

RA Family, RX Family, RL78 Family

RRH46410 Sample Software Manual

Introduction

This Application Note describes sample software for RRH46410 gas sensor that operates on RA Family, RX Family and RL78 Family.

Target Device

RA2E1 Group

RA0E1 Group

RX140 Group

RL78/G23 Group

Target Sensor Board

Digital Gas Sensor Module for Indoor Air Quality Applications (QCIOT-RRH46410POCZ)

The setting example described in this application note is an example when using the sensor board mentioned above.

Therefore, you will need to review the following settings according to the target circuit.

- Interrupt Signal Circuit: Refer to "6.5 Notes for Interrupt Signal Circuits".
- RESET Signal Circuit: Refer to "6.6 Notes for RESET Signal Circuits".

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.



Contents

1. 1.1	Overview Terms/Abbreviations	
2.	Environment for Confirming Operation	5
2.1	Environment for Confirming Operation on RA Family MCU	
2.2	Environment for Confirming Operation on RX Family MCU	
2.3	Environment for Confirming Operation on RL78/G23 Group MCU	
3.	RRH46410 Sensor Specifications	10
3.1	Sensor Specifications Overview	10
3.2	Sensor Function and Methods	10
3.2.1	1 Conversion of Output Data – Firmware / API / Algorithms	10
4.	Specification of Sample Software	11
4.1	Sample Software Structure	11
4.2	Specification of Sensor Control Module API Functions	11
4.2.1	1 List of Sensor Control Module API Functions	11
4.2.2	2 API Usage Guide	12
4.3	Flowchart of Main Processing in Non-OS Version of Sample Software	
4.4	Flowchart of OS Version of Sample Software	16
5.	Configuration Settings	18
5.1	RRH46410 Sensor Control Module Settings	
5.1.1	1 RA Family	
5.1.2	2 RX Family	19
5.1.3	3 RL78 Family	20
5.2	I2C Communication Middleware (COMMS_I2C) Settings	21
5.2.1	1 RA Family	21
5.2.2	2 RX Family	22
5.2.3	3 RL78 Family	23
5.3	I2C Driver Settings	
5.3.1	1 RA Family	
5.3.2	2 RX Family	27
5.3.3	3 RL78 Family	30
5.4	IRQ Driver Settings	31
5.4.1	1 RA Family	31
5.4.2	2 RX Family	32
5.4.3	3 RL78 Family	32
6.	Guide for Changing Target Device	33
6.1	Importing Sample Project	33
6.2	RA Sample Project	35



RA Family, RX Family, RL78 Family

6.2.1	Modifying Settings of FSP Configurator	35
6.2.2	Changing Sample Code	45
6.2.3	Changing when not using IRQ	45
6.2.4	Changing Toolchain Setting	45
6.3	RX Sample Project	46
6.3.1	Modifying Settings of Smart Configurator	46
6.3.2	Changing Sample Code	55
6.3.3	Changing Toolchain Setting	55
6.3.4	Notes for Build on GCC	55
6.3.5	When using IAR Integrated Development Environment "IAR Embedded Workbench"	55
6.4	RL78 Sample Project	56
6.4.1	Modifying Settings of Smart Configurator	56
6.4.2	Modifying Generated Code	66
6.4.3	Changing Sample Code	67
6.4.4	Changing Toolchain Setting	68
6.4.5	Notes for Build on LLVM	70
6.5	Notes for Interrupt Signal Circuits	71
6.6	Notes for RESET Signal Circuits	72
6.7	Pull-up Resister Circuit Configuration when Daisy Chain Connections of Renesas Sensor Pmod	
	Boards	72
7. `	Viewing Gas Data	73
Revis	sion History	75

Trademarks

FreeRTOS[™] is a trademark of Amazon Web Services, Inc.

 $\mathsf{Pmod}^{\mathsf{TM}}$ is a trademark of Digilent Inc.



1. Overview

This sample software acquires data from the RRH46410 gas sensor and calculates the result values. In combination with the I2C driver of the FSP/FIT or the Code Generator, the sample software controls the RRH46410 through the I2C in the MCU to measure gas environment, acquires Measurement data, converts and calculates the acquired results.

1.1 Terms/Abbreviations

The terms and their abbreviations are listed below.

Terms	Abbreviation			
RRH46410 Sensor Control	Sensor Control Module			
Module	When MCU is RA Family, "rm_rrh46410"			
	When MCU is RX Family, "r_rrh46410_rx"			
	When MCU is RL78 Family, "r_rrh46410"			
I2C Communication	COMMS_I2C			
Middleware	When MCU is RA Family, "rm_comms_i2c"			
	When MCU is RX Family, "r_comms_i2c_rx"			
	When MCU is RL78 Family, "r_comms_i2c"			
I2C Driver	When MCU is RA Family, "r_iic_master", "r_sci_i2c", "r_iica_master"			
	When MCU is RX Family, "r_riic_rx", "r_sci_iic_rx"			
	When MCU is RL78 Family, "r_iica_master"			
IRQ Driver	When MCU is RA Family, "r_icu"			
	When MCU is RX Family, "r_irq_rx"			
	When MCU is RL78 Family, "Interrupt Controller"			
Serial Communications	When MCU is RA Family, "SCI", "SCI I/F"			
Interface	When MCU is RX Family, "SCI", "SCI I/F"			
Serial Array Unit	When MCU is RA Family, "SAU", "SAU I/F"			
	When MCU is RL78 Family, "SAU", "SAU I/F"			
I2C Bus Interface	When MCU is RA Family, "IIC", "IIC I/F"			
	When MCU is RX Family, "RIIC", "RIIC I/F"			
I2C Bus Interface (IICA)	When MCU is RA Family, "IICA", "IICA I/F"			
Serial Interface IICA	When MCU is RL78 Family, "IICA", "IICA I/F"			
General Term for	"I2C I/F"			
I2C Bus Interface,				
I2C Bus Interface (IICA),				
Serial Interface (IICA)				
General Term for	"ICU I/F" (Interrupt Controller Unit)			
Interrupt Controller				
General purpose I/O Port	"GPIO", "GPIO I/F"			
Pin No.1 (#1) of Renesas	"IRQ#" (L output when an interrupt occurs)			
Pmod Type 6A Sensor				
Board				



2. Environment for Confirming Operation

2.1 Environment for Confirming Operation on RA Family MCU

The operation of this software has been confirmed on RA family MCU in the following environment.

(1) Evaluation Kit for RA2E1 (EK-RA2E1)

Table 2-1 Confirming Operating Environment for EK-RA2E1

Item	Description
Demonstration board	RTK7EKA2E1S00001BE (EK-RA2E1)
Microcontroller	RA2E1 (R7FA2E1A92DFM:64pin)
Operating frequency	48MHz
Operating voltage	5V
Integrated development	Renesas Electronics e ² studio 2024-07
environment	
C compiler GNU ARM Embedded 13.2.1.arm-13-7	
Configuration options	Add the following settings to the compiler default settings:
	ISO C99 (-std=c99), Optimization Level: Default settings (-O2)
FSP	v5.4.0
RTOS	FreeRTOS v10.6.1
Emulator	On board (J-LINK)
Interposer	Interposer Board for Pmod Type2/3 to 6A (US082-INTERPEVZ)
Sensor board	Digital Gas Sensor Module for Indoor Air Quality Applications (QCIOT-
	RRH4N10POCZ)

Table 2-2 Memory Size Used in RA2E1 (When an Operation Mode with Maximum ROM Size Is Set)

Operation Mode	Area	Size (Non-OS) [Bytes]	Size (Free RTOS) [Bytes]		
IAQ 2nd Gen	ROM	2,372	4,240 (Note 1)		
	RAM	288	500		

Note Memory size is calculated for sample code, RRH46410 sensor control module, and COMMS_I2C. In RTOS, memory size does not include memory size of the thread.

Note 1 This includes an increase of 1,554 bytes due to the Relax function.

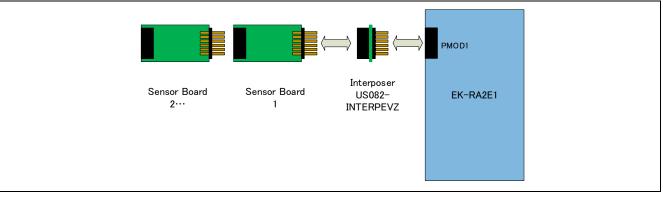


Figure 2-1 Hardware Connections for EK-RA2E1



(2) RA0E1 Fast Prototyping Board (FPB-RA0E1)

Table 2-3 Confirming Operating Environment for FPB-RA0E1

Item	Description		
Demonstration board	RTK7FPA0E1S00001BJ (FPB-RA0E1)		
Microcontroller	RA0E1 (R7FA0E1073CFJ:32pin)		
Operating frequency	32MHz		
Operating voltage	5V		
Integrated development	Renesas Electronics e ² studio 2024-07		
environment			
C compiler	GNU ARM Embedded 13.2.1.arm-13-7		
Configuration options	Add the following settings to the compiler default settings:		
	ISO C99 (-std=c99), Optimization Level: Default settings (-Oz)		
FSP	v5.4.0		
Emulator	On board (J-LINK)		
Interposer Interposer Board for Pmod Type2/3 to 6A (US082-INTERPEVZ)			
Sensor board	Digital Gas Sensor Module for Indoor Air Quality Applications (QCIOT-		
	RRH46410POCZ)		

Table 2-4 Memory Size Used in RA0E1 (When an Operation Mode with Maximum ROM Size Is Set)

Operation Mode	Area	Size (Non-OS) [Bytes]
IAQ 2nd Gen	ROM	2,066
	RAM	276

Note Memory size is calculated for sample code, RRH46410 sensor control module, and COMMS_I2C.

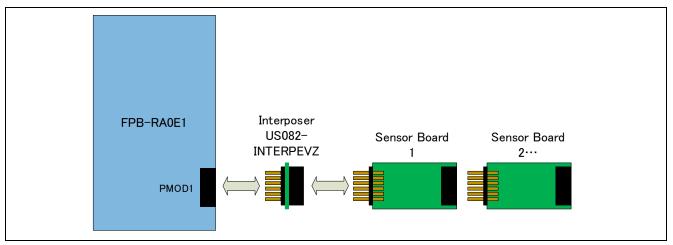


Figure 2-2 Hardware Connections for FPB-RA0E1



(3) Use of Interposer Board

The Interposer Board is an I/F conversion board for connecting Pmod Type 6A sensors by switching the Pmod Type 2A/Type 3A connector of the SCI I/F to the Simple IIC function.

Therefore, it cannot be used with the Pmod Type 2A/Type 3A connector of the SAU I/F. However, it may be usable by switching to the IICA I/F. Refer to the MCU hardware manual.

Table 2-5 Operational Feasibility Depending on Pmod I/F, Serial I/F, and Presence or Absence of Interposer Board

Pmod I/F	Destination MCU Serial I/F	Operational Feasibility
Type 2A, Type 3A	SCI I/F, IICA I/F (Note 1)	It works when using an Interposer Board. (Note 2)
	SAU I/F	It does not work regardless of whether the Interposer Board is present or not.
Туре 6, Туре 6А	SCI I/F, IIC I/F, SAU I/F, IICA I/F	It works without an Interposer Board. (Note 2)

Note 1: These pins are provided for SAU I/F but can be used when it is switchable to IICA pins by multifunction pins assignment. The signal connections when switchable are shown below.

Pmod Pin	Туре 2А /Туре 3А	Destination SAU I/F ICU I/F GPIO I/F	Switching to Multi- Function IICA I/F		Interposer Board		Renesas Pmod Type 6A Sensor Board
#1	CS/CTS	GPIO		\leftrightarrow	× /	\leftrightarrow	IRQ# (Note 3)
#2	MOSI/TXD	SAU TXD	SDAA	\leftrightarrow		\leftrightarrow	RESET#
#3	MISO/RXD	SAU RXD	SCLA	\leftrightarrow		\leftrightarrow	IIC_SCL
#4	SCK/RTS	GPIO		\leftrightarrow		\leftrightarrow	IIC_SDA
#7	INT	IRQ#		\leftrightarrow		\leftrightarrow	BUSY#
#8	RESET	GPIO		\leftrightarrow	\leftarrow	\leftrightarrow	ENABLE
#9	CS2/GPIO	GPIO		\leftrightarrow	\bullet	\leftrightarrow	POWER_ON
#10	CS3/GPIO	GPIO		\leftrightarrow	\bullet	\leftrightarrow	GPIO

Note 2: If an IRQ signal is used, make sure that the IRQ signal on MCU is connected to Pmod #1. Note 3: For an interrupt signal circuit, refer to "6.5 Notes for Interrupt Signal Circuits".

Application example: FPB-RA0E1 Pmod1 is applicable.

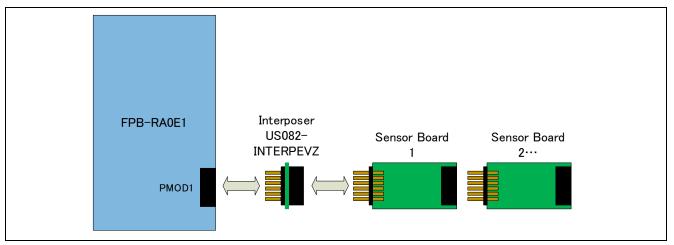


Figure 2-3 Hardware Connections for using IICA at Pmod Type 2A, Type 3A on FPB-RA0E1



2.2 Environment for Confirming Operation on RX Family MCU

The operation of this software has been confirmed on RX family MCU in the following environment.

(1) RX140 Fast Prototyping Board (FPB-RX140)

Table 2-6 Confirming Operating Environment for FPB-RX140

Item	Description			
Demonstration board	RTK5FP1400S00001BE (FPB-RX140)			
Microcontroller	RX140 (R5F51406BGFN: 80pin)			
Operating frequency	48MHz			
Operating voltage	5V			
Integrated development environment	Renesas Electronics e ² studio 2024-10			
C compiler	Renesas Electronics CC-RX V.3.06.00			
	GCC for Renesas RX 8.3.0.202405			
Configuration options	Add the following settings to the compiler default settings:			
	CC-RX: C99 (-lang = c99), Optimization Level: Default settings (Level 2)			
	GCC: ISO C99 (-std = c99), Optimization Level: Default settings (-Og)			
FIT	Board Support Packages (r_bsp) v7.51			
(RX Driver Package GPIO Driver (r_gpio_rx) v5.10				
v1.45)	RRH46410 Sensor Control Module (r_rrh46410_rx) v1.00			
	IIC Communication Middleware (r_comms_i2c_rx) v1.22			
	RIIC Multi Master I2C Driver (r_riic_rx) v3.00			
	Simple IIC Driver (r_sci_iic_rx) v2.80			
	IRQ Driver (r_irq_rx) v4.50			
	CMT Driver (r_cmt_rx) v5.70			
RTOS	FreeRTOS Kernal 10.4.3-rx-1.0.9 FreeRTOS Object 10.4.3-rx-1.0.9			
Emulator	On board (E2OB)			
Interposer	Interposer Board for Pmod Type2/3 to 6A (US082-INTERPEVZ)			
Sensor board	Digital Gas Sensor Module for Indoor Air Quality Applications (QCIOT- RRH46410POCZ)			

Table 2-7 Memory Size Used in RX140 (When an Operation Mode with Maximum ROM Size Is Set)

Operation Mode	Area	Size (Non-OS) [Bytes] (CC-RX)	Size (Free RTOS) [Bytes] (CC-RX)		
IAQ 2nd Gen	ROM	2,855	3,012		
	RAM	313	305		

Note Memory size is calculated for sample code, RRH46410 sensor control module, and COMMS_I2C. In RTOS, memory size does not include memory size of the thread.

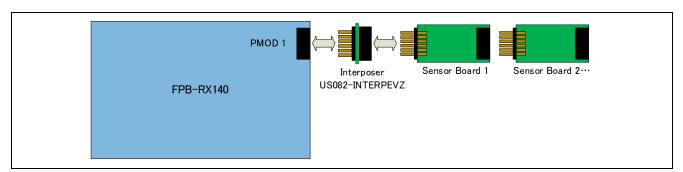


Figure 2-4 Hardware Connections for FPB-RX140

(2) Use of Interposer Board

If you add an Interposer Board to the Pmod Type 2A/Type 3A connector to which the SCI I/F is connected, you can use the Pmod Type 6A Sensor Pmod Board.



2.3 Environment for Confirming Operation on RL78/G23 Group MCU

The operation of this software has been confirmed on RL78/G23 group MCU in the following environment.

(1) RL78/G23-128p Fast Prototyping Board (RL78/G23-128p FPB)

Table 2-8 Confirming Operating Environment for RL78/G23-128p FPB

Item	Description	
Demonstration board	RTK7RLG230CSN000BJ (RL78/G23-128p FPB)	
Microcontroller	RL78/G23 (R7F100GSN2DFB :128pin)	
Operating frequency	32MHz	
Operating voltage	3.3V	
Integrated development environment	Renesas Electronics e2 studio 2024-07	
C compiler	Renesas Electronics CC-RL V1.14.00	
	LLVM for RL78 17.0.1.202409	
Configuration options	Add the following settings to the compiler default settings.	
	CC-RL: C99 (-lang = c99), Optimization Level: Default settings (-Odefault)	
	LLVM: GNU ISO C99 (-std = gnu99), Optimization Level: Default settings (-Og)	
SIS / CG	Board Support Packages (r_bsp) v1.70	
	Ports v1.5.0	
	RRH46410 Sensor Middleware (r_rrh46410) v1.00	
	IIC Communication Driver Interface Middleware (r_comms_i2c) v1.11	
	IIC Communication (Master mode) v1.6.0	
	Interrupt Controller v1.5.0	
	Interval Timer v1.5.0	
Emulator	On board (COM Port)	
Sensor board	Digital Gas Sensor Module for Indoor Air Quality Applications (QCIOT- RRH46410POCZ)	

Table 2-9 Memory Size Used in RL78/G23 (When an Operation Mode with Maximum ROM Size Is Set)

Operation Mode	Area	Size (Non-OS) [Bytes] (CC-RL)
IAQ 2nd Gen	ROM	2,986
	RAM	240

Note Memory size is calculated for sample code, RRH46410 sensor control module, and COMMS_I2C.

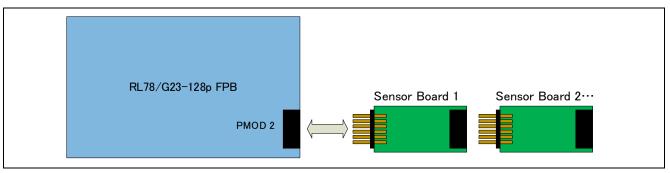


Figure 2-5 Hardware Connections for RL78/G23-128 FPB

(2) Use of Interposer Board

The Interposer Board is an I/F conversion board for connecting a Pmod Type 6A sensor by switching the Pmod Type 2A/Type 3A connector of an SCI I/F for RA/RX to the Simple IIC function.

Therefore, even if the Interposer Board is added to a Pmod Type 2A/Type 3A connector to which an SAU I/F is connected, the Pmod Type 6A Sensor Pmod Board cannot be used.



3. RRH46410 Sensor Specifications

3.1 Sensor Specifications Overview

The RRH46410 Gas Sensor Module is designed to detect typical TVOC contaminations based on studies and international standards for indoor air quality. Please refer to the <u>RRH46410</u> datasheet for more information about the sensor module, including parameters that describe the module's characteristics.

3.2 Sensor Function and Methods

The RRH46410 architecture leverages different "Methods of Operation" which use time, temperature, and signatures from gases that enable unique signals from a highly trained machine learning system and makes use of embedded artificial intelligence (AI) technology. This section discusses the different operation modes of the RRH46410. At present, the following operation modes are released.

Family of IAQ software releases:

Operation Mode 1:	IAQ 2nd Generation	; IAQ, Relative IAQ, TVOC and eCO ₂
Operation Mode 2:	IAQ 2nd Generation ULP	; IAQ, Relative IAQ, TVOC and eCO ₂
Operation Mode 3:	PBAQ	; TVOC measurement to meet PBAQ standards

By default, the IAQ 2nd Generation (Mode 1) operation is recommended for new designs.

Additional technical information on sensitivity, selectivity, and stability for all operation modes is available in Renesas' ZMOD4410 Application Note – TVOC Sensing. For more information, including application notes, white papers, blog, and manuals, visit the <u>RRH46410</u> webpage.

3.2.1 Conversion of Output Data – Firmware / API / Algorithms

To operate the RRH46410, a firmware provided by Renesas containing an API and an example should be used. For implementing the sensor module in a customer-specific application, detailed information on the programming is available. For downloading these documents, please visit the <u>RRH46410</u> webpage.



4. Specification of Sample Software

This sample software package contains a total of 8 projects:

- RA2E1 group: Non-OS project and OS (FreeRTOS) project
- RA0E1 group: Non-OS project
- RX140 group: Non-OS (CC-RX / GCC) projects and OS (FreeRTOS) project
- RL78/G23 group: Non-OS (CC-RL / LLVM) projects.

For the FreeRTOS settings for RX family, refer to the FAQ.

4.1 Sample Software Structure

The following shows the layer diagram of the sample software.

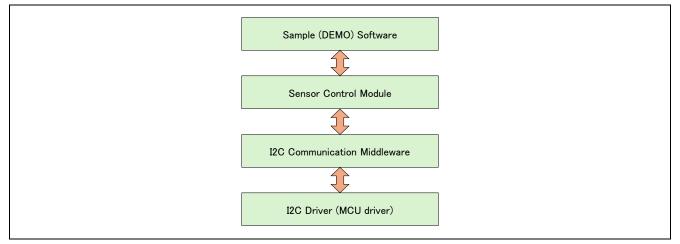


Figure 4-1 Layer diagram of Sample Software

4.2 Specification of Sensor Control Module API Functions

4.2.1 List of Sensor Control Module API Functions

The Sensor Control Module API includes the following functions.

For details on the Function API, see below.

RA Flexible Software Package Documentation Renesas Sensor Control Modules Firmware Integration Technology (R01AN5892) Renesas Sensor Control Modules Software Integration System (R01AN6192)

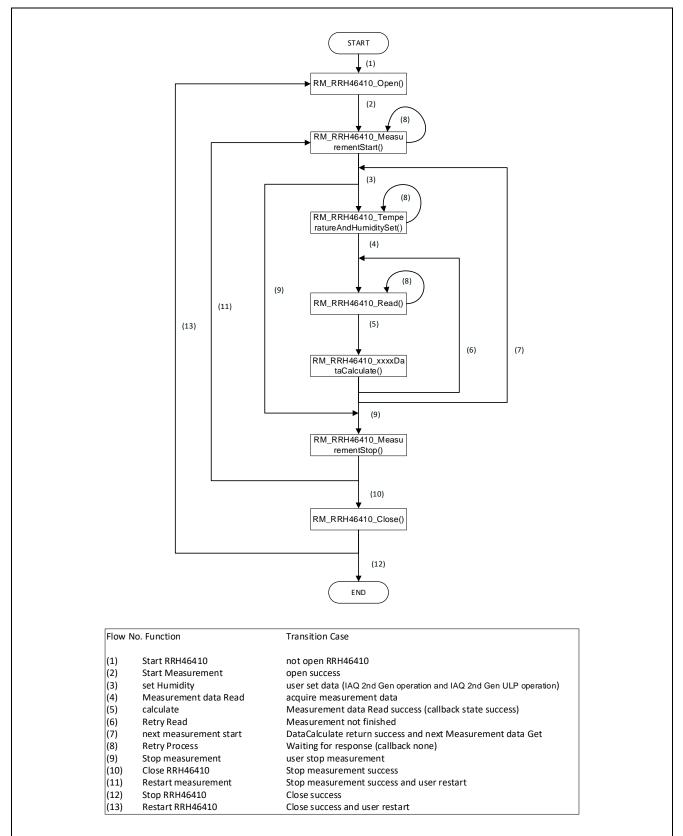
Table 4-1 List of	Sensor (Control Module	API Functions
-------------------	----------	-----------------------	----------------------

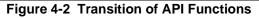
Function	Feature
RM_RRH46410_Open()	Opens a sensor.
RM_RRH46410_Close()	Closes a sensor.
RM_RRH46410_MeasurementStart()	Starts measurement.
RM_RRH46410_MeasurementStop()	Stops measurement.
RM_RRH46410_Read()	Reads measurement data.
RM_RRH46410_TemperatureAndHumiditySet()	Sets humidity. Temperature is not supported.
	(IAQ_2nd_Gen operation and IAQ_2nd_Gen_ULP operation)
RM_RRH46410_laq2ndGenDataCalculate()	Calculates IAQ 2nd Gen. values from measurement data.
RM_RRH46410_PbaqDataCalculate()	Calculates PBAQ values from measurement data.



4.2.2 API Usage Guide

Figure 4-2 Transition of API Functions shows the transition diagram of functions calling order as the usage condition of API functions.







RA Family, RX Family, RL78 Family

The calling conditions for each function are as follows:

- RM_RRH46410_Open():
- RM_RRH46410_Close():

RM_RRH46410_Read():

• RM_RRH46410_MeasurementStart():

RM_RRH46410_MeasurementStop():

RM_RRH46410_xxxxDataCalculate():

RM_RRH46410_TemperatureAndHumiditySet():

- (1) When starting RRH46410
- (13) When re-stating after RM_RRH46410_Close()
- (10) After RM_RRH46410_MeasurementStop()
- (2) When starting measurement after RM RRH46410 Open()
- (11) When re-stating measurement after RM RRH46410 MeasurementStop()
- (8) When retrying by waiting for measurement start response
- (9) When stopping measurement
- (4) When acquiring measurement data
- (6) When retrying due to measurement not finished
- (8) When retrying due to waiting for data acquisition response
- (3) When setting humidity (IAQ 2nd Gen and IAQ 2nd Gen ULP, Improved calculation accuracy)
- (8) When retrying due to waiting for humidity set response
- (5) When calculating data after RM_RRH46410_Read()"xxxx" is the name of each sensor function (laq2ndGen, Pbaq).

Note:

Since RM_RRH46410_Open() checks the state of the I2C driver, the I2C driver must be opened before the RM_RRH46410_Open() processing.

Regarding how to open the I2C driver of RA family and RX family, refer to the g_comms_i2c_bus0_quick_setup() function in the sample software. For RL78 family, this is not necessary because the I2C driver will be opened in the startup processing.

When using this API functions in an RTOS system, bus controlling by using semaphore by user is required if controlling the sensors at the same time in multiple threads/tasks.



4.3 Flowchart of Main Processing in Non-OS Version of Sample Software

This sample software first starts the driver and then repeats the processing for acquiring data from the sensor and calculating values from the results of measurement.

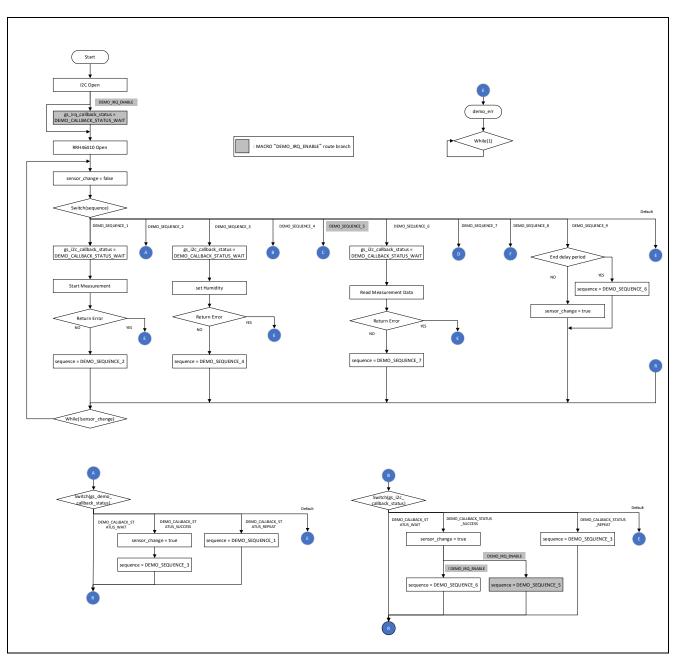


Figure 4-3 Flowchart of Main Processing in Non-OS Version of RRH46410 Sample Software (1)



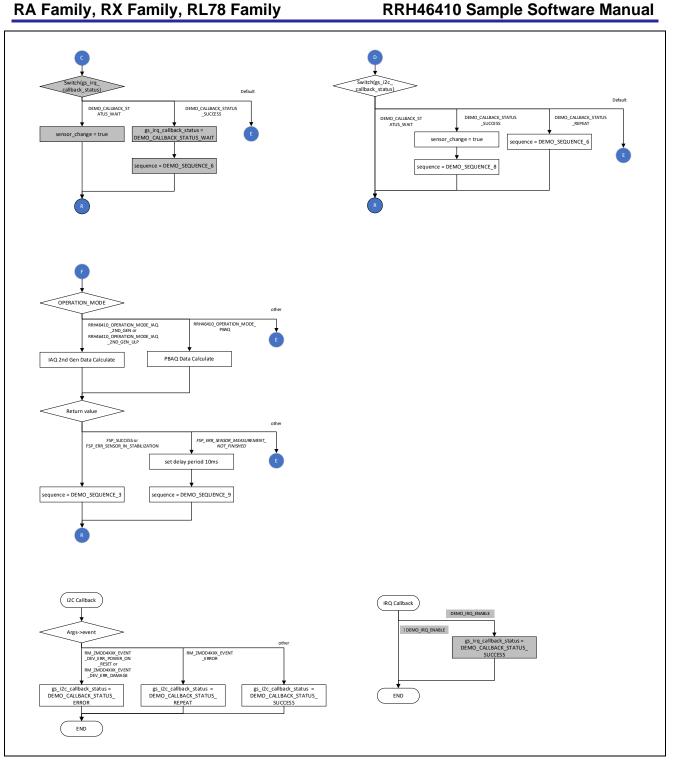


Figure 4-4 Flowchart of Main Processing in Non-OS Version of RRH46410 Sample Software (2)



4.4 Flowchart of OS Version of Sample Software

The OS version uses a semaphore in control of the sensor and operates a thread for controlling the sensor.

The sensor control in thread first starts the driver and then repeats the processing for acquiring data from the sensor and calculating values from the results of measurement.

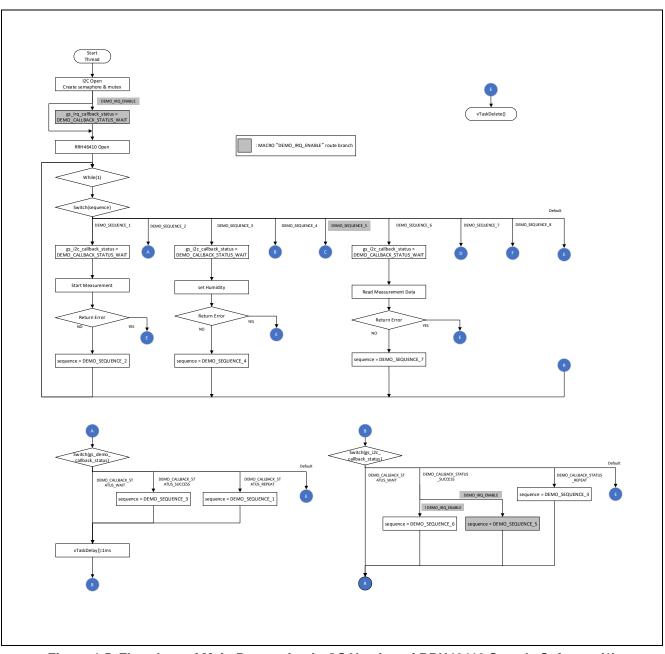


Figure 4-5 Flowchart of Main Processing in OS Version of RRH46410 Sample Software (1)



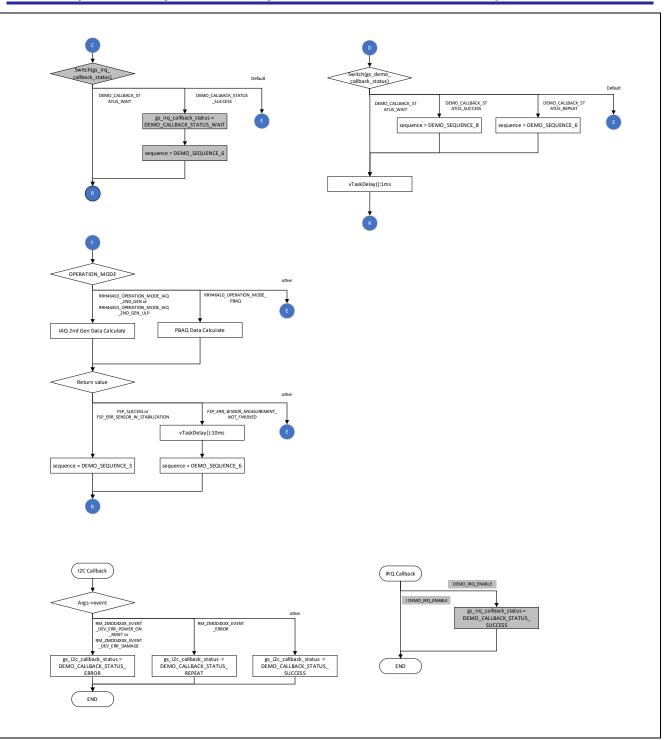


Figure 4-6 Flowchart of Main Processing in OS Version of RRH46410 Sample Software (2)



5. Configuration Settings

The following items and values can be specified.

Green setting value is an item selected by default, and Orange setting value is an item that cannot be changed.

For module names and callback function names, specify names that conform to the C language standard.

When using the different module version, the settings items and values shown below may differ.

5.1 RRH46410 Sensor Control Module Settings

5.1.1 RA Family

Select the "**rm_rrh46410**" stack in the "Stack" tabbed page of the FSP Configurator, and the configurable items are shown in the "Properties" tabbed page.

Table 5-1 RRH46410 Settings for RA Family

Configurable Item	Value	Description
Common		
Parameter Checking	Default (BSP)	Specify the include parameter check processing in
	Enabled	code.
	Disabled	When "Disabled" is specified, excluding in the code. When "Enabled" is specified, including in the code.
Operation Mode	IAQ 2nd Gen and Rel IAQ	Specify RRH46410 sensor operation mode.
	IAQ 2nd Gen ULP and Rel IAQ ULP	
	Public Building AQ Standard (PBAQ)	
Module g_rrh46410_se	nsor0 RRH46410 Gas Sensor Module (rr	n_rrh46410)
Name	g_rrh46410_sensor0	Specify the name of the module.
Comms I2C Callback	rrh46410_comms_i2c_callback	Specify the name of the user callback function. When "NULL" is specified, no callback function is used.
IRQ Callback	rrh46410_irq_callback	Specify the IRQ user callback function name. When "NULL" is specified, no callback function is used.



5.1.2 RX Family

Select the "**r_rrh46410_rx**" component on the "Component" tabbed page of the Smart Configurator, and the configurable items will be shown in the "Configure" panel.

Table 5-2 RRH46410 Settings for RX Family

Configurable Item	Value	Description
Configurations		
Parameter Checking	System Default Enabled Disabled	Specify the include parameter check processing in code. When "Disabled" is specified, excluding in the code.
		When "Enabled" is specified, including in the code.
Number of RRH46410 Sensors	1	Specify the number of RRH46410 sensor.
Operation mode of	Not selected	Specify RRH46410 sensor operation mode.
RRH46410 Sensor{x}	IAQ 2nd Gen.	
(x = 0)	IAQ 2nd Gen. Ultra-Low Power	
	PBAQ	
I2C Communication device No. for RRH46410 sensor device{x} (x = 0)	I2C Communication Device{x} (x = 0 - 15)	Specify the communications device number to be used by the sensor.
I2C callback function for RRH46410 sensor device{x} (x = 0)	rrh4640_user_i2c_callback0	Specify the name of I2C callback function.
Enable IRQ from	Disable	Specify IRQ enable or disable.
RRH46410 sensor device{x} (x = 0)	Enable	
IRQ Callback function for RRH46410 sensor device{x} (x = 0)	rrh46410_user_irq_callback0	Specify the name of IRQ callback function.
IRQ number for RRH46410 sensor device{x} (x = 0)	IRQ{x} (x = 0 - 15)	Specify valid IRQ number.
IRQ trigger for RRH46410	Low Level	Specify IRQ trigger.
sensor device {x}	Falling	When using RRH46410 Sensor Pmod Board, specify
(x = 0)	Rising	Falling. (Note 1)
	Both Edges	
IRQ interrupt priority for RRH46410 sensor device{x} (x = 0)	Priority{x} (x = 0 - 15)	Specify IRQ interrupt priority.

Note 1: The interrupt trigger setting depends on the interrupt signal circuit. For information of the interrupt circuit configuration, refer to "6.5 Notes for Interrupt Signal Circuits".

If the circuit configuration connects the RRH46410 INT signal directly to the MCU interrupt terminal, set it to "Rising".



5.1.3 RL78 Family

Select the "**r_rrh46410**" component on the "Component" tabbed page of the Smart Configurator, and the configurable items will be shown in the "Configure" panel.

Table 5-3 RRH46410 Settings for RL78 Family

Configurable Item	Value	Description
Configurations		
Parameter Checking	System Default Enabled Disabled	Specify the include parameter check processing in code. When "Disabled" is specified, excluding in the code. When "Enabled" is specified, including in the code.
Number of RRH46410 Sensors	1	Specify the number of RRH46410 sensor.
Operation mode of RRH46410 Sensor{x} (x = 0)	Not selected IAQ 2nd Gen. IAQ 2nd Gen. Ultra-Low Power PBAQ	Specify RRH46410 sensor operation mode.
I2C Communication device No. for RRH46410 sensor device{x} (x = 0)	g_comms_i2c_device{x} (x = 0 - 4)	Specify the communications device number to be used by the sensor.
I2C callback function for RRH46410 sensor device{x} (x = 0)	rrh46410_user_i2c_callback0	Specify the name of I2C callback function.
Enable INTC from RRH46410 sensor device{x} (x = 0)	Disable Enable	Specify INTC enable or disable.
INTC Callback function for RRH46410 sensor device{x} (x = 0)	rrh46410_user_irq_callback0	Specify the name of INTC callback function.
INTC number for RRH46410 sensor device{x} (x = 0)	INTP{x} (x = 0 - 15)	Specify the number of INTC



5.2 I2C Communication Middleware (COMMS_I2C) Settings

5.2.1 RA Family

Select the "**rm_comms_i2c**" stack in the "Stack" tabbed page of the FSP Configurator, and the configurable items are shown in the "Properties" tabbed page.

Table 5-4 COMMS_I2C Settings for RA Family

Configurable Item	Value	Description		
Common				
Parameter Checking	Default (BSP) Enabled	Specify the include parameter check processing in code.		
	Disabled	When "Disabled" is specified, excluding in the code. When "Enabled" is specified, including in the code.		
Module g_comms_i2c_	device0 I2C Communication Device (m	n_comms_i2c)		
Name	g_comms_i2c_device0	Specify the name of the module.		
Semaphore Timeout	0xFFFFFFF	For an RTOS project, specify the time of semaphore timeout.		
Slave Address	0x38	Specify the slave address. No setting is required as this will be overwritten by the Sensor Control module.		
Address Mode	7-Bit	Specify the number of slave address bits. No setting is required as this will be overwritten by the Sensor Control module.		
Callback	rm_rrh46410_comms_i2c_callback	Specify the name of the user callback function. No setting is required as this will be overwritten by the Sensor Control module.		
Module g_comms_i2c_	bus0 I2C Shared Bus (rm_comms_i2c)			
Name	g_comms_i2c_bus0	Specify the name of the I2C module.		
Bus Timeout	0xFFFFFFFF	Specify the time of I2C bus timeout.		
Semaphore for blocking	Unuse Use	For an RTOS project, enable or disable the blocking processing.		
Recursive Mutex for Bus	Unuse Use	For an RTOS project, enable or disable the recursive operation when blocking processing is enabled.		
Channel	0	Specify the channel number to be used. This setting is valid only when the I2C driver is "r_iic_master". When using other I2C drivers, this setting is invalid.		
Rate	Standard	Specify the bit rate.		
	Fast-mode	When using RRH46410, Standard or Fast-mode can be set. If other devices are connected on the same		
	Fast-mode plus	bus, set the transfer rate taking into consideration the transfer rate that can be set for those devices. When using "r_iica_master", specify "Standard"due to the electrical characteristics of IICAx. When using "r_sci_i2c", this setting is invalid.		



5.2.2 RX Family

Select the "**r_comms_i2c_rx**" component on the "Component" tabbed page of the Smart Configurator, and the configurable items are shown in the "Configure" panel.

Table 5-5 COMMS_I2C Settings for RX Family

Configurable Item	Value	Description
Configurations		-
Parameter Checking	System Default Enabled Disabled	Specify the include parameter check processing in code. When "Disabled" is specified, excluding in the code. When "Enabled" is specified, including in the code.
Number of I2C Shared Buses	Unused 1 2 - 16	Specify the number of communications bus lines that can be connected.
Number of I2C Communication Devices	Unused 1 2 - 16	Specify the number of I2C device that can be connected.
Blocking operation supporting with RTOS	Disabled Enabled	For an RTOS project, enable or disable the blocking operation.
Bus lock operation supporting with RTOS	Disabled Enabled	For an RTOS project, enable or disable the bus lock operation.
I2C Driver Type for I2C Shared bus{x} (x = 0 - 15)	RIIC SCI IIC Not selected	Specify the I2C bus type to be used for the communication bus. When using the "RIIC", r_riic_rx is necessary. When using the "SCI IIC", "r_sci_iic_rx is necessary. If an unused FIT module is deleted, a warning message will appear, but this does not affect the operation.
Channel No. for I2C Shared bus{x} (x = 0 - 15)	0	Specify the I2C channel number to be used for the communication bus.
Timeout for the bus lock of I2C Shared Bus{x} (x = 0 - 15)	0xFFFFFFF	Specify the time of I2C bus lock timeout.
I2C Shared Bus No. for I2C Communication Device{x} (x = 0 - 15)	I2C Shared Bus{x} (x = 0 - 15)	Specify the configuration of used communication bus.
Slave address for I2C Communication device{x} (x = 0 - 15)	0x00	Specify the slave address of the device to be connected to the communications bus. When using RRH46410, specify 0x38.
Address mode for I2C Communication device{x} (x = 0 - 15)	7 bit address mode	Specify the slave address mode. When using RRH46410, specify the 7-bit address mode.
Callback function for I2C Communication device $\{x\}$ (x = 0 - 15)	comms_i2c_user_callback{x} (x = 0 - 15)	Specify the name of the user callback function. When using r_ rrh46410_rx, specify rm_ rrh46410_callback{y} (y = 0).
Timeout for the blocking bus of I2C Communication device{x} (x = 0 - 15)	0xFFFFFFFF	Specify the time of I2C bus blocking timeout.



5.2.3 RL78 Family

Select the **"r_comms_i2c"** component on the "Component" tabbed page of the Smart Configurator, and the configurable items will be shown in the "Configure" panel.

Table 5-6 COMMS_I2C Settings for RL78 Family

Configurable Item	Value	Description
Configurations		
Parameter Checking	System Default Enabled Disabled	Specify the include parameter check processing in code. When "Disabled" is specified, excluding in the code.
		When "Enabled" is specified, including in the code.
Number of I2C Shared Buses	Unused	Specify the number of communication bus lines
	1	that can be connected.
	2 - 5	
Number of I2C communication	Unused	Specify the number of I2C devices can be
Devices	1	connected.
	2 - 5	
I2C Driver Type for I2C	IICA	Specify the I2C type to be used for the
Shared bus{x}	SAU IIC	communication bus.
(x = 0 - 4)	Not selected	When using RRH46410, specify "IICA".
Component name for the I2C bus{x} (x = 0 - 4)	Config_IIC00	Specify the I2C bus component name to be used for the communication bus.
I2C Shared Bus No. for I2C	I2C bus0	Specify the I2C bus configuration to be used for
Communication Device{x}	I2C bus1	the communication bus.
(x = 0 - 4)	I2C bus2	
	I2C bus3	
	I2C bus4	
Slave address for I2C Communication device{x} (x = 0 - 4)	0x00	Specify the slave address of the device to be connected to the communications bus. When using RRH46410, specify 0x38.
Callback function for I2C Communication device $\{x\}$ (x = 0 - 4)	comms_i2c_user_callback{x} (x = 0 - 4)	Specify the name of the user callback function. When using r_rrh46410, specify rm_rrh46410_callback{y} (y = 0).



5.3 I2C Driver Settings

5.3.1 RA Family

Select the "r_iic_master", "r_sci_i2c" or "r_iica_master" stack in the "Stack" tabbed page of the FSP Configurator, and the configurable items are shown in the "Properties" tabbed page.

Simplified I2C using Serial Array Unit (SAU) cannot be used because the clock stretch function is not supported.

(1) r_iic_master

Table 5-7 r_iic_master Settings for RA Family

Configurable Item	Value	Description			
Common	Common				
Parameter Checking	Default (BSP) Enabled Disabled	Specify the include parameter check processing in code. When "Disabled" is specified, excluding in the code. When "Enabled" is specified, including in the code.			
DTC on Transmission and Reception	Enabled Disabled	Specify whether to use the DTC for transmission and reception.			
10-bit slave addressing	Enabled Disabled	Specify whether to support 10-bit addressing for the slave address. No setting is required as this will be overwritten by COMMS_I2C.			
Module g_i2c_master0	I2C Master (r_iic_master)				
Name	g_i2c_master0	Specify the name of the module.			
Channel	0	Specify the channel number to be used. No setting is required as this will be overwritten by COMMS_I2C.			
Rate	Standard Fast-mode Fast-mode plus	Specify the bit rate. No setting is required as this will be overwritten by COMMS_I2C.			
Custom Rate (bps)	0	Specify the custom bit rate. This setting is valid when the value is other than 0. Use this setting when you want to set the low bitrate within the "Rate" setting range.			
Rise Time (ns)	120	Specify the SCL rise time according to the specifications of the target board to be used.			
Fall Time (ns)	120	Specify the SCL fall time according to the specifications of the target board to be used.			
Duty Cycle (%)	50	Specify the SCL duty cycle.			
Slave Address	0x00	Specify the slave address for the device to be connected. No setting is required as this will be overwritten by COMMS_I2C.			
Address Mode	7-Bit 10-Bit	Specify the salve address mode for the device to be connected. No setting is required as this will be overwritten by COMMS_I2C.			
Timeout Mode	Short Mode Long Mode	Specify the time of I2C bus timeout.			
Timeout during SCL low	Enabled Disabled	Specify whether to timeout can occur when SCL is held low for a duration longer than what is set in the timeout mode.			
Callback	rm_comms_i2c_callback	Set the user callback function name. No setting is required as this will be overwritten by COMMS_I2C.			
Interrupt Priority Level	Priority 0 (highest) Priority 1 Priority 2 Priority 3	Specify the interrupt priority level of the I2C bus driver.			
Pins					
SDA	Рххх	The pin numbers to be used by the driver are displayed.			
SCL	Pxxx	Use the "Pins" tabbed page to modify the pin configuration.			



(2) r_sci_i2c

Table 5-8 r_sci_i2c Settings for RA Family

Configurable Item	Value	Description
Common		
Parameter Checking	Default (BSP)	Specify the include parameter check processing in code.
	Enabled	When "Disabled" is specified, excluding in the code.
	Disabled	When "Enabled" is specified, including in the code.
DTC on Transmission	Enabled	Specify whether to use the DTC for transmission and reception.
and Reception	Disabled	
10-bit slave	Enabled	Specify whether to support 10-bit addressing for the slave
addressing	Disabled	address.
		No setting is required as this will be overwritten by COMMS_I2C.
Module g_i2c0 I2C Mas		
Name	g_i2c0	Specify the name of the module.
Channel	0	Specify the channel number to be used.
Slave Address	0x00	Specify the slave address for the device to be connected.
		No setting is required as this will be overwritten by COMMS_I2C.
Address Mode	7-Bit	Specify the salve address mode for the device to be connected.
<u> </u>	10-bit	No setting is required as this will be overwritten by COMMS_I2C.
Rate	Standard	Specify the bit rate.
	Fast-mode	When using RRH46410, Standard or Fast-mode can be set. If other devices are connected on the same bus, set the transfer
		rate taking into consideration the transfer rate that can be set for
		those devices.
Custom Rate (bps)	0	Specify the custom bit rate.
	Ū	This setting is valid when the value is other than 0. Use this
		setting when you want to set the low bitrate within the "Rate"
		setting range.
SDA Output Delay	300	Specify the SDA output delay time.
(nano seconds)		
Noise filter setting	Use clock signal divided	Specify the noise filter to be used for input signals.
	by 1 with noise filter	
	Use clock signal divided by 2 with noise filter	
	Use clock signal divided	
	by 4 with noise filter	
	Use clock signal divided	
	by 8 with noise filter	
Bit Rate Modulation	Enable	Enable or disable the bit rate modulation function.
	Disable	
Callback	rm_comms_i2c_callback	Set the user callback function name.
		No setting is required as this will be overwritten by COMMS_I2C.
Interrupt Priority	Priority 0 (highest)	Specify the interrupt priority level of the I2C bus driver.
Level	Priority 1	
	Priority 2	
	Priority 3	
RX Interrupt Priority	Priority 0 (highest)	When using DTC, specify the priority level of the reception
Level [Only used	Priority 1	interrupt.
when DTC is	Priority 2	
enabled]	Priority 3	
	Disabled	
Pins		
SDA	Pxxx	The pin numbers to be used by the driver are displayed.
SCL	Pxxx	Use the "Pins" tabbed page to modify the pin configuration.



(3) r_iica_master

When configuring IICA using FSP v5.4.0 or higher, set "SCLA Pin" and "SDAA Pin" in "Stacks" tabbed page to Pin numbers only.

Table 5-9 r_iica_master Settings for RA Family

Configurable Item	Value	Description		
Common				
Parameter	Default (BSP)	Specify the include parameter check proce	ssina in code.	
Checking	Enabled	When "Disabled" is specified, excluding in		
0	Disabled	When "Enabled" is specified, including in the		
10-bit slave	Enabled	Specify whether to support 10-bit addressing		
addressing	Disabled	address.	ig for the slave	
addrooonig	Disabled	No setting is required as this will be overwr	itten by COMMS_I2C.	
Module a jica maste	0 IICA Master (r_iica_maste			
Name	g_iica_master0	Specify the name of the module.		
Rate	Standard	Specify the bit rate.		
	Fast-mode	No setting is required as this will be overwr	itten by COMMS 12C	
	Fast-mode plus			
Custom Rate (bps)	0	Specify the custom bit rate.		
Ousion Nate (bps)	0	This setting is valid when the value is other	than 0 Llea this sattir	
		when you want to set the low bitrate within		
		range.	and read obtaining	
Signal Rising Times	0	Specify the SCL rise time according to the	specifications of the	
(us)	5	target board to be used.		
Signal Falling Times	0	Specify the SCL fall time according to the s	specifications of the	
(us)		target board to be used.		
Duty Cycle (%)	53	Specify the SCL duty cycle.		
Digital Filter	Enabled	Specify whether to use the digital filter.		
- ignai i niei	Disabled			
Address Mode	7-Bit	Specify the salve address mode for the dev	vice to be connected.	
	10-Bit	No setting is required as this will be overwr		
Slave Address	0x00	Specify the slave address for the device to	•	
		No setting is required as this will be overwritten by COMMS_I2C		
Communication	Enabled	Specify whether to use the communication		
reservation	Disabled			
Callback	rm_comms_i2c_callback	Set the user callback function name.		
		No setting is required as this will be overwr	itten by COMMS 12C.	
IICA0	Priority 0 (highest)	Specify the interrupt priority level of the I20		
communication	Priority 1			
interrupt priority	Priority 2			
	Priority 3	1		
SCLA Pin	Pxxx	Specify the pin numbers to be used.		
SDAA Pin	Pxxx	No setting is required in "Pins" tabbed page	э.	
		g_iica_master0 IICA Master (r_iica_master)		
			Value	
		Settings Property	value	
		API Info > Common		
		Name	g_iica_master0	
		Rate Signal Rising Time (us)	Standard	
		Signal Falling Time (us)	0	
		Duty Cycle (%)	53	
		Digital Filter Address Mode	Disabled 7-Bit	
		Slave Address	0x00	
		Communication reservation	Disabled	
		Callback	frm_comms_i2c_callback Priority 2	
		SCLA Pin	Priority 2 P100	
		SDAA Pin	P101	



5.3.2 RX Family

Select the "**r_riic_rx**" or "**r_sci_iic_rx**" component on the "Component" tabbed page of the Smart Configurator, and the configurable items will be shown in the "Configure" panel.

(1) r_riic_rx

Table 5-10 r_riic_rx Settings for RX Family

Configurable Item	Value	Description
Configurations		-
Set parameter checking enable	System Default Not Include	Specify the include parameter check processing in code. When "Not" is specified, excluding in the code. When "Include" is specified, including in the code.
MCU supported channels for CH{x} (x = 0 - 2)	Not supported Supported	Specify whether to support the operation of the channel. When "Not supported" is specified, excluding in the code. When "Supported" is specified, including in the code.
CH{x} RIIC bps(kbps) (x = 0 - 2)	400	Specify the bit rate. When using RRH46410, set it to 400kbps or less. If other devices are connected on the same bus, set the transfer rate taking into consideration the transfer rate that can be set for those devices.
Digital filter for CH{x} (x = 0 - 2)	Not One IIC phi Two IIC phi Three IIC phi Four IIC phi	Specify the digital filter for input signals. When "Not" is specified, disable the digital filter.
Setting port setting processing	Not include port setting Include port setting	Specify whether to include the pin function settings in the code to be generated. When "Not include port setting" is specified, excluding in the code. When "Include port setting" is specified, including in the code.
Master arbitration lost detection function for $CH{x}$ (x = 0 - 2)	Unused Used	Specify whether to use the master arbitration lost detection function. If using it in a multi-master environment, set it to "Used". When "Unused" is specified, disable. When "Used" is specified, enable.
Address $\{y\}$ format for CH $\{x\}$ (x = 0 - 2, y = 0 - 2)	Not 7 bit address format 10 bit address format	Specify whether to support 7-bit addressing or 10-bit addressing for the slave address. When using RRH46410, select "7 bit address format". Do not connect devices with different address formats on the same bus.
Slave Address {y} for CH{x} (x = 0 - 2, y = 0 - 2)	0x0025	Specify the slave address of the designated device. No setting is required as this will be overwritten by COMMS_I2C.
General call address for CH{x}	Unused Used	Specify whether to use the general call function. When "Unused" is specified, disable. When "Used" is specified, enable.
CH{x} RXI INT Priority Level (x = 0 - 2)	Level 1 Level 2 Level 14 Level 15 (highest)	Specify the priority level of the reception interrupt.



RA Family, RX Family, RL78 Family

CH{x} TXI INT Priority	Level 1	Specify the priority level of the transmission interrupt.
Level	Level 2	
(x = 0 - 2)		
	Level 14	
	Level 15 (highest)	
CH{x} EEI INT Priority	Level 1	Specify the priority level of the error interrupt.
Level	Level 2	
(x = 0 - 2)		
	Level 14	
	Level 15 (highest)	
CH{x} TEI INT Priority	Level 1	Specify the priority level of the transmission end interrupt.
Level	Level 2	
(x = 0 - 2)		
	Level 14	
	Level 15 (highest)	
Timeout function for CH{x}	Unused	Specify whether to use the timeout function.
(x = 0 - 2)	Used	When "Unused" is specified, disable.
		When "Used" is specified, enable.
Timeout detection time for	Long mode	Specify the time for timeout detection.
CH{x}	Short mode	When "Long mode" is specified, select the long mode.
(x = 0 - 2)		When "Short mode" is specified, select the short mode.
Count up during low period of timeout detection for	Unused	Specify whether to increment the counter for detecting a timeout while SCL is at the low level when the "Timeout
CH{x}	Used	function" for the specified channel is enabled.
(x = 0 - 2)		When "Unused" is specified, disable.
		When "Used" is specified, enable.
Count up during high	Unused	Specify whether to increment the counter for detecting a
period of timeout detection	Used	timeout while SCL is at the high level when the "Timeout
for CH{x}		function" for the specified channel is enabled.
(x = 0 - 2)		When "Unused" is specified, disable. When "Used" is specified, enable.
Set Counter of checking	1000	Specify the counter value to be judged to represent the bus
bus busy	1000	busy state.
Resources		-
SCLx Pins	Checked	Specify the pins to be used.
	Unchecked	Set the pin to "Checked".
SDAx Pins	Checked	
	Unchecked	



(2) r_sci_iic_rx

Table 5-11 r_sci_iic_rx Settings for RX Family

Configurable Item	Value	Description
Configurations		·
Set parameter checking	System Default	Specify the include parameter check processing in code.
enable	Not	When "Not" is specified, excluding in the code.
	Include	When "Include" is specified, including in the code.
MCU supported	Not supported	Specify whether to support the operation of channel.
channels for CH{x} (x = 0 - 12)	Supported	
SCI IIC bitrate (bps) for CH{x} (x = 0 - 12)	384000	Specify the bit rate. When using RRH46410, set it to 38400bps or less. If other devices are connected on the same bus, set the transfer rate taking into consideration the transfer rate that can be set for those devices.
Interrupt Priority for	Level 1	Specify the interrupt priority level.
CH{x}	Level 2	
(x = 0 - 12)		
	Level 14	
	Level 15 (highest)	
Digital noise filter	Disable	Specify whether to use the digital noise filter.
(NFEN bit) for CH{x} (x = 0 - 12)	Enable	
Noise Filter Setting	The clock divided by 1	Specify the sample clock of the digital noise filter.
Register (NFCS bit) for	The clock divided by 2	
CH{x} (x = 0 - 12)	The clock divided by 4	
	The clock divided by 8	
I2C Mode Register 1 (IICDL bit) for CH{x} (x = 0 - 12)	18	Specify the number of SDA output delay cycles relative to the falling edge of SSCL pin output. Set in the range of 1 to 31.
Software bus busy check counter	1000	Specify the counter value to be judged to represent the bus busy state.
Port Setting Processing	Not include port setting	Specify whether to include the pin function settings in the code
	Include port setting	for using the ports as SSCL and SSDA pins.
		Not include port setting: Omitted from the code.
Resources		Include port setting: Included in the code
SSCLx Pins	Checked	Specify the pins to be used.
	Unchecked	Set the pin to "Checked".
SSDAx Pins	Checked	
	Unchecked	
	Uncheckeu	



5.3.3 RL78 Family

Select "**IICAx**" as resource the IIC Communication (Master mode) component in the Smart Configurator, and the configurable items will be shown in the "Configure" panel.

Simplified I2C using Serial Array Unit (SAU) cannot be used because the clock stretch function is not supported.

(1) IICAx

Table 5-12 IICAx Settings for RL78 Family

Configurable Item	Value	Description
Configurations	-	
Clock mode setting	fCLK	Specify the clock to drive counting.
	fCLK/2	
Address	16	Specify the local address.
Operation mode setting	Standard	Specify the operating mode.
	Fast mode	When using RRH46410, Standard or Fast-mode can be set. If
	Fast mode plus	other devices are connected on the same bus, set the transfer rate taking into consideration the transfer rate that can be set for those devices.
Digital filter on	Checked	Specify whether to use the digital filtering.
	Unchecked	
Transfer clock (fSCL)	100000	Specify the bit rate.
		Due to the electrical characteristics of IICAx, specify 100000bps or less.
Set tR and tF manually	Checked	Manually set the SDAAn and SCLAn signal rising / falling times.
	Unchecked	
tR	0	Specify the SDAAn and SCLAn signal rising times.
tF	0	Specify the SDAAn and SCLAn signal falling times.
Communication end	Level0 (high)	Specify the priority level of the communication end interrupt.
interrupt priority	Level1	
(INTIICAx)	Level2	
	Level3 (low)	
Master transmission end	Checked	Specify whether to use the callback function when master
	Unchecked	transmission ends.
Master reception end	Checked	Specify whether to use the callback function when master
	Unchecked	reception ends.
Master error	Checked	Specify whether to use the callback function when a
	Unchecked	communication error occurs.
Generated stop condition	Checked	Specify whether to generate a stop condition in the callback
in master transmission / reception end callback function	Unchecked	function. Set to "Unchecked".



5.4 IRQ Driver Settings

For information on matching the interrupt signal circuit, refer to "6.5 Notes for Interrupt Signal Circuits".

5.4.1 RA Family

Select the "**r_icu**" stack in the "Stack" tabbed page of the FSP Configurator, and the configurable items are shown in the "Properties" tabbed page.

The following is a configuration example for the RRH46410 Sensor Pmod Board.

Table 5-13	r icu	Settinas	for RA	Family
1 4 6 1 6 1 6		o o tuni go		

Configurable Item	Value	Description
Common		
Parameter Checking	Default (BSP)	Specify the include parameter check processing in code.
	Enabled	When "Disabled" is specified, excluding in the code.
	Disabled	When "Enabled" is specified, including in the code.
Module g_external_irq0 Exte	ernal IRQ (r_icu)	
Name	g_external_irq0	Specify the name of the module.
Channel	0	Specify the channel number to be used.
Trigger	Falling	Specify the trigger.
	Rising	When using RRH46410 Sensor Board, select "Falling".
	Both Edges	(Note 1)
	Low Level	
Digital Filtering	Enabled	Specify whether to use the digital filtering.
	Disabled	
Digital Filtering Sample	PCLK / 1	Specify the sample clock of digital filtering.
Clock (Only valid when	PCLK / 8	
Digital Filtering is Enabled)	PCLK / 32	
	PCLK / 64	
Callback	rm_rrh46410_irq_callback	Set the user callback function name.
		The name of the user callback function is automatically specified by r_icu.
Pin Interrupt Priority	Priority 0 (highest)	Specify the interrupt priority level of the IRQ driver.
	Priority 1	
	Priority 2	
	Priority 3	
Pins		
IRQ	Pxxx	The pin numbers to be used by the driver are displayed. Use the "Pins" tabbed page to modify the pin configuration.

Note 1: The interrupt trigger setting depends on the interrupt signal circuit. For information of the interrupt circuit configuration, refer to "6.5 Notes for Interrupt Signal Circuits". If the circuit configuration connects the RRH46410 INT signal directly to the MCU interrupt terminal, set it to "Rising".



5.4.2 RX Family

Select the "**r_irq_rx**" component on the "Component" tabbed page of the Smart Configurator, and the configurable items will be shown in the "Configure" panel.

The following is a configuration example for the RRH46410 Sensor Pmod Board.

Table 5-14 r_irq_rx Settings for RX Family

Configurable Item	Value	Description			
Configurations					
Locking function for	Enabled	Enable or disable the locking functions for the IRQ APIs.			
IRQ APIs	Disabled				
Set parameter checking	System Default Specify the include parameter check processing in code.				
enable	Not	When "Disabled" is specified, excluding in the code.			
	Include	When "Enabled" is specified, including in the code.			
Filter for IRQ{x}	Enabled	Specify whether to use the digital filtering.			
(x = 0 - 15)	Disabled				
Filter clock divisor for	Divisor 1	Specify the sample clock of digital filtering.			
IRQ{x}	Divisor 8				
(x = 0 - 15)	Divisor 32				
	Divisor 64				
IRQx Pins	Checked	Specify the pins to be used.			
	Unchecked	Set the pin to "Checked".			

5.4.3 RL78 Family

Select the "Interrupt Controller" component on the "Component" tabbed page of the Smart Configurator, and the configurable items will be shown in the "Configure" panel.

The following is a configuration example for the RRH46410 Sensor Pmod Board.

Table 5-15 Interrupt Controller Settings for RL78 Family

Configurable Item	Value	Description
Configurations		
INTP{x}	Checked	Enable or disable INTP{x}.
(x = 0 - 11)	Unchecked	(x = 0 - 11)
Valid edge	Falling edge	Specify the trigger.
	Rising edge	When using RRH46410 Sensor Board, select "Falling". (Note 1)
	Both edges	
Priority	Level 0 (high)	Specify the interrupt priority level of the INTP{x}.
	Level 1	(x = 0 - 11)
	Level 2	
	Level 3 (low)	

Note 1: The interrupt trigger setting depends on the interrupt signal circuit. For information of the interrupt circuit configuration, refer to "6.5 Notes for Interrupt Signal Circuits".

If the circuit configuration connects the RRH46410 INT signal directly to the MCU interrupt terminal, set it to "Rising".



6. Guide for Changing Target Device

Use the following procedures to change the target device to a new one and run a sample project on the new device.

Therefore, you will need to review the following settings according to the user target circuit.

- Interrupt Signal Circuit: Refer to "6.5 Notes for Interrupt Signal Circuits".
- RESET Signal Circuit: Refer to "6.6 Notes for RESET Signal Circuits".

6.1 Importing Sample Project

To change a device in a sample project, need to import.

To import a sample project, follow the steps below.

1. Select [Import] from the menu.

The "Import" window will appear. Select "Rename & Import Existing C/C++ Project into Workspace" in the window and press the [Next] button.

🕲 Import		×
Select		-
Rename and Import and Existing C/C++ Project into the workspace	2	
Select an import wizard:		
type filter text		
V 🍃 General		^
🕼 Archive File		
CMSIS Pack		
😭 Existing Projects into Workspace		
😂 File System		
Preferences		
Projects from Folder or Archive		
😭 Rename & Import Existing C/C++ Project into Workspace		
Renesas CC-RX project conversion to Renesas GCC RX		
Renesas CS+ Project for CA78K0R/CA78K0		
Renesas CS+ Project for CC-RX, CC-RL and CC-RH		
Renesas GitHub FreeRTOS (with IoT libraries) Project		
Cample Projects on Renesas Website		
> > C/C++		~
Over the second seco	Cance	:



2. Press the [Browse] button to open the "Select Folder" window.

Select the folder of the original project for the current device from a list of imported sample projects and press the [Select Folder] button.

→ Y ↑ → This PC	C > Downloads > r01an0000xx0100-sense	or 🗸 🖑 Sea	arch r01an0000xx0100	-sensor
rganize 🔻 New folder				•
🖈 Ouick access	Name	Date modified	Type Si	ize
	SENSOR_RA0E1_NonOS	11/21/2024 1:59 PM	File folder	
lene OneDrive	SENSOR_RA2E1_FreeRTOS	11/21/2024 1:58 PM	File folder	
OneDrive - Personal	SENSOR_RA2E1_NonOS	11/21/2024 1:58 PM	File folder	
Chebrite Tersonal	SENSOR_RL78G23_NonOS	11/21/2024 1:49 PM	File folder	
💻 This PC	SENSOR_RL78G23_NonOS_LLVM	11/21/2024 1:49 PM	File folder	
🔿 Network	SENSOR_RX140_FreeRTOS	11/21/2024 1:49 PM	File folder	
	SENSOR_RX140_NonOS	11/21/2024 1:49 PM	File folder	
Folder: S	SENSOR_RA0E1_NonOS			

3. Enter the project name, select the original project for the current device, and press the [Finish] button.

📴 Import				_	_		×	
Rename & Import Project Select a directory to search for existing Eclipse projects.							7	
Project name:	SamplePro	oject						
Use default location								
Location: C:\v		orkspace\e2	2_studio\Sample	Project\SamplePrc	Br	rowse		
	⊡ Cr	eate Directo	ry for Project					
Choose file sys	Choose file system: default \vee							
Import from:								
Select root	directory:	C:\Users\x	xxxxxxx\Downl	pads\r01an000C \vee	Br	owse		
○ Select archi	ive file:			~	Br	owse		
Projects:								
SENSOR_RA	AOE1_NonO	S (C:\Users\	xxxxxxx\Dowr	loads\r01an0000xx	0100-se	nsor\SE	INSC	
<					-		>	
Options								
?	~	Back	Next >	Finish		Cancel		



6.2 RA Sample Project

After importing the sample projects, follow the steps below. Please refer to "6.1 Importing Sample Project" for importing instructions.

The following explains the change procedure for the following board change example. In addition, an Interposer Board is required when using a Pmod Type 2A/3A connector.

 Sample Project "RRH46410_RA2E1_NonOS": Pmod1 (Type 2A/3A: SCI0)
 → Pmod1 (Option Type 6A: IIC1) or Pmod2 (Type 2A: SCI0) of the EK-RA6M4 board

6.2.1 Modifying Settings of FSP Configurator

Double-click on "Configuration.xml" in the project tree to open the FSP Configurator.

(1) BSP

Change the settings of "Board" and "Device" in the "BSP" tabbed page.

When selecting a Renesas board, modify the "Board" setting only.

When selecting a board provided from other companies, change the "Board" setting to "Custom User Board (Any Device)" and then change the "Device" setting to the new device to be used.

Board Supp	oort Package Configuration	Generate Project Content
		🔜 Restore Defaults
Device Selec	tion	
Device Selec FSP version: Board: Device: Core: RTOS:		Board Details Evaluation kit for RA2E1 MCU Group Visit <u>https://www.renesas.com/ra/ek-ra2e1</u> to get kit user's manual, quick start guide, errata, design package, example projects, etc.
Summary BSP	Clocks Pins Interrupts Event Links Stacks Compone	ents



(2) Clocks

Set up the clocks in the "Clocks" tabbed page.

When "Custom User Board (Any Device)" is selected for "Board", set up the clocks according to the specifications of the target board to be used.

When a Renesas board is selected for "Board", the clocks are automatically set up.

Clocks Configuration	Generate Project Content				
	Restore Defaults				
XTAL 24MHz Clock Src: PLL V XIAL Div /1	✓ → ICLK 200MHz				
→ PCLKA Div /2	✓ → PCLKA 100MHz				
HOCO 20MHz V	✓ → PCLKB 50MHz				
LOCO 32768Hz	✓ → PCLKC 50MHz				
MOCO 8MHz >> PCLKD Div /2	✓ → PCLKD 100MHz				
SUBCLK 32768Hz	✓ → BCLK 100MHz				
> PLL Src: XTAL V EBCLK Div /2	✓ → EBCLK 50MHz				
PLL Div /3 V	✓ → FCLK 50MHz				
PLL Mul x25.0 V					
PLL 200MHz					
PLL2 Disabled V					
PLL2 Div /2 \checkmark CLKOUT Disabled \checkmark CLKOUT Div /1	✓ → CLKOUT 0Hz				
PLL2 Mul x20.0 \checkmark \Rightarrow UCLK Disabled \checkmark \rightarrow UCLK Div /5	✓ → UCLK 0Hz				
PLL2 0Hz OCTASPICLK Disabled V> OCTASPICLK Di	iv /1 V> OCTASPICLK 0Hz				
Summary BSP Clocks 🕴 Pins Interrupts Event Links Stacks Components					



(3) Pins

(a) Changing Board

In the "Pins" tabbed page, modify the pin configuration according to the specifications of the target board to be used.

When using a Renesas board, change the selection for "Select Pin Configuration" from "RA2E1 EK" to the target board; appropriate pins are automatically assigned.

Pin Configuration				Generate Project Content
Select Pin Configuration		📑 Export to CS	V file 🔚 Configure P	in Driver Warnings
RA2E1 EK	✓ Manage configurations	Gener	ate data: g_bsp_pin_c	fg
Pin Selection $\exists \exists \exists \exists \exists z \in I^{a}$	Pin Configuration			😲 Cycle Pin Group
Type filter text	Name	Value	Link	
> V P0 > V P1 > V P2 > V P3				
> V P4 > V P5 > P6 > P7				
> P8 > * Other Pins * Peripherals > * Analog:ADC				
 Analog:ANALOG Analog:DAC Connectivity:CAN Connectivity:ETHERC 				
	<			>
Connectivity:SSI Connectivity:USB Input:CTSU Monitoring:CAC				
Pin Function Pin Number Summary BSP Clocks O Pins Interrupts	Event Links Stacks Components			

If the desired board is not displayed in the drop-down list for "Select Pin Configuration", click on [Manage Configuration] to open the "Manage Pin Configurations" window and select the desired board in the window.

Manage Pin Configurations		×
Multiple Pin Configuration Management		
Modify pin configuration list or import/export external file		
RA2E1 EK (Current) RA6M4 EK	Add	
R7FA6M4AF3CFB.pincfg	Remove	:
	Rename.	
	Duplicat	e
	Merge to	
	Import	
	Export	
	0.1	
	OK	



(b) Changing I2C I/F Pins

However, the assignment on the above "(a)Changing Board" will apply the SPI communication pin settings that support Pmod Type 2A on the EK-RA6M4 board.

This sample software uses Pmod Type 6A, therefore it is necessary to change the I2C communication pin settings that support Pmod Type 6A.

IIC1 (Pmod1 #3 P512 SCL1 and Pmod1 #4 P511 SDA1) is assigned to Pmod1 and SCI0 (Pmod2 #3 P410 SCL0 and Pmod2 #2 P411 SDA0) is assigned to Pmod2 on the EK-RA6M4 board.

Therefore, the pins used for I2C communication are as follows, so after automatic assignment of "Select Pin Configuration", reconfigure in "Pin Configuration":

- When using Pmod1 (Option Type 6A), set SCL1 to P512 and SDA1 to P511.
- When using Pmod2 (using the Interposer Board), set SCL0 to P410 and SDA0 to P411.

Pin Configuration					Generate Project Content
Select Pin Configuration		📑 Ехро	ort to CSV file 📲	Configure	Pin Driver Warnings
RA6M4 EK	✓ Manage configurations		Generate data:	g_bsp_pin	_cfg_6m4
Pin Selection $ \equiv $	Pin Configuration				😲 Cycle Pin Group
Type filter text	Name	Value	Lock	Link	
> V Other Pins	Pin Group Selection	Mixed			
V V Peripherals	Operation Mode	Simple I2C			
> ✓ Analog:ADC	✓ Input/Output			$\langle \rangle$	
> ✓ Analog:ANALOG	TXD0	None		\Rightarrow	
> Analog:DAC	RXD0	None			
> Connectivity:CAN	SCK0	None		\Rightarrow	
> Connectivity:ETHERC</td <td>CTS0</td> <td>None</td> <td></td> <td>\Rightarrow</td> <td></td>	CTS0	None		\Rightarrow	
> 🗸 Connectivity:IIC	SDA0	✓ P411		4	
✓ ✓ Connectivity:SCI	SCL0	✓ P410	f	4	
✓ SCI0	CTSRTS0	None		\Rightarrow	
SCI1					
SCI2					
SCI3					
SCI4					
SCI5	<				>
✓ SCI6	Module name: SCI0				
✓ SCI7					
SCI8		le I2C mode, ensure po between I2C and other r			pen drain.
SCI9 🗸	when switching t	etween ize and other r	modes, first disab	ne.	
Die Function Die Number	I L				
Pin Function Pin Number					
Summary BSP Clocks Pins Interrupts Ev	rent Links Stacks Components				



(c) Changing IRQ Pin

For information on configuring the pin of the interrupt signal circuit, refer to "6.5 Notes for Interrupt Signal Circuits".

Set the IRQ pin to match the interrupt signal pin of the sensor.

P301 IRQ06 is assigned to Pmod1 #1 and P414 IRQ09 is assigned to Pmod2 #7 on the EK-RA6M4 board.

Therefore, set the IRQ pin as follows:

- When using Pmod1 (Option Type 6A), set IRQ06 pin to P301.
- When using Pmod2 (using the Interposer Board), set IRQ09 pin to P414.

elect Pin Configuration			Export to CSV file	Configure	Pin Driver Warnings
RA6M4 EK	✓ Manage configuration	<u>ns</u>	🗹 Generate data:	g_bsp_pin_	_cfg_6m4
in Selection 📰 🕀 📮	Pin Configuration				😲 Cycle Pin Group
Type filter text	Name	Value	Lock	Link	^
> ✓ Other Pins ∧	IRQ00	None		\Rightarrow	
✓ Other Pins ✓ Peripherals	IRQ01	None		\Rightarrow	
Analog:ADC	IRQ02	None		\Rightarrow	
> V Analog:ADC	IRQ03	None		\Rightarrow	
	IRQ04	None		\Rightarrow	
	IRQ05	None		\Rightarrow	
> Connectivity:CAN > Connectivity:ETHERC</td <td>IRQ06</td> <td>✓ P409</td> <td></td> <td>\Rightarrow</td> <td></td>	IRQ06	✓ P409		\Rightarrow	
> V Connectivity:Effect	IRQ07	None		\Rightarrow	
> V Connectivity:SCI	IRQ08	P002	La Carlo Car	\Rightarrow	
> V Connectivity:SPI	IRQ09	🗸 P414	a a a a a a a a a a a a a a a a a a a	\Rightarrow	
> Connectivity:SSI	IRQ10	P005		\Rightarrow	
> V Connectivity:USB	IRQ11	P006		\Rightarrow	
> Input:CTSU	IRQ12	P008		\Rightarrow	
✓ ✓ Input:ICU	IRQ13	None		\Rightarrow	
V ICU0	IPO14	Mono			~
 Monitoring:CAC Storage:OSPI Storage:QSPI Storage:SDHI 	Module name: ICU0 Usage: To use IRC	Q function with output or	peripheral modes, cha	nge directly	



(d) Changing General Purpose I/O Port Pin: RESET

For information on configuring the pin of the RESET signal circuit, refer to "6.6 Notes for RESET Signal Circuits".

Set the RESET pin to match the RESET signal pin of the sensor.

P203 Port is assigned to Pmod1 #2 and P412 Port is assigned to Pmod2 #4 on the EK-RA6M4 board.

Therefore, set the RESET pin as follows:

- When using Pmod1 (Option Type 6A), set P203 to GPIO Port Output pin.
- When using Pmod2 (using the Interposer Board), set P412 to GPIO Port Output pin. For both, select "Output mode (Initial High)" as "Mode".

Pin Configuration				Generate Project Content
Select Pin Configuration		📑 Export to CS	SV file 🗄	Configure Pin Driver Warnings
RA6M4 EK	✓ Manage configurations	🗹 Gener	ate data:	g_bsp_pin_cfg_6m4
Pin Selection $\exists \exists \exists \exists \exists \exists z $	Pin Configuration			😲 Cycle Pin Group
Type filter text > * P2 > * P3 * * P400 * P401 * P402 * P403 * P404 * P405 * P406 * P407 * P408 * P409 * P410 * P411 * P412 P413 * P414 * P415	Name Symbolic Name Comment Mode Pull up Drive Capacity Output type VInput/Output P412 Module name: P412 Port Capabilities: AGT1: AGTEE1 CTSU0: TS08 ETHERC_MII0: E	Value PMOD2_CLK Output mode (Initial High) None Low CMOS GPIO GPIO TO_ETXD0	Link	
Pin Function Pin Number Summary BSP Clocks Pins Interrupts Ex	ent Links Stacks Components			

When you change the device, "Generate data" will be disabled. The next page explains how to enable it.

Pin Configuration				Generate Project Content
Select Pin Configuration			Export to CSV file	e 🖺 Configure Pin Driver Warnings
RA6M4 EK	✓ Manage configurations		Generate d	ata:
Pin Selection	Pin Configuration			😲 Cycle Pin Group
Type filter text	Name	Value	Link	



To enable generation of pin settings, check [Generate data] check-box and enter a desired name in the text field.

The entered name is linked to the pin configuration, therefore must use a unique name that does not duplicate with other pin configurations.

The following is an example named "g_bsp_pin_cfg_6m4".

in Configuration				Generate Project Content
elect Pin Configuration			🛔 Export to CSV file 🛛 🖺 Configure Pin Di	iver Warnings
RA6M4 EK	✓ Manage configurations		Generate data: g_bsp_pin_cfg_6	m4
Pin Selection $ \equiv $	Pin Configuration			😲 Cycle Pin Group
Type filter text Ports PO P1 P2 P3 P4 P5 P6 P7 P8 Other Pins Peripherals Analog:ADC Analog:ADC Analog:DAC Connectivity:CAN Connectivity:ETHERC Connectivity:SCI Connectivity:SPI To Connectivity:SPI To Connectivity:SPI 	Name	Value	Link Link I I I I I I I I I I I I I I I I I I	



(4) Stacks

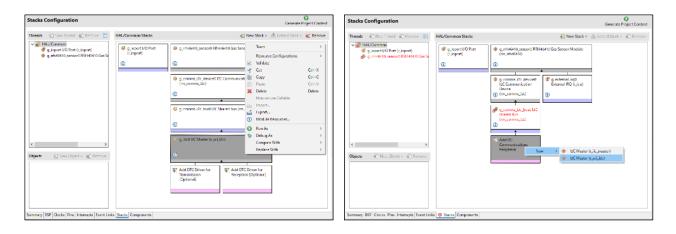
Modify the configuration of individual components in the "Stacks" tabbed page.

(a) Changing COMMS_I2C Settings and I2C Driver Settings

Modify the settings of COMMS_I2C and I2C driver according to the specifications of the target board. To use the pins of the I2C I/F, delete the unnecessary stack and add the new stack to use.

Table 6-1 Settings of I2C I/F and Channel for EK-RA6M4

EK-RA6M4	12C I/F	g_comms_i2c_bus0 I2C Shared Bus (rm_comms_i2c)	g_i2c_master0 I2C Master
Pmod1 Option Type 6A	IIC1	Channel: 1	Check Pins
Pmod2 Type 2A	SCI0	Channel: 0	Check Pins



g_comms_i2c_bus0 I2C Shared Bus (rm_comms_i2c)							
Settings API Info	Property ✔ Common	Value					
	Parameter Checking	Default (BSP)					
	 Module g_comms_i2c_bus0 I2C Shared Bus (rm_comms_i2c) 						
	Name	g_comms_i2c_bus0					
	Bus Timeout	0xFFFFFFF					
	Semaphore for Blocking (RTOS only)	Use					
	Recursive Mutex for Bus (RTOS only)	Use					
	Channel	1					
	Rate	Standard					

g S

ttings	Property	Value
l Info	✓ Common	
TIMO	Parameter Checking	Default (BSP)
	DTC on Transmission and Reception	Disabled
	10-bit slave addressing	Disabled
	 Module g_i2c_master0 I2C Master (r_iic_master) 	
	Name	g_i2c_master0
	Channel	1
	Rate	🔒 Standard
	Custom Rate (bps)	0
	Rise Time (ns)	120
	Fall Time (ns)	120
	Duty Cycle (%)	50
	Slave Address	0x00
	Address Mode	7-Bit
	Timeout Mode	Short Mode
	Timeout during SCL Low	Enabled
	Callback	🔒 rm_comms_i2c_callback
	Interrupt Priority Level	Priority 12
	✓ Pins	
	SDA1	P511
	SCL1	P512

j_i2c0 l20	Master (r_sci_i2c)	
Settings	Property	Value
API Info	✓ Common	
	Parameter Checking	Default (BSP)
	DTC on Transmission and Rec	Disabled
	10-bit slave addressing	Disabled
	✓ Module g_i2c0 I2C Master (r_sci_	
	Name	g_i2c0
	Channel	0
	Slave Address	0x00
	Address Mode	7-Bit
	Rate	Standard
	Custom Rate (bps)	0
	SDA Output Delay (nano seco	300
	Noise filter setting	Use clock signal divided by 1 with noise filter
	Bit Rate Modulation	Enable
	Callback	🔒 rm_comms_i2c_callback
	Interrupt Priority Level	Priority 2
	RX Interrupt Priority Level [Or	Disabled
	✓ Pins	
	SDA0	P411
	SCL0	P410



(b) Changing General Purpose I/O Port Driver Settings: RESET

For information on how to modify the source of the RESET signal control, refer to "6.2.2(1) RESET Signal Control".

Enter the pin configuration name to use in "Pin Configuration Name" of "g_ioport I/O Port".

The following is an example named "g_bsp_pin_cfg_6m4".

Threads New Thread Remove HAL/Common Stacks New Stack > Extend Stack > Image: Stack > Image	rate Project Content
Image: Construction of the sensor of the	ack > 📓 Remove
Image: state	
(rm_comms_i2c) Extern	
(rm_comms_i2c) Extern	
	rnal_irq0 al IRQ (r_icu)
to g_comms_i2c_bus0 l2C Shared Bus (rm_comms_i2c)	
0	
Objects 🔄 New Object > 🎪 Remove	
Add DTC Driver for Transmission [Optional]	
Summary BSP Clocks Pins Interrupts Event Links Stacks Components	

g_ioport	I/O Port (r_ioport)		
Settings	Property v Common	Value	^
API Info	Parameter Checking V Module g_ioport I/O Port (r_ioport)	Default (BSP)	
	Name 1st Port ELC Trigger Source	g_ioport Disabled	
	2nd Port ELC Trigger Source	Default (BSP) g_ioport Disabled Disabled Disabled g_bsp_pin_cfg_6m4 P300 P110 P109 P108 <unavailable> <unavailable> <unavailable> <unavailable> <unavailable> <unavailable> <unavailable> <unavailable> <unavailable> <unavailable></unavailable></unavailable></unavailable></unavailable></unavailable></unavailable></unavailable></unavailable></unavailable></unavailable>	
	3rd Port ELC Trigger Source 4th Port ELC Trigger Source	Disabled	
	Pin Configuration Name Pins	g_bsp_pin_ctg_6m4	
	TCK TDI		- 11
	TDO TMS		
	SWCLK	<unavailable></unavailable>	
	TRACESWO	<unavailable></unavailable>	
	TCLK TDATA0	- arran arrange	_
	TDATA1	<unavailable></unavailable>	~
	<		>



(c) Changing IRQ Driver Settings

For information on matching the interrupt signal circuit, refer to "6.5 Notes for Interrupt Signal Circuits".

Modify the settings of r_icu according to the specifications of the target board.

P301 IRQ6 is assigned to Pmod1 #1 and P414 IRQ9 is assigned to Pmod2 #7 on the EK-RA6M4 board.

Therefore, set IRQ channel as follows:

- When using Pmod1 (Option Type 6A), set "Channel" to 6.
- When using Pmod2 (using the Interposer Board), set "Channel" to 9.

Stacks Co	nfiguration				Generate Project Content
Threads	🖹 New Thread 💼 Remove 📄	HAL/Common Stacks		🛃 New Stack > 💡	坐 Extend Stack > 🛛 😹 Remove
	/Common _ioport I/O Port (r_ioport) _rrh46410_sensor0 RRH46410 Gas Se	<pre> g_ioport I/O Port (r_ioport) </pre>	🕀 g_rrh46410_sensor0 RRH	146410 Gas Sensor Module (rm_	rrh46410)
		i	(j)		
			g_comms_i2c_device0 I2	C Communication Device	₽ g_external_irq0
			(rm_comms_i2c)		External IRQ (r_icu)
			(i)	<u>۸</u>	0
			g_comms_i2c_bus0 l2C \$	Shared Bus (rm_comms_i2c)	
			i		
<	>		g_i2c_master0 I2C Master	er (r_iic_master)	
Objects	🐑 New Object > 🔊 Remove		1		
				•	
			Add DTC Driver for Transmission [Optional]	Add DTC Driver for Reception [Optional]	
Summary BSI	P Clocks Pins Interrupts Event Link	s Stacks Components			
g_extern	al_irq0 External IRQ (r_i	icu)			
Settings	Property			Value	
API Info	V Common				
	Parameter Checking			Default (BSP)	
	 Module g_external_irq0 Ex 	(ternal IRQ (r_icu)			
	Name			g_external_irq0	
	Channel			9	
	Trigger			Falling	
	Digital Filtering			Disabled	
		e Clock (Only valid when Di	gital Filtering is Enabled)		
	Callback			rm_rrh46410_irq_call	back
	Pin Interrupt Priority			Priority 2	
	✓ Pins				
	IRQ09			P414	
	<				>

If an error is displayed in other stacks, modify the specified item according to the displayed error.

(5) Code Generation and Build

After modifications are finished, press [Generate Project Content] to generate files.

Build the project after implementing "6.2.2 Changing Sample Code"

Select [Debug Configurations] from the menu and modify the debugger settings according to the specifications of the emulator to be connected to the target board.



6.2.2 Changing Sample Code

(1) **RESET Signal Control**

For pin configuration, refer to "6.2.1(3)(d) Changing General Purpose I/O Port Pin: RESET"

Open "hal_entry.c" (Non-OS) or "rrh46410_sensor_thread_entry.c" (FreeRTOS) and change the reset operation when resetting the sensor. Also, change the reset control logic to an appropriate one.

Modify RESET pin designation according to the specifications of the target board.

With the above settings, the EK-RA6M4 board is configured as follows:

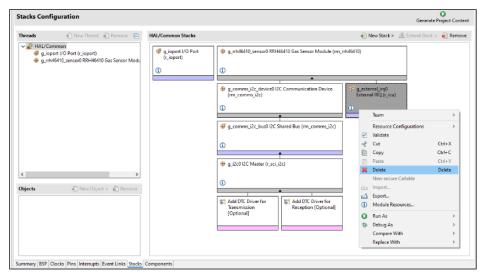
- When using Pmod1 (Option Type 6A), RESET pin is assigned to P203.
- When using Pmod2 (using the Interposer Board), RESET pin is assigned to P412.

```
/* Reset RRH46410 sensor (active low). Please change to the IO port connected to
the RES_N pin of the RRH46410 sensor on the customer board. */
R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_04_PIN_12, BSP_IO_LEVEL_HIGH);
R_BSP_SoftwareDelay(10, BSP_DELAY_UNITS_MILLISECONDS);
R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_04_PIN_12, BSP_IO_LEVEL_LOW);
R_BSP_SoftwareDelay(10, BSP_DELAY_UNITS_MILLISECONDS);
R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_04_PIN_12, BSP_IO_LEVEL_HIGH);
R_BSP_SoftwareDelay(10, BSP_DELAY_UNITS_MILLISECONDS);
R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_04_PIN_12, BSP_IO_LEVEL_HIGH);
R_BSP_SoftwareDelay(10, BSP_DELAY_UNITS_MILLISECONDS);
```

6.2.3 Changing when not using IRQ

If IRQ is not used, delete stack, and change the values of constants defined.

In the "Stacks" tabbed page, delete "g_external_irq0 External IRQ".



Open "RA_RRH46410.c" (Non-OS) or "rrh46410_sensor_thread_entry.c" (FreeRTOS) and change the value of "G_RRH46410_SENSOR0_IRQ_ENABLE" to "0".

/* TODO: Enable if you want to open RRH46410 */ #define G RRH46410 SENSORO IRQ ENABLE (0)

6.2.4 Changing Toolchain Setting

If you want to use a toolchain other than the GCC ARM Embedded toolchain, copy RA_RRH46410.c (Non-OS) or rrh46410_sensor_thread_entry.c, sensor_thread_common.c and sensor_thread_common.h (FreeRTOS) from this project to create a new project.



6.3 RX Sample Project

After importing the sample projects, follow the steps below. Please refer to "6.1 Importing Sample Project" for importing instructions.

The following explains the change procedure for the following board change example. In addition, an Interposer Board is required when using a Pmod Type 2A/3A connector.

 Sample project "RRH46410_RX140_NonOS": Pmod1 (Type2A: SCI5)
 → Pmod1 (Option Type 6A: RIIC0) or Pmod2 (Option Type6A: SCI11) of the EK-RX671 board

6.3.1 Modifying Settings of Smart Configurator

On the project tree, double-click on the .scfg file of the imported project in the Smart Configurator window will open.

Project Explorer 🗙	
✓ ﷺ SampleProject [HardwareDebug]	
> 🔊 Includes	
> 🔑 src	
🗴 SampleProject HardwareDebug.launch	
SENSOR_RX140_NonOS.rcpc	
SENSOR_RX140_NonOS.scfg	
> ⑦ Developer Assistance	

(1) Board

1. On the Board tab, click the [...] button.

Device s	election		🐻 Generate Code	🕒 Generate Report
Device se	lection			2 2
Board: Device:	FPB-RX140 R5F51406BxFN Download more boards			
	Selection			i
		n from the table below and click on the "Add" button. ponent can be further configured in the "Components" page.		
Features	;	Components	Action	Link
Appl	ication Header	١	Add	
LEDs		Ports	Add	
PMO	D 1 (UART/SPI/IIC)	 SCI Driver - UART (r_sci_rx) 	Add	\Rightarrow
PMO	D 2 (UART/SPI/IIC)	 SCI Driver - UART (r_sci_rx) 	Add	
User	Switches	IRQ Driver (r_irq_rx)	Abda	
<				>
Overview B	oard Clocks System Compo	nents Pins Interrupts		



2. Select a desired board or device in the "Change Device" window and press the [Next] button.

Refactoring				_		×
Change Devic Select the new	e device for SamplePro	oject				
Current Device: Current Board:						
Target Board:	EK-RX671					~
Target Device:	R5F5671EHxFB			<u>Download ac</u>	Unlock Dev	
					OTHOCK DEV	<u>ices</u>
Bank Mode	Single Bank					\sim
?		< Back	Next >	Finish	Cance	el l

3. If a warning message appears, read it and check if there is a problem in proceeding with the procedure. Press [Next] to move to the next step.

Review the information provided in the list below. Click 'Next >' to view the next item or 'Finish'.					
Change Device					4
Review the information provided in the	list below. Click 'l	Next >' to view the r	next item or 'Fin	ish'. 🖃	
Found problems				ł	🖟 û
		•			ı may n
	se make sure you	backup this project	before continui	ng.	
<					>
?	< Back	Next >	Finish	Cance	el



4. The changes you have made in the settings will be displayed. Press the [Finish] button to apply the changes to the project.

Sefactoring	_		×
Change Device			-
The following changes to 4 files are necessary to perform the refactoring.		-6	
Changes to be performed		₽ 0	7 -
🗸 🔽 🔂 Change Device for SampleProject			
V 🖂 🔁 Launch Configurations			
SampleProject HardwareDebug			
> 🔽 🔂 Build Settings			
✓ A Project Files			
🗹 🚖 Smart Configurator			
No preview available			
	_		
Sack Next > Finish		Can	cel

5. Select the "Board" tabbed page to check that the board and device have been changed correctly.

election
lection
EK-RX671 ~
R5F5671EHxFB
Download more boards



(2) Clocks

Set up the clocks in the "Clocks" tabbed page.

When "Custom User Board (Any Device)" is selected for "Board", set up the clocks according to the specifications of the target board to be used.

When a Renesas board is selected for "Board", the clocks are automatically set up.

		SCKCR (FCK[3:0])	FlashIF clock (FCLK)
VCC: 3.3 (V) (Actual value: 3.3)	PLL circuit	×1/4 *	60 (MHz)
	Frequency Division:	SCKCR (ICK[3:0])	System clock (ICLK)
	• tx	• x1/2 •	120 (MHz)
✓ Main clock	Frequency Multiplication:	SCKCR (PCKA[3:0])	Peripheral module clock (PCLKA)
Oscillation source: Resonator +	x10.0 +	• x1/2 •	120 (MHz)
Frequency: 24 (MHz)		SCKCR (PCKB[3:0])	Peripheral module clock (PCLKB)
	+	• x1/4 •	60 (MHz)
Oscillation wait time: 9980 (µs) (Actual value: 10000)		SCKCR (PCKC[3:0])	Peripheral module clock (PCLKC)
9980 (ps) (Actual value, 10000)		x1/4 •	60 (MHz)
		SCKCR (PCKD[3:0])	Peripheral module clock (PCLKD)
		x1/4 -	60 (MHz)
Sub-clock		SCKCR (BCK[3:0])	External bus clock (BCLK)
Frequency: 32.768 (kHz)		• x1/4 •	60 (MHz)
Oscillator drive capacity: Standard CL 👻			BCKCR (BCLKDIV) External bus clock pin (BCLK pin)
Oscillation wait time:			• x1/2 • - (MHz)
2000 (ms) (Actual value: 2047.939)			SDRAM clock (SDCLK)
			(MHz)
		SCKCR2 (UCK[3:0])	USB clock (UCLK)
HOCO clock		×1/5 *	48 (MHz)
Enable HOCO oscillation after reset			
Frequency: 16 v (MHz)	L		
Enable FLL function		CKOCR (CKODIV[2:0])	CLKOUT pin
·		×1/8 *	- (MHz)
LOCO dock			
Frequency: 240 (kHz)			CANMCLK/CACMCLK
		-	24 (MHz)
			CACLCLK
			- (kHz)
			CACHCLK
			- (MHz)
IWDT-dedicated clock			IWDTCLK/CACILCLK
Frequency: 120 (kHz)			(kHz)
120 (MT4/			CACSCLK
•			32.768 (kHz)
			REMSCLK
			32.768 (kHz)
	•		VBATCLK 32.768 (kHz)
EXCIN .			
			RTCSCLK
			32.768 (kHz)



(3) Components

Modify the settings of individual components in the "Components" tabbed page according to the specifications of the target board.

(a) Changing I2C Driver Settings

RIIC0 is assigned to Pmod1 and SCI11 to Pmod2 on the EK-RX671 board.

When using Pmod1 (Option Type 6A), set it as follows:

- For r_riic_rx, set "MCU supported channels for CH0" to "Supported".
- For r_riic_rx, add the check settings of "RIIC0", "SCL0 Pin" and "SDA0 Pin" under "Resources".

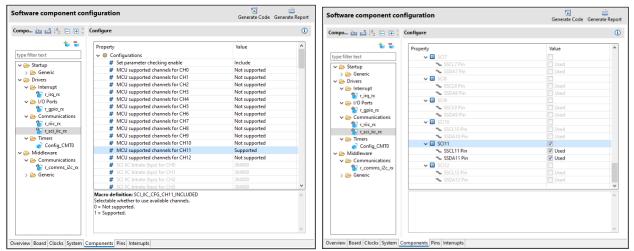
ompo 🚵 🛃 🖓 🕀 🕀 🕻	Configure		(i)	Compo 🚵 🛃 🎘 📄 🕀 🗎	Configure		0
type filter text	Property \$\sigma\$ Configurations \$\# Set parameter checking enable	Value	^	type filter text	Property # Count up during low period of timeout detect # Count up during low period of timeout detect		^
	# CH1 RIIC bps(kbps)	Supported Not supported Not supported 400 400		> @ Generic > @ Drivers > @ Interrupt * r_irq_rx > @ I/O Ports	Count up during high period of timeout detee Count up during high period of timeout detee Count up during high period of timeout detee Set Counter of checking bus busy Conter of checking bus busy Conter of checking bus busy	tior Used	
	# SCL rise time in Standard Mode # SCL fail time in Standard Mode # SCL rise time in Fast Mode # SCL fail time in Fast Mode Plus # SCL fail time in Fast Mode Plus # Digital filter for CHD # Digital filter for CHD	400 1000E-9 300E-9 300E-9 120E-9 120E-9 Two IIC phi Two IIC phi Two IIC phi Include port setting	Sing ices in the second secon	✓		V Used Used Used Used Used Used Used	
			^				^

When using Pmod2 (Option Type 6A), set it as follows:

• For r_sci_iic_rx, set "MCU supported channels for CH5" to "Not supported",

and set "MCU supported channels for CH11" to "Supported".

• For r_sci_iic_rx, add the check settings of "SCI11", "SSCL11 Pin" and "SSDA11 Pin" under "Resources".



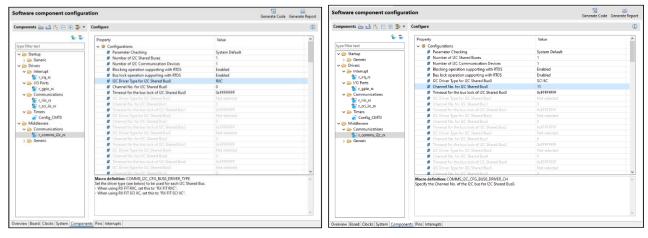


(b) Changing COMMS_I2C Settings

If you have changed the I2C driver or channel, you will need to change these settings.

Set "I2C Driver Type for I2C Shared BusX" (X: Bus No.) and "Channel No. for I2C Shared BusX" (X: Bus No.) in **r_comms_i2c_rx** as follows:

- When using Pmod1 (Option Type 6A), set "I2C Driver Type for I2C Shared Bus0" to "RIIC", set "Channel No. for I2C Shared Bus0" to "0".
- When using Pmod2 (Option Type 6A), set "I2C Driver Type for I2C Shared Bus0" to "SCI_IIC", set "Channel No. for I2C Shared Bus0" to "11".





(c) Changing IRQ Driver Settings

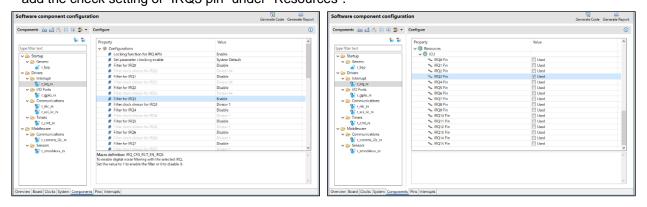
For information on matching the interrupt signal circuit, refer to "6.5 Notes for Interrupt Signal Circuits".

For interrupt settings, the following explains the example of modifying the ZMOD4XXX FIT module (r_zmod4xxx_rx), since the sensor control module uses a common IRQ driver. In the following, treat ZMOD4XXX as RRH46410.

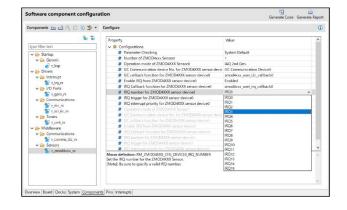
P83 IRQ3 is assigned to Pmod1 #1 and P74 IRQ12 to Pmod2 #1 on the EK-RX671 board.

Therefore, set IRQ channel in "r_irq_rx" as follows and disable any unnecessary settings, also:

- When using Pmod1 (Option Type 6A), set "Filter of IRQ4" to "Disable", set "Filter of IRQ3" to "Enable", remove the check setting of "IRQ4 pin" under "Resources", add the check setting of "IRQ3 pin" under "Resources".
- When using Pmod2 (Option Type 6A), set "Filter of IRQ4" to "Disable", set "Filter of IRQ12" to "Enable", remove the check setting of "IRQ4 pin" under "Resources", add the check setting of "IRQ3 pin" under "Resources".



Change the settings of "IRQ number for ZMOD4XXX sensor device0" to "IRQ3" or "IRQ12" in r_zmod4xxx_rx. For RRH46410, there is also "IRQ number for RRH46410 sensor device0". Change it in the same way.





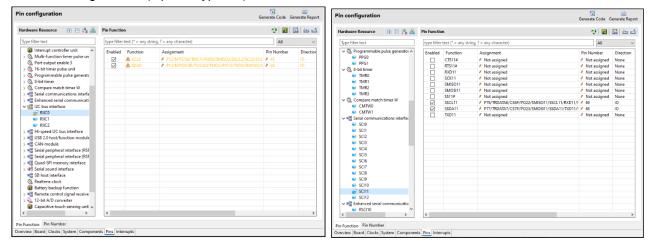
(4) Pins

(a) Changing I2C I/F Pins

RIIC0 is assigned to Pmod1 and SCI11 to Pmod2 on the EK-RX671 board.

Therefore, set the pins used for I2C communication in "Pin Function" on "Pins" tabbed page as follows:

- When using Pmod1 (Option Type 6A): Enable RIIC0, P12 SCL0 and P13 SDA0.
- When using Pmod2 (Option Type 6A): Enable SCI11, P76 SSCL11 and PB7 SSDA11.



As the use of Pmod1 at "High-Speed I2C Bus Interface (RIICHS)" is specified in the EK-RX671 board information, a warning message will appear when RIIC is used, but this does not produce any problems.

(b) Changing IRQ Pin

For information on configuring the pin of the interrupt signal circuit, refer to "6.5 Notes for Interrupt Signal Circuits".

P83 IRQ3 is assigned to Pmod1 #1 and P74 IRQ12 to Pmod2 #1 on the EK-RX671 board.

Therefore, select "Interrupt controller unit" in "Pin Function" on "Pins" tabbed page and set pins as follows:

- When using Pmod1 (Option Type 6A): Enable IRQ3 and set the IRQ3 pin to P83.
- When using Pmod2 (Option Type 6A): Enable IRQ12 and set the IRQ12 pin to P74.

lardwareResource 🛛 🕀 📄 🔩 🟯	Pin Functio	n		्रे 🔳 🖫	i ès s
Type filter text	type filter	text (* = any str	ing, ? = any character)	All	```
All Clock generator Clock frequency accuracy me Operating mode control On-chip emulator On-chip emulator EXDMA controller EXDMA controller EXDMA controller Controlle		Function IRQ0 IRQ1 IRQ2 IRQ3 IRQ4 IRQ5 IRQ6 IRQ7 IRQ8 IRQ9 IRQ10 IRQ10 IRQ11 IRQ12 IRQ13 IRQ14 IRQ15 NMI	Assignment Not assigned	Pin Number Pin Number Not assigned Not assigned	Direction None None None None None None None No



(c) Changing General Purpose I/O Port Pin: RESET

For information on configuring the pin of the RESET signal circuit, refer to "6.6 Notes for RESET Signal Circuits".

The RESET pin is assigned to P82 Port for Pmod1 #2 and P77 Port for Pmod2 #2 on the EK-RX671 board.

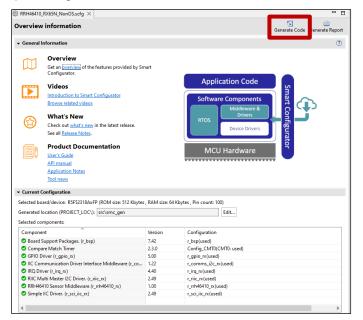
Therefore, select "I/O Ports" in "Pin Function" on "Pins" tabbed page and set pins as follows:

- When using Pmod1 (Option Type 6A): Enable P82.
- When using Pmod2 (Option Type 6A): Enable P77.

lardware Resource 🛛 🕀 🖃 🖧 🖧	Pin Function	n		ર	🔣 🛄 è	
Type filter text	type filter	text (* = any str	ing, ? = any character)	1	All	
P ■ Serial peripheral interface (RSPI) RSPI0 RSPI1 RSPI2 RSPI2 RSPIA0 Realtime clock Battery backup function REMC0 REMC0 SI2AD1 SI2AD0 SI2AD1 Capacitive touch sensing unit Digital power supply Analog power supply		Pro Pro Pr1 Pr2 Pr3 Pr4 Pr5 Pr6 Pr7 P80 P81 P82 P83 P86 P87 P90 P91 P90 P91 P92 P93 PA0	Assignment Not assigned	Pin Number Pin Number Not assigned Not assigned	None None None None None None None None	
Sill I/O Ports		PA1 PA2	 Not assigned Not assigned 	 Not assigned Not assigned 	None None	
< >	<					>

(5) Code Generation and Build

Press the [Generate Code] icon to generate code.



Build the project after implementing "6.3.2 Changing Sample Code".

Select [Debug Configurations] from the menu and modify the debugger settings according to the specifications of the emulator to be connected to the target board.



6.3.2 Changing Sample Code

(1) **RESET Signal Control**

For pin configuration, refer to "6.3.1(4)(c) Changing General Purpose I/O Port Pin: RESET".

Open main.c and change the reset operation when resetting the sensor. Also, change the reset control logic to an appropriate one.

Modify RESET pin designation according to the specifications of the target board.

With the above settings, the EK-RX671 board is configured as follows:

- When using Pmod1 (Option Type 6A), RESET pin is assigned to P82.
- When using Pmod2 (Option Type 6A), RESET pin is assigned to P77.

```
/* Reset RRH46410 sensor (active low). Please change to the IO port
connected to the RES_N pin of the RRH46410 sensor on the customer board. */
R_GPIO_PinWrite(GPIO_PORT_8_PIN_2, GPIO_LEVEL_HIGH);
R_GPIO_PinDirectionSet(GPIO_PORT_8_PIN_2, GPIO_DIRECTION_OUTPUT);
R_BSP_SoftwareDelay(10, BSP_DELAY_MILLISECS);
R_GPIO_PinWrite(GPIO_PORT_8_PIN_2, GPIO_LEVEL_LOW);
R_BSP_SoftwareDelay(10, BSP_DELAY_MILLISECS);
R_GPIO_PinWrite(GPIO_PORT_8_PIN_2, GPIO_LEVEL_LOW);
R_BSP_SoftwareDelay(10, BSP_DELAY_MILLISECS);
R_GPIO_PinWrite(GPIO_PORT_8_PIN_2, GPIO_LEVEL_HIGH);
R_BSP_SoftwareDelay(10, BSP_DELAY_MILLISECS);
```

6.3.3 Changing Toolchain Setting

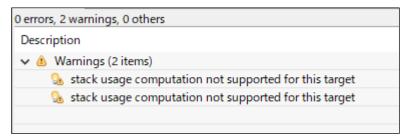
If you want to use a toolchain other than the CC-RX toolchain, copy main.c and RX_RRH46410.c (Non-OS), or main.c and rrh46410_sensor_thread_entry.c (FreeRTOS) from this project to create a new project.

6.3.4 Notes for Build on GCC

The following Warning occurs when building an GCC project.

These Warnings are occurring because specifying a stack size limit in the compiler options, and the target portion (the inline assembler processing portion) is not included in the calculation of stack usage.

Therefore, there is no problem in operation even if warnings occur.



6.3.5 When using IAR Integrated Development Environment "IAR Embedded Workbench"

You can use the RX Smart Configurator to import source files into IAR Embedded Workbench.

For instructions, see below.

RX Smart Configurator User's Guide: IAREW



6.4 RL78 Sample Project

After importing the sample projects, follow the steps below. Please refer to "6.1 Importing Sample Project" for importing instructions.

The following explains the change procedure for the following board change example.

 Sample project "RRH46410_RL78G23_NonOS": Pmod2 (Type 6A: IICA1) → Grove (IICA0) of RL78/G22 Fast Prototyping Board With interrupt control → Without interrupt control RESET control pin → Change to any Port pin

Set J17 to 2-3 to change Grove's VDD to 3.3V.

conne	ect with jur	per wires as	snown below.			
	RL78/G22	2			Renesas	Pmod Type 6A
	Fast Prot	otyping Boa	rd (Grove)		Sensor B	oard
	Pin	Function	IICA I/F, Power Supply		Pin	Function
	#1	SCL	SCLA0		#1	IRQ
	#2	SDA	SDAA0		#2	RESET
	#3	VCC	3.3V		#3	IIC_SCL
	#4	GND	GND		#4	IIC_SDA
					#5	GND
	RL78/G22	2			#6	VCC
	Fast Prot	otyping Boa	rd		#7	BUSY#
	Pin	Function			#8	ENABLE
	Any	RESET	Port	K	#9	POWER_ON

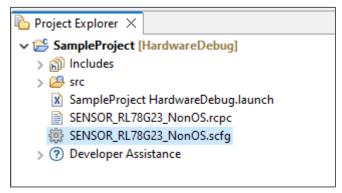
Also, connect with jumper wires as shown below.

6.4.1 Modifying Settings of Smart Configurator

On the project tree, double-click on the .scfg file of the imported project in the Smart Configurator window will open.

#10

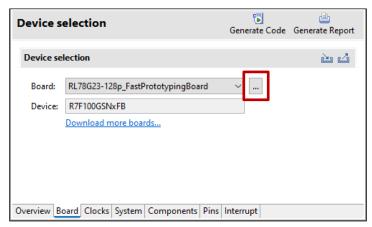
GPIO





(1) Board

1. On the Board tab, click the [...] button.



2. Select a desired board or device in the "Change Device" window and press the [Next] button.

Refactoring						×
Change Devic Select the new	e device for SampleProj	ect				
	R7F100GSNxFB RL78G23-128p_FastPro	ototypingBoard				
Target Board:	RL78G22_FastPrototy	pingBoard				~
	D7E102COEED			<u>Download a</u>	dditional bo	ards
larget Device:	R7F102GGExFB				Unlock De	
					UTILOCK DE	/ices
?		< Back	Next >	Finish	Canc	el



3. If a warning message appears, read it and check if there is a problem in proceeding with the procedure. Press [Next] to move to the next step.

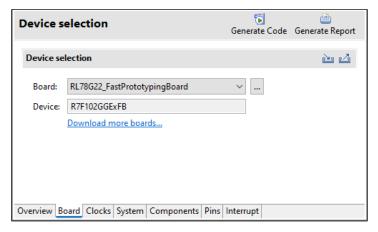
Refactoring			\times
Change Device Review the information provided in the list below. Click 'Next >' to view the next item	n or 'Finish'		
Found problems		ł	3 6
(a) This change cannot be undone. Please make sure you backup this project before c	ontinuing.		
No context information available			
? < Back Next > Finish		Cance	1

4. The changes you have made in the settings will be displayed. Press the [Finish] button to apply the changes to the project.

Refactoring				×
Change Device				-
The following changes to 4 files are necessary to perform the refactoring.			- 50	
Changes to be performed		ł	ን የ	7 -
🗸 🗹 🛃 Change Device for SampleProject				
🗸 🖂 🔁 Launch Configurations				
🖂 🚖 SampleProject HardwareDebug				
> 🗹 🔂 Build Settings				
🔽 🔁 Project Files				
🗹 🚖 Smart Configurator				
No preview available				
		_		
Kext > Fi	nish		Cance	el



5. Select the "Board" tabbed page to check that the "Board" and "Device" have been changed correctly.



(2) Clocks

When "Custom User Board (Any Device)" is selected for "Board", set up the clocks according to the specifications of the target board to be used.

When a Renesas board is selected for "Board", the clocks are automatically set up.

ocks configuration	🕞 Generate Code	Generate Rep
peration mode: High-speed main mode 4.0(V)~5.5(V)		
High-speed on-chip oscillator		
requency: 32 • (MHz)		
HOCO start setting: Normal 👻	fiHP 32	(MHz)
There is setting for starting the high-speed on-chip oscillator at	fMAIN	
he times of release from STOP mode and of transitions to NOOZE mode.)	32	(MHz)
	FCLK	
	32000 fimp ()	(kHz)
Middle-speed on-chip oscillator		(MHz)
requency: 4 (MHz)		
Divider		
X1 oscillator	fMXP 🧯	
Deperation mode: X1 oscillation		(MHz)
requency: 5 (MHz)		
table time: 2^18/fx - 52428.8(µs)		
ow-speed on-chip oscillator	fiL 32.768	(kHz)
requency: 32.768 (kHz)		(
the flL runs while WDT is operating or fSXP select Low-speed	fSXP	
on-chip oscillator	32.768	kHz)
XT1 oscillator	fSXR 32.768	kHz)
peration mode: XT1 oscillation		
32.768 (kHz)		
T1 oscillation mode: Low power consumption 1 🔹		
upply mode: Enables supply in STOP, HALT mode 🔹		
		•



(3) Components

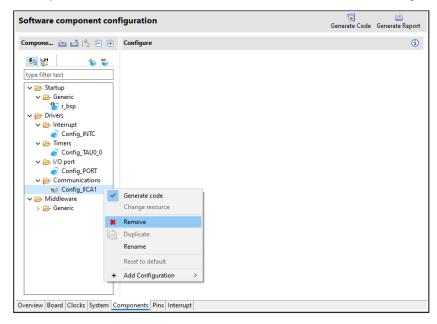
Modify the settings of individual components in the "Components" tabbed page according to the specifications of the target board.

(a) Changing I2C Driver Settings

The SAU cannot be used because it does not support the clock stretching function required by RRH46410 sensor.

To change I2C driver setting, follow the steps below.

1. In RL78/G22, the only resource that can be used as IICA is IICA0, so delete Config_IICA1.



2. In "Software Component Selection", select "IIC Communication (Master Mode)" and specify "IICA0" as the resource.

New Component		_		>	<	New Component		- 0
oftware Component Selection Select component from those available in list						Add new configuration	n for selected component	#
Category All						- IIC Communication (Ma	aster mode)	
						Configuration name:	Config IICA0	
Function All				`	1	-		
Filter						Resource:	IICA0	`
Components	Short Name	Туре	Versio	on ^	•			
H A/D Converter		Code Generator	1.5.0					
🌐 Board Support Packages v1.62	r_bsp	RL78 Software I	1.62					
H Clock Output /Buzzer Output Controller		Code Generator	1.4.1					
# Data Transfer Controller		Code Generator	1.3.1					
🖶 Delay Counter		Code Generator	1.4.1					
Divider Function		Code Generator	1.4.2					
🖶 Event Link Controller		Code Generator	1.2.0					
External Event Counter		Code Generator	1.4.1					
FS1015 Sensor Middleware	r_fs1015	RL78 Software I	х.хх					
FS2012 Sensor Middleware	r_fs2012	RL78 Software I	х.хх					
FS3000 Sensor Middleware	r_fs3000	RL78 Software I	x.xx					
HS300x Sensor Middleware	r_hs300x	RL78 Software I	x.xx					
HS400x Sensor Middleware	r_hs400x	RL78 Software I	x.xx					
IIC Communication (Master mode)		Code Generator	1.5.1					
IIC Communication (Slave mode)		Code Generator	1.4.1					
BIC Communication Driver Interface Mid	r_comms_i2c	RL78 Software I	x.xx					
Input Pulse Interval/Period Measurement		Code Generator	1.4.3					
🖶 Input Signal High-/Low-Level Width Me		Code Generator	1.4.2	~				
Show only latest version					_			
lescription								
This is a clocked communication function (ML using two lines: serial clock (SCL) and serial da compload RL78 Software Integration System m configure general settings	ta (SDA).	ate with two or more	devices	by ^	x			
?) < Ва	:k Next >	Finish	Can	icel		?	< Back Next >	Finish Cancel



3. Change the setting of "Clock mode setting" to "fCLK/2" and uncheck "Generated stop condition in master transmission/reception end callback function".

Software component co	onfiguration		Generate Code Generate Report
Compo 🚵 🛃 🎝 🕀 🕀	Configure		^
type filter text	Clock mode setting Clock mode setting Local address setting Address	fCLK fCLK fCLK/2 16	(Clock frequency: 32000 kHz)
 ✓ German Drivers ✓ German Drivers ✓ Config_INTC ✓ German Drivers 	Operation mode setting Standard Digital filter on	○ Fast mode	○ Fast mode plus
Config_TAU0_0 Config_PORT Config_PORT Config_ICA0 Config_ICA0 Config_ICA0	Transfer clock (fSCL) tR and tF setting Set tR and tF manually tR tF	0 (μs) 0 (μs) 0 (μs)	(Actual value: 99688.474)
	Interrupt setting Communication end interrupt priority (INTIICA0)	Level 3 (low) \lor	
	Callback function setting Master transmission end Callback function enhanced feature setting Callback function enhanced feature setting	Master reception end	Master error
Overview Board Clocks System	Components Pins Interrupt	•	>

(b) Changing COMMS_I2C Settings

П

Review the settings to make sure they are appropriate. If you have changed the I2C driver, you will need to review them.

Change the setting of "Component name for the I2C Shared Bus0" to "Config_IICA0" in r_comms_i2c.

	E Configure		
🐼 👘 😜 😜	Property	Value	
e filter text	✓ [⊕] Configurations		
⇒ Startup	# Parameter Checking	System Default	
🤟 Startup 🖉 🗁 Generic	# Number of I2C Shared Buses	1	
Por senence Por se	# Number of I2C communication Devices	1	
Drivers	# I2C Driver Type for I2C Shared Bus0	IICA	
⊳ Interrupt	# Component name for the I2C Shared Bus0	Config_IICA0	
→ Interrupt → Timers	# Driver Type for I2C Shared Bus1	Not selected	
Config_TAU0_0	# Component name for the I2C Shared Bus1	Config_IIC00	
► I/O port	# Driver Type for I2C Shared Bus2	Not selected	
Config_PORT	# Component name for the I2C Shared Bus2	Config_IIC00	
Communications	# Driver Type for I2C Shared Bus3	Not selected	
Config_IICA0	# Component name for the I2C Shared Bus3	Config_IIC00	
eware	# Driver Type for I2C Shared Bus4	Not selected	
eric	# Component name for the I2C bus4	Config_IIC00	
:omms_i2c	# I2C Shared Bus No. for I2C Communication Device0	I2C bus0	
_comms_ize	# Slave address for I2C Communication Device0	0x00	
	# Callback function for I2C Communication Device0	comms_i2c_user_callback0	
	# I2C Shared Bus No. for I2C Communication Device1	I2C bus0	
	# Slave address for I2C Communication Device1	0x00	
	Macro definition: COMMS I2C CFG BUS0 COMPONENT	an anna 10 a cuinn an Illean Ist	



(c) Changing INTC Driver Settings

For information on matching the interrupt signal circuit, refer to "6.5 Notes for Interrupt Signal Circuits".

There is no interrupt pin assigned to Grove on the RL78/G22 Fast Prototyping Board. Please use another pin if necessary.

For interrupt settings, the following explains the example of modifying the ZMOD4XXX SIS module (r_zmod4xxx_rl), since the sensor control module uses a common INTC driver.

1. Set INTP pin in Config_INTC according to the specifications of the target board.

Software component configura	tion	Generate Code Generate Report
Components 🚵 🖄 🖓 🕒 🕀	Configure	(i) ^
Type filter text	INTPO setting INTPO Valid edge Falling edge Valid valid edge Vali	~
 ✓ ➢ Startup ✓ ➢ Generic ♀ ♀ r_bsp 	□INTP1 setting □INTP1 Valid edge Falling edge ∨ Priority Level 3 (low)	~
✓ ➢ Drivers ✓ ➢ Interrupt ✓ Config_INTC	INTP2 setting INTP2 Valid edge Falling edge Valid edge I falling edge I falli	~
✓ (⇒ Timers	□INTP3 setting □INTP3 Valid edge Falling edge ∨ Priority Level 3 (low)	Y .
Config_PONI Communications	INTP4 setting INTP4 Valid edge Falling edge Valid edge I Valid edge Verify Level 3 (low)	Y .
> 🦢 Generic	INTPS valid edge Falling edge v Priority Level 3 (low)	~
	INTP6 setting INTP6 Valid edge Falling edge Valid edge I falling edge I falling edge I falling edge Valid edge I falling edge I falling edge I falling edg	×
	INTP8 setting INTP8 Valid edge Falling edge Valid edge I falling edge I falling edge I falling edge Valid edge I falling edge I falling edge I falling edg	×
Overview Board Clocks System Componen	INTP9 setting INTD0 Mild adap Estime adap or Dripping Laurit Maut s Pins Interrupt	~

2. Change "INTC number for ZMOD4XXX sensor device0" in r_zmod4xxx_rl to the INTP pin set above.

omponents 🚵 🛃 🎘 🗎 🕀	Configure	
Statup ✓ ➢ Startup ✓ ➢ Generic ৺ 尔 bsp	Property	Value System Default 1 IAQ 1st Gen. (Continuous)
✓	I2C Communication device No. for ZMOD4XXX sensor device0 I2C callback function for ZMOD4XXX sensor device0 Enable INTC from ZMOD4XXX sensor device0 INTC Callback function for ZMOD4XXX sensor device0	I2C Communication Device0 zmod4xxx_user_i2c_callback0 Enabled zmod4xxx user irg_callback0
 ✓ Emers Config_TAU0_0 ✓ Ev Do port ✓ Config_PORT ✓ Config_IICA0 ✓ Config_IICA0 ✓ Middleware ✓ (appendix comms_j2c) T_zmod4xxx 	INTC number for ZMOD4XXX sensor device0 Operation mode of ZMOD4XXX sensor device0 IZC Communication device No. for ZMOD4XXX sensor device1 IZC callback function for ZMOD4XXX sensor device1 INTC from ZMOD4XXX sensor device1 INTC Callback function for ZMOD4XXX sensor device1 IRQ number for ZMOD4XXX sensor device1	INTP0 INTP0 INTP1 INTP2 INTP3 INTP4 INTP5 INTP6 INTP6 INTP7 INTP8 INTP9 INTP9 INTP10 INTP11
	Macro definition: RM_ZMOD4XXX_CFG_DEVICE0_IRQ_NUMBER Set the INTC number for the ZMOD4XXX Sensor. [Note]: Be sure to specify a valid INTC number.	INTP12 INTP13 INTP14 INTP15



(d) Changing General Purpose I/O Port Driver Settings: RESET

For information on how to modify the source of the RESET signal control, refer to "6.4.3(1) RESET Signal Control".

There is no RESET pin assigned to Grove on the RL78/G22 Fast Prototyping Board. Therefore, assign any port.

Set RESET pin on Pmod in Config_PORT according to the specifications of the target board.

Set "Out" and enable "Output 1" in the "Port selection" tabbed page and the "PORT(x)" tabbed page.

Software component configu	Iration	🐻 Generate Code	📄 Generate Report
Components 🚵 🛃 🖓 🕀 🕀	Configure		^
Image: Startup Image:	Port selection PORT0 PORT1 PORT2 PORT3 PORT4 PORT5 PORT6 PORT7 PORT12 PORT13 PORT14 PORT14		
Overview Board Clocks System Compo	nents Pins Interrupt		

	OFF" is eff re not usin				r an alternative function, "Input buffer OFF".	or the pin is n	10t used. Please make
"Input buffer peripherals an Apply to al Unused	OFF" is eff re not usin II	g the alternat	tive input function	on before selecting	"Input buffer OFF".	or the pin is n	10t used. Please make
eripherals an Apply to al	re not usin II	g the alternat	tive input function	on before selecting	"Input buffer OFF".	or the pin is n	not used. Please make
	⊖In	Out	Pull-up	TI buffer			
P10					Input buffer OFF	N-ch	Output 1
Unused	() In	Out	Pull-up	TTL buffer	Input buffer OFF	N-ch	Output 1
P11 Unused	() In	Out	Pull-up	TTL buffer	Input buffer OFF	N-ch	Output 1
P12 Unused	() In	Out	Pull-up		Input buffer OFF	N-ch	Output 1
P13 O Unused	() In	Out	Pull-up	TTL buffer	Input buffer OFF	N-ch	Output 1
P14 Unused	() In	Out	Pull-up	TTL buffer	Input buffer OFF	N-ch	Output 1
Dir							>
	 Unused P11 Unused P12 Unused P13 Unused P14 Unused 	Image: Unused Image: Imag	 ● Unused ○ In ○ Out P11 ● Unused ○ In ○ Out P12 ● Unused ○ In ○ Out P13 ○ Unused ○ In ● Out P14 ● Unused ○ In ○ Out 	 ● Unused ○ In ○ Out □ Pull-up P11 ● Unused ○ In ○ Out □ Pull-up P12 ● Unused ○ In ○ Out □ Pull-up P13 ○ Unused ○ In ● Out □ Pull-up P14 ● Unused ○ In ○ Out □ Pull-up P14 ● Unused ○ In ○ Out □ Pull-up 	Unused In Out Pull-up TTL buffer P11 Unused In Out Pull-up P12 Unused In Out Pull-up TTL buffer P13 Out Out Pull-up TTL buffer P14 Oln Out Pull-up TTL buffer P14 Out Out Pull-up TTL buffer P14	Image: Unused Image: Out Pull-up TTL buffer Input buffer OFF P11 Image: Out Pull-up TTL buffer Input buffer OFF P12 Image: Out Pull-up Image: Out Image: Out Image: Out P13 Image: Out Image: Out Pull-up Image: Out Image: Out Image: Out P14 Image: Out Image: Out Pull-up Image: Out Image: Out Image: Out P14 Image: Out Image: Out Image: Out Image: Out Image: Out Image: Out P14 Image: Out Image: Out Image: Out Image: Out Image: Out Image: Out Image: Out Image: Out Image: Out Image: Out Image: Out Image: Out Image: Out Image: Out Image:	Image: Unused Image: Out Pull-up TTL buffer Input buffer OFF N-ch P11 Image: Out Pull-up TTL buffer Input buffer OFF N-ch P12 Image: Out Out Pull-up Image: Out Im



- (4) Pins
- (a) Changing I2C I/F Pins
- (b) Changing INTP Pins
- (c) Changing General Purpose I/O Port Pin: RESET

For information on configuring the pin of the interrupt signal circuit, refer to "6.5 Notes for Interrupt Signal Circuits".

For information on configuring the pin of the RESET signal circuit, refer to "6.6 Notes for RESET Signal Circuits".

Select "IICA0", "Interrupt Function" and "I/O Ports" in "Pins" tabbed page and check that functions are assigned to the IICA pins, INTP pin, and RESET pin in the "Pin Function" panel.

							de Generate Rep
lardware Resource 🛛 🕀 🖻 📲 🖧	Pin Function	n				3	🖬 🔛 🔛 e
Type filter text	type filter	text (* = any stri	ng, ? = any chara	icter)		All	- · · ·
PCLBUZ1 ^	Enabled	Function	PIOR	Assignment		Pi	n Number
A/D Converter		SCLA0	PIOR2	/ P60/SCLA0		1	
✓		SDAA0	PIOR2	P61/SDAA0		/	
SAU0							
SAU01							
SAU02							
SAU03							
SAU1							
 SAU11 							
v n∰ Serial Interface IICA							
S IICA0							
v n Serial Interface UARTA							
Interrupt Function							
Key Interrupt							
Reset Function							
Capacitive Sensing Unit							
On-Chip Debug Power Supply							
Voltage Regulator							
< ×	1						
Function Pin Number							
erview Board Clocks System Components	Pins Interrupt						
n configuration						Generate Cor	de Generate Rep
						Generate cov	se Generate neg
ardware Resource 🛛 🕀 📄 🖧 💑	Pin Function	n				3	🖬 🔛 🔤 i
Type filter text	type filter	text (* = any stri	ng, ? = any chara	icter)		All	
PCLBUZ1	Enabled						
A/D Converter		Function INTP0	PIOR	Assignment	Pin Number	Direction	Remarks
✓ P ^B Serial Array Unit		INTP0 INTP1		 Not assigned Not assigned 	 Not assigned Not assigned 	None	
✓ SAU0		INTP2		Not assigned	Not assigned	None	
SAU00 SAU01		INTP3		Not assigned	Not assigned	None	
SAU01		INTP4		Not assigned	Not assigned	None	
SAU03		INTP5 INTP6		Not assigned	Not assigned	None	
✓ SAU1		INTP6		 Not assigned Not assigned 	 Not assigned Not assigned 	None	
SAU10		INTP9		 Not assigned 	Not assigned	None	
SAU11 ✓ № Serial Interface IICA							
✓ sqiji Serial interface liCA ✓ IICA0							
✓ P Serial Interface UARTA							
UARTA0							
Interrupt Function							
Key Interrupt Reset Function							
Capacitive Sensing Unit							
On-Chip Debug							
Power Supply							
📥 Voltage Regulator 🗸 🗸							
< >	<						
Function Pin Number							
erview Board Clocks System Components	Pins Interrupt						
	and an arrested						
n configuration						1	<u></u>
in configuration						Generate Coo	le Generate Rep
lardware Resource 🛛 🕀 🖃 🖧	Pin Functio	n					8 🔛 🗠 🛛
Type filter text	1		ing, ? = any chara			All	
All All	Enabled	Function	PIOR	Assignment	Pin Number	Direction	Remarks 4
		P00		Not assigned	Not assigned	None	
Clock Generator		P01		Not assigned	Not assigned	None	
 Clock Generator Cliner Array Unit 		P10 P11		 Not assigned Not assigned 	 Not assigned Not assigned 	None	
✓ [™] / ₄ Timer Array Unit ✓ TAU0		P12		Not assigned	 Not assigned Not assigned 	None	
 ✓ Ži Timer Array Unit ✓ TAU0 ✓ TAU00 				Not assigned	Not assigned	None	
 ✓ Ži Timer Array Unit ✓ TAU0 ✓ TAU00 ✓ TAU01 		P13			Not assigned	None	
 ✓ (0, Timer Array Unit ✓ TAU0 ☑ TAU00 ☑ TAU01 ☑ TAU02 		P14		Not assigned			
 ✓ (0, Timer Array Unit) ✓ TAU0 ☑ TAU00 ☑ TAU01 ☑ TAU01 ☑ TAU02 		P14 P15		Not assigned	Not assigned	None	
 ✓ (³/₂, Timer Array Unit ✓ TAU0 ✓ TAU00 ✓ TAU01 ✓ TAU02 ✓ TAU03 ✓ TAU03 ✓ TAU04 ✓ TAU04 		P14 P15 P16		 Not assigned Not assigned 	 Not assigned Not assigned 	None	
 ✓ (2), Timer Array Unit TAU00 ✓ TAU00 ✓ TAU01 ✓ TAU02 ✓ TAU03 ✓ TAU04 ✓ TAU04 ✓ TAU05 ✓ TAU05 		P14 P15		 Not assigned Not assigned Not assigned 	 Not assigned Not assigned Not assigned 	None None	_
✓ 03 Timer Array Unit ✓ TAU0 ✓ TAU00 ✓ TAU00 ✓ TAU00 ✓ TAU00 ✓ TAU02 ✓ TAU02 ✓ TAU04 ✓ TAU04 ✓ TAU05 ✓ TAU05 ✓ TAU05 ✓ TAU05		P14 P15 P16 P17 P20 P21		 Not assigned Not assigned 	 Not assigned Not assigned 	None	
 ✓ Ø Timer Array Unit ✓ TAU00 ➡ TAU01 ➡ TAU01 ➡ TAU01 ➡ TAU03 ➡ TAU04 ➡ TAU04 ➡ TAU04 ➡ TAU05 ➡ TAU05 ➡ TAU06 ➡ TAU06 ➡ TAU06 ➡ TAU06 		P14 P15 P16 P17 P20 P21 P22		 Not assigned 	 Not assigned 	None None None None	
✓ 03 Timer Array Unit ✓ TAU0 ✓ TAU00 ✓ TAU00 ✓ TAU00 ✓ TAU00 ✓ TAU02 ✓ TAU02 ✓ TAU04 ✓ TAU04 ✓ TAU05 ✓ TAU05 ✓ TAU05 ✓ TAU05		P14 P15 P16 P17 P20 P21 P22 P23		 Not assigned 	 Not assigned 	None None None None None	
√20: Timer Array Unit √10: √21: Timer Array Unit √10: √21: Timer Array Unit √21: TAU00 √21: TAU000 √21: TAU0000 √21: TAU0000 √21: TAU0000 √21: TAU0000 √21: TAU00		P14 P15 P16 P17 P20 P21 P22 P23 P24		 Not assigned 	 Not assigned 	None None None None None None	
		P14 P15 P16 P17 P20 P21 P22 P23		Not assigned	 Not assigned 	None None None None None None None	
		P14 P15 P16 P17 P20 P21 P22 P22 P23 P24 P25		Not assigned	 Not assigned 	None None None None None None	
		P14 P15 P16 P17 P20 P21 P22 P23 P24 P25 P26 P26 P27 P30		 Not assigned 	 / Not assigned 	None None None None None None None None	
		P14 P15 P16 P17 P20 P21 P22 P23 P24 P24 P25 P26 P27 P30 P31		Not assigned	 / Not assigned 	None None None None None None None None	
		P14 P15 P16 P17 P20 P21 P22 P23 P24 P25 P26 P26 P27 P30		 Not assigned 	 / Not assigned 	None None None None None None None None	



(5) Code Generation and Build

Press the [Generate Code] icon to generate code.

erview information		1
		Generate Code enerate Repo
General Information		C
Overview		
Get an overview of the features provided by Sn	hart	
Configurator.		
		Application Code
Videos		ST
Introduction to Smart Configurator		Software Components
Browse related videos		Middleware & 5
What's New		Drivers
Check out what's new in the latest release.		RTOS
See all Release Notes.		Device Drivers
See on <u>Handwell Hora</u>		a a
Product Documentation		MCU Hardware
User's Guide		
API manual		
Application Notes		
Tool news		
Current Configuration		
Current Configuration		
		Kbytes Pin count: 100)
elected board/device: R5F52318AxFP (ROM size: 512 Kbytes	, KAIVI SIZE: 04	
elected board/device: R5F52318AxFP (ROM size: 512 Kbytes enerated location (PROJECT_LOC\): src\smc_gen	, KAIVI SIZE: 04	Edit
enerated location (PROJECT_LOC\): src\smc_gen	, KAIVI SIZE: 04	
enerated location (PROJECT_LOC\): src\smc_gen	Version	
enerated location (PROJECT_LOC\): src\smc_gen elected components:		Edit
enerated location (PROJECT_LOC\): src\smc_gen elected components: Component Deard Support Packages. (r_bsp)	Version	Configuration
enerated location (PROJECT_LOC\): src\smc_gen	Version 7.42	Edit Configuration r_bsp(used)
enerated location (PROJECT_LOC\): src\smc_gen lected components: Component Ø Board Support Packages. (r_bsp) Ø Compare Match Timer	Version 7.42 2.3.0	Edit Configuration r.bsp(used) Config_CMT0(CMT0: used)
enerated location (PROJECT_LOC\): src\smc_gen slected components: Component 9 Board Support Packages. (r_bsp) 9 Compare Match Timer 9 GPO Driver (r gaio, n)	Version 7.42 2.3.0 5.00	Edit Configuration r_bsp(used) Config_CMT0(CMT0: used) C_gpio_rX(used)
enerated location (PROJECT_LOC): [src/smc_gen leteted components: Component 9 Gond Support Packages, (r_btp) 9 Compare Match Timer 9 GPD Driver (r_ggin, n) 9 IIC Communication Driver Interface Middleware (r_co 9 IRQ Driver (r_ing.n) 9 IRC Muth Mater (2 Driver, (r_mic_m)	Version 7.42 2.3.0 5.00 1.22 4.40 2.49	Edit Configuration r_brip(used) Config_CMT0(CMT0: used) r_gpio_rx(used) r_comme_l2c_rx(used) r_rir_rx(used)
enerated location (PROJECT_LOC): src/smc_gen sleated components: Component © Board Support Packages. (r_bsp) © Compare Match Timer © GPID Driver (r.gsio, n) © IC Communication Driver Interface Middleware (r_co © RD Driver (r.g.n, n)	Version 7.42 2.3.0 5.00 1.22 4.40	Edt Configuration r_bsp(used) Config_CMT0(CMT0: used) r_comms_i2c_r(used) r_comms_i2c_r(used) r_ir_cn(used)

Build the project after implementing "6.4.2 Modifying Generated Code" and "6.4.3 Changing Sample Code"

Select [Debug Configurations] from the menu and modify the debugger settings according to the specifications of the emulator to be connected to the target board.



6.4.2 Modifying Generated Code

Open Config_IICA0_user.c and add the following code.

Definition for including r_comms_i2c_if.h:

Addition of the rm_comms_i2c_bus0_callback() function to the callback function:

Specify the "false" parameter for the transmission and reception end callback functions and the "true" parameter for the error callback function.

```
* Function Name: r Config IICA0 callback master sendend
* Description : This function is a callback function when IICAO finishes master
transmission.
* Arguments
           : None
* Return Value : None
static void r Config IICA0 callback master sendend(void)
/* Start user code for r_Config_IICA0_callback_master_sendend. Do not edit comment
generated here */
 rm comms i2c bus0 callback(false);
/* End user code. Do not edit comment generated here */
* Function Name: r Config IICAO callback master receiveend
* Description : This function is a callback function when IICAO finishes master
reception.
* Arguments
          : None
* Return Value : None
static void r_Config_IICA0_callback_master_receiveend(void)
/* Start user code for r Config IICAO callback master receiveend. Do not edit comment
generated here */
 rm comms i2c bus0 callback(false);
/* End user code. Do not edit comment generated here */
}
* Function Name: r Config IICAO callback master error
* Description : This function is a callback function when IICAO master error occurs.
* Arguments : flag -
               status flag
* Return Value : None
static void r_Config_IICA0_callback_master_error(MD_STATUS flag)
   /* Start user code for r Config IICAO callback master error. Do not edit comment
generated here */
 rm comms i2c bus0 callback(true);
   /* End user code. Do not edit comment generated here */
```



6.4.3 Changing Sample Code

(1) **RESET Signal Control**

For pin configuration, refer to "6.4.1(4)(c) Changing General Purpose I/O Port Pin: RESET".

There is no RESET pin assigned to Grove on the RL78/G22 Fast Prototyping Board. Therefore, assign any port.

Open RRH46410_RL78G23_NonOS.c and change the reset operation when resetting the sensor.

Modify RESET pin designation according to the specifications of the target board. Also, change the reset control logic to an appropriate one.

The following is an example of using P13 as RESET pin.



6.4.4 Changing Toolchain Setting

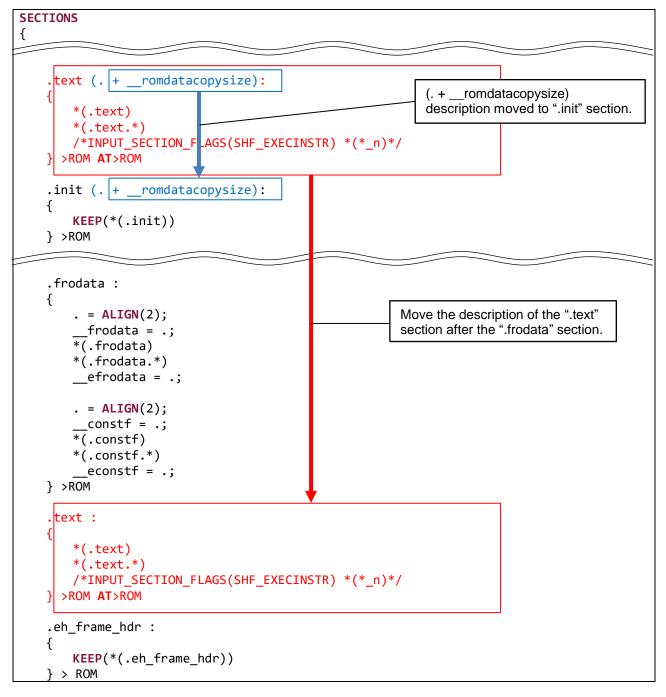
If you want to use a toolchain the LLVM toolchain, use "RRH46410_RL78G23_NonOS_LLVM".

If you want to use a toolchain other than the CC-RL toolchain or LLVM toolchain, copy RRH46410_RL78G23_NonOS.c and RL78_RRH46410.c from this project to create a new project.

Also, when using the LLVM toolchain, build errors may occur due to section placement. In this case, the linker script must be modified.

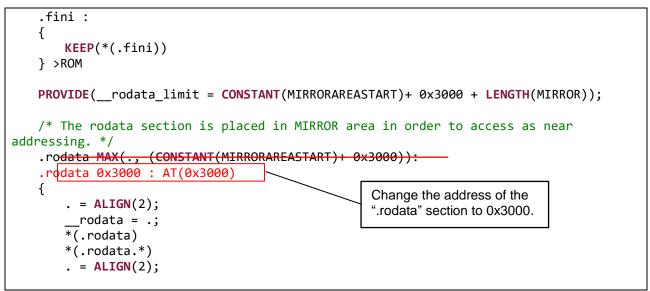
The following describes an example of modifying linker_script.ld in the sample project "RRH46410_RL78G23_NonOS_LLVM".

1. Place the ".text" section after the ".frodata" section.

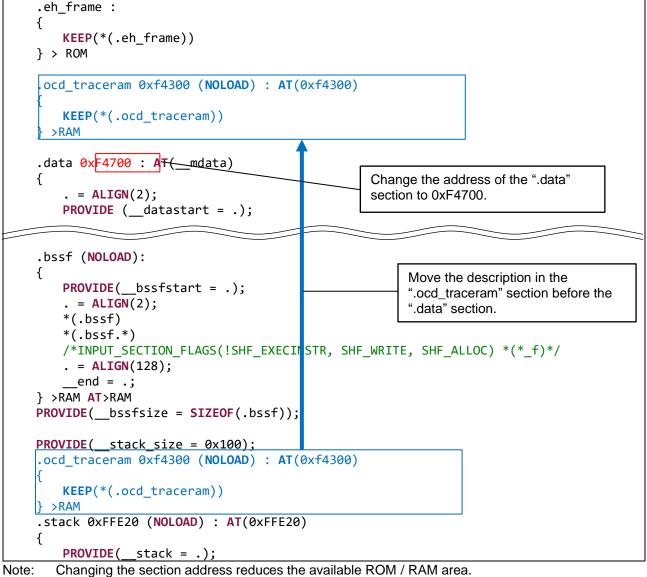




2. Fix the address of the ".rodata" section to the top address of the mirror area.



3. Change the ".data" section to the address after the ".ocd_traceram" section.





6.4.5 Notes for Build on LLVM

The following Warning occurs when building an LLVM project.

These Warnings are occurring because "slave_address" and "bytes" used in COMMS_I2C are handled as 32-bit type.

Since "slave_address" is 7-bit data and "bytes" is 16-bit data, no loss occurs due to conversion.

Therefore, warning messages will appear, but this does not affect the operation.

Problems X	78-0
0 errors, 5 warnings, 0 others	
Description	Resource
Vanings (5 items)	
9 implicit conversion loses integer precision: 'const uint32_t' (aka 'const unsigned long') to 'uint16_t' (aka 'unsigned short') [-Wimplicit-int-conversion]	rm_comms_i2c_drive
9 implicit conversion loses integer precision: 'const uint32_t' (aka 'const unsigned long') to 'uint16_t' (aka 'unsigned short') [-Wimplicit-int-conversion]	rm_comms_i2c_drive
😘 implicit conversion loses integer precision: 'uint32_t' (aka 'unsigned long') to 'uint8_t' (aka 'unsigned char') [-Wimplicit-int-conversion]	rm_comms_i2c_drive
😘 implicit conversion loses integer precision: 'uint32_t' (aka 'unsigned long') to 'uint8_t' (aka 'unsigned char') [-Wimplicit-int-conversion]	rm_comms_i2c_drive
😘 implicit conversion loses integer precision: 'uint32_t' (aka 'unsigned long') to 'uint8_t' (aka 'unsigned char') [-Wimplicit-int-conversion]	rm_comms_i2c_drive
٢	>



6.5 Notes for Interrupt Signal Circuits

It is necessary to configure settings according to the circuit to be embedded.

When using the sensor interrupt request signal (SensorINT), set the trigger level correctly.

(1) Circuit Configuration of RRH46410 Sensor Pmod Board

The circuit configuration of the interrupt signal of the RRH46410 Sensor Pmod Board is shown below. For details, please refer to the board datasheet.

 Table 6-2 Circuit Configuration of Interrupt Signal on RRH46410 Sensor Pmod Board

Sensor Pmod Board	Circuit Configuration (Pull-up resistor can be set to on or off)
#1: IRQ# L Output: Data Available	Inverted output of SensorINT (Open-Drain output) and fixedly connected to IRQ#. (Output of the inverted SensorINT signal cannot be disabled.)
#7: BUSY#	- (No SensorINT signal output circuit)

(2) How to Control Interrupt Trigger Control when Multiple Sensor Pmod Boards are Daisy-chained

If the IRQ# of each board is an open-drain output, daisy chain connection is possible. (Note)

With this connection, if an interrupt request is asserted, it will be difficult to identify the I2C device that generated the interrupt request.

If multiple interrupt output signals are connected to IRQ# of Pmod #1 due to the circuit configuration, operate all I2C devices without interrupts. (Note)

Note: Do not connect I2C devices with outputs other than Open-Drain (Push-Pull output of CMOS or TTL). If connected, the I2C device and the peripheral circuit ICs connected to the interrupt signal may be damaged.

(3) How to Control Interrupt Triggers when using a Sensor Pmod Board that Allows Changing an Interrupt Request Output Destination

Some sensor Pmod boards can output an interrupt request signal to Pmod #7 (BUSY#) by changing the circuit on the board.

By daisy-chaining a board that switches this interrupt request signal to BUSY# output and a board with a standard IRQ# output, it becomes possible to trigger interrupts from two devices at the same time.

Note that an operating condition is that Pmod #1 and #7 on the MCU side have to input interrupt signals.

Table 6-3 Circuit Configuration of Interrupt Signal on Sensor Pmod Board (Board Dependent)

Sensor Pmod Board	Circuit Configuration (Pull-up resistor can be set to on or off)
#1: IRQ#	The Sensor Pmod Board allows the following circuit modifications (Board dependent):
#7: BUSY#	 SensorINT non-inverted or inverted output (open-drain output or push-pull output) is possible for either #1 or #7. (Note 1)
	SensorINT Output is disabled and open for both #1 and #7.

Note 1: Non-inverted or inverted output, Open-Drain or Push-Pull output are board-dependent.



6.6 Notes for RESET Signal Circuits

It is necessary to configure settings according to the circuit to be embedded.

(1) Circuit Configuration of RRH46410 Sensor Pmod Board

The circuit configuration of the RESET signal of the RRH46410 Sensor Pmod Board is shown below. For details, please refer to the board datasheet.

Table 6-4 Circuit Configuration of RESET Signal on RRH46410 Sensor Pmod Board

Sensor Pmod Board	Circuit Configuration (Pull-up resistor can be set to on or off)
#2: RESET#	Directly connected to the RESET signal of the sensor device
L Input: Device Reset	

(2) When You Want to Daisy-chain Multiple I2C Devices and Operate them using Interrupt Triggers

A single RESET output signal from the MCU can control the reset of multiple I2C devices.

In that case, it is necessary to configure a RESET signal circuit that matches the reset control logic of the multiple I2C devices and to implement a reset sequence that meets the requirements of the multiple I2C devices.

In addition, when connecting multiple I2C devices, the pin load capacitance will be large, so be sure to generate a sufficient reset pulse period.

6.7 Pull-up Resister Circuit Configuration when Daisy Chain Connections of Renesas Sensor Pmod Boards

The recommended method for connecting the pull-up resistors in a daisy chain is shown below. Also, disable the pull-ups on other Renesas Sensor boards.

If the pull-up resistors of many Renesas Sensor boards are enabled at the same time, the sensor boards may not function properly.

Pmod Sensor Board	Recommended Circuit Configuration of Pull-up Resisters
Type 6A Singal Name	
#1: IRQ# (Note 1)	Enable only the board closest to the MCU board for boards with pull-up resistor circuits.
#2: RESET# (Note 1)	Enable only the board closest to the MCU board for boards with pull-up resistor circuits.
#3: SCL	Enable only the board closest to the MCU board for boards.
#4: SDA	Enable only the board closest to the MCU board for boards.
#7: BUSY# (Note 1, 2)	Enable only the board closest to the MCU board for boards with pull-up resistor circuits.

Table 6-5 Target Board that Enable Pull-up Resistors when Daisy-chaining

Note 1 There are the boards without the pull-up resisters.

Note 2 Configure when using as an Interrupt signal.



7. Viewing Gas Data

To check the real-time gas data, follow the steps below.

Take the "IAQ 2nd Generation" as an example.

1. After running the Debug, open the "Expressions" window.

"Expressions" window is available from [Window]→[Show View]→[Expressions].

e2_studio - C:\Project\SensorSW_2024\RA-RRH46410_SampleProject\src\RRH46410_R							- 0	a x
File Edit Source Refactor Navigate Search Project Renesas Views Run R						0	12 10 C/C++	the Debug
☆Debug X □			(the Maria De De	Deni	AC Free		물 Peri [] 10	
KP Debug ∧ Provide the set of the	103 Appearance > tup for g rrh46418 sensor0, */	-	vano br	ea Proj			27 Pen. 10.	
 RRH46410_RA2E1_NonOS.elf [1] [cores: 0] 	184				Value	Address	A X L C	
Thread #1 1 (single core) [core: 0] (Suspended : Step)	Show View ** Breakpoints Alt-Shift+Q, B 186 Perspective • </td <td></td> <td>Expression 1</td> <td>lype</td> <td>value</td> <td>Address</td> <td></td> <td><u> </u></td>		Expression 1	lype	value	Address		<u> </u>
g_rrh46410_sensor0_quick_setup() at RA_RRH46410.c:115 0x4f6 hal_entry() at hal_entry.c:35 0x7ee	187		- Augu nev					
main() at main.c5 0x7te	199 Bebugger Console S MATT:							
📓 arm-none-eabi-gdb (12.1)	110 Preferences 111 UNDOWNOP g5_1rg Debug Sources 5_MAIT;							
🔎 Renesas GDB server (Host)	112 #endif 🔤 Disassembly							
	113 114 /* Ope 💇 ErrorLog Alt+Shift+Q,L be done before calling any RRH46418 API */							
	15 000004f6 err = D Executables trl, &g rrh46410 sensor0 cfg);							
	116 000004fa ⊖ 1f (FS 02 Expressions 117 (
	18 0000051c de D Memory							
	120							
	121 /* Ope ■ Modules 122 rrh464 9= Outline Alt-Shift=Q.O							
	125 /* Duick s m Premer							
	126 estatic voi Project Explorer							
	128 fsp_er 101 Registers							
	129 130 /* Ope 54 Signals Fore calling any timer API */							
	131 000004fc err = 3 Smart Browser lay, o ctrl, rrh46410 delay, o cfg):							
	132 00000516 e if (FS) Smart Manual							
	134 00000522 de Templates							
	136 } Venables Alt+shm+U, V							
	137 138 @static voi Other Alt+Shift+Q.Q rrh46410 delay units t units)							
	139 {							
	140 fsp_err_t err; 141							
	142 /* Convert to units of RRH46410 DELAY PERIOD */ 143 000006c8 gs rrh46410 delay count = (delay * gs rrh46410 delay time[units]) / RRH46410 DELAY PERIOD;							
	144							
	145 /* Stop timer */ 146 00000664 err = rn46410 delav.p api->stop(rn46410 delav.p ctrl):	~						
							<	>
	🛛 Console 🔀 🔢 Registers 👔 Problems 🌒 Smart Browser 🕞 Debugger Console 🔋 Memory				X % B	51 B (P	8 - 0 - 1	9 0
	0H46410_RA2E1_NonOS Debug_Flat (Renesas GDB Hardware Debugging) (pid: 18)							
	ardware breakpoint set at address 0x7f8 ardware breakpoint set at address 0x7f8							^
	ardware breakpoint set at address 0x78a ardware breakpoint set at address 0x79c							
	ardware breakpoint set at address 0x7a6							
	ardware breakpoint set at address 0x7b8 ardware breakpoint set at address 0x7c2							
	ardware breakpoint set at address 0x7d4							
	ardware breakpoint set at address 0x7de ardware breakpoint set at address 0x7ea							
								~
< >>								>
Suspended 🧼 0x000004f6 🧭 1002.0 ms	System Time				1			

2. Click "Add new expression" in the "Expressions" and add "gs_iaq_2nd_gen_data".

• Debag × • Obbag	File File Source Refector Navinate Search Droject Renerative Due Deserve	E I_WONUS/SIC/R	RA_RRH46410.	I.c - e ² stu	udio										- 0	×
• Debag × • Obbag	The Post Source meracros Maxidate believe Erolect venezaz News Why years	iesas Al <u>W</u> indo	ow Help													
v 2 Beekeld (A21 Hoods Detes full Reveases OB betweeke Detes gang) v 2 Beekeld (A21 Hoods Cell (Core 5) v	📓 🕲 = 🍕 = 🖉 🖓 = 📾 📾 📾 🗶 🧐 🖉 = 🚱 = 🕼	0a • 🍇 🕪	* 000 1/2 😵	36	1									9	😰 🗟 C/C++ 📑	Debug
> ■ Sh4450 (3x00) 22 2000 St0000CC 7, 2000 St000CC 7,	🍄 Debug 🗙 📄 🖻 🗮 📄 🗮 🗎 📍	startup.c	main.c	1	bsp_delay.c	c 💡	hal_entry.c	RA RA	RRH46410.c ×		- 0	(1) Vari 🗣 Brea 🔂 P	roj 😚 Expr >	< 📲 Eve 7	Peri 10	- 0
<pre></pre>	Debug X Second Secon	Construction Construction 21 22 22 23 23 23 24 24 25 26 27 29 20 31 31 34 35 36 41 42 42 44 40 42 42 35 53 54 54 55 55 54 56 57 57 57 58 64 61 62 62 63 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 <	maince typeped () de () de () de () () de () () de () () () de () () () () () () () () () (DEND S. DEND S. DEND S. DEND S. DEND S. DEND S. DEND S. DEND S. DEND C. DEND C	bap.doty.co bap.doty.co scores.co sc	<pre>callb :rarus_ :</pre>	ack_status SUCCES() SUCCES() SUCCES() Relay_units action of the success action of the	/* R	equested delay anount i equested delay anount i oid); iny, rrhe6418_delay_uni gs_izc_callback_ste gs_irq_callback_ste gs_irq_callback_ste gs_preaded_preaded_preaded grant on procept gs_preaded_preaded grant on procept gs_preaded_preaded_preaded grant on procept gs_preaded_p	s in milliseconds */ s in seconds */ ts_t units); tus = DEMD_CALLBACK_STATUS_WAIT tus = DEMD_CALLBACK_STATUS_WAIT SHO_SEMD_ULP)) ; ; ; ;		Expression	Toppe Toppe Fore(113) float float float float float float float float float float float float		<u>a</u> nen. [10.] n % (3. 3. 11)	



3. Right-click on the added variable and select the "Enable Real-time Refresh".

2_studio - C:\Project\SensorSW_2024\RA-RRH46410_SampleProject\src\RRH46410_R	A2E1_NonOS\src\	RA_RRH46410.c - e ² studio					-	o ×
File Edit Source Refactor Navigate Search Project Renesas Views Run R	enesas Al Wind	ow Help						
i 📰 🏵 = 🔦 = i 🖉 🔌 i 🗶 🖿 💷 🖷 🗵 🧐 🖈 i i i i i i i i i i i i i i i i i i	• 🗠 • 🍇 🕪	- 00 🕼 🖏 🕹 🌙				Q	i 😭 🖬 C/	C++ 🎄 Debug
🍄 Debug × 🗈 🗎 🗎 🗎 🗎 🗎 🗎	1 startup.c	📸 main.c 🔞 bsp_delay.c 🐔 hal_entry.c	1 RA_RRH46410.c ×	- 0	🗱 Vari 🍫 Brea 🛍	Proj 🥂 Expr 🗙 📲 Eve	🖁 Peri 🚺	10 🗖 🗖
Portuge XI Portuge XI	21 21 22 24 24 24 25 26 26 27 27 28 20 20 27 28 20 20 20 20 20 20 20 20 20 20 20 20 20	<pre>Ord_SCORM.C., ord_SCORM.C., bruc_SCORM.C., bruc_SCORM.C., bruc_SCORM.C., bruc_SCORM.C., bruc_SCORM.C., bruc_SCORM.C., bruc_SCORM.C., bruc_SCORM.C., ord</pre>	<pre>/* Requested delay amount is in microseconds */ /* Requested delay amount is in milliseconds */ /* Requested delay amount is in milliseconds */ etup(void); (widd); g_t delay, rrh44410_delay_units_t units); g_iirq_callback_status = 0000_c4LLBACY_STATUS_MIT; g_iirq_callback_status = 0000_c4LLBACY_STATUS_MIT; = MB4640_0000_mode_pad_200_0000 \ * M44518_0000_mode_pad_200_0000 \ * M44518_0000_mode_pad_200_00000 \ * M44518_0000_mode_pad_200_0000 \ * M44518_0000_mode_pad_200_00000 \ * M44518_0000_mode_pad_2000000000000000000000000000000000000</pre>		Especialon gring_2nd_gen_de gring_2nd_gen_de gring_2nd_gen_de gring_2nd gring_and grin	http://file file file file http://file file file file http://file file file file file file file file file <td>Peri. Image: 1 Pari. Image: 1 Pari.</td> <td>0. • • • • • • • • • • • • • • • • • • •</td>	Peri. Image: 1 Pari. Image: 1 Pari.	0. • • • • • • • • • • • • • • • • • • •
< >	<							>
Suspended	 System Time 							

4. Start the Debug.

It is possible to check the real-time values.

😰 e2_studio - RRH46410_RA2E1_NonOS/ra_gen/main.c - e														122	o ×
Eile Edit Source Refactor Navigate Search Proje				and the second se											
[] S • 															++ 🔯 Debug
🍄 Debug 🗙	🕒 🙀 ite 🕴 🗖 🗖	startup.c					i main.c ×	RA_RRH46410.c	»2	- 0	(*)= Vari 🗣 Brea 🛍 P				
 El Briddini P. Add X, Mondo Dung, File (Reveas Odd		2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	/* gen #inclu #int ma { 8 ha	erated main sour- de "hal data.h"	e file - do net edi						Expression ig g. u.e. g. g. u	Type volatile rm_zm float [13] float	Value		
										4					Ŷ
			×								< ۱			C	
					mart Browser 🛛 🙀 Debugg		emory					■ × 注 ◎			- - -
		RRH46410_RA2E Target conne	1_NonOS Debu	g_Flat [Renesas GDB H	Hardware Debugging] [pi	± 10]									
		Starting dow Option Funct SECMPUXXX, w Finished dow Hardware bre Hardware bre Hardware bre Hardware bre	nload ion Select, riting to a nload akpoint set akpoint set akpoint set akpoint set	writing to addre	8 8	data ffffffffd fffffefeefcffef	H⊂effff Geffff0f00								
¢		¢													>
Running 🧼 🛶	0)										1			



Revision History

		Description	
Rev.	Date	Page	Summary
1.00	Dec.19.24	-	First Release



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 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 systemevaluation test for the given product.

Notice

- 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.
- 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.
- 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.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 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: <u>www.renesas.com/contact/</u>.