RA2L1/RA2E1 Group

Example of Low Power Application (Data Logger)

Introduction

This application note describes the features of RA2L1 MCU that are useful for low-power operation, typically required for logging data for long durations. The EK-RA2L1 kits is used for creating such a low-power data logging system. The functionality demonstrated here is typically required in products such as fitness trackers and fleet tracking devices. The data logger uses the 12-bit A/D converter (ADC), data transfer controller (DTC), asynchronous general-purpose timer (AGT), real-time clock (RTC), event link controller (ELC), 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 is described using RA2L1 MCU. However, it can also be applied to products that works with RA2E1 MCU. It includes a project that works with EK-RA2E1.

Prerequisites

We assume that you have developed using the Renesas e^2 studio IDE and Flexible Software Package (FSP). We recommend that you build and run the Blinky project according to the board's quick-start guide 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 board is working correctly.

Required Resources

This application project is created for the Renesas RA Family MCU RA2L1. 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. This application note describes the case using the following sensor modules, but you can check the operation by inputting any analog signal without using them. See 3.4(4) Disable Sensor Dependent Process for more information.

Hardware

- Renesas RA Kit EK-RA2L1
- Renesas RA Kit EK-RA2E1 (optional)
- Renesas RA Kit EK-RA6M3 Serial Number 721 or later[1][2]
- Seeed Grove Base Shield V2.0 for Arduino (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)

Development Tools and Software

- e^2 studio IDE version 2020-10 (20.10.0) or later
- Renesas Flexible Software Package (FSP) version 2.3.0 or later
- GCC ARM Embedded Toolchain version 9.2.1.20191025 or later
- Segger J-Link RTT Viewer version 6.88 or later[2]

Note 1: Please refer to errata document (R20ZZ0213) of EK-RA6M3 kit for more information.

Note 2: Required for checking the operation of low-power data logger application

Contents

1. Overview

1.1 Overview of Specifications

This application project is assumed to operate as a sub-module in the entire system consisting of a main module responsible for consuming data generated by the sub-module. This sub-module acquires sensor data, compares against thresholds and processes the accumulated data at regular intervals. These processes are intermittently executed with Snooze mode during Software Standby mode to achieve operation as a low-power data logger. When a wakeup request is received from the main module, the processed data is transmitted.

In this application project, Snooze mode is entered every minute 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. In addition, the low power mode is canceled every hour, and the measured data accumulated in the buffer is processed. After the data processing, the process repeats. When a wakeup request is received from the main module, the low power mode is canceled, and the requested data is transmitted. Table 1.1 explains the details of these processes for each function.

Table 1.1. List of functions

Function	Process Description
Data Acquisition Function	AGT count underflow event triggers A/D conversion and conversion results are stored in the measurement data buffer.
Level Judgement Function	The level of A/D conversion results is judged by two methods: the ADC compare function and the Data Operation Circuit (DOC). The judgment result is output to the external pin.
Data Processing Function	The RTC alarm interrupt sets the MCU to Normal mode and the MCU processes the accumulated data.
Data Output Function	When a wakeup request is received, the MCU shifts to Normal mode and the data is output by UART communication.

Figure 1.1 shows the MCU states and mode transition events, and Figure 1.2 shows a conceptual diagram of the operation modes and current consumption.
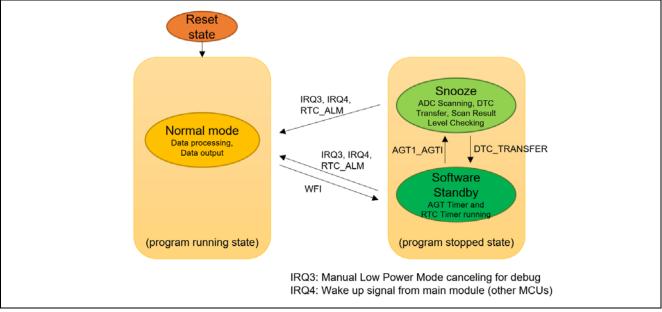

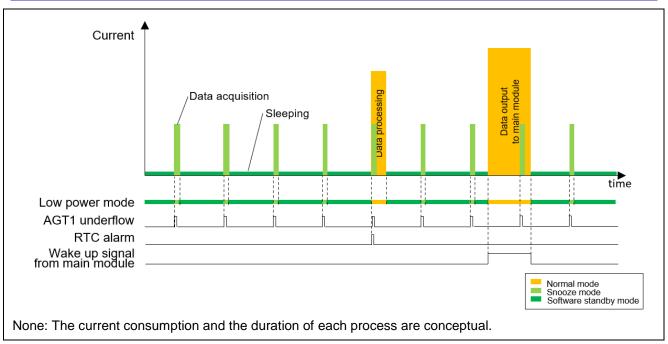**Figure 1.1. MCU status and mode transition events**

None: The current consumption and the duration of each process are conceptual.

Figure 1.2.   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 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. The DTC transfer completion event returns the system to Software Standby mode.

The level judgement function judges the level of A/D conversion results by two methods, the ADC compare function and the Data Operation Circuit (DOC). The conversion result of channel 0 is judged by the DOC subtraction mode, and the conversion result of channel 1 is judged by the ADC compare function. If the conversion result is greater than or equal to the threshold value, the port group function of the ELC is used to output High to the external output pin.

The processing sequence for the above functions is described using Figure 1.3.

① A transition to Snooze mode is requested when an AGT underflow occurs by the AGT timer set to 1 minute.

② The ADC starts A/D conversion by a synchronous trigger from the ELC linked event from AGT or Snooze request event and compares the result with the threshold by the compare function.

③ 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 is transferred to the channel 0 measurement data buffer.

    b.   The conversion result of A/D converter channel 1 is transferred to the channel 1 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 0 is transferred to the DOC data input register.

④ When the level judgment result event (Data Operation Circuit interrupt) of channel 0 is output to the ELC, pin P105 of I/O port 1 is output to High. When the level judgment result event (ADC12 compare match) of channel 1 is output to the ELC, pin P208 of I/O port 2 is output to High.

⑤ When a Not DTC transfer completion interrupt (DTC_TRANSFER) is generated due to completion of the transfer processing of ③, the system returns to the Software Standby mode.
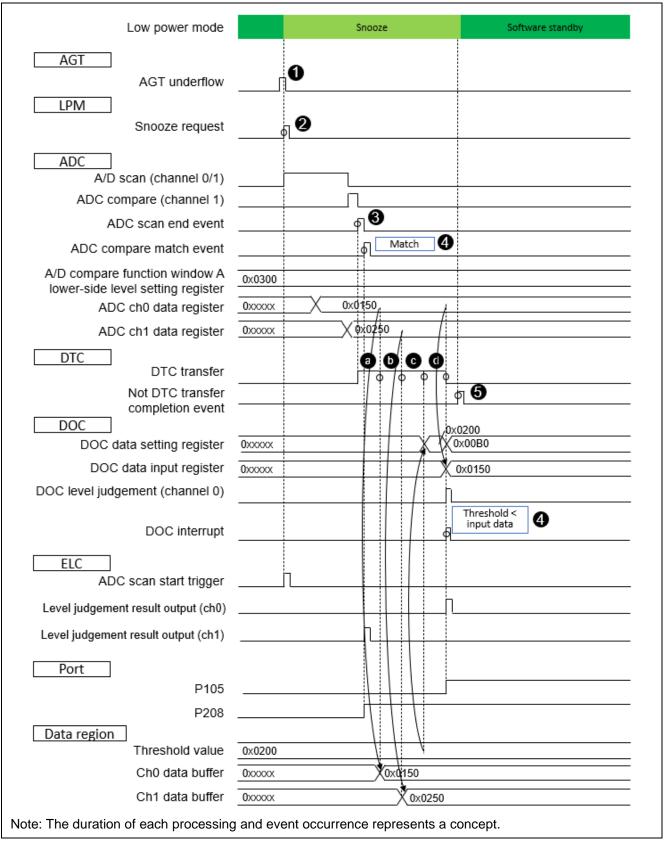
Figure 1.3.   Timing chart for data acquisition and level judgement

1.1.2   Data Processing Function

An RTC alarm interrupt causes the data processing function to transition from Software Standby mode to Normal mode, and the accumulated measurement data is processed by the CPU. After the data processing

is completed, 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 The RTC generates an alarm event every hour and cancels the low power mode.

② After the low power mode is canceled, the data accumulated in the measurement data buffer is processed. It calculates the average, minimum, and maximum values. The processed result is stored in the processed data buffer.

③ Sets the next RTC alarm after data processing is completed.

④ Enters the low power mode.

1.1.3 Data Output Function

The data output function transitions from Software Standby mode to Normal mode by receiving a wakeup request from the main module (External IRQ interrupt) and outputs data by UART communication. After confirming that the wakeup request pin has fallen, it transits to low power mode.

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

① When the wakeup request pin rises, the ICU generates an interrupt and cancels 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 confirming that the wakeup request pin has fallen.

1.2 Peripheral Modules of RA2L1 MCU

Table 1.2 lists the main peripheral modules of the RA2L1 MCU used in this application project and their typical uses. These are illustrated in the functional overview of RA2L1 MCU in Figure 1.4.

Table 1.2. Main peripheral modules

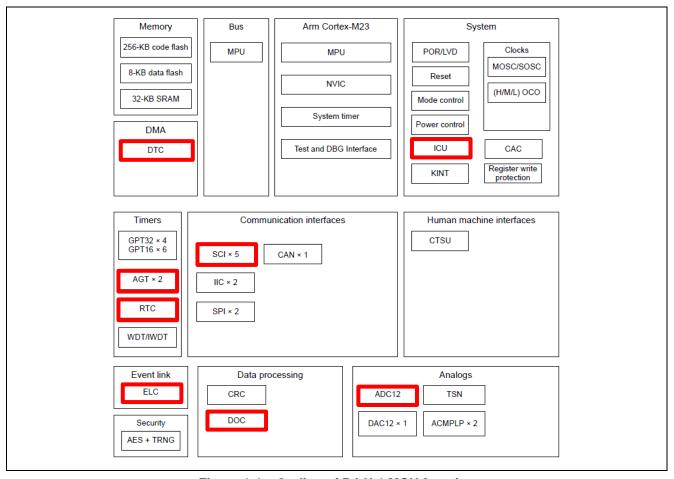| Module	Typical Uses
Low Power Modes	The Snooze mode and the Software Standby mode are used to achieve low power consumption.
12-bit ADC	Converts the analog continuous time signal from the sensor to discrete digital signal value. The compare function is used to judge the level of this conversion result.
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	Uses the subtraction mode to judge the level of the A/D conversion result.
16-bit AGT	Generates a transition request to Snooze mode and start data acquisition timing every minute.
RTC	Measures the time since power on. The hourly alarm interrupt is used to cancel the low power mode and generate the timing of data processing.
SCI	Performs UART communication with the main module.
ELC	Connects a Snooze transition request event or an AGT underflow event to the A/D converter start conversion event. In addition, the level judgment result event by the ADC compare function and the DOC are connected to each port output.

Figure 1.4.   Outline of RA2L1 MCU function

2.   Description of Functions Used

This section describes the functions of the LPM, ADC, DTC, ELC, and RTC modules of the RA2L1 MCU and explains how to set them to achieve the expected operation.

2.1   Low Power Modes

2.1.1   Available Low Power Modes

Table 10.2 in the RA2L1 User's Manual (R01UH0853) 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

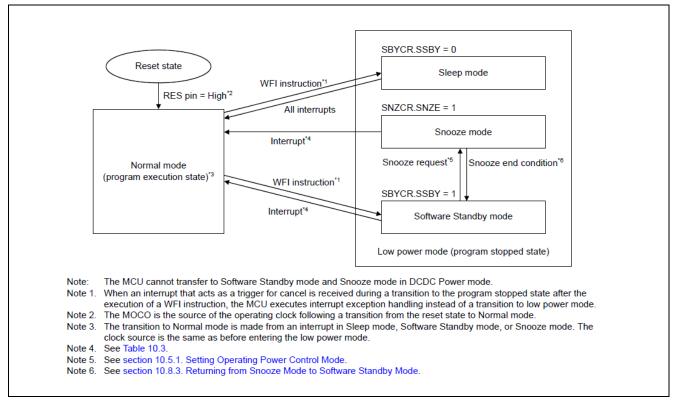Figure 2.1 shows a schematic diagram of the low power mode transitions.

Figure 2.1. 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 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 using interrupts 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 RA2L1 User's Manual (R01UH0853) 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 2.1 lists the mode transition events used in this application project.

Table 2.1. Mode transition events used in this application project

Transition request	Event
Request for transition to Snooze mode	AGT_AGTI
Snooze mode end request	DTC_TRANSFER
Request for transition to Normal mode from Snooze mode or Software Standby mode	PORT_IRQ3 PORT_IRQ4 RTC_ALM

2.2   12-bit A/D Converter

The RA2L1 MCU includes 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 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 Section 30.3.6 "Analog Input Sampling and Scan Conversion Time" and Table 41.12 "Operating and standby current" in RA2L1 User's Manual (R01UH0853) for deep understanding of difference between A/D conversion modes.

2.2.1   A/D Conversion Channel

The RA2L1 MCU 12-bit A/D converter unit (ADC12) can select up to 19 channels (AN000AN014, AN017AN020) of analog input, temperature sensor, internal reference voltage, or CTSU TSCAP voltage. This application project uses Seeed's Grove Base Shield V2.0 for Arduino to connect the sensor through the Arduino-compatible connector of the EK-RA2L1 kit. J19-1 and J19-2 of the Arduino-compatible connector are connected to analog channels 0 (AN000) and 1 (AN001) of the RA2L1 MCU. They can be read by selecting channels 0 and 1. Figure 2.2 shows the analog signal pin connections for Arduino-compatible connectors of the EK-RA2L1 kit.

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).
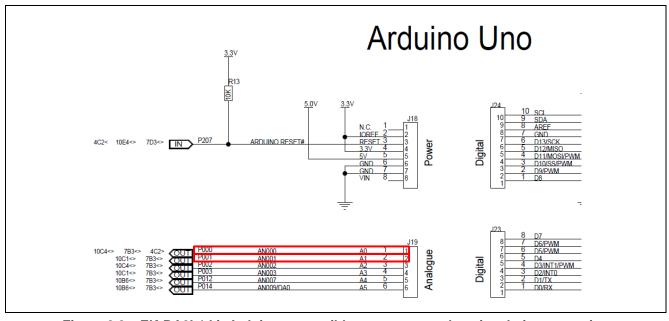

Figure 2.2.   EK-RA2L1 kit Arduino compatible connector analog signal pin connection

2.2.2   Conversion Start Event

Figure 2.3 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.

A/D conversion of a synchronous trigger from the ELC can be started by the preset event of the ELC specified by 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
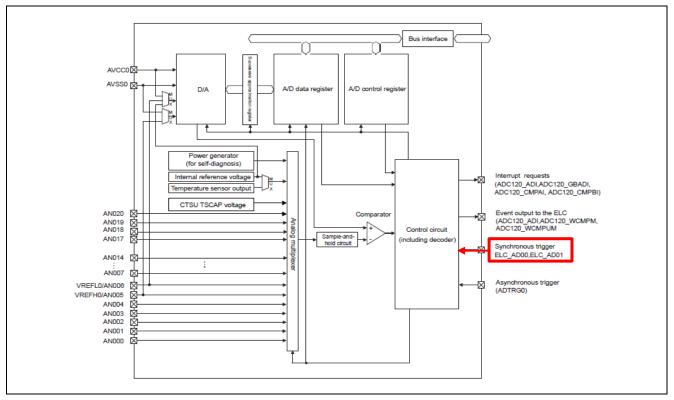- Select ELC_AD01 in the ELC. ELSR9 register

Figure 2.3.   ADC12 Input/Output events

2.2.3   Data Register Automatic Clearing Function

When A/D data register is read by the CPU or DTC, A/D data register (ADDRn, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR) can be automatically cleared to 0x0000 by setting the ADCER.ACE bit to 1.

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

2.2.4   Compare Function

The A/D converter compare function compares the reference value with the A/D conversion result. The compare function can generate the following events.

- ADC_WCMPM: When the ADC conversion result matches the comparison condition
- ADC_WCMPUM: When the ADC conversion result does not match the comparison conditions

Compare Window A can set four comparison conditions using two control register bits as shown in Table 2.2.

Table 2.2.   Comparative conditions of 2 ADC compare functions

Window function setting (ADCMPCR.WCMPE)	Compare Window A Comparison Condition Select (ADCMPLR0.CMPLCHAn)	Comparison condition
Disabled (0)	Less than reference value (0)	ADCMPDR0 > A/D-converted
	Greater than reference value (1)	ADCMPDR0 < A/D-converted
Enabled (1)	Out of Condition Range (0)	A/D-converted value < ADCMPDR0, or ADCMPDR1 < A/D-converted
	Within the condition range (1)	ADCMPDR0 < A/D-converted value < ADCMPDR1 value

In this application project, the window function is disabled for analog signal channel 1 to judge whether the value is greater than or equal to the threshold value. In this case, the ADC registers are set as follows:

```
//--- Set the Window_A ------
//Select a channel for Window_A Comparison
R_ADC0->ADCMPANSR_b[0].CMPCHA1 = 0x1;
//Set the Window_A Lower boundary
R_ADC0->ADCMPDR0 ADCMPDR0_b. = Level judgement threshold;
//Set the Window_A comparison condition
R_ADC0->ADCMPLR_b[0].CMPLCHA1 = 0x1;
//Enable Window_A operation
R_ADC0->ADCMPCR_b.CMPAE = 0x1;
//Enable Window_A interrupt for meeting the condition, ADC140_CMPAI
R_ADC0->ADCMPCR_b.CMPAIE = 0x1;
```

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 the DTC, the data to be transferred must be stored in an 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 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 the IELSRn.DTCE bit in ICU is set to 1, the 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 2.4 shows DTC activation by an interrupt request from a peripheral module.
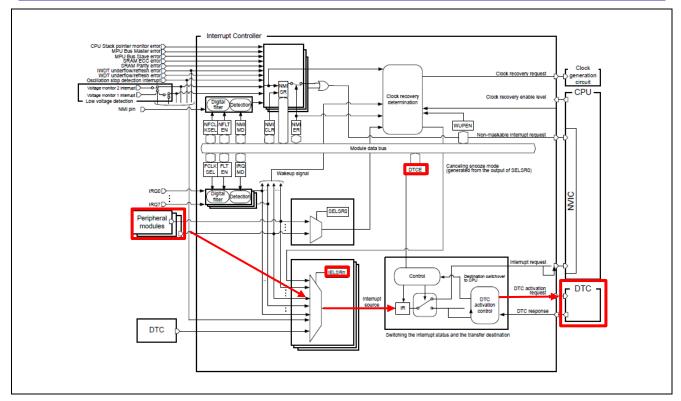
Figure 2.4.   DTC activation by an interrupt request from a peripheral module

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 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 2.5 shows the chain transfer operation in this case.

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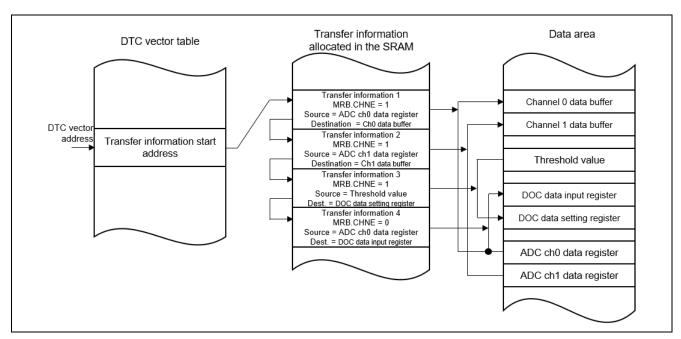- The conversion result of A/D converter channel 0 is stored in the DOC data input register.

Figure 2.5.   Chain transfer operation

When chain transfer is set using the DTC driver of 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 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 shown in Figure 2.5 is set as follows.

```
transfer_info_t g_dtc_data_transfer_info[4] = {
{
  .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,
},
{
  .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,
},
{
  .dest_addr_mode = TRANSFER_ADDR_MODE_FIXED,
  .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,
},
{
  .dest_addr_mode = TRANSFER_ADDR_MODE_FIXED,
  .repeat_area = TRANSFER_REPEAT_AREA_DESTINATION,
  .irq = TRANSFER_IRQ_END,
  .chain_mode = TRANSFER_CHAIN_MODE_DISABLED,
  .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,
 }
 };
```

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. In this application project, when the measured data is greater than or equal to the threshold value, the level judgment operation to generate an event is realized by the DOC subtraction mode. The procedure for determining the level using the DOC subtraction mode is shown below. Figure 2.6 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 is stored in the data input register (DODIR).

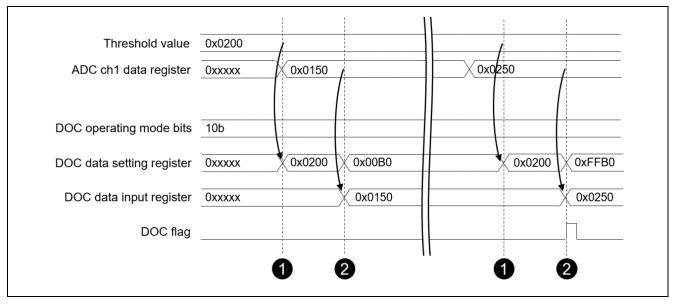③ If the subtraction is less than 0x0000, DOCR.DOPCF flag is set to 1.

Figure 2.6.   Level judgement using DOC subtraction mode

2.5 ELC Event Connection

The ELC uses event signals/indications generated by peripheral modules as source signals. It passes them to other modules to provide direct linkage between modules without interrupting the CPU operation. As a result, Interrupt Service Routines to 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 16.3 in the RA2L1 User's Manual (R01UH0853) for the event number to be set in the ELSRn register.

2.5.1 ELC Event Connection for ADC Start Conversion

As described above, the A/D converter starts conversion with a synchronous trigger from the ELC. In this application project, the event that triggers the ADC conversion start is changed 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 2.7 shows the event connection in the ELC module at that time.
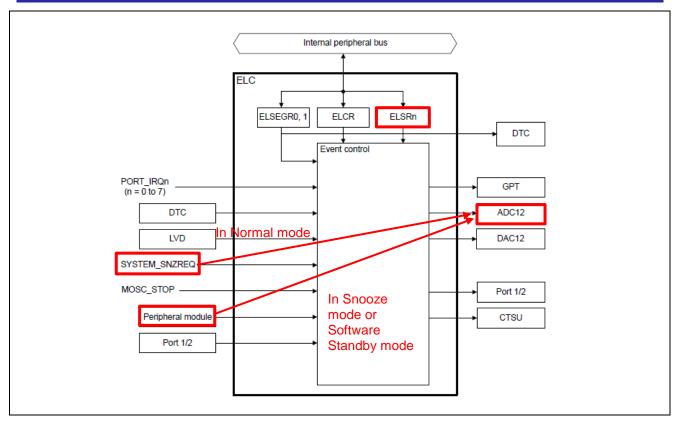
Figure 2.7.   ELC event connection for ADC start conversion

2.5.2   ELC Event Connection for Outputting the Level Judgment Result

In this application project, the level judgment result is output to the external pin without CPU. Connect the events to the I/O ports as shown in Table 2.3. Figure 2.8 shows the connection in the ELC module.

Table 2.3.   ELC event connection for level judgment result output

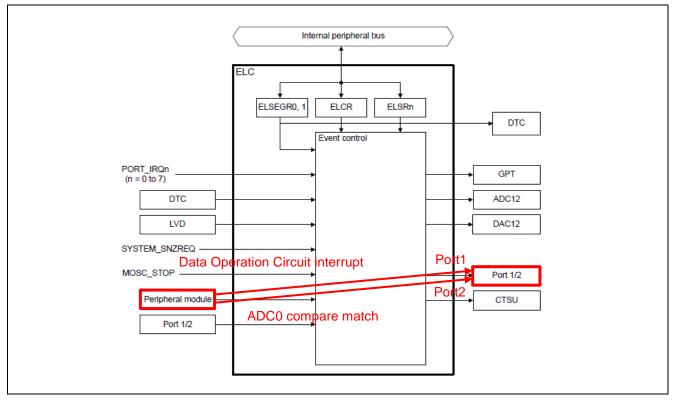| Event	Destination Module
Data Operation Circuit interrupt (DOC_DOPCI)	ELC_PORT1:PORT1
ADC12 compare match (ADC120_WCMPM)	ELC_PORT2:PORT2

Figure 2.8.   ELC event connection for level judgment result output

2.5.3   Port Output by ELC Event

Port 1 and port 2 are assigned to the ELC port group function by the RA2L1 MCU. When the ELC_PORTn (n = 1, 2) signal is generated, data is output from PODR to the external pin based on EOSR and EORR register settings as shown below. Figure 2.9 shows the port output circuit by an ELC event.

- When EOSR is set to 1, PODR register outputs 1 to the external pin when the ELC_PORTn (n = 1, 2) signal is generated. If EOSR = 0, PODR is retained.
- When EORR is set to 1, PODR register outputs 0 to the external pin when the ELC_PORTn (n = 1, 2) signal is generated. If EORR = 0, PODR is retained.
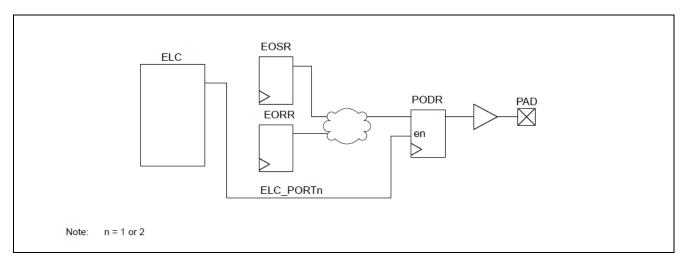

Figure 2.9.   Port output by ELC event

2.5.4   Configuring ELC Event Connections in the Configurator

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 seen on **Stacks** tab. The ELC Allocation setting status can be checked with Allocations by opening Event Links as shown in Figure 2.10. 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.
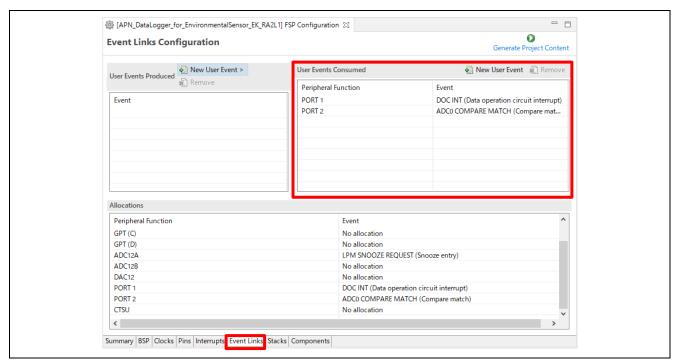

Figure 2.10.   ELC connection settings in the configurator

2.6   RTC Mode Selection

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 in low-power modes. For more information, see Table 22.1 in the RA2L1 User's Manual (R01UH0853).

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 RCR2.CNTMD bit. When this bit is rewritten, the next processing is performed after confirming that the value has been rewritten. If the count mode is set again, the RTC software reset is executed and redo the initial settings. This bit is updated synchronously with the count source, but the count mode is switched after a RTC software reset.

2.7   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 CLKOUT pin, set CKOCR.CKOEN to 1.

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

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 3.1 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 0V and 3.3V respectively.
- Acquisition of data from each sensor is performed every minute regardless of the low power mode status (using Snooze request event or AGT underflow interrupt).
- The size of the measurement data buffer is 120 samples.

3.1.2   Data Level Judgement Function

- After acquiring the data, the analog signal pin AN000 judges the level by DOC, and the analog signal pin AN001 judges the level by the ADC compare function.
- When AN000 is equal to or greater than the threshold (600Lux), output a high level to the level judgement result (channel 0) output pin.
- When AN001 is equal to or greater than the threshold (30.00°C), output a high level to the level judgment result (channel 1) output pin.

3.1.3   Data Processing Function

- Processing by the hourly RTC alarm interrupt.
- The data processing calculates the average, minimum, and maximum value of the data accumulated in the measurement data buffer.
- The processed data buffer size is 50 times (hours).
- Outputs a high level to the data full pin when the processed data buffer is full.

3.1.4   Data Output Function

- Output all data only when the wakeup request signal is High.
- When a command is received, data is transmitted according to the command.
- UART communication format is 115,200 bps baud rate, 8-bit data length, no parity, 1 stop bit.
- When response to the Data-Output-Request command is completed, output a low level to the data full pin set a high level by Data Processing Function. When data output for the level judgement result reception complete command is completed, output a low level to the level judgement result output pin set a high level by Data Level Judgment Function.
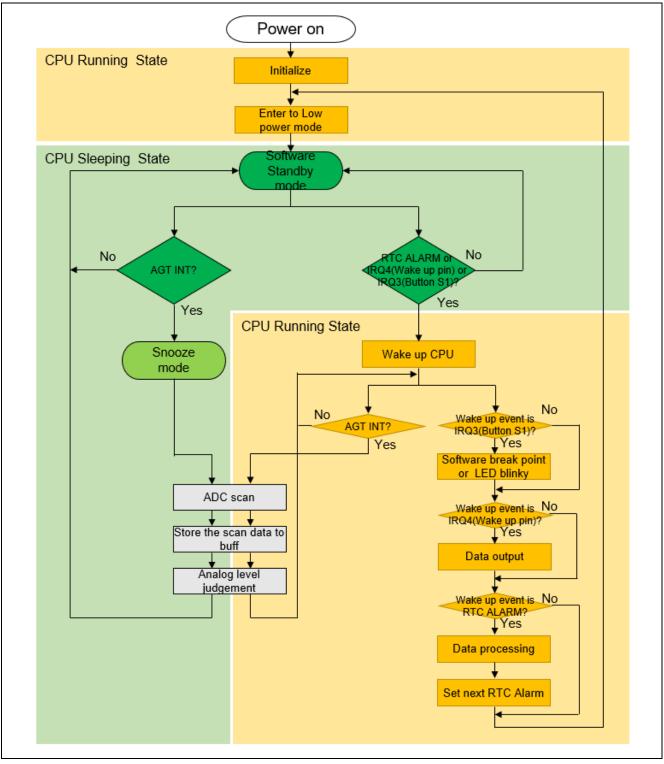
Figure 3.1.　Overall algorithm of the low-power data logger

3.2　User Interface

This chapter describes the user interface for the Low-Power Data Logger application. This application is designed to connect and execute with the main module (such as EK-RA6M3). Therefore, this chapter contains connection and communication specifications for outputting the data to main module. See chapter 4.4 "Checking the operation of the low-power data logger" for more information of checking operation using main module.

3.2.1 Connecting the Sensor

Figure 3.2 shows the connection diagram for the sensor. In this application project, attach Seeed's Grove Base Shield V2.0 for Arduino to the Arduino-compatible connector of the EK-RA2L1 kit. Attach the sensor module to the A0 and A1 connectors of this Base Shield with the Grove cable. This connects the analog signal pins (AN000 and AN001) of RA2L1 MCU with the analog output pins of the two sensors.

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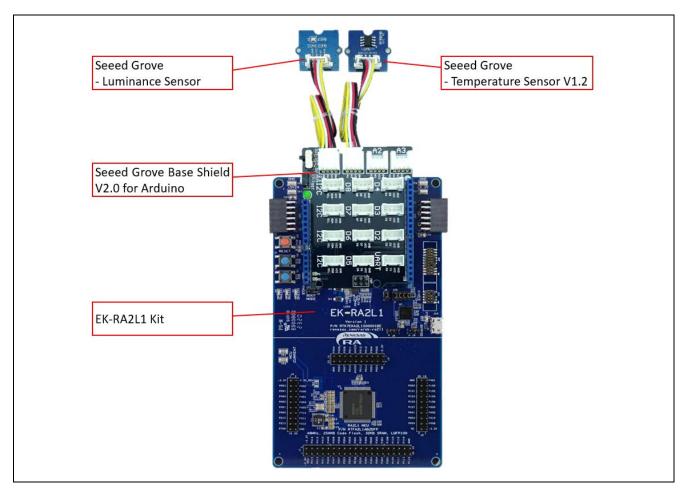

Figure 3.2.   Sensor connection

3.2.2 Connecting the Main Module

Table 3.1 and Figure 3.3 show the connection pins to the main module. Connect PMOD1 connector of the EK-RA2L1 kit using jumper wires.

Table 3.1.   Pins to be connected to the main module (EK-RA2L1 kit)

RA2L1 pin	EK-RA2L1 Pin	Typical Uses
P101/TxD0	PMOD1-2	Serial communication (transmission)
P100/RxD0	PMOD1-3	Serial communication (reception)
P402/IRQ4	PMOD1-7	Wakeup request reception
P208	PMOD1-8	Output level judgment result (channel 1)
P105	PMOD1-9	Output level judgment result (channel 0)
P106	PMOD1-10	Output data full status of processed data buffer

Serial Communications TX

Serial Communications RX

Pin 6

Pin 12

Pin 1

Pin 7

Data Full Status Output of Processed Buffer

Level judgement (channel 0) result output

Level judgement (channel 1) result output
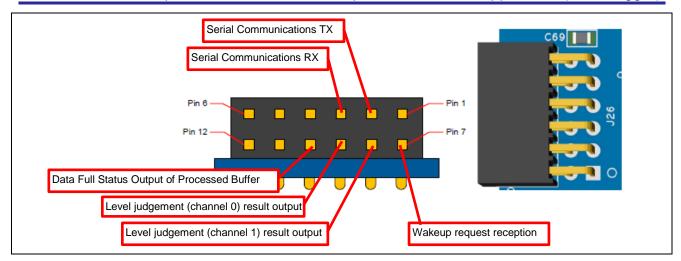
Wakeup request reception

Figure 3.3.   Connecting to the Main Module (PMOD1)

3.2.3   Data Communication Specifications

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

3.2.3.1   Communication Procedure

Figure 3.4 shows the sequence diagram for the main module. The sub-module cancels the low power mode when it receives a wakeup request from the main module and sends a wakeup completion notification in response. After that, when a command is received from the main module, data according to that command is transmitted. Then, when the main module output a low level to the wakeup request pin, the sub-module terminates data output and enters to the low power mode.
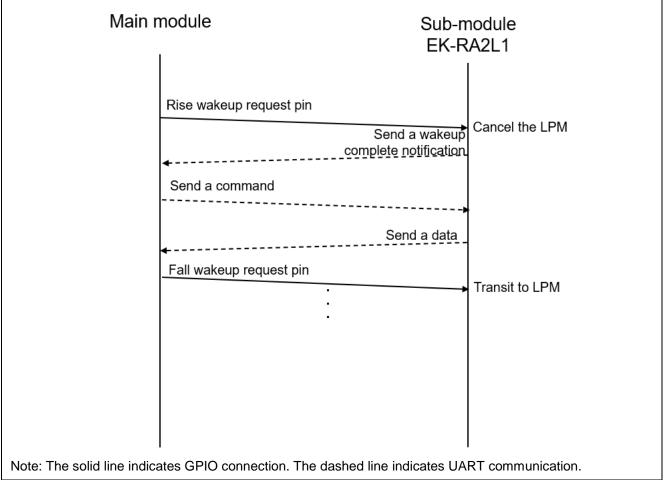
Figure 3.4.   Sequence diagram with main module

3.2.3.2   Communication Data Format

(1)   Receive Format

Below is the packet format/protocol of data received by the EK-RA2L1.

Packet format:

(a) Command

Description	Code	Length	Function
(a) Command	ASCII Code	1 byte	'1': Request average only data '2': Request all data (average/minimum/maximum) '3': Level judgement result reception complete

(2)   Transmission Format

Below is the packet format/protocol of data transmitted by the EK-RA2L1.

Packet format:

(a) Start code	(b) Attribute code	(c) Data length code	(d) Data	(e) End code

Description	Code	Length	Function
(a) Start code	'^'	1 byte	Start of the packet
(b) Attribute code	ASCII Code	1 byte	'1': Output average only data '2': Output all data '3': Output ADC current data '4': Send wakeup completion notification '5': Send command error notification
(c) Data length code	ASCII Code	4 bytes	Indicates the length of the data section.
(d) Data	ASCII Code	Number of bytes specified in (c)	See the following Data Format
(e) End code	'$'	1 byte	End of the packet

(2). a. Data format

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

- Attribute code 1 (Send average only data)

    Format of data part:

Number of data	'/'	First data	'/'	Second data	'/'	・・・	'/'	Last data

Time data	','	Light data Average	','	Temperature Data Average

- Attribute code 2 (send all data)

    Format of data part:

Number of data	'/'	First data	'/'	Second data	'/'	・・・	'/'	Last data

Time data	','	Light Data Average	','	Light Data Minimum	','	Light Data Maximum.	','	Temperature Data Average	','	Temperature Data Minimum	','	Temperature Data Maximum.

- Attribute code 3 (Transmit data immediately before ADC)

    Format of data part:

Time data	','	ADC current data Light	','	ADC current data Temperature

- Attribute code 4/5 (Send start completion notification/Send command error notification)

    Format of data part:

Space (0x20)

The data comprising the format of attribute codes 1 to 3 is generated as follows. Attribute codes 4 and 5 consist of fixed data (space) of 1 byte.

- The data count consists of 0 to 9 ASCII codes. The length is 2 bytes. Example: When the system transmits "10", it sends 0x31, 0x30.
- The time/light/temperature/ADC measurement data consists of 0 to 9 ASCII codes. Example: When the system transmits a time value "1234", it sends 0x31, 0x32, 0x33, 0x34.
- The time data is composed of a 10-digit decimal number representing elapsed time.
- The light data is composed of 4-digit illuminance values rounded to the nearest whole number, and the effective range is from 0 to 1000 Lux.
- The composition of temperature data is rounded off to the third decimal place, and it is a 4-digit temperature value excluding the decimal point. The valid range is 0.00°C to 99.99°C. Example: "2840" at 28.40°C.
- The delimiter between the processed data is '/', the delimiter between the time data, the illumination data, and the temperature data is ','.

Note: When the Disable Sensor Dependent Process function is enabled, the measurement/processed data of channel 0 is stored in the illuminance data part, and the measurement/processed data of channel 1 is stored in the temperature data part in 4-digit ASCII code.

3.2.4   Input/Output Pins for Debugging

Table 3.2 and Figure 3.5 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 3.2.   Debugging I/O pins (EK-RA2L1 kit)

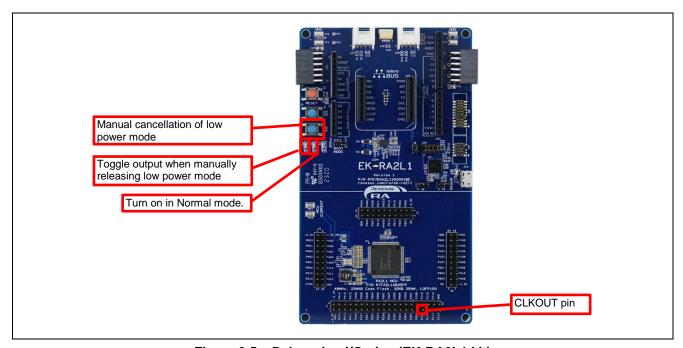| RA2L1 pin	EK-RA2L1 pins or connecting components	Typical Uses
P109/CLKOUT	J1-33	Clock output
P503	LED1/Blue LED	Blinks when low power mode is canceled manually.
P504	LED2/Green LED	Turn on in Normal mode
P004/IRQ3	User switch S1	Manual cancellation of low power mode

Figure 3.5.   Debugging I/O pins (EK-RA2L1 kit)

3.2.5   User Interface on EK-RA2E1 Kit

Table 3.3 and Table 3.4 show the pins for connecting to the main module and for debugging when EK-RA2E1 is used. Refer to the EK-RA2E1 User's Manual (R20UT4825) for the location of the pins.

Table 3.3. Pins to be connected to the main module (EK-RA2E1 kit)

RA2E1 pin	EK-RA2E1 Pin	Typical Uses
P101/TxD0	PMOD1-2	Serial communication (transmission)
P100/RxD0	PMOD1-3	Serial communication (reception)
P411/IRQ4	PMOD1-7	Wakeup request reception
P208	PMOD1-8	Output level judgment result (channel 1)
P105	PMOD1-9	Output level judgment result (channel 0)
P106	PMOD1-10	Output data full status of processed data buffer

Table 3.4. I/O pins for debugging (EK-RA2E1 kit)

RA2E1 pin	EK-RA2E1 pin or connecting component	Typical Uses
P109/CLKOUT	J2-27	Clock output
P915	LED1/Blue LED	Blinks when low power mode is canceled manually.
P914	LED2/Green LED	Turn on in Normal mode
P004/IRQ3	User switch S1	Manual canceling of low power mode

3.3   Clock Input

The clock from the external oscillator must be input in order for the RTC or AGT to perform an accurate time count. EK-RA2L1 kit implements a 20-MHz clock crystal and a 32.768-kHz clock crystal, but by default it is not connected to the RA2L1 MCU by a solder jumper. When accurate time counting is required, connect using solder bridge. The project attached with this application note selects a sub-clock oscillator (external 32.768-kHz clock crystal) as the clock source.

Figure 3.6 and Figure 3.7 show the positions of the solder jumpers for external oscillator connections.
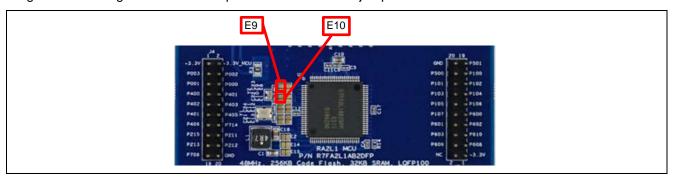

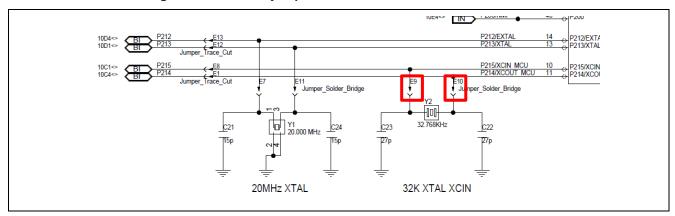**Figure 3.6.   Solder jumper for external oscillator connection**

Figure 3.7.   Solder jumper for external oscillator connection (circuit diagram)

3.4   Debugging

This application project implements pin input/output function for checking the low power mode status, fast cycle debug function, and disable sensor dependent process function.

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

User switch S1 of EK-RA2L1 can be pressed to generate an IRQ3 interrupt and cancel the low power mode. Setting a break point 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 pin.

When LED2 is 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, CLKOUT pin retains either High or Low state. When toggling out, the mode is changed to Normal mode or a 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.

Table 3.5 lists the pin output states in each low power mode, and Figure 3.8 and Figure 3.9 show examples of pin outputs in each low power mode.

Table 3.5. Pin output states in each low power mode

Low Power Modes	P109/CLKOUT	P504/LED2
Normal mode	Toggle	High
Software Standby mode	High or Low	Low
Snooze mode	Toggle	Low

Figure 3.8. Example of pin output in Snooze mode

Figure 3.9. Example of pin output in Normal mode

3.4.3 Fast Cycle Debug

This sample project implements a function to speed up the execution cycle of data acquisition and data processing in order to check the operation in a short time. To enable this feature, define `DEBUG_FAST_CYCLE` macro in `app_common.h`. When this `DEBUG_FAST_CYCLE` macro is defined, the following process is enabled, and the data acquisition period is set every 1 second and the data processing period is set every 1 minute.

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

3.4.4 Disable Sensor Dependent Process

In this application project, you can check the operation of the data logger application using any analog signal without using the specified sensor module. To enable this feature, define the `DISABLE_SENSOR_DEPENDENCE` macro in `app_common.h`. Defining this `DISABLE_SENSOR_DEPENDENCE` macro enables the following process, and you can check the operation of data processing and data output using raw ADC read value.

- Disable unit conversion for the specified sensor module.
- Change the level judgment threshold of channel 0/1 to an arbitrary value (2500 by default).
- Change communication data format.

3.5 Flowcharts

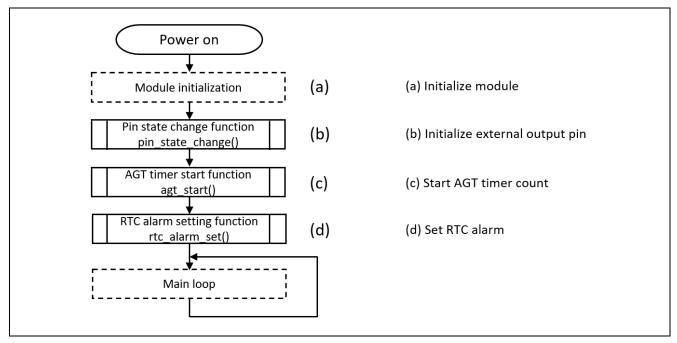Figure 3.10 shows the overall flow of the application project.

Figure 3.10. Overall flow

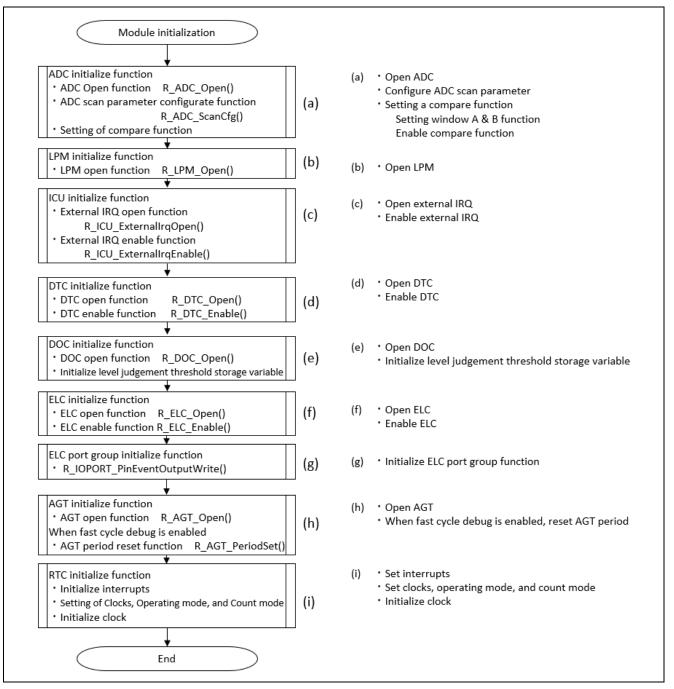Figure 3.11 shows the module initialization processing flow.

Figure 3.11.　Module initialization processing
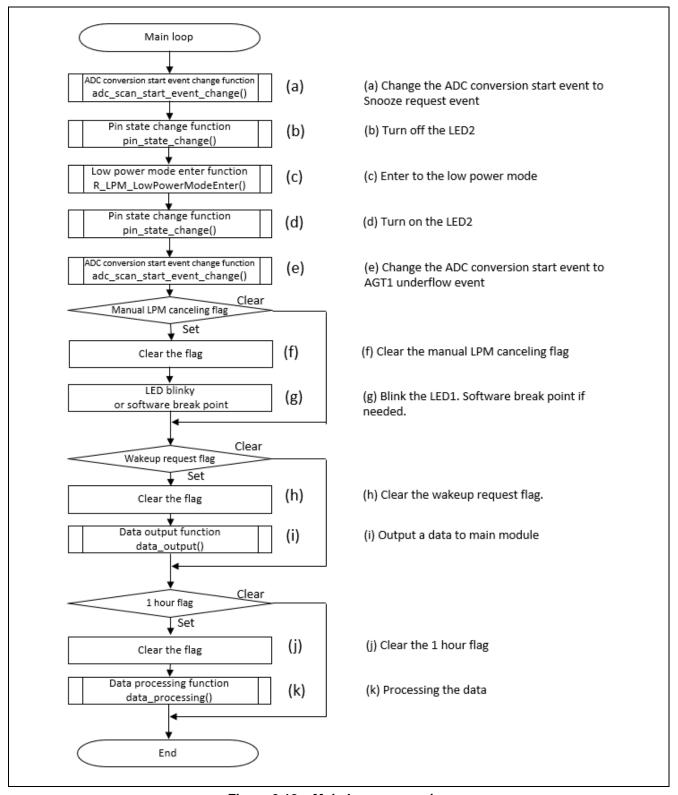
Figure 3.12 shows the main loop processing flow.

Figure 3.12.   Main loop processing

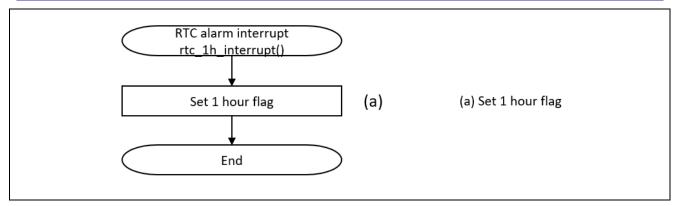Figure 3.13 shows the flow of RTC alarm interrupt processing.

Figure 3.13.   RTC alarm interrupt processing

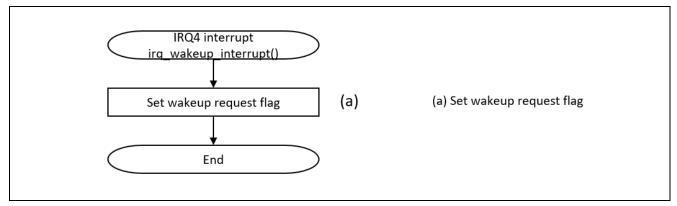Figure 3.14 shows the flowchart for IRQ4 interrupt processing.

Figure 3.14.   IRQ4 interrupt processing

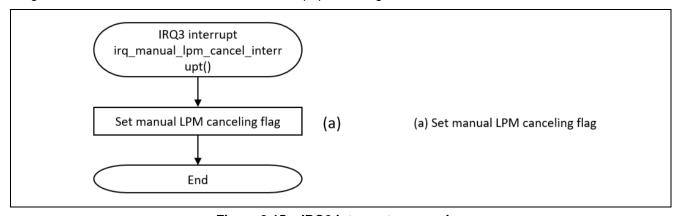Figure 3.15 shows the flowchart for IRQ3 interrupt processing.

Figure 3.15.   IRQ3 interrupt processing

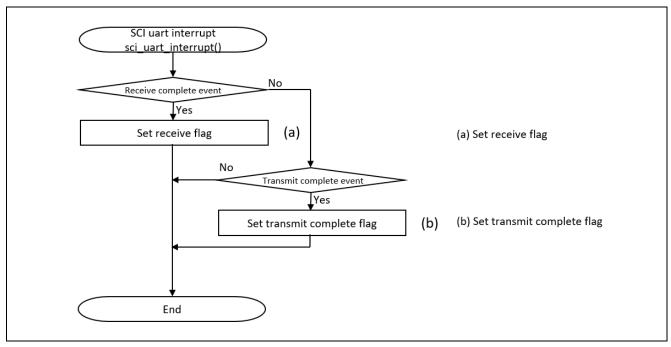Figure 3.16 shows the SCI interrupt processing flow.

Figure 3.16.   SCI interrupt processing

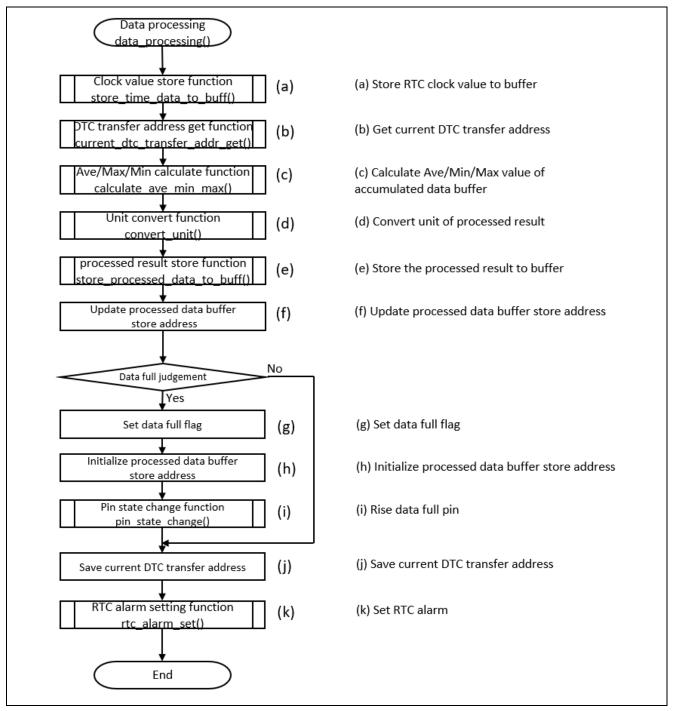Figure 3.17 shows the flow of data processing.

Figure 3.17. Data processing

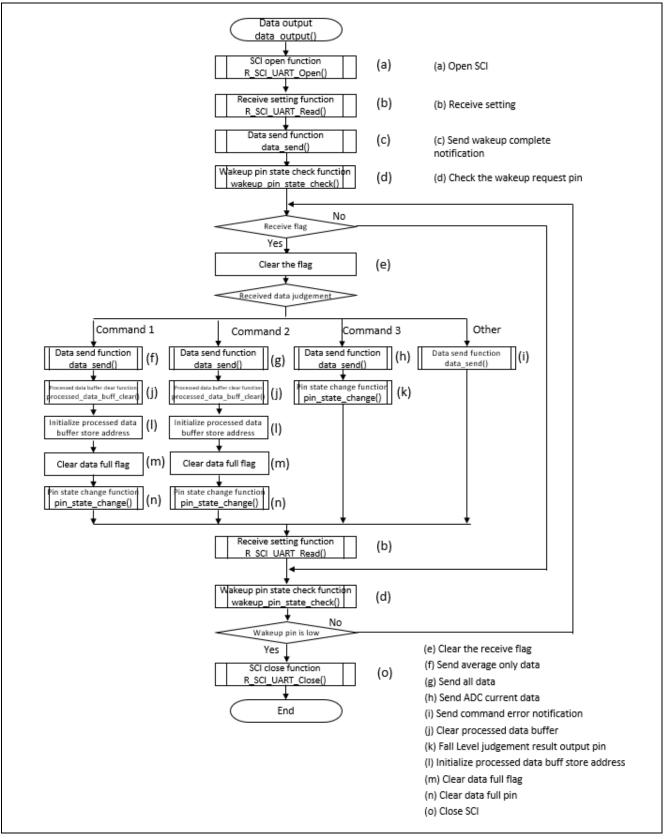Figure 3.18 shows the flow of data output processing.

Figure 3.18.   Data output processing

4.   Evaluating Applications

4.1   Import and Building a Project

To build an application project with e² studio IDE, proceed as follows:

1. Launch e² studio IDE.
2. Select any workspace in Workspace launcher.
3. Close **Welcome** window.
4. Select **File > Import**.
5. Select **Existing Projects into Workspace** from the **Import** dialog box.
6. Select archive file.
7. Select the project you want to import and click **Finish**.
8. Clicking **Generate Project Content** in the **Configurator** window
9. Select **Project > Build Project**.

4.2   Script Settings for Low Power Mode Debugging

Change the debug settings to debug applications using low power modes. If you use J-Link to debug low power modes for RA devices, the default debug configuration does not allow debugging. If the application tries to enter Software Standby mode, the connection between the CPU and the IDE is disconnected and the IDE debug session is terminated. You can continue to debug in low power modes by specifying a debug script and configuring low power handling option as shown in Figure 4.1.

This debug script is provided to develop applications. Therefore, **please note that accurate Icc values cannot be measured when measuring the current consumption Icc**. Disable OCD when measuring.

The script file is included with this application project and can be used by specifying it in the debug settings.
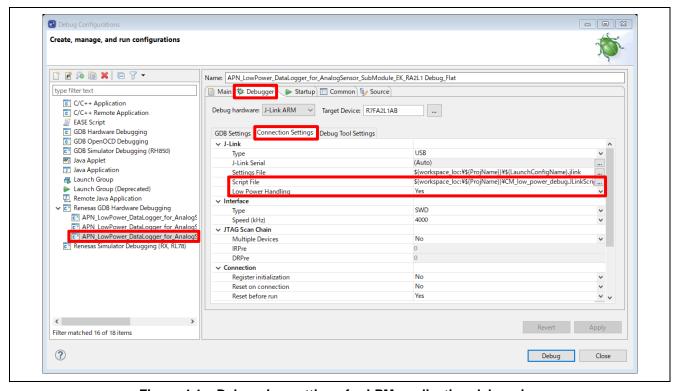

Figure 4.1.   Debugging settings for LPM application debugging

4.3   Evaluating the Current Consumption

The EK-RA2L1 kit provides a current measuring resistor with a resistance of 5 mΩ and a test point (TP1 and TP3) for measuring the current of the MCU power of 3.3 V. The current consumption can be calculated by measuring the voltage drop across this resistor and using Ohm's law. See Figure 4.2 and Figure 4.3 for the location of TP1 and TP3.
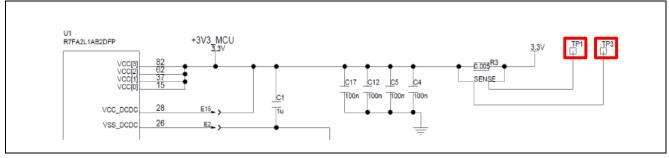
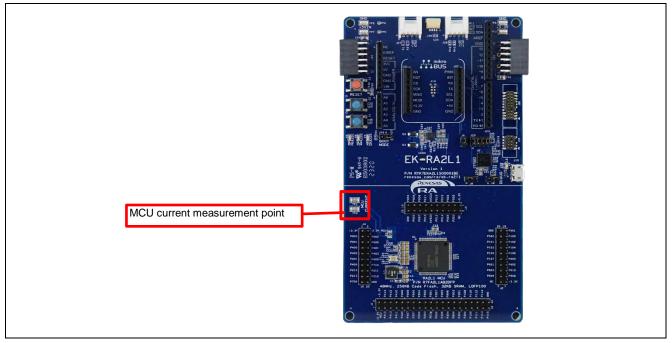**Figure 4.2.   MCU Current Measuring Points (Circuit Diagram)**

MCU current measurement point

Figure 4.3.   MCU Current Measuring Points

4.4   Checking the operation of the low-power data logger

This chapter describes the main module program for checking the operation of this application project. The main module acquires data using UART communication when the user switch is pressed or when it receives the event signal from the sub-module. The acquired data is displayed using Segger J-Link RTT Viewer.

4.4.1   Overview

This application operates as a main module and acquires data using UART communication from a sub-module that stores sensor data at low power consumption. Data acquisition is performed when each event occurs. The acquired data is displayed on J-Link RTT Viewer on the PC.

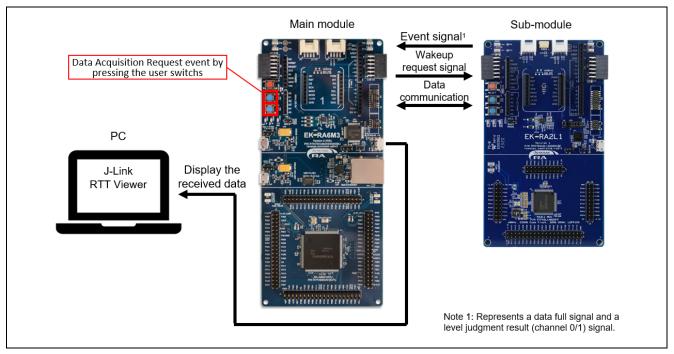Figure 4.4 shows a schematic diagram of the operation check method.

Figure 4.4. Overview

4.4.2 Description of Operation

This application acquires data from the sub-module in response to each event of the data acquisition request. As shown in Table 4.1 and Table 4.2, the data acquisition request events are 3 types of event signal reception from the sub-module and two types of user switch presses of the main module. When each event occurs, the communication sequence shown in Figure 3.4 is executed to acquire data from the sub-module. The results of the analysis of the acquired data are displayed on J-Link RTT Viewer.

For details on the format of communication data packets, refer to 3.2.3, Data Communication Specifications. Also, if the Disable Sensor Dependent Process function is enabled in the sub-module, the application of the main module should also define the `DISABLE_SENSOR_DEPENDENCE` macro in `app_common.h` to enable this function.

The processing sequence at the rising edge of the level judgment result (channel 0) pin is shown below.

1. When a rising event of the level judgment result (channel 0) pin is detected, output a high level to the wakeup request pin.
2. After receiving the wakeup completion notification from the sub-module, send the data request command according to the event.
3. When data is received, the data packet is analyzed, and the result is displayed in J-Link RTT Viewer.
4. Outputs a low level to the wakeup request pin.

Figure 4.5 shows the overall algorithm of the main module.

Table 4.1. Request data for sub-module generated events

Event	Requested data
Rising edge of data full pin	All data (Average, Minimum, Maximum)
Rising edge of level judgment result (channel 0) pin	ADC current data
Rising edge of level judgment result (channel 1) pin	ADC current data

Table 4.2. Request data for main module generated events

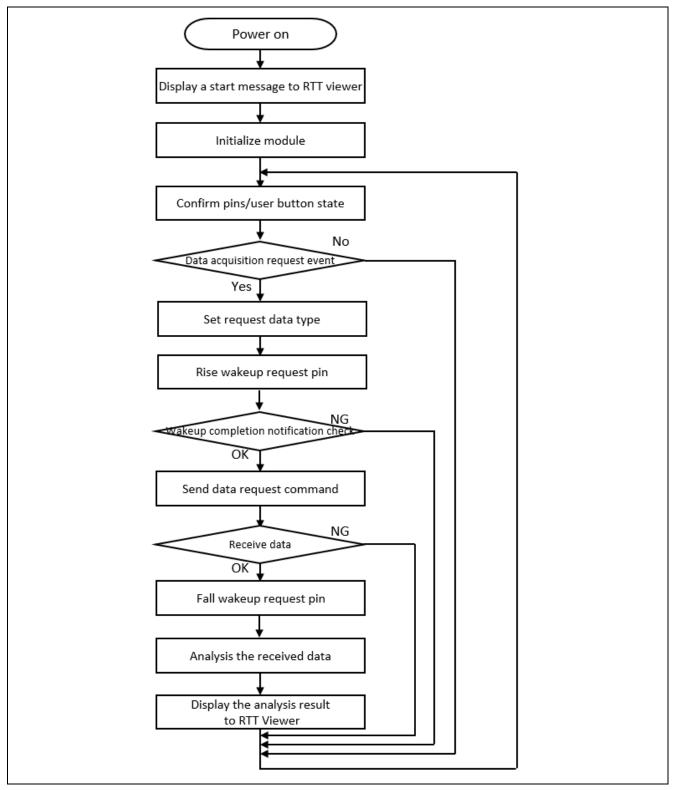| Event	Requested data
User switch S1 pressed	Average only data
User switch S2 pressed	All data (Average, Minimum, Maximum)

Figure 4.5.   Overall algorithm of main module

4.4.3   User Interface of Main Module

4.4.3.1   Connecting the Sub-module

Table 4.3 and Figure 4.6 show the connection pins to the sub-module. Connect PMOD2 connector of EK-RA6M3 kit using jumper wires.

Table 4.3. Pins connected to sub-module

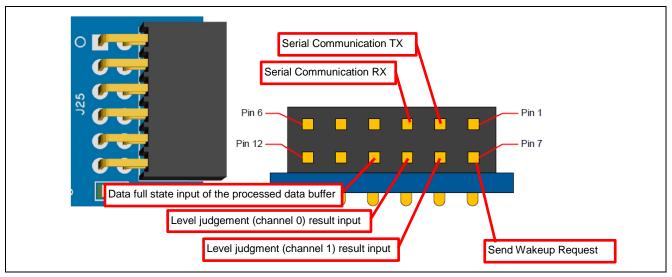| RA6M3 pin	EK-RA6M3 Pin	Typical Uses
P105/TxD8	PMOD2-2	Serial communication (transmission)
P104/RxD8	PMOD2-3	Serial communication (reception)
P708	PMOD2-7	Send Wakeup Request
P803	PMOD2-8	Input level judgment result (channel 1)
P804	PMOD2-9	Input level judgment result (channel 0)
P805	PMOD2-10	Input data full state of the processed data buffer

Figure 4.6.   Connecting to sub-module (PMOD2)

4.4.3.2 User Switch

In this application, data acquisition can be executed at any time by pressing the user switch.

Table 4.4 lists the user switch connection pins, and Figure 4.7 shows the user switch locations.

Table 4.4.   User switches

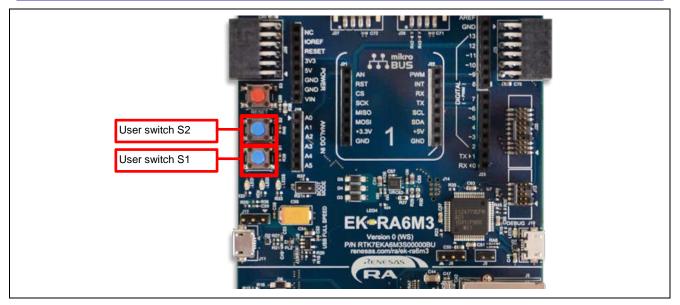| RA6M3 pin	EK-RA6M3 connecting component	Typical Uses
P009/IRQ13	User switch S1	Average only data acquisition event
P008/IRQ12	User switch S2	All data acquisition event

Figure 4.7. Locations of the user switch

4.4.3.3 On-board LEDs

In this application, the status of the pins connected to the sub-module can be checked by checking the LEDs mounted on EK-RA6M3.

Table 4.5 lists the connection pins and checks of the onboard LEDs, and Figure 4.8 shows the location of the onboard LEDs.

Table 4.5. On-board LEDs

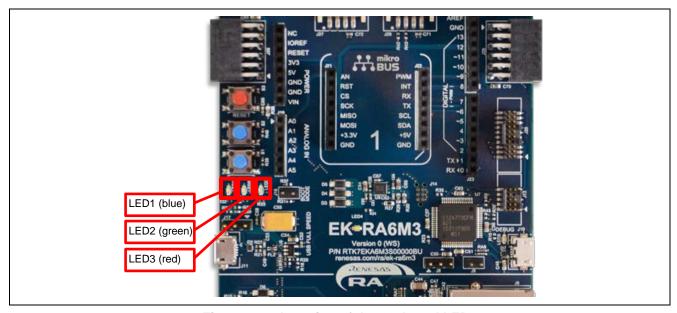| RA6M3 pin	EK-RA6M3 connecting component	Typical Uses
P403	LED1 (blue)	Check the status of the wakeup request pin
P400	LED2 (green)	Check the status of the level judgment result (channel 0/1) pin.
P100	LED3 (red)	Check the status of the data full pin

Figure 4.8. Location of the On-board LEDs

4.4.4 Connecting J-Link RTT Viewer

This application uses Segger J-Link RTT Viewer for outputting the data received from sub-module. The J-Link RTT Viewer can be connected as follows.

1. Launch J-Link RTT Viewer.
2. Select **R7FA6M3AH** in **Specify Target Device** in the **Configuration** dialogue.
3. Select **OK**.

If the project is running and a startup message is displayed as in Figure 4.9, the connection is successful.
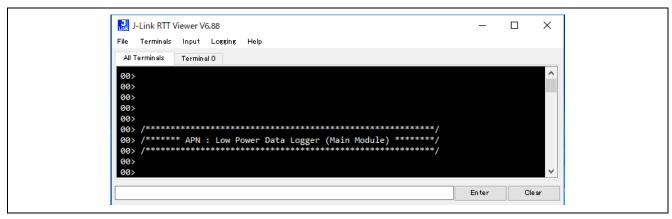

Figure 4.9. Startup message on J-Link RTT Viewer

4.4.5 Checking Procedure

The procedure for checking the operation of the sub-module is shown below.

1. Connect the main module (EK-RA6M3 kit) and sub-module (EK-RA2L1 kit or EK-RA2E1 kit) with jumper wires, referring to Table 4.3 and connection pins on the sub-module side.
2. Power on both modules.
3. Connect the J10 connector of the main module and PC with the USB cable.
4. Connect the Main Module to J-Link RTT Viewer.
5. Confirm the operation status of the sub-module.

The sub-module enters the data full state every 50 hours (every 50 minutes if fast cycle debug is enabled) and performs data acquisition (see Figure 4.10). In addition, data acquisition is executed when the values of the illuminance sensor and temperature sensor exceed the threshold value (see Figure 4.11). The acquired data can be viewed on J-Link RTT Viewer. In addition, data acquisition can be performed at any time by pressing the user switch S1 or S2 on the main module (see Figure 4.12).
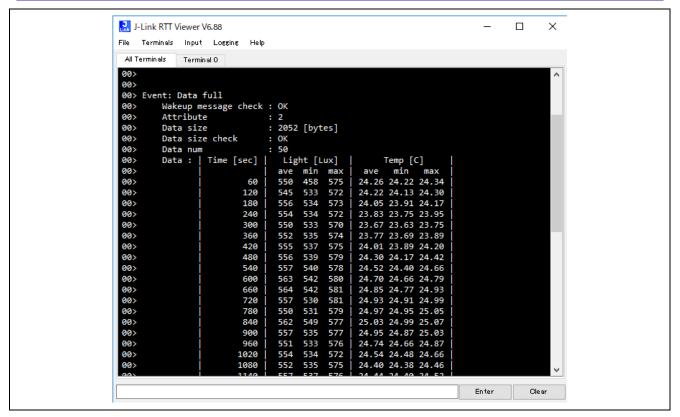
Figure 4.10.   Example of data acquisition triggered by data full state

Figure 4.11.   Example of data acquisition triggered by analog level judgment result

Figure 4.12.   Example of data acquisition triggered by pressing the button on the main module

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 EK-RA2L1

Website and Support

- RA Product information   www.renesas.com/ra
- RA Product Support Forum   www.renesas.com/ra/forum
- RA Flexile Software Package   www.renesas.com/FSP
- Renesas Support   www.renesas.com/support

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Jan.27.21	-	Initial version

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/.

© 2021 Renesas Electronics Corporation. All rights reserved.