RA2A2 Group

Board Control Program for QE for AFE

Introduction

The control program operates on RA2A2 on the Evaluation Kit for RA2A2E (EK-RA2A2), communicates commands with the AFE development support tool ‘QE for AFE’, and can set registers for analog IP and obtain the A/D conversion values as shown below.

— 24-Bit Sigma-Delta A/D Converter (SDADC24)
Hi-Speed USB-UART conversion adapter enables continuous measurement at data rate in 7 channels / 8 kHz sampling setting.

— 12-Bit A/D Converter (ADC12)

Target Device

RA2A2 (R7FA2A2AD3CFP)

Board to Be Operated

EK-RA2A2 for RA2A2 Microcontroller Group

Available Communication I/F:

— SCI UART Communication: A separate USB-UART conversion adapter is required.
Hi-Speed 12Mbps
USB 2.0 Hi-Speed to UART Cable C232HD- EDHSP-0 manufactured by Future Technology Devices International (FTDI)
(Hereafter abbreviated as Hi-Speed 12Mbps USB-UART conversion adapter)
3Mbps
PMOD I/F: FTD, PmodUSBUART™ manufactured by FTDI
mikroBUS™ I/F: USB UART click (MIKROE-1203) manufactured by MikroElektronika
(Hereinafter, both are abbreviated as 3Mbps USB-UART conversion adapter)

The SDADC24 is capable of high-data rate output, so when using SDADC24, set the connection Bitrate to 3 Mbps or higher. For details, refer to "Table 1-9 SDADC24: Sampling Mode, Number of Channels and Supported Bitrate when Using UART".

Contents

1. Overview

This control program is intended to be used in combination with 'QE for AFE'.

Therefore, refer to the 'QE for AFE' document and use this control program.

1.1 System Overview

This control program (hereinafter abbreviated as 'FW') operates on RA2A2 on the EK-RA2A2.

You can communicate using 'QE for AFE' via SCI UART and control the following according to the command request from 'QE for AFE':

— Register Settings for 24-Bit Sigma-Delta A/D Converter (SDADC24)
— Register Settings for 12-Bit A/D Converter (ADC12)

The differences in specifications/functions due to differences in connection methods are shown below. Select the connection method according to your purpose.

For the specific connection method, refer to '2.2.1 Connecting to PC'.

Table 1-1 Connection Method and Differences in Specifications/Functions

<table>
<tr><th colspan="2">Items</th><th colspan="2">SCI UART</th></tr>
<tr><td colspan="2">System Clock</td><td>HOCO 64MHz
XTAL 12MHz</td><td>HOCOC 48MHz</td></tr>
<tr><td colspan="2">ICLK Frequency (max.)</td><td>32MHz: divided HOCO</td><td>48MHz: divided HOCO</td></tr>
<tr><td colspan="2">PCLKB Frequency (max.)</td><td>32MHz: divided HOCO</td><td>24MHz: divided HOCO</td></tr>
<tr><td colspan="2">PCLKD(=ADCLK) Frequency (max.)</td><td>32MHz: divided HOCO</td><td>48MHz: divided HOCO</td></tr>
<tr><td rowspan="3">ADC12</td><td>ADCLK Frequency (max.)</td><td>32MHz [Note 2]</td><td>48MHz</td></tr>
<tr><td>Continuous Measurement of A/D conversion [Note 1]</td><td colspan="2">Support under specific conditions</td></tr>
<tr><td>One-shot Measurement of A/D conversion [Note 1]</td><td colspan="2">Supported</td></tr>
<tr><td>SDADC24</td><td>Continuous Measurement of A/D Conversion [Note 1]</td><td colspan="2">Supported</td></tr>
</table>

Note 1: For details, refer to '1.2 Measured Value Transmission Operation during A/D Conversion and Comparison'

Note 2: For EK-RA2A2, because AVCC = 3.3V.

1.1.1 About the Included HEX Files and e2 studio Projects

The overview of the included HEX files and the e2 studio projects are shown below.

(1) HEX Files

Evaluation is possible by writing the HEX file to the EK-RA2A2.

For the FW writing method, refer to ‘2.1.1 Writing using Renesas Flash Programmer’.

Also, refer to ‘1.4 Operation Confirmation Environment’ for the settings when creating a HEX file.

For information on the port assignments, refer to ‘Table 3-2 Pins used List’.

Table 1-2 HEX Files

	HEX Files	Destination
UART0	RTS/CTS flow control [Note 1] • PCLKB frequency: 32MHz • PCLKD frequency: 32MHz	Hex └─ek_ra2a2 └─ek_ra2a2_uart0_32MHz_rev210.hex
	Without RTS/CTS flow control [Note 2] For mikroBUS • PCLKB frequency: 32MHz • PCLKD frequency: 32MHz	Hex └─ek_ra2a2 └─ek_ra2a2_uart0nohwflow_32MHz_rev210.hex
	RTS/CTS flow control [Note 3] • PCLKB frequency: 24MHz • PCLKD frequency: 48MHz	Hex └─ek_ra2a2 └─ek_ra2a2_uart0_24MHz_rev210.hex
	RTS/CTS flow control [Note 4, 8] • PCLKB frequency: 12MHz • PCLKD frequency: 12MHz	Hex └─ek_ra2a2 └─ek_ra2a2_uart0_12MHz_rev210.hex
	RTS/CTS flow control [Note 5, 8] • PCLKB frequency: 8MHz • PCLKD frequency: 8MHz	Hex └─ek_ra2a2 └─ek_ra2a2_uart0_8MHz_rev210.hex
UART3	RTS/CTS flow control [Note 6] • PCLKB frequency: 32MHz • PCLKD frequency: 32MHz	Hex └─ek_ra2a2 └─ek_ra2a2_uart3_32MHz_rev210.hex
	RTS/CTS flow control [Note 7, 8] • PCLKB frequency: 8MHz • PCLKD frequency: 8MHz	Hex └─ek_ra2a2 └─ek_ra2a2_uart3_8MHz_rev210.hex

Note 1: This is the same setting as "ek_ra2a2_uart0_32MHz" in "Table 1-3 e2 studio Projects".
Note 2: This is the same setting as " ek_ra2a2_uart0nohwflow_32MHz" in "Table 1-3 e2 studio Projects".
Note 3: This is the same setting as "ek_ra2a2_uart0_24MHz" in "Table 1-3 e2 studio Projects".
Note 4: This is a setting in which the clock frequency is changed based on "ek_ra2a2_uart0_24MHz" in "Table 1-3 e2 studio Projects"..
Note 5: This is a setting in which the clock frequency is changed based on "ek_ra2a2_uart0_32MHz" in "Table 1-3 e2 studio Projects"..
Note 6: This is the same setting as "ek_ra2a2_uart3_32MHz" in "Table 1-3 e2 studio Projects".
Note 7: This is a setting in which the clock frequency is changed based on "ek_ra2a2_uart3_32MHz" in "Table 1-3 e2 studio Projects"..
Note 8: This is the Hex file for ADC12 continuous measurement. For details, see "1.2 Measured Value Transmission Operation during A/D Conversion and Comparison". When ushing SDADC24 continuous measurement, Use one with a high PCLKB frequency.

(2) e2 studio Projects

Evaluation is possible by importing the e2 studio project and writing it to the EK-RA2A2.

It is provided with the setting to be automatically executed when the **'Build'** is executed,  is pressed, and the **'Debug'** mode is launched.

For the FW writing method, refer to '3.4 Write and Build Using e2 studio Integrated Development Environment (IDE)'.

Also, refer to '1.4 Operation Confirmation Environment' for the settings when creating a HEX file.

For information on the port assignments, refer to 'Table 3-2 Pins used List'.

Note that HEX files of 'Table 1-2 HEX Files' were created using the following projects or the clock frequency was changed.

Table 1-3 e2 studio Projects

	e2 studio Project	Destination
UART0	RTS/CTS flow control • PCLKB frequency: 32MHz • PCLKD frequency: 32MHz	ek_ra2a2 └─ek_ra2a2_uart0_32MHz
	Without RTS/CTS flow control For mikroBUS • PCLKB frequency: 32MHz • PCLKD frequency: 32MHz	ek_ra2a2 └─ek_ra2a2_uart0nohwflow_32MHz
	RTS/CTS flow control • PCLKB frequency: 24MHz • PCLKD frequency: 48MHz	ek_ra2a2 └─ek_ra2a2_uart0_24MHz
UART3	RTS/CTS flow control • PCLKB frequency: 32MHz • PCLKD frequency: 32MHz	ek_ra2a2 └─ek_ra2a2_uart3_32MHz

1.2 Measured Value Transmission Operation during A/D Conversion and Comparison

During A/D conversion, the measured values are sent to the PC for Continuous measurement. This measurement is defined as **'Continuous measurement'**.

When using ADC12 (Analog Input Pins), which is capable of high output data rate, the transmission process may not be in time. In that case, after acquiring the specified number of A/D values, A/D conversion is stopped and the acquired A/D values are sent together. This measurement is defined as **'One-shot measurement'**.

1.2.1 When Using SCI UART

Table 1-4 shows the measurement operation for each measurement target.

Table 1-4 Measurement Operation of SDADC24, ADC16, ACMPHS, ACMPLP

Measurement Target	Measurement Operation
SDADC24	**Continuous measurement**
ADC12 (Analog Input Pins)	Switched to **Continuous measurement** or **One-shot measurement** depending on operating conditions "Table 1-6"
ADC12 (Temperature Sensor Output)	Single measurement with single scan
ADC12 (Internal Reference Voltage)	

(1) ADC12

Switching between **Continuous measurement** and **One-Shot measurement** operating conditions and precautions are indicated.

(a) ADC12 Continuous Measurement

Table 1-5 shows the continuous measurement operating conditions of ADC16 when using SCI UART and USB PCDC. The FW switches to the Continuous measurement or One-shot measurement depending on both PCLKB frequency and the output data rate setting of ADC16.

Table 1-5 Continuous Measurement Operating Conditions of ADC16 when Using SCI UART and USB PCDC

PCLKB Frequency [Note1]	ADSSTRn (n=00 – 08, L) [Note1]
12MHz	All ADSSTR values of ch to use: 0x8A (138) or more
Under 12MHz	All ADSSTR values of ch to use: 0x78 (120) or more

Note 1: Operating condition settings assuming PCLKD frequency = PCLKB frequency.
If PCLKD frequency > PCLKB frequency, the continuous measurement setting will be set, but the number of measurement data will be too large, and the transmission process will stop.

Table 1-6 shows an example of settings for which continuous measurement operation has been confirmed. These settings do not guarantee the continuous measurement operation.

Table 1-6 ADC12 Continuous Measurement Operation Confirmed Setting Example

System Clock	Communication I/F	ICLK Frequency	PCLKB Frequency = PCLKD Frequency	ADSSTRn (n=0 – 3, L) of Ch to Use	UART Bitrate Setting [bps]	Measurement Time [minutes]	Remarks
HOCO 48MHz	UART	48MHz	12MHz	All 0xB4(180) or more [Note 3]	2,000,000 [Note 1]	About 16	
HOCO 64MHz		32MHz	8MHz	All 0x78(120) or more [Note 4]	1,333,333 [Note 1]		[Note 2]

Note 1: If the value other than the specified value is set, FW will not be able to measure because the transmission will not be in time.
Note 2: It is the setting of the HEX file 'ek_ra2a2-uart0-8MHz-rev210.hex' shown in "Table 1-2 HEX Files".

Note 3: In the QE for AFE setting, it is possible to input from “11.5 [us]”, but “15 [us]” or more is recommended. If the value is less than “15 [us]”, drawing may stop in less than 1 minute.
Note 4: In the QE for AFE setting, input the following.
If PCLKB = PCLKD = 8MHz, input “15 [us]” or more.

In addition, QE for AFE may miss data depending on PC environment during the Continuous measurement operation. In that case, the following error will be displayed on QE for AFE.

```
[Error]Some data missed in communication. Please check missing data in [Raw Data] view.
```

Figure 1-1 Error Message when Data Is Missed

Therefore, please evaluate it when other applications are stopped or network offline. If data is still missing, increase the ADSSTR value and lower the output data rate.

The following shows the relationship between the number of channels and the UART Bitrate. These settings do not guarantee the continuous measurement operation.

Table 1-7 ADC12: Number of Channels and Supported Bitrate when Using UART

Number of Channels	Bitrate [bps]		Remarks
	2M	1.333333M	
1 to 4	Available [Note 1] All 0xB4(180) or more	Available [Note 2] All 0x78(120) or more	

Note 1: It is the setting of the HEX file ‘ek_ra2a2-uart0-12MHz-rev210.hex’ shown in “Table 1-2 HEX Files”. In the QE for AFE setting, it is possible to input from “11.5 [us]”, but “15 [us]” or more is recommended. If the value is less than “15 [us]”, drawing may stop in less than 1 minute.
Note 2: It is the setting of the HEX file ‘ek_ra2a2-uart0-8MHz-rev210.hex’ shown in “Table 1-2 HEX Files”.

(b) ADC12 One-Shot Measurement

The number of measured values and the time of measurement that can be acquired in the case of One-shot measurement settings are shown below.

Table 1-8 Number of Storable Measured and Measurement Time for ADC12 One-shot Measurement

Items	Contents
Number of measured values for One-shot measurement	8,192 (Max.) **Number of Measured Values of Each Channel** = 8,192/(Number of measurement channels); (Rounded down)
Measurement Time of One-shot measurement	Sum of [(**Number of Measured Values of Each Channel)** x (sampling time of each channel)]

When combined with QE for AFE, intermittent measurement [Auto] that repeats One-shot measurement is possible.

Note that in the case of this measurement, the start and end of measurement are repeated, so the start of the measurement after the end of the measurement (e.g., the 8192nd measurement) and the start of the next measurement (e.g., the 8193rd measurement) is not continuous data because there is a gap in the measurement time.

(2) SDADC24

The operating conditions and precautions for continuous measurement are indicated.

(a) SDADC24 Continuous Measurement

The SDADC24 is capable of high-output data rate output. Therefore, a high Bitrate setting of the UART is required to achieve continuous measurement operation.

The following shows the relationship between the sampling mode, the number of channels, and the UART Bitrate. These settings do not guarantee the continuous measurement operation.

Select a USB-UART conversion adapter and set the maximum possible Bitrate according to the SDADC24 measurement conditions.

Table 1-9 SDADC24: Sampling Mode, Number of Channels and Supported Bitrate when Using UART

<table>
<tr><th rowspan="2">Sampling Mode</th><th rowspan="2">Number of Channels</th><th colspan="3">Connection Bitrate [bps] [Note 1]</th><th rowspan="2">Remarks</th></tr>
<tr><th>3M</th><th>4M</th><th>5.333333M</th></tr>
<tr><td>4kHz</td><td>1 to 4</td><td rowspan="2">Available</td><td rowspan="3">Available</td><td rowspan="10">Available</td><td></td></tr>
<tr><td rowspan="4">8kHz</td><td>1 to 4</td><td></td></tr>
<tr><td>5</td><td rowspan="3">Not Available</td><td></td></tr>
<tr><td>6</td><td rowspan="2">Not Available</td><td></td></tr>
<tr><td>7</td><td></td></tr>
<tr><td rowspan="3">8kHz/4kHz
Hybrid</td><td>1 to 2</td><td>Available</td><td rowspan="2">Available</td><td></td></tr>
<tr><td>3</td><td>Available
[Note 3]</td><td></td></tr>
<tr><td>4</td><td>Not Available</td><td>Available
[Note 2]</td><td></td></tr>
</table>

Note 1: For details on the USB-UART conversion adapter, refer to 'Table 1-11 Operating Confirmation Conditions' and 'Table 1-12 Clock Settings and Tools to Use (SCI UART)'.
For Bitrate, refer to 'Table 3-3 Serial Communication Settings'.

Note 2: If no data is output on the GUI, reduce the number of measurement channels.

Note 3: It depends on the FW you use.
ek_ra2a2_uart0_32MHz_rev210.hex: Not available
ek_ra2a2_uart0_24MHz_rev210.hex: Available. If no data is output on the GUI, reduce the number of measurement channels.

Note that during continuous measurement operation, there is a possibility that QE for AFE may miss data depending on the PC environment. In this case, the following error is displayed on QE for AFE:

```
[Error]Some data missed in communication. Please check missing data in [Raw Data] view.
```

Figure 1-2 Error Message when Missing Data

Therefore, please evaluate it in the stopped state of other applications or in the network/offline state. If data is still missing, reduce the number of configured channels.

The measurement times for the continuous measurement setup are shown below.

Table 1-10 SDADC24 Measurement Time when Setting up Continuous Measurement

Items	Descriptions
Measurement Time of Continuous Measurement	5 minutes (fixed regardless of the number of channels and sampling mode setting)

(b) Notes on SDADC24 8kHz/4kHz Hybrid sampling Mode measurement

When measuring in 8kHz/4kHz Hybrid sampling Mode, there will be a time difference between the measured values of 8kHz sampling data and 4kHz sampling data. This is due to the difference in delay time due to the difference in the circuit configuration of SDADC24.

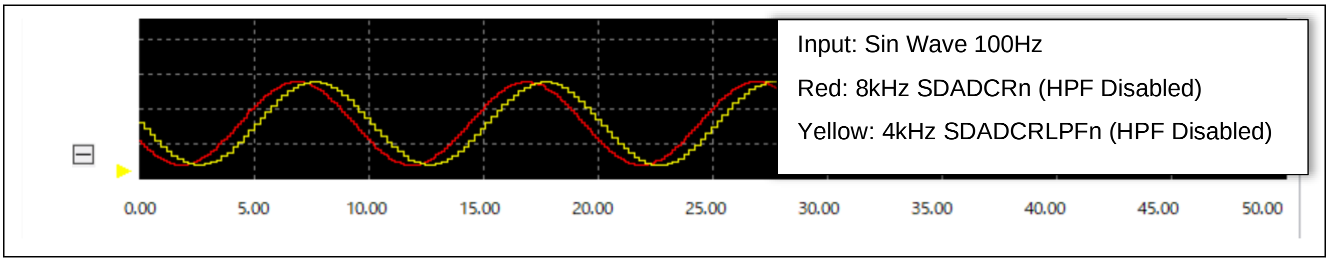


Figure 1-3 Waveform example of time difference between measured values of 8kHz sampling data and 4kHz sampling data during 8kHz/4kHz Hybrid sampling

1.3 File Configurations

The following is a list of file configurations. The description of some folders and files is omitted.

```
r01an7280xx0210-ra2a2-serial
├─ek_ra2a2
│  └─ek_ra2a2_uart0_32MHz                e2 studio project for SCI UART0 PCLKB_32MHz
│  │   ├─.settings
│  │   ├─ra
│  │   ├─ra_cfg
│  │   ├─ra_gen
│  │   ├─script
│  │   └─src
│  │     ├─hal_entry.c
│  │     ├─hal_entry.h
│  │     ├─r_common_config.h
│  │     ├─r_common_func.c
│  │     ├─r_common_func_if.h
│  │     ├─r_communication_control_api.c
│  │     ├─r_communication_control_api_if.h
│  │     ├─r_interrupt_callback.c
│  │     ├─r_interrupt_callback_if.h
│  │     ├─r_reg_write.c
│  │     └─r_reg_write_if.h
│  │
│  ├─ek_ra2a2_uart0noflow_32MHz          e2 studio project for SCI UART0 without RTS/CTS Flow Control PCLKB_32MHz
│  │    The folder structure and file structure are the same as the ek_ra2a2_uart0_32MHz, so they are omitted.
│  │
│  ├─ek_ra2a2_uart0_24MHz                e2 studio project for SCI UART0 PCLKB_24MHz
│  │    The folder structure and file structure are the same as the ek_ra2a2_uart0_32MHz, so they are omitted.
│  │
│  └─ek_ra2a2_uart3_32MHz                e2 studio project for SCI UART3 PCLKB_32MHz
│       The folder structure and file structure are the same as the ek_ra2a2_uart0_32MHz, so they are omitted.
│
├─Hex
│  ├─ek_ra2a2
│  │  ├─ek_ra2a2_uart0_32MHz_rev210.hex          Hex file for UART0 PCLKB=PCLKD=32MHz
│  │  ├─ek_ra2a2_uart0_24MHz_rev210.hex          Hex file for UART0 PCLKB=24MHz, PCLKD=48MHz
│  │  ├─ek_ra2a2_uart0_12MHz_rev210.hex          Hex file for UART0 PCLKB=PCLKD=12MHz
│  │  ├─ek_ra2a2_uart0_8MHz_rev210.hex           Hex file for UART0 PCLKB=PCLKD=8MHz
│  │  ├─ek_ra2a2_uart0noflow_32MHz_rev210.hex    Hex file for UART0 PCLKB=PCLKD=32MHz without RTS/CTS Flow Control
│  │  ├─ek_ra2a2_uart3_32MHz_rev210.hex          Hex file for UART3 PCLKB=PCLKD=32MHz
│  │  └─ek_ra2a2_uart3_8MHz_rev210.hex           Hex file for UART3 PCLKB=PCLKD=8MHz
│  └─ek_ra2a1
│     ├─ek_ra2a1-uart-32MHz-rev200.hex           Hex file for UART0 PCLKB=PCLKD=32MHz
│     ├─ek_ra2a1-uart-8MHz-rev200.hex            Hex file for UART0 PCLKB=PCLKD=8MHz
│     ├─ek_ra2a1-usb-24MHz-rev200.hex            Hex file for USB PCLKB=PCLKD=24MHz
│     └─ek_ra2a1-usb-12MHz-rev200.hex            Hex file for USB PCLKB=PCLKD=12MHz
│
├─r01an7280ej0210-ra2a2-serial.pdf
└─r01an7280jj0210-ra2a2-serial.pdf
```

Figure 1-4 File Configurations

1.4 Operation Confirmation Environment

This FW is confirmed under the operating conditions shown in Table 1-11 and Table 1-12.

The settings of the included HEX file are shown below.

Table 1-11 Operating Confirmation Conditions

Items	Descriptions
MCU	R7FA2A2AD3CFP (Renesas RA2A2 MCU Group) Supply voltage: 3.3V
IDE	Renesas e2 studio V2024-01.1 (24.1.1)
FSP	v5.2.0
Tool Chain	GNU ARM Embedded 13.1.1.arm-13-7
Emulator	SEGGER J-Link®
FW Writing Tool	Renesas Flash Programmer V3.11.01
USB-UART conversion adapter	Hi-Speed 12Mbps USB-UART conversion adapter • USB 2.0 Hi-Speed to UART Cable C232HD-EDHSP-0 3Mbps USB-UART conversion adapter • PMOD I/F: PmodUSBUART • mikroBUS I/F: USB UART click (MIKROE-1203)
FTDI Driver for PC	Virtual COM port (VCP) drivers V2.12.36.4 URL: VCP Drivers - FTDI (ftdichip.com)

Table 1-12 Clock Settings and Tools to Use (SCI UART)

<table>
<tr><th>Items</th><th colspan="5">SCI UART</th></tr>
<tr><td>System Clock</td><td colspan="3">HOCO 64MHz</td><td colspan="2">HOCO 48MHz</td></tr>
<tr><td>ICLK</td><td colspan="3">32MHz</td><td colspan="2">48MHz</td></tr>
<tr><td>PCLKB [Note 3]</td><td>32MHz</td><td>16MHz</td><td>8MHz</td><td>24MHz</td><td>12MHz</td></tr>
<tr><td>PCLKD [Note 3]</td><td>32MHz</td><td>16MHz</td><td>8MHz</td><td>48MHz</td><td>12MHz</td></tr>
<tr><td>FCLK</td><td colspan="3">32MHz</td><td colspan="2">24MHz</td></tr>
<tr><td>SDADCCLK Clock Source</td><td colspan="3">HOCO/4(=16MHz)</td><td colspan="2">HOCO/4(=12MHz)</td></tr>
<tr><td>USB-UART Conversion Adapter</td><td colspan="5">• Hi-Speed 12Mbps USB-UART conversion adapter:
use for 5.333333Mbps or less
• 3Mbps USB-UART conversion adapter:
use for 3Mbps or less</td></tr>
<tr><td>HEX File</td><td>Included
[Note 1]</td><td></td><td>Included
[Note 1]</td><td>Included
[Note 1]</td><td>Included
[Note 1]</td></tr>
<tr><td>e2 Project/
Folder name</td><td>Included
[Note 2]</td><td></td><td></td><td>Included
[Note 2]</td><td></td></tr>
</table>

Note 1: Refer to 'Table 1-2 HEX Files'.

Note 2: Refer to 'Table 1-3 e2 studio Projects'.

Note 3: If the frequency is set to less than 8 MHz, it may not work.

1.5 Related Documentation

- Renesas RA2A2 Group User's Manual: Hardware (R01UH1005)
- Renesas RA2A2 Group Evaluation Kit for RA2A2 Microcontroller Group EK-RA2A2 v1 User's Manual (R20UT5333

2. How to Use

2.1 How to Write FW

There are two ways to write the FW to the EK-RA2A2.

- Writing HEX files using Renesas Flash Programmer
  — References to Connecting to PC: ‘2.1.1(1)EK-RA2A2 writing preparation’
  — References to Writing using Renesas Flash Programmer: ‘2.1.1(2)Launch the Renesas Flash Programmer and Communication Settings’

- Writing a project using the e2 studio integrated development environment (IDE)
  — References to Connecting to PC：‘2.2.1Connecting to PC’
  — References to Debugging operation: ‘3.4Write and Build Using e2 studio Integrated Development Environment (IDE)’

2.1.1 Writing using Renesas Flash Programmer

You can write HEX files to the RA2A2 on the EK-RA2A2 using Renesas Flash Programmer. For the included HEX files, refer to ‘For information on the port assignments, refer to ‘Table 3-2 Pins used List’.

Table 1-2 HEX Files’.

Get the Renesas Flash Programmer V3.11.01 or later that supports RA Family from the following.

https://www.renesas.com/software-tool/renesas-flash-programmer-programming-gui

The operation procedure when using Renesas Flash Programmer V3.11.01 is shown below.

(1) EK-RA2A2 writing preparation

Use the on-board debug mode of the EK-RA2A2's USB debugging I/F.

By default, the jumper is enabled with the onboard debugger.

For details on how to set the jumper, refer to "5.2 Debugging" in the "RA2A2 MCU Group Evaluation Kit User's Manual".

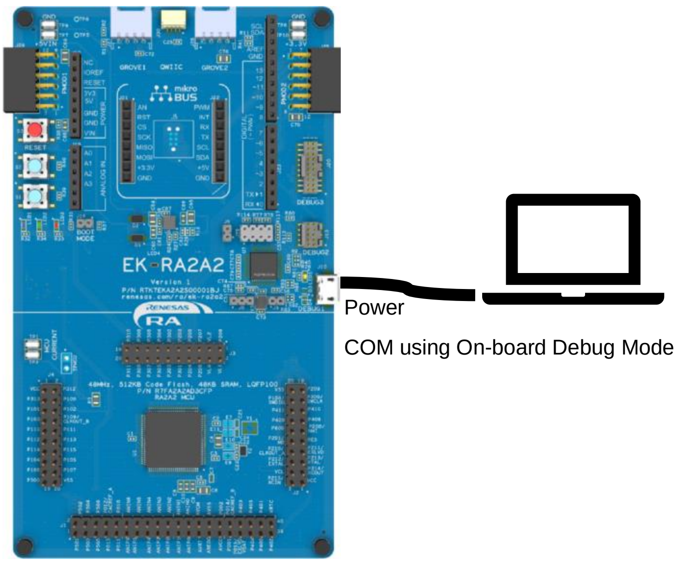


Figure 2-1 How to connect to a PC when writing a HEX file

(2) Launch the Renesas Flash Programmer and Communication Settings

The following two cases are described.

Without a Renesas Flash Programmer project
Renesas Flash Programmer Project has been Created

(a) If No Renesas Flash Programmer Project for RA Family Has Been Created

1. Create a New Project.

Click ‘New Project ...’ in ‘File’.

Set the ‘Microcontroller’ in ‘Project Information’ to “RA”.

Set the ‘Tool’ of ‘Communication’ to "J-Link" and ‘Interface’ to “2 wire UART”.

Click ‘Connect’.

Renesas Flash Programmer starts the connection process. Confirm that the connection is successful.

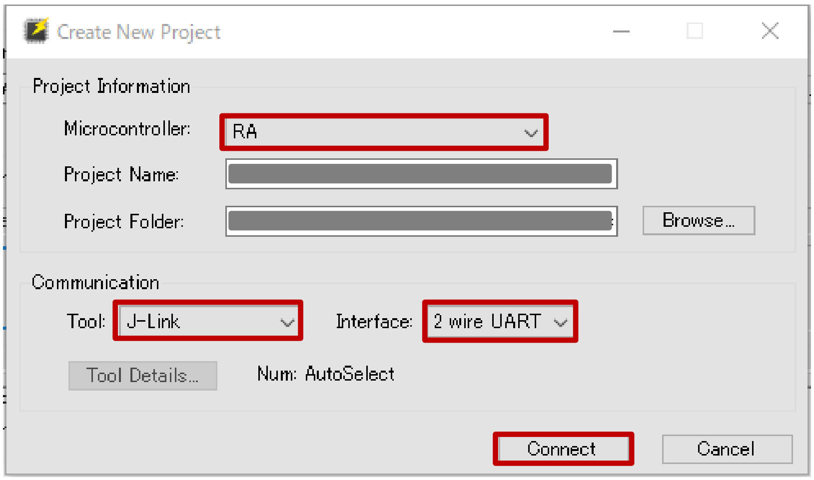


Figure 2-2 Setting of ‘Microcontroller’, ‘Tool’ and ‘Interface’

2. Set Communication Speed.

The communication speed is 9,600 bps. It can be set to 115,200 bps. If necessary, change it.

Here's how to set it up:

Click on the ‘Connections’ tab. Set ‘Speed’ to "115,200".

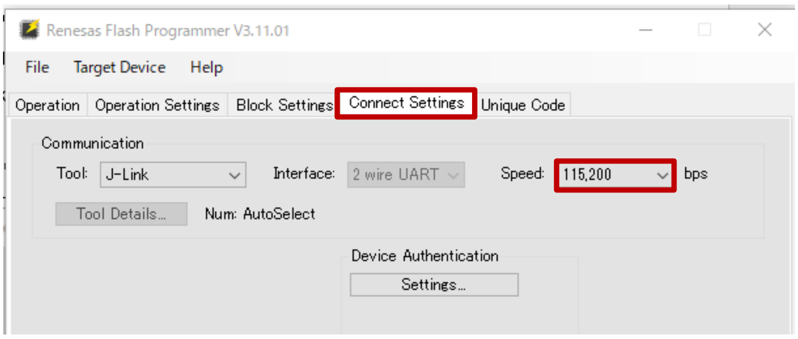


Figure 2-3 Setting of Communication Speed 115,200 bps

(b) If Renesas Flash Programmer Project for RA Family Has Been Created

1. Open the project.

Click ‘Open Project ...’ in ‘File’ and select the project file.

Click the ‘Connection Settings’ tab to check the communication speed.

If you want to set the communication speed to 115,200 bps, refer to ‘Figure 2-3 Setting of Communication Speed 115,200 bps’

(3) Writing to RA2A1

Write the FW according to the operating procedure of Renesas Flash Programmer.

2.2 Run Project

2.2.1 Connecting to PC

Debug USB I/F provides a power supply voltage for the EK-RA2A2. Refer to the 'EK-RA2A2 v1 User's Manua'.

The following shows the relationship between available USB-UART conversion adapters, connection methods, and corresponding e2 studio Project/HEX Files.

For details on the UART pin assignments, refer to 'Table 3-2 Pins used List'.

Table 2-1 Relationship between Available USB-UART Conversion Adapters, Connection Methods, and Corresponding e2 studio Project/HEX Files

Connection I/F	USB 2.0 Hi-Speed to UART Cable C232HD-EDHSP-0	PmodUSBUART	USB UART click (MIKROE-1203)	e2 studio Project, HEX File
Arduino Uno I/F [Note 1]	Refer to 'Arduino Uno I/F: Using Hi-Speed 12Mbps USB-UART Conversion Adapter'	Refer to 'Arduino Uno I/F: Using 3Mbps USB-UART Conversion Adapter'		• ek_ra2a2_uart0_32MHz project • ek_ra2a2_uart0_24MHz project • ek_ra2a2_uart0_32MHz_rev210.hex • ek_ra2a2_uart0_24MHz_rev210.hex • ek_ra2a2_uart0_12MHz_rev210.hex • ek_ra2a2_uart0_8MHz_rev210.hex
mikroBUS I/F [Note 1]			Refer to 'mikroBUS I/F: Using 3Mbps USB-UART Conversion Adapter'	• ek_ra2a2_uart0nohwflow_32MHz project • ek_ra2a2_uart0nohwflow_32MHz_rev210.hex
PMOD2 I/F		Refer to 'PMOD2 I/F: Using 3Mbps USB-UART Conversion Adapter'		• ek_ra2a2_uart3_32MHz project • ek_ra2a2_uart3_32MHz_rev210.hex • ek_ra2a2_uart3_8MHz_rev210.hex

Note 1: The SCI used is SCI0 for both. Therefore, connect a USB-UART conversion adapter to one of the connectors. Do not connect anything to the other connector.

A connection example is shown below.

(1) Arduino Uno I/F: Using Hi-Speed 12Mbps USB-UART Conversion Adapter

Connect the Arduino Uno connector to the USB 2.0 Hi-Speed to UART Cable C232HD-EDHSP-0.

The following shows how to connect the Arduino Uno pins and C232HD-EDHSP-0.

If you make a mistake in connection, the board and USB-UART conversion adapter may be damaged.

Table 2-2 How to Connect Arduino Uno Connector and USB-UART Conversion Adapter

EK-RA2A2		**USB-UART Conversion Adapter**	
Arduino Uno Digital (J24 Connector)	**Pin Assign**	**C232HD-EDHSP-0**	**PmodUSBUART**
SCL	P411/SCL0	--	--
SDA	P410/SDA0	--	--
AREF	+3.3V	-- [Note 1]	#6: VCC [Note 2]
GND	GND	GND / BLACK	#5: GND
D13/SCK0	P102/SCK0	CTS# / BROWN	#4: CTS#
D12/MISO	P101/RXD0	TXD / ORANGE	#3: TXD
D11/MOSI	P100/TXD0	RXD0 / YELLOW	#2: RXD
D10/SS	P103/CTS0 [Note 3]	RTS# / GREEN	#1: RTS#
D9	P105/GPIO	--	--
D8	P303/GPIO	--	--

Note 1: Do not connect the VCC of the USB-UART conversion adapter because there is a voltage difference between the board VDD and the USB-UART conversion adapter VCC. The same applies even if the VCC of the USB-UART conversion adapter is 3.3V.

Note 2: To supply voltage from EK-RA2A2 AREF to PmodUSBUART VCC, set the jumper on PmodUSBUART to "LCL-VCC".

Note 3: The P103 is also wired to the PMOD1 #10 GPIO. When using PMOD1 at the same time, consider the circuit configuration to prevent malfunctions due to signals from PMOD1 #10 pin.

Below is a connection example when using USB 2.0 Hi-Speed to UART Cable C232HD-EDHSP-0.

Connect to Digital ~9, ~10, ~11, 12, 13, and GND.

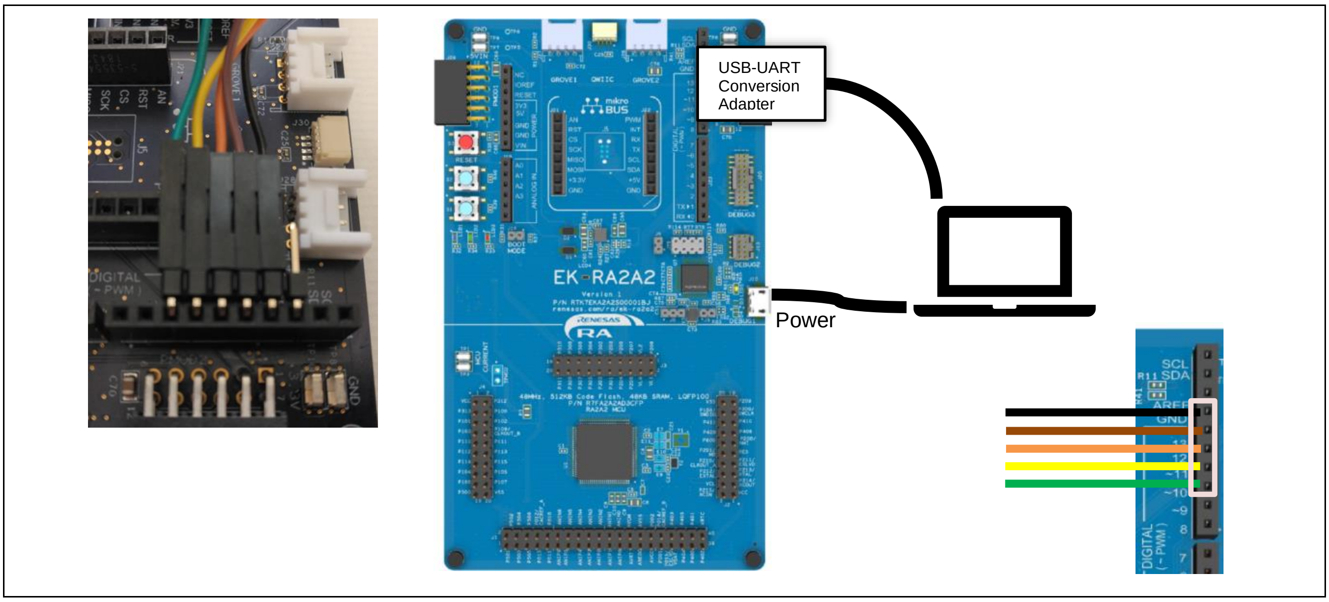


Figure 2-4 Example of Connecting USB 2.0 Hi-Speed to UART Cable C232HD-EDHSP-0 to Arduino Uno Connector

(2) Arduino Uno I/F: Using 3Mbps USB-UART Conversion Adapter

Due to the circuit configuration of the EK-RA2A2 Arduino Uno I/F, PmodUSBUART can be directly inserted.

For information on how to connect the Arduino Uno pinsl and PmodUSBUART, refer to 'Table 2-2 How to Connect Arduino Uno Connector and USB-UART Conversion Adapter'.

Also, in order to supply voltage from EK-RA2A2 VCC to PmodUSBUART VCC, set the jumper on PmodUSBUART to "LCL-VCC".

If you make a mistake in connection, the board and USB-UART conversion adapter may be damaged.

Below is a connection example when using PmodUSBUART.

Connect to Digital ~9, ~10, ~11, 12, 13, GND and AREF.

Since AREF is supplied with the board's 3.3V power, it can be used as VCC for PmodUSBUART.

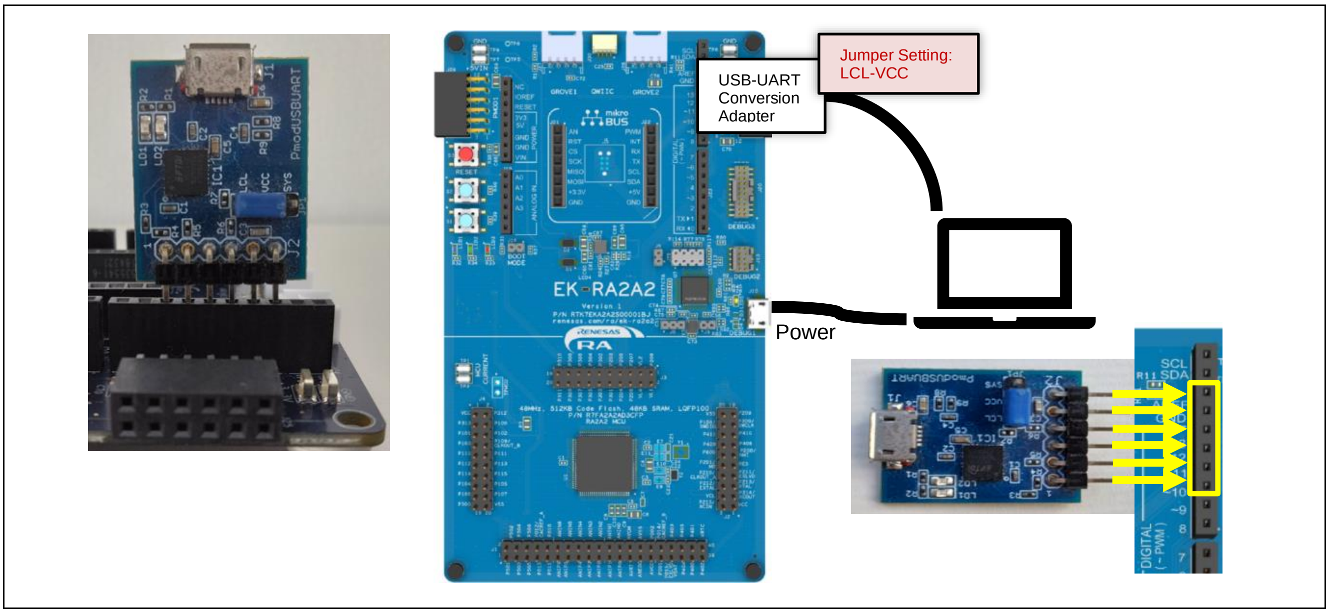


Figure 2-5 Example of Connecting PmodUSBUART to Arduino Uno Connector

(3) mikroBUS I/F: Using 3Mbps USB-UART Conversion Adapter

Due to the EK-RA2A2 circuit, RTS/CTS flow control is not possible. Therefore, FW performs 2-wire control using only TXD/RXD. For more information, refer to ‘Table 3-2 Pins used List’.

Below is a connection example when using USB UART click (MIKROE-1203).

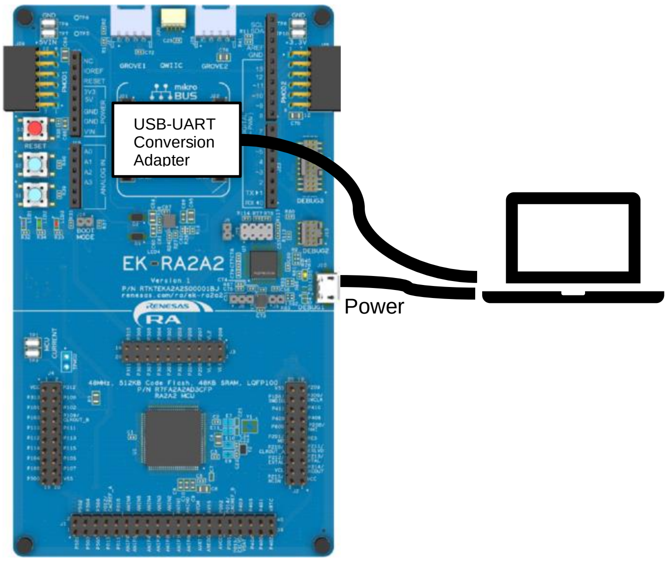


Figure 2-6 Example of Connecting USB UART click (MIKROE-1203) to mikroBUS Connecter

(4) PMOD2 I/F: Using 3Mbps USB-UART Conversion Adapter

The default circuit of the PMOD2 connector part of EK-RA2A2 is the circuit setting for Type-6A. Therefore, it is necessary to change to a Type-3A circuit. Please refer to the EK-RA2A2 user's manual (R20UT5333) for how to change.

Also, in order to supply voltage from EK-RA2A2 VCC to PmodUSBUART VCC, set the jumper on PmodUSBUART to "LCL-VCC".

Below is a connection example when using PmodUSBUART.

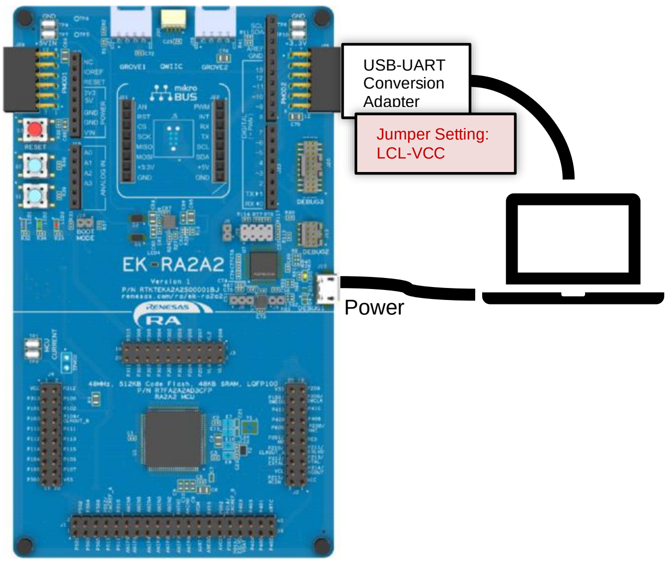


Figure 2-7 Example of Connecting PmodUSBUART to PMOD2 Connector

2.2.2 Launch QE for AFE

'QE for AFE' has a **plugin version** that works in the e2 studio environment and a **standalone version**.

Here, the procedure for measuring using the **standalone version** is shown.

For details on the operation of 'QE for AFE', refer to the 'QE for AFE' Help.

2.2.2.1 Preparation

Prepare the following in advance.

(1) Writing FW

Refer to '2.1 How to Write FW'.

(2) Connection between PC and EK-RA2A2

Refer to '2.2.1 Connecting to PC'.

2.2.2.2 Launching QE for AFE and Connecting to Target Board

Follow the steps below to connect the target board.

(1) Launching QE for AFE

(2) Importing Configuration [Note] File if Configuration file Has Been Prepared

If there is already a set file, the imported set value during the connection process with the target boat will be written.

(3) Connection with Target Board

Check the COM number of USB-UART Conversion Adapter in the PC device manager and select the COM number from 'COM Port:'.

Figure 2-8 Selecting COM Number

Also, check the following display on the console.

```
[Info]Connect succeeds.
```

Figure 2-9 Message on Console when Connection is Successful

If it has been imported, the setting value will be written subsequently. The following is displayed on the console.

```
[Info]Write $$$ register value to the target board successfully.
```

Figure 2-10 Message on Console when Connection is Successful ($$$: IP Name)

2.2.2.3 Message at Time of Completion of QE for AFE Continuous Measurement and Time to Output

QE for AFE has an upper limit on the number of acquired data.

Depending on the IP used, the message when the upper limit is reached will differ.

(1) For SDADC24

When setting continuous measurement, the number setting of channels is set and the sampling mode setting are independent, and data for 5 minutes can be acquired.

The message when the limit is reached is as follows:

```
[Error]The reserved buffer is full. Tuning is forced to stop.
```

Figure 2-11 Message for SDADC24

(2) For ADC12

The continuous measurement time depends on the setting of ADSSTRn.

The approximate time until the message for ADC12 is output during continuous measurement of ADC12 is shown below.

For the approximate time to output the message during ADC12 continuous measurements, refer to 'Table 1-6 ADC12 Continuous Measurement Operation Confirmed Setting Example'

The message when the limit is reached is as follows:

```
[Error]The reserved buffer is full. Tuning is forced to stop. If longer tuning time is required, please increase the average number or the over sampling ratio.
```

Figure 2-12 Message for ADC12

3. Program Description

3.1 Overview

This FW supports the Command/Response method communication.

The UART communication with 'QE for AFE' via SCI UART of EK-RA2A2. Then A/D measurement and Comparison measurement are executed according to the Command Request from 'QE for AFE'.

3.2 Peripherals to Use and Pin Settings

3.2.1 Peripherals to Use

The following shows the list of peripherals used in this FW, and the settings for each peripheral function are shown below.

Table 3-1 Peripheral Features List

Project	Intended Use
SDADC24	A/D measurement
ADC12	A/D measurement
SCIn	Communication: Used of UART SCI communication: When using, connect USB-UART conversion adapter. Refer to 'Table 1-11 Operating Confirmation Conditions'.
DTC	Used for SCI0 UART communication and data acquisition from the following registers • ADC12 A/D data registers y(ADDRy) • ADC12 A/D Temperature Sensor Data Register (ADTSDR) • ADC12 A/D Internal Reference Voltage Data Register (ADOCDR) • SDADC24 Sigma-Delta A/D Conversion Result Register Type 1(SDADCRn) • SDADC24 Sigma-Delta A/D Conversion Result Register (LPF) Type 1(SDADCRLPFn) • SDADC24 Sigma-Delta A/D Conversion Result Register Type 2(SDADCRmT2) • SDADC24 Sigma-Delta A/D Conversion Result Register (LPF) Type 2(SDADCRLPFmT2)

3.2.2 Pin Settings

3.2.2.1 Pin List

The following is a list of pins used in this FW.

The blue text indicates the functions set by this FW.

Table 3-2 Pins used List

No	Pin	Configuration Function	Content
1	P015	AN003/EXLVDBAT	Used as an analog pin
2	P403	GTIOC48/MISO0	
3	P404	MOSI0	
4	P405	RSPCK0	
5	P400	IRQ9/RTCIC0	
6	P401	IRQ10/RTCIC1	
7	P402	RTCOUT/IRQ11/RTCIC2	[Note 2]
8	VRTC	VRTC	
9	P215	VCIN	
10	P214	XCOUT	
11	VSS	VSS	
12	VCL	VCL	
13	P213	XTAL	
14	P212	EXTAL	
15	VCC	VCC	
16	P211	EXLVD	
17	P210	GTIOC5B/CLKOUT/ADTRG0/IRQ8	
18	RES	RES	
19	P201	MD	
20	P200	NMI	
21	P600	-	
22	P408	GTIOC9A/SSLA0	
23	P409	GTIOC9B/SSLA1	
24	P410	GTIOC6A/SDA00	
25	P411	GTIOC7A/SCL0	
26	P300	GTIOC6B/SWCLK/SWCLK	
27	P108	SWDIO/SWDIO/DTOC7B/RTCOUT	
28	P209	CAPH	
29	P208	CAPL	
30	VL1	-	
31	VL2	-	
32	VL4	-	
33	P207	VL3	
34	P206	COM0	
35	P205	COM1	
36	P204	COM2	
37	P203	COM3	
38	P301	SEG0/COM4	
39	P302	SEG1/COM5	
40	P303	SEG2/COM6	

41	P304	SEG3/COM7	
42	P305	SEG4	
43	P306	SEG5/IRQ0	
44	P307	SEG6/IRQ1	
45	P308	IRQ2/SEG7	
46	P309	IRQ3/SEG8	For LED1 control (Red LED)
47	P310	IRQ4/SEG9	
48	P311	IRQ5/SEG10	
49	P312	SEG11/IRQ6	
50	P313	SEG12/IRQ7	[Note 2]
51	P100	AGTOA0/AGTOB0/AGTO0/TXD0 /GTIOC8A/AGTEE0/SEG13	TXD0 pin for SCI0 UART [Note 1,2]
52	P101	AGTOA1/AGTWO0/AGTOB1/RXD0 /AGTO1/GTIV/GTIOC8B/AGTEE1/SEG14	RXD0 pin for SCI0 UART [Note 1,2]
53	P102	AGTOA2/AGTOB2/AGTO2/GTIOC6A/GTIW /ADTRG0/AGTWEE0/AGTEE2/SEG15/SCK0	RTS pin for SCI0 UART [Note 1,2]
54	P103	AGTOA3/GTOUUP/AGTOB3/CTS_RTS0/AGTO3 /GTIOC6B/SSLA3/AGTWIO0/SEG16/AGTEE3	CTS pin for SCI0 UART [Note 1,2]
55	P109	AGTOA4/AGTOB4/SEG17/AGTWOB0 /GTOULO/AGTO4/CLKOUT/TXD9/AGTEE4	
56	P110	AGTOA5/AGTOB5/RXD9/SEG18/GTOVUP /AGTWOA0/AGTO5/AGTEE5	
57	P111	GTOVLO/SEG19/AGTOB6/AGTOA6/AGTO6 /GTIO5A/AGTEE6/SCK9	
58	P112	AGTOB7/AGTOA7/CTS_RTS9/GTIOC5B /GTOWUP/AGT07/SSLA2/SEG20/AGTEE7	
59	P113	SEG21	
60	P114	SEG22	
61	P115	SEG23	
62	VCC	VCC	
63	VSS	VSS	
64	P104	SEG24/GTIOC8A/SCK2	
65	P105	CTS_RTS2/SEG25/GTIOC8B	
66	P106	TXD2/SEG26	
67	P107	RXD2/SSLA1/SEG27	
68	P500	RXD3/IRQ4/SEG28 /AGTIO0/GTOWLO/AGTWEE1	TXD3 pin for SCI3 UART [Note 3]
69	P501	GTETRGA/SEG29/AGTIO1/AGTWIO1 /TXD3/IRQ5	RXD3 pin for SCI3 UART [Note 3]
70	P502	AGTWO1/GTIOc9A/RSPCK0/GETERGB /SEG30/AGTIO2/IRQ6/SCK3	RTS pin for SCI3 UART [Note 3]
71	P503	GTIOC9B/CTS_RTS3/AGTWOA1/AGTIO3 /SSLA0/SEG31/IRQ7	CTS pin for SCI3 UART [Note 3]
72	P504	AGTWOB1/AGTIO4/SEG32/SCK1/MOSI0	
73	P505	MISO0/CTS_RTS1/AGTIO5/SEG33	
74	P506	AGTIO6/IRW0/TXD1/SEG34	
75	P013	IRW1/RXD1/AGTIO7/SEG35/SDA1	
76	P012	IRQ2/GTIOC5A/SEG36/SCL1/CACREF	
77	P011	GTIOC4A/SEG37	
78	P010	SEG38	
79	ANIP6	ANIP6	Used as an analog pin
80	ANIN6	ANIN6	Used as an analog pinl

81	ANIP5	ANIP5	Used as an analog pin
82	ANIN5	ANIN5	Used as an analog pinl
83	ANIP4	ANIP4	Used as an analog pin
84	ANIN4	ANIN4	Used as an analog pinl
85	ANIP3	ANIP3	Used as an analog pin
86	ANIN3	ANIN3	Used as an analog pinl
87	ANIP2	ANIP2	Used as an analog pin
88	ANIN2	ANIN2	Used as an analog pinl
89	ANIP1	ANIP1	Used as an analog pin
90	ANIN1	ANIN1	Used as an analog pinl
91	ANIP0	ANIP0	Used as an analog pin
92	ANIN0	ANIN0	Used as an analog pinl
93	AVRT	AVRT	
94	AVCM	AVCM	
95	AREGC	AREGC	
96	AVSS	AVSS	
97	AVCC	AVCC	
98	P002	AN002/VREFL0	Used as an analog pin
99	P001	AN001/VREFH0	Used as an analog pin
100	P014	AN000	Used as an analog pinl

Note 1: This is used for TXD/RXD 4-wire control with RTS/CTS flow using the Arduino Uno connector.
On the Arduino Uno connector, these four pins are placed side by side.
The projects that make use of these pins are shown in ‘Table 1-3 e2 studio Projects’.
ek_ra2a2_uart0_32MHz
ek_ra2a2_uart0_24MHz

Note 2: This is used for TXD/RXD 2-wire control using the mikroBUS connector.
The projects that make use of these pins are shown in ‘Table 1-3 e2 studio Projects’.
ek_ra2a2_uart0nohwflow_32MHz
Due to the circuit of the EK-RA2A2, hardware CTS flow control of the SCI UART is not possible, so no RTS/CTS flow control is realized by the following connection.
miicroBUS RST (P402)] (Connected to FTDI FT232RL CTS#): Set to GPIO Input (IRQ Disabled)
mikroBUS INT (P313) (Connected to FTDI FT232RL RTS#): Set to GPIO L Output
In the EK-RA2A2, P102 and P103 are set as follows.
P102: Default settings
P103 (CTS_RTS0 pin): Port L output.
The CTS_RTS0 pin (P103) for UART0 is connected to PMOD1 #10 pin.

Note 3: This is used for TXD/RXD 4-wire control with RTS/CTS flow using the PMOD2 connector.
The projects that make use of these pins are shown in ‘Table 1-3 e2 studio Projects’.
ek_ra2a2_uart3_32MHz

3.2.3 How LED Work

LED3 (Red LED) on the EK-RA2A2 lights up during the following operations.

— During A/D conversion operation of SDADC24 or ADC12

.

3.3 Communication Specifications

The communication specifications for ‘QE for AFE’ and FW are as follows.

3.3.1 UART Serial Communication Settings

The serial communication settings for UART SCI communication are as follows.

When measuring SDADC24, set the maximum transfer rate of the USB-UART conversion adapter.

Table 3-3 Serial Communication Settings

Items	Settings
Transfer Speed (Bitrate)	Default: 1M bps (Lower Limit) Maximum: 3M bps [Note 1] Maximum: 5.333333M bps [Note 5] / 4M bps [Note 4] The bitrate can be changed after the initial communication. The bitrate that has been confirmed to work are shown below. If you set the bitrate other than the following, normal communication may not be possible. — 5.333333M bps [Note 5] — 4M bps [Note 4] — 3M bps — 2M bps — 1.5M bps — 1.333333M bps [Note 3] — 1M bps
Data Length	8-bit
Parity	No parity
Stop Bit	1-bit
Hardware Flow Control [Note 2]	CTS pin: Set to CTS function enabled (RTS function disabled) RTS pin: Port control

Note 1: This value is when using a 3Mbps USB-UART conversion adapter.

Note 2: Refer to ‘Table 3-2 Pins used List’. Depending on the project, the pins are different.
RTS/CTS flow control is standard. Depending on the project, RTS/CTS flow control may not be available.

Note 3: This is the maximum value when PCLKB frequency = 8 MHz, and communication has been confirmed only when PCLKB frequency = 8 MHz.

Note 4: PCLKB frequency = 24MHz, which is the maximum value when using the Hi-Speed 12Mbps USB-UART conversion adapter.

Note 5: PCLKB frequency = 32MHz, which is the maximum value when using the Hi-Speed 12Mbps USB-UART conversion adapter.

However, in the following cases, change the bitrate setting of QE for AFE back to 1M bps.

— When the EK-RA2A2 is reset

3.4 Write and Build Using e2 studio Integrated Development Environment (IDE)

Import the project, build the project, and write to RA2A1 on the EK-RA2A2. For the included projects, refer to ‘Table 1-3 e2 studio Project’.

3.4.1 Import Procedure

The import procedure is shown in the figure below.

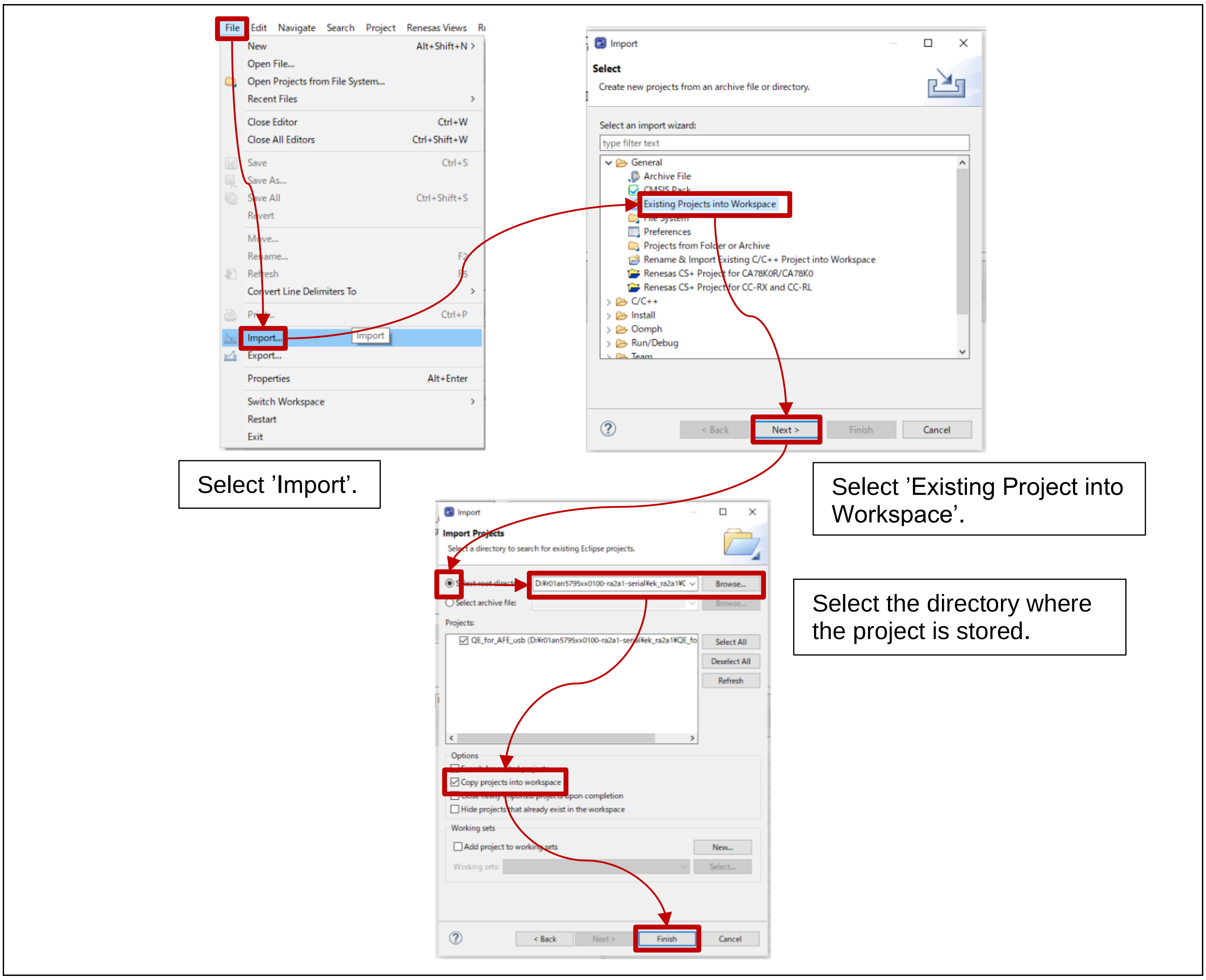


Figure 3-1 Steps to Import a Project into e2 studio

3.4.2 Launch Debug Mode

Execute the **‘Build’** and press . Then the **‘Debug’** mode will be executed automatically.

3.4.3 Notes on building E2 Studio projects

The build execution results in two warnings.

- p_data：
  This argument is used when setting the USB I/F on other MCUs that have USB I/F.
  It doesn’t be used in this project. Therefore, ignore it.
- avcc0_range：
  This argument is used when setting the low voltage on other MCU boards that can set the low voltage.
  It doesn’t be used in this project. Therefore, ignore it.

```
../src/hal_entry.c: In function 'send_packet':
../src/hal_entry.c:1104:39: warning: unused parameter 'p_data' [-Wunused-parameter]
 1104 | static fsp_err_t send_packet(int8_t * p_data, size_t len)  // 'p_data' is used for USB I/F
      |                              ~~~~~~~~~^~~~~~
../src/r_reg_write.c: In function 'QE4AFE_RegWrite':
../src/r_reg_write.c:1567:102: warning: unused parameter 'avcc0_range' [-Wunused-parameter]
 1567 | fsp_err_t QE4AFE_RegWrite(uint32_t * p_wr_adrs, uint32_t wr_data, uint8_t write_access_size, uint8_t avcc0_range)
      |                                                                                              ~~~~~~~~^~~~~~~~~~
```

Figure 3-2 Warning that Occurs when Building the Project

3.4.4 Stack Size

The stack size is set as follows.

Table 3-4 Stack Size Setting

Project	Stack Size
SCI UART communication	0x600

3.4.5 About e2 studio Project Source Changes

In FSP Configuration, only the clock setting is allowed to be changed.

Therefore, select the project on which you want to base your changes. For details on setting up a project, '1.1.1(2)e2 studio Projects'.

Follow the steps below to display the 'Clocks Configuration'.

[Project Explorer] -> [configuration.xml] file -> Click [Clocks] tab]

Figure 3-3 shows Clock setting changes.

I is possible to change the clock setting in the FSP environment shown in 'Table 1-11 Operating Confirmation Conditions'.

It is also possible to change the clock setting of SDADCCLK.

For details on clock settings, refer to User's Manual: Hardware.

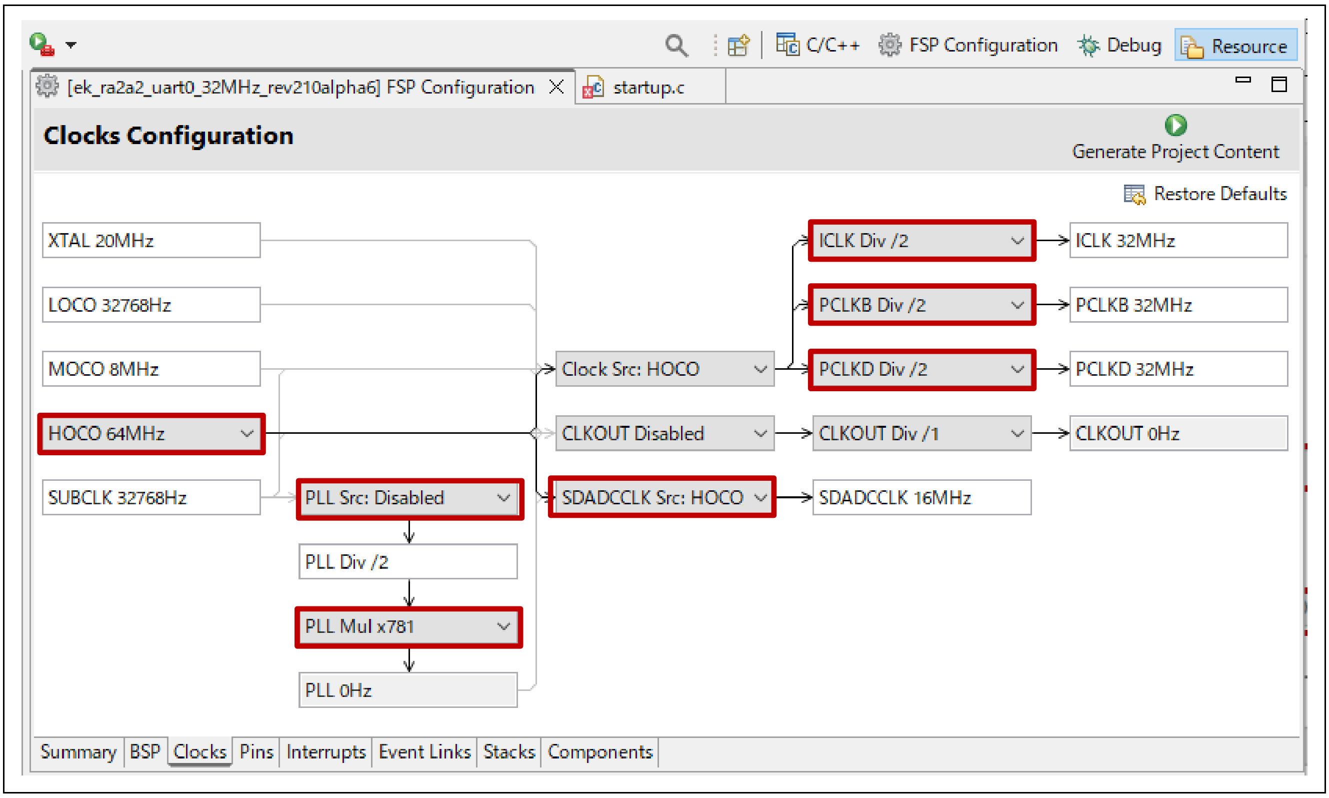


Figure 3-3 FSP Configuration Clock Setting Screen when using UART I/F

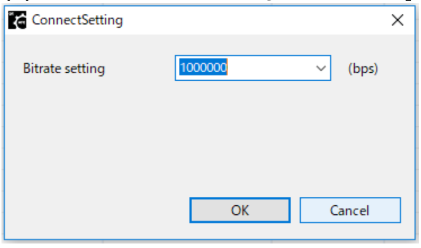
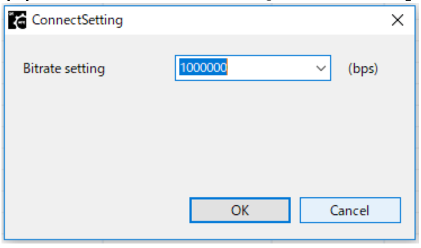
For the clocks and their frequencies that are allowed to change, Renesas recommend setting of the clocks and their frequencies shown in 'Table 1-12 Clock Settings and Tools to Use (SCI UART)'.

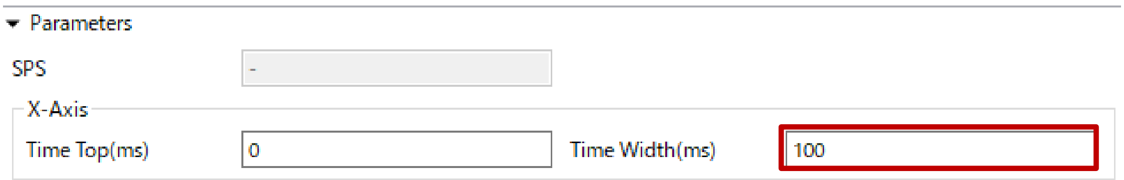
During continuous measurement operation, if the clock frequency is lowered, it may not be possible to measure normally. Therefore, please evaluate it sufficiently.

For clock settings during continuous measurement operation, refer to 'Table 1-6 ADC12 Continuous Measurement Operation Confirmed Setting Example' and 'Table 1-9 SDADC24: Sampling Mode, Number of Channels and Supported Bitrate when Using UART'.

4. Trouble-solving Method

The trouble cases are shown below. Please refer to it when evaluating.

1	• **RFP: Unable to connect to board after writing with Renesas Flash Programmer.**
	You must restore the jumper settings that you set for writing to their original state. Check the jumper settings.
2	• **GUI: When using UART I/F: Cannot connect to the board.**
	● There may be a bit rate mismatch. Follow the steps below. (1) Reset the board. (2) Set the Bitrate of ‘QE for AFE’ [ConnectSetting] to 1M bps.  (3) Connect to the board using ‘QE for AFE’. Note: Since it is necessary to change the bit rate during continuous measurement of ADC16/12, it is necessary to set the bitrate of ‘QE for AFE’ to 1M bps if the board reset is executed.
	● You may set the bitrate that does not work and performed board connection processing. Follow the steps below. (1) Reset the board. (2) Set the Bitrate of ‘QE for AFE’ [ConnectSetting] to 1M bps.  (3) Connect to the board using ‘QE for AFE’. (4) Set the bitrate to the different value and try changing the bitrate. Note: If the bit rate is set to a large value, communication may not be possible.

3	• **GUI: QE for AFE drawing stops during Continuous measurement.**
	The following are the causes of drawing stoppage. ● The measurement was automatically stopped because the upper limit of the number of data acquired by QE for AFE was exceeded. For measurable time, refer to '2.2.2.3 Message at Time of Completion of QE for AFE Continuous Measurement and Time to Output'. ● Since the FTDI driver is old, data was missed on the PC and drawing stopped. Please install the version shown in '1.4 Operation Confirmation Environment' or later on your PC. ● During continuous measurement, FW became unable to transmit data or QE for AFE stopped drawing due to variations in data acquisition timing on the PC USB side. As a result of signal analysis with PC USB, it has been confirmed that data communication may rarely be performed for a period longer than the data transmission interval. In order to avoid the period when the PC is not communicating, we recommend that you do not run other PC applications as much as possible and evaluate in the offline state. Still, the data missing may occur. Refer to '1.2.1(1)(a) ADC12 Continuous Measurement'. Also, if drawing is stopped due to this cause, the LED may be lit. In that case, stop tuning QE for AFE and check that the LED is off. If the LED is lit, reset the board and try connecting again.
4	• **GUI: QE for AFE drawing stops during One-shot measurement.**
	The following are the causes of drawing stoppage. ● The measurement was automatically stopped because the upper limit of the number of data acquired by QE for AFE was exceeded. For measurable time, refer to '1.2.1(1)(b) ADC12 One-Shot Measurement'.
5	• **GUI: Graph of QE for AFE is not drawn when measuring ADC16/12 or SDADC24.**
	● It is possible that the **Time Width** setting is not appropriate. Check the following. (1) If the **Time Width(ms)** is larger than the initial value of 100ms, set it to "**100ms**". 
	● In the case of SDADC24, it is possible that the Bitrate is a small value. Make sure to set the Bitrate to a large value.
6	• GUI: Clicking 'Stop AFE monitoring' does not stop it.

	● This may occur if the FW cannot accept the STOP command during continuous measurement at a high data rate. In that case, the following message will be displayed. Please follow the message.
	`[Error]STOP command failed to execute. Please try to press [Stop AFE monitoring] button again.`
	`If tuning still can not stop, please reset the board and re-connect.`

7	• GUI: Re-Clicking 'Start AFE monitoring' does not start it.
	● Even if you run it again, you may see the same message below. In this case, "Disconnect" → "Connect" again. If the reconnection is successful, the Bitrate setting remains the same as the changed value. `[Error]RUN command failed to execute. Please try again.`
8	• GUI: Writes by register settings are not successful.
	● When writing the register settings, a normal response may not be displayed. Communication did not end normally and has not been written. To restore communication, change the settings and write the values. After that, you may be able to write the value you want to set again. Also, "Disconnect" → "Connect" may restore communication.

Revision History

Rev.	Date	Description	
		Page	Summary
2.10	Mar.14.24	-	First Release

Pmod™ is the trademark of Digilent Inc.

Arduino® is the registered trademark of Arduino SA.

mikroBUS™ is the trademarks of MikroElektronika.

Other brands and names mentioned in this document may be the trademarks or registered trademarks of their respective holders.

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.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

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/.