RL78/G15

Wireless Communication with the XBee ZB S2C and HS300x

Introduction

This application note describes a sample program to control the XBee ZB S2C* on the RL78/G15 to perform wireless communication. Humidity and temperature data acquired from the HS300x (humidity and temperature sensor) is communicated wirelessly. This application note also describes how to control the HS300x on the RL78/G15.

Target Device

RL78/G15

Contents

1. Specification

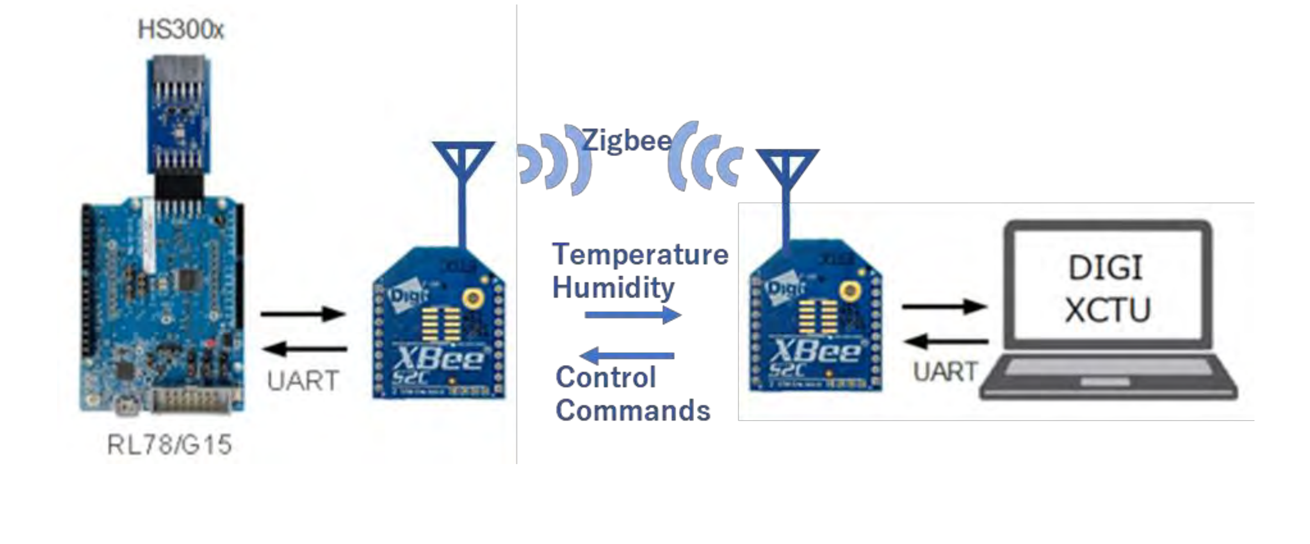
1.1 Overview

This application note describes how to connect the RL78/G15 with the XBee ZB S2C and wirelessly communicate humidity and temperature data acquired from the HS300x. The sample program uses the XBee ZB S2C by the UART function incorporated into the RL78/G15 to wirelessly communicate humidity and temperature data.

In addition, this application note describes how to acquire and calculate humidity and temperature data from the HS300x humidity and temperature sensor on the RL78/G15. The sample program controls the HS300x by the built-in I2C driver of the RL78/G15, and acquires humidity and temperature ADC data from the HS300x. Additionally, this sample program calculates acquired data on the RL78/G15.

Furthermore, this application note describes how to operate the RL78/G15 through the XBee ZB S2C on a PC.

Figure 1-1 Schematic Diagram of the Entire System



2. Operation Confirmation Conditions

The operation of the sample code provided with this application note has been tested under the following conditions.

Table 2-1 Operation Confirmation Conditions

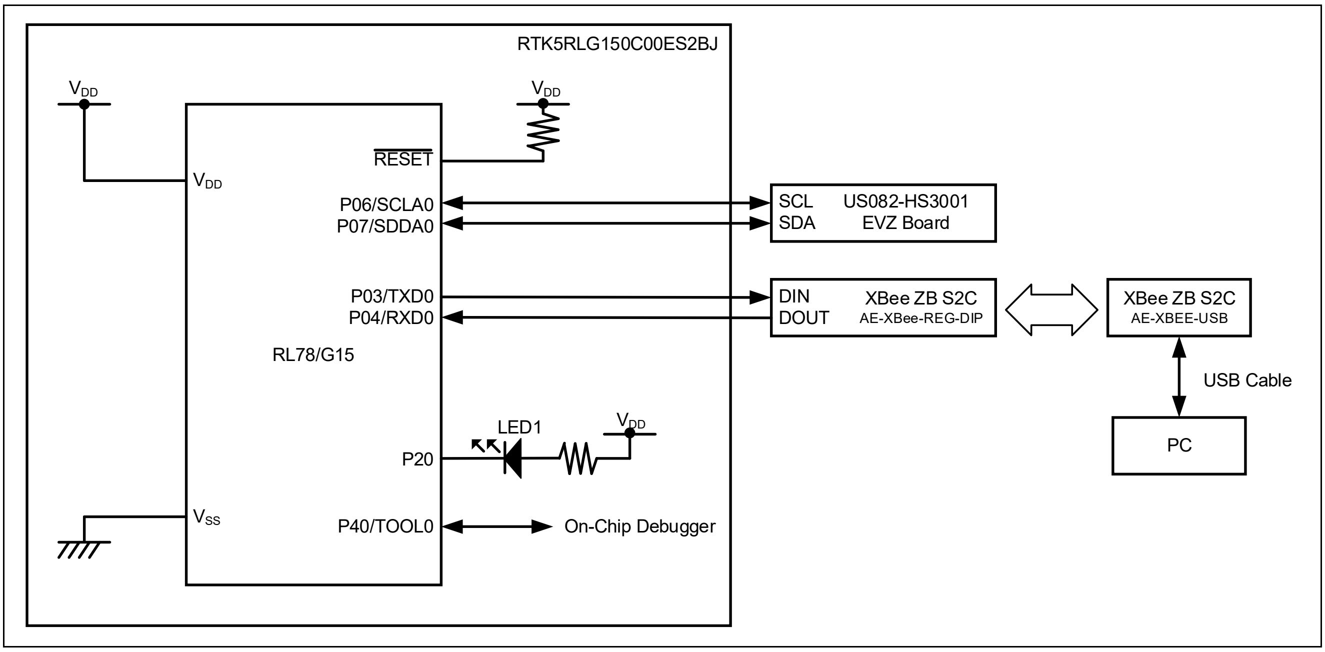
Item	Description
MCU used	RL78/G15 (R5F12068)
Operating frequency	High-speed on-chip oscillator clock (f_{IH}): 16 MHz
Memory size (memory used)	ROM 8KB (6441 bytes used) RAM 1KB (119 bytes used)
Operating voltage	5.0 V (can be operated at 2.4 V to 5.5 V) SPOR operation At rising edge TYP. 2.57 V (2.44 V to 2.68 V) At falling edge TYP. 2.52 V (2.40 V to 2.62 V)
Integrated development environment (CS+)	CS+ V8.11.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.13.00 from Renesas Electronics Corp.
Integrated development environment (e2studio)	e2 studio V2024-1 from Renesas Electronics Corp.
C compiler (e2studio)	CC-RL V1.13.00 from Renesas Electronics Corp.
Smart Configurator	V.1.9.0 from Renesas Electronics Corp.
Board support package (r_bsp)	V.1.60 from Renesas Electronics Corp.
Board used	RL78/G15-20p Fast Prototyping Board (RTK5RLG150CLG000BJ)
Temperature/Humidity Sensor Modules	Relative Humidity Sensor Pmod™ Board (US082-HS3001EVZ)
Data transmission module	XBee ZB S2C

3. Hardware Descriptions

3.1 Example of Hardware Configuration

Figure 3-1 shows an example of the hardware configuration used in the application note.

Figure 3-1 Hardware Configuration



Note 1. This schematic circuit diagram is simplified to show the outline of connections. When creating circuits, design them so that they meet electrical characteristics by properly performing pin processing. (Connect input-only ports to V_{DD} or V_{SS} individually through a resistor.)

Note 2. VDD must not be lower than the reset release voltage (VSPOR) that is specified for the SPOR.

3.2 List of Pins to be Used

Table 3-1 lists the pins to be used and their functions.

Table 3-1 Pins to be Used and Their Functions

Pin name	I/O	Function
P03 / TxD0	Output	UART Data transmission pin (Via XBee ZB S2C)
P04 / RxD0	Input	UART Data reception pin (From XBee ZB S2C)
P06 / SCLA0	Input/Output	IICA serial clock (with HS300x)
P07 / SDDA0	Input/Output	IICA serial data bus (with HS300x)
P20	Output	LED1 control pin (Low Active)

4. Module Specifications

This section describes the specifications of the XBee ZB S2C and HS300x.

4.1 Specifications of XBee ZB S2C

Table 4-1 outlines the specifications of the XBee ZB S2C.

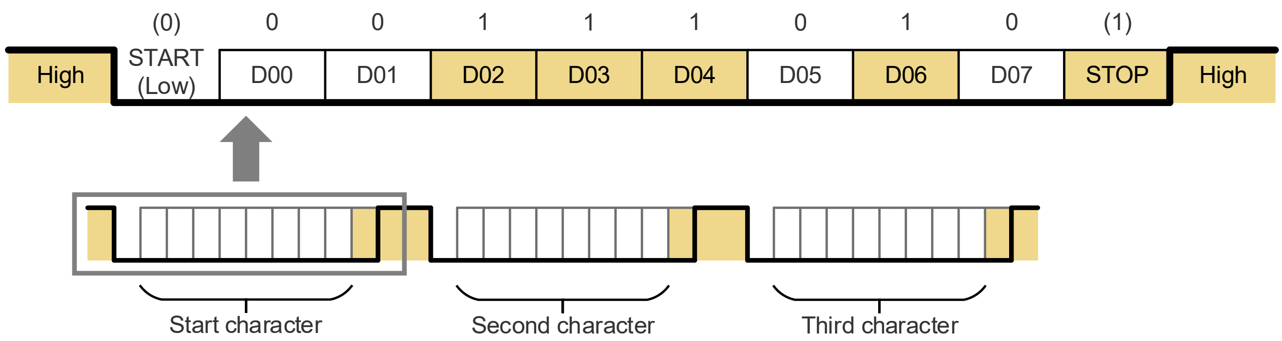
Table 4-1 Specifications of XBee ZB S2C

Item	Description
Data rate	RF 250 Kbps, serial (up to 1 Mbps)
Indoors/urban range	Up to 60 m
Outdoors/expected range	Up to 1,200 m
Transmission power	3.1 mW (+5 dBm) / 6.3 mW (+8 dBm) in boost mode
Receiver sensitivity (1% PER)	-100 dBm / -102 dBm in boost mode
Serial data interface	UART, SPI
Configuration method	API or AT command, local or wireless
Frequency band	ISM 2.4 GHz
Operating voltage	2.7 to 3.6 V
Auxiliary board for connecting MCU	AE-XBee-REG-DIP (2.54-mm pitch conversion board)
Auxiliary board for connecting PC	AE-XBEE-USB (USB interface board)

4.1.1 UART communication interface

The following figure shows the UART format for communicating humidity/temperature data.

Figure 4-1 UART Communication Format



If " : " (start character[0x3A]) is entered as the first character, the command is judged.
The second character is used to judge whether to acquire data (G) or control LED (L).
The third character determines temperature (T) or humidity (H) or ON (1) or OFF (0).

4.2 Specifications of HS300x

Table 4-2 outlines the specifications of the HS300x humidity and temperature sensor.

Table 4-2 Specifications of HS300x Sensor

Item	Description
Humidity measurement range	0 to 100%RH
Humidity accuracy	±1.5%RH (typ.) (HS3001, 10 to 90%RH, 25℃)
ADC resolution	0.01%RH (typ.), (14 bits)
Measurement resolution	8, 10, 12, 14 bits
RH response speed	1 second (typ.) (with 1 m/s air flow) 4 seconds (typ.) (in a sealed space)
Temperature sensor accuracy	±0.2℃ (typ.) (HS3001, HS3002, -10 to 80℃)
Average current	Average 24.4 uA (14-bit resolution, power supply voltage 3.3 V supplied), 1 RH and temperature measurement per second
Sleep current	-40 to 85℃: 0.6 μA (typ.) -40 to 125℃: 1 μA (typ.)
Power supply voltage	2.3 V to 5.5 V: 3.3 V (typ.)
Extended power supply voltage	1.8 V (-20℃ to +125℃)
Operating temperature	-40℃ to +125℃

Table 4-3 List of Sensor Functions

Item	Description
I^2C communication	Sensor data are transferred through I^2C communication.
Measurement mode	This software operates with the sensor in sleep mode. On completion of measurement triggered by a measurement request, the sensor enters sleep mode.
Measurement request	The sensor in sleep mode is placed in the measurement state upon receiving a measurement request.
Data fetch	At the end of a measurement cycle, valid data can be acquired.
Status bits	The status bits for the results of measurement indicate whether the current data are valid.

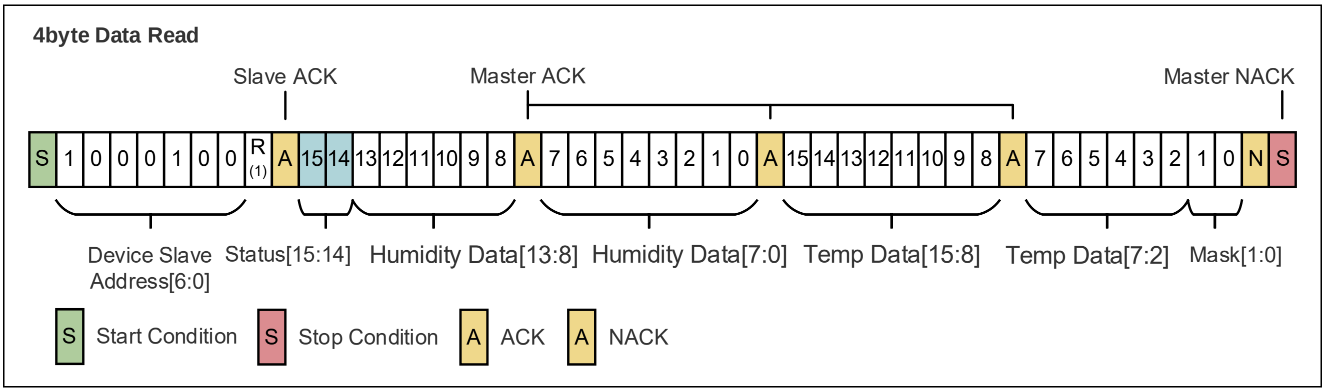
Note: This software does not support the following functions of the HS300x:

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

4.2.1 I2C Communication Interface

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

Figure 4-2 Format of I2C Communication



The status bits indicate the state of data as follows.

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

01B: Invalid data that have already been acquired

4.2.2 Expressions for Converting Output Values to Humidity and Temperature

The HS300x software converts the acquired ADC data to values for humidity and temperature, and then outputs them.

The humidity conversion expression is as follows.

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

The temperature conversion expression is as follows.

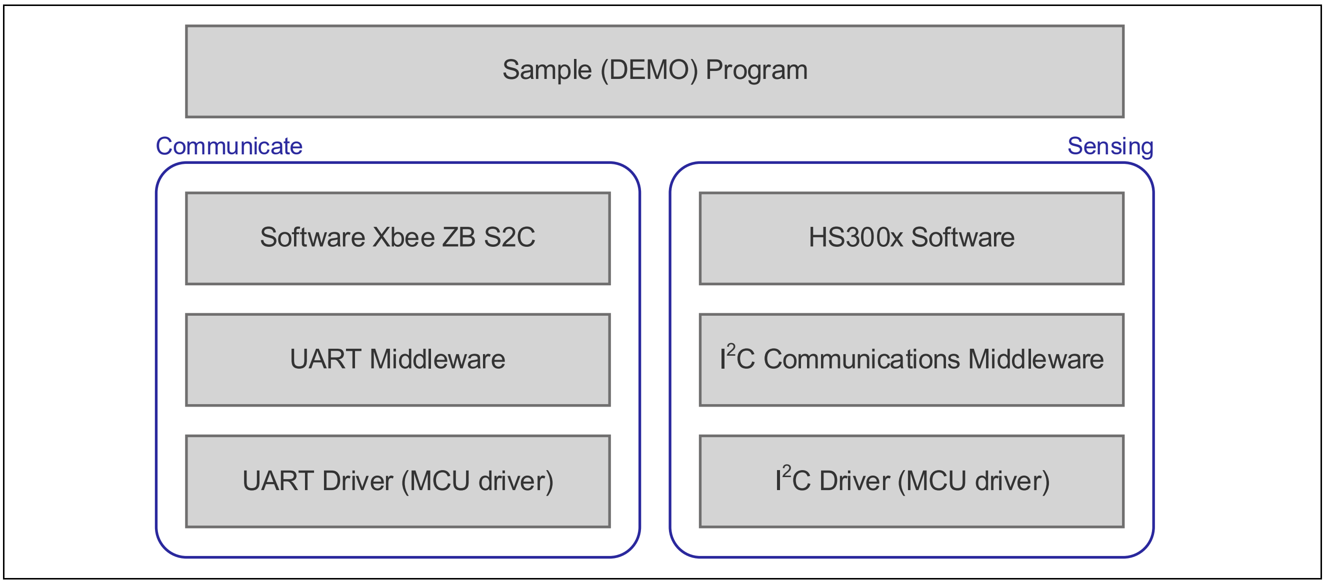
$$\mathrm{Temperature}[°\mathrm{C}] = \left(\frac{\mathrm{Temperature}[15:2]}{2^{14}-1}\right) * 165 - 40$$

5. Sample Program

5.1 Sample Program Structure

Figure 5-1 is a block diagram of the sample program structure.

Figure 5-1 Block Diagram of Software



5.2 Sample Program Command Specifications

The following table lists the sample program commands to control the RL78/G15 through the XBee ZB S2C from PCs or other devices.

Table 5-1 List of Sensor Functions

Command	Description
:L0	Turns off LED1.
:L1	Turns on LED1.
:GT	Acquires temperature data.
:GH	Acquires humidity data.

5.3 Specifications of HS300x API Functions

5.3.1 List of HS300x API functions

The following table lists the sensor API functions. For details of the API functions, refer to the separately provided HS300x Sensor API FIT Module application note (R01AN5893) and Renesas Sensor Control Modules Firmware Integration Technology (R01AN5892).

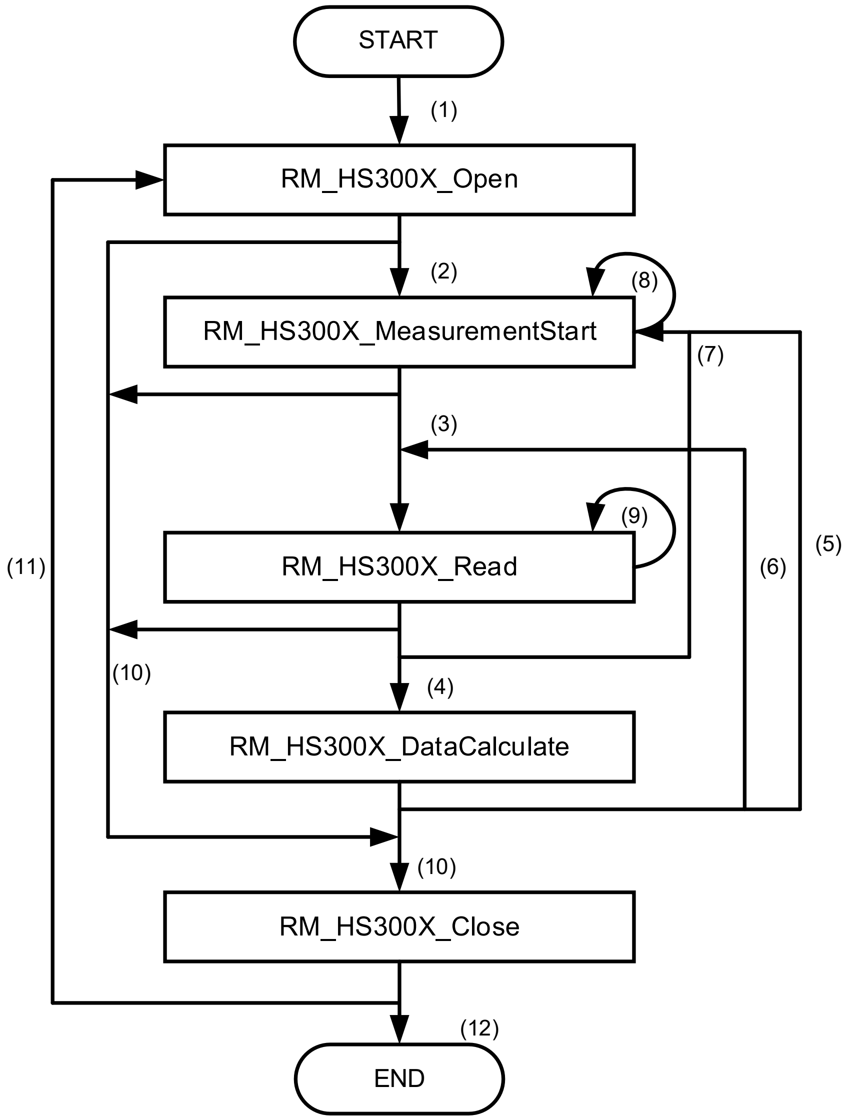
Table 5-2 List of HS300x API Functions

Function	Function
RM_HS300X_Open	Starts control of the sensor.
RM_HS300X_Close	Terminates control of the sensor.
RM_HS300X_MeasurementStart	Starts measurement by the sensor.
RM_HS300X_Read	Acquires data from the sensor.
RM_HS300X_DataCalculate	Calculates values from the data acquired from the sensor.

5.3.2 Guide to Using the API Functions

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

Figure 5-2 Diagram of Transitions between API Function Calls



Flow No.	Function	Transition Case
(1)	Start HS300x	not open HS300x
(2)	Start Measurement HS300x	open success
(3)	ADC Read	Measurement success (callback state success)
(4)	Humidity & Temp. calculate	ADC Read success (callback state success)
(5)	next measurement start	DataCalculate return success and next ADC data Get
(6)	get retry ADC data	User get retry ADC data or DataCalculate return invalid data
(7)	next measurement (Not calculate)	The user requests only ADC data and measures the next data
(8)	Retry Measure Start	Measurement start Waiting for response (callback none and polling)
(9)	Retry ADC Read	Read ADC Waiting for response (callback none and polling)
(10)	Close HS300x	Error state or user stop
(11)	Restart HS300x	Close success and user restart
(12)	Stop HS300x	Close success or Error

The conditions for calling the individual functions are shown below.

・RM_HS300X_Open:	(1) Activation of HS300x, or (11) restart after a call of RM_HS300X_Close
・RM_HS300X_Close:	(10) Successful completion or abnormal end of individual processing
・RM_HS300X_MeasurementStart:	(2) Start of measurement after a call of RM_HS300X_Open, (5) (7) acquisition of the next measured data, or (8) retry after waiting for the response to the measurement start request
・RM_HS300X_Read:	(3) Acquisition of measured data after a call of RM_HS300X_MeasurementStart, or (9) retry after waiting for the response to the data acquisition request
・RM_HS300X_DataCalculate:	(4) Calculation of humidity and temperature data after a call of RM_HS300X_Read

Notes: Since RM_HS300X_Open checks the state of the I^2C driver, the I^2C driver must be opened before the RM_HS300X_Open processing.

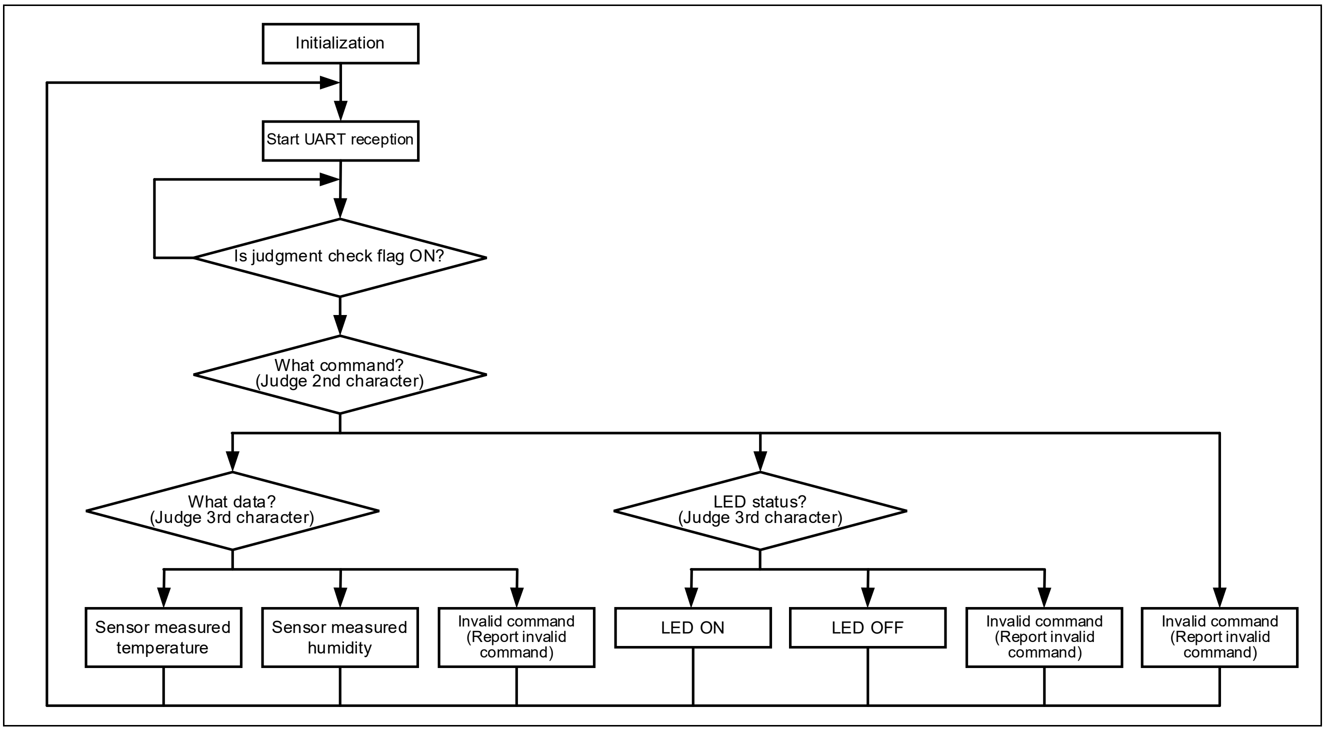
When measurement is started by RM_HS300X_MeasurementStart, the sensor stops measurement after outputting the ADC data. Therefore, RM_HS300X_MeasurementStart must be called at least once every time before RM_HS300X_Read processing.

5.4 Flowcharts

5.4.1 Overall Flowchart

The following figure shows the overall flowchart.

Figure 5-3 Overall Flowchart



5.5 Sample Program Structure

5.5.1 Peripheral Functions to Be Used

The following table lists the peripheral functions used in the sample program.

Table 5-3 Peripheral Functions to Be Used

Peripheral Function	Use
PORT	Used to control LED1 mounted on RL78/G15_FPB.
IICA0	Used for I²C communication with HS300x and acquisition of humidity and temperature data from HS300x.
UART0	Used for UART communication with XBee ZB S2C, command transmission to XBee ZB S2C, and confirmation of response results returned from XBee ZB S2C.

5.5.2 Settings of Peripheral Functions

The following table lists the settings of Smart Configurator used in the sample program. The items and settings in each table of Smart Configurator Settings are described with the names displayed on the actual setting screen.

Table 5-4 Smart Configurator Settings (1/2)

Category	Item	Settings
Smart Configurator >> Clock		The settings on the [Clock] tab are as follows:
	VDD setting	2.4 V ≦ VDD ≦ 5.5 V
	High-speed on-chip oscillator	Checked Frequency: 16 MHz
	X1 oscillator circuit	Unchecked
	Low-speed on-chip oscillator	15 kHz
	Source selection for main system clock (f_{MAIN})	Select [High-speed on-chip oscillator (f_{IH})]. (f_{MAIN}: 16 MHz)
	Operating clock supply for 12-bit interval timer	Unchecked (f_{IL}: 15 kHz)
Smart Configurator >> System		The settings on the [System] tab are as follows:
	Pseudo RRM/DMM function setting	Not used
	Start/Stop function setting	Not used
	Security ID setting	Set security ID. Security ID "0x00000000000000000000"
	RESET pin setting	Used
	Operation mode setting	2.52 V
Smart Configurator >> Component >> r_bsp		Use the default settings except for the following changes:
	API functions disable(R_BSP_StartClock, R_BSP_StopClock)	Disable
	API functions disable(R_BSP_SetClockSource)	Disable
	API functions disable(R_BSP_ChangeClockSetting)	Disable
	API functions disable(R_BSP_SoftwareDelay)	Disable
Smart Configurator >> Component >> PORT		Use the default settings except for the following changes:
	Port selection	PORT2
	PORT2	P20: Select [Output], [Output 1].
Smart Configurator >> Component >> Config_IICA0		The settings on [Config_IICA0] are as follows:
	Local address	16
	Operation mode setting	Set [Standard] and transfer clock (f_{SCL}): 100000 (bps)
	tR and tF settings	Unchecked
	Interrupt setting	Level 3 (lowest priority)
	Callback function setting	Select [Master transmission end], [Master reception end], and [Master error].
	Callback extension setting	Unchecked

Table 5-5 Smart Configurator Settings (2/2)

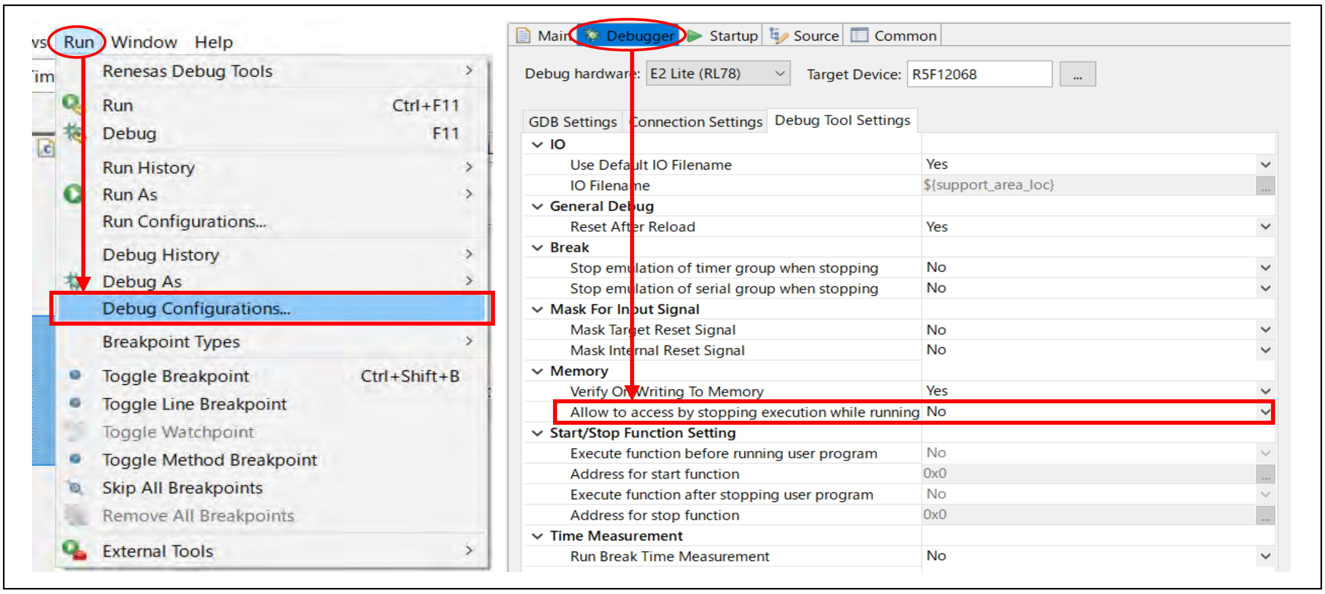
<table>
<tr><th>Category</th><th colspan="2">Item</th><th>Settings</th></tr>
<tr><td colspan="3">Smart Configurator >> Component >> Config_UART</td><td>The settings on [Config_UART0] are as follows:</td></tr>
<tr><td rowspan="19"></td><td rowspan="10">Transmission</td><td>UART0 clock setting</td><td>Operating clock: CK00
Clock source: f_{CLK}/2^3</td></tr>
<tr><td>Transfer mode setting</td><td>Continuous transfer mode</td></tr>
<tr><td>Data bit length setting</td><td>8 bits</td></tr>
<tr><td>Data transfer direction setting</td><td>LSB</td></tr>
<tr><td>Parity setting</td><td>No parity bit</td></tr>
<tr><td>Stop bit length setting</td><td>1 bit</td></tr>
<tr><td>Transmit data level setting</td><td>Non-inversion (normal)</td></tr>
<tr><td>Transfer rate</td><td>9600 (bps)</td></tr>
<tr><td>Interrupt setting</td><td>Level 3 (lowest priority)</td></tr>
<tr><td>Callback function setting</td><td>End of transmission</td></tr>
<tr><td rowspan="8">Reception</td><td>UART0 clock setting</td><td>Operating clock: CK00
Clock source: f_{CLK}/2^3</td></tr>
<tr><td>Data bit length setting</td><td>8 bits</td></tr>
<tr><td>Data transfer direction setting</td><td>LSB</td></tr>
<tr><td>Parity setting</td><td>No parity bit</td></tr>
<tr><td>Transmit data level setting</td><td>Non-inversion (normal)</td></tr>
<tr><td>Transfer rate</td><td>9600 (bps)</td></tr>
<tr><td>Interrupt setting</td><td>Level 3 (lowest priority)</td></tr>
<tr><td>Callback function setting</td><td>Reception end, reception error</td></tr>
<tr><td colspan="3">Smart Configurator >> Pin</td><td>The settings on the [Pin] tab are as follows:</td></tr>
<tr><td rowspan="2"></td><td colspan="2">Serial array unit (SAU0)</td><td>SAU00
RxD0: P04
TxD0: P03</td></tr>
<tr><td colspan="2">Serial interface IICA (IICA0)</td><td>IICA0
SCLA0: P06
SDAA0: P07</td></tr>
</table>

5.5.2.1 How to Prevent Memory from being Overwritten

If you use the default debugger settings as is, the debugger will occupy the top address (4 bytes) of the RAM (RRM function). Therefore, you need to change the settings.

As shown in Figure 5-4, select the debugger memory in [Debug configuration] and change [Interrupt execution to access memory] to "No" to stop the RRM function and prevent the first 4 bytes of the RAM from being overwritten.

Figure 5-4 Change Debugger Settings



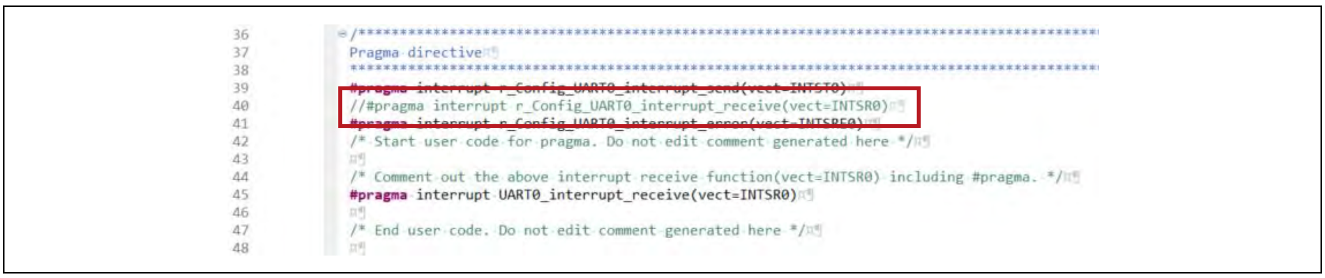
5.5.2.2 Notes on Generating Code with Smart Configurator

The sample program provides a dedicated interrupt function to prepare for command recognition in the UART reception interrupt function. Although the #pragma command to call this function is also coded together, when new code is generated by Smart Configurator, the #pragma function (former reception interrupt function) is also coded. As a result, an interrupt table overload error occurs at build time.

After generating new code with the Smart Configurator, before building, make sure that the #pragma command to call the former reception interrupt function is commented out.

Figure 5-5 Part to be Commented-out after Code Generation

Target file: Config_UART0_user.c, line 40



5.5.3 File Structure

The following table lists the file structure of the sample program.

Table 5-6 File Structure

Folder name, file name	Explanation
src	Program storage folder
├ command	Command related program storage folder
│ ├ command.c	Command related source file
│ └ command.h	Command related header file
├ r_comms_i2c_rl	I²C communication middleware layer storage folder (the one for G23 is used)
├ r_hs300x	Sensor middleware layer storage folder (the one for G23 is used)
├ smc_gen	Smart Configurator generation folder
│ ├ Config_IICA0	
│ ├ Config_PORT	
│ ├ Config_UART0	
│ ├ general	
│ ├ r_bsp	
│ └ r_config	
├ main.c	Main processing source file
├ rl78_hs300x.c	Sensor software measurement flow source file
└ rl78_hs300x.h	Sensor software measurement flow header file

For details on the I²C communication middleware layer storage folder, sensor middleware layer storage folder, and Smart Configurator generation folder, refer to the application notes stored in the doc folder in each folder.

6. Description of Software Operation

This sample program uses e² studio as a tool for developing MCU programs and checking operation, and XCTU as a tool for controlling the XBee ZB S2C and checking its operation.

6.1 How to Operate XCTU

XCTU is the XBee development tool provided by Digi International. You can use this tool to easily operate and control the XBee ZB S2C system settings through the GUI, and you can also monitor communication details of the XBee ZB S2C. This section explains how to initialize the XBee ZB S2C by using XCTU.

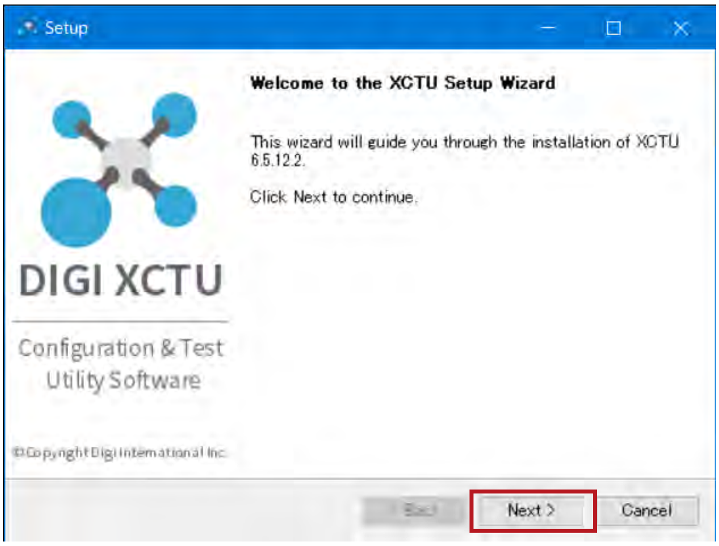
6.1.1 How to Install XCTU

XCTU can be downloaded from the Digi International web page. Download the XCTU installer for the OS running on your PC from [Resources & Utilities] at the bottom of the web page at the following URL.

https://hub.digi.com/support/products/xctu/

Launch the downloaded installer to start the setup wizard. Press [Next] to start setup.

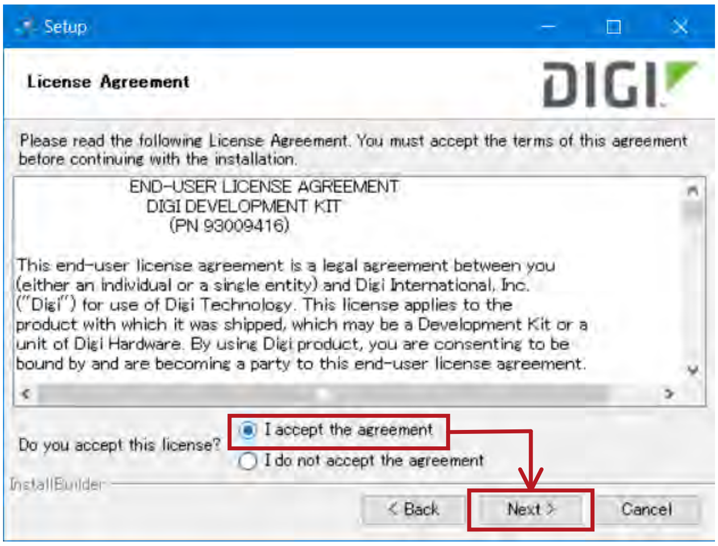
Figure 6-1 Installation of XCTU (1/7)



Use of XCTU requires a license agreement with Digi International.

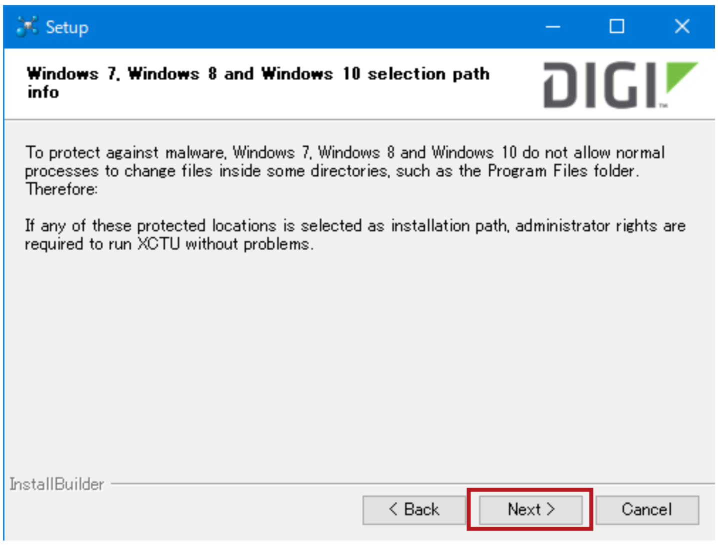
If you agree with the contents, select [I accept the agreement] and press [Next].

Figure 6-2 Installation of XCTU (2/7)



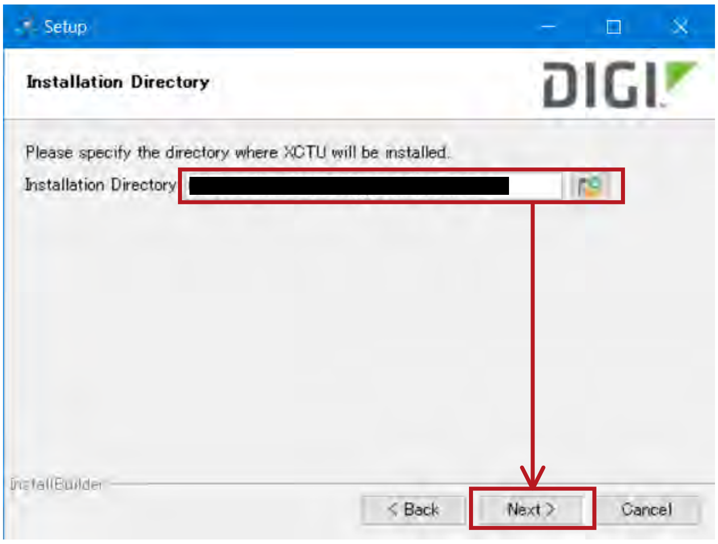
Some directories are commonly protected on a PC for anti-malware. Therefore, if you try to install XCTU in a protected directory, you will be informed that it must be installed with administrator privileges. Confirm the displayed contents and then press [Next].

Figure 6-3 Installation of XCTU (3/7)



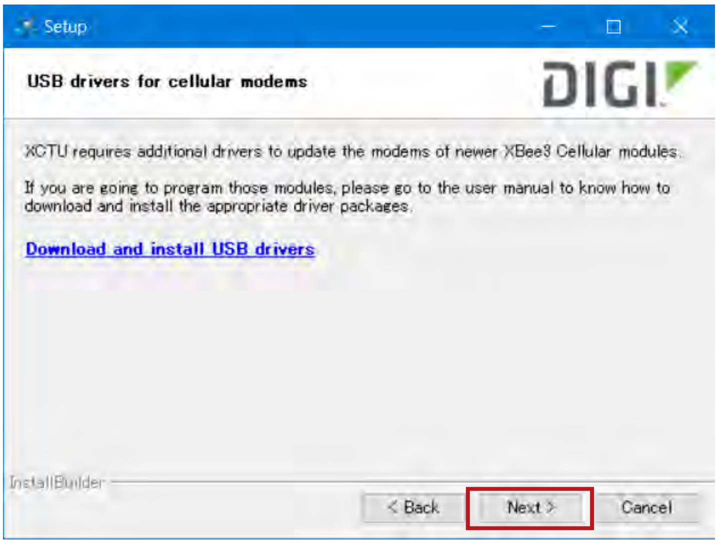
Then, specify the directory to install XCTU. After specifying the installation destination, press [Next].

Figure 6-4 Installation of XCTU (4/7)



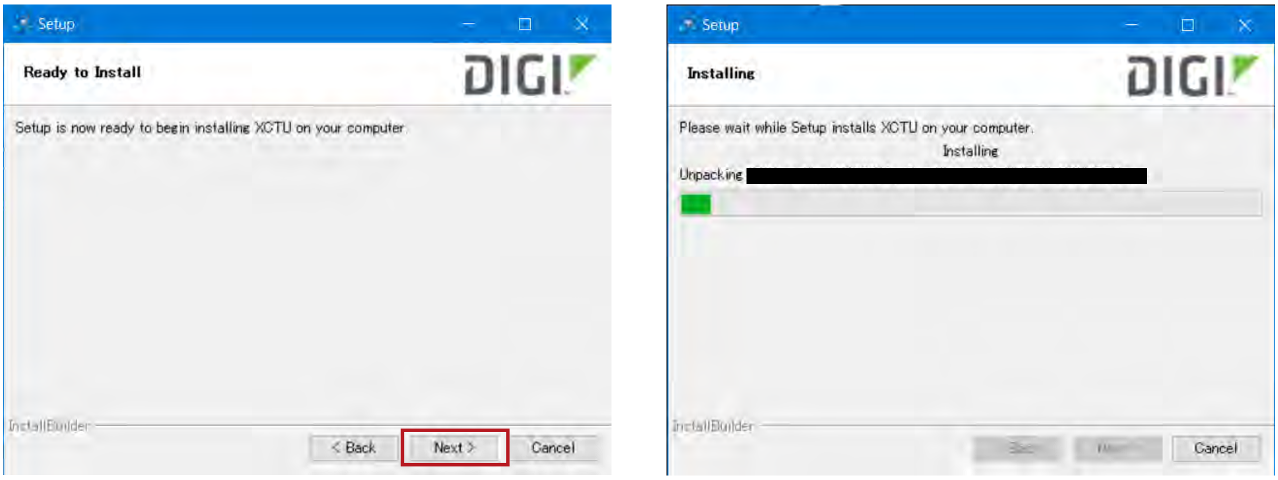
When using the XBee cellular model, USB driver installation is additionally required. However, since this device is not the target here, press [Next] to continue.

Figure 6-5 Installation of XCTU (5/7)



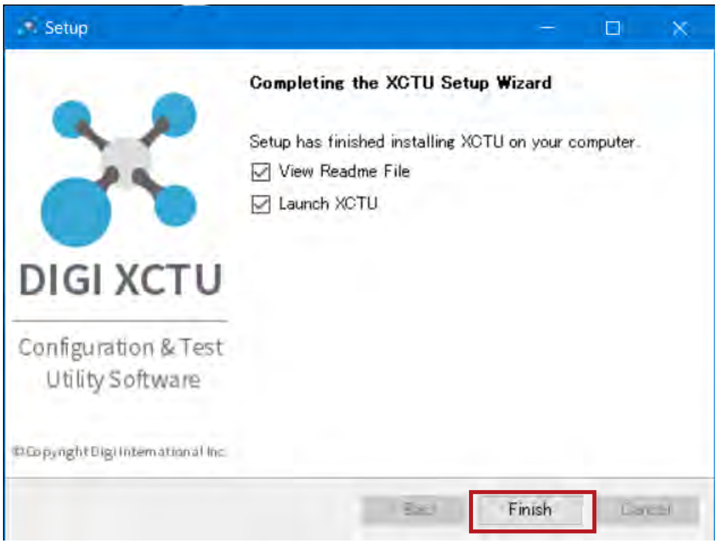
When an installation preparation completion screen appears, press [Next] to start installation. It takes some time for the installation to finish.

Figure 6-6 Installation of XCTU (6/7)



The display of a setup completion screen indicates that the XCTU installation has been completed.

Figure 6-7 Installation of XCTU (7/7)

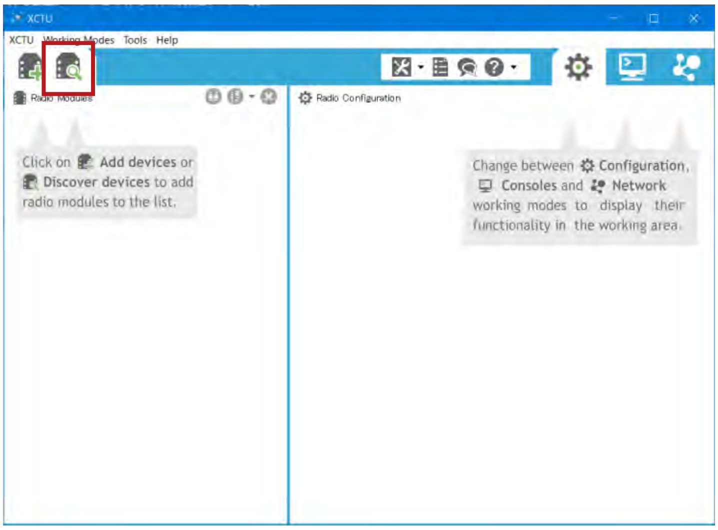


6.1.2 How to Connect XCTU with XBee ZB S2C

After installation of XCTU is complete, connect XCTU with the XBee ZB S2C. This section explains how to connect XCTU with the XBee ZB S2C.

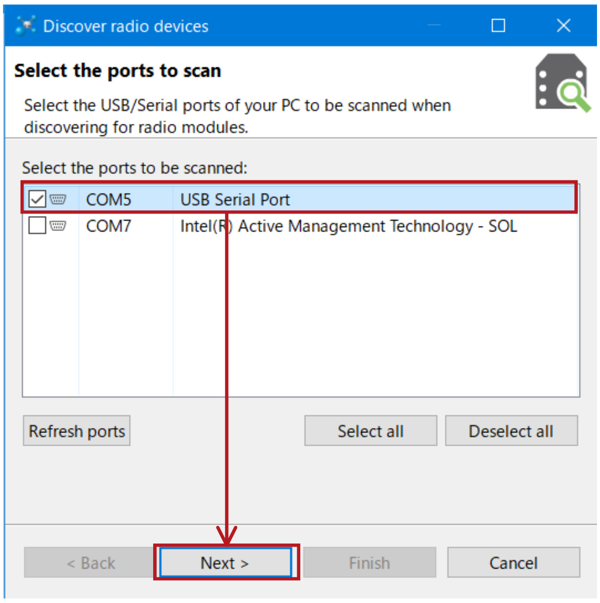
After starting XCTU and connecting the PC with the XBee ZB S2C by a USB cable, click the magnifying glass icon (Discover devices) at the upper left of the XCTU window.

Figure 6-8 How to Connect XCTU with XBee ZB S2C (1/5)



When you click the Discover devices icon, a window showing the port number connected to the PC opens. The port number of the connected XBee ZB S2C is displayed here. Select the port of the XBee ZB S2C and press [Next].

Figure 6-9 How to Connect XCTU with XBee ZB S2C (2/5)



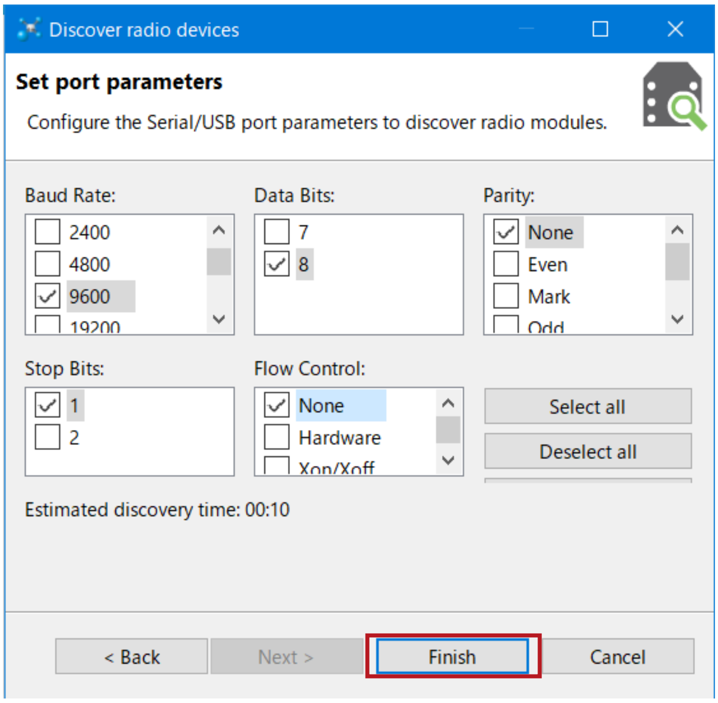
The number following COM differs depending on your PC environment. If you do not know the port number of the connected XBee ZB S2C, remove the XBee ZB S2C from the PC and then press [Refresh ports]. The port number that disappeared at that time is the port number of the XBee ZB S2C.

Then, a window opens showing the UART parameters required for the USB connection between the PC and the XBee ZB S2C. The following table lists the default connection settings of the XBee ZB S2C. Check the setting values in the table, select the setting values required for the connection, and press [Finish].

Table 6-1 UART Connection Parameters when Connecting PC with XBee ZB S2C

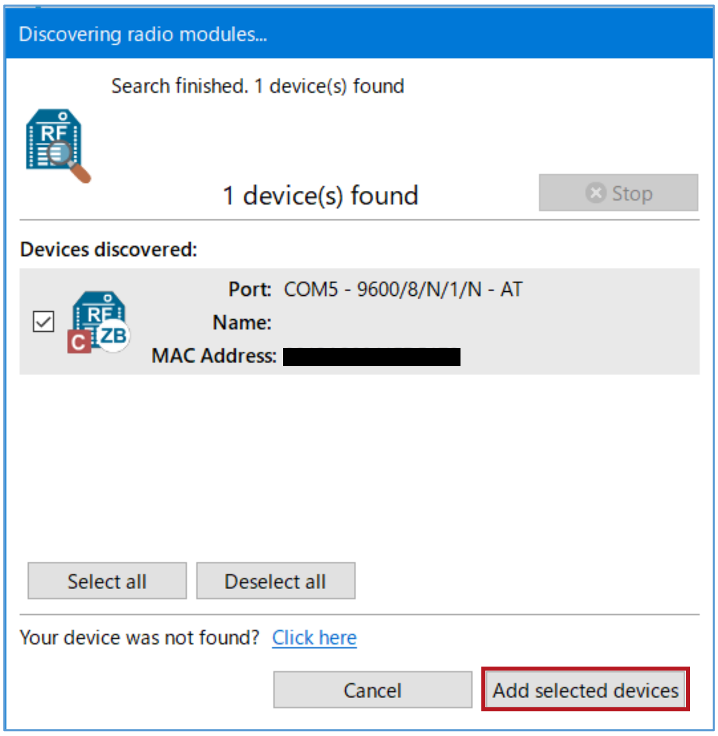
Item	Setting value (default)
Baud rate (bps)	9600
Data length	8 bits
Parity bit	None
Stop bit	1 bit
Flow control	None

Figure 6-10 How to Connect XCTU with XBee ZB S2C (3/5)



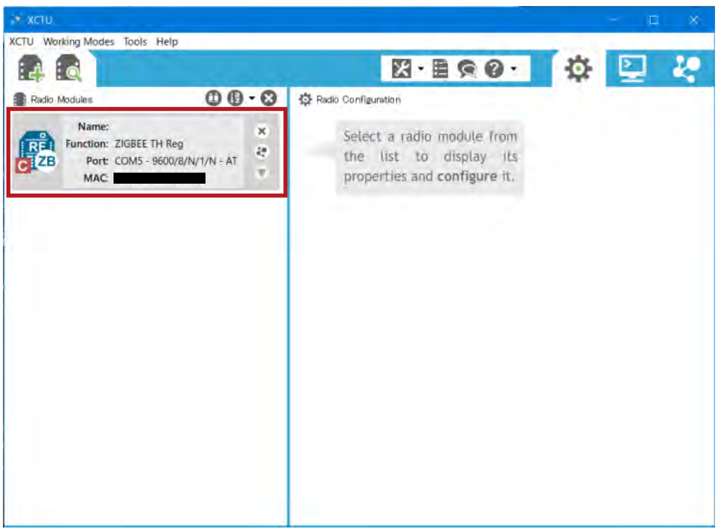
After finishing searching for the device with the specified connection parameters, XCTU displays the detected device. Confirm that the device shown in [Devices discovered] is selected, and then press [Add selected devices].

Figure 6-11 How to Connect XCTU with XBee ZB S2C (4/5)



After device detection ends successfully, you are brought back to the screen after XCTU starts up. If the new device name is displayed in [Radio Modules] on the left side of the XCTU window, the connection between XCTU and the XBee ZB S2C has been completely established.

Figure 6-12 How to Connect XCTU with XBee ZB S2C (5/5)



6.1.3 How to Initialize XBee ZB S2C using XCTU

Even if you prepare two XBee ZB S2C devices, they cannot communicate with each other by the default settings. This section describes initialization for communication between XBee ZB S2C devices using XCTU.

When XBee ZB S2C devices are detected and you click the name of the connected device through XCTU, a list of setting parameters of the connected XBee ZB S2C is shown on the right side of the XCTU window. You can change and save these parameters to set them.

Figure 6-13 How to Initialize XBee ZB S2C Using XCTU

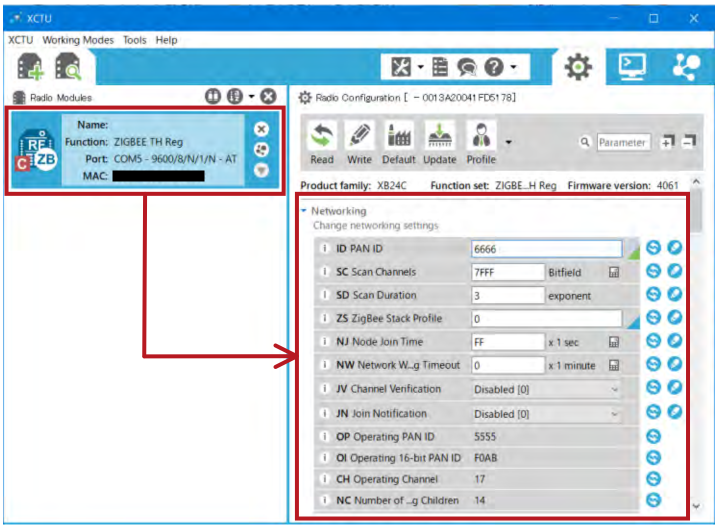


Table 6-2 List of XBee ZB S2C Setting Items Configured by XCTU

Setting items configured by XCTU	Setting value	Description
ID PAN ID	Any value However, both devices must have the same value.	The XBee ZB S2C can only communicate with the device with the same ID parameter.
DH Destination Address High	13A200	Specify the upper address of the communication destination. The value of the XBee ZB S2C is fixed to 13A200.
DL Destination Address Low	DL value for both devices	Specify the lower address of the communication destination. The value of the XBee ZB S2C can be checked by connecting to the XCTU to check the DL value, or by checking the main unit (MAC).
CE Coordinator Enable	Either one is Enabled [1]. The other is Disabled [0].	When communicating between XBee ZB S2C devices, at least one must be a Coordinator.
AP API Enable	Both devices must be in Transparent mode [0].	Select a mode to pass data as is without modification between the MCUs on the PC.

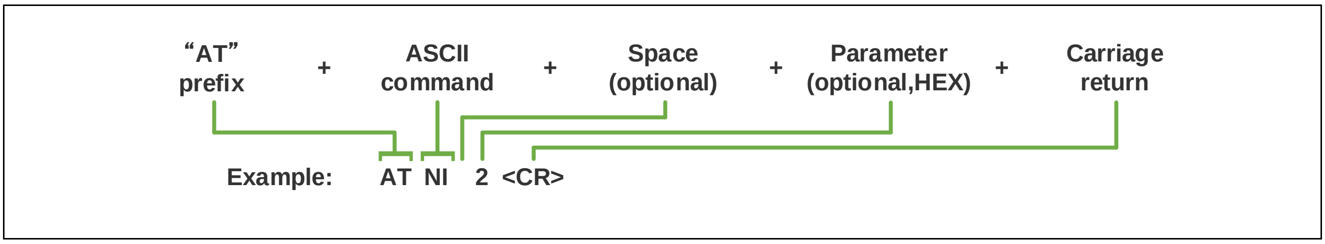
Five settings are required to communicate between the XBee ZB S2C devices. Change and save the settings for both the XBee ZB S2C devices, respectively.

When you change a setting value, XCTU displays a green triangle at the right corner of the item. This indicates that the changed setting has not been saved. After changing a setting, press the pencil mark on the right side of the item to save the value. The green triangle disappears when the setting has been saved.

6.1.4 Initialization Using the AT Commands

For initialization of the XBee ZB S2C, a method of sending the AT commands through the UART interface from an MCU or PC without using XCTU is also provided. As in the case of XCTU, for items to be set, use the AT commands to write communication-related parameter values to the XBee ZB S2C. The AT command corresponds to the first two characters of "Setting items configured by XCTU" in Table 6-2.

Figure 6-14 Transmission Format of AT Commands



To control the XBee ZB S2C using the AT commands, it is necessary to enter command mode before sending the AT commands. Refer to the following online manual for details about how to enter command mode and about parameters to be sent for each AT command.

https://www.digi.com/resources/documentation/Digidocs/90001500/Default.htm

6.2 Preparation of Hardware

Connect the RL78/G15 with the XBee ZB S2C and HS300x as shown below.

Figure 6-15 Overall View of Connection

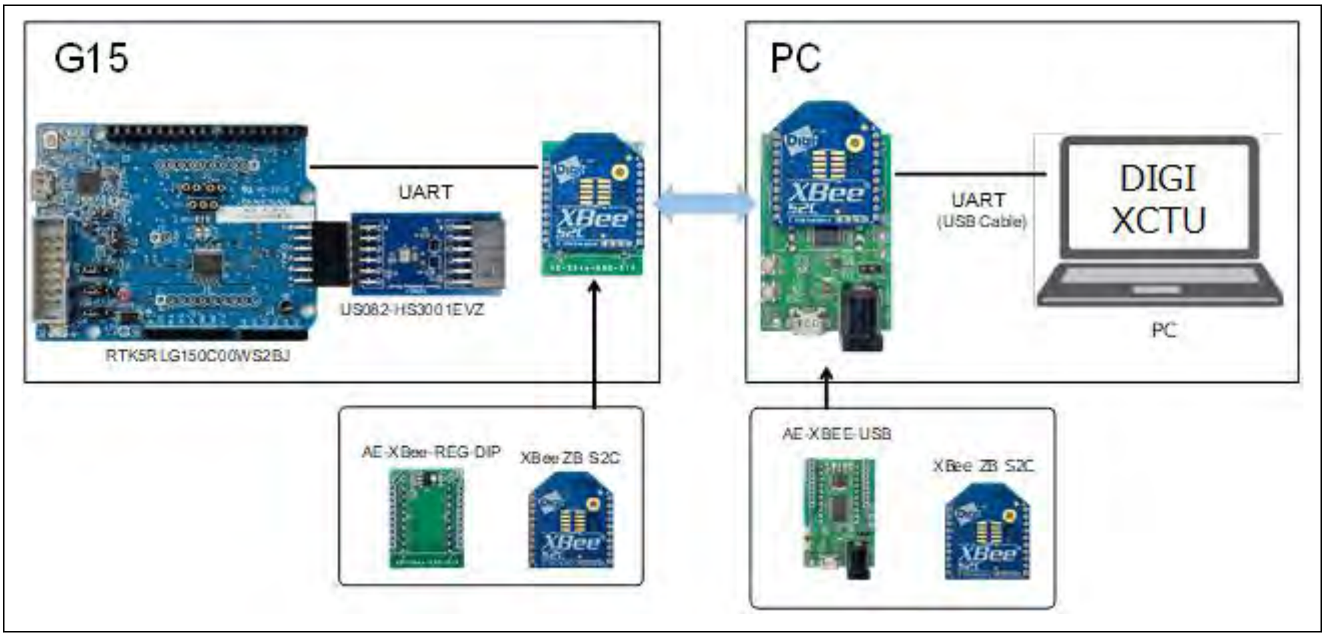


Table 6-3 Pins to which RL78/G15 and XBee ZB S2C Connect

RL78/ G15		XBee ZB S2C		Explanation
Pin number	Name	Pin number	Name	
-	VDD	Pin1	VCC	Power supply from RL78/G15 to XBee ZB S2C
-	GND	Pin10	GND	GND of RL78/G15 and XBee ZB S2C
P03	TXD0	Pin3	DIN	UART transmission from RL78/G15 to XBee ZB S2C
P04	RXD0	Pin2	DOUT	UART Reception from XBee ZB S2C to RL78/G15

Table 6-4 Pins to which RL78/G15 and HS300x Connect

RL78/ G15		HS300x		Explanation
Pin number	Name	Pin number	Name	
-	VDD	Pin4	VDD	Power supply from RL78/G15 to HS300x
-	GND	Pin6	VSS	GND of RL78/G15 and HS300x
P06	SCLA0	Pin1	SCL	I^2C clock transmission from RL78/G15 to HS300x
P07	SDDA0	Pin2	SDA	I^2C data transmission from RL78/G15 to HS300x

6.3 Operation Confirmation Methods

This sample software can send commands from terminal software such as XCTU and Tera Term to the RL78/G15, control LED1 mounted on the RL78/G15 Fast Prototyping Board, and control the HS300x sensor connected to the RL78/G15 Fast Prototyping Board.

After completing the initialization of the XBee ZB S2C described above, check the operation as follows:

(1) Detect the XBee ZB S2C connected with the PC using XCTU. To do this, click the [Discover devices] icon at the upper left of the XCTU window and connect the XBee ZB S2C on the PC side with the PC via a USB cable.

(2) Start the PC application, XCTU.

(3) Detect the XBee ZB S2C connected with the PC using XCTU. To do this, click the [Discover devices] icon at the upper left of the XCTU window and configure the UART connection settings.

(4) After detecting the XBee ZB S2C on the PC side using XCTU, click the PC icon (Consoles) at the upper right of the XCTU window to enter Console mode.

(5) If the console log is entirely grayed out, it is in a closed state in which the log cannot be referenced. So, in such a case, click [Open] above the console log. This completes the preparation of the XBee ZB S2C on the PC side.

(6) Supply power to the RL78/G15. At this point, the XBee ZB S2C on the G15 side is also turned on and the preparation is complete.

(7) "Ready." is shown in the console log of XCTU (command wait state).

(8) Enter a command (such as the temperature display command ":GT") in the console log of XCTU.

(9) "OK." is shown in the console log of XCTU, and the current temperature is displayed after 4 seconds.

(10) "Ready." is shown in the console log of XCTU (command wait state).

(11) Loop (8) and subsequent steps.

7. Reference Documents

・RL78/G15 User’s Manual: Hardware (R01UH0959)

・RL78/G15 Fast Prototyping Board User’s Manual (R12UM0042)

・HS300x Datasheet (R36DS0010EU0701)

・RA Family, RX Family, RL78 Family, RE01 256KB / 1500KB Group, RZ Family　HS300x Sample Software Manual (R01AN5897)

・RA Family, RX Family, RL78 Family, RZ Family Sensor Software Combination Manual (R01AN6591)

・RL78 Family Board Support Package Module Using Software Integration System（R01AN5522）

・RX Family Renesas HS300x Sensor Control Module Firmware Integration Technology (R01AN5893)

・RL78 Family Renesas Sensor Control Modules Software Integration System（R01AN6192）

・RX Family Renesas Sensor Control Modules Firmware Integration Technology（R01AN5892）

・RL78 Family Renesas Sensor I2C Communication Middleware Control Module Software Integration System (R01AN6193）

・RL78 Family Renesas HS300x Sensor Control Module Software Integration System（R01AN6194）

The latest versions can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar.7.2023	-	First Edition
1.01	Mar.8.2024	4	Update of operational verification conditions
		14	Change to using the reset terminal in the Smart Configurator settings.

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 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.

Corporate Headquarters

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

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/.