

RA Family, RX Family, RL78 Family, RZ Family

HS300x Sample Software Manual

Introduction

This application note describes the sample software that is for use with the HS300x humidity and temperature sensor and runs on certain MCUs of the RA family, RX family, RL78 family, and RZ family.

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.

 ${\sf Microsoft}^{{\scriptscriptstyle (\!\! R \!\!\!)}}$ Azure RTOS is trade marks of the Microsoft group of companies. All other trade marks are property of their respective owners.

Contents

1.	Overview4
2.	Environment for Confirming Operation4
2.1	Environment for Confirming Operation on an RA Family MCU4
2.2	Environment for Confirming Operation on an RX Family MCU6
2.3	Environment for Confirming Operation on an RL78/G14 Group MCU7
2.4	Environment for Confirming Operation on an RL78/G23 Group MCU8
2.5	Environment for Confirming Operation on an RZ Family MCU9
3.	Sensor Specifications
3.1	Overview of Sensor Specifications
3.2	Sensor Functions10
3.2.1	I2C Communication Interface11
3.2.2	Expressions for Converting Output Values to Humidity and Temperature
4.	Sample Software Specifications12
4.1	Configuration of the Sample Software12
4.2	Specifications of Sensor API Functions12
4.2.1	List of Sensor API functions



RA Family, RX Family, RL78 Family, RZ Family

4.2.2	Guide to Using the API Functions	13
4.3	Flowchart of the Main Processing in the Non-OS Version of the Sample Software	16
4.4	Flowchart of the OS Version of the Sample Software	18
4.5	Azure RTOS Project	20
_		
	Configuration Settings	
5.1	HS300x Humidity and Temperature Sensor Settings	
5.1.1	RA Family	
5.1.2		
5.1.3		
5.1.4		
5.2	Communication Driver Middleware Settings	
5.2.1	RA Family	
5.2.2		
5.2.3	,	
5.2.4		
5.3	I2C Driver Settings	
5.3.1	RA Family	
5.3.2	RX Family	
5.3.3	RL78 Family	
5.3.4	RZ Family	40
6. (Guide to Changing the Target Device	42
6.1	RA Sample Project	
6.1.1	Importing the Sample Project	
6.1.2		
0.1.2	Modifying Settings of the ESP Configurator	11
612		
6.1.3	Changing toolchain setting	49
6.2	Changing toolchain setting RX Sample Project	49 50
6.2 6.2.1	Changing toolchain setting RX Sample Project Importing the Sample Project	49 50 50
6.2 6.2.1 6.2.2	Changing toolchain setting RX Sample Project Importing the Sample Project Changing the Device	49 50 50 52
 6.2 6.2.1 6.2.2 6.2.3 	Changing toolchain setting RX Sample Project Importing the Sample Project Changing the Device Modifying Settings of the Smart Configurator	49 50 50 52 54
6.26.2.16.2.26.2.36.2.4	Changing toolchain setting RX Sample Project Importing the Sample Project Changing the Device Modifying Settings of the Smart Configurator Changing toolchain setting	49 50 50 52 54 56
 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.3 	Changing toolchain setting RX Sample Project Importing the Sample Project Changing the Device Modifying Settings of the Smart Configurator Changing toolchain setting RL78 Sample Project	49 50 50 52 54 56 57
 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.3 6.3.1 	Changing toolchain setting RX Sample Project Importing the Sample Project Changing the Device Modifying Settings of the Smart Configurator Changing toolchain setting RL78 Sample Project Creating a New Project	
 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.3 6.3.1 6.3.2 	Changing toolchain setting RX Sample Project Importing the Sample Project Changing the Device Modifying Settings of the Smart Configurator Changing toolchain setting RL78 Sample Project Creating a New Project Settings of the Code Generator	
 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.3 6.3.1 6.3.2 6.3.3 	Changing toolchain setting RX Sample Project Importing the Sample Project Changing the Device Modifying Settings of the Smart Configurator Changing toolchain setting RL78 Sample Project Creating a New Project Settings of the Code Generator Modifying the Generated Code	
 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.3 6.3.1 6.3.2 6.3.3 6.3.4 	Changing toolchain setting RX Sample Project Importing the Sample Project Changing the Device Modifying Settings of the Smart Configurator Changing toolchain setting RL78 Sample Project Creating a New Project Settings of the Code Generator Modifying the Generated Code Modifying Sample Source Files	49 50 52 54 56 57 57 57 59 63 67
 6.2 6.2.1 6.2.3 6.2.4 6.3 6.3.1 6.3.2 6.3.3 6.3.4 6.4 	Changing toolchain setting RX Sample Project Importing the Sample Project Changing the Device Modifying Settings of the Smart Configurator Changing toolchain setting RL78 Sample Project. Creating a New Project Settings of the Code Generator Modifying the Generated Code Modifying Sample Source Files RZ Sample Project	49 50 52 54 54 57 57 59 63 67 72
 6.2 6.2.1 6.2.2 6.2.3 6.2.4 6.3 6.3.1 6.3.2 6.3.3 6.3.4 6.4 6.4.1 	Changing toolchain setting RX Sample Project Importing the Sample Project Changing the Device Modifying Settings of the Smart Configurator Changing toolchain setting RL78 Sample Project Creating a New Project Settings of the Code Generator Modifying the Generated Code Modifying Sample Source Files RZ Sample Project Importing the Sample Project	49 50 52 54 54 57 57 57 57 63 67 72 72
 6.2 6.2.1 6.2.3 6.2.4 6.3 6.3.1 6.3.2 6.3.3 6.3.4 6.4 	Changing toolchain setting RX Sample Project Importing the Sample Project Changing the Device Modifying Settings of the Smart Configurator Changing toolchain setting RL78 Sample Project Creating a New Project Settings of the Code Generator Modifying the Generated Code Modifying Sample Source Files RZ Sample Project Importing the Sample Project	49 50 52 54 56 57 57 57 59 63 67 72 72 74



7. Viewing Temperature and Humidity Data	7
Revision History	'9
General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products8	30
Notice	31
Corporate Headquarters	31
Contact information	31
Trademarks	31



1. Overview

This sample software acquires humidity and temperature data from the HS300x humidity and temperature sensor and handles calculations on the data. In combination with the I2C driver of the FSP or FIT, the sample software controls the HS300x through the I2C in the MCU to measure humidity and temperature, acquire ADC data, and calculate the acquired data.

2. Environment for Confirming Operation

2.1 Environment for Confirming Operation on an 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: 144 pins)
Operating frequency	200 MHz
Operating voltage	5 V
Integrated development environment	e ² Studio 2023-01
C compiler	GCC 10.3.1.20210824
	IAR ANSI C/C++ Compiler V8.50.9.278/LNX for ARM
	ARM Compiler 6.16
FSP	V.3.7.0
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	Relative Humidity Sensor Pmod [™] Board (US082-HS3001EVZ)

Table 2-1 Operating Environment for the RA Family MCU

Table 2-2 Amount of Memory Used in the RA Family MCU

Area	Size
ROM	Non-OS version: 1,317 bytes
	FreeRTOS version: 1,630 bytes
	Azure RTOS version: 1,606 bytes
RAM	Non-OS version: 77 bytes
	FreeRTOS version: 257 bytes
	Azure RTOS version: 426 bytes

Memory size is calculated by functions and variables only related to HS300x sensor. In RTOS, memory size does not include memory size of the thread.



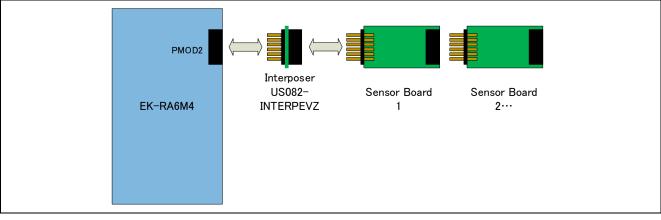


Figure 2-1 Hardware Connections for the RA Family



2.2 Environment for Confirming Operation on an 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: 144 pins)
Operating frequency	12 MHz
Operating voltage	5 V
Integrated development environment	e ² Studio 2023-01
	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	Relative Humidity Sensor Pmod [™] Board (US082-HS3001EVZ)

Table 2-3 Operating Environment for the RX Family MCU

Table 2-4 Amount of Memory Used in the RX Family MCU

Area	Size
ROM	Non-OS version: 1,652 bytes
	FreeRTOS version: 1,992 bytes
	Azure RTOS version: 2,041 bytes
RAM	Non-OS version: 133 bytes
	FreeRTOS version: 149 bytes
	Azure RTOS version: 289 bytes

Memory size is calculated by functions and variables only related to HS300x sensor. In RTOS, memory size does not include memory size of the thread.

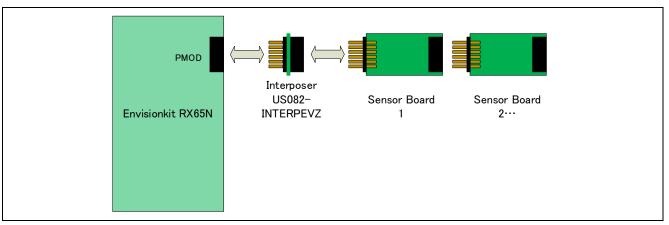


Figure 2-2 Hardware Connections for the RX Family



2.3 Environment for Confirming Operation on an RL78/G14 Group MCU

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

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

Item	Description
Demonstration board	RTK5RLG140C00000BJ (RL78/G14 Fast Prototyping Board)
Microcontroller	RL78/G14 (R5F104MLAFB :80pin)
Operating frequency	32MHz
Operating voltage	3.3V
Integrated development environment	e ² Studio 2023-01
	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)(RL78/G14)
Sensor board	Relative Humidity Sensor Pmod [™] Board (US082-HS3001EVZ)

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

Area	Size
ROM	1,542 bytes
RAM	83 bytes

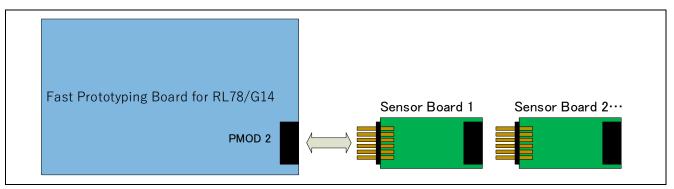


Figure 2-3 Hardware Connections for the RL78/G14 Group



2.4 Environment for Confirming Operation on an RL78/G23 Group MCU

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

Table 2-7	Operating Environment for the RL78/G23 Group MCU
-----------	--

Item	Description
Demonstration board	RTK7RLG230CSN000BJ (RL78/G23-128p Fast Prototyping
	Board)
Microcontroller	(R7F100GSN2DFB :128pin)
Operating frequency	32MHz
Operating voltage	3.3V
Integrated development environment	e ² Studio 2023-01
	IAR EW for RL78 4.21.1
C compiler	C compiler package for RL78 family V1.10.00
	LLVM for RL78 10.0.0.202209
	IAR Toolchain for RL78 4.21.1.2409
Emulator	E2 Lite
Sensor board	Relative Humidity Sensor Pmod [™] Board (US082-HS3001EVZ)

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

Area	Size
ROM	1,690 bytes
RAM	80 bytes

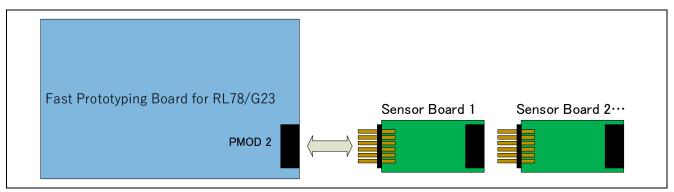


Figure 2-4 Hardware Connections for the RL78/G23 Group



2.5 Environment for Confirming Operation on an RZ Family MCU

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

Table 2-9	Operating Environment for the RZ Family MCU
-----------	---

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	5 V
Integrated development environment	e ² Studio 2023-01
C compiler	GCC 10.3.1.20210824
FSP	V.1.2.0
RTOS	FreeRTOS™
Emulator	SEGGER J-LINK BASE
Sensor board	Relative Humidity Sensor Pmod [™] Board (US082-HS3001EVZ)

Table 2-10 Amount of Memory Used in the RZ Family MCU

Area	Size	
ROM	Non-OS version: 1,574 bytes	
	FreeRTOS version: 1,882 bytes	
RAM	Non-OS version: 149 bytes	
	FreeRTOS version: 429 bytes	

Memory size is calculated by functions and variables only related to HS300x sensor. In RTOS, memory size does not include memory size of the thread.

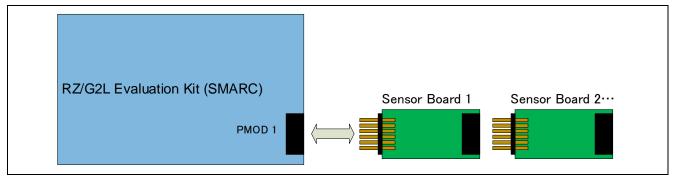


Figure 2-5 Hardware Connections for the RZ Family



3. Sensor Specifications

3.1 Overview of Sensor Specifications

Table 3-1 gives an overview of the specifications of the HS300x humidity and temperature sensor.

Table 3-1 Overview of Sensor Specifications

Item	Description
Humidity range	0%RH to 100%RH
Humidity accuracy	±1.5%RH (typ.) (HS3001: 10%RH to 90%RH, 25°C)
14-bit resolution	0.01%RH (typ.)
Independent programmable resolution settings	14 bits
Fast RH response time	1 second (typ.) (with 1 m/s air flow)
	4 seconds (typ.) (in a sealed space)
Temperature sensor accuracy	±0.2°C typ. (HS3001 and HS3002: –10°C to 80°C)
Average Current	14-bit resolution at 3.3 V: 24.4 μA (max.)
Sleep Current	–40°C to 85°C: 0.6 μA (typ.)
	–40°C to 125°C: 1 μA (typ.)
Supply voltage	2.3 V to 5.5 V, 3.3 V (typ.)
Extended supply voltage	1.8 V (–20°C to +125°C)
Operating temperature	-40°C to +125°C

3.2 Sensor Functions

This software supports the following functions of HS300x.

Function	Description	
I2C communications	Sensor data are transferred through I2C communications.	
Measurement mode	This software operates with the sensor in sleep mode most of the time.	
	On completion of measurement triggered by a measurement request, the sensor enters sleep mode.	
Measurement request	The sensor in sleep mode is placed in the measurement state upon receiving a measurement request.	
Data fetch	At the end of a measurement cycle, valid data can be acquired.	
Status bits	The status bits for the results of measurement indicate whether the current data are valid or old.	

Note: This software does not support the following functions of HS300x.

- Access to non-volatile memory
- Setting of the measurement resolution
- Reading of the HS300x ID number



3.2.1 I2C Communication Interface

The following shows the format of measurement data transferred through I2C communications.

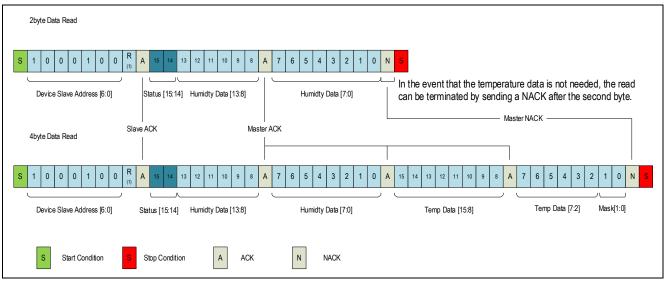


Figure 3-1 Format of I2C Communications

The status bits indicate the state of data as follows.

00_B: Valid data acquired at the end of a measurement cycle

 $01_{\mbox{\scriptsize B}}$. Invalid data that have already been acquired

3.2.2 Expressions for Converting Output Values to Humidity and Temperature

The HS300x software uses functions for converting the acquired ADC data to values for humidity and temperature. The following shows the conversion expressions.

• Humidity conversion expression

Humidity [%RH] =
$$\left(\frac{Humidity [13:0]}{2^{14} - 1}\right) * 100$$

• Temperature conversion expression

Temperature
$$[{}^{\circ}C] = \left(\frac{Temperature [15:2]}{2^{14}-1}\right) * 165 - 40$$



4. Sample Software Specifications

This sample software package contains a total of thirteen projects: non-OS and OS versions for the RA family, non-OS and OS versions for the RX family, non-OS version for the RL78 family, and a non-OS and OS versions for the RZ family. This section describes these projects.

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

4.1 Configuration of the Sample Software

Figure 4-1 is a block diagram of the sample software.

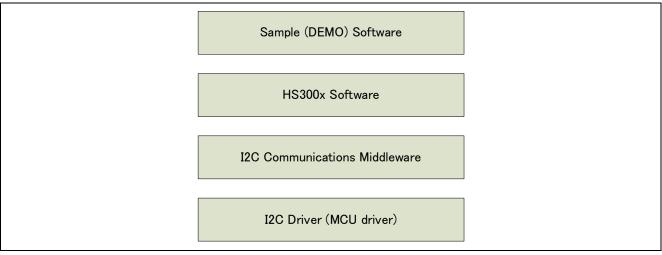


Figure 4-1 Block Diagram of the Sample Software

4.2 Specifications of Sensor API Functions

4.2.1 List of Sensor API functions

The following table lists the sensor API functions. For details of the API functions, refer to the separately provided RX Family HS300x Sensor API FIT Module application note (R01AN5893) and RL78 Family Renesas Sensor Control Modules application note (R01AN5896).

Function	Description
RM_HS300X_Open	Starts control of the sensor.
RM_HS300X_Close	Terminates control of the sensor.
RM_HS300X_MeasurementStart	Starts measurement by the sensor.
RM_HS300X_Read	Acquires data from the sensor.
RM_HS300X_DataCalculate	Calculates values from the data acquired from the sensor.
RM_HS300X_ProgrammingModeEnter	Starts programming mode.
RM_HS300X_SensorIdGet	Get ID of the sensor.
RM_HS300X_ResolutionChange	Change the measurement resolution.
RM_HS300X_ProgrammingModeExit	Terminates programming mode.

Table 4-1 List of Sensor API Functions



4.2.2 Guide to Using the API Functions

The following diagram of API function transitions shows the conditions on the usage of the individual HS300x API functions and the expected orders of function calls.

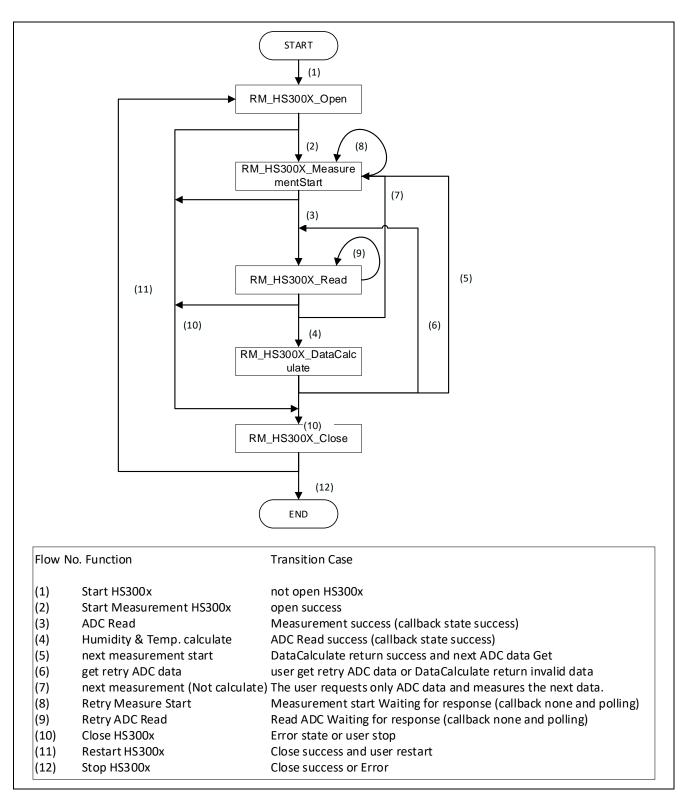


Figure 4-2 Diagram of Transitions between API Function Calls



The conditions for calling the individual functions are shown below.

• RM HS300X Open: (1) Activation of HS300x or (11) restart after a call of RM HS300X Close RM HS300X Close: (10) Successful completion or abnormal end of individual processing (2) Start of measurement after a call of RM HS300X Open, RM HS300X MeasurementStart: (5) (7) acquisition of the next measured data, or (8) retry after waiting for the response to the measurement start request • RM HS300X Read: (3) Acquisition of measured data after a call of RM_HS300X_MeasurementStart or (9) retry after waiting for the response to the data acquisition request (4) Calculation of humidity and temperature data after a call of RM_HS300X_DataCalculate: RM HS300X Read

Notes:

Since RM_HS300X_Open checks the state of the I2C driver, the I2C driver must be opened before the RM_HS300X_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 measurement is started by RM_HS300X_MeasurementStart, the sensor stops measurement after having updated the ADC data. Therefore, be sure to call RM_HS300X_MeasurementStart at least once every time before executing processing by RM_HS300X_Read.

When using an OS and controlling the sensor with multiple threads or tasks simultaneously in use, the user will need to use a semaphore to control the bus. For the timing of the semaphore being raised and the control of blocking, refer to 4.4 Flowchart of the OS Version of the Sample Software, Flowchart of the OS Version of the Sample Software.



Shows the flow and timing of Programming mode.

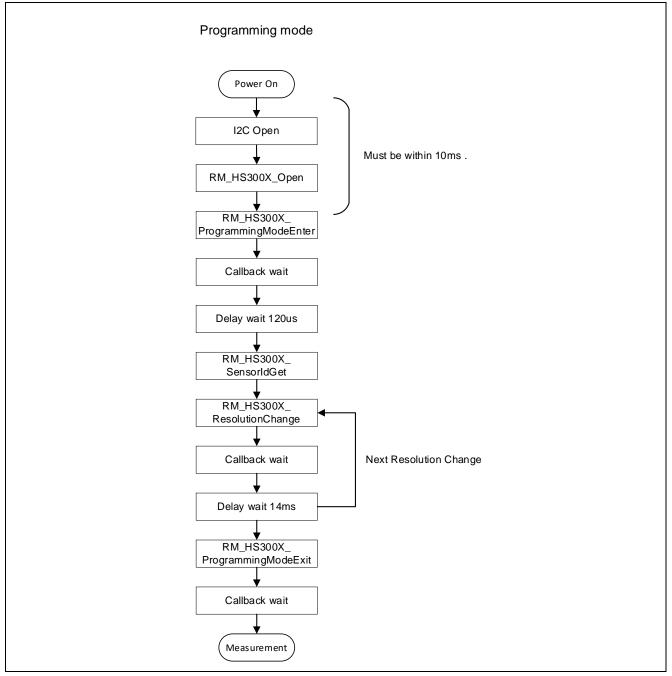


Figure 4-3 Diagram of Programming mode.



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

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

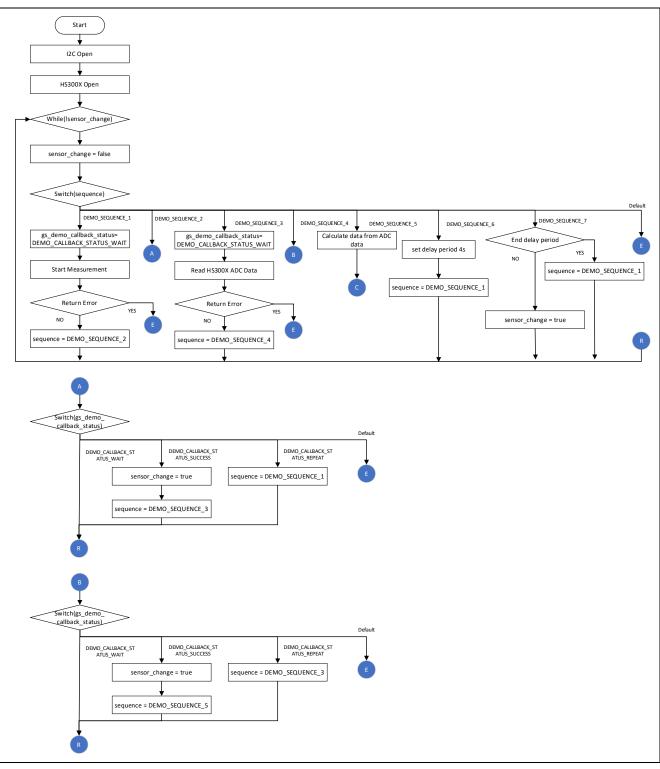


Figure 4-4 Flowchart of the Main Processing in the Non-OS Version of the Sample Software (1)

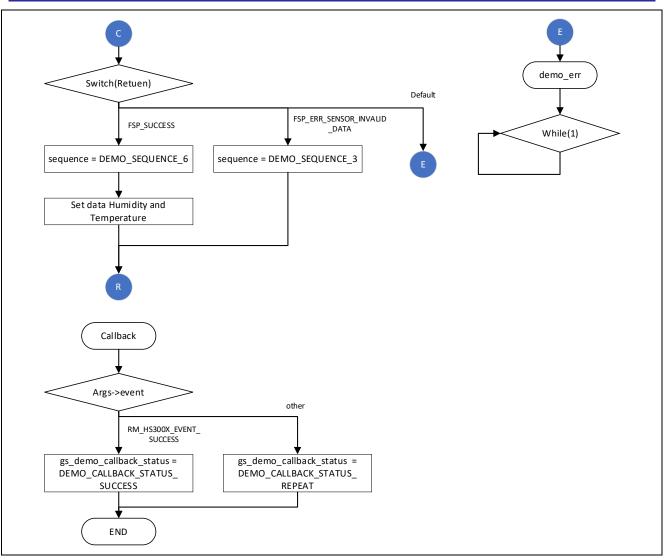


Figure 4-5 Flowchart of the Main Processing in the Non-OS Version of the Sample Software (2)



4.4 Flowchart of the OS Version of the Sample Software

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

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

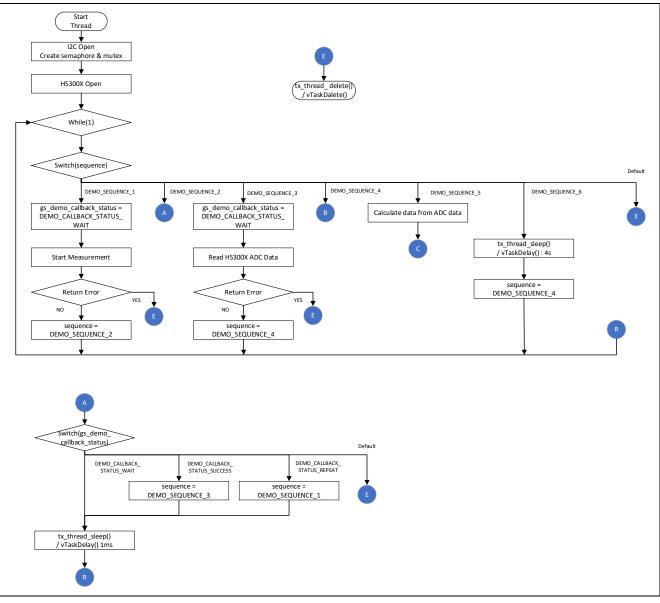


Figure 4-6 Flowchart of the Main Processing in the OS Version of the Sample Software (1)



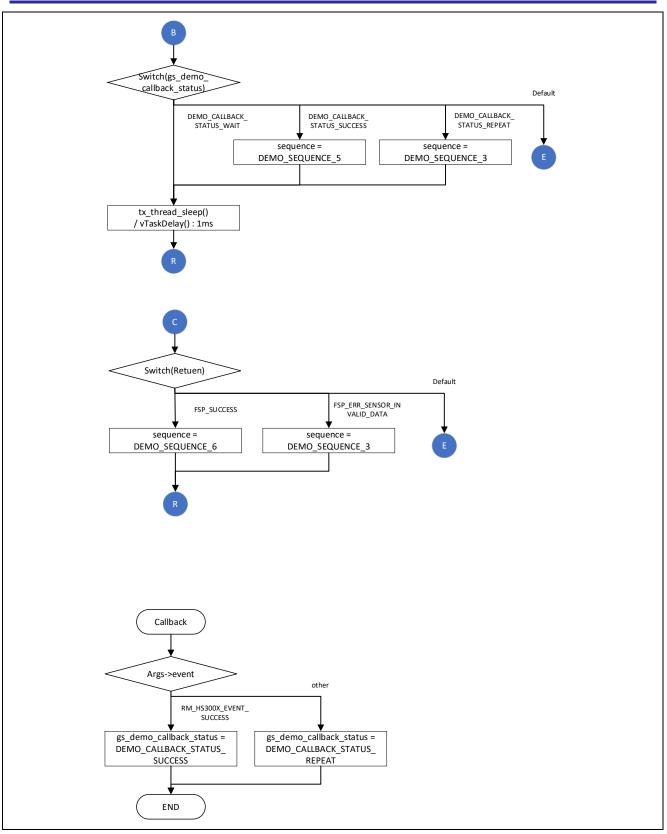


Figure 4-7 Flowchart of the Main Processing in the OS Version of the Sample Software (2)



4.5 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/hs300x_sensor_thread_entry.c
 27th line: Change from #include "azurertos_object_init.h" to #include "tx_api.h"



5. Configuration Settings

5.1 HS300x Humidity and Temperature Sensor Settings

5.1.1 RA Family

Select the rm_hs300x stack on the [Stack] tabbed page of the FSP Configurator, and the configurable items will be shown on the [Properties] tabbed page.

The following items and values can be specified.

Table 5-1 HS300x Settings for the RA Family MCU

Configurable Item	Value	Description		
Common	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.		
Data type	Both humidity and temperature	Specify the type of data to be acquired from the		
	Humidity only	sensor.		
		Both the humidity and temperature or the humidity alone can be selected.		
Programming Mode	OFF	Specify programming mode.		
Module g_hs300	0x_sensor HS300X on rr	n_hs300x		
Name	g_hs300x_sensor0	Specify the name of the module. The module name must conform with the C language standard.		
Callback	hs300x_callback	Specify the name of the user callback function. The callback function name must conform with the C language standard. When "NULL" is specified, no callback function is used.		



5.1.2 RX Family

Select the $r_hs_{300x}rx$ 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.

Table 5-2 HS300x Settings for the RX Family MCU

Configurable Item	Value	Description		
Configurations	Configurations			
Parameter Checking	System Default	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.		
Number of HS300x	1	Specify the number of HS300x sensors to be		
sensors	2	connected.		
Data types from	Humidity and temperature	Specify the type of data to be acquired from the		
HS300x sensors	Humidity only	sensor.		
		Both the humidity and temperature or the humidity alone can be selected.		
Programming Mode	OFF	Specify programming mode.		
I2C communication	I2C Communication Device0	Specify the communications device number to		
device No. for HS300x Sensor	I2C Communication Device1	be used by the sensor.		
Device x	I2C Communication Device2			
(x = 0 or 1)	I2C Communication Device3			
	I2C Communication Device4			
	Enabled			
Callback function for HS300x sensor device x (x = 0 or 1)	hs300x_user_callback(x) (x = 0 or 1)	Specify the name of the user callback function. The callback function name must conform with the C language standard.		



5.1.3 RL78 Family

Settings can be modified by changing the values of constants defined in the $r_config_r_hs_{300x_rl}_config_h$ file in the project tree of the sample project.

The following items and values can be specified.

Constant Name	Value	Description
Configurations		
RM_HS300X_CFG_PARAM _CHECKING_ENABLE	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_HS300X_CFG_DEVICE	1	Specify the number of HS300x sensors
_NUM_MAX	2	to be connected.
RM_HS300X_CFG_DATA_B	0	Specify the type of data to be acquired
OTH_HUMIDITY_TEMPERA	1	from the sensor.
TURE		The humidity alone (0) or both the humidity and temperature (1) can be selected.
RM_HS300X_CFG_PROGR AMMING_MODE	0	Specify programming mode.
RM_HS300X_CFG_DEVICE x_COMMS_INSTANCE (x = 0 or 1)	g_comms_i2c_device(x) (x = 0 or 1)	Specify the instance name of the communications line to be used.
RM_HS300X_CFG_DEVICE x_CALLBACK	hs300x_user_callback1(x) (x = 0 or 1)	Specify the name of the user callback function.
(x = 0 or 1)		The callback function name must
		conform with the C language standard.

Table 5-3 HS300x Settings for the RL78 Family MCU



5.1.4 RZ Family

Select the rm_hs300x stack on the [Stack] tabbed page of the FSP Configurator, and the configurable items will be shown on the [Properties] tabbed page.

The following items and values can be specified.

Table 5-4 HS300x Settings for the RZ Family MCU

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.
Data type	Both humidity and temperature	Specify the type of data to be acquired from the
	Humidity only	sensor.
		Both the humidity and temperature or the humidity alone can be selected.
Programming Mode	OFF	Specify programming mode.
Module g_hs300	0x_sensor HS300X on rr	n_hs300x
Name	g_hs300x_sensor0	Specify the name of the module. The module name must conform with the C language standard.
Callback	hs300x_callback	Specify the name of the user callback function. The callback function name must conform with the C language standard. When "NULL" is specified, no callback function is used.



5.2 Communication Driver Middleware Settings

5.2.1 RA Family

Select the rm_comms_i2c stack on the [Stack] tabbed page of the FSP Configurator, and the configurable items will be shown on the [Properties] tabbed page.

The following items and values can be specified.

Table 5-5 Communication Driver Settings for the RA Family MCU

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 comm	ns_i2c_device I2C Comm	nunication Device on
rm_comms_i2c		
Name	g_comms_i2c_device0	Specify the name of the module.
		The module name must conform with the C language standard.
Semaphore Timeout	0xFFFFFFF	For an RTOS project, specify the time of semaphore timeout.
Slave Address	0x44	Specify the slave address.
		When rm_hs300x is used, this value is automatically specified and cannot be modified.
Address Mode	7-Bit	Specify the number of slave address bits.
		When rm_hs300x is used, this value is automatically specified and cannot be modified.
Callback	rm_hs300x_callback	Specify the name of the user callback function.
		When rm_hs300x is used, this value is
	a ila hual Ila Charad	automatically specified and cannot be modified.
Name	ns_i2c_bus0 I2C Shared g_comms_i2c_bus0	Specify the name of the I2C module.
Bus Timeout	0xFFFFFFF	Specify the time of I2C bus timeout.
Semaphore for	Unuse	For an RTOS project, enable or disable
blocking	Use	processing for blocking.
Recursive Mutex for	Unuse	
Bus	Use	For an RTOS project, enable or disable recursive operation when blocking is enabled.
240	056	



5.2.2 RX Family

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

The following items and values can be specified.

Table 5-6	Communication Driver Settings for the RX Family MCU	
Table 5-0	Communication Driver Settings for the KA Family WCO	

Configurable Item	Value	Description
Configurations		
Parameter Checking	System Default Enabled	Enable or disable the parameter check processing.
	Disabled	When "Enabled" is selected, the project is built so that the generated code includes the parameter check processing.
Number of	Unused	Specify the number of communications bus
communication lines	1	lines that can be connected.
	2	
	3	
	4	
	5	
Number of I2C	Unused	Specify the number of I2C device that can be
Devices	1	connected.
	2	
	3	
	4	
	5	
Blocking operation	Disabled	For an RTOS project, enable or disable the
supporting with RTOS	Enabled	blocking operation.
Bus lock operation	Disabled	For an RTOS project, enable or disable the bus
supporting with RTOS	Enabled	lock operation.
IIC Driver Type for	RIIC	Specify the I2C bus type to be used for the
I2C Shared bus(x)	SCI IIC	communication bus.
(x = 0 - 4)	Not selected	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 - 4)	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 - 4)	0xFFFFFFF	Specify the time of I2C bus(x) timeout. (x = 0 - 4)
I2C Shared Bus No. for I2C Communication Device(x) (x = 0 - 4)	I2C Shared Bus(x) (x = 0 - 4)	Specify the configuration of used communication bus.
Slave address for communication device(x)	0x44	Specify the slave address of the device to be connected to the communications bus. If you are using rm_hs300x, specify 0x44.



(x = 0 - 4)		
Slave address mode for communication device(x) (x = 0 - 4)	7 bit address mode	Specify the slave address mode. If you are using rm_hs300x, specify the 7-bit address mode.
Callback function for Communication device(x) (x = 0 - 4)	comms_i2c_user_callback(x) (x = 0 - 4)	Specify the name of the user callback function. When using rm_hs300x, specify rm_hs300x_callback(y) (y = 0-1).



5.2.3 RL78 Family

Settings can be modified by changing the values of constants defined in the $r_config_r_comms_i2c_rl_config.h$ file in the project tree of the sample project.

The following items and values can be specified.

	• · · ·		•			
Table 5-7	Communication	Driver	Settings	for the	RL78	Family MCU

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 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.
$COMMS_{12C}CFG_DEVIC$ E(x)_BUS_CH (x = 0 - 4)	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)	0x44	Specify the slave address of the device to be connected to the communications bus. If you are using rm_hs300x, specify 0x44.
COMMS_I2C_CFG_DEVIC E(x)_CALLBACK (x = 0 - 4)	comms_i2c_user_callback(x) (x = 0 - 4)	Specify the name of the user callback function. When using rm_hs300x, specify rm_hs300x_callback(y) (y = 0 - 1).



5.2.4 RZ Family

Select the rm_comms_i2c stack on the [Stack] tabbed page of the FSP Configurator, and the configurable items will be shown on the [Properties] tabbed page.

The following items and values can be specified.

Table 5-8 Communication Driver Settings for the RA Family MCU

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_comm rm_comms_i2c	ns_i2c_device I2C Comr	
Name	g_comms_i2c_device0	Specify the name of the module. The module name must conform with the C language standard.
Semaphore Timeout	0xFFFFFFF	For an RTOS project, specify the time of semaphore timeout.
Slave Address	0x44	Specify the slave address. When rm_hs300x is used, this value is automatically specified and cannot be modified.
Address Mode	7-Bit	Specify the number of slave address bits. When rm_hs300x is used, this value is automatically specified and cannot be modified.
Callback	rm_hs300x_callback	Specify the name of the user callback function. When rm_hs300x is used, this value is automatically specified and cannot be modified.
Module g_comm	ns_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 processing for blocking.
Recursive Mutex for	Unuse	For an RTOS project, enable or disable
Bus	Use	recursive operation when blocking is enabled.



5.3 I2C Driver Settings

5.3.1 RA Family

Select the r_iic_master or r_sci_i2c stack on the [Stack] tabbed page of the FSP Configurator, and the configurable items will be shown on the [Properties] tabbed page.

The following items and values can be specified.

Table 5-9 r_iic_master Settings for the RA Family MCU

Configurable Item	Value	Description
Common	1	
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 in transmission
Transmission and Reception	Disabled	and reception.
10-bit slave	Enabled	Specify whether to support 10-bit slave
addressing	Disabled	addresses. If you are using rm_hs300x, select "Disabled".
Module g_i2c_m	aster0 I2C Master Drive	er on r_iic_master
Name	g_i2c_master0	Specify the name of the module.
Channel	0	Specify the channel number to be used.
Rate	Standard	Specify the bit rate.
	Fast-mode	If you are using rm_hs300x, select "Standard" or "Fast-mode".
	Fast-mode plus	
Rise Time (ns)	120	Specify the SCL rise time to suit the specifications of the target board to be used.
Fall Time (ns)	120	Specify the SCL fall time to suit 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 any setting that is made here is overwritten by the setting in rm_comms_i2c.
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 any setting that is made here is overwritten by the setting in rm_comms_i2c.
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) Priority 1 Priority 2 Priority 3 Priority 4 Priority 5 Priority 5 Priority 7 Priority 8 Priority 9 Priority 10	Specify the interrupt priority level of the I2C bus driver.
	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 if this is required.



Table 5-10 r_sci_i2c Settings for the RA Family MCU

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.
DTC on	Enabled	Specify whether to use the DTC in transmission
Transmission and Reception	Disabled	and reception.
10-bit slave addressing	Enabled	Specify whether to support 10-bit slave addresses.
addressing	Disabled	If you are using rm_hs300x, select "Disabled".
Module g_i2c0 I	2C Master Driver on r_s	ci_i2c
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 any setting that is made here is overwritten by the setting in rm_comms_i2c.
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 any setting that is made here is overwritten by the setting in rm_comms_i2c.
Rate	Standard	Specify the bit 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 division value of the noise filter for input signals.
	Use clock signal divided by 2 with noise filter	
	Use clock signal divided by 4 with noise filter	
	Use clock signal divided by 8 with noise filter	
Bit Rate Modulation	Enable	Enable or disable the bit rate modulation
A	Disable	function.
Callback	rm_comms_i2c_callback	The name of the user callback function is automatically specified by rm_comms_i2c.



Interrupt Priority	Priority 0 (highest)	Specify the interrupt priority level of the I2C bus
Level	Priority 1	driver.
	Priority 2	
	Priority 3	
	Priority 4	
	Priority 5	
	Priority 6	
	Priority 7	
	Priority 8	
	Priority 9	
	Priority 10	
	Priority 11	
	Priority 12	
	Priority 13	
	Priority 14	
	Priority 15	
RX Interrupt Priority	Priority 0 (highest)	When the DTC is to be used, specify the priority
Level [Only used	Priority 1	level of the reception interrupt.
when DTC is enabled]	Priority 2	
enabled	Priority 3	
	Priority 4	
	Priority 5	
	Priority 6	
	Priority 7	
	Priority 8	
	Priority 9	
	Priority 10	
	Priority 11	
	Priority 12	
	Priority 13	
	Priority 14	
	Priority 15	
	Disabled	
Pins		
SDA	Рххх	The pin numbers to be used by the driver are
SCL	Pxxx	displayed.
		Use the [Pins] tabbed page to modify the pin configuration if this is required.



5.3.2 RX Family

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

The following items and values can be specified.

Table 5-11 r_riic_rx Settings for the RX Family MCU

Configurable Item	Value	Description
Configurations		
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.
CHx RIIC bps(kbps)	400	Specify the bit rate.
(x = 0 - 2)		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 CHx	Used	detection function.
(x = 0 - 2)		
Address y format for CHx	Not	This item specifies the slave address mode for
(x = 0 - 2, y = 0 - 2)	7 bit address format	slave address y but the user does not need to make this setting because any setting that is made here is
(x = 0 = 2, y = 0 = 2)		overwritten by the setting in rm_comms_i2c.
Slave Address y for CHx (x = 0 - 2, y = 0 - 2)	0x0025	This item specifies slave address y but the user does not need to make this setting because any setting that is made here is overwritten by the 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	
	Level 14	
	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 (x = 0 – 2)	Level 2	
	Level 3	
	Level 4	
	Level 5	
	Level 6	
	Level 7	
	Level 8	
	Level 9	
	Level 10	1
	Level 11	
	Level 12	
	Level 13	
	Level 14	1
	Level 15 (highest)	1



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



Configurable Item	Value	Description
Configurations		
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 $(x = 0 - 12)$	Supported	х.
SCI IIC bitrate (bps)	384000	Specify the bit rate.
for CHx		Set to 384000 or a smaller value.
(x = 0 - 12)		
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	Enable	
(x = 0 - 12)		
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	
· · ·	The clock divided by 8	
I2C Mode Register 1	18	Specify the number of SDA output delay cycles.
(IICDL bit) for CHx		
(x = 0 - 12)	1000	Chapter the counter value to be indeed to represent
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.
L	1	

Table 5-12 r_sci_iic_rx Settings for the RX Family MCU



5.3.3 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.

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_hs300x, select IICxx.
	IICxx	
IICxx		
Transfer rate	1000000	Specify the bit rate.
		If you are using rm_hs300x, specify 100000.
Transfer end interrupt priority (INTIICxx)	High	Specify the priority level of the transfer end
	Level1	interrupt.
	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.
IICAx		
Transfer mode		
Transfer mode	Unused	Specify the communication function of the
	Single master	channel to be used.
	Slave	Select "Single master".

Table 5-13 Serial Settings for the RL78 Family MCU



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 interrupt priority (INTIICAx)	High	Specify the priority level of the communication
	Level1	end interrupt.
	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 end callback function		Deselect the checkbox.



5.3.4 RZ Family

Select the r_iic_master or r_sci_i2c stack on the [Stack] tabbed page of the FSP Configurator, and the configurable items will be shown on the [Properties] tabbed page.

The following items and values can be specified.

Table 5-14 r_iic_master Settings for the RZ Family MCU

Configurable Item	Value	Description
Common	1	
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 in transmission
Transmission and Reception	Disabled	and reception.
10-bit slave	Enabled	Specify whether to support 10-bit slave
addressing	Disabled	addresses. If you are using rm_hs300x, select "Disabled".
Module g i2c m	haster0 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 bit rate.
	Fast-mode	If you are using rm_hs300x, select "Standard" or "Fast-mode".
	Fast-mode plus	
Rise Time (ns)	120	Specify the SCL rise time to suit the specifications of the target board to be used.
Fall Time (ns)	120	Specify the SCL fall time to suit 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 ϕ cycle.
	3	Removes noise below the 3 IIC ϕ cycle.
	4	Removes noise below the 4 IIC ϕ 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 any setting that is made here is overwritten by the setting in rm_comms_i2c.
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 any setting that is made here is overwritten by the setting in rm_comms_i2c.
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	Priority 0 (highest)	Specify the interrupt priority level of the I2C bus
Level	Priority 1	driver.
	Priority 2	
	Priority 3	
	Priority 4	
	Priority 5	
	Priority 6	-
	Priority 7	
	Priority 8	
	Priority 9	
	Priority 10	
	Priority 11	
	Priority 12	
	Priority 13	
	Priority 14	
	Priority 15	



6. Guide to 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.

6.1 RA Sample Project

Use the following procedures to modify a sample project.

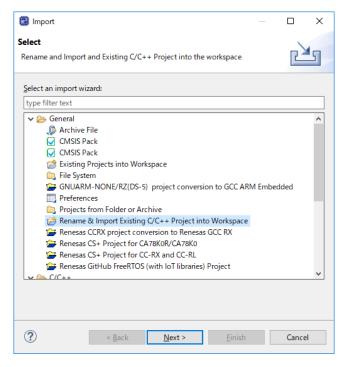
This section describes how to modify the "HS300x_RA6M4_NonOS" sample project so that it can be used on the EK-RA2E1 board as an example.

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

6.1.1 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 the list of imported sample projects and press the [Select Folder] button.

Select Folder			×
\leftarrow \rightarrow \checkmark \uparrow \square > e2_studio >	workspace v ඊ	Search workspace	Q
Organize 🔻 New folder			- ?
e2_studio	^ Name	Date modified	Туре ^
📙 workspace	HS300x_RA6M4_FreeRTOS	5/26/2021 5:27 PM	File folc
.metadata	HS300x_RA6M4_NonOS	5/26/2021 5:30 PM	File fold
HS300x_RA6M4_FreeRTOS	HS300x_RL78G14_NonOS	5/19/2021 8:41 PM	File folc
HS300x RA6M4 NonOS	HS300x_RX65N_FreeRTOS	5/25/2021 5:15 PM	File fold
HS300x_RL78G14_NonOS	HS300x_RX65N_NonOS	5/25/2021 3:03 PM	File folc
HS300x_RX65N_FreeRTOS			~
HS300x_RX65N_NonOS	~ <		>
Folder: HS300x	_RA6M4_NonOS		
		Select Folder Ca	ancel

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

📴 Import		_	\times
Rename & Import P	roject	F	
Select a directory to se	arch for existing Eclipse projects.		
Project name: HS30	k_RA2E1_NonOS		
✓ Use <u>d</u> efault locat	on		
Location:	C:¥Users¥a5090534¥e2_studio¥workspace¥HS300x_F	Browse	
E	Create Directory for Project		
Choose file system:	lefault \vee		
Import from:			
• Select roo <u>t</u> directe	ry: C:¥Users¥xxxxxx¥e2_studio¥workspace ~	B <u>r</u> owse.	
O Select <u>a</u> rchive file	×	B <u>r</u> owse.	
<u>P</u> rojects:			
HS300x_RA6M4_N	onOS (C:¥Users¥xxxxxxx¥e2_studio¥workspace¥H	S300x_RA6M	l4_Nor
<			>
Options			
Keep build config	uration output folders		
(7)	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Canc	el



6.1.2 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" on the [BSP] tabbed page.

When selecting a Renesas board, you will only need to modify the "Board" setting.

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.

@ [HS300x_RA2	E1_NonOS] FSP Configuration 🔀			
Board Supp	oort Package Configuration		Generate Project Conte	ant
			🐯 Restore Defau	lts
Device Selection	on			
	3.0.0-rc1+20210426.9fd4d31a	E	Board Details Evaluation kit for RA6M4 MCU Group	
Board:	EK-RA6M4 ~ Custom User Board (Any Device)		Visit https://www.renesas.com/ra/ek-ra6m4 to get kit user's manual, quick start guide, errata, design package, example projects, etc.	
Device:	EEK-RA2L1			
RTOS:	EC-RA2A1 EC-RA2A1 EC-RA2A1 EC-RA4A0 EC-RA4A02 EC-RA4A03 EC-R			
Summary BSP	Clocks Pins Interrupts Event Links St	tacks Comp	ponents	

Set up the clocks on the [Clocks] tabbed page.

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

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

황 *[HS300x_RA2E1_NonOS] FSP Configuration 없			-
Clocks Configuration			Generate Project Conter
			🔣 Restore Defau
XTAL 20MHz		KLK Div /1	✓ → ICLK 48MHz
LOCO 32768Hz		→ PCLKB Div /2	✓ → PCLKB 24MHz
MOCO 8MHz	Clock Src: HOCO	✓ PCLKD Div /1	✓ PCLKD 48MHz
SUBCLK 32768Hz	_		
HOCO 48MHz V	_		
	CLKOUT Disabled	✓ → CLKOUT Div /1	✓ → CLKOUT 0Hz
ummary BSP Clocks 📀 Pins Interrupts Event Link			



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 will be automatically assigned.

*(HS300x_RA6M4_NonOS) FSP Conf Pin Configuration	iguration 33				- : 0
Select Pin Configuration		E Export to CS	file 📧 Configure Pin Driver V	Naminos	Generate Project Conten
RANMA EX	Manage configurations		te data: g_bsp_pin_cfg		
Pin Selection	🕀 📄 🖓 Pin Con	figuration			Cycle Pin Group
Type Hirst tot	Name	Allow Holes	Lee		
In Function Fin 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		\times
Multiple Pin Configuration Management		
Modify pin configuration list or import/export external file		
RA6M4 EK (Current) R7FA6M4AF3CFB.pincfg	Add	
RA2E1 EK	Remove	e
R7FA2E1A92DFM.pincfg	Rename.	
	Duplicat	e
	Merge to	
	Import Export	
	OK	



RA Family, RX Family, RL78 Family, RZ Family

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					Generate Project Conten
Select Pin Configuration		📑 Expor	t to CSV file	Configure Pin	Driver Warnings
RA2E1 EK	✓ Manage configurations	Generate data: g_bsp_pin_cfg_2e1			
Pin Selection 📰 🖽 📮	Pin Configuration				😲 Cycle Pin Group
Type filter text	Name	Value	Lock	Link	
✓ ✓ Peripherals ∧	Pin Group Selection	Mixed			
> Analog:ACMP	Operation Mode	Simple I2C			
> ✓ Analog:ADC	✓ Input/Output			$\langle \rangle$	
> ✓ Analog:ADC	TXD1	None		\Rightarrow	
> ✓ Connectivity:IIC	RXD1	None		\Rightarrow	
✓ ✓ Connectivity:SCI	SCK1	None		\Rightarrow	
✓ SCI0	CTS1	None		\Rightarrow	
✓ SCI1	SDA1	✓ P401	<u> </u>	\Rightarrow	
 SCI2 SCI9 	SCL1	✓ P402			
✓ Connectivity:SPI Input:CTSU					
> Input:ICU > Input:KINT	<	1	1		3
> Monitoring:CAC > System:CGC >		ple I2C mode, ensure por between I2C and other n			ı drain.



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".
-----------------------	---------	-------	-------	--------

[HS300x_RA2E1_NonOS] FSP Configuration	×			- 8
Pin Configuration				O Generate Project Content
Select Pin Configuration		Export to CSV fi	le 🔚 Configure Pin Driver War	mings
RA2E1 EK	<u>Manage configurations</u>	🗹 Generate d	data: g_bsp_pin_cfg_2e1	
Pin Selection $ \blacksquare \blacksquare \blacksquare \downarrow^{a}_{*Z} $	Pin Configuration			📢 Cycle Pin Group
Type filter text ✓ ▲ Ports △ ▲ P0 > ▲ P1 > ✓ P2 ▲ P3 > ✓ P4 > ✓ P5 > ✓ P9 > ✓ Other Pins ✓ Preripherals > Analog:ACMP > ✓ Analog:ADC > ✓ Analog:ANALOG ✓ Connectivity:IIC > ✓ Connectivity:SCI > ✓ Connectivity:SPI Input:CTSU Input:ICU > Input:KINT > Monitoring:CAC > System:SYSTEM > Timers:AGT	Name	Value Value	Link	
Pin Function Pin Number				



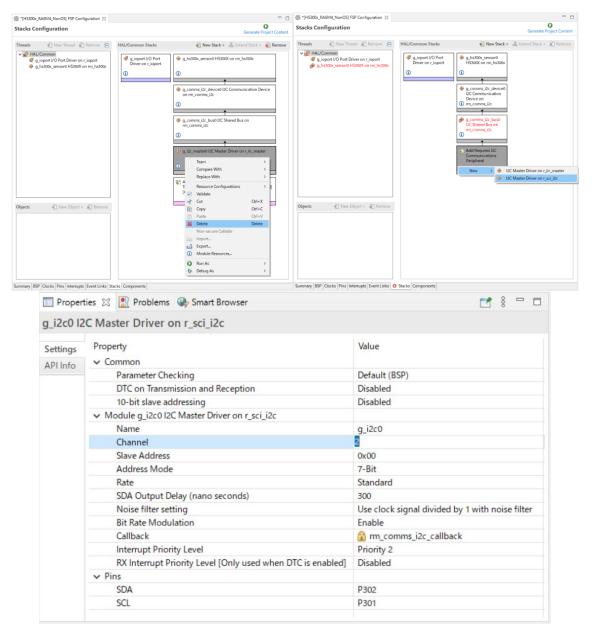
Modify the configuration of individual components on the [Stacks] tabbed page.

Modify the settings for r_iic_master or r_sci_i2c to suit 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_mster} " 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 it 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".

Stacks				
	Configuration			Generate Project Conte
Threads	🐑 New Thread 📓 Remove 📄	HAL/Common Stacks	🕢 New Stack > 🚊	🚆 Extend Stack > 🛛 🙀 Remov
	IAL/Common g_ioport I/O Port (r_ioport) g_hs300x_sensor0 HS300X Temperature/Humidity hs300x_delay Timer, General PWM (r_gpt)	g_ioport I/O Port (r_ioport) ()	 g_hs300x_sensor0 HS300X Temperature/Humidity Sensor (rm_hs300x) g_comms_i2c_device0 I2C Communication Device (rm_comms_i2c) g_comms_i2c_bus0 I2C Shared Bus (rm_comms_i2c) g_i2c0 I2C Master (r_sci_i2c) 	 hs300x_delay Timer, General PWM (r_gpt) i
ummarv	BSP Clocks Pins Interrupts Event Links Stacks	Components		
	e 🔲 Renesas Debug Virtual Console 🔲 Propertie		Smart Manual 🖙 Progress 🍛 Smart Browser	1 8 1
			· · · · · · · · · · · · · · · · · · ·	
ioport	t I/O Port (r joport)			
	t I/O Port (r_ioport)	Value		
Settings	Property	Value	70)	
ettings	Property Parameter Checking	Value Default (B	3P)	
ettings	Property Parameter Checking V Module g_ioport I/O Port (r_ioport)	Default (B	5P)	
ettings	Property Parameter Checking V Module g_ioport I/O Port (r_ioport) Name	Default (B g_ioport	SP)	
ettings	Property Parameter Checking V Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source	Default (B g_ioport Disabled	SP)	
ettings	Property Parameter Checking V Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source	Default (B g_ioport Disabled Disabled	SP)	
ettings	Property Parameter Checking V Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 3 ELC Trigger Source	Default (8 g_ioport Disabled Disabled Disabled	SP)	
ettings	Property Parameter Checking V Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 3 ELC Trigger Source Port 4 ELC Trigger Source	Default (B g_ioport Disabled Disabled Disabled Disabled	5P)	
ettings	Property Parameter Checking V Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 3 ELC Trigger Source Port 4 ELC Trigger Source Port B ELC Trigger Source	Default (8: g_ioport Disabled Disabled Disabled Disabled Disabled	5P)	
ettings	Property Parameter Checking V Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 3 ELC Trigger Source Port 8 ELC Trigger Source Port C ELC Trigger Source	Default (B g_ioport Disabled Disabled Disabled Disabled Disabled Disabled	5P)	
ettings	Property Parameter Checking V Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 4 ELC Trigger Source Port C ELC Trigger Source Port D ELC Trigger Source	Default (B g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled	SP)	
ettings	Property Parameter Checking V Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 3 ELC Trigger Source Port 4 ELC Trigger Source Port B ELC Trigger Source Port D ELC Trigger Source Port E ELC Trigger Source Port E ELC Trigger Source	Default (8: g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled		
Settings	Property Parameter Checking V Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port B ELC Trigger Source Port D ELC Trigger Source Port D ELC Trigger Source Port E ELC Trigger Source	Default (B g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled		
J_ioport	Property Parameter Checking V Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 3 ELC Trigger Source Port 4 ELC Trigger Source Port B ELC Trigger Source Port D ELC Trigger Source Port E ELC Trigger Source Port E ELC Trigger Source	Default (8: g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled		

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

Press [Generate Project Content] to generate files.

Build the project.

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

6.1.3 Changing toolchain setting

If you want to use a toolchain other than the GCC ARM Embedded toolchain, copy RA_HS300X.c (Non-OS) or hs300x_sensor_thread_entry.c, sensor_thread_common.c, and sensor_thread_common.c (FreeRTOS, Azure) from this project to create a new project.



6.2 RX Sample Project

Use the following procedures to modify a sample project.

This section describes how to modify the "HS300x_RX65N_NonOS" sample project so that it can be used on the RSKRX231 board as an example.

6.2.1 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.

Import			×
Select			-
Rename and Import and Existing C/C++ Project into the workspace			-
Select an import wizard:			
type filter text			
V 🗁 General			^
🕼 Archive File			
CMSIS Pack			
CMSIS Pack			
😭 Existing Projects into Workspace			
🚔 File System			
GNUARM-NONE/RZ(DS-5) project conversion to GCC ARM I	Embed	ded	
Preferences			
Projects from Folder or Archive			
Rename & Import Existing C/C++ Project into Workspace			
Renesas CCRX project conversion to Renesas GCC RX			
Renesas CS+ Project for CA78K0R/CA78K0			
Renesas CS+ Project for CC-RX and CC-RL Renesas GitHub FreeRTOS (with IoT libraries) Project			
C/C++			~
(?) < <u>Back</u> <u>Next</u> > <u>Finish</u>		Cance	4

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

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

Select Folder			×
← → × ↑ 📙 > e2_studio > v	vorkspace v Č	Search workspace	Ą
Organize 🔻 New folder		833	• ?
HS300x_RA6M4_FreeRTOS HS300x_RA6M4_NonOS HS300x_RL78G14_NonOS HS300x_RX65N_FreeRTOS HS300x_RX65N_NonOS	 Name HS300x, RA6M4_FreeRTOS HS300x, RA6M4_NonOS HS300x, RL78G14_NonOS HS300x, RX65N_FreeRTOS HS300x, RX65N_NonOS 	Date modified 5/26/2021 5:27 PM 5/26/2021 5:30 PM 5/19/2021 8:41 PM 5/25/2021 5:15 PM 5/25/2021 3:03 PM	Type File folc File folc File folc File folc
_	× <		>
Folder, Fiscory		Select Folder Ca	ancel



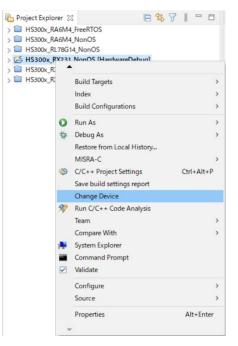
Enter the project name, select the original project for the current device, and press the [Finish] button.

📴 Import	_	
Rename & Import Proje	ct	
Select a directory to search	for existing Eclipse projects.	
Project name: HS300x_R	(231_NonOS	
Use <u>d</u> efault location		
Location: C:¥U:	sers¥a5090534¥e2_studio¥workspace¥HS300x_F	B <u>r</u> owse
Cre	eate Directory for Project	
Choose file system: defau	ult \sim	
Import from:		
• Select roo <u>t</u> directory:	C:¥Users¥xxxxxx¥e2_studio¥workspace ~	Browse
O Select <u>a</u> rchive file:	 	B <u>r</u> owse
Projects:		
HS300x_RX65N_NonOS	S (C:¥Users¥xxxxxxx¥e2_studio¥workspace¥HS	300x_RX65N_Non(
<		>
Options		
Keep build configurati	on output folders	
?	<u>Back</u> <u>N</u> ext > <u>Finish</u>	Cancel



6.2.2 Changing the Device

Select the imported project from the project tree and right-click on the name to open the context menu. Select [Change Device] from the menu.



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

Refactoring				×
Change Device Select the new of	e device for HS300x_RX231_NonOS			
Current Device: Current Board: B				
Target Board:	RSKRX231			~
		Download a	dditional bo	ards
Target Device:	R5F52318AxFP			
			Unlock Dev	vices
?	< <u>B</u> ack <u>N</u> ext >	<u>F</u> inish	Cance	el
-	hanna	-		



If a warning message appears, read it and check if it will create a problem in continuing with the procedure. Press [Next] to go to the next step.

Refactoring		Х
Change Device Review the information provided in the list below. Click 'Next >' to view the next item or 'Finish'.		
Found problems		00
A This change cannot be undone. Please make sure you backup this	is project	before
<		>
No context information available		
? < <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cance	el

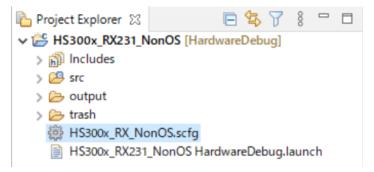
The changes you have made to the settings will be displayed. Press the [Finish] button to apply the changes to the project.

Refactoring			×
Change Device		R	~
The following changes are necessary to perform the refactoring.			
Changes to be performed		₽ û	7 -
✓ 🗹 🔂 Change Device for HS300x_RX231_NonOS			
🗸 🗹 🛃 Launch Configurations			
HS300x_RX231_NonOS HardwareDebug			
> 🖉 🔂 Build Settings			
Project Files			
No preview available			
(?) < <u>Back</u> <u>Next</u> > <u>Finish</u>	1	Cance	el



6.2.3 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; this will open the Smart Configurator window.



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

HS300x,	RY N	005 cc	fo M								-	П
			'y w							1	a	-
Device 9	selec	tion								Generate Code		ort
Device se	election	ı									22	1
Board:	DCV	RX231 (1.00)			×						
						·						
Device:		52318A										
	Dow	nload n	nore boa	rds								
Quantian	Roard	Clocks	Sustam	Components	Dine	Interrupte		 	 			
Jverview b	board	CIOCKS	system	components	PINS	interrupts	5					

Set up the clocks on the [Clocks] tabbed page to suite the specifications of the target board to be used.

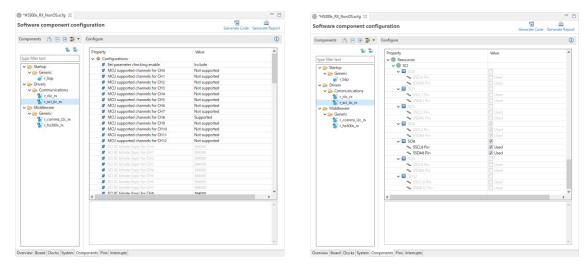
HS300x_RX_NonOS.scfg 33		International Contemporary C	° 8
Clocks configuration	Generate Code Generate Report	Clocks configuration	Generate Code Generate Report
		VC 2 Provide St Provid	Hore M 200 Hore M
M0T-sed carel los-speed door	Les geur tier rédis (JPCLA)	W0 ¹ -seclared low-speed clock	Los gener riner doix (LPCUs) - DR40 - DR40 - R040 - R040- R040 -
	. Seko		- Devia



Modify the settings for individual components on the [Components] tabbed page to suit 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.

Select "SSCL8 Pin" and "SSDA8 Pin" for "SCI8" under "Resources".



Open the [Pins] tabbed page and check that SCI8 pin functions are assigned to the desired pins in the [Pin function] panel.

n configuration				Generate Code	e Generate Rej
lardware Resource 🛛 🕀 📮 💑	Pin Functio	n		ି ଏ 🛯	8 🖬 🔤
Type filter text	type filter	text (* = any str	ing, ? = any character)	All	
TMR2 TMR2 TMR2 TMR2 TMR2 TMR2 SCI TMR2 SCI S	Enabled	Function CTS8# RTS8# RTS8# RXD8 SCK8 SMISO8 SMO38 SSM038 SSB# db SSCL8 db SSCL8 db SSCL8	Assignment Vot assigned Not assigned ProceMICCC/MTCLRA/TMCI2/RDD/SP PC/US/MTCCCA/MTCLRA/TMCI2/RDD/SP Vot assigned Not assigned	MISO8/SSCL8	Yin Number Not assigned Not assigned Not assigned Not assigned Not assigned 46 45 Not assigned
✓ ▲ Comparator B	4				

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

Connecting a sensor board requires a board for converting the PMOD Type 2A interface to PMOD Type 6A.



Press the [Generate Code] icon to generate the code.

*HS300x_RX_NonOS.scfg 🔀				
verview information			G	enerate Code Generate Repo
General Information				?
				Ū.
This editor allows you to modify the settings stored in config	juration file (.sc	rfg)		
Board				
Allow board and device selection				
			Application une	der
Clocks			developmen	t
Allow clock configuration				- Components
			Middley	ware
Components			Device driver	TOS
Allow software component selection and configuration				- Pins
			1000000000	Pins
Dine				
Yins Allow general pin configuration and pin configuration for se nterrupt	elected software	e component		
Allow general pin configuration and pin configuration for se	ation for select	ted software component		
Allow general pin configuration and pin configuration for se nterupt Allow general interrupt configuration and interrupt configur	ation for select	ted software component		
Allow general pin configuration and pin configuration for se nterupt Allow general interrupt configuration and interrupt configur Dick here to get more information on <u>User's Manual Releas</u>	ration for select	ted software component ation Notes Tool News		
Allow general pin configuration and pin configuration for se nterupt Allow general interrupt configuration and interrupt configur Click here to get more information on <u>User's Manual Releas</u> Current Configuration	ration for select	ted software component ation Notes Tool News I Kbytes , Pin count: 1000	Edit	
Allow general pin configuration and pin configuration for se nterupt Allow general interrupt configuration and interrupt configur Click here to get more information on <u>Lizer's Manual, Releas</u> Current Configuration Selected board/device: R5F52318AxFP (ROM size: 512 Kbytes Senerated location (PROJECT_LOC%): <u>serveme_gen</u> Selected components:	ration for select	ted software component ation Notes Tool News I Kbytes , Pin count: 1000		
Allow general pin configuration and pin configuration for se nterupt Allow general interrupt configuration and interrupt configur- Click here to get more information on <u>User's Manual Release</u> Current Configuration Selected board/device: RSFS2318AsFP (ROM size: 512 Kbyter Senerated location (PROJECT_LOC9): <u>serVsmc.gen</u>	ration for select	ted software component ation Notes Tool News I Kbytes , Pin count: 1000		
Allow general pin configuration and pin configuration for se nterupt Allow general interrupt configuration and interrupt configur- Click here to get more information on <u>User's Manual Release</u> Current Configuration Selected board/device: RF552318AvFP (ROM size: 512 Kbytes Senerated location (PROJECT_LOC%): <u>secVanc.gen</u> Selected components:	ation for select	ted software component ation Notes Tool News Kbytes , Pin count: 100		
Allow general pin configuration and pin configuration for se nterupt Allow general interrupt configuration and interrupt configur Dick here to get more information on <u>User's Manual, Release</u> Current Configuration Selected board/device: RSF52318AxFP (ROM size: S12 Kbytes Senerated location (PROJECT_LOC®): <u>ser Vame, gen</u> Selected components Component M Board Support Packages. (r_bsp) HS3005 (Smoth Midleware (r_bs300x,rx))	ation for select te Note Applica s , RAM size: 64 Version	ted software component ation Notes Tool News Kbytes , Pin count: 100 Configuration	cdit	
Allow general pin configuration and pin configuration for se nterupt Allow general interrupt configuration and interrupt configur- Click here to get more information on <u>User's Manual Release</u> Current Configuration Selected board/device: RSF52318AsFP (ROM size: S12 Kbyter Senerated location (PROJECT_LOCP). <u>ser/smc.gen</u> Selected components: Component © Board Support Packages. (r_bsp)	s, RAM size: 64 Version 5.66	ted software component ation Notes Tool News Kbytes , Pin count: 100 Configuration r_bsp(used)	idit	
Allow general pin configuration and pin configuration for se nterupt Allow general interrupt configuration and interrupt configur Dick here to get more information on <u>User's Manual, Release</u> Current Configuration Selected board/device: RSF52318AxFP (ROM size: S12 Kbytes Senerated location (PROJECT_LOC®): <u>ser Vame, gen</u> Selected components Component M Board Support Packages. (r_bsp) HS3005 (Smoth Midleware (r_bs300x,rx))	s, RAM size: 64 Version 5.66 0.62	ied software component ation Notes Tool News (Rbytes, Pin count: 100) Configuration r_bsp(used) r_hs300_r(used)	idit	

Build the project.

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

6.2.4 Changing toolchain setting

If you want to use a toolchain other than the CC-RX toolchain, copy RA_HS300X.c (Non-OS), or main.c and hs300x_sensor_thread_entry.c (FreeRTOS), or hs300x_sensor_thread_entry.c, sensor_thread_common.c, and sensor_thread_common.c (Azure) from this project to create a new project.



6.3 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 the RSK RL78/G1G board.

6.3.1 Creating a New Project

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

📴 New C/C-	Project — 🗆	Х							
Templates f	r Renesas RL78 Project								
All C/C++	GCC for Renesas RL78 C/C++ Library Project A C/C++ Library Project for Renesas RL78 using the GCC for Renesas RL78 Toolchain.	^							
	LLVM for Renesas RL78 C/C++ Executable Project A C/C++ Executable Project for Renesas RL78 using LLVM for Renesas RL78 Toolchain.								
	LLVM for Renesas RL78 C/C++ Library Project A C/C++ Library Project for Renesas RL78 using LLVM for Renesas RL78 Toolchain.								
	Renesas CC-RL C Executable Project A C Executable Project for Renesas RL78 using the CCRL toolchain.								
	Renesas CC-RL C Library Project A C Library Project for Renesas RL78 using the CCRL toolchain								
?	< Back Next > Finish Cancel								

Enter the project name ("HS300x_RL78G1G_NonOS" in this example) and press the [Next] button.

8		_		Х
New Renesas	CC-RL Executable Project			\$
New Renesas C	C-RL Executable Project			4
Project name:	HS300x_RL78G1G_NonOS			
✓ Use defaul	t location			
Location:	C:¥Users¥a5090534¥e2_studio¥workspace¥HS300x_RL78G1G_NonOS		Browse.	
	Create Directory for Project			
Choose file s <u>y</u> s	tem: default \sim			
Working sets				
Add proje	c <u>t</u> to working sets		Ne <u>w</u>	
W <u>o</u> rking sets			S <u>e</u> lect	
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish		Cance	I



Change "Target Device" to the desired device (R5F11EFA in this example) and press the [Next] button.

0							8	_
Device Selection								
You can filter devices by	regular ev	pression					New Renesas CC-RL Executable Project	
rou currinter devices by	regulares	pression					Select toolchain, device _debug settings	
Search Device								
Device	RAM	ROM	Pin	RTOS	Smart Co	Peripher	Toolchain Settings	
> RL78 - F1E							Language: C C++	
> RL78 - G10							Toolchain: Renesas CCRL V	
> RL78 - G11							Toolchain: Renesas CCRL ~	
> RL78 - G12							Toolchain Version: v1.10.00	
> RL78 - G13								
> RL78 - G13A							Manage Toolchains	
> RL78 - G14							Device Settings	Configurations
> RL78 - G1A							Device Settings	Configurations
> RL78 - G1C							Target Board: Custom ~	Create Hardware Debug
> RL78 - G1D								E4 (E20 (BL 70)
> RL78 - G1E								E1/E20 (RL78)
> RL78 - G1F							Target Device: R5F11EFA	
✓ RL78 - G1G							Unlock Devices	Create Debug Configurat
> RL78 - G1G 30pin							Unlock Devices	RL78 Simulator
> RL78 - G1G 32pin							Endian: Little 🗸 🗸	
 RL78 - G1G 44pin 								Create Release Configura
> R5F11EF8							Project Type: Default	
R5F11EFA								
R5F11EFA	1 KB	16 KB	44		×	√		
> RL78 - G1H								
> RL78 - G1K								

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

9			×
New Renesas CC-RL Executable Project			4
Select Coding Assistant settings			4
Use Smart Configurator ^{&}			
Use Peripheral Code Generator			
The e2 studio peripheral code generator automatically generates programs (device drivers) for MCU periph timers, serial interfaces, A/D converters, DMA controllers, etc.) based on settings entered via a graphical us Functions are provided as application programming interfaces (APIs) and are not limited to initialization of	er interface	e (GUI).	
Automatic			
UART Timer generation Application under CSI Timer of peripheral development			
A/D But Software			
DMA Clock driver RTOS	ller		
"TTTTTTTTTT"			
	ish	Cance	9

Press the [Finish] button.

Ø				_		×
New Renesas CC-RL Ex Summary of project "HS30	-					2
TOOLCHAIN NAME: TOOLCHAIN VERSION : GENERATION FILES : generateKata.ssm generateKstknint.asm generateKstknint.asm generateKstknint.ssm generateKstknint.ssm	Renesas CCRL v1.10.00					~
?		< <u>B</u> ack	<u>N</u> ext >	Einish	Cance	I



6.3.2 Settings of the Code Generator

Modify the pin assignment on the [Pin assignment] tabbed page for "Peripheral Functions" to suit the specifications of the target board to be used.

Peripheral Fun	🛛 📓 Coi	de Preview 🎾	Device Top View 🚆 De	evice List View 🔲 Prop				
						nerate Code	0	000
		Block diagram	On-chip debug setting	Confirming reset sour	ce Safety functi	ons		^
Pin assignment s	etting							
Once the pin assi	gnments have	been fixed it is r	ot possible to change th	em later.				
A new project mu	st be created t	o change the se	tings.					
PIOR register	Function	Port setting		7				
PIOR11, PIOR10	TRJIO0	P01		1				
PIOR13, PIOR12	TRJOI	P30		~				
	1							
								Ų

Modify the clock settings on the [Clock setting] tabbed page for "Peripheral Functions" to suit the specifications of the target board.

emia							_	_
💯 Peripheral Fun 🔀 🛃 Code Previ	iew 📲 Devi	ce lop View 💥 Di	evice List View	V 📋 P	roperti	P.244		•
handling						🐻 Generate Code	<u>()</u>	8
-	diagram On-	-chip debug setting	Confirming	reset so	ource	Safety functions		î
-Operation mode setting		~						
● High-speed main mode 4.0 (V) ≤ ○ High-speed main mode 3.6 (V) ≤								
- ·								
○ High-speed main mode 2.7 (V) ≤								
O Low-speed main mode 2.7 (V) ≤	VDD ≤ 5.5 (V	7						
- Main system clock (fMAIN) setting		0.1F1 1	/0.	0.0				
High-speed OCO (fIH)		 High-speed sys 	tem Clock (TIV	120				
-High-speed OCO clock setting	_							
Operation	Frequency	48 (fHOCO=48, fIH	1=24)	~ (MHz)			
 High-speed system clock setting 								
Operation								
X1 oscillation (fX)		 External clock i 	nput (fEX)					
Frequency		5		()	MHz)			
Stable time		2^18/fX	52428.8	(us)			
-Low-speed oscillation clock (fIL) settin	ne							
Frequency		15		0	<hz)< td=""><td></td><td></td><td></td></hz)<>			
-Interval timer operation clock/Timer R	J count sourc	e setting						
Interval timer operation clock/Time	r RJ count so	uri Stop	✓ 15		(kHz)			ł
-CPU and peripheral clock setting								
CPU and peripheral clock (fCLK)		fIH	24000	0	<hz)< td=""><td></td><td></td><td></td></hz)<>			
<							>	~
							-	



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

💯 Peripheral Fun 🙁 🛒 Code Preview 🚆 Device Top View 🚆 Device List View 🔲 Properties 🌉 FIT C	Configura	
	enerate Code	3
Pin assignment Clock setting Block diagram On-chip debug setting Confirming reset source Safety func	tions	^
-On-chip debug operation setting		
O Unused		
- RRM function setting		
O Unused		
-Security ID setting		
Use Security ID		
Security ID 0x0000000000000000000000000000000000		
-Security ID authentication failure setting		
O 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" on the [Serial Array Unit] or [Serial] tabbed page.

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

🕎 *Peripheral F	u 🖾	ode Preview	v 📲 Device Top View	🔛 Device List View	Properties			
						🐻 Generate Code	٥	8
Serial Array Uni								^
	TO UAP	RT1 CSI00 IC	00					
Function								
Channel 0	IC00	~						
Channel 1	Unuse	d v						
Channel 2	Unuse	d v						
Channel 3	Unuse	d v						
_								¥
<)	



On 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".

💯 *Peripheral Fu 💥 🛒 Code Preview 💯 Dev	vice Top View 📃 Device List View	TI Properties 💯 FIT Configura 🖙 🔞 🕅 Properties 🕅 Generate Code	
Serial Array Unit 0		(E) Generate Code	8
Channel UARTO UARTI CSIOO ICOO			_
- Transfer rate setting			
Transfer rate	100000 ~	(bps) (Actual value: 100000)	
-Interrupt setting			
Transfer end interrupt priority (INTEC00)	Low ~]	
-Callback function setting			
Master transmission end	🖂 Master reception end	Master error	
4			

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

👮 *Peripheral Fu 🔀 📑 Code	Preview 🕎 Device Top View	🕎 Device List View	Properties	💯 FIT Configura		
	,	/1m		Generate Code		
Transfer mode Setting				_		^
home and here and her					_	
 Unused 						
Single master						
◯ Slave						
<					>	×



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

👮 *Peripheral Fu 🔀 📑 Code Preview	🕎 Device Top View	🕎 Device List View	Properties	💯 FIT Configura		
				🐻 Generate Code	0	8
Transfer mode <u>Setting</u>						^
_Clock mode setting						
◯ fCLK 🚺	fCLK/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 interrupt priority () Callback function setting		v				
Master transmission end	Master reception e	nd 🖂 Mas	ter error			
-Callback function enhanced feature settin	e					
Generated stop condition in master		n end callback function	n			
						ł
						`
<)	

On 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 Fu	🔀 📑 Code Prev	view 📲 Device Top Vi	ew 📲 Device List View	Properties			
					🐻 Generate Code	0	8
	Channel 0 Channel	1 Channel 2 Channel	el 8				1
Functions							
Channel 0	Interval timer		\sim				
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 μ s", enable timer interrupts, and set the interrupt priority to a desired level.

🕎 Peripheral Functions 🗙		🐻 Generate Code	0	000		
TAU0 TAU1 TMRJ0 TMRD0 TMRD1 TMRG						^
General setting Channel 0 Channel 1 Channel 2	Channel 3					
_Interval timer setting						
Interval value (16 bits)	100 µs 🗸 (Actual value: 100)					
Generates INTTM00 when counting is started						
_Interrupt setting						
End of timer channel 0 count, generate an inter	rupt (INTTM00)					
Priority	Low ~					
						¥
<					>	

Press the [Code Generate] button to generate the code.

6.3.3 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.

Directive for including r_comms_i2c_if.h:

Addition of the rm_comms_i2c_bus0_callback() function as 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
* Arguments : None
* 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:

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.

Declaration of prototype for each function:

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

```
/* Open the Bus */
g_comms_i2c_bus0_quick_setup();
   /* Open HS300x */
g_hs300x_sensor0_quick_setup();
   while (1U)
   {
     start_hs300x_demo();
   }
```

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

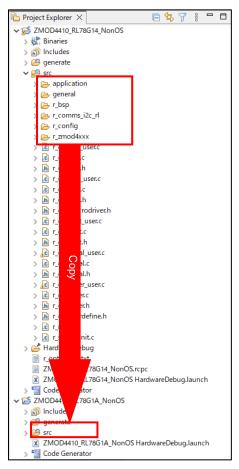
```
void g_comms_i2c_bus0_quick_setup(void)
{
    /* bus has been opened by startup process */
}
void demo_err(void)
{
    while(1)
    {
        // nothing
    }
}
```



6.3.4 Modifying Sample Source Files

Right-click on the "application" "general" "r_bsp" "r_comms_i2c_rl" "r_config" "r_hs300x" folder in the project tree of the "HS300x_RL78G14_NonOS" sample project and select [Copy] from the context menu.

After that, 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 the "r_comms_i2c_rl_config.h" file 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 */	
<pre>#define COMMS_I2C_CFG_BUS0_DRIVER_CH</pre>	(0) /* Channel No. */

When channel 0 of the serial interface IICA is used:

For the other definitions, refer to section <u>5, Configuration Settings</u>.

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

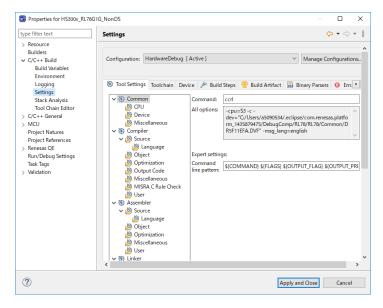
Modify "r_cg_serial.h" to "r_cg_sau.h" or "r_cg_iica.h":

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.

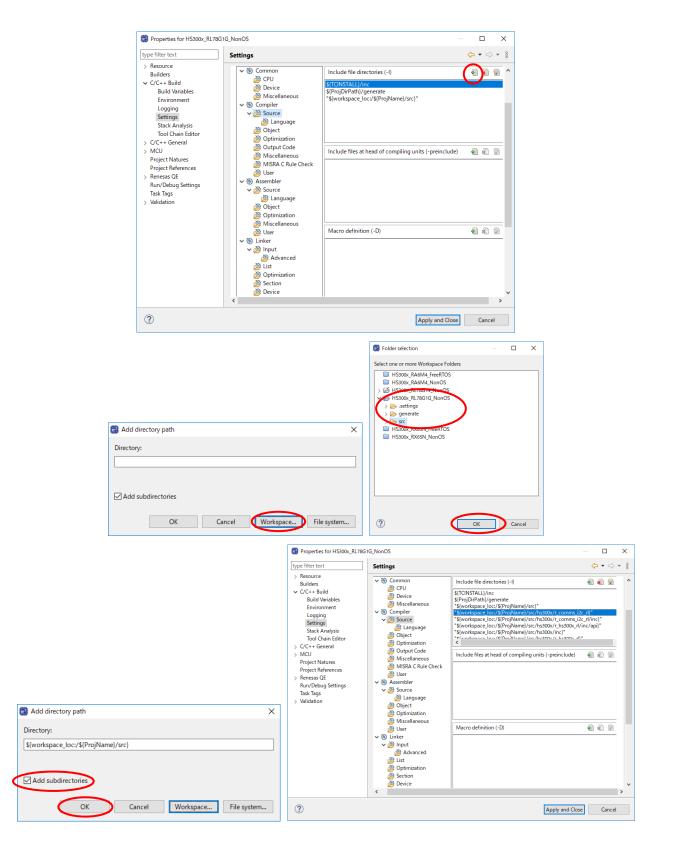




Select [Compiler] \rightarrow [Source] on the [Tool Settings] tabbed page and press the [Add] icon.

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 from the list and press the [OK] button.

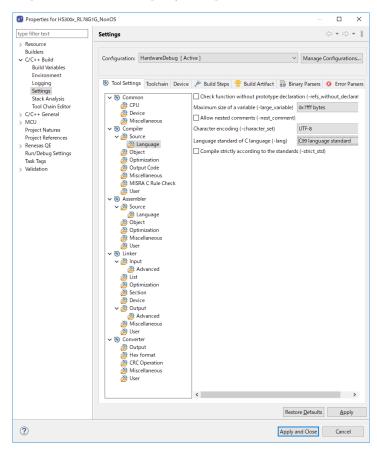
Select the checkbox for "Add subdirectories" and press the [OK] button.





Select [Compiler] \rightarrow [Source] \rightarrow [Language] on 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.



Build the project.

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



6.4 RZ Sample Project

Use the following procedures to modify a sample project.

This section describes how to modify the "HS300x_RZG2L_NonOS" sample project so that it can be used on the RZ/G2L Evaluation Kit (SMARC) board as an example.

6.4.1 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	
Select an import wizard:	
type filter text	
V 🍃 General	7
📮 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	
Project	
Cample Projects on Renesas Website	
✓ C/C++ C/C++ Executable	
	_
Mext > Finish Cancel	



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

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

Select Folder			×
$\leftarrow \rightarrow \checkmark \uparrow \square \rightarrow \infty$	xxxxx → e2_studio → import →	✓ Č	n import
Organize 🔻 New folde			:== • ?
🖈 Quick access	Name	Date modified	Type S
Mr Quick access	HS300x_RZG2L_FreeRTOS	7/11/2022 4:27 PM	File folder
📥 OneDrive	HS300x_RZG2L_NonOS	7/11/2022 4:24 PM	File folder
💻 This PC			
i Network			
	<		>
Folder	r: HS300x_RZG2L_NonOS		
	L	Select Folde	r Cancel

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

📴 Import				×
Rename & Import	Project o search for existing Eclipse projects.			
Project name: HS	00x_RZG2L_NonOS			
Use <u>d</u> efault loc	ation			
Location:	C:\Users\xxxxxx\e2_studio\workspace\HS300	x_R	B <u>r</u> owse.	
	Create Directory for Project			
Choose file system	default \vee			
Import from:				
Select root direct	C:\Users\xxxxxx\e2_studio\import\HS3	~	B <u>r</u> owse	
○ Select <u>a</u> rchive fi	le:	~	B <u>r</u> owse	
<u>P</u> rojects:				
HS300x_RZG2L_	NonOS (C:\Users\xxxxxxx\e2_studio\import\HS3	00x_F	RZG2L_Non	OS)
Options				
	iguration output folders			
?	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>		Cance	el



6.4.2 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" on the [BSP] tabbed page.

When selecting a Renesas board, you will only need to modify the "Board" setting.

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.

🇱 [HS300x_RZG	2L_NonOS] FSP Configuration $ imes$		- 8
Board Supp	ort Package Configuration		O Generate Project Content
			🔯 Restore Defaults
Device Select	ion		
FSP version:	1.1.0-rc0+20220712.e6be8f11 ~	Board Details	
Board:	RZ/G2L Evaluation Kit (SMARC) 🗸 🔛		
Device:	Custom User Board (Any Device) RZ/G2LC Evaluation Kit (SMARC)		
Core:	RZ/G2L Evaluation Kit (SMARC) RZ/G2UL Evaluation Kit (SMARC)		
RTOS:	No RTOS ~		
Summary BSP	Clocks Pins Interrupts Event Links Stacks C	Components	

Set up the clocks on the [Clocks] tabbed page.

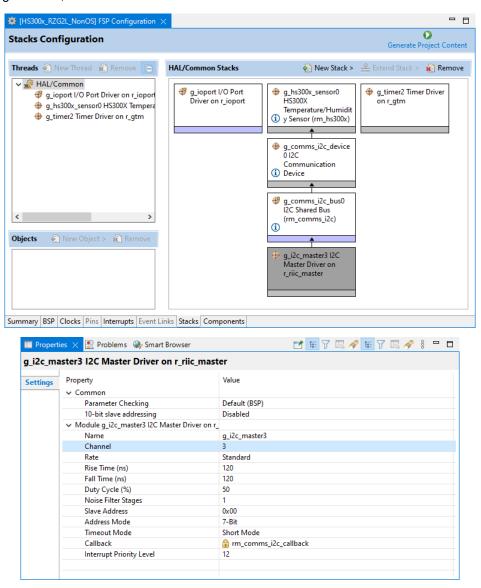
When "Custom User Board (Any Device)" is selected for "Board", set up the clocks to suit the specifications of the target board to be used.

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

[HS300x_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 Components	×



Modify the configuration of individual components on the [Stacks] tabbed page. Modify the settings for r_iic_master to suit the specifications of the target board. On the RZ/G2L Evaluation Kit (SMARC) board, PMOD1 is assigned RIIC3. If you are using PMOD1, set channel to 3.



Press [Generate Project Content] to generate files.

Build the project.

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



6.4.1 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.

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



7. Viewing Temperature and Humidity Data

Use the following procedure to view temperature and humidity data in real time.

After executing debugging, select [Window] \rightarrow [Show View] \rightarrow [Expressions] to open the [Expressions] tabbed page.

🖕 🕼 💼 🕼 Debug 🗸 💽 RA_HSTOCK Desug_Flat	New Window Editor Appearance	0-5-2000	* • 😰 x 0 = 1 = N x, 0 x 0 = 2 = 0 X 0	• • • • • • • •	8.004	9 9 • 1			9 Configuration () Dubug
Debug 22 🛛 🗧 👘 🗄 👘	Show View	> % Breekprünts Alt+9-9	con p @ main.c @ bsp_delay.c		00- Variables	Sreakpol	Project E.	St Expression	22 C ORegis
Classical Concerning and Concer	Parapectors Hotigator Parlement Parlement	Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 Image: Construct And Section 2016 <td>1</td> <td></td> <td>Expression</td> <td>Type</td> <td>Value</td> <td>Addres</td> <td>in (1 ≥ 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 =</td>	1		Expression	Type	Value	Addres	in (1 ≥ 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 =
54	54			~					
	RA HSIXIN Delwg F	lat (Berenan GDB Hardware Deingging) Select, writing to address 8x810	DebuggerConsole Wext	>			n 26 % B	61 % S 9	
	タウンボード除す ハードウェア・ブレージボ ハードウェア・ブレージボ ハードウェア・ブレージボ ハードウェア・ブレージボ ハードウェア・ブレージボ	Select, writing to address evelu (>NG7FL2ex30.[38](4) (>NG7FL2ex30.[38](4) (>NG7FL2ex72.[38](4) (>NG7FL2ex72.[38](4) (>NG7FL2ex72.[38](4) (>NG7FL2ex72.[38](4) (>NG7FL2ex72.[38](4)	9220 «Ith data fffaffffffffffffffffffffffffffffff						

Click on [Add new expression] on the [Expressions] tabbed page and add "gs_hs300x_data".

S S Debug V HS300k, RA6M4, NonOS Debug	RYO B.	8018-	5-2000	14-13(%)×10	• II = N 3.9.1	[1 号] (合 3]	Q + Q + 10 + 8 II	四天夏冬春月。	0.00			
									-			C++ Debug
Debug 11 B 1 H 1 Im Im	le statupe		"hal deta.h"					00-Vari 🗣 Bre 🛋 Mod				
 W HS300x RA6M4 NonDSaff 11 Icores 01 	2							1000 C		6.6.4		ione : gs_hs3i
✓ P Thread #11 (single core) [core 0] (Suspended : Step)	3	/* Sequen						Expression	Type	Value		Details:()
start_demo() at RA_HS300K.cr72 0v2b8	- A -	- typedef e	num e_demo_sequence	(v 遭 qs_hs300x_data	volatile m_s.		0x20000	Default:(
hal_entry() at hal_entry.c:15.0x370	6	DEVID	SEQUENCE J = (1),						int16.t	0	0x20000 0x20000	Decimal:{ Hex:{}
main() at main.cl5 0x38e	7	DEVID	sequence 2,					be- integer_part 66- decimal_part	int16.1	0	0x20000	Binery IC.
am-nore-eabi-gdb (7.8.2)	1.1		SEQUENCE 3, SEQUENCE 4.					v (B temperature	mt hs300x		0x20000	Octal:{
Renesas GDB server (Host)	18		SEQUENCE 5.					64- integer part	int16 t	0	0x20000	
	11		SEQUENCE_6,					00- decimal part	int16.1	0	0x2000X	
	12 13) demo_se	dneuce_t1					Add new expression				
	14	/* Callba	ck status */									
	15		mum e demo callbaci	_status								
	16	(
	17		CALLBACK_STATUS_WAL CALLBACK_STATUS_SUR									
	10	DEVID	CALLBACK STATUS ARA	MAT.								
	20) demo_ca	ilback_status_t;									
	23		veloper Assistance	10.000.00000000								
	22		ems 12c bus0 quick									
	24		300x sensor8 quick									
	25											
	26	veid	start_demo(void); id demo err(void);									
	28	static vo	rg geno_erv(void);									
	29	static vo	latile demo callbas	ck_status_t_gs_damo_ci	allback statum;							
	30	static vo	datile rs hslobs de	sta_t gs_hs300x	data;							
	31 32	-	setup for a comis 1									
	32 00000754	- Quick	ens 12c bus0 guick	setun(unid)								
	34	(and Ter have dered	(second (second)								
	35	fsp_e	rr_t err;									
	35 09000240 37	iZc_m	ester_instance_t *	p_driver_instance = (i2c_mester_instance_t	*) g_comms_i2c_bus@	extended_cfg.p_driv					
	38	(5:00	en 120 deluer, this	s must be done before -	calling new COMPS and	n #27						
	39 00000246	err -	a driver instance	->p api->coen(p driver	instance->p ctrl. p	driver instance->p cf	171					
	48 00000258	asser	t(FSP_SUCCESS == et	1():								
	41 42	2										
	114	1.111.00					· · ·	1				< > >
							,	K				
				ms 🙀 Debugger Console 🛛	あ、スマート・ブラウザー [] M	emory			H 26 (b) (b)	PT 16	8 2 0	· * * * *
		+OS Debug_Rat	Renesas GDB Hardware	Debugging)								
	ターゲット接続終了											
	GDB: 61071 ターゲット補続状況 -											
	ターゲット構成状況 - ターゲット構成状況 -											
	92/0-1984											
	Option Function			0100a100 with data fff								
		Select, wri	ting to address God	0100s200 with data fff	dfffffffffffffffffffffffff	ffffffff						
	\$5.0-F887		and the second second									
	ハードウェア・ブレータボ	I VIENTUXES	concroscity,									
c	6 C	THE OWNER.										
SSDSecure doc00002bd	404 ms	DWT	-81								2.40	0=20



Right-click on the added variables and select [Enable Real-time Refresh].

🚯 🔳 🕸 Debug 🗸 🗹 HS3006, RAKM4, NovOS Debug, Fo	•○ 1世·回归(句·\$•回》(四日》)、[新闻》(四日》(四日))(句·2)(句·3)(句·\$•回)(句·2)(句·2)(句·2)(句·2)(句·2)(句·2)(句·2)(句·2	杨 다 田 남 🗑 🕹	の何・臣・臣の	· · · ·	
				Q 18 5	COC++ to Debus
Nbug 27 19 1 = 0	a startupic a mainic a RA H5300Lc 12	=	e 📸 Mod 🚡 Proj 🚥 B	- X -15 3.Pr	ni 🖬 10 😐 🖸
1 Holds, Mark Mark Mark Mark Mark Mark Mark Mark	<pre>1 fields "signed"</pre>	▲ Lowester v cft colored v cft	Type Select Al Copy toposatorial Remove Al Remove Al Remove Al Remove Al Shave Details As Add Watch Expression.	Vive Addre Cahi C boo Cahi C boo Cahi C boo Doo Doo Doo Doo Doo Doo Doo Doo Doo	k k control of the second sec
		> <		7	
	Console 12 JW Registers [] Debug Shell 🛃 Problems 😨 Debugger Console 🌒 27-1-75915- 🛛 Memory		# X % 8	12895	0 + M + * *
	DBM JABAH (Hand) Shong Jit (Drones GAR Headwar Dhoogyng) 				

Start debugging, and the temperature and humidity values will be updated in real time.

🖕 😰 Debug 🗸 🐑 HS300x_RA6M4_NorrOS Debu	(1) (日・日日) 8・4・日 (1) (8 (4)・1) (1) (1) (1) (1) (1) (1) (1) (1) (1) (· · · · · · · · · · · · · · · · · · ·			
Control (Control (Contro) (Control (Contro) (Contro) (Contro) (Contro) (Contro) (Contro)	0. mode: 0. mode:	Comparison Compar	Q. M Q. C. BAR		
	Console 11 III Registers (1) Debug Shell 🖉 Problems 🚱 Debugger Console 🔿 27-1-7979- 👔 Memory				
	HSDDx, RAUM4, NaxOS Debug, Filt (Remeas GDB Hardware Debugging)				
	>>BBMTY main table ->>>BBMTY ->>>BBMTY ->>>BBMTY ->>>BBMTY ->>>BBMTY ->>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>				



Revision History

		Description		
Rev.	Date	Page	Summary	
1.00	June 30, 2021	-	First Release	
1.10	September 30,	P10, P13	Add: programming mode API	
	2021	P7, P14,	Add: RE01 256KB items	
		P36, P41,		
		P45, P56		
1.20	December 20,	-	Add: Support multiple ZMOD sensors usage	
	2021		Add: Supoort RX Azure	
			Other minor changes	
1.30	March 1, 2022	P31	Add: Support multiple I2C devices	
1.40	May 12, 2022	P4,P5,	Changed: RA tables	
		P9,P10	Changes: RX tables	
			Changed: programming mode	
			Changed: RE01 256KB / 1500KB tables	
1.50	August 30, 2022	-	Add: RZ items	
1.51	March 3, 2023	-	Updated: Environments for RL78	
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	
1.53	September 7,	-	Updated: Guide for Changing the Target Device	
	2023		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 must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

- Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- 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.
- Renease Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renease 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
- (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/.