

RA Family, RX Family, RL78 Family, RZ Family HS400x Sample Software Manual

Introduction

This application note describes the sample software that is for use with the HS400x humidity and temperature sensor and runs on certain MCUs of the RA family, RX family, RL78 family, and RZ family.

Target Devices

RA6M4 Group RA0E1 Group RX65N Group RL78/G14 Group RL78/G23 Group RZ/G2L Group

Trademarks

FreeRTOS[™] and FreeRTOS.org[™] are trade marks of Amazon Web Services, Inc. Microsoft[®] Azure RTOS is a trademark of the Microsoft group of companies. All other trademarks are property of their respective owners.

Contents

1.	Overview	4
2.	Environment for Confirming Operation	4
2.1	Environment for Confirming Operation on the RA6M4 Group MCU	
2.2	Environment for Confirming Operation on the RA0E1 Group MCU	6
2.3	Environment for Confirming Operation on the RX Family MCU	7
2.4	Environment for Confirming Operation on the RL78/G14 Group MCU	
2.5	Environment for Confirming Operation on the RL78/G23 Group MCU	9
2.6	Environment for Confirming Operation on the RZ Family MCU	10
3.	Sensor Specifications	11
3.1	Overview of Sensor Specifications	
3.2	Sensor Functions	
3.2.1	I2C Communication Interface	
3.2.2	2 Expressions for Converting Output Values to Humidity and Temperature	
4.	Specifications of Sample Software	13



4.1	Configuration of the Sample Software	13
4.2	Specifications of Sensor API Functions	13
4.2.1	List of Sensor API Functions	13
4.2.2	Guide to Using the API Functions	14
4.3	Flowchart of the Main Processing in the Non-OS Version of the Sample Software	
4.4	Flowchart of the OS Version of the Sample Software	
4.5	Azure RTOS Project	
F (04
	Configuration Settings	
5.1	HS400x Humidity and Temperature Sensor Settings	
5.1.1	RA Family	
5.1.2	RX Family	
5.1.3	RL78 Family	
5.1.4	RZ Family	
5.2	Sensor Communication Middleware Settings	
5.2.1	RA Family	
5.2.2	RX Family	
5.2.3	RL78 Family	
5.2.4	RZ Family	
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	
6. (Guide to Changing the Target Device	46
6.1	RA Sample Project	
6.1.1	Importing the Sample Project	
6.1.2	Modifying Settings of the FSP Configurator	
6.1.2.		
6.1.2.		
6.1.2.3		
6.1.2.4		
(1)	I2C Driver	
()	(a) When using the same I2C driver as the original sample project	
	(b) When using the different I2C driver as the original sample project	
(2)	I/O Port	
(3)	Others	
6.1.3	Changing the Toolchain Setting	
6.2	RX Sample Project	
6.2.1	Importing the Sample Project	



6.2.2	Changing the Device
6.2.3	Modifying Settings of the Smart Configurator59
6.2.4	Changing the Toolchain Setting61
6.3	RL78 Sample Project
6.3.1	Creating a New Project62
6.3.2	Settings of the Code Generator64
6.3.3	Modifying the Generated Code68
6.3.4	Modifying Sample Source Files72
6.4	RZ Sample Project
6.4.1	Importing the Sample Project
6.4.2	Modifying Settings of the FSP Configurator
6.4.1	Changing sample code
7. \	/iewing Temperature and Humidity Data82
Revis	sion History
Gene	ral Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products85
Notic	e86
Corp	orate Headquarters
Conta	act information
Trade	emarks



1. Overview

This sample software acquires humidity and temperature data from the HS400x 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 HS400x 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 the RA6M4 Group MCU

The operation of this software has been confirmed on the MCU of the RA6M4 group 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.3.8.0
RTOS	FreeRTOS [™] or Microsoft [®] Azure RTOS
Emulator	On-board debugger (J-LINK)
Interposer	Interposer board to convert Type2/3 to Type 6A PMOD standard
	(US082-INTERPEVZ)
Sensor board	Relative humidity sensor Pmod [™] board (US082-HS4001EVZ)

Table 2-1 Operating Environment for the RA6M4 Group MCU

Table 2-2 Amount of Memory Used in the RA6M4 Group MCU

Area	Size
ROM	Non-OS version: 1,785 bytes
	FreeRTOS version: 2,158 bytes
	Azure RTOS version: 2,130 bytes
RAM	Non-OS version: 101 bytes
	FreeRTOS version: 281 bytes
	Azure RTOS version: 450 bytes



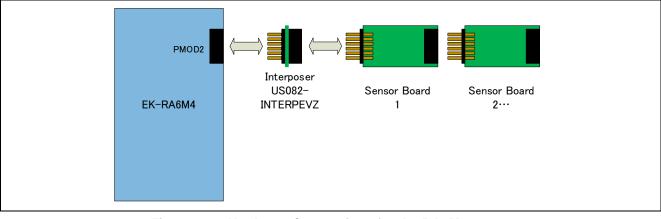


Figure 2-1 Hardware Connections for the RA6M4 Group



2.2 Environment for Confirming Operation on the RA0E1 Group MCU

The operation of this software has been confirmed on the MCU of the RA0E1 group in the following environment.

Item	Description
Demonstration board	RTK7FPA0E1S00W2BJ (RA0E1 Fast Prototyping Board)
Microcontroller	RA0E1 (R7FA0E1073CFJ:32pin)
Operating frequency	32 MHz
Operating voltage	5 V
Integrated development environment	e ² Studio 2024-01.1
C compiler	GNU ARM Embedded 13.2.1.arm-13-7
FSP	V.5.2.0
RTOS	FreeRTOS™
Emulator	On-board debugger (J-LINK)
Sensor board	Relative humidity sensor Pmod [™] board (US082-HS4001EVZ)

Table 2-3 Operating Environment for the RA0E1 Group MCU

Table 2-4 Amount of Memory Used in the RA0E1 Group MCU

Area	Size
ROM	Non-OS version: 1,280 bytes
	FreeRTOS version: 1,414 bytes
RAM	Non-OS version: 86 bytes
	FreeRTOS version: 164 bytes

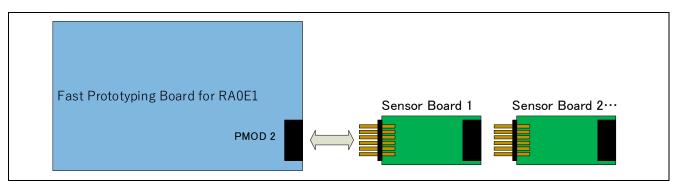


Figure 2-2 Hardware Connections for the RA0E1 Group



2.3 Environment for Confirming Operation on the RX Family MCU

The operation of this software has been confirmed on the 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 [™] / Microsoft [®] Azure RTOS
Emulator	On-board debugger (E2OB)
Interposer	Interposer board to convert Type2/3 to Type 6A PMOD standard (US082-INTERPEVZ)
Sensor board	Relative humidity sensor Pmod [™] board (US082-HS4001EVZ)

Table 2-5 Operating Environment for the RX Family MCU

Table 2-6 Amount of Memory Used in the RX Family MCU

Area	Size
ROM	Non-OS version: 2,075 bytes
	FreeRTOS version: 2,300 bytes
	Azure RTOS version: 2,360 bytes
RAM	Non-OS version: 177 bytes
	FreeRTOS version: 237 bytes
	Azure RTOS version: 446 bytes

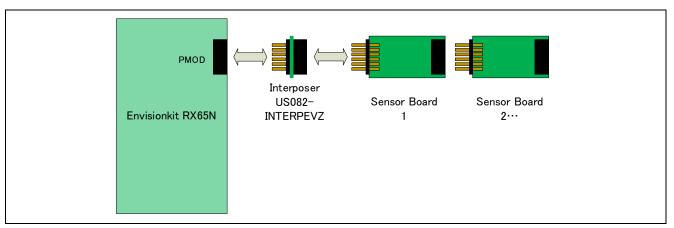


Figure 2-3 Hardware Connections for the RX Family



2.4 Environment for Confirming Operation on the RL78/G14 Group MCU

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

Table 2-7	Operating Environment for the RL78/G14 Group MCU
-----------	--

Item	Description
Demonstration board	RTK5RLG140C00000BJ (RL78/G14 Fast Prototyping Board)
Microcontroller	RL78/G14 (R5F104MLAFB: 80 pins)
Operating frequency	32 MHz
Operating voltage	3.3 V
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 debugger (E2OB)
Sensor board	Relative humidity sensor Pmod [™] board (US082-HS4001EVZ)

Table 2-8 Amount of Memory Used in the RL78/G14 Group MCU

Area	Size
ROM	1,992 bytes
RAM	118 bytes

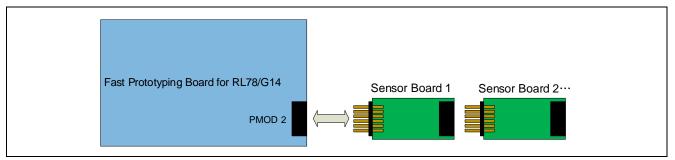


Figure 2-4 Hardware Connections for the RL78/G14 Group



2.5 Environment for Confirming Operation on the RL78/G23 Group MCU

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

Table 2-9	Operating Environment for the RL78/G23 Group MCU
-----------	--

Item	Description	
Demonstration board	RTK7RLG230CSN000BJ (RL78/G23-128p Fast Prototyping	
	Board)	
Microcontroller	RL78/G23 (R7F100GSN2DFB: 128 pins)	
Operating frequency	32 MHz	
Operating voltage	3.3 V	
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	
	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-HS4001EVZ)	

Table 2-10 Amount of Memory Used in the RL78/G23 Group MCU

Area	Size
ROM	2,361 bytes
RAM	118 bytes

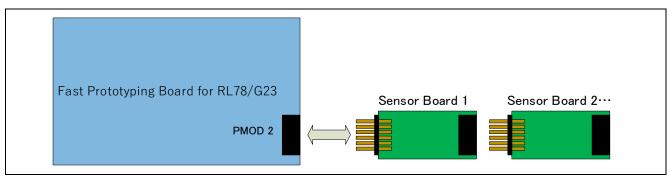


Figure 2-5 Hardware Connections for the RL78/G23 Group



2.6 Environment for Confirming Operation on the RZ Family MCU

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

Item	Description	
Demonstration board	RTK9744L23S01000BE (RZ/G2L Evaluation Kit (SMARC))	
Microcontroller	RZ/G2L (R9A07G044L23GBG :456pin)	
Operating frequency	Arm® Cortex®-M33 : 200MHz、Arm® Cortex®-A55 : 1.2GHz	
Operating voltage	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 (QCIOT-HS40		

Table 2-11 Operating Environment for the RZ Family MCU

Table 2-12	Amount of Memory Used in the RZ Family MCU
------------	--

Area	Size
ROM	Non-OS version: 2,060 bytes
	FreeRTOS version: 2,460 bytes
RAM	Non-OS version: 169 bytes
	FreeRTOS version: 449 bytes

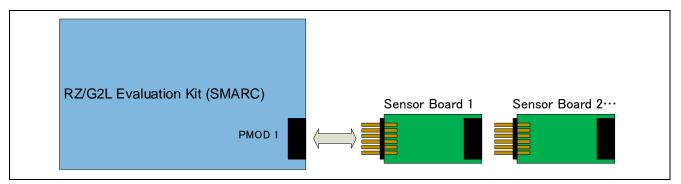


Figure 2-6 Hardware Connections for the RZ Family



3. Sensor Specifications

3.1 Overview of Sensor Specifications

Table 3-1 gives an overview of the specifications of the HS400x humidity and temperature sensor.

Item	Description	
Humidity range	0%RH to 100%RH	
Humidity accuracy	±1.5%RH (typ.) (HS4001: 10%RH to 90%RH, 25°C)	
Humidity resolution	0.04%RH (typ.)	
Hysteresis	±1.0%RH (max.)	
Humidity response time	4 seconds (typ.) (in a sealed space)	
Temperature range	–40°C to +125°C	
Temperature accuracy	±0.2°C (typ.) (HS4001 and HS4002: -10°C to 80°C)	
Temperature resolution	0.01°C (typ.)	
Temperature response time	2 seconds (typ.) (in a sealed space)	
Long-term stability	0.03°C/Yr (max.)	
Supply voltage dependency	0.03°C/V (typ.)	

Table 3-1 Overview of Sensor Specifications

3.2 Sensor Functions

This software supports the following functions of HS400x.

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.	

Table 3-2 List of Sensor Functions



3.2.1 I2C Communication Interface

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

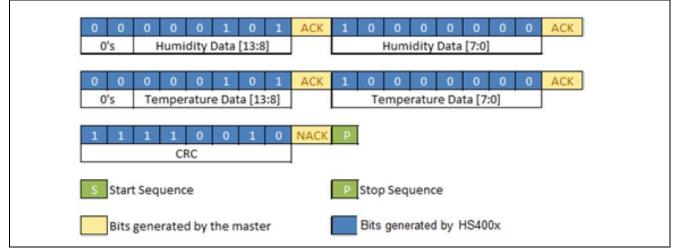


Figure 3-1 Format of I2C Communications

3.2.2 Expressions for Converting Output Values to Humidity and Temperature

The HS400x 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

$$Temperature[^{\circ}C] = \left(\frac{Temperature[13:0]}{2^{14}-1}\right) * 165-40$$



4. Specifications of Sample Software

This section describes the specifications of the sample software. For the FreeRTOS settings for the RX family, refer to the FAQ.

4.1 Configuration of the Sample Software

Figure 4-1 shows the configuration of sample software blocks.

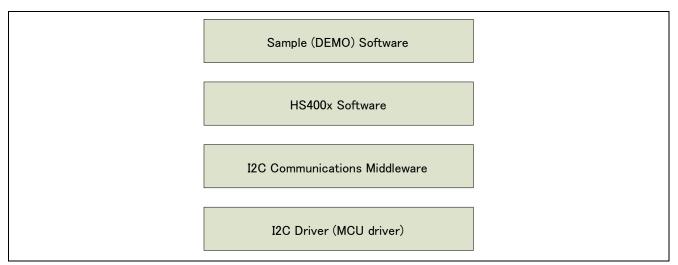


Figure 4-1 Block Diagram of the Sample Software

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 HS400x Sensor API FIT Module application note (R01AN5893) and RL78 Family Renesas Sensor Control Modules application note (R01AN5896).

Table 4-1 List of Sensor API Function	S
---------------------------------------	---

Function	Description	
RM_HS400X_Open	Starts control of the sensor.	
RM_HS400X_Close	Terminates control of the sensor.	
RM_HS400X_MeasurementStart	Starts measurement by the sensor.	
RM_HS400X_MeasurementStop	Stops measurement by the sensor.	
RM_HS400X_Read	Acquires raw data from the sensor.	
RM_HS400X_DataCalculate	Calculates values from the raw data acquired from the sensor.	



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 HS400x API functions and the expected orders of function calls.

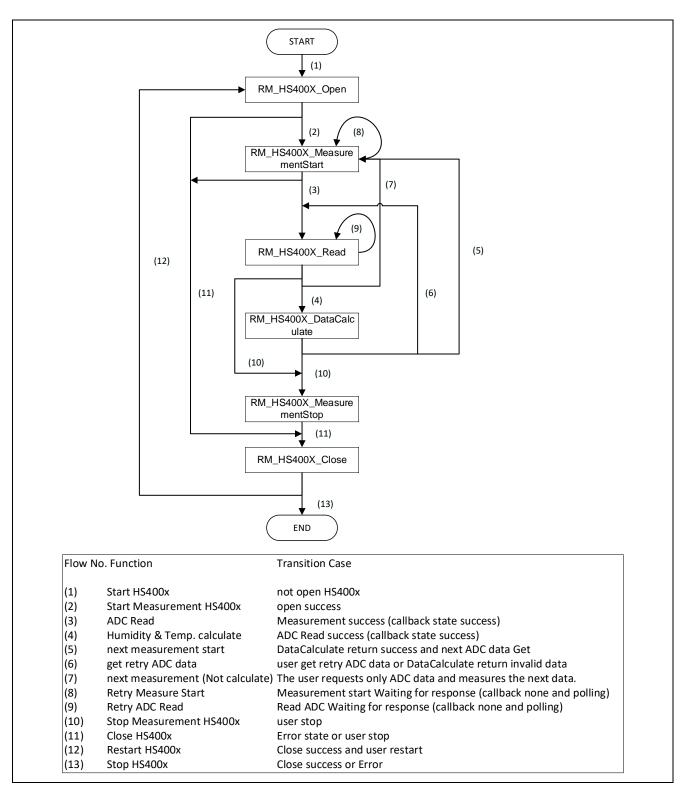


Figure 4-2 Transitions between API Function Calls



The conditions for calling the individual functions are shown below.

• RM_HS400X_Open:	(1) Starting the HS400x or(12) Restarting after a call of RM_HS400X_Close
RM_HS400X_Close:	(11) Successful completion or abnormal end of individual processing
 RM_HS400X_MeasurementStart: 	 (2) Starting measurement after a call of RM_HS400X_Open, (5) (7) reading the next measured data, or (8) retry after waiting for the response to the measurement start request
 RM_HS400X_MeasurementStop: 	(10) Stop measurement
• RM_HS400X_Read:	 (3) Reading measured data after a call of RM_HS400X_MeasurementStart or (9) retry after waiting for the response to the data acquisition request
RM_HS400X_DataCalculate:	(4) Calculating humidity and temperature data after a call of RM_HS400X_Read

Notes:

Since RM_HS400X_Open checks the state of the I2C driver, the I2C driver must be opened before the RM_HS400X_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_HS400X_MeasurementStart, the sensor stops measurement after having updated the ADC data. Therefore, be sure to call RM_HS400X_MeasurementStart at least once every time before executing processing by RM_HS400X_Read.

When using these API functions in an RTOS system, the user will need to control the bus by using a semaphore if the sensor is controlled in multiple threads or tasks at the same time. For the timing of the semaphore being raised and the control of blocking, refer to section 4.4, Flowchart of the OS Version of the Sample Software.



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

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

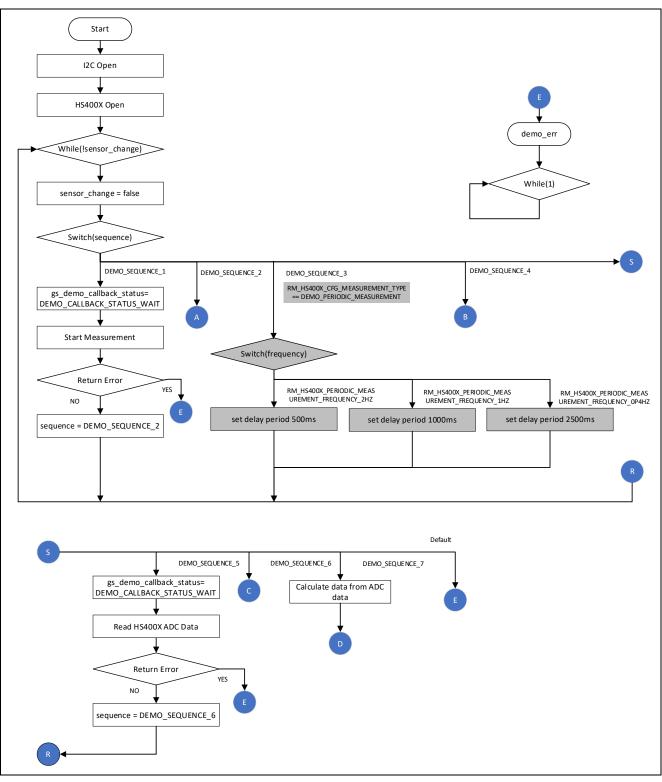


Figure 4-3 Flowchart 1 of the Main Processing in the Non-OS Version of the Sample Software



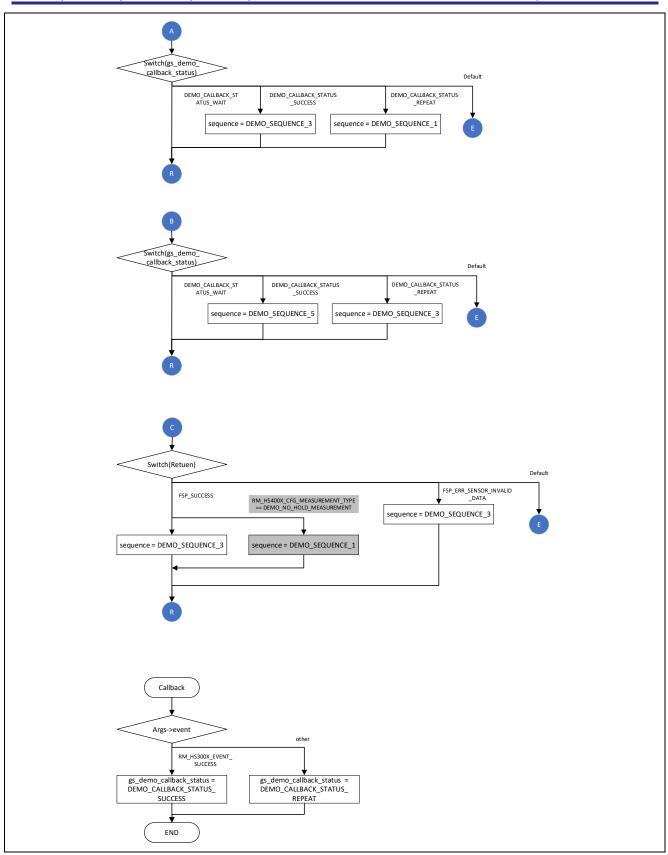


Figure 4-4 Flowchart 2 of the Main Processing in the Non-OS Version of the Sample Software

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.

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

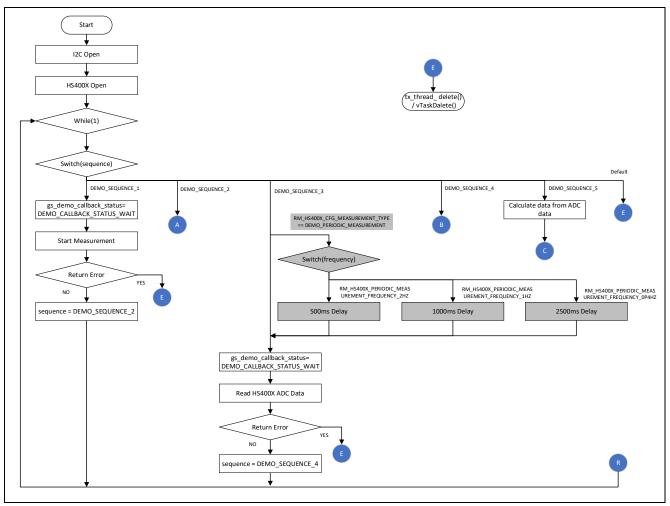


Figure 4-5 Flowchart 1 of the Main Processing in the OS Version of the Sample Software



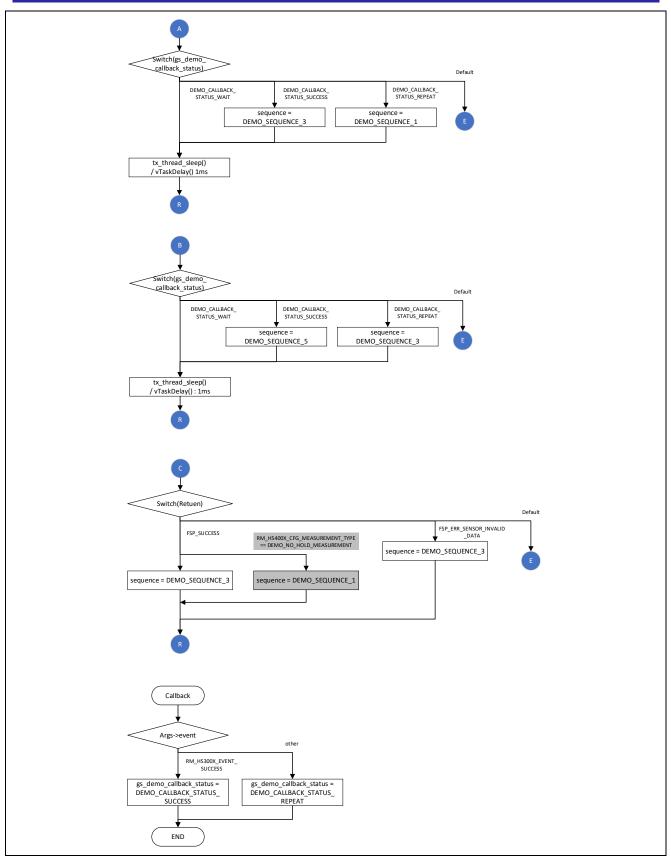


Figure 4-6 Flowchart 2 of the Main Processing in the OS Version of the Sample Software

4.5 Azure RTOS Project

The RX project for use with the Azure RTOS has the following changes from the default source files generated by the RX Smart Configurator.

1. src/demo_thread.c

Line 57: Addition of extern void tx_application_define_user (void);

Line 178: Addition of tx_application_define_user();

2. libs/threadx/common/inc/tx_api.h

Line 224: Change of TX_TIMER_TICKS_PER_SECOND ((ULONG) 1000)



5. Configuration Settings

5.1 HS400x Humidity and Temperature Sensor Settings

5.1.1 RA Family

Select the rm_hs400x 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 HS400x Settings for the RA Family MCU

Configurable Item	Value	Description	
Common			
Parameter Checking	Default (BSP)	Specify whether to include the processing to	
	Enabled	check parameters in the code to be generated.	
	Disabled	When "Disabled" is selected, the generated code does not include the processing to check parameters. When "Enabled" is selected, the generated code includes the processing to check parameters.	
Measurement Type	Hold Measurement	Specify the type of measurement.	
	No-Hold Measurement	Select from among "Hold Measurement" (one-	
	Periodic Measurement	shot measurement), "No-Hold Measurement", or "Periodic Measurement" (sequential measurement).	
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.	
Module g_hs400	x_sensor HS400X on rr	· · · · · · · · · · · · · · · · · · ·	
Name	g_hs400x_sensor0	Specify the name of the module. A module name conforming to the C language standard can be specified.	
Temperature	8-bit	Specify the resolution of temperature	
Resolution	10-bit	measurement.	
	12-bit		
	14-bit		
Humidity Resolution	8-bit	Specify the resolution of humidity measurement.	
	10-bit		
	12-bit		
	14-bit		
Frequency for	0.4Hz	Specify the frequency of periodic measurement.	
Periodic	1Hz	Measurements proceed at the specified	
Measurement	2Hz	frequency.	
Callback	hs400x_callback	Specify the name of the user callback function. A callback function name conforming to the C language standard can be specified. When "NULL" is specified, no callback function is used.	



5.1.2 RX Family

Select the r_hs400x_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.

設定項目		│ 説明
Configurations		
Parameter Checking	System Default Enabled Disabled	Enable or disable the parameter check processing. When "Enabled" is selected, the project is built so that the generated code includes the parameter check processing.
Number of HS400X sensors	1 2	Specify the number of HS400X sensors to be connected.
Measurement Type of HS400X sensors	Hold Measurement No-Hold Measurement Periodic Measurement	Specify the type of measurement. Select from among "Hold Measurement" (one- shot measurement), "No-Hold Measurement", or "Periodic Measurement" (sequential measurement).
Data Types of HS400X sensors	Both humidity and temperature Temperature only	Specify the type of data to be acquired from the sensor. Both the humidity and temperature or the humidity alone can be selected.
I2C Communication device No. for HS400X sensor device x ($x = 0$ or 1)	I2C Communication Device(y) (y = 0 - 15)	Specify the communication line to be used by the sensor.
Temperature Resolution for HS400X sensor device x (x = 0 or 1)	8-bit 10-bit 12-bit 14-bit	Specify the resolution of temperature measurement.
Humidity Resolution for HS400X sensor device x ($x = 0$ or 1)	8-bit 10-bit 12-bit 14-bit	Specify the resolution of humidity measurement.
Frequency for Periodic Measurement for HS400X sensor device x (x = 0 or 1)	0.4Hz 1Hz 2Hz	Specify the frequency of periodic measurement. Measurements proceed at the specified frequency.
Callback function for HS400X sensor device x ($x = 0$ or 1)	hs400x_user_callback(x) (x = 0 or 1)	Specify the name of the user callback function. A callback function name conforming to the C language standard can be specified.

Table 5-2 HS400x Settings for the RX Family MCU



5.1.3 RL78 Family

Settings can be modified by changing the values of constants defined in the \r_config\r_hs400x_rl_config.h file in the project tree of the sample project.

The following items and values can be specified.

Table 5-3 HS400x Settings for the RL78 Family MCU

Constant Name	Value	Description
Configurations		
RM_HS400X_CFG_PARAM _CHECKING_ENABLE	0	Specify whether to include the processing to check parameters in the code to be generated.
	1	When "0" is specified, the generated code does not include the processing to check parameters. When "1" is specified, the generated code includes the processing to check parameters.
RM_HS400X_CFG_DEVIC E_NUM_MAX	1 2	Specify the number of HS400x sensors to be connected.
RM_HS400X_CFG_MEASU	0	Specify the type of measurement.
REMENT_TYPE	1	Select from among "Hold Measurement"
	2	 (0) (one-shot measurement), "No-Hold Measurement" (1), or "Periodic Measurement" (2) (sequential measurement).
RM_HS400X_CFG_DATA_	0	Specify the type of data to be acquired
BOTH_HUMIDITY_TEMPE RATURE	1	from the sensor. The humidity alone (0) or both the humidity and temperature (1) can be selected.
RM_HS400X_CFG_DEVIC Ex_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_HS400X_CFG_DEVIC Ex_CALLBACK (x = 0 or 1)	hs400x_user_callback(x) (x = 0 or 1)	Specify the name of the user callback function. A callback function name conforming to the C language standard can be specified.
RM_HS400X_CFG_DEVIC Ex_TEMPERATURE_RESO LUTION (x = 0 or 1)	RM_HS400X_TEMPERATUR E_RESOLUTION_8BIT RM_HS400X_TEMPERATUR E_RESOLUTION_10BIT RM_HS400X_TEMPERATUR E_RESOLUTION_12BIT RM_HS400X_TEMPERATUR E_RESOLUTION_14BIT	Specify the resolution of humidity measurement.



RM_HS400X_CFG_DEVIC Ex_HUMIDITY_RESOLUTI ON (x = 0 or 1)	RM_HS400X_HUMIDITY_RE SOLUTION_8BIT	Specify the resolution of temperature measurement.	
	RM_HS400X_HUMIDITY_RE SOLUTION_10BIT		
	RM_HS400X_HUMIDITY_RE SOLUTION_12BIT		
	RM_HS400X_HUMIDITY_RE SOLUTION_14BIT		
RM_HS400X_CFG_DEVIC Ex_PERIODIC_MEASURE MENT_FREQUENCY (x = 0 or 1)	RM_HS400X_PERIODIC_MEAS UREMENT_FREQUENCY_0P4H Z	Specify the frequency of periodic measurement. Measurements proceed at the specified	
	RM_HS400X_PERIODIC_MEAS UREMENT_FREQUENCY_1HZ	frequency.	
	RM_HS400X_PERIODIC_MEAS UREMENT_FREQUENCY_2HZ		



5.1.4 RZ Family

Select the rm_hs400x 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	HS400x Settings for the RZ Family MCU

Configurable Item	Value	Description
Common		
Parameter Checking	Default (BSP) Enabled	Specify whether to include the processing to check parameters in the code to be generated. When "Disabled" is selected, the generated
	Disabled	code does not include the processing to check parameters. When "Enabled" is selected, the generated code includes the processing to check parameters.
Measurement Type	Hold Measurement	Specify the type of measurement.
	No-Hold Measurement	Select from among "Hold Measurement" (one-
	Periodic Measurement	shot measurement), "No-Hold Measurement", or "Periodic Measurement" (sequential measurement).
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.
Module g_hs400)x_sensor HS400X on rr	-
Name	g_hs400x_sensor0	Specify the name of the module. A module name conforming to the C language standard can be specified.
Temperature	8-bit	Specify the resolution of temperature
Resolution	10-bit	measurement.
	12-bit	
	14-bit	
Humidity Resolution	8-bit	Specify the resolution of humidity measurement.
	10-bit	
	12-bit	
	14-bit	
Frequency for	0.4Hz	Specify the frequency of periodic measurement.
Periodic	1Hz	Measurements proceed at the specified
Measurement	2Hz	frequency.
Callback	hs400x_callback	Specify the name of the user callback function. A callback function name conforming to the C language standard can be specified. When "NULL" is specified, no callback function is used.



5.2 Sensor Communication 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 Middleware Settings for the RA Family MCU

Configurable Item	Value	Description
Common		
Parameter Checking	Default (BSP) Enabled Disabled	Specify whether to include the processing to check parameters in the code to be generated. When "Disabled" is selected, the generated code does not include the processing to check parameters. When "Enabled" is selected, the generated code includes the processing to check parameters.
Module g_comm rm_comms_i2c	ns_i2c_device I2C Comn	nunication Device on
Name	g_comms_i2c_device0	Specify the name of the module. A module name conforming to the C language standard can be specified.
Semaphore Timeout	0xFFFFFFF	For an RTOS project, specify the time of semaphore timeout.
Slave Address	0x54	Specify the slave address. When rm_hs400x is used, this value is automatically specified and cannot be modified.
Address Mode	7-Bit	Specify the number of slave address bits. When rm_hs400x is used, this value is automatically specified and cannot be modified.
Callback	rm_hs400x_callback	Specify the name of the user callback function. When rm_hs400x 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 Bus	Unuse Use	For an RTOS project, enable or disable recursive operation when blocking is enabled.



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.

Configurable Item	Value	Description
Configurations		
Parameter Checking	System Default Enabled Disabled	Specify whether to include the processing to check parameters in the code to be generated. When "Disabled" is selected, the generated code does not include the processing to check parameters. When "Enabled" is selected, the generated code includes the processing to check parameters.
Number of I2C Shared Buses	Unused 1 2 – 16	Specify the number of I2C bus lines that can be connected.
Number of I2C Devices	Unused 1 2 - 16	Specify the number of I2C devices that can be connected.
Blocking operation supporting with RTOS	Disabled Enabled	For an RTOS project, enable or disable the blocking operation.
Bus lock operation supporting with RTOS	Disabled Enabled	For an RTOS project, enable or disable the bus lock operation.
IIC Driver Type for I2C Shared bus(x) (x = 0 - 15)	RIIC SCI IIC Not selected	Specify the I2C type to be used for the communications bus. Using the RIIC requires r_riic_rx. Using the SCI IIC requires r_sci_iic_rx. If an unused FIT module is deleted, a warning message will appear but this will not affect the operation.
Channel No. for I2C Shared bus(x) (x = 0 - 15)	0	Specify the I2C channel number to be used for the communications bus.
Timeout for the bus lock of the I2C bus for I2C Shared Bus(x) (x = 0 - 15)	0xFFFFFFF	Specify the time of I2C bus lock timeout for I2C bus x. ($x = 0 - 15$)
I2C Shared Bus No. for I2C Communication Device(x) (x = 0 - 15)	I2C Shared Bus(x) (x = 0 – 15)	Specify the configuration of the I2C bus to be used for the communications bus.
Slave address for communication device(x) (x = 0 - 15)	0x54	Specify the slave address of the device to be connected to the communications bus. When using r_hs400x_rx, specify 0x54.
Slave address mode for communication device(x) (x = 0 - 15)	7 bit address mode	Specify the slave address mode. When using r_ hs400x_rx, specify the 7-bit address mode.
Callback function for Communication device(x) (x = 0 - 15)	comms_i2c_user_callback(x) (x = 0 - 15)	Specify the name of the user callback function. When using r_ hs400x_rx, specify rm_ hs400x_callback(y) ($y = 0$).

Table 5-6	Communication	Middleware	Settinas f	or the RX	Family MCU
	••••••••••••••••••••••••••••••••••••••	maaromaro	eetinge i	•••••••••	



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.

Constant Name	Value	Description	
Configurations			
COMMS_I2C_CFG_PARA M_CHECKING_ENABLE	0	Specify whether to include the processing to check parameters in the code to be generated.	
	1	When "0" is specified, the generated code does not include the processing to check parameters. When "1" is specified, the generated code includes the processing to check parameters.	
COMMS_I2C_CFG_BUS_N	1	Specify the number of communications	
UM_MAX	2	bus lines that can be connected.	
	3		
	4		
	5		
COMMS_I2C_CFG_DEVIC	1	Specify the number of I2C devices that	
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 the	
$DRIVER_TYPE$ $(x = 0 - 4)$	COMMS_DRIVER_SAU_I2C	communications bus.	
$COMMS_I2C_CFG_DEVIC$ E(x)_BUS_CH (x = 0 - 4)	g_comms_i2c_bus(x)_extended _cfg (x = 0 - 4)	Specify the configuration of the I2C bus to be used for the communications bus.	
$\begin{array}{l} COMMS_I2C_CFG_DEVIC\\ E(x)_SLAVE_ADDR\\ (x = 0 - 4) \end{array}$	0x54	Specify the slave address of the device to be connected to the communications bus. When using rm_hs400x, specify 0x54.	
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_hs400x, specify rm_hs400x_callback(y) (y = 0 or 1).	

Table 5-7 Communication Middleware Settings for the RL78 Family MCU



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 Middleware Settings for the RZ Family MCU

Configurable Item	Value	Description
Common		
Parameter Checking	Default (BSP) Enabled Disabled	Specify whether to include the processing to check parameters in the code to be generated. When "Disabled" is selected, the generated code does not include the processing to check parameters. When "Enabled" is selected, the generated code includes the processing to check parameters.
Module g_comm rm_comms_i2c	ns_i2c_device I2C Comr	nunication Device on
Name	g_comms_i2c_device0	Specify the name of the module. A module name conforming to the C language standard can be specified.
Semaphore Timeout	0xFFFFFFF	For an RTOS project, specify the time of semaphore timeout.
Slave Address	0x54	Specify the slave address. When rm_hs400x is used, this value is automatically specified and cannot be modified.
Address Mode	7-Bit	Specify the number of slave address bits. When rm_hs400x is used, this value is automatically specified and cannot be modified.
Callback	rm_hs400x_callback	Specify the name of the user callback function. When rm_hs400x 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	For an RTOS project, enable or disable processing for blocking.
	Use	
Recursive Mutex for Bus	Unuse Use	For an RTOS project, enable or disable recursive operation when blocking is enabled.



5.3 I2C Driver Settings

5.3.1 RA Family

Select the r_iic_master, r_sci_i2c or r_sau_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	Common				
Parameter Checking	Default (BSP)	Specify whether to include the processing to			
	Enabled	check parameters in the code to be generated. When "Disabled" is selected, the generated code			
	Disabled	does not include the processing to check parameters. When "Enabled" is selected, the generated code includes the processing to check parameters.			
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. When using rm_hs400x, select "Disabled".			
Module g_i2c_m	naster0 I2C Master (r_ii				
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	When using rm_hs400x, select "Standard" or "Fast-mode".			
	Fast-mode plus				
Rise Time (ns)	120	Specify the time for the SCL signal to rise according to the specifications of the target board to be used.			
Fall Time (ns)	120	Specify the time for the SCL signal to fall according to the specifications of the target board to be used.			
Duty Cycle (%)	50	Specify the SCL duty cycle.			
Slave Address	0x00	This item specifies the slave address of the device to be connected but the user does not need to make this setting because rm_comms_i2c overwrites any setting made here.			
Address Mode	7-Bit	This item specifies the salve address mode for the			
	10-Bit	device to be connected but the user does not need to make this setting because rm_comms_i2c overwrites any setting made here.			
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 6 Priority 7 Priority 8 Priority 8 Priority 8 Priority 9 Priority 10 Priority 11 Priority 12 Priority 13 Priority 15	Specify the interrupt priority level of the I2C bus driver.
Pins		<u> </u>
SDA	Рххх	The pin numbers to be used by the driver are
SCL	Рххх	displayed. Use the [Pins] tabbed page to modify the pin configuration.



Table 5-10 r_sci_i2c Settings for the RA Family MCU

Configurable Item	Value	Description
Common	1	
Parameter Checking	Default (BSP)	Specify whether to include the processing to
	Enabled	check parameters in the code to be generated. When "Disabled" is selected, the generated
	Disabled	code does not include the processing to check parameters. When "Enabled" is selected, the generated code includes the processing to check parameters.
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. When using rm_hs400x, select "Disabled".
Module g_i2c0 I	2C Master (r_sci_i2c)	
Name	g_i2c0	Specify the name of the module.
Channel	0	Specify the channel number to be used.
Slave Address	0x00	This item specifies the slave address of the device to be connected but the user does not need to make this setting because rm_comms_i2c overwrites any setting made here.
Address Mode	7-Bit	This item specifies the salve address mode for
	10-bit	the device to be connected but the user does not need to make this setting because rm_comms_i2c overwrites any setting made here.
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 noise filter to be used 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 function.
	Disable	
Callback	rm_comms_i2c_callback	The name of the user callback function is automatically specified by rm_comms_i2c.



Interrupt Priority Level	Priority 0 (highest)	Specify the interrupt priority level of the I2C bus
	Priority 1	driver.
	Priority 2	
	Priority 3	
	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 using the DTC, specify the priority level of
Level [Only used	Priority 1	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	Pxxx	The pin numbers to be used by the driver are
SCL	Pxxx	displayed.
		Use the [Pins] tabbed page to modify the pin configuration.



Table 5-11 r_sau_i2c Settings for the RA Family MCU

Configurable Item	Value	Description
Common	<u>I</u>	
Parameter Checking	Default (BSP)	Specify whether to include the processing to
	Enabled	check parameters in the code to be generated. When "Disabled" is selected, the generated code does not include the processing to check parameters. When "Enabled" is selected, the generated code includes the processing to check parameters.
	Disabled	
Enable Critical Section	Enabled	Set enable or disable of critical section.
	Disabled	When using multiple channels on the same SAU unit, specify Enabled.
Manual Start-Stop	Enabled	Specify whether the user calls the start condition and stop condition function.
	Disabled	Condition and stop condition function. When "Disabled" is selected, the generated code does not include calls to the start condition and stop condition function. When "Enabled" is selected, the generated code includes calls to the start condition and stop condition function.
Enable Single Channel	00	Does not include processing other than the
	20	specified channel. When "Disabled" is selected, all channels are
	Disabled	supported.
DTC Support	Enabled	Specify whether to support the DTC.
	Disabled	
Module g_i2c0 l2	C Master (r_sau_i2c)	
Name	g_i2c0	Specify the name of the module.
Channel	20	Specify the channel number to be used.
Operation clock	СКО	Specify the I2C operation clock.
	CK1	-
Slave Address	device to be connected but the use need to make this setting because rm_comms_i2c overwrites any sett	device to be connected but the user does not
Rate	Standard	Specify the bit rate.
	Fast-mode	Select "Standard" or "Fast-mode".
	Fast-mode plus	
Delay time (Microseconds)	5	Specify the SDA output delay time.
Callback	rm_comms_i2c_callback	The name of the user callback function is automatically specified by rm_comms_i2c.
Transfer end interrupt	Priority 0 (highest)	Specify the interrupt priority level of the I2C bus driver.
priority	Priority 1	
	Priority 2	
	Priority 3	



Pins		
SCL	Рххх	The pin numbers to be used by the driver are displayed.Use the [Pins] tabbed page to modify the pin configuration.
SDA	Рххх	



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 5-12 r_riic_rx Settings for the RX Family MCU

Configurable Item	Value	Description
Configurations	'	
Set parameter checking enable	System Default Not Include	Specify whether to include the processing to check parameters in the code to be generated. When "Not" is selected, the generated code does not include the processing to check parameters. When "Include" is selected, the generated code includes the processing to check parameters.
MCU supported channels for CHx (x = 0 - 2)	Not supported Supported	Specify whether to use channel x. When a channel is not to be used, select "Not supported". When "Not supported" is selected, the generated code does not include processing for the given channel. When "Supported" is selected, the generated code includes processing for the given channel.
CHx RIIC bps(kbps) $(x = 0 - 2)$	400	Specify the bit rate. Set this to a value no greater than 400 when using rm_hs400x.
Digital filter for CHx $(x = 0 - 2)$	Not One IIC phi Two IIC phi Three IIC phi Four IIC phi	Specify the number of stages in the noise filter for the specified channel. When "Not" is selected, the noise filter is disabled.
Setting port setting processing	Not include port setting Include port setting	Specify whether to include the settings for using port pins as the SCL and SDA pins in the code to be generated. When "Not include port setting" is selected, the generated code does not include the processing for setting port pins to serve as serial pins. When "Include port setting" is selected, the generated code includes the processing for setting port pins to serve as serial pins.
Master arbitration lost detection function for CHx (x = 0 - 2)	Unused Used	Enable or disable the master loss-in-arbitration detection function for the specified channel. When using multiple masters, select "Used" (enabled). When "Unused" is selected, master loss-in- arbitration detection is disabled. When "Used" is selected, master loss-in-arbitration detection is enabled.
Address y format for CHx (x = 0 - 2, y = 0 - 2)	Not 7 bit address format 10 bit address format	Specify whether to support 7-bit addressing or 10- bit addressing for the slave address of the specified RIIC. When using rm_hs400x, select "7 bit address format".



Slave Address y for CHx (x = 0 - 2, y = 0 - 2)	0x0025	This item specifies the slave address for the specified channel but the user does not need to make this setting because rm_comms_i2c overwrites any setting made here.
General call address	Unused	Enable or disable the use of the general call
for CHx	Used	address with the specified channel. When "Unused" is selected, the use of the general call address is disabled. When "Used" is selected, the use of the general call address is enabled.
CHx RXI INT Priority	Level 1	Specify the priority level of the reception data full
Level	Level 2	interrupt (RXI) for the specified channel.
(x = 0 - 2)	Level 3	Specify a level from 1 to 15.
	Level 4	
	Level 5	
	Level 6	
	Level 7	
	Level 8	1
	Level 9	
	Level 10	1
	Level 11	
	Level 12	
	Level 13	
	Level 14	
	Level 15 (highest)	
CHx RXI INT Priority	Level 1	Specify the priority level of the transmission data
Level	Level 2	empty interrupt (TXI) for the specified channel.
(x = 0 - 2)	Level 3	Specify a level from 1 to 15.
	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 communication error
Level	Level 2	or event generation interrupt (EEI) for the specified
(x = 0 - 2)	Level 3	channel.
	Level 4	Specify a level from 1 to 15.
	Level 5	
	Level 6	
	Level 7	
	Level 8	
	Level 9	
	Level 10	
	Level 11	



	Level 12	
	Level 13	
	Level 14	
	Level 15 (highest)	
CHx TEI INT Priority	Level 1	Specify the priority level of the transmission end
Level	Level 2	interrupt (TEI) for the specified channel.
(x = 0 - 2)	Level 3	Specify a level from 1 to 15.
, ,	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	Enable or disable the timeout detection function for
CHx	Used	the specified channel.
(x = 0 - 2)		When "Unused" is selected, timeout detection is
		disabled. When "Used" is selected, timeout detection is
		enabled.
Timeout detection time	Long mode	Specify the time for timeout detection for the
for CHx	Short mode	specified channel.
(x = 0 - 2)	Short mode	
Count up during low	Unused	Enable or disable incrementing of the internal
period of timeout	Used	counter for detecting a timeout while the SCL signal
detection for CHx		is at the low level when the timeout detection
(x = 0 - 2)		function is enabled for the specified channel.
		When "Unused" is selected, incrementing of the
		counter while the SCL signal is at the low level is
		disabled.
		When "Used" is selected, incrementing of the
		counter while the SCL signal is at the low level is enabled.
Count up during high	Unused	Enable or disable incrementing of the internal
period of timeout	Used	counter for detecting a timeout while the SCL signal
detection for CHx		is at the high level when the timeout detection
(x = 0 - 2)		function is enabled for the specified channel.
		When "Unused" is selected, incrementing of the
		counter while the SCL signal is at the high level is
		disabled.
		When "Used" is selected, incrementing of the
		counter while the SCL signal is at the high level is enabled.
Set Counter of	1000	The value of the timeout counter (the counter for
checking bus busy		bus checking) to be used in the processing of
		checking the bus busy state by API functions can
		be specified by software.
Resources		
100001000		



SDAx Pins	Checked	Specify the pins to be used. Select the checkboxes for the desired pins.
SCLx Pins	Checked	



Configurable Item	Value	Description
Configurations	1	· · · · · · · · · · · · · · · · · · ·
Set parameter checking enable	System Default Not Include	Specify whether to include the processing to check parameters in the code to be generated. When "Not" is selected, the generated code does not include the processing to check parameters. When "Include" is selected, the generated code includes the processing to check parameters.
MCU supported channels for CHx (x = 0 - 12)	Not supported Supported	Specify whether to use channel x. When "Not supported" is selected, the generated code does not include processing for the given channel. When "Supported" is selected, the generated code includes processing for the given channel.
SCI IIC bitrate (bps) for CHx (x = 0 - 12)	384000	Specify the bit rate. Set to 384000 (384 Kbits/s) or a smaller value.
Interrupt Priority for CHx (x = 0 - 12) Digital noise filter (NFEN bit) for CHx (x = 0 - 12)	Level 1 Level 2 Level 3 Level 4 Level 5 Level 6 Level 7 Level 8 Level 9 Level 10 Level 10 Level 11 Level 12 Level 13 Level 13 Level 14 Level 15 (highest) Disable Enable	Specify the priority level of interrupts triggered by the detection of a start or stop condition, reception, transmit data empty, and transmit end. Specify a level from 1 to 15. Specify whether to use the noise cancellation function for the SSCL and SSDA input signals. When "Disable" is selected, the noise cancellation function is disabled. When "Enable" is selected, the noise cancellation function is enabled.
Noise Filter Setting Register (NFCS bit) for CHx (x = 0 - 12)	The clock divided by 1 The clock divided by 2 The clock divided by 4 The clock divided by 8	Specify the sampling clock of the digital noise filter.
I2C Mode Register 1 (IICDL bit) for CHx (x = 0 - 12)	18	Specify the number of SSDA output delay cycles from the falling edge of the SSCL output. Specify a value from 1 to 31.
Software bus busy check counter	1000	Specify the counter value to be judged to represent the bus busy state. The value of the timeout counter (the counter for bus checking) to be used in the processing of checking the bus busy state by API functions for the simplified I2C can be specified by software.

Table 5-13 r_sci_iic_rx Settings for the RX Family MCU



Setting port setting	Not include port setting	Specify whether to include the settings for using
processing	Include port setting	port pins as the SSCL and SSDA pins in the code to be generated. When "Not include port setting" is selected, the generated code does not include the processing for setting port pins to serve as serial pins. When "Include port setting" is selected, the generated code includes the processing for setting port pins to serve as serial pins.
Resources		
SSDAx Pins	Checked	Specify the pins to be used.
SSCLx Pins	Checked	Select the checkboxes for the desired pins.



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	When using r_hs400x, select IICxx.
	IICxx	
IICxx		
Transfer rate	100000	Specify the bit rate.
		When using rm_hs400x, specify 100000.
Transfer end interrupt	High	Specify the priority level of the transfer end
priority (INTIICxx)	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	1	
Transfer mode		
Transfer mode	Unused	Specify the communication function of the
	Single master	channel to be used.
	Slave	Select "Single master".
Setting		
Clock mode setting	fCLK	Specify the clock to drive counting.
5	fCLK/2	
Address	16	Specify the local address.
Operation mode	Standard	Specify the operating mode.
setting	Fast mode/Fast mode plus	
Transfer clock (fSCL)	100000	Specify the bit rate. Set to 400000 or a smaller value.
Communication end	High	Specify the priority level of the communication
interrupt priority	Level1	end interrupt.
(INTIICAx)	Level2	
	Low	
Master transmission end	Checked	Specify whether to use the callback function when master transmission ends.
Master reception end	Checked	Specify whether to use the callback function when master reception ends.
Master error	Checked	Specify whether to use the callback function when a communication error occurs.

Table 5-14 Serial Settings for the RL78 Family MCU



G	enerated stop	Checked	Specify whether to generate a stop condition in
cc	ondition in master		the callback function.
tra	ansmission/reception		Deselect the checkbox.
er	nd callback function		



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 5-15	r iic maste	er Settings for the	RZ Family MCU
	1_110_11140tt		

Configurable Item	Value	Description		
Common	, 			
Parameter Checking	Default (BSP)	Enable or disable the parameter check		
	Enabled	processing. When "Enabled" is selected, the project is built		
	Disabled	so that the generated code includes the parameter check processing.		
DTC on	Enabled	Specify whether to use the DTC 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_hs400x, 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_hs400x, select "Standard" "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 cycle.		
	4	Removes noise below the 4 IIC cycle.		
Slave Address	0x00	This item specifies the slave address of the device to be connected but the user does not need to make this setting because 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 an example of modifying the "HS400x_RA6M4_NonOS" sample project so that it can be used on the EK-RA2E1 board.

However, when using the I2C driver different from the original sample project, describes for the FPB-RA0E1 board.

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.

S Import – 🗆	×
Select Rename and Import and Existing C/C++ Project into the workspace	5
Select an import wizard:	
type filter text	
 ✓ E General Archive File CMSIS Pack 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 Renease CRX 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 	< >
(?) < <u>Back</u> <u>Next</u> > Einish 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				\times
\leftarrow \rightarrow \checkmark \uparrow \square \diamond e2_studio \diamond	workspace	√ Ö	Search workspace	Q
Organize 🔻 New folder				- 0
e2_studio	^ Name		Date modified	Type ^
📙 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_FreeRTOS	HS300x_RX65N_NonOS		5/25/2021 3:03 PM	File folc
				~
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	Project				5
Select a directory to	search for existing Ecli	pse projects.			
Project name: HS3	00x_RA2E1_NonOS				
Use <u>d</u> efault loca	ation				
Location:	C:¥Users¥a5090534¥e	2_studio¥work	space¥HS300x_F	B <u>r</u> owse	
	Create Directory fo	r Project			
Choose file system:	default \sim				
Import from:					
• Select roo <u>t</u> direc	tory: C:¥Users¥xxxx	xxxx¥e2_studio	¥workspace 🗸	Browse	
O Select <u>a</u> rchive fi	e:		~	B <u>r</u> owse	
<u>P</u> rojects:					
HS300x_RA6M4	NonOS (C:¥Users¥xxx	«xxxx¥e2_stud	io¥workspace¥HS3	00x_RA6M4	_Nor
<					>
Options	iguration output folde	rs			
	,				
?	< <u>B</u> ack	Next >	<u>F</u> inish	Cance	I.



6.1.2 Modifying Settings of the FSP Configurator

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

6.1.2.1 [BSP] Setting

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.

		Restore Defau
levice Selecti		
evice Selecti	ion	
FSP version:	3.0.0-rc1+20210426.9fd4d31a V	Board Details Evaluation kit for RA6M4 MCU Group
Board:	EK-RA6M4 🗸 🔛	Visit https://www.renesas.com/ra/ek-ra6m4 to get kit user's manual, quick start guide,
Device:	Custom User Board (Any Device) EEK-RA2L1	errata, design package, example projects, etc.
RTOS:	EK-RA2A1 EK-RA2E1	
	EK-RA2L1	
	EK-RA4M1	
	EK-RA4M2	
	EK-RA4M3	
	EK-RA4W1	
	EK-RA6M1	
	EK-RA6M2 EK-RA6M3	
	EK-RA6M3G	
	EK-RA6M4	
	EK-RA6M5	
	FPB-RA2E1	
	RA2A1 TBB	
	RA4M1 TBB	
	RA4W1 ADK	
	RA6M1 TBB	
	RA6M2 TBB	
	RA6M3 PK	
	RSSK-RA2L1	
	RSSK-RA6T1	

6.1.2.2 [Clocks] Setting

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_RA2E1_NonOS] FSP Configuration 🛛			
locks Configuration			Generate Project Co
			Restore De
TAL 20MHz		ICLK Div /1	✓ → ICLK 48MHz
OCO 32768Hz		PCLKB Div /2	✓ → PCLKB 24MHz
IOCO 8MHz	Clock Src: HOCO	V PCLKD Div /1	✓ → PCLKD 48MHz
UBCLK 32768Hz	_		
IOCO 48MHz			
	CLKOUT Disabled	✓ → CLKOUT Div /1	✓ → CLKOUT 0Hz
	CERCOT Disabled	CEROOT DIV/T	CROOT ON 2



6.1.2.3 [Pins] Setting

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 Configuration 20				- 0
Pin Configuration				Generate Project Content
Select Pin Configuration	Exp	ort to CSV file 🛛 Configure i	Pin Driver Warnings	
AISM4 EX V Manage confi	gurations	Generate data: g_bsp_pin_	cfg	
Pin Selection 🗉 🗎 🖓	Pin Configuration			Cycle Pin Group
The first hit ✓ Φ. Prit > Φ. AnalogADLO > Φ. NoncodC. • Φ. SystemOBLO > Timera/GT > Timera/GT > Timera/GT > Timera/GT > Timera/GT > Timera/GT > Timera/GT		Wile		
Summary BSP Clocks @ Pins Interrupts Event Links @ Stack	ks Components			

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

Manage Pin Configurations		×
Multiple Pin Configuration Management		
Modify pin configuration list or import/export external file		
RA6M4 EK (Current) R7FA6M4AF3CFB.pincfg	Add.	
RA2E1 EK	Remo	ve
R7FA2E1A92DFM.pincfg	Renam	e
	Duplica	ate
	Merge t	io
	Import	t
	Export	
	OK	



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

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

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

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

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

in Configuration						O Generate Project Conten
Select Pin Configuration			📑 Expor	t to CSV file 🛛 🗄	Configure Pir	n Driver Warnings
RA2E1 EK		✓ Manage configurations		Generate data:	g_bsp_pin_cfg	g_2e1
Pin Selection 📔 🕀 🛛	⊒ ↓ <mark>a</mark>	Pin Configuration				😲 Cycle Pin Group
Type filter text		Name	Value	Lock	Link	
	_	Pin Group Selection	Mixed			
V V Peripherals	^	Operation Mode	Simple I2C			
> Analog:ACMP		✓ Input/Output				
 Analog:ADC Analog:ANALOG Connectivity:IIC Connectivity:SCI 		TXD1	None		\Rightarrow	
		RXD1	None		\Rightarrow	
		SCK1	None		\Rightarrow	
✓ ✓ Connectivity.sci ✓ SCI0		CTS1	None		\Rightarrow	
✓ SCI1		SDA1	✓ P401		\Rightarrow	
✓ SCI2		SCL1	✓ P402	E E	\Rightarrow	
SCI9						
> 🗸 Connectivity:SPI						
> Input:CTSU						
> Input:ICU						
> Input:KINT						
> Monitoring:CAC		<				2
> System:CGC		Module name: SCI1				
> 🛩 System:DEBUG			ole I2C mode, ensure por	t pins output ty	pe is n-ch oper	n drain.
> 🖋 System:SYSTEM			between I2C and other n			
> Timers:AGT	\sim					



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.

[HS400x_RA2E1_NonOS] FSP Configuration	×			
Pin Configuration				O Generate Project Conten
Select Pin Configuration		📑 Export to	CSV file 🔚 Configure Pin E	Driver Warnings
RA2E1 EK	Manage configurations	Ger	nerate data: g_bsp_pin_cfg_	2e1
Pin Selection $\blacksquare \blacksquare \blacksquare \downarrow_Z^a P$	in Configuration			😲 Cycle Pin Group
Implementation ✓ ▲ Ports > ▲ Po > ▲ P1 > ✓ P2 > ▲ P3 > ✓ P4 > ✓ P5 > ✓ P9 > ✓ Other Pins ✓ Pripherals > ✓ Analog:ACMP > ✓ Analog:ANALOG > ✓ Connectivity:IIC	Name <	Value	Link Link Link IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	



6.1.2.4 [Stacks] Setting

Modify the configuration for individual components on the [Stacks] tabbed page. Modify the settings of I2C driver to suit the specifications of the target board.

(1) I2C Driver

(a) When using the same I2C driver as the original sample project

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

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

Proper	ties 🛿 🔝 Problems 👒 Smart Browser	📑 8 🗖 E		
_i2c0 12	C Master Driver on r_sci_i2c			
Settings API Info	Property V Common	Value		
arring.	Parameter Checking	Default (BSP)		
	DTC on Transmission and Reception	Disabled		
	10-bit slave addressing	Disabled		
	 Module g_i2c0 I2C Master Driver on r_sci_i2c 			
	Name	g_i2c0		
	Channel	2		
	Slave Address	0x00		
	Address Mode	7-Bit		
	Rate	Standard		
	SDA Output Delay (nano seconds)	300		
	Noise filter setting	Use clock signal divided by 1 with noise filter		
	Bit Rate Modulation	Enable		
	Callback	frm_comms_i2c_callback		
	Interrupt Priority Level	Priority 2		
	RX Interrupt Priority Level [Only used when DTC is enabled]	Disabled		
	✓ Pins			
	SDA	P302		
	SCL	P301		



(b) When using the different I2C driver as the original sample project

SAU0 is assigned to PMOD1 and SAU1 is assigned to PMOD2 on the FPB-RA0E1 board.

Delete the "I2C Master (r_sci_i2c)" stack and then add the "I2C Master (r_sau_i2c)" stack.

						-	SS *FHS40	x_RA0E1_NonOS] FSP Configuration ×			
Stacks Configuration				(Senerate Project Conte			Configuration		Ge	nerate Project Content
Threads New Thread Remove Permove Image: Additional and the state of the stat	g_ioport I/O Port (r_ioport)	g_his400x_sensor0 HS400 g_comms_i2c_device0 I2 g		nsor (rm_hs40		re (Threads	New Thread Remove E	g_ioport I/O Port (r_ioport)	 New Stack> Extend g_hs400x_sensor0 H5400X Temperature/Humidity (i) Sensor (rm_hs400x) f g_comms_j2c_device0 L2C Communication 	
د >		g_comms_i2c_bus012C S g g_i2c012C Master (r.sci_i g		>			<			Device Device () (m_comms_i2c)	
Objects New Object > 🎪 Remove	<	×	Validate Cut Copy Paste Delete Delete Import Export Module Resources	Ctrl+X Ctrl+C Ctrl+V Delete	DTC Driver for ption [Optional]	>	Objects	n New Object > n Remov	<		2C Master (r_sau_i2c)

To use PMOD1, set "Channel" to "00". To use PMOD2, set to "20".

🔲 Properties 🗙 🗐 Console 🚨 Renesas Debug	🎋 Debug	🛷 Search	🤤 Smart Manual	Progress	🏶 Sma	rt Browser		
						። 🝸 🗖	1	000

g_i2c0 I2C Master (r_sau_i2c)

ttings	Property	Value		
PI Info	✓ Common			
	Parameter Checking	Default (BSP)		
	Enable Critical Section	Disabled		
	Manual Start-Stop	Disabled		
	Enable Single Channel	Disabled		
	DTC Support	Disable		
	 Module g_i2c0 I2C Master (r_sau_i2c) 			
	Name	g_i2c0		
	Channel	20		
	Operation clock	СКО		
	Slave Address	0x00		
	Rate	Standard		
	Delay time (Microseconds)	5		
	Callback	🔒 rm_comms_i2c_callback		
	Transfer end interrupt priority	Priority 2		
	✓ Pins			
	SCL20	P112		
	SDA20	P110		



(2) I/O Port

Enter the name linked to the pin configuration to be used for "Pin Configuration Name" of "g_ioport I/O Port". In our example, it is "g_bsp_pin_cfg_2e1".

₿ [HS400x_RA2E1_NonOS] FSP Configuration ×			- 8
Stacks Configuration			Generate Project Content
Threads 🕢 New Thread 🔊 Remove 🕞	HAL/Common Stacks	New Stack	> 🚔 Extend Stack > 🙀 Remove
Image: Summary BSP Clocks Pins Interrupts Event Links S	g_ioport I/O Port (r_ioport)	 g_hs400x_sensor0 HS400X Temperature/Humidity Sensor (rm_hs400x) g_comms_j2c_device0 I2C Communication Device (rm_comms_j2c) g_comms_j2c_bus0 I2C Shared Bus (rm_comms_j2c) g_i2c0 I2C Master (r_sci_i2c) j 	 hs400x_delay Timer, General PWM (r_gpt) i
		🔗 Search 😳 Smart Manual 🔫 Progress 🌸 Smart Browser	rt 8 🗖 🖬
g_ioport I/O Port (r_ioport)			
Settings Property API Info 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 0 ELC Trigger Source Pin Configuration Name 		Value g_ioport Disabled	
		P300 D110	v

(3) Others

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 the Toolchain Setting

If you want to use a toolchain other than the GCC ARM Embedded toolchain, copy RRA_HS400X.c (for a non-OS system) or hs400x_sensor_thread_entry.c, sensor_thread_common.c, and sensor_thread_common.c (for a FreeRTOS or Azure RTOS system) from this project and create a new project.



6.2 RX Sample Project

Use the following procedures to modify a sample project.

This section describes an example of modifying the "HS400x_RX65N_NonOS" sample project so that it can be used on the RSKRX231 board.

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	
 ✓ Constant ✓ Archive File ✓ CMSIS Pack ✓ CMSIS Pack ✓ 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 ✓ Reneame & Import Existing C/C++ Project into Workspace ✓ Renesas CCRX project conversion to Renesas GCC RX ✓ Renesas CS+ Project for CA78K0R/CA78K0 ✓ Renesas GS+ Project for CC-RX and CC-RL ✓ Renesas GitHub FreeRTOS (with IoT libraries) Project 	~
(?) < <u>B</u> ack <u>N</u> ext > <u>F</u> inish 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.

← → × ↑ 📙 → e2_studio >	wor	rkspac	e	\sim	Ö	Search workspace	Q
Organize 🔻 New folder						8==	- ?
HS300x_RA6M4_FreeRTOS	^	Na	me			Date modified	Туре
HS300x_RA6M4_NonOS			HS300x_RA6M4_FreeRTOS			5/26/2021 5:27 PM	File fold
HS300x_RL78G14_NonOS			HS300x_RA6M4_NonOS			5/26/2021 5:30 PM	File fold
HS300x RX65N FreeRTOS			HS300x_RL78G14_NonOS			5/19/2021 8:41 PM	File fold
HS300x_RX65N_NonOS			HS300x_RX65N_FreeRTOS			5/25/2021 5:15 PM	File fold
	1		HS300x_RX65N_NonOS			5/25/2021 3:03 PM	File fold
	~	<					>
Folder: HS300	x_RX6	5N_N	onOS				
						Select Folder C	ancel



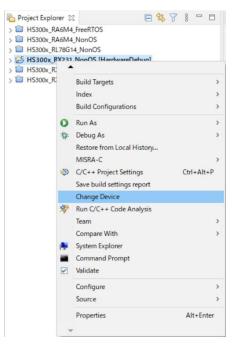
Enter the project name, select the original project for the current device, and press the [Finish] button.

📴 Import	_	
Rename & Import Proje		
Select a directory to search	for existing Eclipse projects.	
Project name: HS300x_R	(231_NonOS	
Use default location		
Location: C:¥U:	sers¥a5090534¥e2_studio¥workspace¥HS300x_F	B <u>r</u> owse
	eate Directory for Project	
Choose file system: defau	ilt \vee	
Import from: Select roo <u>t</u> directory:	C:¥Users¥xxxxxxx¥e2_studio¥workspace ~	Browse
O Select archive file:		Browse
Projects:		
	S (C:¥Users¥xxxxxxx¥e2_studio¥workspace¥HS	300x_RX65N_Non(
<		>
Options Keep build configurati	on output folders	
	onoupuriodels	
	K <u>B</u> ack <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 it to open the context menu. Select [Change Device] from the menu.



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

Refactoring				×
Change Device Select the new o	e device for HS300x_RX231_NonOS			
Current Device: Current Board: B				
Target Board:	RSKRX231			\sim
		Download ad	ditional boa	rds
Target Device:	R5F52318AxFP			
		1	Unlock Devic	ces
?	< <u>B</u> ack <u>N</u> ext >	<u>F</u> inish	Cancel	



If a warning message appears, read it and check if it will create a problem in proceeding with the procedure. Press [Next] to move 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
It is change cannot be undone. Please make sure you backup this	s project	before
No context information available		
? < <u>B</u> ack <u>N</u> ext > Einish	Cance	el

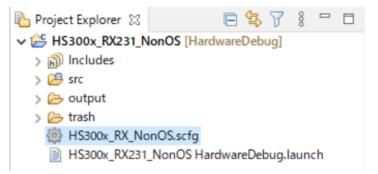
The changes you have made to the settings will be displayed. Press the [Finish] button to apply the changes to the project.

I Refactoring			×
Change Device			-
The following changes are necessary to perform the refactoring.			
Changes to be performed		₽ 🗘	7 -
✓ ✓ 叠 Change Device for HS300x_RX231_NonOS			
🗸 🗹 🚰 Launch Configurations			
🖂 🚖 HS300x_RX231_NonOS HardwareDebug			
> 🗹 🔂 Build Settings			
🗹 🔂 Project Files			
🗹 🚖 Smart Configurator			
No preview available			
? < <u>Back</u> Next > Einis	h	Cance	el l



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_F	X_NonOS.scfg ⊠		- 6
Device s	election	Generate Code	Generate Report
Device sel	ection		èď
Board:	RSKRX231 (1.00) 🗸		
Device:	R5F52318AxFP		
	Download more boards		
venview D.	oard Clocks System Components Pins Interrupts		

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

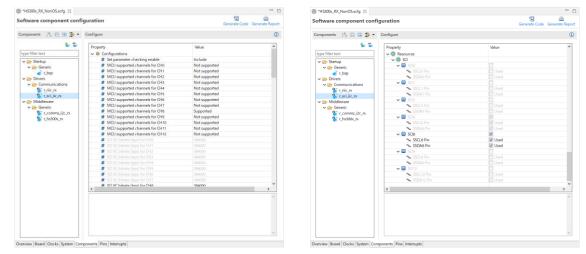
	HS300x_RX_NonOS.scfg 33			御 *HS300x_RX_NonOS.scfg 11		
	Clocks configuration			Clocks configuration		
			4 0445 5 5 6 5 6 6 6 6 6 6 6 7 6 7 6 7 6 7 6 7 6 7 6 7 7 7 6 7 7 7 6 7 7 7 6 7 7 7 6 7 7 7 7 7 8 7 9 7 9 7 9 7 9 7 9 7 9 7 9 7 9 7 9 7 9 7 9 7 9 7		Hang tan An and t	4 0.84 2000 10000 10000 2000 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000 200 10000 10000
Image: Second						
erview [Board Clocks [System [Components Piss] Interrupts]	MOT-skelisatel kon-speed dook		CASEGUE (MAR) - grower free rules (JPICES) - (MAR) MINTOLICACION - (MAR) MINTOLICACION - (MAR)	M01-Sed seed to-speed dock		CACEGOLY - Dolla Low gauge time class (UPPER) - Dolla MISTOCIACAD(CX - BOHD - BOHD - ATCOCIX
	Dverview Board Clocks System Components P	ins Interrupts		Overview Board Clocks System Components P	Pins Interrupts	



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 the checkboxes for "SSCL8 Pin" and "SSDA8 Pin" for "SCI8" under "Resources".



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

configuration			Generate G	ode	Generate Re
rdware Resource 🛛 🕀 🗎 🛔	Pin Function	n	2		I 🖬 🔤
ype filter text	type filter	text (* = any s	tring, ? = any character)	di	
TMR2 TMR2 TMR2 TMR3 TMR3 Scrial communications intr SCI0 SCI	Enabled	Function CTS8# RTS8# RXD8 SCK8 SMISO8 SMISO8 SMISO8 SMISO8 SMISO8 SMISO8 SMISO8 SMISO8 SMISO8 SSDA8 TXD8	Assignment Not assigned Not ass	111111	n Number Not assignee Not assignee Not assignee Not assignee Not assignee Assignee Assignee Assignee Assignee
CMPB3 Capacitive touch sensing u Analog power supply W I/O ports	<				

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

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 code.

verview information			Generate Code Generate Rep
General Information			0
his editor allows you to modify the settings stored in config	juration file (.sc	rfg)	
loard			
llow board and device selection			
Docks Allow clock configuration			Application under development
Components Allow software component selection and configuration			Middleware Device RTOS
Pins			
Allow general pin configuration and pin configuration for se	elected software	e component	
nterrupt Allow general interrupt configuration and interrupt configur Tick have to get more information on User's Manual Palace			
Ilow general interrupt configuration and interrupt configur Ilick here to get more information on <u>User's Manual, Releas</u>			
Now general interrupt configuration and interrupt configur Lick here to get more information on <u>User's Manual, Releas</u> Current Configuration	e Note Applica	ation Notes Tool News	
Now general interrupt configuration and interrupt configur Lick here to get more information on <u>User's Manual, Releas</u> Current Configuration elected board/device: RSF52318AvFP_(ROM size: S12 Kbytes	e Note Applica	ation Notes Tool News	e
Now general interrupt configuration and interrupt configur Lick here to get more information on <u>User's Manual, Releas</u> Current Configuration	e Note Applica	ation Notes Tool News	R
Allow general interrupt configuration and interrupt configur Click here to get more information on <u>User's Manual Release</u> Current Configuration elected board/device: RSFS2318AxFP (ROM size: S12 Kbyter inerrated location (PROJECT_LOC#): sec¥smc_gen	e Note Applica	ation Notes Tool News	R.,
Now general interrupt configuration and interrupt configur Lick here to get more information on <u>User's Manual, Release</u> Current Configuration elected board/device: RSFS2318AvFP (ROM size: S12 Kbytes Generated location (PROJECT_LOC®): <u>srcRsmc_gen</u> elected components:	e Note Applica	ation Notes Tool News Kbytes , Pin count: 100) Ec	R
Now general interrupt configuration and interrupt configur Lick here to get more information on <u>User's Manual Releas</u> Current Configuration elected board/device: R5F5218AxFP (ROM size: 512 Kbyter isenerated location (PROJECT_LOCØ): <u>scrEvanc_gen</u> elected components: Component	e Note Applica s , RAM size: 64 Version	Ation Notes Tool News Kbytes , Pin count: 100) Ec Configuration	it
Allow general interrupt configuration and interrupt configur Current Configuration elected board/device: R5F52318AxFP (RCM size: 512 Kbyter Enerated location (PROJECT_LOCF): src4smc_gen elected component: Component © Board Support Packages. (r_bsp)	e Note Applica s , RAM size: 64 Version 5.66	tion Notes Tool News Kbytes , Pin count: 100) Ec Configuration r_bsp(used)	
Now general interrupt configuration and interrupt configur Lick here to get more information on <u>User's Manual, Release</u> Current Configuration elected board/device: RSF52318AxFP (ROM size: 512 Kbyte) ielected contion (RPOJECT_LOCQF): generated location (RPOJE	e Note Applica s , RAM size: 64 Version 5.66 0.62	Kbytes , Pin count: 100) Configuration r_bsp(used) r_hs300c_r(used)	

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 the Toolchain Setting

If you want to use a toolchain other than the CC-RX toolchain, copy RX_HS400X.c (for a non-OS system), main.c and hs400x_sensor_thread_entry.c (for a FreeRTOS system), or hs400x_sensor_thread_entry.c, sensor_thread_common.c, and sensor_thread_common.c (for an Azure RTOS system) from this project and create a new project.



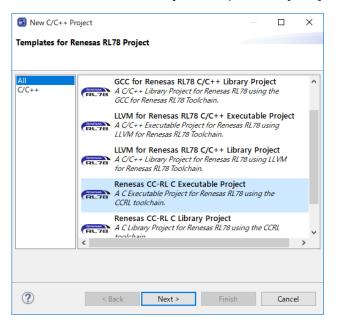
6.3 RL78 Sample Project

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.



Enter the project name ("HS400x_RL78G1G_NonOS" in this example) and press the [Next] button.

0		—		×
New Renesas CC-	RL Executable Project			4
New Renesas CC-R	L Executable Project			4
Project name: HS	300x_RL78G1G_NonOS			
Use <u>d</u> efault lo	cation			
Location:	C:¥Users¥a5090534¥e2_studio¥workspace¥HS300x_RL78G1G_NonOS		B <u>r</u> owse.	
	Create Directory for Project			
Choose file system	$_{ m s}$ default \sim			
Working sets				
Add projec <u>t</u> t	o working sets		Ne <u>w</u>	
Working sets:		~	S <u>e</u> lect	
0	< Back Next > Finish		Cance	
	< <u>Back</u> <u>N</u> ext > <u>Finish</u>		Cance	1



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

0							×
Device Selection You can filter devices by	regular exp	pression					-
Search Device							
Device > RL78 - F1E > RL78 - G10 > RL78 - G11 > RL78 - G12 > R178 - G13	RAM	ROM	Pin	RTOS	Smart Co	Peripher	
> RL78 - G13A > RL78 - G14 > RL78 - G14							
> RL78 - G1C > RL78 - G1D > RL78 - G1E > RL78 - G1F							
 RL78 - G1G RL78 - G1G 30pin RL78 - G1G 32pin RL78 - G1G 44pin R5F11EF8 							
✓ R5F11EFA							
R5F11EFA > RL78 - G1H > RL78 - G1K	1 KB	16 KB	44		×	1	,

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

8			\times
New Renesss CC-RL Executable Project Select Coding Assistant settings			\$
Use Smart Configurator	interface ripheral f	(GUI).	
(?) < gack Next > Einish		Cance	ł

Press the [Finish] button.

0						×
New Renesas CC-RL Ex Summary of project "HS3						\$
TODICHAIN NAME: TODICHAIN VERSION : GENERATION FILES : generateKistart.asm generateKistini.asm generateKistini.asm generateKistini.asm	Reneas CCRL v1.10.00					~
?		< <u>B</u> ack	<u>N</u> ext >	<u>E</u> inish	Cancel	



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	🛛 🦼 Co	de Preview 📲	Device Top View 📲 De	evice List View 🔲 Proper				=
					🐻 Genera	ate Code	0	000
		Block diagram	On-chip debug setting	Confirming reset source	Safety functions			^
Pin assignment s	etting							
Once the pin assi A new project mu	st be created ·	been fixed it is r to change the se Fix settings	not possible to change th ttings.	em later.				
PIOR register	Function	Port setting		1				
PIOR11, PIOR10	TRJIO0	P01		/				
PIOR13, PIOR12	TRJO0	P30		/				
								ł
								1

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

	amil	and a state		_				
📱 Peripheral Fun 🔀 📑 Code Previ	aw 📲 Device Top	View 归 D	evice List Vie	ew 🔲	Propert	-	-	
						🐻 Gener	ate Code	0
	diagram On-chip o	lebug setting	Confirmine	g reset	source	Safety functions		
Operation mode setting								
● High-speed main mode 4.0 (V) ≤								
○ High-speed main mode 3.6 (V) ≤								
○ High-speed main mode 2.7 (V) ≤								
○ Low-speed main mode 2.7 (V) ≤	VDD ≤ 5.5 (V)							
Main system clock (fMAIN) setting								
 High-speed OCO (fIH) 	⊖ Hi	gh-speed sys	tem clock (f	MX)				
High-speed OCO clock setting								
Operation	Frequency 48 (f	HOCO=48, fIH	H=24)	~	(MHz)			
High-speed system clock setting								
Operation								
	O E	ternal clock i	input (fEX)					
Frequency	5				(MHz)			
Stable time	2^18	/fX	52428.8		(µs)			
Low-speed oscillation clock (fIL) settin	e							
Frequency	15				(kHz)			
Interval timer operation clock/Timer R.	Lount source setti	ne						
Interval timer operation clock/Timer			✓ 15		(kHz)			
		r	-					
CPU and peripheral clock setting CPU and peripheral clock (fCLK)	fIH		24000		(kHz)			
or o and peripheral clock (rock)	1411		24000		(1112)			



Select "Used" for "On-chip debug operation setting" on the [On-chip debug setting] tabbed page for "Peripheral Functions".

🧱 Peripheral Fun 🛛 😹 Code Preview 🚆 Device Top View 🚆 Device List View 🔲 Properties 🌉 FIT	Configura.	
	Generate Code	
Pin assignment Clock setting Block diagram On-chip debug setting Confirming reset source Safety fun		×
-On-chip debug operation setting		
O Unused Used		
- RRM function setting		
O Unused Used		
-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 [Channel] tabbed page in the [Serial Array Unit] or [Serial] setting window.

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

💯 *Peripheral F	u 🛙	🧾 Code Pre	view 🕎	Device Top View	💹 Device List View	Properties	🕎 FIT Configura	-	
							🐻 Generate Code	0	000
<u>Serial Array Uni</u>	<u>t 0</u>								^
Channel UAR	TO UAP	RT1 CSI00	IC00						
-Function									
Channel 0	IIC00	~							
Channel 1	Unuse								
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 level, and enable all functions under "Callback function setting".

	vice Top View 🚆 Device List View 📃	🐻 Generate Code	3
Serial Array Unit 0			,
Channel UARTO UARTI CSIOO			
Transfer rate setting			
Transfer rate	100000 v (t	ops) (Actual value: 100000)	
Interrupt setting			
Transfer end interrupt priority (INTIC00)	Low		
-Callback function setting			
Master transmission end	🖂 Master reception end	🖂 Master error	

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

👮 *Peripheral Fu 🔀	🛃 Code Preview	🕎 Device Top View	Device List View	Properties	💯 FIT Configura	-	
					🐻 Generate Code	0	000
Transfer mode Settin	£						^
 Unused 							
Single master							
◯ Slave							
<							۷
						>	



On the [Setting] tabbed page for the channel set to 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 a desired level, enable all functions under "Callback function setting", and disable "Callback function enhanced feature setting".

Transfer mode Setting							
نــــــــــــــــــــــــــــــــــــ					🐻 Generate Code	01	000
Clock mode setting							^
🔾 folk 🚺		fCLK/2					
-Local address setting —							
Address		16					
-Operation mode setting							
 Standard 		◯ Fast mod	de/Fast mode plus	Digital filter	on		
Transfer clock (fSCL))	100000	(bps) (Actual value: 991	73.554)		
-Interrupt setting			~				
-Callback function setting Master transmissi		Master reception e	nd 🖂 Mas	ter error			
-Callback function enhand		ransmission/reception	n end callback function	n			-
<						>	. `

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 Preview 月 Device	e Top View 🛛 🔛 Device List View	Properties	💯 FIT Configura	
				🐻 Generate Code	3
<u>General setting</u>	Channel 0 Channel 1 Channel 2	Channel 3			^
Functions					
Channel 0	Interval timer	~			
Channel 1	Unused	~			
Channel 2	Unused	~			
Channel 3	Unused	~			
c					>



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	00	-	
TAU0 TAU1 TMRJ0 TMRD0 TMRD1 TMRG0	1					^
General setting Channel 0 Channel 1 Channel 2	Channel 3					
_Interval timer setting						
Interval value (16 bits)	100 µs ~ (Actual va	lue: 100)				
Generates INTTM00 when counting is started						
_Interrupt setting						
☑ End of timer channel 0 count, generate an inter	rrupt (INTTM00)					
Priority	Low \checkmark					
						¥
<					>	

Press the [Generate Code] button to generate 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.

Definition for including r_comms_i2c_if.h:

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

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

```
* Function Name: r iic00 callback master error
* Description : This function is a callback function when IIC00 master err
* Arguments : flag -
           status flag
* Return Value : None
static void r_iic00_callback_master_error(MD_STATUS flag)
  /* Start user code. Do not edit comment generated here */
  rm comms i2c bus0 callback(true);
  /* End user code. Do not edit comment generated here */
}
* Function Name: r iic00 callback master receiveend
* Description : This function is a callback function when IIC00 finishes
* Arguments : None
* Return Value : None
static void r iic00 callback master receiveend(void)
{
  /* Start user code. Do not edit comment generated here */
  rm comms i2c bus0 callback(false);
  /* End user code. Do not edit comment generated here */
}
* Function Name: r iic00 callback master sendend
* Description : This function is a callback function when IIC00 finishes
* 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 HS400X */
g_hs400x_sensor0_quick_setup();
while (1U)
{
   start_hs400x_demo();
}
```

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

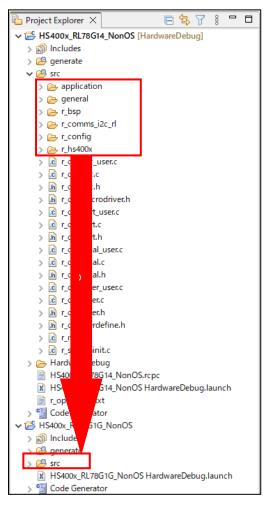
```
void g_comms_i2c_bus0_quick_setup(void)
{
    /* bus has been opened by startup process */
}
void demo_err(void)
{
    while(1)
    {
        // nothing
    }
}
```



6.3.4 Modifying Sample Source Files

Right-click on the "application" "general" "r_bsp" "r_comms_i2c_rl" "r_config" "r_hs400x" folder in the project tree of the "HS400x_RL78G14_NonOS" sample project and select [Copy] from the context menu.

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





Open 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:

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

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

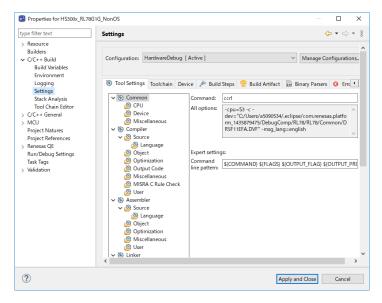
Modify "r_cg_serial.h" to "r_cg_sau.h" or "r_cg_iica.h":

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.

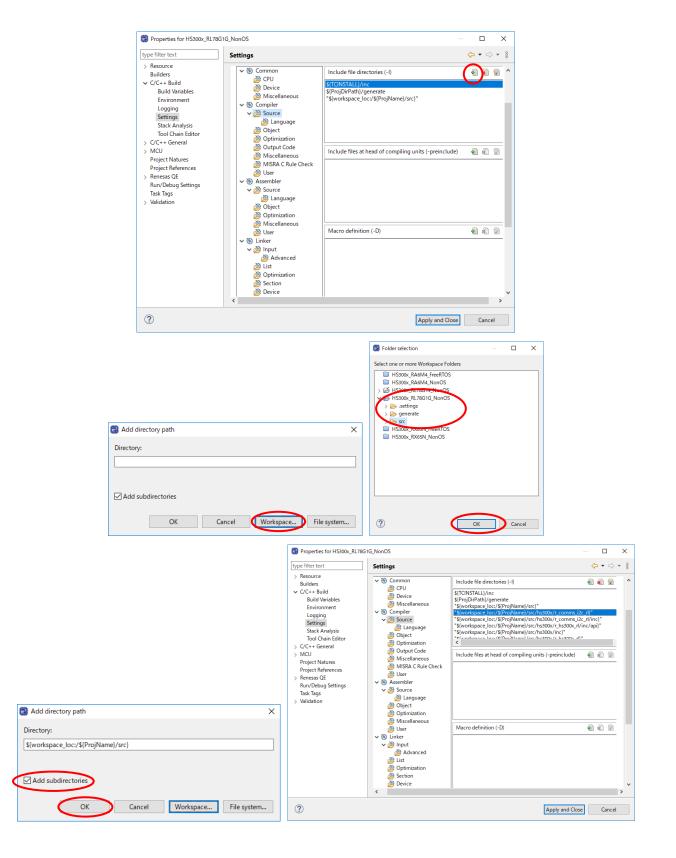




Select [Compiler] \rightarrow [Source] on the [Tool Settings] tabbed page and press the [Add] icon.

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.

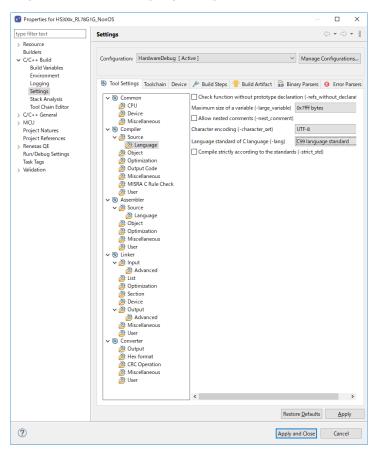
Select the checkbox for "Add subdirectories" and press the [OK] button.





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 "HS400x_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
🕼 Archive File
CMSIS Pack
😭 Existing Projects into Workspace
😂 File System
GNUARM-NONE/RZ(DS-5) project conversion to GCC ARM Embedded
Preferences
😂 Projects from Folder or Archive
Rename & Import Existing C/C++ Project into Workspace
😂 Renesas CS+ Project for CA78K0R/CA78K0
😂 Renesas CS+ Project for CC-RX and CC-RL
😂 Renesas GitHub FreeRTOS (with IoT libraries) Project
🕐 Sample Projects on Renesas Website
✓ (> C/C++
C/C++ Executable
Image: Second



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 xxx	xxxxx > e2_studio > import	ע פֿע גע Search	n import
Organize 🔻 New folde			:== ▼ ?
	Name	Date modified	Type S
📌 Quick access	HS400x_RZG2L_FreeRTOS	7/28/2022 3:40 PM	File folder
len OneDrive	HS400x_RZG2L_NonOS	7/28/2022 3:41 PM	File folder
This PC			
Network			
VIELWORK			
	<		>
Folder	HS400x_RZG2L_NonOS		
		Select Folde	r Cancel

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

📴 Import				\times				
Rename & Import Project Select a directory to search for existing Eclipse projects.								
Project name: HS	400x_RZG2L_NonOS							
Use <u>d</u> efault loc	ation							
Location:	C:\Users\xxxxxx\e2_studio\workspace\HS400	Dx_R	B <u>r</u> owse.					
	Create Directory for Project							
Choose file system	default 🖂							
Import from:								
Select root dire	ctory: C:\Users\xxxxxx\e2_studio\import\HS4	~	B <u>r</u> owse					
◯ Select <u>a</u> rchive f	ile:	~	B <u>r</u> owse					
<u>P</u> rojects:								
HS400x_RZG2L_	NonOS (C:\Users\xxxxxxx\e2_studio\import\HS	400x_F	RZG2L_Non	OS)				
Options								
Keep build con	figuration output folders							
?	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>		Cance	el l				



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.

🌞 [HS400x_RZ	G2L_Nor	nOS] F	SP Configu	ration $ imes$							
Board Sup	port P	acka	ige Cont	figuratio	n				Generate F	Oroject Co	ontent
									R	estore De	efaults
Device Selec	tion										
FSP version	: 1.1.0-r	c0+20	220727.044	08375	~	Board De	etails				
Board:	RZ/G2	L Eval	uation Kit (SMARC)	- 24	1					
Device:	RZ/G2	LC Eva	r Board (An aluation Kit	(SMARC)		1					
Core:			uation Kit (S aluation Kit		~						
RTOS:	No RT	DS			~						
Summary BSP	Clocks	Pins	Interrupts	Event Links	Stacks	Components					

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

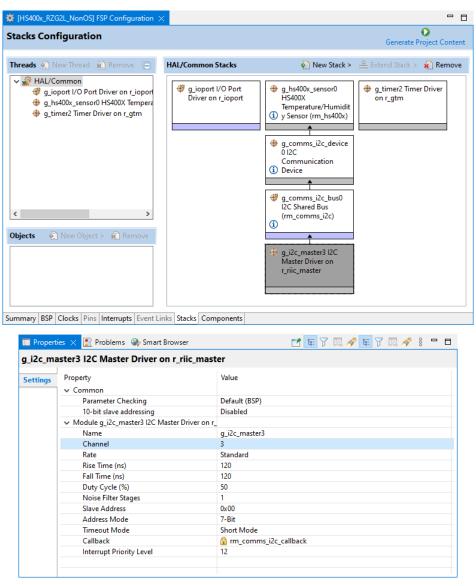
When "Custom User Board (Any Device)" is selected for "Board", set up the clocks to suit the specifications of the target board to be used.

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

🔅 [HS400x_RZG2L_NonOS] FSP Configuration \times	
Clocks Configuration	Generate Project Content
	🔣 Restore Defaults
OSC 24MHz	^
ICLK 1200MHz	
I2CLK 200MHz	
GCLK 500MHz	
SOCLK 12kHz	
S1CLK 6kHz	
SPI0CLK 200MHz	
SPI1CLK 100MHz	
SDOCLK 533MHz	
SD1CLK 533MHz	~
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.

In this chapter, HS300X is used as example figures, but HS400X works the same way.

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

🐐 🐞 💼 🔯 Debug 🧹 💽 RA_HS300K Debug_Flat	New Window Editor	,	🛛 🕤 • 🐔 • 🔝 🖉	🕐 🏉 🛷 • 📃	9 2 9 1 = 1 3, 9 , 5 1 = 1 = 4 3	••• ••••••••••••••••••••••••••••••••••	8.60 49.14	> ∯ + ∦ ·			
	Appearance										SP Configuration 🔅 Debu
E Bebug 22 E 🛬 🕸 8 ™	Show Yere		Breakpoints	A1+9-0-8	p 🗟 main.c 📓 bsp_delay.c	- 0	On Veriables	🍫 Breakpol			22 🛅 10 Regis 🐃
RA_HStock Debug_Flat [Renesas GD8 Hardware Debugging]	Perspective		Console	AR+ShiR+O.C		^				S 🚳 🖂 🔶	at % 🚯 📚 🖪 🖻
RA_HS300K.eff [1] (cores: 0]			b Debug	Ant animation of			Expression	Type	Value	Address	
 Phread #11 (vingle core) [core: 0] (Suspended: Breakpoint) start demoi) at RA HS300Kcs11 0x278 	Navigation		 Debugger Consolo 				🔶 Add n	5			
hal entry() at hal entry()14 0x364	Preferences										
main() at main.c5 0x3x2	19		Debug Shell								
am-non-esti-adb (7.6.2)	218		Debug Sources								
I Renesas GDB server (Host)	21		Disassembly								
	22	1.1	Error Log	Alt+Shift+Q.L	ect */						
	24		D frecutables		1						
	25		12 Expressions								
	26										
	27 28		Memory								
	29		Memory Browser		gs demo callback status;						
	310	1.1	Modules		gs down humidity;						
	21		Outine	All+Shift+Q.Q	gs_demo_temperature;						
	32		Peripherals								
	23 24										
	35		Problems	Alt+Shift+Q, X							
	36		Progress								
	57		Project Explorer								
	38 39		11 Registers		DVENCE_1;						
	29		Signals								
	41 0000027										
	42		S Templates								
	43	0	0: Variables	Alt+Shift+Q, V							
	41 0000028 45	· •	スマート・ブラウザー								
	40	9 0	B 37-1-71171								
	47		4 ALL 1-100								
	48 0000028	: e	Other	Alt+Shift+Q,Q							
	49		1								
	50		case DENO	SEQUENCE_1 1							
	52		/* Cle	ar status 1/							
	53 000002a	5	gs_den	o_callback_stat	us - DEND_CALLBACK_STATUS_NAIT;						
	54					~					
		<				>					<
					apper Console 🌒 スマート・ブラウブー 👖 Marrory						
			mesas GDB Hardware Debu		addit commer all Xx+5-2222- 0 manual				a w 961 u	a di na Pasia	
					ith data ffffffffffffffffffffffffffffffffff						
	Option Functs	on sere	ict, writing to addre	55 0x01005100 w	ith data fffdffffffffffffffffffffffffffffff						
	972/0-H447	an aver	ee, writing to some								
	ハードウォアーブレーク	成した	TRUZEN INCOMELLEN.								
			アドレスex278に終至します。								
			(アドレスex278に検索します。								
	ハードウェア・ブレー	味白日	「アドレスex270に指定します。								
	ハードウェア・ブレー	速(2)10	フドレス8×278に発達します。								
	0-r917-70-5	im 2N	にアドレスex278に設定します。								
55D Secure - 0x0000278	Ø 372548 ns										
SSD.Secure			DWT 🚓 74		Writable	Smart Insert 41:1					

Click on [Add new expression] on the [Expressions] tabbed page and add "gs_hs400x_data".

🖌 🎲 🔳 🅸 Debug 🗸 🔀 HS3034_RA6M4_NorOS Debug_R	4×0 10-	10 D	•呜•自父郎郎父•汉帝公女第日弟父亲父郎弟父母父母亲。	杨田田	四馬 四 学 (金) 但 -	51-995	2.0.1	3	
							٩	18 Bac	C++ (Debe
Debug 🖂 📴 👘 👔 🖳 🗆	(i) startup.c				00 Vari 🔩 Bre 📓 Mod	Pa Proj 💀 De	e 🛪 🚜 16	55 Peri	i in 😐 i
E HS30x, RA9M4, NoxOS Debug, Fait (Remasas GD8 Hantware Debugging)	1	minclud	e "hel_cate.h"	^		4	14 0 4		8 0 8
V B HS300x_RAnM4_NonOS.elf [1] [correct]	2	/* Seco	ance "/		Expression	Type	Value	Addres 8	land 1 gi_hi3
 ✓ Intend #1.1 (single core) [core: 0] (Suspended) Step) Start. domo() at RA. HS300X.cr2 0x208 	6		man s_demo_sequence		w C co loster data	velatile em	6.3	(hc2D3D)	Details:{
 start_domoj) at KA_Hissolucu/2 0x20s Isal entry() at hal entry of \$ 0x300 	5	- (v A humidity	m_hs300v	6-3	0x20300	Decimalif
main() at main();5 0(35e	0	001	2_sequence_2 = (3), 5_sequence_2,		(+)- integer_part	incl6_t	0	0.20000	Hex: {}
am-none-esti-odb (7.8.2)		0029	STONEDUCE 2.		b)- decinal part	int16.1	0	0+20300	Binary:{.
JR Feneral GER server (Hott)	9		septence_4,		v 进 temperature	m_hs500x	6-3	0.2030	Octal:{
No of the state of	18	DEM	. SEQUENCE 5, 2. SEQUENCE 6,	-	pil- integer_part	int16_t	0	0x20300	
	12) dens	sequence to		bl> decimal.pert	intl6,1	0	0.20000	
	13	T outro	Access of the second seco		4 Address supression				
	14	/* Call	back status "/						
	15	8 typedef	enum s_demo_callback_status						
	16	1	P CALLWER STATUS WALT - (0).						
	18		D CALLBACK STATUS SUCCESS.						
	19	DCH	CALLBACK STATUS REPEAT,						
	-20] demo_	callback_status_t;						
	21		beveloper Assistance in the project */						
	23		comes 12c bus0 quick setup(void):						
	24	wid g]	hs100x sensor0 quick setup(void);						
	25								
	26 27	void	start_demo(void); void demo err(void);						
	27	static	vore demo_env(vore):						
	29	static .	velatile demo_callback_status_t _gs_demo_callback_status;						
	20	static	volatile re holdow data t go holdow data;						
	31								
	32		k setup for g comms lize bus0. "/ rommes lize bus0 quick setup(vold)						
	34	dura gu	the first free free free free free free free fre						
	35	fsp	err t errs						
	36 88888248	120	master_instance_t * p_driver_instance - (ilc_master_instance_t *) g_comms_ilc_bus0_extended_ctg.p_dri	LV .					
	37 38		Deen 12C onlyon, this must be done before calling any COMPS API */						
	38 38 080802.05		> p driver, this must be done before calling any COMPS And 17 > p driver instance-to api-topen(p driver instance-to ctr), p driver instance-to cfg);						
	48 89696259		ert[FD SECESS err]]						
	41	>	South States						
	42			~					
		<		>	¢			> 1	
	Console 12	Registers	Debug Shell 🔝 Problems 🔛 Debugger Console 👒 37 - 1-7577 - 👖 Memory			I X & B	ht 70 4 3	D	
	HIGHON PARMA NO	and S Debug Fl	at (Renesas GDR Hardware Debugging)						
	S-P-SHIMMET								
	608: 61871								
	なープトー経営状況	- CK							
	ターデャー 保留状況	- CK							
	9550-F#85								
			riting to address dw0100a100 with data ffffffffffffffffffffffffffffffffff						
	STUD-NET	e perect' M	ricing to depress exercedate with data interimitation interimitation interimitation in the						
	ハードウェア・フレーク		erses: Neti 17.						
	71-1927-71-95	ポイントをアドレス	ex.588(1970),27,						



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

🚯 📑 🕲 Delway 🗸 💽 HS300x, RA6M4, NaxOS Delway	NYO ICHING BIS BIRD AND IN MINING AND	18 CP III [2] [2] [2] [2]	3 8 - 9 - 9 - 9			
						It Deb
ig 🛙 👘 🙀 🖬 👘 🗖 🗖		= 🗆 🕫 Vari % Bre			5 Peri	
NEISSE KARK NACH Sharp Full Fundament (Charabar Sharpar) (2006). Mark (1006) (100 mt) (1006). Mark (1006) (100 mt) (1006). Mark (1006) (1006) (1006). Mark (1006). Mark (1006) (1006). Mark (1006). Mark (1006). Arrange (1006). Mark (1006). Arrange (1006). Mark (1006). Arrange (1006). Mark (1006).	<pre>Higher to (a, b)</pre>		Type Select All Copy Inpressions Remove Remove All Number Format Add Econesion Grou Find Show Details As Add Watch Expression	Chi+F Musi Musifut expression i Musi C++J	Addres: Na 2006 2006 2006 2006 2006 2006	E C C C
	4 Januar Marcallan and Carlos and				> <	
		· .		A DI 2 5 5		
	😰 Console 💠 Will Registres 👔 Debug Shell 🐮 Problems 🎡 Debugger Console 📦 スマート・ブラウザー 👔 Memory H1300x, RABMA, NonOS Debug, Fait (Romass GBB Hardware Debugger g)		# X %	原加成合金		13 * *
	Is -for BMRYT Mins start North Mins (Mins) North Mins (Mins) North Mins (Mins) North Mins (Mins) Mins (Mins) Mins) Mins (Mins) Mins (Mins)					

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

			Q	10 1 10 OC++ 10 D
ing 12 B H B P I		🗆 (H-Vari 💁 Bro 📷 Mod	🔁 Paj 💶 Dap 🗙 🥔 (K 15 Per 10 10 10
(c) A part (b) (C) (c	<pre> ()</pre>	C S	L L <thl< th=""> <thl< th=""> <thl< th=""> <thl< th=""></thl<></thl<></thl<></thl<>	Addrem 0.0007
	Console 12 Will Registers D Debug Shell 22 Problems R Debugger Console @ 27-1-7517 1 Memory HSI00: RASMA NextOS Dobus, Flat (Remotes GDB Hordware Debugger (or)		= x % 6 61 P 5	5 0 · C · ·
	 			



Revision History

		Description					
Rev.	Date	Page	Summary				
1.00	June 30, 2022	-	First Release				
1.10	August 30,2022	-	Add : RZ items				
1.11	March 3, 2023	-	Updated: Environments for RL78				
1.12	March 29, 2023	-	Updated: Environments for RA, RX, RL78, RZ				
			Updated: Main Processing Flow of Sample Software				
			Updated: Guide for Changing the Target Device				
1.13	September 7,	-	Updated: Guide for Changing the Target Device				
	2023		Deleted: RE01 items				
1.14	April 9, 2024	-	Add : Environments for RA0E1				
			Updated : I2C Driver Settings of RA Family				
			Updated : Guide for Changing the Target Device of RA				
			Sample Project				



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.
- Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
 - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

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

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

- 7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
- 8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas
- Electronics products. (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled
- (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/.