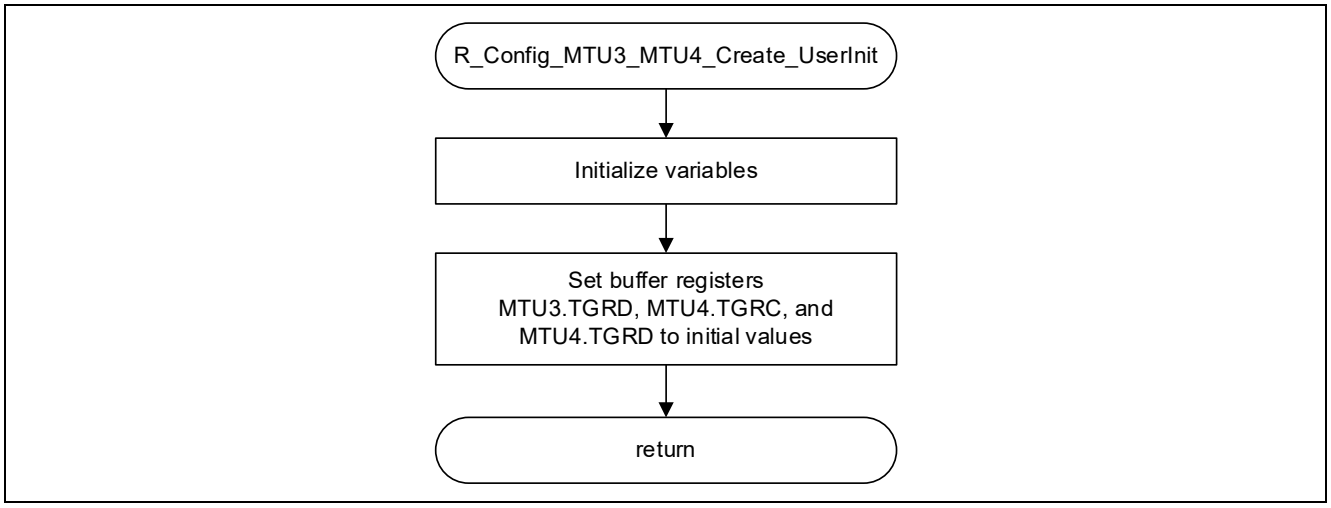


Figure 3.104 User Initialization Function

The TGIA3 interrupt handler function changes the values of the next buffer registers (MTU3.TGRD, MTU4.TGRC and MTU4.TGRD) sequentially according to the values read from the duty cycle list.

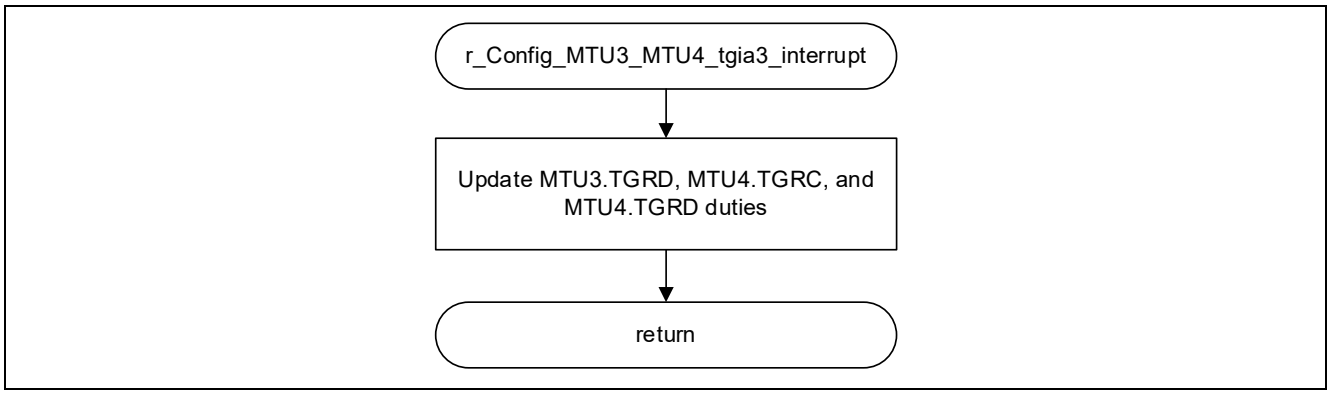


Figure 3.105 TGIA3 Interrupt Handler Function

3.14.5 Usage Notes

3.14.5.1 Pin Settings

The MTIOC3C and MTIOC6C pins cannot be used as the timer input/output pins in the complementary PWM mode. Set them to I/O ports.

For details, refer to RX66T Group User's Manual: Hardware, section 22.3.8 Complementary PWM Mode, Table 22.74 Output Pins for Complementary PWM Mode.

3.14.5.2 Buffer Register Value Updating

When modifying the data in the buffer register, be sure to modify to MTU4.TGRD (MTU7.TGRD) last. Data is transferred from the buffer register to all five temporary registers simultaneously after the write to MTU4.TGRD (MTU7.TGRD).

Even when not updating all five registers or when not updating the data in MTU4.TGRD (MTU7.TGRD), be sure to write to MTU4.TGRD (MTU7.TGRD) after writing data to the registers to be updated. In this case, the data to be written to MTU4.TGR (MTU7.TGRD) should be the same as the data prior to the write operation.

3.14.5.3 Buffer Operation Settings

In complementary PWM mode, use the buffer operation to modify the PWM period setting registers (MTU3.TGRA and MTU6.TGRA), timer period data registers (TCDRA and TCDRB), and duty setting registers (MTU3.TGRB, MTU4.TGRA, MTU4.TGRB, MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB).

The MTIOC4C (MTIOC7C) and MTIOC4D (MTIOC7D) pins cannot output waveforms, when the buffer operation bits MTU4.TMDR1.BFA (MTU7.TMDR1.BFA), MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) are set to 1. Set MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) and MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) bits to 0.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.14 Buffer Operation Setting in Complementary PWM Mode.

3.14.5.4 Output Level Settings

When MTU3 and MTU4 (or MTU6 and MTU7) are in the complementary PWM mode, PWM waveform output level is determined by the TOCR1A.OLSP, TOCR1A.OLSN, TOCR1B.OLSP, and TOCR1B.OLSN bits, and the TOCR2A.OLSnP, TOCR2A.OLSnN, TOCR2B.OLSnP, and TOCR2B.OLSnN (n = 0 to 3) bits. Set the TIOR register to 00h.

If the TDERA.TDER (TDERB.TDER) bit is set to 0 (dead time is not generated) in the complementary PWM mode, the negative-phase output is the inverted level of the positive-phase output according to the settings of the TOCR1A.OLSP (TOCR1B.OLSP) and TOCR2A.OLSnP (TOCR2B.OLSnP) (n = 0 to 3) bits, and does not depend on the settings of the TOCR1A.OLSN (TOCR1B.OLSN) and TOCR2A.OLSnN (TOCR2B.OLSnN) (n = 0 to 3) bits.

When dead time is not generated, if only the output of the negative-phase side is enabled and the output of the positive-phase side is prohibited in the TOER register, the negative-phase side does not be output.

For details, refer to RX66T Group User's Manual: Hardware, section 22.2.22 Timer Output Control Registers 1 (TOCR1A, TOCR1B) and section 22.2.23 Timer Output Control Registers 2 (TOCR2A, TOCR2B).

3.15 Complementary PWM Mode Duty Cycles 0% and 100%

- Target sample code file name: r01an5995_rx66t_mtu3_complementary_pwm_0to100.zip

3.15.1 Overview

The MTU complementary PWM mode can be used to output 3-phase complementary PWM waveforms with dead time.

This sample code describes a sample code that uses complementary PWM mode 2 (transfer at trough) and repeats waveform output alternating between duty cycles 0% and 100%.

The basic operation is to make changes to the duty cycle by transferring from the buffer register to the temporary register when a TCNT counter overflow occurs, and from the temporary register to the compare register when an underflow occurs. The buffer register is modified when a period register MTU3.TGRA compare match occurs.

The following list provides the MTU settings used in the sample code.

- Use complementary PWM mode 2 (transfer at trough)
- Use channels 3 and 4
- Carrier period = 1ms
- Dead time = 50µs
- Timer count clock = 40MHz (PCLKC/4)
- Set MTU3.TGRA to MTU3.TCNT upper limit value
  (1/2 carrier period + dead time)
  — MTIOC3A pin toggle output setting
- Set buffer transfer timing
  — Transfers data at the trough of the count
- Initial output value is low, active level is high
- Use MTU3.TGRB as U-phase duty register
  — Positive-phase:
  High output at up-counting compare match
  Low output at down-counting compare match
  — Negative-phase:
  Low output at up-counting compare match
  High output at down-counting compare match
- Use MTU4.TGRA as V-phase duty register
  — Positive-phase:
  High output at up-counting compare match
  Low output at down-counting compare match
  — Negative-phase:
  Low output at up-counting compare match
  High output at down-counting compare match
- Use MTU4.TGRB as W-phase duty register
  — Positive-phase:
  High output at up-counting compare match
  Low output at down-counting compare match
  — Negative-phase:
  Low output at up-counting compare match
  High output at down-counting compare match
- Use buffer register
  — Use MTU3.TGRD as buffer register of MTU3.TGRB
  — Use MTU4.TGRC as buffer register of MTU4.TGRA
  — Use MTU4.TGRD as buffer register of MTU4.TGRB
  — Refer to Figure 3.107 for buffer register initial values
- Change duty cycle for each period
  — Change duty cycle at MTU3.TGRA compare match interrupt
  — Refer to Figure 3.107 for details on timing for duty cycle changes

Set in Smart Configurator.
For setting methods,
refer to section 3.15.3.

Complementary PWM mode output for this sample code is shown below.

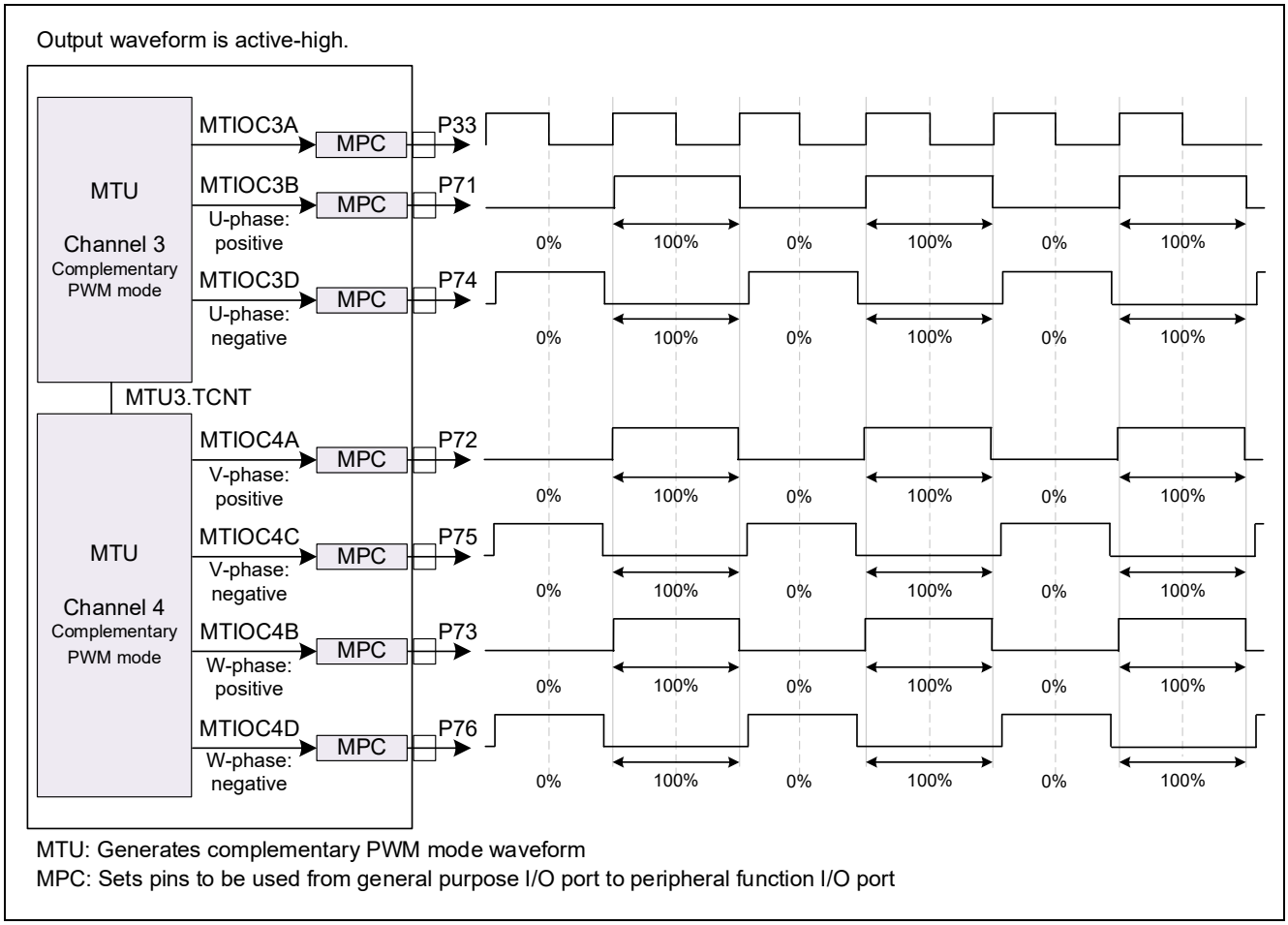


Figure 3.106 Complementary PWM Mode Output

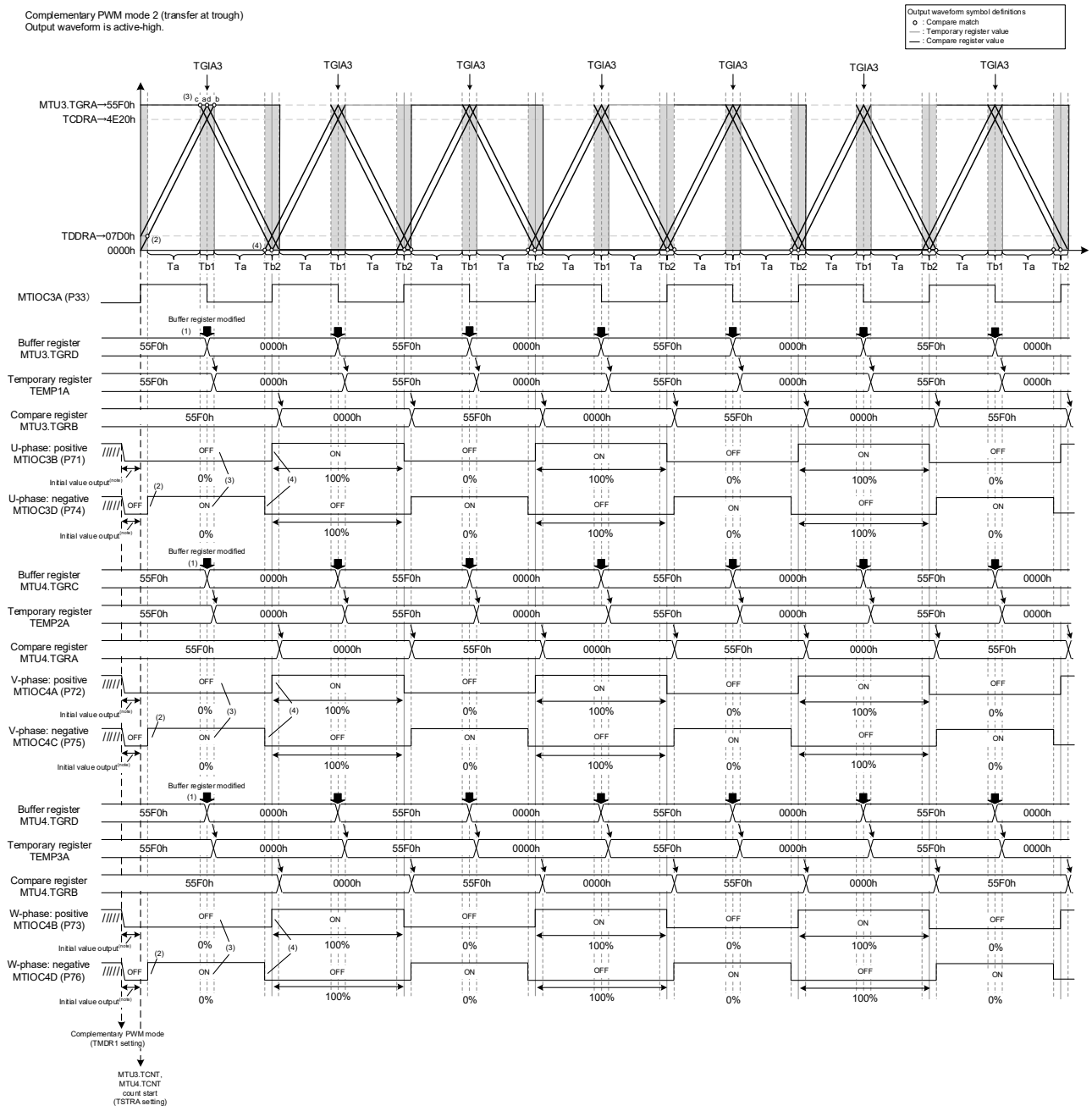
3.15.2 Operation Details

The sample code operations are shown in Figure 3.107. The duty cycle is switched between 0% and 100% with each period by modifying the values of the buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD) at the compare match interrupt (TGIA3) of register MTU3.TGRA, which is set in the upper limit value of MTU3.TCNT ((1) in Figure 3.107).

Initial output is OFF for both positive-phase output and negative-phase output, according to the settings of TOCR2A bits OLS1P, OLS1N, OLS2P, OLS2N, OLS3P, and OLS3N. After setting complementary PWM mode with MTU3.TMDR1, output is continued until MTU4.TCNT is greater than the value of the TDDRA register ((2) in Figure 3.107).

- 0% Duty Cycle Output
  When outputting 0% duty cycle, set the buffer register to the same value as MTU3.TGRA in order to make the compare register the same value as MTU3.TGRA.
  The negative-phase ON period is shorter than the positive-phase OFF period by the dead time.
  The following compare matches (a, b, c and d) occur, but the waveforms do not change. The positive-phase wave retains low output and the negative-phase wave retains high output ((3) in Figure 3.107).
  — For compare match c, a compare match occurs between the compare register and the counter register, and the positive-phase turns OFF, but the waveform does not change since the phase goes from OFF to OFF.
  — When compare matches that turn the negative-phase output ON/OFF occur at the same time, both compare matches are ignored and the waveform does not change.
  — For compare match b, since compare match c, which turns output OFF, occurs in the same section, b is ignored, and the waveform does not change.

- 100% Duty Cycle Output
  When outputting 100% duty cycle, set 0000h at the buffer register modify timing one cycle before to make the compare register value 0000h.
  The compare and temporary registers are enabled in the Tb2 interval. When a compare match occurs between the temporary and counter registers, the positive-phase goes high and the negative-phase goes low ((4) in Figure 3.107).
  The negative-phase OFF period is longer than the positive-phase ON period by the dead time.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 3.107 Sample Code Operations

3.15.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU as described below. For details on how to add components, refer to section 3.1.4 Adding Components.

Table 3.18 Adding Components

Item	Description
Component	Complementary PWM Mode Timer
Configuration name	Config_MTU3_MTU4
Operation	Complementary PWM Mode 2 (transfer at trough)
Resource	MTU3_MTU4

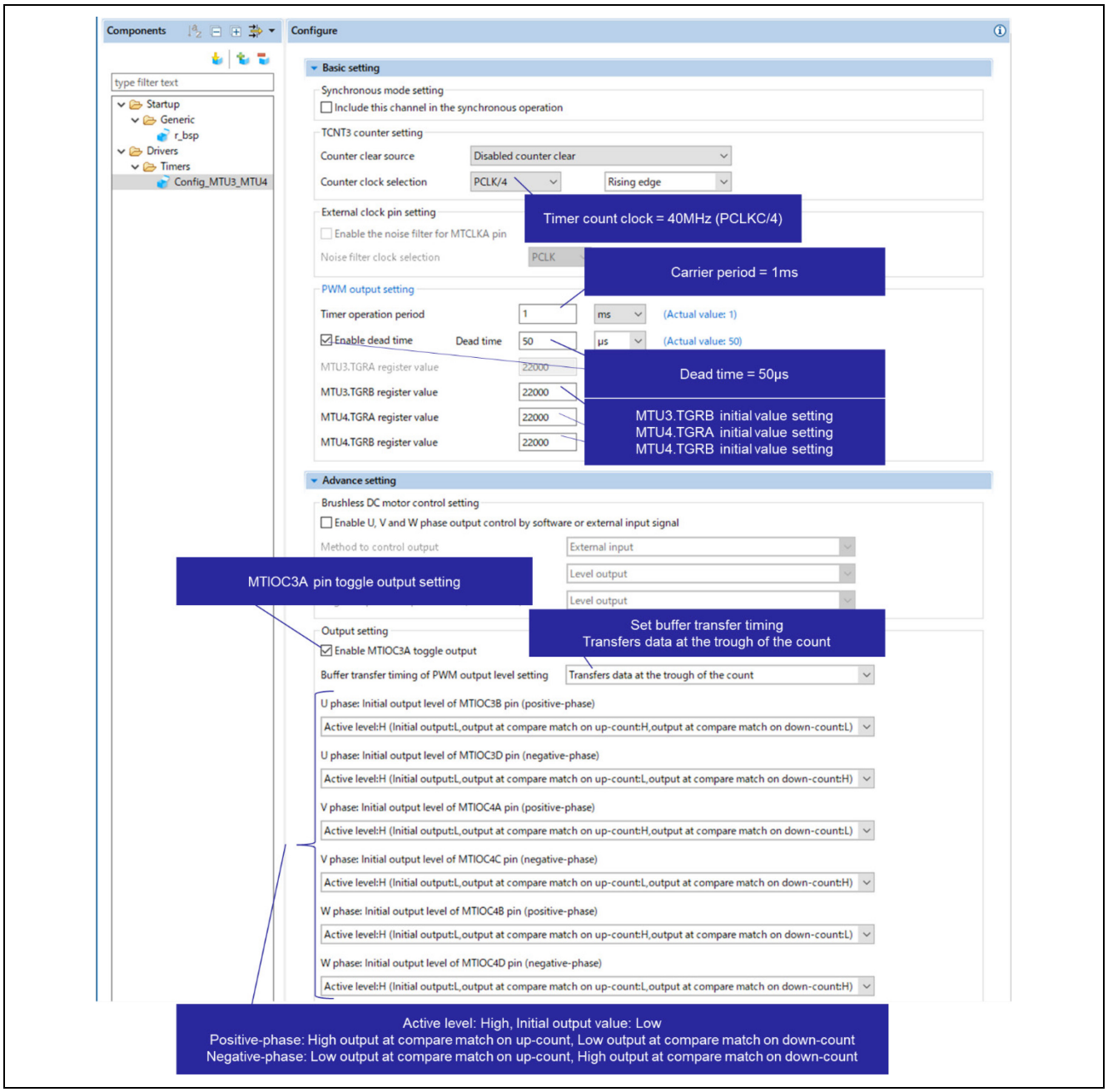


Figure 3.108 MTU3 and MTU4 Settings (1/2)

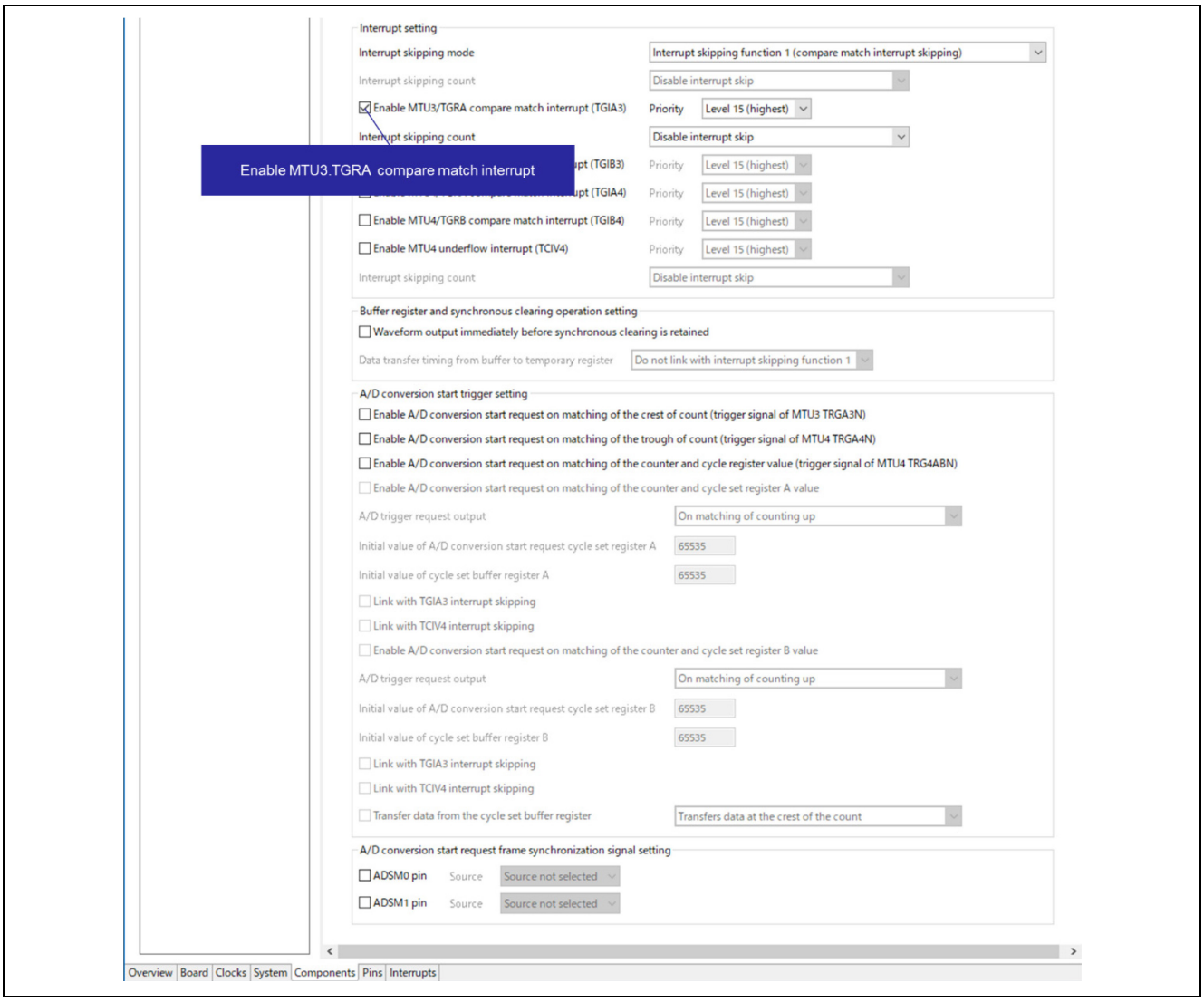


Figure 3.109 MTU3 and MTU4 Settings (2/2)

3.15.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

MTU3.TCNT and MTU4.TCNT counting is started in the main function.

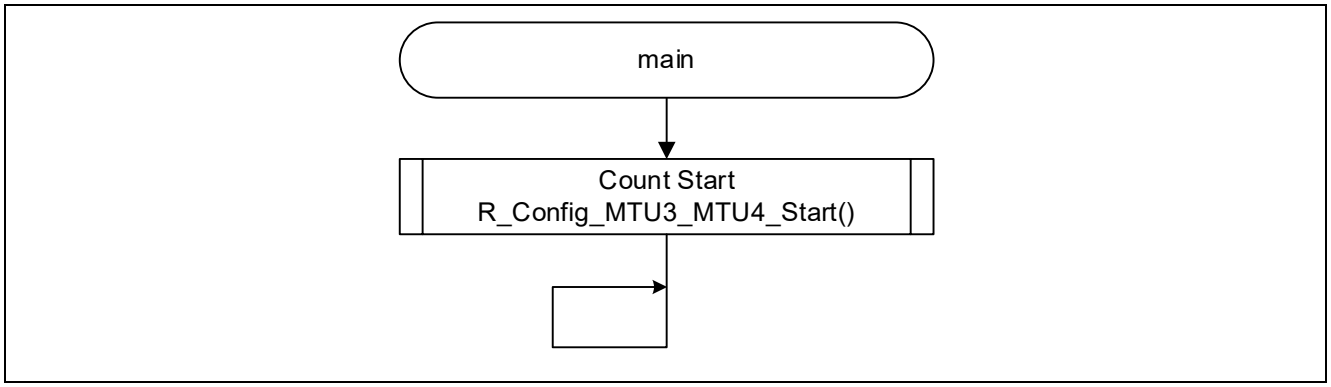


Figure 3.110 main Function

The TGIA3 interrupt handler function changes the values of the next buffer registers to be set according to the duty cycle set in the buffer registers (MTU3.TGRD, MTU4.TGRC, MTU4.TGRD).

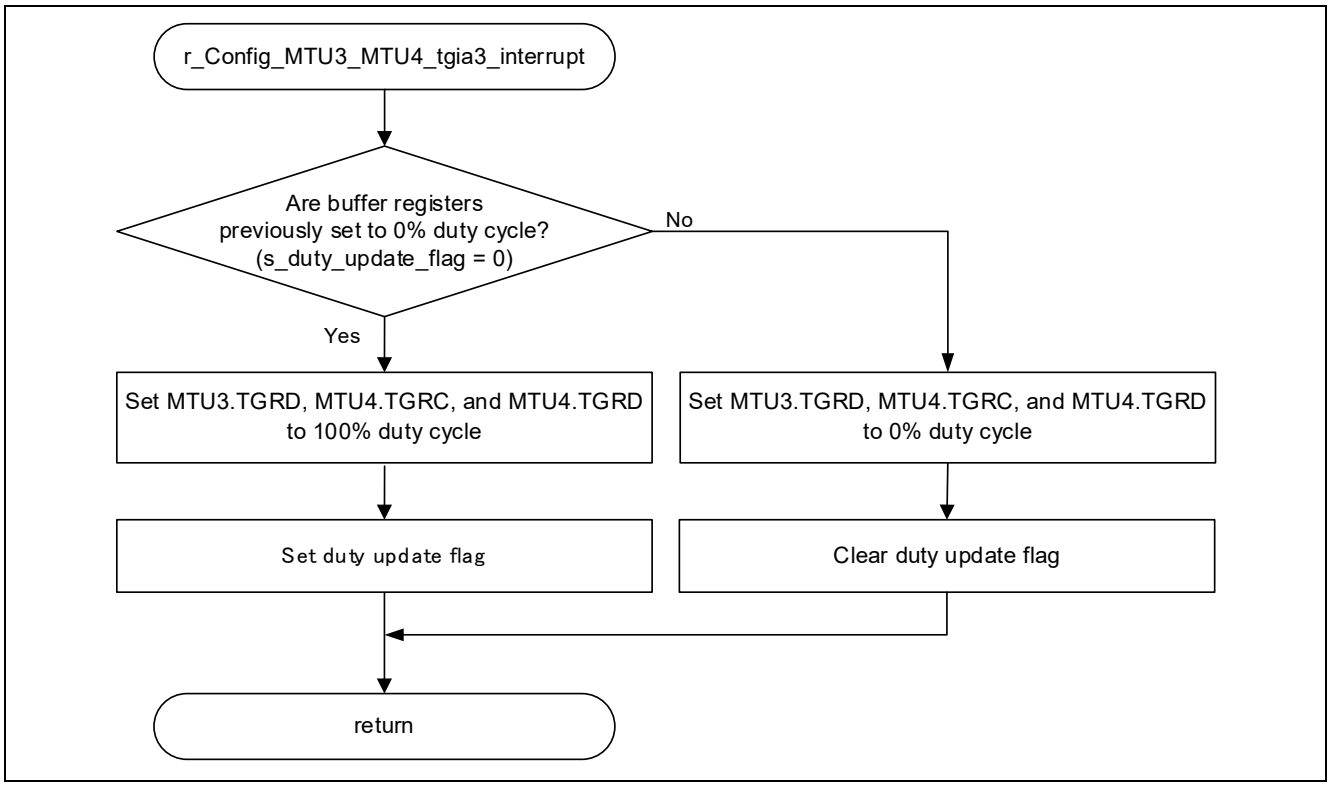


Figure 3.111 TGIA3 Interrupt Handler Function

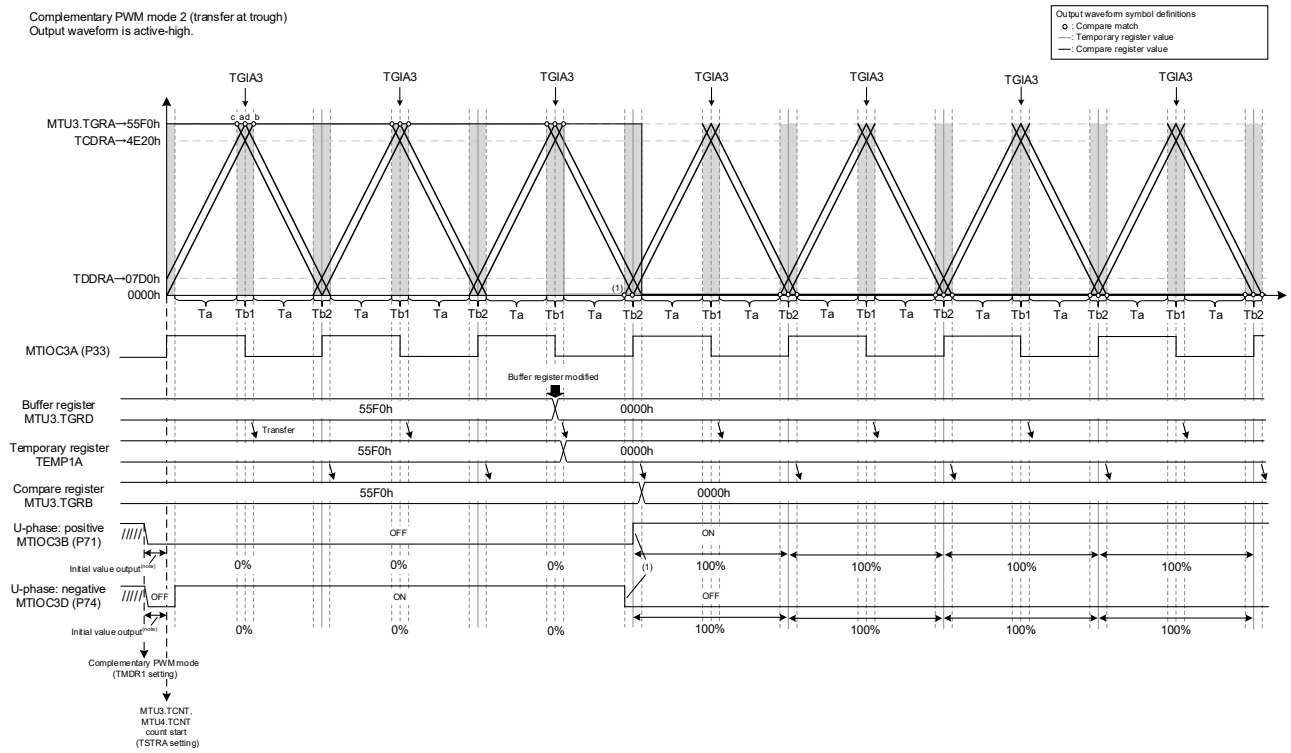
3.15.5 Related Operations

3.15.5.1 Change Duty Cycle to 100% After Holding 0% at Initial Value Low

The following shows an example of operations when the duty cycle is switched to 100% after duty has been output at 0% for several cycles.

The compare and temporary registers are enabled in the Tb2 interval. When a compare match occurs between the temporary and counter registers, negative-phase output goes low and then positive-phase output goes high ((1) in figure below).

The negative-phase OFF period is longer than positive-phase ON period by the dead time.



Note: The waveform in the sample code starts outputting initial values when the PMR register is set.

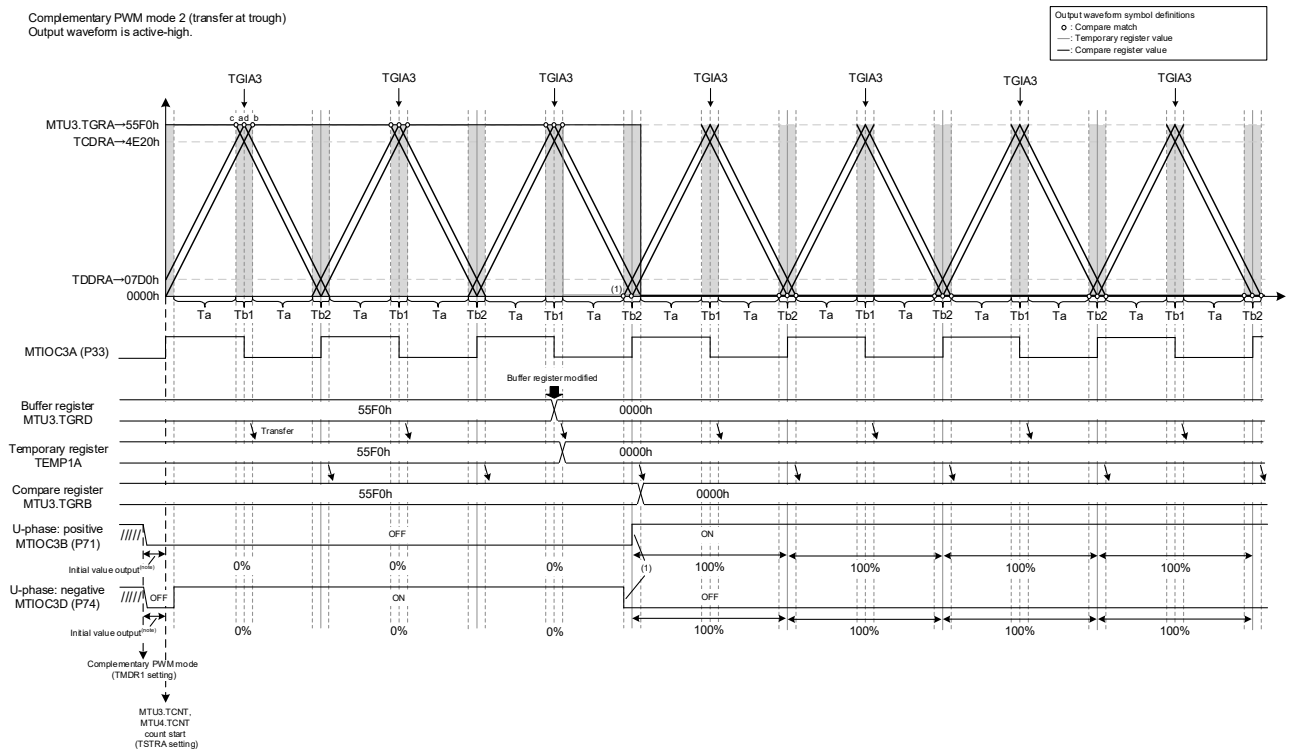
Figure 3.112 Change Duty Cycle to 100% After Holding 0%

3.15.5.2 Change Duty Cycle to 0% After Holding 100% at Initial Value Low

The following shows an example of operations when the duty cycle is switched to 0% after duty has been output at 100% for several cycles.

The compare and temporary registers are enabled in the Tb2 interval. When a compare match occurs between the compare and counter registers, positive-phase output goes low and then negative-phase output goes high ((1) in figure below).

The negative-phase ON period is shorter than the positive-phase OFF period by the dead time.



Note: The waveform in the sample code starts outputting initial values when the PMR register is set.

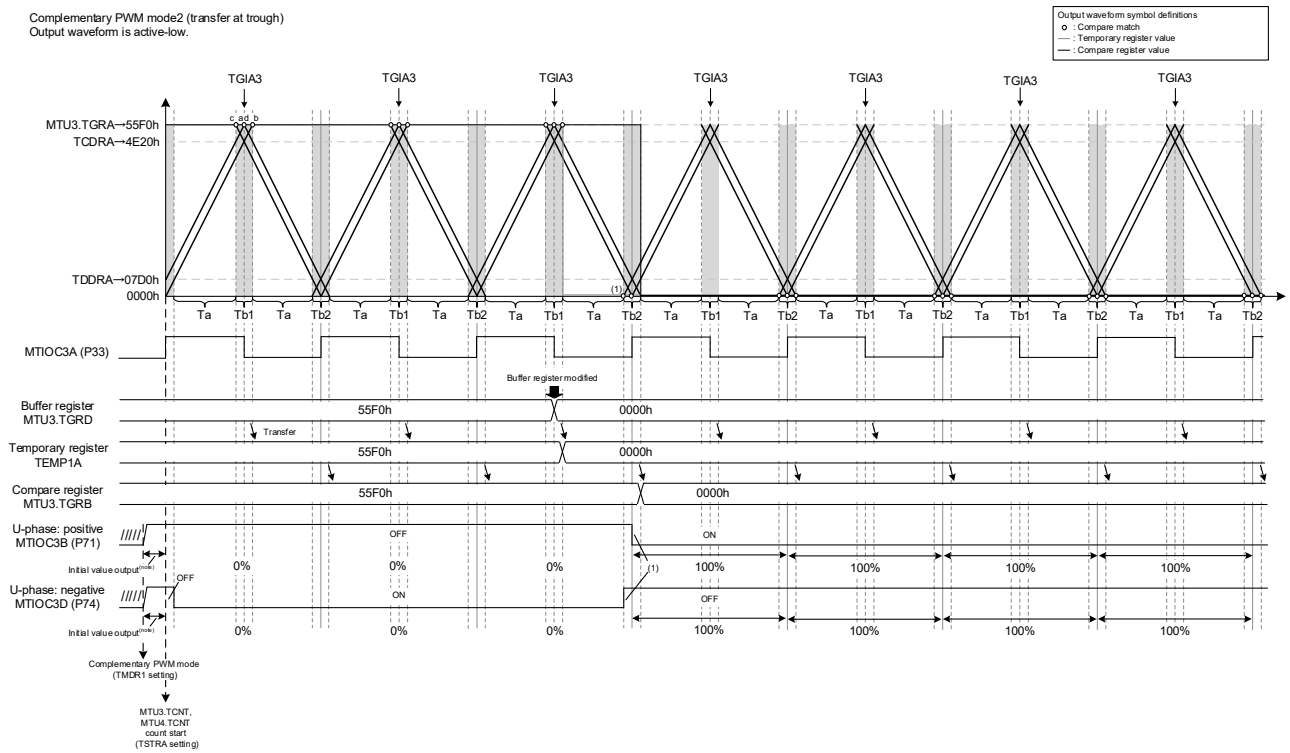
Figure 3.113 Change Duty Cycle to 0% After Holding 100%

3.15.5.3 Change Duty Cycle to 100% After Holding 0% at Initial Value High

The following shows an example of operations when the duty cycle is switched to 100% after duty has been output at 0% for several cycles.

The compare and temporary registers are enabled in the Tb2 interval. When a compare match occurs between the temporary and counter registers, negative-phase output goes high and then positive-phase output goes low ((1) in figure below).

The negative-phase OFF period is longer than the positive-phase ON period by the dead time.



Note: The waveform in the sample code starts outputting initial values when the PMR register is set.

Figure 3.114 Change Duty Cycle to 100% After Holding 0%

To make the initial value high, set “initial output level” of each pin to “active level: L”. In complementary PWM mode, when the initial output is high, the active level goes to low.

For details, refer to RX66T Group User’s Manual: Hardware, section 22.2.23 Timer Output Control Registers 2 (TOCR2A, TOCR2B).

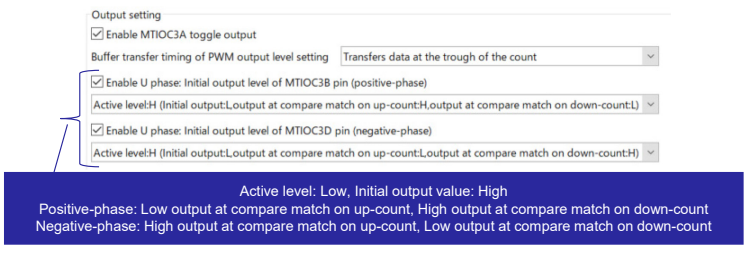


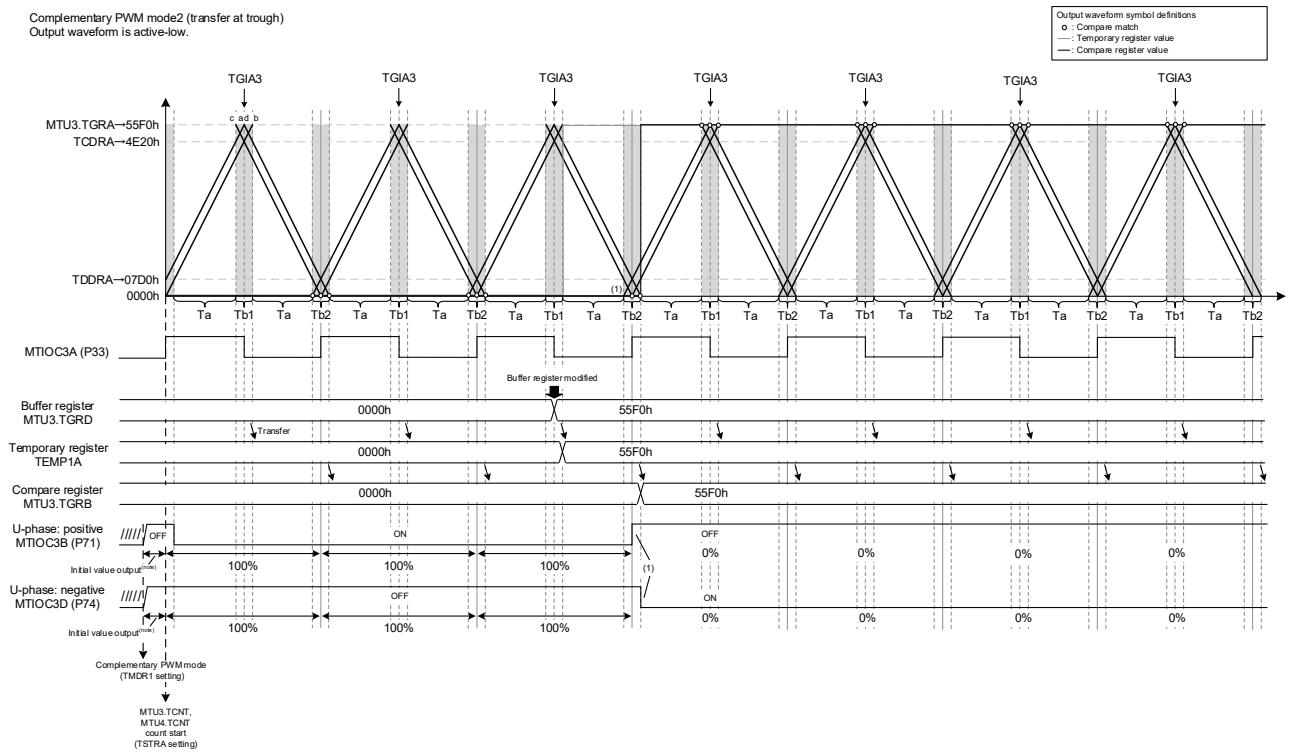
Figure 3.115 Smart Configurator Setting for Holding Value High

3.15.5.4 Change Duty Cycle to 0% After Holding 100% at Initial Value High

The following shows an example of operations when the duty cycle is switched to 0% after duty has been output at 100% for several cycles.

The compare and temporary registers are enabled in the Tb2 interval. When a compare match occurs between the compare and counter registers, positive-phase output goes high and then negative-phase output goes low ((1) in figure below).

The negative-phase ON period is shorter than the positive-phase OFF period by the dead time.



Note: The waveform in the sample code starts outputting initial values when the PMR register is set.

Figure 3.116 Change Duty Cycle to 0% After Holding 100%

To make the initial value high, set the “initial output level” of each pin to “active level: L”. For details, refer to Figure 3.115.

3.15.6 Usage Notes

3.15.6.1 Pin Settings

The MTIOC3C and MTIOC6C pins cannot be used as the timer input/output pins in the complementary PWM mode. Set them to I/O ports.

For details, refer to RX66T Group User's Manual: Hardware, section 22.3.8 Complementary PWM Mode, Table 22.74 Output Pins for Complementary PWM Mode.

3.15.6.2 Buffer Register Value Updating

When modifying the data in the buffer register, be sure to modify to MTU4.TGRD (MTU7.TGRD) last. Data is transferred from the buffer register to all five temporary registers simultaneously after the write to MTU4.TGRD (MTU7.TGRD).

Even when not updating all five registers or when not updating the data in MTU4.TGRD (MTU7.TGRD), be sure to write to MTU4.TGR (MTU7.TGRD) after writing data to the registers to be updated. In this case, the data to be written to MTU4.TGRD (MTU7.TGRD) should be the same as the data prior to the write operation.

3.15.6.3 Buffer Operation Settings

In complementary PWM mode, use the buffer operation to modify the PWM period setting registers (MTU3.TGRA and MTU6.TGRA) and timer period data registers (TCDRA and TCDRB), and duty setting registers (MTU3.TGRB, MTU4.TGRA, MTU4.TGRB, MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB).

The MTIOC4C (MTIOC7C) and MTIOC4D (MTIOC7D) pins cannot output waveforms, when the buffer operation bits MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) and MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) are set to 1. Set MTU4.TMDR1.BFA (MTU7.TMDR1.BFA) and MTU4.TMDR1.BFB (MTU7.TMDR1.BFB) bits to 0.

For details, refer to RX66T Group User's Manual: Hardware, section 22.6.14 Buffer Operation Setting in Complementary PWM Mode.

3.15.6.4 Output Level Settings

When MTU3 and MTU4 (or MTU6 and MTU7) are in the complementary PWM mode, PWM waveform output level is determined by the TOCR1A.OLSP, TOCR1A.OLSN, TOCR1B.OLSP, and TOCR1B.OLSN bits, and the TOCR2A.OLSnP, TOCR2A.OLSnN, TOCR2B.OLSnP, and TOCR2B.OLSnN (n = 0 to 3) bits. Set the TIOR register to 00h.

If the TDERA.TDER (TDERB.TDER) bit is set to 0 (dead time is not generated) in the complementary PWM mode, the negative-phase output is the inverted level of the positive-phase output according to the settings of the TOCR1A.OLSP (TOCR1B.OLSP) and TOCR2A.OLSnP (TOCR2B.OLSnP) (n = 0 to 3) bits, and does not depend on the settings of the TOCR1A.OLSN (TOCR1B.OLSN) and TOCR2A.OLSnN (TOCR2B.OLSnN) (n = 0 to 3) bits.

When dead time is not generated, if only the output of the negative-phase side is enabled and the output of the positive-phase side is prohibited in the TOER register, the negative-phase side does not be output.

For details, refer to RX66T Group User's Manual: Hardware, section 22.2.22 Timer Output Control Registers 1 (TOCR1A, TOCR1B) and section 22.2.23 Timer Output Control Registers 2 (TOCR2A, TOCR2B).

4. GPTW Sample Code

4.1 Common Codes

4.1.1 Sample Code List

This application note provides the following sample codes created with the Smart Configurator. If the duty cycle is not indicated, waveform output is repeated in duty cycles 80%→60%→40%→20%→・・・.

Sample code can be downloaded from the Renesas Electronics website.

Table 4.1 GPTW Sample Code List (1/2)

Name	Description	Ref.
Sawtooth-Wave PWM Mode r01an5995_rx66t_gptw_sawtooth_pwm.zip	Use sawtooth-wave PWM mode Buffer transfer when TCNT overflow occurs, and 2-phase PWM waveform output	4.2
Sawtooth-Wave One-Shot Pulse	Use sawtooth-wave one-shot pulse mode Buffer transfer when TCNT overflow and compare match occur, and 1-phase PWM waveform output	4.3
r01an5995_rx66t_gptw_sawtooth_1shotpls_dt.zip	Use automatic dead time setting function	4.3
r01an5995_rx66t_gptw_sawtooth_1shotpls.zip	No use of automatic dead time setting function	4.3.5.1
Sawtooth-Wave PWM Mode 3-Phase Complementary r01an5995_rx66t_gptw_sawtooth_pwm_3phase.zip	Use sawtooth-wave PWM mode 3-phase complementary PWM waveform output without dead time Each phase outputs 25%, 50%, and 75% fixed duty cycle	4.4
Sawtooth-Wave One-Shot Pulse 3-Phase Complementary r01an5995_rx66t_gptw_sawtooth_1shotpls_3phase_dt.zip	Use sawtooth-wave one-shot pulse mode Buffer transfer when TCNT overflow and compare match occur, and 3-phase complementary PWM waveform output Output duty cycles: 20%→40%→60%→80%→60%→・・・	4.5
Triangle-Wave PWM Mode 1	Use triangle-wave PWM mode 1 (32-bit transfer at trough) Output symmetric 3-phase complementary PWM waveform with buffer Output duty cycles: 20%→40%→60%→80%→60%→・・・	4.6
r01an5995_rx66t_gptw_triangle_pwm1_dt.zip	Use automatic dead time setting function	4.6
r01an5995_rx66t_gptw_triangle_pwm1.zip	No use of automatic dead time setting function	4.6.5.3
Triangle-Wave PWM Mode 2	Use triangle-wave PWM mode 2 (32-bit transfer at crest and trough) Output laterally asymmetric 3-phase complementary PWM waveform with buffer Output duty cycles: 20%→40%→60%→80%→60%→・・・	4.7
r01an5995_rx66t_gptw_triangle_pwm2_dt.zip	Use automatic dead time setting function	4.7
r01an5995_rx66t_gptw_triangle_pwm2.zip	No use of automatic dead time setting function	4.7.5.3

Table 4.2 GPTW Sample Code List (2/2)

Name	Description	Ref.
Triangle-Wave PWM Mode 3	Use triangle-wave PWM mode 3 (64-bit transfer at trough) Output laterally asymmetric 3-phase complementary PWM waveform with double buffer Output duty cycles: 20%→40%→60%→80%→60%→・・・	4.8
r01an5995_rx66t_gptw_triangle_pwm3_dt.zip	Use automatic dead time setting function	4.8
r01an5995_rx66t_gptw_triangle_pwm3.zip	No use of automatic dead time setting function	4.8.5.3
Sawtooth-Wave PWM Mode Duty Cycles 0% to 100% r01an5995_rx66t_gptw_sawtooth_pwm_50to100.zip	Use sawtooth-wave PWM mode PWM waveform output including duty cycles 0% and 100% Output duty cycles: 50%→80%→100%→80%→50%→0%→…	4.9
Sawtooth-Wave One-Shot Pulse Duty Cycles 0% to 100% r01an5995_rx66t_gptw_sawtooth_1shotpls_50to100.zip	Use sawtooth-wave one-shot pulse mode PWM waveform output including duty cycles 0% and 100% Output duty cycles: 50%→80%→100%→80%→50%→0%→・・・	4.10
Sawtooth-Wave PWM Mode Duty Cycles 0% and 100% r01an5995_rx66t_gptw_sawtooth_pwm_0to100.zip	Use sawtooth-wave PWM mode PWM waveform output, alternating between duty cycles 0% and 100% Output duty cycles: 0%→100%→0%→100% →・・・	4.11
Triangle-Wave PWM Mode 1 Duty Cycles 0% to 100% r01an5995_rx66t_gptw_triangle_pwm_50to100_dt.zip	Use triangle-wave PWM mode 1 (32-bit transfer at trough) PWM waveform output including duty cycles 0% and 100% Output duty cycles: 50%→80%→100%→80%→50%→0%→・・・	4.12
Triangle-Wave PWM Mode Duty Cycles 0% and 100% r01an5995_rx66t_gptw_triangle_pwm_0to100_dt.zip	Use triangle-wave PWM mode 1 (32-bit transfer at trough) PWM waveform output, alternating between duty cycles 0% and 100% Output duty cycles: 0%→100%→0%→100% →・・・	4.13

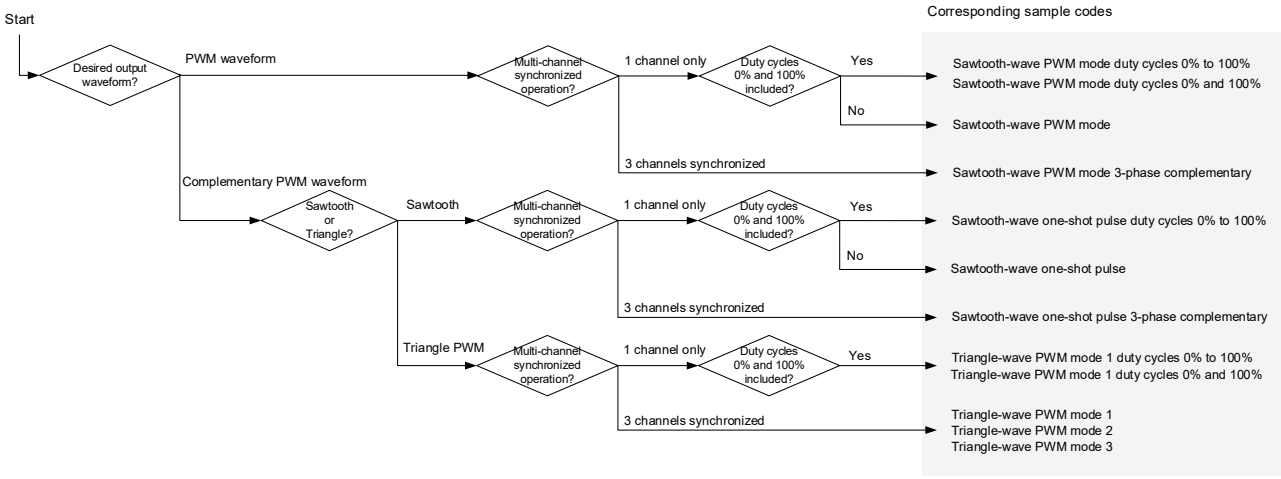


Figure 4.1 Sample Codes According to Purpose

4.1.2 Folder Structure

The main folder structure of a sample code is as follows.

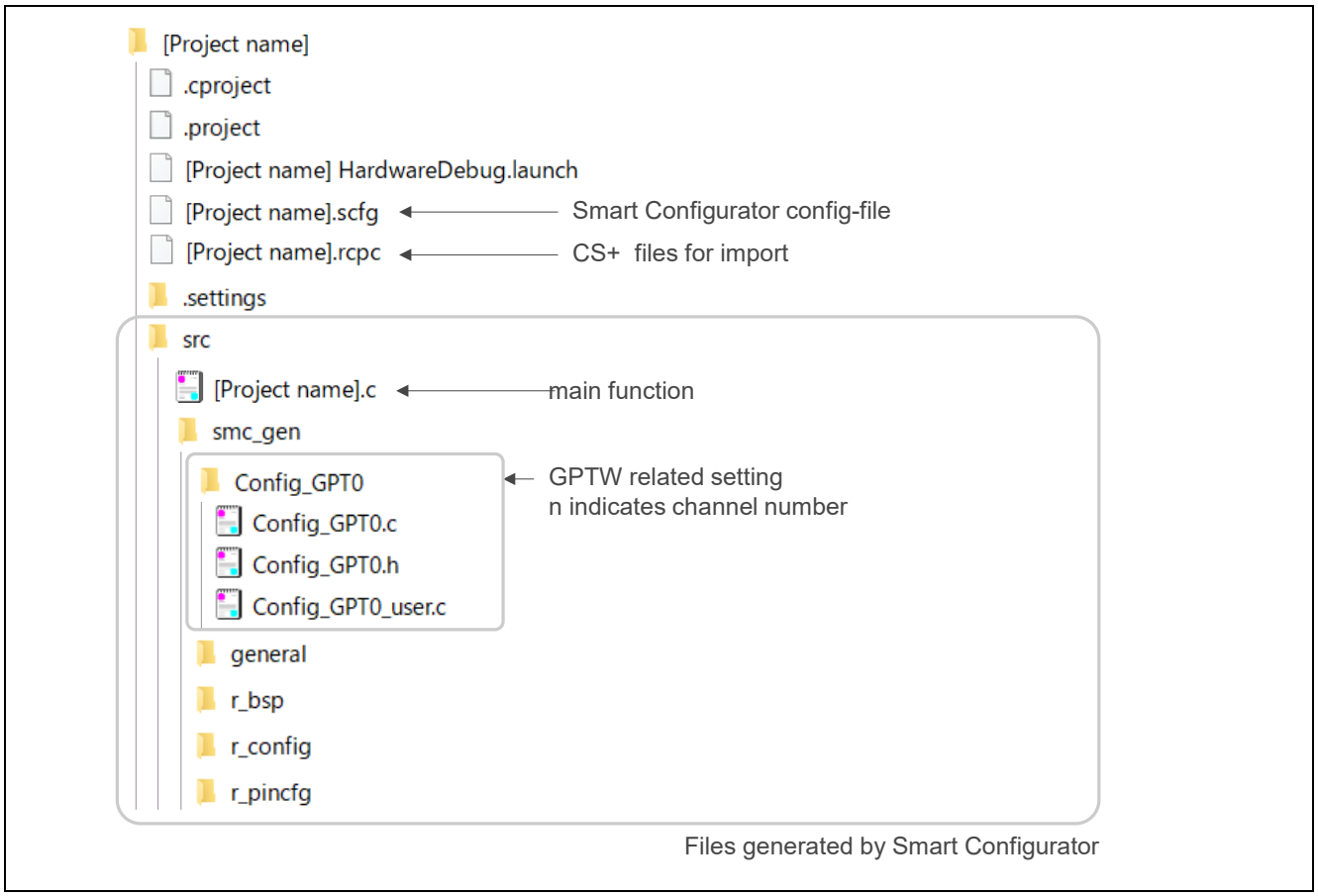


Figure 4.2 GPTW Folder Structure

4.1.3 File Structure

The main file structure of a sample code is as follows.

Table 4.3 GPTW File Structure

File Name	Description
[Project name].c	main function This is the main function. The Smart Configurator generates an empty function. The necessary processing for each sample code is described here.
Config_GPTn.c*	R_Config_GPTn_Create function This is the GPTW's initialization function. The initialization function based on the settings in the Smart Configurator is generated by the Smart Configurator. The call for this function is generated by the Smart Configurator. This function is called in the R_SystemInit function executed before the main function.
	R_Config_GPTn_Start function This is the GPTW's count start function. This function is generated by the Smart Configurator. In the sample codes, this function is called from the main function.
	R_Config_GPTn_Stop function This is the GPTW's count stop function. This function is generated by the Smart Configurator. This function is not used in the sample codes.
Config_GPTn_user.c*	r_Config_GPTn_Create_UserInit function This is the GPTW's user initialization function. The Smart Configurator generates an empty function. The necessary processing for each sample code is described here. This is the last function to be called in the R_Config_GPTn_Create function generated by the Smart Configurator.
	r_Config_GPTn_[interrupt name]_interrupt function This is the interrupt handler function. The Smart Configurator generates an empty function. The necessary processing for each sample code is described here.
Config_GPTn.h*	This is the header file that defines GPTW related functions. This file is included in the r_smc_entry.h file generated by the Smart Configurator. To use GPTW related functions, be sure to include the r_smc_entry.h file.

*: n indicates channel number

4.1.4 Adding Components

The sample codes use the Smart Configurator to add the GPTW as described below.

Table 4.4 Adding Components

Item	Description
Component	General PWM Timer ((1) in figure below)
Configuration name	Sample codes use the default setting name
Work mode	Reference the section for each sample code ((2) in figure below)
Resource	Reference the section for each sample code ((3) in figure below)

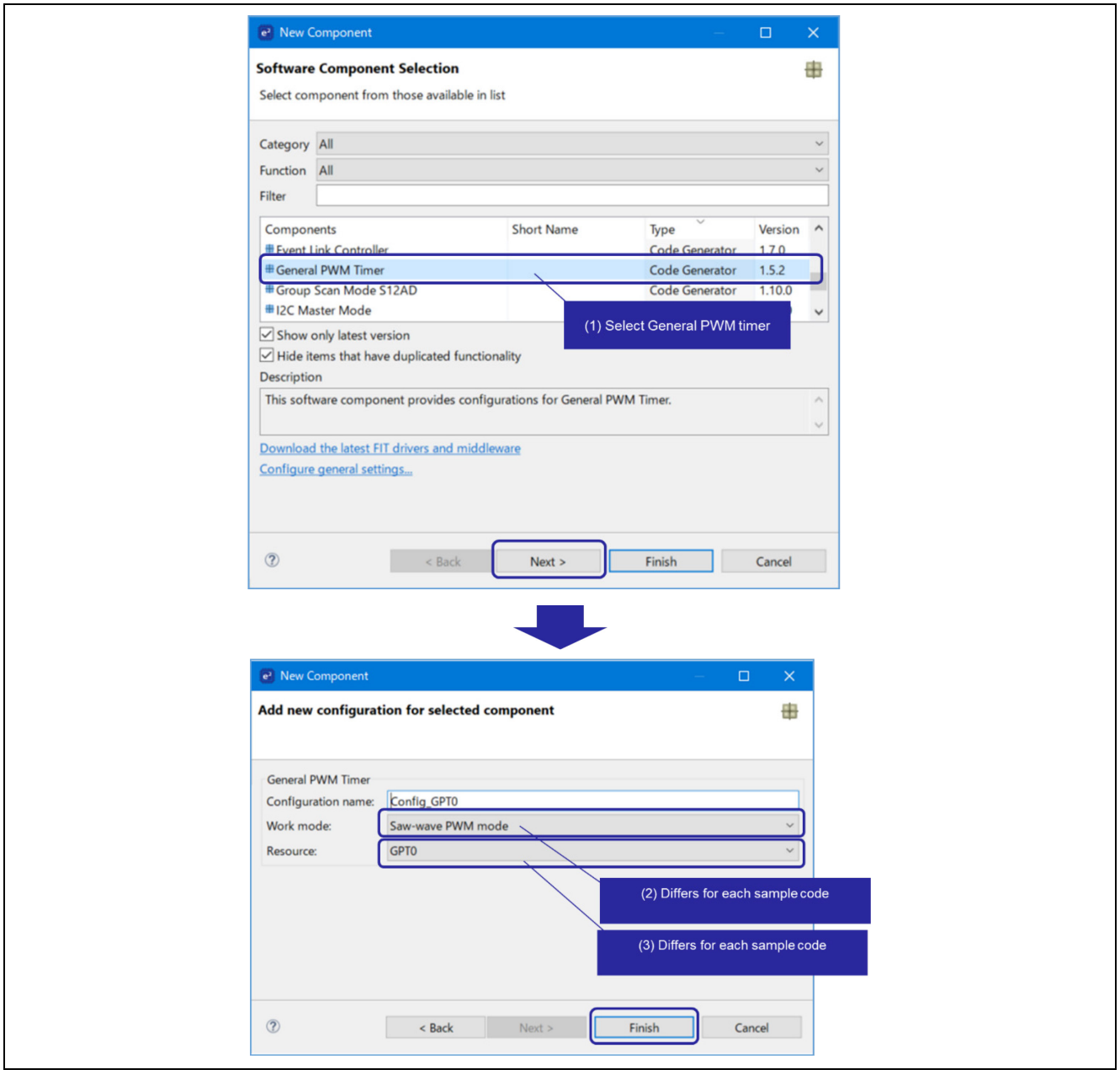


Figure 4.3 Adding Components

4.1.5 Pin Settings

Figure 4.4 Pin Settings shows an example of pin settings using the Smart Configurator.

Configure the pins after setting the GPTW. For GPTW settings, refer to “Smart Configurator Settings” for each sample code.

Pin settings are carried out in the R_Config_GPTn_Create function generated by the Smart Configurator.

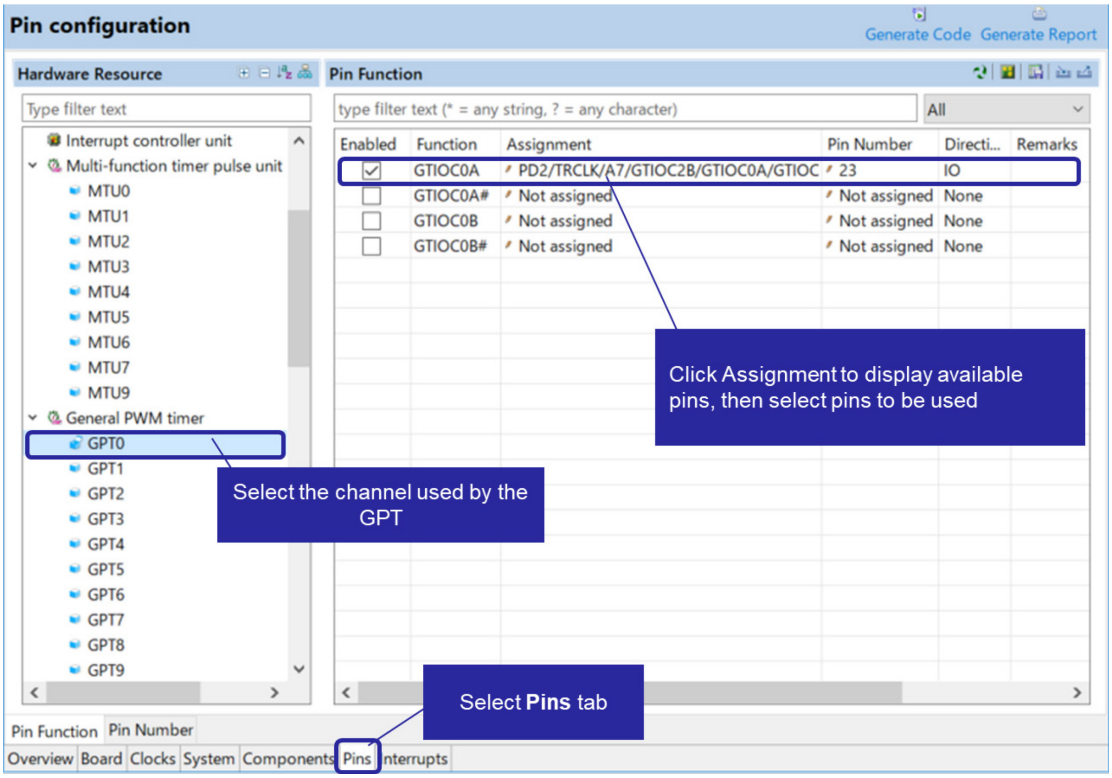


Figure 4.4 Pin Settings

4.1.6 Interrupt Settings

Figure 4.5 Interrupt Settings shows an example of interrupt settings using the Smart Configurator. For details on Software Configurable Interrupt A, refer to Renesas RX66T Group User's Manual Hardware, section 14.4.5.1 Software Configurable Interrupt A.

Configure interrupts after setting the GPTW settings. For GPTW settings, refer to "Smart Configurator Settings" for each sample code.

Interrupt settings can be configured in the R_Config_GPTn_Create function, R_Config_GPTn_Start function, and R_Config_GPTn_Stop function, all of which are generated by the Smart Configurator.

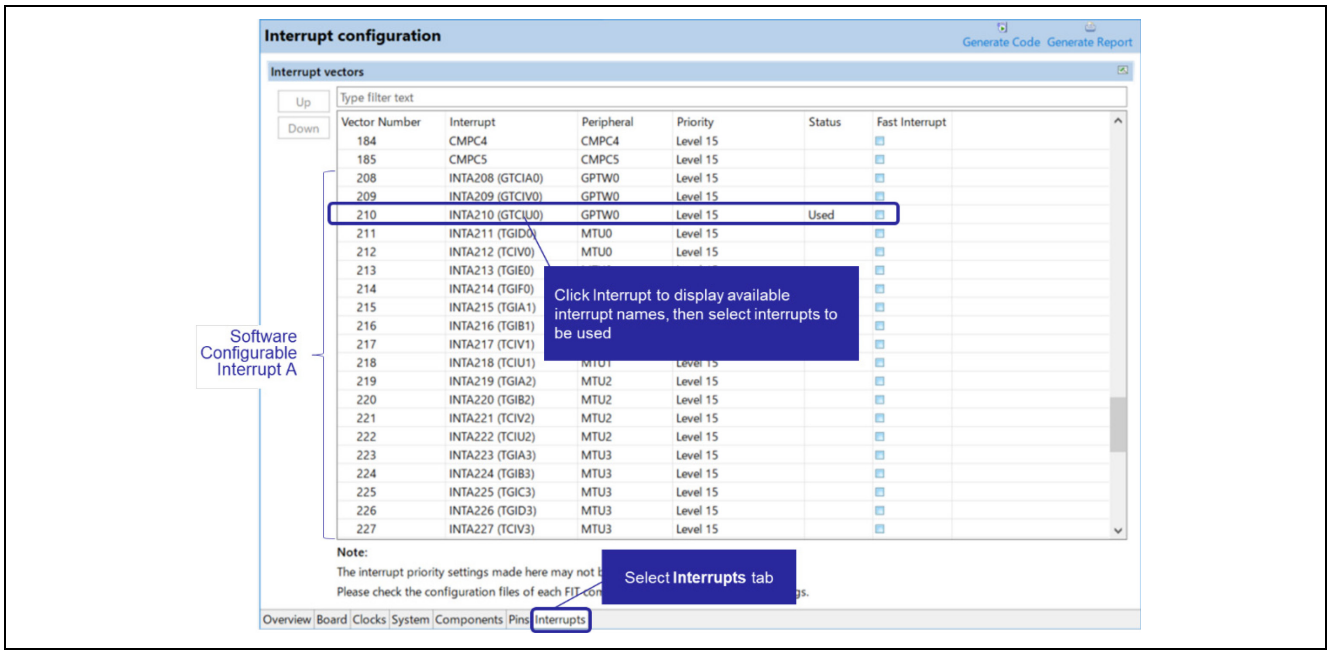


Figure 4.5 Interrupt Settings

Only GTCIE0, GTCIF0 and GDTE0 are selected for GPTW interrupts by default in the Interrupts tab of the Smart Configurator. To use interrupts configured in the Components tab, the interrupts must be selected in the Interrupts tab. The following shows the status and error message when a selection is missing.

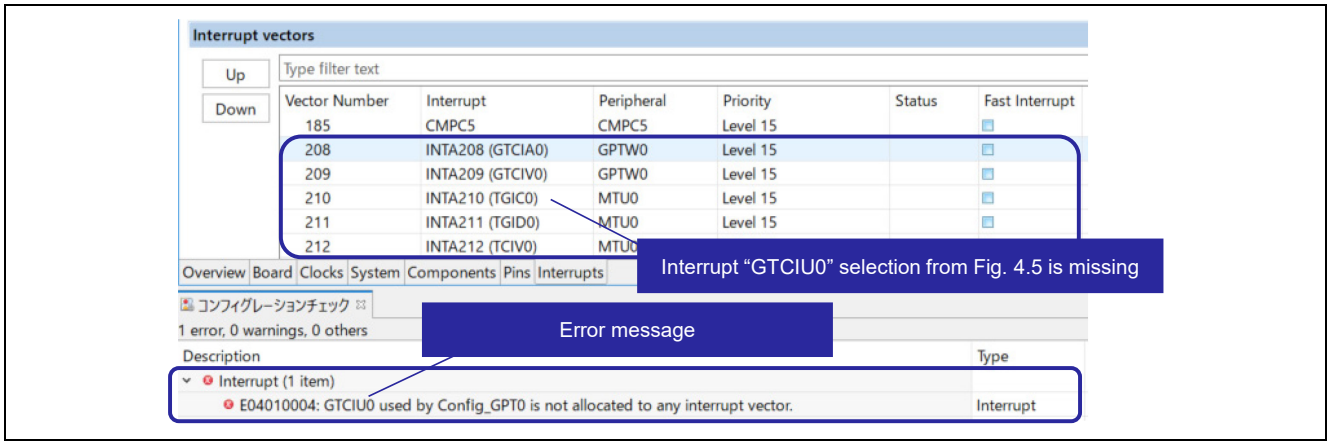


Figure 4.6 Interrupt Settings (interrupt selection missing)

4.2 Sawtooth-Wave PWM Mode

- Target sample code file name: r01an5995_rx66t_gptw_sawtooth_pwm.zip

4.2.1 Overview

The GPTW sawtooth-wave PWM mode can be used to output PWM waveforms of duty cycles 0% to 100% according to the GTCCR register setting.

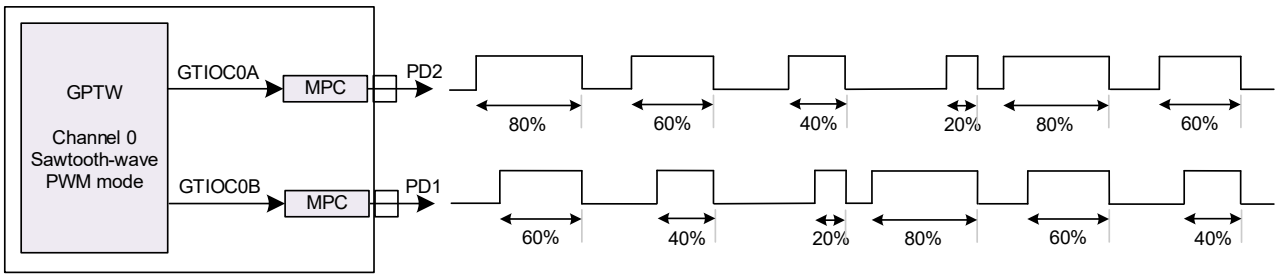
This sample code describes a sample code that uses the sawtooth-wave PWM mode and repeats waveform output of duty cycles 80% → 60% → 40% → 20% → 80% → ･･･. The duty cycle is changed using buffer registers GTCCRC and GTCCRE to transfer the value of GTCCRC and GTCCRE to compare registers GTCCRA and GTCCRB when an overflow occurs.

The following list provides the GPTW settings used in the sample code.

- Use sawtooth-wave PWM mode
- Use channel 0
- Carrier period = 1ms
- Timer counter clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Count direction = up-counting
  - Initial value of counter= 0
- Use GTCCRA as duty output compare match
  - Set GTIOC0A pin as PWM output pin
  - Low output when counting starts
  - High output at GTCCRA compare match
  - Low output at cycle end
- Use GTCCRB as duty output compare match
  - Set GTIOC0B pin as PWM output pin
  - Low output at counting starts
  - High output at GTCCRB compare match
  - Low output at cycle end
- Use buffer register
  - GTCCRC as the buffer register of GTCCRA
  - GTCCRE as the buffer register of GTCCRB
- Software source count start enabled
- Duty changes at each cycle
  - Duty changes at the GTCNT counter overflow interrupt
  - Refer to Figure 4.8 for details on duty change timing

Set in Smart Configurator.
For Setting Methods,
refer to section 4.2.3.

Sawtooth-wave PWM mode output for this sample code is shown below.

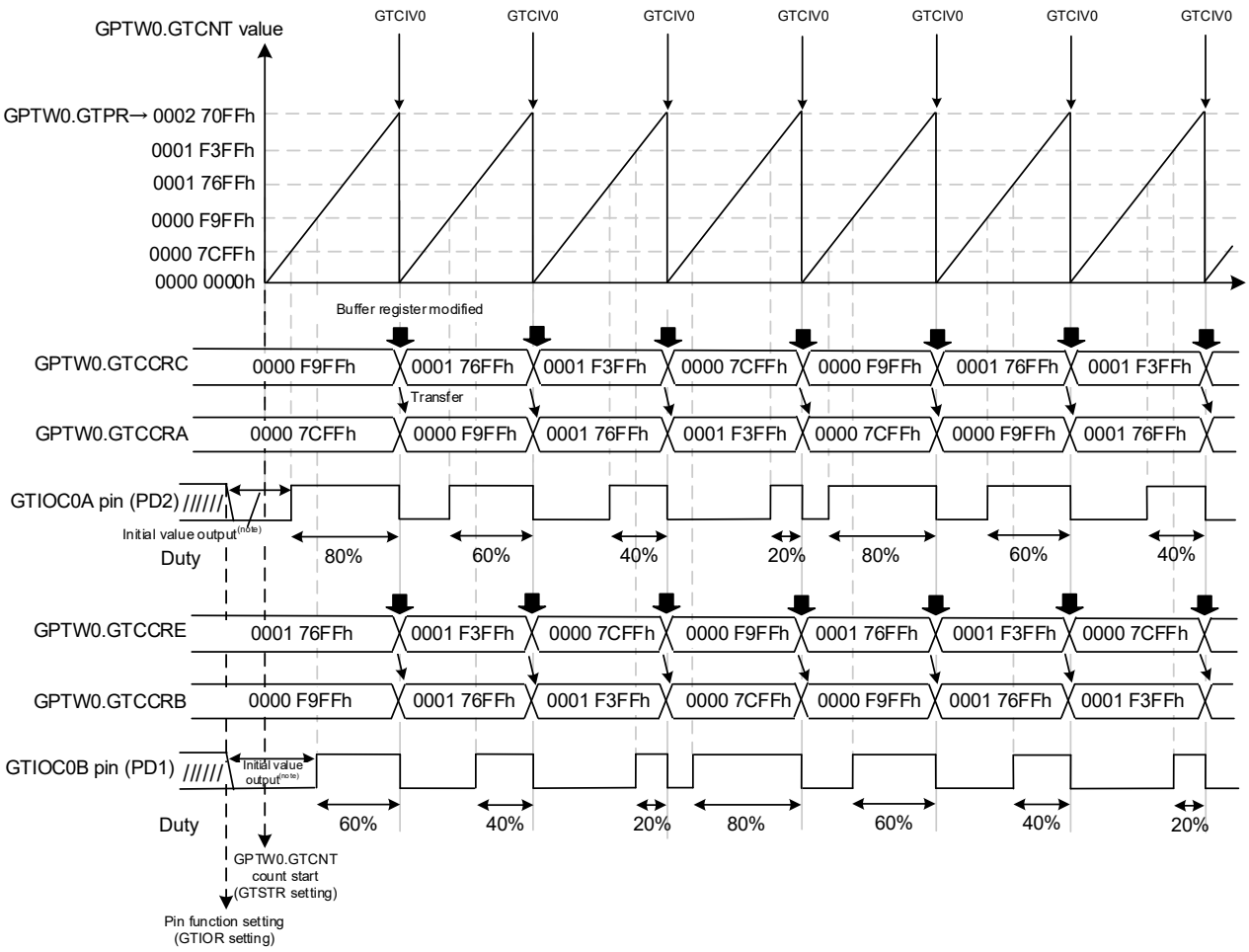


GPTW: Generates sawtooth-wave PWM mode waveform
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

Figure 4.7 Sawtooth-wave PWM Mode Output

4.2.2 Operation Details

The sample code operations are shown in Figure 4.8. The settings of the duty cycle are changed with each period by modifying the values of buffer registers GTCCRC and GTCCRE at a GTCNT overflow interrupt (GTCIV0). The values of GTCCRC and GTCCRE are transferred to compare registers GTCCRA and GTCCRB when a CTCNT overflow occurs.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.8 Sample Code Operations

4.2.3 Smart Configurator Setting

The sample codes use the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.5 Adding Components

Item	Description
Component	General PWM Timer
Configuration name	Config_GPT0
Work mode	Sawtooth-wave PWM mode
Resource	GPT0

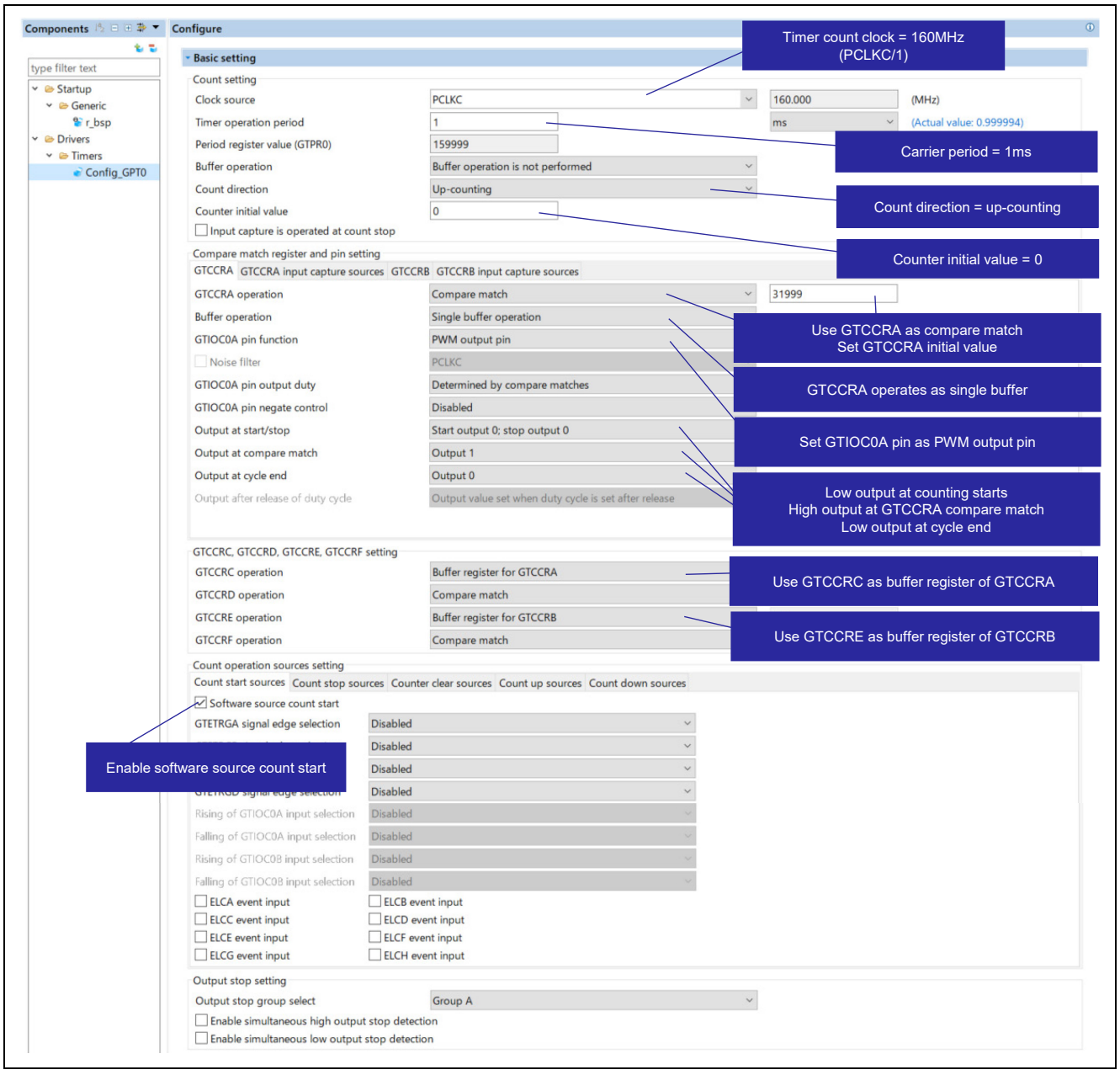


Figure 4.9 GPT0 Settings (1/2)

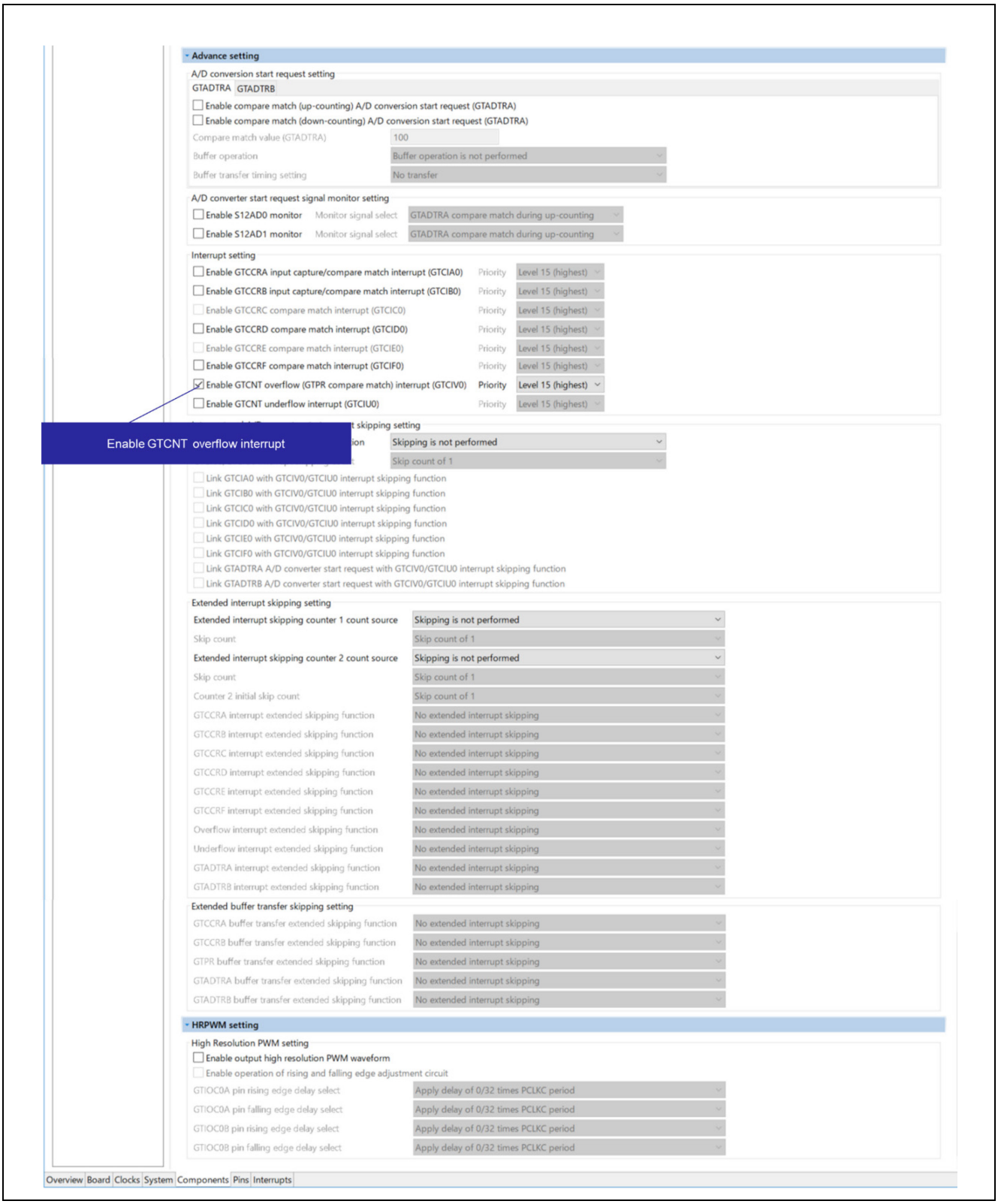


Figure 4.10 GPT0 Settings (2/2)

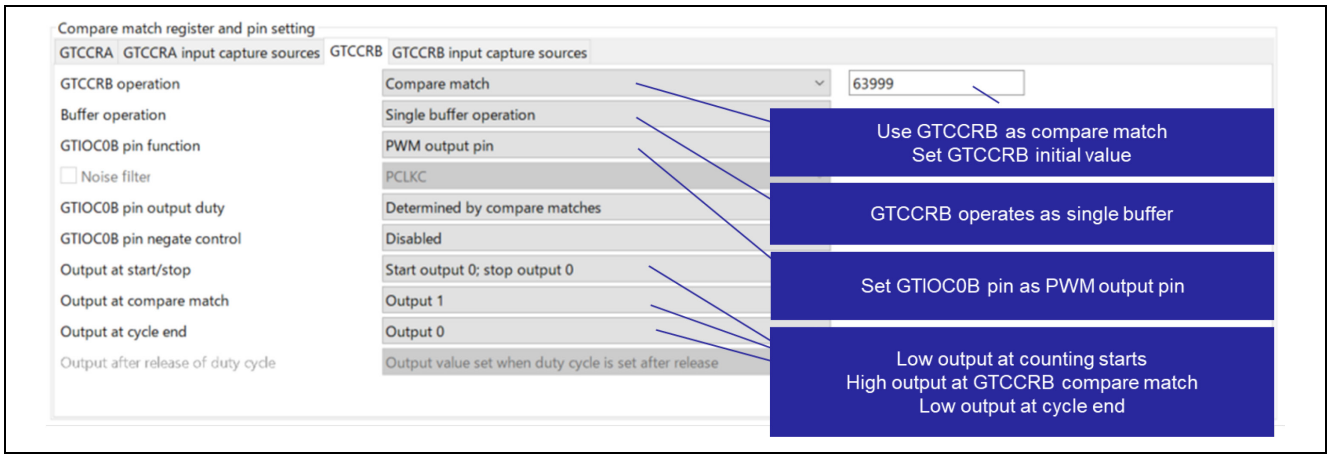


Figure 4.11 GPT0 Settings (Compare Match Register and Pin Setting of GTCCRB)

4.2.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

Counting is started in the main function.

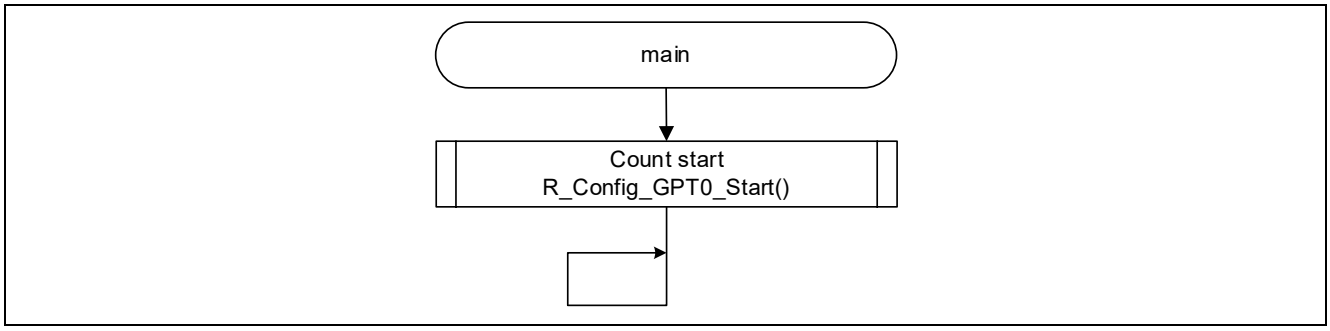


Figure 4.12 main Function

The user initialization function R_Config_GPT0_Create_UserInit, which is executed before the main function, sets the initial values of the buffer registers. This function is called from within the R_Config_GPT0_Create function.

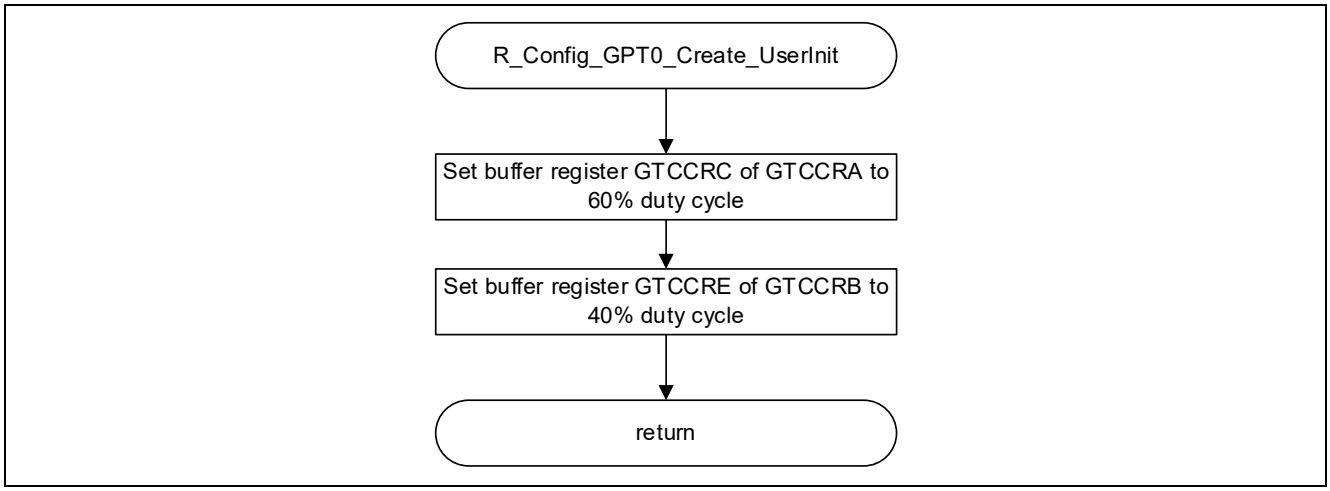


Figure 4.13 User Initialization Function

The GTCIV0 interrupt handler function changes the values of buffer registers GTCCRC and GTCCRE according to the current duty cycle.

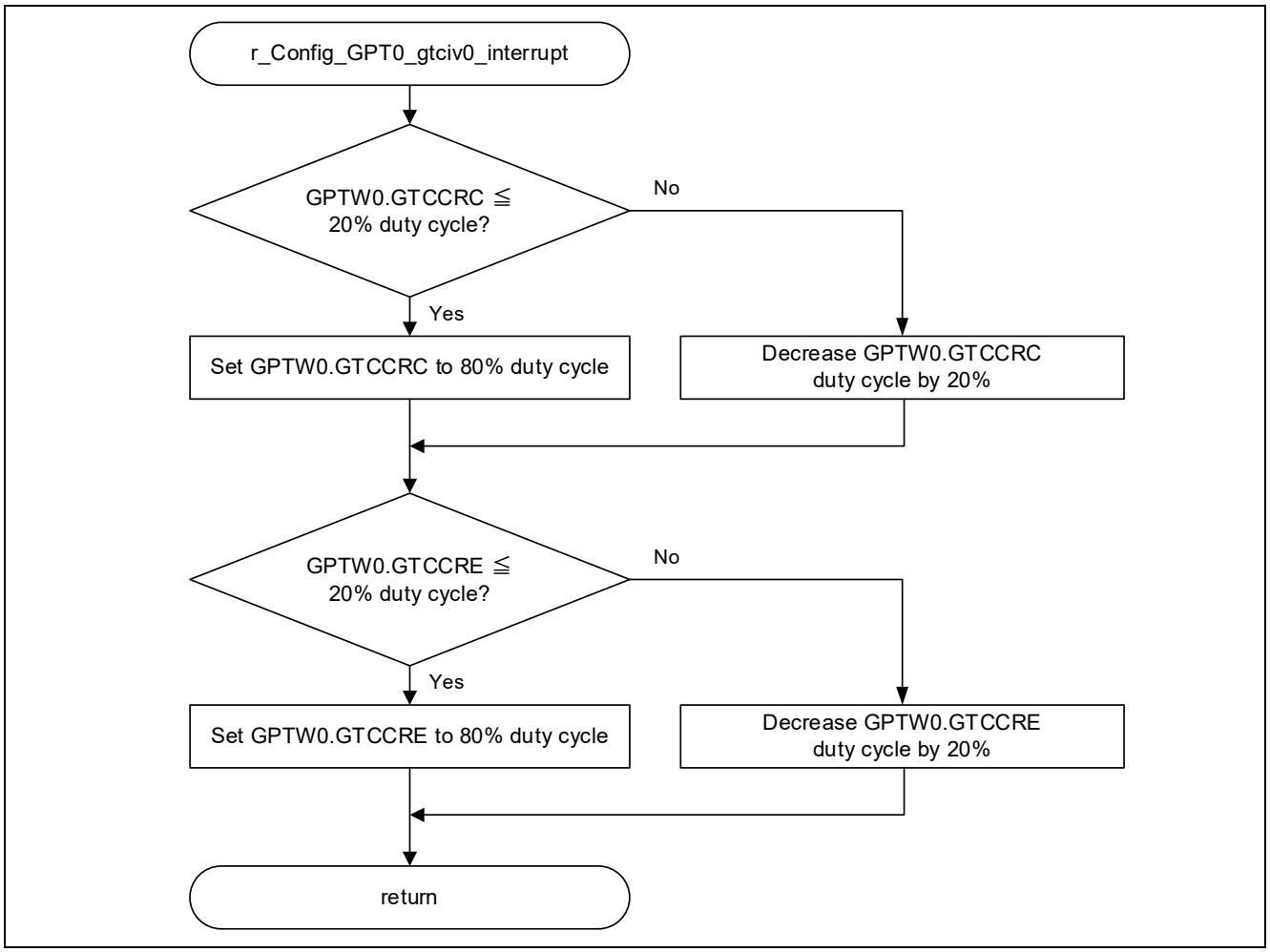


Figure 4.14 GTCIV0 Interrupt Handler Function

4.2.5 Usage Notes

4.2.5.1 Settings of GTCCRm Register during Compare Match Operation (m = A to F)

Set compare registers GTCCRA and GTCCRB to a value higher than 0000 0001h but less than the setting value of the GTPR register. If set to 0000 0000h or the same value as the GTPR register, a compare match occurs within the cycle only when the compare match register value is 0000 0000h or the compare register is set to the same value as the GTPR register. If the compare register is set to a value that exceeds the setting value of to the GTPR register, no compare match occurs.

For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F), (5) In Sawtooth-wave PWM Mode.

4.3 Sawtooth-Wave One-Shot Pulse

- Target sample code file name: r01an5995_rx66t_gptw_sawtooth_1shotpls_dt.zip

4.3.1 Overview

In the GPTW sawtooth-wave one-shot pulse mode, the GTCNT counter operates in sawtooth-wave (half-wave) by setting the period in the GTPR register, and PWM waveforms can be output from the GTIOCnA and GTIOCnB pins (n = 0 to 9) by the compare match between the GTCCRA and GTCCRB registers.

This sample code describes a sample code that uses the automatic dead time setting function in the sawtooth-wave one-shot pulse mode and repeats the following waveform output.

- GTIOC0A pin high-width switching: 80% → 60% → 40% → 20% → 80% → 60% → ⋯
- GTIOC0B pin low-width switching: 90% → 70% → 50% → 30% → 90% → 70% → ⋯

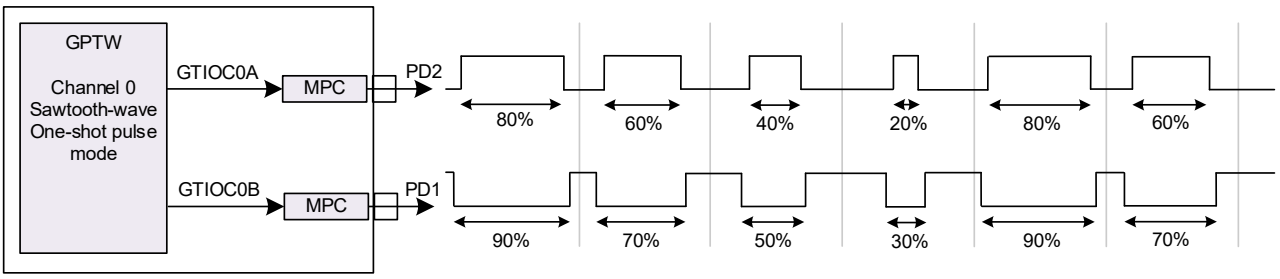
A value of temporary register A is transferred to compare register GTCCRA when a GTCCRA compare match occurs. The duty cycle is changed by transferring the value from buffer register GTCCRD to temporary register A and from buffer register GTCCRC to compare register GTCCRA when a GTCNT counter overflow occurs.

The following list provides the GPTW settings used in the sample code.

- Use sawtooth-wave one-shot pulse mode
- Use channel 0
- Carrier period = 1ms
- Timer count clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Count direction = up-counting
  - Counter initial value = 0
- Use GTCCRA as duty output compare match
  - Set GTIOC0A pin as PWM output pin
  - Low output at counting starts
  - Toggle output at GTCCRA compare match
  - Retain output at cycle end
- Use GTCCRB as duty output compare match
  - Set GTIOC0B pin as PWM output pin
  - High output at counting starts
  - Toggle output at GTCCRB compare match
  - Retain output at cycle end
- Use double buffer register
  - Use GTCCRC and GTCCRD as buffer registers of GTCCRA
- Use automatic dead time generation
- Software source count start enabled
- Duty changes at each cycle
  - Duty changes at the GTCNT counter overflow interrupt
  - Refer to Figure 4.16 for details on duty change timing

Set in Smart Configurator.
For Setting Methods,
refer to section 4.3.3.

Sawtooth-wave one-shot pulse mode output for this sample code is shown below.



GPTW: Generates sawtooth-wave one-shot pulse mode waveform
MPC: Sets pins to be used from general purpose I/O port to peripheral function I/O port

Figure 4.15 Sawtooth-wave One-Shot Pulse Mode Output

4.3.2 Operation Details

The sample code operations are shown in Figure 4.16. The settings of the duty cycle are changed with each period by modifying the values of buffer registers GTCCRC and GTCCRD at a GTCNT overflow interrupt (GTCIV0) ((4) in Figure 4.16).

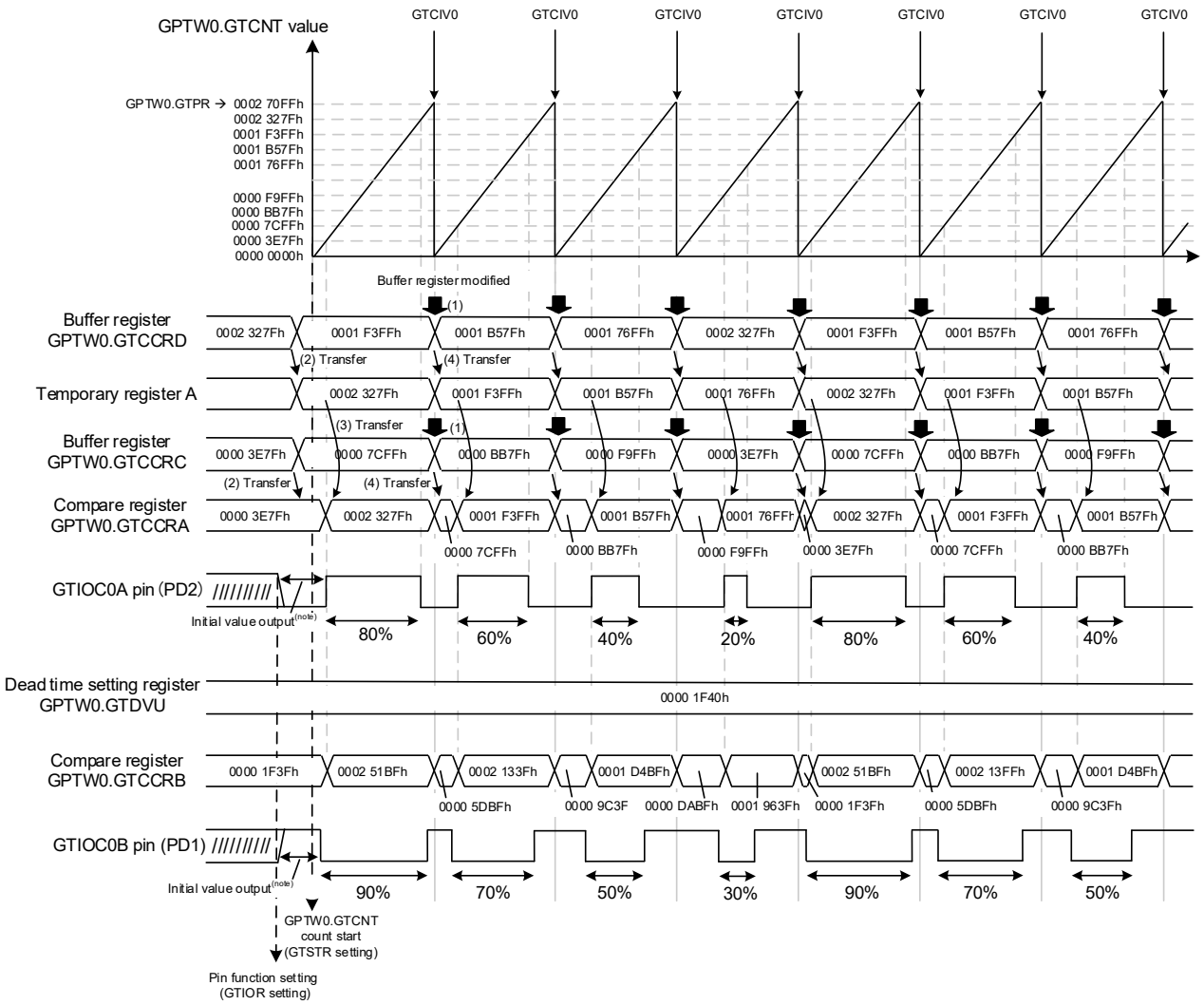
In this sample code, the initial values of the buffer registers for the code generated using the Smart Configurator are set to the same values as the compare register. As a result, the buffer register values are set in the user initialization function R_Config_GPT0_Create_UserInit before the counting is started.

The values set in the registers are transferred from buffer register GTCCRD to temporary register A and from buffer register GTCCRC to compare register GTCCRA, respectively, by forced buffer transfers ((2) in Figure 4.16).

After the counting starts, steps 1 and 2 below are repeated to update the data in the compare register.

1. Transfer from temporary register A to compare register GTCCRA when a GTCCRA compare match occurs ((3) in Figure 4.16).
2. Transfer from buffer register GTCCRD to temporary register A and from buffer register GRCCRC to compare register GTCCRA, respectively, when a GTCNT counter overflow occurs ((4) in Figure 4.16).

In addition, the GTCCRB register is automatically set according to the GTCCRA update because the automatic dead time function is used. The same values are set for the GTDVU and GTDVD.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.16 Sample Code Operations

4.3.3 Smart Configurator Settings

The sample codes use the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.6 Adding Components

Item	Description
Component	General PWM Timer
Configuration name	Config_GPT0
Work mode	Sawtooth-wave One-shot Pulse Mode
Resource	GPT0

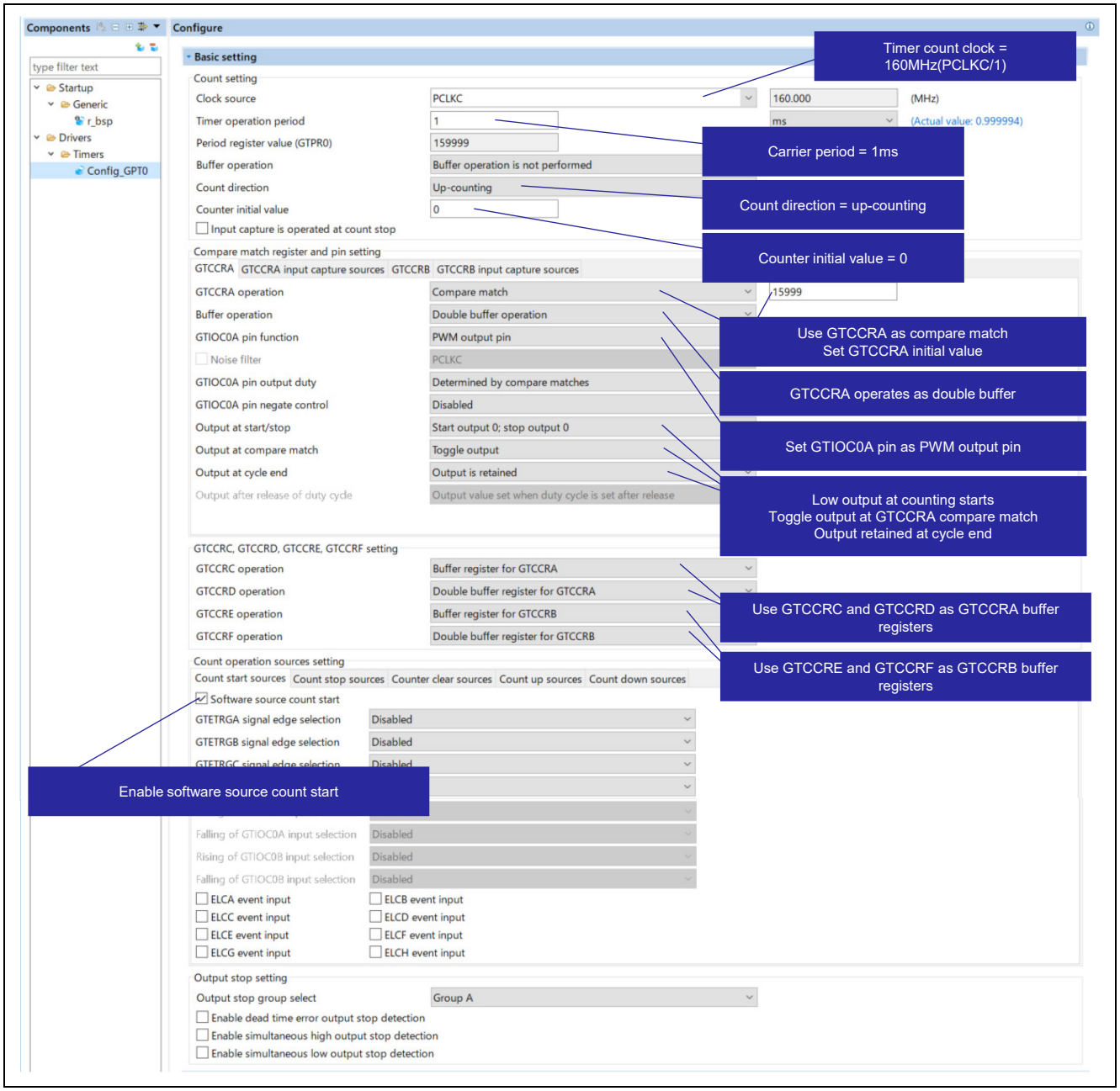


Figure 4.17 GPT0 Settings (1/2)

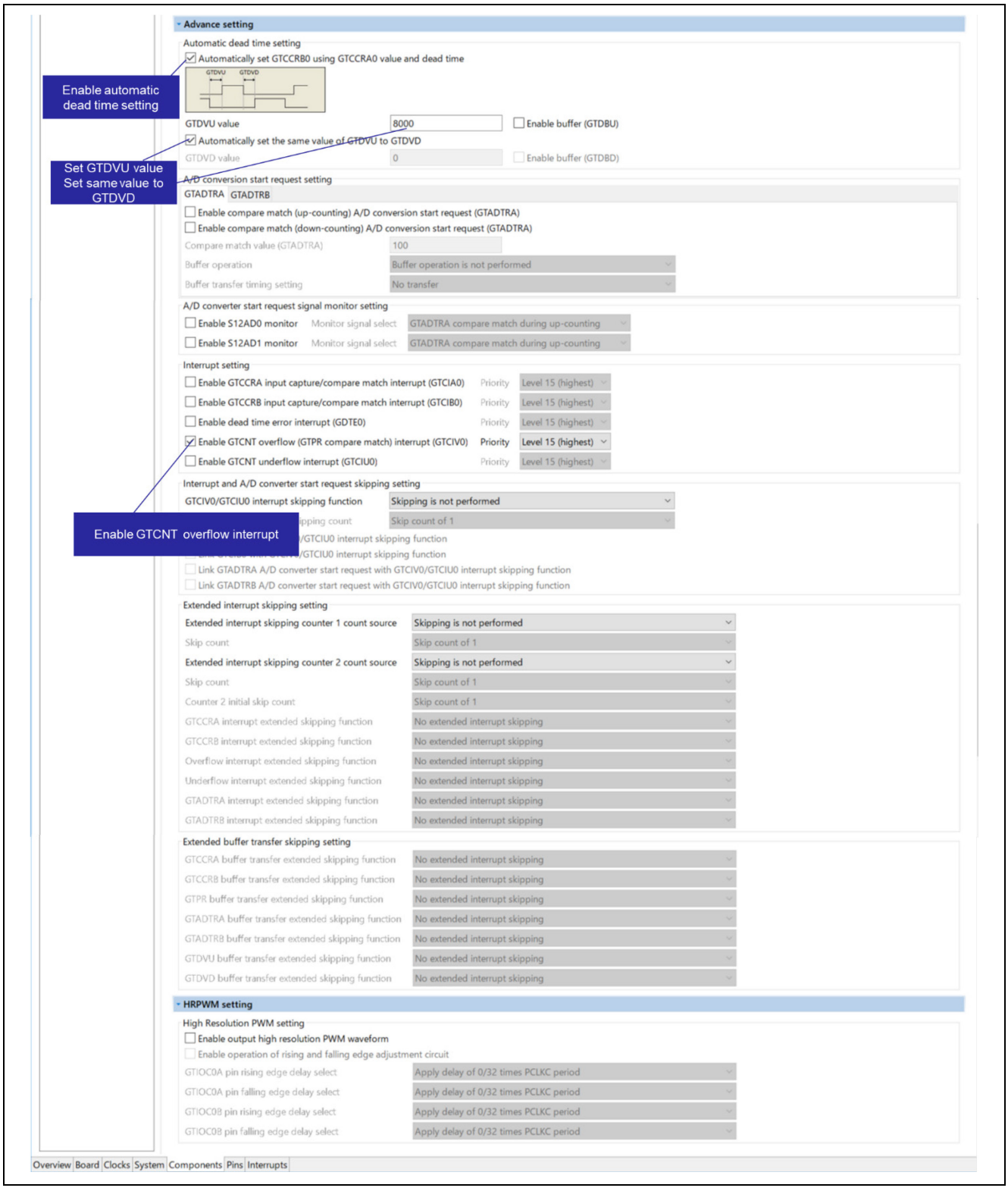


Figure 4.18 GPT0 Settings (2/2)

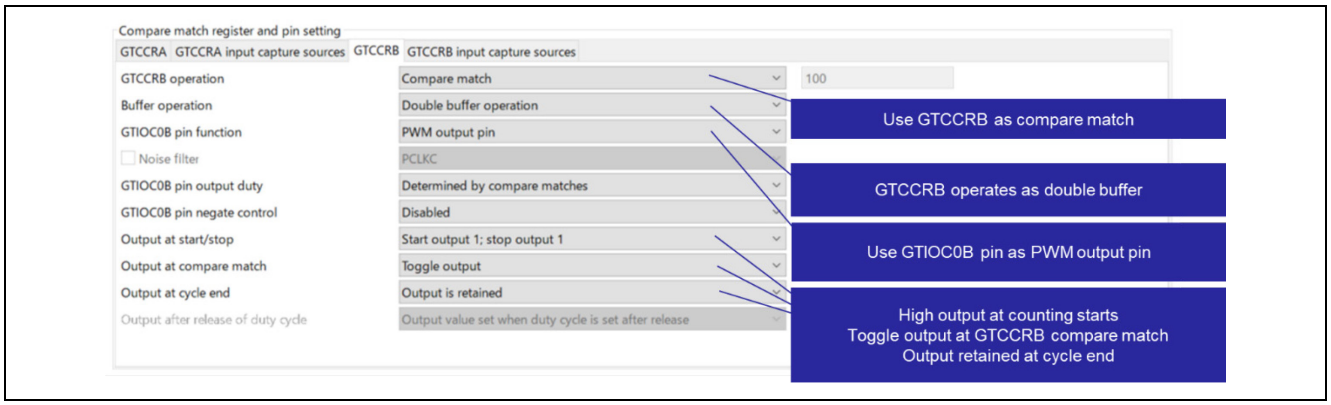


Figure 4.19 GPT0 Settings (Compare Match Register and Pin Setting of GTCCRB)

4.3.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

Counting is started in the main function.

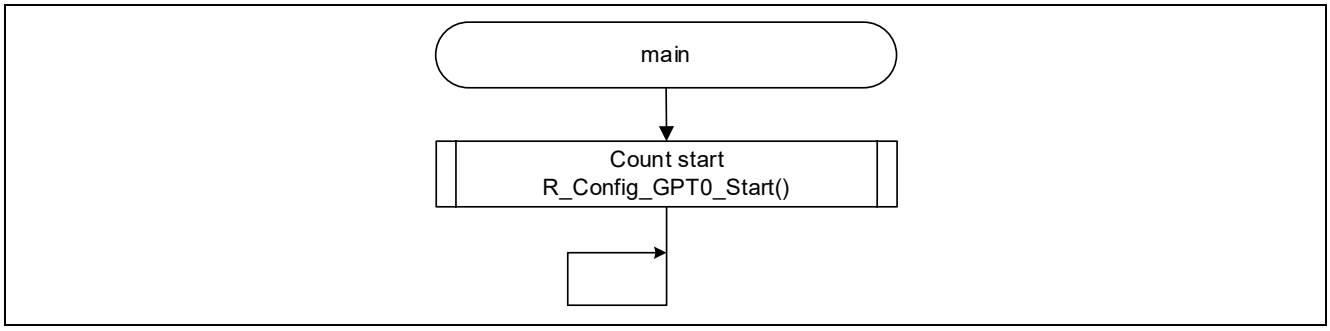


Figure 4.20 main Function

The user initialization function R_Config_GPT0_Create_UserInit, which is executed before the main function, sets the initial values of the buffer registers. In order to set the second compare match register value in the 1st cycle, a forced buffer transfer is performed after setting the buffer register value, and then the temporary register and compare register values are set. This function is called from within the R_Config_GPT0_Create function.

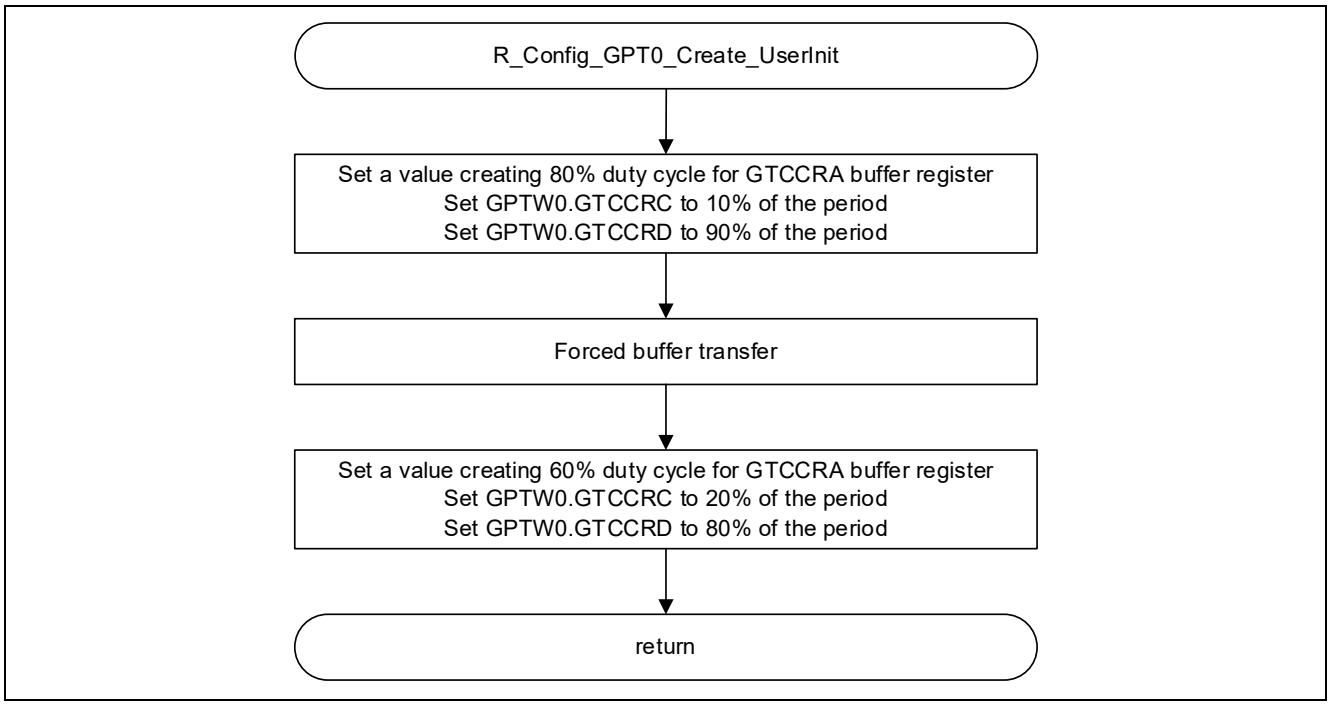


Figure 4.21 User Initialization Function

The GTCIV0 interrupt handler function changes the values of buffer registers GTCCRC and GTCCRD according to the current register values.

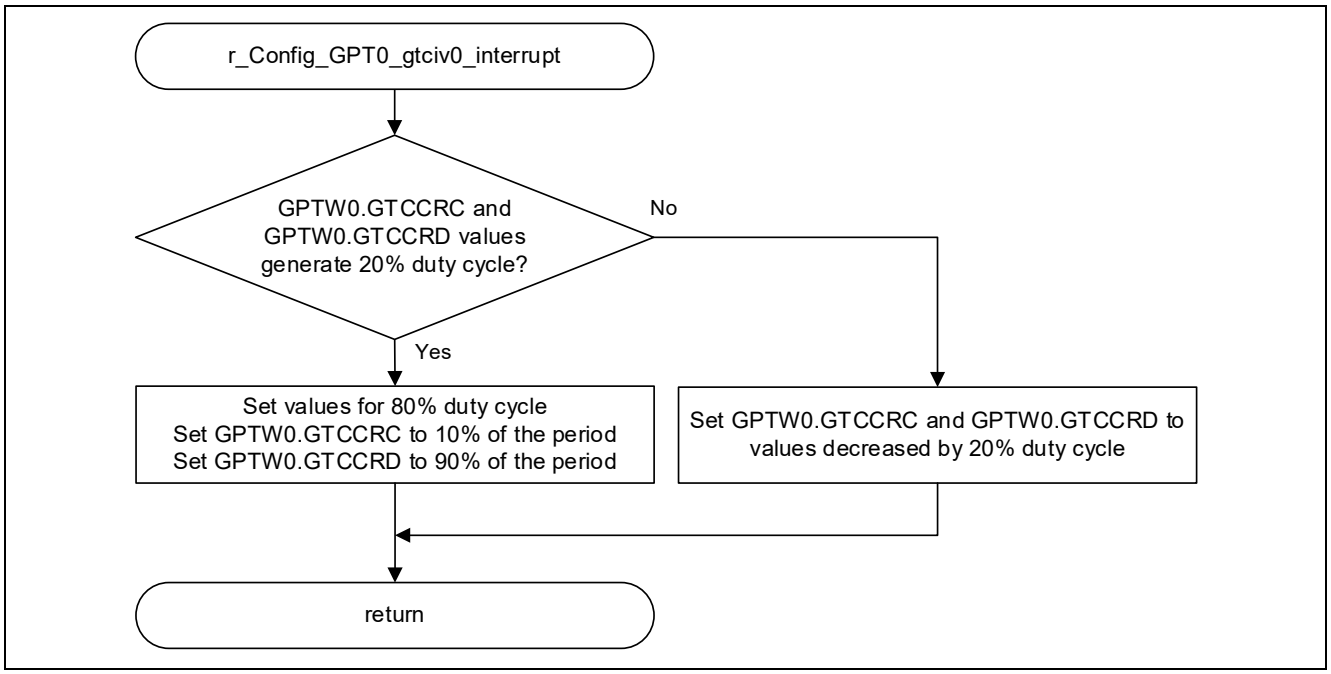


Figure 4.22 GTCIV0 Interrupt Handler Function

4.3.5 Related Operations

4.3.5.1 When Automatic Dead Time Setting Function is Not Used

- Target sample code file name: r01an5995_rx66t_gptw_sawtooth_1shotpls.zip

Figure 4.23 shows operations when the automatic dead time setting function is not used (GTDTCR.TDE bit is 0).

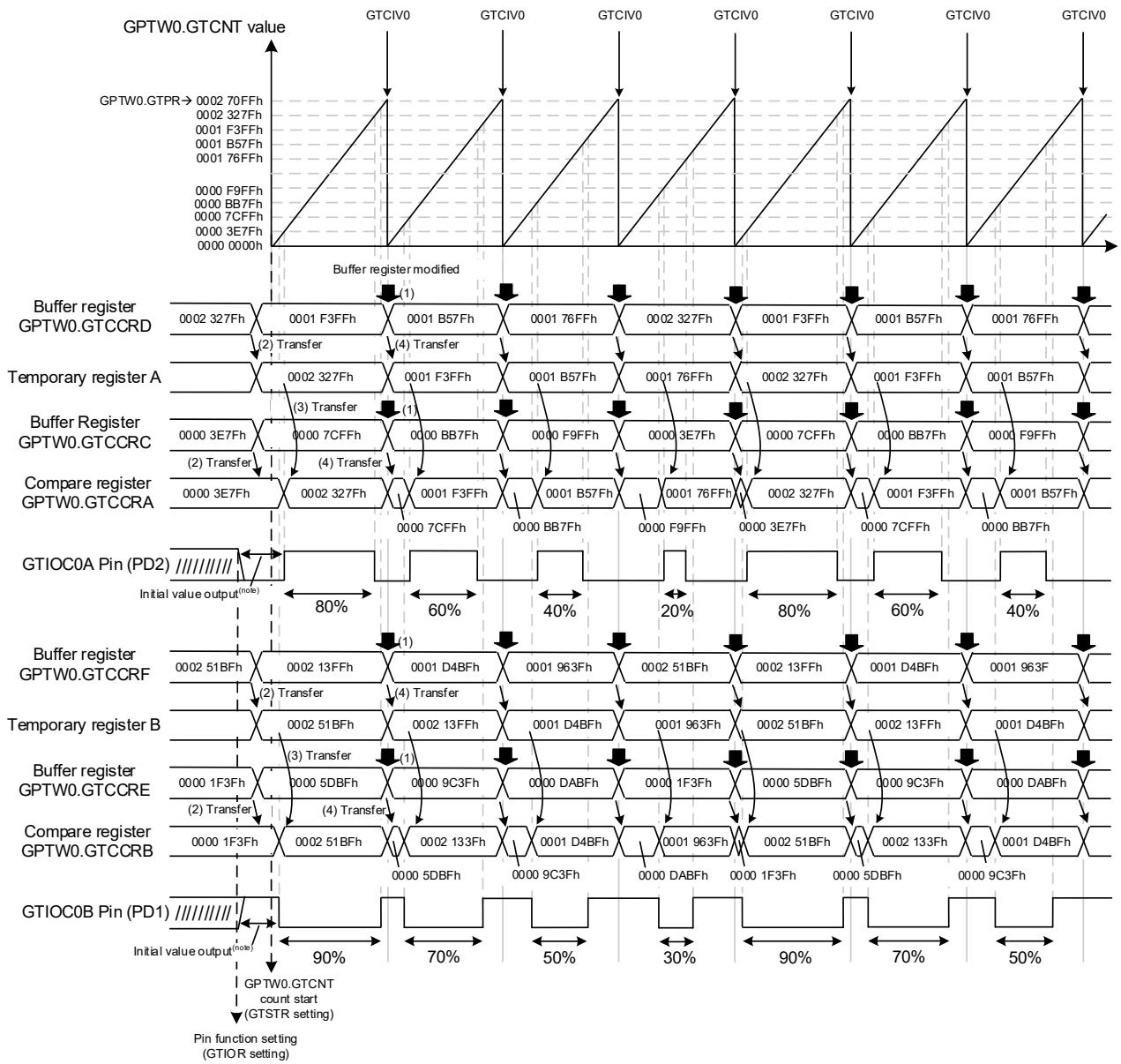
In the above sample code, the value of buffer registers GTCCRC, GTCCRD, GTCCRE, and GTCCRF are modified by the GTCNT counter overflow interrupt (GTCIV0) because the cycle duty is changed with each period ((1) in Figure 4.23).

In this sample code, the initial values of the buffer registers for the code generated using the Smart Configurator are set to the same value as the compare register. As a result, the buffer register values are set in the user initialization function R_Config_GPT0_Create_UserInit before the counting starts. The values set in the registers are transferred from buffer register GTCCRD (GTCCRF) to temporary register A (B) and from buffer register GTCCRC (GTCCRE) to compare register GTCCRA (GTCCRB), respectively, by forced buffer transfers ((2) in Figure 4.23).

After the counting starts, steps 1 and 2 below are repeated to update the data in the compare register.

1. Transfer from temporary register A (B) to compare register GTCCRA (GTCCRB) when a GTCCRA (GTCCRB) compare match occurs ((3) in Figure 4.23).
2. Transfer from buffer register GTCCRD (GTTCRF) to temporary register A (B) and from buffer register GRCCRC (GTCCRE) to compare register GTCCRA (GTCCRE), respectively, when a GTCNT counter overflow occurs ((4) in Figure 4.23).

In addition, the same dead time period is secured as shown in the operations in Figure 4.16.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.23 When Automatic Dead Time Setting Function is Not Used (Sawtooth-Wave One-Shot Pulse Mode)

4.3.6 Usage Notes

4.3.6.1 Settings of GTCCRm Register during Compare Match Operation (m = A to F

This sample code automatically sets the dead time and uses the value of compare match register GTCCRA, the register for positive-phase waveform, to update the value of compare match register GTCCRB, the register for negative-phase waveform.

The values of buffer registers GTCCRC and GTCCRD should be set to satisfy the following restrictions. If the restrictions are not satisfied, correct output waveforms with secured dead time may not be obtained.

- In up-counting:
  GTCCRC < GTCCRD
  GTCCRC > GTDVU
  GTCCRD < GTPR – GTDVD
- In down-counting:
  GTCCRC > GTCCRD
  GTCCRC < GTPR – GTDVU
  GTCCRD > GTDVD

Further, if the dead time is not automatically set, buffer registers GTCCRC and GTCCRD (or GTCCRE and GTCCRF) should be set to satisfy the following restrictions. If the restrictions are not satisfied, the compare match does not occur twice and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC (GTCCRE) < GTCCRD (GTCCRF) < GTPR
- In down-counting: GTPR > GTCCRC (GTCCRE) > GTCCRD (GTCCRF) > 0

For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F), (3) When Automatic Dead Time Setting has been Made in Sawtooth-Wave One-Shot Pulse Mode and (4) When Automatic Dead Time Setting has not been Made in Sawtooth-Wave One-Shot Pulse Mode.

4.4 Sawtooth-Wave PWM Mode 3-Phase Complementary

- Target sample code file name: r01an5995_rx66t_gptw_sawtooth_pwm_3phase.zip

4.4.1 Overview

Channels 0, 1, and 2 can be used in the GPTW sawtooth-wave PWM to output PWM waveforms of duty cycles 0% to 100% according to the GTCCR register setting.

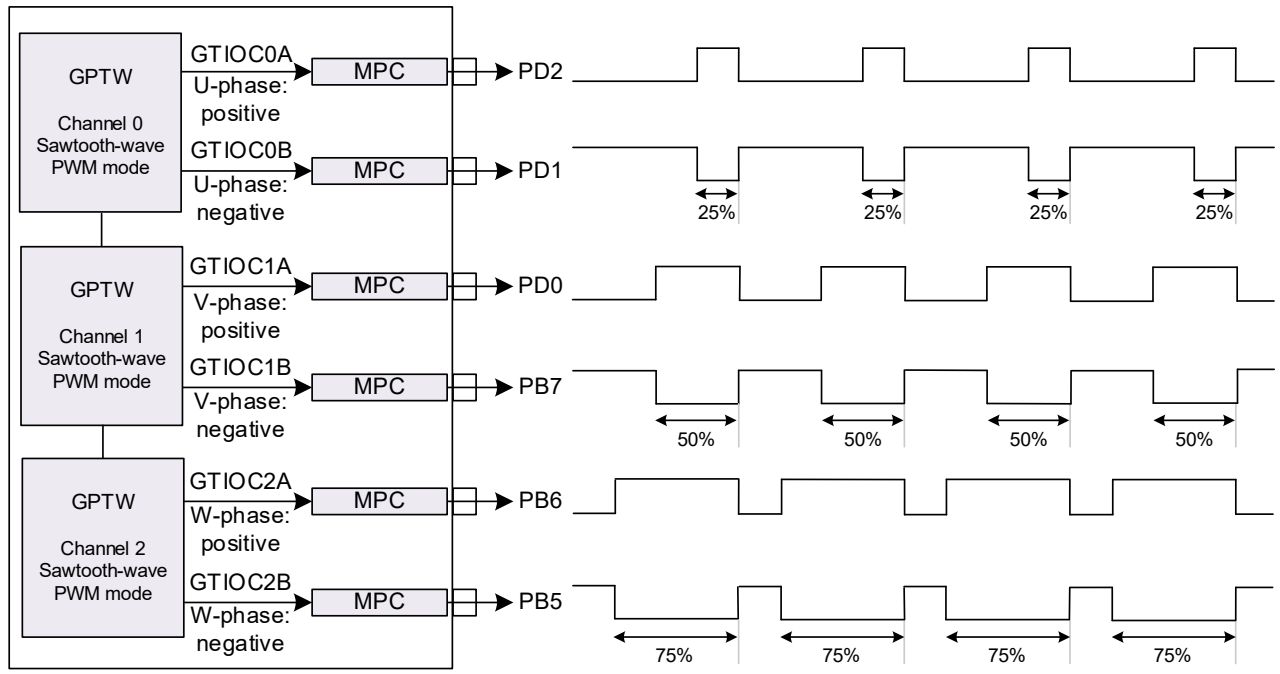
This sample code describes a sample code that uses the sawtooth-wave PWM mode to repeatedly output PWM waveforms (positive and negative phases) for 3 phases each, for a total of 6 phases, with a constant duty cycle.

The following list provides the GPTW settings used in the sample code.

- Use sawtooth-wave PWM mode
- Use channels 0, 1 and 2 (channel numbers: n = 0, 1, and 2)
- Carrier period = 1ms
- Timer count clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Count direction = up-counting
  - Counter initial value = 0
- Use GTCCRA as duty output compare match
  - Set GTIOCnA pin as PWM output pin
  - Low output at counting starts
  - High output at GTCCRA compare match
  - Low output at cycle end
- Use GTCCRB as duty output compare match
  - Set GTIOCnB pin as PWM output pin
  - High output at counting starts
  - Low output at GTCCRB compare match
  - High output at end of cycle
- Buffer operation not used
- Software source count start enabled

Set in Smart Configurator.
For Setting Methods,
refer to section 4.4.3.

Sawtooth-wave PWM mode output for this sample code is shown below.

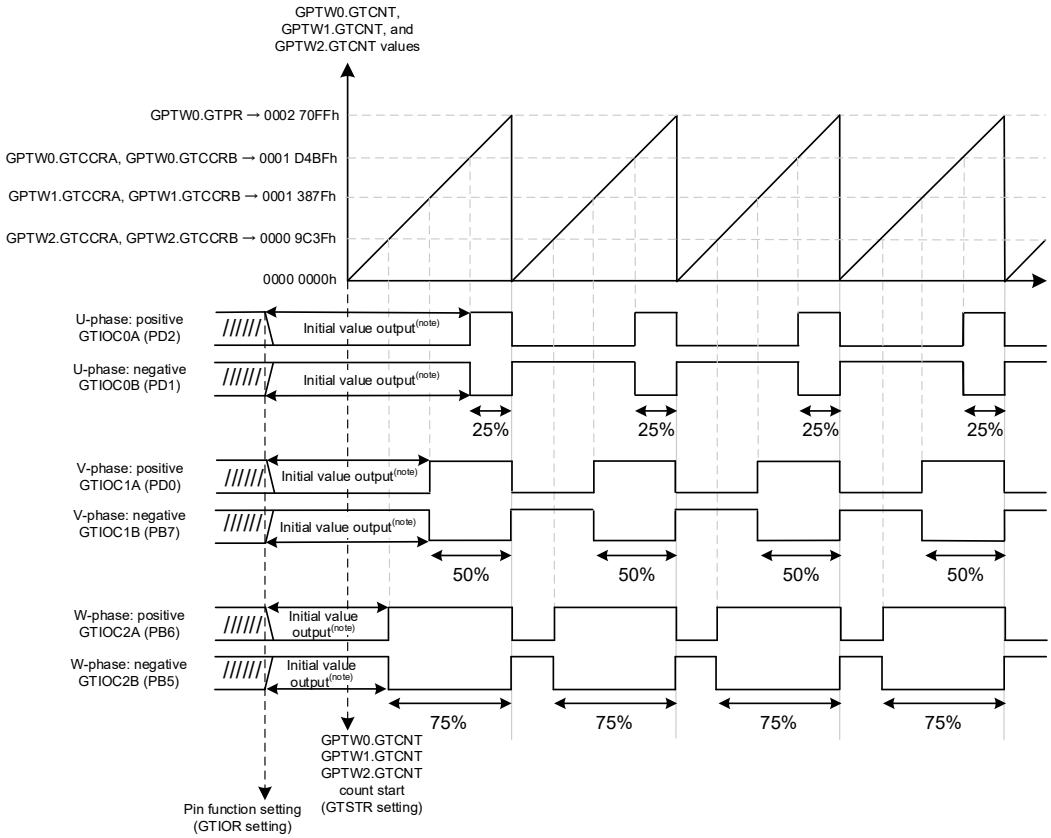


GPTW: Generates sawtooth-wave PWM mode waveform
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

Figure 4.24 Sawtooth-wave PWM Mode Output

4.4.2 Operation Details

The sample code operations are shown in Figure 4.25. The PWM output pin toggles output every time a compare match occurs for GTCCRA and GTCCRB registers of each channel and when a GTCNT counter overflow (counter clear) occurs. The same waveform output as in MTU reset-synchronized PWM mode is obtained. Refer to section 3.8 for details.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.25 Sample Code Operations

4.4.3 Smart Configurator Settings

The sample codes use the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.7 Adding Components

Item	Description		
Component	General PWM Timer		
Configuration name	Config_GPT0	Config_GPT1	Config_GPT2
Work mode	Sawtooth-wave PWM mode		
Resource	GPT0	GPT1	GPT2

Figure 4.26 and Figure 4.27 show the Config_GPT0 settings. The same settings apply for GPT1 and GPT2.

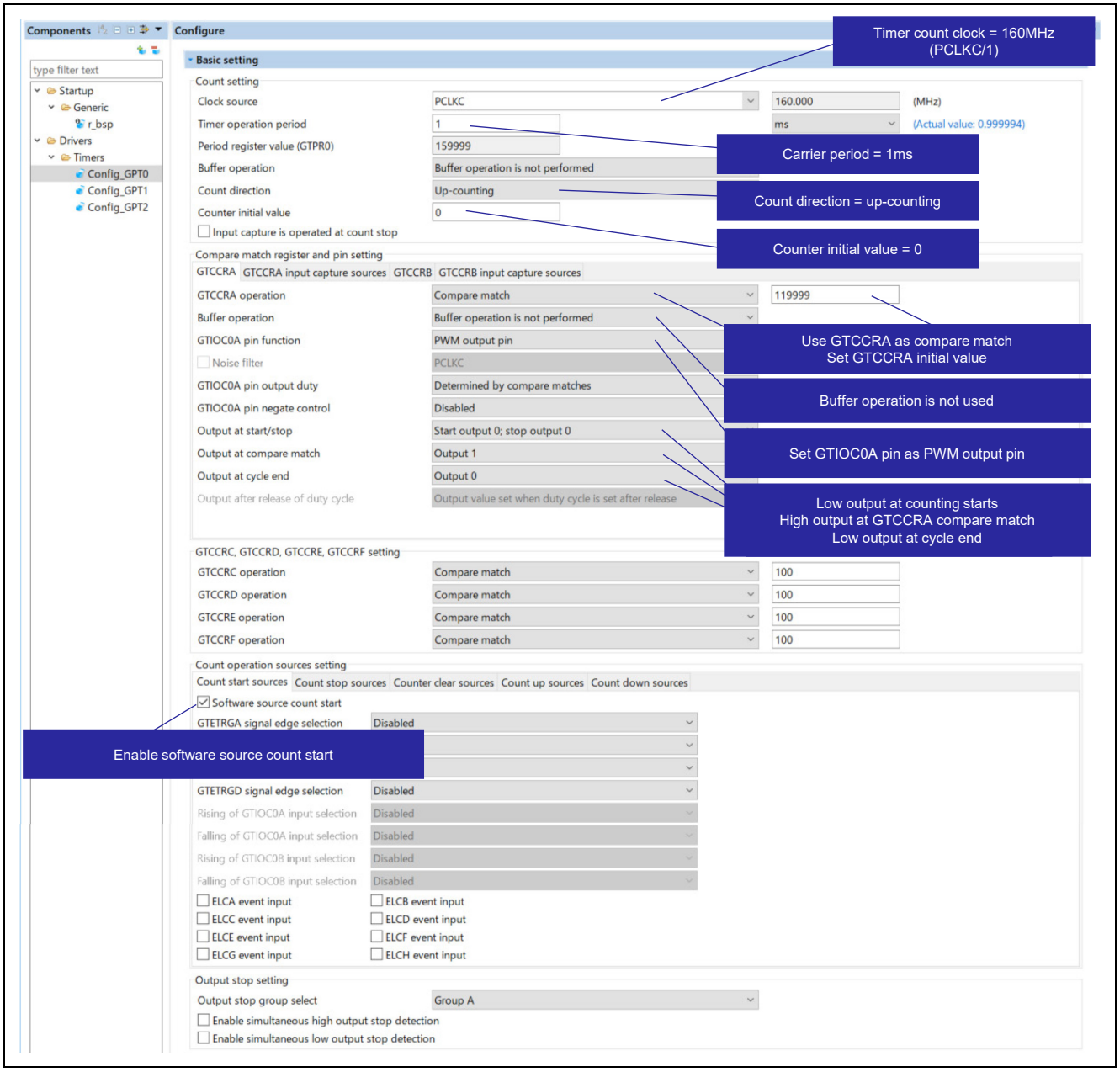


Figure 4.26 GPT0 Setting

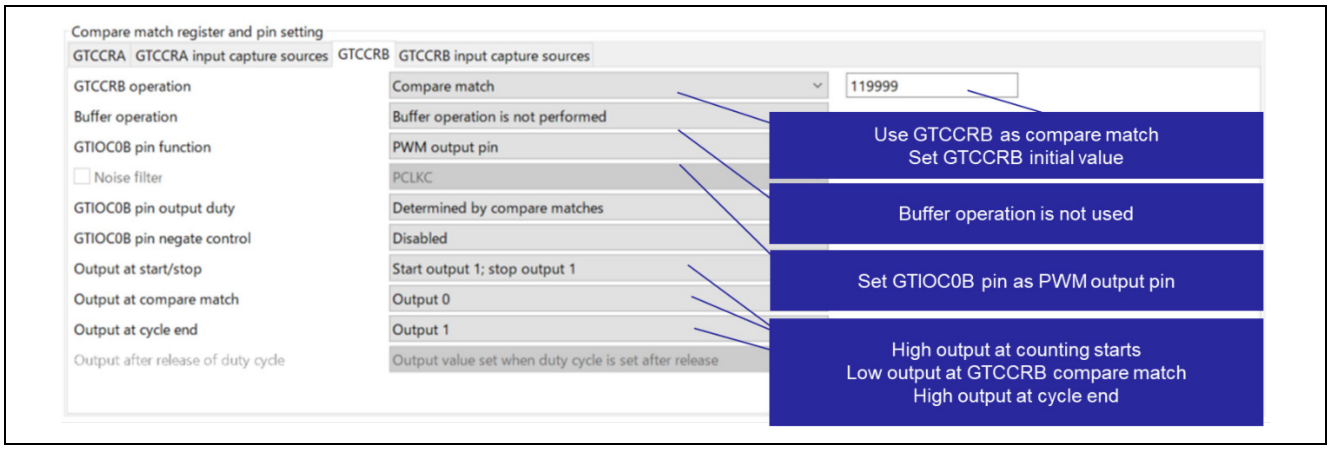


Figure 4.27 GPT0 Setting (Compare Match Register and Pin Setting of GTCCRB)

4.4.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

In the main function, count start function gpt0_gpt1_gpt2_start is read and counting is started.

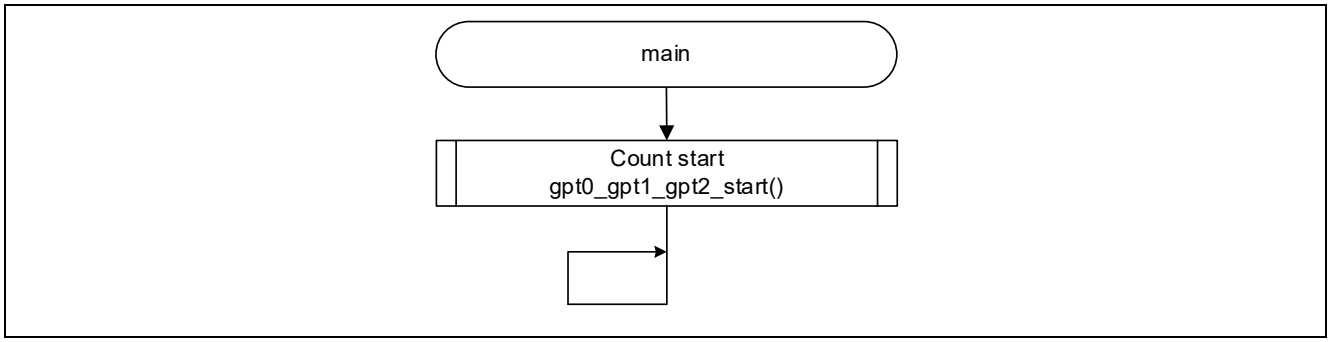


Figure 4.28 main Function

The GPT0, GPT1, and GPT2 counting is started in the count start function.

This function is newly created after code generation by the Smart Configurator.

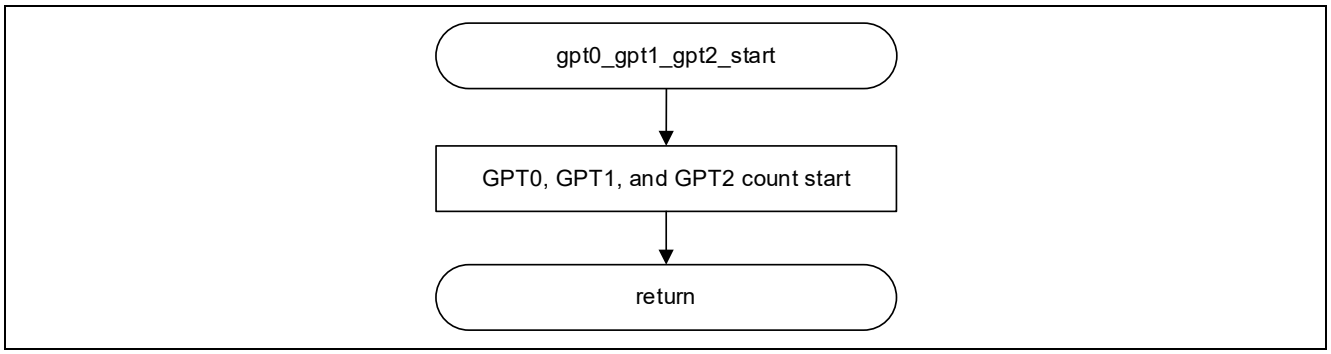


Figure 4.29 Count Start Function

4.4.5 Usage Notes

4.4.5.1 Counting Starts for Multiple Channels

In this sample code, the CSTRT0, CSTRT1, and CSTRT2 bits of timer software start register GTSTR are set at the same time in the gpt0_gpt1_gpt2_start function to start GPTW0, GPTW1, and GPTW2 counting at the same time.

When using the R_Config_GPT0_Start, R_Config_GPT1_Start, and R_Config_GPT2_Start functions generated by the Smart Configurator, the counting starts timing may not be the same because each of the functions are read.

Refer to RX66T Group User's Manual: Hardware, section 24.3.8.1 Synchronous Operation by Software.

4.4.5.2 Settings of GTCCRm Register during Compare Match Operation (m = A to F)

Set compare registers GTCCRA and GTCCRB to values higher than 0000 0001h but less than the setting value of the GTPR register. If the registers are set to 0000 0000h or the same value as the GTPR register, a compare match occurs within the cycle only when the compare match register value is 0000 0000h or the compare register is set to the same value as the GTPR register. If the compare register is set to a value that exceeds the setting value of the GTPR register, no compare match occurs.

For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F, (5) In Sawtooth-wave PWM Mode.

4.5 Sawtooth-Wave One-Shot Pulse 3-Phase Complementary

- Target sample code file name: r01an5995_rx66t_gptw_sawtooth_1shotpls_3phase_dt.zip

4.5.1 Overview

Channels 0, 1, and 2 can be used in the GPTW sawtooth-wave one-shot pulse mode to output 3-phase complementary PWM waveforms with dead time.

This sample code describes a sample code that uses the automatic dead time setting function in the sawtooth-wave one-shot pulse mode and repeats the following waveform output. The double buffer is used to generate each duty cycle with laterally asymmetric PWM waveforms.

- U-phase duty switching: 20% → 40% → 60% → 80% → 60% → 40% → ⋯
- V-phase duty switching: 40% → 60% → 80% → 60% → 40% → 20% → ⋯
- W-phase duty switching: 60% → 80% → 60% → 40% → 20% → 40%→ ⋯

Duty cycle is changed by transferring the value of the temporary register A to the compare register GTCCRA at a compare match of compare register GTCCRA and transferring the value of buffer register GTCCRD to temporary register A and the value of buffer register GTCCRC to compare register GTCCRA when a GTCNT counter overflow occurs.

The following list provides the GPTW settings used in the sample code.

- Use sawtooth-wave one-shot pulse mode
- Use channels 0, 1 and 2 (channel numbers: n = 0, 1, and 2)
- Carrier period = 1ms
- Timer count clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Counter up-counts from initial value 0
- Use GTCCRA as duty output compare match
  - Use GTIOCnA pin as PWM output pin
  - High output at counting starts, high output at counting stops
  - Toggle output at GTCCRA compare match
  - Retain output at cycle end
- Use GTCCRB as duty output compare match
  - Use GTIOCnB pin as PWM output pin
  - Low output at counting starts, low output at counting stops
  - Toggle output at GTCCRB compare match
  - Retain output at cycle end
- Use buffer register
  - GTCCRA and GTCCRB operate as double buffer
  - Use GTCCRC and GTCCRD as buffer registers of GTCCRA
  - Use GTCCRE and GTCCRF as buffer registers of GTCCRB
- Use automatic dead time generation
- Software source count start enabled
- Duty changes at each cycle
  - Duty changes at the GTCNT counter overflow interrupt
  - Refer to Figure 4.32 for details on duty change timing

Set in Smart Configurator.
For Setting Methods,
refer to section 4.5.3.

Sawtooth-wave one-shot pulse mode output for this sample code is shown below.

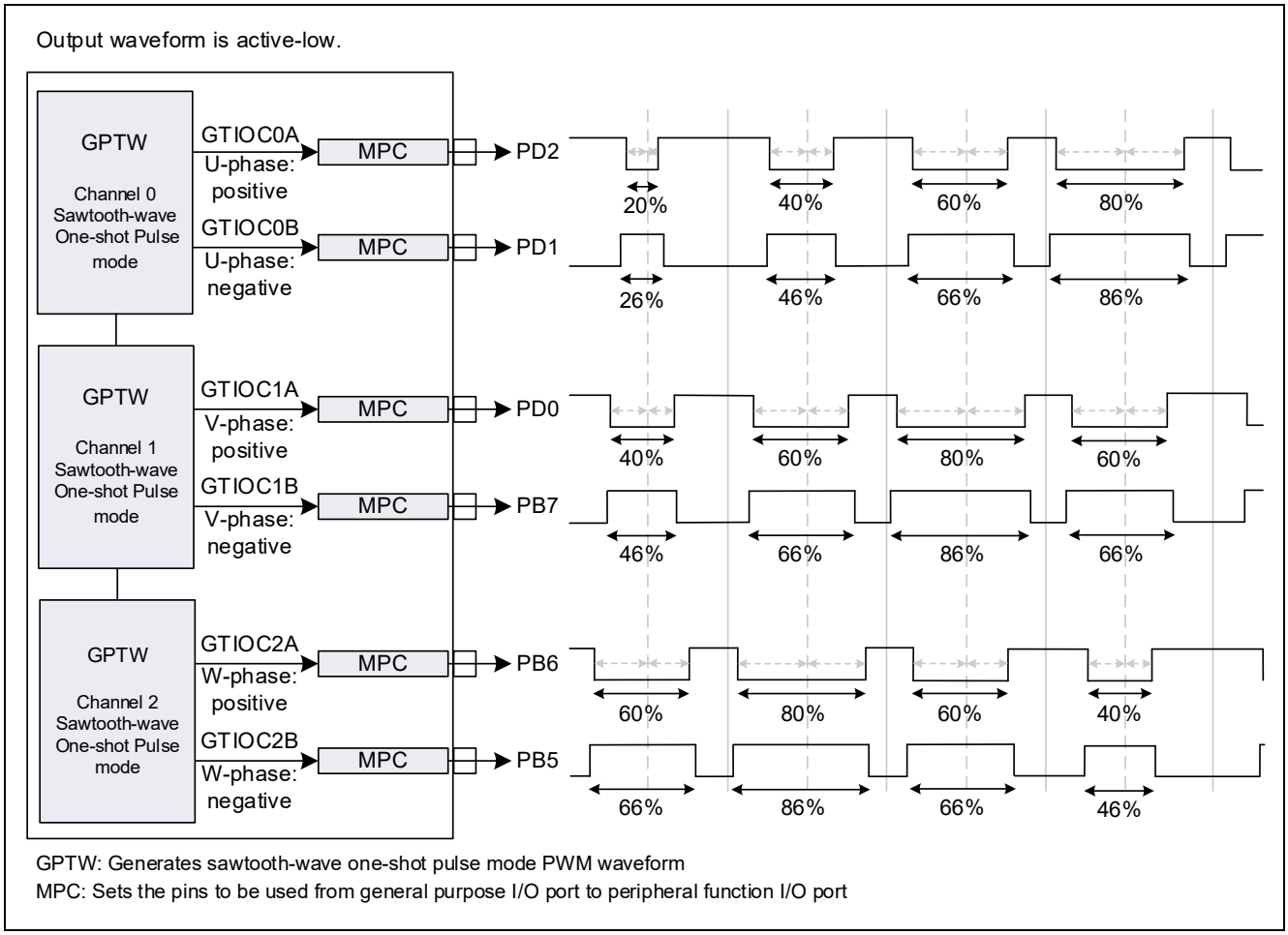


Figure 4.30 Sawtooth-wave One-shot Pulse Mode Output

4.5.2 Operation Details

The sample code operations are shown in Figure 4.32. The settings of the duty cycle are changed with each period by modifying the values of buffer registers GTCCRC and GTCCRD at a GTCNT counter overflow interrupt (GTCIV0) ((1) in Figure 4.32).

In this sample code, the initial values of the buffer registers for the code generated using the Smart Configurator are set to the same values as the compare register. As a result, the buffer register values are set in the user initialization function R_Config_GPT0_Create_UserInit before the counting starts.

The values set in the registers are transferred from buffer register GTCCRD to temporary register A and from buffer register GTCCRC to compare register GTCCRA, respectively, by forced buffer transfers ((2) in Figure 4.32).

After the counting starts, steps 1 and 2 below are repeated to update the data in the compare register.

1. Transfer from temporary register A to compare register GTCCRA when a GTCCRA compare match occurs ((3) in Figure 4.32).
2. Transfer from buffer register GTCCRD to temporary register A and from buffer register GRCCRC to compare register GTCCRA, respectively, when a GTCNT counter overflow occurs ((4) Figure 4.32).

In addition, the GTCCRB register is automatically set according to the GTCCRA update because the automatic dead time function is used. The same values are set for the GTDVU and GTDVD.

After counting starts, GTCCRA and GTCCRB compare matches occur, the negative-phase output turns OFF and the positive-phase output turns ON ((5) in Figure 4.32).

- Laterally Asymmetric PWM Waveform Output
  The duty cycle in each period generates a different duty cycle for the first and second half of each period.

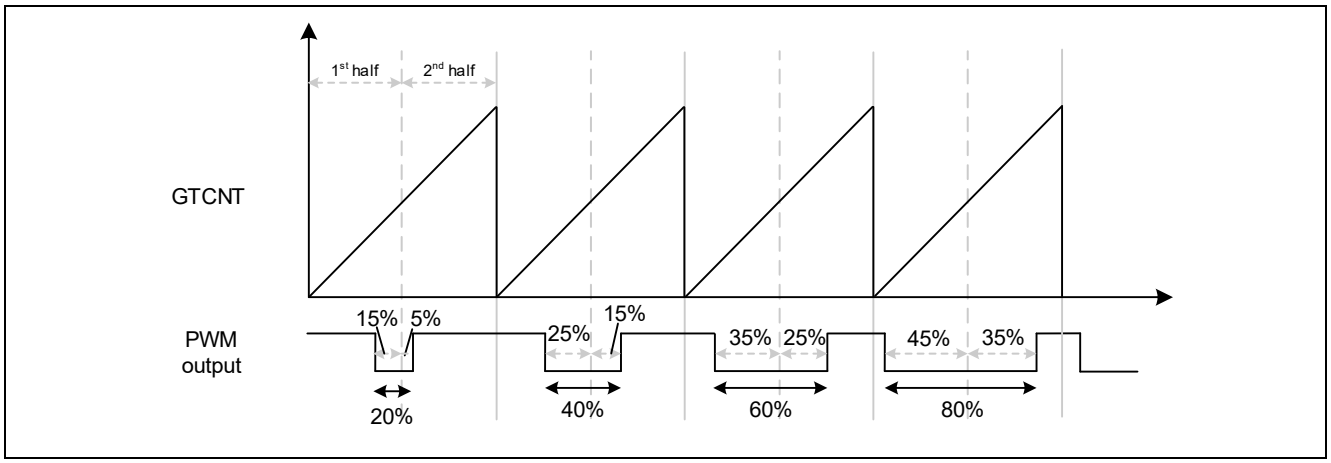
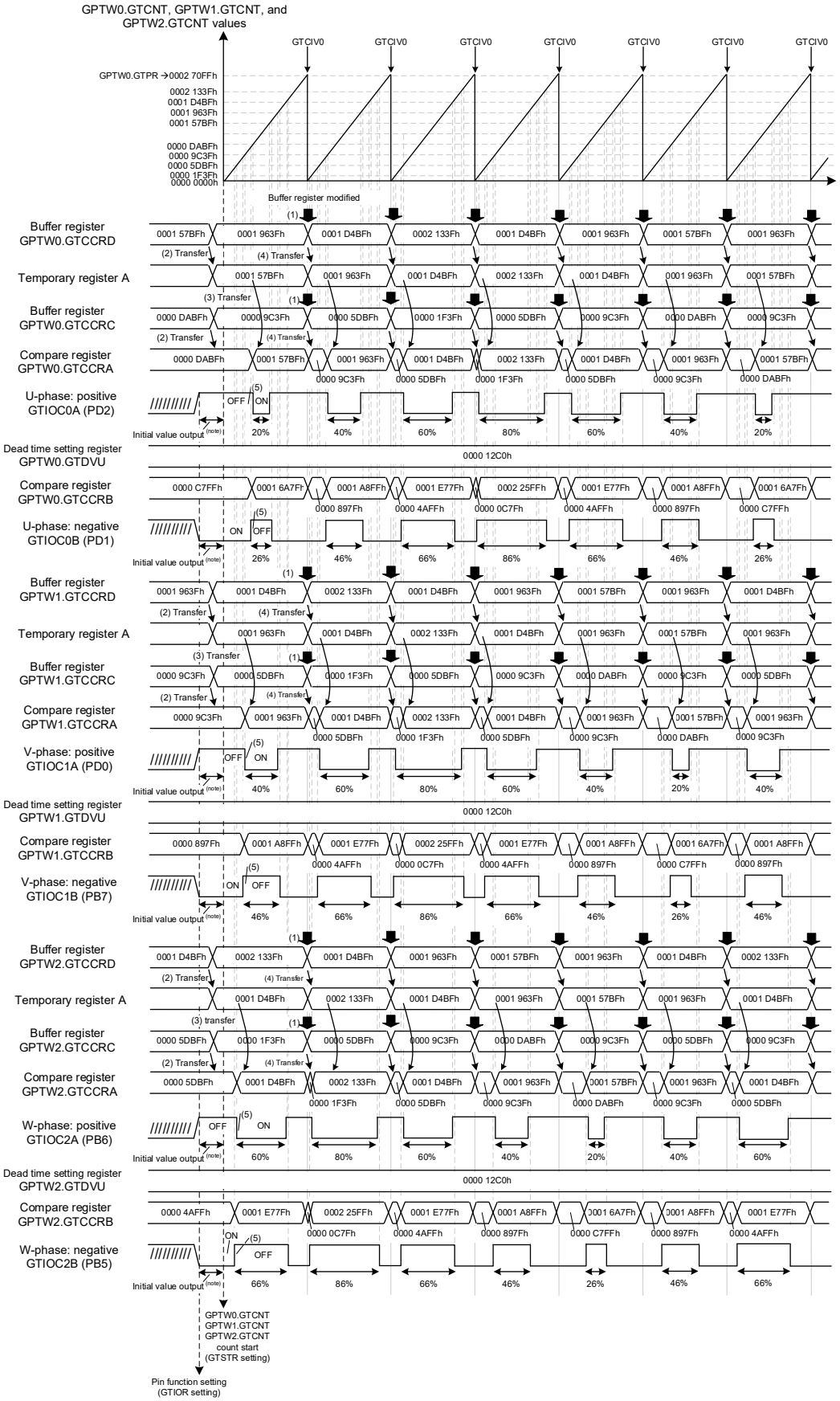


Figure 4.31 Laterally Asymmetric PWM Waveform Output



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.32 Sample Code Operations

4.5.3 Smart Configurator Settings

The sample codes use the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.8 Adding Components

Item	Description		
Component	General PWM Timer		
Configuration name	Config_GPT0	Config_GPT1	Config_GPT2
Work mode	Sawtooth-wave One-shot Pulse Mode		
Resource	GPT0	GPT1	GPT2

Figure 4.33 to Figure 4.35 show the Config_GPT0 settings. The same settings apply for GPT1 and GPT2.

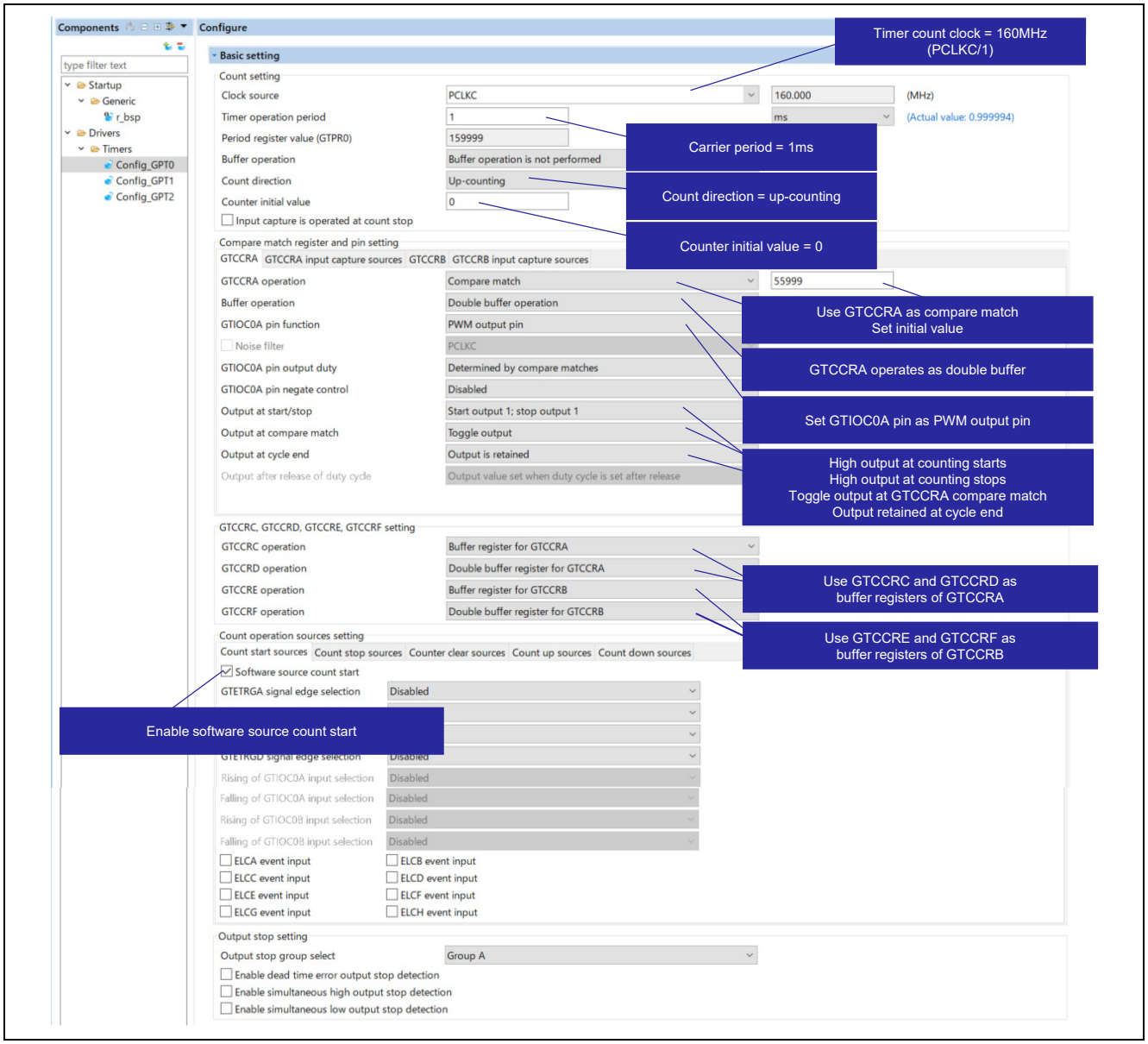


Figure 4.33 GPT0 Settings (1/2)

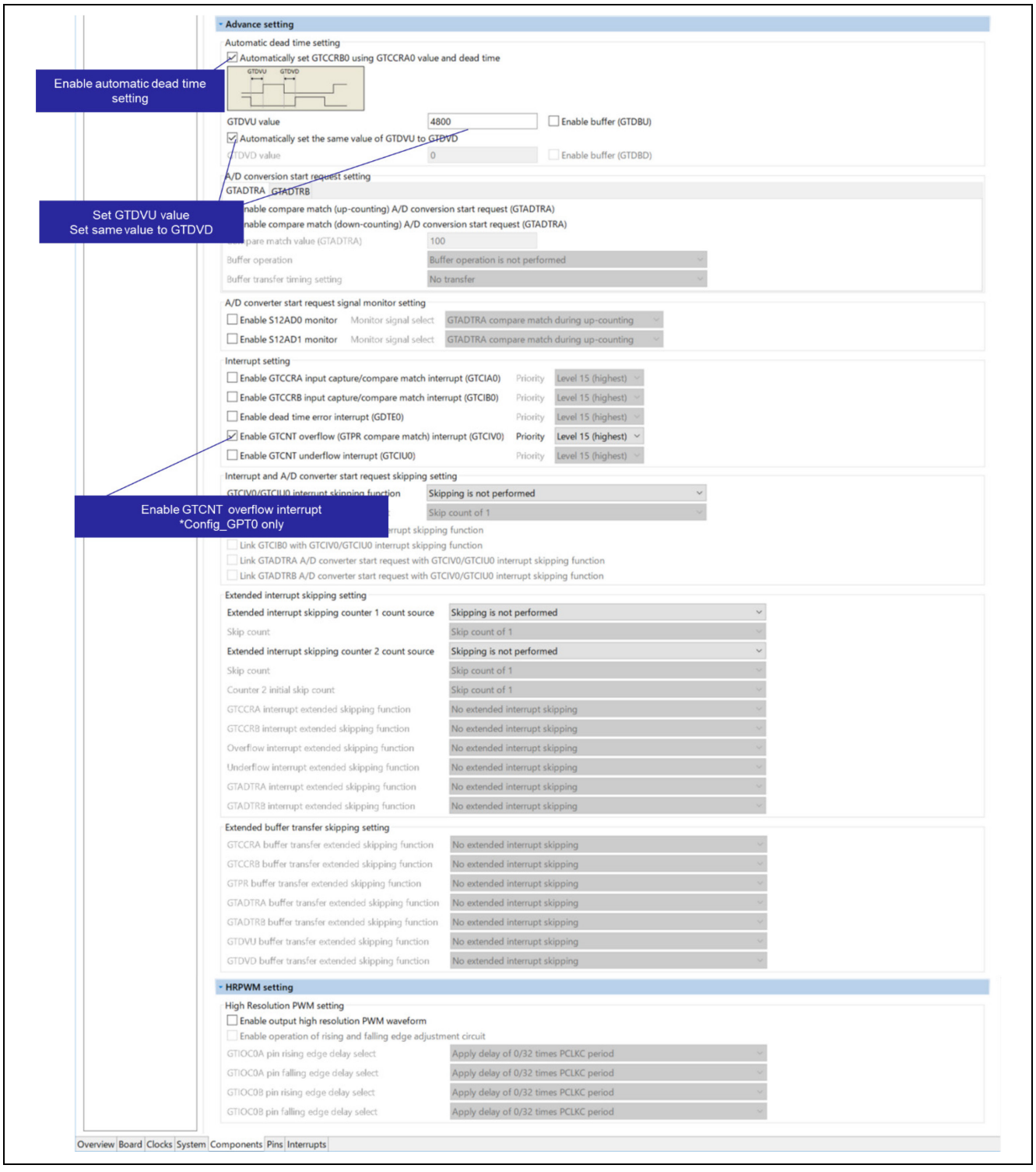


Figure 4.34 GPT0 Settings (2/2)

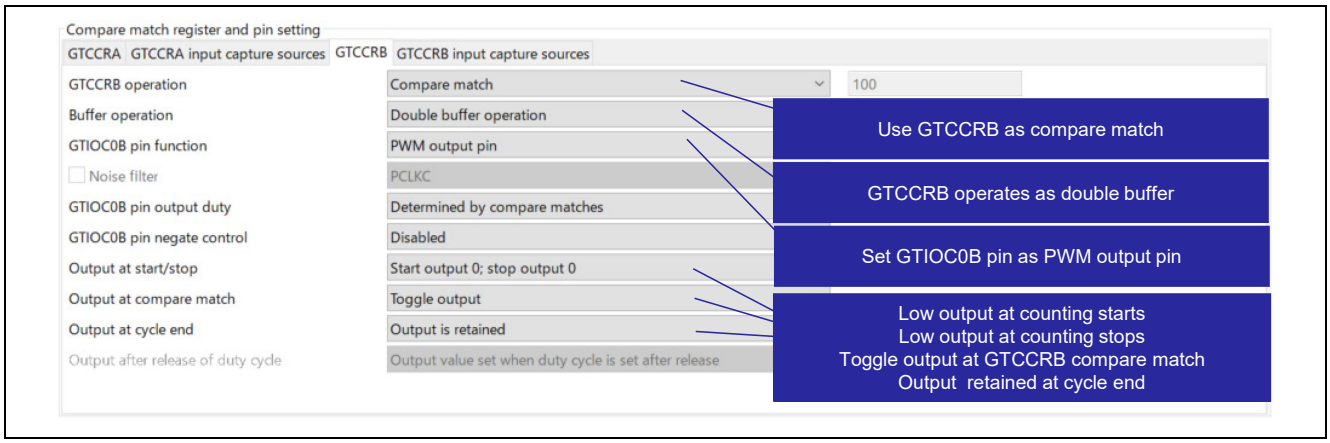


Figure 4.35 GPT0 Settings (Compare Match Register and Pins Setting of GTCCRB)

4.5.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

In the main function, count start function gpt0_gpt1_gpt2_start is read and counting is started.

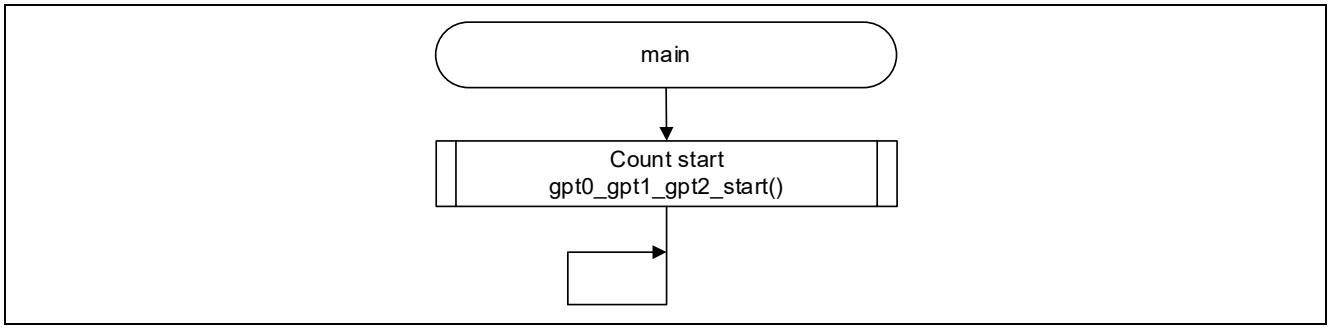


Figure 4.36 main Function

In the count start function, the GPT0, GPT1, and GPT2 counting is started after the GTCIV0 interrupt is enabled.

This function is newly created after code generation by the Smart Configurator.

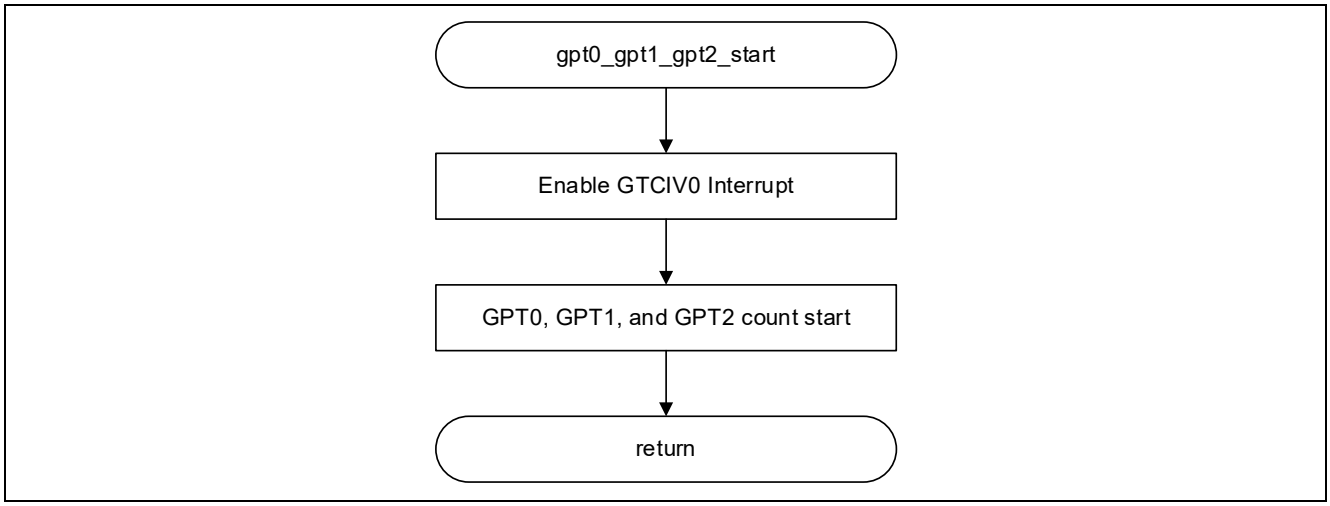


Figure 4.37 Count Start Function

The user initialization function R_Config_GPT0_Create_UserInit, which is executed before the main function, sets the initial values of the buffer registers and initializes the variables. This function is called from within the R_Config_GPT0_Create function.

This sample code uses the following variable.

- s_uptmg_prv: variable for retaining the previous GPTW0.GTCCRC register value (at GTIOC0A pin output rising time)

R_Config_GPT1_Create_UserIni and R_Config_GPT2_Create_UserInit also perform the same processes.

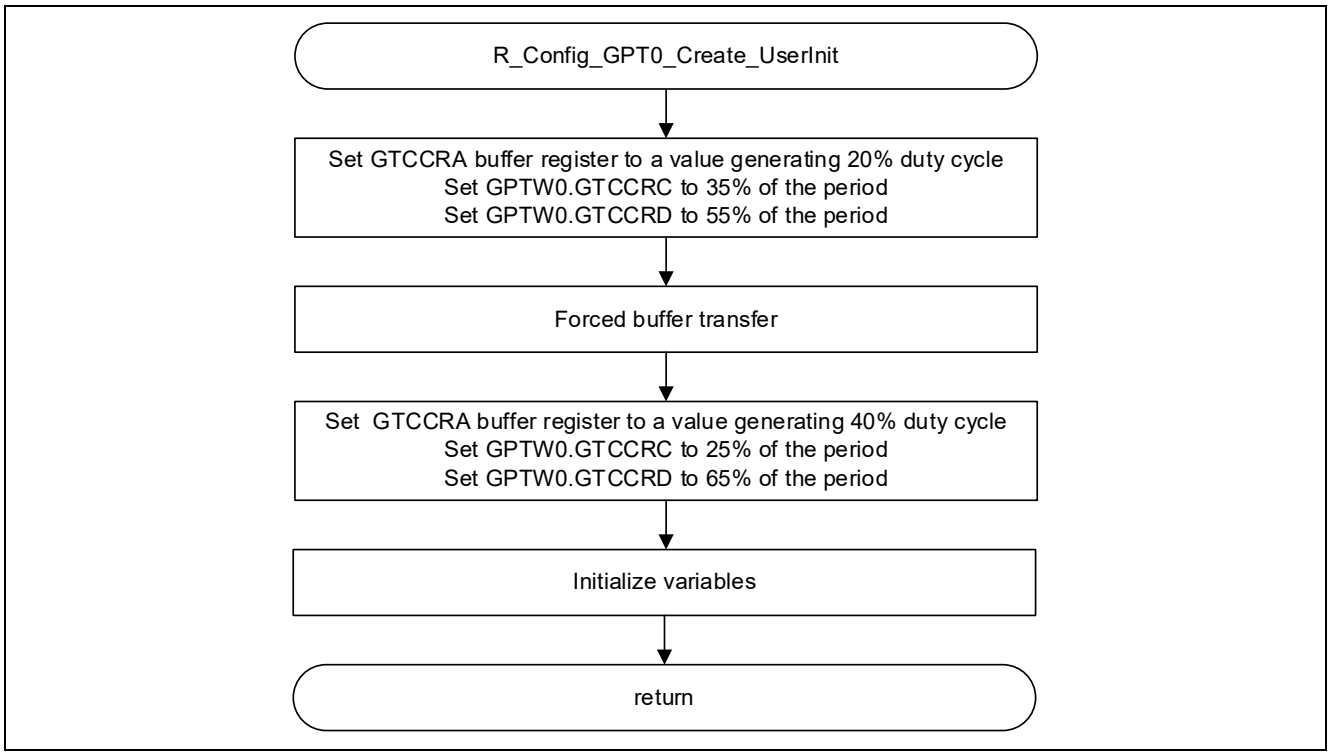


Figure 4.38 User Initialization Function

The GTCIV0 interrupt handler function changes the values of the buffer registers according to the current value of buffer registers GPTW0.GTCCRC, GPTW1.GTCCRC, and GPTW2.GTCCRC and the values set in the previous buffer registers.

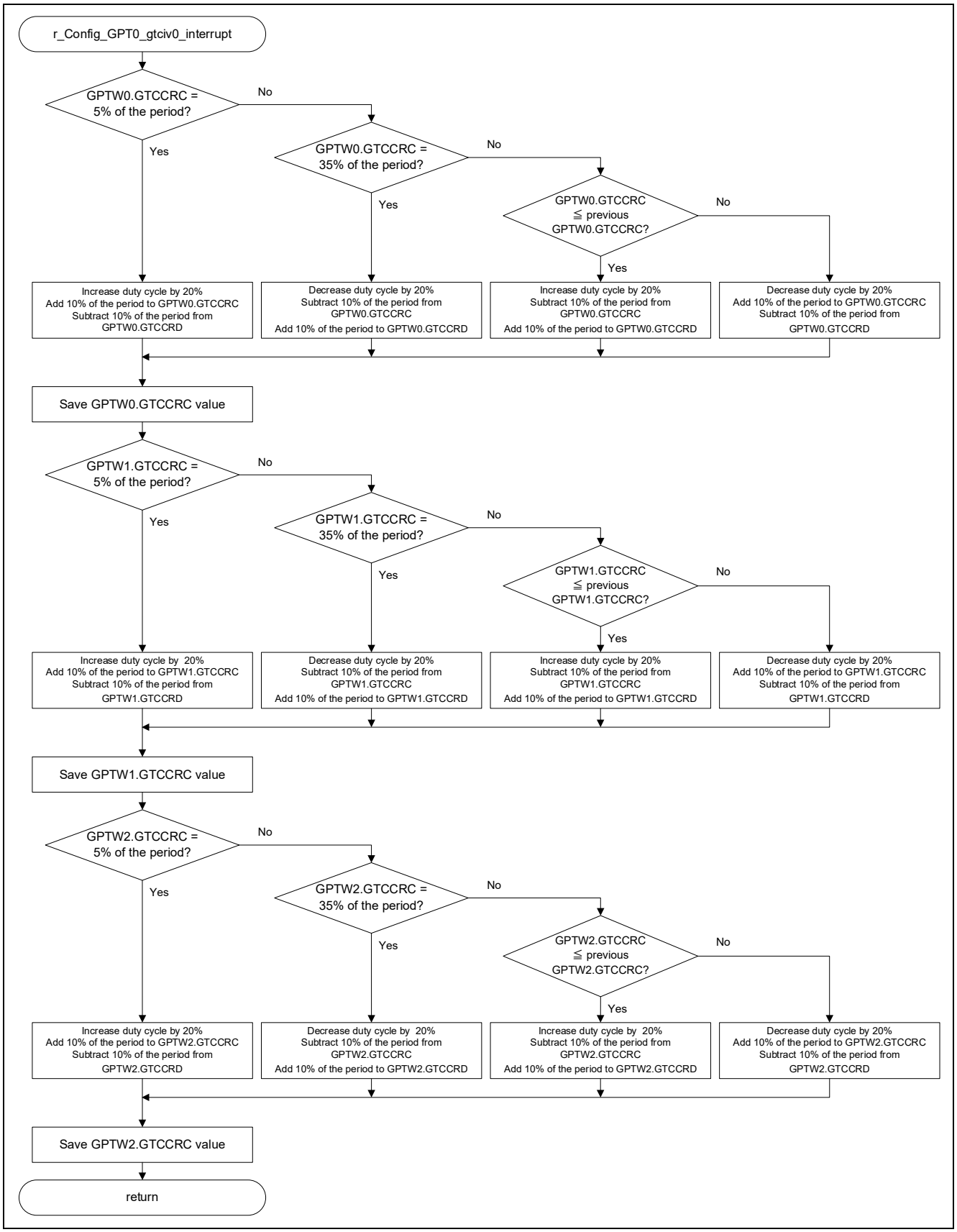


Figure 4.39 GTCIV0 Interrupt Handler Function

4.5.5 Related Operations

4.5.5.1 Separate Automatic Dead Time Settings for Each Interval

This sample code uses the automatic dead time setting function and a dead time is generated with a common switching point in the first half and second half of the negative-phase by setting the GTDTCR.TDFER bit to 1.

In the automatic dead time setting function, the dead time can be separately set for the first half and second half of a waveform. Dead time for the changing point in the first half of a negative-phase is set in the GTDVU register and that in the second half is set in the GTDVD register.

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.4 Automatic Dead Time Setting Function.

4.5.6 Usage Notes

4.5.6.1 Counting Starts for Multiple Channels

In this sample code, the CSTRT0, CSTRT1, and CSTRT2 bits of timer software start register GTSTR are set at the same time in the gpt0_gpt1_gpt2_start function to start the GPTW0, GPTW1, and GPTW2 counting at the same time.

When using the R_Config_GPT0_Start, R_Config_GPT1_Start, and R_Config_GPT2_Start functions generated by the Smart Configurator, the counting starts timing may not be the same because each of the functions are read.

Refer to RX66T Group User's Manual: Hardware, section 24.3.8.1 Synchronous Operation by Software.

4.5.6.2 Settings of GTCCRm Register during Compare Match Operation (m = A to F)

This sample code automatically sets the dead time and uses the value of compare match register GTCCRA, the register for positive-phase waveform, to update the value of compare match register GTCCRB, the register for negative-phase waveform.

The values of buffer registers GTCCRC and GTCCRD should be set to satisfy the following restrictions. If the restrictions are not satisfied, correct normal output waveform with secured dead time may not be obtained.

- In up-counting:
  GTCCRC < GTCCRD
  GTCCRC > GTDVU
  GTCCRD < GTPR – GTDVD
- In down-counting:
  GTCCRC > GTCCRD
  GTCCRC < GTPR – GTDVU
  GTCCRD > GTDVD

Further, if the dead time is not automatically set, buffer registers GTCCRC and GTCCRD (or GTCCRE and GTCCRF) should be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC (GTCCRE) < GTCCRD (GTCCRF) < GTPR
- In down-counting: GTPR > GTCCRC (GTCCRE) > GTCCRD (GTCCRF) > 0

For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F), (3) When Automatic Dead Time Setting has been Made in Sawtooth-Wave One-Shot Pulse Mode and (4) When Automatic Dead Time Setting has not been Made in Sawtooth-Wave One-Shot Pulse Mode.

4.6 Triangle-Wave PWM Mode 1

- Target sample code file name: r01an5995_rx66t_gptw_triangle_pwm1_dt.zip

4.6.1 Overview

GPTW triangle-wave PWM mode 1 can be used to output 3-phase complementary PWM waveforms with dead time.

This sample code describes a sample code that that uses the automatic dead time setting function in triangle-wave PWM mode 1 (32-bit transfer at trough) and repeats the following output waveforms. Each duty cycle generates symmetric PWM waveforms using the buffer (not a double buffer).

- U-phase duty switching: 20% → 40% → 60% → 80% → 60% → 40% → ⋯
- V-phase duty switching: 40% → 60% → 80% → 60% → 40% → 20% → ⋯
- W-phase duty switching: 60% → 80% → 60% → 40% → 20% → 40%→ ⋯

The duty cycle is changed by transferring the value of buffer register GTCCRC to compare register GTCCRA when a GTCNT counter underflow occurs.

The following list provides the GPTW settings used in the sample code.

- Use triangle-wave PWM mode 1
- Use channels 0, 1 and 2 (channel numbers: n = 0, 1, and 2)
- Carrier period = 1ms
- Timer count clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Counter up-counts from initial value 0
- Use GTCCRA as duty output compare match
  - Use GTIOCnA pin as PWM output pin
  - Use GTCCRA as compare match
  - High output at counting starts, high output at counting stops
  - Toggle output at GTCCRA compare match
  - Retain output at cycle end
- Use GTCCRB as duty output compare match
  - Use GTIOCnB pin as PWM output pin
  - Use GTCCRB as compare match
  - Low output at counting starts, low output at counting stops
  - Toggle output at GTCCRB compare match
  - Retain output at cycle end
- Use buffer register
  - GTCCRA operates as single buffer
  - Use GTCCRC as buffer register of GTCCRA
- Use automatic dead time generation
- Software source count start enabled
- Duty changes at each cycle
  - Duty changes at the GTCNT counter underflow interrupt
  - Refer to Figure 4.42 for details on duty change timing

Set in Smart Configurator.
For Setting Methods,
refer to section 4.6.3.

Triangle-wave PWM mode 1 output for this sample code is shown below.

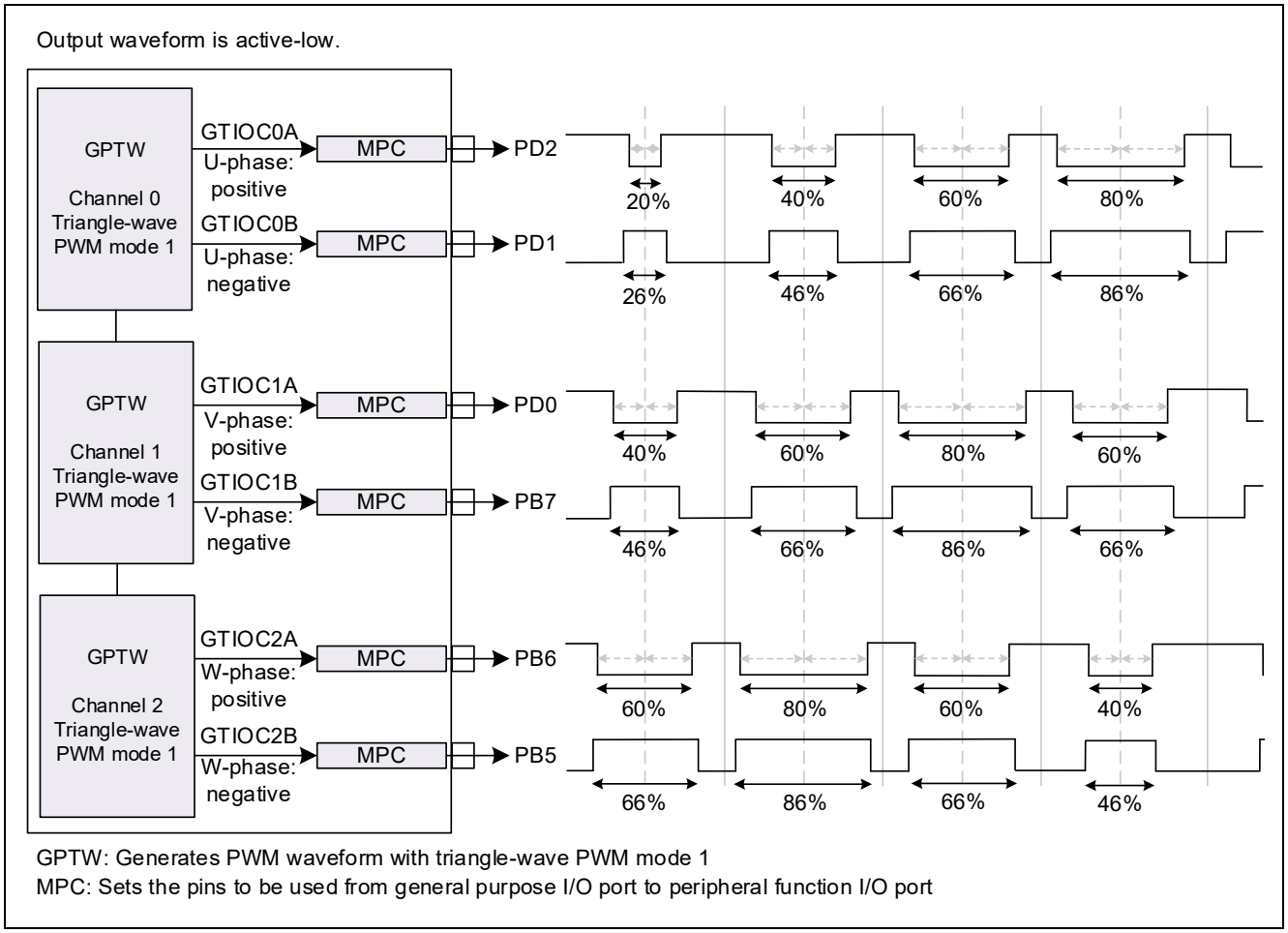


Figure 4.40 Triangle-wave PWM Mode 1 Output

4.6.2 Operation Details

The sample code operations are shown in Figure 4.42. The settings of the duty cycle are changed with each period by modifying the value of buffer register GTCCRC at the GTCNT counter underflow interrupt (GTCIU0) ((1) in Figure 4.42).

This sample code uses triangle-wave PWM mode 1, which transfers at the trough, to update data by transferring the value of the buffer register to compare register GTCCRA when a GTCNT counter underflow occurs ((2) in Figure 4.42).

In addition, the GTCCRB register is automatically set according to the GTCCRA update because the automatic dead time function is used. The same values are set for the GTDVU and GTDVD. This sample code automatically sets the GTCCRB value at counting starts ((3) in Figure 4.42).

After counting starts, a compare match occurs between the compare register and the counter register, negative-phase output turns OFF, and then positive-phase output turns ON ((4) in Figure 4.42)).

- Symmetric PWM Waveform Output
  The duty cycle in each period generates the1/2 duty cycle for the up-counting period and down-counting period.

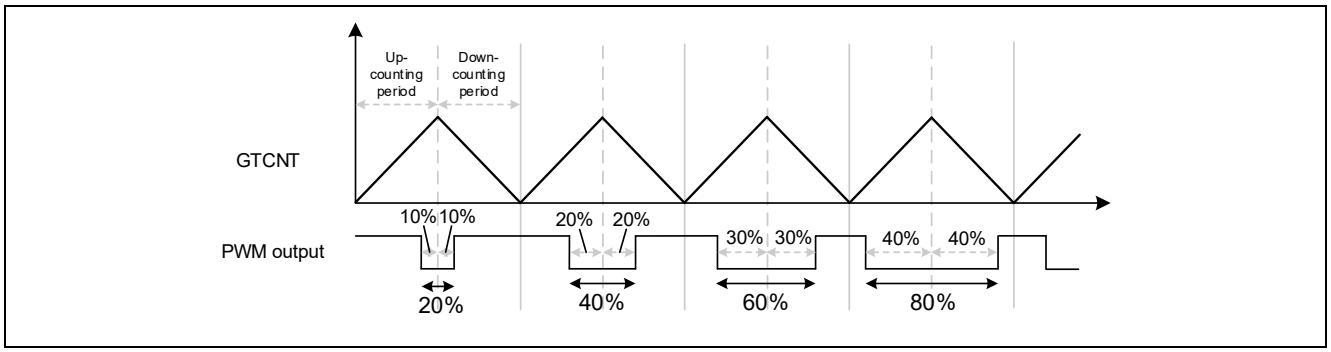
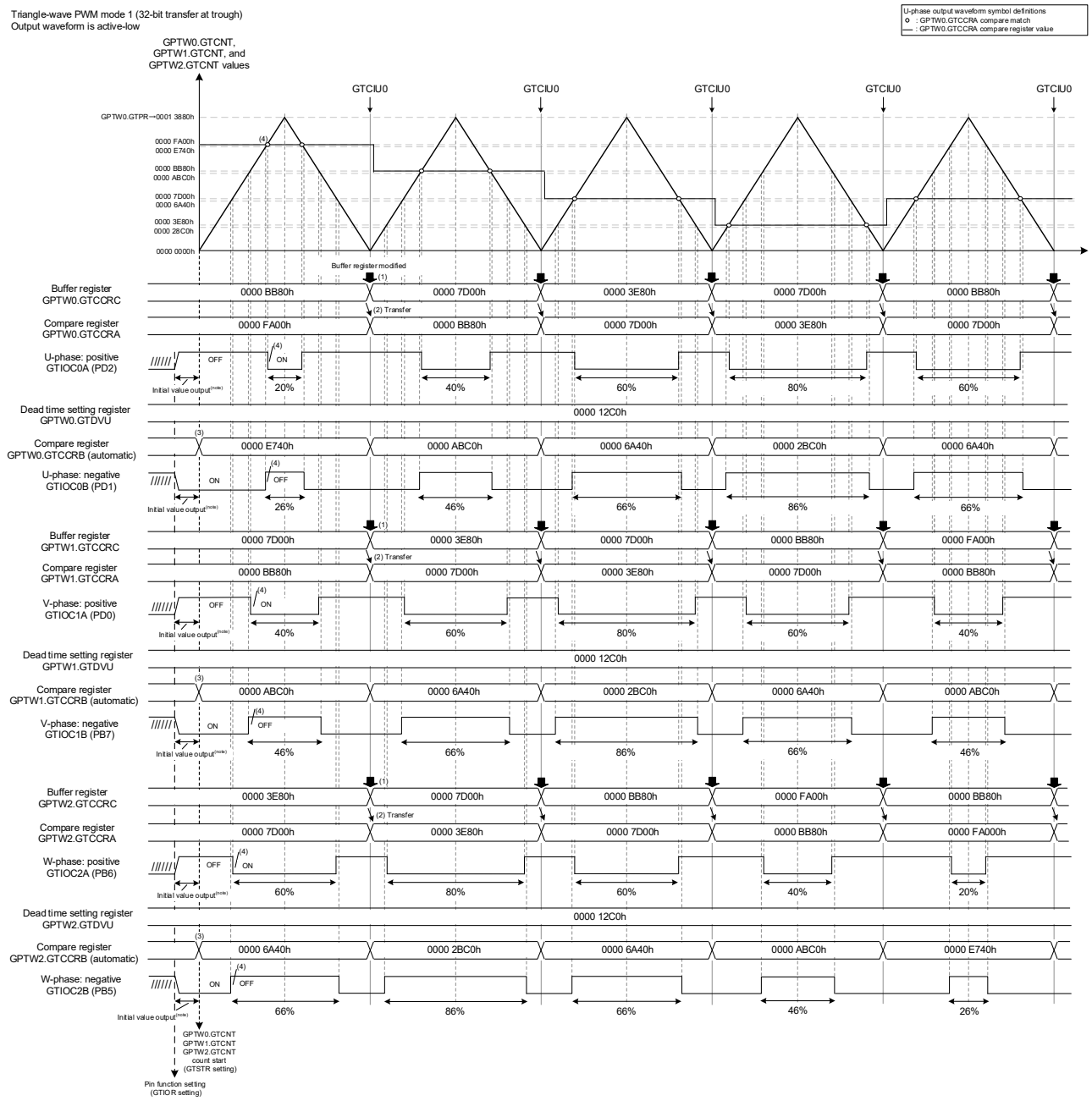


Figure 4.41 Symmetric PWM Output Waveform



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.42 Sample Code Operations

4.6.3 Smart Configurator Settings

The sample codes use the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.9 Adding Components

Item	Description		
Component	General PWM Timer		
Configuration Name	Config_GPT0	Config_GPT1	Config_GPT2
Work mode	Triangle-wave PWM Mode1		
Resource	GPT0	GPT1	GPT2

Figure 4.43 to Figure 4.45 show the Config_GPT0 settings. The same settings apply for GPT1 and GPT2.

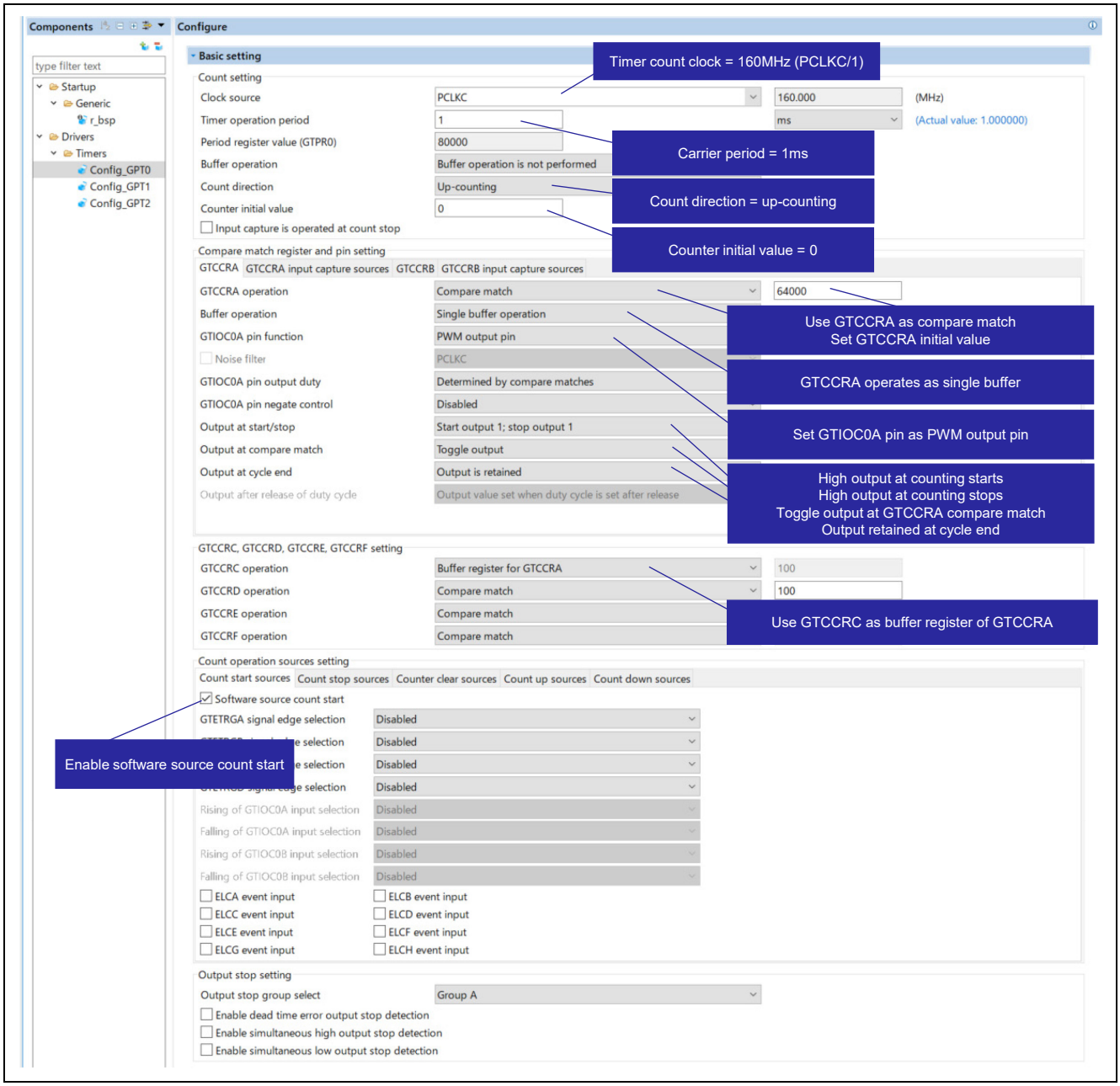


Figure 4.43 GPT0 Settings (1/2)

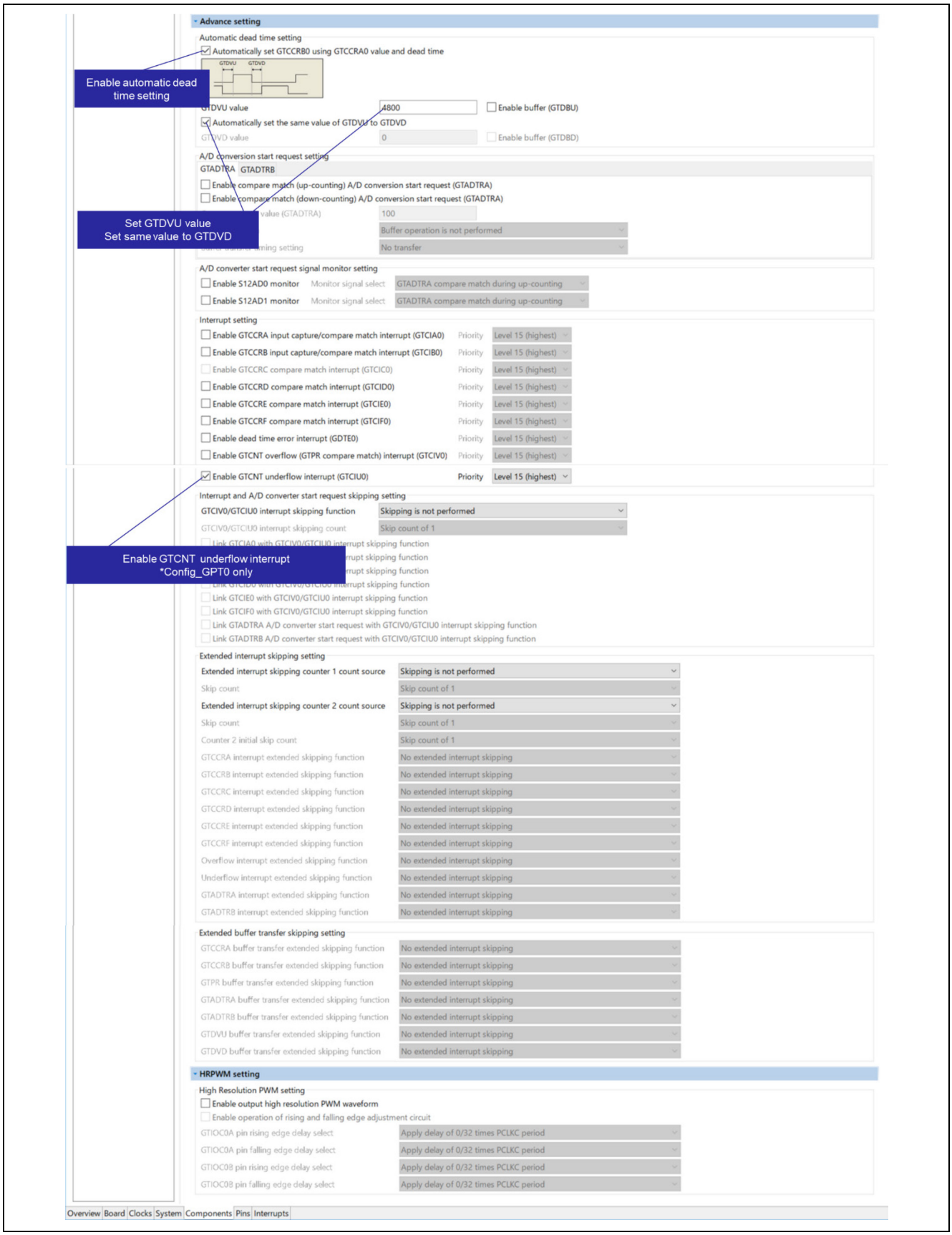


Figure 4.44 GPT0 Settings (2/2)

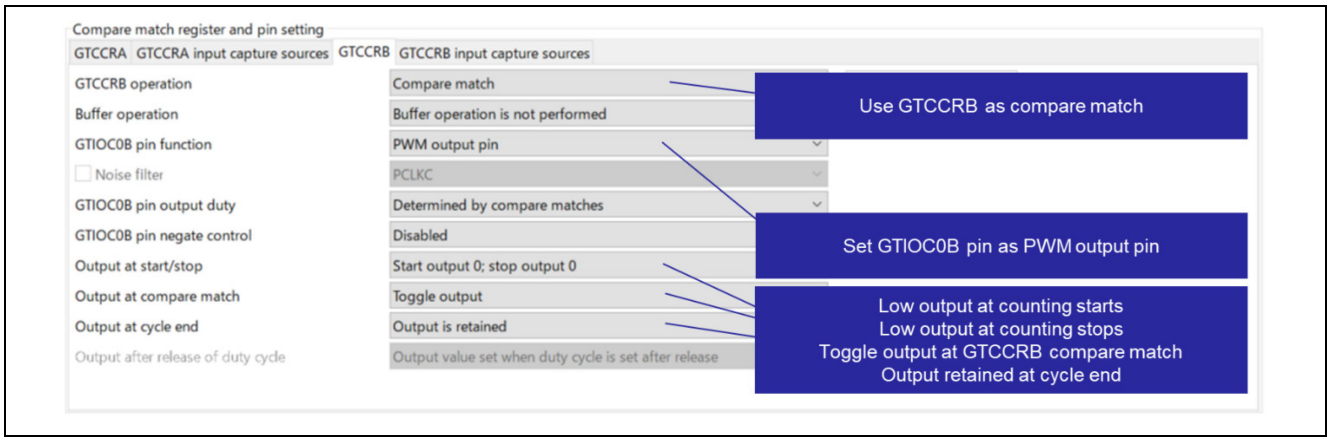


Figure 4.45 GPT0 Settings (Compare Match Register and Pins Setting of GTCCRB)

4.6.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

In the main function, count start function gpt0_gpt1_gpt2_start is read and counting is started.

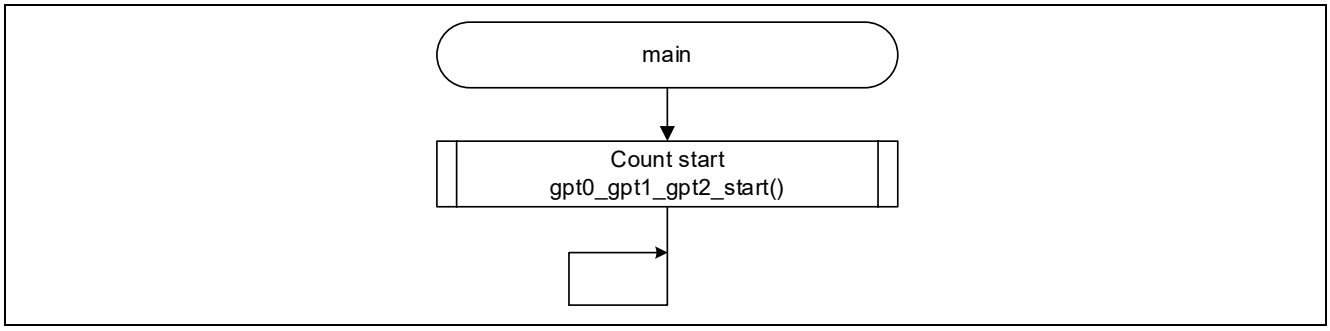


Figure 4.46 main Function

In the count start function, the GPT0, GPT1, and GPT2 counting is started after the GTCIU0 interrupt is enabled.

This function is newly created after code generation by the Smart Configurator.

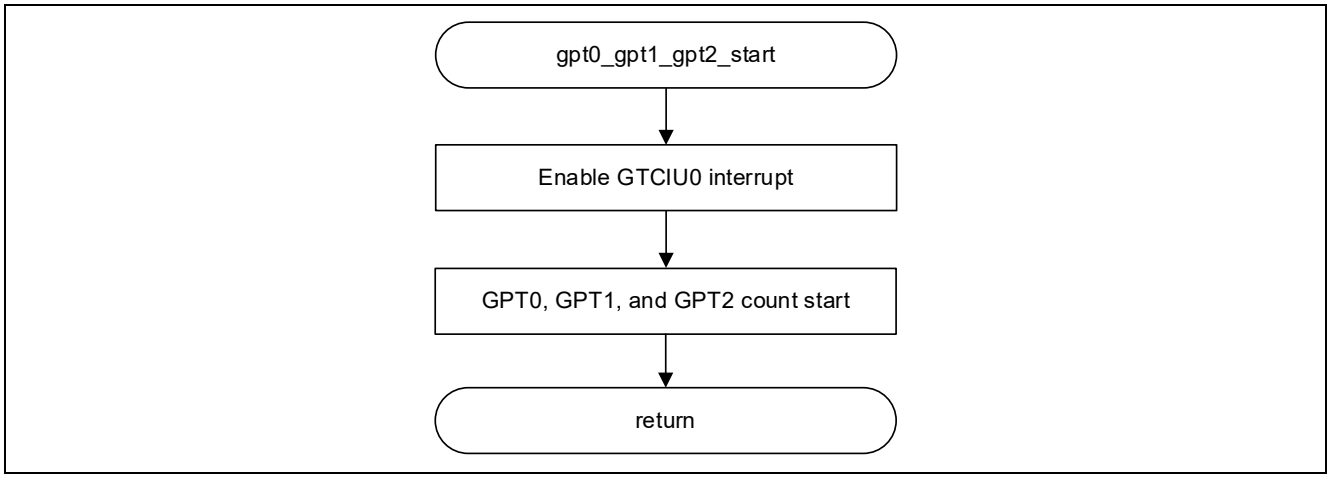


Figure 4.47 Count Start Function

The user initialization function R_Config_GPT0_Create_UserInit, which is executed before the main function, sets the initial values of the buffer registers and initializes the variables. This function is called from within the R_Config_GPT0_Create function.

This sample code uses the following variable.

- g_ucduty_prv0: variable for retaining the previous GPTW0.GTCCRC register value

R_Config_GPT1_Create_UserIni and R_Config_GPT2_Create_UserInit also perform the same processes.

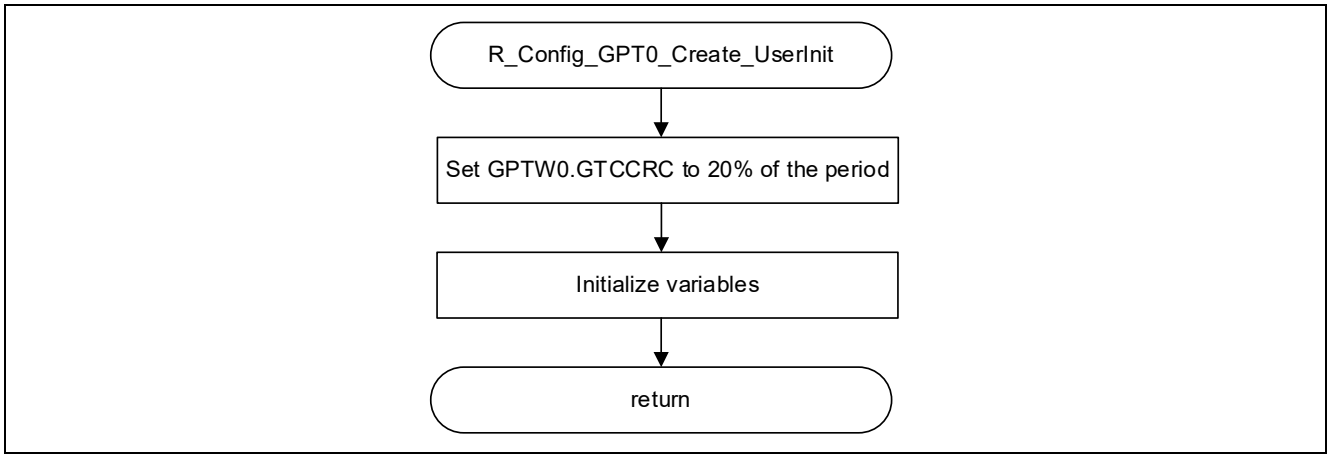


Figure 4.48 User Initialization Function

The GTCIU0 interrupt handler function changes the values of the buffer registers according to the current value of buffer registers GPTW0.GTCCRC, GPTW1.GTCCRC, and GPTW2.GTCCRC and the values set in the previous buffer registers.

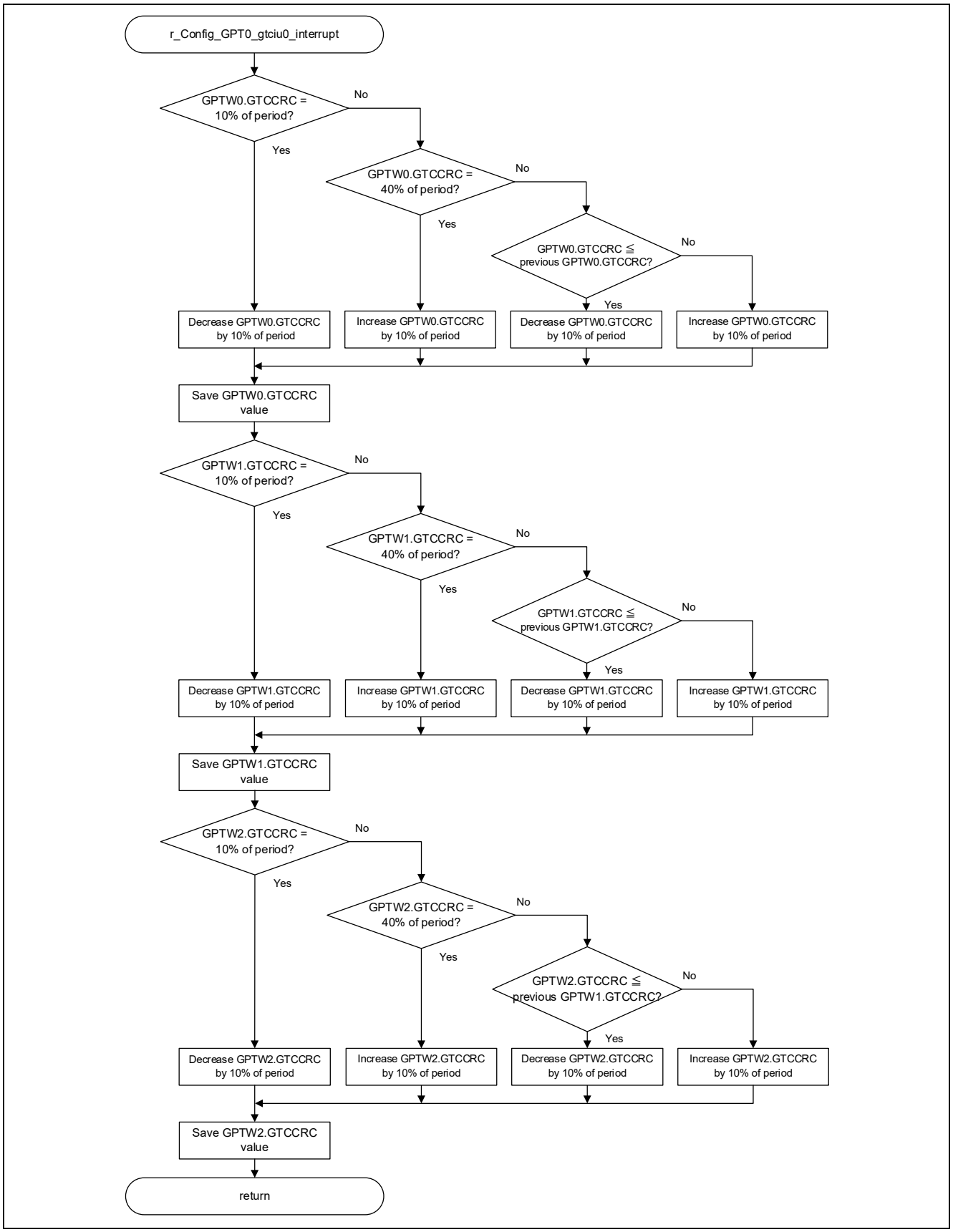


Figure 4.49 GTCIU0 Interrupt Handler Function

4.6.5 Related Operations

4.6.5.1 Separate Automatic Dead Time Settings for Each Interval

This sample code uses the automatic dead time setting function and a dead time is generated with a common switching point in the first half and second half of the negative-phase by setting the GTDTCR.TDFER bit to 1.

In the automatic dead time setting function, the dead time can be separately set for the first half and second half of a waveform. Dead time for the changing point in the first half of a negative-phase is set in the GTDVU register and that in the second half is set in the GTDVD register.

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.4 Automatic Dead Time Setting Function.

4.6.5.2 Buffer Operations with Automatic Dead Time Setting Function

This sample code uses the automatic dead time setting function to generate a negative-phase waveform that secures the fixed dead time period set by the Smart Configurator.

The automatic dead time setting function uses the GTDBU as buffer register GTDVU and the GTDBD as buffer register GTDVD to update the dead time period during the count by performing buffer transfer at the end of the count cycle (when a GTCNT counter underflow occurs (trough)).

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.4 Automatic Dead Time Setting Function.

Smart Configurator settings are as shown below.

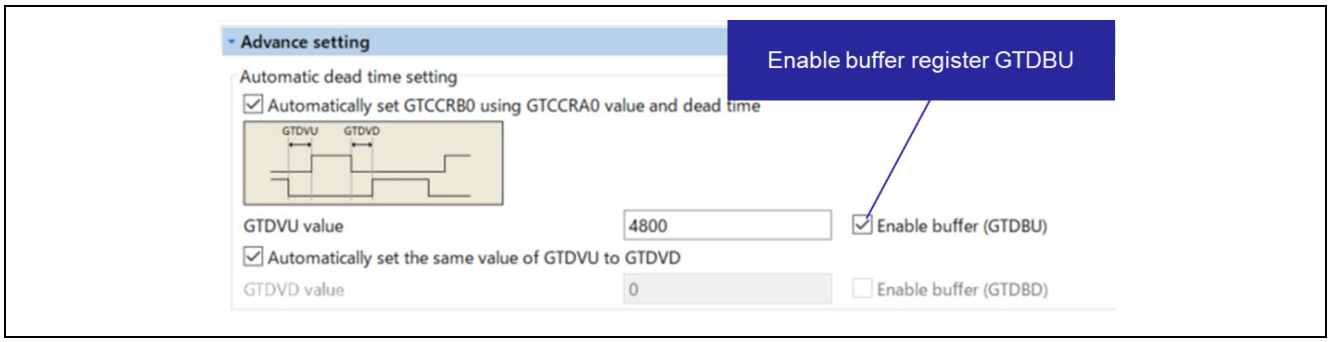
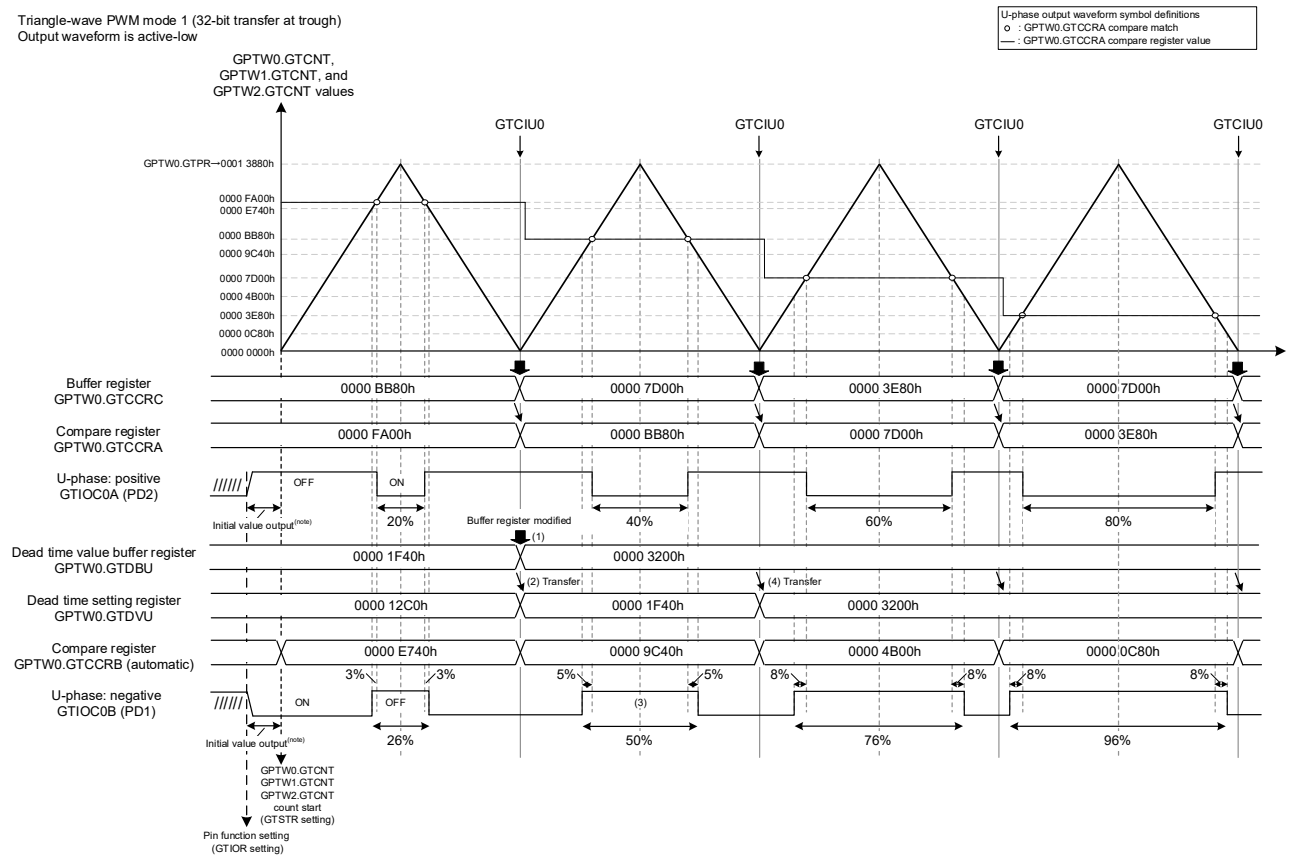


Figure 4.50 Smart Configurator Automatic Dead Time Setting

Figure 4.51 shows an example of operations of the automatic dead time setting function using the buffer register of dead time value.

The value of the GTDBU is modified by the GTCNT counter underflow interrupt (GTCIU0) ((1) in Figure 4.51). The value of the GTDBU is transferred to dead time setting register GTDVU) when a GTCNT counter underflow occurs (at trough) ((2) in Figure 4.51), and the waveform that secures the dead time period after update is output ((3) in Figure 4.51).

In the 3rd cycle, the GTDBU value that was modified at the end of the 1st cycle is transferred to the GTDVU ((4) in Figure 4.51).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.51 Buffer Operations with Automatic Dead Time Setting Function

4.6.5.3 When Automatic Dead Time Setting Function is Not Used

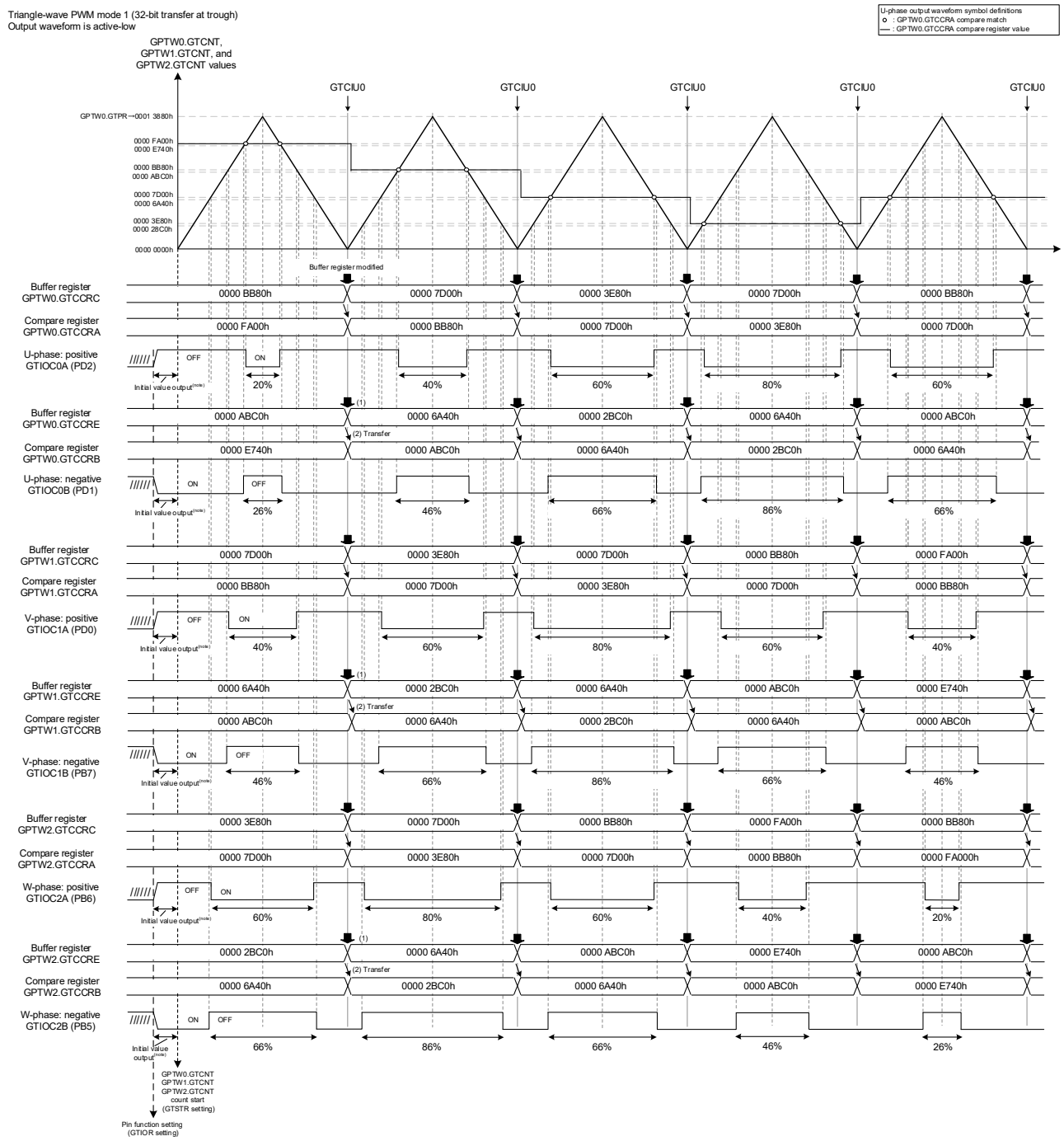
- Target sample code file name: r01an5995_rx66t_gptw_triangle_pwm1.zip

Figure 4.52 shows an example of operations when the automatic dead time setting function is not used (GTDTCR.TDE bit is 0).

In the above sample code, a waveform that secures the dead time period is generated by setting a value to the compare match register GTCCRB for negative-phase waveforms and buffer register GTCCRE.

Similar to the positive-phase, in the negative-phase the compare value is updated with every cycle by modifying buffer register GTCCRE ((1) in Figure 4.52) and transferring from the buffer register to compare register GTCCRB ((2) in Figure 4.52) when a GTCNT counter underflow occurs (trough).

In addition, the same dead time period is secured as shown in the operations in Figure 4.16.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.52 When Automatic Dead Time Setting Function is Not Used (Triangle-wave PWM Mode 1)

4.6.6 Usage Notes

4.6.6.1 Count Start for Multiple Channels

In this sample code, the CSTRT0, CSTRT1, and CSTRT2 bits of timer software start register GTSTR are set at the same time in the gpt0_gpt1_gpt2_start function to start GPTW0, GPTW1, and GPTW2 counting at the same time.

When using the R_Config_GPT0_Start, R_Config_GPT1_Start, and R_Config_GPT2_Start functions generated by the Smart Configurator, the counting starts timing may not be the same because each of the functions are read.

Refer to RX66T Group User's Manual: Hardware, section 24.3.8.1 Synchronous Operation by Software.

4.6.6.2 Settings of GTCCRm Register during Compare Match Operation (m = A to F)

This sample code automatically sets the dead time and uses the value of compare match register GTCCRA, the register for positive-phase waveform, to update the value of compare match register GTCCRB, the register for negative-phase waveform.

The values of compare register GTCCRA should be set to satisfy the following restrictions.

GTCCRA > GTDVU
GTCCRA > GTDVD
GTCCRA < GTPR

If the GTCCRA is set to 0000 0000h or a value greater than the setting value of the GTPR during the counting operation, the output protection function is activated.

However, if the conditions below are not satisfied, the function does not operate normally.

When the GTCCRA register value at the start of count operation is greater than 0000 0001h and less than the setting value of the GTPR register

For details on the output protection function, refer to RX66T Group User's Manual: Hardware, section 24.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 9; m=A, B).

When not using the automatic dead time setting function, set a value greater than 0000 0001h and less than the setting value of the GTPR register in the GTCCRA (GTCCRB) register. When 0000 0000h or the same value as the GTPR register is set in the GTCCRA register, a compare match occurs in a cycle only when [GTCCRA (GTCCRB) = 0000 0000h] or [GTCCRA (GTCCRB) = GTPR] is met. Furthermore, if a value exceeding the setting value of the GTPR register is set in the GTCCRA, a compare match does not occur.

For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F), (1) When Automatic Dead Time Setting has been Made in Triangle-Wave PWM Mode and (2) When Automatic Dead Time Setting has not been Made in Triangle-Wave PWM Mode.

4.7 Triangle-Wave PWM Mode 2

- Target sample code file name: r01an5995_rx66t_gptw_triangle_pwm2_dt.zip

4.7.1 Overview

The GPTW triangle-wave PWM mode 2 can be used to output 3-phase complementary PWM waveforms with dead time.

This sample code describes a sample code that uses the automatic dead time setting function in triangle-wave PWM mode 2 (32-bit transfer at crest and trough) and repeats the following output waveforms. Each duty cycle generates laterally asymmetric PWM waveforms using the buffer (not a double buffer).

- U-phase duty switching: 20% → 40% → 60% → 80% → 60% → 40% → ⋯
- V-phase duty switching: 40% → 60% → 80% → 60% → 40% → 20% → ⋯
- W-phase duty switching: 60% → 80% → 60% → 40% → 20% → 40%→ ⋯

The duty cycle is changed by transferring the value of buffer register GTCCRC to compare register GTCCRA when a GTCNT counter overflows and when a GTCNT counter underflow occurs.

The following list provides the GPTW settings used in the sample code.

- Use triangle-wave PWM mode 2
- Use channels 0, 1 and 2 (channel numbers: n = 0, 1, and 2)
- Carrier period = 1ms
- Timer count clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Counter up-counts from initial value 0
- Use GTCCRA as duty output compare match
  - Use GTIOCnA pin as PWM output pin
  - Use GTCCRA as compare match
  - High output at counting starts, high output at counting stops
  - Toggle output at GTCCRA compare match
  - Retain output at cycle end
- Use GTCCRB as duty output compare match
  - Use GTIOCnB pin as PWM output pin
  - Use GTCCRB as compare match
  - Low output at counting starts, low output at counting stops
  - Toggle output at GTCCRB compare match
  - Retain output at cycle end
- Use buffer register
  - GTCCRA operates as single buffer
  - Use GTCCRC as GTCCRA buffer register
- Use automatic dead time generation
- Software source count start enabled
- Duty changes at each cycle
  - Duty changes at the GTCNT counter overflow interrupt and GTCNT counter underflow interrupt
  - Refer to Figure 4.55 for details on duty change timing

Set in Smart Configurator.
For Setting Methods,
refer to section 4.7.3.

Triangle-wave PWM mode 2 output for this sample code is shown below.

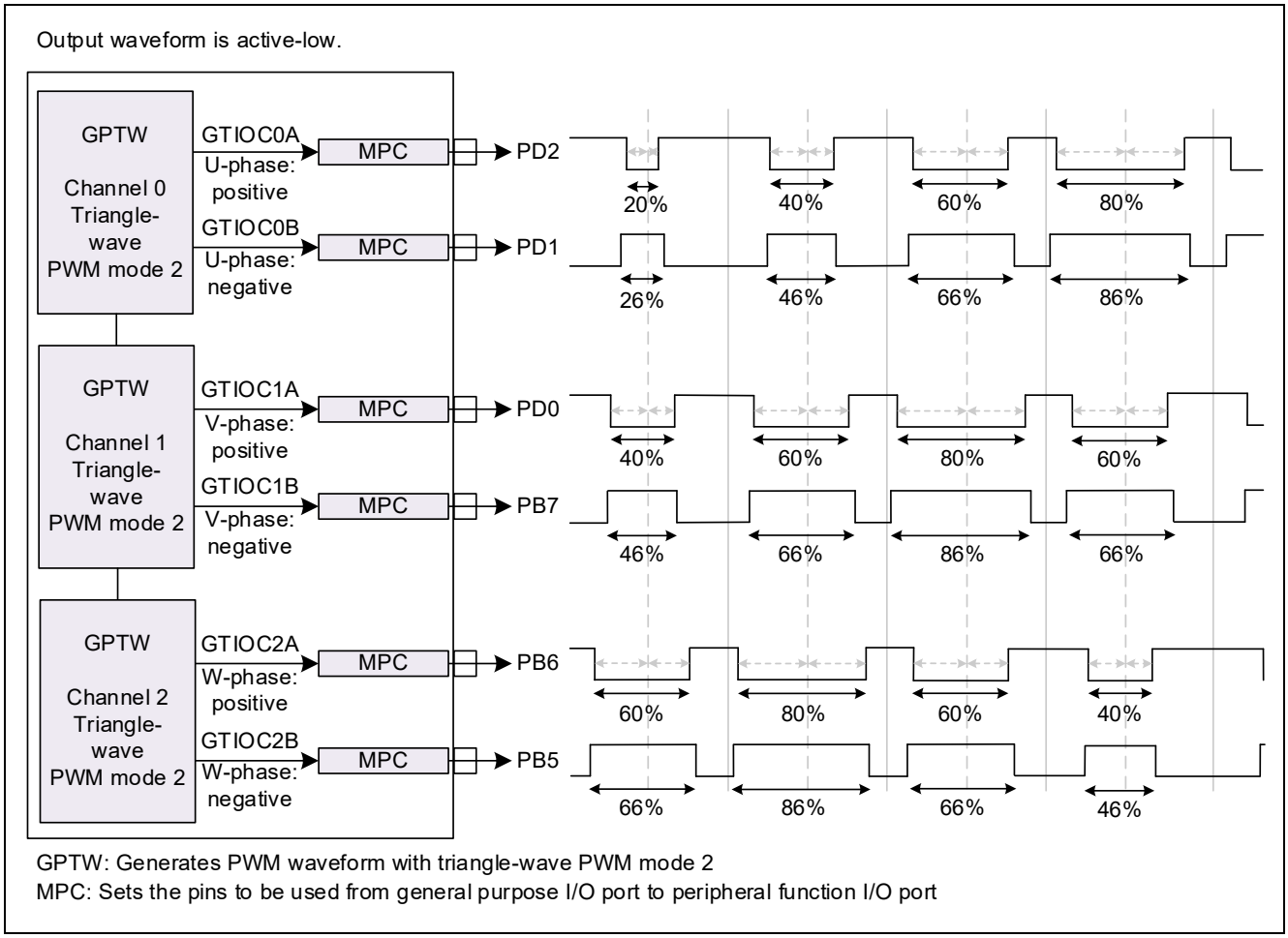


Figure 4.53 Triangle-Wave PWM Mode 2 Output

4.7.2 Operation Details

The sample code operations are shown in Figure 4.55. The settings of the duty cycle are changed with each period by modifying the value of buffer register GTCCRC when a GTCNT counter overflow interrupt (GTCIV0) and a GTCNT counter underflow interrupt (GTCIU0) are generated ((1) in Figure 4.55).

This sample code uses triangle-wave PWM mode 2 to update data by transferring buffer register GTCCRC to compare register GTCCRA when a GTCNT counter overflow (crest) and a GTCNT counter underflow (trough) occur ((2) in Figure 4.55).

In addition, the GTCCRB register is automatically set according to the GTCCRA update because the automatic dead time function is used. The same values are set for the GTDVU and GTDVD. This sample code automatically sets the GTCCRB value at counting starts ((3) in Figure 4.55).

After counting starts, a compare match occurs between the compare register and the counter register, negative-phase output turns OFF, and then positive-phase output turns ON ((4) in Figure 4.55).

- Laterally Asymmetric PWM Waveform Output
  The duty cycle in each period generates a different duty cycle for the up-counting period and down-counting periods.

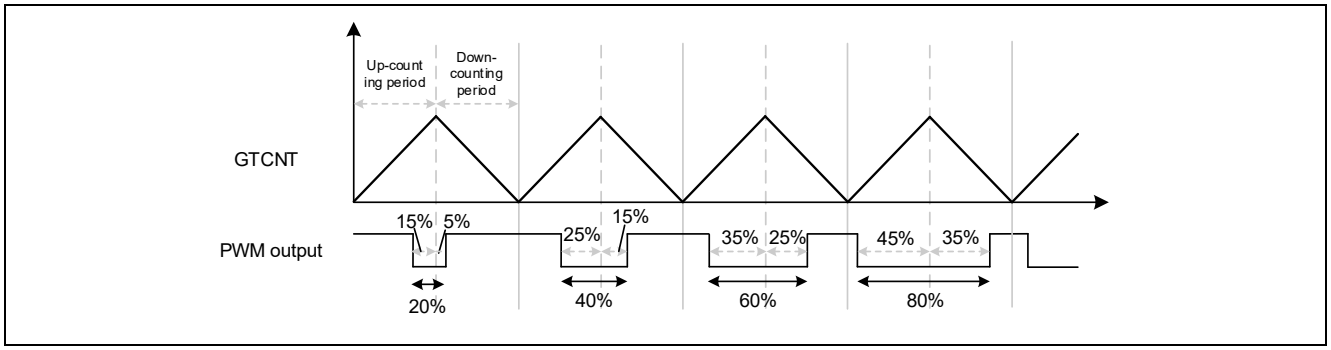
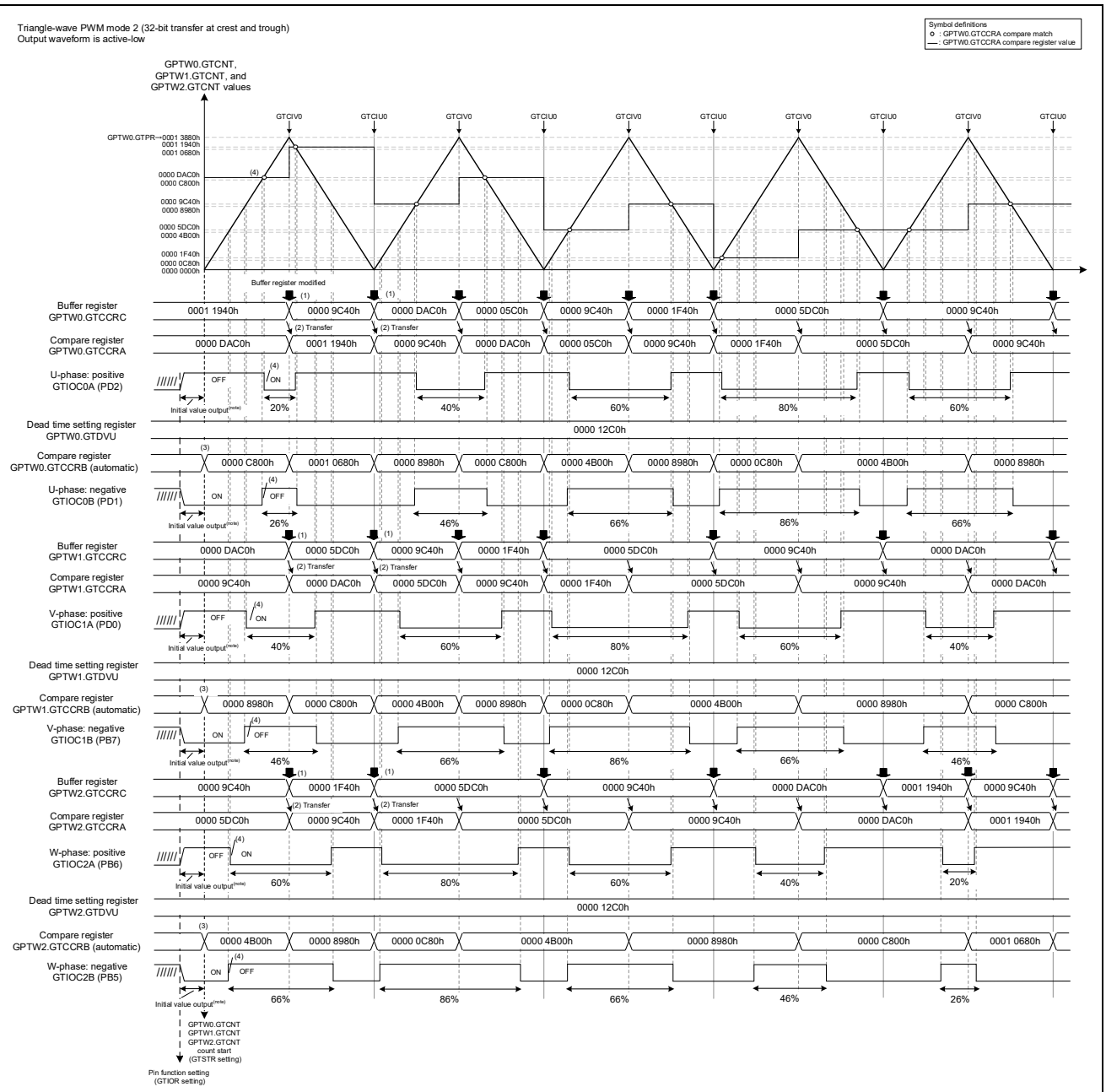


Figure 4.54 Laterally Asymmetric PWM Output Waveform



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.55 Sample Code Operations

4.7.3 Smart Configurator Settings

The sample codes use the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.10 Adding Components

Item	Description		
Component	General PWM Timer		
Configuration Name	Config_GPT0	Config_GPT1	Config_GPT2
Work mode	Triangle-wave PWM Mode 2		
Resource	GPT0	GPT1	GPT2

Figure 4.56 to Figure 4.58 show the Config_GPT0 settings. The same settings apply for GPT1 and GPT2.

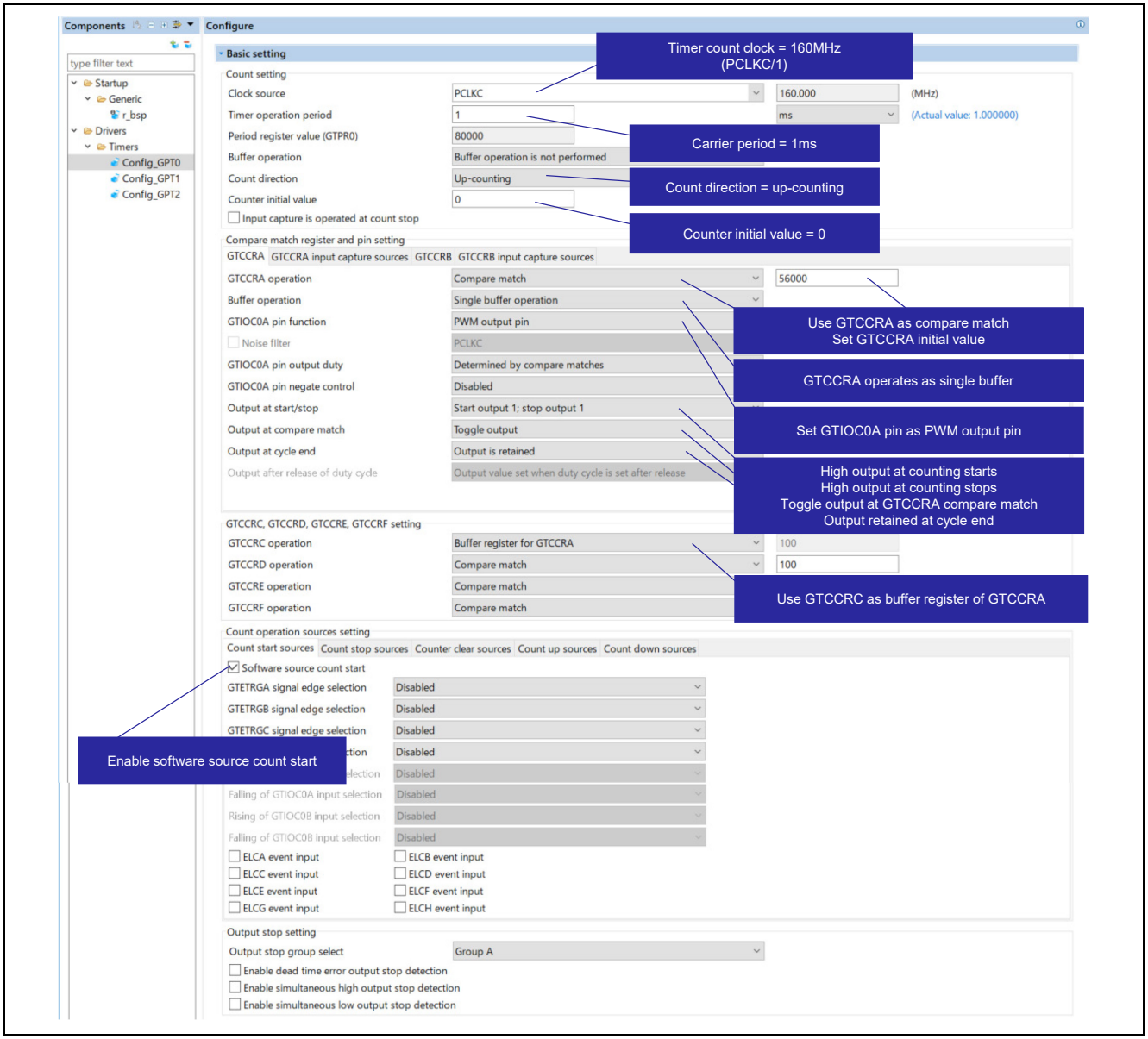


Figure 4.56 GPT0 Settings (1/2)

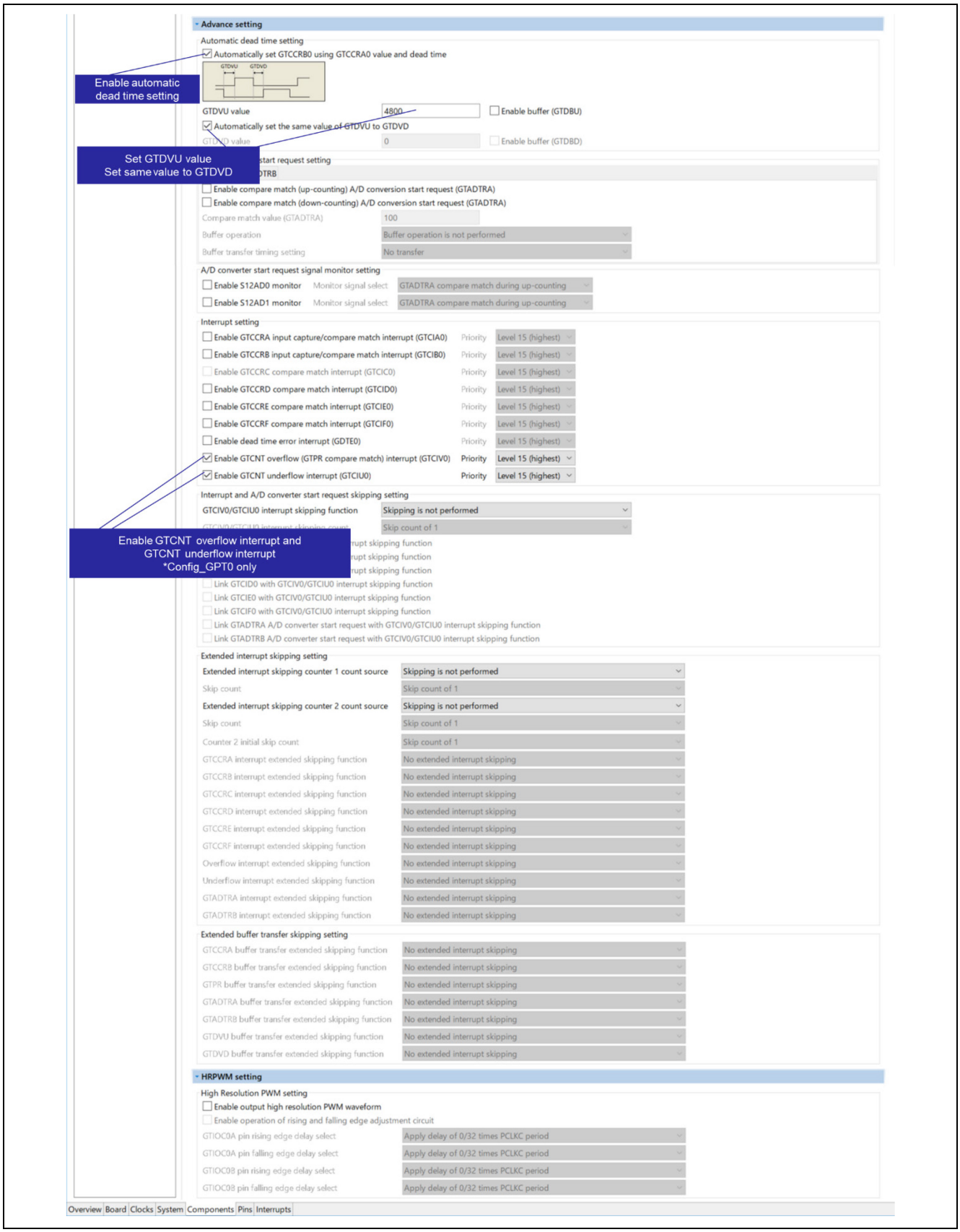


Figure 4.57 GPT0 Settings (2/2)

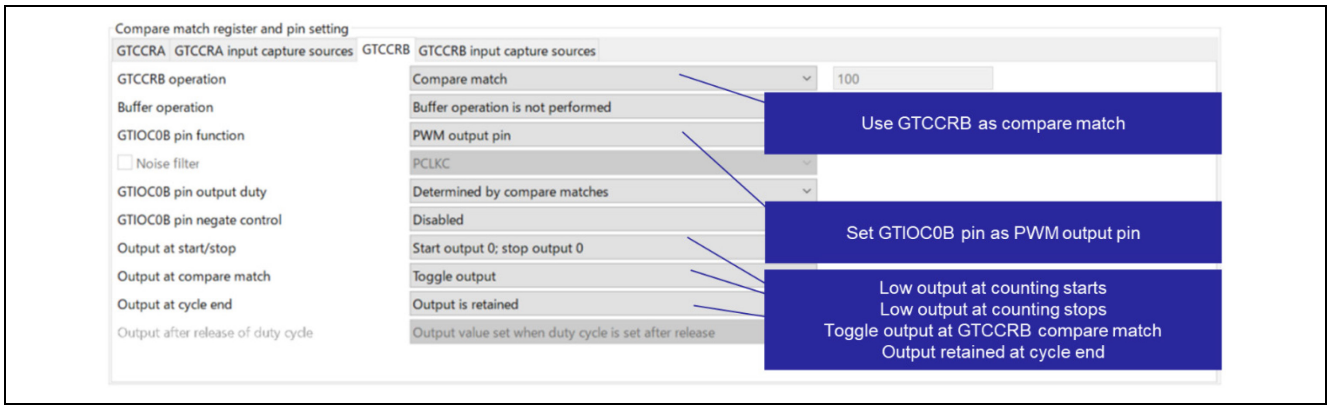


Figure 4.58 GPT0 Settings (Compare Match Register and Pins Setting of GTCCRB)

4.7.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

In the main function, count start function gpt0_gpt1_gpt2_start is read and counting is started.

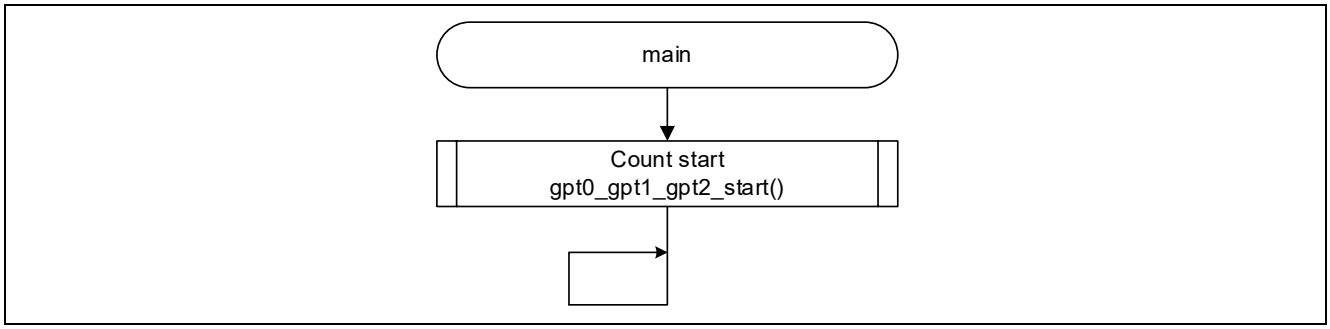


Figure 4.59 main Function

In the count start function, the GPT0, GPT1, and GPT2 counting is started after the GTCIV0 and GTCIU0 interrupts are enabled.

This function is newly created after code generation by the Smart Configurator.

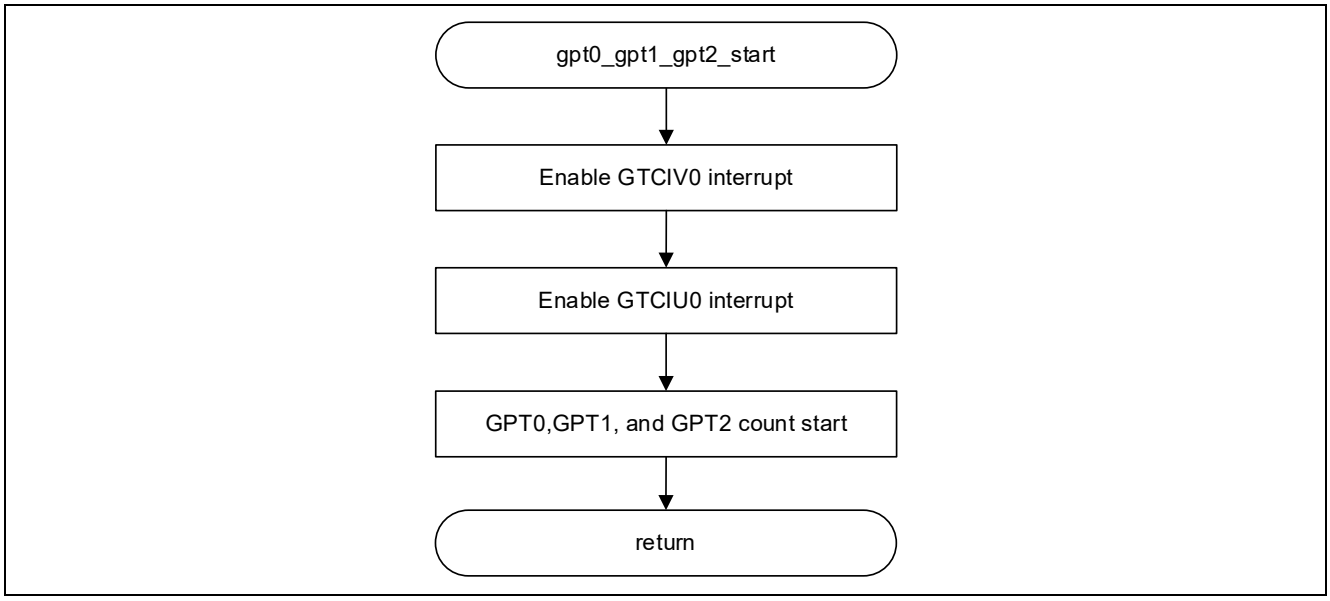


Figure 4.60 Count Start Function

The user initialization function R_Config_GPT0_Create_UserInit, which is executed before the main function, sets the initial value of the buffer register. This function is called from within the R_Config_GPT0_Create function.

R_Config_GPT1_Create_UserInit and R_Config_GPT2_Create_UserInit also perform the same processes.

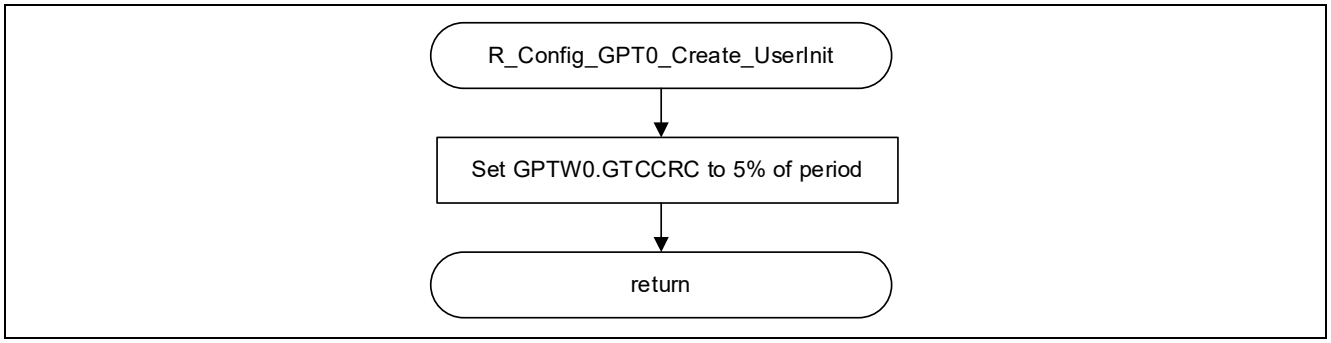


Figure 4.61 User Initialization Function

The GTCIV0 interrupt handler function changes the values of the buffer registers according to the changes to the output duty cycles.

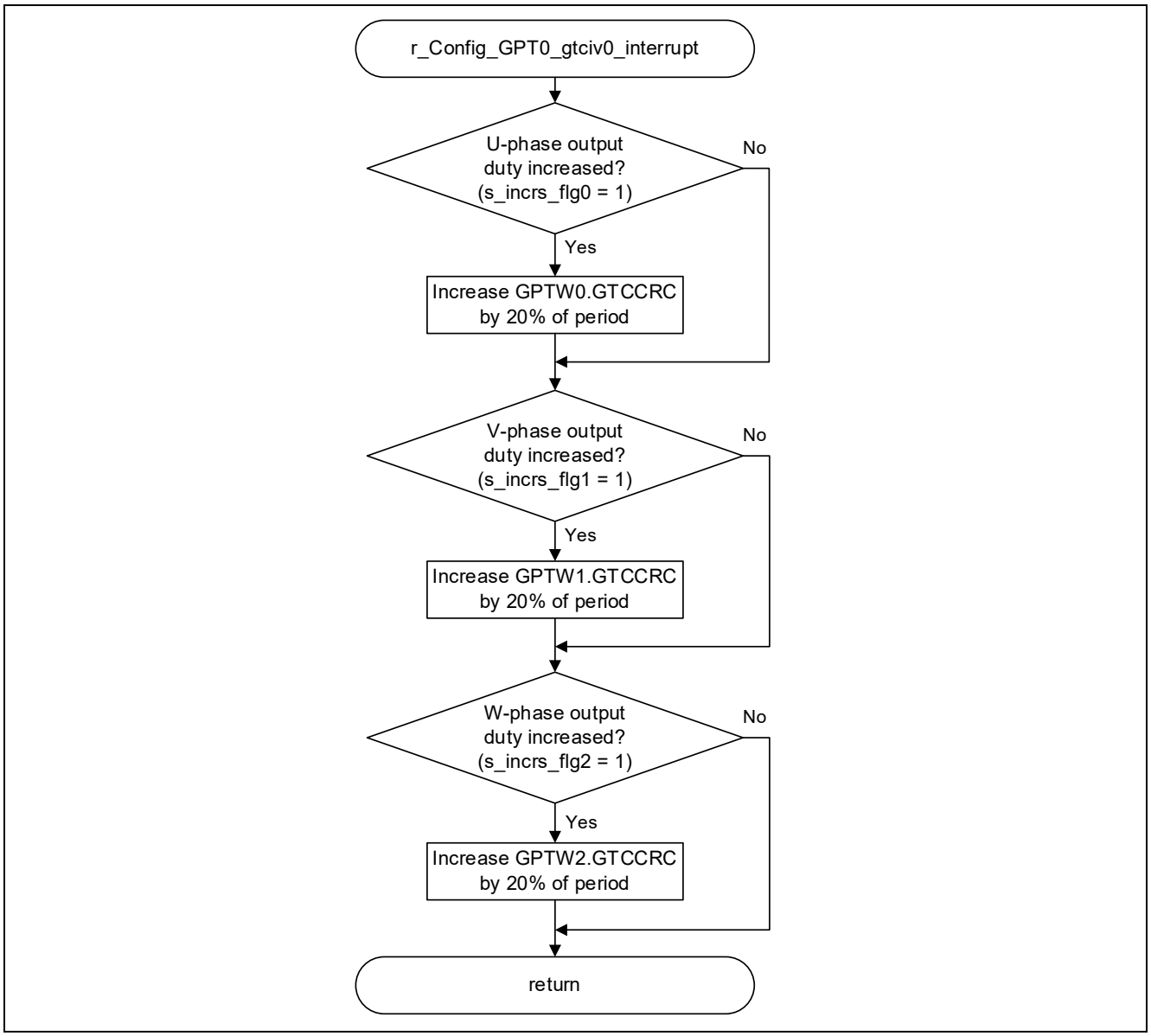


Figure 4.62 GTCIV0 Interrupt Handler Function

The GTCIU0 interrupt handler function changes the values of the buffer registers according to the current values of buffer registers.

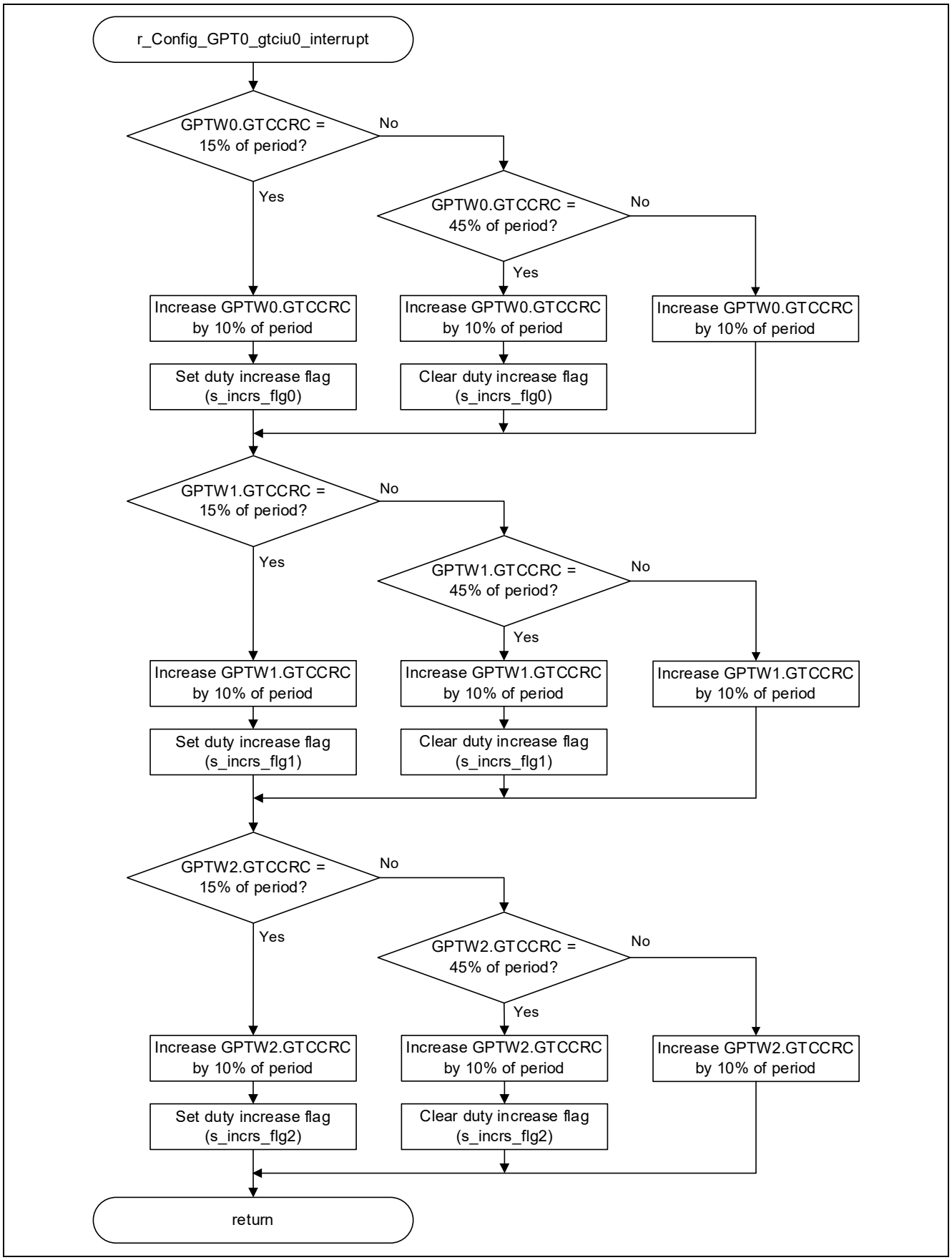


Figure 4.63 GTCIU0 Interrupt Handler Function

4.7.5 Related Operations

4.7.5.1 Separate Automatic Dead Time Settings for Each Interval

This sample code uses the automatic dead time setting function and a dead time is generated with a common switching point in the first half and second half of the negative-phase by setting the GTDTCR.TDFER bit to 1.

In the automatic dead time setting function, the dead time can be separately set for the first half and second half of a waveform. Dead time for the changing point in the first half of a negative-phase is set in the GTDVU register and that in the second half is set in the GTDVD register.

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.4 Automatic Dead Time Setting Function.

4.7.5.2 Buffer Operations with Automatic Dead Time Setting Function

This sample code uses the automatic dead time setting function to generate a negative-phase waveform that secures the fixed dead time period set by the Smart Configurator.

The automatic dead time setting function uses the GTDBU as buffer register GTDVU and the GTDBD as buffer register GTDVD to update the dead time period during the count by performing buffer transfer at the end of the count cycle (when a GTCNT counter underflow occurs (trough)).

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.4 Automatic Dead Time Setting Function.

Smart Configurator settings are as shown below.

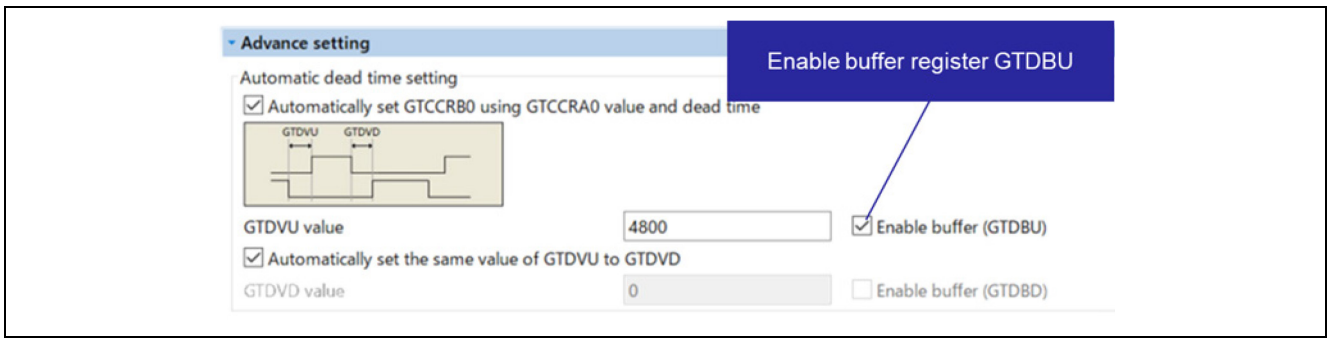


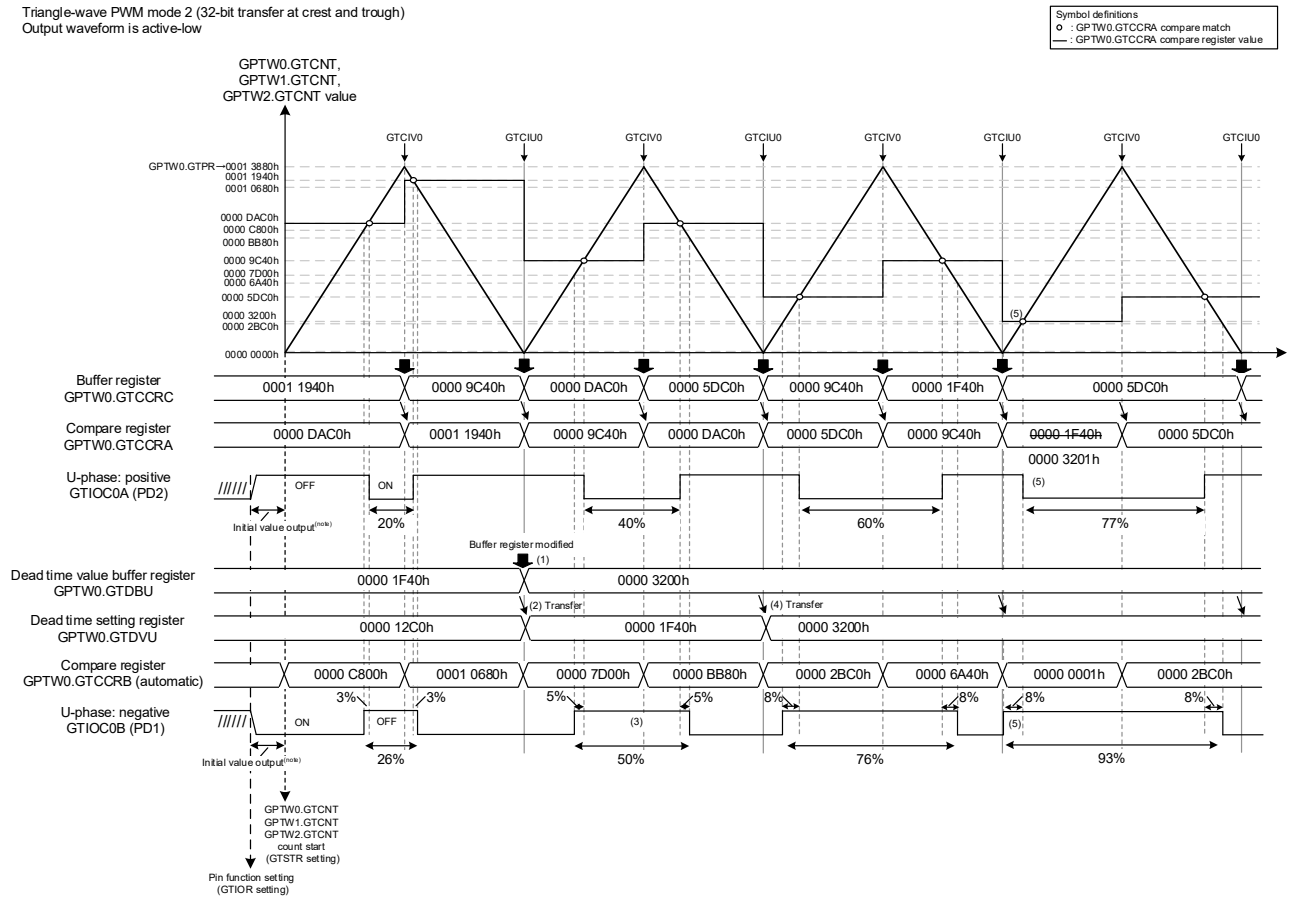
Figure 4.64 Smart Configurator Automatic Dead Time Setting

Figure 4.65 shows an example of operations of the automatic dead time setting function using the buffer register of dead time value.

The value of the GTDBU is modified by the GTCNT counter underflow interrupt (GTCIU0) ((1)) in Figure 4.65). The value of the GTDBU is transferred to the dead time setting register GTDVU when a GTCNT counter underflow occurs (trough) ((2)) in Figure 4.65), and the waveform that secures the dead time period after update is output ((3) in Figure 4.65).

In the 3rd cycle, the GTDBU value that was modified at the end of the 1st cycle is transferred to the GTDVU ((4) in Figure 4.65).

The dead time setting of the 4th cycle results in a dead time error (GTCCRA-GTDVU<0) at the changing point in the first half of the negative-phase, so waveforms of the positive-phase and negative-phase with corrected changing points are output ((5) in Figure 4.65).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.65 Buffer Operations with Automatic Dead Time Setting Function

4.7.5.3 When Automatic Dead Time Setting Function is Not Used

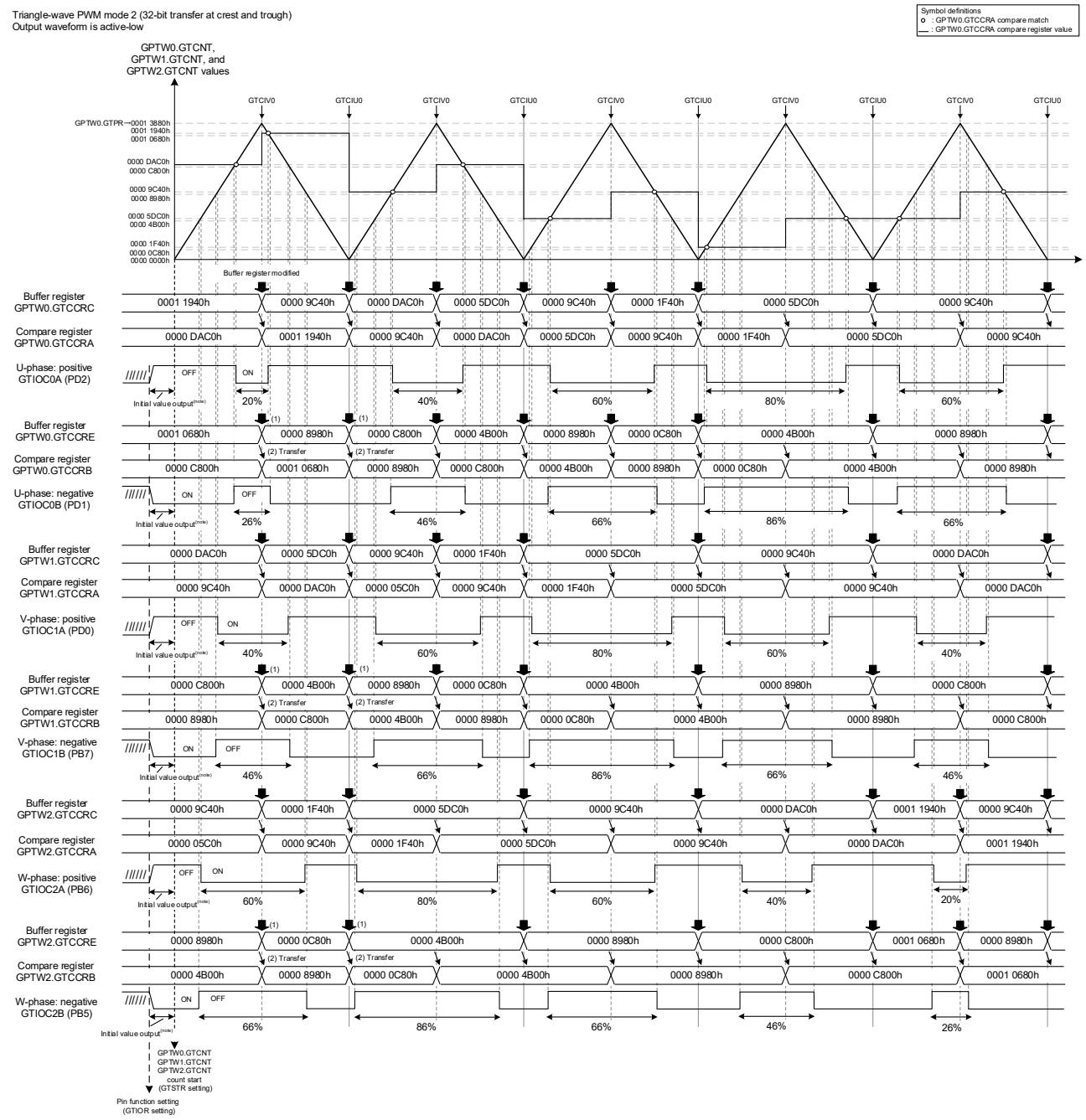
- Target sample code file name: r01an5995_rx66t_gptw_triangle_pwm2.zip

Figure 4.66 shows an example of operations when the automatic dead time setting function is not used (GTDTCR.TDE bit is 0).

When the automatic dead time setting function is not used, a waveform that secures the dead time period is generated by setting a value to compare match register GTCCRB, the register for negative-phase waveform, and buffer register GTCCRE.

Similar to the positive-phase, in the negative-phase the compare value is updated with every cycle by modifying buffer register GTCCRE when the GTCNT counter overflow interrupt (GTCIV0) and GTCNT counter underflow interrupt (GTCIU0) are generated ((1) in Figure 4.66) and transferring buffer register GTCCRE to the compare register GTCCRB when the GTCNT counter overflow (crest) occurs ((2) in Figure 4.66).

In addition, the same dead time period is secured as shown in the operations in Figure 4.55.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.66 When Not Using Automatic Dead Time Setting Function (Triangle-wave PWM Mode 2)

4.7.6 Usage Notes

4.7.6.1 Count Start for Multiple Channels

In this sample code, the CSTRT0, CSTRT1, and CSTRT2 bits of timer software start register GTSTR are set at the same time in the gpt0_gpt1_gpt2_start function to start GPTW0, GPTW1, and GPTW2 counting at the same time.

When using the R_Config_GPT0_Start, R_Config_GPT1_Start, and R_Config_GPT2_Start functions generated by the Smart Configurator, the counting starts timing may not be the same because each of the functions are read.

Refer to RX66T Group User's Manual: Hardware, section 24.3.8.1 Synchronous Operation by Software.

4.7.6.2 Settings of GTCCRm Register during Compare Match Operation (m = A to F)

This sample code automatically sets the dead time and uses the value of compare match register GTCCRA, the register for positive-phase waveform, to update the value of compare match register GTCCRB, the register for negative-phase waveform.

The values of compare register GTCCRA should be set to satisfy the following restrictions

GTCCRA > GTDVU
GTCCRA > GTDVD
GTCCRA < GTPR

If the GTCCRA is set to 0000 0000h or a value greater than the setting value of the GTPR during the counting operation, the output protection function is activated.

However, if the conditions below are not satisfied, the function does not operate normally.

When the GTCCRA register value at the start of count operation is greater than 0000 0001h and less than the setting value of the GTPR register

For details on the output protection function, refer to RX66T Group User's Manual: Hardware, section 24.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 9; m=A, B).

When not using the automatic dead time setting function, set a value greater than 0000 0001h and less than the setting value of the GTPR register in the GTCCRA (GTCCRB) register. When 0000 0000h or the same value as the GTPR register is set in the GTCCRA register, compare match occurs in a cycle only when [GTCCRA (GTCCRB) = 0000 0000h] or [GTCCRA (GTCCRB) = GTPR] is met. Furthermore, if a value exceeding the setting value of the GTPR register is set in the GTCCRA, and compare match does not occur.

For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F), (1) When Automatic Dead Time Setting has been Made in Triangle-Wave PWM Mode and (2) When Automatic Dead Time Setting has not been Made in Triangle-Wave PWM Mode.

4.8 Triangle-Wave PWM Mode 3

- Target sample code file name: r01an5995_rx66t_gptw_triangle_pwm3_dt.zip

4.8.1 Overview

The GPTW triangle-wave PWM mode 3 can be used to output 3-phase complementary PWM waveforms with dead time.

This sample code describes a sample code that uses the automatic dead time setting function in triangle-wave PWM mode 3 (64-bit transfer at trough) and repeats the following waveform output. Each duty cycle generates laterally asymmetric PWM waveforms using the double buffer.

- U-phase duty switching: 20% → 40% → 60% → 80% → 60% → 40% → ⋯
- V-phase duty switching: 40% → 60% → 80% → 60% → 40% → 20% → ⋯
- W-phase duty switching: 60% → 80% → 60% → 40% → 20% → 40%→ ⋯

The duty cycle is changed by transferring the value of the temporary register A to the compare register GTCCRA when a GTCNT counter overflow occurs, and transferring the value of buffer register GTCCRD to temporary register A and the value of buffer register GTCCRC to compare register GTCCRA when an underflow occurs.

The following list provides the GPTW settings used in the sample code.

- Use triangle-wave PWM mode 3
- Use channels 0, 1 and 2 (channel numbers: n = 0, 1, and 2)
- Carrier period = 1ms
- Timer count clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Counter up-counts from initial value 0
- Use GTCCRA as duty output compare match
  - Use GTIOCnA pin as PWM output pin
  - Use GTCCRA as compare match
  - High output at counting starts, high output at counting stops
  - Toggle output at GTCCRA compare match
  - Retain output at cycle end
- Use GTCCRB as duty output compare match
  - Use GTIOCnB pin as PWM output pin
  - Use GTCCRB as compare match
  - Low output at counting starts, low output at counting stops
  - Toggle output at GTCCRB compare match
  - Retain output at cycle end
- Use buffer register
  - GTCCRA and GTCCRB operate as double buffers
  - Use GTCCRC and GTCCRD as buffer registers of GTCCRA
  - Use GTCCRE and GTCCRF as buffer registers of GTCCRB
- Use automatic dead time generation
- Software source count start enabled
- Duty changes at each cycle
  - Duty changes at the GTCNT counter underflow interrupt
  - Refer to Figure 4.69 for details on duty change timing

Set in Smart Configurator.
For Setting Methods,
refer to section 4.8.3.

Triangle-wave PWM mode 3 output for this sample code is shown below.

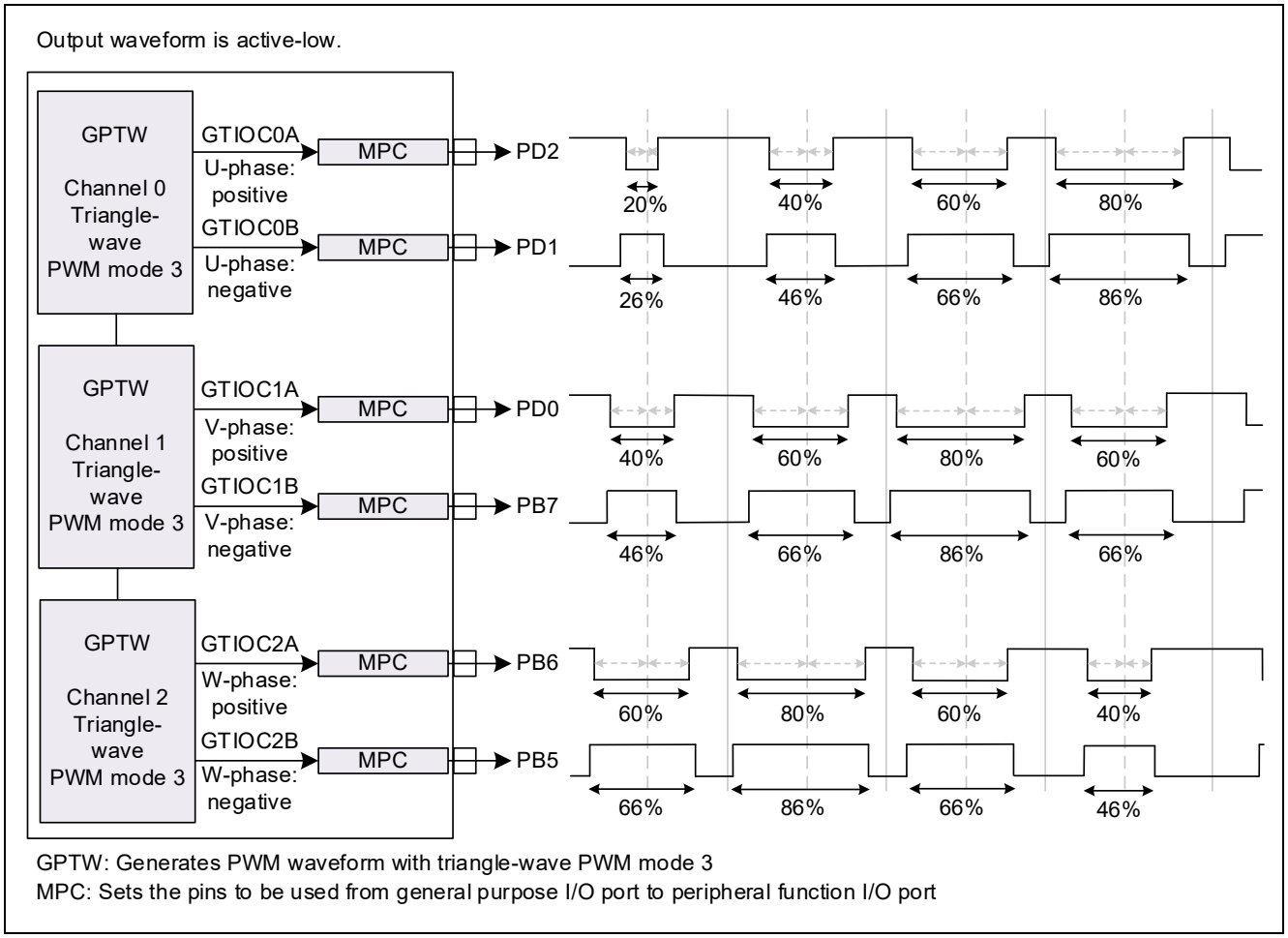


Figure 4.67 Triangle-Wave PWM Mode 3 Output

4.8.2 Operation Details

The sample code operations are shown in Figure 4.69. The settings of the duty cycle are changed with each period by modifying the value of buffer registers GTCCRC and GTCCRD at the GTCNT counter underflow interrupt (GTCIU0) ((1) in Figure 4.69).

The initial values of the buffer registers for the code generated using the Smart Configurator are set to the same value as the compare register. As a result, the buffer register values are set in the user initialization function R_Config_GPT0_Create_UserInit before the counting starts. The values set in the registers are transferred from buffer register GTCCRD to temporary register A and from buffer register GTCCRC to compare register GTCCRA, respectively, by forced buffer transfers ((2) in Figure 4.69).

This sample code uses triangle-wave PWM mode 3 to update data by transferring temporary register A to compare register GTCCRA when a GTCNT counter overflow (crest) occurs ((3) in Figure 4.69) and transferring buffer register GTCCRD to temporary register A, and buffer register GTCCRC to compare register GTCCRA when a GTCNT counter underflow (trough) occurs ((4) in Figure 4.69).

In addition, the GTCCRB register value is automatically set according to the GTCCRA update because the automatic dead time function is used. The same values are set for the GTDVU and GTDVD. This sample code automatically sets the GTCCRB value at counting starts ((5) in Figure 4.55).

After counting starts, a compare match occurs between the compare register and the counter register, negative-phase output turns OFF, and then positive-phase output turns ON ((6) in Figure 4.69).

- Laterally Asymmetric PWM Waveform Output
  The duty cycle in each period generates a different duty cycle is for the up-counting and down-counting periods.

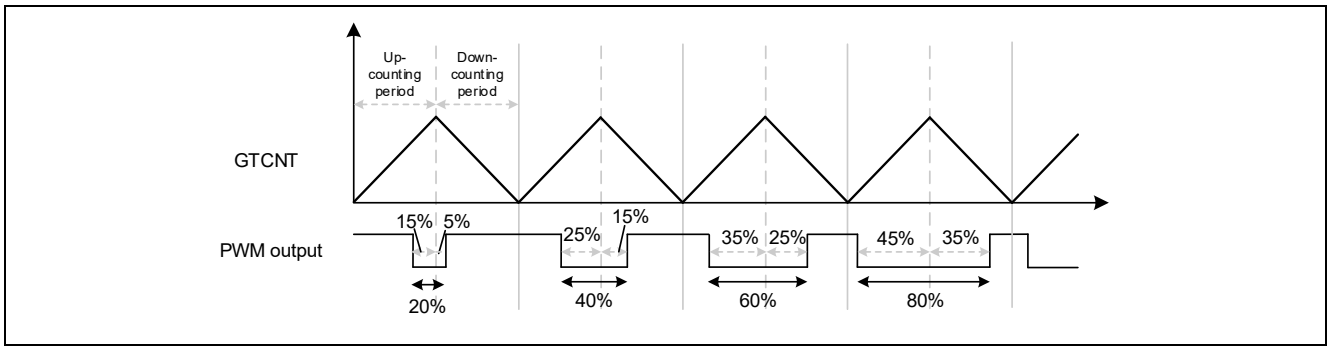
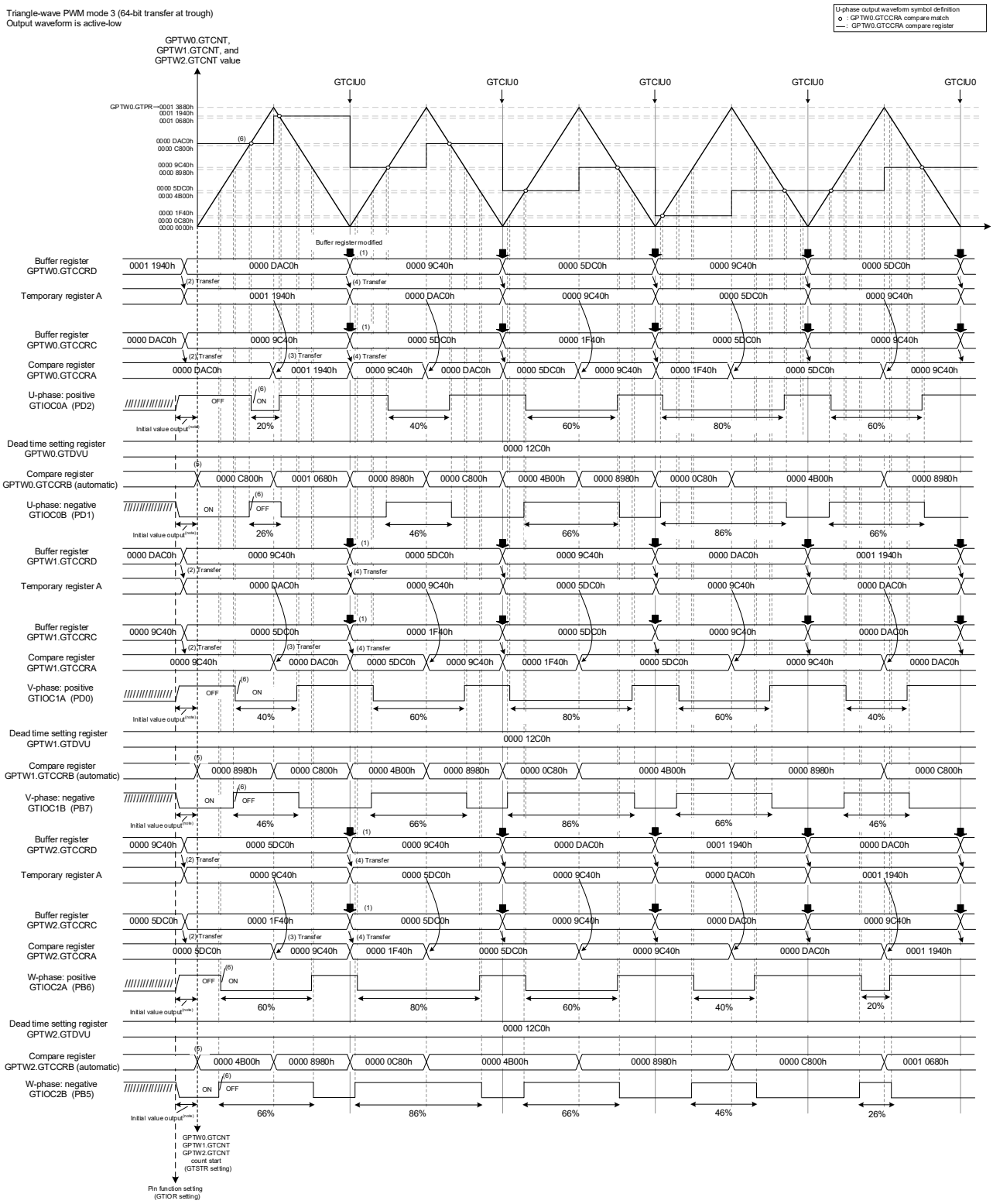


Figure 4.68 Laterally Asymmetric PWM Output Waveform



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.69 Sample Code Operations

4.8.3 Smart Configurator Settings

The sample codes use the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.11 Adding Components

Item	Description		
Component	General PWM Timer		
Configuration Name	Config_GPT0	Config_GPT1	Config_GPT2
Work mode	Triangle-wave PWM Mode 3		
Resource	GPT0	GPT1	GPT2

Figure 4.70 to Figure 4.72 show the Config_GPT0 settings. The same settings apply for GPT1 and GPT2.

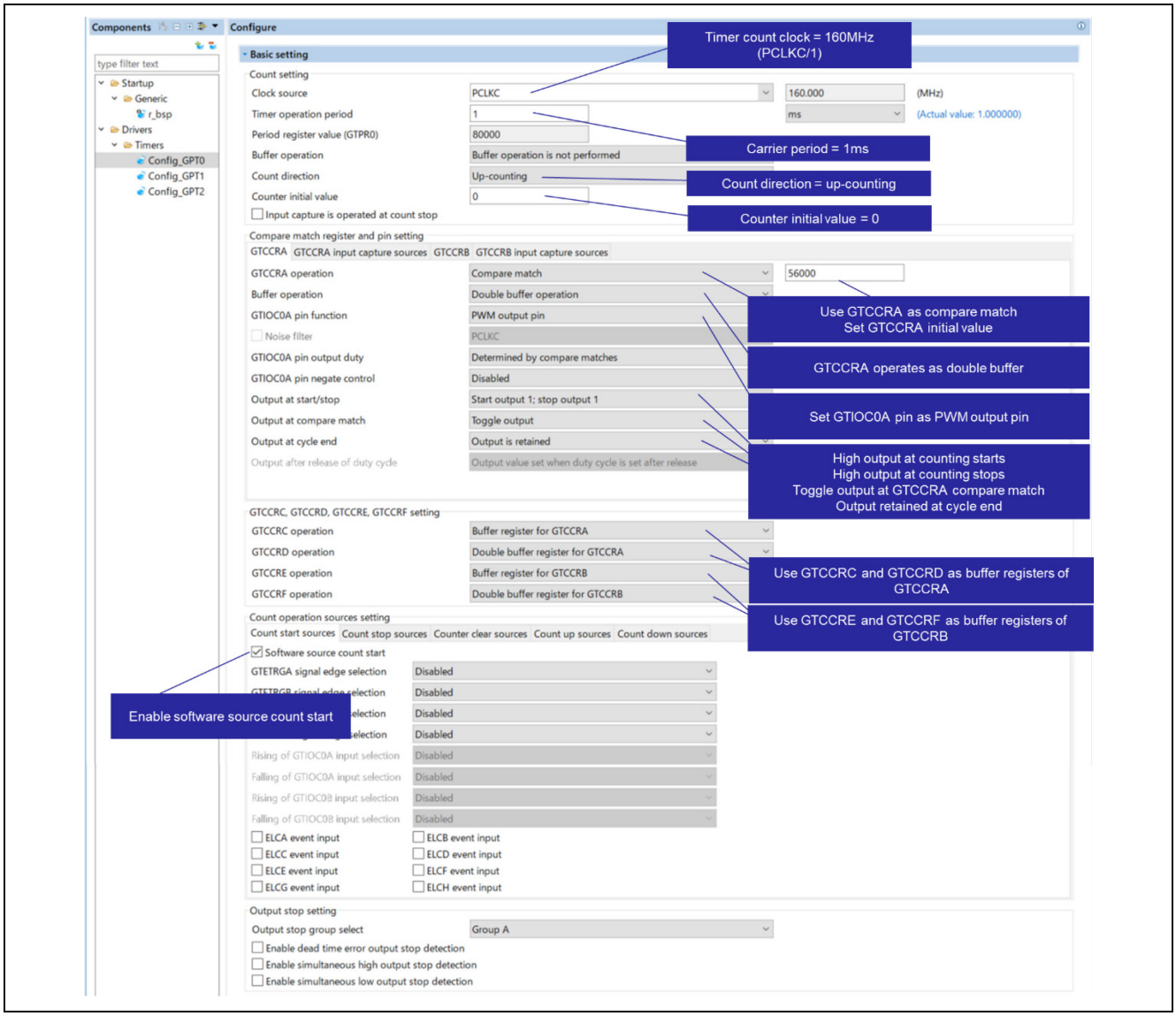


Figure 4.70 GPT0 Settings (1/2)

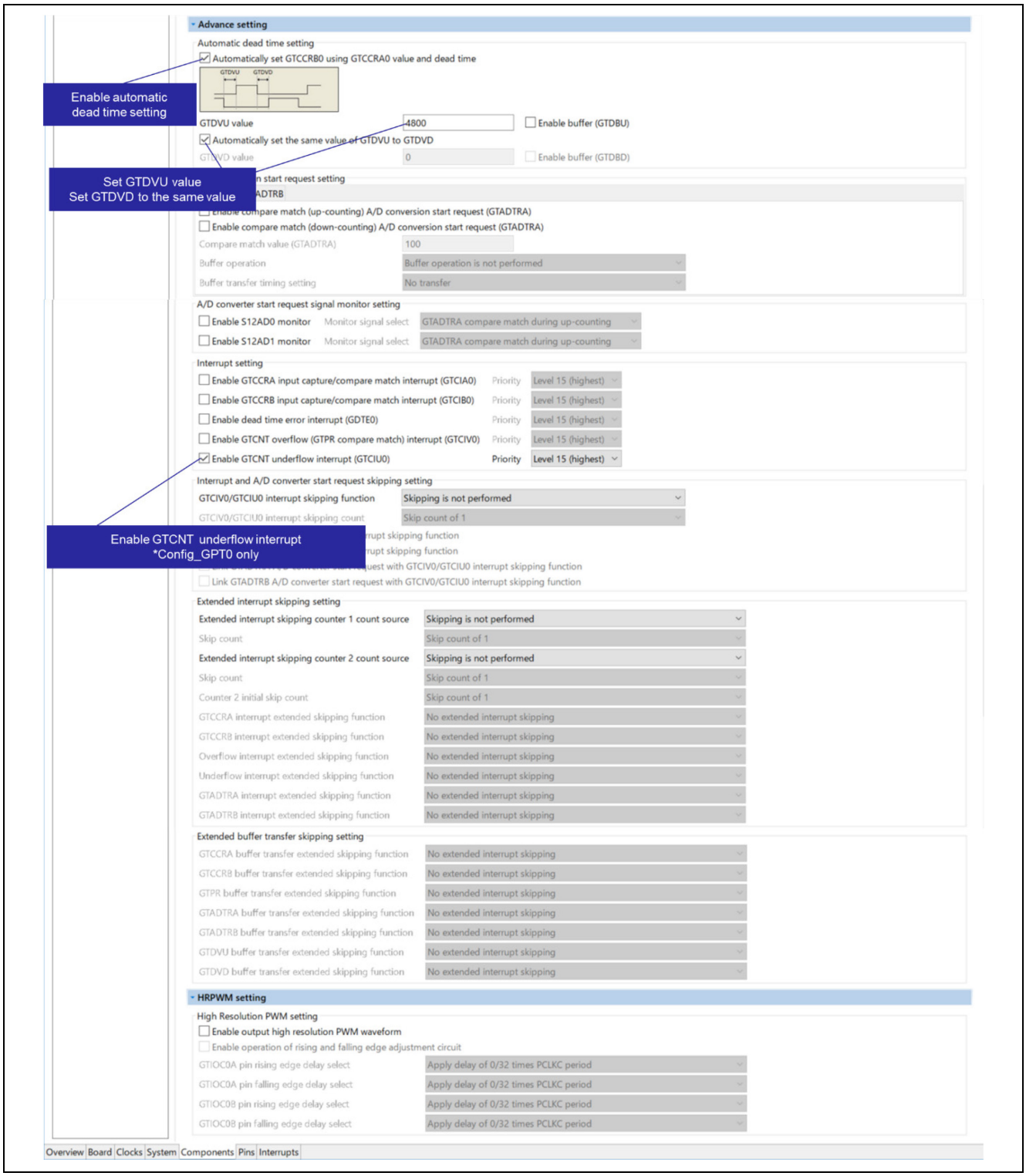


Figure 4.71 GPT0 Settings (2)

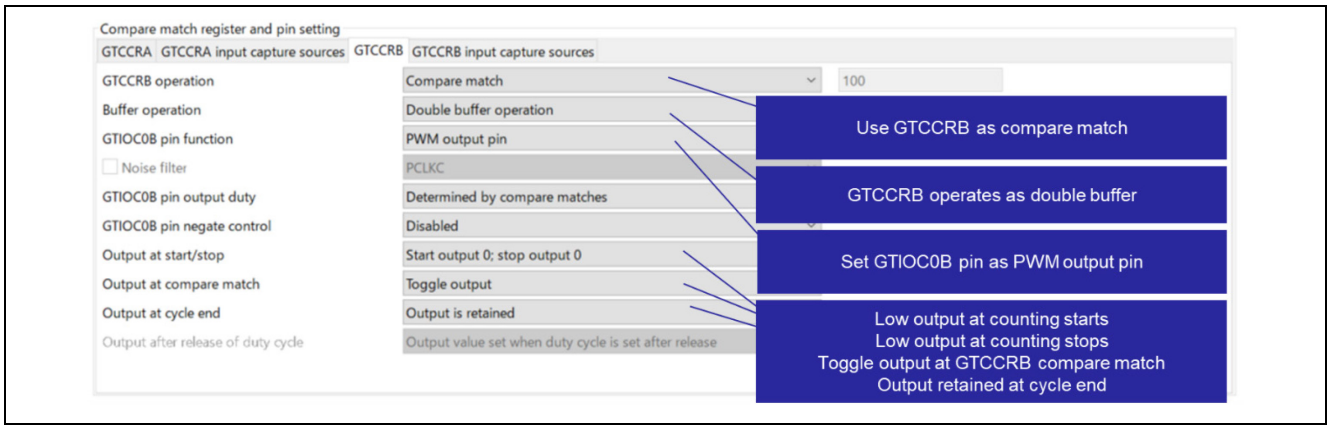


Figure 4.72 GPT0 Settings (Compare Match Register and Pins Setting of GTCCRB)

4.8.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

In the main function, count start function gpt0_gpt1_gpt2_start is read and counting is started.

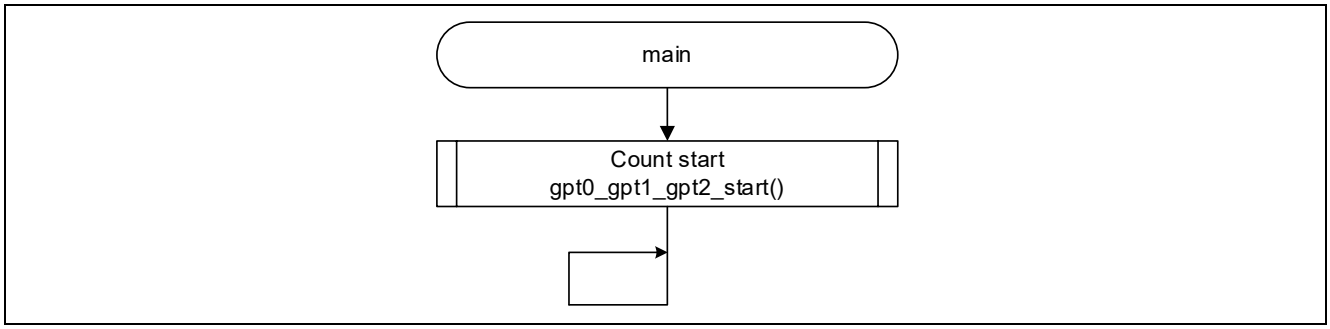


Figure 4.73 main Function

In the count start function, the GPT0, GPT1, and GPT2 counting is started after the GTCIU0 interrupt is enabled.

This function is newly created after code generation by the Smart Configurator.

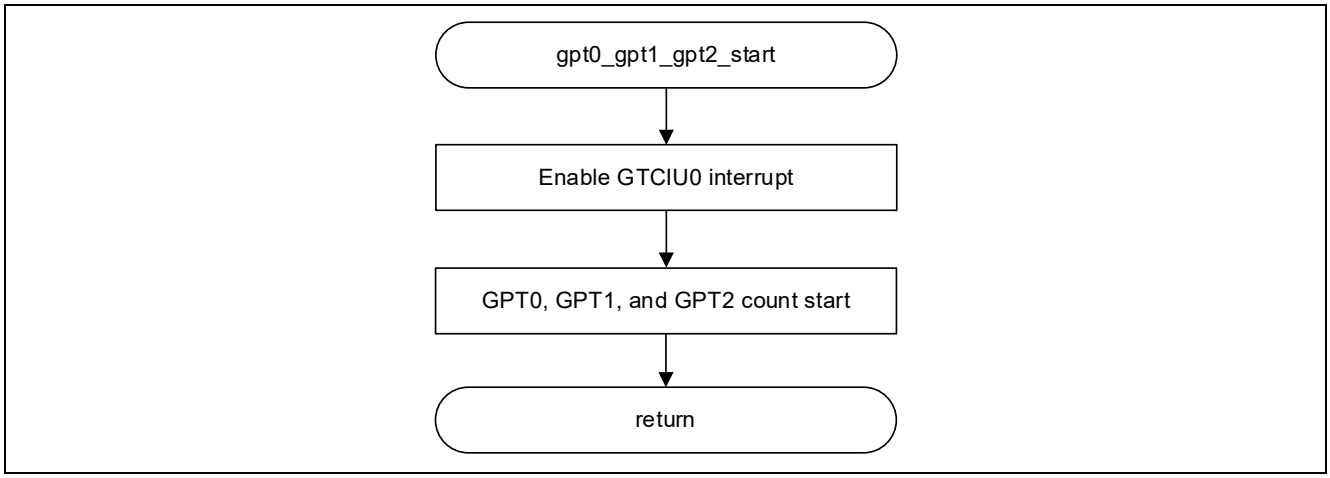


Figure 4.74 Count Start Function

The user initialization function R_Config_GPT0_Create_UserInit, which is executed before the main function, sets the initial values of the buffer registers and initializes the variables. In order to set the second compare match register value in the 1st cycle, a forced buffer transfer is performed after setting the buffer register value, and then the temporary register and compare register values are set. This function is called from within the R_Config_GPT0_Create function.

This function initializes the following variable used in this sample code.

- g_ucduty_prv0: variable for retaining the previous GPTW0.GTCCRC register value

R_Config_GPT1_Create_UserInit and R_Config_GPT2_Create_UserInit also perform the same processes.

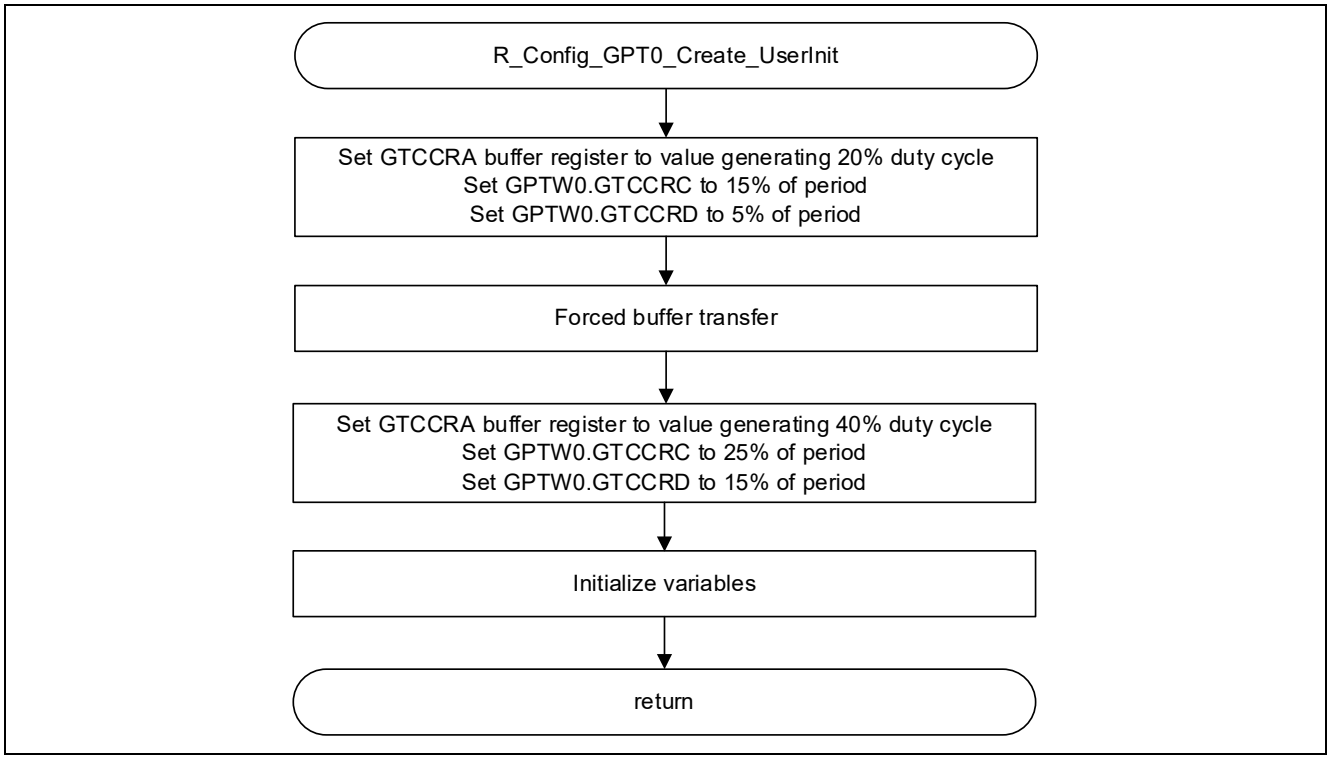


Figure 4.75 User Initialization Function

The GTCIV0 interrupt handler function changes the values of the buffer registers according to the current value of buffer register GPTW0.GTCCRC and the value set in the previous buffer registers.

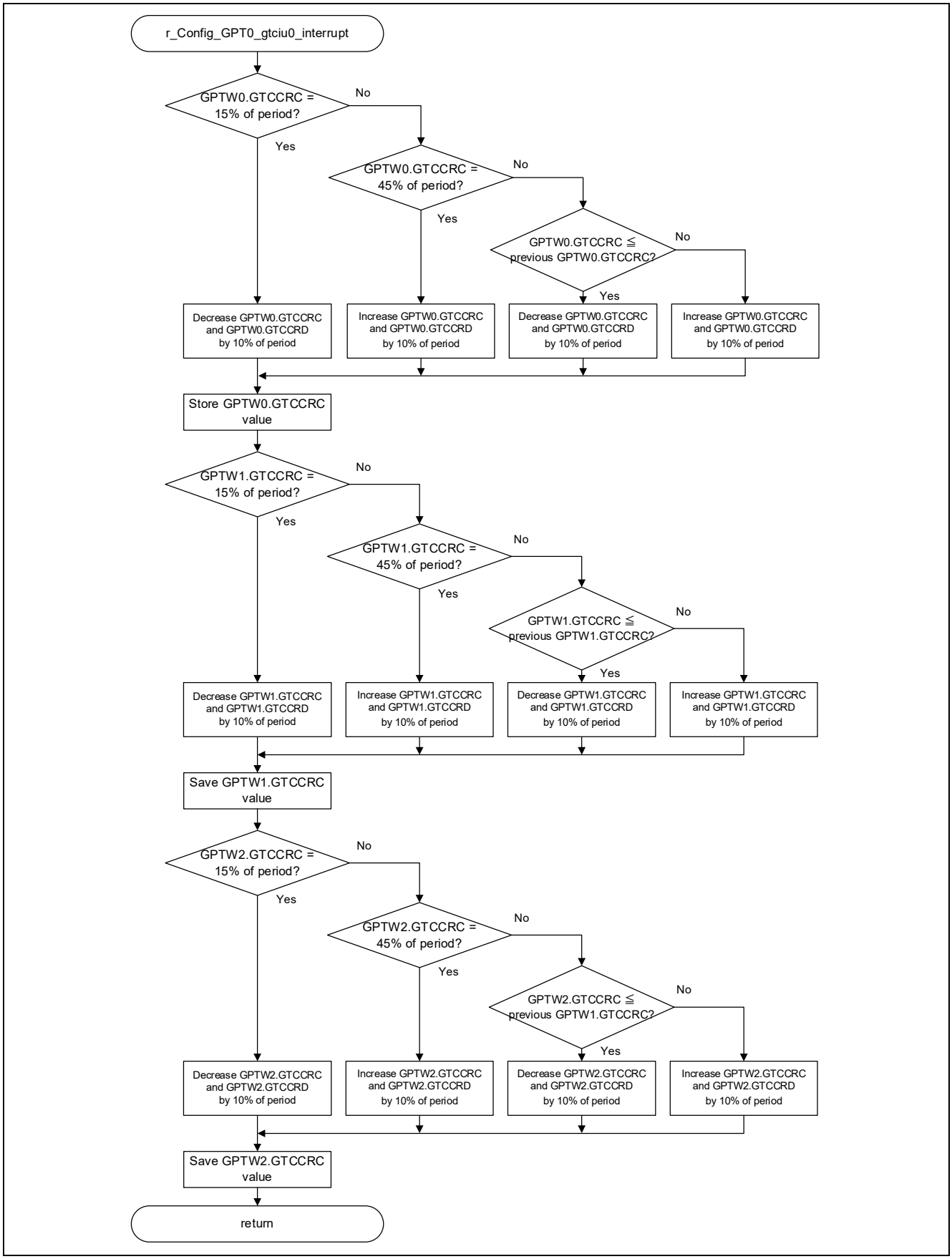


Figure 4.76 GTCIU0 Interrupt Handler Function

4.8.5 Related Operations

4.8.5.1 Separate Automatic Dead Time Settings for Each Interval

This sample code uses the automatic dead time setting function and a dead time is generated with a common switching point in the first half and second half of the negative-phase by setting the GTDTCR.TDFER bit to 1.

In the automatic dead time setting function, the dead time can be separately set for the first half and second half of a waveform. Dead time for the changing point in the first half of a negative-phase is set in the GTDVU register and that in the second half is set in the GTDVD register.

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.4 Automatic Dead Time Setting Function.

4.8.5.2 Buffer Operations with Automatic Dead Time Setting Function

This sample code uses the automatic dead time setting function to generate a negative-phase waveform that secures the fixed dead time period set by the Smart Configurator.

The automatic dead time setting function uses the GTDBU as buffer register GTDVU and the GTDBD as buffer register GTDVD to update the dead time period during the count by performing buffer transfer at the end of the count cycle (when a GTCNT counter underflow occurs (trough).

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.4 Automatic Dead Time Setting Function.

Smart Configurator settings are as shown below.

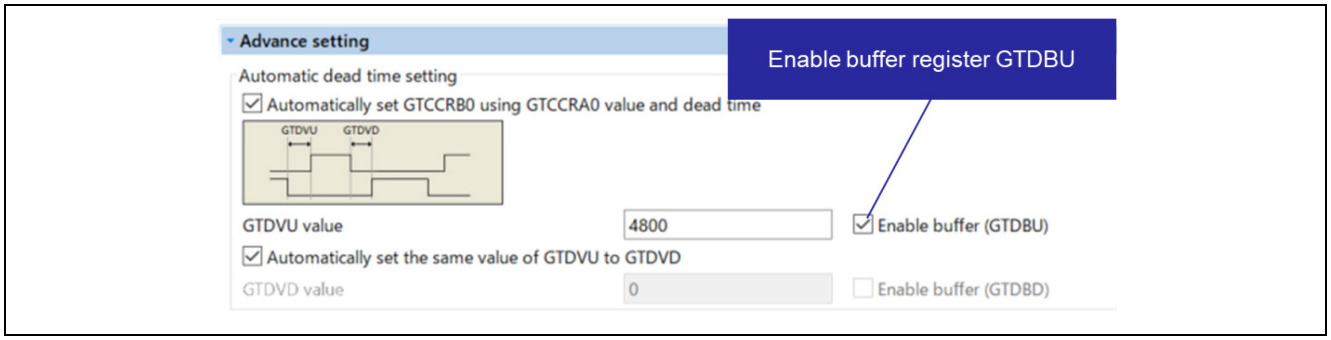


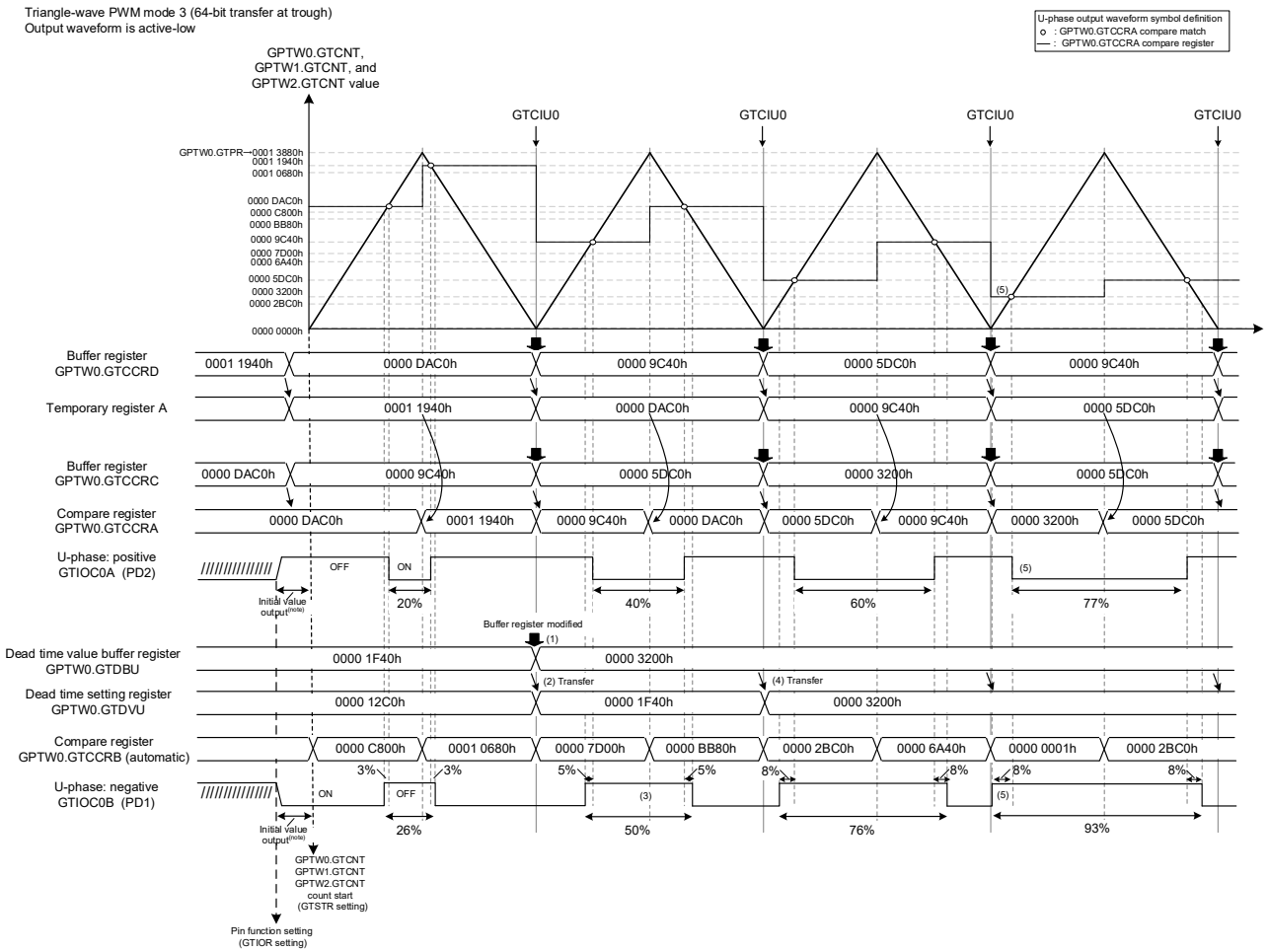
Figure 4.77 Smart Configurator Automatic Dead Time Setting

The figure below shows an example of operations of the automatic dead time setting function using the buffer register of dead time value.

The value of the GTDBU is modified by the GTCNT counter underflow interrupt (GTCIU0) ((1)) in Figure 4.77). The value of the GTDBU is transferred to the dead time setting register GTDVU when a GTCNT counter underflow occurs (at trough) ((2)) in Figure 4.77), and the waveform that secures the dead time period after update is output ((3) in Figure 4.77).

In the 3rd cycle, the GTDBU value that was modified at the end of the 1st cycle is transferred to the GTDVU ((4) in Figure 4.77).

The dead time setting of the 4th cycle results in a dead time error (GTCCRA-GTDVU<0) at the changing point in the first half of the negative-phase, so waveforms of the positive-phase and negative-phase with corrected changing points are output ((5) in Figure 4.77).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.78 Buffer Operations with Automatic Dead Time Setting Function

4.8.5.3 When Automatic Dead Time Setting Function is Not Used

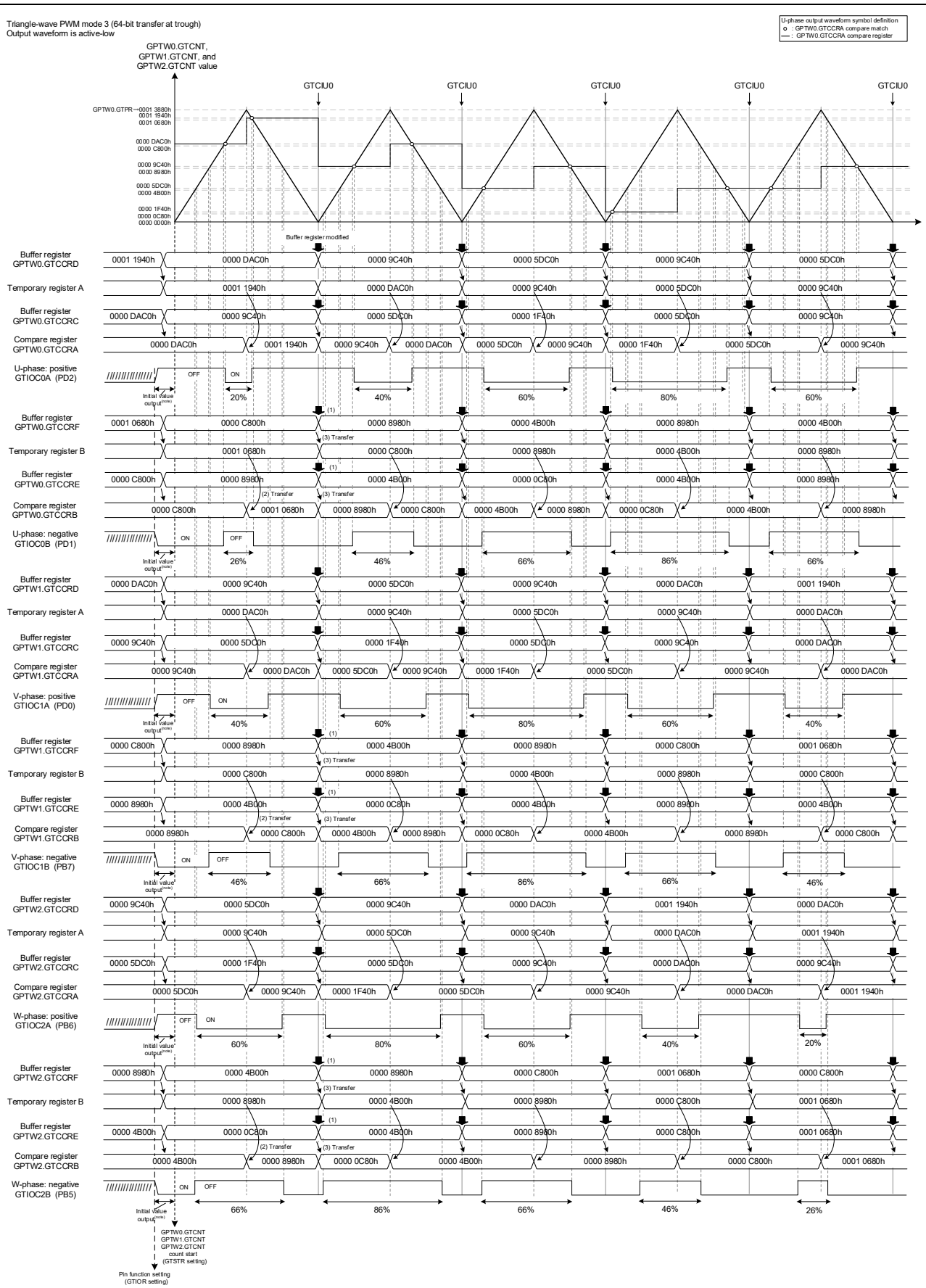
- Target sample code file name: r01an5995_rx66t_gptw_triangle_pwm3.zip

Figure 4.79 shows operations when the automatic dead time setting function is not used (GTDTCR.TDE bit is set to 0).

When the automatic dead time setting function is not used, a waveform that secures the dead time period is generated by setting a value to compare match register GTCCRB, the register for negative-phase waveform, and buffer registers GTCCRE and GTCCRF.

Similar to the positive-phase, in the negative-phase the compare value is updated with every cycle by modifying buffer registers GTCCRE and GTCCRF when a GTCNT counter underflow (trough) occurs ((1) Figure 4.79), transferring temporary register B to compare register GTCCRB when a GTCNT counter overflow (crest) occurs ((2) in Figure 4.79), and transferring buffer register GTCCRF to temporary register B and buffer register GTCCRE to compare register GTCCRB when GTCNT counter underflow (trough) occurs ((3) in Figure 4.79).

In addition, the same dead time period is secured as shown in the operations in Figure 4.69.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.79 When Not Using Automatic Dead Time Setting Function (Triangle-Wave PWM Mode 3)

4.8.6 Usage Notes

4.8.6.1 Count Start for Multiple Channels

In this sample code, the CSTRT0, CSTRT1, and CSTRT2 bits of timer software start register GTSTR are set at the same time in the gpt0_gpt1_gpt2_start function to start GPTW0, GPTW1, and GPTW2 counting at the same time.

When using the R_Config_GPT0_Start, R_Config_GPT1_Start, and R_Config_GPT2_Start functions generated by the Smart Configurator, the counting starts timing may not be the same because each of the functions are read.

Refer to RX66T Group User's Manual: Hardware, section 24.3.8.1 Synchronous Operation by Software.

4.8.6.2 Settings of GTCCRm Register during Compare Match Operation (m = A to F)

This sample code automatically sets the dead time and uses the value of compare match register GTCCRA, the register for positive-phase waveform, to update the value of compare match register GTCCRB, the register for negative-phase waveform.

The values of compare register GTCCRA should be set to satisfy the following restrictions.

GTCCRA > GTDVU
GTCCRA > GTDVD
GTCCRA < GTPR

If the GTCCRA is set to 0000 0000h or a value greater than the setting value of the GTPR during the counting operation, the output protection function is activated.

However, if the conditions below are not satisfied, the function does not operate normally.

When the GTCCRA register value at the start of count operation is greater than 0000 0001h and less than the setting value of the GTPR register

For details, refer to RX66T Group User's Manual: Hardware, section 24.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 9; m=A, B).

When not using the automatic dead time setting function, set a value greater than 0000 0001h and less than the setting value of the GTPR register in the GTCCRA (GTCCRB) register. When 0000 0000h or the same value as the GTPR register is set in the GTCCRA register, a compare match occurs in a cycle only when [GTCCRA (GTCCRB) = 0000 0000h] or [GTCCRA (GTCCRB) = GTPR] is met. Furthermore, if a value exceeding the setting value of the GTPR register is set in the GTCCRA, a compare match does not occur.

For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F), (1) When Automatic Dead Time Setting has been Made in Triangle-Wave PWM Mode and (2) When Automatic Dead Time Setting has not been Made in Triangle-Wave PWM Mode.

4.9 Sawtooth-Wave PWM Mode Duty Cycles 0% to 100%

- Target sample code file name: r01an5995_rx66t_gptw_sawtooth_pwm_50to100.zip

4.9.1 Overview

The GPTW sawtooth-wave PWM mode can be used to output PWM waveforms of duty cycles 0% to 100% according to the GTCCRA register compare match and GTUDDTYC register setting.

This sample code describes a sample code that uses the sawtooth-wave PWM mode and repeats the following waveform output, including duty cycles 0% and 100%.

- Duty switching: 50% → 80% → 100% → 80% → 50% → 0% → ・・・

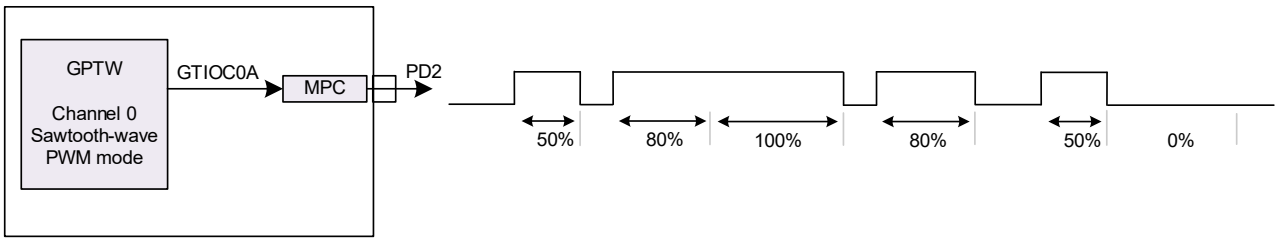
The basic operation is to make changes to the duty cycle by transferring the value of the buffer register to the GTCCRA when a GTCNT counter overflow occurs using buffer register GTCCRC. When switching between duty cycles 0% and 100%, the process to modify the GTUDDTYC register is performed when a GTCNT counter overflow occurs.

The following list provides the GPTW settings used in the sample code.

- Use sawtooth-wave PWM mode
- Use channel 0
- Initial output value = low
- Carrier period = 1ms
- Timer count clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Count direction = up-counting
  - Counter initial value = 0
- Use GTCCRA as duty output compare match
  - Use GTIOC0A pin as PWM output pin
  - Use GTCCRA as compare match
  - Low output at counting starts
  - High output at GTCCRA compare match
  - Low output at cycle end
- Use buffer register
  - GTCCRA operates as single buffer
  - Use GTCCRC as buffer register of GTCCRA
- Software source count start enabled
- Duty changes at each cycle
  - Duty changes at the GTCNT counter overflow interrupt
  - Refer to Figure 4.81 for details on duty change timing

Set in Smart Configurator.
For Setting Methods,
refer to section 4.9.3.

Sawtooth-wave PWM mode output for this sample code is shown below.



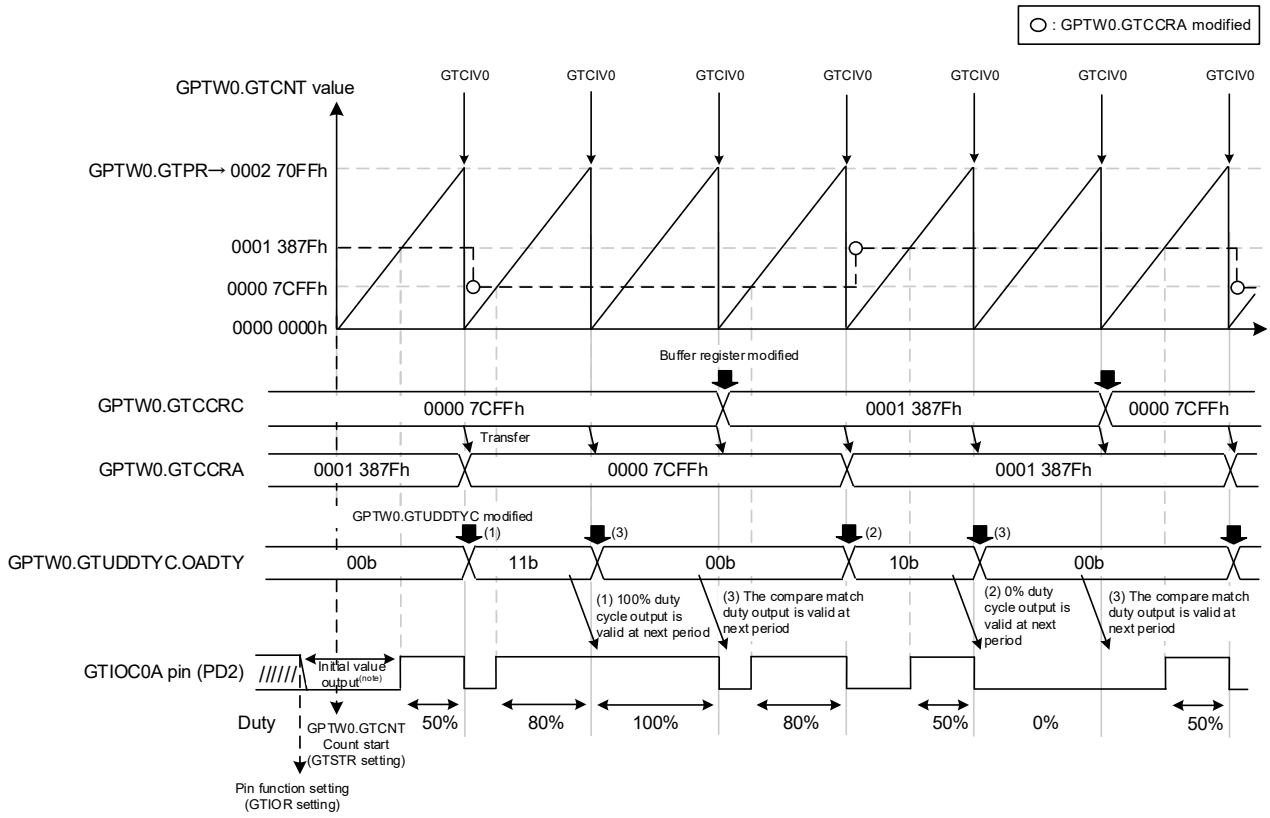
GPTW: Generates sawtooth-wave PWM mode waveform
MPC: Sets pins to be used from general purpose I/O port to peripheral function I/O port setting

Figure 4.80 Sawtooth-Wave PWM Mode Output

4.9.2 Operation Details

The sample code operations are shown in Figure 4.81. The basic operation is to make changes the settings of the duty cycle with each period by modifying the value of buffer register GTCCRC at the GTCNT counter overflow interrupt (GTCIV0) and transferring the value of GTCCRC to the GRCCRA when a GTCNT counter overflow occurs.

- 100% Duty Cycle Output ((1) in figure below)
  Output goes to high from the next cycle by setting the GTUDDTYC.OADTY bits to 11b. The waveform does not change even if a GTCCRA compare match occurs.
- 0% Duty Cycle Output ((2) in figure below)
  Output goes to low from the next cycle by setting the GTUDDTYC.OADTY bits to 10b. The waveform does not change even if a compare match occurs.
- Switching from Duty Cycle 100% or 0% ((3) in figure below)
  The duty output can be changed from the next cycle by a compare by setting the GTUDDTYC.OADTY bit to 00b.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.81 Sample Code Operations

4.9.3 Smart Configurator Settings

The sample codes use the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.12 Adding Components

Item	Description
Component	General PWM Timer
Configuration Name	Config_GPT0
Work mode	Sawtooth-wave PWM Mode
Resource	GPT0

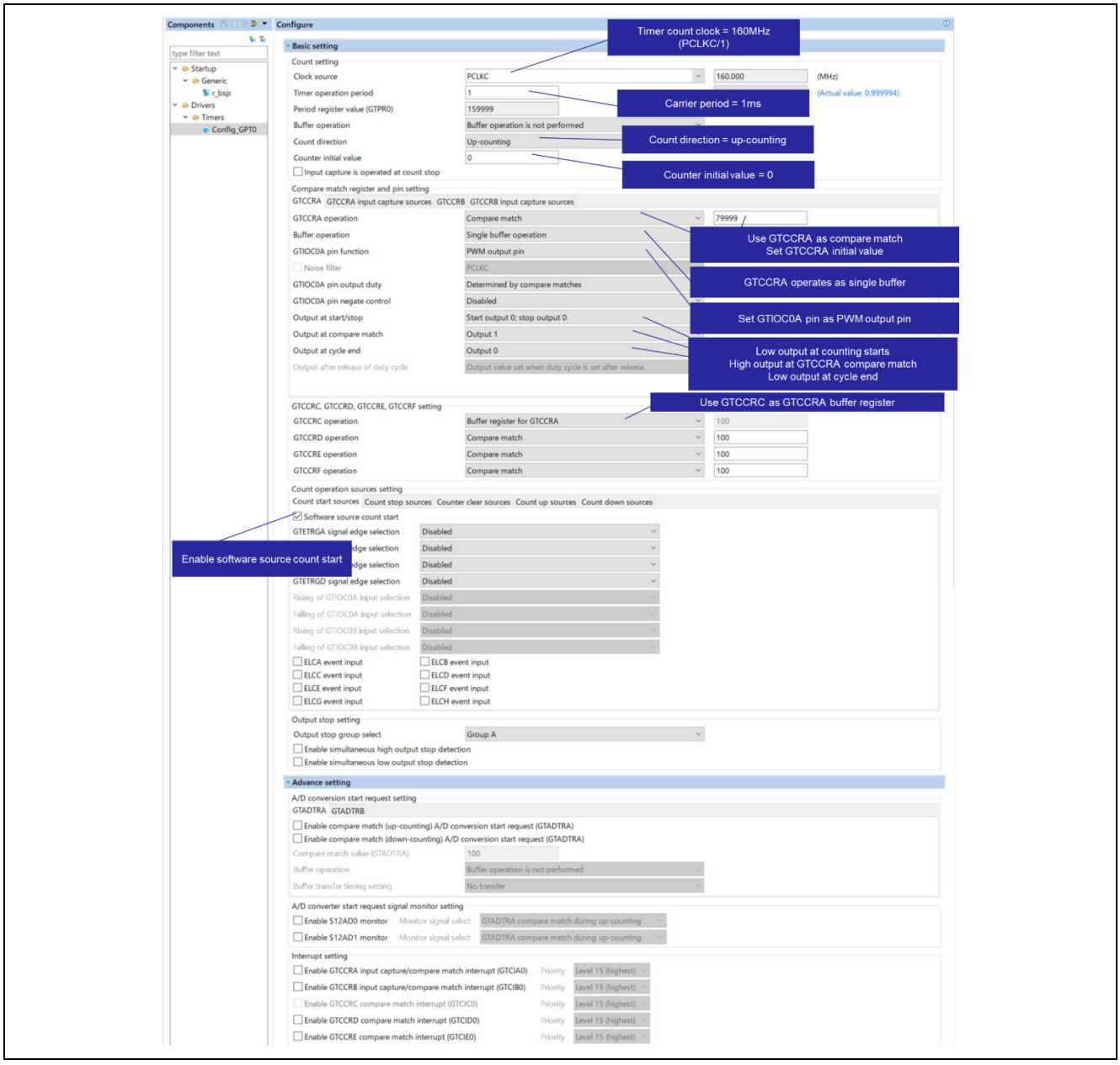


Figure 4.82 GPT0 Settings (1/2)

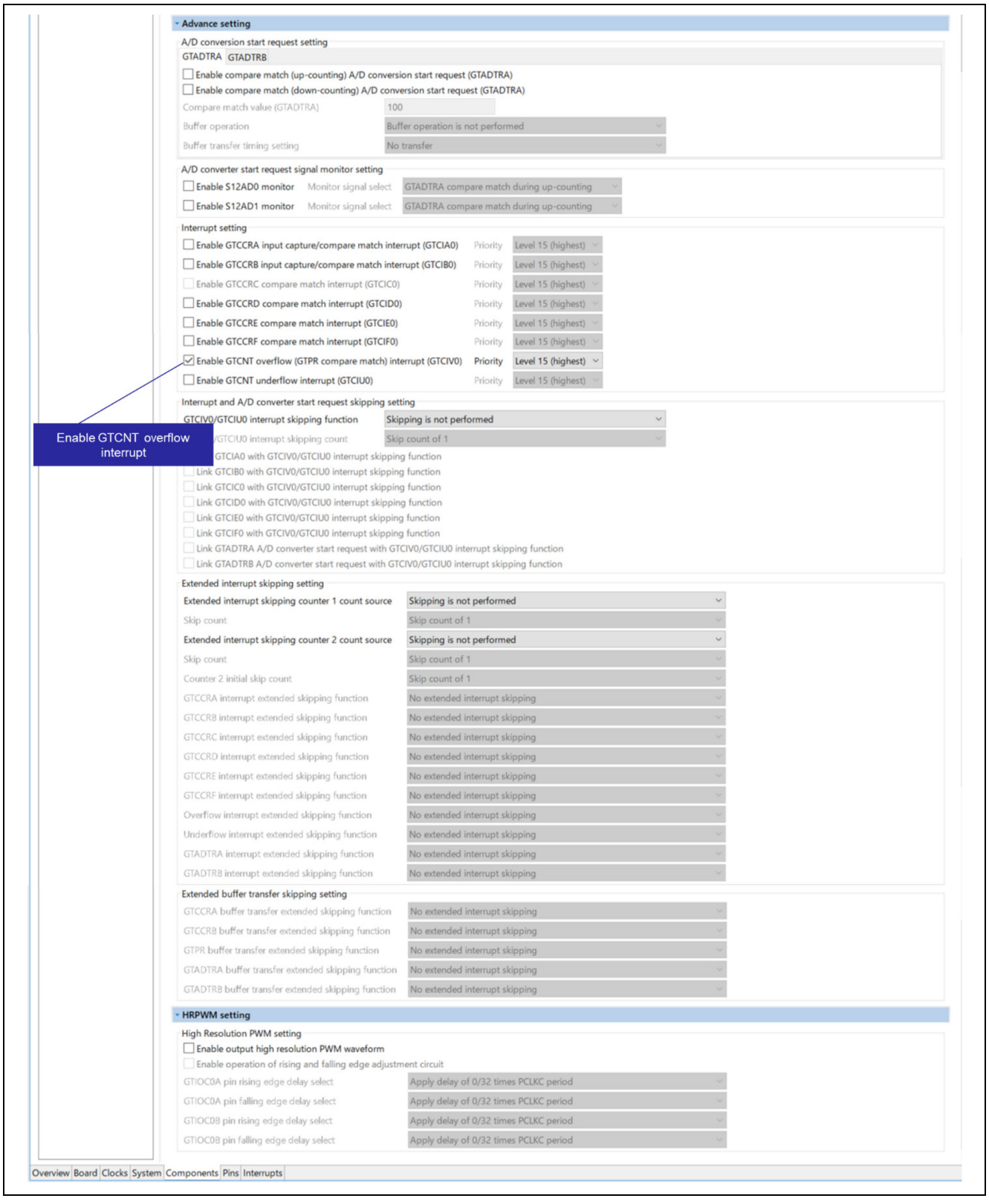


Figure 4.83 GPT0 Settings (2/2)

4.9.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

Counting is started in the main function.

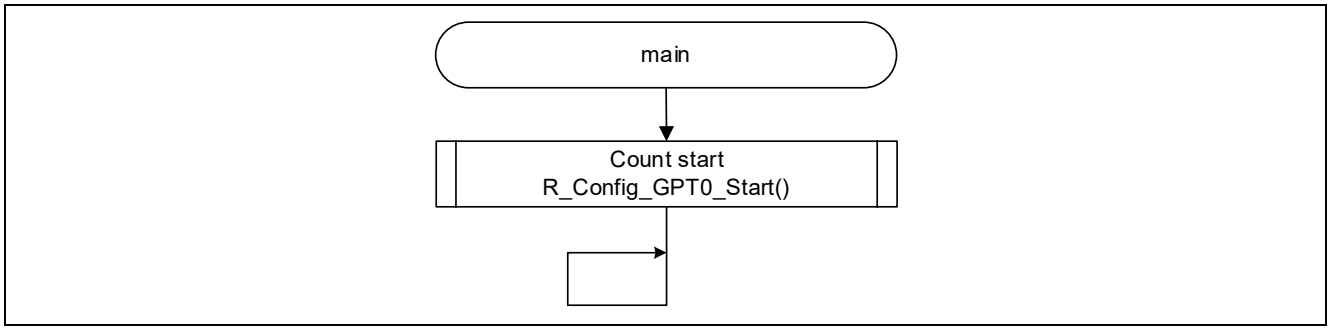


Figure 4.84 main Function

The user initialization function R_Config_GPT0_Create_UserInit, which is executed before the main function, sets the initial values of the buffer registers and initializes the variables. This function is called from within the R_Config_GPT0_Create function.

This sample code uses the following variable.

- s_duty_list_counter: counter variable for reading from the duty cycle list

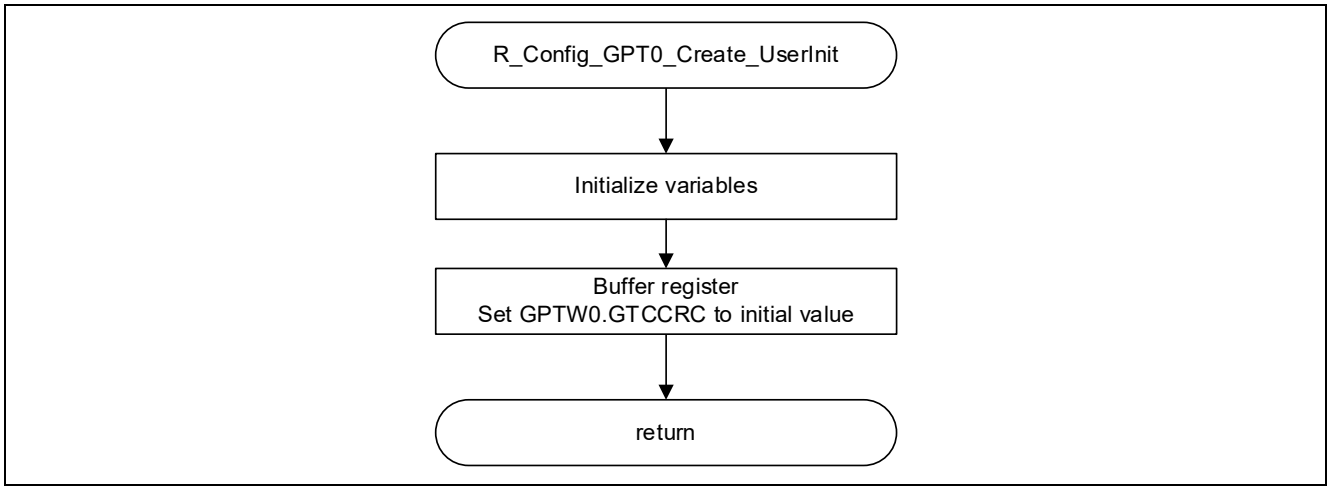


Figure 4.85 User Initialization Function

The GTCIV0 interrupt handler function changes the values of the GTCCRC register and the GTUDDTYC register according to the next duty cycle to be set.

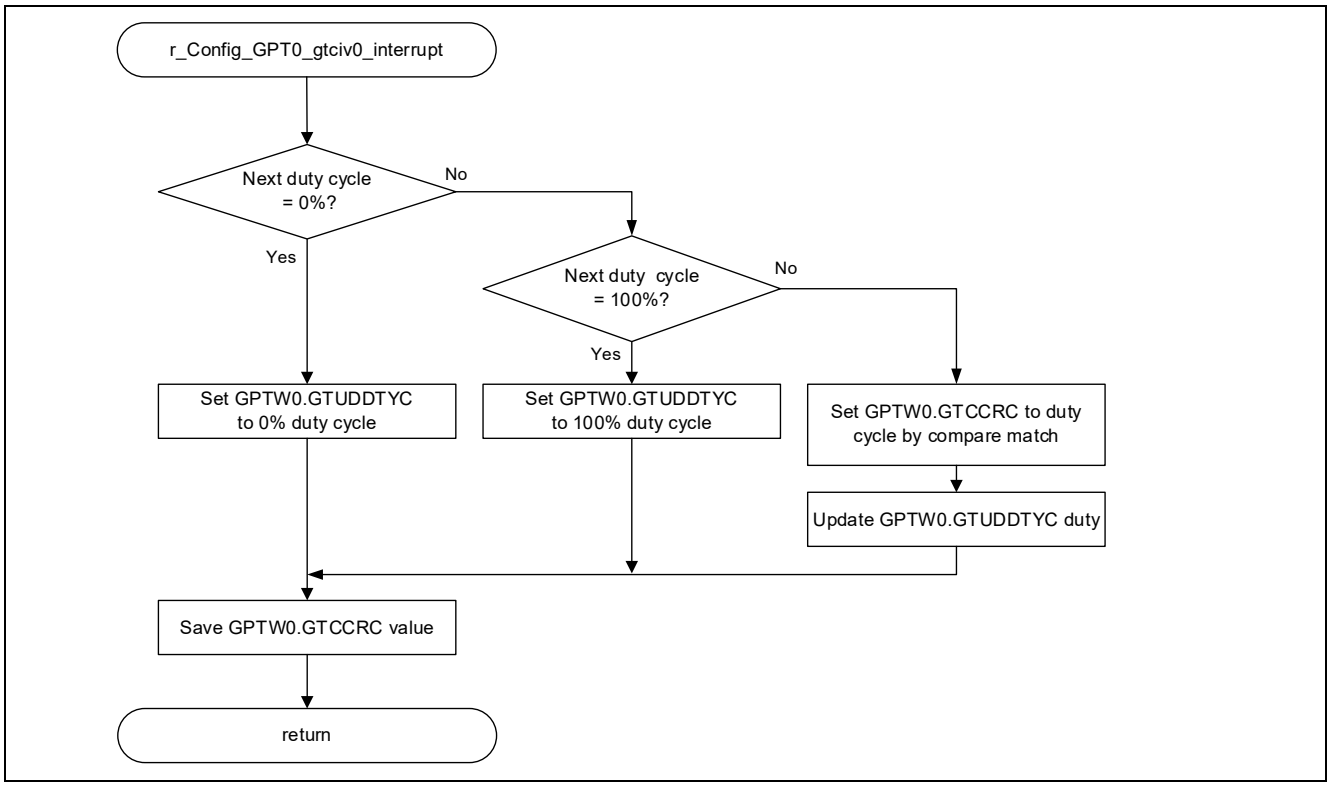


Figure 4.86 GTCIV0 Interrupt Handler Function

4.9.5 Usage Notes

4.9.5.1 Settings of GTCCRm Register during Compare Match Operation (m = A to F)

This sample code automatically sets the dead time and uses the value of compare match register GTCCRA, the register for positive-phase waveform, to update the value of compare match register GTCCRB, the register for negative-phase waveform.

The values of buffer registers GTCCRC and GTCCRD should be set to satisfy the following restrictions. If the restrictions are not met, correct output waveforms with secured dead time may not be obtained.

- In up-counting:
  GTCCRC < GTCCRD
  GTCCRC > GTDVU
  GTCCRD < GTPR – GTDVD
- In down-counting:
  GTCCRC > GTCCRD
  GTCCRC < GTPR – GTDVU
  GTCCRD > GTDVD

Further, if the dead time is not automatically set, buffer registers GTCCRC and GTCCRD (or GTCCRE and GTCCRF) should be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC (GTCCRE) < GTCCRD (GTCCRF) < GTPR
- In down-counting: GTPR > GTCCRC (GTCCRE) > GTCCRD (GTCCRF) > 0

For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F), (3) When Automatic Dead Time Setting has been Made in Sawtooth-Wave One-Shot Pulse Mode and (4) When Automatic Dead Time Setting has not been Made in Sawtooth-Wave One-Shot Pulse Mode.

4.9.5.2 100% Duty Cycle Output at Compare Match

Output duty cycle cannot be switched to high-active 100% duty at compare match without changing the GTUDDTYC register. To output high-active 100% duty cycle, set the GTUDDTYC.OADTY bits to 11b.

If the GTCCRA register is set to 0 in this sample code settings, 100% duty cycle cannot be output because after a GTCNT counter overflow occurs, low is output for one clock cycle followed by high output.

The MTU can output 100% duty cycle because the duty register and period register are set to the same value and the waveform does not change when counter clear and compare match occur at the same time. In the GPTW sawtooth-wave PWM mode, 100% duty cycle cannot be output in the same manner because when the end of a cycle and a compare match occur at the same time, the output settings at the end of the period have priority and the waveform changes.

For details on waveform output when GTCNT counter overflow and compare match occur at the same time in the GPTW, refer to the notes following Table 24.4 in RX66T Group User's Manual: Hardware, section 24.2.14 General PWM Timer I/O Control Register (GTIOR).

4.9.5.3 Compare Match Operation during Duty Cycles 0% and 100% Output

In this sample code, output duty cycles 0% and 100% are determined based on the values set to the GTUDDTYC.OADTY bits. When duty cycle is set to either 0% or 100%, the compare match operation continues in the GPTW and interrupt output and buffer transfer operation are performed.

This sample code does not use the compare match interrupt, so if you are using the compare match interrupt, do so with caution interrupt during duty cycle 0% and 100% output.

4.10 Sawtooth-Wave One-Shot Pulse Duty Cycles 0% to 100%

- Target sample code file name: r01an5995_rx66t_gptw_sawtooth_1shotpls_50to100_dt.zip

4.10.1 Overview

When using the GPTW sawtooth-wave one-shot pulse mode, the GTCNT counter can be operate in sawtooth-wave (half-wave) by setting the period in the GTPR register, and PWM waveforms of duty cycles 0% to 100% can be output from the GTIOCnA and GTIOCnB pins (n = 0 to 9) by the compare match between the GTCCRA and GTCCRB registers.

This sample code describes a sample code that uses the automatic dead time setting function in the sawtooth-wave one-shot pulse mode and repeats the following waveform output, including duty cycles 0% and 100%.

- GTIOC0A pin high-width switching: 50% → 80% → 100% → 80% → 50% → 0% → ⋯
- GTIOC0B pin low-width switching: 60% → 90% → 100% → 90% → 60% → 0% → ⋯

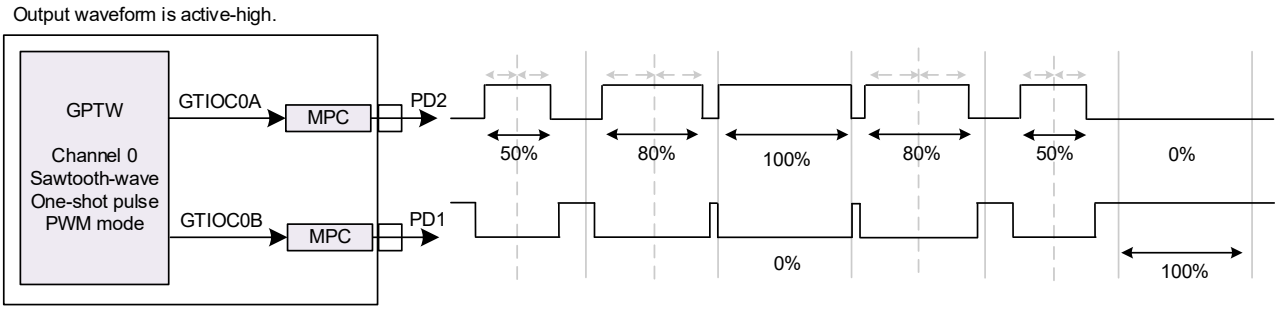
The value of temporary register A is transferred to compare register GTCCRA when a GTCCRA compare match occurs. The duty cycle is changed by the transfer from buffer register GTCCRD to temporary register A and from buffer register GRCCRC to compare register GTCCRA, respectively, when a GTCNT counter overflow occurs.

The following list provides the GPTW settings used in the sample code.

- Use sawtooth-wave one-shot pulse mode
- Use channel 0
- Carrier period = 1ms
- Timer count clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Count direction = up-counting
  - Counter initial value = 0
- Use GTCCRA as duty output compare match
  - Set GTIOC0A pin as PWM output pin
  - Low output at counting starts
  - Toggle output at GTCCRA compare match
  - Retain output at cycle end
- Use GTCCRB as duty output compare match
  - Set GTIOC0B pin as PWM output pin
  - High output at counting starts
  - Toggle output at GTCCRB compare match
  - Retain output at cycle end
- Use double buffer register
  - Use GTCCRC and as buffer registers of GTCCRD
- Use automatic dead time generation
- Software source count start enabled
- Duty changes at each cycle
  - Duty changes at the GTCNT counter overflow interrupt
  - Refer to Figure 4.89 for details on duty change timing

Set in Smart Configurator.
For Setting Methods,
refer to section 4.10.3.

Sawtooth-wave one-shot pulse mode output for this sample code is shown below.



GPTW: Generates sawtooth-wave one-shot pulse mode waveform
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

Figure 4.87 Sawtooth-Wave One-Shot Pulse Mode Output

4.10.2 Operation Details

The sample code operations are shown in Figure 4.89. The settings of the duty cycle are changed with each period by modifying the value of buffer registers GTCCRC and GTCCRD at the GTCNT counter overflow interrupt (GTCIV0) ((1) in Figure 4.89).

The initial values of the buffer registers for the code generated using the Smart Configurator are set to the same value as the compare register in this sample code. As a result, the buffer register values are set in the user initialization function R_Config_GPT0_Create_UserInit before the counting starts. The values set in the registers are transferred from buffer register GTCCRD to temporary register A and from buffer register GTCCRC to compare register GTCCRA, respectively, by forced buffer transfers ((2) in Figure 4.89).

After the counting starts, the basic operation is to update the data in the compare register by repeating steps 1 and 2 below.

1. Transfer from temporary register A to compare register GTCCRA when a GTCCRA compare match occurs ((3) in Figure 4.89).
2. Transfer from buffer register GTCCRD to temporary register A and from buffer register GRCCRC to compare register GTCCRA, respectively, when a GTCNT counter overflow occurs ((4) in Figure 4.89).

In addition, the GTCCRB register is automatically set according to the GTCCRA update because the automatic dead time function is used. The same values are set for the GTDVU and GTDVD.

- 100% Duty Cycle Output ((5) in Figure 4.89)
  The GTIOC0A pin outputs high from the time of the next GTCNT counter overflow occurs by setting the GTUDDTYC.OADTY bits to 11b, and the waveform does not change even if a GTCCRA compare match occurs.
  The GTIOC0B pin outputs low from the time of the next GTCNT counter overflow occurs by setting the GTUDDTYC.OBDTY bits to 10b, and the waveform does not change even if a GTCCRB compare match occurs. In this sample code, the bits are set to 10b so that the GTIOC0B pin outputs low for one cycle period.

- 0% Duty Cycle Output ((6) in Figure 4.89)
  The GTIOC0A pin outputs low from the time of the next GTCNT counter overflow occurs by setting the GTUDDTYC.OADTY bits to 10b, and the waveform does not change even if a GTCCRA compare match occurs.
  The GTIOC0B pin outputs high from the time of the next GTCNT counter overflow occurs by setting the GTUDDTYC.OBDTY bits to 11b, and the waveform does not change even if a GTCCRB compare match occurs. In this sample code, the bits are set to 11b so that the GTIOC0B pin outputs high for one cycle period.

- Switching from Duty Cycles 100% and 0% ((7) in Figure 4.89)
  The GTIOC0A pin outputs duty cycle according to the GTCCRA compare match from the time of the next GTCNT counter overflow occurs by setting the GTUDDTYC.OADTY bits to 00b. The output after duty cycles 100% and 0% are released is determined by the GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR bits. In this sample code, the compare match output value low which was masked is output.
  The GTIOC0B pin outputs duty cycle according to the GTCCRB compare match from the time of the next GTCNT counter overflow occurs by setting the GTUDDTYC.OBDTY bits to 00b. The output after duty cycles 100% and 0% are released is determined by the GTIOR.GTIOB[3:2] and GTUDDTYC.OBDTYR bits. In this sample code, the compare match output value high which was masked is output.

  The following shows example of the GTIOC0A pin output after duty cycles 100% and 0% are released.

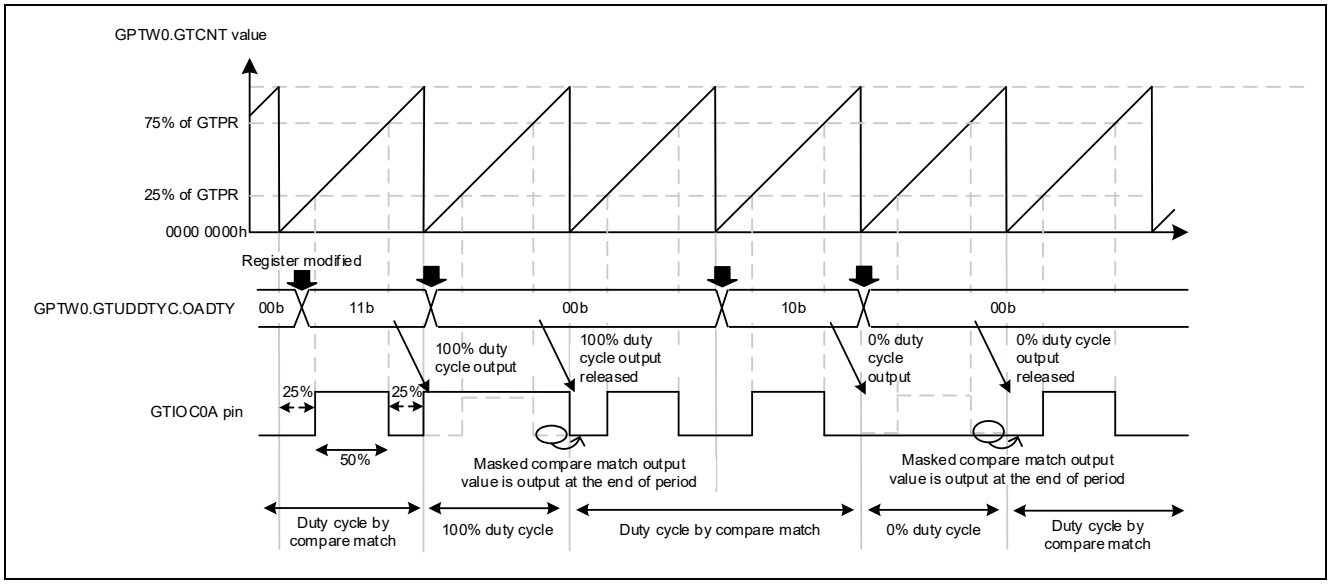
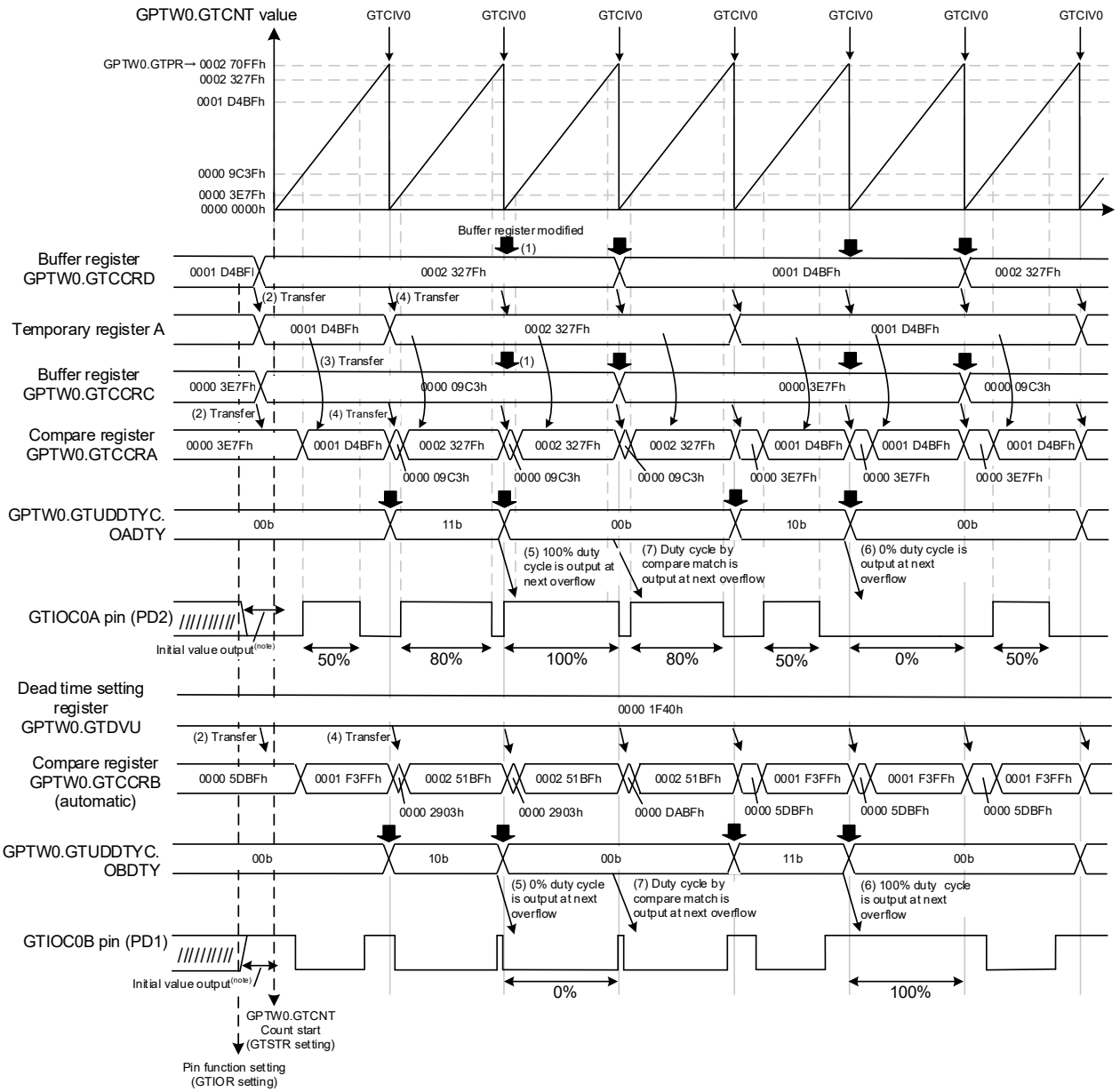


Figure 4.88 Duty Generation and Output After Release of Duty Cycles 100% and 0%



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.89 Sample Code Operations

4.10.3 Smart Configurator Settings

The sample codes use the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.13 Adding Components

Item	Description
Component	General PWM Timer
Configuration Name	Config_GPT0
Work mode	Sawtooth-wave One-shot Pulse Mode
Resource	GPT0

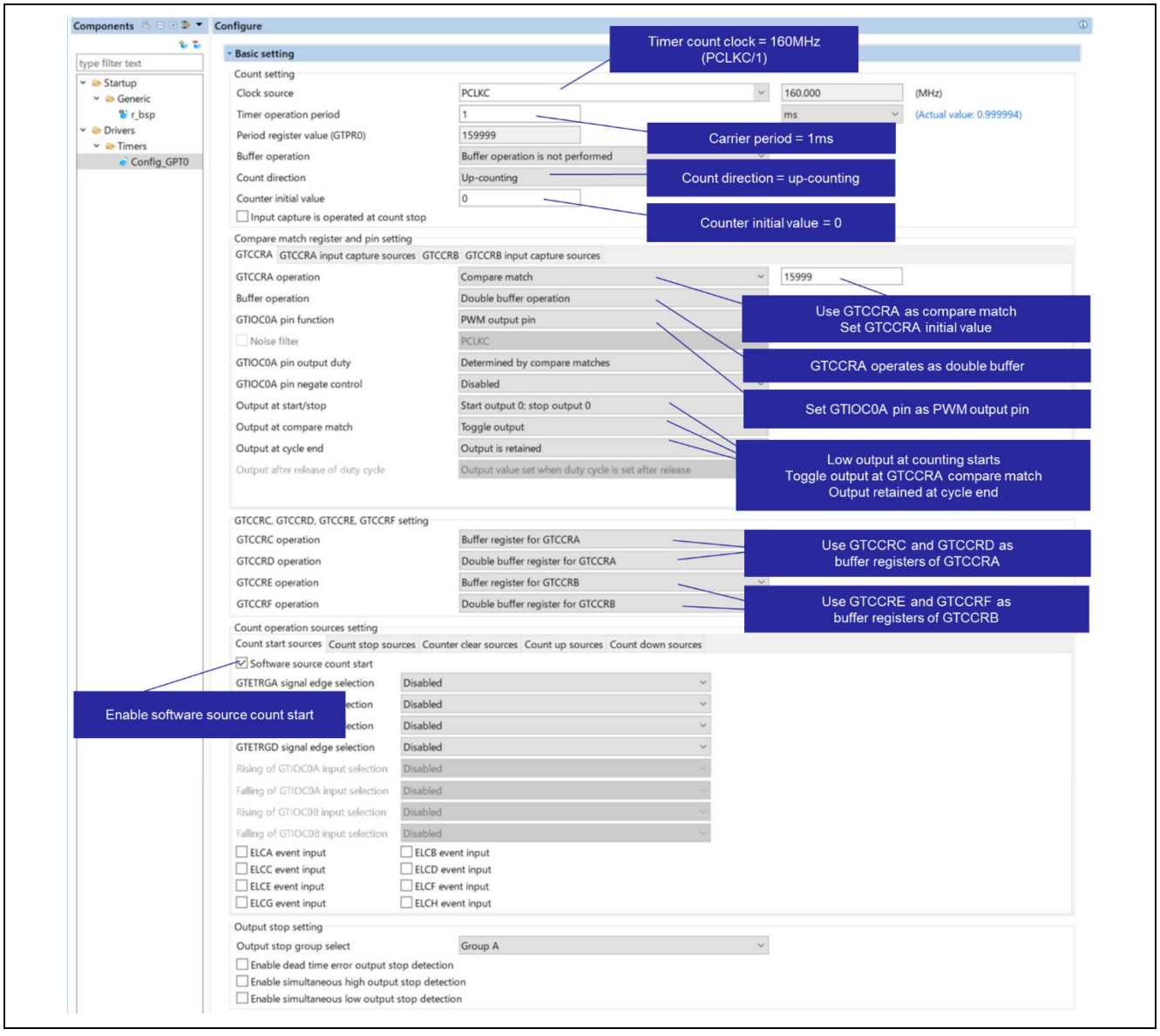


Figure 4.90 GPT0 Settings (1/2)

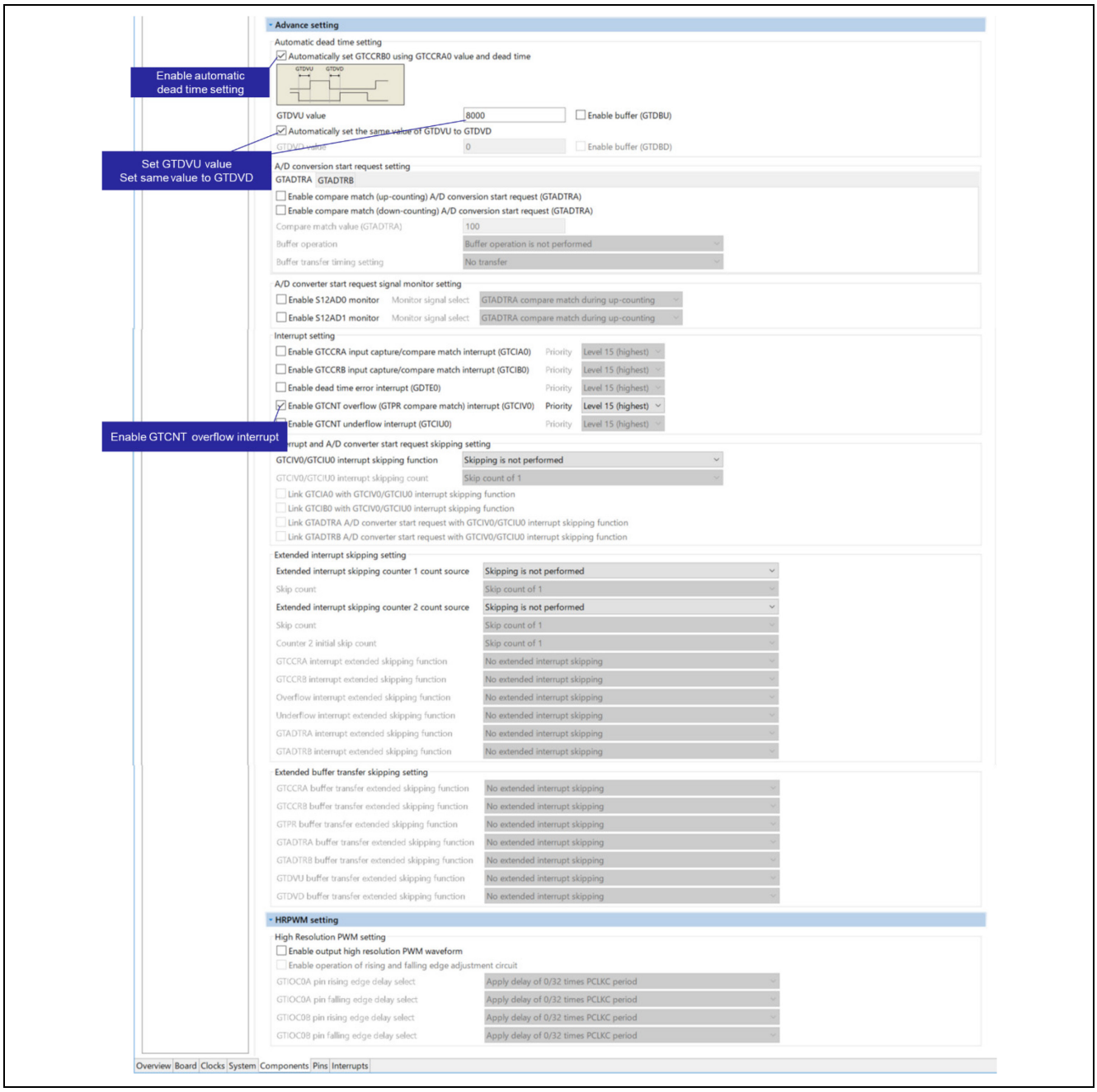


Figure 4.91 GPT0 Settings (2/2)

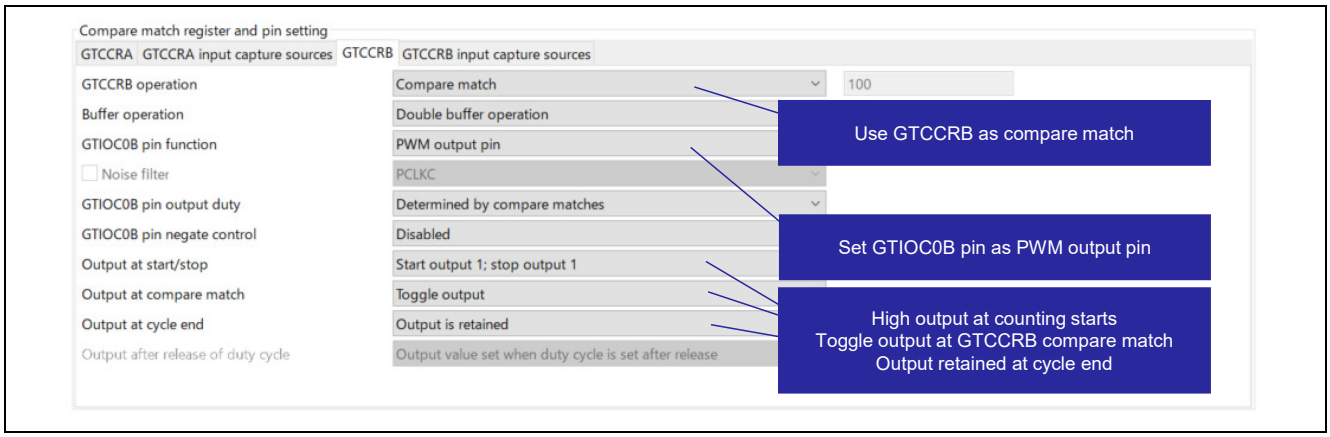


Figure 4.92 GPT0 Settings (Compare Match Register and Pins Setting of GTCCRB)

4.10.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

Counting is started in the main function.

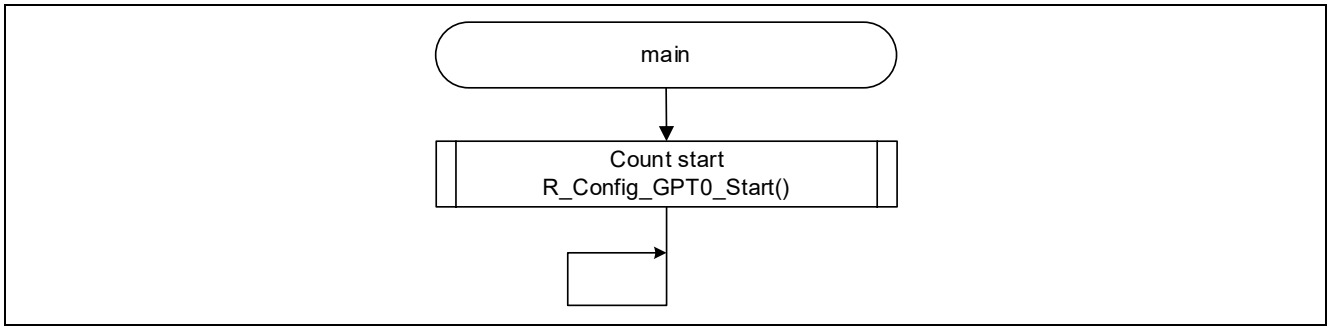


Figure 4.93 main Function

The user initialization function R_Config_GPT0_Create_UserInit, which is executed before the main function, sets the initial values of the buffer registers. In order to set the second compare match register value in the 1st cycle, a forced buffer transfer is performed after setting the buffer register value, and then the temporary register and the compare register values are set. This function is called from within the R_Config_GPT0_Create function.

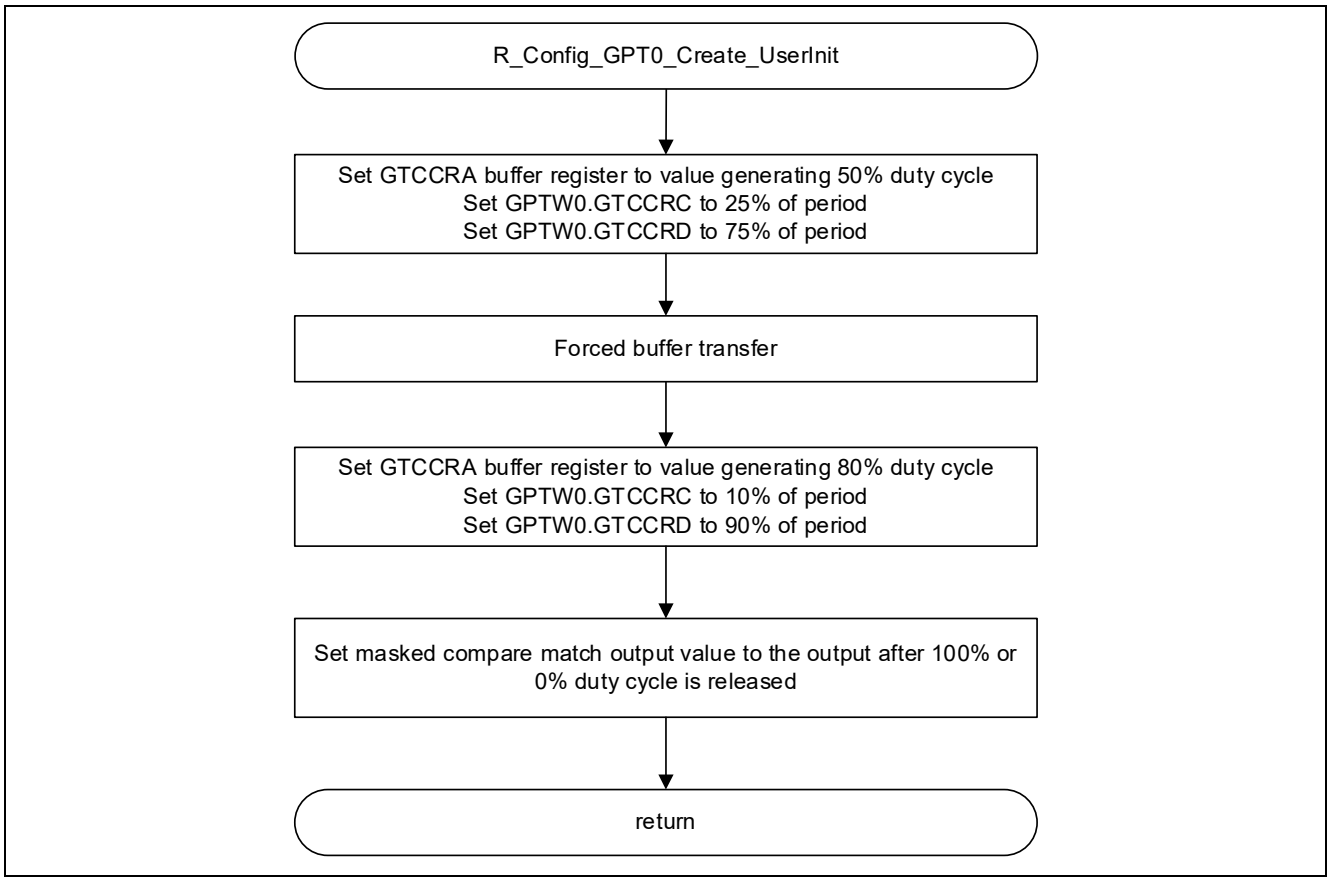


Figure 4.94 User Initialization Function

The GTCIV0 interrupt handler function changes the values of buffer registers GTCCRC and GTCCRD and the GTUDDTY register.

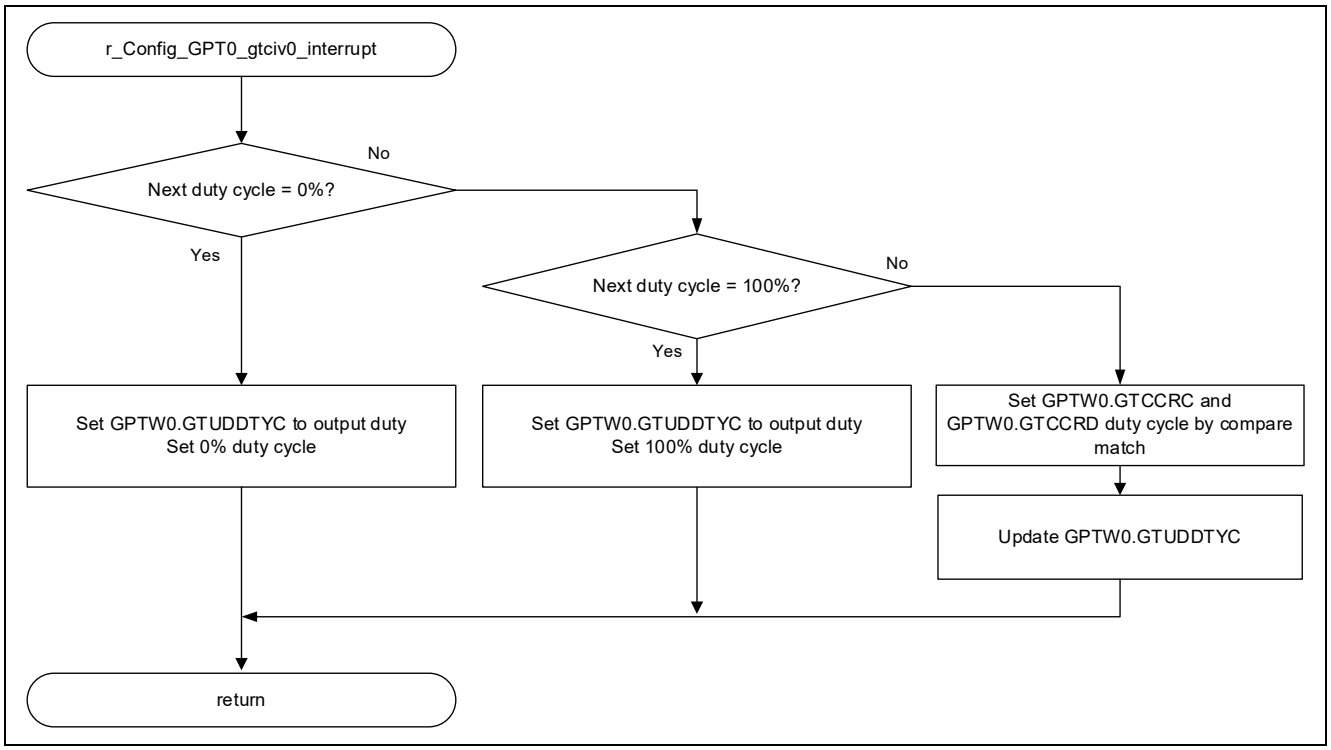


Figure 4.95 GTCIV0 Interrupt Handler Function

4.10.5 Usage Notes

4.10.5.1 Settings of GTCCRm Register during Compare Match Operation (m = A to F)

This sample code automatically sets the dead time and uses the value of compare match register GTCCRA, the register for positive-phase waveform, to update the value of compare match register GTCCRB, the register for negative-phase waveform.

The value of compare register GTCCRA should be set to satisfy the following restrictions. If the restrictions are not satisfied, correct output waveforms with secured dead time may not be obtained.

- In up-counting:
  GTCCRC < GTCCRD
  GTCCRC > GTDVU
  GTCCRD < GTPR – GTDVD
- In down-counting:
  GTCCRC > GTCCRD
  GTCCRC < GTPR – GTDVU
  GTCCRD > GTDVD

Further, if the dead time is not automatically set, buffer registers GTCCRC and GTCCRD should be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC (GTCCRE) < GTCCRD (GTCCRF) < GTPR
- In down-counting: GTPR > GTCCRC (GTCCRE) > GTCCRD (GTCCRF) > 0

For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F), (3) When Automatic Dead Time Setting has been Made in Sawtooth-Wave One-Shot Pulse Mode and (4) When Automatic Dead Time Setting has not been Made in Sawtooth-Wave One-Shot Pulse Mode.

4.10.5.2 100% Duty Cycle Output at Compare Match

Output duty cycle cannot be switched to 100% duty at compare match without changing the GTUDDTYC register. To output 100% duty cycle, set the GTUDDTYC.OADTY bits to 11b and the GTUDDTYC.OBDTY bits to 10b.

If the GTCCRA and GTCCRB register are set to 0 in this sample code settings, 100% duty cycle cannot be output because one clock cycle is output after a GTCNT counter overflow occurs.

When the GTCCRA and GTCCRB registers are set to the same value as the GTPR and a GTCNT counter overflow and compare match occur at the same time, the output setting at the timing of the compare match is prioritized and toggled, disabling 100% duty cycle output.

For details on waveform output when GTCNT counter overflow and compare match occur at the same time in the GPTW, refer to the notes following Table 24.4 in RX66T Group User's Manual: Hardware, section 24.2.14 General PWM Timer I/O Control Register (GTIOR).

4.10.5.3 Compare Match Operation during Duty Cycles 0% and 100% Output

In this sample code, output duty cycles 0% and 100% are determined based on the values set to the GTUDDTYC.OADTY and GTUDDTYC.OBDTY bits. When duty cycle is set to either 0% or 100%, the compare match operation continues in the GPTW and interrupt output and buffer transfer operations are performed.

This sample code does not use the compare match interrupt, so if you are using the compare match interrupt, do so with caution interrupt during duty cycle 0% and 100% output.

4.10.5.4 Switching from Duty Cycles 0% and 100%

If the output setting is changed by a compare match after duty cycle 0% or 100% is set, the output value at the end of the period is determined by the values of the GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR bits.

The following are the settings for this sample code. Note that if the initial hardware value of the GTUDDTYC.OADTYR bit is 0, the same operation as this sample code cannot be performed. The same applies for the GTIOC0B pin.

- GTIOR.GTIOA[3:2] = 00b: Retain output at cycle end
- GTUDDTYC.OADTYR = 1: After releasing the duty cycle 0%/100% setting, apply the GTIOA [3: 2] bit function to the compare match output value that was masked.
- GTIOR.GTIOB[3:2] = 00b: Retain output at cycle end
- GTUDDTYC.OBDTYR = 1: After releasing the duty cycle 0%/100% setting, apply the GTIOA [3: 2] bit function to the compare match output value that was masked.

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.6 Duty Cycle 0%/100% Output Function.

4.11 Sawtooth-Wave PWM Mode Duty Cycles 0% and 100%

- Target sample code file name: r01an5995_rx66t_gptw_sawtooth_pwm_0to100.zip

4.11.1 Overview

The GPTW sawtooth-wave PWM mode can be used to output PWM waveforms of duty cycles 0% to 100% according to the GTCCRA register compare match and the GTUDDTYC register setting.

This sample code describes a sample code that uses the sawtooth-wave PWM mode and repeats waveform output alternating between duty cycles 0% and 100%.

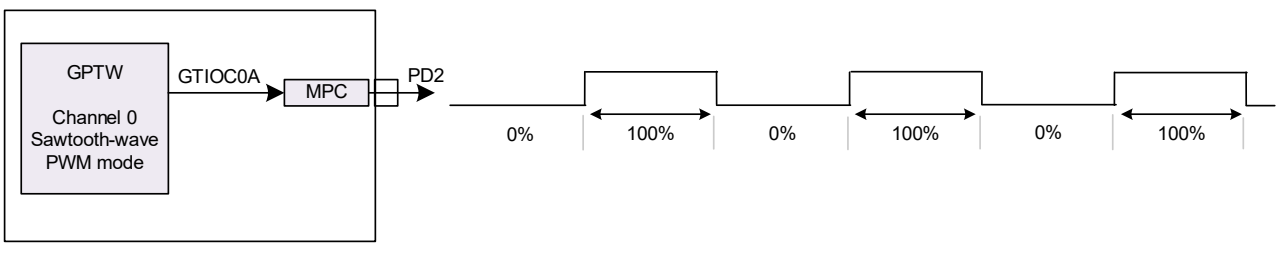
Output switches between duty cycles 0% and 100% by modifying the GTUDDTYC register when a GTCNT counter overflow occurs.

The following list provides the GPTW settings used in the sample code.

- Use sawtooth-wave PWM mode
- Use channel 0
- Initial output value = low
- Carrier period = 1ms
- Timer count clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Count direction = up-counting
  - Counter initial value = 0
- Use GTCCRA as duty output compare match
  - Set GTIOC0A pin as PWM output pin
  - GTCCRA as compare match
  - Low output at counting starts
  - High output at GTCCRA compare match
  - Low output at cycle end
  - Forced output 0% duty cycle at counting start*
- Use buffer registers
  - GTCCRA operates as single buffer
  - Use GTCCRC as buffer register of GTCCRA
- Software source count start enabled
- Duty changes at each cycle
  - Duty changes at the GTCNT counter overflow interrupt
  - Refer to Figure 4.97 for details on duty change timing

Set in Smart Configurator.
For Setting Methods,
refer to section 4.11.3
(except for items marked with *).

The following shows the sawtooth-wave PWM mode output for the sample code.



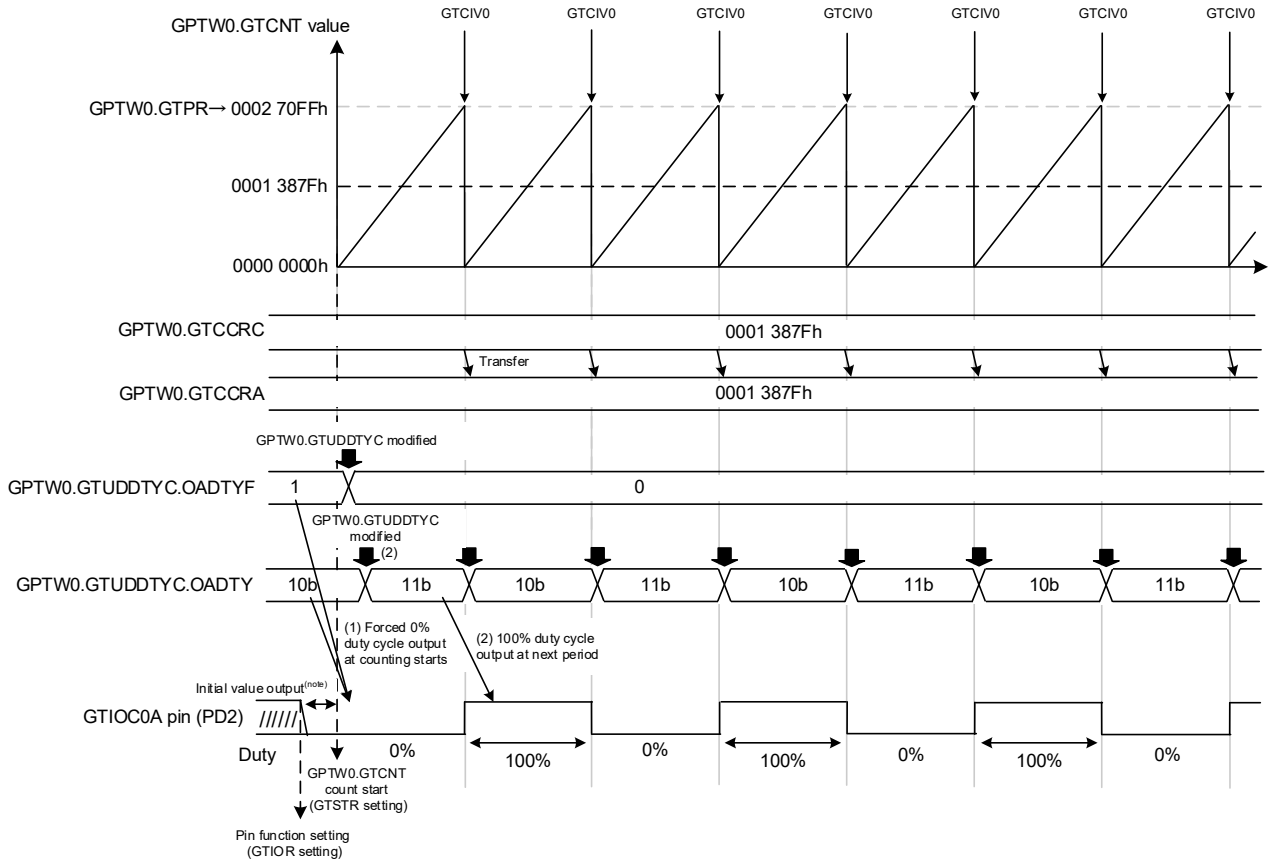
GPTW: Generates sawtooth-wave PWM waveform
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

Figure 4.96 Sawtooth-Wave PWM Mode Output

4.11.2 Operation Details

The sample code operations are shown in Figure 4.97. The duty cycle is alternated between 0% and 100% with each period by modifying the value of the GTUDDTYC register at the GTCNT counter overflow interrupt (GTCIV0).

- 0% Duty Cycle Output After Counting Starts ((1) in figure below)
  Low is output immediately after the counting starts by setting the GTUDDTYC.OADTY bits to 10b while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped. The waveform does not change even if a GTCCRA compare match occurs.
- 0% Duty Cycle Output at 2nd Cycle ((2) in figure below)
  High is output from the 2nd cycle by setting the GTUDDTYC.OADTY bits to 11b before the first GTCNT counter overflow occurs after the counting starts. The waveform does not change even if a GTCCRA compare match occurs.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

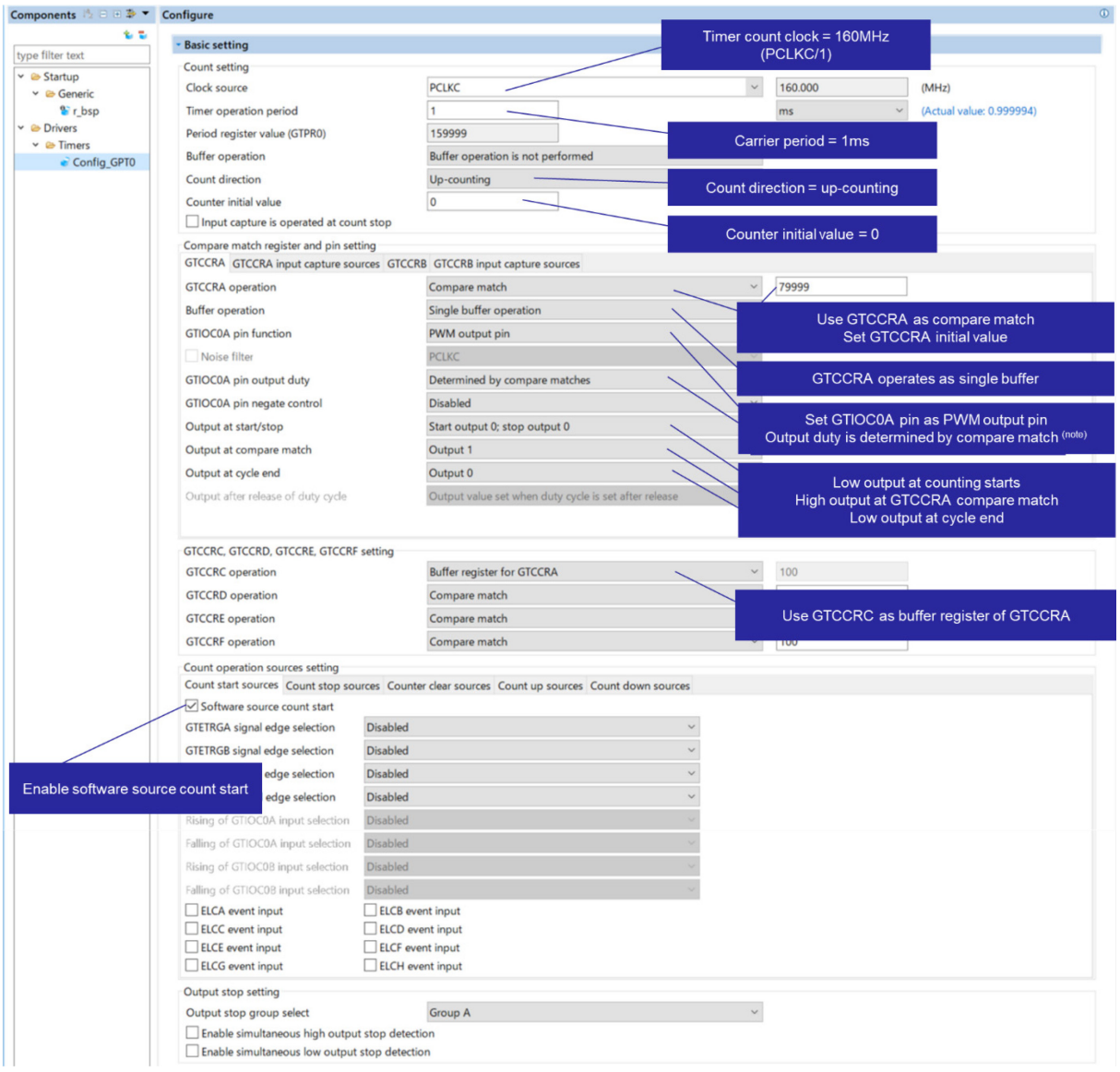
Figure 4.97 Sample Code Operations

4.11.3 Smart Configuration Settings

The sample codes use the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.14 Adding Components

Item	Description
Component	General PWM Timer
Configuration Name	Config_GPT0
Work mode	Sawtooth-wave PWM mode
Resource	GPT0



Note: After setting to forced output duty when the counting starts, the output duty cycle needs to be set to 0%. Set 0% duty cycle output in the R_Config_GPT0_Create_UserInit function

Figure 4.98 GPT0 Settings (1/2)

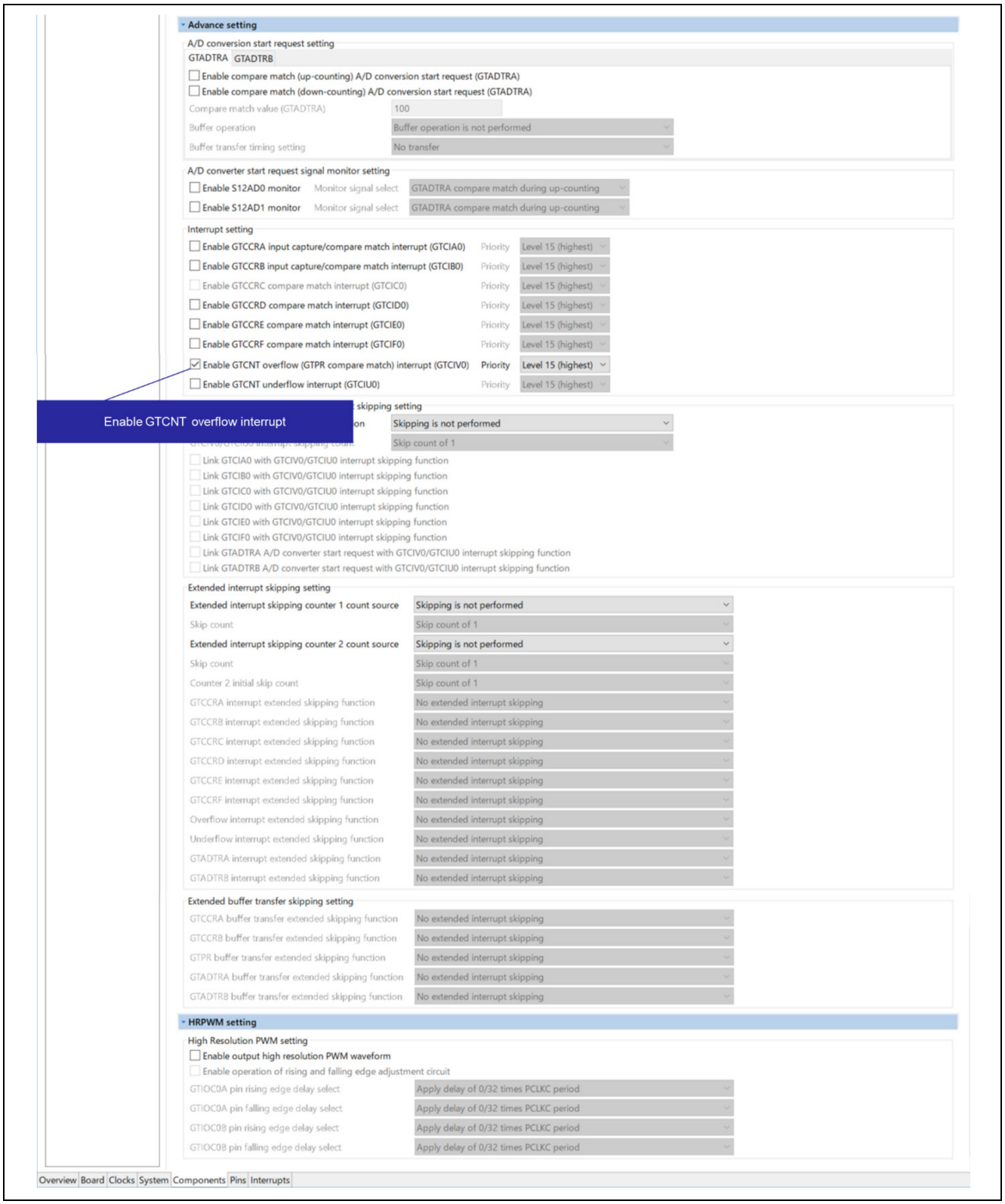


Figure 4.99 GPT0 Settings (2/2)

4.11.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

In the main function, counting is started and output is set to 100% duty for the next cycle.

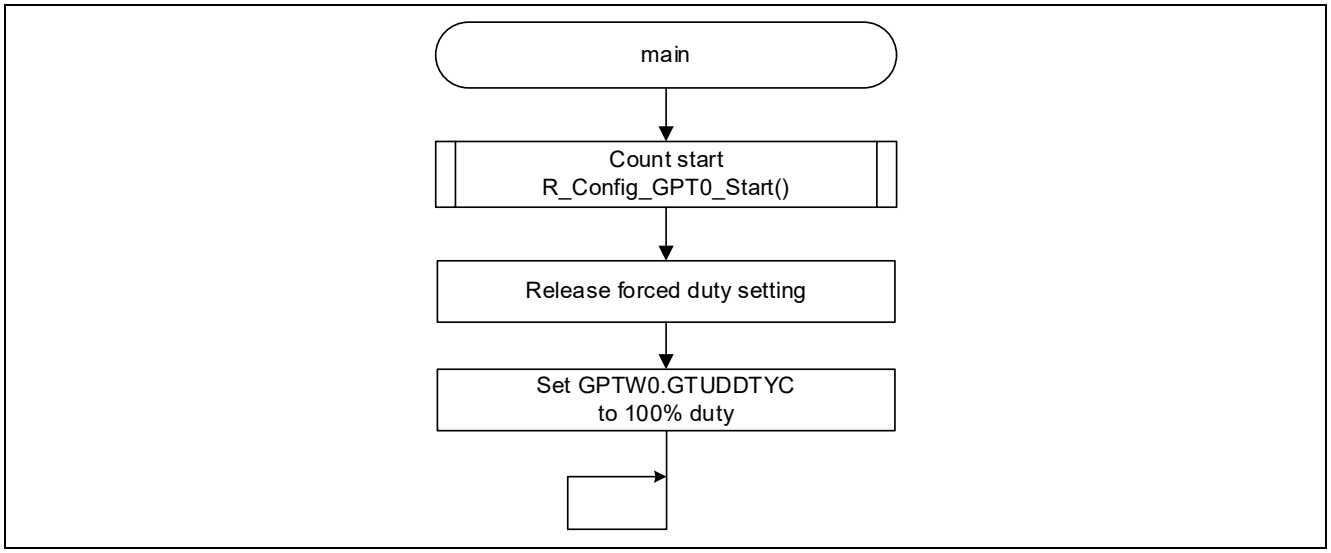


Figure 4.100 main Function

The user initialization function R_Config_GPT0_Create_UserInit, which is executed before the main function, sets forced output duty at counting starts and output duty cycle to 0%. This function is called from within the R_Config_GPT0_Create function.

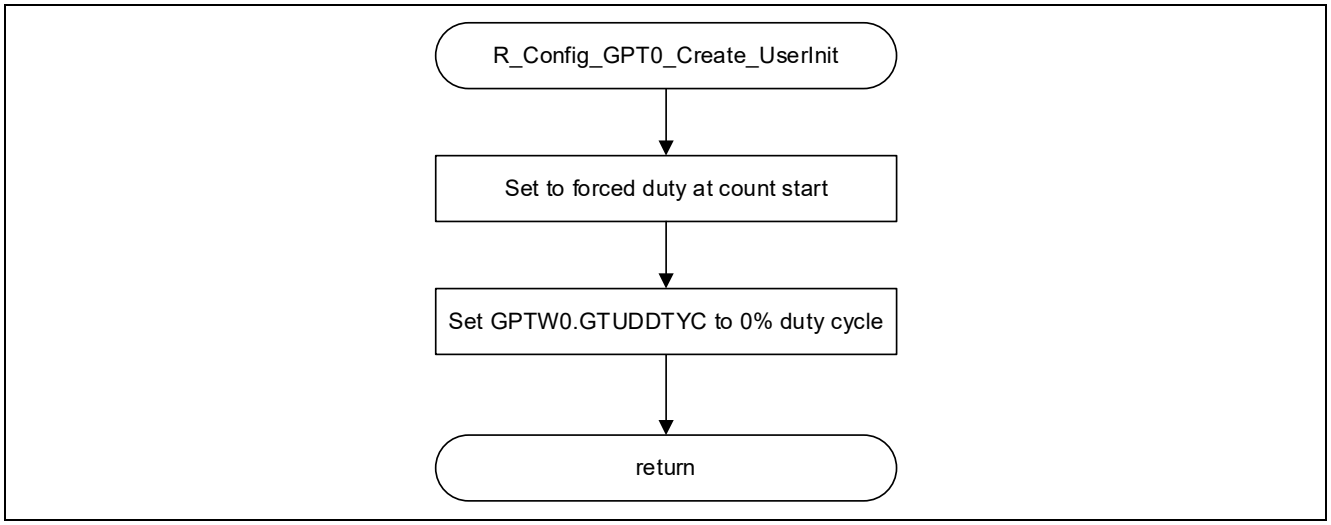


Figure 4.101 User Initialization Function

The GTCIV0 interrupt handler function changes the value of the GTUDDTYC register.

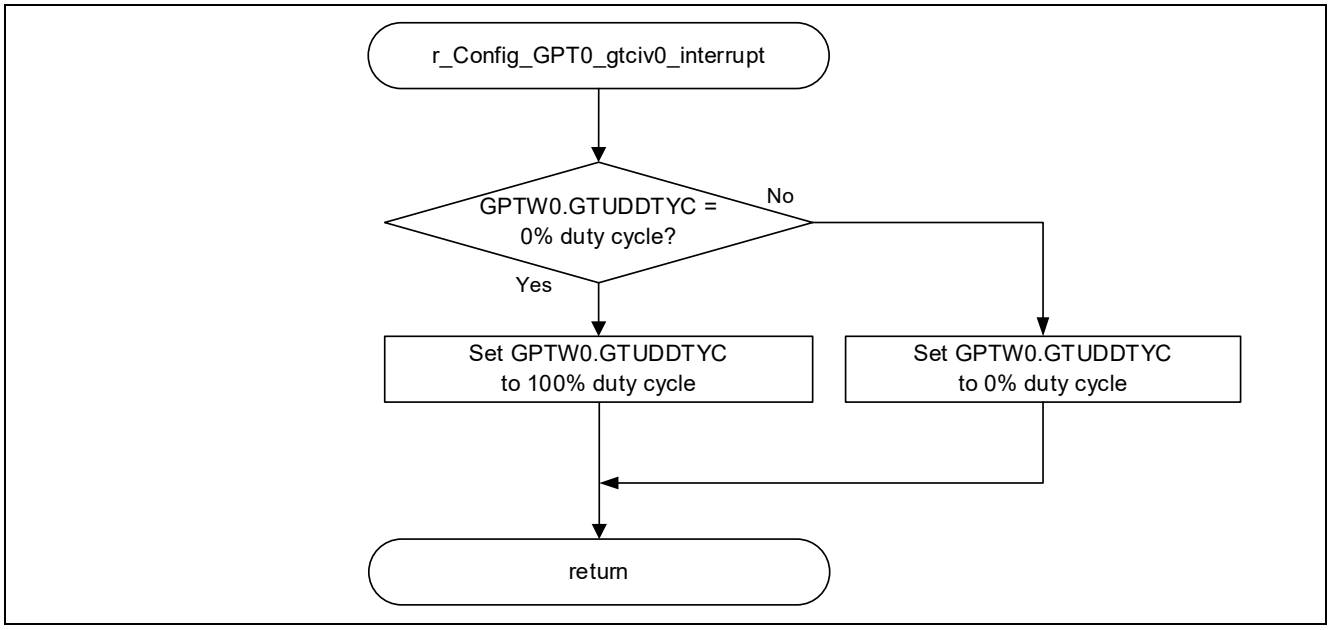


Figure 4.102 GTCIV0 Interrupt Handler Function

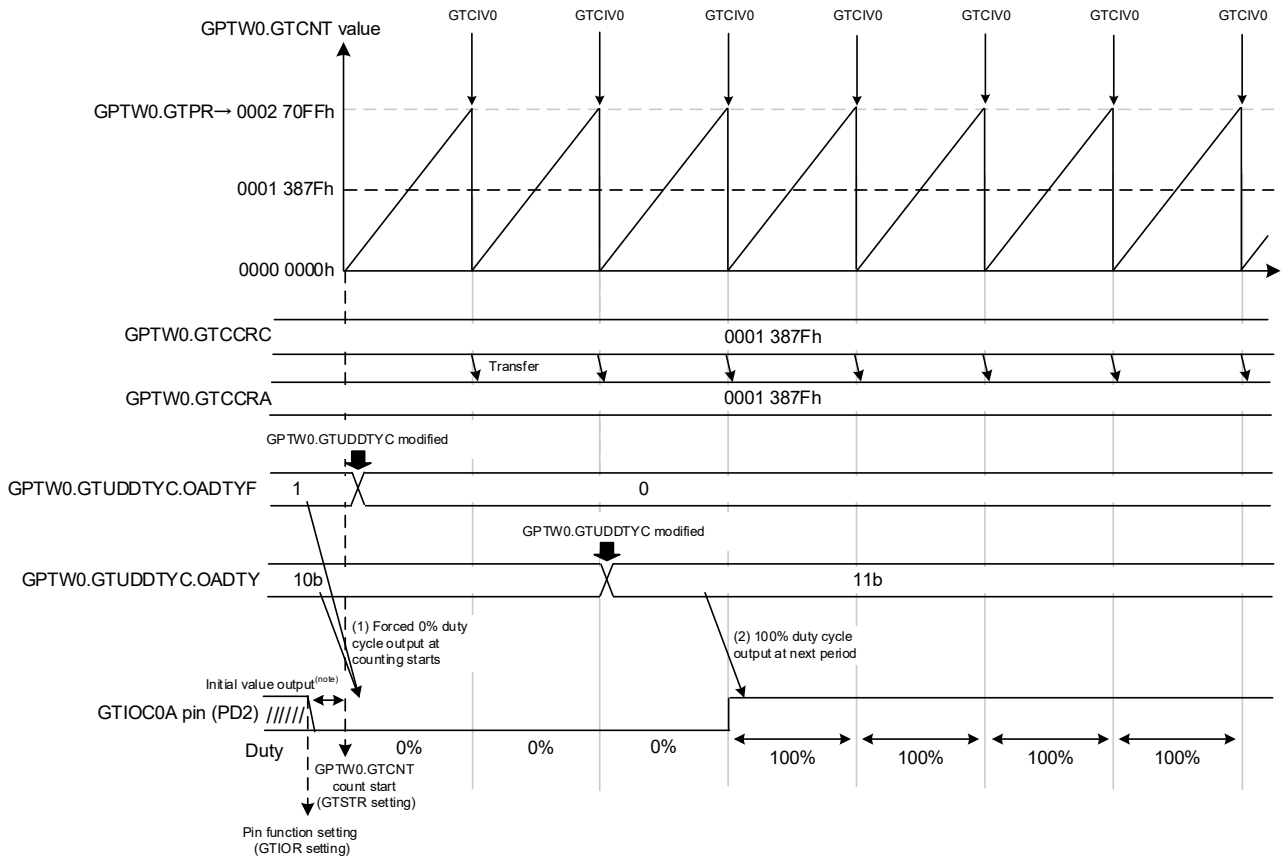
4.11.5 Related Operations

4.11.5.1 Change Duty Cycle to 100% After Holding 0% at Initial Value Low

The following shows an example of operations when the duty cycle is switched to 100% after the counting has started and 0% is output for several cycles.

Low is immediately output after counting starts and output duty switches to 0% by setting the GTUDDTYC.OADTY bits to 10b while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped ((1) in figure below).

During the count operation, the output goes to high and duty cycle goes to 100% in the next cycle by setting the GTUDDTYC.OADTY bits to 11b ((2) in figure below).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

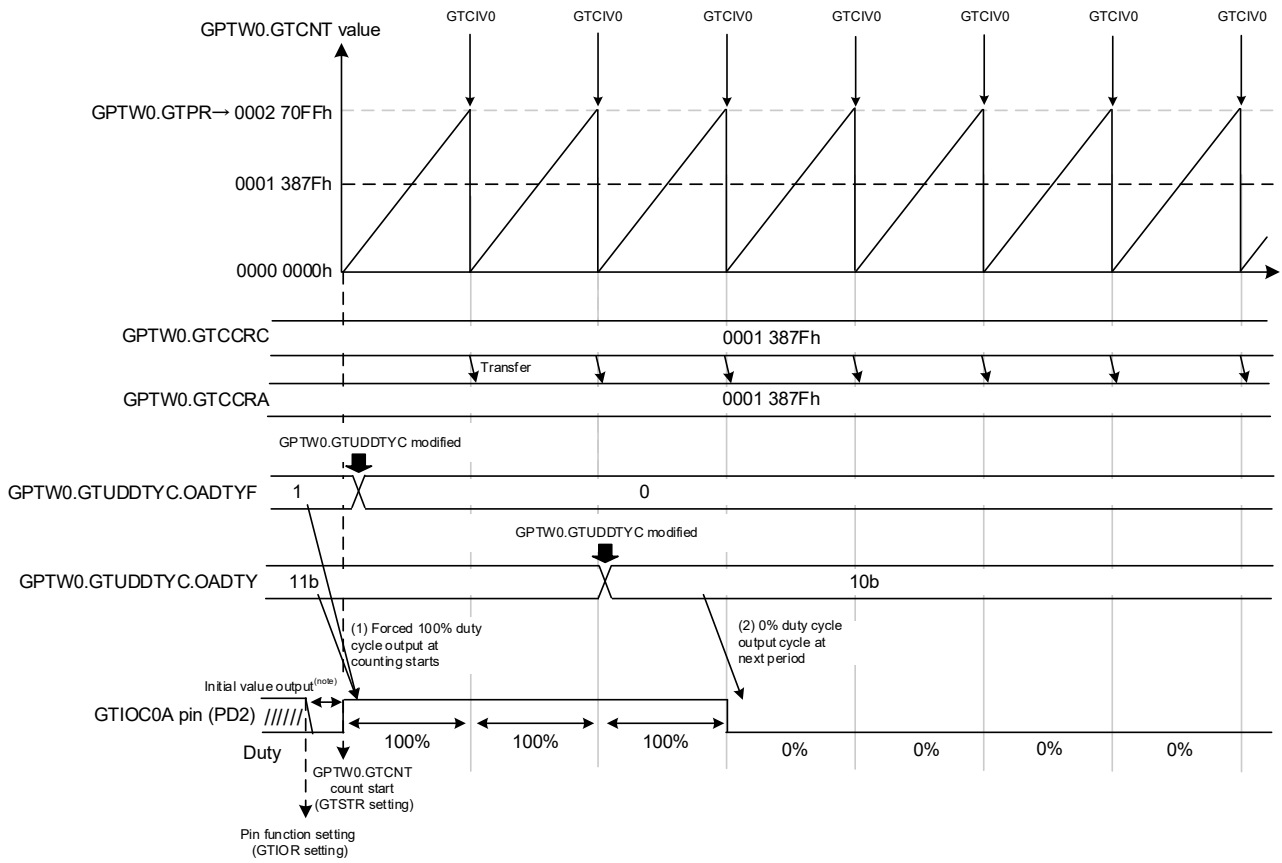
Figure 4.103 Change Duty Cycle to 100% After Holding 0%

4.11.5.2 Change Duty Cycle to 0% After Holding 100% at Initial Value Low

The following shows an example of operations when the duty cycle is switched to 0% after the counting has started and 100% is output for several cycles.

High is immediately output after counting starts and output duty switches to 100% by setting the GTUDDTYC.OADTY bits to 11b while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped ((1) in figure below).

During the count operation, the output goes to low and duty cycle goes to 0% in the next cycle by setting the GTUDDTYC.OADTY bits to 10b ((2) in figure below).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

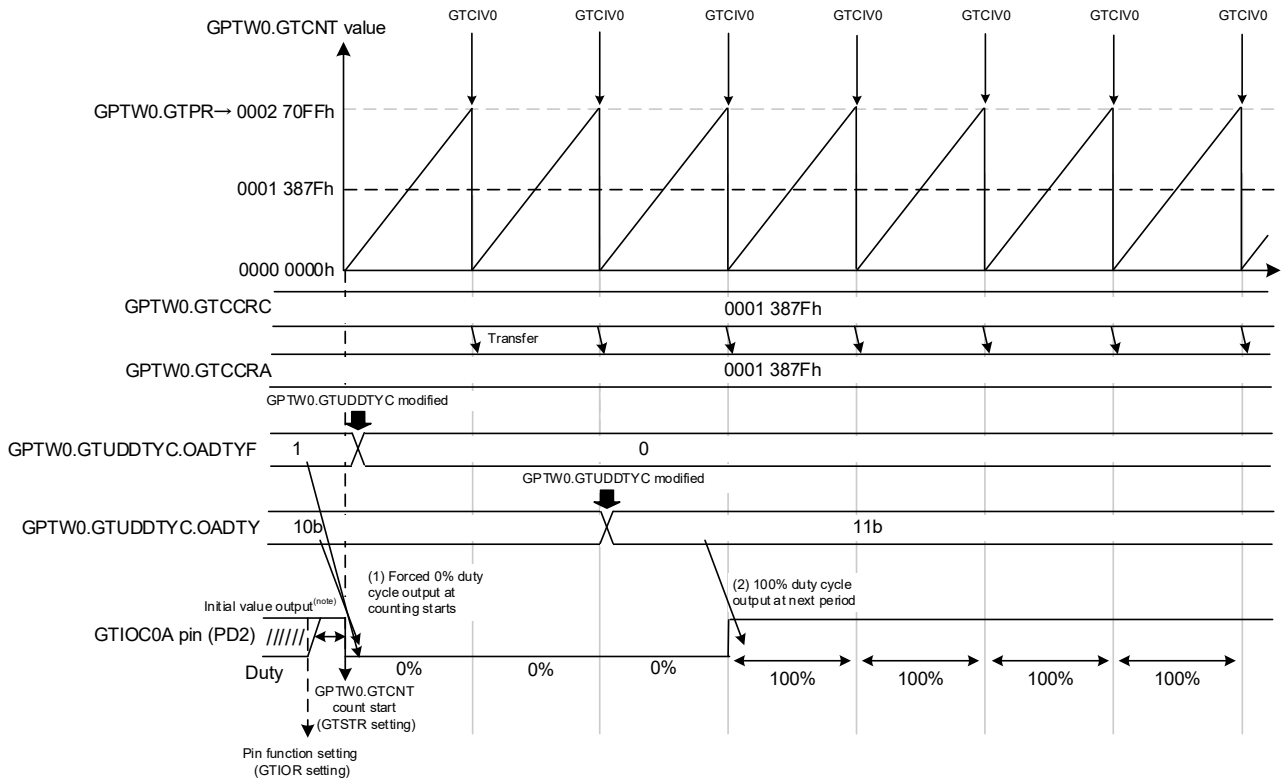
Figure 4.104 Change Duty Cycle to 0% After Holding 100%

4.11.5.3 Change Duty Cycle to 100% After Holding 0% at Initial Value High

The following shows an example of operations when the duty cycle is switched to 100% after counting has started and 0% is output for several cycles.

Low is immediately output after counting starts and output duty switches to 0% by setting the GTUDDTYC.OADTY bits to 10b while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped ((1) in figure below).

During the count operation, output goes to high and duty cycle goes to 100% in the next cycle by setting the GTUDDTYC.OADTY bits to 11b ((2) in figure below).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.105 Change Duty Cycle to 100% After Holding 0%

To set the initial value to high, set “Output at start/stop” to “stop output 1.”

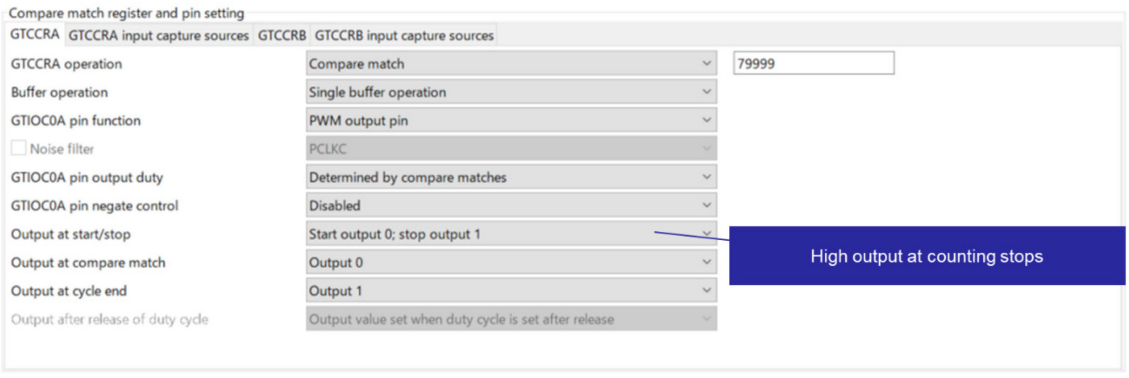


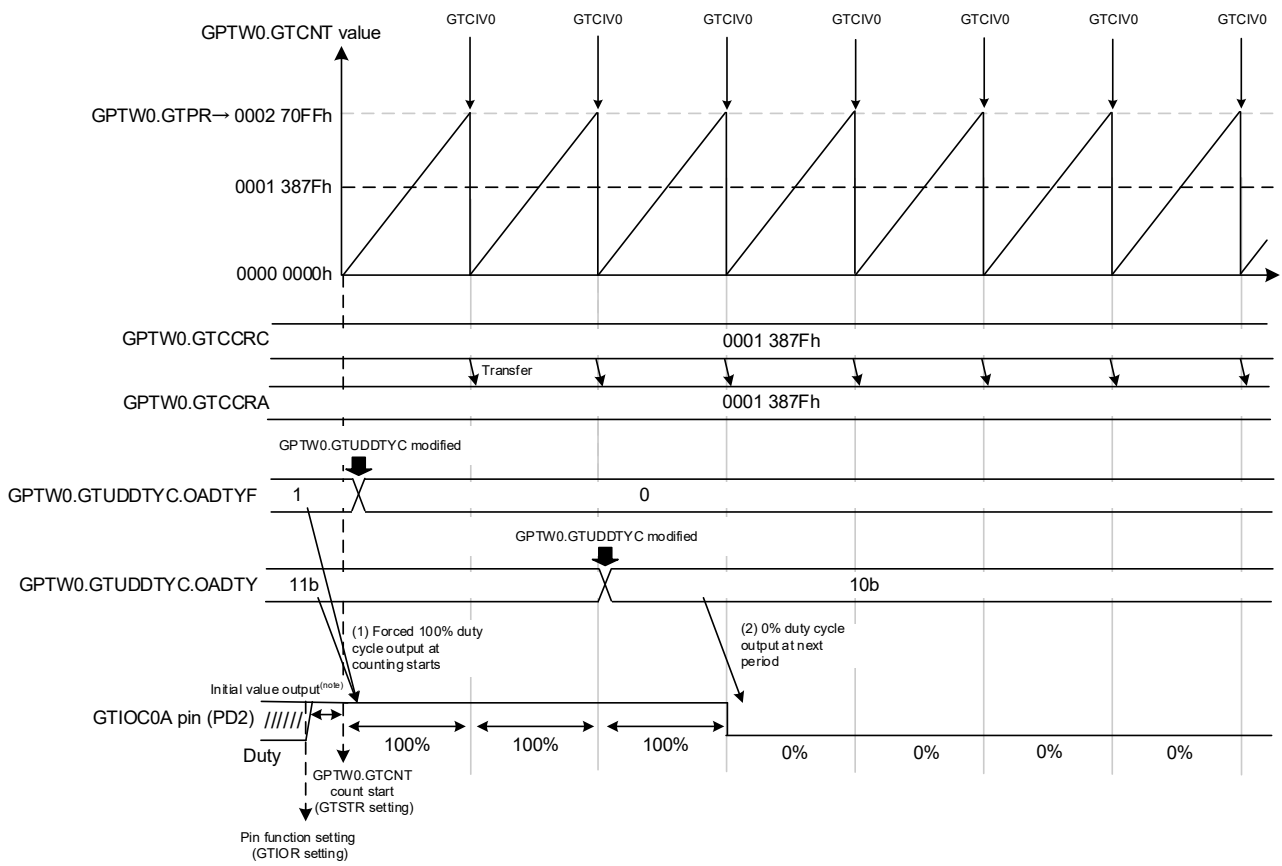
Figure 4.106 Smart Configurator Setting for Initial Value High

4.11.5.4 Change Duty Cycle to 0% After Holding 100% at Initial Value High

The following shows an example of operations when the duty cycle is switched to 0% after the counting has started and 100% is output for several cycles.

High is immediately output after the counting starts and output duty switches to 100% by setting the GTUDDTYC.OADTY bits to 11b while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped ((1) in figure below).

During count operation, the output goes to low and duty cycle goes to 0% in the next cycle by setting the GTUDDTYC.OADTY bits to 10b ((2) in figure below).



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.107 Change Duty Cycle to 0% After Holding 100%

To set the initial value to high, set “Output at start/stop” to “start output 1.” Refer to Figure 4.106 for details.

4.11.5.5 Change Duty Cycle to 100% After Holding at 50%

The following shows an example of operations when the GTUDDTYC.OADTY bit is changed and duty cycle is switched to 100% after 50% has been output for several cycles.

During count operation, 100% duty cycle is output from the next cycle by setting the GTUDDTYC.OADTY bits to 11b ((1) in figure below).

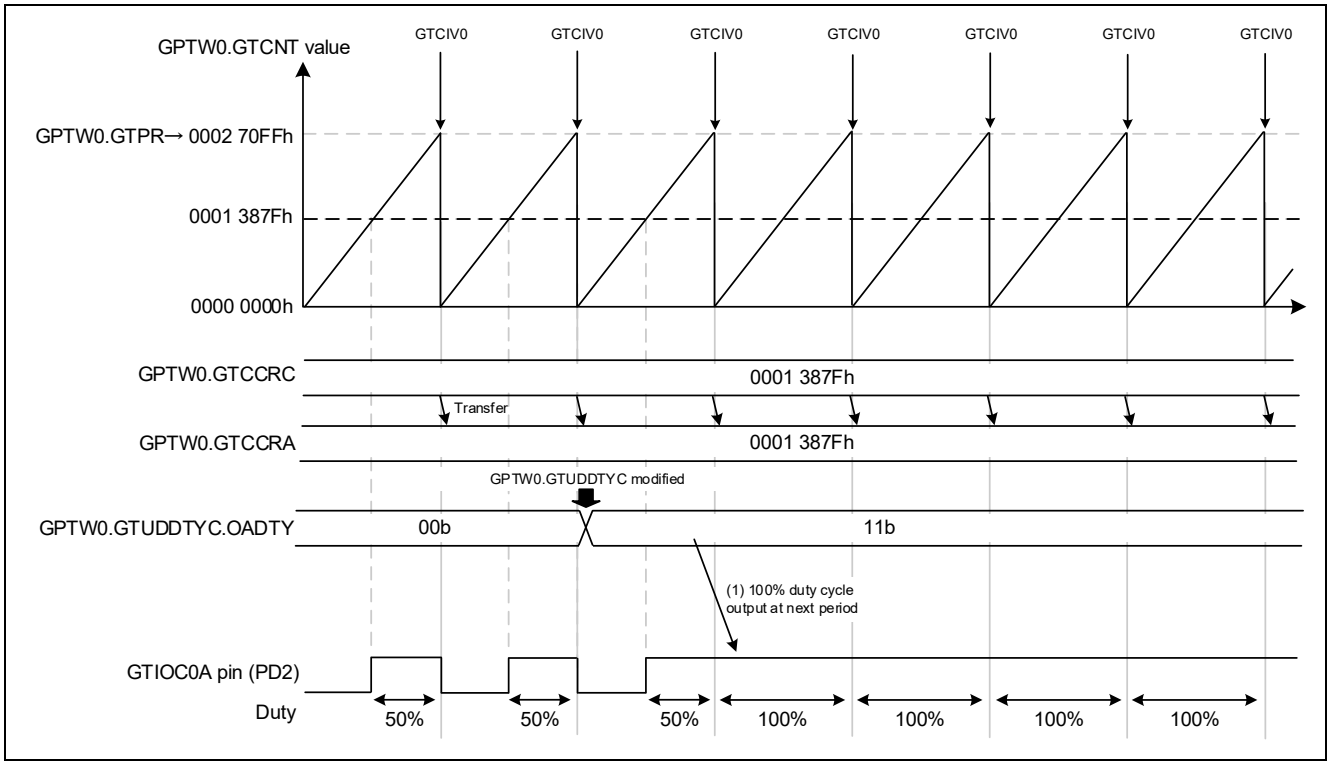


Figure 4.108 Change Duty Cycle to 100% After Holding 50%

Duty cycle cannot be switched to 100% by compare match without changing the GTUDDTYC.OADTY bits during count operation.

4.11.5.6 Change Duty Cycle to 0% After Holding at 50%

The following shows an example of operations when the GTUDDTYC.OADTY bit is changed and duty cycle is switched to 0% after 50% has been output for several cycles.

During count operation, 0% duty cycle is output from the next cycle by setting the GTUDDTYC.OADTY bits to 10b ((1) in figure below).

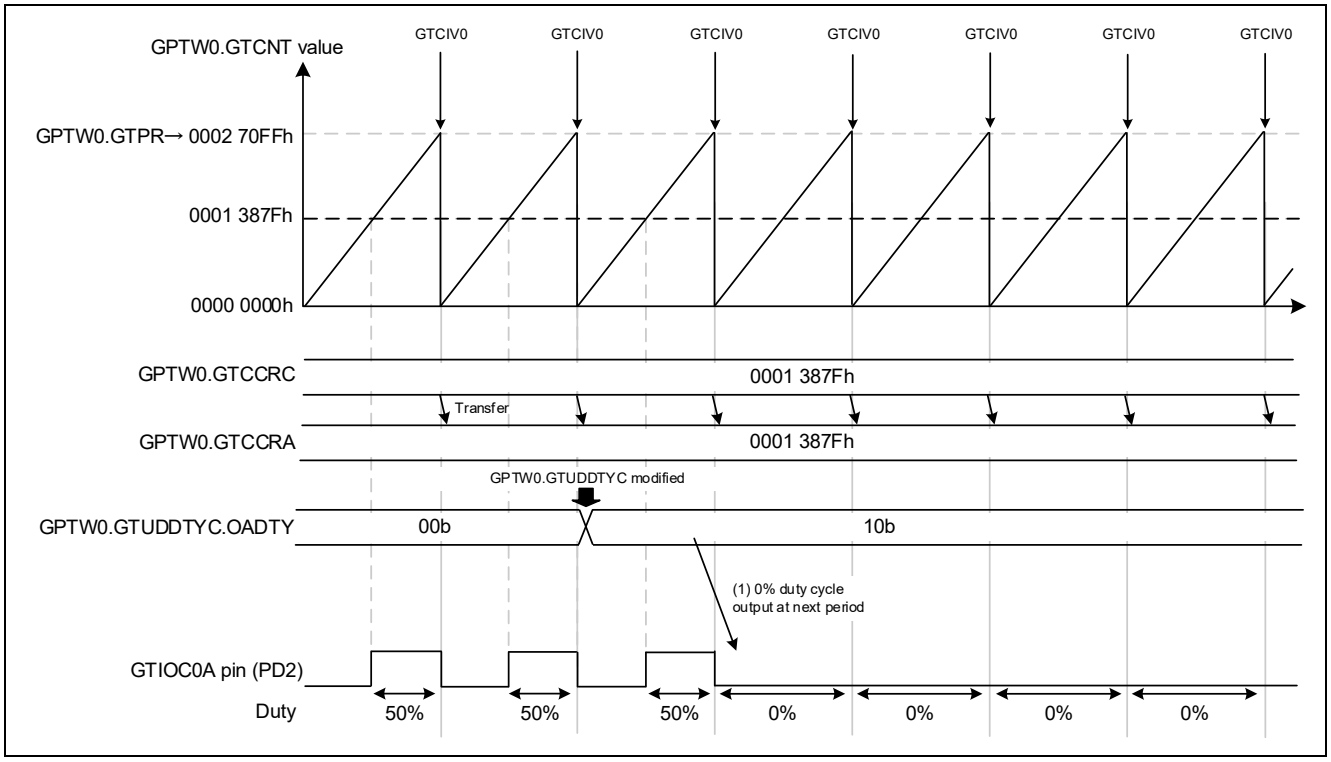


Figure 4.109 Change Duty Cycle to 0% After Holding 50%

4.11.6 Usage Notes

4.11.6.1 Settings of the GTCCRm Register during Compare Match Operation (m = A to F)

Set compare registers GTCCRA and GTCCRB to a value higher than 0000 0001h but less than the setting value of the GTPR register. If set to 0000 0000h or the same value as the GTPR register, a compare match occurs within the cycle only when the compare match register value is 0000 0000h or the compare register is set to the same value as the GTPR register. If the compare register is set to a value that exceeds the setting value of to the GTPR register, no compare match occurs.

The following is an example of operations when a value exceeding the setting value of the GTPR register is set in the compare register. Because a compare match does not occur, the GTIOC0A pin retains low output ((1) in figure below).

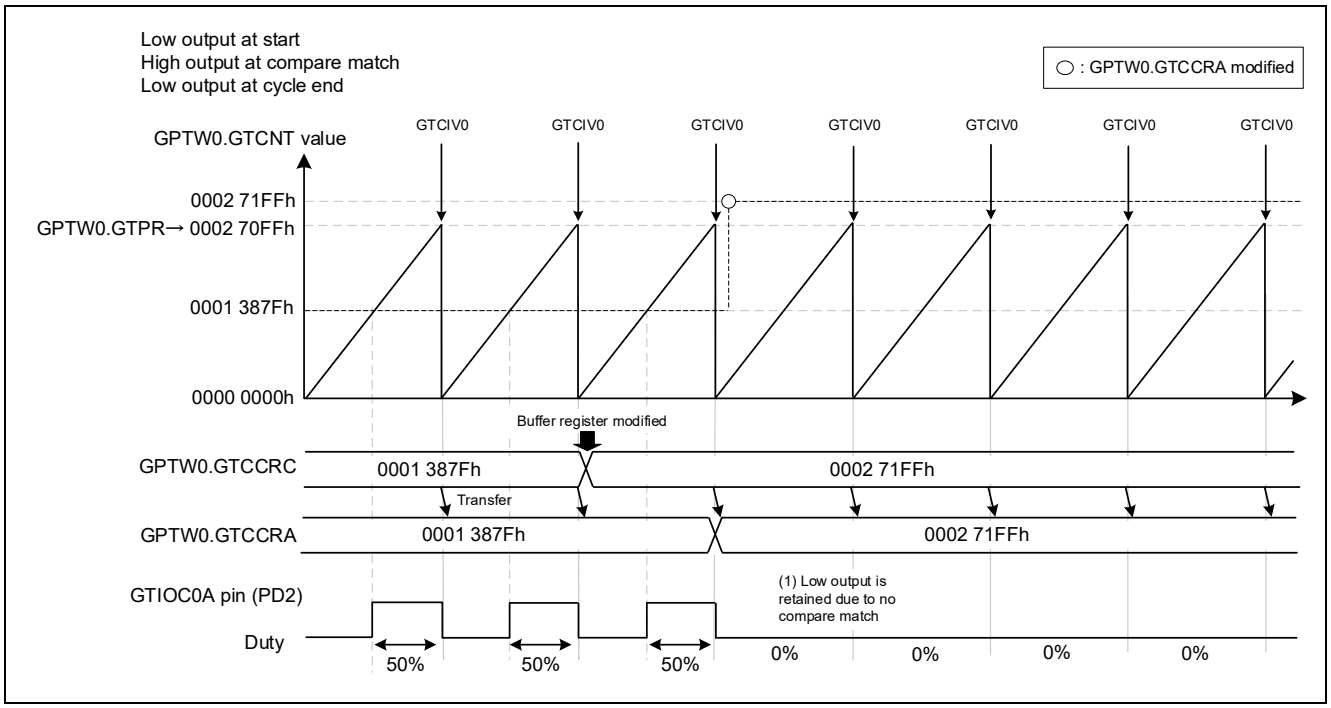


Figure 4.110 Operations when Setting Exceeds GTPR Register Setting Value

For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F), (5) In Sawtooth-wave PWM Mode.

4.11.6.2 Reflection of GTUDDTYC.OADTY Setting at Counting Starts

In this sample code, the count operation duty is determined by the GTUDDTYC.OADTYF and GTUDDTYC.OADTY bits set while the counting is stopped.

If the value of the GTUDDTYC.OADTY bit is changed while the GTUDDTYC.OADTYF bit is 0b and the counting is stopped, the output duty setting changed at the start of counting is not reflected. To reflect the setting from when the counting starts, it is necessary to change the value of the GTUDDTYC.OADTY bit while the GTUDDTYC.OADTYF bit is 1b and counting is stopped, and then counting is started.

For details, refer to RX66T Group User's Manual Hardware, section 24.3.6 Duty Cycle 0%/100% Output Function.

4.11.6.3 Reflection of Duty Cycle at Counting Starts Using Smart Configurator

When the Smart Configurator is used and pin output duty cycle is set at 0% or 100% Figure 4.111, the duty is not reflected at the start of counting because the GTUDDTYC.OADTYF bit is 0b. To reflect the duty from the start of counting, the user must create a code to set the GTUDDTYC.OADTY bits to 10b or 11b and while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped.

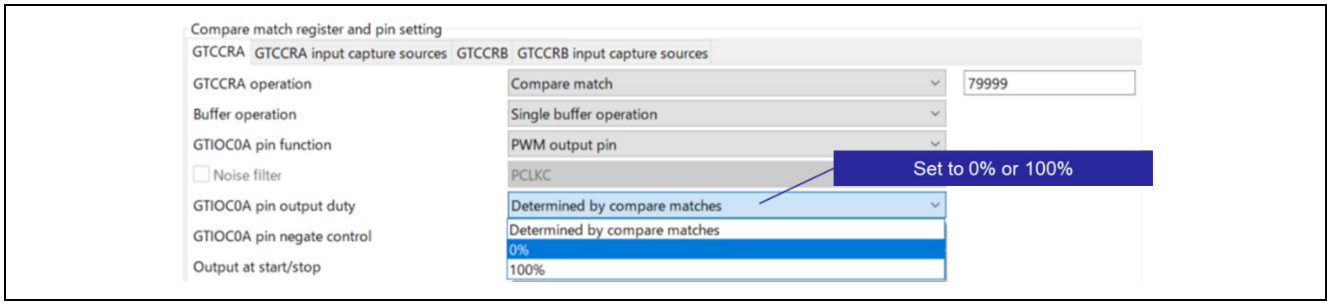


Figure 4.111 Set Pin Output Duty Cycle to 0% or 100%

For details on Smart Configurator settings, refer to section 4.11.3 Smart Configuration Settings, and for an example of user generated code, refer to section 4.11.4 Flowcharts.

4.11.6.4 100% Duty Cycle Output at Compare Match

Output duty cycle cannot be switched to 100% at compare match without changing the GTUDDTYC register. To output 100% duty, set the GTUDDTYC.OADTY bits to 11b.

If the GTCCRA register is set to 0 in this sample code settings, 100% duty cycle cannot be output because after a GTCNT counter overflow occurs, low is output for one clock cycle followed by high output.

The MTU can output 100% duty cycle because the duty register and period register are set to the same value and the waveform does not change when counter clear and compare match occur at the same time. The GPTW cannot output 100% duty cycle in the same way because the waveform changes even if the GTCNT counter overflow and compare match occur at the same time.

For details on waveform output when GTCNT counter overflow and compare match occur at the same time in the GPTW, refer to the notes following Table 24.4 in RX66T Group User's Manual: Hardware, section 24.2.14 General PWM Timer I/O Control Register (GTIOR).

4.11.6.5 Compare Match Operation during Duty Cycles 0% and 100% Output

In this sample code, output duty cycles 0% and 100% are determined based on the values set to the GTUDDTYC.OADTY bits. When duty cycle is set to either 0% or 100%, the compare match operation continues in the GPTW and interrupt output and buffer transfer operation are performed.

This sample code does not use the compare match interrupt, so if you are using the compare match interrupt, do so with caution interrupt during duty cycle 0% and 100% output.

4.12 Triangle-Wave PWM Mode 1 Duty Cycles 0% to 100%

- Target sample code file name: r01an5995_rx66t_gptw_triangle_pwm_50to100_dt.zip

4.12.1 Overview

The GPTW triangle-wave PWM mode 1 can be used to output PWM waveforms of duty cycles 0% to 100% according to the GTCCRA register compare match and the GTUDDTYC register setting.

This sample code describes a sample code that that uses the automatic dead time setting function in triangle-wave PWM mode 1 and repeats the following output waveforms, including duty cycles 0% and 100%.

- GTIOC0A pin high-width switching:50% → 80% → 100% → 80% → 50% → 0% → ・・・
- GTIOC0B pin low-width switching: 60% → 90% → 100% → 90% → 60% → 0% → ・・・

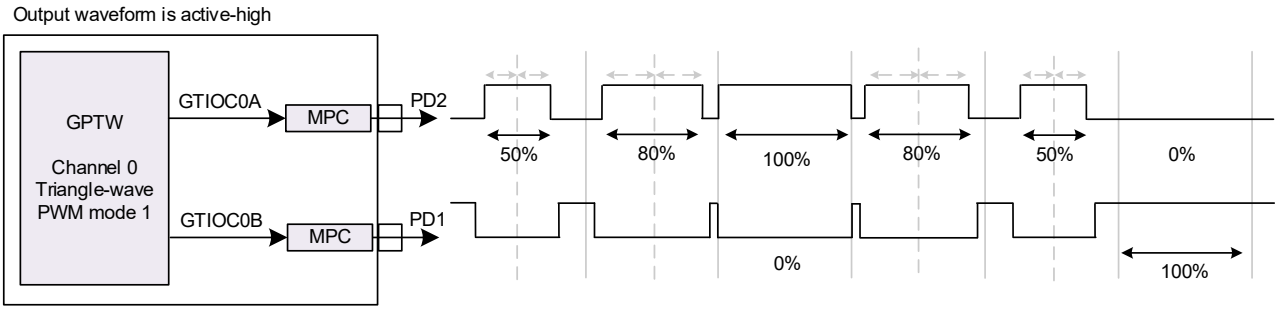
The basic operation is to make changes to the duty cycle by transferring the value of the buffer register to the GTCCRA when a GTCNT counter underflow occurs using buffer register GTCCRC. When switching between duty cycles 0% and 100%, the process to modify the GTUDDTYC register is performed when a GTCNT counter underflow occurs.

The following list provides the GPTW settings used in the sample code.

- Use triangle-wave PWM mode 1
- Use channel 0
- Carrier period = 1ms
- Timer count clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Counter up-counts from initial value 0
- Use GTCCRA as duty output compare match
  - Use GTIOC0A pin as PWM output pin
  - Use GTCCRA as compare match
  - Low output at counting starts
  - Toggle output at GTCCRA compare match
  - Retain output at cycle end
  - After releasing duty cycle 0%/100%, output the compare match output value that was masked*
- Use GTCCRB as duty output compare match
  - Use GTIOC0B pin as PWM output pin
  - Use GTCCRB as compare match
  - High output at counting starts
  - Toggle output at GTCCRB compare match
  - Retain output at cycle end
  - After releasing duty cycle 0%/100%, output the compare match output value that was masked*
- Use buffer register
  - GTCCRA operates as single buffer
  - Use GTCCRC as buffer register of GTCCRA
- Use automatic dead time generation
- Software source count start enabled
- Duty changes at each cycle
  - Duty changes at the GTCNT counter underflow interrupt
  - Refer to Figure 4.114 for details on duty change timing

Set in Smart Configurator.
For Setting Methods,
refer to section 4.12.3
(except for items marked with *).

The following shows triangle-wave PWM mode 1 output for the sample code.



GPTW: Generates PWM waveform with triangle-wave PWM mode 1
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

Figure 4.112 Triangle-wave PWM Mode 1 Output

4.12.2 Operation Details

The sample code operations are shown in Figure 4.114. The basic operation is to make changes to the settings of the duty cycle with each period by modifying the value of buffer register GTCCRC at the GTCNT counter underflow interrupt (GTCIU0) and transferring the value of the buffer register to the GRCCRA when a GTCNT counter underflow occurs.

In addition, the GTCCRB register is automatically set according to the GTCCRA update because the automatic dead time function is used. The same values are set for the GTDVU and GTDVD.

The duty cycle in each period generates the 1/2 duty cycle for the up-counting period and down-counting period. The duty cycles generated in the up-counting and down-counting periods are shown in Figure 4.113.

- 100% Duty Cycle Output ((1) in Figure 4.114)
  The GTIOC0A pin outputs high from the time of the next GTCNT counter underflow occurs (trough) by setting the GTUDDTYC.OADTY bits to 11b, and the waveform does not change even if a GTCCRA compare match occurs.
  The GTIOC0B pin outputs low from the time of the next GTCNT counter underflow (trough) occurs by setting the GTUDDTYC.OBDTY bits to 10b, and the waveform does not change even if a GTCCRB compare match occurs. In this sample code, the bits are set to 10b so that the GTIOC0B pin outputs low for one cycle period.

- 0% Duty Cycle Output ((2) in Figure 4.114)
  The GTIOC0A pin outputs low from the time of the next GTCNT counter underflow occurs (trough) by setting the GTUDDTYC.OADTY bits to 10b. The waveform does not change even if a GTCCRA compare match occurs.
  The GTIOC0B pin outputs high (trough) from the time of the next GTCNT counter underflow occurs by setting the GTUDDTYC.OBDTY bits to 11b, and the waveform does not change even if a GTCCRB compare match occurs. In this sample code, the bits are set to 11b so that the GTIOC0B pin outputs high for one cycle period.

- Switching from Duty Cycles 100% and 0% ((3) in (Figure 4.114)
  The GTIOC0A pin outputs duty cycle according to the GTCCRA compare match from the time of the next GTCNT counter underflow (trough) occurs by setting the GTUDDTYC.OADTY bits to 00b. After duty cycles 100% and 0% are released, the output is determined by the GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR bits. In this sample code, the masked compare match output value low which was masked is output.
  The GTIOC0B pin outputs duty cycle according to the GTCCRB compare match from the time of the next GTCNT counter underflow occurs by setting the GTUDDTYC.OBDTY bits to 00b. After duty cycles 100% and 0% are released, the output is determined by the GTIOR.GTIOB[3:2] and GTUDDTYC.OBDTYR bits. In this sample code, the masked compare match output value high which was masked is output.
  Figure 4.113 shows an example of the GTIOC0A pin output after duty cycles 100% and 0% are released.

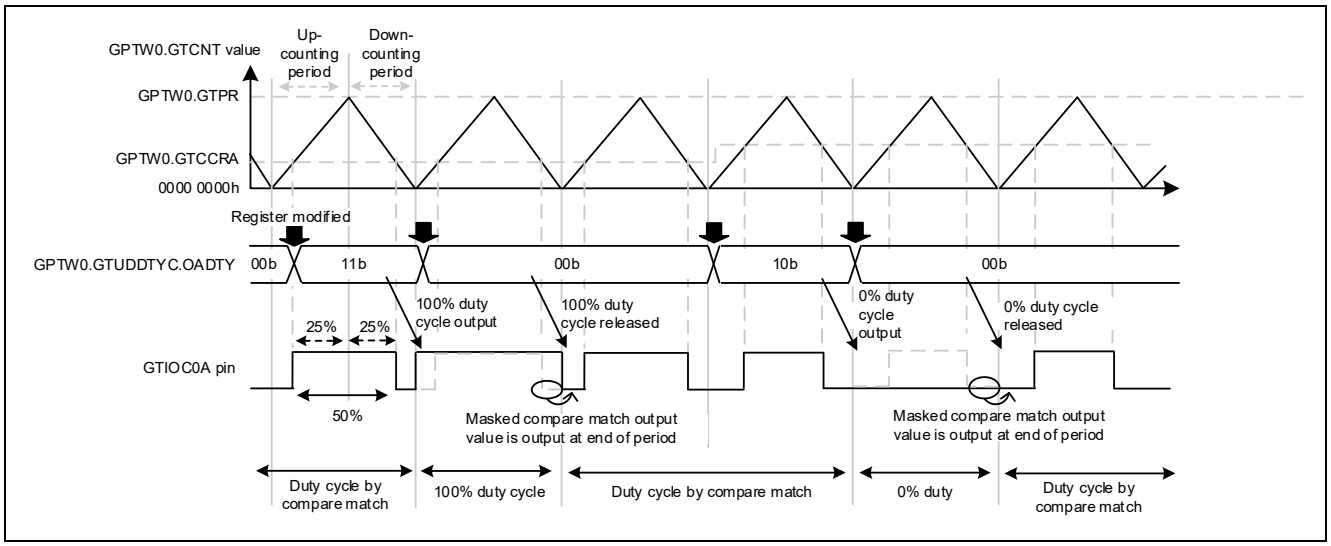
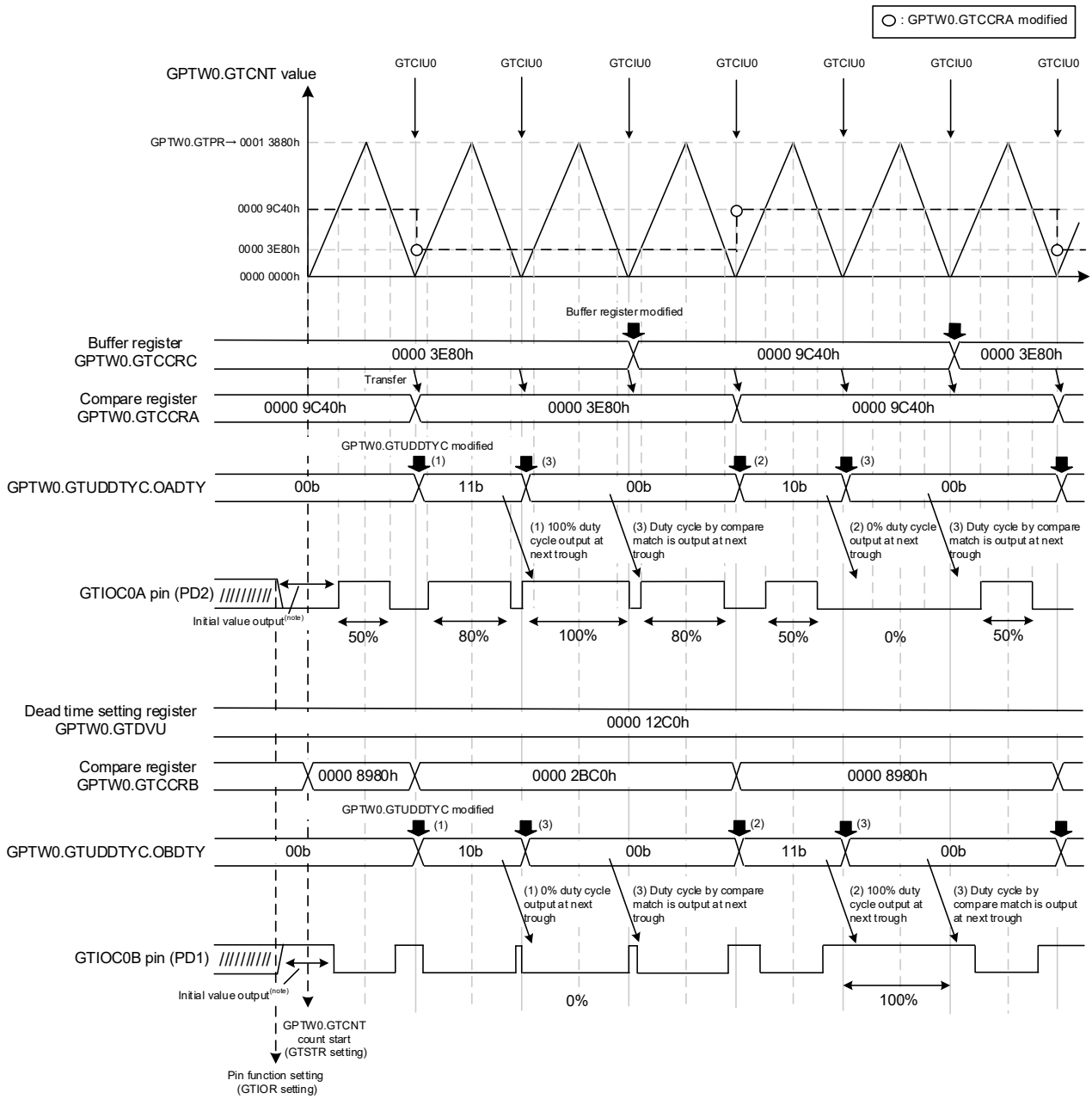


Figure 4.113 1/2 Duty Generation and Output After Release of Duty Cycles 100% and 0%



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.114 Sample Code Operations

4.12.3 Smart Configurator Settings

The sample codes use the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.15 Adding Components

Item	Description
Component	General PWM Timer
Configuration Name	Config_GPT0
Work mode	Triangle PWM Mode 1
Resource	GPT0

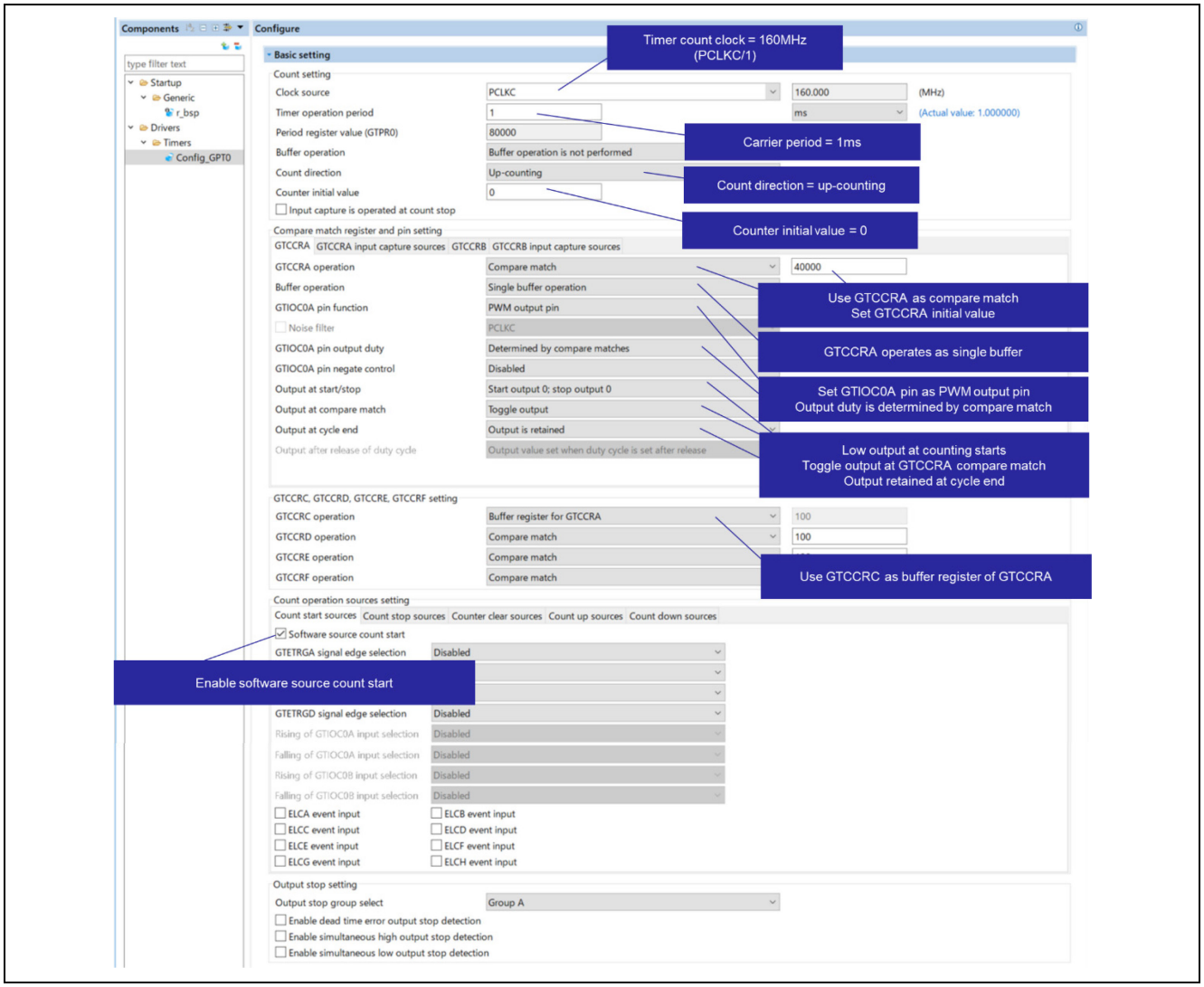


Figure 4.115 GPT0 Settings (1/2)

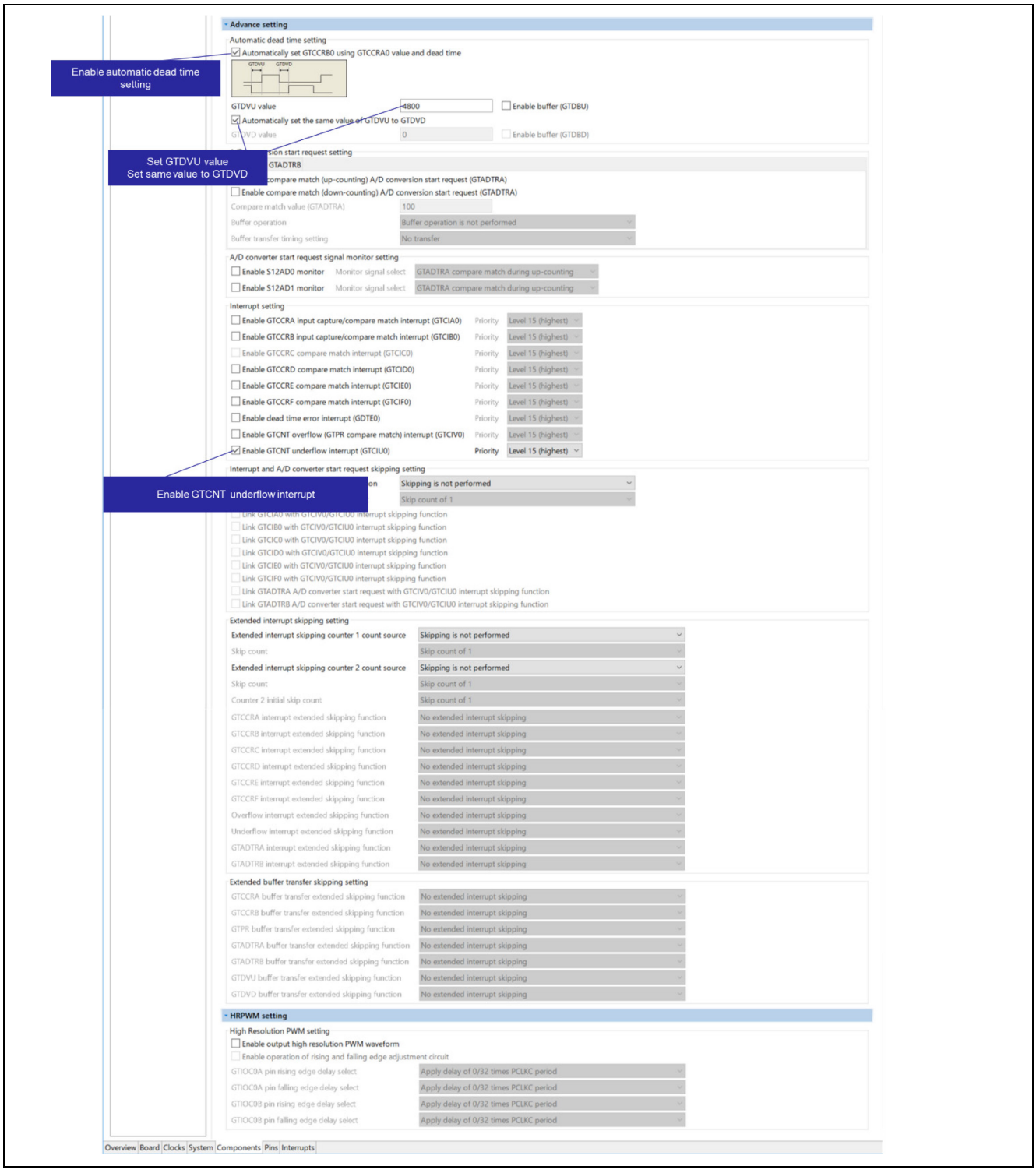


Figure 4.116 GPT0 Settings (2/2)

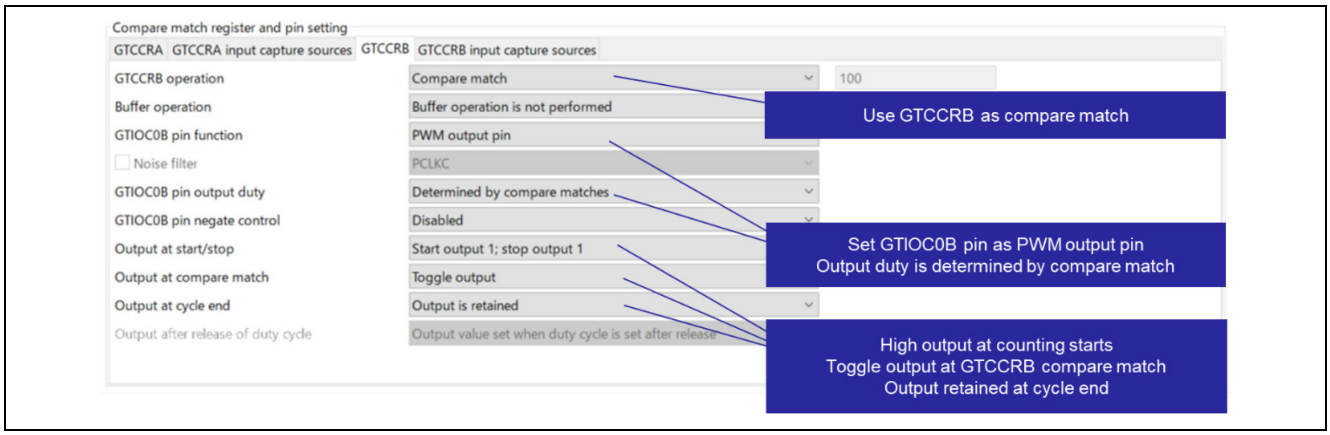


Figure 4.117 GPT0 Settings (Compare Match Register and Pins Setting of GRCCRB)

4.12.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

Counting is started in the main function.

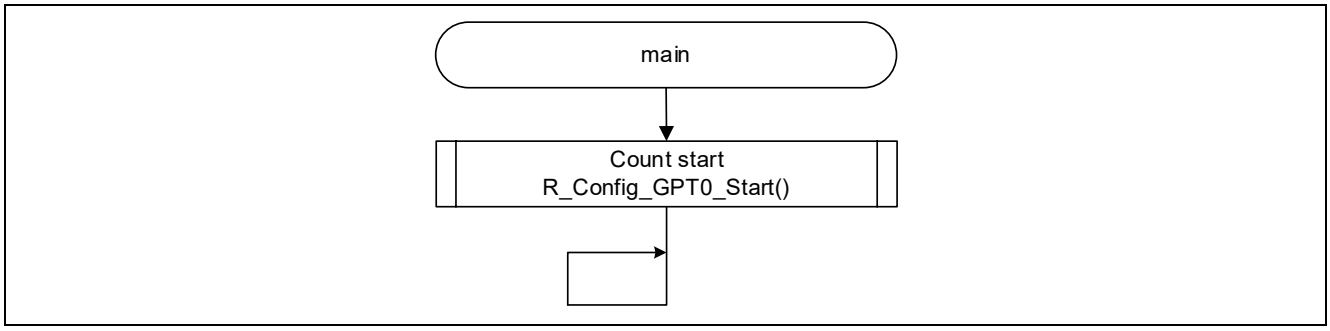


Figure 4.118 main Function

The user initialization function R_Config_GPT0_Create_UserInit, which is executed before the main function, initializes the variables, sets the initial values of the buffer register, and sets output values for after the release of duty cycles 100% and 0%. This function is called from within the R_Config_GPT0_Create function.

This sample code uses the following variable.

- s_duty_list_counter: counter variable for reading from the duty cycle list

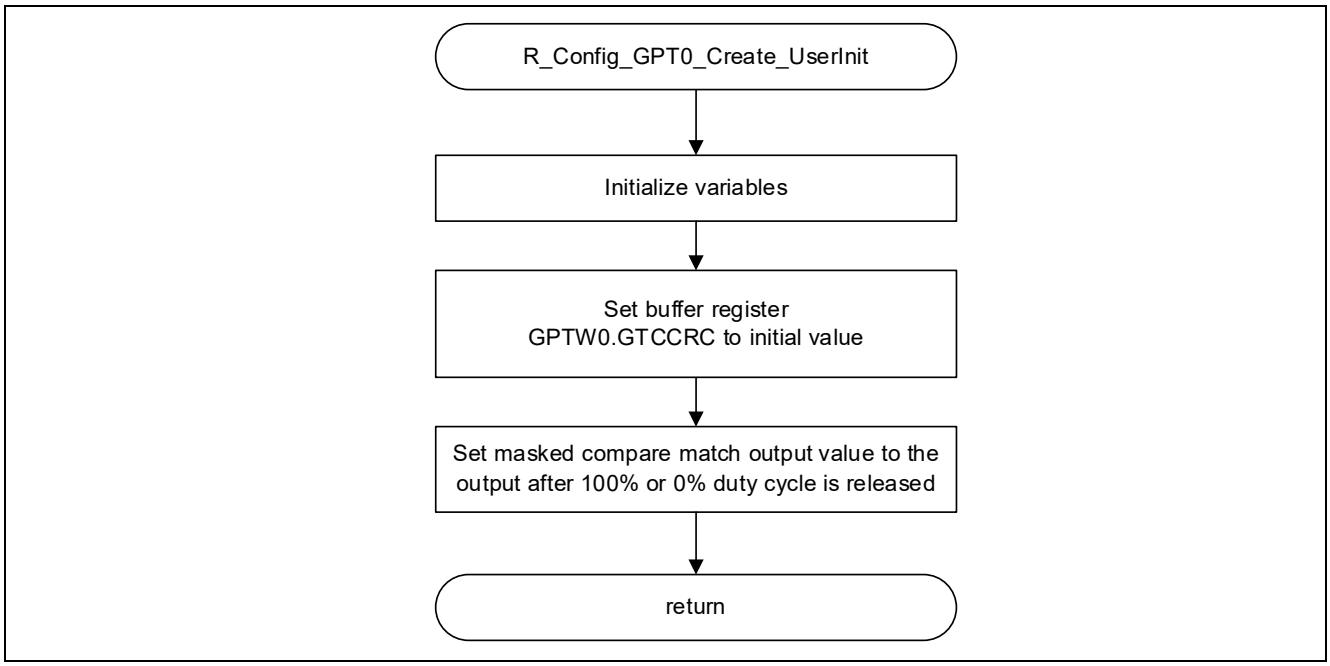


Figure 4.119 User Initialization Function

The GTCIU0 interrupt handler function changes the values of the buffer register and the GTUDDTYC register according to the next duty cycle to be set.

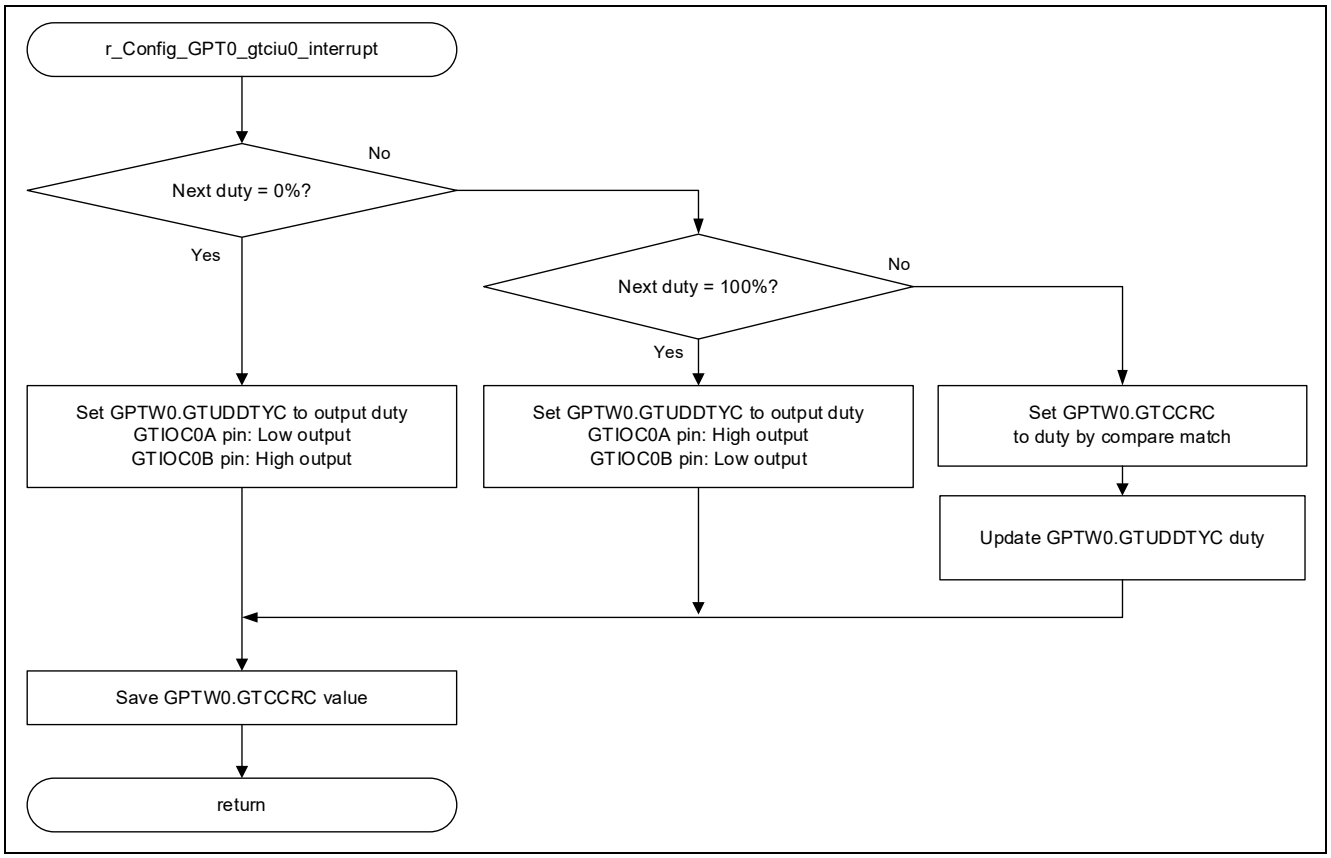


Figure 4.120 GTCIU0 Interrupt Handler Function

4.12.5 Usage Notes

4.12.5.1 Settings of GTCCRm Register during Compare Match Operation (m = A to F)

This sample code automatically sets the dead time and uses the value of compare match register GTCCRA, the register for positive-phase waveform, to update the value of compare match register GTCCRB, the register for negative-phase waveform.

The values of compare register GTCCRA should be set to satisfy the following restrictions.

GTCCRA > GTDVU
GTCCRA > GTDVD
GTCCRA < GTPR

If the GTCCRA is set to 0000 0000h or a value greater than the setting value of the GTPR during the counting operation, the output protection function is activated

However, if the conditions below are not satisfied, the function does not operate normally.

When the GTCCRA register value at the start of count operation is greater than 0000 0001h and less than the setting value of the GTPR register

For details, refer to RX66T Group User's Manual: Hardware, section 24.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 9; m=A, B).

When not using the automatic dead time setting function, set a value greater than 0000 0001h and less than the setting value of the GTPR register in the GTCCRA (GTCCRB) register. When 0000 0000h or the same value as the GTPR register is set in the GTCCRA register, a compare match occurs in a cycle only when [GTCCRA (GTCCRB) = 0000 0000h] or [GTCCRA (GTCCRB) = GTPR] is met. Furthermore, if a value exceeding the setting value of the GTPR register is set in the GTCCRA, compare match does not occur.

For details, refer to RX66T Group User's Manual: Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F), (1) When Automatic Dead Time Setting has been Made in Triangle-Wave PWM Mode and (2) When Automatic Dead Time Setting has not been Made in Triangle-Wave PWM Mode.

4.12.5.2 100% Duty Cycle Output at Compare Match

Output duty cycle cannot be switched to 100% at compare match without changing the GTUDDTYC register. To output 100% duty, set the GTUDDTYC.OADTY bits to 11b and the GTUDDTYC.OBDTY bits to 10b.

If the GTCCRA and GTCCRB registers are set to 0 in this sample code settings, 100% duty cycle cannot be output because one clock cycle is output after a GTCNT counter underflow occurs.

When the GTCCRA and GTCCRB registers are set to the same value as the GTPR and the GTCNT counter underflow and compare match occur at the same time, the output setting at the timing of the compare match is prioritized and toggled, disabling 100% duty cycle output.

For details regarding the waveform output when the GPTW's GTCNT counter underflow and compare match occur at the same time, refer to the notes under Table 24.4 in RX66T Group User's Manual Hardware, section 24.2.14 General PWM Timer I/O Control Register (GTIOR).

4.12.5.3 Compare Match Operation during Duty Cycles 0% and 100% Output

In this sample code, output duty cycles 0% and 100% are determined based on the values set to the GTUDDTYC.OADTY bits. When duty cycle is set to either 0% or 100%, the compare match operation continues in the GPTW and interrupt output and buffer transfer operation are performed.

This sample code does not use the compare match interrupt, so if you are using the compare match interrupt, do so with caution interrupt during duty cycle 0% and 100% output.

4.12.5.4 Switching from Duty Cycles 0% and 100%

If the output setting is changed by a compare match after duty cycle 0% or 100% is set, the output value at the end of the period is determined by the values of the GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR bits.

The following are the settings for this sample code. Note that if the initial hardware value of the GTUDDTYC.OADTYR bit is 0, the same operation as this sample code cannot be performed. The same applies for the GTIOC0B pin.

- GTIOR.GTIOA[3:2] = 00b: Retain output at cycle end
- GTUDDTYC.OADTYR = 1: After releasing the duty cycle 0%/100% setting, apply the GTIOA [3: 2] bit function to the compare match output value that was masked.

For details, refer to RX66T Group User's Manual: Hardware, section 24.3.6 Duty Cycle 0%/100% Output Function.

4.13 Triangle-Wave PWM Mode Duty Cycles 0% and 100%

- Target sample code file name: r01an5995_rx66t_gptw_triangle_pwm_0to100_dt.zip

4.13.1 Overview

The GPTW triangle-wave PWM mode 1 can be used to output duty cycle 0% to 100% according to compare match of the GTCCRA and GTCCRB registers and setting the GTUDDTYC register.

This sample code describes a sample code that uses the automatic dead time setting function in triangle-wave PWM mode 1 and repeats waveform output alternating between duty cycles 0% and 100%.

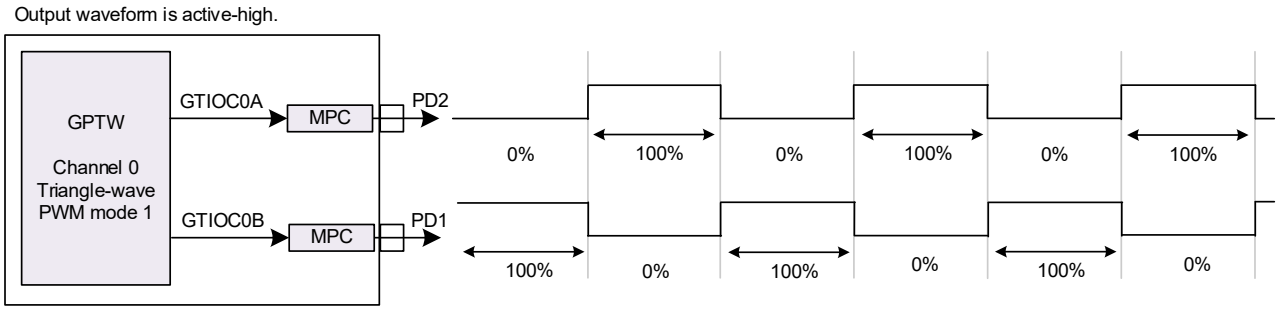
Output switches between duty cycles 0% and 100% by modifying the GTUDDTYC register when a GTCNT counter underflow occurs.

The following list provides the GPTW settings used in the sample code.

- Use triangle-wave PWM mode 1
- Use channel 0
- Carrier period = 1ms
- Timer count clock = 160MHz (PCLKC/1)
- Use GTPR as period register
  - Counter up-counts from initial value 0
- Use GTCCRA as duty cycle output compare match
  - Use GTIOC0A pin as PWM output pin
  - Use GTCCRA as compare match
  - Low output at counting starts
  - Toggled output by GTCCRA compare match
  - Output retained at cycle end
  - Forced output 0% duty cycle at counting starts*
- Use GTCCRB as duty output compare match
  - Use GTIOC0B pin as PWM output pin
  - Use GTCCRB as compare match
  - High output at counting starts
  - Toggled output at GTCCRB compare match
  - Output retained at cycle end
  - Forced output 0% duty cycle at counting start*
- Use buffer registers
  - GTCCRA operates as single buffer
  - Use GTCCRC as buffer register of GTCCRA
- Use automatic dead time generation
- Software source count start enabled
- Duty changes at each cycle
  - Duty changes at the GTCNT counter underflow interrupt
  - Refer to Figure 4.122 for details on duty change timing

Set in Smart Configurator.
For setting method,
refer to section 4.13.3
(except for items marked with *).

The following shows triangle-wave PWM mode 1 output for the sample code.



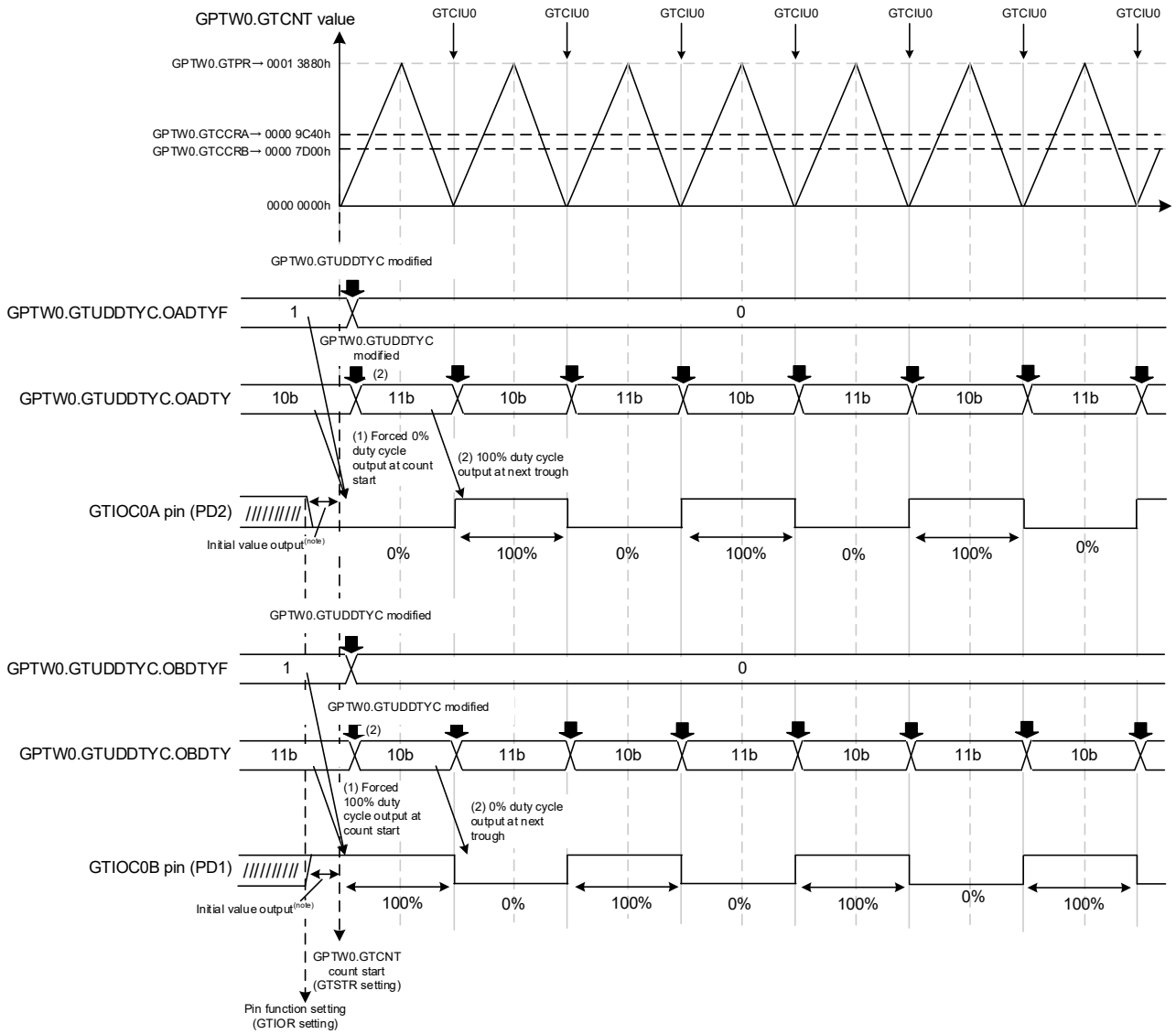
GPTW: Generates PWM waveform with triangle-wave PWM mode 1
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

Figure 4.121 Triangle-wave PWM Mode 1 Output

4.13.2 Operation Details

The sample code operations are shown in Figure 4.122. The duty cycle is switched between 0% and 100% with each period by modifying the value of the GTUDDTY register at the GTCNT counter underflow interrupt (GTCIU0).

- 0% Duty Cycle Output After Counting Starts ((1) in Figure 4.122)
  The GTIOC0A pin outputs low immediately after the counting starts by setting the GTUDDTYC.OADTY bits to 10b while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped. The waveform does not change even if a GTCCRA compare match occurs.
  The GTIOC0B pin will output high immediately after the counting starts by setting the GTUDDTYC.OBDTY bits to 11b while the GTUDDTYC.OBDTYF bit is 1b and the counting is stopped. The waveform does not change even if a GTCCRB compare match occurs. In this sample code, the GTUDDTYC.OBDTY bits are set to 11b for the GTIOC0B pin to output high for one cycle period.
- 100% Duty Cycle Output at 2nd Cycle ((2) in Figure 4.122)
  The GTIOC0A pin will output high from the 2nd cycle by setting the GTUDDTYC.OADTY bits to 11b before the first GTCNT counter underflow occurs after the counting starts. The waveform does not change even if a GTCCRA compare match occurs.
  The GTIOC0B pin will output low from the 2nd cycle by setting the GTUDDTYC.OBDTY bits to 10b before the first GTCNT counter underflow occurs after the counting starts. The waveform does not change even if a GTCCRB compare match occurs. In this sample code, the GTUDDTYC.OBDTY bits are set to 10b for the GTIOC0B pin to output low for one cycle period.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

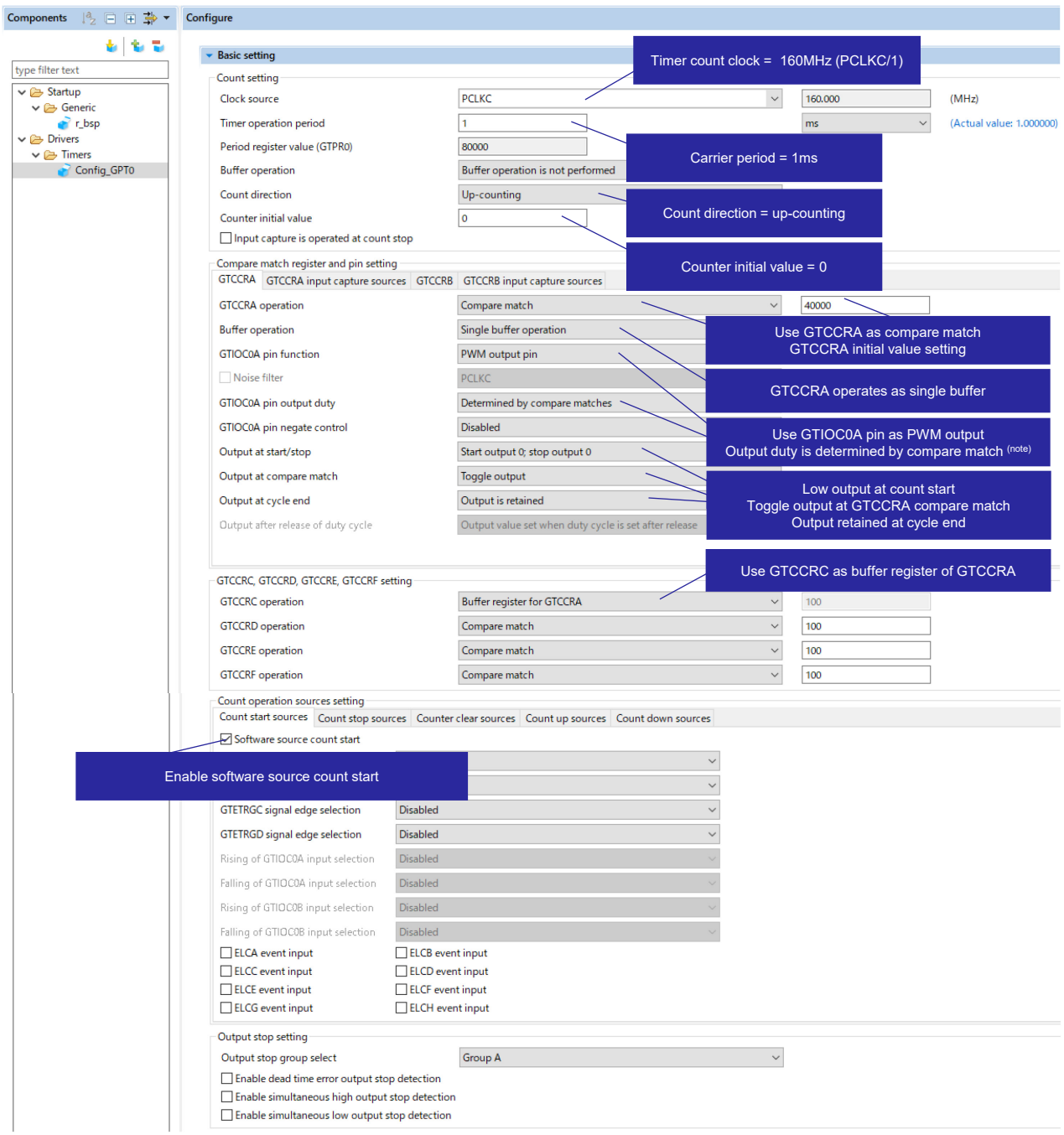
Figure 4.122 Sample Code Operations

4.13.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the GPTW as described below. For details on how to add components, refer to section 4.1.4 Adding Components.

Table 4.16 Adding Components

Item	Description
Component	General PWM Timer (GPTW)
Configuration name	Config_GPT0
Work mode	Triangle-wave PWM Mode 1
Resource	GPT0



Note: After setting to forced output duty when the counting starts, the output duty cycle needs to be set to 0%. Set 0% duty cycle output in the R_Config_GPT0_Create_UserInit function.

Figure 4.123 GPT0 Setting (1/2)

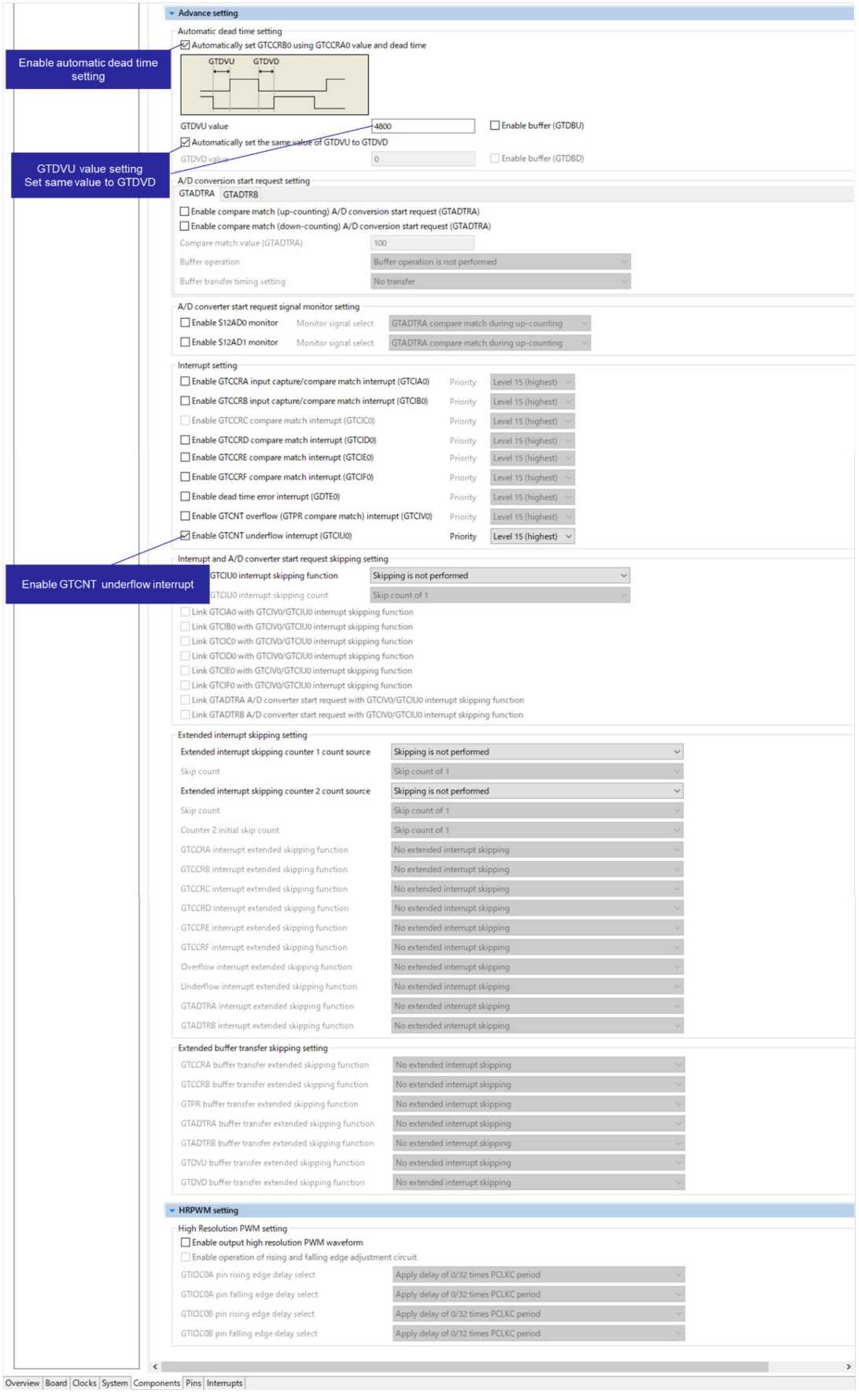


Figure 4.124 GPT0 Setting (2/2)

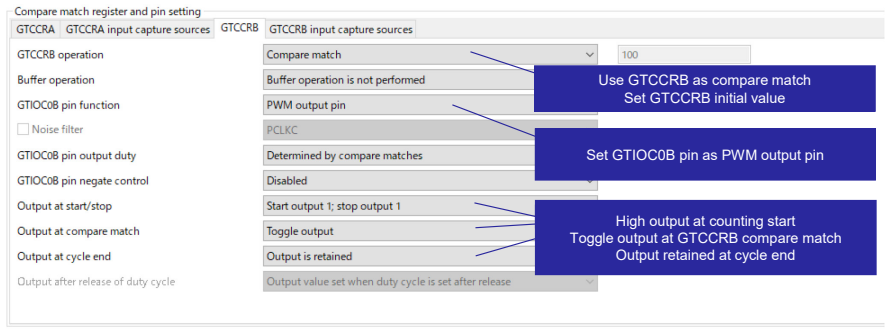


Figure 4.125 GPT0 Settings (Compare Match Register and Pins Setting of GTCCRB)

4.13.4 Flowcharts

The following flowcharts show the processing of a function added after code generation by the Smart Configurator.

In the main function, the counting is started and output is set to 100% duty for the next cycle.

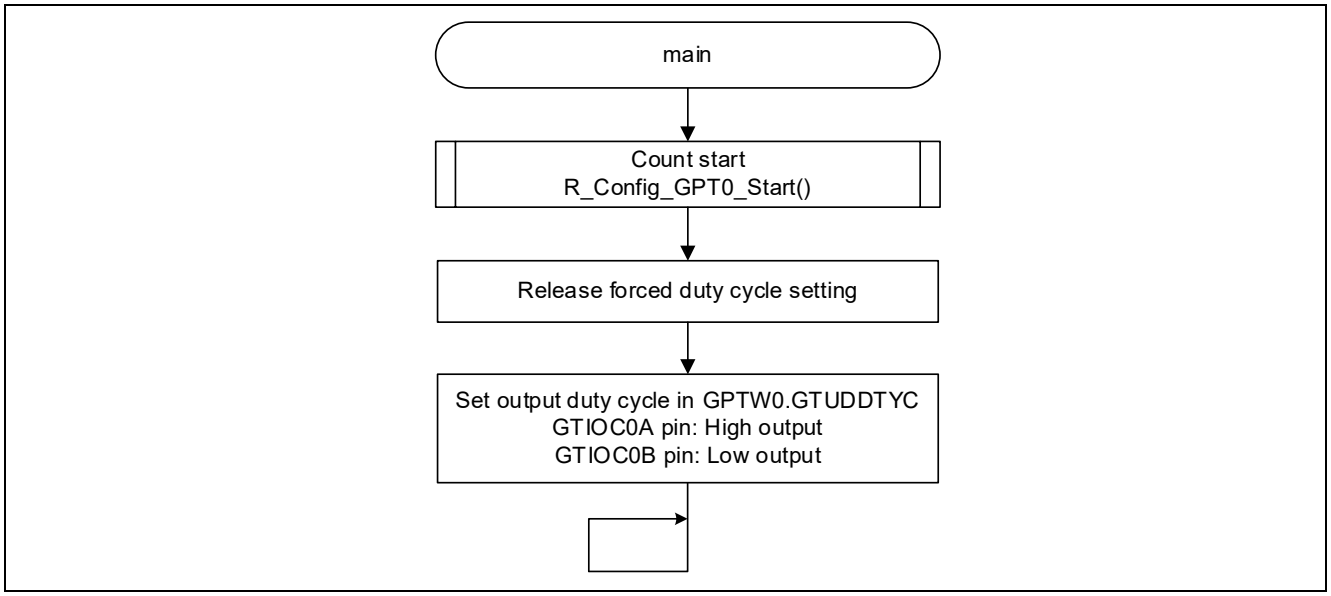


Figure 4.126 main Function

User initialization function R_Config_GPT0_Create_UserInit, which is executed before the main function, sets the forced output duty at counting starts and output duty cycle to 0%. This function is called from within the R_Config_GPT0_Create function.

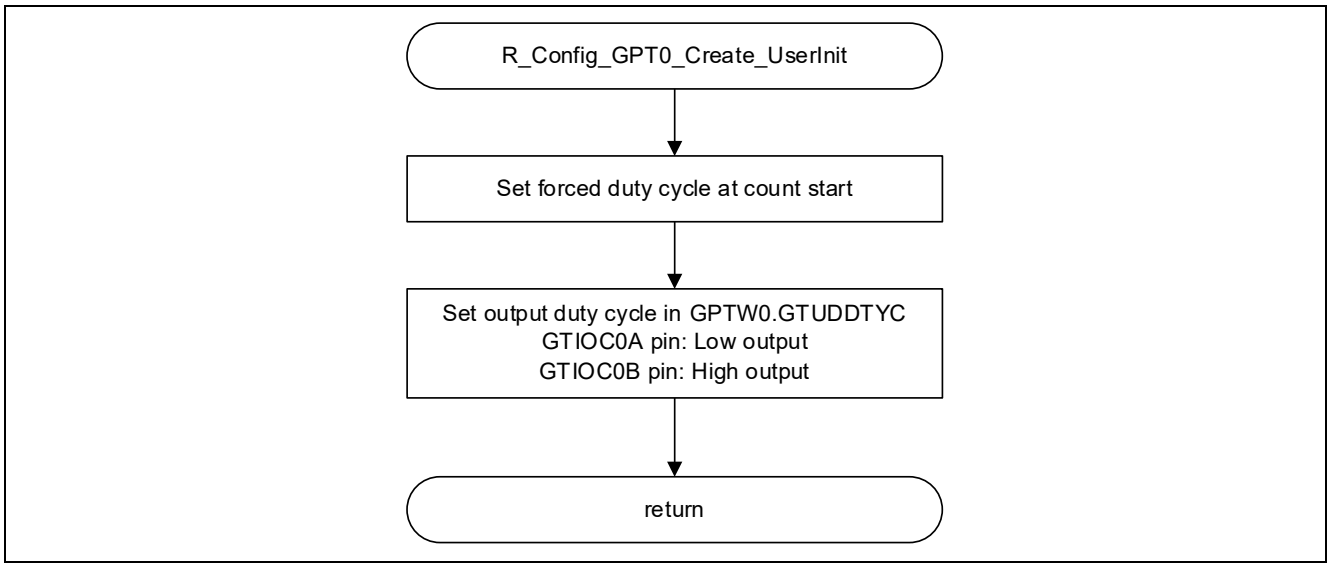


Figure 4.127 User Initialization Function

The GTCIU0 interrupt handler function changes the value of the GTUDDTYC register.

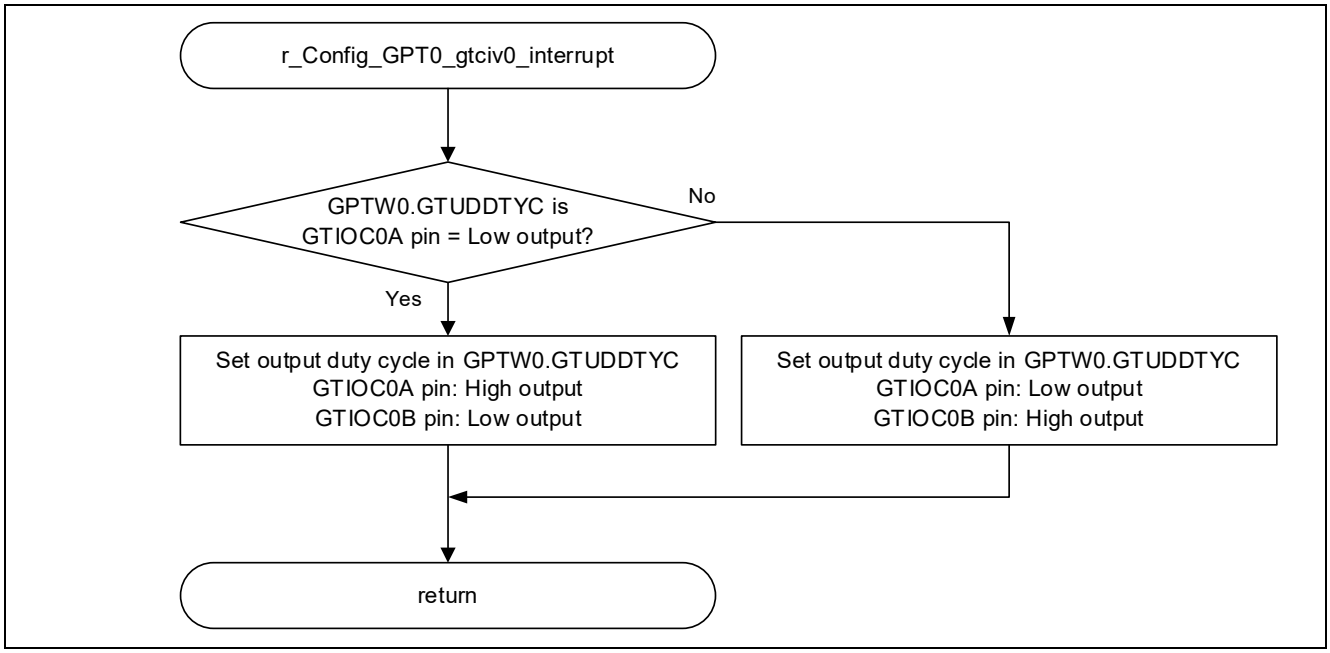


Figure 4.128 GTCIU0 Interrupt Handler Function

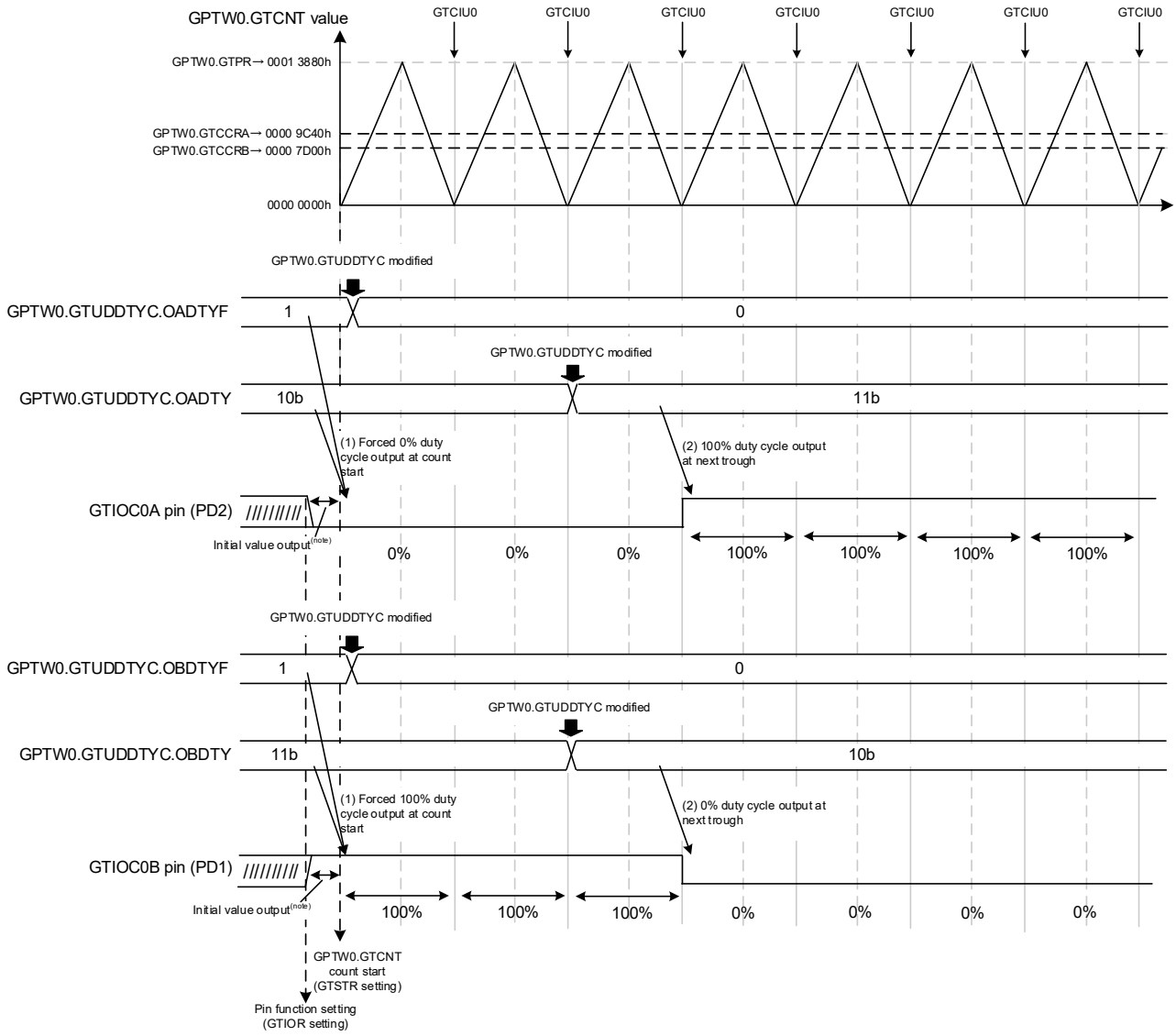
4.13.5 Related Operations

4.13.5.1 Change Duty Cycle to 100% After Holding 0% at Initial Value Low

The following shows an example of operations when the duty cycle is switched to 100% after the counting has started and 0% is output for several cycles.

The GTIOC0A pin outputs low immediately after the counting starts and output duty switches to 0% by setting the GTUDDTYC.OADTY bits to 10b while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped ((1) in figure below). During count operation, the output goes to high and duty cycle goes to 100% in the next cycle by setting the GTUDDTYC.OADTY bits to 11b ((2) in figure below).

The GTIOC0B pin outputs high immediately after the counting starts and output duty switches to 0% by setting the DDTYC.OBDTY bits to 11b while the GTUDDTYC.OBDTYF bit is 1b and the counting is stopped ((1) in figure below). During the count operation, output goes to low and duty cycle goes to 0% in the next cycle by setting the GTUDDTYC.OBDTY bits to 10b ((2) in figure below). In this example, GTUDDTYC.OBDTY bits are set to 11b to fix the GTIOC0B pin output to high and to output 100% duty cycle, and set to 10b to fix the GTIOC0B pin output to low and to output 0% duty cycle.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

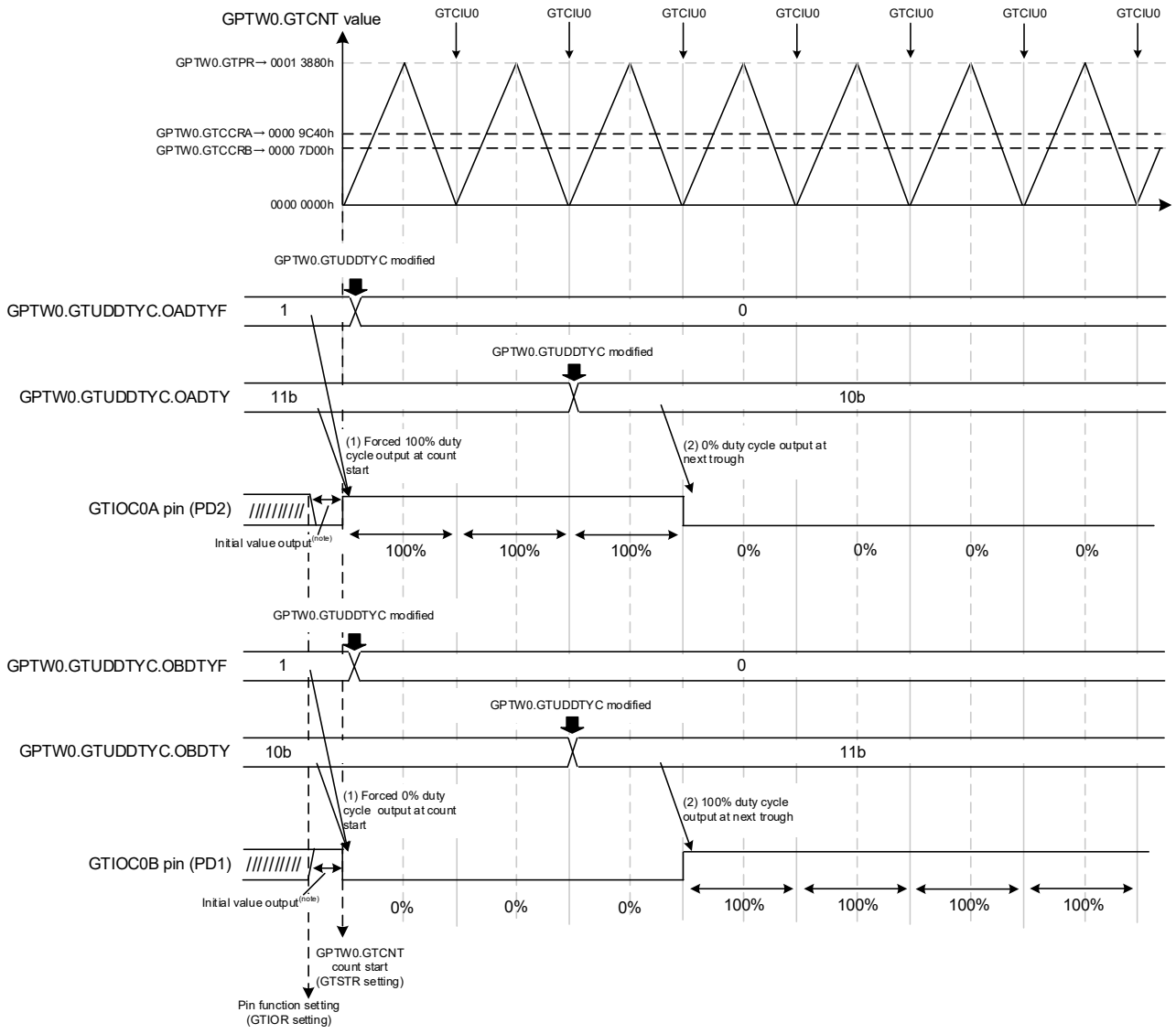
Figure 4.129 Change Duty Cycle to 100% After Holding 0%

4.13.5.2 Change Duty Cycle to 0% After Holding 100% at Initial Value Low

The following shows an example of operations when the duty cycle is switched to 0% after the counting has started and 100% is output for several cycles.

The GTIOC0A pin outputs high immediately after counting starts and output duty switches to 100% by setting the GTUDDTYC.OADTY bits to 11b while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped ((1) in figure below). During count operation, output goes to low and duty cycle goes to 0% in the next cycle by setting the GTUDDTYC.OADTY bits to 10b ((2) in figure below).

The GTIOC0B pin outputs low immediately after the counting starts and output duty switches to 0% by setting the GTUDDTYC.OBDTY bits to 10b while the GTUDDTYC.OBDTYF bit is 1b and the counting is stopped ((1) in figure below). During count operation, output goes to high and duty cycle goes to 100% in the next cycle by setting the GTUDDTYC.OBDTY bits to 11b ((2) in figure below). In this example, GTUDDTYC.OBDTY bits are set to 10b to fix the GTIOC0B pin output to high and to output 100% duty cycle, and set to 11b to fix the GTIOC0B pin output to low and to output 0% duty cycle.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

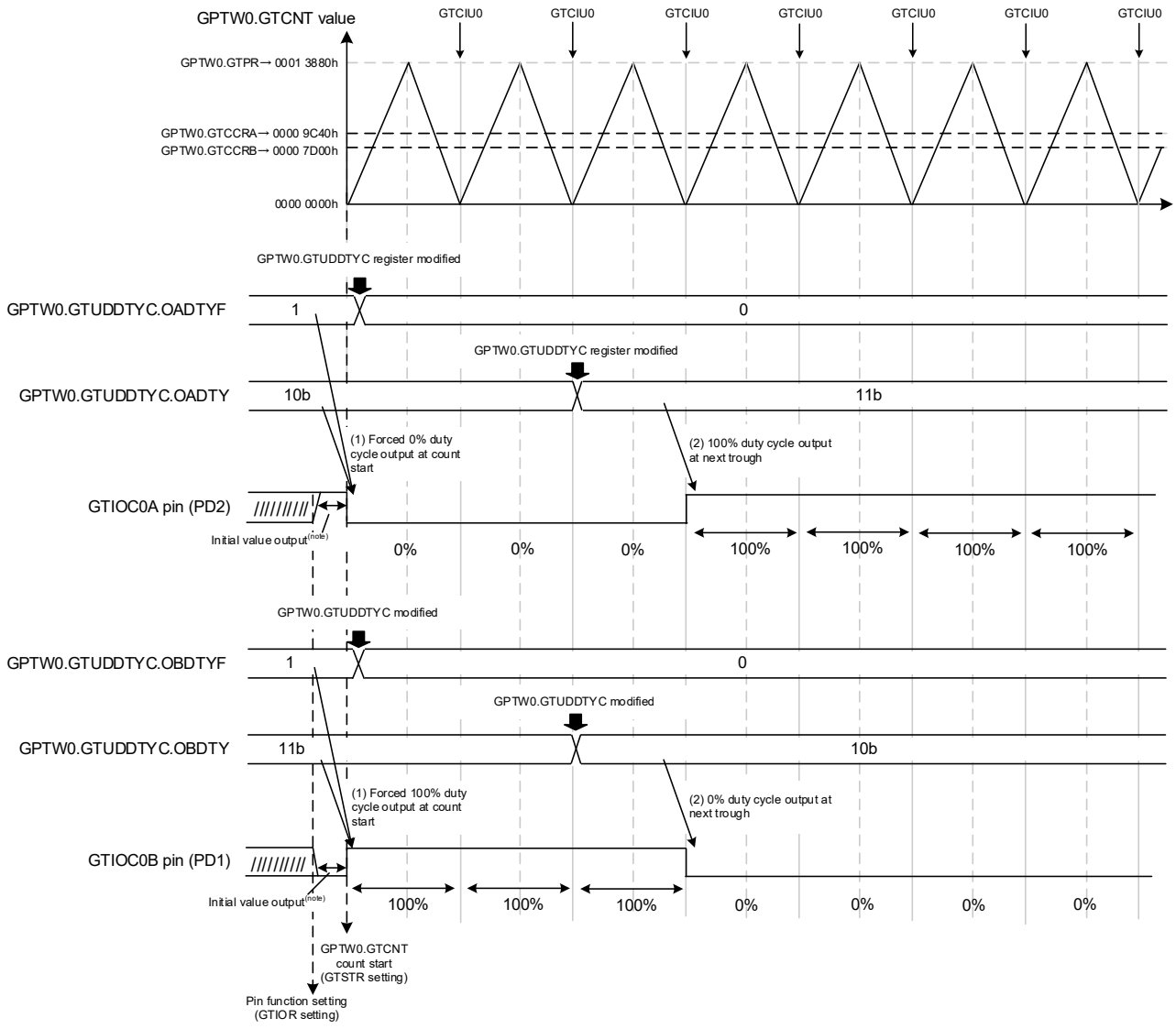
Figure 4.130 Change Duty Cycle to 0% After Holding 100%

4.13.5.3 Change Duty Cycle to 100% After Holding 0% at Initial Value High

The following shows an example of operations when the duty cycle is switched to 100% after the counting has started and 0% is output for several cycles.

The GTIOC0A pin outputs low immediately after the counting starts and output duty switches to 0% by setting the GTUDDTYC.OADTY bits to 10b while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped ((1) in figure below). During count operation, output goes to high and duty cycle goes to 100% in the next cycle by setting the GTUDDTYC.OADTY bits to 11b ((2) in figure below).

The GTIOC0B pin outputs high immediately after the counting starts and output duty switches to 100% by setting the GTUDDTYC.OBDTY bits to 11b while the GTUDDTYC.OBDTYF bit is 1b and the counting is stopped ((1) in figure below). During count operation, output goes to low and duty cycle goes to 0% in the next cycle by setting the GTUDDTYC.OBDTY bits to 10b ((2) in figure below). In this example, GTUDDTYC.OBDTY bits are set to 11b to fix the GTIOC0B pin output to high and to output 100% duty cycle, and set to 10b to fix the GTIOC0B pin output to low, and to 0% duty cycle.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.131 Change Duty Cycle to 100% After Holding 0%

To make the initial value of the GTIOC0A pin high, set “Output at start/stop” to “stop output 1.”

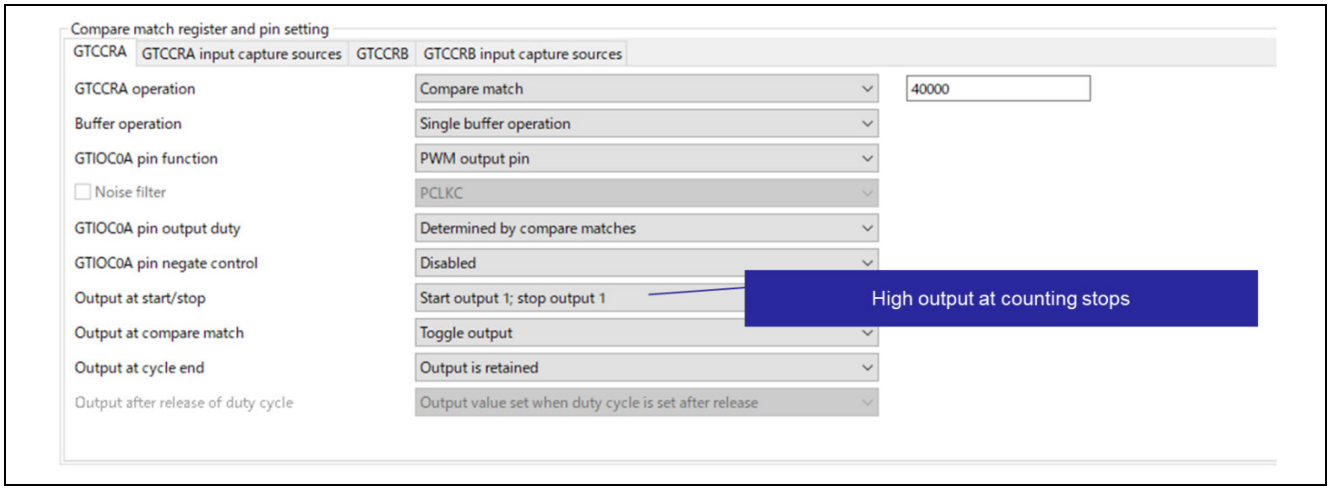


Figure 4.132 Smart Configurator Setting for Initial Value High

To make the initial value of the GTIOC0B pin low, set “Output at start/stop” to “stop output 0.”

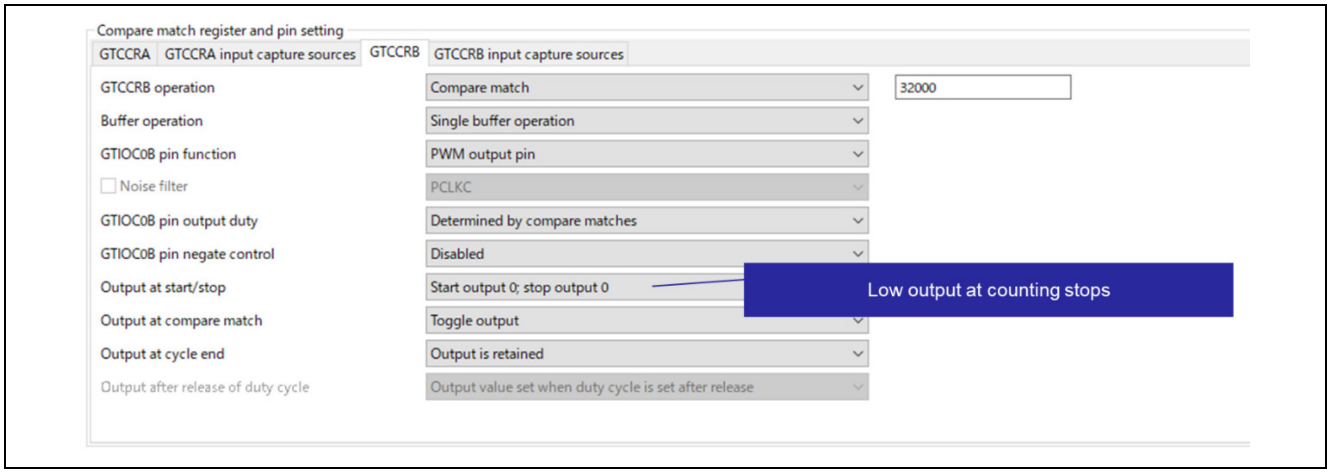


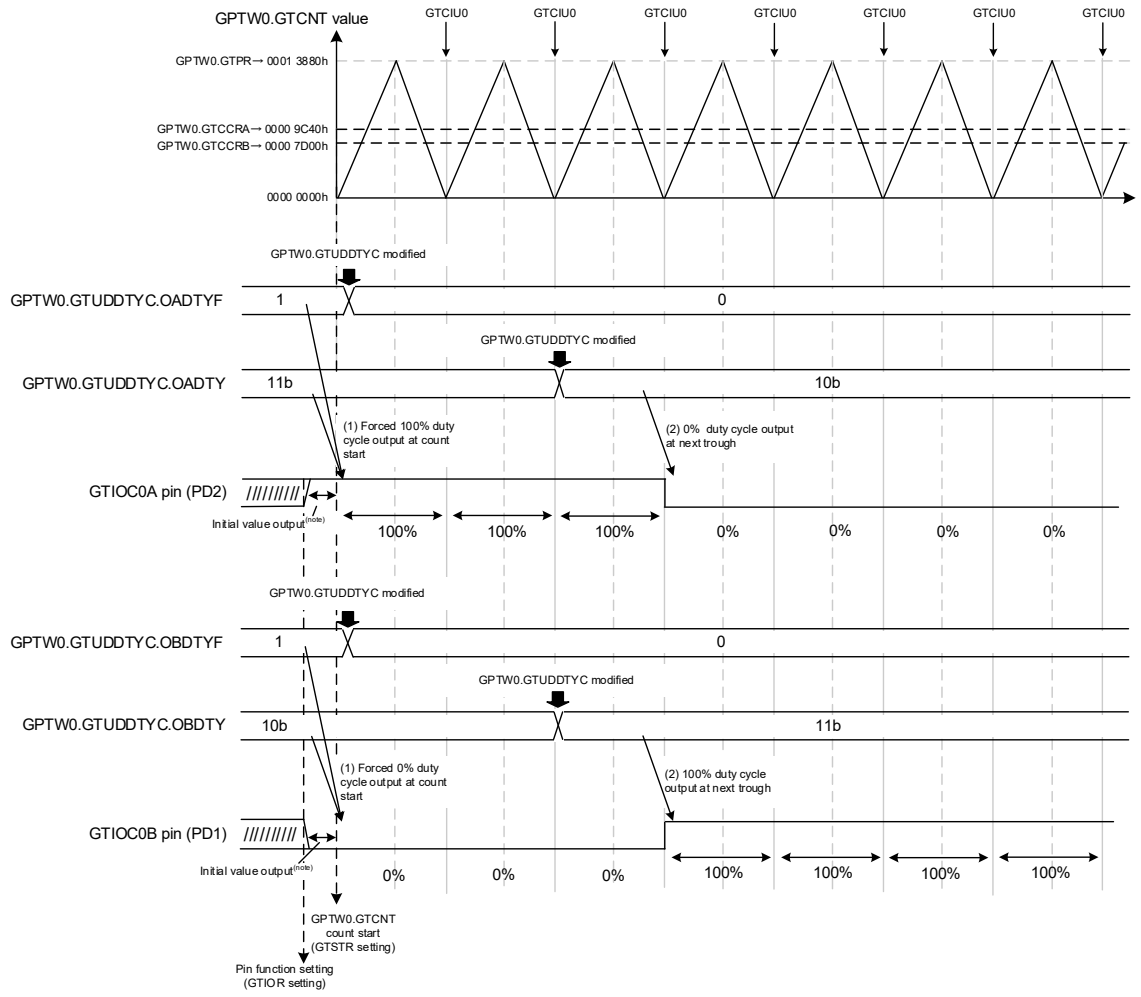
Figure 4.133 Smart Configurator Setting for Initial Value Low

4.13.5.4 Change Duty Cycle to 0% After Holding 100% at Initial Value High

The following shows an example of operations when the duty cycle is switched to 0% after the counting has started and 100% is output for several cycles.

The GTIOC0A pin outputs high immediately after the counting starts and output duty switches to 100% by setting the GTUDDTYC.OADTY bits to 11b while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped ((1) in figure below). During count operation, the output goes to low and duty cycle goes to 0% in the next cycle by setting the GTUDDTYC.OADTY bits to 10b ((2) in figure below).

The GTIOC0B pin outputs low immediately after the counting starts and output duty switches to 0% by setting the GTUDDTYC.OBDTY bits to 10b while the GTUDDTYC.OBDTYF bit is 1b and the counting is stopped % ((1) in figure below). During count operation, the output goes to high and duty cycle goes to 100% in the next cycle by setting the GTUDDTYC.OBDTY bits to 11b ((2) in figure below). In this example, the GTUDDTYC.OBDTY bits are set to 10b to fix the GTIOC0B pin output to low and to output 0% duty cycle, and to 11b to fix the GTIOC0B pin output to high and to output 100% duty cycle.



Note: The sample code waveform starts outputting the initial values when the PMR register is set.

Figure 4.134 Change Duty Cycle to 0% After Holding 100%

To make the initial value of the GTIOC0A pin high, set “Output at start/stop” to “start output 1.” Refer to Figure 4.132 for details.

In the same manner, to make the initial value of the GTIOC0B pin low, set “Output at start/stop” to “start output 0.” Refer to Figure 4.133 for details.

4.13.5.5 Change Duty Cycle to 100% After Holding 50%

The following shows an example of operations when the GTUDDTYC.OADTY and GTUDDTYC.OBDTY bits are changed to switch to 100% duty cycle after 50% duty has been output for several cycles.

The GTIOC0A pin output goes to high and 100% duty cycle in the next cycle by setting the GTUDDTYC.OADTY bits to 11b during count operation ((1) in figure below).

The GTIOC0B pin output goes to low and 0% duty cycle in the next cycle by setting the GTUDDTYC.OBDTY bits to 10b during count operation ((1) in figure below).

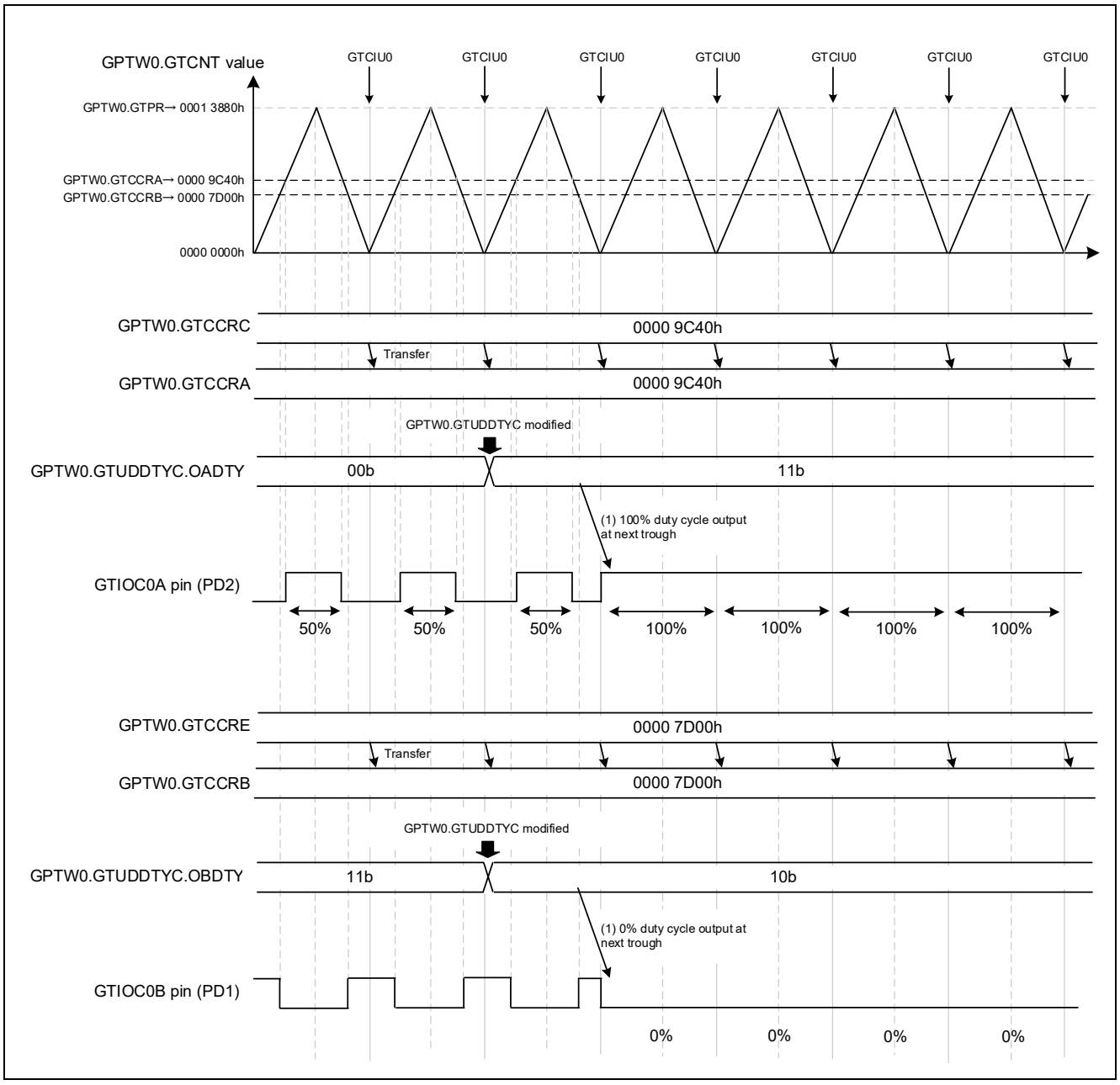


Figure 4.135 Change Duty Cycle to 100% After Holding 50%

Duty cycle cannot be switched to 100% at compare match without changing the GTUDDTYC.OADTY and GTUDDTYC.OBDTY bits during count operation.

4.13.5.6 Change Duty Cycle to 0% After Holding 50%

The following shows an example of operations when switching to 0% duty cycle after 50% duty has been output for several cycles without changing the GTUDDTYC.OADTY and GTUDDTYC.OBDTY bits.

When the GTCCRA and GTCCRB registers are set to values greater than the GTPR value, 0% duty cycle can be output because a compare match does not occur.

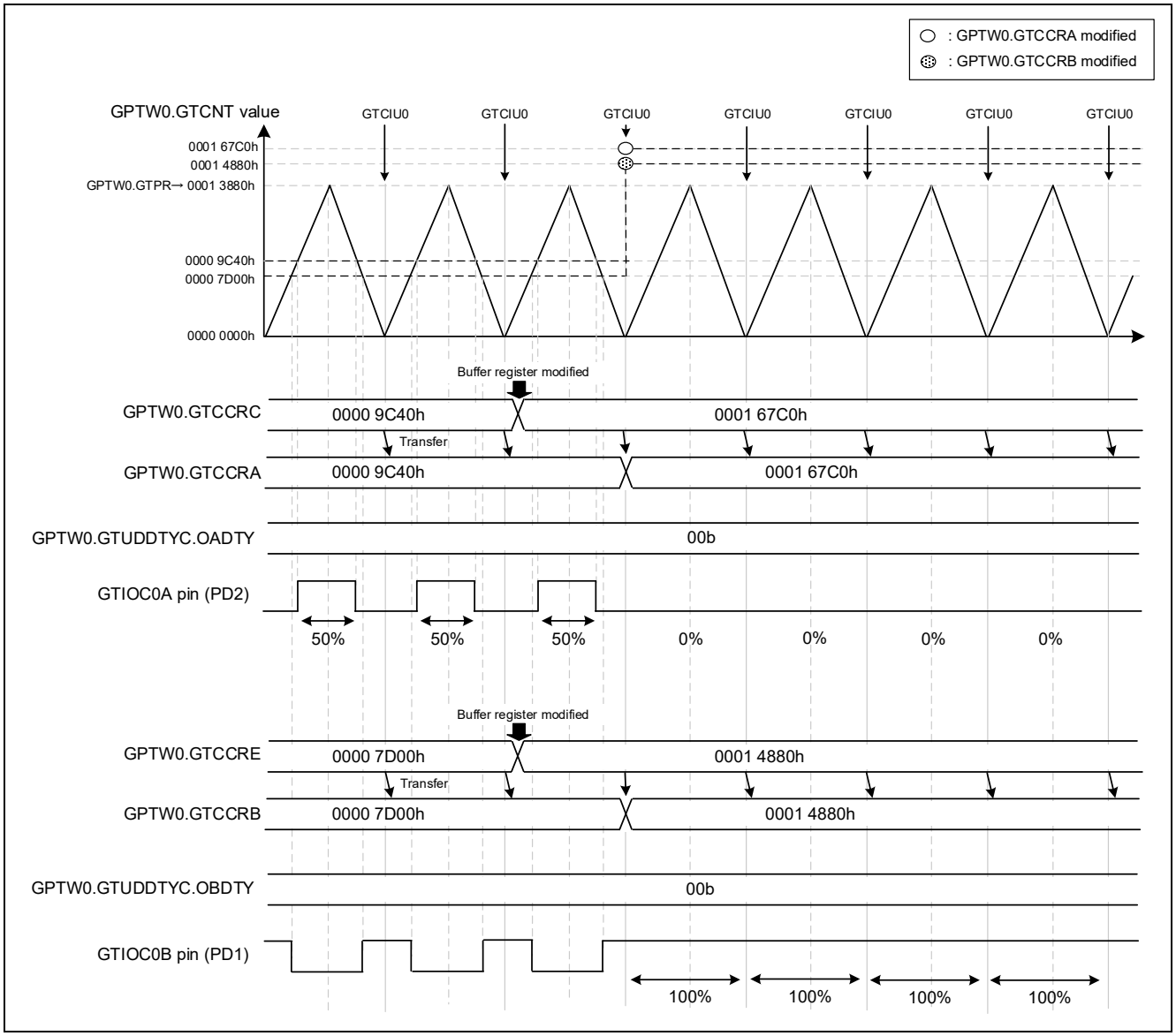


Figure 4.136 Change Duty Cycle to 0% After Holding 50%

The GTUDDTYC.OADTY and GTUDDTYC.OBDTY bits are set to 10b and 11b, respectively, during the counting operation to enable 0% duty cycle from the next cycle.

4.13.6 Usage Notes

4.13.6.1 Settings of the GTCCRm Register during Compare Match Operation (m = A to F)

This sample code automatically sets the dead time and uses the value of compare match register GTCCRA, the register for positive-phase waveform, to update the value of compare match register GTCCRB, the register for negative-phase waveform.

Set compare match register GTCCRA value to satisfy the following restrictions.

GTCCRA > GTDVU,
GTCCRA > GTDVD, and
GTCCRA < GTPR.

When the GTCCRA register is set to 0000 0000h or a value greater than or equal to the GTPR register value during count operation, the output protection function is activated.

However, if the following conditions are not satisfied, output protection does not operate normally.

When the value of GTCCRA when counting starts is 0000 0001h or greater and less than the setting value of GTPR

For details regarding the output protection function, refer to RX66T Group User's Manual Hardware, section 24.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 9; m = A, B).

When not using the automatic dead time setting function, set a value greater than 0000 0001h and less than the setting value of the GTPR register in the GTCCRA (GTCCRB) register. When 0000 0000h or the same value as the GTPR register is set in the GTCCRA register, a compare match occurs in a cycle only when [GTCCRA (GTCCRB) register = 0000 0000h] or [GTCCRA (GTCCRB) register = GTPR register] is met. Furthermore, if a value exceeding the setting value of the GTPR register is set in the GTCCRA register, a compare match does not occur.

For details, refer to RX66T Group User's Manual Hardware, section 24.10.2 Settings of the GTCCRm Register during Compare Match Operation (M = A to F): (1) When Automatic Dead Time Setting has been Made in Triangle-Wave PWM Mode and (2) When Automatic Dead Time Setting has not been Made in Triangle-Wave PWM Mode.

4.13.6.2 Reflection of GTUDDTYC.OADTY Setting at Counting Starts

In this sample code, the count operation duty is determined by the GTUDDTYC.OADTYF and GTUDDTYC.OADTY bits set while the counting is stopped.

If the value of the GTUDDTYC.OADTY bit is changed while the GTUDDTYC.OADTYF bit is 0b and the counting is stopped, the output duty setting changed at the start of counting is not reflected. To reflect the setting from when the counting starts, it is necessary to change the value of the GTUDDTYC.OADTY bit while the GTUDDTYC.OADTYF bit is 1b and counting is stopped, and then counting is started.

For details, refer to RX66T Group User's Manual Hardware, section 24.3.6 Duty Cycle 0%/100% Output Function.

4.13.6.3 Reflection of Duty Cycle at Counting Starts Using Smart Configurator

When the Smart Configurator is used and pin output duty cycle is set at 0% or 100% Figure 4.111, the duty is not reflected at the start of counting because the GTUDDTYC.OADTYF bit is 0b. To reflect the duty from the start of counting, the user must create a code to set the GTUDDTYC.OADTY bits to 10b or 11b and while the GTUDDTYC.OADTYF bit is 1b and the counting is stopped.

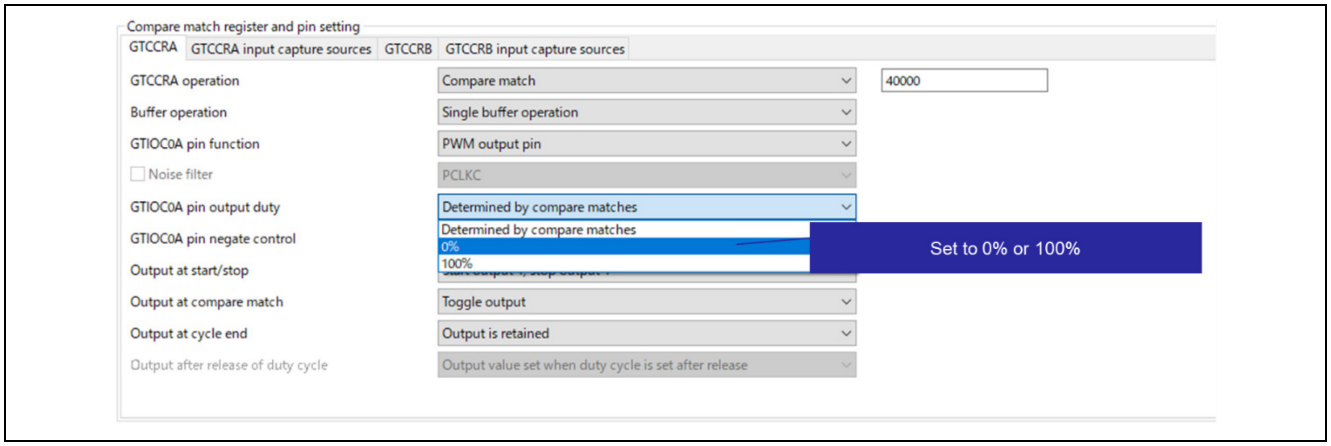


Figure 4.137 Set Pin Output Duty Cycle to 0% or 100%

For details on Smart Configurator settings, refer to section 4.13.3 Smart Configurator Settings, and for an example of user generated code, refer to section 4.13.4 Flowcharts.

4.13.6.4 100% Duty Cycle Output at Compare Match

Output duty cycle cannot be switched to 100% at compare match without changing the GTUDDTYC register. To output 100% duty, set the GTUDDTYC.OADTY bits to 11b and the GTUDDTYC.OBDTY bits to 10b.

If the GTCCRA and GTCCRB registers are set to 0 in this sample code settings, 100% duty cycle cannot be output because one clock cycle is output after a GTCNT counter underflow occurs.

When the GTCCRA and GTCCRB registers are set to the same value as the GTPR and a GTCNT counter underflow and compare match occur at the same time, the output setting at the timing of the compare match is prioritized and toggled, disabling 100% duty cycle output.

For details regarding the waveform output when the GPTW's GTCNT counter underflow and compare match occur at the same time, refer to the notes under Table 24.4 in RX66T Group User's Manual Hardware, section 24.2.14 General PWM Timer I/O Control Register (GTIOR).

4.13.6.5 Compare Match Operation during Duty Cycles 0% and 100% Output

In this sample code, output duty cycles 0% and 100% are determined based on the values set to the GTUDDTYC.OADTY bits. When duty cycle is set to either 0% or 100%, the GPTW internally continues to perform the compare match operation, interrupt output and buffer transfer operation.

This sample code does not use the compare match interrupt, so if you are using the compare match interrupt, do so with caution interrupt during duty cycle 0% and 100% output.

5. How to Import the Project

The sample code is provided in the format 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 Importing with e² studio

When using the sample code in e² studio, import it into e² studio using the following steps.

(The actual screen may vary according to the version of e² studio you are using.)

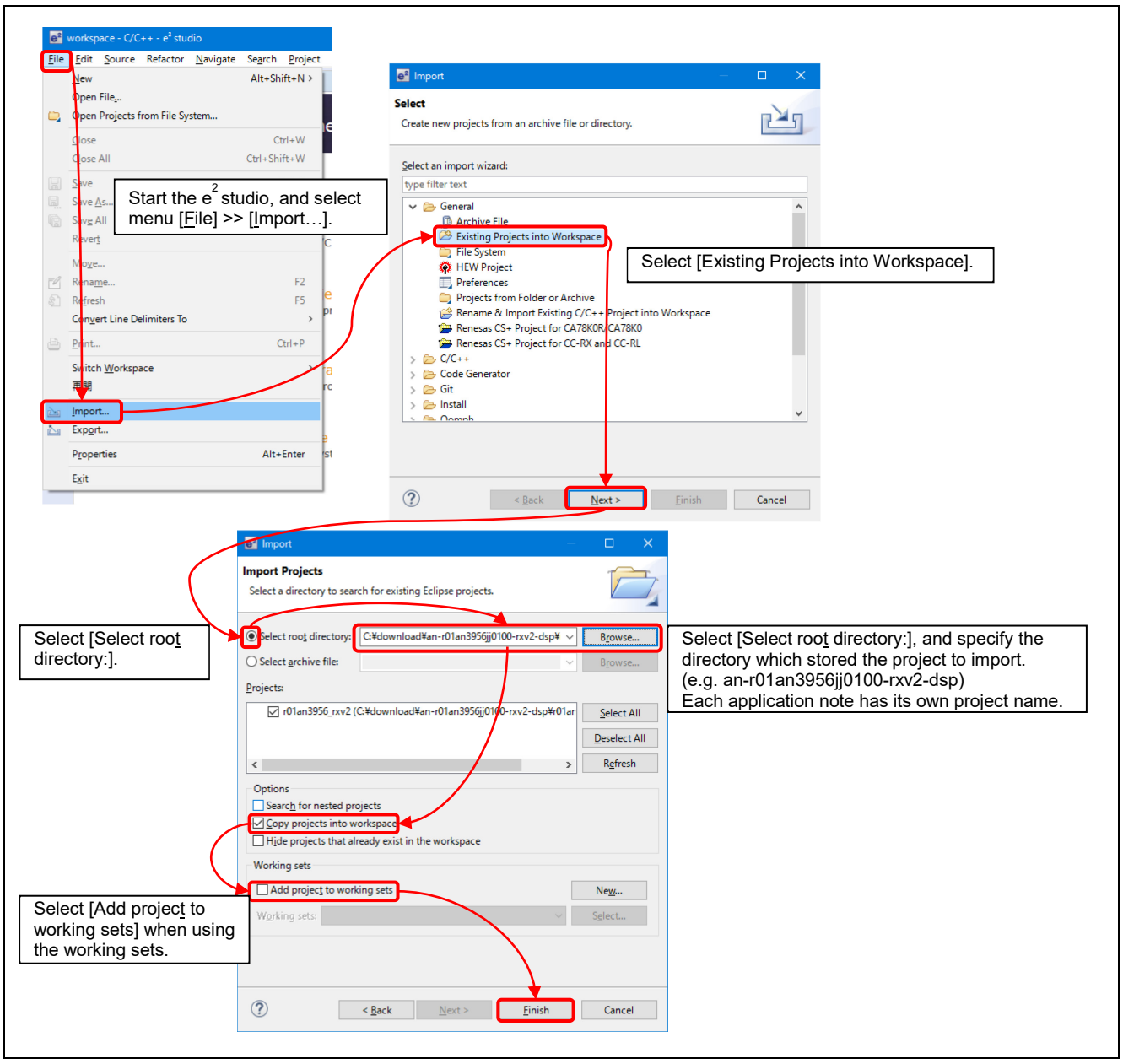


Figure 5.1 How to Import a Project into e² studio

5.2 Importing with CS+

When using the sample code with CS+, import the code to CS+ using the following steps.

(The actual screen may vary according to the version of CS+ you are using.)

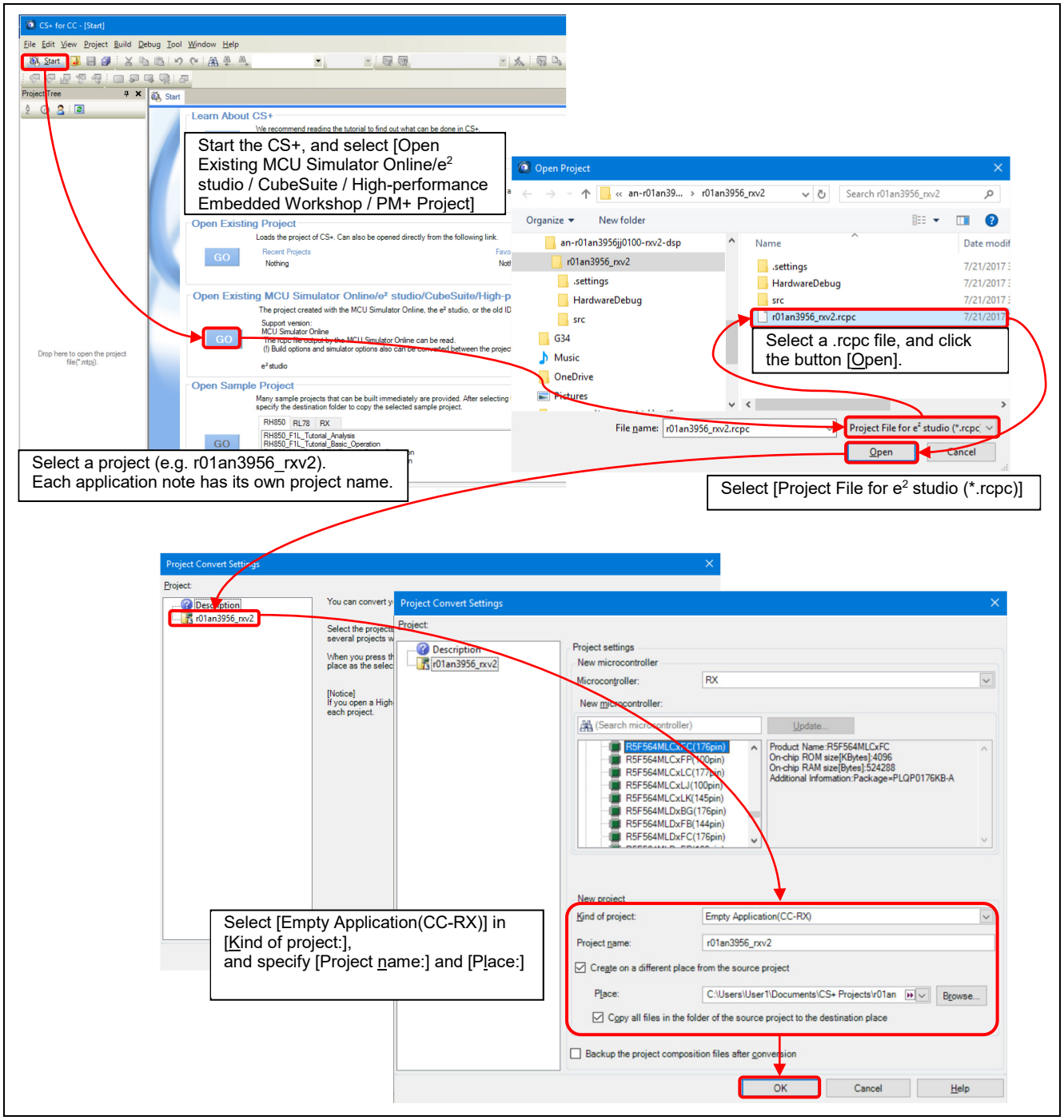


Figure 5.2 How to Import a Project into CS+

6. Reference Documents

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Apr.21.22	—	First edition issued
1.10	Jan.6.23	24	Change "Table1.14 Setting Methods for Duty Cycles 0% and 100%".
		33	Change "Figure 3.5 Interrupt Settings".
		92, 103, 146, 159	Change the note "Output Level Settings".
		170	Change "Figure 4.9 GPT0 Settings (1/2)".
		180	Change "Figure 4.17 GPT0 Settings (1/2)".
		191	Change "Figure 4.26 GPT0 Setting".

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/.