# RL78/G23

## Visualization of Sensor Information with Connecting Amazon Web Services in LTE Communication: RL78/G23-128p Fast Prototyping Board + FreeRTOS

### Introduction

This document describes how to connect to Amazon Web Services (AWS) using a Renesas MCU board combined with a cellular IoT module. The device uploads the Sensor data to the IoT Core of AWS.

### Related Documents

RL78/G23 User's Manual: Hardware (R01UH0896)

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

RL78/G23-128p Fast Prototyping Board User's Manual (R20UT4870)

Renesas Flash Driver RL78 Type 01 User's Manual (R20UT4830)

HS300x High-Performance Relative Humidity and Temperature Sensor (R36DS0010)

HS40xx High-Performance Relative Humidity and Temperature Sensor with Digital Output (R36DS0036)

RL78/G23 Getting Started Guide for Connecting Amazon Web Services in LTE Communication: RL78/G23-128p Fast Prototyping Board + FreeRTOS (R20AN0666)

### Notification: End-Of-Life (EOL) process on RYZ024A Cellular module

Renesas announces to discontinue the existing Sequans-sourced LTE module known as the RYZ024A part number and will no longer be shipping this product.

If you have one in a current design or in production, the Sequans part number, GM02S is pin for pin, form fit and function exact drop-in replacement from the RYZ024A. Below Cellular driver of RX family works the below alternate product combination.

- RYZ024A Cellular module control module: Sequans GM02S is the compatible module.

Regarding EOL notice of the RYZ024A, see the link at the product page.

## Contents

1. Overview ....4
1.1 Overview of Demo Projects ....4
1.2 Operation Confirmation Conditions ....5
1.3 Equipment List ....7
2. Description of Hardware ....8
2.1 System Configuration ....8
2.2 List of Pin Used ....8
3. Description of Software ....9
3.1 Demo Project ....9
3.2 List of Option Bytes Settings ....9
3.3 Folder Structure ....10
3.4 Code Size ....11
4. Description of the Demo Project ....12
5. Setup to Demo Projects ....12
5.1 Hardware Setup ....12
5.1.1 Overall Configuration ....12
5.1.2 Connecting Hardware ....12
5.2 Software Setup ....16
5.2.1 Terminal Software Settings ....16
5.2.2 Installing Flash Writer ....16
5.2.3 Adding SIM Card Information to the Demo Project ....16
5.2.4 Adding AWS IoT Connection Settings to the Demo Project ....16
5.2.5 Selecting the Sensor board to use ....16
5.3 Settings for OTA Update ....16
5.3.1 Installing Tools ....16
5.3.2 Generating Keys for Signature Generation and Verification ....16
5.4 Preparing the AWS Cloud ....17
5.4.1 Settings for visualizing sensor data ....17
5.4.1.1 Settings for Amazon CloudWatch ....17
5.4.2 Settings for OTA Update ....28
5.4.2.1 Creating Amazon S3 Buckets ....28
5.4.2.2 Creating an OTA Update Service Role ....28
5.4.2.3 Creating an OTA Update User Policy ....28
5.4.2.4 Allocating an OTA Update Policy to IAM User ....28
5.4.2.5 Granting Access Permissions to AWS IoT Code Signature ....28
5.5 Creating an Initial Image ....29
5.5.1 Creating a Bootloader ....29

5.5.1.1 Importing the Bootloader Project ..... 29
5.5.1.2 Adding the Firmware Verification Key to the Bootloader Project ..... 29
5.5.1.3 Building the Bootloader Project ..... 29
5.5.2 Creating an Initial Application ..... 30
5.5.2.1 Importing the Initial Application ..... 30
5.5.2.2 Adding the Firmware Verification Key to the Initial Application ..... 31
5.5.2.3 Building the Initial Application ..... 31
5.5.3 Creating an Initial Image by Using Renesas Image Generator ..... 32
5.6 Creating an Update Image ..... 33
5.6.1 Creating an Update Application ..... 33
5.6.1.1 Changing the Source Code of the Application ..... 33
5.6.1.2 Building the Update Application ..... 33
5.6.1.3 Renaming the MOT File of the Update Application ..... 34
5.6.2 Generating an Update Image by Using Renesas Image Generator ..... 34
5.7 Running the Demo Project ..... 35
5.7.1 Programming the Initial image (initial_image.mot) to Board ..... 35
5.7.2 Registering the Update Image (aws_ryz024a_sensor_rl78g23-fpb_093.rsu) with OTA Jobs ..... 38
5.8 Debugging the Initial Application ..... 46

6. Using Renesas Flash Programmer ..... 47

7. Debug Procedure ..... 47

8. Appendix ..... 47
8.1 Precautions on Porting Third-Party Libraries to RL78 ..... 47
8.1.1 Width of int Is 16 Bits ..... 47
8.1.2 Size Limitation of Section ..... 47
8.2 License Information for Open Source Software Used with Demo Projects ..... 47

9. Websites and Supports ..... 48

Revision History ..... 49

Notes:

AWS™ is a trademark of Amazon.com, Inc. or its affiliates. (https://aws.amazon.com/trademark-guidelines/)
FreeRTOS™ is a trademark of Amazon Web Services, Inc. (https://freertos.org/copyright.html)
GitHub® is a trademark of GitHub, Inc. (https://github.com/logos)

# 1. Overview

The sample program iot-reference-rl78 provides the reference of IoT solution using RL78 family, AWS, and FreeRTOS. You can easily try to run AWS IoT demos while it works with our various other products.

## 1.1 Overview of Demo Projects

The sample program contains the following demo projects. These demo projects realize the operation for connecting to the AWS clouds using the RL78/G23-128p Fast Prototyping Board of the Renesas MCU board and cellular IoT module.

**Table 1-1 List of demo projects**

| ItemName of Demo Project | Description |
|---|---|
| Demo project (PubSub) | Perform simple data upload via MQTT communication. |
| Demo project (OTA) | Perform firmware update via OTA. |
| Demo project (Sensor) | Perform combination with sensor data acquisition, MQTT communication, sensor data visualization, and OTA updates. |

This application note describes about the demo project (sensor), which is hereinafter referred to as demo project. About other demo projects, refer to the Getting Started Guide (R20AN0666).

The initial firmware of demo project operates to upload temperature data obtained from the temperature/humidity sensor board (hereinafter referred to as sensor board) to the AWS cloud via MQTT communication. Uploaded sensor data can be visualized using the Amazon CloudWatch service (hereinafter referred to as CloudWatch) provided by AWS. Next, update the initial firmware to the updated firmware using OTA update, and start the updated firmware. The updated firmware is added the feature to obtain and upload humidity data in addition to temperature data.

For details about summary of demo projects, refer to the following chapters.

- 2 Description of Hardware
- 3 Description of Software

For details about how to run the demo projects, refer to the following chapters.

- 5 Setup to Demo Projects

## 1.2 Operation Confirmation Conditions

Demo project operations have been confirmed in the following conditions.

**Table 1-2 Operation Confirmation Conditions (RL78/G23)**

| Item | Description |
|---|---|
| MCU used | RL78/G23 (R7F100GSN CF 768KB) |
| Board used | RL78/G23-128p Fast Prototyping Board (RTK7RLG230CSN000BJ) |
| Operating frequency | High-speed on-chip oscillator clock: 32 MHz |
| Operating voltage | 3.3 V |
| IDE (Integrated Development Environment) | Renesas Electronics<br>e$^2$ studio 2024-04 |
| C compiler | Renesas Electronics<br>CC-RL V1.12.01 |
| Firmware programming tool | Renesas Flash Programmer V3.14.00 |
| Smart Configurator (SC) | Renesas Smart Configurator for RL78 24.4.0.v20240402-0340 |
| Board support package (BSP) | v1.60 (r_bsp) |
| Flash library (RFD) | Renesas Flash Driver (RFD) RL78 Type01 for RL78/G2x V1.20<br>Note: Code Flash Libraries (Flash Self Programming Libraries) -><br>Renesas Flash Driver RL78 Type 01 Package V1.20 for RL78/G2x |
| Firmware update module (FWUP) | RL78/G22,RL78/G23,RL78/G24 Firmware Update Module v2.01<br>(Application note, sample code) |
| Utility tool to generate firmware images | Renesas Image Generator V3.03<br>Note: Included in the firmware update module (FWUP) |
| Python | Python 3.10.1 |
| OpenSSL | OpenSSL 3.1.4 |

**Table 1-3 Operation Confirmation Conditions (Sensor board)**

| Item | Description |
|---|---|
| Sensor board (HS3001) | US082-HS3001EVZ Board |
| Sensor board (HS4001) | QCIOT-HS4001POCZ |

**Table 1-4 Operation Confirmation Conditions (Others, such as OSS Library)**

| Item | Description |
|---|---|
| iot-reference-rl78 | v202210.01-LTS-rl78-1.0.0 (Based FreeRTOS 202210.01-LTS)<br>https://github.com/renesas/iot-reference-rl78/tree/v202210.01-LTS-rl78-1.0.0 |
| FreeRTOS Cellular Interface | 1.3.0<br>https://github.com/FreeRTOS/FreeRTOS-Cellular-Interface |
| FreeRTOS Kernel | 10.5.1<br>https://github.com/FreeRTOS/FreeRTOS-Kernel |
| backoffAlgorithm | 1.3.0<br>https://github.com/FreeRTOS/backoffAlgorithm |
| coreJSON | 3.2.0<br>https://github.com/FreeRTOS/coreJSON |
| coreMQTT Client | 2.1.1<br>https://github.com/FreeRTOS/coreMQTT |
| coreMQTT Agent | 1.2.0<br>https://github.com/FreeRTOS/coreMQTT-Agent |
| AWS IoT Over-the-air Update | 3.4.0<br>https://github.com/aws/ota-for-aws-iot-embedded-sdk |
| tinycbor | 0.5.2<br>https://github.com/intel/tinycbor |
| FreeRTOS-Plus network_transport | No version<br>https://www.freertos.org/network-interface.html |
| Logging Interface | 1.1.3<br>https://github.com/aws/amazon-freertos/tree/main/libraries/logging |
| TinyCrypt Cryptographic Library | 0.2.8<br>https://github.com/intel/tinycrypt |

## 1.3 Equipment List

The following lists the equipment required for the demo projects.

**Table 1-5 Equipment List**

| Item | Description |
|---|---|
| MCU board | RL78/G23-128p Fast Prototyping Board<br>RTK7RLG230CSN000BJ - RL78/G23-128p Fast Prototyping Board |
| Cellular IoT module | PMOD Expansion Board for RYZ024A (referred to as RYZ024A hereafter)<br>RTKYZ024A0B00000BE - PMOD Expansion Board for RYZ024A<br>F/W ver: LR8.2 |
| Sensor board | US082-HS3001EVZ Board (referred to as HS300x hereafter) or<br>QCIOT-HS4001POCZ (referred to as HS400x hereafter) |
| SIM card | LTE communication must be possible.<br>Example: SIM card by Truphone bundled with RTKYZ024A0B00000BE (Note)<br>DHA-SIM-132 by Nippon SIM |
| USB-UART conversion board | Pmod USBUART<br>https://reference.digilentinc.com/reference/pmod/pmodusbuart/start |
| Micro USB Type-B cable x 3 | • Used to connect the USB-UART conversion board to the PC<br>• Used to connect the MCU board to the PC<br>• Used to supply power to RYZ024A |
| Jumper wire x 3 | Used to connect the USB-UART conversion board to the MCU board |
| Jumper pin x 3 | Pins J15, J16, and J19 are used to select the MCU board power supply. |

Note:

When using a SIM card by Truphone bundled with PMOD Expansion Board for RYZ024A (RTKYZ024A0B00000BE), you must activate your SIM card by the following document: RA6M5 Group RYZ024A PMOD LTE Connectivity with RA6M5 MCU Quick Start Guide (R21QS0007).

Overall figure of equipment connections

- demo project: Figure 5-1 Overall Hardware Configuration of the Demo Project

Precaution about equipment for debugging

The demo projects use the COM port for debugging, but debugging with the emulator is also possible. When using the emulator, you need to mount the connector for connecting the emulator and change the circuit. For details, refer to section 7.2.1 Setting Jumper Pins, Mounting the Connector, and Cutting Patterns in the Getting Started Guide.

**Table 1-6 Debug Equipment**

| Item | Description |
|---|---|
| Emulator | E2 emulator Lite<br>https://www.renesas.com/software-tool/e2-emulator-lite-rte0t0002lkce00000r |

# 2. Description of Hardware

## 2.1 System Configuration

The following shows the system configuration of the demo project.

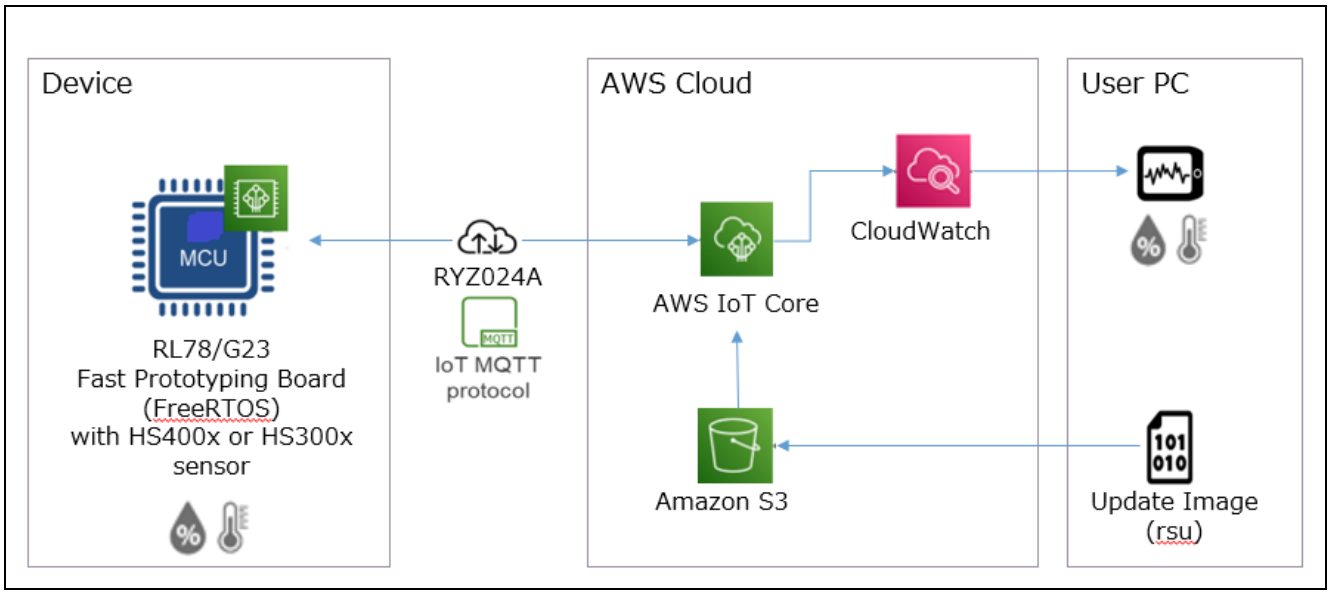


**Figure 2-1 System Configuration of Demo Project**

## 2.2 List of Pin Used

The following lists and describes the pins used with the demo project.

**Table 2-1 Pins Used with Demo Project and Their Functions**

| Pin Name | I/O | Description |
|---|---|---|
| P143/RxD3 | Input | UART communication (reception) with RYZ024A |
| P144/TxD3 | Output | UART communication (transmission) with RYZ024A |
| P00 | Output | Reset to RYZ024A |
| P142 | Output | UART communication (RTS) with RYZ024A |
| P14/RxD2 | Input | Terminal input |
| P13/TxD2 | Output | Terminal output |
| P50 | Output | LED1 |
| P51 | Output | LED2 |
| P62/SCLA1 | Input/Output | I2C communication (clock) with sensor board |
| P63/SDAA1 | Input/Output | I2C communication (data) with sensor board |

# 3. Description of Software

## 3.1 Demo Project

Refer to section 3.2.1 Demo Project Structure in the Getting Started Guide.

## 3.2 List of Option Bytes Settings

The following shows the option bytes settings.

**Table 3-1 Option Bytes Settings**

| Address | Settings | Description |
|---|---|---|
| 000C0H/040C0H | 11101111B | Stops the watchdog timer operation.<br>(Stops counting after the release from the reset state.) |
| 000C1H/040C1H | 00111010B | LVD0 off (using an external reset input from the RESET pin) |
| 000C2H/040C2H | 11101000B | HS (high-speed main) mode and<br>High-speed on-chip oscillator clock (fIH): 32 MHz |
| 000C3H/040C3H | 10000100B | Enables on-chip debugging. |

## 3.3 Folder Structure

The following shows the folder structure of the sample program.

**Table 3-2 Folder Structure of the Sample Program**

| Folder Name | Description |
|---|---|
| iot-reference-rl78 | The sample program is described in this Getting Started Guide. |
| ├─Common | |
| │ ├─FreeRTOS_common | |
| │ └─ports | |
| │ └─ota_pal | |
| ├─Configuration | |
| │ └─rl78g23-fpb | |
| │ └─sensor | Sensor demo configurations. |
| ├─Demos | |
| │ ├─common | |
| │ ├─include | |
| │ ├─mqtt_agent | |
| │ ├─OtaOverMqtt | OTA demo source codes. |
| │ ├─Sensor | Sensor demo source codes. |
| │ └─SimplePubSub | PubSub demo source codes. |
| ├─IDT_config | |
| ├─Middleware | |
| │ ├─3rdparty | |
| │ ├─Application-Protocols | |
| │ │ └─network_transport | |
| │ ├─AWS | |
| │ │ └─ota-for-aws-iot-embedded-sdk | |
| │ ├─FreeRTOS | FreeRTOS Kernel and libraries. |
| │ │ ├─backoffAlgorithm | |
| │ │ ├─coreJSON | |
| │ │ ├─coreMQTT | |
| │ │ ├─coreMQTT-Agent | |
| │ │ ├─FreeRTOS-Cellular-Interface | |
| │ │ └─FreeRTOS-Kernel | |
| │ └─logging | |
| ├─Projects | |
| │ └─rl78g23-fpb | |
| │ ├─application_code | |
| │ ├─flash_proj | |
| │ ├─helper | |
| │ ├─modules | |
| │ ├─projects | Import below folders to IDE. |
| │ │ ├─aws_ryz024a_rl78g23-fpb | PubSub demo and OTA demo. Select by Build Configurations. |
| │ │ ├─aws_ryz024a_sensor_rl78g23-fpb | Sensor demo (including OTA demo). |
| │ │ ├─boot_loader | Boot loader for OTA demo. |
| │ │ └─test_aws_cellular_ryz024a | |
| │ └─rtos_skelton | |
| ├─Test | |
| └─Tools | |

## 3.4 Code Size

The following table shows the ROM and RAM size of demo projects confirmed in the following conditions.

- CC-RL
  — Compile options:
- - Odefault: Optimization that is effective for both the object size and execution speed.
  — Link options:
- - optimize=symbol_delete: Deleting variables or functions that have not been referenced even once.

**Table 3-3 ROM and RAM Size of Demo Projects**

| Demo Project Name | ROM (byte) | RAM (byte) |
|---|---|---|
| aws_ryz024a_sensor_rl78g23-fpb | 233229 | 43204 |
| boot_loader | 22147 | 1348 |

# 4. Description of the Demo Project

The behavior of the demo project is shown below.

(1) The initial firmware uploads temperature data and temperature warning (False if below 25 degrees, True if above 25 degrees) from the sensor board to the AWS cloud.

(2) Download the update firmware via OTA update and update the firmware.

(3) After a firmware update, in addition to temperature data and temperature warning, humidity data is also uploaded to the AWS cloud. You can check the uploaded temperature data, temperature warning, and humidity data in logs and CloudWatch.

(4) Complete the upload to the AWS Cloud if any of the following conditions are true.

— When the number of uploads reaches 2000
— If sensor data acquisition fails

# 5. Setup to Demo Projects

The following describes the setup procedure applicable to demo project.

## 5.1 Hardware Setup

### 5.1.1 Overall Configuration

First, the following shows the overall configuration of hardware that makes up the demo project.

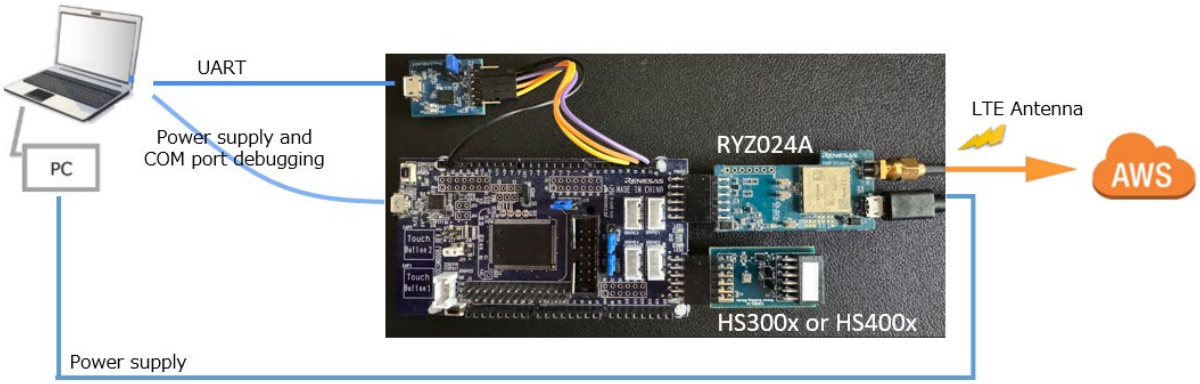


**Figure 5-1 Overall Hardware Configuration of the Demo Project**

### 5.1.2 Connecting Hardware

The following describes how to connect hardware.

(1) Insert the activated SIM card into RYZ024A.

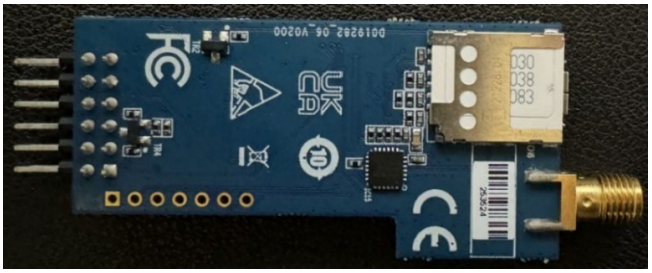

**Figure 5-2 Inserting Activated SIM Card into RYZ024A**

(2) Connect the antenna and power supply USB cable to RYZ024A.

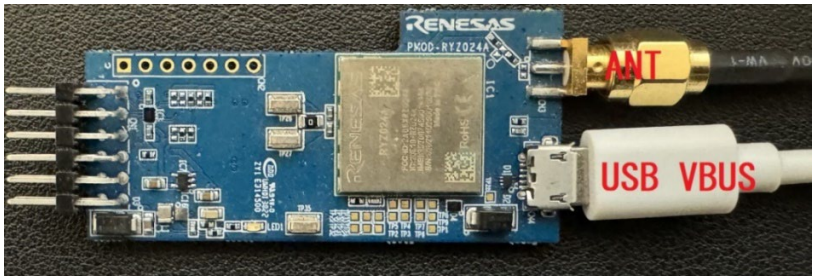


**Figure 5-3 Connecting Antenna and Power Supply USB Cable to RYZ024A**

(3) Connect RYZ024A to PMOD1 of the MCU board.

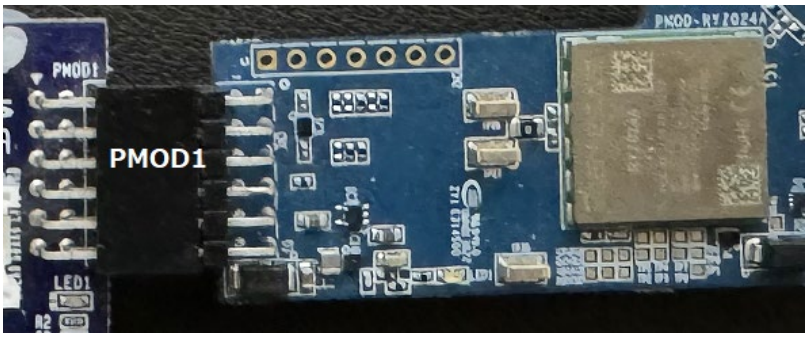


**Figure 5-4 Connecting RYZ024A to PMOD1 of the MCU Board**

(4) Connect Sensor Board to PMOD2 of the MCU board.

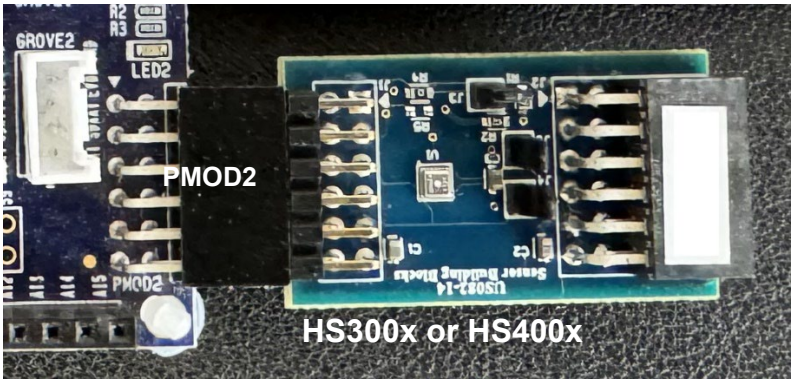


**Figure 5-5 Connecting Sensor Board to PMOD2 of the MCU Board**

(5) Connect the USB-UART conversion board to the MCU board.

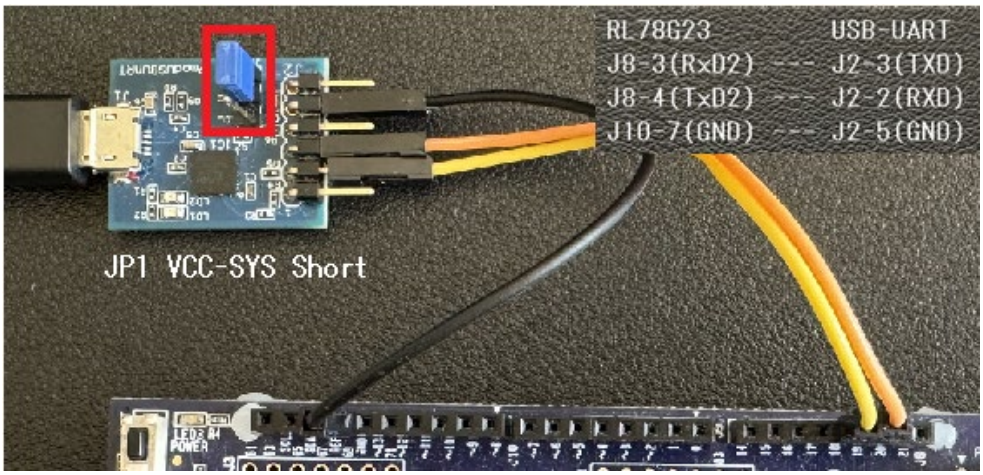


**Figure 5-6 Connecting the USB-UART Conversion Board to the MCU Board**

(6) On the MCU board, set the power supply selection header to J20 2-3 Short to select 3.3 V power supply.

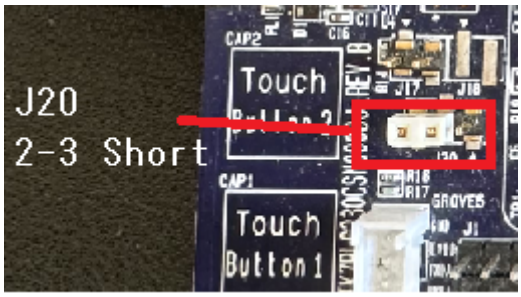


**Figure 5-7 Setting MCU Board Power Supply to 3.3 V**

(7) If you changed circuit to mount emulator connector on the MCU board, configure the COM port debugging that uses a USB-to-serial converter.
If you don't change circuit, you don't need this process.

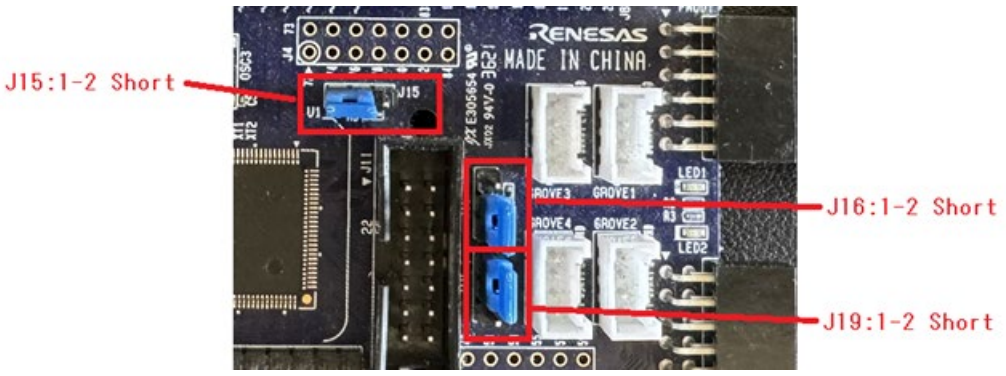


**Figure 5-8 Settings for Using COM Port Debugging (Top Side)**

(8) Connect the USB cable to supply power to the MCU board.

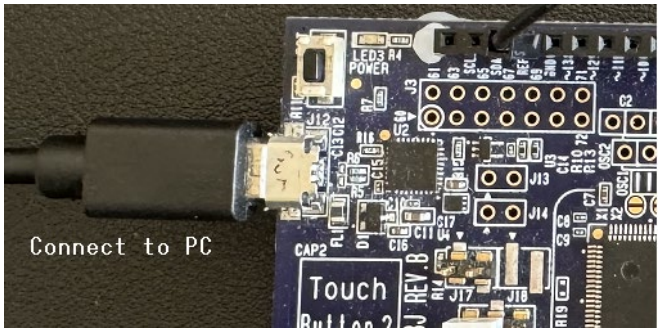


**Figure 5-9 Supplying Power to the MCU Board**

(9) Confirm the COM port number by using such as the device manager on Windows PC.
The COM port number will be used for programming and debugging firmware.

(10) Remove the USB cable to temporarily stop power supply to the MCU board.

## 5.2 Software Setup

### 5.2.1 Terminal Software Settings

Terminal software (example: Tera Term) is required to output demo project logs. The following shows the serial port settings.

**Table 5-1 Serial Port Settings**

| Item | Description |
|---|---|
| Baud rate | 115200 bps |
| Data | 8 bits |
| Parity | None |
| Stop bit | 1 bit |
| Flow control | None |

### 5.2.2 Installing Flash Writer

A flash writer is used for programming initial images.

Renesas Flash Programmer (Programming GUI)

### 5.2.3 Adding SIM Card Information to the Demo Project

Add the SIM card information after importing the project.

Refer to section 4.2.3 Adding SIM Card Information to the Demo Project in the Getting Started Guide.

### 5.2.4 Adding AWS IoT Connection Settings to the Demo Project

Add the AWS IoT connection settings after importing the project.

Refer to section 4.2.4 Adding AWS IoT Connection Settings to the Demo Project in the Getting Started Guide.

### 5.2.5 Selecting the Sensor board to use

Change the definition of the "SENSOR_CFG_TYPE" macro according to the sensor board you are using.

```
iot-reference-rl78\Projects\rl78g23-fpb\projects\aws_ryz024a_sensor_rl78g23-
fpb\src\application_code\sensor.h
For HS400x
#define SENSOR_CFG_TYPE (SENSOR_TYPE_HS400X)

For HS300x
#define SENSOR_CFG_TYPE (SENSOR_TYPE_HS300X)
```

## 5.3 Settings for OTA Update

### 5.3.1 Installing Tools

Refer to section 6.1.1 Installing Tools in the Getting Started Guide.

### 5.3.2 Generating Keys for Signature Generation and Verification

Refer to section 6.1.2 Generating Keys for Signature Generation and Verification in the Getting Started Guide.

## 5.4 Preparing the AWS Cloud

### 5.4.1 Settings for visualizing sensor data

To visualize the data received from the sensor board in graph format, follow these steps to configure Amazon CloudWatch and AWS IoT Core. First, log in to the AWS Management Console.

AWS Management Console | AWS (amazon.com)

Check the region displayed at the top right of the AWS Management Console screen and select the same region as the one you set when logging in.

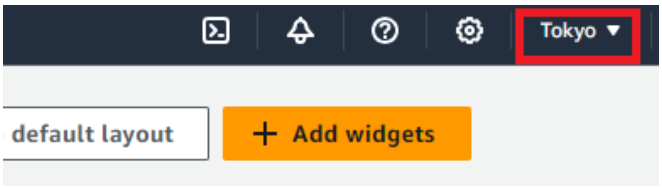


**Figure 5-10 Check the region**

#### 5.4.1.1 Settings for Amazon CloudWatch

**(1) Creating Rule on AWS IoT**

- AWS IoT > Message routing > Rules > Click “Create rule”

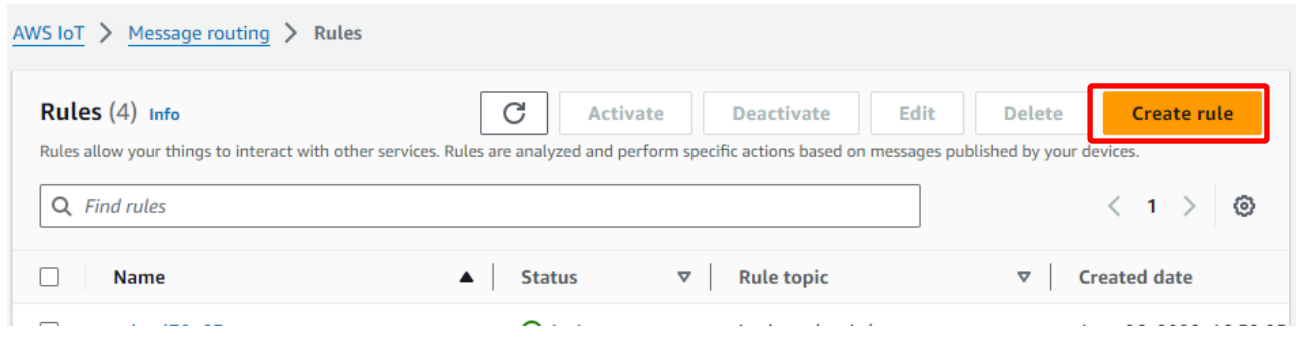


**Figure 5-11 Creating Rule on AWS IoT**

### (2) Specify rule properties

- Enter the Rule name and click “Next”

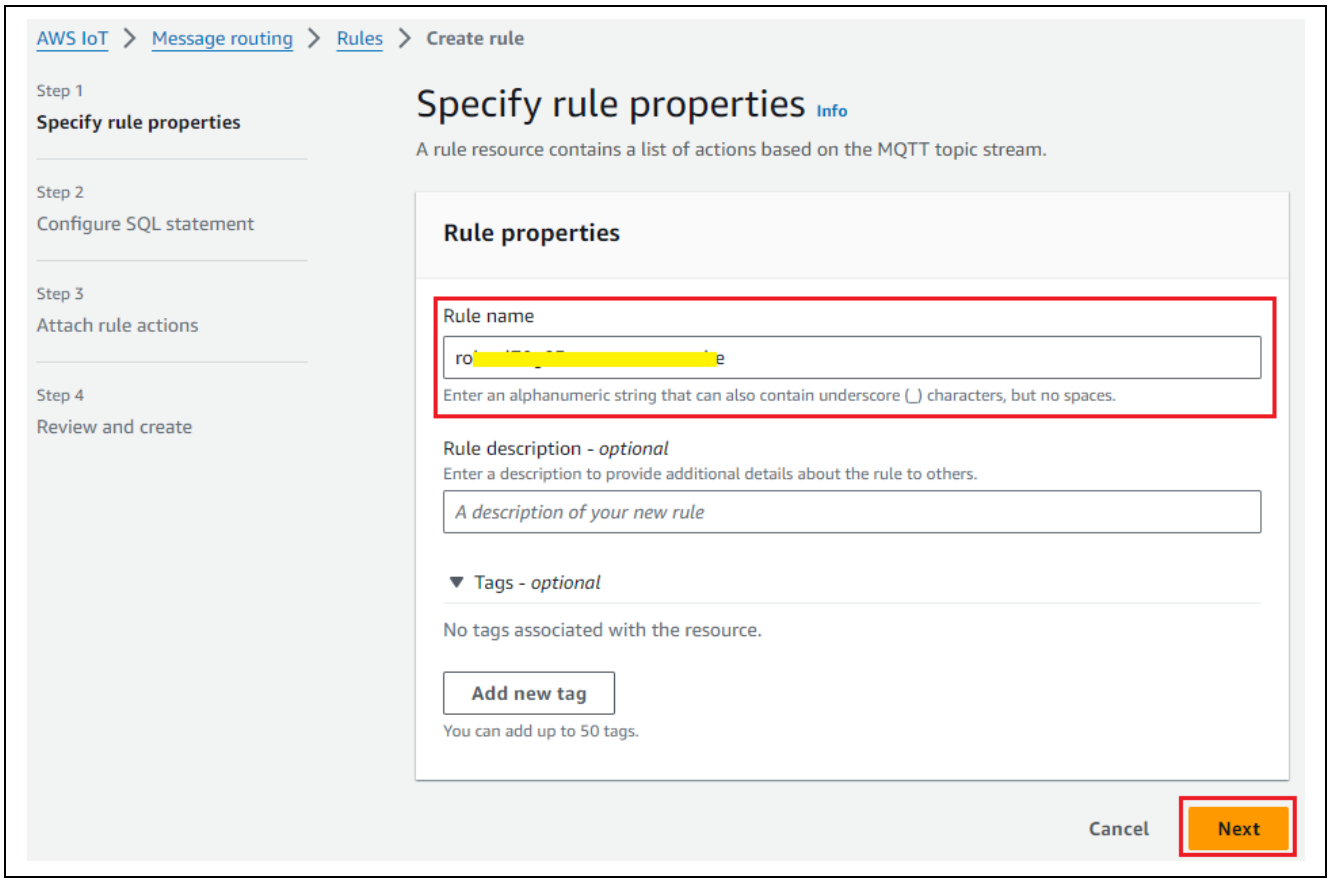


**Figure 5-12 Enter the Rule name and click “Next”**

### (3) Configure SQL statement

- Enter the following in the SQL statement and click “Next”

```
SELECT *, timestamp() as timestamp FROM '[thing-name]_tempdata'
(Need to start a new line)
```

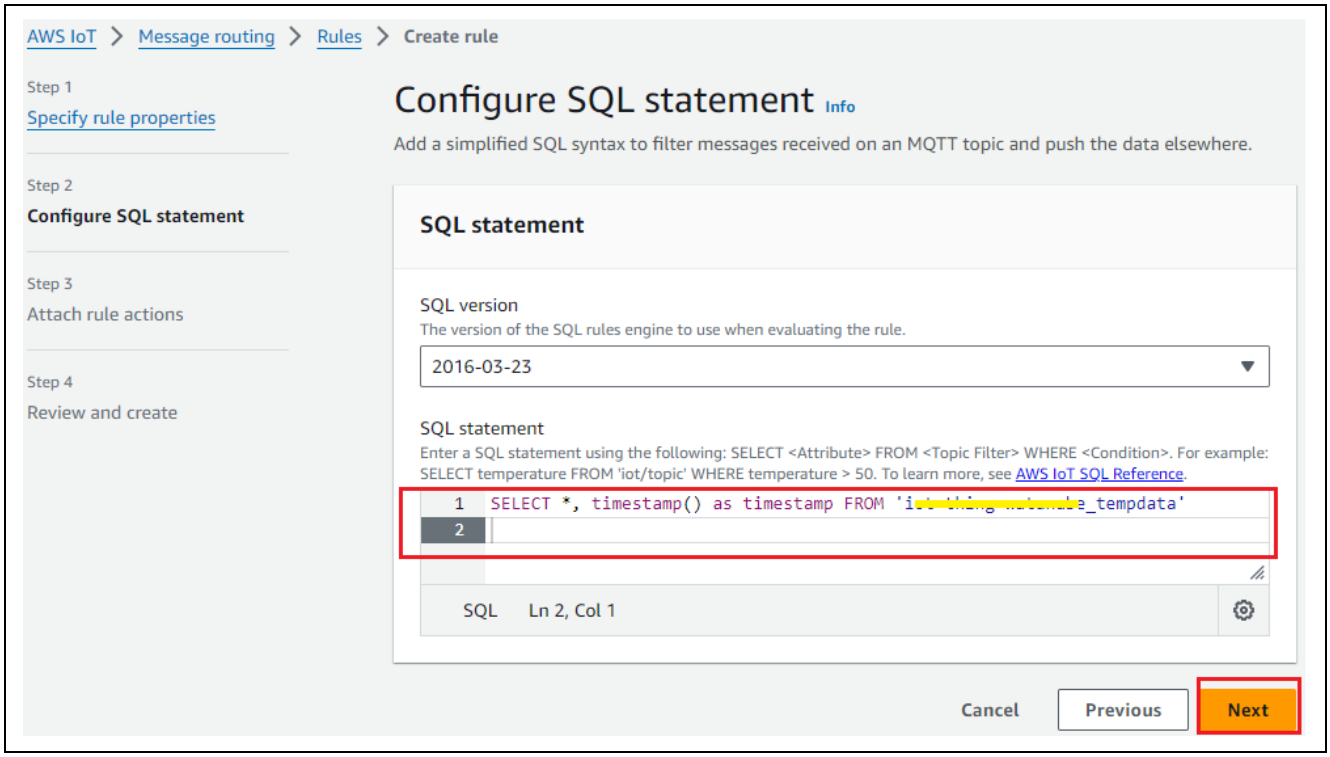


**Figure 5-13 Enter the following in the SQL statement and click “Next”**

### (4) Attach rule actions

- Select “CloudWatch logs” for Action1 and click “Create CloudWatch Log group”

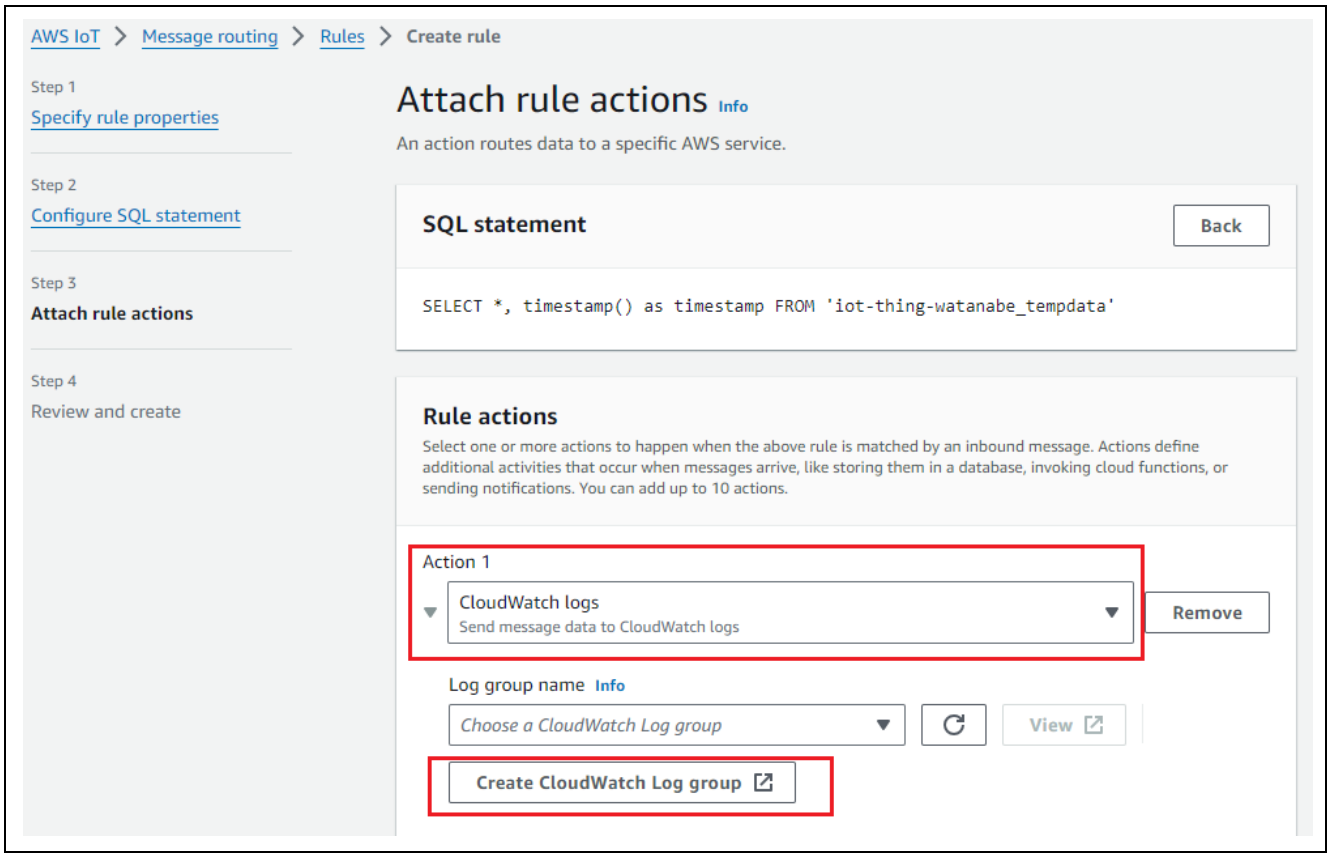


**Figure 5-14 Select “CloudWatch logs” for Action1**

- A new CloudWatch tab launches and confirms that it is the Create Log Group screen.

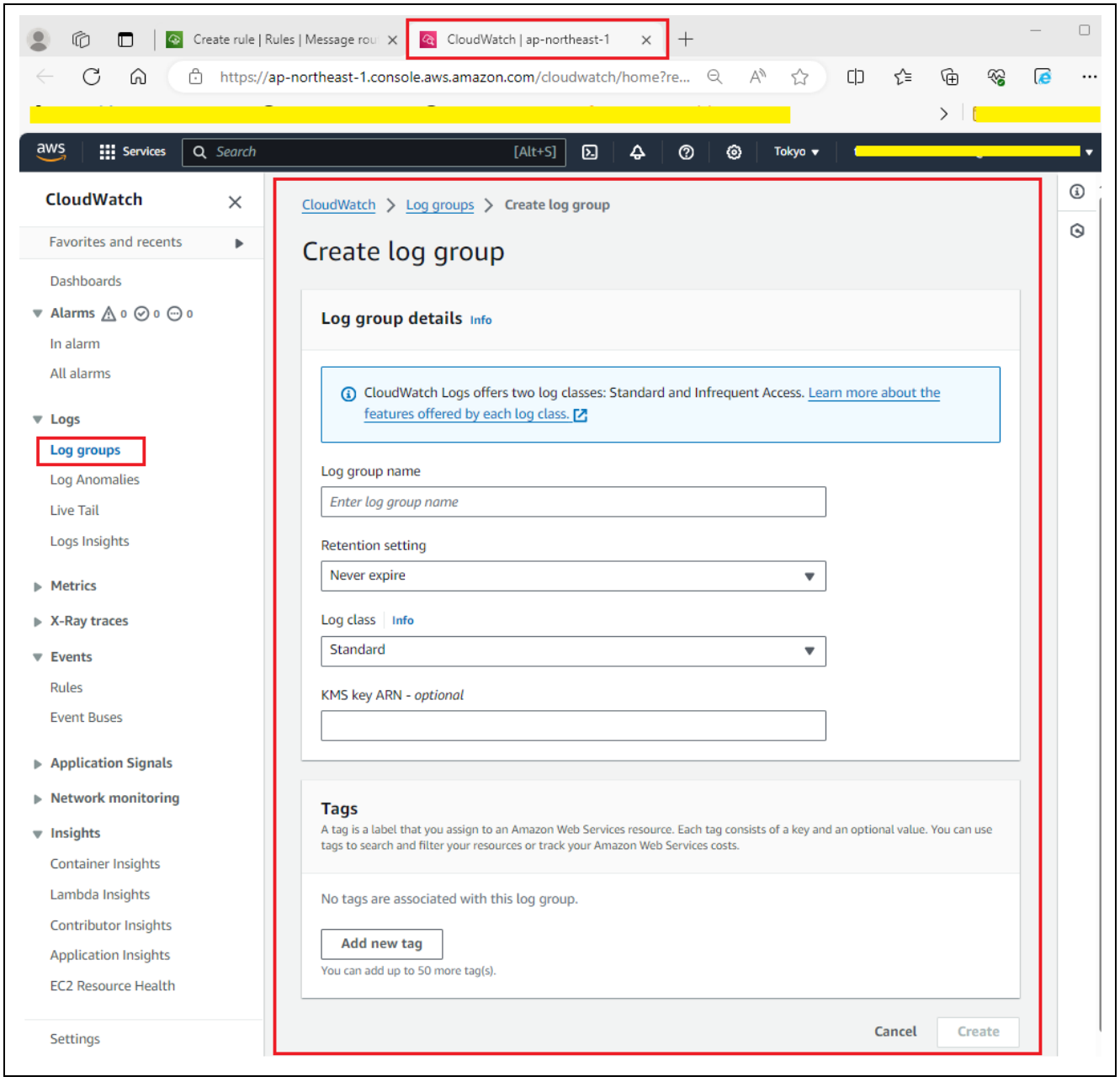


**Figure 5-15 A new CloudWatch tab launches and confirms that it is the Create Log Group screen**

### (5) Create log group

- Enter the Log group name and click “Create” to confirm that the log group has been created.

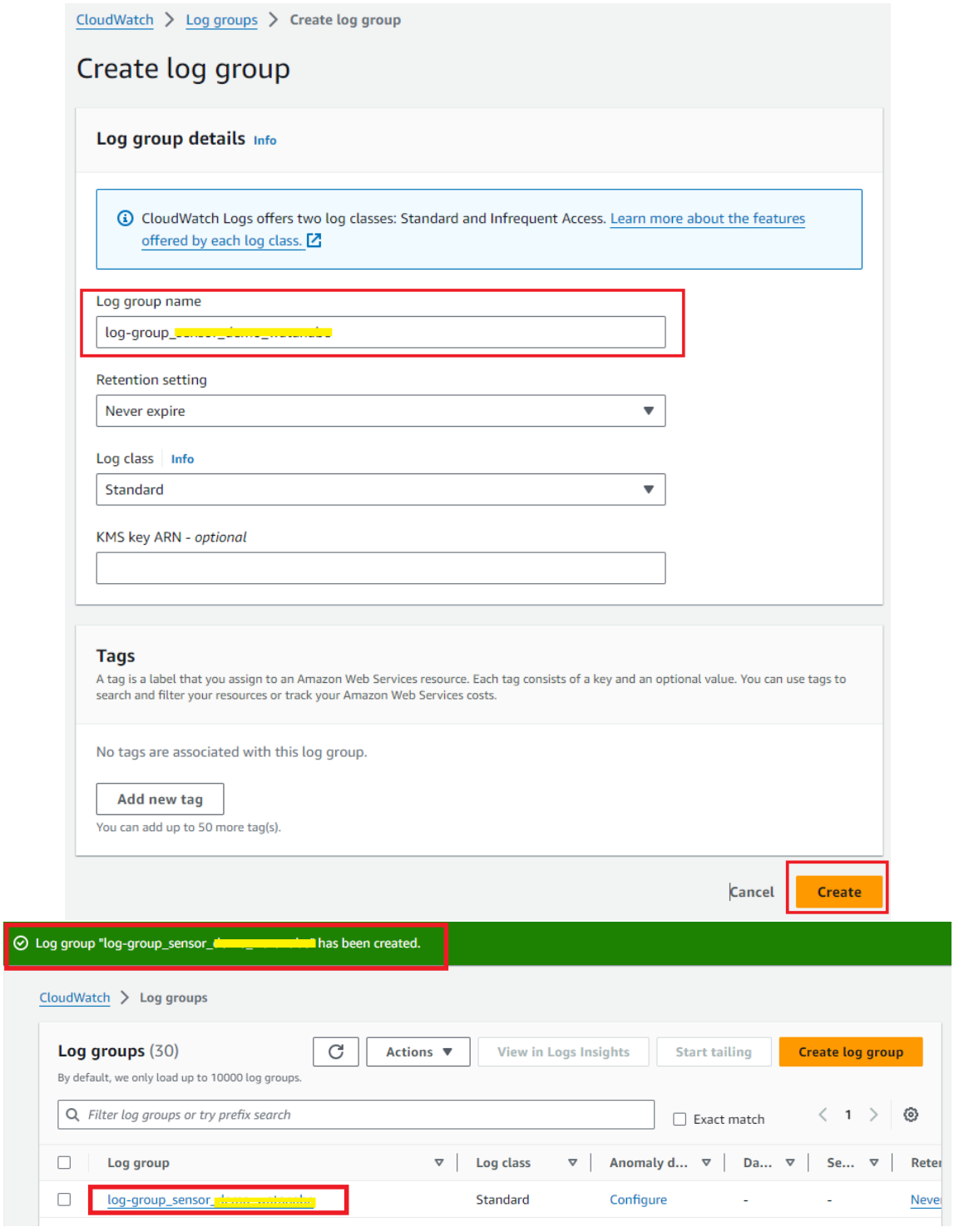


**Figure 5-16 Confirm the log group is created correctly**

### (6) Select the log group you created

- Click on the Rules tab to activate.

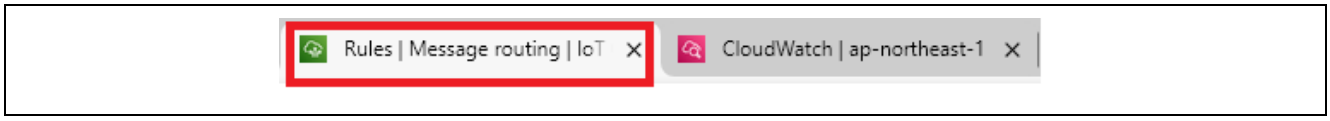


**Figure 5-17 Click on the Rules tab to activate**

- Select the log group created in (5) for the Log group name and click “Create new role”.

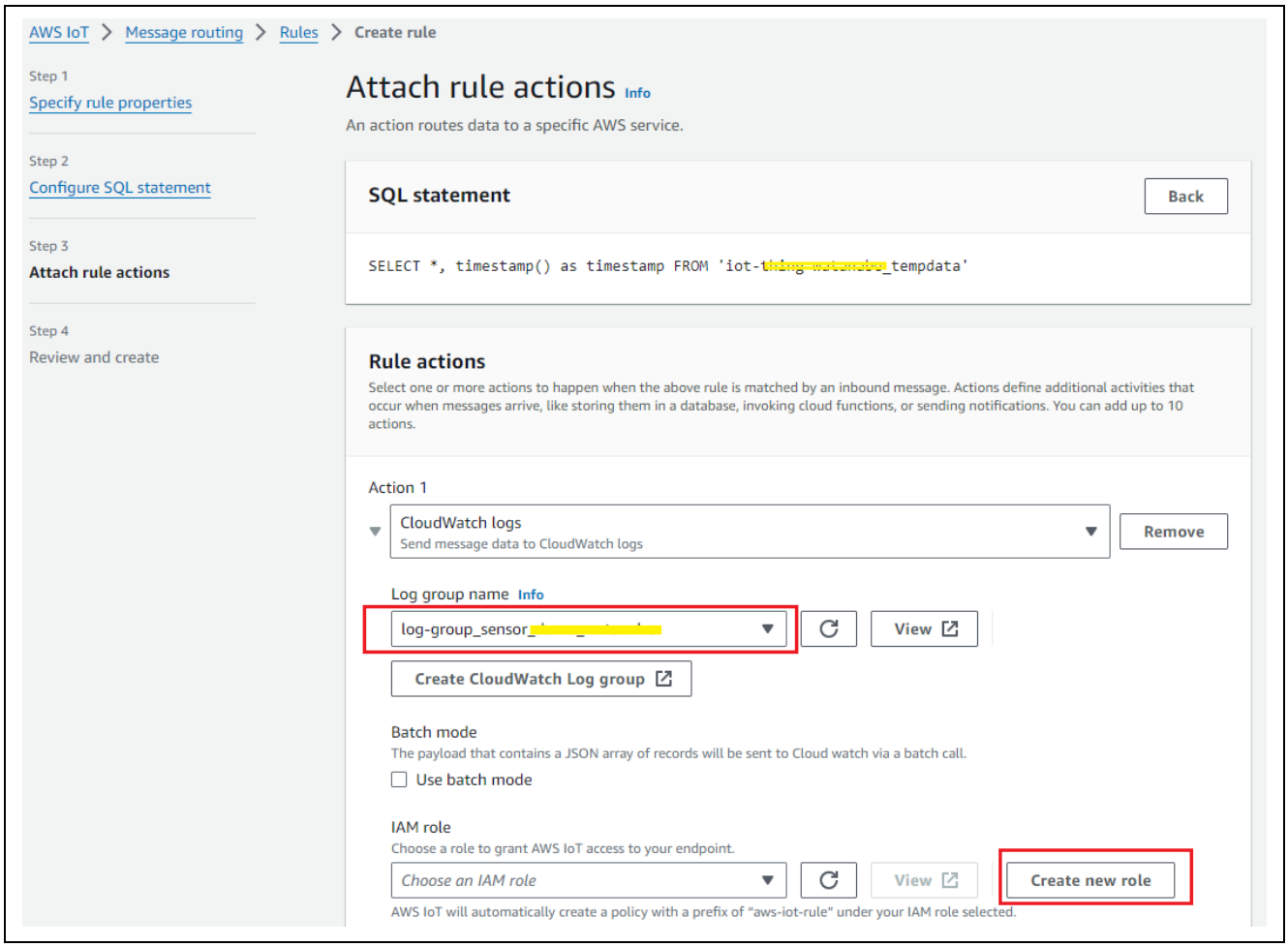


**Figure 5-18 Select the created log group for the Log group name**

### (7) Create IAM role

- Enter the Role name and click “Create”

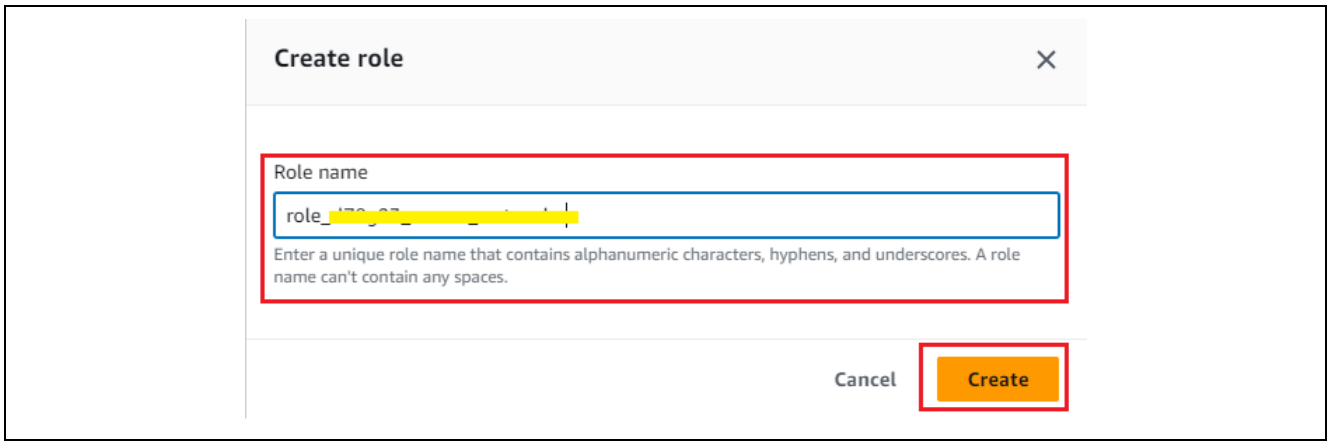


**Figure 5-19 Enter the Role name and click “Create”**

### (8) Choose the IAM role

- Choose the IAM role created in (7) and click "Next".

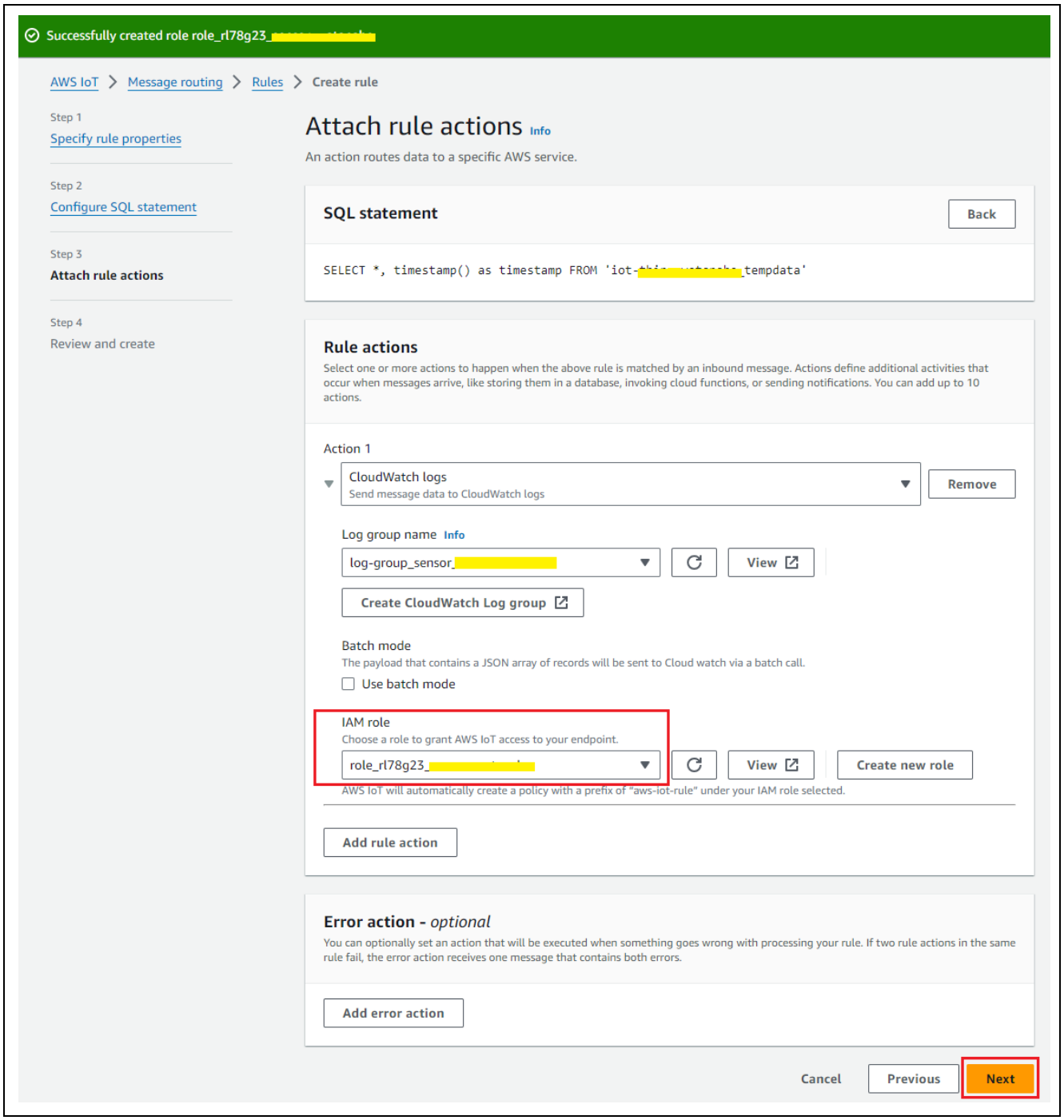


**Figure 5-20 Choose the IAM role and click "Next"**

### (9) Check the contents and create the rules

- Check the contents of the rule and click “Create”

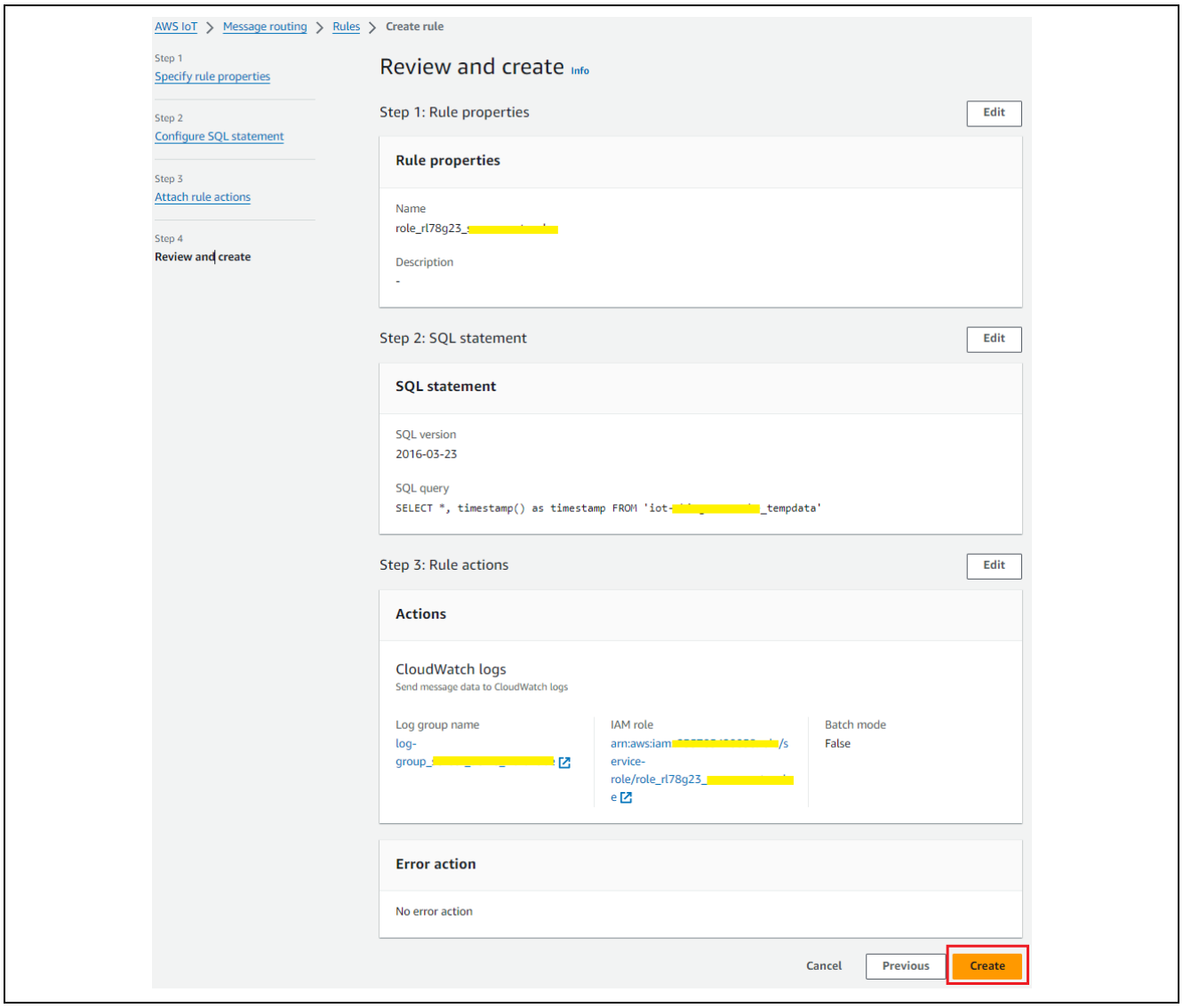


**Figure 5-21 Check the contents of the rule and click “Create”**

- Check the rule was created


**Figure 5-22 Check the rule was created**

### (10) Display graphs using CloudWatch

- CloudWatch > Logs Insights
  - Log group: Select the log group created in 0
  - Query: Enter the query below

```
stats avg(humidity), avg(temperature), max(warning) by bin(10s)
```

  - Visualization: Select “Visualization”

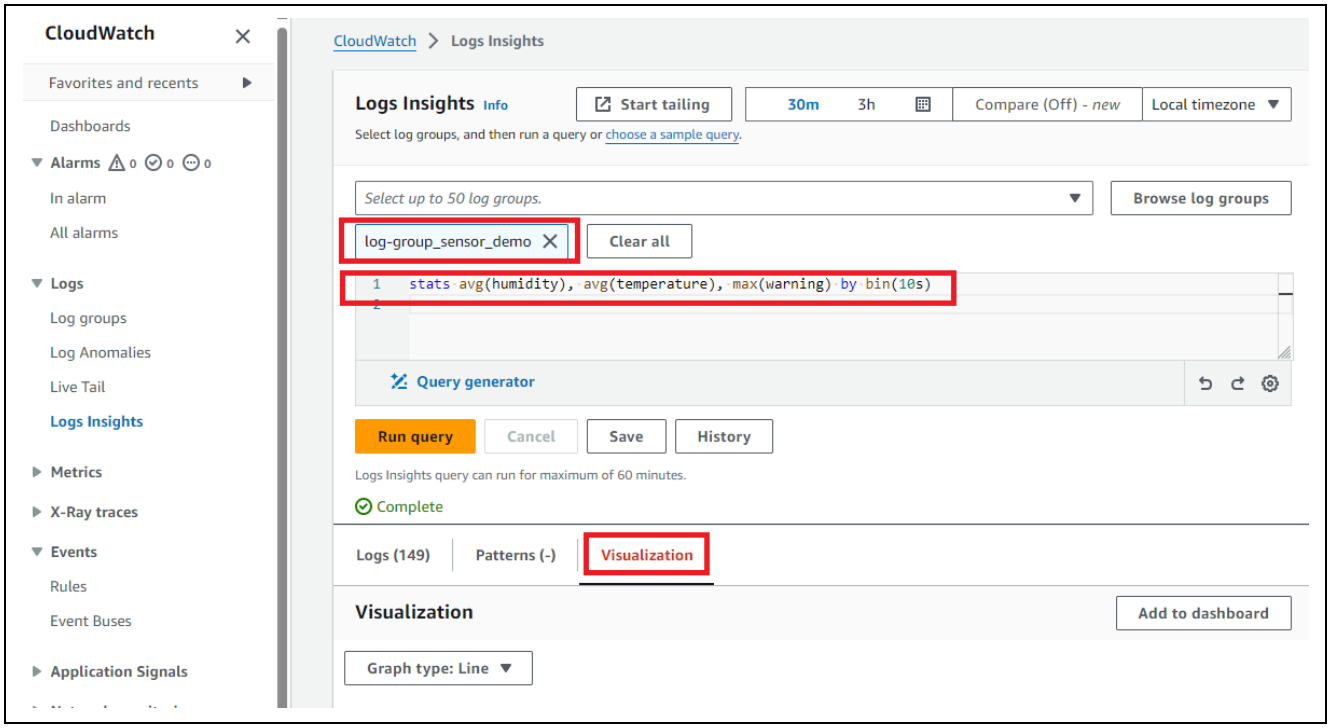


**Figure 5-23 CloudWatch > Logs Insights**

  - After running the demo project, click [Run query] to display the graph

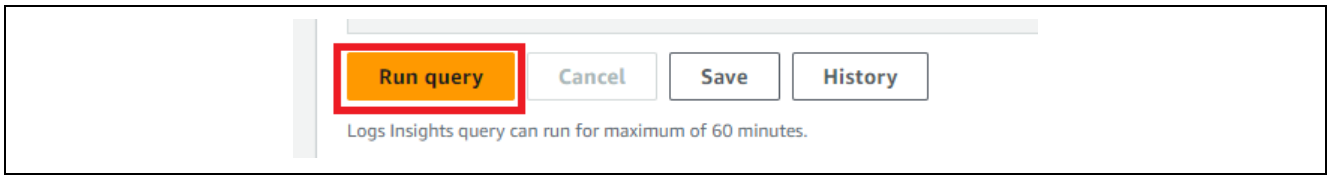


**Figure 5-24 Click [Run query] to display the graph**

### 5.4.2 Settings for OTA Update

#### 5.4.2.1 Creating Amazon S3 Buckets

Refer to section 6.1.3.1 Creating Amazon S3 Buckets in the Getting Started Guide.

#### 5.4.2.2 Creating an OTA Update Service Role

Refer to section 6.1.3.2 Creating an OTA Update Service Role in the Getting Started Guide.

#### 5.4.2.3 Creating an OTA Update User Policy

Refer to section 6.1.3.3 Creating an OTA Update User Policy in the Getting Started Guide.

#### 5.4.2.4 Allocating an OTA Update Policy to IAM User

Refer to section 6.1.3.4 Allocating an OTA Update Policy to IAM User in the Getting Started Guide.

#### 5.4.2.5 Granting Access Permissions to AWS IoT Code Signature

Refer to section 6.1.3.5 Granting Access Permissions to AWS IoT Code Signature in the Getting Started Guide.

## 5.5 Creating an Initial Image

An initial image is a MOT file generated by joining a bootloader's MOT file and an initial application's MOT file by using Renesas Image Generator.

Renesas Image Generator is a tool provided with the RL78/G22, RL78/G23, RL78/G24 firmware update module. For details, refer to the application note in this link.

The file names related to an initial image are as follows in this document.

- Bootloader: boot_loader.mot
- Initial application: aws_ryz024a_sensor_rl78g23-fpb.mot
- Initial image: initial_image.mot

### 5.5.1 Creating a Bootloader

The steps in this chapter are the same as the Getting Started Guide.

#### 5.5.1.1 Importing the Bootloader Project

Refer to section 6.2.1.1 Importing the Bootloader Project in the Getting Started Guide.

#### 5.5.1.2 Adding the Firmware Verification Key to the Bootloader Project

Refer to section 6.2.1.2 Adding the Firmware Verification Key to the Bootloader Project in the Getting Started Guide.

#### 5.5.1.3 Building the Bootloader Project

Refer to section 6.2.1.3 Building the Bootloader Project in the Getting Started Guide.

### 5.5.2 Creating an Initial Application

The steps in this chapter are the same as those in the Getting Started Guide. Replace the project name, command parameters, etc. with the contents of this application note.

#### 5.5.2.1 Importing the Initial Application

Import the aws_ryz024a_sensor_rl78g23-fpb project to e$^2$ studio. Open the Import wizard according to the following process.

File > Import… > Existing Projects into Workspace > Next

Next, select the aws_ryz024a_sensor_rl78g23-fpb project. Ensure that copy projects into the workspace is not selected. Then click the Finish button.

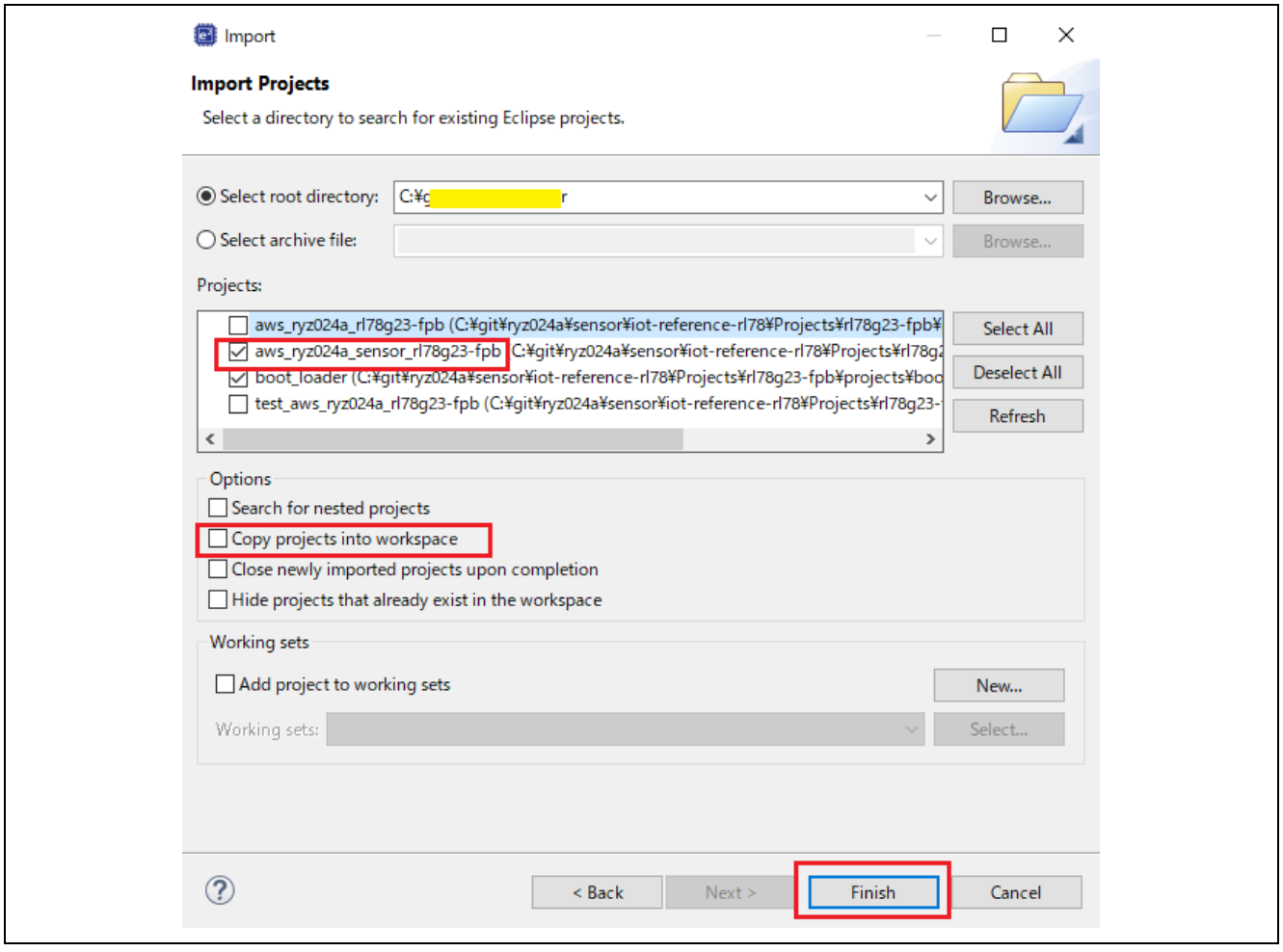


**Figure 5-25 Selecting the aws_ryz024a_sensor_rl78g23-fpb Project**

The imported project is shown in the Project Explorer view.

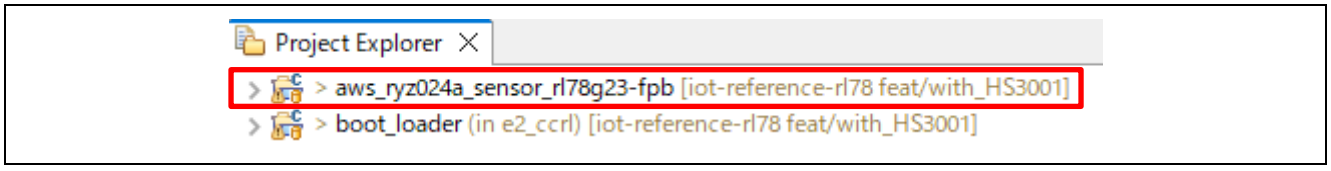


**Figure 5-26 Completing to Import the aws_ryz024a_sensor_rl78g23-fpb Project**

### 5.5.2.2 Adding the Firmware Verification Key to the Initial Application

Add the firmware verification key (secp256r1.publickey) to code_signer_public_key.h in the aws_ryz024a_sensor_rl78g23-fpb project.

Note: Add \ to the end of each line.

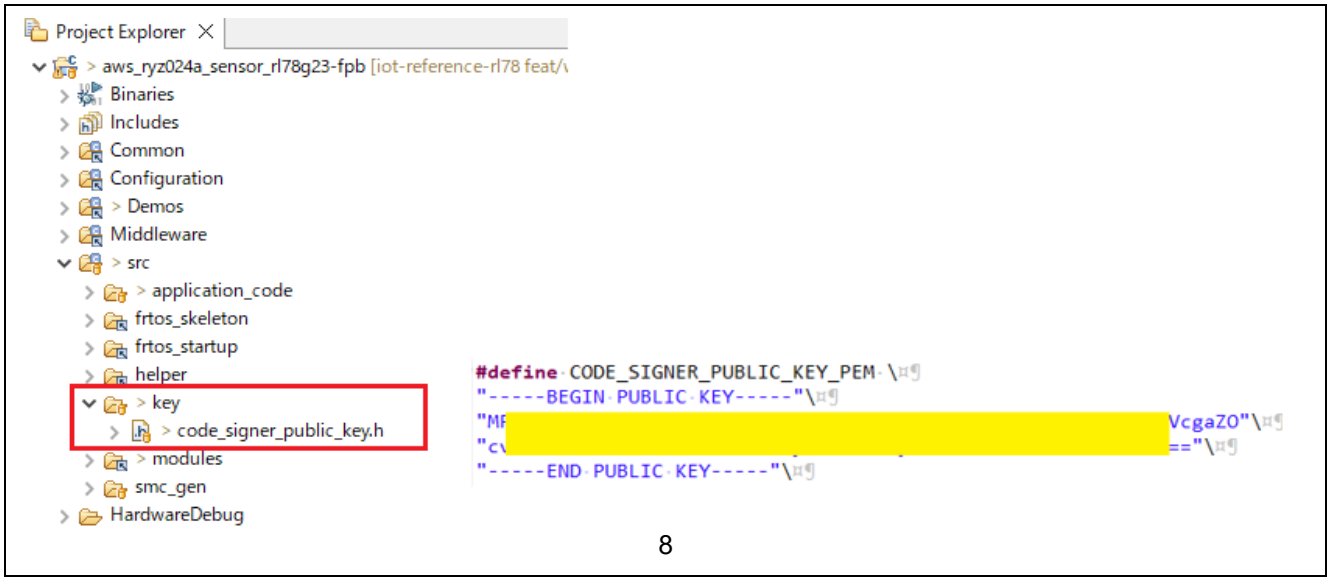


**Figure 5-27 Adding the Firmware Verification Key to the Initial Application**

### 5.5.2.3 Building the Initial Application

Build the aws_ryz024a_sensor_rl78g23-fpb project to create a MOT file.
Then, make sure that aws_ryz024a_sensor_rl78g23-fpb.mot has been created in the HardwareDebug folder directly under the project folder.

### 5.5.3 Creating an Initial Image by Using Renesas Image Generator

Join the bootloader and the initial application by using Renesas Image Generator to generate the initial image.

(1) Store the following files in the same folder as Renesas Image Generator.

- Bootloader: boot_loader.mot
- Initial application: aws_ryz024a_sensor_rl78g23-fpb.mot
- Private key for initial application verification: secp256r1.privatekey

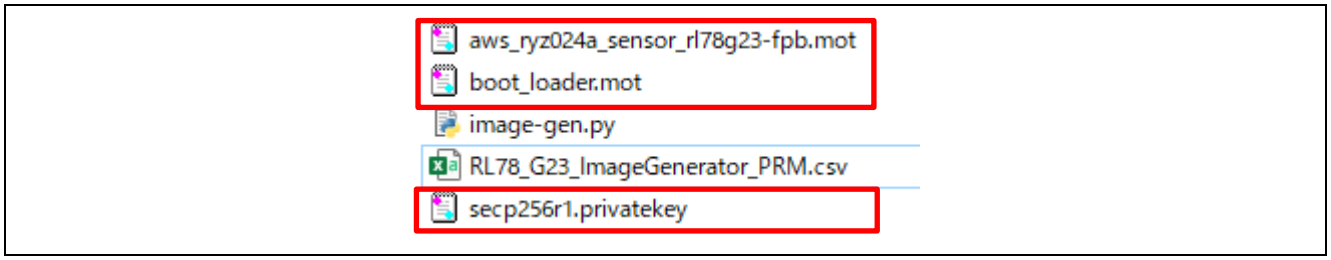


**Figure 5-28 Storing Necessary Files in the Same Folder as Renesas Image Generator**

(2) Run the following command to generate the initial image.

```
python image-gen.py -iup .\aws_ryz024a_sensor_rl78g23-fpb.mot -ibp
boot_loader.mot -o initial_image -ip .\RL78_G23_ImageGenerator_PRM.csv
```

(3) Make sure that the initial image (initial_image.mot) has been generated.

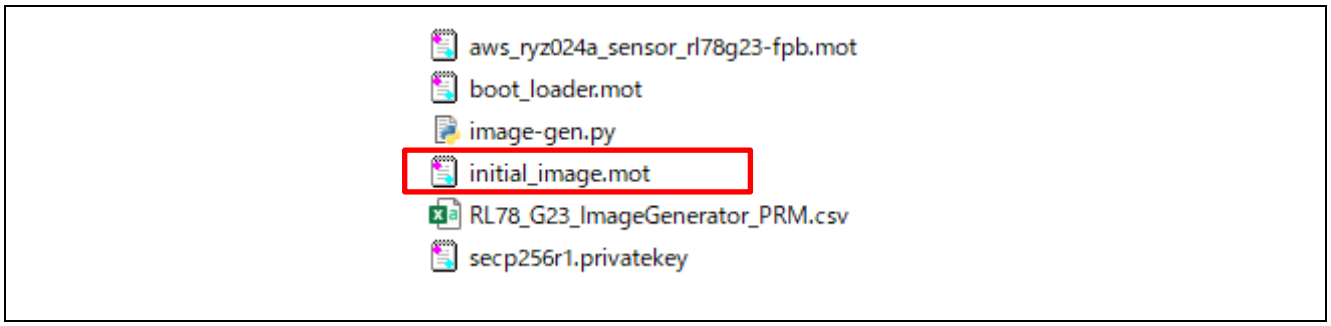


**Figure 5-29 Initial Image Generated in the Same Folder as Renesas Image Generator**

## 5.6 Creating an Update Image

An Update Image is a binary format (extension: rsu) firmware used for the update which are converted an update application's MOT file by using Renesas Image Generator. update images can be generated by Renesas Image Generator. For details about the Update Image format, refer to "RL78/G22,RL78/G23,RL78/G24 Firmware Update Module".

The file names related to an update image are as follows in this document.

- Update application: aws_ryz024a_sensor_rl78g23-fpb_093.mot
- Update image: aws_ryz024a_sensor_rl78g23-fpb_093.rsu

### 5.6.1 Creating an Update Application

The steps in this chapter are the same as those in the Getting Started Guide. Replace the project name, command parameters, etc. with the contents of this application note.

#### 5.6.1.1 Changing the Source Code of the Application

To create an update application,
in iot-reference-rl78\Configuration\rl78g23-fpb\sensor\cellular\frtos_config\demo_config.h, change the definition of the APP_VERSION_BUILD macro from 2 to 3.

```
iot-reference-rl78\Configuration\rl78g23-
fpb\sensor\cellular\frtos_config\demo_config.h

/**
 * @brief Build version of the firmware.
 *
 * This is used in the OTA demo to set the appFirmwareVersion variable that
is
 * declared in the ota_appversion32.h file in the OTA library.
 */
#ifndef APP_VERSION_BUILD
    #define APP_VERSION_BUILD    3
#endif
```

In iot-reference-rl78\Demos\Sensor\SensorDemo.c, change the definition of the SENSOR_ENABLE_HUMIDITY macro from 0 to 1.

```
iot-reference-rl78\Demos\Sensor\SensorDemo.c

/* Sensor for HSxxxx series */
#include "sensor.h"
#define SENSOR_ENABLE_HUMIDITY                          ( 1 )
```

#### 5.6.1.2 Building the Update Application

Build the aws_ryz024a_sensor_rl78g23-fpb project to create a MOT file.
Then, make sure that aws_ryz024a_sensor_rl78g23-fpb.mot has been overwritten and created in the HardwareDebug folder directly under the project folder.

#### 5.6.1.3 Renaming the MOT File of the Update Application

Rename aws_ryz024a_sensor_rl78g23-fpb.mot to aws_ryz024a_sensor_rl78g23-fpb_093.mot.

### 5.6.2 Generating an Update Image by Using Renesas Image Generator

Convert the update application to an update image by using Renesas Image Generator.

(1) Store the following files in the same folder as Renesas Image Generator.

- MOT file of the update application: aws_ryz024a_sensor_rl78g23-fpb_093.mot
- Private key for update application verification: secp256r1.privatekey

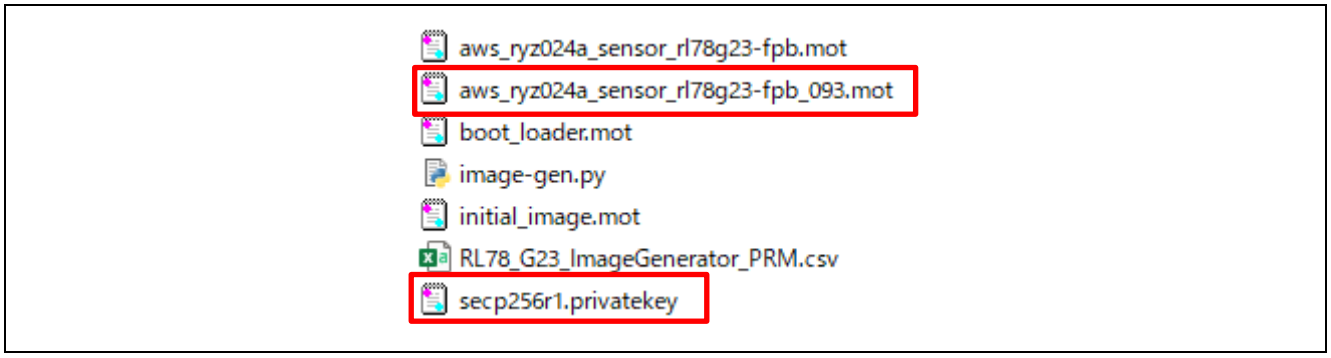


**Figure 5-30 Storing Necessary Files in the Same Folder as Renesas Image Generator**

(2) Run the following command to generate an update image (aws_ryz024a_sensor_rl78g23-fpb_ota_093.rsu) in RSU format.

```
python image-gen.py -iup .\aws_ryz024a_sensor_rl78g23-fpb_093.mot -o
aws_ryz024a_sensor_rl78g23-fpb_093 -ip .\RL78_G23_ImageGenerator_PRM.csv -vt
ecdsa -ff RTOS
```

(3) Make sure that aws_ryz024a_sensor_rl78g23-fpb_ota_093.rsu has been generated.

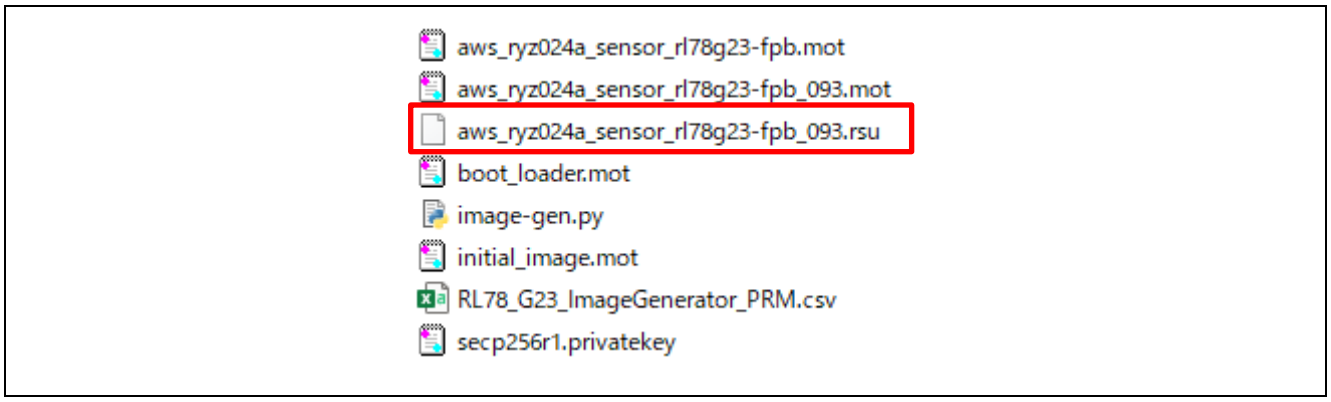


**Figure 5-31 Update Image Generated in the Same Folder as Renesas Image Generator**

## 5.7 Running the Demo Project

The following describes the running procedure for the demo project.

### 5.7.1 Programming the Initial image (initial_image.mot) to Board

(1) Program the initial image (initial_image.mot).
For the programming method, refer to Chapter 6 Using Renesas Flash Programmer

(2) Using Renesas Flash Programmer .

(3) Remove the USB cable to temporarily stop the power supply to the MCU board.

(4) Reconnect the USB cable to start power supply to the MCU board and start the project.

(5) Check the terminal to make sure that the initial application (version 0.9.2) has started and sending temperature data obtained from the sensor board.

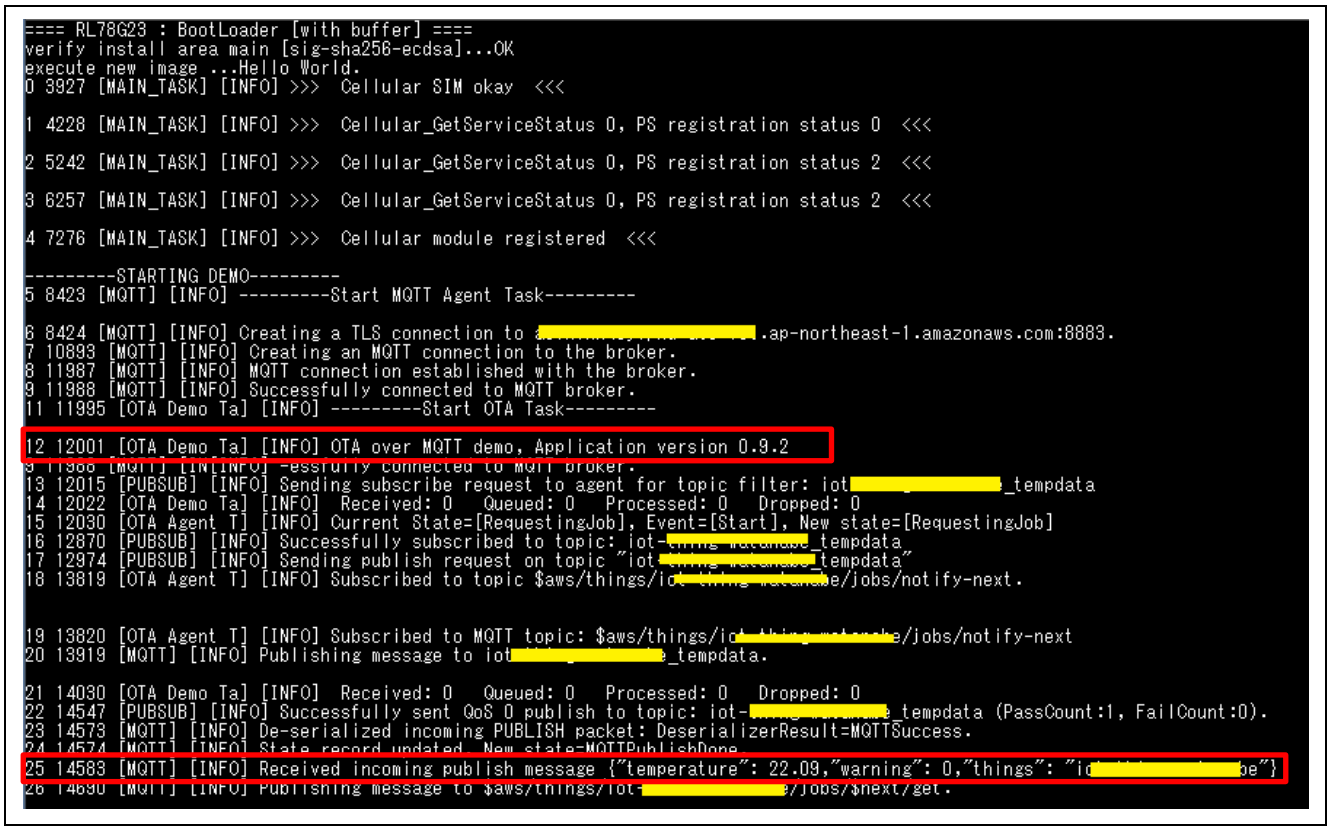


**Figure 5-32 Initial Application (Version 0.9.2) Started**

(6) View the graph in CloudWatch.

- CloudWatch > Logs Insights > Click “Run query”
  - temperature: This is a temperature graph.
  - warning: This is a temperature warning. False if less than 25 degrees (value is 0). True if 25 degrees or higher (value is 50).

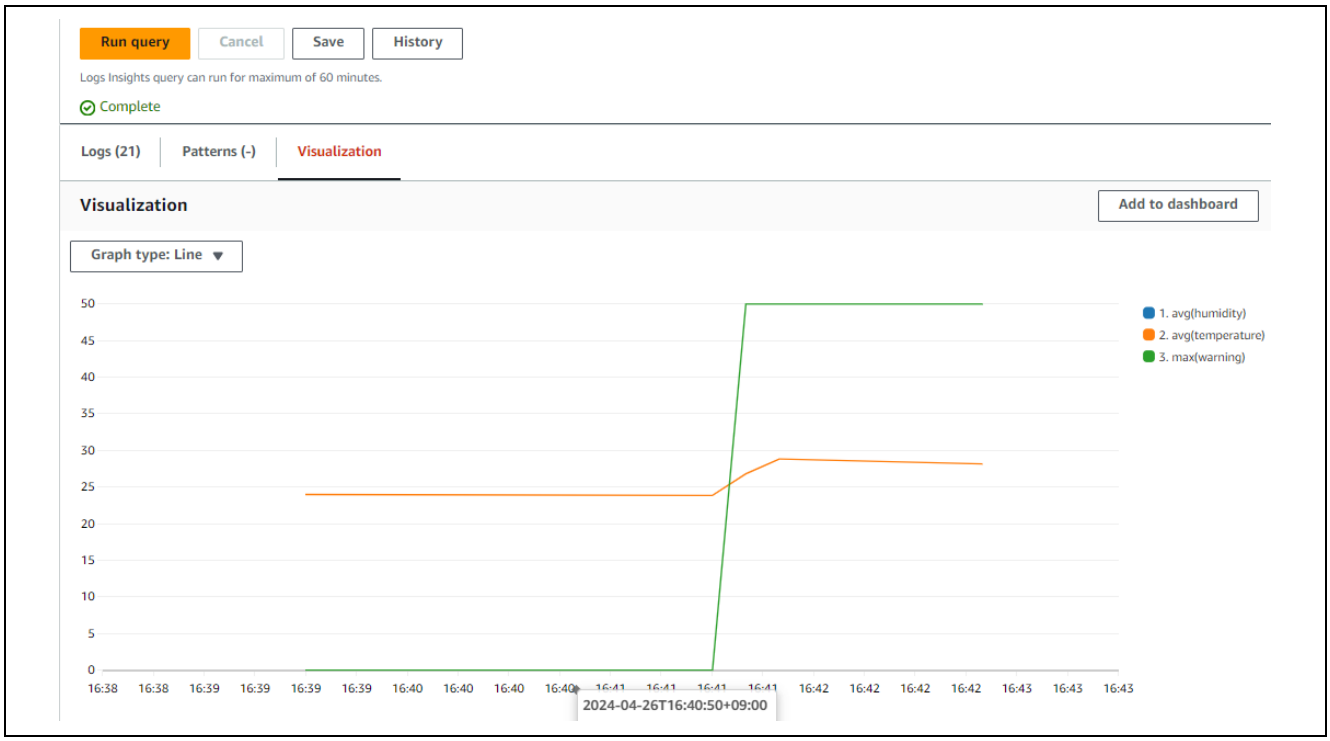


**Figure 5-33 Displaying graphs in CloudWatch after launching the Initial Application (version 0.9.2)**

### 5.7.2 Registering the Update Image (aws_ryz024a_sensor_rl78g23-fpb_093.rsu) with OTA Jobs

(1) AWS IoT > Manage > Remote actions > Jobs > Click “Create job”

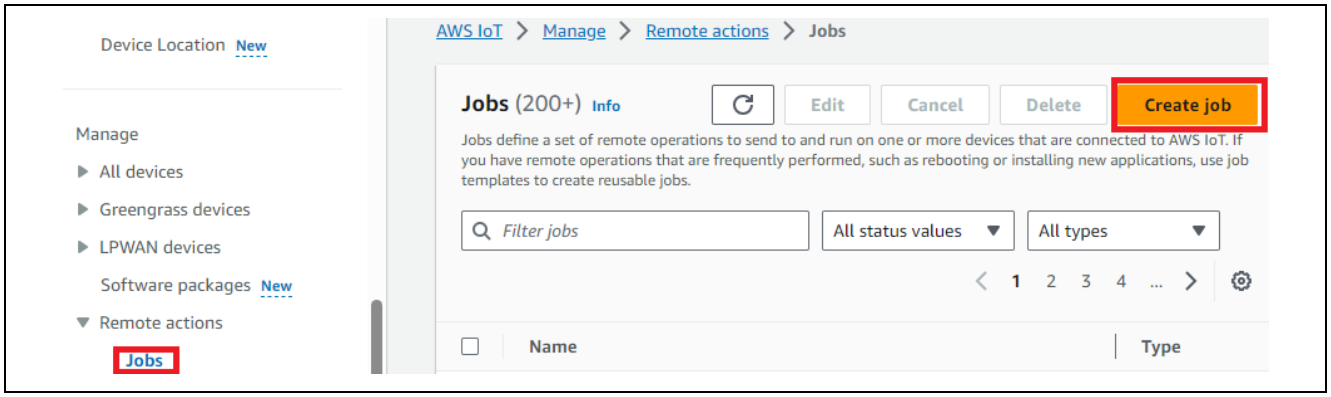


**Figure 5-34 Jobs**

(2) Check “Create FreeRTOS OTA update job” > Click “Next”

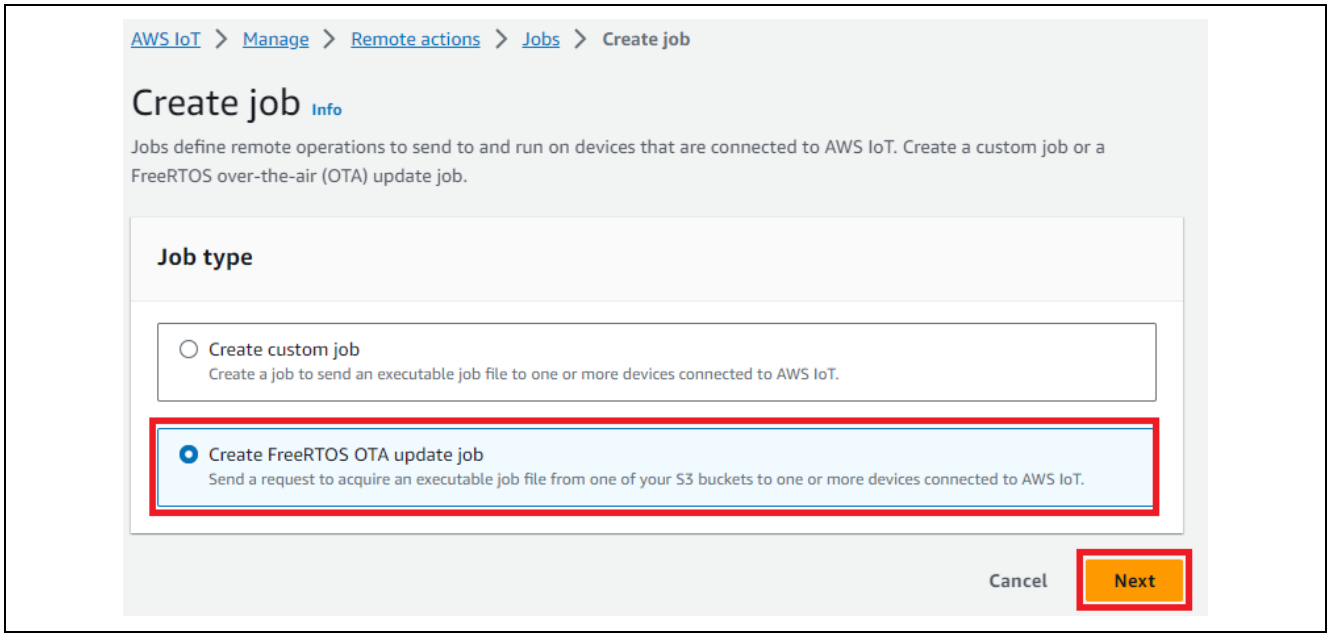


**Figure 5-35 Create Job**

(3) Step1: OTA job properties

- Job name: Any

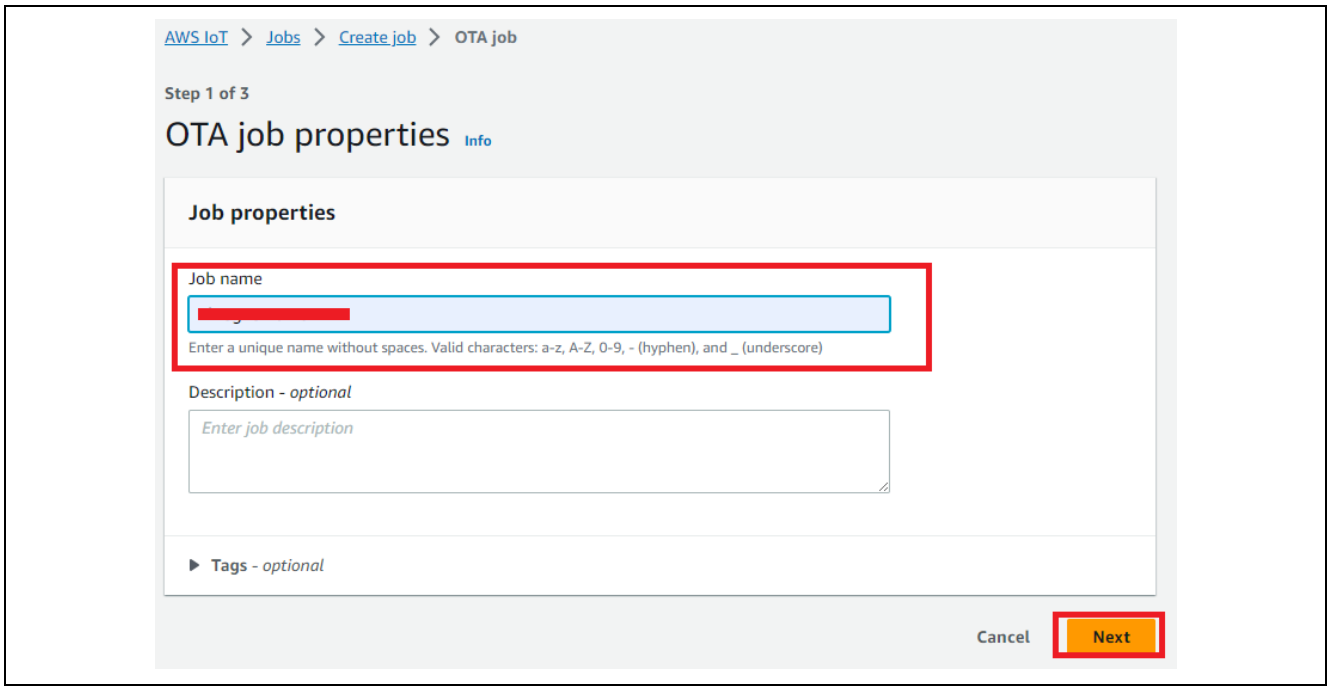


**Figure 5-36 Step1: OTA job properties**

(4) Step2: OTA file configuration > Devices

- Devices to update: “Name of the thing” in aws_clientcredential.h

```
#define clientcredentialIOT_THING_NAME          "YOUR_THING_NAME
```

- Select the protocol for file transfer: MQTT

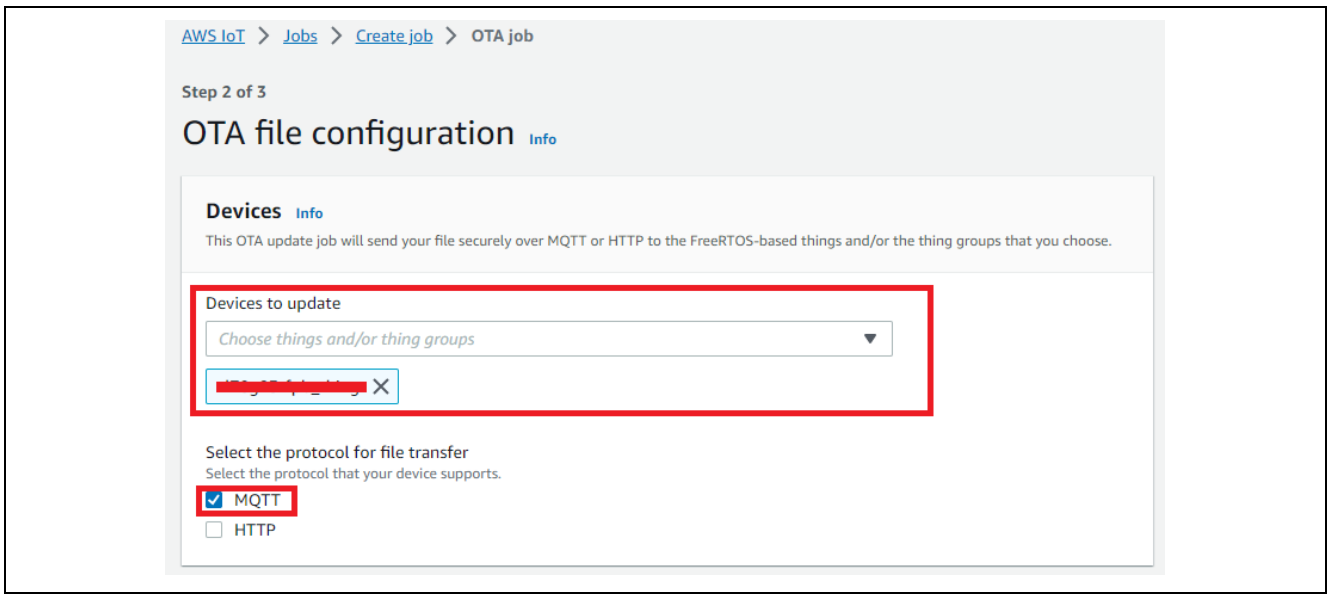


**Figure 5-37 Step2: OTA file configuration > Devices**

(5) Step2: OTA file configurations > File

- Sign and choose your file: Sign a new file for me.

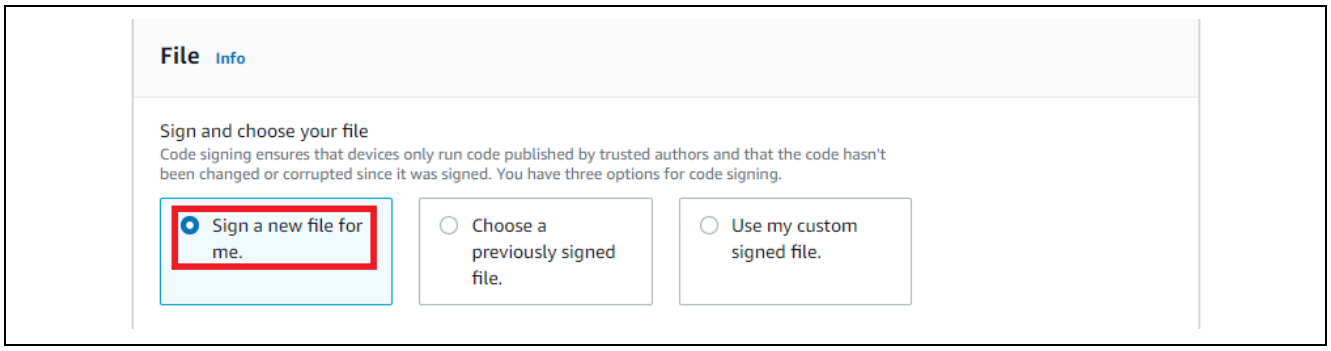


**Figure 5-38 Step2: OTA file configurations > File (1)**

- Code signing profile: Click “Create new profile”

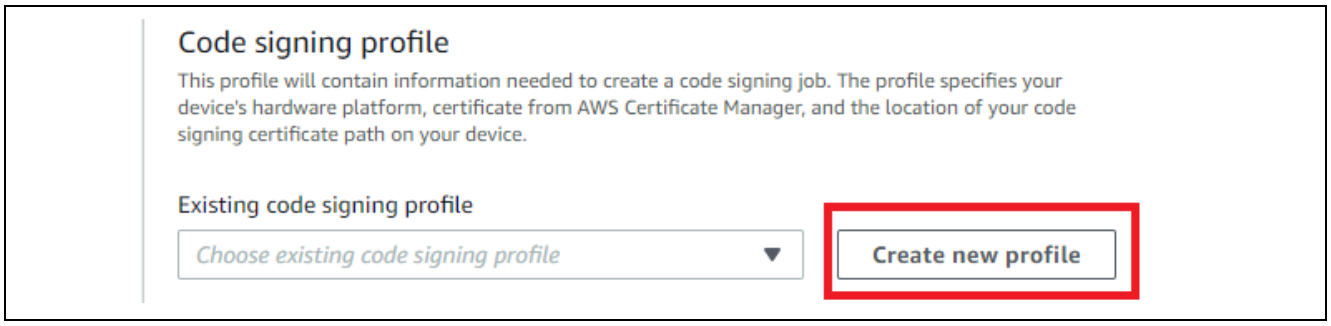


**Figure 5-39 Step2: OTA file configurations > File (2)**

- Create a code signing profile.
  - Profile name: Any (Example: rl78g23_fpb_ota_cert)
  - Device hardware platform: Windows Simulator
  - Code signing certificate: “Import new code signing certificate”
  - Certificate body: secp256r1.crt
  - Certificate private key: secp256r1.privatekey
  - Certificate chain - optional: ca.crt
  - Path name of code signing certificate on device: Any

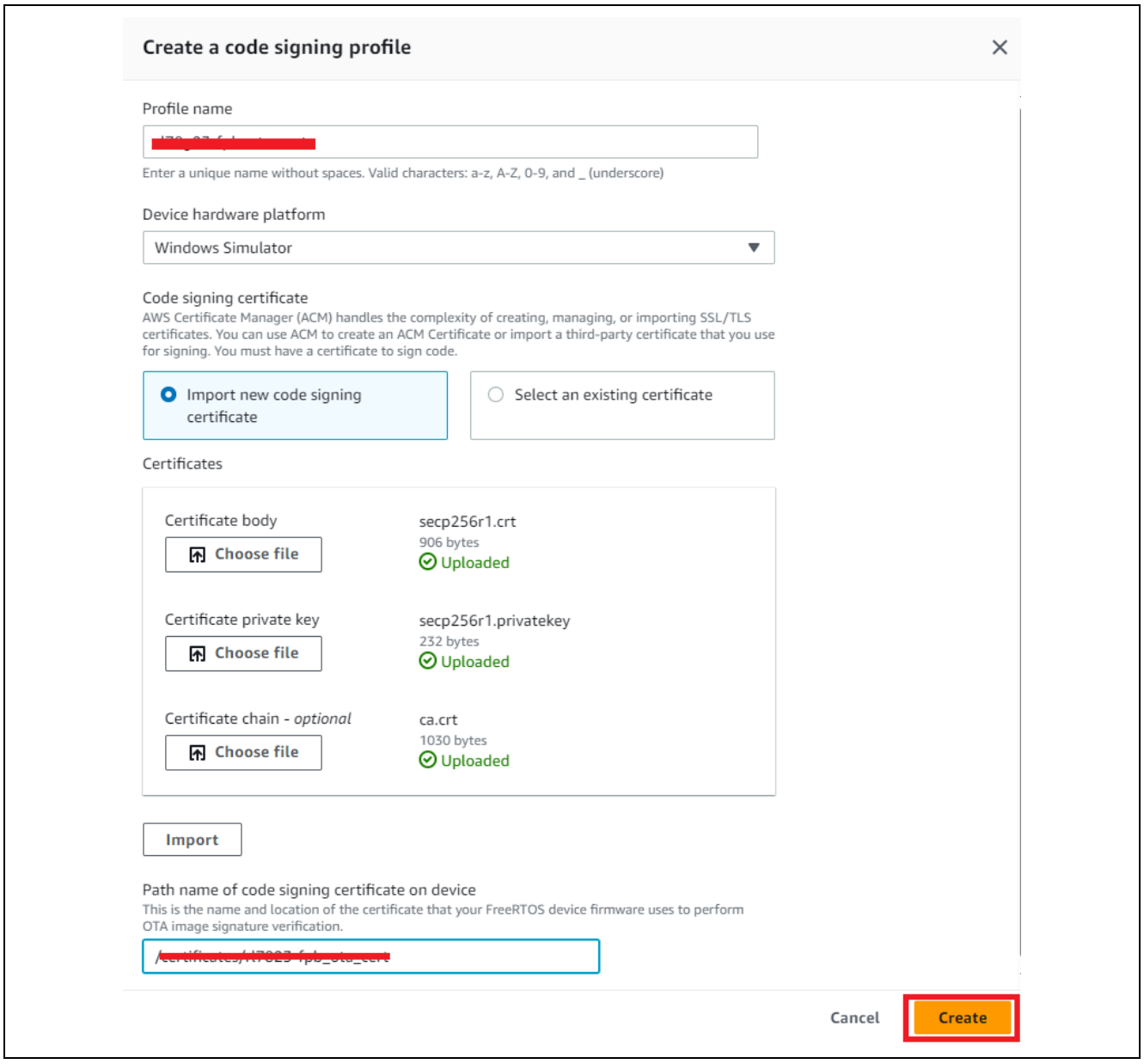


**Figure 5-40 Create a code signing profile**

- File > “Upload a new file.” > “Choose file” > aws_ryz024a_sensor_rl78g23-fpb_093.rsu

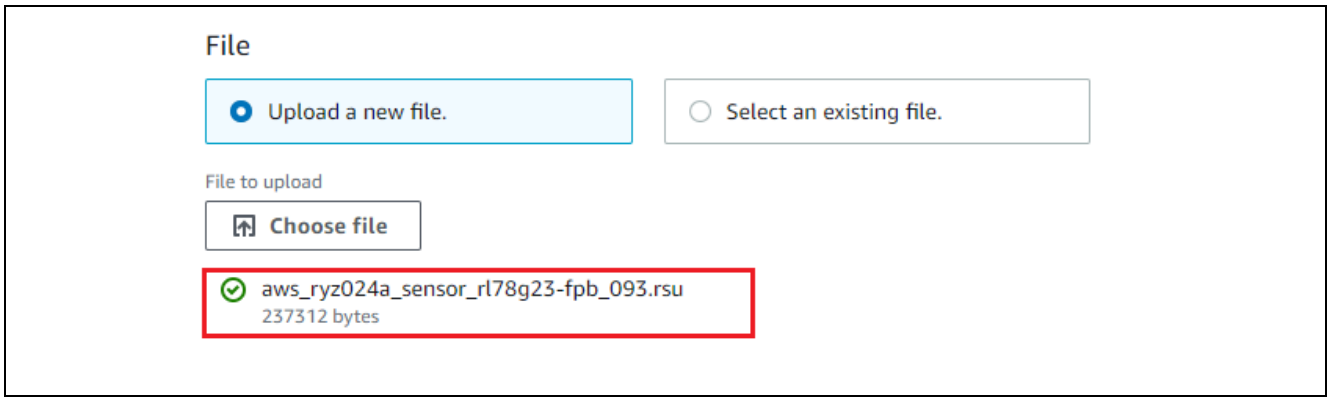


**Figure 5-41 Upload a new file > aws_ryz024a_sensor_rl78g23-fpb_093.rsu**

- File upload location in S3: Specify the created bucket (Bucket name specified in 5.4.2.1 Creating Amazon S3 Buckets)
- Path name of file on the device: Any

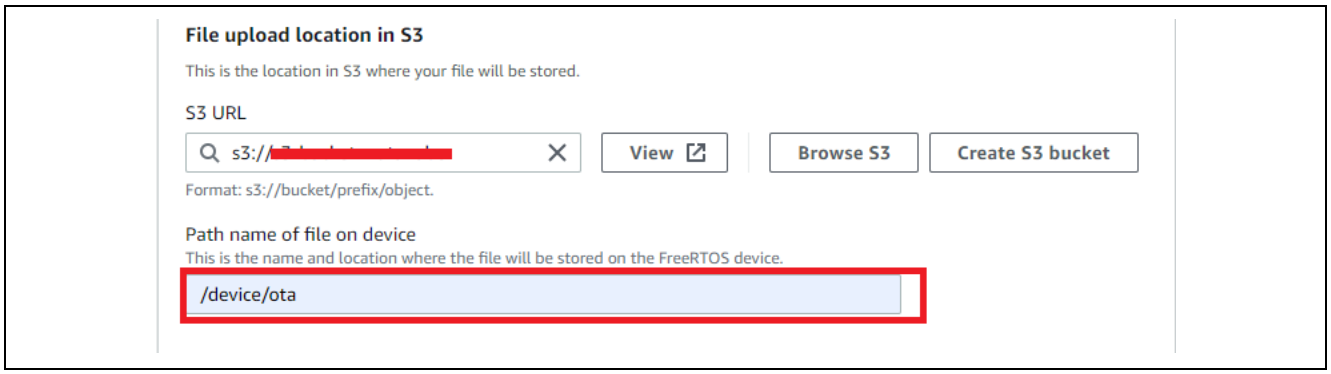


**Figure 5-42 File upload location in S3**

(6) Step2: OTA file configurations > IAM role

- Role: Specify the created role (Role name specified in 5.4.2.2 Creating an OTA Update Service Role )

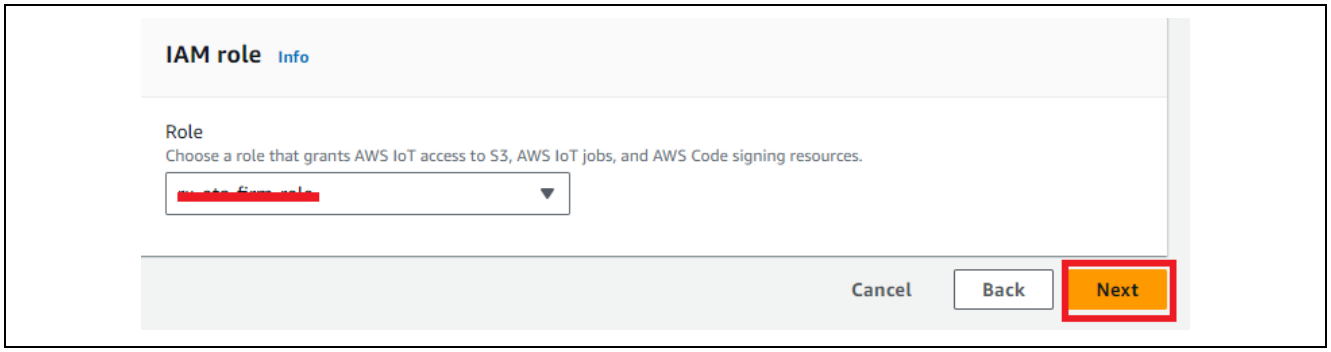


**Figure 5-43 Step2: OTA file configurations > IAM role**

(7) Step3: OTA job configuration

- Job run type: Your job will complete after deploying to the devices and groups that you chose (snapshot)

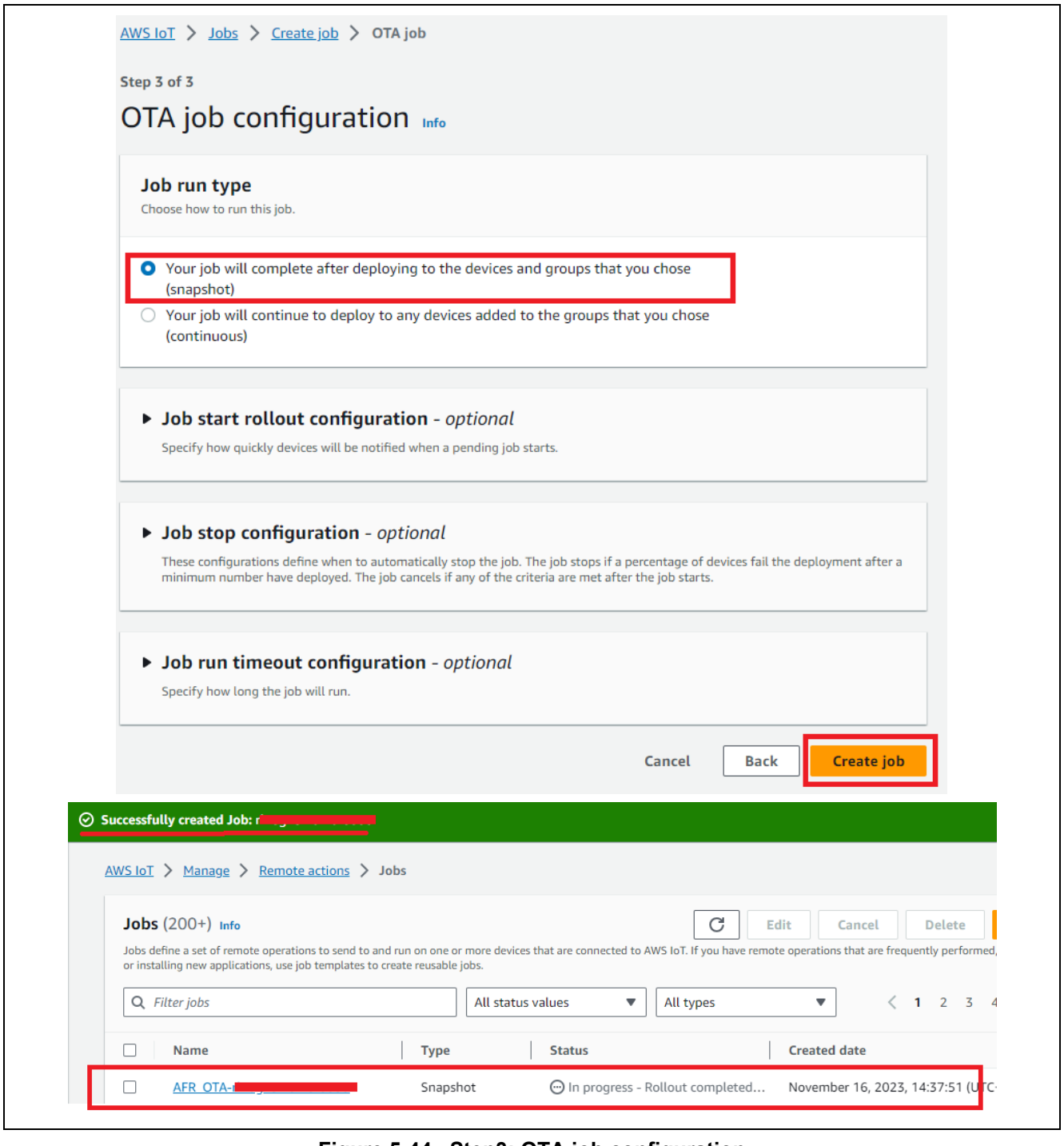


**Figure 5-44 Step3: OTA job configuration**

(8) After a while (it may be about 10 minutes), the log of programming the update image to the MCU board is output to the terminal.

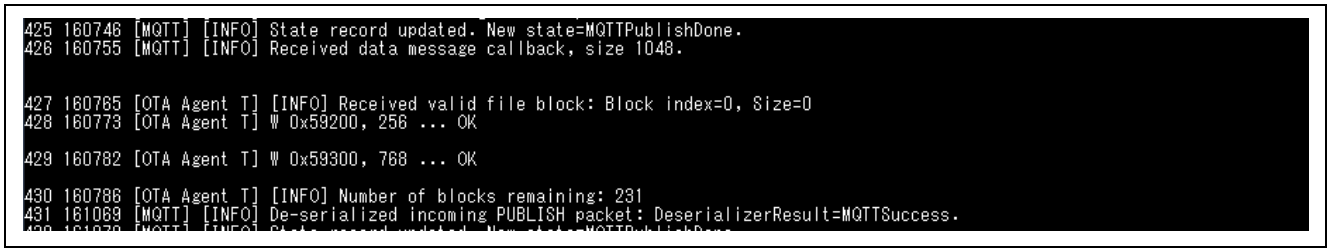


**Figure 5-45 Programming the Update Image to the MCU Board**

(9) After the writing is completed, confirm that the Update Image (version 0.9.3) starts and sends the temperature and humidity data obtained from the sensor board.

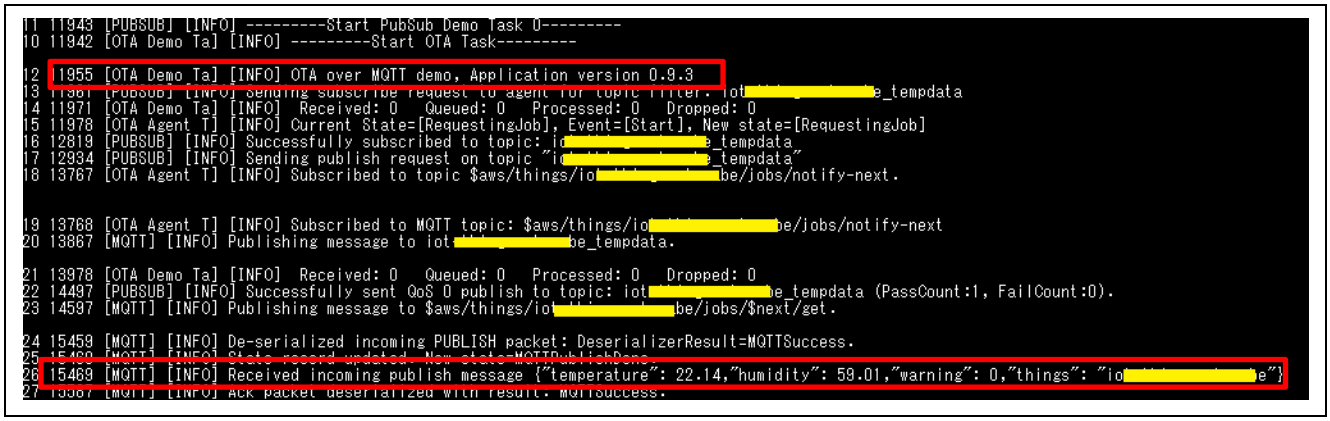


**Figure 5-46 Update Image (Version 0.9.3) Started after Programming Terminates**

(10) View the graph in CloudWatch.

- CloudWatch > Logs Insights > Click “Run query”
  - humidity: This is a humidity graph.
  - temperature: This is a temperature graph.
  - warning: This is a temperature warning. False if less than 25 degrees (value is 0). True if 25 degrees or higher (value is 50).

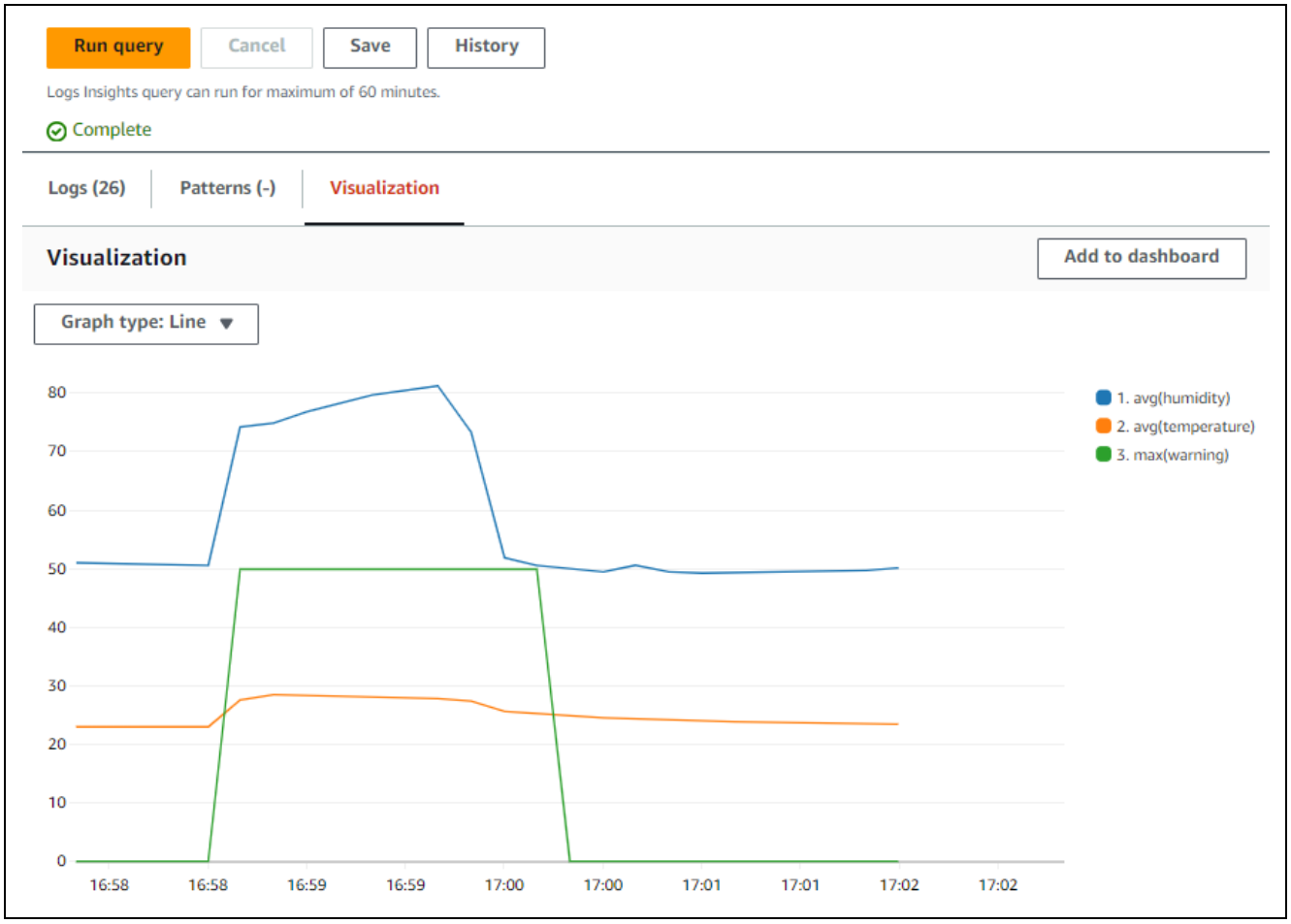


**Figure 5-47 Displaying graphs in CloudWatch after launching the Update Image (version 0.9.3)**

## 5.8 Debugging the Initial Application

The following describes the procedure for starting the initial application from e$^2$ studio and debugging it. Because the bootloader is not used in this procedure, downloaded update images cannot be started.

(1) Change the setting to not use the bootloader.

Change the “USE_BOOTLOADER_V2” macro of the aws_ryz024a_sensor_rl78g23-fpb project to 0, and then click “Apply and Close”.

- Configuration: HardwareDebug
- Languages: GNU C
- USE_BOOTLOADER_V2: 0

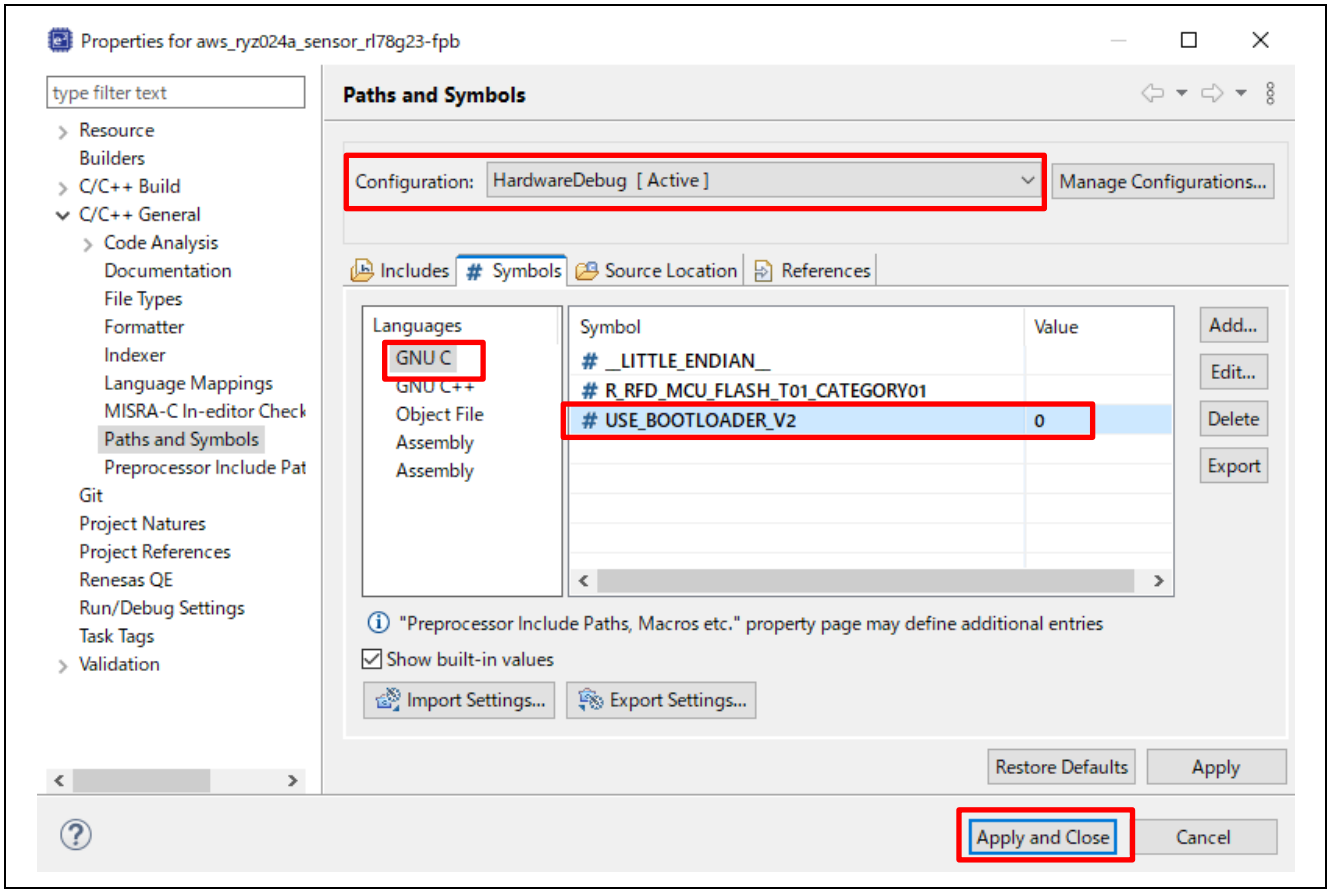


**Figure 5-48 Setting the “USE_BOOTLOADER_V2” Macro to 0**

(2) Build the aws_ryz024a_sensor_rl78g23-fpb project.

(3) Start debugging.

Refer to Chapter 7, Debug Procedure.

# 6. Using Renesas Flash Programmer

Refer to section 7. Using Renesas Flash Programmer in the Getting Started Guide.

# 7. Debug Procedure

Refer to section 8. Debug Procedure in the Getting Started Guide.

# 8. Appendix

## 8.1 Precautions on Porting Third-Party Libraries to RL78

Because RL78 is a 16-bit system, the following must be noted when applying a third-party library to RL78.

### 8.1.1 Width of int Is 16 Bits

Refer to section 9.1.1 Width of int Is 16 Bits in the Getting Started Guide.

### 8.1.2 Size Limitation of Section

Refer to section 9.1.2 Size Limitation of Section in the Getting Started Guide.

## 8.2 License Information for Open Source Software Used with Demo Projects

The demo projects of this product use the Open Source Software (OSS). The user must comply with the license terms stipulated by OSS. Check the license terms on the official website of the respective OSS. Table 1-4 Operation Confirmation Conditions (Others, such as OSS Library) shows the link of each OSS.

## 9. Websites and Supports

Sample programs in this Getting Started Guide: https://github.com/renesas/iot-reference-rl78

AWS forum: http://forums.aws.amazon.com

## Revision History

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.00 | Jul. 31, 2024 | - | First edition issued |

# 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/.