# RENESAS

## Quick Connect IoT

Quick Connect IoT is a combination of hardware and software modules that simplifies the design process of developing system solutions. It allows you to put together systems to evaluate sensors, connectivity, and the MCU in a full system environment with a minimal amount of set-up time or without writing the basic firmware structure.

Hardware modules are available with low-level drivers and middleware that allow you to immediately start writing the application layer code.

This manual reviews the Quick Connect IoT with an example that uses a specific sensor, but you are encouraged to consider other scenarios and explore all the devices available.

## Contents

1.	Refe	erence Documents	2					
2.	Harc	lware Example	2					
	2.1	MCU Selection	2					
	2.2	MCU Setup	3					
	2.3	Sensor Selection	4					
	2.4	Connectivity Selection	4					
	2.5	Complete RA Hardware Setup	5					
	2.6	Complete RX Hardware Setup	6					
3.	RAF	RA FSP						
	3.1	Start Project	7					
	3.2	Insert Middleware	7					
	3.3	Resolve User Items Related to the Sensor Stack	7					
	3.4	Generate Code	9					
	3.5	API Examples	C					
4.	RX S	Smart Configurator	D					
	4.1	Start Project	1					
	4.2	Insert Component	2					
	4.3	Resolve User Items Related to the Sensor Stack	3					
	4.4	Generate Code	4					
	4.5	API Examples	5					
5.	Add	itional Information	5					
6.	Revi	sion History	5					
Ар	pendix	κ Α	6					



## 1. Reference Documents

Document number	Title
R20UT4827EG0100 Rev 1.00	EK-RA2L1 V1 User Manual
R20UT3558EG0100	RSK-RX651/RX65N User Manual
R36UZ0002EU0100	US082-ZMOD4410EVZ Evaluation Board Manual
R36UZ0004EU0100	US082-HS3001EVZ Evaluation Board Manual
R36UZ0006EU0100	US082-INTERPEVZ Evaluation Board Manual
R01AN5892EJ0100	Renesas Sensor Control Modules Firmware Integration Technology
R01AN5893EJ0100	Renesas HS300x Sensor Control Module Firmware Integration Technology
R01AN5897EJ0100	HS300x Sample Software Manual

## 2. Hardware Example

For hardware, a typical system might consist of an MCU, sensors, and a connectivity solution (the Connect IoT part).

## 2.1 MCU Selection

The Renesas MCUs are supported by various EKs, RSK, and target boards. For Quick Connect IoT, you can use any one of a number of MCU boards. The majority of Renesas MCU boards have headers that support standard form factor add-on boards. Typically, this includes some combination of PMODTM, Arduino, Mikro CLICK, and Grove.

For this manual, the example uses the EK-RA2L1; this device has memory footprints and is chosen based on full system requirements.



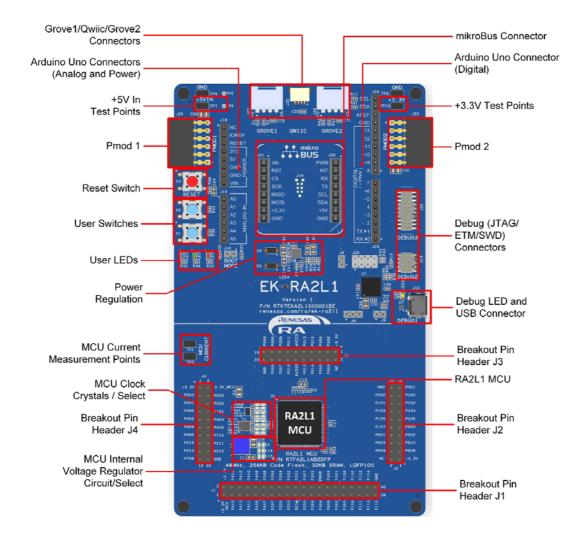


Figure 1. EK-RA2L1

## 2.2 MCU Setup

The amount of setup required depends on the interfaces and plug-in modules chosen. Some interfaces cannot be reconfigured; for example, if you use Arduino or MikroCLICK, there is no configuration required. If you choose to use PMODs, there are many PMOD pinouts available to connect to specific IO standards. Most of the MCU development boards support Digilent Type2A, extended SPI, Type3A, extended UART, Type6A, and extended I<sup>2</sup>C. For the EK-RA2L, configure a PMOD to be Type3A (UART) for our connectivity choice and another for Type6A (I<sup>2</sup>C) for the sensor connection. Reference the specific EK that you are using to reconfigure to the correct IO that is required. See the EK-RA2L1 User Manual for PMOD1 reconfiguration to Type6A. PMOD2 is the UART PMOD in this application.

**Note**: Some of the older RSKs and EK may require an interposer board available from Renesas to support Type6A. See Table 2 in the appendix for a list of boards requiring interposer.



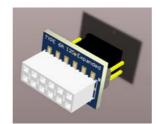


Figure 2. Type 6A Interposer

### 2.3 Sensor Selection

Renesas, non-Renesas sensors, and plug-in modules are supported and are chosen to meet the available plug-in module headers on the MCU.

Part Number	Туре
HS3001	Temperature and Humidity
ZMOD4410	Indoor Air Quality and Gas and Odor
ZMOD4510	Outdoor Air Quality

#### Table 1. Sensor Board List

**Note**: This table is a selection of parts. Renesas continually releases new sensors, peripherals, and plug-in modules. Visit the Renesas website for more information.

### 2.4 Connectivity Selection

Renesas supports numerous connectivity choices from UART (wired) to Bluetooth and Wi-Fi. This manual focuses on a common connectivity solution for an IoT example, Wi-Fi using a PMOD.



## 2.5 Complete RA Hardware Setup

In the hardware setup, plugged in is the selected PMODs and the USB cable for the debugger interface, and in Figure 3, a complete solution for a connected Air Quality Sensor system is shown.

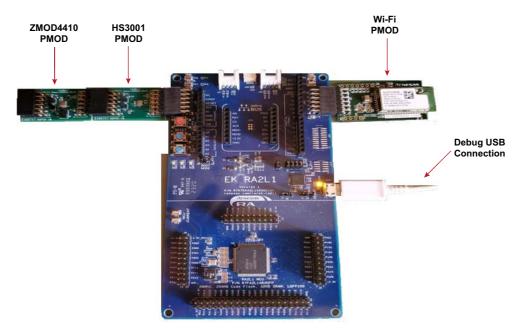


Figure 3. Air Quality System



## 2.6 Complete RX Hardware Setup

In Figure 4, the RX Sensor setup using the RX65N Envision Kit is an example of an RX hardware setup. Like the RA hardware setup, plug in the PMOD sensors that are used for the example solution and the USB debug cable. The example shown uses the RX65N Envision Kit.

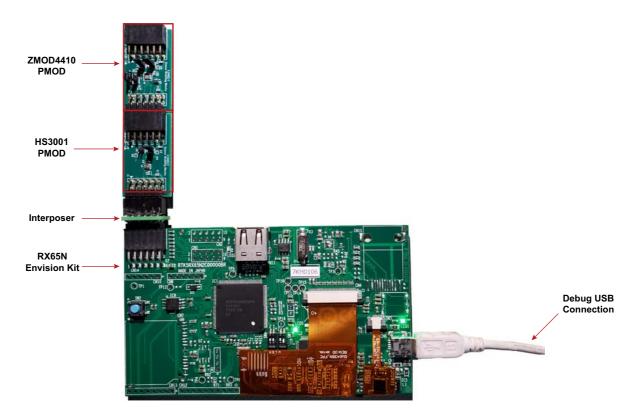


Figure 4. RX Air Quality Setup

## 3. RA FSP

The steps to write the application code are straight-forward. For this example, the steps are as follows:

- 1. Start a new RA project.
- 2. Select a BSP.
- 3. Insert Middleware Stacks.
- 4. Resolve middleware issues (remove the red out) by defining user items such as which I<sup>2</sup>C port to use.
- 5. Generate code.

At this point, you have a buildable project with limited to no debug required, so you can start to write your application code. Typically, this consists of instantiating USER buffers for the data, and then a simple POSIX such as APIs to talk to the devices with middleware instantiated.

**Note**: This is not intended as training on FSP, but rather an overview of the new middleware available to provide you with a system solution of MCUs, sensors, and connectivity.



### 3.1 Start Project

Starting a project is as simple as follows:

- 1. Select the correct project type.
- 2. Name the project.
- 3. Select the BSP.
- 4. Select the type of project (executable or Library) including RTOS support. Because this example is for non-RTOS, Bare Metal Minimal is chosen.

### 3.2 Insert Middleware

After setting up the project, you are in the FSP Configuration View. (For additional information, see the FSP manuals that are available through the SmartBrowser or help facilities of e<sup>2</sup>Studio.) The following steps are for adding a single sensor:

```
Select New Stack \rightarrow Middleware \rightarrow Sensor \rightarrow ZMODXXXon rm_zmod4xxx. (See Figure 5.)
```

[QCIoT_UserGuide] FSP Configuration X	🗖 🗖 Package 📑	🕈 MCU Package 🔀	
Stacks Configuration		Generate Project Content	🗩 🍺 🖻 Type pin funct
Threads     €     New Thread     €     Remove       ♥     ∰ HAL/Common       ♥     g_ioport I/O Port Driver on r_ioport	HAL/Common Stacks g_ioport I/O Port Driver on r_ioport 3		HS300X on rm_hs300x ZMOD4XXX on rm_zmod4xx



### 3.3 Resolve User Items Related to the Sensor Stack

When the stack is instantiated, it displays red indicating that you need to select user-configurable items. In the case of the ZMOD, you need to select the device, library type, the interface, and the source for the measurement trigger.

g_ioport I/O Port Driver on r_ioport	d∰ g_zmod4∞o	_sensor0 ZMOD	4XXX on rm_zmod4xxx	
ī)	(i)			
			•	
	Add Require	es ZMOD 🛛 👼	Add Timer Driver for	Add IRQ Driver for
	Library	New	> 🕀 ZMOD4410 IAQ	1st Generation on rm_zmod4xx
			🚽 🚸 ZMOD4410 IAQ	2nd Generation on rm_zmod4x
	L		🕂 🕀 ZMOD4410 Odo	or on rm_zmod4xxx
			A 71 100 1110 C 11	ur-based Odor on rm zmod4xx
			ZMOD4410 Sulfe	ur-based odor on nin_zinou400
				2 1st Generation on rm_zmod4

Figure 6. ZMOD4410 2<sup>nd</sup> Generation Indoor Air Quality Selection



g_ioport I/O Port Driver on r_ioport	g_zmod4xxx_sensor0 ZMOD4XXX on rm_zmoo	d4xxx
)	1	
	ZMOD4110 IAQ 2nd Generation on rm_zmod4xxx     Generation on     The second sec	er for Add IRQ Driver for measurement [optional]
	g_comms_i2c_device     0 12C     Communication     Device on	
	<b>A</b>	
	g_comms_i2c_bus0 I2C Shared Bus on rm_comms_i2c €	
	Add I2C	
	Commu > ++++++++++++++++++++++++++++++++++	r Driver on r_iic_master
	🕂 🕂 I2C Master	r Driver on r_sci_i2c

Figure 7. I<sup>2</sup>C Master on r\_iic\_master Selection

AL/Common Stacks		New St	ack > 🔮 Extend Stack > 🔊 Remo			
<ul> <li>g_ioport I/O Port</li> <li>Driver on r_ioport</li> <li>I</li> </ul>	g_zmod4xxx_sensor0 ZMOD4XXX on rm_zmod4xxx					
	ZMOD4410 IAQ 2nd Generation on rm_zmod4xxx	Add Timer Driver for measurer New	and the second			
	<b>A</b>					

#### Figure 8. Pop-Up Help on GPT Error

**Hint**: The HAL/Common Stacks blocks remain red until all the configurations items are satisfied. Hovering on the red X does pop-up the error that is in the particular block. In this case, the GPT trigger for measurement is red. See Figure 9 for an example of pop-up help.

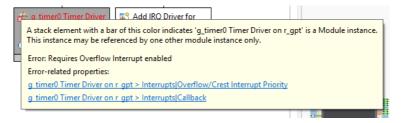


Figure 9. Pop-Up Help On GPT Error



Next, this if fixed by enabling the Overflow interrupt on the GPT. The callback function and name are defined by the middleware.

Propert	ies 🕺 🔝 Problems 👒 Smart Browser		📑 🕴 🗖
g_timer(	) Timer Driver on r_gpt		
Settings	Property	Value	^
API Info	Pin Output Support	Disabled	
	Write Protect Enable	Disabled	
	<ul> <li>Module g_timer0 Timer Driver on r_gpt</li> </ul>		
	> General		
	> Output		
	> Input		
	✓ Interrupts		
	Callback	film rm_zmod4xxx_timer_callback	
	Overflow/Crest Interrupt Priority	Priority 1	
	Capture A Interrupt Priority	Disabled	
	Capture B Interrupt Priority	Disabled	
	Trough Interrupt Priority	Disabled	
	> Extra Features		
	✓ Pins		
	GTIOCA	<unavailable></unavailable>	
	GTIOCB	<ur><li><unavailable></unavailable></li></ur>	
			¥

#### Figure 10. GPT Overflow Interrupt Setting

### 3.4 Generate Code

After all the stacks/HAL code is satisfied and all the user selections are made, you must **Generate Project Content**. See Figure 11.

Stacks Configuration	Generate Project Content				
✓ A HAL/Common	iew Thread Remove E er on r_ioport MOD4XXX on rm_zmod4xxx	HAL/Common Stacks g_ioport I/O Port Driver on r_ioport (1)	g_zmod4xxx_sensor0 ZMOD4XXX on rm_zmod4xxx	New Stack >	Extend Stack > 🙀 Remove





## 3.5 API Examples

As we indicated, the function calls are now available to you so that you can start writing your application (i.e. how you will use the temperature, humidity and air quality values in you application). The data types are all defined, so you only need to instantiate buffers for you application.

#### Example of Buffer instantiation:

```
fsp_err_t err;
rm_zmod4xxx_raw_data_t raw_data;
rm_zmod4xxx_iaq_2nd_data_t zmod4xxx_data;
```

#### ZMOD Public API:

```
fsp_err_t RM_ZMOD4XXX_Open(rm_zmod4xxx_ctrl_t * const p_api_ctrl, rm_zmod4xxx_cfg_t
const * const p_cfg);
fsp_err_t RM_ZMOD4XXX_MeasurementStart(rm_zmod4xxx_ctrl_t * const p_api_ctrl);
fsp_err_t RM_ZMOD4XXX_MeasurementStop(rm_zmod4xxx_ctrl_t * const p_api_ctrl);
fsp_err_t RM_ZMOD4XXX_StatusCheck(rm_zmod4xxx_ctrl_t * const p_api_ctrl);
fsp_err_t RM_ZMOD4XXX_Read(rm_zmod4xxx_ctrl_t * const p_api_ctrl);
fsp_err_t RM_ZMOD4xxx_raw_data_t * const p_raw_data);
```

**Note**: Since the middleware will support multiple instances of ZMOD sensors on multiple I<sup>2</sup>C buses, these will typically be abstracted one layer to account for multiple device instances and multiple configurations. In this case, your calls may be abstracted by a function table in the configuration instance. For details, see to the FSP manuals.

For this case shown, it is defined by the ctrl instance for Sensor 0:

```
rm_zmod4xxx_instance_ctrl_t g_zmod4xxx_sensor0_ctrl;
```

#### ZMOD API Examples Device 0 defined by:

```
err = g_zmod4xxx_sensor0.p_api->measurementStart(g_zmod4xxx_sensor0.p_ctrl)
```

## 4. RX Smart Configurator

The steps to follow so that you can write the application code are straight-forward. For this example, the steps are as follows:

- 1. Start a new RX project.
- 2. Select the BSP and device.
- 3. Insert components.
- 4. Resolve component configuration issues by defining user items such as which I<sup>2</sup>C port to use.
- 5. Generate Project Code.

At this point we will have buildable project with limited to no debugging required. The user can then start to write their application code. Typically, this consists of instantiating USER buffers for the data, and then a simple POSIX such as APIs to talk to the to devices with middleware instantiated.

**Note**: This is not intended to be training on the RX Smart Configurator, but rather an overview of the new middleware available to get you to a system solution of MCUs, sensors, and connectivity.



## 4.1 Start Project

Starting a project is as simple as follows:

- 1. Select the correct project type.
- 2. Name the project.
- 3. Select the BSP.
- 4. Select the type of project (executable or Library) including RTOS support. Because this example is for non-RTOS, Bare Metal Minimal is chosen.

After generating the RX C/C++ executable project, it opens in the RX Configuration View as shown in Figure 12.

<ul> <li>RC_2MOD_Dr = RC_Temp_Humidity/RC_Temp_Humidity:sclg = e<sup>2</sup> stu- File_Edit Navigate Search Project Renease Views Run Wind</li> </ul>	ow Help						- ø ×
🐔 🕸 🔳 the Debug 🗸 🗉 RK_Temp_Hum	dty Debug VQ D•	010-21	🖬 🚅 New Connection 👻 🕅	14 〒+ 0 0+#000氏の名(4	0.350.4.0		
	-						😰 🗟 C/C++ 🧟 Smart Configurator
Project Explorer 13     Poject Explorer 13     Poject Explorer 13     Poject Explorer 13	RX_Temp_Humidity.scfg 11					MCU Package II 📗 MMU	
> 💋 Includes	Overview information					<b>G A F F B</b>	E Tarr portion ( Assig. )
3 25 src RX_Temp_Humidity Debug.launch	• General Information (2) *						
RX, Temp, Humidity HandwareDebugJaunch	This editor allows you to modify the Board	settings stored in c	onfiguration file (.scfg)				
	Allow board and device selection						
	Cocks Allow closeling stations Components Allow software component selection and configuration Pro Moles general pin configuration and pin configuration for selected software component Interrupt Allow general interrupt configuration and interrupt configuration for selected software component Cock have to get more information on (part) Manual Palesan False Association Hater Teat News Cock for the to get more information on (part) Manual Palesan False Association Hater Teat News Cock for the to get more information on (part) Manual Palesan False Association Hater Teat News Cock for the to get more information on (part) Manual Palesan False Association Hater Teat News Cock for the to get more information (Manual Palesan False Association Hater Teat News Cock for the to get more information Cock for the to get more information (Manual Palesan False Association Hater Teat News Cock for the to get more information Cock for the t					Renesas	
	Selected components: Component Configuration						1.1
	© r_bsp	Version 5.50	Configuration r_http:/used)				
	Overview Board Clocks Componen	ta Pina Interneta			~	+ Legend	
Console 11				Configuration Problems		Lossifier	1. v = n
Smart Configurator Output				0 items			4 0
NOSGONIZI File generatedisrc/sec_gen/e_pincfg/bfin.c NOSGOND2: File generatedisrc/sec_gen/generativ_sec_im NOSGOND2: File generatedisrc/sec_gen/generativ_sec_im NOSGOND2: File generatedisrc/sec_gen/confg/v_bp_iv NOSGOND2: File generatedisrc/sec_gen/confg/v_bp_iv	terrupt_config.h		^	Description		Type	
M03000004: File modified: <u>src\smc gen\r config\r hsp cor</u>	fight						Installing Smart Manual
8							
			0 items selected			http://s	Change Smart Manual Installation Preferences

Figure 12. RX Smart Configuration View (Empty Project)



## 4.2 Insert Component

After setting up the project, you are in the FSP Configuration View. The following example adds a single sensor, the HS3001 Humidity and Temperature sensor. For additional information, reference the Smart Configuration manuals that are available through the SmartBrowser or help facilities of e<sup>2</sup>Studio.

Select Componen	ts Tab $\rightarrow$ r_hs3001	$rx \rightarrow Finish.$	(See Figure	ə 13.)

e <sup>2</sup> New C	omponent						×	
	Component nponent from		e in list					
Туре	Type All							
Function	All						~	
Filter								
Compor	nents		Туре		Version		^	
+ r_flas			FIT		4.60			
r_flas			FIT		3.01			
🖶 r_gpi			FIT		3.70			
🖶 r_hs3	001_rx		FIT		0.50			
🖶 r_irq_	nx		FIT		3.60			
r_iwdt_rx			FIT		3.60			
🖶 r_jpe	gd_rx		FIT		2.06			
🖶 r_jpe	ge_rx		FIT		1.01			
🖶 r_lon	gq		FIT		1.82			
			FIT		2.00	8	>	
Descriptio Depende Depende	ency : r_bsp ve ency : r_drvif_ii	rsion(s) 5.62 ic version(s) 0.	50 Renesas Electro	onics HS300x s	ensors API		~ ~	
Download	d more softwa	re component	<u>5</u>					
<u>Configure</u>	e general settir	<u>ngs</u>						
?		< Back	Next >	Finis	h	Can	cel	

Figure 13. RX Component Selection



### 4.3 Resolve User Items Related to the Sensor Stack

After the component is instantiated, it scans for dependencies and inserts the required I<sup>2</sup>C driver to support the device. Your component list reflects this change. See Figure 14.

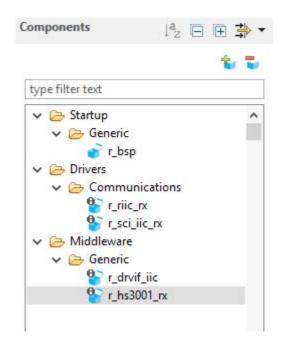


Figure 14. RX Component List

Next, the user choices require configuring. Select the number of  $I^2C$  channels that are supported in the driver. Select **r\_drvif\_iic** in the component list followed by selecting the number of IIC communications lines. In this case, select 1 and choose **RX FIT IIC**. See Figure 15.

oftware component configuration			
mponents 🏻 🎘 🗄 🕂	Configure		
ype filter text	Property V 🍪 Configurations	Value	
<ul> <li>&gt; Startup</li> <li>&gt; Generic</li> <li>~ [&gt; Drivers</li> <li>&gt; Communications</li> <li><sup>0</sup> r_riic_rx     <li><sup>0</sup> r_ciic[rx     <li>&gt; Middleware     <li>&gt; Generic     <li><sup>0</sup> r_drvif_iic     <li><sup>0</sup> r_drvif_ix     </li> </li></li></li></li></li></ul>	# Parameter Checking	System Default	
	# Number of IIC communication lines	1	
	# IIC Driver Type for Comm0	RX FIT RIIC	
	# Channel No. for Comm0	0	
	# IIC Driver Type for Comm1	Not selected	
	# Channel No. for Comm1	0	
	# IIC Driver Type for Comm2	Not selected	
	# Channel No. for Comm2	0	
	# IIC Driver Type for Comm3	Not selected	
	# Channel No. for Comm3	0	
	# IIC Driver Type for Comm4	Not selected	
	# Channel No. for Comm4	0	

Figure 15. r\_drvif\_iic configuration

Finally, the actual pin connection is chosen. Go to the **Pins Configuration** tab, select **RIIC0** in the list, and the two used pins display. Use the **Pin Number** pulls downs, and choose the pins connected in your design. The used pins appear red. See Figure 16.

X_Temp_Humidity.scfg 😥					- 0	
configuration					5	Type pin function Assig
rdware Resource 🛛 🗉 🛱 🖧	Pin Functio	in			2 🗉 🖬 🔤 🕹	
fype filter text	type filter	text (* = any str	ing, ? = any character)		All 🗸	P22 VVS5 VVC P70 P73 P73 P73 P73
All     Digital power supply     Clock generator     Clock frequency accuracy measuremer     Operating mode control	Enabled	Function SCL SDA	Assignment PB1/MTIOC0C/MTIOC0C#/GTADSM1/TMCI0/RXD6/ PB2/MTIOC08/MTIOC08#/GTADSM0/TMRI0/TXD6/	Direction IO IO	Remarks	2 다 다 유 유 유 가 가 다 유 유 가 가 다 다 다 다 다 다 다 다
System control On-chip emulator Interrupt controller unit Multi-function timer pulse unit 3						10 992 19 993 19 994 19 994 19 995
General PW/M timer     G. P-bit timer     G. P-ot output enable 3     Serial communications interface     El 2C bus interface						26 VS5 27 P96 24 VCC
RICO     CAN module						RX66T 22 P80 22 P81 SCL
CANO Serial peripheral interface RSPI0						R5F566TAAxFM 21 282 SDA 20 983 19 994
<ul> <li>4 12-bit A/D converter</li> <li> <ul> <li>S12AD0</li> <li>S12AD1</li> <li>S12AD2</li> </ul> </li> </ul>						18 P85
12-bit D/A converter     Comparator C     CMPC0						R25         1           Part         7           V45         8           Part         9           Part         1           Part         1
CMPC1 CMPC2	۲.				>	
unction Pin Number						

Figure 16. I<sup>2</sup>C Pin selection on RX

**Hint**: Configuration problems are show in the **Configuration Problems** tab. In default layout, this is in the lower right of screen.

### 4.4 Generate Code

After all the component settings are satisfied and there are 0 items in the Configuration problems, you simply generate code with the feature, **Generate Project Content**. (See Figure 17.)

RX_Temp_Humidity.scfg 🔀			
oftware component configur	ation		ũ
mponents	Configure		
٤	Property	Value	
type filter text		System Default	
	# Number of IIC communication lines	1	
	# IIC Driver Type for Comm0	RX FIT RIIC	
	# Channel No. for Comm0	0	
	# IIC Driver Type for Comm1	Not selected	
	# Channel No. for Comm1	0	
	# IIC Driver Type for Comm2	Not selected	
	# Channel No. for Comm2	0	
	# IIC Driver Type for Comm3	Not selected	
	# Channel No. for Comm3	0	
😜 r_drvif_iic	# IIC Driver Type for Comm4	Not selected	
🍄 r_hs3001_rx	# Channel No. for Comm4	0	

Figure 17. Generate Project Code



### 4.5 API Examples

As indicated, the function calls are available to you so you can start writing your application (for example, how you use the temperature and humidity values in your application). The data types are all defined, so you only need to instantiate buffers for you application.

#### Example of Buffer instantiation:

```
e_hs3001_err err;
```

#### HS3001 Public API:

```
hs3001_err_t R_HS3001_Open (hs3001_ctrl_t *const p_ctrl, hs3001_cfg_t *const p_cfg);
```

hs3001\_err\_t R\_HS3001\_Read (hs3001\_ctrl\_t \*const p\_ctrl, uint8\_t \*p\_dest, uint32\_t
bytes, uint16\_t datatype);

hs3001\_err\_t R\_HS3001\_IOCtl (hs3001\_ctrl\_t \*const p\_ctrl, uint8\_t \*p\_buf, uint32\_t
bytes, uint16\_t command);

hs3001\_err\_t R\_HS3001\_Close (hs3001\_ctrl\_t \*const p\_ctrl);

hs3001\_err\_t R\_HS3001\_GetVersion (hs3001\_version\_t \* p\_version);

```
void r_hs3001_callback(drvif_iic_event_t event, drvif_iic_instance_t * p_inst,
hs3001_cfg_t * p_cfg);
```

**Note**: Because the Middleware supports multiple instances of the HS3001 sensors on multiple I<sup>2</sup>C busses, these are typically abstracted one layer to account for multiple device instances and multiple configurations. In this case, your calls may be abstracted by a function table in the configuration instance. For details see the Smart Configurator manuals.

For this case shown, it is defined by the ctrl instance for Sensor 0:

#### HS3001 API Examples Device 0 defined by:

```
err = g_hs300x_sensor0.p_api->measurementStart(g_hs300x_sensor0.p_ctrl);
```

## 5. Additional Information

For additional information on the Quick Connect IoT solutions and supporting documents, visit Renesas Quick Connect.

## 6. Revision History

Revision	Date	Description
1.0	Jul 2, 2021	Initial release.



## Appendix A

Boards that require interposer.

Family/Device Group	Board Name	
RA/RA4W1	EK-RA4W1	
RA/RA2A1	EK-RA2A1	
RA/RA4M1	EK-RA2A1	
RA/RA6M1	EK-RA6M1	
RA/RA6M2	EK-RA6M2	
RA/RAM3	EK-RA6M3	
RA/RAM3G	EK-RA6M3G	
RX/RX111	RX111-Starter-Kit	
RX/RX231	RX231-Starter-Kit	
RX/RX23W	RX23W-Starter-Kit	
RX/RX23T	RX23T-Starter-Kit	
RX/RX24T	RX24T-Starter-Kit	
RX/RX24U	RX24U-Starter-Kit	
Synergy/S5D9	PK-S5D9	
Synergy/S3A7	DK-S3A7	
Synergy/S128	DK-S128	
Synergy/S1JA	TB-S1JA	
Synergy/S3A6	TB-S3A6	
Synergy/S7G2	DK-S7G2	

#### Table 2. Kits Requiring Type 6A Interposer



#### IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES ("RENESAS") PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Rev.1.0 Mar 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/