RL78/G24

Sharing A/D Converter by CPU and FAA

Introduction

This application note describes the functions of the RL78/G24 CPU and flexible application accelerator (FAA), and how to share the A/D converter with the CPU and FAA.

Target Device

RL78/G24

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Contents

1. FAA Specifications

1.1 Description of Functions

The flexible application accelerator (FAA) is a Renesas original application accelerator with a Harvard architecture. It can execute 32-bit multiplication, addition, and subtraction in a single cycle. The FAA can directly access some peripheral functions and combined operation of the CPU and FAA is also possible, thus enabling improvement of the system characteristics.

For details about FAA option settings, tool operation, and debug method, see the **Flexible Application Accelerator (FAA) Tool Guide: CS+ (e^2 studio)**.

Figure 1-1 provides an overview of the FAA. Table 1-1 describes the functions of the FAA.

Figure 1-1 Image of the FAA

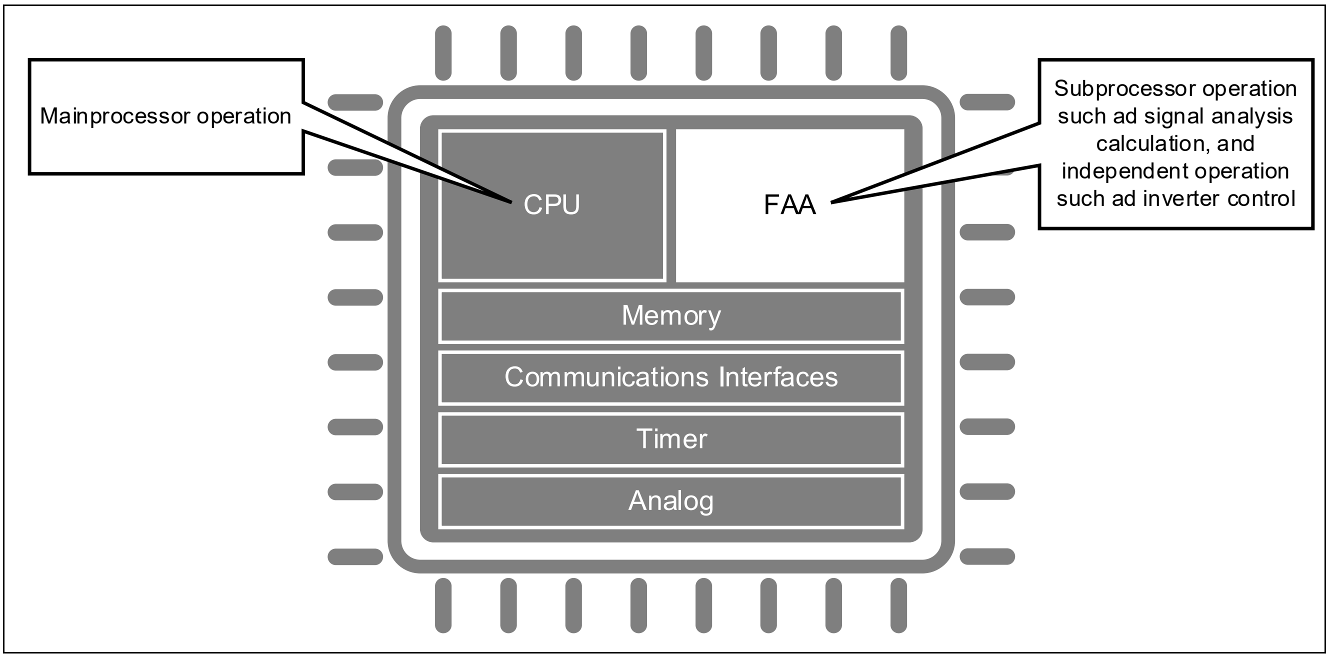

Mainprocessor operation
CPU
FAA
Subprocessor operation such ad signal analysis calculation, and independent operation such ad inverter control
Memory
Communications Interfaces
Timer
Analog

Table 1-1 Functions of the FAA

Item	Function
Processor	A single operation instruction is executed in a single cycle. (When multiple operation instructions are executed consecutively, the second and the subsequent instructions are executed in two cycles each.) • Multiplication: 32-bit signed × 32-bit signed → 32-bit signed Results of multiplication (64-bit) can be right-shifted by a desired number of bits. • Addition: 32-bit signed + 32-bit signed → 32-bit signed Internally calculated with 33-bit precision • Subtraction: 32-bit signed – 32-bit signed → 32-bit signed Internally calculated with 33-bit precision • Limit operation: Operation parameter registers (33 bits × 4) in which upper and lower limits can be set. Processor internal registers • Operation parameter registers (32 bits × 6) • Address pointer registers (12 bits × 6) • Flag bit register (16 bits × 1 (including 4 valid flag bits))
Memory	• Instruction code memory: 4Kbytes • Data memory: 2Kbytes • Data shared memory (SHDMEM) 32 bytes
Interruption	• Multiple interrupts available • Interrupt sources Input event detection interrupts: 10 Timing compare-match interrupts: 6
Input event processing	• Input channels: 10 • Detected edges: Rise, fall, or both
Timing processing	• Reference timing counter bits: 24 • Compare-match channels: 6

1.2 Allocating Peripheral Functions

The FAA can control the following RL78/G24 peripheral functions.

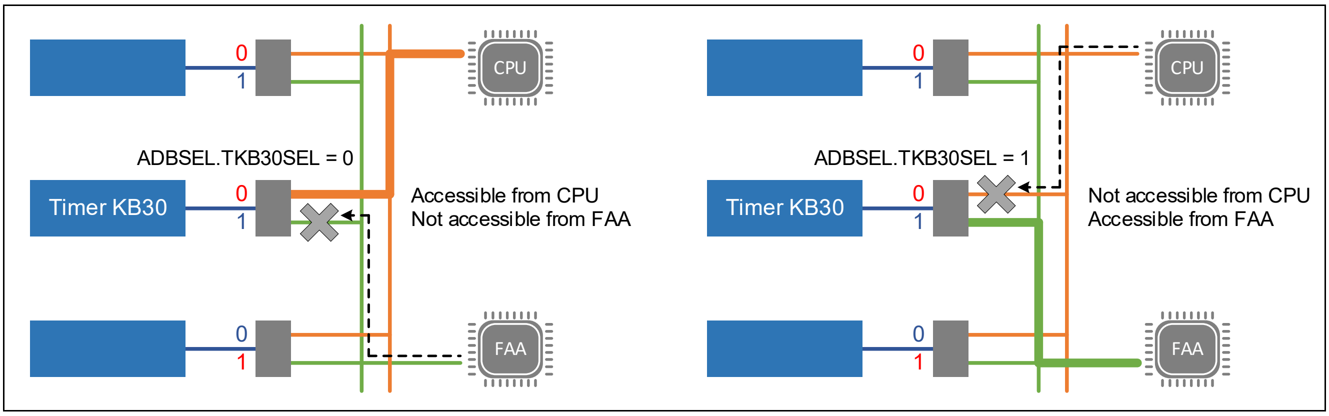
- A/D converter
- D/A converter
- Timer RD2/PWM option unit A (PWMOPA)
- Timer RG2
- Timer RX
- 16-bit timers KB30, KB31, and KB32
- Comparator
- Programmable gain amplifier
- Serial array unit
- Timer array unit
- Port function (port 1)
- Security function (TRNG)

The address bus select register (ADBSEL) allows you to select whether these functions are allocated to the CPU or the FAA. Setting 0 in the bus select bit (xxSEL) corresponding to the peripheral function enables bus access from the CPU. Setting 1 in this bit enables bus access from the FAA. The FAA cannot access any peripheral functions allocated to the CPU. This also applies vice versa.

For details about the address bus select register (ADBSEL), see the **RL78/G24 User's Manual: Hardware (R01UH0961)**.

Figure 1-2 provides an overview of the address bus select function. For allocation in the address bus select register (ADBSEL) of the sample code in this application note, see Figure 2-1.

Figure 1-2 Overview of Address Bus Select Function



2. Application Specifications

2.1 Overview of Specifications

This application note uses two processors (CPU and FAA), each of which performs A/D conversion of analog input and outputs the conversion results.

The A/D converter is allocated to the FAA. When the CPU performs A/D conversion, the FAA controls the A/D converter in place of the CPU. The PWM duty ratio output by TKB31 for the CPU or output by TKB30 for the FAA is changed according to the conversion result.

The CPU program specifies the initial settings of the A/D converter, TKB30, TKB31, and timer array unit (TAU0), and then starts the FAA operation at fixed intervals (10 μs) by using a TAU channel 0 count end interrupt (INTTM00). It also requests the FAA to perform A/D conversion and, after the conversion is complete, acquires the conversion result and changes the duty ratio of TKB31.

The FAA program controls the A/D converter and TKB30. This program starts operation when an INTTM00 is generated, performs A/D conversion of the P22/ANI2 pin, changes the duty ratio of TKB30, and then stops operation. If A/D conversion is requested by the CPU, the FAA program also performs A/D conversion of the P23/ANI3 pin, stores the conversion result in SHDMEM, and then stops operation.

Table 2-1 shows the peripheral functions and their usage.

Table 2-1 Peripheral Functions and Their Usage

Peripheral Function	Usage
Flexible application accelerator (FAA)	Controls the A/D converter and TKB30, and changes the duty ratio of the PWM output from the TKBO00 pin
A/D converter (Advanced mode enabled)	Performs A/D conversion of analog input voltage of the P22/ANI2 and P23/ANI3 pins
16-bit timer KB30 (TKB30)	PWM output from the TKBO00 pin
16-bit timer KB31 (TKB31)	PWM output from the TKBO10 pin
Timer array unit (TAU)	Generates a count end interrupt (INTTM00) at 10 μs intervals
Event Link Controller (ELC)	Links a TAU channel 0 count end interrupt (INTTM00) with an FAA input event

Figure 2-1 shows allocation in the address bus select register (ADBSEL).

Figure 2-1 Allocation in Address Bus Select Register (ADBSEL)

	15	14	13	12	11	10	9	8
ADBSEL	FAADIVSEL	0	TRNGSEL	PORTSEL	TKB32SEL	TKB31SEL	TKB30SEL	TRGSEL
Setting value	0	0	0	0	0	0	1	0

7	6	5	4	3	2	1	0
TRD0SEL	PWMOP SEL	TRXSEL	DACSEL	PGACMP SEL	ADCSEL	SAU0SEL	TAU0SEL
0	0	0	0	0	1	0	0

0	Bus access by the CPU is enabled.
1	Bus access by the FAA is enabled.

TKB31SEL: 16-bit timer KB31

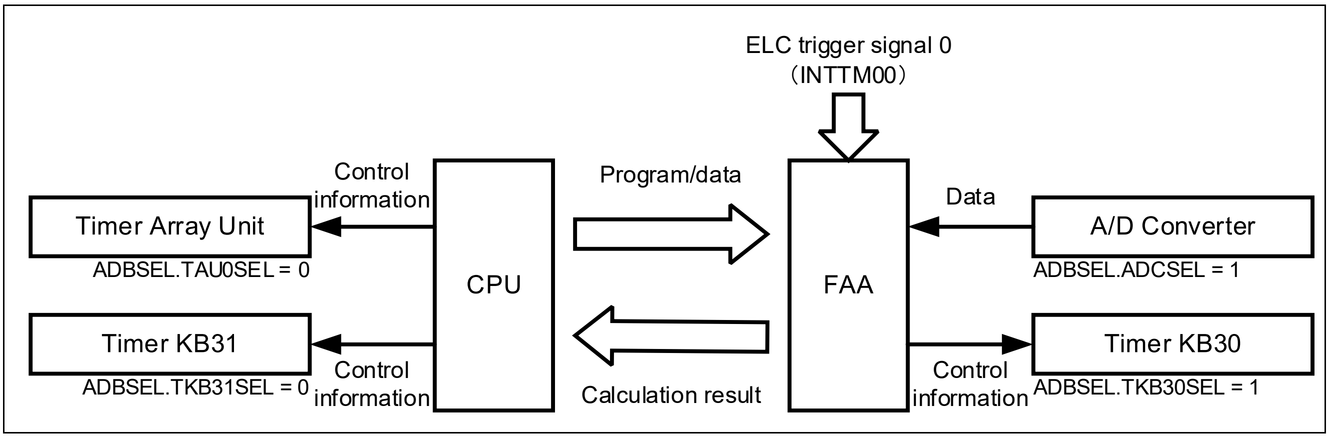
TKB30SEL: 16-bit timer KB30

ADCSEL: A/D converter

TAR0SEL: Timer array unit

Figure 2-2 provides an overview of processor operation.

Figure 2-2 Overview of Processor Operation



2.2 Operation

In this sample code, the 16-bit timer KB30 (TKB30) and 16-bit timer KB31 (TKB31) are used in standalone mode (period controlled by the TKBCR00 register), and the P12/TKBO00 and P14/TKBO10 pins are used for PWM output. The duty ratio of PWM output uses the results of conversion by the A/D converter.

With the TAU0 period set to 10 μs, the CPU starts the FAA by using an interrupt (INTTM00) that is generated each time channel 0 count ends. Then, the FAA performs A/D conversion and then compares the duty ratio of the PWM output of TKB30.

1. [CPU program] Controls operation of the A/D converter, TKB310, and TKB31.
2. [CPU program] Sets SFR access of the A/D converter and TKB30 to the FAA bus.
3. [CPU program] Starts operating TAU0, and then requests the FAA to perform A/D conversion for the CPU.
4. [CPU program] After the timer starts operation, a TAU channel 0 count end interrupt (IMTTM00) is generated every 10 μs to enable clock supply to the FAA, and then FAA operation is enabled.
5. [CPU program] Sets the FAA stack pointer and the start address of the FAA program, and then starts FAA operation.
6. [FAA program] Performs A/D conversion of the P22/ANI2 pin, and then updates the compare register (TKBCR01) according to the conversion result and changes the duty ratio of TKBO00 output.
7. [FAA program] If A/D conversion is requested by the CPU, the FAA performs A/D conversion of the P23/ANI3 pin, stores the results in SHDMEM, and then stops operation.
8. [CPU program] Stops clock supply to the FAA when execution of the FAA program is complete, and then disables FAA operation.
9. [CPU program] Acquires the conversion result from SHDMEM if the FAA has completed A/D conversion normally, and then updates the compare register (TKBCR11) according to the conversion result and changes the duty ratio of TKBO10 output.
10. [CPU program] Requests the FAA to perform the next A/D conversion, and then enters HALT mode.
11. [CPU program] Returns to step 4, and then waits for generation of a TAU channel 0 count end interrupt (INTTM00) again.

3. Operation Confirmation Conditions

The sample code described in this application note has been confirmed under the following conditions.

Table 3-1 Operation Confirmation Conditions

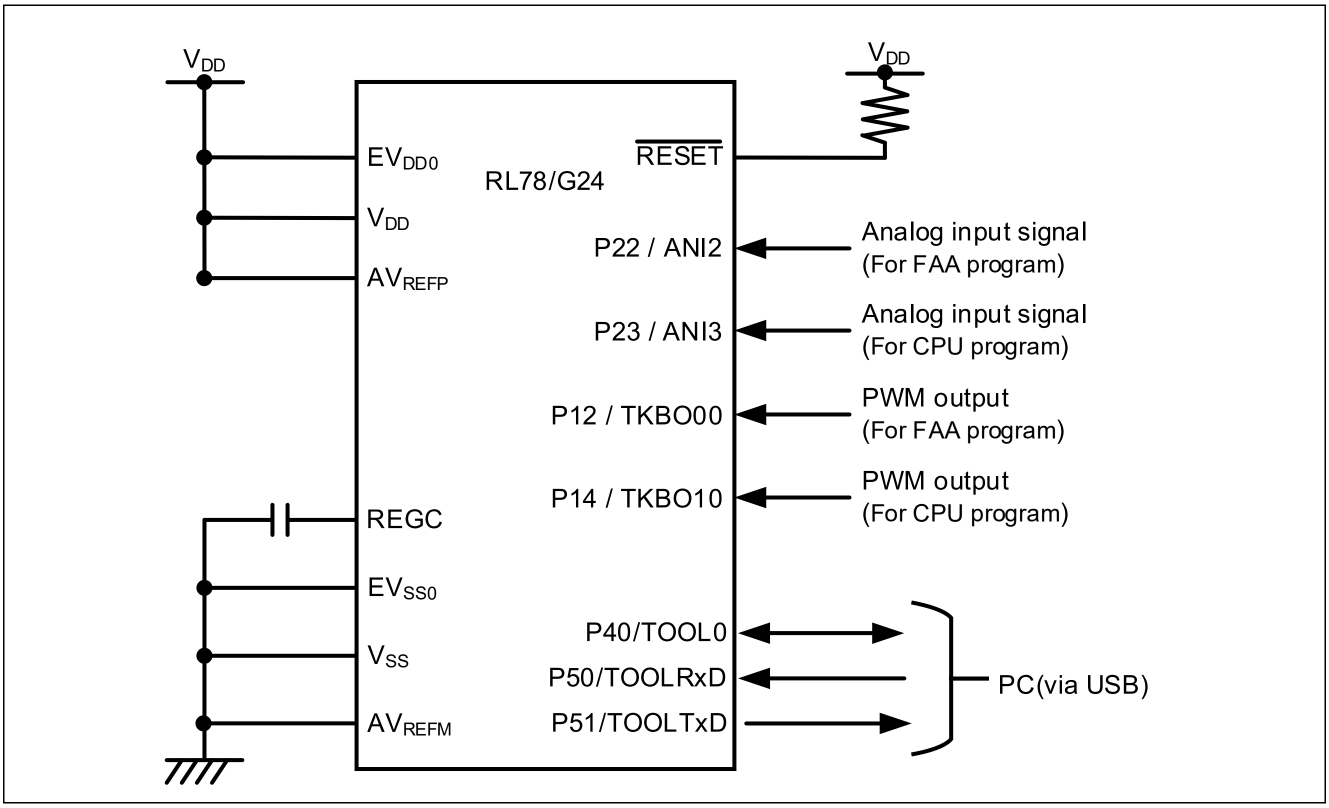
Item	Description
MCU used	RL78/G24 (R7F101GLG)
Operating frequency	• High-Speed On-Chip Oscillator Clock (f_{HOCO}): 8MHz • PLL Oscillator Circuit Output (f_{PLL}): 96MHz • CPU/Peripheral Hardware Clock (f_{CLK}): 48MHz
Operating voltage	• 3.3V (Can operate between 2.7V to 5.5V) • LVD0 Operation (V_{LVD0}): Reset Mode Rising edge TYP. 2.97V Falling edge TYP. 2.91V
Integrated development environment (CS+)	CS+ for CC V8.12.00 Manufactured by Renesas Electronics
C compiler (CS+)	CC-RL V1.14.00 Manufactured by Renesas Electronics
Integrated development environment (e^2 studio)	e^2 studio 2024-10 (24.10.0) Manufactured by Renesas Electronics
C compiler (e^2 studio)	CC-RL V1.14.00 Manufactured by Renesas Electronics
Integrated development environment (IAR)	IAR Embedded Workbench for Renesas RL78 V5.10.3 Manufactured by IAR Systems
C compiler (IAR)	
Smart Configurator	V.1.11.0
Board Support Package (r_bsp)	V.1.70
Emulator	CS+, e^2 studio: COM port IAR: E2 Emulator Lite
Board used	RL78/G24 Fast Prototyping Board (RTK7RLG240C00000BJ)

4. Hardware Description

4.1 Example of Hardware Configuration

Figure 4-1 shows the hardware configuration example used in the sample code for this application.

Figure 4-1 Example of Hardware Configuration



Note 1. This simplified circuit diagram was created to show an overview of connections only. When actually designing your circuit, make sure the design includes appropriate pin handling and meets electrical characteristic requirements (connect each input-only port to V_{DD} or V_{SS} through a resistor).

Note 2. Connect any pins whose name begins with EV_{SS} to V_{SS}, and any pins whose name begins with EV_{DD} to V_{DD}, respectively.

Note 3. V_{DD} must not be lower than the reset release voltage (V_{LVD0}) that is specified for the LVD0.

4.2 List of used Pins

Table 4-1 shows the pins used and their functions.

Table 4-1 Pins used and Their Functions

Pin name	I/O	Function
P22 / ANI2	Input	A/D Converter Analog Input (for the FAA program)
P23 / ANI3	Input	A/D Converter Analog Input (for the CPU program)
P12 / TKBO00	Output	PWM Output (for the FAA program)
P14 / TKBO10	Output	PWM Output (for the CPU program)

Caution: In this application note, only the used pins are processed. When designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.

5. Software Description

5.1 Smart Configurator Settings

The following describes the Smart Configurator settings in this sample code. The items and their descriptions in each table in the Smart Configurator settings are contained in the description of the configuration screen.

5.1.1 System settings

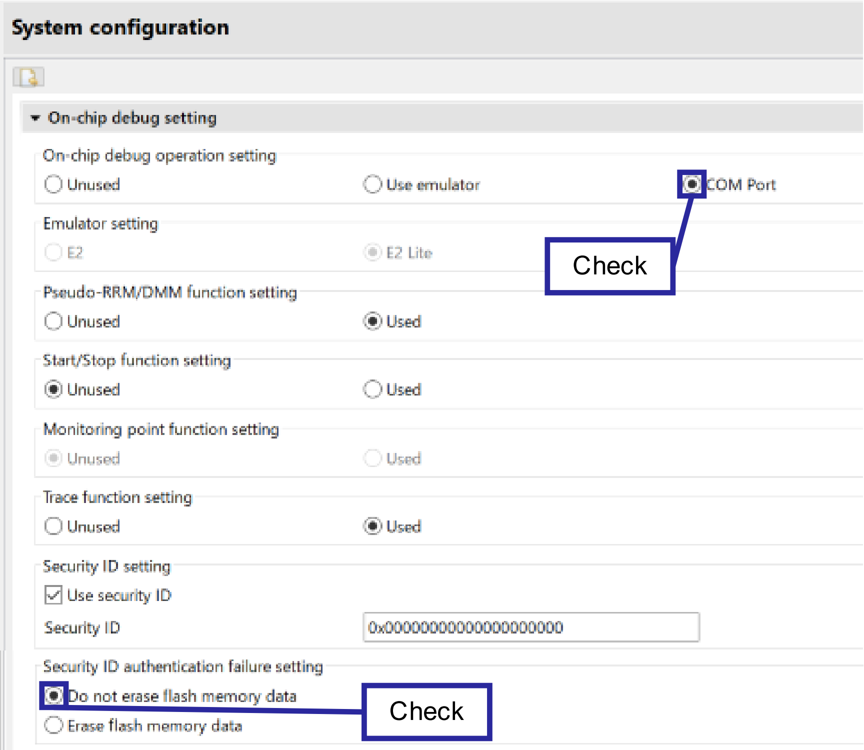
The following shows the system settings used in this sample code.

Note that the system settings used in this sample code are the same for integrated development environments e^2 studio and CS+ but are different for IAR. Specify appropriate settings according to your environment.

Figure 5-1 shows the system settings used in this sample code (e^2 studio and CS+).

To perform COM port debugging on the RL78/G24 Fast Prototyping Board (RTK7RLG240C00000BJ), you need to appropriately specify the settings in the integrated development environment (e^2 studio and CS+). For details, see **7.1 Using COM Port Debugging with the e^2 studio** in the **RL78/G24 Fast Prototyping Board User's Manual (R20UT5091J)**.

Figure 5-1 System Settings (e^2 studio and CS+)



System configuration

On-chip debug setting

On-chip debug operation setting
○ Unused ○ Use emulator ● COM Port

Emulator setting
○ E2 ● E2 Lite

Pseudo-RRM/DMM function setting
○ Unused ● Used

Start/Stop function setting
● Unused ○ Used

Monitoring point function setting
● Unused ○ Used

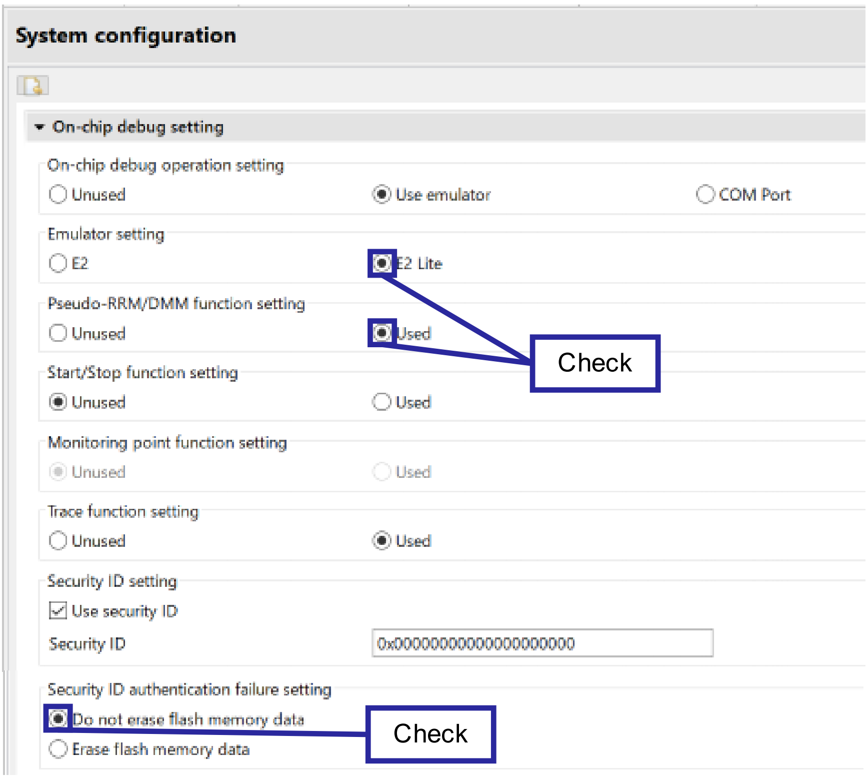
Trace function setting
○ Unused ● Used

Security ID setting
☑ Use security ID
Security ID: 0x00000000000000000000

Security ID authentication failure setting
● Do not erase flash memory data
○ Erase flash memory data

Figure 5-2 shows the system configurations used in this sample program for IAR.

Figure 5-2 System Configurations (IAR)



System configuration

▾ On-chip debug setting

On-chip debug operation setting
○ Unused ● Use emulator ○ COM Port

Emulator setting
○ E2 ● E2 Lite

Pseudo-RRM/DMM function setting
○ Unused ● Used

Start/Stop function setting
● Unused ○ Used

Monitoring point function setting
● Unused ○ Used

Trace function setting
○ Unused ● Used

Security ID setting
☑ Use security ID
Security ID 0x00000000000000000000

Security ID authentication failure setting
● Do not erase flash memory data
○ Erase flash memory data

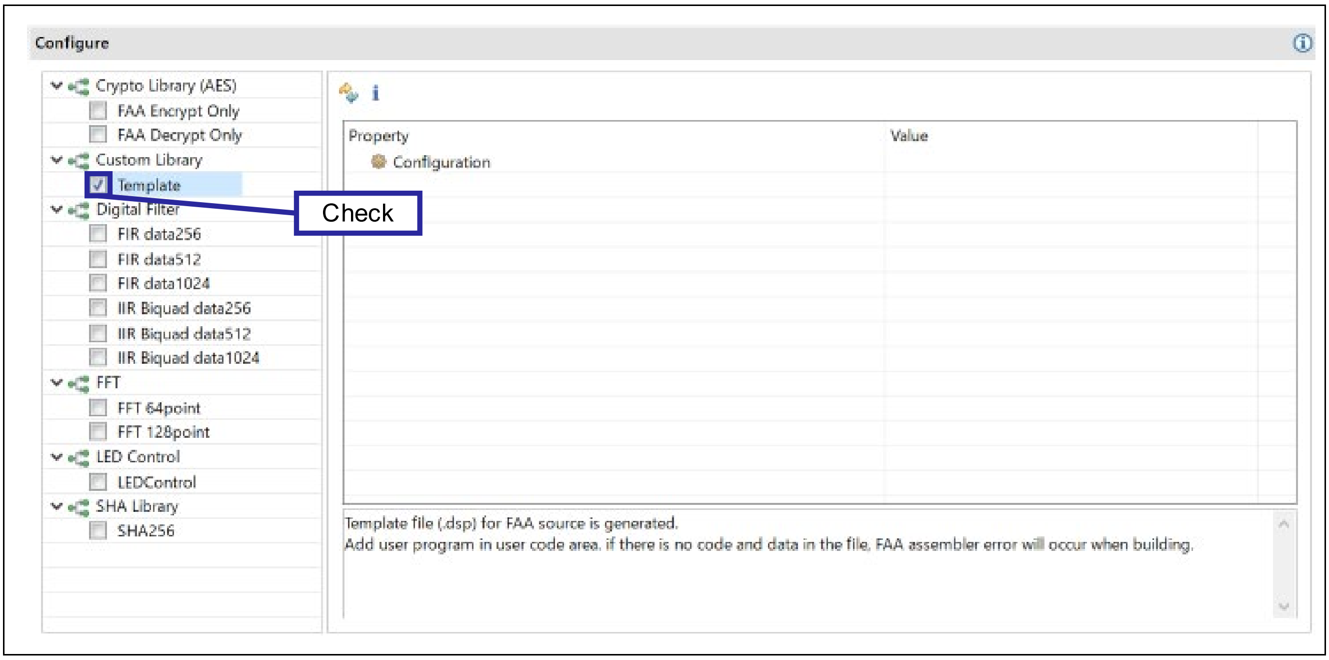
5.1.2 Component Configurations

This section presents the component configurations used in this sample program.

Table 5-1 Component Configurations (FAA)

Item	Content
Component	Flexible Application Accelerator
Configuration Name	Config_FAA
Resource	FAA

Figure 5-3 Configuration of FAA



Configure

- Crypto Library (AES)
  - □ FAA Encrypt Only
  - □ FAA Decrypt Only
- Custom Library
  - ☑ Template
- Digital Filter
  - □ FIR data256
  - □ FIR data512
  - □ FIR data1024
  - □ IIR Biquad data256
  - □ IIR Biquad data512
  - □ IIR Biquad data1024
- FFT
  - □ FFT 64point
  - □ FFT 128point
- LED Control
  - □ LEDControl
- SHA Library
  - □ SHA256

Property	Value
Configuration	

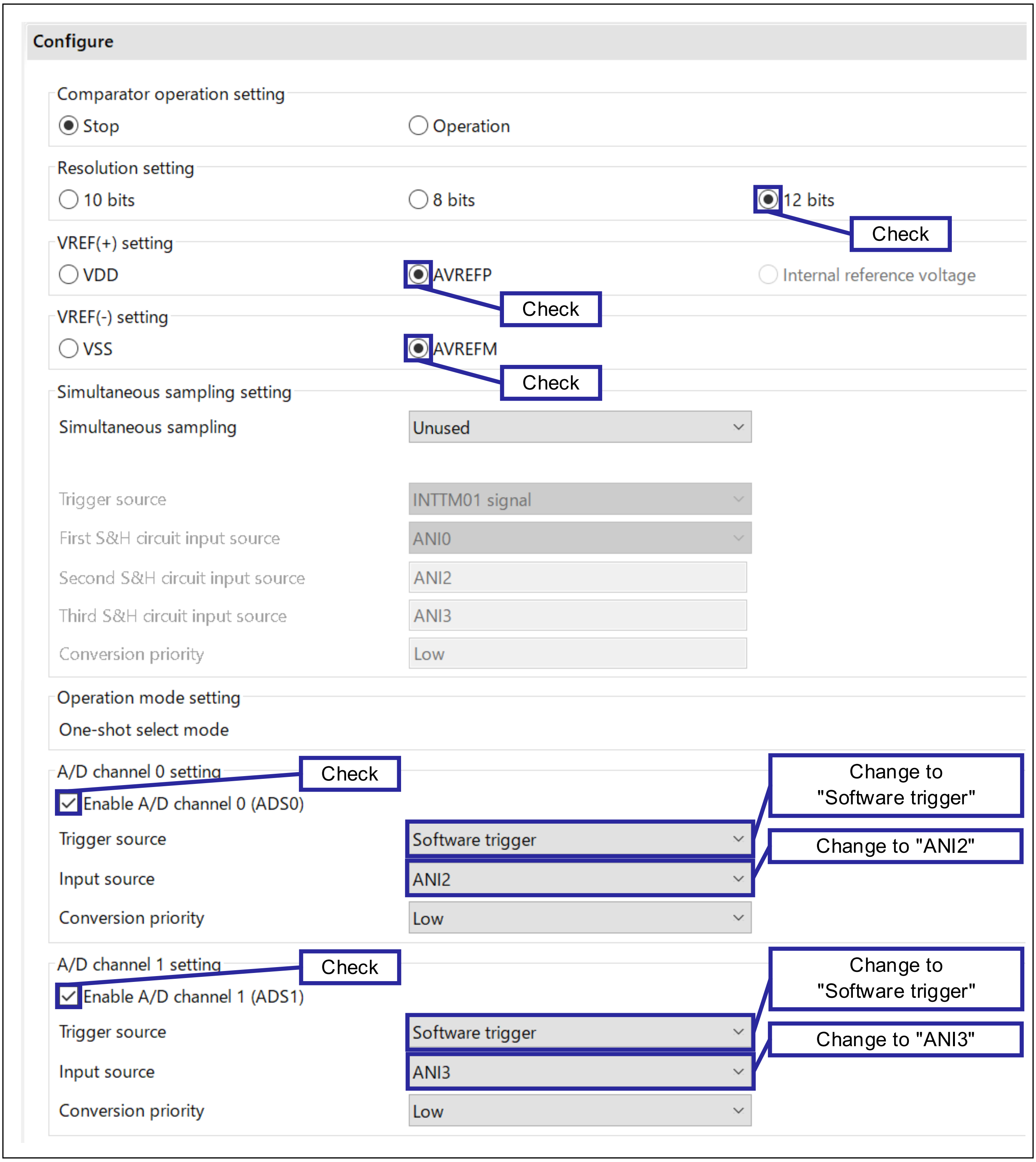
Template file (.dsp) for FAA source is generated.
Add user program in user code area. if there is no code and data in the file, FAA assembler error will occur when building.

Note After loading the sample code, if the FAA library is not displayed, please refer to "2.3.1 Adding FAA Components" in the "Flexible Application Accelerator (FAA) Tool Guide for CS+ (e2 studio edition)" and download the FAA library.

Table 5-2 Component Configurations (A/D Converter)

Item	Content
Component	A/D Converter
Configuration Name	Config_ADC
Resource	ADC
Operation Mode	Advanced mode

Figure 5-4 Configuration of A/D Converter (1/2)



Configure

Comparator operation setting
● Stop ○ Operation

Resolution setting
○ 10 bits ○ 8 bits ● 12 bits

VREF(+) setting
○ VDD ● AVREFP ○ Internal reference voltage

VREF(-) setting
○ VSS ● AVREFM

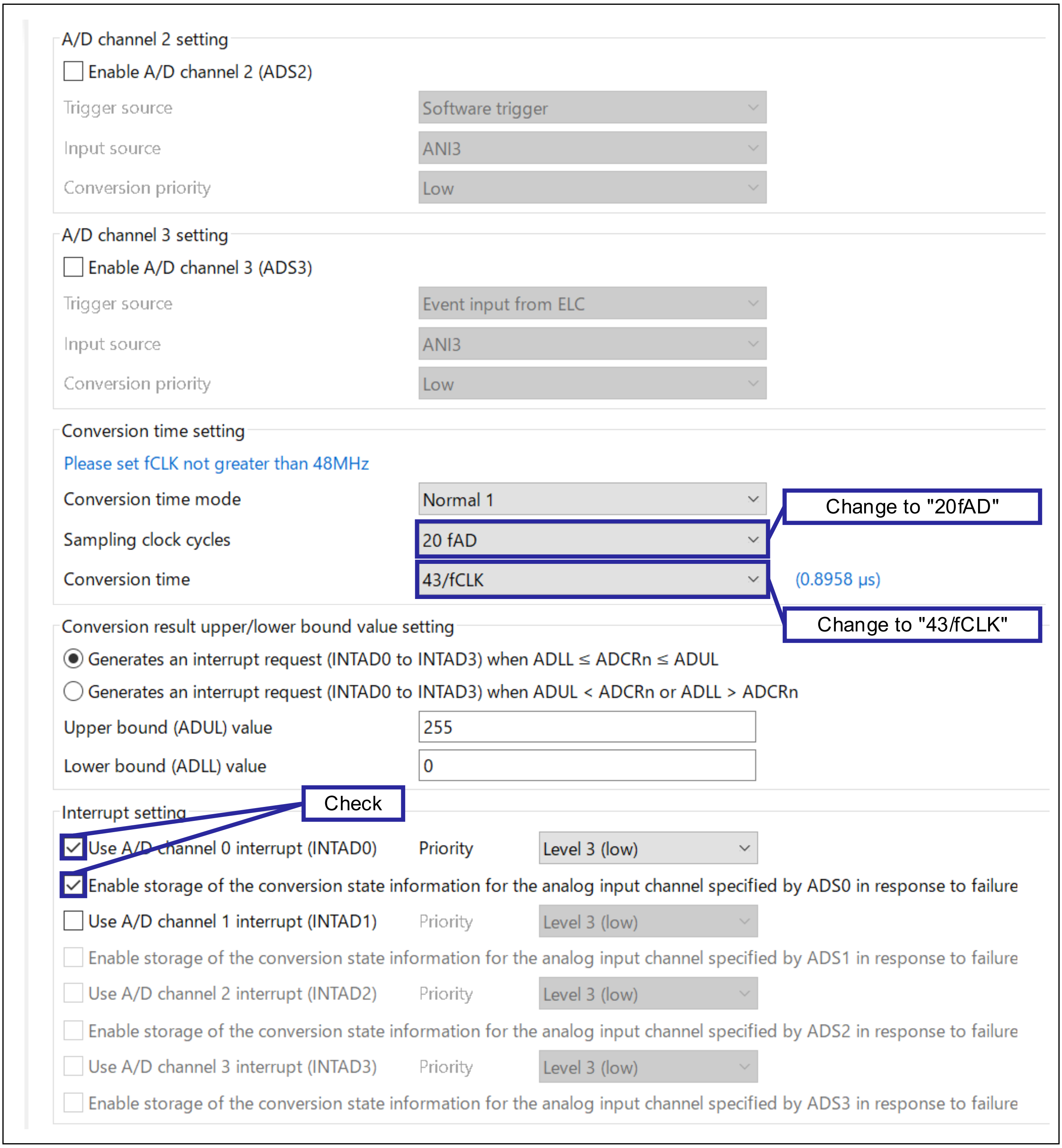
Simultaneous sampling setting
Simultaneous sampling: Unused
Trigger source: INTTM01 signal
First S&H circuit input source: ANI0
Second S&H circuit input source: ANI2
Third S&H circuit input source: ANI3
Conversion priority: Low

Operation mode setting
One-shot select mode

A/D channel 0 setting
☑ Enable A/D channel 0 (ADS0)
Trigger source: Software trigger
Input source: ANI2
Conversion priority: Low

A/D channel 1 setting
☑ Enable A/D channel 1 (ADS1)
Trigger source: Software trigger
Input source: ANI3
Conversion priority: Low

Figure 5-5 Configuration of A/D Converter (2/2)



A/D channel 2 setting

☐ Enable A/D channel 2 (ADS2)

Trigger source: Software trigger

Input source: ANI3

Conversion priority: Low

A/D channel 3 setting

☐ Enable A/D channel 3 (ADS3)

Trigger source: Event input from ELC

Input source: ANI3

Conversion priority: Low

Conversion time setting

Please set fCLK not greater than 48MHz

Conversion time mode: Normal 1

Sampling clock cycles: 20 fAD

Conversion time: 43/fCLK (0.8958 μs)

Change to "20fAD"

Change to "43/fCLK"

Conversion result upper/lower bound value setting

● Generates an interrupt request (INTAD0 to INTAD3) when ADLL ≤ ADCRn ≤ ADUL

○ Generates an interrupt request (INTAD0 to INTAD3) when ADUL < ADCRn or ADLL > ADCRn

Upper bound (ADUL) value: 255

Lower bound (ADLL) value: 0

Interrupt setting

Check

☑ Use A/D channel 0 interrupt (INTAD0) Priority: Level 3 (low)

☑ Enable storage of the conversion state information for the analog input channel specified by ADS0 in response to failure

☐ Use A/D channel 1 interrupt (INTAD1) Priority: Level 3 (low)

☐ Enable storage of the conversion state information for the analog input channel specified by ADS1 in response to failure

☐ Use A/D channel 2 interrupt (INTAD2) Priority: Level 3 (low)

☐ Enable storage of the conversion state information for the analog input channel specified by ADS2 in response to failure

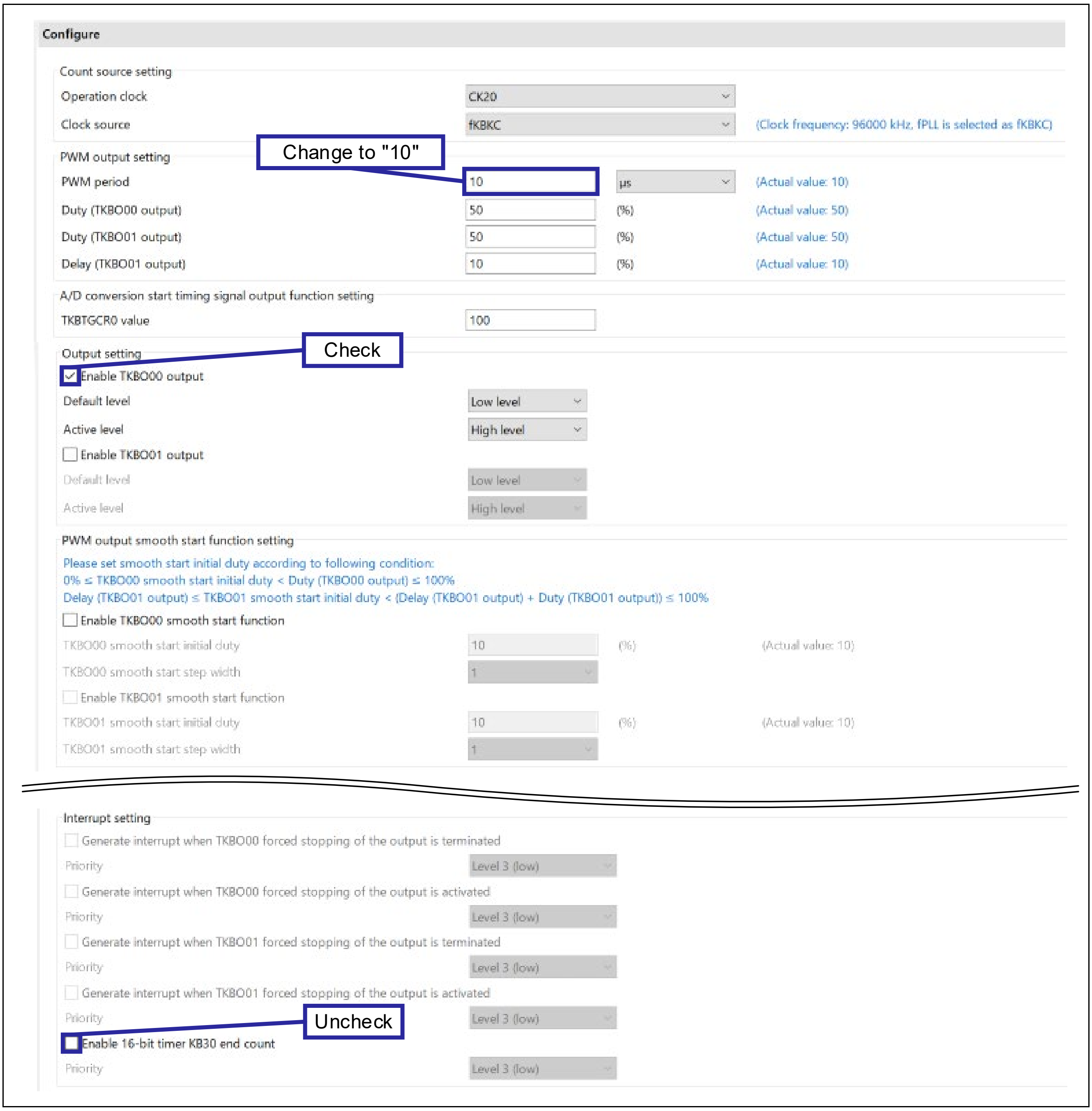
☐ Use A/D channel 3 interrupt (INTAD3) Priority: Level 3 (low)

☐ Enable storage of the conversion state information for the analog input channel specified by ADS3 in response to failure

Table 5-3 Component Configurations (16-bit Timer KB30)

Item	Content
Component	PWM Output
Configuration Name	Config_TKB0
Resource	TKB0
Operation	Standalone operation mode (period control via TKBCR00)

Figure 5-6 Configuration of16-bit Timer KB30



Configure

Count source setting

Operation clock: CK20

Clock source: fKBKC (Clock frequency: 96000 kHz, fPLL is selected as fKBKC)

PWM output setting

PWM period: 10 μs (Actual value: 10) Change to "10"

Duty (TKBO00 output): 50 (%) (Actual value: 50)

Duty (TKBO01 output): 50 (%) (Actual value: 50)

Delay (TKBO01 output): 10 (%) (Actual value: 10)

A/D conversion start timing signal output function setting

TKBTGCR0 value: 100

Output setting

☑ Enable TKBO00 output Check

Default level: Low level

Active level: High level

□ Enable TKBO01 output

Default level: Low level

Active level: High level

PWM output smooth start function setting

Please set smooth start initial duty according to following condition:
0% ≤ TKBO00 smooth start initial duty < Duty (TKBO00 output) ≤ 100%
Delay (TKBO01 output) ≤ TKBO01 smooth start initial duty < (Delay (TKBO01 output) + Duty (TKBO01 output)) ≤ 100%

□ Enable TKBO00 smooth start function

TKBO00 smooth start initial duty: 10 (%) (Actual value: 10)

TKBO00 smooth start step width: 1

□ Enable TKBO01 smooth start function

TKBO01 smooth start initial duty: 10 (%) (Actual value: 10)

TKBO01 smooth start step width: 1

Interrupt setting

□ Generate interrupt when TKBO00 forced stopping of the output is terminated

Priority: Level 3 (low)

□ Generate interrupt when TKBO00 forced stopping of the output is activated

Priority: Level 3 (low)

□ Generate interrupt when TKBO01 forced stopping of the output is terminated

Priority: Level 3 (low)

□ Generate interrupt when TKBO01 forced stopping of the output is activated

Priority: Level 3 (low)

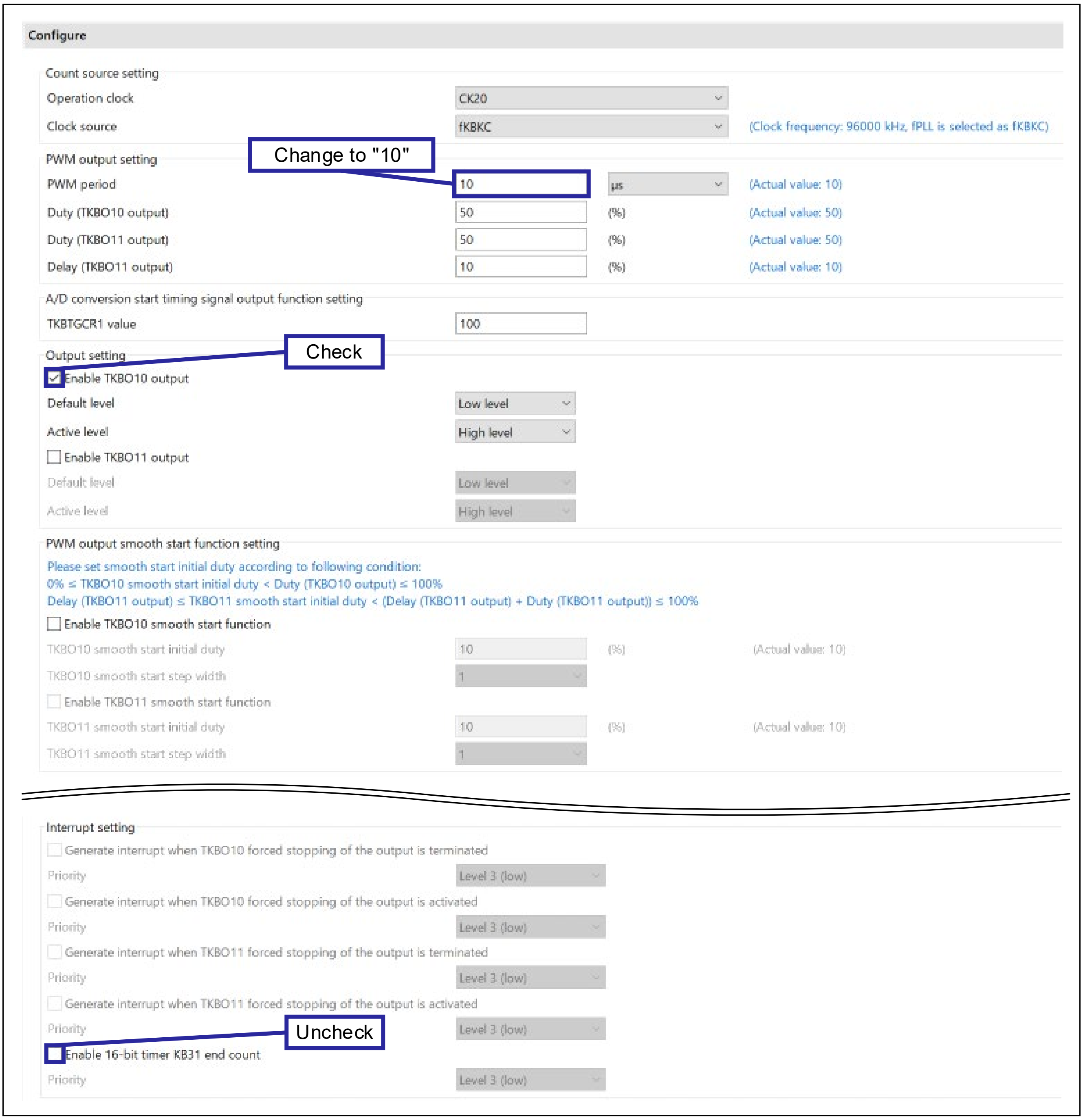
□ Enable 16-bit timer KB30 end count Uncheck

Priority: Level 3 (low)

Table 5-4 Component Configurations (16-bit Timer KB31)

Item	Content
Component	PWM Output
Configuration Name	Config_TKB1
Resource	TKB1
Operation	Standalone operation mode (period control via TKBCR00)

Figure 5-7 Configuration of 16-bit Timer KB31



Configure

Count source setting

Operation clock: CK20

Clock source: fKBKC (Clock frequency: 96000 kHz, fPLL is selected as fKBKC)

PWM output setting

PWM period: 10 μs (Actual value: 10)

Duty (TKBO10 output): 50 (%) (Actual value: 50)

Duty (TKBO11 output): 50 (%) (Actual value: 50)

Delay (TKBO11 output): 10 (%) (Actual value: 10)

A/D conversion start timing signal output function setting

TKBTGCR1 value: 100

Output setting

☑ Enable TKBO10 output

Default level: Low level

Active level: High level

□ Enable TKBO11 output

Default level: Low level

Active level: High level

PWM output smooth start function setting

Please set smooth start initial duty according to following condition:
0% ≤ TKBO10 smooth start initial duty < Duty (TKBO10 output) ≤ 100%
Delay (TKBO11 output) ≤ TKBO11 smooth start initial duty < (Delay (TKBO11 output) + Duty (TKBO11 output)) ≤ 100%

□ Enable TKBO10 smooth start function

TKBO10 smooth start initial duty: 10 (%) (Actual value: 10)

TKBO10 smooth start step width: 1

□ Enable TKBO11 smooth start function

TKBO11 smooth start initial duty: 10 (%) (Actual value: 10)

TKBO11 smooth start step width: 1

Interrupt setting

□ Generate interrupt when TKBO10 forced stopping of the output is terminated

Priority: Level 3 (low)

□ Generate interrupt when TKBO10 forced stopping of the output is activated

Priority: Level 3 (low)

□ Generate interrupt when TKBO11 forced stopping of the output is terminated

Priority: Level 3 (low)

□ Generate interrupt when TKBO11 forced stopping of the output is activated

Priority: Level 3 (low)

□ Enable 16-bit timer KB31 end count

Priority: Level 3 (low)

Change to "10"

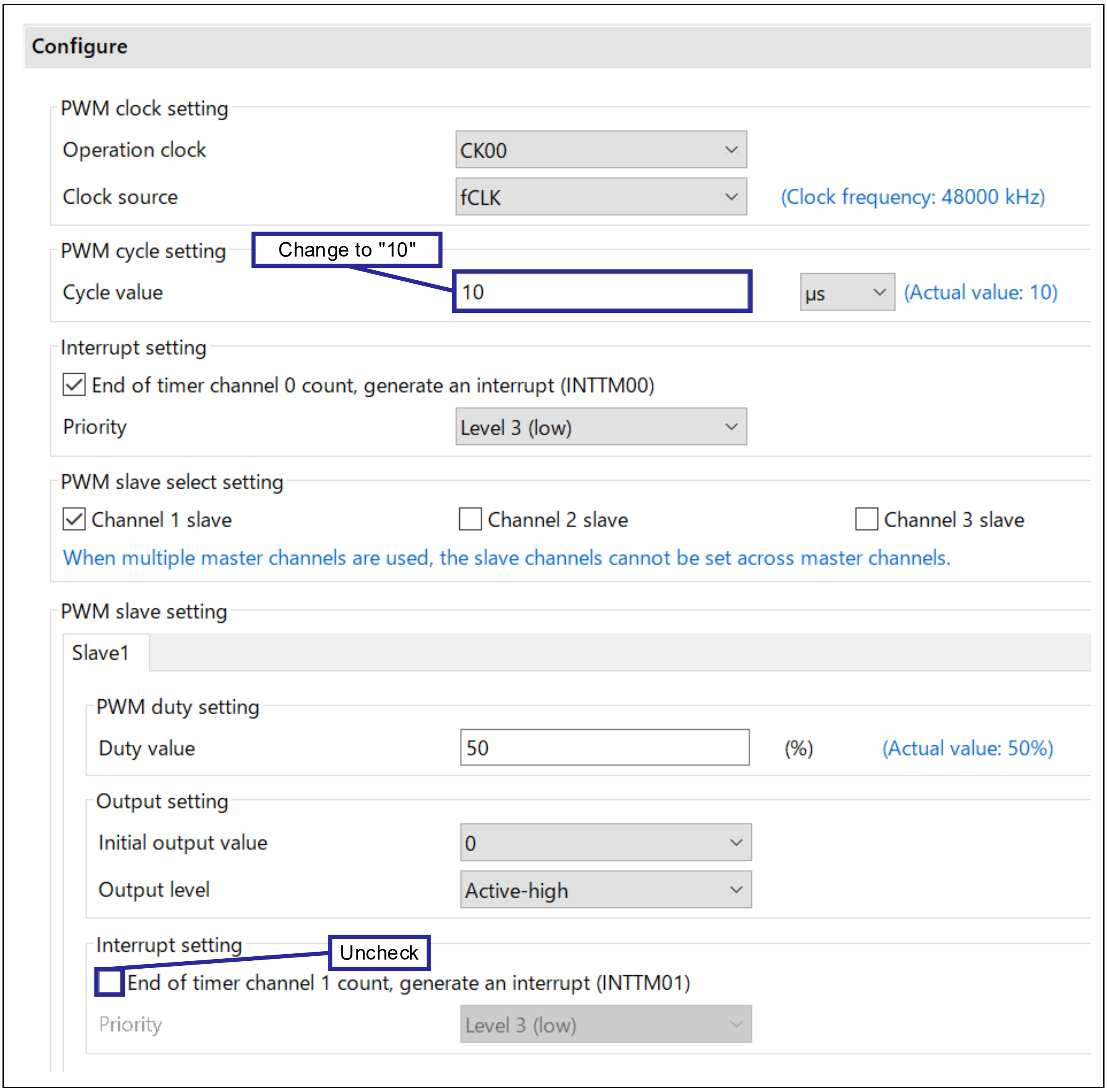
Check

Uncheck

Table 5-5 Component Configurations (Timer Array Unit)

Item	Content
Component	PWM Output
Configuration Name	Config_TAU0_0
Resource	TAU0_0
Operation	PWM function

Figure 5-8 Configuration of Timer Array Unit



Configure

PWM clock setting

Operation clock: CK00

Clock source: fCLK (Clock frequency: 48000 kHz)

PWM cycle setting

Change to "10"

Cycle value: 10 μs (Actual value: 10)

Interrupt setting

☑ End of timer channel 0 count, generate an interrupt (INTTM00)

Priority: Level 3 (low)

PWM slave select setting

☑ Channel 1 slave ☐ Channel 2 slave ☐ Channel 3 slave

When multiple master channels are used, the slave channels cannot be set across master channels.

PWM slave setting

Slave1

PWM duty setting

Duty value: 50 (%) (Actual value: 50%)

Output setting

Initial output value: 0

Output level: Active-high

Interrupt setting

Uncheck

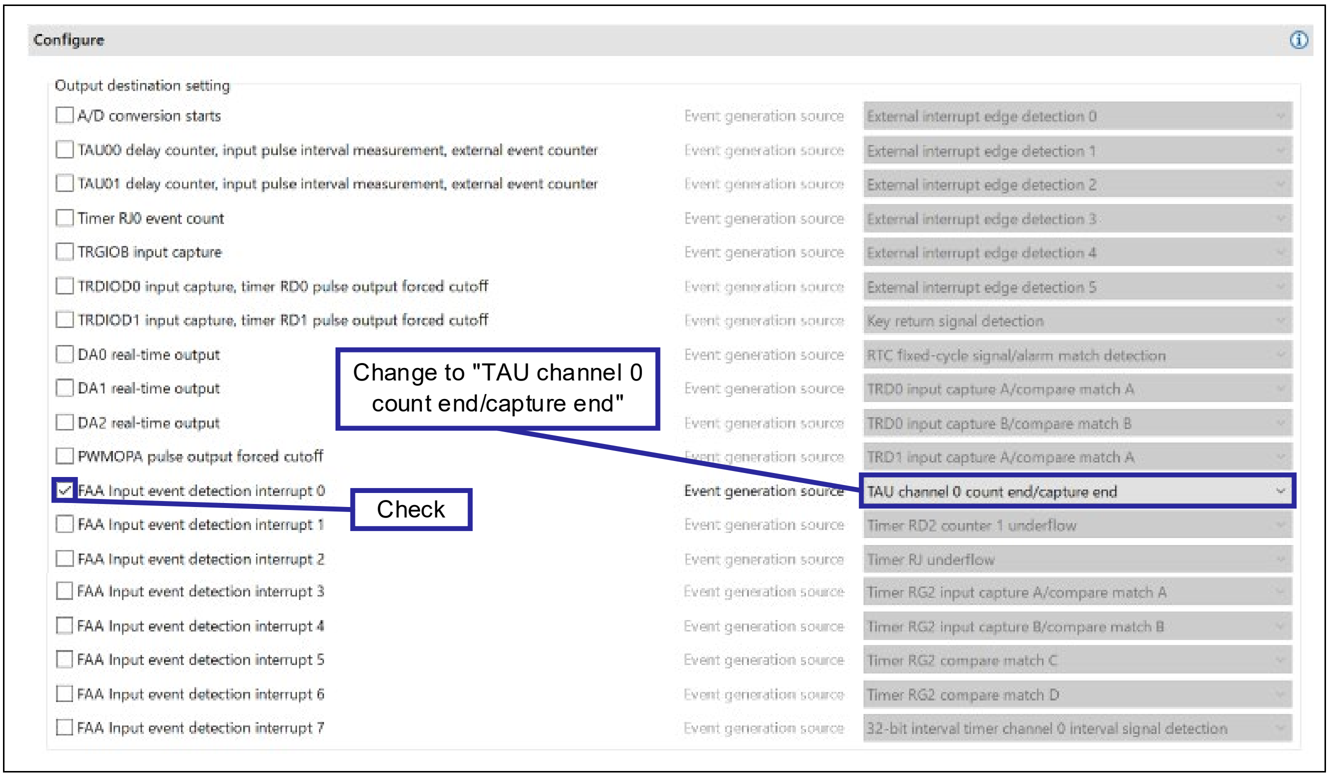
☐ End of timer channel 1 count, generate an interrupt (INTTM01)

Priority: Level 3 (low)

Table 5-6 Component Configurations (Event Link Controller)

Item	Content
Component	Event Link Controller
Configuration Name	Config_ELC
Resource	ELC

Figure 5-9 Configuration of Event Link Controller



Configure

Output destination setting

☐ A/D conversion starts	Event generation source	External interrupt edge detection 0
☐ TAU00 delay counter, input pulse interval measurement, external event counter	Event generation source	External interrupt edge detection 1
☐ TAU01 delay counter, input pulse interval measurement, external event counter	Event generation source	External interrupt edge detection 2
☐ Timer RJ0 event count	Event generation source	External interrupt edge detection 3
☐ TRGIOB input capture	Event generation source	External interrupt edge detection 4
☐ TRDIOD0 input capture, timer RD0 pulse output forced cutoff	Event generation source	External interrupt edge detection 5
☐ TRDIOD1 input capture, timer RD1 pulse output forced cutoff	Event generation source	Key return signal detection
☐ DA0 real-time output	Event generation source	RTC fixed-cycle signal/alarm match detection
☐ DA1 real-time output	Event generation source	TRD0 input capture A/compare match A
☐ DA2 real-time output	Event generation source	TRD0 input capture B/compare match B
☐ PWMOPA pulse output forced cutoff	Event generation source	TRD1 input capture A/compare match A
☑ FAA Input event detection interrupt 0	Event generation source	TAU channel 0 count end/capture end
☐ FAA Input event detection interrupt 1	Event generation source	Timer RD2 counter 1 underflow
☐ FAA Input event detection interrupt 2	Event generation source	Timer RJ underflow
☐ FAA Input event detection interrupt 3	Event generation source	Timer RG2 input capture A/compare match A
☐ FAA Input event detection interrupt 4	Event generation source	Timer RG2 input capture B/compare match B
☐ FAA Input event detection interrupt 5	Event generation source	Timer RG2 compare match C
☐ FAA Input event detection interrupt 6	Event generation source	Timer RG2 compare match D
☐ FAA Input event detection interrupt 7	Event generation source	32-bit interval timer channel 0 interval signal detection

Change to "TAU channel 0 count end/capture end"

Check

5.2 Folder Structure

Table 5-7、Table 5-8 shows the structure of the source files/header files used in the sample code. Note that files automatically generated by the integrated development environment and files from the BSP environment are excluded.

Table 5-7 Folder Structure (1/2)

Folder/File Name	Description	Generated by Smart Configurator
\r01an7250_faa_ad_cotrol<DIR>NOTE 3	Sample code folder	
\src<DIR>	Program storage folder	
main.c	Sample code source file	
r_faa_ad_control.c	FAA source file	
r_faa_ad_control.h	FAA header file	
\smc_gen<DIR>	Smart configurator generated folder	√
\Config_ADC<DIR>	ADC program storage folder	√
Config_ADC.c	ADC source file	√
Config_ADC.h	ADC header file	√
Config_ADC_user.c	ADC interrupt source file	√NOTE 1
\Config_ELC<DIR>	ELC program storage folder	√
Config_ELC.c	ELC source file	√
Config_ELC.h	ELC header file	√
Config_ELC_user.c	ELC interrupt source file	√NOTE 1
\Config_FAA<DIR>	FAA program storage folder	√
Config_FAA_common.c	Common FAA module source file	√
Config_FAA_common.h	Common FAA module header file	√
Config_FAA_common.inc	FAA include file	√
Config_FAA_src.dsp	FAA assembler source file	√NOTE 2
\Config_TAU0_0<DIR>	TAU0_0 program storage folder	√
Config_TAU0_0.c	TAU0_0 source file	√
Config_TAU0_0.h	TAU0_0 header file	√
Config_TAU0_0_user.c	TAU0_0 interrupt source file	√NOTE 1
\Config_TKB0<DIR>	TKB0 program storage folder	√
Config_TKB0.c	TKB0 source file	√
Config_TKB0.h	TKB0 header file	√
Config_TKB0_user.c	TKB0 interrupt source file	√NOTE 1

Table 5-8 Folder Structure (2/2)

Folder/File Name	Description	Generated by Smart Configurator
\r01an7250_faa_ad_cotrol<DIR>[NOTE 3]	Sample code folder	
\src<DIR>	Program storage folder	
\Config_TKB1<DIR>	TKB1 program storage folder	√
Config_TKB1.c	TKB1 source file	√
Config_TKB1.h	TKB1 header file	√
Config_TKB1_user.c	TKB1 interrupt source file	√[NOTE 1]
¥general<DIR>	Initialization and common program storage folder	
¥r_bsp<DIR>	BSP program storage folder	
¥r_config<DIR>	Program storage folder	

Note ”<DIR>” indicates a directory.

Note1. This sample code does not use it.

Note2. In this sample code, the Custom Library of the FAA library is selected, so immediately after code generation, only the templete is provided, and no actual code is included. Code has been added for the sample code. Additionally, the FAA program must be written in assembly language. For detailed information on the FAA assembly instructions, please refer to section 4.16 “Explanation on Instructions” in “RL78/G24 User’s Manual: Hardware” (R01UH0961).

Note3. The sample code for IAR contains the r01an7250_faa_ad_cotrol.ipcf file. For details on the .ipcf file, please refer to "RL78 Smart Configurator User’s Guide: IAR” (R20AN0581).

s

5.3 List of Option Byte Settings

Table 5-9 shows the option byte settings.

Table 5-9 Option Byte Settings

Address	Setting Value	Description
000C0H/040C0H	1110 1111B (EFH)	Watchdog Timer stopped operation (Count stops after reset release)
000C1H/040C1H	1111 1011B (FBH)	LVD0 reset mode Detection voltage: Rising 2.97V / Falling 2.91V
000C2H/040C2H	1110 1010B (EAH)	Flash operation mode: High-speed main mode High-speed on-chip oscillator frequency: 8MHz
000C3H/040C3H	1000 0101B (85H)	On-chip debug operation allowed

5.4 List of Constants

Table 5-10, Table 5-11 shows the list of constants used in this sample code.

Table 5-10 Constants Used in Sample Code (CPU Program)

Constant Name	Set Value	Description
FAA_ALREADY_RUNNING	1	The value indicating FAA running
FAA_SUCCESS	0	The value indicating FAA process successful
FAA_ADC_COMPLETED	1	The value indicating A/D conversion completed
FAA_ADC_FAILED	2	The value indicating A/D conversion failed
FAA_ADC_NOTCOMPLETED	0	The value indicating A/D conversion incomplete

Table 5-11 Constants Used in Sample Code (FAA Program)

Constant Name	Set Value	Description
_C_ADControl_ADS	2	Set value of the analog channel used for A/D conversion
_C_ADControl_0	0	Conversion results when A/D conversion failed (for the CPU program)
_C_ADControl_1	1	The value indicating successful A/D conversion
_C_ADControl_2	2	The value indicating whether A/D conversion is requested by the CPU
_C_ADControl_3	3	The value indicating A/D conversion failure
_C_ADControl_4	4	Unused
_C_ADControl_8	8	Unused
_C_ADControl_FAAAP_ADS0	#ADS0_PTR (015H)	Set value for accessing the ADS0 register by using the FAA address pointer
_C_ADControl_FAAAP_ADINST	#ADINTST_PTR (029H)	Set value for accessing the ADINTST register by using the FAA address pointer
_C_ADControl_FAAAP_ADM3	#ADM3_PTR (014H)	Set value for accessing the ADM3 register by using the FAA address pointer
_C_ADControl_ADCS	80H	Set value used to place the A/D converter in the trigger standby state (ADM0 set value)
_C_ADControl_ADTRSWT	80H	Set value used to generate software triggers (ADM3 set value)
_C_ADControl_FAAAC_ADINTST_CLEAR	00H	Set value used to clear the status of A/D conversion results (ADINTST set value)
_C_ADControl_TKBTRG_TKBRDT_REQ	01H	Request of TKB30 compare register simultaneous update (TKBRDT0)
_C_ADControl_AD_FAILD_VALUE	FFFFH	Conversion results when A/D conversion failed (for the FAA program)
_C_ADControl_ADINTST_ST0F_ST0S	03H	Value used to determine whether to wait for completion of A/D conversion
_C_ADControl_ADINTST_ST0S	02H	Value used to determine whether A/D conversion is successful

5.5 Variable List

Table 5-12 lists the variables used in this sample code.

Table 5-12 Variables Used in Sample Code (CPU Program)

Variable Name	Type	Description	Function Used
result_buffer	uint16_t	Stores A/D conversion results.	main
ad_status	uint16_t	Stores the status of A/D conversion results	main
adc_status	uint16_t	Stores the status of A/D conversion results from SHDMEM	r_faa_adcontrol_getad
adc_result	e_faa_result_adc_t	Stores a return value indicating the status of A/D conversion	r_faa_adcontrol_getad
buffer	uint16_t	Stores A/D conversion results from SHDMEM	r_faa_adcontrol_getad

5.6 Function List

Table 5-13 and Table 5-14 show the functions and processing used in the sample code.

Table 5-13 Function List (CPU Program)

Function Name	Overview	Source File
main	Main processing	main.c
r_faa_adcontrol_start	FAA start processing	r_faa_ad_control.c
r_faa_adcontrol_stop	FAA stop processing	r_faa_ad_control.c
r_faa_adcontrol_requestadc	A/D conversion request processing of the specified analog channel	r_faa_ad_control.c
r_faa_adcontrol_getad	A/D conversion result acquisition processing	r_faa_ad_control.c

Table 5-14 Processing List (FAA Program)

Label Name	Overview	Source File
_P_ADControl	Enabling interrupts and waiting for an interrupt.	Config_FAA_src.dsp
_P_ADControl_TAU0_Interrupt	Changing the duty ratio of the TKBO00 pin, and storing the duty ratio for the TKBO10 pin in SHDMEM	Config_FAA_src.dsp
_P_ADControl_GetAd	Performing A/D conversion	Config_FAA_src.dsp

5.7 Function Specifications

The following describes the function specifications of the sample code.

CPU program

[Function name] main

Overview	Main processing
Headers	r_smc_entry.h, platform.h, r_faa_ad_control.h, Config_TKB0.h, Config_TKB1.h, Config_ADC.h
Declaration	void main (void);
Description	This function specifies the initial settings of the A/D converter, TKB30, TKB31, and timer array unit (TAU0), and starts the FAA operation at fixed intervals (10 μs) by using a TAU channel 0 count end interrupt (INTTM00). It also requests the FAA to perform A/D conversion and, after the conversion is complete, acquires the conversion result and changes the duty ratio of TKB31.
Arguments	None
Return values	None
Remarks	None

[Function name] r_faa_adcontrol_start

Overview	FAA start processing	
Headers	Config_FAA_common.h, r_faa_ad_control.h	
Declaration	FAA_Status_t r_faa_adcontrol_start(void);	
Description	This function sets the FAA stack pointer and the start address of the FAA program, and then starts FAA operation.	
Arguments	None	
Return values	FAA_SUCCESS	Successful FAA processing
	FAA_ALREADY_RUNNING	FAA running
Remarks	None	

[Function name] r_faa_adcontrol_stop

Overview	FAA stop processing
Headers	Config_FAA_common.h, r_faa_ad_control.h
Declaration	void r_faa_adcontrol_stop(void);
Description	This function stops FAA operation.
Arguments	None
Return values	None
Remarks	None

[Function name] r_faa_adcontrol_requestadc

Overview	A/D conversion request processing of the specified analog channel
Headers	Config_FAA_common.h, r_faa_ad_control.h
Declaration	void r_faa_adcontrol_requestadc (e_ad_channel_t channel);
Description	This function requests the FAA program to perform A/D conversion for the specified analog channel.
Arguments	channel: Analog channel used for A/D conversion
Return values	None
Remarks	None

[Function name] r_faa_adcontrol_getad

Overview	A/D conversion result acquisition processing	
Headers	Config_FAA_common.h, r_faa_ad_control.h	
Declaration	e_faa_result_adc_t r_faa_adcontrol_getad (uint16_t * const buffer);	
Description	This function acquires the result of conversion requested by r_faa_adcontrol_requestadc().	
Arguments	buffer: A/D conversion result	
Return values	FAA_ADC_COMPLETED	A/D conversion completed
	FAA_ADC_FAILED	A/D conversion failed
	FAA_ADC_NOTCOMPLETED	A/D conversion incomplete
Remarks	None	

FAA program

[Label name] _P_ADControl

Overview	Enabling interrupts and waiting for an interrupt.
Headers	Config_FAA_common.inc
Declaration	-
Description	This processing enables interrupts and waits for generation of a TAU channel 0 count end interrupt (INTTM00).
Arguments	None
Return values	None
Remarks	None

[Label name] _P_ADControl_TAU0_Interrupt

Overview	Changing the duty ratio of the TKBO00 pin, and storing the duty ratio for the TKBO10 pin in SHDMEM
Headers	Config_FAA_common.inc
Declaration	-
Description	This processing changes the PWM output duty ratio of the TKBO00 pin based on the A/D conversion results. If A/D conversion is requested by the CPU, this processing also performs A/D conversion of the specified analog channel, and then stores the conversion result in SHDMEM.
Arguments	None
Return values	None
Remarks	None

[Label name] _P_ADControl_GetAd

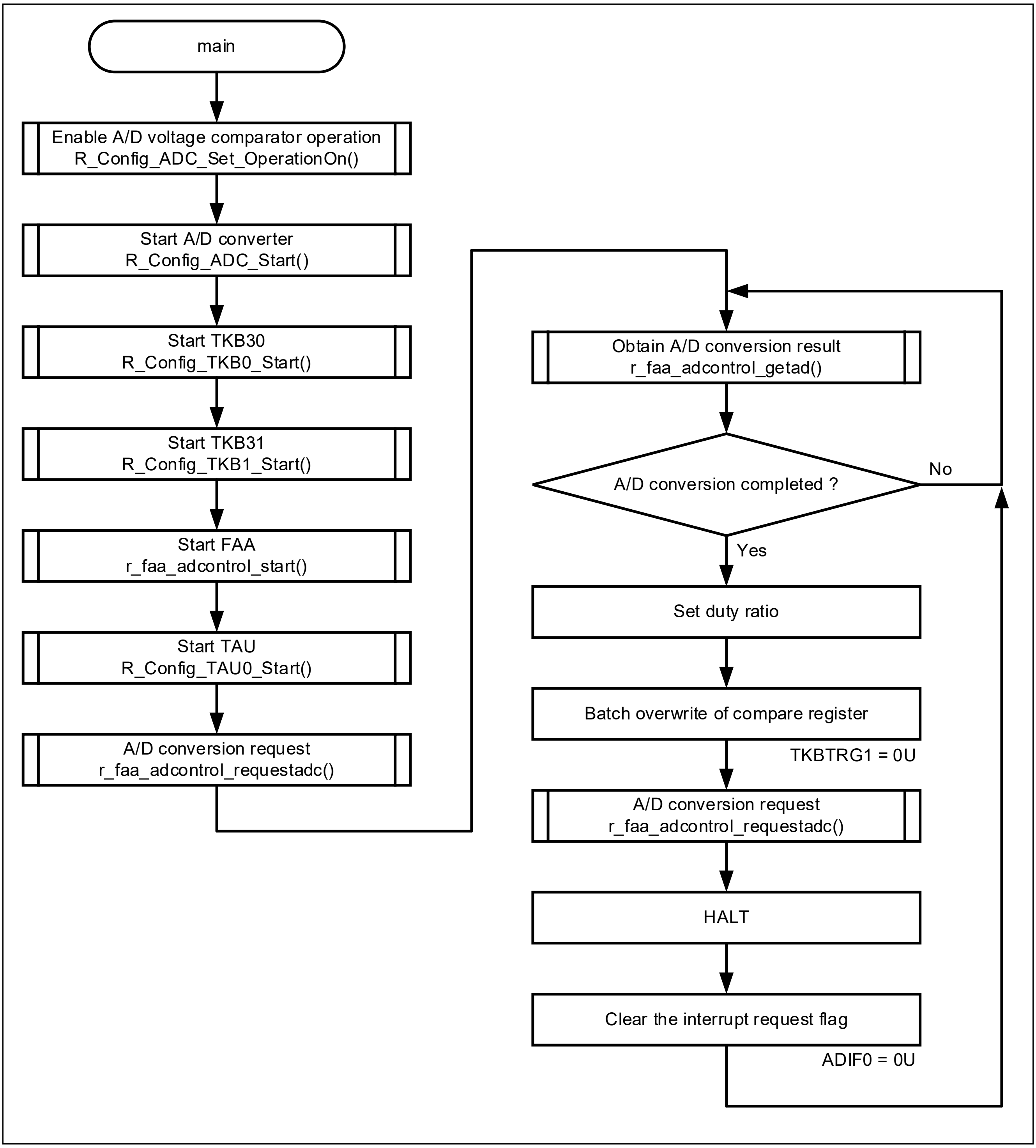
Overview	Performing A/D conversion
Headers	Config_FAA_common.inc
Declaration	-
Description	This processing performs A/D conversion and then clears the status of conversion results.
Arguments	None
Return values	None
Remarks	None

5.8 Flowchart

5.8.1 Main process

Figure 5-10 shows the flowchart for the main process.

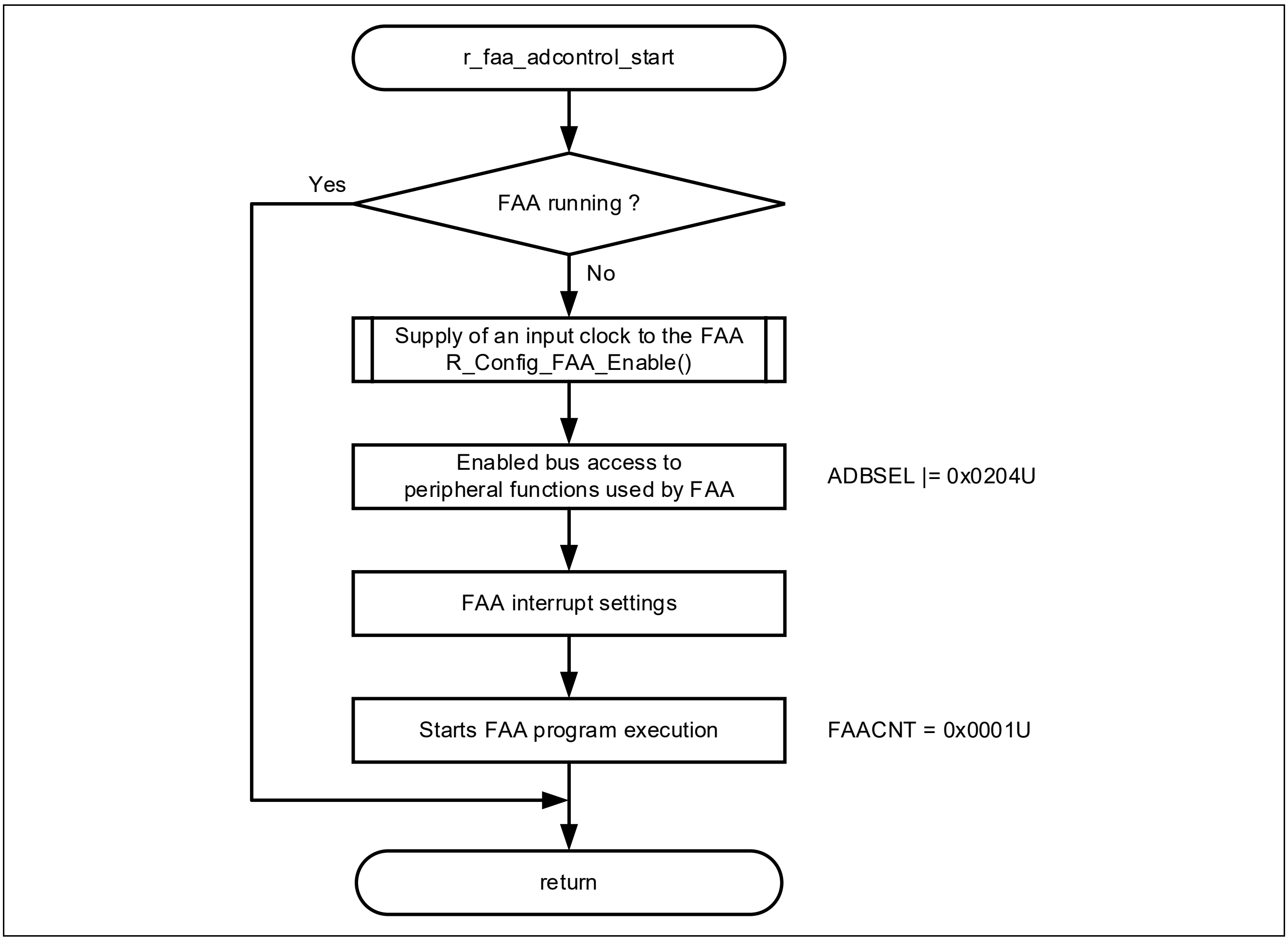
Figure 5-10 Main Process



5.8.2 r_faa_adcontrol_start function

Figure 5-11 shows the flowchart of r_faa_adcontrol_start function.

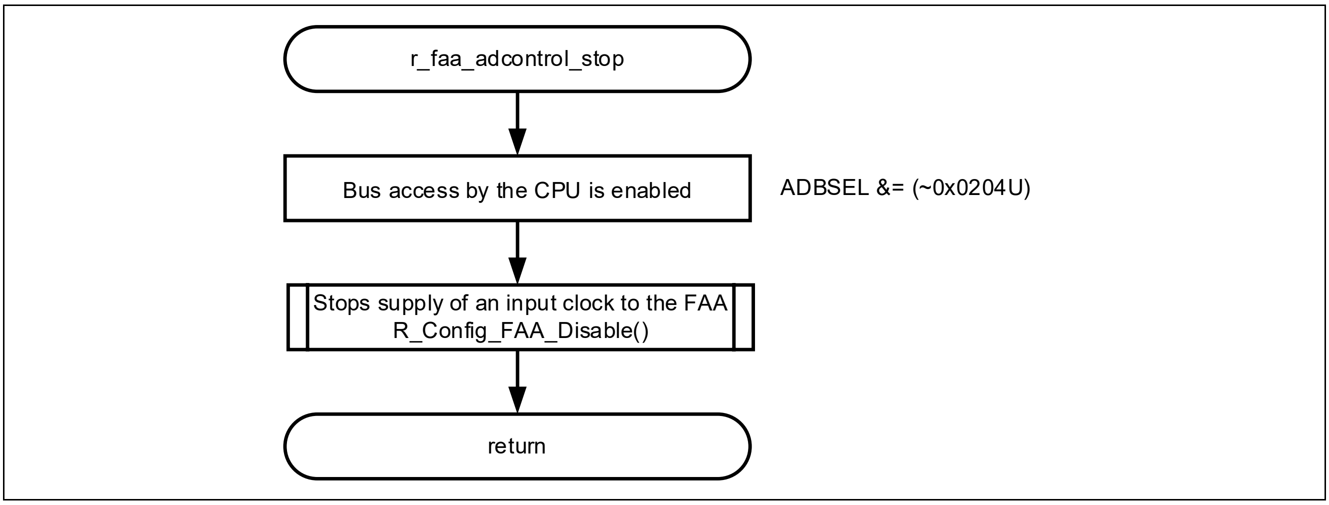
Figure 5-11 r_faa_adcontrol_start function



5.8.3 r_faa_adcontrol_stop function

Figure 5-12 shows the flowchart of r_faa_adcontrol_stop function.

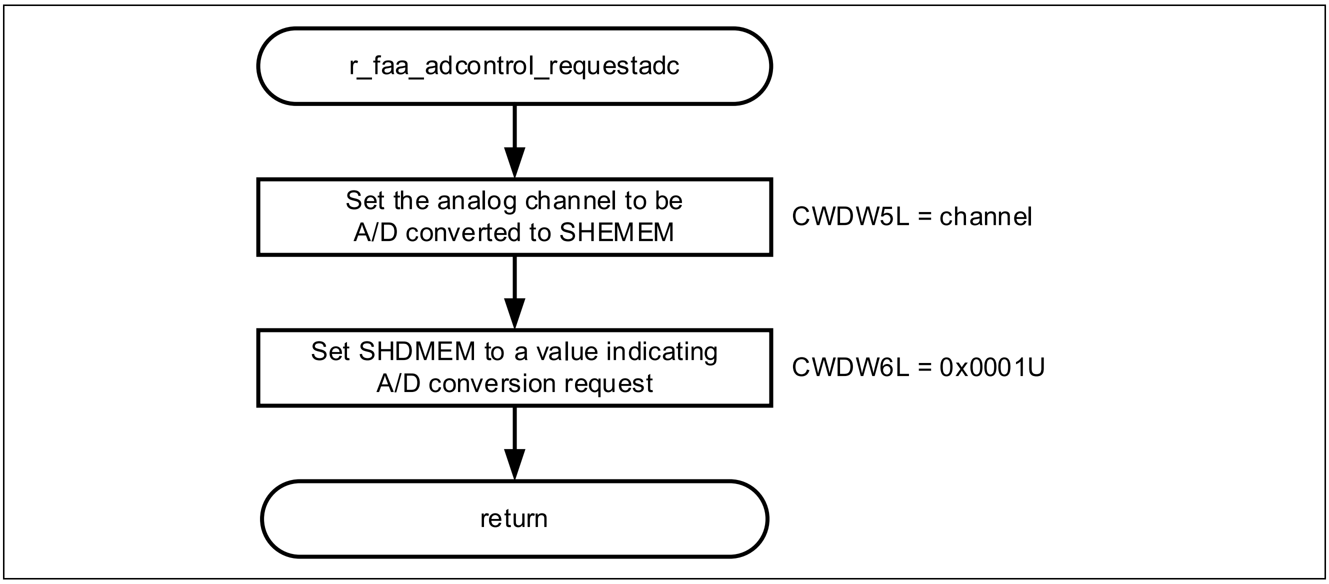
Figure 5-12 r_faa_adcontrol_stop function


r_faa_adcontrol_stop
Bus access by the CPU is enabled
ADBSEL &= (~0x0204U)
Stops supply of an input clock to the FAA
R_Config_FAA_Disable()
return

5.8.4 r_faa_adcontrol_requestadc function

Figure 5-13 shows the flowchart of r_faa_adcontrol_requestadc function.

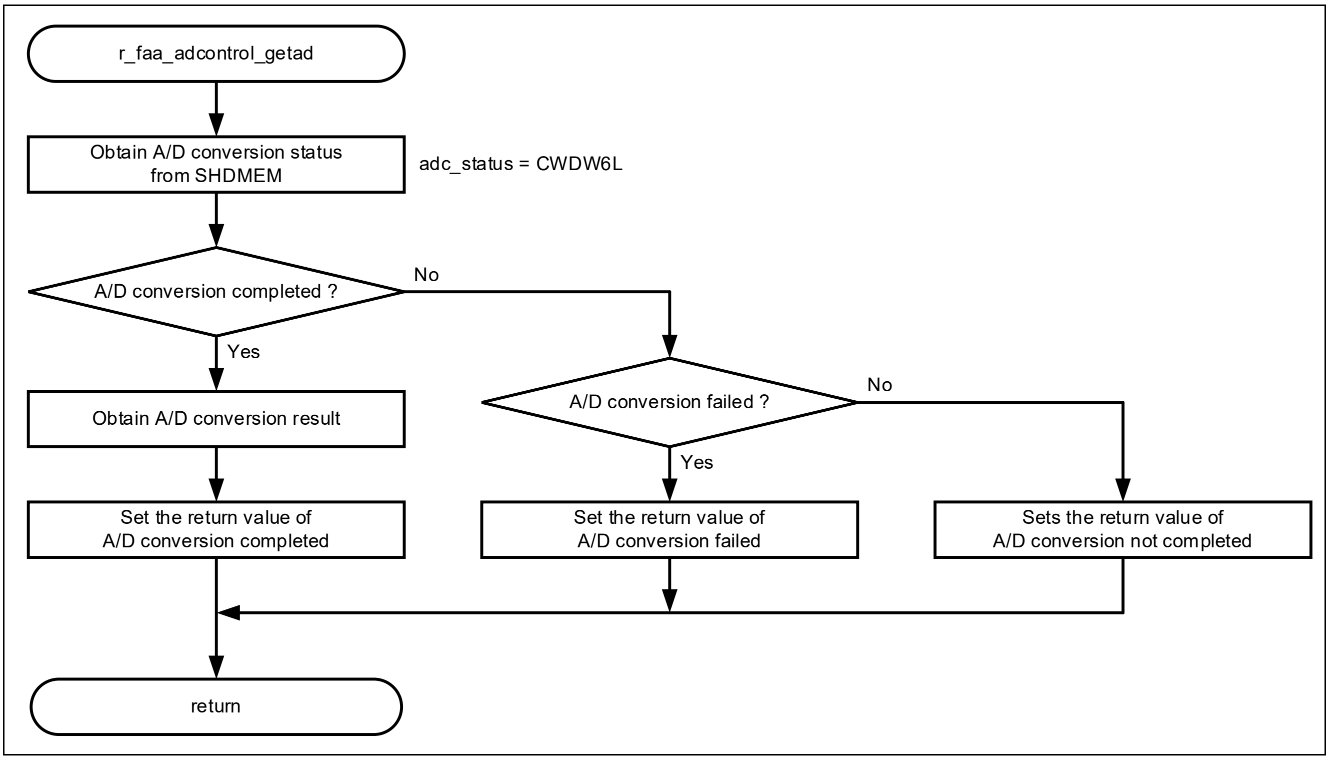
Figure 5-13 r_faa_adcontrol_requestadc function


r_faa_adcontrol_requestadc
Set the analog channel to be A/D converted to SHEMEM
CWDW5L = channel
Set SHDMEM to a value indicating A/D conversion request
CWDW6L = 0x0001U
return

5.8.5 r_faa_adcontrol_getad function

Figure 5-14 shows the flowchart of r_faa_adcontrol_getad function.

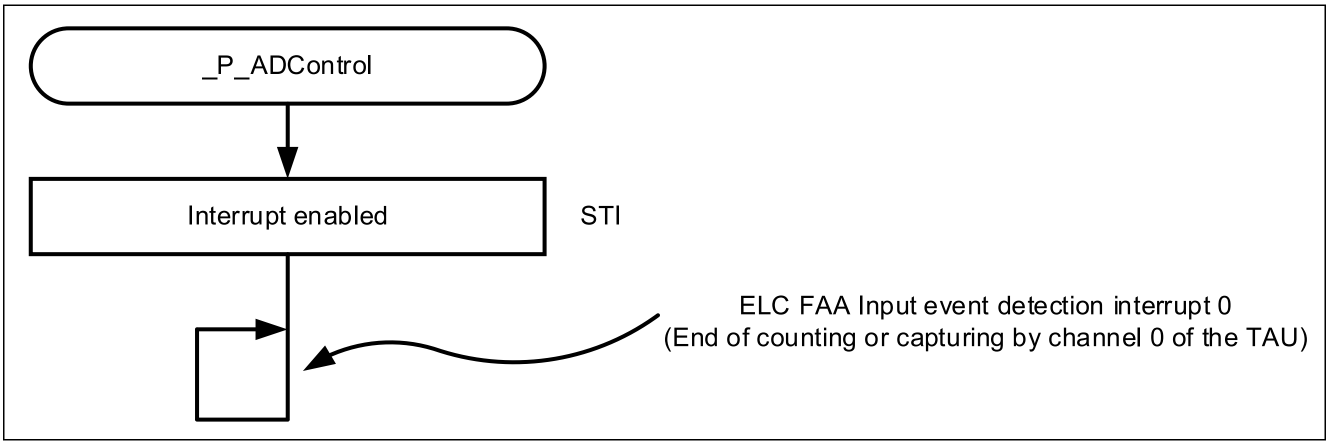
Figure 5-14 r_faa_adcontrol_getad function


r_faa_adcontrol_getad
Obtain A/D conversion status from SHDMEM
adc_status = CWDW6L
A/D conversion completed ?
No
Yes
Obtain A/D conversion result
A/D conversion failed ?
No
Yes
Set the return value of A/D conversion completed
Set the return value of A/D conversion failed
Sets the return value of A/D conversion not completed
return

5.8.6 FAA Process: _P_ADControl

Figure 5-15 shows the flowchart of _P_ADControl.

Figure 5-15 _P_ADControl


_P_ADControl
Interrupt enabled
STI
ELC FAA Input event detection interrupt 0
(End of counting or capturing by channel 0 of the TAU)

5.8.7 FAA Process: _P_ADControl_TAU0_Interrupt

Figure 5-16, Figure 5-17 shows the flowchart of _P_ADControl_TAU0_Interrupt.

Figure 5-16 _P_ADControl_TAU0_Interrupt (1/2)

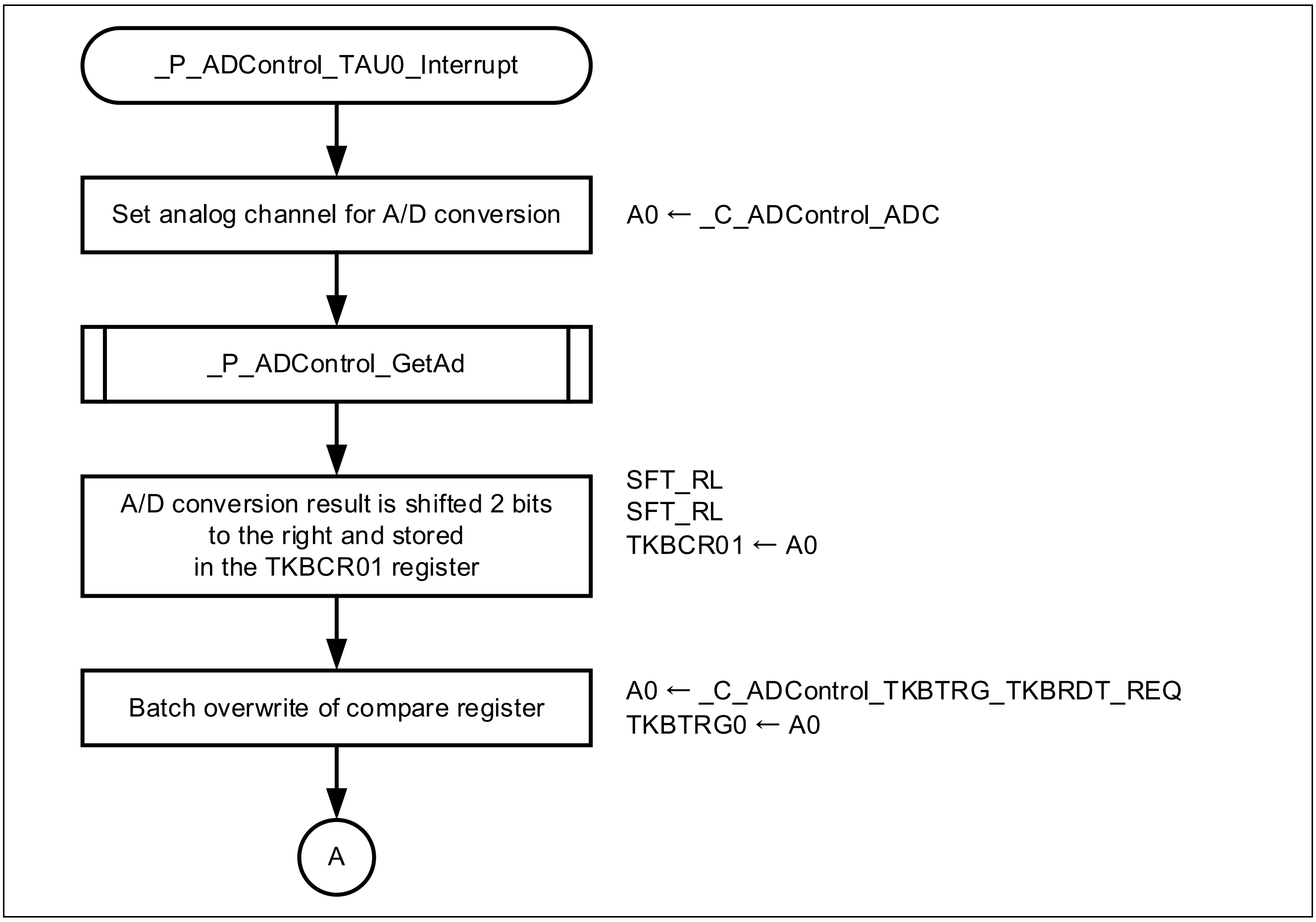
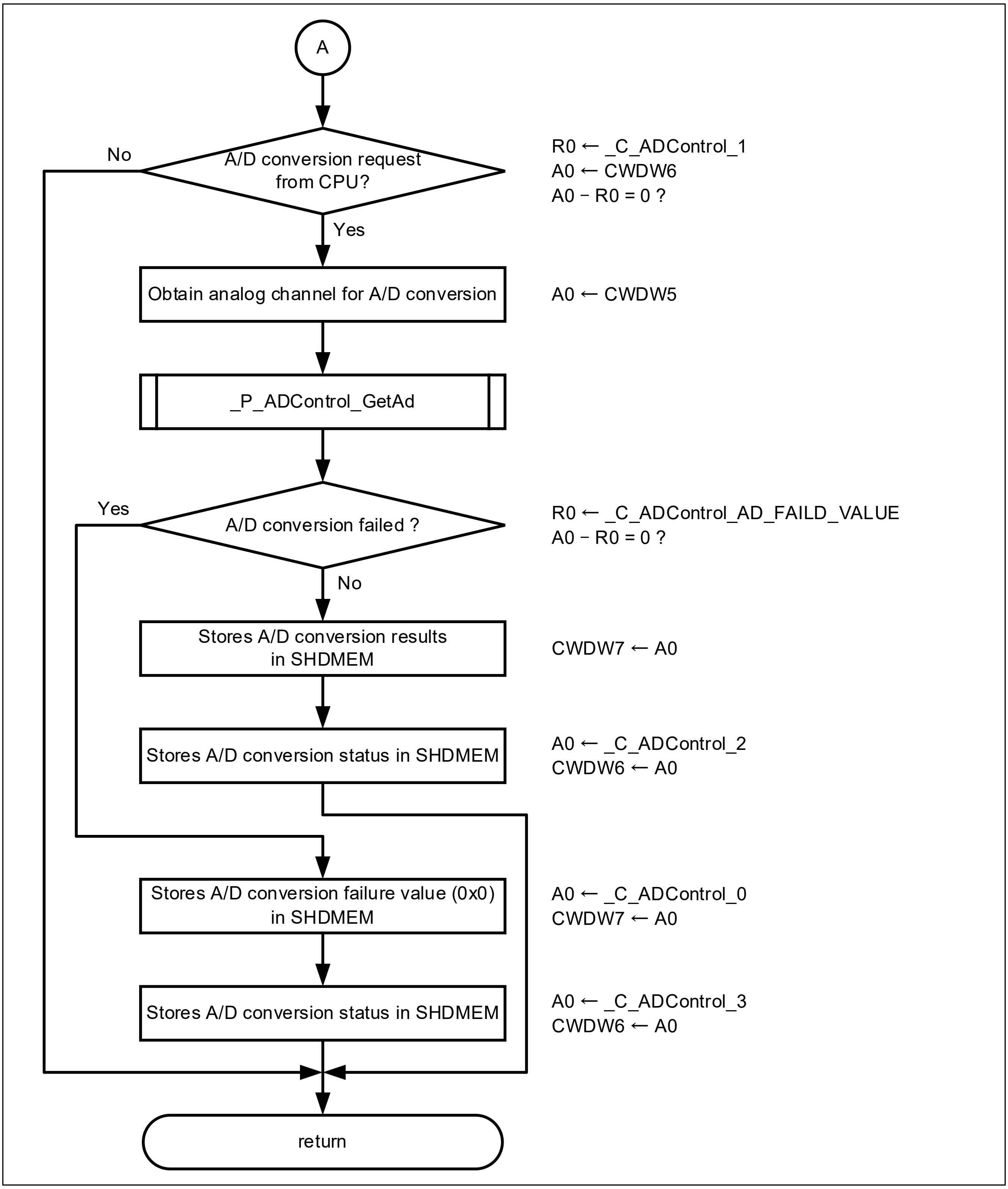

_P_ADControl_TAU0_Interrupt
Set analog channel for A/D conversion
A0 ← _C_ADControl_ADC
_P_ADControl_GetAd
A/D conversion result is shifted 2 bits to the right and stored in the TKBCR01 register
SFT_RL
SFT_RL
TKBCR01 ← A0
Batch overwrite of compare register
A0 ← _C_ADControl_TKBTRG_TKBRDT_REQ
TKBTRG0 ← A0
A

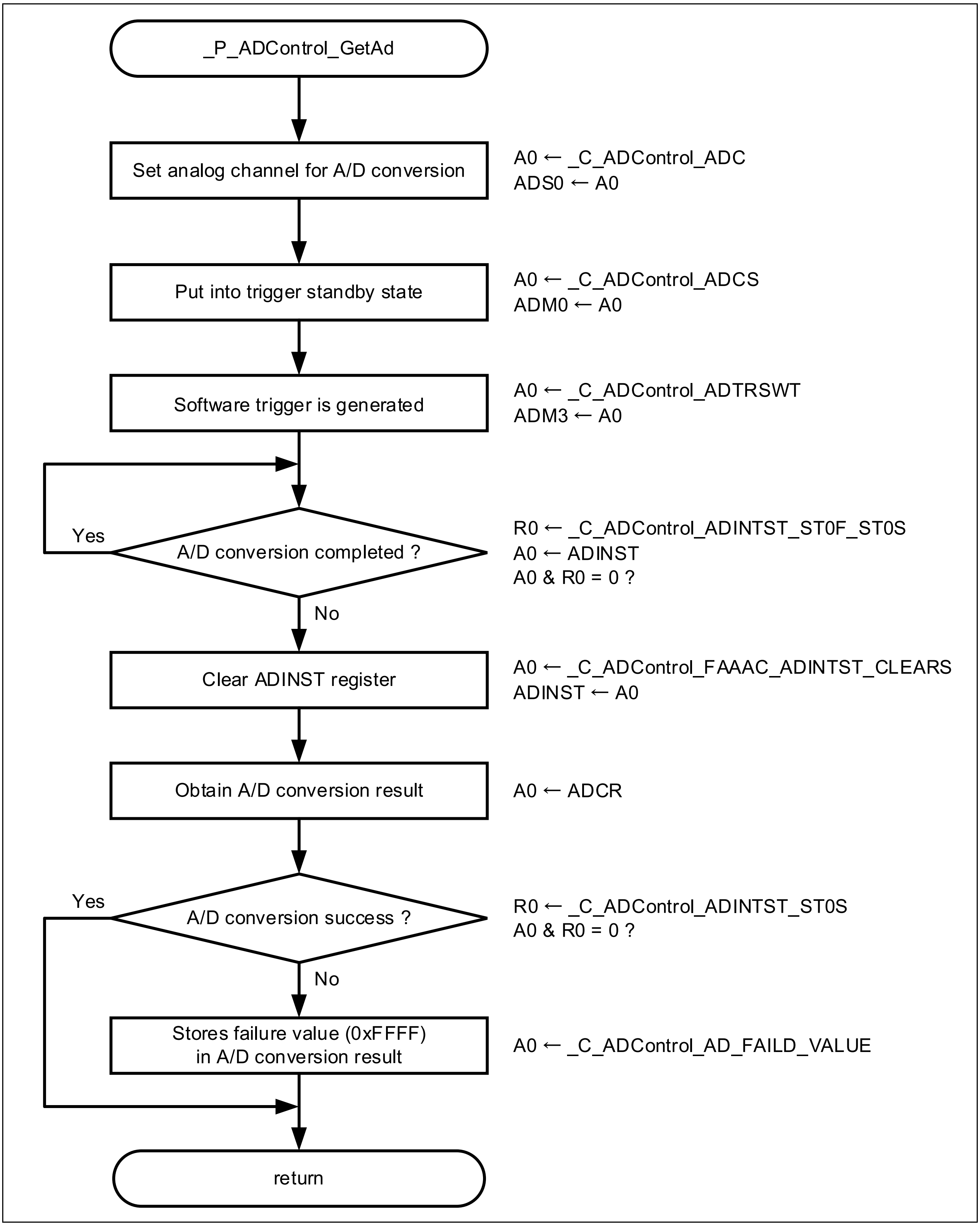
Figure 5-17 _P_ADControl_TAU0_Interrupt (2/2)


A
A/D conversion request from CPU?
No
Yes
R0 ← _C_ADControl_1
A0 ← CWDW6
A0 − R0 = 0 ?
Obtain analog channel for A/D conversion
A0 ← CWDW5
_P_ADControl_GetAd
A/D conversion failed ?
Yes
No
R0 ← _C_ADControl_AD_FAILD_VALUE
A0 − R0 = 0 ?
Stores A/D conversion results in SHDMEM
CWDW7 ← A0
Stores A/D conversion status in SHDMEM
A0 ← _C_ADControl_2
CWDW6 ← A0
Stores A/D conversion failure value (0x0) in SHDMEM
A0 ← _C_ADControl_0
CWDW7 ← A0
Stores A/D conversion status in SHDMEM
A0 ← _C_ADControl_3
CWDW6 ← A0
return

5.8.8 FAA Process: _P_ADControl_GetAd

Figure 5-18 shows the flowchart of _P_ADControl_GetAd.

Figure 5-18 _P_ADControl_GetAd


_P_ADControl_GetAd
Set analog channel for A/D conversion
A0 ← _C_ADControl_ADC
ADS0 ← A0
Put into trigger standby state
A0 ← _C_ADControl_ADCS
ADM0 ← A0
Software trigger is generated
A0 ← _C_ADControl_ADTRSWT
ADM3 ← A0
Yes
A/D conversion completed ?
R0 ← _C_ADControl_ADINTST_ST0F_ST0S
A0 ← ADINST
A0 & R0 = 0 ?
No
Clear ADINST register
A0 ← _C_ADControl_FAAAC_ADINTST_CLEARS
ADINST ← A0
Obtain A/D conversion result
A0 ← ADCR
Yes
A/D conversion success ?
R0 ← _C_ADControl_ADINTST_ST0S
A0 & R0 = 0 ?
No
Stores failure value (0xFFFF)
in A/D conversion result
A0 ← _C_ADControl_AD_FAILD_VALUE
return

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

RL78/G24 User’s Manual: Hardware (R01UH0961)

RL78 family User's Manual: Software (R01US0015)

RL78/G24 Fast Prototyping Board User’s Manual (R20UT5091)

RL78 Smart Configurator User’s Gude: CS+ (R20AN0580)

RL78 Smart Configurator User’s Gude: e2 studio (R20AN0579)

RL78 Smart Configurator User’s Gude: IAR (R20AN0581)

Flexible Application Accelerator (FAA) Tool Guide: CS＋ (R01AN7094)

Flexible Application Accelerator (FAA) Tool Guide: e^2 studio (R01AN7095)

(The latest version can be downloaded from the Renesas Electronics website.)

Technical Update/Technical News

(The latest version can be downloaded from the Renesas Electronics website.)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2024.12.05	—	First Edition

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/.