RA2E1/RA2E2 Group

Low Power Application (Use of ADC, DTC and ELC at Snooze mode) for FPB-RA2E1 and FPB-RA2E2

Introduction

This application note describes the features of RA2E1 and RA2E2 MCUs that are useful for low-power operation, typically required for logging data for long durations. The FPB-RA2E1 or FPB-RA2E2 kit is used for creating such a low-power data logging system. The functionality demonstrated here is typically required in products such as fitness trackers, fleet tracking devices, and so forth. The data logger uses the 12-bit A/D Converter (ADC), Data Transfer Controller (DTC), Asynchronous General-purpose Timer (AGT), Real-time Clock (RTC) (Applicable for FPB-RA2E1), Event Link Controller (ELC), Data Operation Circuit (DOC), and Low-power Mode (LPM). The Snooze mode and Software Standby mode are used to reduce power consumption by minimizing the CPU operation time. Application projects use the Integrated Development Environment e^2 studio IDE and Flexible Software Package (FSP) provided for the RA family.

This application note has been changed to a simple function from application note *RA2L1/RA2E1 Group Low e2Power Application Example (Data Logger)* (R30AN0384) so that operation can be confirmed with FPB-RA2E1 and FPB-RA1E2.

Prerequisites

We assume that you have developed Renesas e^2 studio IDE and Flexible Software Package (FSP). We recommend that you build and run the Blinky project according to the section “Tutorial: Your First RA MCU Project – Blinky” in the FSP User’s Manual (R11UM0155) prior to trying out this application. You can then become familiar with the e^2 studio IDE and FSP and verify that the debugging connectivity to the boards is working correctly.

Required Resources

This application project is created for Renesas RA Family MCU RA2E1 and RA2E2. When applying this application note to other MCUs, be sure to change it according to the specifications of the MCU and evaluate it carefully. The resources required for this application project are as follows.

Hardware

- Renesas RA Kit FPB-RA2E1 or FPB-RA2E2
- Seeed Grove Base Shield V2.0 for Arduino (applicable for FPB-RA2E1) (www.seeedstudio.com/Base-Shield-V2.html)
- Seeed Grove - Luminance Sensor (www.seeedstudio.com/Grove-Luminance-Sensor.html)
- Seeed Grove - Temperature Sensor (www.seeedstudio.com/Grove-Temperature-Sensor.html)
- USB – TTL Serial connector
- Grove Male Jumper wire

Development Tools and Software

- e^2 studio IDE version 2024-07
- Renesas Flexible Software Package (FSP) version 5.5.0
- GCC ARM Embedded Toolchain version 13.2.1.arm-13-7
- Terminal software (like Tera Term)

Contents

1. Overview

1.1 Overview of Specifications

This application project acquires sensor data and compares it against thresholds at regular intervals. These processes are intermittently executed with Snooze mode during Software Standby mode to achieve operation as a low-power data logger.

In this application project, Snooze mode is entered every 30 minutes during Software Standby mode. While operating in Snooze mode, A/D conversion and the level judgment of the result are performed. After judging the level, the system returns to the Software Standby mode. However, if the conversion result is greater than or equal to the threshold value, Snooze mode is canceled, and MCU is transitioned to Normal mode. In addition, the low-power mode is canceled every 24 hours, and the measured data accumulated in the buffer is transmitted. After the data transmission, the process repeats. When a user button is pressed, the low-power mode is canceled, and the requested data is transmitted. Table 1 explains the details of these processes for each function.

Table 1. List of Functions

Function	Process Description
Data Acquisition Function	A/D conversion is triggered by the AGT count underflow event, and conversion results are stored in the measurement data buffer.
Level Judgement Function	The level of A/D conversion results is judged by the Data Operation Circuit (DOC). If the conversion result is greater than or equal to the threshold value, the interrupt is occurred.
Data Output Function	When the RTC alarm interrupt*1, external IRQ interrupt, and DOC interrupt occur, the mode shifts to Normal mode and the data is output by UART communication.

Note: 1. RTC alarm interrupt available in RA2E1.

Figure 1 shows the RA2E1 MCU states and mode transition events, and Figure 2 shows a conceptual diagram of the operation modes and current consumption for RA2E1.

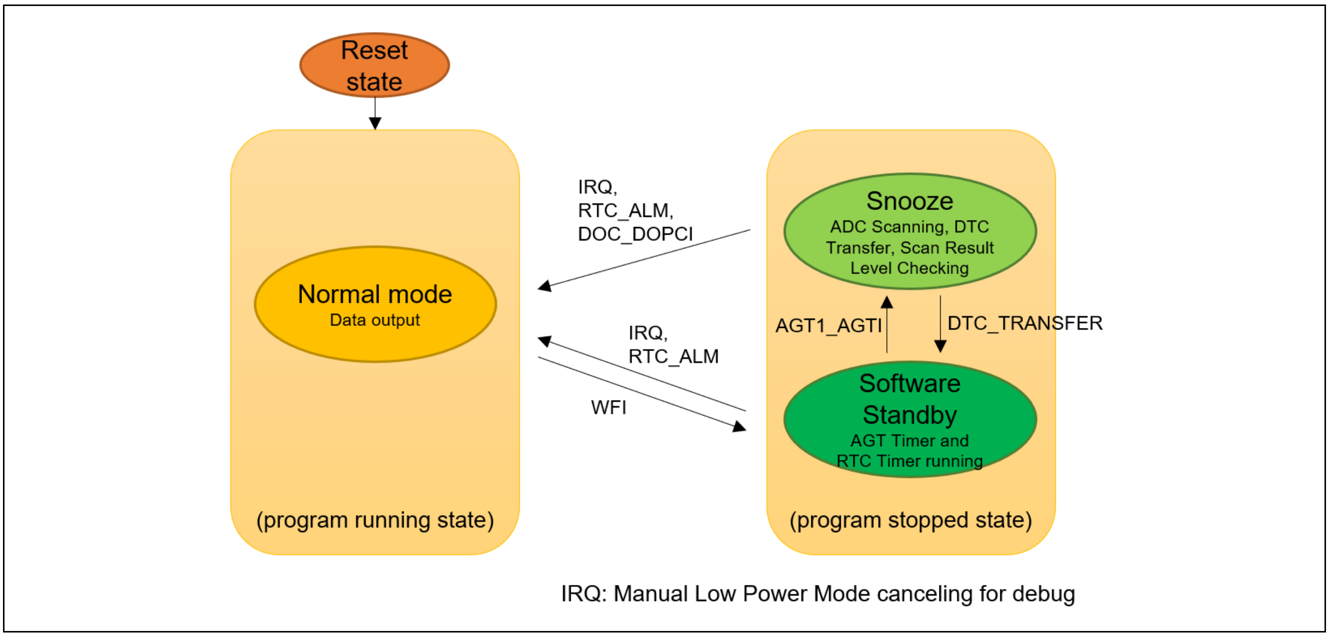
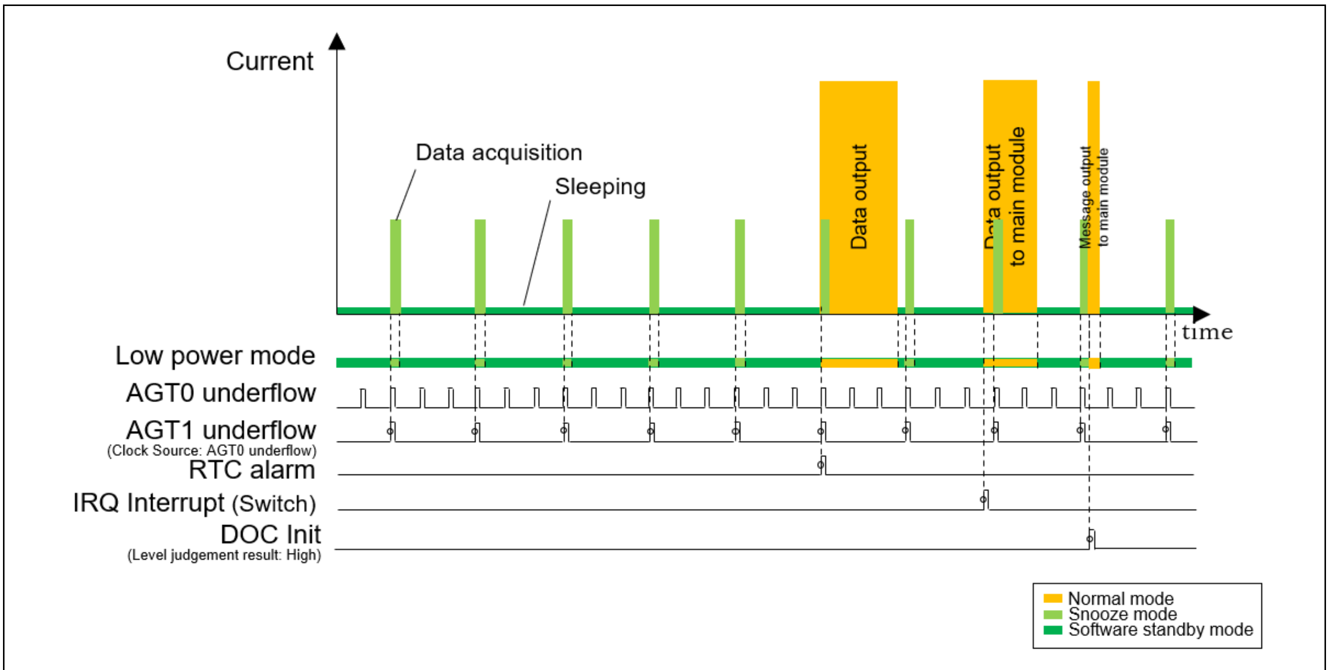


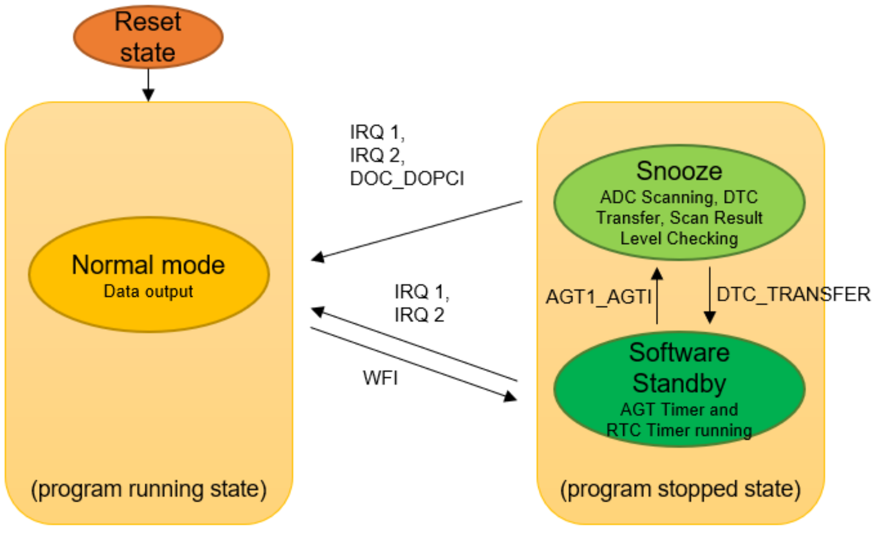
Figure 1. RA2E1 MCU Status and Mode Transition Events



Note: The current consumption and the duration of each process are conceptual.

Figure 2. RA2E1 Conceptual Diagram of Operation Mode and Current Consumption

Figure 3 shows the RA2E2 MCU states and mode transition events, and Figure 4 shows a conceptual diagram of the operation modes and current consumption for RA2E2.



IRQ 1: Manual (switch) Low Power Mode canceling for debug

IRQ 2: Periodic interrupt generated from AGTW 0 timer

Figure 3. RA2E2 MCU Status and Mode Transition Events

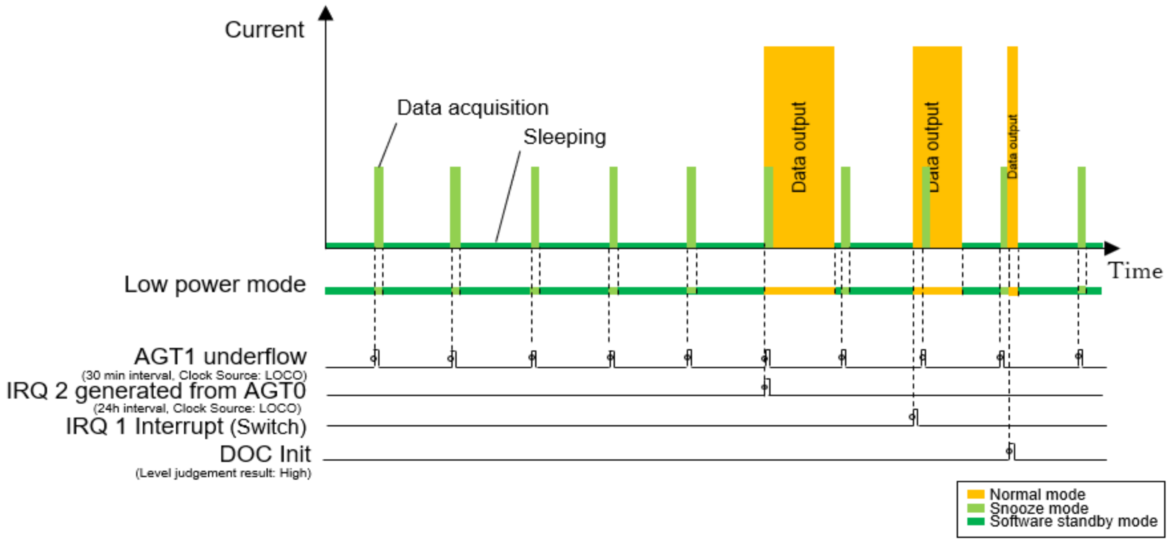


Figure 4. RA2E2 Conceptual Diagram of Operation Mode and Current Consumption

1.1.1 Data Acquisition Function/Level Judgment Function

In the data acquisition function, the AGT timer interrupt*1 transitions from Software Standby mode to Snooze mode and performs A/D conversion on two channels. The A/D conversion result is stored in the measurement data buffer using the DTC. By the DTC transfer completion event, the system returns to Software Standby mode.

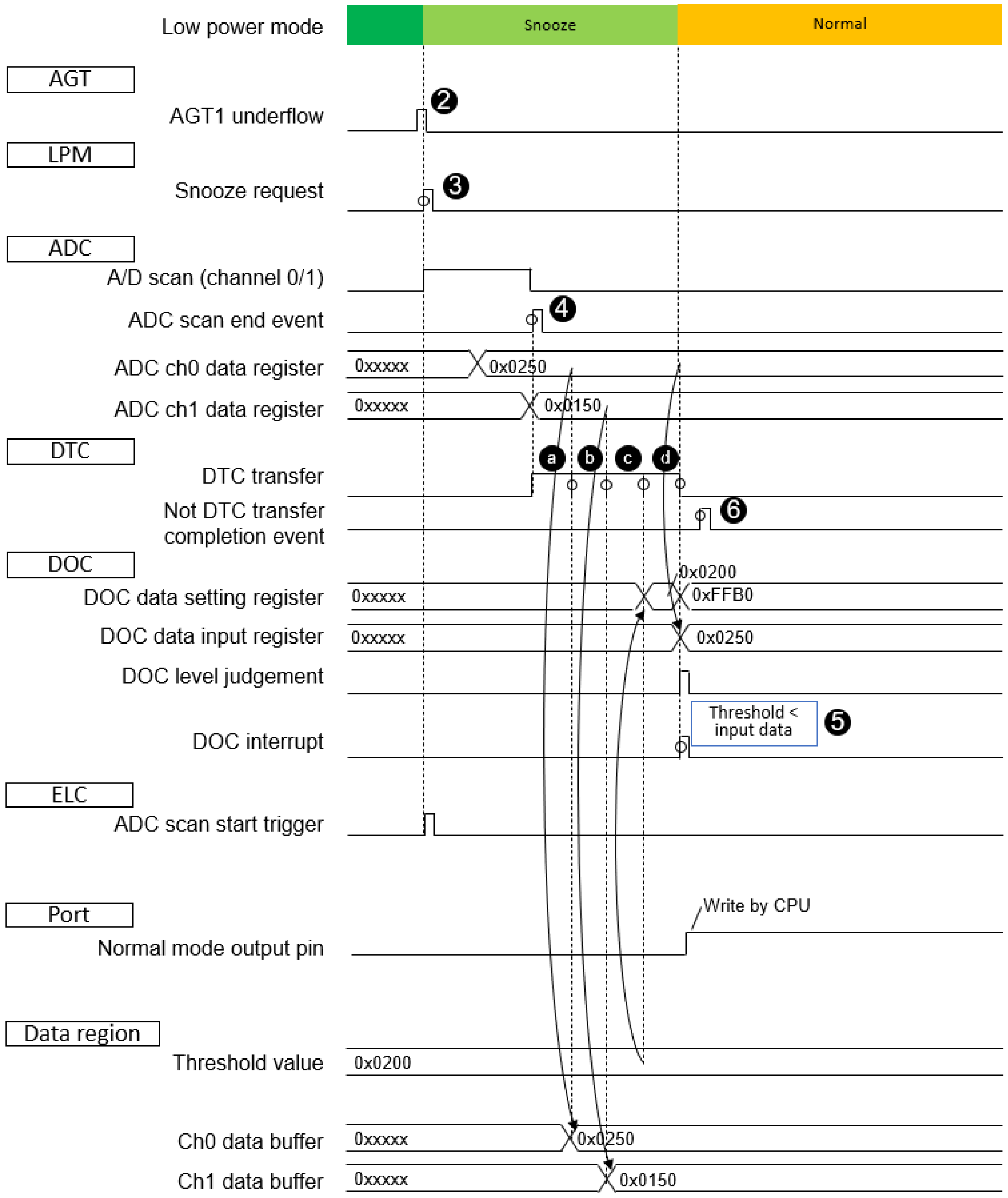
Note: 1. In order to create long-term periodic wakeup event, a two-channel connected AGT (16-bit AGT0 + 16-bit AGT1) is used for RA2E1 MCU. RA2E2 MCU can use the single AGTW because AGTW is a 32-bit timer.

The level judgement function uses the Data Operation Circuit (DOC) to judge the level of the A/D conversion results. If the conversion result is greater than or equal to the threshold value, the DOC interrupt occurs, and the MCU transitions to Normal mode.

The processing sequence for the above functions is described using Figure 5

1. For RA2E1, the AGT0 timer generates an AGT0 underflow signal at every 1 minute.
2. For RA2E1, the AGT1 timer uses the AGT0 underflow signal as the count source and creates an AGT1 underflow interrupt at 30 counts (every 30 minutes) of this signal. When an AGT1 underflow interrupt occurs, it requires a transition to Snooze mode.
3. For RA2E2, the single channel AGTW1 generates the underflow interrupt every 30 minutes, which requests a transition to Snooze mode.
4. The ADC starts A/D conversion by a synchronous trigger from the ELC-linked event from AGT1 or Snooze request event.
5. When an ADC conversion end interrupt is generated, the DTC performs transfer processing in the following order according to the transfer information set for repeat transfer mode and chain transfer.
    a. The conversion result of A/D converter channel 0 or 19 is transferred to the channel 0 or 19 measurement data buffer.
    b. The conversion result of A/D converter channel 1 or 21 is transferred to the channel 1 or 21 measurement data buffer.
    c. The value of the level judgment threshold storage variable is transferred to the DOC data setting register.
    d. The conversion result of A/D converter channel 1 or 21 is transferred to the DOC data input register.
6. When the level judgment result event (Data Operation Circuit interrupt) of channel 1 or 21 occurs, the Snooze mode is canceled.
7. When a Not DTC transfer completion interrupt (DTC_TRANSFER) is generated due to the completion of the transfer processing of step 4, the system returns to the Software Standby mode.

Note: ADC channels 0, 1 are used for RA2E1 and channels 19, 21 are used for RA2E2.



Note: The duration of each processing and event occurrence represents a concept.

Note: The processing sequence ① is omitted. Refer to section 2.6 for details.

Figure 5. Timing Chart for Data Acquisition and Level Judgement

1.1.2 Data Output Function

The data output function transitions from Software Standby mode to Normal mode by RTC alarm interrupt, external IRQ interrupt, and DOC interrupt and outputs data by UART communication. After the data output is completed, it transits to low-power mode. However, if the transition request event to Normal mode is the RTC alarm interrupt, the RTC alarm for the next transition from Software Standby mode to Normal mode is set, and then the mode transitions to the low-power mode.

The above functions are described in the processing sequence as follows:

1. For the RTC alarm interrupt, external IRQ interrupt, and DOC interrupt, cancel the low-power mode.
2. After the low-power mode is canceled, transmit data is generated, and the SCI outputs the data.
3. Transits to low-power mode after the data output is completed.

Note: RA2E2 MCU does not support the RTC. To generate 24-hour periodic intervals, AGTW0 and IRQ2 are used. AGTW0 is configured with a periodic timer at 24 hours, which generates AGTO output. A jumper wire connected to AGT0 output to IRQ 2 detects the external interrupt on the IRQ2 pin.

1.2 Peripheral Modules of RA2E1 and RA2E2 MCU

Table 2 lists the main peripheral modules of RA2E1 and RA2E2 MCU used in this application project and their typical uses. These are illustrated in the functional overview of RA2E1 MCU in Figure 6 and RA2E2 in Figure 7 respectively.

Table 2. Main Peripheral Modules

Module	Typical Uses
Low Power Modes	The Snooze mode and the Software Standby mode are used to achieve low-power consumption.
12-bit ADC	Converts the analog continuous-time signal from the sensor to a discrete digital signal value.
DTC	The analog conversion result of the A/D converter is transferred to the buffer using the DTC in repeat transfer mode. A DTC chain transfer is used to transfer the data from the ADC data register, and from RAM/ROM memory to the DOC register for level judgement.
DOC	Use the subtraction mode to judge the level of the A/D conversion result.
16-bit AGT 0/1 *1	Generates a transition request to Snooze mode and starts data acquisition timing every 30 minutes.
RTC	Measure the time since the power is on. The 24-hour alarm interrupt is used to cancel the low-power mode and generate the timing of data output.
SCI	Performs UART communication with the external equipment.
ELC	Connect a Snooze transition request event or an AGT underflow event to the A/D converter start conversion event.
32-bit AGTW 0/1 *2	Single channel 32-bit AGTW0 timer with IRQ2 is used to cancel the low-power mode and generate the periodic interrupt of data output. Single channel 32-bit AGTW1 underflow event generates a transition request to Snooze mode and starts data acquisition timing every 30 minutes
ICU	The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC) and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts.

Notes: 1. 16-bit AGT 0/1 available in RA2E1.
2. 32-bit AGTW 0/1 available in RA2E2.

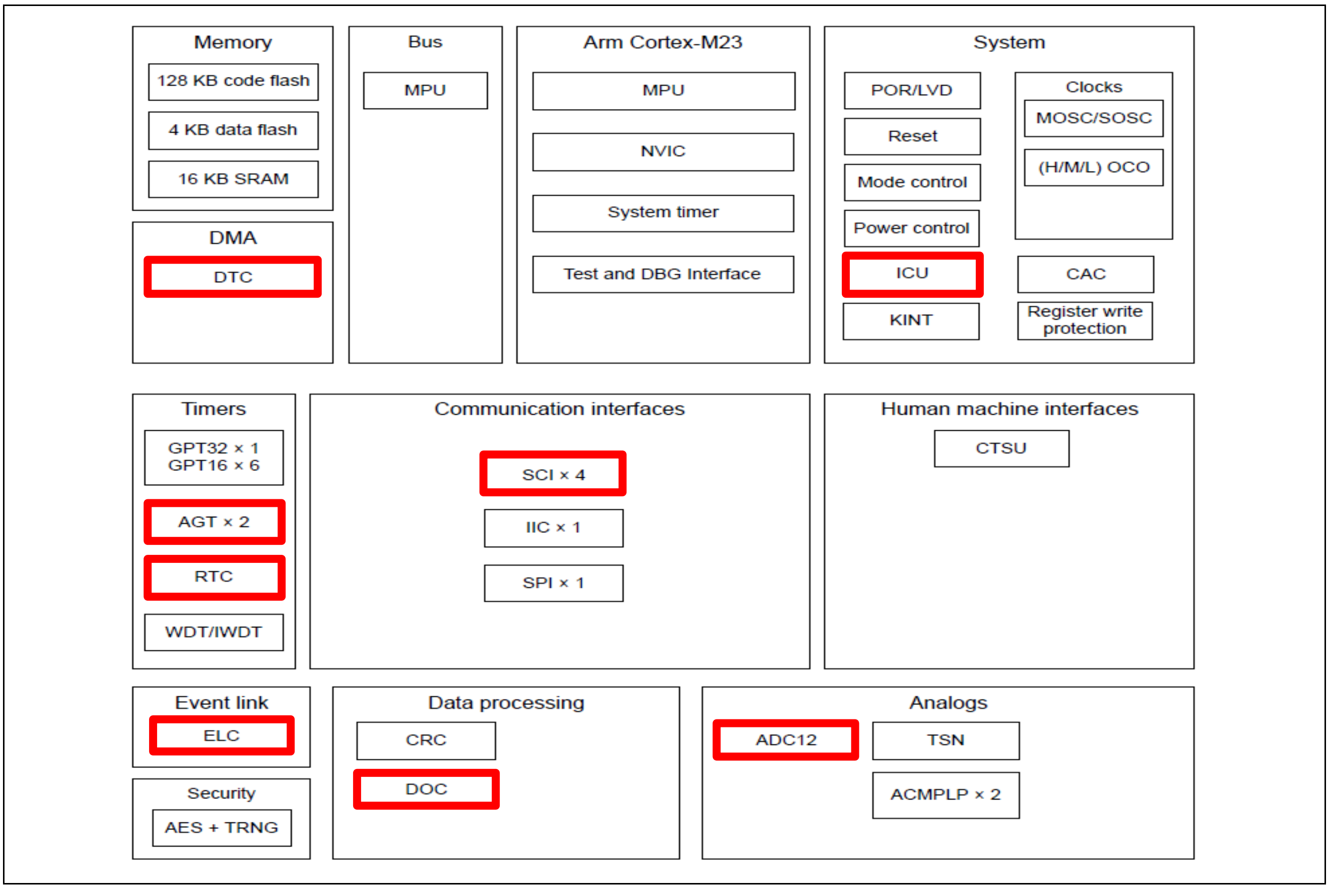


Figure 6. Outline of RA2E1 MCU Function

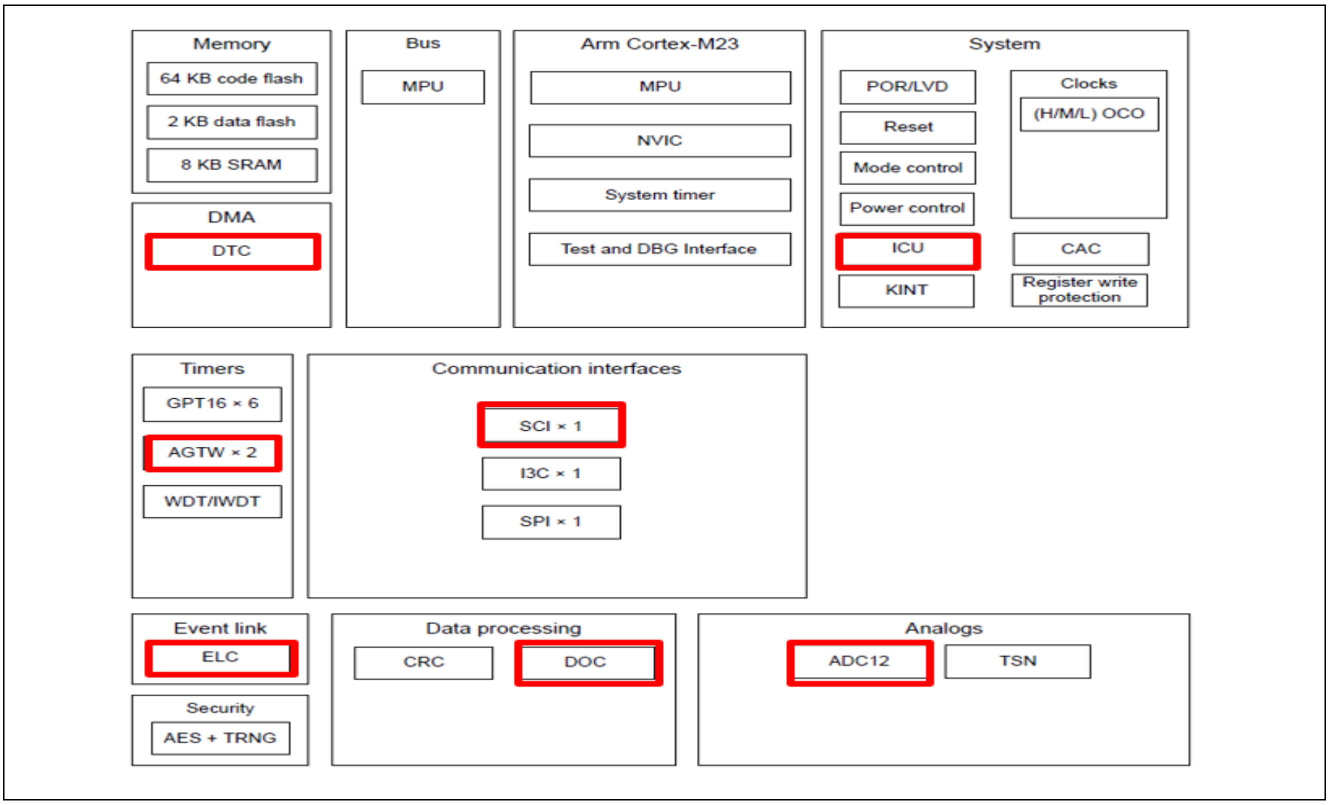


Figure 7. Outline of RA2E2 MCU Function

2. Description of Functions Used

This section describes the functions of the LPM, ADC, DTC, ELC, AGT, and RTC modules of RA2E1 or RA2E2 MCUs and explains how to set them to achieve the expected operation.

2.1 Low Power Modes

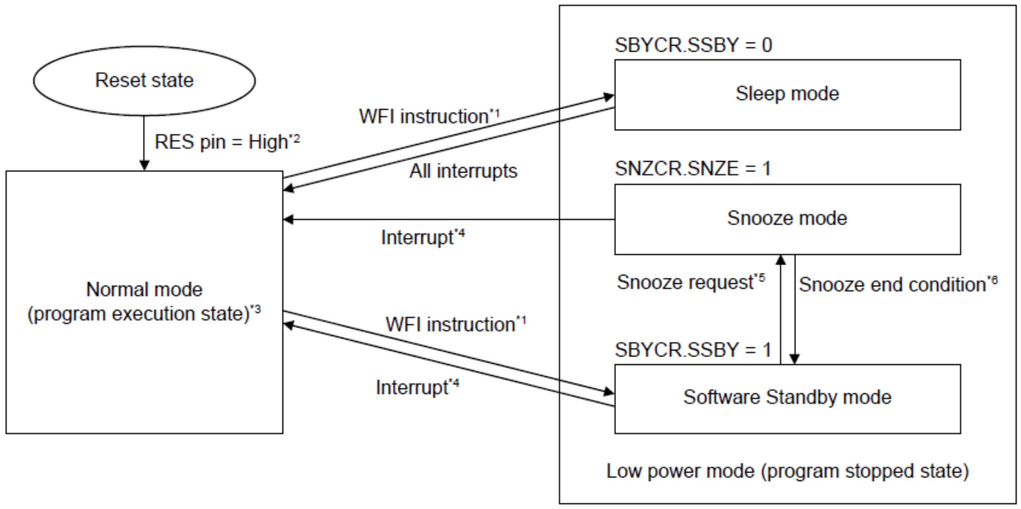
2.1.1 Available Low Power Modes

The table “Operating conditions of each low-power mode” in the RA2E1 (R01UH0852) or RA2E2 (R01UH0919) User's Manual describes the conditions to transition to low-power modes, the states of the CPUs and peripheral modules, and the condition to cancel each mode.

The available low-power modes are as follows:

- Sleep mode
- Software Standby mode
- Snooze mode

Figure 8 shows a schematic diagram of the low-power mode transitions.



Note 1. When an interrupt that acts as a trigger for cancel is received during a transition to the program stopped state after the execution of a WFI instruction, the MCU executes interrupt exception handling instead of a transition to low power mode.
Note 2. The MOCO is the source of the operating clock following a transition from the reset state to Normal mode.
Note 3. The transition to Normal mode is made from an interrupt in Sleep mode, Software Standby mode, or Snooze mode. The clock source is the same as before entering the low power mode.
Note 4. See Table 10.3.
Note 5. See section 10.5.1. Setting Operating Power Control Mode.
Note 6. See section 10.8.3. Returning from Snooze Mode to Software Standby Mode.

Figure 8. Schematic Diagram of Low Power Mode Transitions

2.1.1.1 Sleep Mode

Generally, CPU operation is the most significant factor for the increase in power consumption. In this mode, the CPU halts operation but retains the value of the CPU's internal registers. Peripheral functions other than the CPU are not stopped. The Sleep mode is canceled when an available reset or interrupt occurs in Sleep mode. All interrupt sources are available. When interrupts are used to cancel Sleep mode, the corresponding IELSRn register must be set prior to executing WFI instruction.

2.1.1.2 Software Standby Mode

This mode dramatically reduces power consumption by stopping the CPU, most on-chip peripheral functions, and oscillators. However, the CPU’s internal registers, SRAM data, the on-chip peripheral functions, and I/O port status are retained.

2.1.1.3 Snooze Mode

This mode is an extension of Software Standby mode that allows limited peripheral modules to operate with the CPU halted. This reduces current consumption by flexible operation of peripheral modules required by the application. Snooze mode can be entered from Software Standby mode by a specified interrupt request. Interrupt requests available in Snooze mode can also transition from Snooze mode to Normal mode or Software Standby mode.

2.1.2 Low Power Mode Transition Events

For details on the available low-power mode transitions, refer to the relevant sections in the RA2E1 (R01UH0852) or RA2E2 (R01UH0919) User's Manual below.

- Table: Available Snooze requests to switch to Snooze mode.
- Table: Available Snooze end requests (triggers to return to Software Standby mode).
- Table: Available interrupt sources to transition to Normal mode from Snooze mode and Software Standby mode.

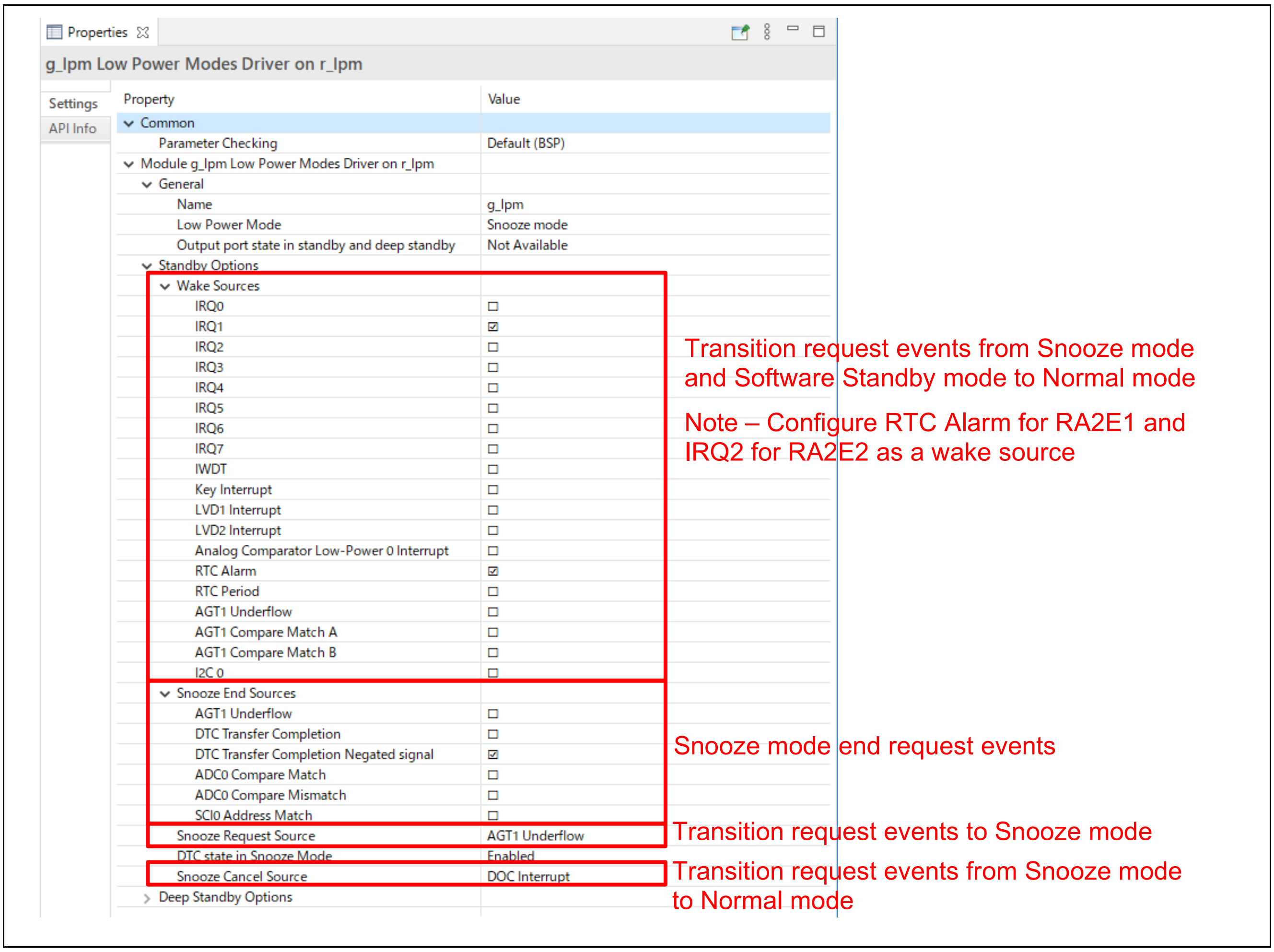
Table 3 lists the mode transition events used in this application project. Figure 9 also shows the low-power transition event settings on the FSP configurator.

Table 3. Mode Transition Events Used in this Application Project

Transition Request	Event
Request for transition to Snooze mode	AGT1_AGTI
Snooze mode end request	DTC_TRANSFER
Request for transition to Normal mode from Snooze mode or Software Standby mode	PORT_IRQ1 PORT_IRQ2[*1] RTC_ALM[*2] DOC_DOPCI

Notes: 1. Applicable for RA2E2.

2. Applicable for RA2E1.



g_lpm Low Power Modes Driver on r_lpm

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_lpm Low Power Modes Driver on r_lpm	
General	
Name	g_lpm
Low Power Mode	Snooze mode
Output port state in standby and deep standby	Not Available
Standby Options	
Wake Sources	
IRQ0	☐
IRQ1	☑
IRQ2	☐
IRQ3	☐
IRQ4	☐
IRQ5	☐
IRQ6	☐
IRQ7	☐
IWDT	☐
Key Interrupt	☐
LVD1 Interrupt	☐
LVD2 Interrupt	☐
Analog Comparator Low-Power 0 Interrupt	☐
RTC Alarm	☑
RTC Period	☐
AGT1 Underflow	☐
AGT1 Compare Match A	☐
AGT1 Compare Match B	☐
I2C 0	☐
Snooze End Sources	
AGT1 Underflow	☐
DTC Transfer Completion	☐
DTC Transfer Completion Negated signal	☑
ADC0 Compare Match	☐
ADC0 Compare Mismatch	☐
SCI0 Address Match	☐
Snooze Request Source	AGT1 Underflow
DTC state in Snooze Mode	Enabled
Snooze Cancel Source	DOC Interrupt
Deep Standby Options	

Transition request events from Snooze mode and Software Standby mode to Normal mode

Note – Configure RTC Alarm for RA2E1 and IRQ2 for RA2E2 as a wake source

Snooze mode end request events

Transition request events to Snooze mode

Transition request events from Snooze mode to Normal mode

Figure 9. Low Power Mode Transition Event Settings on FSP Configurator

2.2 12-Bit A/D Converter

The RA2E1 and RA2E2 MCU include a 12-bit successive approximation A/D converter (ADC12) unit. There are three types of scan conversion operation modes and two types of conversion modes, as shown below.

- Single scan mode: Executes the scan of the specified channel once.
- Continuous scan mode: Scanning of the specified channels is repeated until ADCSR.ADST bit is set to 0 by the software.
- Group scan mode: The scan of the channel selected in Groups A and B is executed once by the synchronous trigger.
- High-speed A/D conversion mode*[1]: Executes A/D conversion at high speed.
- Low-power A/D conversion mode*[1]: The A/D conversion time is longer than the high-speed conversion mode, but it executes with low-power consumption.

Note: 1. Refer to the section “Analog Input Sampling and Scan Conversion Time” and Table “Operating and standby current” in the RA2E1 (R01UH0852) or RA2E2 (R01UH0919) User's Manual for a deep understanding of the difference between A/D conversion modes.

2.2.1 A/D Conversion Channel

The 12-bit A/D converter unit (ADC12) can select up to 13 channels (AN000–AN010, AN017–AN022)*[1] of analog input, temperature sensor, internal reference voltage, or CTSU TSCAP voltage for RA2E1 and up to 8 channels (AN005, AN006, AN009, AN010, AN019–AN022)*[2] of analog input, temperature sensor, internal reference voltage for RA2E2.

Notes 1 and 2: The available input channels depend on a package type.

2.2.2 Analog Pin Connection for FPB-RA2E1

The application project uses Seeed's Grove Base Shield V2.0 for Arduino to connect the sensor through the Arduino-compatible connector of the FPB-RA2E1 kit. J6-1 and J6-2 of Arduino-compatible connectors of FPB-RA2E1 are connected to analog channels 0 (AN000) and 1 (AN001) of the RA2E1 MCU and can be read by selecting channels 0 and 1. Figure 10 shows the analog signal pin connections for Arduino-compatible connectors in the FPB-RA2E1 kit. Figure 11 shows the ADC scan channels settings in the FSP configurator.

The A/D conversion channel is selected by the ADANSA0, ADANSA1 register. A/D conversion starts from the smallest channel number according to the conversion sequence for the analog input channel (ANn). The A/D conversion result is stored in the corresponding A/D data register n (ADDRn).

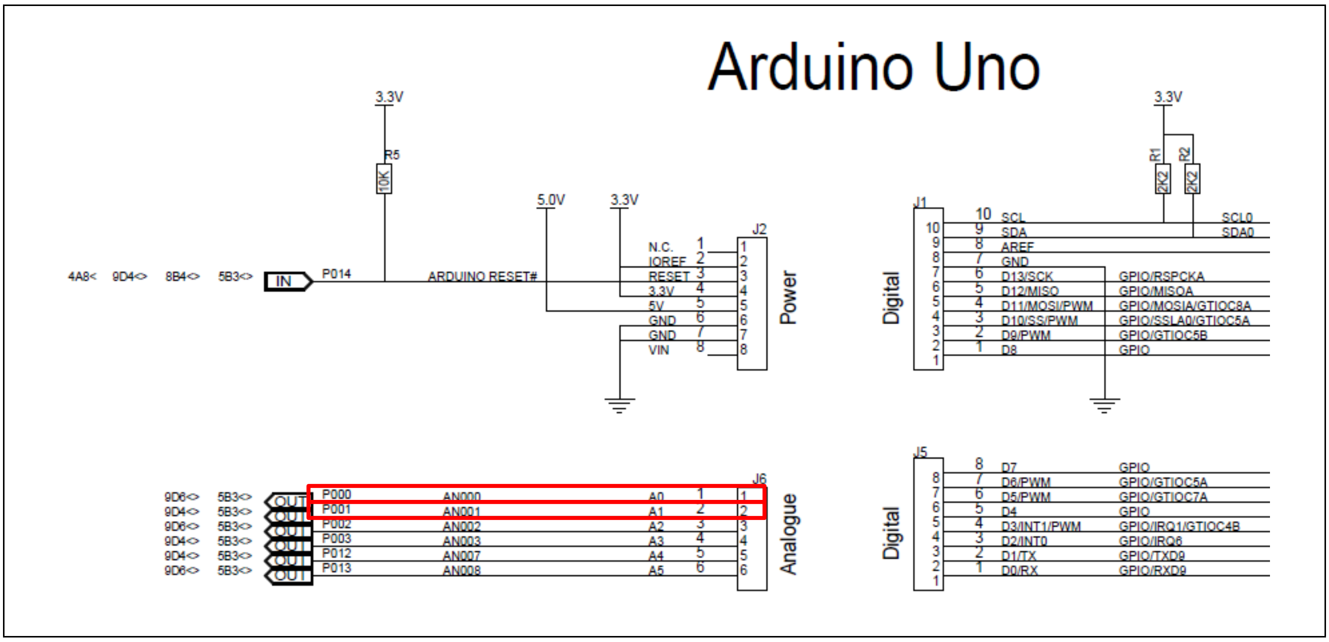
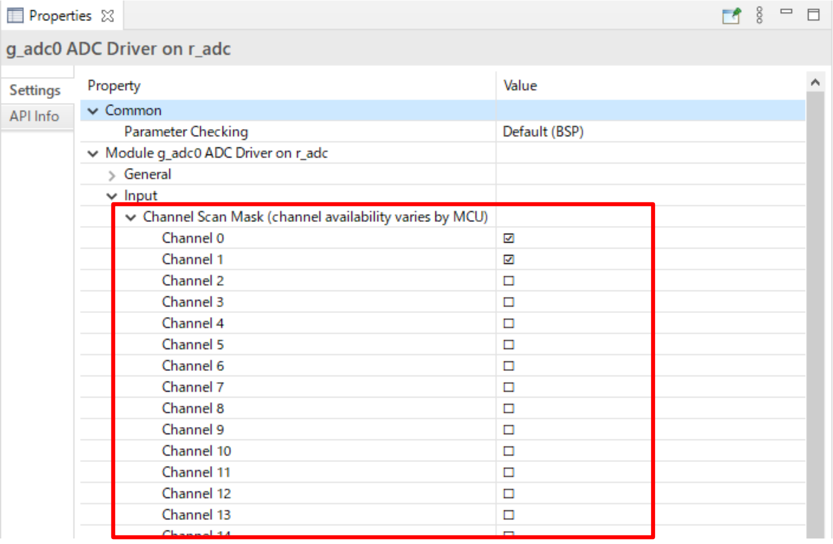


Figure 10. FPB-RA2E1 Kit Arduino-Compatible Connector Analog Signal Pin Connection


Properties

g_adc0 ADC Driver on r_adc

Settings
API Info

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_adc0 ADC Driver on r_adc	
General	
Input	
Channel Scan Mask (channel availability varies by MCU)	
Channel 0	☑
Channel 1	☑
Channel 2	□
Channel 3	□
Channel 4	□
Channel 5	□
Channel 6	□
Channel 7	□
Channel 8	□
Channel 9	□
Channel 10	□
Channel 11	□
Channel 12	□
Channel 13	□

Figure 11. FPB-RA2E1 ADC Scan Channels Settings on FSP Configurator

2.2.3 Analog Pin Connection for FPB-RA2E2

The application project uses a PMOD1 connector to connect the sensor through the grove-male jumper wire in the FPB-RA2E2 kit. Both signal pins of the sensor module are connected to analog channels 19 (AN019) and 21 (AN021) of RA2E2 MCU and can be read by selecting channels 19 and 21. Figure 12 shows the analog signal pin connections in the FPB-RA2E2 kit. Figure 13 shows the ADC scan channel settings in the FSP configurator.

The A/D conversion channel is selected by the ADANSA0 and ADANSA1 register. A/D conversion starts from the smallest channel number according to the conversion sequence for the analog input channel (ANn). The A/D conversion result is stored in the corresponding A/D data register n (ADDRn).

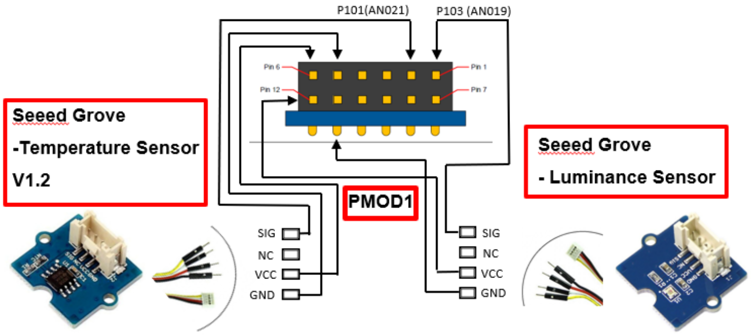
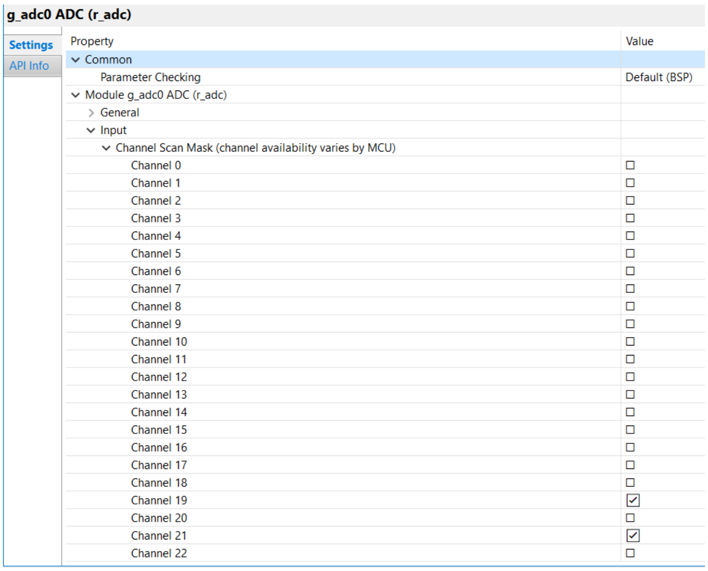


Figure 12. FPB-RA2E2 kit PMOD1 Connection with Analog Sensors



g_adc0 ADC (r_adc)

Settings
API Info

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_adc0 ADC (r_adc)	
General	
Input	
Channel Scan Mask (channel availability varies by MCU)	
Channel 0	□
Channel 1	□
Channel 2	□
Channel 3	□
Channel 4	□
Channel 5	□
Channel 6	□
Channel 7	□
Channel 8	□
Channel 9	□
Channel 10	□
Channel 11	□
Channel 12	□
Channel 13	□
Channel 14	□
Channel 15	□
Channel 16	□
Channel 17	□
Channel 18	□
Channel 19	☑
Channel 20	□
Channel 21	☑
Channel 22	□

Figure 13. FPB-RA2E2 ADC Scan Channels Settings on FSP Configurator

2.2.4 Conversion Start Event

Figure 14 shows ADC12 Input/Output events. The ADC12 conversion start condition can be selected from three triggers: software trigger, synchronous trigger from ELC, and asynchronous trigger by external trigger ADTRG0 pin. In this application project, the synchronous trigger from the ELC is selected as the conversion start condition. Figure 15 shows the A/D conversion start trigger event settings in the FSP configurator.

A/D conversion of a synchronous trigger from the ELC can be started by the preset event of the ELC specified by the ELSRn register, as shown below. Set ADCSR.TRGE bit to 1 and ADCSR.EXTRG bit to 0 and select the corresponding source in ADSTRGR.TRSA[5:0] and ADSTRGR.TRSB[5:0] bits.

- Select ELC_AD00 in the ELC. ELSR8 register
- Select ELC_AD01 in the ELC. ELSR9 register

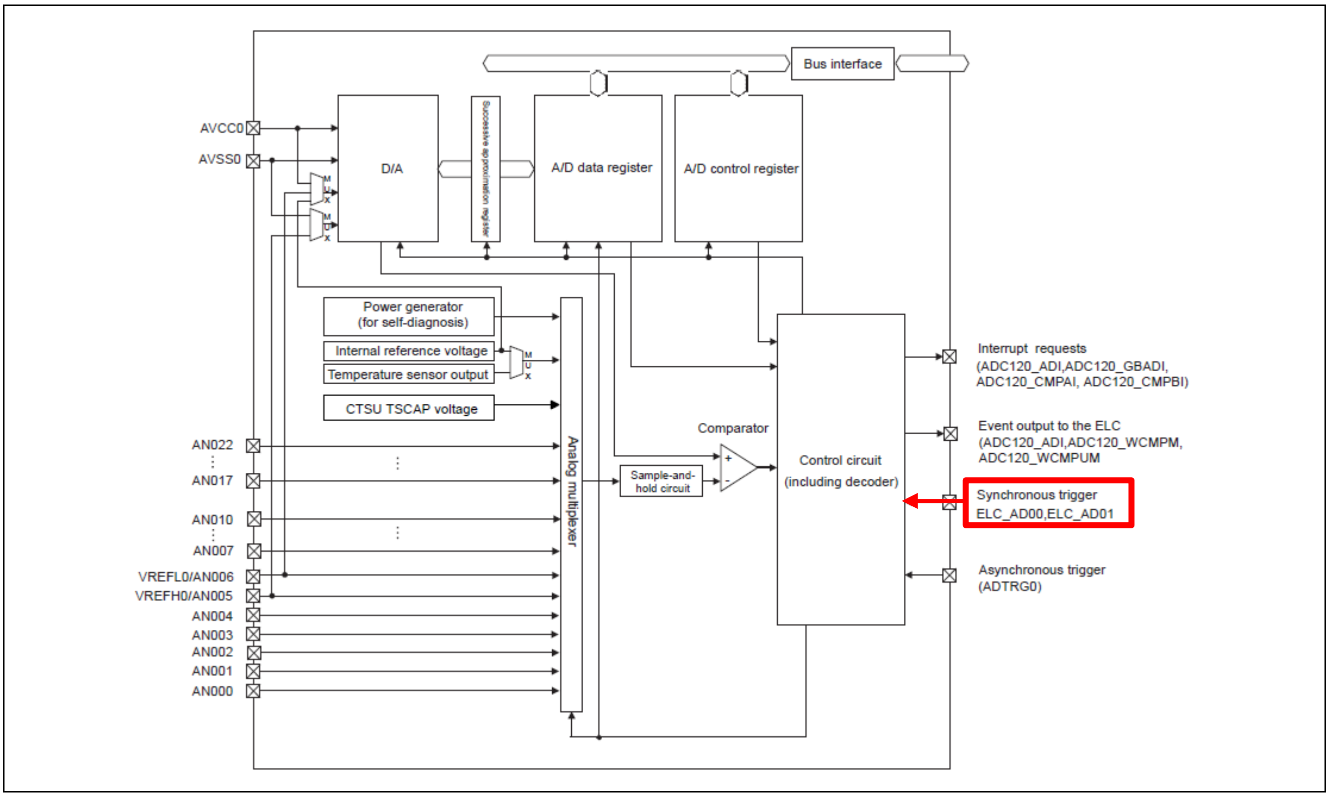
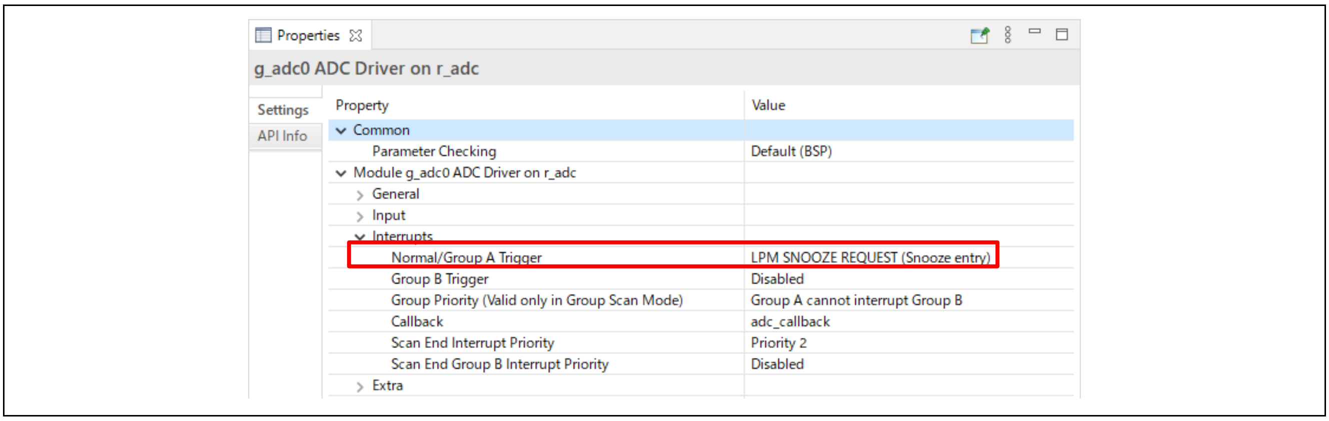


Figure 14. ADC12 Input/Output Events



g_adc0 ADC Driver on r_adc

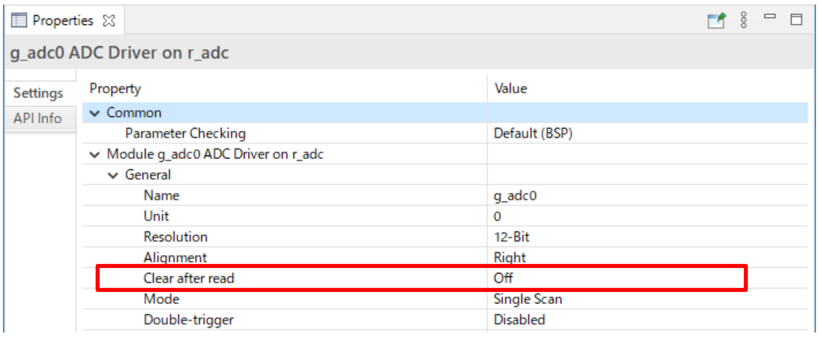
Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_adc0 ADC Driver on r_adc	
General	
Input	
Interrupts	
Normal/Group A Trigger	LPM SNOOZE REQUEST (Snooze entry)
Group B Trigger	Disabled
Group Priority (Valid only in Group Scan Mode)	Group A cannot interrupt Group B
Callback	adc_callback
Scan End Interrupt Priority	Priority 2
Scan End Group B Interrupt Priority	Disabled
Extra	

Figure 15. A/D convert Start Trigger Event Settings in FSP Configurator

2.2.5 Data Register Automatic Clearing Function

When the A/D data register is read by the CPU or DTC, the A/D data register (ADDRn, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR) can be automatically cleared to 0x0000 by setting ADCER.ACE bit to 1. Figure 16 shows the ADC data register auto-clearing function settings in the FSP configurator.

In this application project, the value of the data register (ADDR0) must be read twice in order to transfer the conversion result of the A/D converter channel 0 or 19 to the two data areas. Therefore, the data register automatic clearing function is disabled (set the ACE bit in ADCER to 0).



g_adc0 ADC Driver on r_adc

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_adc0 ADC Driver on r_adc	
General	
Name	g_adc0
Unit	0
Resolution	12-Bit
Alignment	Right
Clear after read	Off
Mode	Single Scan
Double-trigger	Disabled

Figure 16. DC Data Register Auto Clearing Function Settings in FSP Configurator

2.3 DTC Transfer

When the data transfer controller (DTC) is activated by an interrupt request, it transfers data according to the transfer information. To operate DTC, the data to be transferred must be stored in the SRAM area in advance.

The following three transfer modes are available:

- Normal transfer mode: One data transfer is performed by one activation.
- Repeat transfer mode: One data transfer is performed by one activation. When data of repeat size is transferred, it returns to the address at the start of the transfer.
- Block transfer mode: One block is transferred at one activation.

2.3.1 Transfer Start Event

The DTC can be activated by an interrupt request. When IELSRn.DTCE bit in ICU is set to 1, and DTC is activated by the corresponding interrupt. The selector output number n (n = 0 to 31) set in the IELSRn register in the ICU is defined as the interrupt vector number. For the enabled interrupt, a particular DTC interrupt source corresponding to each interrupt vector number n is selected by IELSRn.IELS[4:0] (n = 0 to 31) bits in the ICU.

Figure 17 shows DTC activation by an interrupt request from a peripheral module. Figure 18 also shows the DTC transfer start event settings in the FSP configurator.

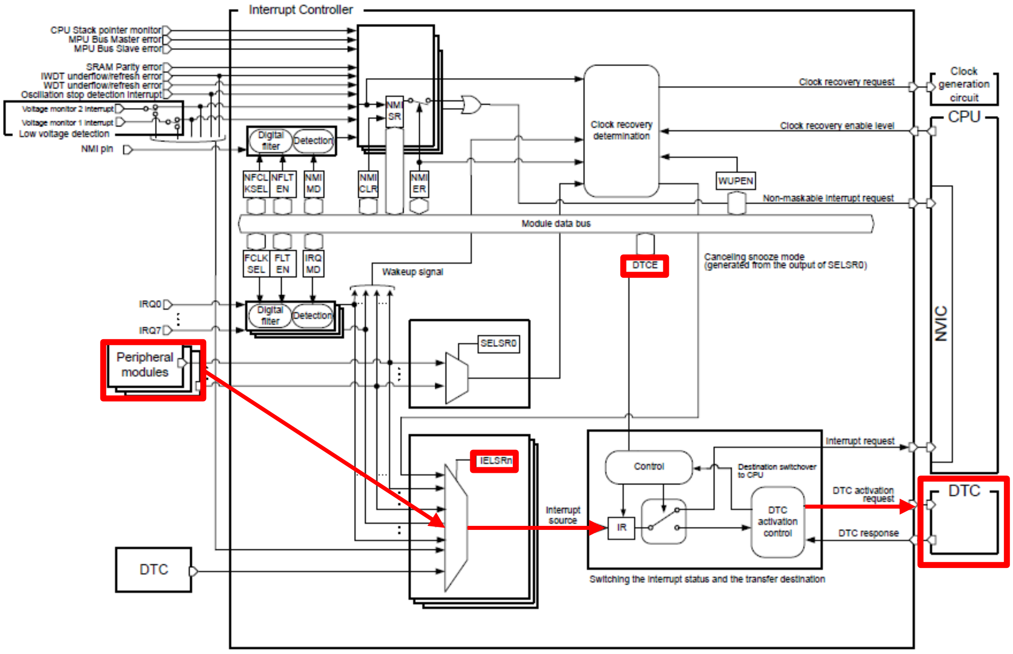
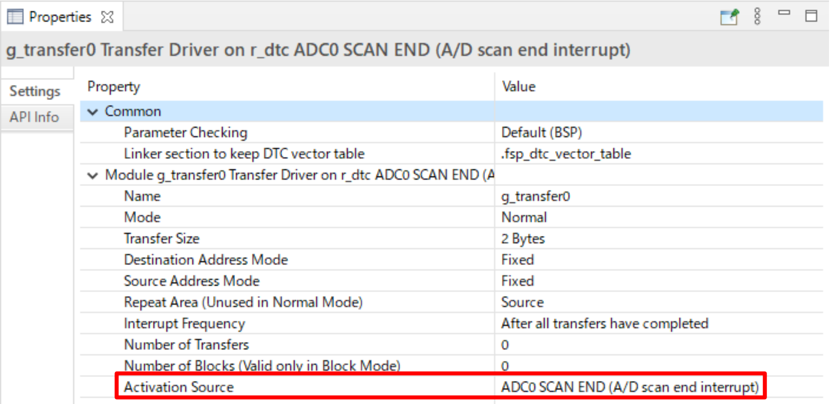


Figure 17. DTC Activation by an Interrupt Request from a Peripheral Module



g_transfer0 Transfer Driver on r_dtc ADC0 SCAN END (A/D scan end interrupt)

Property	Value
Common	
Parameter Checking	Default (BSP)
Linker section to keep DTC vector table	.fsp_dtc_vector_table
Module g_transfer0 Transfer Driver on r_dtc ADC0 SCAN END (A	
Name	g_transfer0
Mode	Normal
Transfer Size	2 Bytes
Destination Address Mode	Fixed
Source Address Mode	Fixed
Repeat Area (Unused in Normal Mode)	Source
Interrupt Frequency	After all transfers have completed
Number of Transfers	0
Number of Blocks (Valid only in Block Mode)	0
Activation Source	ADC0 SCAN END (A/D scan end interrupt)

Figure 18. DTC Transfer Start Event Settings in FSP Configurator

2.3.2 Chain Transfer

The DTC can perform chain transfers in which multiple data transfers are performed continuously for a single activation source. When the CHNE bit in MRB is set to 1, this chain transfer is enabled.

In this application project, the following transfer is executed by one activation factor. Figure 19 shows the chain transfer operation in this case.

- The conversion result of A/D converter channel 0 or 19 is stored in the channel 0 or 19 measurement data buffer.
- The conversion result of A/D converter channel 1 or 21 is stored in the channel 1 or 21 measurement data buffer.
- The value of the level judgment threshold storage variable is stored in the DOC data setting register.
- The conversion result of A/D converter channel 1 or 21 is stored in the DOC data input register.

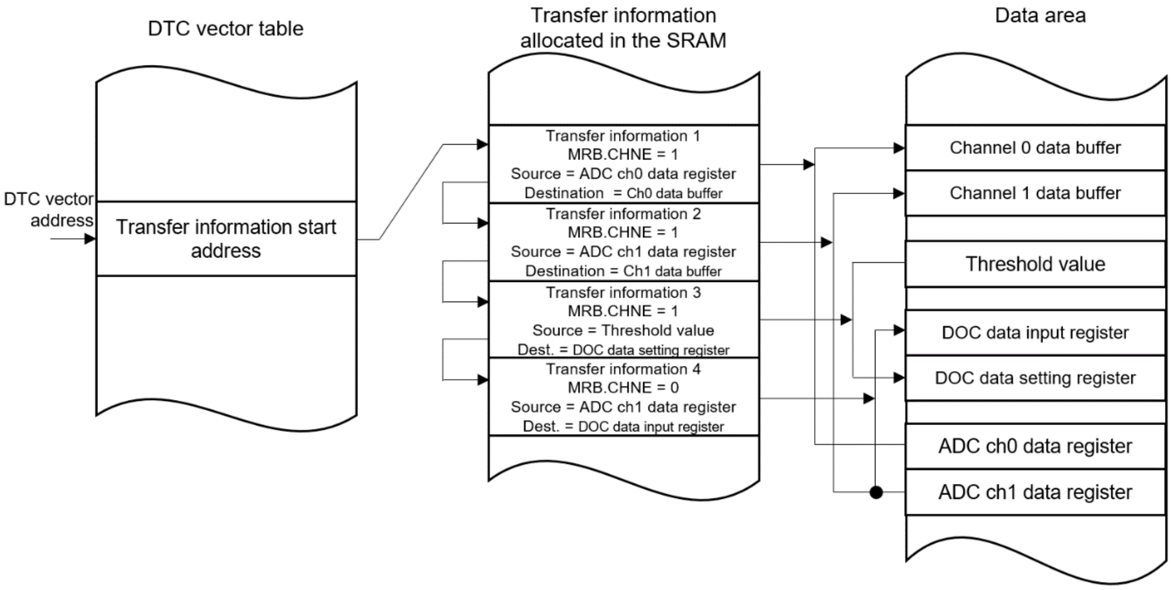


Figure 19. Chain Transfer Operation

When chain transfer is set using the DTC driver of the FSP, the transfer information can be declared as an array of transfer_info_t structures, and the pointer of the first transfer information can be passed to the driver. At this time, the member chain_mode of all transfer_info_t structures, except for the last transfer, must be set to TRANSFER_CHAIN_MODE_EACH or TRANSFER_CHAIN_MODE_END and configured for chain mode. Set the chain_mode of the last transfer to TRANSFER_CHAIN_MODE_DISABLED.

Since the FSP Configurator does not allow the creation of chained Transfer Control Blocks, the R_DTC_Reconfigure function should be used to override the FSP-defined transfer_info_t data passed through the transfer_cfg_t data structure instance.

The chain transfer is shown in Figure 19 is set as follows.

```
transfer_info_t g_dtc_data_transfer_info[4] = {
{
  .transfer_settings_word_b = {
        .dest_addr_mode = TRANSFER_ADDR_MODE_INCREMENTED,
        .repeat_area = TRANSFER_REPEAT_AREA_DESTINATION,
        .irq = TRANSFER_IRQ_END,
        .chain_mode = TRANSFER_CHAIN_MODE_EACH,
        .src_addr_mode = TRANSFER_ADDR_MODE_FIXED,
        .size = TRANSFER_SIZE_2_BYTE,
        .mode = TRANSFER_MODE_REPEAT,
    },
  .p_dest = address of the channel 0 data buffer,
  .p_src = address of ADC channel 0 data register,
  .num_blocks = 0,
  .length = size of the channel 0 data buffer,
},
{
  .transfer_settings_word_b = {
        .dest_addr_mode = TRANSFER_ADDR_MODE_INCREMENTED,
        .repeat_area = TRANSFER_REPEAT_AREA_DESTINATION,
        .irq = TRANSFER_IRQ_END,
        .chain_mode = TRANSFER_CHAIN_MODE_EACH,
        .src_addr_mode = TRANSFER_ADDR_MODE_FIXED,
        .size = TRANSFER_SIZE_2_BYTE,
        .mode = TRANSFER_MODE_REPEAT,
    },
  .p_dest = address of the channel 1 data buffer,
  .p_src = address of ADC channel1 data register,
  .num_blocks = 0,
  .length = size of the channel1 data buffer,
},
{
  .transfer_settings_word_b = {
        .dest_addr_mode = TRANSFER_ADDR_MODE_INCREMENTED,
        .repeat_area = TRANSFER_REPEAT_AREA_DESTINATION,
        .irq = TRANSFER_IRQ_END,
        .chain_mode = TRANSFER_CHAIN_MODE_EACH,
        .src_addr_mode = TRANSFER_ADDR_MODE_FIXED,
        .size = TRANSFER_SIZE_2_BYTE,
        .mode = TRANSFER_MODE_REPEAT,
    },
  .p_dest = address of the DOC data setting register,
  .p_src = address of the level judgment threshold storage variable,
  .num_blocks = 0,
  .length = 1,
},
  {
    .transfer_settings_word_b = {
         .dest_addr_mode = TRANSFER_ADDR_MODE_INCREMENTED,
         .repeat_area = TRANSFER_REPEAT_AREA_DESTINATION,
         .irq = TRANSFER_IRQ_END,
         .chain_mode = TRANSFER_CHAIN_MODE_EACH,
         .src_addr_mode = TRANSFER_ADDR_MODE_FIXED,
         .size = TRANSFER_SIZE_2_BYTE,
         .mode = TRANSFER_MODE_REPEAT,
      },
    .p_dest = address of the DOC data input register,
    .p_src = address of ADC channel 0 data register,
    .num_blocks = 0,
    .length = 1,
  }
};
```

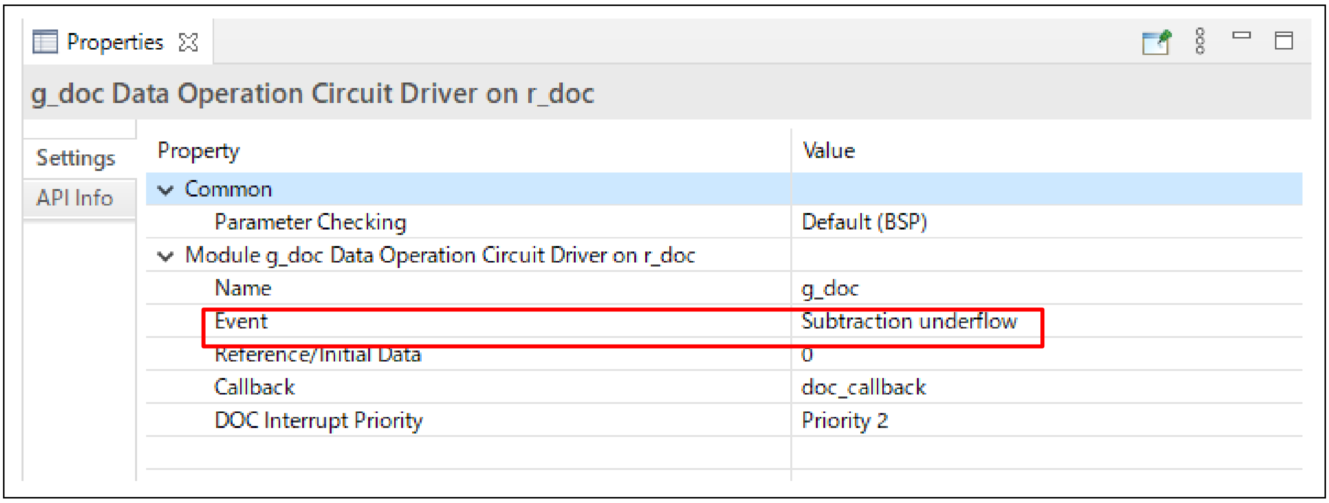
Note: A/D channels 0, 1 are used in RA2E1 and channels 19, 21 used in RA2E2.

2.4 DOC Subtraction Mode

The Data Operation Circuit (DOC) compares, adds, or subtracts 16-bit data. If the selected condition is met, an interrupt request is generated. There are three types of interrupt occurrence conditions that can be selected as follows:

- The compared values either match or mismatch
- The result of data addition is greater than 0xFFFF
- The result of data subtraction is less than 0x0000

Figure 20 shows the DOC function settings in the FSP configurator.



Properties

g_doc Data Operation Circuit Driver on r_doc

Settings
API Info

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_doc Data Operation Circuit Driver on r_doc	
Name	g_doc
Event	Subtraction underflow
Reference/Initial Data	0
Callback	doc_callback
DOC Interrupt Priority	Priority 2

Figure 20. DOC Function Settings on FSP Configurator

In this application project, when the measured data is greater than or equal to the threshold value, the level judgment operation uses the DOC subtraction mode to generate an event. The procedure for determining the level using the DOC subtraction mode is shown below. Figure 21 shows the states of the registers at that time.

Procedure for the level judgement using DOC subtraction mode:

① The threshold of the level judgment is stored in the data setting register (DODSR).

② The conversion result of A/D converter channel 1 or 21 is stored in the data input register (DODIR).

③ If the difference is less than 0x0000, DOCR, DOPCF flag is set to 1.

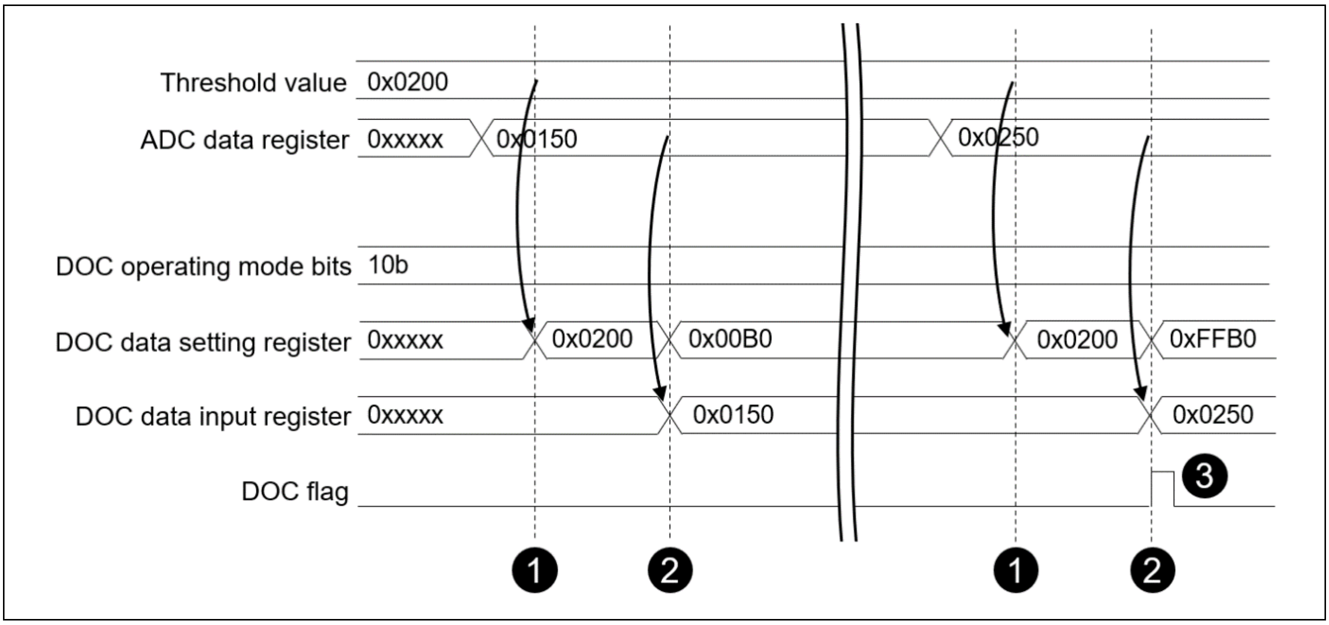


Figure 21. Level Judgement using DOC Subtraction Mode

2.5 ELC Event Connection

The ELC uses event signals/indications generated by peripheral modules as source signals and passes them to other modules to provide direct linkage between modules without interrupting the CPU operation. As a result, Interrupt Service Routines for the CPU from the event-generating module are not required.

When event linking is used, set the ELS[7:0] bits in ELSRn for the module to which the event is linked. Also, ELCR.ELCON bits are set to 1 to enable all event links. Refer to Table “Association between event signal names set in ELSRn.ELS[7:0] bits and signal numbers” in the RA2E1 (R01UH0852) or RA2E2 (R01UH0919) User's Manual for the event number to set in ELSRn register.

As described above, this application project uses the synchronous trigger from the ELC as the A/D conversion start trigger. Also, in this application project, the event that triggers the ADC conversion starts to change according to the state of the low-power mode. In Snooze or Software Standby mode, connect the Snooze request event to ADC12. Otherwise, connect AGT1 underflow interrupt to ADC12. Figure 22 shows the event connection in the ELC module at that time.

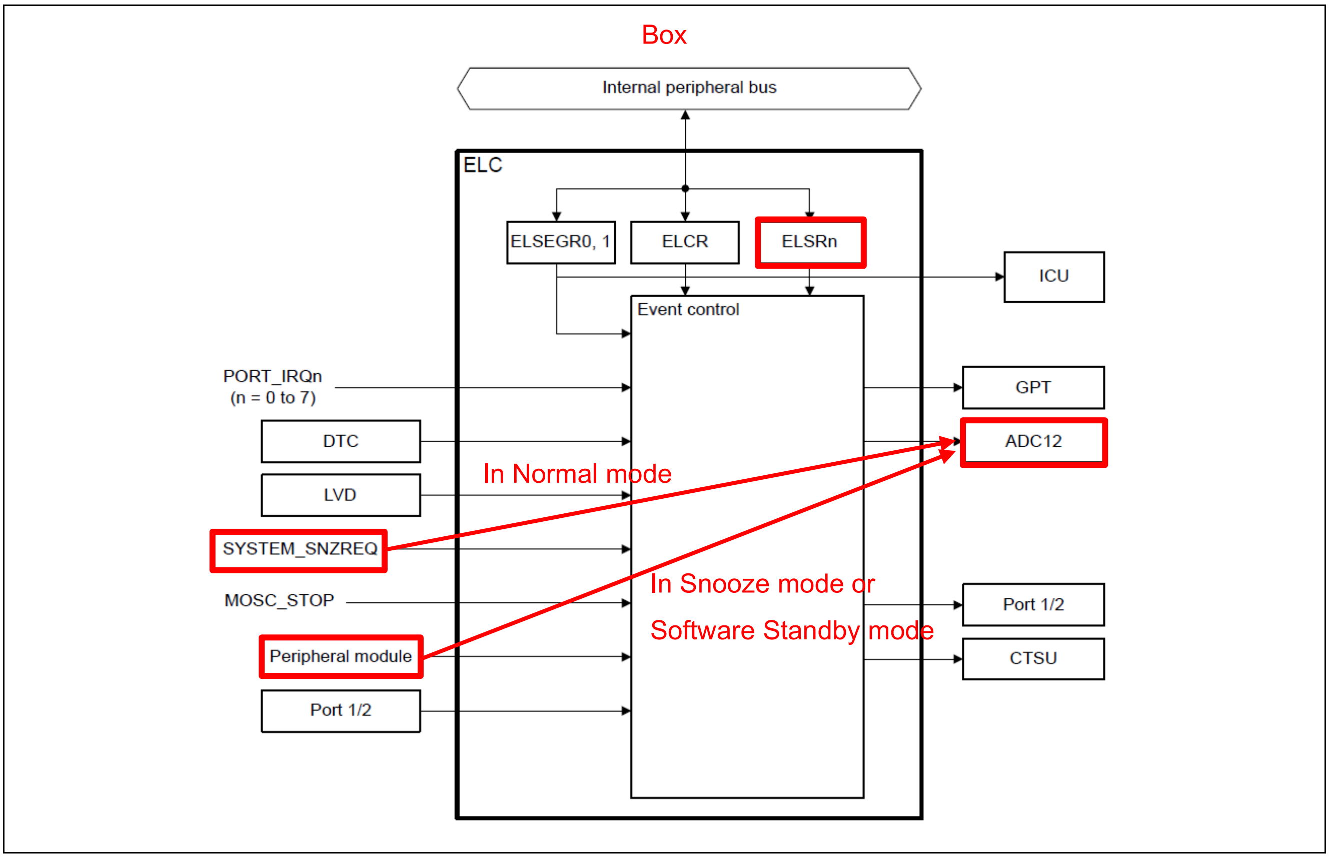
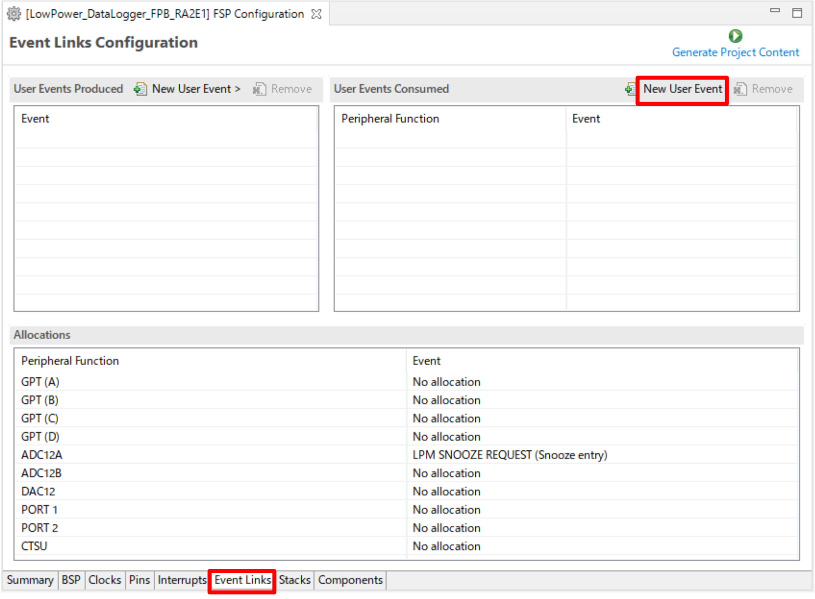


Figure 22. ELC Event Connection for ADC Start Conversion

The ELC event connection can be configured in the FSP Configurator. Some ELC event connections (Example: ADC12 scan start event) can be conveniently configured from the **Properties** view, which is visible upon selecting the module on the **Stacks** tab. The ELC Allocation setting status can be checked with allocations by opening **Event Links,** as shown in Figure 23. To set additional links, set them from **User Events Consumed** > **New User Event**. As shown in the figure below, we have added settings for port-out that cannot be set on the **Stacks** tab.



Peripheral Function	Event
GPT (A)	No allocation
GPT (B)	No allocation
GPT (C)	No allocation
GPT (D)	No allocation
ADC12A	LPM SNOOZE REQUEST (Snooze entry)
ADC12B	No allocation
DAC12	No allocation
PORT 1	No allocation
PORT 2	No allocation
CTSU	No allocation

Figure 23. ELC Connection Settings on the Configurator

2.6 Two-Channel Connected AGT in RA2E1

RA2E1 MCU implements a 16-bit Asynchronous General Purpose Timer (AGTn) with two channels (n=0, 1). This AGT can be used over a two-channel connection to generate a long-period interrupt (Figure 24). To connect to two channels, select the underflow event signal from AGT0 (101b) as the count source in the ATG1.AGTMR1.TCK bit. However, the only combination that can be selected is the one that uses the AGT0 underflow event signal as the count source of AGT1. Refer to the relevant part of the RA2E1 User's Manual (R01UH0852) for more information. Figure 25 also shows AGT count source settings in the FSP configurator.

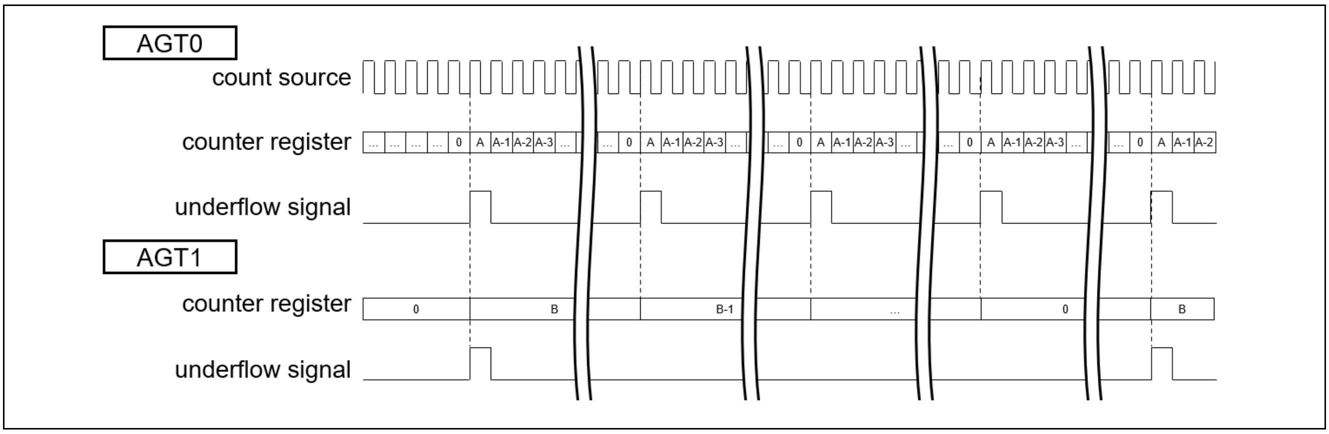
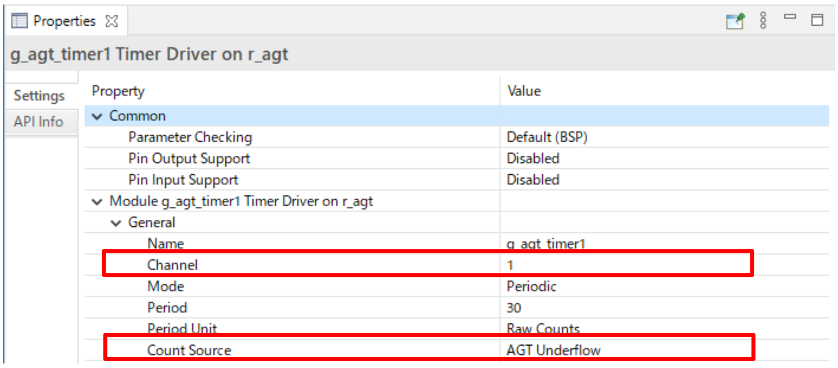


Figure 24. Two-Channel Connected AGT



g_agt_timer1 Timer Driver on r_agt

Property	Value
Common	
Parameter Checking	Default (BSP)
Pin Output Support	Disabled
Pin Input Support	Disabled
Module g_agt_timer1 Timer Driver on r_agt	
General	
Name	g_agt_timer1
Channel	1
Mode	Periodic
Period	30
Period Unit	Raw Counts
Count Source	AGT Underflow

Figure 25. AGT Count Source Settings in FSP Configurator

The maximum period for a single-channel AGT during software standby mode is 256 seconds (approximately 4.3 minutes) when AGTLCLK or AGTSCLK is selected as the count source and 1/128 as the clock frequency division ratio. For a two-channel AGT, the maximum period is approximately 16,777,216 seconds (approximately 194 days).

In this application project, a two-channel AGT is used because it is necessary to generate a periodic interrupt every 30 minutes as a transition request to Snooze mode and the timing of data acquisition.

2.7 Two-Channel AGTW and IRQ2 in RA2E2

RA2E2 MCU implements a 32-bit Asynchronous General Purpose Timer (AGTWn) with two channels (n=0, 1). The AGTW 0 is used as a periodic timer with pin output enabled (looped with IRQ2 pin, which acts as a wake source from standby mode) to generate a long-period interrupt (Figure 26). With the AGTO0 signal connected to the external interrupt 2 (IRQ2) through the jumper wire, the IRQ2 interrupt acts as a wake-up source from the software standby mode to output the stored data at regular intervals of 24 hours. Refer to the relevant part of the RA2E2 User's Manual (R01UH0919) for more information.

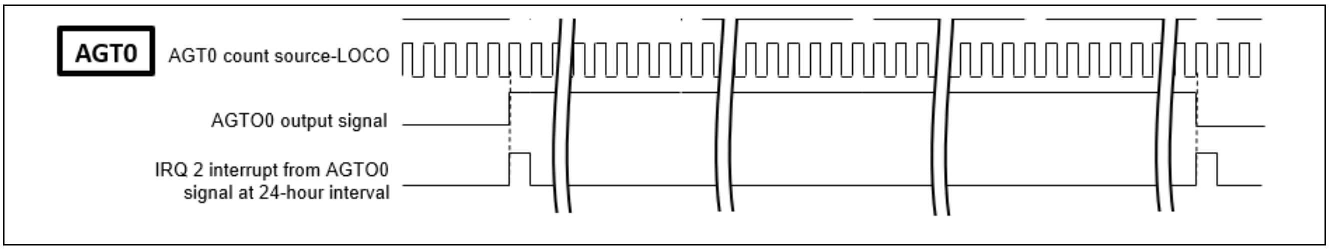


Figure 26. AGTW0 Timer and IRQ2 for Periodic Data Output

The RA2E2 AGTW1 32-bit single-channel timer is used to generate the snooze request through the AGT1 underflow event at 30-minute intervals, as shown in Figure 27.

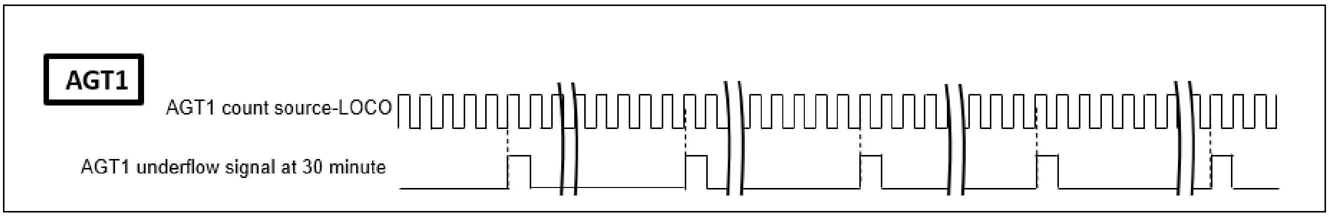
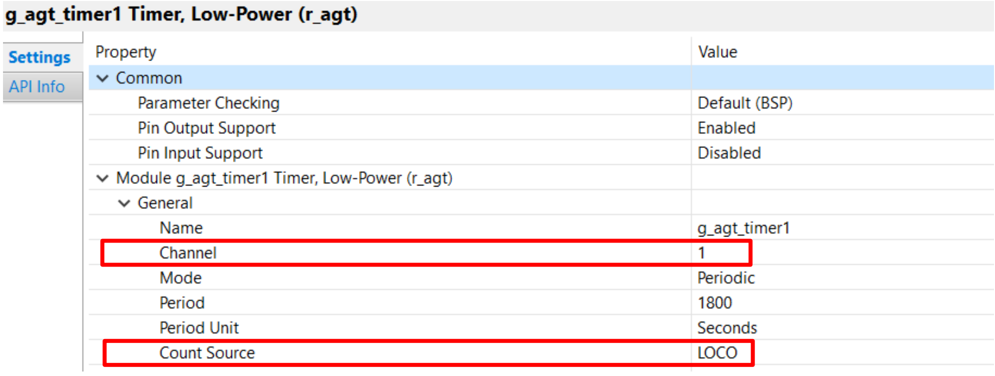


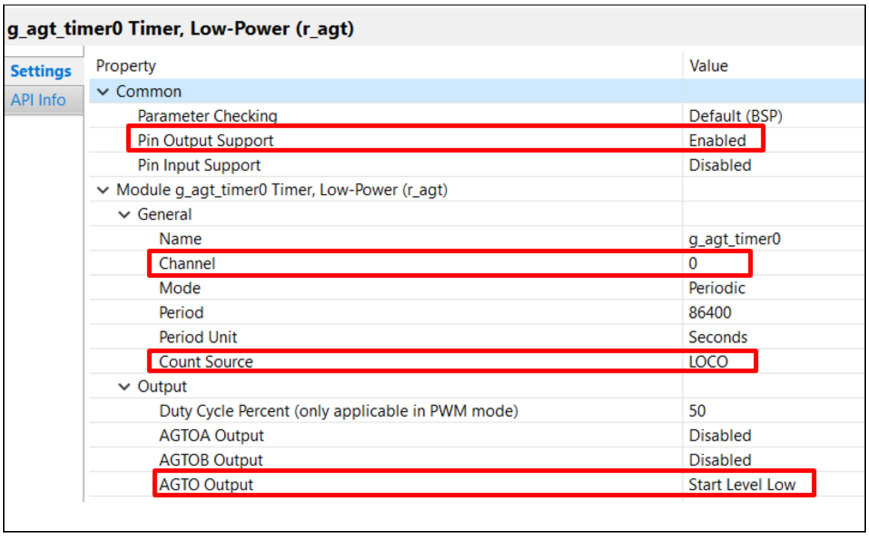
Figure 27. AGTW1 Timer for Snooze Request

Figure 28 and Figure 29 show AGT1 and AGT0 count source settings in the FSP configurator.


g_agt_timer1 Timer, Low-Power (r_agt)

Property	Value
⌄ Common	
Parameter Checking	Default (BSP)
Pin Output Support	Enabled
Pin Input Support	Disabled
⌄ Module g_agt_timer1 Timer, Low-Power (r_agt)	
⌄ General	
Name	g_agt_timer1
Channel	1
Mode	Periodic
Period	1800
Period Unit	Seconds
Count Source	LOCO

Figure 28. AGT1 Count Source Settings in FSP Configurator


g_agt_timer0 Timer, Low-Power (r_agt)

Property	Value
⌄ Common	
Parameter Checking	Default (BSP)
Pin Output Support	Enabled
Pin Input Support	Disabled
⌄ Module g_agt_timer0 Timer, Low-Power (r_agt)	
⌄ General	
Name	g_agt_timer0
Channel	0
Mode	Periodic
Period	86400
Period Unit	Seconds
Count Source	LOCO
⌄ Output	
Duty Cycle Percent (only applicable in PWM mode)	50
AGTOA Output	Disabled
AGTOB Output	Disabled
AGTO Output	Start Level Low

Figure 29. AGTW0 Count Source, Pin Output Settings in FSP Configurator

2.8 RTC Mode Selection for RA2E1

The RTC has two operation modes and two count modes.

Operation modes include normal operation mode and low-power consumption clock mode. Some features are limited to low-power modes. For more information, see Table “RTC specifications” in RA2E1 User's Manual (R01UH0852).

RCR4.ROPSEL can be used to select the operating mode. In this application project, the normal operation mode is selected to use the RTC alarm interrupt (RCR4.ROPSEL is set to 0).

There are two counting modes: calendar count mode and binary count mode. The calendar count mode retains the 100-year calendar from 2000 to 2099 and automatically corrects the leap year date. In binary count mode, seconds are counted, and the information is saved as a serial value. In this application project, the binary count mode is selected to measure the time (elapsed seconds) since startup.

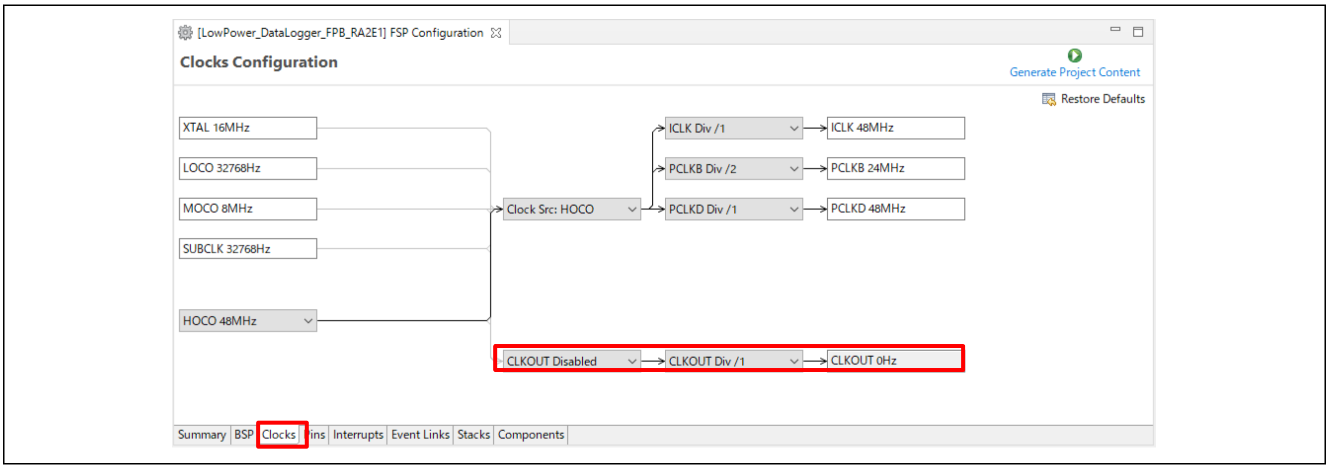
The count mode can be selected by using the RCR2.CNTMD bit. When this bit is rewritten, the next process is performed after confirming that the value has been rewritten. If the count mode is set again, the RTC software reset is executed, and the initial settings are redone. This bit is updated synchronously with the count source, but the count mode is switched after an RTC software reset.

2.9 Clock Output

This application project allows checking the status of the low-power mode by outputting the clock externally. This is possible because some oscillators stop in Software Standby mode.

The clock sources output from CLKOUT pins and their division factors are set by the registers below. To enable outputting from the CLKOUT pin, set CKOCR.CKOEN to 1. The FSP configurator sets the clock output from the **Clocks** tab, as shown in Figure 30.

- CKOCR.CKODIV[2:0], or CKOCR.CKOSEL [2:0]
- OFS1.HOCOFRQ1[2:0]


[LowPower_DataLogger_FPB_RA2E1] FSP Configuration

Clocks Configuration

XTAL 16MHz
LOCO 32768Hz
MOCO 8MHz
SUBCLK 32768Hz
HOCO 48MHz
Clock Src: HOCO
ICLK Div /1 → ICLK 48MHz
PCLKB Div /2 → PCLKB 24MHz
PCLKD Div /1 → PCLKD 48MHz
CLKOUT Disabled → CLKOUT Div /1 → CLKOUT 0Hz

Figure 30. Clock Output Settings in FSP Configurator

3. Low-Power Data Logger Application

This chapter describes the detailed design of the low-power data logger application using the module features described in the previous chapter.

3.1 Functional Specification

This application project is implemented based on the following specifications. Figure 31 also shows the overall algorithm for the low-power data logger.

3.1.1 Data Acquisition Function

- Analog signals of two sensors (Illuminance sensor Seeed Grove-Luminance Sensor, Temperature sensor Seeed Grove-Temperature Sensor) connected to external pins are acquired by A/D converters.
- Analog signal input values are between the low-potential reference voltage (VREFL and VSS) and the high-potential reference voltage (VREFH and VCC) set between 0 V and 3.3 V, respectively.
- Acquisition of data from each sensor is performed every 30 minutes regardless of the low-power mode status (using Snooze request event or AGT1 underflow interrupt)
- The size of the measurement data buffer is 96 samples (48 hours of sampling every 30 minutes).

3.1.2 Data Level Judgement Function

- After acquiring the data, the analog signal pin AN001(For FPB-RA2E1 application) or AN021(For FPB-RA2E2 application) judges the level by DOC.
- When AN001(For FPB-RA2E1 application) or AN021(For FPB-RA2E2 application) is equal to or greater than the threshold (30.00°C), the low-power mode is canceled, and data is output.

3.1.3 Data Output Function

- Perform data output every 24 hours triggered by RTC alarm interrupt (For FPB-RA2E1 application) or external IRQ interrupt generated by AGT0 signal loopback (For FPB-RA2E2 application), external IRQ interrupt generated by user switch and DOC interrupt.
- UART communication format is 115,200 bps baud rate, 8-bit data length, no parity, 1 stop bit.

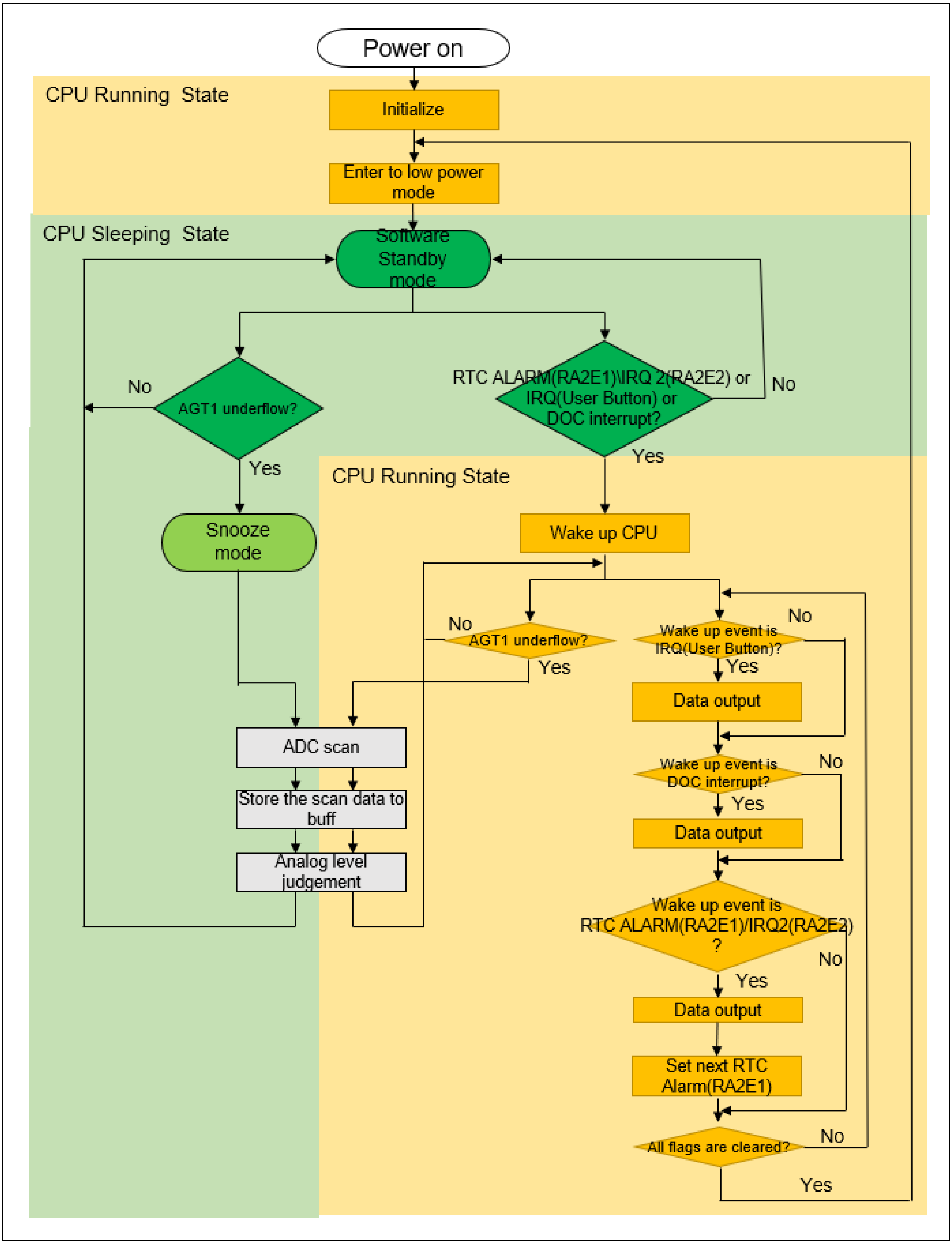


Figure 31. Overall Algorithm of the Low Power Data Logger

3.2 User Interface

This section describes the user interface for the Low Power Data Logger Application. See chapter 4.3 "Procedure for Checking the Operation of the Low Power Data Logger" for more information on checking operations using the main module.

Figure 32 and Figure 33 shows the system overview of this application project.

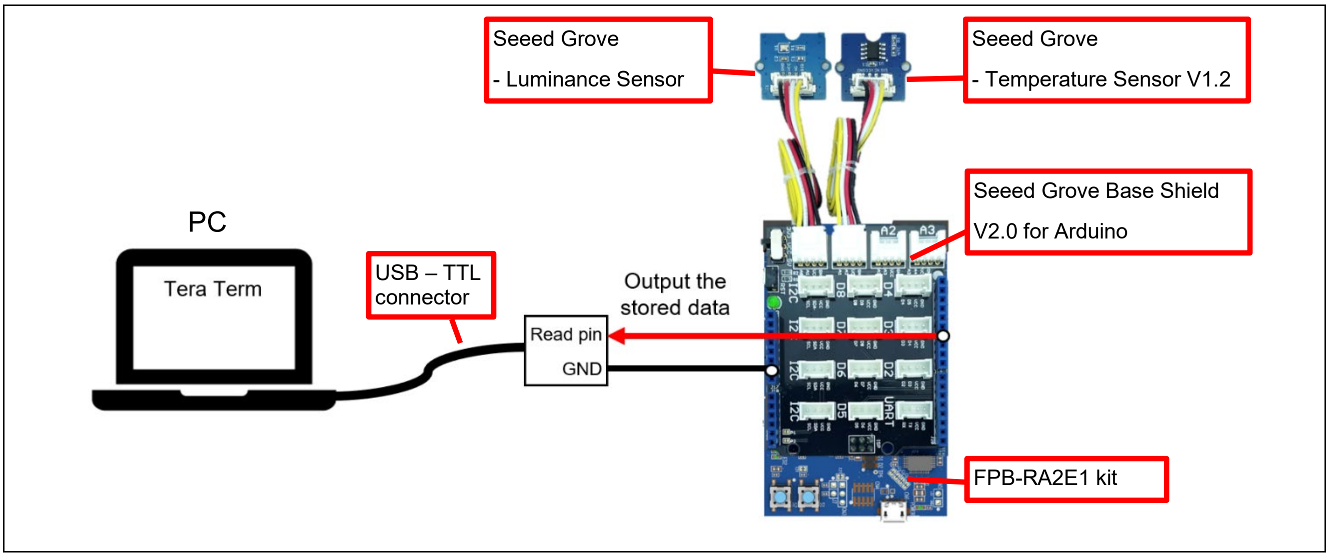


Figure 32. FPB-RA2E1 System Overview

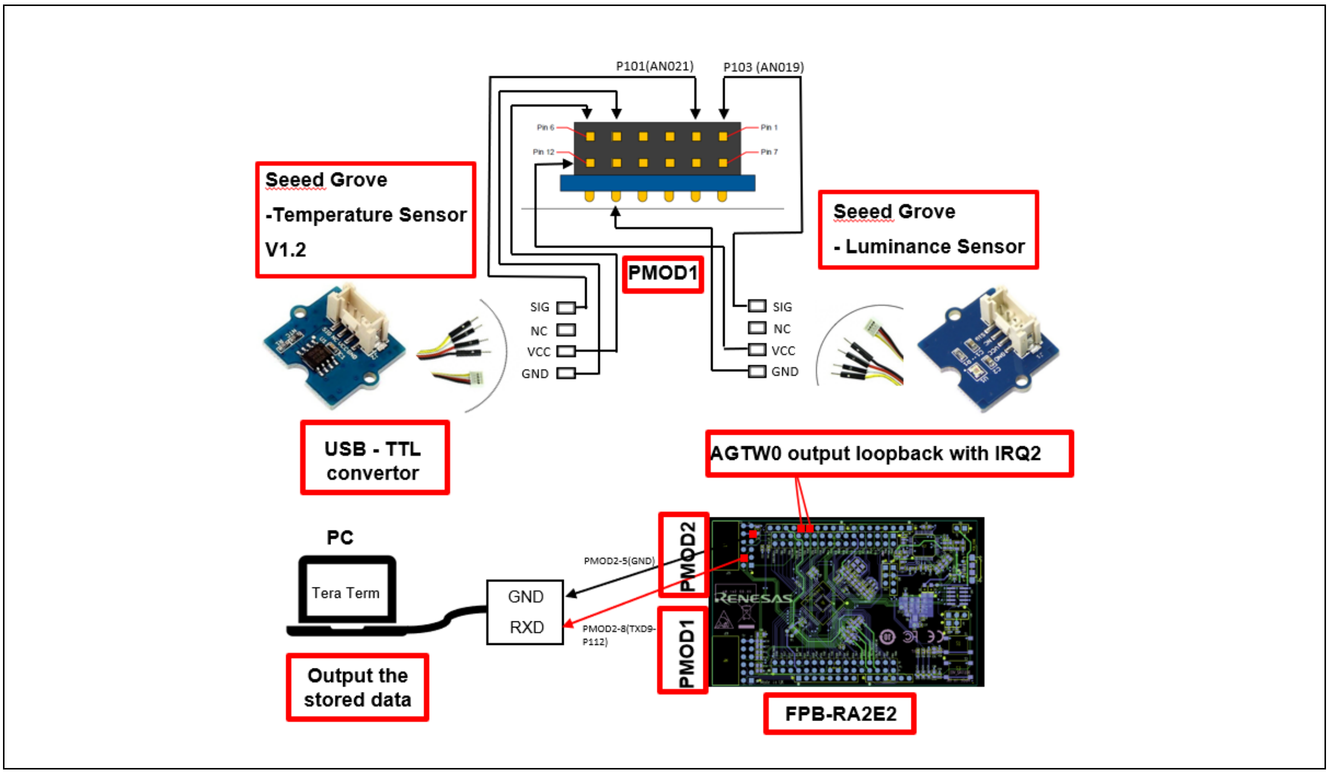


Figure 33. FPB-RA2E2 System Overview

3.2.1 Connecting the Sensor

In the FPB-RA2E1 application project, attach Seeed's Grove Base Shield V2.0 for the Arduino-to-Arduino compatible connector of the FPB-RA2E1 kit. Attach the sensor module to the A0 and A1 connectors of this Base Shield with Grove cable. This connects the analog signal pins (AN000 and AN001) of RA2E1 MCU with the analog output pins of the two sensors. Refer to Figure 34.

In the FPB-RA2E2 application project, the PMOD1 connector of the FPB-RA2E2 kit connects the sensor module to the analog signal pins (AN019 and AN021) with the analog output pins of the two sensors using jumper wires. Refer to Figure 35.

- Light Sensor Module: Seeed Grove – Luminance Sensor
  Equipped with illuminance sensor APDS-9002, Operating voltage: 2.4–5V, Measurement range: 0–1000Lux
- Temperature Sensor Module: Seeed Grove – Temperature Sensor
  Equipped with NTC thermistor NCP18WF104F03RC, Operating voltage: 3.3–5V, Operating temperature range: -40 °C – 125°C

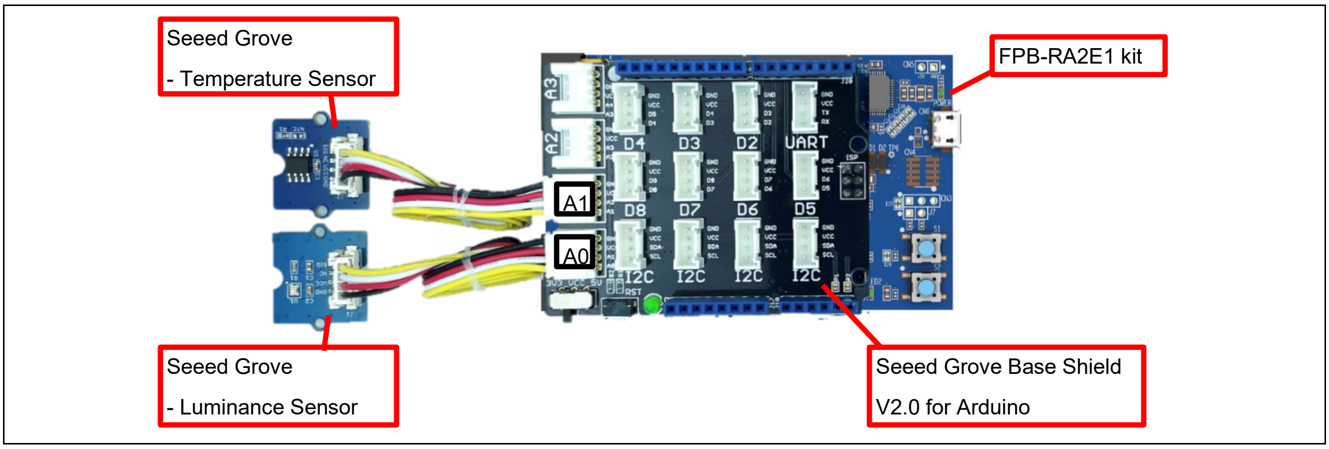


Figure 34. FPB-RA2E1 Sensor Connection

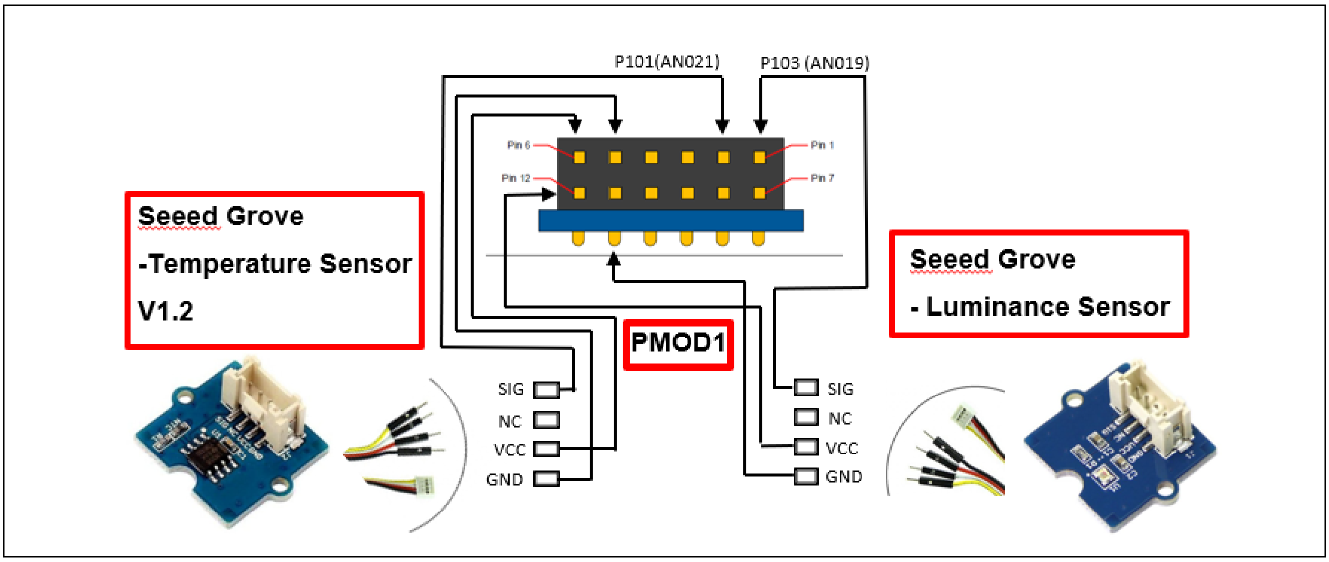


Figure 35. FPB-RA2E2 Sensor Connection

3.2.2 Connect the USB-TTL Connector

Table 4 and Figure 36 show the connection pins to the USB-TTL connector. Connect the Arduino-compatible connector of the FPB-RA2E1 kit using jumper wires.

Table 4. Pins to be Connected to the USB-TTL Connector (FPB-RA2E1 kit)

RA2E1 Pin	FPB-RA2E1 Pin	Typical Uses
P101/TxD0	J1-4	Serial communication (transmission). Connect to the RXD pin of the USB-TTL connector.
VSS	GND	Ground. Connect to the GND pin of the USB-TTL connector.

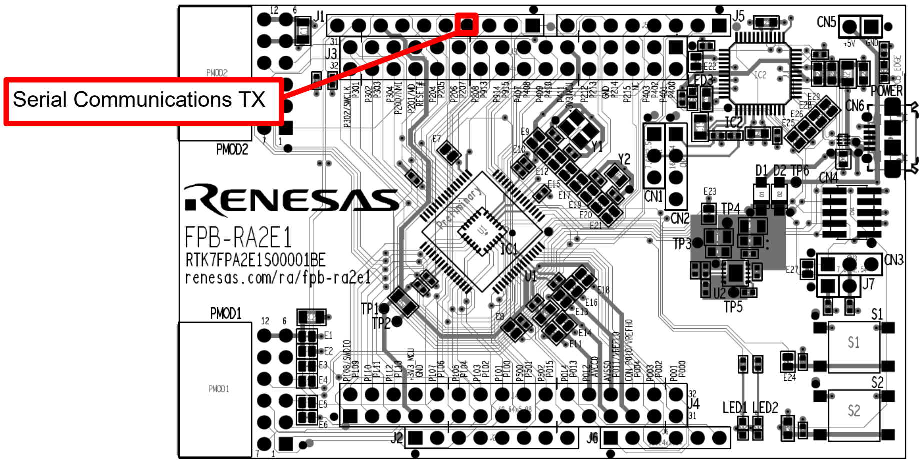


Figure 36. FPB-RA2E1 Connecting to the External Equipment

Table 5 and Figure 37 show the connection pins to the USB-TTL connector. Connect the Arduino-compatible connector of the FPB-RA2E2 kit using jumper wires.

Table 5. Pins to be Connected to the USB-TTL Connector (FPB-RA2E2 kit)

RA2E2 Pin	FPB-RA2E2 Pin	Typical Uses
P112/TxD9_J	PMOD2-8	Serial communication (transmission). Connect to the RXD pin of the USB-TTL connector.
VSS	GND	Ground Connect to the GND pin of the USB-TTL connector.

3.2.3 Hardware Setup for AGTW Signal Loopback (For FPB-RA2E2 Application)

Table 6 and Figure 37 show the connection pins to loop back the AGTW0 output pin P102 and external IRQ pin P100.

Table 6. Pins to be Connected for AGTW0 signal loop-back (FPB-RA2E2 kit)

RA2E2 Pin	FPB-RA2E2 Pin	Typical Uses
P102 (AGTW0) loopback with P100 (IRQ2)	J1-6 and J1-5	AGTW0 output pin loopback with IRQ2 pin for periodic wake-up.

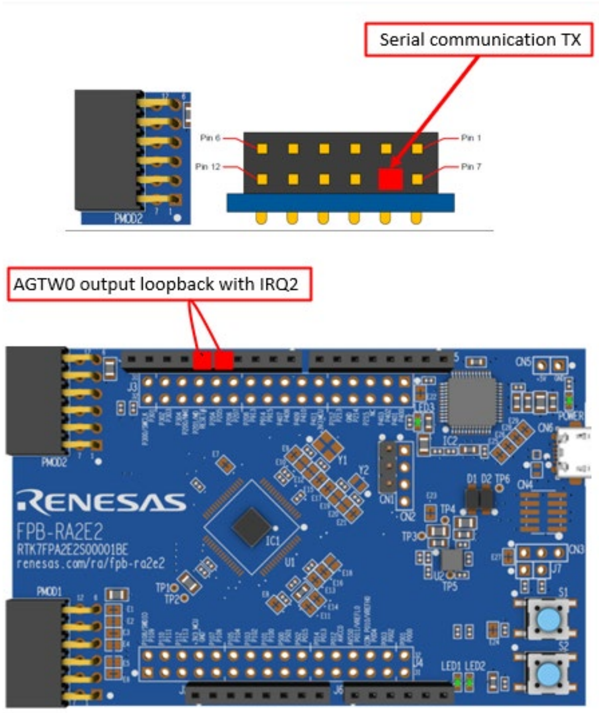


Figure 37. FPB-RA2E2 Connecting to the External Equipment

3.2.4 Data Communication Specifications

This application project performs data communication according to the following procedure and format.

3.2.4.1 Communication Data Format

- Transmission Format
  The following is the packet format/protocol of data transmitted by the FPB-RA2E1 and FPB-RA2E2.
  Packet format:

(a) Start code	(b) Attribute code	(c) Delimiter code	(d) Data length code	(c) Delimiter code	(e) Data	(f) End code	(g) Linefeed code

Description	Code	Length	Function
(a) Start code	'^'	1 byte	Start of the packet
(b) Attribute code	ASCII Code	1 byte	Represent a data output occurrence event '1': RTC alarm event or IRQ 2 event '2': External IRQ event '3': DOC interrupt event
(c) Delimiter code	‘,’	1 byte	Code delimiting code
(d) Data length code	ASCII Code	4 bytes	Indicates the length of the data section.
(e) Data	ASCII Code	Number of bytes specified in (d)	See the following Data Format
(f) End code	'$'	1 byte	End of the packet
(g) Linefeed code	‘\n’	1 byte	Linefeed

(e) Data format

The format of the data part varies depending on the attribute code, as shown below.

- Attribute code 1/2 (output data by RTC alarm\IRQ 2 and external IRQ interrupt)
  Format of data part:

Time data	‘,’	Number of data	','	First data	','	Second data	','	・・・	','	Last data

Light Data	'/'	Temperature Data

- Attribute code 3 (output data by DOC interrupt)
  Format of data part:

Time data	','	Light data	'/'	Temperature data

The data comprising the format of attribute codes 1 to 3 is generated as follows.

- The data count consists of 0 to 9 ASCII codes. The length is 2 bytes. Example: When the system transmits 10 data, it sets 0x31, 0x30.
- The time/light/temperature data consists of 0 to 9 ASCII codes. Example: When the system transmits a time value “1234”, it sets 0x31, 0x32, 0x33, 0x34.
- The time data is composed of 10-digit decimal numbers representing elapsed time.
- The light and temperature data are composed of a 4-digit decimal number representing ADC read value. When the system transmits data “2000”, it sets 0x32, 0x30, 0x30, 0x30.
- The delimiter between the code is ','. The delimiter between the time and light data is '/'.

Note: Time data and preceding delimiter code are not available in the protocol format of the FPB-RA2E2 application as it does not have an RTC module.

3.2.5 Input/Output Pins for Debugging (FPB-RA2E1 and FPB-RA2E2 kit)

Table 7 and Figure 38 show the I/O pins for debugging this application project. The debug pins can be observed and toggled as shown below to cancel the low-power modes and check the status of the modes.

Table 7. Debugging I/O Pins (FPB-RA2E1/FPB-RA2E2 kit)

RA2E1/RA2E2 Pin	FPB-RA2E1/FPB-RA2E2 Pins or Connecting Components	Typical Uses
P109/CLKOUT	J5-2	Clock output
P914	LED2	Turn on in Normal mode
P015	LED1	Error condition turn on when API call failed
P205/IRQ1	User switch S1	Manual cancellation of low-power mode

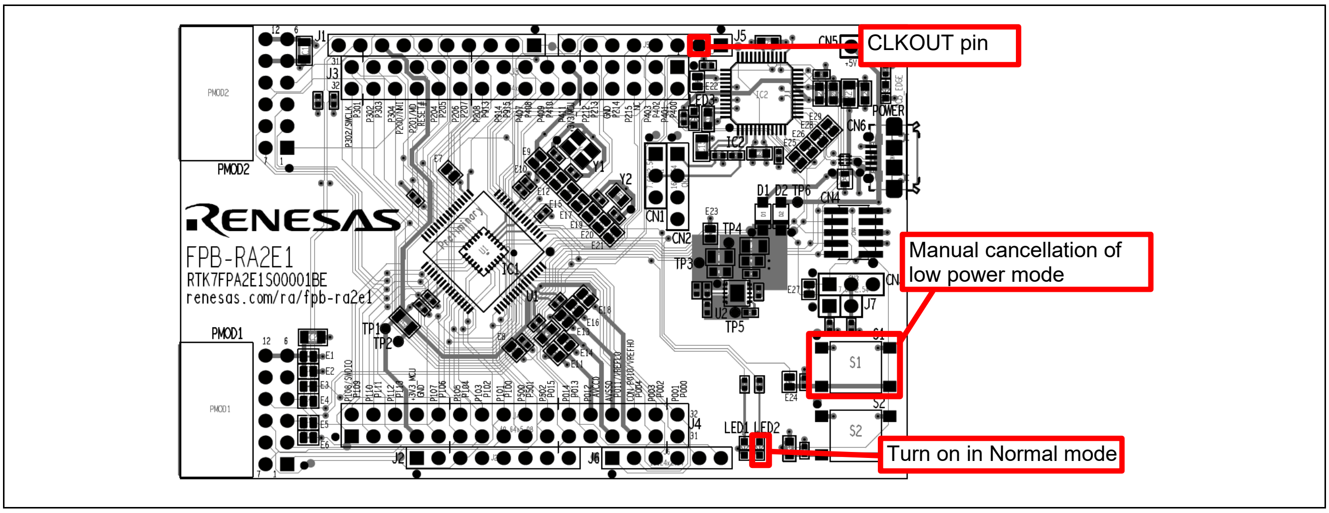


Figure 38. Debugging I/O Pins

3.3 Clock Input (For RA2E1)

The clock from the external oscillator must be input so that the RTC or AGT can perform an accurate time count. FPB-RA2E1 kit implements a 32.768-kHz clock crystal, but by default, it is not connected to the RA2E1 MCU by solder jumpers (E19 and E20). Therefore, the project attached to this application note selects an internal clock (LOCO) as the count source for AGT0 and RTC.

When accurate time counting is required, connect the external oscillator using solder bridges.

Also, the following settings should be changed:

- Change the sub-clock populating setting to “Populated” in the **BSP** tab in the FSP configurator.
- Change the count source for AGT0 to “SUBCLOCK” in the **Stacks** tab in the FSP configurator.
- Change the count source for RTC to the sub-clock in `rtc_initialize()` in `app_common.c`.

Note: RA2E2 MCU does not support an external oscillator.

Figure 39 and Figure 40 show the positions of the solder jumpers for external oscillator connections.

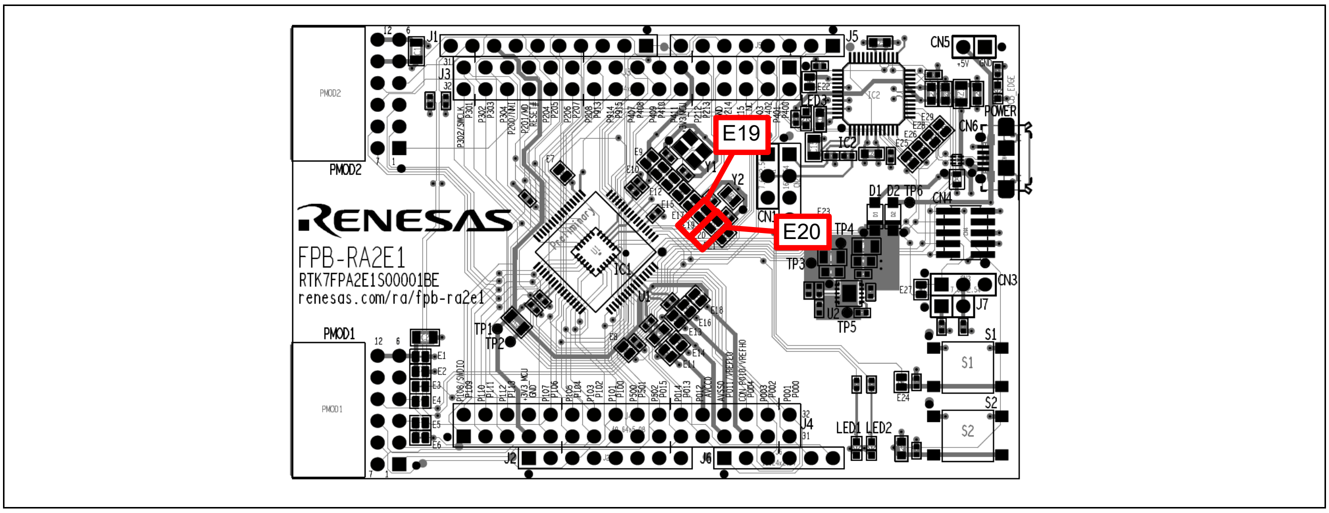


Figure 39. Solder Jumper for External Oscillator Connection

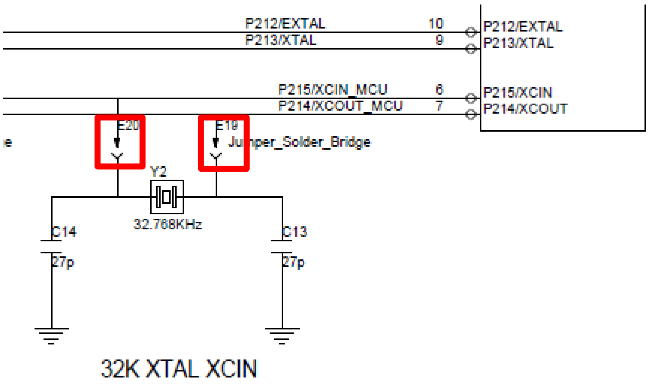


Figure 40. Solder Jumper for External Oscillator Connection (Circuit Diagram)

3.4 Debugging

This application project implements a pin input/output function to check the low-power mode status, fast cycle debug function, and disabling sensor-dependent process function.

3.4.1 Canceling the Low Power Mode by Pressing Down the User Switch

The user switches S1 of FPB-RA2E1 or FPB-RA2E2, which can be pressed to generate an IRQ1 interrupt and cancel the low-power mode. Setting a breakpoint after canceling stops the program being executed, and each register and data can be checked.

3.4.2 Checking the Low Power Mode Status

The low-power mode status can be checked by observing LED2 and CLKOUT pins.

When LED2 is turned on, the mode is Normal. In Software Standby mode, HOCO, MOCO, and MOSC clocks are stopped. Therefore, when these clocks are set as the clock output source, the CLKOUT pin retains either a High or Low state. When toggling out, the mode is changed to Normal mode or the low-power mode other than the Software Standby mode. **When the debugger is connected, the clock does not stop even if Software Standby mode is entered. Disconnecting the debugger is required to check the mode status.**

In addition, port P109 (CLKOUT pin) of the FPB-RA2E1 or FPB-RA2E2 kit is connected to the onboard E2 emulator by default. If this pin is used for clock output, disconnect solder jumpers (E25 and E29) as needed.

Figure 43 shows the position of the solder jumper for connecting the P109 (CLKOUT pin) to the onboard E2 emulator.

Table 8 lists the pin output states in each low-power mode and Figure 41 and Figure 42 show examples of pin outputs in each low-power mode.

Table 8. Pin Output States in each Low Power Mode

Low-Power Modes	P109/CLKOUT	P914/LED2
Normal mode	Toggle	High
Software Standby mode	High or Low	Low
Snooze mode	Toggle	Low

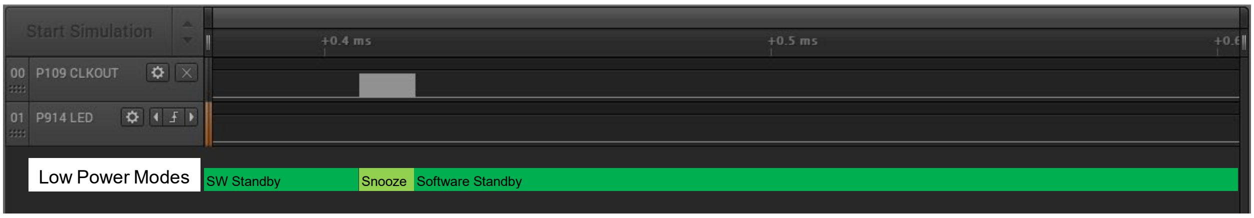


Figure 41. Example of Pin Output in Snooze Mode

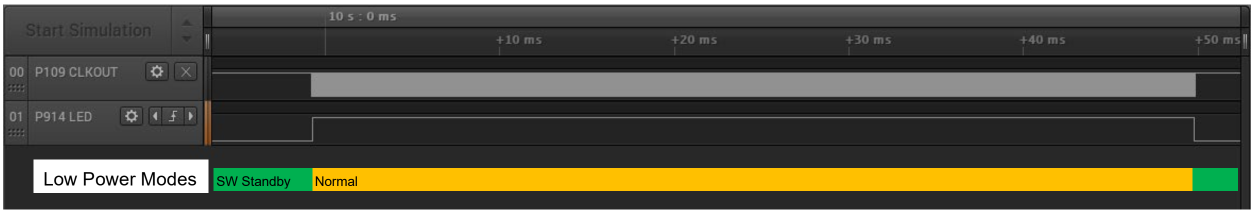


Figure 42. Example of Pin Output in Normal Mode

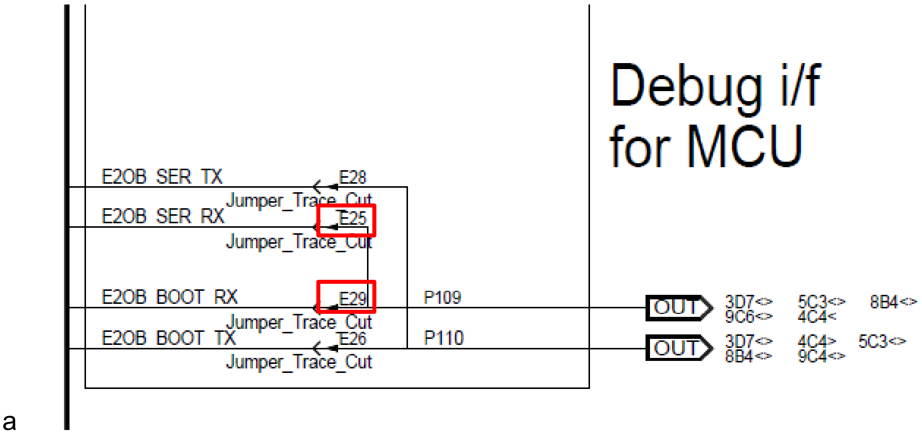


Figure 43. Solder Jumper for Connecting P109 to Onboard E2 Emulator

3.4.2.1 Fast Cycle Debug

This sample project implements a function to speed up the execution cycle of data acquisition and data processing to check the operation in a short time. To enable this feature, define the `DEBUG_FAST_CYCLE` macro in the `app_common.h`. When this `DEBUG_FAST_CYCLE` macro is defined, the following process is enabled: the data acquisition period is set every 30 seconds, and the data processing period is set every 24 minutes.

- Resetting the AGT0 timer count (AGT)
- Changing the Added Value of the RTC Alarm Setting

Note: In the FPB-RA2E2 application, AGTW0 at 24-minute and AGTW1 at 30-second intervals are configured with `DEBUG_FAST_CYCLE` mode.

3.5 Flowchart

This section describes the flowchart of the Low Power Data Logger Application for the case of FPB-RA2E1. For the FPB-RA2E2 application, please refer to the source code.

Figure 44 shows the overall flow of the application project.

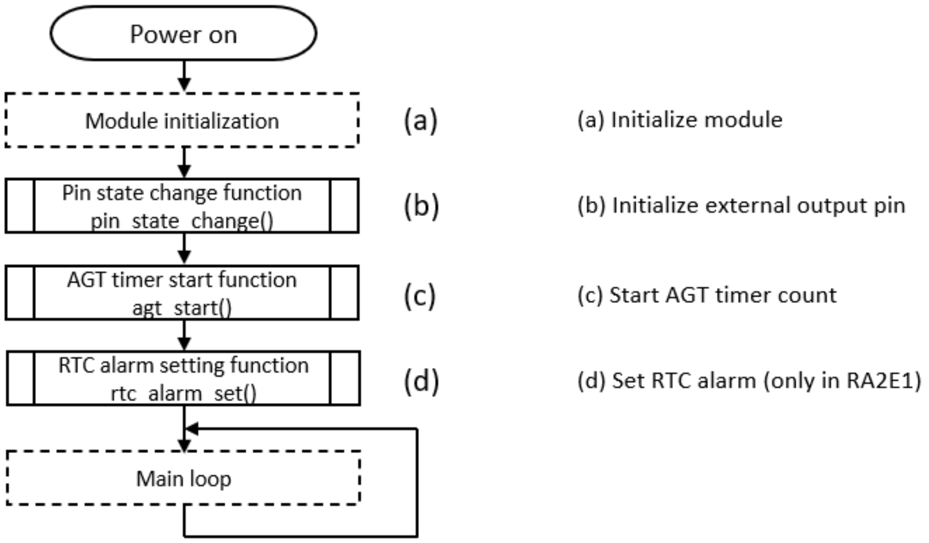

Power on
Module initialization
(a)
(a) Initialize module
Pin state change function
pin_state_change()
(b)
(b) Initialize external output pin
AGT timer start function
agt_start()
(c)
(c) Start AGT timer count
RTC alarm setting function
rtc_alarm_set()
(d)
(d) Set RTC alarm (only in RA2E1)
Main loop

Figure 44. Overall Flow

Figure 45 shows the module initialization processing flow.

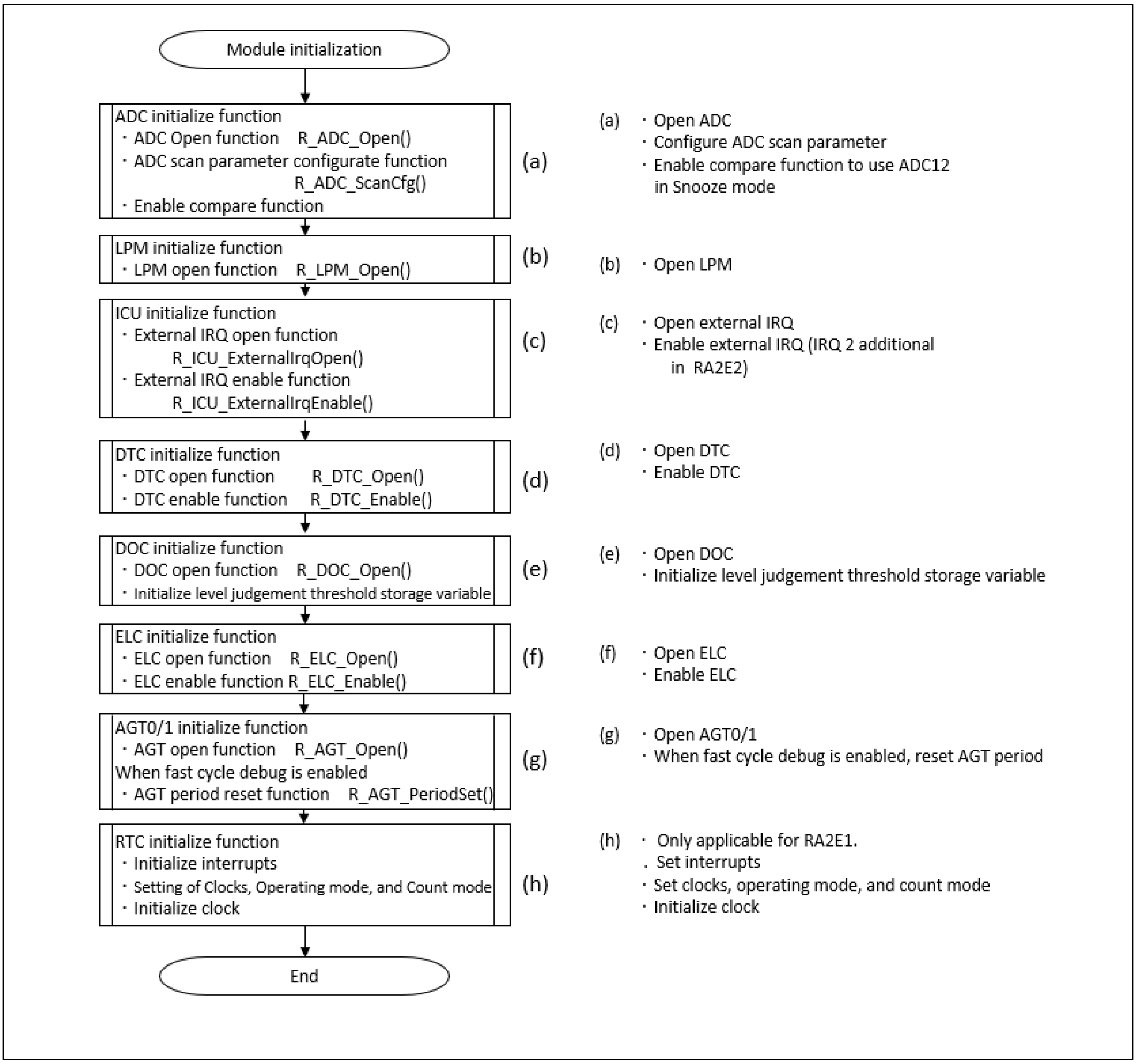

Module initialization
ADC initialize function
・ADC Open function R_ADC_Open()
・ADC scan parameter configurate function R_ADC_ScanCfg()
・Enable compare function
(a)
LPM initialize function
・LPM open function R_LPM_Open()
(b)
ICU initialize function
・External IRQ open function R_ICU_ExternalIrqOpen()
・External IRQ enable function R_ICU_ExternalIrqEnable()
(c)
DTC initialize function
・DTC open function R_DTC_Open()
・DTC enable function R_DTC_Enable()
(d)
DOC initialize function
・DOC open function R_DOC_Open()
・Initialize level judgement threshold storage variable
(e)
ELC initialize function
・ELC open function R_ELC_Open()
・ELC enable function R_ELC_Enable()
(f)
AGT0/1 initialize function
・AGT open function R_AGT_Open()
When fast cycle debug is enabled
・AGT period reset function R_AGT_PeriodSet()
(g)
RTC initialize function
・Initialize interrupts
・Setting of Clocks, Operating mode, and Count mode
・Initialize clock
(h)
End
(a) ・Open ADC
・Configure ADC scan parameter
・Enable compare function to use ADC12 in Snooze mode
(b) ・Open LPM
(c) ・Open external IRQ
・Enable external IRQ (IRQ 2 additional in RA2E2)
(d) ・Open DTC
・Enable DTC
(e) ・Open DOC
・Initialize level judgement threshold storage variable
(f) ・Open ELC
・Enable ELC
(g) ・Open AGT0/1
・When fast cycle debug is enabled, reset AGT period
(h) ・ Only applicable for RA2E1.
. Set interrupts
・Set clocks, operating mode, and count mode
・Initialize clock

Figure 45. Module Initialization Processing

Figure 46 shows the main loop processing flow.

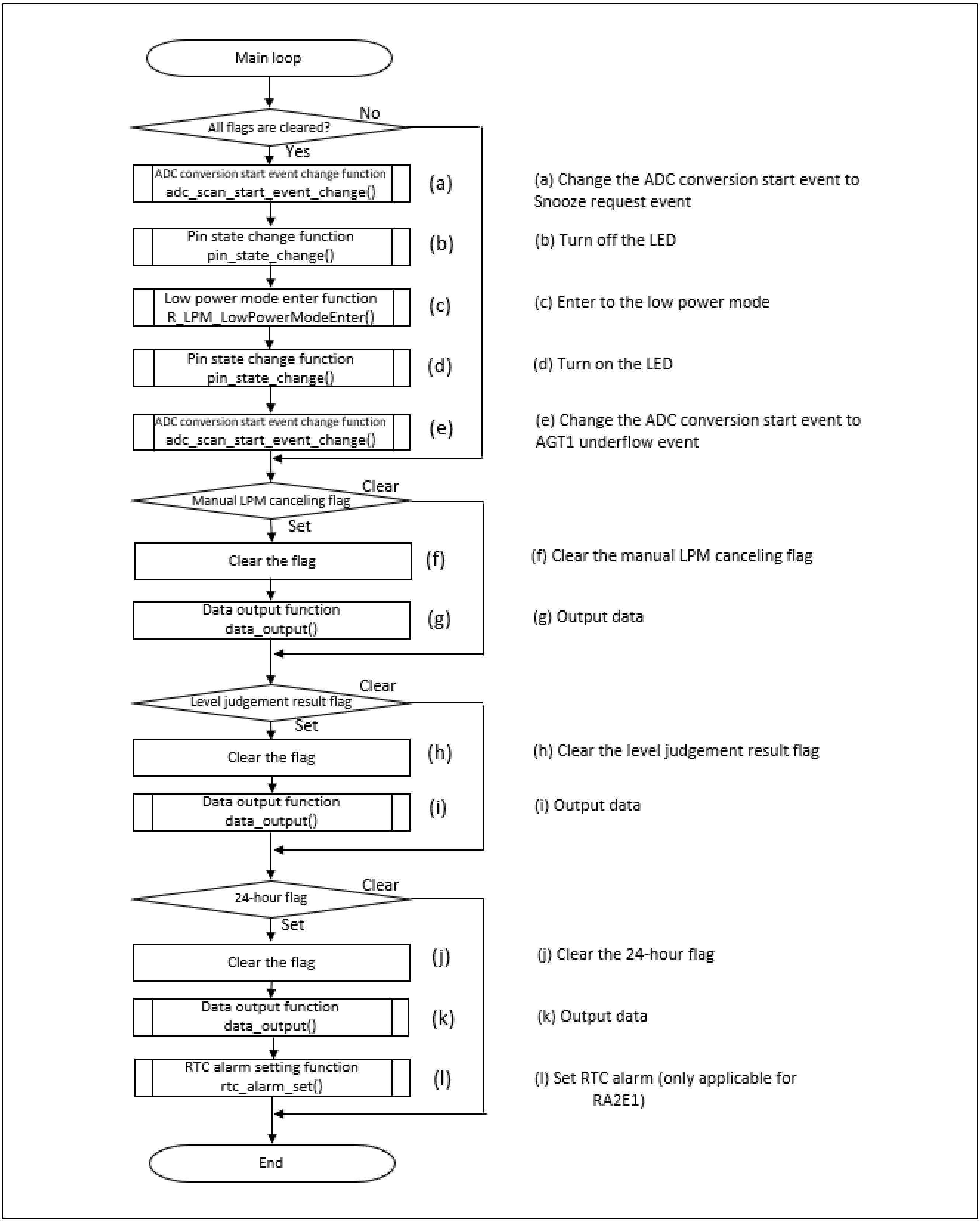

Main loop
All flags are cleared?
No
Yes
ADC conversion start event change function
adc_scan_start_event_change()
(a)
(a) Change the ADC conversion start event to Snooze request event
Pin state change function
pin_state_change()
(b)
(b) Turn off the LED
Low power mode enter function
R_LPM_LowPowerModeEnter()
(c)
(c) Enter to the low power mode
Pin state change function
pin_state_change()
(d)
(d) Turn on the LED
ADC conversion start event change function
adc_scan_start_event_change()
(e)
(e) Change the ADC conversion start event to AGT1 underflow event
Manual LPM canceling flag
Clear
Set
Clear the flag
(f)
(f) Clear the manual LPM canceling flag
Data output function
data_output()
(g)
(g) Output data
Level judgement result flag
Clear
Set
Clear the flag
(h)
(h) Clear the level judgement result flag
Data output function
data_output()
(i)
(i) Output data
24-hour flag
Clear
Set
Clear the flag
(j)
(j) Clear the 24-hour flag
Data output function
data_output()
(k)
(k) Output data
RTC alarm setting function
rtc_alarm_set()
(l)
(l) Set RTC alarm (only applicable for RA2E1)
End

Figure 46. Main Loop Processing

Figure 47 shows the flow of RTC alarm interrupt processing.

Note: RTC alarm interrupt processing is only applicable for RA2E1.

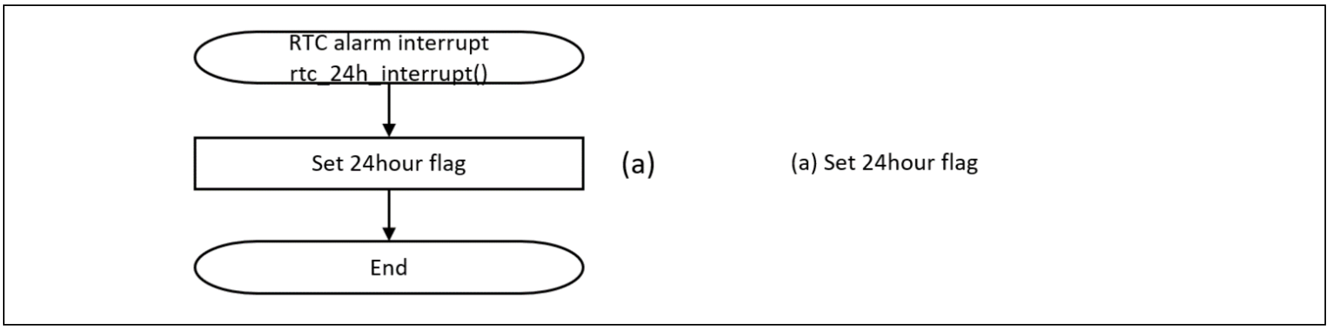


Figure 47. RTC Alarm Interrupt Processing

Figure 48 shows the flowchart for RA2E2 IRQ 2 interrupt processing

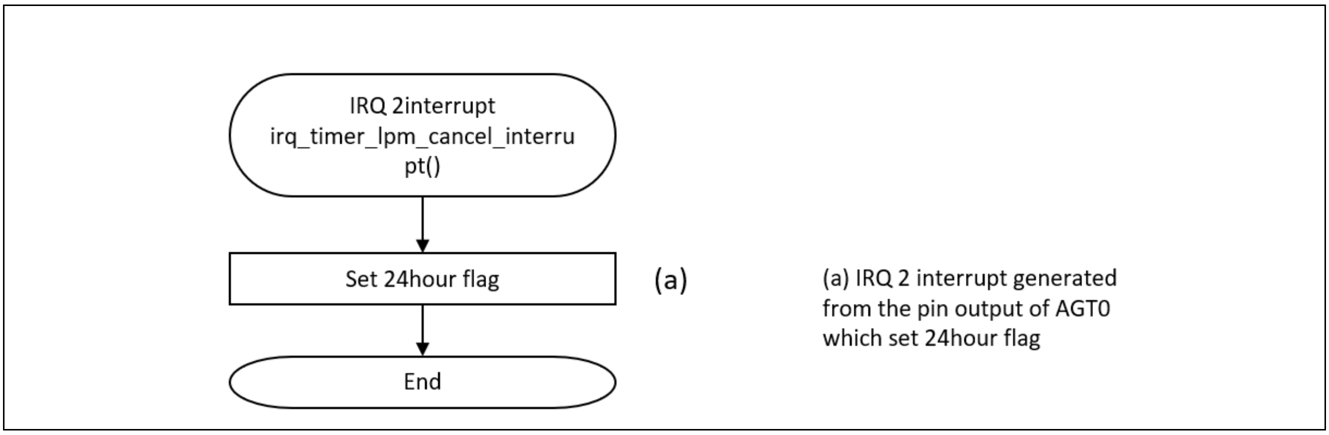

IRQ 2interrupt
irq_timer_lpm_cancel_interru
pt()
Set 24hour flag
(a)
End
(a) IRQ 2 interrupt generated
from the pin output of AGT0
which set 24hour flag

Figure 48. RA2E2 IRQ 2 Interrupt Processing

Figure 49 shows the flowchart for IRQ interrupt processing.

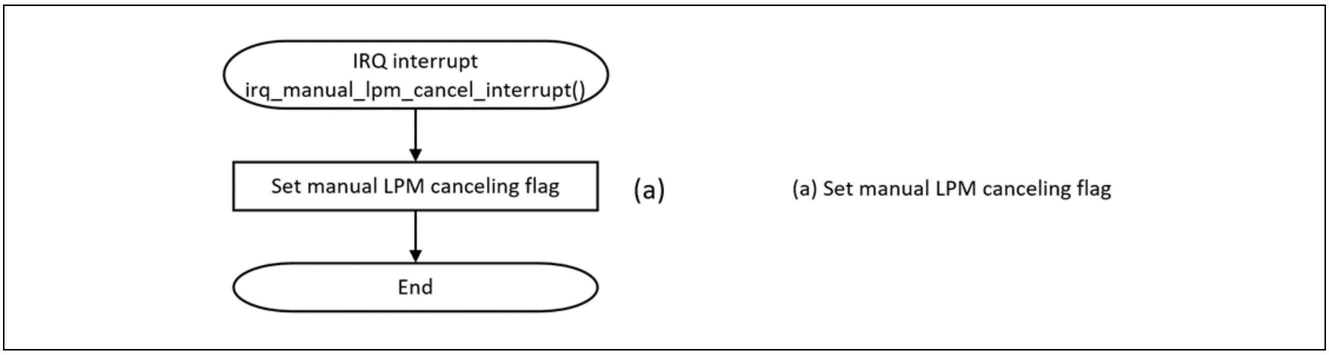

IRQ interrupt
irq_manual_lpm_cancel_interrupt()
Set manual LPM canceling flag
(a)
End
(a) Set manual LPM canceling flag

Figure 49. IRQ Interrupt Processing

Figure 50 shows the flowchart for DOC interrupt processing.

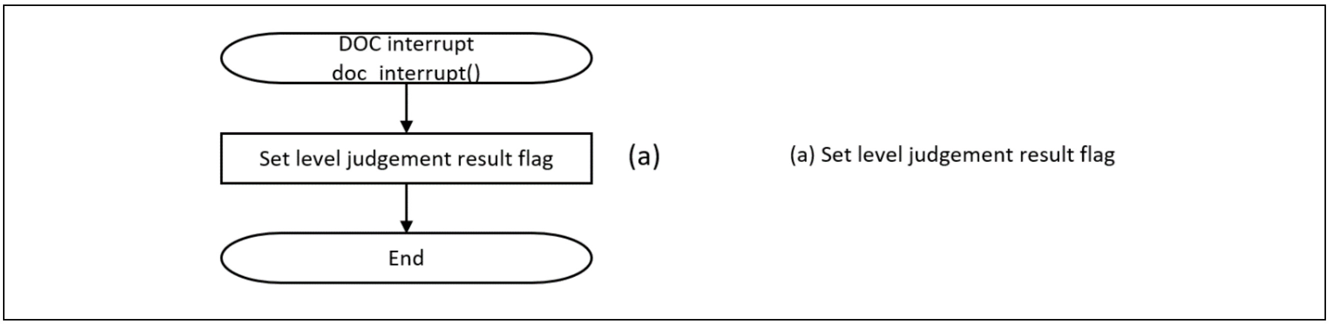

DOC interrupt
doc_interrupt()
Set level judgement result flag
(a)
End
(a) Set level judgement result flag

Figure 50. DOC Interrupt Processing

Figure 51 shows the SCI interrupt processing flow.

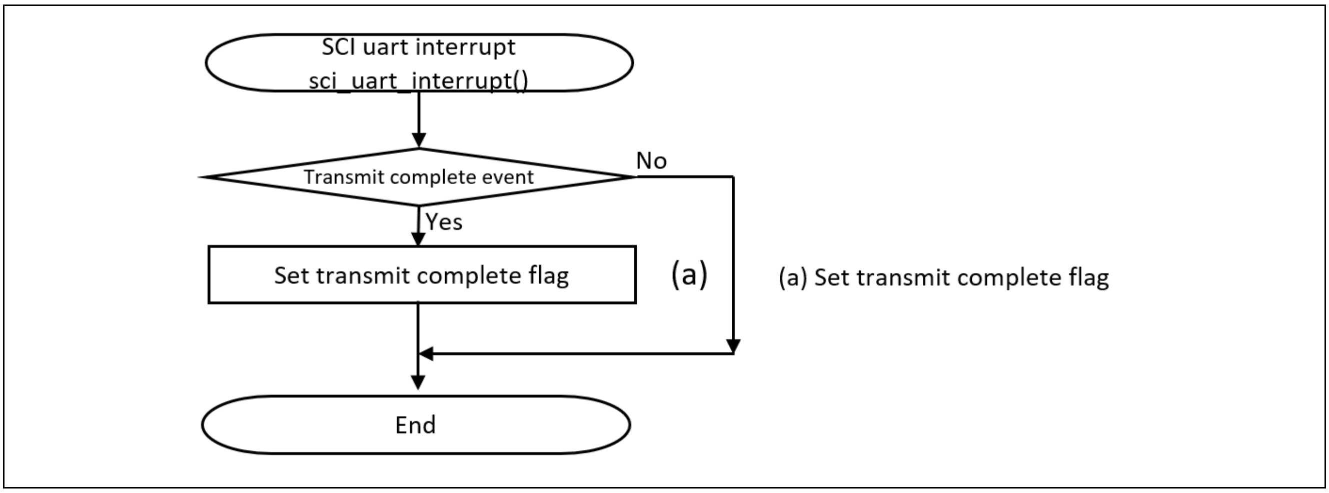


Figure 51. SCI Interrupt Processing

Figure 52 shows the flow of data processing.

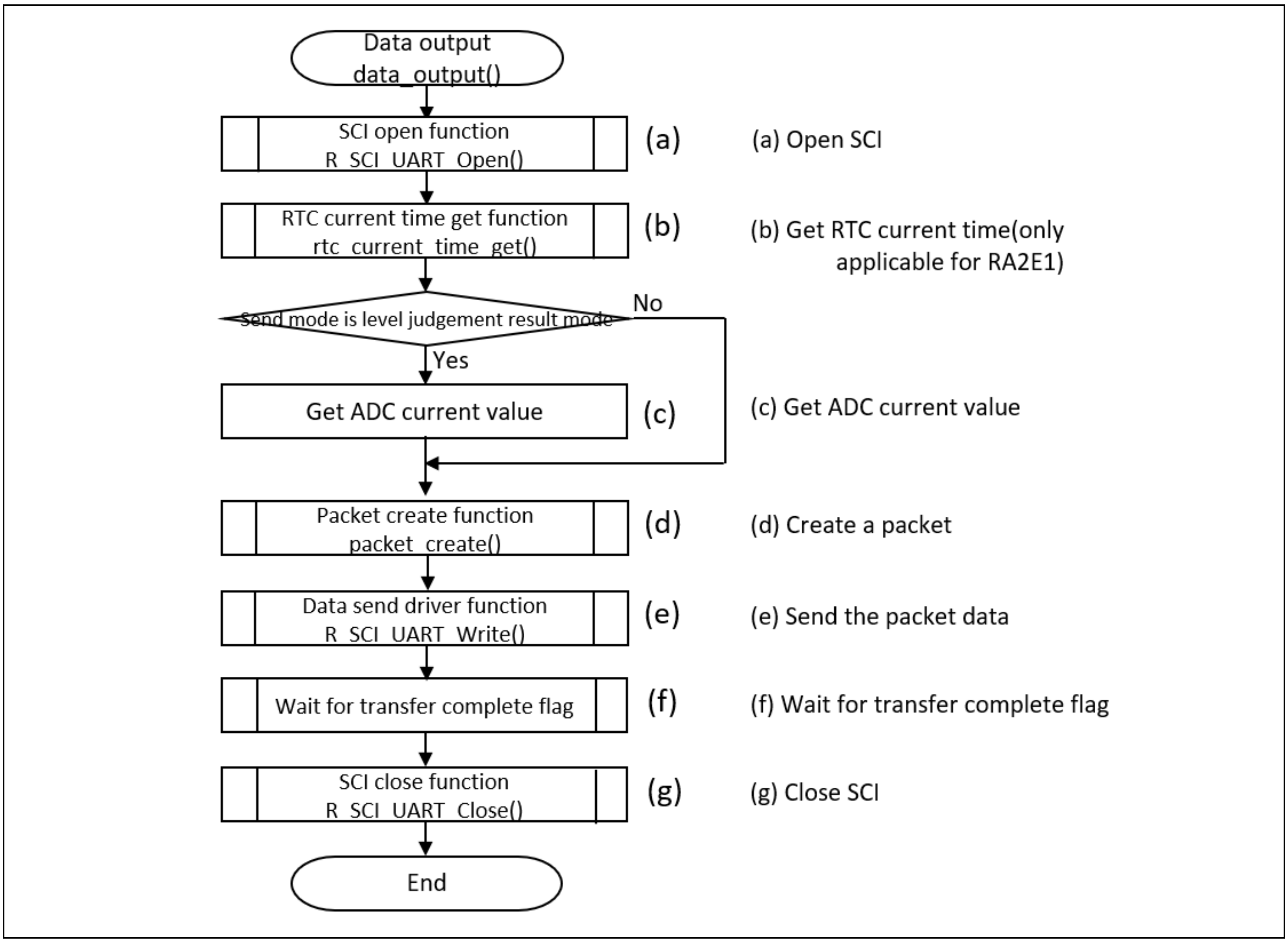


Figure 52. Data Processing

4. Evaluating Applications

4.1 Import and Building a Project

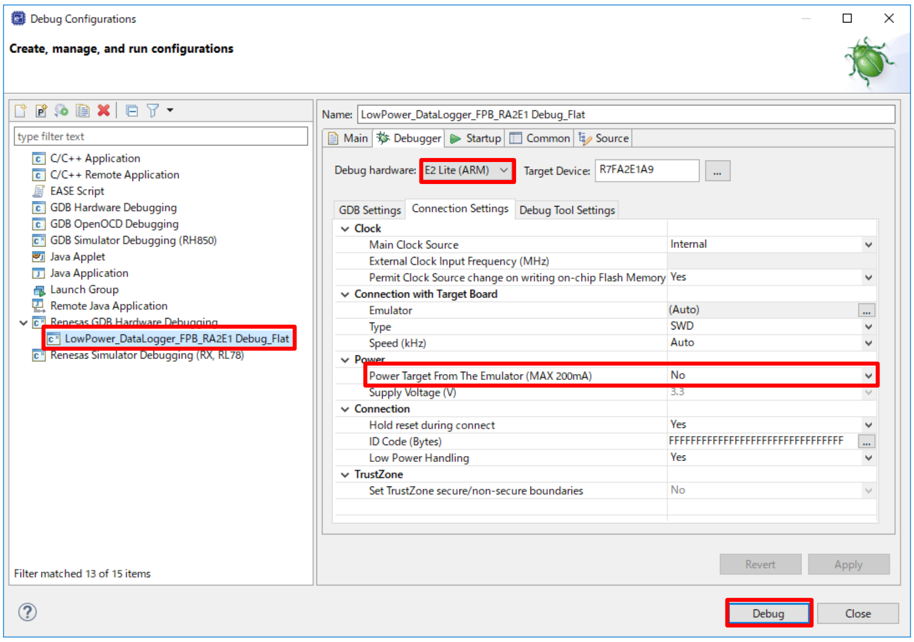
To build an application project with e^2 studio IDE, proceed as follows:

1. Launch e^2 studio IDE.
2. Select any workspace in the **Workspace launcher**.
3. Close the **Welcome** window.
4. Select **File** > **Import**.
5. Select **Existing Projects into Workspace** from the **Import** dialog box.
6. Select the archive file.
7. Select the project you want to import and click **Finish**.
8. Open configuration.xml and click **Generate Project Content** in the **Configurator** window
9. Select **Project** > **Build Project**.

4.2 Download and Debug a Project

To download and debug an application project using e^2 studio and onboard E2 emulator, proceed as follows:

1. Connect the connector CN6 of FPB-RA2E1 or FPB-RA2E2 and PC using USB cable.
2. Change the setting of the header CN1 on the FPB-RA2E1 or FPB-RA2E2 to a debugger-enabling setting (connect pins 1 and 2).
3. Open the **Debug Configuration** window.
4. Change settings as follows and as shown in Figure 53.
   — **Debug hardware**: **E2 Lite (ARM)**
   — **Power Target From The Emulator**: **No**
5. Clicking **Debug**



Name: LowPower_DataLogger_FPB_RA2E1 Debug_Flat

Debug hardware: E2 Lite (ARM) Target Device: R7FA2E1A9

Clock	
Main Clock Source	Internal
External Clock Input Frequency (MHz)	
Permit Clock Source change on writing on-chip Flash Memory	Yes
Connection with Target Board	
Emulator	(Auto)
Type	SWD
Speed (kHz)	Auto
Power	
Power Target From The Emulator (MAX 200mA)	No
Supply Voltage (V)	3.3
Connection	
Hold reset during connect	Yes
ID Code (Bytes)	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Low Power Handling	Yes
TrustZone	
Set TrustZone secure/non-secure boundaries	No

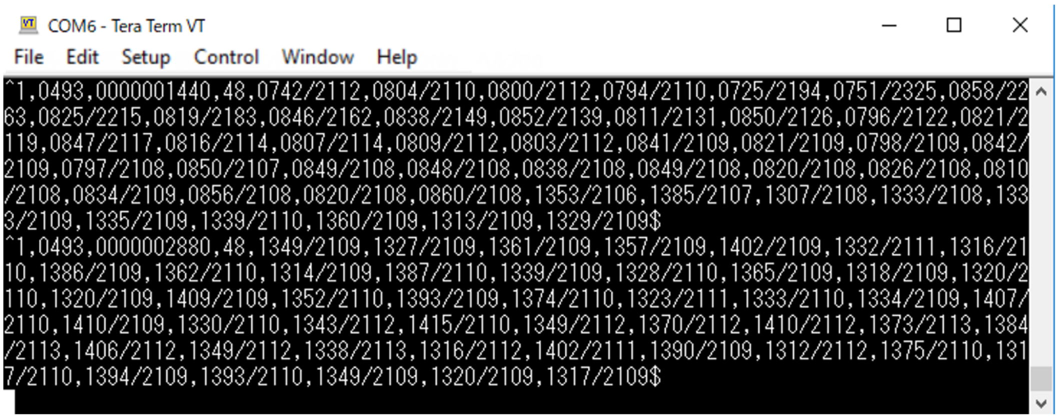
Figure 53. Debug Configuration

4.3 Procedure for Checking the Operation of the Low Power Data Logger

To check the operation of the low-power data logger, proceed as follows:

1. Connect the FPB-RA2E1 or FPB-RA2E2 and PC using USB - TTL connector and jumper wires. Refer to section 3.2.2 for detailed connections.
2. Launch terminal software on the PC and change the settings as follows:
   — Transfer speed: 115,200 bps
   — Data length: 8 bits
   — Parity: None
   — Stop-bit length: 1 bit
3. Referring to section 4.5, change the setting of the header CN1 on the FPB-RA2E1 or FPB-RA2E2 to a free-run operation setting.
4. Power on the kit.
5. Confirm the operation.

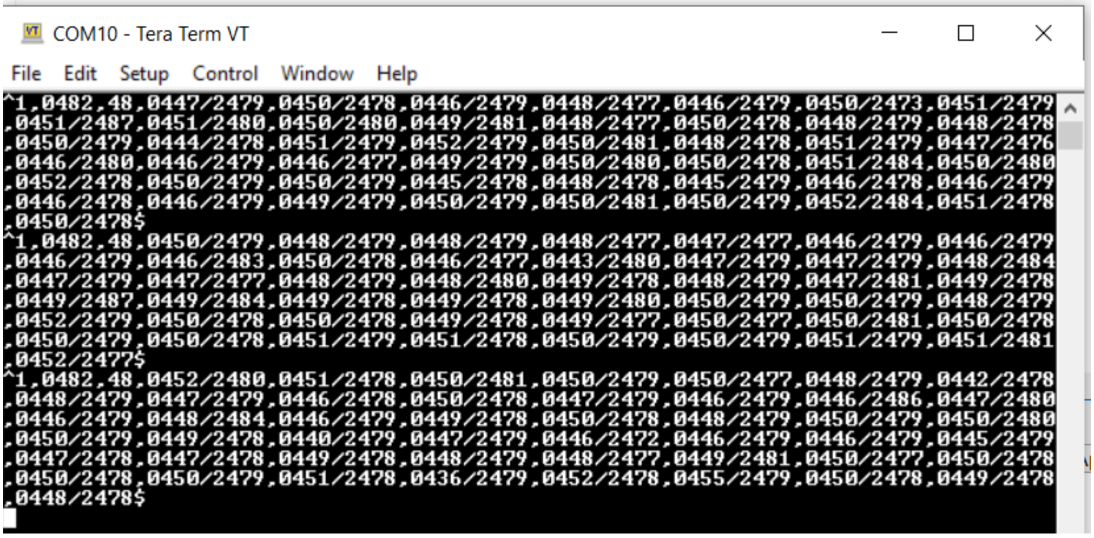
This application project performs data output every 24 hours (every 24 minutes in fast cycle debug is enabled) (See Figure 54 or Figure 55). In addition, data output is executed when the values of the light sensor and temperature sensor exceed the threshold value (See Figure 56 or Figure 57). The output data can be viewed on terminal software on the PC. Furthermore, pressing the user switch S1 of the FPB-RA2E1 or FPB-RA2E2 kit will output the progress of accumulated measurement data at any time (See Figure 58 or Figure 59).



```
^1,0493,0000001440,48,0742/2112,0804/2110,0800/2112,0794/2110,0725/2194,0751/2325,0858/22
63,0825/2215,0819/2183,0846/2162,0838/2149,0852/2139,0811/2131,0850/2126,0796/2122,0821/2
119,0847/2117,0816/2114,0807/2114,0809/2112,0803/2112,0841/2109,0821/2109,0798/2109,0842/
2109,0797/2108,0850/2107,0849/2108,0848/2108,0838/2108,0849/2108,0820/2108,0826/2108,0810
/2108,0834/2109,0856/2108,0820/2108,0860/2108,1353/2106,1385/2107,1307/2108,1333/2108,133
3/2109,1335/2109,1339/2110,1360/2109,1313/2109,1329/2109$
^1,0493,0000002880,48,1349/2109,1327/2109,1361/2109,1357/2109,1402/2109,1332/2111,1316/21
10,1386/2109,1362/2110,1314/2109,1387/2110,1339/2109,1328/2110,1365/2109,1318/2109,1320/2
110,1320/2109,1409/2109,1352/2110,1393/2109,1374/2110,1323/2111,1333/2110,1334/2109,1407/
2110,1410/2109,1330/2110,1343/2112,1415/2110,1349/2112,1370/2112,1410/2112,1373/2113,1384
/2113,1406/2112,1349/2112,1338/2113,1316/2112,1402/2111,1390/2109,1312/2112,1375/2110,131
7/2110,1394/2109,1393/2110,1349/2109,1320/2109,1317/2109$
```

Note: A fast cycle debug is enabled

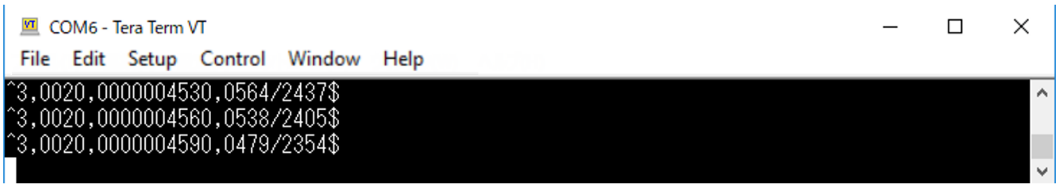
Figure 54. FPB-RA2E1 Example of Data Output Triggered by 24 Hours Passing



```
^1,0482,48,0447/2479,0450/2478,0446/2479,0448/2477,0446/2479,0450/2473,0451/2479
,0451/2487,0451/2480,0450/2480,0449/2481,0448/2477,0450/2478,0448/2479,0448/2478
,0450/2479,0444/2478,0451/2479,0452/2479,0450/2481,0448/2478,0451/2479,0447/2476
,0446/2480,0446/2479,0446/2477,0449/2479,0450/2480,0450/2478,0451/2484,0450/2480
,0452/2478,0450/2479,0450/2479,0445/2478,0448/2478,0445/2479,0446/2478,0446/2479
,0446/2478,0446/2479,0449/2479,0450/2479,0450/2481,0450/2479,0452/2484,0451/2478
,0450/2478$
^1,0482,48,0450/2479,0448/2479,0448/2479,0448/2477,0447/2477,0446/2479,0446/2479
,0446/2479,0446/2483,0450/2478,0446/2477,0443/2480,0447/2479,0447/2479,0448/2484
,0447/2479,0447/2477,0448/2479,0448/2480,0449/2478,0448/2479,0447/2481,0449/2478
,0449/2487,0449/2484,0449/2478,0449/2478,0449/2480,0450/2479,0450/2479,0448/2479
,0452/2479,0450/2478,0450/2478,0449/2478,0449/2477,0450/2477,0450/2481,0450/2478
,0450/2479,0450/2478,0451/2479,0451/2478,0450/2479,0450/2479,0451/2479,0451/2481
,0452/2477$
^1,0482,48,0452/2480,0451/2478,0450/2481,0450/2479,0450/2477,0448/2479,0442/2478
,0448/2479,0447/2479,0446/2478,0450/2478,0447/2479,0446/2479,0446/2486,0447/2480
,0446/2479,0448/2484,0446/2479,0449/2478,0450/2478,0448/2479,0450/2479,0450/2480
,0450/2479,0449/2478,0440/2479,0447/2479,0446/2472,0446/2479,0446/2479,0445/2479
,0447/2478,0447/2478,0449/2478,0448/2479,0448/2477,0449/2481,0450/2477,0450/2478
,0450/2478,0450/2479,0451/2478,0436/2479,0452/2478,0455/2479,0450/2478,0449/2478
,0448/2478$
```

Note: A fast cycle debug is enabled

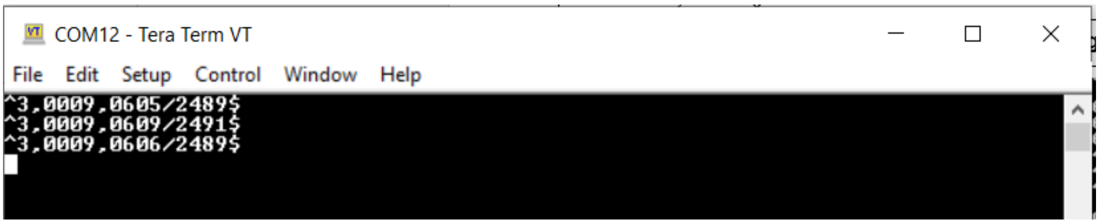
Figure 55. FPB-RA2E2 Example of Data Output Triggered by 24 Hours Passing



```
^3,0020,0000004530,0564/2437$
^3,0020,0000004560,0538/2405$
^3,0020,0000004590,0479/2354$
```

Note: A fast cycle debug is enabled

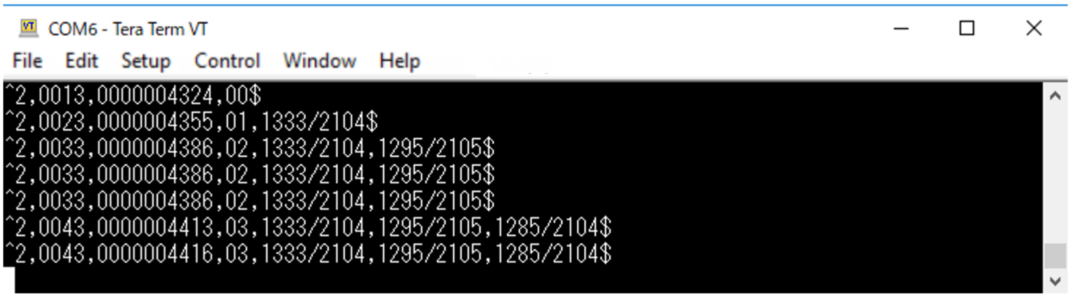
Figure 56. FPB-RA2E1 Example of Data Output Triggered by Analog Level Judgment Result



```
^3,0009,0605/2489$
^3,0009,0609/2491$
^3,0009,0606/2489$
```

Note: A fast cycle debug is enabled

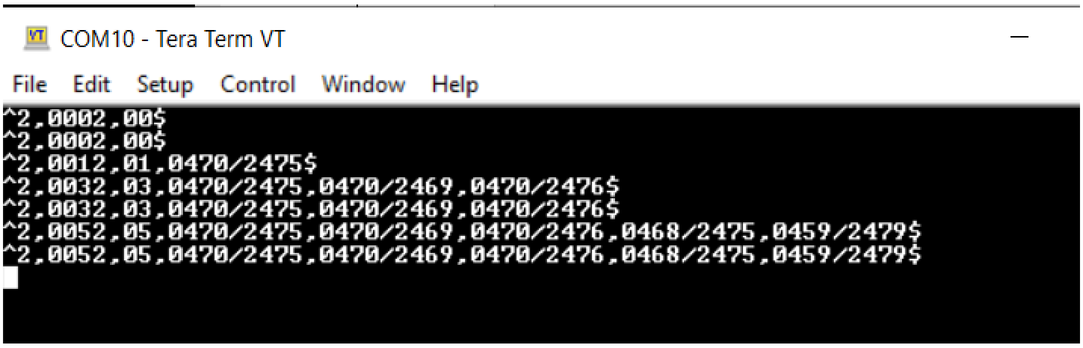
Figure 57. FPB-RA2E2 Example of Data Output Triggered by Analog Level Judgment Result



```
^2,0013,0000004324,00$
^2,0023,0000004355,01,1333/2104$
^2,0033,0000004386,02,1333/2104,1295/2105$
^2,0033,0000004386,02,1333/2104,1295/2105$
^2,0033,0000004386,02,1333/2104,1295/2105$
^2,0043,0000004413,03,1333/2104,1295/2105,1285/2104$
^2,0043,0000004416,03,1333/2104,1295/2105,1285/2104$
```

Note: A fast cycle debug is enabled

Figure 58. FPB-RA2E1 Example of Data Output Triggered by Pressing the User Switch



```
^2,0002,00$
^2,0002,00$
^2,0012,01,0470/2475$
^2,0032,03,0470/2475,0470/2469,0470/2476$
^2,0032,03,0470/2475,0470/2469,0470/2476$
^2,0052,05,0470/2475,0470/2469,0470/2476,0468/2475,0459/2479$
^2,0052,05,0470/2475,0470/2469,0470/2476,0468/2475,0459/2479$
```

Note: A fast cycle debug is enabled

Figure 59. FPB-RA2E2 Example of Data Output Triggered by Pressing the User Switch

4.4 Evaluating the Current Consumption

FPB-RA2E1 or FPB-RA2E2 implements a resistor (R3) and test points (TP1 and TP2) for measuring the current of the MCU power of 3.3 V. When measuring the current amount of the MCU, remove resistor R3 and connect the measuring equipment to the test points.

See Figure 60 and Figure 61 for the location of TP1 and TP2.

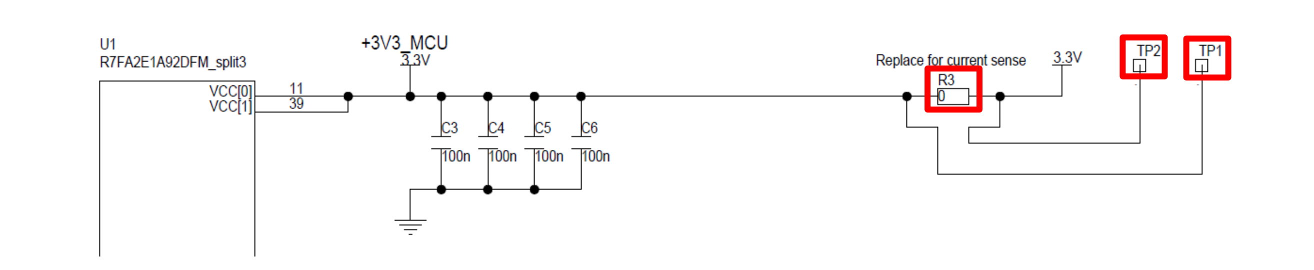


Figure 60. MCU Current Measuring Points (Circuit Diagram)

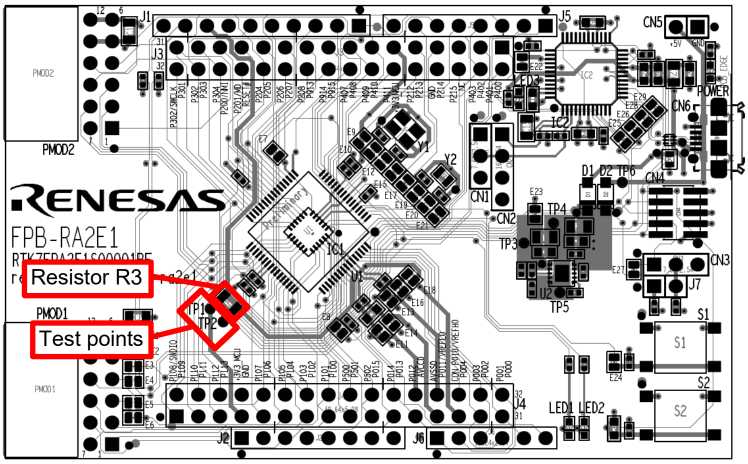


Figure 61. MCU Current Measuring Points

4.5 Notes on Free-Running Operation with the FPB-RA2E1 or FPB-RA2E2 Kit

If the FPB-RA2E1 or FPB-RA2E2 is in Free-Run operation, connect pins 2 and 3 of the header CN1. Connecting these pins forces the onboard E2 emulator into a reset state, and it can operate alone without control from the IDE.

See Figure 62 for the location of CN1.

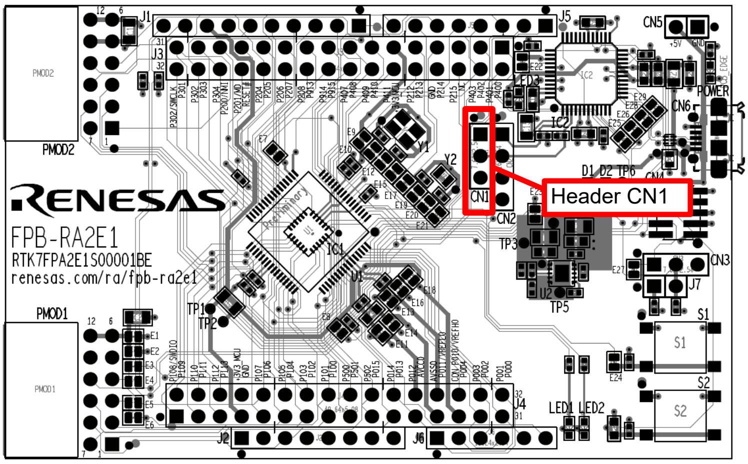


Figure 62. Header CN1 for Force Reset of Onboard E2 Emulator

5. References

- Renesas FSP User's Manual renesas.github.io/fsp
- Renesas RA MCU datasheet Select the relevant MCUs from the www.renesas.com/ra
- Example Projects github.com/renesas/ra-fsp-examples
- Application Note: *Getting Started with Low Power Applications for RA2L1/RA2E1 Group*（R11AN0480）
- Application Note: *Low Power Applications (Usage of ADC, DTC, and ELC at Snooze Mode)* for RA2E1(R30AN0392).

Website and Support

Visit the following URLs to learn about key elements of the RA family, download components and related documentation, and get support.

RA Product Information	www.renesas.com/ra
RA Product Support Forum	www.renesas.com/ra/forum
RA Flexible Software Package	www.renesas.com/FSP
Renesas Support	www.renesas.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct.12.21	-	Initial version
1.10	May.11.22	-	Added additional support for FPB-RA2E2
1.20	Sep.10.24	-	Update to FSP version 5.5.0

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/.