

RX Family

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

Preface

This application note explains how to use the OTA demo application on FreeRTOS (with IoT Libraries). For details about security, see the Renesas MCU Firmware Update Design Policy ([R01AN5548](#))

Note: The procedures in this application note are for FreeRTOS-v202210.01-LTS-rx-1.1.0 and later. For earlier versions, see *How to implement FreeRTOS OTA by using Amazon Web Services on RX65N* ([R01AN5549](#)).

Target Device

RX65N and RX651 groups

Hardware

CK-RX65N

Related Documents

Renesas MCU Firmware Update Design Policy ([R01AN5548](#))

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

Contents

1. Overview	3
1.1 System overview	3
1.2 Operation verification environment for Hardware.....	4
1.3 Operation verification environment for Software	4
2. Prerequisites.....	5
2.1 Installing Tera Term.....	5
2.2 Installing Python	6
2.3 Installing OpenSSL.....	7
2.4 Installing Renesas Image Generator.....	8
2.5 Connecting the CK-RX65N	10
3. Setting Up AWS.....	11
3.1 Signing in to the AWS Console	12
3.2 Setting your region in AWS	14
3.3 Registering your device in AWS.....	14
3.3.1 Setting policies	14
3.3.2 Registering your device as a <i>thing</i> in AWS IoT.....	16
3.3.3 Checking the endpoint.....	20
3.4 Creating an Amazon S3 bucket.....	21
3.5 Allocating OTA execution permission to IAM users	24
4. Setting Up the Device	33
4.1 Generating key pairs and certificates	33
4.2 Creating the initial version of the firmware	36
4.2.1 Importing the project.....	36
4.2.2 Checking the project environment settings	38
4.2.3 Setting up projects.....	40
4.2.4 Creating the initial firmware.....	46
4.2.5 Registering AWS IoT information	50
5. Updating the Firmware	57
5.1 Creating the updated firmware	57
5.1.1 Changing the firmware version.....	57
5.2 Updating the firmware	58
6. Troubleshooting.....	66
Revision History.....	67

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

1. Overview

1.1 System overview

This section shows an overview of implementing OTA using a RX65N microcontroller that supports the dual bank function and has the CK-RX65N cloud kit installed.

The dual bank function divides the ROM of the microcontroller into an execution area and a temporary area. By dynamically swapping the execution area and the temporary area, updated firmware can be written to ROM while the existing software continues to run in the execution area.

The following explains how memory is allocated during OTA updates, and what happens when memory banks are swapped using the dual bank function.

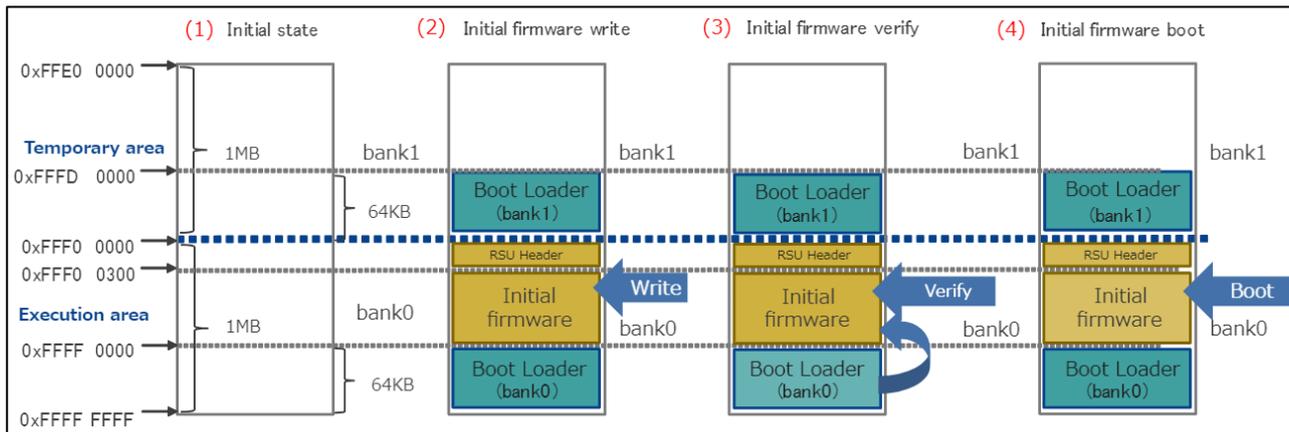


Figure 1.1 Overview of OTA operation (1)

- (1) The state that erased all the data by Renesas Flash Programmer.
- (2) Memory state after writing combined bootloader and initial firmware data* in Renesas Flash Programmer.

* It refers to the data that Boot Loader(bank0) + Initial firmware + RSU Header + BootLoader(bank1) are combined. For details of RSU Header, please refer to 4.2 image file of RX Family Firmware Update module Using Firmware Integration Technology Application Notes([R01AN6850](#)).

- (3) After reset, bootloader (bank0) verifies Initial firmware.
- (4) Initial firmware is booted.

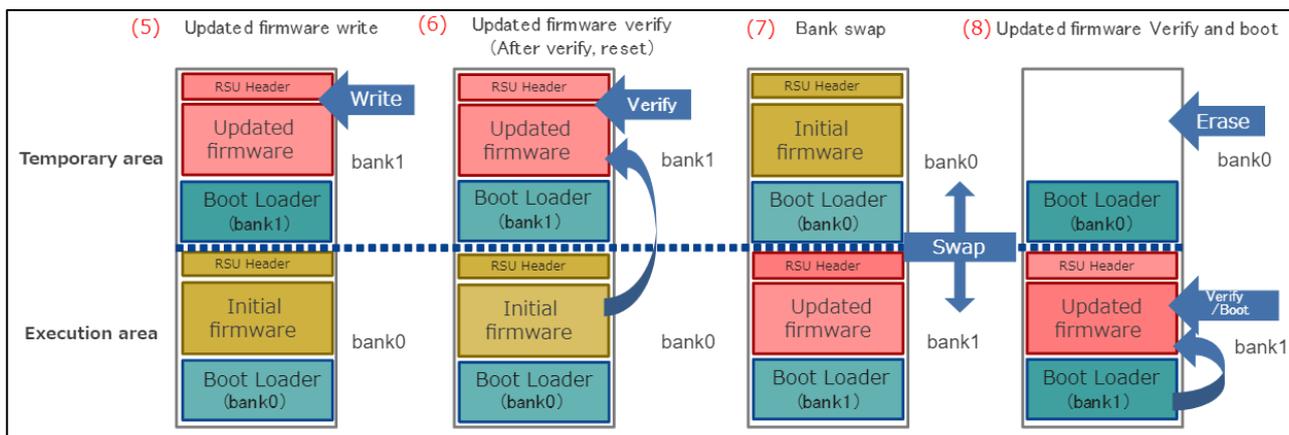


Figure 1.2 Overview of OTA operation (2)

- (5) When new firmware is received from AWS, it is written to bank1.
Initial firmware operation is executed by BGO function while writing to bank1.

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

- (6) Updated firmware is verified by the initial firmware.
- (7) Bank0 and bank1 are swapped, and bank1 is designated as the execution area.
- (8) Updated firmware is verified by bootloader.

Erase Initial firmware in bank0 and execute Updated firmware written to bank1.

1.2 Operation verification environment for Hardware

Table 1-1 Table 1-2 Operation verification environment for Hardware

Item	
Board	CK-RX65N (Cellular/Ethernet) *1
Cellular module	RYZ014A (incorporated into CK-RX65N)
SIM	LTE Cat-M1-compatible SIM (micro-SIM) *2

Notes: 1. Cellular communication is used in this sample.

2. To use the SIM card supplied with the CK-RX65N kit, activate the SIM card by following the procedure in [4.1.5 Activating SIM card](#) in the following application note:

[SIM activation, Creating the trial account and using Dashboard with RYZ014A or Ethernet Application for AWS - Getting Started Guide \(R01QS0064\)](#)

1.3 Operation verification environment for Software

Table 1-3 Operation verification environment for Hardware

Item	
Integrated development environment	e ² studio 2023-10
Compiler	Renesas CC-RX v3.05.00
FreeRTOS	v202210.01-LTS-rx-1.1.0
Log monitoring tool	TeraTerm v4.106
Python	Python 3.11.0
Keygen tool	Win64 OpenSSL v3.0.12
Flash programming tool	Renesas Flash Programmer V3.12.00
Renesas Image Generator	Version 3.02 (supplied with Firmware Update module Rev.2.01)

2. Prerequisites

2.1 Installing Tera Term

- (1) Access the Tera Term download site
[Tera Term download web site \(GitHub\)](#)
- (2) Download the Tera Term installer

Jul 12

M nmaya

teraterm-4_106

0433752

Compare

Tera Term 4.106 Latest

Source code is not available.

▼ Assets 4

teraterm-4.106.exe	12.2 MB	Jul 12
teraterm-4.106.zip	8.63 MB	Jul 12
Source code (zip)		Jul 12
Source code (tar.gz)		Jul 12

👍 4 4 people reacted

- (3) Run the installer and follow the prompts to install Tera Term
- (4) Confirm that Tera Term starts when you click the Tera Term icon in the Start menu

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

2.2 Installing Python

(1) Access the Python download web site

[Python download web site](#)

(2) Download the Python 3.11.0 installer

Click the **Download** link for Python 3.11.0

Looking for a specific release?
Python releases by version number:

Release version	Release date		Click for more
Python 3.9.16	Dec. 6, 2022	Download	Release Notes
Python 3.8.16	Dec. 6, 2022	Download	Release Notes
Python 3.7.16	Dec. 6, 2022	Download	Release Notes
Python 3.11.0	Oct. 24, 2022	Download	Release Notes
Python 3.9.15	Oct. 11, 2022	Download	Release Notes
Python 3.8.15	Oct. 11, 2022	Download	Release Notes

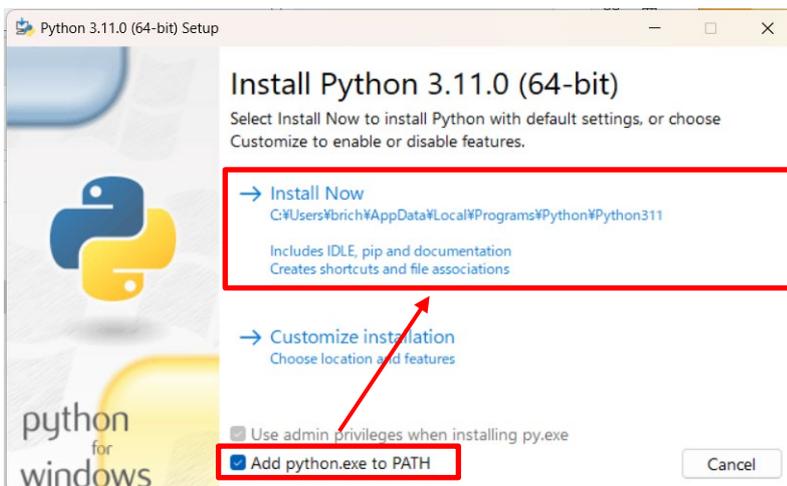
Download the installer for the operating system you are using.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore	
Gzipped source tarball	Source release		c5f77f1ea256dc5bdb0897eeb4d35bb0	26333656	SIG	CRT	SIG
XZ compressed source tarball	Source release		fe92acfa0db9b9f5044958edb451d463	19819768	SIG	CRT	SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	98fa94815780c9330fc2154559365834	42602603	SIG	CRT	SIG
Windows embeddable package (32-bit)	Windows		0888959642cc8af087d88da3866490a5	9560053	SIG	CRT	SIG
Windows embeddable package (64-bit)	Windows		7df0f4244e5a66760b7c7caaed58e86c93	10545380	SIG	CRT	SIG
Windows embeddable package (ARM64)	Windows		e3dbbd5d63c6cb203adc6c0c8ca5f5f7	9765886	SIG	CRT	SIG
Windows installer (32-bit)	Windows		e369a267acaad62487223bd835279bb9	23987136	SIG	CRT	SIG
Windows installer (64-bit)	Windows	Recommended	4fe11b2b0bb0c744cf74aff537f7cd7f	25157416	SIG	CRT	SIG
Windows installer (ARM64)	Windows	Experimental	18e5bd9a4854109adf3b77c7c9dc1ded	24289144	SIG	CRT	SIG

(3) Run the installer and follow the prompts to install Python

On the installation screen, select the **Add python.exe to PATH** check box.



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

- Open a command prompt, and confirm that Python 3.11.0 is installed
Execute the following command and confirm that information appears.

```
python -V
```

```
C:\Users>python -V  
Python 3.11.0
```

- Install the Python encryption library (pycryptodome)
Install the encryption library by executing the following command:

```
pip install pycryptodome
```

```
C:\Users>pip install pycryptodome  
Requirement already satisfied: pycryptodome in c:\users\os:\_ppdata\local\programs\python\python311\lib\site-packages (3.18.0)  
  
[notice] A new release of pip is available: 23.1.2 -> 23.2.1  
[notice] To update, run: python.exe -m pip install --upgrade pip
```

2.3 Installing OpenSSL

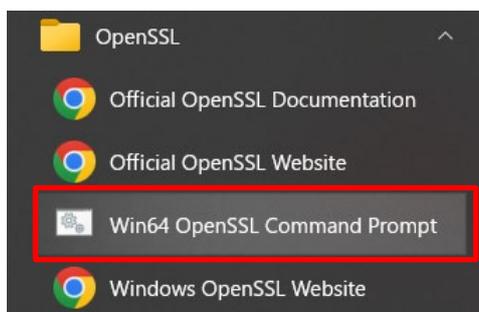
- Access the Win32/Win64 download web site for OpenSSL
[Win32/Win64 OpenSSL Installer for Windows - Shining Light Productions \(slproweb.com\)](http://slproweb.com)

- Download the OpenSSL installer

Download the installer for the operating system you are using.

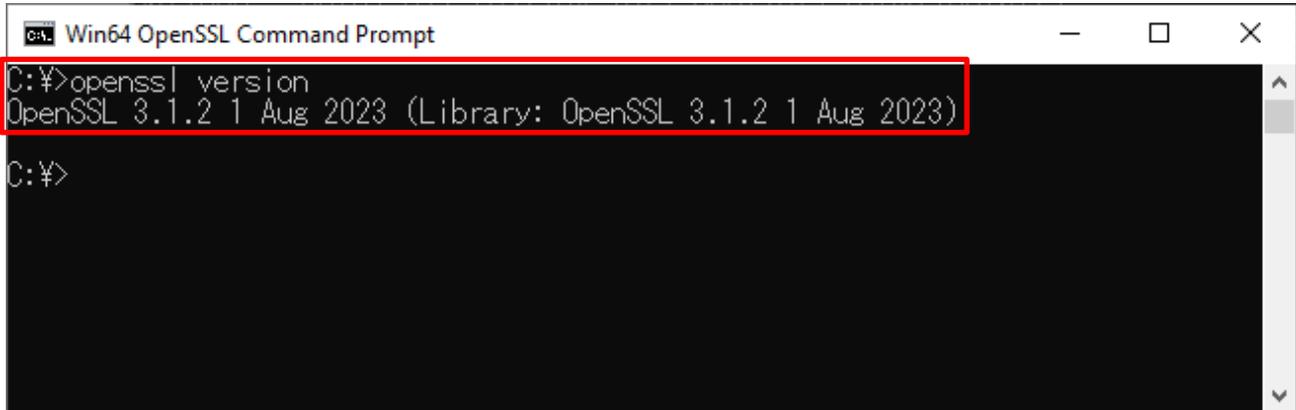
Win64 OpenSSL v3.0.12 Light EXE MSI	5MB Installer	Installs the most commonly used essentials of Win64 C by the creators of OpenSSL . Only installs on 64-bit ve chipsets. Note that this is a default build of OpenSSL a information can be found in the legal agreement of the
Win64 OpenSSL v3.0.12 EXE MSI	140MB Installer	Installs Win64 OpenSSL v3.0.12 (Recommended for sc OpenSSL). Only installs on 64-bit versions of Windows this is a default build of OpenSSL and is subject to loca found in the legal agreement of the installation.
Win32 OpenSSL v3.0.12 Light EXE MSI	4MB Installer	Installs the most commonly used essentials of Win32 C 32-bit OpenSSL for Windows. Note that this is a defau and state laws. More information can be found in the le
Win32 OpenSSL v3.0.12	116MB Installer	Installs Win32 OpenSSL v3.0.12 (Only install this if you

- Run the installer and follow the prompts to install OpenSSL
Select the option to copy the OpenSSL DLLs to the OpenSSL binaries directory.
- From the Start menu, open the Win64 OpenSSL Command Prompt



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

- (5) Confirm that you can run the openssl command from the command prompt
Execute the following command and confirm that version information appears.
openssl version



```
C:\>openssl version
OpenSSL 3.1.2 1 Aug 2023 (Library: OpenSSL 3.1.2 1 Aug 2023)
C:\>
```

2.4 Installing Renesas Image Generator

Renesas Image Generator is a tool that generates the firmware images used by the firmware update module. Renesas Image Generator can generate the following images for use by the firmware update module:

- Initial image: An image file containing the bootloader and application program written by flash writer during initial system configuration (extension: mot)
- Update image: An image file containing the updated firmware (extension: rsu)

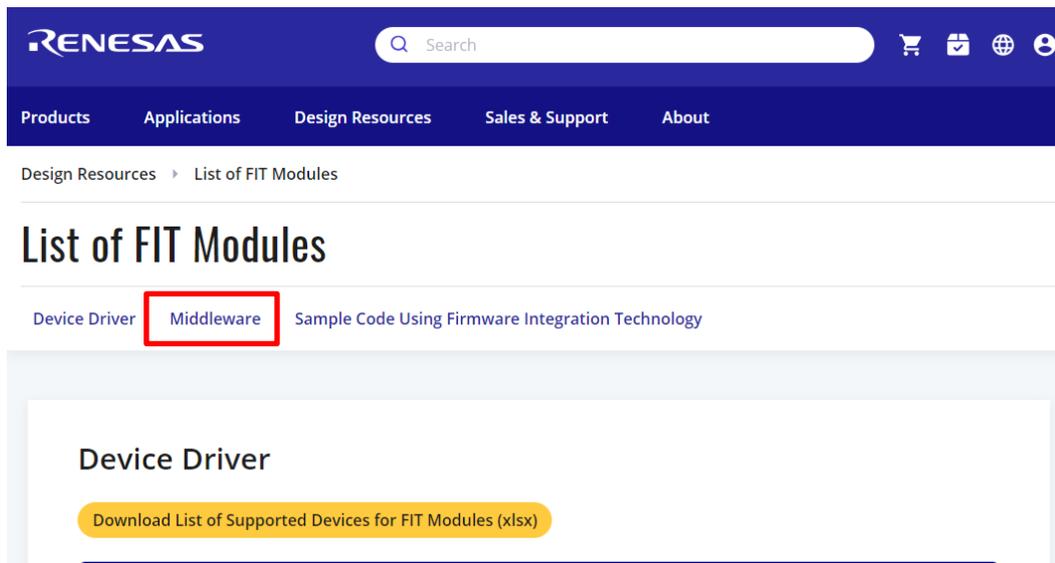
Renesas Image Generator is provided as part of the Firmware Update FIT module.

Note: Version Rev.2.00 and later of the Firmware Update module only support firmware generation using Python scripts.

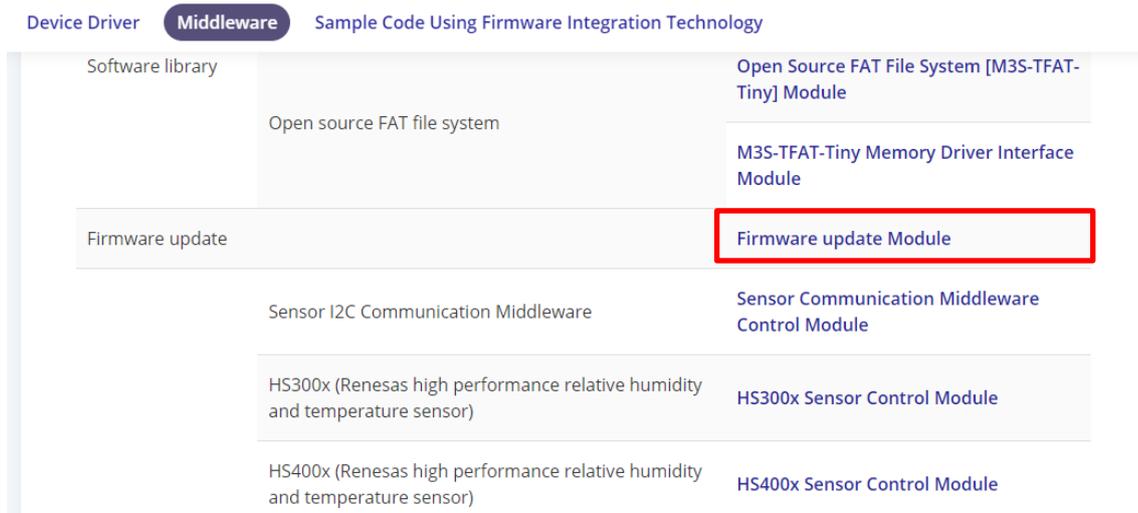
RX Family How to implement FreeRTOS OTA using Amazon Web Services in RX65N (for v202210.01-LTS-rx-1.1.0 or later)

(1) Access the FIT module list page and select middleware

[FIT module list page](#)

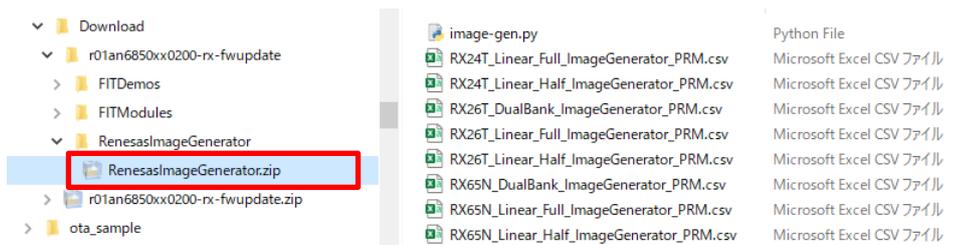


(2) Download the firmware update module



(3) Extract the downloaded firmware update module

Extract the file RenesasImageGenerator.zip in the firmware update module. The RenesasImageGenerator folder contains the Renesas Image Generator script file (image-gen.py) and the parameter files for various devices (*_ImageGenerator_PRM.csv).



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

2.5 Connecting the CK-RX65N



Figure 2.1 Back of RYZ014A PMOD

- (1) Insert the SIM card into the CN6 slot on the RYZ014A PMOD

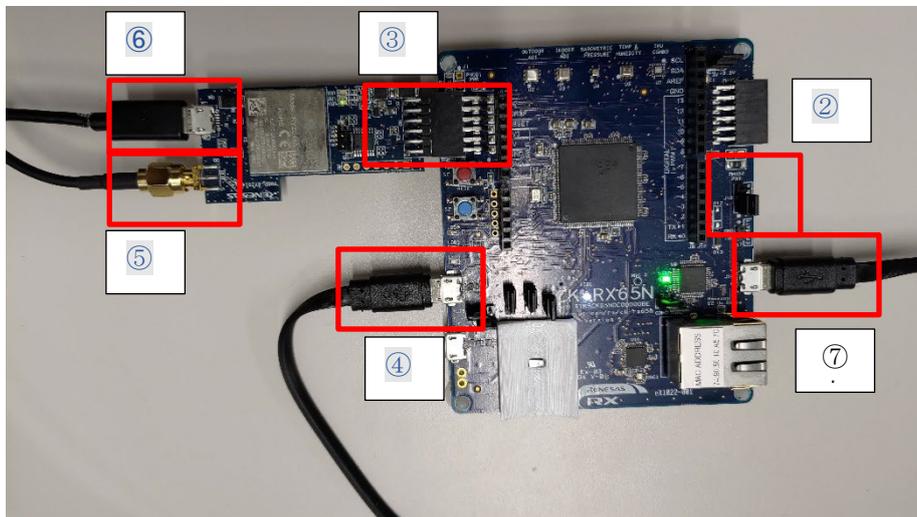


Figure 2.2 Front of base board and RYZ014A PMOD

- (2) On the base board, position the jumper on pins 1-2 of J16 to enable debugging mode
- (3) Connect the RYZ014A PMOD to PMOD1 on the base board
- (4) Connect J20 on the base board to a PC using a USB cable (USB serial connection)
- (5) Connect an antenna to CN3 of the RYZ014A PMOD
- (6) Supply power by connecting a USB cable to CN4 of the RYZ014A PMOD
- (7) Connect J14 on the base board to a PC using a USB cable (debugger connection)

Note: Perform step (6) if you have a spare USB cable available.

If you do not supply power to the RYZ014A PMOD, communication might become unstable.

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

3. Setting Up AWS

To run the FreeRTOS demo, you must have an AWS account (the root user, or an IAM user with permissions to access AWS IoT and FreeRTOS cloud services).

For details on how to sign up for an AWS account and add permissions to users, see

<https://docs.aws.amazon.com/freertos/latest/userguide/freertos-prereqs.html>.

For details on how to set up OTA updates, see <https://docs.aws.amazon.com/freertos/latest/userguide/ota-prereqs.html>.

You must then register the board with AWS IoT by following the instructions in

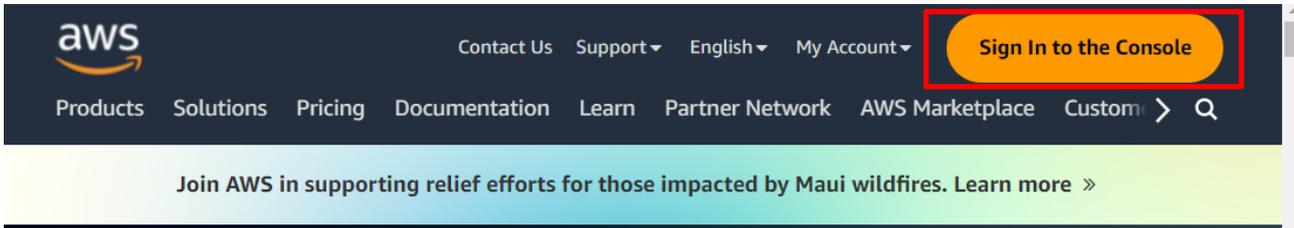
<https://docs.aws.amazon.com/freertos/latest/userguide/freertos-prereqs.html>.

You must also configure the source code as explained in chapter 2 to allow the demo to communicate with AWS.

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

3.1 Signing in to the AWS Console

- (1) Access the AWS web site (<https://aws.amazon.com/>) and click **Sign In to the Console**



- (2) Enter your email address or account ID, and then click **Next**

If the account you are using to sign in is the root user, enter the root user email address. If the account is an IAM user, enter the account ID. (You might skip this step if you have already signed in)



Sign in

Root user

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user

User within an account that performs daily tasks. [Learn more](#)

Root user email address

Next

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

(3) Enter your password and then click **Sign in**

For root users



Root user sign in ⓘ

Email:

Password

[Forgot password?](#)

Sign in

[Sign in to a different account](#)

[Create a new AWS account](#)

For IAM users



Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

Remember this account

Sign in

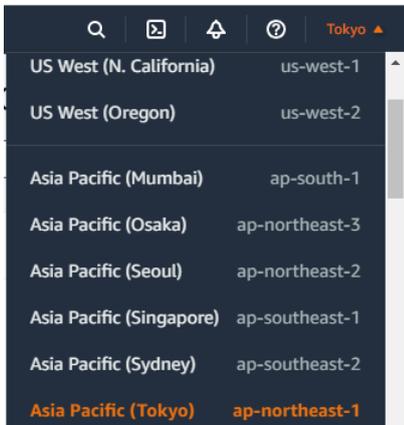
[Sign in using root user email](#)

[Forgot password?](#)

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

3.2 Setting your region in AWS

After logging in to AWS, select your region in the top right of the screen.



3.3 Registering your device in AWS

The following explains the preparations necessary to run the demo project in AWS. Set up AWS by referring to the following tutorial.

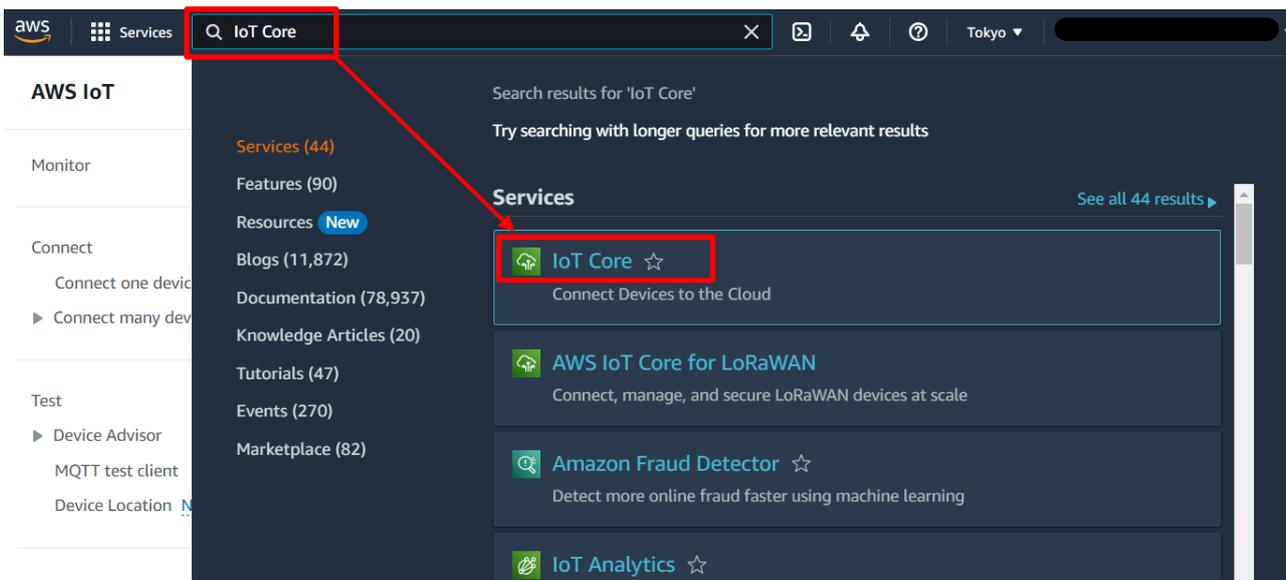
3.3.1 Setting policies

Assign access permissions (policies) for AWS and other resources to the device you want to connect to AWS.

Assign the following policies to the device connected in this application note:

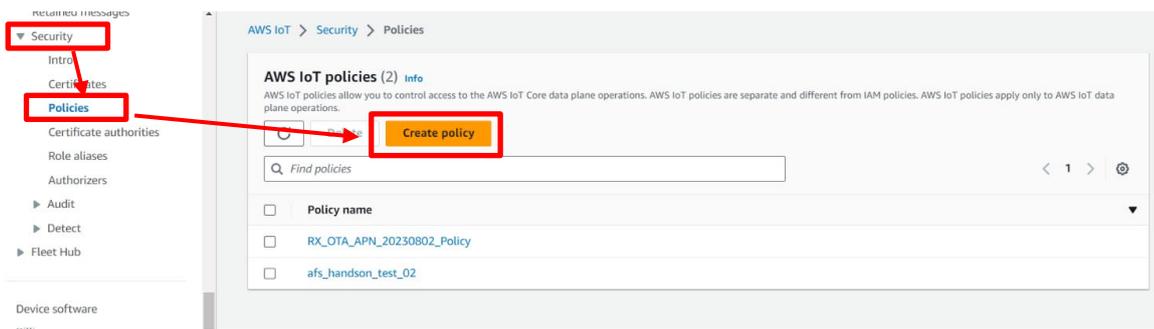
- **iot:Connect:** Connects to AWS IoT
- **iot:Publish:** Publishes a topic
- **iot:Subscribe:** Subscribes to a topic
- **iot:Receive:** Receives messages from AWS IoT

(1) Enter IoT Core in the search box at the top of the screen, and click **IoT Core** in the search results

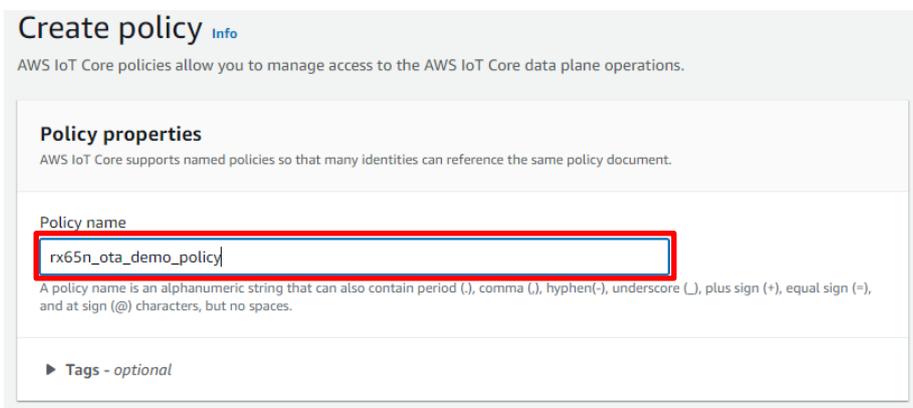


RX Family How to implement FreeRTOS OTA using Amazon Web Services in RX65N (for v202210.01-LTS-rx-1.1.0 or later)

(2) In the menu, click **Security** and then **Policies**, and then click the **Create policy** button

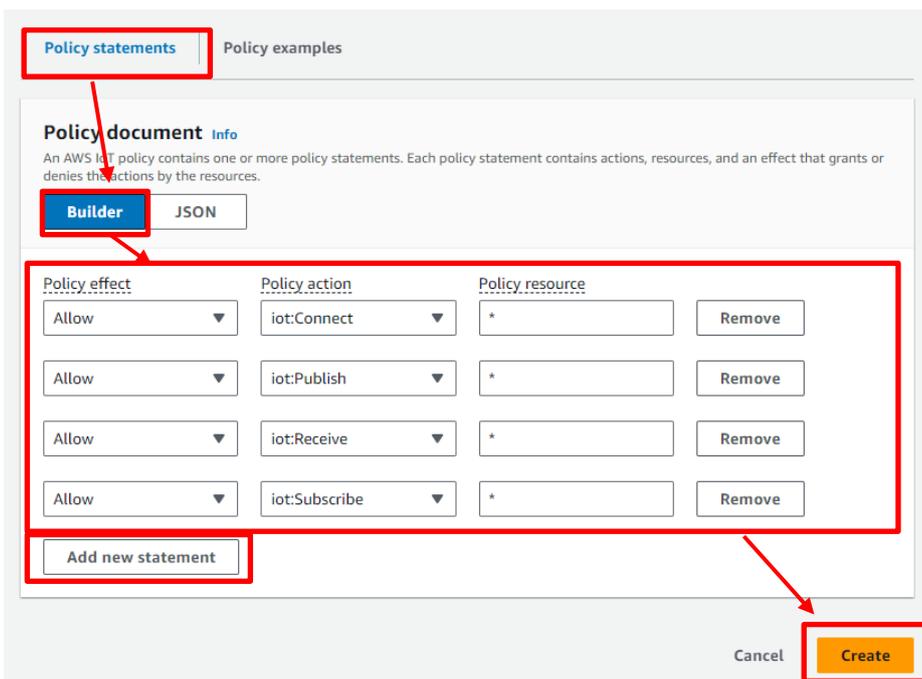


(3) Enter a policy name (for example: rx65n_ota_demo_policy)



(4) Click the **Policy statements** tab, and in the **Policy document** area, click **Builder**. Enter the policy settings as shown in the following figure, and then click **Create**

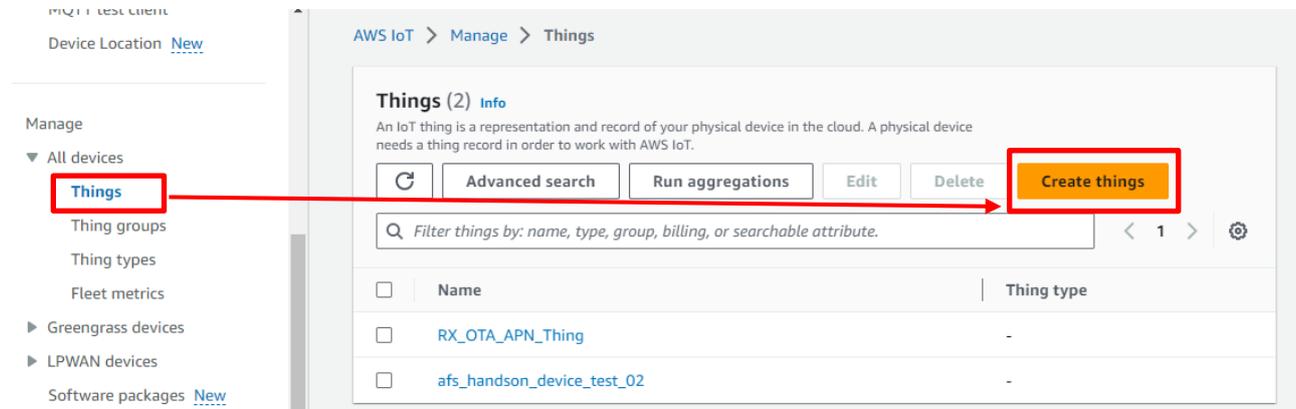
Because the policy initially contains only one statement, you must add more statements by clicking the **Add new statement** button.



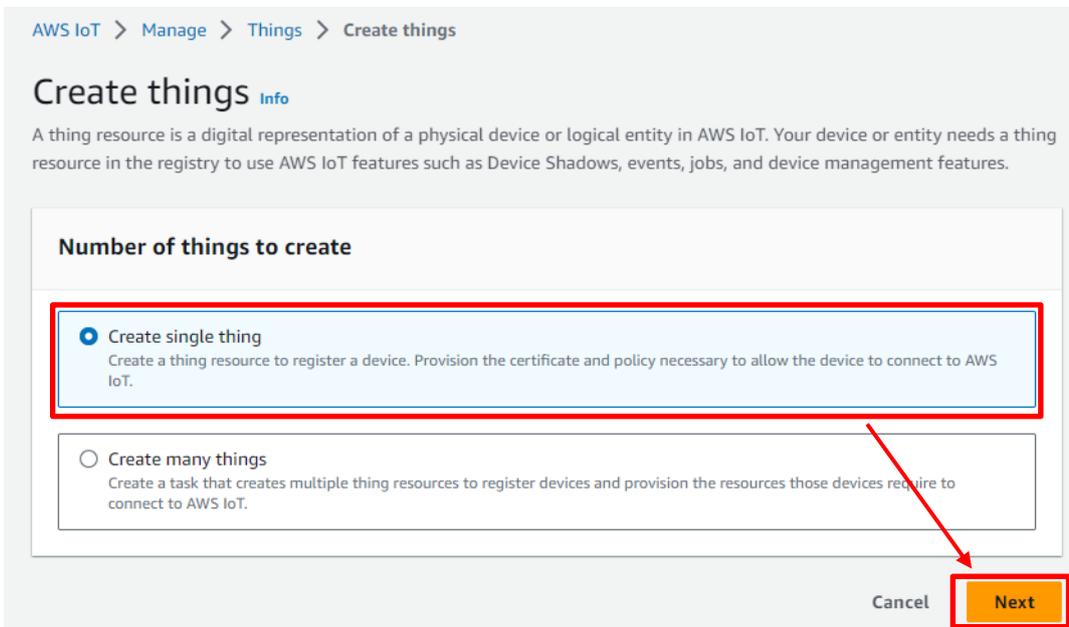
RX Family How to implement FreeRTOS OTA using Amazon Web Services in RX65N (for v202210.01-LTS-rx-1.1.0 or later)

3.3.2 Registering your device as a *thing* in AWS IoT

(1) In the menu, click **Manage**, **All devices**, and **Things**, and then click the **Create things** button



(2) Select **Create single thing** and then click **Next**



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

(3) Enter a thing name (example: rx65n_ota_demo_thing), and then click **Next**

Make a note of the thing name you entered. You will need it in a later process.

AWS IoT > Manage > Things > Create things > Create single thing

Step 1
Specify thing properties

Step 2 - optional
Configure device certificate

Step 3 - optional
Attach policies to certificate

Specify thing properties [Info](#)

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

Thing properties [Info](#)

Thing name

rx65n_ota_demo_thing

Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

Additional configurations

You can use these configurations to add detail that can help you to organize, manage, and search your things.

Device Shadow [Info](#)

Device Shadows allow connected devices to sync states with AWS. You can also get, update, or delete the state information of this thing's shadow using either HTTPs or MQTT topics.

- No shadow
- Named shadow
Create multiple shadows with different names to manage access to properties, and logically group your devices properties.
- Unnamed shadow (classic)
A thing can have only one unnamed shadow.

Cancel **Next**

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

(4) In the Device certificate area, select **Auto-generate a new certificate** and then click **Next**

Step 1
Specify thing properties

Step 2 - optional
Configure device certificate

Step 3 - optional
Attach policies to certificate

Configure device certificate - optional [Info](#)

A device requires a certificate to connect to AWS IoT. You can choose how to register a certificate for your device now, or you can create and register a certificate for your device later. Your device won't be able to connect to AWS IoT until it has an active certificate with an appropriate policy.

Device certificate

- Auto-generate a new certificate (recommended)**
Generate a certificate, public key, and private key using AWS IoT's certificate authority.
- Use my certificate
Use a certificate signed by your own certificate authority.
- Upload CSR
Register your CA and use your own certificates on one or many devices.
- Skip creating a certificate at this time
You can create a certificate for this thing and attach a policy to the certificate at a later time.

Cancel Previous **Next**

(5) Attach the policy to the certificate

Select the policy you created in 3.3.1 Setting policies, and then click the **Create thing** button

Step 1
Specify thing properties

Step 2 - optional
Configure device certificate

Step 3 - optional
Attach policies to certificate

Attach policies to certificate - optional [Info](#)

AWS IoT policies grant or deny access to AWS IoT resources. Attaching policies to the device certificate applies this access to the device.

Policies (1/3)

Select up to 10 policies to attach to this certificate.

Filter policies

Name
<input checked="" type="checkbox"/> rx65n_ota_demo_policy
<input type="checkbox"/> af[REDACTED]
<input type="checkbox"/> RX[REDACTED]

Cancel Previous **Create thing**

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

(6) Download the certificate and key files

The certificate and private key are equivalent to a password for the device (thing). When you register a certificate and private key on a device, the device can use this certificate and private key to connect to AWS.

You must download the certificate, public key, and private key now. You will not have another opportunity to download them.

Download certificates and keys

Download certificate and key files to install on your device so that it can connect to AWS.

Device certificate

You can activate the certificate now, or later. The certificate must be active for a device to connect to AWS IoT.

Device certificate
[redacted]te.pem.crt

[Deactivate certificate](#) [Download](#)

Key files

The key files are unique to this certificate and can't be downloaded after you leave this page. Download them now and save them in a secure place.

⚠ This is the only time you can download the key files for this certificate.

Public key file
[redacted]-public.pem.key [Download](#)

Private key file
[redacted]-private.pem.key [Download](#)

Root CA certificates

Download the root CA certificate file that corresponds to the type of data endpoint and cipher suite you're using. You can also download the root CA certificates later.

Amazon trust services endpoint
RSA 2048 bit key: Amazon Root CA 1 [Download](#)

Amazon trust services endpoint
ECC 256 bit key: Amazon Root CA 3 [Download](#)

If you don't see the root CA certificate that you need here, AWS IoT supports additional root CA certificates. These root CA certificates and others are available in our developer

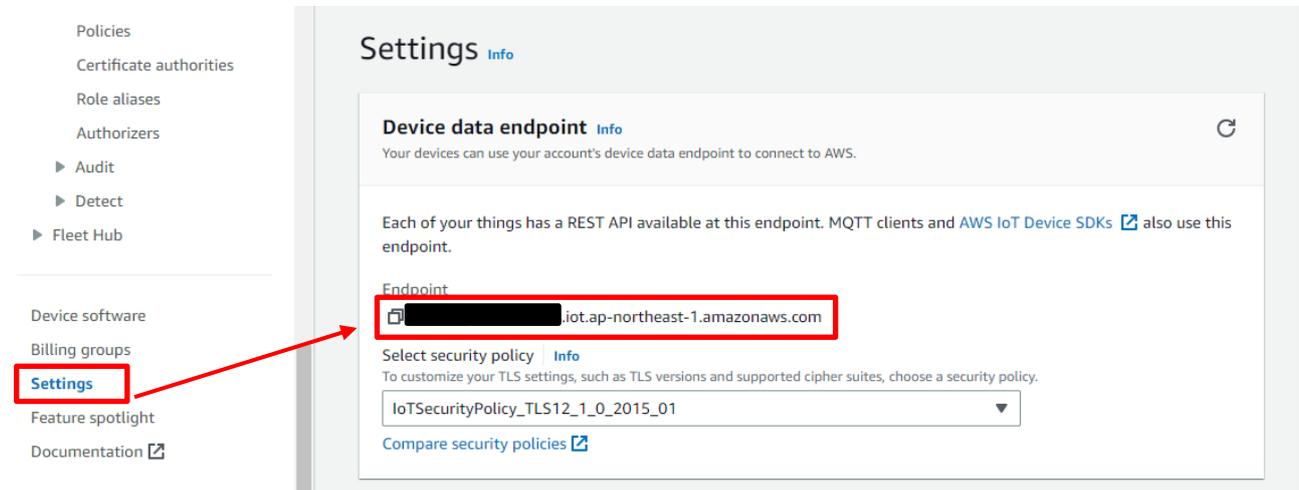
[Done](#)

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

3.3.3 Checking the endpoint

The endpoint is equivalent to a connection destination (URL) for the device (thing). The device will connect to the endpoint registered for the device.

- (1) In the menu, click **Settings** and make a note of the endpoint.

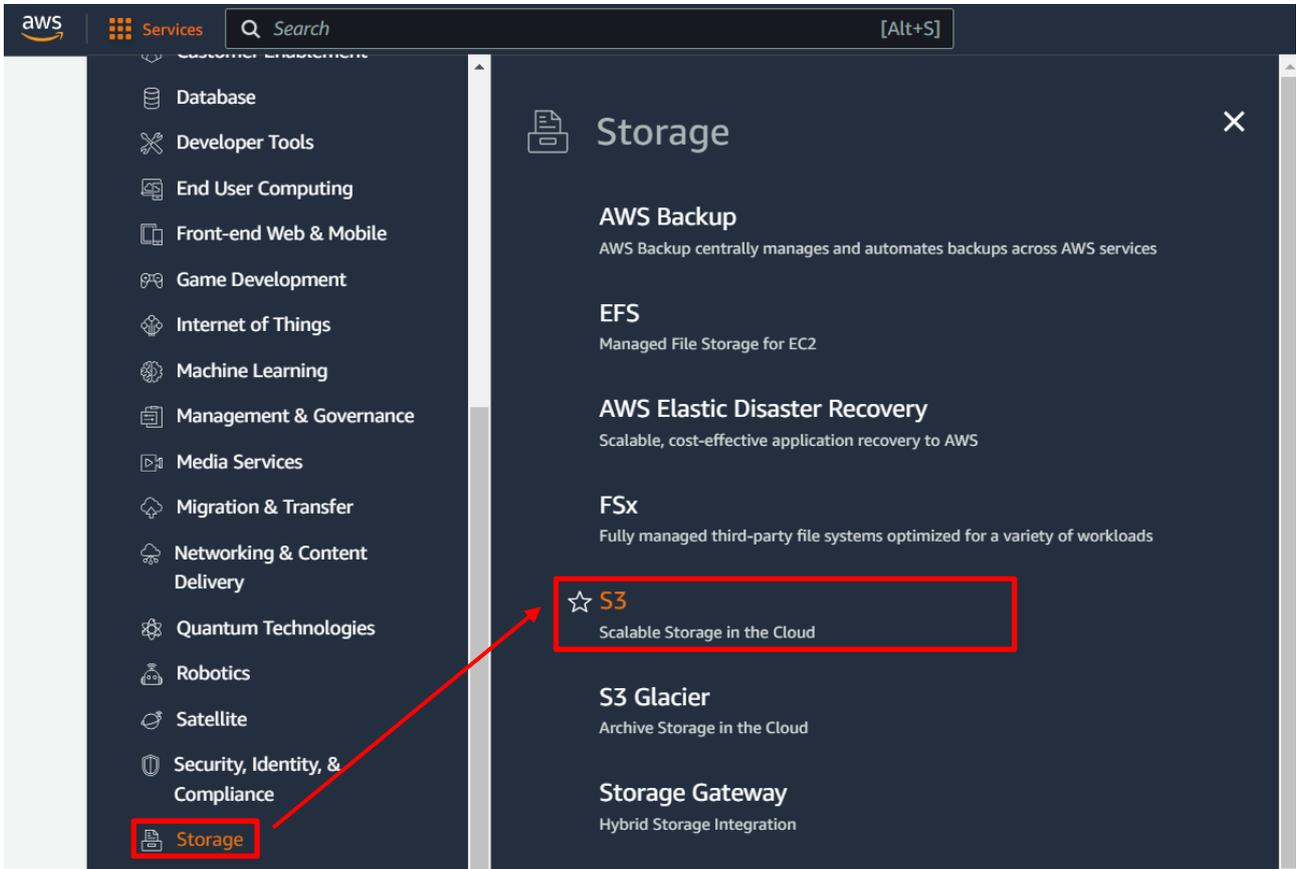


RX Family How to implement FreeRTOS OTA using Amazon Web Services in RX65N (for v202210.01-LTS-rx-1.1.0 or later)

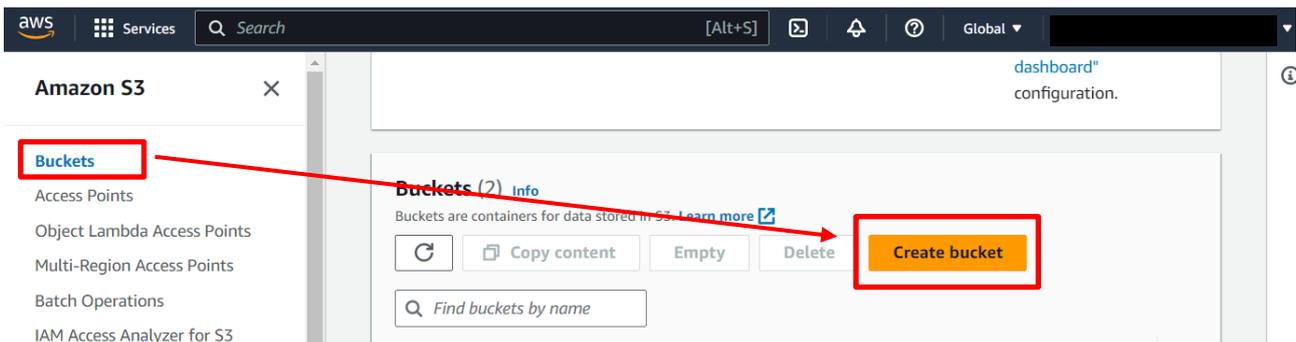
3.4 Creating an Amazon S3 bucket

Amazon S3 is an online storage web service used to store the firmware with which the device will be updated.

(1) From the **Services** menu, select **Storage** and then **S3**



(2) On the **Buckets** page, click the **Create bucket** button



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

(3) Enter a bucket name (example: s3test-rx65n)

Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

AWS Region

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

The bucket name must be globally unique. The following error message appears if the bucket name is already in use. In this case, use another name.

Bucket name

 Bucket with the same name already exists

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

(4) Create the bucket

Enter the settings as follows, and then click the **Create bucket** button

- **Block Public Access setting for this bucket: Block all public access**
- **Bucket Versioning: Enable**

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through *new* access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through *any* access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through *new* public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Disable

Enable

► **Advanced settings**

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

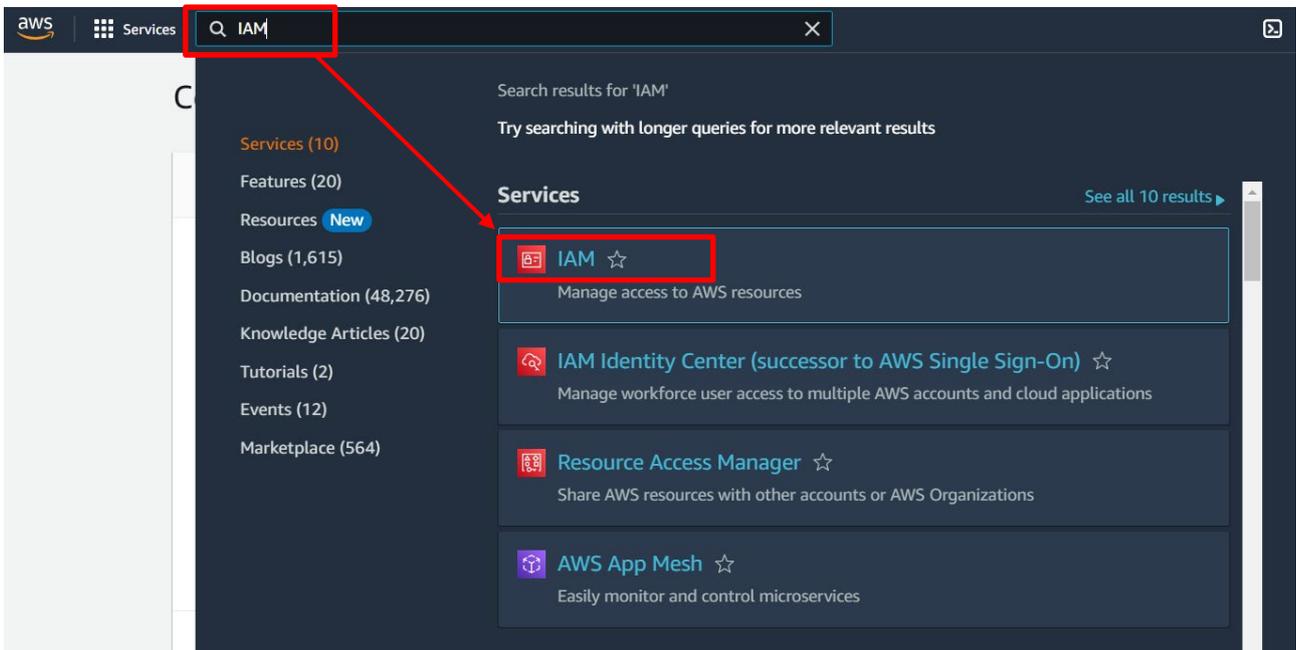
Cancel **Create bucket**

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

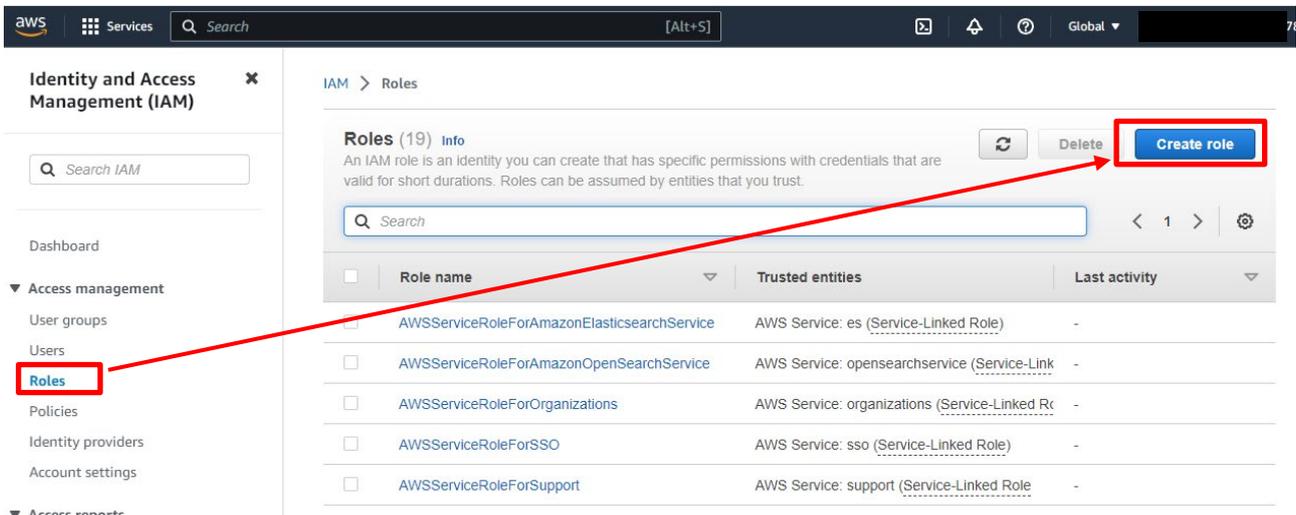
3.5 Allocating OTA execution permission to IAM users

Create a role with the appropriate access permissions to create OTA update jobs.

(1) Enter IAM in the search box at the top of the screen, and click **IAM** in the search results



(2) In the menu, click **Roles** and then click the **Create role** button



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

(3) Under **Select trusted entity**, enter the following settings and then click **Next**:

- Under **Trusted entity type**, select **AWS service**
- Under **Use cases for other AWS services**, select **IoT**
- Select the **IoT** option button

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity [Info](#)

Trusted entity type

AWS service

Allow AWS services like EC2, Lambda, or others to perform actions in this account.

AWS account

Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

Web identity

Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

SAML 2.0 federation

Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

Custom trust policy

Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Search:

- IoT**
Allows IoT to call AWS services on your behalf.
- IoT - Device Defender Audit**
Provides AWS IoT Device Defender read access to IoT and related resources.
- IoT - Device Defender Mitigation Actions**
Provides AWS IoT Device Defender write access to IoT and related resources for execution of Mitigation Actions.

Cancel

Next

(4) Click **Next** on the **Add permissions** page without making any changes

Add permissions [Info](#)

Permissions policies (3) [Info](#)

The type of role that you selected requires the following policy.

Policy name ↗	Type	Attached entities
<input type="checkbox"/> AWSIoTRuleActions	AWS m...	1
<input type="checkbox"/> AWSIoTLogging	AWS m...	1
<input type="checkbox"/> AWSIoTThingsRegi...	AWS m...	1

▶ Set permissions boundary - optional [Info](#)

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel

Previous

Next

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

(5) Enter a role name (example: ota_role_rx65n), and then click the **Create role** button
Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

ota_role_rx65n

Maximum 64 characters. Use alphanumeric and '+-,@_-' characters.

Description
Add a short explanation for this role.

Allows IoT to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+-,@_-' characters.

Add tags - optional [Info](#)
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

[Cancel](#) [Previous](#) [Create role](#)

(6) Click the role you created

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings

Access reports

- Access analyzer
- Archive rules
- Analyzers
- Settings
- Credential report

IAM > Roles

Roles (20) [Info](#)

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

<input type="checkbox"/>	Role name	Trusted entities
<input type="checkbox"/>	AWSServiceRoleForAmazonElasticsearchService	AWS Service: es (Service)
<input type="checkbox"/>	AWSServiceRoleForAmazonOpenSearchService	AWS Service: opensearch
<input type="checkbox"/>	AWSServiceRoleForOrganizations	AWS Service: organization
<input type="checkbox"/>	AWSServiceRoleForSSO	AWS Service: sso (Service)
<input type="checkbox"/>	AWSServiceRoleForSupport	AWS Service: support (Service)
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvis
<input type="checkbox"/>	[Redacted]	AWS Service: iot
<input type="checkbox"/>	[Redacted]	AWS Service: iot
<input type="checkbox"/>	ota_role_rx65n	AWS Service: iot
<input type="checkbox"/>	[Redacted]	AWS Service: iot

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

(7) Select **Attach policies**

Permissions Trust relationships Tags Access Advisor Revoke sessions

Permissions policies (3) Info
You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter.

Refresh Simulate Remove **Add permissions** ▲
Attach policies
Create inline policy

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	AWSIoTRuleActions	AWS managed	Allows access to all AWS services supported in AWS IoT Rule Actions
<input type="checkbox"/>	AWSIoTLogging	AWS managed	Allows creation of Amazon CloudWatch Log groups and streaming logs to the groups
<input type="checkbox"/>	AWSIoTThingsRegistrati...	AWS managed	This policy allows users to register things at bulk using AWS IoT StartThingRegistrati...

(8) Enter AmazonFreeRTOSOTAUpdate in the **Permissions policies** search box, and then press the **Enter** key

Permissions policies (6) Info
You can attach up to 10 managed policies.

Refresh Simulate Remove Add permissions ▼

Search: AmazonFreeRTOSOTAUpdate

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	AWSIoTRuleActions	AWS managed	Allows access to all AWS services supported in AWS IoT Rule Actions
<input type="checkbox"/>	AWSIoTLogging	AWS managed	Allows creation of Amazon CloudWatch Log groups and streaming logs to the groups
<input type="checkbox"/>	AWSIoTThingsRegistration	AWS managed	This policy allows users to register things at bulk using AWS IoT StartThingRegistrationTask API

(9) Select the check box beside the AmazonFreeRTOSOTAUpdate policy, and then click the **Add permissions** button

Attach policy to ota_role_rx65n

▶ Current permissions policies (3)

Other permissions policies (Selected 1/881)

Refresh Create policy

Filter policies by property or policy name and press enter. 1 match

"AmazonFreeRTOSOTAUpdate" Clear filters

<input checked="" type="checkbox"/>	Policy name	Type	Description
<input checked="" type="checkbox"/>	AmazonFreeRTOSOTAUpdate	AWS managed	Allows user to access Amazon FreeRTOS OTA Update

Cancel **Add permissions**

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

(10) From the **Add permissions** drop-down list, select **Create inline policy**

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Permissions policies (4) [Info](#)
You can attach up to 10 managed policies.

<input type="checkbox"/>	Policy name ↗	Type	Description
<input type="checkbox"/>	 AWSIoTRuleActions	AWS managed	Allows access to all AWS services supported in AWS IoT Rule Actions
<input type="checkbox"/>	 AWSIoTLogging	AWS managed	Allows creation of Amazon CloudWatch Log groups and streaming logs to the gro...
<input type="checkbox"/>	 AWSIoTThingsRegistration	AWS managed	This policy allows users to register things at bulk using AWS IoT StartThingRegist...
<input type="checkbox"/>	 AmazonFreeRTOSOTAUpd...	AWS managed	Allows user to access Amazon FreeRTOS OTA Update

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

(11) Click **JSON**, paste the following code, and then click **Next**

This code grants permission to pass the IAM role to AWS services.

Code to paste:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "*"
    }
  ]
}
```

Specify permissions [Info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

The screenshot shows the AWS IAM console's 'Specify permissions' page. At the top, there are three tabs: 'Visual', 'JSON', and 'Actions'. The 'JSON' tab is selected and highlighted with a red box. Below the tabs is a large text area for the JSON policy, which is also highlighted with a red box. The JSON code is:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "*"
    }
  ]
}
```

 To the right of the JSON editor is the 'Edit statement' section, which contains a 'Select a statement' prompt and an 'Add new statement' button. At the bottom right of the page, there are 'Cancel' and 'Next' buttons. The 'Next' button is highlighted with a red box. A red arrow points from the 'JSON' tab to the 'Next' button.

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

(12) Enter a policy name (example: rx65n_ota_demo_iam_policy), and then click the **Create policy** button

Review and create

Review the permissions, specify details, and tags.

Policy details

Policy name

Enter a meaningful name to identify this policy.

rx65n_ota_demo_iam_policy

Maximum 128 characters. Use alphanumeric and '+', '@', '-' characters.

Permissions defined in this policy [Info](#)

Permissions in the policy document specify which actions are allowed or denied.

Search

Allow (1 of 384 services)

Show remaining 383 services

Service

Access level

Resource

Request condition

IAM

Limited: Read, Write

All resources

None

Cancel

Previous

Create policy

(13) Again, from the **Add permissions** drop-down list, select **Create inline policy**

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

Permissions policies (4) [Info](#)

You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter.

Add permissions

Attach policies

Create inline policy

<input type="checkbox"/>	Policy name ↗	Type	Description
<input type="checkbox"/>	AWSIoTRuleActions	AWS managed	Allows access to all AWS services supported in AWS IoT Rule Actions
<input type="checkbox"/>	AWSIoTLogging	AWS managed	Allows creation of Amazon CloudWatch Log groups and streaming logs to the gro...
<input type="checkbox"/>	AWSIoTThingsRegistration	AWS managed	This policy allows users to register things at bulk using AWS IoT StartThingRegistr...
<input type="checkbox"/>	AmazonFreeRTOSOTAUpd...	AWS managed	Allows user to access Amazon FreeRTOS OTA Update

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

(14) Click **JSON**, paste the following code, and then click **Next**

This code allows access to Amazon S3 where the updated firmware is stored.

Code to paste:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObjectVersion",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Specify permissions [Info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

Visual **JSON** Actions ▾

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

[+ Add new statement](#)

JSON Ln 17, Col 0 999 of 10128 characters remaining

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

Cancel **Next**

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

(15) Enter a policy name (example: rx65n_ota_demo_s3_policy), and then click the **Create policy** button

Review and create

Review the permissions, specify details, and tags.

Policy details

Policy name

Enter a meaningful name to identify this policy.

rx65n_ota_demo_s3_policy

Maximum 128 characters. Use alphanumeric and '+', '@', '-' characters.

Permissions defined in this policy [Info](#)

Edit

Permissions in the policy document specify which actions are allowed or denied.

Q Search

Allow (1 of 384 services)

Show remaining 383 services

Service	Access level	Resource	Request condition
S3	Limited: Read, Write	All resources	None

Cancel

Previous

Create policy

4. Setting Up the Device

4.1 Generating key pairs and certificates

- (1) From the **Start** menu, open the **Win64 OpenSSL Command Prompt**



- (2) Execute the command to create a CA private key using ECDSA

Execute the following command:

```
openssl ecparam -genkey -name secp256r1 -out ca.key
```

Execution results:

```
C:\%openssl>openssl ecparam -genkey -name secp256r1 -out ca.key
using curve name prime256v1 instead of secp256r1
```

- (3) Execute the command to create a CA certificate from the CA private key you created

Execute the following command: You can enter any character string for **Country Name** onward.

```
openssl req -x509 -sha256 -new -nodes -key ca.key -days 3650 -out ca.crt
```

```
C:\%openssl>openssl req -x509 -sha256 -new -nodes -key ca.key -days 3650 -out ca.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:JP
State or Province Name (full name) [Some-State]:Tokyo
Locality Name (eg, city) []:Kodaira
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Renesas Electronics
Organizational Unit Name (eg, section) []:Software Development Division
Common Name (e.g. server FQDN or YOUR name) []:Renesas Tarou
Email Address []:Tarou.Renesas@sample.com
```

Enter any character string for these attributes

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

- (4) Execute the command to create an ECDSA key pair

Execute the following command:

```
openssl ecparam -genkey -name secp256r1 -out secp256r1.keypair
```

```
C:\>openssl>openssl ecparam -genkey -name secp256r1 -out secp256r1.keypair
using curve name prime256v1 instead of secp256r1
```

- (5) Execute the command to create a certificate signing request from the ECDSA key pair you created

Execute the following command: You can enter any character string for **Country Name** onward. For the last two lines, press **Enter** without entering anything.

```
openssl req -new -sha256 -key secp256r1.keypair > secp256r1.csr
```

```
C:\>openssl>openssl req -new -sha256 -key secp256r1.keypair > secp256r1.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
```

```
What you are about to enter is what is called a Distinguished Name or a DN.
```

```
There are quite a few fields but you can leave some blank
```

```
For some fields there will be a default value,
```

```
If you enter '.', the field will be left blank.
```

```
-----
```

```
Country Name (2 letter code) [AU]:JP
State or Province Name (full name) [Some-State]:Tokyo
Locality Name (eg, city) []:Kodaira
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Renesas Electronics
Organizational Unit Name (eg, section) []:Software Development Division
Common Name (e.g. server FQDN or YOUR name) []:Renesas Tarou
Email Address []:Tarou.Renesas@sample.com
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
```

```
A challenge password []:
```

```
An optional company name []:
```

Enter any character string for these attributes

Press **Enter** without entering anything

- (6) Execute the command to create a certificate from the certificate signing request, CA certificate, and CA private key you created

Execute the following command:

```
openssl x509 -req -sha256 -days 3650 -in secp256r1.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out secp256r1.crt
```

```
C:\>openssl>openssl x509 -req -sha256 -days 3650 -in secp256r1.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out secp256r1.crt
Signature ok
subject=C = JP, ST = Tokyo, L = Kodaira, O = Renesas Electronics, OU = Software Development Division, CN = Renesas Tarou, emailAddress = Tarou.Renesas@sample.com
Getting CA Private Key
```

- (7) Execute the command to extract the private key from the ECDSA key pair

Execute the following command:

```
openssl ec -in secp256r1.keypair -outform PEM -out secp256r1.privatekey
```

```
C:\>openssl>openssl ec -in secp256r1.keypair -outform PEM -out secp256r1.privatekey
read EC key
writing EC key
```

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

- (8) Execute the command to extract the public key from the ECDSA key pair

Execute the following command:

```
openssl ec -in secp256r1.keypair -outform PEM -pubout -out secp256r1.publickey
```

```
C:\¥openssl>openssl ec -in secp256r1.keypair -outform PEM -pubout -out secp256r1.publickey
read EC key
writing EC key
```

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

4.2 Creating the initial version of the firmware

The following explains how to create the initial version of the firmware.

4.2.1 Importing the project

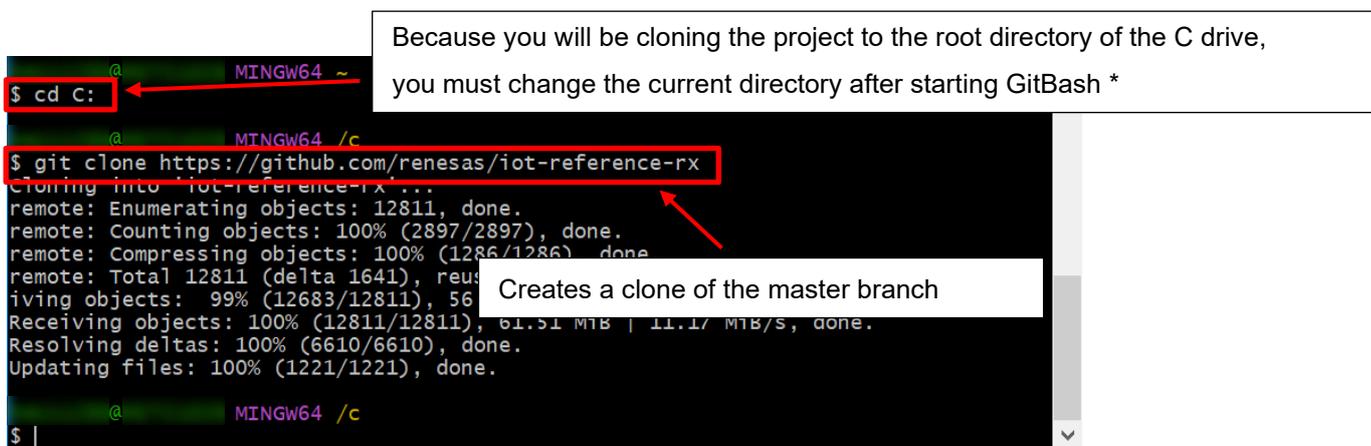
(1) Clone the demo project

Clone the demo project from GitHub ([iot-reference-rx: FreeRTOS reference repository](#)). This document assumes the reader is using [Git for Windows](#) when explaining the cloning process.

Open GitBash and execute the following commands:

```
cd c:
```

```
git clone https://github.com/renesas/iot-reference-rx
```

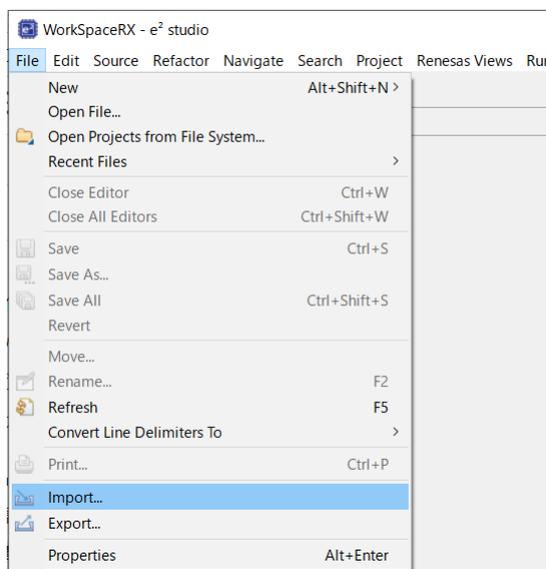


Note: Due to restrictions in e² studio, the length of the path of the cloning destination (including any folder names) must not exceed 35 characters. If you specify a path with 36 or more characters, an error occurs when building the project.

In the preceding example, the project is cloned to the root directory of the C drive.

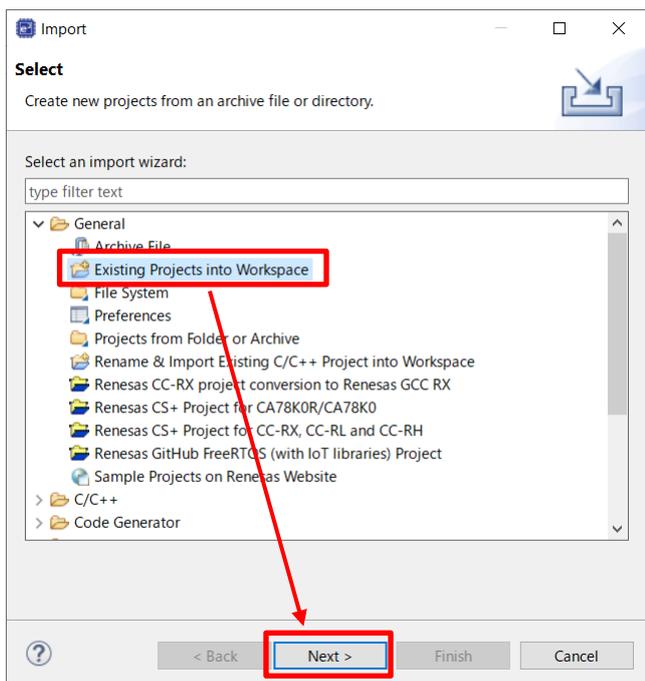
(2) Start e² studio

(3) From the **File** menu, select **Import**



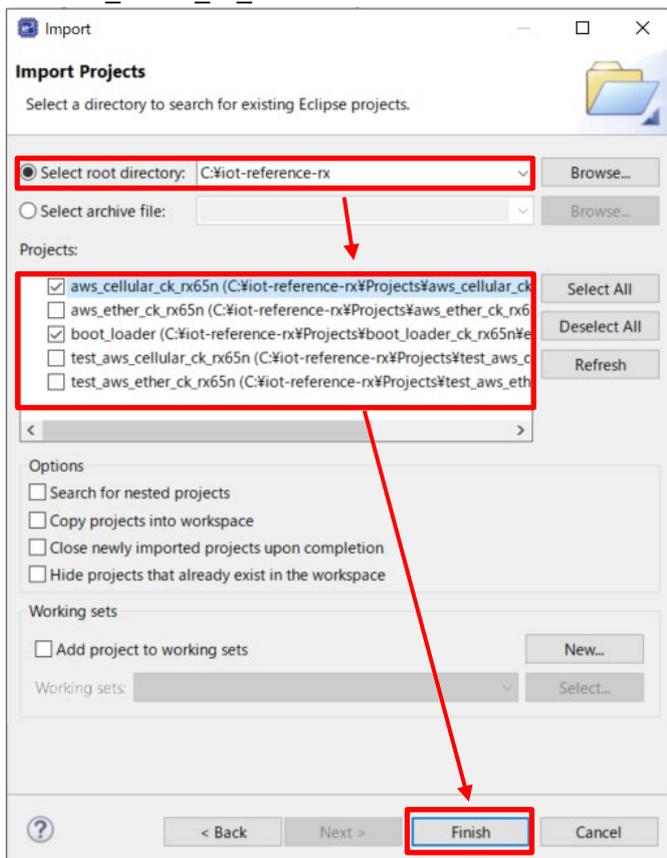
RX Family How to implement FreeRTOS OTA using Amazon Web Services in RX65N (for v202210.01-LTS-rx-1.1.0 or later)

(4) Select Existing Projects into Workspace



(5) In **Select root directory**, select the folder you cloned in 4.2.1(1), select the check boxes for the following projects, and then click **Finish**

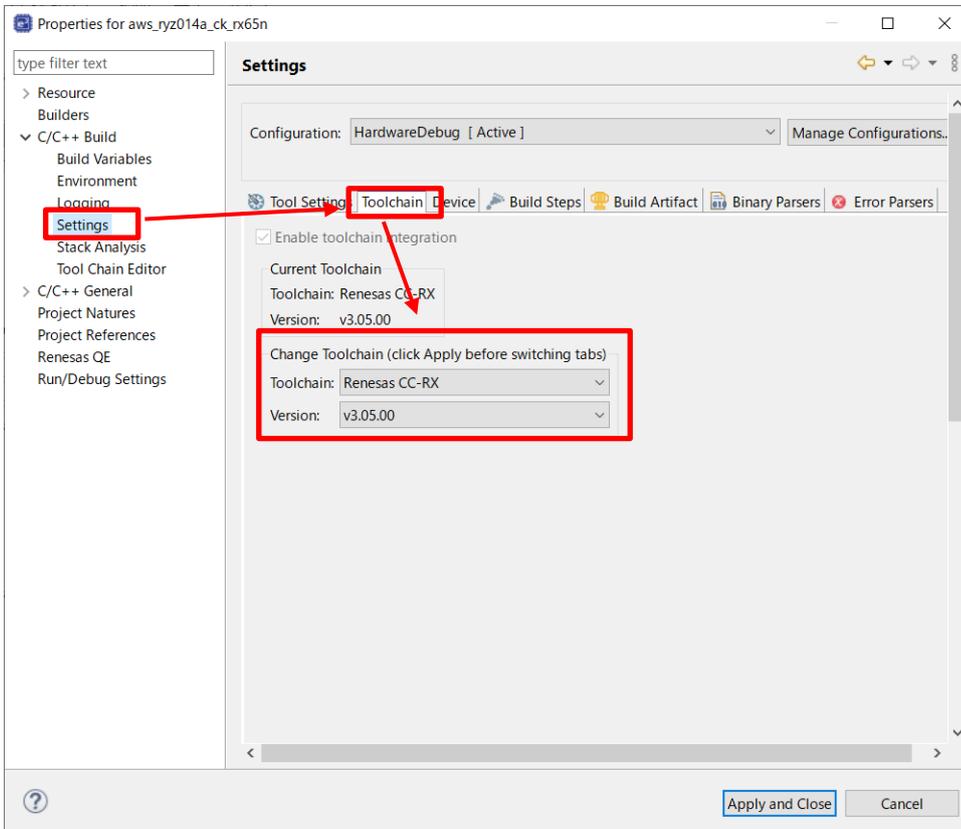
- aws_ryz014a_ck_rx65n
- boot_loader_ck_rx65n



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

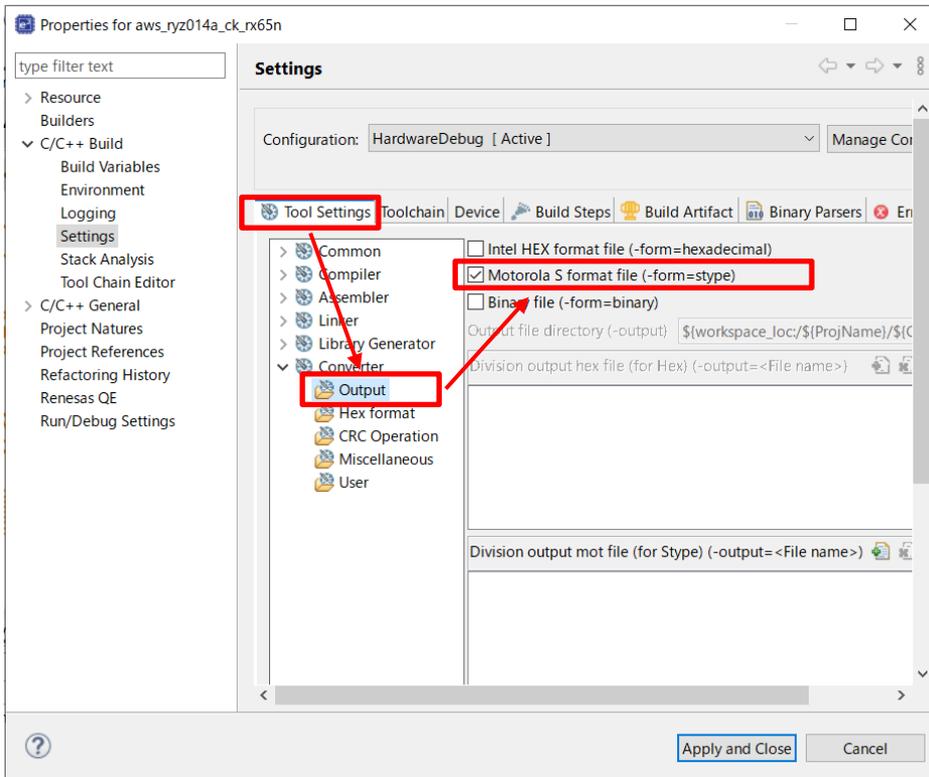
4.2.2 Checking the project environment settings

- (1) For both projects, from the **Projects** menu, select **Properties**, expand the **C/C++ Build** menu, and click **Settings**. On the **Toolchain** tab, confirm that the toolchain is **Renesas CC-RX**



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

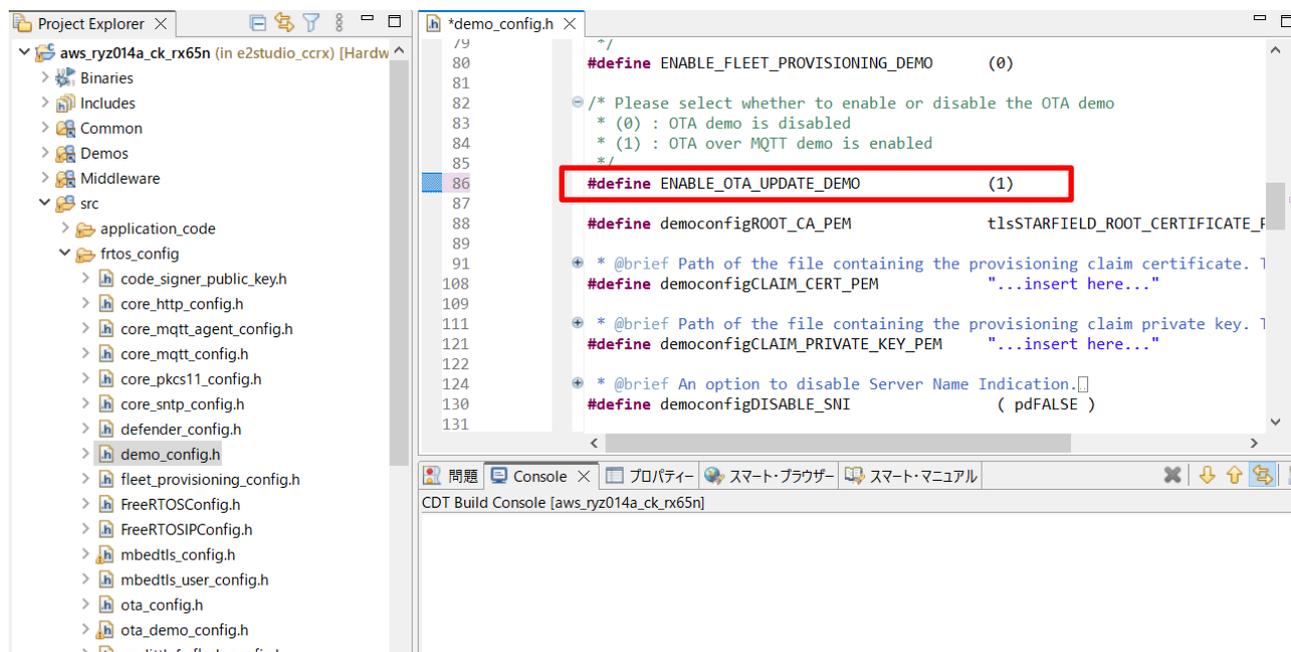
- (2) On the **Tool Settings** tab, expand the **Converter** menu and select **Output**. Confirm that the **Motorola S format file** check box is selected



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

(2) Set the definition for the OTA update demo to Enable

Set `ENABLE_OTA_UPDATE_DEMO` to 1 (Enable) in `aws_ryz014a_ck_rx65n\src\frtos_conf\demo_config.h`. (The default is 0)



The screenshot shows an IDE window with the Project Explorer on the left and the demo_config.h file open in the main editor. The Project Explorer shows the project structure for 'aws_ryz014a_ck_rx65n'. The main editor displays the following code:

```
80 #define ENABLE_FLEET_PROVISIONING_DEMO (0)
81
82 /* Please select whether to enable or disable the OTA demo
83 * (0) : OTA demo is disabled
84 * (1) : OTA over MQTT demo is enabled
85 */
86 #define ENABLE_OTA_UPDATE_DEMO (1)
87
88 #define democonfigROOT_CA_PEM t1sSTARFIELD_ROOT_CERTIFICATE_F
89
90 * @brief Path of the file containing the provisioning claim certificate.
91 #define democonfigCLAIM_CERT_PEM "...insert here..."
92
93 * @brief Path of the file containing the provisioning claim private key.
94 #define democonfigCLAIM_PRIVATE_KEY_PEM "...insert here..."
95
96 * @brief An option to disable Server Name Indication.
97 #define democonfigDISABLE_SNI ( pdFALSE )
```

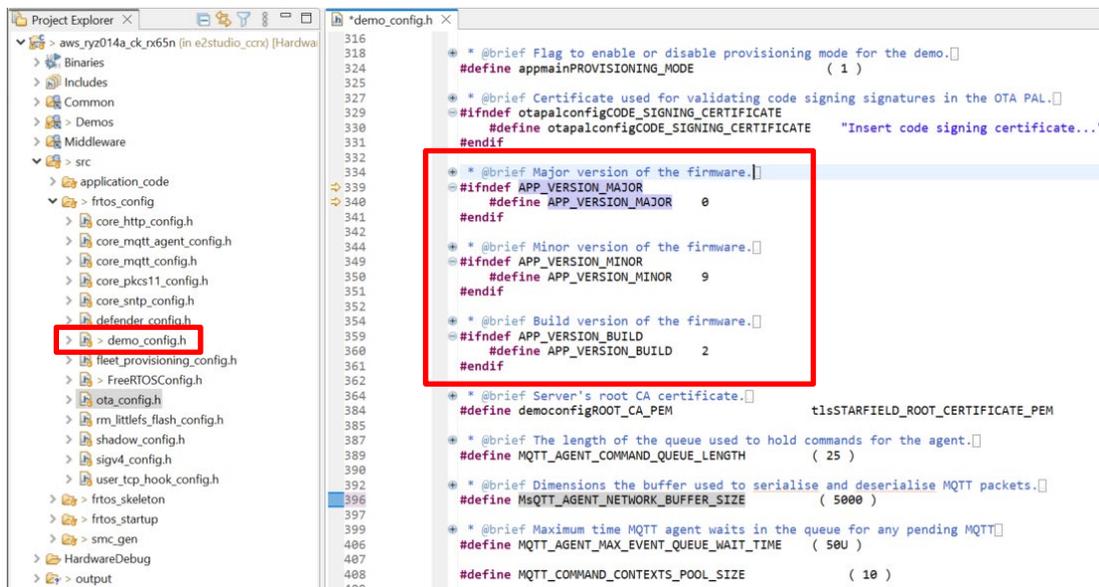
The line `#define ENABLE_OTA_UPDATE_DEMO (1)` is highlighted with a red box. The bottom status bar shows 'CDT Build Console [aws_ryz014a_ck_rx65n]'.

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

(3) Confirm that the initial project version is 0.92

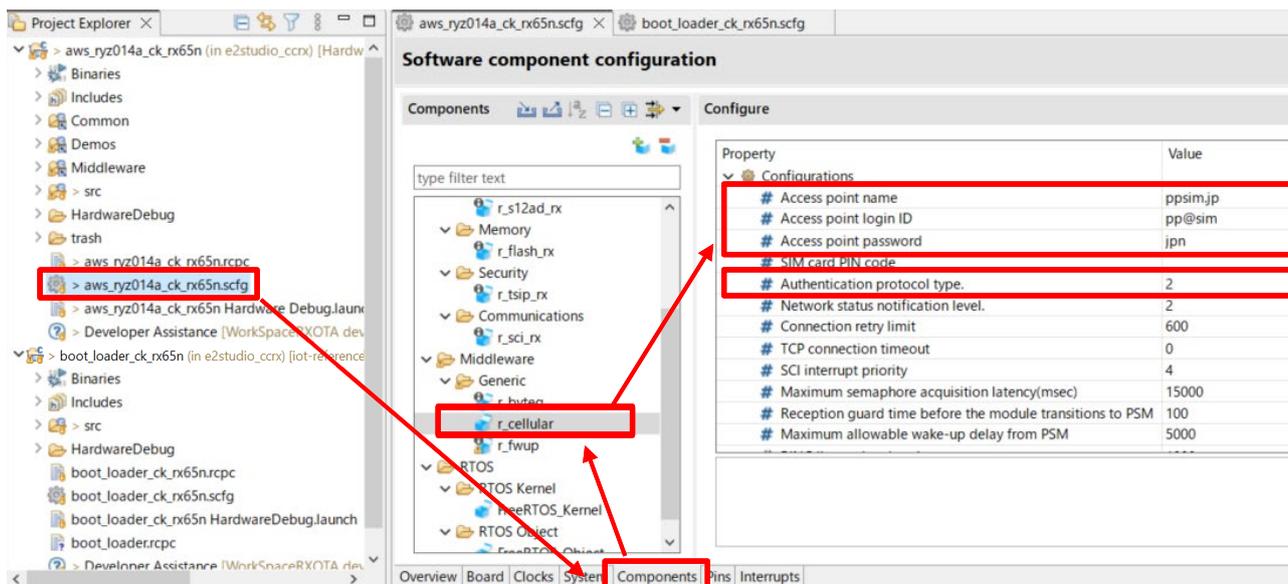
Confirm that the version definitions in `aws_ryz014a_ck_rx65n\src\lrtps_config\demo_config.h` are as follows:

- APP_VERSION_MAJOR 0
- APP_VERSION_MINOR 9
- APP_VERSION_BUILD 2



(4) Configure the RYZ014A Cellular Module Control FIT module (r_cellular)

Open the file `aws_ryz014a_ck_rx65n.scfg`, and click the **Components** tab. For the `r_cellular` component, set **Access point name**, **Access point login ID**, **Access point password**, and **Authentication protocol type** according to your SIM card.



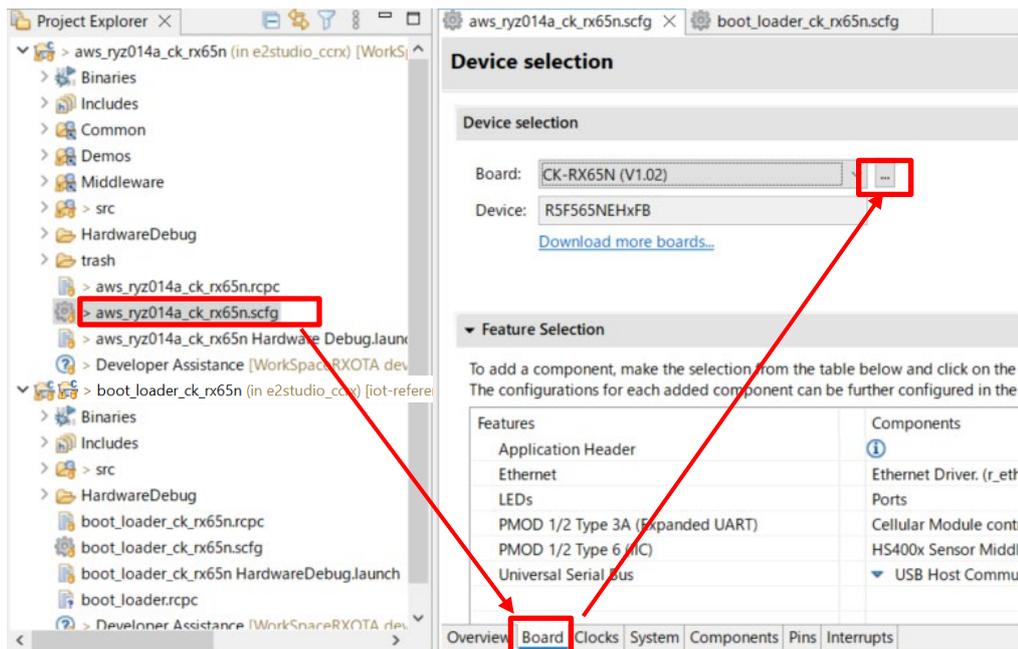
RX Family How to implement FreeRTOS OTA using Amazon Web Services in RX65N (for v202210.01-LTS-rx-1.1.0 or later)

To use the SIM card supplied with the CK-RX65N kit, activate the SIM card by following the procedure in 4.1.5 *Activating SIM card* in the following application note:

[SIM activation, Creating the trial account and using Dashboard with RYZ014A or Ethernet Application for AWS - Getting Started Guide \(R01QS0064\)](#)

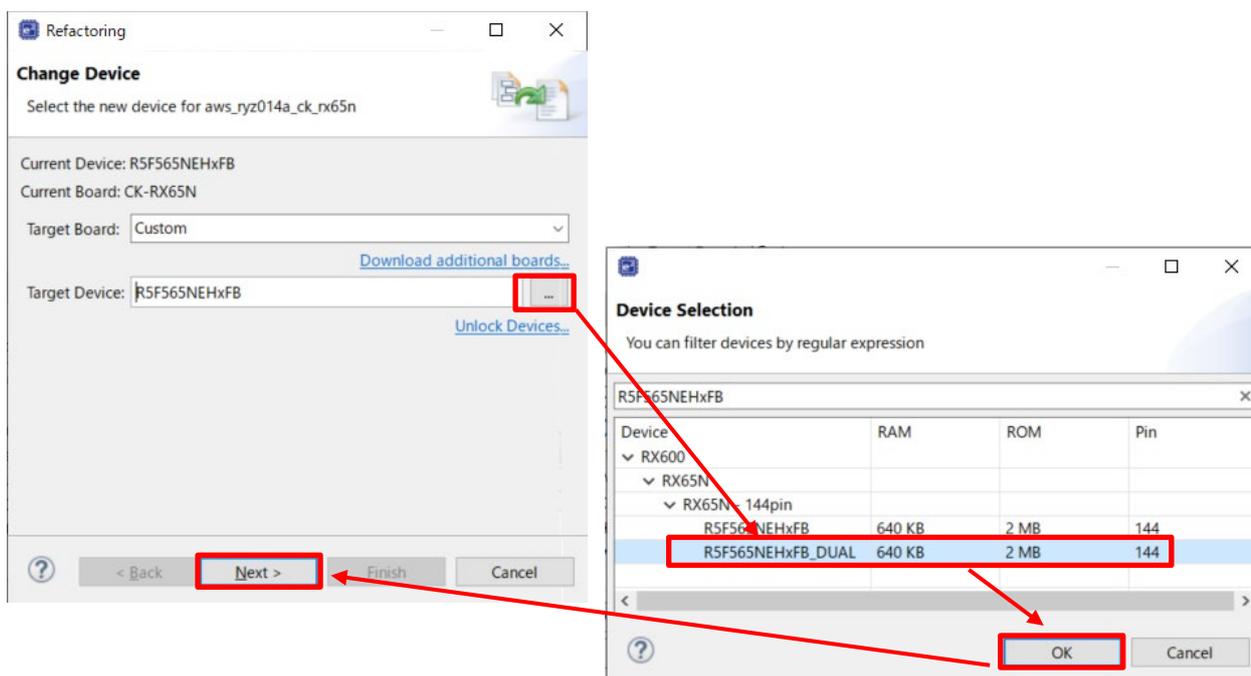
(5) Firmware device settings (1)

Open the file `aws_ryz014a_ck_rx65n.scfg`, and click the **Board** tab. Click the ellipsis (...) beside the **Board** field in the **Device selection** area.



(6) Firmware device settings (2)

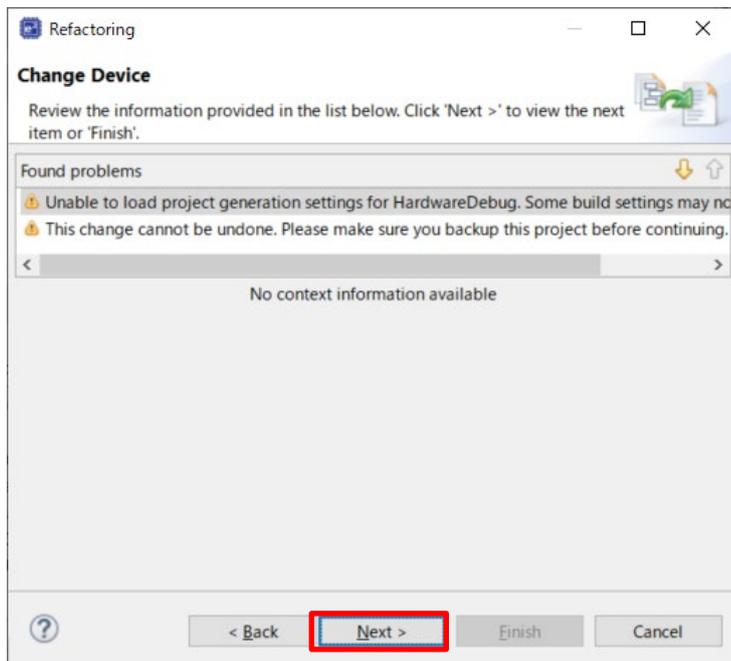
Click the ellipsis (...) beside the **Target Device** field, and select `R5F565NEHxFB_DUAL`. The value in the **Target Board** drop-down list changes to **Custom**.



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

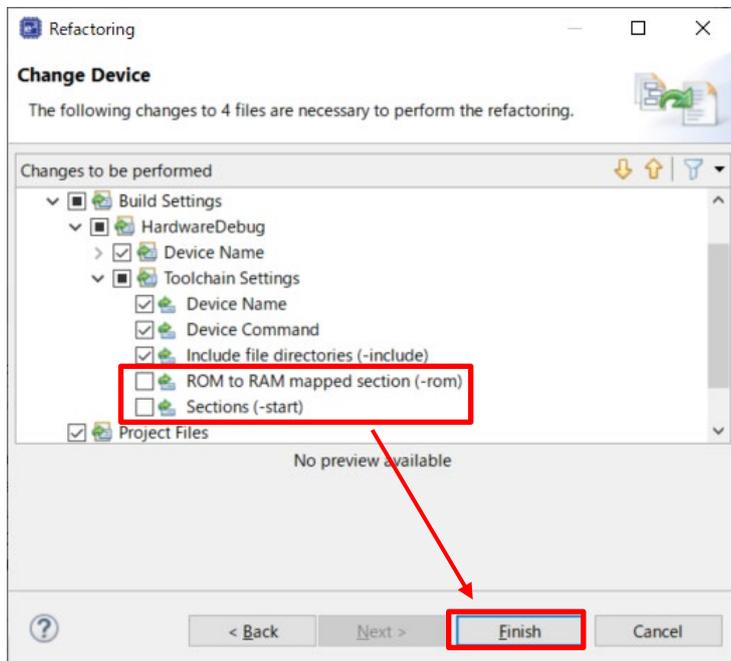
(7) Firmware device settings (3)

When you change a device, the following dialog box appears. Click **Next** to continue.



(8) Firmware device settings (4)

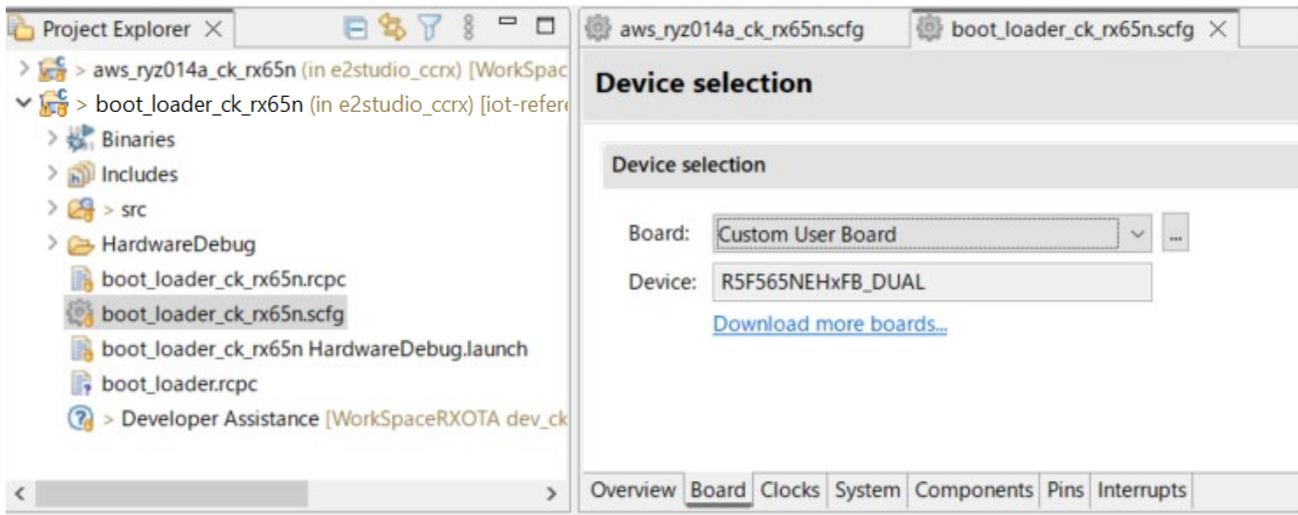
Under **Build Settings > HardwareDebug > Toolchain Settings**, clear the **ROM to RAM mapped section (-rom)** and **Sections (-start)** check boxes and then click **Finish**.



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

(9) Check the boot loader device

Open the file `boot_loader_ck_rx65n.scfg`, and click the **Board** tab. Confirm that `R5F565NEHxFB_DUAL` appears in the **device** field.



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

4.2.4 Creating the initial firmware

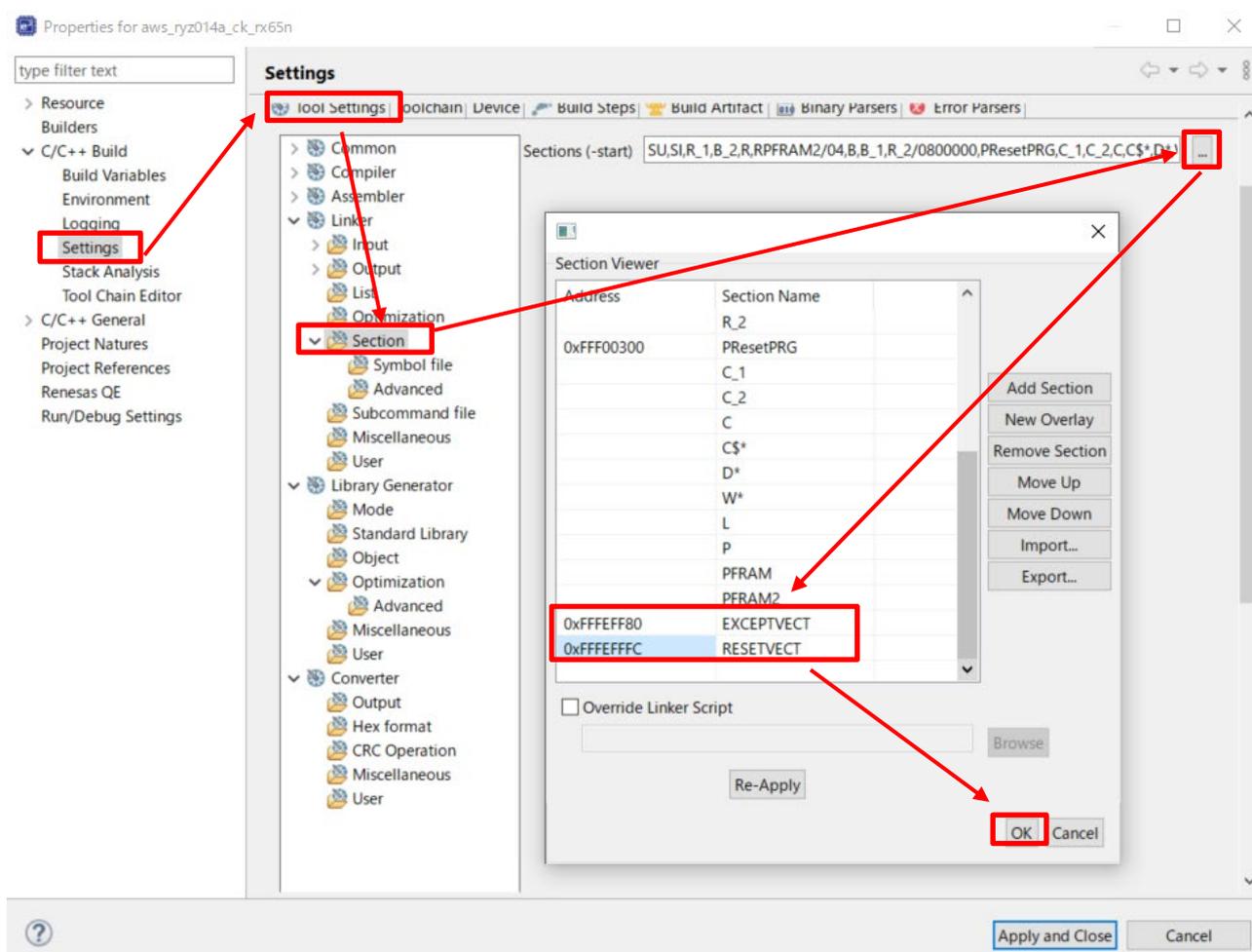
The following explains how to create the initial firmware that combines the boot loader (boot_loader_ck_rx65n) and the firmware (aws_ryz014a_ck_rx65n).

- (1) Change the firmware (aws_ryz014a_ck_rx65n) vector

Open the aws_ryz014a_ck_rx65n project, and select **Project** and then **Properties**.

Expand the **C/C++ Build** menu, and click **Settings**. In the menu tree on the **Tool Settings** tab, expand the **Linker** menu and click **Section**, and open the Section Viewer. Allocate EXCEPTVECT to 0xFFFFF80 and RESETVECT to 0xFFFFEFC.

You can then build the firmware.



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

(2) Use Renesas Image Generator to generate the initial firmware

Place the following files in the Renesas Image Generator folder:

- The results of the build process in 4.2.4(1): `aws_ryz014a_ck_rx65n.mot`
- The results of building the boot loader: `boot_loader_ck_rx65n .mot`
- The private key created in 4.1(7): `secp256r1.privatekey`

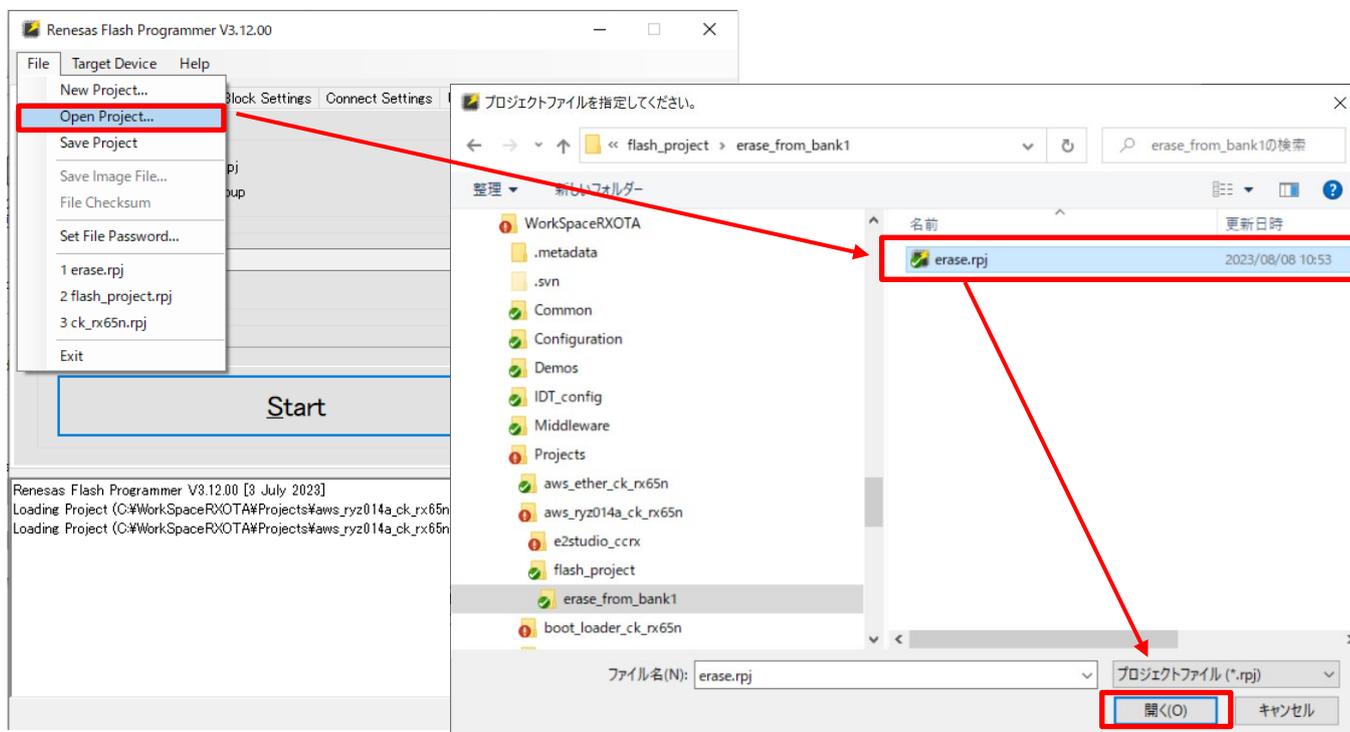
Open a command prompt, navigate to the Renesas Image Generator folder, and execute the following command to generate the file `userprog.mot`.

```
python image-gen.py -iup aws_ryz014a_ck_rx65n.mot -ip RX65N_DualBank_ImageGenerator_PRM.csv -o userprog -ibp boot_loader_ck_rx65n.mot -key secp256r1.privatekey -vt ecdsa -ff RTOS
```

(3) Start Renesas Flash Programmer and open the `erase.rpj` project

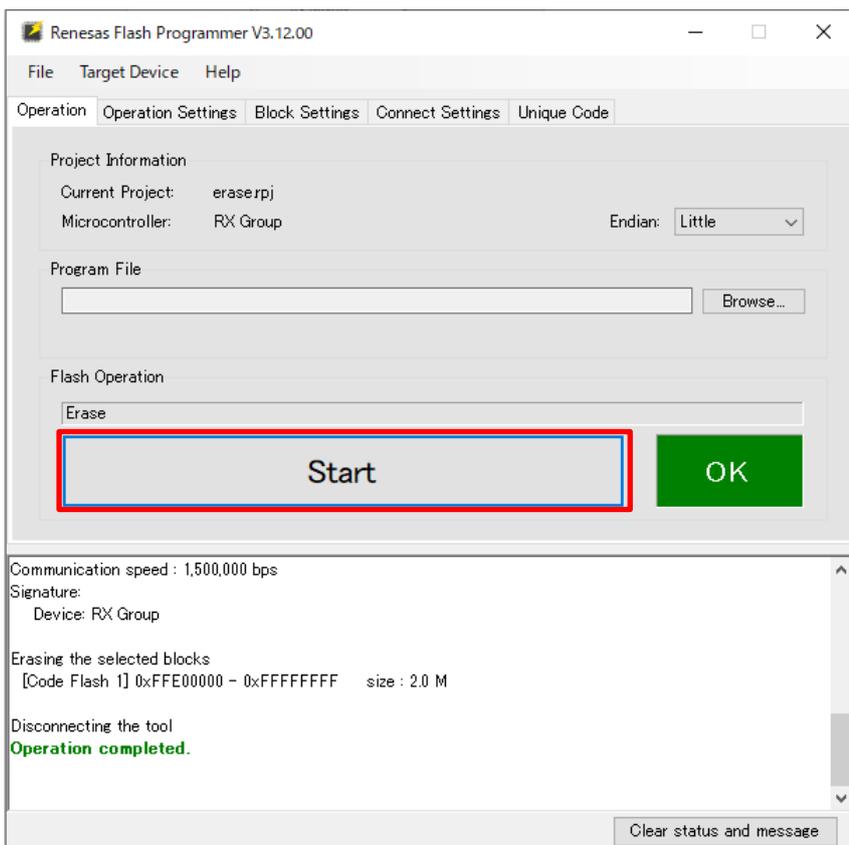
The `erase.rpj` project is located in the following folder of the sample program:

```
\\Projects\aws_ryz014a_ck_rx65n\flash_project\erase_from_bank1
```



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

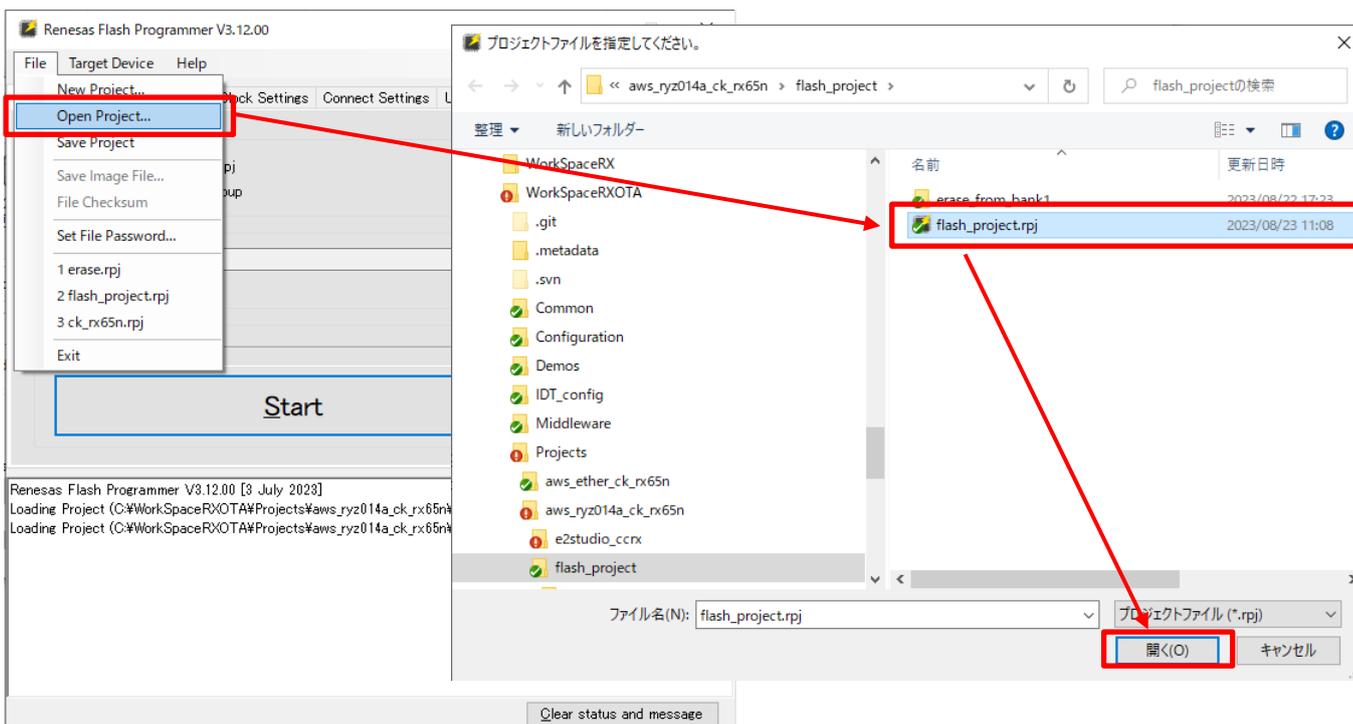
(4) Click **Start** to erase the device



(5) Open the flash_project.rpj project

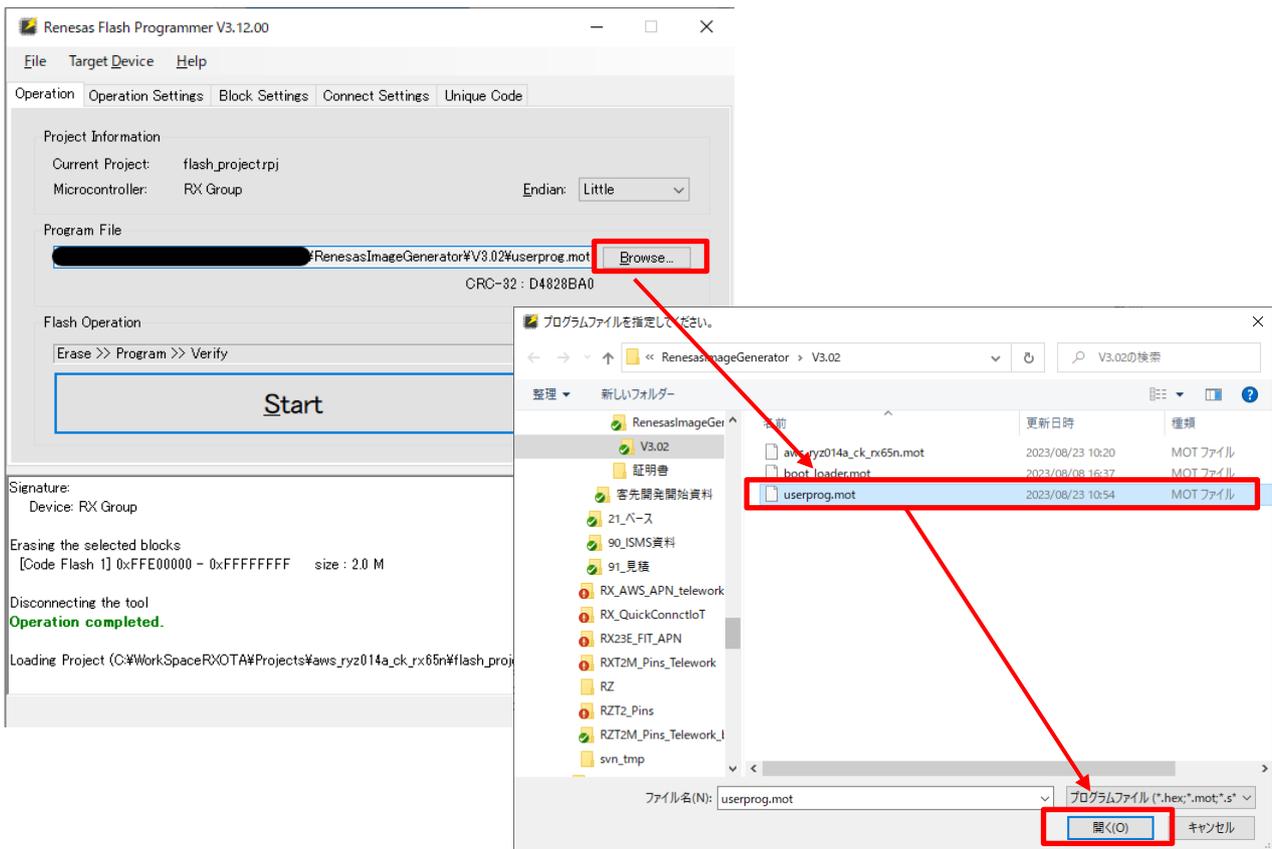
The flash_project.rpj project is located in the following folder of the sample program:

\\Projects\aws_ryz014a_ck_rx65n\flash_project\

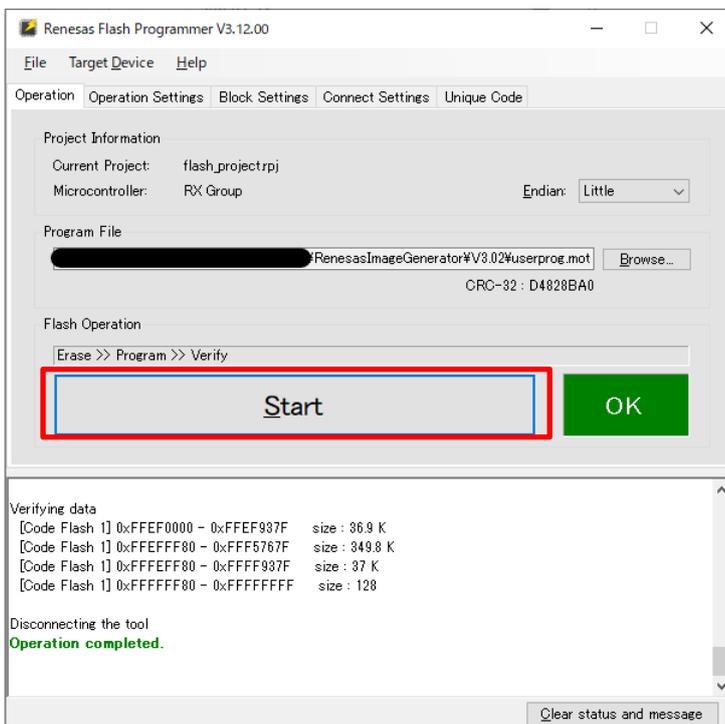


RX Family How to implement FreeRTOS OTA using Amazon Web Services in RX65N (for v202210.01-LTS-rx-1.1.0 or later)

(6) Select the initial firmware (userprog.mot) created in 4.2.4(2)



(7) Write the firmware

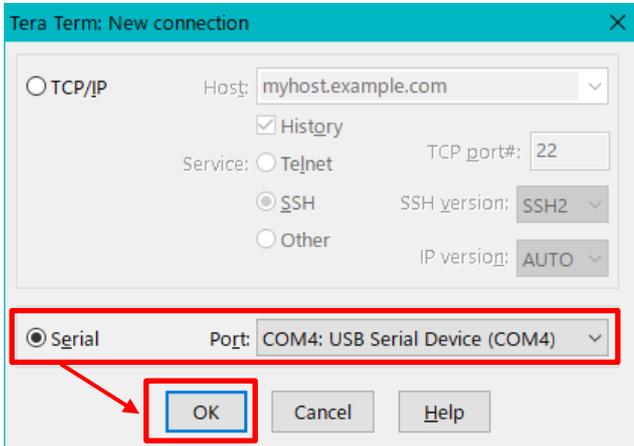


RX Family How to implement FreeRTOS OTA using Amazon Web Services in RX65N (for v202210.01-LTS-rx-1.1.0 or later)

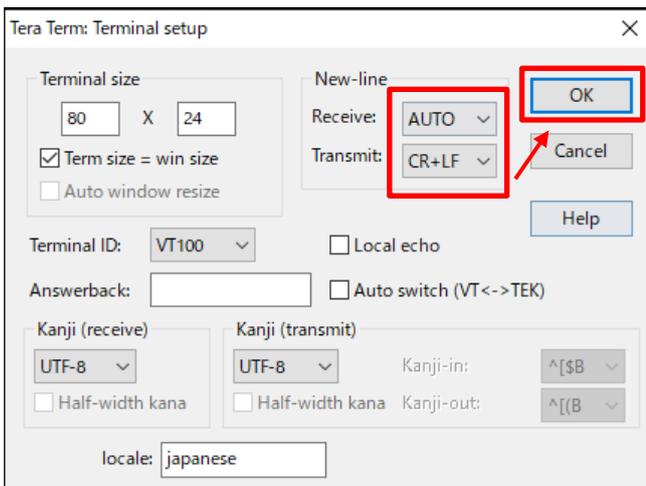
4.2.5 Registering AWS IoT information

The following explains how to set AWS IoT information in Tera Term by running the aws_ryz014a_ck_rx65n project. The information set by this process is written to data flash memory.

- (1) Open Tera Term, and from the **File** menu, select **New Connection**. In the dialog box that appears, select **Serial** and then click **OK**

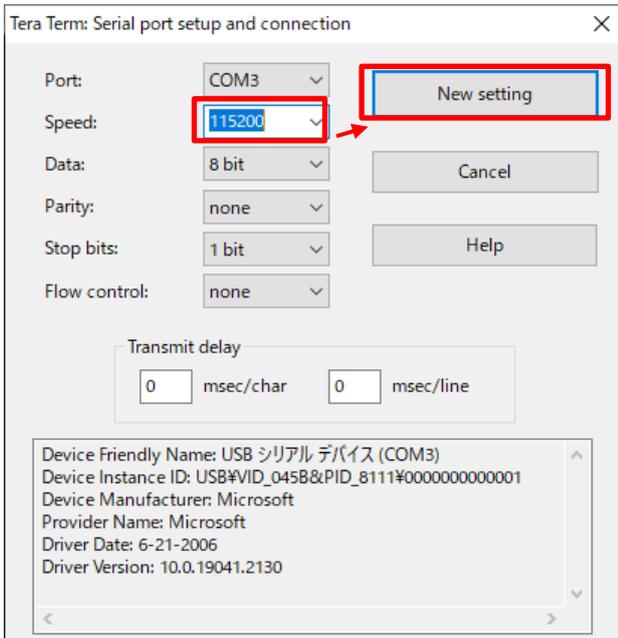


- (2) From the **Setup** menu, select **Terminal**. In the **New-line** area of the dialog box that appears, select **AUTO** for **Receive** and **CR+LF** for **Transmit**

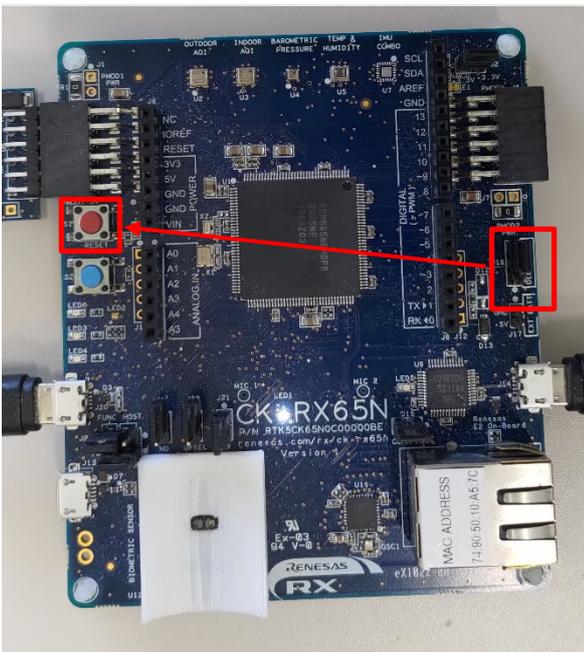


RX Family How to implement FreeRTOS OTA using Amazon Web Services in RX65N (for v202210.01-LTS-rx-1.1.0 or later)

- (3) From the **Setup** menu, select **Serial port**. In the dialog box that appears, set **Speed** to 115200 and then click **New setting**



- (4) Move the jumper on J16 of the CK-RX65N board to the RUN setting, and then press the RESET switch



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

(5) Enter CLI and press the **Enter** key within 10 seconds of the menu appearing in the Tera Term window

```
==== RX65N : BootLoader [dual bank] ====
verify install area 0 [sig-sha256-ecdsa]...OK
execute new image ...
■FreeRTOS command server.
Type Help to view a list of registered commands.

    Standard procedure:
      1. Set value for endpoint/thingname/certificate/key/codesigncert
      .
      2. Write the key value to Internal Data Flash Memory with 'commi
t' command.
      3. Reset the program to start the demo.

>Press CLI and enter to switch to CLI mode or wait 10secs to run demo!

>CLI

Going to FreeRTOS-CLI !
```

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

(6) Register the certificate you downloaded in 3.3.2(6)

Enter conf set cert in Tera Term, and then drag and drop the certificate file (xxxx-certificate.pem.crt) to the Tera Term window to send the file. Finally, press **Enter** in the Tera Term window.

Enter conf set cert, drag and drop the certificate file to the Tera Term window, and then press Enter

The screenshot illustrates the steps to register a certificate in Tera Term:

- File Explorer:** Shows the local disk path `Local Disk (C:) > openssl > Cert`. Three files are listed: `ec1-certificate.pem.crt` (Security Certificate, 2 KB), `ec1-private.pem.key` (KEY File, 2 KB), and `ec1-public.pem.key` (KEY File, 1 KB).
- Tera Term File Drag and Drop:** A dialog box is open, showing the file `ec1-certificate.pem.crt` has been dragged in. The "Send File (Paste content of file)" option is selected. The "OK" button is highlighted.
- Terminal:** The command `>conf set cert -----BEGIN CERTIFICATE-----` is entered. A large block of certificate data is pasted into the terminal. The data ends with `-----END CERTIFICATE-----`. The terminal shows `OK.` and a prompt `>`.

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

(7) Register the private key you downloaded in 3.3.2(6)

Enter conf set key in Tera Term, and then drag and drop the private key file (xxxx- private.pem.key) to the Tera Term window to send the file. Finally, press **Enter** in the Tera Term window.

Enter conf set key, drag and drop the private key file to the Tera Term window, and then press Enter

```
>conf set key -----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAtNt60e3u4T1RRdyjBu8PaEchudXtZkoFJGM25fD8/Z
M5VuCPH9F7P1fn8d20GIvswmXwScWAcyng40I/nCdE1aPpCBLKr6i8iVJ9
wiIwkh1Ho91g4UpeXkAWrDxj0F16vvBuX32k/8IBmculI1HbA+07H1g5q
FIq1th0QDeL4Y0AB3W0nv4Nbpj0uAxeMfPm9T61Mb1aJq1uRoGRM6BoC10
XUVmeJ3out6MJxGdTpjLnwYooBE3Mm88h+SqHNHogCne1bi0wrVprCAZh
G9vM4xctSA5GuLj1EKtBAvRnWRjpnqSwfFGsQIDAQABAgIBAEFK0kZx62
rE7jH8y+w10hQZq3JDsh/1P9qz76mVUwWDTIPImQskhgB+AM1Xb+PcmDp
dN+s8CRRDUv9dPIRY5Km56sBSZqW/uvjsza+zCJmObaBIwrUapDFcGG17S
pmyr2bPLVLUmWv8s1JyphJULwvHVX478WYE8QvIzyKm8iyaCU8Fh7fHFN
lnhkUzH49nwEgtQgEA7tfvRDGYE84i+RVWjdHMptxRpAKEO/KYnT7s1s5X
oNOB9VVU1mw8VoVewC6B19F1cT6v8bSWy1Cv6jviH360y4VCj1ZXmsQDCo
scSLP0ECgYEA/CZ219PNu9ZHV+/uvEdk90tCNaG0nc9Q3IL+G3rsp5ztaz
tMhMQ3zQkfW8sFTwFQLascUdWn+zGs5MDjY2sMHEaNAwNGAg+FPzr1igRI
LDbaqGp0vEmCTbAo2b202Be6yxcX2P1swcMjQC1tRaV3Lx1BMCyC9akCgY
f0pWf6J++PRA3ukKgx0vJLkVWAAtVqoRxxvUvL8JiQPNGnD5qvHzBe/THvdx
26EhQfePFdh2yYyC1XnL6DgV2IDENEuzsB06QhEzTg6w6HUWSBWeEgN+xQ
watCKDuomkakUJIfJWcPQKS5Sap5UcAo0AjgXckCgYEAyb1QTGsY/1CA8U
1jT6WxkY0IdhMckDUdfVxSotP0xKZJwLvqRLHrBvbVARhkw94comdS+716w86R
+CxVwz68pjsi1EKxHPPK0IDKowS6GwN076Fe0pDNoHAM06g+hS7D1riYqyx13yJm
ZPh7kjLEXhnmJLj6S1bQy0ECgYEAvaq6XzQG34+78tU8fz2QNE4oBTyHwe1jYIzQ
ZFhtctDa+jkPXC44E100zyFPXxNgDF106rSkwRCEYziKiGgj5y0hdUaxc0ULEVX/
kEUmyuAAjoi1TJFHk2iryNFo1EBjVn1iTiJazwu41SzrOZ1P59000bSyhh7kU8G8
vL4ZamkCgYAHIDoyaOPP9PuX5K8IKb7Sgper7Zy3tTsCKtDASWnB3vLiLlWMnQ9B
YN56j1BcRcBIRQcXE5bfAg2uyocnvhf5/VoYocnUZ7/qFkBgP1ar1Vh26V9jSrP
Ickj19SEp+kJ01xCRLLVh8++pDmgpFbdgnp/1DndDAWUJfJxfxRvoA==
-----END RSA PRIVATE KEY-----
```

OK.

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

- (8) Register the thing name you set in 3.3.2(3) and the endpoint you made a note of in 3.3.3(1)

Execute the following commands in Tera Term:

```
conf set thingname thing-name
conf set endpoint endpoint-name
```

```
>conf set thingname rx65n_ota_demo_thing
OK.
>conf set endpoint a1k[REDACTED]ap-northeast-1.amazonaws.com
OK.
```

- (9) Register the key pair certificate (secp256r1.crt) generated in 4.1(6)

Enter conf set codesigncert in Tera Term, and then drag and drop the key pair certificate (secp256r1.crt) to the Tera Term window to send the file.

Note: Change the linefeed code of the certificate file to LF before pasting the file contents

Enter conf set codesigncert, drag and drop the key pair certificate to the Tera Term window, and then press Enter

```
>conf set codesigncert -----BEGIN CERTIFICATE-----
MIIB/zCCAaUCFAsT434WX7IKRUKZEchX2aP9+jdtMAoGCCqGSM49BAMCMIGB
CQYDVQQGEwJKUDEOMAwGAT1UECAwFT3Nha2ExDjAMBgNVBAcMbu9zYyW+hMREw
VQOKDAhQaXN5c3RlbnRlMAkGA1UECwwCUOUxEDA0BgNVBAMMbu9zYyW1cmEx
BgkqhkiG9w0BCQEWew9zYyW1cmFAcGlzLmNvbnR1cm9udA2MTA2ZDZlMzEz
DTMzMDgwMTA2MTAzMFowYExCzAJBgNVBAYTAkpQM04wDAYDVQQIDAVFb2Fz
MAwGA1UEBwwFT3Nha2ExETAPBgNVBAoMCFBpc3IzdGVtMQswCQYDVQQLDAJ1
MA4GA1UEAwwHb3NhbXVyeTEgMB4GCStGSM49BAMCMIGB
anAwWTATBgkqhkiG9w0BCQEWew9zYyW1cmFAcGlzLmNvbnR1cm9udA2MTA2ZDZlMzEz
DTMzMDgwMTA2MTAzMFowYExCzAJBgNVBAYTAkpQM04wDAYDVQQIDAVFb2Fz
MAwGA1UEBwwFT3Nha2ExETAPBgNVBAoMCFBpc3IzdGVtMQswCQYDVQQLDAJ1
MA4GA1UEAwwHb3NhbXVyeTEgMB4GCStGSM49BAMCMIGB
AIEAgX6t1y22ZuK8GC7cFTf18F6Ej9p20VzKIZQoI r6qBX0=
-----END CERTIFICATE-----
```

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

(10) Commit the AWS IoT settings (write the settings to data flash memory)

Execute the following commands in Tera Term:

```
conf commit
```

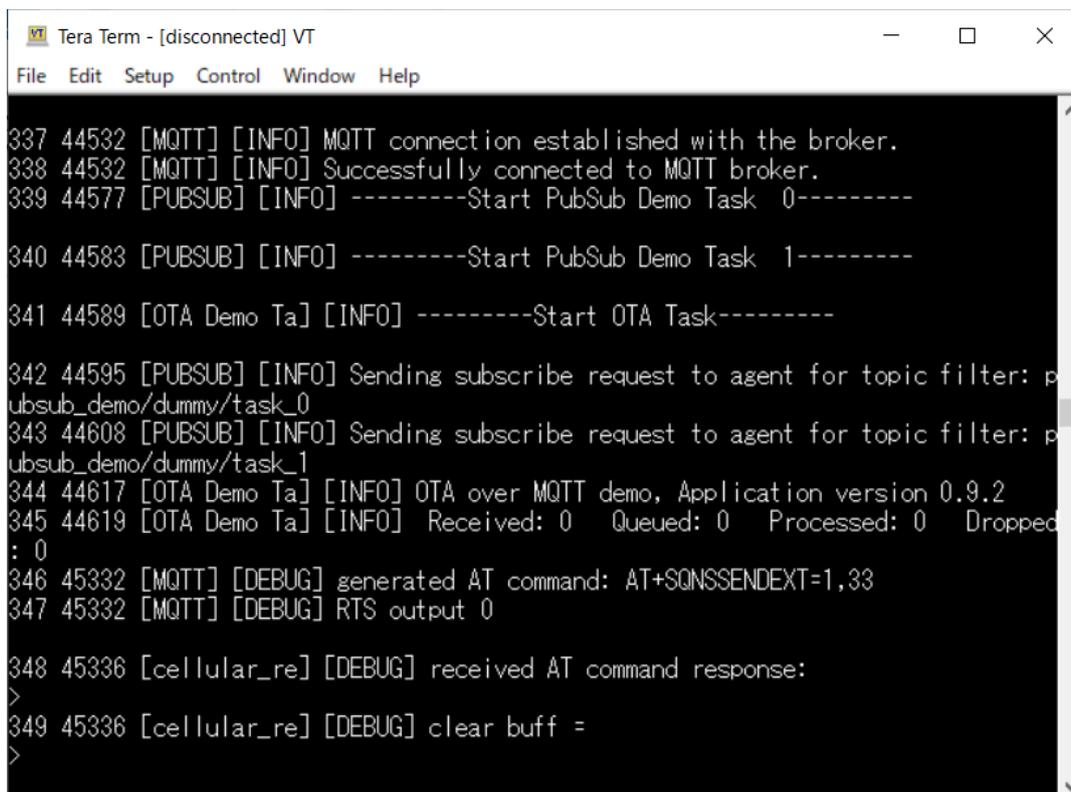
```
>conf commit
0 4472481 [CLI] Destroyed Certificate.
1 4472485 [CLI] Write certificate...
2 4472545 [CLI] Destroyed Private key.
3 4472685 [CLI] Write Private key...
Configuration saved to Data Flash and used 2879 bytes.
```

(11) Perform a reset

Execute the following commands in Tera Term:

```
reset
```

After the reset process is complete, confirm that Tera Term displays a communication log and the application is waiting for OTA jobs.



The screenshot shows a Tera Term window titled "Tera Term - [disconnected] VT". The terminal output displays the following log entries:

```
337 44532 [MQTT] [INFO] MQTT connection established with the broker.
338 44532 [MQTT] [INFO] Successfully connected to MQTT broker.
339 44577 [PUBSUB] [INFO] -----Start PubSub Demo Task 0-----
340 44583 [PUBSUB] [INFO] -----Start PubSub Demo Task 1-----
341 44589 [OTA Demo Ta] [INFO] -----Start OTA Task-----
342 44595 [PUBSUB] [INFO] Sending subscribe request to agent for topic filter: p
ubsub_demo/dummy/task_0
343 44608 [PUBSUB] [INFO] Sending subscribe request to agent for topic filter: p
ubsub_demo/dummy/task_1
344 44617 [OTA Demo Ta] [INFO] OTA over MQTT demo, Application version 0.9.2
345 44619 [OTA Demo Ta] [INFO] Received: 0 Queued: 0 Processed: 0 Dropped
: 0
346 45332 [MQTT] [DEBUG] generated AT command: AT+SQNSSENDEXT=1,33
347 45332 [MQTT] [DEBUG] RTS output 0
348 45336 [cellular_re] [DEBUG] received AT command response:
>
349 45336 [cellular_re] [DEBUG] clear buff =
>
```

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

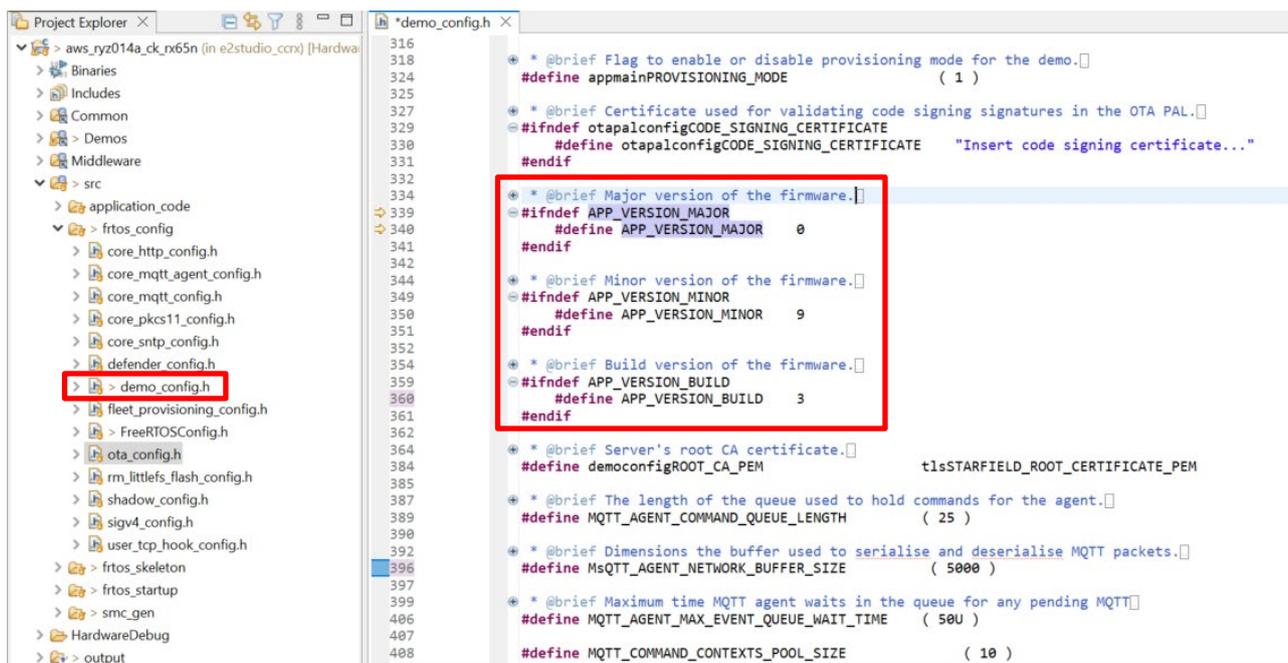
5. Updating the Firmware

5.1 Creating the updated firmware

5.1.1 Changing the firmware version

- (1) Change the firmware version to v0.9.3

Repeat the build process, this time with 3 specified for the APP_VERSION_BUILD definition in aws_ryz014a_ck_rx65n\src\frtos_config\demo_config.h.



- (2) Use Renesas Image Generator to generate the updated firmware

Overwrite the file in the Renesas Image Generator folder with the firmware you rebuilt in 5.1.1(1) (aws_ryz014a_ck_rx65n.mot), and then execute the following command at the command prompt:

```
python image-gen.py -iup aws_ryz014a_ck_rx65n.mot -ip RX65N_DualBank_ImageGenerator_PRM.csv -o user_093 -key secp256r1.privatekey -vt ecdsa -ff RTOS
```

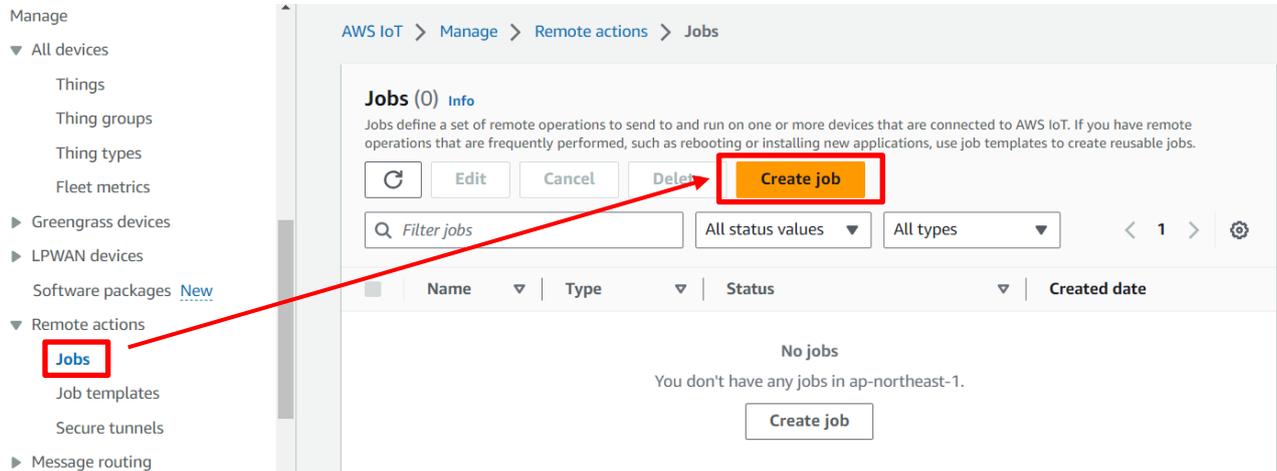
This command generates a file named user_093.rsu.

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

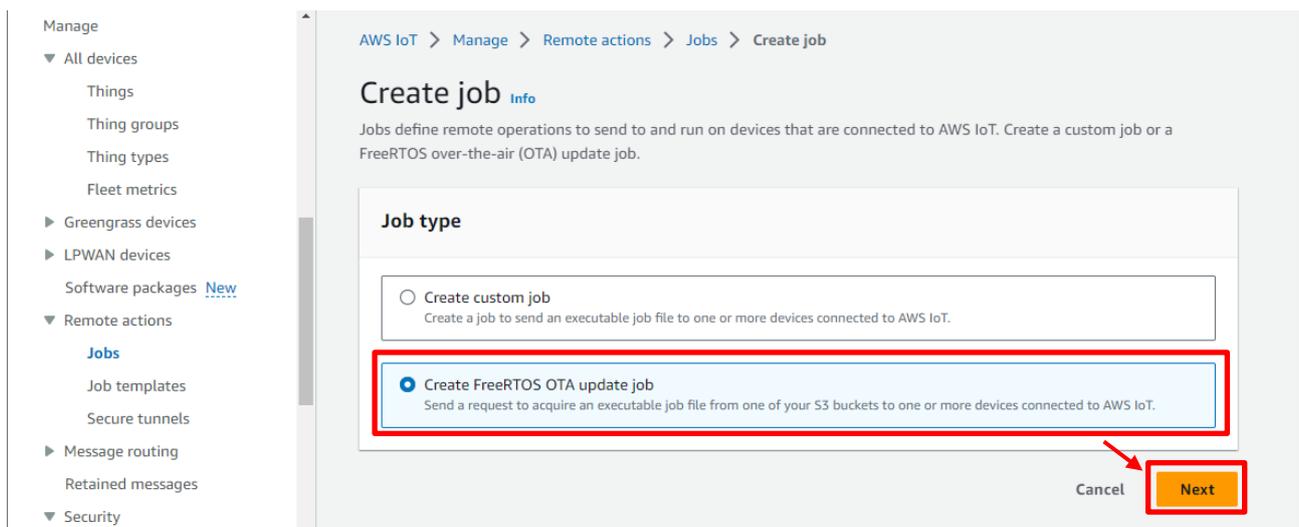
5.2 Updating the firmware

In AWS, create an OTA update job that will update the firmware.

- (1) In the IOT Core menu, select **Manage, Remote actions, and Jobs**, and then click the **Create job** button

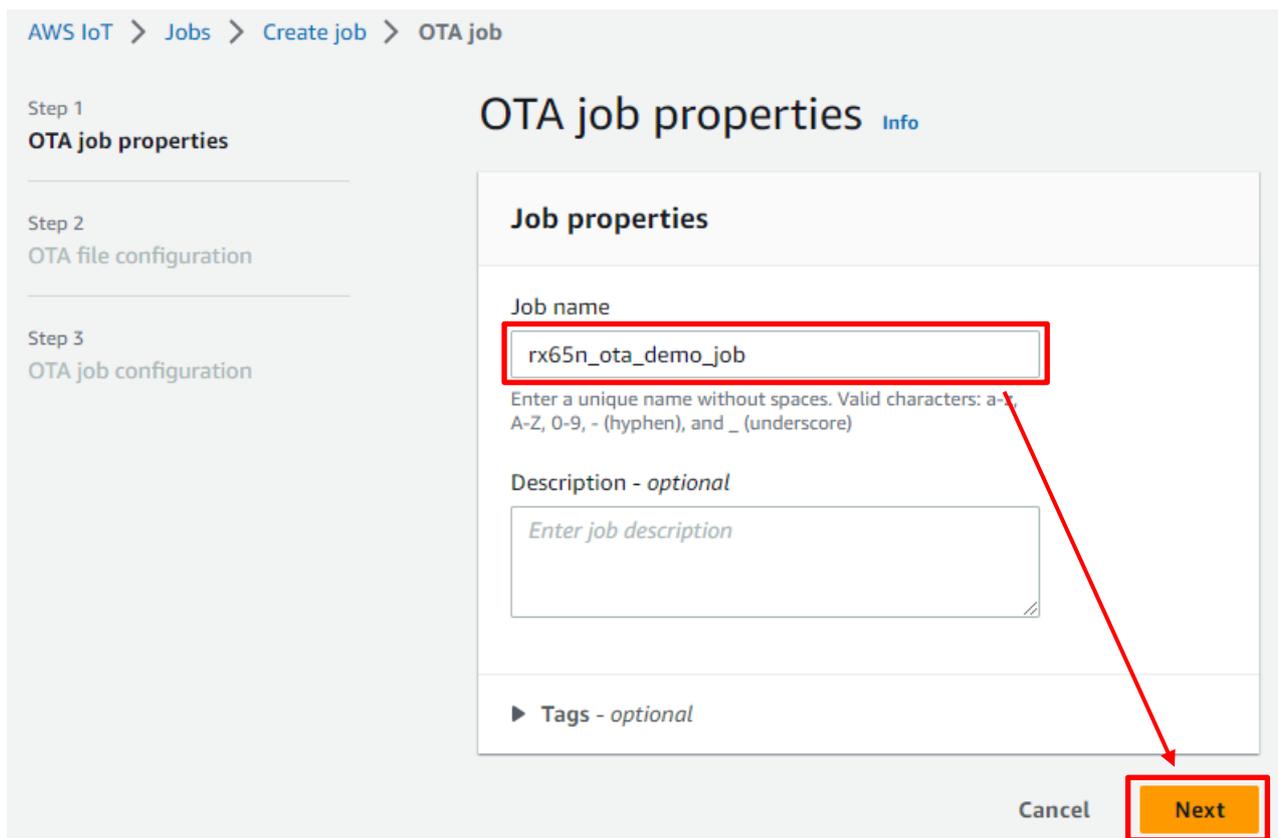


- (2) Select **Create FreeRTOS OTA update job** and then click **Next**



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

(3) Enter a job name (example: rx65n_ota_demo_job) and then click **Next**



AWS IoT > Jobs > Create job > OTA job

Step 1
OTA job properties

Step 2
OTA file configuration

Step 3
OTA job configuration

OTA job properties [Info](#)

Job properties

Job name

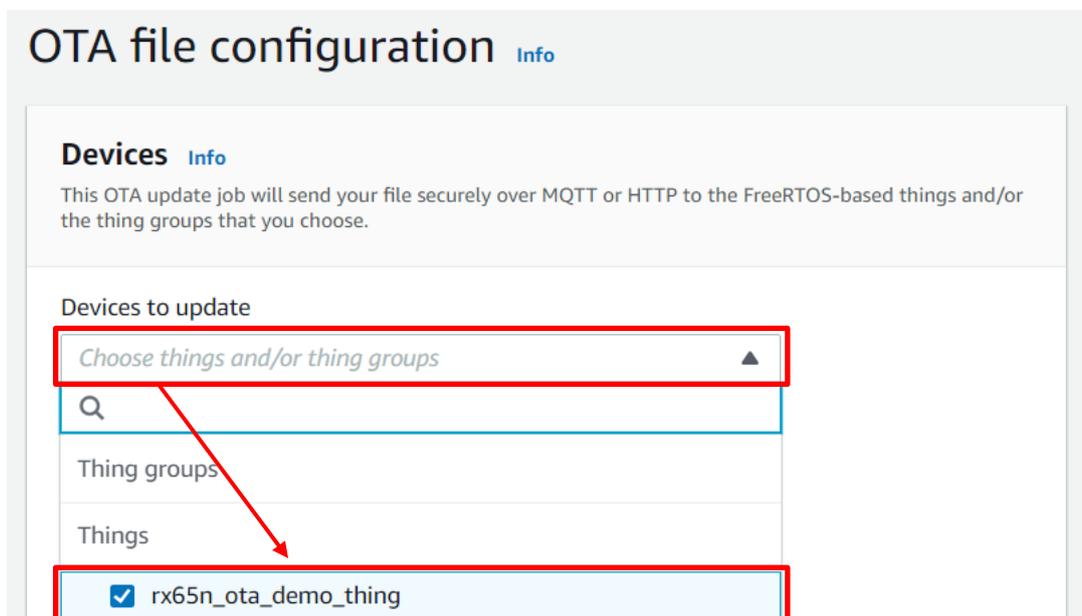
Enter a unique name without spaces. Valid characters: a-z, A-Z, 0-9, - (hyphen), and _ (underscore)

Description - *optional*

▶ Tags - *optional*

Cancel **Next**

(4) Click the **Devices to update** drop-down list and select the device to update



OTA file configuration [Info](#)

Devices [Info](#)

This OTA update job will send your file securely over MQTT or HTTP to the FreeRTOS-based things and/or the thing groups that you choose.

Devices to update

Q

Thing groups

Things

rx65n_ota_demo_thing

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

(5) Click **Create new profile**

File Info

Sign and choose your file
Code signing ensures that devices only run code published by trusted authors and that the code hasn't been changed or corrupted since it was signed. You have three options for code signing.

Sign a new file for me. Choose a previously signed file. Use my custom signed file.

Code signing profile
This profile will contain information needed to create a code signing job. The profile specifies your device's hardware platform, certificate from AWS Certificate Manager, and the location of your code signing certificate path on your device.

Existing code signing profile
Choose existing code signing profile ▼ **Create new profile**

You can skip steps (5) to (9) if you have already created a profile. Click **Choose existing code signing profile** and select the profile you created from the drop-down list.

Existing code signing profile

Choose existing code signing profile ▲

Q

rx65n_ota_demo_profile2	/dummy
SHA256	ECDSA
rx65n_ota_demo_profile	dummy
SHA256	ECDSA

File to upload

Select an existing file.

Create new profile

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

- (6) Create a profile (1): Profile name and device hardware platform
- Enter the profile name (example: rx65n_ota_demo_profile)
 - Select **Windows Simulator** as the device hardware platform

Create a code signing profile [X]

Profile name
rx65n_ota_demo_profile
Enter a unique name without spaces. Valid characters: a-z, A-Z, 0-9, and _ (underscore)

Device hardware platform
Windows Simulator

- (7) Create a profile (2): Import a certificate
- In the **Code signing certificate** area, click **Import new code signing certificate**
 - In **Certificate body**, select the file secp256r1.crt you created in 4.1(6)
 - In **Certificate private key**, select the file secp256r1.privatekey you created in 4.1
 - In **Certificate chain**, select the file ca.crt you created in 4.1(3)
 - Click **Import**

Create a code signing profile [X]

Code signing certificate
AWS Certificate Manager (ACM) handles the complexity of creating, managing, or importing SSL/TLS certificates. You can use ACM to create an ACM Certificate or import a third-party certificate that you use for signing. You must have a certificate to sign code.

Import new code signing certificate Select an existing certificate

Certificates

Certificate body	secp256r1.crt 753 bytes Uploaded
Certificate private key	secp256r1.privatekey 232 bytes Uploaded
Certificate chain - optional	ca.crt 890 bytes Uploaded

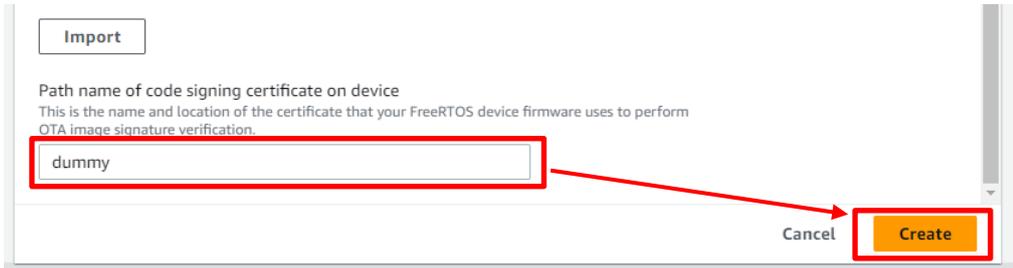
Import

Path name of code signing certificate on device
This is the name and location of the certificate that your FreeRTOS device firmware uses to perform OTA image signature verification.

Cancel Create

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

- (8) Create a profile (3): Enter the path of the code signing certificate of the device and then click **Create**
You can enter any path. (Example: dummy)



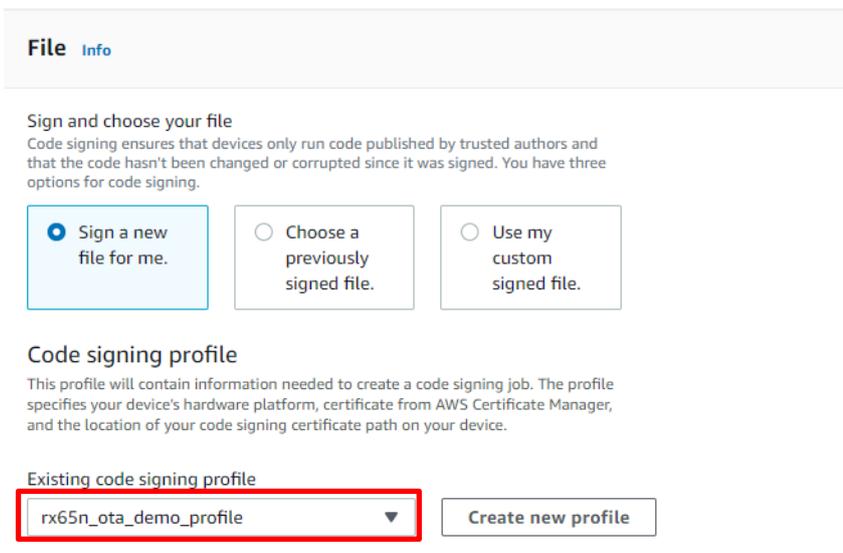
Import

Path name of code signing certificate on device
This is the name and location of the certificate that your FreeRTOS device firmware uses to perform OTA image signature verification.

dummy

Cancel Create

- (9) Confirm that the name of the profile you created earlier is selected in the **Existing code signing profile** drop-down list



File Info

Sign and choose your file
Code signing ensures that devices only run code published by trusted authors and that the code hasn't been changed or corrupted since it was signed. You have three options for code signing.

Sign a new file for me. Choose a previously signed file. Use my custom signed file.

Code signing profile
This profile will contain information needed to create a code signing job. The profile specifies your device's hardware platform, certificate from AWS Certificate Manager, and the location of your code signing certificate path on your device.

Existing code signing profile
rx65n_ota_demo_profile ▼

Create new profile

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

(10) Update the firmware

- Select **Upload a new file**
- In **File to upload**, select the file `usr093.rsu` you created in 5.1.1(2)
- Click **Browse S3** and select the S3 bucket you created in 3.4
- Enter a path name in **Path name of file on device** (You can enter any path name. Example: `/device/updates`)

File

Upload a new file. Select an existing file.

File to upload

`usr093.rsu`
357888 bytes

File upload location in S3

This is the location in S3 where your file will be stored.

S3 URL

Format: `s3://bucket/prefix/object`.

Path name of file on device

This is the name and location where the file will be stored on the FreeRTOS device.

▶ File type - optional

(11) In the **Role** drop-down list, select the role you created in 3.5(5) and then click **Next**

IAM role [Info](#)

Role

Choose a role that grants AWS IoT access to S3, AWS IoT jobs, and AWS Code signing resources.

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

(12) Click **Create job**

OTA job configuration [Info](#)

Job run type
Choose how to run this job.

- Your job will complete after deploying to the devices and groups that you chose (snapshot)
- Your job will continue to deploy to any devices added to the groups that you chose (continuous)

► **Job start rollout configuration - optional**
Specify how quickly devices will be notified when a pending job starts.

► **Job stop configuration - optional**
These configurations define when to automatically stop the job. The job stops if a percentage of devices fail the deployment after a minimum number have deployed. The job cancels if any of the criteria are met after the job starts.

► **Job run timeout configuration - optional**
Specify how long the job will run.

Cancel Back **Create job**

(13) Wait until firmware reception is complete

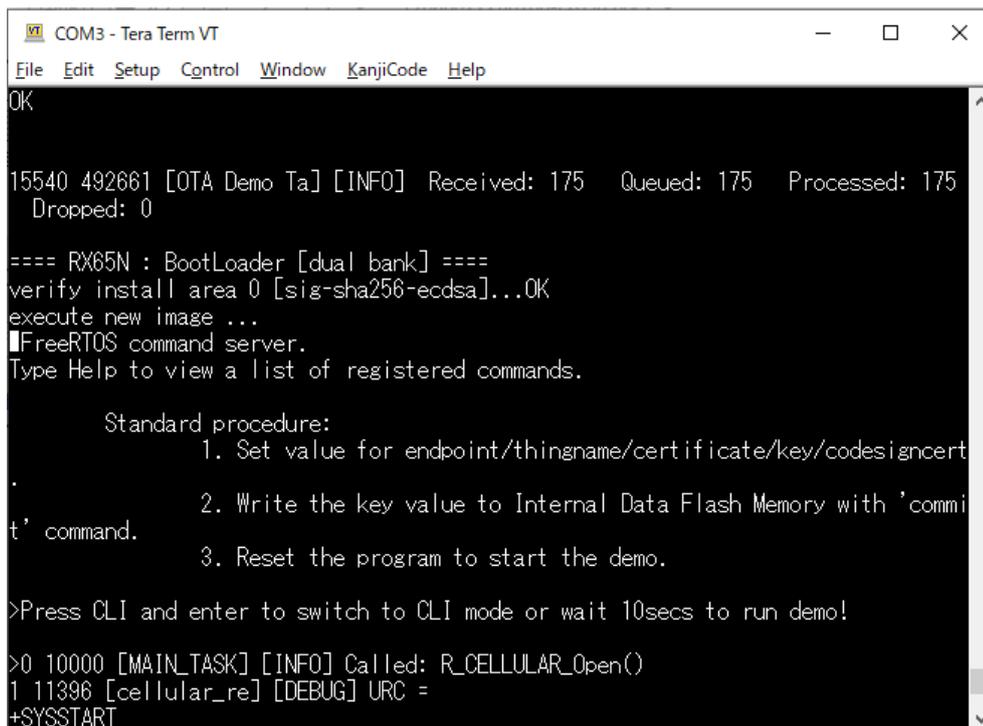
When the job starts, the job receives and writes the firmware.

```
COM3 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
4049 186594 [cellular_re] [DEBUG] clear buff =
OK
4050 186663 [OTA Demo Ta] [INFO] Received: 11 Queued: 11 Processed: 11 Dr
opped: 0
4051 187230 [cellular_re] [DEBUG] URC =
+SQNSRING: 1,1024
4052 187231 [MQTT] [DEBUG] generated AT command: AT+SQNSRECV=1,5
4053 187231 [MQTT] [DEBUG] RTS output
4054 187236 [cellular_re] [DEBUG] URC =
+SQNSRECV: 1,5
4055 187238 [cellular_re] [DEBUG] received AT command response:
OK
4056 187238 [cellular_re] [DEBUG] clear buff =
OK
```

The **Received** counter is incremented when reception starts

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

When the update process is complete, the device resets and the initial menu appears.



```
COM3 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
OK
15540 492661 [OTA Demo Ta] [INFO] Received: 175   Queued: 175   Processed: 175
      Dropped: 0

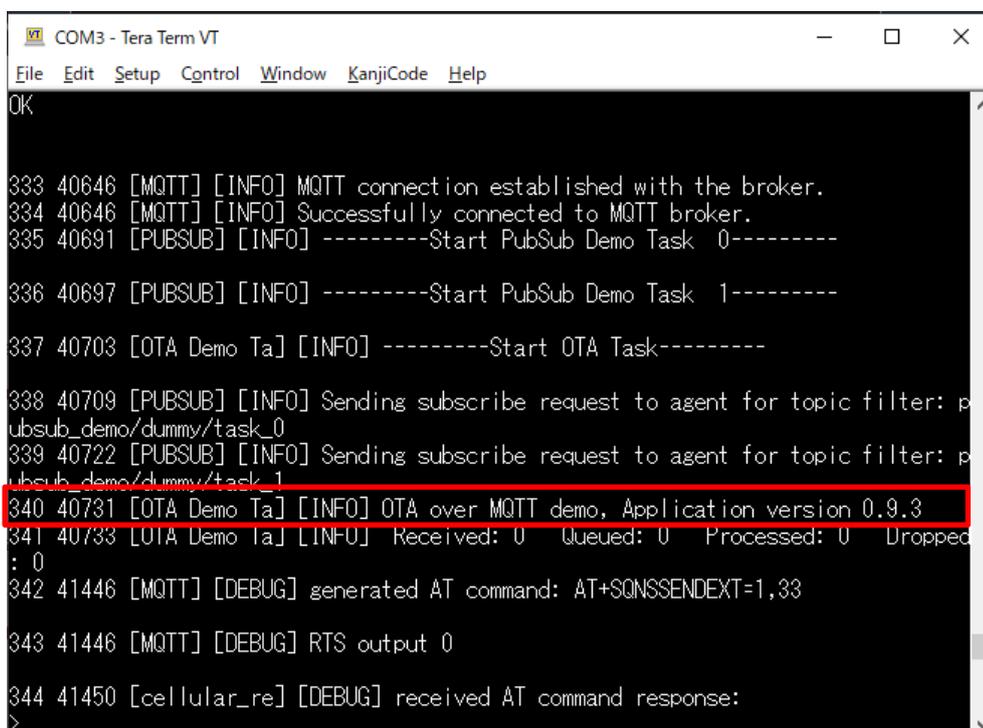
==== RX65N : BootLoader [dual bank] ====
verify install area 0 [sig-sha256-ecdsa]...OK
execute new image ...
FreeRTOS command server.
Type Help to view a list of registered commands.

      Standard procedure:
      1. Set value for endpoint/thingname/certificate/key/codesigncert
      2. Write the key value to Internal Data Flash Memory with 'commit'
      3. Reset the program to start the demo.

>Press CLI and enter to switch to CLI mode or wait 10secs to run demo!

>0 10000 [MAIN_TASK] [INFO] Called: R_CELLULAR_Open()
1 11396 [cellular_re] [DEBUG] URC =
+SYSSSTART
```

(14) Confirm that the firmware version is Ver. 0.9.3



```
COM3 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
OK
333 40646 [MQTT] [INFO] MQTT connection established with the broker.
334 40646 [MQTT] [INFO] Successfully connected to MQTT broker.
335 40691 [PUBSUB] [INFO] -----Start PubSub Demo Task 0-----
336 40697 [PUBSUB] [INFO] -----Start PubSub Demo Task 1-----
337 40703 [OTA Demo Ta] [INFO] -----Start OTA Task-----
338 40709 [PUBSUB] [INFO] Sending subscribe request to agent for topic filter: p
ubsub_demo/dummy/task_0
339 40722 [PUBSUB] [INFO] Sending subscribe request to agent for topic filter: p
ubsub_demo/dummy/task_1
340 40731 [OTA Demo Ta] [INFO] OTA over MQTT demo, Application version 0.9.3
341 40733 [OTA Demo Ta] [INFO] Received: 0   Queued: 0   Processed: 0   Dropped
: 0
342 41446 [MQTT] [DEBUG] generated AT command: AT+SQNSSENDEXT=1,33
343 41446 [MQTT] [DEBUG] RTS output 0
344 41450 [cellular_re] [DEBUG] received AT command response:
>
```

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

6. Troubleshooting

The following table lists issues that might arise when executing the sample, and how to resolve them.

Table 6-1 Troubleshooting

No.	Issue	Cause	Solution	Refer to
1	The command to create the initial firmware fails	The Python installation folder is not set correctly in the Path variable	Reinstall python. Also, make sure that the Add python.exe to PATH check box is selected when you perform the steps in 2.2(3).	2.2
2		The encryption library is not installed	Install the encryption library.	2.2(5)
3	The initial firmware cannot be written	The CK-RX65N kit is not in debug mode	Make sure that the jumper on J16 of the CK-RX65N board is on pins 1-2 (debug mode).	2.5
4	The initial firmware does not start	The CK-RX65N kit is not in RUN mode	Make sure that the jumper on J16 of the CK-RX65N board is on pins 2-3 (RUN mode).	4.2.5(4)
5	Cellular communication cannot start	The RYZ014A PMOD board is not connected properly	Check the connection of the RYZ014A PMOD board.	2.5
6		No SIM card is inserted	Insert the SIM card.	2.5
7		The SIM card is configured incorrectly	Revise the configuration of the r_cellular module.	4.2.3(4)
8		You are using the SIM card supplied with the CK-RX65N kit but the SIM card is not activated	Activate the SIM card.	4.2.3(4)
9	An error occurs during cellular communication	The communication environment is poor	Connect an antenna and power supply to the RYZ014A PMOD board. Also, place the antenna in an area with good reception such as near a window.	2.5
10	An error occurs when connecting to AWS	The AWS IoT information is not set or is set incorrectly	Set the AWS IoT information again.	0
11	The firmware does not start after starting the boot loader	The public key is not correctly set in the boot loader	Review the public key setting in the boot loader.	4.2.3(1)
12	The firmware does not start after an OTA update	The public key is not correctly set in the firmware	Review the public key setting in the firmware.	4.2.3(1)
13		Device selection is incorrect	Review the device setting in the firmware and the boot loader.	4.2.3(5) to 4.2.3(9)

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

Revision History

Rev.	Date issued	Details	
		Page	Nature of revision
1.00	Sept. 15, 2023	—	Initial publication

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/.