RX Family

Example of Using the A/D Conversion Start Request Delaying Function with MTU3/GPTW

Introduction

This application note describes the operation of the A/D conversion start request delaying function by using MTU3d and GPTW.

RX66T Group MCUs are equipped with the Multi-Function Timer Pulse Unit 3 (MTU3d) and the General PWM Timer (GPTW) to support generation of A/D conversion start request signals linked with the interrupt skipping function.

The descriptions in this application note target RX Family devices equipped with MTU3 and 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 Devices

RX Family devices with the MTU and GPTW

Confirmed Devices

RX66T Group

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

Contents

1. A/D Conversion Start Request Delaying Function of the MTU and GPTW

1.1 Overview of the A/D Conversion Start Request Delaying Function

Some of the interrupts generated by the MTU and GPTW can be used as A/D conversion start triggers. Table 1.1 lists the triggers that the MTU can use as A/D conversion start triggers, and Table 1.2 lists the triggers that the GPTW can use as A/D conversion start triggers. An existing A/D conversion start trigger using an interrupt is not effective to start A/D conversion at a fixed cycle, because it is linked to a change in duty cycle.

Therefore, the MTU and GPTW have two compare registers for A/D conversion start triggers separately from compare registers for PWM output, and the compare match timing with either register is output as a synchronous trigger. This synchronous trigger can also be selected as the A/D conversion start trigger. This makes it possible to generate A/D conversion timing independent of the PWM output compare match timing, allowing the A/D conversion timing to be set freely. This function to generate A/D conversion start triggers is called the A/D conversion start request delaying function in the MTU, and the A/D conversion start request function in the GPTW. For details, refer to section 22.3.9, A/D Conversion Start Request Delaying Function, and section 24.5, A/D Converter Start Request, in the RX66T Group User's Manual: Hardware.

Table 1.1 A/D Conversion Start Triggers for the MTU

Target Registers	Interrupt Source	A/D Conversion Start Request Signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGA0N
MTU9.TGRA and MTU9.TCNT		TRGA9N
MTU9.TGRA and MTU9.TCNT, MTU9.TGRE and MTU9.TCNT		TRG9AEN
MTU0.TGRA and MTU0.TCNT, MTU0.TGRE and MTU0.TCNT		TRG0AEN
MTU0.TGRA and MTU0.TCNT, MTU9.TGRA and MTU9.TCNT		TRGA09N
MTU1.TGRA and MTU1.TCNT		TRGA1N
MTU2.TGRA and MTU2.TCNT		TRGA2N
MTU3.TGRA and MTU3.TCNT		TRGA3N
MTU4.TGRA and MTU4.TCNT		TRGA4N
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU6.TGRA and MTU6.TCNT	Input capture/compare match	TRGA6N
MTU7.TGRA and MTU7.TCNT		TRGA7N
MTU7.TCNT	MTU7.TCNT trough in complementary PWM mode	
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0N
MTU9.TGRE and MTU9.TCNT		TRG9N
MTU0.TGRE and MTU0.TCNT, MTU9.TGRE and MTU9.TCNT		TRG09N
MTU4.TADCORA and MTU4.TCNT	Compare match (A/D conversion start request delaying function)	TRG4AN
MTU4.TADCORB and MTU4.TCNT		TRG4BN
MTU7.TADCORA and MTU7.TCNT		TRG7AN
MTU7.TADCORB and MTU7.TCNT		TRG7BN
MTU4.TADCORA and MTU4.TCNT, MTU4.TADCORB and MTU4.TCNT		TRG4ABN
MTU7.TADCORA and MTU7.TCNT, MTU7.TADCORB and MTU7.TCNT		TRG7ABN

Table 1.2 A/D Conversion Start Triggers for the GPTW

Target Registers	Interrupt Source	A/D Conversion Start Request Signal
GTCCRA and GTCNT	Input capture/compare match	GTCIAn
GTCCRB and GTCNT		GTCIBn
GTCCRC and GTCNT	Compare match	GTCICn
GTCCRD and GTCNT		GTCIDn
GTCCRE and GTCNT		GDTEn
GTCCRF and GTCNT		GTCIEn
GTCNT	GTCNT counter overflow (compare match of the GTPR register)	GTCIFn
	GTCNT counter underflow	GTCIVn
GTADTRA and GTCNT	Compare match	A/D conversion start request A
GTADTRB and GTCNT		A/D conversion start request B

The figure below shows an example of MTU operation in complementary PWM mode (MTU3/MTU4 used, 1 phase only).

MTU3.TGRB: Compare register for PWM output
MTIOC3B: PWM output pin (positive phase)
MTIOC3D: PWM output pin (negative phase)
MTU4.TADCORA: Timer A/D conversion start request cycle setting register A
MTU4.TADCORB: Timer A/D conversion start request cycle setting register B
TRG4AN: A/D conversion start request signal (enabled only when MTU4.TCNT is up-counting)
TRG4BN: A/D conversion start request signal (enabled when MTU4.TCNT is up-/down-counting)

An A/D conversion start trigger (TGR4AN, TGR4BN) is output on a compare match between the TADCORA/TADCORB setting and MTU4.TCNT.

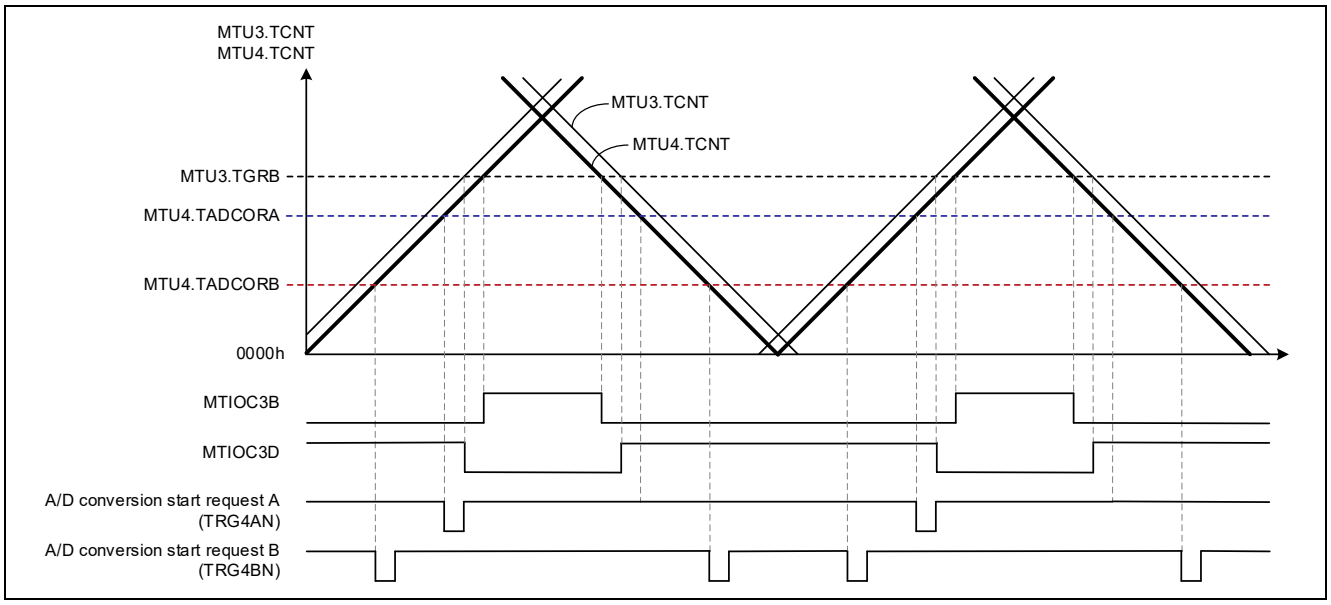


Figure 1.1 A/D Conversion Start Request Delaying Function of the MTU

From the next page, the A/D conversion start request delaying function of the MTU and GPTW is described in detail.

1.2 Basic Operation of the A/D Conversion Start Request Delaying Function

The table below lists the basic specifications for the A/D conversion start request delaying function of the MTU and the A/D conversion start request function of the GPTW.

Both the MTU and GPTW can have two timings per channel.

Table 1.3 Specifications of the A/D Conversion Start Request Delaying Function

Item	MTU	GPTW
Channel	Channels 4 and 7	Channels 0 to 9
Operating mode	• Complementary PWM mode 1/2/3 • Reset-synchronized PWM mode • PWM mode 1 • Normal mode	• Triangle-wave PWM mode 1/2/3 • Sawtooth-wave one-shot pulse mode • Sawtooth-wave PWM mode
A/D conversion start request register	Timer A/D conversion start request cycle setting register (TADCORA, TADCORB)	A/D conversion start request timing register (GTADTRA, GTADTRB)
A/D conversion start request generation timing	• During counting up • During counting down • During counting up and counting down	• During counting up • During counting down • During counting up and counting down
A/D conversion start request buffer configuration	• Single buffer (TADCOBRA, TADCOBRB)	• Single buffer (GTADTBRA, GTADTBRB) • Double buffer (GTADTDBRA, GTADTDBRB)
A/D conversion start request buffer transfer timing	TADCORA and TADCORB operate at the same timing, which can be selected from the following: • Crest/overflow • Trough • Crest and trough	GTADTRA and GTADTRB can be set to different timings, which can be selected from the following: • Crest/overflow • Trough • Crest and trough

The MTU supports the A/D conversion start request delaying function only on channel 4 (MTU4) and channel 7 (MTU7). To start the A/D converter on other channels, use functions other than the A/D conversion start request delaying function. For details, refer to section 22.4.3, A/D Converter Trigger Sources, in the RX66T Group User's Manual: Hardware.

Both the MTU and GPTW support the A/D conversion start request delaying function only in the operation modes shown in Table 1.3. For details on the MTU to start the A/D converter in an operation mode other than the above, refer to section 22.4.3, A/D Converter Trigger Sources, in the RX66T Group User's Manual: Hardware. For details on the GPTW, refer to section 24.6, Operations Linked by the ELC, in the RX66T Group User's Manual: Hardware, and use other than the generation of A/D conversion start request A/B.

1.2.1 Basic Operation of the MTU

The A/D conversion start request delaying function can be used on channel 4 (MTU4) and channel 7 (MTU7) of the MTU. This application note uses MTU4 as an example.

A/D conversion start requests can be issued by setting the timer A/D conversion start request control register (MTU4.TADCR), the timer A/D conversion start request cycle setting registers (MTU4.TADCORA, MTU4.TADCORB), and the timer A/D conversion start request cycle setting buffer registers (MTU4.TADCOBRA, MTU4.TADCOBRB).

At a compare match between MTU4.TCNT counter and MTU4.TADCORA and MTU4.TADCORB, the A/D conversion start request delaying function generates the respective A/D conversion start request (TRG4AN, TRG4BN).

The figure below shows an example of basic operation when the buffer transfer timing is set to troughs of MTU4.TCNT and the A/D conversion start request signal (TRG4AN) is set to be output when MTU4.TCNT is down-counting in complementary PWM mode.

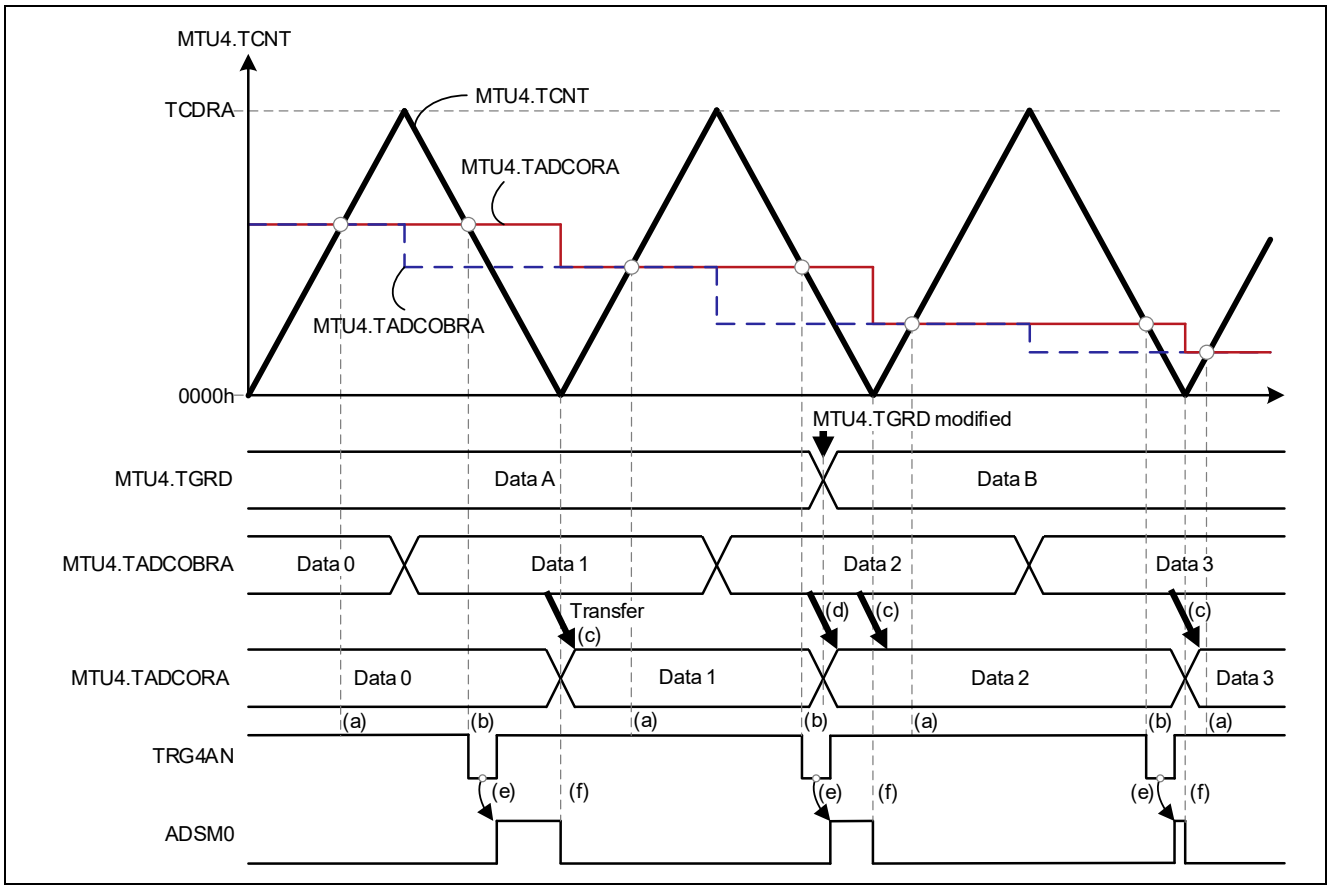


Figure 1.2 Example of Basic Operation of A/D Conversion Start Request Signal (TRG4AN) (Complementary PWM Mode, TRG4AN Output Enabled During Down-Counting, Buffer Transfer Timing: Trough)

Even if the MTU4.TCNT counter and the MTU4.TADCORA register match during the up-counting period of the MTU4.TCNT counter (0 ≤ MTU4.TCNT ≤ TCDRA - 1), a request to start A/D conversion (TRG4AN) is not issued (see (a) in Figure 1.2). If the MTU4.TCNT counter and the MTU4.TADCORA register match during the down-counting period of the MTU4.TCNT counter (TCDRA ≥ MTU4.TCNT ≥ 1), a request to start A/D conversion (TRG4AN) is issued ((b) in Figure 1.2).

Data is transferred from MTU4.TADCOBRA to MTU4.TADCORA at troughs of MTU4.TCNT ((c) in Figure 1.2). In complementary PWM mode, data is also transferred from MTU4.TADCOBRA to MTU4.TADCORA at the rewrite timing of the MTU4.TGRD register ((d) in Figure 1.2).

The MTU allows the generation timing of the A/D conversion start request signal to be monitored with an external pin by using the A/D conversion start request frame synchronization signal. A pulse signal is output from the ADSM0 pin that goes high ((e) in Figure 1.2) at the timing of A/D conversion start request signal generation and goes low ((f) in Figure 1.2) at the end of an MTU4.TCNT cycle.

(1) Enabling an A/D conversion start request

When the MTU4.TCNT counter matches MTU4.TADCORA and MTU4.TADCORB within the period allowed by the UT4AE, DT4AE, UT4BE, and DT4BE bits in the MTU4.TADCR register, the respective A/D conversion start request signals (TRG4AN and TRG4BN) are generated.

Note that the bits that can be used in complementary PWM mode and other modes differ as follows.

Table 1.4 Operating Mode-Dependent TADCR Settings

Symbol	Complementary PWM Mode	Reset-Synchronized PWM Mode PWM Mode 1 Normal Mode
UT4AE	A/D conversion start requests (TRG4AN) enabled/disabled during MTU4.TCNT up-count operation	
DT4AE	A/D conversion start requests (TRG4AN) enabled/disabled during MTU4.TCNT down-count operation	0 (Setting prohibited)
UT4BE	A/D conversion start requests (TRG4BN) enabled/disabled during MTU4.TCNT up-count operation	
DT4BE	A/D conversion start requests (TRG4BN) enabled/disabled during MTU4.TCNT down-count operation	0 (Setting prohibited)

For details, refer to section 22.2.35, Timer A/D Conversion Start Request Control Register (TADCR), in the RX66T Group User's Manual: Hardware.

(2) Buffer transfer

The data in the timer A/D conversion start request cycle setting registers (MTU4.TADCORA, MTU4.TADCORB) is updated by writing data to the buffer registers (MTU4.TADCOBRA, MTU4.TADCOBRB). The transfer timing from the buffer register to the timer A/D conversion start request cycle setting register can be selected by setting the BF[1:0] bits of the MTU4.TADCR register.

The transfer timing depends on the timer operation mode and the setting of BF[1:0] bits in the MTU4.TADCR register. For details, see the table below.

Table 1.5 Difference in Transfer Timing Depending on Operation Mode and TADCR.BF[1:0] Bit Setting (for MTU4)

Symbol		Description			
BF[1]	BF[0]	Complementary PWM Mode	Reset-Synchronized PWM Mode	PWM Mode 1	Normal Mode
0	0	No transfer	No transfer	No transfer	No transfer
0	1	Transfer at crests of MTU4.TCNT	Transfer at compare matches between MTU3.TCNT and MTU3.TGRA	Transfer at compare matches between MTU4.TCNT and MTU4.TGRA	Transfer at compare matches between MTU4.TCNT and MTU4.TGRA
1	0	Transfer at troughs of MTU4.TCNT	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Transfer at both crests and troughs of MTU4.TCNT	Setting prohibited	Setting prohibited	Setting prohibited

For details, refer to section 22.2.35, Timer A/D Conversion Start Request Control Register (TADCR), in the RX66T Group User's Manual: Hardware.

In complementary PWM mode, data is also transferred from the buffer register to the timer A/D conversion start request cycle setting register at the rewrite timing of the MTU4.TGRD register.

When using buffer transfer in complementary PWM mode, note the following regarding buffer transfer timing.

- When the MTU4.TADCOBRA and MTU4.TADCOBRB registers are set to 0000h, the UT4AE and UT4BE bits of the MTU4.TADCR register are set to 1b, and a buffer transfer is performed at a trough of the MTU4.TCNT counter, an A/D conversion start request is not issued during the up-counting period immediately after the transfer.
- When the MTU4.TADCOBRA and MTU4.TADCOBRB registers are set to the same value as the TCDR register, the DT4AE and DT4BE bits of the MTU4.TADCR register are set to 1b, and a buffer transfer is performed at a crest of the MTU4.TCNT counter, an A/D conversion start request is not issued during the down-counting period immediately after the transfer.

For details, refer to section 22.6.28, Usage Notes on A/D Conversion Start Request Delaying Function in Complementary PWM Mode, in the RX66T Group User's Manual: Hardware.

(3) A/D conversion start request frame synchronization signal

Select the A/D conversion request signal to be monitored with TADSTRSn[4:0] bits (n = 0, 1) of the TADSTRGRn register and enable the ADSMn pin output with the TADSTRGRn.TADSMENn bit. By doing this, a pulse signal is output from the ADSMn pin that goes high at the timing of A/D conversion start request signal generation and goes low in the timer cycle used to generate the A/D conversion start request signal.

For the A/D conversion start request delaying function, set the TADSTRSn[4:0] bits as follows.

Table 1.6 Difference in A/D Conversion Start Request Frame Synchronization Signal Depending on the Setting of the TADSTRSn[4:0] Bits (n = 0, 1)

TADSTRSn[4:0]					Source	Description
[4]	[3]	[2]	[1]	[0]		
0	1	0	0	1	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT
0	1	0	1	0	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT
0	1	1	0	0	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and between MTU4.TADCORB and MTU4.TCNT (when using interrupt skipping function 2)

For details, refer to section 22.2.43, A/D Conversion Start Request Select Register 0 (TADSTRGR0), section 22.2.44, A/D Conversion Start Request Select Register 1 (TADSTRGR1), and section 22.3.15, A/D Conversion Start Request Frame Synchronization Signal, in the RX66T Group User's Manual: Hardware.

1.2.2 Basic Operation of the GPTW

The A/D conversion start request function can be used on each channel of the GPTW.

A/D conversion start requests can be issued by setting the general PWM timer interrupt output setting register (GTINTAD), A/D conversion start request timing register (GTADTRA, GTADTRB), A/D conversion start request timing buffer register (GTADTBRA, GTADTBRB), and A/D conversion start request timing double-buffer register (GTADTDBRA, GTADTDBRB).

At a compare match between the GTCNT counter and GTADTRA/GTADTRB, the respective A/D conversion start request signals (A/D conversion start request A and A/D conversion start request B) are generated.

The figure below shows an example of basic operation when the buffer transfer timing is set to troughs of GTCNT and the A/D conversion start request is set to be output when GTCNT is down-counting in triangle-wave PWM mode.

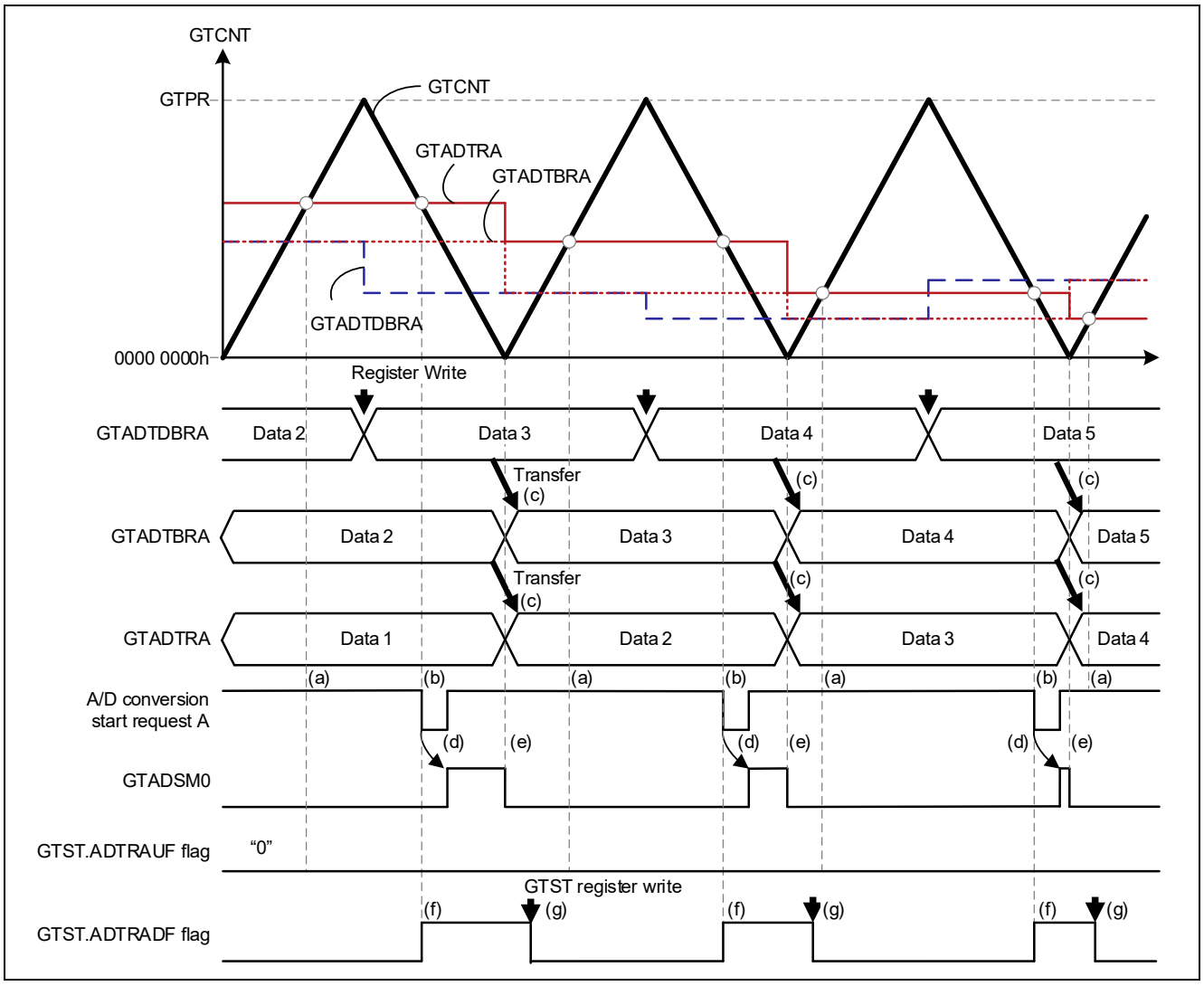


Figure 1.3 Example of Basic Operation of A/D Conversion Start Request Signal (Triangle-Wave PWM Mode, A/D Conversion Start Request A Output Enabled During Down-Counting, Buffer Transfer Timing: Trough)

Even if the GTCNT counter and the GTADTRA register match during the up-counting period of the GTCNT counter (0 ≤ GTCNT ≤ GTPR - 1), a request to start A/D conversion (A/D conversion start request A) is not issued (see (a) in Figure 1.3). If the GTCNT counter and the GTADTRA register match during the down-counting period of the GTCNT counter (GTPR ≥ GTCNT ≥ 1), a request to start A/D conversion (A/D conversion start request A) is issued ((b) in Figure 1.3).

Data is transferred from GTADTDBRA to GTADTBRA, and then to GTADTRA at troughs of GTCNT ((c) in Figure 1.3).

The generation timing of the A/D conversion start request signal can be monitored with an external pin. A pulse signal is output from the GTADSM0 pin that goes high ((d) in Figure 1.3) at the generation timing of A/D conversion start request A and goes low ((e) in Figure 1.3) at the end of a GTCNT cycle.

The A/D conversion start request A can be checked with the A/D conversion start request flag in the general PWM timer status register (GTST). When the GTCNT counter matches the GTADTRA register during the down-counting period of the GTCNT counter, A/D conversion start request A is generated and the ADTRADF flag of the GTST register is set to 1b ((f) in Figure 1.3). The ADTRADF flag is cleared by setting it to 0b ((g) in Figure 1.3). Note that in the setting shown in Figure 1.3, the ADTRAUF flag is held at 0b because A/D conversion start request A does not occur during the up-counting period of the GTCNT counter.

(1) Enabling an A/D conversion start request

When the GTCNT counter matches GTADTRA/GTADTRB within the period allowed by the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register, the respective A/D conversion start request signals (A/D conversion start request A and A/D conversion start request B) are generated.

There is no difference depending on the mode of operation.

Table 1.7 Relationship Between GTINTAD Register Setting and A/D Conversion Start Request Signal

Symbol	Triangle-Wave PWM Mode 1/2/3 Sawtooth-Wave One-Shot Pulse Mode Sawtooth-Wave PWM Mode
ADTRAUEN	This bit enables or disables A/D conversion start requests generated by compare matches between GTCNT and the GTADTRA register during GTCNT up-counting.
ADTRADEN	This bit enables or disables A/D conversion start requests generated by compare matches between GTCNT and the GTADTRA register during GTCNT down-counting.
ADTRBUEN	This bit enables or disables A/D conversion start requests generated by compare matches between GTCNT and the GTADTRB register during GTCNT up-counting.
ADTRBDEN	This bit enables or disables A/D conversion start requests generated by compare matches between GTCNT and the GTADTRB register during GTCNT down-counting.

Note that the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits are disabled during event count operation and no A/D conversion start request is generated.

For details, refer to section 24.2.15, General PWM Timer Interrupt Output Setting Register (GTINTAD), in the RX66T Group User's Manual: Hardware.

(2) Buffer transfer

The table below shows the relationship between the GTADTRA and GTADTRB registers and the buffer registers.

Table 1.8 Relationship Between the GTADTRA and GTADTRB Registers and the Buffer Registers

A/D Conversion Start Request Timing Register	Single Buffer	Double Buffer
GTADTRA	GTADTBRA	GTADTDBRA
GTADTRB	GTADTBRB	GTADTDBRB

To set GTADTRA or GTADTRB to function as a single buffer or a double buffer, set BD[2], ADTDA, and ADTDB bits in the GTBER register as follows.

Table 1.9 Difference in Buffer Operation Depending on GTBER Register Setting

Symbol			Description
BD[2]	ADTDA	ADTDB	
1	x	x	Disable buffer operation of the GTADTRA and GTADTRB registers.
0	0	0	• The GTADTRA register operates as a single buffer. GTADTBRA → GTADTRA • The GTADTRB register operates as a single buffer. GTADTBRB → GTADTRB
0	1	0	• The GTADTRA register operates as a double buffer. GTADTDBRA → GTADTBRA → GTADTRA • The GTADTRB register operates as a single buffer. GTADTBRB → GTADTRB
0	0	1	• The GTADTRA register operates as a single buffer. GTADTBRA → GTADTRA • The GTADTRB register operates as a double buffer. GTADTDBRB → GTADTBRB → GTADTRB
0	1	1	• The GTADTRA register operates as a double buffer. GTADTDBRA → GTADTBRA → GTADTRA • The GTADTRB register operates as a double buffer. GTADTDBRB → GTADTBRB → GTADTRB

The transfer timing can be set differently for the GTADTRA and GTADTRB registers. Set ADTTA[1:0] and ADTTB[1:0] bits in the GTBER register as follows.

Table 1.10 Transfer Timing of GTADTRA Register

Symbol		Description	
ADTTA[1]	ADTTA[0]	Triangle-Wave PWM Mode 1/2/3	Sawtooth-Wave One-Shot Pulse Mode Sawtooth-Wave PWM Mode
0	0	No transfer	No transfer
0	1	Transfer at crests of GTCNT	• Transfer at underflows (during down-counting) • Transfer at overflows (during up-counting) • Transfer at counter clearing
1	0	Transfer at troughs of GTCNT	
1	1	Transfer at both crests and troughs of GTCNT	

Table 1.11 Transfer Timing of GTADTRB Register

Symbol		Description	
ADTTB[1]	ADTTB[0]	Triangle-Wave PWM Mode 1/2/3	Sawtooth-Wave One-Shot Pulse Mode Sawtooth-Wave PWM Mode
0	0	No transfer	No transfer
0	1	Transfer at crests of GTCNT	• Transfer at underflows (during down-counting) • Transfer at overflows (during up-counting) • Transfer at counter clearing
1	0	Transfer at troughs of GTCNT	
1	1	Transfer at both crests and troughs of GTCNT	

For details, refer to section 24.2.17, General PWM Timer Buffer Enable Register (GTBER) and section 24.3.2.3, Buffer Operation for the GTADTRA and GTADTRB Registers, in the RX66T Group User's Manual: Hardware.

(3) A/D conversion start request monitor output pin

The timing of the generation of requests to start A/D conversion can be monitored by an external pin. When the A/D conversion start request signal to be monitored is selected in the GTADSMR.ADSMSk bit (k = 0, 1) and when the output is enabled in the ADSMENk bit, a signal is output in synchronization with a cycle frame of the timer used to generate the A/D conversion start request signal, of which the output is driven high at the generation of the A/D conversion start request signal by the GTADSMk pin, or at the end of the cycle at which the output is driven low. When the counter stops, the value when the counter stopped is retained for output. Set the ADSMENk bit to 0b to output the low level.

When a signal to request the start of A/D conversion is generated at the end of a timer cycle, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next cycle.

When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPTW.

The GTADSMR.ADSMSk bit (k = 0, 1) can be set as follows.

Note that, unlike the MTU specifications for the A/D conversion start request frame synchronization signal, only the A/D conversion start request signal during up-counting or down-counting can be monitored.

Table 1.12 Difference in A/D Conversion Start Request Monitor Output Pin Depending on the Setting of the ADSMSk[1:0] Bits (k = 0, 1)

ADSMSk		
[1]	[0]	Description
0	0	A/D conversion start request signal generated by the GTADTRA register during up-counting
0	1	A/D conversion start request signal generated by the GTADTRA register during down-counting
1	0	A/D conversion start request signal generated by the GTADTRB register during up-counting
1	1	A/D conversion start request signal generated by the GTADTRB register during down-counting

For details, refer to section 24.2.32, General PWM Timer A/D Converter Start Request Signal Monitoring Register (GTADSMR), in the RX66T Group User's Manual: Hardware.

(4) Delay of the A/D conversion start request signal

The A/D conversion start request signal generated by the GPTW can be used as an A/D conversion start trigger by outputting it as a startup source to the A/D converter via the ELC. Therefore, there is a delay between the generation of the A/D conversion start request signal and the actual start of A/D conversion.

For details, refer to Figure 24.117, Example of A/D Converter Start Request Timing Operation, in the RX66T Group User's Manual: Hardware.

1.3 A/D Conversion Start Request Skipping Function

As with the timer interrupt skipping function, A/D conversion start requests can be skipped. The table below list the types of skipping.

Table 1.13 Types of Timers and Skipping Functions

Timer	Function Name	Operating Mode	Description
MTU	Interrupt skipping function 1	• Complementary PWM mode 1/2/3	Skips A/D conversion start requests in conjunction with skipped TGIA3 (TGIA6) and TCIV4 (TCIV7)
	Interrupt skipping function 2	• Complementary PWM mode 1/2/3 • Reset-synchronized PWM mode • PWM mode 1 • Normal mode	Skips A/D conversion start requests by using a dedicated counter
GPTW	Interrupt skipping function by using the GTITC register	• Triangle-wave PWM mode 1/2/3 • Sawtooth-wave one-shot pulse mode • Sawtooth-wave PWM mode	Skips A/D conversion start requests in conjunction with the overflow or underflow count of GTCNT
	Extended interrupt skipping function	• Triangle-wave PWM mode 1/2/3 • Sawtooth-wave one-shot pulse mode • Sawtooth-wave PWM mode	Skips A/D conversion start requests and buffer transfers in conjunction with the overflow or underflow count of GTCNT

1.3.1 Interrupt Skipping Function 1 (MTU)

In complementary PWM mode, interrupts TGIA3 (TGIA6) (at crests) and TCIV4 (TCIV7) (at troughs) can be skipped up to seven times by setting the TITCR1A (TITCR1B) register. This function is called interrupt skipping function 1.

A/D conversion start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be skipped in coordination with interrupt skipping function 1 by setting the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the MTU4.TADCR (MTU7.TADCR) register.

In modes other than complementary PWM mode, the A/D conversion start request delaying function linked with interrupt skipping function 1 cannot be used. Set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the MTU4.TADCR (MTU7.TADCR) register to 0b.

The figure below shows an example of A/D conversion start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and down-counting and A/D conversion start requests are linked with interrupt skipping 1.

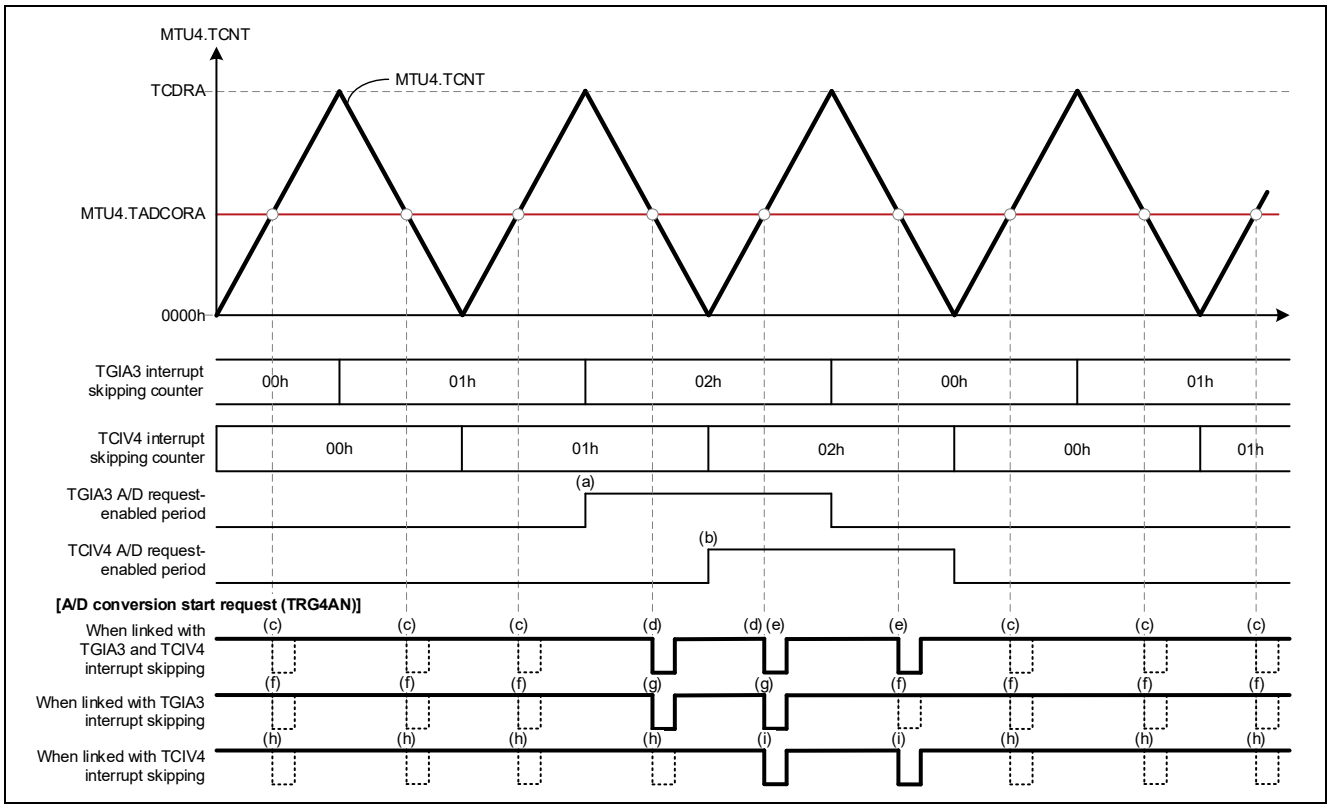


Figure 1.4 Example of A/D Conversion Start Request Signal (TRG4AN) Operation When Linked with Interrupt Skipping Function 1 (Complementary PWM Mode, TRG4AN Output Enabled During Up-/Down-Counting, Interrupt Skipping Count: 2)

When MTU3 and MTU4 start counting, the TGIA3 interrupt skipping counter and TCIV4 interrupt skipping counter start counting-up operation. Bits T4VCOR[2:0] and T3ACOR[2:0] in the timer interrupt skipping setting register 1 (TITCR1A) are set to 010b and 010b, respectively, so both the interrupt skipping counts for TGIA3 and TCIV4 are 2. While the value of the TGIA3 interrupt skipping counter is 02h, the A/D request enabled period of TGIA3 is high ((a) in Figure 1.4). Similarly, while the value of the TCIV4 interrupt skipping counter is 02h, the A/D request enabled period of TCIV4 is high ((b) in Figure 1.4).

When A/D conversion start requests are set to be linked with TGIA3 and TCIV4 interrupt skipping, TRG4AN that occurs while the A/D request enabled period of TGIA3 is low and the A/D request enabled period of TCIV4 is low is disabled ((c) in Figure 1.4). TRG4AN that occurs while the A/D request enabled period of TGIA3 is high or the A/D request enabled period of TCIV4 is high is enabled ((d) and (e) in Figure 1.4).

When A/D conversion start requests are set to be linked with TGIA3 interrupt skipping, TRG4AN that occurs while the A/D request enabled period of TGIA3 is low is disabled ((f) in Figure 1.4). TRG4AN that occurs while the A/D request enabled period of TGIA3 is high is enabled ((g) in Figure 1.4).

When A/D conversion start requests are set to be linked with TCIV4 interrupt skipping, TRG4AN that occurs while the A/D request enabled period of TCIV4 is low is disabled ((h) in Figure 1.4). TRG4AN that occurs while the A/D request enabled period of TCIV4 is high is enabled ((i) in Figure 1.4).

(1) Register settings

To use with interrupt skipping function 1, registers must be set as follows.

Table 1.14 Registers Used for Interrupt Skipping Function 1

For MTU3 and MTU4	For MTU6 and MTU7	Register Name	Description
TITMRA	TITMRB	Timer interrupt skipping mode register	Selects two types of skipping functions
TITCR1A	TITCR1B	Timer interrupt skipping setting register 1	Enables or disables interrupt skipping and specify the interrupt skipping count (0 to 7)
TITCNT1A	TITCNT1B	Timer interrupt skipping counter 1	Skipping counter for each interrupt source
MTU4.TADCR	MTU7.TADCR	Timer A/D conversion start request control register	Specifies whether to link A/D conversion start requests with the interrupt skipping function

The TITCR1A and TITCR1B settings are valid only when the TITMRA.TITM and TITMRB.TITM bits are set to 0b, respectively. When the TITMRA.TITM and TITMRB.TITM bits are set to 1b, the values of TITCR1A, TITCNT1A, TITCR1B, and TITCNT1B are cleared.

TITCNT1A and TITCNT1B count up when an interrupt source (TGIA3, TCIV4, TGIA6, TCIV7) occurs, and are cleared to 0b when the count matches the skipping count set by TITCR1A and TITCR1B. They retain their values even after stopping the count operation of MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT).

Table 1.15 shows the TITCR1A settings for MTU3 and MTU4, and Table 1.16 shows the MTU4.TADCR settings.

Table 1.15 Differences of Interrupt Skipping Function 1 Depending on TITCR1A Register Settings

Symbol	Description
T4VCOR[2:0]	Sets TCIV4 interrupt skipping count within the range from 0 to 7
T4VEN	0: Disables TCIV4 interrupt skipping 1: Enables TCIV4 interrupt skipping
T3ACOR[2:0]	Sets TGIA3 interrupt skipping count within the range from 0 to 7
T3AEN	0: Disables TGIA3 interrupt skipping 1: Enables TGIA3 interrupt skipping

When the interrupt skipping count is set to 0 (TITCR1A.T4VCOR[2:0] = 000b or TITCR1A.T3ACOR[2:0] = 000b), no skipping is performed.

Table 1.16 Differences of Interrupt Skipping Function 1 Depending on MTU4.TADCR Register Settings

Symbol	Description
ITB4VE	0: A/D conversion start request signal TRG4BN is not linked with TCIV4 interrupt skipping 1. 1: A/D conversion start request signal TRG4BN is linked with TCIV4 interrupt skipping 1.
ITB3AE	0: A/D conversion start request signal TRG4BN is not linked with TGIA3 interrupt skipping 1. 1: A/D conversion start request signal TRG4BN is linked with TGIA3 interrupt skipping 1.
ITA4VE	0: A/D conversion start request signal TRG4AN is not linked with TCIV4 interrupt skipping 1. 1: A/D conversion start request signal TRG4AN is linked with TCIV4 interrupt skipping 1.
ITA3AE	0: A/D conversion start request signal TRG4AN is not linked with TGIA3 interrupt skipping 1. 1: A/D conversion start request signal TRG4AN is linked with TGIA3 interrupt skipping 1.

The A/D conversion start request (TRG4AN and TRG4BN) can be set to be linked/not linked with skipped TGIA3 and TCIV4, respectively.

For details on interrupt skipping function 1, refer to (3) Interrupt Skipping Function 1 in Complementary PWM Mode in section 22.3.8, Complementary PWM Mode, in the RX66T Group User's Manual: Hardware.

For details of each register, refer to section 22.2.35, Timer A/D Conversion Start Request Control Register (TADCR), section 22.2.39, Timer Interrupt Skipping Set Register 1m (TITCR1m) (m = A, B), and section 22.2.40, Timer Interrupt Skipping Counter 1m (TITCNT1m) (m = A, B), in the RX66T Group User's Manual: Hardware.

(2) Notes when linking with interrupt skipping function 1

Note the following when linking with interrupt skipping function 1.

- When interrupt skipping 1 is disabled, the A/D conversion start request delaying function should be set to not link with interrupt skipping 1.
  Interrupt skipping is disabled when:
  TITCR1A.T4VEN = 0b, TITCR1A.T3AEN = 0b, or TITCR1A.T4VCOR[2:0] = 000b,
  TITCR1A.T3ACOR[2:0] = 000b (when TITCR1B.T7VEN = 0b, TITCR1B.T6AEN = 0b, or when TITCR1B.T7VCOR[2:0] = 000b and TITCR1B.T6ACOR[2:0] = 000b)
  Settings not to link with Interrupt skipping 1:
  Set ITA3AE = 0b, ITA4VE = 0b, ITB3AE = 0b, ITB4VE = 0b in the MTU4.TADCR register
  (set ITA6AE = 0b, ITA7VE = 0b, ITB6AE = 0b, ITB7VE = 0b in the MTU7.TADCR register).
- To link with interrupt skipping function 1, the MTU4.TADCORA and MTU4.TADCORB (MTU7.TADCORA and MTU7.TADCORB) registers should be set to the following values:
  2 ≤ MTUn.TADCORA/TADCORB ≤ TCDR - 2 (n = 4, 7)
- The TITCR1A (TITCR1B) register should be set while the TITM bit in the timer interrupt skipping mode register (TITMRA or TITMRB) is set to 0b, TGIA3 (TGIA6) interrupt requests are disabled by setting the MTU3.TIER (MTU6.TIER) register, and TCIV4 (TCIV7) interrupt requests are disabled by setting the MTU4.TIER (MTU7.TIER) register when a compare match is not generated.
  Before changing the skipping count, make sure that you clear the T3AEN (T6AEN) and T4VEN (T7VEN) bits to 0b to clear the skipping counter.

For details, refer to (3) Interrupt Skipping Function 1 in Complementary PWM Mode in section 22.3.8, Complementary PWM Mode, (5) A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 1 in section 22.3.9, A/D Conversion Start Request Delaying Function, and section 22.6.28, Usage Notes on A/D Conversion Start Request Delaying Function in Complementary PWM Mode, in the RX66T Group User's Manual: Hardware.

1.3.2 Interrupt Skipping Function 2 (MTU)

Skipping function 2 is a method of counting the occurrences of A/D conversion start requests (TRG4AN, TRG4BN (TRG7AN, TRG7BN)) using a dedicated counter to skip A/D conversion start requests.

By setting the TITM bit to 1b in the TITMRA (TITMRB) register, the counter starts down-counting from the value (0 to 7) set in the TRG4COR[2:0] (TRG7COR[2:0]) bits in TITCR2A (TITCR2B) register every time an existing A/D conversion start trigger (TGR4AN or TRG4BN (TGR7AN or TRG7BN)) is generated. When the counter reaches 0 and is reloaded, the TRG4AN and TRG4BN (TRG7AN and TRG7BN) interrupts become valid and the A/D conversion start request signal (TRG4ABN (TRG7ABN)) is output.

This function is valid only when the A/D conversion start request delaying function is enabled.

The figure below shows an example of interrupt skipping function 2 operation in complementary PWM mode when TRG4AN and TRG4BN outputs are enabled during MTU4.TCNT up-counting and down-counting.

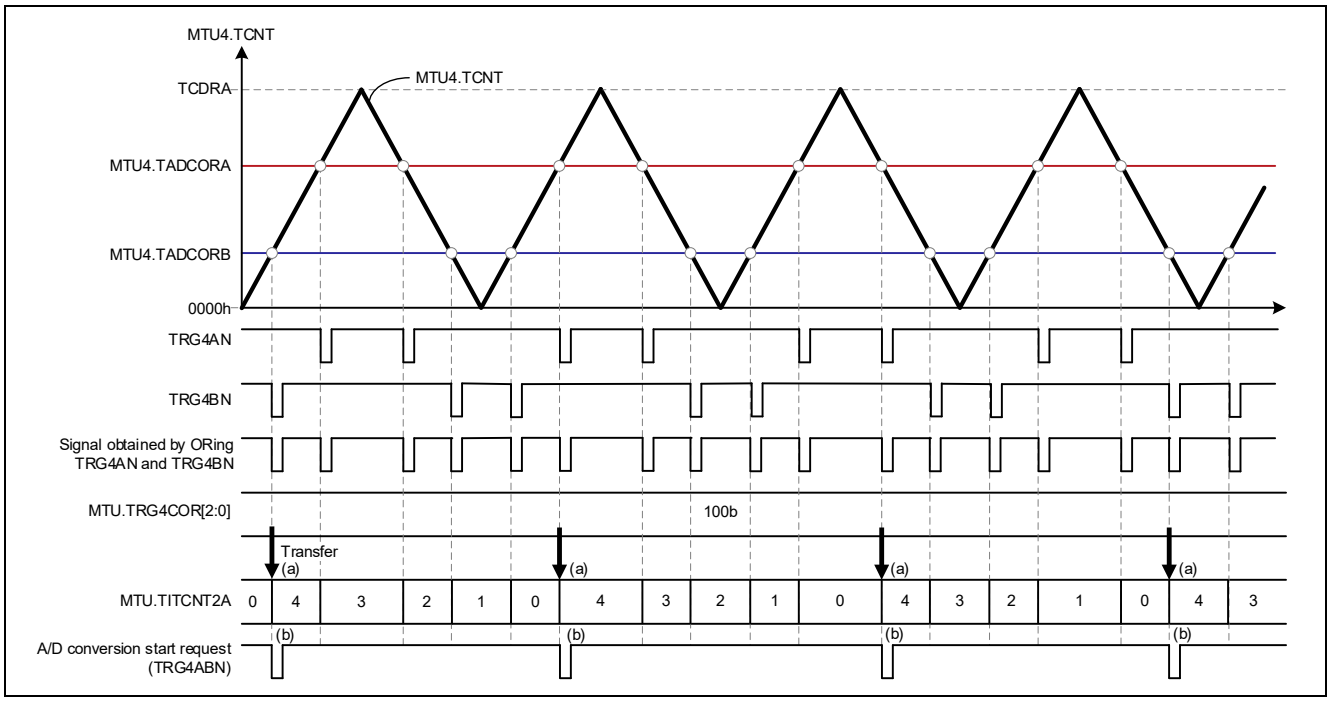


Figure 1.5 Example of A/D Conversion Start Request Signal (TRG4ABN) Operation of Interrupt Skipping Function 2
(Complementary PWM Mode, TRG4AN and TRG4BN Output Enabled During Up-/Down-Counting, Interrupt Skipping Count: 4)

When MTU3 and MTU4 start counting and a compare match occurs between the MTU4.TCNT counter and the set values of TADCORA and TADCORB, a request to start A/D conversion (TRG4AN and TRG4BN) is generated, respectively.

Bits TRG4COR[2:0] in the timer interrupt skipping setting register 2 (TITCR2A) are set to 100b, so the interrupt skipping count is 4.

The timer interrupt skipping count counter 2 (TITCNT2A) counts down from the value set by TRG4COR[2:0] bits each time TRG4AN or TRG4BN occurs, and when the counter reaches 0 and is reloaded ((a) in Figure 1.5), TRG4AN and TRG4BN interrupts are enabled and TRG4ABN is output ((b) in Figure 1.5).

(1) Register settings

To use interrupt skipping function 2, registers must be set as follows.

Table 1.17 Registers Used for Interrupt Skipping Function 2

For MTU3 and MTU4	For MTU6 and MTU7	Register Name	Description
TITMRA	TITMRB	Timer interrupt skipping mode register	Selects two types of skipping functions
TITCR2A	TITCR2B	Timer interrupt skipping setting register 2	Specifies an interrupt skipping count (0 to 7)
TITCNT2A	TITCNT2B	Timer interrupt skipping counter 2	Skipping counter

The TITCR2A and TITCR2B settings are valid only when the TITMRA.TITM and TITMRB.TITM bits are set to 1b, respectively. When the TITMRA.TITM and TITMRB.TITM bits are set to 0b, the values of TITCNT2A and TITCNT2B are cleared.

The TITCNT2A and TITCNT2B registers start counting from the values set in the TITCR2A.TRG4COR[2:0] and TITCR2B.TRG7COR[2:0] bits and the count decrements every time TRG4AN or TRG4BN (TITCNT2A) is generated or TRG7AN or TRG7BN (TITCNT2B) is generated. When the counter reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts or the TRG7AN and TRG7BN interrupts become valid and the A/D conversion start request signal (TRG4ABN, TRG7ABN) is output.

At the first MTU3 and MTU4 count start after reset release, TITCNT2A (TITCNT2B) starts counting from the initial value of 0, a reload occurs at the first TRG4AN and TRG4BN (TRG7AN and TRG7BN) output, and TRG4ABN (TRG7ABN) is output.

The skipping count counter retains its values even after MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) stops counting.

For details on interrupt skipping function 2, refer to (6) A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 2 in section 22.3.9, A/D Conversion Start Request Delaying Function, in the RX66T Group User's Manual: Hardware.

For details of each register, refer to section 22.2.41, Timer Interrupt Skipping Set Register 2m (TITCR2m) (m = A, B), and section 22.2.42, Timer Interrupt Skipping Counter 2m (TITCNT2m) (m = A, B), in the RX66T Group User's Manual: Hardware.

(2) Notes when using interrupt skipping function 2

When interrupt skipping function 2 is in use and the values in MTU4.TADCORA and MTU4.TADCORB marginally differ, skipped interrupts may not be counted correctly, in which case requests for A/D conversion may not be generated with the expected timing.

The following setting conditions apply.

For MTU6 and MTU7, the same conditions apply to the settings of MTU7.TADCORA and MTU7.TADCORB.

- When the skipping count is zero
  - The values in MTU4.TADCORA and MTU4.TADCORB must differ by at least four.
  - The MTU4.TADCORA comparison interval must be at least four cycles of PCLKC clock.
    (The updated value of MTU4.TADCORA is set to the previous value plus four or more, or minus four or less.)
  - The MTU4.TADCORB comparison interval must be at least four cycles of PCLKC clock.
    (The updated value of MTU4.TADCORB is set to the previous value plus four or more, or minus four or less.)
- When the skipping count is one or more
  - The values in MTU4.TADCORA and MTU4.TADCORB must differ by at least two.
  - The MTU4.TADCORB comparison interval must be at least two cycles of PCLKC clock.
    (The updated value of MTU4.TADCORB is set to the previous value plus two or more, or minus two or less.)

For details, refer to section 22.6.22, Interrupt Skipping Function 2, in the RX66T Group User's Manual: Hardware.

1.3.3 Interrupt Skipping Function by Using the GTITC Register (GPTW)

In the GPTW, for the GTCNT counter overflow (GTPR register's compare match) interrupt (GTCIV) and underflow interrupt (GTCIU), up to seven interrupts can be skipped by setting the general PWM timer interrupt and A/D conversion start request skipping setting register (GTITC). This function is called GTCIV/GTCIU interrupt skipping function. A/D conversion start requests can be skipped in coordination with the GTCIV/GTCIU skipping function by setting the GTITC register.

The interrupt skipping function by using the GTITC register operates independently of the extended interrupt skipping function.

The figure below shows an example of A/D conversion start request signal A operation of the interrupt skipping function by the GTITC register when A/D conversion start request signal A output is enabled during GTCNT up-counting and down-counting in triangle-wave PWM mode.

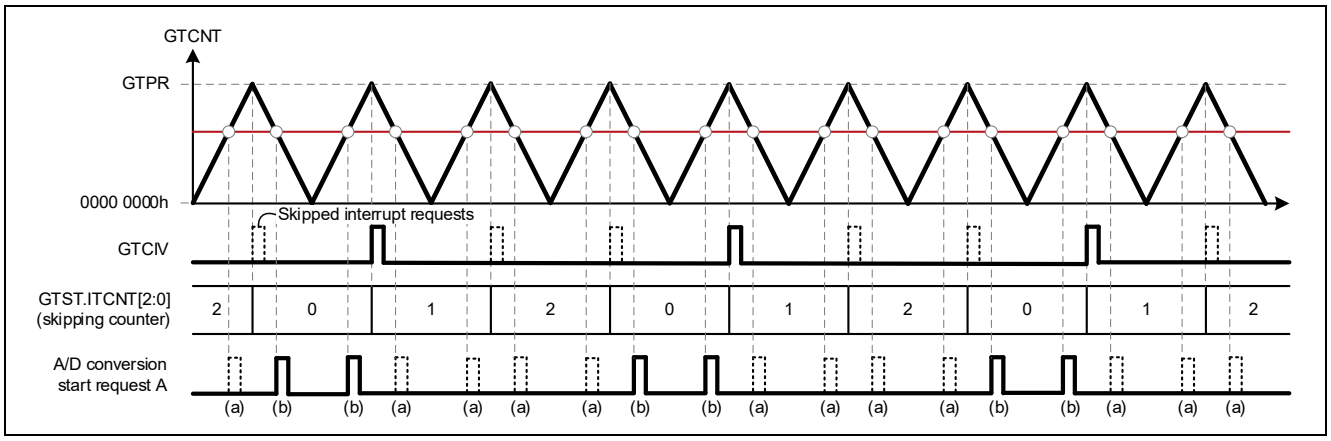


Figure 1.6 Operation Example of A/D Conversion Start Request Signal A of the Interrupt Skipping Function by Using the GTITC Register (Triangle-Wave PWM Mode, A/D Conversion Start Request Signal A Output Enabled During Up-/Down-Counting When ITCNT Counts Crests with the Interrupt Skipping Count of 2)

Bits IVTC[1:0] in the GTITC register are set to 01b, so count-up operation is performed when a GTCIV interrupt is generated.

Bits IVTT[2:0] in the GTITC register are set to 010b, so the interrupt skipping count is 2. While the interrupt skipping counter value is 000b, the output of A/D conversion start request signal A is enabled ((b) in Figure 1.6). While the counter value is not 000b, the output of A/D conversion start request signal A is disabled ((a) in Figure 1.6).

For details on the interrupt skipping function by the GTITC register, refer to 24.4.3.1, Interrupt Skipping Function by GTITC Register, in the RX66T Group User's Manual: Hardware.

(1) Register settings

To use the interrupt skipping function by using the GTITC register, registers must be set as follows.

Table 1.18 Relationship Between the Interrupt Skipping Function and GTITC Register Settings

Symbol	Description
IVTC[1:0]	0 0: Skipping is not performed. 0 1: Both overflows and underflows for sawtooth waves and crests for triangle waves are counted and interrupts are skipped. 1 0: Both overflows and underflows for sawtooth waves and troughs for triangle waves are counted and interrupts are skipped. 1 1: Both overflows and underflows for sawtooth waves and both crests and troughs for triangle waves are counted and interrupts are skipped.
IVTT[2:0]	Specify a skipping count within the range from 0 to 7. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed.
ADTAL	Specifies whether to link A/D conversion start request A with the GTCIV/GTCIU interrupt skipping function.
ADTBL	Specifies whether to link A/D conversion start request B with the GTCIV/GTCIU interrupt skipping function.

In event count operation (at least one bit in the GTUPSR or GTDNSR register is set to 1b), the settings of the GTITC register are invalid.

The interrupt skipping function by using the GTITC register operates independently of the extended interrupt skipping function by using the general PWM timer extended interrupt skipping counter control register (GTEITC).

For details, refer to section 24.2.18, General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register (GTITC), in the RX66T Group User's Manual: Hardware.

The number of interrupts skipped by using the GTITC register can be read from the ITCNT[2:0] bits of the general PWM timer status register (GTST).

When the interrupt skipping function by the GTITC register is used (the GTITC.IVTC[1:0] bits are set to a value other than 00b), the ITCNT[2:0] bit value is incremented by one every time the GTCIV/GTCIU interrupt source that is selected in the IVTC[1:0] bits is generated.

While the counting operation of GTCNT is stopped, the ITCNT[2:0] bits are set to 000b.

For details, refer to section 24.2.16, General PWM Timer Status Register (GTST), in the RX66T Group User's Manual: Hardware.

(2) Notes when linking with the interrupt skipping function by using the GTITC register

Note the following when linking with the interrupt skipping function by using the GTITC register:

- The interrupt skipping function only depends on the setting of GTITC register and is independent of the setting of interrupt enable bits in the GTINTAD register.
  It also operates independently of the extended interrupt skipping function by the GTEITC register.
- When both troughs and crests are counted for interrupt skipping in triangle-wave PWM mode, if the number of times of skipping is set to an odd value, GTCIV/GTCIU interrupt requests cannot be generated at troughs only or at crests only depending on the skipping counter start timing.
  Therefore, in order to count both troughs and crests for interrupt skipping and generate the GTCIV/GTCIU interrupts at troughs only or crests only in triangle-wave PWM mode, the number of times of skipping should be even.

  The figure below shows an example of operation with both troughs and crests counted for interrupt skipping in triangle-wave PWM mode, and with the interrupt skipping count set to 3. Under the following conditions, all GTCIV interrupts are skipped (not generated) ((a) in Figure 1.7). Therefore, the A/D conversion start request signal A, which is to be generated during up-counting, is also not generated ((b) in Figure 1.7).

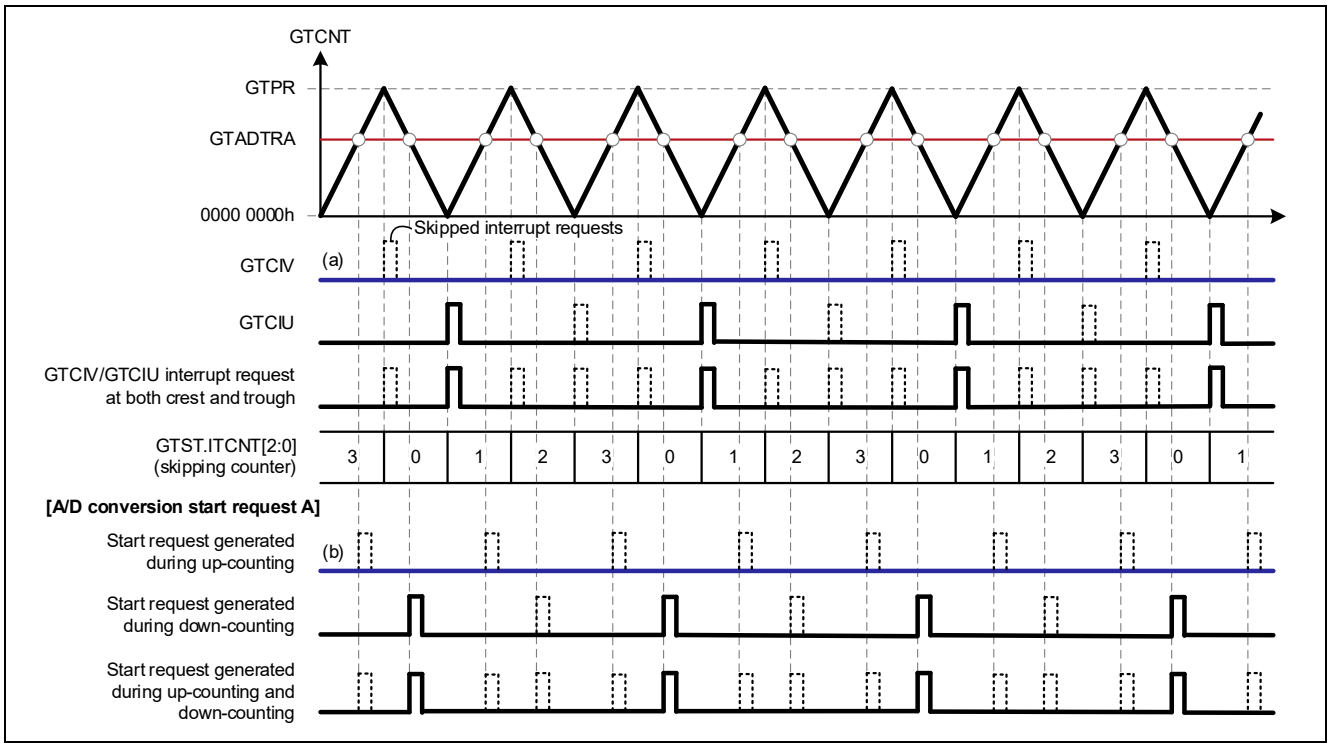


Figure 1.7 Operation Example of A/D Conversion Start Request Signal A Linked with the Interrupt Skipping Function by Using the GTITC Register (Triangle-Wave PWM Mode, A/D Conversion Start Request Signal A Output Enabled During Up-/Down-Counting When ITCNT Counts Both Troughs and Crests with the Interrupt Skipping Count of 3)

- When both overflows and underflows are counted for interrupt skipping with the count direction changed in sawtooth-wave PWM mode, GTCIV/GTCIU interrupt requests may not be generated at overflows only or at underflows only.
  Therefore, to count both overflows and underflows for interrupt skipping with the count direction changed and generate GTCIV/GTCIU interrupts at overflows only or underflows only in sawtooth-wave PWM mode, the skipping state should be carefully checked before using.

  The figure below shows an example of operation with both overflows and underflows counted for interrupt skipping with the count direction changed in sawtooth-wave PWM mode, and the interrupt skipping count set to 4. Under the following conditions, no GTCIV interrupts may be generated ((a) in Figure 1.8).

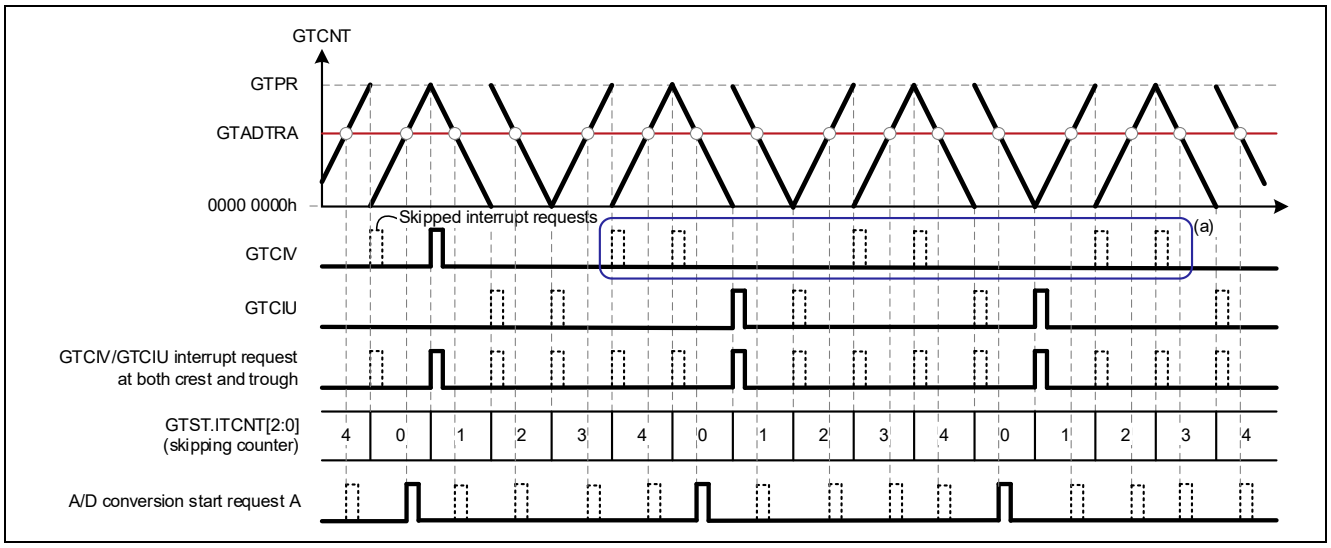


Figure 1.8 Operation Example of A/D Conversion Start Request Signal A Linked with the Interrupt Skipping Function by Using the GTITC Register (Operation with the Count Direction Changed in Sawtooth-Wave PWM Mode, A/D Conversion Start Request Signal A Output Enabled During Up-/Down-Counting When ITCNT Counts both Troughs and Crests with the Interrupt Skipping Count of 4)

- When changing the skipping count, make sure that you release the skipping count setting (GTITC.IVTC[1:0] bits = 00b).

For details, refer to section 24.4.3.1, Interrupt Skipping Function by GTITC Register, in the RX66T Group User's Manual: Hardware.

1.3.4 Extended Interrupt Skipping Function (GPTW)

Overflow/underflow interrupts, compare match/input capture interrupts, A/D conversion start requests, and buffer transfers can be skipped by counting the GTCNT counter overflows or underflows based on settings of the general PWM timer extended interrupt skipping counter control register (GTEITC), general PWM timer extended interrupt skipping setting register 1 (GTEITLI1), general PWM timer extended interrupt skipping setting register 2 (GTEITLI2), and general PWM timer extended buffer transfer skipping setting register (GTEITLB). A dead time error interrupt cannot be skipped.

This function is called the extended skipping function and operates independently of the interrupt skipping function by the GTITC register.

Whether interrupts are skipped and the skipping period (duration during which interrupts are skipped) can be set separately for the A/D conversion start request in the GTEITLI2 register and for the buffer transfer in the GTEITLB register. For example, the buffer transfer that also outputs A/D conversion start requests can be set in such a way that the A/D conversion start request is skipped once while the buffer transfer is skipped twice.

The skipping period depends on the operation of two independent extended interrupt skipping counters (GTEITC.EITCNT1[3:0] and EITCNT2[3:0] bits) and the settings of the extended interrupt skipping setting registers (GTEITLI2 and GTEITLB). For details, refer to (3), Skipping period.

The figure below shows an operation example of A/D conversion start request signal A of the extended interrupt skipping function and buffer transfer of the GTADTRA register when A/D conversion start request signal A output is enabled during GTCNT up-counting and down-counting in triangle-wave PWM mode.

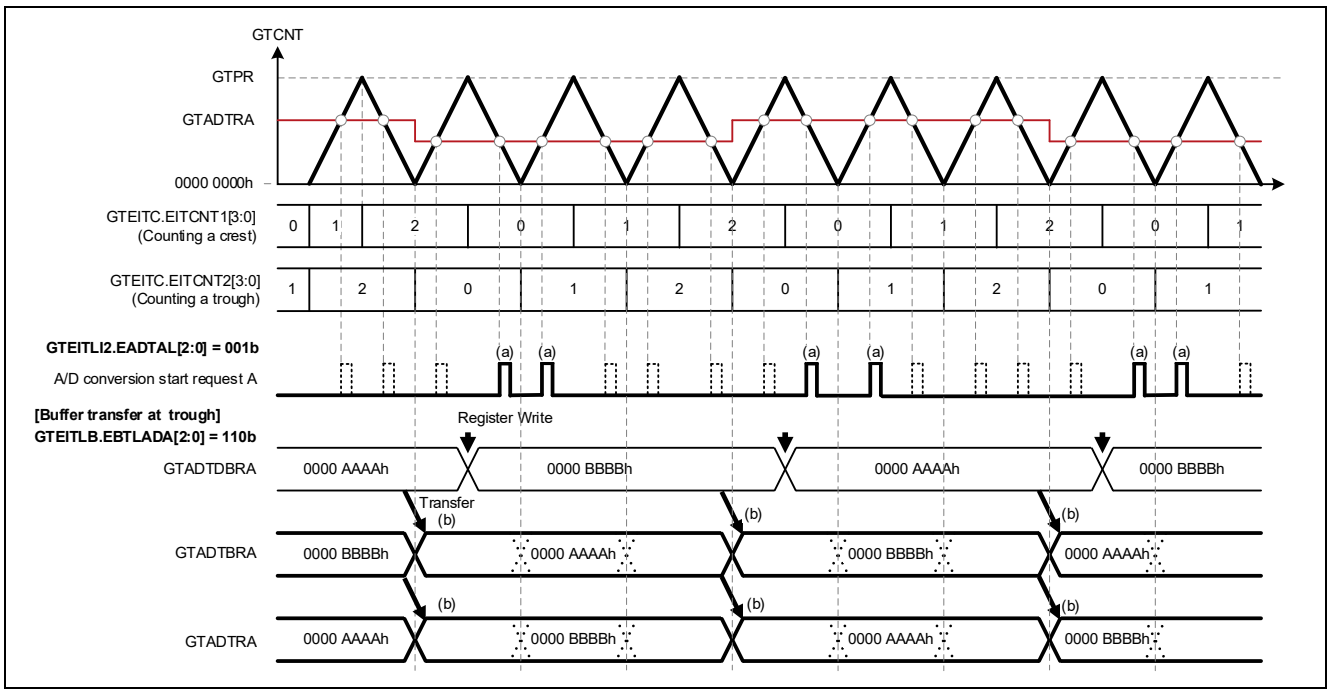


Figure 1.9 Operation Example of A/D Conversion Start Request Signal A of the Extended Interrupt Skipping Function
(Triangle-Wave PWM Mode, A/D Conversion Start Request Signal A Output Enabled During Up-/Down-Counting When EITCNT1 Counts Crests with the Interrupt Skipping Count of 2 and EITCNT2 Counts Troughs with the Interrupt Skipping Count of 2 When the Initial Value is 1)

EITCNT1 counts crests with the interrupt skipping count of 2, and EITCNT2 counts troughs with the interrupt skipping count of 2.

A/D conversion start request signal A is output in the period when EITCNT1 = 0 because the skipping period (GTEITLI2.EADTAL) is set 001b ((a) in Figure 1.9). Buffer transfers of the GTADTRA register are performed in a period when EITCNT2 = 2 (skipping count), since skipping period (GTEITLB.EBTLADA) is set to 110b ((b) in Figure 1.9).

For details on the extended interrupt skipping function, refer to 24.4.3.2, Extended Interrupt Skipping Function, in the RX66T Group User's Manual: Hardware.

(1) Register settings

To use the extended interrupt skipping function, registers must be set as follows.

Table 1.19 Registers Used for the Extended Interrupt Skipping Function

Symbol	Register Name	Description
GTEITC	General PWM timer extended interrupt skipping counter control register	Specifies the count source and interrupt skipping count of the skipping counter
GTEITLI2	General PWM timer extended interrupt skipping setting register 2	Selects the extended skipping function for A/D conversion start requests
GTEITLB	General PWM timer extended buffer transfer skipping setting register	Selects the extended skipping function for buffer transfers

In event count operation (at least one bit in the GTUPSR or GTDNSR register is set to 1b), the settings of the GTEITC, GTEITLI2, and GTEITLB registers are invalid. The extended interrupt skipping function cannot be used.

To skip A/D conversion start requests and buffer transfers, in addition to setting GTEITC, set A/D conversion start requests and buffer transfers to be skipped individually in GTEITLI2 and GTEITLB.

Buffer transfer skipping by the GTEITLB register is performed in all the buffer operations which are enabled in the GTBER and GTDTCR registers, or all buffer operations performed by sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3.

For details, refer to section 24.2.33, General PWM Timer Extended Interrupt Skipping Counter Control Register (GTEITC), section 24.2.35, General PWM Timer Extended Interrupt Skipping Setting Register 2 (GTEITLI2), and section 24.2.36, General PWM Timer Extended Buffer Transfer Skipping Setting Register (GTEITLB), in the RX66T Group User's Manual: Hardware.

(2) Counter operation

Two independent skipping counters are available for the extended interrupt skipping function. There is also one counter, independent of these two counters, used in the interrupt skipping function by the GTITC register. The table below shows the differences in the specifications of the two interrupt skipping counters of the extended interrupt skipping function and the interrupt skipping counter of the interrupt skipping function by the GTITC register.

Table 1.20 Interrupt Skipping Counter Specification Differences

Item	Extended Interrupt Skipping Counter 1	Extended Interrupt Skipping Counter 2	Counter of Interrupt Skipping by the GTITC Register
Count source setting bits	GTEITC.EIVTC1[1:0]	GTEITC.EIVTC2[1:0]	GTITC.IVTC[1:0]
Count source type	0 0: Skipping is not performed. 0 1: Both overflows and underflows for sawtooth waves and crests for triangle waves are counted and interrupts are skipped. 1 0: Both overflows and underflows for sawtooth waves and troughs for triangle waves are counted and interrupts are skipped. 1 1: Both overflows and underflows for sawtooth waves and both crests and troughs for triangle waves are counted and interrupts are skipped.		
Skipping count setting bits	GTEITC.EIVTT1[3:0]	GTEITC.EIVTT2[3:0]	GTITC.IVTT[2:0]
Skipping count	0 to 15		0 to 7
Initial bit setting bit	—	GTEITC.EITCNT2IV[3:0]	—
Counter initial value	0	GTEITC.EITCNT2IV[3:0] (Setting value)	0
Counter	GTEITC.EITCNT1[3:0]	GTEITC.EITCNT2[3:0]	GTST.ITCNT[2:0]
Count direction	Up-counting		
When GTCNT counting stops	Value retained		Reset to 0

Figure 1.10 shows an operation example of the extended interrupt skipping counter and the counter of the interrupt skipping function by using the GTITC register.

Extended interrupt skipping counter 1 (EITCNT1) counts crests (EIVTC1[1:0] = 01b) and the interrupt skipping count is set to 2 (EIVTT1[3:0] = 0010b). Extended interrupt skipping counter 2 (EITCNT2) counts troughs (EIVTC2[1:0] = 10b), the interrupt skipping count is set to 2 (EIVTT2[3:0] = 0010b), and the initial value is set to 1 (EITCNT2IV[3:0] = 0001b). The GTITC register-based interrupt skipping counter (GTST.ITCNT[2:0]) counts crests (IVTC1[1:0] = 01b) and the interrupt skipping count is set to 2 (IVTT[2:0] = 010b). Each counter counts up according to the set source, and restarts counting from 0 when the set skipping count is reached.

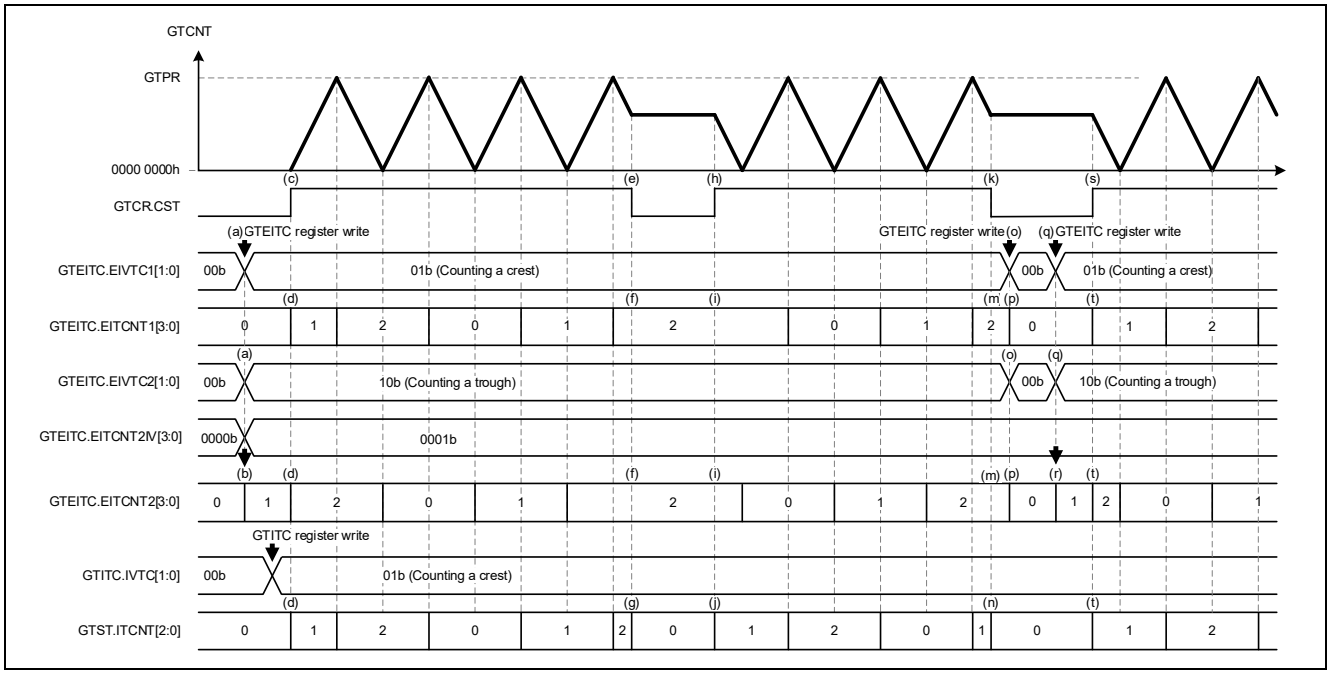


Figure 1.10 Example of Interrupt Skipping Counter Operation (Triangle-Wave PWM Mode, EITCNT1: Counts Crests with the Skipping Count of 2, EITCNT2: Counts Troughs with the Skipping Count of 2, Initial Value Set to 1, ITCNT: Counts Crests with the Skipping Count of 2)

- Before GTCNT starts counting:
  Initial value of EITCNT1: 0
  Initial value of EITCNT2:
  Reflects the written value of EITCNT2IV[3:0] ((b) in Figure 1.10) when writing to the GTEITC register ((a) in Figure 1.10).
  Initial value of ITCNT: 0
- Start GTCNT counting ((c) in Figure 1.10).
  EIVTCNT1, EIVTCNT2, and ITCNT starts up-counting ((d) in Figure 1.10).
- Stop GTCNT counting ((e) in Figure 1.10).
  EIVTCNT1 and EIVTCNT2: Retain their values ((f) in Figure 1.10).
  ITCNT: Reset to 000b ((g) in Figure 1.10)
- Start GTCNT counting ((h) in Figure 1.10).
  EIVTCNT1 and EIVTCNT2: Start up-counting from the retained values ((i) in Figure 1.10).
  ITCNT: Starts up-counting ((j) in Figure 1.10).
- Stop GTCNT counting ((k) in Figure 1.10).
  EIVTCNT1 and EIVTCNT2: Retain their values ((m) in Figure 1.10).
  ITCNT: Reset to 000b ((n) in Figure 1.10)
- Change EIVTCNT1 and EIVTCNT2 to settings not to skip (EIVTC1[1:0] = EIVTC2[1:0] = 00b) ((o) in Figure 1.10).
  Reset EIVTCNT1 and EIVTCNT2 to 0000b ((p) in Figure 1.10).
- Change EIVTCNT1 and EIVTCNT2 back to settings to skip ((q) in Figure 1.10).
  Set EIVTCNT2 to the initial value ((r) in Figure 1.10).
- Start GTCNT counting ((s) in Figure 1.10).
  EIVTCNT1, EIVTCNT2, and ITCNT start up-counting ((t) in Figure 1.10).

(3) Skipping period

The skipping period during which A/D conversion start requests are skipped can be selected by setting bits EADTmL[2:0] (m = A, B) in the GTEITLI2 register. Similarly, the period during which the GTADTRm register buffer transfers (among the GTADTRm, GTADTBRm, and GTADTDBRm registers) are skipped can be selected by setting bits EBTLADm[2:0] (m = A, B) in the GTEITLB register.

Table 1.21 Differences in Skipping Period

EADTmL[2:0], EBTLADm[2:0]	Function
0 0 0	Does not perform extended interrupt skipping.
0 0 1	Skips A/D conversion start requests or buffer transfers during the period when the value of extended interrupt skipping counter 1 is not 0. (A/D conversion start requests or buffer transfers are performed while EITCNT1 = 0.)
0 1 0	Skips A/D conversion start requests or buffer transfers during the period when the value of extended interrupt skipping counter 2 is not 0. (A/D conversion start requests or buffer transfers are performed while EITCNT2 = 0.)
0 1 1	Skips A/D conversion start requests or buffer transfers during the period when the value of extended interrupt skipping counter 1 or 2 is not 0. (A/D conversion start requests or buffer transfers are performed while EITCNT1 = EITCNT2 = 0.)
1 0 0	Setting prohibited
1 0 1	Skips A/D conversion start requests or buffer transfers during the period when the value of extended interrupt skipping counter 1 is not the skipping count. (A/D conversion start requests or buffer transfers are performed while EITCNT1 = EIVTT1.)
1 1 0	Skips A/D conversion start requests or buffer transfers during the period when the value of extended interrupt skipping counter 2 is not the skipping count. (A/D conversion start requests or buffer transfers are performed while EITCNT2 = EIVTT2.)
1 1 1	Skips A/D conversion start requests or buffer transfers during the period when the value of extended interrupt skipping counter 1 or 2 is not the skipping count. (A/D conversion start requests or buffer transfers are performed while EITCNT1 = EIVTT1 and EITCNT2 = EIVTT2.)

Even if GTEITLI2 and GTEITLB are set, interrupts are not skipped in the following cases:

- Even if GTEITLI2 or GTEITLB is set, skipping is not performed if the target skipping counter is set to not count (EIVTCk[1:0] = 00b or EIVTTk[3:0] = 0000b) (k = 1, 2).
- When EADTmL[2:0] or EBTLADm[2:0] (m = A, B) is set to 011b or 111b, if either skipping counter is set to not count (EIVTCk [1:0] = 00b or EIVTTk[3:0] = 0000b) (k = 1, 2), skipping is not performed.

Figure 1.11 shows the differences in the skipping period of the extended interrupt skipping function.

EITCNT1 counts crests (EIVTC1[1:0] = 01b) and the interrupt skipping count is set to 2 (EIVTT1[3:0] = 0010b). EITCNT2 counts troughs (EIVTC2[1:0] = 10b), the interrupt skipping count is set to 2 (EIVTT2[3:0] = 0010b), and the initial value is set to 0 (EITCNT2IV[3:0] = 0000b). Each counter counts up according to the set source, and restarts counting from 0 when the set skipping count is reached.

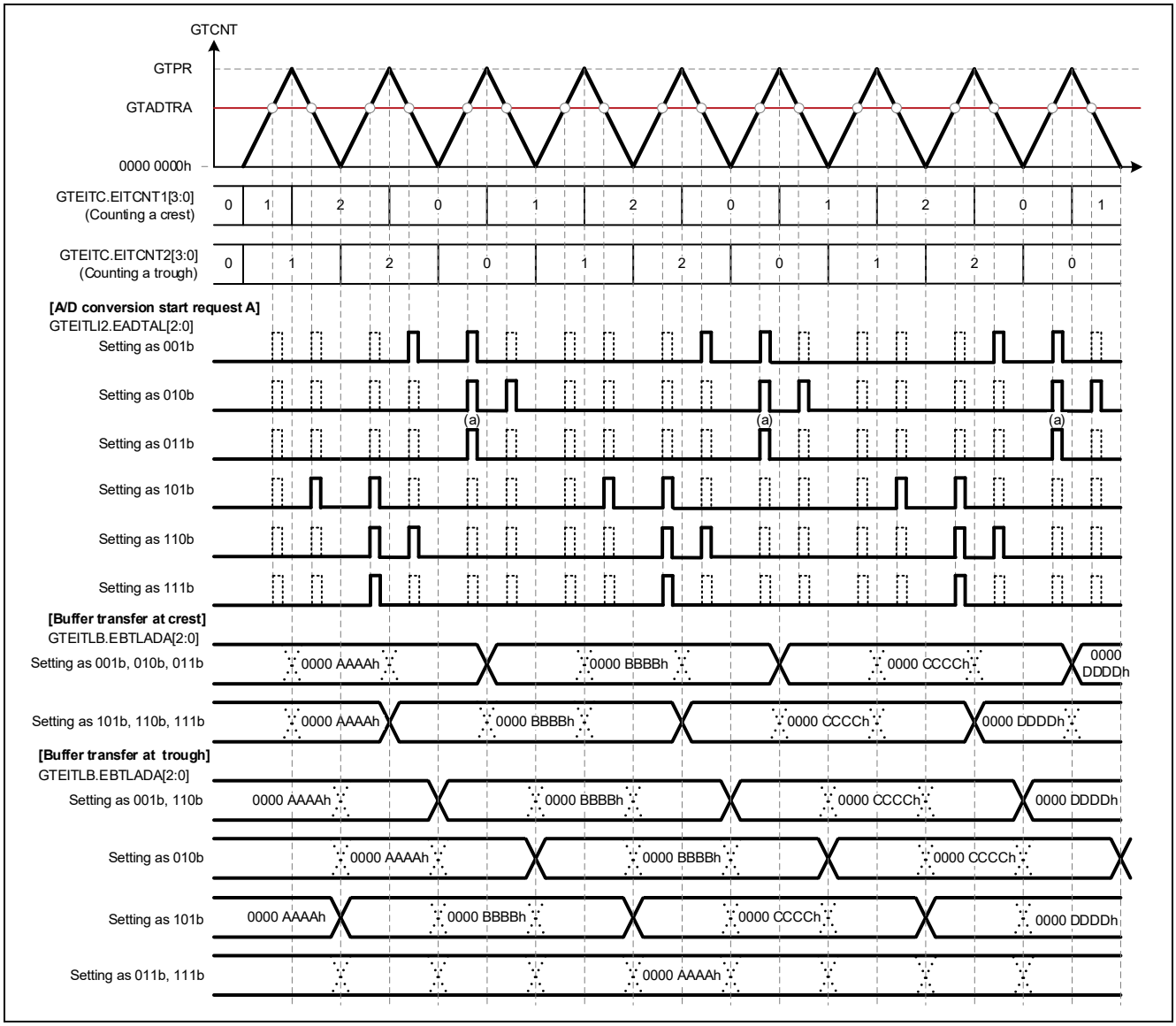


Figure 1.11 Operation Example of the Extended Interrupt Skipping Function (Triangle-Wave PWM Mode, A/D Conversion Start Request Signal A Output Enabled During Up-/Down-Counting, EITCNT1: Counts Crests with the Interrupt Skipping Count of 2, EITCNT2: Counts Troughs with the Interrupt Skipping Count of 2 and the Initial Value as 0)

As representative examples, the following two operations are described.

- Operation of A/D conversion start request A when GTEITLI2.EADTAL[2:0] = 011b
  A/D conversion start request A is output in the period when GTEITC.EITCNT1[3:0] = 0 and GTEITC.EITCNT2[3:0] = 0 ((a) in Figure 1.11).

- Operation of buffer transfer at troughs when GTEITLB.EBTLADA[2:0] = 111b
  Buffer transfer is supposed to be performed at troughs in the period when GTEITC.EITCNT1[3:0] = 2 and GTEITC.EITCNT2[3:0] = 2, but no buffer transfer is performed because there is no state that matches the condition.

(4) Notes when using the extended interrupt skipping function

- When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either the EIVTC1[1:0] bits or EIVTC2[1:0] bits to 00b).
- When A/D conversion start request skipping which can be set by the GTEITLI2 register is performed, the ELC event output depends on the A/D conversion start request enable bit in the GTINTAD register. Any operation by A/D conversion start request that is set to disabled in the GTINTAD register is not performed.

(5) Simultaneous use of the extended interrupt skipping function and the interrupt skipping function by using the GTITC register

The extended interrupt skipping function of A/D conversion start requests by using the GTEITLI2 can be used simultaneously with the interrupt skipping function by using the GTITC register. In this case, the respective skipping periods are ORed for the skipping period.

The figure below shows corresponding interrupt skipping operations by different registers are performed simultaneously.

The GTITC register-based interrupt skipping counter (GTST.ITCNT[2:0]) counts crests (IVTC1[1:0] = 01b) and the interrupt skipping count is set to 1 (IVTT[2:0] = 001b). Extended interrupt skipping counter 2 (EITCNT2) counts crests (EIVTC2[1:0] = 01b), the interrupt skipping count is set to 2 (EIVTT2[3:0] = 0010b), and the initial value is set to 1 (EITCNT2IV[3:0] = 0001b). Each counter counts up according to the set source, and restarts counting from 0 when the set skipping count is reached.

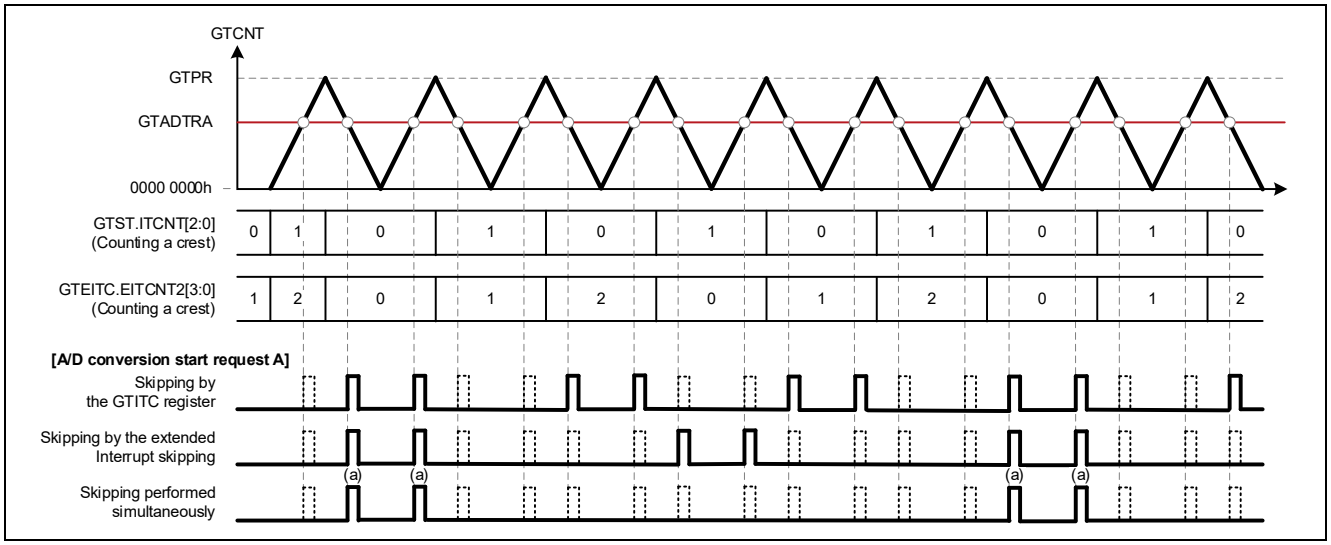


Figure 1.12 Operation Example of Simultaneous Use of Interrupt Skipping Counters (Triangle-Wave PWM Mode, A/D Conversion Start Request Signal A Output Enabled During Up-/Down-Counting, ITCNT: Counts Crests with the Interrupt Skipping Count of 1, EITCNT2: Counts Crests with the Interrupt Skipping Count of 2 and the Initial Value as 1)

A/D conversion start request A is output in the period when GTST.ITCNT[2:0] = 0 and GTEITC.EITCNT2[3:0] = 0 ((a) in Figure 1.12).

1.4 A/D Conversion Start Request Signal, and Settings of the A/D Converter and Event Link Controller (ELC)

The MTU outputs synchronous triggers to the A/D converter. In contrast, the GPTW does not output synchronous triggers to the A/D converter. To output synchronous triggers to the A/D converter using the GPTW, signals must be routed through the ELC. Therefore, to use GPTW output signals as A/D conversion start triggers, ELC settings are required.

The figure below shows the configuration.

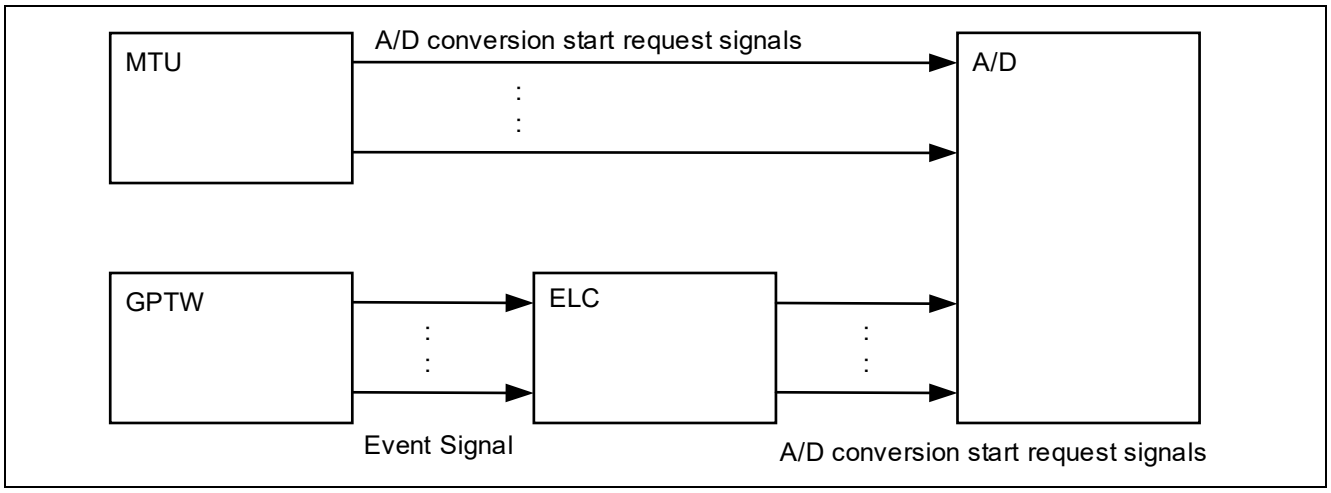


Figure 1.13 Configuration of the MTU/GPTW and A/D Converter

When using the A/D conversion start request delaying function of the MTU or the A/D conversion start request function of the GPTW, set the ADCSR.TRGE bit to 1b and the ADCSR.EXTRG bit to 0b so that the 12-bit A/D converter (S12ADH) should start A/D conversion by a synchronous trigger.

For the MTU, set the A/D conversion start conditions as shown in Table 1.22. For the GPTW, set them as shown in Table 1.23.

Table 1.22 Synchronous Triggers for the 12-Bit A/D Converter (MTU)

Item	A/D Converter Startup Source			MTU Setting
	S12AD	S12AD1	S12AD2	
Compare match between MTU4.TADCORA and MTU4.TCNT	TRG4AN			When using the A/D conversion start request delaying function or interrupt skipping function 1
Compare match between MTU4.TADCORB and MTU4.TCNT	TRG4BN			
Compare match between MTU4.TADCORA and MTU4.TCNT, or between MTU4.TADCORB and MTU4.TCNT	TRG4AN or TRG4BN			
Compare match between MTU4.TADCORA and MTU4.TCNT, and between MTU4.TADCORB and MTU4.TCNT	TRG4ABN			When using interrupt skipping function 2
Compare match between MTU7.TADCORA and MTU7.TCNT	TRG7AN			When using the A/D conversion start request delaying function or interrupt skipping function 1
Compare match between MTU7.TADCORB and MTU7.TCNT	TRG7BN			
Compare match between MTU7.TADCORA and MTU7.TCNT, or between MTU7.TADCORB and MTU7.TCNT	TRG7AN or TRG7BN			
Compare match between MTU7.TADCORA and MTU7.TCNT, and between MTU7.TADCORB and MTU7.TCNT	TRG7ABN			When using interrupt skipping function 2

Table 1.23 Synchronous Triggers for the 12-Bit A/D Converter (GPTW)

Item	A/D Converter Startup Source			GPTW Setting
	S12AD	S12AD1	S12AD2	
ELC triggers	ELCTRG00N	ELCTRG10N	ELCTRG20N	When using A/D conversion start request A/B (including the skipping function)
	ELCTRG01N	ELCTRG11N	ELCTRG21N	
	ELCTRG00N or ELCTRG01N	ELCTRG10N or ELCTRG11N	ELCTRG20N or ELCTRG21N	

Note that the issuance interval of triggers for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}) of the A/D converter for both the MTU and GPTW. For details, refer to section 38.2.15, A/D Conversion Start Trigger Select Register (ADSTRGR), in the RX66T Group User's Manual: Hardware.

When using the A/D conversion start request function of the GPTW, set the event link setting register n (ELSRn) and ELS[7:0] bits in the ELC as follows:

Table 1.24 Differences in Event Signals Depending on the ELSRn Register and ELS[7:0] Bit Settings

ELSRn Register	Value of the ERSRn.ELS[7:0] Bits	Event Signal Set in ELSRn
ELSR15: S12AD (ELCTRG00N) ELSR45: S12AD1 (ELCTRG10N) ELSR46: S12AD2 (ELCTRG20N) ELSR56: S12AD (ELCTRG01N) ELSR57: S12AD1 (ELCTRG11N) ELSR58: S12AD2 (ELCTRG21N)	50h	GPTW0 A/D conversion start request A
	51h	GPTW0 A/D conversion start request B
	5Ah	GPTW1 A/D conversion start request A
	5Bh	GPTW1 A/D conversion start request B
	64h	GPTW2 A/D conversion start request A
	65h	GPTW2 A/D conversion start request B
	6Eh	GPTW3 A/D conversion start request A
	6Fh	GPTW3 A/D conversion start request B
	78h	GPTW4 A/D conversion start request A
	79h	GPTW4 A/D conversion start request B
	82h	GPTW5 A/D conversion start request A
	83h	GPTW5 A/D conversion start request B
	8Ch	GPTW6 A/D conversion start request A
	8Dh	GPTW6 A/D conversion start request B
	96h	GPTW7 A/D conversion start request A
	97h	GPTW7 A/D conversion start request B
	A0h	GPTW8 A/D conversion start request A
	A1h	GPTW8 A/D conversion start request B
	Aah	GPTW9 A/D conversion start request A
	ABh	GPTW9 A/D conversion start request B

2. Operation Confirmation Conditions

The operation of the sample code described in this application note has been confirmed under the conditions listed in the table below.

Table 2.1 Operation Confirmation Environments

Item	Description
MCU	R5F566TEADFP (included in Renesas Starter Kit for RX66T)
Operating frequency	Main clock: 8 MHz PLL: 160 MHz (main clock x 1/1 x 20) HOCO: Stopped LOCO: Stopped System clock (ICLK) 160 MHz (PLL x 1/1) Peripheral module clock A (PCLKA): 80 MHz (PLL x 1/2) Peripheral module clock B (PCLKB): 40 MHz (PLL x 1/4) Peripheral module clock C (PCLKC): 160 MHz (PLL x 1/1) Peripheral module clock D (PCLKD): 40 MHz (PLL x 1/4) FlashIF clock (FCLK): 40 MHz (PLL x 1/4)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics e² studio Version 2022-10
C compiler*1	Renesas Electronics C/C++ Compiler Package for RX Family V3.04.00
	Compiler options The integrated development environment default settings are used.
RX Smart Configurator	V2.15.0
Board support package (r_bsp)	V7.20
Endian	Little endian
Operating 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: 1. 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. MTU Sample Codes

3.1 Common

3.1.1 Sample Code List

This application note provides the following sample codes created with the Smart Configurator.

Sample codes can be downloaded from the Renesas Electronics website.

Table 3.1 MTU Sample Code List

Name	Sample Code Usage Conditions	Ref.
A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 1 r01an6643_rx66t_mtu3_ad_delay_1.zip	• Complementary PWM mode 2 • Single buffer	3.2
A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 2 r01an6643_rx66t_mtu3_ad_delay_2.zip	• Complementary PWM mode 2 • Single buffer	3.3

3.1.2 Folder Structure

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

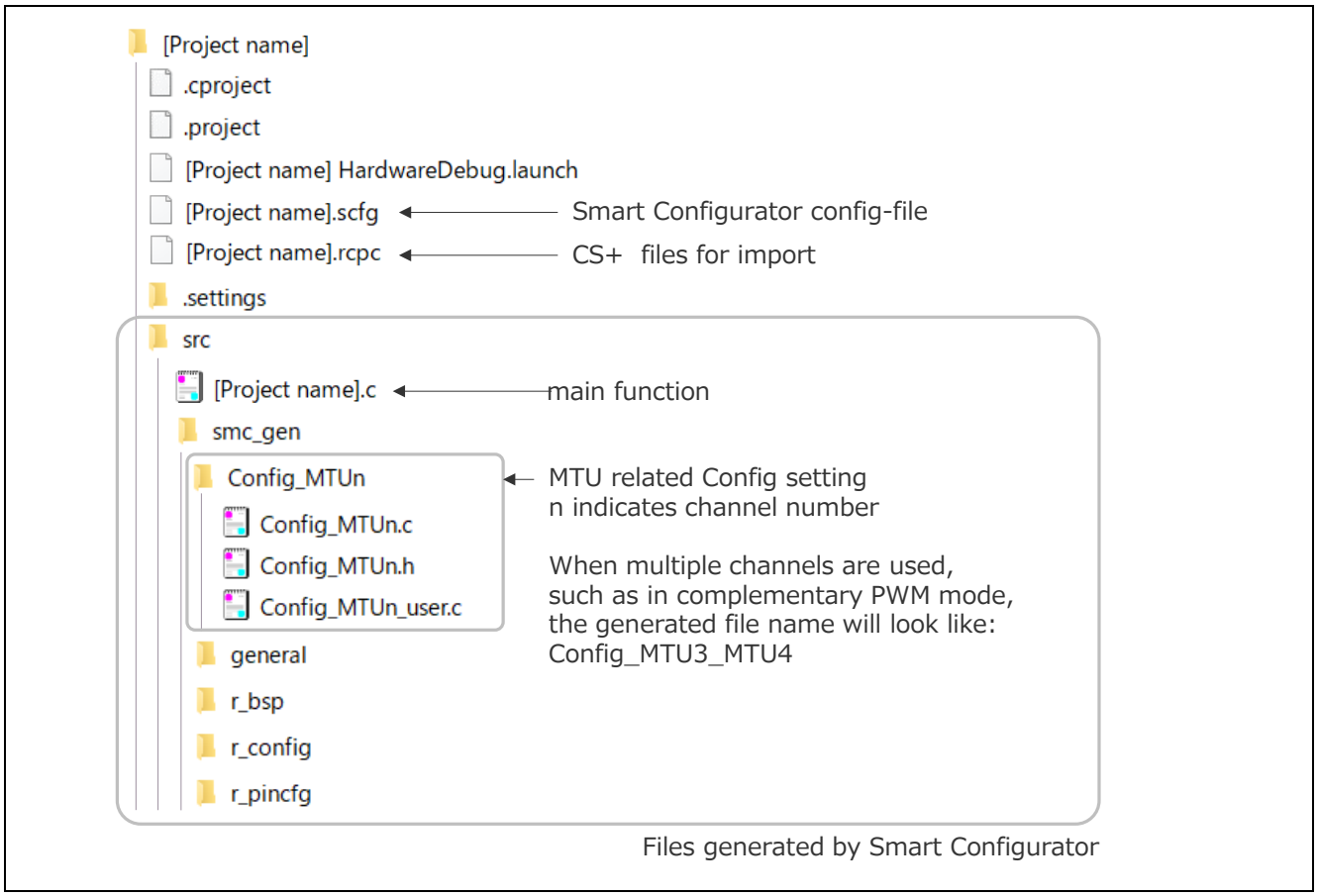


Figure 3.1 MTU Folder Structure

3.1.3 File Structure

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

Table 3.2 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*1	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*1	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*1	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, make sure that you include the r_smc_entry.h file.

Note: 1. n indicates a channel number.

3.1.4 Adding Components

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

Table 3.3 Adding Components

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

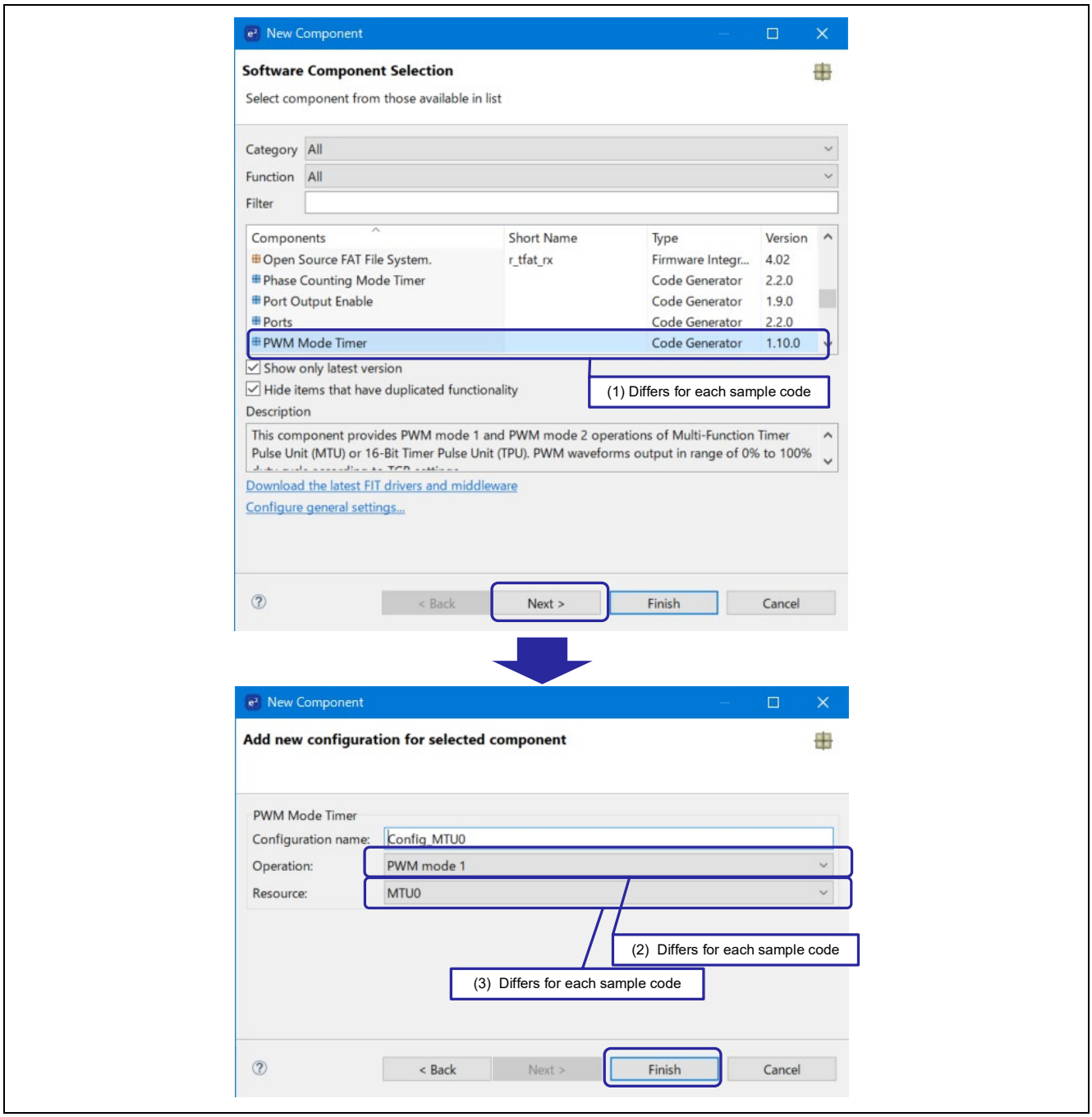


Figure 3.2 Adding Components

3.1.5 Pin Settings

Figure 3.3 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.

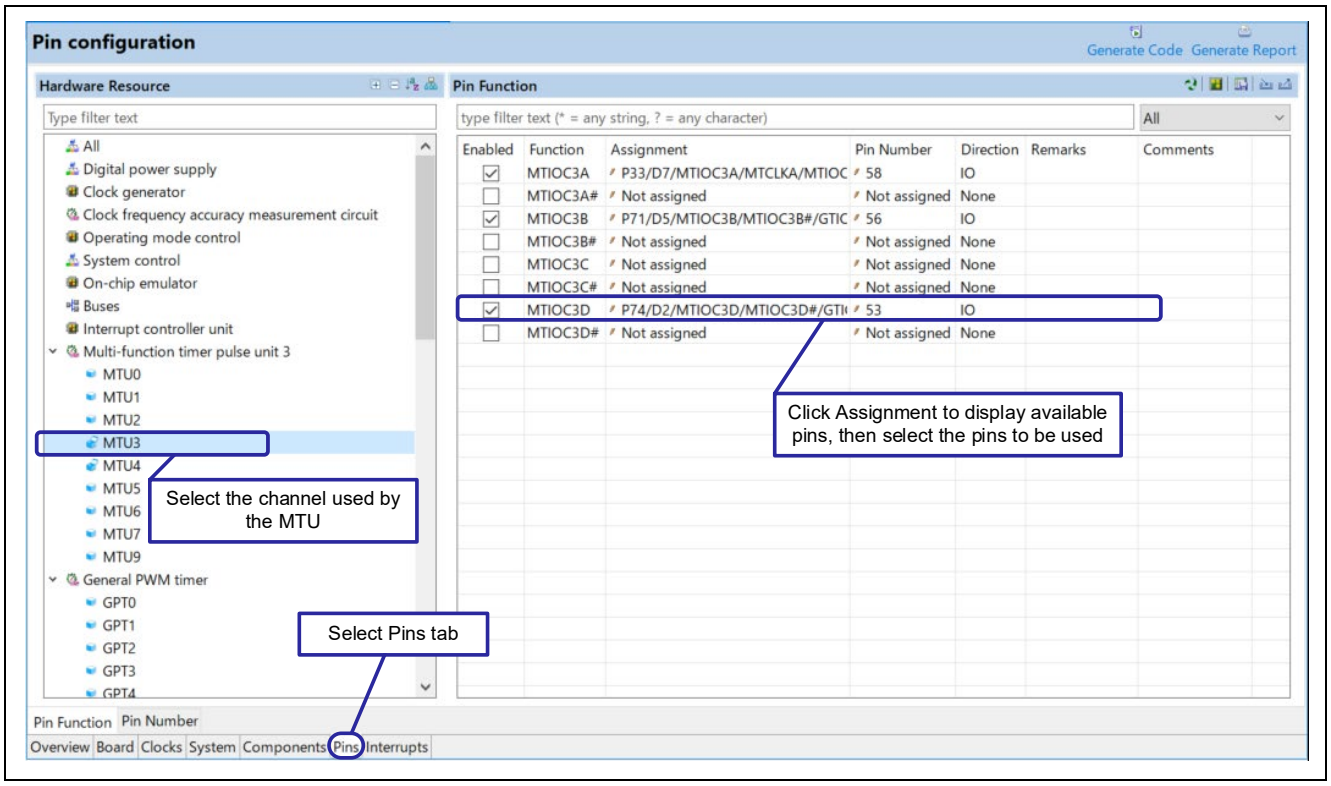


Figure 3.3 Pin Settings

3.1.6 Interrupt Settings

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

Configure interrupts after configuring 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.

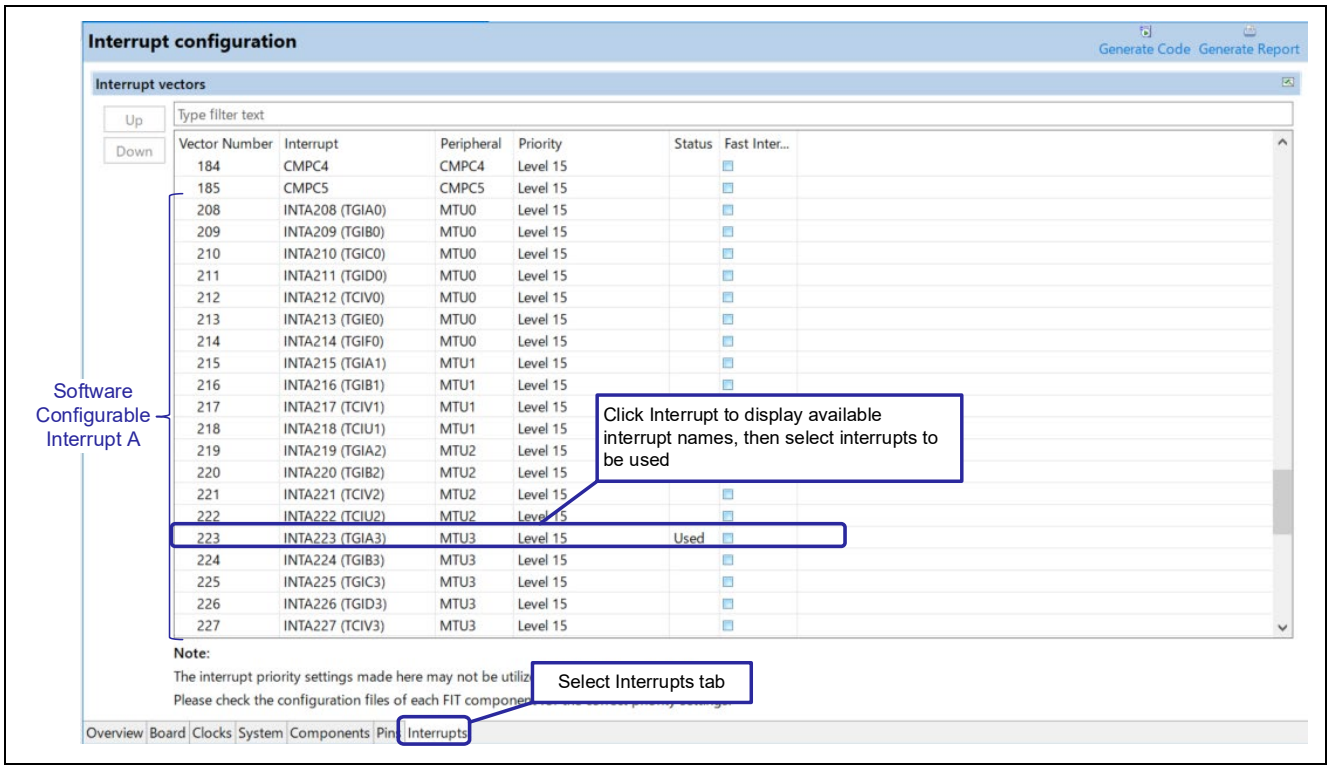


Figure 3.4 Interrupt Settings

3.2 A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 1

- Target sample code file name: r01an6643_rx66t_mtu3_ad_delay_1.zip

3.2.1 Overview

This sample code performs the A/D conversion start request delaying function in conjunction with interrupt skipping function 1 of the MTU.

In conjunction with TGIA3 interrupt skipping, the A/D conversion start request signal is generated to perform A/D conversion in S12AD0.

Complementary PWM mode in use can be used to generate three-phase waveform output (U, V, and W phases), but the sample code generates only single-phase (positive phase and negative phase) waveform output.

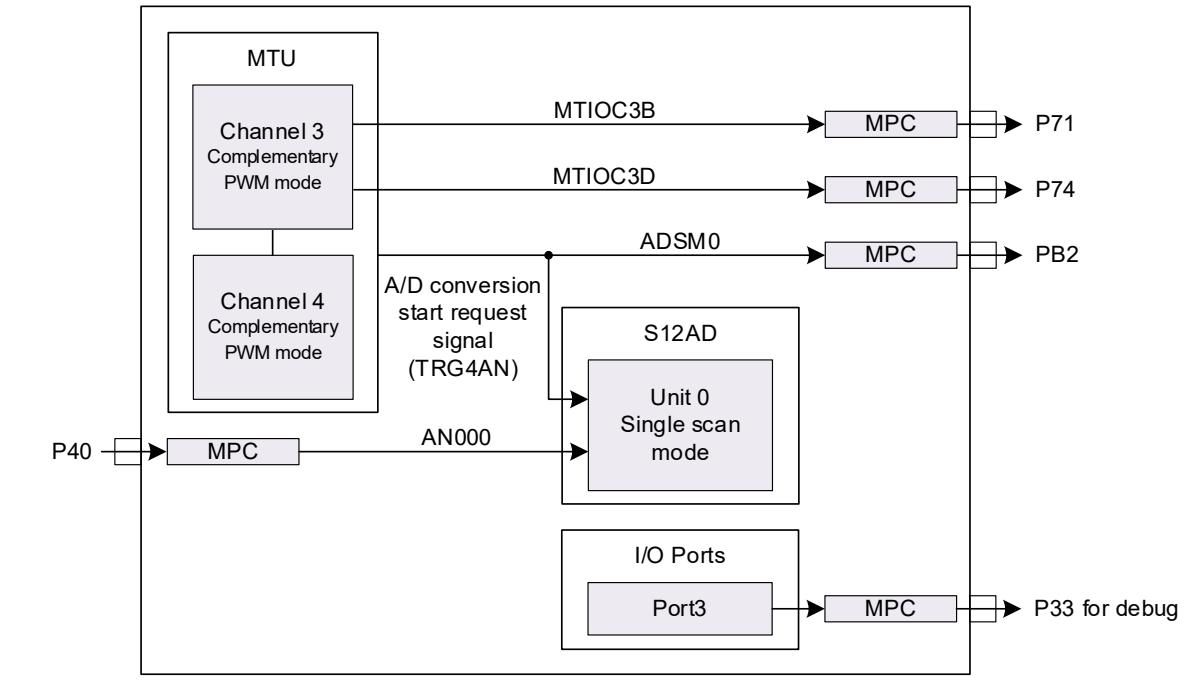
The following list provides the MTU and S12AD settings used in the sample code.

- MTU3 and MTU4 (channels 3 and 4)
  - Use complementary PWM mode 2 (transfer at trough).
  - Carrier period = 1 ms
  - Dead time = 30 µs
  - Timer counter clock = 40 MHz (PCLKC/4)
  - MTU3.TGRA sets the upper limit value of MTU3.TCNT (1/2 of carrier period + dead time).
  - The initial output value is high, and the active level is low.
  - Use MTU3.TGRB as the 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
  - A/D conversion start request linked with interrupt skipping function 1
    - Set MTU3.TGRA compare-match interrupt skipping count to 2 (counting once every 3 times).
    - Output A/D conversion start requests at compare matches between MTU4.TADCORA and MTU4.TCNT during up-counting
- S12AD0 (unit 0)
  - Use single scan mode
  - A/D conversion start trigger:
    Compare match between MTU4.TADCORA and MTU4.TCNT

Set in Smart Configurator.

Refer to section 3.2.3 for details on the setting method.

The structure of this sample code is shown below.



MTU: Generates complementary PWM mode waveform
S12AD: A/D conversion by A/D conversion start request from MTU
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

Figure 3.5 Sample Code Structure

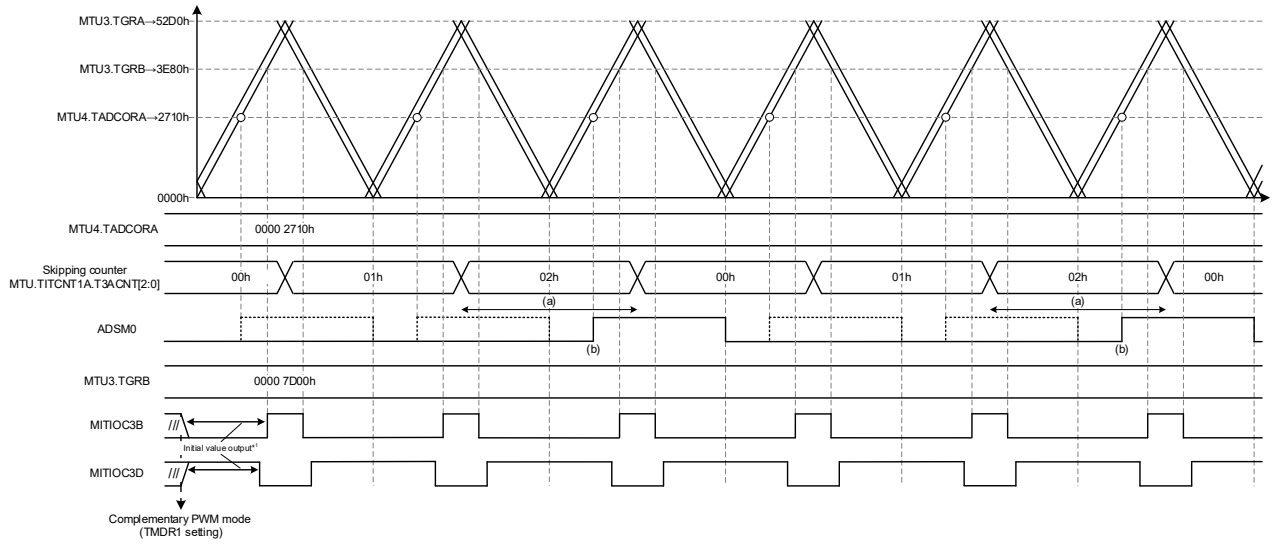
3.2.2 Operation Details

This section describes the operation of this sample code.

By setting the TGIA3 interrupt skipping count to 2, the A/D conversion request is enabled during the period when the value of the T3ACNT[2:0] bits is 02h ((a) in Figure 3.6), and the A/D conversion request signal generated during this period is valid. While the value of the T3ACNT[2:0] bits is not 02h, the A/D conversion request is invalid (dotted line for ADSM0 in Figure 3.6). A/D conversion start requests are generated by compare matches between MTU4.TADCORA and MTU4.TCNT which is up-counting, and the ADSM0 pin goes high ((b) in Figure 3.6).

For debugging purposes, this sample code is set to toggle output on the P33 pin when an A/D conversion end interrupt occurs. If the P33 pin is not to be set to toggle output, change Config_S12AD0_user.c to the following settings.

```
#define PRV_PORT_OUTPUT_ON        (0)
```



Note: 1. An initial value output starts when the PMR register is set.

Figure 3.6 Sample Code Operations
(A/D Conversion Start Request by Compare Match with TADCORA During Up-Counting, Interrupt Skipping Count: 2)

3.2.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the MTU and S12AD as described below. For details on how to add components, refer to section 3.1.4, Adding Components.

Table 3.4 Adding Components (MTU3, MTU4)

Item	Description
Component	Complementary PWM mode timer
Configuration name	Config_MTU3_MTU4
Operation	Complementary PWM mode 2 (transfer at trough).
Resource	MTU3_MTU4

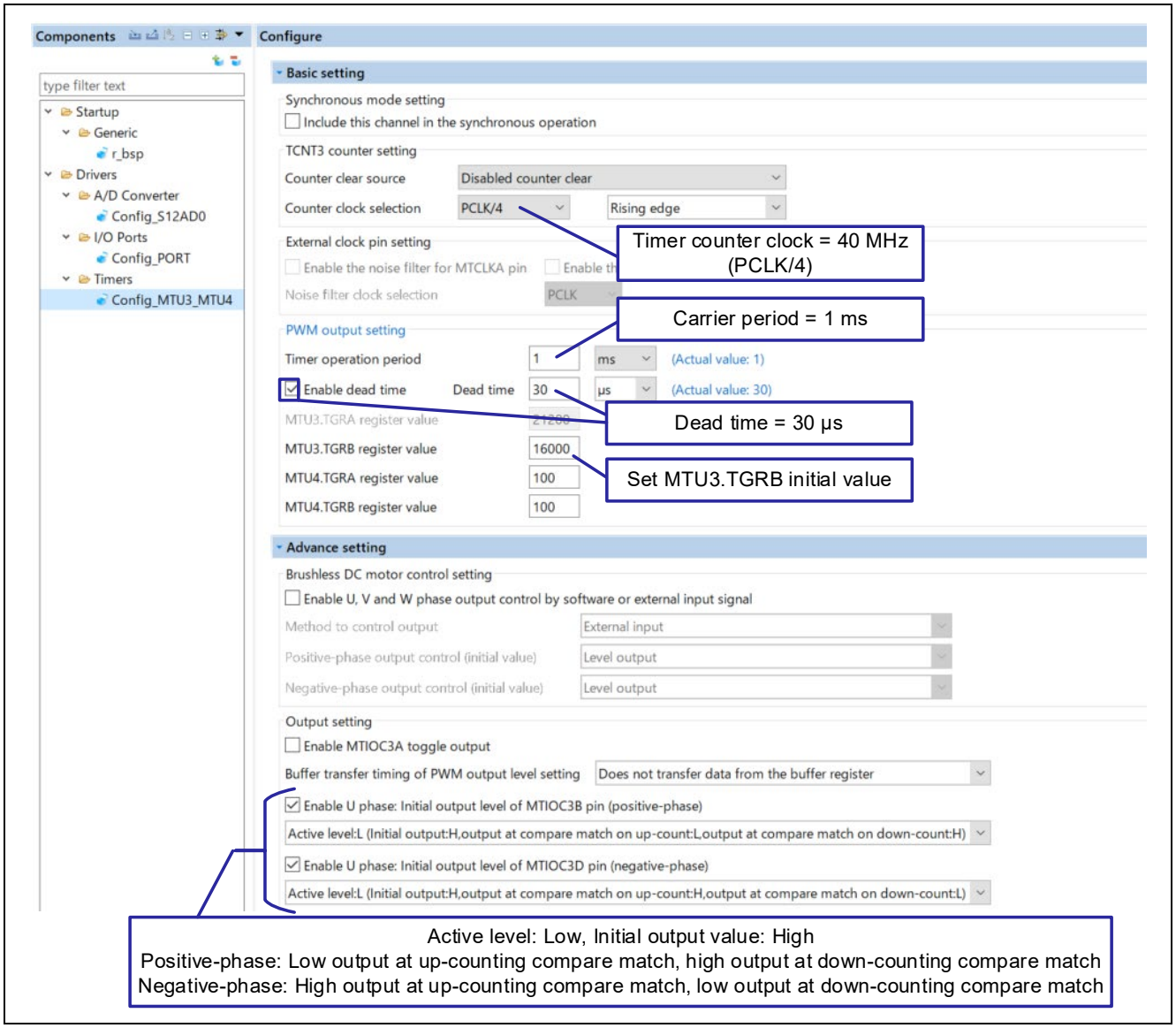


Figure 3.7 MTU3, MTU4 Settings (1/2)

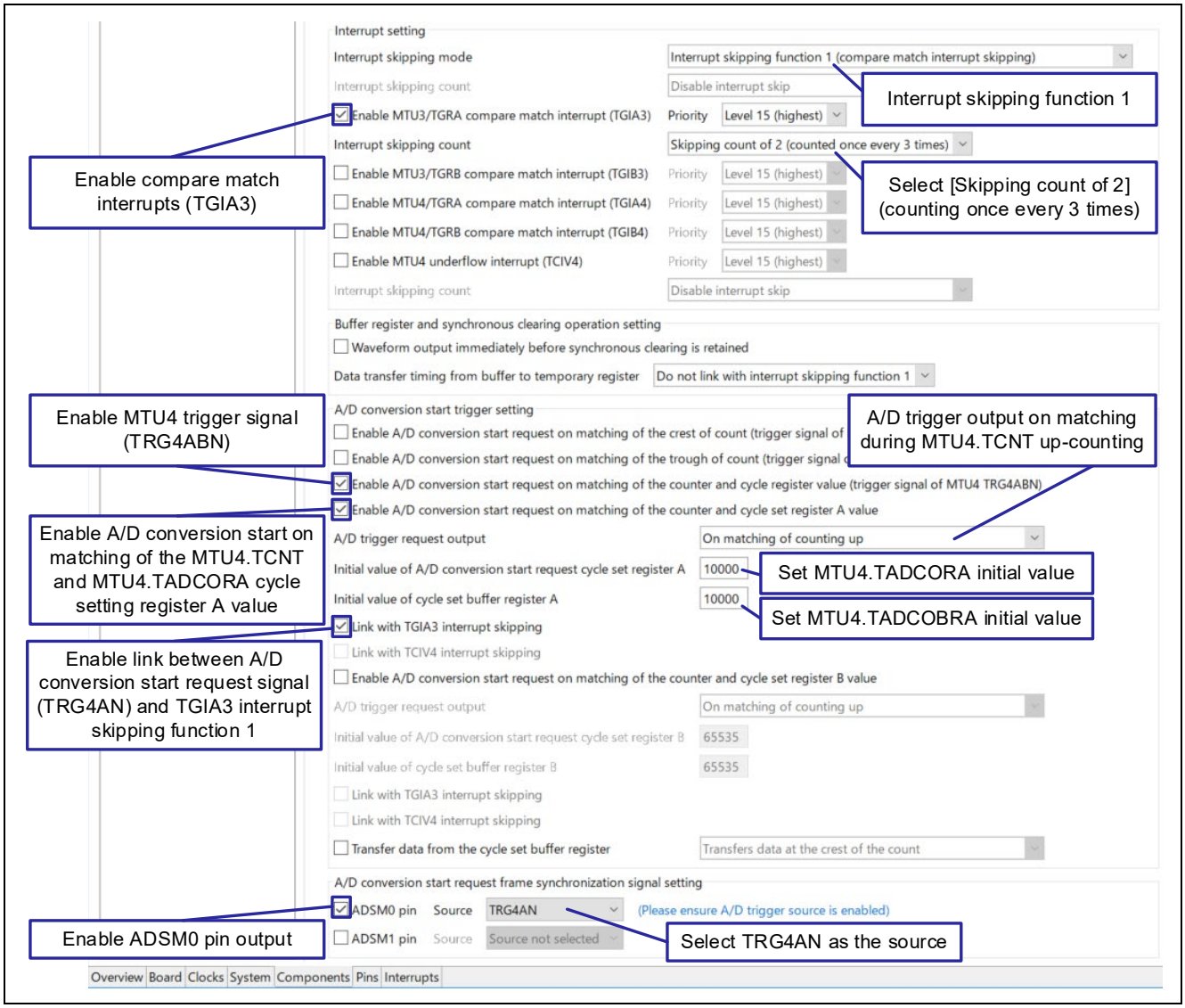


Figure 3.8 MTU3, MTU4 Settings (2/2)

Table 3.5 Adding Components (S12AD)

Item	Description
Component	Single-scan mode S12AD
Configuration name	Config_S12AD0
Resource	S12AD0

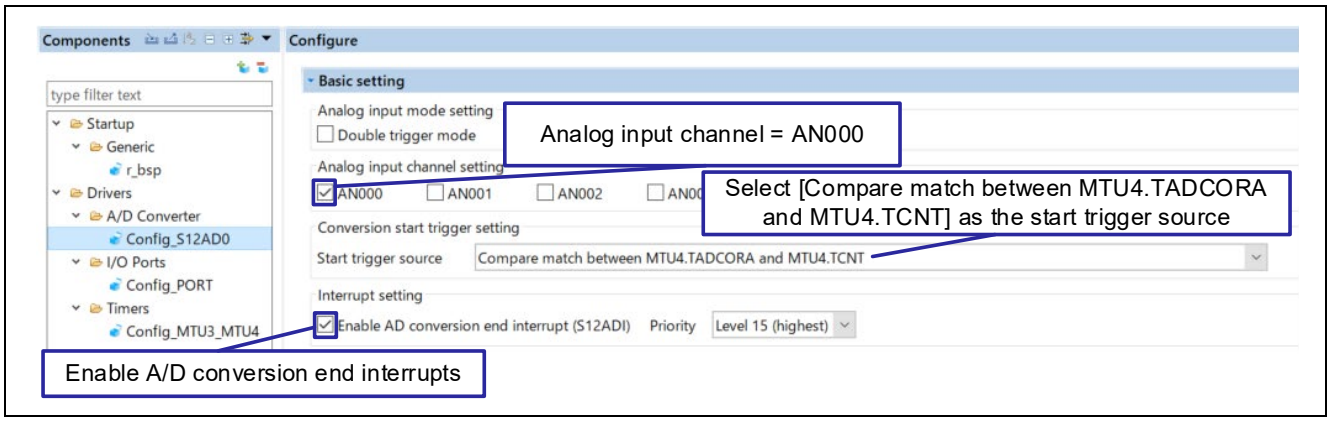


Figure 3.9 S12AD0 Settings

3.2.4 Flowchart

The following flowchart shows the main function processing added after code generation by the Smart Configurator.

In the main function, count start function R_Config_MTU3_MTU4_Start is read and counting is started.

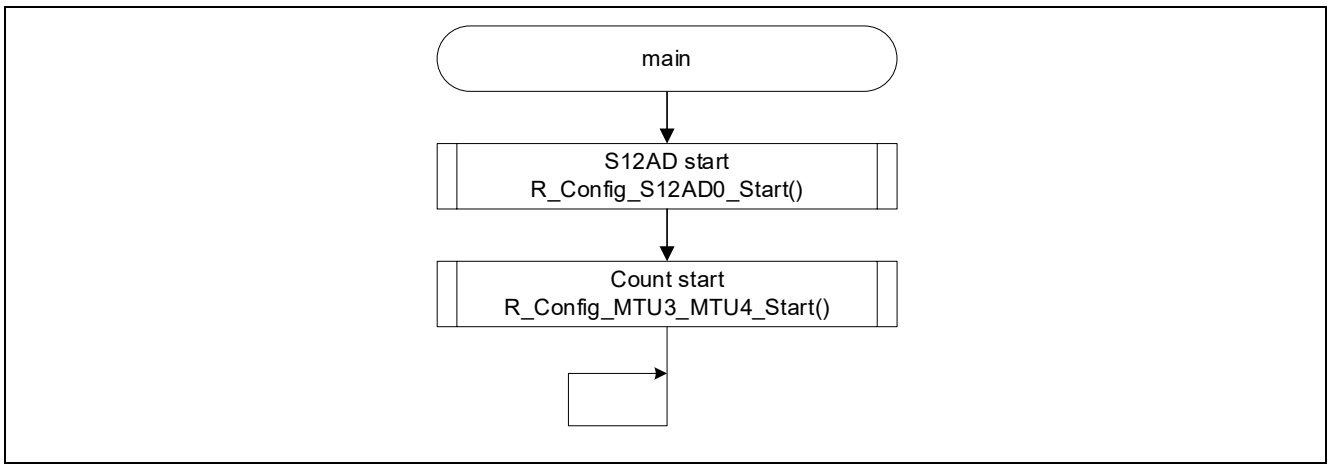


Figure 3.10 main Function

3.2.5 Related Operation

3.2.5.1 A/D Conversion Delaying Function Linked with Underflow Interrupts (TCIV4) Skipping

Generation of an A/D conversion start request signal in conjunction with overflow interrupt (TGIA3) skipping in this sample code was described above, but it is also possible to generate an A/D conversion start request signal in conjunction with underflow interrupt (TCIV4) skipping.

The figure below shows the Smart Configurator settings for generating an A/D conversion start request signal in conjunction with underflow interrupt (TCIV4) skipping.

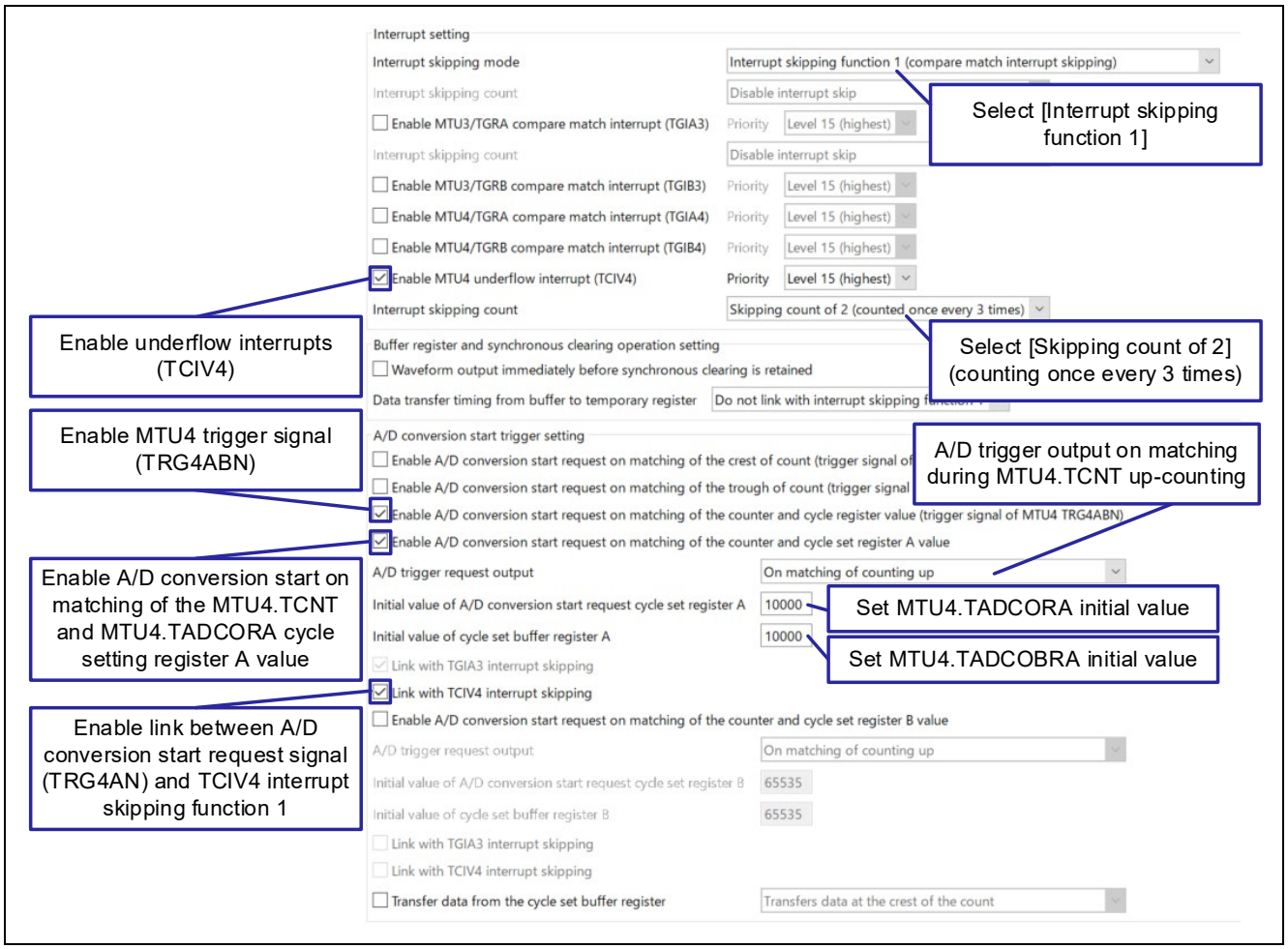


Figure 3.11 MTU3 and MTU4 Settings in Smart Configurator
(TCIV4 Interrupt Skipping Count: 2, A/D Conversion Start Request at Compare Match with MTU4.TADCORA During MTU4.TCNT Up-Counting)

3.2.6 Usage Notes

3.2.6.1 TITCR1A, TITCR1B, TADCORA, and TADCORB Register Settings

Refer to section 1.3.1(2), Notes when linking with interrupt skipping function 1, in this application note.

3.2.6.2 TADCOBRA and TADCOBRB Register Settings

Refer to section 1.2.1(2), Buffer transfer, in this application note.

3.2.6.3 Setting Linked Operation with the Skipping Function When Interrupt Skipping Is Disabled (Not Used)

Refer to section 1.3.1(2), Notes when linking with interrupt skipping function 1, in this application note.

3.3 A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 2

- Target sample code file name: r01an6643_rx66t_mtu3_ad_delay_2.zip

3.3.1 Overview

This sample code performs the A/D conversion start request delaying function in conjunction with interrupt skipping function 2 of the MTU.

Whenever the A/D conversion start trigger (TRG4AN or TRG4BN) occurs, the interrupt counter counts down, and when the counter reaches 0 and is reloaded, the AD conversion start request signal (TRG4ABN) is generated and A/D conversion is performed in S12AD0.

This function is valid only when the A/D conversion request delaying function is used.

Complementary PWM mode in use can be used to generate three-phase waveform output (U, V, and W phases), but the sample code generates only single-phase (positive phase and negative phase) waveform output.

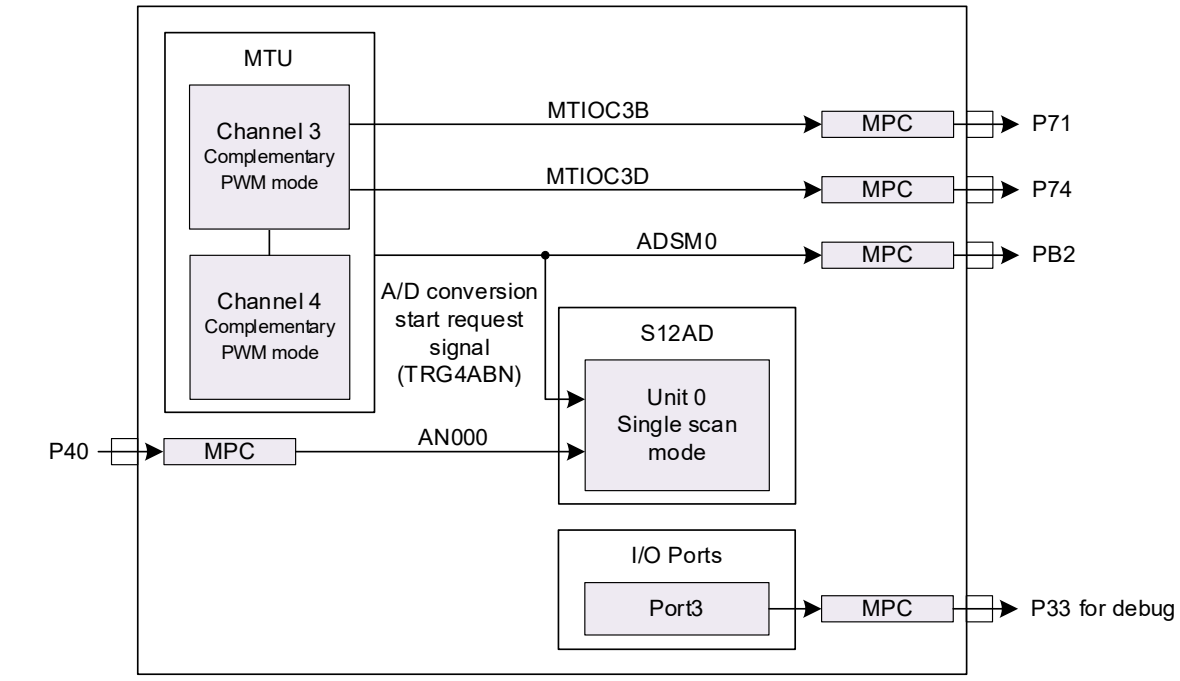
The following list provides the MTU and S12AD settings used in the sample code.

- MTU3 and MTU4 (channels 3 and 4)
  - Use complementary PWM mode 2 (transfer at trough).
  - Carrier period = 1 ms
  - Dead time = 30 µs
  - Timer counter clock = 40 MHz (PCLKC/4)
  - Set MTU3.TCNT upper limit value in MTU3.TGRA (1/2 carrier period + dead time).
  - The initial output value is high, and the active level is low.
  - Use MTU3.TGRB as the 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
  - A/D conversion start request linked with interrupt skipping function 2
    - Set TRG4AN and TRG4BN interrupt skipping count to 4 (counting once every 5 times).
    - Output A/D conversion start requests at compare matches between MTU4.TADCORA and MTU4.TCNT during up/down-counting
    - Output A/D conversion start requests at compare matches between MTU4.TADCORB and MTU4.TCNT during up/down-counting
- S12AD0 (unit 0)
  - Use single scan mode
  - A/D conversion start trigger:
    Compare match between MTU4.TADCORA and MTU4.TCNT,
    Compare match between MTU4.TADCORB and MTU4.TCNT

Set in Smart Configurator.

Refer to section 3.3.3 for details on the setting method.

The structure of this sample code is shown below.



MTU: Generates complementary PWM mode waveform
S12AD: A/D conversion by A/D conversion start request from MTU
MPC: Sets the pins to be used from general purpose I/O port to peripheral function I/O port

Figure 3.12 Sample Code Structure

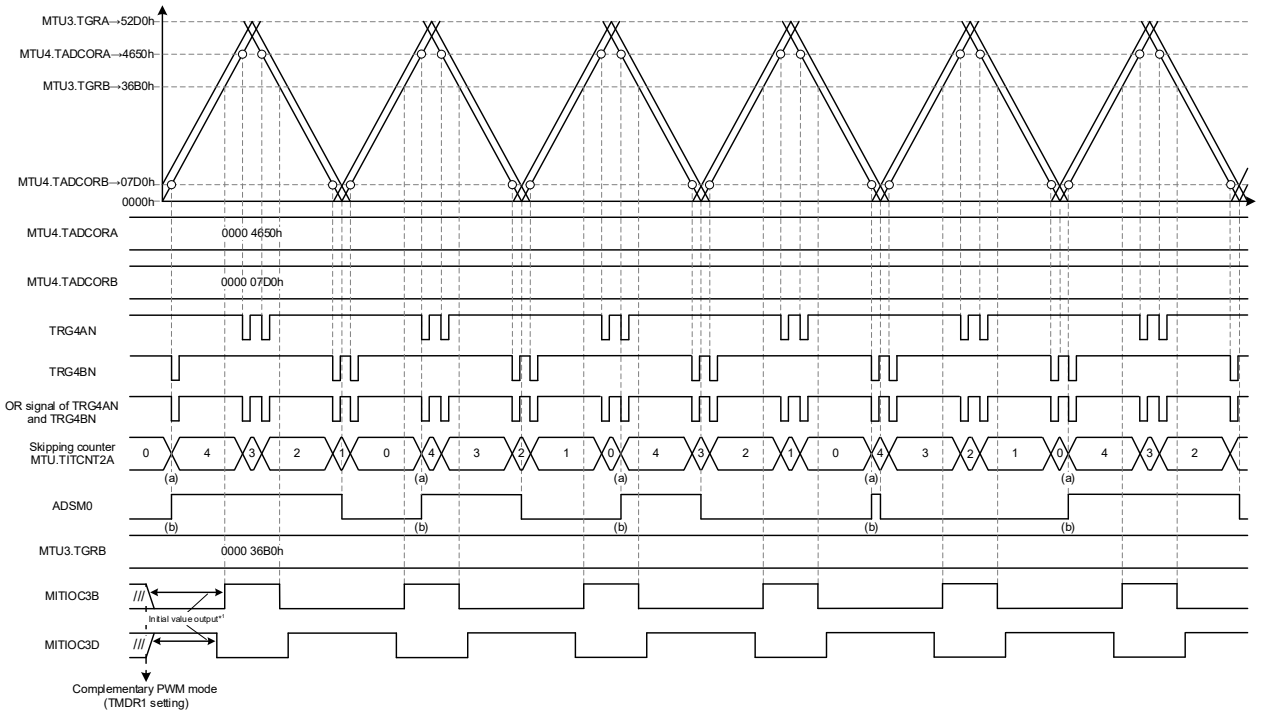
3.3.2 Operation Details

This section describes the operation of this sample code.

Setting the interrupt skipping count to 4 in interrupt skipping function 2 sets the counter value 4 in timer interrupt skipping count counter 2A (TITCNT2A). The value of TITCNT2A counts down from 4 each time the A/D conversion start trigger TRG4AN or TRG4BN occurs. When the counter value becomes 0 and the counter is reloaded to the skipping count value (4), the A/D conversion start request signals TRG4AN and TRG4BN become valid ((a) in Figure 3.13). A/D conversion start request signal TRG4ABN is output and the ADSM0 pin goes high ((b) in Figure 3.13).

For debugging purposes, this sample code is set to toggle output on the P33 pin when an A/D conversion end interrupt occurs. If the P33 pin is not to be set to toggle output, change Config_S12AD0_user.c to the following settings.

```
#define PRV_PORT_OUTPUT_ON        (0)
```



Note: 1. An initial value output starts when the PMR register is set.

Figure 3.13 Sample Code Operations
(A/D Conversion Start Request by Compare Match with TADCORA and TADCORB During Up-/Down-Counting, Interrupt Skipping Count: 4)

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 (MTU3, MTU4)

Item	Description
Component	Complementary PWM mode timer
Configuration name	Config_MTU3_MTU4
Operation	Complementary PWM mode 2 (transfer at trough).
Resource	MTU3_MTU4

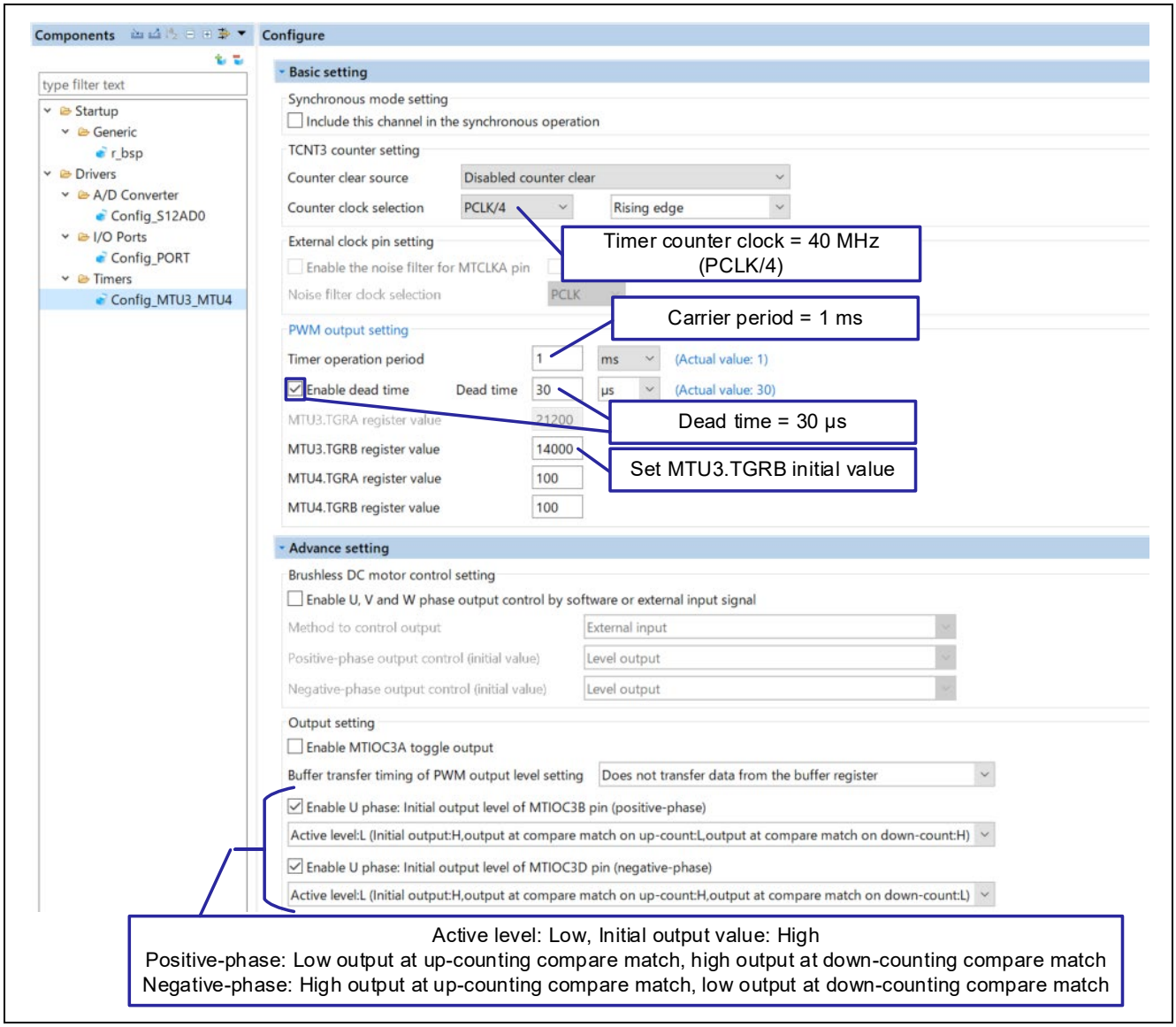


Figure 3.14 MTU3, MTU4 Settings (1/2)

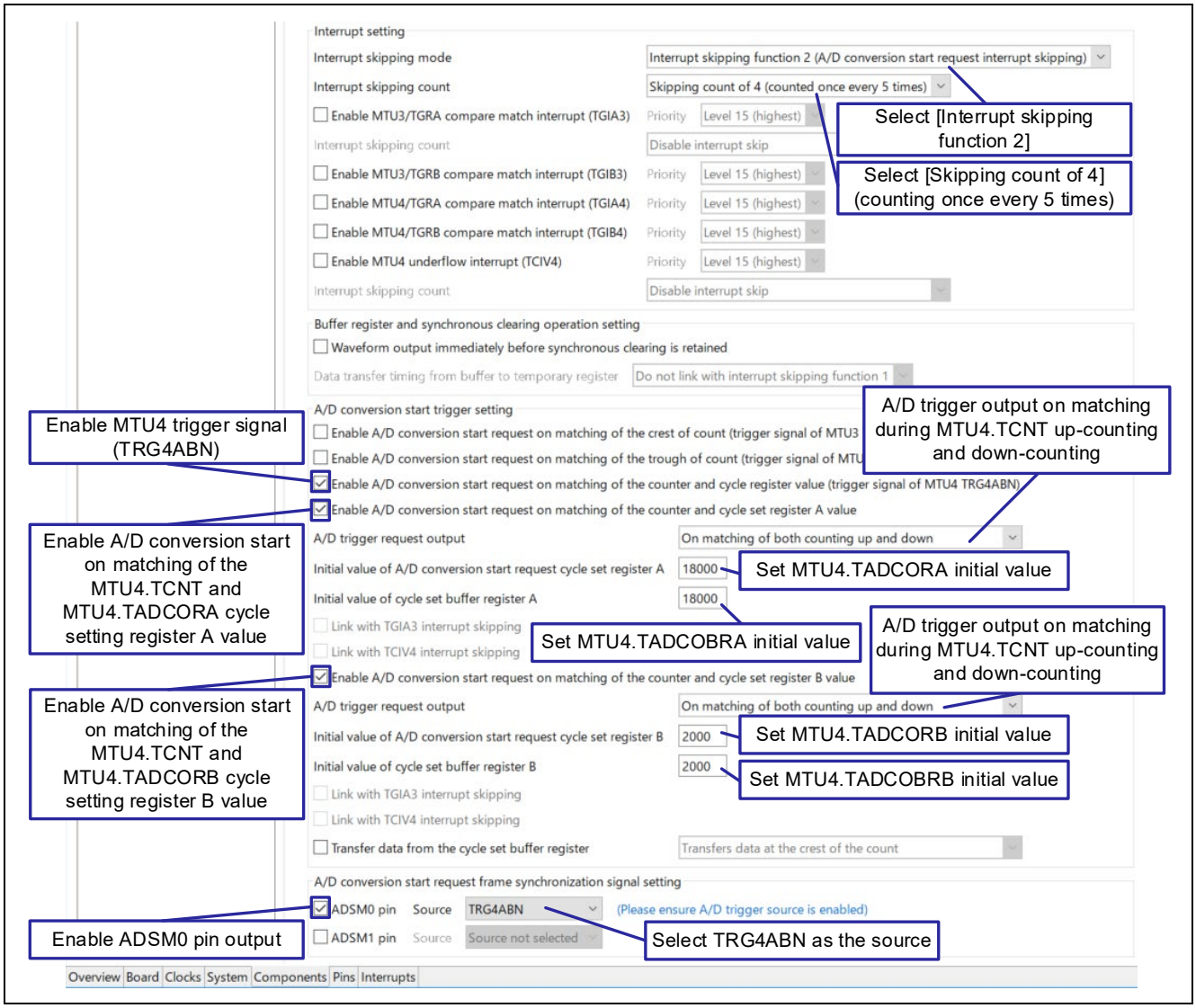


Figure 3.15 MTU3, MTU4 Settings (2/2)

Table 3.7 Adding Components (S12AD)

Item	Description
Component	Single-scan mode S12AD
Configuration name	Config_S12AD0
Resource	S12AD0

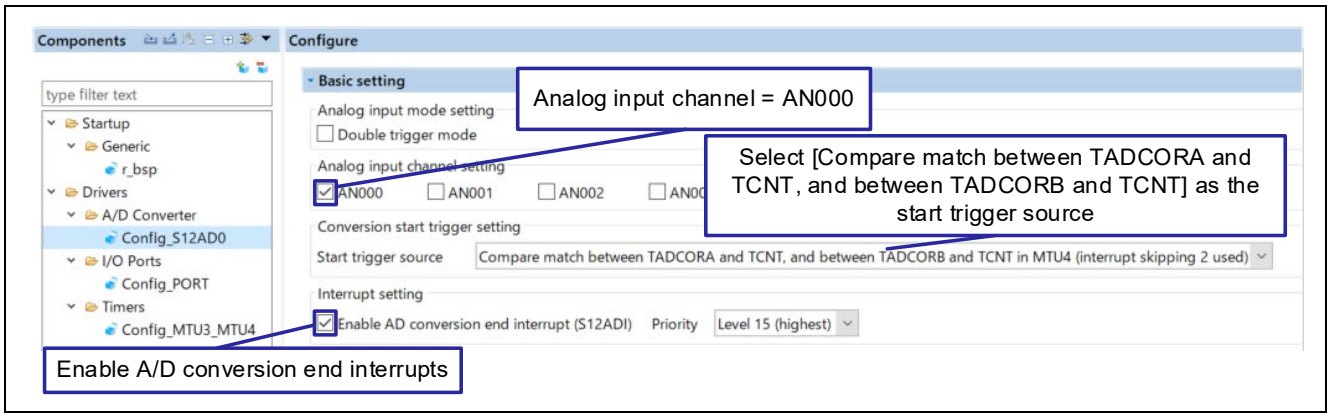


Figure 3.16 S12AD0 Settings

3.3.4 Flowchart

The following flowchart shows the main function processing added after code generation by the Smart Configurator.

In the main function, count start function R_Config_MTU3_MTU4_Start is read and counting is started.

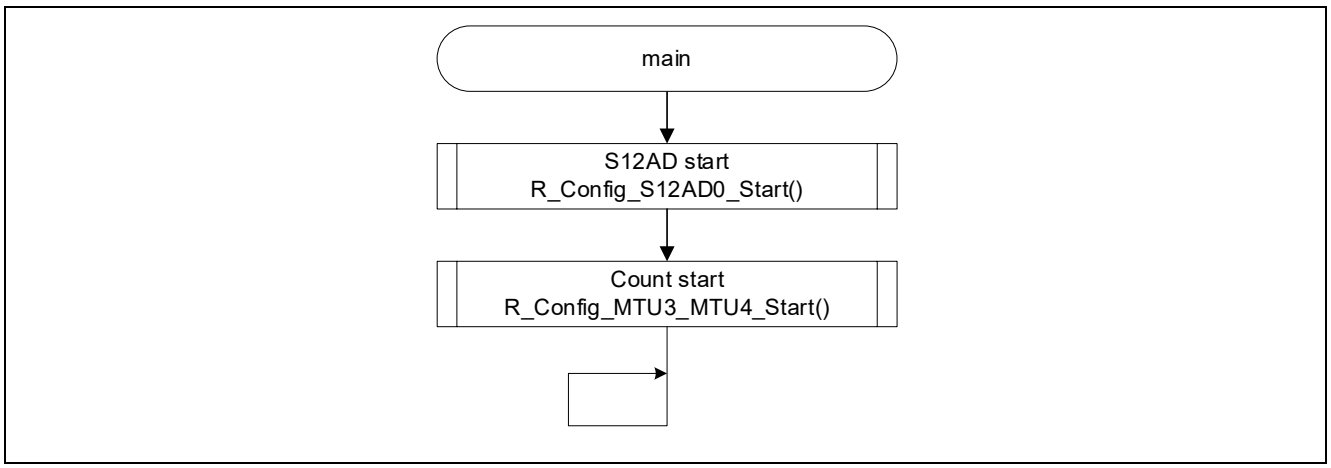


Figure 3.17 main Function

3.3.5 Related Operation

3.3.5.1 A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 2 in Mode Other Than Complementary PWM Mode

The Smart Configurator environment in Version 2022-10 of e^2 studio does not support the A/D conversion request delaying function linked with skipping function 2 in PWM mode 1, normal mode, and reset-synchronized PWM mode.

To use the A/D conversion request delaying function in conjunction with skipping function 2 in PWM mode 1 or normal mode, add the TITMRA and TITCR2A register settings to the user initialization function R_Config_MTU3_MTU4_Create_UserInit (R_Config_MTU6_MTU7_Create_UserInit) after code generation by the Smart Configurator.

The following are examples of TITMRA and TITCR2A register settings.

- Register settings of skipping function 2: enabled, and skipping count: 4 (counted once every 5 times)

```
MTU.TITMRA.BIT.TITM = 1U;
MTU.TITCR2A.BYTE = _04_MTU_TRGCOR_4_7_SKIP_COUNT_4;
```

To use the A/D conversion request delaying function in conjunction with skipping function 2 in reset-synchronized PWM mode, use the normal mode component in the Smart Configurator, generate code, and add register settings to the user initialization function as described above. For details on how to output waveforms in reset-synchronized PWM mode using normal mode components, refer to section 3.8, Reset-Synchronized PWM Mode in Application Note: RX Family PWM Output Methods Using MTU3/GPTW.

3.3.6 Usage Notes

3.3.6.1 TADCORA and TADCORB Register Settings

Refer to section 1.3.2(2), Notes when using interrupt skipping function 2, in this application note.

4. GPTW Sample Codes

4.1 Common

4.1.1 Sample Code List

This application note provides the following sample codes created with the Smart Configurator.

Sample codes can be downloaded from the Renesas Electronics website.

Table 4.1 GPTW Sample Code List

Name	Description	Ref.
A/D Conversion Start Request Function Linked with the Interrupt Skipping Function by the GTITC Register r01an6643_rx66t_gptw_ad_delay_1.zip	• Triangle-wave PWM mode 1 • Single buffer	4.2
A/D Conversion Start Request Function Linked with the Extended Interrupt Skipping Function r01an6643_rx66t_gptw_ad_delay_2.zip	• Triangle-wave PWM mode 1 • Single buffer	4.3

4.1.2 Folder Structure

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

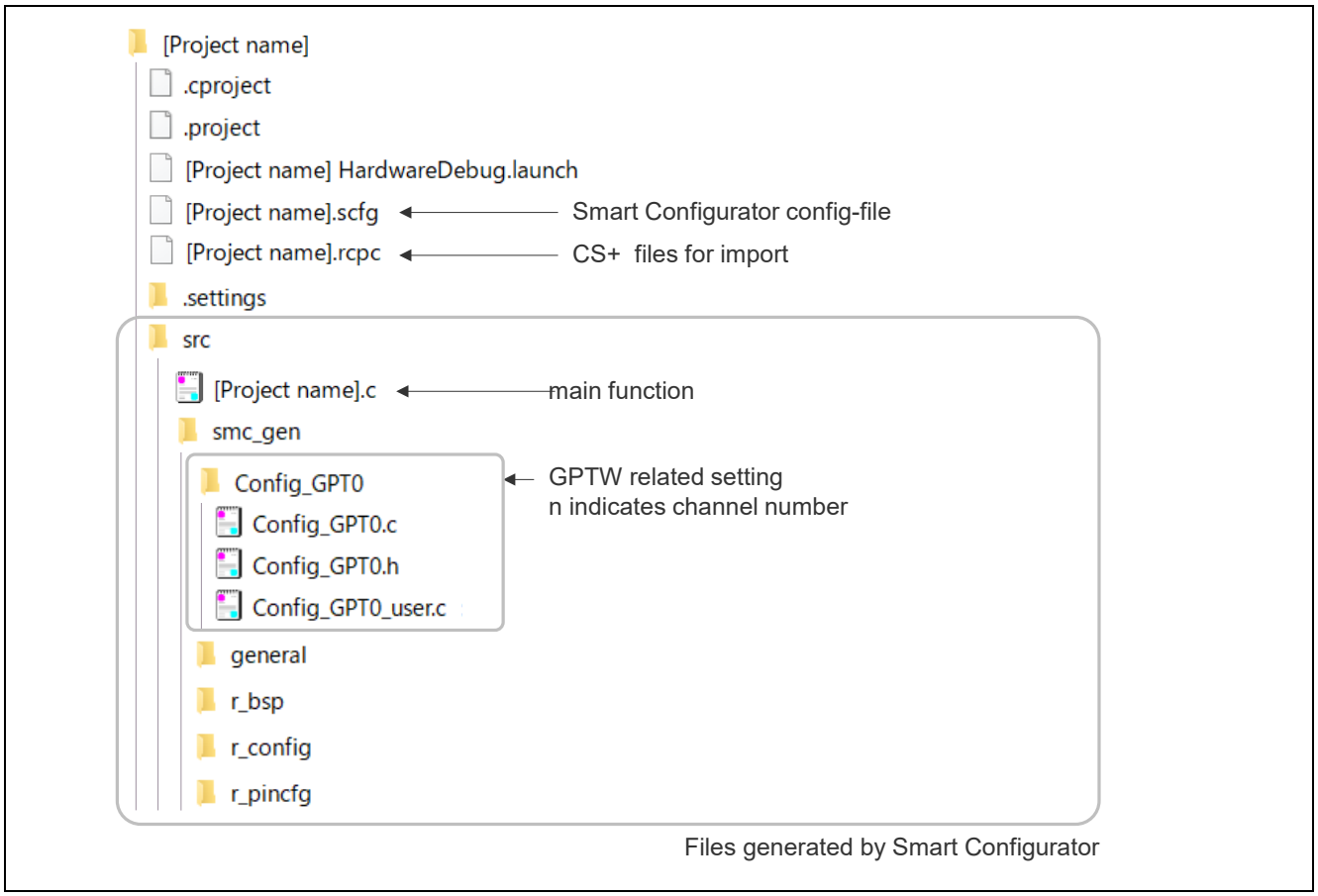


Figure 4.1 GPTW Folder Structure

4.1.3 File Structure

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

Table 4.2 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*1	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*1	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*1	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, make sure that you include the r_smc_entry.h file.

Note: 1. n indicates a channel number.

4.1.4 Adding Components

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

Table 4.3 Adding Components

Item	Description
Component	General PWM timer ((1) in the figure below)
Configuration name	Sample codes use the default setting name.
Operation	Refer to the section for each sample code ((2) in the figure below).
Resource	Refer to the section for each sample code ((3) in the figure below).

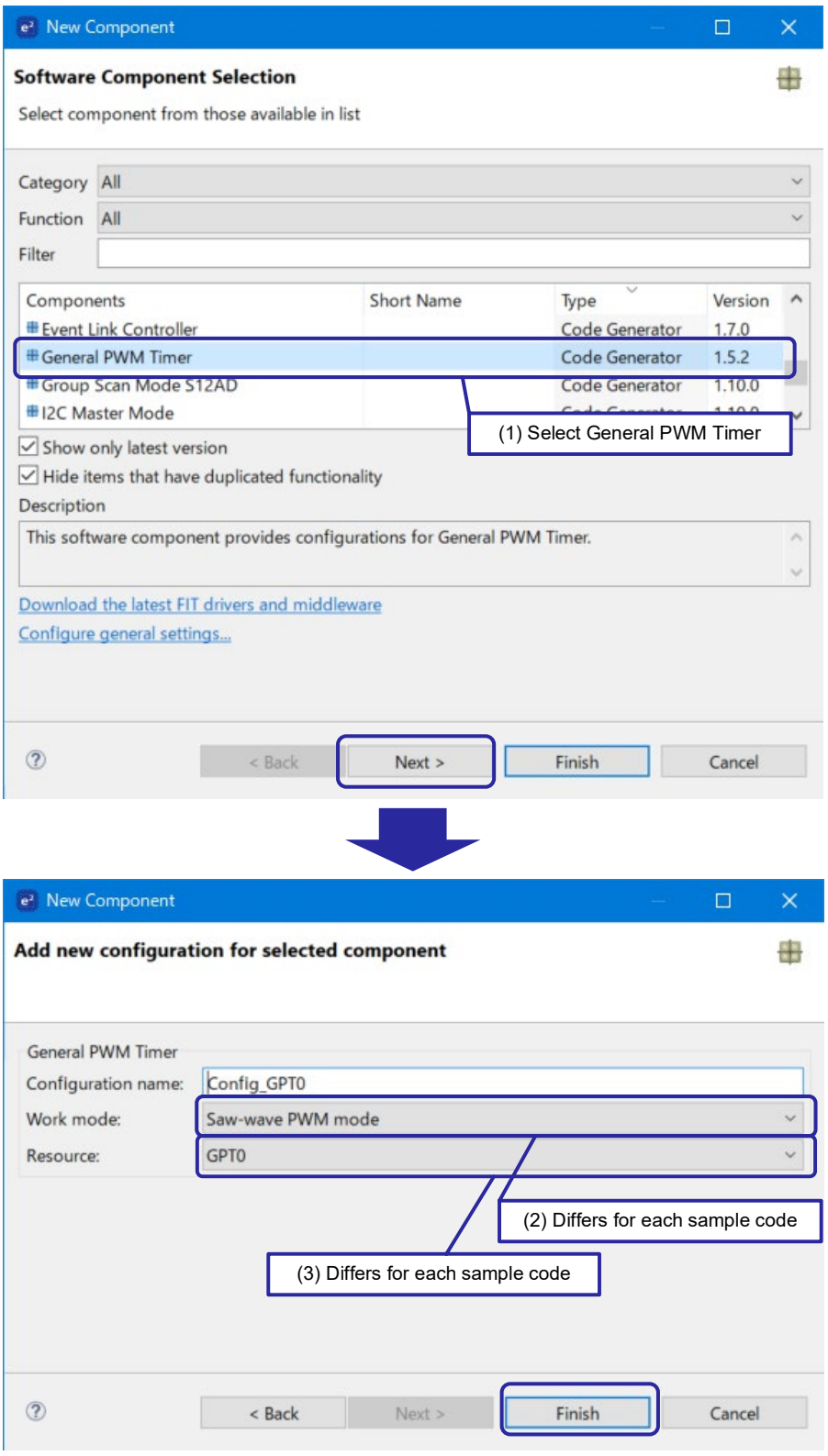


Figure 4.2 Adding Components

4.1.5 Pin Settings

Figure 4.3 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.

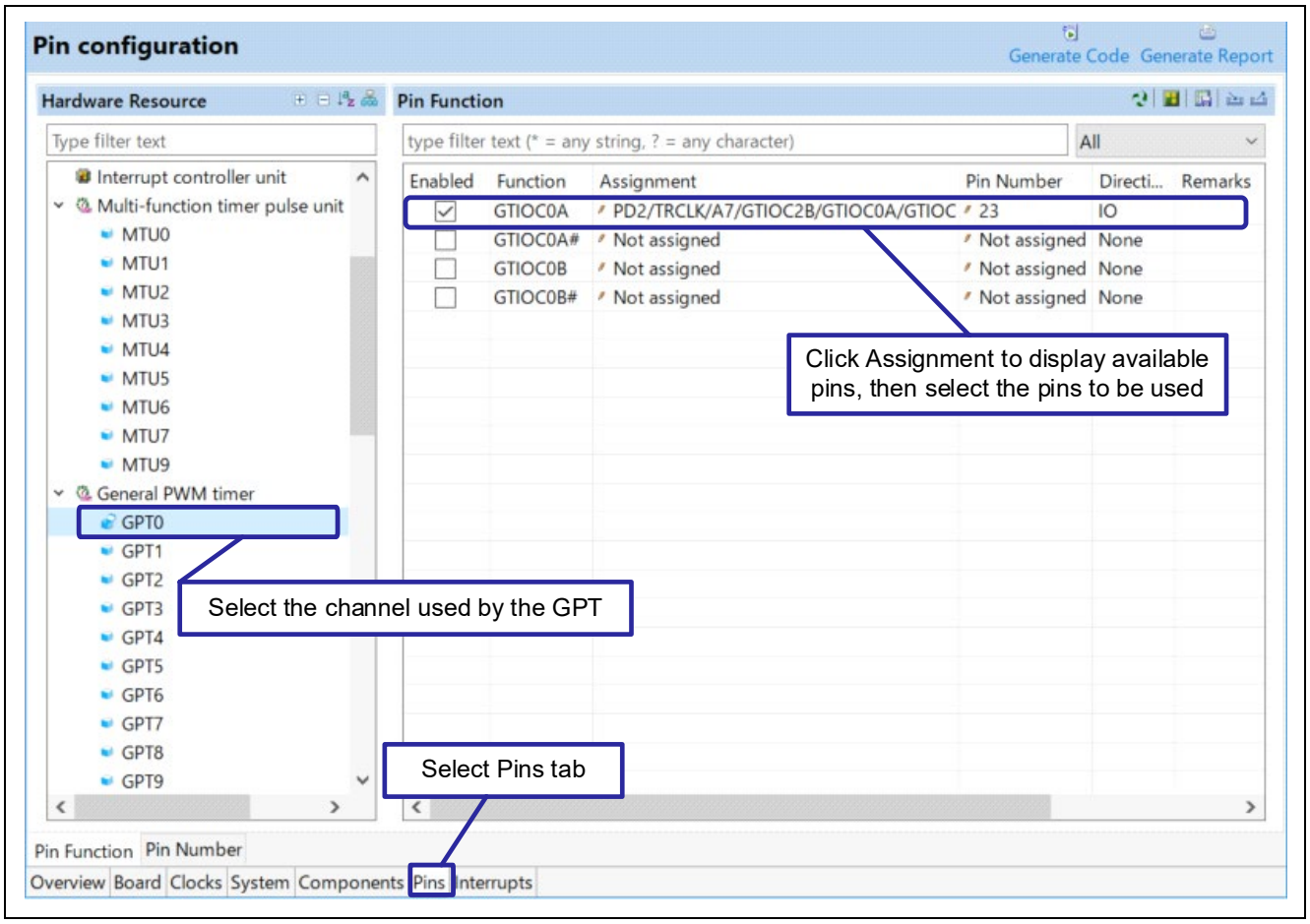


Figure 4.3 Pin Settings

4.1.6 Interrupt Settings

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

Configure interrupts after configuring 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.

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

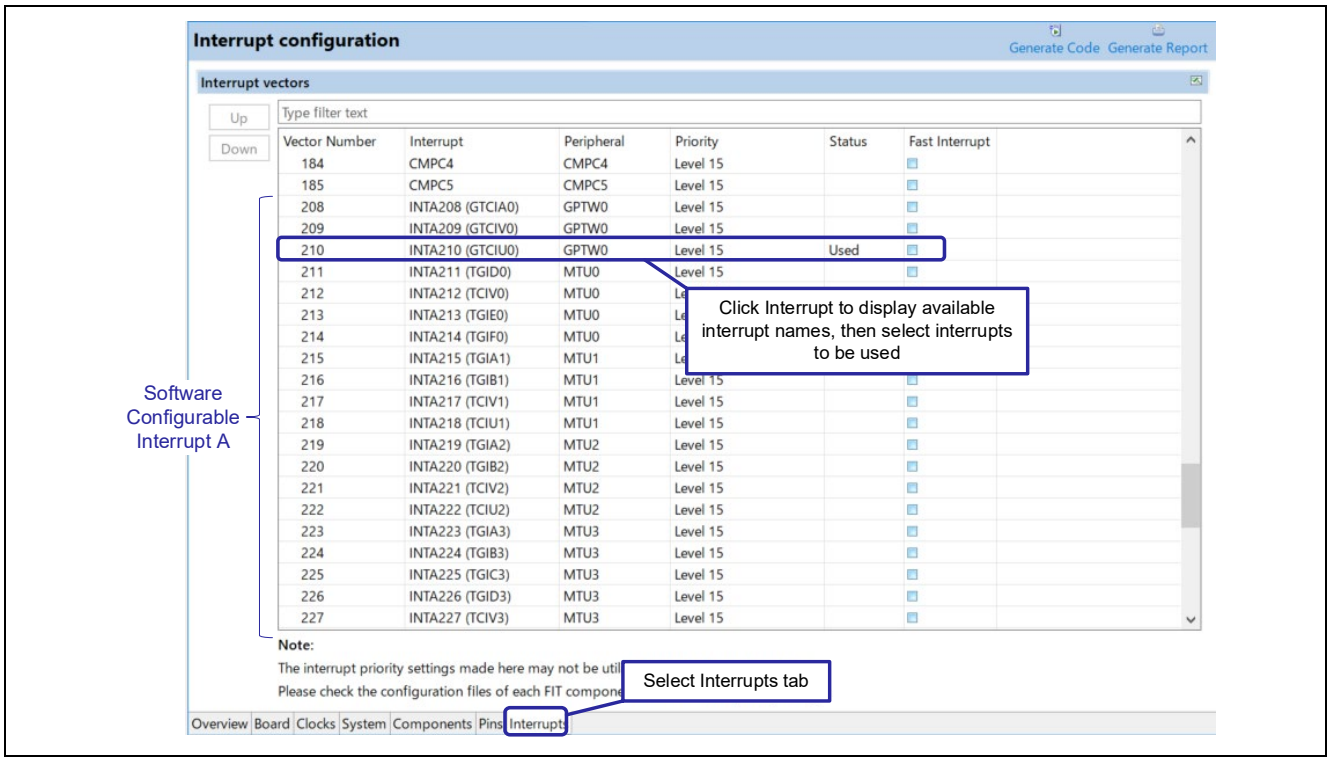


Figure 4.4 Interrupt Settings

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.

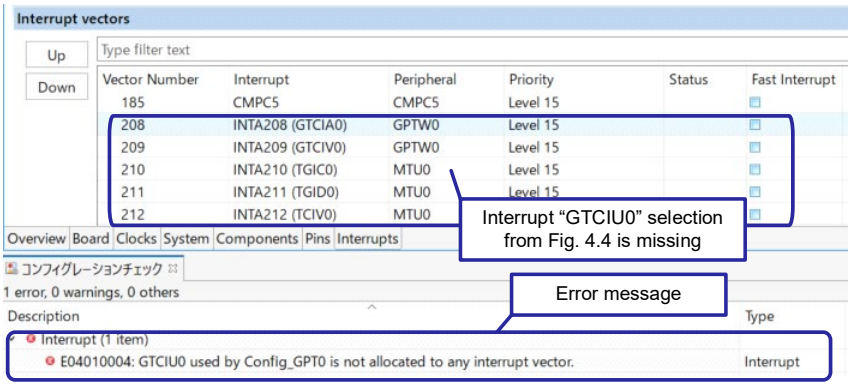


Figure 4.5 Interrupt Settings (Interrupt Selection Missing)

4.2 A/D Conversion Start Request Function Linked with the Interrupt Skipping Function by the GTITC Register

- Target sample code file name: r01an6643_rx66t_gptw_ad_delay_1.zip

4.2.1 Overview

This sample code performs the A/D conversion start request function in conjunction with the interrupt skipping function by the GTITC register of the GPTW.

In conjunction with the GTCNT underflow interrupt skipping function, the A/D conversion start request signals are generated to perform A/D conversion in S12AD0 through ELC event signals.

The following list provides the GPTW, S12AD, and ELC settings used in the sample code.

- GPTW0 (channel 0)
  - Use triangle-wave PWM mode 1
  - Timer counter clock = 160 MHz (PCLKC)
  - Carrier period = 1 ms
  - Use GTPR as the cycle setting register
    - Counts are up-counted from the initial value of 0
  - Use GTCCRA for compare matches of the duty cycle output
    - Use the GTIOC0A pin as the PWM output pin
    - High output when counting starts and stops
    - Toggle output at GTCCRA compare match
    - Retain output at cycle end
  - Use GTCCRB for compare matches of the duty cycle output
    - Use the GTIOC0B pin as the PWM output pin
    - Low output when counting starts and stops
    - Toggle output at GTCCRB compare match
    - Retain output at cycle end
  - Software source count start enabled
  - Use automatic dead time generation
  - Enable A/D conversion start request
    - A/D conversion start requests generated by compare matches between GTCNT and the GTADTRA register during GTCNT up-counting
    - Select [GTADTRA compare match during up-counting] for A/D conversion start request signal monitor 0
  - Use the GTCNT underflow interrupt skipping function
    - Count troughs with the skipping count of 2
    - Enable linked operation between A/D conversion start request signal generated by GTADTRA and the GTCIU0 interrupt skipping function
- S12AD0 (unit 0)
  - Use single scan mode
  - Start A/D conversion by A/D startup source 0 from ELC
- ELC
  - Select GPT0 A/D conversion start request A for an event
  - Select S12AD0 (ELCTRG00N) as the transmission destination resource

Set in Smart Configurator.

Refer to section 4.2.3 for details on the setting method.

The structure of this sample code is shown below.

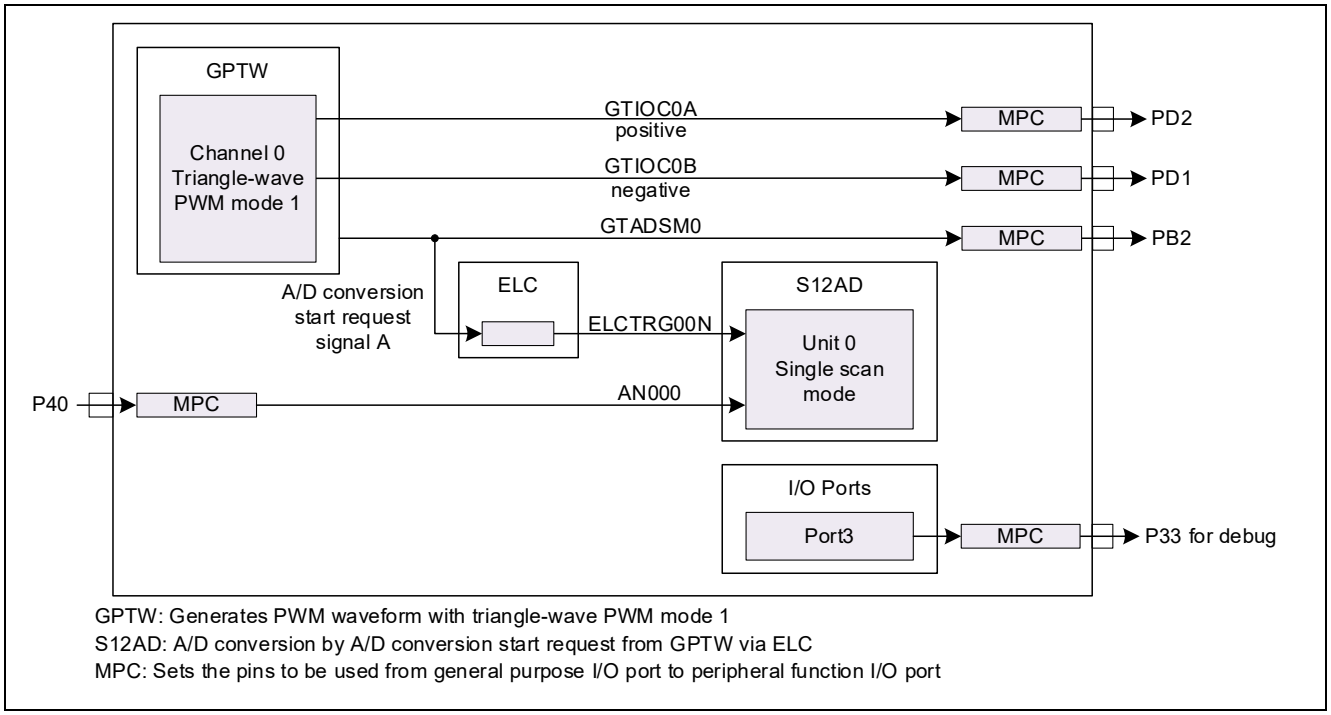


Figure 4.6 Sample Code Structure

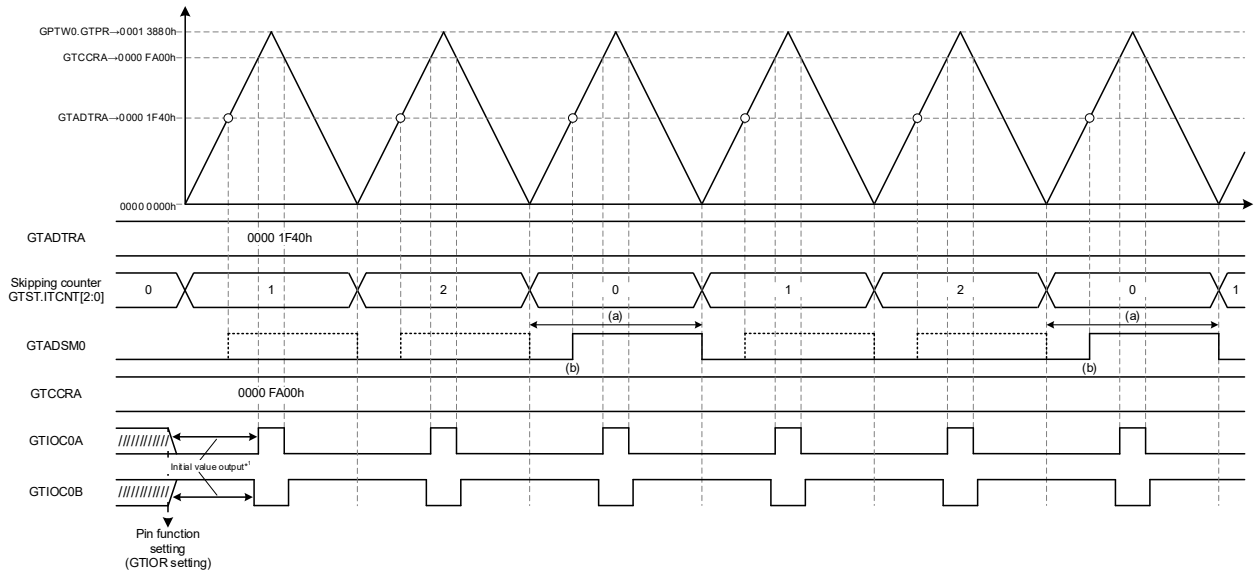
4.2.2 Operation Details

This section describes the operation of this sample code.

By setting the GTCNT underflow interrupt skipping count to 2, the A/D conversion start request is enabled during the period when the value of the ITCNT[2:0] bits is 0b ((a) in Figure 4.7), and the A/D conversion request signal generated during this period is valid. While the value of the ITCNT[2:0] bits is not 0b, GTCNT underflow interrupts are skipped and the A/D conversion start request is invalid (dotted line for GTADSM0 in Figure 4.7). A/D conversion start requests are generated by compare matches with GTADTRA during up-counting, and the ADSM0 pin goes high ((b) in Figure 4.7).

For debugging purposes, this sample code is set to toggle output on the P33 pin when an A/D conversion end interrupt occurs. If the P33 pin is not to be set to toggle output, change Config_S12AD0_user.c to the following settings.

```
#define PRV_PORT_OUTPUT_ON        (0)
```



Note: 1. An initial value output starts when the PMR register is set.

Figure 4.7 Sample Code Operations
(A/D Conversion Start Request by Compare Match with GTADTRA During Up-Counting, ITCNT: Counts Troughs with the Skipping Count of 2)

4.2.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the GPTW, S12AD, and ELC as described below. For details on how to add components, refer to section 4.1.4, Adding Components.

Table 4.4 Adding Components (GPTW0)

Item	Description
Component	General PWM timer
Configuration name	Config_GPT0
Work mode	Triangle-wave PWM mode 1
Resource	GPT0

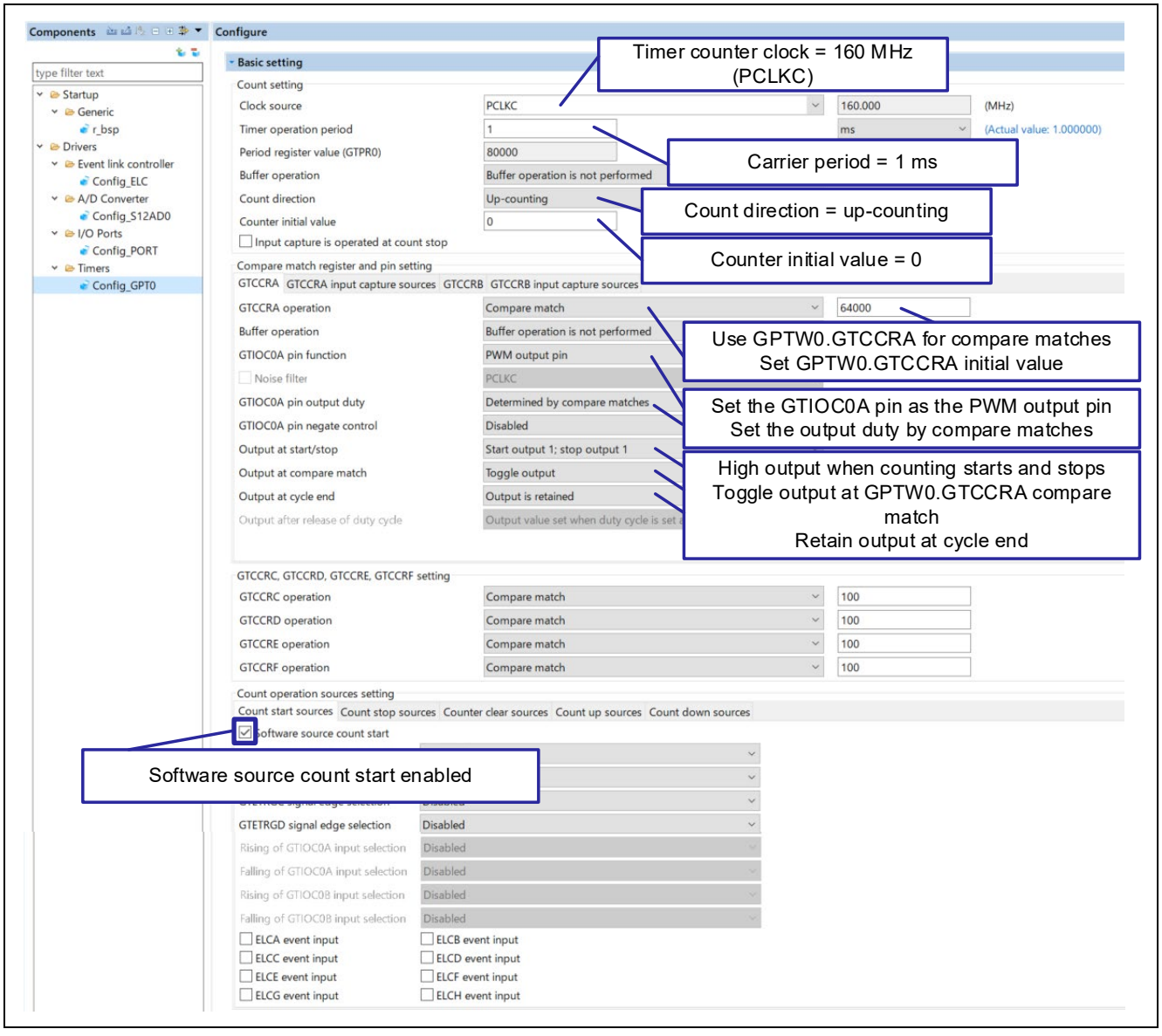


Figure 4.8 GPT0 Settings (1/3)

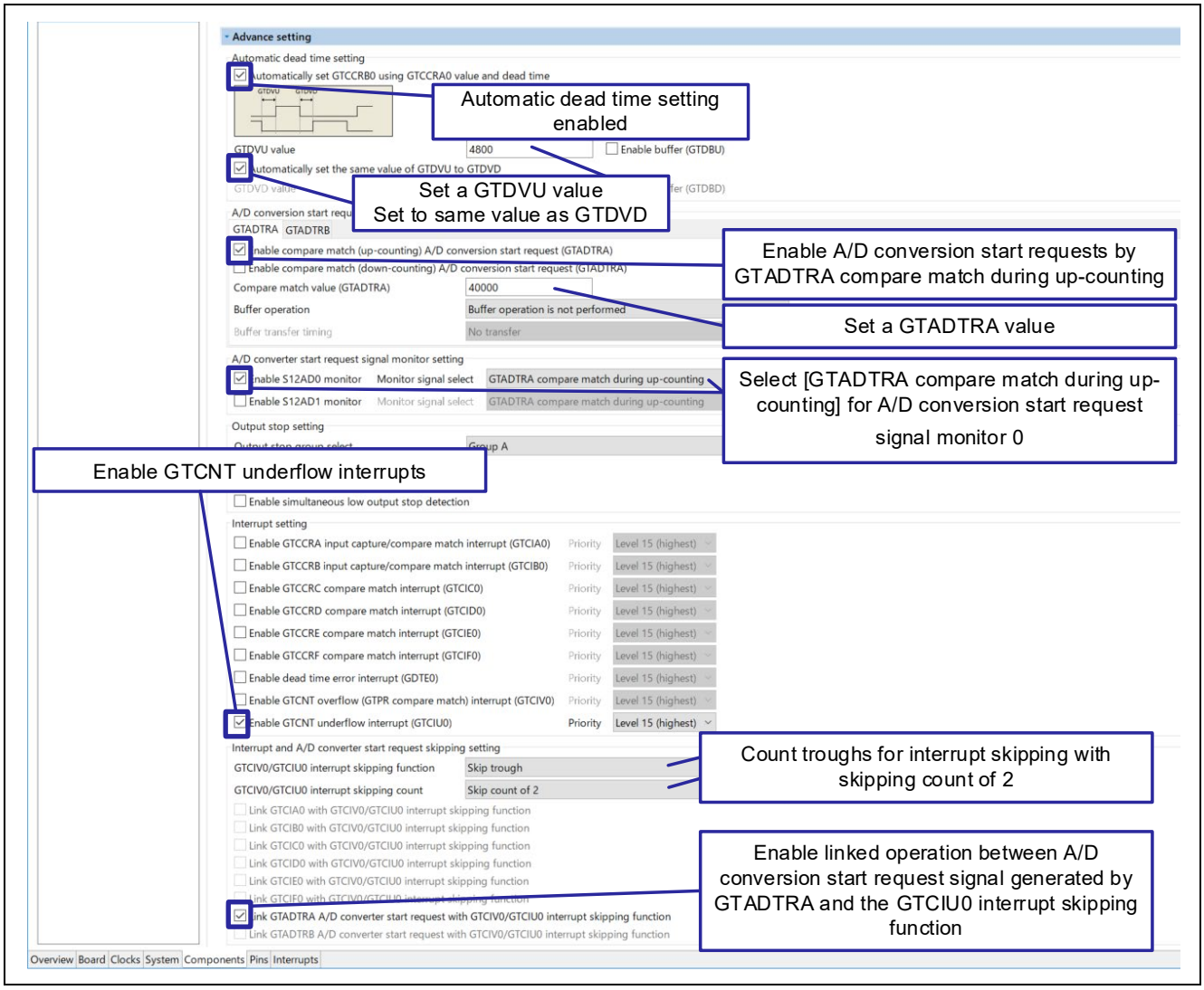


Figure 4.9 GPT0 Settings (2/3)

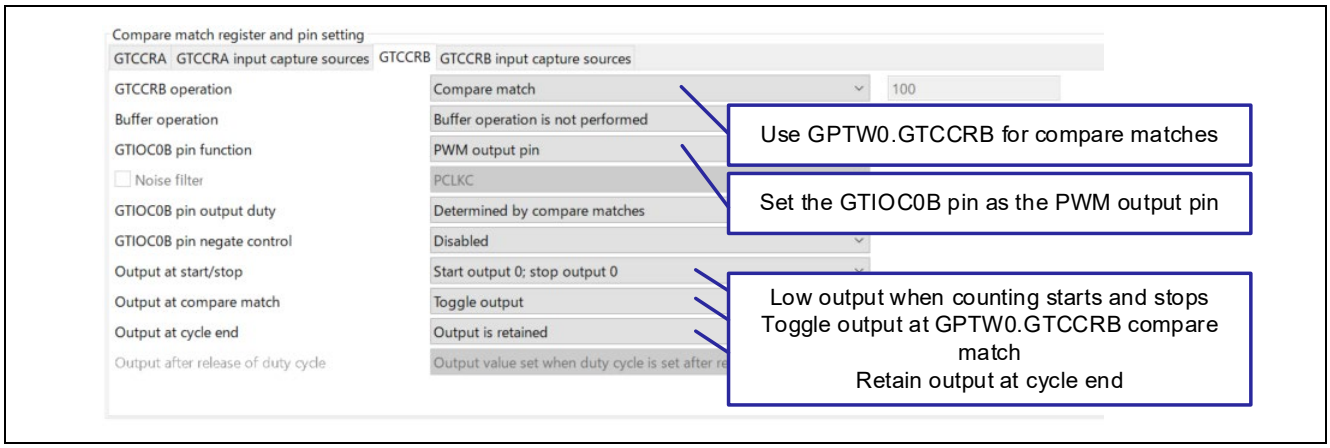


Figure 4.10 GPT0 Settings (3/3)

Table 4.5 Adding Components (S12AD)

Item	Description
Component	Single-scan mode S12AD
Configuration name	Config_S12AD0
Resource	S12AD0

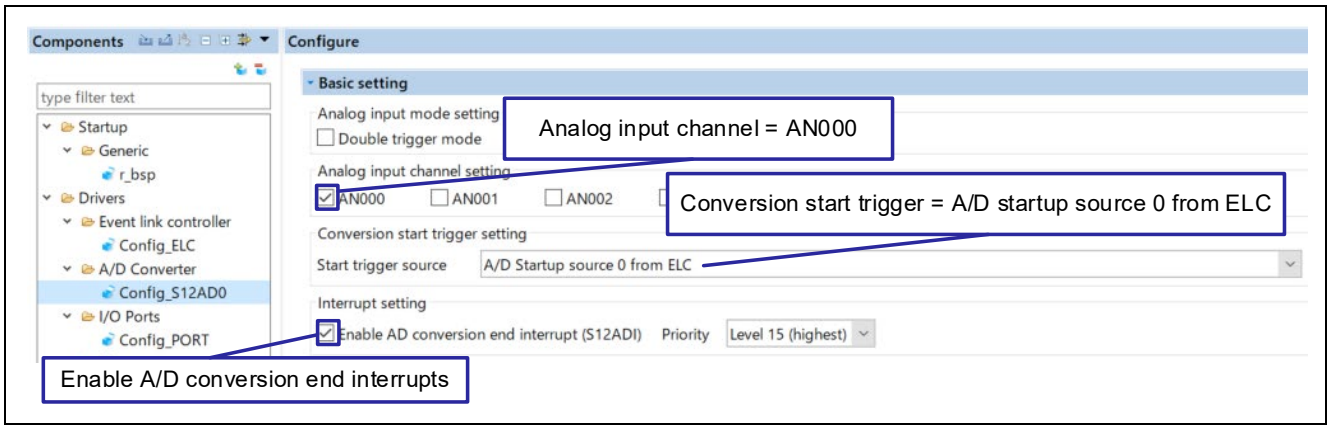


Figure 4.11 S12AD0 Settings

Table 4.6 Adding Components (ELC)

Item	Description
Component	Event Link Controller
Configuration name	Config_ELC
Resource	ELC

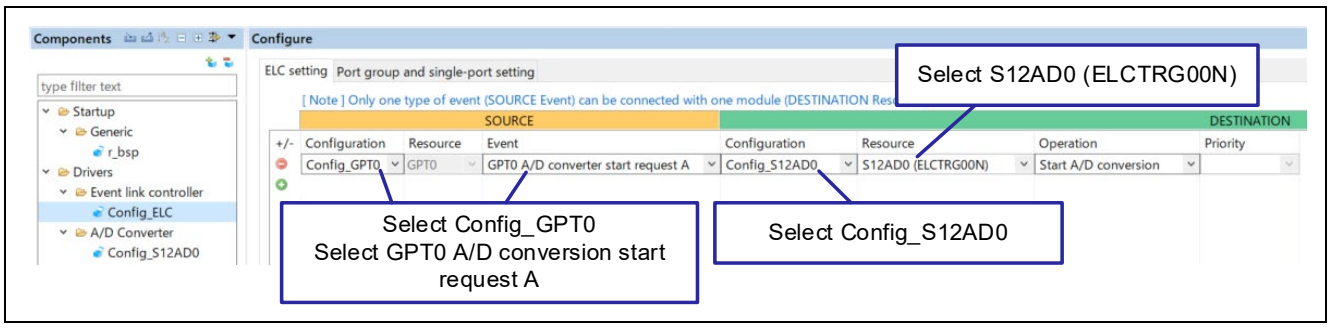


Figure 4.12 ELC Settings

4.2.4 Flowchart

The following flowchart shows the main function processing added after code generation by the Smart Configurator.

In the main function, count start function R_Config_GPT0_Start is read and counting is started.

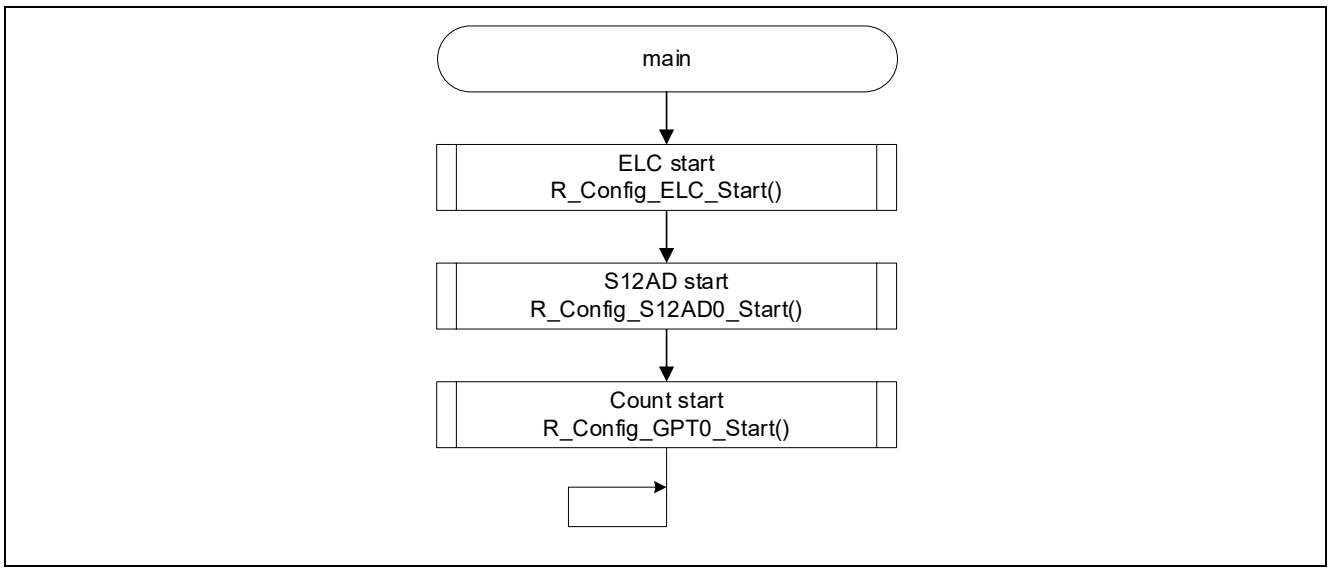


Figure 4.13 main Function

4.2.5 Usage Notes

4.2.5.1 GTITC Register Settings

Refer to section 1.3.3(2), Notes when linking with the interrupt skipping function by using the GTITC register, in this application note.

4.3 A/D Conversion Start Request Function Linked with the Extended Interrupt Skipping Function

- Target sample code file name: r01an6643_rx66t_gptw_ad_delay_2.zip

4.3.1 Overview

This sample code performs the A/D conversion start request function in conjunction with the extended interrupt skipping function of the GPTW.

The A/D conversion start request signal is skipped by extended interrupt skipping counter 1, and a corresponding ELC event signal is sent to S12AD0 to perform A/D conversion. In addition, buffer transfer skipping of extended interrupt skipping counter 2 changes the GTADTRA register value and updates the A/D conversion start request timing.

The following list provides the GPTW, S12AD, and ELC settings used in the sample code.

- GPTW0 (channel 0)
  - Use triangle-wave PWM mode 1
  - Timer counter clock = 160 MHz (PCLKC)
  - Carrier period = 1 ms
  - Use GTPR as the cycle setting register
    - Counts are up-counted from the initial value of 0
  - Use GTCCRA for compare matches of the duty cycle output
    - Use the GTIOC0A pin as the PWM output pin
    - High output when counting starts and stops
    - Toggle output at GTCCRA compare match
    - Retain output at cycle end
  - Use GTCCRB for compare matches of the duty cycle output
    - Use the GTIOC0B pin as the PWM output pin
    - Low output when counting starts and stops
    - Toggle output at GTCCRB compare match
    - Retain output at cycle end
  - Software source count start enabled
  - Use automatic dead time generation
  - Enable A/D conversion start request
    - A/D conversion start requests generated by compare matches between GTCNT and the GTADTRA register during GTCNT up-counting
    - Select [GTADTRA compare match during up-counting] for A/D conversion start request signal monitor 0
  - Use the extended interrupt skipping function
    - Skipping counter 1 count crests with the skipping count of 2
    - Skipping counter 2 count troughs with the skipping count of 3 Initial value: 0*1
    - Skip GTADTRA interrupts if skipping counter 1 is not 0
    - Skip GTADTRA buffer transfers if skipping counter 2 is not 0*1

Note: 1. Set by the user initialization function R_Config_GPT0_Create_UserInit

- S12AD0 (unit 0)
  - Use single scan mode
  - Start A/D conversion by A/D startup source 0 from ELC

- ELC
  - Select GPT0 A/D conversion start request A for an event
  - Select S12AD0 (ELCTRG00N) as the transmission destination resource

Set in Smart Configurator.

Refer to section 4.3.3 for details on the setting method.

(Excluding those marked with an asterisk)

The structure of this sample code is shown below.

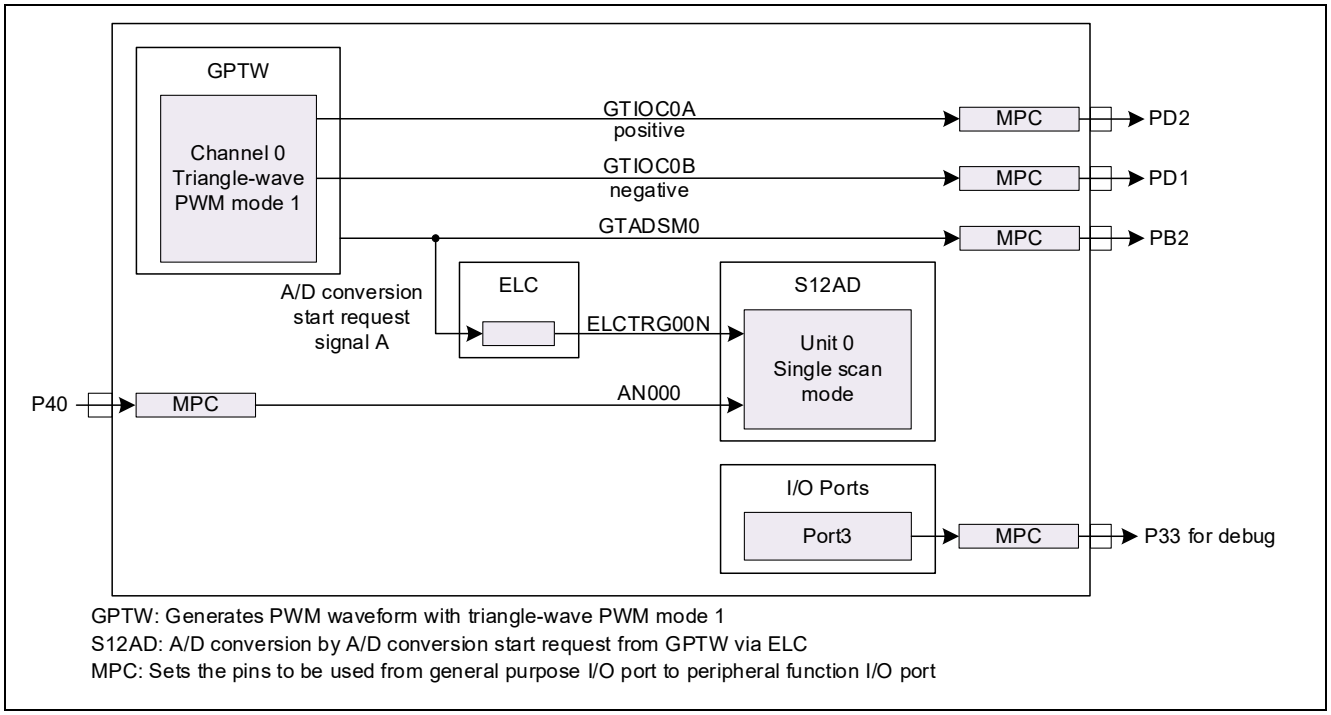


Figure 4.14 Sample Code Structure

4.3.2 Operation Details

This section describes the operation of this sample code.

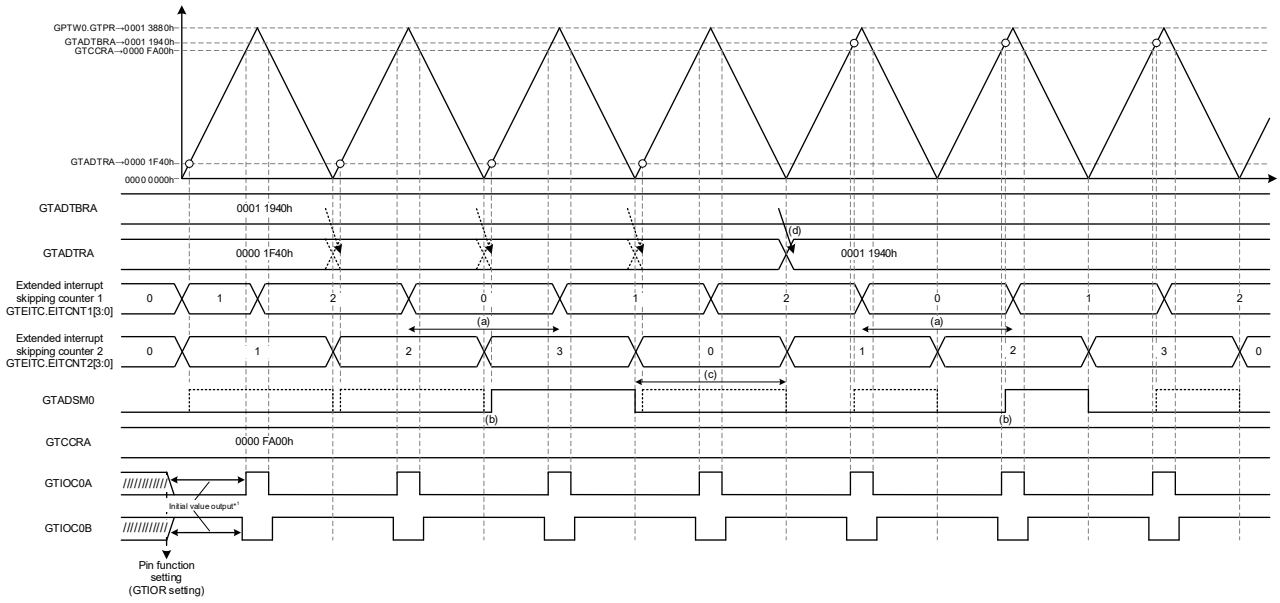
The extended interrupt skipping function skips A/D conversion start requests with skipping counter 1, and buffer transfers of A/D conversion start request timing register GTADTRA with skipping counter 2.

- Skipping counter 1: Used for skipping A/D conversion start requests
  - Setting: GTCNT overflows (crests) counted, skipping count: 2
  - Operation:
    When the counter value is 0, GTADTRA interrupts are enabled ((a) in Figure 4.15), and when a GTADTRA compare match occurs, an A/D conversion start request is generated and the GTADSM0 pin changes ((b) in Figure 4.15).

- Skipping counter 2:
  Used for buffer transfer skipping of A/D conversion start request timing register GTADTRA
  - Setting: GTCNT underflows (troughs) counted, skipping count: 3, initial value: 0
  - Operation:
    When counter value is 0, buffer transfer is enabled ((c) in Figure 4.15), and buffer transfer is performed at troughs ((d) in Figure 4.15).

For debugging purposes, this sample code is set to toggle output on the P33 pin when an A/D conversion end interrupt occurs. If the P33 pin is not to be set to toggle output, change Config_S12AD0_user.c to the following settings.

```
#define PRV_PORT_OUTPUT_ON        (0)
```



Note: 1. An initial value output starts when the PMR register is set.

Figure 4.15 Sample Code Operations
(A/D Conversion Start Request by Compare Match with GTADTRA During Up-Counting,
EITCNT1: Counts Crests with the Skipping Count of 2,
EITCNT2: Counts Troughs with the Skipping Count of 3 with Its Initial Value as 0)

4.3.3 Smart Configurator Settings

The sample code uses the Smart Configurator to add the GPTW, S12AD, and ELC as described below. For details on how to add components, refer to section 4.1.4, Adding Components.

Table 4.7 Adding Components (GPTW0)

Item	Description
Component	General PWM timer
Configuration name	Config_GPT0
Work mode	Triangle-wave PWM mode 1
Resource	GPT0

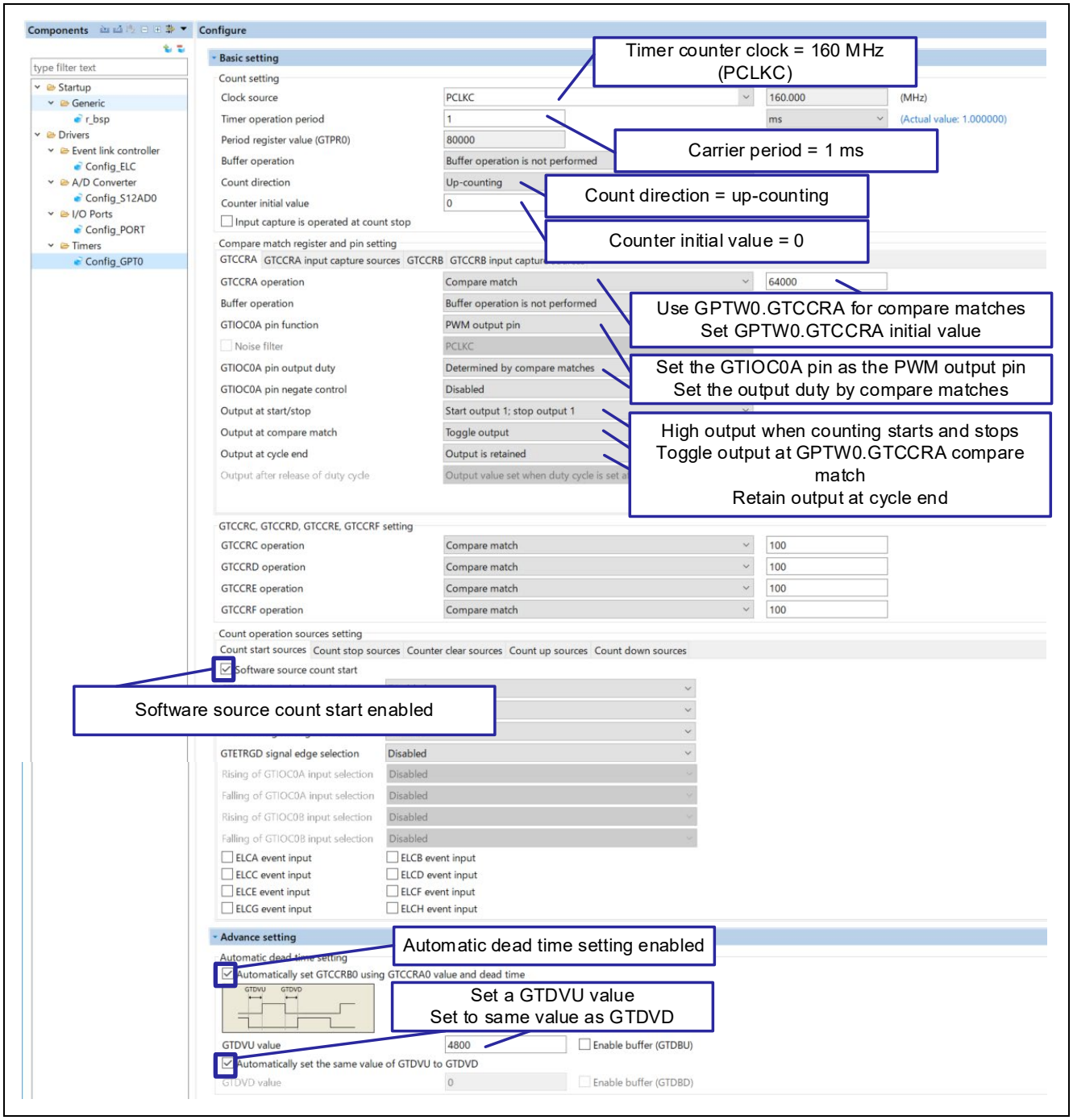
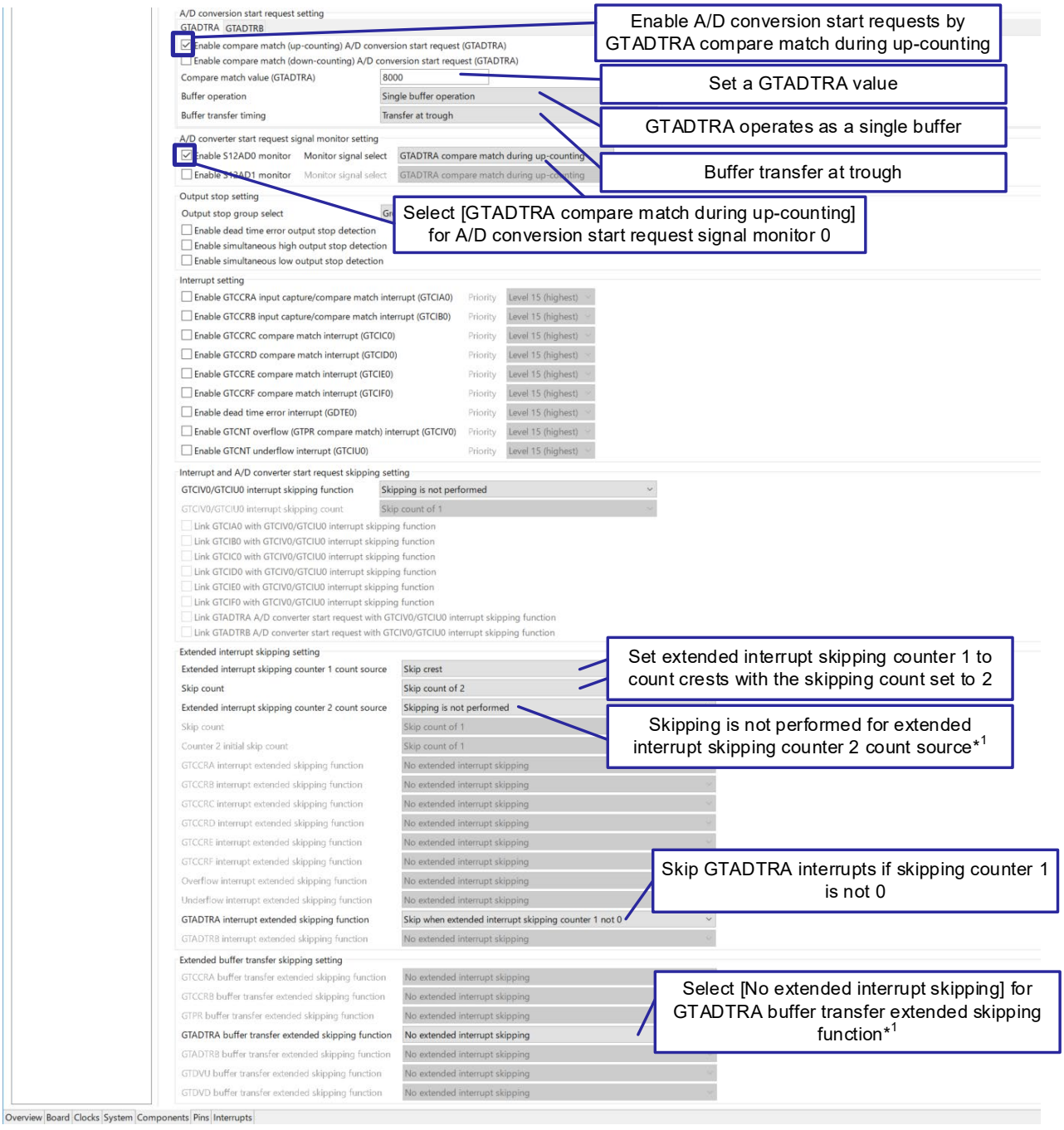


Figure 4.16 GPT0 Settings (1/3)



Note: 1. In the Smart Configurator environment of e² studio Version 2022-10, the initial value of skipping counter 2 cannot be set to 0.
Therefore, the GTADTRA buffer transfer settings using extended interrupt skipping counter 2 (troughs are counted, skipping count: 3, initial value: 0) and counter 2 are performed by the user initialization function R_GPT0_Create_UserInit after code generation by the Smart Configurator. For details, see section 4.3.4, Flowchart.

Figure 4.17 GPT0 Settings (2/3)

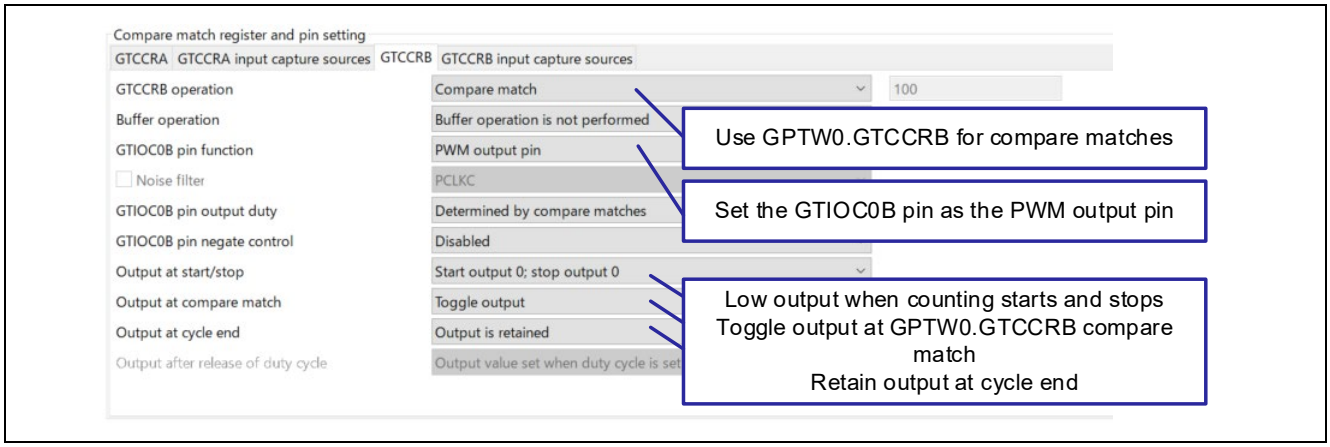


Figure 4.18 GPT0 Settings (3/3)

Table 4.8 Adding Components (S12AD)

Item	Description
Component	Single-scan mode S12AD
Configuration name	Config_S12AD0
Resource	S12AD0

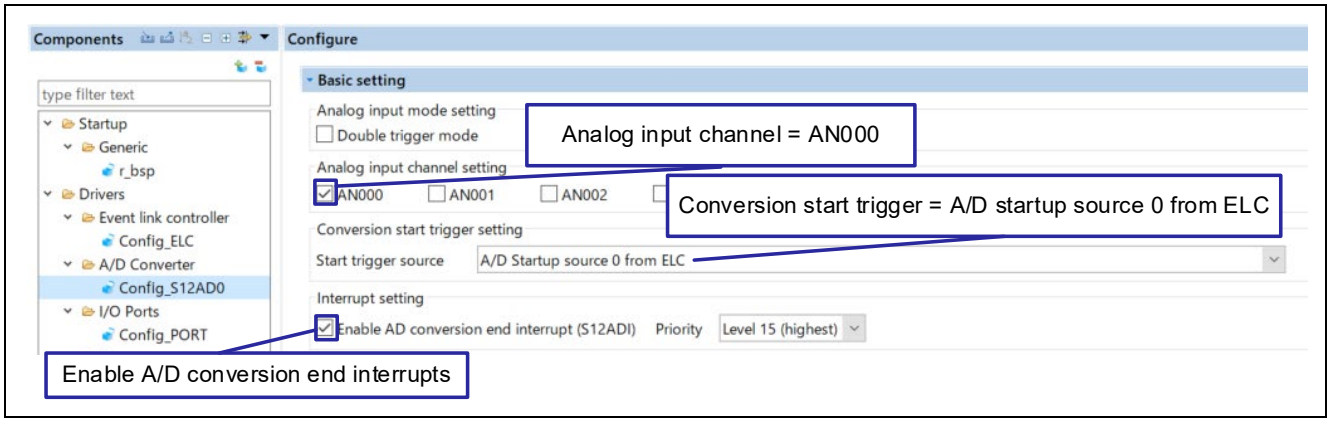


Figure 4.19 S12AD0 Settings

Table 4.9 Adding Components (ELC)

Item	Description
Component	Event Link Controller
Configuration name	Config_ELC
Resource	ELC

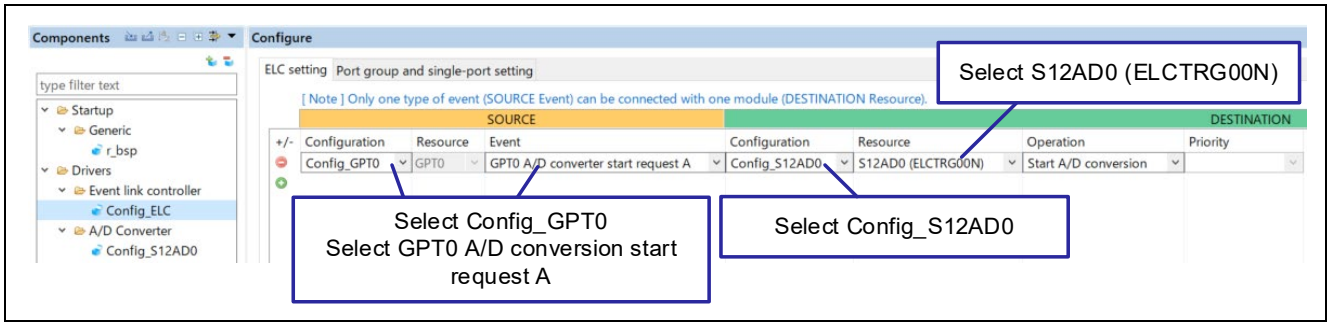


Figure 4.20 ELC Settings

4.3.4 Flowcharts

The following flowchart shows the main function processing added after code generation by the Smart Configurator.

In the main function, count start function R_Config_GPT0_Start is read and counting is started.

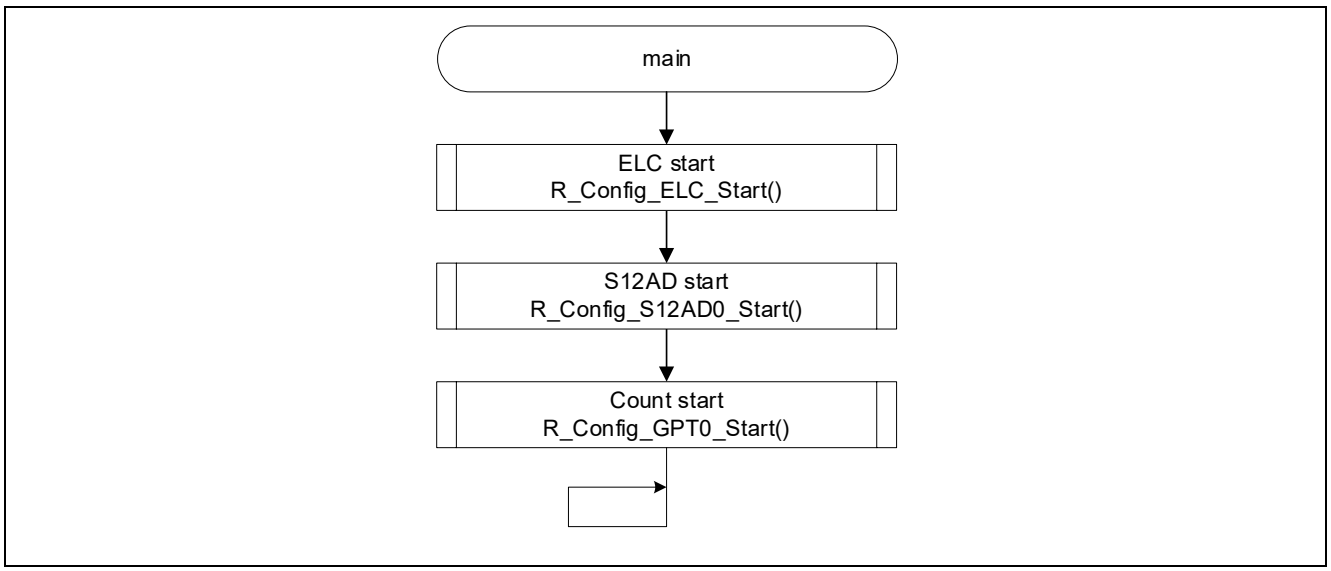


Figure 4.21 main Function

The user initialization function R_Config_GPT0_Create_UserInit, executed before the main function, sets the initial value of the buffer register GTADTBRA and the extended interrupt skipping counter function control registers GTEITC and GTEITLB.

In this sample code, extended interrupt skipping counter 2 is used with an initial value of 0, so code is not generated using the Smart Configurator. Instead, the count source and skipping count of counter 2 are set in the GTEITC register of this function ((a) in Figure 4.22), and the buffer transfer skipping function by using extended interrupt skipping counter 2 is set in the GTEITLB register ((b) in Figure 4.22).

This function is called in the R_Config_GPT0_Create function.

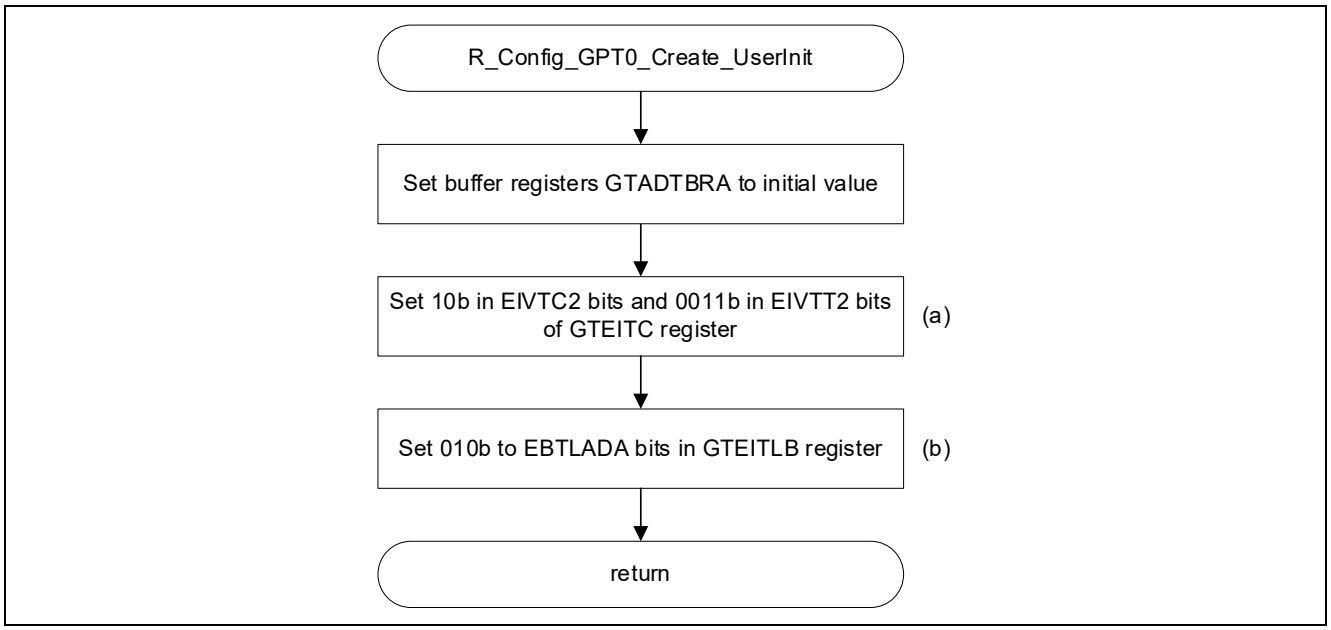


Figure 4.22 User Initialization Function

4.3.5 Related Operation

4.3.5.1 Setting the Initial Value of Extended Interrupt Skipping Counter 2 to 1 or More

When the initial value of extended interrupt skipping counter 2 is set to 1 to 15, code generation can be performed using the Smart Configurator.

Figure 4.23 shows the Smart Configurator settings when the initial value is set to 1 and Figure 4.24 shows the operation.

Select [Skip count of 1] for [Counter 2 initial skip count].

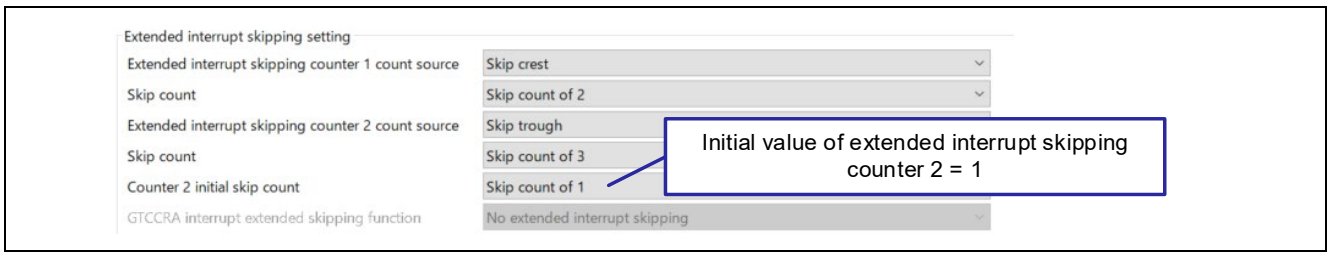


Figure 4.23 GPTW0 Smart Configurator Settings

The operation with the initial value set to 1 differs from the operation with the initial value set to 0 in the sample code (Figure 4.15) in the value of extended interrupt skipping counter 2 before the GTCNT count starts ((a) in Figure 4.24) and the buffer transfer timing ((b) in Figure 4.24).

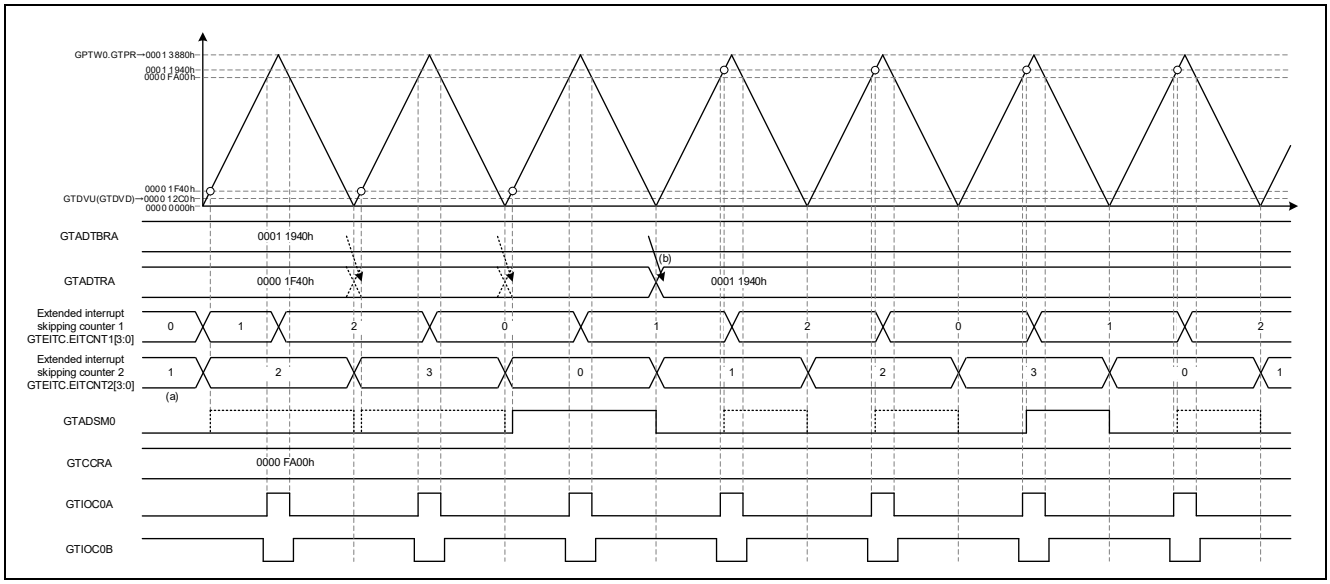


Figure 4.24 Extended Interrupt Skipping Counter 2 Operation with the Initial Value of 1

4.3.6 Usage Notes

4.3.6.1 GTEITLI2 Register Settings

Refer to section 1.3.4(4), Notes when using the extended interrupt skipping function, in this application note.

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.

Also visit the following Renesas Electronics website:

https://www.renesas.com/software-tool/migration-e2studio-to-csplus

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.)

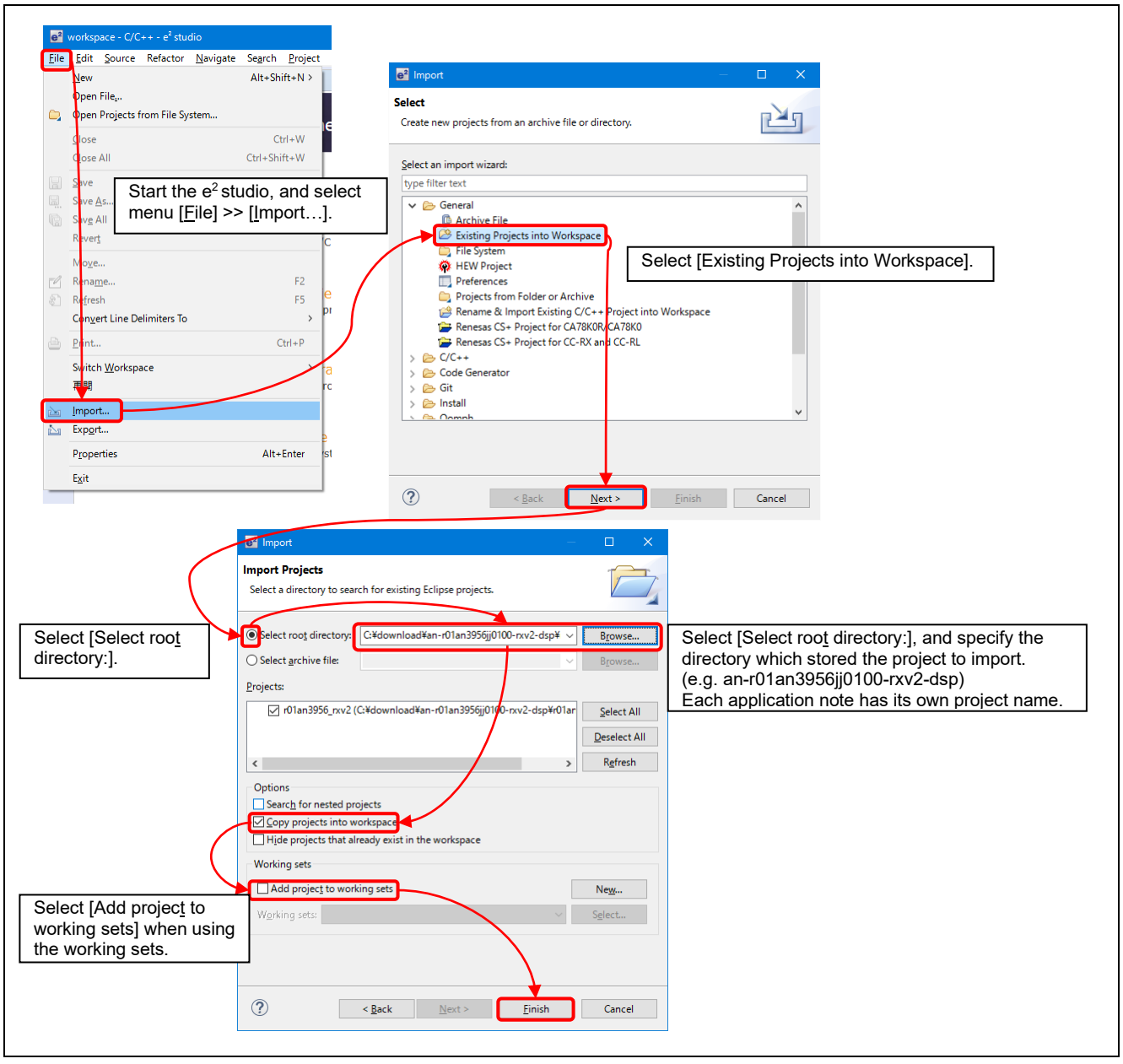


Figure 5.1 How to Import a Project into e² studio

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.)

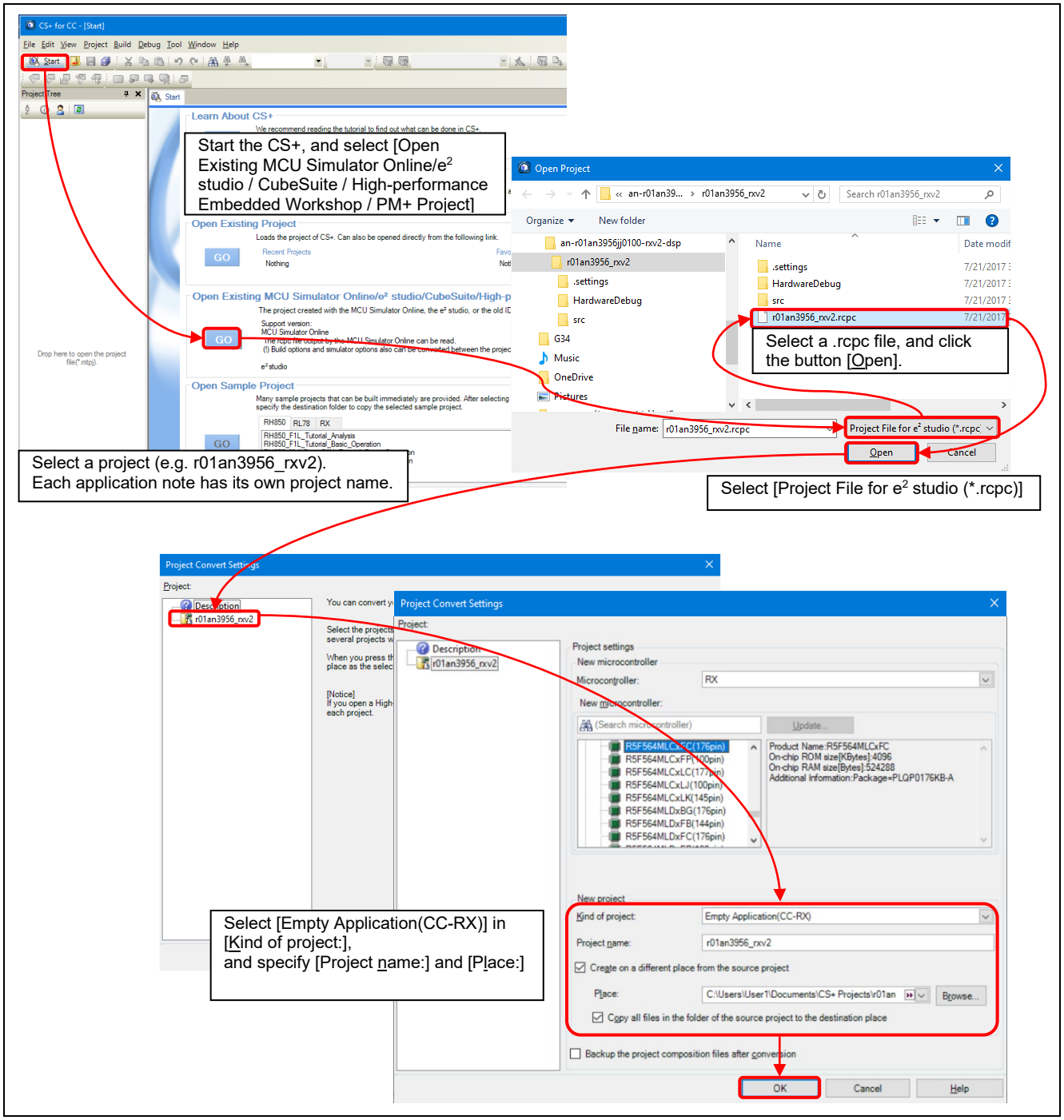


Figure 5.2 How to Import a Project into CS+

6. Reference Documents

- User's Manual: Hardware
  RX66T Group User's Manual: Hardware (R01UH0749)
  (Please obtain the latest version from the Renesas Electronics Corp. website.)
- Technical Updates/Technical News
  (Please obtain the latest version from the Renesas Electronics Corp. website.)
- User's Manual: Development Environment
  RX Family CC-RX Compiler User's Manual (R20UT3248)
  (Please obtain the latest version from the Renesas Electronics Corp. website.)
- User's Manual: Development Environment
  RX66T Group Renesas Starter Kit User's Manual (R20UT4150)
  (Please obtain the latest version from the Renesas Electronics Corp. website.)
- Application Note
  RX Family PWM Output Methods Using MTU3/GPTW (R01AN5995)
  (Please obtain the latest version from the Renesas Electronics Corp. website.)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec. 27, 2022	—	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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