RL78/G24

A/D Converter (Advanced Mode Enabled)

Introduction

This application note describes the functions and operations of the RL78/G24 A/D converter (advanced mode enabled) by using two usage examples.

For details on the operation with the A/D converter (advanced mode disabled), see the **Application Note for RL78/G24 A/D Converter (Advanced Mode Disabled) (R01AN6992)**.

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. Description of A/D Converter Advanced Mode Functions

Advanced mode is added to the A/D converter mounted on the RL78/G24.

Enable the advanced mode when setting the CPU/peripheral hardware clock frequency (f_{CLK}) to 48 MHz. In advanced mode, conversion of signals on four channels is possible, so conversion of up to four analog input signals is specifiable. Different conversion triggers can be allocated to each channel. Sequential A/D conversion of channels allocated to generated triggers and simultaneous sampling of up to three channels are possible. Moreover, whether to generate an interrupt request signal (INTAD0 to INTAD3) at the end of A/D conversion can be specified for each conversion channel. Operation of advanced mode is fixed to no wait mode and one-shot conversion mode.

Table 1-1 and Table 1-2 show the functional comparison when the advanced mode is enabled or disabled.

Table 1-1 The functional comparison when the advanced mode is enabled or disabled (1/2)

	Advanced Mode Enabled	Advanced Mode Disabled
Maximum Clock Frequency (f_{CLK})	48MHz	32MHz
Resolution	8 bit /10 bit /12 bit	8 bit /10 bit /12 bit
Simultaneous Sampling Channels	1-3 channels	1 channel
Input Channel	ANI0-ANI7, ANI16-ANI30 Temperature sensor output voltage, internal reference voltage PGA output	ANI0-ANI7, ANI16-ANI30 Temperature sensor output voltage, internal reference voltage
Trigger Mode	【Software trigger (no-wait mode)】 A/D conversion is started by setting the ADM3.ADTRSWT to 1.	【Software trigger no-wait mode】 Conversion is started by setting the ADCE bit to 1 by software, and then setting ADCS to 1 after the A/D power supply stabilization wait time has Passed.
		【Software trigger wait mode】 The power is turned on by setting the ADCS bit to 1 by software while A/D conversion is stopped, and conversion is then started automatically after the A/D power supply stabilization wait time has passed.
	【Hardware trigger (no-wait mode)】 A/D conversion is started by detecting a hardware trigger configurated for each conversion channel.	【Hardware trigger no-wait mode】 Conversion is started by detecting a hardware trigger.
		【Hardware trigger wait mode】 The power to the A/D converter is turned on by detecting a hardware trigger while the A/D converter is off and, in the conversion, standby state, and conversion is then started automatically after the stabilization wait time passes. When using the SNOOZE mode, specify the hardware trigger wait mode.

Table 1-2 The functional comparison when the advanced mode is enabled or disabled (2/2)

	Advanced Mode Enabled	Advanced Mode Disabled
Conversion Trigger	• Timer array unit channel 01 count or capture end interrupt signal (INTTM01) • Realtime clock interrupt signal (INTRTC) • 32-bit interval timer channel 0 interrupt signal (ELCITL0) • Event input from ELC • 16-bit timer KB30 A/D trigger signal • 16-bit timer KB31 A/D trigger signal • 16-bit timer KB32 A/D trigger signal • timer RD2 A/D conversion trigger 0 • timer RD2 A/D conversion trigger 1 • Software trigger	• Timer array unit channel 01 count or capture end interrupt signal (INTTM01) • Realtime clock interrupt signal (INTRTC) • 32-bit interval timer channel 0 interrupt signal (ELCITL0) • Event input from ELC • Software trigger
Channel Selection Mode	Up to four analog input channels and their corresponding conversion triggers are assigned. When a trigger occurs, the analog input channels assigned to the trigger are converted to digital sequentially.	【Select mode】 A/D conversion is performed on the analog input of one selected channel.
		【Scan mode】 A/D conversion is performed on the analog input of four channels in order. Four consecutive channels can be selected from ANI0 to ANI7 as analog input channels.
Conversion Operation Mode	The selected analog input channel for each conversion channel is converted to digital once.	【One-shot conversion mode】 A/D conversion is performed on the selected channel once.
		【Sequential conversion mode】 A/D conversion is sequentially performed on the selected channels until it is stopped by software.

2. Usage Examples and Operation

2.1 Usage Example 1: Operation Triggered by Hardware and Software

2.1.1 Overview of specifications

In usage example 1, the advanced mode is used for A/D conversion of analog input voltage of the P22/ANI2 pin, P23/ANI3 pin, P03/ANI16 pin, and PGA output. A/D conversion is performed for each channel. The conversion uses a software trigger for channels 0 and 1, hardware trigger (count end interrupt by interval timer: Once per 100 μs) for channel 3, and hardware trigger (realtime clock interrupt by realtime clock: Once per 0.5 second) for channel 3. After A/D conversion of each channel ends, an interrupt is generated, and the conversion result is stored in the variable of the internal RAM.

Table 2-1 shows an example of channel settings.

Table 2-1 Example of Channel Settings (Usage Example 1)

	Advanced mode	Wait Mode	Conversion Mode
Setting	ON	No-wait mode (fixed)	One-shot conversion mode (fixed)

	Channel Setting Conditions				
	Simultaneous sampling setting ADSn.ADSPSCn[1:0]	Trigger source ADTRn.ADTRSn[3:0]	Priority	Analog input channel ADSn.ADSn[4:0]	A/D conversion result register
Channel 0	unused 00B	software trigger 1111B	low	ANI2 00010B	ADCR0
Channel 1	unused 00B	software trigger 1111B	low	ANI3 00011B	ADCR1
Channel 2	unused 00B	INTTM01 signal 0000B	low	ANI16 10000B	ADCR2
Channel 3	unused 00B	INTRTC signal 0010B	high	PGA output 11111B	ADCR3

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

Table 2-2 Peripheral Functions and Their Usage (Example 1)

Peripheral Function	Usage
A/D converter (Advanced mode ON)	Performs A/D conversion of analog input voltage of the P22/ANI2 pin, P23/ANI3 pin, P03/ANI16 pin, and PGA output
Realtime Clock (RTC)	Uses a realtime clock interrupt signal (INTRTC) as a hardware trigger
Timer array unit (TAU)	Uses a count end interrupt signal (INTTM01) as a hardware trigger
Programmable gain amplifier (PGA)	Used as an analog input voltage with the gain set to x4

Table 2-3 shows the trigger source of each channel and the components to be used.

Table 2-3 Trigger Source of Each Channel and Used Components

Channel	Input Source	Trigger Source	Component Used for Trigger Source
Channel 0	ANI2	Software trigger	-
Channel 1	ANI3	Software trigger	-
Channel 2	ANI16	Hardware trigger (INTTM01 signal)	Interval timer (generating a count end interrupt on channel 01 at 100 μs intervals)
Channel 3	PGAI0	Hardware trigger (INTRTC signal)	Realtime clock (generating a realtime clock interrupt at 0.5 second intervals)

2.1.2 Operation

The following shows the A/D converter settings.

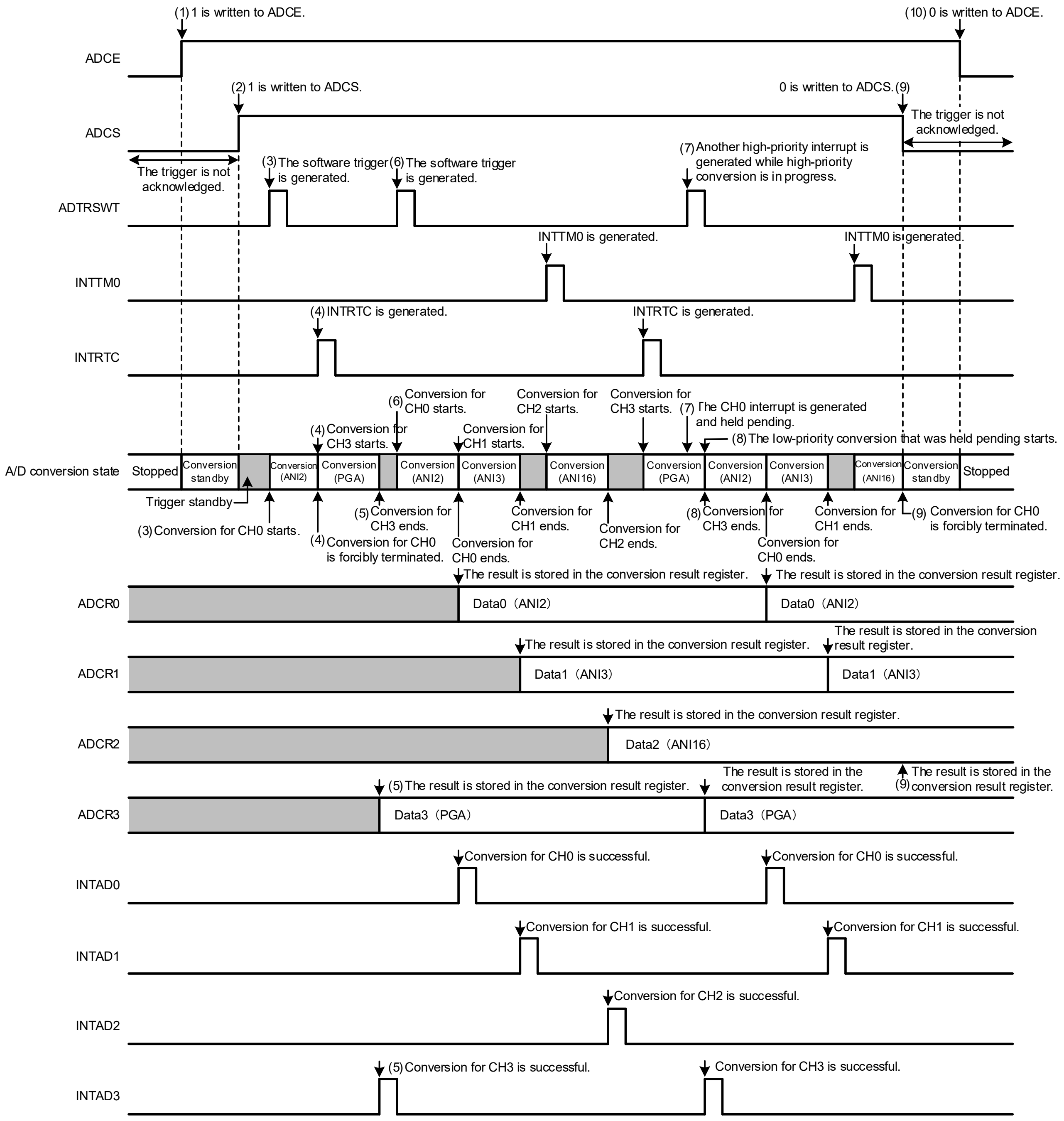
<Settings>

- The advanced mode is used in the A/D converter.
- The resolution is set to 12 bits.
- AV_{REFP} is set as the + side reference voltage of the A/D converter, and AV_{REFM} is set as the - side reference voltage of the A/D converter.
- Simultaneous sampling is not used.
- For A/D channel 0, the trigger source is set to software trigger, the input source is set to ANI2, and the priority is set to Low.
- For A/D channel 1, the trigger source is set to software trigger, the input source is set to ANI3, and the priority is set to Low.
- For A/D channel 2, the trigger source is set to INTTM01 signal (count end interrupt signal at 100 μs intervals), the input source is set to ANI16, and the priority is set to Low.
- For A/D channel 3, the trigger source is set to INTRTC signal (realtime clock interrupt signal at 0.5 second intervals), the input source is set to PGA output (gain is set to x4), and the priority is set to High.
- The operating voltage mode is set to normal mode 1, and the conversion time is set to $55/f_{CLK}$.
- The upper limit (ADUL) of A/D conversion results is set to 255, and the lower limit (ADLL) is set to 0. [Note:]
- A/D conversion end interrupts (INTAD0 to INTAD3) are enabled.

Note: When the 12-bit resolution is selected, the set values of the upper and lower limits of A/D conversion results are checked by comparing the ADCRn[11:4] bits with the values of the ADUL and ADLL registers. For details, see **20.3.16 Conversion result comparison upper limit setting register (ADUL)** and **20.3.17 Conversion result comparison lower limit setting register (ADLL)** in the **RL78/G24 User's Manual: Hardware**.

Figure 2-1 shows the operation of A/D converter.

Figure 2-1 Operation of A/D converter (Example 1)


(1) 1 is written to ADCE.
(10) 0 is written to ADCE.
ADCE
(2) 1 is written to ADCS.
0 is written to ADCS. (9)
ADCS
The trigger is not acknowledged.
The trigger is not acknowledged.
(3) The software trigger is generated.
(6) The software trigger is generated.
(7) Another high-priority interrupt is generated while high-priority conversion is in progress.
ADTRSWT
INTTM0 is generated.
INTTM0 is generated.
INTTM0
(4) INTRTC is generated.
INTRTC is generated.
INTRTC
(6) Conversion for CH0 starts.
Conversion for CH2 starts.
Conversion for CH3 starts.
(7) The CH0 interrupt is generated and held pending.
(4) Conversion for CH3 starts.
Conversion for CH1 starts.
(8) The low-priority conversion that was held pending starts.
A/D conversion state
Stopped
Conversion standby
Conversion (ANI2)
Conversion (PGA)
Conversion (ANI2)
Conversion (ANI3)
Conversion (ANI16)
Conversion (PGA)
Conversion (ANI2)
Conversion (ANI3)
Conversion (ANI16)
Conversion standby
Stopped
Trigger standby
(3) Conversion for CH0 starts.
(5) Conversion for CH3 ends.
(4) Conversion for CH0 is forcibly terminated.
Conversion for CH0 ends.
Conversion for CH1 ends.
Conversion for CH2 ends.
(8) Conversion for CH3 ends.
Conversion for CH0 ends.
Conversion for CH1 ends.
(9) Conversion for CH0 is forcibly terminated.
The result is stored in the conversion result register.
The result is stored in the conversion result register.
ADCR0
Data0（ANI2）
Data0（ANI2）
The result is stored in the conversion result register.
The result is stored in the conversion result register.
ADCR1
Data1（ANI3）
Data1（ANI3）
The result is stored in the conversion result register.
ADCR2
Data2（ANI16）
(5) The result is stored in the conversion result register.
The result is stored in the conversion result register.
(9) The result is stored in the conversion result register.
ADCR3
Data3（PGA）
Data3（PGA）
Conversion for CH0 is successful.
Conversion for CH0 is successful.
INTAD0
Conversion for CH1 is successful.
Conversion for CH1 is successful.
INTAD1
Conversion for CH2 is successful.
INTAD2
(5) Conversion for CH3 is successful.
Conversion for CH3 is successful.
INTAD3

CHn: channel n (n = 0-3)

(1) In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the standby state. The trigger will not be accepted while ADCS bit is 0.

(2) After the software counts to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the trigger standby state (and conversion does not start at this stage).

(3) If a software trigger is input (ADTRSWT = 1) while the trigger standby state, A/D conversion of CH0 starts.

(4) If a high-priority trigger occurs while the A/D conversion of low-priority CH0 is running, the A/D conversion of CH0 is forced to terminate. At the same time, the A/D conversion of high-priority CH3 starts.

(5) When the A/D conversion of CH3 finishes, the following operations will be performed. If there are no pending A/D conversions, the A/D conversion will enter a trigger waiting state.

- Result of A/D conversion is stored in A/D conversion result register ADCR3.
- The A/D conversion end request signal (INTAD3) is generated.

(6) When a trigger occurs while in the trigger standby state, the corresponding A/D conversion will start. Here, a software trigger for CH0, and the A/D conversion of CH0 starts.

(7) While A/D conversions with the same priority are running, or when a trigger occurs during a high-priority A/D conversion, the A/D conversion is held off. Here, when a software trigger for low-priority CH0 during A/D for high-priority CH3 occurs, the A/D conversion of CH0 is held off.

(8) The A/D conversion that was held off will continue once the ongoing A/D conversion is completed.

The A/D conversions that are held off will be executed sequentially according to their priority.

- In this example, since CH0 and CH1 are held off, the A/D conversions for CH0 and CH1 will be started.

(9) When ADCS is cleared to 0 during A/D conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the standby state.

- In this example, since CH2 was forcefully terminated, the A/D conversion result will not be updated.

(10) When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state.

Remarks: The ADINST register indicating the status of conversion results is available for advanced mode. This register determines whether A/D conversion of each channel is completed or failed. For details, see **20.3.14 Conversion interrupt status register (ADINTST)** in the **RL78/G24 User’s Manual: Hardware**.

2.2 Usage Example 2: Simultaneous Sampling

2.2.1 Overview of specifications

In the A/D converter advanced mode, three sample & hold (S & H) circuits are available, so simultaneous sampling of three analog signals is possible. The S&H circuit samples analog input voltages sent from the input circuit, and then enters the hold state when a certain period has passed. This circuit sends the sampled analog input voltages to the A/D conversion comparator and retains the analog input voltage until A/D conversion ends.

In usage example 2, the advanced mode is used for simultaneous sampling of the P22/ANI2, P23/ANI3, and P24/ANI4 pins, and for A/D conversion of the analog input voltage of the P25/ANI5 pin. A/D conversion is performed for each channel. The conversion for channels 0, 1, and 2, uses a hardware trigger (A/D conversion trigger 0 signal by timer RD: Once per 100 μs), and conversion for channel 3 uses a software trigger. After A/D conversion of each channel ends, an interrupt is generated, and the conversion result is stored in the variable of the internal RAM.

Table 2-4 shows an example of channel settings.

Table 2-4 Example of Channel Settings (Usage Example 2)

	Advanced Mode	Wait Mode	Conversion Mode
Setting	ON	No-wait Mode (fixed)	One-shot conversion mode (fixed)

	Channel Setting Conditions				
	Simultaneous sampling setting ADSn.ADSPSCn[1:0]	Trigger source ADTRn.ADTRSn[3:0]	Priority	Analog input channel ADSn.ADSn[4:0]	A/D conversion result register
Channel 0	1st S&H 01B	timer RD2 A/D conversion trigger 0 1011B	initial value (low)	ANI4 00100B	ADCR0
Channel 1	2nd S&H 10B	timer RD2 A/D conversion trigger 0 1011B	initial value (low)	initial value	ADCR1
Channel 2	3rd S&H 11B	timer RD2 A/D conversion trigger 0 1011B	initial value (low)	initial value	ADCR2
Channel 3	unused 00B	software trigger 1111B	low	ANI5 00101B	ADCR3

Table 2-5 shows the used peripheral functions and their usage.

Table 2-5 Used Peripheral Functions and Their Usage

Peripheral Function	Usage
A/D converter (Advanced mode enabled)	Performs simultaneous sampling of the P22/ANI2, P23/ANI3, and P24/ANI4 pins Performs A/D conversion of analog input voltage of the P25/ANI5 pin
Timer RD2 (TRD2)	Uses a timer RD2 A/D conversion trigger 0 signal as a hardware trigger

Table 2-6 shows the trigger source of each channel and the components to be used.

Table 2-6 Trigger Source of Each Channel and Used Components

Channel	Input Source	Trigger Source	Component Used for Trigger Source
Channel 0	ANI4	Hardware trigger (Timer RD2 A/D conversion trigger 0)	Timer RD2 (generating an A/D conversion trigger 0 signal at 100 μs intervals)
Channel 1	ANI2	Hardware trigger (Timer RD2 A/D conversion trigger 0)	Timer RD2 (generating an A/D conversion trigger 0 signal at 100 μs intervals)
Channel 2	ANI3	Hardware trigger (Timer RD2 A/D conversion trigger 0)	Timer RD2 (generating an A/D conversion trigger 0 signal at 100 μs intervals)
Channel 3	ANI5	Software trigger	-

2.2.2 Operation

The following shows the A/D converter settings.

<Settings>

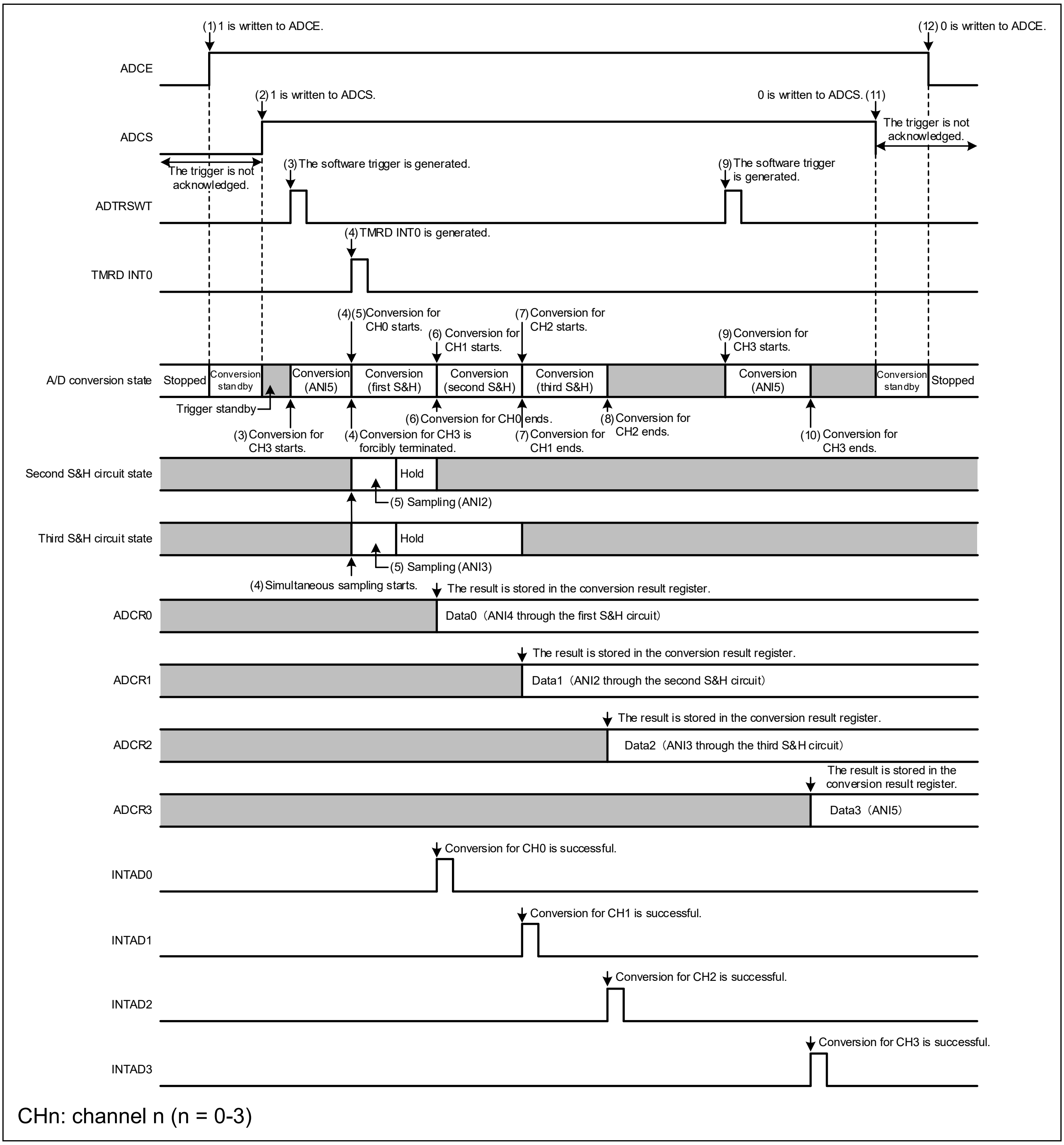
- The advanced mode is used in the A/D converter.
- The resolution is set to 12 bits.
- AV_{REFP} is set as the + side reference voltage of the A/D converter, and AV_{REFM} is set as the - side reference voltage of the A/D converter.
- Three channels (ADS0, ADS1, ADS2) are specified in the simultaneous sampling setting.
- For simultaneous sampling, the trigger source is set to timer RD2 A/D conversion trigger 0 (trigger 0 signal starting A/D conversion at 100 μs intervals), and the 1st S&H circuit input source is set to ANI4.
- For A/D channel 3, the trigger source is set to software trigger, the input source is set to ANI5, and the priority is set to Low.
- The operating voltage mode is set to normal mode 1, the sampling clock cycle is set to 20 fAD Note 1, and the conversion time is set to $86/f_{CLK}$.
- The conversion result comparison upper limit (ADUL) is set to 255, and the lower limit (ADLL) is set to 0. Note 2:
- A/D conversion end interrupts (INTAD0 to INTAD3) are enabled.

Note: 1. If ANI0 to ANI7 are only in use for analog input channels, the sampling clock cycle can be set to 20 fAD, For details, see **20.3.15 A/D conversion sampling mode specification register (ADSPMOD)** in the **RL78/G24 User’s Manual: Hardware**.

Note: 2. When the 12-bit resolution is selected, the set values of the upper and lower limits of A/D conversion results are checked by comparing the ADCRn[11:4] bits with the values of the ADUL and ADLL registers. For details, see **20.3.16 Conversion result comparison upper limit setting register (ADUL)** and **20.3.17 Conversion result comparison lower limit setting register (ADLL)** in the **RL78/G24 User’s Manual: Hardware**.

Figure 2-2 shows the operation of A/D converter.

Figure 2-2 Operation of A/D converter (Example 2)


(1) 1 is written to ADCE.
(12) 0 is written to ADCE.
ADCE
(2) 1 is written to ADCS.
0 is written to ADCS. (11)
The trigger is not acknowledged.
ADCS
The trigger is not acknowledged.
(3) The software trigger is generated.
(9) The software trigger is generated.
ADTRSWT
(4) TMRD INT0 is generated.
TMRD INT0
(4)(5) Conversion for CH0 starts.
(6) Conversion for CH1 starts.
(7) Conversion for CH2 starts.
(9) Conversion for CH3 starts.
A/D conversion state
Stopped
Conversion standby
Conversion (ANI5)
Conversion (first S&H)
Conversion (second S&H)
Conversion (third S&H)
Conversion (ANI5)
Conversion standby
Stopped
Trigger standby
(3) Conversion for CH3 starts.
(4) Conversion for CH3 is forcibly terminated.
(6) Conversion for CH0 ends.
(7) Conversion for CH1 ends.
(8) Conversion for CH2 ends.
(10) Conversion for CH3 ends.
Second S&H circuit state
Hold
(5) Sampling (ANI2)
Third S&H circuit state
Hold
(5) Sampling (ANI3)
(4) Simultaneous sampling starts.
The result is stored in the conversion result register.
ADCR0
Data0（ANI4 through the first S&H circuit）
The result is stored in the conversion result register.
ADCR1
Data1（ANI2 through the second S&H circuit）
The result is stored in the conversion result register.
ADCR2
Data2（ANI3 through the third S&H circuit）
The result is stored in the conversion result register.
ADCR3
Data3（ANI5）
Conversion for CH0 is successful.
INTAD0
Conversion for CH1 is successful.
INTAD1
Conversion for CH2 is successful.
INTAD2
Conversion for CH3 is successful.
INTAD3

CHn: channel n (n = 0-3)

(1) In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the standby state. The trigger will not be accepted while ADCS bit is 0.

(2) After the software counts to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the trigger standby state (and conversion does not start at this stage).

(3) If a software trigger is input (ADTRSWT = 1) while the trigger standby state, A/D conversion of CH0 starts.

(4) If a simultaneous sampling trigger TMRD INT0 occurs while the A/D conversion of low-priority CH3 is running, the A/D conversion of CH3 is forced to terminate. At the same time, simultaneous sampling configured for CH0-2 starts.

(5) In the case of simultaneous sampling, when a trigger occurs, CH0 configured as the1st S&H (ADS0.ADSPSCn[1:0] = 01B) performs the A/D conversion as usual. CH1 configured as the 2nd S&H (ADS1.ADSPSCn[1:0] = 10B) and CH2 configured as the 3rdS&H (ADS2.ADSPSCn[1:0] = 11B) hold the values sampled by S&H circuits.

(6) When the A/D conversion of CH0 finishes, the following operations will be performed. The A/D conversion for CH1 configured as the 2nd S&H is automatically initiated. The analog input to be converted for CH1 is the value held by the 2nd S&H circuit.

- Result of A/D conversion is stored in A/D conversion result register ADCR0.
- The A/D conversion end request signal (INTAD0) is generated.

(7) When the A/D conversion of CH1 finishes, the following operations will be performed. The A/D conversion for CH2 configured as the 3rd S&H is automatically initiated. The analog input to be converted for CH2 is the value held by the 3rd S&H circuit.

- Result of A/D conversion is stored in A/D conversion result register ADCR1.
- The A/D conversion end request signal (INTAD1) is generated.

(8) When the A/D conversion of CH2 finishes, the following operations will be performed. If there are no pending A/D conversions, the A/D conversion will enter a trigger waiting state.

- Result of A/D conversion is stored in A/D conversion result register ADCR2.
- The A/D conversion end request signal (INTAD2) is generated.

(9) Generate a software trigger to start the A/D conversion for CH3.

(10) When the A/D conversion of CH3 finishes, the following operations will be performed. If there are no pending A/D conversions, the A/D conversion will enter a trigger waiting state.

- Result of A/D conversion is stored in A/D conversion result register ADCR3.
- The A/D conversion end request signal (INTAD3) is generated.

(11) When ADCS is cleared to 0 while in the trigger standby state, the A/D converter enters the A/D conversion standby state.

(12) When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state.

Remarks: The ADINST register indicating the status of conversion results is available for advanced mode. This register determines whether A/D conversion of each channel is completed or failed. For details, see **20.3.14 Conversion interrupt status register (ADINTST)** in the **RL78/G24 User's Manual: Hardware**.

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	Descriptions
MCU used	RL78/G24（R7F101GLG）
Operating frequency	Advanced mode enabled • 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 - 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)

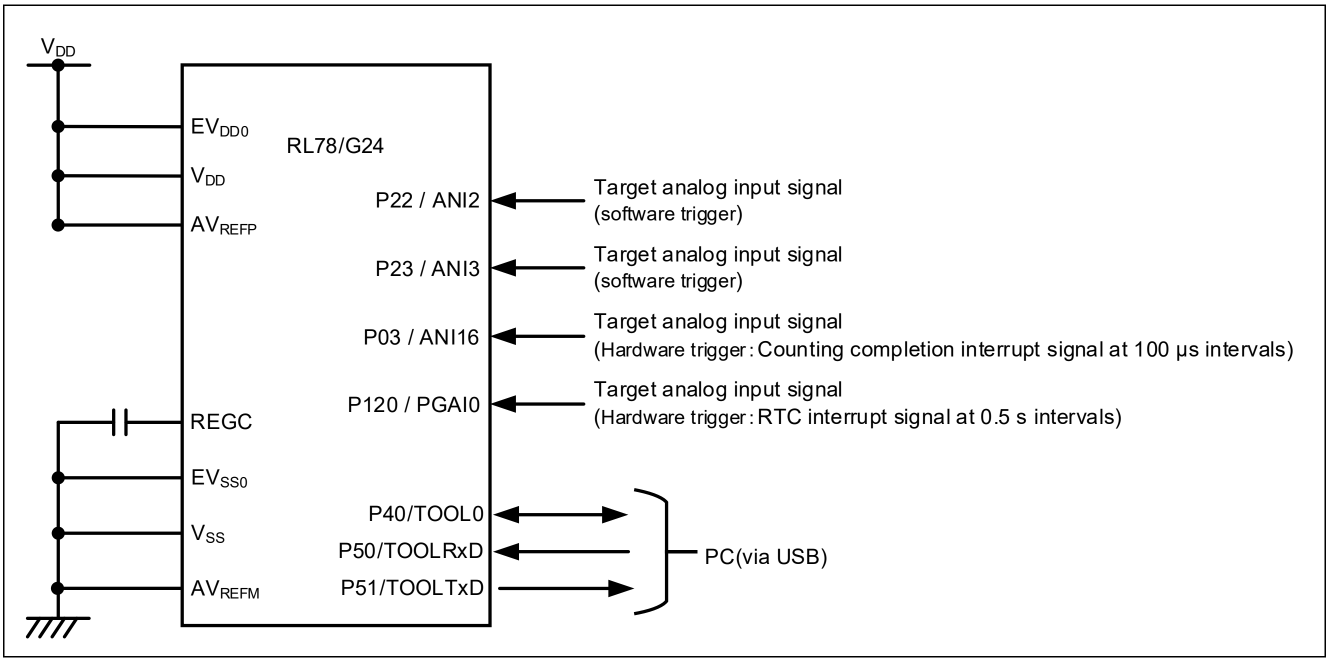
4. Hardware Description

4.1 Example 1：Mixed Hardware and Software Trigger

4.1.1 Example of Hardware Configuration

Figure 4-1 shows an example of the hardware configuration for Example 1.

Figure 4-1 Example of Hardware Configuration


V_{DD}
EV_{DD0}
RL78/G24
V_{DD}
AV_{REFP}
P22 / ANI2
Target analog input signal
(software trigger)
P23 / ANI3
Target analog input signal
(software trigger)
P03 / ANI16
Target analog input signal
(Hardware trigger : Counting completion interrupt signal at 100 μs intervals)
P120 / PGAI0
Target analog input signal
(Hardware trigger : RTC interrupt signal at 0.5 s intervals)
REGC
EV_{SS0}
V_{SS}
AV_{REFM}
P40/TOOL0
P50/TOOLRxD
P51/TOOLTxD
PC(via USB)

Note 1. This simplified circuit diagram was created to show an overview of connections only. When 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.

4.1.2 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
P23 / ANI3	Input	A/D converter Analog input
P03 / ANI16	Input	A/D converter Analog input
P120 / PGAI0	Input	Programmable Gain Amplifier Analog input

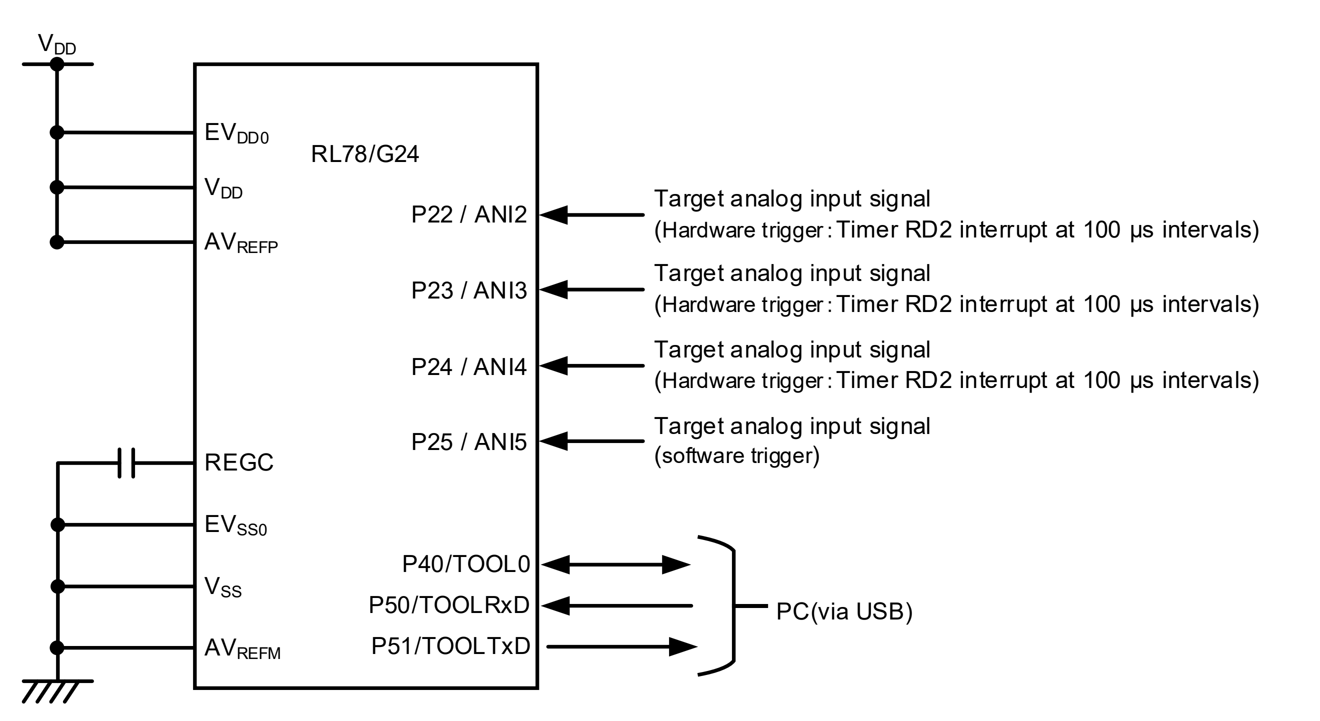
Caution: In this application note, only the used pins are processed. When designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.

4.2 Example 2：Simultaneous sampling

4.2.1 Example of Hardware Configuration

Figure 4-2 shows an example of the hardware configuration for Example 2.

Figure 4-2 Example of Hardware Configuration


V_{DD}
EV_{DD0}
RL78/G24
V_{DD}
AV_{REFP}
P22 / ANI2
P23 / ANI3
P24 / ANI4
P25 / ANI5
Target analog input signal
(Hardware trigger : Timer RD2 interrupt at 100 μs intervals)
Target analog input signal
(Hardware trigger : Timer RD2 interrupt at 100 μs intervals)
Target analog input signal
(Hardware trigger : Timer RD2 interrupt at 100 μs intervals)
Target analog input signal
(software trigger)
REGC
EV_{SS0}
V_{SS}
AV_{REFM}
P40/TOOL0
P50/TOOLRxD
P51/TOOLTxD
PC(via USB)

Note 1. This simplified circuit diagram was created to show an overview of connections only. When 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.

4.2.2 List of used Pins

Table 4-2 shows the pins used and their functions.

Table 4-2 Pins used and Their Functions

Pin Name	I/O	Functions
P22 / ANI2	Input	A/D converter Analog input
P23 / ANI3	Input	A/D converter Analog input
P24 / ANI4	Input	A/D converter Analog input
P25 / ANI5	Input	A/D converter Analog input

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 Usage Example 1: Operation Triggered by Hardware and Software

5.1.1 Smart Configurator Settings

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

5.1.1.1 System settings

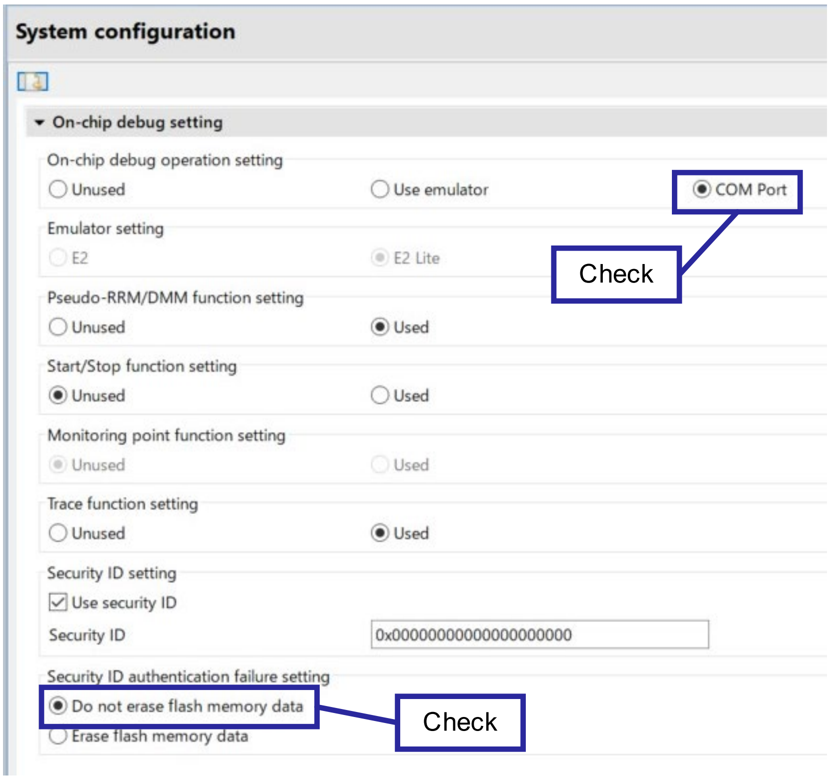
The following shows the system settings used in this sample program.

Note that the system settings used in this sample program 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 program (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 Configuration (e^2 studio, 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

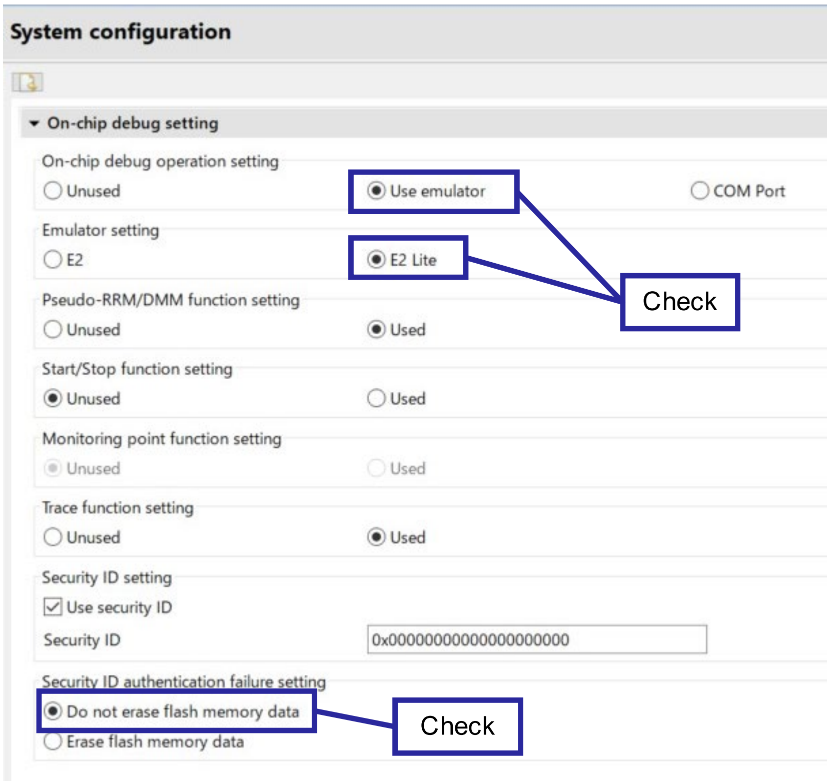
Trace function setting
○ Unused ● Used

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

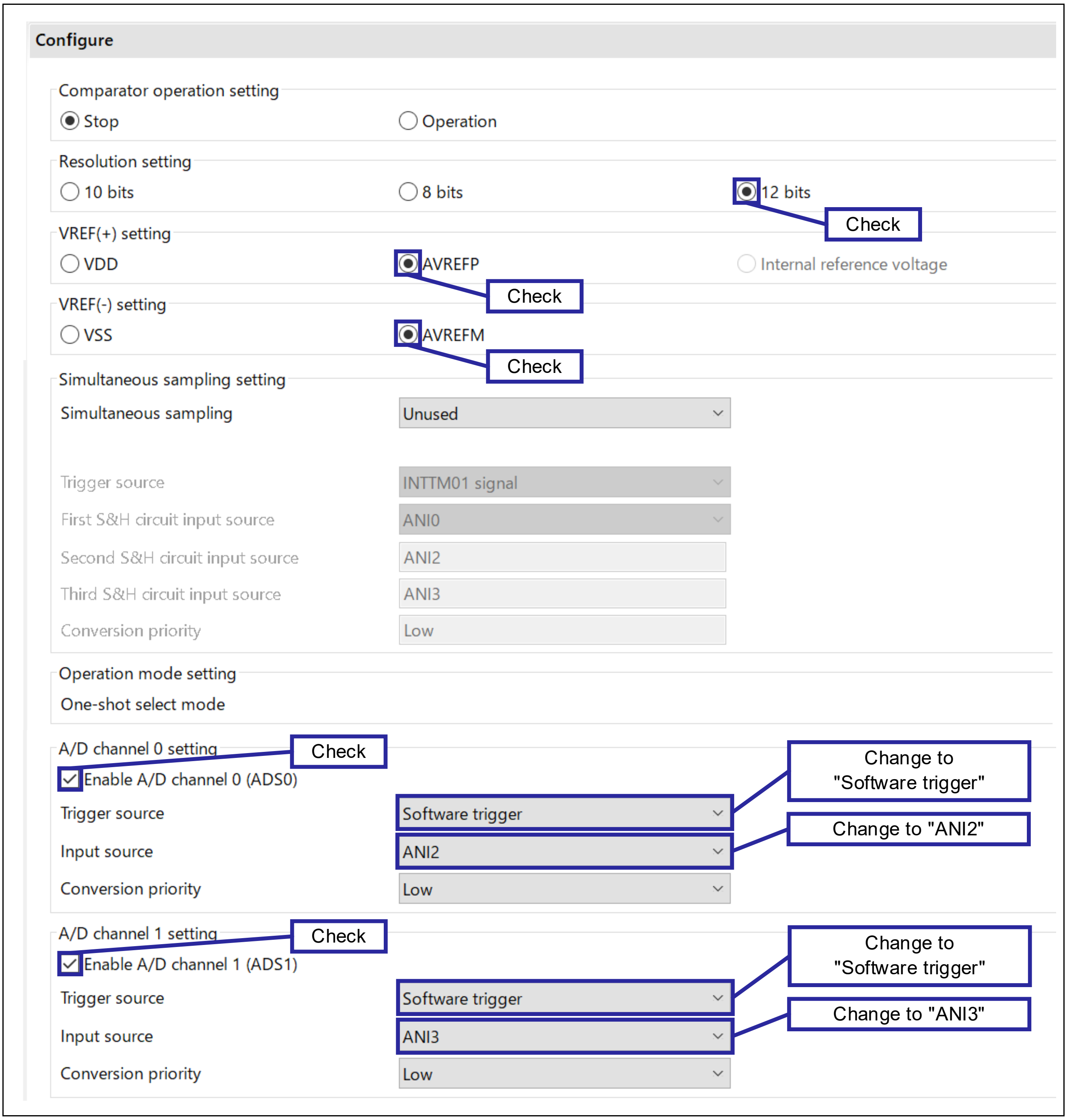
5.1.1.2 Component Configurations

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

Table 5-1 Component Configurations (A/D Converter)

Item	Content
Component	A/D converter
Configuration Name	Config_ADC
Resource	ADC
Operation mode	Advanced mode enabled

Figure 5-3 Configuration of A/D Converter (1/2)



Configure

Comparator operation setting
● Stop ○ Operation

Resolution setting
○ 10 bits ○ 8 bits ● 12 bits
Check

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

VREF(-) setting
○ VSS ● AVREFM
Check

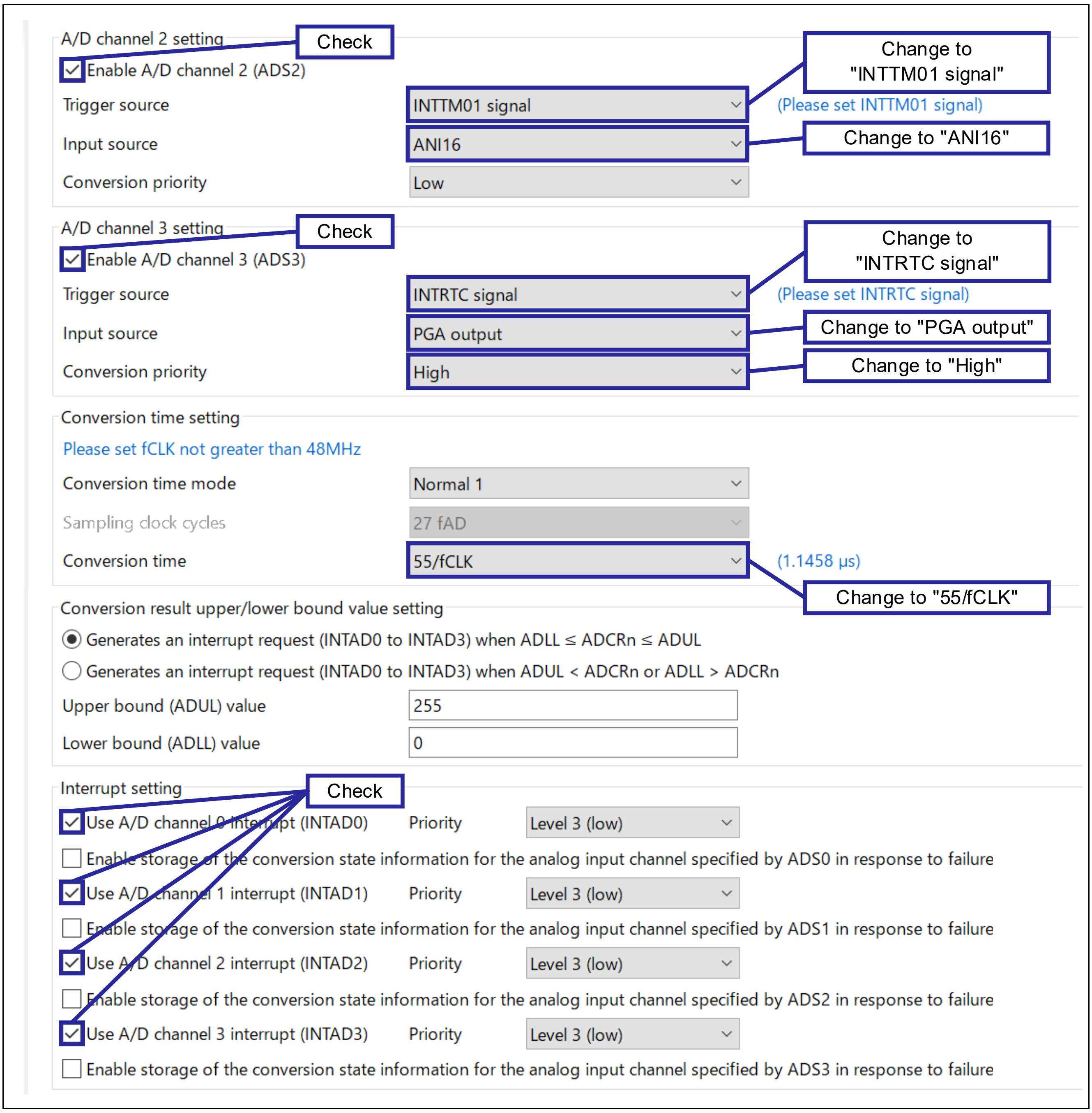
Simultaneous sampling setting
Simultaneous sampling: Unused
Trigger source: INTTM01 signal
First S&H circuit input source: ANI0
Second S&H circuit input source: ANI2
Third S&H circuit input source: ANI3
Conversion priority: Low

Operation mode setting
One-shot select mode

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

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

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



A/D channel 2 setting

☑ Enable A/D channel 2 (ADS2)

Check

Trigger source: INTTM01 signal

Change to "INTTM01 signal"

(Please set INTTM01 signal)

Input source: ANI16

Change to "ANI16"

Conversion priority: Low

A/D channel 3 setting

☑ Enable A/D channel 3 (ADS3)

Check

Trigger source: INTRTC signal

Change to "INTRTC signal"

(Please set INTRTC signal)

Input source: PGA output

Change to "PGA output"

Conversion priority: High

Change to "High"

Conversion time setting

Please set fCLK not greater than 48MHz

Conversion time mode: Normal 1

Sampling clock cycles: 27 fAD

Conversion time: 55/fCLK (1.1458 μs)

Change to "55/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

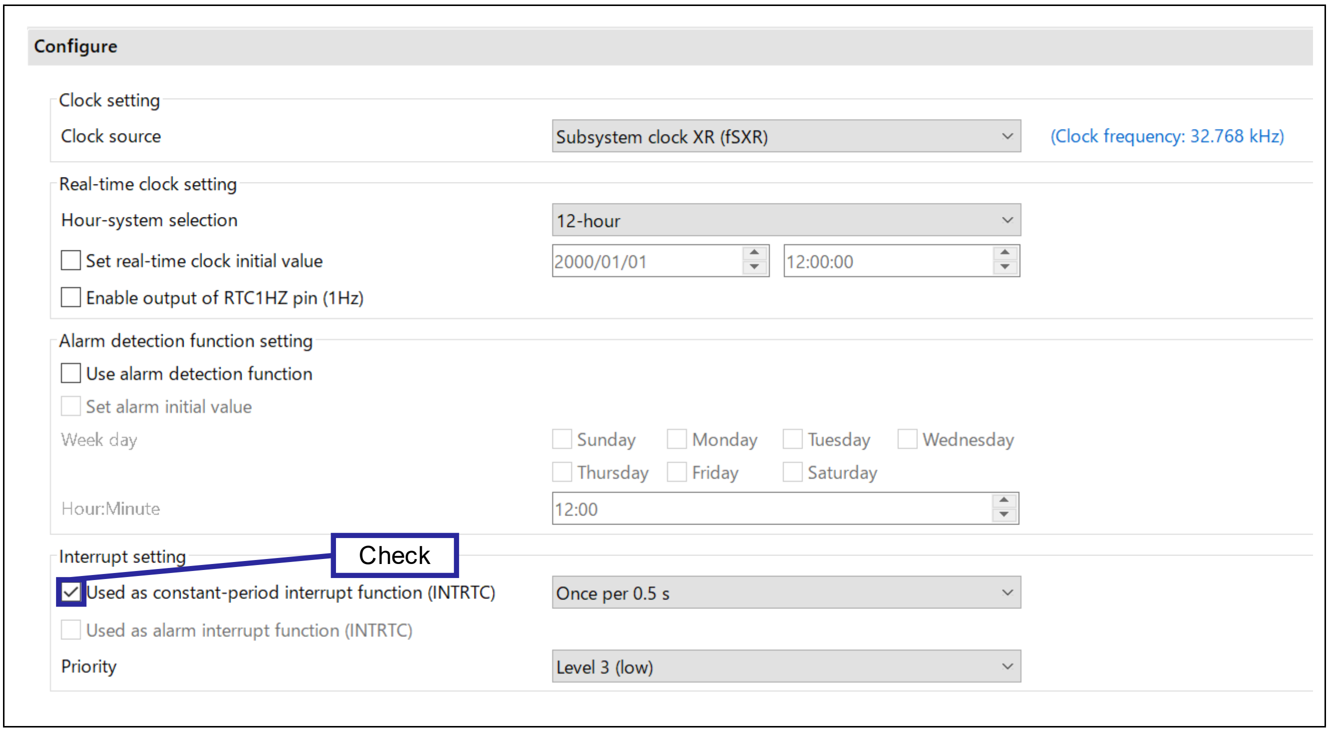
☑ Use A/D channel 3 interrupt (INTAD3) Priority: Level 3 (low)

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

Table 5-2 Component Configuration (Realtime Clock)

Item	Content
Component	Realtime clock
Configuration Name	Config_RTC
Resource	RTC

Figure 5-5 Configuration of Realtime Clock



Configure

Clock setting

Clock source: Subsystem clock XR (fSXR) (Clock frequency: 32.768 kHz)

Real-time clock setting

Hour-system selection: 12-hour

□ Set real-time clock initial value 2000/01/01 12:00:00

□ Enable output of RTC1HZ pin (1Hz)

Alarm detection function setting

□ Use alarm detection function

□ Set alarm initial value

Week day: □ Sunday □ Monday □ Tuesday □ Wednesday □ Thursday □ Friday □ Saturday

Hour:Minute: 12:00

Interrupt setting

☑ Used as constant-period interrupt function (INTRTC): Once per 0.5 s

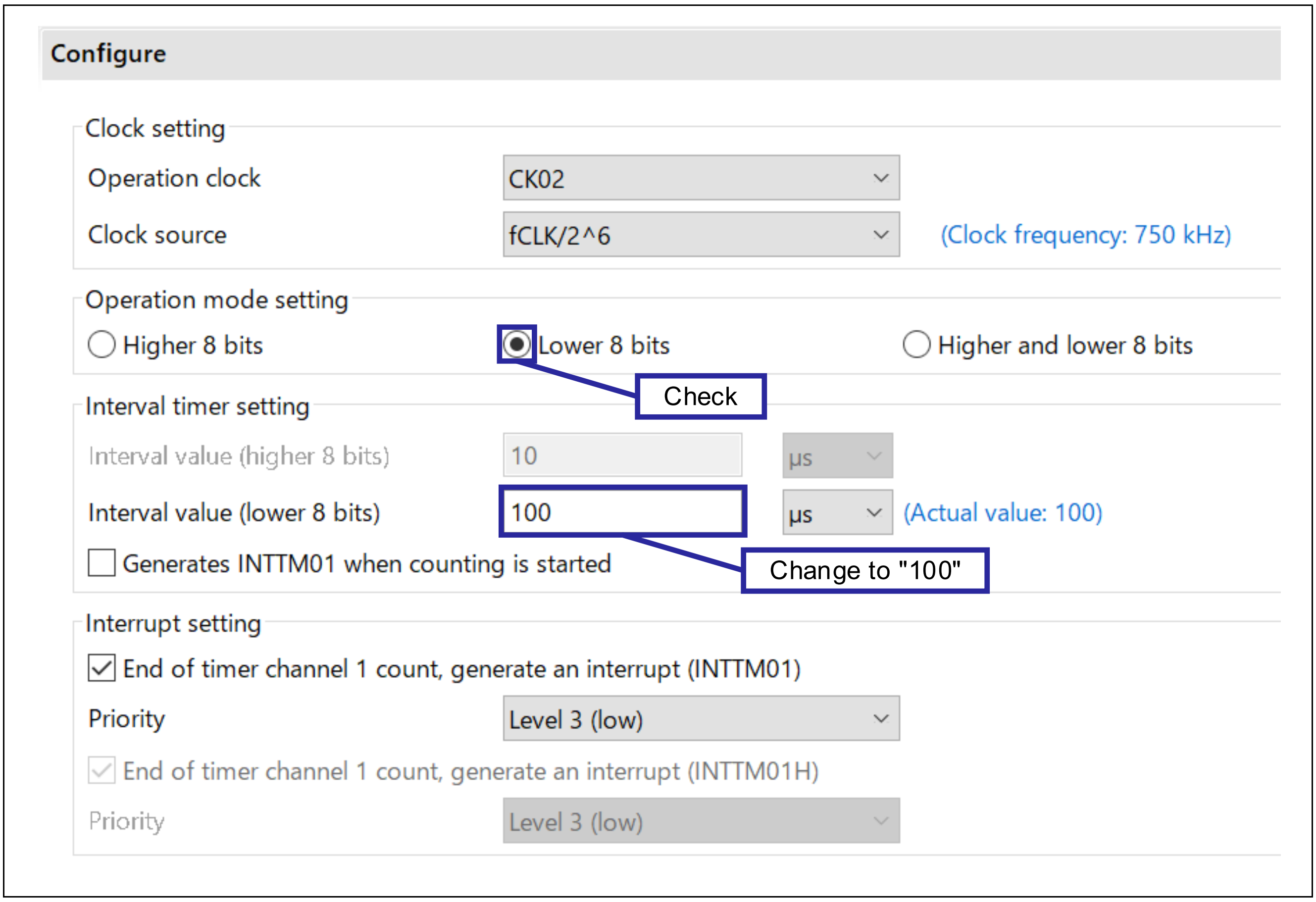
□ Used as alarm interrupt function (INTRTC)

Priority: Level 3 (low)

Table 5-3 Component Configurations (Interval Timer)

Item	Content
Component	Interval timer
Configuration Name	Config_TAU0_1
Resource	TAU0_1
Operation mode	8bit counter mode

Figure 5-6 Configuration of Interval Timer



Configure

Clock setting

Operation clock: CK02

Clock source: fCLK/2^6 (Clock frequency: 750 kHz)

Operation mode setting

○ Higher 8 bits ● Lower 8 bits ○ Higher and lower 8 bits

Check

Interval timer setting

Interval value (higher 8 bits): 10 μs

Interval value (lower 8 bits): 100 μs (Actual value: 100)

Change to "100"

□ Generates INTTM01 when counting is started

Interrupt setting

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

Priority: Level 3 (low)

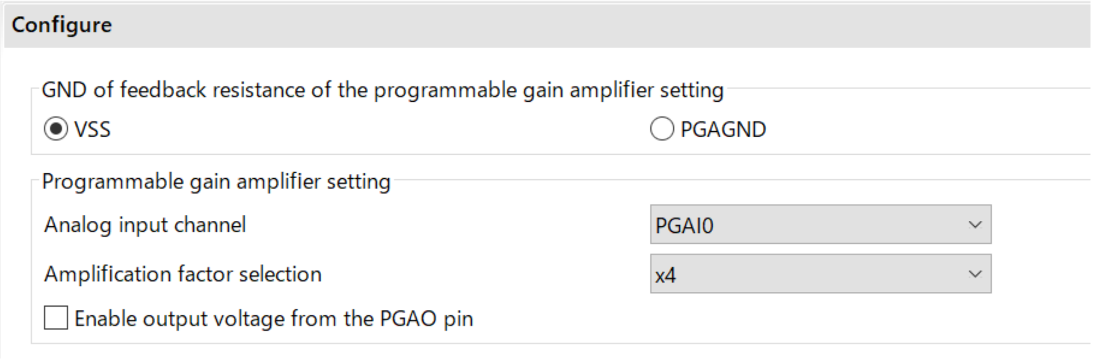
☑ End of timer channel 1 count, generate an interrupt (INTTM01H)

Priority: Level 3 (low)

Table 5-4 Component Configurations (Programmable Gain Amplifier)

Item	Content
Component	Programmable gain amplifier
Configuration Name	Config_PGA
Resource	PGA

Figure 5-7 Configuration of Programmable Gain Amplifier



Configure

GND of feedback resistance of the programmable gain amplifier setting

● VSS ○ PGAGND

Programmable gain amplifier setting

Analog input channel: PGAI0

Amplification factor selection: x4

□ Enable output voltage from the PGAO pin

5.1.2 Folder Structure

Table 5-5 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 5-5 Folder Structure

Folder/File Name	Description	Generated by Smart Configurator
\r01an6973_adc_advanced_mode_trigger_mix<DIR>[NOTE2]	Sample code folder	
\src<DIR>	Program storage folder	
main.c	Sample code source 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	√
\Config_PGA<DIR>	PGA program storage folder	√
Config_PGA.c	PGA source file	√
Config_PGA.h	PGA header file	√
Config_PGA_user.c	PGA interrupt source file	√[NOTE 1]
\Config_RTC<DIR>	RTC program storage folder	√
Config_RTC.c	RTC source file	√
Config_RTC.h	RTC header file	√
Config_RTC_user.c	RTC interrupt source file	√[NOTE 1]
\Config_TAU0_1<DIR>	TAU0_1 program storage folder	√
Config_TAU0_1.c	TAU0_1 source file	√
Config_TAU0_1.h	TAU0_1 header file	√
Config_TAU0_1_user.c	TAU0_1 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.

Note: 1. This sample code does not use it.

Note: 2. The sample code for IAR contains the r01an6973_adc_advanced_mode_trigger_mix.ipcf file. For details on the .ipcf file, please refer to “RL78 Smart Configurator User’s Guide: IAR” (R20AN0581).

5.1.3 List of Option Byte Settings

Table 5-6 shows the option byte settings.

Table 5-6 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

5.1.4 List of Constants

Constant is not used in the sample code.

5.1.5 List of Variables

Table 5-7 shows the list of variables used in the sample code.

Table 5-7 Variables used in the sample code

Variable Name	Type	Contents	Function that uses the variable
g_result_buffer0	uint16_t	Storage of A/D conversion results for Channel 0	r_Config_ADC_ad0_interrupt
g_result_buffer1	uint16_t	Storage of A/D conversion results for Channel 1	r_Config_ADC_ad1_interrupt
g_result_buffer2	uint16_t	Storage of A/D conversion results for Channel 2	r_Config_ADC_ad2_interrupt
g_result_buffer3	uint16_t	Storage of A/D conversion results for Channel 3	r_Config_ADC_ad3_interrupt

5.1.6 Function list

Table 5-8 shows the functions used in the sample code. However, functions generated by the Smart Configurator that have not been modified are excluded.

Table 5-8 Function List

Function Name	Overview	Source file
main	Main processing	main.c
r_Config_ADC_ad0_interrupt	A/D converter channel 0 interrupt processing	Config_ADC_user.c
r_Config_ADC_ad1_interrupt	A/D converter channel 1 interrupt processing	Config_ADC_user.c
r_Config_ADC_ad2_interrupt	A/D converter channel 2 interrupt processing	Config_ADC_user.c
r_Config_ADC_ad3_interrupt	A/D converter channel 3 interrupt processing	Config_ADC_user.c

5.1.7 Function specifications

The following describes the function specifications of the sample code.

[Function name] main

Overview	Main processing
Headers	r_smc_entry.h
Declaration	void main (void);
Description	This function starts operation of the PGA, TAU, RTC, and A/D converter.
Arguments	None
Return values	None
Remarks	None

[Function name] r_Config_ADC_ad0_interrupt

Overview	A/D converter channel 0 interrupt processing
Headers	r_cg_macrodriver.h, r_cg_userdefine.h, Config_ADC.h
Declaration	static void __near r_Config_ADC_ad0_interrupt(void);
Description	When A/D conversion ends, this function reads the A/D conversion result from the ADCR0 register and then stores the result in the variable of the internal RAM.
Arguments	None
Return values	None
Remarks	None

[Function name] r_Config_ADC_ad1_interrupt

Overview	A/D converter channel 1 interrupt processing
Headers	r_cg_macrodriver.h, r_cg_userdefine.h, Config_ADC.h
Declaration	static void __near r_Config_ADC_ad1_interrupt(void);
Description	When A/D conversion ends, this function reads the A/D conversion result from the ADCR1 register and then stores the result in the variable of the internal RAM.
Arguments	None
Return values	None
Remarks	None

[Function name] r_Config_ADC_ad2_interrupt

Overview	A/D converter channel 2 interrupt processing
Headers	r_cg_macrodriver.h, r_cg_userdefine.h, Config_ADC.h
Declaration	static void __near r_Config_ADC_ad2_interrupt(void);
Description	When A/D conversion ends, this function reads the A/D conversion result from the ADCR2 register and then stores the result in the variable of the internal RAM.
Arguments	None
Return values	None
Remarks	None

[Function name] r_Config_ADC_ad3_interrupt

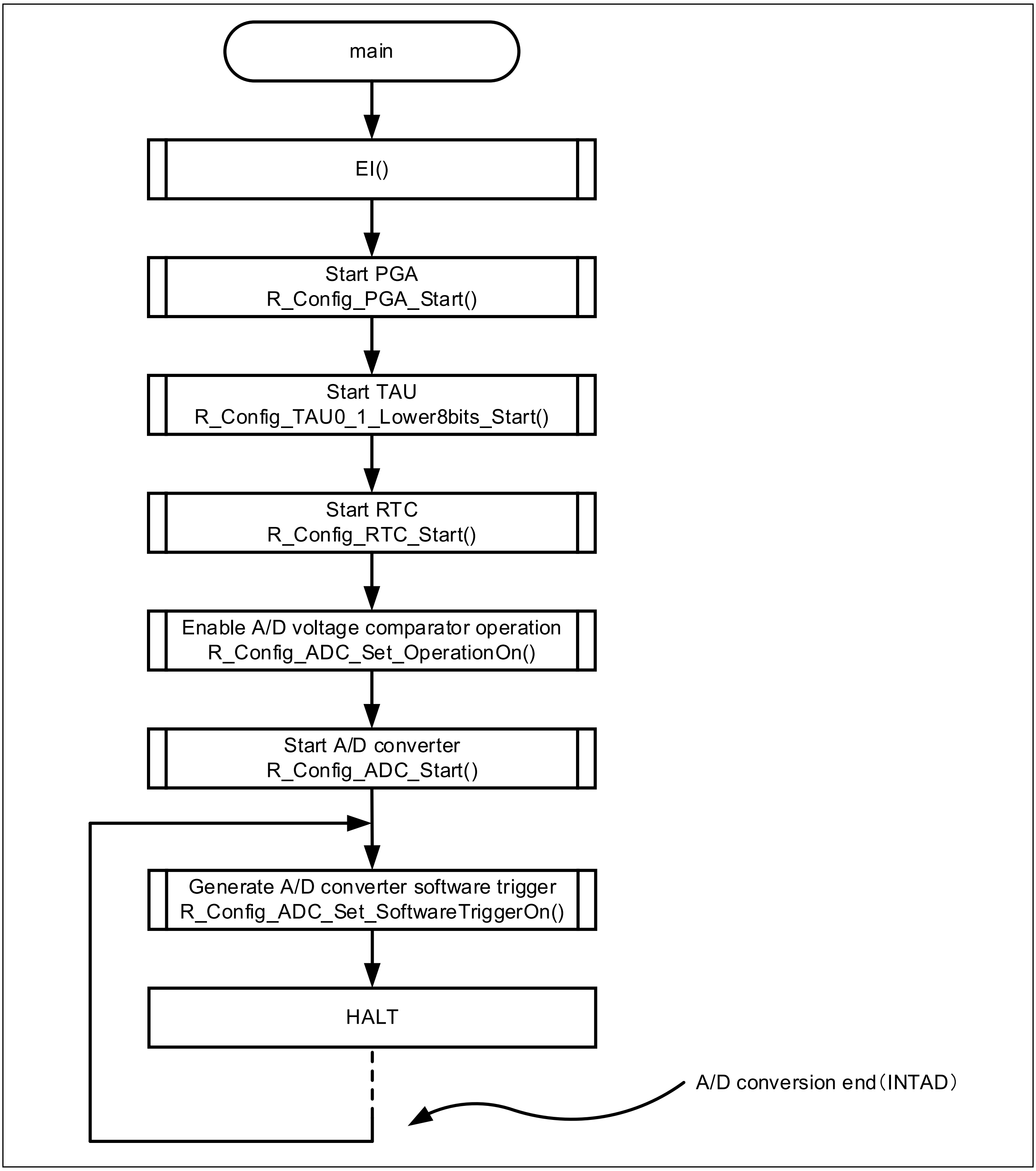
Overview	A/D converter channel 3 interrupt processing
Headers	r_cg_macrodriver.h, r_cg_userdefine.h, Config_ADC.h
Declaration	static void __near r_Config_ADC_ad3_interrupt(void);
Description	When A/D conversion ends, this function reads the A/D conversion result from the ADCR3 register and then stores the result in the variable of the internal RAM.
Arguments	None
Return values	None
Remarks	None

5.1.8 Flowchart

5.1.8.1 Main Process

Figure 5-8 shows the flowchart for the main process.

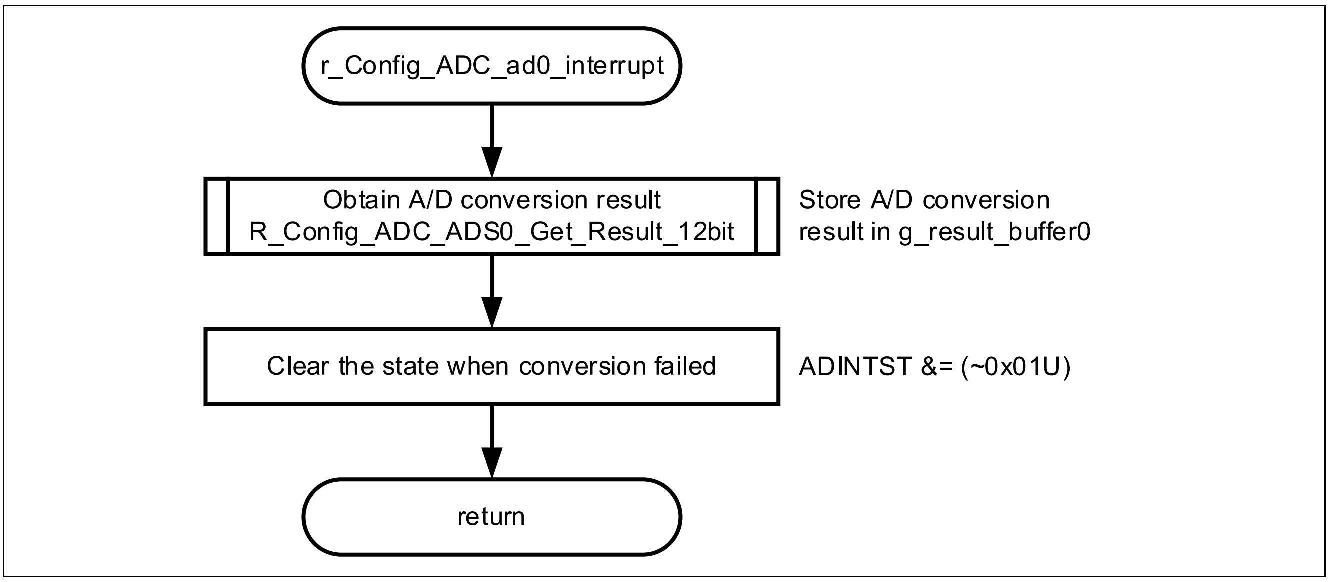
Figure 5-8 Main Process


main
EI()
Start PGA
R_Config_PGA_Start()
Start TAU
R_Config_TAU0_1_Lower8bits_Start()
Start RTC
R_Config_RTC_Start()
Enable A/D voltage comparator operation
R_Config_ADC_Set_OperationOn()
Start A/D converter
R_Config_ADC_Start()
Generate A/D converter software trigger
R_Config_ADC_Set_SoftwareTriggerOn()
HALT
A/D conversion end（INTAD）

5.1.8.2 r_Config_ADC_ad0_interrupt function

Figure 5-9 shows the flowchart of r_Config_ADC_ad0_interrupt function.

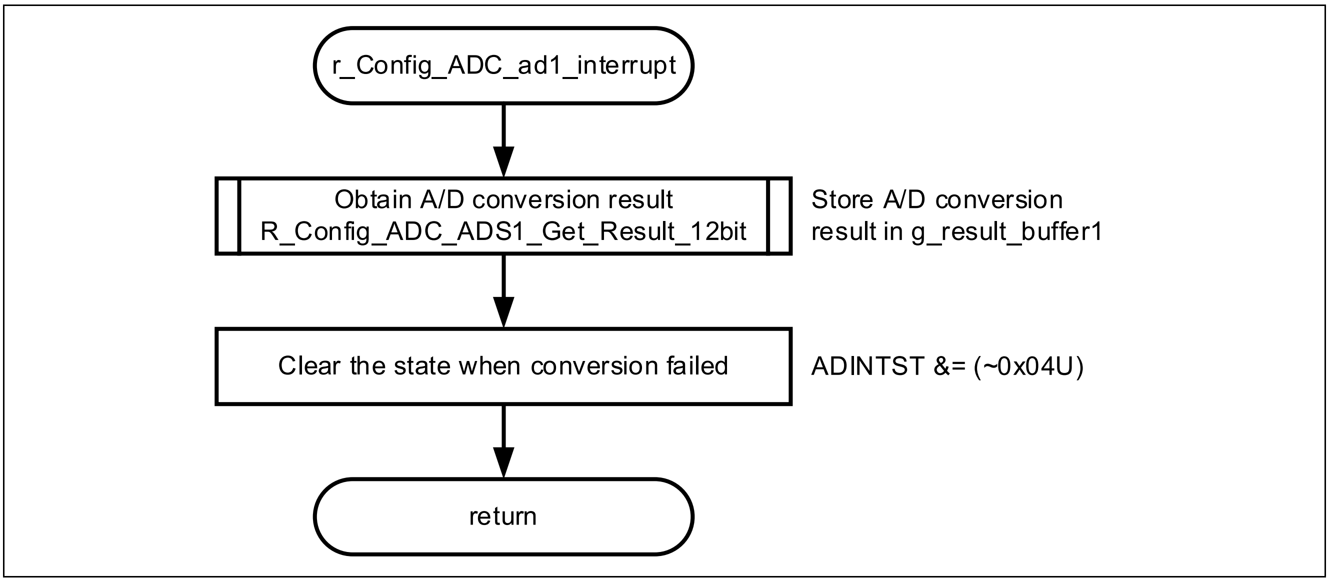
Figure 5-9 r_Config_ADC_ad0_interrupt function


r_Config_ADC_ad0_interrupt
Obtain A/D conversion result
R_Config_ADC_ADS0_Get_Result_12bit
Store A/D conversion result in g_result_buffer0
Clear the state when conversion failed
ADINTST &= (~0x01U)
return

5.1.8.3 r_Config_ADC_ad1_interrupt function

Figure 5-10 shows the flowchart of r_Config_ADC_ad1_interrupt function.

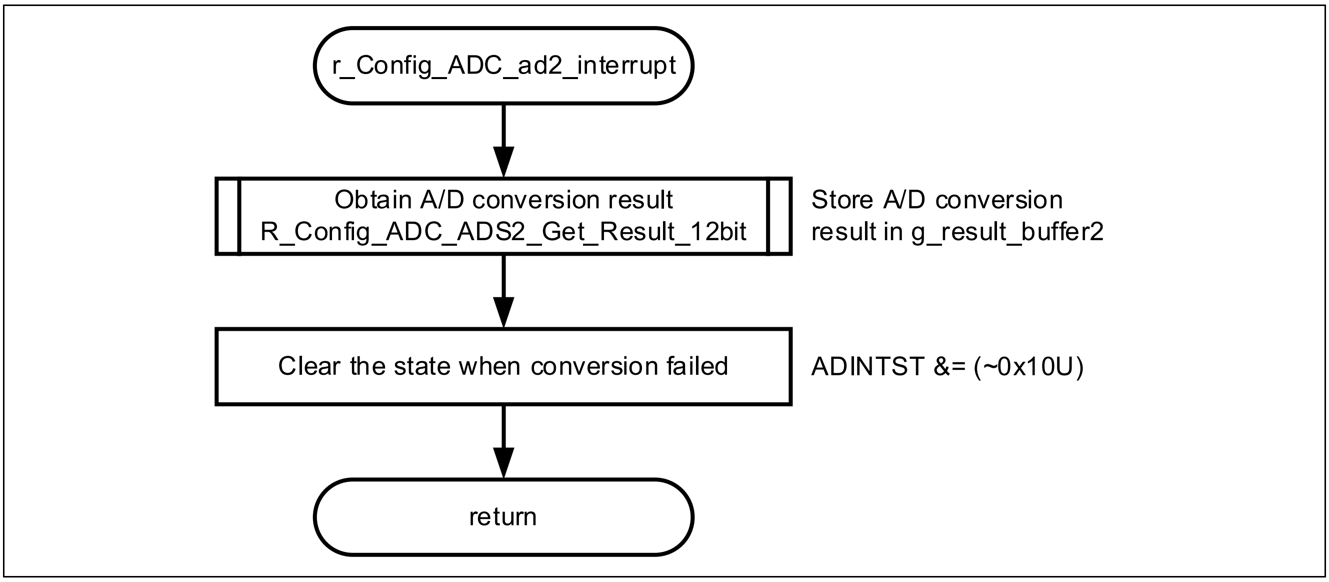
Figure 5-10 r_Config_ADC_ad1_interrupt function


r_Config_ADC_ad1_interrupt
Obtain A/D conversion result
R_Config_ADC_ADS1_Get_Result_12bit
Store A/D conversion result in g_result_buffer1
Clear the state when conversion failed
ADINTST &= (~0x04U)
return

5.1.8.4 r_Config_ADC_ad2_interrupt function

Figure 5-11 shows the flowchart of r_Config_ADC_ad2_interrupt function.

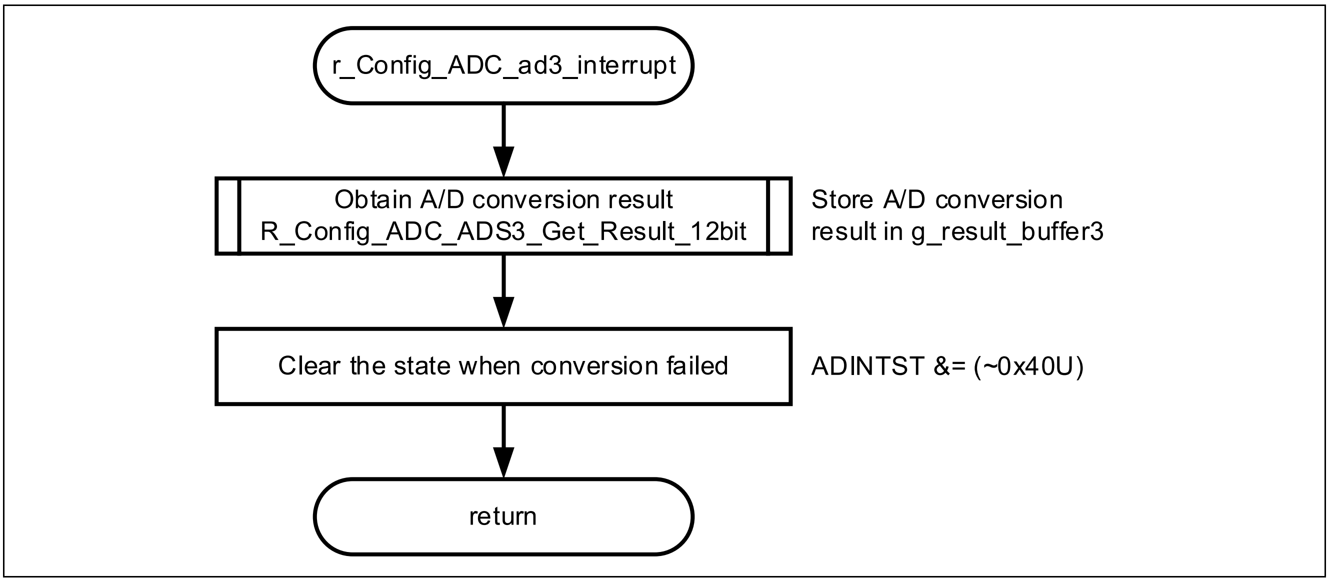
Figure 5-11 r_Config_ADC_ad2_interrupt function


r_Config_ADC_ad2_interrupt
Obtain A/D conversion result
R_Config_ADC_ADS2_Get_Result_12bit
Store A/D conversion
result in g_result_buffer2
Clear the state when conversion failed
ADINTST &= (~0x10U)
return

5.1.8.5 r_Config_ADC_ad3_interrupt function

Figure 5-12 shows the flowchart of r_Config_ADC_ad3_interrupt function.

Figure 5-12 r_Config_ADC_ad3_interrupt function


r_Config_ADC_ad3_interrupt
Obtain A/D conversion result
R_Config_ADC_ADS3_Get_Result_12bit
Store A/D conversion
result in g_result_buffer3
Clear the state when conversion failed
ADINTST &= (~0x40U)
return

5.2 Usage Example 2：Simultaneous Sampling

5.2.1 Smart Configurator Settings

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

5.2.1.1 System settings

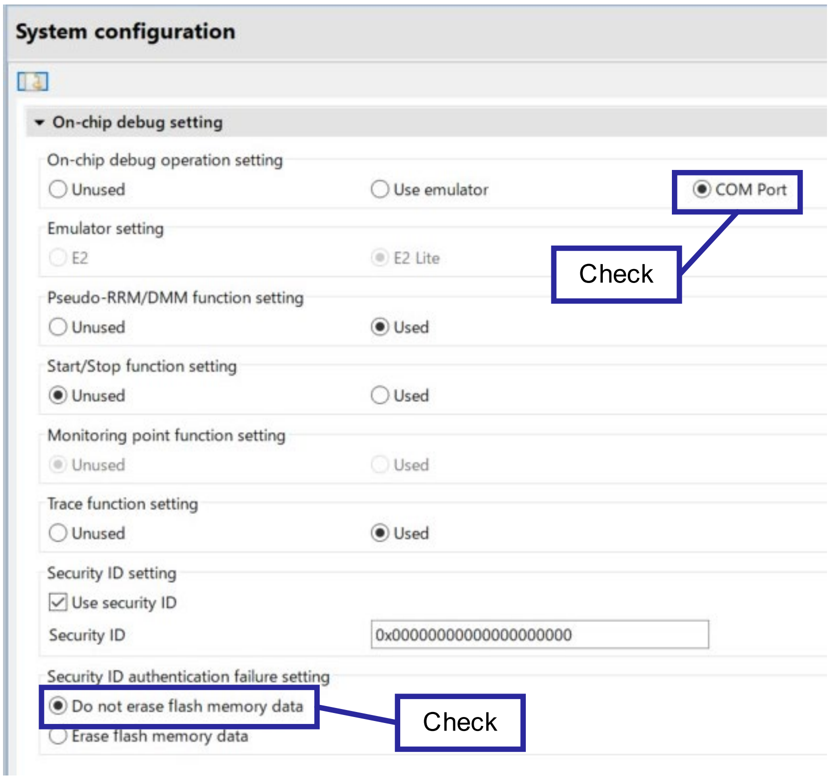
The following shows the system settings used in this sample program.

Note that the system settings used in this sample program are the same for integrated development environments e2 studio and CS+ but are different for IAR. Specify appropriate settings according to your environment.

Figure 5-13 shows the system settings used in this sample program (e2 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 (e2 studio and CS+). For details, see 7.1 Using COM Port Debugging with the e2 studio in the RL78/G24 Fast Prototyping Board User's Manual (R20UT5091J).

Figure 5-13 System Configuration (e^2 studio, 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

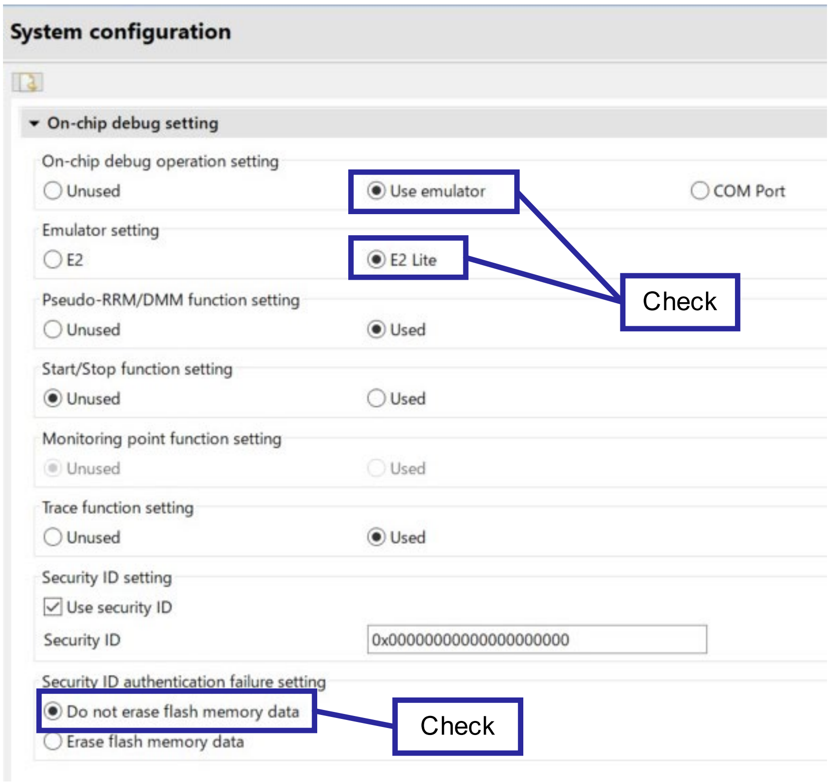
Security ID authentication failure setting
● Do not erase flash memory data
○ Erase flash memory data

Check

Check

Figure 5-14 shows the system configurations used in this sample program for IAR.

Figure 5-14 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

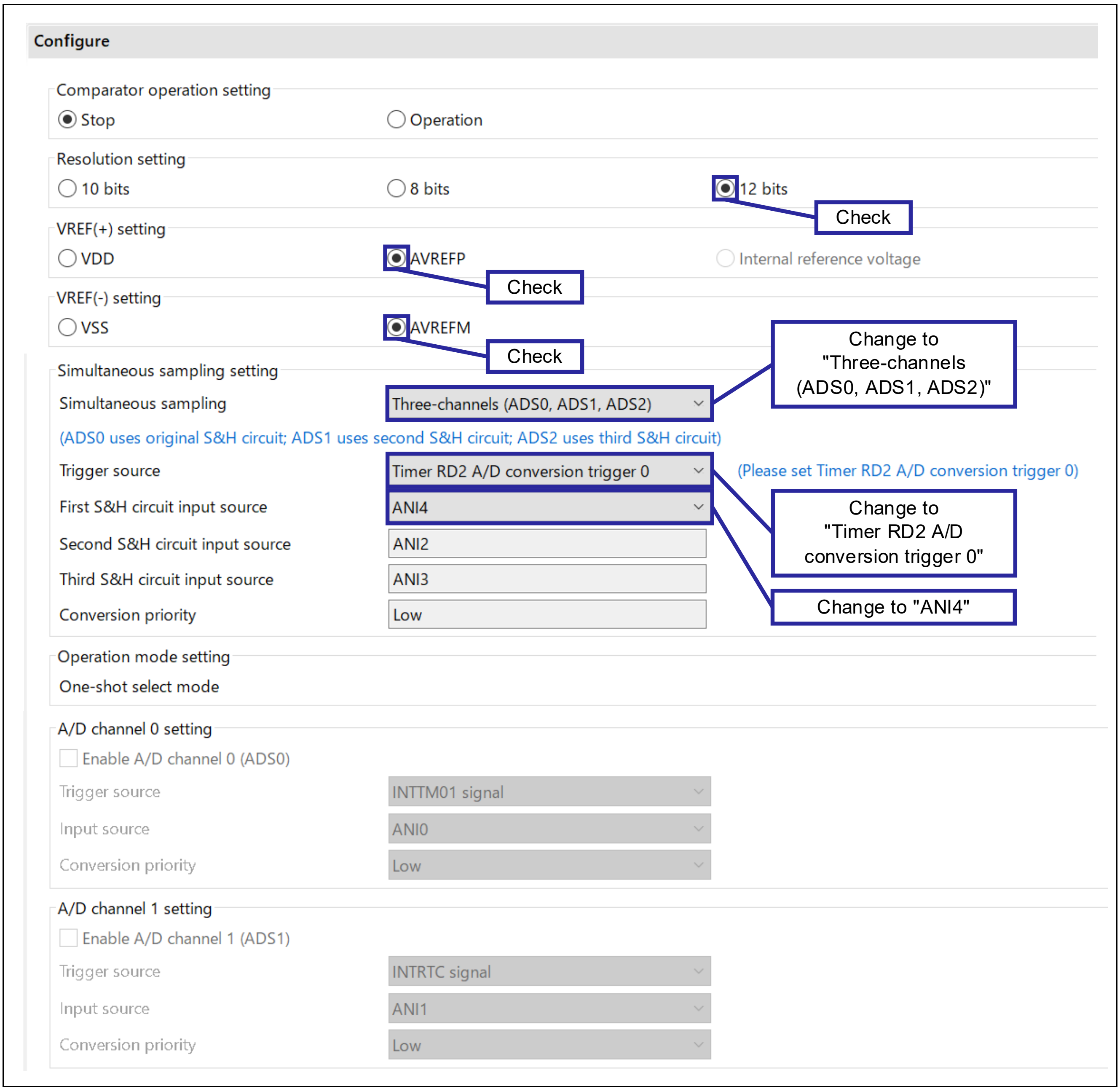
5.2.1.2 Component Configurations

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

Table 5-9 Component Configurations (A/D Converter)

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

Figure 5-15 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: Three-channels (ADS0, ADS1, ADS2)
(ADS0 uses original S&H circuit; ADS1 uses second S&H circuit; ADS2 uses third S&H circuit)
Trigger source: Timer RD2 A/D conversion trigger 0 (Please set Timer RD2 A/D conversion trigger 0)
First S&H circuit input source: ANI4
Second S&H circuit input source: ANI2
Third S&H circuit input source: ANI3
Conversion priority: Low

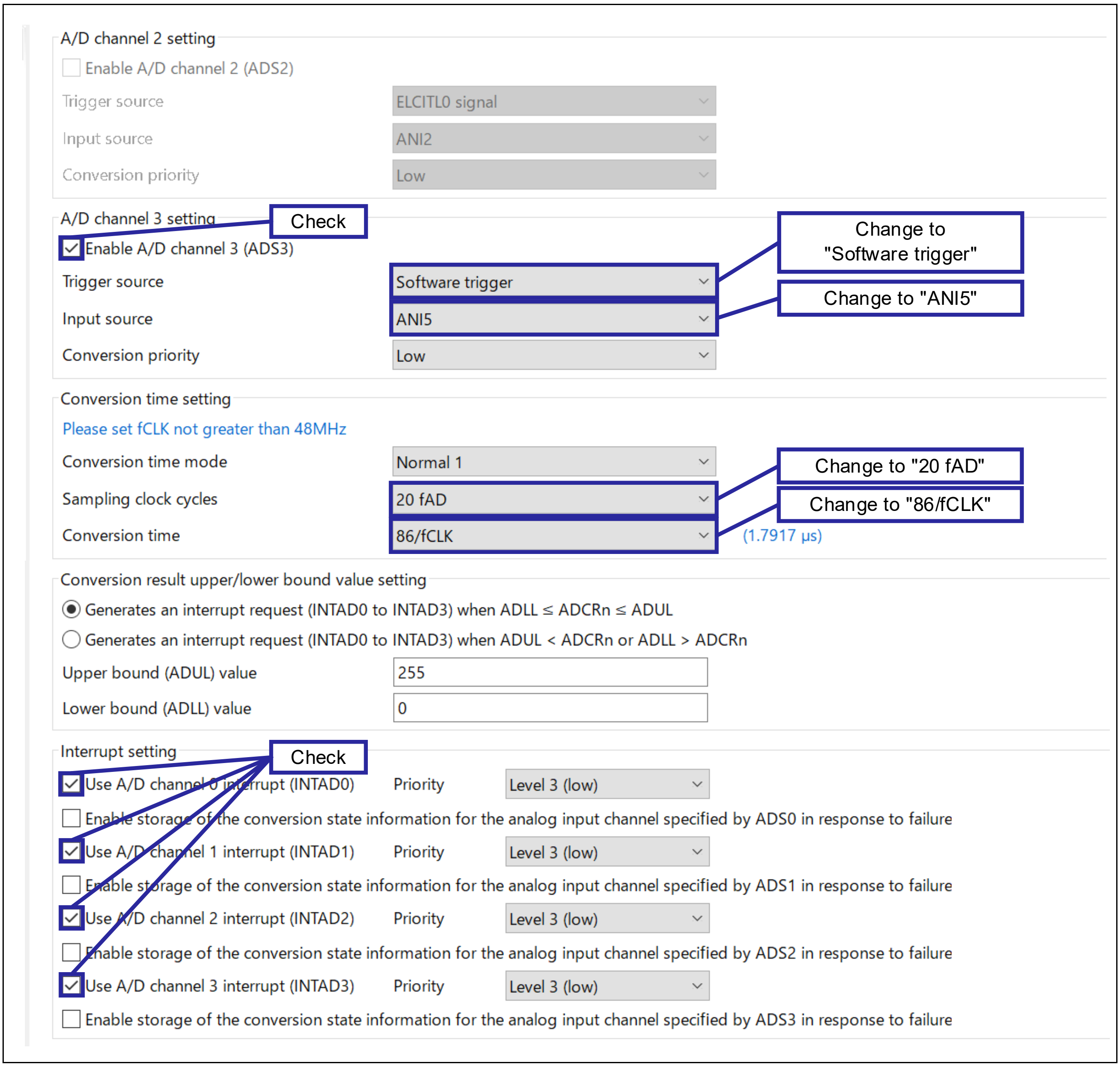
Operation mode setting
One-shot select mode

A/D channel 0 setting
□ Enable A/D channel 0 (ADS0)
Trigger source: INTTM01 signal
Input source: ANI0
Conversion priority: Low

A/D channel 1 setting
□ Enable A/D channel 1 (ADS1)
Trigger source: INTRTC signal
Input source: ANI1
Conversion priority: Low

Check
Check
Check
Change to "Three-channels (ADS0, ADS1, ADS2)"
Change to "Timer RD2 A/D conversion trigger 0"
Change to "ANI4"

Figure 5-16 Configuration of A/D Converter (2/2)



A/D channel 2 setting

□ Enable A/D channel 2 (ADS2)

Trigger source: ELCITL0 signal

Input source: ANI2

Conversion priority: Low

A/D channel 3 setting

☑ Enable A/D channel 3 (ADS3)

Trigger source: Software trigger

Input source: ANI5

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: 86/fCLK (1.7917 µs)

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

☑ 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

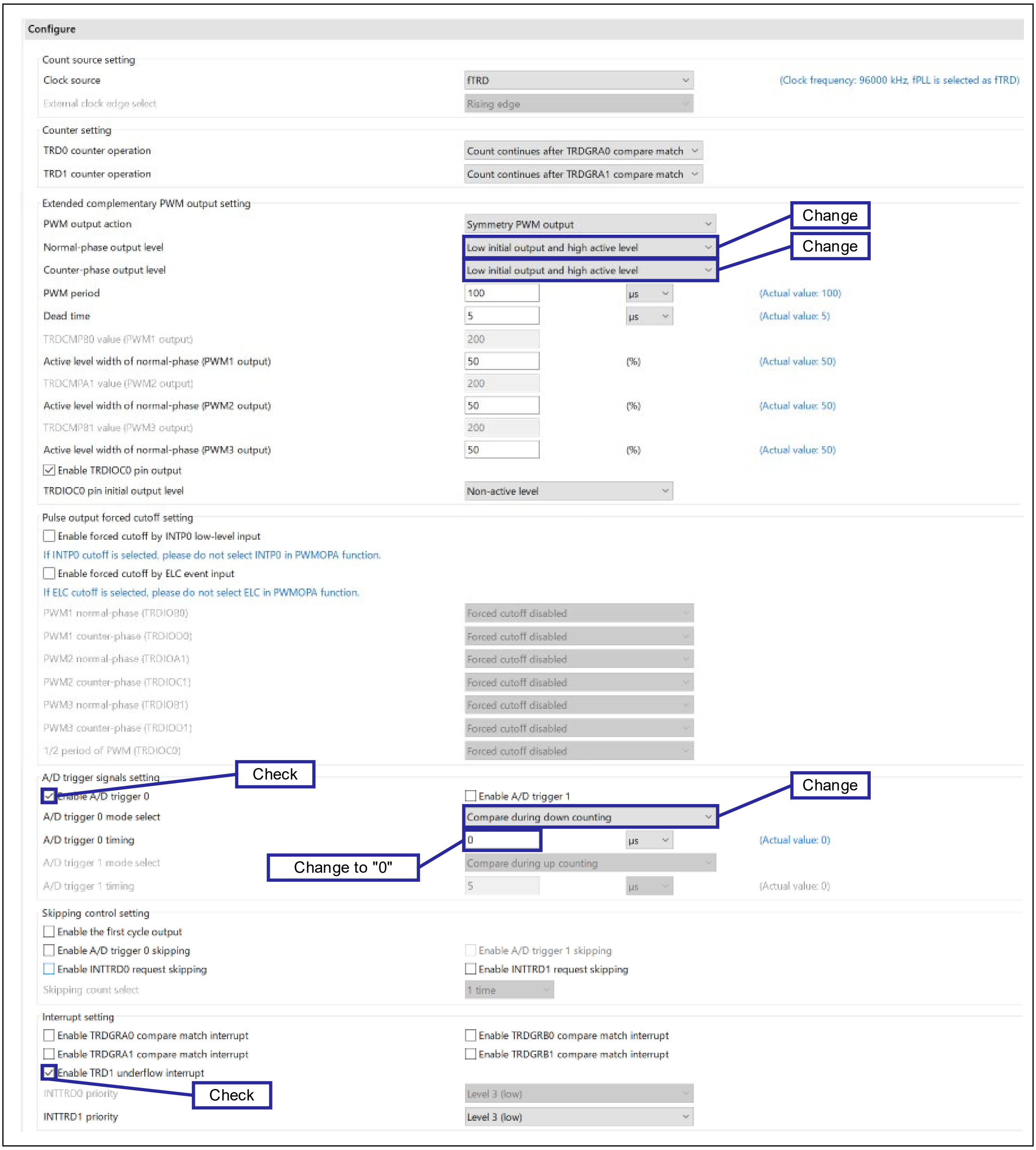
☑ Use A/D channel 3 interrupt (INTAD3) Priority: Level 3 (low)

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

Table 5-10 Component Configurations (Three-phase PWM output)

Item	Content
Component	Three-phase PWM output
Configuration Name	Config_TRD0_TRD1
Resource	TRD0_TRD1
Operation Mode	Extension of complementary PWM mode

Figure 5-17 Configuration of three-phase PWM output



Configure

Count source setting

Clock source	fTRD		(Clock frequency: 96000 kHz, fPLL is selected as fTRD)
External clock edge select	Rising edge		

Counter setting

TRD0 counter operation	Count continues after TRDGRA0 compare match
TRD1 counter operation	Count continues after TRDGRA1 compare match

Extended complementary PWM output setting

PWM output action	Symmetry PWM output		
Normal-phase output level	Low initial output and high active level		
Counter-phase output level	Low initial output and high active level		
PWM period	100	μs	(Actual value: 100)
Dead time	5	μs	(Actual value: 5)
TRDCMPB0 value (PWM1 output)	200		
Active level width of normal-phase (PWM1 output)	50	(%)	(Actual value: 50)
TRDCMPA1 value (PWM2 output)	200		
Active level width of normal-phase (PWM2 output)	50	(%)	(Actual value: 50)
TRDCMPB1 value (PWM3 output)	200		
Active level width of normal-phase (PWM3 output)	50	(%)	(Actual value: 50)

☑ Enable TRDIOC0 pin output

TRDIOC0 pin initial output level	Non-active level

Pulse output forced cutoff setting

☐ Enable forced cutoff by INTP0 low-level input

If INTP0 cutoff is selected, please do not select INTP0 in PWMOPA function.

☐ Enable forced cutoff by ELC event input

If ELC cutoff is selected, please do not select ELC in PWMOPA function.

PWM1 normal-phase (TRDIOB0)	Forced cutoff disabled
PWM1 counter-phase (TRDIOD0)	Forced cutoff disabled
PWM2 normal-phase (TRDIOA1)	Forced cutoff disabled
PWM2 counter-phase (TRDIOC1)	Forced cutoff disabled
PWM3 normal-phase (TRDIOB1)	Forced cutoff disabled
PWM3 counter-phase (TRDIOD1)	Forced cutoff disabled
1/2 period of PWM (TRDIOC0)	Forced cutoff disabled

A/D trigger signals setting

☑ Enable A/D trigger 0 ☐ Enable A/D trigger 1

A/D trigger 0 mode select	Compare during down counting		
A/D trigger 0 timing	0	μs	(Actual value: 0)
A/D trigger 1 mode select	Compare during up counting		
A/D trigger 1 timing	5	μs	(Actual value: 0)

Skipping control setting

☐ Enable the first cycle output

☐ Enable A/D trigger 0 skipping ☐ Enable A/D trigger 1 skipping

☐ Enable INTTRD0 request skipping ☐ Enable INTTRD1 request skipping

Skipping count select	1 time

Interrupt setting

☐ Enable TRDGRA0 compare match interrupt ☐ Enable TRDGRB0 compare match interrupt

☐ Enable TRDGRA1 compare match interrupt ☐ Enable TRDGRB1 compare match interrupt

☑ Enable TRD1 underflow interrupt

INTTRD0 priority	Level 3 (low)
INTTRD1 priority	Level 3 (low)

Change

Change

Check

Change

Change to "0"

Check

5.2.2 Folder Structure

Table 5-11 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 5-11 Folder Structure

Folder/File Name	Description	Generated by Smart Configurator
\ r01an6973_adc_advanced_mode_simultaneous_sampling<DIR>[NOTE 2]	Sample code folder	
\src<DIR>	Program storage folder	
main.c	Sample code source 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	√
\Config_TRD0<DIR>	TRD0 program storage folder	√
Config_TRD0.c	TRD0 source file	√
Config_TRD0.h	TRD0 header file	√
Config_TRD0_user.c	TRD0 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.

Note: 1. This sample code does not use it.

Note: 2. The sample code for IAR contains the r01an6973_adc_advanced_mode_simultaneous_sampling.ipcf file. For details on the .ipcf file, please refer to “RL78 Smart Configurator User’s Guide: IAR” (R20AN0581).

5.2.3 List of Option Byte Settings

Table 5-12 shows the option byte settings.

Table 5-12 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

5.2.4 List of Constants

Constant is not used in the sample code.

5.2.5 List of Variables

Table 5-13 shows the variables used in the sample code.

Table 5-13 Variables used in the sample code

Variable Name	Type	Contents	Function that uses the variable
g_result_buffer0	uint16_t	Storage of A/D conversion results for Channel 0	r_Config_ADC_ad0_interrupt
g_result_buffer1	uint16_t	Storage of A/D conversion results for Channel 1	r_Config_ADC_ad1_interrupt
g_result_buffer2	uint16_t	Storage of A/D conversion results for Channel 2	r_Config_ADC_ad2_interrupt
g_result_buffer3	uint16_t	Storage of A/D conversion results for Channel 3	r_Config_ADC_ad3_interrupt
g_sampling_flg	uint8_t	Flag indicating that simultaneous sampling is in progress	r_Config_TRD0_TRD1_trd1_interrupt r_Config_ADC_ad2_interrupt

5.2.6 Function List

Table 5-14 shows the function used in the sample code. However, functions generated by the Smart Configurator that have not been modified are excluded.

Tablw 5-14 Function List

Function Name	Overview	Source file
main	Main processing	main.c
r_Config_TRD0_TRD1_trd1_interrupt	Timer RD20 interrupt processing	Config_TRD0_TRD1_user.c
r_Config_ADC_ad0_interrupt	A/D converter channel 0 interrupt processing	Config_TRD0_user.c
r_Config_ADC_ad1_interrupt	A/D converter channel 1 interrupt processing	Config_TRD0_user.c
r_Config_ADC_ad2_interrupt	A/D converter channel 2 interrupt processing	Config_TRD0_user.c
r_Config_ADC_ad3_interrupt	A/D converter channel 3 interrupt processing	Config_TRD0_user.c

5.2.7 Function specifications

The following describes the function specifications of the sample code,

[Function name] main

Overview	Main processing
Headers	r_smc_entry.h
Declaration	void main (void);
Description	This function starts operation of the timer RD2 and A/D converter.
Arguments	None
Return values	None
Remarks	None

[Function name] r_Config_TRD0_TRD1_trd1_interrupt

Overview	Timer RD20 interrupt processing
Headers	r_cg_macrodriver.h, r_cg_userdefine.h, Config_TRD0_TRD1.h
Declaration	static void __near r_Config_TRD0_TRD1_trd1_interrupt(void);
Description	This function sets the flag to 1 to indicate that simultaneous sampling is being performed.
Arguments	None
Return values	None
Remarks	None

[Function name] r_Config_ADC_ad0_interrupt

Overview	A/D converter channel 0 interrupt processing
Headers	r_cg_macrodriver.h, r_cg_userdefine.h, Config_ADC.h
Declaration	static void __near r_Config_ADC_ad0_interrupt(void);
Description	When A/D conversion ends, this function reads the A/D conversion result from the ADCR0 register and then stores the result in the variable of the internal RAM.
Arguments	None
Return values	None
Remarks	None

[Function name] r_Config_ADC_ad1_interrupt

Overview	A/D converter channel 1 interrupt processing
Headers	r_cg_macrodriver.h, r_cg_userdefine.h, Config_ADC.h
Declaration	static void __near r_Config_ADC_ad1_interrupt(void);
Description	When A/D conversion ends, this function reads the A/D conversion result from the ADCR1 register and then stores the result in the variable of the internal RAM.
Arguments	None
Return values	None
Remarks	None

[Function name] r_Config_ADC_ad2_interrupt

Overview	A/D converter channel 2 interrupt processing
Headers	r_cg_macrodriver.h, r_cg_userdefine.h, Config_ADC.h
Declaration	static void __near r_Config_ADC_ad2_interrupt(void);
Description	When A/D conversion ends, this function reads the A/D conversion result from the ADCR2 register and then clears the flag that indicates simultaneous sampling is being performed .
Arguments	None
Return values	None
Remarks	None

[Function name] r_Config_ADC_ad3_interrupt

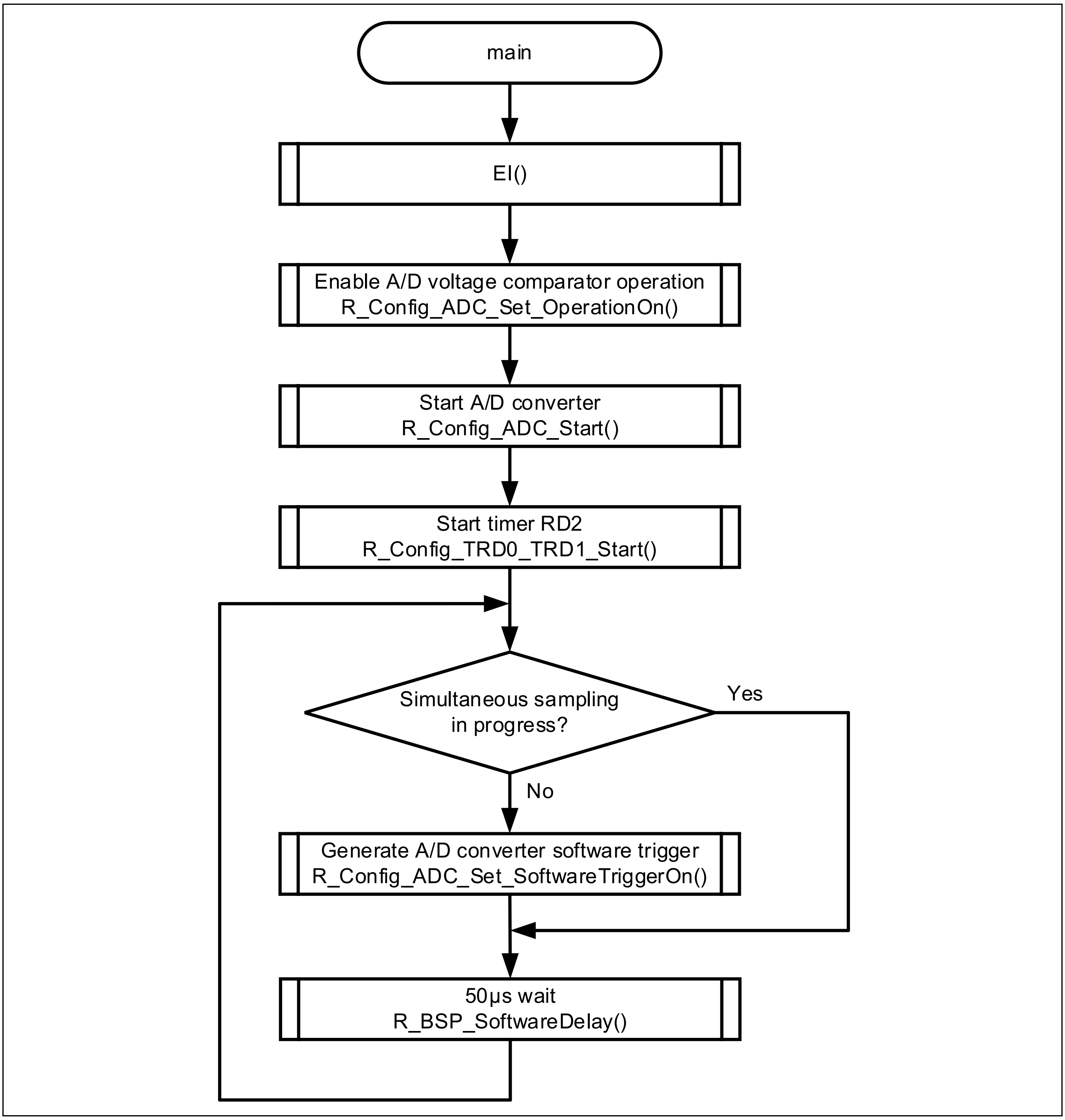
Overview	A/D converter channel 3 interrupt processing
Headers	r_cg_macrodriver.h, r_cg_userdefine.h, Config_ADC.h
Declaration	static void __near r_Config_ADC_ad3_interrupt(void);
Description	When A/D conversion ends, this function reads the A/D conversion result from the ADCR3 register and then stores the result in the variable of the internal RAM.
Arguments	None
Return values	None
Remarks	None

5.2.8 Flowchart

5.2.8.1 Main process

Figure 5-18 shows the flowchart for the main process.

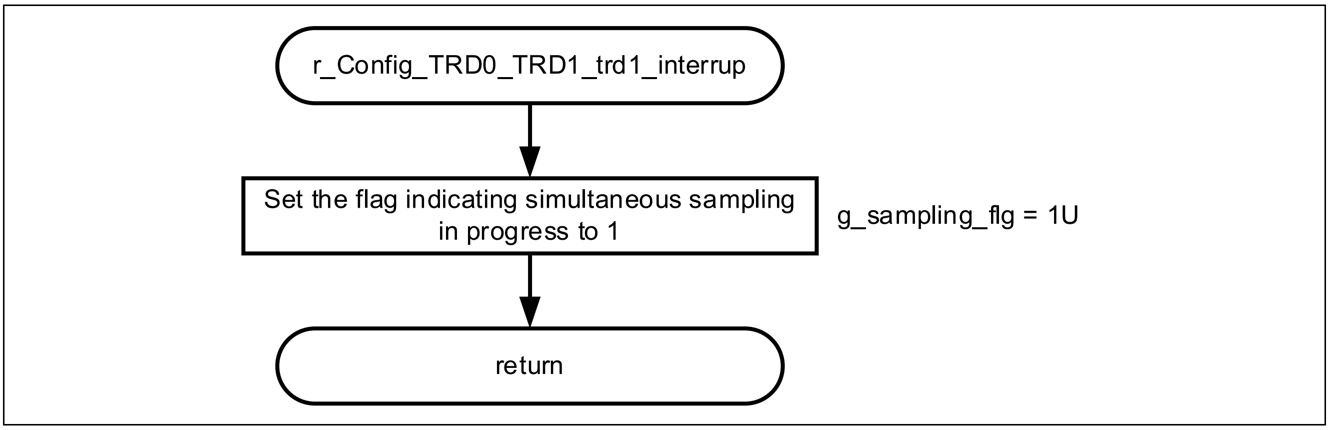
Figure 5-18 Main Process


main
EI()
Enable A/D voltage comparator operation
R_Config_ADC_Set_OperationOn()
Start A/D converter
R_Config_ADC_Start()
Start timer RD2
R_Config_TRD0_TRD1_Start()
Simultaneous sampling
in progress?
Yes
No
Generate A/D converter software trigger
R_Config_ADC_Set_SoftwareTriggerOn()
50µs wait
R_BSP_SoftwareDelay()

5.2.8.2 r_Config_TRD0_TRD1_trd1_interrupt function

Figure 5-19 shows the flowchart of r_Config_TRD0_TRD1_trd1_interrupt function.

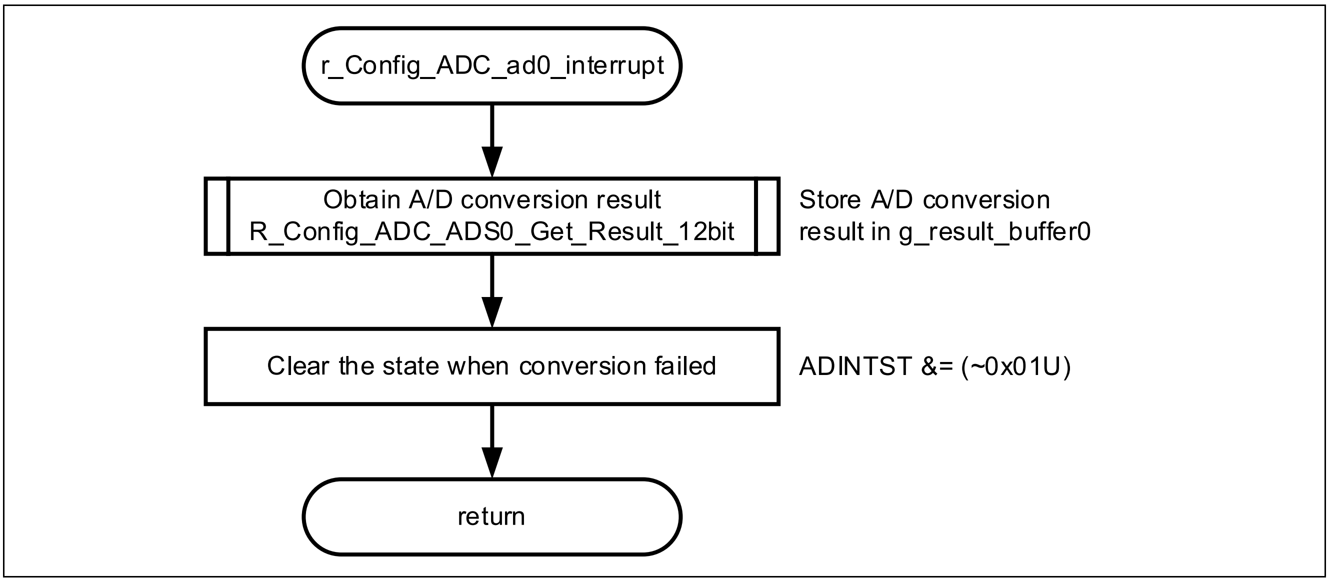
Figure 5-19 r_Config_TRD0_TRD1_trd1_interrup function


r_Config_TRD0_TRD1_trd1_interrup
Set the flag indicating simultaneous sampling in progress to 1
g_sampling_flg = 1U
return

5.2.8.3 r_Config_ADC_ad0_interrupt function

Figure 5-20 shows the flowchart of r_Config_ADC_ad0_interrupt function.

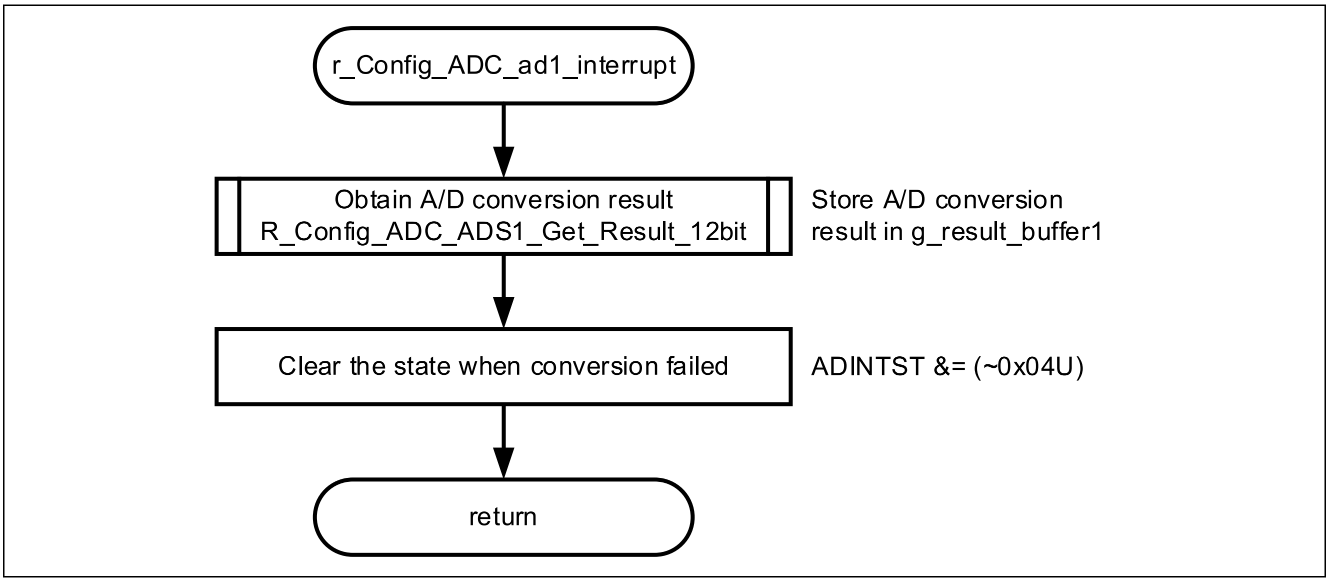
Figure 5-20 r_Config_ADC_ad0_interrupt function


r_Config_ADC_ad0_interrupt
Obtain A/D conversion result
R_Config_ADC_ADS0_Get_Result_12bit
Store A/D conversion result in g_result_buffer0
Clear the state when conversion failed
ADINTST &= (~0x01U)
return

5.2.8.4 r_Config_ADC_ad1_interrupt function

Figure 5-21 shows the flowchart of r_Config_ADC_ad1_interrupt function.

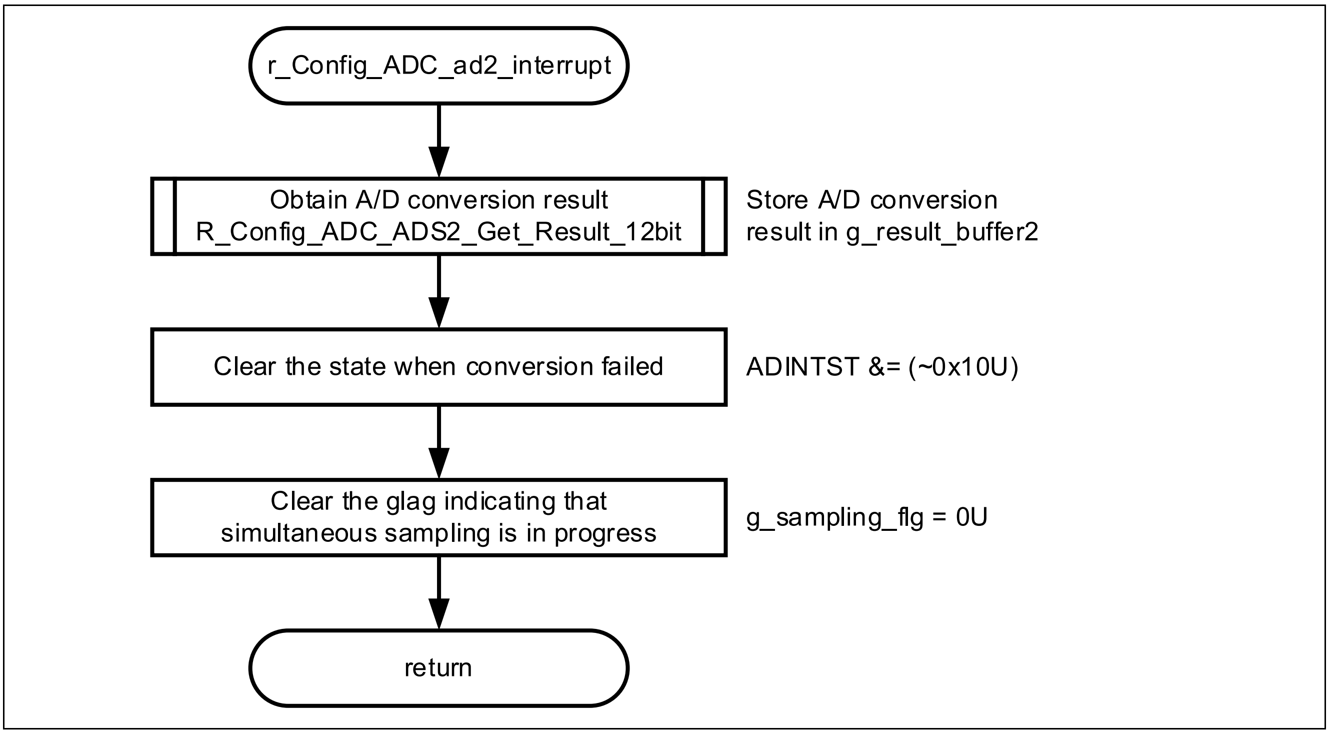
Figure 5-21 r_Config_ADC_ad1_interrupt function


r_Config_ADC_ad1_interrupt
Obtain A/D conversion result
R_Config_ADC_ADS1_Get_Result_12bit
Store A/D conversion result in g_result_buffer1
Clear the state when conversion failed
ADINTST &= (~0x04U)
return

5.2.8.5 r_Config_ADC_ad2_interrupt function

Figure 5-22 shows the flowchart of r_Config_ADC_ad2_interrupt function.

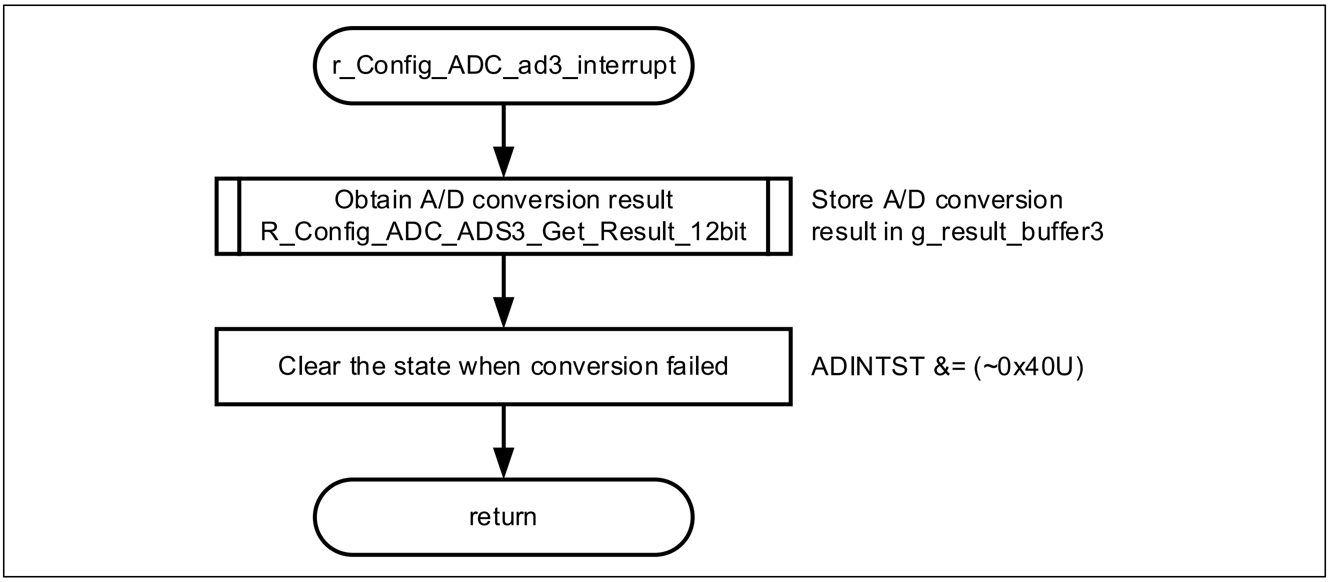
Figure 5-22 r_Config_ADC_ad2_interrupt function


r_Config_ADC_ad2_interrupt
Obtain A/D conversion result
R_Config_ADC_ADS2_Get_Result_12bit
Store A/D conversion result in g_result_buffer2
Clear the state when conversion failed
ADINTST &= (~0x10U)
Clear the glag indicating that simultaneous sampling is in progress
g_sampling_flg = 0U
return

5.2.8.6 r_Config_ADC_ad3_interrupt function

Figure 5-23 shows the flowchart of r_Config_ADC_ad3_interrupt function.

Figure 5-23 r_Config_ADC_ad3_interrupt function


r_Config_ADC_ad3_interrupt
Obtain A/D conversion result
R_Config_ADC_ADS3_Get_Result_12bit
Store A/D conversion
result in g_result_buffer3
Clear the state when conversion failed
ADINTST &= (~0x40U)
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)

RL78/G24 A/D Converter (Advanced Mode Disabled) (R01AN6992)

(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	2024.11.25	-	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/.