RX Family

Firmware Update Software Development Guide using AWS/Azure QE for OTA

Introduction

This application note describes how to install and use the cloud-based OTA (Over the Air) firmware updates using QE for OTA, a software development support tool for cloud systems.

Target Devices

CK-RX65N (Cloud Kit for RX65N Microcontroller Group) Cellular Version

Contents

1. Overview 3
1.1 System Overview 3
1.2 QE for OTA Structure 3
1.2.1 OTA Main (QE) View 3
1.2.2 OTA Manage IoT Device (QE) View 4
1.3 Operating Environment 5
1.4 Reference Document 6

2. Preparation 7
2.1 Install QE for OTA 8
2.1.1 Using Renesas Software Installer 8
2.1.2 Download and Install QE from the Web 9
2.1.3 Installation Procedure (common) 10
2.2 Create a Cloud Account 11
2.2.1 For AWS 11
2.2.2 For Azure 11

3. Development Procedure 12

4. Operation Example 13
4.1 Preparation 13
4.2 Start QE for OTA 13
4.3 Cloud Settings 14
4.3.1 Sign-in to Cloud 14
4.3.1.1 For AWS 14
4.3.1.2 For Azure 15
4.4 Project Preparation 19
4.4.1 Project Selection 19
4.4.1.1 For AWS 19
4.4.1.2 For Azure 30

4.4.1.3 Download Smart Configurator Components 38
4.4.2 Select Provisioning 39
4.5 Manage IoT Device 40
4.5.1 Manage IoT Device 40
4.5.2 Create Initial Firmware 42
4.5.2.1 Confirm Toolchain Version 42
4.5.2.2 Create Initial Firmware 43
4.5.2.3 Sample Project Firmware 45
4.5.2.4 Code Signing Certificate 46
4.5.3 Write Program to IoT Device 47
4.6 OTA 50
4.6.1 Create Update Firmware 50
4.6.2 Execute OTA and Check Status 52

5. Help Function 56

Revision History 57

1. Overview

1.1 System Overview

QE for OTA is a solution toolkit that runs on the integrated development environment (IDE) e^2 studio. It is a development tool that enables easy apply Over the Air (OTA) using the AWS or Microsoft Azure cloud service. By operating QE for OTA according to the provided workflow, you can obtain cloud-related information necessary to register with your cloud system to implement OTA, embed security information into the MCU, and execute OTA.

1.2 QE for OTA Structure

QE for OTA consists of two views: the OTA Main (QE) view, which guides the implementation of OTA using the Cloud, and the OTA Manage IoT Device (QE) View, which provides windows for creating firmware, setting OTA, viewing the operating status.

1.2.1 OTA Main (QE) View

This is the main view for QE for OTA. It guides your through the workflow to implement OTA using the Cloud.

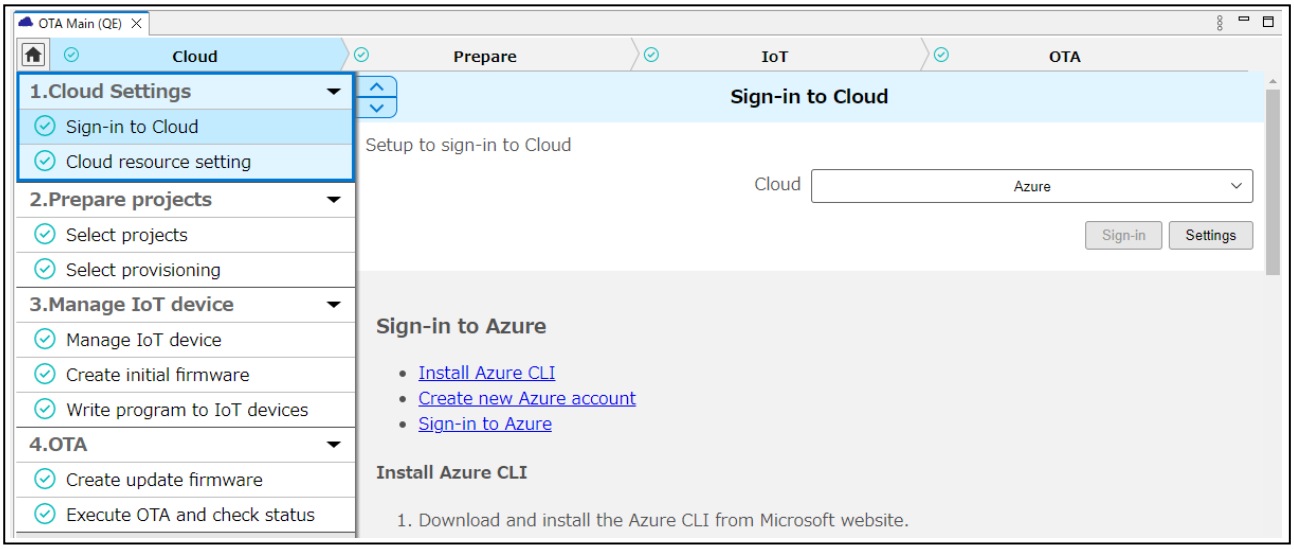

Figure 1-1 OTA Main (QE) View

- Cloud settings
  Setup to sign-in to the Cloud.
  Set up the Cloud resources needed for OTA when with Azure.
- Prepare projects
  Create OTA projects.
- Manage IoT device
  Add, delete, and view information about IoT devices. Also, create the initial firmware and write programs to IoT devices.
- OTA
  Create update firmware and execute OTA.

1.2.2 OTA Manage IoT Device (QE) View

This view consists of five functions: IoT Device, Initial Firmware, Update Firmware, OTA, and Firmware Log. Switch between functions using the tabs displayed on the right side of the view.

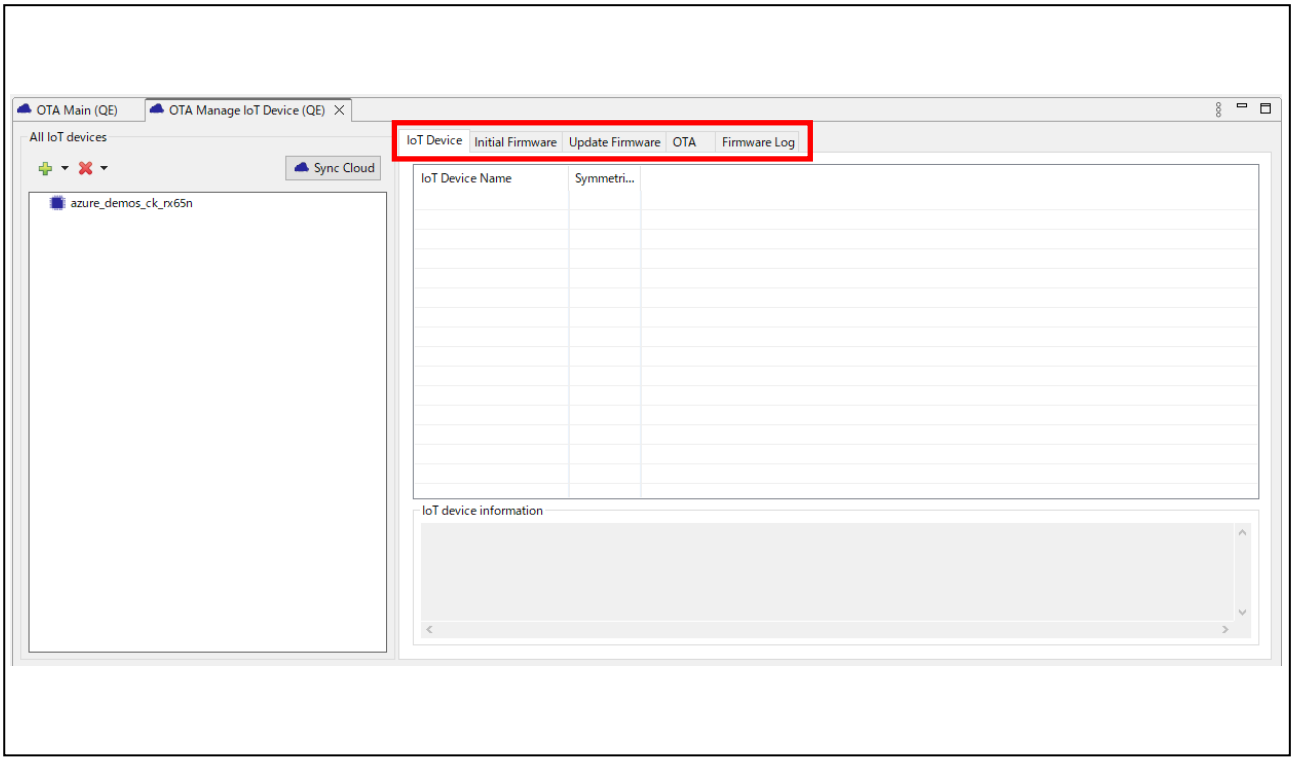

Figure 1-2 OTA Manage IoT Device (QE) View

- IoT Device
  View the device certification, security key and other IoT device information generated when IoT devices are created.
- Initial Firmware
  Embed information for each IoT device in the source code, create the initial firmware.
- Update Firmware
  Embed information for each IoT device in the source code and create update firmware.
- OTA
  Execute OTA.
- Firmware Log
  View operation log for the target board.

1.3 Operating Environment

The following table shows the operating environment of this application note.

Table 1-1 Operating Environment

Items	Contents
PC OS	Windows 10, Windows 11 (64-bit version of either)
IDE	Renesas e^2 studio 2023-04 version or later
Tool chain	CC-RX V3.04 or later / GCC for Renesas RX 8.3.0.202104 or later
QE	QE for OTA V1.1.0 or later
Target device	RX65N
Evaluation board	CK-RX65N (including RYZ014A PMOD bundled in kit)
SIM card	micro SIM for cellular LTE communications (included with CK-RX65N)
RX Driver Package (RDP)	AWS: V1.34 / Azure: V1.37 (Note 1)
OpenSSL	V3.0.4 Light or later (Note 2)
Renesas Flash Programmer	V3.09 or later (Note 2)
FreeRTOS	v202107.00-rx-1.0.1
Azure RTOS	6.2.1_rel-rx-1.0.1
Azure CLI	V2.49.0 or later (Note 2)

Note 1. This is the version used by the RTOS utilized in this application note. The version that corresponds to the RTSO used in the target project must be installed.
2. This is not included with e^2 studio or QE, and it must be installed separately.

In this application note, The sample project using AWS operates with cellular communications using the RYZ014A board. The sample project using Azure operates with Ethernet.

The RYZ014A board's power supply ("USB Cable - Power Supply" in the figure below) can be used with either the power supply from the PC's USB port or the USB power supply connected to the outlet.

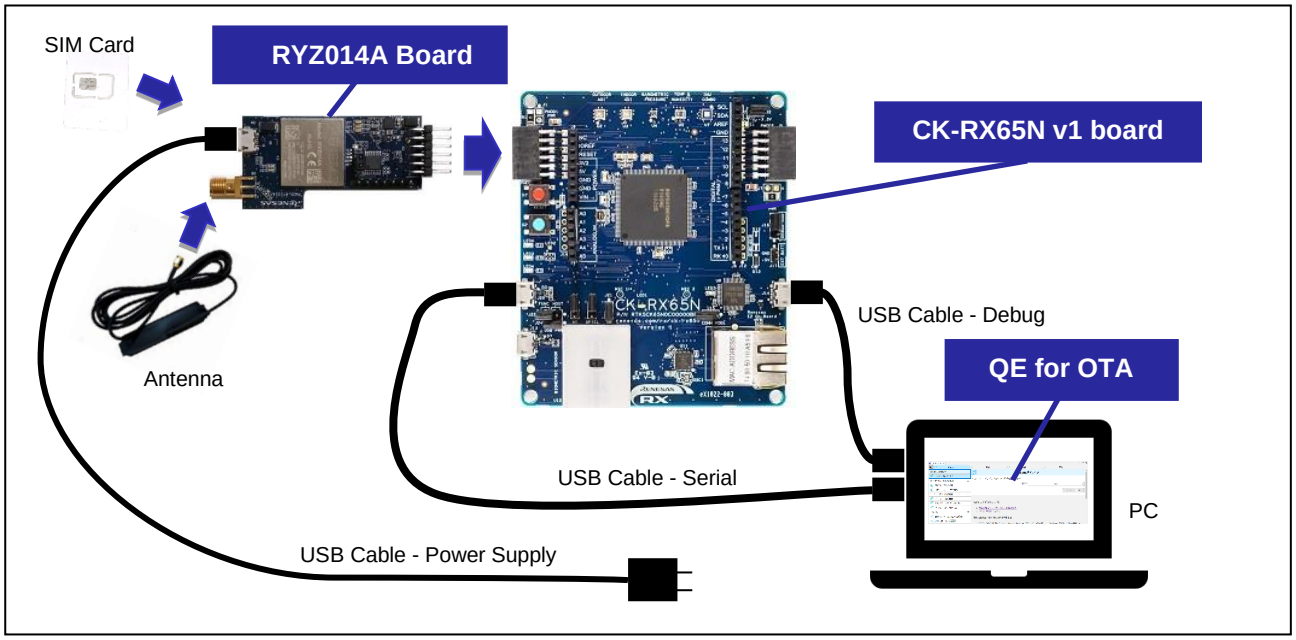

Figure 1-3 Operating Environment Connection Diagram for Cellular Communications

The following describes the configuration for connecting a device when using Ethernet. Use an Internet-ready LAN cable to connect to the Ethernet port on CK-RX65N.

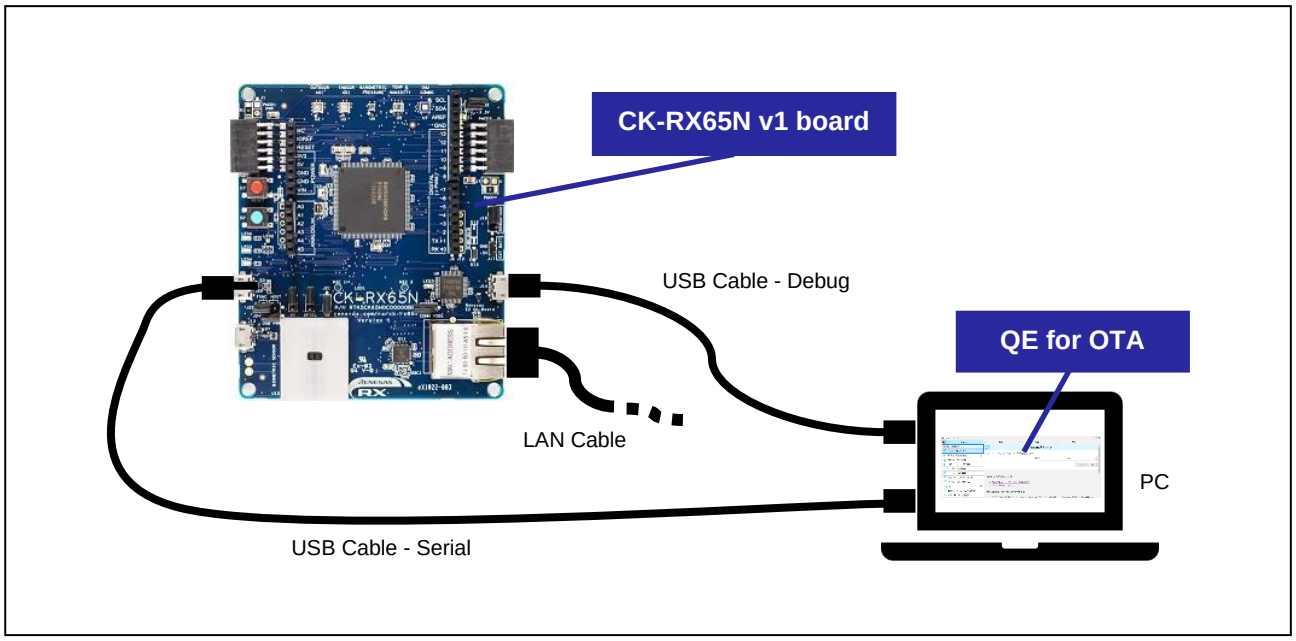

Figure 1-4 Operating Environment Connection Diagram for Ethernet Communications

1.4 Reference Document

- R01UH0590
  RX65N Group, RX651 Group User's Manual: Hardware
- R20UT5100
  RX65N Group Cloud Kit for RX65N Microcontroller Group CK-RX65N v1 User's Manual
- R01AN5549
  RX Family How to implement FreeRTOS OTA by using Amazon Web Services on RX65N
- R01AN6357
  Creating a Microsoft Azure ADU Environment
- R01AN5548
  Renesas MCU Firmware Update Design Policy
- R19DS0111
  RYZ014A LTE Category M1 Module

2. Preparation

The preparations required for OTA installation are as follows.

- Install e^2 studio
- Apply the following tools to e^2 studio
  - Toolchain
  - QE for OTA
- Install Renesas Flash Programmer
- Connect to evaluation board
- Create AWS/Azure Account
- Install OpenSSL
- Install Azure CLI (when using Azure)

This chapter describes how to install QE for OTA and how to create an AWS/Azure account.

For details regarding how to install OpenSSL, refer to the OpenSSL website.

2.1 Install QE for OTA

2.1.1 Using Renesas Software Installer

This section explains how to install QE from the Renesas Software Installer in e² studio.

Select Renesas Software Installer from the e² studio menu. After the Renesas Software Installer window appears, select “Renesas QE” and click “Next >”.

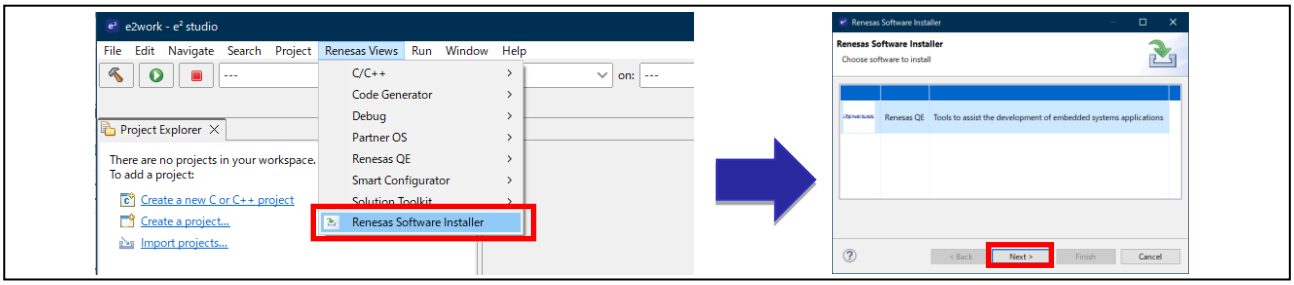

Figure 2-1 Renesas Software Installer

When the “Install Extensions” window appears, check the box in front of “QE for OTA” and click “Finish”.

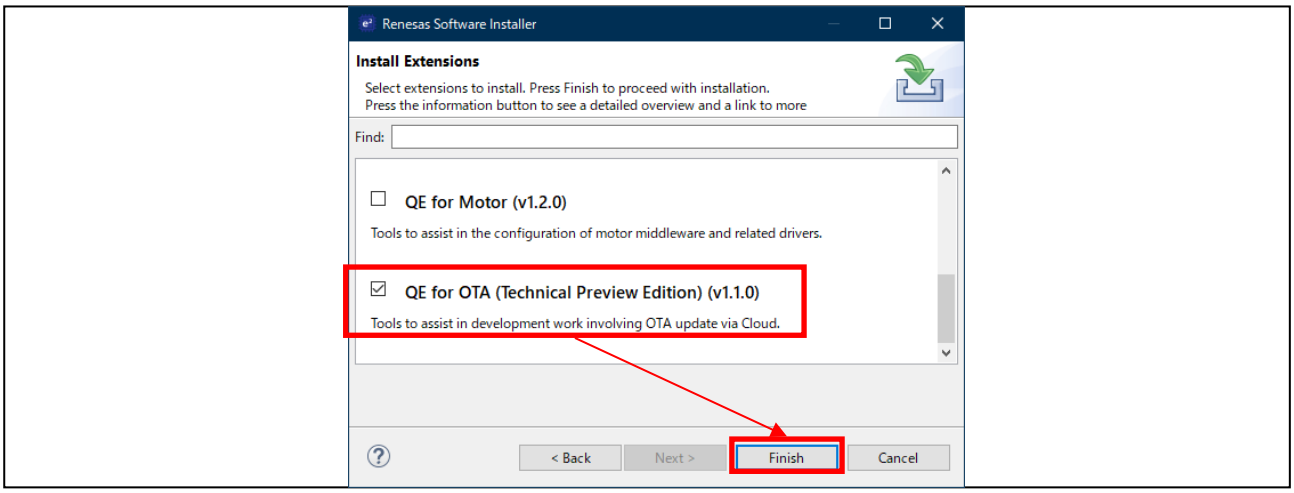

Figure 2-2 Install Extensions

Check “Renesas QE for OTA” as the item you want to install and click “Next >”.

Check to make sure the item to be installed is “Renesas QE for OTA” and click “Finish.”

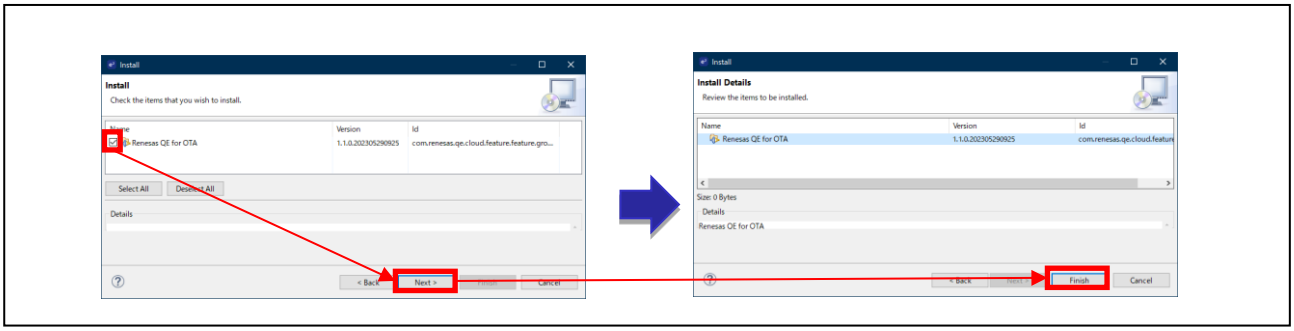

Figure 2-3 Install QE for OTA

For further details, refer to 2.1.3 Installation Procedure (common).

2.1.2 Download and Install QE from the Web

Select “Help” → “Install New Software” menu from the e² studio to open the “Install” dialog.

Next, click “Add…” to open the “Add Repository” dialog.

Click “Archive…” and select the target installation file (zip file) in the file selection dialog, and then click “Open”. Click “Add” at the bottom of the “Add Repository” dialog.

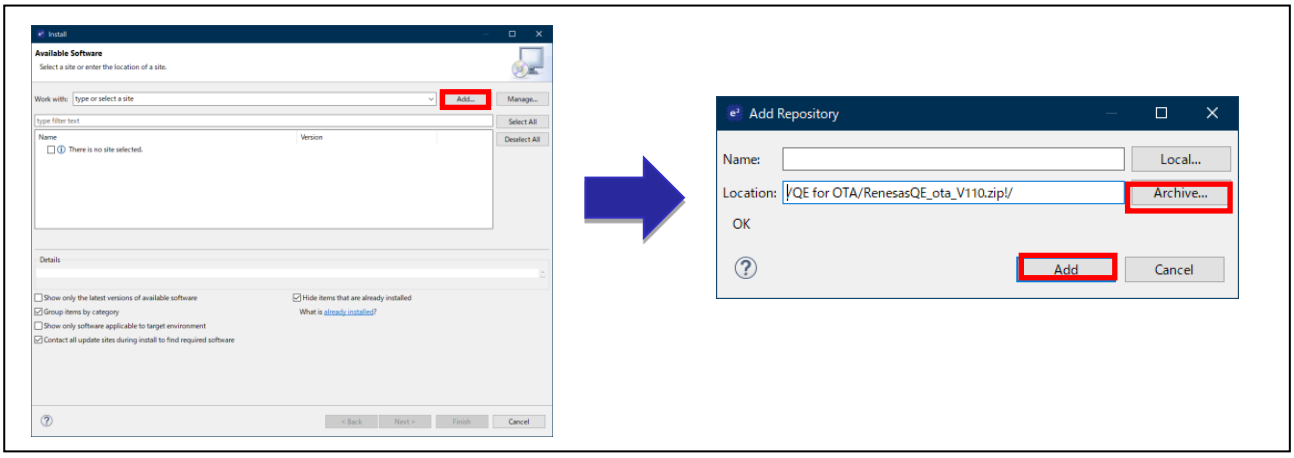

Figure 2-4 Download and Install QE from the Web 1

Expand “Renesas QE” displayed in the “Install” dialog, check the box(es) for the displayed software, and then click “Next >”. To shorten the installation time, remove the check from for “Contact all update sites during install to find required software”.

Confirm that the installation target is correct, and then click “Finish”.

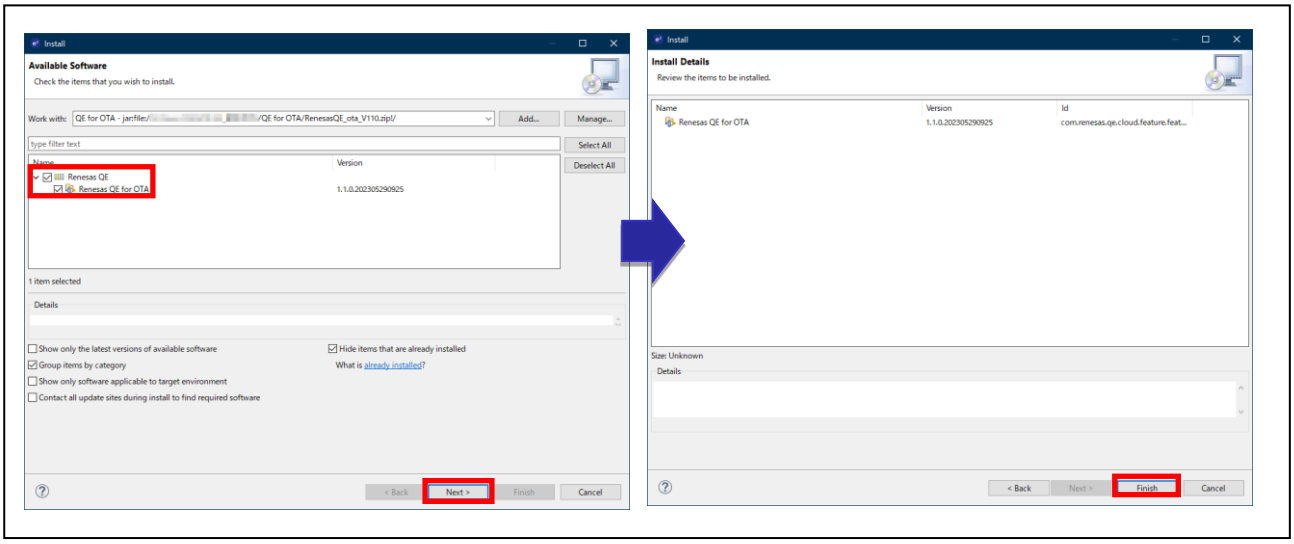

Figure 2-5 Download and Install QE from the Web 2

For further details, refer to 2.1.3 Installation Procedure (common).

2.1.3 Installation Procedure (common)

If the following dialog appears, set the unsigned content as a trusted tool and the installation will be executed.

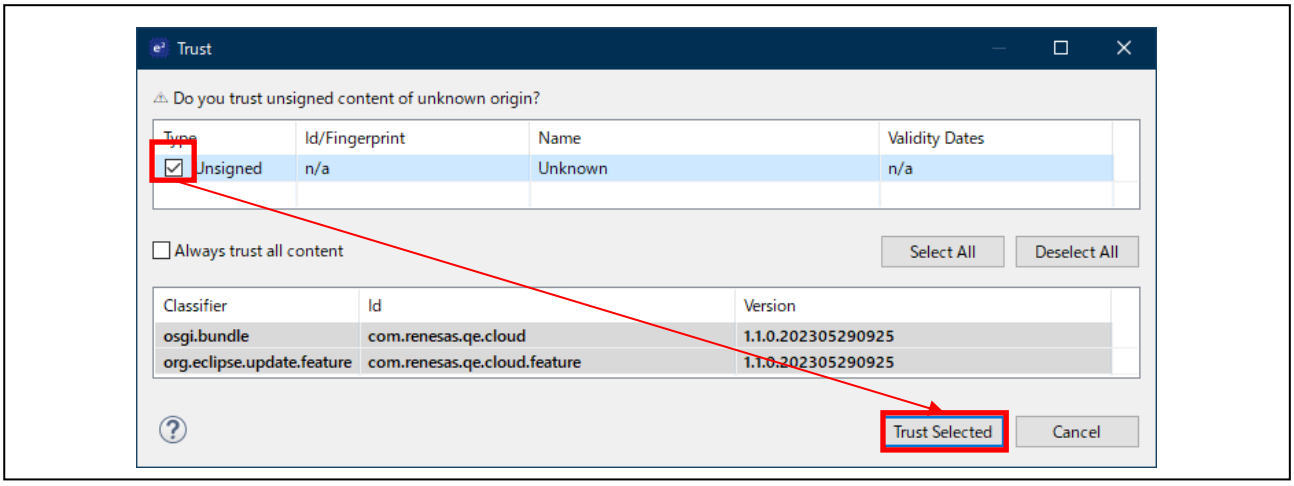

Figure 2-6 Trusted Tool

The installation status of QE for OTA is displayed at the bottom of e² studio. Installation is complete when you are asked to restart e² studio.

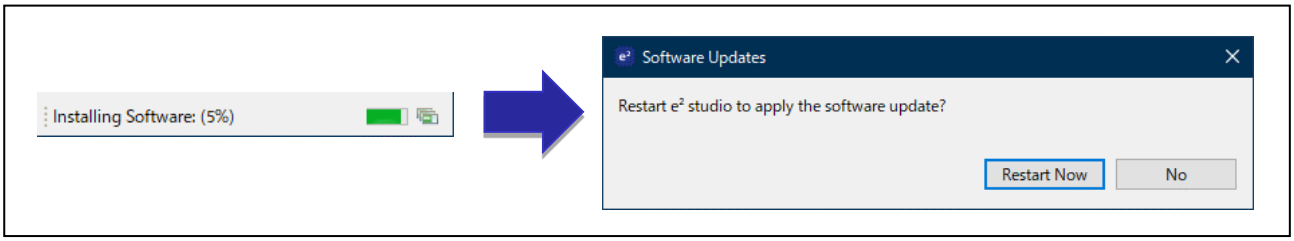

Figure 2-7 Installation Status

2.2 Create a Cloud Account

This section describes how to create an account for AWS and Azure. For further details, refer to the corresponding service guide.

2.2.1 For AWS

Preparation for using AWS is described below.

1. Access the AWS website and created an AWS account.
2. Sign-in to the AWS console from your WEB browser and create an IAM user entity to execute OTA. Make sure you add full access permissions (IAMFullAccess) to all IAM services when creating your IAM user entity.
3. Open the security credentials in the AWS console and create an access key in the AWS IAM credential information settings. This will generate an access key and a secret access key.

2.2.2 For Azure

Preparation for using Azure is described below.

1. Access the Azure website and create an Azure account.
2. Sign-in to your Azure account from your WEB browser and confirm your subscription ID.
3. Download and install Azure CLI from the Microsoft website.
   After Azure CLI is installed, the first time QE for OTA is started up from your browser, you will be asked to sign into Azure. Input your account information and sign-in.

3. Development Procedure

By proceeding according to the workflow of QE for OTA, you can create the initial firmware and execute OTA.

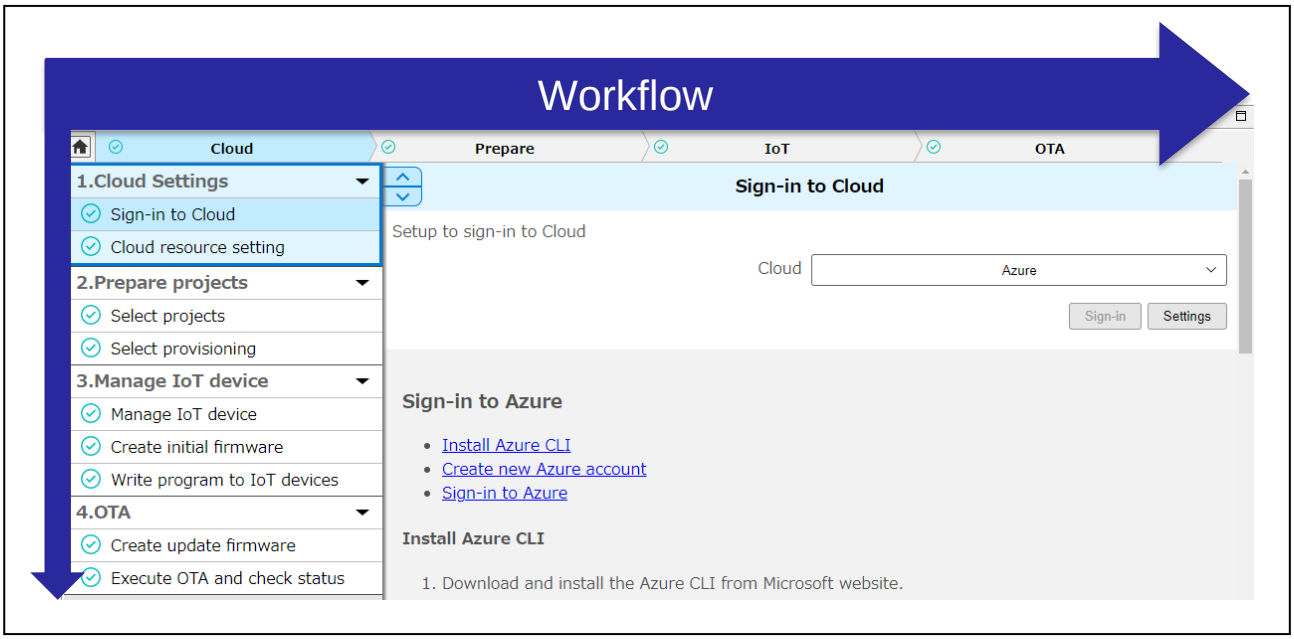

Figure 3-1 OTA Main (QE) View

The items and procedures are listed in the table below.

Table 3-1 QE for OTA Items and Contents

Item		Contents	
Cloud Settings	Sign-in to cloud	Setup to sign-in to cloud, and sign-in	4.3.1
	Cloud resource setting	Set resources (Azure only)	4.3.1
Prepare Projects	Select projects	Create sample project and import	4.4.1
	Select provisioning	Confirm provisioning	4.4.2
Manage IoT device	Manage IoT device	Register and manage IoT device	4.5.1
	Create initial firmware	Create and build initial firmware	4.5.2
	Write program to IoT device	Write initial firmware	4.5.3
OTA	Create update firmware	Create and build firmware to be used by OTA	4.6.1
	Execute OTA and check status	Execute OTA and check status	4.6.2

4. Operation Example

This chapter describes the workflow for executing OTA using QE for OTA.

Although AWS (FreeRTOS) and Azure (Azure RTOS) differ in some areas, unless the difference is major, this description uses AWS as the example. Please apply the contents to Azure accordingly.

The following operations are performed.

- Write the initial firmware and debug firmware V1.0.0 operations.
- After writing the update firmware by executing OTA, switch to V1.1.0 operations using bank swap.

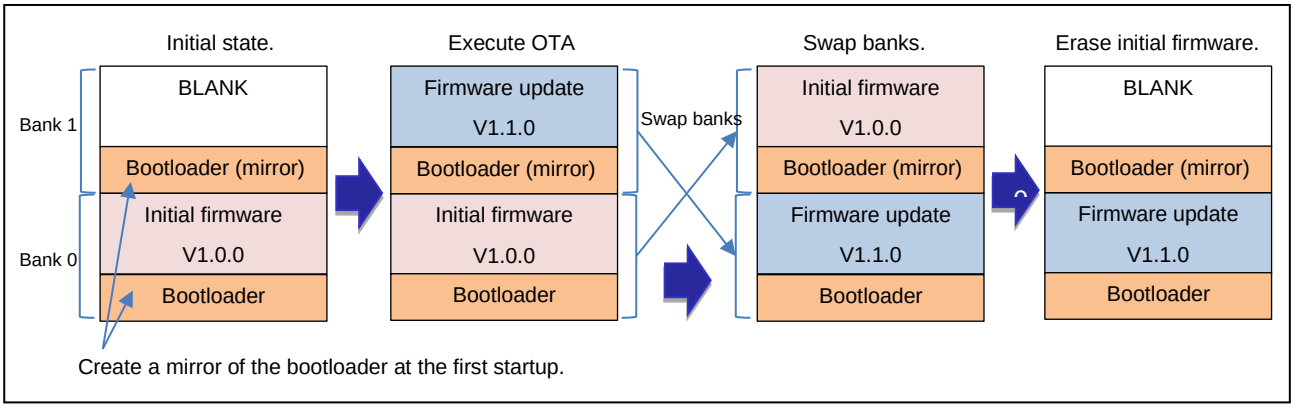

Figure 4-1 OTA Image

4.1 Preparation

Start e^2 studio. Prepare the following folders in any location.

- e^2 studio workspace folder
- RTOS module download folder
- Project folder

This application note uses the following workspace and project folders. Folders not listed here are considered optional folders.

- C:¥workspace (workspace folder)
- C:¥workspace¥freertos (for AWS FreeRTOS)
- C:¥workspace¥azure_demos (for Azure RTOS ADU projects)
- C:¥workspace¥azure_boot_loader (for Azure RTOS bootloader project)
- C:¥afr (FreeRTOS module download folder)

4.2 Start QE for OTA

Select OTA Main (QE) from the e^2 studio menu to display the OTA Main (QE) view as shown below.

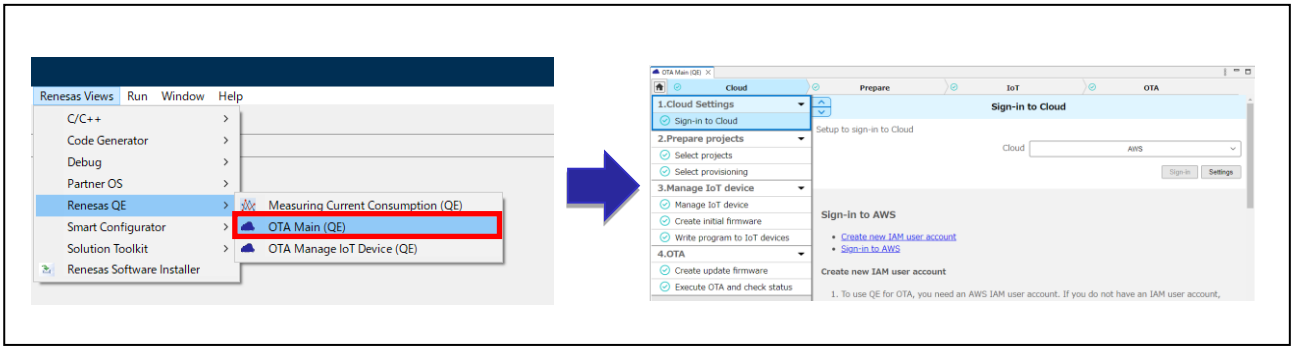

Figure 4-2 Select OTA Main (QE)

4.3 Cloud Settings

This section describes how to sign-in to the cloud. When using Azure, it is also necessary to set resources accordingly.

4.3.1 Sign-in to Cloud

Select “Sign-in to Cloud” from the menu displayed on the left side of the OTA Main QE view.

Select the cloud connection, and then click “Settings” to open the sign-in dialog.

Figure 4-3 Sign-in to Cloud

4.3.1.1 For AWS

Enter your IAM user Access Key ID and Secret Access Key, and then click “Sign-in”.

If the IAM user does not have the required permissions to execute OTA, a warning icon will appear next to “Add required permissions for OTA” (right side of dialog). In this case, click “Add required permissions for OTA” and the required permissions will be added automatically.

In the Region box, select the region in which you will be using the AWS services. If the upload region and current region differ, OTA cannot be executed. Make sure you select the same region for both upload and current region.

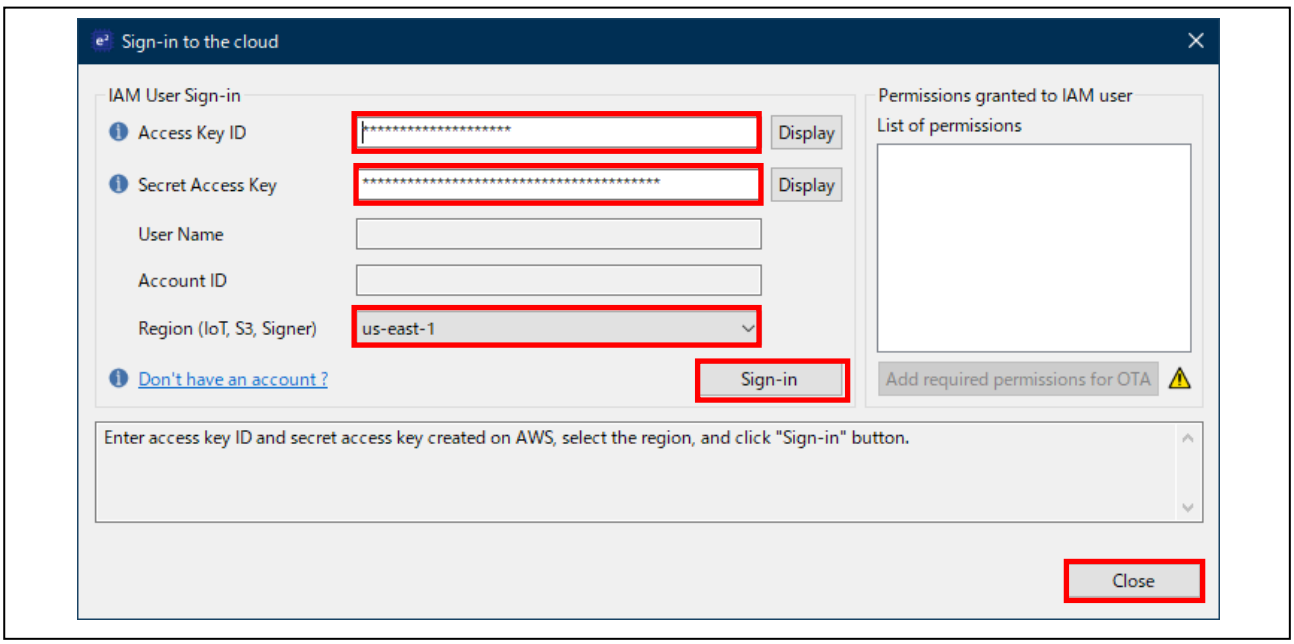

Figure 4-4 Sign-in to Cloud (AWS)

After signing in, make sure that you have received the “Signed in to cloud.” Then, click “Close” to close the dialog.

4.3.1.2 For Azure

(1) Sign-in

(a) To create an application

Enter the application name and subscription ID, and then click “Register Application”. Take note of the following:

- Subscription owner rights are required.
- An application name that is already registered will be rejected.
- You will not be able to create an application if you have not installed Azure CLI.

Set your Client ID, Client Secret, and Tenant ID, and then click “Sign-in”.

If you do not have the required permissions to execute OTA, a warning icon will appear. In this case, click “Add required permissions for OTA” and the required permissions will be added automatically.

After signing in, make sure that you have received the “Signed in to cloud.” Then, click “Close” to close the dialog.

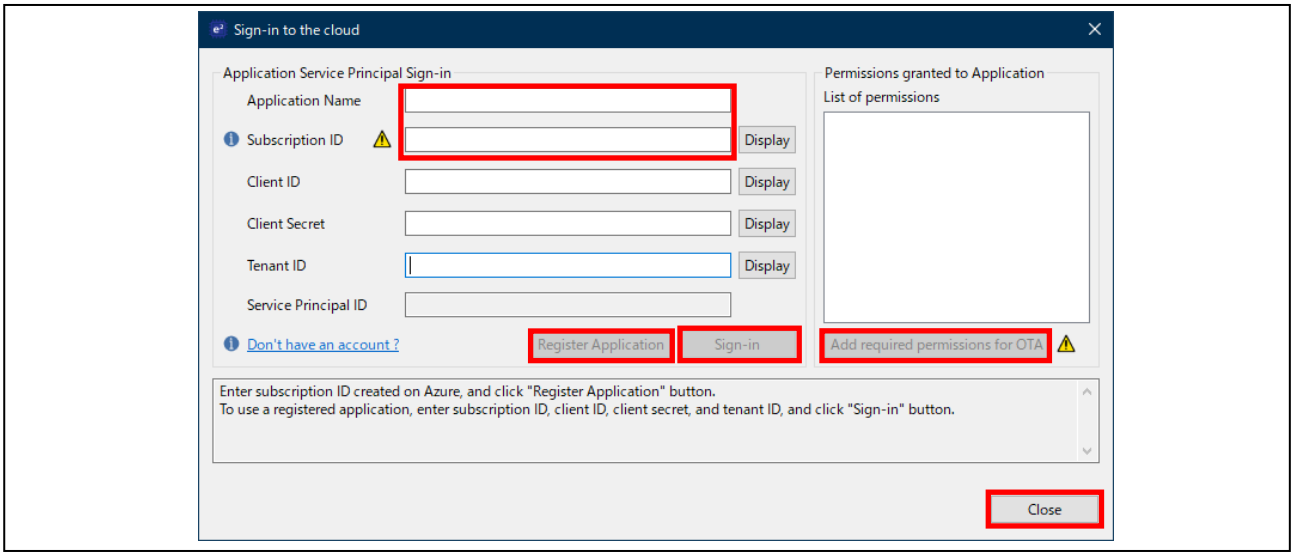

Figure 4-5 Sign-in to Cloud (Azure)

(b) To use a registered application

If you application is already registered, enter your subscription ID, client ID, client secret, and tenant ID, and then click “Sign-in”.

After signing in, click “Close” to close the dialog.

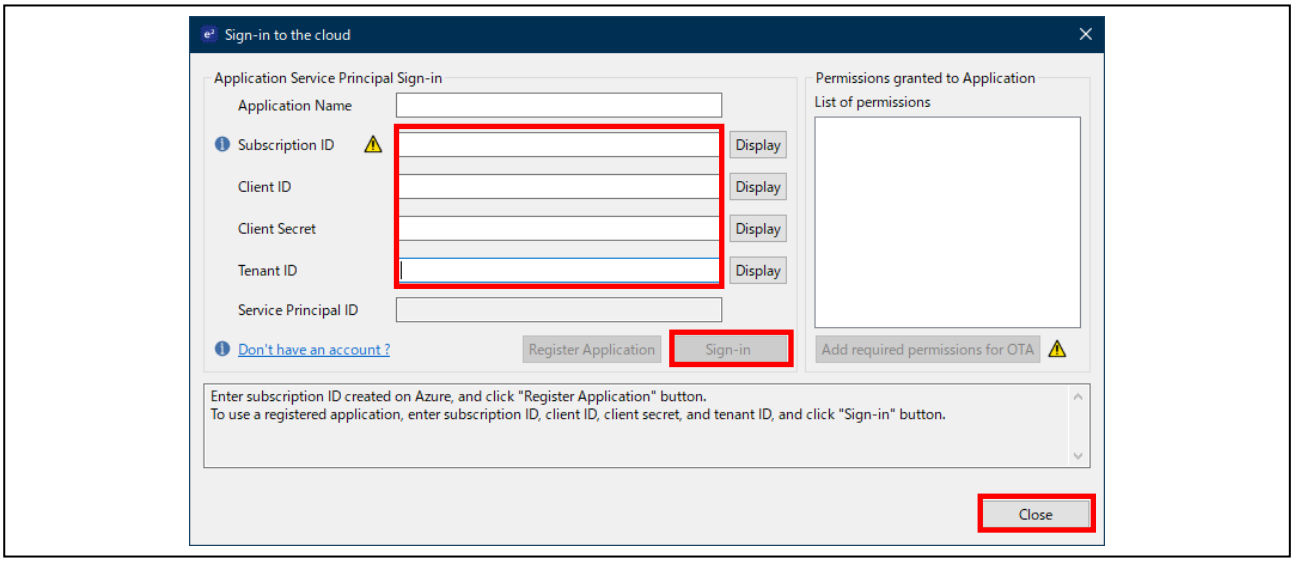

Figure 4-6 Sign-in to Cloud Using Registered Application (Azure)

(2) Cloud resource setting

To use Azure, you will need to set cloud resources.

Select “Cloud resource setting” from the menu displayed on the left side of the OTA Main (QE) view.

If you don’t see the resource group created by QE on the Azure side (qe_ota_resource_group), click the “Create” button that appears next to the “Setting” button. If there is no “Create”, click “Setting”.

If “qe_ota_resource_group” has already been created, select it as the resource group.

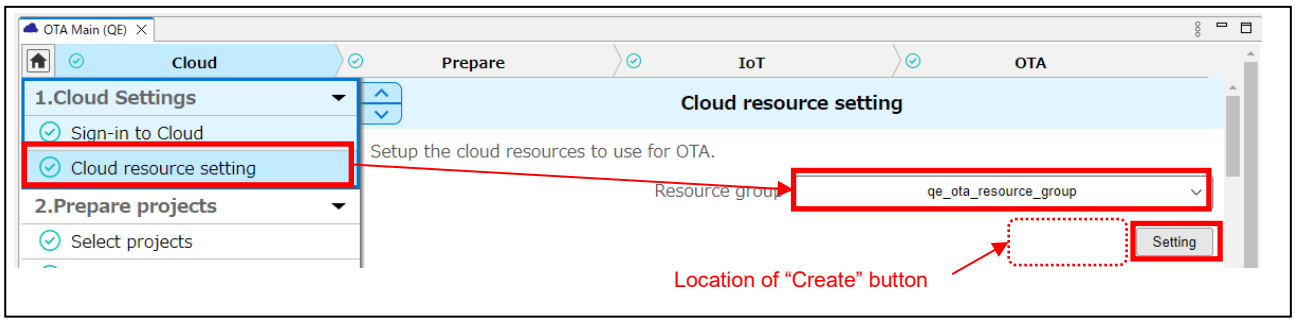

Figure 4-7 Cloud Resource Setting (Azure)

When the “Resource settings” dialog opens, if you do not see resource group “qe_ota_resource_group”, click “Create All”, and proceed to create new resources.

To set up resources individually, set each item of each resource separately.

Information corresponding to the current settings are displayed in the middle column of the dialog. Check the information here to make sure every resource is set correctly.

Also, if a warning icon is displayed for any of the resources, click “Add required permissions for OTA” to add permissions.

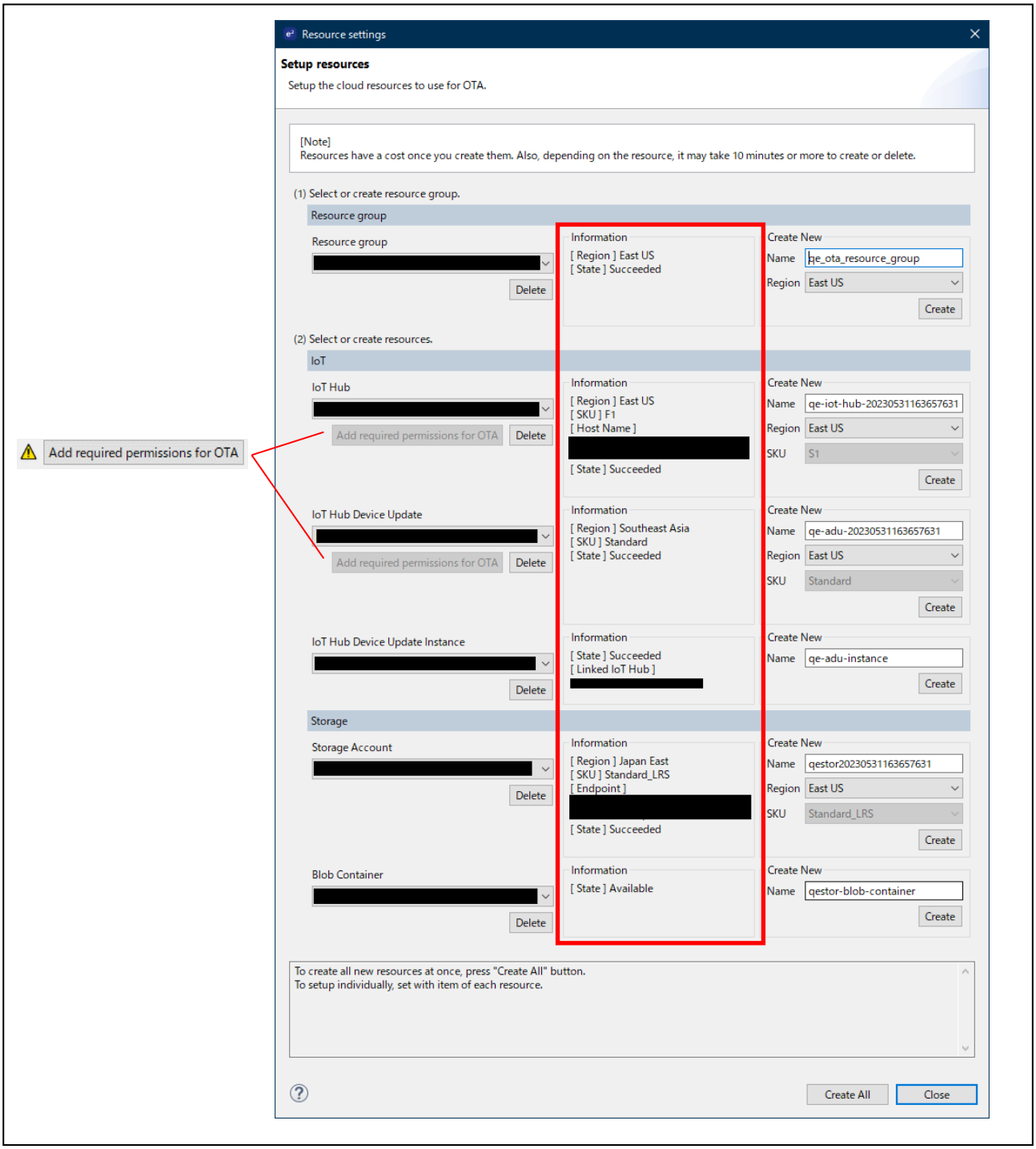

Figure 4-8 Resource Settings (Azure)

4.4 Project Preparation

4.4.1 Project Selection

This section describes how to prepare a sample OTA project. The procedure differs for AWS and Azure.

4.4.1.1 For AWS

(1) Import projects

Select “Select projects” from the menu displayed on the left side of the OTA Main (QE) view and click “Import”. When using existing projects, skip the import step and select projects from the dropdown.

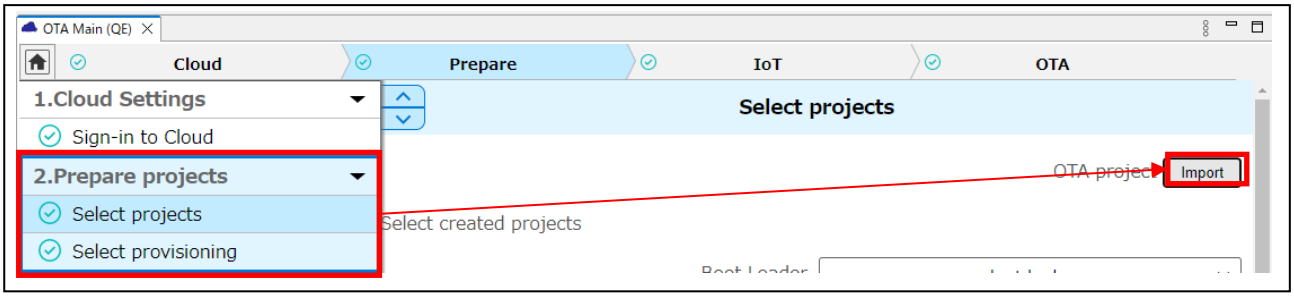

Figure 4-9 Project Selection (AWS)

Click “Browse…” and specify the folder for copying the RTOS (the folder prepared in 4.1).

Next, set the RTOS version.

If a list of versions is displayed, check the setting contents, and then click “Next>” (see Figure 4-12 FreeRTOS Version Selection - Next). If the target version is not listed, click “Check for more version…”.

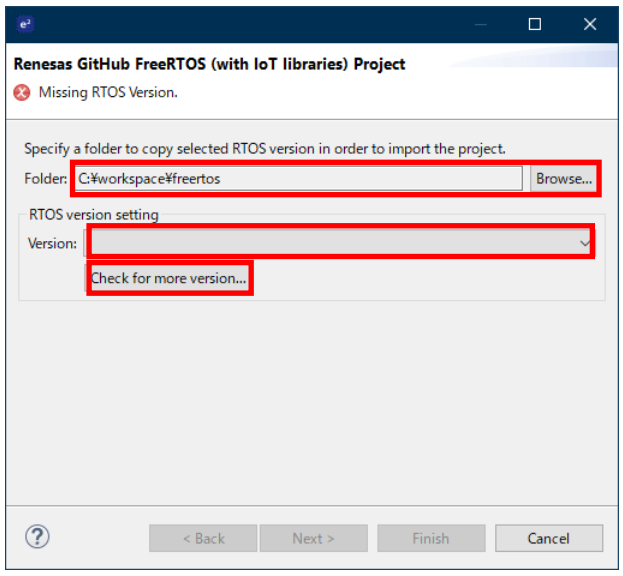

Figure 4-10 FreeRTOS Version Selection

In the next dialog, select the “Module Folder Path” (the folder prepared in 4.1) indicating the download version and download location, and then click “Download”. The disclaimer dialog with appear. Check the contents and click “Accept”.

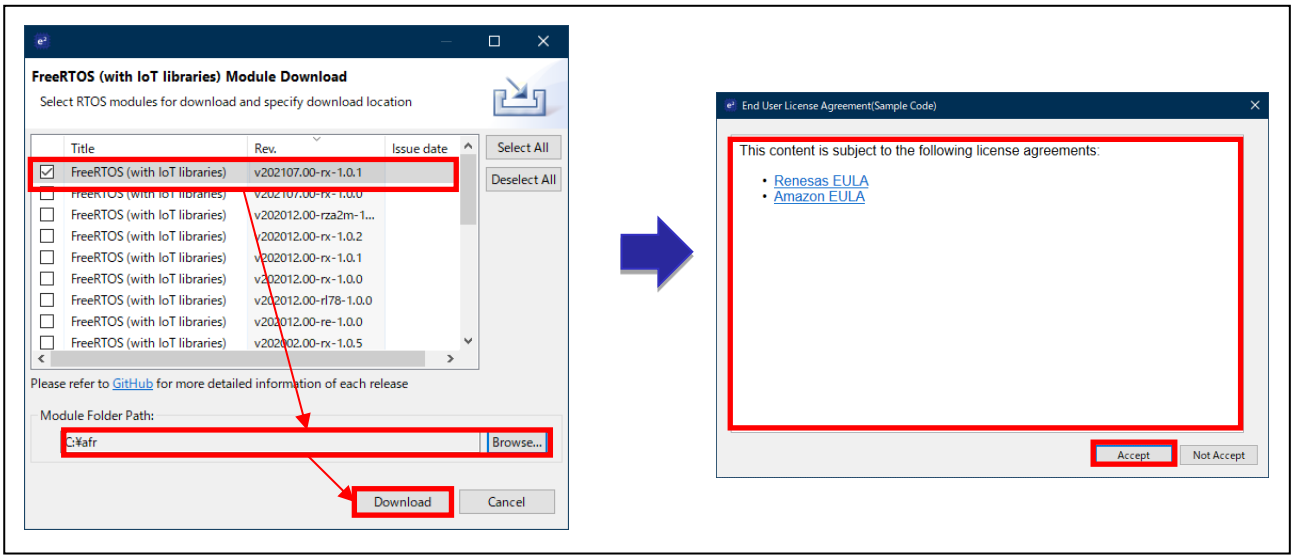

Figure 4-11 FreeRTOS Module Download

For details and selection of available revisions, refer to GitHub (Releases · renesas/amazon-freertos (github.com)).

Check the setting contents, and then click “Next>”.

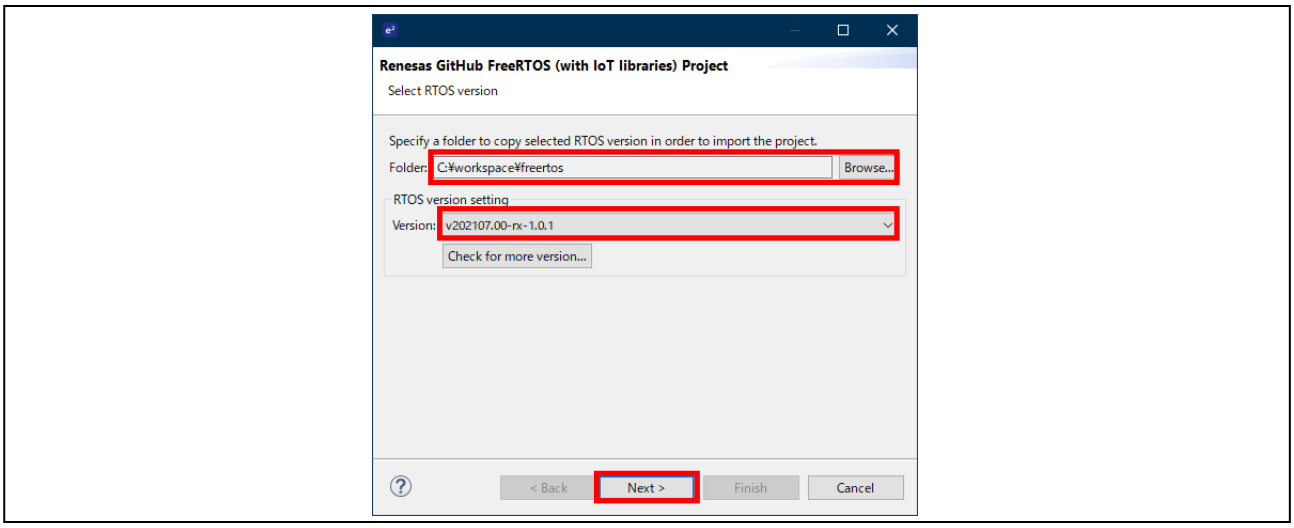

Figure 4-12 FreeRTOS Version Selection - Next

Select the projects to be imported from the list. As project selections will differ according to the environment, select the appropriate project for your operating environment.

The figure below is an example of using cellular communications with the RYZ014A board on the CK-RX65N board and with CC-RX selected as the compiler.

After selecting an aws_demos and boot_loader project, each according to the operating environment, click “Finish”.

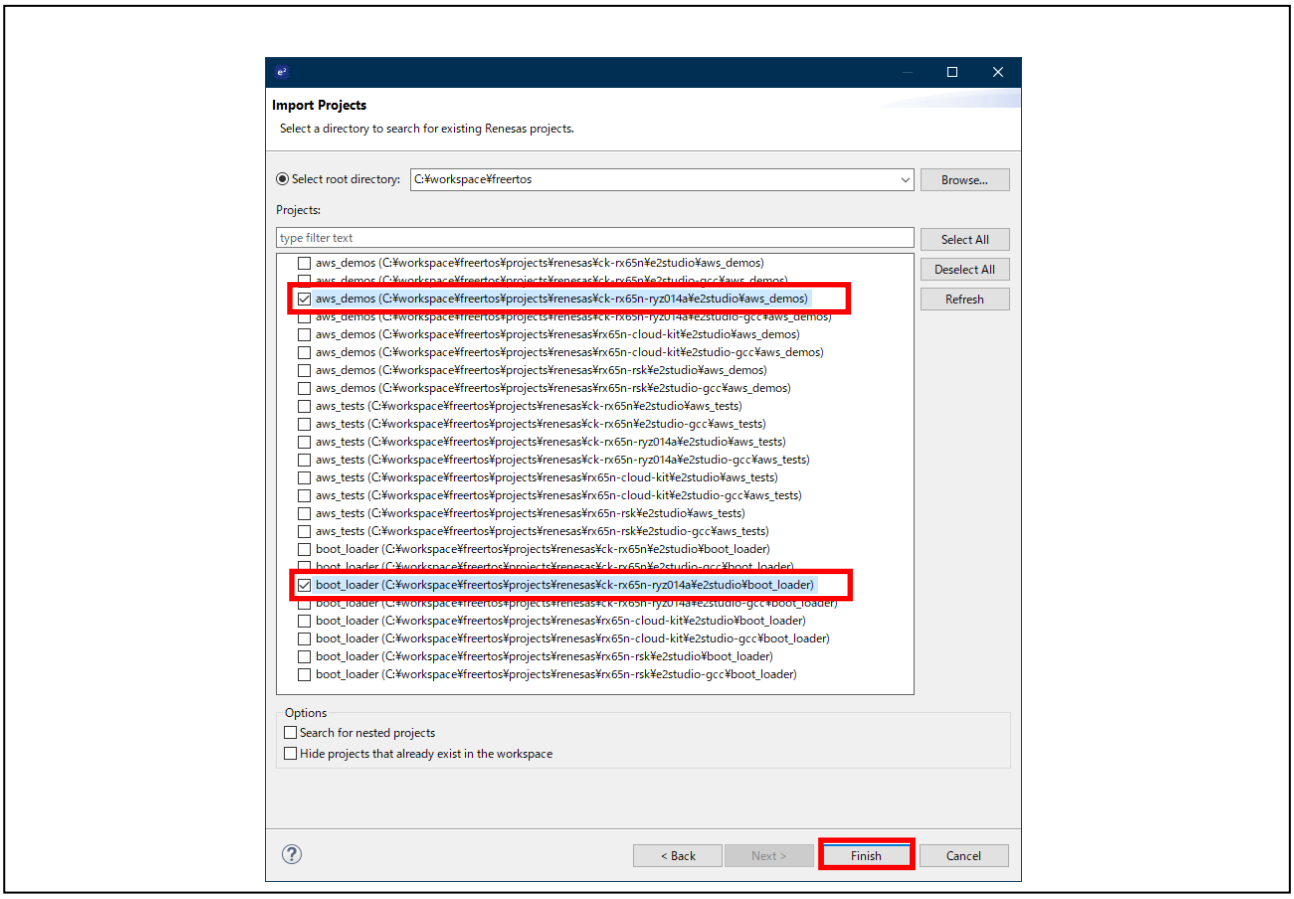

Figure 4-13 Import Project Selection

The following shows examples of project combinations.

- Using RYZ014A (CC-RX compiler):
  aws_demos (<FreeRTOS destination>¥projects¥renesas¥ck-rx65n-ryz014a¥e2studio¥aws_demos)
  boot_loader (<FreeRTOS destination>¥projects¥renesas¥ck-rx65n-ryz014a¥e2studio¥boot_loader)
- Using RYZ014A (GCC compiler):
  aws_demos (<FreeRTOS destination>¥projects¥renesas¥ck-rx65n-ryz014a¥e2studio-gcc¥aws_demos)
  boot_loader (<FreeRTOS destination>¥projects¥renesas¥ck-rx65n-ryz014a¥e2studio-gcc¥boot_loader)
- Using Ethernet (CC-RX compiler):
  aws_demos (<FreeRTOS destination>¥projects¥renesas¥ck-rx65n¥e2studio¥aws_demos)
  boot_loader (<FreeRTOS destination>¥projects¥renesas¥ck-rx65n¥e2studio¥boot_loader)
- Using Ethernet (GCC compiler):
  aws_demos (<FreeRTOS destination>¥projects¥renesas¥ck-rx65n¥e2studio-gcc¥aws_demos)
  boot_loader (<FreeRTOS destination>¥projects¥renesas¥ck-rx65n¥e2studio-gcc¥boot_loader)

Project information is automatically selected after the projects are imported, as shown in the following figure.

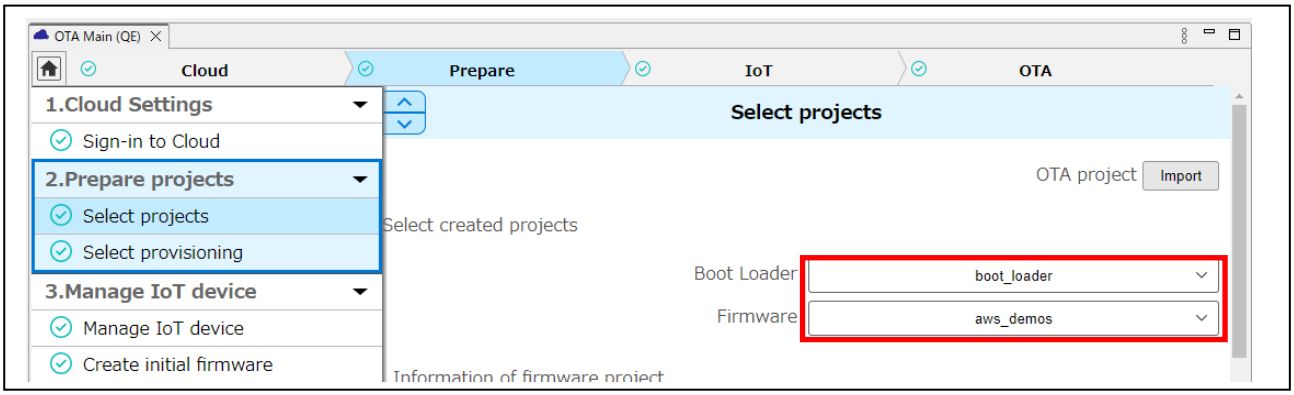

Figure 4-14 AWS: After Project Import

To reserve a 256KB-area for the boot_loader project, the aws_demos section settings need to be modified.

(a) Section settings for CC-RX

Select the aws_demos project from the project explorer and right-click to open the properties. Select “Settings”, then open “Section setting” from the button at the bottom of the figure.

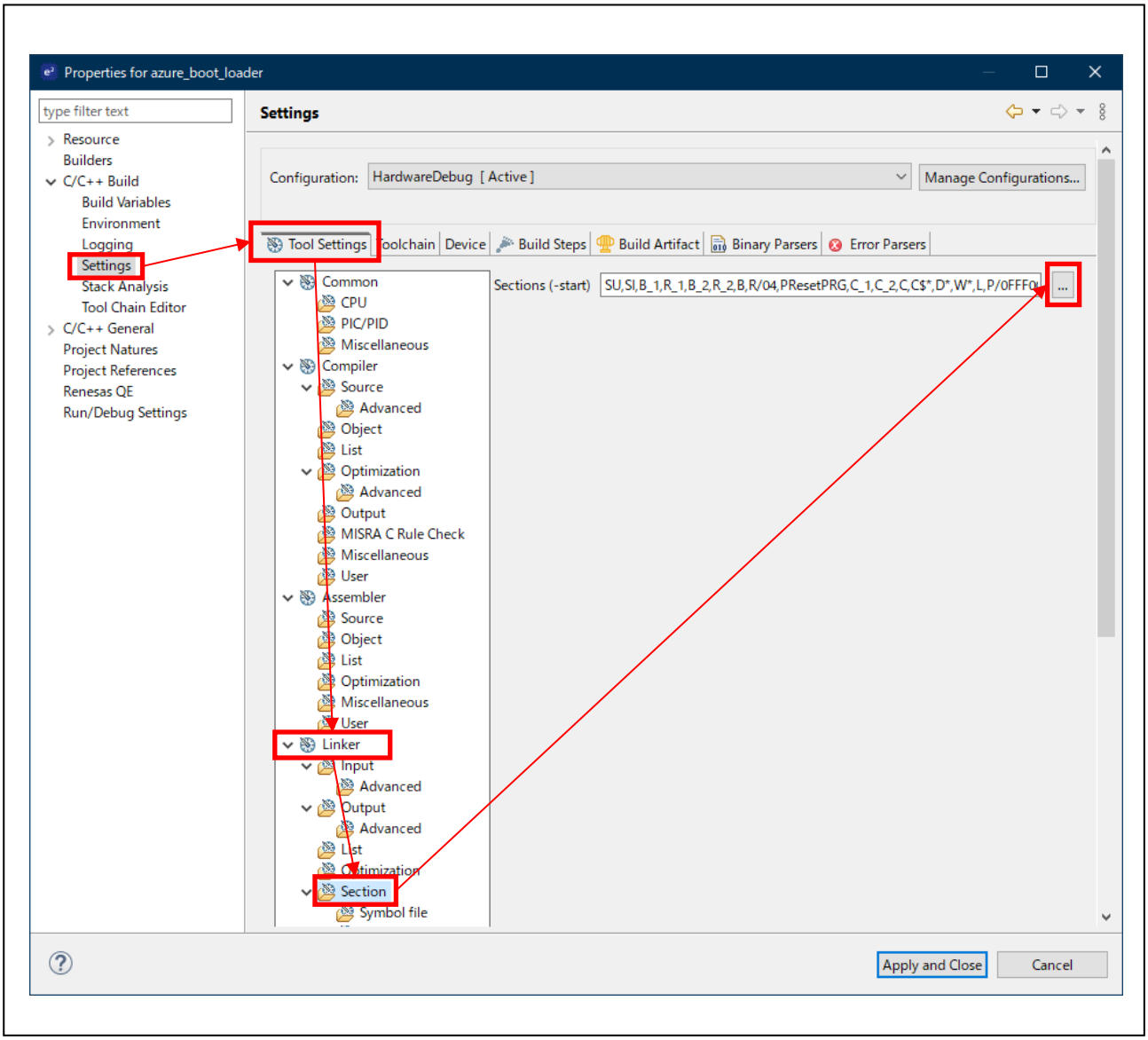

Figure 4-15 aws_demos Project Section Settings 1 (CC-RX) 1

Change the addresses of “EXCEPTVECT” and “RESETVECT” and click “OK”.

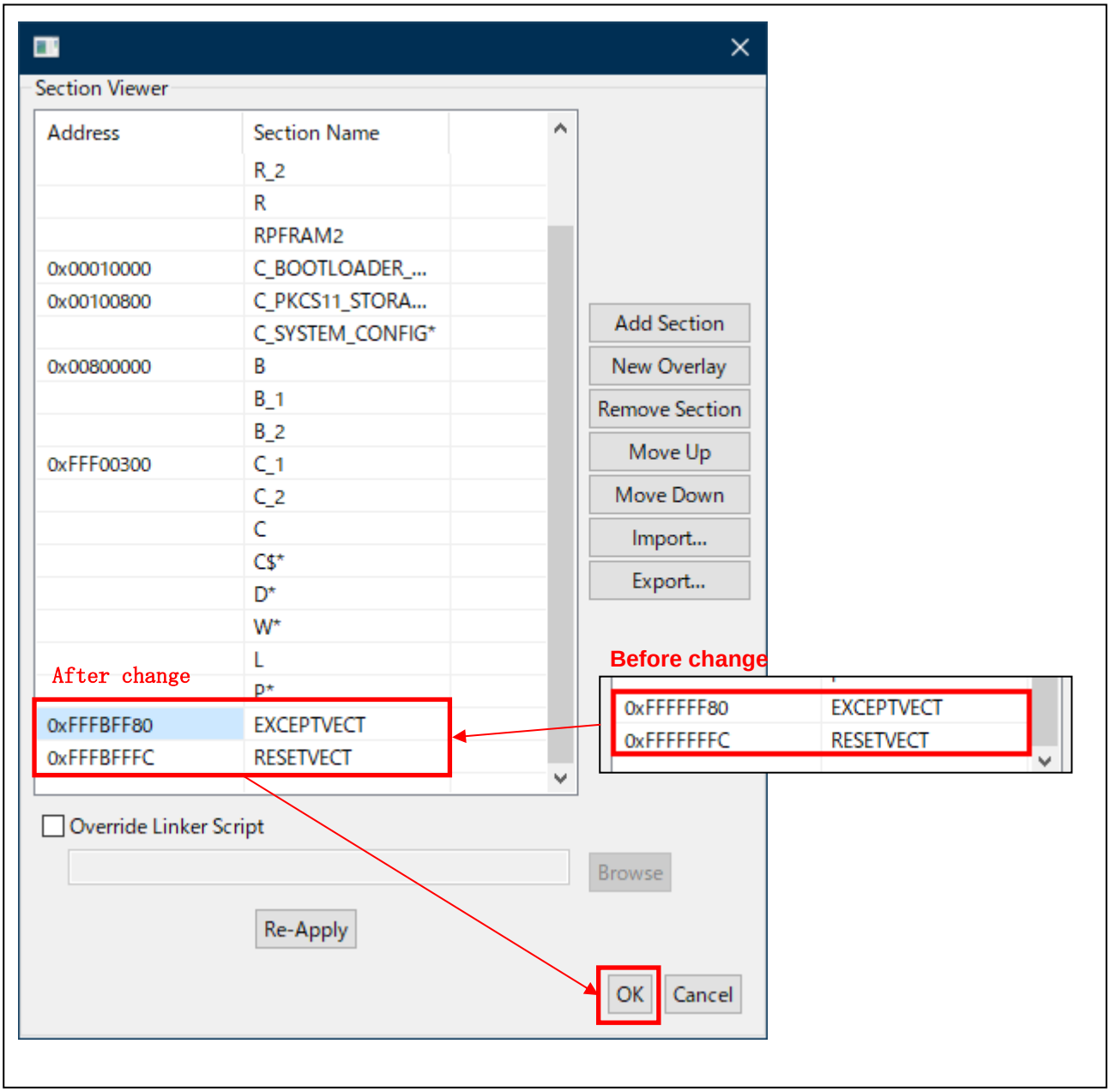

Figure 4-16 aws_demos Project Section Settings 2 (CC-RX)

Click either “Re-Apply” or “Re-Apply and Close” to apply the address changes.

(b) Section settings for GCC

For GCC, edit the “linker_script.ld” file in the aws_demos project from the project explorer.

Double-click the name of the file indicated above and select the “linker_script.ld” tab. Change the addresses of ”.exvectors” and “.fvectors” as shown in the following figure.

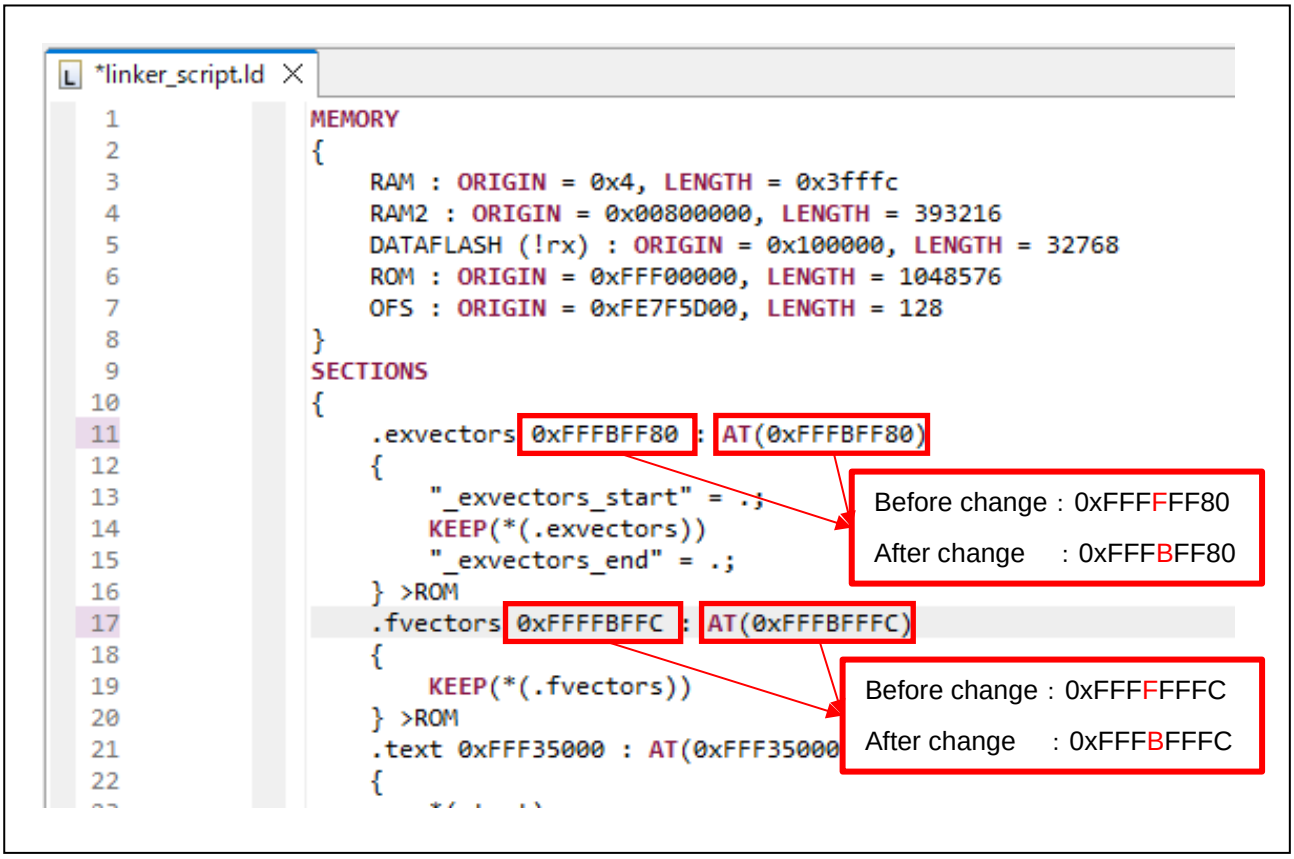

Figure 4-17 aws_demos Project Section Settings (GCC)

Error messages may appear when editing sections with the linker script editor, but please ignore them.

(2) Changing devices (when using FreeRTOS v202107.00-rx-1.0.x)

This section describes how to change a device. This procedure is necessary for both aws_demos and boot_loader projects, but the description uses the aws_demos project as an example.

Double-click “aws_demos/aws_demos.scfg” in the project explorer to open the Smart Configurator.

Select the “Board” tab to open the board selection dialog.

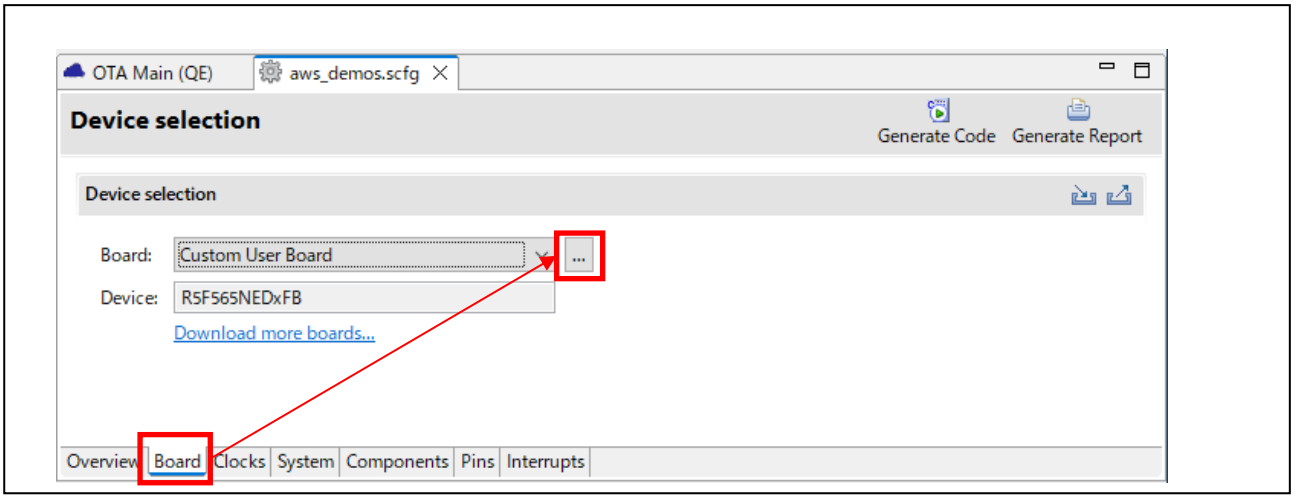

Figure 4-18 Device Change 1

Click the Browse button to view a list of target devices.

To use the configuration utilized in this application note, select “R5F565NEDxFB_DUAL” and click “OK”.

Finally, click “Next >”.

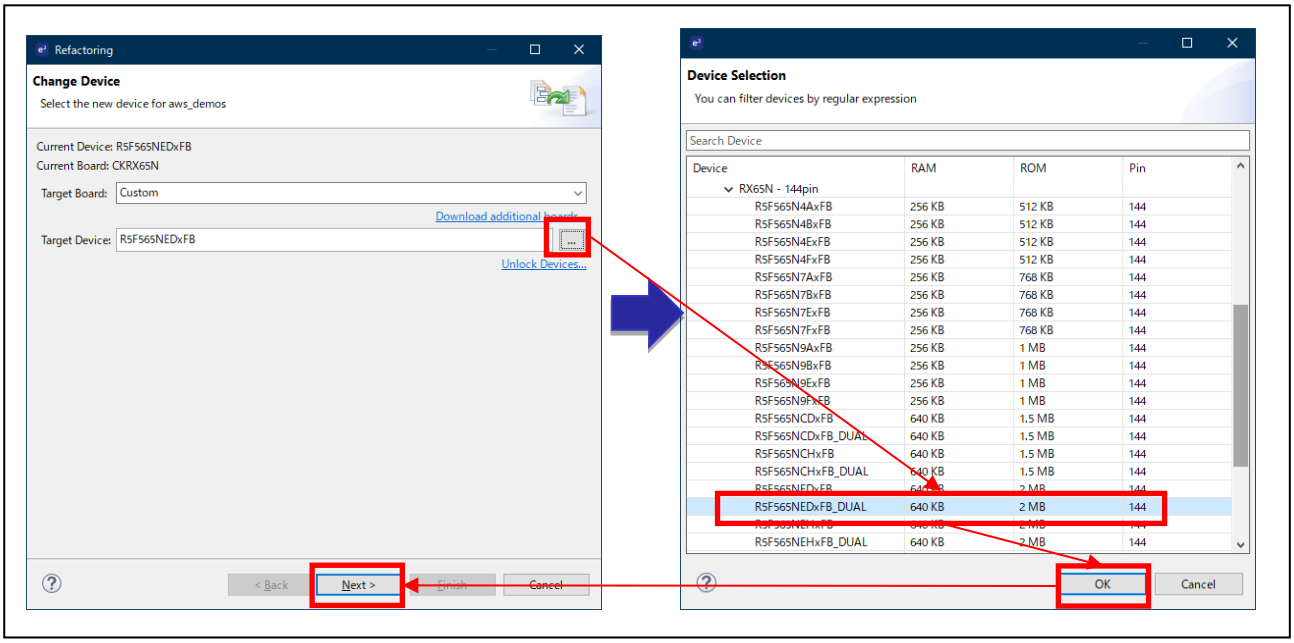

Figure 4-19 Device Change 2

When the following dialog appears, click “Next >”.

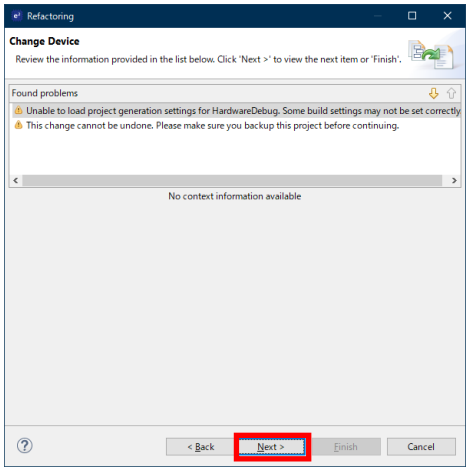

Figure 4-20 Device Change 3

(a) Device change for CC-RX

Select “Build Settings” > “HardwareDebug” > “Toolchain Settings”.

Remove the checks from “ROM to RAM mapped section (-rom)” and “Section (-start)”.

(When using the boot_loader project for RYZ014A cellular, “Section (-start) will not be listed.)

Click “Finish.”

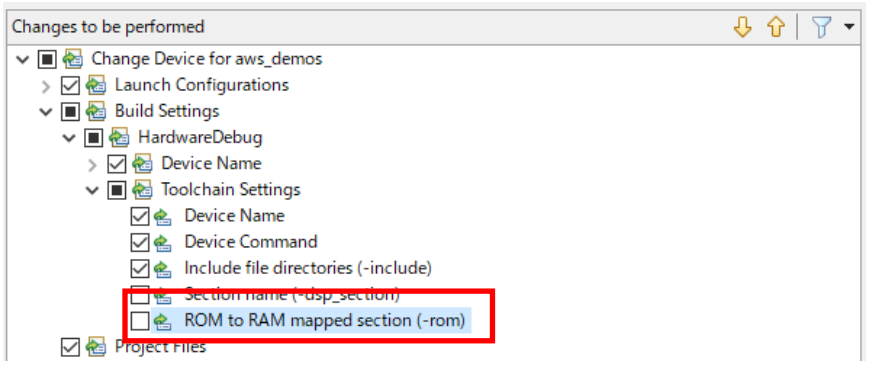

Figure 4-21 Device Change 4 (CC-RX)

(b) Device chance for GCC

Select “Project Files”.

Remove the check from “src/linker_script.ld”.

Click “Finish”.

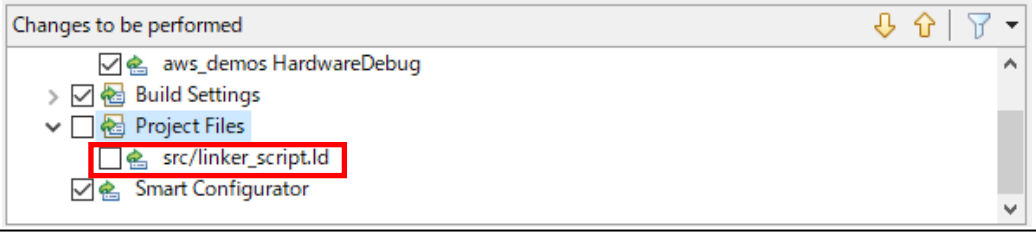

Figure 4-22 Device Change 4 (GCC)

(3) RYZ014A setting (when using RYZ014A)

When using RYZ014A, the components need to be configured.

Double-click “aws_demos/aws_demos.scfg” in the project explorer to open the Smart Configurator.

Open the “Component” dialog and select “Middleware/Generic/r_cellular.

Enter the following values in the “Configurations” area. Refer to the SIM card instructions for a description of each value.

- Access point name
- Access point login ID
- Access point password
- SIM card PIN code

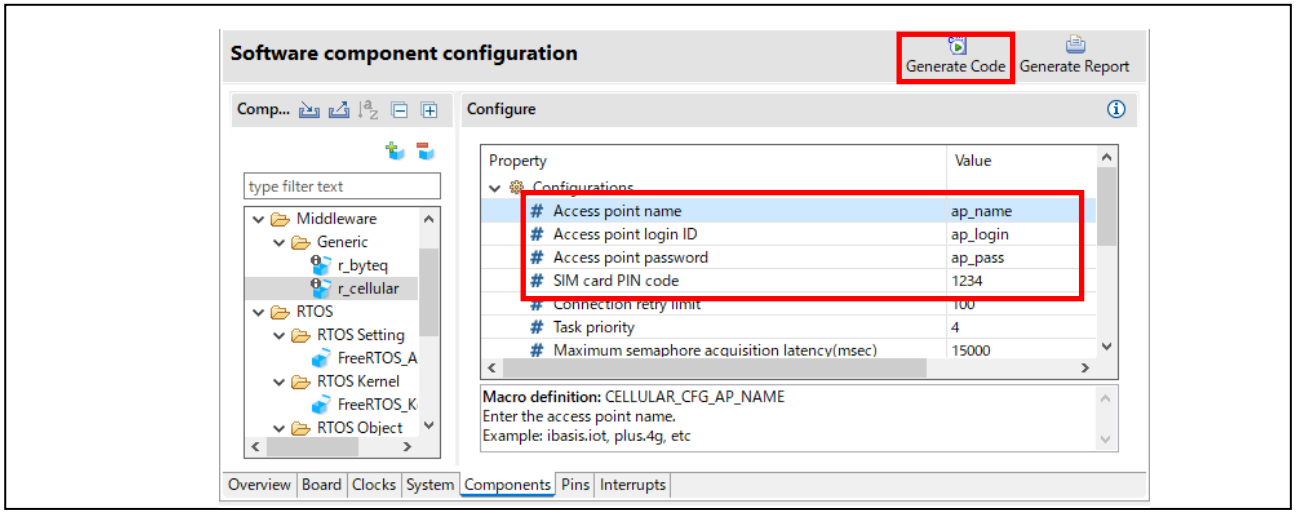

Figure 4-23 Cellular Component Configuration

If you add any changes from the Smart Configurator, click “Generate Code” to generate the code.

(4) Wi-Fi setting (when using RYZ014A)

When connecting RYZ014A to CK-RX65N, Wi-Fi will not be available. To work around the execution error (network setup failure), you can use the Wi-Fi SSID and PASSWORD to set up a dummy character string.

Open “aws_demos¥demos¥include¥aws_clientcredential.h” and set the dummy string in the next macro (“a” has been set as an example in the following figure).

- clientcredentialWIFI_SSID
- clientcredentialWIFI_PASSWORD

```
59          * @brief Wi-Fi network to join.
60          *
61          * @todo If you are using Wi-Fi, set this to your network name.
62          */
63         #define clientcredentialWIFI_SSID                    "a"
64
65         /*
66          * @brief Password needed to join Wi-Fi network.
67          * @todo If you are using WPA, set this to your network password.
68          */
69         #define clientcredentialWIFI_PASSWORD                "a"
70
```

Figure 4-24 Wi-Fi Network Dummy Setting

4.4.1.2 For Azure

(1) Create projects

This section describes how to create the ADU sample project and the bootloader project. We will start by creating the bootloader project. Except for a few differences, the applies for the ADU sample project as well. The project is created in the project folder described in section 4.1. When using existing projects, skip this step.

Select “File” > “New” > “C/C++ Project” form the e^2 studio menu.

The dialog for creating a new project will open. Select either “Renesas CC-RX C/C++ Executable Project” or “GCC for Renesas RX C/C++ Executable Project”, according to the compiler, and then click “Next >”. Enter the desired project name, and then click “Next >”

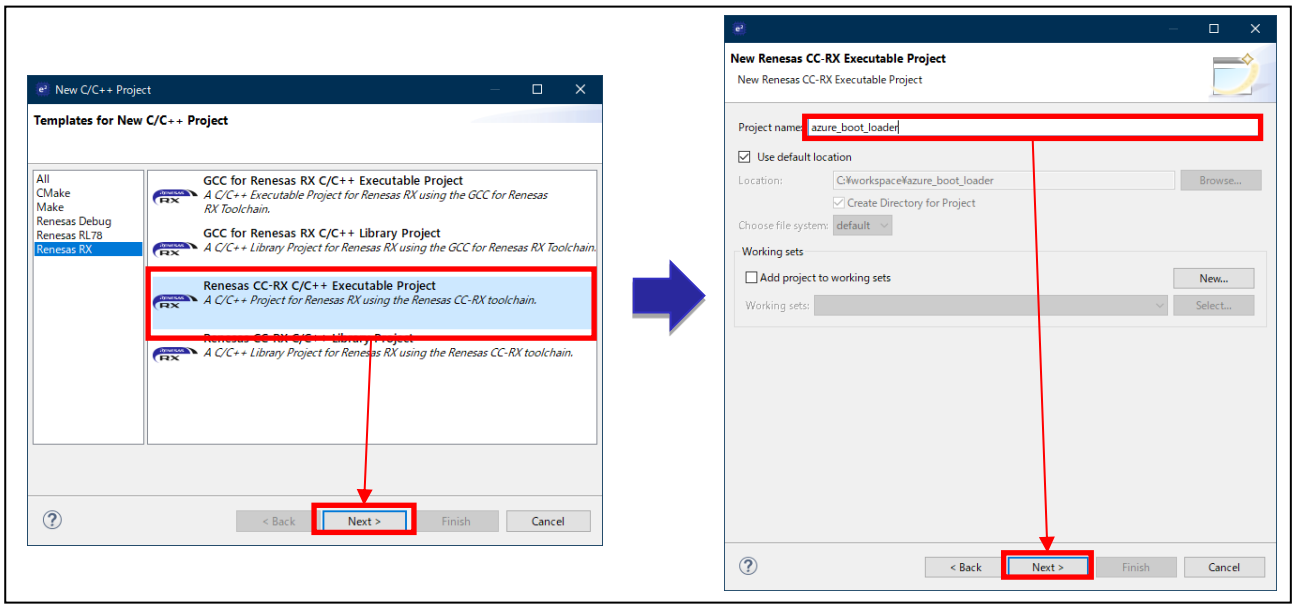

Figure 4-25 Azure Project Creation 1

Select “Azure RTOS” from the “RTOS” combo box.

Select “6.2.1_rel-rx-1.0.1” from the “RTOS Version” combo box. If the combo box is empty, click the link for “Manage RTOS Versions…” below the box to open to the download dialog. From the dialog, specify “Module Folder Path” and download “6.2.1_rel-rx-1.0.1”.

After downloading the version, you will be able to select the downloaded version in the “RTOS Versions” combo box.

For details and selection of available revisions, refer to GitHub (Releases · renesas/azure-rtos (github.com)).

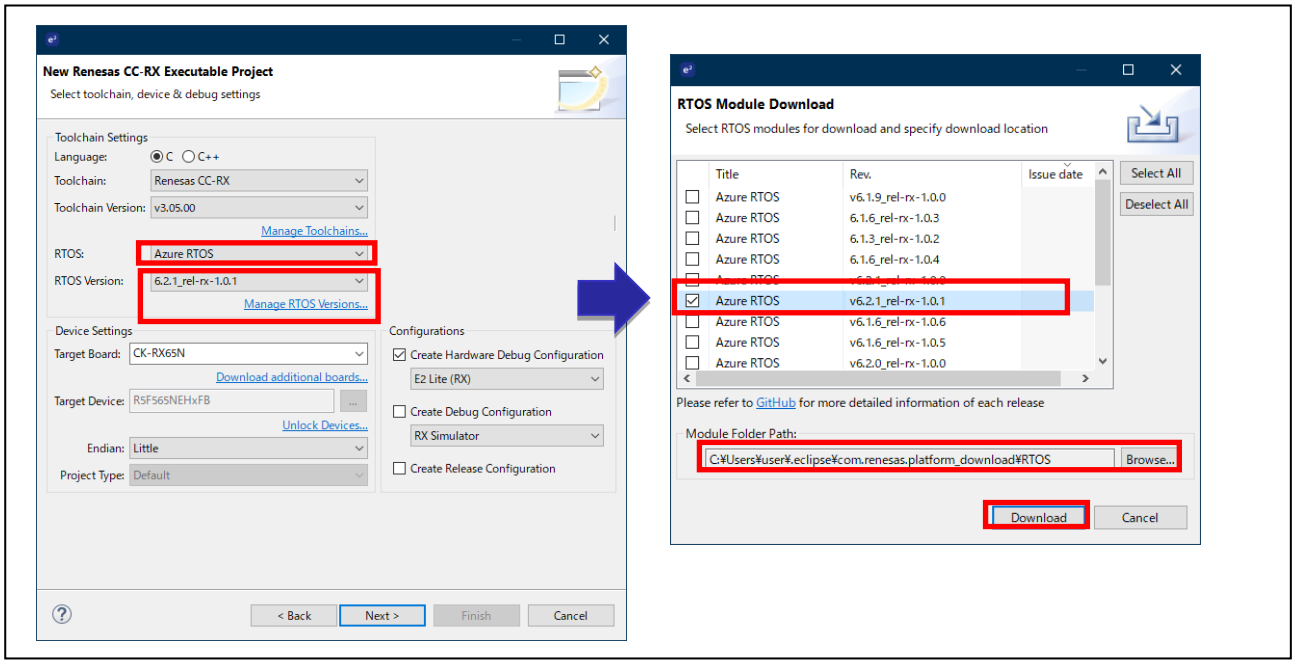

Figure 4-26 Azure Project Creation 2

On the “Target Board”, select “CK-RX65N”, and then click “Next >”.

After the Smart Configurator description appears, click “Next >”.

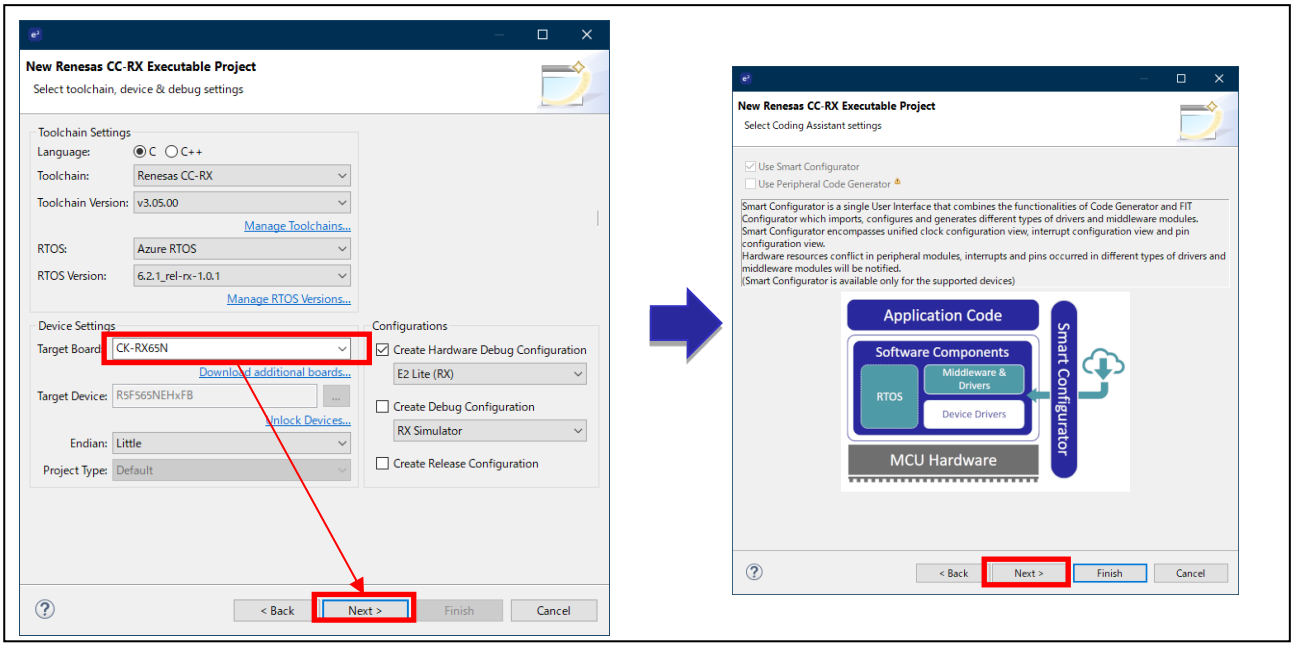

Figure 4-27 Azure Project Creation 3

Select “Secure bootloader sample project” in the “Select application area of “Select RTOS Project Settings”, and then click “Finish”.

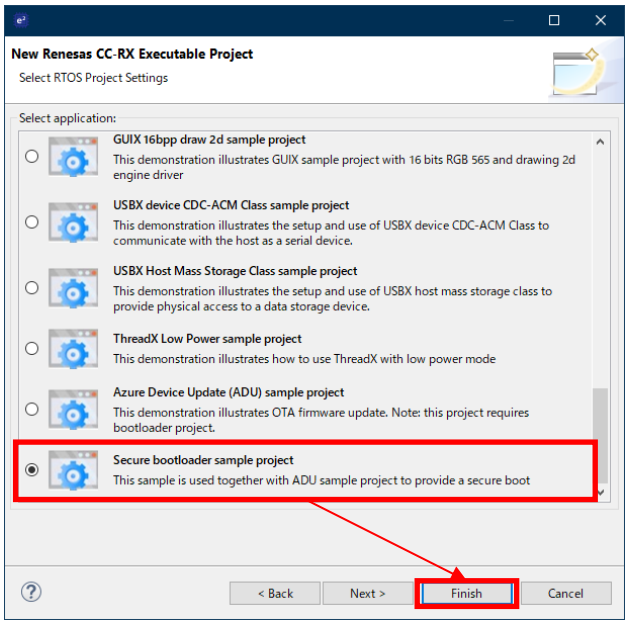

Figure 4-28 Azure RTOS Project Selection - Bootloader Sample Project

The “README.md” file is automatically opened when the project is generated. Follow the instructions in the file to configure the project.

Double-click “ProjectName.scfg” in the project explorer to open the Smart Configurator. After that, select the “Board” tab to open the board selection dialog.

Append “_DUAL” to the end of the target device name.

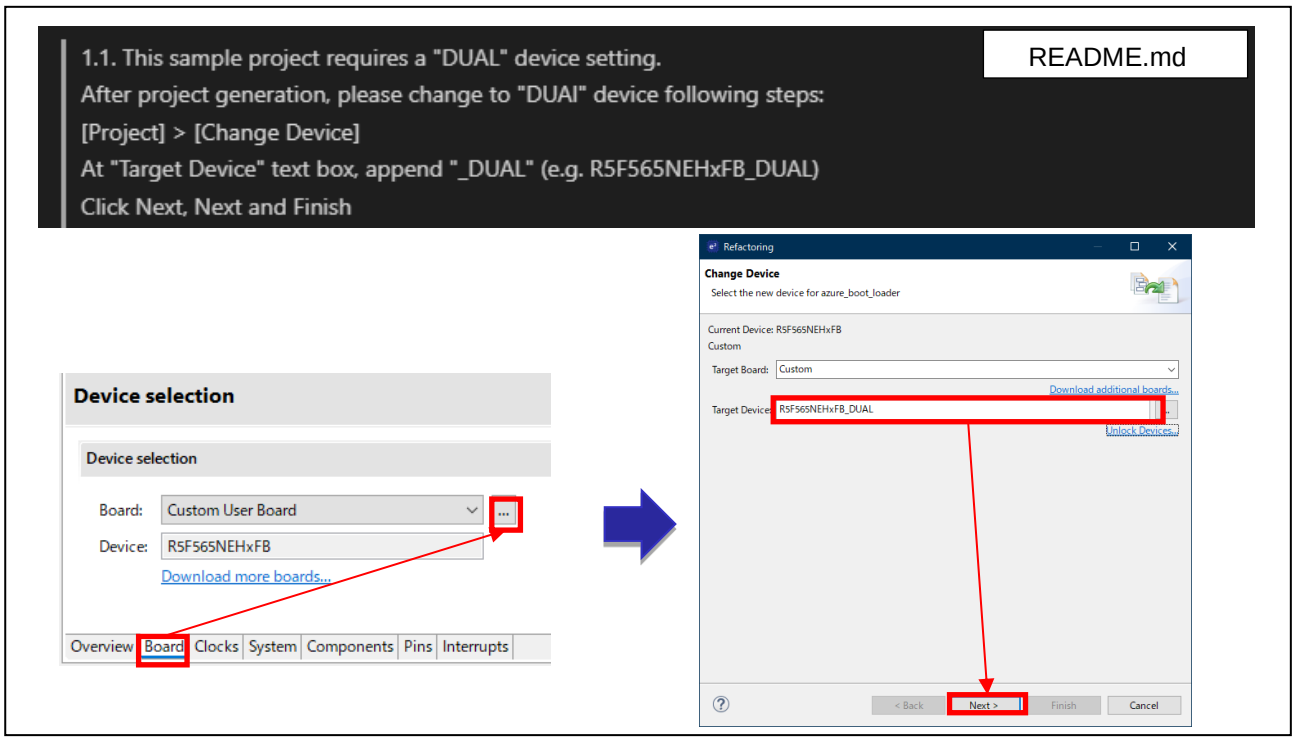

Figure 4-29 Azure Project Device Selection 1

Finally, click “Next >” then “Finish” to complete the device change. Please note that when a device is changed, the compiler settings are automatically reset.

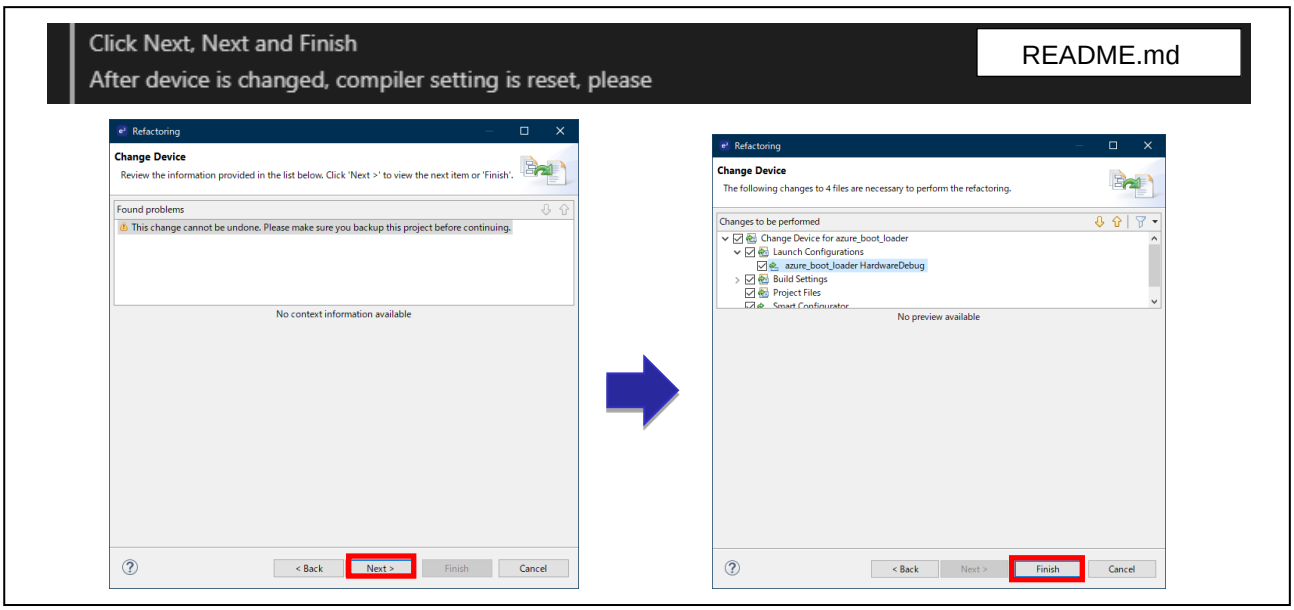

Figure 4-30 Azure Project Device Selection 2

If the following dialog appears during the above procedure, click “Yes” and continue.

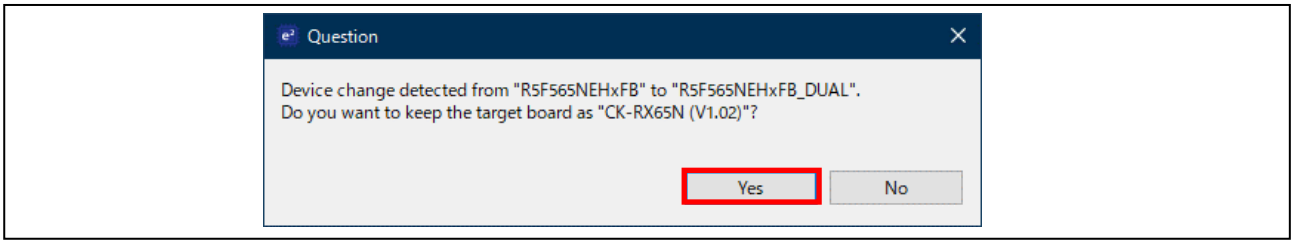

Figure 4-31 Dialog During Azure Project Device Selection

(a) Section Settings for CC-RX

This section describes how to set up sections. Open Properties using the right click. Select “Settings” and open the Section settings using the button as shown below. When the Section Viewer dialog appears, click “Import…”.

This will open the Select File dialog. Select and import the “linker_section_sample.esi” file located in the projects src folder. (Note 1)

Note 1. After changing the target device name, the section settings may be changed during the initial firmware creation, and the initial firmware creation may fail. In that case, please try the section settings again.

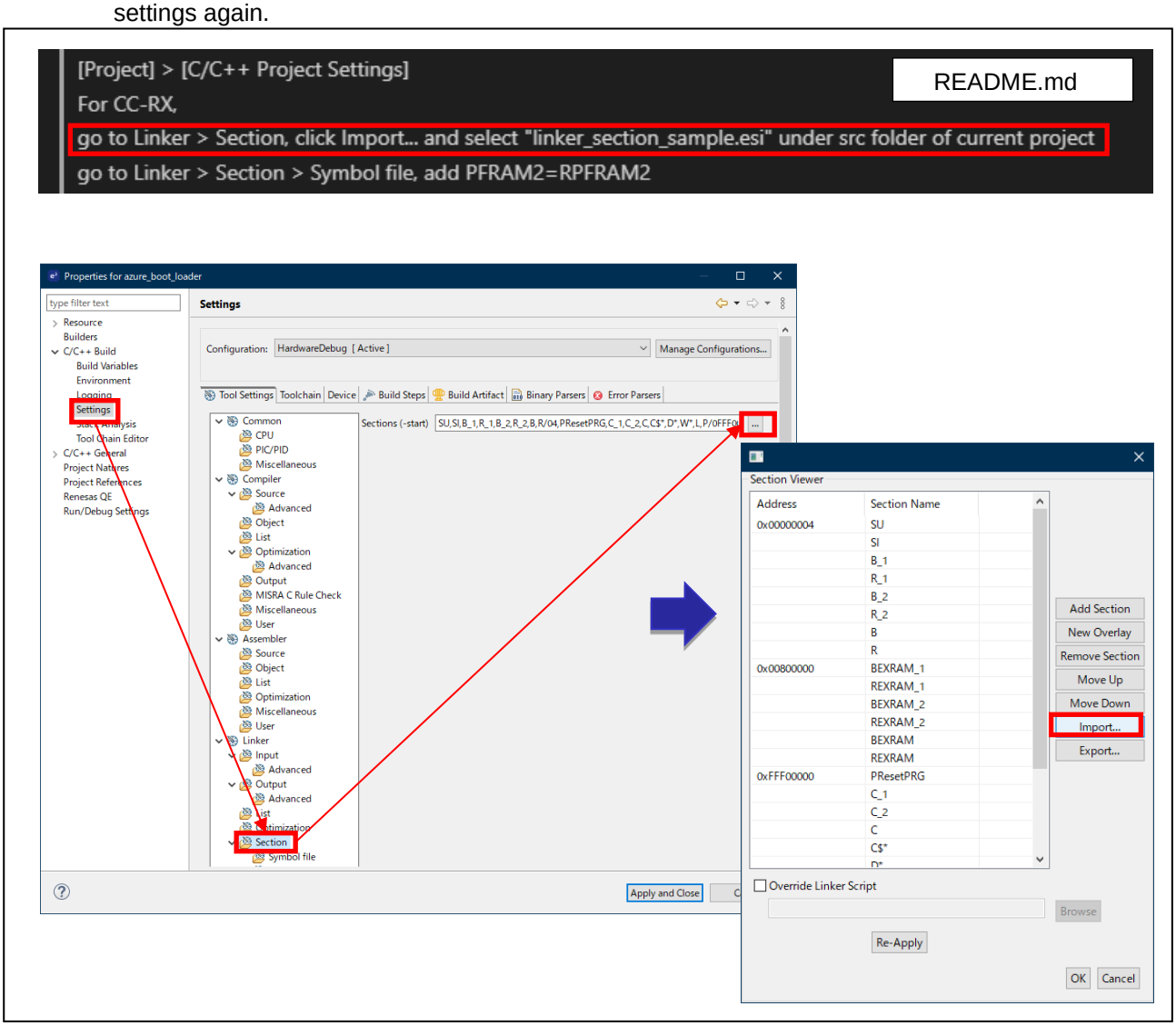

Figure 4-32 Azure Project Section Settings 1

Next, go to “Section” > “Symbol file”, and add “PFRAM2=RPFRAM2”.

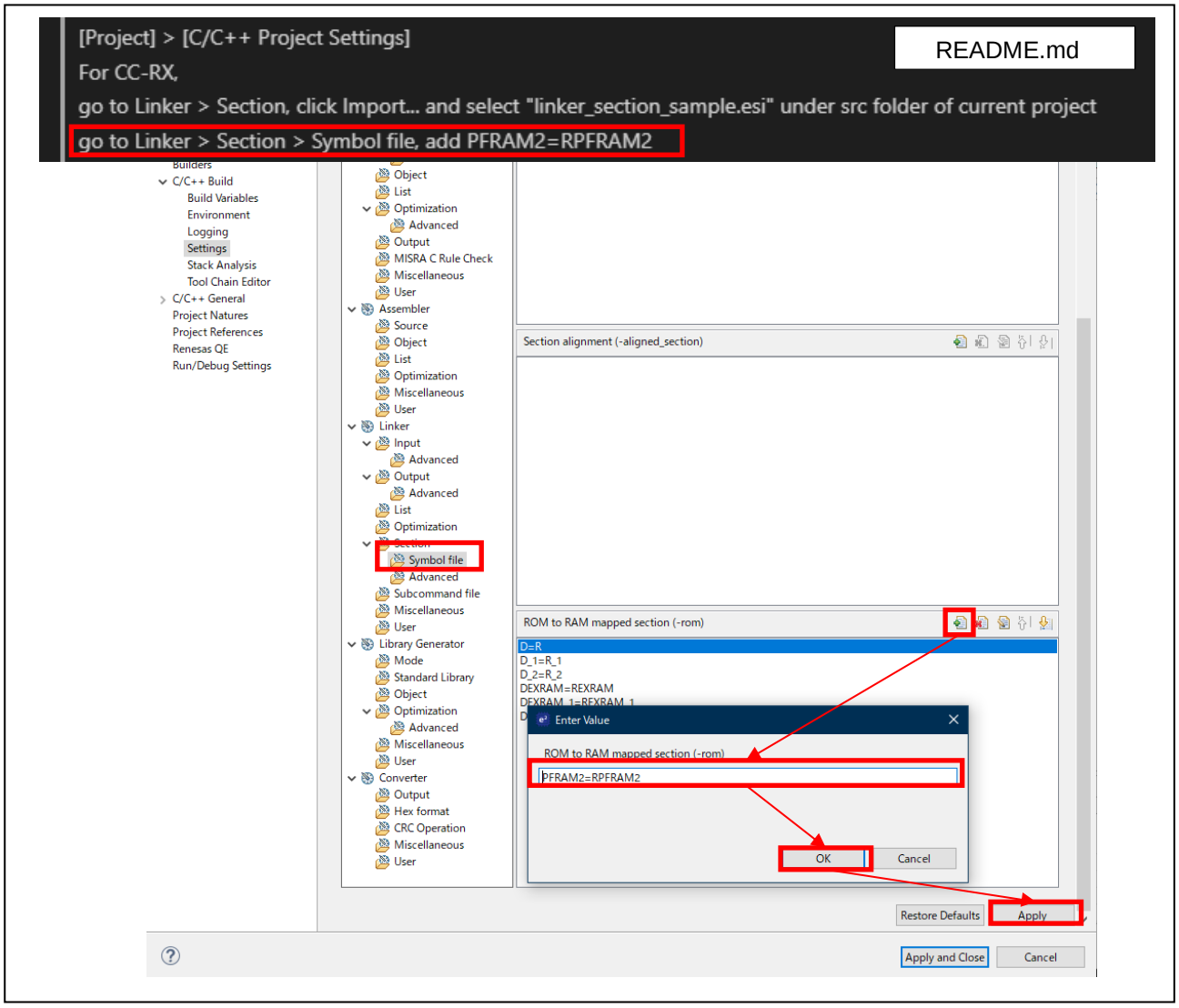

Figure 4-33 Azure Project Section Settings 2

(b) Section Settings for GCC

To use GCC, you will need to replace the linker setting file.

Original location:

<project folder>/src/linker_script_sample.ld

Location after replacement:

<project folder>/src/linker_script.ld

Figure 4-34 Azure Project Section Settings 3

(2) Create ADU sample project

The next step is to create an ADU sample project. The procedure is the same as that for the bootloader project, but you will need to select “Azure Device Update (ADU) sample project” as the RTOS project setting.

The same exception applies to changing a device or setting a section.

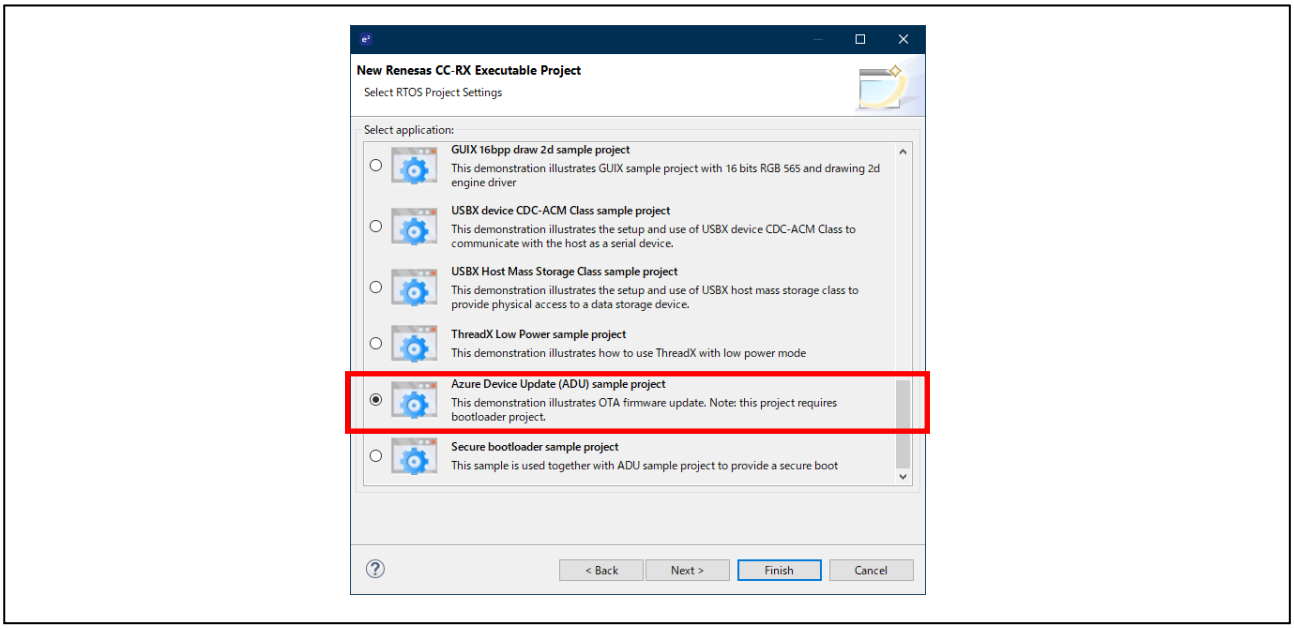

Figure 4-35 Azure RTOS Project Selection - ADU Project

(3) Select projects

Select the created projects.

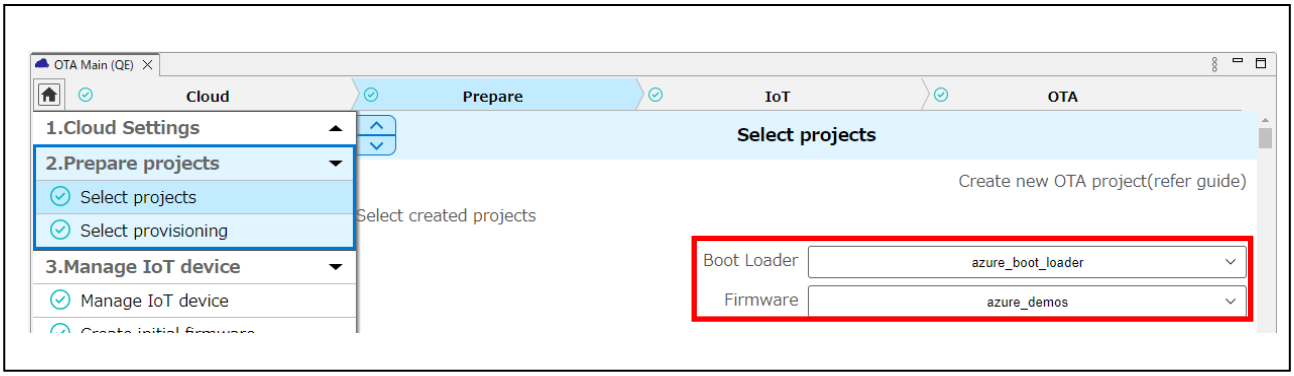

Figure 4-36 Select Projects Created for Azure

4.4.1.3 Download Smart Configurator Components

After you have created the sample project and bootloader project, open the Smart Configurator by double-clicking the “<Project name>.scfg” file in each project using the project explorer.

[Open the “Components” tab. If you see a gray cube icon next to any node, select the node and click the “downloading it” link to download the component.

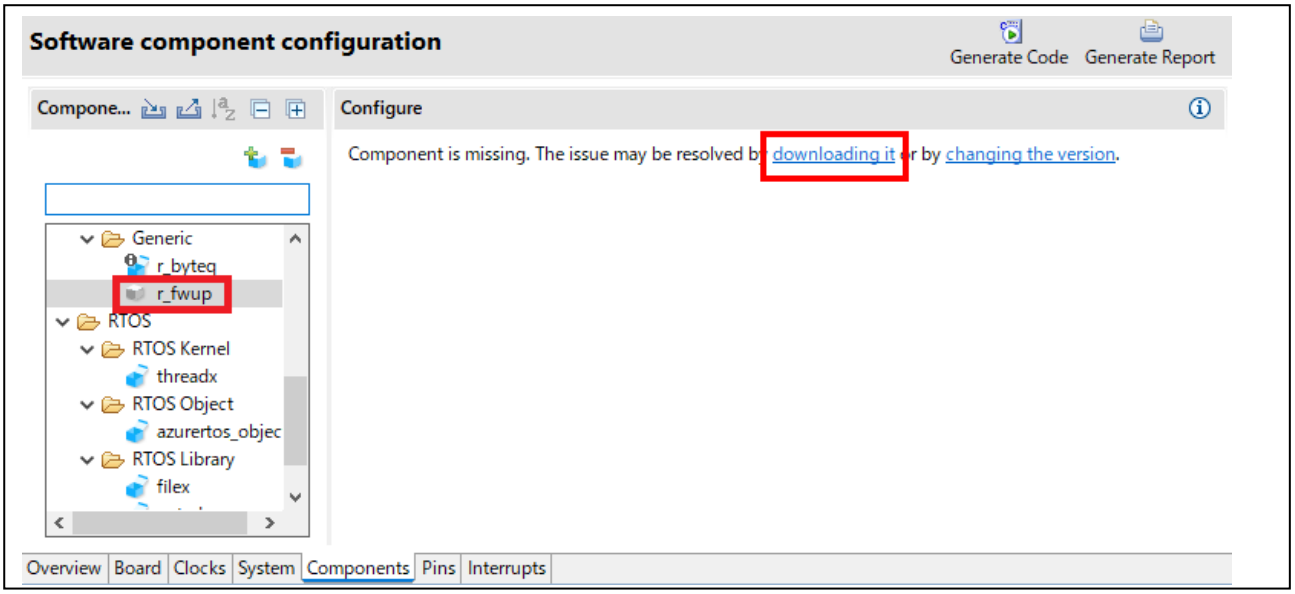

Figure 4-37 Download Components

4.4.2 Select Provisioning

As QE for OTA V1.1.0 supports only one provisioning, no settings are necessary.

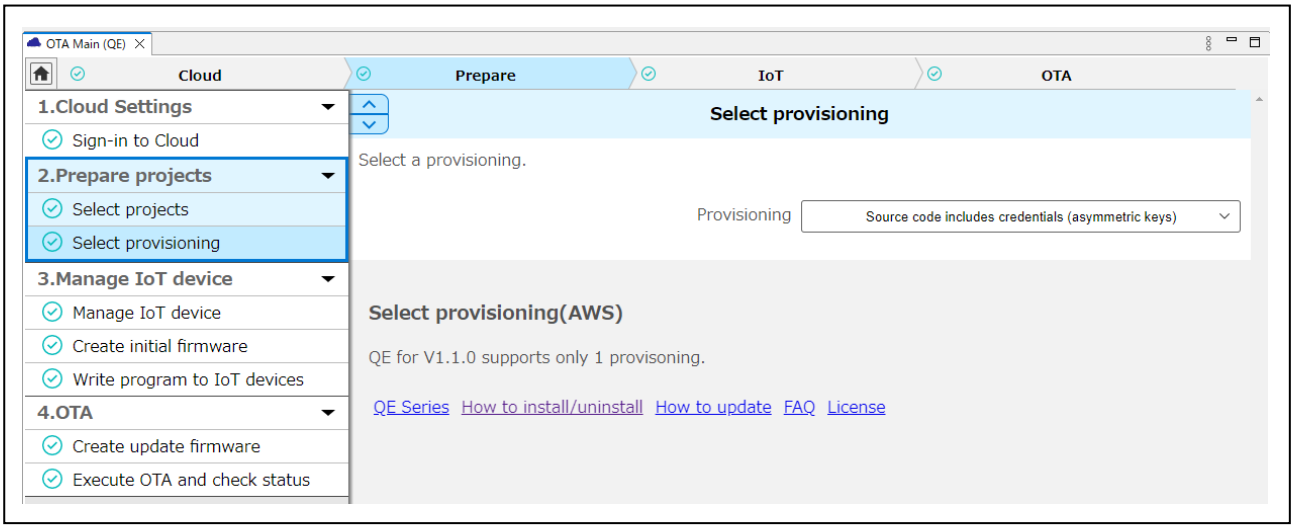

Figure 4-38 Provisioning Selection

4.5 Manage IoT Device

Create a new IoT device on the cloud.

4.5.1 Manage IoT Device

Select “Manage IoT device” from the menu displayed on the left side of the OTA Main (QE) view and click “Open view”.

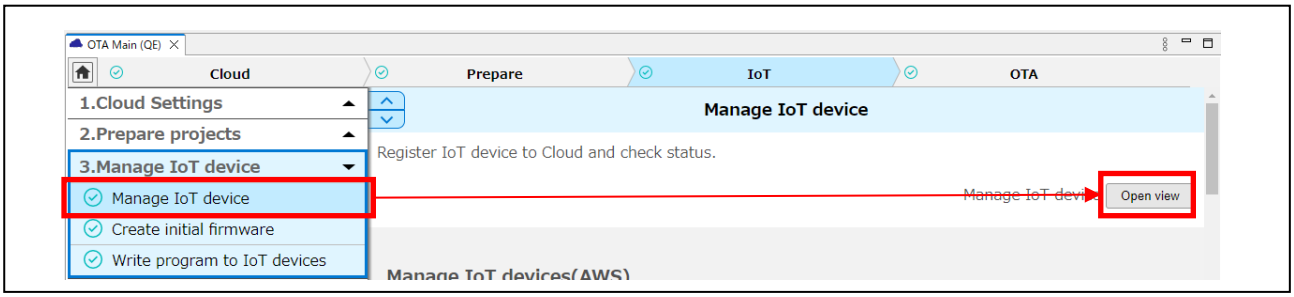

Figure 4-39 Manage IoT Device

The OTA Manage IoT Device view will open.

Click the “Add” icon, then select “Create IoT device”.

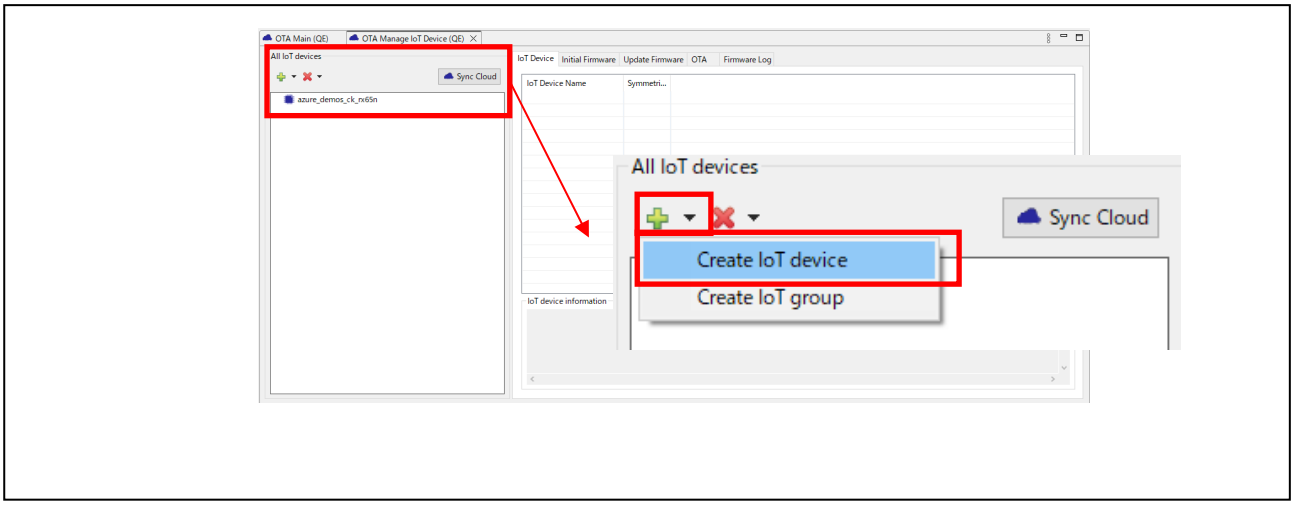

Figure 4-40 Manage IoT Device - Create IoT Device

Enter the information about the IoT device you are creating and click “Create.”

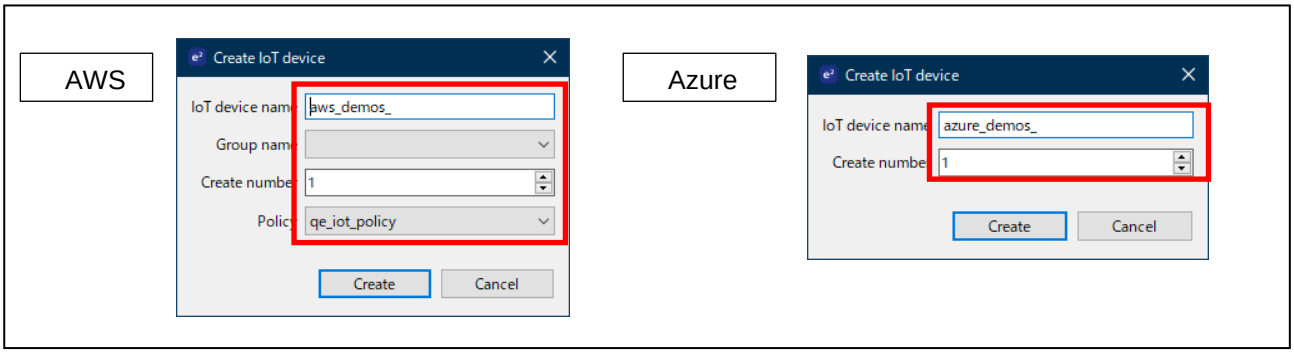

Figure 4-41 Create IoT Device Dialog

The contents of each item of IoT device information are shown below.

Table 4-1 Parameters for Creating IoT Device

Item	Contents
IoT device name	Specify the name of the IoT devices. When creating two or more devices, QE automatically adds a sequential number to the IoT device name.
Group name	Specify a group name when required (AWS only).
Create number	Specify the number of devices to create.
Policy	Select a policy. If “Create new policy” is selected QE will automatically create it (AWS only).

Confirm the information for the created IoT device.

The device certification and key will be automatically downloaded and displayed.

Note that the notation for AWS and Azure differ.

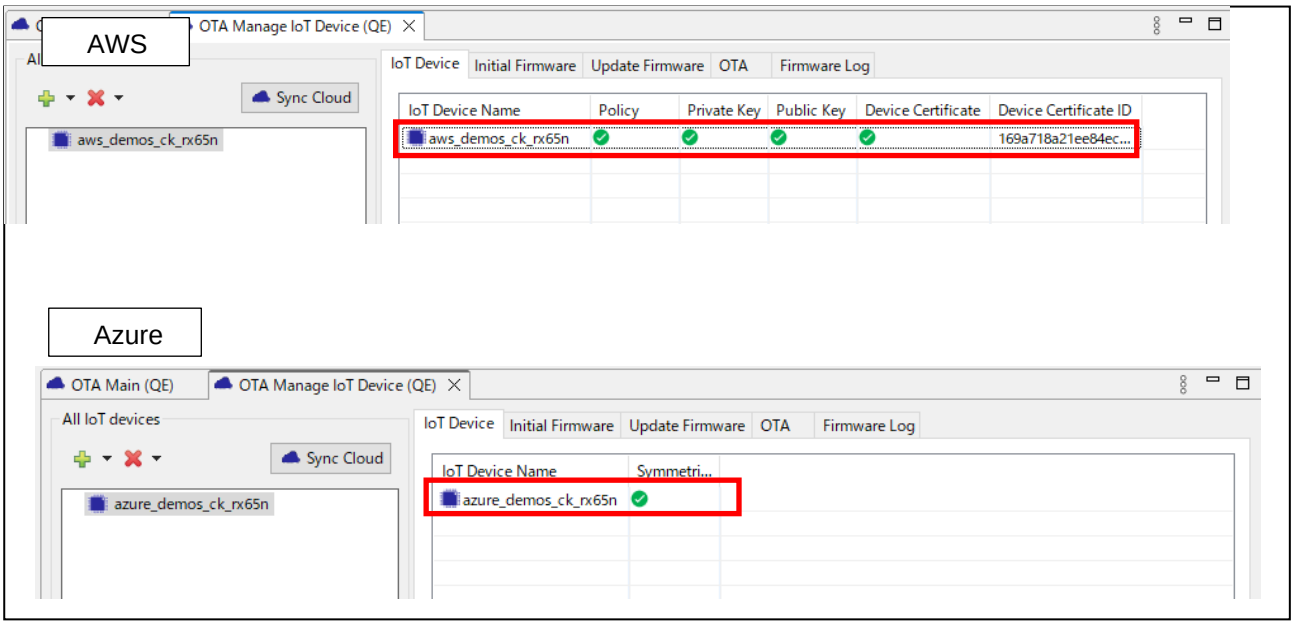

Figure 4-42 Manage IoT Device - Display After IoT Device Creation

4.5.2 Create Initial Firmware

This section describes the procedure for creating the initial firmware.

4.5.2.1 Confirm Toolchain Version

If the installed toolchain version is not the same as the toolchain versions used by each project, an error will occur during the build.

In this case, you will either need to install the same or later version of the toolchain.

The toolchain version can be confirmed in the property settings of each project.

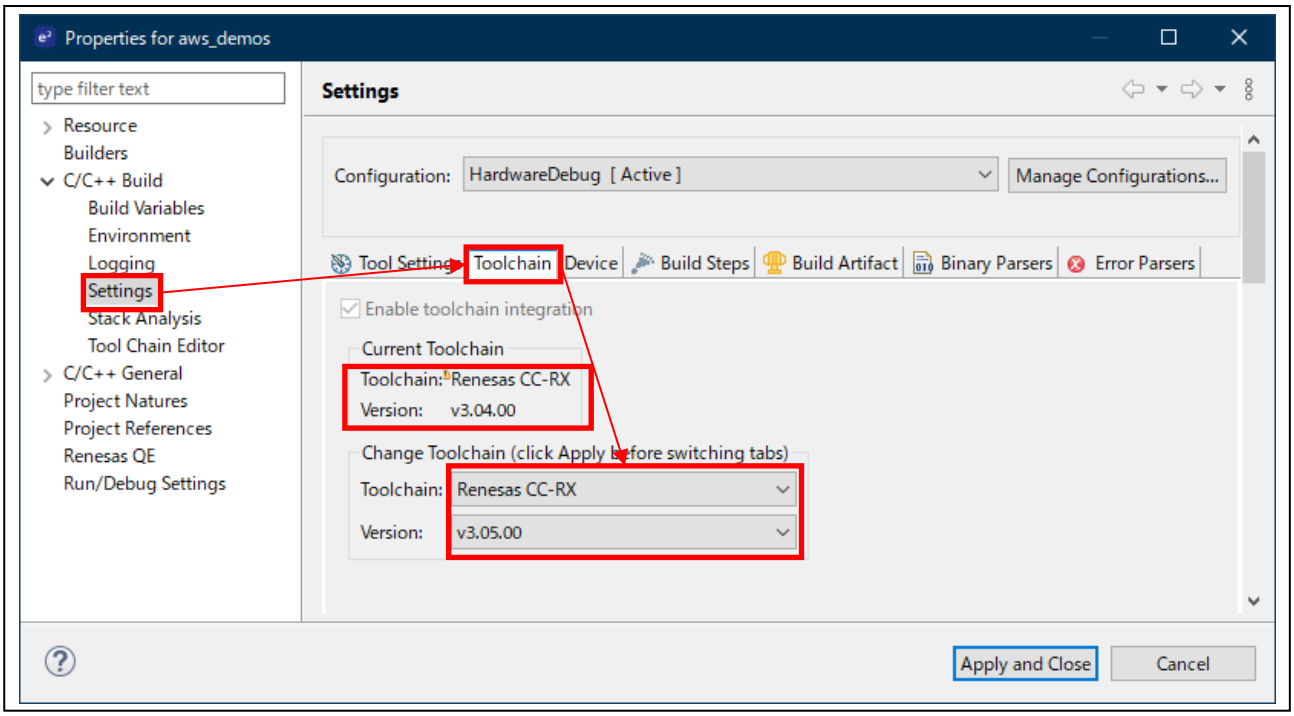

Figure 4-43 Toolchain Version

4.5.2.2 Create Initial Firmware

Create the initial firmware after completing the necessary settings.

Select “Create initial firmware” from the menu displayed on the left side of the OTA Main (QE) view and click “Open view.” The “Initial Firmware” tab in the OTA Manage IoT device (QE)” view will open.

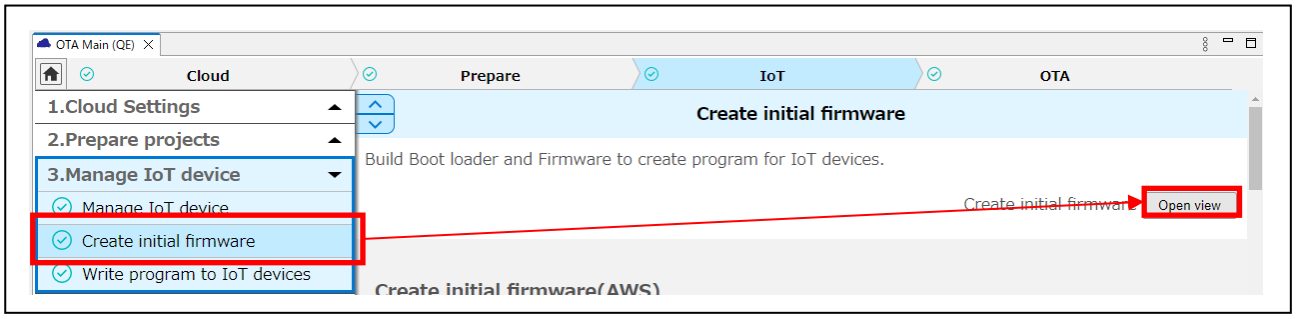

Figure 4-44 Create Initial Firmware - Open View

Select an IoT device and click “Add”. The status will appear on the right side. To register all IoT devices that are displayed, click “Add all.

Next, select the IoT device you want to write from the list of registered devices, specify the version of the initial firmware, and click “Create initial firmware”.

This same procedure applies to both AWS and Azure, but the display of registered IoT devices in the upper right of the tab differs.

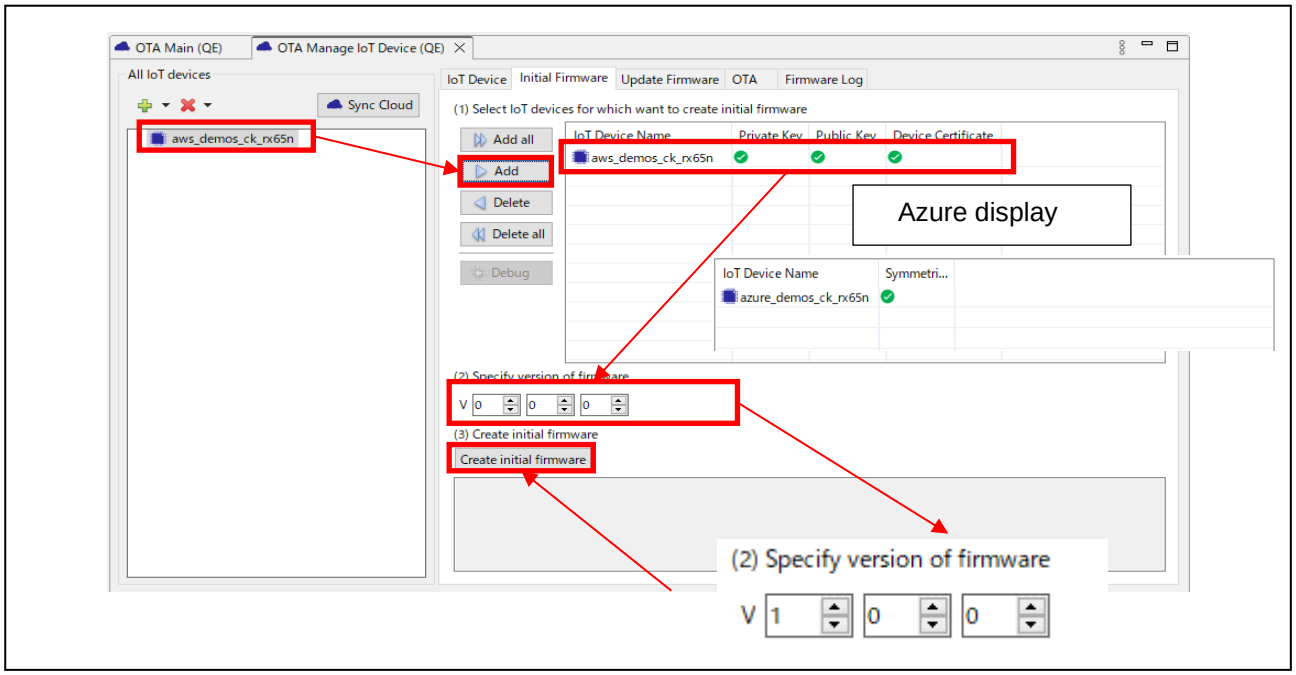

Figure 4-45 Create Initial Firmware

Clicking “Create initial firmware” opens the Code Signing dialog. If you have already created a code signing certificate, the Certificate Copy dialog with open. (See Figure 4-51 Copy Code Signing Certificate Files Dialog.)

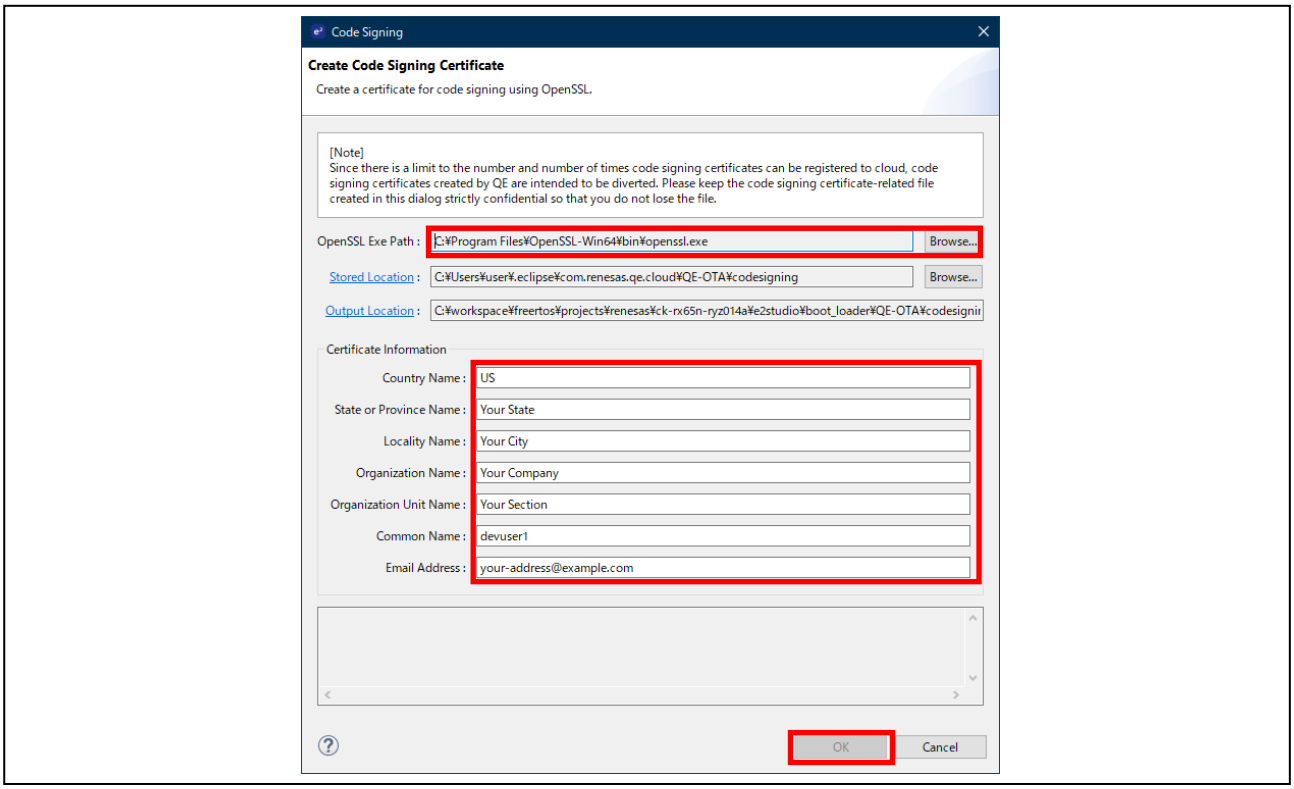

Figure 4-46 Create Code Signing Certificate Dialog

After setting the Open SSL path and entering your certificate information, click “OK”. This will automatically create the initial firmware.

Note that since there is a limit to the number and number of times that code signing certificates can be registered to the cloud, code signing certificates created by QE are intended to be diverted. When a certificate diverted, it will no longer be displayed in the code singing dialog. Diverting certificates is explained later in this document (see 4.5.2.4 Code Signing Certificate).

When a code signing certificate is created successfully, it will look like the following. This indicates that the initial firmware creation process has completed.

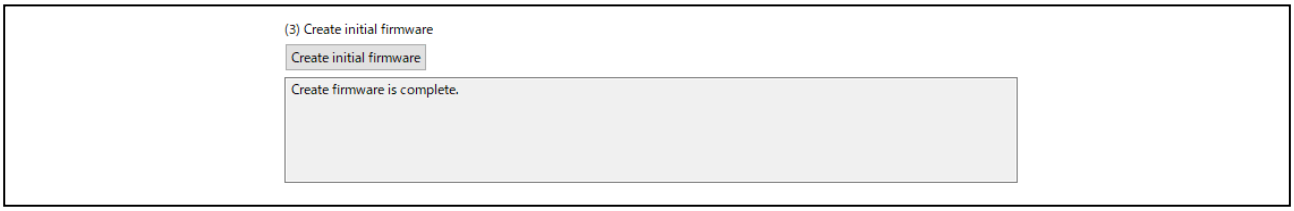

Figure 4-47 Create Initial Firmware - Successful Completion

The following dialog may appear during this procedure. In this case, select the target board and click “OK.”

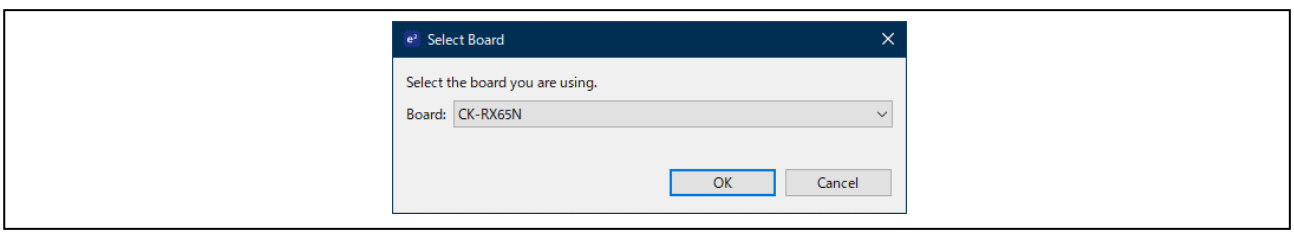

Figure 4-48 Select Board Dialog

4.5.2.3 Sample Project Firmware

Firmware is generated at the location shown in the figure below with the following contents.

<project folder>¥QE-OTA¥apl¥<version>¥<IoT device name >

- <Project name>.mot
  This is the sample project HEX file in Motorola Format.
- <Project name>.x (CC-RX の場合), <Project name>.elf (for GCC)
  This is the sample project load module file.
- merged.mot
  This is the HEX file of the merged sample and bootloader projects in Motorola Format. Write this file to the IoT device.

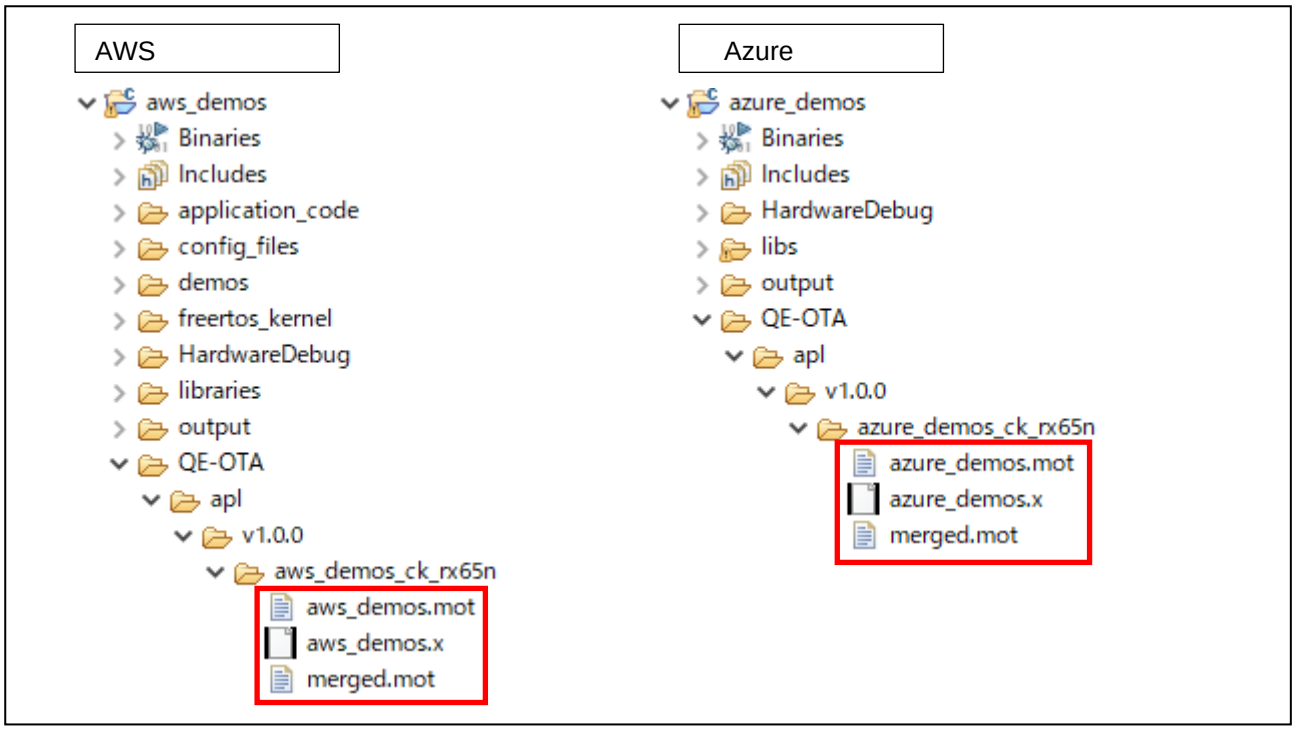

Figure 4-49 Locations of Created Initial Firmware (CC-RX)

4.5.2.4 Code Signing Certificate

The created code signing certificates are stored in the following locations in the bootloader project.

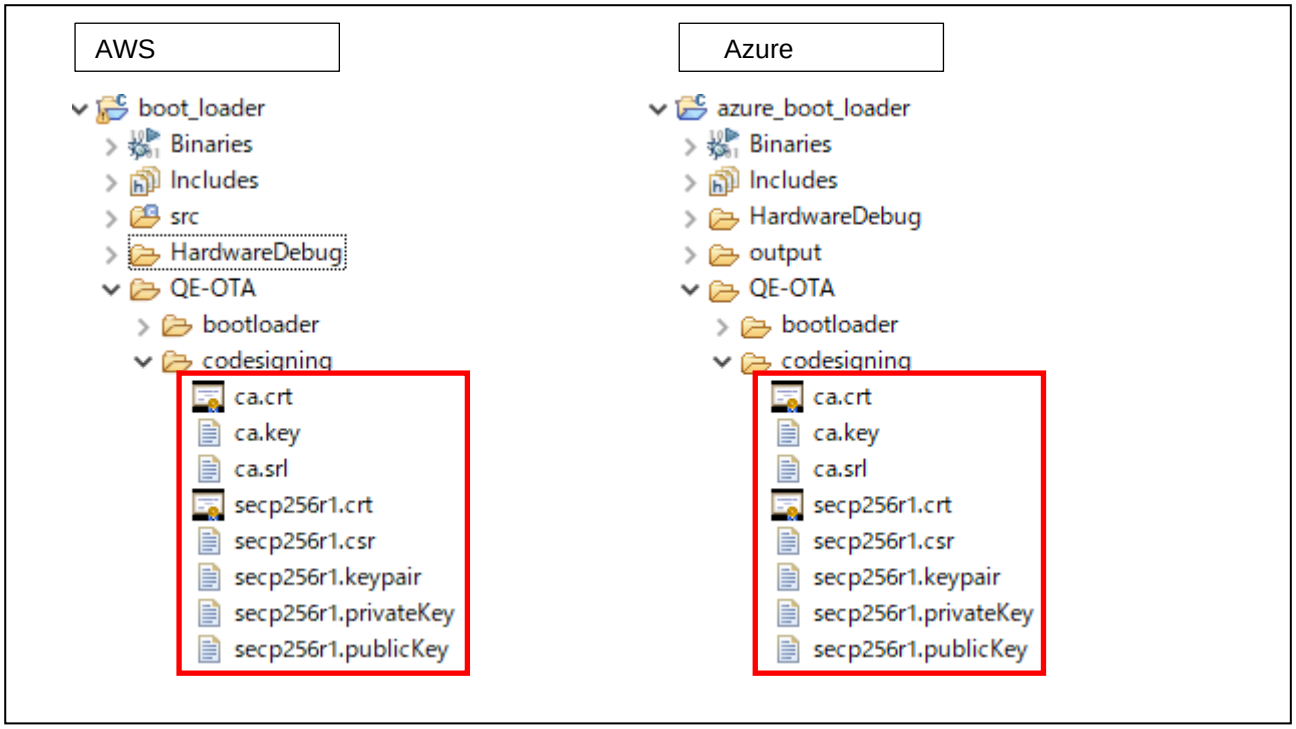

Figure 4-50 Code Signing Certificate Locations

Make sure to carefully manage code signing certificate and related files so that you do not lose them. When executing OTA, the code signing certificate generated here are used to create a code signing profile (qe_code_signing), but once the code signing profile is created, it cannot be recreated in QE for OTA. When using a different code signing, use the AWS console to create a code signing profile.

For this reason, certificates created in QE for OTA are intended to be diverted. The following procedure describes how to reuse a certificate.

1. Go to the following folder in the project where you created the code signing certificate and copy the codesigning folder.
   <bootloader project folder>/QE-OTA/codesigning
2. Paste the codesigning folder in the same location in the new project.
   <bootloader project folder /QE-OTA/codesigning

Note that when certificate is stored in the “Stored location” specified in the “Created Code Signing Certificate” dialog, it may be automatically diverted.

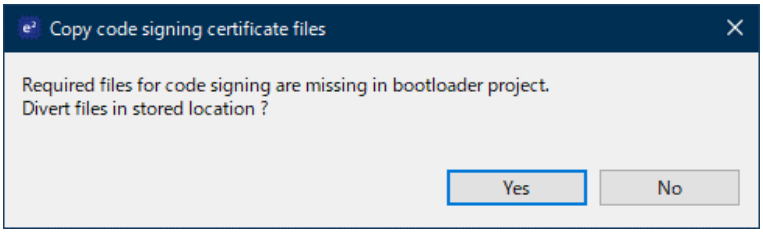

Figure 4-51 Copy Code Signing Certificate Files Dialog

4.5.3 Write Program to IoT Device

Write the initial firmware to the board.

Select “Write program to IoT devices” from the menu displayed on the left side of the OTA Main (QE) view and click “Open view”.

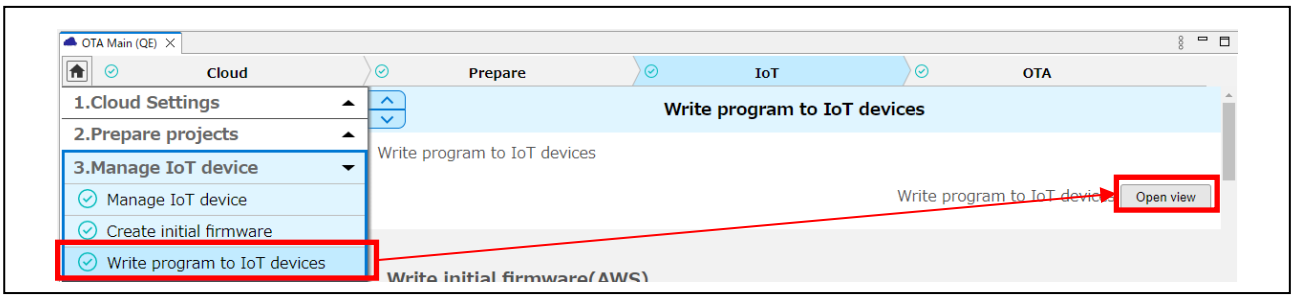

Figure 4-52 Write Program to IoT Devices

The Initial Firmware tab will open. Select the IoT device to be written to the board from the list of registered devices and click “Debug”.

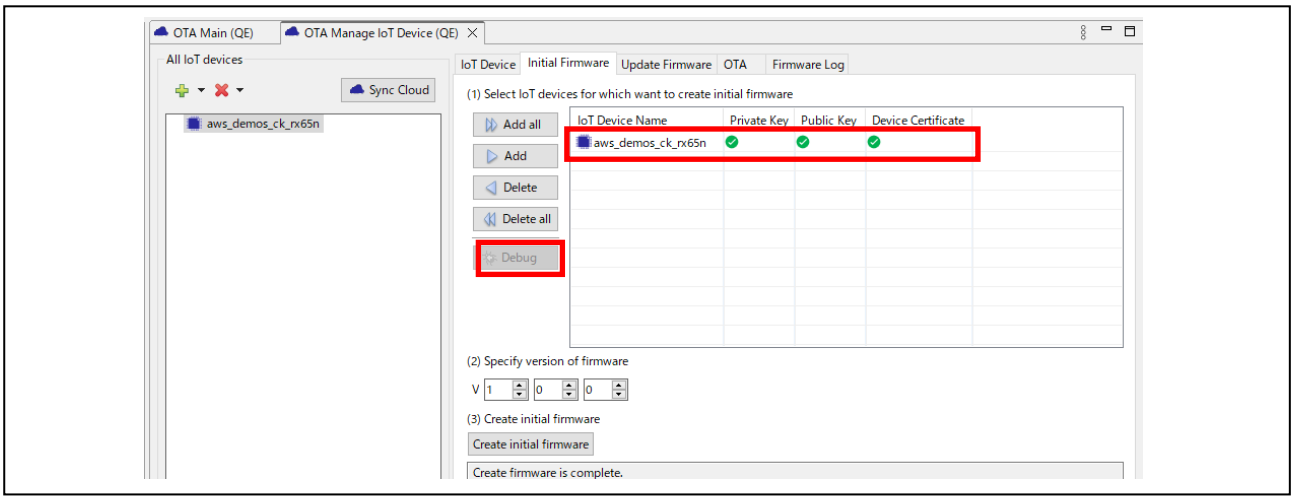

Figure 4-53 Debug Button

The Flash Programming dialog for writing the firmware will open.

Connect the board and supply power.

Specify the Renesas Flash Programmer (rfp-cli.exe) path and ID code, as well as the target board and emulator, and then click “Write”. When the write is successfully completed, “Operation successful” will be displayed.

To continue to debug programs, click “Debug”.

The initial firmware can be run by clicking “Debug”.

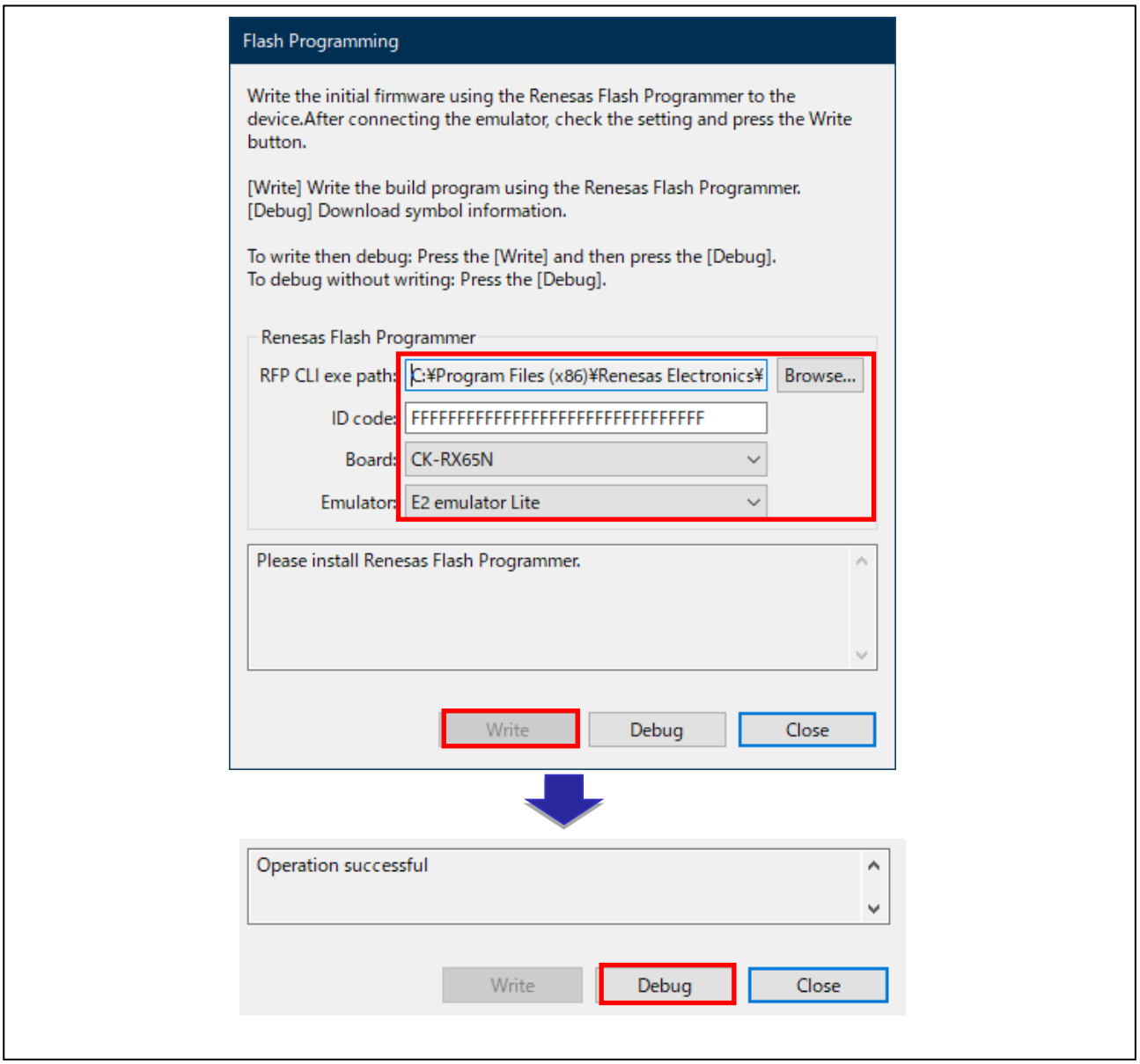

Figure 4-54 Write Initial Firmware

Immediately after the debug operation is started, it stops by default at the start function break (twice by default).

Operation logs can be monitored in the “Firmware Log” tab of the “OTA Manage IoT Device” view Specify the COM port (or set to “auto”) and click “Connect” to display the firmware operation logs.

To check logs, you can display all the logs from the start by connecting to COM during the break.

After connecting to COM, click “Resume” to restart the program.

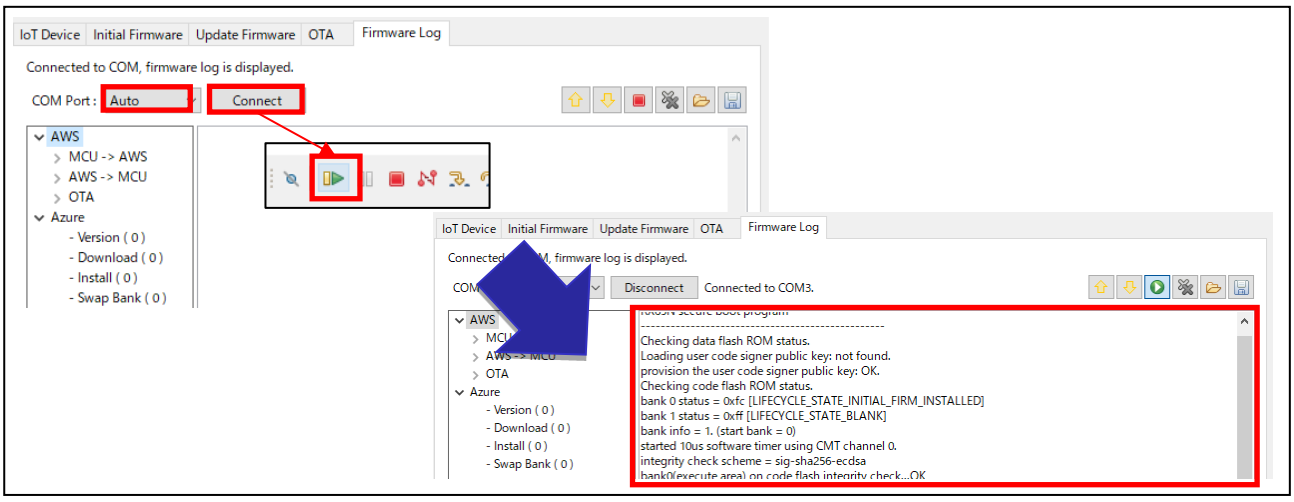

Figure 4-55 Firmware Dialog Display

When the initial firmware runs successfully, the following contents will appear in the firmware log. Check to make sure there are no error logs and that the displayed version is 1.0.0.

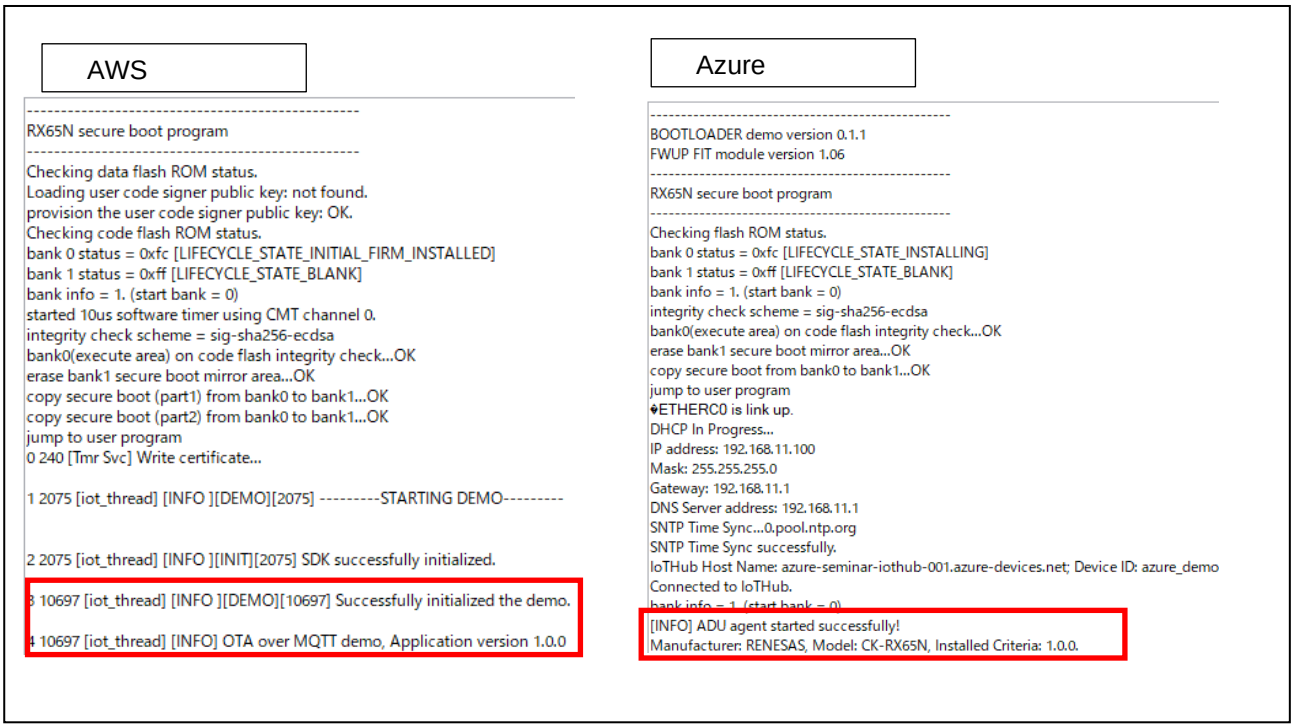

Figure 4-56 Firmware Dialog for Initial Firmware

4.6 OTA

This chapter describes how to create update firmware and how to execute OTA.

4.6.1 Create Update Firmware

Select “Create update firmware” from the menu displayed on the left side of the OTA Main (QE) view and click “Open view”. This will open the “Create update firmware” tab in the OTA Manage IoT device (QE) view.

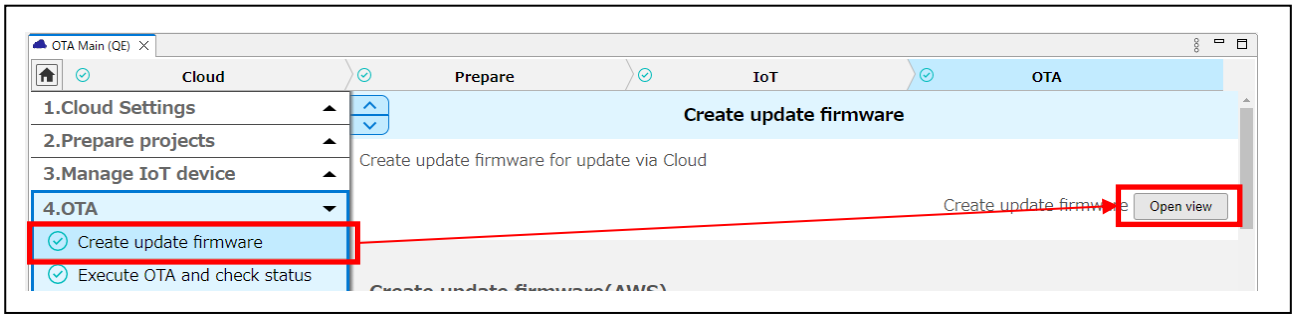

Figure 4-57 Create Update Firmware

The flow of for creating update firmware is the same as that for creating the initial firmware.

First, update the source file for the updated firmware with feature additions and fixes. This change is omitted in this application note, but make the necessary changes.

Register and select IoT devices, specify the version for the updated firmware, and then click “Create update firmware”. If the firmware is successfully created, the “Create firmware is complete” message will appear.

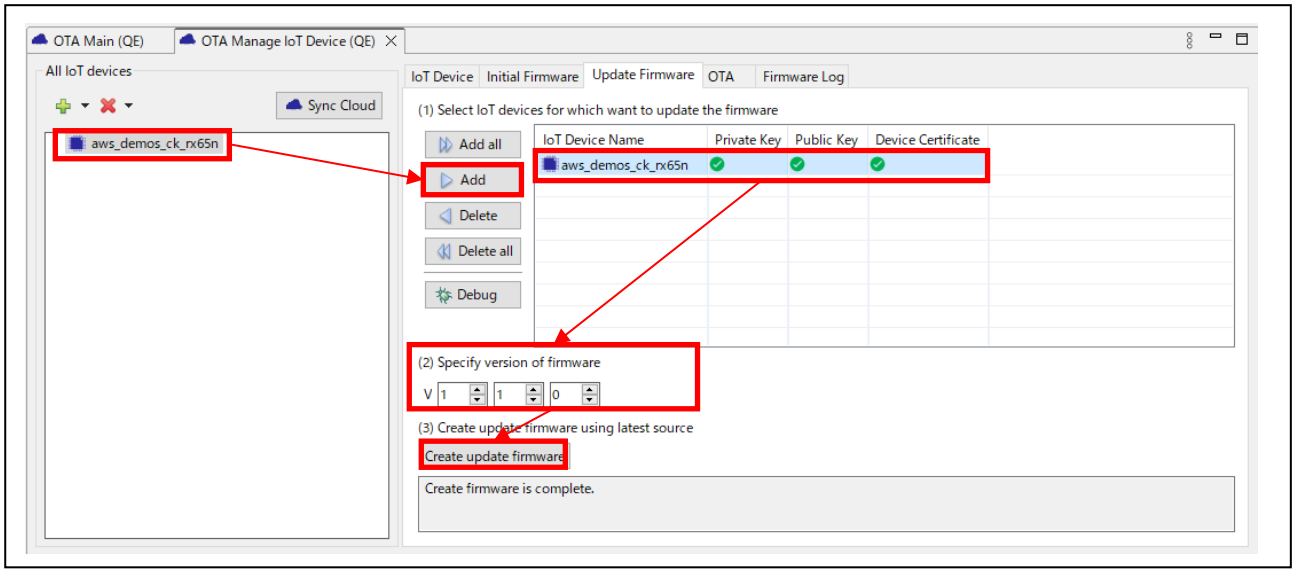

Figure 4-58 Update Firmware Creation Procedure

The update firmware is generated at the location shown in the figure below with the following contents.

<project folder>¥QE-OTA¥apl¥<version>¥<IoT device name>

- <Project name>.mot
  This is the sample project HEX file in Motorola Format.
- <Project name>.x (for CC-RX), <Project name>.elf (for GCC)
  This is the sample project load module file.
- <IoT device name>.rsu
  This is the update firmware created with the Renesas Secure Flash Programmer. Upload this file to the cloud.
- <IoT device name>.importmanifest.json (for Azure)
  This is the import manifest file. Upload this file to the cloud.

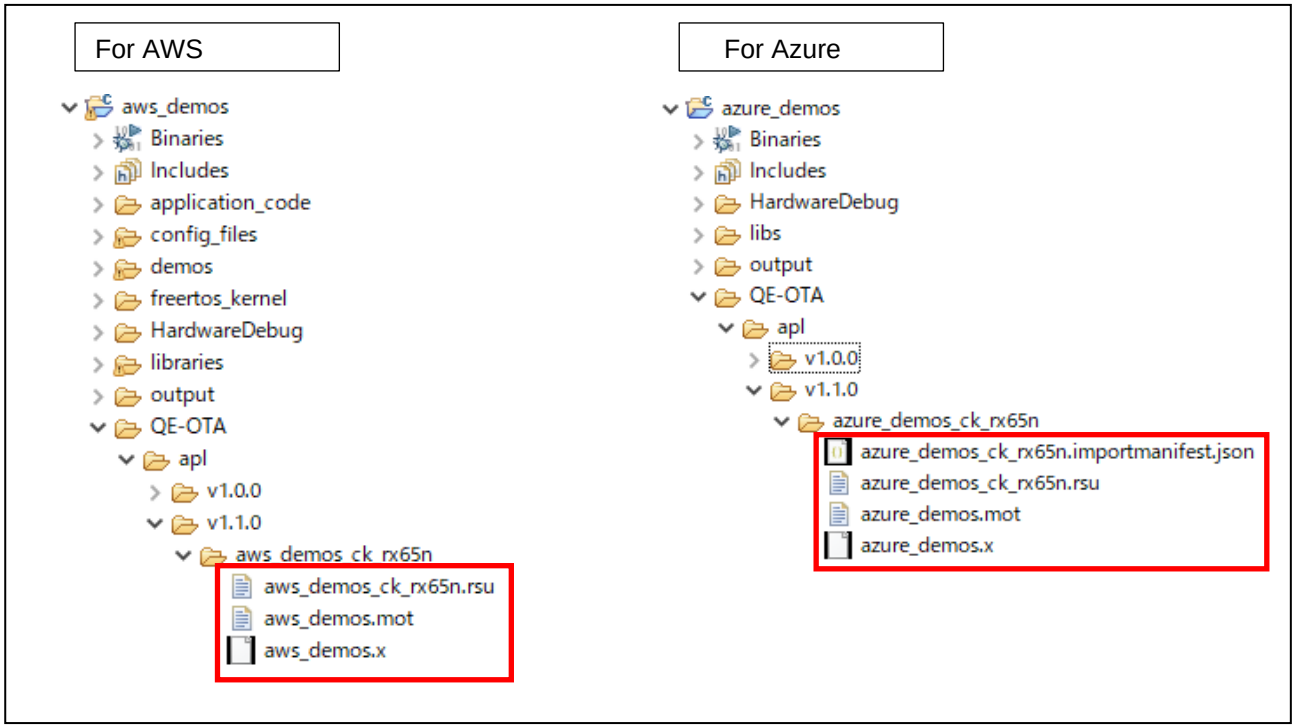

Figure 4-59 Location of Update Firmware Creation (CC-RX)

4.6.2 Execute OTA and Check Status

Select “Execute OTA and check status” from the menu displayed on the left side of the OTA Main (QE) view and click “Open view” located next to “Execute OTA”. This will open the “OTA” tab in the OTA manage IoT device (QE) view.

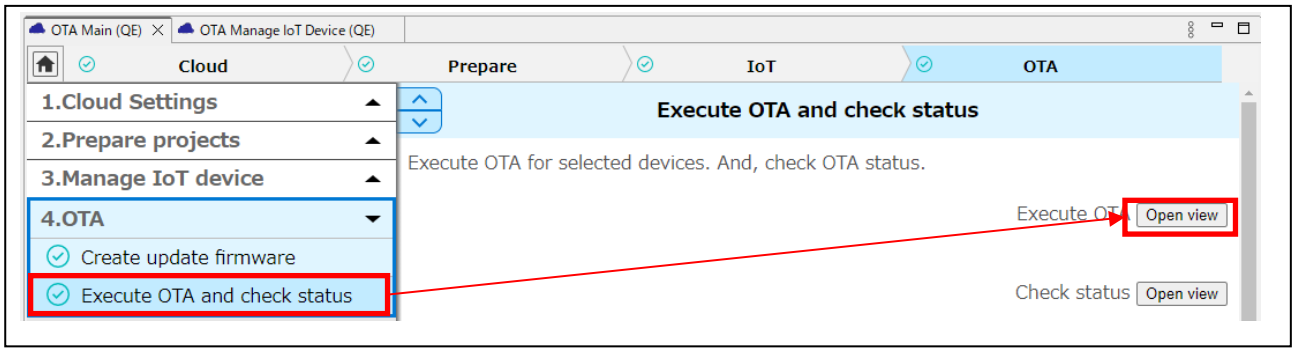

Figure 4-60 Execute OTA and Check Status

As with the other tabs, after selecting the IoT device that will execute OTA, click “Add” to register the device.

Select the registered IoT device and specify the version of the created update firmware.

For AWS, specify the role, update location, and code signing profile for the OTA to be executed. If the profile does not appear, select “Create new”, and QE for OTA will automatically create the profile when OTA is executed. For Azure, this section does not appear, and settings are not necessary.

Click “Execute OTA” to start running OTA.

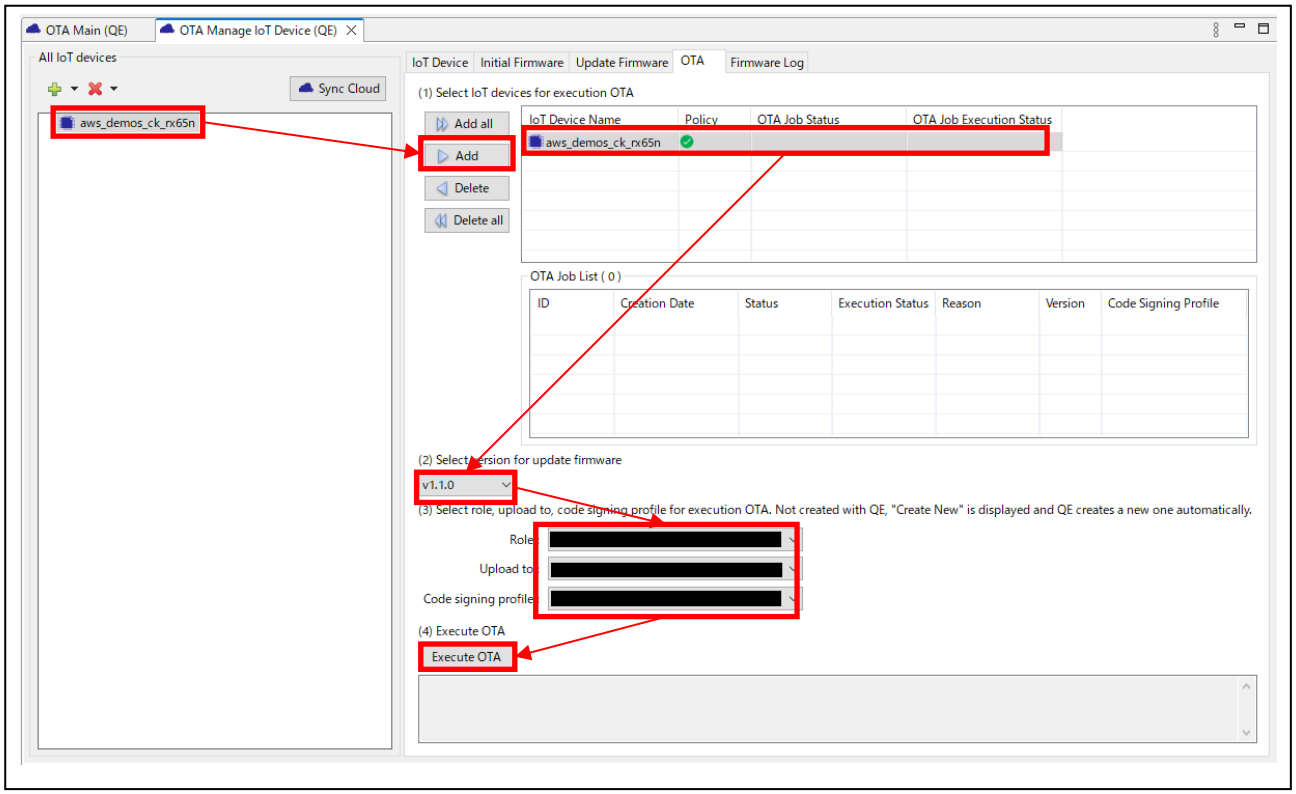

Figure 4-61 OTA Settings (AWS)

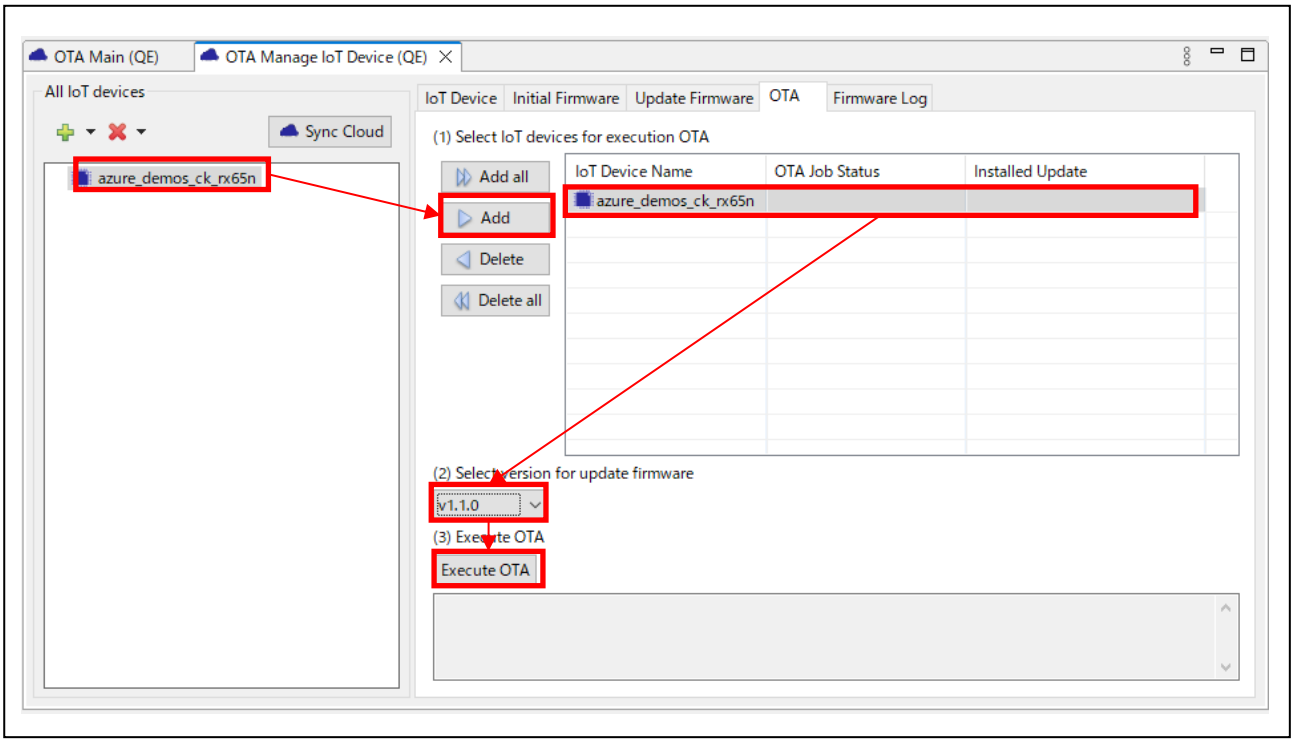

Figure 4-62 OTA Settings (Azure)

The OTA execution job is created automatically. For AWS, the OTA job status goes to “IN_PROGRESS”, and for Azure, it goes to “InProgress”.

When the IoT device (target board) is running in this state and connected to the network, it will detect the update and automatically start the firmware update. If the IoT device is not connected to the network, the firmware update will start after the network is connected.

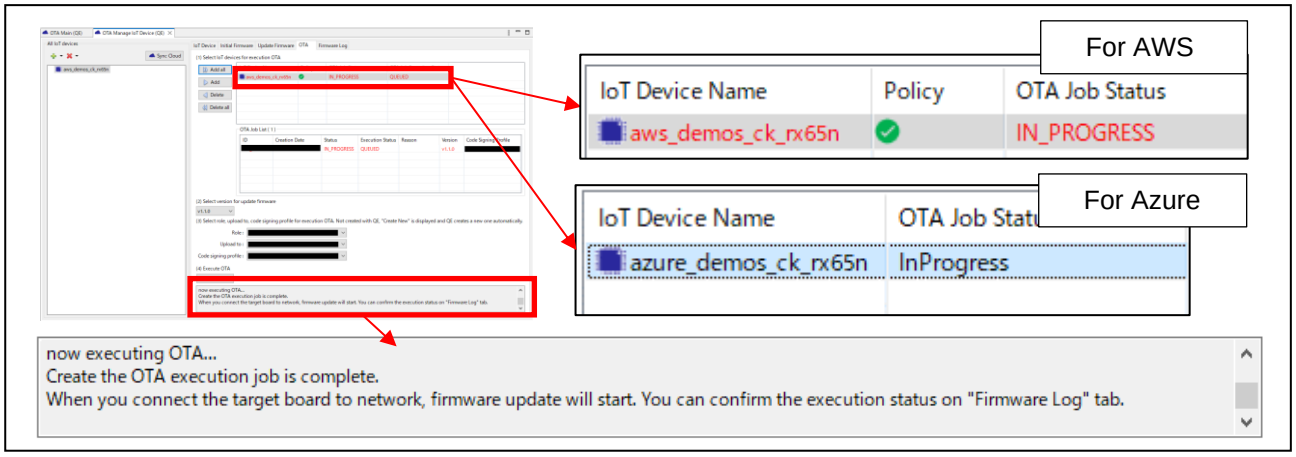

Figure 4-63 After Executing OTA

You can check the progress in the “Firmware Logs” tab.

Select “Execute OTA and check status” from the menu displayed on the left side of the OTA Main (QE) view and click “Open view” located next to “Check firmware”.

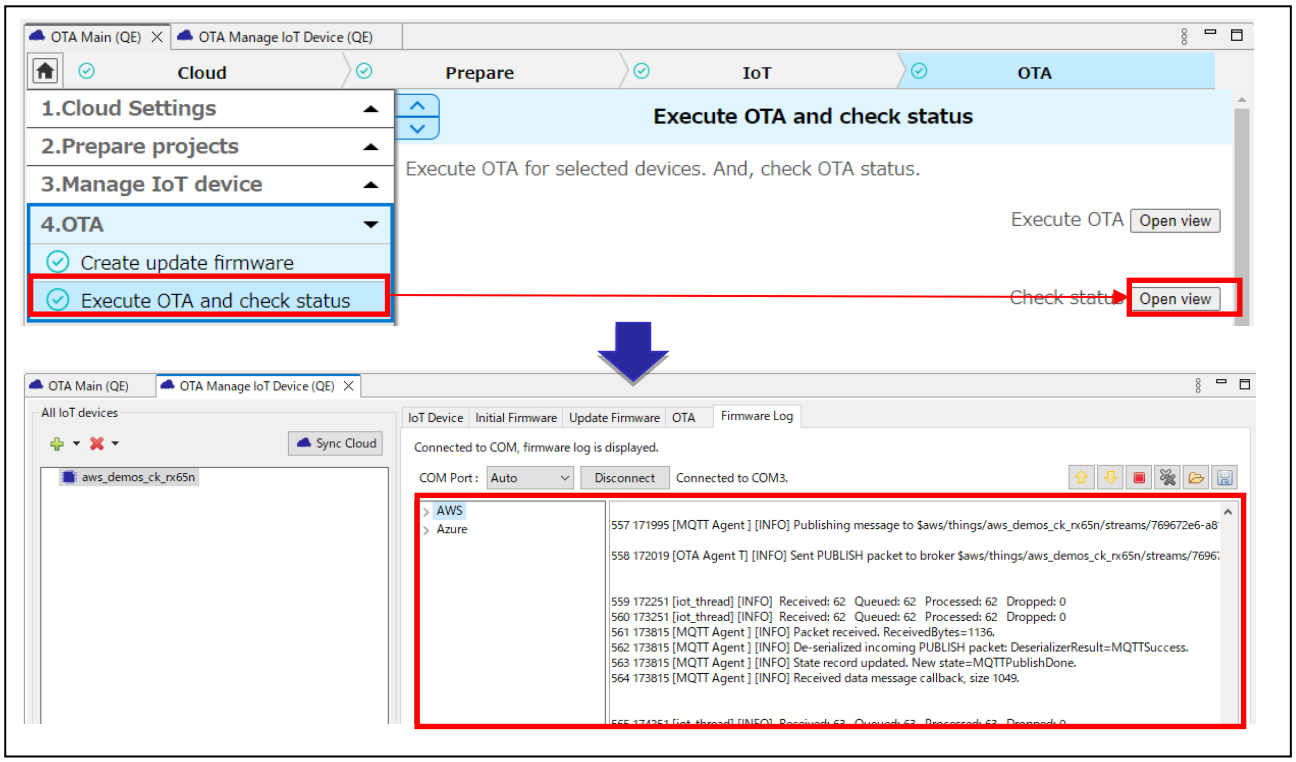

Figure 4-64 Check Firmware Log

After the download is completed, the firmware is checked. If there are no problems, the IoT device will restart and the updated firmware will start running.

Confirm that OTA succeeded by checking the indicated version in the firmware log. Find the version simply by searching the contents in the menu on the left side of the firmware log.

For AWS: [AWS > OTA > Version (x)]

For Azure: [Azure > Version (x)]

The version number will differ before and after OTA execution, indicated that the version has been updated.

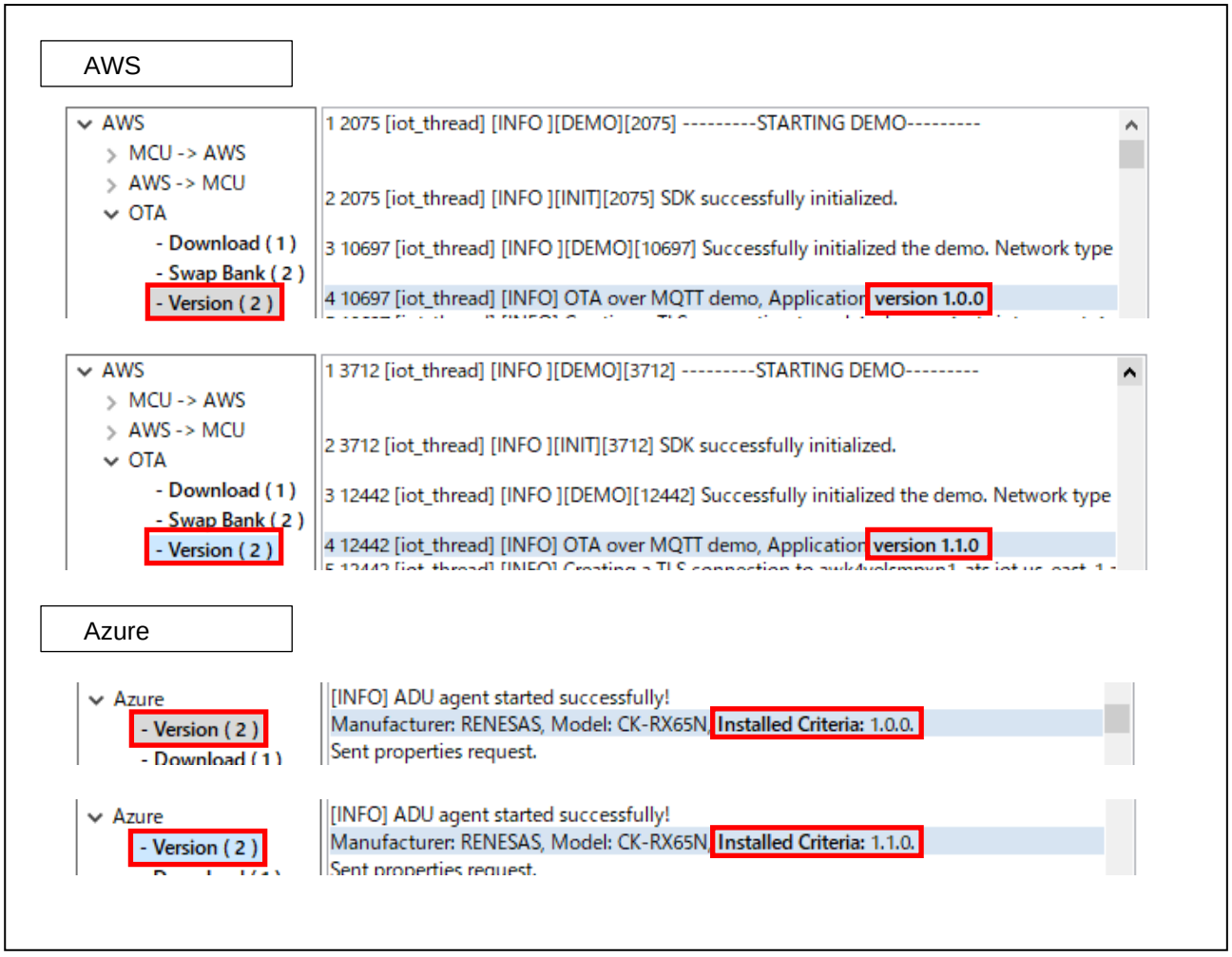

Figure 4-65 Firmware Log

After the update is completed, return to the “OTA” tab and check to see that the OTA job is complete on the cloud side.

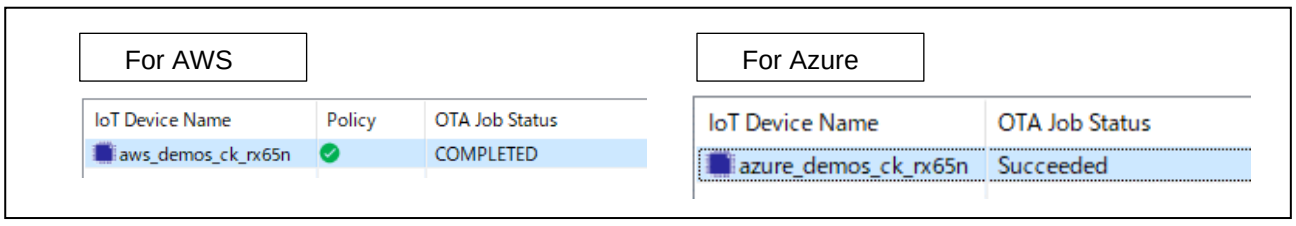

Figure 4-66 Job Status after OTA Execution

5. Help Function

You can check the details of the functions of “QE for OTA” from the Help tab in e² studio.

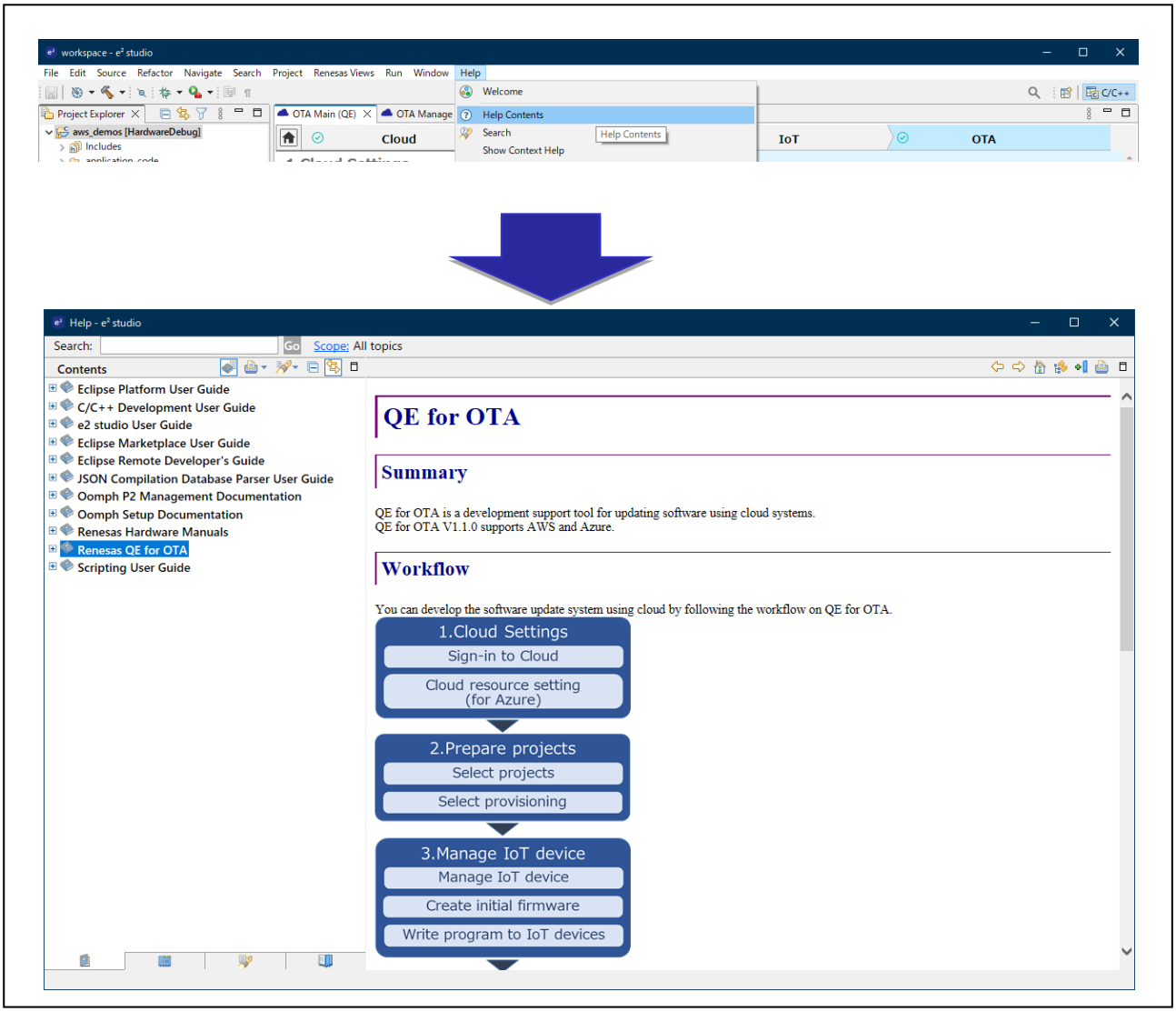

Figure 5-1 Help Function

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun. 30. 23	–	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

© 2023 Renesas Electronics Corporation. All rights reserved.