

RA Family, RX Family, RL78 Family, RZ Family ZMOD4xxx Sample Software Manual

Introduction

This sample software acquires gas data from the ZMOD4410, ZMOD4450 and ZMOD4510 gas sensors and calculates the result. In combination with the I2C driver of the FSP, the sample software controls the ZMOD4410 and ZMOD4510 through the I2C in the MCU to measure gases, acquire ADC data, and calculate the acquired result.

Target Devices

RA6M4 Group

RX65N Group RL78/G14 Group RL78/G23 Group

RZ/G2L Group

Trademarks

FreeRTOS[™] and FreeRTOS.org[™] are trade marks of Amazon Web Services, Inc.

Microsoft Azure RTOS is trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

1.	Overview4
2.	Environment for Confirming Operation
2.1	Environment for Confirming Operation on RA Family MCU4
2.2	Environment for Confirming Operation on RX Family MCU5
2.3	Environment for Confirming Operation on RL78/G14 Group MCU6
2.4	Environment for Confirming Operation on RL78/G23 Group MCU7
2.5	Environment for Confirming Operation on RZ Family MCU8
3.	ZMOD4410 Sensor specifications9
3.1	Sensor Specifications Overview9
3.2	Sensor function and methods10
3.2.1	Conversion of output data – Firmware / API / algorithms12
3.2.2	2 Typical hardware requirements
4.	ZMOD4450 Sensor specifications14
4.1	Sensor Specifications Overview
4.2	Sensor function and methods15
4.2.1	Conversion of output data – Firmware / API / algorithms16



4.2.2	Typical hardware requirements	. 17
5. Z	ZMOD4510 Sensor specifications	.18
5.1	Sensor Specifications Overview	. 18
5.2	Sensor function and methods	. 19
5.2.1	Conversion of output data – Firmware / API / algorithms	. 20
5.2.2	Typical hardware requirements	. 20
6. 3	Specification of Sample Software	.22
6.1	Sample Software Structure	
6.2	Specification of Sensor API Functions	
6.2.1	List of Sensor API Functions	
6.2.2	API Usage Guide	
6.3	Main Processing Flow of Sample Software	
6.3.1	ZMOD4410, ZMOD4450	
6.3.2	ZMOD4510	
6.3.3	Azure RTOS Project	
0.0.0		. 01
7. (Configuration Settings	.32
7.1	ZMOD4xxxx Gas Sensor Settings	
7.1.1	RA Family	
7.1.2	RX Family	
7.1.3	RL78 Family	
7.1.4	RZ Family	
7.2	Sensor Communication Middleware Settings	
7.2.1	RA Family	
7.2.2	RX Family	
7.2.3	RL78 Family	
7.2.4	RZ Family	
7.3	I2C Driver Settings	
7.3.1	RA Family	
7.3.2	RX Family	
7.3.3	RL78 Family	
7.3.4	RZ Family	
7.4	IRQ Driver Settings	
7.4.1	RZ Family	
1.4.1		. 00
8. (Guide for Changing the Target Device	.51
8.1	RA Sample Project	
8.1.1	Importing the Sample Project	
8.1.2	Modifying Settings of the FSP Configurator	
8.1.3	Changing sample code	
8.1.4	Changing when not using IRQ	
8.1.5	Changing toolchain setting	
8.2	RX Sample Project	
o.z 8.2.1	Importing the Sample Project	
8.2.1	Changing the Device	
0.2.2		. 00



8.2.3	Modifying Settings of the Smart Configurator	68
8.2.4	Changing toolchain setting	70
8.3	RL78 Sample Project	71
8.3.1	Sample project using Code Generator	71
8.4	RZ Sample Project	
8.4.1		
8.4.2	Modifying Settings of the FSP Configurator	91
8.4.3		
9. \	Viewing gas data	95
Revis	sion History	97
Gene	eral Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Pro-	ducts98
Notic	e	99



1. Overview

This sample software acquires data from the ZMOD4xxx gas sensor and calculates the result values. In combination with the I2C driver of the FSP, the sample software controls the ZMOD4xxx through the I2C in the MCU to measure gas environment, acquire ADC data, convert and calculate the acquired results.

2. Environment for Confirming Operation

2.1 Environment for Confirming Operation on RA Family MCU

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

Item	Description
Demonstration board	RTK7EKA6M4S00001BE (EK-RA6M4)
Microcontroller	RA6M4 (R7FA6M4AF3CFB :144pin)
Operating frequency	200MHz
Operating voltage	5V
Integrated development environment	e ² Studio 2023-01
C compiler	GCC 10.3.1.20210824
	IAR ANSI C/C++ Compiler V9.20.2.320/LNX for ARM
	ARM Compiler 6.18
FSP	V.4.5.0
RTOS	FreeRTOS [™] / Microsoft [®] Azure RTOS
Emulator	On board (J-LINK)
Interposer	Interposer Board to convert Type2/3 to Type 6A PMOD standard (US082-INTERPEVZ)
Sensor board	TVOC and Indoor Air Quality Sensor Pmod [™] Board (US082-ZMOD4410EVZ) Outdoor Air Quality Sensor Pmod [™] Board (US082-ZMOD4510EVZ)

Table 2-1 Operating Environment for RA Family

Table 2-2 Amount of Memory Used in RA Family

Area	Size (Non-OS)			Size (Non-OS) Size (Free RTOS)		Size (Azure RTOS)	
ZMOD	4410	4510	4450	4410+4510	4450	4410+4510	4450
sensor							
ROM	6.856	5.588	2 556	10.472	2 002	10.600	3.840
[bytes]	0,000	5,500	3,556	10,472	3,892	10,600	3,040
RAM	710	694	200	1 471	590	1 469	740
[bytes]	719	684	390	1,471	580	1,468	749

Memory size is calculated by functions and variables only related to ZMOD4xxx sensor. In RTOS, memory size does not include memory size of the thread.

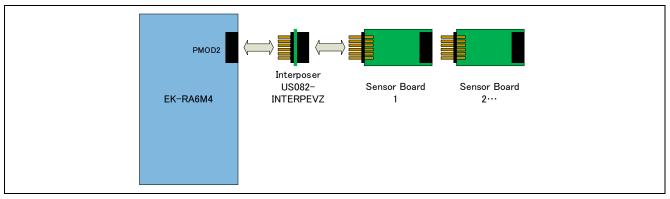


Figure 2-1 Hardware Connections for RA Family



2.2 Environment for Confirming Operation on RX Family MCU

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

Item	Description
Demonstration board	RPBRX65N (Envision Kit RX65N)
Microcontroller	RX65N (R5F565NEDDFB: 144pin)
Operating frequency	12MHz
Operating voltage	5V
Integrated development	e ² Studio 2023-01
environment	IAR EW for RX 4.20.1
C compiler	Renesas Electronics C/C++ compiler for RX family V.3.03.00
	GCC 8.3.0.202004
	IAR Toolchain for RX 8.4.10.7051
FIT	BSP V.7.20
RTOS	FreeRTOS™
Emulator	On board (E2OB)
Interposer	Interposer Board to convert Type2/3 to Type 6A PMOD standard (US082-INTERPEVZ)
Sensor board	TVOC and Indoor Air Quality Sensor Pmod [™] Board (US082-ZMOD4410EVZ)
	Outdoor Air Quality Sensor Pmod [™] Board (US082-ZMOD4510EVZ)

Table 2-3 Operating Environment for RX Family

Table 2-4 Amount of Memory Used in RX Family

Area	Size (Non-OS)			Size (Non-OS) Size (Free RTOS)		Size (Azure RTOS)	
ZMOD	0 4410 4510 4450		4410+4510	4450	4410+4510	4450	
sensor							
ROM	6.256	5.398	3.460	9,056	3.621	9,177	3,751
[bytes]	0,250	5,590	3,400	9,050	5,021	9,177	3,731
RAM	782	651	692	1.376	752	1,513	961
[bytes]	102	051	092	1,370	7.52	1,010	901

Memory size is calculated by functions and variables only related to ZMOD4XXX sensor. In RTOS, memory size does not include memory size of the thread.

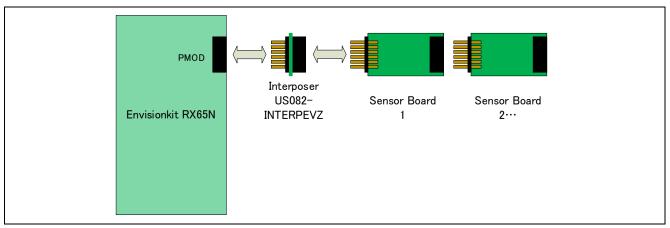


Figure 2-2 Hardware Connections for RX Family



2.3 Environment for Confirming Operation on RL78/G14 Group MCU

The operation of this software has been confirmed on an MCU of the RL78/G14 Group in the following environment.

ltem	Description
Demonstration board	RTK5RLG140C00000BJ (RL78/G14 Fast Prototyping Board)
Microcontroller	RL78/G14 (R5F104MLAFB: 80pin)
Operating frequency	20MHz
Operating voltage	3.3V
Integrated development	e ² studio 2023-01
environment	IAR EW for RL78 4.21.1
C compiler	C compiler package for RL78 family V1.11.00
	GCC for Renesas RL78 4.9.2.202103
	IAR Toolchain for RL78 4.21.1.2409
Emulator	On board (E2OB)
Sensor board	TVOC and Indoor Air Quality Sensor Pmod [™] Board (US082-ZMOD4410EVZ)
	Outdoor Air Quality Sensor Pmod [™] Board (US082-ZMOD4510EVZ)

Table 2-5	Operating	Environment for	RL78/G14 Group
-----------	-----------	-----------------	----------------

Table 2-6 Amount of Memory Used in RL78/G14 Group

Area	Size	Remarks
ROM	9,499 bytes	Include ZMOD4410 IAQ 2 nd Gen
RAM	551 bytes	library
ROM	7,263 bytes	Include ZMOD OAQ 2 nd Gen library
RAM	444 bytes	
ROM	5,235 bytes	Include ZMOD RAQ library
RAM	497 bytes	

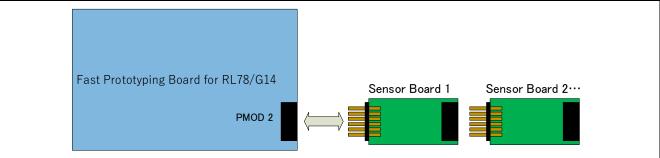


Figure 2-3 Hardware Connections for RL78/G14 Group



2.4 Environment for Confirming Operation on RL78/G23 Group MCU

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

ltem	Description
Demonstration board	RTK7RLG230CSN000BJ (RL78/G23-128p Fast Prototyping Board)
Microcontroller	RL78/G23 (R7F100GSN2DFB :128pin)
Operating frequency	32MHz
Operating voltage	3.3V
Integrated development	e ² studio 2023-01
environment	IAR EW for RL78 4.21.1
C compiler	C compiler package for RL78 family V1.11.00
	LLVM for RL78 10.0.0.202209
	IAR Toolchain for RL78 4.21.1.2409
Emulator	E2 Lite
Sensor board	TVOC and Indoor Air Quality Sensor Pmod [™] Board (US082-ZMOD4410EVZ)
	Outdoor Air Quality Sensor Pmod [™] Board (US082-ZMOD4510EVZ)

Table 2-7	Operating	Environment for	RL78/G23 (Group
-----------	-----------	-----------------	------------	-------

Table 2-8 Amount of Memory Used in RL78/G23 Group

Area	Size	Remarks
ROM	9,596 bytes	Include ZMOD4410 IAQ 2 nd Gen
RAM	479 bytes	library
ROM	7,288 bytes	Include ZMOD OAQ 2 nd Gen library
RAM	444 bytes	
ROM	5,500 bytes	Include ZMOD RAQ library
RAM	497 bytes	

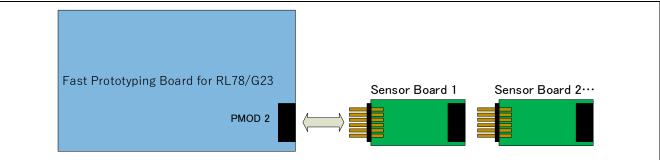


Figure 2-4 Hardware Connections for RL78/G23 Group



2.5 Environment for Confirming Operation on RZ Family MCU

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

Item	Description
Demonstration board	RTK9744L23S01000BE (RZ/G2L Evaluation Kit (SMARC))
Microcontroller	RZ/G2L (R9A07G044L23GBG:456pin)
Operating frequency	Arm® Cortex®-M33:200MHz、Arm® Cortex®-A55:1.2GHz
Operating voltage	5V
Integrated development	e ² Studio 2023-01
environment	
C compiler	GCC 10.3.1.20210824
FSP	V.1.2.0
RTOS	FreeRTOS™
Emulator	SEGGER J-LINK BASE
Sensor board	TVOC and Indoor Air Quality Sensor Pmod [™] Board (US082-ZMOD4410EVZ)
	Outdoor Air Quality Sensor Pmod [™] Board (US082-ZMOD4510EVZ)

Table 2-9	Operating	Environment for RZ Family
	oporanig	

Table 2-10 Amount of Memory Used in RZ Family

Area	Size (Non-OS)		Size (FreeRTOS)
ZMOD sensor	4410	4510	4410+4510
ROM[bytes]	7,271	5,693	10,494
RAM[bytes]	803	636	1,713

Memory size is calculated by functions and variables only related to ZMOD4XXX sensor. In RTOS, memory size does not include memory size of the thread.

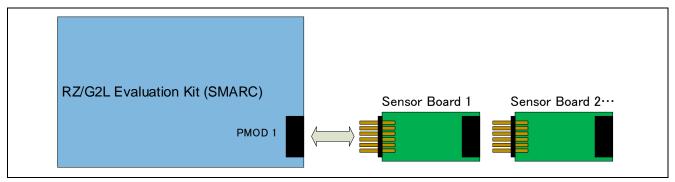


Figure 2-5 Hardware Connections for RZ Family



3. ZMOD4410 Sensor specifications

3.1 Sensor Specifications Overview

The ZMOD4410 Gas Sensor Module is designed to detect typical TVOC contaminations based on studies and international standards for indoor air quality. Characteristic module parameters are shown in Table 3-1. The response time for a gas stimulation is always within a few seconds, depending on the TVOC and its concentration. An active or direct airflow onto the sensor module is not necessary because diffusion of ambient gas does not limit the sensor module's response time.

Important: The ZMOD4410 also can detect safety-relevant gases for indoor air, such as carbon monoxide (CO); however, the sensor module is not designed to detect these interferants reliably and therefore it is not approved for use in any safety-critical or life-protecting applications. It must not be used in such applications, and Renesas disclaims all liability for any such use.

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit ^[a]
	Resistance	Ethanol in air	0	-	1000	ppm
	Measurement Range	Ethanol in air	0	-	1000000	ppb ^[c]
	IAQ and TVOCspecified	Ethanol in air for IAQ 1 st Gen	160	-	30000	ppb
	Measurement Range to meet UBA ^[b] TVOC specified	Ethanol in air for IAQ 2 st Gen	160	-	10000	ppb
	TVOC specified Measurement Range for PBAQ	Ethanol in air	1	-	2000	µg/m³
	Humidity Range for Sensor	All ZMOD operations except PBAQ operation, Non-condensing	0	-	90	% RH
	Module Operation	Specification to meet PBAQ, Non-condensing	-	50	-	% RH
S	Sensitivity over Lifetime	Resistance in Clean Air / Resistance at 10ppm Ethanol (R _{Air} /R _{Gas})		5		Ω/Ω
T-80	Sensor Module Response Time ^[d]	Samples needed to change to 80% of end value (all operation modes except ULP)	-	15	-	Samples
Vdd	Supply Voltage		1.7	-	3.6	V
Тамв	Ambient Temperature Range for Sensor Module Operation	All operation modes except PBAQ, Non-condensing	-40	-	65	°C
		Specification according to PBAQ, Non-condensing	-	21	-	°C
	Average Power: IAQ	Continuous and Odor Operation Mode	-	23	-	mW
	1 st Gen	Low Power Operation Mode	-	1.5	-	mW
	Average Power: IAQ 2 ^s Gen, Relative IAQ and Sulfur-based Odor Discrimination		-	6	-	mW

Table 3-1- Gas Sensor Module Specifications during Operation



	Average Power: IAQ 2 ^s Gen ULP and Relative IAQ ULP	Ultra-Low Power operation	-	0.16	-	mW
	Average Power: PBAQ		-	1	-	mW
	Average Power: Odor Operation		-	23	-	mW
LAGTINE	ACTIVE Supply Current, Active Mode including Heater Current for IAQ 1 st Gen Continuous and Odor Operation Mode	At VDD = 1.8 V	-	13	-	mA
IACTIVE		At VDD = 3.3V	-	7	-	mA
1	I _{ACTIVE} Supply Current, Active Mode including Heater Current for IAQ 2 ^s Gen, Relative IAQ, Ultra-low Power and Sulfur-based Odor Discrimination	At VDD = 1.8 V	-	7.4	16.2	mA
IACTIVE		At VDD = 3.3V	-	5.2	10.3	mA
	Supply Current, Active Mode including	At VDD = 1.8 V	-	9.9	16.4	mA
ACTIVE	Heater Current for PBAQ	At VDD = 3.3V	-	6.9	10.6	mA
ISLEEP	Current during measurement delays	Sleep Mode ASIC	-	450	-	nA

[a] The abbreviation ppm stands for "parts per million," and ppb is an abbreviation for "parts per billion." For example, 1 ppm equals 1000 ppb.

- [b] Source: Umweltbundesamt, *Beurteilung von Innenraumluftkontaminationen mittels Referenz- und Richtwerten*, (Bundesgesundheitsblatt Gesundheitsforschung Gesundheitsschutz, 2007).
- [c] Conversion from ppm to mg/m³ for most common TVOC is by the factor approximately 2; for example, 5ppm equals approximately 10mg/m³.
- [d] Response times depend on TVOC gas and concentration.

3.2 Sensor function and methods

The ZMOD 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 ZMOD4410. At present, five operation modes are released.

Family of IAQ software releases:

- Operation Mode 1: IAQ 1st Generation: Continuous Measurement of UBA levels for IAQ and eCO2
- Operation Mode 2: IAQ 1st Generation: Low Power Measurement of UBA levels for IAQ and eCO2
- Operation Mode 3: IAQ 2nd Generation: Using AI for improved ppm TVOC, IAQ and eCO2 functionality (recommended for new designs)
- Operation Mode 4: Odor Control signal based on Air Quality Changes
- Operation Mode 5: Sulfur-based Odor Discrimination
- Operation Mode 6: Relative IAQ Measure Relative IAQ based on Air Quality Changes
- Operation Mode 7: PBAQ TVOC measurement to meet PBAQ standards



By default, the IAQ 2nd Generation (Mode 3) operation should be used for new designs. In case of the need for a slightly faster sample rate and a larger VOC range (up to 30ppm), it is recommended to use the IAQ 1Gen algorithms.

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>ZMOD4410</u> webpage.

Parameter	Conditions	Minimum	Typical	Maximum	Unit
	Without additional calibration		±25		%
Accuracy for TVOC	With additional calibration		±15		%
Accuracy for IAQ	Without additional calibration		±10		%
Consistency	Part-to-Part Variation		±25		%
Durability to Siloxanes	Change in sensitivity		±5		%

Table 3-2 Typical ZMOD4410 Sensor Module Accuracy Achievable with Calibration



3.2.1 Conversion of output data - Firmware / API / algorithms

To operate the ZMOD4410, a firmware provided by Renesas containing an API, algorithm libraries, 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>ZMOD4410</u> webpage

The structure of theZMOD4410 firmware is described and illustrated in Figure 3-1 below.

- The "Hardware Abstraction Layer (HAL)" contains hardware-specific drivers include ZMOD4410 sensor API functions with gas measurement libraries, sensor communication middleware functions and low-level I2C drivers.
- The "ZMOD4410 sensor API (Application Programming Interface)" block contains the functions needed to operate the ZMOD4410.
 The "Gas Measurement Libraries" work with sensor API, it contains the functions and data structures needed to calculate the firmware specific results for the Indoor Air Quality related parameters, such as IAQ, TVOC, eCO2 (IAQ 1st Gen and IAQ 2nd Gen) or Air Quality Change (Odor), or Sulfur Odor result. These algorithms cannot be used in parallel. This block also contains the optional cleaning procedure and the optional math library for code size reduction. The libraries are described in more detail in the documents ZMOD4410-IAQ_xxx_Gen-lib.pdf, ZMOD4410-Odor-lib.pdf, and ZMOD4410-Sulfur_Odor-lib.pdf, ZMOD4410-PBAQ-lib.pdf, ZMOD4410-Rel_IAQ-lib.pdf, ZMOD4410-Rel_IAQ_ULP-lib.pdf. All of these files are part of the downloadable sample software packages.
- The "Programming Example" block provides a code example that is used to initialize the ZMOD4410, perform measurements, display the data output for each specific example, and start the optional cleaning function.
- The low-level driver "r_iic_master and r_sci_iic" block is the hardware-specific implementation of the I2C interface for Renesas RA MCU. This block contains read and write functions to communicate with the ZMOD4410 sensor via I2C buses.

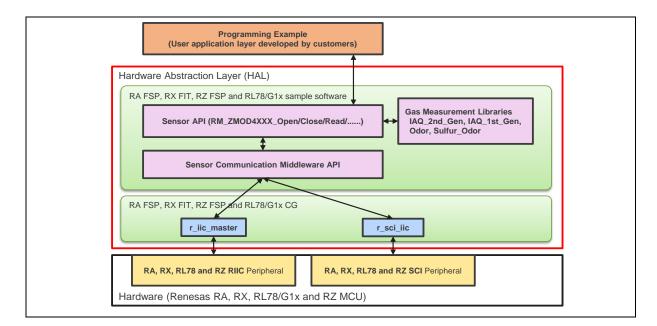


Figure 3-1 File Overview of ZMOD4410 Firmware



3.2.2 Typical hardware requirements

To operate the ZMOD4410, customer-specific hardware with a microcontroller unit (MCU) is needed. Depending on the sensor configuration and on the hardware itself, the requirements differ, and the following minimum requirements are provided as an orientation only:

- 12 to 20 kB program flash for ZMOD4410-related firmware code (MCU architecture and compiler dependent), note also Table 3-3
- 1kB RAM for ZMOD4410-related operations, note also Table 3-3
- Capability to perform I2C communication, timing functions, and floating-point instructions
- The algorithm functions work with variables saved in background and need memory retention between each call

	IAQ 2 nd Gen	IAQ 2 nd Gen ULP	IAQ 1 st Gen	PBAQ	Relative IAQ	Relative IAQ ULP	Odor	Sulfur Odor
Program flash	15.8	14.6	10.9	12.7	10.7	10.7	8.7	8.4
usage in kB								
RAM usage	496	480	284	436	400	400	168	402
(required variables)								
in bytes								
RAM usage (stack	496	336	-	448	224	224	-	368
size for library								
functions) in bytes								

Table 3-3 Exemplary Memory Footprint of ZMOD4410 Implementation on a RL78-G13 MCU



4. ZMOD4450 Sensor specifications

4.1 Sensor Specifications Overview

The ZMOD4450 Gas Sensor Module is designed to detect typical gases inside refrigeration applications associated with food ripening or rotting. Characteristic module parameters are shown in Table 4-1. The response time for a gas stimulation is always within a few seconds, depending on the gas and its concentration. An active or direct airflow onto the sensor module is not necessary because diffusion of ambient gas does not limit the sensor module's response time.

Important: The ZMOD4450 also can detect safety-relevant gases for indoor air, such as carbon monoxide (CO); however, the sensor module is not designed to detect these interferants reliably and therefore it is not approved for use in any safety-critical or life-protecting applications. It must not be used in such applications, and Renesas disclaims all liability for any such use.

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit [a]
	Resistance	Ethylene (C ₂ H ₄) in air	0		10	ppm
	Measurement	Trimethylamine (C ₃ H ₉ N) in air	0		600	ppb
	Range	Dimethyl sulfide (C₂H₅S) in air	0		180	ppb
RAQ	Refrigeration Air Quality Range	Change rate based on resistance	0		3	
	Temperature Range		0		25	°C
	Repeatability	Variation in sensor module signal		±10		%
T-90	Sensor Module Response Time	Time to change to 90% of end value		10		S
V _{DD}	Supply Voltage		1.7	-	3.6	V
Т _{амв}	Ambient Temperature Range for Sensor Module Operation		-40	25	65	°C
	Average Power ZMOD4450	Continuous Operation	-	23	-	mW
_	Supply Current, Active Mode	At VDD = 1.8 V	-	13	-	mA
I _{ACTIVE} ir	including Heater Current.	At VDD = 3.3V	-	7	-	mA
SLEEP	Current during measurement delays	Sleep Mode ASIC	-	450	-	nA

Table 4-1- Gas Sensor Module Specifications during Operation

[a] The abbreviation ppm stands for "parts per million," and ppb is an abbreviation for "parts per billion." For example, 1 ppm equals 1000 ppb.



4.2 Sensor function and methods

The ZMOD 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 ZMOD4450. At present, five operation modes are released.

Family of IAQ software releases:

- Operation Mode 1: RAQ – Control signal based on Refrigeration Air Quality Changes

In addition, details for sensitivity, reliability, sample rates, and sensor module influences are explained in detail in the following sections. All graphs and information show the typical responses that are to be expected from the sensor module upon exposure to a variety of test conditions. For more information, including application notes, white papers, blog, and manuals, visit the <u>ZMOD4450</u> product page.

Table 4-2 Typical ZMOD4450 Sensor Module Accuracy Achievable with Calibration

Parameter	Conditions	Minimum	Typical	Maximum	Unit
Accuracy for RAQ	Without additional calibration		±10		%
Consistency	Part-to-Part Variation		±25		%
Durability to Siloxanes	Change in sensitivity		±5		%



4.2.1 Conversion of output data – Firmware / API / algorithms

To operate the ZMOD4450, a firmware provided by Renesas containing an API, algorithm libraries, 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>ZMOD4450</u> webpage

The structure of the ZMOD4450 firmware is described and illustrated in Figure 4-1 below.

- The "Hardware Abstraction Layer (HAL)" contains hardware-specific drivers include ZMOD4450 sensor API functions with gas measurement libraries, sensor communication middleware functions and low-level I2C drivers.
- The "ZMOD4450 sensor API (Application Programming Interface)" block contains the functions needed to operate the ZMOD4450.
 The "Gas Measurement Libraries" work with sensor API, it contains the functions and data structures needed to calculate the firmware specific results for the Refrigeration Air Quality related parameters. These algorithms cannot be used in parallel. This block also contains the optional cleaning procedure and the optional math library for code size reduction. The library is described in more detail in the document ZMOD4450-raq-lib.pdf. All of these files are part of the downloadable sample software packages.
- The "Programming Example" block provides a code example that is used to initialize the ZMOD4450, perform measurements, display the data output for each specific example, and start the optional cleaning function.
- The low-level driver "r_iic_master and r_sci_iic" block is the hardware-specific implementation of the I2C interface for Renesas RA MCU. This block contains read and write functions to communicate with the ZMOD4450 sensor via I2C buses.

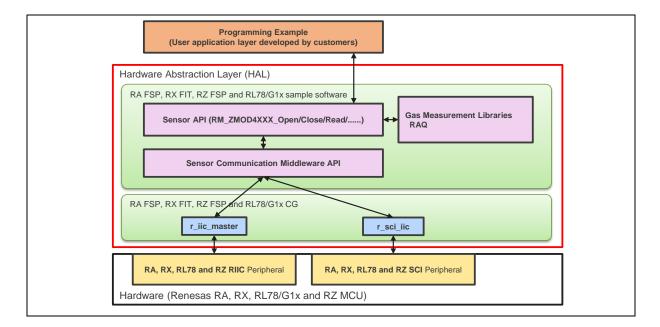


Figure 4-1 File Overview of ZMOD4450 Firmware



4.2.2 Typical hardware requirements

To operate the ZMOD4450, customer-specific hardware with a microcontroller unit (MCU) is needed. Depending on the sensor configuration and on the hardware itself, the requirements differ, and the following minimum requirements are provided as an orientation only:

- 10 to 20 kB program flash for ZMOD4450-related firmware code (MCU architecture and compiler dependent), note also Table 4-3
- 1kB RAM for ZMOD4450-related operations, note also Table 4-3
- Capability to perform I2C communication, timing functions, and floating-point instructions
- The algorithm functions work with variables saved in background and need memory retention between each call

Table 4-3 Exemplary Memory Footprint of ZMOD4450 Implementation on a RA6M4 MCU

	RAQ
Program flash usage in kB	1.8
RAM usage in Bytes	174



5. ZMOD4510 Sensor specifications

5.1 Sensor Specifications Overview

The ZMOD4510 Gas Sensor Module detects typical gases based on studies and international standards for outdoor air quality. Characteristic module parameters are shown in Table 5-1. The ZMOD4510 uses a sequence of applied temperatures in order to sample the air and report an Air Quality Index (AQI) based on the EPA standard¹. The sensor does not require an active or direct airflow onto the sensor module because diffusion of ambient gas does not limit the sensor response time.

Important: The ZMOD4510 can also detect safety-relevant gases; however, the sensor is not designed to detect these interferants reliably and therefore it is not approved for use in any safety-critical or life-protecting applications. It must not be used in such applications, and Renesas disclaims all liability for any such use.

Symb ol	Parameter	Conditions	Minim um	Typical	Maximum	Unit ^[a]
AQI	Air Quality Index	OAQ 1 st Gen: Rating according to EPA for ozone and nitrogen dioxide	0		500	
		OAQ 2 nd Gen: Rating according to EPA for ozone	0		500	
	Massurement Dange OAO	Ozone (non-selective)	20		500	ppb
	Measurement Range OAQ 1 st Gen	Nitrogen dioxide (non- selective)	20		500	ppb
	Maggurament Dange OAO	Ozone (selective)	20		500	ppb
	Measurement Range OAQ 2 nd Gen	NO ₂ cross sensitivity at 200ppb		25		AQI Level
RH	Humidity Range	Non-condensing	5		90	% RH
т	Temperature Range	Typical outdoor environment	-20		50	°C
		Extended range	-40		65	°C
Vdd	Supply Voltage for ZMOD4510 Sensor Module		1.7	-	3.6	V
Тамв	Ambient Temperature Range for Sensor Operation		-40	-	65	°C
T _{OPERA} TION	Operation Temperature Sequence of Sense Element ^[a]		200		450	°C
	Average Power: OAQ 1 st Gen	Outdoor Air Quality	-	21	-	mW
	Average Power: OAQ 2 nd Gen	Selective ozone with ultra-low power	-	0.2	-	mW

Table 5-1- Gas Sensor Module Specifications during Operation



¹ AirNow, US Environmental Protection Agency, Air Quality Index (AQI) Basics; available at: <u>https://airnow.gov/index.cfm?action=aqibasics.aqi</u>

	Supply Current, Active	At V _{DD} = 1.8 V	-	11	13	mA
IACTIVE	Mode including Heater Current for OAQ 1 st Gen	At V _{DD} = 3.3 V	-	8	10	mA
_	Supply Current, Active	At V _{DD} = 1.8 V	-	10	12	mA
IACTIVE	Mode including Heater Current for OAQ 2 nd Gen	At V _{DD} = 3.3 V	-	6	8	mA
ISLEEP	Current during measurement delays	Sleep Mode ASIC	_	450	-	nA

[b] The abbreviation ppm stands for "parts per million," and ppb is an abbreviation for "parts per billion." For example, 1 ppm equals 1000 ppb.

5.2 Sensor function and methods

The ZMOD 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 ZMOD4510. Currently, two operation modes are released using the same ZMOD4510 hardware:

Family of Outdoor Air Quality (OAQ) software releases:

- Operation Mode 1 OAQ 1st Generation: Measurement of Air Quality
- Operation Mode 2 OAQ 2nd Generation: Selective Ozone featuring Ultra-Low Power

In addition, details for sensitivity, reliability, sample rates, and sensor module influences are explained in detail in the following sections. All graphs and information show the typical responses that are to be expected from the sensor module upon exposure to a variety of test conditions. For more information, including application notes, white papers, blog, and manuals, visit the <u>ZMOD4510</u> product page.

Parameter	Conditions	Minimu m	Typical	Maximum	Unit
Accuracy	Without additional calibration		±50		AQI
Consistency	Part-to-Part Variation		±50		AQI
Durability to Siloxanes	Change in sensitivity during 15,000 ppm·h exposure with D4 and D5		±8		%



5.2.1 Conversion of output data - Firmware / API / algorithms

To operate the ZMOD4510, a firmware provided by Renesas containing an API, algorithm libraries, and an example should be used. For implementing the sensor module in a customer-specific application, detailed information on the programming is available. More information and guidance on the firmware integration, architecture, and supported platforms are available in the ZMOD4510 Programming Manual – Read Me. Code Examples in C and additional firmware descriptions for API, HAL, libraries, etc., are included at no cost in the downloadable firmware package from the <u>ZMOD4510</u> product page.

The structure of the ZMOD4510 firmware is described and illustrated in Figure 5-1 below.

- The "Hardware Abstraction Layer (HAL)" contains hardware-specific drivers include ZMOD4510 sensor API functions with gas measurement libraries, sensor communication middleware functions and low-level I2C drivers.
- The "ZMOD4510 sensor API (Application Programming Interface)" block contains the functions needed to operate the ZMOD4510.
 The "Gas Measurement Libraries" work with sensor API, it contains the functions and data structures needed to calculate the firmware specific results for the Air Quality Index (AQI). These algorithms cannot be used in parallel. This block also contains the optional cleaning library. The libraries are described in more detail in the documents ZMOD4510 OAQ_1st_Gen-lib.pdf and ZMOD4510 OAQ_2nd_Gen-lib.pdf.
- The "Programming Example" block provides a code example that is used to initialize the ZMOD4510, perform measurements, display the data output for each specific example, and start the optional cleaning function.
- The low-level driver "r_iic_master and r_sci_iic" block is the hardware-specific implementation of the I2C interface for Renesas RA MCU. This block contains read and write functions to communicate with the ZMOD4510 sensor via I2C buses.

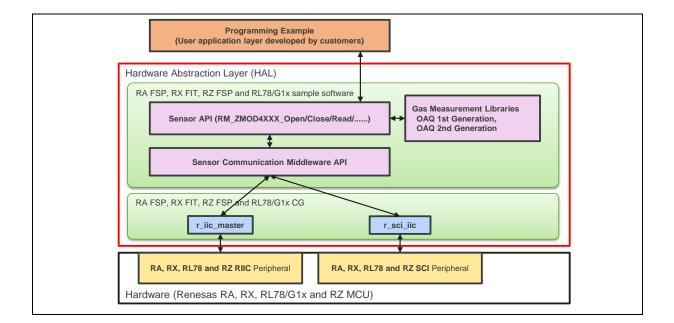


Figure 5-1 File Overview of ZMOD4410 Firmware

5.2.2 Typical hardware requirements



To operate the ZMOD4510, customer-specific hardware with a microcontroller unit (MCU) is needed. Depending on the sensor configuration and on the hardware itself, the requirements differ, and the following minimum requirements are provided as an orientation only:

- 12 to 20 kB program flash for ZMOD4510-related firmware code (MCU architecture and compiler dependent), note also Table 5-3
- 1kB RAM for ZMOD4510-related operations, note also Table 5-3
- Capability to perform I2C communication, timing functions, and floating-point instructions
- The algorithm functions work with variables saved in background and need memory retention between each call

Table 5-3 Exemplar	v Memory F	Footprint of ZMOD4510	Implementation on a	a RI 78-G13 MCU
	y wichiory i		implomontation on a	

	OAQ 2 nd Gen	OAQ 1 st Gen
Program flash usage in kB	12.9	10.2
RAM usage (required variables) in bytes	250	266
RAM usage (stack size for library functions, worst case) in bytes	544	244



6. Specification of Sample Software

The sample software package contains 3 projects. Each project is described below.

6.1 Sample Software Structure

Figure 6-1 **Block diagram of Sample** Software shows structure of sample software blocks.

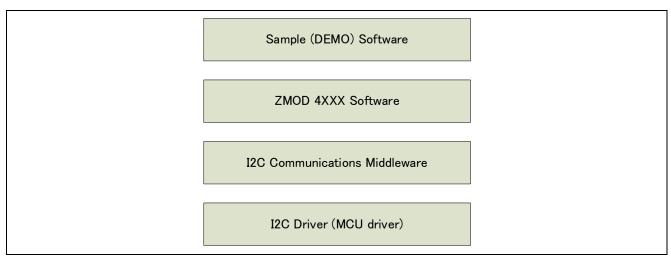


Figure 6-1 Block diagram of Sample Software

6.2 Specification of Sensor API Functions

6.2.1 List of Sensor API Functions

The sensor API includes following functions. Please refer to (RA Flexible Software Package Documentation) for the detail of API functions.

Function	Feature
RM_ZMOD4XXX_Open	Opens sensor
RM_ZMOD4XXX_Close	Closes the sensor
RM_ZMOD4XXX_MeasurementStart	Starts a measurement
RM_ZMOD4XXX_MeasurementStop	Stops a measurement
RM_ZMOD4XXX_StatusCheck	Read status of the sensor
RM_ZMOD4XXX_Read	Reads ADC data
RM_ZMOD4XXX_TemperatureAndHumiditySet	Sets temperature and humidity. (ZMOD4410
	IAQ_2nd_Gen_ULP operation and ZMOD4510
	OAQ_2nd_Gen operation)
RM_ZMOD4XXX_DeviceErrorCheck	ADC Data Validity Check Process
RM_ZMOD4XXX_laq1stGenDataCalculate	Calculate IAQ 1st Gen. values from ADC data
RM_ZMOD4XXX_laq2ndGenDataCalculate	Calculate IAQ 2nd Gen. values from ADC data
RM_ZMOD4XXX_OdorDataCalculate	Calculate Odor values from ADC data
RM_ZMOD4XXX_SulfurOdorDataCalculate	Calculate Sulfur Odor values from ADC data
RM_ZMOD4XXX_Oaq1stGenDataCalculate	Calculate OAQ 1st Gen. values from ADC data
RM_ZMOD4XXX_Oaq2ndGenDataCalculate	Calculate OAQ 2nd Gen. values from ADC data
RM_ZMOD4XXX_RaqDataCalculate	Calculate RAQ values from ADC data
RM_ZMOD4XXX_RellaqDataCalculate	Calculate Rel IAQ. values from ADC data
RM_ZMOD4XXX_PbaqDataCalculate	Calculate PBAQ values from ADC data



6.2.2 API Usage Guide

Figure 6-2 Transition of API Functions shows the transition diagram of functions calling order as the usage condition of ZMOD4XXX API functions.

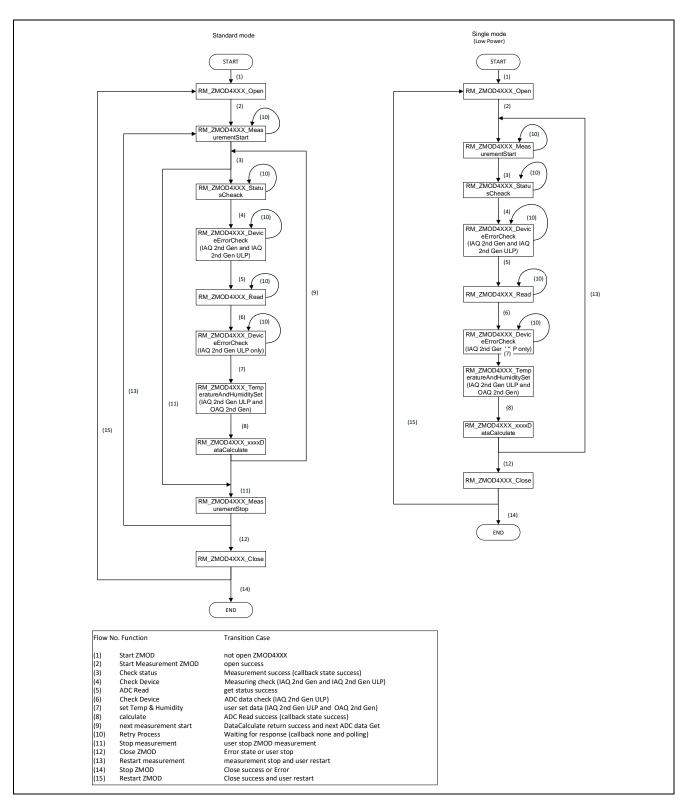


Figure 6-2 Transition of API Functions



The calling conditions for each function are as follows:

- RM_ZMOD4XXX_Open:	(1) When starting ZMOD4xxx
	(15) Re-stating after RM_ZMOD4XXX_Close
- RM_ZMOD4XXX_Close:	(12) Successful completion or abnormal end of
	individual processing
- RM_ZMOD4XXX_MeasurementStart :	(2) Starting measurement after
	RM_ZMOD4XXX_Open
	(13) When read next measurement data
	(10) Retry by waiting for measurement start response
- RM_ZMOD4XXX_MeasurementStop :	(11) When stop measurement
- RM_ZMOD4XXX_StatusCheack :	(3) Status cheack by polling
- RM_ZMOD4XXX_Read :	(5) When acquiring measurement data after
	RM_ZMOD4XXX_StatusCheck
	(10) Retry due to waiting for data acquisition response
RM_ZMOD4XXX_TemperatureAndHumiditySet :	(5) RM_ZMOD4XXX_xxxxDataCalculate
	preprocessing (IAQ-2nd-Gen-ULP and OAQ-2nd- Gen ambient compensation)
 RM_ZMOD4XXX_DeviceErrorCheck : 	(4) Get the status being measured
	(6) Checking ADC data
RM_ZMOD4XXX_xxxxDataCalculate :	(8) Calculate data after RM_ZMOD4XXX_Read

Note:

Since RM_ZMOD4XXX_Open checks the state of the I2C driver, the I2C driver must be opened before the RM_ZMOD4XXX_Open processing.

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

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



6.3 Main Processing Flow of Sample Software

The sample software performs the driver opening process, and then repeats the process of starting sensor measurement, acquiring sensor data, and calculating measurement results.

The OS version is controlled by semaphores, and the two threads that control the sensors run in parallel.

6.3.1 ZMOD4410, ZMOD4450

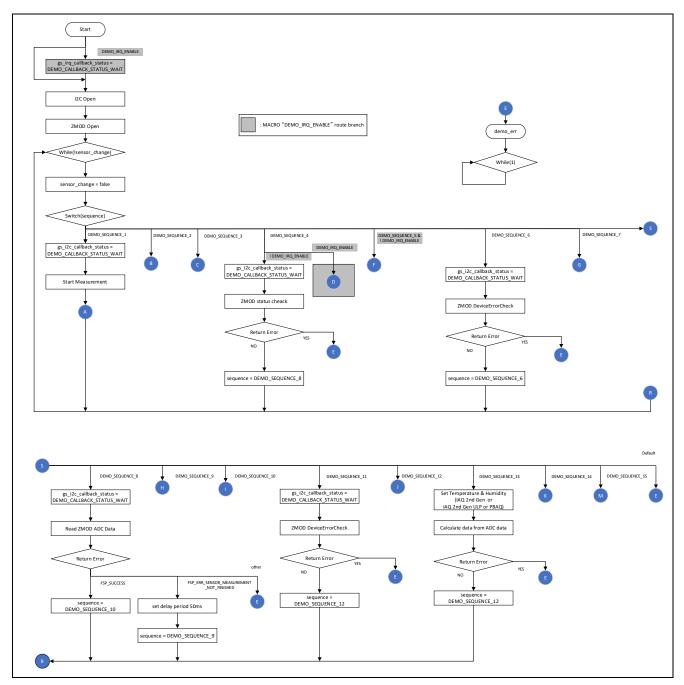
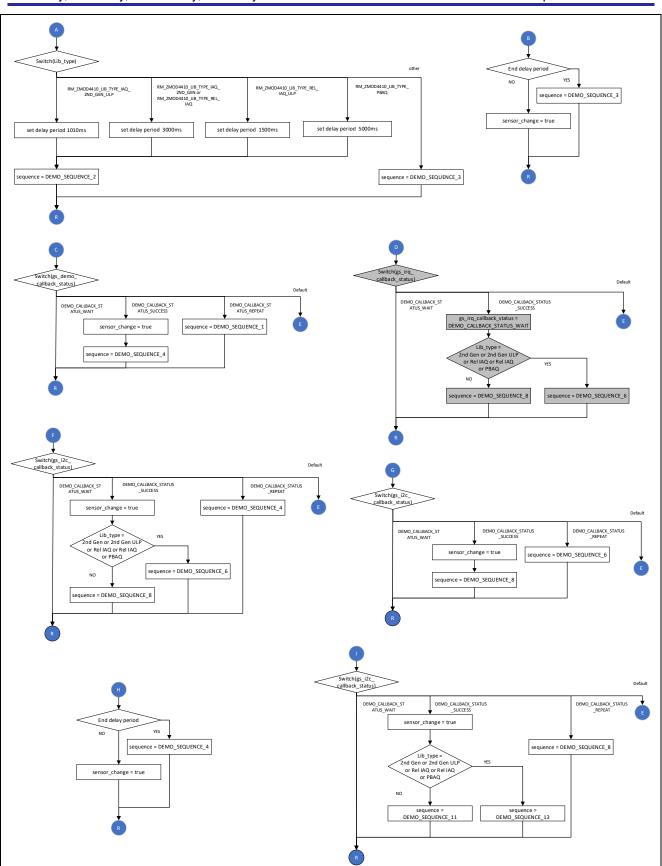


Figure 6-3 Flow 1 of ZMOD4410, ZMOD4450 Sample Software Main Processing





ZMOD4xxx Sample Software Manual

Figure 6-4 Flow 2 of ZMOD4410, ZMOD4450 Sample Software Main Processing



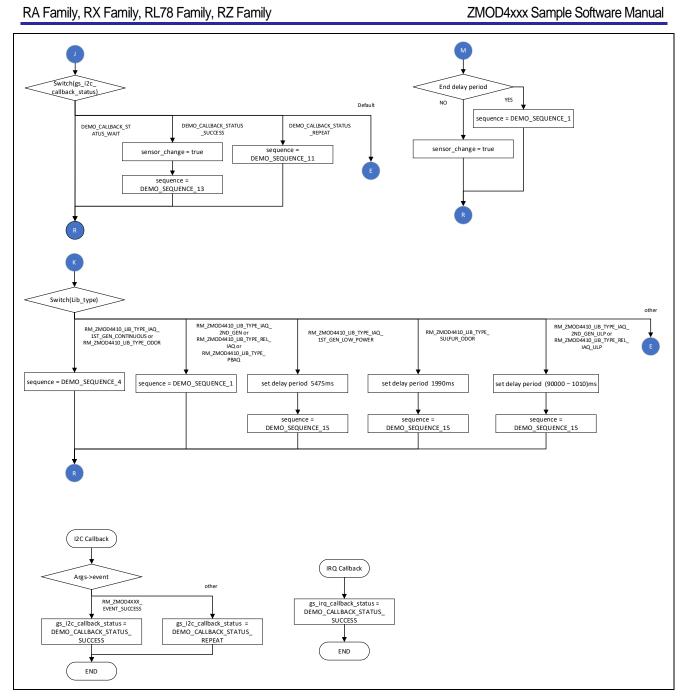


Figure 6-5 Flow 3 of ZMOD4410, ZMOD4450 Sample Software Main Processing



6.3.2 ZMOD4510

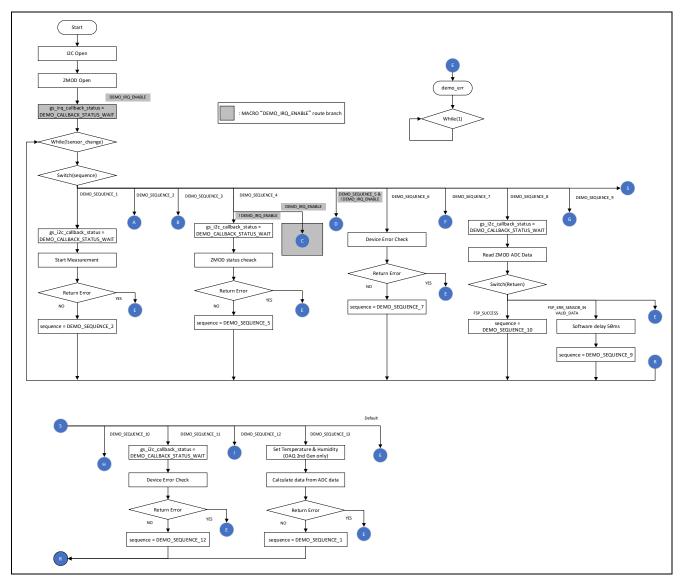


Figure 6-6 Flow 1 of ZMOD4510 Sample Software Main Processing

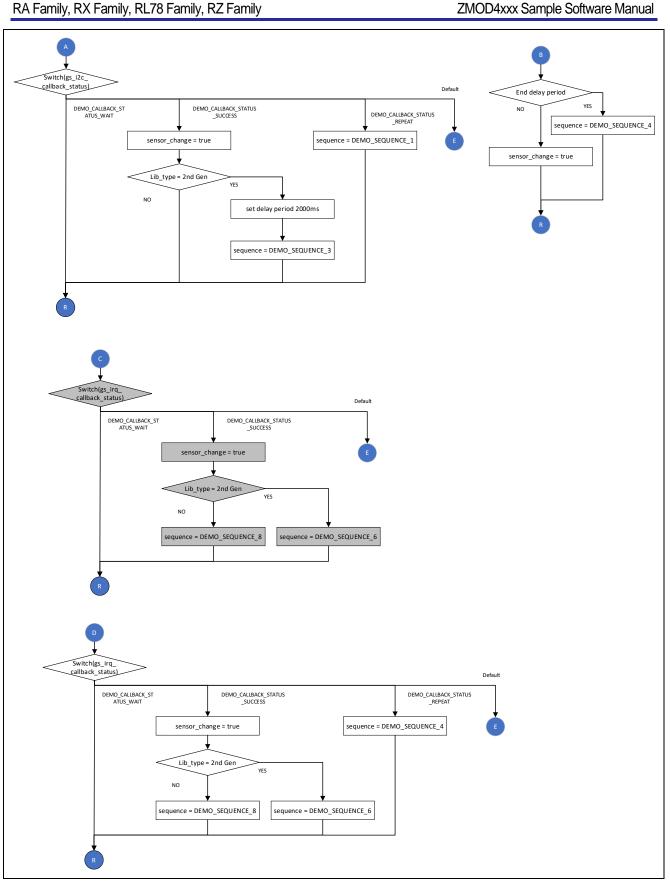


Figure 6-7 Flow 2 of ZMOD4510 Sample Software Main Processing

RENESAS

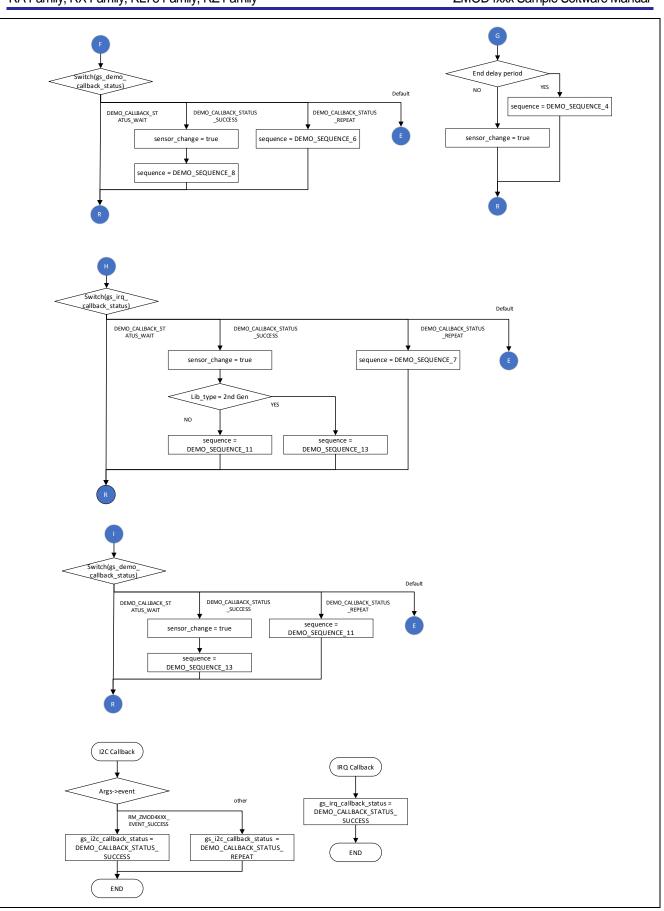


Figure 6-8 Flow 3 of ZMOD4510 Sample Software Main Processing



6.3.3 Azure RTOS Project

The RX Azure RTOS project has the following changes from the default.

1. src/hardware_setup.c

25th line: Change from 100u to 1000u

2. src/demo_thread.c

57th line: Add extern void tx_application_define_user (void);

179th line : Add tx_application_define_user();

src/rtos_skelton/zmod4410_sensor_thread_entry.c , zmod4510_sensor_thread_entry.c
 27th line: Change from #include "azurertos_object_init.h" to #include "tx_api.h"



7. Configuration Settings

7.1 ZMOD4xxxx Gas Sensor Settings

7.1.1 RA Family

Select the rm_zmod4xxx stack in the "Stack" tabbed page of the FSP Configurator, and the configurable items are shown in the "Properties" tabbed page.

The following items and values can be specified.

Important: Only one rm_zmod4xxx stack can be registered, and multiple registrations are not allowed.

Configurable Item	Value	Description		
Common				
Parameter Checking	Default (BSP)	Enable or disable the parameter check		
	Enabled	processing.		
	Disabled	When "Enabled" is selected, the project is built so that the generated code includes the parameter check processing.		
Module g_zmod	l4xxxx_sensor ZMOD4XX	KX on rm_zmod4xxx		
Name	g_zmod4xxx_sensor0	Specify the name of the module. A module name conforming to the C language standard can be specified.		
Callback	zmod4xxx_comms_i2c_callback	Specify the name of the user callback function. A callback function name conforming to the C language standard can be specified. When "NULL" is specified, no callback function is used.		
IRQ Callback	zmod4xxx_irq_callback	Specify the IRQ user callback function name. A callback function name conforming to the C language standard can be specified. When "NULL" is specified, no callback function is used.		

Table 7-1 ZMOD4xxx Settings for RA Family



7.1.2 RX Family

Select the r_zmod4xxx_rx component on the [Component] tabbed page of the Smart Configurator, and the configurable items will be shown in the [Configure] panel.

The following items and values can be specified.

Important: Only one rm_zmod4xxx stack can be registered, and multiple registrations are not

設定項目	設定値	説明			
Configurations	Configurations				
Parameter Checking	Default (BSP)	Enable or disable the parameter check			
	Enabled	processing. When "Enabled" is selected, the project is built			
	Disabled	so that the generated code includes the parameter check processing.			
Operation mode of ZMOD4XXX Sensors	IAQ 1st Gen. (Continuous)	Specify ZMOD4xxx sensor operation mode. 1: IAQ 1st Gen. (Continuous) 2: IAQ 1st Gen. (Low Power) 3: IAQ 2nd Gen. 4: Odor 5: Sulfur-based Odor 6: OAQ 1st Gen. 7: OAQ 2nd Gen. 8: IAQ 2nd Gen. Ultra-Low Power 9: RAQ 10:Rel IAQ. 11:Rel IAQ. Ultra-Low Power mode 12:PBAQ.			
I2C communication device number	I2C Communication Device(x) (x = 0 - 15)	Specify the communications device number to be used by the sensor.			
I2C Callback function name	zmod4xxx_user_i2c_callback0	Specify the name of I2C callback function. A callback function name conforming to the C language standard can be specified.			
Enable IRQ	Enable or Disable	Specify IRQ enable or disable.			
IRQ Callback function name	zmod4xxx_user_irq_callback0	Specify the name of IRQ callback function. A callback function name conforming to the C language standard can be specified.			
IRQ number	IRQ(x) (x = 0 - 15)	Specify valid IRQ number.			
IRQ Trigger	Rising	Specify IRQ trigger from Low Level, Falling, Rising, Both Edges.			
IRQ Interrupt Priority	Priority(x) (x = 0 - 15)	Specify IRQ interrupt priority.			

Table 7-2 ZMOD4xxx Settings for RX Family



7.1.3 RL78 Family

Settings can be modified by changing the values of constants defined in the \zmod4xxx\r_config\r_zmod4xxx_rl_config.h file in the project tree of the sample project.

The following items and values can be specified.

定数名	設定値	説明
Configurations		
RM_ZMOD4XXX_CFG_P ARAM_CHECKING_ENAB LE	0	Enable (1) or disable (0) the parameter check processing.
	1	When "1" is specified, the project is built so that the generated code includes the parameter check processing.
RM_ZMOD4XXX_CFG_D EVICE_NUM_MAX	1	Specify the number of ZMOD4xxx sensor
RM_ZMOD4XXX_CFG_D EVICE(x)_OPERATION_M ODE (x = 0 - 1)	1	Specify ZMOD4xxx sensor operation mode.* 1: IAQ 1st Gen. (Continuous) 2: IAQ 1st Gen. (Low Power) 3: IAQ 2nd Gen. 8: IAQ 2 nd Gen. Ultra-Low Power 4: Odor 5: Sulfur-based Odor 6: OAQ 1st Gen. 7: OAQ 2nd Gen. 9: RAQ 10:Rel IAQ. 11:Rel IAQ. Ultra-Low Power mode 12:PBAQ.
$RM_ZMOD4XXX_CFG_D$ EVICE(x)_COMMS_INST ANCE (x = 0 - 1)	g_comms_i2c_device(x) (x = 0 - 4)	Specify the communications device number to be used by the sensor.
$RM_ZMOD4XXX_CFG_D$ EVICE(x)_COMMS_12C_C ALLBACK (x = 0 - 1)	zmod4xxx_user_i2c_callback 0	Specify the name of I2C callback function. A callback function name conforming to the C language standard can be specified.
$\begin{array}{l} RM_ZMOD4XXX_CFG_D\\ EVICE(x)_IRQ_ENABLE\\ (x=0-1) \end{array}$	Enable or Disable	Specify IRQ enable or disable.
RM_ZMOD4XXX_CFG_D EVICE(x)_IRQ_CALLBAC K (x = 0 - 1)	zmod4xxx_user_irq_callback 0	Specify the name of IRQ callback function. A callback function name conforming to the C language standard can be specified.
RM_ZMOD4XXX_CFG_D EVICE(x)_IRQ_NUMBER (x = 0 - 1)	R_INTC(x) (x = 0 - 11)	Specify the number of IRQ

*When using Code Generator or using Smart Configurator of e² studio 2021-10 or later, library settings are not automatically set in the build settings. Please set the library settings manually after code generation.



7.1.4 RZ Family

Select the rm_zmod4xxx stack in the "Stack" tabbed page of the FSP Configurator, and the configurable items are shown in the "Properties" tabbed page.

The following items and values can be specified.

Important: Only one rm_zmod4xxx stack can be registered, and multiple registrations are not allowed.

Configurable Item	Value	Description	
Common			
Parameter Checking	Default (BSP)	Enable or disable the parameter check processing. When "Enabled" is selected, the project is built so that the generated code includes the parameter check processing.	
	Enabled		
	Disabled		
Module g_zmod4xxxx_sensor0 ZMOD4XXX Gas Sensor (rm_zmod4xxx)			
Name	g_zmod4xxx_sensor0	Specify the name of the module. A module name conforming to the C language standard can be specified.	
Callback	zmod4xxx_comms_i2c_callback	Specify the name of the user callback function. A callback function name conforming to the C language standard can be specified. When "NULL" is specified, no callback function is used.	
IRQ Callback	zmod4xxx_irq_callback	Specify the IRQ user callback function name. A callback function name conforming to the C language standard can be specified. When "NULL" is specified, no callback function is used.	

Table 7-4 ZMOD4xxx Settings for RZ Family



7.2 Sensor Communication Middleware Settings

7.2.1 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.

The following items and values can be specified.

Table 7-5 Communication Middleware Settings for RA Family

Configurable Item	Value	Description		
Common				
Parameter Checking	Default (BSP) Enabled Disabled	Enable or disable the parameter check processing. When "Enabled" is selected, the project is built so that the generated code includes the parameter check processing.		
Module g_comms_i2c_device I2C Communication Device on				
rm_comms_i2c				
Name	g_comms_i2c_device0	Specify the name of the module. A module name conforming to the C language standard can be specified.		
Semaphore Timeout	0xFFFFFFFF	For an RTOS project, specify the time of semaphore timeout.		
Slave Address	0x32	Specify the slave address. When rm_zmod4xxx is used, this value is automatically specified and cannot be modified.		
Address Mode	7-Bit	Specify the number of slave address bits. When rm_zmod4xxx is used, this value is automatically specified and cannot be modified.		
Callback	rm_zmod4xxx_comms_i2c_callback	Specify the name of the user callback function. When rm_zmod4xxx is used, this value is automatically specified and cannot be modified.		
Module g_comms_i2c_bus0 I2C Shared Bus on rm_comms_i2c				
Name	g_comms_i2c_bus0	Specify the name of the I2C module.		
Bus Timeout	0xFFFFFFF	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 is enabled.		



7.2.2 RX Family

Select the r_comms_i2c_rx component on the [Component] tabbed page of the Smart Configurator, and the configurable items will be shown in the [Configure] panel.

The following items and values can be specified.

Configurable Item	Value	Description
Configurations	Value	Description
Parameter Checking	System Default Enabled Disabled	Enable or disable the parameter check processing. When "Enabled" is selected, the project is built so that the generated code includes the
Number of communication lines	Unused, 1 – 16 (default: 1)	parameter check processing. Specify the number of communications bus lines that can be connected.
Number of I2C Devices	Unused, 1 – 16 (default: 1)	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.
IIC 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 the I2C bus for I2C Shared Bus(x) (x = 0 - 15)	0xFFFFFFF	Specify the time of I2C bus(x) timeout. (x = 0 - 15)
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 communication device(x) (x = 0 - 15)	0x32	Specify the slave address of the device to be connected to the communications bus. If you are using r_zmod4xxx_rx, specify 0x32.
Slave address mode for communication device(x) (x = 0 - 15)	7 bit address mode	Specify the slave address mode. If you are using r_zmod4xxx_rx, specify the 7- bit address mode.
Callback function for 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_zmod4xxx_rx, specify rm_zmod4xxx_callback(y) (y = 0).

Table 7-6 Communication Driver Settings for the RX Family MCU



7.2.3 RL78 Family

Settings can be modified by changing the values of constants defined in the \zmod4xxx\r_config\r_comms_i2c_rl_config.h file in the project tree of the sample project.

The following items and values can be specified.

Constant Name	Value	Description
Configurations		
COMMS_I2C_CFG_PARA M_CHECKING_ENABLE	0	Enable (1) or disable (0) the parameter check processing. When "1" is selected, the project is
	1	built so that the generated code includes the parameter check processing.
COMMS_I2C_CFG_BUS_N	1	Specify the number of
UM_MAX	2	communication bus lines that can be connected.
	3	
	4	
	5	
COMMS_I2C_CFG_DEVIC	1	Specify the number of I2C devices
E_NUM_MAX	2	can be connected.
	3	
	4	
	5	
COMMS_I2C_CFG_BUS(x)	COMMS_DRIVER_I2C	Specify the I2C type to be used for
_DRIVER_TYPE (x = 0 - 4)	COMMS_DRIVER_SAU_I2C	the communication bus.
$\begin{array}{l} \hline COMMS_I2C_CFG_DEVIC\\ E(x)_BUS_CH\\ (x = 0 - 4) \end{array}$	g_comms_i2c_bus(x)_extended_cf g (x = 0 - 4)	Specify the I2C bus configuration to be used for the communication bus.
COMMS_I2C_CFG_DEVIC E(x)_SLAVE_ADDR (x = 0 - 4)	0x32	Specify the slave address of the device to be connected to the communications bus. If you are using rm_zmod4xxx, specify 0x32.
COMMS_I2C_CFG_BUSx_ CALLBACK_ENABLE	0	Enable (1) or disable (0) the user callback function.
(x = 0 - 4)	1	
COMMS_I2C_CFG_BUSx_ CALLBACK (x = 0 - 4)	comms_i2c_user_callback1(x) (x = 0 - 4)	Specify the name of the user callback function. When using rm_zmod4xxx, specify rm_zmod4xxx_callback(y) (y = 0).

Table 7-7 Communication Driver Settings for the RL78 Family MCU



7.2.4 RZ 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.

The following items and values can be specified.

	Table 7-8 Communication Middleware Settings for RZ Family		
Configurable Item	Value	Description	
Common			
Parameter Checking	Default (BSP)	Enable or disable the parameter check processing.	
	Enabled	When "Enabled" is selected, the project is	
	Disabled	built so that the generated code includes the parameter check processing.	
Module g_comr	ns_i2c_device0 I2C Commu	unication Device	
(rm_comms_i2d	c)		
Name	g_comms_i2c_device0	Specify the name of the module. A module name conforming to the C language standard can be specified.	
Semaphore Timeout	0xFFFFFFFF	For an RTOS project, specify the time of semaphore timeout.	
Slave Address	0x32	Specify the slave address. When rm_zmod4xxx is used, this value is automatically specified and cannot be modified.	
Address Mode	7-Bit	Specify the number of slave address bits. When rm_zmod4xxx is used, this value is automatically specified and cannot be modified.	
Callback	rm_zmod4xxx_comms_i2c_callback	Specify the name of the user callback function. When rm_zmod4xxx is used, this value is automatically specified and cannot be modified.	
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	0xFFFFFFF	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	Unuse	For an RTOS project, enable or disable the	
Bus	Use	recursive operation when blocking is enabled.	

Table 7-8 Communication Middleware Settings for RZ Family



7.3 I2C Driver Settings

7.3.1 RA Family

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

The following items and values can be specified.

Configurable Item	Value	Description
Common	1	1
Parameter Checking	Default (BSP)	Enable or disable the parameter check processing.
	Enabled	When "Enabled" is selected, the project is built
	Disabled	so that the generated code includes the parameter check processing.
DTC on Transmission and	Enabled	Specify whether to use the DTC for transmission and reception.
Reception	Disabled	
10-bit slave	Enabled	Specify whether to support 10-bit addressing for
addressing	Disabled	the slave address. When using rm_zmod4xxx, select "Disabled".
Module g_i2c_m	aster0 I2C Master Drive	
Name	g_i2c_master0	Specify the name of the module.
Channel	0	Specify the channel number to be used.
Rate	Standard	Specify the baud rate.
	Fast-mode	When using rm_zmod4xxx, select "Standard" or "Fast-mode".
	Fast-mode plus	
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	This item specifies the slave address of the device to be connected but the user does not need to make this setting because rm_comms_i2c overwrites it.
Address Mode	7-Bit	This item specifies the salve address mode for
	10-Bit	the device to be connected but the user does not need to make this setting because rm_comms_i2c overwrites it.
Timeout Mode	Short Mode	Specify the time of I2C bus timeout.
	Long Mode	
Callback	rm_comms_i2c_callback	The name of the user callback function is automatically specified by rm_comms_i2c.
Interrupt Priority Level	Priority 0 (highest)	Specify the interrupt priority level of the I2C bus
	Priority 1	driver.
	Priority 2	
	Priority 3	
		1
	Priority 4	

Table 7-9 r_iic_master Settings for RA Family



	Priority 6	
	Priority 7	
	Priority 8	
	Priority 9	
	Priority 10	
	Priority 11	
	Priority 12	
	Priority 13	
	Priority 14	
	Priority 15	
Pins	·	
SDA	Рххх	The pin numbers to be used by the driver are
SCL	Рххх	displayed. Use the "Pins" tabbed page to modify the pin configuration.

Table 7-10 r_sci_i2c Settings for RA Family

Configurable Item	Value	Description
Common		
Parameter Checking	Default (BSP)	Enable or disable the parameter check
	Enabled	processing. When "Enabled" is selected, the project is built
	Disabled	so that the generated code includes the parameter check processing.
DTC on	Enabled	Specify whether to use the DTC for
Transmission and Reception	Disabled	transmission and reception.
10-bit slave	Enabled	Specify whether to support 10-bit addressing
addressing	Disabled	for the slave address. When using rm_zmod4xxx, select "Disabled".
Module g_i2c0 I	2C Master Driver on r_s	
Name	g_i2c0	Specify the name of the module.
Channel	0	For an RTOS project, specify the time of semaphore timeout.
Slave Address	0x00	This item specifies the slave address of the device to be connected but the user does not need to make this setting because rm_comms_i2c overwrites it.
Address Mode	7-Bit	This item specifies the salve address mode for
	10-bit	the device to be connected but the user does not need to make this setting because rm_comms_i2c overwrites it.
Rate	Standard	Specify the baud rate.
	Fast-mode	Select "Standard" or "Fast-mode".
	Fast-mode plus	
SDA Output Delay (nano seconds)	300	Specify the SDA output delay time.
Noise filter setting	Use clock signal divided by 1 with noise filter	Specify the noise filter to be used for input signals.
	Use clock signal divided by 2 with noise filter	



SCL	Pxxx	Use the "Pins" tabbed page to modify the pin configuration.
SDA	Pxxx	The pin numbers to be used by the driver are displayed.
Pins	Duny	The sign purphers to be used by 0 a 12 and
D :	Disabled	
	Priority 15	_
	Priority 14	_
	Priority 13	_
	Priority 12	_
	Priority 11	_
	Priority 10	_
	Priority 9	
	Priority 8	
	Priority 7	
	Priority 6	
	Priority 5	
	Priority 4	
-	Priority 3	
enabled]	Priority 2	
Level [Only used when DTC is	Priority 1	the reception interrupt.
RX Interrupt Priority	Priority 0 (highest)	When using the DTC, specify the priority level of
	Priority 15	
	Priority 14	_
	Priority 13	_
	Priority 12	_
	Priority 11	_
	Priority 10	_
	Priority 9	_
	Priority 8	_
	Priority 7	_
	Priority 6	_
	Priority 5	_
	Priority 4	
	Priority 3	_
	Priority 2	_
Level	Priority 1	driver.
Interrupt Priority	Priority 0 (highest)	Specify the interrupt priority level of the I2C bus
Callback	rm_comms_i2c_callback	The name of the user callback function is automatically specified by rm_comms_i2c.
Callback		function.
Bit Rate Modulation	Enable	Enable or disable the bit rate modulation
	with noise filter	
	with noise filter Use clock signal divided by 8	-
	Use clock signal divided by 4	



7.3.2 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.

The following items and values can be specified.

Configurable Item	Value	Description
Configurations	Value	Description
Set parameter	System Default	Enable or disable the parameter check processing.
checking enable	Not	When "Include" is selected, the project is built so
	Include	that the generated code includes the parameter
		check processing.
MCU supported	Not supported	Specify whether to support the operation of channel
channels for CHx $(x = 0, -2)$	Supported	Х.
(x = 0 - 2) CHx RIIC bps(kbps)	400	Specify the bit rate.
(x = 0 - 2)	-00	Set to 400 or a smaller value.
Digital filter for CHx	Not	Specify the digital filter for input signals.
(x = 0 - 2)	One IIC phi	
	Two IIC phi	
	Three IIC phi	
	Four IIC phi	
Setting port setting	Not include port setting	Specify whether to include the pin function settings
processing	Include port setting	in the code to be generated.
Master arbitration lost	Unused	Specify whether to use the master arbitration lost
detection function for	Used	detection function.
CHx $(x = 0 - 2)$		
Address y format for	Not	This item specifies the slave address mode for
CHx	7 bit address format	slave address y but the user does not need to make
(x = 0 - 2, y = 0 - 2)	10 bit address format	this setting because any setting that is made here is
		overwritten by the setting in rm_comms_i2c.
Slave Address y for CHx	0x0025	This item specifies slave address y but the user does not need to make this setting because any
(x = 0 - 2, y = 0 - 2)		setting that is made here is overwritten by the
(x = 0 2, y = 0 2)		setting in rm_comms_i2c.
General call address	Unused	Specify whether to use the general call function.
for CHx	Used	
CHx RXI INT Priority	Level 1	Specify the priority level of the reception interrupt.
Level	Level 2	
(x = 0 - 2)	Level 3	
	Level 4	
	Level 5	
	Level 6	
	Level 7	
	Level 8	
	Level 9	
	Level 10	
	Level 11	
	Level 12	
	Level 13	

Table 7-11 r_riic_rx Settings for the RX Family MCU



	Level 14	1
	Level 15 (highest)	
CHx RXI INT Priority	Level 1	Specify the priority level of the transmission
Level	Level 2	interrupt.
(x = 0 - 2)	Level 3	
· · ·	Level 4	
	Level 5	
	Level 6	
	Level 7	
	Level 8	
	Level 9	
	Level 10	
	Level 11	
	Level 12	
	Level 13	
	Level 14	
	Level 15 (highest)	
CHx EEI INT Priority	Level 1	Specify the priority level of the error interrupt.
Level	Level 2	
(x = 0 - 2)	Level 3	
· · · · ·	Level 4	
	Level 5	
	Level 6	
	Level 7	
	Level 8	
	Level 9	
	Level 10	
	Level 11	
	Level 12	
	Level 13	
	Level 14	
	Level 15 (highest)	
CHx TEI INT Priority	Level 1	Specify the priority level of the transmission end
Level	Level 2	interrupt.
(x = 0 - 2)	Level 3	
· · · ·	Level 4	
	Level 5	
	Level 6	
	Level 7	
	Level 8	
	Level 9	
	Level 10	
	Level 11	
	Level 12	
	Level 13	
	Level 14	
	Level 15 (highest)	
Timeout function for	Unused	Specify whether to use the timeout function.
CHx	Used	opeony whether to use the timeout function.
(x = 0 - 2)		
Timeout detection time	Long mode	Specify the time for timeout detection.
L	.	· •



for CHx $(x = 0 - 2)$	Short mode	
Count up during low period of timeout detection for CHx (x = 0 - 2)	Unused Used	Specify whether to increment the counter for detecting a timeout while SCL is at the low level.
Count up during high period of timeout detection for CHx (x = 0 - 2)	Unused Used	Specify whether to increment the counter for detecting a timeout while SCL is at the high level.
Set Counter of checking bus busy	1000	Specify the counter value to be judged to represent the bus busy state.
Resources		
SDAx Pins	Checked	Specify the pins to be used. Select the checkboxes for the desired pins.
SCLx Pins	Checked	Gelect the checkboxes for the desired pins.

Table 7-12 r_sci_iic_rx Settings for the RX Family MCU

Configurable Item	Value	Description
Configurations		
Set parameter checking enable	System Default Not Include	Enable or disable the parameter check processing. When "Include" is selected, the project is built so that the generated code includes the parameter check processing.
MCU supported channels for CHx (x = 0 - 12)	Not supported Supported	Specify whether to support the operation of channel x.
SCI IIC bitrate (bps) for CHx (x = 0 - 12)	384000	Specify the bit rate. Set to 384000 or a smaller value.
Interrupt Priority for	Level 1	Specify the interrupt priority level.
CHx	Level 2	
(x = 0 - 12)	Level 3	
	Level 4	
	Level 5	
	Level 6	
	Level 7	
	Level 8	
	Level 9	
	Level 10	
	Level 11	
	Level 12	
	Level 13	
	Level 14	
	Level 15 (highest)	
Digital noise filter	Disable	Specify whether to use the digital noise filter.
(NFEN bit) for CHx $(x = 0 - 12)$	Enable	
Noise Filter Setting	The clock divided by 1	Specify the function of the digital noise filter.
Register (NFCS bit)	The clock divided by 2	
for CHx $(x = 0 - 12)$	The clock divided by 4	
(x = 0 - 12)	The clock divided by 8	



I2C Mode Register 1 (IICDL bit) for CHx (x = 0 - 12)	18	Specify the number of SDA output delay cycles.
Software bus busy check counter	1000	Specify the counter value to be judged to represent the bus busy state.
Setting port setting	Not include port setting	Specify whether to include the pin function settings
processing	Include port setting	in the code to be generated.
Resources		
SSDAx Pins	Checked	Specify the pins to be used.
SSCLx Pins	Checked	Select the checkboxes for the desired pins.



7.3.3 RL78 Family

Select "Serial" from the peripheral functions in the Code Generator, and the configurable items will be shown on the [Peripheral Functions] tabbed page.

The following items and values can be specified.

Table 7-13 Serial Settings for the RL78 Family MCU	Table 7-13	Serial	Settings	for the	RL78	Family	MCU
--	------------	--------	----------	---------	-------------	--------	-----

Configurable Item	Value	Description		
SAUx				
Channel				
Channel x	Unused	Specify the communication function of the		
	UARTxx	channel to be used.		
	CSIxx	If you are using r_zmod4xxx, select IICxx.		
	IICxx			
IICxx				
Transfer rate	1000000	Specify the bit rate.		
		If you are using r_zmod4xxx, specify 100000.		
Transfer end interrupt	High	Specify the priority level of the transfer end		
priority (INTIICxx)	Level1	interrupt.		
	Level2			
	Low			
Master transmission	Checked	Specify whether to use the callback function		
end		when master transmission ends.		
Master reception end	Checked	Specify whether to use the callback function when master reception ends.		
Master error	Checked	Specify whether to use the callback function		
		when a communication error occurs.		
IICAx				
Transfer mode				
Transfer mode	Unused	Specify the communication function of the		
	Single master	channel to be used.		
	Slave	Select "Single master".		
Setting				
Clock mode setting	fCLK	Specify the clock to drive counting.		
	fCLK/2			
Address	16	Specify the local address.		
Operation mode	Standard	Specify the operating mode.		
setting	Fast mode/Fast mode plus			
Transfer clock (fSCL)	100000	Specify the bit rate.		
		Set to 400000 or a smaller value.		
Communication end	High	Specify the priority level of the communication		
interrupt priority	Level1	end interrupt.		
(INTIICAx)	Level2			
	Low			
Master transmission end	Checked	Specify whether to use the callback function when master transmission ends.		
Master reception end	Checked	Specify whether to use the callback function		
		when master reception ends.		
Master error	Checked	Specify whether to use the callback function		
		when a communication error occurs.		
Generated stop	Checked	Specify whether to generate a stop condition in		
condition in master		the callback function.		
transmission/reception		Deselect the checkbox.		
end callback function				



7.3.4 RZ Family

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

The following items and values can be specified.

Configurable Item	Value	Description		
Common	1			
Parameter Checking	Default (BSP)	Enable or disable the parameter check processing. When "Enabled" is selected, the project is built		
	Enabled			
	Disabled	so that the generated code includes the parameter check processing.		
10-bit slave addressing	Enabled	Specify whether to support 10-bit addressing for the slave address.		
addressing	Disabled	When using rm_zmod4xxx, select "Disabled".		
Module g_i2c_m	naster3 I2C Master Drive	er on r_riic_master		
Name	g_i2c_master3	Specify the name of the module.		
Channel	3	Specify the channel number to be used.		
Rate	Standard	Specify the baud rate. When using rm_zmod4xxx, select "Standard" or		
	Fast-mode	"Fast-mode".		
	Fast-mode plus			
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.		
Noise Filter Stages	1	Removes noise below the 1 IIC cycle.		
	2	Removes noise below the 2 IIC cycle.		
	3	Removes noise below the 3 IIC cycle.		
	4	Removes noise below the 4 IIC cycle.		
Slave Address	0x00	This item specifies the slave address of the device to be connected but the user does not need to make this setting because rm_comms_i2c overwrites it.		
Address Mode	7-Bit	This item specifies the salve address mode for the device to be connected but the user does		
	10-Bit	not need to make this setting because rm_comms_i2c overwrites it.		
Timeout Mode	Short Mode	Specify the time of I2C bus timeout.		
	Long Mode			
Callback	rm_comms_i2c_callback	The name of the user callback function is automatically specified by rm_comms_i2c.		

Table 7-14 r_riic_master Settings for RZ Family



Interrupt Priority	0 (highest)	Specify the interrupt priority level of the I2C bus
Level	1	driver.
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
	10	
	11	
	12	
	13	
	14	
	15	



7.4 IRQ Driver Settings

7.4.1 RZ Family

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

The following items and values can be specified.

Configurable Item	Value	Description			
Common		•			
Parameter Checking	Default (BSP)	Enable or disable the parameter check			
C C	Enabled	processing. When "Enabled" is selected, the project is built			
	Disabled	so that the generated code includes the parameter check processing.			
Module g_extern	nal_irq7 External IRQ Dr				
Name	g_external_irq7	Specify the name of the module.			
Channel	7	Specify the channel number to be used.			
Trigger	Falling	Specify the trigger.			
	Rising	When using rm_zmod4xxx, select "Rising".			
	Both Edges				
	Low Level				
Callback	rm_zmod4xxx_irq_callback	The name of the user callback function is automatically specified by r_intc_irq.			
Pin Interrupt Priority	0 (highest)	Specify the interrupt priority level of the IRQ			
	1	driver.			
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	10				
	11				
	12				
	13				
	14				
	15				
Pins	I				
IRQ7	<unavailable></unavailable>	Pin settings are made in pin_data.c in the src folder. See "8.4.3 Changing sample code" for the setting method.			

Table 7-15 r_intc_irq Settings for RZ Family



8. Guide for Changing the Target Device

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

Before switching to a new device, import the original sample project for the current device to the workspace.

8.1 RA Sample Project

Use the following procedures to modify a sample project.

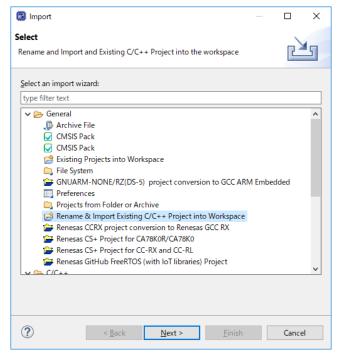
This section describes an example of modifying the sample project " ZMOD4xxx_RA6M4_NonOS" so that it can be used on the EK-RA2E1 board.

The description of PMOD1 is the procedure when using a board to which "OptionType6A" is applied.

8.1.1 Importing the Sample Project

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.



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.

Select Folder					×
← → × ↑ 📙 > e2_studio > works	pace	~	Ö	Search workspace	p
Organize 🔻 New folder				833 👻	?
e2_studio	^	Name		Date modified	^
workspace		ZMOD4XXX_RA6M4_Azu	re	6/16/2021 11:36 AM	6
.metadata		ZMOD4XXX_RA6M4_Free	RTOS	6/16/2021 2:55 PM	c .
ZMOD4XXX_RA6M4_Azure		ZMOD4XXX_RA6M4_Nor	nOS	6/16/2021 3:54 PM	2
ZMOD4XXX_RA6M4_FreeRTOS					
ZMOD4XXX_RA6M4_NonOS					
					~
	~	<			
Folder: ZMOD4XXX_R	A6M4	NonOS			
			- [Select Folder Cancel	

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

📴 Import				×
Rename & Imp	ort Project			5
Select a director	to search for existing Eclipse projects.			
Designation	ZMOD4XXX_RA2E1_NonOS			
Project name:				
Use default	location			
Location:	C:\Users\a5090534\e2_studio\workspace\Z	MOD4X	Browse	
	Create Directory for Project			
Choose file syst	em: default 🖂			
Import from:				
Select root of a sel	irectory: C:\Users\a5090534\e2_studio\workspi	ace\ ~	Browse	
◯ Select archiv	e file:	~	Browse	
Projects:				
ZMOD4XXX	RA6M4_NonOS (C:\Users\a5090534\e2_studio\w	orkspace\.	ZMOD4XXX	RA6
<				>
				-
Options	onfiguration output folders			
?	< Back Next > Fin	ish	Cance	I



8.1.2 Modifying Settings of the FSP Configurator

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

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.

		Restore Defa
evice Selecti	on	
CO unrion	3.0.0-rc1+20210426.9fd4d31a ~	Board Details
For versions	5/0/01/C1+20210420/9104031a	Evaluation kit for RA6M4 MCU Group
Board:	EK-RA6M4 V	Visit https://www.renesas.com/ra/ek-ra6m4 to get kit user's manual, quick start guide,
	Custom User Board (Any Device)	errata, design package, example projects, etc.
Device:	EEK-RA2L1 ····	
RTOS:	EK-RA2A1	
KIUS:		
	EK-RA2L1 EK-RA4M1	
	EK-RA4M1 EK-RA4M2	
	EK-RA4M3	
	FK-RA4W1	
	EK-RA6M1	
	EK-RA6M2	
	EK-RA6M3	
	EK-RA6M3G	
	EK-RA6M4	
	EK-RA6M5 FPB-RA2E1	
	RA2A1 TBB	
	RA4M1 TBB	
	RA4W1 ADK	
	RA6M1 TBB	
	RA6M2 TBB	
	RA6M3 PK	
	RSSK-RA2L1	
	RSSK-RA6T1	

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.

徽 *[HS300x_RA2E1_NonOS] FSP Configuration 🛛		
Clocks Configuration		Generate Project Content
		🔣 Restore Defaults
XTAL 20MHz	→ ICLK Div /1 ~	→ ICLK 48MHz
LOCO 32768Hz	→ PCLKB Div /2 ~	PCLKB 24MHz
MOCO 8MHz	Clock Src: HOCO V PCLKD Div /1 V	PCLKD 48MHz
SUBCLK 32768Hz		
HOCO 48MHz		
HOCO 48MHz V		
	CLKOUT Disabled V CLKOUT Div /1 V	→ CLKOUT 0Hz
Summary BSP Clocks @ Pins Interrupts Event Link	s 🙆 Stacks Components	



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 "RA6M4 EK" to the target board; appropriate pins are automatically assigned.

Pin Configuration					Generate Project Content
Select Pin Configuration			🔛 Export to CSV file 🛛	Configure Pin Driver Warnings	
RAGM4 EK	 Manage configuration 	05	Generate data:	_bsp_pin_cfg	
Pin Selection	🕀 😑 🖓 Pin G	onfiguration			😲 Cycle Pin Group
Sper fort test ✓ ▲ Pel ✓ ▲ Pel > → AsologACM > → SpetterGNCM > → SpetterGNCM </th <th></th> <th></th> <th>Value Value </th> <th></th> <th></th>			Value Value 		
Pin Function Pin Number					

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 Configuration" 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		
RA6M4 EK (Current) R7FA6M4AF3CFB.pincfg	Add	
RA2E1 EK	Remove	2
R7FA2E1A92DFM.pincfg	Rename.	
	Duplicat	te
	Merge to)
	Import.	•
	Export	
	OK	



However, the "Select Pin Configuration" assignment will apply the SPI communication pin settings that support PMOD Type 2A on the EK-RA2E1 board.

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

SCI2 is assigned to PMOD1 and SCI1 to PMOD2 on the EK-RA2E1 board.

I2C communication is assigned to P301 and P302 on PMOD1(OptionType6A), and it is assigned to P401 and P402 on PMOD2.

After automatic assignment of "Select Pin Configuration", reconfigure in "Pin Configuration".

in Configuration					O Generate Project Conten
elect Pin Configuration		📑 Export	to CSV file 🛛 🕅	Configure Pin Driv	er Warnings
RA2E1 EK	✓ Manage configur	rations	Generate data: g	_bsp_pin_cfg_2e1	l
in Selection 🗄 🕀 🖻	$\downarrow^{a}_{\mathbb{Z}}$ Pin Configuration				😲 Cycle Pin Group
Type filter text	Name Pin Group Select Operation Mode		Lock	Link	
 Peripherals Analog:ACMP Analog:ADC Analog:ANALOG Connectivity:IIC Connectivity:SCI SCI0 SCI1 SCI2 SCI9 	✓ Input/Output TXD1 RXD1 SCK1 CTS1 SDA1 SCL1	None None None None Vone Vone Vone Vone Vone			
	Usage: W	211 /hen using Simple I2C mode, ensure p /hen switching between I2C and other			rain.



IRQ exists as an interrupt signal pin of the sensor, but pin setting is not performed by automatic assignment of "Select Pin Configuration".

IRQ6 is assigned to PMOD2 on the EK-RA2E1 board.

To use PMOD2, P409 set to IRQ6 pin.

To use PMOD1(OptionType6A), IRQ cannot be used. Refer to section <u>8.1.4, Changing when not using IRQ</u>.

in Configuration					Generate Project Conter
elect Pin Configuration		📑 Ехро	ort to CSV file	Configure Pir	Driver Warnings
RA2E1 EK	✓ Manage configurations		Generate data:	g_bsp_pin_cfg	J_2e1
in Selection 📰 🕀 🖻	$\downarrow^{a}_{\mathbb{Z}}$ Pin Configuration				😲 Cycle Pin Group
Type filter text	Name	Value	Lock	Link	
✓ ✓ Peripherals	Operation Mode	Enabled			
	▲ Input/Output				
 Analog:ACMP Analog:ADC 	NMI	None		\Rightarrow	
> V Analog:ADC > V Analog:ANALOG	IRQ00	None		\Rightarrow	
> V Connectivity:IIC	IRQ01	None		\Rightarrow	
> ✓ Connectivity:NC	IRQ02	None		\Rightarrow	
Connectivity:SPI	IRQ03	None		\Rightarrow	
> Input:CTSU	IRQ04	None		\Rightarrow	
✓ ✓ Input:ICU	IRQ05	None		\Rightarrow	
 ICU0 	IRQ06	✓ P409	l di la constante di la consta	\Rightarrow	
> Input:KINT	IRQ07	None		\Rightarrow	
 Monitoring:CAC System:CGC 	<				
 System:DEBUG System:SYSTEM Timer: AGT 	Very Module name: ICU0 Usage: To use IRQ funct	ion with output or perip	oheral modes, cha	ange directly in	port dialog



In "Select Pin Configuration", it is automatically assigned as Type 2A.

Therefore, the pin corresponding to the RESET pin of Type 6A must be changed to I/O port pin.

To use PMOD1(OptionType6A), P101 set to I/O port pin.

To use PMOD2, P400 set to I/O port pin.

Select "Output mode (initial High)" for "Mode".

[ZMOD4410_RA2E1_NonOS] FSP Config	$_{ m juration}$ $ imes$		
Pin Configuration			Generate Project Conter
Select Pin Configuration		Export to CS	V file 🖺 Configure Pin Driver Warnings
RA2E1 EK	✓ Manage configurations	🗹 Genera	ate data: g_bsp_pin_cfg_2e1
Pin Selection 🗄 🕀 🏳	Pin Configuration		🔁 Cycle Pin Group
Type filter text ▼ ▲ Ports > ▲ P0 > ▲ P1 > ♥ P2 > ▲ P3 ♥ ♥400 ● ₱400 ♥ ₱410 ♥ ₱5	Name Symbolic Name Comment Mode Pull up IRQ Output type ✓ Input/Output P400 ✓ Module name: P400 Port Capabilities: AGT1: AGTI01 CAC0: CAC0:	Value LED3 Output mode (Initial High) None CMOS ✓ GPIO	Link
Pin Function Pin Number			



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.

In our example, it is "g_bsp_pin_cfg_2e1".

[ZMOD4410_RA2E1_NonOS] FSP Configuration	n X			- 8
Pin Configuration				O Generate Project Content
Select Pin Configuration		Export to CSV file	e 🔚 Configure Pin Driver Warr	lings
RA2E1 EK	 Manage configurations 	🗹 Generate d	lata: g_bsp_pin_cfg_2e1	
Pin Selection $ \blacksquare \ \boxdot \ label{eq:Pin_selection} \blacksquare \ \blacksquare \ label{eq:Pin_selection} \blacksquare \ \blacksquare \ \blacksquare \ label{eq:Pin_selection} \blacksquare \ \blacksquare \ \blacksquare \ label{eq:Pin_selection}$	Pin Configuration			😲 Cycle Pin Group
Type filter text	Name	Value	Link	
 Input:CTSU Input:ICU Input:KINT Monitoring:CAC System:CGC System:DEBUG System:SYSTEM Timers:AGT 	<			>



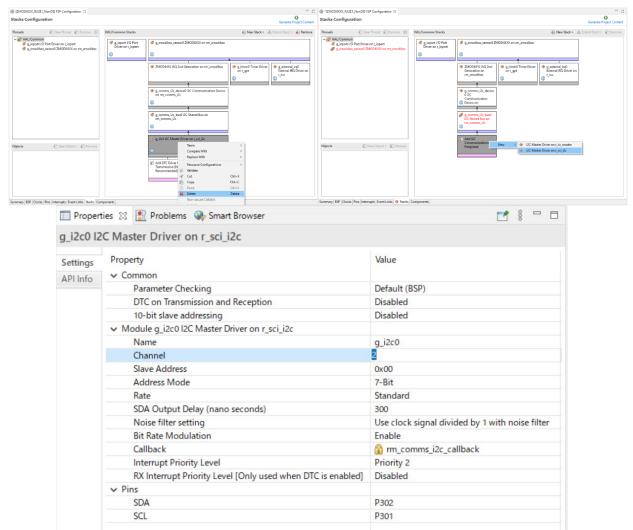
Modify the configuration of individual components in the "Stacks" tabbed page.

Modify the settings of r_iic_master or r_sci_i2c according to the specifications of the target board.

To use the pins of the IIC, delete the "I2C Master Driver on r_sci_i2c" stack and then add the "I2C Master Driver on r_iic_master" stack.

SCI2 is assigned to PMOD1 and SCI1 is assigned to PMOD2 on the EK-RA2E1 board.

To use PMOD1, set "Channel" to "2". To use PMOD2, set to "1".





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

In our example, it is "g_bsp_pin_cfg_2e1".

# *[ZMOD4410_RA2E1_NonOS] FSP Configuration ×			- 8
Stacks Configuration		G	enerate Project Content
Threads 🕢 🐑 New Thread 🔬 Remove 📄	HAL/Common Stacks	🐑 New Stack > 🚊 Extend	Stack > 🙀 Remove
 HAL/Common g_ioport I/O Port (r_ioport) g_zmod4xxx_sensor0 ZMOD4XXX Gas Sensor zmod4410_delay Timer, General PWM (r_gpt 		g_zmod4xxx_sensor0 ZMOD4XXX Gas Sensor (rm_zmod g) ZMOD4410 IAQ 2nd Generation (rm_zmod4xxx) g_comms_i2c_device0 I2C Communication Device (rm_comms_i2c) g_comms_i2c_bus0 I2C Shared Bus (rm_comms_i2c)	d4xxx)
Summary BSP Clocks Pins Interrupts Event Links St			
📮 Console 🔲 Renesas Debug Virtual Console 🔲 Pro	perties 🗙 🎋 Debug 🔗 Search		- 8
			🛃 🔚 🍸 🗔 🛷 🕴
g_ioport I/O Port (r_ioport)			
Settings Property API Info Common	Value		^
Parameter Checking	Default (BSP)		
 Module g_ioport I/O Port (r_ioport) 			
Name	g_ioport		
1st Port ELC Trigger Source	Disabled		
2nd Port ELC Trigger Source	Disabled		
3rd Port ELC Trigger Source	Disabled		
4th Port ELC Trigger Source	Disabled		
Pin Configuration Name	g_bsp_pin_cfg	_2e1	
✓ Pins			
SWCLK	P300		
SMDIO	D108		*



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

IRQ6 is assigned to PMOD2 on the EK-RA2E1 board.

To use PMOD2, set "Channel" to "6".

To use PMOD1(OptionType6A), IRQ cannot be used. Refer to section <u>8.1.4, Changing when not using IRQ</u>.

🔲 Properti	es 🗙 🗐 Console 📮 Renesas Debug Virtual Console 🕸 Debug 🔗 Search 🗔	🌢 Smart Manual 🔫 Progress 👒 Smart Browser 🛛 🗖 🗖
		📑 🐨 🖾 🔗 🖇
g_extern	al_irq0 External IRQ (r_icu)	
Settings	Property	Value
API Info	Common Parameter Checking	Default (BSP)
	 Module g_external_irq0 External IRQ (r_icu) 	
	Name	g_external_irq0
	Channel	6
	Trigger	Rising
	Digital Filtering	Disabled
	Digital Filtering Sample Clock (Only valid when Digital Filtering is Enabled)	PCLK / 64
	Callback	rm_zmod4xxx_irq_callback
	Pin Interrupt Priority	Priority 2
	✓ Pins	
	IRQ06	P409

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



8.1.3 Changing sample code

Open "hal_entry.c" (Non-OS) or "(sensor_name)_sensor_thread_entry.c" (FreeRTOS, Azure) and change the write process when resetting the sensor.

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

P101 is assigned to PMOD1(OptionType6A) and P400 to PMOD2 on the EK-RA2E1 board.

To use PMOD1(OptionType6A), specify P101. To use PMOD2, specify P400.

/* Reset ZMOD sensor (active low). Please change to the IO port connected to the RES_N pin of the ZMOD sensor on the customer board. */

R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_04_PIN_00, BSP_IO_LEVEL_HIGH);

R_BSP_SoftwareDelay(10, BSP_DELAY_UNITS_MILLISECONDS);

R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_04_PIN_00, BSP_IO_LEVEL_LOW);

R_BSP_SoftwareDelay(10, BSP_DELAY_UNITS_MILLISECONDS);

R_IOPORT_PinWrite(&g_ioport_ctrl, BSP_IO_PORT_04_PIN_00, BSP_IO_LEVEL_HIGH);

R_BSP_SoftwareDelay(10, BSP_DELAY_UNITS_MILLISECONDS);

Press [Generate Project Content] to generate files.

Build the project.

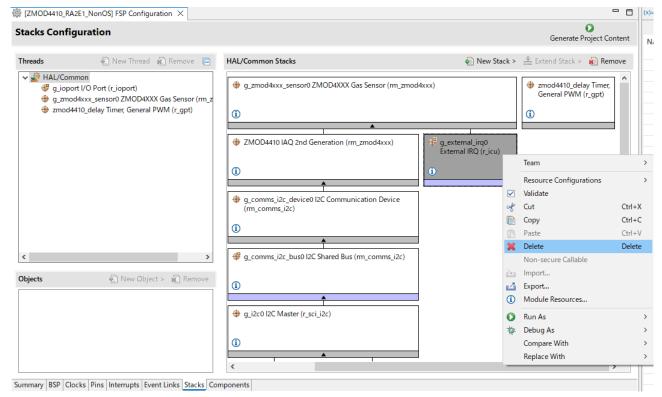
Select [Debug Configurations] from the menu and modify the debugger settings according to the specifications of the emulator to be connected to the target board.



8.1.4 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_(sensor_name).c" (Non-OS) or "(sensor_name)_sensor_thread_entry.c" (FreeRTOS, Azure) and change the value of "G_ZMOD4XXX_SENSOR0_IRQ_ENABLE" to "0".

/* TODO: Enable if you want to open ZMOD4XXX */
#define G_ZMOD4XXX_SENSOR0_IRQ_ENABLE (0)

8.1.5 Changing toolchain setting

If you want to use a toolchain other than the GCC ARM Embedded toolchain, copy RA_ZMOD4410.c and RA_ZMOD4510.c (Non-OS) or zmod4410_sensor_thread_entry.c, zmod4510_sensor_thread_entry.c, sensor_thread_common.c, and sensor_thread_common.c (FreeRTOS, Azure) from this project to create a new project.



8.2 RX Sample Project

Use the following procedures to modify a sample project.

This section describes an example of modifying the sample project "ZMOD4410_RX65N_NonOS" so that it can be used on the RSKRX231 board.

8.2.1 Importing the Sample Project

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.

S Import – 🗆	×
Select Rename and Import and Existing C/C++ Project into the workspace	5
Select an import wizard:	
type filter text	
 ✓ Constant Const	~
(?) < <u>Back</u> <u>Next</u> > <u>Finish</u> Cancel	

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.

-> • 🛧 📴 « xxxxxx	xx > e2_studio	> wo	orkspace >	√ Ö	Search workspa	ice ,	م
rganize 👻 New folder							8
workspace		^	Name	^		Date modified	
.metadata			.metadata			9/16/2021 6:32 PM	1
ZMOD4XXX_RA	6M4_Azure		ZMOD4XXX	RA6M4_Azure		9/16/2021 6:35 PM	1
ZMOD4XXX_RA	6M4_FreeRTOS		ZMOD4XXX	RA6M4_FreeRTC	os	9/16/2021 6:35 PM	1
ZMOD4XXX_RA	6M4_NonOS		ZMOD4XXX	RA6M4_NonOS		9/16/2021 6:35 PM	١
ZMOD4410 RL7	8G14 NonOS		ZMOD4410_	RL78G14_NonOS	5	9/16/2021 6:35 PM	١
ZMOD4410 RX6			ZMOD4410_	RX65N_Azure	·	9/16/2021 6:35 PM	1
ZMOD4410 RX6			ZMOD4410_	RX65N_FreeRTO	S	9/16/2021 6:35 PM	1
ZMOD4410_RX6	-		ZMOD4410_	RX65N_NonOS		9/16/2021 6:34 PM	1
			-	RL78G14_NonOS		9/16/2021 6:34 PM	
ZMOD4510_RL7	-			RX65N_FreeRTO		9/16/2021 6:34 PM	
ZMOD4510_RX6	55N_FreeRTOS		ZMOD4510_	RX65N_NonOS		9/16/2021 6:35 PM	1
ZMOD4510_RX6	5N_NonOS	~	<				
Folder:	ZMOD4410_RX65	iN_No	nOS				T
L				L.	Select Folder	Cancel	-



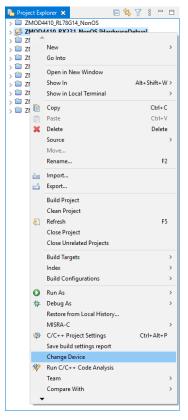
Enter the project name, select the original project for the current device, and press the [Finish] button.

📴 Import							\times
Rename & Impo Select a directory	-		g Eclipse projects				7
Project name: Z	MOD441	0_RX231_N	onOS				
✓ Use <u>d</u> efault le	ocation						
Location:	C:\U	sers\xxxxxxxx	x\e2_studio\wor	kspace\ZMOD441	D	B <u>r</u> owse	
	_		ory for Project				
Choose file s <u>y</u> ste	m: defa	ult \vee					
Import from:							
Select root di		C:\Users\)	oooooooox\e2_studi	o\workspace\Z ~		B <u>r</u> owse	
Select <u>archive</u>	e file:			~		B <u>r</u> owse	
<u>P</u> rojects:							
ZMOD4410_R	X65N_No	nOS (C:\Us	ers\xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	tudio\workspace\	ZMO	D4410_RX	(65N_
<							>
Options							
Keep build co	onfigurati	ion output	folders				
?		< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish		Cance	1



8.2.2 Changing the Device

Select the imported project from the project tree and right-click on int to open the context menu. Select "Change Device" from the menu.



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

Refactoring						×
Change Device Select the new of	e device for ZMOD4410)_RX231_NonOS				
Current Device:	R5F565NEDxFB RX65NEnvisionKit					
Target Board:	RSKRX231					~
				<u>Download</u>	additional bo	ards
Target Device:	R5F52318AxFP					
					Unlock Dev	ices
?		< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cance	el



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	-		×
Change Device Review the information provided in the list below. Click 'Ne view the next item or 'Finish'.	xt >' to		
Found problems		ł	3 6
(b) This change cannot be undone. Please make sure you b	ackup thi	s project	before
<			>
? < <u>B</u> ack <u>N</u> ext > <u>F</u> inish		Cance	

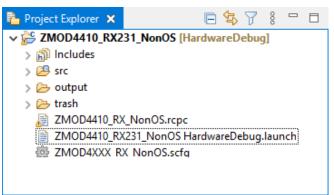
The changes you have made in the settings will be displayed. Press the [Finish] button to apply the changes to the project.

Befactoring			×
Change Device			_
The following changes are necessary to perform the refactoring.		- 27	
Changes to be performed		₽	7 -
🗸 🗹 🚰 Change Device for ZMOD4410_RX231_NonOS			^
🗸 🔽 🛃 Launch Configurations			
ZMOD4410_RX231_NonOS HardwareDebug			
> 🖂 😓 Build Settings			
🗹 🔂 Project Files			~
No preview available			
(?) < <u>B</u> ack <u>Next</u> > <u>Finish</u>	h	Cance	2



8.2.3 Modifying Settings of the Smart Configurator

On the project tree, double-click on the .scfg file of the imported project in which the target device has been changed; the Smart Configurator window will open.



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

u 🛆

Set up the clocks in the "Clocks" tabbed page according to the specifications of the target board to be used.

Clocks configuration	Clocks configuration
	Model and American difference No.med D000000 Model and Model Model and American difference Model and Model Model and Model Model and Model Model and Model and Model Model and Model Model and Model Model and Model Model and Model and Model Model and Model Model and Model Model and Model Model and Model and Model Model and Model Model and Model Model and Model Model and Model and Model and Model Model and Model Model and Model Model and Model Model and Model
In State and a second s	Alexandra de la construir
10.000 mm (10.000 mm (21 000 000 0000 0000 0000 0000 0000 000
Overview Board Clocks System Components Pins Interrupts	Overview Board Clocks System Components Pins Interrupts

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

As SCI8 is assigned to PMOD on the RSK RX231 board, change the setting of "MCU supported channels for CH2" to "Not supported" and "MCU supported channels for CH8" to "Supported" in r_sci_iic_rx.

Check the settings of "SSCL8 Pin" and "SSDA8 Pin" for "SCI8" under "Resources".



omponents 🛛 👌 🖃 🖶 🕈 🔻	Configure		① Components 🕴 📄	🗄 🖆 👻 Configure		(
₩ ₩ type filter test ₩ ₩ Batep ₩ ₩ Genetic ₩ Usep ₩ ₩ Usep ₩ ₩ Usep ₩ Usep ♥ Usep ♥ Usep ♥ Usep ♥ Usep ♥ Usep ♥ Usep ♥ Usep	Property ♥ Corrigutions # 64 grammeter checking enable # 64 grammeter checking enable # 64 grammeter checking enable # 64 grammeter checking enable # 64 grammeter checking enable # 64 grammeter checking # 64 grammeter checking enable # 64 grammeter checking # 64 grammeter checking enable # 64 grammeter checking # 64 grammeter checking enable # 64 grammeter checking # 64 grammeter checking enable # 64 grammeter checking # 64 grammeter checking enable # 64 grammeter checking # 64 grammeter checking enable 64 grammeter checkingrameter checking <	Value Include Include Include Include Include Inclusponted Inclusponted Not supported Secono	v to state v to		Value Used Used Used Used Used Used Used Use	

And IRQ3 is assigned to PMOD on the RSK RX231 board, check the settings of "IRQ3 Pin" for "Interrupts" under "Resources" in r_irq_rx and change the settings of "IRQ number for ZMOD4XXX sensor device0" to "IRQ3" in r_zmod4xxx_rx.

Components 🛛 🖄 🖃 🖶 🌩 🔻	Configure	0	Components 🛛 🖗 🔁 🗮 🊔 🔻	Configure		(
witereil Baard Clocks I Smeen Com	Property ● Configurations # B cale parameter checking enable # MCD supported channels for CH # MCD supported channels for CH	Value A Include A Not supported B Not supported B Not supported B	Components Image: The first set of the set of t	Property \$50.0 Fm \$50.0 Fm \$50.0 Fm \$50.1 Fm \$50.0 Fm \$50.0 State \$50.0 Fm	When Used		
	# SOIL: Selver, Regist For CH # SOIL: Selver, Regist For CH # SOIL: Selver, Regist For CH # SOIL: Selver, Regist For CH # SOIL: Selver, Regist For CH # SOIL: Selver, Regist For CH # SOIL: Selver, Regist For CH # SOIL: Selver, Regist For CH # SOIL: Selver, Regist For CH # SOIL: Selver, Regist For CH # SOIL: Selver, Regist For CH # Soil: Selver, Regist For CH # Soil: Selver, Regist For CH # Soil: Selver, Regist For CH I = Segretified. I = Segretified.	15000 10000 100000 100000 100000 1000000	Dreview Baard Clocks System Comp	SDAD Pin SDAD Pin SDC12 Pin SDC12 Pin SDCA12 Pin SDCA12 Pin SDCA12 Pin	Ured Ured Ured		



Open the "Pins" tabbed page and check that functions are assigned to the SCI8 pins and IRQ3 pins in the "Pin function" panel.

Hardware Resource 🛛 🕀 🖃 🖧 🚵	Pin Functio	n		े 🕄 🔣 🗉	1 2 2	Hardware Resource 🛛 🕀 🖃 🖓 🚵	Pin Functi	on		ି 🕹 🔣 🗄	1 2 2
Type filter text	type filter	text (" = any string,	? = any character)	All	~	Type filter text	type filte	r text (* = any stri	ng, ? = any character)	All	~
Q. Pot oxput enable 2 ∧ Q. Ratine dock ∧ Q. Ratine dock ∧ V 700 0 > V 700 0 >	Enabled	RTS8# RXD8 SCK8 SMISO8 SMOSI8 SS8# A SSCL8 SS8# SSDA8	Anigeneed A for assigned I for assigned		None None None IO IO	All Cost All All Cost All All	Enabled	Function (RQ0 (RQ1 (RQ1 (RQ2 (RQ2) (RQ4 (RQ5 (RQ6 (RQ6 (RQ7 (RQ7 (RQ7) (RQ7	Not assigned Pa3.MTDCc0/TMR3/POE3#/TIOCD0/RXD6/SMSOE Not assigned Not assigned Not assigned Not assigned	Pin Number / Not assigned / Not assigned / Not assigned / 17 / Not assigned / Not assigned	None None None None None None

As the use of PMOD Type 2A (extended SPI) is specified in the RSK RX231 board information, a warning message will appear when I2C is used, but this does not produce any problems.

To connect a sensor board, a board for converting PMOD Type 2A to PMOD Type 6A is necessary.

Press the [Generate Code] icon to generate code.

erview information				Generate Code	Generate Report
General Information					0
is editor allows you to modify the settings stored in confi	guration file (.so	:fg)			
pard					
low board and device selection					
ocks Iow clock configuration			Application under development	-Components	
			Middleware		
omponents			Device RTOS		
low software component selection and configuration				- Pins	
15			10000000000000	- Fills	
Iow general pin configuration and pin configuration for set	lasted coff.				
ion general princonngaration and princonngaration for se	freeten sonthore	component			
ick here to get more information on <u>User's Manual, Relea</u> Current Configuration	se Note Applici	ation Notes Tool News			
lected board/device: R5F52318AxFP (ROM size: 512 Kbyte:	s , RAM size: 64	Kbytes, Pin count: 100)			
enerated location (PROJECT_LOC\): src\smc_gen	-		dit		
ected components:					
Component	Version	Configuration			
Board Support Packages. (r_bsp)	6.21	r_bsp(used)			
CMT driver (r_cmt_rx)	4.90	r_cmt_n(used)			
IIC Communication Driver Interface Middleware (r_co	1.10	r_comms_i2c_rx(u	sed)		
IRQ Driver (r_irq_nx)	3.80	r_irq_rx(used)			
RIIC Multi Master I2C Driver. (r_riic_rx)	2.49	r_riic_rx(used)			
Simple IIC Driver. (r_sci_iic_rx)	2.49	r_sci_iic_nx(used)			
 ample no priver (Laci nc tx) 					
ZMOD4XXX Sensor Middleware (r_zmod4xxx_rx)	1.00	r_zmod4xxx_rx(use	d)		

Build the project.

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.

8.2.4 Changing toolchain setting

If you want to use a toolchain other than the CC-RX toolchain, copy RX_ZMOD4410.c and RX_ZMOD4510 (Non-OS), or main.c and zmod4410_sensor_thread_entry.c and zmod4510_sensor_thread_entry.c (FreeRTOS), or zmod4410_sensor_thread_entry.c, zmod4510_sensor_thread_entry.c, sensor_thread_common.c, and sensor_thread_common.c (Azure) from this project to create a new project.



8.3 RL78 Sample Project

Changing the target device within the RL78 family requires creating a new project.

This section describes an example of creating a new project that can be used on a RL78/G1A custom board.

8.3.1 Creating a New Project

Select [File] \rightarrow [New] \rightarrow [Renesas C/C++ project] \rightarrow [Renesas RL78] from the menu. Select the template "Renesas CC-RL C Executable Project" and press the [Next] button.

🗐 New C/C	rerojett			×	
emplates f	or Renesas RL				
All		GCC for Renesas RL78 C/C++ Library		+	
2/C++	RL78	A C/C++ Library Project for Renesas RL78 GCC for Renesas RL78 Toolchain.			
	RL78	LLVM for Renesas RL78 C/C++ Exect A C/C++ Executable Project for Renesas in LLVM for Renesas RL78 Toolchain.			1
	RL78	LLVM for Renesas RL78 C/C++ Librai A C/C++ Library Project for Renesas RL78 for Renesas RL78 Toolchain.			
	RL78	Renesas CC-RL C Executable Project A C Executable Project for Renesas RL78 of CCRL toolchain.		e	
	RL78	Renesas CC-RL C Library Project A C Library Project for Renesas RL78 using toolchain	g the CC		>
0	< P	Back Next > Finish		Cancel	

Enter the project name (example: "ZMOD4410_RL78G1A_NonOS") and press the [Next] button.

0			×
	-RL Executable Project		-\$
New Renesas CC-	RL Executable Project		1
Project name: Z	MOD4410_RL78G1G_NonOS		
Use <u>d</u> efault le	ocation		
Location:	C:\Users\a5090534\e2_studio\workspace\ZMOD4410_RL78G1G_NonOS	Brows	e
	Create Directory for Project		
Choose file s <u>y</u> ste	m: default \vee		
Working sets			
Add projec <u>t</u>	to working sets	Ne <u>w</u>	
Working sets:	~	S <u>e</u> lect	
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cano	el



Change "Target Device" to a desired device and press the [Next] button.

8							×	🕲 – 🗆 X
Device Selection You can filter devices by	/ regular exp	pression						New Renesas CC-RL Executable Project Select toolchain, device & debug settings
Search Device								Toolchain Settings
Device > RL78 - G14 > RL78 - G15 ✓ RL78 - G1A > RL78 - G1A 25pin > RL78 - G1A 32pin	RAM	ROM	Pin	RTOS	Smart Co	Peripher	^	Language: C C++ Toolchain: Renesas CC-RL V1.12.00 Manage Toolchains
 > RL78 - G1A 48pin > RL78 - G1A 48pin RSF 10ELC RSF 10ELC RSF 10ELC > RL78 - G1C > RL78 - G1C > RL78 - G1C > RL78 - G1G > RL78 - G1H > RL78 - G1M > RL78 - G1M > RL78 - G1N 	2 KB 3 KB 4 KB	32 KB 48 KB 64 KB	64 64 64		× × × × ×	✓ ✓ ✓		Device Settings Configurations Target Board: Custom Target Device: RSF10ELE Indian: Unlock Devices Endian: Little Project Type: Default
?					OK	Cancel		(?) < Back Next > Finish Cancel

Select the checkbox for "Use Peripheral Code Generator" and press the [Next] button.

0			×
New Renesas CC-RL Executable Project Select Coding Assistant settings			2
Use Smart Configurator Use Smart Configurator Use Peripheral Code Generator The e2 studio peripheral code generator automatically generates programs (device drivers) for MCU peripheral to times, serial interface, A/D converters, DMA controllers, etc.) based on settings entered via a graphical user in functions are provided as application programming interfaces (APA) and are not limited to initialization of peripheral Log Timer DMA Clock DMA Clock	terface	(GUI).	
(?) < Back Next > Einish		Cance	4

Press the [Finish] button.

0				_		×		
	ew Renesas CC-RL Executable Project ummary of project "ZMOD4410_RL78G1A_NonOS"							
TOOLCHAIN NAME : TOOLCHAIN VERSION : GENERATION FILES : generate¥kcstart.asm generate¥hdwinit.asm generate¥init.int.asm generate¥iodefine.h	Renesas CC-RL v1.12.00					~		
?		< Back	Next >	Finish	Cance	I		



8.3.2 Settings of the Code Generator

Modify the pin assignment in the "Pin assignment" tabbed page for "Common/Clock Generator" according to the specifications of the target board to be used.

归 *Peripheral F	unctions X 🚽	🔮 Code Preview			🐻 Generate Code	💿 👔 🖇 🗖 🗖
Pin assignment	Clock setting	Block diagram	On-chip debug set	ting (Confirming reset source	Safety function ^
– Pin assignment	setting					
🗌 PIOR0 bi	t = 1					
🔄 PIOR1 bi	t = 1					
Once the pin them later. A	i assienments h new project mi Fix se	ust be created to	is not possible to c change the settings	hange S.		
Pin		Function				
P70		KR0				
P71		KR1				
P72		KR2				
P73		KR3				
P74		KR4				
P75		KR5				
P76		KR6				
P77		KR7				
P05		KR8				
P06		KR9				
						~
<						>

Modify the clock settings in the "Clock setting" tabbed page for "Common/Clock Generator" according to the specifications of the target board.

📱 Peripheral Fun 🔀 🛛 🛃 Code Preview	👮 Device Top View 📃 Device List View 🔲 Pro	perties 📲 FIT Configura 👘 🗇
		👸 Generate Code 🗕 🖇
Pin assignment Clock setting Block diag	rram On-chip debug setting Confirming reset sour	ce Safety functions
Operation mode setting		
● High-speed main mode 4.0 (V) ≤ VE		
○ High-speed main mode 3.6 (V) ≤ VE		
○ High-speed main mode 2.7 (V) ≤ VE	DD ≤ 5.5 (V)	
○ Low-speed main mode 2.7 (V) ≤ VD	D ≤ 5.5 (V)	
Main system clock (fMAIN) setting		
High-speed OCO (fIH)	◯ High-speed system clock (fMX)	
High-speed OCO clock setting		
Operation F	requency 48 (fHOCO=48, fIH=24) V (MH	z)
High-speed system clock setting		
Operation		
 X1 oscillation (fX) 	 External clock input (fEX) 	
Frequency	5 (MH	z)
Stable time	2^18/fX 🗸 52428.8 (µs)	
Low-speed oscillation clock (fIL) setting _		
Frequency	15 (kH	2)
Interval timer operation clock/Timer RJ co	unt source setting	
Interval timer operation clock/Timer R.		Hz)
CPU and peripheral clock setting CPU and peripheral clock (fCLK)	fIH ~ 24000 (kH	
or o and peripricital clock (rock)		



Select "Used" for "On-chip debug operation setting" in the "On-chip debug setting" tabbed page for "Common/Clock Generator".

💯 Peripheral Fun 🛛 📓 Code Preview 👮 Device Top View 👮 Device List View 🔲 Properties 💯 FIT (Configura	
		3 8
Pin assignment Clock setting Block diagram On-chip debug setting Confirming reset source Safety func		^
- On-chip debug operation setting		
O Unused Used		
- RRM function setting		
O Unused		
- Security ID setting		_
Use Security ID		
Security ID 0x0000000000000000000000000000000000		
-Security ID authentication failure setting		
◯ Do not erase flash memory data		
Erase flash memory data		
		~
<		>

To use the serial array unit, set the channel assigned to PMOD on the target board to "IICxx" in the "Serial Array Unit" or "Serial" tabbed page.

Note: The corresponding pin must be selected as N-ch by "Port".

🕎 *Peripheral Fu	🔀 📑 Code P	review	Device Top View	🕎 Device List View	Properties	🕎 FIT Configura		
						🐻 Generate Code	0	8
<u>Serial Array Unit 0</u>								^
Channel UARTO	UART1 CSI00	IC00						
- Function								
Channel 0 🛄	× 00							
Channel 1 Ur	iused \checkmark							
Channel 2 Ur	iused 🗸 🗸							
Channel 8 Ur	used 🗸							
								~
<							>	•



In the tabbed page for IICxx enabled in the serial array unit, set "Transfer rate" to 400000 or 100000, set "Transfer end interrupt priority" to a desired value, and enable all functions under "Callback function setting". Note : When using a serial array unit, the Nch open drain of the pin to be used is set automatically. If an error icon on the port was displayed, open the Ports tab and check the port settings.

🕎 *Peripheral Fu 🔀 🦪 Code Preview 👮 Dev	ice Top View 📲 Device List View 🔲 P		
		🐻 Generate Code	3
Serial Array Unit 0			^
Channel UART0 UART1 CSI00 IC00			
- Transfer rate setting			
Transfer rate	100000 v (bp	s) (Actual value: 100000)	
-Interrupt setting			
Transfer end interrupt priority (INTIIC00)	Low ~		
-Callback function setting			
Master transmission end	Master reception end	Master error	
			~
<			>

To use the serial interface IICA, select "Single master" in the "Transfer mode" tabbed page for the channel assigned to PMOD on the target board in the "Serial Interface IICA" or "Serial" setting window.

💯 *Peripheral F	u 🛙	ode Preview	🕎 Device Top View	📃 Device List View	Properties	💯 FIT Configura		
1.000						🐻 Generate Code	0	8
Transfer mode	Setting							^
O Unused								
Single ma	aster							
◯ Slave								
								۷
<							>	



In the "Setting" tabbed page for the channel set to the single master, set "Operation mode setting" to either a combination of "Fast mode" and "400000" or a combination of "Standard" and 100000, set the interrupt priority to a desired level, enable all functions under "Callback function setting", and disable "Callback function enhanced feature setting".

👮 *Peripheral Fu 🛛 🛒 Coo	de Preview 📲 Device Top View	🕎 Device List View	Properties	💯 FIT Configura		
				🐻 Generate Code	0	8
Transfer mode Setting						^
-Clock mode setting						
🔾 folk 🚺	ICLK/2					
-Local address setting						
Address	16					
-Operation mode setting						
 Standard 	◯ Fast mo	de/Fast mode plus	Digital filter	on		
Transfer clock (fSCL)	100000	(bps) (/	Actual value: 99	173.554)		
_Interrupt setting Communication end interrup -Callback function setting ⊡ Master transmission end		~ 	ter error			
-Callback function enhanced fea						
<					>	

In the "General setting" tabbed page for a desired channel of the timer array unit or a desired TAU of the timer, select "Interval timer" under "Functions".

🕎 *Peripheral Fo	J 🔀 📓 Code Preview 🛛 🗮 Device Top View 月 Device Li		
		🛐 Generate Code 🛛 🙆	00
	Channel 0 Channel 1 Channel 2 Channel 3		^
- Functions			
Channel 0	Interval timer 🗸 🗸		
Channel 1	Unused ~		
Channel 2	Unused ~		
Channel 3	Unused ~		
<		>	~



In the page for the channel set to the interval timer, set "Interval value" to "100 µs", enable timer interrupts, and set the interrupt priority to a desired level.

🕎 Peripheral Functions 🗙	🐻 Generate Code	0	00 -	'E	3
TAU0 TAU1 TMRJ0 TMRD0 TMRD1 TMRG0					^
General setting Channel 1 Channel 2 Channel 3					
-Interval timer setting				-	
Interval value (16 bits) 100 µs V (Actual value: 100	0				
Generates INTTM00 when counting is started					
-Interrupt setting					
Priority Low V					
					1
					v
<				>	

Set IRQ pin on PMOD in the "External Interrupt" tabbed page for "Interrupt" according to the specifications of the target board.

📲 *Peripheral Func	tions ×					🐻 Generate Code	8		
External Interrupt	Key input inter	rupt							^
_INTP0 setting									_
INTP0	Valid edge	Falling	\sim	Priority	Low	\sim			
_INTP1 setting									-
INTP 1	Valid edge	Falling	\sim	Priority	Low	\sim			
_INTP2 setting									-
INTP2	Valid edge	Falling	\sim	Priority	Low	\sim			
_INTP3 setting									-
INTP3	Valid edge	Falling	\sim	Priority	Low	\sim			
_INTP4 setting									-
INTP4	Valid edge	Falling	\sim	Priority	Low	\sim			
_INTP5 setting									-
INTP5	Valid edge	Falling	\sim	Priority	Low	\sim			
_INTP6 setting									-
INTP6	Valid edge	Falling	\sim	Priority	Low	\sim			
_INTP7 setting									-
INTP7	Valid edge	Falling	\sim	Priority	Low	\sim			
_INTP8 setting									-
INTP8	Valid edge	Falling	\sim	Priority	Low	\sim			
_INTP9 setting		_							-
INTP9	Valid edge	Falling	\sim	Priority	Low	\sim			¥
<								>	



Set RESET pin on PMOD in the "Port (x)" tabbed page for "Port" according to the specifications of the target board.

💹 Peripheral	Functions	🗙 🛃 Code	Preview		🐻 Gene	erate Code	8	- 6	3
Port0 Port1	Port2	Port3 Port4	Port5 Port6	Port7 Port12	Port13 Port14	Port15			^
Unused P01	l O In	🔾 Out	Pull-up	TTL buffer		1			
Onused P02	d 🔾 In	⊖ Out	🗌 Pull-up	TTL buffer		1			
Onused P03	d OIn	🔾 Out	🗌 Pull-up		🗌 N-ch	1			
O Unused P04	d OIn	🔾 Out	🗌 Pull-up	TTL buffer	🗌 N-ch	1			
Unused P05	d OIn	🔾 Out	🗌 Pull-up	TTL buffer	🗌 N-ch	1			
Unused P06	d OIn	🔾 Out	🗌 Pull-up			1			
Unused	d OIn	Out	🗌 Pull-up			1			
<								>	Y

Press the [Code Generate] button to generate code.

8.3.3 Modifying the Generated Code

Perhaps Code Generator output destination different from this sample software, because Code Generator version differs depending on the MCU used.



Open r_cg_sau_user.c, r_cg_iica_user.c, or r_cg_serial_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 iic00 callback master error
* Description : This function is a callback function when IIC00 master
err
* Arguments : flag -
             status flag
* Return Value : None
static void r iic00 callback master error(MD STATUS flag)
{
  /* Start user code. Do not edit comment generated here */
  rm comms i2c bus0 callback(true);
  /* End user code. Do not edit comment generated here */
}
* Function Name: r iic00 callback master receiveend
* Description : This function is a callback function when IIC00 finishes
* Arguments
          : None
* Return Value : None
              *****
static void r iic00 callback master receiveend(void)
  /* Start user code. Do not edit comment generated here */
  rm comms i2c bus0 callback(false);
  /* End user code. Do not edit comment generated here */
}
* Function Name: r iic00 callback master sendend
* Description : This function is a callback function when IIC00 finishes
          : None
* Arguments
* Return Value : None
static void r_iic00_callback_master_sendend(void)
{
   /* Start user code. Do not edit comment generated here */
  rm comms i2c bus0 callback(false);
  /* End user code. Do not edit comment generated here */
}
```

Open t_cg_tau_user.c or r_cg_timer_user.c and add the following code.

Declaration of external for the (sensor_name)_delay_callback() function:

Addition of the call of the (sensor_name)_delay_callback() function to the timer interrupt callback function:

Open t_cg_tau.c or r_cg_timer.c and add the following code.

Define the R_TAU0_Channel0_Reset() function in the user code description part:

```
void R_TAU0_Channel0_Reset(void)
{
     /* function not supported by this module */
}
```

Open t_cg_tau.h or r_cg_timer.h and add the following code.

Declaration of prototype for the R_TAU0_Channel0_Reset() function:



Open r_cg_main.c or r_main.c and add the following code.

Definition for including r_bsp_common.h:

Declaration of prototype for each function:

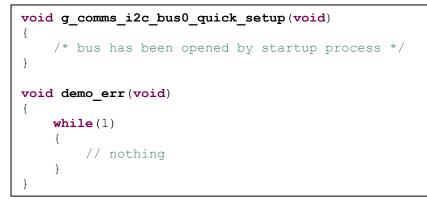


Addition of the following code to the main() function:

Modify RESET pin designation according to the target board to be used.

```
/* Open the Bus */
g_comms_i2c_bus0_quick_setup();
/* Reset ZMOD sensor (active low). Please change to the IO port
connected
 to the RES N pin of the ZMOD sensor on the customer board. \star/
P5 = _00_Pn7_OUTPUT_0 | _00_Pn6_OUTPUT_0 | _00_Pn5_OUTPUT_0 |
       _00_Pn4_OUTPUT_0 | _00_Pn3_OUTPUT_0 | _00_Pn2_OUTPUT_0 |
_02_Pn1_OUTPUT_1 | _00_Pn0_OUTPUT_0;
R BSP SoftwareDelay (10, BSP DELAY MILLISECS);
P5 = _00_Pn7_OUTPUT_0 | _00_Pn6_OUTPUT_0 | _00_Pn5_OUTPUT_0 |
_00_Pn4_OUTPUT_0 | _00_Pn3_OUTPUT_0 | _00_Pn2_OUTPUT_0 |
_00_Pn1_OUTPUT_0 | _00_Pn0_OUTPUT_0;
R_BSP_SoftwareDelay(10, BSP_DELAY_MILLISECS);
P5 = _00_Pn7_OUTPUT_0 | _00_Pn6_OUTPUT_0 | _00_Pn5_OUTPUT_0 |
_00_Pn4_OUTPUT_0 | _00_Pn3_OUTPUT_0 | _00_Pn2_OUTPUT_0 |
_02_Pn1_OUTPUT_1 | _00_Pn0_OUTPUT_0;
R BSP SoftwareDelay(10, BSP DELAY MILLISECS);
/* Open ZMOD4XXX */
g zmod4xxx sensor0 quick setup();
while (1U)
{
     start zmod4410 demo();
}
```

Define of the g_comms_i2c_bus0_quick_setup() function and the demo_err() function:

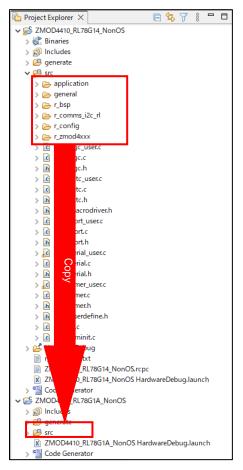




8.3.4 Modifying Sample Source Files

Right-click on the "application" "general" "r_bsp" "r_comms_i2c_rl" "r_config" "r_zmod4xxx"folder in the project tree of the sample project "ZMOD4410_RL78G14_NonOS", "ZMOD4450_RL78G14_NonOS" or "ZMOD4510_RL78G14_NonOS" and select "Copy" from the context menu.

Then, right-click on the "src" folder in the newly created project and select "Paste" from the context menu to paste the copied files to the folder.





Open r_comms_i2c_rl_config.h in the "r_config" folder and modify the values of the following definitions.

- COMMS_I2C_CFG_BUSx_DRIVER_TYPE
- COMMS I2C CFG BUSx DRIVER CH

When channel 0 of the serial array unit is used:

```
/* SPECIFY DRIVER TYPE, CHANNEL NO. */
/* For Bus No.0 */
#define COMMS_I2C_CFG_BUS0_DRIVER_TYPE (COMMS_DRIVER_SAU_I2C) /*
Driver type of I2C Bus */
#define COMMS_I2C_CFG_BUS0_DRIVER_CH (0) /* Channel No. */
```

When channel 0 of the serial interface IICA is used:

For the other definitions, refer to section 7, Configuration Settings.

When "serial array unit", "serial interface IICA", or "timer array unit" is used as a peripheral function name in the code generator, modify the sample source code as follows.

src/general/r_smc_entry.h

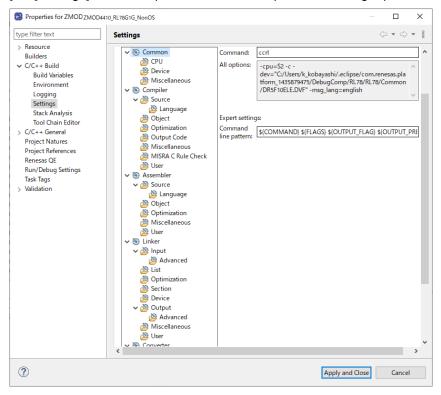
Modify "r_cg_serial.h" to "r_cg_sau.h" or "r_cg_iica.h".

Modify "r_cg_timer.h" to "r_cg_tau.h".



Open the "Properties" window for the project.

Select [C/C++ Build] \rightarrow [Settings] in the "Properties" window to open the "Settings" panel.

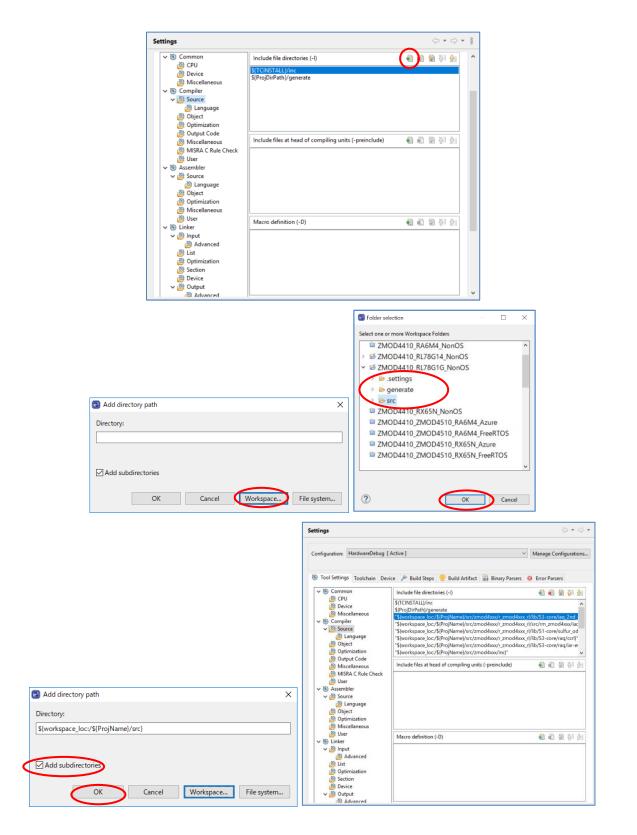




Select [Compiler] \rightarrow [Source] in the "Tool Settings" tabbed page and press the [Add] icon.

Press the [Workspace] button in the [Add directory path] dialog box and a list of projects will appear. Select the "src" folder for the newly created project in the list and press the [OK] button.

Select the checkbox for "Add subdirectories" and press the [OK] button.





Select [Compiler] \rightarrow [Source] \rightarrow [Language] in the "Tool Settings" tabbed page and change the setting of "Language standard of C language" to "C99 language standard".

Press the [Apply and Close] button to close the "Properties" window.

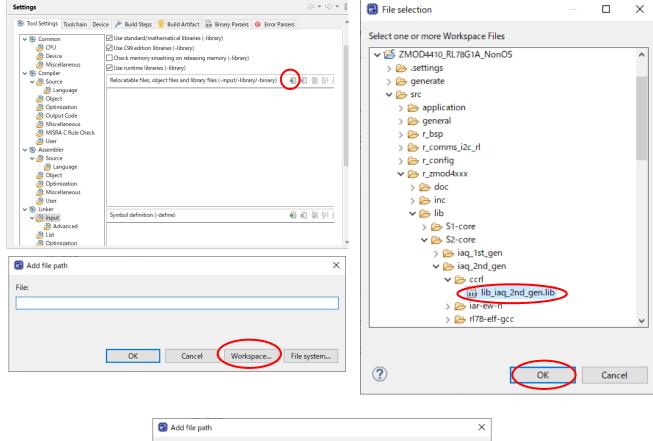
Settings			▼
Configuration: HardwareDebug [/	Active]	~	Manage Configurations
Tool Settings Toolchain Device	ce 🎤 Build Steps 🚇 Build Artifact 📓	Binary Parsers 😣 Error Parse	rs
 Common CPU Device Miscellaneous Compiler Compiler	Check function without prototype declar Maximum size of a variable (-large_variable) Allow nested comments (-nest_commen Character encoding (-character_set) Language standard of C language (-lang) Compile strictly according to the standar	0x7fff bytes t) UTF-8 [C99 language standard	

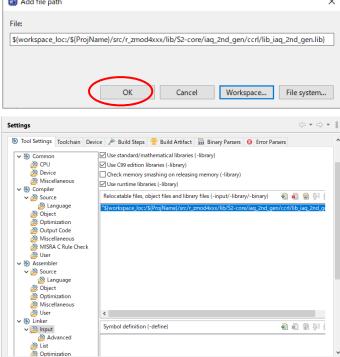


Select [Linker] \rightarrow [Input] in the "Tool Settings" tabbed page and press the [Add] icon.

Press the [Workspace] button in the [Add directory path] dialog box and a list of projects will appear.

Select the "src" folder for the newly created project in the list and select the lib file that according to MCU architecture, measurement mode, and compiler to be used from the "r_zmod4xxx/lib" folder.





Build the project.

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.



8.4 RZ Sample Project

Use the following procedures to modify a sample project.

This section describes an example of modifying the sample project "ZMOD4410_RZG2L_NonOS" so that it can be used on the RZ/G2L Evaluation Kit (SMARC) board.

8.4.1 Importing the Sample Project

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	5
Select an import wizard:	
type filter text	
V 🔁 General	^
📮 Archive File	
CMSIS Pack	
😭 Existing Projects into Workspace	
🔁 File System	
GNUARM-NONE/RZ(DS-5) project conversion to GCC ARM Embedded	
Preferences	
Projects from Folder or Archive	
Rename & Import Existing C/C++ Project into Workspace	
Renesas CS+ Project for CA78K0R/CA78K0	
Renesas CS+ Project for CC-RX and CC-RL	
Renesas GitHub FreeRTOS (with IoT libraries) Project	
Cample Projects on Renesas Website	
> > C/C++	~
< Back Next > Einish Cancel	

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.

Select Folder			×
\leftarrow \rightarrow \checkmark \uparrow \square \rightarrow xx	xxxxx > e2_studio > import	✓ ບັ ,> Sear	ch import
Organize 👻 New fold	er		:== ▼ (?)
>	Name	Date modified	Type Si:
/ M Quick access	ZMOD4410_RZG2L_NonOS	10/24/2022 2:45 PM	File folder
> 🦲 OneDrive	ZMOD4410_ZMOD4510_RZG2L_FreeRTOS	10/24/2022 2:45 PM	File folder
> 💻 This PC	ZMOD4510_RZG2L_NonOS	10/24/2022 2:45 PM	File folder
> 💣 Network			
	<		>
Folde	r: ZMOD4410_RZG2L_NonOS		
		Select Fold	ler Cancel:



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.									
<u>P</u> roject name: ZMC	D4410_RZG2L_NonOS									
Use <u>d</u> efault locat	ion									
Location:	C:\Users\xxxxxxx\e2_studio\workspace\ZMO	D441	B <u>r</u> owse							
	Create Directory for Project									
Choose file system:	default \sim									
Import from:										
Select root direct	ory: C:\Users\xxxxxx\e2_studio\import\ZM	۱ -	B <u>r</u> owse							
O Select <u>a</u> rchive file	8	~	B <u>r</u> owse							
Projects:										
ZMOD4410_RZG2	L_NonOS (C:\Users\xxxxxxx\e2_studio\impor	:\ZMO	D4410_RZG2	L_No						
<				>						
Options	guration output folders									
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish		Cance	I						



8.4.2 Modifying Settings of the FSP Configurator

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

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.

🔅 [ZMOD4	4410_RZG2	L_NonC	OS] FSP Cor	figuration >	<					- 8	
Board S	upport	Packa	age Con	figuratio	n				Genera	Description of the text of tex	
										Restore Defaults	
Device S	election										
FSP ver	sion: 1.1.0)-rc0+2(0221019.bc	a5f7f3	~	Board De	tails				
Board:	RZ/	G2L Eva	luation Kit	(SMARC)	/ 🖄						
Device:	RZ/0	G2LC Ev	r Board (Ai aluation Kit uation Kit ((SMARC)							
Core:			aluation Ki		_ ~						
RTOS:	No	RTOS			\sim						
Summary E	BSP Clock	s Pins	Interrupts	Event Links	Stacks	Components					

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.

[ZMOD4410_RZG2L_NonOS] FSP Configuration ×	
Clocks Configuration	Generate Project Content
	Restore Defaults
OSC 24MHz	^
ICLK 1200MHz	
I2CLK 200MHz	
GCLK 500MHz	
SOCLK 12kHz	
S1CLK 6kHz	
SPIOCLK 200MHz	
SPI1CLK 100MHz	
SD0CLK 533MHz	
SD1CLK 533MHz	
MOCLK 200MHz Summary BSP Clocks Pins Interrupts Event Links Stacks Compo	vents



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

Modify the settings of r_riic_master and r_intc_irq according to the specifications of the target board.

RIIC3 and IRQ7 are assigned to PMOD1 on the RZ/G2L Evaluation Kit (SMARC) board.

To use PMOD1, set "Channel" to "3" on r_riic_master.

🔅 [ZMOD4410_RZG2L_Nor	nOS] FSP Configuration $ imes$			=	- D
Stacks Configuratio	on			Generate Project Cor	ntent
Threads	HAL/Common Stacks		🛃 New Stack >	🐣 Extend Stack > 🛛 🙀 Rem	ove
Remove Remove Remove All/Common Giport I/O Por g_zmod4xxx_se g_timer2 Timer		IOD4410 IAQ 2nd neration n_zmod4xxx)	ADD4XXX Gas Sensor	g_timer2 Timer Driver on r_gtm	
Summary BSP Clocks Pin	Interrupts Event Links Stacks Componen	ts			
Properties	🗙 🖹 Problems 👒 Smart Browser			📑 8 🗖 🗖	
g_i2c_maste	er3 I2C Master Driver on r_riic_mast	ter			
second	operty Common	Value			
	Parameter Checking	Default (BSP)			
	10-bit slave addressing	Disabled			
~	Module g_i2c_master3 I2C Master Driver on r	g_i2c_master3			
	Name				
	Channel Rate	3 Standard			
	Rise Time (ns)	120			
—	Fall Time (ns)	120			
—	Duty Cycle (%)	50			
	Noise Filter Stages	1			
—	Slave Address	0x00			
	Address Mode	7-Bit			
	Timeout Mode	Short Mode			
	Callback	fm_comms_i2c_	callback		
	Interrupt Priority Level	12	-		



To use PMOD1, set "Channel" to "7" on r_intc_irq.

🔅 [ZMOD4410_RZG2L_	NonOS]	FSP Configuration $ imes$					- 8
Stacks Configura	ation						Operation of the second sec
Threads		HAL/Common Stacks			🛃 New Stack >	$\frac{\Phi}{1}$ Extend Stack >	🔊 Remove
HAL/Common Giport I/O g.jopot I/O g.g.timer2 Tin g.timer2 Tin) Port x_sens mer Di	g_joport I/O Port Driver on r_joport	(rm () () () () () () () () () ()	D_zmod4xxx)	MOD4XXX Gas Sensor	g_timer2 Timer [on r_gtm	Driver
Summary BSP Clocks	Pins Int	errupts Event Links Stacks Co	mponen	ts			
T Propertie	es 🗙 🚦	Problems 🏼 🛞 Smart Browser				📑 8 🗖	
g_externa	al_irq7	External IRQ Driver on r	_intc_ir	9			
Settings	Propert	-		Value			
	V Com	imon arameter Checking		Default (BSP)			
		lule g_external_irq7 External IRQ	Driver or				
		lame hannel		g_external_irq7 7			
		rigger		Rising			
		allback		ftising rm_zmod4xxx_	irg callback		
		in Interrupt Priority		12	-		
	✓ Pins						
	IR	RQ7		<unavailable></unavailable>			
							_
							+
							+

Press [Generate Project Content] to generate files.

Build the project.

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.



8.4.3 Changing sample code

Open pin_data.c in the src folder and change the g_bsp_pin_cfg_data settings to match the board you are using.

On the RZ/G2L Evaluation Kit (SMARC) board, PMOD1 is assigned RIIC3 and IRQ7.

If you are using PMOD1, set the P18_0 to RIIC3_SDA (Function 3), the P18_1 to RIIC3_SCL (Function 3), and the P3_1 to IRQ7 (Function 1).

9. Viewing gas data

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

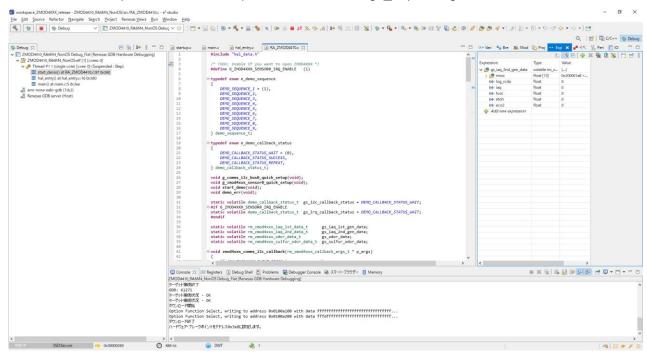
Take the "IAQ 2nd Generation" as an example.

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

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

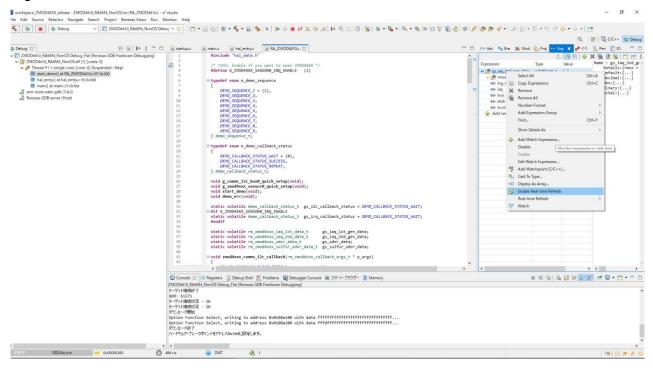
🐔 🎄 🔳 🎋 Debug	V RA ZMOD4XXX BareMotal De	New Window	1 8 - 6 - 10 1 a lat 1	N 3. 3	e i to = = 101	x + + + + + + + + + + + + + + + + + + +	0000	8 8 4 · 1 3131 · 51 · 60 0 0	· +	
		Editor	>					0 10	E C/C++ @ FSP Configure	
		Appearance	3							
t≱ Debug ∑	日常神子中日	Show View	> % Breakpoints Alt+	Shift-Q B Smar	ic () hal_entryic	RA_ZMOD4XXX_IAQ2ndGen.c 🕄		107 V. 🗣 B. 🖂 P. 🏔 M., 🏠 P.		
	[Flat [Renesas GDB Hardware Debugging]	Perspective		Shift+Q.C			•		12 14 14 14 14 14 14 14	* 24
RA ZMOD400X BaneMetaLeff [Second #11 (single core) [cold second #11 (single core)]		the second second	bs Debug					Expression	T., Volue	
	() at RA_ZMOD4XXX_(AU2ndGen.c:18 0i294	Navigation	- Debugger Console					🙀 Add new supremian		
hal entry() at hal entrys:		Preferences	Debug Shell							
main() at main c3 0x3fe		7	Debug Sources							
📓 ann-none-eabi-gdb (7.8.2)		5								
🖉 Renesas GDB server (Host)		CK 18	Disassembly	100000						
		Q 11		Shift+Q, L						
		12	D Executables							
		14	🛠 Expressions	e data	8					
		15	Memory	DENCE	2;					
		1n 17	Minnery Browner							
		18 80808294	Nodules							
		19	E Outline Alt+5	hift+Q.0						
		28 21 88888248	- Penpherals	1.1						
		22		Shift+Q.X						
		23	B Rogers	and the second						
		24 25 000002a4	Project Explorer							
		26								
		27	g III Registers							
		28	5. Signals							
		38 eeeee2bc	S Templates	0010	CALLBACK STATUS	AIT;				
		31	= (0) Variables Alt+	Shift+Q.V						
		32 888882be	Q₂ スマート・ブラウザー	00400	CALLBACK STATUS	AIT;				
		34	U3 37-1-7117W							
		35	Other, Alt+5							
		36 000002c6	0 Alt+2		si-seeasurementsta	rt(g_zmod4xxx_sensor0.p_ctrl);				
		38	(
		39	sequence	- DEMO_SEQUENCE	23					
		8 48	else							
		42								
		6				3	¢	3	1.5 2	
		E Console 17 10	Registers 1 Debug Shell R Problem	ns Debugger Cor	sole 08 27-1-797	- I Memory		= 31 W 1		
			Metal Debug Flat (Renesas GDB Hardwar			0.000			a go at recently to a	
	DLM sta									
	Debug 1	evel : 2								
	Secure/I	VSC memory partition size Code Flash Secure (k8)	1							
			: 22							
			: 0							
		SRAM Secure (kB)	1 3							
			SRAM NSC (KB)	: 5						
		Unable to read	the connected device IO (device	10 address not	specified).					
	,									,
SSD/Secure	- 0x0000294	5 ns	🕡 DWT 🚷 1							0007

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





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



Start the Debug and check the real-time values.

						Q	官 l 酤 C/C+-	to De
ebug 🛙 📄 🙀 🕴 🖗 🗇 🗖	startup.c	@ main.c @ hal_entry.c @ RA_ZMOD4410.c 13	- 0	(H-Vari 💁 Bre 🛋 Mod	Proj •• E	xp 🗙 🧈 4K.	R Peri	0 =
2400641 (24,04AM, NonOS Telloy, Tai (Reneas 00B Hordows Pobugging) 2400041 (24,04AM, NonOS Eff ()) (Concert () (Norming) am nonce edit 11 (single cons) (concert () (Norming) am nonce edit (0,12) Reneas 00B server (Not)	1 2 3 4 5 5 5 5 5 6 7 7 8 9 9 10 11 11 11 12 13 13 13 13 13 13 13 13 13 13	<pre>#include "bud_data," # f* DDD: Cpling if yew uset to upper DEDUCTOR */ define %_DEDUCTOR */ define %_DEDUCTOR */ prove SEQUENCE_1 () fyedef run %_dem_sequence { cord SEQUENCE_1 cord SEQUENCE_1</pre>		Expension v @ equip.ind.pnr, data) @ more ide hou, rula ide hou, rula ide hou ide hou	Type Volatile rm, z fload (13) float float float float	Ish e Ib Value Ib Contract Ib State Ib <	× × 0 ×	



Revision History

			Description
Rev.	Date	Page	Summary
Rev 1.00	June 30, 2021	-	First Release
Rev 1.10	September 30, 2021	-	Add RX Family, RL78 Family and RE01 256KB Group
D			supporting
Rev.1.20	December 20, 2021	-	Add: Support multiple ZMOD sensors usage
			Add: Support RX Azure
			Other minor changes
Rev.1.30	March 1, 2022	-	Update chapters 2, 5 and 6 to add IAQ 2nd Gen ULP
			mode
Rev.1.31	March 11, 2022	-	Fix RA sample projects
		P9	Changed Figure 2-6 Hardware Connections for RE01 1500KB
Rev.1.40	June 30, 2022	P8, P9	Changed Environment for Confirming Operation on an RE01
			Fixed: RE01 sample project codes
	August 31, 2022	-	Added: ZMOD4450 new.
			Fixed: Sample project codes
Rev.1.50	November 30, 2022	-	Added: RZ items
			Other minor changes
Rev.1.51	February 20, 2023	-	Bug fixes
			Updated: Environments for RL78
Rev.1.52	March 29, 2023	-	Updated: Environments for RA, RX, RL78, RZ
			Updated: Main Processing Flow of Sample Software
			Updated: Guide for Changing the Target Device
Rev.1.53	June 28, 2023	-	Updated: Environments for RA
			Updated: ZMOD4410 Sensor specifications
			Updated: Specification of Sample Software
			Updated: MOD4xxx Settings for RX, RL78
Rev.1.54	September 7, 2023	-	Updated: Guide for Changing the Target Device
			Deleted: RE01 items



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. Proceeding a transport of the 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 system-evaluation test for the given product.

Notice

- 1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- 2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
- 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
- 5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
- 6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
 - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

- 7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
- 8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- 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: www.renesas.com/contact/.