# RL78/G24

## A/D Converter (Advanced Mode Disabled)

### Introduction

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

For details on the operation with the A/D converter (advanced mode enabled), see the **Application Note for RL78/G24 A/D Converter (Advanced Mode Enabled) (R01AN6973)**.

### 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. Usage Examples and Operation

## 1.1 Usage Example 1: Software Trigger No-Wait Mode (Internal Reference Voltage)

### 1.1.1 Overview of specifications

In usage example 1, select mode is used and the internal reference voltage and ANI2 are converted alternately, and then the conversion results are stored in the variable of the internal RAM.

Table 1-1 shows an example of channel settings.

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

| | Advanced Mode | Wait Mode | Conversion Mode |
|---|---|---|---|
| Setting | OFF | No-wait mode | Select mode<br>One-shot conversion mode |

| | Channel Setting Conditions | | |
|---|---|---|---|
| | Trigger mode<br>ADM1.ADTMDn[1:0] | Analog input channel<br>ADS.ADISS, ADSn[4:0] | A/D conversion result register |
| Setting | Software trigger no-wait mode<br>00B | Internal reference voltage selected<br>100001B<br><br>ANI2 selected<br>000010B | ADCR |

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

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

| Peripheral Function | Usage |
|---|---|
| A/D converter<br>(Advanced mode disabled) | Performs A/D conversion of internal reference voltage and analog input voltage of the P22/ANI2 pin |

### 1.1.2 Operation

The following shows the A/D converter settings.

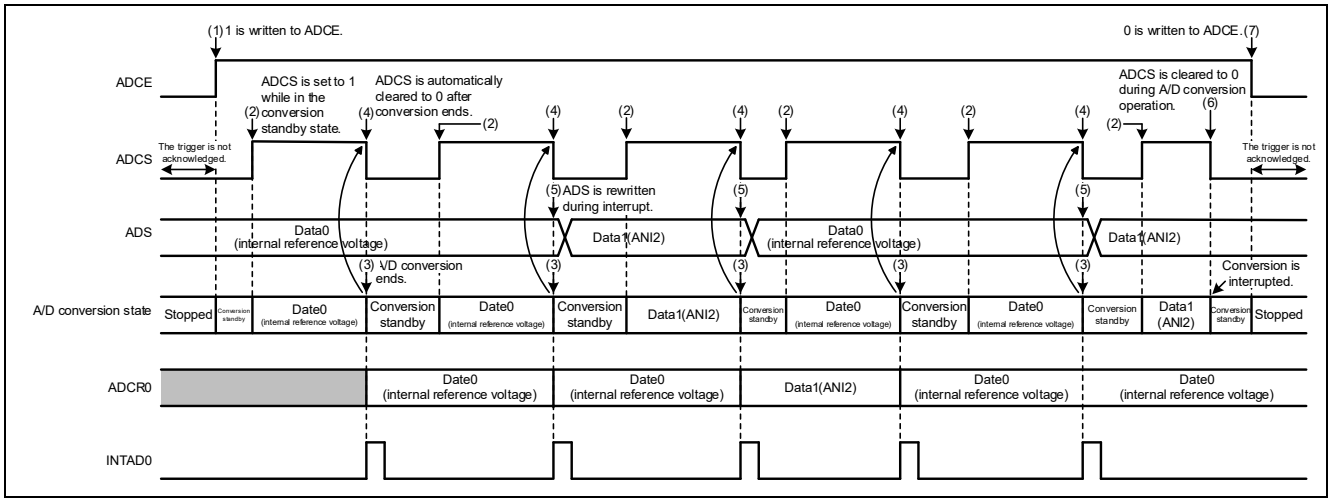
<Settings>

- Normal 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.
- The trigger mode is set to software trigger no-wait mode.
- The operating mode is set to one-shot select mode, and the A/D channel selection is set to internal reference voltage.
- The operating voltage mode is set to normal mode 2, and the conversion time is set to $219/f_{CLK}$.
- The upper limit of A/D conversion result comparison (ADUL) is set to 255, and the lower limit (ADLL) is set to 0. Note:
- A/D conversion end interrupt (INTAD0) is 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 1-1 shows the operation of A/D converter.

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




(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.
(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 perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).
(3) When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR and ADCR0), and the A/D conversion end interrupt request signal (INTAD0) is generated.
(4) After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the standby state.
(5) The register will be modified by an interruption. ADS will be switched when the second interrupt triggered by the internal reference voltage, or the first interrupt triggered by ANI2 occurs.
(6) When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the standby state.
(7) When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby state.

Remarks: The result of the first conversion after the internal reference voltage is set in the ADS register cannot be used. Therefore, the sample program stores the result of the second conversion in the variable of the internal RAM. For the detailed setting flow, see **20.7.6 Settings when temperature sensor output voltage or internal reference voltage is selected (example for software trigger no-wait mode and one-shot conversion** in the **RL78/G24 User's Manual: Hardware**.

## 1.2 Usage Example 2: Hardware Trigger No-Wait Mode (Scan Mode)

### 1.2.1 Overview of specifications

In usage example 2, scan mode is used, and A/D conversion is performed for analog voltages input to the P22/ANI2, P23/ANI3, P24/ANI4, and P25/ANI5 pins by using hardware trigger (realtime clock interrupt by realtime clock: Once per 0.5 second). After A/D conversion ends, the conversion results are stored in the variable of the internal RAM.

Table 1-3 shows an example of channel settings.

Table 1-3 Example of Channel Settings (Usage Example 2)

| | Advanced Mode | Wait Mode | Conversion Mode |
|---|---|---|---|
| Setting | OFF | No-wait mode | Scan mode<br>One-shot conversion mode |

| | Channel Setting Conditions | | | |
|---|---|---|---|---|
| | Trigger mode<br>ADM1.ADTMDn[1:0] | Hardware trigger signal<br><br>ADM1.ADTRSn[2:0] | Analog input channel<br>ADS.ADISS, ADSn[4:0] | A/D conversion result register |
| Setting | Hardware trigger no-wait mode<br>10B | INTRTC signal<br>010B | ANI2-ANI5<br>000100B | ADCR0<br>ADCR1<br>ADCR2<br>ADCR3 |

Table 1-4 shows the peripheral functions and their usage.

Table 1-4 Peripheral Functions and Their Usage (Example 2)

| Peripheral Function | Usage |
|---|---|
| A/D converter<br>(Advanced mode disabled) | Performs A/D conversion of analog input voltage of the P22/ANI2, P23/ANI3, P24/ANI4, and P25/ANI5 pins |
| Realtime Clock (RTC) | Uses a realtime clock interrupt signal (INTRTC) as a hardware trigger |

### 1.2.2 Operation

The following shows the A/D converter settings.

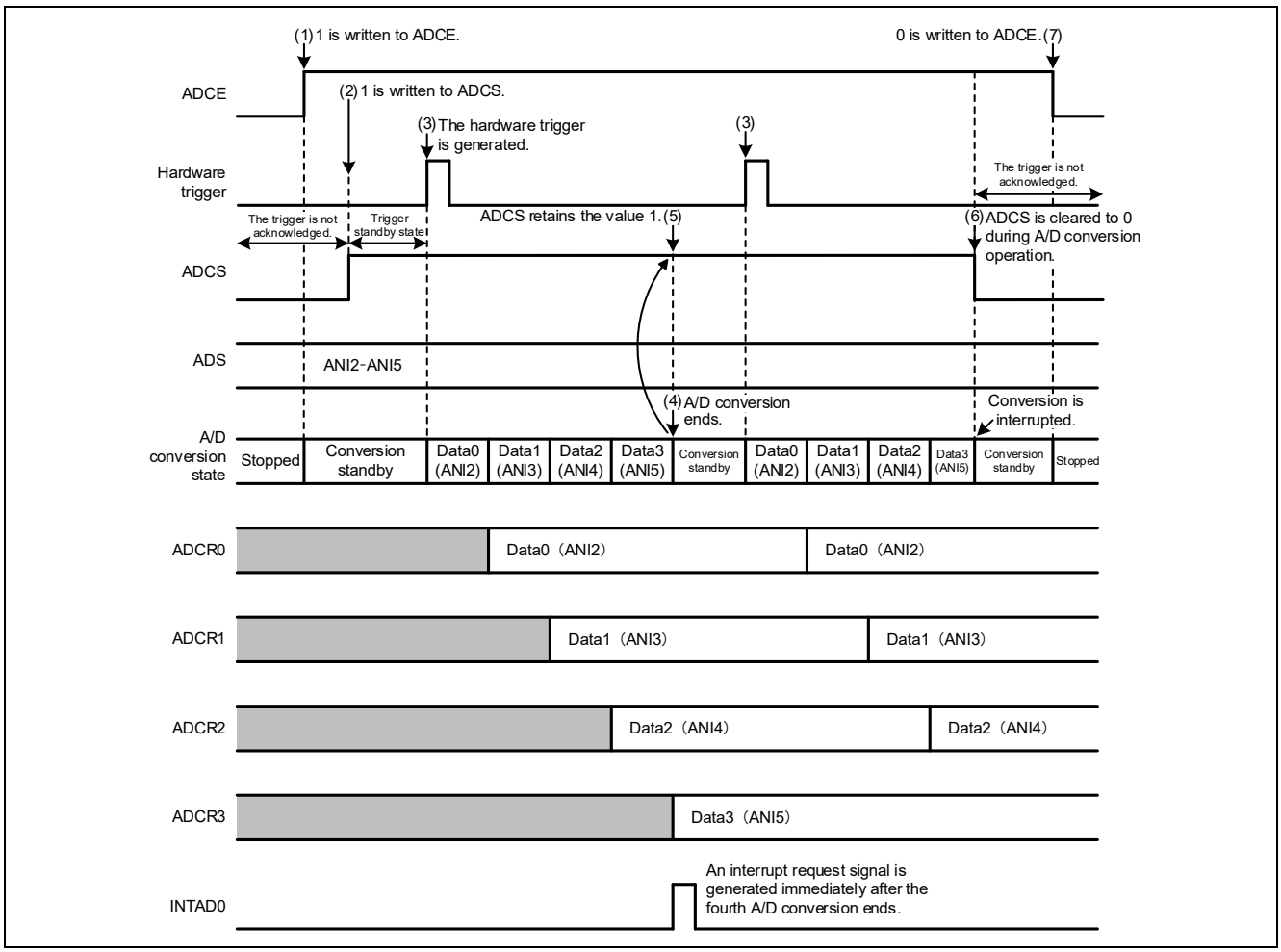
<Settings>

- Normal 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.
- The trigger mode is set to hardware trigger no-wait mode with INTRTC (realtime clock interrupt signal at 0.5 second intervals).
- The operating mode is set to one-shot select mode, and the A/D channel selection is set to ANI2-ANI5.
- The operating voltage mode is set to normal mode 1, and the conversion time is set to 258/$f_{CLK}$.
- The upper limit of A/D conversion result comparison (ADUL) is set to 255, and the lower limit (ADLL) is set to 0. Note:
- A/D conversion end interrupt (INTAD0) is 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 1-2 shows the operation of A/D converter.

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




(1) Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.

(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 hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, A/D conversion does not start even if ADCS is set to 1.

(3) If a hardware trigger INTRTC is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

(4) A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result registers (ADCRn) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD0) is generated immediately after A/D conversion of the four channels ends.

(5) After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the A/D converter enters the standby state.

(6) When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the standby state. However, the A/D converter does not stop in this state.

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

When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

## 2. Operation Confirmation Conditions

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

Table 2-1 Operation Confirmation Conditions

| Item | Description |
|---|---|
| MCU used | RL78/G24 (R7F101GLG) |
| Operating frequency | Advanced mode enabled<br>• High-Speed On-Chip Oscillator ($f_{HOCO}$): 8MHz<br>• PLL Oscillator Circuit Output ($f_{PLL}$): 96MHz<br>• CPU/Peripheral Hardware Clock ($f_{CLK}$): 24MHz |
| Operating voltage | • 3.3V (Can operate between 2.7V - 5.5V)<br>• LVD0 Operation ($V_{LVD0}$): Reset Mode<br>Rising edge TYP. 2.97V<br>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<br>IAR: E2 Emulator Lite |
| Board used | RL78/G24 Fast Prototyping Board (RTK7RLG240C00000BJ) |

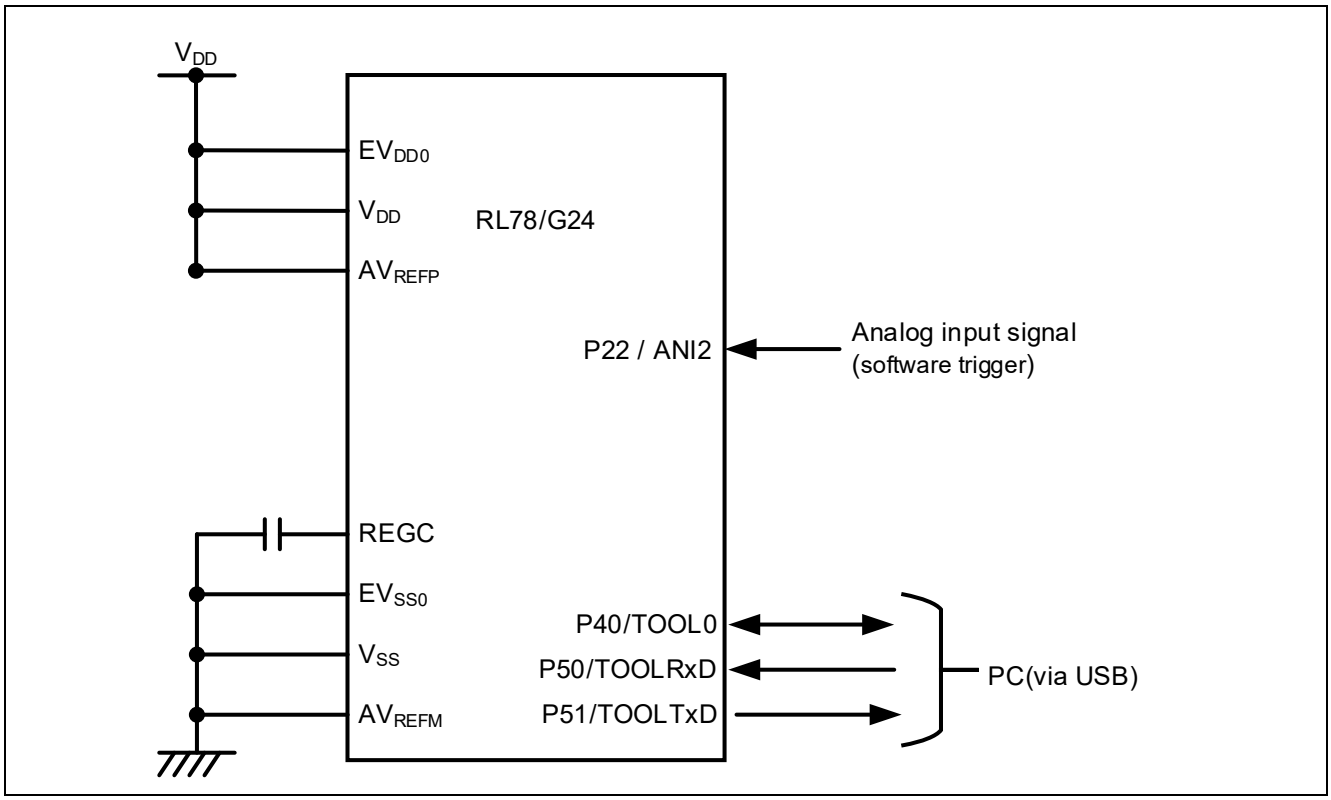
# 3. Hardware Description

## 3.1 Example 1 : Software Trigger No-wait Mode (Internal reference Voltage)

### 3.1.1 Example pf Hardware Configuration

Figure 3-1 shows an example of the hardware configuration for Example 1.

Figure 3-1 Example of Hardware Configuration




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 characteristics 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.

### 3.1.2 List of used Pins

Table 3-1 shows the pins used and their functions.

Table 3-1 Pins Used and Their Functions

| Pin Name | I/O | Functions |
|---|---|---|
| P22 / ANI2 | Input | A/D converter Analog input |

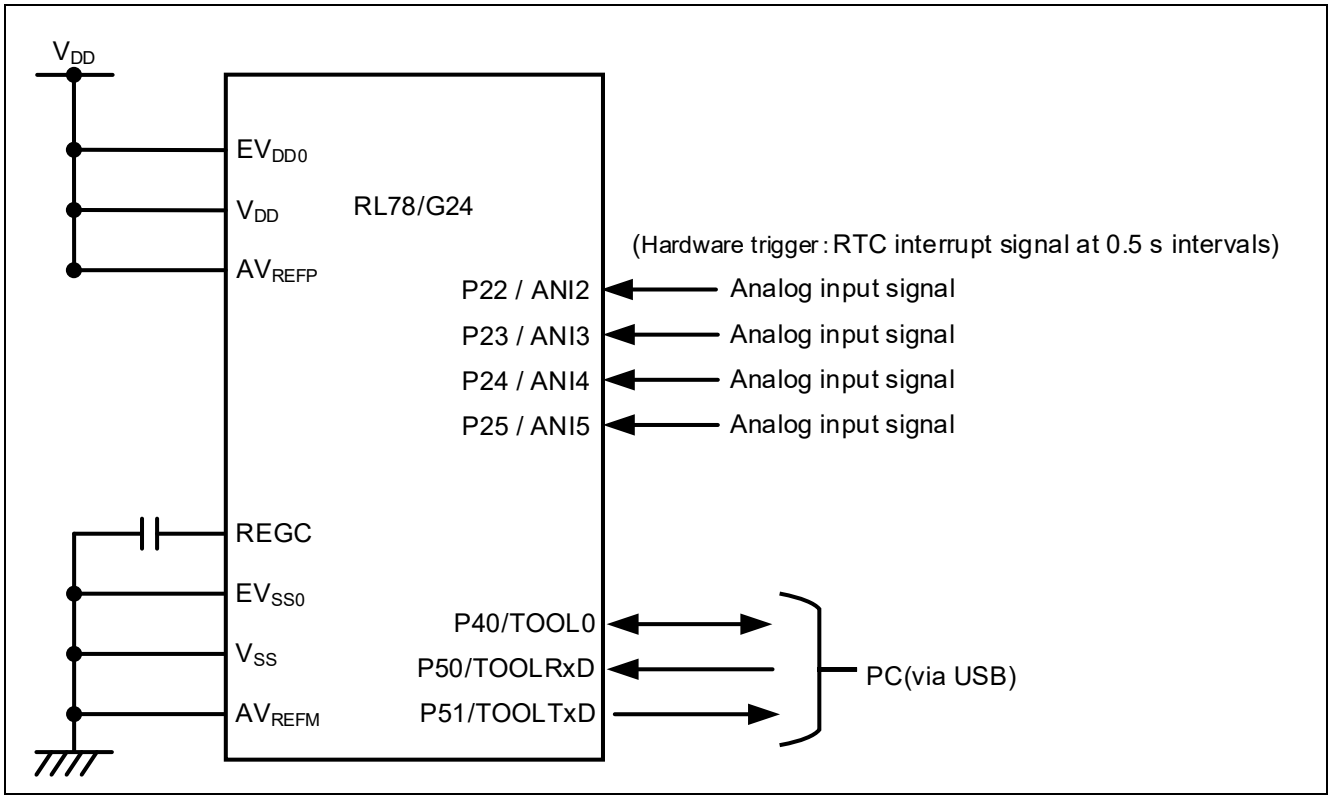
Caution: In this application note, only the used pins are processed. When designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.

## 3.2 Example 2 : Hardware Trigger No-wait Mode (Scan mode)

### 3.2.1 Example of Hardware Configuration

Figure 3-2 shows an example of the hardware configuration for Example 2.

Figure 3-2 Example of Hardware Configuration




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.

### 3.2.2 List of used Pins

Table 3-2 shows the pins used and their functions.

Table 3-2 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.

# 4. Software Description

## 4.1 Usage Example 1: Software Trigger No-Wait Mode (Internal Reference Voltage)

### 4.1.1 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.

#### 4.1.1.1 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 4-1 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 (R20UT5091)**.

Figure 4-1 System Configuration (e$^2$ studio, CS+)

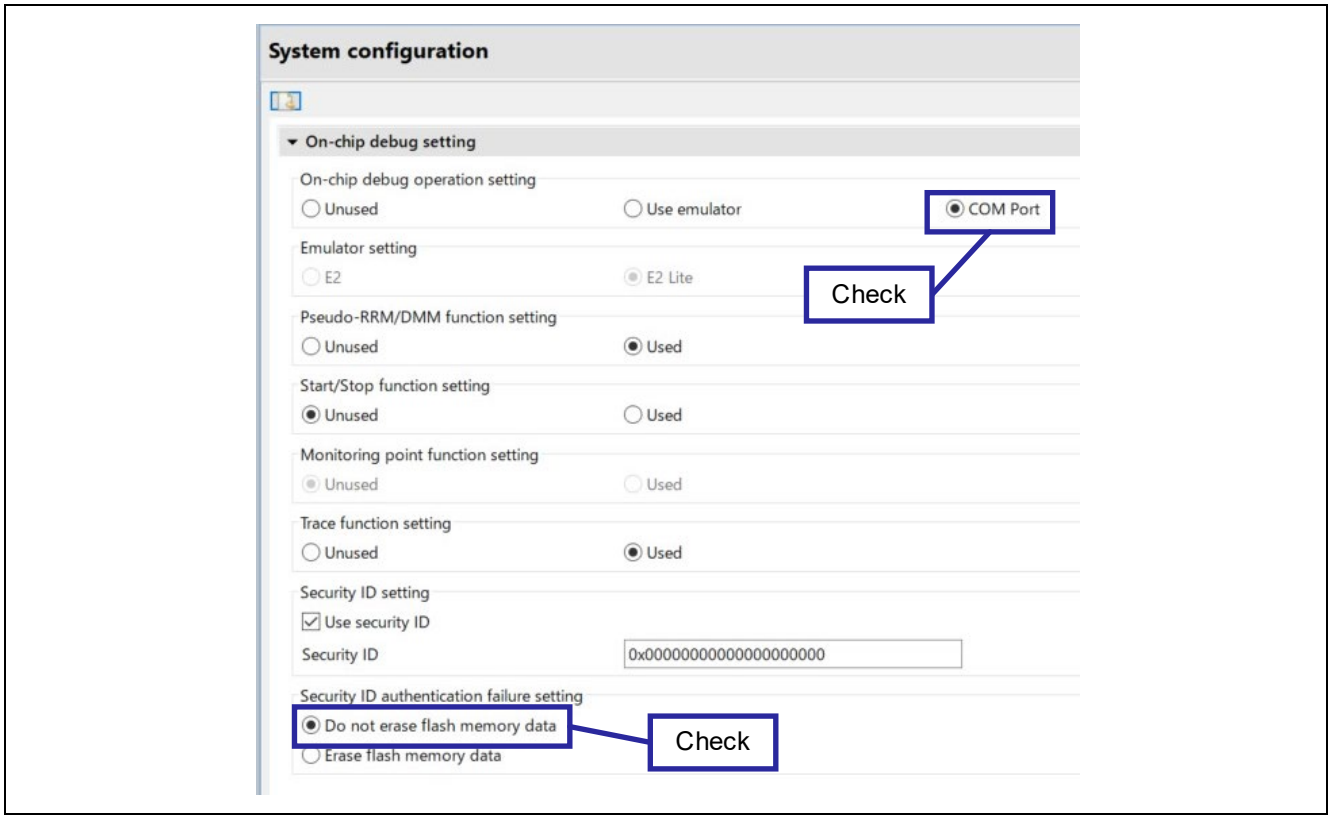
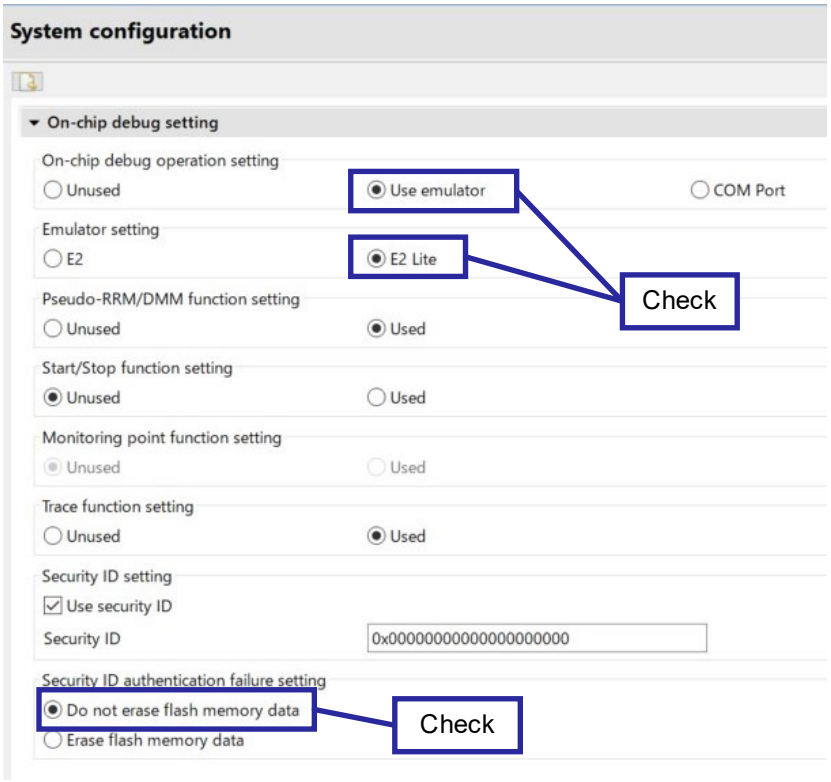


Figure 4-2 shows the system configurations used in this sample program for IAR.

Figure 4-2 System Configurations (IAR)



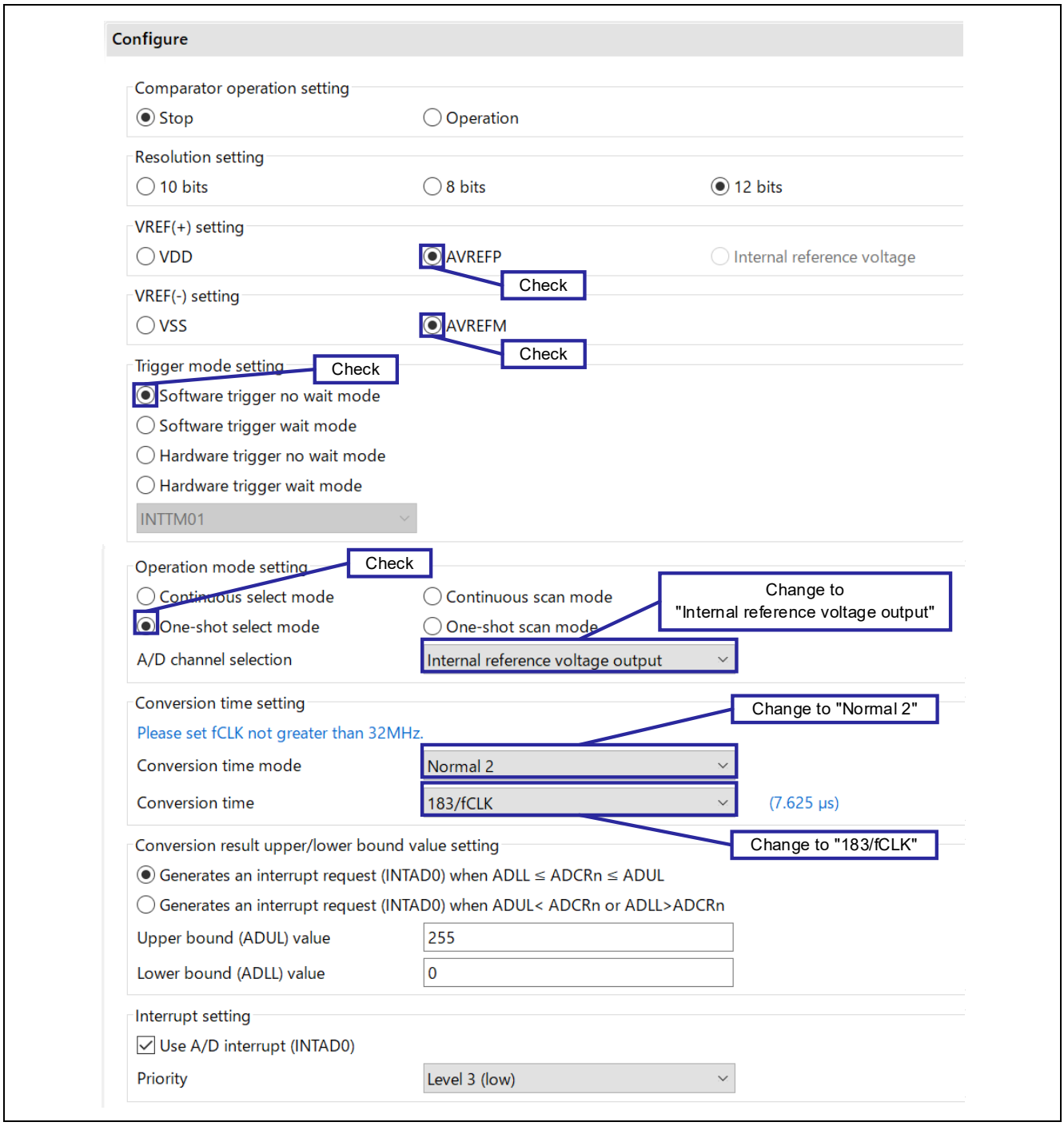
### 4.1.1.2 Component Configurations

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

Table 4-1 Component Configurations (A/D Converter)

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

Figure 4-3 Configuration of A/D Converter



### 4.1.2 Folder Structure

Table 4-2 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 4-2 Folder Structure

| Folder/File Name | Description | Generated by Smart Configurator |
|---|---|---|
| \r01an6992_adc_nomal_mode_internal_reference_voltage<DIR>NOTE 1 | 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 | √ |
| ¥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. The sample code for IAR contains the r01an6992_adc_nomal_mode_internal_reference_voltage.ipcf file. For details on the .ipcf file, please refer to “RL78 Smart Configurator User’s Guide: IAR” (R20AN0581).

### 4.1.3 List of Option Byte Settings

Table 4-3 shows the option byte settings.

Table 4-3 Option Byte Settings

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

### 4.1.4 List of Constant

Constant is not used in the sample code.

### 4.1.5 List of Variables

Table 4-4 shows the list of variables used in the sample code.

Table 4-4 Variables used in the sample code

| Variable name | Type | Contents | Function that uses the variable |
|---|---|---|---|
| g_result_buffer | uint16_t | Storage of A/D conversion results | r_Config_ADC_interrupt |
| g_adc_flg | uint16_t | A/D conversion execution flag | r_Config_ADC_interrupt |

### 4.1.6 Function list

Table 4-5 shows the functions used in the sample code. However, functions generated by the Smart Configurator that have not been modified are excluded.

Table 4-5 Function List

| Function Name | Overview | Source file |
|---|---|---|
| main | Main processing | main.c |
| R_Config_ADC_Create_UserInit | P22/ANI2 initial settings | Config_TRD0_user.c |
| r_Config_ADC_interrupt | A/D converter interrupt processing | Config_TRD0_user.c |

### 4.1.7 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 A/D converter. |
| Arguments | None |
| Return values | None |
| Remarks | None |

[Function name] R_Config_ADC_Create_UserInit

| | |
|---|---|
| Overview | P22/ANI2 initial settings |
| Headers | r_cg_macrodriver.h, r_cg_userdefine.h, Config_ADC.h |
| Declaration | static void __near r_Config_ADC_ad0_interrupt(void); |
| Description | This function specifies the P22/ANI2 initial settings. |
| Arguments | None |
| Return values | None |
| Remarks | None |

[Function name] r_Config_ADC_interrupt

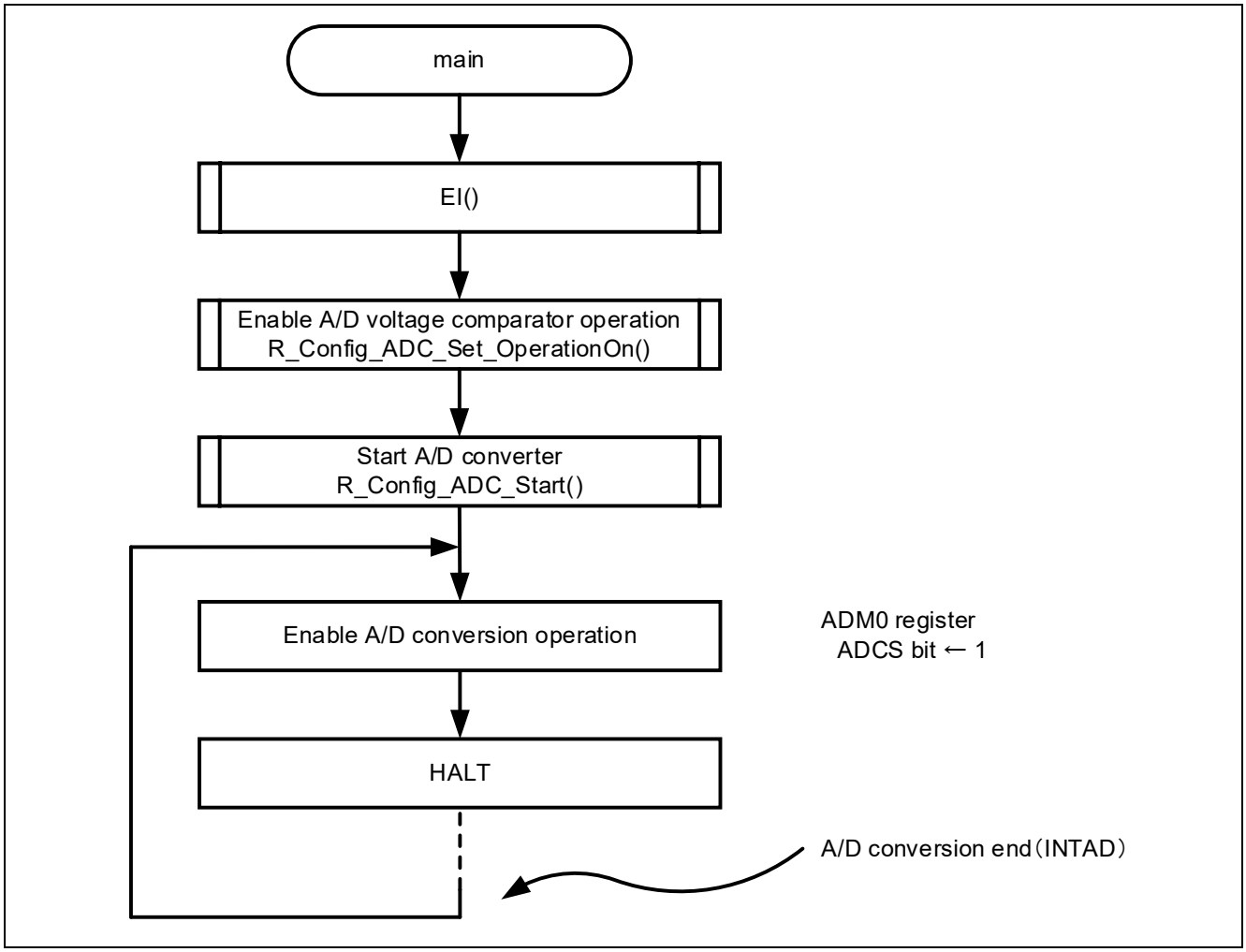
| | |
|---|---|
| Overview | A/D converter interrupt processing |
| Headers | r_cg_macrodriver.h, r_cg_userdefine.h, Config_ADC.h |
| Declaration | static void __near r_Config_ADC_ad0_interrupt(void); |
| Description | When g_adc_flg is set to 1, this function reads the A/D conversion result from the ADCR register and then stores the result in the variable of the internal RAM. It also switches the internal reference voltage of the input channel and ANI2 alternately. |
| Arguments | None |
| Return values | None |
| Remarks | None |

### 4.1.8 Flowchart

#### 4.1.8.1 Main Process

Figure 4-4 shows the flowchart for the main process.

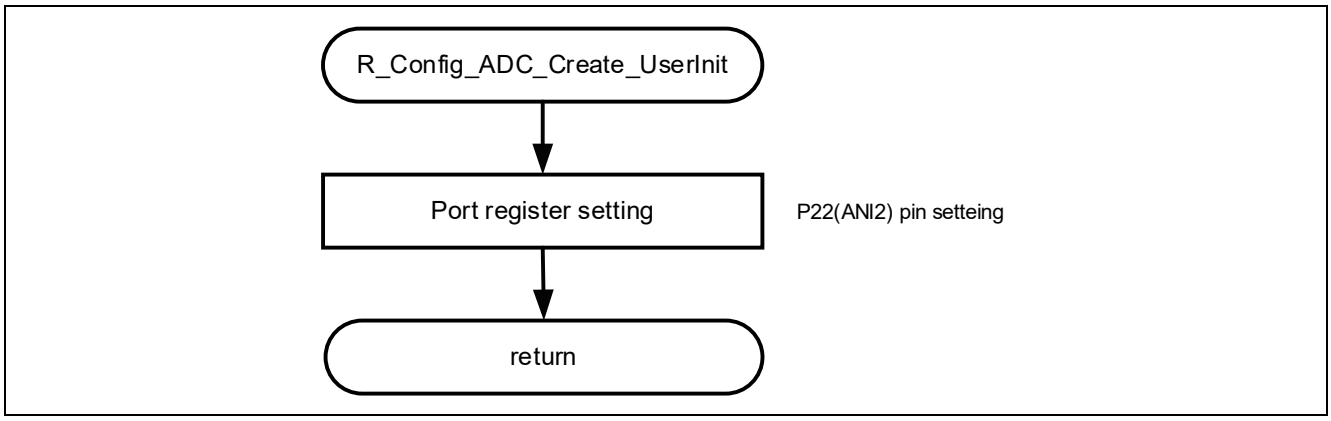
Figure 4-4 Main Process




#### 4.1.8.2 R_Config_ADC_Create_UserInit function

Figure 4-5 shows the flowchart of R_Config_ADC_Create_UserInit function.

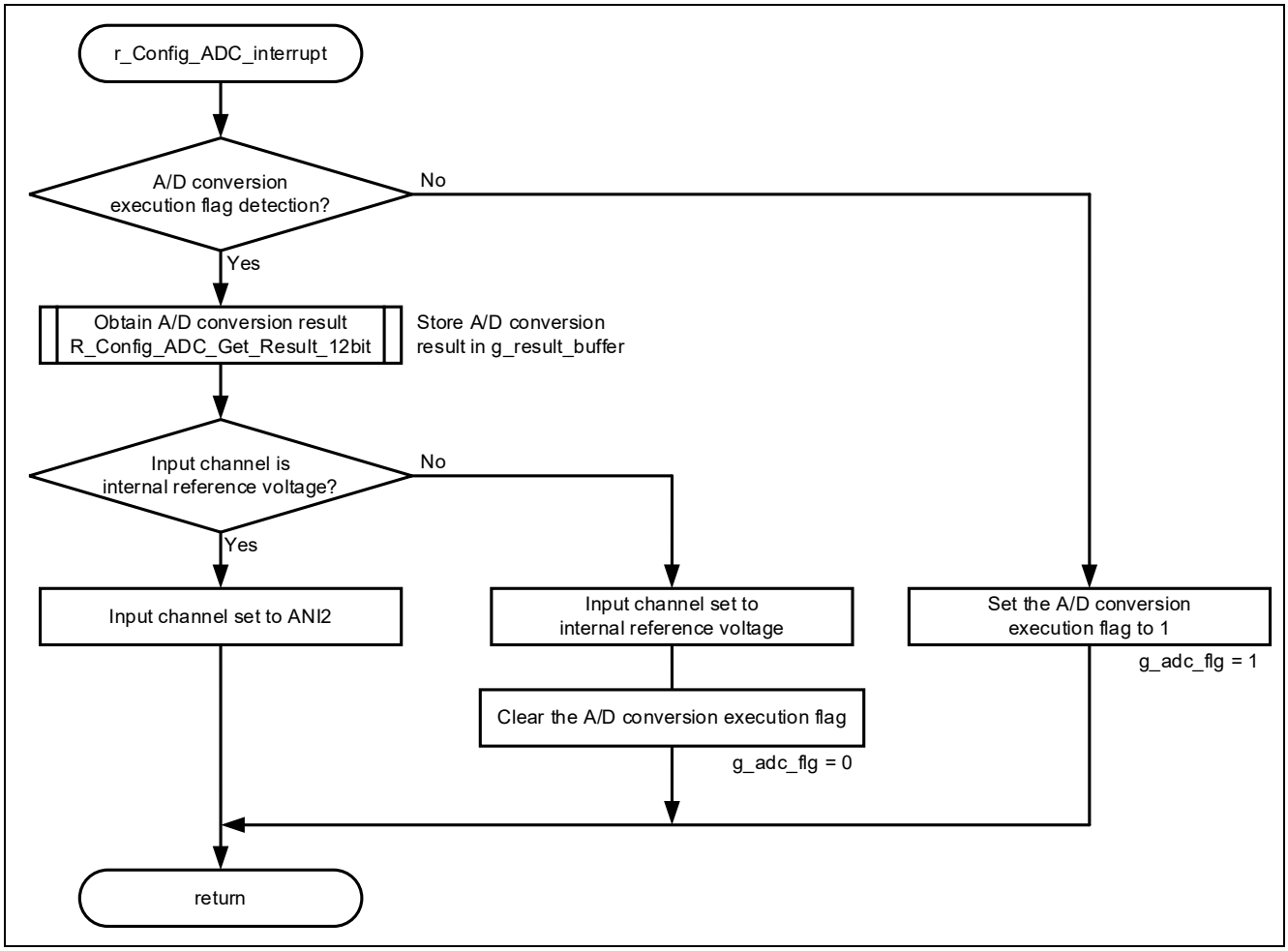
Fugire 4-5 R_Config_ADC_Create_UserInit function



#### 4.1.8.3 r_Config_ADC_interrupt function

Figure 4-6 shows the flowchart of r_Config_ADC_interrupt function.

Figure 4-6 r_Config_ADC_interrupt function




Note The initial value of the A/D conversion execution flag (g_adc_flg) is 0.

## 4.2 Usage Example 2: Hardware Trigger No-Wait Mode (Scan Mode)

### 4.2.1 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.

#### 4.2.1.1 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 e$^2$ studio and CS+ but are different for IAR. Specify appropriate settings according to your environment.

Figure 4-7 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** and **7.2 Using COM Port Debugging with CS+** in the **RL78/G24 Fast Prototyping Board User's Manual (R20UT5091)**.

Figure 4-7 System Configuration (e$^2$ studio, CS+)

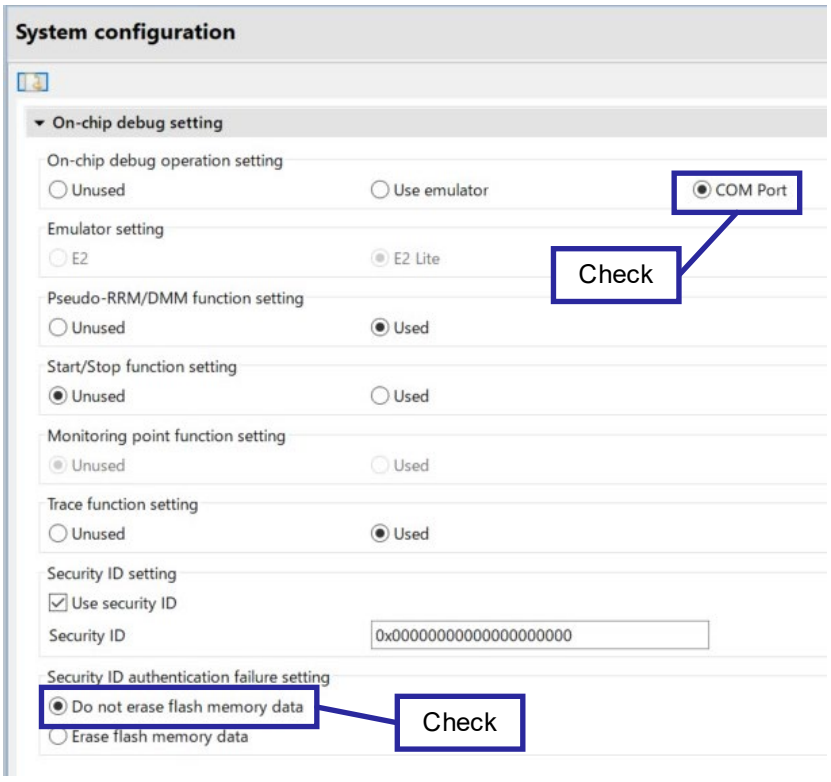
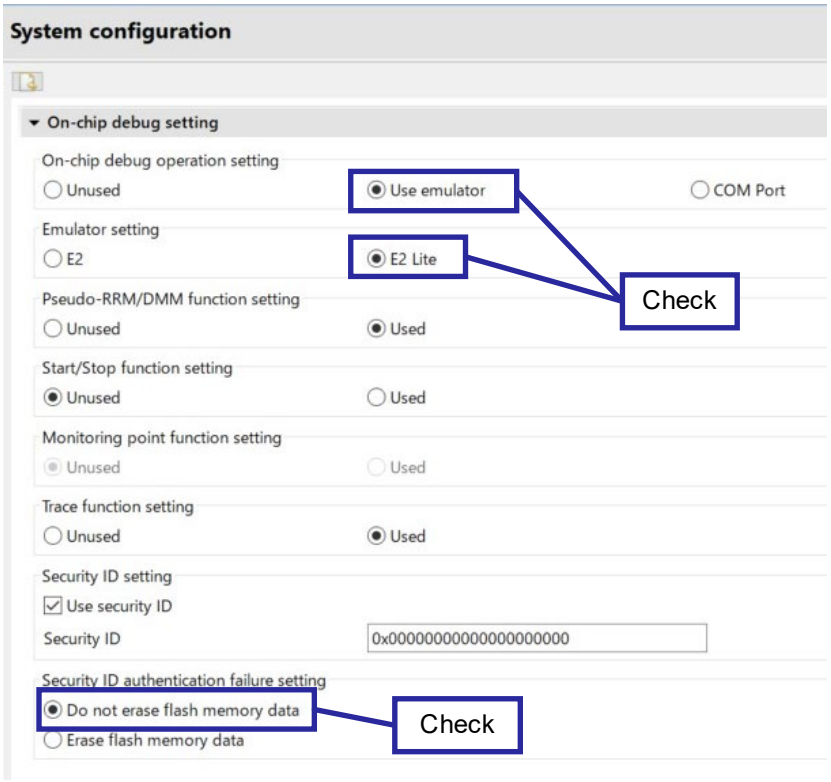


Figure 4-8 shows the system configurations used in this sample program for IAR.

Figure 4-8 System Configurations (IAR)



### 4.2.1.2 Component Configurations

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

Table 4-6 Component Configurations (A/D Converter)

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

Figure 4-9 Configuration of A/D Converter

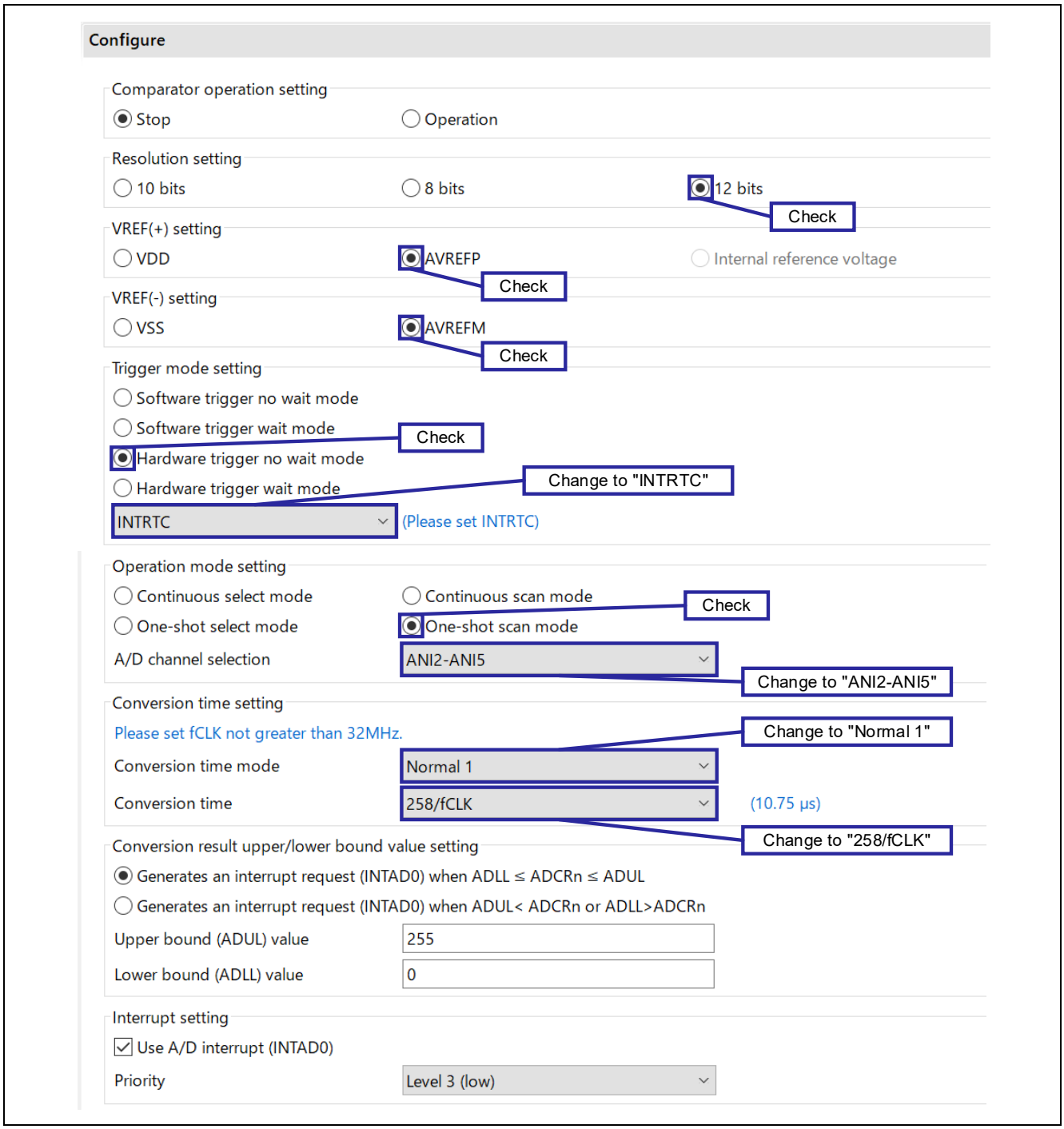
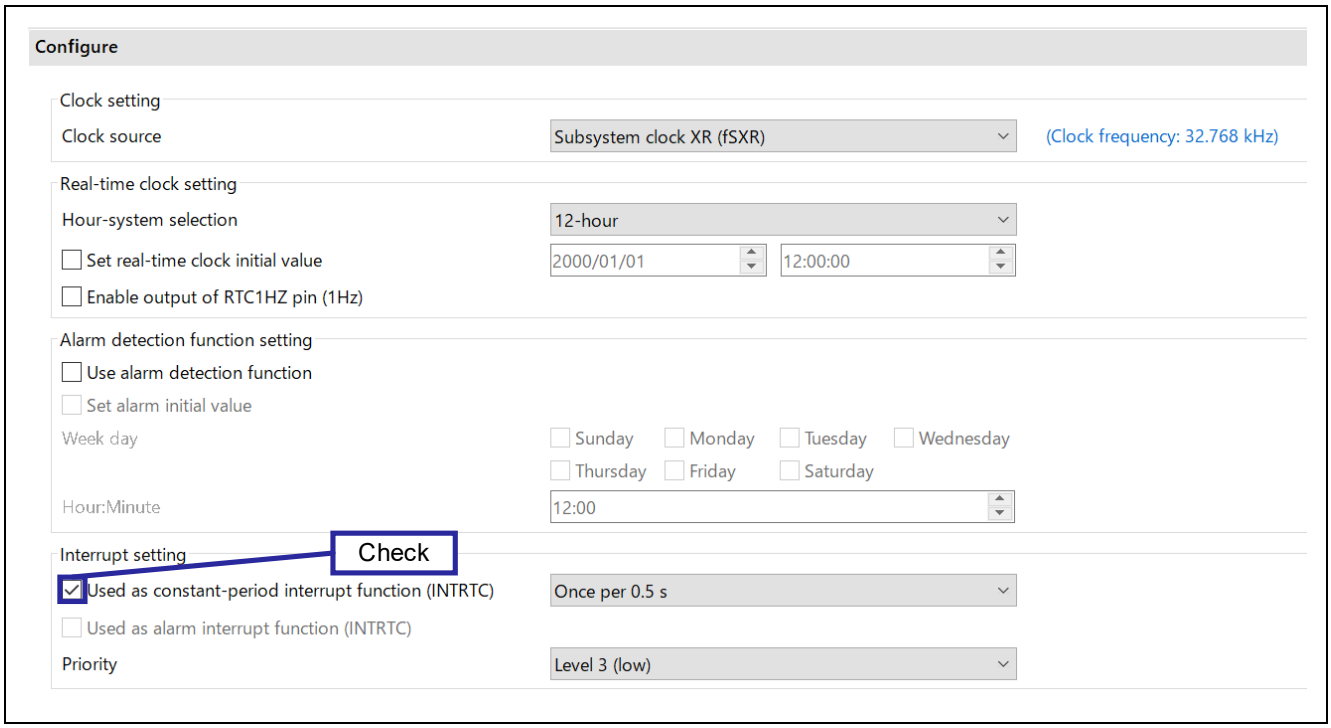


Table 4-7 Component Configurations (Realtime Clock)

| Item | Content |
|---|---|
| Component | Realtime clock |
| Configuration Name | Config_RTC |
| Resource | RTC |

Figure 4-10 Configuration of Realtime Clock



### 4.2.2 Folder Structure

Table 4-8 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 4-8 Folder Structure

| Folder/File Name | Description | Generated by Smart Configurator |
|---|---|---|
| \ r01an6992_adc_nomal_mode_scan_mode <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_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 |
| ¥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).

### 4.2.3 List of Option Byte Settings

Table 4-9 shows the option byte settings.

Table 4-9 Option Byte Settings

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

### 4.2.4 List of Constants

Constant is not used in the sample code.

### 4.2.5 List of Variables

Table 4-10 shows the variables used in the sample code.

Table 4-10 Variables used in the sample code

| Variables Name | Type | Contents | Function that uses the variables |
|---|---|---|---|
| g_result_buffer[4] | uint16_t | Storage of A/D conversion results | r_Config_ADC_interrupt |

### 4.2.6 Function list

Table 4-11 Function List shows the functions used in the sample code. However, functions generated by the Smart Configurator that have not been modified are excluded.

Table 4-11 Function List

| Function Name | Overview | Source file |
|---|---|---|
| main | Main processing | main.c |
| r_Config_ADC_interrupt | A/D converter interrupt processing | Config_TRD0_user.c |

### 4.2.7 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 RTC and A/D converter. |
| Arguments | None |
| Return values | None |
| Remarks | None |

[Function name] r_Config_ADC_interrupt

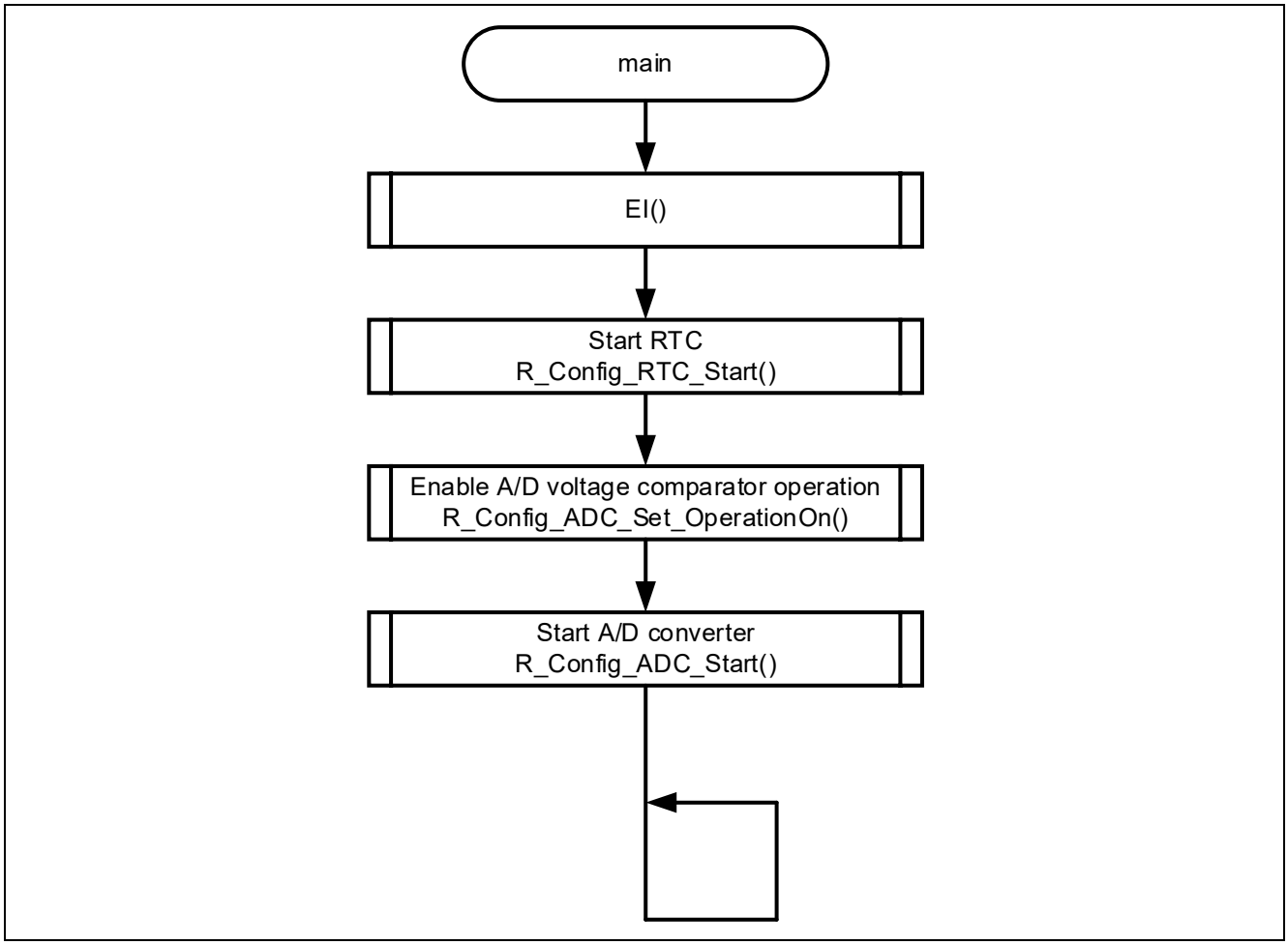
| | |
|---|---|
| Overview | A/D converter interrupt processing |
| Headers | r_cg_macrodriver.h, r_cg_userdefine.h, Config_ADC.h |
| Declaration | static void __near r_Config_ADC_ad0_interrupt(void); |
| Description | When A/D conversion ends, this function reads the A/D conversion result from the ADCR0-3 register and then stores the result in the variable of the internal RAM. |
| Arguments | None |
| Return values | None |
| Remarks | None |

### 4.2.8 Flowchart

#### 4.2.8.1 Main Process

Figure 4-11 shows the flowchart for the main process.

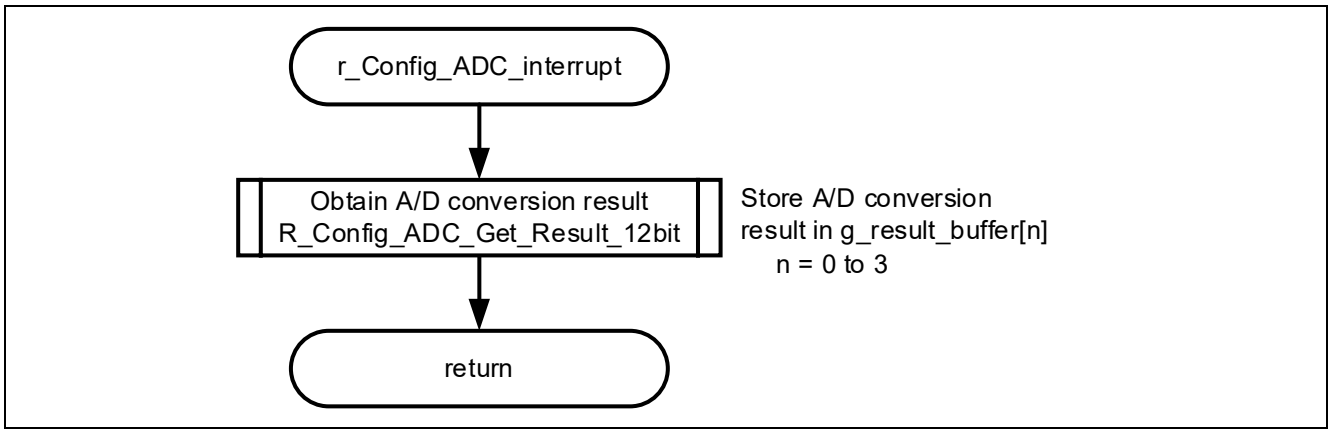
Figure 4-11 Main Process




#### 4.2.8.2 r_Config_ADC_interrupt function

Figure 4-12 shows the flowchart of r_Config_ADC_interrupt function.

Figure 4-12 r_Config_ADC_interrupt function



## 5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

## 6. 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 Enabled) (R01AN6973)

(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/.