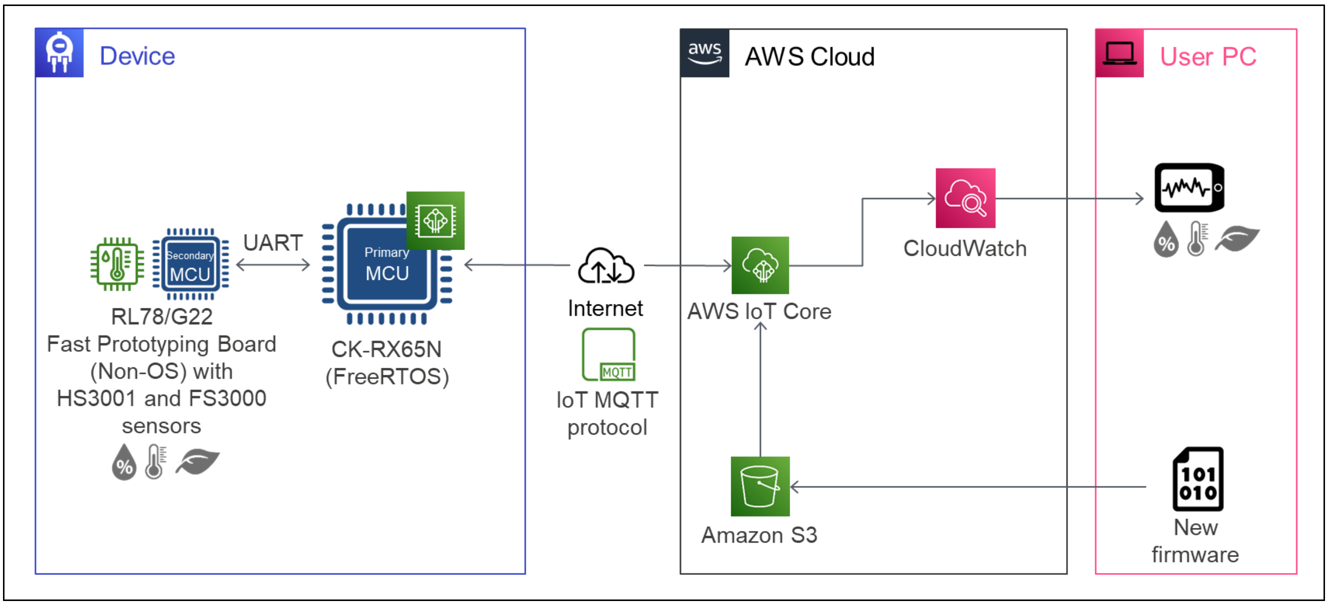
RL78/G22

OTA Firmware Update for a Secondary MCU

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 RL78 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) firmware update of the secondary MCU (hereafter, referred to as "secondary OTA update").



Devices Used in Confirming Operation

- RX65N
- RL78/G22
- Sensors
  HS3001 high-performance relative humidity and temperature sensor (HS3001 sensor)
  FS3000 air velocity sensor module (FS3000 sensor)

Boards Used in Confirming Operation

- Primary MCU: CK-RX65N (RTK5CK65N0S04000BE)
- Secondary MCU: RL78/G22 Fast Prototyping Board (RTK7RLG220C00000BJ)
- Sensors
  Relative Humidity Sensor Pmod™ Board (US082-HS3001EVZ)
  FS3000 Pmod™ Board (US082-FS3000EVZ)

Related Documents

RL78/G22 User’s Manual: Hardware (R01UH0978)

RL78 Family User's Manual: Software (R01US0015)

RL78/G22, RL78/G23, RL78/G24 Firmware Update Module (R01AN6374)

RL78/G22 Fast Prototyping Board User’s Manual (R20UT5121)

RX65N Group CK-RX65N v1 User’s Manual (R20UT5100)

RX65N Group Sample Code for OTA Update of Secondary Device with Amazon Web Services Using FreeRTOS (R01AN6220)

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

(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

Amazon Web Services, the “Powered by AWS” logo, and any other AWS trademarks used in such materials are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries.

FreeRTOS™ and FreeRTOS.org™ are trademarks of Amazon Web Services, Inc.

Pmod is a trademark of Digilent Inc.

All trademarks and registered trademarks are the property of their respective owners.

1. Overview

This demonstration involves using a secondary OTA update to add a working sensor and confirming addition of the working sensor 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 (RL78/G22)

Item	Description
MCU	RL78/G22 (R7F102GGE2DFB)
Board	RL78/G22 Fast Prototyping Board (RTK7RLG220C00000BJ)
Operating frequency	High-speed on-chip oscillator clock: 32 MHz
Operating voltage	3.3 V
Integrated development environment (IDE)	e^2 studio V2024-01 (24.1.0) made by Renesas Electronics Corporation
C compiler	CC-RL v1.13.00 made by Renesas Electronics Corporation
Firmware programming tool	Renesas Flash Programmer V3.14.00
Smart Configurator (SC)	Renesas Smart Configurator for RL78 23.4.0.v20230320-0633
Board Support Package (BSP)	v1.62 (r_bsp)

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

Item	Description
Relative humidity and temperature sensor board	US082-HS3001EVZ board
Air velocity sensor board	US082-FS3000EVZ board

Table 2-3 Conditions for Confirming Demo Operation (RX65N)

Item	Description
MCU	RX65N (R5F565NEHDFB)
Board	CK-RX65N (RTK5CK65N0S04000BE)
IDE	e^2 studio V2024-01 (24.1.0)
C compiler	CC-RX v3.06.00
RTOS	FreeRTOS v202210.01-LTS-1.1.3

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

Item	Description
QE for OTA	V2.00
Python	3.10.4

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

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

3. Description of Hardware

3.1 System Configuration

The system consists of an RX65N microcontroller (primary MCU) that provides functionality for controlling communications with AWS and an RL78 microcontroller (secondary MCU) connected to the HS3001 and FS3000 sensors. The two microcontrollers communicate with each other via UARTs.

Via UART communications, this demonstration can perform secondary OTA updating of the RL78 microcontroller connected to sensors, the uploading of sensor data acquired from sensor boards to the cloud, and the display of sensor data.

The system configuration is shown in Figure 3-1.

An RL78/G22 Fast Prototyping Board (hereafter referred to as “RL78/G22 FPB”) equipped with an RL78/G22 microcontroller is used as the secondary MCU.

The CK-RX65N equipped with an RX65N microcontroller is used as the primary MCU.

Also, a UART connection is made between the CK-RX65N and RL78/G22 FPB.

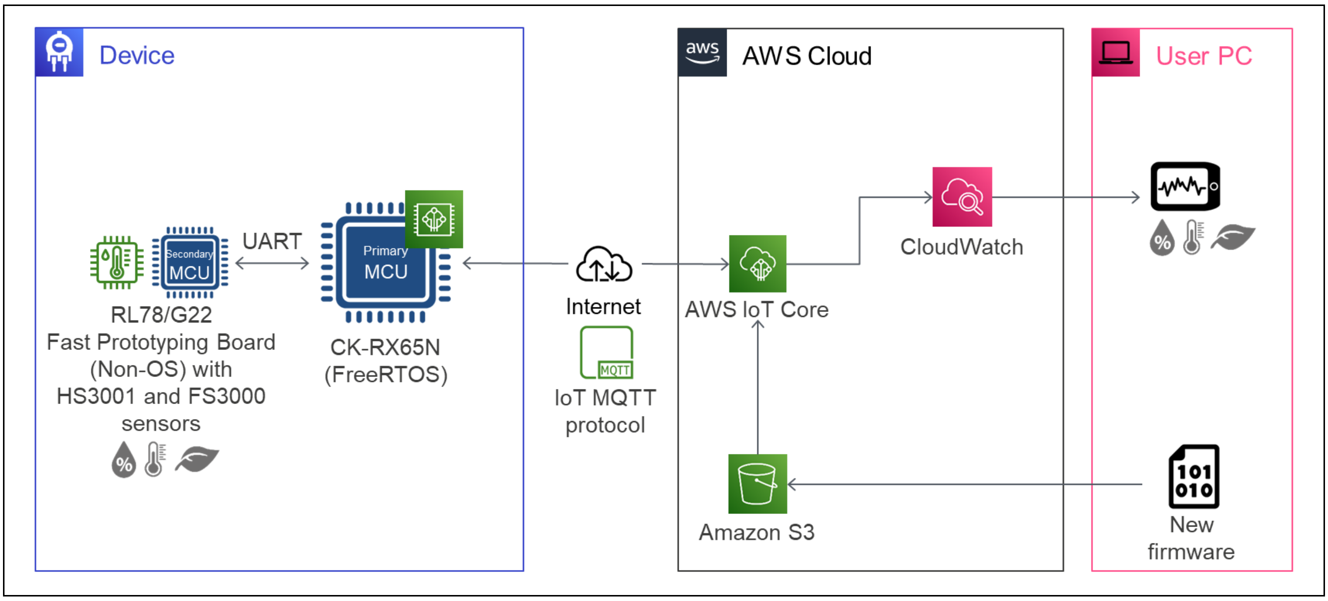


Figure 3-1 System Configuration of This Demo

3.2 List of Pins Used

Table 3-1 lists the pins of the RL78/G22 microcontroller that are used and their functions.

Table 3-1 Pins Used and Their Functions

Pin Name	Input/Output	Description
P12/TxD0	Output	Log output to the PC
P00/TxD1	Output	UART communications with the RX65N (transmission)
P01/RxD1	Input	UART communications with the RX65N (reception)
P70/SCL21	Output	I^2C communications with sensors (clock)
P71/SDA21	Input/Output	I^2C communications with sensors (data)

Caution **In this application note, only the pins that are used are handled correctly. When actually creating a circuit, handle all pins appropriately and be sure to design a circuit that satisfies the hardware’s electrical characteristics.**

4. Description of Software

"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 RL78 microcontroller, which is received from AWS, is then transmitted to the secondary MCU, where the firmware update is applied.

The RL78 microcontroller on the secondary MCU side uses "RL78/G22,RL78/G23,RL78/G24 Firmware Update Module Rev.2.01" to control OTA updating of the secondary MCU.

4.1 Firmware Update Methods

This application note provides sample programs for two among the methods provided by the firmware update module. The two methods are "partial update method (buffer side is internal flash)" (hereafter referred to as "partial update method") and "full update method (without buffer side)" (hereafter referred to as "full update method"). For details on these methods, refer to "1.3 Firmware Update Operation" in "RL78/G22,RL78/G23,RL78/G24 Firmware Update Module Rev.2.01".

4.1.1 Partial Update Method

The memory map of the sample program for executing partial update is shown in the following.

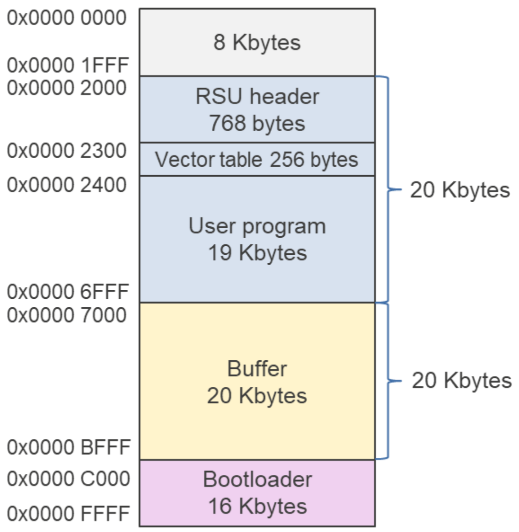


Figure 4-1 Memory Map of the Sample Program for Executing Partial Update

Operations for the partial update method are summarized in Figure 4-2. The states of the ROM in each phase of the OTA update are shown in Figure 4-3. Note that the red frames in Figure 4-3 indicate the programs under execution at the given times.

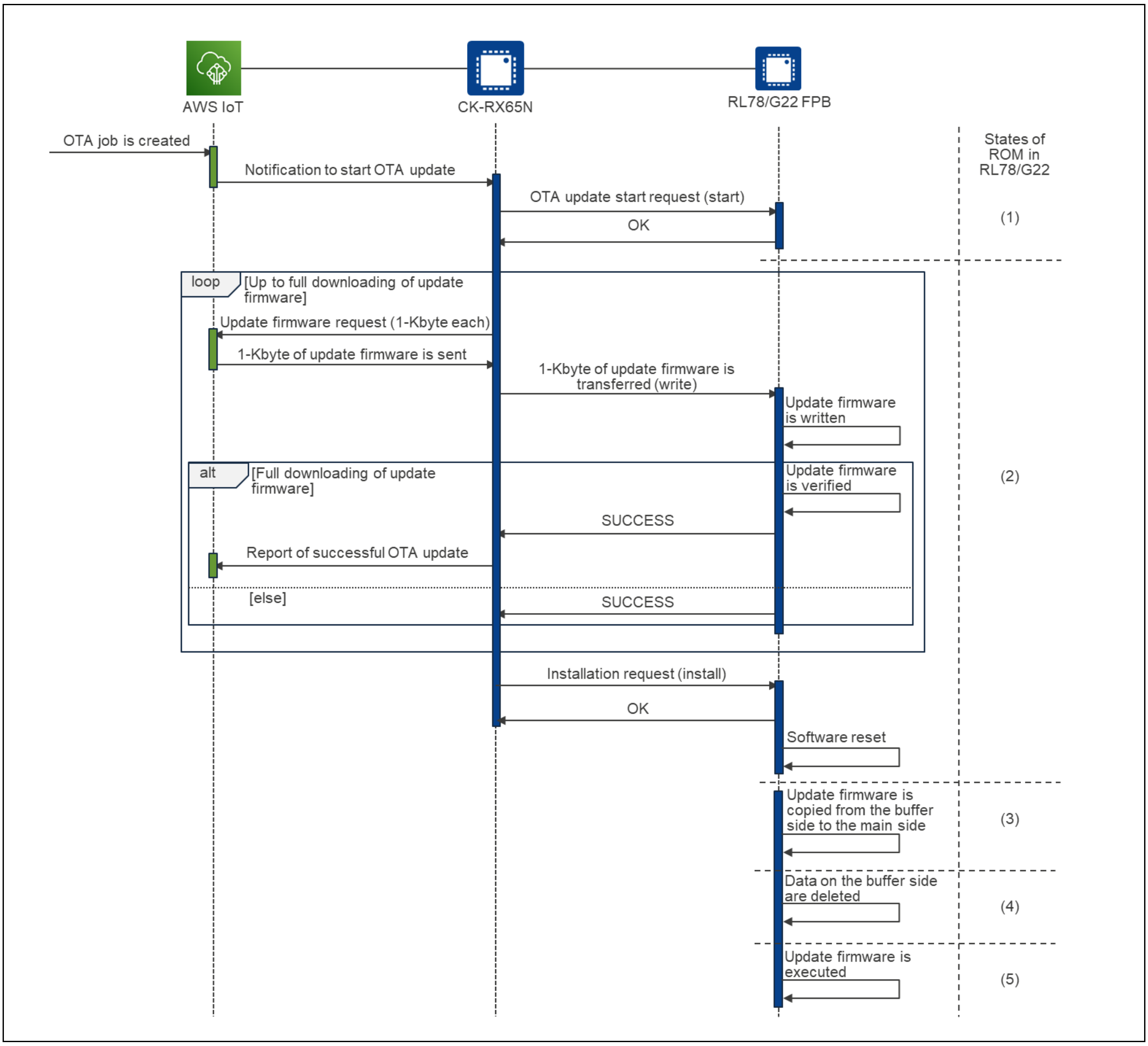


Figure 4-2 Overview of Operations for the Partial Update Method

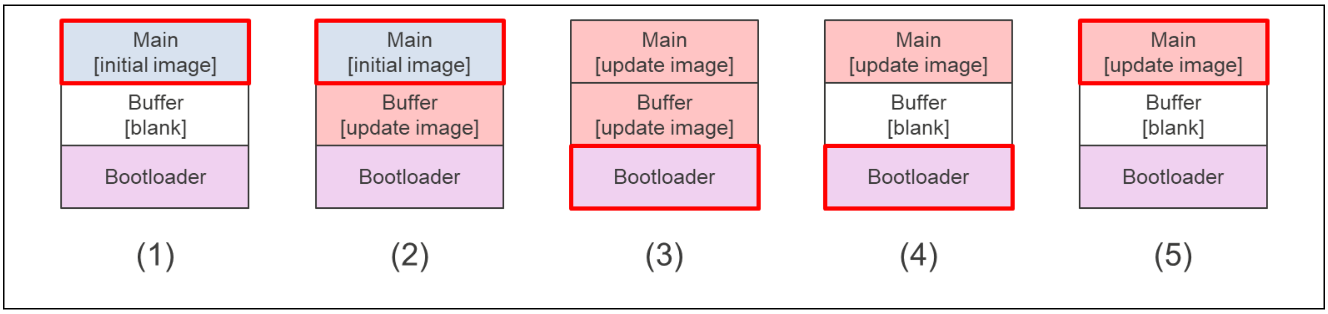


Figure 4-3 States of ROM in Each Phase of OTA Updating by the Partial Update Method

4.1.2 Full Update Method

The memory map of the sample program for executing full update is shown in the following.

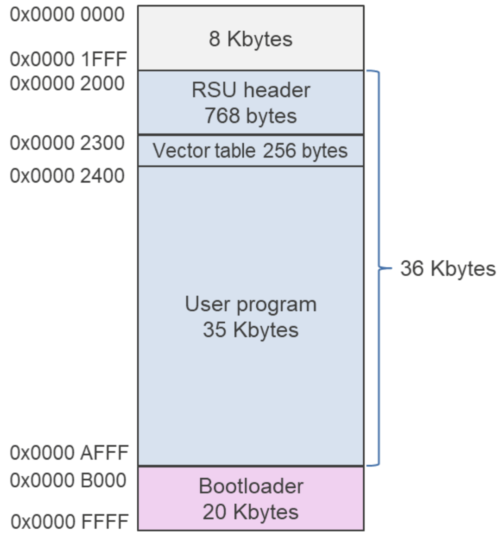


Figure 4-4 Memory Map of the Sample Program for Executing Full Update

Operations for the full update method are summarized in Figure 4-5. The states of the ROM in each phase of the OTA update are shown in Figure 4-6.

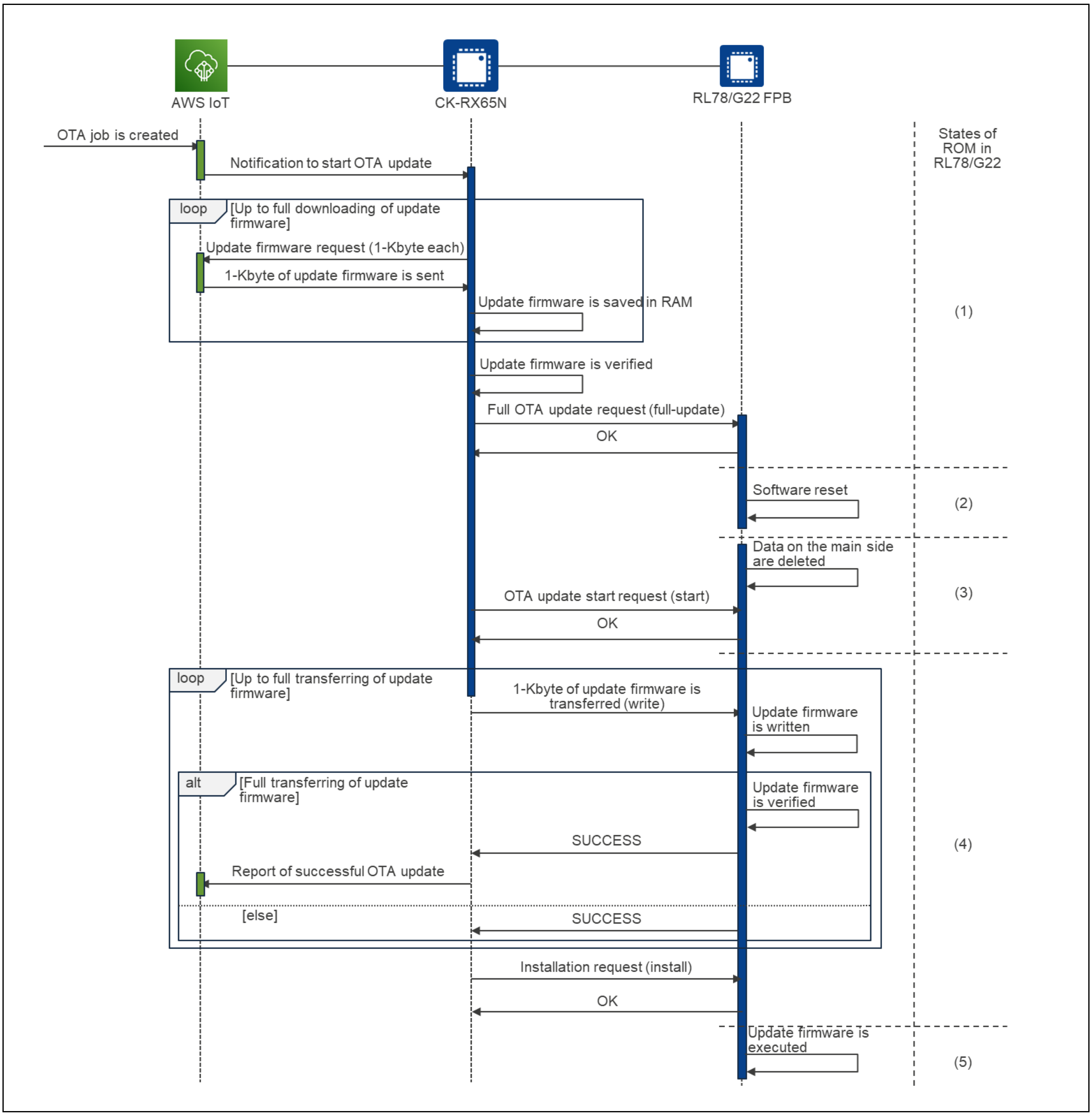


Figure 4-5 Overview of Operations for the Full Update Method

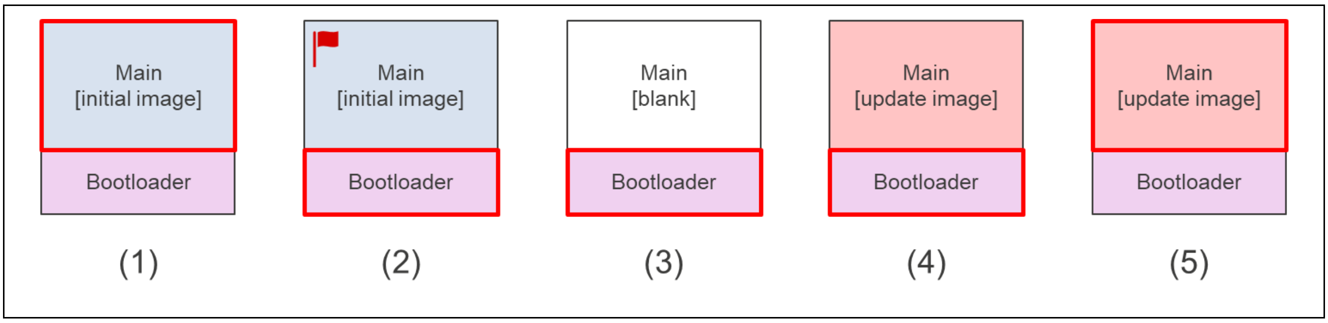


Figure 4-6 States of ROM in Each Phase of OTA Updating by the Full Update Method

4.2 Settings of Option Bytes

The settings of option bytes are listed below.

Table 4-1 Settings of Option Bytes

Address	Setting	Details
000C0H or 020C0H	01101110B	Stops operation of the watchdog timer (counting is stopped after the microcontroller is released from the reset state).
000C1H or 020C1H	11111110B	LVD detection voltage: Reset mode In the case of rising: Typ. 1.90 V (1.84 V to 1.95 V) In the case of falling: Typ. 1.86 V (1.80 V to 1.91 V)
000C2H or 020C2H	11101000B	High-speed main (HS) mode or high-speed on-chip oscillator clock (f_{IH}): 32 MHz
000C3H or 020C3H	00000100B	Disables on-chip debugging.

4.3 Folder and File Structure

Figure 4-7 shows the folder structure of the sample programs.

```
r01an6935xx0110-rl78g22
├─Demo
│   ├─afr-v202210.01-LTS-rx-1.1.3
│   ├─rl78g22_fpb_2ndota_demo
│   ├─rl78g22_fpb_bootloader
│   ├─rl78g22_fpb_2ndota_demo_full
│   ├─rl78g22_fpb_bootloader_full
│   ├─ck_rx65n_2ndota_demo
│   ├─ck_rx65n_demo_bootloader
│   └─RenesasImageGenerator
├─r01an6935ej0110-rl78g22.pdf
└─r01an6935jj0110-rl78g22.pdf
```

Figure 4-7 Folder Structure of Sample Programs

The ck_rx65n_2ndota_demo folder and ck_rx65n_demo_bootloader folder contain project files for the CK-RX65N.

The rl78g22_fpb_2ndota_demo folder and rl78g22_fpb_bootloader folder contain project files for the partial update method of the RL78/G22 FPB.

The rl78g22_fpb_2ndota_demo_full folder and rl78g22_fpb_bootloader_full folder contain project files for the full update method of the RL78/G22 FPB.

4.4 Code Size

Table 4-2 shows the code size for each project.

Table 4-2 Code Size

Project	ROM	RAM
rl78g22_fpb_bootloader	11 Kbytes	0.8 Kbytes
rl78g22_fpb_2ndota_demo	16 Kbytes	2 Kbytes
rl78g22_fpb_bootloader_full	14 Kbytes	2 Kbytes
rl78g22_fpb_2ndota_demo_full	18 Kbytes	2 Kbytes

5. Operations of the Demonstration

The operations of the demonstration are common to the partial update method and full update method.

(1) In the initial state of the demonstration, the RL78/G22 FPB only acquires temperature and humidity data by using the HS3001 sensor.

(2) The secondary OTA update mechanism is used to download the update firmware for the RL78/G22 FPB from AWS via the CK-RX65N and then update the firmware.

(3) After the firmware updating is complete, the RL78/G22 FPB acquires flow data from the FS3000 sensor as well as the 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 RL78/G22 FPB and connection of the HS3001 and FS3000 sensors, 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 display of the sensor data from AWS.

6.1 Setting up the Hardware

6.1.1 Overall Structure

Firstly, the overall hardware structure for this demonstration is shown below. For an actual image after setup is complete, see Figure 6-12. The methods for setting up each of the boards are described in detail in the subsequent subsections.

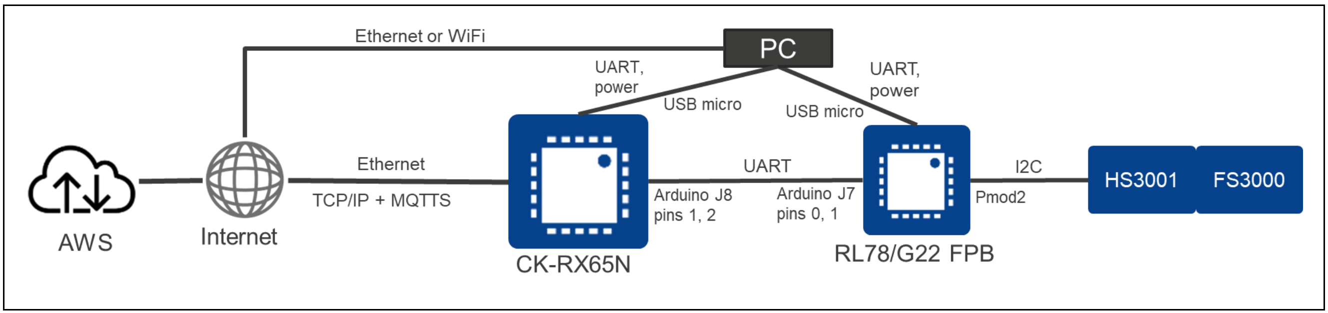


Figure 6-1 Overall Hardware Structure for This Demo

6.1.2 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 RL78/G22 FPB

TXD, RXD, and GND for UART communications with the RL78/G22 FPB are allocated to the following pins on the J8 and J3 connectors of the CK-RX65N. Connect the pins on the RL78/G22 FPB side as described in 6.1.3(2), with the corresponding UART signals listed in the table below.

Table 6-1 UART Connection between the CK-RX65N and RL78/G22 FPB

CK-RX65N			RL78/G22 FPB	
UART Signal	Connector Pin Name		UART Signal	Connector Pin Name
TXD	J8-2 (TX▸1)	→	RXD	J7-1
RXD	J8-1 (RX◂0)	←	TXD	J7-2
GND	J3-7 (GND)	—	GND	J8-7

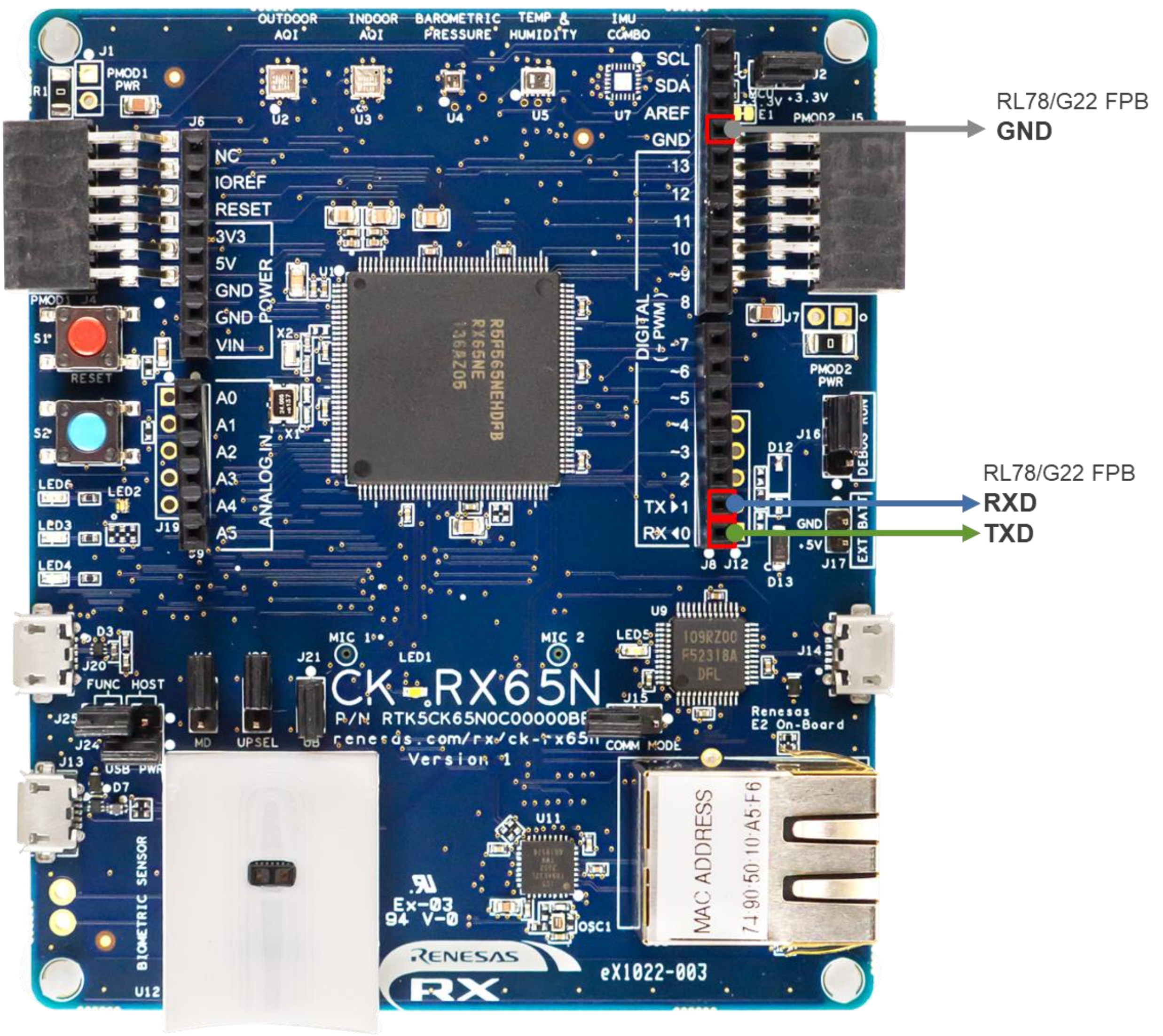


Figure 6-2 Locations of Pins on the CK-RX65N Used for UART Communications between the Microcontrollers

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

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

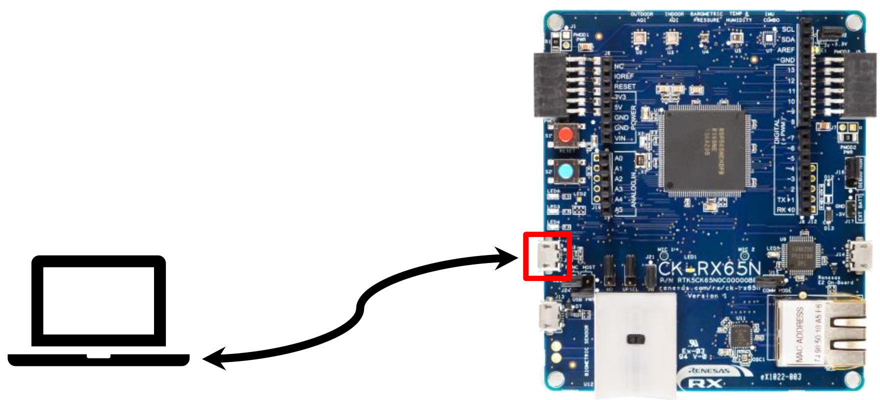


Figure 6-3 Connection for Log Output to the PC

(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.

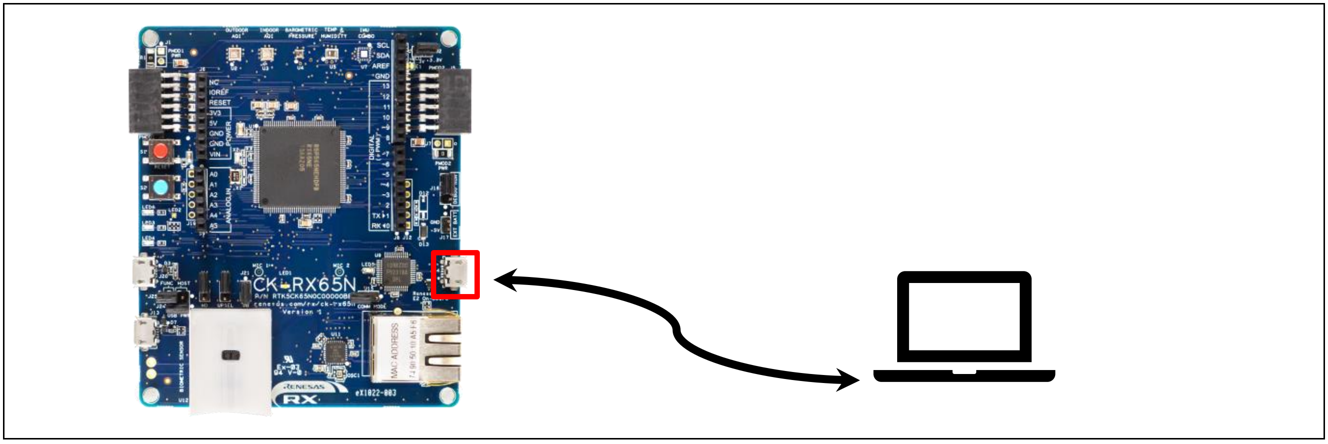


Figure 6-4 Connection of the Power Supply and Debugger

(4) Connecting the LAN cable for Internet connection

Connect the LAN cable connected with the Internet to the Ethernet connector on the CK-RX65N.

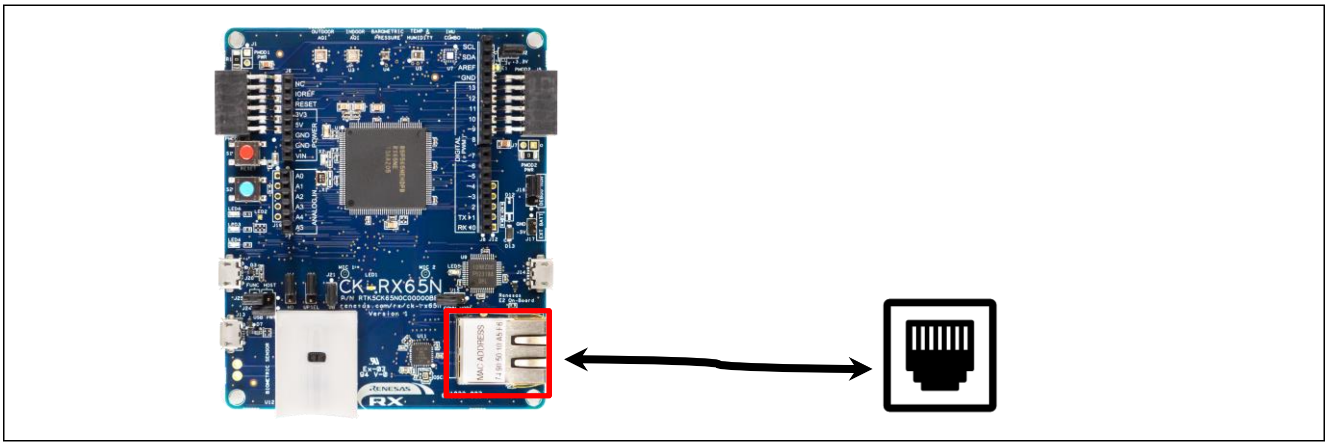


Figure 6-5 Wired Internet Connection with the Ethernet

(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).

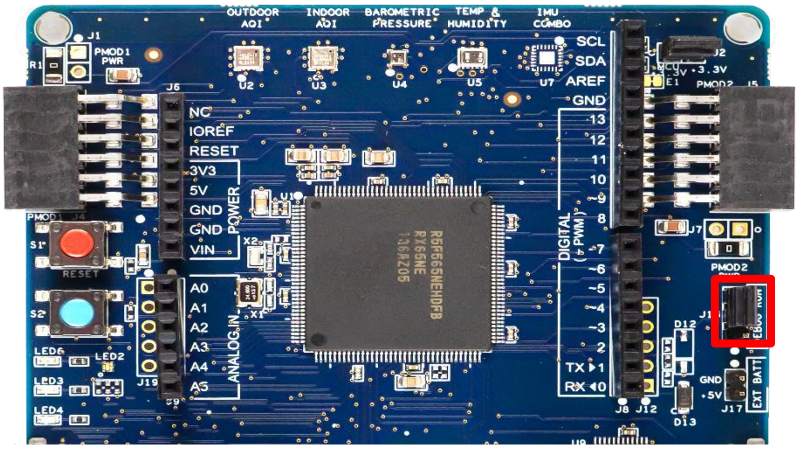


Figure 6-6 Location of Jumper Block J16

6.1.3 Setting up the RL78/G22 FPB

The procedure for setting up the RL78/G22 FPB is described in the following passages.

(1) Setting the operating voltage to 3.3 V

To operate the RL78/G22 from a 3.3-V power supply, close the power selection header (J17) on the 2-3 side.

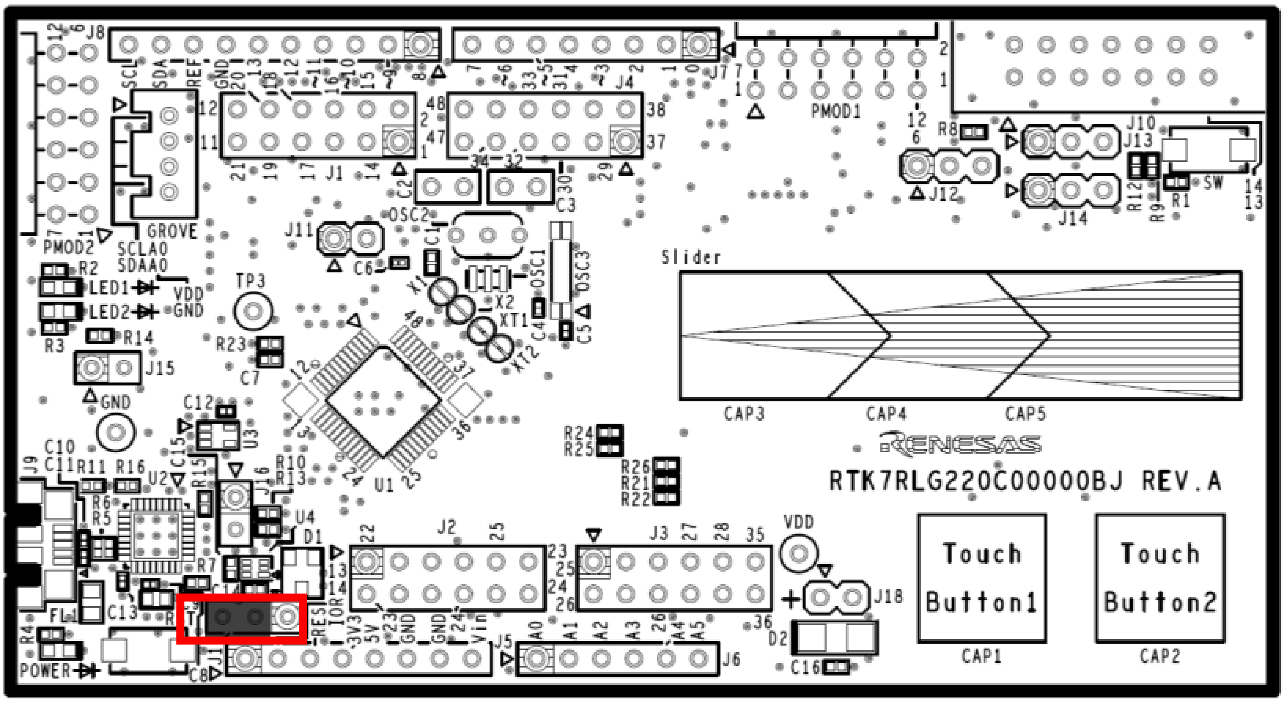


Figure 6-7 Location of the Power Selection Header (J17)

(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 J7 and J8 connectors of the RL78/G22 FPB. Connect the pins on the CK-RX65N side as described in 6.1.2(1), with the corresponding UART signals listed in Table 6-1.

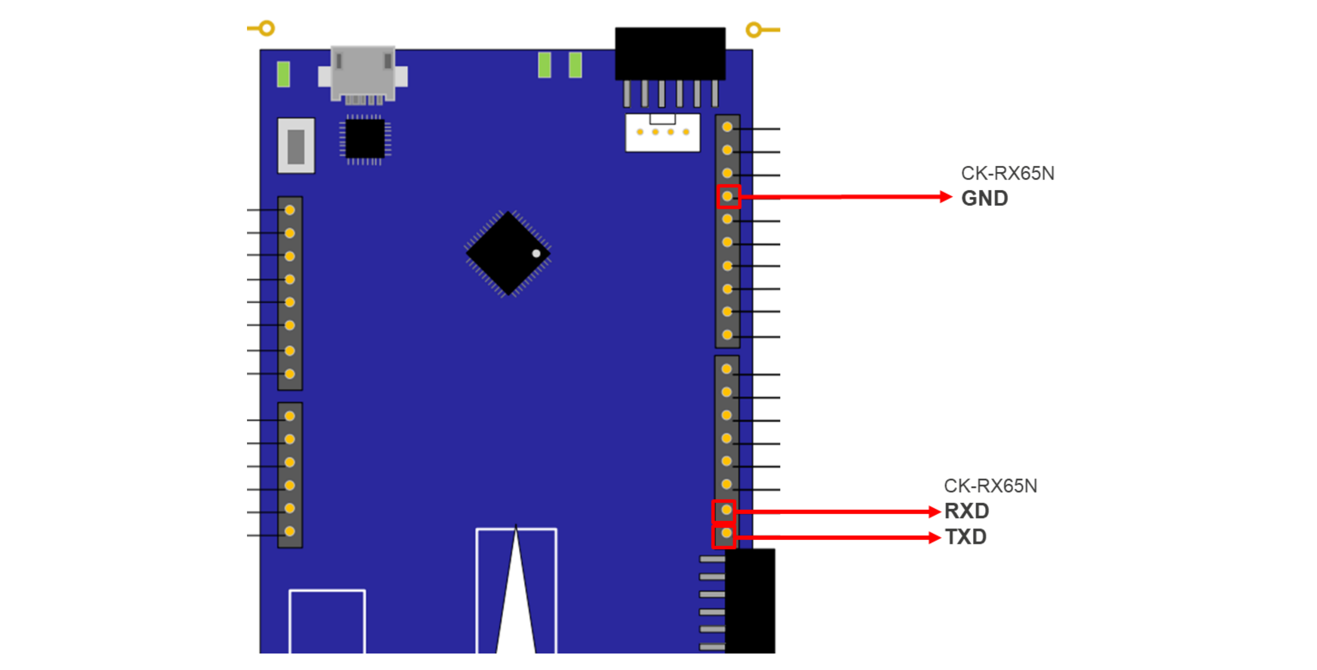


Figure 6-8 Locations of Pins on the RL78/G22 FPB Used for UART Communications between the Microcontrollers

(3) Connecting the cable for log output to the PC and power supply

Connect the PC and the micro USB Type-B connector on the RL78/G22 FPB with a USB cable.

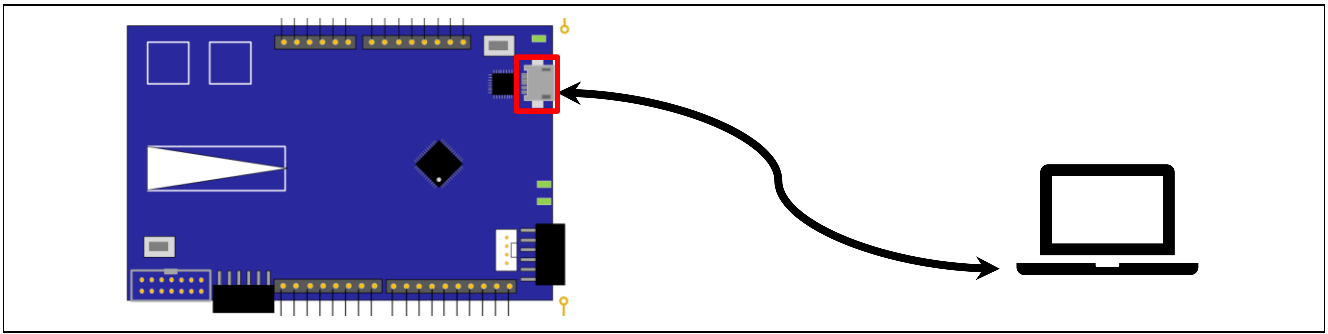


Figure 6-9 Connecting the PC and RL78/G22 FPB

(4) Connecting the HS3001 board and FS3000 board

Daisy-chain an HS3001 sensor board (hereafter referred to as "HS3001 board") and an FS3000 sensor board (hereafter referred to as "FS3000 board") to the Pmod2 connector on the RL78/G22 FPB. The order of connection of the sensor boards does not matter.

To pull up the I^{2}C bus signal, close the two jumper block pins (J4 and J5) on the HS3001 board, and similarly close the two jumper block pins J4 and J5 on the FS3000 board. If the boards do not operate properly in this state, open the jumper block pins on one of the boards and adjust the resistance value.

Since the Pmod2 connector on the RL78/G22 FPB is Pmod Interface Type 2A/3A, so is not directly connectable to the HS3001 board and FS3000 board, which are Type 6A modules, they must be wired as shown in Figure 6-10.

Alternatively, the RL78/G22 FPB has a cut pattern that enables use of the Pmod2 connector as Type-6A, so this method can also be used. Refer to the "Pmod™ Connectors" section in the RL78/G22 Fast Prototyping Board User's Manual for details.

Caution The US082-INTERPEVZ conversion board cannot be used with the RL78/G22 FPB.

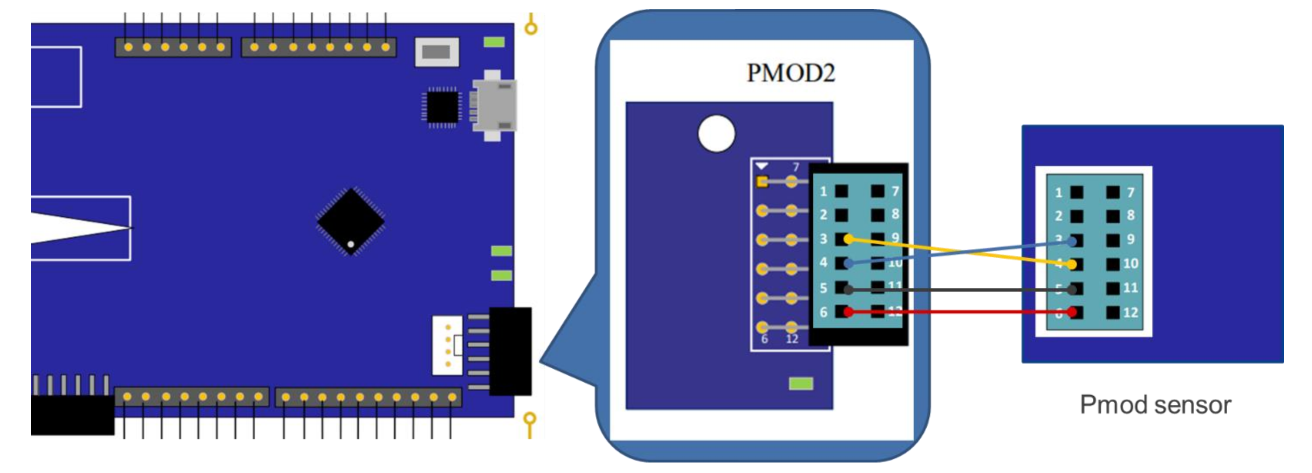


Figure 6-10 Connecting the Pmod2 Connector on the RL78/G22 FPB to Sensor Boards

(5) Opening the USB-to-serial converter reset header (J15)

To use Micro USB as a COM port, open the USB-to-serial converter reset header (J15).

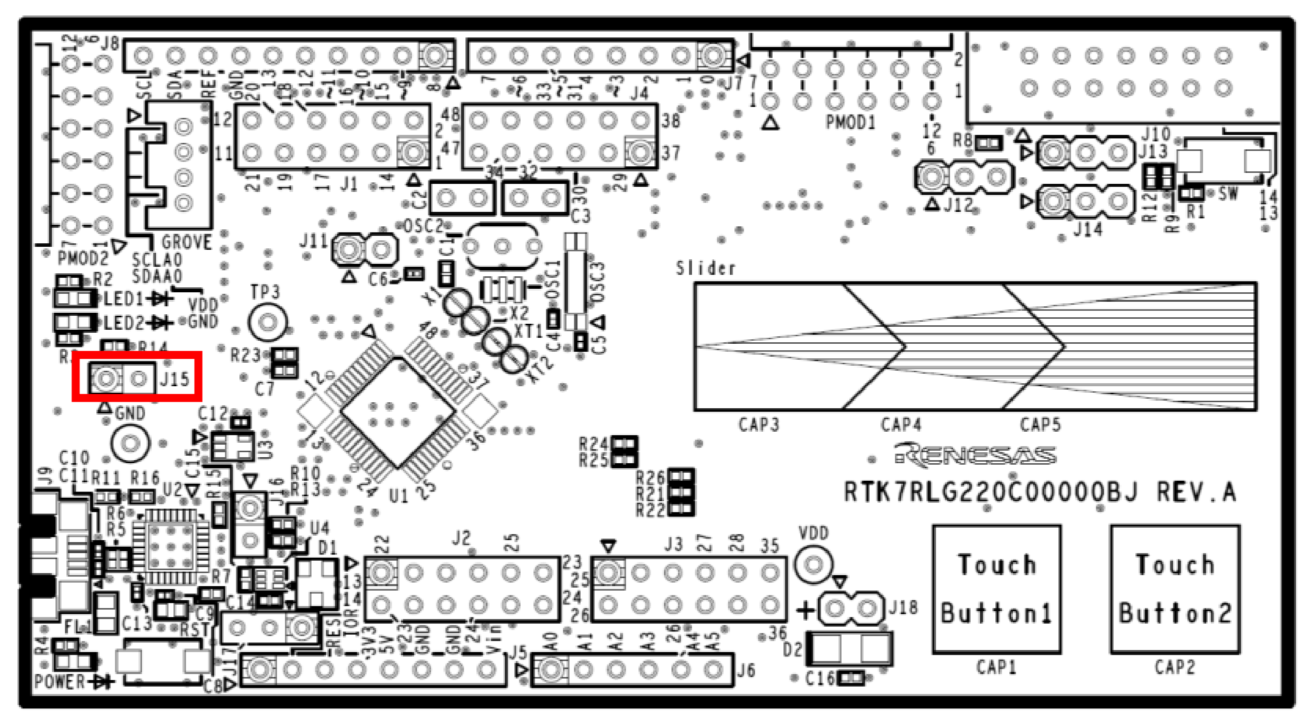


Figure 6-11 Location of the USB-to-Serial Converter Reset Header (J15)

The hardware setup for the demonstration is now completed. Figure 6-12 is an image of the overall configuration for the demonstration.

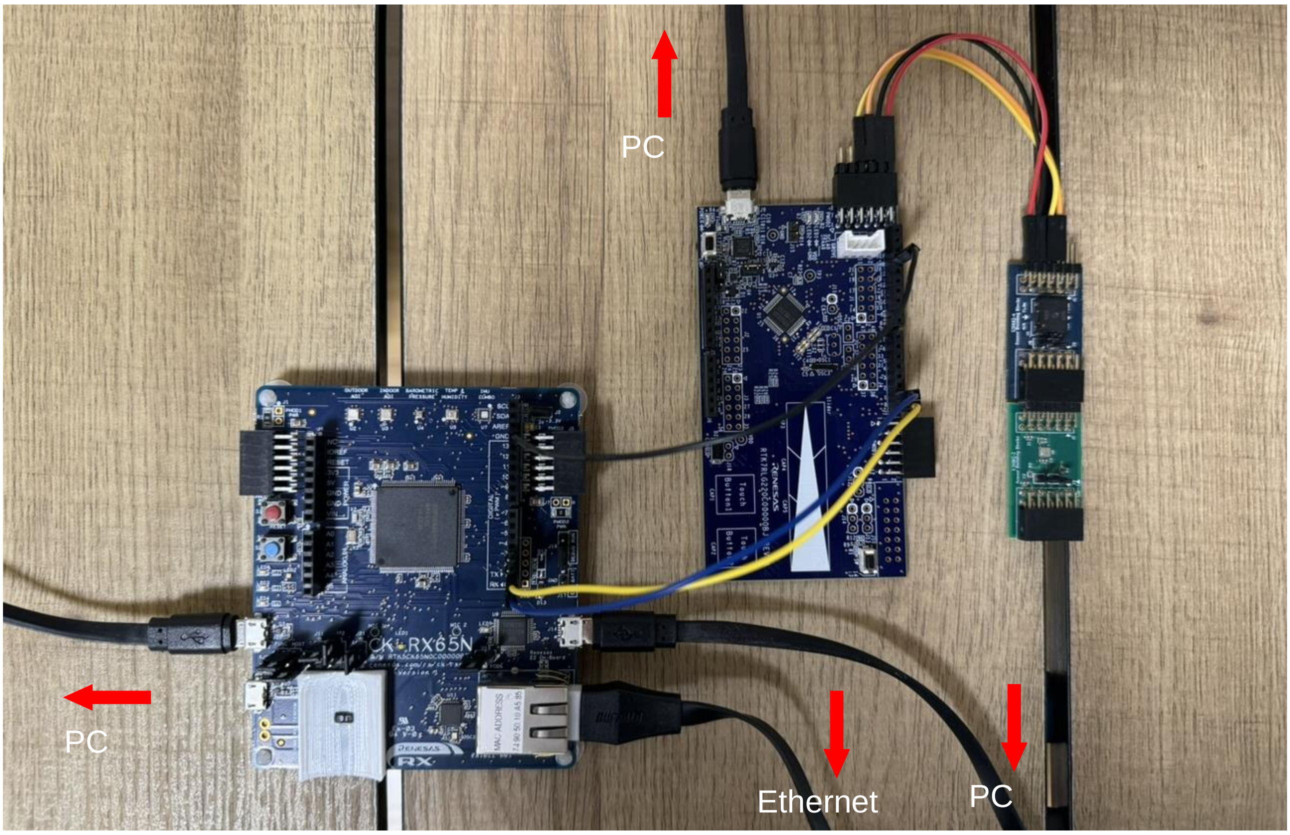


Figure 6-12 Image of the Overall Configuration for the Demo

6.2 Setting up the Software

6.2.1 Setting the Terminal Software

The terminal software (e.g., Tera Term) is required to generate log output using serial communication. The serial port settings are shown in the following.

Table 6-2 Serial Port Settings

Item	Setting
Baud rate	115,200 bps
Data	8 bits
Parity	None
Stop	1 bit
Flow control	None

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

Create initial firmware for the CK-RX65N by using QE for OTA and execute debugging in e^2 studio. The procedure is described below.

(1) Importing projects

Import the “ck_rx65n_demo_bootloader” project, a bootloader for the CK-RX65N, and the “ck_rx65n_2ndota_demo” project, a user program, into e^2 studio.

Click on [Import] from the [File] menu of e^2 studio. Select [Existing Projects into Workspace] and click on [Next].

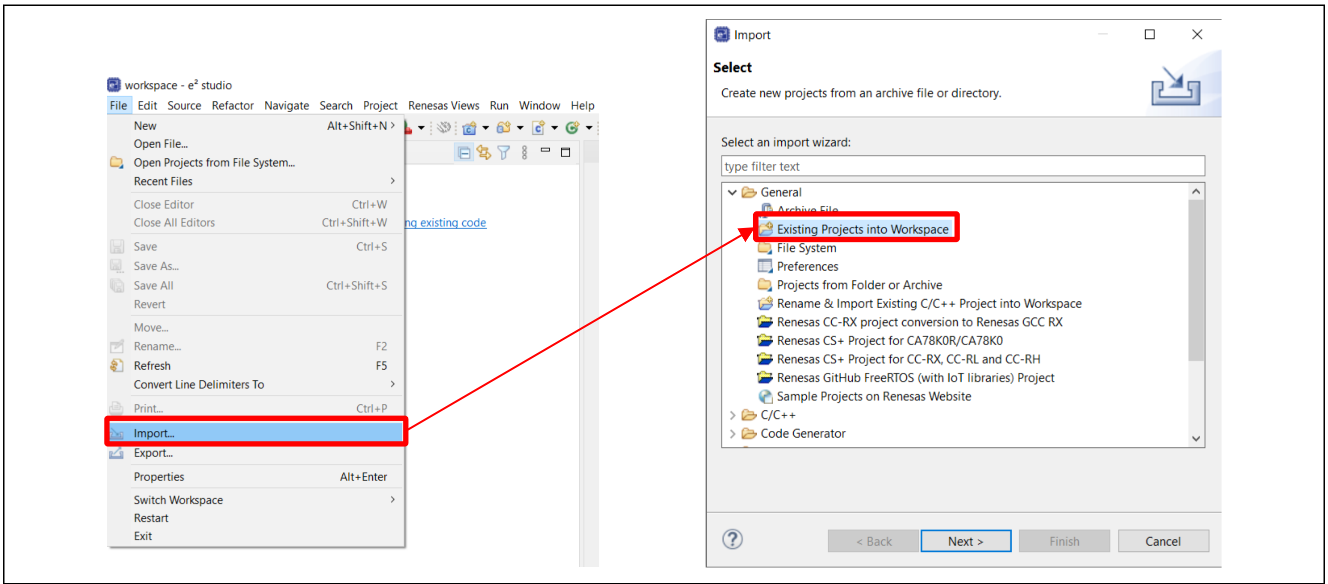


Figure 6-13 Procedure for Importing e^2 studio Projects (1)

Select the folder of the sample program in [Select root directory] and tick the checkboxes of “ck_rx65n_2ndota_demo” and “ck_rx65n_demo_bootloader” among the displayed projects. After making sure that the [Copy projects into workspace] checkbox in the [Options] field is not ticked, click on [Finish].

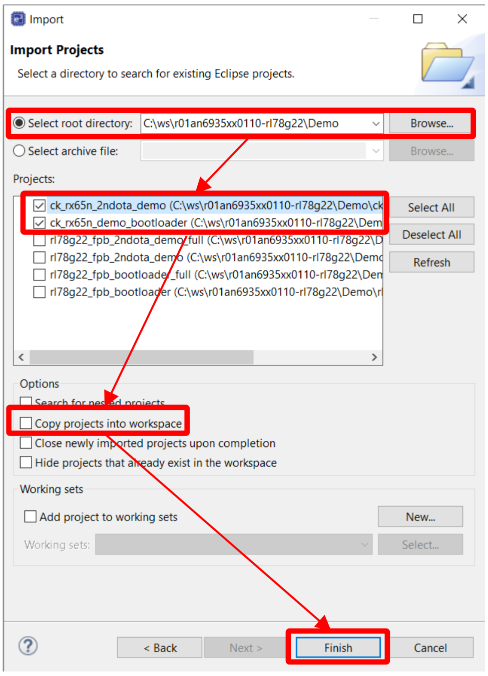


Figure 6-14 Procedure for Importing e² studio Projects (2)

(2) Opening the QE for OTA window

From the e² studio menu bar, select [Renesas Views] → [Renesas QE] → [OTA Main (QE)].

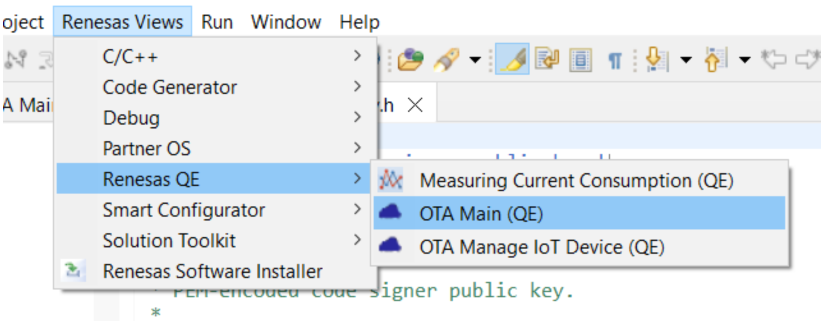


Figure 6-15 Opening the QE for OTA Window

(3) <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.

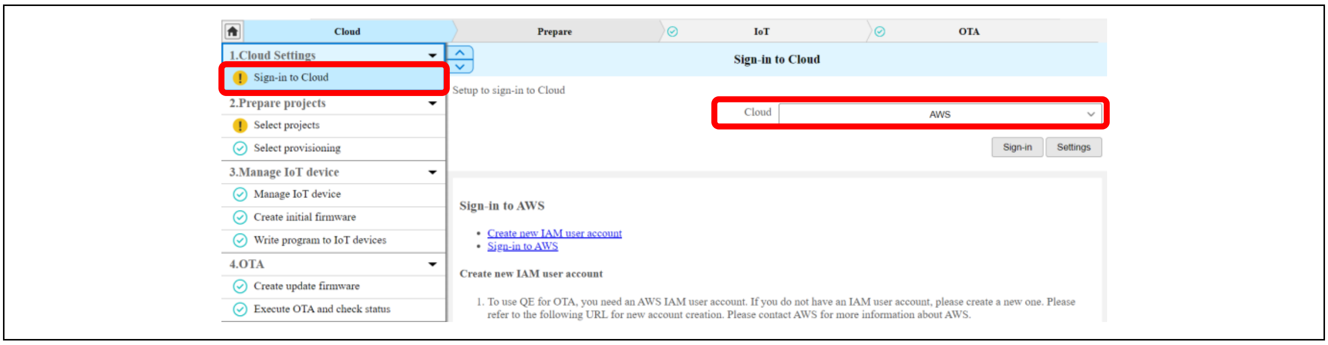


Figure 6-16 Signing in to AWS with QE for OTA

(4) <QE for OTA> [2. Prepare projects] → [Select projects]

Select the ck_rx65n_demo_bootloader project and the ck_rx65n_2ndota_demo project that were imported into e^2 studio earlier.

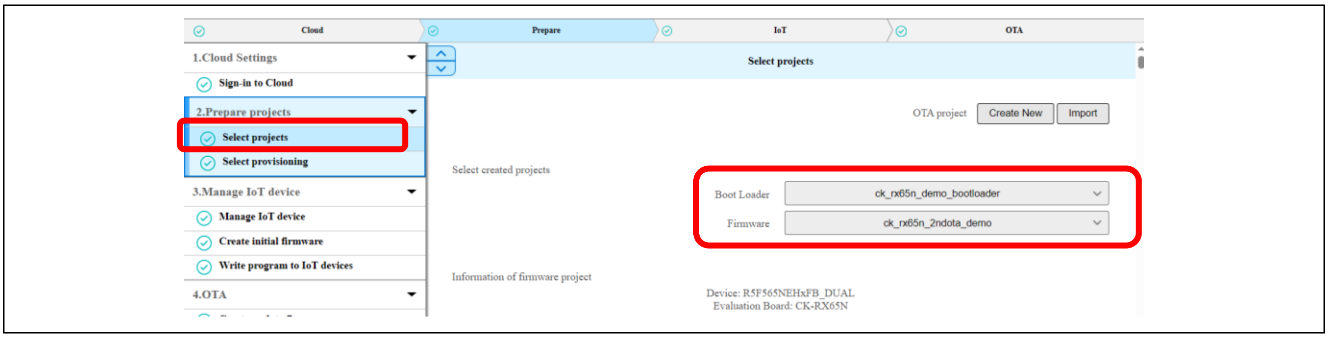


Figure 6-17 Selecting Projects

(5) <QE for OTA> [2. Prepare projects] → [Select provisioning]

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

Figure 6-18 Selecting the Provisioning Method

(6) <QE for OTA> [3. Manage IoT device] → [Manage IoT device]

Create an IoT device following the procedure of QE for OTA.

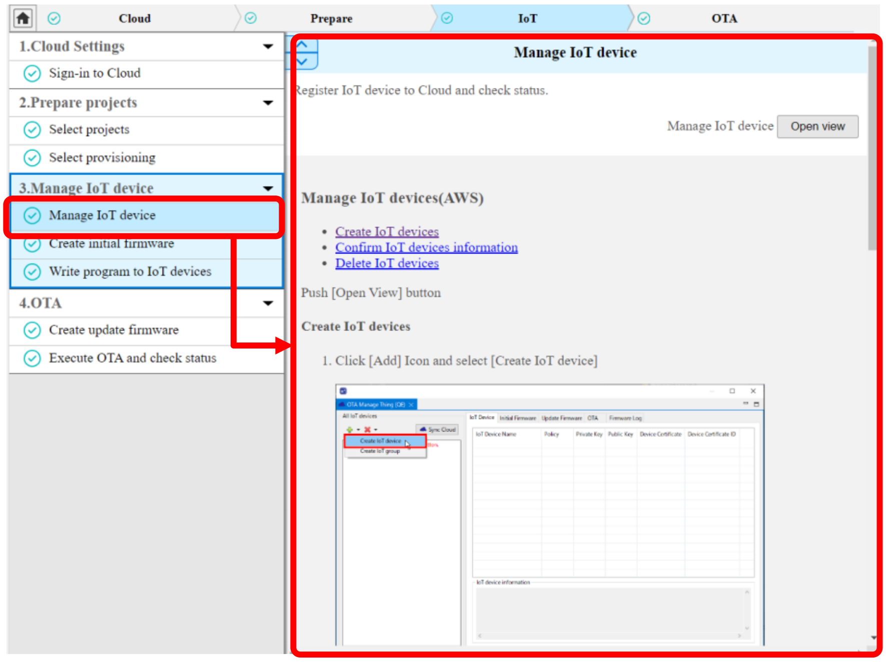


Figure 6-19 Creating an IoT Device

(7) <QE for OTA> [3. Manage IoT device] → [Create initial firmware]

Create initial firmware following the procedure of QE for OTA.

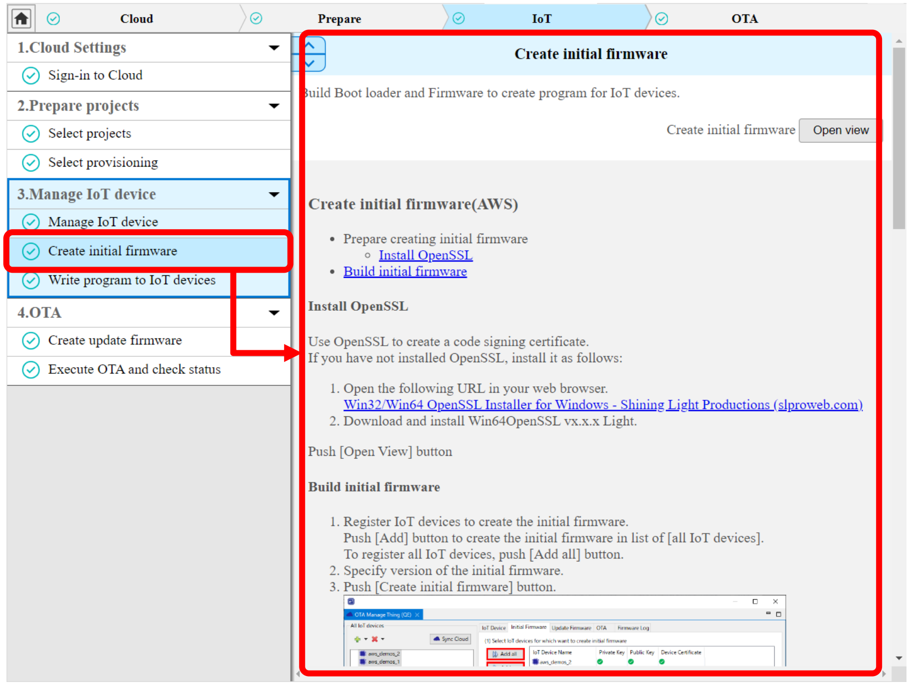


Figure 6-20 Creating the Initial Firmware

(8) <QE for OTA> [3. Manage IoT device] → [Write program to IoT devices]

Write the initial firmware to the CK-RX65N following the procedure of QE for OTA.

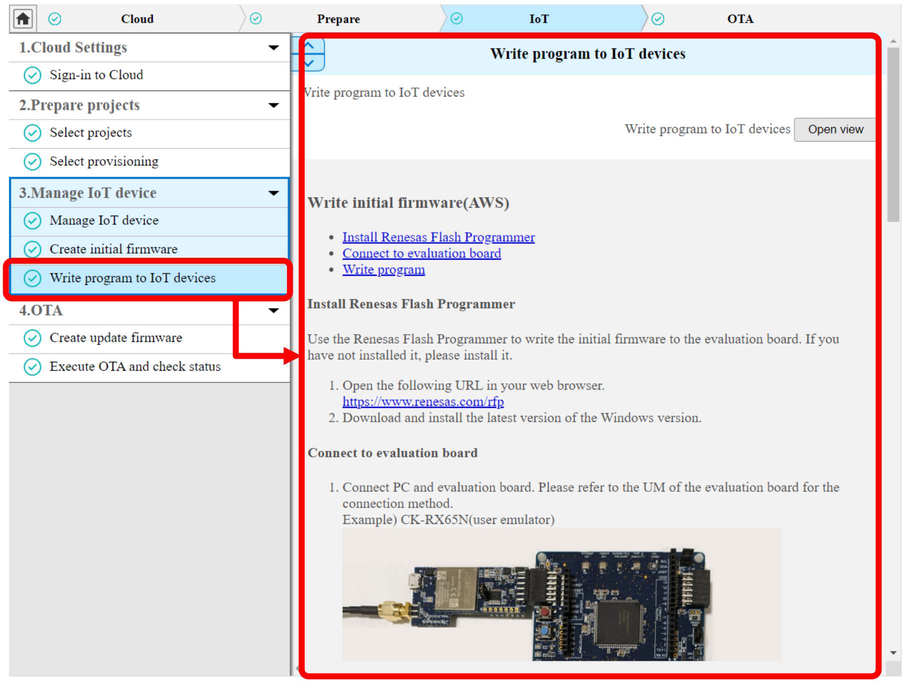


Figure 6-21 Writing the Initial Firmware

(9) Checking operation

Close jumper block J16 in Figure 6-6 on the RUN side (pins 2-3).

Launch the terminal software, and if the log is output as shown in Figure 6-22, the CK-RX65N is ready to run.

If the terminal software fails to connect to the COM port, open the “Firmware Log” view of QE for OTA and check if QE for OTA is not connected to the COM port.

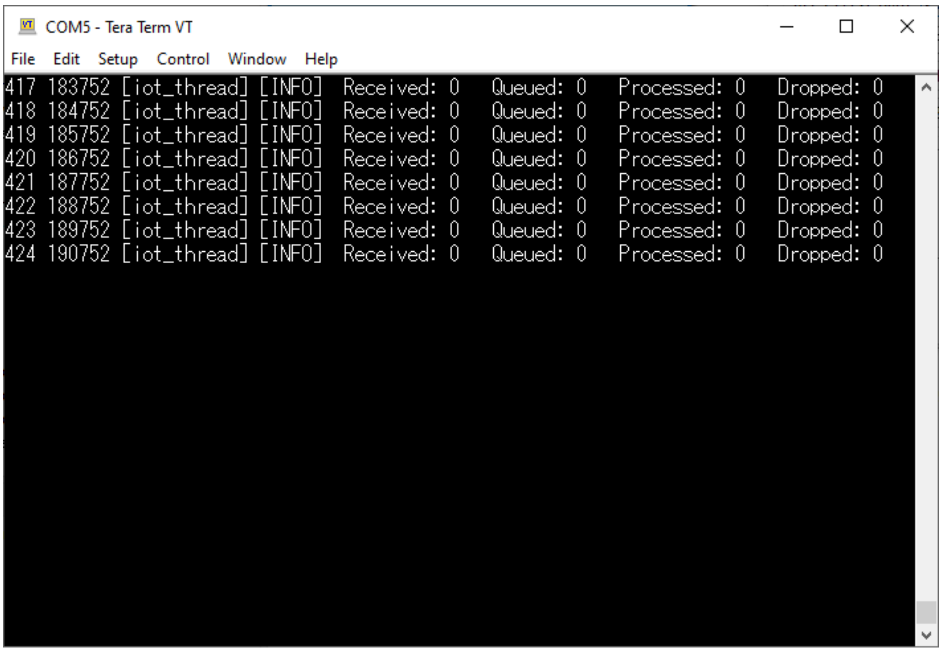


Figure 6-22 Log Screen of the CK-RX65N

6.2.3 Creating and Running the Initial Firmware for the RL78/G22 FPB

Create initial firmware for the RL78/G22 FPB and write it to the microcontroller by using the Renesas Flash Programmer. The procedure to be used in the case of the partial update method is described below. For the full update method, replace the names of the projects to be used, that is, replace the rl78g22_fpb_2ndota_demo project with the rl78g22_fpb_2ndota_demo_full project and the rl78g22_fpb_bootloader project with the rl78g22_fpb_bootloader_full project.

(1) Advance preparation

Execute the procedure in 5.2.2 in “5.2 Operating environment preparation” of “RL78/G22,RL78/G23,RL78/G24 Firmware Update Module Rev.2.01” to build the execution environment of the Renesas Image Generator.

(2) Importing projects

Import the “rl78g22_fpb_bootloader” project, which is a bootloader for the RL78/G22 FPB, and the “rl78g22_fpb_2ndota_demo” project, which is a user program, into e² studio.

Click on [Import] from the [File] menu of e² studio.

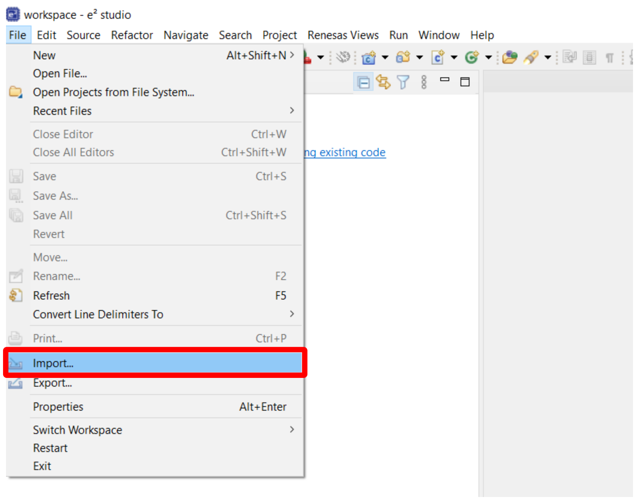


Figure 6-23 Procedure for Importing e² studio Projects (1)

Select [Existing Projects into Workspace] and click on [Next].

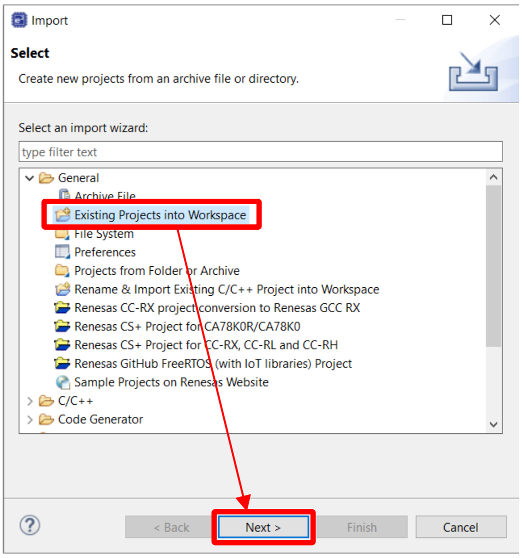


Figure 6-24 Procedure for Importing e² studio Projects (2)

Select the folder of the sample program in [Select root directory] and tick the checkboxes of "rl78g22_fpb_2ndota_demo" and "rl78g22_fpb_bootloader" among the displayed projects. After making sure that the [Copy projects into workspace] checkbox in the [Options] field is not ticked, click on [Finish].

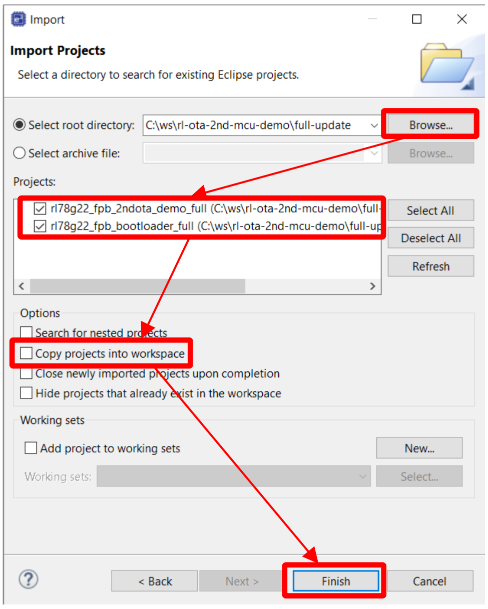


Figure 6-25 Procedure for Importing e² studio Projects (3)

(3) Building projects

Build the rl78g22_fpb_bootloader project and the rl78g22_fpb_2ndota_demo project and create a MOT file for each. The MOT files are created in the HardwareDebug folder directly under the project folder.

(4) Creating the initial firmware

The initial firmware for the RL78/G22 FPB is created by combining the created MOT files of the rl78g22_fpb_bootloader and rl78g22_fpb_2ndota_demo projects. The Renesas Image Generator is a tool for use in combining MOT files and is included with the "RL78/G22,RL78/G23,RL78/G24 Firmware Update Module Rev.2.01". For details, refer to the "Renesas Image Generator" section in the application note at the link above.

Execute the following command in the r01an6935jj0100-rl78g22/RenesasImageGenerator folder to create the initial firmware "initial_firm.mot".

```
> python image-gen.py -ip .\RL78_G22_ImageGenerator_PRM.csv -
ibp ..\rl78g22_fpb_bootloader\HardwareDebug\rl78g22_fpb_bootloader.mot -
iup ..\rl78g22_fpb_2ndota_demo\HardwareDebug\rl78g22_fpb_2ndota_demo.mot -o
initial_firm
```

The command to be used in the case of the full update method is as follows:

```
> python image-gen.py -ip .\RL78_G22_FullUpdate_ImageGenerator_PRM.csv -
ibp ..\rl78g22_fpb_bootloader_full\HardwareDebug\rl78g22_fpb_bootloader_full.mot -
iup ..\rl78g22_fpb_2ndota_demo_full\HardwareDebug\rl78g22_fpb_2ndota_demo_full.mot -o
initial_firm
```

(5) Writing the initial firmware

Using the Renesas Flash Programmer, write the initial firmware “initial_firm.mot” that was created in the previous step to the RL78/G22 FPB.

Launch the Renesas Flash Programmer and click on [New Project] from the [File] menu.

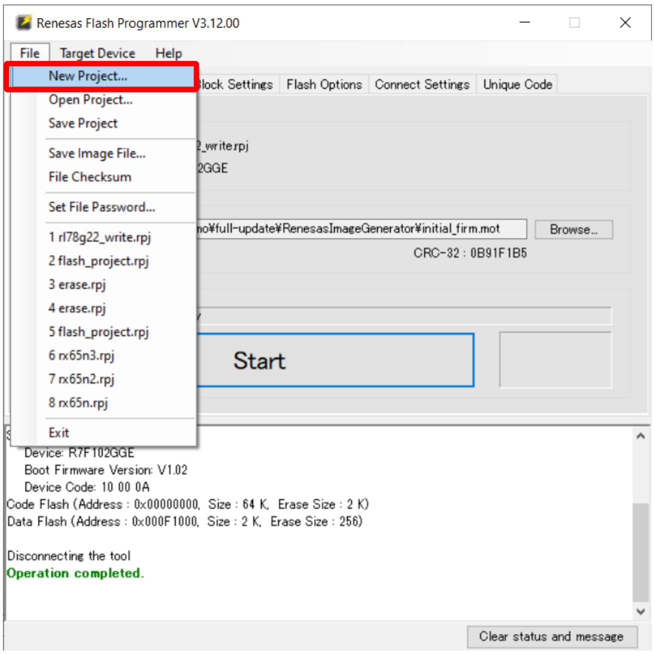


Figure 6-26 Procedure for Writing the MOT File (1)

Select “RL78/G2x” for [Microcontroller] and enter a project name in [Project Name].

Next, select “COM port” for [Tool] and “2 wire UART” for [Interface] in the [Communication] field.

Click on [Tool Details], then select the virtual COM port of the RL78/G22 FPB and click on [OK].

Finally, click on [Connect] and confirm that “**Operation completed.**” will be displayed.

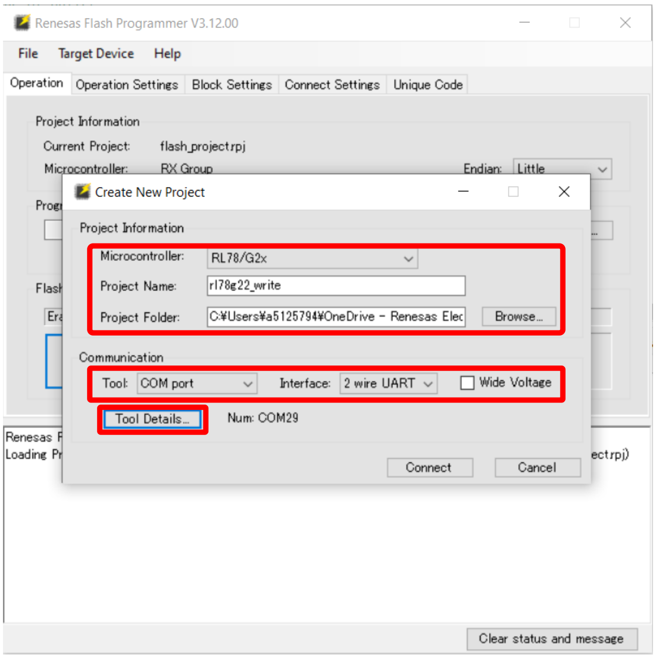


Figure 6-27 Procedure for Writing the MOT File (2)

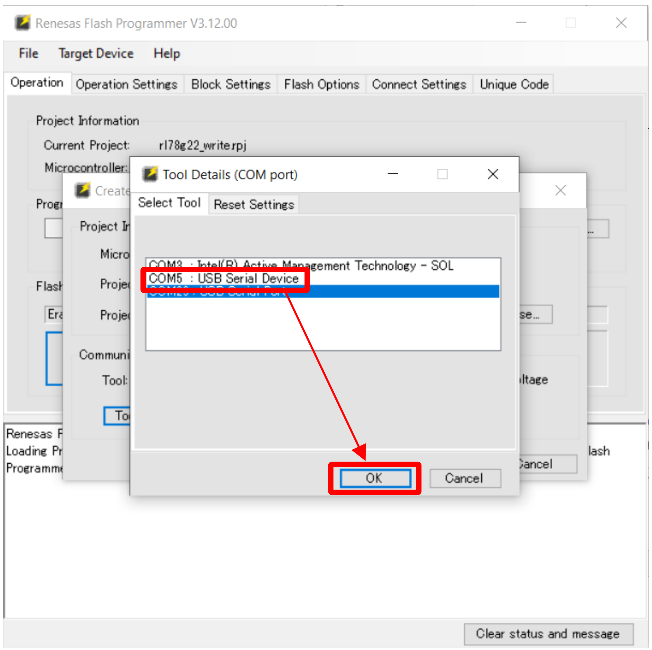


Figure 6-28 Procedure for Writing the MOT File (3)

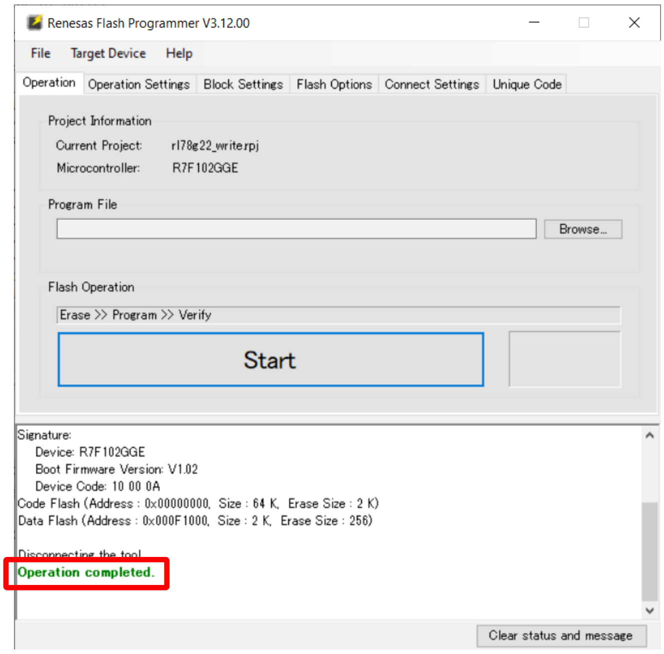


Figure 6-29 Procedure for Writing the MOT File (4)

Click on [Browse...] in the [Program File] field and select “initial_firm.mot” that was created by the Renesas Image Generator.

Click on [Start] to start the write operation.

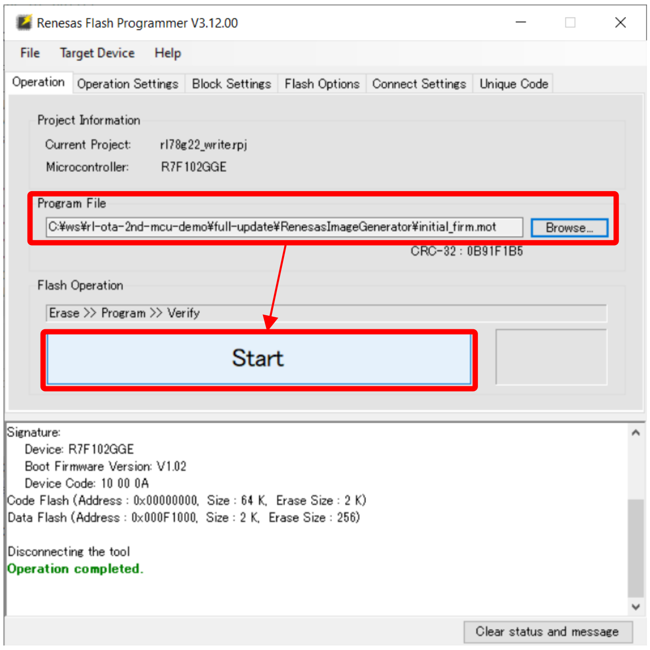


Figure 6-30 Procedure for Writing the MOT File (5)

(6) Checking operation

Launch the terminal software, and if the values for temperature and humidity are output in the log as shown in Figure 6-31, the RL78/G22 FPB is ready to run.

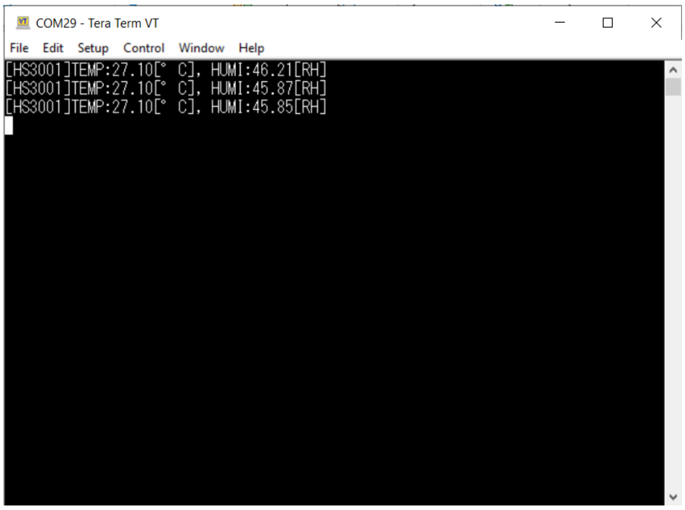


Figure 6-31 Log Screen of the RL78/G22 FPB

6.3 Preparations for Using the AWS Cloud

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.

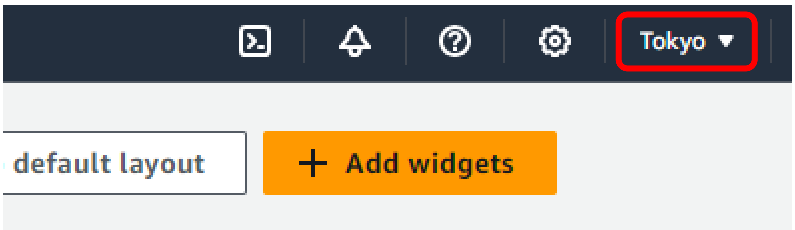


Figure 6-32 Confirming the Region

6.3.1 Settings for the OTA Update

Refer to the "RX Family How to implement FreeRTOS OTA using Amazon Web Services in RX65N (for v202210.01-LTS-rx-1.1.0 or later)" Application Note and make the necessary settings.

(1) Create an Amazon S3 bucket according to the procedure described in "3.4 Creating an Amazon S3 bucket" in the above application note. The bucket name set here will be used when running the demonstration.

(2) Create a service role according to the procedure described in "3.5 Allocating OTA execution permission to IAM users" in the above application note. The service role name set here will be used when running the demonstration.

(3) Register a code signing certificate according to the procedure described in (5) to (9) in "5.2 Updating the firmware" in the above application note. The code signing certificate to be registered here is the certificate created when the initial firmware for the CK-RX65N was created by using QE for OTA in 6.2.2(7).

The certificate is created in "ck_rx65n_demo_bootloader/QE-OTA/codesigning". Specify secp256r1.crt for the certificate, secp256r1.privateKey for the private key, and ca.crt for the certificate chain.

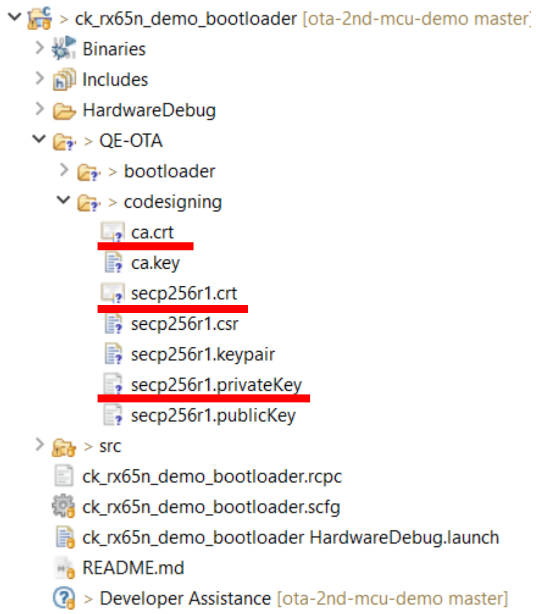


Figure 6-33 Location Where the Code Signing Certificate is Created

6.3.2 Settings for Displaying the Sensor Data

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 0.
In this case, as shown in Figure 6-34, 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.

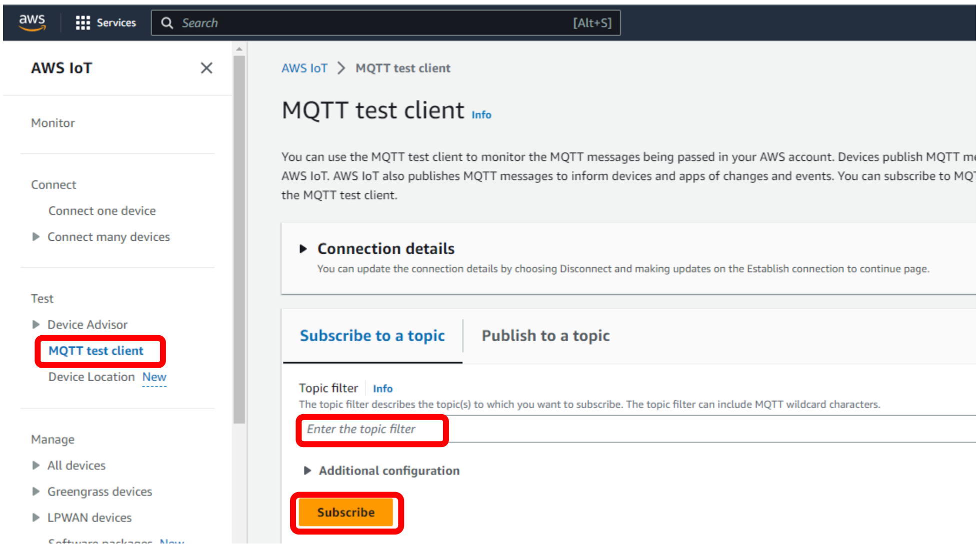
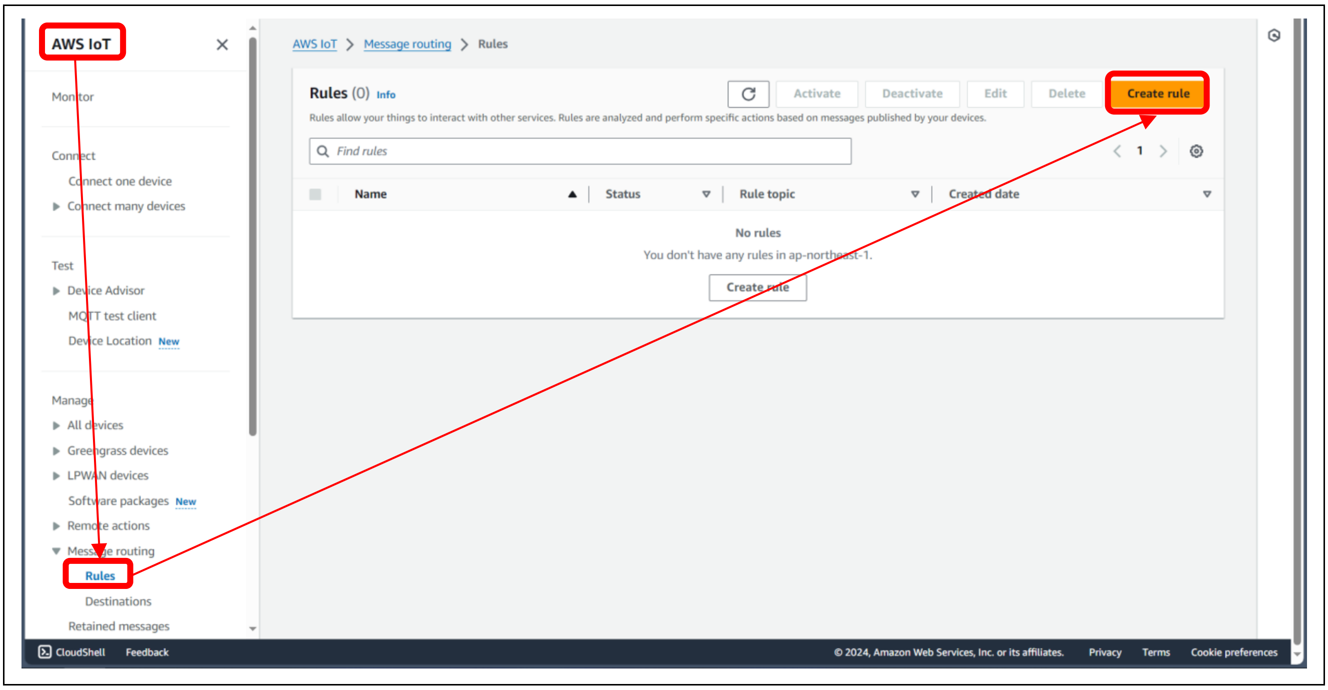


Figure 6-34 Confirming Data Reception by the MQTT Test Client

6.3.2.1 Setting up Amazon CloudWatch

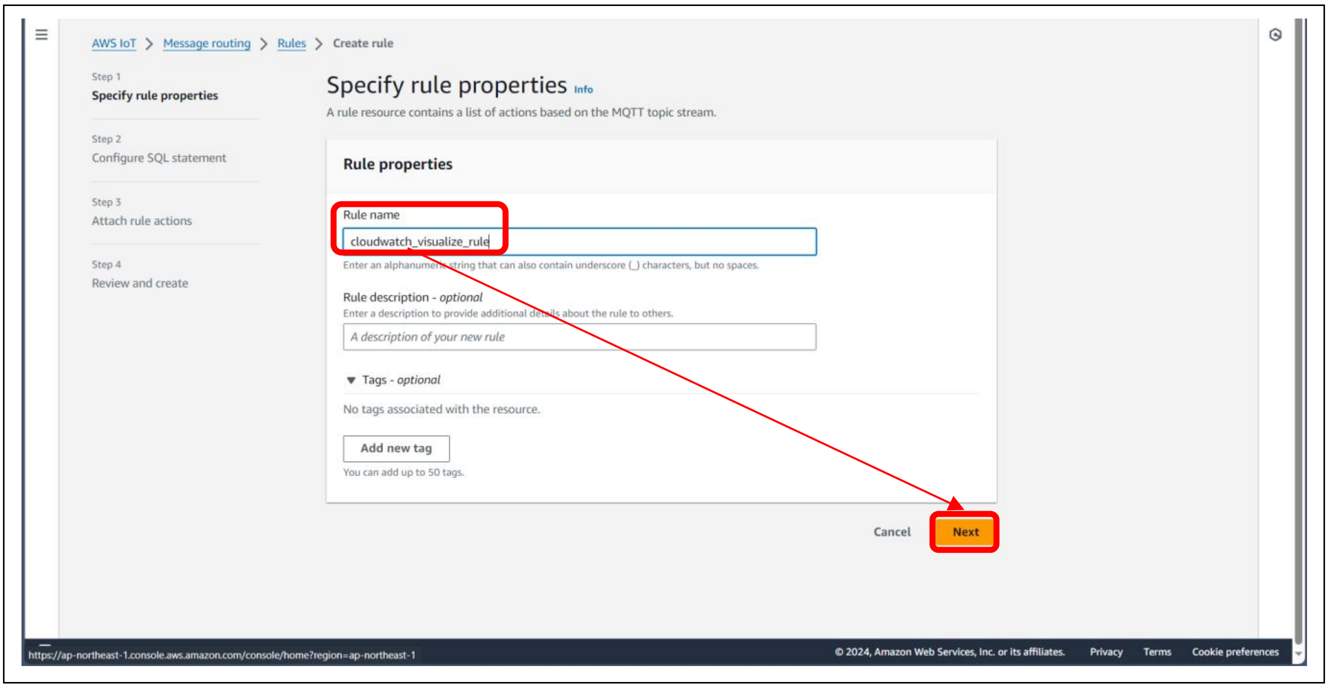
(1) Creating rules in AWS IoT

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



(2) Specifying the rule properties

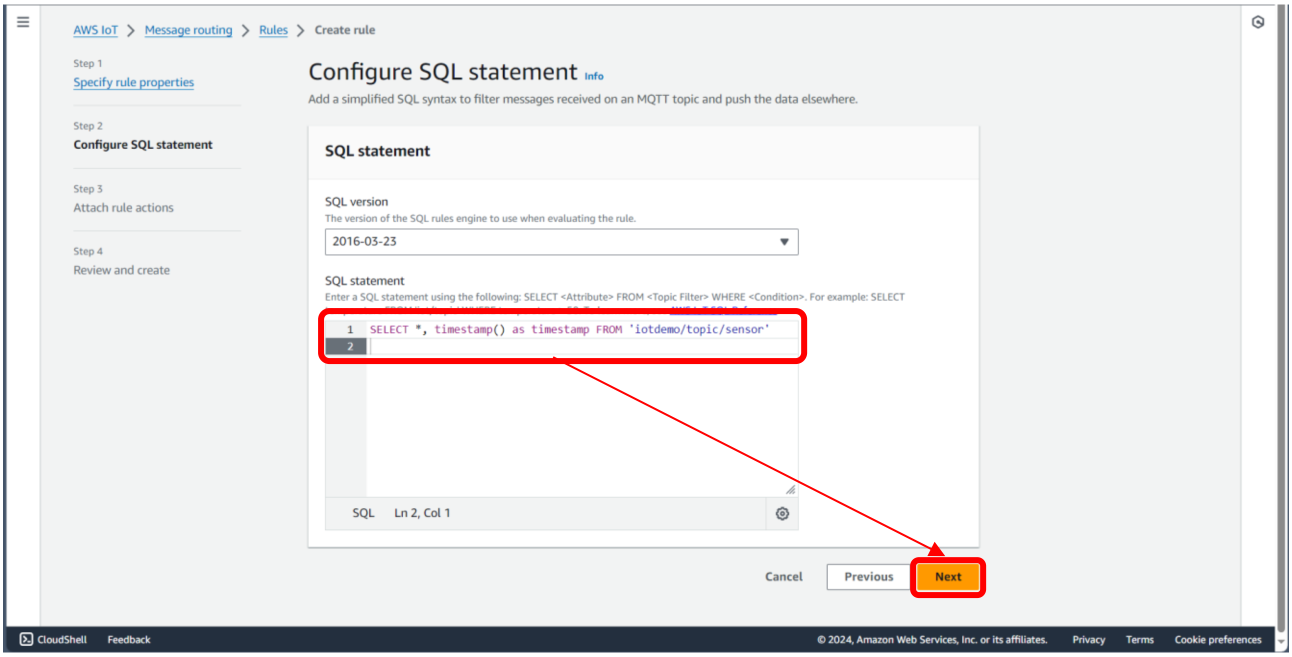
Enter a rule name in [Rule name] and click on [Next].



(3) Setting the SQL statement

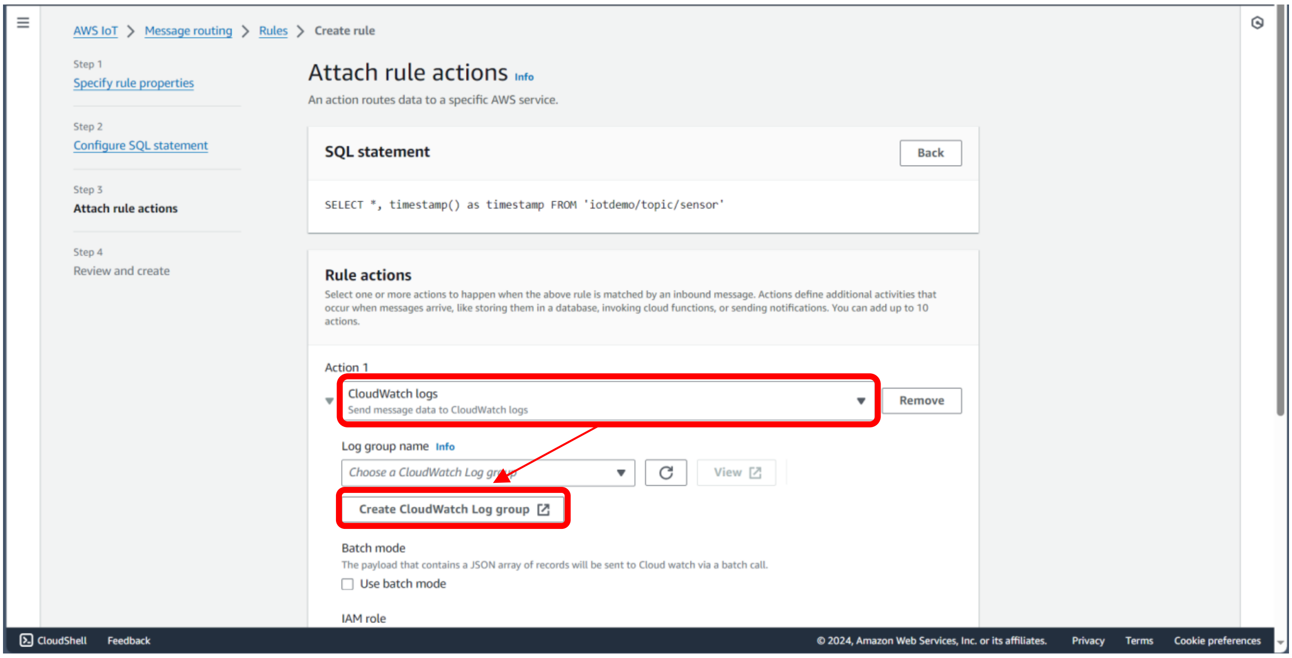
Enter the SQL statement by entering code like the following in the text editor field for [SQL statement]. Be sure to add a new line character at the end.

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



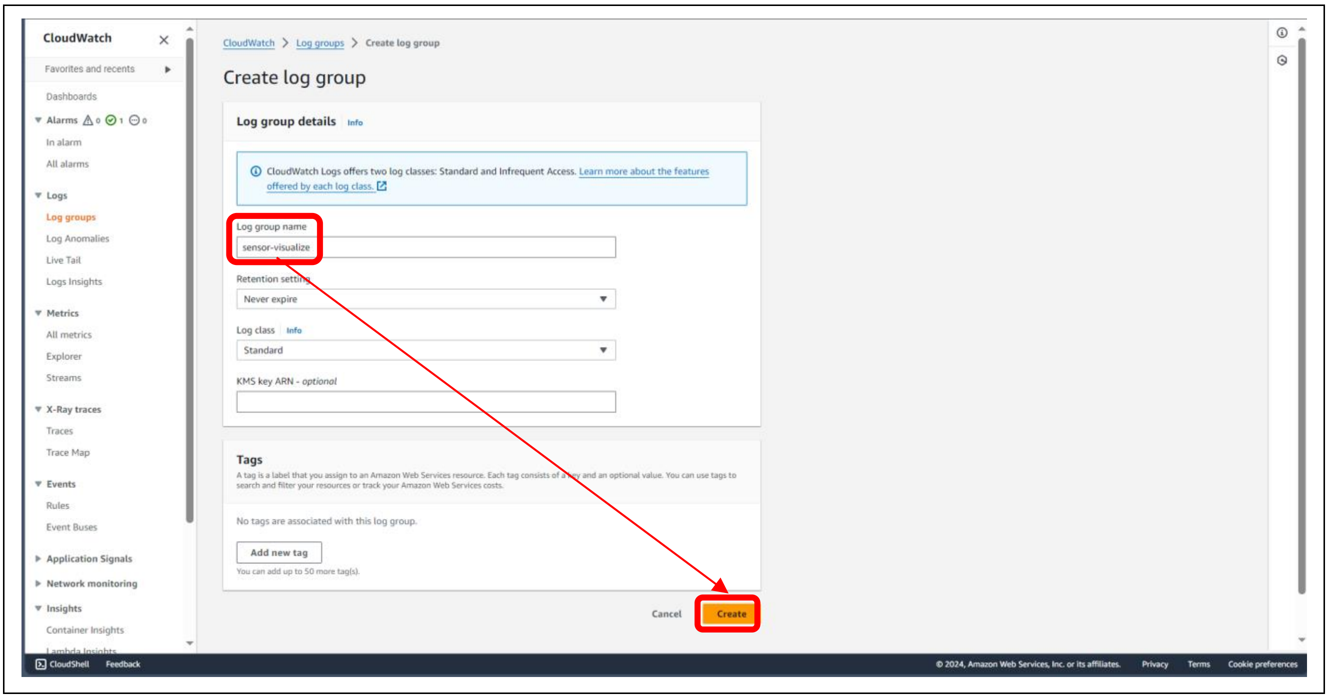
(4) Selecting rule actions in the [Attach rule actions] step

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



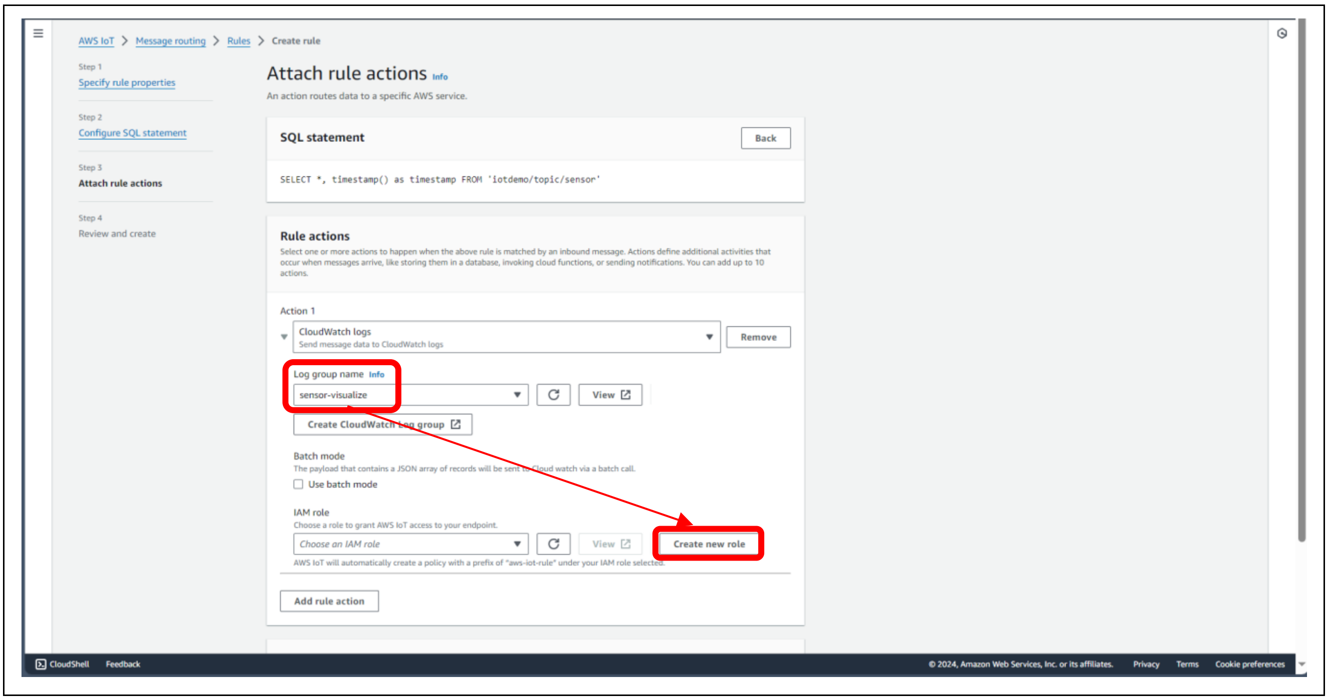
(5) Creating a log group

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



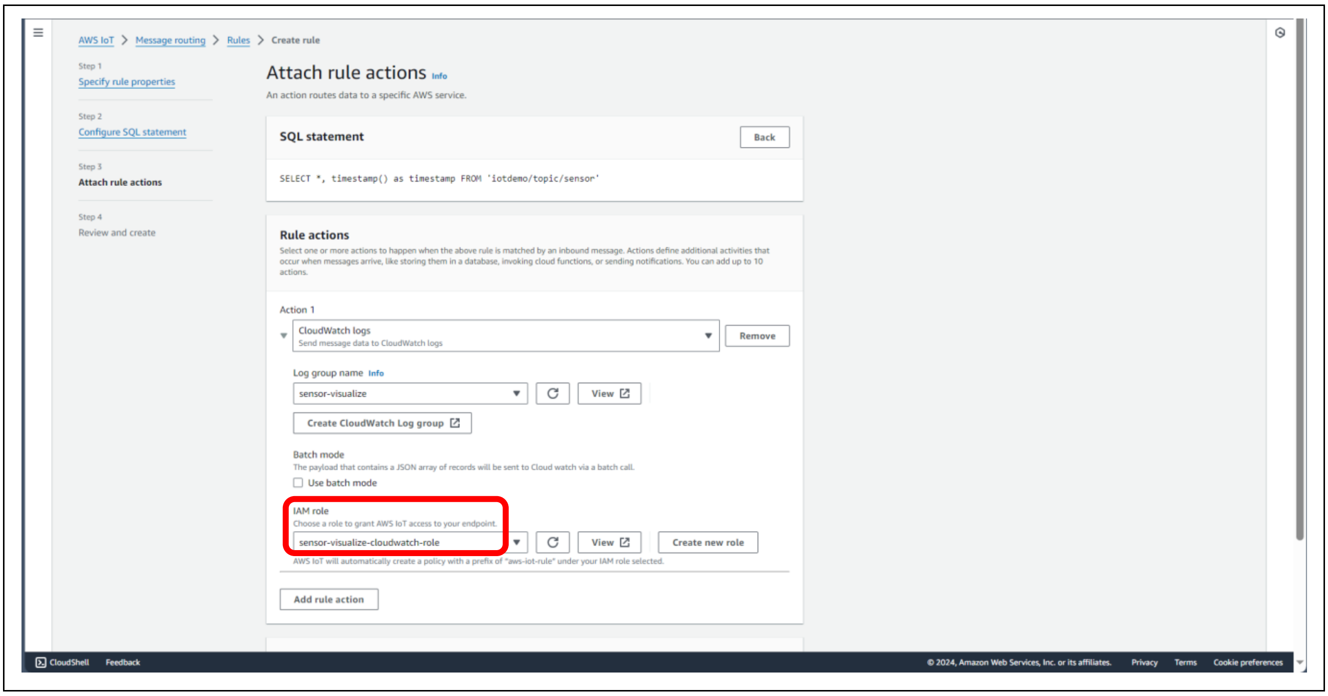
(6) Creating a new role

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



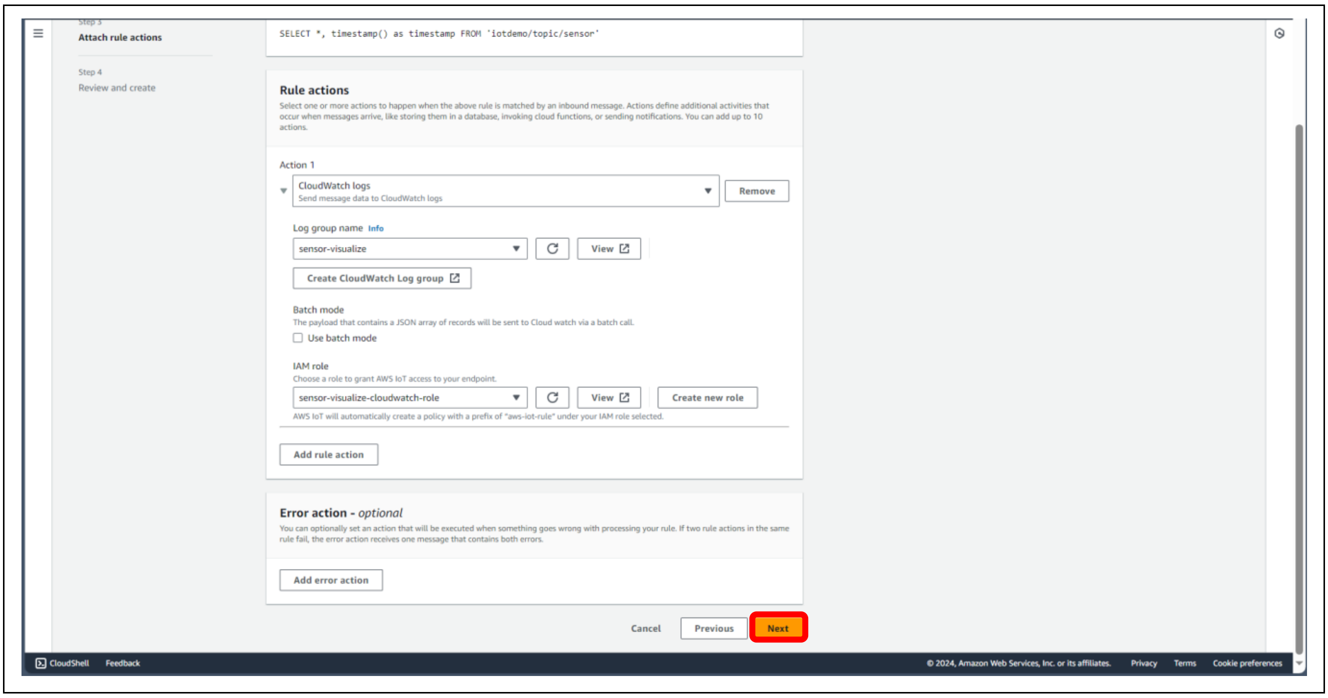
(7) Selecting the created IAM role

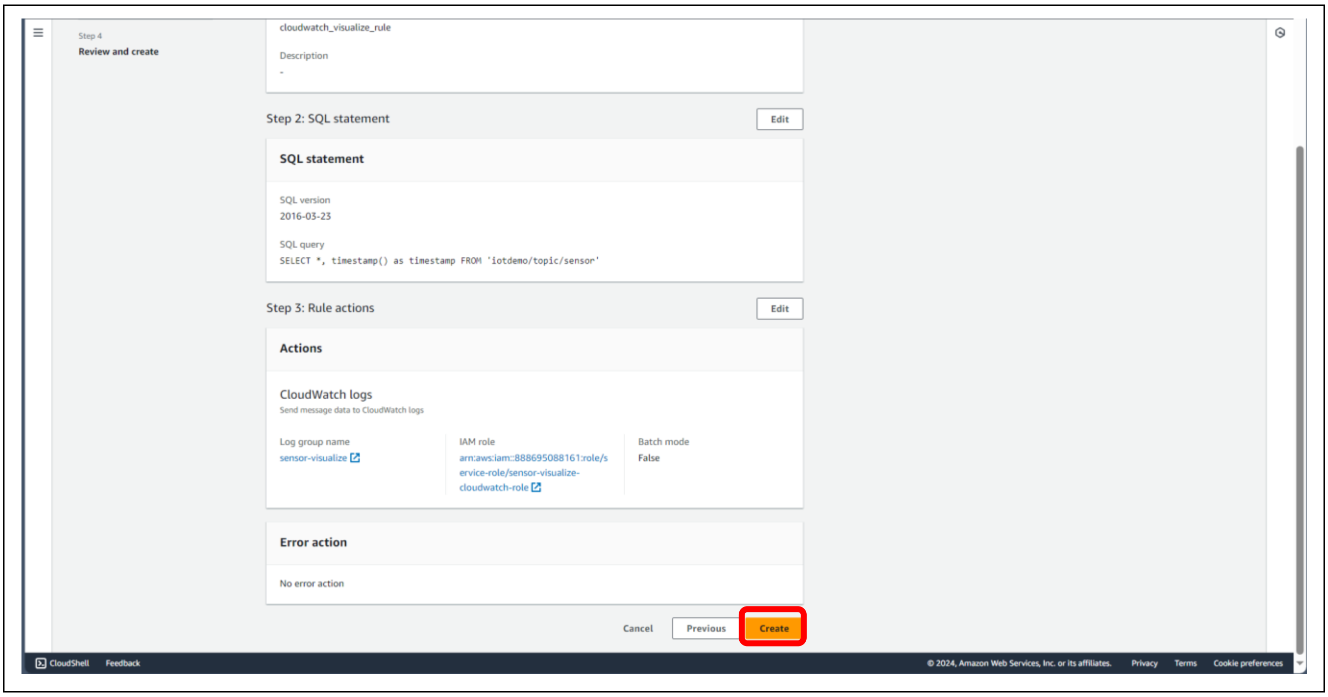
Select the created role in [IAM role].

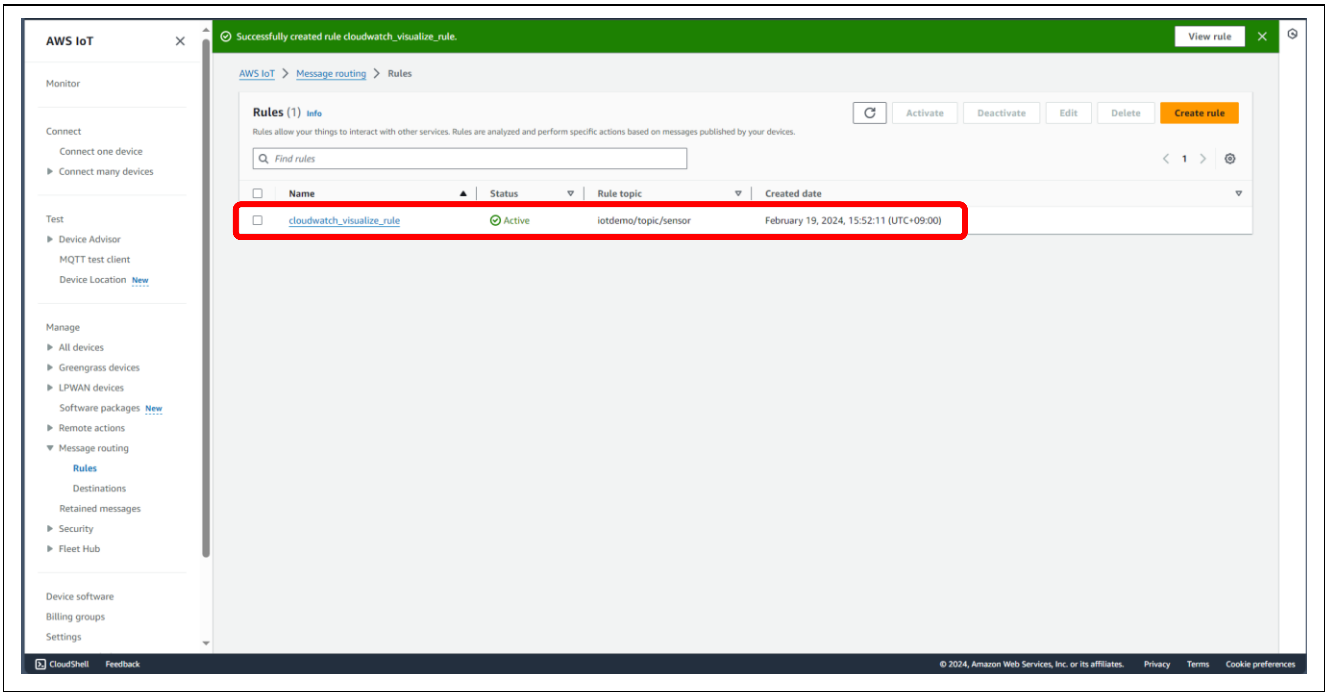


(8) Confirming successful creation of the rule

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.




Step 4
Review and create
cloudwatch_visualize_rule
Description
Step 2: SQL statement
Edit
SQL statement
SQL version
2016-03-23
SQL query
SELECT *, timestamp() as timestamp FROM 'iotdemo/topic/sensor'
Step 3: Rule actions
Actions
CloudWatch logs
Send message data to CloudWatch logs
Log group name
sensor-visualize
IAM role
arn:aws:iam::888695088161:role/service-role/sensor-visualize-cloudwatch-role
Batch mode
False
Error action
No error action
Cancel
Previous
Create


Successfully created rule cloudwatch_visualize_rule.
View rule
AWS IoT
AWS IoT > Message routing > Rules
Rules (1) Info
Rules allow your things to interact with other services. Rules are analyzed and perform specific actions based on messages published by your devices.
Activate
Deactivate
Edit
Delete
Create rule
Find rules
Name
Status
Rule topic
Created date
cloudwatch_visualize_rule
Active
iotdemo/topic/sensor
February 19, 2024, 15:52:11 (UTC+09:00)

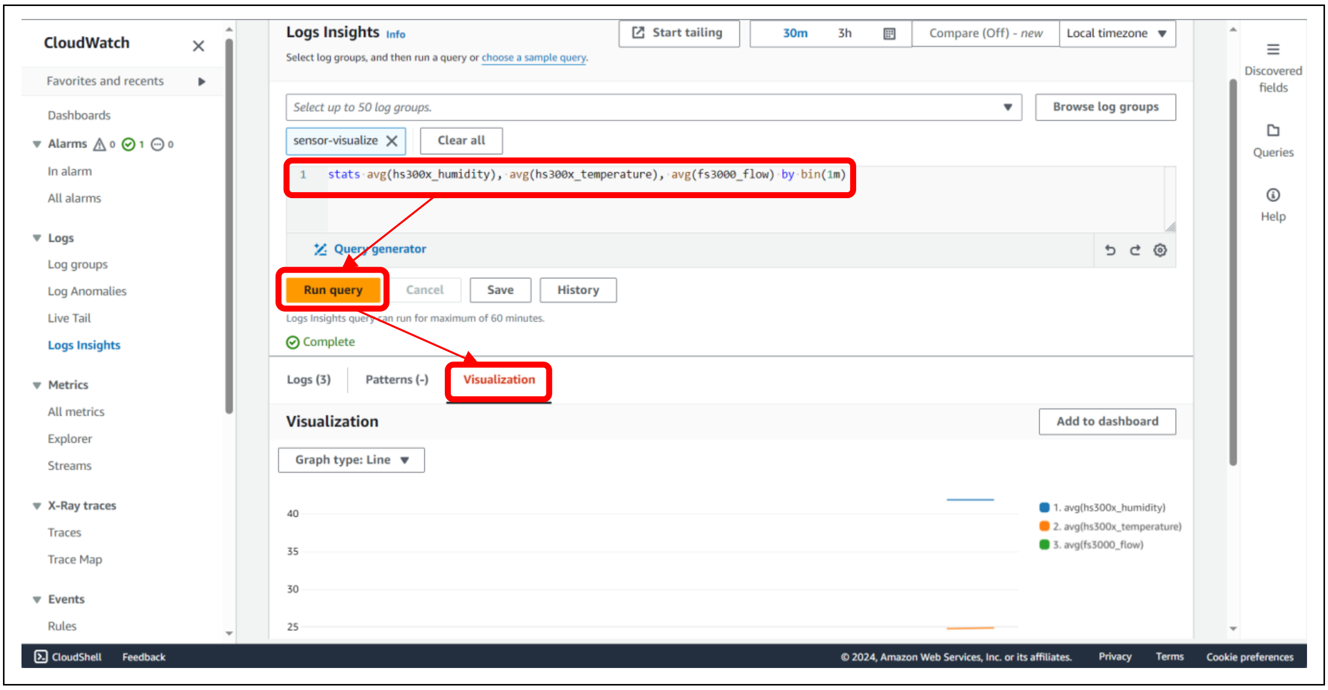
(9) Checking the graphical display in Amazon CloudWatch

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), avg(fs3000_flow) 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 in detail below.

7.1 Checking the Initial State of Operation

With the setup for the demonstration described in section 0 completed, press the reset switch (RST) on the RL78/G22 FPB to apply a hardware reset. Similarly, press the reset switch (S1) on the CK-RX65N to apply a hardware reset.

Check the logs from each of the microcontrollers by using terminal software.

Figure 7-1 shows the log screen of the CK-RX65N. Confirm that data from the HS3001 sensor are being output. Since the FS3000 sensor is not acquiring data, the value for that is 0.00. Below that, you can also see the log of sensor data being sent to AWS via MQTT communications.

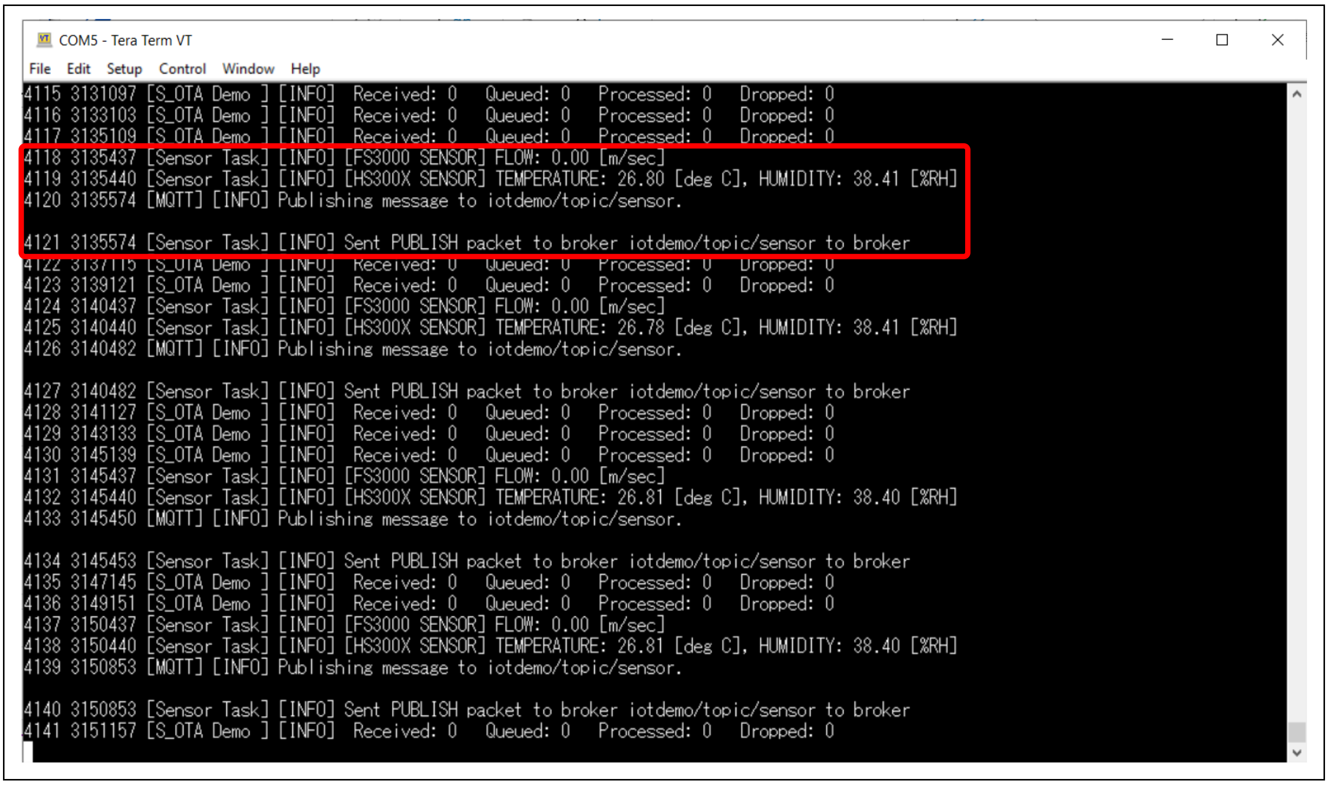


Figure 7-1 Log Screen of the CK-RX65N

Next, Figure 7-2 shows the log screen of the RL78/G22 FPB. Confirm that only data from the HS3001 sensor are displayed.

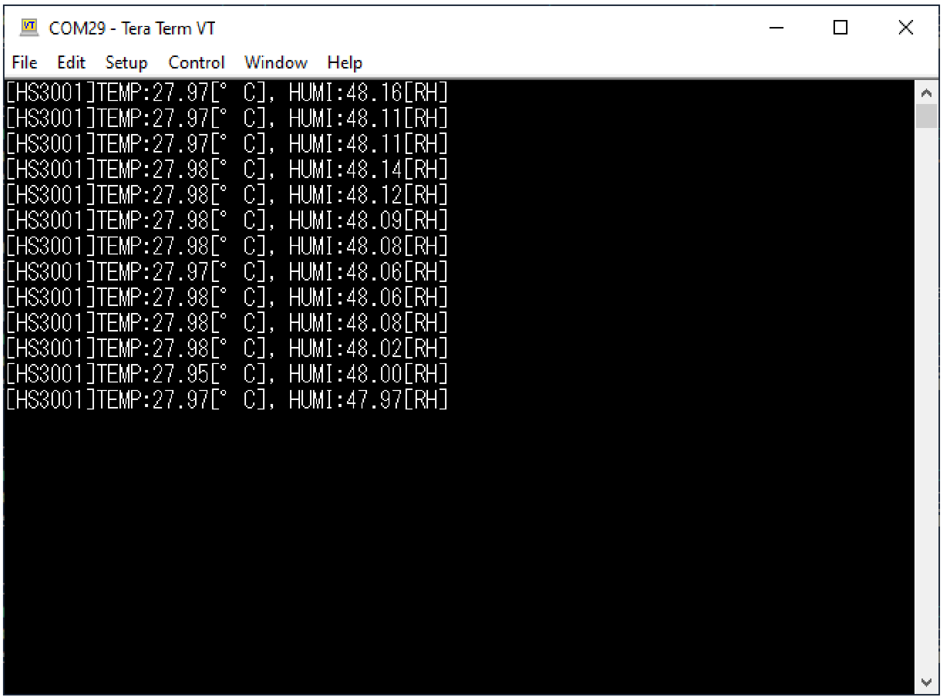


Figure 7-2 Log Screen of the RL78/G22 FPB

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

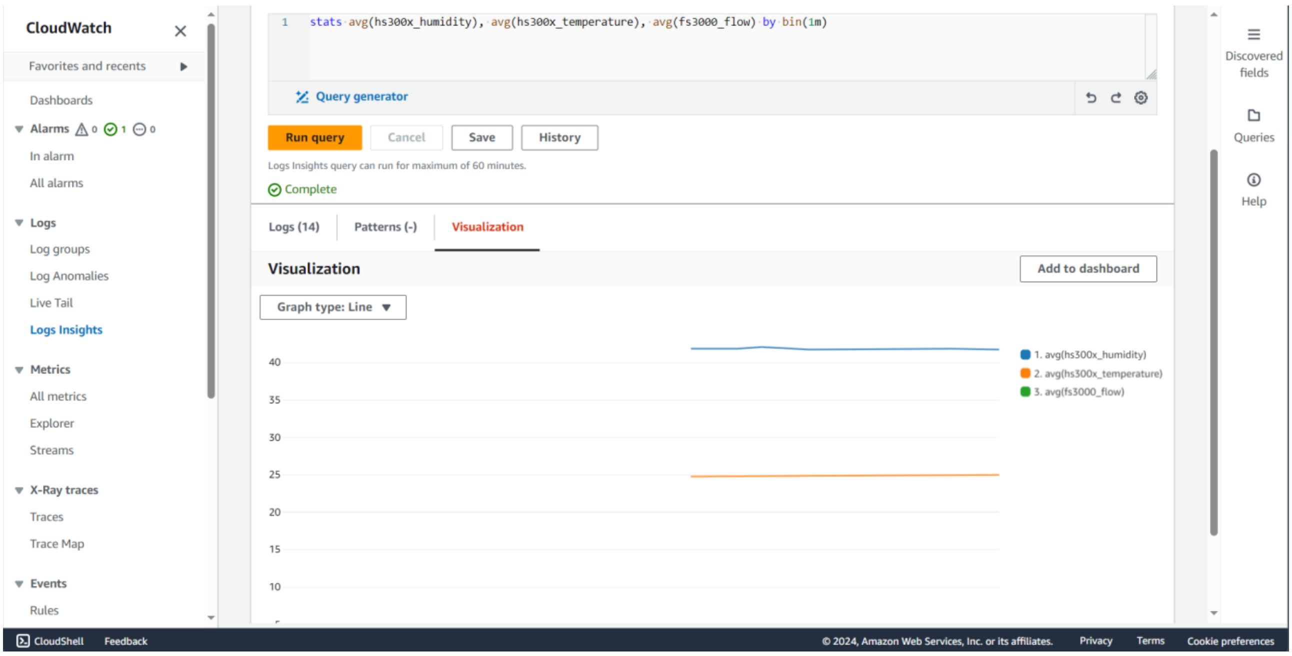


Figure 7-3 Graphical Display of Amazon CloudWatch before the Secondary OTA Update

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

RENESAS

7.2 Executing the OTA Update of the RL78/G22 FPB

7.2.1 Creating the Update Firmware

7.2.1.1 Partial Update Method

(1) Changing the source code of the rl78g22_fpb_2ndota_demo project

Change the definition of the “`USE_SENSOR_FS3000`” macro at line 53 of rl78g22_fpb_2ndota_demo/src/rl78g22_fpb_2ndota_demo.c from `0` to `1`.

You can also change the firmware version displayed in the log output by setting `FWUP_DEMO_VER_MAJOR`, `MINOR`, and `BUILD` to desired values.

```
50  Macro definitions
52  #define USE_SENSOR_HS3001        (1)
53  #define USE_SENSOR_FS3000        (0)
54
55  #define FWUP_DEMO_VER_MAJOR      (1)
56  #define FWUP_DEMO_VER_MINOR      (0)
57  #define FWUP_DEMO_VER_BUILD      (0)
```

(2) Creating the update firmware (MOT file format)

Build the rl78g22_fpb_2ndota_demo project and create a MOT file.

(3) Creating the update firmware (RSU file format)

Convert the created rl78g22_fpb_2ndota_demo MOT file into update firmware in the RSU format by using the Renesas Image Generator.

Run the following command in the r01an6935jj0100-rl78g22/RenesasImageGenerator folder to create the RSU-format update firmware “update_firm.rsu”.

```
> python .\image-gen.py -ip .\RL78_G22_ImageGenerator_PRM.csv -
iup ..\rl78g22_fpb_2ndota_demo\HardwareDebug\rl78g22_fpb_2ndota_demo.mot -o update_firm
```

7.2.1.2 Full Update Method

Create the update firmware by following a procedure similar to that in “7.2.1.1 Partial Update Method”. Replace the project to be used, that is, the rl78g22_fpb_2ndota_demo project, with the rl78g22_fpb_2ndota_demo_full project.

The command for creating an RSU file is as follows:

```
> python .\image-gen.py -ip .\RL78_G22_FullUpdate_ImageGenerator_PRM.csv -
iup ..\rl78g22_fpb_2ndota_demo_full\HardwareDebug\rl78g22_fpb_2ndota_demo_full.mot -o
update_firm
```

7.2.2 Creating an OTA Job in AWS

(1) Sign in to the AWS Management Console and select [Services] in the upper-left corner, then select [Internet of Things] → [IoT Core].

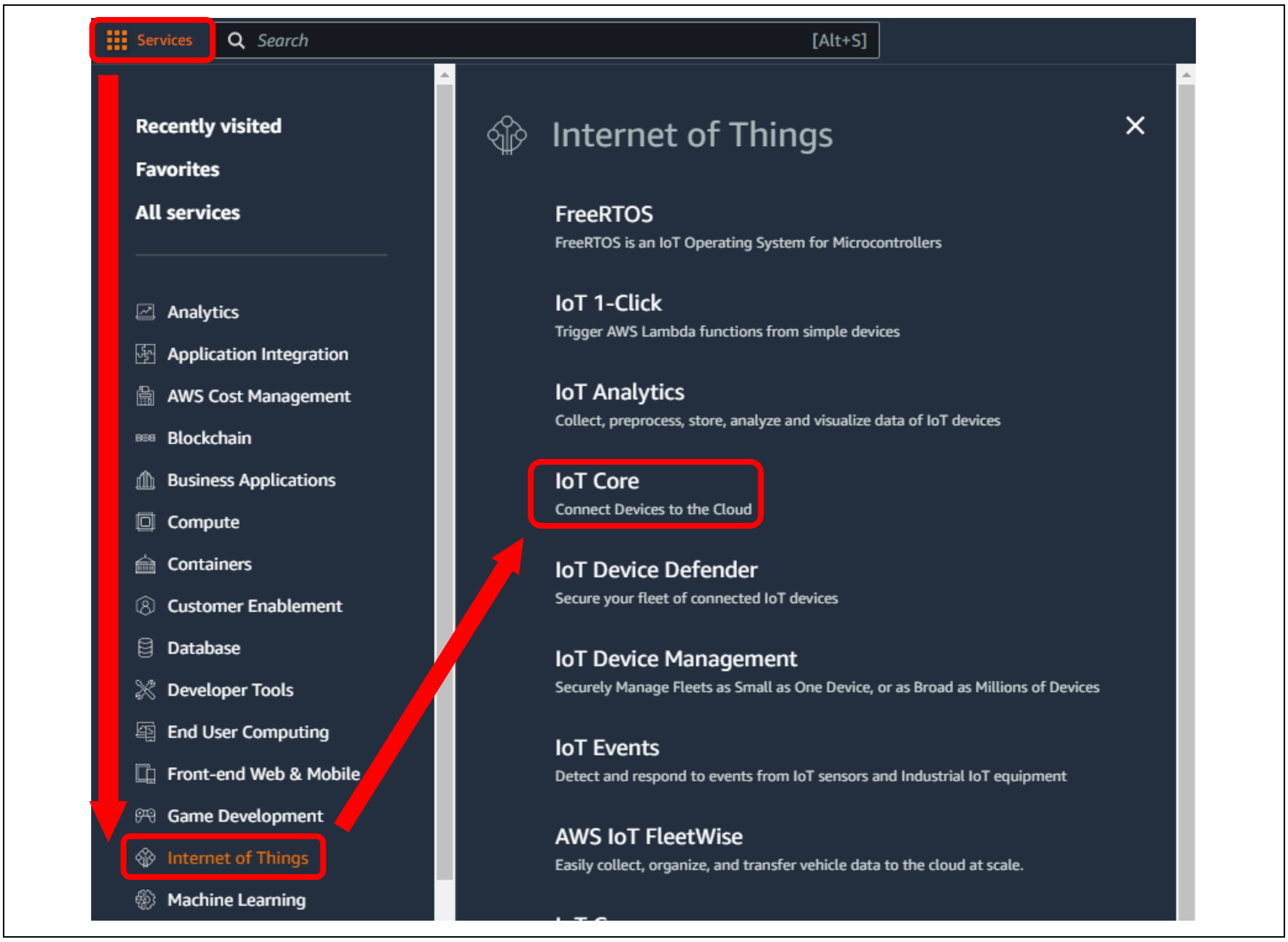


Figure 7-4 Window of Services of AWS

(2) Select [Remote action] → [Jobs] from the menu at left in AWS IoT Core and click on [Create job].

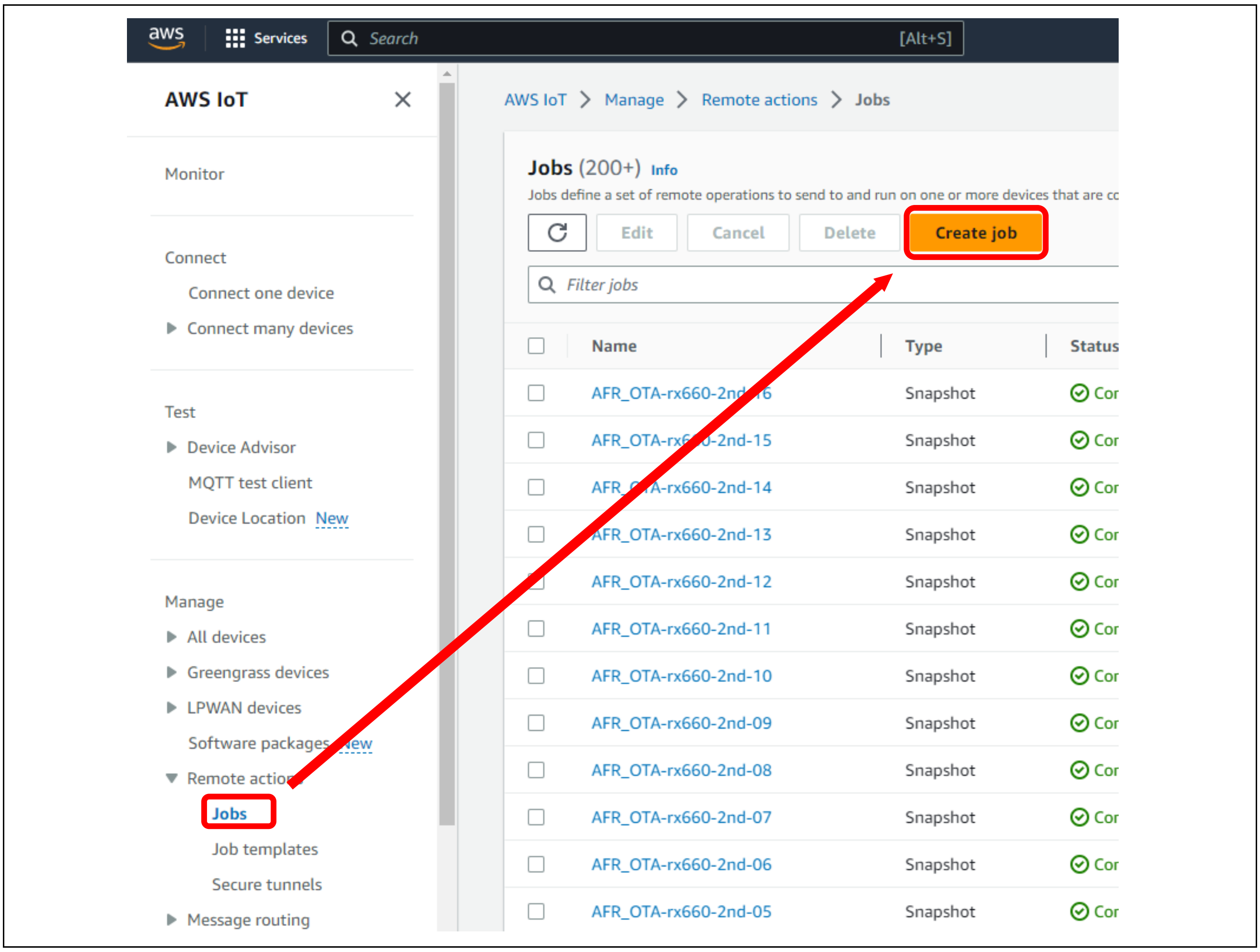


Figure 7-5 Window of AWS IoT Core

(3) On the [Create job] page, select "Create FreeRTOS OTA update job" and click on [Next].

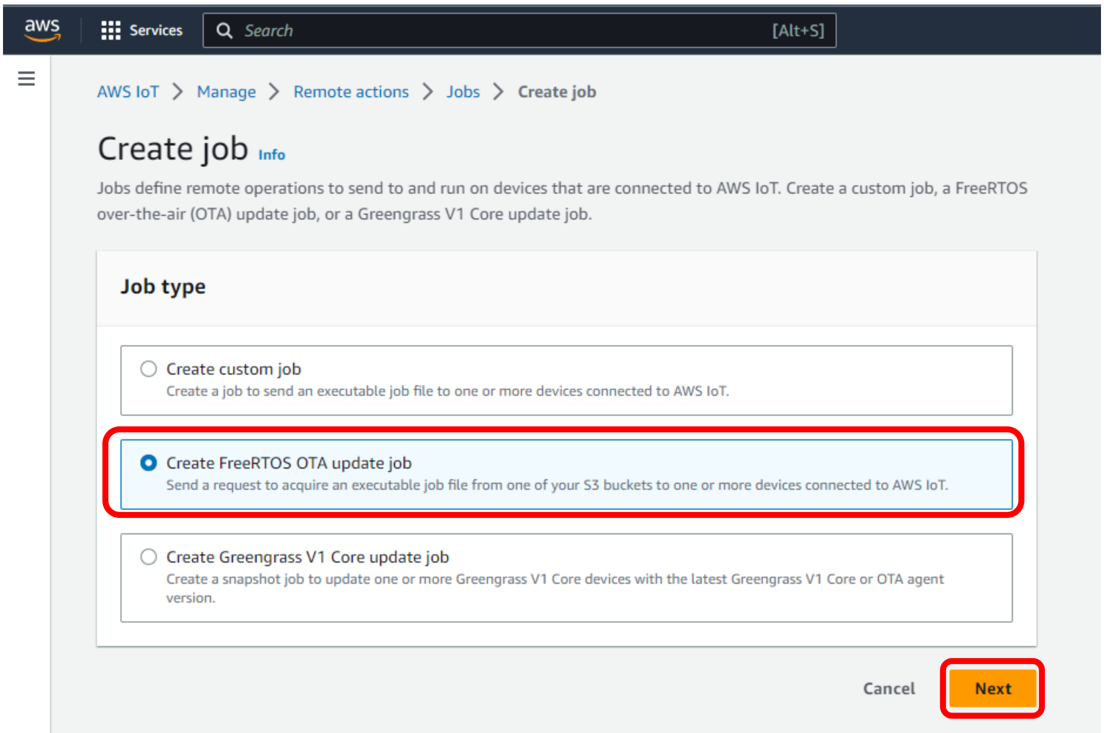


Figure 7-6 Page for Creating a Job

(4) On the [OTA job properties] page, enter a job name in [Job name] and click on [Next].

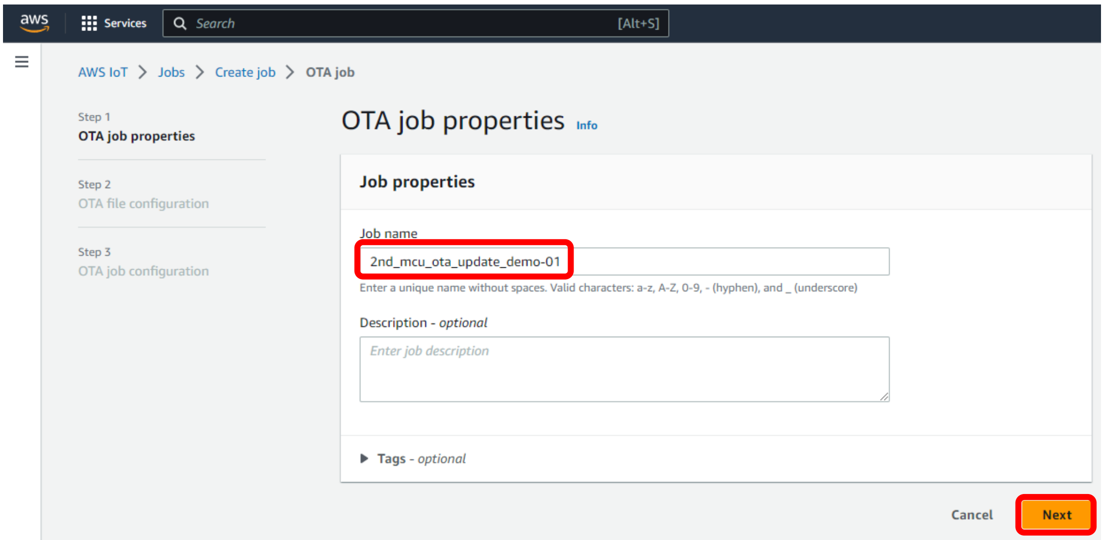


Figure 7-7 Page for Entering the Properties of the OTA Job

(5) Enter the various items indicated below on the [OTA file configuration] page.

1. In [Devices to update], select the IoT device name that was set in QE for OTA in 0.
2. In [Select the protocol for file transfer], select “MQTT”.
3. In [Sign and choose your file], select “Sign a new file for me”.
4. In [Code signing profile], select the code signing profile that was created in 6.3.1(3).

   Note: The code signing certificate profile specified here is not used for code signing verification of the firmware of the secondary MCU, so any profile can be specified. The code signing will be written in the RSU file of the update firmware when it is created by the Renesas Image Generator.
5. In [File], select “Upload a new file.”.
6. In [File to upload], click on [Choose file] and select the firmware (.rsu format) that was created in 7.2.1 for use in updating the RL78/G22 FPB.
7. In [S3 URL], click on [Browse S3] and select the Amazon S3 bucket that was set in 6.3.1(1).
8. In [Path name of file on device], enter a desired string of characters.
9. In [File type], enter “1” for the partial update method and “2” for the full update method.
10. In [Role], select the service role for the OTA update that was set in 6.3.1(2).

After entering the above, click on [Next].

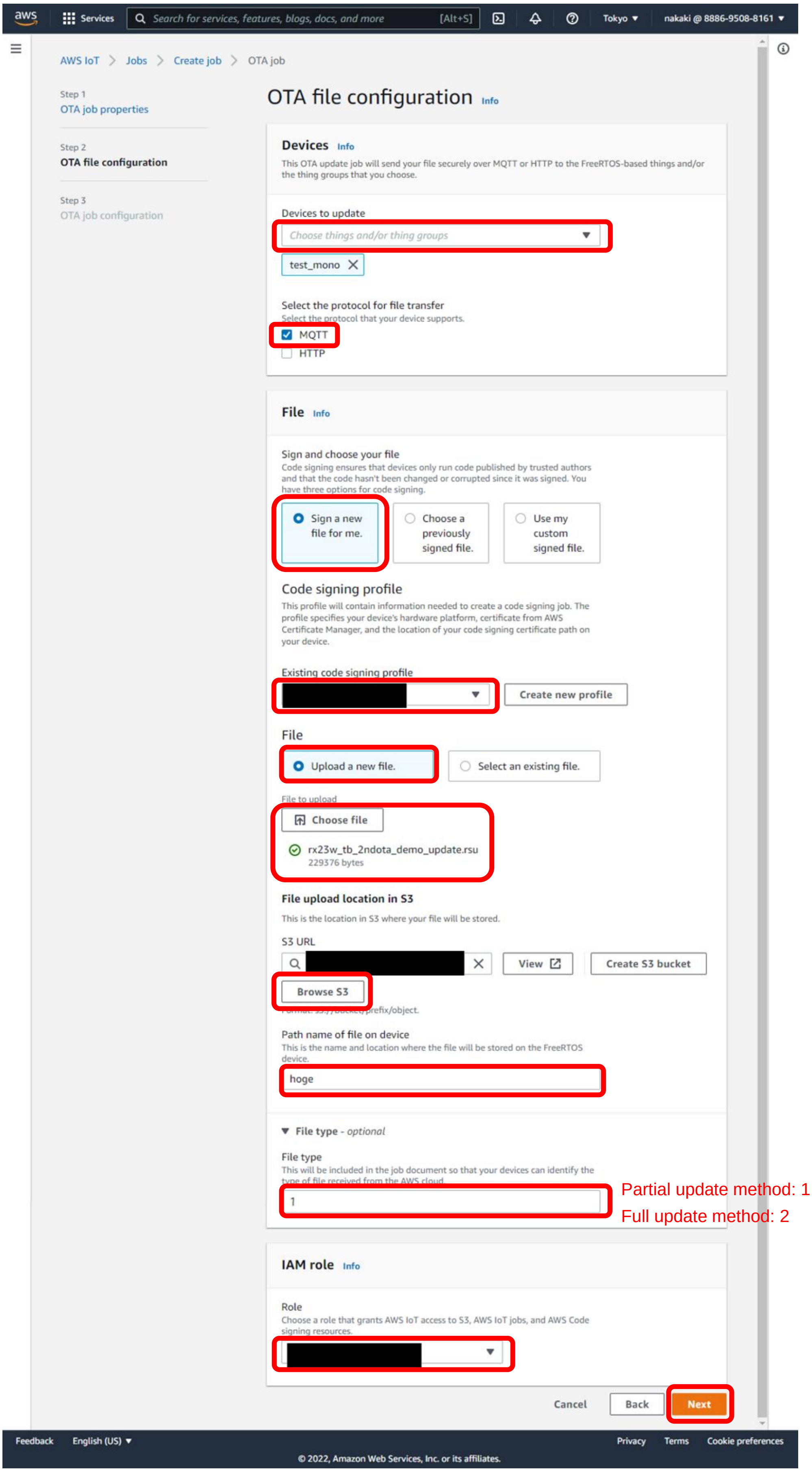


Figure 7-8 Page for Setting up the OTA File

(6) Just click on [Create job] on the [OTA job configuration] page as it is not necessary to make any changes.

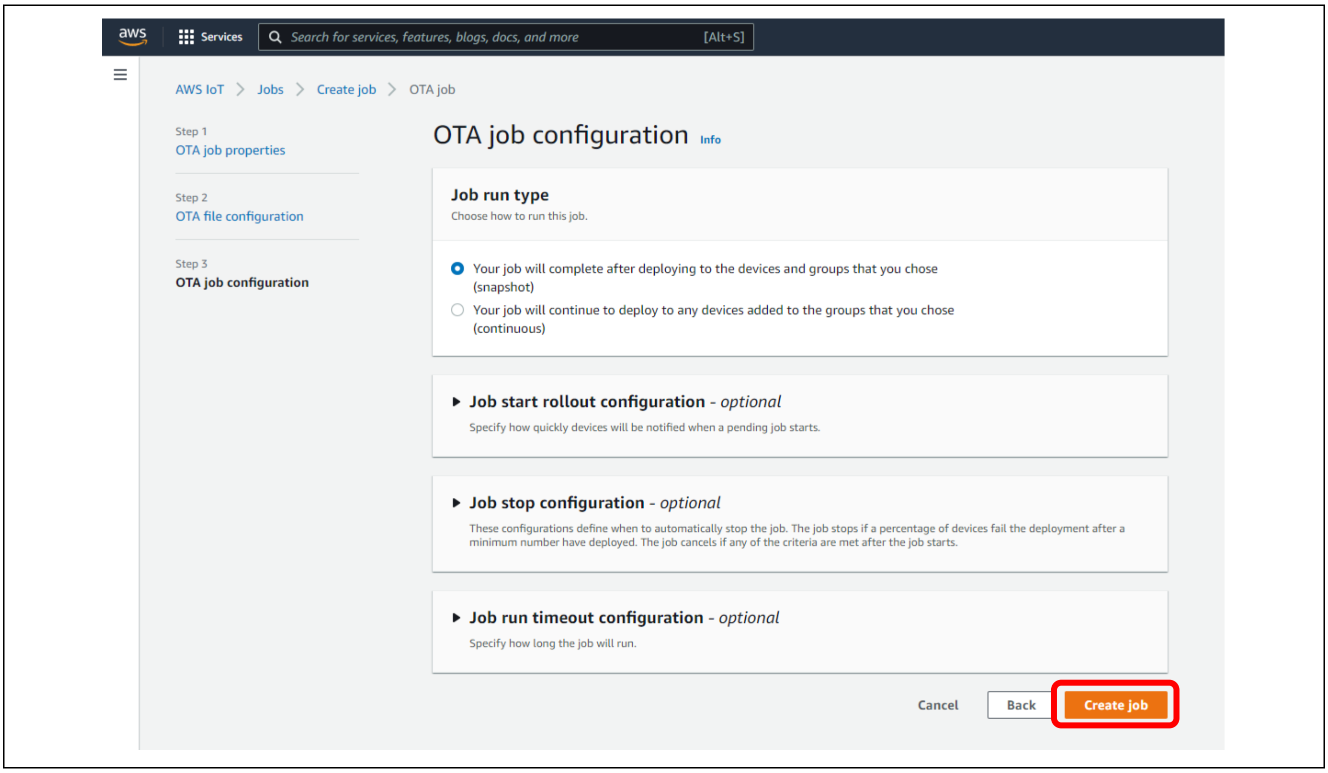


Figure 7-9 Page for Setting up the OTA Job

An OTA job for the secondary OTA update is created by following the above steps, and the OTA job is delivered to the specified 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 RL78/G22 FPB will output logs of the progress of the secondary OTA update.

7.3 Checking Operation after the OTA Update

Figure 7-10 shows the log screen of the CK-RX65N after the update.

You can see that data from the FS3000 sensor are displayed in addition to those from the HS3001 sensor.

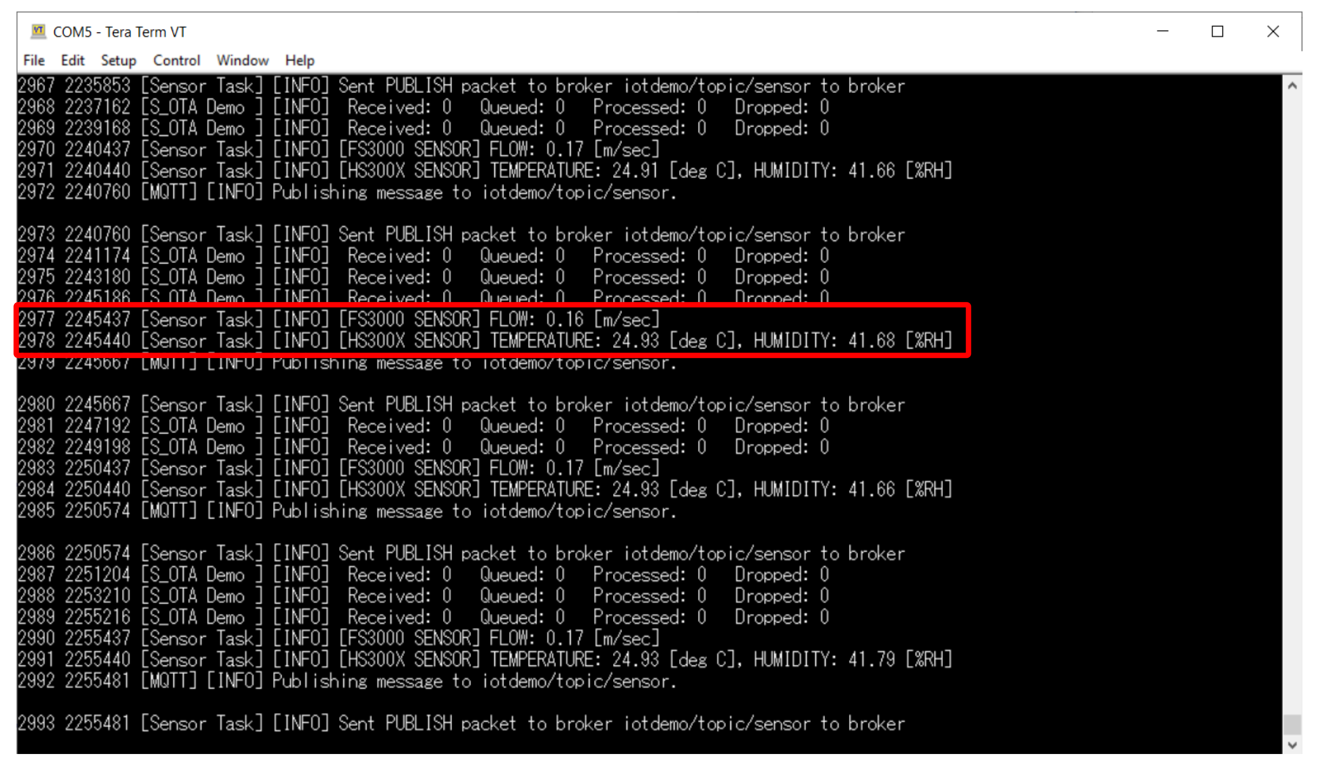


Figure 7-10 Log Screen of the CK-RX65N after the Firmware Update

Next, Figure 7-11 shows the log output for the RL78/G22 FPB after the update.

If the firmware update of the RL78/G22 FPB was successful, measured data are acquired from both the HS3001 and FS3000 sensors.

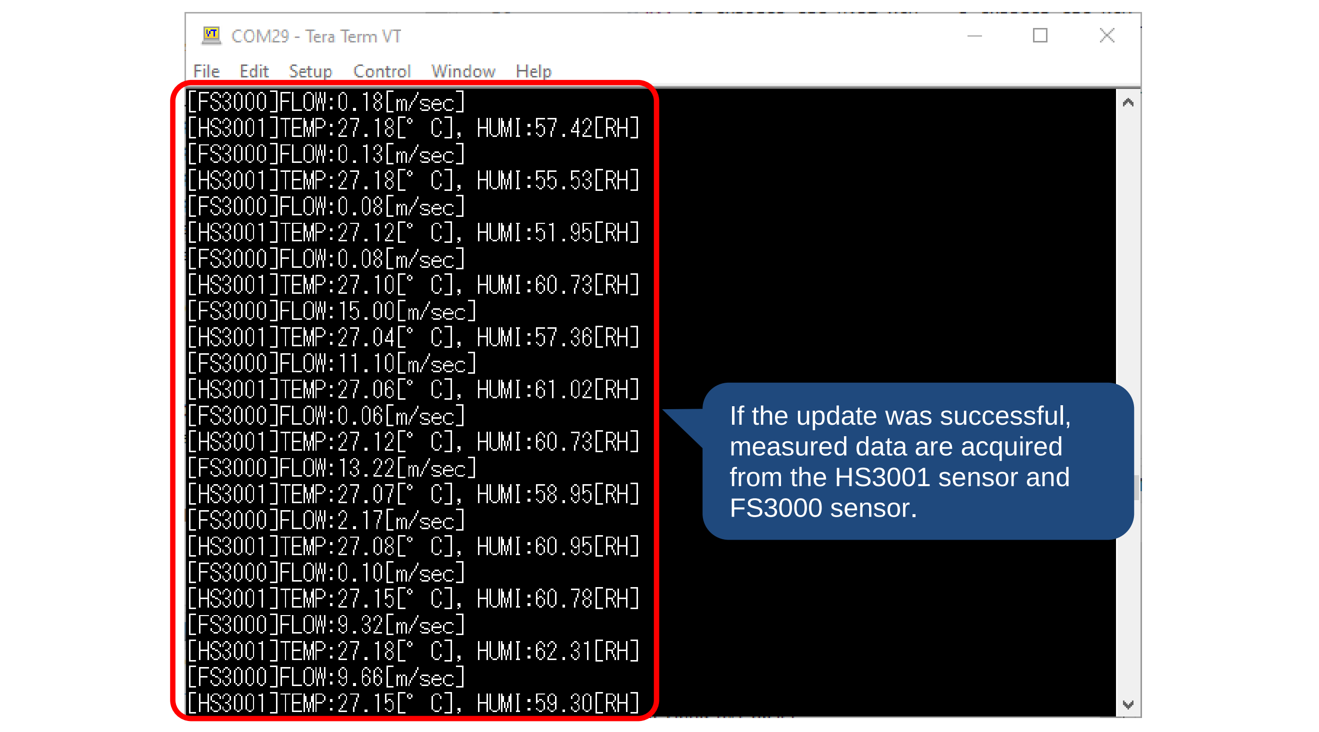


Figure 7-11 Log Screen of the RL78/G22 FPB after the Firmware Update

Finally, Figure 7-12 shows the display for Amazon CloudWatch. Confirm that the flow data acquired from the FS3000 sensor as well as the temperature and humidity data acquired from the HS3001 sensor are displayed as a graph.

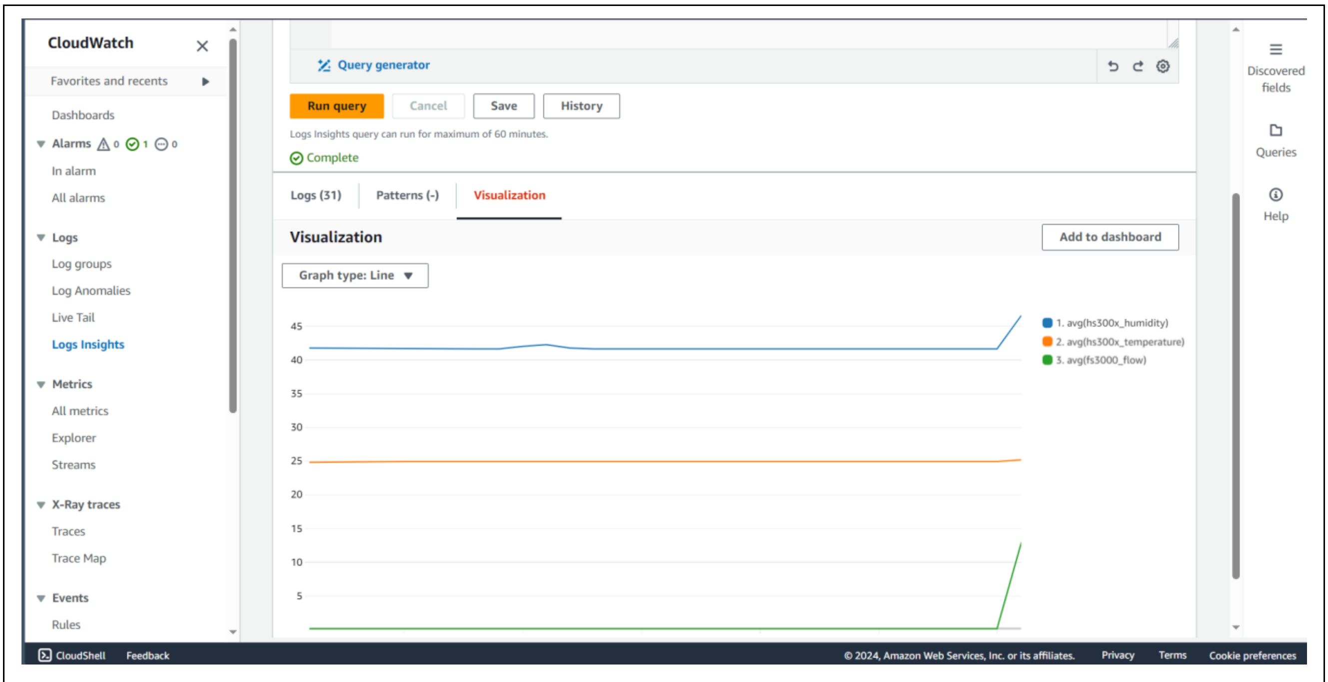


Figure 7-12 Graphical Display of Amazon CloudWatch after the Secondary OTA Update

Operations for the demonstration are completed at this point.

8. Precautions

8.1 License Information on the Open-Source Software in Use

The following open-source software is used.

- TinyCrypt Cryptographic Library
  — URL: https://01.org/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.

8.2 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.

8.3 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
1.00	Jun 30, 2023	—	First edition issued.
1.10	Mar 29, 2024	—	The full update method was added to the firmware update methods.
		—	The term of 1st MCU was changed to primary MCU and the term of 2nd MCU was changed to secondary MCU.
		5	The version of each software in the environment for confirming operation was updated.
		17	A description on the operating voltage was added.
		20	An image of the overall configuration for the demonstration was added.
		25-26	Descriptions on how to import projects were added.
		27-29	Descriptions on how to write firmware using the Renesas Flash Programmer were added.
		31	The application note to be referenced when preparing to use the AWS cloud was modified.
		32	A description on the procedure for confirming the AWS region was added.
		33-38	The method for displaying sensor data on the screen of AWS was modified from OpenSearch to CloudWatch.
		50	A description on the fees for using AWS was added to “8. Precautions”.

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/.