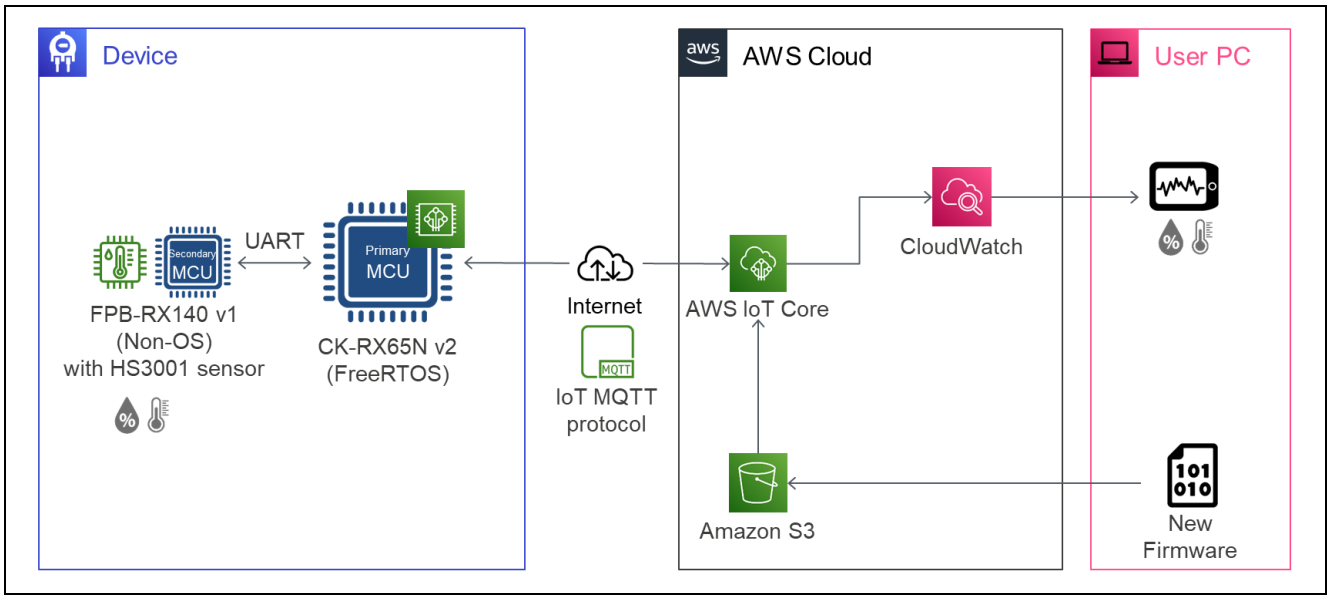
RX65N Group

Sample Code for OTA Update of a Secondary Device by Amazon Web Services with the Use of FreeRTOS

Introduction

This application note is for a system in which an RX65N microcontroller is used as a primary MCU that communicates with Amazon Web Services™ (hereafter, referred to as “AWS”) and an RX microcontroller is used as a secondary MCU that receives data measured by sensors. This application note describes a demonstration where AWS services are used to perform an over-the-air (OTA) update of the secondary MCU (hereafter, referred to as “secondary OTA update”).



Devices Used in Confirming Operation

- Primary MCU: RX65N
- Secondary MCU: RX140
- Temperature and humidity sensor: HS3001 high-performance relative humidity and temperature sensor

Boards Used in Confirming Operation

- Primary MCU: CK-RX65N v2 (RTK5CK65N0S04000BE)
- Secondary MCU: FPB-RX140 v1 (RTK5FP1400S00001BE)
- Temperature and humidity sensor: Relative Humidity Sensor Pmod™ Board (US082-HS3001EVZ)

Related Documents

This application note refers to and further explains information in the following documents. Updating of a document may lead to changes to the structure of chapters and other items. Take care on this point when referring to the following documents.

RX Family Firmware Update Module Using Firmware Integration Technology (R01AN6850)

RX Family Firmware Updating Communications Module Using Firmware Integration Technology (R01AN7757)

RX Family Firmware Update Software Development Guide using AWS/Azure QE for OTA (R20AN0712)

RX Family How to implement FreeRTOS OTA using Amazon Web Services in RX65N (for v202210.01-LTS-rx-1.1.0 or later) (R01AN7037)

RX65N Group CK-RX65N v2 User's Manual (R20UT5100)

FPB-RX140 v1 - User's Manual (R20UT5376)

(Download the latest versions from the Renesas Electronics Corp. website.)

Technical Updates and Technical News

(Download the latest versions from the Renesas Electronics Corp. website.)

Contents

1. Overview

This demonstration involves using a secondary OTA update to add a working sensor and confirming addition of the sensor data to be acquired by the display of sensor data in an AWS display in your browser.

IoT devices require the appropriate fixing of security vulnerabilities and updating of functions in response to customer requests. Implementing the secondary OTA update to supplement OTA updating of the primary MCU that has been provided in the past enables product development that supports measures against vulnerabilities in the secondary MCU and the updating of flexible services.

2. Conditions for Confirming Operation

The sample demonstration programs for this application note have been confirmed to operate correctly under the following conditions.

Table 2-1 Conditions for Confirming Demo Operation (RX65N)

Item	Description
MCU	RX65N (R5F565NEHDFB)
Board	CK-RX65N v2 (RTK5CK65N0S04000BE)
Operating voltage	3.3 V
RTOS	FreeRTOS v202210.01-LTS-1.3.1
Integrated development environment (IDE)	e² studio 2025-04 QE for OTA v2.2.0
C compiler	C/C++ Compiler Package for RX Family [CC-RX] v3.07.00 GCC for Renesas RX 8.3.0.202411
Flash memory programming tool	Renesas Flash Programmer V3.19.00

Table 2-2 Conditions for Confirming Demo Operation (RX140)

Item	Description
MCU	RX140 (R5F51406BGFN)
Board	FPB-RX140 v1 (RTK5FP1400S00001BE)
Operating voltage	3.3 V
Integrated development environment (IDE)	e² studio 2025-04 QE for OTA v2.2.0
C compiler	C/C++ Compiler Package for RX Family [CC-RX] v3.07.00 GCC for Renesas RX 8.3.0.202411
Flash memory programming tool	Renesas Flash Programmer V3.19.00
USB-UART converter	Pmod USBUART™

Table 2-3 Condition for Confirming Demo Operation (Sensor)

Item	Description
Temperature and humidity sensor board	US082-HS3001EVZ board
Conversion board	US082-INTERPEVZ

Note : The HS3001 sensor will reach end-of-life (EOL) on September 30, 2025. For replacement products and other details, please refer to the EOL notification document below. The demonstration in this application note can be performed without the sensor. The method for performing the demonstration without the sensor is described in “8 How to do the demo without using the HS3001 sensor.”
https://www.renesas.com/document/eln/plc-250010-end-life-eol-process-select-part-numbers

Table 2-4 Condition for Confirming Demo Operation (Others)

Item	Version
Python	3.12.6

QE for OTA is available at https://www.renesas.com/qe-ota/.

Python is available at https://www.python.org/.

3. Description of Hardware

The system consists of an RX65N microcontroller (primary MCU) that provides functionality for controlling communications with AWS and an RX140 microcontroller (secondary MCU) connected to the HS3001 sensor. The two microcontrollers communicate with each other via UARTs.

The system configuration is shown in Figure 3-1.

The CK-RX65N v2 (hereafter referred to as “CK-RX65N”) equipped with an RX65N microcontroller is used as the primary MCU.

The FPB-RX140 v1 (hereafter referred to as “FPB-RX140”) equipped with an RX140 microcontroller is used as the secondary MCU.

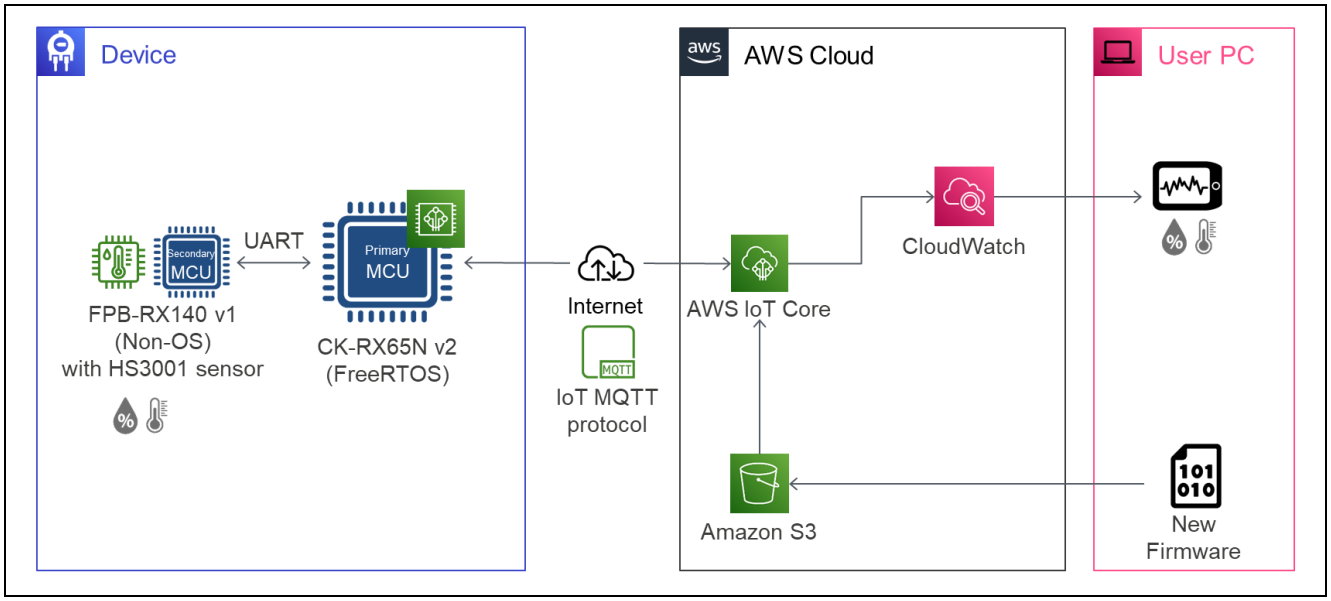


Figure 3-1 System Configuration of This Demo

4. Description of Software

4.1 Control methods for cloud connection and OTA

“FreeRTOS™ with IoT Library” is implemented in the RX65N firmware, which utilizes AWS-certified programs. This allows the use of AWS IoT Core and AWS IoT Device Management, which are managed services provided by AWS, to perform OTA firmware updating and data uploading to the cloud via MQTT communications.

The RX65N microcontroller on the primary MCU side uses the AWS IoT Over-the-air Update Library to control OTA updating of the secondary MCU. The update firmware for the secondary MCU, which is received from AWS, is transferred to the secondary MCU, where the firmware update is applied.

To communicate data between the primary MCU and secondary MCU, use the “RX Family Firmware Updating Communications Module Using Firmware Integration Technology”

The RX microcontroller on the secondary MCU side uses “RX Family Firmware Update Module Using Firmware Integration Technology” to control firmware updating of the secondary MCU.

4.2 Firmware Update Methods

The mechanism for firmware update in the secondary MCU of this sample program uses “linear mode partial update method” among the methods provided by the firmware update module. For details on this method, refer to “linear mode partial update method” in “1.3 Firmware Update Operation” in “RX Family Firmware Update Module Using Firmware Integration Technology”.

Operations for the secondary OTA update are summarized in Figure 4-1. The states of the ROM in each phase during updating are shown in Figure 4-2. Note that the red frames in Figure 4-2 indicate the programs under execution at the given times.

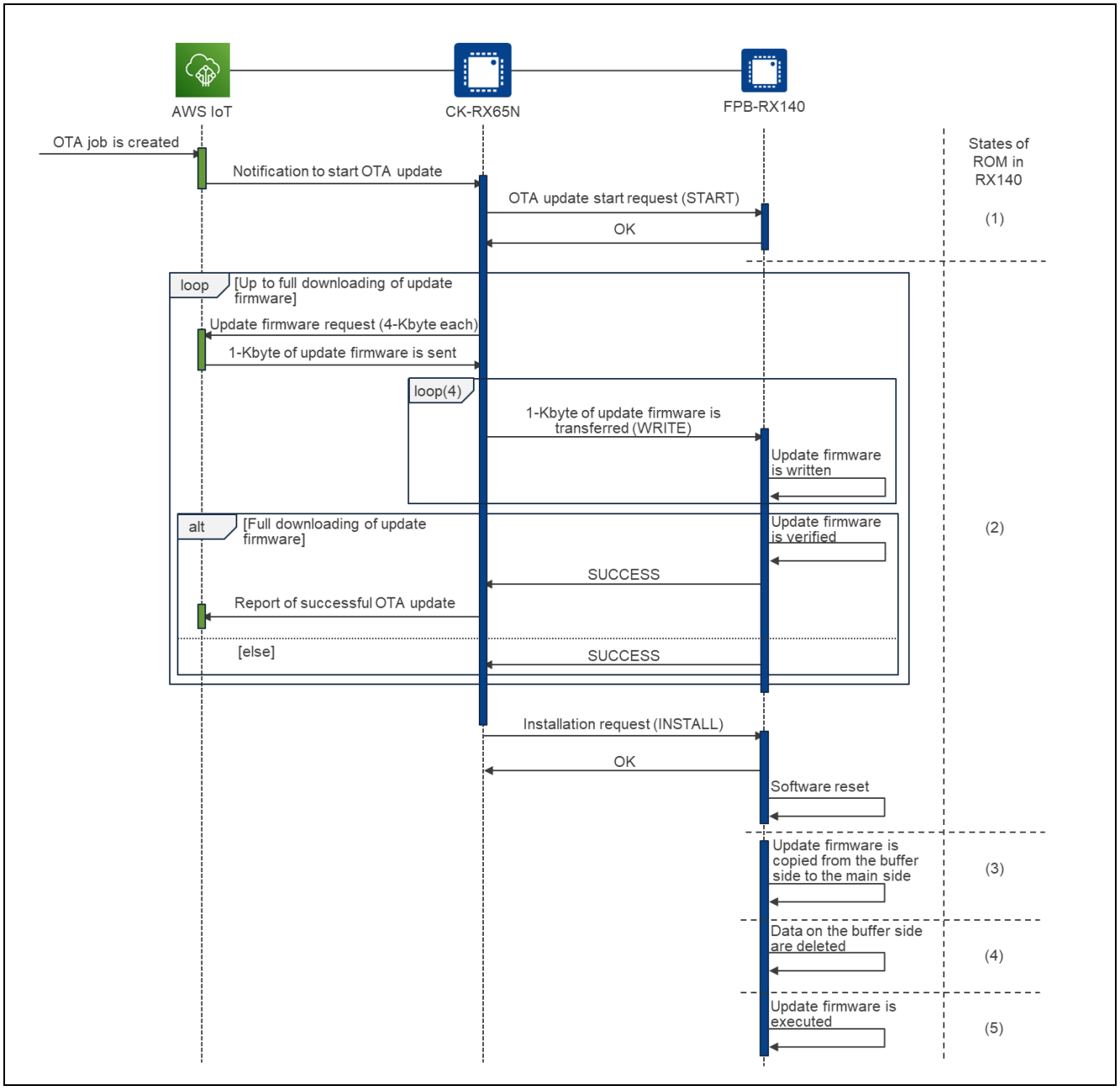


Figure 4-1 Overview of Operations for the Secondary OTA Update

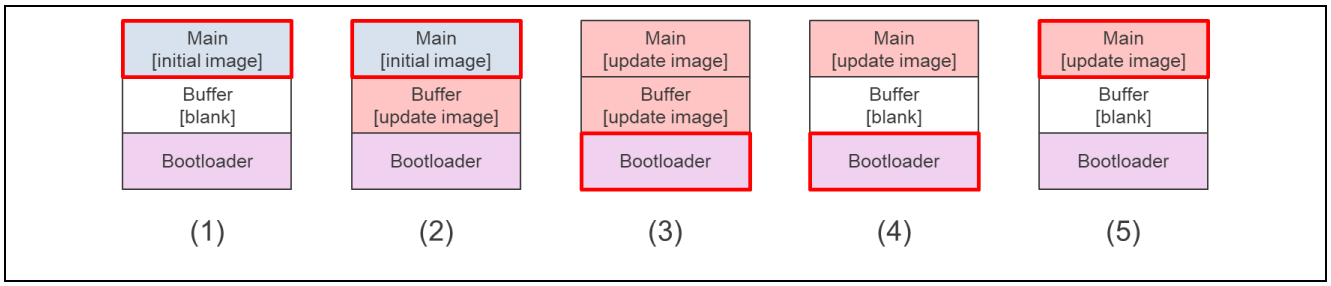


Figure 4-2 States of ROM in the Secondary MCU during Updating

4.3 UART Communications between the Microcontrollers

To communicate data between the primary MCU and secondary MCU, use the "RX Family Firmware Updating Communications Module Using Firmware Integration Technology".

Communications related to secondary OTA update (communications between CK-RX65N and FPB-RX140 in the sequence diagram in Figure 4-1) use the commands from the FWUP command class.

In addition, the DATA_RECV command from the Common command class is used to receive sensor data from the secondary MCU, and when the command argument is 1, it is used as a sensor data transmission request.

For details on each command, refer to "1.6 Specifications of Commands" in "RX Family Firmware Updating Communications Module Using Firmware Integration Technology".

4.3.1 Settings of UART Communications

The UART communications settings are shown in Table 4-1.

Table 4-1 Settings of UART Communications between the Microcontrollers

Item	Setting
Data length	8 bits
Parity bit	None
Stop bit	1 bit
Flow control	None
Bitrate	1 Mbps

4.4 Folder/File Structure

Figure 4-3 shows the folder/file structure.

```
r01an6220xx0300-rx-2nd-ota-apl
├─Demo
│   ├─afr-v202210.01-LTS-rx-1.3.1
│   ├─ccrx
│   │   ├─demo_rx65n_ck_primary
│   │   ├─demo_bl_rx65n_ck_primary
│   │   ├─demo_app_rx140_fpb_w_buffer
│   │   └─bootloader_rx140_fpb_w_buffer
│   └─gcc
│       ├─demo_rx65n_ck_primary
│       ├─demo_bl_rx65n_ck_primary
│       ├─demo_app_rx140_fpb_w_buffer
│       └─bootloader_rx140_fpb_w_buffer
├─r01an6220ej0300-rx-2nd-ota-apl.pdf
└─r01an6220jj0300-rx-2nd-ota-apl.pdf
```

Figure 4-3 Folder/File Structure

The demo_rx65n_ck_primary folder and demo_bl_rx65n_ck_primary folder contain project files for the CK-RX65N.

The demo_app_rx140_fpb_w_buffer folder and bootloader_rx140_fpb_w_buffer folder contain project files for the FPB-RX140.

The projects for the CK-RX65N and the FPB-RX140 support the CC-RX and GCC compilers.

4.5 Code Size

The ROM and RAM sizes for projects included in the sample code of this application note are listed in the tables below. The values in the tables were confirmed under the following conditions.

- CC-RX
  - Compiler
    Optimization level (-optimize): Level 2: Performs whole module optimization
    Optimization type (-speed/-size): Optimizes with emphasis on code size
  - Linker
    Optimization type (-nooptimize/-optimize): All
  - Library Generator
    Optimization level (-optimize): Level 2: Performs whole module optimization
    Optimization type (-speed/-size): Optimizes with emphasis on code size

Table 4-2 Code Size (CC-RX)

Project	ROM	RAM
demo_bl_rx65n_ck_primary	33 Kbytes	8 Kbytes
demo_rx65n_ck_primary	601 Kbytes	382 Kbytes
bootloader_rx140_fpb_w_buffer	29 Kbytes	6 Kbytes
demo_app_rx140_fpb_w_buffer	50 Kbytes	13 Kbytes

- GCC
  Optimization level: Optimize for debug (-Og)

Table 4-3 Code Size (GCC)

Project	ROM	RAM
demo_bl_rx65n_ck_primary	51 Kbytes	10 Kbytes
demo_rx65n_ck_primary	654 Kbytes	383 Kbytes
bootloader_rx140_fpb_w_buffer	24 Kbytes	13 Kbytes
demo_app_rx140_fpb_w_buffer	41 Kbytes	12 Kbytes

5. Operations of the Demonstration

(1) In the initial state of the demonstration, the FPB-RX140 only acquires humidity data by using the HS3001 sensor.

(2) The secondary OTA update mechanism is used to download the update firmware for the FPB-RX140 from AWS via the CK-RX65N and then update the firmware.

(3) After the firmware updating is complete, the FPB-RX140 acquires temperature data in addition to humidity data from the HS3001 sensor.

In this sequence, the type of sensor data from which data are being acquired and the values can be checked from the log output from both microcontrollers to your PC and from the dashboard on AWS.

6. Setting up the Demonstration

This section describes the setting up required to run the demonstration covered by this application note.

The necessary steps are setting up the hardware, such as the wiring of the CK-RX65N and FPB-RX140 and connection of the HS3001 sensor, setting up the software, such as creating and writing the initial firmware for each microcontroller board, and the preparation on the AWS cloud side for execution of the OTA update and display of the sensor data from AWS.

6.1 Setting up the Hardware

Firstly, the overall hardware structure for this demonstration is shown below. For an actual image after setup is complete, see Figure 6-2. The methods for setting up each of the boards are described in detail in the subsequent subsections.

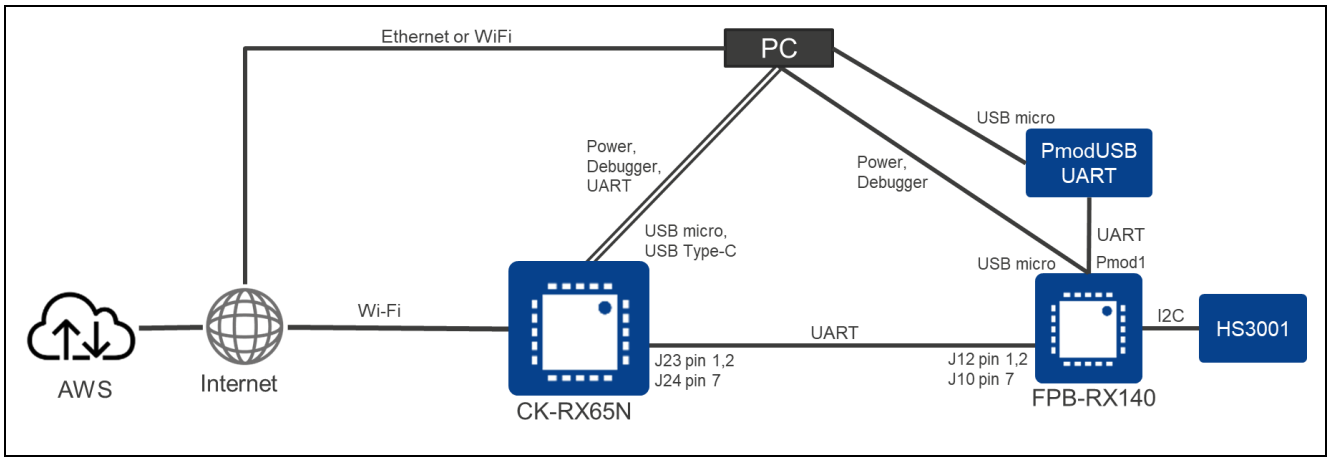


Figure 6-1 Overall Hardware Structure for This Demo

6.1.1 Setting up the CK-RX65N

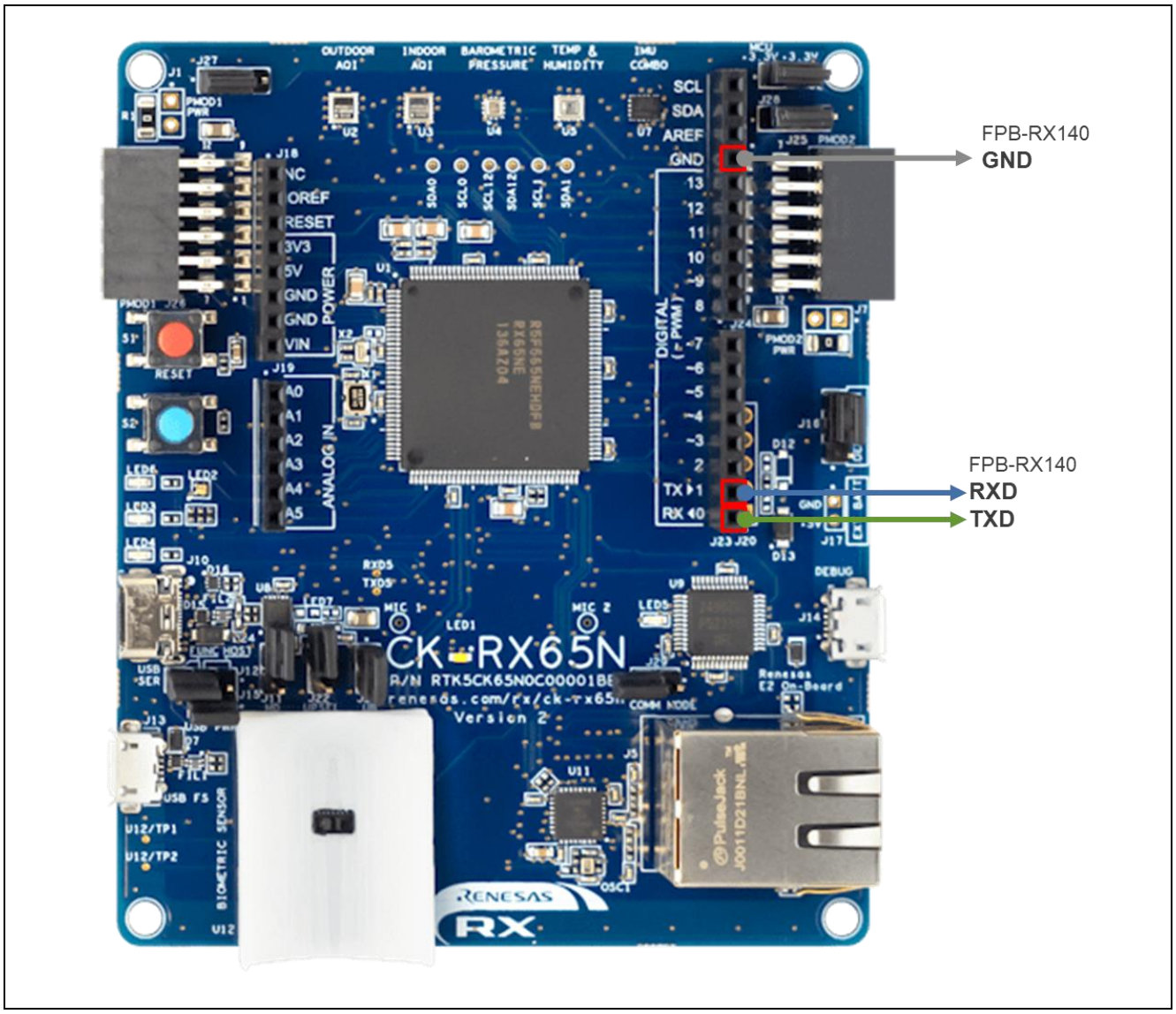
The procedure for setting up the CK-RX65N is described in the following passages.

(1) Connecting the cable for UART communications with the FPB-RX140

TXD, RXD, and GND for UART communications with the FPB-RX140 are allocated to the following pins on the J23 and J24 connectors of the CK-RX65N. Connect the pins on the FPB-RX140 side as described in 6.1.2(2), with the corresponding UART signals listed in the table below.

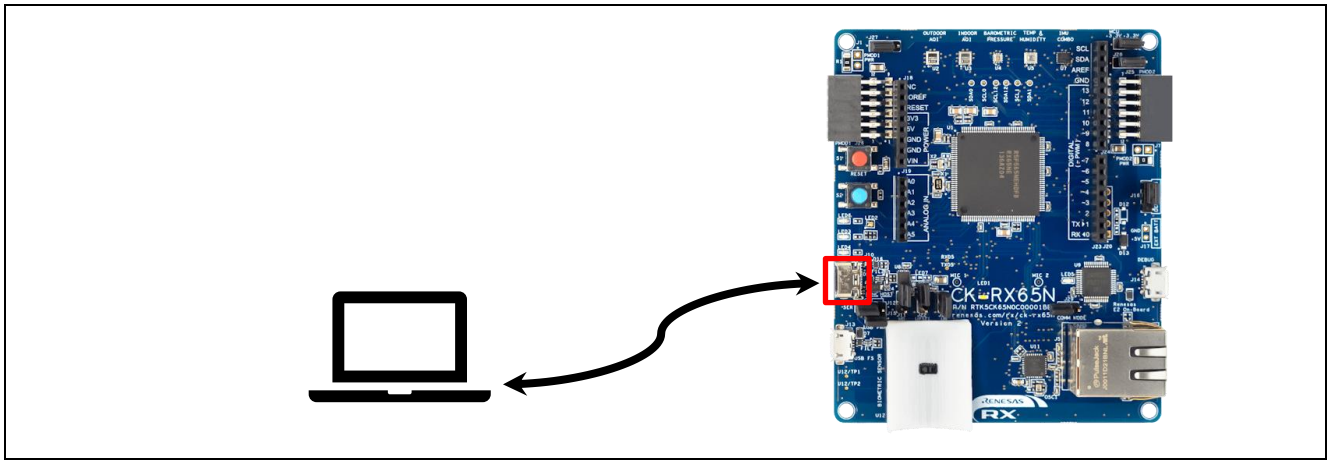
Table 6-1 UART Connection Method between the Microcontrollers (CK-RX65N ↔ FPB-RX140)

CK-RX65N		FPB-RX140
J23 Pin 1: D0/RX	⇔	J12 Pin 2: D1/TX
J23 Pin 2: D1/TX	⇔	J12 Pin 1: D0/RX
J24 Pin 7: GND	⇔	J10 Pin 7: GND



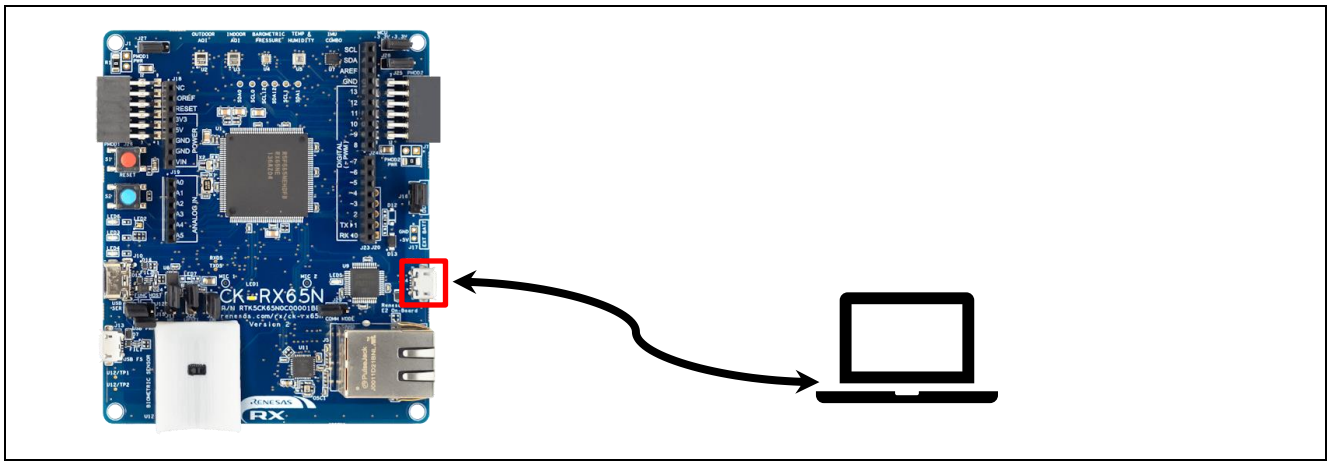
(2) Connecting the cable for log output to the PC

Connect the PC to the USB serial connector (USB Type-C) on the CK-RX65N with a USB cable.



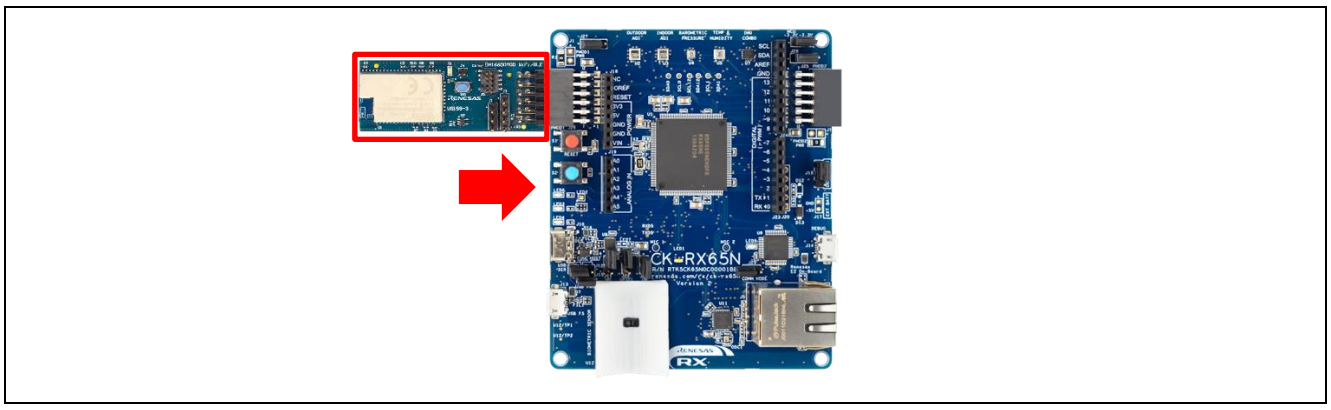
(3) Connecting the power supply and debugger

Connect the PC to the E2OB Debugger connector (micro USB Type-B) on the CK-RX65N with a USB cable.



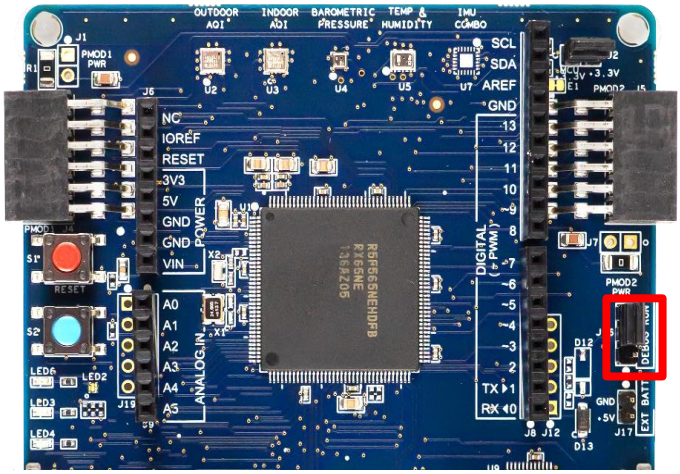
(4) Connecting the DA16600 Wi-Fi module

Connect the DA16600 Pmod module to the Pmod1 connector on the CK-RX65N.



(5) Closing jumper block J16 on the DEBUG side

To set the CK-RX65N to debug mode, close jumper block J16 on the DEBUG side (pins 1-2).

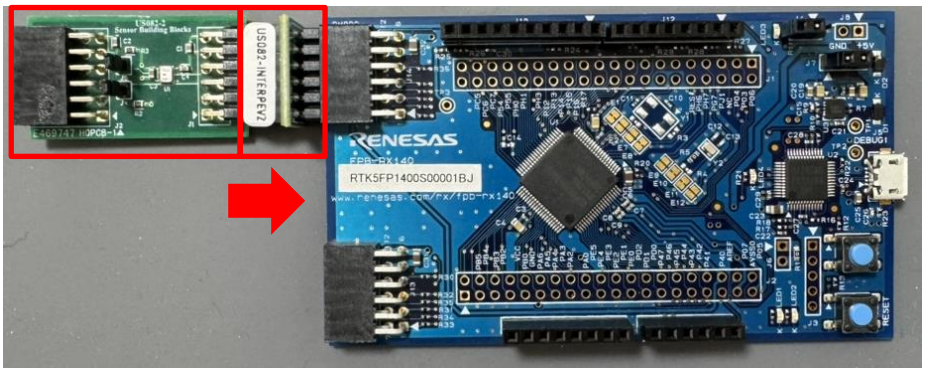


6.1.2 Setting up the FPB-RX140

The procedure for setting up the FPB-RX140 is described in the following passages.

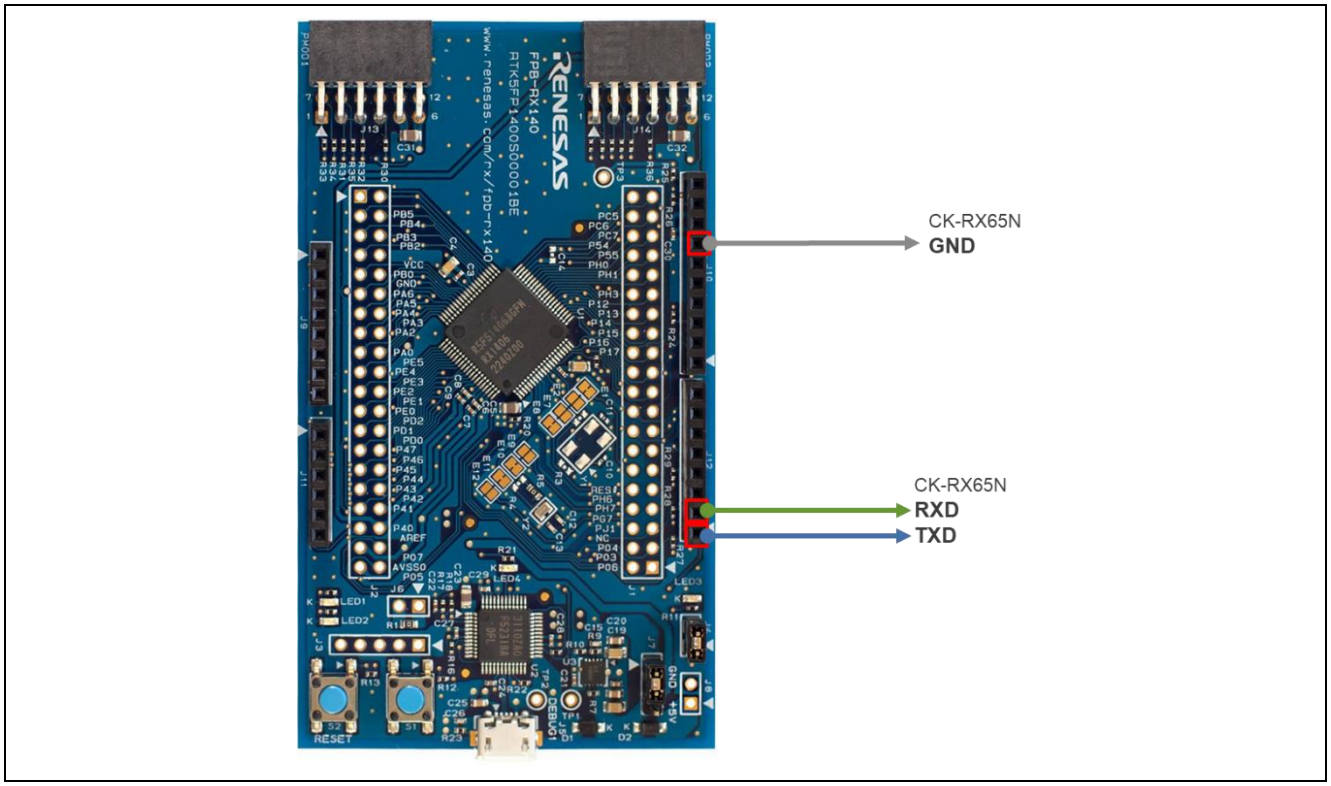
(1) Connecting the HS3001 board

Connect the HS3001 board and the conversion board to the Pmod2 connector on the FPB-RX140.



(2) Connecting the cable for UART communications with the CK-RX65N

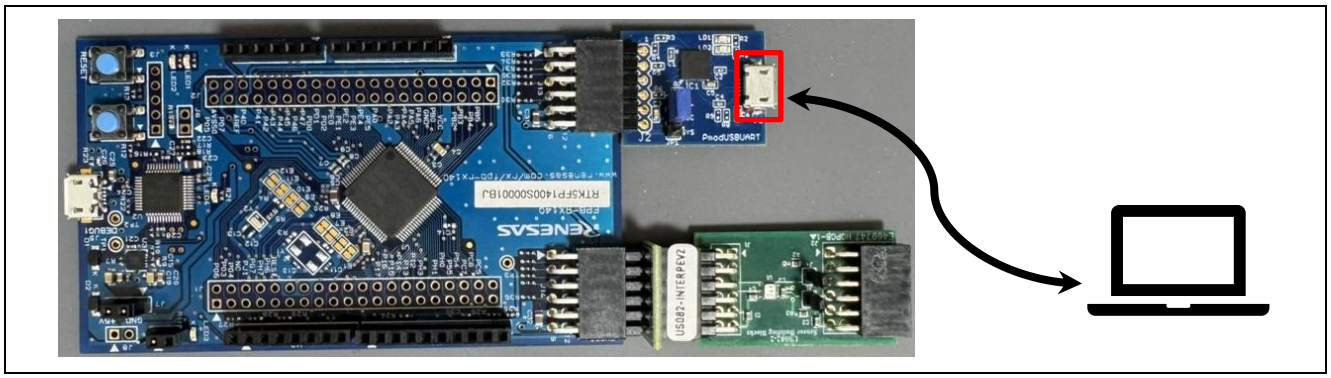
TXD, RXD, and GND for UART communications with the CK-RX65N are allocated to the following pins on the J12 and J10 connectors of the FPB-RX140. Connect the pins on the CK-RX65N as described in 6.1.1(1), with the corresponding UART signals listed in Table 6-1.



(3) Connecting the cable for serial communications to be used in log output to the PC

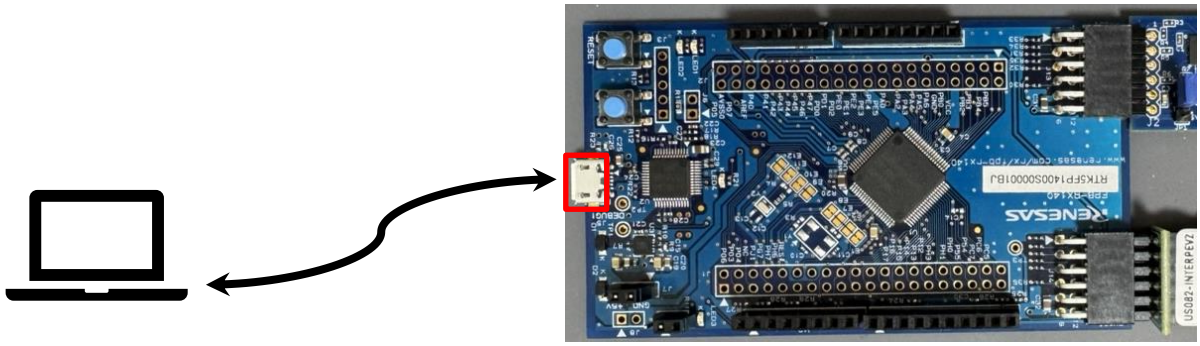
Connect the Pmod USBUART conversion board to the pins 1-6 of the Pmod1 connector on the FPB-RX140.

Also, connect the PC and the micro USB Type-B connector on the Pmod USB-to-UART converter with a USB cable.



(4) Connecting the cable for power supply

Connect the PC and the micro USB Type-B connector on the FPB-RX0 with a USB cable.



(5) Opening the emulator reset header (J4)

Open the emulator reset header (J4) on the FPB-RX140.

The hardware setup for the demonstration is now completed.

Figure 6-2 is an image of the overall configuration for the demonstration.

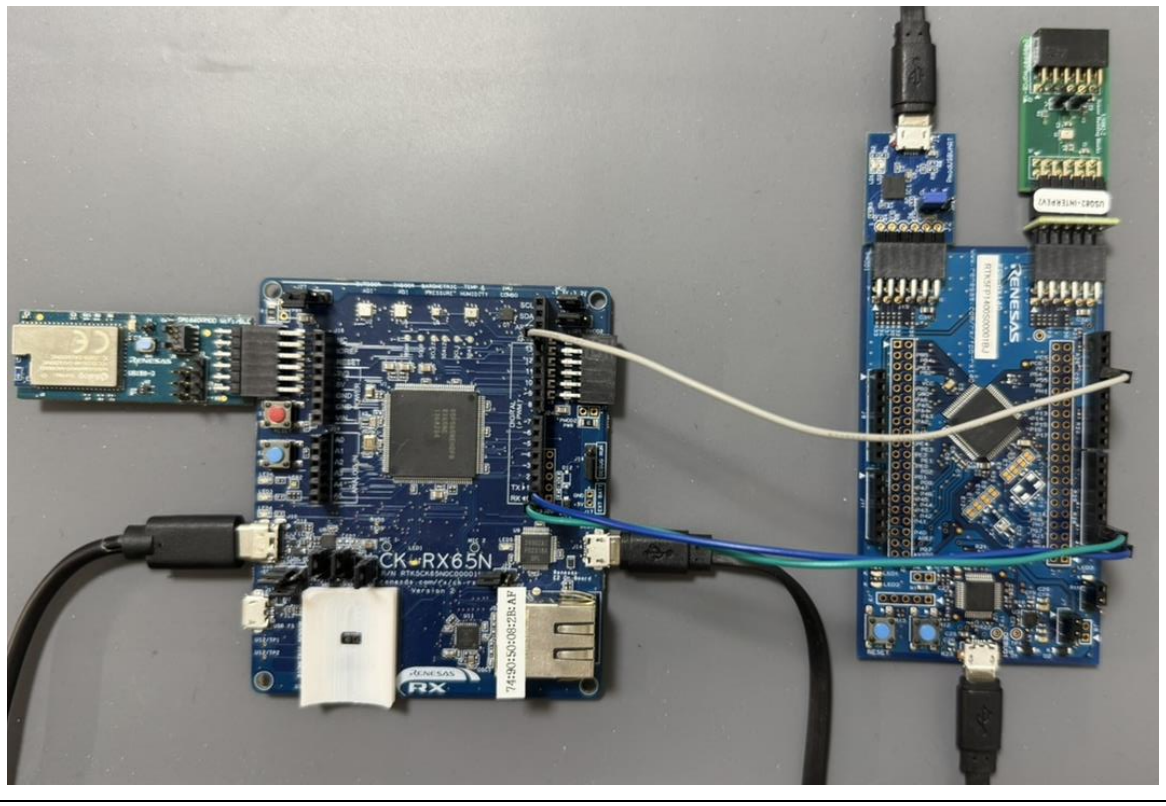


Figure 6-2 Image of the Overall Configuration for the Demo

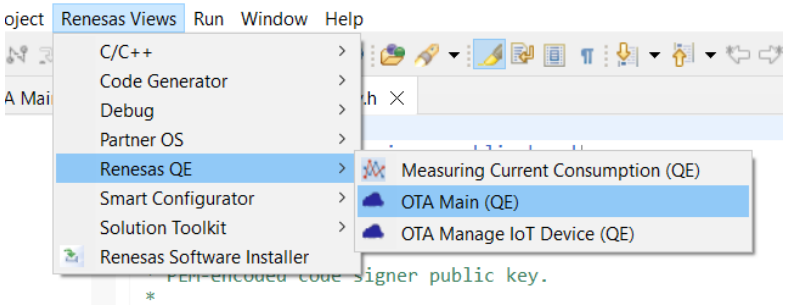
6.2 Setting up the Software

6.2.1 Advance Preparation

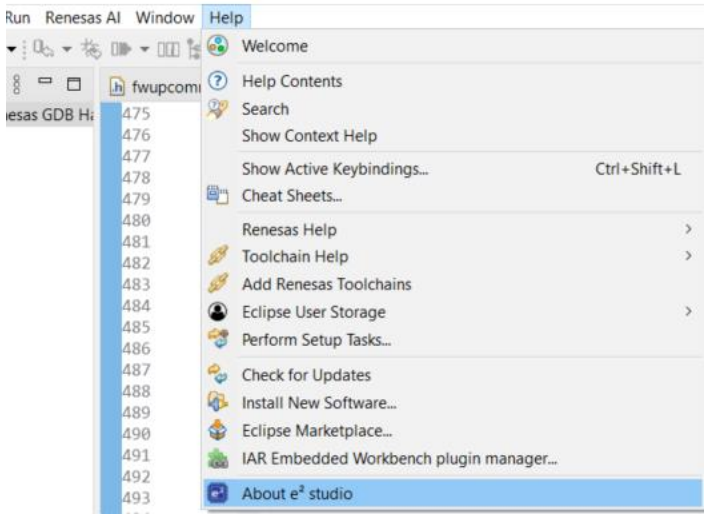
For each of the software versions used in confirming operation, see Table 2-1, Table 2-2, Table 2-3 and Table 2-4.

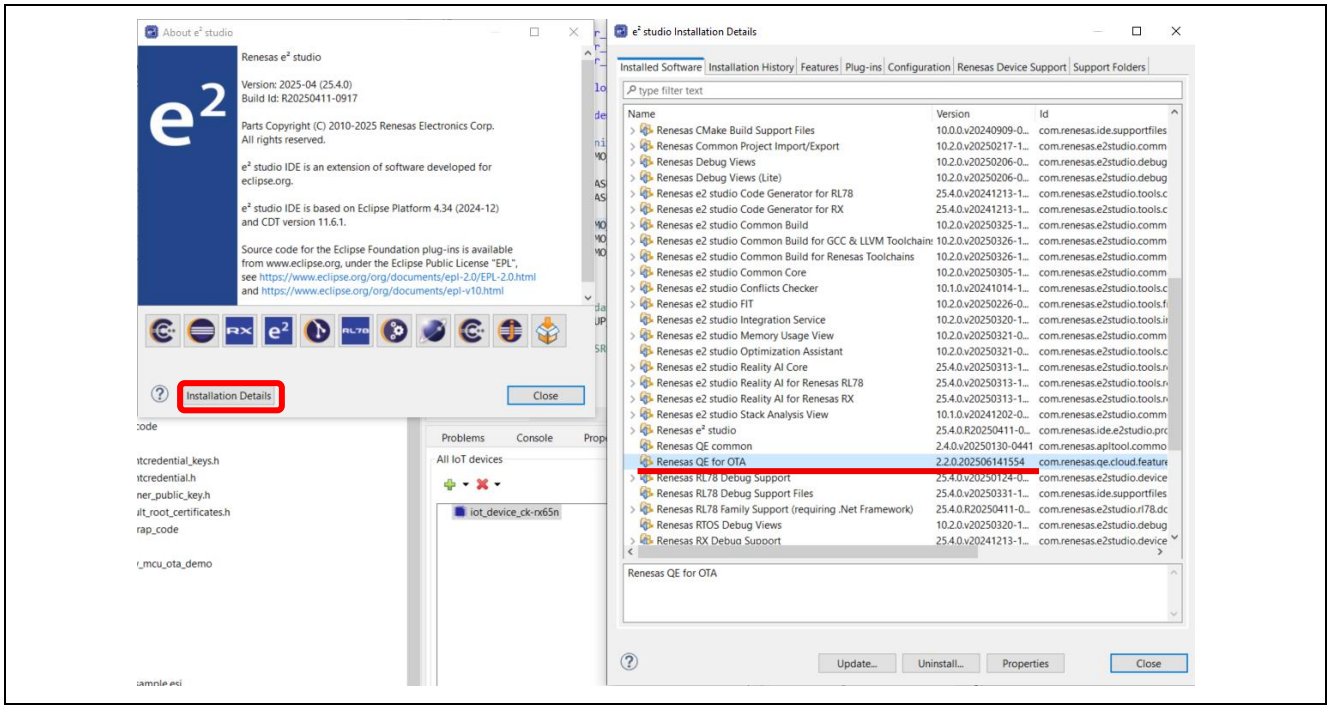
(1) Installing QE for OTA

From the e^2 studio menu bar, select [Renesas Views] → [Renesas QE] to check whether QE for OTA is installed. If [OTA Main (QE)] and [OTA Manage IoT Device (QE)] are displayed, installation is completed.



Regarding the version, from the e^2 studio bar, select [Help] → [About e2 studio] → [Installation Details] to confirm that the version of “Renesas QE for OTA” is “2.2.0.~” or higher.





If they are not displayed, refer to “2.1 Install QE for OTA” in “RX Family Firmware Update Software Development Guide using AWS/Azure QE for OTA (R20AN0712)” and install QE for OTA.

(2) Installing the Python execution environment

Python can be downloaded from https://www.python.org/.

The pycryptodome library of Python is also to be used. After installing Python, execute the follow pip command to install it.

```
> pip install pycryptodome
```

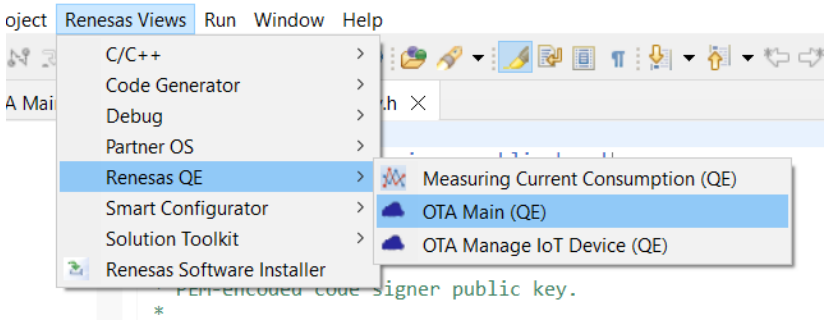
(3) Installing the Renesas Flash Programmer

The Renesas Flash Programmer can be downloaded from Renesas Flash Programmer (Programming GUI) | Renesas.

6.2.2 Logging in to AWS with QE for OTA

(1) Opening the QE for OTA window

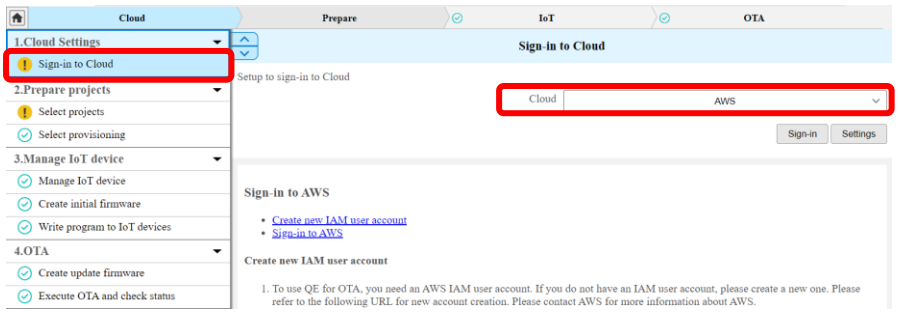
From the e^2 studio menu bar, select [Renesas Views] → [Renesas QE] → [OTA Main (QE)].



(2) <QE for OTA> [1. Cloud Settings] → [Sign-in to Cloud]

From here, follow the steps displayed in the GUI window of QE for OTA.

Start by selecting “AWS” for [Cloud] and sign in. An AWS resource is generated in the region selected at the time of login.



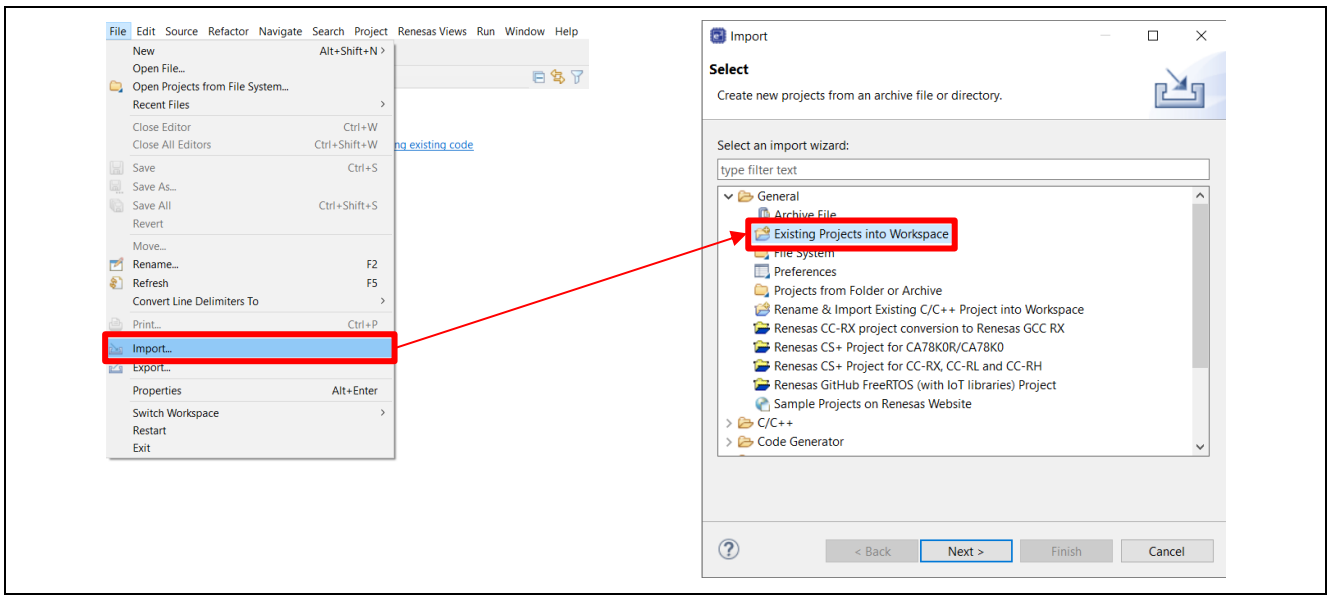
6.2.3 Creating and Running the Initial Firmware for the CK-RX65N

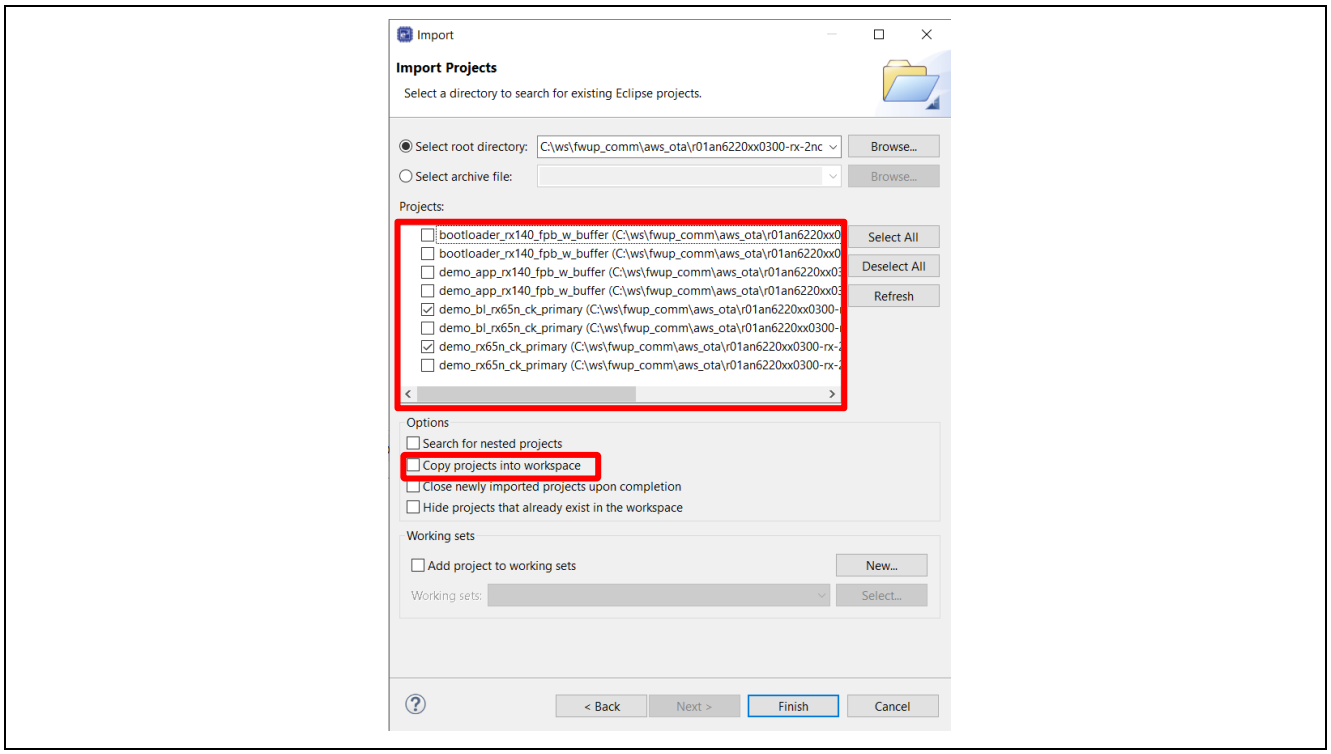
Create initial firmware for the CK-RX65N by using QE for OTA, then writing and running it. The procedure is described below.

(1) Importing projects

Import the “demo_bl_rx65n_ck_primary” project, a bootloader for the CK-RX65N, and the “demo_rx65n_ck_primary” project, a user program, into e^2 studio. There are CC-RX and GCC versions of the CK-RX65N project, but here we will use the CC-RX version.

When importing projects, uncheck [Copy projects into workspace] in the [Options] field.



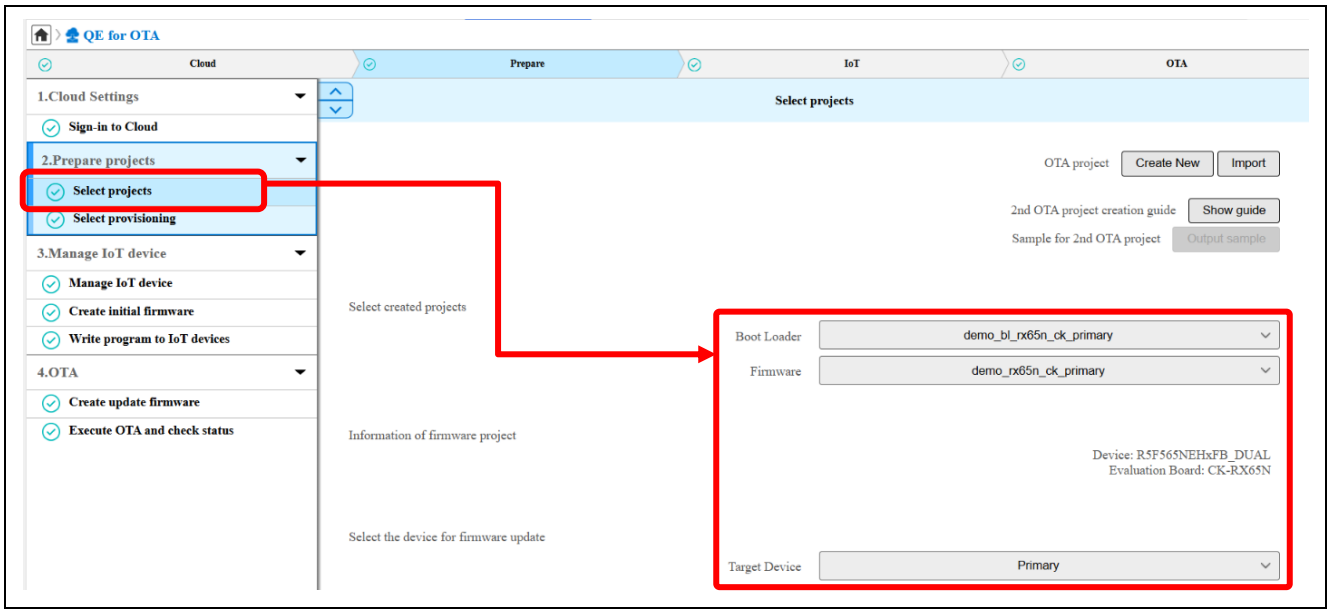


(2) Entering Wi-Fi connection information

Enter the SSID of the Wi-Fi access point you want to use in `clientcredentialWIFI_SSID` and the password in `clientcredentialWIFI_PASSWORD`, as defined in "src/application_code/include/aws_clientcredential.h" in the demo_rx65n_ck_primary project.

(3) Selecting projects

Select the ck_rx65n_demo_bootloader project and the ck_rx65n_2ndota_demo project that were imported into e² studio earlier. Also, select "Primary" for Target Device.



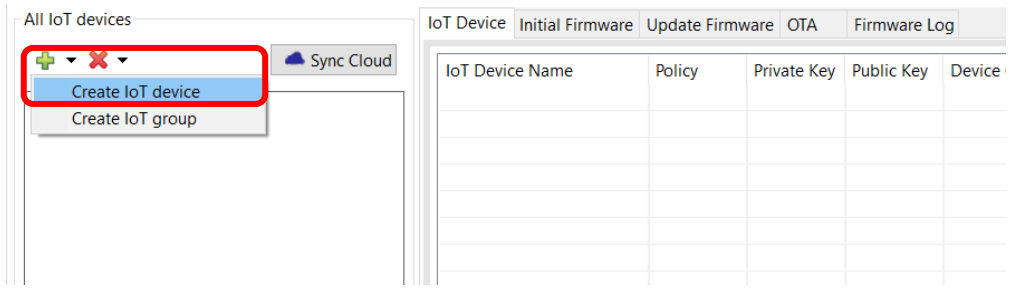
(4) Selecting the provisioning method

Select "Source code includes credentials (asymmetric keys)" as the provisioning method.

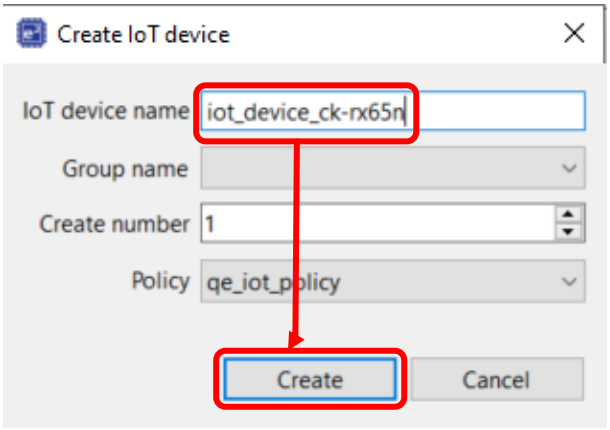
(5) Creating an IoT device

Click “Open view” in Manage IoT device to open the “OTA Manage IoT Device (QE)” screen.

Click the “+” button under “All IoT devices” and select “Create IoT device.”

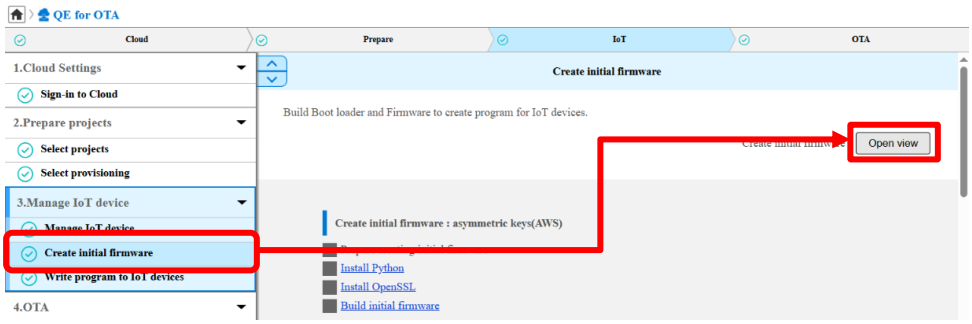


Enter an appropriate name in the “IoT device name” field of the dialog box, and then click “Create.”

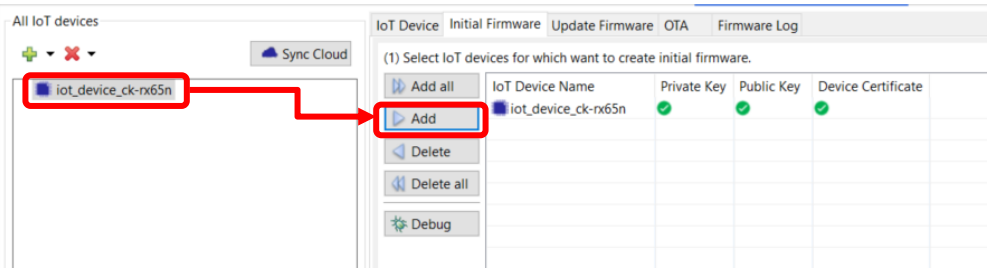


(6) Creating the initial firmware

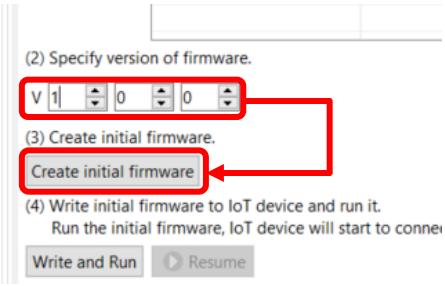
Click “Open view” in Create initial firmware.



In the “Initial Firmware” tab, select the IoT device you created earlier and click “▶Add”.



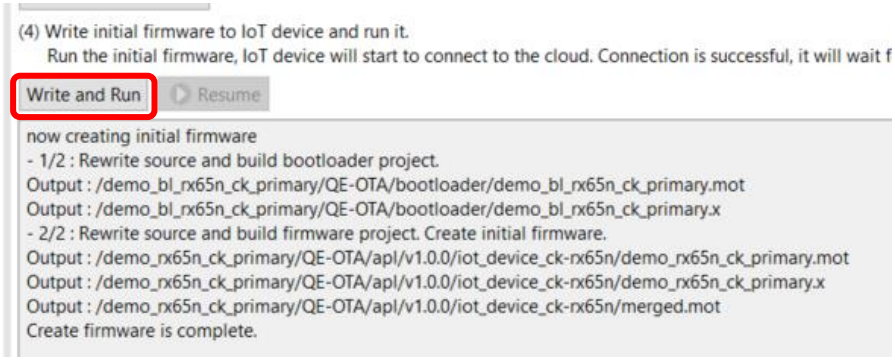
Enter “1.0.0” for the version and click “Create initial firmware.”



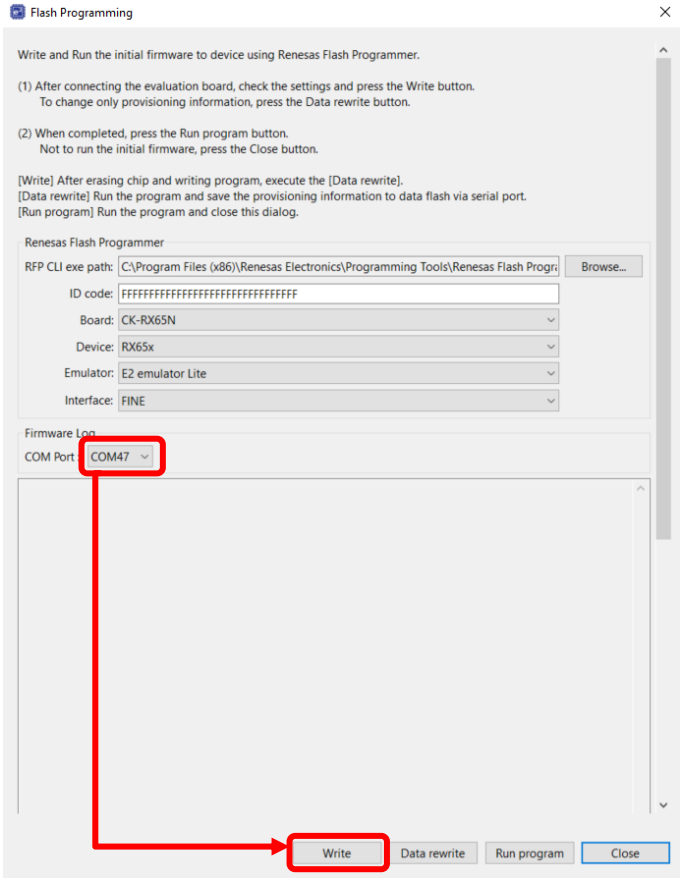
The demo_bl_rx65n_ck_primary project and demo_rx65n_ck_primary project builds are executed, and if “Create firmware is complete.” is displayed in the “Initial Firmware” tab, the process is successful.

(7) Writing and running the initial firmware to the CK-RX65N

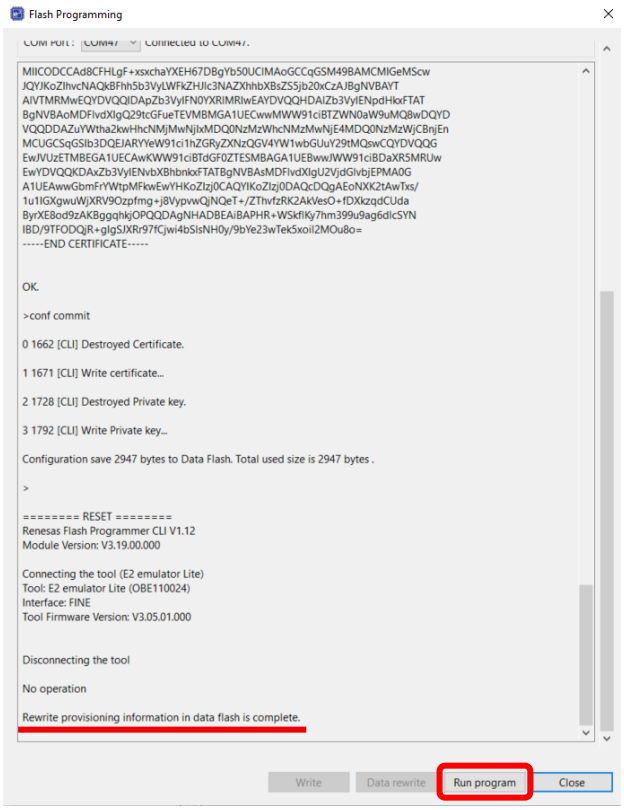
Click “Write and Run” in the “Initial Firmware” tab.



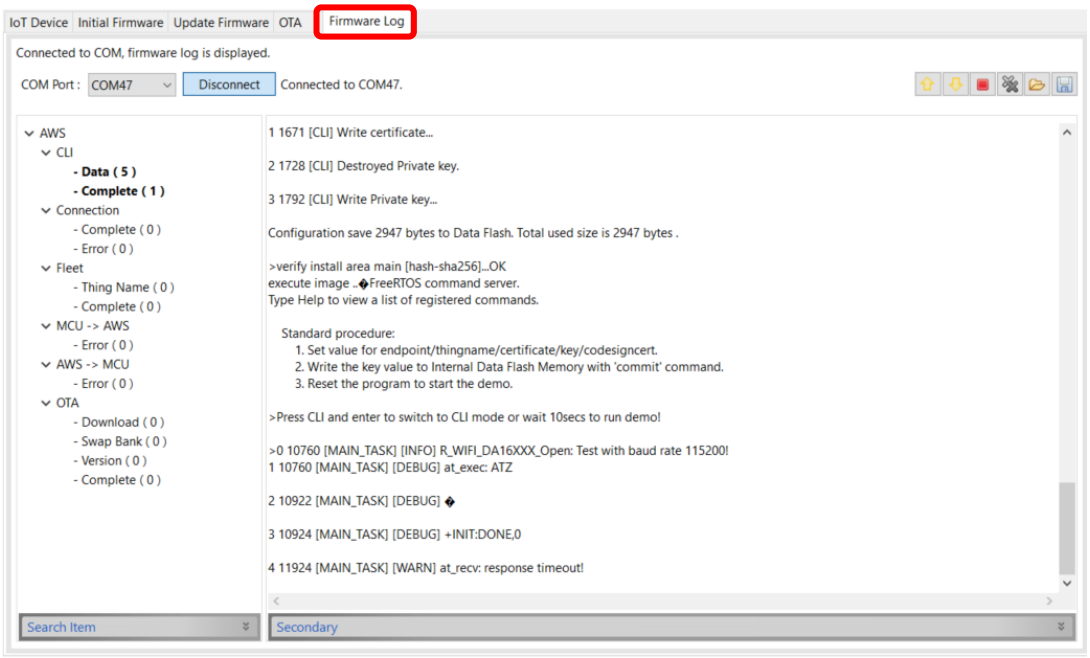
Select the COM port number for the CK-RX65N and click “Write.” Each parameter for the Renesas Flash Programmer will be automatically entered if the CK-RX65N onboard emulator is connected. If any fields are blank, check that the CK-RX65N is connected to the PC and that jumper J16 is connected to the DEBUG side.



If “Rewrite provisioning information in data flash is complete.” is displayed, the operation was successful. Click “Run program.”

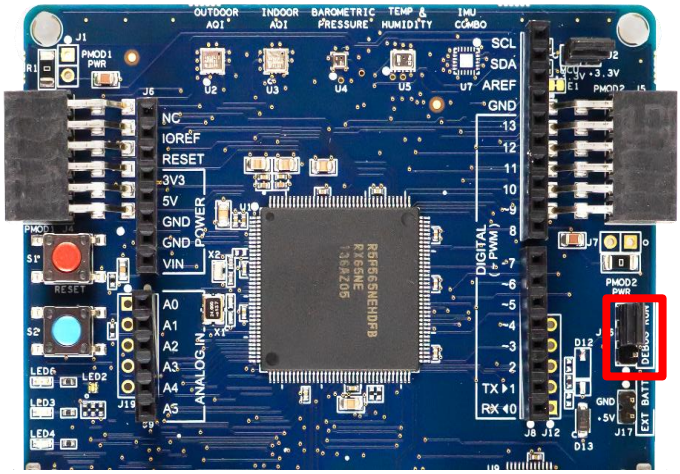


Open the “Firmware Log” tab. Here you can see the logs output from the CK-RX65N.



(8) Closing jumper block J16 on the RUN side

Close jumper block J16 on the CK-RX65N on the RUN side (pins 2-3).



6.2.4 Creating and Running the Initial Firmware for the FPB-RX140

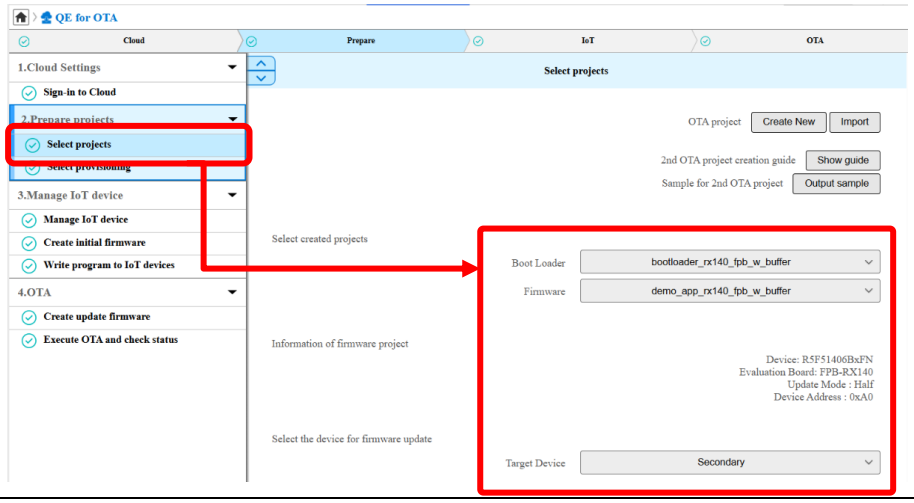
Create initial firmware for the FPB-RX140 by using QE for OTA, then writing and running it. The procedure is described below.

(1) Importing projects

Similarly to the procedure for importing projects for the CK-RX65N described in the previous subsection, import the “bootloader_rx140_fpb_w_buffer” project, a bootloader for the FPB-RX140, and the “demo_app_rx140_fpb_w_buffer” project, a user program, into e² studio.

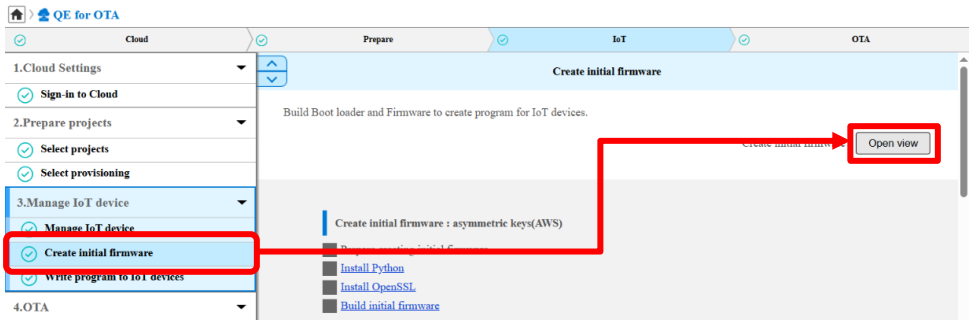
(2) Selecting projects

Select the bootloader_rx140_fpb_w_buffer project and the demo_app_rx140_fpb_w_buffer project that were imported into e² studio earlier. Also, select “Secondary” for Target Device.

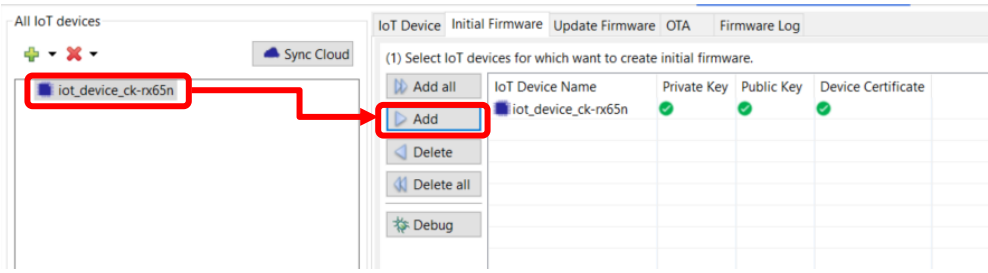


(3) Creating the initial firmware

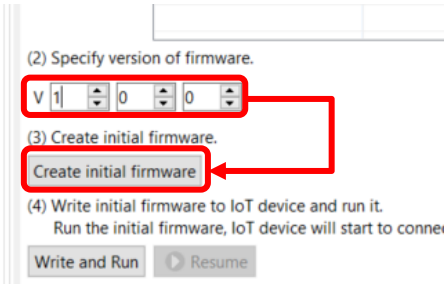
Click “Open view” in Create initial firmware to open the “OTA Manage IoT Device (QE)” screen.



In the “Initial Firmware” tab, select the IoT device created 6.2.3(5) and click “▶Add”.



Enter “1.0.0” for the version and click “Create initial firmware.”

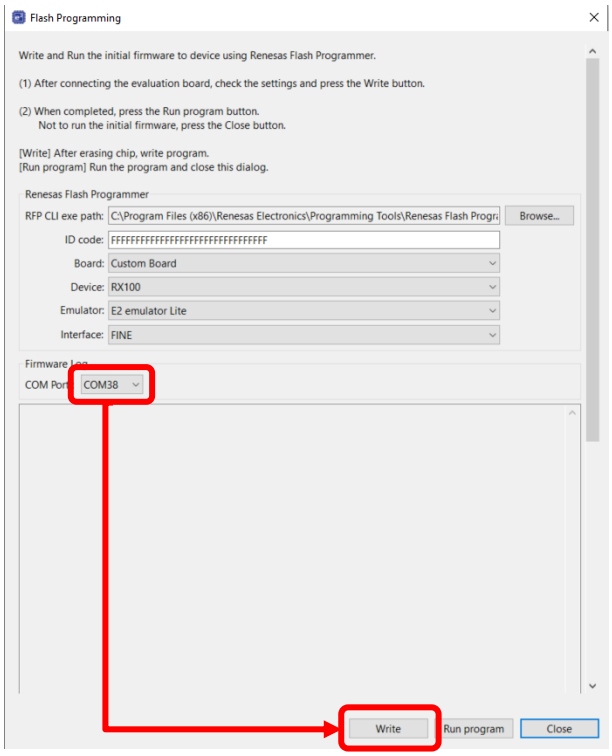


The bootloader_rx140_fpb_w_buffer project and demo_app_rx140_fpb_w_buffer project builds are executed, and if “Create firmware is complete.” is displayed in the “Initial Firmware” tab, the process is successful.

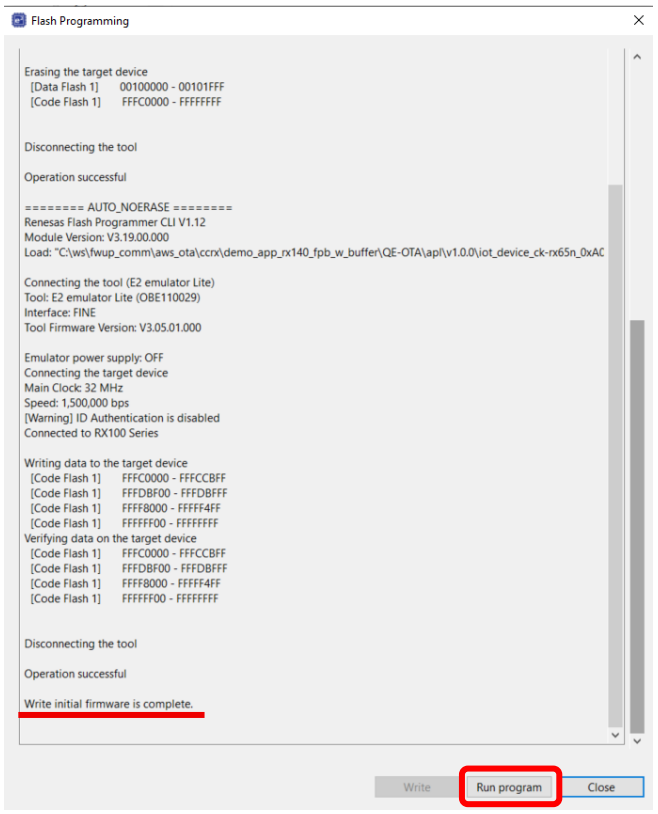
(4) Writing and running the initial firmware to the FPB-RX140

Click “Write and Run” in the “Initial Firmware” tab.

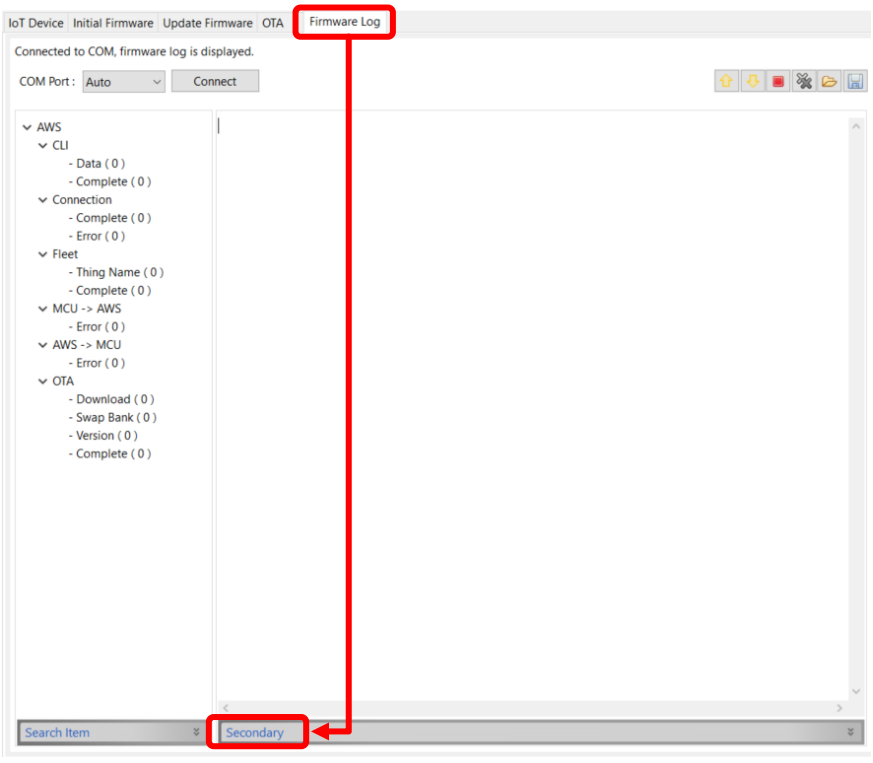
Select the COM port number for the Pmod USBUART conversion board connected to the FPB-RX140 and click “Write.” Each parameter for the Renesas Flash Programmer will be automatically entered if the FPB-RX140 onboard emulator is connected. If any fields are blank, check that the FPB-RX140 is connected to the PC and that jumper J4 is open.



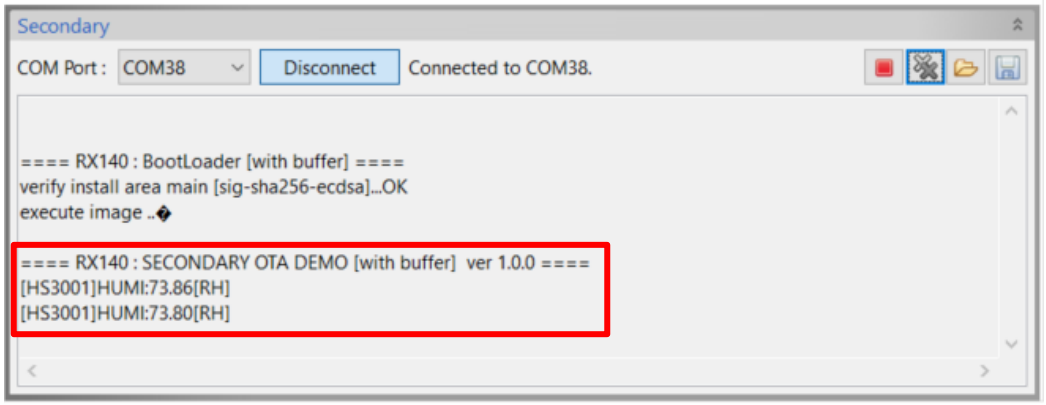
If “Write initial firmware is complete.” is displayed, the operation was successful. Click “Run program.”



Open the “Firmware Log” tab and click “Secondary.”



Here you can see the logs output from the FPB-RX140. If the humidity data measured by the HS3001 sensor is displayed after “ver 1.0.0” is displayed, the operation is successful.



6.3 Preparations for Displaying the Sensor Data Using the AWS Cloud

To display the received sensor data in a graphical format, set up Amazon CloudWatch and AWS IoT Core through the following steps.

Note: If you do not need to display the data in a graphical format, but only need to confirm in your browser that the data have been received by AWS, you can omit the entire procedure in 6.3.
In this case, as shown in Figure 6-3, you can subscribe to "`iotdemo/topic/sensor`" in [MQTT test client] of AWS IoT to confirm in a text format that the sensor data are being received as expected.

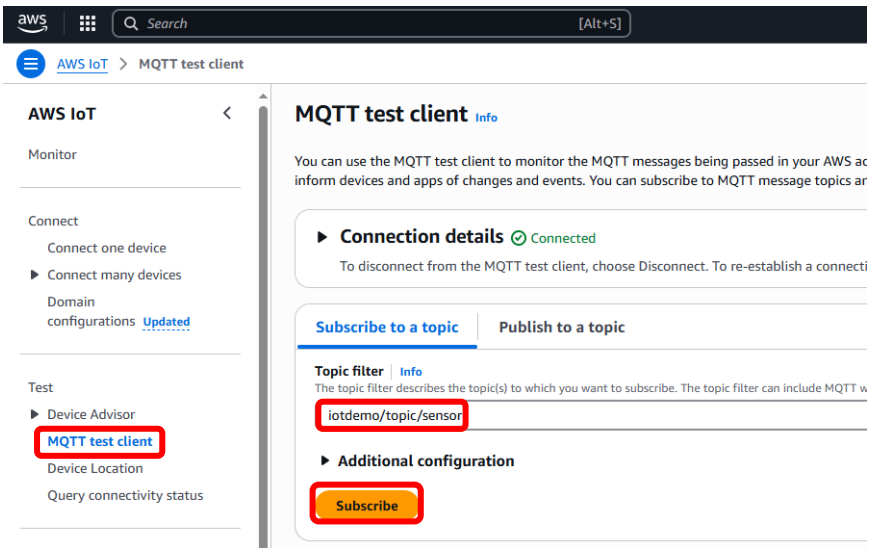


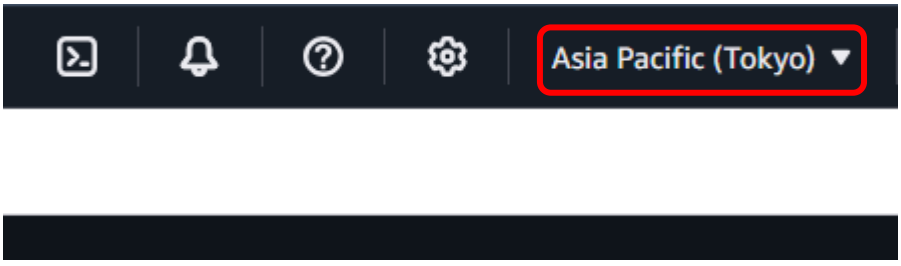
Figure 6-3 Confirming Data Reception by the MQTT Test Client

(1) Logging in to the AWS Management Console

Start by logging in to the AWS Management Console.

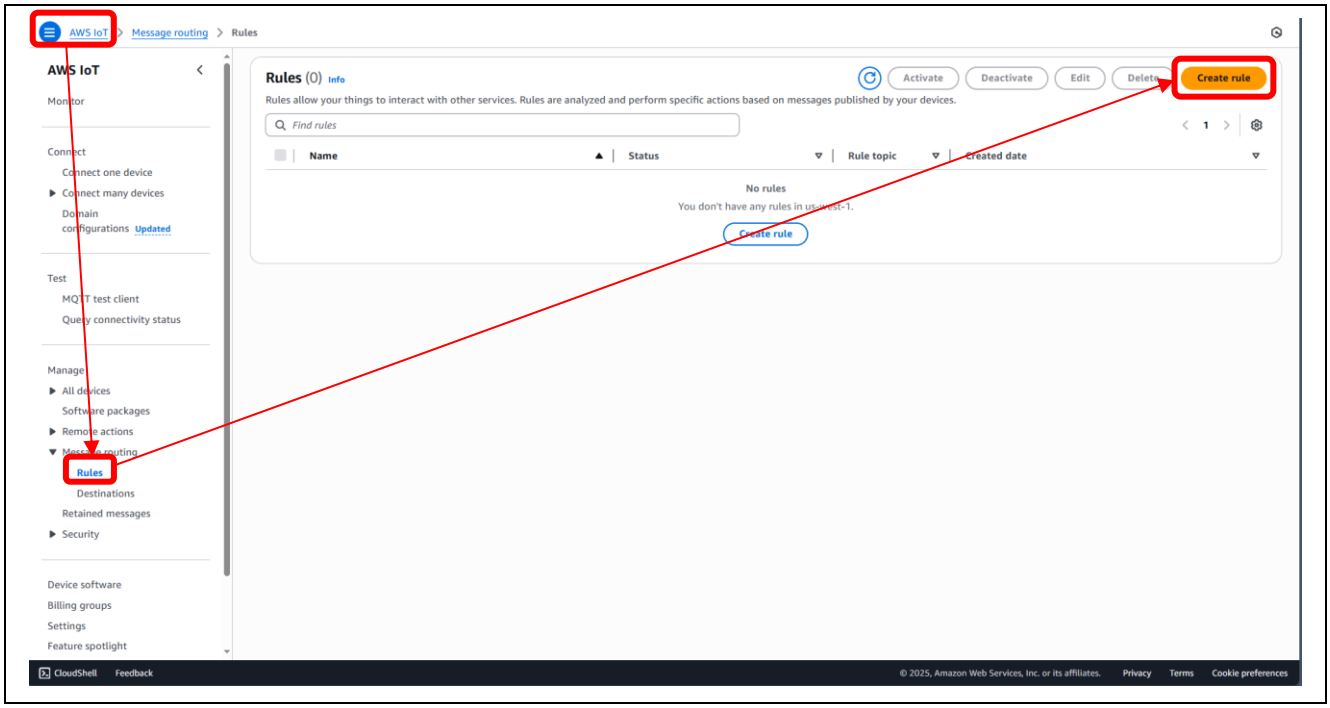
Manage AWS Resources - AWS Management Console - AWS (amazon.com)

Confirm the region displayed in the upper-right corner of the management console screen and select the same region as that set at the time of logging in to QE for OTA.



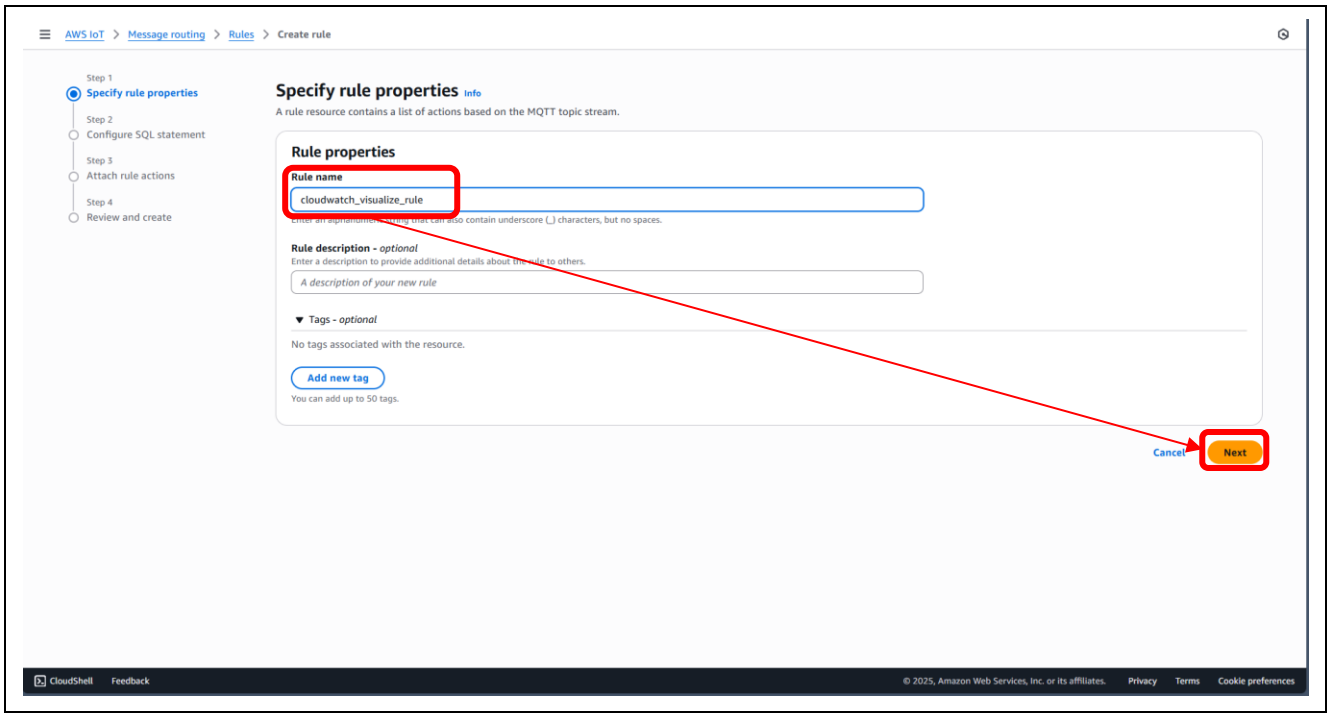
(2) Creating rules in AWS IoT

Click on [AWS IoT] → [Rules] → [Create rule].



(3) Specifying the rule properties

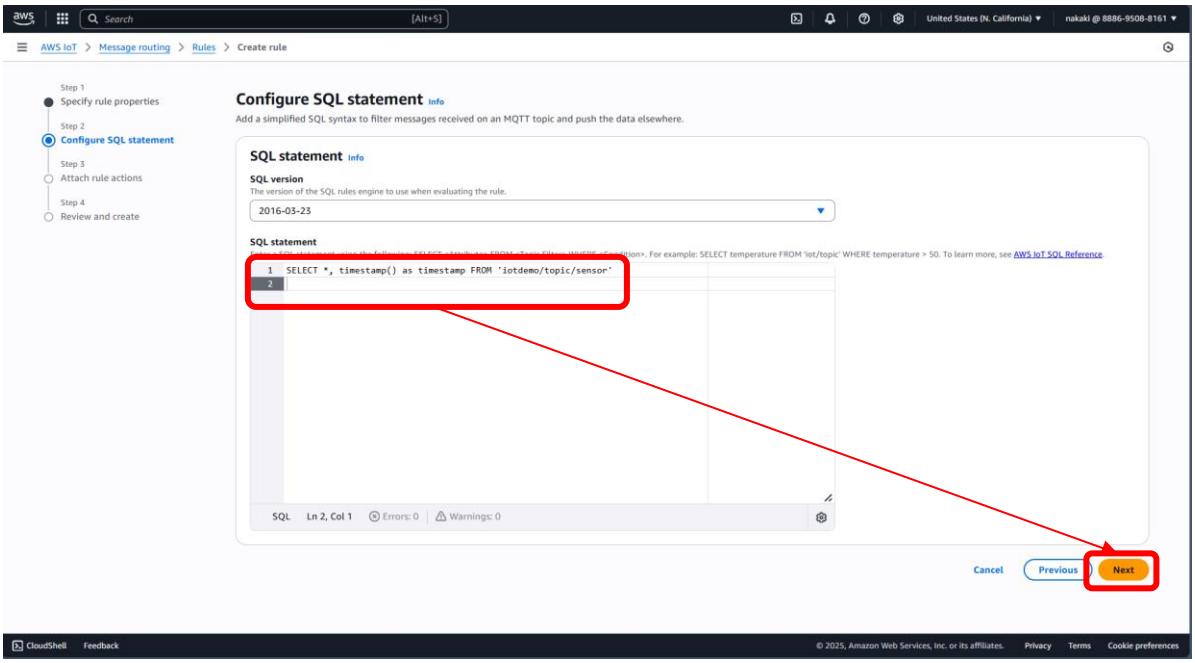
Enter a rule name in [Rule name] and click on [Next].



(4) Setting the SQL statement

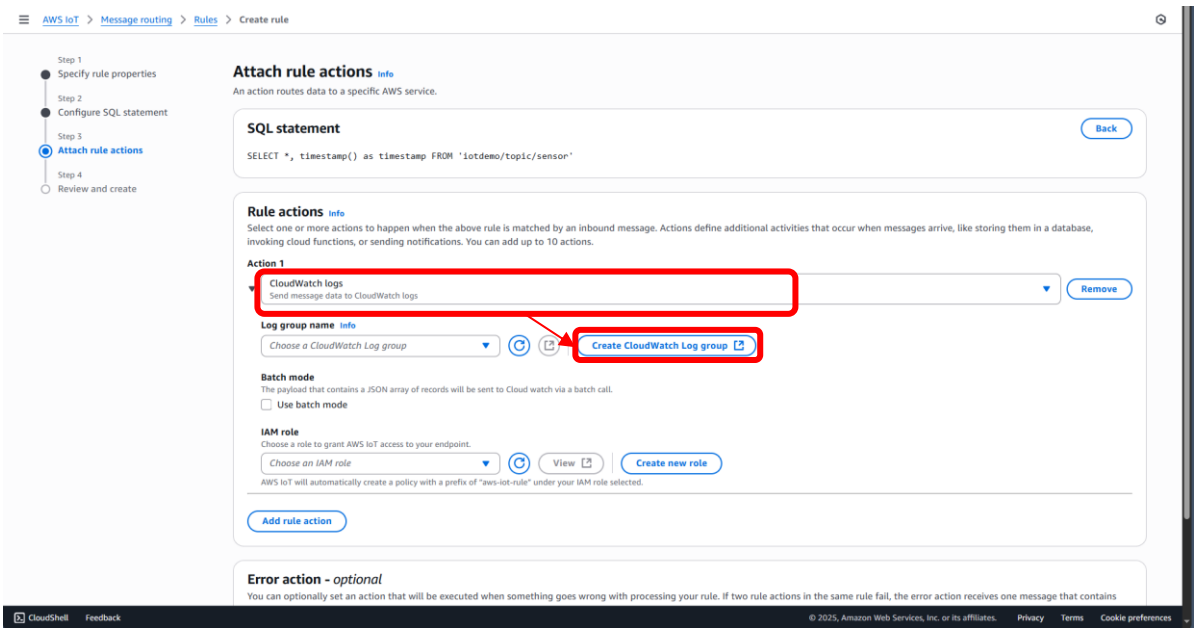
Enter the SQL statement by entering code like the following in the text editor field for [SQL statement]. Be sure to add a new line character at the end.

```
SELECT *, timestamp() as timestamp FROM 'iotdemo/topic/sensor'
```
(A new line has to be entered at the end of the line above.)



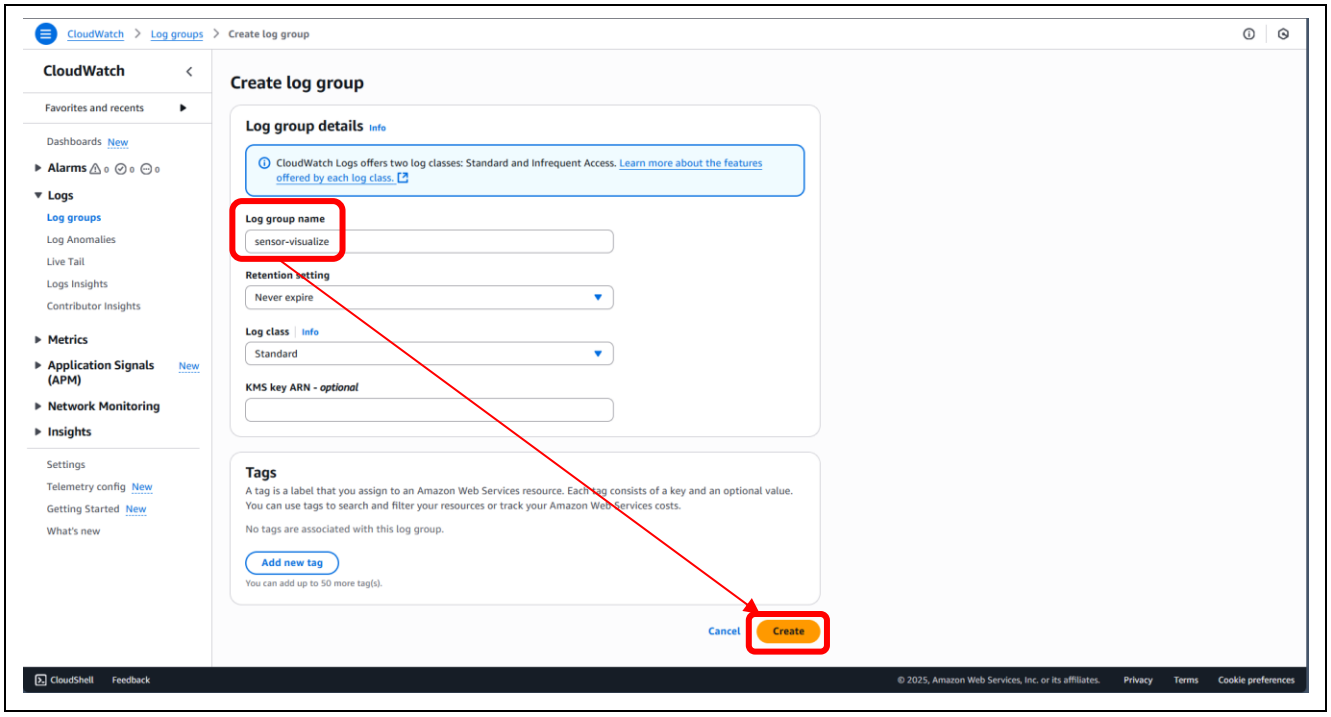
(5) Selecting rule actions in the [Attach rule actions] step

Select “CloudWatch logs” for [Action 1] and click on [Create CloudWatch Log group].



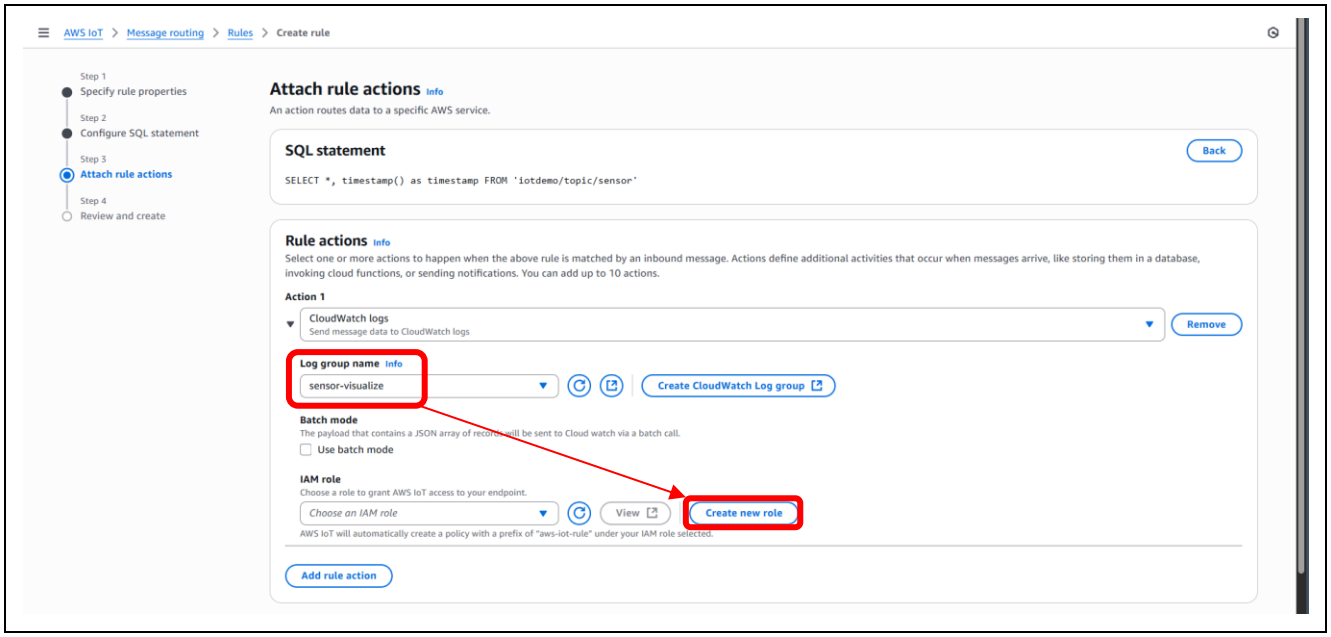
(6) Creating a log group

Enter a log group name and click on [Create].

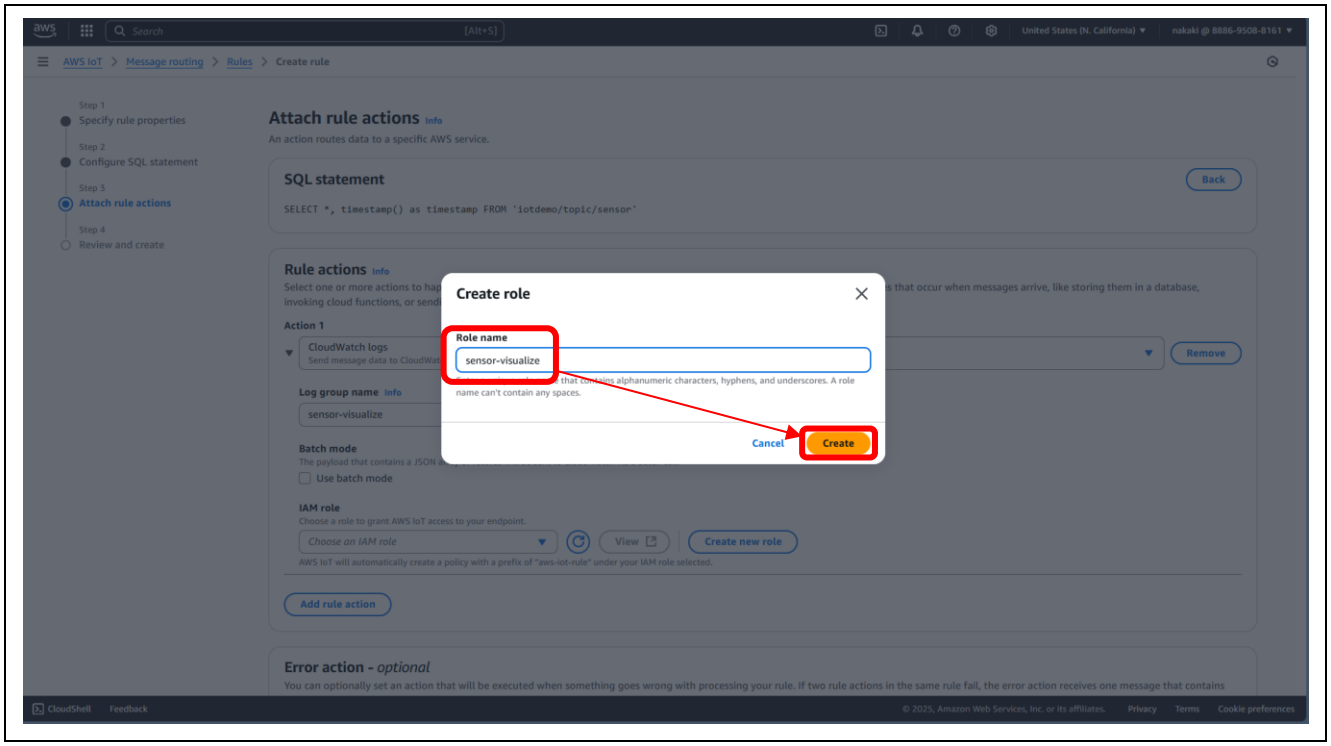


(7) Creating a new role

Select the created log group in [Log group name] and click on [Create new role].

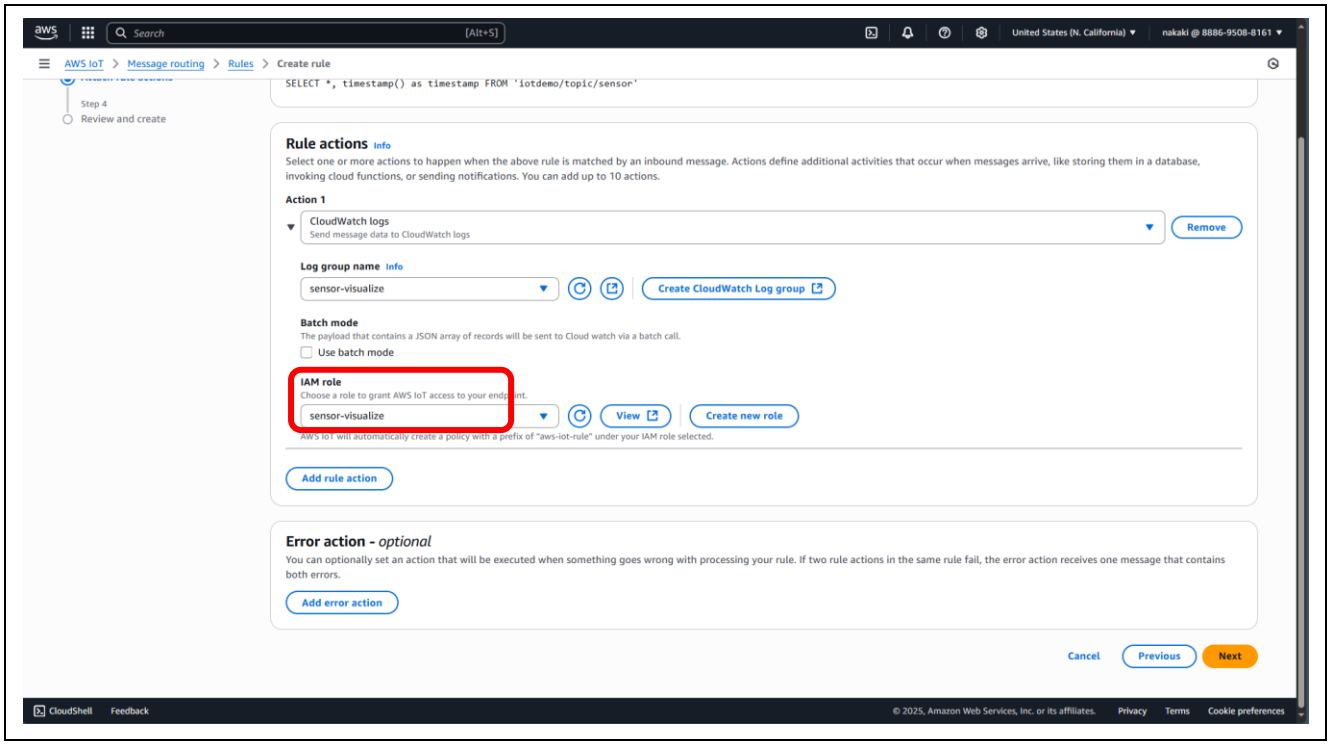


Enter the roll name and click [Create].



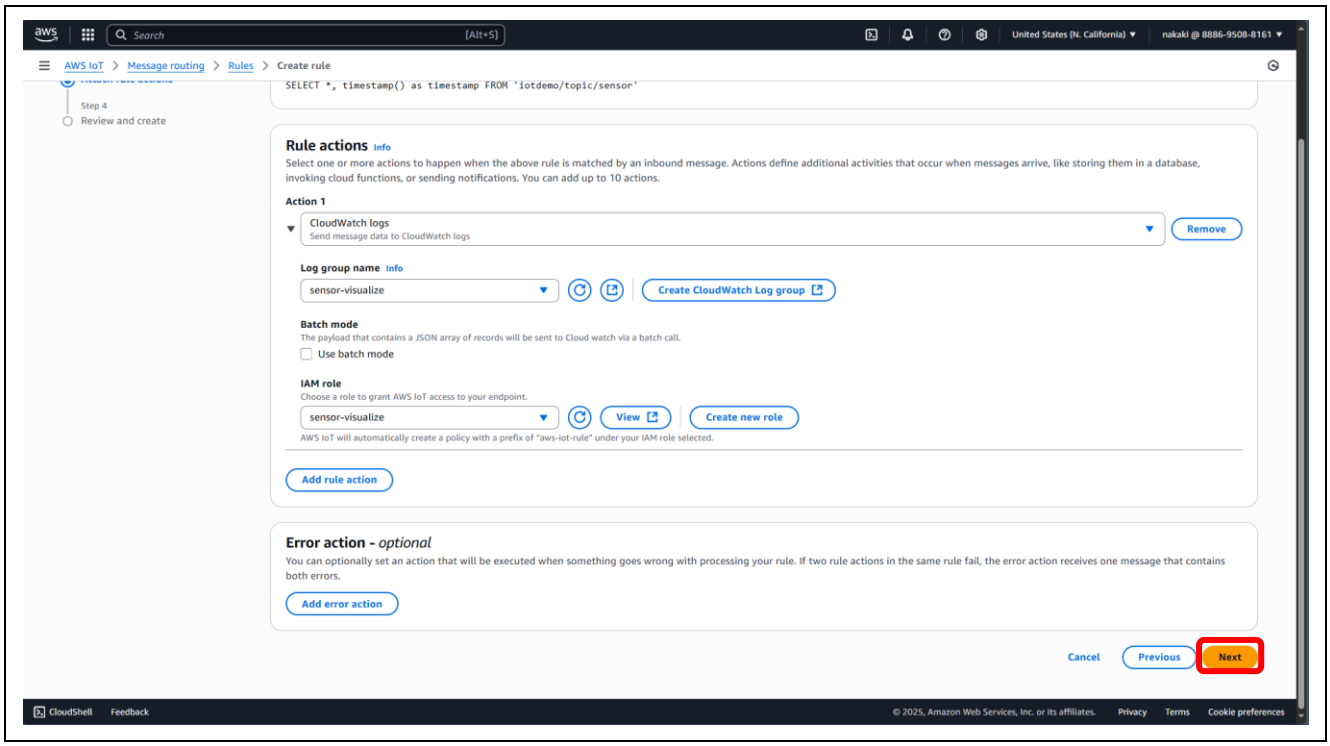
(8) Selecting the created IAM role

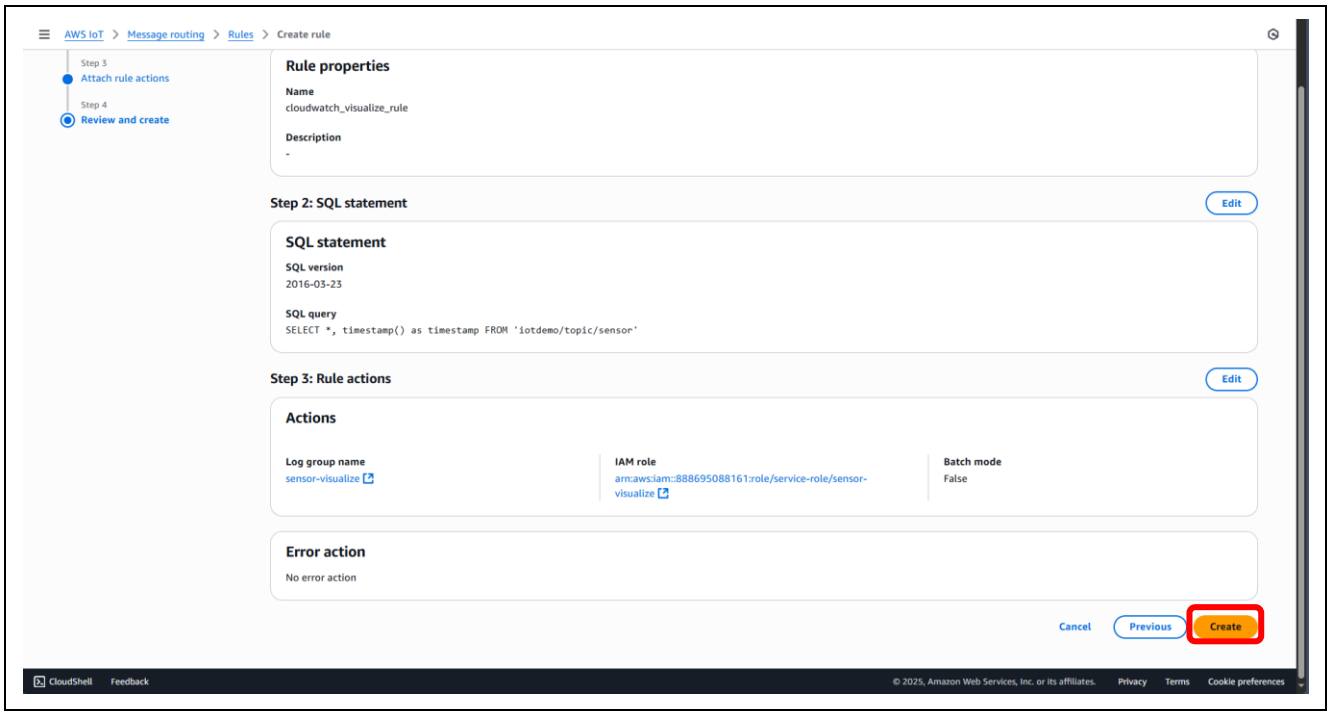
Select the created role in [IAM role].

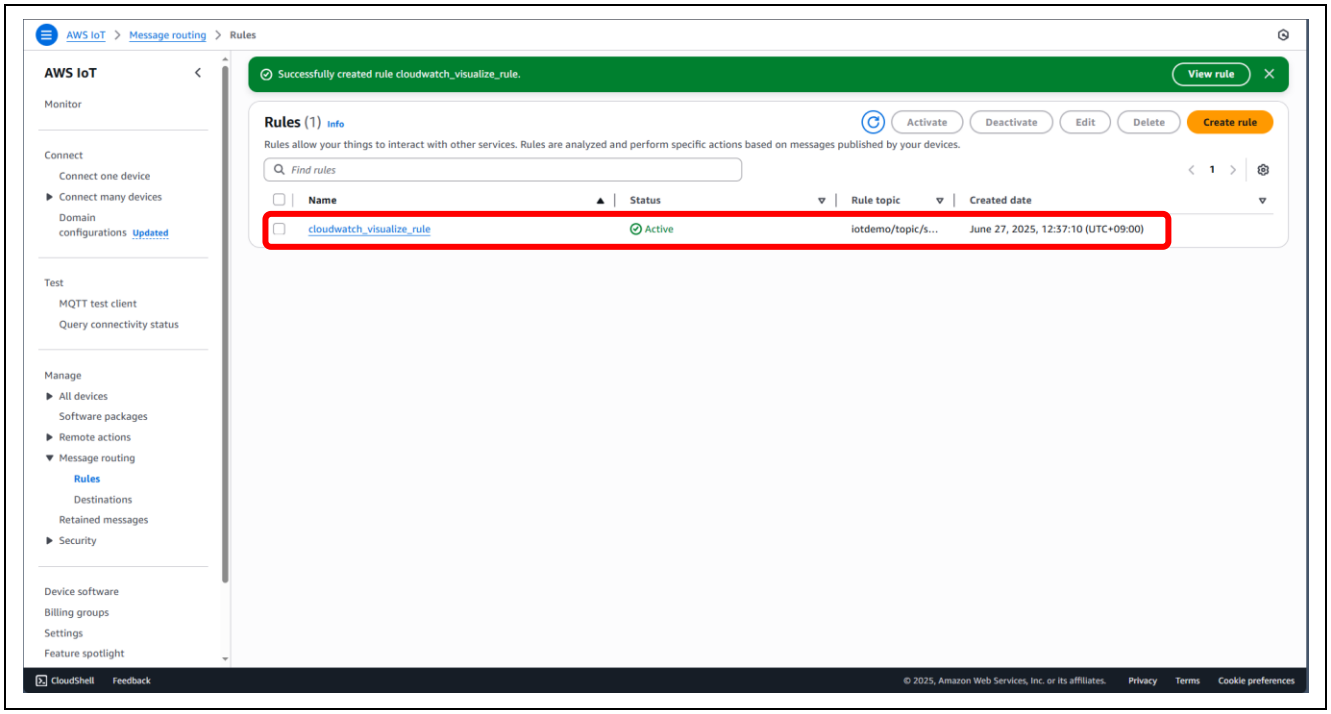


(9) Confirming successful creation of the rule

Click on [Next] and then click on [Create] on the subsequent page. Finally, confirm that the created rule is displayed in the list of rules.



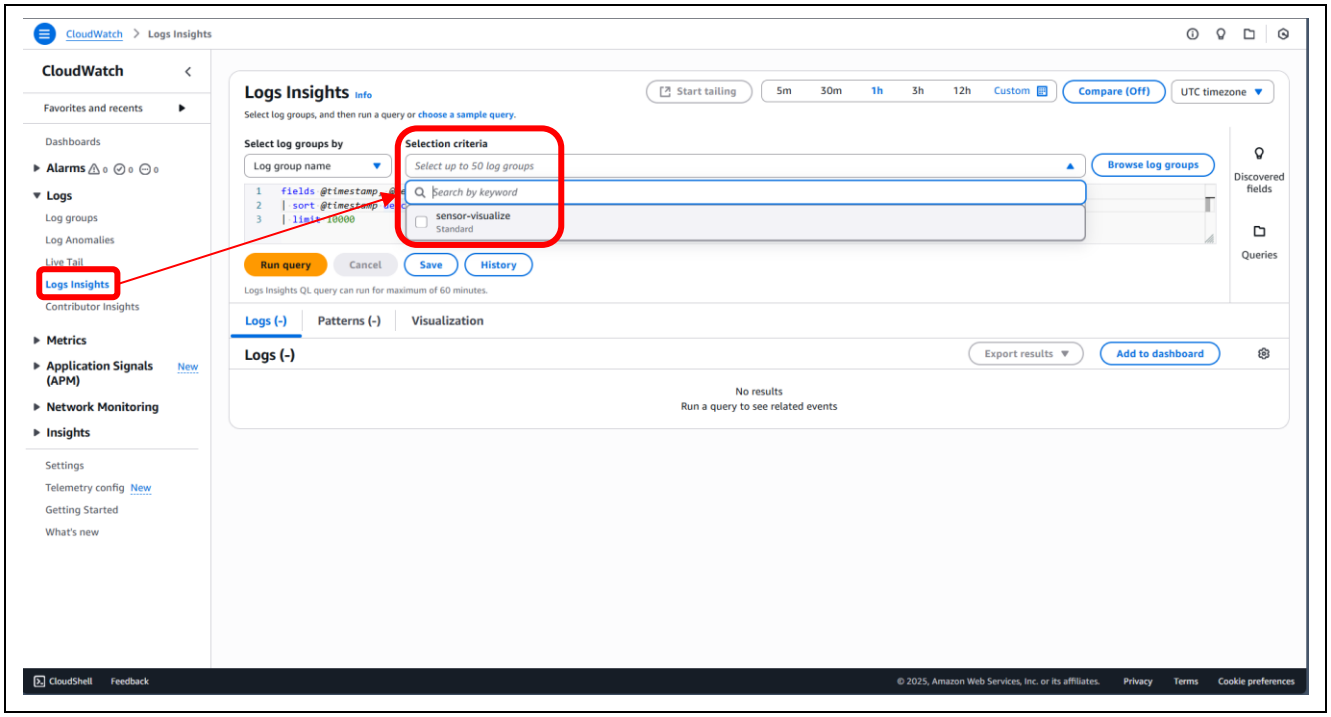




(10) Checking the graphical display in Amazon CloudWatch

Display the screen of Amazon CloudWatch and click on [Logs Insights] on the menu at left.

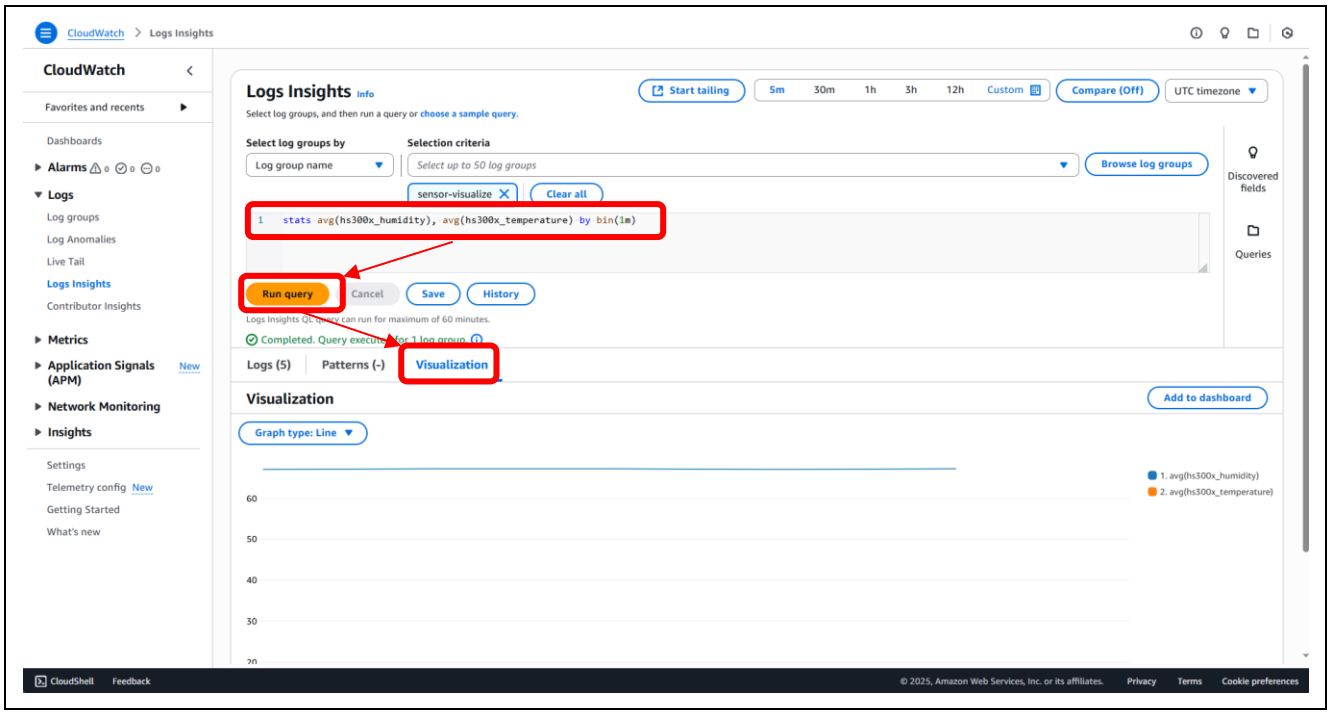
Select the group that was created in 6.3.2.1(5) as the log group.



Enter the following query and click on [Run query].

```
stats avg(hs300x_humidity), avg(hs300x_temperature) by bin(1m)
```

A graph is displayed on the [Visualization] tabbed page.



7. Procedure for Running the Demonstration

The procedure for running the demonstration is described below.

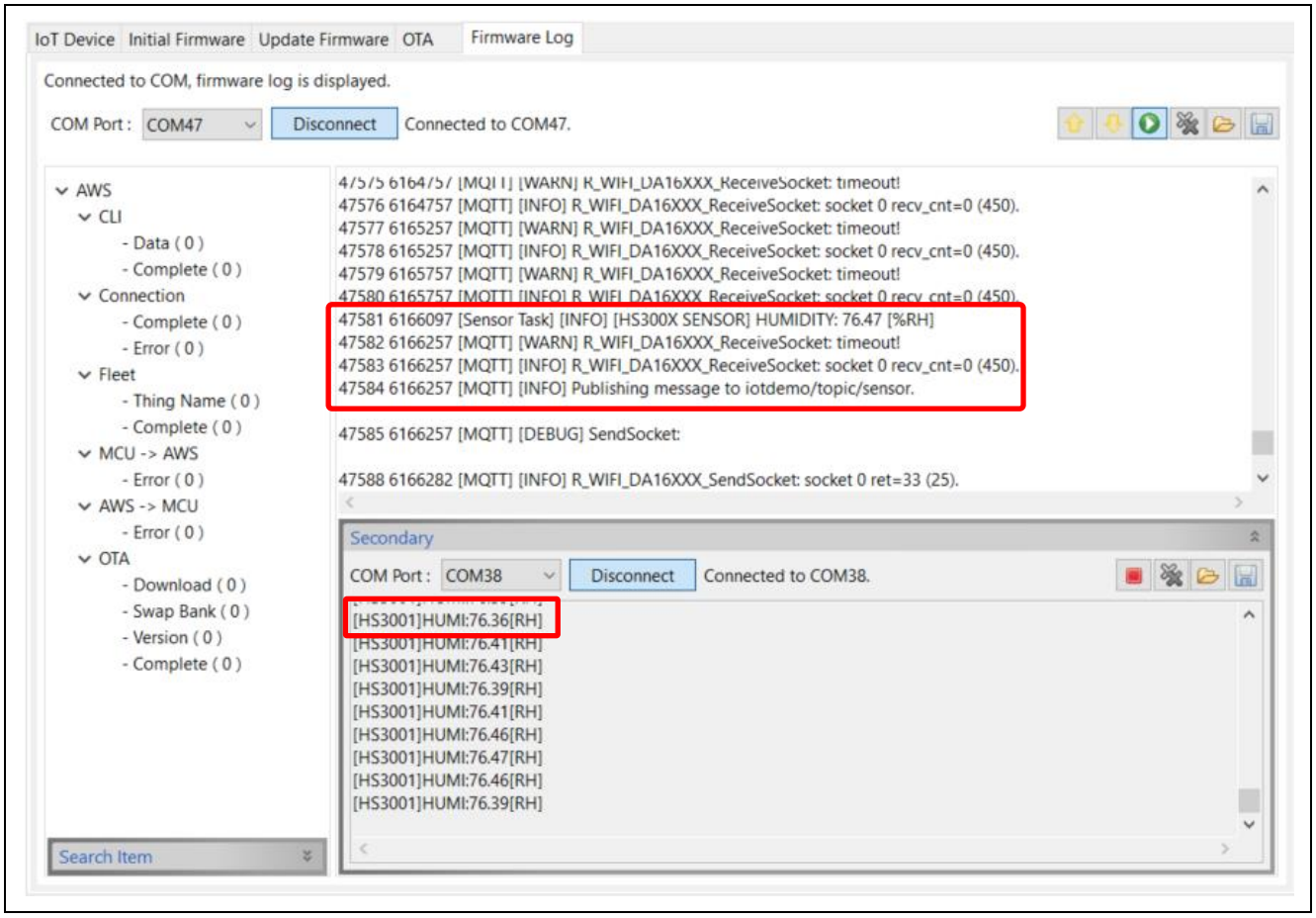
7.1 Checking the Initial State of Operation

With the setup for the demonstration described in section 6 completed, check the logs from each microcontroller in the “Firmware Log” tab of QE for OTA.

Confirm that humidity data from the HS3001 sensor is being output in the CK-RX65N log. Below that, you can also see the log of sensor data being sent to AWS via MQTT communications.

Next, confirm that only the humidity data from the HS3001 sensor is being output in the FPB-RX140 log. Also, In the initial state, LED1 of the FPB-RX140 is blinking.

If they are not displayed correctly, press the reset switch (S2 RESET) on the FPB-RX140 to apply a hardware reset. Similarly, press the reset switch (S1 RESET) on the CK-RX65N to apply a hardware reset.



Finally, Figure 7-1 shows the display for Amazon CloudWatch. Click on [Run query] and confirm that the humidity data acquired from the HS3001 sensor are displayed as a graph.

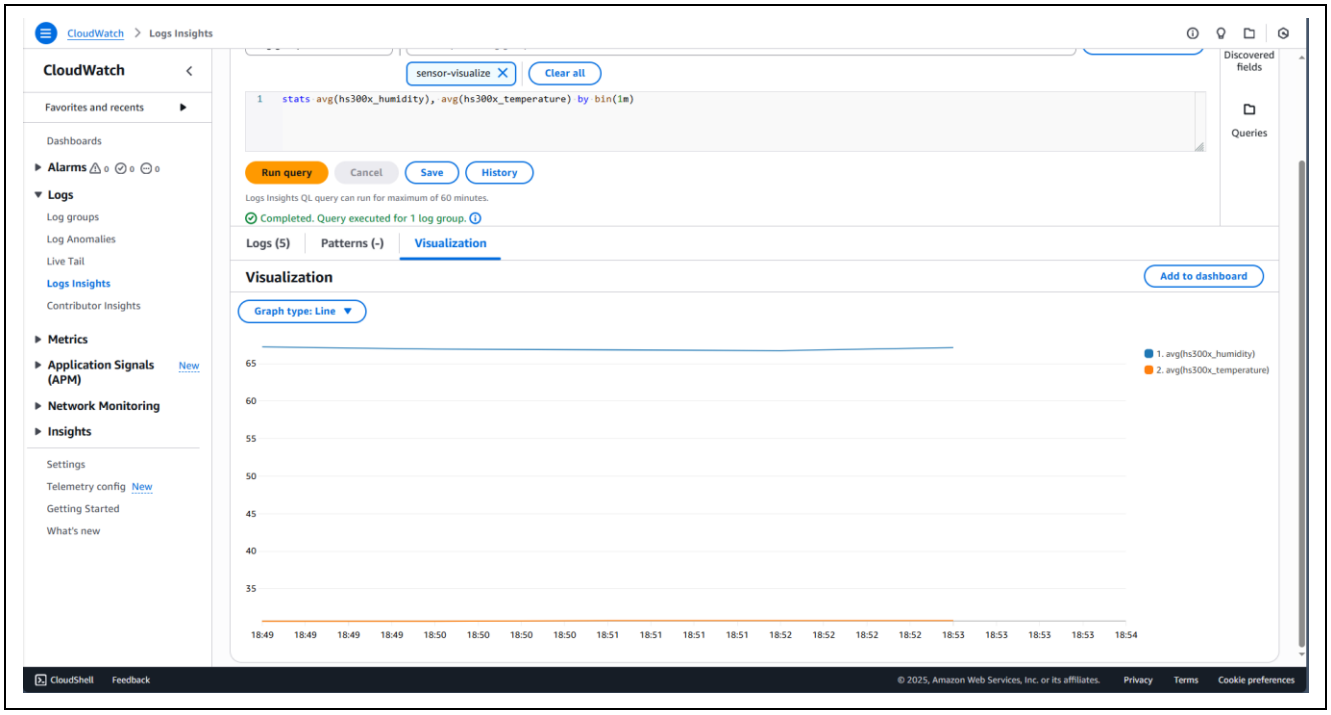


Figure 7-1 Graphical Display of Amazon CloudWatch before the Secondary OTA Update

This is the initial state before the secondary OTA update is run.

7.2 Executing the OTA Update of the FPB-RX140

7.2.1 Creating the Update Firmware

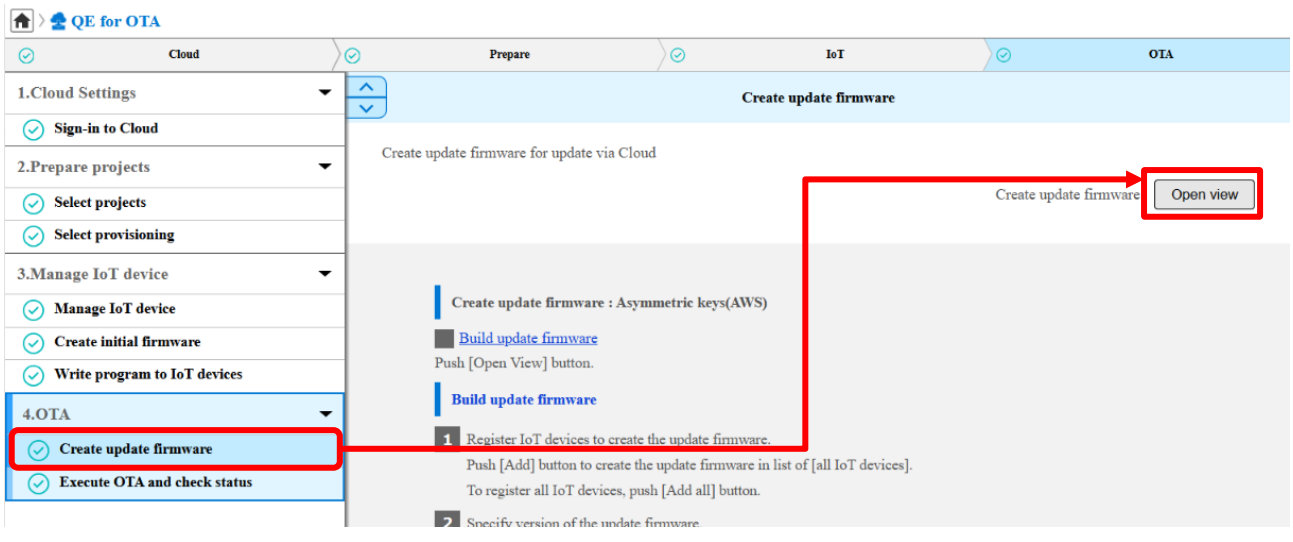
(1) Changing the source code of the demo_app_rx140_fpb_w_buffer project

Set the MEASURE_TEMPERATURE macro of demo_app_rx140_fpb_w_buffer/src/fwupcomm_demo_main.h to (1).

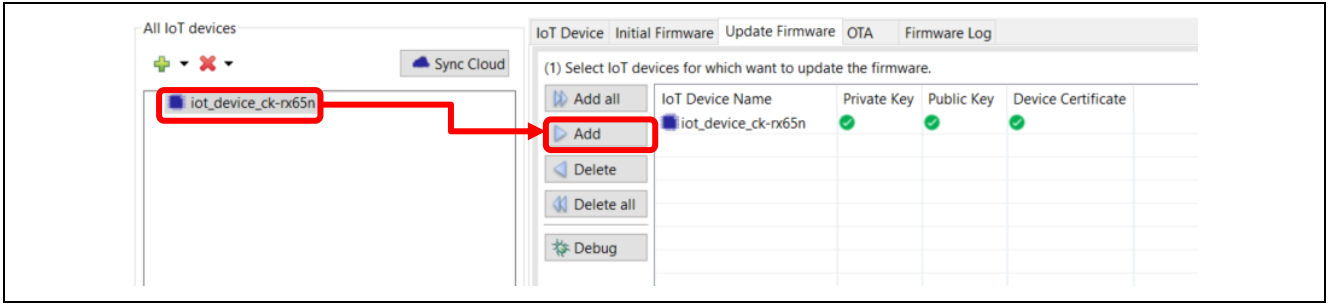
```
32  #define MEASURE_HUMIDITY        (1)
33  #define MEASURE_TEMPERATURE     (1)
34
35  #define DEMO_VER_MAJOR          (1)
36  #define DEMO_VER_MINOR          (0)
37  #define DEMO_VER_BUILD          (0)
```

(2) Creating the update firmware

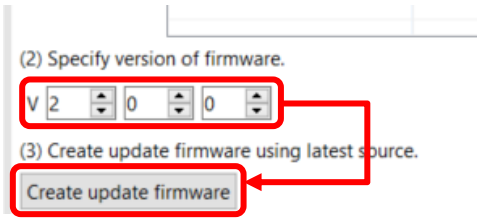
Click “Open view” in Create initial firmware.



In the “Update Firmware” tab, select the IoT device created 6.2.3(5) and click “▶Add”.



Enter “1.0.0” for the version and click “Create update firmware.”

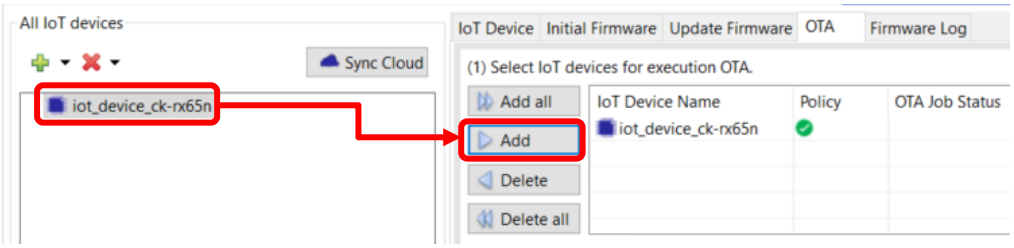


The demo_app_rx140_fpb_w_buffer project build is executed, and if “Create firmware is complete.” is displayed in the “Update Firmware” tab, the process is successful.

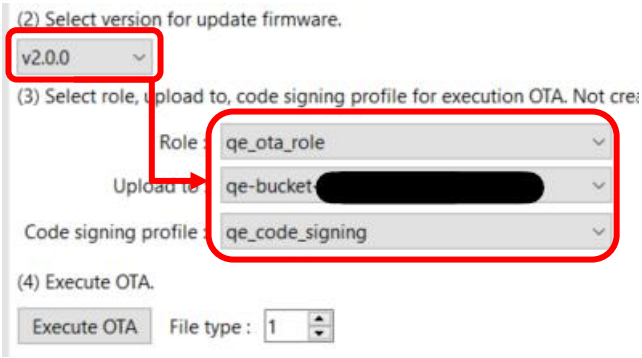
7.2.2 Creating an OTA Job

(1) Creating an OTA job

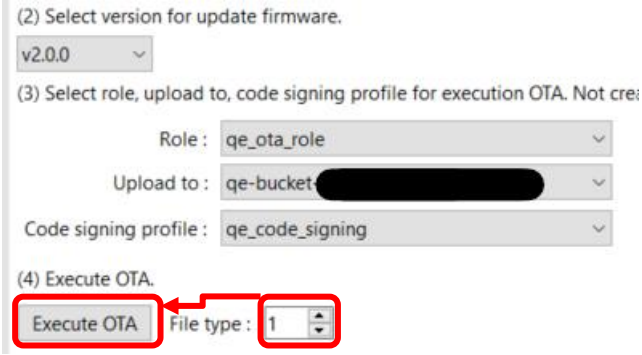
In the “OTA” tab, select the IoT device created 6.2.3(5) and click “▶Add”.



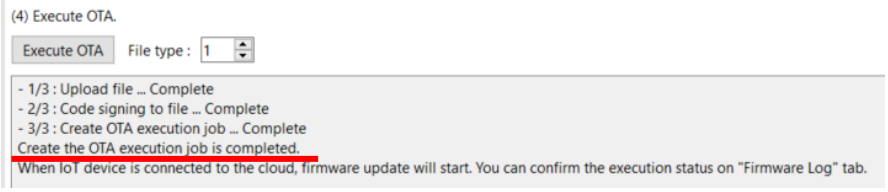
Select “v2.0.0” as the version for update firmware, and confirm that the role, upload to, and code signing profile have been specified.



Enter “1” for the file type and click “Execute OTA.”

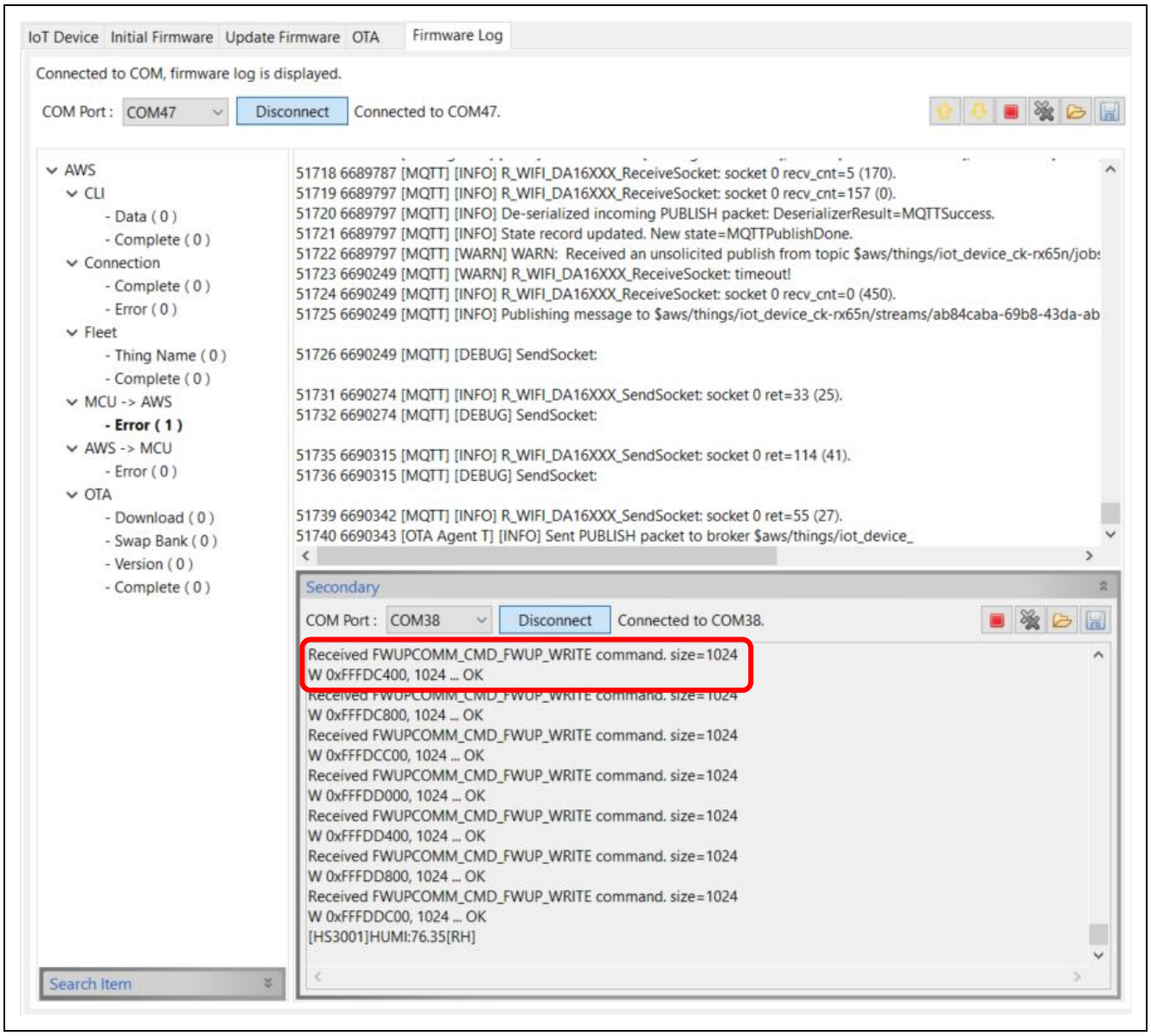


If “Create the OTA execution job is completed.” is displayed, the operation was successful. An OTA job for the secondary OTA update is created by following the above steps, and the OTA job is delivered to the specified IoT device.



7.2.3 Checking Operation during Execution of the Secondary OTA Update

The OTA update starts within a few seconds after creation of the job. Both the CK-RX65N and FPB-RX140 will output logs of the progress of the secondary OTA update.



7.3 Checking Operation after the OTA Update

Figure 7-2 shows the log screen of the CK-RX65N and FPB-RX140 after the update.

You can see that the temperature data are newly displayed in addition to the humidity data acquired from the HS3001 sensor.

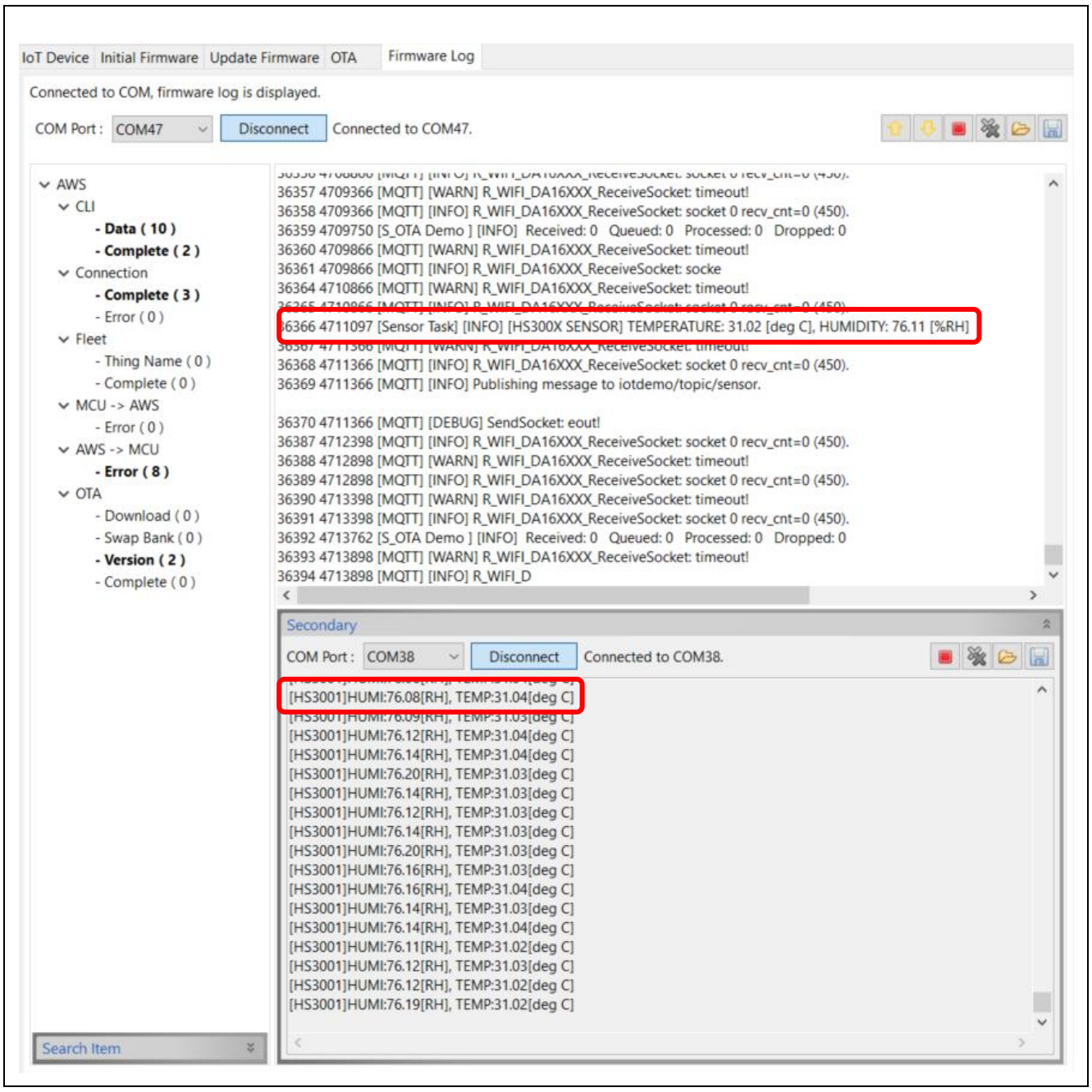


Figure 7-2 Log Screen of the CK-RX65N and FPB-RX140 after the Firmware Update

Furthermore, LED2 will now also be blinking in addition to LED1 that was blinking in the initial state, in the FPB-RX140.

Finally, Figure 7-3 shows the display for Amazon CloudWatch. Confirm that the measured temperature and humidity data acquired from the HS3001 sensor are displayed as a graph.

Figure 7-3 Graphical Display of Amazon CloudWatch after the Secondary OTA Update

Operations for the demonstration are completed at this point.

Note : In 7.2.2(1), if the update method for the project subject to secondary OTA updates is “partial update method,” set the File type value to “1.” If it is “full update method,” set the File type value to “2.”

8. How to do the demo without using the HS3001 sensor

This section describes how to perform a secondary OTA update demo without connecting the HS3001 sensor to the FPB-RX140. In this case, the following sensor-related functions in the demo won't work.

- Acquire temperature and humidity data from the HS3001 sensor connected to the FPB-RX140 and output it to a log.
- Send temperature and humidity data from CK-RX65N to AWS and display it on AWS.

8.1 Changes to the demo procedure

Please change the following steps in the demo procedure for sections 6 and 7. The other steps are the same as when using the sensor.

(1) Skip the step “6.1.2(1) Connecting the HS3001 board.”

(2) After completing “6.2.4(2) Selecting projects” change the `MEASURE_HUMIDITY` macro in demo_app_rx140_fpb_w_buffer/src/fwupcomm_demo_main.h to `(0)`.

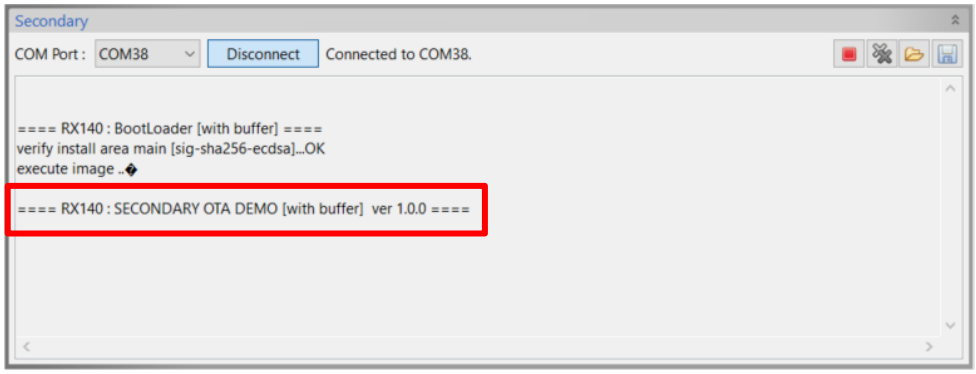
(3) Skip the step “6.3 Preparations for Displaying the Sensor Data Using the AWS Cloud.”

(4) Skip the step “7.2.1(1) Changing the source code of the demo_app_rx140_fpb_w_buffer project.”

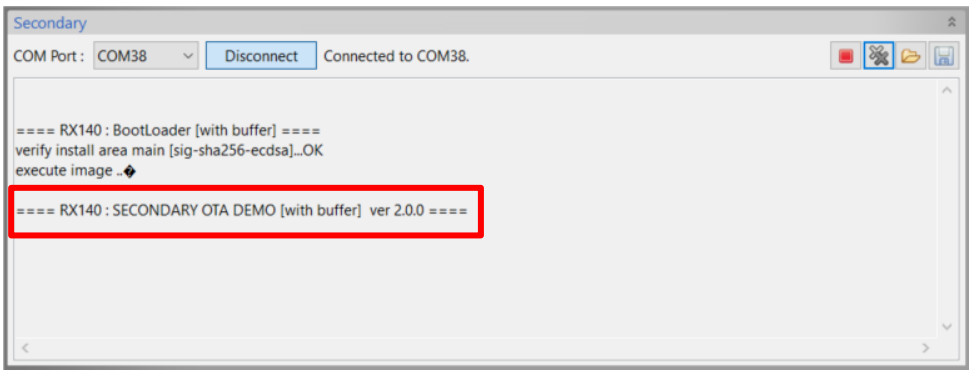
8.2 How to check demo operation when not using sensors

If the sensor is not used, it is not possible to confirm whether the firmware update has been performed by the type of sensor data acquired. Please check the version information displayed in the log when the FPB-RX140 starts up.

<Before updating>



<After updating>



9. Precautions

9.1 License Information on the Open-Source Software in Use

The following open-source software is used.

- TinyCrypt Cryptographic Library
  - URL: https://github.com/intel/tinycrypt
  - License: https://github.com/intel/tinycrypt/blob/master/LICENSE

- FreeRTOS
  - URL: https://www.freertos.org/
  - License: FreeRTOS open source licensing, FreeRTOS license description, FreeRTOS license terms and OpenRTOS commercial licensing options.

9.2 Region and User Privileges of AWS for the Demonstration

Regarding the setup of AWS for running the demonstration, notes on the region of use and user privileges are given below.

<Region of use>

This demonstration is provided in the ap-northeast-1 (Asia Pacific (Tokyo)) region of AWS.

If you want to run this demonstration in another region, confirm that the services used in the demonstration are available in that region beforehand.

<User privileges>

This demonstration is to be run by a user with Administrator Access permission in the AWS Identity and Access Management (IAM) system. Therefore, there is no particular description regarding the granting of necessary permissions in IAM when using various services.

9.3 Fees for Using AWS

A charge may apply to the cloud resources created and used in the demonstration depending on how AWS is used. To avoid inadvertently incurring charges, deleting the resources created in the cloud after running the demonstration is recommended.

Revision History

Rev.	Date	Description	
		Page	Summary
1.01	Jan. 24, 2022	—	First edition issued.
1.10	Mar. 31, 2022	—	Supported AWS IoT Over-the-air Update Library v3.0.0.
		5 - 8	Added .settings folder to the folder structure of each project.
		5 - 8	Revised package and folder structure for RX65N project.
		9	Updated IDE environment to e2studio 2022-01, Toolchain to CC-RX V3.04.00 and FreeRTOS for RX65N project to Version 2021.07.
		9	Updated code size.
		19 - 20	Updated initial firmware creation method due to RX65N project changes.
		47	Updated screenshot of output log due to RX65N project changes.
		51 - 55	Changed the method of executing an update.
		56	Updated screenshot of output log due to RX65N project changes.
2.00	Mar. 31, 2024	—	The used boards were changed to the CK-RX65N and TB-RX660.
		—	The FreeRTOS package for the projects for the RX65N was updated.
		—	The version of the firmware update module (FWUP) using Firmware Integration Technology (FIT) for the projects for the RX660 was updated.
		—	The entire application note was revised due to changes in the used boards and projects.
3.00	2025/06/30	—	The used boards were changed to the CK-RX65N v2 and FPB-RX140 v1.
		—	Changed communication control between microcontrollers to Firmware Updating Communications Module.
		—	Changed the demo procedure to use the secondary OTA function of QE for OTA V2.2.0.

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.

Notice

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