

RA Family, RX Family, RL78 Family FS3000 Sample Software Manual

Introduction

This application note describes the sample software that is for use with the FS3000 flow sensor and runs on certain MCUs of the RA family, and RX family, RL78 family.

Target Devices

RA6M4 Group

RX65N Group

RL78/G14 Group

RL78/G23 Group

Trademarks

FreeRTOS[™] and FreeRTOS.org[™] are trade marks of Amazon Web Services, Inc. Microsoft[®] Azure RTOS is trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

Con	tents	1
1.	Overview	3
2.	Environment for Confirming Operation	
2.1	Environment for Confirming Operation on the RA Family MCU	3
2.2	Environment for Confirming Operation on the RX Family MCU	
2.3	Environment for Confirming Operation on the RL78/G14 Group MCU	
2.4	Environment for Confirming Operation on the RL78/G23 Group MCU	6
3.	Sensor Specifications	7
3.1	Overview of Sensor Specifications	
3.2	Sensor Functions	8
4.	Sample Software Specifications	9
4.1	Configuration of the Sample Software	
4.2	Specifications of Sensor API Functions	
4.2.1		
4.2.2	5	
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 Software1	3



4.4.1	Azure RTOS Project	15
5. (Configuration Settings	16
5.1	FS3000 Air Velocity Sensor Settings	16
5.1.1	RA Family	
5.1.2	RX Family	
5.1.3	RL78 Family	
5.2	Communication Driver Middleware Settings	19
5.2.1	RA Family	
5.2.2	RX Family	
5.2.3	RL78 Family	21
5.3	I2C Driver Settings	
5.3.1	RA Family	
5.3.2	RX Family	
5.3.3	RL78 Family	
6. (Guide for Changing the Target Device	30
6.1	RA Sample Project	
6.1.1	Importing the Sample Project	
6.1.2	Modifying Settings of the FSP Configurator	
6.1.3	Changing toolchain setting	
6.2	RX Sample Project	
6.2.1	Importing the Sample Project	
6.2.2	Changing the Device	40
6.2.3	Modifying Settings of the Smart Configurator	
6.2.4	Changing toolchain setting	
6.3	RL78 Sample Project	45
6.3.1	Creating a New Project	45
6.3.2	Settings of the Code Generator	47
6.3.3	Modifying the Generated Code	51
6.3.4	Modifying Sample Source Files	
7. `	Viewing Air Velocity Data	60
Povie	sion History	62
I/CVI		02
Gene	eral Precautions in the Handling of Microprocessing Unit and Microcontroller Unit P	roducts63
Notic	ce	64
Corp	orate Headquarters	64
Cont	act information	64
Trade	emarks	64



1. Overview

This sample software acquires data from the FS3000-1005 air flow sensor and handles calculations on the data. In combination with the I2C driver of the FSP or FIT, the sample software controls the FS3000-1005 through the I2C in the MCU to acquire ADC data from the sensor and calculate the air velocity.

2. Environment for Confirming Operation

2.1 Environment for Confirming Operation on the RA Family MCU

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

Item	Description
Demonstration board	RTK7EKA6M4S00001BE (EK-RA6M4)
Microcontroller	RA6M4 (R7FA6M4AF3CFB: 144 pins)
Operating frequency	200 MHz
Operating voltage	5 V
Integrated development environment	e ² Studio 2023-01
C compiler	GCC 10.3.1.20210824
	IAR Toolchain for ARM - (8.x) 8.1.0.202011101213
	MDK-ARM Ver.5.34
FSP	v3.8.0
RTOS	FreeRTOS [™] and 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	PMOD Daughter Card for FS3000 flow sensor (US082- FS3000EVZ)

Table 2-1 Operating Environment for RA Family

Table 2-2 Amount of Memory Used in RA Family

Area	Size (Non-OS)	Size (FreeRTOS)	Size (Azure RTOS)
ROM	1,295 bytes	1,628 bytes	1,572 bytes
RAM	73 bytes	253 bytes	422 bytes

Calculation of these sizes in memory only takes functions and variables related to the FS3000 sensor into account. They do not include the sizes of RTOS threads for the RTOS versions.

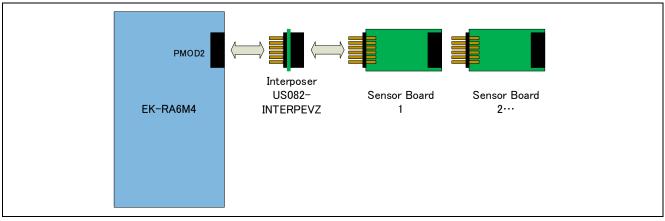


Figure 2-1 Hardware Connections for RX Family



2.2 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.

ltem	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.3.02.00
	GCC 8.3.0.202004
	IAR Toolchain for RX 8.4.10.7051
FIT	BSP v7.20
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	PMOD Daughter Card for FS3000 flow sensor (US082-
	FS3000EVZ)

 Table 2-3
 Operating Environment for RX Family

Table 2-4 Amount of Memory Used in RX Family

Area	Size (Non-OS)	Size (FreeRTOS)	Size (Azure RTOS)
ROM	1,467 bytes	1,685 bytes	1,745 bytes
RAM	145 bytes	205 bytes	414 bytes

Calculation of these sizes in memory only takes functions and variables related to the FS3000 sensor into account. They do not include the sizes of RTOS threads for the RTOS versions.

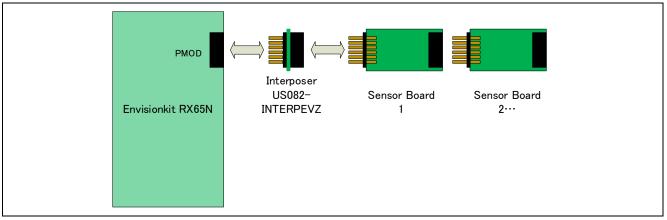


Figure 2-2 Hardware Connections for RX Family

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

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

ltem	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	PMOD Daughter Card for FS3000 flow sensor (US082-
	FS3000EVZ)

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

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

Area	Size
ROM	1,359 bytes
RAM	92 bytes

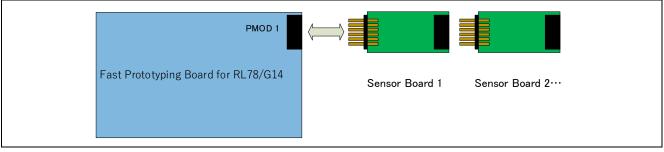


Figure 2-3 Hardware Connections for RL78/G14 Group

2.4 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.

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	e ² Studio 2023-01
environment	IAR EW for RL78 4.21.1
C compiler	C compiler package for RL78 family V1.10.00
	LLVM for RL78 10.0.0.202209
	IAR Toolchain for RL78 4.21.1.2409
Emulator	E2 Lite
Sensor board	PMOD Daughter Card for FS3000 flow sensor (US082-FS3000EVZ)

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

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

Area	Size
ROM	1,684 bytes
RAM	92 bytes

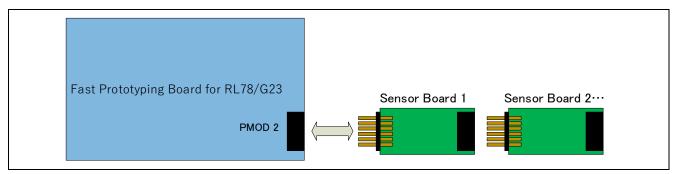


Figure 2-4 Hardware Connections for the RL78/G23 Group

3. Sensor Specifications

3.1 Overview of Sensor Specifications

The following gives an overview of the specifications of the FS3000-1005 air flow sensor.

ltem	Description	
Air velocity range	0 to 7.23 (m/sec)	
Count range	409 to 3686 (numbers counted)	
Resolution	12 bits	
Accuracy	5% (at 25°C)	
Measurement time	125 (ms)	
I2C clock frequency	100 kHz and 400 kHz are supported.	
Slave address	0x28	
Addressing mode	Only 7-bit addressing is supported.	

Table 3-1	Overview of Sensor Specifications

A value for air velocity is calculated from the value counted by the sensor according to the following curve.

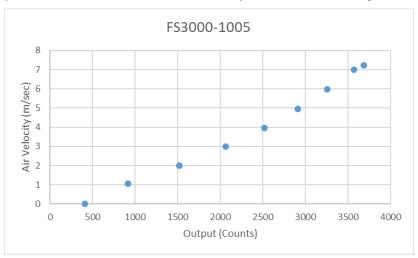


Figure 3-1 Relationship between the Values Counted by FS3000-1005 and Air Velocity

The following shows the relationship between the Air Velocity and Value Counted

Table 3-2	Relationship between the Air Velocity and Value Counted
-----------	---

Air Velocity	Output
(m/sec)	(Count)
0	409
1.07	915
2.01	1522
3.00	2066
3.97	2523
4.96	2908
5.98	3256
6.99	3572
7.23	3686



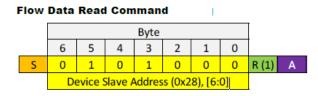
3.2 Sensor Functions

The FS3000 sample software supports the FS3000-1005 air flow sensor; it does not support the FS3000-1015 sensor.

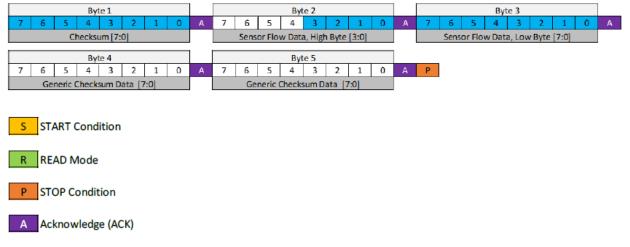
The sensor begins measurement as soon as the power supply is turned on.

To obtain data from the sensor, send the command shown below and then read the five bytes of data.

The air velocity is represented by 12 bits; the four lower-order bits of the second byte (Byte 2) are valid.



Flow Data from FS3000





Whether the value read is valid can be checked by calculating the checksum data as follows.

Example:

Byte 1: 0xCC (Checksum)

Byte 2: 0x01 (Flow data, high byte)

Byte 3: 0x99 (Flow data, low byte)

Byte 4: 0x01 (General checksum)

Byte 5: 0x99 (General checksum)

Add all values except the Checksum value.

Sum = 0x01 + 0x99 + 0x01 + 0x99 = 0x134

Add the Checksum value to the above Sum. When the eight lower-order bits of the result are 0x00, the value read is valid.

Checksum + Sum = $0xCC + 0x134 = 0x200 \leftarrow$ The eight lower-order bits are 0x00; the value read is valid.



4. Sample Software Specifications

This sample software package contains a total of six projects: non-OS and OS (FreeRTOS and Azure RTOS) versions for the RA family, non-OS and OS (FreeRTOS) versions for the RX family, and a non-OS version for the RL78 family. This section describes these projects.

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

4.1 Configuration of the Sample Software

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

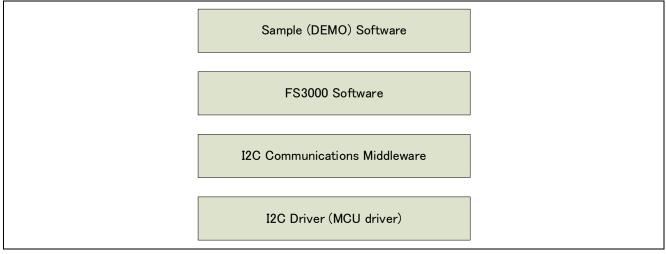


Figure 4-1 Block Diagram of the Sample Software

4.2 Specifications of Sensor API Functions

4.2.1 List of Sensor API functions

The following table lists the sensor API functions. For details of the API functions, refer to the separately provided RX Family FS3000 Sensor API FIT Module application note (R01AN5894) and RL78 Family Renesas Sensor Control Modules application note (R01AN5896)

Function	Description
RM_FS3000_Open	Starts control of the sensor.
RM_FS3000_Close	Terminates control of the sensor.
RM_FS3000_Read	Acquires data from the sensor.
RM_FS3000_DataCalculate	Calculates values from the data acquired from the sensor.

Table 4-1 List of Sensor API Functions
--

4.2.2 Guide to Using the API Functions

The following diagram of API function transitions shows the conditions on the usage of the individual FS3000 API functions and the expected orders of function calls.

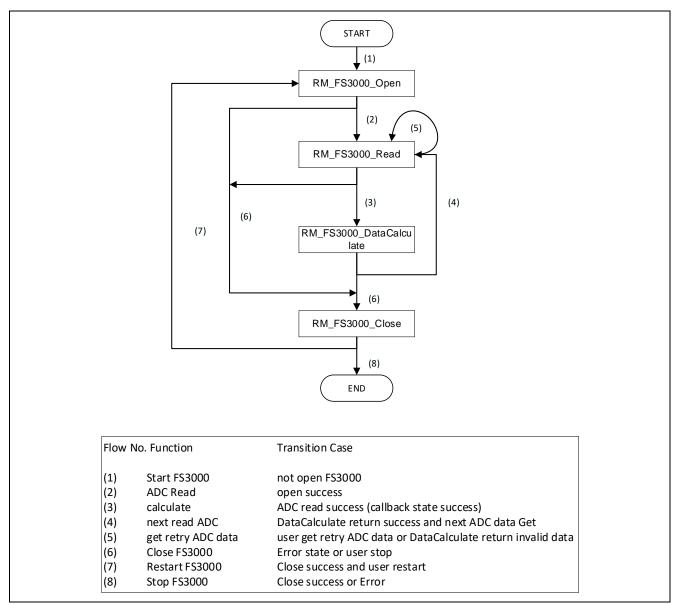


Figure 4-2 Diagram of Transitions between API Function Calls

The conditions for calling the individual functions are shown below.

• RM_FS3000_Open:	(1) Activation of FS3000 or (7) restart after a call of RM_FS3000_Close
RM_FS3000_Close:	(6) Successful completion or abnormal end of individual processing
• RM_FS3000_Read:	(2) Acquisition of measured data after the start of measurement or(5) retry after waiting for the response to the data acquisition request
RM_FS3000_DataCalculate:	(3) Calculation of data after a call of RM_FS3000_Read
lote.	

Note:

When using an OS and controlling the sensor with multiple threads or tasks simultaneously in use, the user will need to use a semaphore to control the bus. For the timing of the semaphore being raised and the control of blocking, refer to 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 starts the driver and then repeats the processing for acquiring data from the sensor and calculating values from the results of measurement.

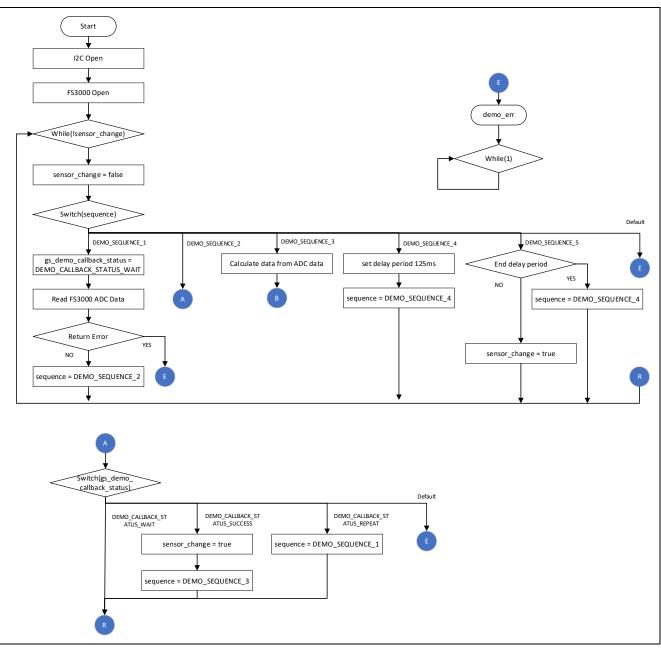


Figure 4-3 Flowchart of the Main Processing in the Non-OS Version of the Sample Software (1)

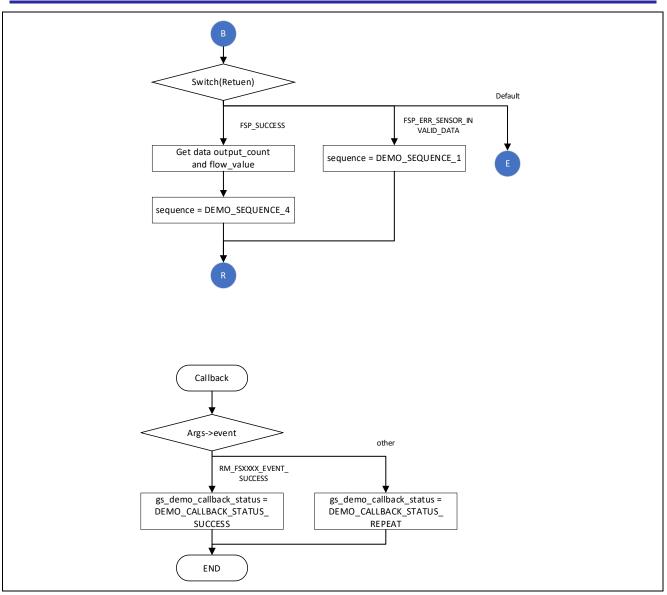


Figure 4-4 Flowchart of the Main Processing in the Non-OS Version of the Sample Software (2)



4.4 Flowchart of the OS Version of the Sample Software

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

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

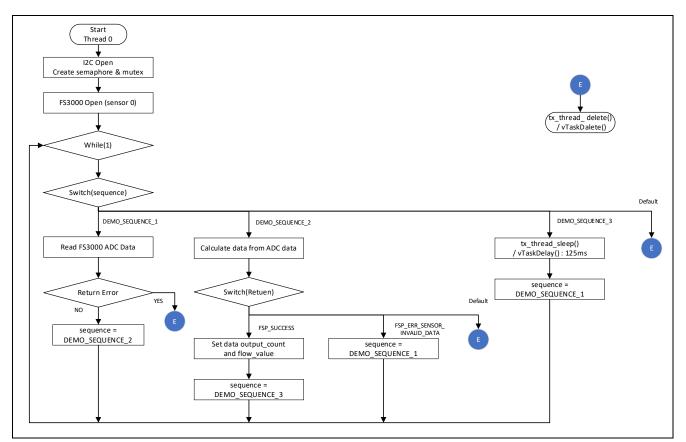


Figure 4-5 Flowchart of the Main Processing in the OS Version of the Sample Software (1)



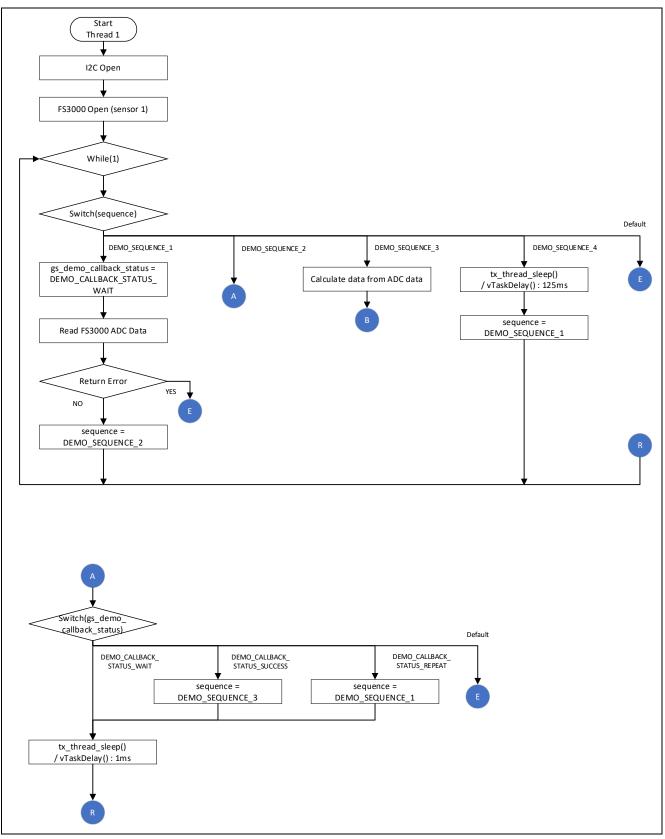


Figure 4-6 Flowchart of the Main Processing in the OS Version of the Sample Software (2)



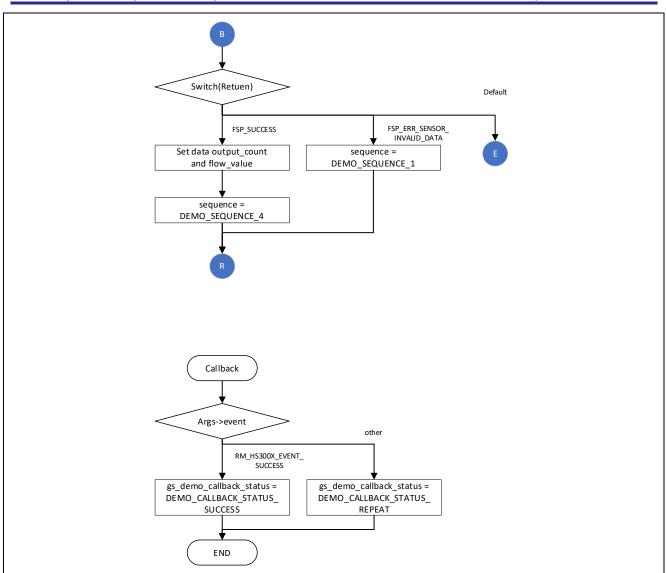


Figure 4-7 Flowchart of the Main Processing in the OS Version of the Sample Software (3)

4.4.1 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 FS3000 Air Velocity Sensor Settings

5.1.1 RA Family

Select the rm_fs3000 stack in the "Stack" tabbed page of the FSP Configurator, and the configurable items are shown in the "Properties" tabbed page.

The following items and values can be specified.

Configurable Item	Value	Description
Common		
Parameter Checking	Default (BSP)	Enable or disable the parameter check
	Enabled	processing.
	Disabled	When "Enabled" is selected, the project is built so that the generated code includes the parameter check processing.
Device type	FS3000-1005	Specify the type of device to be acquired from the sensor. "FS3000-1005" only can be selected.
Module g_fs3000_sensor FS3000 on rm_fs3000		
Name	g_fs3000_sensor0	Specify the name of the module. A module name conforming to the C language standard can be specified.
Callback	fs3000_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.

Table 5-1	FS3000 Settings for RA Family



5.1.2 RX Family

Select the r_fs3000_rx component in the "Component" tabbed page of the Smart Configurator, and the configurable items are shown in the "Configure" panel.

The following items and values can be specified.

Configurable Item	Value	Description
Configurations		
Parameter Checking	System Default	Enable or disable the parameter check
	Enabled	processing.
	Disabled	When "Enabled" is selected, the project is built so that the generated code includes the parameter check processing.
Number of FS3000 sensors	1	Specify the number of FS3000 sensors to be
	2	connected.
Device type of FS3000 Sensors	FS3000-1005	Specify the type of sensor. "FS3000-1005" can be selected.
Using communication line number for FS3000 sensor device x (x = 0 or 1)	I2C Communication Device(y) (y = 0 - 15)	Specify the communication line to be used by the sensor.
Callback function for FS3000 sensor device x (x = 0 or 1)	fs3000_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 FS3000 Settings for RX Family



5.1.3 RL78 Family

Settings can be modified by changing the values of the constants defined in the ¥r_config¥r_fs3000_rl_config.h file in the project tree of the sample project.

The following items and values can be specified.

Table 5-5 T 55000 Settings for ICE76 T anniy		
Constant Name	Value	Description
Configurations		
RM_FS3000_CFG_PARAM _CHECKING_ENABLE	0	Enable (1) or disable (0) the parameter check processing. When "1" is specified, the project is built
	1	so that the generated code includes the parameter check processing.
RM_FS3000_CFG_DEVICE _NUM_MAX	1	Specify the number of FS3000 sensors to
	2	be connected.
RM_FS3000_CFG_DEVICE _TYPE	RM_FS3000_DEVICE_TYPE _1005	Specify the type of sensor. "FS3000-1005" can be selected.
RM_FS3000_CFG_DEVICE x_COMMS_INSTANCE (x = 0 or 1)	g_comms_i2c_device0	Specify the instance name of the communication line to be used.
RM_FS3000_CFG_DEVICE x_CALLBACK (x = 0 or 1)	fs3000_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.

Table 5-3	FS3000 Settings for RL78 Family

5.2 Communication Driver Middleware Settings

5.2.1 RA Family

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

The following items and values can be specified.

Table 5-4 Communication Driver Settings for RA Family		
Configurable Item	Value	Description
Common		
Parameter Checking	Default (BSP)	Enable or disable the parameter check processing.
	Enabled	When "Enabled" is selected, the project is built
	Disabled	so that the generated code includes the parameter check processing.
Module g_comm rm_comms_i2c	ns_i2c_device I2C Comm	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	0x28	Specify the slave address. When rm_fs3000 is used, this value is automatically specified and cannot be modified.
Address Mode	7-Bit	Specify the number of slave address bits. When rm_fs3000 is used, this value is automatically specified and cannot be modified.
Callback	rm_fs3000_callback	Specify the name of the user callback function. When rm_fs3000 is used, this value is automatically specified and cannot be modified.
Module g_comms_i2c_bus0 I2C Shared Bus on rm_comms_i2c		
Name	g_comms_i2c_bus0	Specify the name of the I2C module.
Bus Timeout	0xFFFFFFF	Specify the time of I2C bus timeout.
Semaphore for	Unuse	For an RTOS project, enable or disable the
blocking	Use	blocking processing.
Recursive Mutex for	Unuse	For an RTOS project, enable or disable the
Bus	Use	recursive operation when blocking is enabled.

Table 5-4 Communication Driver Settings for RA Family

5.2.2 RX Family

Select the r_comms_i2c_rx component in the "Component" tabbed page of the Smart Configurator, and the configurable items are 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)	0x28	Specify the slave address of the device to be connected to the communications bus. When using r_fs3000_rx, specify 0x28.	
Slave address mode for communication device(x) (x = 0 - 15)	7 bit address mode	Specify the slave address mode. When using r_ fs3000_rx, specify the 7-bit address mode.	
Callback function for Communication device(x) (x = 0 - 15)	comms_i2c_user_callbac k(x) (x = 0 - 15)	Specify the name of the user callback function. When using r_ fs3000_rx, specify rm_ fs3000_callback(y) (y = 0).	

Table 5-5 Communication Driver Settings for RX Family



5.2.3 RL78 Family

Settings can be modified by changing the values of the 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.

Value	Description			
Configurations				
0	Enable (1) or disable (0) the parameter check processing. When "1" is selected, the project is			
1	built so that the generated code includes the parameter check processing.			
1	Specify the number of communication bus lines that can be			
2	connected.			
3				
4				
5				
COMMS_DRIVER_I2C	Specify the I2C type to be used for			
COMMS_DRIVER_SAU_I2C	the communication bus.			
g_comms_i2c_bus(x)_extended_cf g (x = 0 - 4)	Specify the I2C channel number to be used for the communication bus.			
0x28	Specify the slave address of the device to be connected to the communication bus. When using rm_fs3000, specify 0x48.			
COMMS_I2C_ADDR_7BIT	Specify the slave address mode. The RL78 code generator only supports the 7-bit address mode.			
rm_fs3000_callback0	Specify the name of the user callback function. When "NULL" is specified, no callback function is used.			
	0 1 1 1 2 3 4 5 COMMS_DRIVER_I2C COMMS_DRIVER_SAU_I2C g_comms_i2C_bus(x)_extended_cf g (x = 0 - 4) 0x28 COMMS_I2C_ADDR_7BIT			

Table 5-6	Communication Driver Settings for RL78 Family
Table J-0	

5.3 I2C Driver Settings

5.3.1 RA Family

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

The following items and values can be specified.

Configurable Item	Value	Description
Common	'	
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 Transmission and	Enabled	Specify whether to use the DTC for transmission and reception.
Reception	Disabled	
10-bit slave	Enabled	Specify whether to support 10-bit addressing for
addressing	Disabled	the slave address. When using rm_fs3000, select "Disabled".
Module g_i2c_m	naster0 I2C Master Drive	
Name	g_i2c_master0	Specify the name of the module.
Channel	0	Specify the channel number to be used.
Rate	Standard	Specify the baud rate.
	Fast-mode	When using rm_fs3000, select "Standard" or "Fast-mode".
	Fast-mode plus	
Rise Time (ns)	120	Specify the SCL rise time according to the specifications of the target board to be used.
Fall Time (ns)	120	Specify the SCL fall time according to the specifications of the target board to be used.
Duty Cycle (%)	50	Specify the SCL duty cycle.
Slave Address	0x00	This item specifies the slave address of the device to be connected but the user does not need to make this setting because rm_comms_i2c overwrites it.
Address Mode	7-Bit	This item specifies the salve address mode for
	10-Bit	the device to be connected but the user does not need to make this setting because rm_comms_i2c overwrites it.
Timeout Mode	Short Mode	Specify the time of I2C bus timeout.
	Long Mode	
Callback	rm_comms_i2c_callback	The name of the user callback function is automatically specified by rm_comms_i2c.

Table 5-7	r_iic_master Settings for RA Family



Interrupt Priority Level	Priority 0 (highest)Priority 1Priority 2Priority 3Priority 4Priority 5Priority 6Priority 7Priority 8Priority 9Priority 10Priority 11Priority 12Priority 13Priority 15	Specify the interrupt priority level of the I2C bus driver.
Pins		
SDA	Pxxx	The pin numbers to be used by the driver are
SCL	Рххх	displayed Use the "Pins" tabbed page to modify the pin configuration.



Configurable Item	Value	Description
Common		
Parameter Checking	Default (BSP)	Enable or disable the parameter check processing.
	Enabled	When "Enabled" is selected, the project is built
	Disabled	so that the generated code includes the
		parameter check processing.
DTC on Transmission and	Enabled	Specify whether to use the DTC for transmission and reception.
Reception	Disabled	
10-bit slave	Enabled	Specify whether to support 10-bit addressing for
addressing	Disabled	the slave address.
		When using rm_fs3000, select "Disabled".
	I2C Master Driver on r_s	
Name	g_i2c0	Specify the name of the module.
Channel	0	For an RTOS project, specify the time of
Slave Address	0.00	semaphore timeout.
Slave Address	0x00	This item specifies the slave address of the device to be connected but the user does not
		need to make this setting because
		rm_comms_i2c overwrites it.
Address Mode	7-Bit	This item specifies the salve address mode for
	10-bit	the device to be connected but the user does not need to make this setting because
		rm_comms_i2c overwrites it.
Rate	Standard	Specify the baud rate.
	Fast-mode	Select "Standard" or "Fast-mode".
	Fast-mode plus	_
SDA Output Delay	300	Specify the SDA output delay time.
(nano seconds)		
Noise filter setting	Use clock signal divided by 1	Specify the noise filter to be used for input
	with noise filter	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	
Dit Doto Modulation	with noise filter	Enable or disable the bit rate modulation
Bit Rate Modulation	Enable Disable	Enable or disable the bit rate modulation function.
Callback	rm_comms_i2c_callback	The name of the user callback function is
		automatically specified by rm_comms_i2c.

Table 5-8	r_sci_i2c Settings for RA Family
-----------	----------------------------------



Interrupt Priority	Priority 0 (highest)	Specify the interrupt priority level of the I2C bus
Level	Priority 1	driver.
	Priority 2	
	Priority 3	
	Priority 4	
	Priority 5	
	Priority 6	
	Priority 7	
	Priority 8	
	Priority 9	
	Priority 10	
	Priority 11	
	Priority 12	
	Priority 13	—
	Priority 14	—
	Priority 15	
RX Interrupt Priority	Priority 0 (highest)	When 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	Рххх	The pin numbers to be used by the driver are
SCL	Рххх	 displayed Use the "Pins" tabbed page to modify the pin configuration.



5.3.2 RX Family

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

The following items and values can be specified.

O and an and the liter	Table 5-9 r_riic_rx Settings for RX Family		
Configurable Item	Value	Description	
Configurations			
Set parameter checking enable	System Default Not Include	Enable or disable the parameter check processing. When "Include" is selected, the project is built so that the generated code includes the parameter check processing.	
MCU supported channels for CHx (x = 0 - 2)	Not supported Supported	Specify whether to support the operation of channel x.	
CHx RIIC bps(kbps) (x = $0 - 2$)	400	Specify the baud rate. Set to 400 or a smaller value.	
Digital filter for CHx $(x = 0 - 2)$	Not One IIC phi Two IIC phi Three IIC phi Four IIC phi	Specify the digital filter for input signals.	
Setting port setting processing	Not include port setting Include port setting	Specify whether to include the pin function settings in the code to be generated.	
Master arbitration lost detection function for CHx (x = 0 - 2)	Unused Used	Specify whether to use the master arbitration lost detection function.	
Address y format for CHx (x = 0 - 2, y = 0 - 2)	Not 7 bit address format 10 bit address format	This item specifies the slave address mode for slave address y but the user does not need to make this setting because rm_comms_i2c overwrites it.	
Slave Address y for CHx (x = 0 - 2, y = 0 - 2)	0x0025	This item specifies slave address y but the user does not need to make this setting because rm_comms_i2c overwrites it.	
General call address for CHx	Unused Used	Specify whether to use the general call function.	
CHx RXI INT Priority Level (x = 0 – 2)	Level 1 Level 2 Level 3 Level 4 Level 5 Level 6 Level 7 Level 8 Level 9 Level 10 Level 11 Level 12	Specify the priority level of the reception interrupt.	
	Level 13 Level 14 Level 15 (highest)		

Table 5-9 r_riic_rx Settings for RX Family



	•	
CHx RXI INT Priority	Level 1	Specify the priority level of the transmission
Level	Level 2	interrupt.
(x = 0 - 2)	Level 3	
	Level 4	
	Level 5	
	Level 6	
	Level 7	
	Level 8	
	Level 9	
	Level 10	
	Level 11	
	Level 12	
	Level 13	
	Level 14	
	Level 15 (highest)	
CHx EEI INT Priority	Level 1	Specify the priority level of the error interrupt.
Level	Level 2	
(x = 0 - 2)	Level 3	
	Level 4	
	Level 5	
	Level 6	
	Level 7	
	Level 8	
	Level 9	
	Level 10	
	Level 11	
	Level 12	
	Level 13	
	Level 14	
	Level 15 (highest)	
CHx TEI INT Priority	Level 1	Specify the priority level of the transmission end
Level	Level 2	interrupt.
(x = 0 - 2)	Level 3	
	Level 4	
	Level 5	
	Level 6	
	Level 7	
	Level 8	
	Level 9	
	Level 10	
	Level 11	
	Level 12	
	Level 13	
	Level 14	
Time and (Level 15 (highest)	
Timeout function for CHx $(x = 0 - 2)$	Unused	Specify whether to use the timeout function.
	Used	
Timeout detection time for CHx	Long mode	Specify the time for timeout detection.
(x = 0 - 2)	Short mode	
Count up during low	Unused	Specify whether to increment the count for
period of timeout	Used	detecting a timeout while SCL is at the low level.
detection for CHx	0000	
(x = 0 - 2)		
Count up during high	Unused	Specify whether to increment the count for
period of timeout detection for CHx	Used	detecting a timeout while SCL is at the high level.
(x = 0 - 2)		
		1



Set Counter of checking bus busy	1000	Specify the count to detect the bus busy state.
Resources		
SDAx Pins	Checked	Specify the pins to be used. Select the checkboxes for the desired pins.
SCLx Pins	Checked	

	Table 5-10 r_sci_iic_rx	- · ·
Configurable Item	Value	Description
Configurations		
Set parameter	System Default	Enable or disable the parameter check processing.
checking enable	Not	When "Include" is selected, the project is built so
	Include	that the generated code includes the parameter
		check processing.
MCU supported	Not supported	Specify whether to support the operation of channel
channels for CHx	Supported	Х.
(x = 0 - 12)	384000	Specify the heyd rate
SCI IIC bitrate (bps) for CHx	364000	Specify the baud rate. Set to 384000 or a smaller value.
(x = 0 - 12)		
Interrupt Priority for	Level 1	Specify the interrupt priority level.
CHx	Level 2	
(x = 0 - 12)	Level 3	
	Level 4	
	Level 5	
	Level 6	
	Level 7	
	Level 8	
	Level 9	
	Level 10	
	Level 11	
	Level 12	
	Level 13	
	Level 14	
	Level 15 (highest)	
Digital noise filter	Disable	Specify whether to use the digital noise filter.
(NFEN bit) for CHx	Enable	
(x = 0 - 12)		
Noise Filter Setting	The clock divided by 1	Specify the function of the digital noise filter.
Register (NFCS bit) for CHx	The clock divided by 2	
(x = 0 - 12)	The clock divided by 4	
, , , , , , , , , , , , , , , , , , ,	The clock divided by 8	
I2C Mode Register 1	18	Specify the number of SDA output delay cycles.
(IICDL bit) for CHx $(x = 0 - 12)$		
Software bus busy	1000	Specify the count to detect the bus busy state.
check counter		
Setting port setting	Not include port setting	Specify whether to include the pin function settings
processing	Include port setting	in the code to be generated.
Resources		
SSDAx Pins	Checked	Specify the pins to be used.
SSCLx Pins	Checked	Select the checkboxes for the desired pins.
	UNGUNGU	

Table 5-10 r_sci_iic_rx Settings for RX Family



5.3.3 RL78 Family

Select "Serial" from the peripheral functions in the Code Generator, and the configurable items are shown in 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_fs3000, select IICxx.			
	llCxx				
llCxx					
Transfer rate	1000000	Specify the baud rate.			
		When using rm_fs3000, specify 100000.			
Transfer end interrupt	High	Specify the priority level of the transfer end			
priority (INTIICxx)	Level1	interrupt.			
	Level2				
	Low				
Master transmission	Checked	Specify whether to use the call back function			
end		when master transmission ends.			
Master reception end	Checked	Specify whether to use the call back function			
		when master reception ends.			
Master error	Checked	Specify whether to use the call back function			
IICAx		when a communication error occurs.			
Transfer mode					
Transfer mode	Unused	Specify the communication function of the			
	Single master	channel to be used. Select "Single master".			
	Slave	Select Single master .			
Setting					
Clock mode setting	fCLK	Specify the clock for counting.			
	fCLK/2				
Address	16	Specify the local address.			
Operation mode	Standard	Specify the operating mode.			
setting	Fast mode/Fast mode plus				
Transfer clock (fSCL)	100000	Specify the baud 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	Checked	Specify whether to use the call back function			
end Master reception end	Checked	when master transmission ends. Specify whether to use the call back function			
master reception enu		when master reception ends.			
Master error	Checked	Specify whether to use the call back function			
		when a communication error occurs.			
Generated stop	Checked	Specify whether to generate a stop condition in			
condition in master		a callback.			
transmission/reception		Deselect the checkbox.			
end callback function					

Table 5-11 Serial Settings for RL78 Family



6. Guide for Changing the Target Device

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

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

6.1 RA Sample Project

Use the following procedures to modify a sample project.

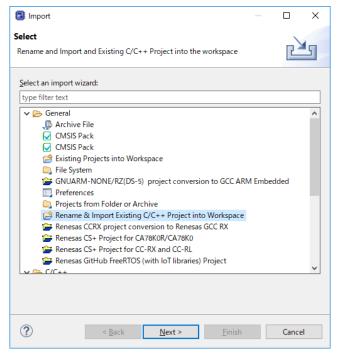
This section describes an example of modifying the sample project "FS3000_RA6M4_NonOS" so that it can be used on the EK-RA2E1 board.

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

6.1.1 Importing the Sample Project

Select [Import] from the menu.

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





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

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

→ Yutaka Oi:	¦hi≯e	2_studio > workspace	v Ö	Search workspace	م ر
ganize 👻 New folder					- 6
📙 workspace	^	Name		Date modified	Туре
.metadata		.metadata		5/28/2021 6:29 PM	File fol
FS3000_RA6M4_FreeRTOS	5	FS3000_RA6M4_FreeRTOS	5	5/28/2021 3:53 PM	File fo
FS3000_RA6M4_NonOS		FS3000_RA6M4_NonOS		5/28/2021 4:29 PM	File fo
FS3000_RL78G14_NonOS		FS3000_RL78G14_NonOS		5/25/2021 7:42 PM	File fo
FS3000 RX65N FreeRTOS		FS3000_RX65N_FreeRTOS		5/28/2021 1:02 PM	File fo
FS3000 RX65N NonOS		FS3000_RX65N_NonOS		5/25/2021 7:41 PM	File foi
	~	<		_)
Folder: FS30	00_RA6	M4_NonOS			
				Select Folder C	ancel

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

📴 Import				\times
Rename & Imp Select a director	ort Project / to search for existing Eclipse projects.			
<u>P</u> roject name:	FS3000_RA2E1_NonOS			
✓ Use <u>d</u> efault	location			
Location:	C:\Users\a5090534\e2_studio\workspace\	FS3000_R	B <u>r</u> owse	
	Create Directory for Project			
Choose file s <u>y</u> st	em: default 🖂			
Import from:				
Select root	lirectory: C:\Users\ xxxxxxx \e2_studio\works	pace\ ~	B <u>r</u> owse	
O Select <u>a</u> rchi	re file:	~	B <u>r</u> owse	
K	//4_NonOS (C:\Users\ xxxxxxx \e2_studio\work:	space\FS300	0_RA6M4_N	NonC >
Options	onfiguration output folders			
	< <u>B</u> ack <u>N</u> ext > <u>F</u> i	inish	Cance	



6.1.2 Modifying Settings of the FSP Configurator

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

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

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

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

ort Package Configuration	Generate Project Cor
	Restore Def
n	
3.0.0-rc1+20210426.9fd4d31a	Board Details Evaluation kit for RA6M4 MCU Group
	Visit https://www.renesas.com/ra/ek-ra6m4 to get kit user's manual, quick start guide
	errata, design package, example projects, etc.
EK-RA2A1	~
EK-RA2L1	
EK-RA4M1	
EK-RA4M2	
RSSK-RA6T1	
	3.0.0-rc1+20210426.9fd4d31a EK-RA6M4 Custom User Board (Any Device) EK-RA211 EK-RA211 EK-RA411 EK-RA4M1

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

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

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

*[FS3000_RA6M4_NonOS] FSP Configuration 23			- 1
Clocks Configuration			Generate Project Conten
			Restore Defaul
XTAL 20MHz		ICLK Div /1	✓ → ICLK 48MHz
LOCO 32768Hz	_	→ PCLKB Div /2	✓ → PCLKB 24MHz
MOCO 8MHz	➢ Clock Src: HOCO	✓ → PCLKD Div /1	✓ → PCLKD 48MHz
SUBCLK 32768Hz	_		
HOCO 48MHz V	_		
	CLKOUT Disabled	✓ → CLKOUT Div /1	✓ → CLKOUT 0Hz
ummary BSP Clocks 🥹 Pins Interrupts Event Links	8 Stacks Components		



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

When using a Renesas board, change the selection for "Select Pin Configuration" from "RA6M4 EK" to the target board; appropriate pins are automatically assigned.

n Configuration				Generate Project Conte
ect Pin Configuration		🔛 Export to CSV file 🛛 🛅 C	onfigure Pin Driver Warnings	
RA6M4 EK	Manage configurations	Generate data: g	bsp_pin_cfg	
in Selection	🕀 📄 🎝 Pin Configuration			🔁 Cycle Pin Group
Type filter text	Name	Value	Link	
✓ Tots > ▲ Ports > ▲ Assignation > ■ Assignation > ■ Assignation > ■ Assignation > ■ Assignation > > Memorphics > > Memorphics > > Memorphics <td></td> <td></td> <td></td> <td></td>				

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

Manage Pin Configurations		\times
Multiple Pin Configuration Management		
Modify pin configuration list or import/export external file		
RA6M4 EK (Current) R7FA6M4AF3CFB.pincfg	Add	
RA2E1 EK	Remove	:
R7FA2E1A92DFM.pincfg	Rename	
	Duplicate	e
	Merge to.	
	Import Export	
	OK	



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

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

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

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

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

[FS3000_RA2E1_NonOS] FSP Configurati	on ×				
in Configuration					O Generate Project Conten
Select Pin Configuration		Expor	t to CSV file 🛛	Configure Pir	n Driver Warnings
RA2E1 EK	✓ Manage configurations	\checkmark	Generate data:	g_bsp_pin_cf	g_2e1
Pin Selection 🗄 🕀 🏳	Pin Configuration				😲 Cycle Pin Group
Type filter text	Name	Value	Lock	Link	
	Pin Group Selection	Mixed			
> 🗸 Other Pins 🔥	Operation Mode	Simple I2C			
V V Peripherals	✓ Input/Output				
> Analog:ACMP	TXD1	None			
> 🗸 Analog:ADC > 🗸 Analog:ANALOG	RXD1	None		\Rightarrow	
> ✓ Analog:ANALOG > ✓ Connectivity:IIC	SCK1	None		\Rightarrow	
✓ Connectivity:IIC	CTS1	None		\Rightarrow	
✓ Connectivity.sci	SDA1	🗸 P401		\Rightarrow	
✓ SCI0	SCL1	✓ P402		\Rightarrow	
✓ SCI2					
SCI9					
> V Connectivity:SPI					
> Input:CTSU					
> Input:ICU					
> Input:KINT	<				>
> Monitoring:CAC	Module name: SCI1				
> System:CGC		ple I2C mode, ensure po	t pips output ty	ne is n-ch one	n drain
> 🗸 System:DEBUG		between I2C and other r			r urans
> 🗸 System:SYSTEM 🗸					



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

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

In our example, it is "g_bsp_pin_cfg_2e1".

	Manage configurations nfiguration		Generate Project Content Configure Pin Driver Warnings g_bsp_pin_cfg_2e1
RA2E1 EK \checkmark Mi Pin Selection \models $⊕$ \downarrow_Z^a Pin Con			
Pin Selection $\models \blacksquare \models \downarrow^a_{\mathbb{Z}}$ Pin Con		🗹 Generate data:	g_bsp_pin_cfg_2e1
	nfiguration		
Type filter text Name			😲 Cycle Pin Group
Import lice text Import lice text	e Valu	e Link	



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

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

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

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

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

		德 *[HS300x_RA6M4_Non(DS] FSP Configuration 😂							
Stacks (Configuration			Generati	Project Content	Stacks Configurati	on			oject Content
Threads	🐑 New Thread 🔒	🕻 Remove 📄	HAL/Common Stacks	🚯 New Stack > 🏯 Extend Stack	> 🔒 Remove		Thread 🔊 Remove 📄	HAL/Common Stacks	New Stack > 😤 Extend Stack >	Remove
49	ALCORINNE g_fJ3000_sensor g_fJ3000_sensor fJ	en m_fs300	Copport 10 Port Driver on Copport	g.	sster	🖗 g(1000,ums	ev Diject > R Semove	€ g.jopot UC Port Diveronr.jopot	for the second sec	
Summary	BSP Clocks Pins Interrup	ts Event Links St	rks Components			Summary BSP Clocks Dir	is Interrupts Event Links	Stacks Components		
	☐ Properties ☆ Problems ♀ Smart B g_i2c0 I2C Master Driver on r_sci_i2 Settings Property								28 - 1	
						Value				
	API Info	✓ Con	Common							
		F	Parameter Checking				Default (BS	P)		
			OTC on Transmi	ssion and Reception			Disabled			
		1	0-bit slave add	ressing			Disabled			
		✓ Mod	dule g_i2c0 l2C	Master Driver on r_sci_	i2c					
		1	lame				g_i2c0			
		(hannel				2 0x00 7-Bit			
		S	lave Address							
		4	ddress Mode							
		F	late				Standard			
		S	DA Output Del	ay (nano seconds)			300			
		r	loise filter setti	ng			Use clock s	ignal divided b	y 1 with noise filter	
		E	lit Rate Modula	tion			Enable			
		(allback				frm_com	ms_i2c_callbac	k	
		l.	nterrupt Priorit	y Level			Priority 2			
				ority Level [Only used v	when DT(is enabled]	Disabled			
		✓ Pins								
		S	DA				P302			
		5	CL				P301			



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

In our example, it is "g_bsp_pin_cfg_2e1".

(FS3000)	_RA2E1_NonOS] FSP Configuration ×		- 8
Stacks (Configuration		O Generate Project Content
Threads	🐑 New Thread 🔬 Remove 📄 HAL/Common Stacks	🔊 New Stack > 🚊	🖺 Extend Stack > 🛛 😰 Remove
	AL/Common g_ioport I/O New,Thread; not supported on device R7FA6M4AF3CFI g_fs3000_sensor0 FS3000 Flow Sensor (rm, fs3000_delay Timer, General PWM (r_gpt) New Object > K Remove	<pre> g_fs3000_sensor0 FS3000 Flow Sensor (rm_fs3000) g_comms_j2c_device0 l2C Communication Device (rm_comms_j2c) g_comms_j2c_bus0 l2C Shared Bus (rm_comms_j2c) g_j2c0 l2C Master (r_sci_j2c) g_j2c0 l2C Master (r_sci_j2c) </pre>	fs3000_delay Timer, General PWM (r_gpt)
- 1	BSP Clocks Pins Interrupts Event Links Stacks Components		
Properti	ies 🗙 📮 Console 📮 Renesas Debug Virtual Console 🔅 Debug	🔗 Search 🐺 Smart Manual 🔫 Progress 🏟 Smart Browser	📑 õ 🗖 🗖
g_ioport	I/O Port (r_ioport)		
Settings API Info	Property V Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 3 ELC Trigger Source Port 4 ELC Trigger Source Port B ELC Trigger Source Dath C ELC Trigger Source	Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled	^
	Port C ELC Trigger Source Port D ELC Trigger Source Port E ELC Trigger Source Pin Configuration Name V Pins TCK TDI	Disabled Disabled g_bsp_pin_cfg_2e1 P300 P110	~

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 according to the specifications of the emulator to be connected to the target board.

6.1.3 Changing toolchain setting

If you want to use a toolchain other than the GCC ARM Embedded toolchain, copy RA_FS3000.c (Non-OS) or fs3000_sensor_thread_entry.c, sensor_thread_common.c, and sensor_thread_common.c (FreeRTOS, Azure) from this project to create a new project.



6.2 RX Sample Project

Use the following procedures to modify a sample project.

This section describes an example of modifying the sample project "FS3000_RX65N_NonOS" 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	2	5
Select an import wizard:		
type filter text		
V 🗁 General		^
🚇 Archive File		
GMSIS Pack		
GMSIS Pack		
😭 Existing Projects into Workspace		
🗀 File System		
GNUARM-NONE/RZ(DS-5) project conversion to GCC ARM Embedded	ded	
Preferences		
Projects from Folder or Archive		
Rename & Import Existing C/C++ Project into Workspace		
🚔 Renesas CCRX project conversion to Renesas GCC RX		
Renesas CS+ Project for CA78K0R/CA78K0		
Renesas CS+ Project for CC-RX and CC-RL		
🞏 Renesas GitHub FreeRTOS (with IoT libraries) Project		v
	C 1	_
<u> </u>	Cancel	

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

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

→ * ↑ → e2_stud	dio → wor	kspace	~	Q	Search workspace	Q
Organize 🔻 New folder						- 0
workspace	^	Name			Date modified	Туре
.metadata		.metadata			5/28/2021 6:29 PM	File folc
FS3000_RA6M4_FreeR	TOS	FS3000_RA6M4_FreeF	TOS		5/28/2021 3:53 PM	File folc
FS3000_RA6M4_Non0	os	FS3000_RA6M4_NonOS		5/28/2021 4:29 PM		File folc
FS3000_RL78G14_Non	OS	FS3000_RL78G14_NonOS	OS		5/25/2021 7:42 PM	File fold
FS3000 RX65N FreeR	TOS	FS3000_RX65N_FreeR	RX65N_FreeRTOS		5/28/2021 1:02 PM	File fold
FS3000 RX65N NonO		FS3000_RX65N_NonC	IS		5/25/2021 7:41 PM	File fold
		<				>
Folder: F	S3000_RX6	5N_NonOS				
	-				Select Folder C	ancel



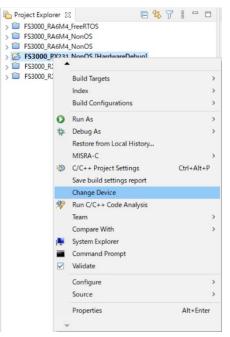
Enter the project name, select the original project for the current device, and press the [Finish] button.

📴 Import		
Rename & Import Proje Select a directory to search	e ct n for existing Eclipse projects.	
Project name: FS3000_R	X231_NonOS	
Use <u>d</u> efault location		
Location: C:¥U	sers¥a5090534¥e2_studio¥workspace¥ FS3000_F	Browse
	eate Directory for Project	
Choose file system: defa	ult 🗠	
Import from:		_
Select root directory:	C:¥Users¥xxxxxxx¥e2_studio¥workspace ~	B <u>r</u> owse
O Select <u>a</u> rchive file:	×	B <u>r</u> owse
Projects:		
FS3000_RX65N_NonO	S (C:¥Users¥xxxxxxx¥e2_studio¥workspace¥FS	3000_RX65N_Non
<		>
Options		
Keep build configurat	ion output folders	
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel



6.2.2 Changing the Device

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



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

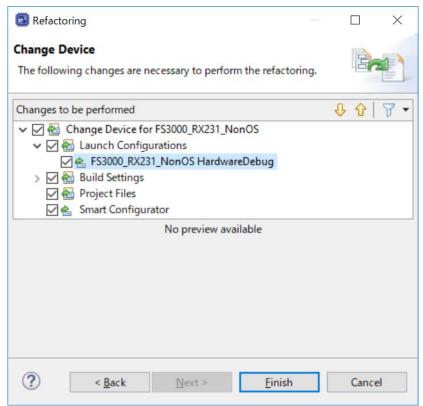
Refactoring				\times
Change Device Select the new o	e device for FS3000_RX231_NonOS		3	
Current Device:	R5F565NEDxFB			
Current Board: E	EnvisionRX65N			
Target Board:	RSKRX231			~
		Download ac	ditional bo	ards
Target Device:	R5F52318AxFP			
			Unlock Dev	vices
?	< <u>B</u> ack <u>N</u> ext >	<u>F</u> inish	Cance	el



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

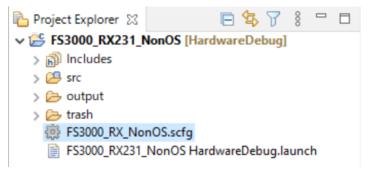
Refactoring	_		×
Change Device Review the information provided in the list below. Click 'Ne view the next item or 'Finish'.	ext >' to		
Found problems		ł	30
		is project	before
<			>
? < <u>B</u> ack <u>N</u> ext > <u>F</u> inish	1	Cance	1

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



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; the Smart Configurator window will open.



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

F\$300x_F	RX_NonOS.scfg ⊠	
evice s	election	Generate Code Generate Rep
evice sele	ection	<u>è</u> e
Board:	RSKRX231 (1.00) ~	
Device:	R5F52318AxFP	
	Download more boards	

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

FS3000_RX_NonOS.scfg 🛞		@ * F53000_RX_NonOS.scfg 12	
ocks configuration	Generate Code Generate Report	Clocks configuration	Generate Code - Generate R
	FaceF data SCol 40 (Med Syme : thick SCol 2 42 (Med 10 (Med 11 (Med 12 (Med 42 (Med	VC ac et iteration data ite Here any finance VC ac et iteration Here any finance VC ac ac Here any finance	60 and 50° million and a set of the set of t
Index and Index	CACHEAR - monte CACHEAR - monte - m	Page 100 fields	
Happengy 4 0040	CACIOX 	6492	CKCJCUK - 0,0446 CKCJCUK - 0,0460
	Line glower (fran clock (j.)PCLR) - Britel WethClocyCoLLOL - Britel EVECUSE - Britel - Britel	tepency () ()	Low prove these advances Development



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

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

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

omponents (* 🔁 🗄 🛱 🏞 🔹	Configure		0	Components 🖓 🗎 🗎 🏂 🔻	Configure		
proponents 1 Provide P	Centry Control	Weine Net supported Net supported Net supported Net supported Net supported Net supported Net supported Supported Net supported Net	* *	Components I's C II D + +	Configure Property Configure Configure State Pro- State Pro-	Value Union	

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

configuration			Generate	Code	e Generate Re
rdware Resource 🛛 🕀 📮 🖧	Pin Function	1	2		
ype filter text	type filter	text (* = any str	ing, ? = any character)	All	
TIMR2 TIMR3 Service Service Softa S		Function Function CTSa# RTSa# RXD8 SCK8 SMISO8 SMISO8 SSMISO8 SSG8 & SSC18 & SSC18 & SSC18 CTXD8	Hort assigned Not assigned	P // // // // //	in Number Not assigned Not assigned Not assigned Not assigned Not assigned 46 45 Not assigned
CMPB2 CMPB3 Capacitive touch sensing u Analog power supply					
\$@ I/O ports v	<				

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

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



Press the [Generate Code] icon to generate code.

verview information			Generate Code Generate Repo
General Information			٢
This editor allows you to modify the settings stored in config	guration file (.sc	cfg)	
Board			
Allow board and device selection			
Clocks			Application under development
Allow clock configuration			- Components
Components			Middleware Device RTOS
Allow software component selection and configuration			
			→ Pins
Pins			
Allow general pin configuration and pin configuration for sel	ected software	e component	
Interrupt Allow general interrupt configuration and interrupt configuration			
Interrupt	ation for select	ted software component	
Interrupt Allow general interrupt configuration and interrupt configura	ation for select	ted software component	
Interrupt Marwyet configuration and interrupt configuration Click here to get more information on <u>User's Manual. Release</u>	ation for select	ted software component ation Notes Tool News	
Interrupt Allow general interrupt configuration and interrupt configura Click here to get more information on <u>User's Manual Release</u> • Current Configuration Selected board/device: RSF52318AvFP (ROM size: S12 Kbytes	ation for select	ted software component ation Notes Tool News i Kbytes , Pin count: 100)	-14-
Interrupt Allow general interrupt configuration and interrupt configura Dick here to get more information on <u>User's Manual Release</u> Current Configuration	ation for select	ted software component ation Notes Tool News i Kbytes , Pin count: 100)	dī
Interrupt Allow general interrupt configuration and interrupt configura Dick here to get more information on <u>User's Manual Release</u> • Current Configuration Selected board/device: RSF52318AxFP (ROM size: 512 Kbytes Generated location (PROJECT_LOC®): <u>src¥smc_gen</u>	ation for select	ted software component ation Notes Tool News i Kbytes , Pin count: 100)	dit
Interrupt Allow general interrupt configuration and interrupt configura Click here to get more information on <u>User's Manual Release</u> Current Configuration Selected board/device: RSF52318ArFP (ROM size: S12 Kbytes Selected components: Selected components:	e Note Applica s, RAM size: 64	ted software component ation Notes Tool News Kbytes , Pin count: 100)	dit
Interrupt Allow general interrupt configuration and interrupt configura Click here to get more information on <u>User's Manual Release</u> • Current Configuration Selected board/device: RSF52318AxFP (ROM size: S12 Kbytes Generated location (PROJECI_LOC®): <u>src¥smc.gen</u> Selected components: Component	ation for select te Note Applica s , RAM size: 64 Version	ted software component ation Notes Tool News & Kbytes , Pin count: 1000 E Configuration	di
Interrupt Allow general interrupt configuration and interrupt configuration Click here to get more information on <u>User's Manual Release</u> Current Configuration Selected board/device: RSF52318ArFP (ROM size: S12 Kbytes Generated location (PROJECT_LOC®): <u>src@smc_gen</u> Selected component: Component @ Board Support Packages. (r_bsp)	ation for select <u>e Note Applica</u> s, RAM size: 64 Version 5.66	ted software component ation Notes Tool News Kbytes , Pin count: 100) E Configuration r_bsp(used)	
Interrupt Allow general interrupt configuration and interrupt configura Click here to get more information on <u>User's Manual Release</u> Current Configuration Selected board/device: RSF52318AxFP (ROM size: S12 Kbytes Senerated location (PROJECT_LOC®): <u>scr@smc.gen</u> Selected components: Component © Board Support Packages. (r_psp) © FS3000 Senort Middleware (r_p53000_nc)	ation for select e Note Applica s, RAM size: 64 Version 5.66 1.00	ted software component ation Notes Tool News Rkbytes, Pin count: 100) E Configuration r_bsp(used)	

Build the project.

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

6.2.4 Changing toolchain setting

If you want to use a toolchain other than the CC-RX toolchain, copy RA_FS3000.c (Non-OS), or main.c and fs3000_sensor_thread_entry.c (FreeRTOS), or fs3000_sensor_thread_entry.c, sensor_thread_common.c, and sensor_thread_common.c (Azure) from this project to create a new project.



6.3 RL78 Sample Project

To change the target device of RL78 family, a new project should be created.

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. Select the template "Renesas CC-RL C Executable Project" and press the [Next] button.

📴 New C/C-	++ Project				×
emplates f	or Renesas RL	78 Project			
All 2/C++	RL78	GCC for Renesas RL78 C/C++ Libra A C/C++ Library Project for Renesas RL7 GCC for Renesas RL78 Toolchain.			^
	RL78	LLVM for Renesas RL78 C/C++ Exec A C/C++ Executable Project for Renesas LLVM for Renesas RL78 Toolchain.			
	RL7B	LLVM for Renesas RL78 C/C++ Libra A C/C++ Library Project for Renesas RL7 for Renesas RL78 Toolchain.			
	RL78	Renesas CC-RL C Executable Project A C Executable Project for Renesas RL78 CCRL toolchain.	c t B <i>using tl</i>	he	
	RL78	Renesas CC-RL C Library Project A C Library Project for Renesas RL78 using toolchain	ing the C	CRL	> ~
?	< B	lack Next > Finish		Cancel	

Enter the project name (example: "FS3000_RL78G1G_NonOS") and press the [Next] button.

9		_		Х
	CC-RL Executable Project C-RL Executable Project			\$
Project name:	HS300x_RL78G1G_NonOS			
Location:	C:¥Users¥a5090534¥e2_studio¥workspace¥HS300x_RL78G1G_NonOS		B <u>r</u> owse.	
Choose file s <u>y</u> s				
Working sets	ct to working sets		Ne <u>w</u>	
W <u>o</u> rking sets	-	~	S <u>e</u> lect	
?	< <u>B</u> ack <u>N</u> ext > Einish		Cance	ł



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

0							×	0	>
Device Selection								New Devices CC DI Eventship Devices	•
You can filter devices by	regular exp	ression						New Renesas CC-RL Executable Project	V
								Select toolchain, device _debug settings	
Search Device									
Device	RAM	ROM	Pin	RTOS	Smart Co	Peripher	^	Toolchain Settings	
> RI 78 - F1F								Language:	
> RL78 - G10							- 1	Toolchain: Renesas CCRL V	
> RL78 - G11							-	Toolchain: Kenesas CCRL V	
> RL78 - G12								Toolchain Version: v1.10.00	
> RL78 - G13									
> RL78 - G13A								Manage Toolchains	
> RL78 - G14								Device Settings	Configurations
> RL78 - G1A									-
> RL78 - G1C								Target Board: Custom ~	Create Hardware Debug Configuration
> RL78 - G1D									E1/E20 (RL78)
> RL78 - G1E									E 1/ E20 (RE70)
> RL78 - G1F								Target Device: R5F11EFA	Create Debug Configuration
✓ RL78 - G1G								Unlock Devices	
> RL78 - G1G 30pin									RL78 Simulator 🗸 🗸
> RL78 - G1G 32pin								Endian: Little 🗸	L
 RL78 - G1G 44pin 								Project Type: Default	Create Release Configuration
> R5F11EF8							- 1	Project type: Detault	-
✓ R5F11EFA									
R5F11EFA	1 KB	16 KB	44		×	√			
> RL78 - G1H							- 1		
> RL78 - G1K							~		

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

6			\times
New Renesas CC-RL Executable Project Select Coding Assistant settings			\$
Use Smart Configurator Use Smart Configurator Use Peripheral Code Generator The e2 studio peripheral code generator automatically generates programs (device drivers) for MCU peripheral fittimes, serial interfaces, APD converters, DMA controllers, etc.) based on settings entered via a graphical user in Functions are provided as application programming interfaces (APD) and are not limited to initialization of peripheral code devices and and and the setting of the setting being and and the setting setting being the setting being and and the setting setting being the setting being and and the setting being bei	terface	(GUI).	
(?) < Back Next > Einish		Cance	I

Press the [Finish] button.

0						×
New Renesas CC-RL Exe Summary of project "HS30	-					Ź
TOOLCHAIN NAME : TOOLCHAIN VERSION : GENERATION FILES : generateKestarLasm generateKestinit.asm generateKestinit.asm generateKestinit.asm	Reneas CCRL v1.1000					~
?		< <u>B</u> ack	<u>N</u> ext >	Einish	Canc	el



6.3.2 Settings of the Code Generator

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

🕎 Peripheral Fur	n 🛛 🛃 Co	de Preview 📃	Device Top View 🧾 De	vice List View 🔲	Properti				
						🐻 Gene	rate Code	0	00
Pin assignment	Clock setting	Block diagram	On-chip debug setting	Confirming reset :	source	Safety function	8		^
Pin assignment s	etting								
Once the pin ass A new project mu	ist be created t	been fixed it is r to change the se Fix settings	not possible to change th ttings.	em later.					
DIOD				-					
PIOR register	Function	Port setting							
PIOR11, PIOR10	TRJIO0	P01		-					
PIOR13, PIOR12	TRJO0	P30		*					
									ł
									,
c								>	

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

💯 Peripheral Fun 🛛 🛃 Code Preview	w 👮 Device Top View 👮 De	evice List View 🔲 Prope	erties 💯 FIT Configura	- 8
			🐻 Generate Cod	e 칠 🖇
Pin assignment Clock setting Block di	agram On-chip debug setting	Confirming reset source	Bafety functions	^
-Operation mode setting				
● High-speed main mode 4.0 (V) ≤ \				
O High-speed main mode 3.6 (V) ≤ V				
O High-speed main mode 2.7 (V) ≤ V				
O Low-speed main mode 2.7 (V) ≤ V	(DD ≤ 5.5 (V)			
-Main system clock (fMAIN) setting				
 High-speed OCO (fIH) 	○ High-speed sys	tem clock (fMX)		
-High-speed OCO clock setting				
Operation	Frequency 48 (fHOCO=48, fI-	=24) 🧹 (MHz)		
-High-speed system clock setting				
Operation				
X1 oscillation (fX)	 External clock in 	nput (fEX)		
Frequency	5	(MHz)		
Stable time	2^18/fX	52428.8 (μs)		
-Low-speed oscillation clock (fIL) setting				
Frequency	15	(kHz)		
-Interval timer operation clock/Timer RJ	count source setting			
Interval timer operation clock/Timer I	RJ count sourd Stop	√ 15 (kH	2)	
-CPU and peripheral clock setting	<u></u>			
CPU and peripheral clock (fCLK)	fIH	24000 (kHz)		
<				> ×
				-



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

🕎 Peripheral Fun 🔀	🛒 Code Preview 😼 Device Top View 🧾 Device List View 🔲 Properties 🦉 FIT Configura 😑	
vum ·	🖓 Generate Code 🗕	8
Pin assignment Clock s	setting Block diagram On-chip debug setting Confirming reset source Safety functions	^
-On-chip debug operation		
 Unused 	 Used 	
-RRM function setting		
O Unused	 Used 	
-Security ID setting		
🗹 Use Security ID		
Security ID	0×000000000000000000000000000000000000	
-Security ID authentication	on failure setting	
 Do not erase flasi 	h memory data	
Erase flash memory	ory data	
		-
		v
<	>	

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

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

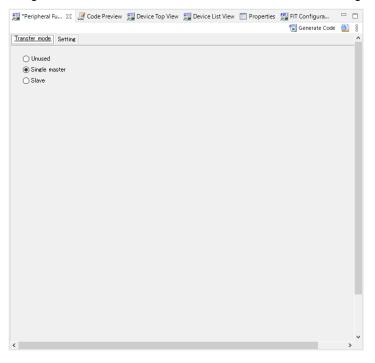
💯 *Peripheral F	u 🛙	🧾 Code P	review	Device Top View	🕎 Device List View	Properties	🕎 FIT Configura		
							🐻 Generate Code	0	8
Serial Array Uni									^
Channel UART	TO UAF	RT1 CSI00	IC00						
-Function			1						
Channel 0	IC00	~							
Channel 1	Unuse]						
Channel 2	Unuse								
Channel 3	Unuse	d v]						
<								>	. *
-									



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

💯 *Peripheral Fu 🛛 📓 Code Preview 👮 Dev	ice Top View 📓 Device List View 🔲 I		
		🐻 Generate Code	3
Serial Array Unit 0			^
Channel UART0 UART1 CSI00 IC00			
-Transfer rate setting			
Transfer rate	100000 🗸 (Б	ps) (Actual value: 100000)	
_Interrupt setting			
Transfer end interrupt priority (INTEC00)	Low 🗸		
-Callback function setting			
Master transmission end	Master reception end	Master error	
<			~ ×

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





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

💯 *Peripheral Fu 🔀 🛒 Code Previ	iew り Device Top View	🕎 Device List View	Properties	💯 FIT Configura		
				🐻 Generate Code	0	8
Transfer mode Setting						^
-Clock mode setting						
🔾 folk 🚺	ICLK/2					
-Local address setting						
Address	16					
-Operation mode setting						
 Standard 	◯ Fast mo	de/Fast mode plus	Digital filter	on		
Transfer clock (fSCL)	100000	(bps) (/	Actual value: 991	173.554)		
Interrupt setting Communication end interrupt priorit	y (INTIICA0)	~				
🖂 Master transmission end	🖂 Master reception e	nd 🔽 Mas	ter error			
-Callback function enhanced feature se		n end callback function	1			
<				_	>	1
`					,	

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	 🛙	S (Code Preview	💹 Devic	e Top View	🕎 Device l	ist View	Properties	💯 FIT Config	ura		
									🐻 Genera	te Code	0	8
General setting	Channe	0 10	Channel 1	Ohannel 2	Channel 3							^
- Functions												
Channel 0	Interva		er			~						
Channel 1	Unused					~						
Channel 2	Unused					~						
Channel 3	Unused	ł				~						
<									_		>	~
•												



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.

TAU0 TAU1 TMRD0 TMRD1 TMRG0 General setting Channel 0 Channel 2 Channel 3 Interval timer setting	^
Interval timer setting Interval value (16 bits) 100 µs ∨ (Actual value: 100) Generates INTTM00 when counting is started Interrupt setting ✓ End of timer channel 0 count, generate an interrupt (INTTM00)	
Interval value (16 bits) 100 µs v (Actual value: 100) Generates INTTM00 when counting is started Interrupt setting Find of timer channel 0 count, generate an interrupt (INTTM00)	
☐ Generates INTTM00 when counting is started Interrupt setting ☐ End of timer channel 0 count, generate an interrupt (INTTM00)	
End of timer channel 0 count, generate an interrupt (INTTM00)	
Priority Low ~	
<	×

Press the [Code Generate] 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 FS3000 */
g_fs3000_sensor0_quick_setup();
while (1U)
{
   start_fs3000_demo();
}
```

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

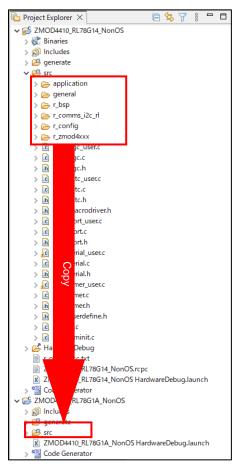
```
void g_comms_i2c_bus0_quick_setup(void)
{
    /* bus has been opened by startup process */
}
void demo_err(void)
{
    while(1)
    {
        // nothing
    }
}
```



6.3.4 Modifying Sample Source Files

Right-click on the "application" "general" "r_bsp" "r_comms_i2c_rl" "r_config" "r_fs3000_rl" folder in the project tree of the sample project "FS3000_RL78G14_NonOS" and select "Copy" from the context menu.

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





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

- COMMS_I2C_CFG_BUSx_DRIVER_TYPE
- COMMS_I2C_CFG_BUSx_DRIVER_CH

When channel 0 of the serial array unit is used:

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

```
(COMMS_DRIVER_SAU_I2C) /*
(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 5, Configuration Settings.

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

src/general/r_smc_entry.h

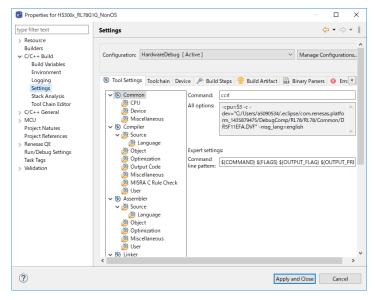
Modify "r_cg_serial.h" to "r_cg_sau.h" or "r_cg_iica.h":

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



Open the "Properties" window for the project.

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

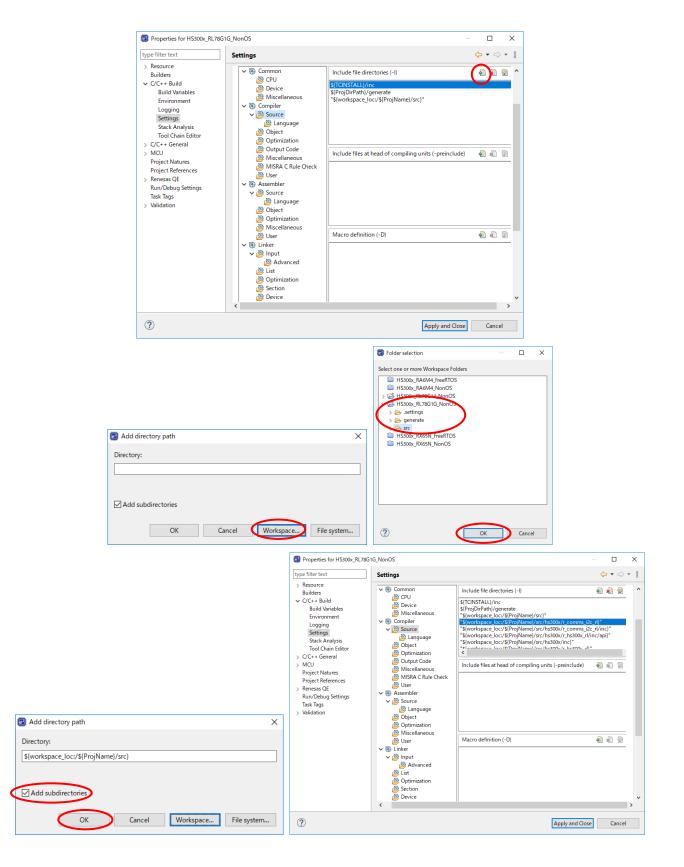




Select [Compiler] \rightarrow [Source] in the "Tool Settings" tabbed page and press the [Add] icon.

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

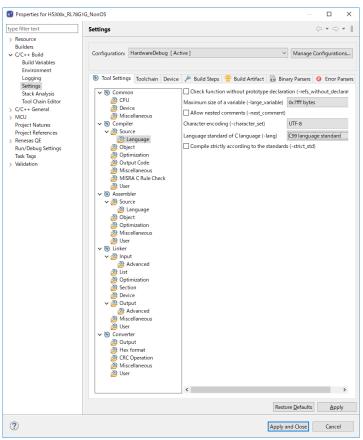
Select the checkbox for "Add subdirectories" and press the [OK] button.





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

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



Build the project.

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



7. Viewing Air Velocity Data

Use the following procedure to view air velocity data in real time.

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

12 Debug 22 🖂 🖂 👘 🖞 👘 🗖 🛄 👘 👘 👘 👘 👘 👘 👘 👘 👘 👘 👘 👘 👘	K b Debug V E RAJS2000, Barel-Hetal Debug	New Window Editor Appenence) 0 - 5 - 2 / 0 0 / }	/ x ≥ 0 = x x ⊕ x H ≡ 2 (0 <u>x</u> b + <u>0</u> + b ₀ + <u>8</u> + m	17 6 6 9 3) ·	• III • 10 (0 <mark>0</mark>)		18 Gi C/C++ (b Datag
Consor 11 Windows () Debug Pel () Headman () Head	 ▼ ■ 0.4,50000 (lises Metal Design, Tel (Renease COB Hostivate Destrogency) ▼ ■ 9.50000 (lises Metal Destroy (lises) ▼ ■ 0.5000 (lises Metal Destroy (lises) ■ 1.5000 (lises Metal Destroy (lises) ■ 1.5000 (lises)	Break Main Prangestov Paragestov Pa	Image: Section of the sectio	Source (11) (100, 100, 100, 100, 100, 100, 100, 100	Expression Type			
がなった##7 ハーサマングローン#KO-1627から2004年5月31月まま。 ハーサ2127月1ー3#KO-1627から2004日5月31月まま。		RA F5000 ExoMats 000: 51185 ターテット特許に対 - ターテット特許に対 - ターテット特許に対 - タンコード的な Option Function タブンロードので フジント特許に対 フジント 大学ンスト フジント特許に対 フジント フジント特許にの フジント フジート フジー フジート フジント フジート フジー フジー フジー フジント フジー フジー フジー フジー フジー フジー フジー フジー	Dokug Fist (Reveau DOB Hardware Dokugging) DK DK Select, writing to address 0x01004200 a Select, writing to address 0x01004200 a Select, writing to address 0x01004200 a	ugger Console 🖗 22-1-75975- 👖 Mercoy		m 26 (kg) l	<u>a ar a cus</u>	() = 9 · 3 · * • •

Click on [Add new expression] on the [Expressions] tabbed page and add "gs_fs3000_data".

🐔 🎓 🔳 🏘 Debug	 RA_FS1000_BareMetal Debug_Fit 	· • 0 5 •	2 G B = K = 2 🖋 🔊 🔊 🛷 - 📝 🖌 I	第四日 10 2 40 2 10 10 10 10 10 10 10 10 10 10 10 10 10	Q 7 4 7	14.4 日日月月日	2. 例 • 例 •	1000 Q + Q +	25
									Q : 2 10 00++ 10 Dates
tp Debug SI	B 1 1 1 1 1	R PA_F53000.c 21	(i) (NA_F50000_BareMetal) FSP Configuration 🛛 startup	ic 📝 main.e	· · · ·	(in-Varia % Bres 26 Mod	- De Proja	🕂 Dept. 23 🥐 4703	. 5 Per IOR
PALFS3000_BareMetal Debug_Flat		11	} demo_sequence_t;		~			志 海市	幸田後日後 12日
V B FA_FS3000_BareMetale#[1] [r		12	" typedef mum e demo callback status			Expression	lype	Value Addre	8
 		14			100	(+)+ qs_dems_output_count	volatile uset2.	. 0 0x2000	00
hal_entry() at hal_entry()		15	DEND_CALLBACK_STATUS_MALT = (0), DEND_CALLBACK_STATUS_SUCCESS,			(e)• gs. clema ficav	verlatile field	0 0x2000	00
main() at main.cc5 0x144		17	DEND CALLBACK STATUS REPEAT.			Add new expression			
(5.5.1) dbg-ides-enon-ma 🔓		18 19	} demo_callback_status_t;						
Renesas GDB server (Host)		20	/* See Developer Assistance in the project */						
		21 22	<pre>void g_comms_12c_bus0_quick_setup(void); void g_fs1000_sensor0_quick_setup(void);</pre>						
		23	Vold g_TsJ000_sensor0_quick_setup(vold);						
		24	<pre>void start_demo(void);</pre>						
		25	static void demo_err(void);						
		27	static volatile dano_callback_status_t gs_de						
		28 29	static volatile uist32_t gs_dc static volatile (lost gs de	mo_output_count; mo_flow;					
		345		- 1 AUR)					
		31 32	e void start_demo(void)						
		33	fsp_err_t err;						
		34	rm fixoxx raw data t raw data; rm fixoxx data t fix5000 data;						
		36	demo sequence t sequence - DEMO SEQUENCE	41					
		37							
		38 39 00000270	<pre>/* Open the Bus */ g comms i2c bus0 quick setup();</pre>						
		A8							
		41 42 0000025#	/* Open F53000 */ g f53000 sensor0 quick setup();						
		43							
		44	o while (1)						
		46 0000028e	e switch (sequence)						
		47 48	+ Case DEND SEQUENCE 1 :						
		49	Case Dero_sequence_1 :						
		Q 50	/* Clover status */	C TO ATTACK CARE THE REPORT OF A CARE					
	51 00000200	gs_demo_callback_status = 067	O_CHEERINCH_STRUES_MALT1						
			¢			x			>
			Registers () Debug Shell (5) Problems (Chapter Con	ole QA 12-b- Tott- I Menon				1 10 10 10 10 10 10	55 - 0 - 1 - "
		Debug Bat (Renewa GDB Hantware Debugging)	and approximately and a second				or say her one way on	NAME OF A COLUMN TWO	
		7-5+HR10E -	OK.						
		ダウンロート開始	Select, writing to address exeineated with data						
		Option Function	Select, writing to address exemption with data	reserverererererererererererererer					
		8050-FM7	La majori de la Composition						
			イントはアドレスのcontactまでします。 イントはアドレスのcontactまです。						
			インドはドドレス012702702702702702702702702702702702702702						
		ハートウェア・フレーラオ	イントはアトレス8427年に設定します。						
		¢	40 mm						,
SSD(Secure	- 0x0000278 Ø	172184 ns	🗑 DWT 🚓 74369	Wittable	Smart inp	st 39:1:967			1 1 1 1 1 2 1



						o	ES ROCK+
Dela 11 Control Contro	III III 12 12 13 13 14 14 15 14 16 15 17 12 18 14 19 14 10 15 11 12 12 12 13 14 14 15 15 15 16 17 17 18 18 15 19 19 10 12 11 12 12 12 13 14 14 15 15 15 16 15 17 18 18 15 19 19 10 15 11 15 12 16 13 16 14 16 15	<pre>) Mone_interment_ii</pre>	fact_word; more: LANC_TINING_WAT; x		© 95 demo colpticitant vector units of a	Lase Address Address Convector Conve	B erectantis executed ginary:0 0xtal:0
	22	¢		· · ·	4		ι
	NA FSTORD Barebler 5-19-00510E 9520-10235 Option Function	Debug Rat (Renesas GDB Harclaser Debugging)			# 2	< 2 5 M 9 8	2월 ở 9 · C · ⁻

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

Start debugging, and the values of the variables will be updated in real time.

		Q, 😰 🛱 C/C++ 💠 Dobug 🛞 FSP Configuration
Detay II	<pre>1) dem_sequence_1: ** proprie me_stem_schlam_schlam_schlam_ ** proprie me_stem_schla</pre>	Made Liphen, Chara, 21 ≠ 400, 5 Made 20 Mar
	Se 1/ Chew status 7/ Status - 2010 CALLBACT (TATUS JAIT) 22 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
	Concel 1: Theophan () Decopying () Protons () Restored Concers () X7+1-7577 () Netroy Theophan Decopying Theorem 20 Netrons Decopying () Theophan Decopying () X750	a x x (3 5 3 5 5 2



Revision History

		Description		
Rev.	Date	Page	Summary	
1.00	June 30, 2022	-	First Release	
1.01	March 3, 2023	-	Updated: environments for RL78	
1.02	March 29, 2023	-	Updated: Environments for RA, RX, RL78, RZ	
			Updated: Main Processing Flow of Sample Software	
			Updated: Guide for Changing the Target Device	
1.03	September 7,	-	Updated: Guide for Changing the Target Device	
	2023		Deleted: RE01 items	



General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

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

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

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

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

- 7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
- 8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- 13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
- 14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
- (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.