RX Family

AWS Cloud Connectivity for MCU Firmware Update Over-the-Air on RX65N Cloud Kit Board with Wi-Fi DA16600

Introduction

This document provides instructions for running the AWS Cloud Connectivity for MCU Firmware Update Over-the-Air project on CK-RX65N v2 using Wi-Fi DA16600, utilizing the MCU firmware update command of Wi-Fi DA16600 module.

Target Device

RX Family

- RX600 Series
  - RX65N Group

Hardware

- CK-RX65N v2

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Note: This demo project has also been successfully reused on the FPB-RX140 board. However, some module configurations must be adjusted to ensure compatibility with the FPB-RX140 device and its memory constraints.

Related Documents

[1] Firmware Integration Technology User's Manual (R01AN1833)

[2] RX Family FWUP Module Using Firmware Integration Technology (R01AN6850)

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

[4] CK-RX65N v2 – User's Manual (R20UT5366)

[5] US159-DA16600EVZ Evaluation Board Manual (R15UZ0006)

[6] RX Family US159-DA16XXXMEVZ Wi-Fi Control Module Using Firmware Integration Technology (R01AN7173)

Contents

1. Overview

This demo project shows the integration of AWS Cloud Connectivity for MCU Firmware Update Over-the-Air on the CK-RX65N v2 development board, utilizing the MCU firmware update command of Wi-Fi DA16600 module for seamless wireless communication.

Key Steps in the Project:

- **Prepare an AWS Account and S3 Bucket:** Set up an AWS account and create an S3 bucket to serve as the server for uploading firmware.
- **Generate Key Pairs and Certificates:** Generate a public key and a private key to create both the initial firmware and the firmware used for updates.
- **Generate and Upload the Firmware File:** Generate the new firmware to be flashed onto the board via OTA and upload it to the Amazon S3 bucket.
- **Generate the initial firmware:** Configure and build the initial firmware, then generate the firmware file using the designated tool.
- **Execute the Demonstration Project:** Execute the demo project to validate the OTA process.

The following section provides a network stack related structure for Firmware Update Over-the-Air (OTA) demonstration.

DA16600 Wi-Fi Module OTA Implementation: This module provides on-chip OTA command functionality by connecting to AWS infrastructure for direct firmware downloads. The retrieved firmware is then deployed to the connected MCU host. The demonstration showcases offloaded HTTP and firmware download processes, where firmware file URLs are securely obtained and pre-signed through web browser authentication.

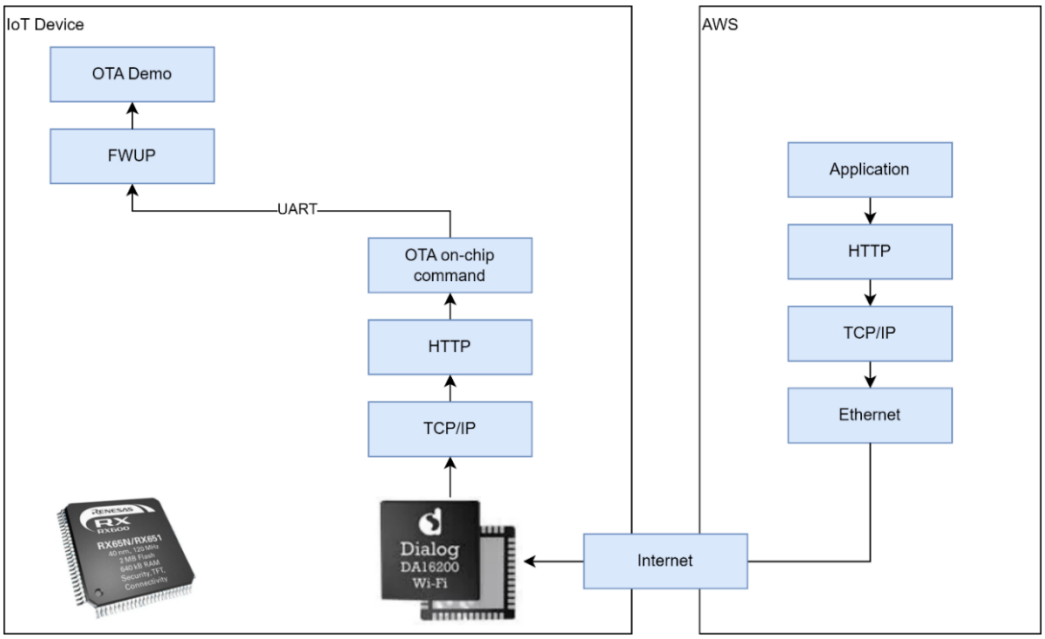

Figure 1.1 OTA Demonstrates with Wi-Fi DA16600 OTA On-Chip Commands

1.1 Workflow

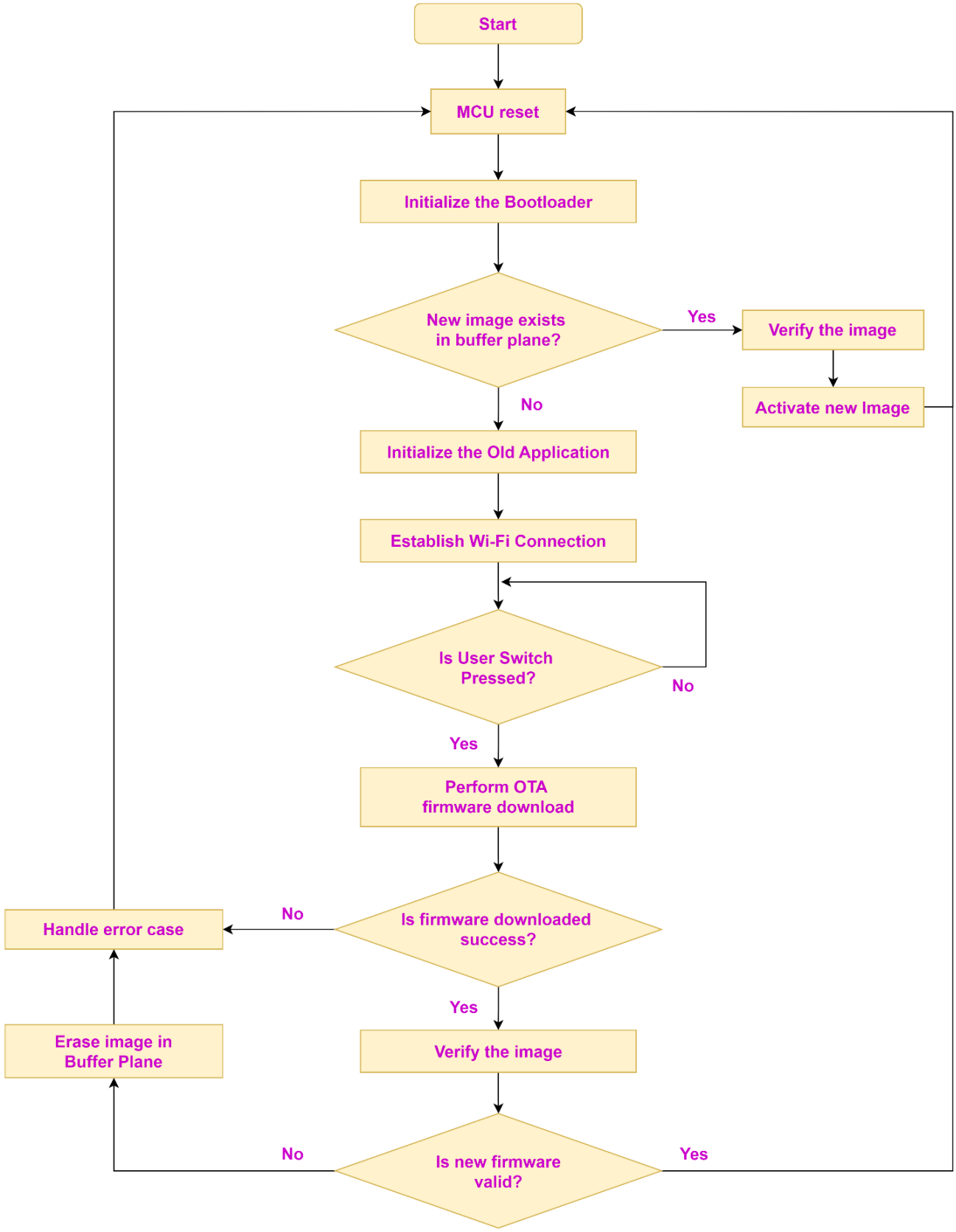

Figure 1.2 End-to-End OTA Workflow

1.2 Operation Confirmation Conditions

Demo project operations have been confirmed in the following conditions.

Table 1.1 Operation Confirmation Conditions

Item	Description
MCU	R5F565NEHxFB
Board	CK-RX65N v2 (Product no.: RTK5CK65N0S08001BE)
IDE (Integrated Development Environment)	Renesas Electronics e^2 studio 2025-07
C compiler	Renesas Electronics CC-RX V3.07.00
Firmware programming tool	Renesas Flash Programmer V3.20.00
Firmware update module	RX Family FWUP Module Using Firmware Integration Technology R01AN6850EJ0204
Python	Python 3.12.7
Keygen tool	Win64 OpenSSL v3.0.12
DA16XXX Wi-Fi control module	RX Family US159-DA16XXXMEVZ Wi-Fi Control Module Using Firmware Integration Technology (R01AN7173EU0140)
Revision of the DA16XXX module	Rev. 1.40
SDK (Software Development Kit)	DA16200/DA16600 SDK V3.2.9.2

1.3 Equipment List

The following lists the equipment required for the demo projects.

Table 1.2 Equipment List

Item	Description
Board	CK-RX65N v2 https://www.renesas.com/ck-rx65n
Wi-Fi DA16600 module	PMOD Expansion Board for DA16600MOD US159-DA16600EVZ - Ultra-Low-Power Wi-Fi + Bluetooth Low Energy Combo Pmod Board
USB TTL device x1	Check the DA16600 firmware and verify the status while the project is running. (*2.2.3 Check DA16600 Wi-Fi SDK Version)*
Micro USB Type-B cable x 1	Connect USB port on the base board to a PC for debugging purposes.
USB Type-C to Type-C cable x 1	One end is connected to ***J10*** (USB Serial Connector) on the board, and the other end is plugged into the PC/Laptop to print logs on ***Tera Term***(*).
Jumper pin x 3	It is used to enable debugging mode.

(*) : If your PC/Laptop does not have it installed yet, please install it according to the instructions 2.2.1.3 Installing Tera Term

1.4 Sample Project Code Sizes

The tables below show the ROM, RAM, and maximum stack sizes for the sample projects included in the package associated with this application note. The values in the table below have been confirmed under the following conditions:

Compiler version: Renesas Electronics C/C++ Compiler for RX Family V3.07.00

CC-RX

- Optimization level: Size and execution speed (-Odefault)
- Delete variables/functions that have never been referenced (optimize=symbol_delete)

Table 1.3 ROM, RAM for Sample Project

ROM, RAM, and Stack Codesize			
Device	**Category**	**Memory Used (byte)**	**Remarks**
RX65N	ROM	34912	boot_loader_ck_rx65n_v2
	RAM	9455	
	ROM	49396	ota_da16600_ck_rx65n_v2
	RAM	52877	

2. Demo Project Setup

2.1 Hardware Setup

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

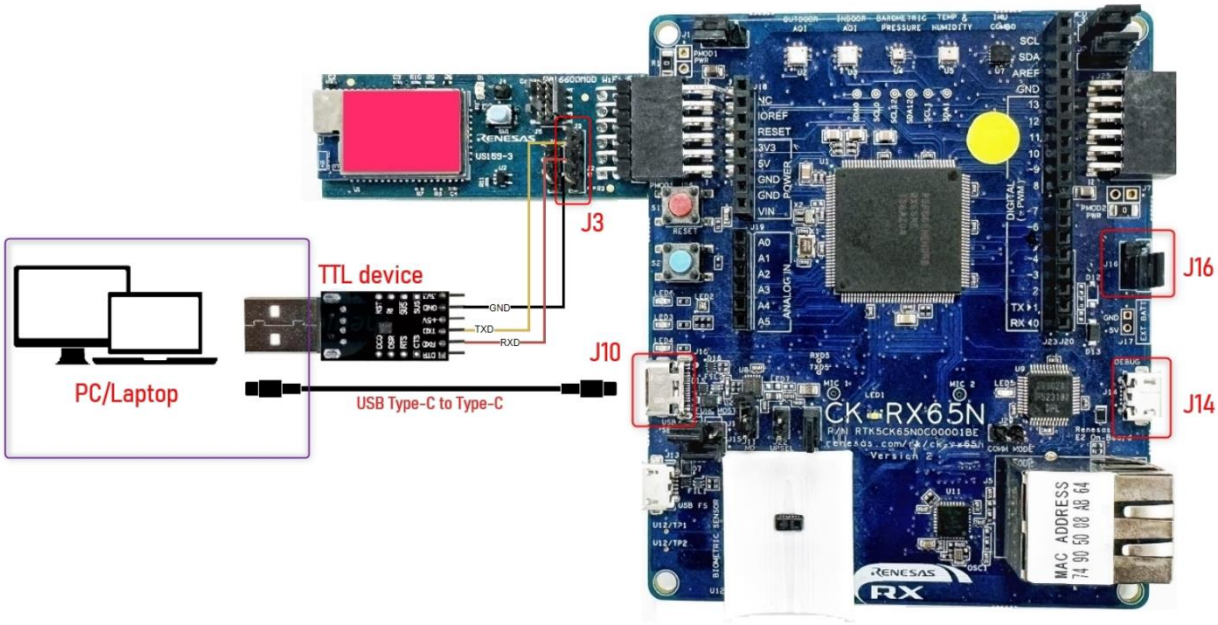

Figure 2.1 Hardware Connection

The following describes how to connect hardware.

On CK-RX65N V2:

- Position the jumper on **pins 1-2** of **J16** to enable debugging mode.
- Connect the **DA16600 PMOD** to **PMOD1** on the base board.
- Connect **J10** on the base board to a PC using a USB cable for serial log.
- Connect **J14** on the base board to a PC using a USB cable for debugging.

On Wi-Fi DA16600:

- Connect the **TXD pin** of the **UART TTL** to **pin 2** of **J3**(*) of the DA16600 Wi-Fi module.
- Connect the **RXD pin** of the **UART TTL** to **pin 3** of **J3**(*) of the DA16600 Wi-Fi module.
- Connect the **GND pin** of the **UART TTL** to **pin 4** of **J3**(*) of the DA16600 Wi-Fi module.

(*): Please note that the RXD, TXD, and GND wires must be connected in the correct order to the ***J3 pin*** (it is very easy to mistake ***J3*** for ***J1***, which can cause errors and potentially damage the board or chip).

2.2 Software Setup

2.2.1 Installing Tool

2.2.1.1 Install Python

Python generates initialization firmware from bootloader and application projects, and application firmware from the new application project.

Follow the steps below to install Python:

(1) Access the Python download web site.

https://www.python.org/downloads/

(2) Download the Python 3.12.7 installer.

Click the **Download** link for Python 3.12.7.

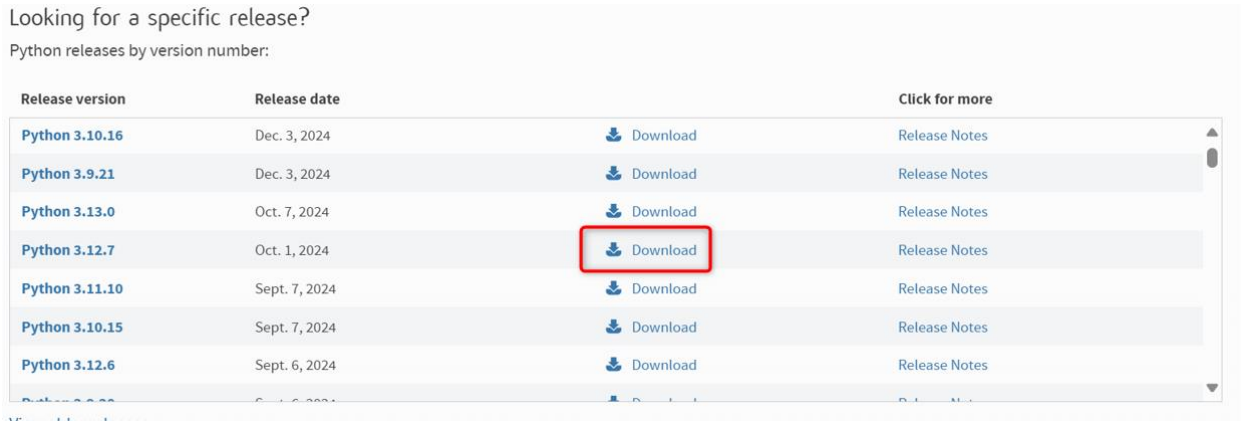

Figure 2.2 The Options for Installing the Release Version of Python

Download the installer for the operating system you are using.

Figure 2.3 Python Windows Installer

2.2.1.2 Installing OpenSSL

OpenSSL is a tool used to generate the cryptographic key pair required for firmware encryption and decryption during both initialization and application firmware creation. OpenSSL can generate the following keys for use in the firmware update process:

- **Private key**: Used to encrypt the firmware and ensure its integrity.
- **Public key**: Used by the bootloader to decrypt and verify the firmware during update.

Follow the steps below to install and configure OpenSSL for this purpose.

(1) Access the Win32/Win64 download website for OpenSSL

https://slproweb.com/products/Win32OpenSSL.html

(2) 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 so 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

Figure 2.4 The Options for Installing the Release Version of OpenSSL

(3) Run the *Installer* and follow the *Prompts* to install OpenSSL.

Select the option to copy the OpenSSL DLLs to the OpenSSL binaries directory.

(4) From the *Start Menu*, open the *Win64 OpenSSL Command Prompt*.

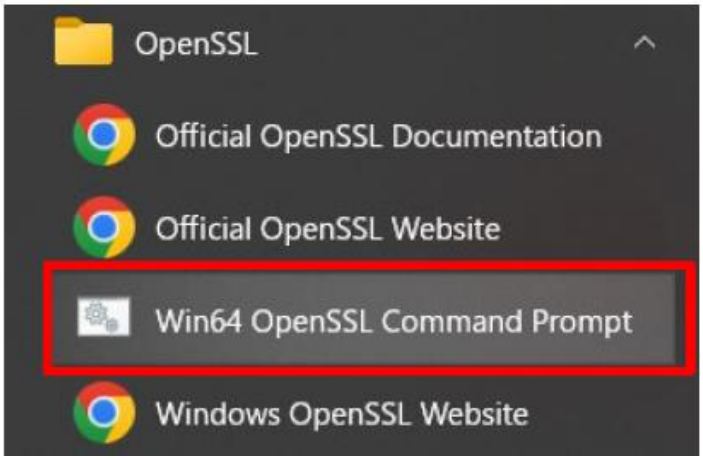

Figure 2.5 OpenSSL Windows (64-bit)

(5) Confirm the OpenSSL command from the *Command Prompt*.

Execute the following command and confirm that version information appears.

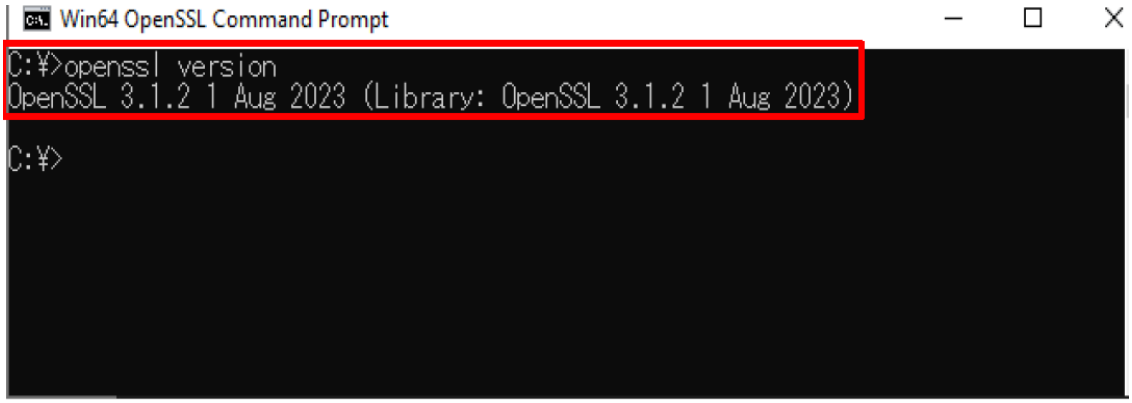

Figure 2.6 Checking OpenSSL Version

2.2.1.3 Installing Tera Term

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

(1) Access the *Tera Term Download* site.

https://github.com/TeraTermProject/teraterm/releases

(2) Download the *Tera Term Installer*.

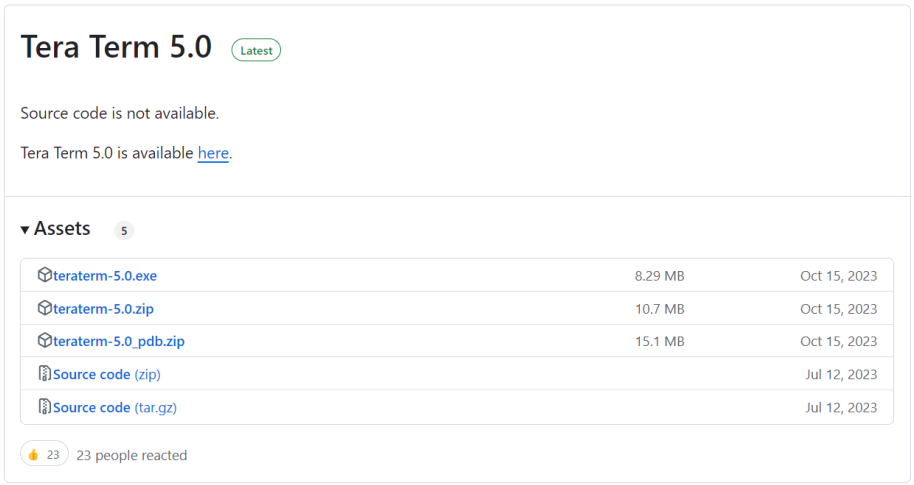

Figure 2.7 Tera Term Version

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

2.2.1.4 Installing Renesas Flash Programmer

Renesas Flash Programmer (RFP) is a utility provided by Renesas that allows users to write firmware to support Renesas MCUs via various interfaces such as USB, UART, or serial programming. It is an essential tool for flashing both the initial firmware and subsequent updates during development and production. Follow the steps below to install Renesas Flash Programmer on your computer.

(1) Access the *Renesas Download* web site.

https://www.renesas.com/software-tool/renesas-flash-programmer-programming-gui

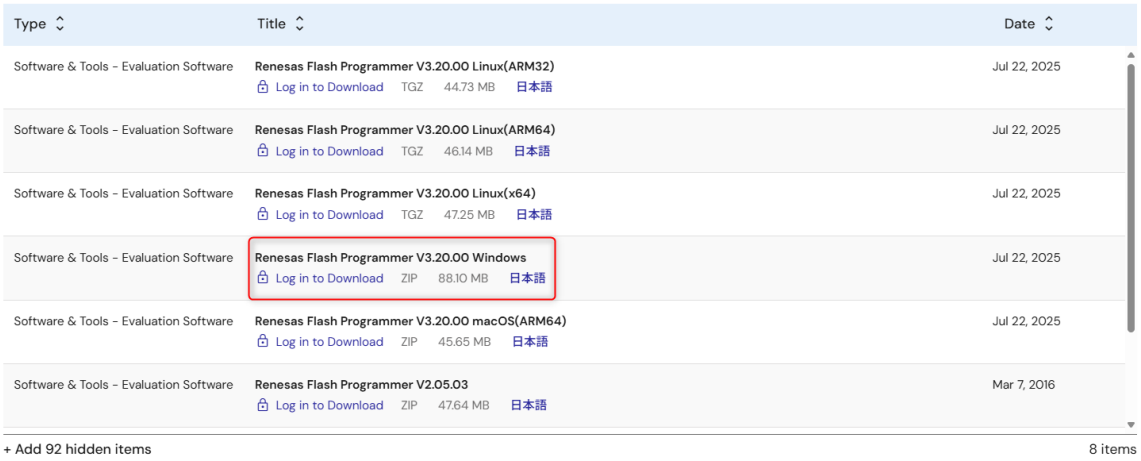

Figure 2.8 Renesas Flash Programmer

2.2.2 Terminal Software Setting

❖ With the UART TTL connection port of the DA16600 *(For check firmware DA16600 - Please note that the TTL device must be attached directly to the DA16600 device itself, not to the board)*:

(1) Open *Tera Term* select *New connection* and select *Serial* and the appropriate COM port for your *UART-to-USB* adapter, and click *OK*

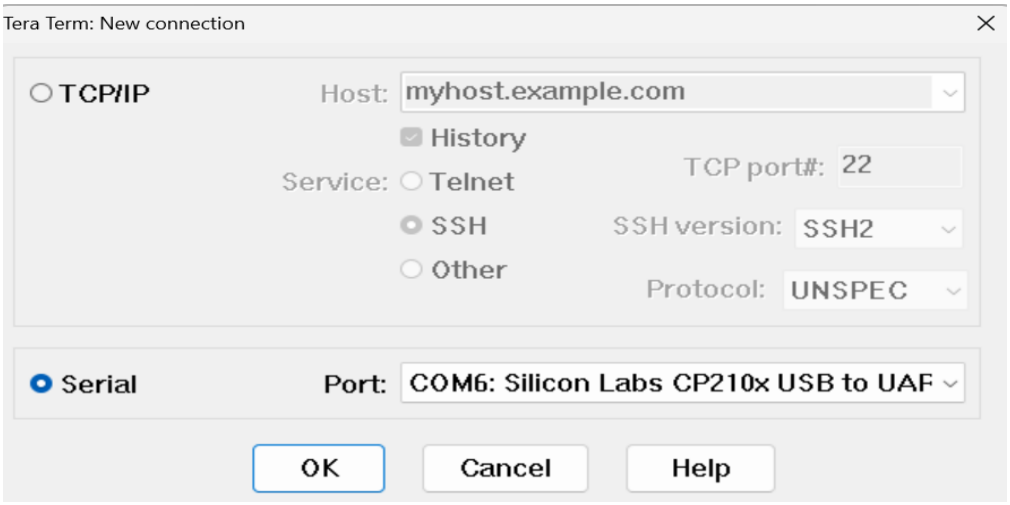

Figure 2.9 Tera Term Serial Connection

(2) Click *Setup* > *Terminal*…, in *"New-line"* section, set *"Receive"* as *AUTO*.

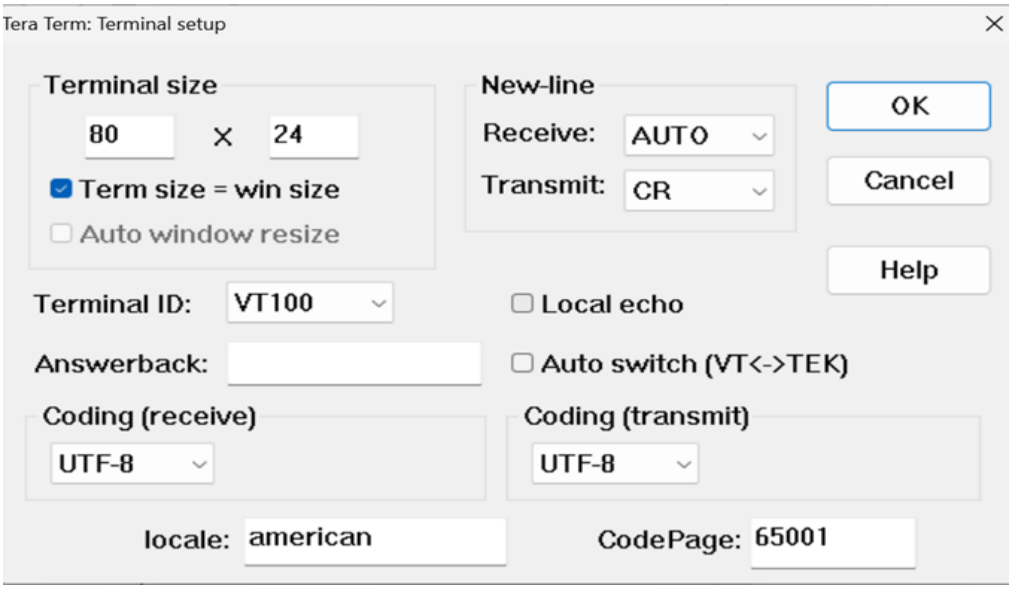

Figure 2.10 Terminal Setup for the UART TTL

(3) Click *Setup* > *Serial port…* and ensure that the speed is set to *230400*.

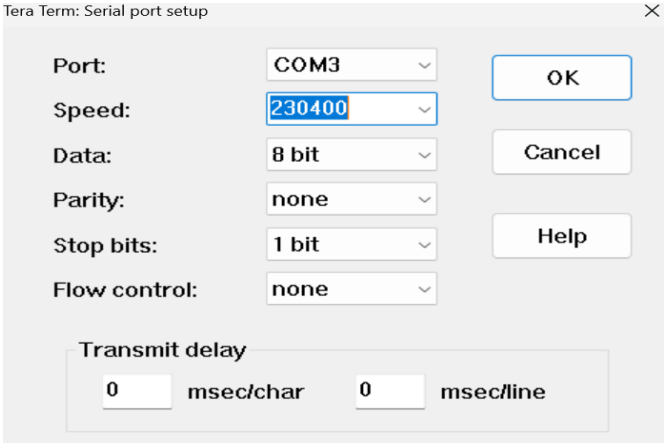

Figure 2.11 Serial Port Setup for UART TTL

(4) Verify the display output on the terminal.

Pressing *Enter* on the terminal will display the line *[/DA16600] #* on the screen.

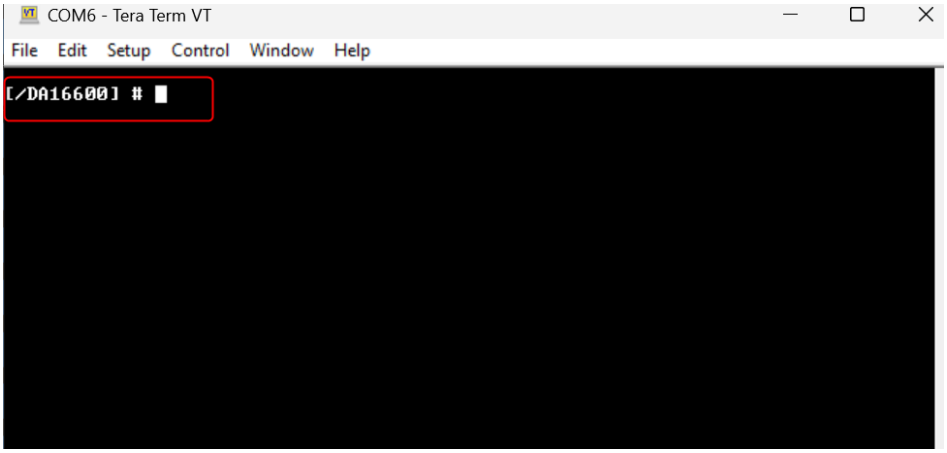

Figure 2.12 UART TTL Terminal

❖ With the USB serial *(USB Type-C)* connection of CK-RX65N v2:

(1) Open an additional Tera Term window, *select New Connection*, then choose Serial and the correct COM port for your *USB Serial* device, and click *OK*.

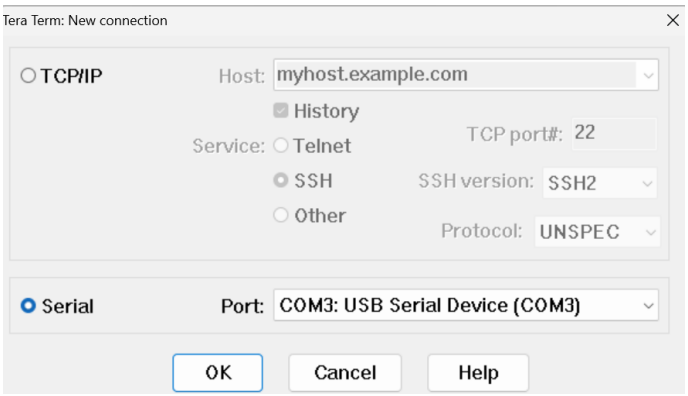

Figure 2.13 Select USB Serial Port

(2) Click *Setup* > *Terminal…*,*"New-line" Receive* as ***AUTO*** and *Transmit* as ***CR + LF***, then tick the ***"Local echo"*** option.

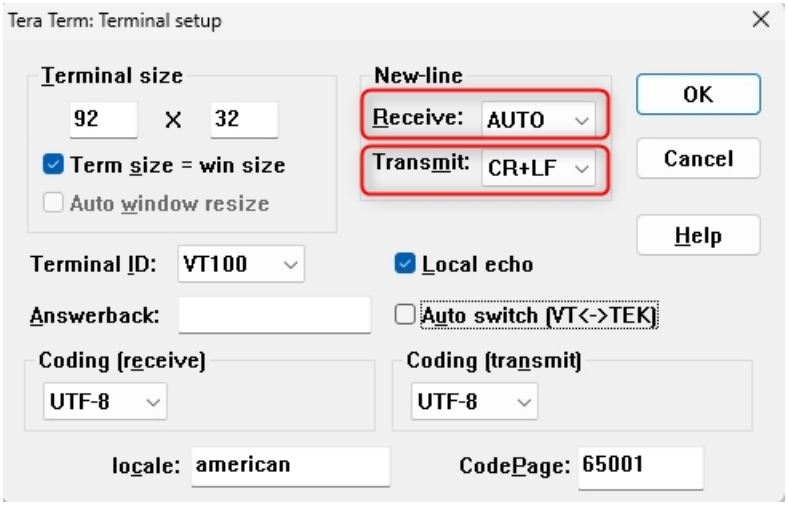

Figure 2.14 Terminal Setup

(3) Click *Setup* > *Serial port…* and ensure that the speed is set to *115200*.

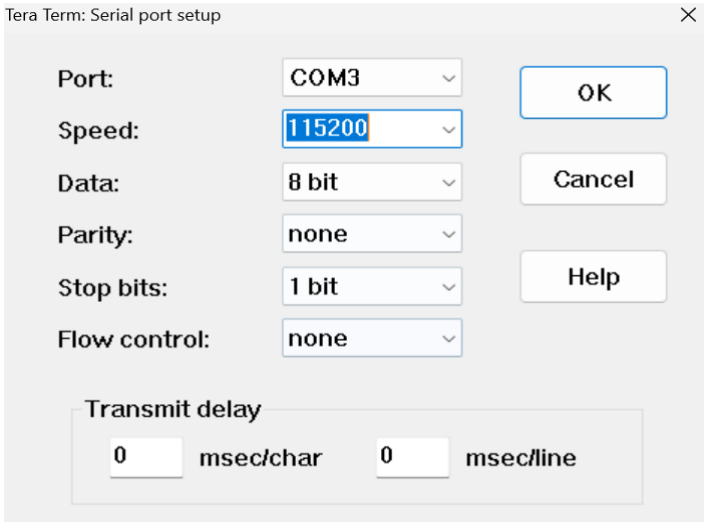

Figure 2.15 Terminal Setup for USB Serial Port

(4) Verify the display output on the terminal.

The terminal output will appear here during the demo execution.

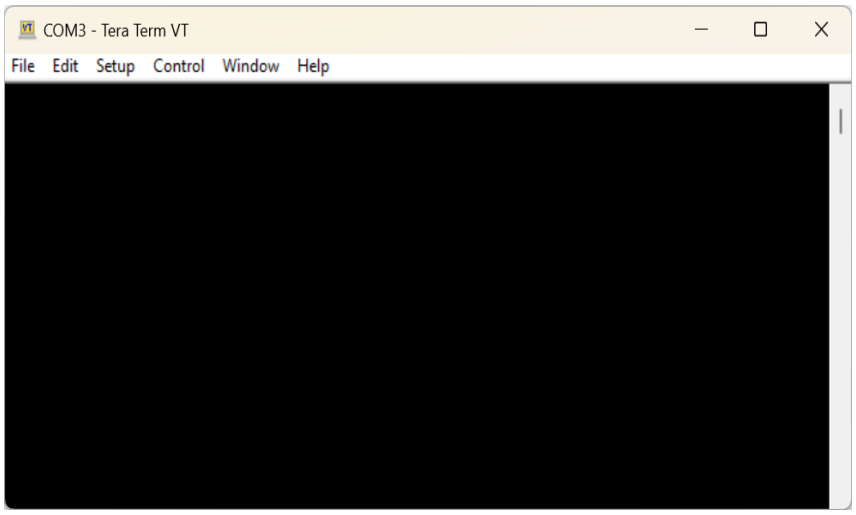

Figure 2.16 CK-RX65N v2 Serial Terminal

2.2.3 Check DA16600 Wi-Fi SDK Version

After completing the setup steps in 2.2.2 Terminal Software Setting, proceed with the steps below to check the SDK version of the DA16600.

(1) On the *UART TTL* terminal in *Tera Term*, type the command *"ver "*.

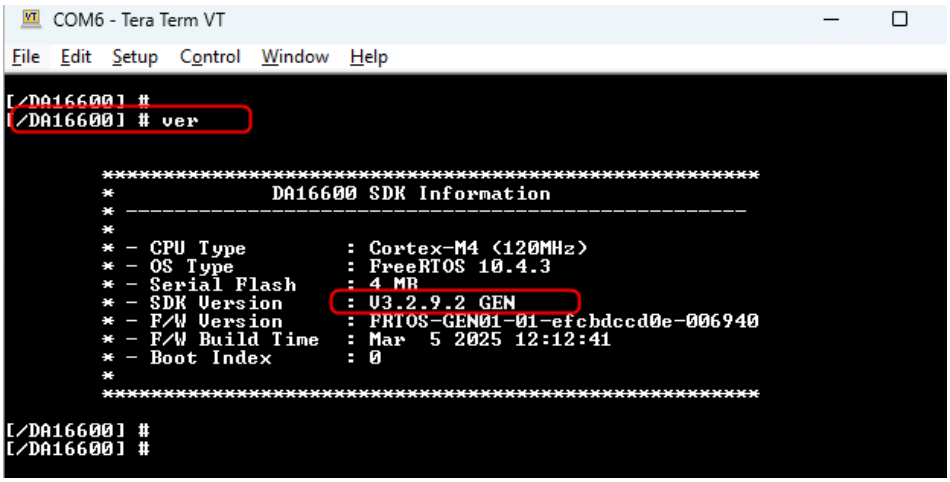

Figure 2.17 Check SDK Version

(2) Check the SDK version. If the current version is ***v3.2.9.2*** or higher, you can proceed to the next step; otherwise, follow the steps below to upgrade:

- Download the firmware using this link: https://www.renesas.com/document/sws/da16200-da16600-freertos-sdk-image-v3292
- Flash the new firmware via Tera Term by following the instructions provided in the https://www.renesas.com/document/qsg/um-wi-056-da16200-da16600-freertos-getting-started-guide (UM-WI-056 DA16200 DA16600 FreeRTOS Getting Started Guide) under *Section 4.5.2 Using Macro Script of Tera Term.*

2.2.4 Generate Key Pairs and Certificates

This section will generate a public key and a private key to create the initial firmware and the firmware used for updates.

To do this, open OpenSSL and enter the ***commands highlighted*** in yellow to generate the firmware verification keys.

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

using curve name prime256v1 instead of secp256r1

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

read EC key

writing EC key

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

read EC key

writing EC key

2.2.5 Project Description

The demo project is structured into multiple components to support the firmware update process. Each component plays a specific role in demonstrating the OTA update mechanism on the RX65N platform.

Table 2.1 Folder Structure of OTA Sample Program

Folder name, file name	Explanation
ota_rx-project	Root folder
├README.md	Instruction guide
├Common	Common directory
│ ├application_bootloader_common	
│ ├application_ota_common	
│ ├flash_module	
│ ├helper_common	
│ ├key_common	
│ └***RenesasImageGenerator***	Generate image file (.rsu/initial_firm.mot)
├Document	Application Note
└Projects	
├***boot_loader_ck_rx65n_v2***	The bootloader responsible for handling the firmware update process.
│ ├e2studio_ccrx_bootloader	This project utilizes the CC-RX toolchain(*)
│ └e2studio_gcc_bootloader	This project utilizes the GCC toolchain(**)
└***ota_da16600_ck_rx65n_v2***	The existing application that the MCU runs **before** the firmware update process begins.
├e2studio_ccrx	This project utilizes the CC-RX toolchain(*)
└e2studio_gcc	This project utilizes the GCC toolchain(**)

(*) : For CC-RX toolchain users:

- Build: e2studio_ccrx and e2studio_ccrx_bootloader projects.
- Ignore: e2studio_gcc and e2studio_gcc_bootloader projects.

(**) : For GCC toolchain users:

- Build: e2studio_gcc and e2studio_gcc_bootloader projects.
- Ignore: e2studio_ccrx and e2studio_ccrx_bootloader projects.

2.2.6 Importing The Demo Project

(1) Clone the demo project

(2) Extract the demo project

(3) Start e² studio

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

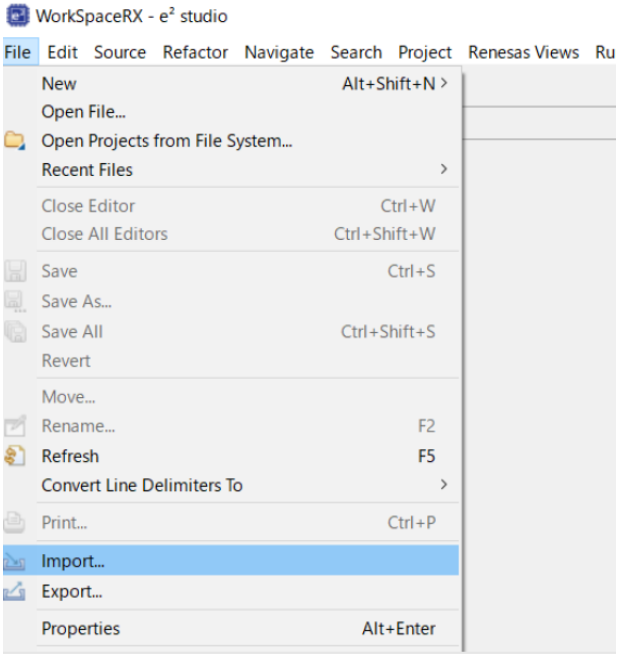

Figure 2.18 Importing the Project

(5) Select *Existing Projects* into *Workspace*

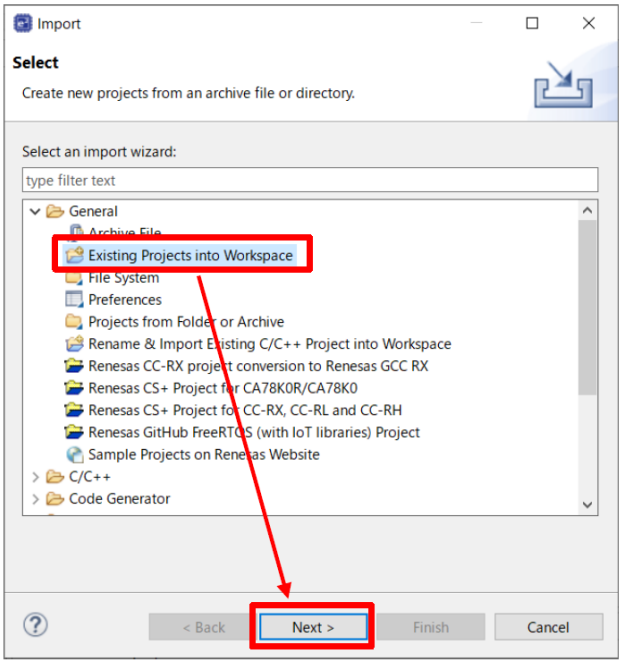

Figure 2.19 Select Existing Projects into Workspace

(6) In *Select root directory*, choose the folder extracted, select the check boxes for the following projects, and then click *Finish*

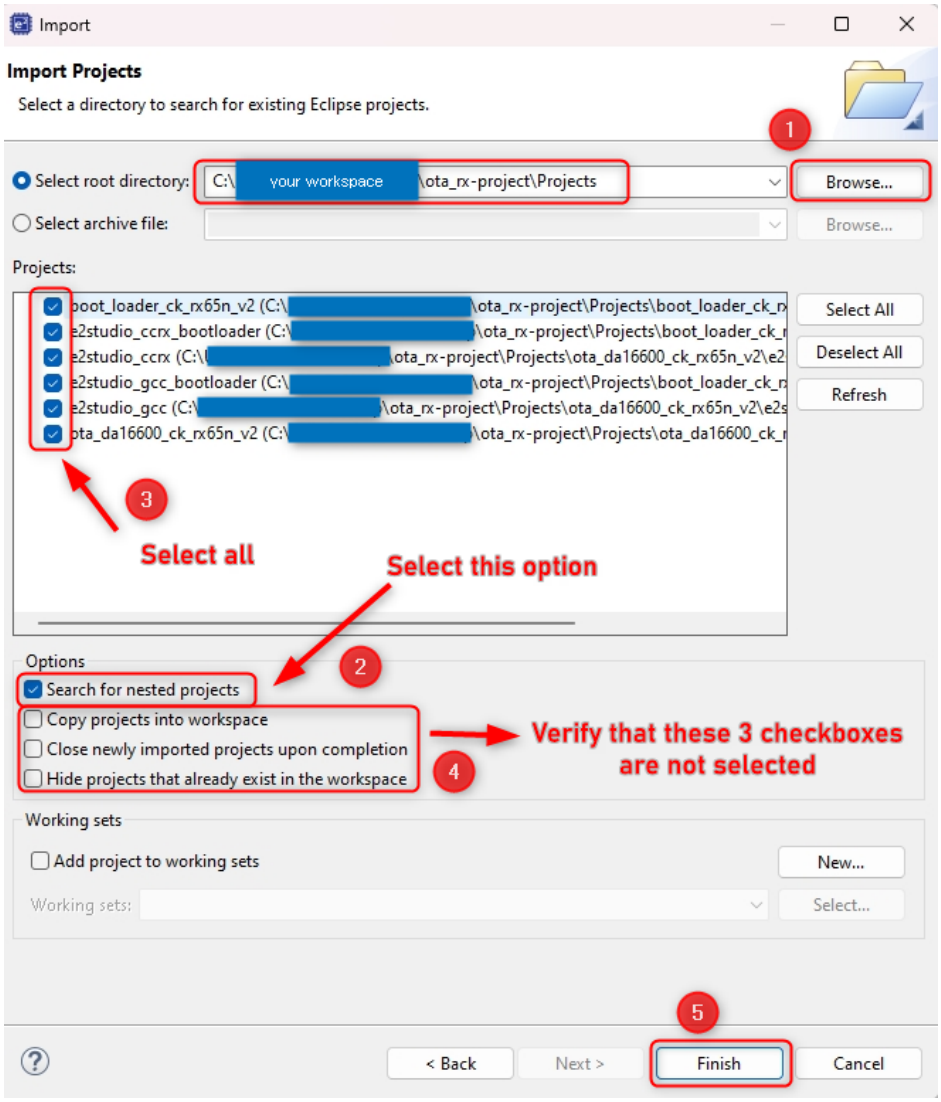

Figure 2.20 Complete Import Project

(7) After importing the parent folder, select the four subfolders shown in the image below

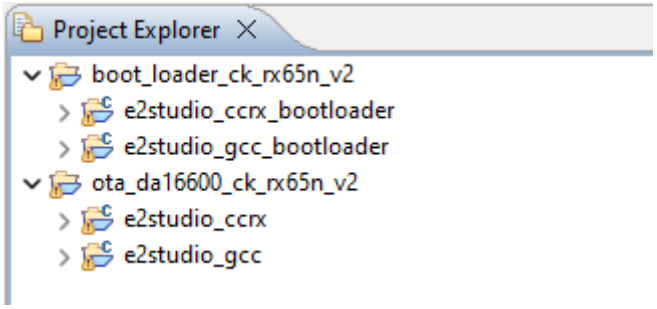

Figure 2.21 Subfolders

The functionality of all 4 demo projects is detailed in Project Description

2.2.7 Create Firmware Initialization

2.2.7.1 Update Public Key

Since the Renesas Image Generator (mentioned in ***2.2.5 Project Description***) is used to create the initial firmware, the public key must be pasted into the bootloader project to match the private key located in the Renesas Image Generator folder. This ensures that the initial firmware can be generated successfully. Follow the instructions below to complete this setup.

1. Copy the contents of the **secp256r1.publickey** file you created in ***2.2.4 Generate Key Pairs and Certificates***
2. Paste the public key into **CODE_SIGNER_PUBLIC_KEY_PEM** in *Common/key_common/code_signer_public_key.h*

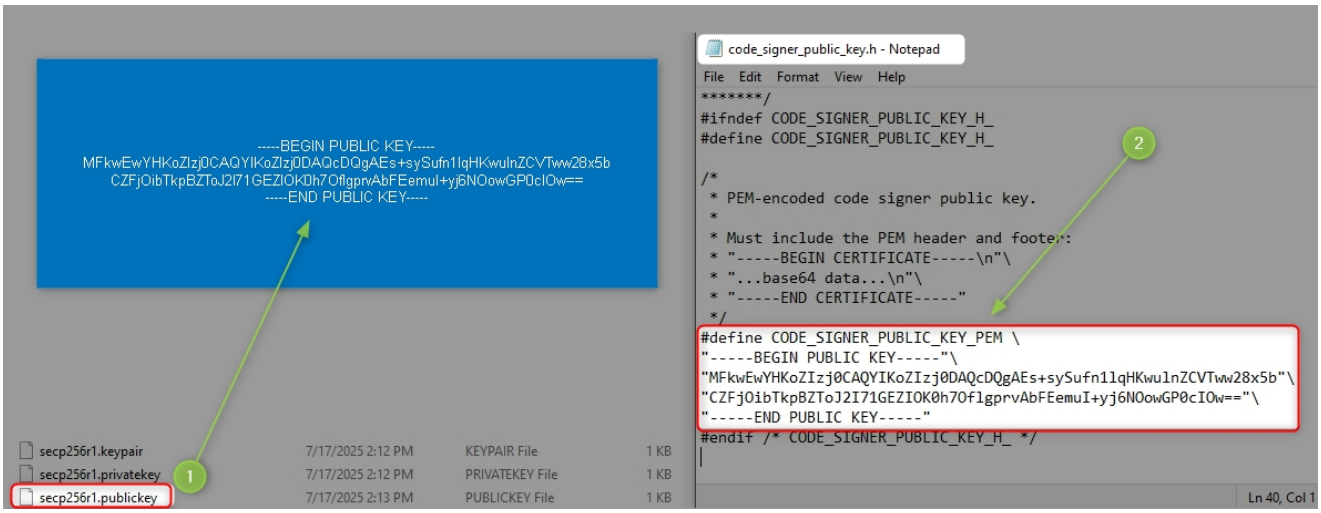

Figure 2.22 Assign a Public Key

2.2.7.2 Configure Firmware Version, Wi-Fi, Firmware Type, and URL

Navigate to ***Common/helper_common/demo_config.h***

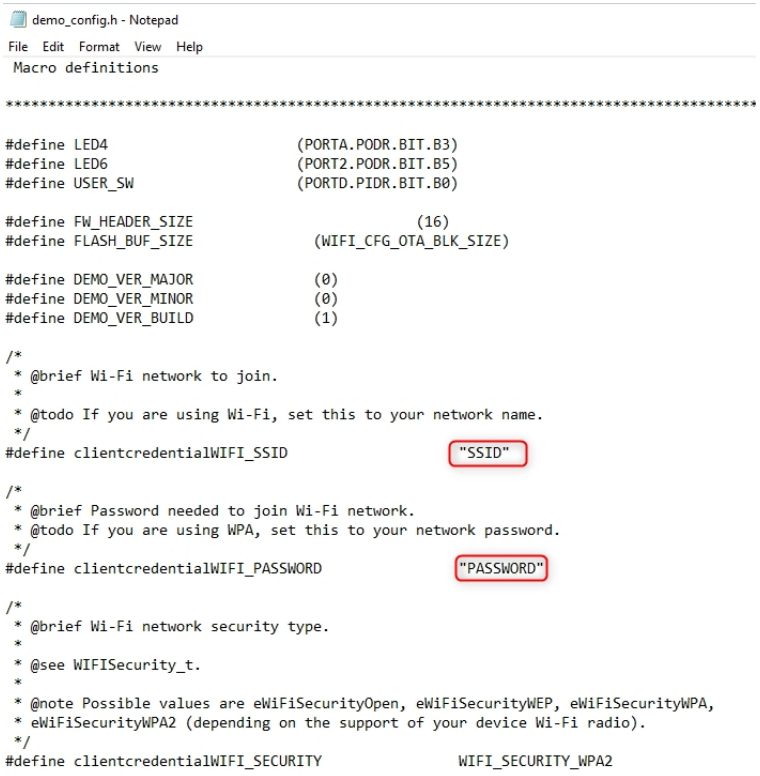

Figure 2.23 Configure Firmware Version, Wi-Fi Network, and Firmware URL

- Check the current firmware version.
- AP_WIFI_SSID: Set the access point name (SSID) of the local Wi-Fi network that the board will connect to.
- AP_WIFI_PASSWORD: Set the password for the local Wi-Fi network

2.2.7.3 Building the Project

After completing configuration as both section ***2.2.7.1*** and ***2.2.7.2***, open e^2 studio and sequentially click "*Build Project*" for ***e2studio_ccrx_bootloader*** and then for ***e2studio_ccrx***

2.2.7.4 Creating the Initial Firmware

This section is used to create the initial firmware ***(initial_firm_ccrx.mot)***, which is executed on the MCU before the Over-The-Air process begins. Follow the setup steps below to generate the initial firmware correctly:

(1) Place the following files in the *Renesas Image Generator* folder:

- The results of the building process in 2.2.7.3 Building the Project: ***e2studio_ccrx.mot*** and ***e2studio_ccrx_bootloader.mot***
- The private key created in 2.2.4 Generate Key Pairs and Certificates: **secp256r1.privatekey**

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

Open a command prompt, navigate to the Renesas Image Generator folder, for example:

❖ \RootFolder\Common\RenesasImageGenerator

Execute the following command to generate the file ***initial_firm_ccrx.mot***

```
python image-gen.py -iup e2studio_ccrx.mot -ip RX65N_DualBank_ImageGenerator_PRM.csv -o initial_firm_ccrx -ibp e2studio_ccrx_bootloader.mot -vt ecdsa -key secp256r1.privatekey
```

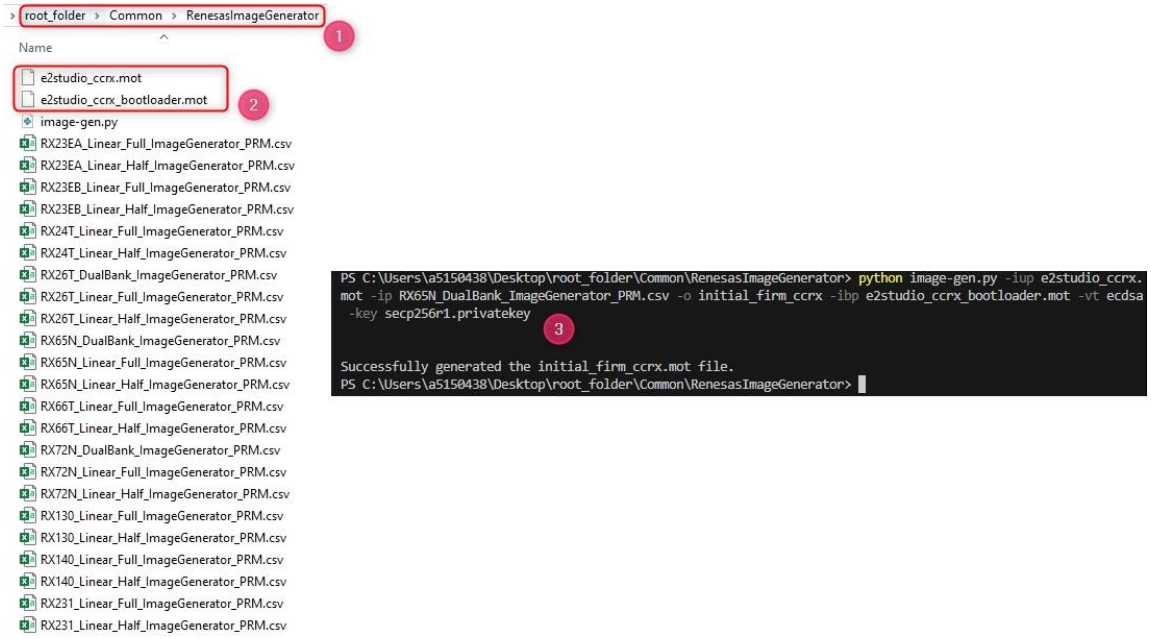

Figure 2.24 Create the Initial Firmware

❖ **Parameter explanation:**

- **-iup**: Input user program (.mot file for application firmware)
- **-ip:** Input parameter file (.csv with image generation settings)
- **-o:** Output file prefix (e.g., userprog.mot)
- **-ibp:** Input bootloader program (.mot file)
- **-vt**: Verification type (e.g., ecdsa for digital signature)

2.2.8 Create Firmware File and Upload to the Amazon S3 Bucket

2.2.8.1 Create Firmware File

(1) Changing the firmware version

Change the firmware version to a higher version. (Example: Because previous versions is 0.0.**1**, so the new version we can choose 1.0.1)

Please refer back to *2.2.7.2 Configure Firmware Version, Wi-Fi, Firmware Type, and URL* and update the firmware configuration to ensure the new version is properly recorded after the firmware update process.

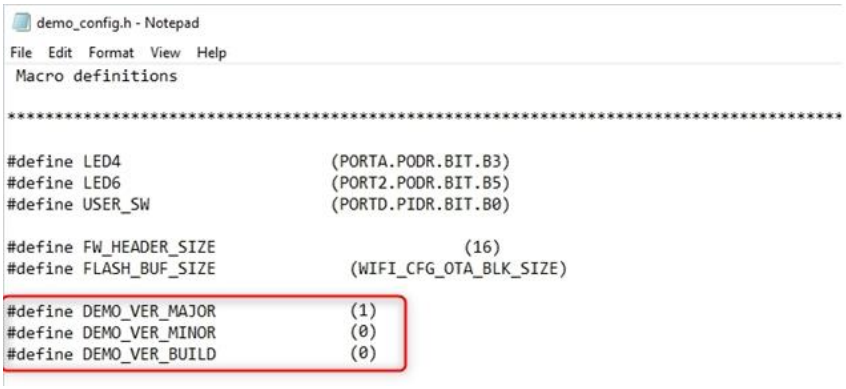

Figure 2.25 Setting New Version for Firmware

(2) Build ***e2studio_ccrx*** project and retrieve the ***e2studio_ccrx.mot*** file

(3) Use *Renesas Image Generator* to generate the updated firmware

Overwrite the ***e2studio_ccrx.mot*** file you retrieved in step (2) above into the *RenesasImageGenerator* folder to proceed with generating the RSU file with following command at the command promt:

```
python image-gen.py -iup e2studio_ccrx.mot -ip RX65N_DualBank_ImageGenerator_PRM.csv -o e2studio_ccrx_v100 -vt ecdsa -key secp256r1.privatekey
```

```
PS C:\Users\a5150438\Desktop\root_folder\Common\RenesasImageGenerator> python image-gen.py -iup e2studio_ccrx.mot
-ip RX65N_DualBank_ImageGenerator_PRM.csv -o e2studio_ccrx_v100 -vt ecdsa -key secp256r1.privatekey


Successfully generated the e2studio_ccrx_v100.rsu file.
```

Figure 2.26 Create the RSU File

2.2.8.2 Uploading Firmware to Amazon S3 Bucket

(1) Create Amazon S3 Bucket

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

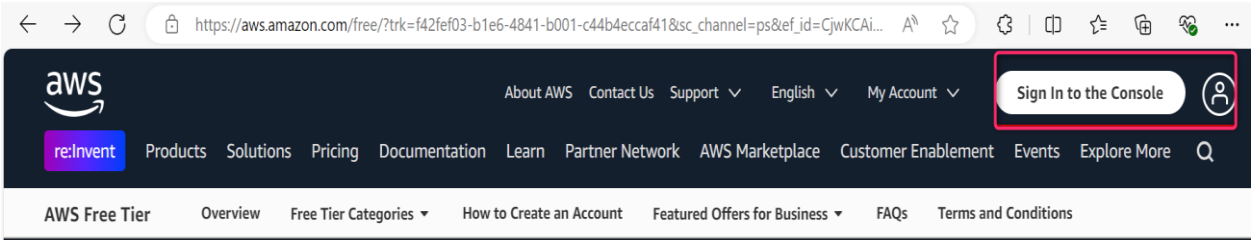

Figure 2.27 Sign-In to Console AWS

- Enter your email address or account ID, and then click **Next.**

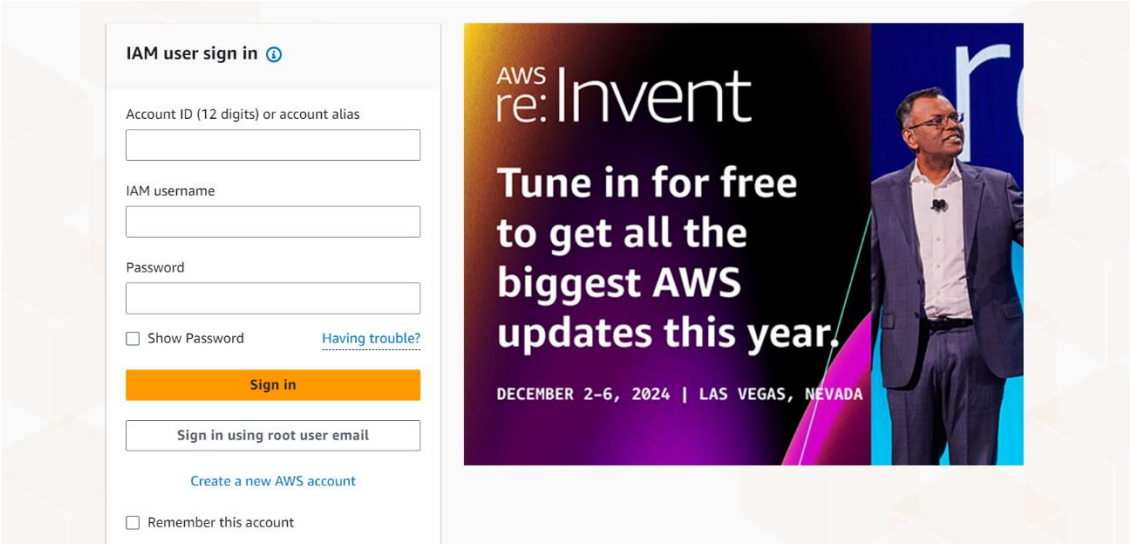

Figure 2.28 User Sign-In

If you are using the root account to sign in, select the "Sign in using root user email" option and enter the email address of the root account. If you are an IAM user, enter the **Account ID** (12-digit number or account alias), **IAM username**, and **Password** in the corresponding fields.

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

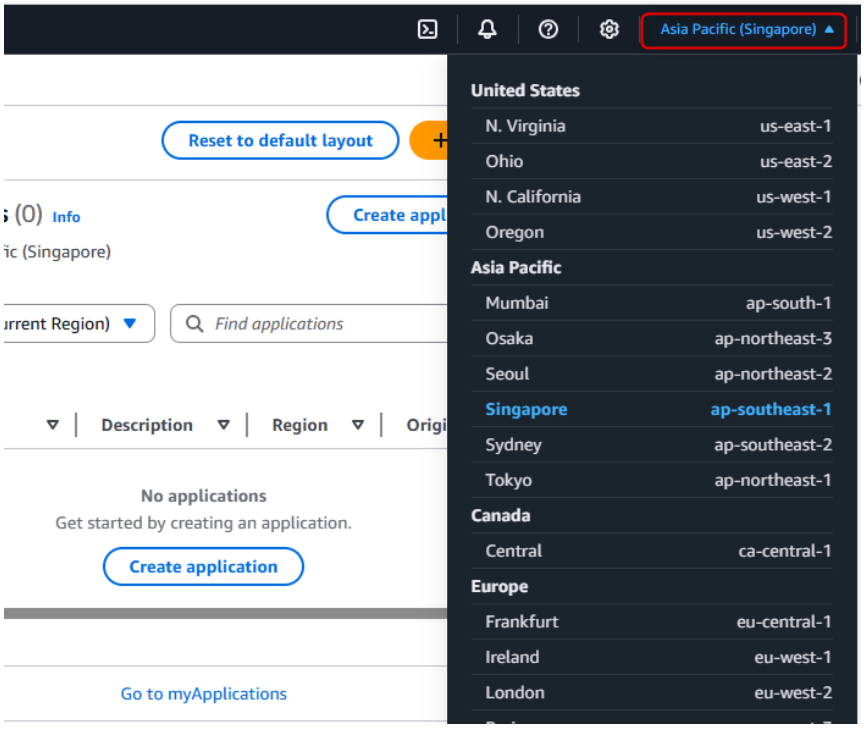

Figure 2.29 Setting Region in AWS

- From the **Services** menu, select **Storage** and then **S3**.

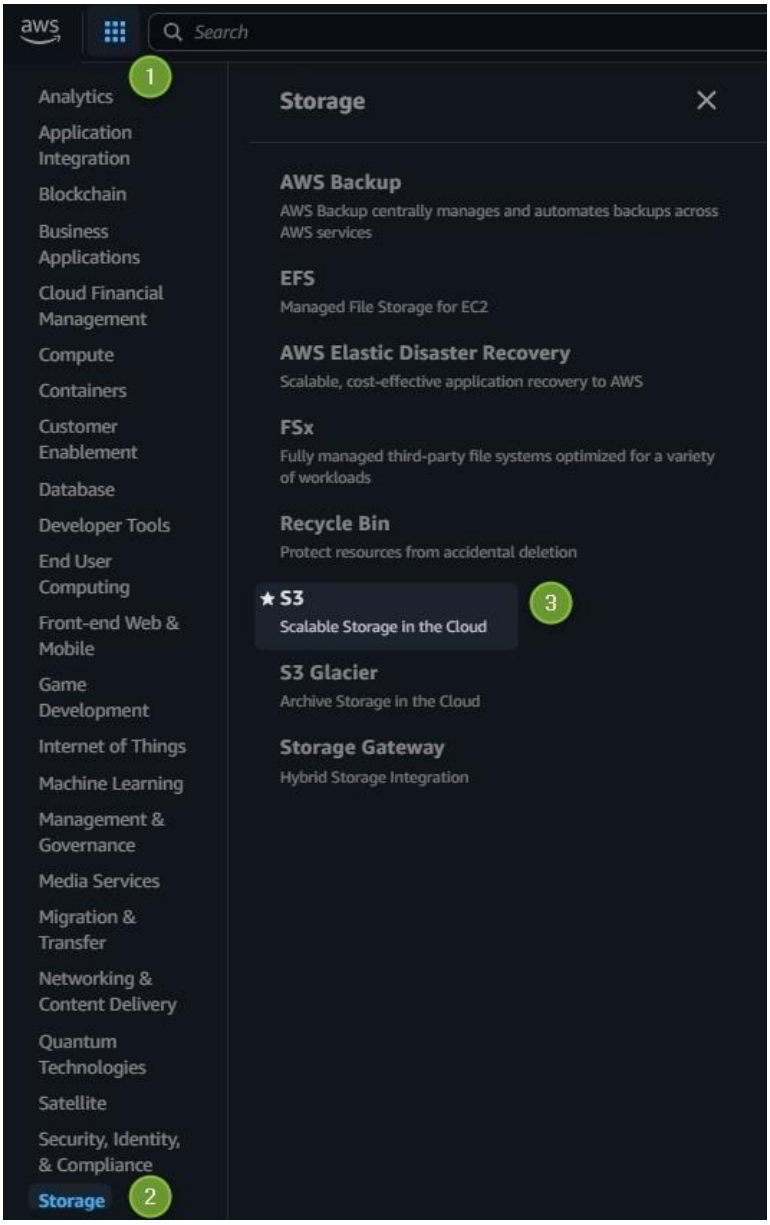

Figure 2.30 S3 AWS Bucket

- On the **Buckets** page, click the **Create bucket** button.

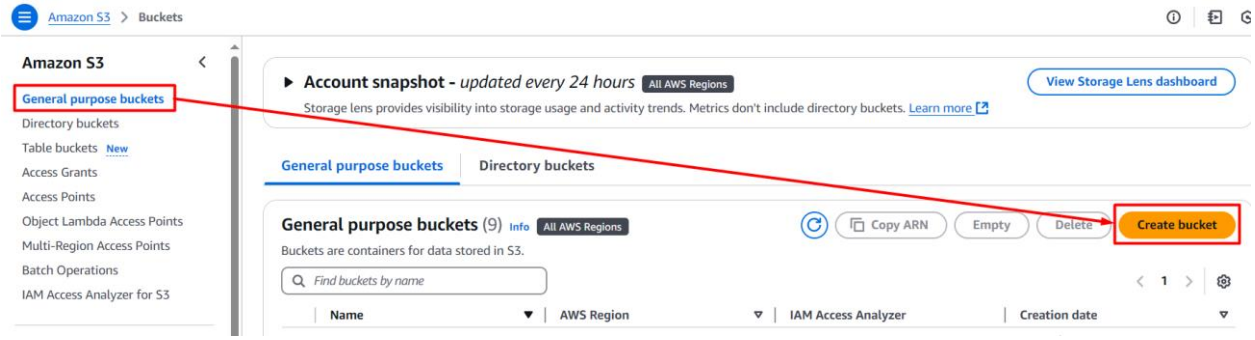

Figure 2.31 Create a Bucket

- Enter an S3 Bucket name.

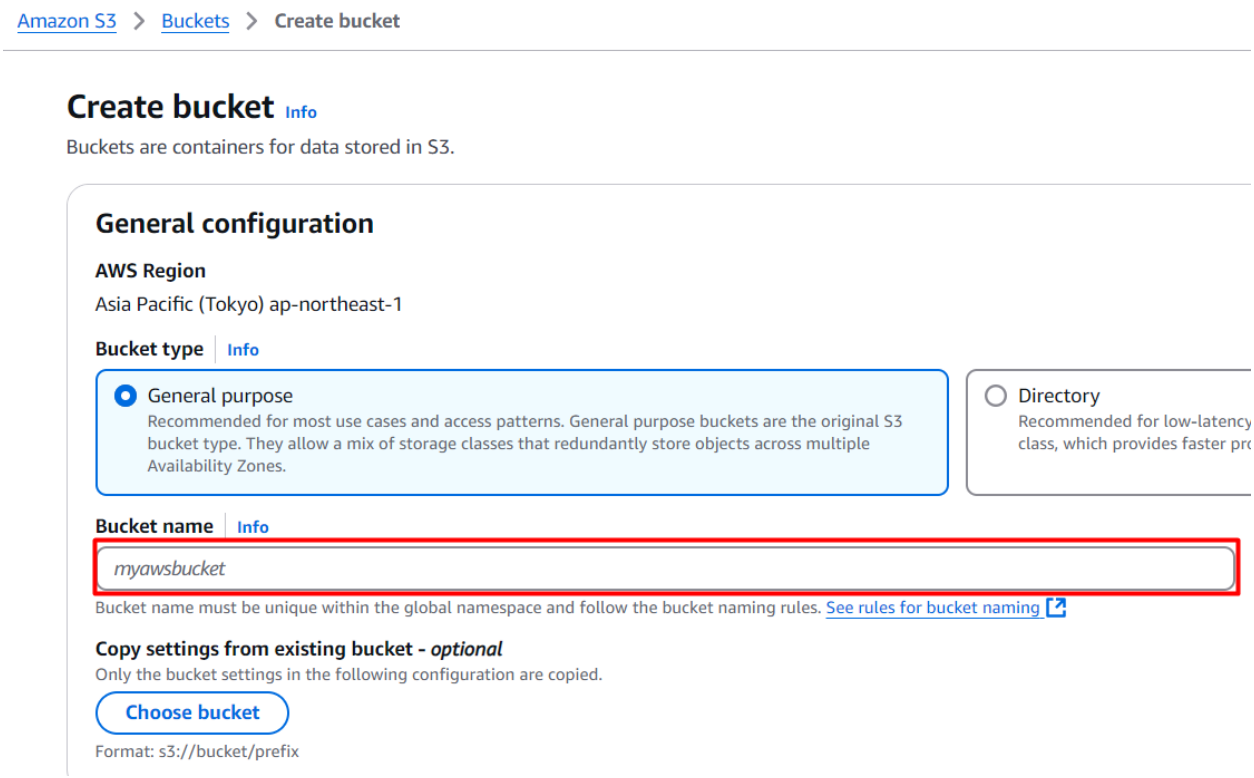

Figure 2.32 Create a Bucket Name

- Create Bucket.

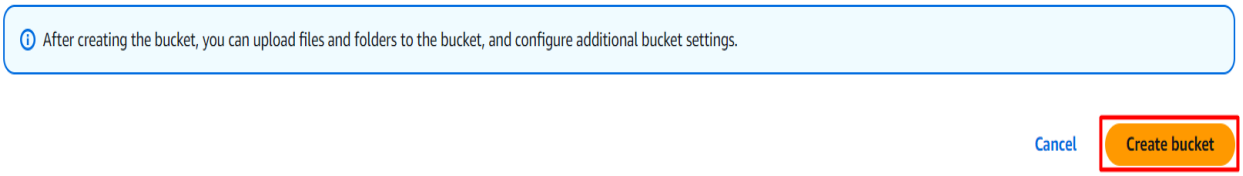

Figure 2.33 Create Bucket

(2) Upload the firmware file (.rsu) to the *Amazon S3 Bucket*

- Choose your S3 bucket, and then click **Upload**.

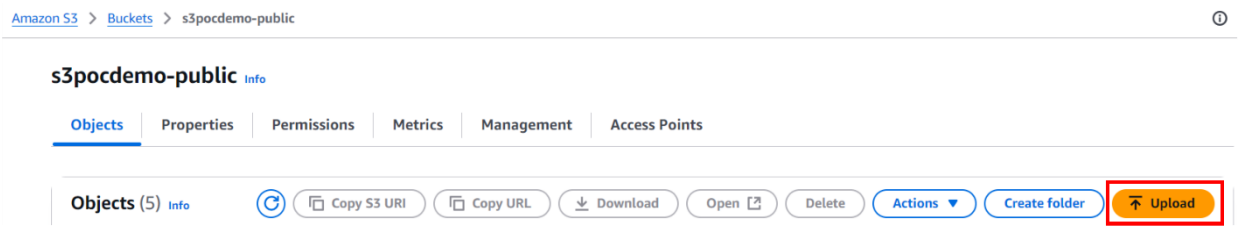

Figure 2.34 Uploading the Firmware File

- Click on **Add File**, then select the firmware you created in the previous section. Double-check to ensure it is the correct firmware before clicking **Upload**.

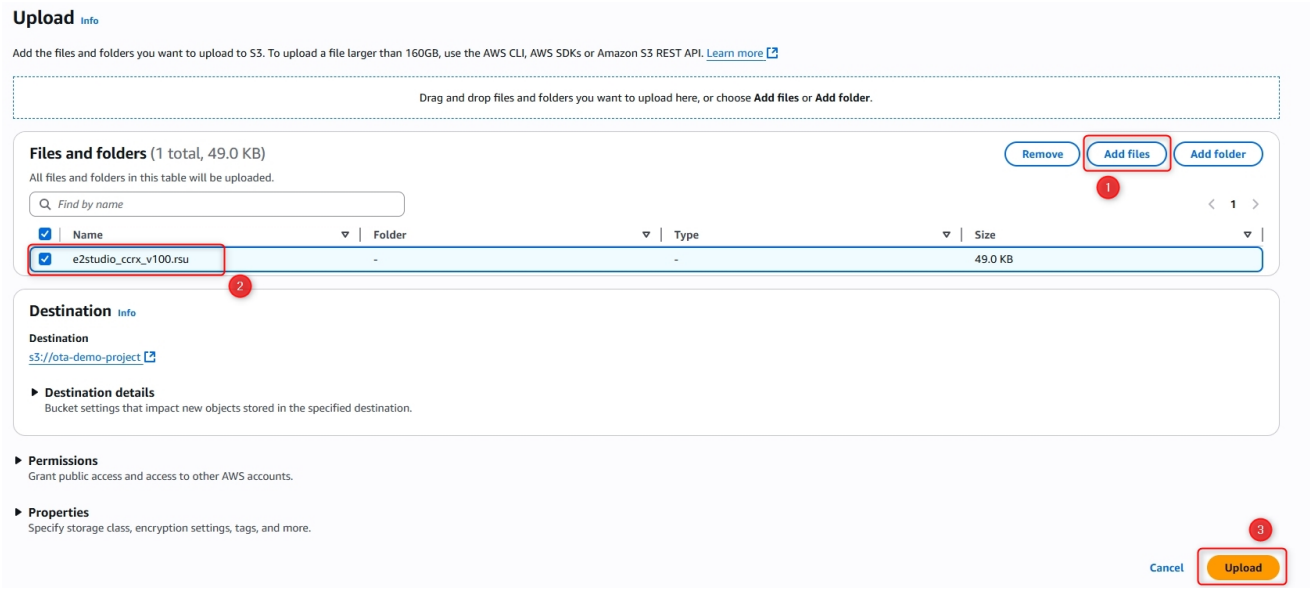

Figure 2.35 Add Firmware File

(3) Get the object *URL* of the firmware

- Create Presigned URL

Reference URL: https://dev.to/johnogbonna/aws-using-s3-and-generating-a-pre-signed-url-1m2j

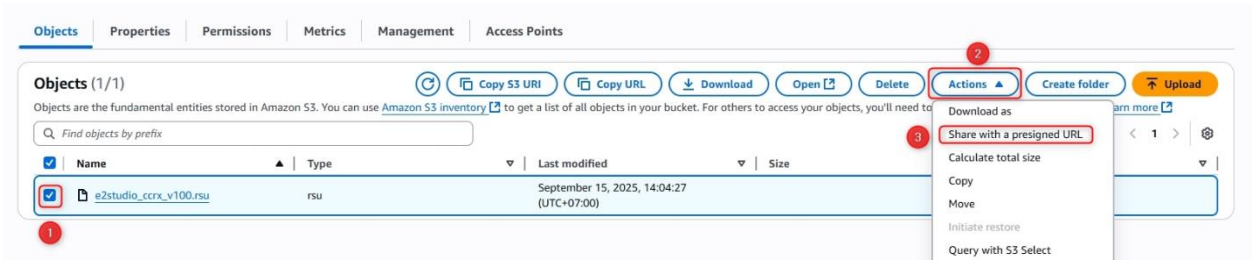

Figure 2.36 Create Presigned URL(1)

Step 1: On the **Objects** page of your S3 bucket, locate and select your firmware file

Step 2: Click on the **Actions** dropdown menu.

Step 3: Click on **Share with a presigned URL** to generate a temporary link to share the file.

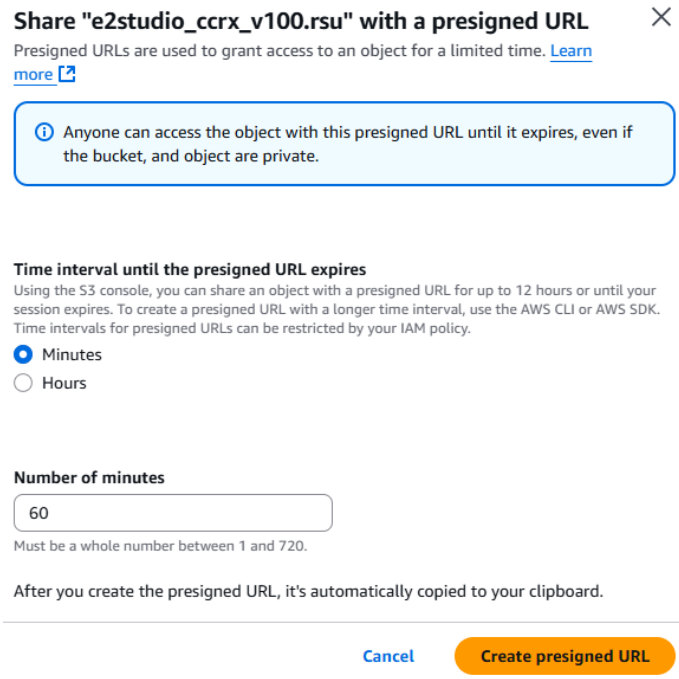

Figure 2.37 Create Presigned URL(2)

Step 4: In the dialog box, set the expiration time for the presigned URL.

- You can choose between **minutes** and **hours**.
- Example: Set to **60 minutes** (must be a whole number between 1 and 720).

Step 5: Click ***Create presigned URL*** button

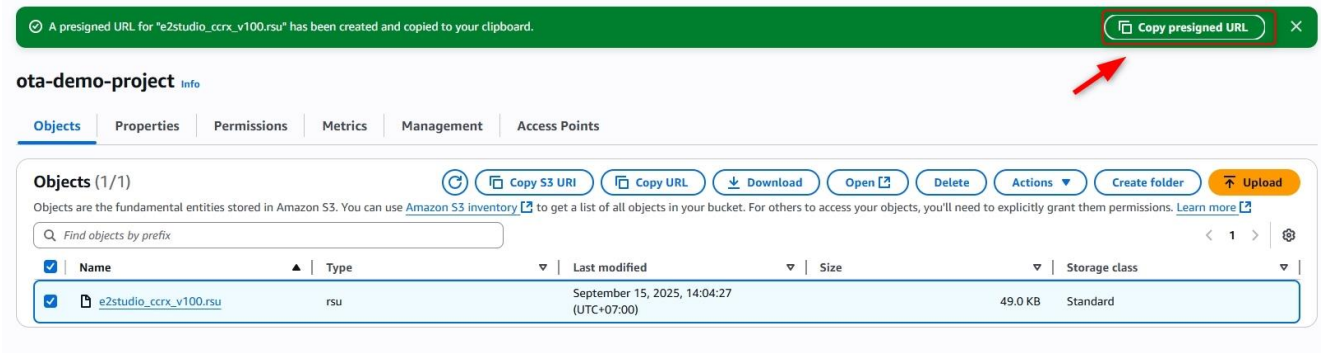

Figure 2.38 Create Presigned URL(3)

- Open API Gateway

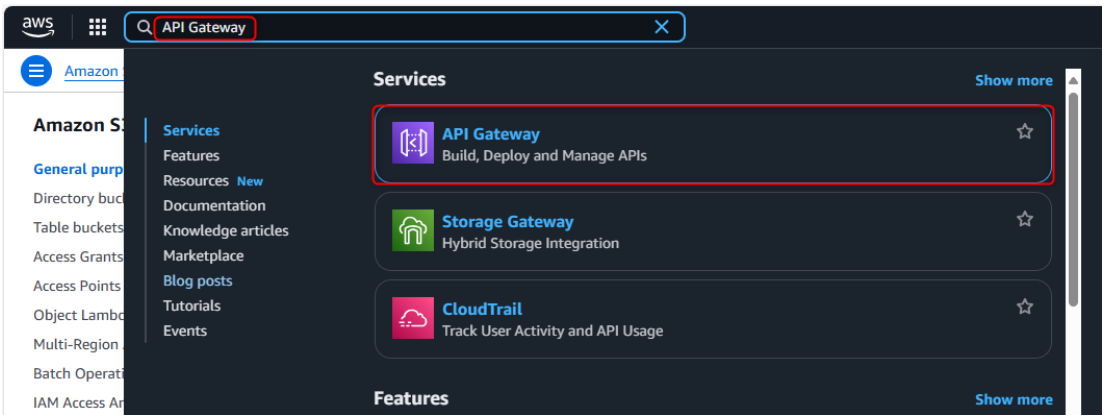

Figure 2.39 API Gateway

Create short link for pre-signed URL

- Click on *Create API* button

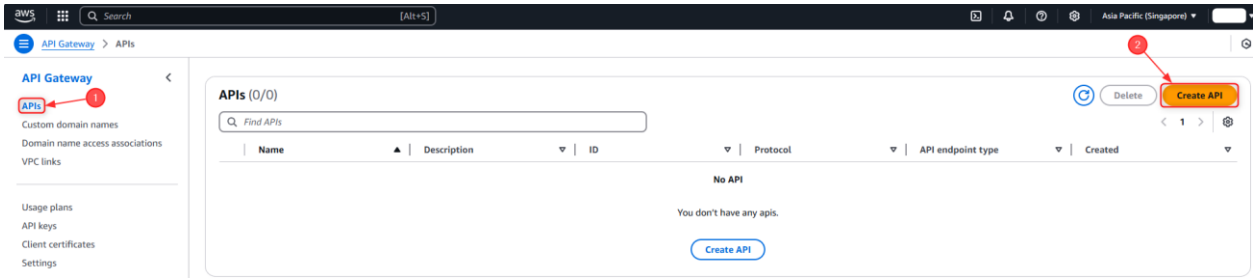

Figure 2.40 Create the API

- Create ***HTTP API***

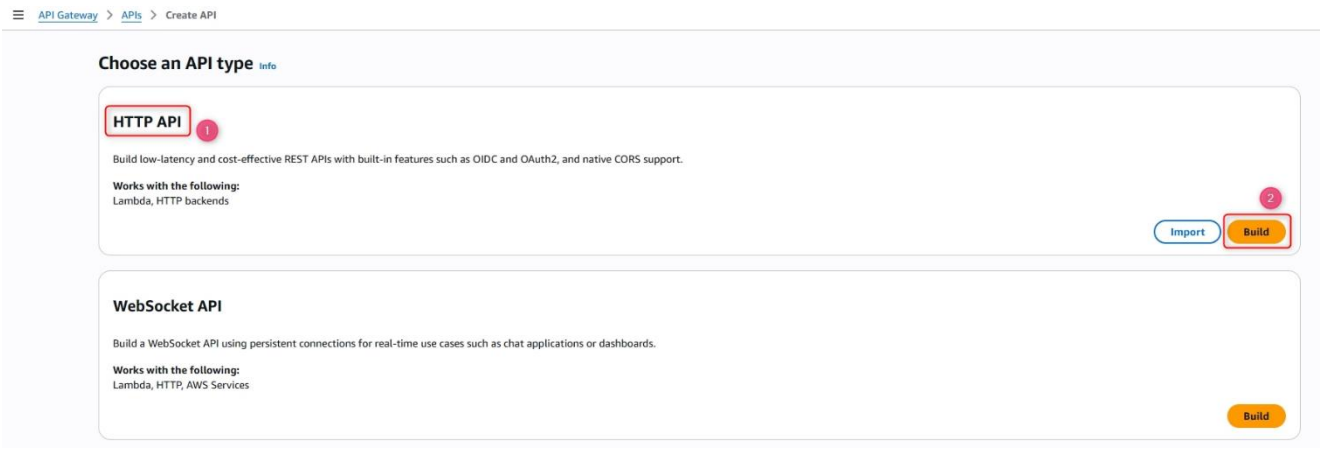

Figure 2.41 Create HTTP API

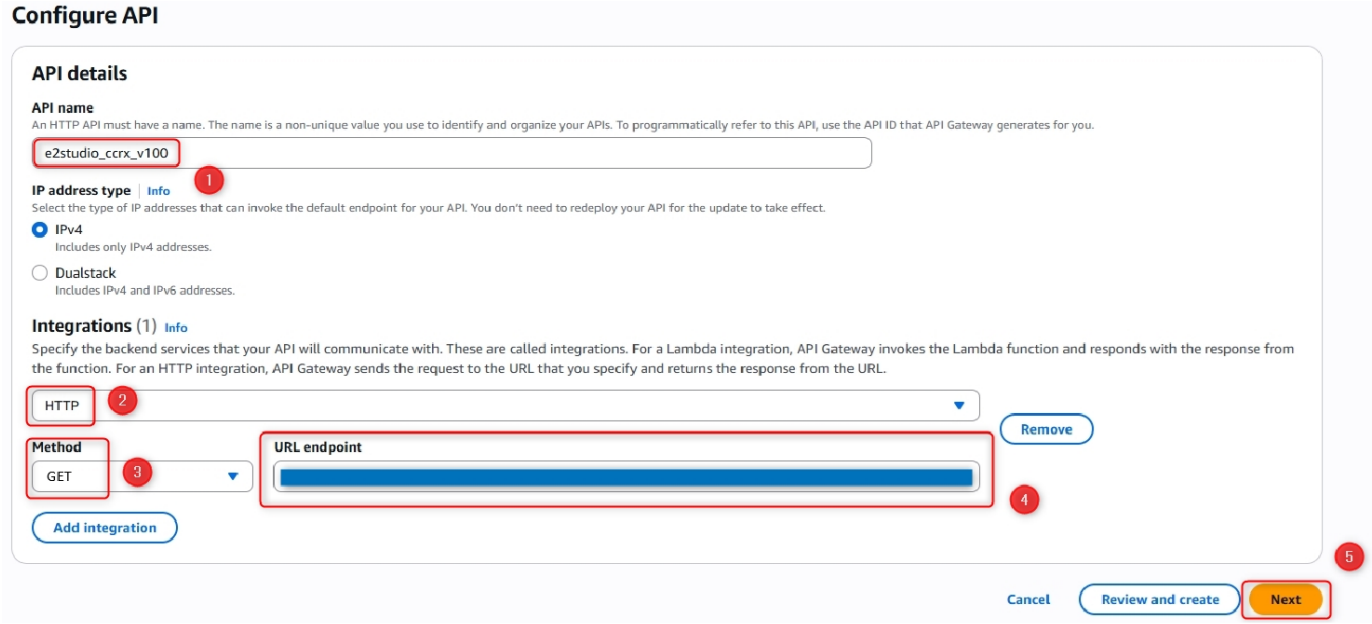

Figure 2.42 Configuration API

Step 1: Enter your *API* name.

Step 2: Select *HTTP*.

Step 3: Select the *GET* method.

Step 4: Enter your generated pre-signed URL from Create presigned URL(3) image.

Step 5: Click *Next* to finish the API configuration.

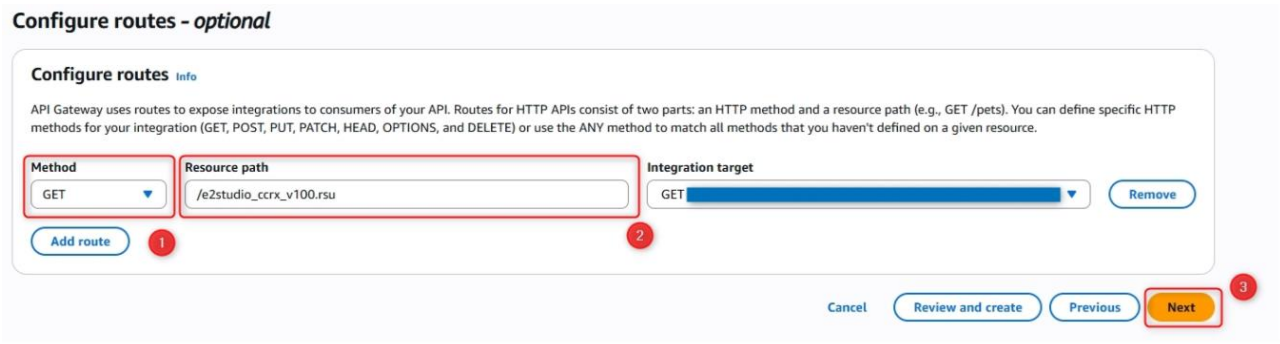

Figure 2.43 Configure Routes

Step 1: Select *GET* method.

Step 2: Enter the “Resource path”, for example: /e2studio_ccrx_v100.rsu

Step 3: Click *Next* to finish the routes configuration.

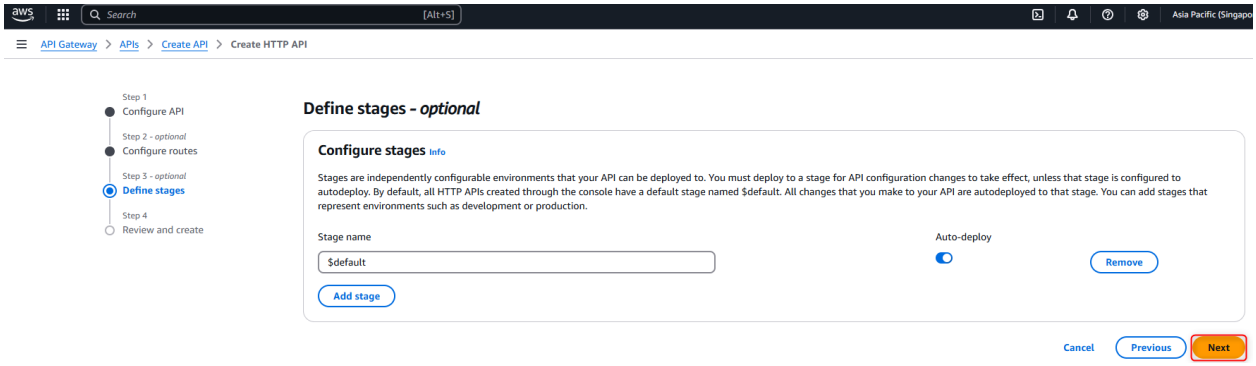

Figure 2.44 Define Stages

- Let the **“Define stages”** by default, click Next to finish the define stages configuration.

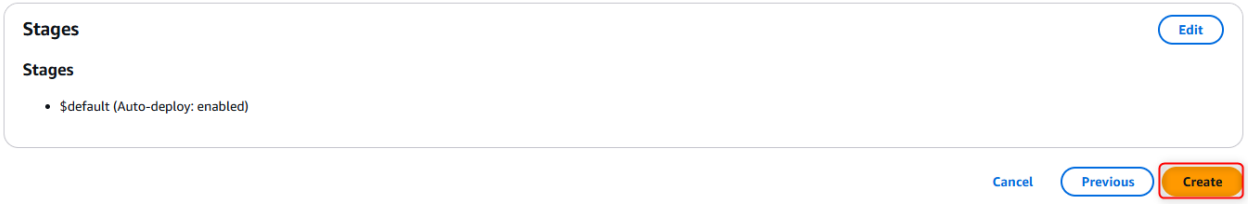

Figure 2.45 Review and Create

- Click **Create** button to finish creating the API.

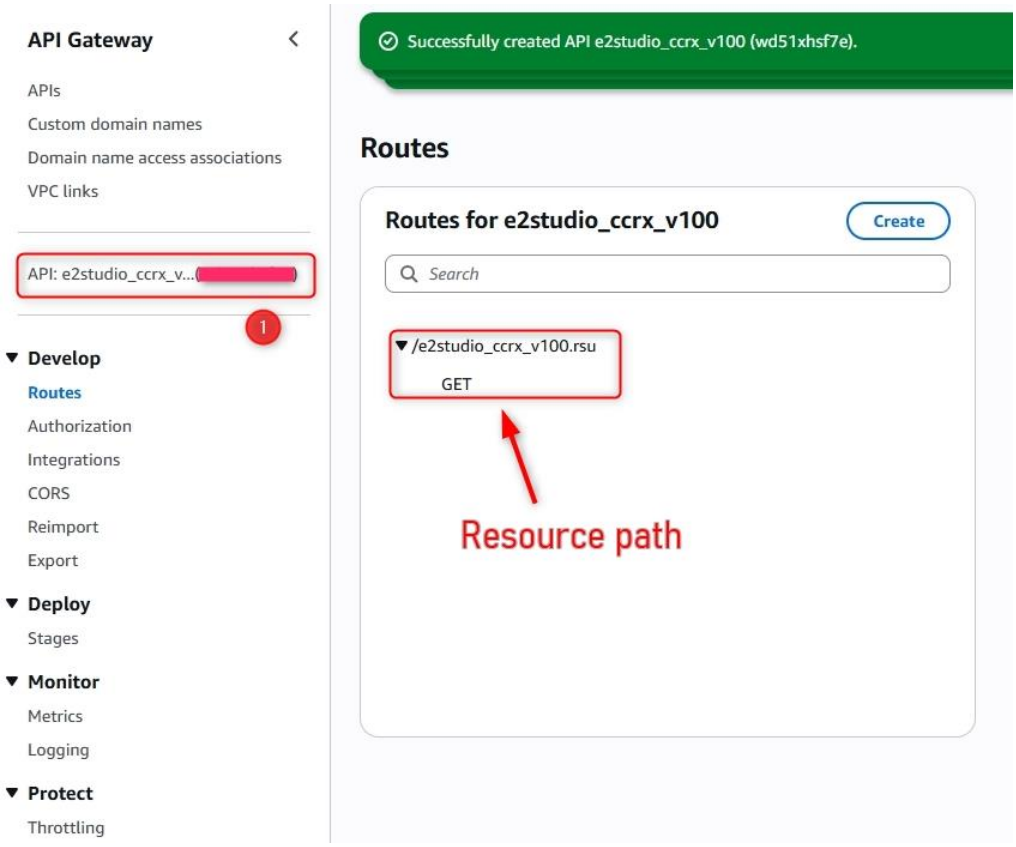

Figure 2.46 Resource Path Identification in API Gateway

- Please take note of the ***Resource path***

Figure 2.47 Invoke URL Identification

- The final URL of .rsu file has a form as below

Final URL = <Invoke URL>/<Resource path>
Example:
https://wd51xhsf7e.execute-api.ap-southeast-1.amazonaws.com/e2studio_ccrx_v100.rsu

3. Execute the Demonstration Project

3.1 Creating a New Project and Connecting to the MCU Board

- Start the Renesas Flash Programmer

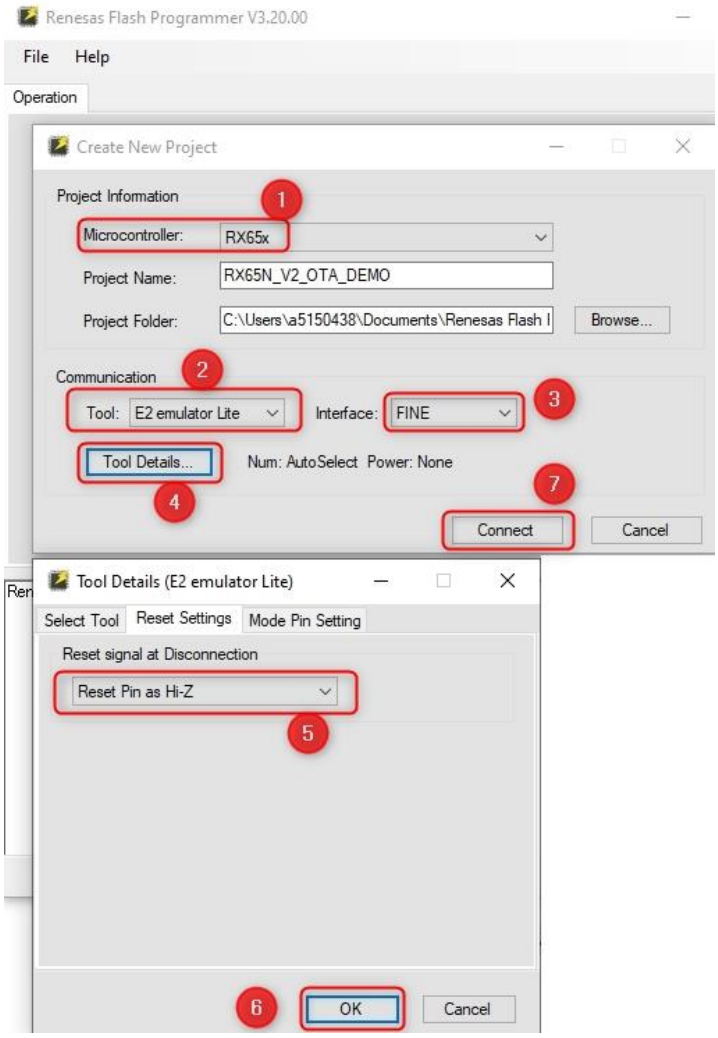

Figure 3.1 Creating a New Project and Connecting to the MCU Board

Step 1: Select appropriate microcontroller "*RX65x*"

Step 2: In the Tool menu, select "*E2 emulator Lite"*

Step 3: In the Interface menu, select "*FINE"*

Step 4: Click "*Tool Details…".*

Step 5, 6: In *Reset Settings*, choose "*Reset Pin as Hi-Z"*, then click "*OK"*

Step 7: Click *"Connect"*

- The connection is successful if the following window appears.

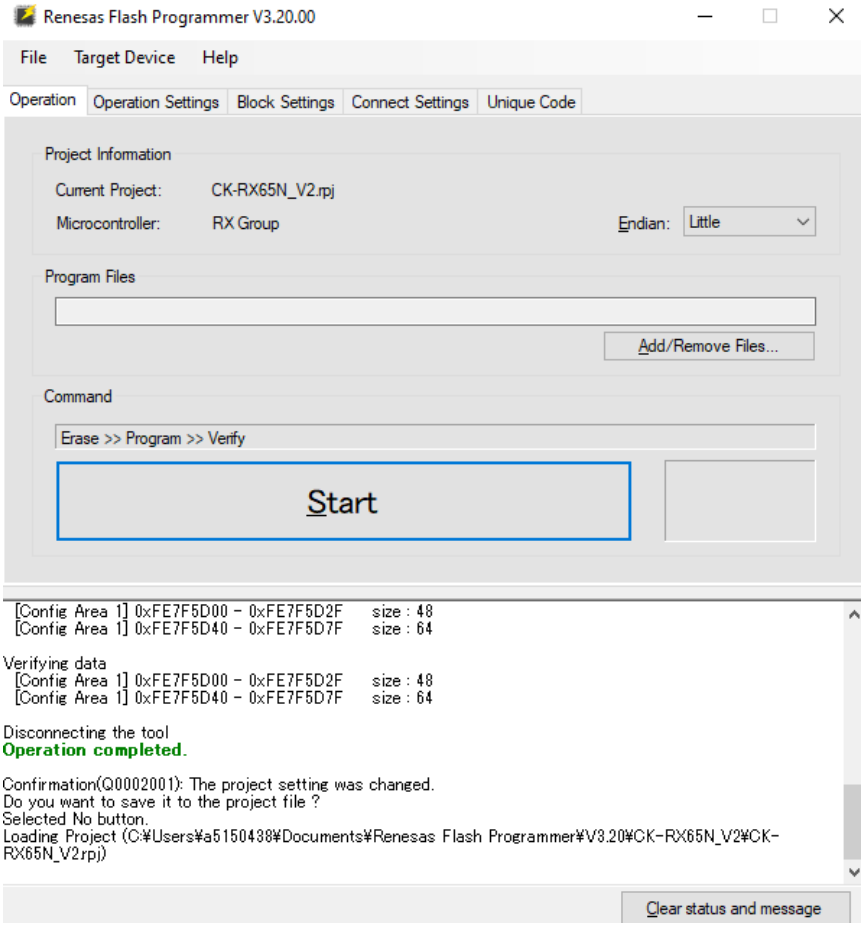

Figure 3.2 Operation completed (Connect)

3.2 Programming a MOT File to the MCU Board

- In the Program File field, enter the path to the MOT file to be programmed which is mentioned *2.2.7.4 Creating the Initial Firmware*, and then click ***"Start"***.
    - Program File: MOT file to be programmed (Example: *initial_firm_ccrx.mot*, *e2studio_ccrx.mot*)
    - Click *"Start"*

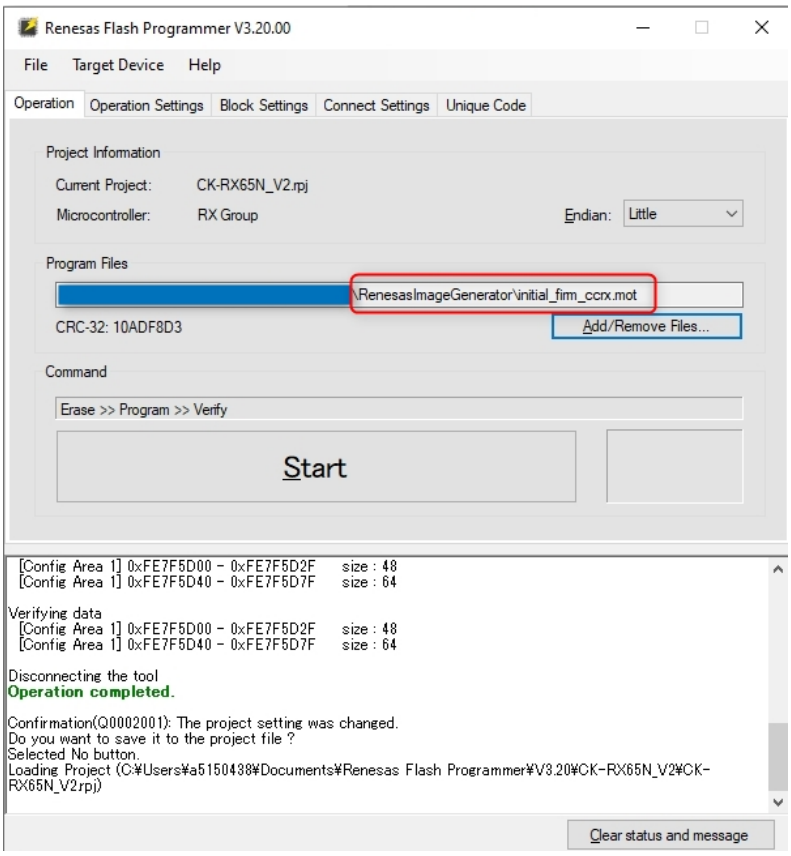

Figure 3.3 Programming a MOT File to the MCU Board

- Make sure that programming is successful.

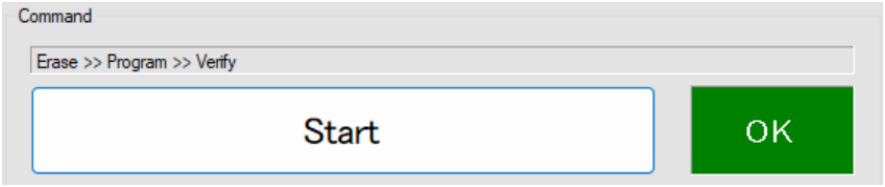

Figure 3.4 Successful programming

3.3 Request to Update the Firmware

- Waiting for the network connection to be successfully established.

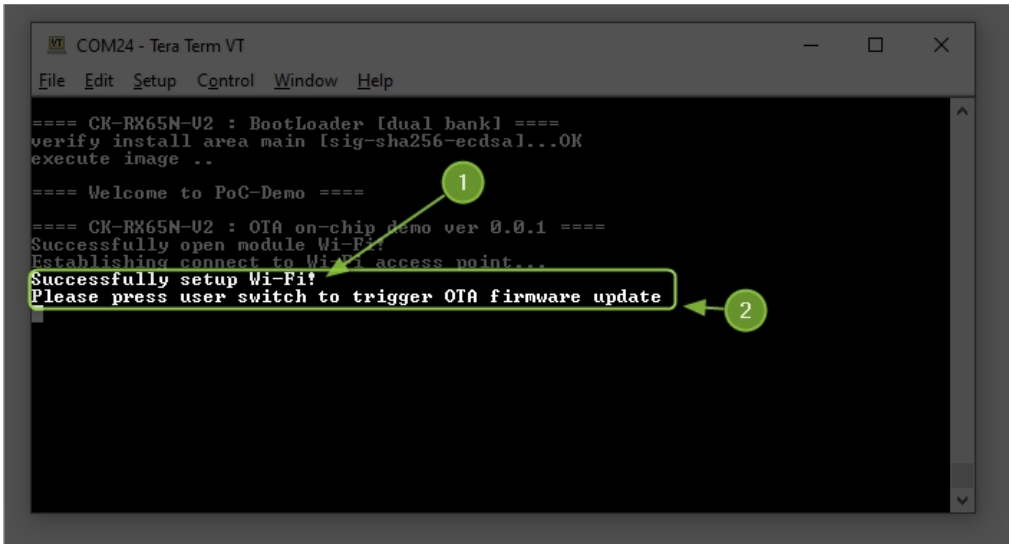

Figure 3.5 Successfully setup Wi-Fi

- Press User Switch button to trigger OTA firmware update

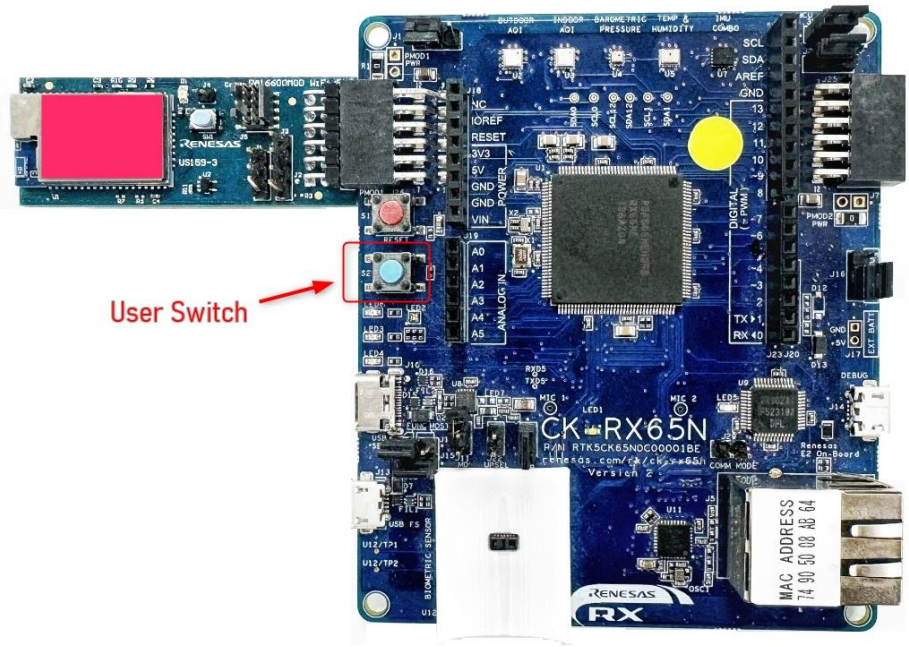

Figure 3.6 User switch button

- After the user switch was pressed, input the firmware URL which is mentioned in *Get the Object URL of the firmware*
- Click the Edit tab of the Tera Term and ***"Paste<CR>"*** and verify and confirm the valid string and press **"OK"**.

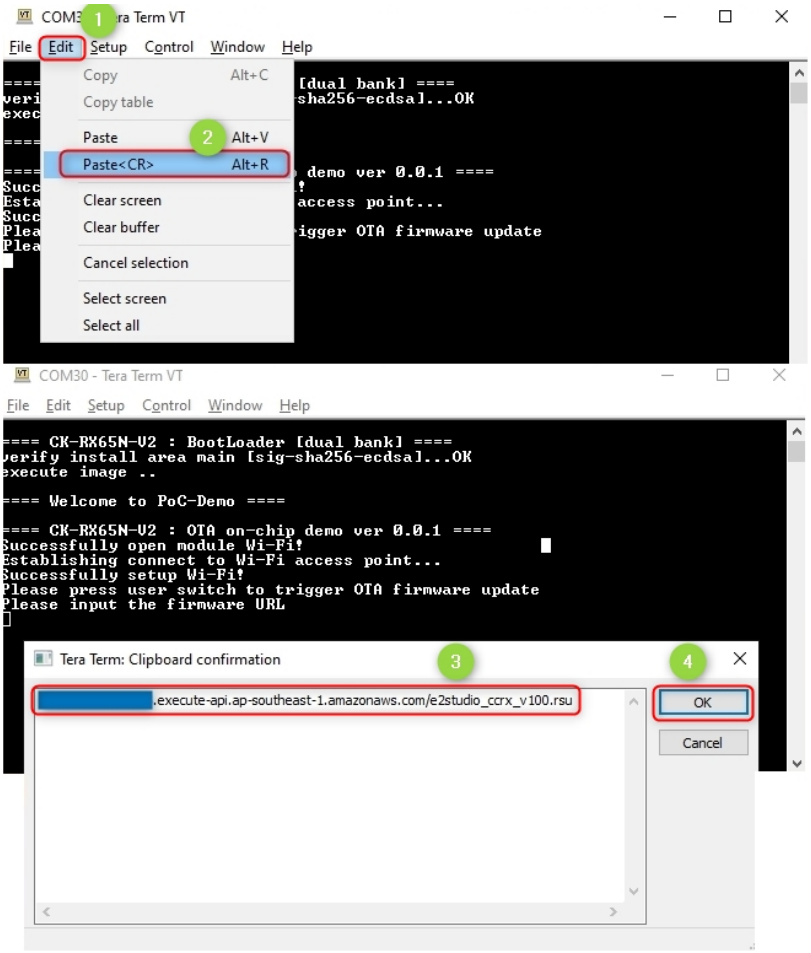

Figure 3.7 Input Firmware URL

3.4 Observe the Firmware Update Over-the-Air Process

- The firmware update process starts with progress messages such as download progress, “W 0xFFE00000, 512” (indicating that address of flash memory and block size that was written new firmware) displayed in the terminal. **Upon successful completion**, the firmware is updated from **0.0.1** to **1.0.0**. If an error occurs during the update, the MCU automatically resets, and the previous stable firmware version is retained.

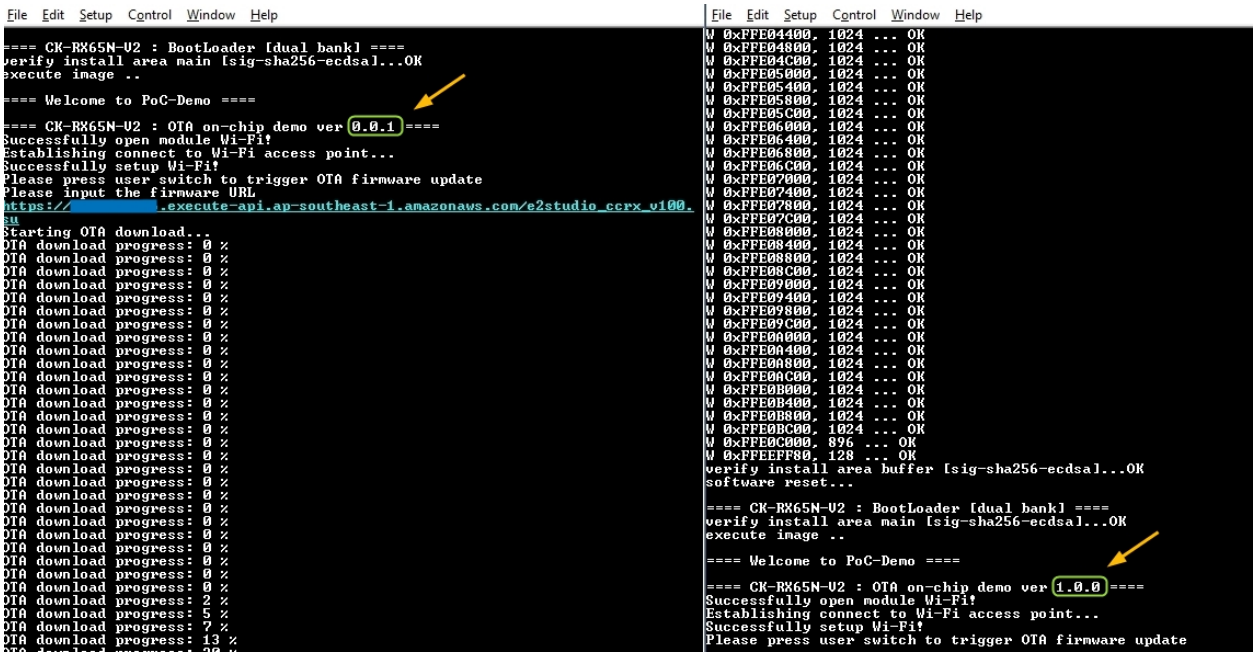

Figure 3.8 Result

- Successfully updated the version from **0.0.1** to **1.0.0**

4. Appendix

4.1 Known Issues for DA16600

4.1.1 Firmware Size Limit Causes Update Failure

The DA16200/DA16600 SDK V3.2.9.2 restricts firmware downloads from the server to under 256 KB. Attempting to download firmware exceeding this limit results in an error, preventing successful completion. This affects users needing larger firmware updates.

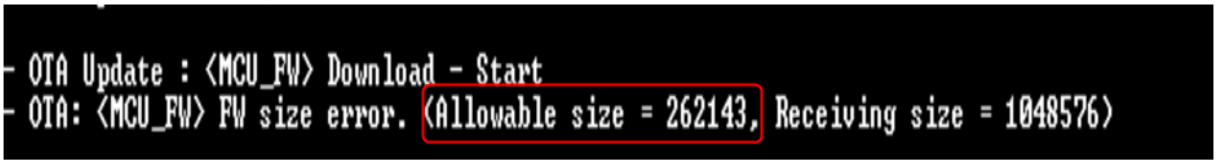

Figure 4.1 Error Log

4.1.2 Resolution

(1) Download **DA16200/DA16600 FreeRTOS SDK V3.2.9.2**

Access the DA16XXX - Ultra-Low Power Wi-Fi SoC for Battery-Powered IoT Devices | Renesas to Download the **DA16200/DA16600 FreeRTOS SDK V3.2.9.2**

(2) Importing **DA16600 FreeRTOS SDK Project** into e² studio.

Importing the **DA16600 FreeRTOS SDK Project** into e² studio is similar to importing **the DA16200 FreeRTOS SDK Project**. Therefore, please refer to UM-WI-056 DA16200 DA16600 FreeRTOS Getting Started Guide under **Section 5.4 Importing DA16200 FreeRTOS SDK Project** into e² studio for detailed instructions on the process.

(3) Modify the project.

- In e² studio Project Explorer, open the file **config_generic_sdk.h** in the folder **da16600\get_started\include\user_main** and modify the macro highlighted as below.

```
#if defined ( __SUPPORT_OTA__ )
#define __OTA_UPDATE_MCU_FW__
#endif //__SUPPORT_OTA__
```

- In e² studio Project Explorer, open the file **da16200_map.h** in the folder **da16600\core\bsp\driver\include\DA16200** and modify the two macros highlighted as below.

```
/* DA14531 BLE Firmware Download start */
#define SFLASH_BLE_FW_BASE                (SFLASH_14531_BLE_AREA_START)
/* DA14531 BLE Security DB Area start */
#define SFLASH_USER_AREA_BLE_SECURITY_DB (SFLASH_BLE_FW_BASE +
__BLE_IMG_SIZE__)
/* SFLASH User Area */
#define SFLASH_USER_AREA_1_START        0x00600000
#define SFLASH_USER_AREA_1_END          0x00800000
```

(4) Building project.

Please refer to https://www.renesas.com/document/qsg/um-wi-056-da16200-da16600-freertos-getting-started-guide (DA16200/DA16600 FreeRTOS Getting Started Guide) under ***Section 5.5 Building Projects*** for detailed instructions on the process.

(5) Flash the new firmware.

Flash the new firmware via Tera Term by following the instructions provided in the UM-WI-056 DA16200 DA16600 FreeRTOS Getting Started Guide under ***Section 4.5.2 Using Macro Script of Tera Term.***

4.2 Debugging

printf() in the project is used to provide additional error information during the debugging process.

Revision History

Rev.	Date	Revision History	
		Page	Summary
1.00	Apr. 22, 2025	-	First edition issued
1.10	Oct. 21, 2025	-	Edit format
		4 - 7	Revise the entire Section 1. Overview
		8	Modified Section 2.1 Hardware Setup
		13	Modifed image 2.2.1.4 Installing Renesas Flash Programmer
		20	Added new folder structure for Section 2.2.5 Project Description
		22	Modified image Section 2.2.6 Importing The Demo Project
		23 - 25	Revise the entire section 2.2.7 Create Firmware Initialization
		26 - 36	Added new Section 2.2.8 and modified content
		37 - 42	Revise the entire Section 3 - Add 4 new subsections

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/.