

RX ファミリ

RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

はじめに

本アプリケーションノートでは、FreeRTOS with IoT Libraries 上で OTA デモアプリケーションを使用する手順を説明します。セキュリティに関する詳細は「ルネサス MCU におけるファームウェアアップデートの設計方針([R01AN5548](#))」を参考してください。

【注】 本アプリケーションノートは FreeRTOS-v202210.01-LTS-rx-1.1.0 以降の FreeRTOS に対応した手順となります。FreeRTOS-v202107.00 以前の FreeRTOS を使用する場合は「RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法([R01AN5549](#))」を参照してください。

動作確認デバイス

RX65N、RX651 グループ

ハードウェア

CK-RX65N

参考ドキュメント

ルネサス MCU におけるファームウェアアップデートの設計方針 ([R01AN5548](#))

目次

1. 概要	3
1.1 システム概要	3
1.2 動作確認環境(ハードウェア).....	4
1.3 動作確認環境(ソフトウェア).....	4
2. 事前準備	5
2.1 TeraTeram のインストール.....	5
2.2 Python のインストール	6
2.3 OpenSSL のインストール.....	7
2.4 Renesas Image Generator のインストール.....	8
2.5 CK-RX65N の接続	10
3. AWS の設定.....	11
3.1 AWS コンソールへのサインイン	12
3.2 AWS のリージョン設定.....	14
3.3 デバイスを AWS に登録する	14
3.3.1 ポリシーの設定	14
3.3.2 デバイス（モノ）を AWS IoT に登録.....	16
3.3.3 エンドポイントの確認	20
3.4 Amazon S3 バケットの作成.....	21
3.5 IAM ユーザーに OTA の実行権限を割り当てる	24
4. デバイスの設定	33
4.1 鍵ペアと証明書の生成.....	33
4.2 初期バージョンのファームウェア構築	36
4.2.1 プロジェクトのインポート	36
4.2.2 プロジェクト環境設定の確認.....	38
4.2.3 プロジェクト設定	40
4.2.4 初期ファームウェアの作成	46
4.2.5 AWS IoT 情報の登録	50
5. ファームウェアの更新	57
5.1 更新用ファームウェア構築	57
5.1.1 バージョンの変更	57
5.2 ファームウェアの更新.....	58
6. トラブルシューティング	66

1. 概要

1.1 システム概要

本項では、CK-RX65N 搭載のデュアルバンク機能対応マイコン RX65N を使用し、OTA を実現する場合の動作概要を示します。

デュアルバンク機能対応マイコンでは、ROM を Execution area（実行領域）と Temporary area(バッファ領域)に分割することが可能です。Execution area と Temporary area は動的に切り替えることができ、Execution area で既存バージョンのソフトウェアを動作させながら更新ファームウェアを ROM に書き込むことが可能です。

以下に OTA 実行時のメモリ配置および、デュアルバンク機能を使用したバンクスワップによるメモリ切り替えの動作を示します。

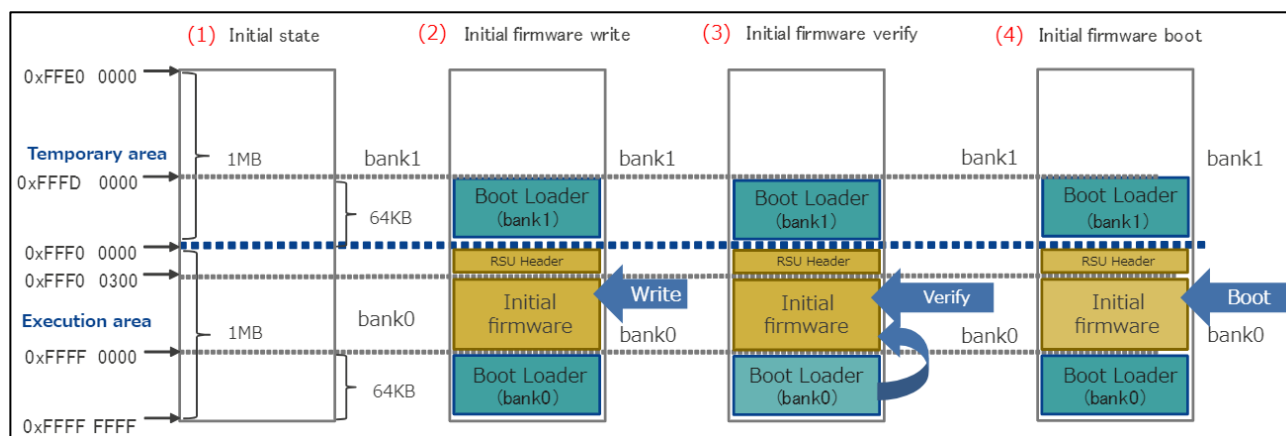


図 1-1 OTA の動作概要(1)

- (1) Renesas Flash Programmer にてイレーズを実行した状態（ブランク状態）
- (2) Renesas Flash Programmer にてブートローダと初期ファームウェアを結合したデータ※を書き込んだ状態 ※Boot Loader(bank0) + Initial firmware + RSU Header + BootLoader(bank1)が結合したデータを指します。RSU Headerの詳細についてはRX ファミリ ファームウェアアップデートモジュール Firmware Integration Technology ([R01AN6850](#))の4.2 イメージファイルをご確認ください。
- (3) リセット解除後、ブートローダ(bank0)がファームウェアの検証を行う
- (4) 初期ファームウェアを起動する

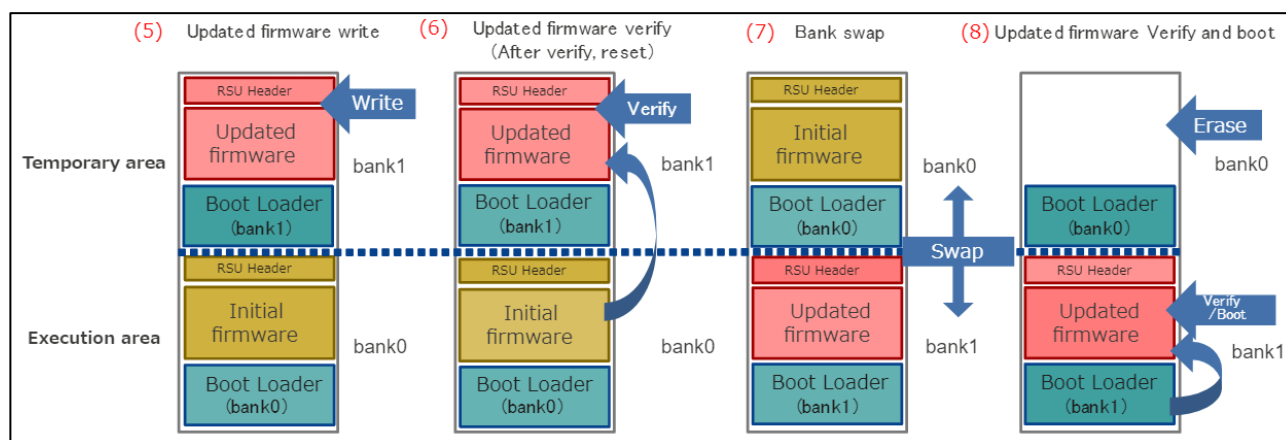


図 1-2 OTA の動作概要(2)

- (5) AWS から更新されたファームウェアを受信すると bank1 に書き込みを行う
bank1 に書き込み中は BGO 機能により初期ファームウェアの動作が実行される

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

- (6) 初期ファームウェアによって更新ファームウェアの検証を行う
- (7) bank0 と bank1 を入れ替え（バンクスワップ）、Execution area に bank1 を配置する
- (8) 更新ファームウェアをブートローダ(bank1)が検証する
bank0 の初期ファームウェアをイレーズし、bank1 に書き込んだ更新ファームウェアを実行する

1.2 動作確認環境(ハードウェア)

本デモプロジェクトの動作確認環境(ハードウェア)を以下に示します。

表 1-1 動作確認環境(ハードウェア)

項目	
使用ボード	CK-RX65N (Cellular / Ethernet) (注 1)
Cellular モジュール	RYZ014A (CK-RX65N 同梱)
SIM	LTE Cat-M1 対応 SIM (microSIM) (注 2)

- 【注】 1. 本サンプルでは Cellular 通信を使用しています。
2. CK-RX65N 付属の SIM カードを使用する場合は、以下アプリケーションノートの「4.1.5 Activating SIM card」を参照し、SIM カードのアクティベーションを行ってください。
[SIM activation, Creating the trial account and using Dashboard with RYZ014A or Ethernet Application for AWS - Getting Started Guide \(R01QS0064\)](#)

1.3 動作確認環境(ソフトウェア)

本デモプロジェクトの動作確認環境(ソフトウェア)を以下に示します。

表 1-2 動作確認環境(ソフトウェア)

項目	
統合開発環境	e ² studio 2023-10
コンパイラ	Renesas CC-RX v3.05.00
FreeRTOS	v202210.01-LTS-rx-1.1.0
ログモニタツール	TeraTerm v4.106
Python	Python 3.11.0
鍵生成ツール	Win64 OpenSSL v3.0.12
フラッシュ書き込みツール	Renesas Flash Programmer V3.12.00
Renesas Image Generator	Version3.02 (Firmware Update module Rev.2.01 同梱)

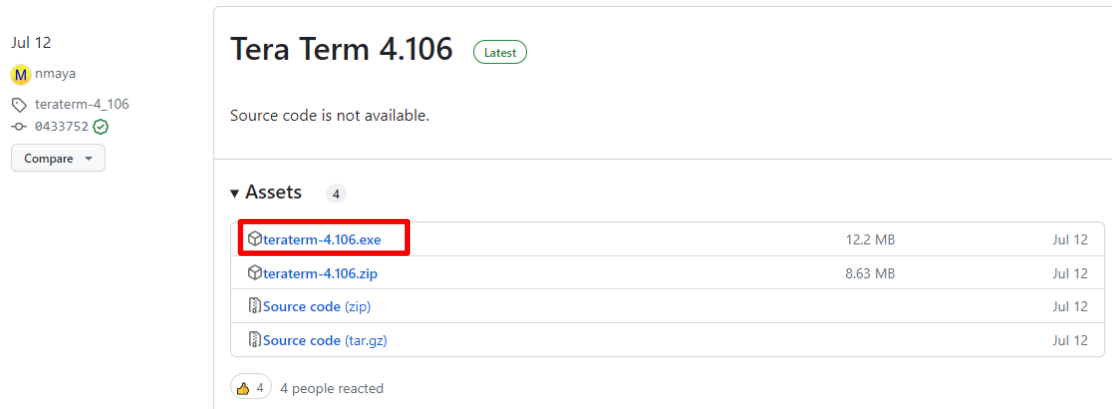
2. 事前準備

2.1 TeraTeram のインストール

- (1) Tera Term のダウンロードサイトにアクセスする

[TeraTerm ダウンロードサイト\(GitHub\)](#)

- (2) Tera Term のインストーラーをダウンロードする



- (3) インストーラーを実行し、案内に沿って Tera Term をインストールする

- (4) スタートメニューから Tera Term のアイコンをクリックして、Tera Term が起動することを確認する

2.2 Python のインストール

(1) Python のダウンロードサイトにアクセスする

[Python ダウンロードサイト](#)

(2) Python 3.11.0 のインストーラーをダウンロードする

Python 3.11.0 の Download を選択

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

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

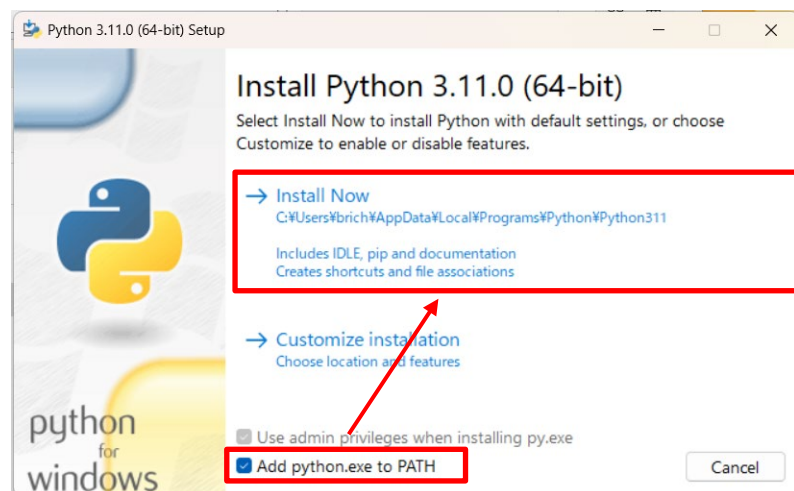
使用する OS に合わせたインストーラーをダウンロードしてください。

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
Gzipped source tarball	Source release		c5f77f1ea256dc5bdb0897eeb4d35bb0	26333656	SIG	CRT SIG
XZ compressed source tarball	Source release		fe92acfa0db9b9f5044958edb451d463	19819768	SIG	CRT SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	98fa94815780c9330fc2154559365834	42602603	SIG	CRT SIG
Windows embeddable package (32-bit)	Windows		0888959642cc8af087d88da3866490a5	9560053	SIG	CRT SIG
Windows embeddable package (64-bit)	Windows		7df0f4244e5a66760b7caaed58e86c93	10545380	SIG	CRT SIG
Windows embeddable package (ARM64)	Windows		e3dbbd5d63c6cb203adc6c0c8ca5f5f7	9765886	SIG	CRT SIG
Windows installer (32-bit)	Windows		e369a267acaad62487223bd835279bb9	23987136	SIG	CRT SIG
Windows installer (64-bit)	Windows	Recommended	4fe11b2b0bb0c744cf74aff537f7cd7f	25157416	SIG	CRT SIG
Windows installer (ARM64)	Windows	Experimental	18e5bd9a4854109adf3b77c7c9dc1ded	24289144	SIG	CRT SIG

(3) インストーラーを実行し、案内に沿って Python をインストールする

インストール画面で「Add python.exe to PATH」にチェックを入れてください。



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

- (4) コマンドプロンプトを起動し、Python 3.11.0 がインストールされていることを確認する

以下のコマンドを実行して、情報が表示されることを確認してください。

python -V

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

- (5) Python に暗号化ライブラリ「pycryptodome」をインストールする

以下のコマンドを実行して、暗号化ライブラリをインストールしてください。

pip install pycryptodome

```
C:\Users>pip install pycryptodome
Requirement already satisfied: pycryptodome in c:\users\%<...>\appdata\local\programs\python\python311\lib\site-packages (3.18.0)

[notice] A new release of pip is available: 23.1.2 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

2.3 OpenSSL のインストール

- (1) Win32/Win64 OpenSSL のダウンロードサイトにアクセスする

[Win32/Win64 OpenSSL Installer for Windows - Shining Light Productions \(slproweb.com\)\)](https://slproweb.com/products/Win32OpenSSL.html)

- (2) OpenSSL のインストーラーをダウンロードする

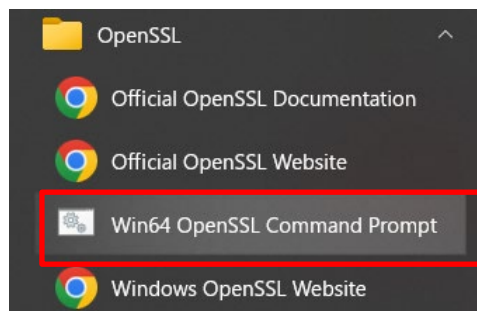
使用する OS に合わせたインストーラーをダウンロードしてください。

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

- (3) インストーラーを実行し、案内に沿って OpenSSL をインストールする

OpenSSL の DLL の保存先には【The OpenSSL binaries directory】を選択してください。

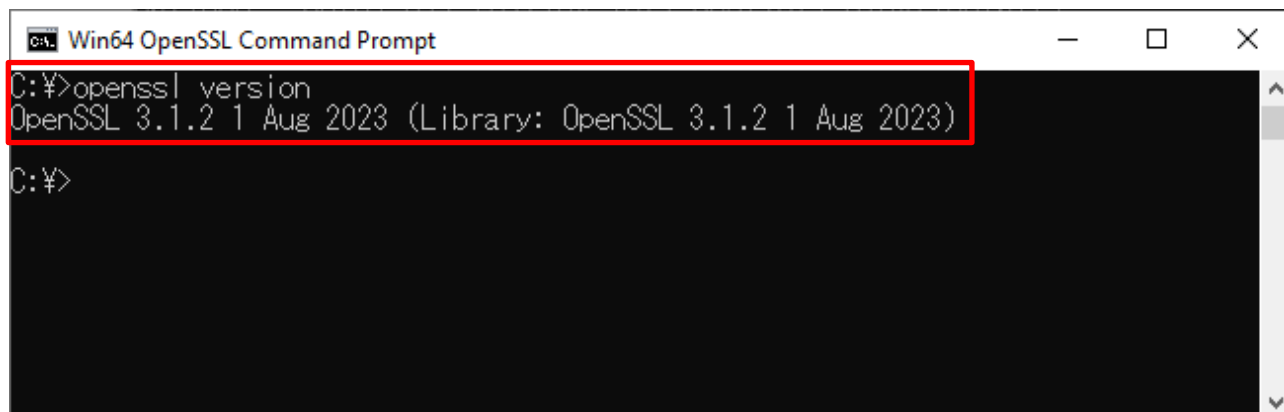
- (4) スタートメニューから Win64 OpenSSL Command Prompt を実行する



(5) コマンドプロンプトで openssl コマンドが実行できることを確認する

以下のコマンドを実行して、バージョン情報が表示されることを確認してください。

openssl version



```
Win64 OpenSSL Command Prompt
C:\>openssl version
OpenSSL 3.1.2 1 Aug 2023 (Library: OpenSSL 3.1.2 1 Aug 2023)
C:\>
```

2.4 Renesas Image Generator のインストール

Renesas Image Generator は、ファームウェアアップデートモジュールで使用するファームウェアイメージを生成するユーティリティツールです。Renesas Image Generator はファームウェアアップデートモジュールが使用する以下のイメージを生成することができます。

- 初期イメージ：ブートローダとアプリケーションプログラムで構成されるシステムの初期設定時にフラッシュライタで書き込むイメージファイル(拡張子 mot)
- 更新イメージ：ファームウェアアップデート対象のイメージファイル(拡張子 rsu)

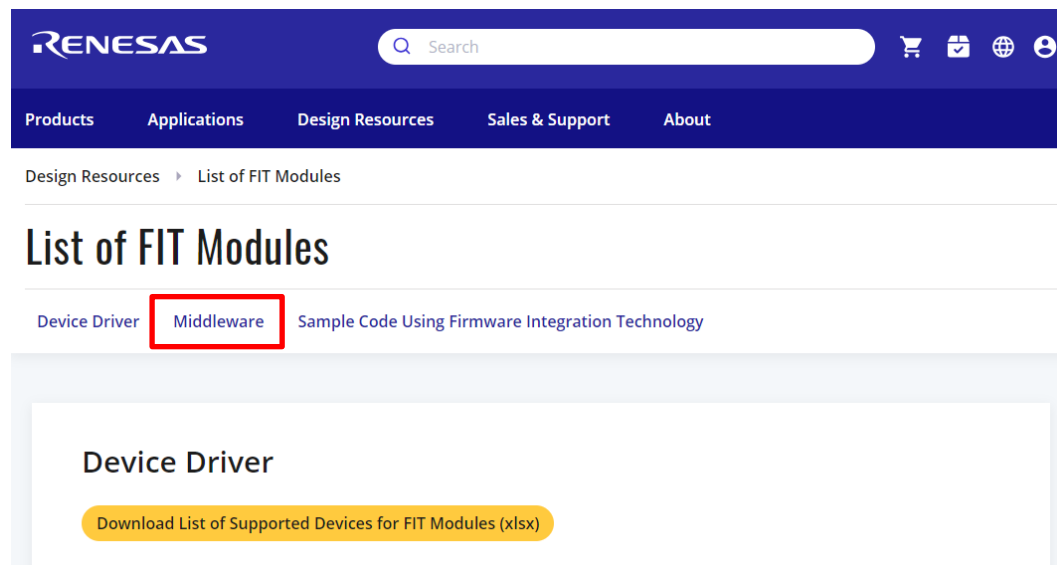
Renesas Image Generator は FIT モジュールの Firmware Update module に同梱されています。

【注】 Firmware Update module Rev.2.00 以降のバージョンでは、Python スクリプトを使用したファームウェア生成にのみ対応しております。

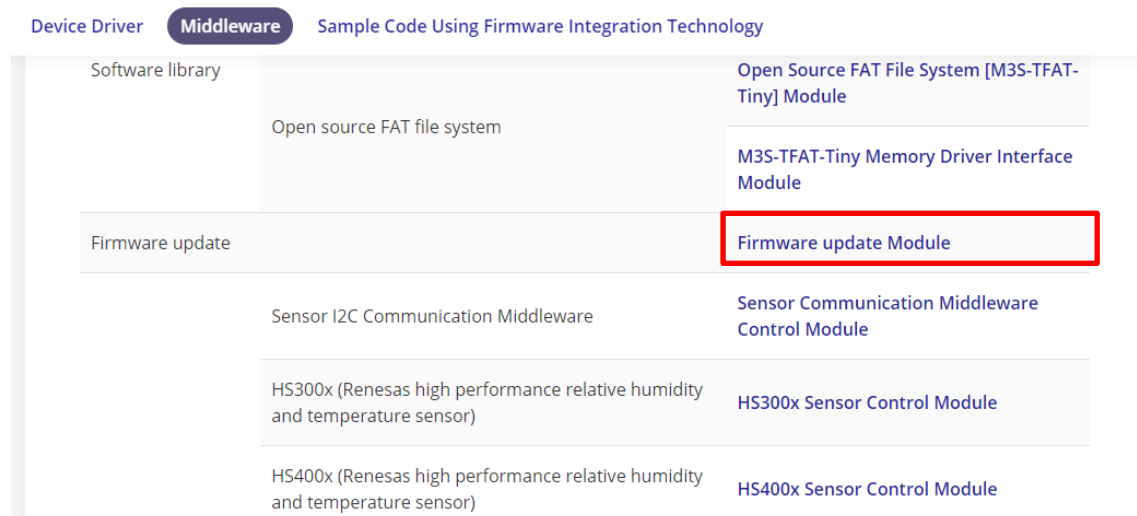
RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(1) FIT モジュールリストのサイトにアクセスしてミドルウェアを選択

[FIT モジュールリストサイト](#)

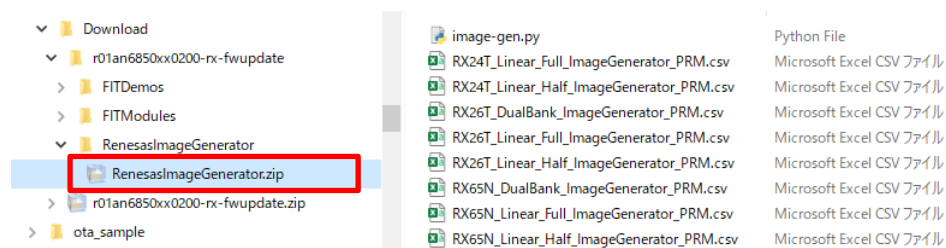


(2) ファームウェアアップデート モジュールをダウンロードする



(3) ダウンロードしたファームアップデートモジュールを解凍する

ファームアップデートモジュール内の RenesasImageGenerator.zip を解凍してください。
RenesasImageGenerator フォルダには Renesas Image Generator スクリプトファイル(image-gen.py)と各デバイス用のパラメータファイル (*_ImageGenerator_PRM.csv) が含まれています。



2.5 CK-RX65N の接続

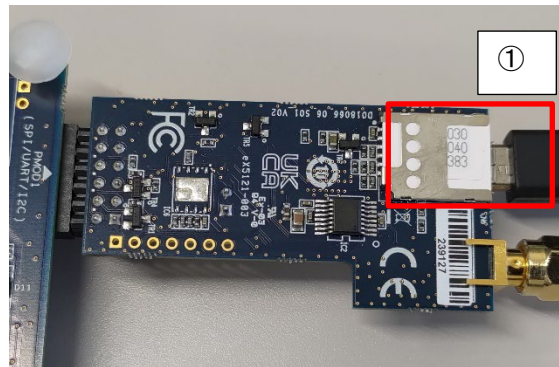


図 2-1 RYZ014A PMOD 裏面

- ① RYZ014A PMOD の CN6 に SIM カードを挿入してください

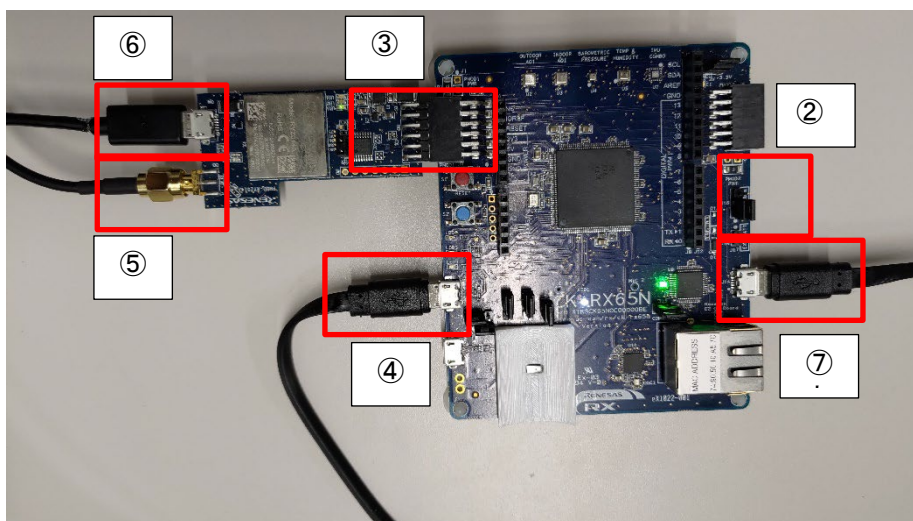


図 2-2 ベースボードおよび RYZ014A PMOD 表面

- ② ベースボードの J16 の 1-2 をショートする（デバッグ許可）
- ③ ベースボードの PMOD1 に RYZ014A PMOD を接続する
- ④ ベースボードの J20 と PC を USB ケーブルで接続する（USB シリアル接続）
- ⑤ RYZ014A PMOD の CN3 にアンテナを接続する
- ⑥ RYZ014A PMOD の CN4 に USB ケーブルを接続し、電源供給する
- ⑦ ベースボードの J14 と PC を USB ケーブルで接続する（デバッグ接続）

【注】 予備の USB ケーブルをお持ちの場合は、手順⑥を実施してください。
RYZ014A PMOD への電源供給を行わない場合、通信が不安定になる場合があります。

3. AWS の設定

FreeRTOS デモを実行するには、AWS アカウント（ルートユーザー、または AWS IoT と FreeRTOS クラウドサービスにアクセスできる権限を持つ IAM ユーザー）が必要です。

AWS のアカウントと権限の設定方法は、<https://docs.aws.amazon.com/freertos/latest/userguide/freertos-prereqs.html>を参照してください。

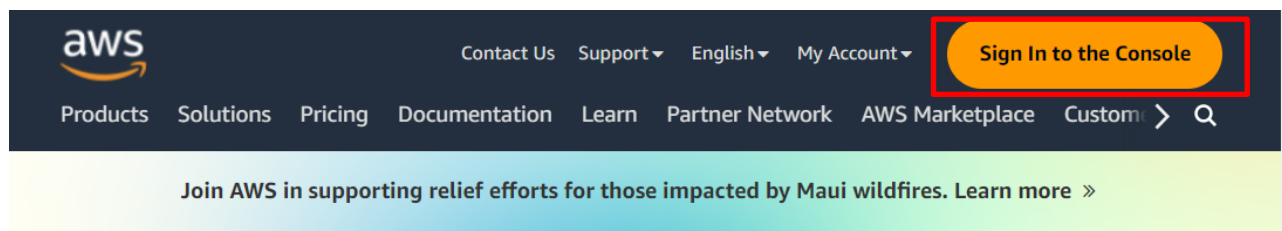
OTA 更新の設定については、<https://docs.aws.amazon.com/freertos/latest/userguide/ota-prereqs.html> を参照してください。

次に、<https://docs.aws.amazon.com/freertos/latest/userguide/freertos-prereqs.html> の説明に従って、AWS IoT にボードを登録する必要があります。

デモが AWS と通信できるようにするには、2 章の説明に従ってソースコードを設定する必要があります。

3.1 AWS コンソールへのサインイン

(1) AWS (<https://aws.amazon.com/>) にアクセスし、AWS コンソールにサインインをクリック



(2) E メールアドレス/アカウント ID を入力して 次へ をクリック

サインインするユーザーがルートユーザーの場合はE メールアドレス、IAM ユーザーの場合はアカウント ID を入力します。（過去サインインしていた場合、本手順はスキップされる場合があります）



Sign in

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

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

Root user email address

(3) パスワードを入力して、サインインをクリック

ルートユーザーの場合



Root user sign in ⓘ

Email:

Password

[Forgot password?](#)

.....

Sign in

[Sign in to a different account](#)

[Create a new AWS account](#)

IAM ユーザーの場合



Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

☐ Remember this account

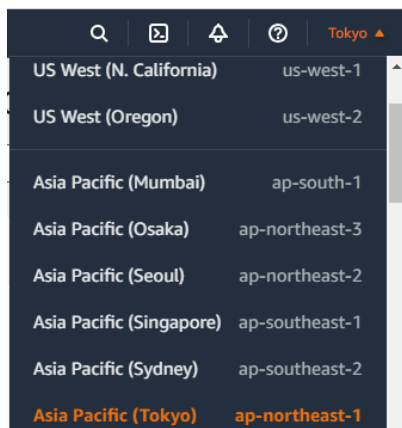
Sign in

[Sign in using root user email](#)

[Forgot password?](#)

3.2 AWS のリージョン設定

AWS にログイン後、画面右上にあるリージョンを設定してください。



3.3 デバイスを AWS に登録する

AWS のデモプロジェクト実行に必要な準備をします。下記のチュートリアルを参考に AWS の設定をしてください。

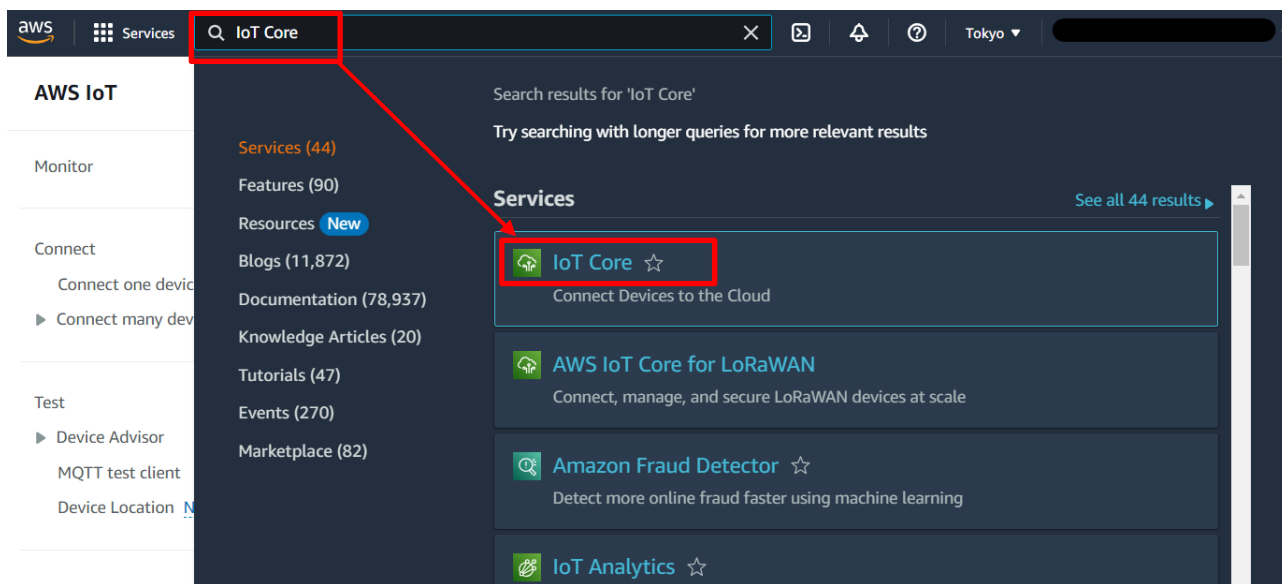
3.3.1 ポリシーの設定

接続するデバイスに対して AWS のリソースなどへアクセス許可（ポリシー）を設定します。

本アプリケーションノートで接続するデバイスには、以下のポリシーを設定します。

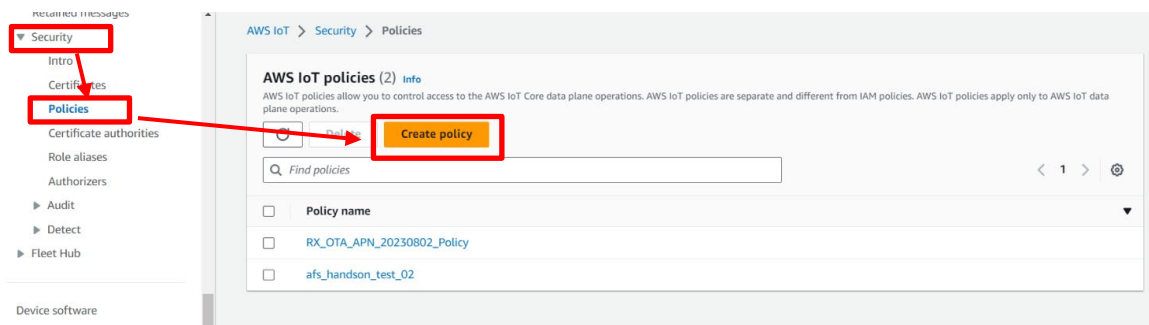
- `iot:Connect` : AWS IoT に接続する
- `iot:Publish` : トピックをパブリッシュ（送信）する
- `iot:Subscribe` : トピックをサブスクライブ（受信）する
- `iot:Receive` : AWS IoT からメッセージを受信する

(1) 画面上部の検索窓に「IoT Core」と入力し、検索結果から IoT Core をクリック

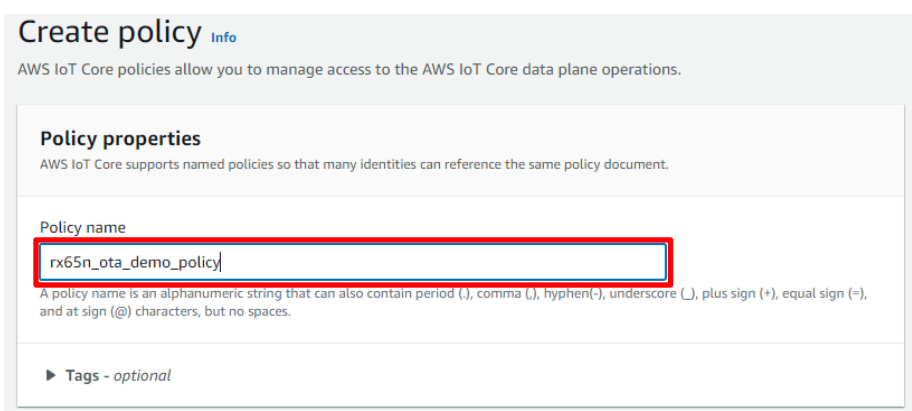


RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(2) メニューの Security (安全性) > Policies (ポリシー) から Create policy (ポリシーを作成) を選択

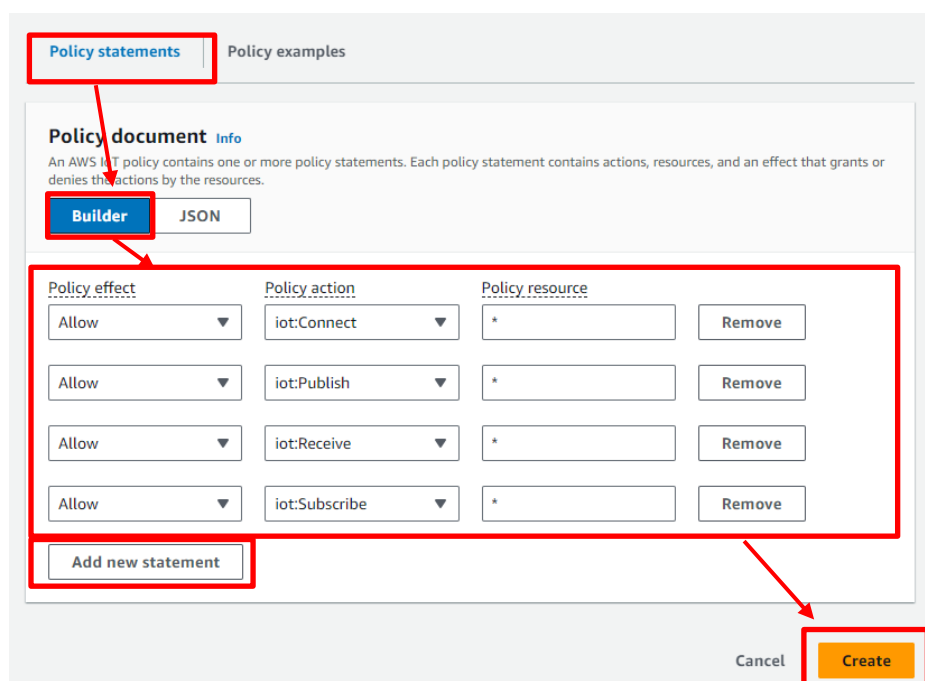


(3) ポリシー名を入力 (例 : rx65n_ota_demo_policy)



(4) Policy statements (ポリシーステートメント) > Policy document (ポリシードキュメント) > Builder (ビルダー)で下図のように設定し、作成をクリック

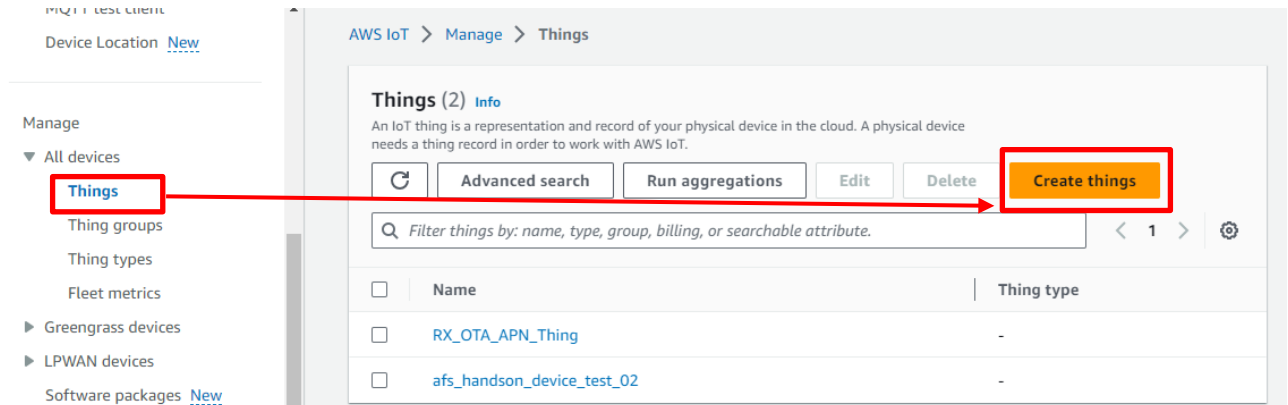
ステートメントは初期状態で 1 つのみのため、"ステートメントを追加 (Add new statement)"を押して追加してください。



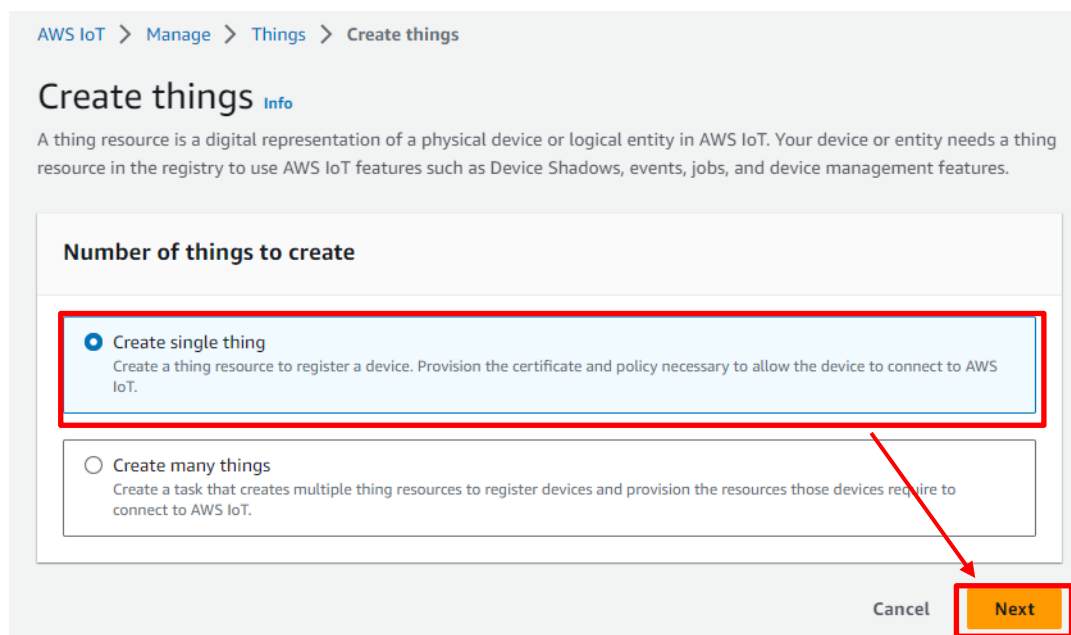
RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

3.3.2 デバイス（モノ）を AWS IoT に登録

- (1) メニューの Manage (管理) > All device (すべてのデバイス) > Things (モノ) から Create things(モノを作成)を選択



- (2) Create single thing (1つのモノを作成)を選択して 次へ をクリック



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(3) モノの名前を入力（例：rx65n_ota_demo_thing）して、次へ をクリック

ここで入力したモノの名前は後の処理で使用するため、メモしておいてください。

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

Step 1
Specify thing properties

Step 2 - optional
Configure device certificate

Step 3 - optional
Attach policies to certificate

Specify thing properties [Info](#)

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

Thing properties [Info](#)

Thing name

rx65n_ota_demo_thing

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

Additional configurations

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

Device Shadow [Info](#)

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

☒ No shadow

☐ Named shadow
Create multiple shadows with different names to manage access to properties, and logically group your devices properties.

☐ Unnamed shadow (classic)
A thing can have only one unnamed shadow.

Cancel **Next**

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

- (4) Device certificate (デバイス証明書) > Auto-generate a new certificate (新しい証明書を自動生成) を選択して 次へ をクリック

Step 1
Specify thing properties

Step 2 - optional
Configure device certificate

Step 3 - optional
Attach policies to certificate

Configure device certificate - optional [Info](#)

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

Device certificate

☒ Auto-generate a new certificate (recommended)
Generate a certificate, public key, and private key using AWS IoT's certificate authority.

☐ Use my certificate
Use a certificate signed by your own certificate authority.

☐ Upload CSR
Register your CA and use your own certificates on one or many devices.

☐ Skip creating a certificate at this time
You can create a certificate for this thing and attach a policy to the certificate at a later time.

Cancel Previous **Next**

- (5) 証明書にポリシーをアタッチ

「3.3.1 ポリシーの設定」で作成したポリシーを選択して、Create thing (モノを作成)をクリック

Step 1
Specify thing properties

Step 2 - optional
Configure device certificate

Step 3 - optional
Attach policies to certificate

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

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

Policies (1/3) [Refresh](#) [Create policy](#)

Select up to 10 policies to attach to this certificate.

	Name
<input checked="" type="checkbox"/>	rx65n_ota_demo_policy
<input type="checkbox"/>	[Redacted]
<input type="checkbox"/>	[Redacted]

Cancel Previous **Create thing**

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(6) 証明書／キーファイルをダウンロード

証明書、プライベートキーはデバイス（モノ）における、パスワードに相当します。デバイスに証明書、プライベートキーを登録することで、これらを使用してデバイスが AWS に接続可能となります。

証明書、パブリックキー、プライベートキーは必ずここでダウンロードしてください。パブリックキー、プライベートキーは、以降ダウンロードできません。

Download certificates and keys

×

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

Device certificate

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

Device certificate
[redacted].pem.crt

Deactivate certificate

Download

Key files

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

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

Public key file
[redacted]-public.pem.key

Download

Private key file
[redacted]-private.pem.key

Download

Root CA certificates

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

Amazon trust services endpoint
RSA 2048 bit key: Amazon Root CA 1

Download

Amazon trust services endpoint
ECC 256 bit key: Amazon Root CA 3

Download

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

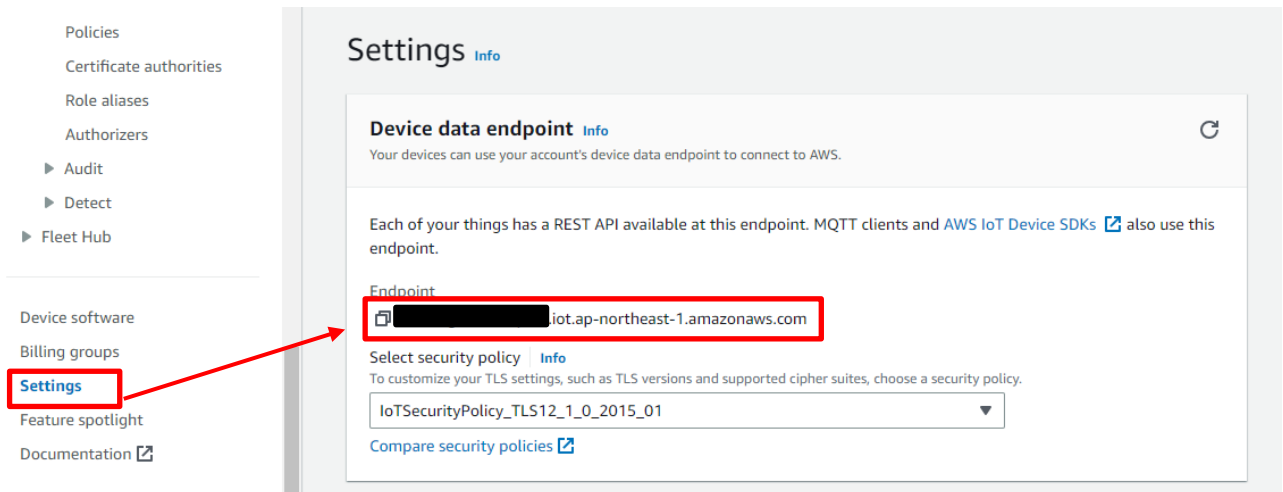
Done

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

3.3.3 エンドポイントの確認

エンドポイントはデバイス(モノ)における接続先(URL)に相当します。デバイスにエンドポイントを登録することで、デバイスは指定したエンドポイントに接続します。

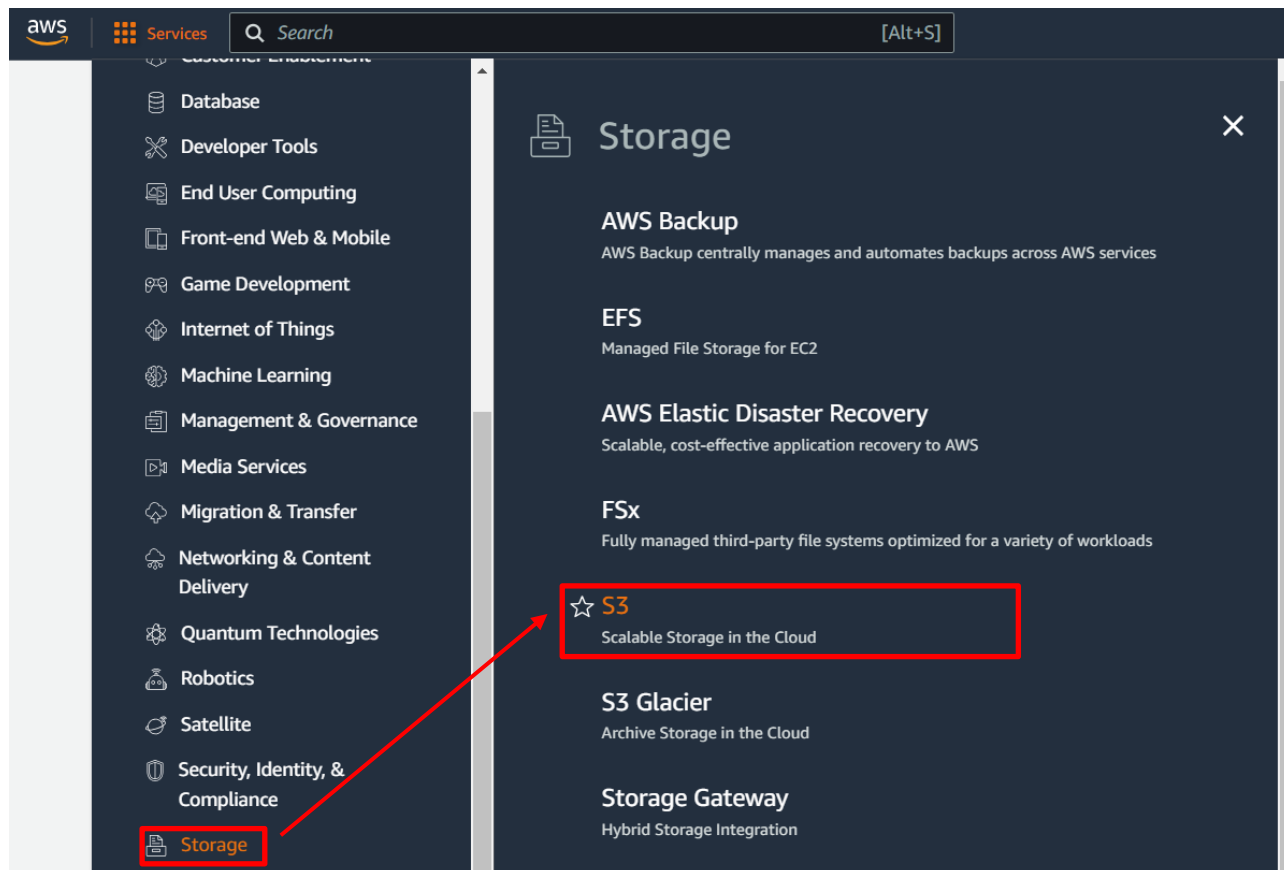
(1) メニューの Settings (設定) からエンドポイントをメモしておきます。



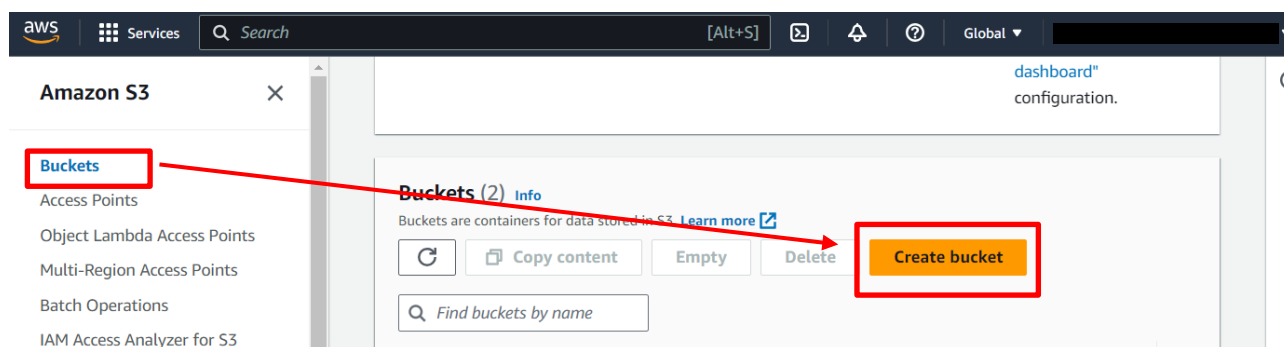
3.4 Amazon S3 バケットの作成

Amazon S3 はオンラインストレージの Web サービスで、更新用ファームウェアを格納するために使用します。

(1) サービスメニューから Storage(ストレージ) > S3 を選択



(2) Buckets (バケット) から Create bucket (バケットを作成)を選択



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(3) バケット名を入力（例：s3test-rx65n）

Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)

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

General configuration

Bucket name

s3test-rx65n

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

AWS Region

Asia Pacific (Tokyo) ap-northeast-1

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

Choose bucket

バケット名はグローバルで一意である必要があります。以下のようなエラーメッセージが表示された場合、すでに使用されている名前のため別の名前を使用してください。

Bucket name

s3test

 Bucket with the same name already exists

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(4) バケットの作成

以下の設定を行い、Create bucket (バケットの作成)をクリック

- Block Public Access setting for this bucket (このバケットのブロックパブリックアクセス設定):
Block all public access (パブリックアクセスをすべて ブロック)
- Bucket Versioning (バケットのバージョンング) : 有効にする

Block Public Access settings for this bucket

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

☒ **Block all public access**

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

☒ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☒ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.

☒ **Block public access to buckets and objects granted through *new* public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☒ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Bucket Versioning

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

Bucket Versioning

☐ Disable

☒ Enable

► Advanced settings

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

Cancel

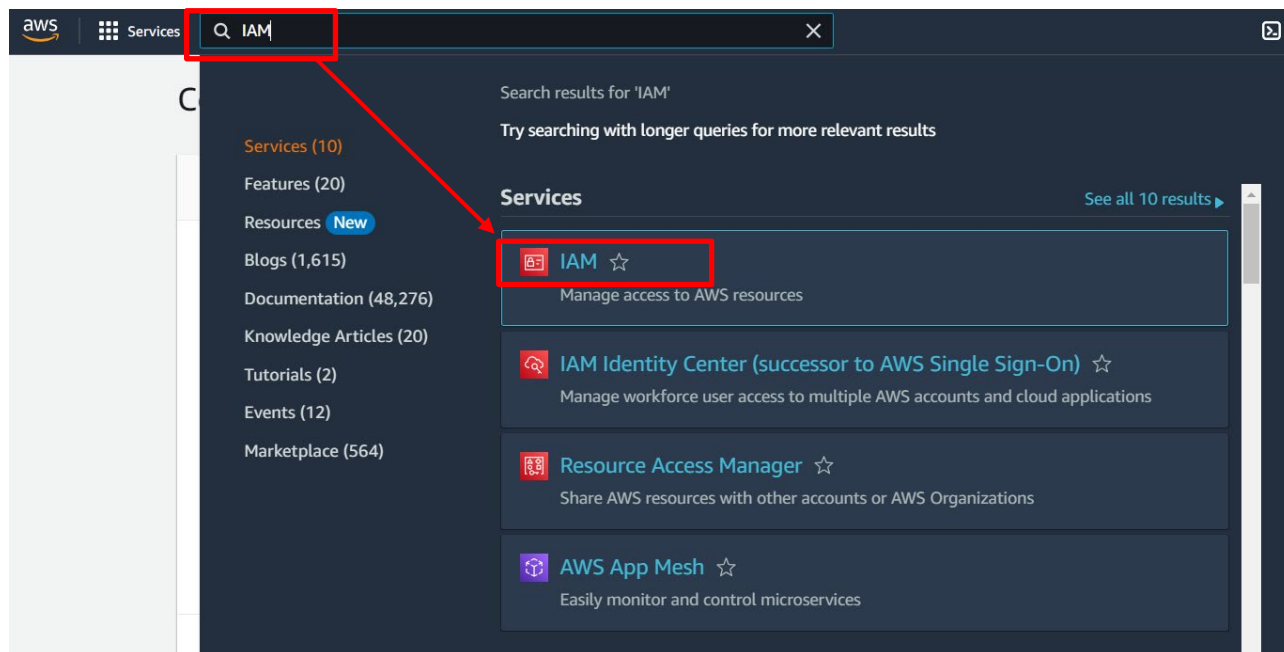
Create bucket

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

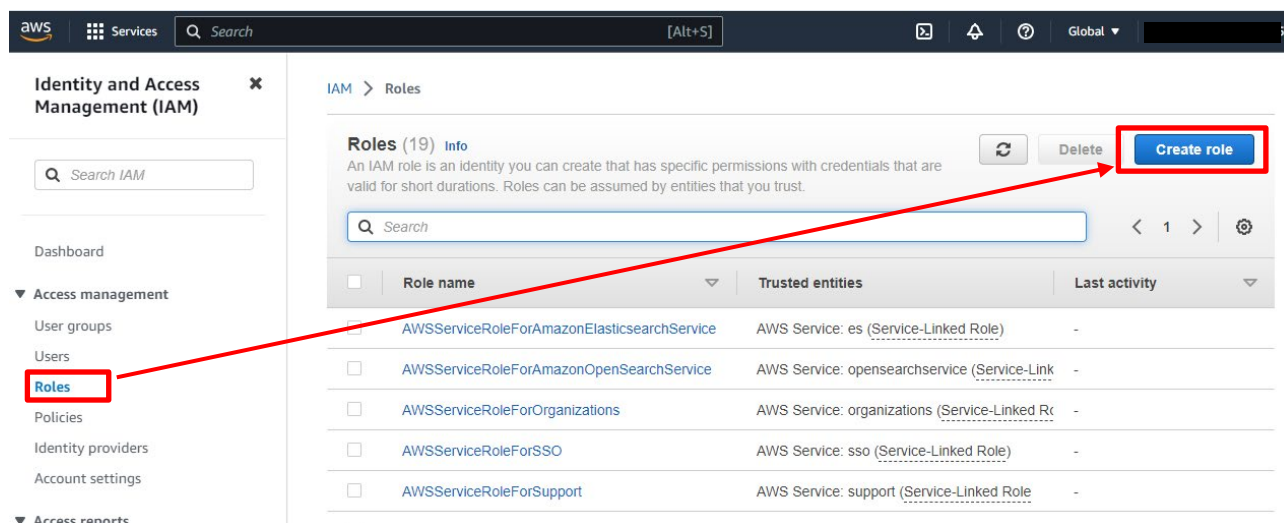
3.5 IAM ユーザーに OTA の実行権限を割り当てる

OTA 更新ジョブを作成するためにアクセス権限が付与されたロールを作成します。

(1) 画面上部の検索窓に「IAM」と入力し、検索結果から IAM をクリック



(2) メニューの Roles (ロール) > Create role (ロールを作成)を選択



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(3) Select trusted entity (信頼されたエンティティを選択) にて以下設定をして次へ

- Trusted entity type (信頼されたエンティティタイプ)で AWS service (AWS サービス)を選択
- Use cases for other AWS services (他の AWS サービスのユースケース)に IoT を選択
- IoT にチェック

[IAM](#) > [Roles](#) > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity [Info](#)

Trusted entity type

☒ AWS service

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

☐ AWS account

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

☐ Web identity

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

☐ SAML 2.0 federation

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

☐ Custom trust policy

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

Use case

Q IoT

IoT

IoT RoboRunner

IoT SiteWise

IoT Things Graph

IoT TwinMaker

IoT

☒ IoT

Allows IoT to call AWS services on your behalf.

☐ IoT - Device Defender Audit

Provides AWS IoT Device Defender read access to IoT and related resources.

☐ IoT - Device Defender Mitigation Actions

Provides AWS IoT Device Defender write access to IoT and related resources for execution of Mitigation Actions.

[Cancel](#)

[Next](#)

(4) Add permissions (許可を追加)は変更せず次へ

Add permissions [Info](#)

Permissions policies (3) [Info](#)

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

Policy name 🔗	Type	Attached entities
AWSIoTRuleActions	AWS m...	1
AWSIoTLogging	AWS m...	1
AWSIoTThingsRegi...	AWS m...	1

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

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

[Cancel](#)

[Previous](#)

[Next](#)

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(5) ロール名を入力 (例 : ota_role_rx65n)して Create role (ロールを作成)をクリック

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

ota_role_rx65n

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

Description

Add a short explanation for this role.

Allows IoT to call AWS services on your behalf.

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

Add tags - optional [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel

Previous

Create role

(6) 作成したロールをクリック

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

IAM > Roles

Roles (20) [Info](#)

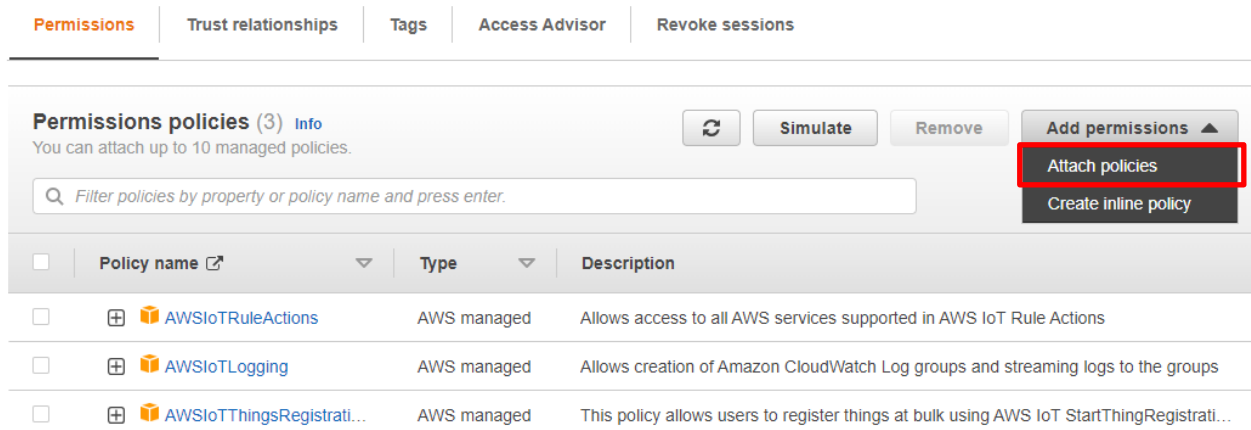
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

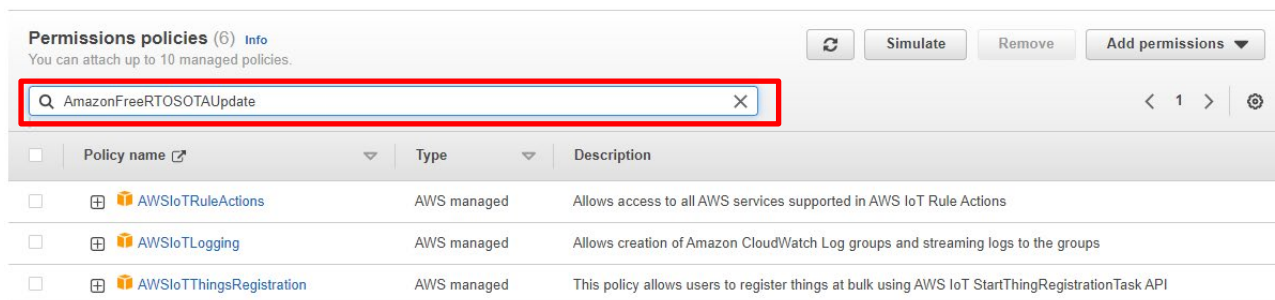
<input type="checkbox"/>	Role name	Trusted entities
<input type="checkbox"/>	AWSServiceRoleForAmazonElasticsearchService	AWS Service: es (Service)
<input type="checkbox"/>	AWSServiceRoleForAmazonOpenSearchService	AWS Service: opensearch
<input type="checkbox"/>	AWSServiceRoleForOrganizations	AWS Service: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

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(7) 「Attach policies (ポリシーをアタッチ)」を選択



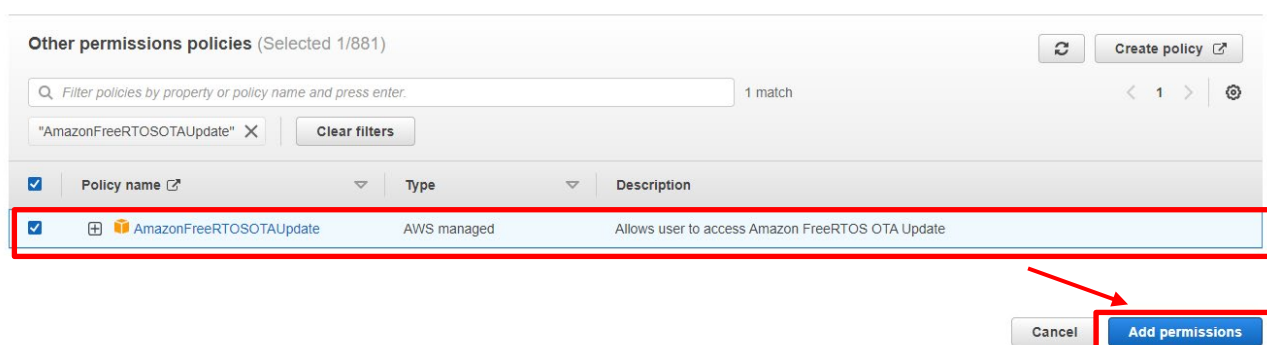
(8) Permissions policies (許可ポリシー)で検索ボックスに” AmazonFreeRTOSOTAUpdate”を入力して Enter キーを押す



(9) ” AmazonFreeRTOSOTAUpdate”にチェックをして Add permissions (許可を追加)をクリック

Attach policy to ota_role_rx65n

► Current permissions policies (3)



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(10) Add permissions (許可を追加)から Create inline policy (インラインポリシーを作成)を選択

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

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

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

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

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(11) JSON をクリックして、以下コードを張り付けて 次へ をクリック

AWS 各種サービスに IAM ロールを渡す許可を設定します。

[張り付けるコード]

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

Specify permissions [Info](#)

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

Policy editor

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "iam:GetRole",
8         "iam:PassRole"
9       ],
10      "Resource": "*"
11    }
12  ]
13 }
14
```

Visual

JSON

Actions ▼

Edit statement

Select a statement

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

[+ Add new statement](#)

JSON Ln 14, Col 0

10128 of 10240 characters remaining

Security: 1 Errors: 0 Warnings: 0 Suggestions: 0

Cancel

Next

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(12) ポリシー名を入力(例 : rx65n_ota_demo_iam_policy)して Create policy (ポリシーを作成)をクリック

Review and create

Review the permissions, specify details, and tags.

Policy details

Policy name

Enter a meaningful name to identify this policy.

rx65n_ota_demo_iam_policy

Maximum 128 characters. Use alphanumeric and +,=, @, _ characters.

Permissions defined in this policy [Info](#)

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

Search

Allow (1 of 384 services)

Show remaining 383 services

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

Cancel

Previous

Create policy

(13) 再度、Add permissions (許可を追加)から Create inline policy (インラインポリシーを作成)を選択

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

Permissions policies (4) [Info](#)

You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter.



Simulate

Remove

Add permissions

Attach policies

Create inline policy

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

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(14) JSON をクリックして、以下コードを張り付けて 次へ をクリック

更新ファームウェアを格納する Amazon S3 へのアクセス許可を追加します。

[張り付けるコード]

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

Specify permissions Info

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

Policy editor

Visual **JSON** Actions ▼

Edit statement

Select a statement

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

[+ Add new statement](#)

JSON Ln 17, Col 0 999 of 10128 characters remaining

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

Cancel **Next**

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(15) ポリシー名を入力(例 : rx65n_ota_demo_s3_policy)して Create policy (ポリシーを作成)をクリック

Review and create

Review the permissions, specify details, and tags.

Policy details

Policy name

Enter a meaningful name to identify this policy.

rx65n_ota_demo_s3_policy

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

Permissions defined in this policy [Info](#)

Edit

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

Search

Allow (1 of 384 services)

Show remaining 383 services

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

Cancel

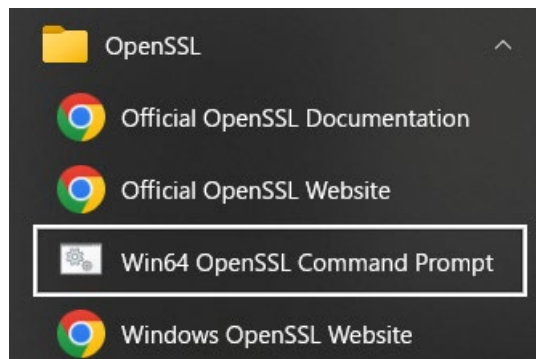
Previous

Create policy

4. デバイスの設定

4.1 鍵ペアと証明書の生成

- (1) スタートメニューから Win64 OpenSSL Command Prompt を起動する



- (2) ECDSA の CA 秘密鍵を作成するコマンドを実行

以下のコマンドを実行します。

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

[実行結果]

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

- (3) 作成した CA 秘密鍵から CA 証明書を作成するコマンドを実行

以下のコマンドを実行します。Country Name 以降は任意の文字列を入力してください。

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

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

任意の文字列を入力する

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(4) ECDSA の鍵ペアを作成するコマンドを実行

以下のコマンドを実行します。

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

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

(5) 作成した ECDSA の鍵ペアから証明書署名要求を作成するコマンドを実行

以下のコマンドを実行します。Country Name 以降は任意の文字列を入力してください。最後の 2 行は空白のまま Enter を押してください。

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

```
C:\¥openssl>openssl req -new -sha256 -key secp256r1.keypair > secp256r1.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:JP
State or Province Name (full name) [Some-State]:Tokyo
Locality Name (eg, city) []:Kodaira
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Renesas Electronics
Organizational Unit Name (eg, section) []:Software Development Division
Common Name (e.g. server FQDN or YOUR name) []:Renesas Tarou
Email Address []:Tarou.Renesas@sample.com
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

任意の文字列を入力する

空白のまま Enter を押す

(6) 作成した証明書署名要求/CA 証明書/CA 秘密鍵から証明書を作成するコマンドを実行

以下のコマンドを実行します。

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

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

(7) ECDSA の鍵ペアから秘密鍵を抽出するコマンドを実行

以下のコマンドを実行します。

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

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

(8) ECDSA の鍵ペアから公開鍵を抽出するコマンドを実行

以下のコマンドを実行します。

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

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

4.2 初期バージョンのファームウェア構築

初期バージョンのファームウェアの構築を行います。

4.2.1 プロジェクトのインポート

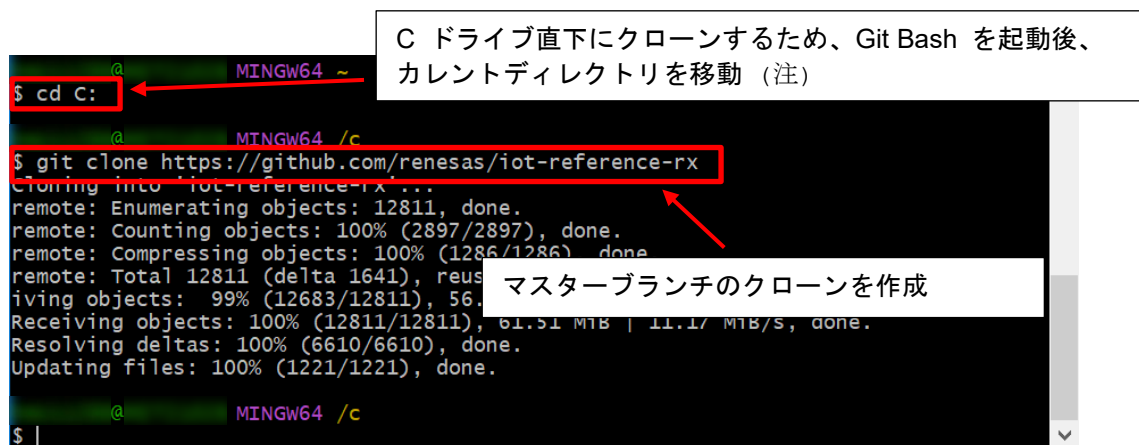
(1) デモプロジェクトのクローン

GitHub ([iot-reference-rx : FreeRTOS reference repository](https://github.com/renesas/iot-reference-rx)) からデモプロジェクトをクローンします。本ドキュメントでは、[Git for Windows](#) を使用した場合のクローン方法を説明します。

Git Bash を起動し、以下のコマンドを実行してください。

```
cd c:
```

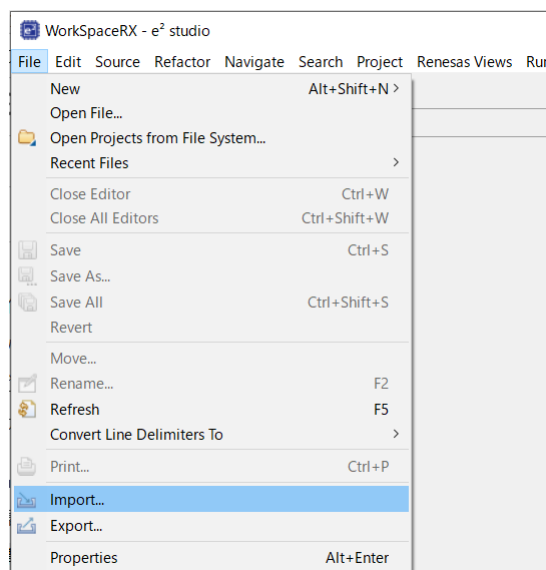
```
git clone https://github.com/renesas/iot-reference-rx
```



【注】 e² studio に制限があるため、クローン先のパス長（任意のフォルダ名を含む）は 35 文字以内としてください。36 文字以上を指定するとプロジェクトのビルド時にエラーとなります。
上記例では、C ドライブ直下にクローンしています。

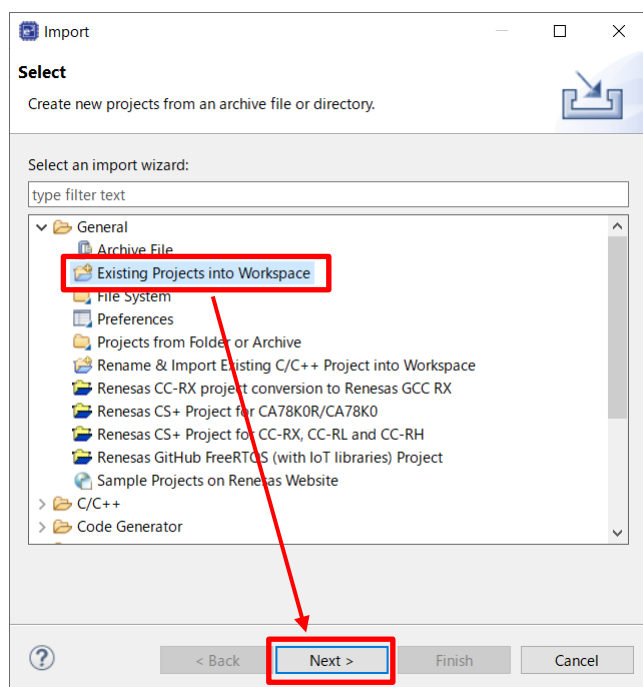
(2) e² studio を起動

(3) [File] > [Import...]を選択



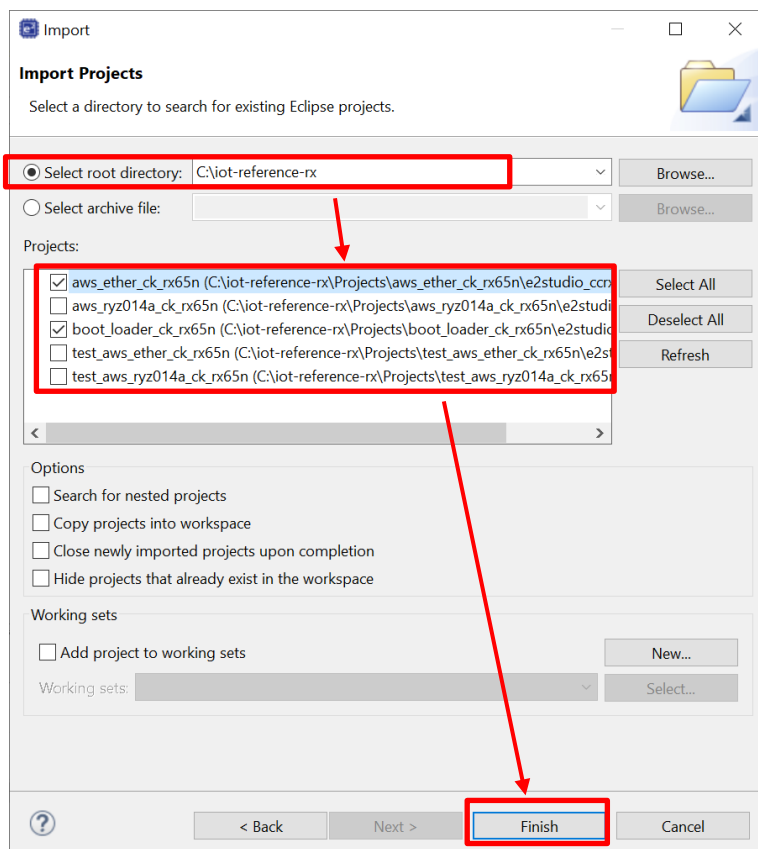
RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(4) Existing Projects into Workspace (既存プロジェクトをワークスペースへ)を選択



(5) Select root directory で 4.2.1(1)にてクローンしたフォルダを選択し、以下のプロジェクトにチェックして Finish をクリック

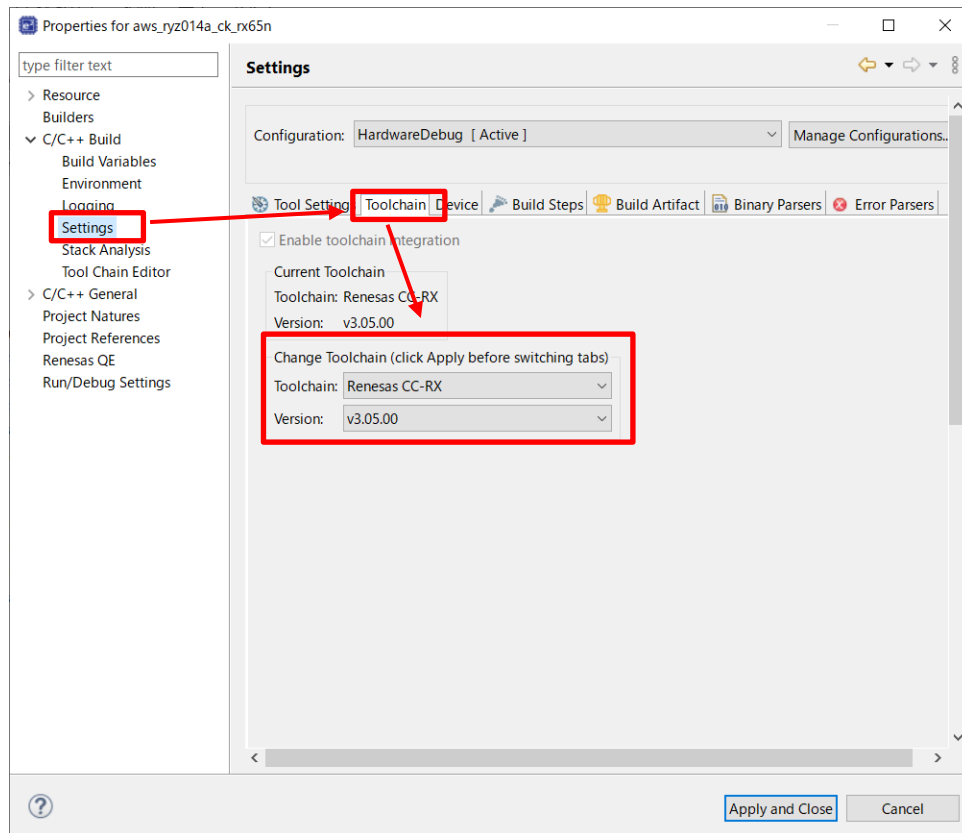
- aws_ryz014a_ck_rx65n
- boot_loader_ck_rx65n



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

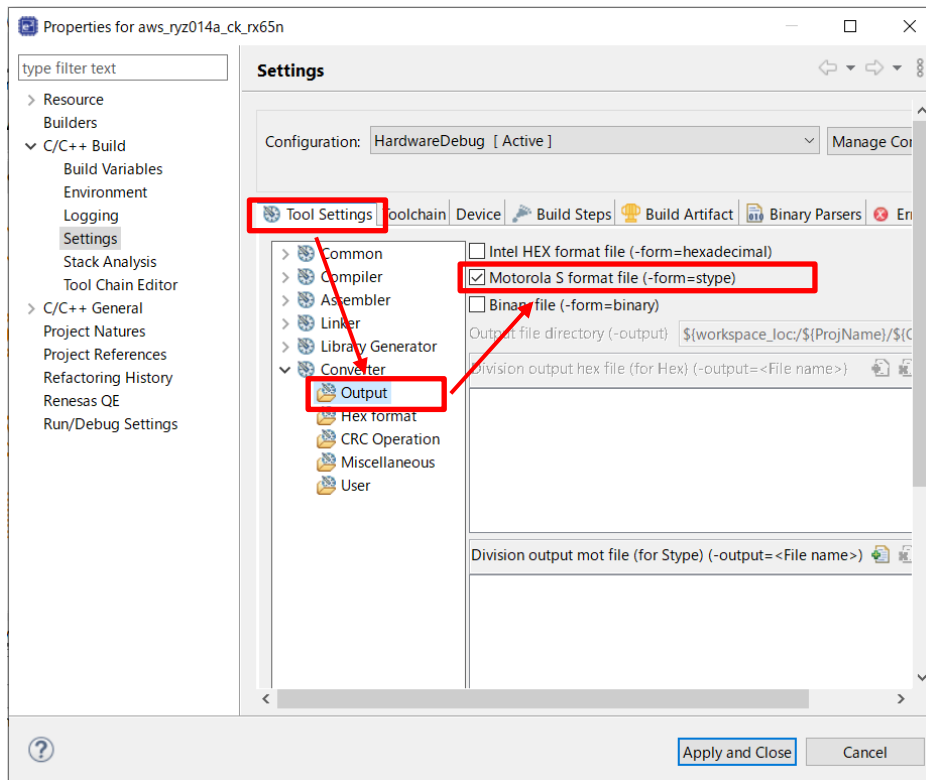
4.2.2 プロジェクト環境設定の確認

- (1) 両方のプロジェクトの [プロジェクト] > [プロパティ] > [C/C++ビルド] > [設定] からツールチェーンを開き、ツールチェーンが Renesas CC-RX になっていることを確認



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

- (2) ツール設定を選択し、[Converter] > [出力]から [モトローラ S 形式ファイルを出力する]が選択されていることを確認



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

4.2.3 プロジェクト設定

(1) 各プロジェクトに公開鍵を設定する

4.1(8)で作成した secp256r1.publickey の内容をコピーし、以下ファイルに定義されている CODE_SIGNENR_PUBLIC_KEY_PEM に張り付けます。

boot_loader_ck_rx65n ¥src¥key¥code_signer_public_key.h

boot_loader_ck_rx65n ¥src¥key¥code_signer_public_key.h の CODE_SIGNENR_PUBLIC_KEY_PEM に公開鍵を張り付けます。

各行は"で囲み、行の最後には¥ が必要です、忘れずに入力してください
(例: "XX"¥

```
1  -----BEGIN PUBLIC KEY-----
2  MFkwEwYHKOZIZj0CAQYIKoZIj0DAQcDQgAEjEx1/hqcMQIsZYRzx5SUGQd2SNh
3  3mSoCn4LCcF3wD+Ua8Gr3dYKpJKy1081NvfTD6+OoB+DpVbBIAy04iD0dQ==
4  -----END PUBLIC KEY-----
5
```

```
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

/* Copyright (C) 2023 Renesas Electronics Corporation. All rights reserved.
 * File Name : code_signer_public_key.h
 * History : DD.MM.YYYY Version Description
 */

#ifndef CODE_SIGNER_PUBLIC_KEY_H
#define CODE_SIGNER_PUBLIC_KEY_H

/*
 * PEM-encoded code signer public key.
 * Must include the PEM header and footer:
 * "-----BEGIN CERTIFICATE-----"
 * "-----END CERTIFICATE-----"
 */

#define CODE_SIGNER_PUBLIC_KEY_PEM \
    "-----BEGIN PUBLIC KEY-----" \
    "MFkwEwYHKOZIZj0CAQYIKoZIj0DAQcDQgAEjEx1/hqcMQIsZYRzx5SUGQd2SNh" \
    "3mSoCn4LCcF3wD+Ua8Gr3dYKpJKy1081NvfTD6+OoB+DpVbBIAy04iD0dQ==" \
    "-----END PUBLIC KEY-----"

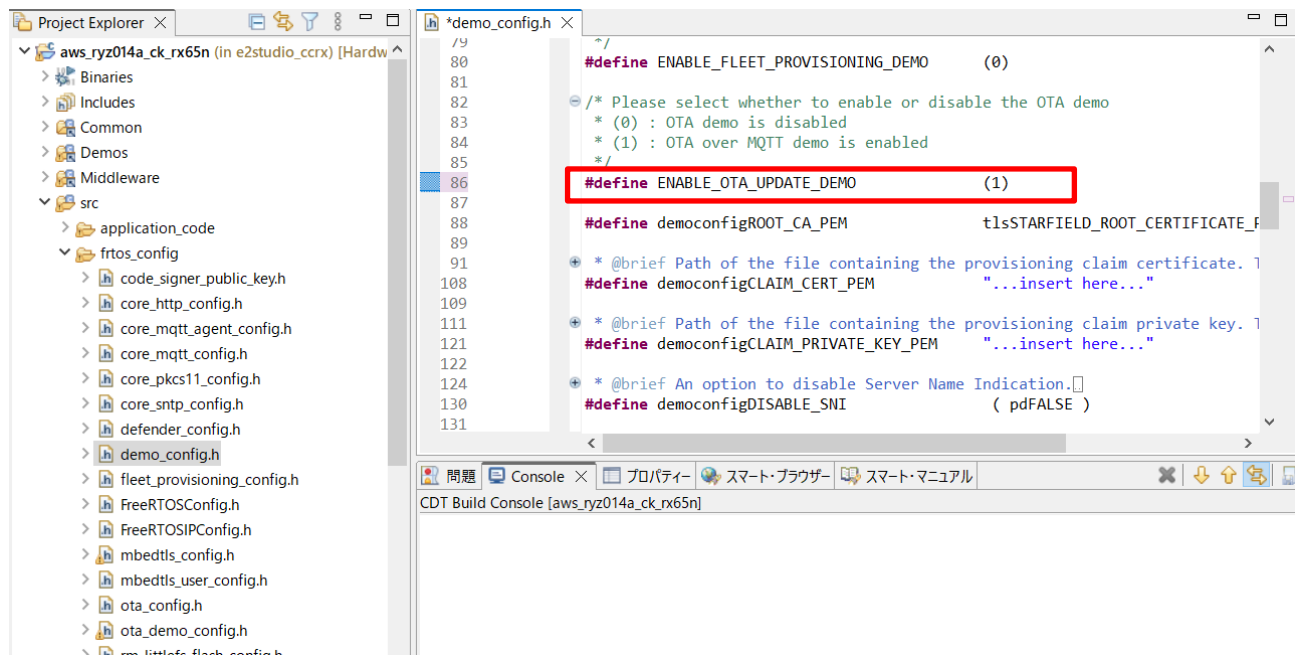
extern const uint8_t g_code_signer_public_key[];
extern const uint32_t g_code_signer_public_key_length;

#endif /* CODE_SIGNER_PUBLIC_KEY_H */
```


RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(2) OTA アップデートデモの定義を許可にする

aws_ryz014a_ck_rx65n¥src¥rtos_config¥demo_config.h にある ENABLE_OTA_UPDATE_DEMO 定義を 1 (許可)に設定してください。(デフォルト 0)

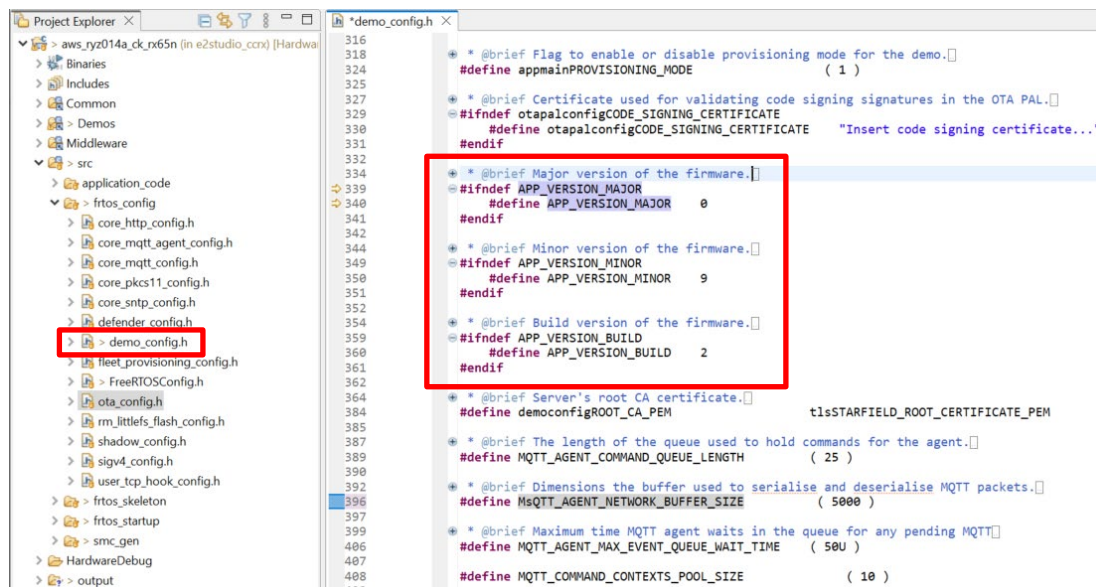


RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(3) プロジェクトの初期バージョンが 0.92 であることを確認する

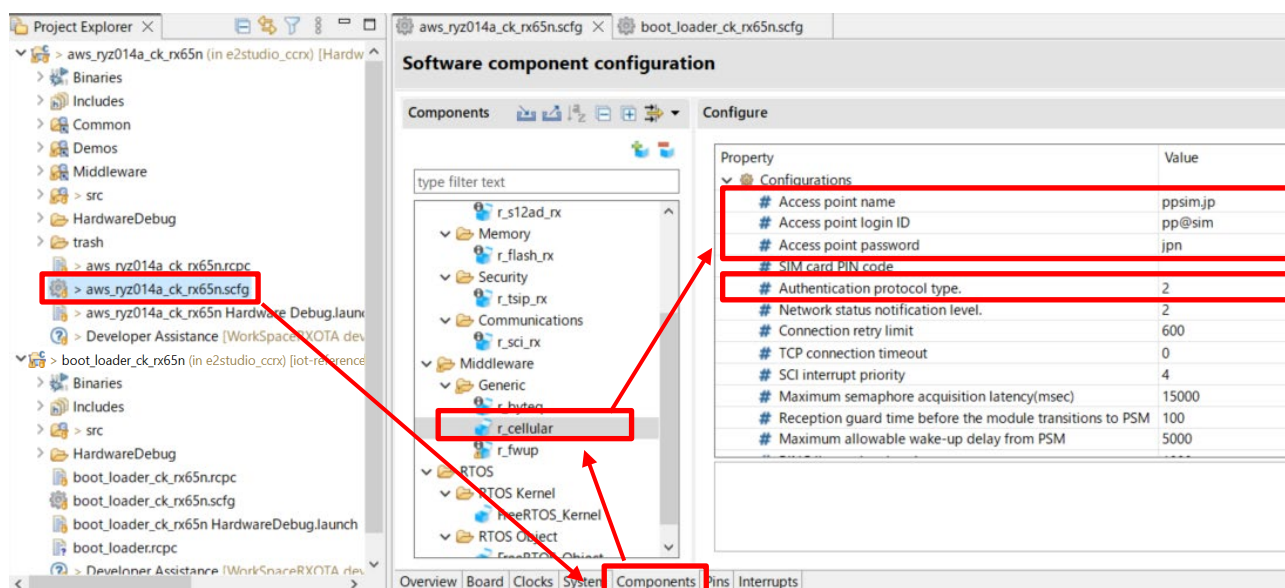
aws_ryz014a_ck_rx65n\src\rtos_config\demo_config.h のバージョン定義が以下になっていることを確認します。

- APP_VERSION_MAJOR 0
- APP_VERSION_MINOR 9
- APP_VERSION_BUILD 2



(4) RYZ014A Cellular モジュール制御 FIT モジュール(r_cellular)の設定

aws_ryz014a_ck_rx65n.scfg を開き、Components タグを選択。r_cellular の Access point name、Access point login ID、Access point password、Authentication protocol type を SIM カードに合わせて設定してください。



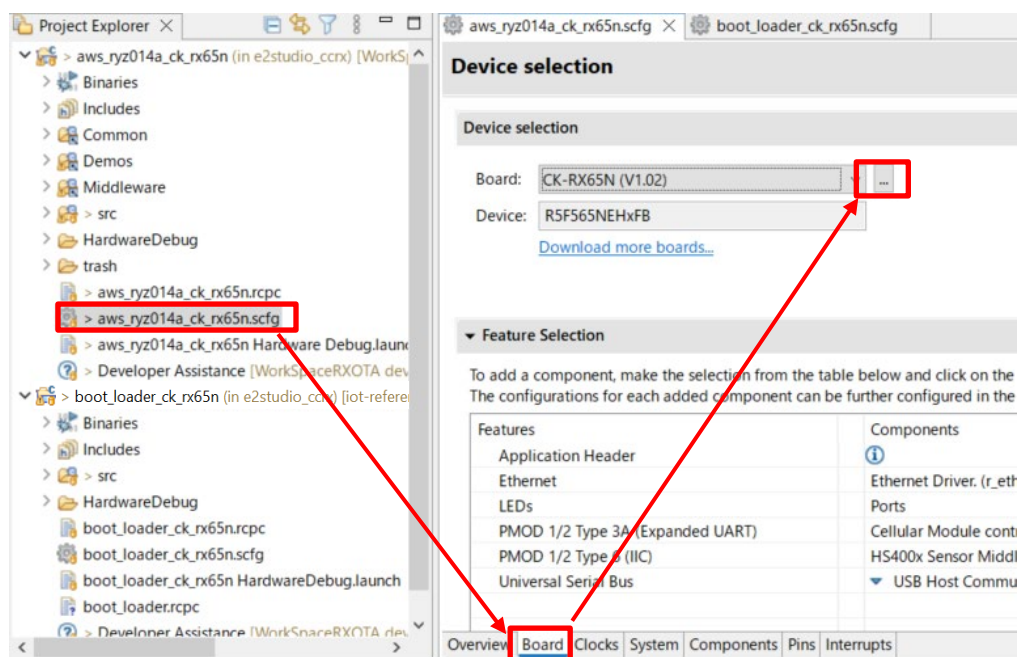
RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

CK-RX65N 付属の SIM カードを使用する場合は、以下アプリケーションノートの「4.1.5 Activating SIM card」を参照し、SIM カードのアクティベーションを行ってください。

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

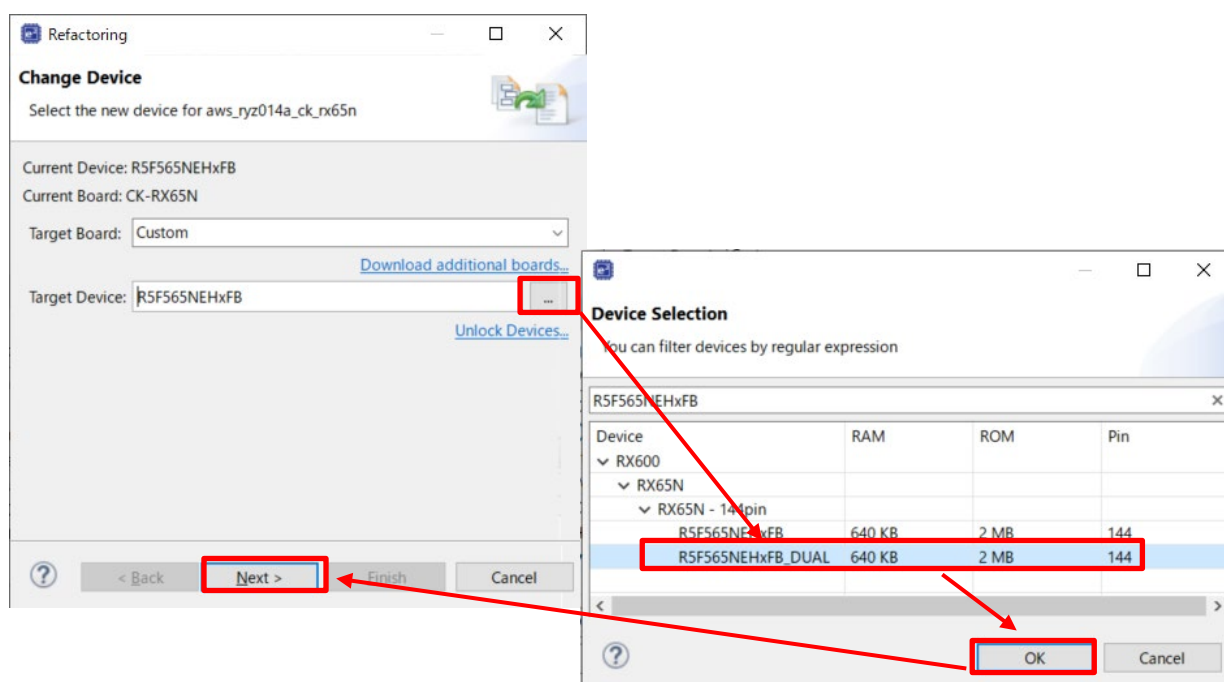
(5) ファームウェアのデバイス設定(1)

aws_ryz014a_ck_rx65n.scfg を開き、Board タグを選択。Board 選択の"... "をクリックしてください。



(6) ファームウェアのデバイス設定(2)

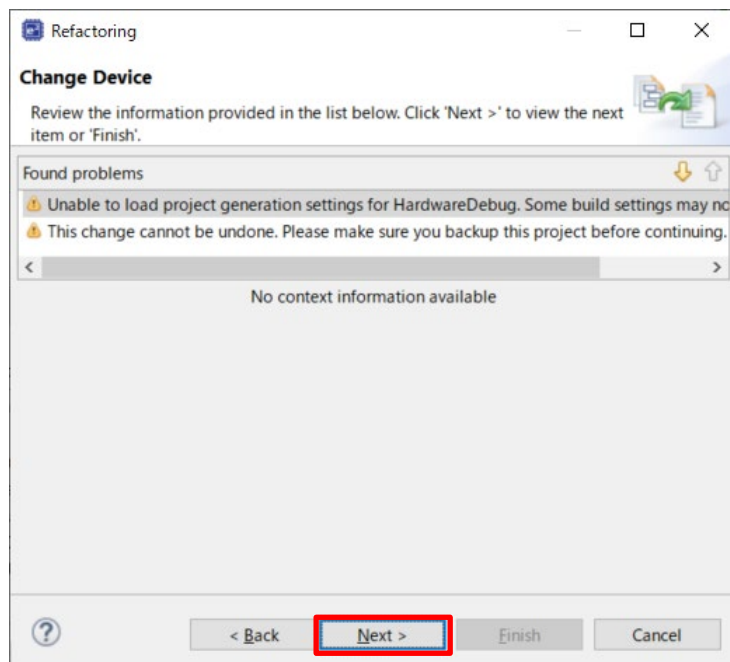
Target Device の"... "をクリックして" R5F565NEHxFB_DUAL"を選択してください。Target Board は"Custom"になります。



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

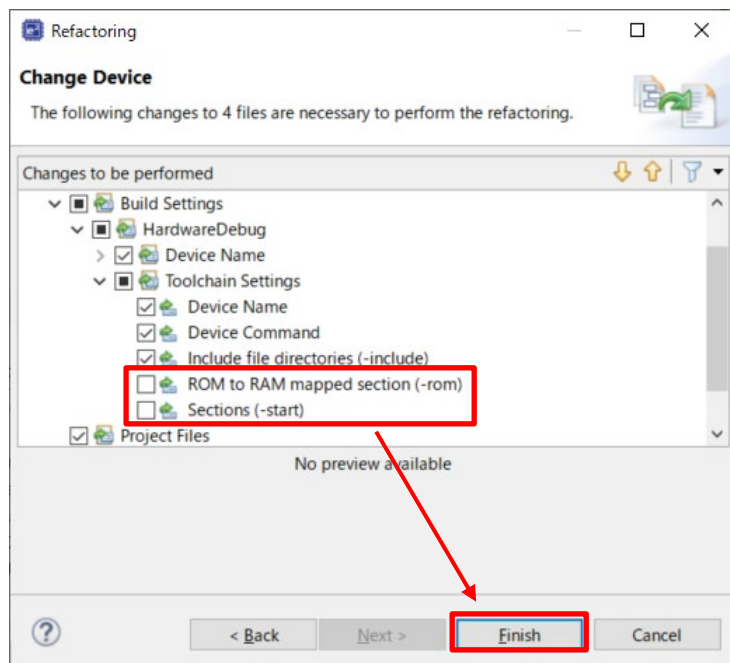
(7) ファームウェアのデバイス設定(3)

デバイスの変更を行うと以下の画面になるため”次へ”をクリック。



(8) ファームウェアのデバイス設定(4)

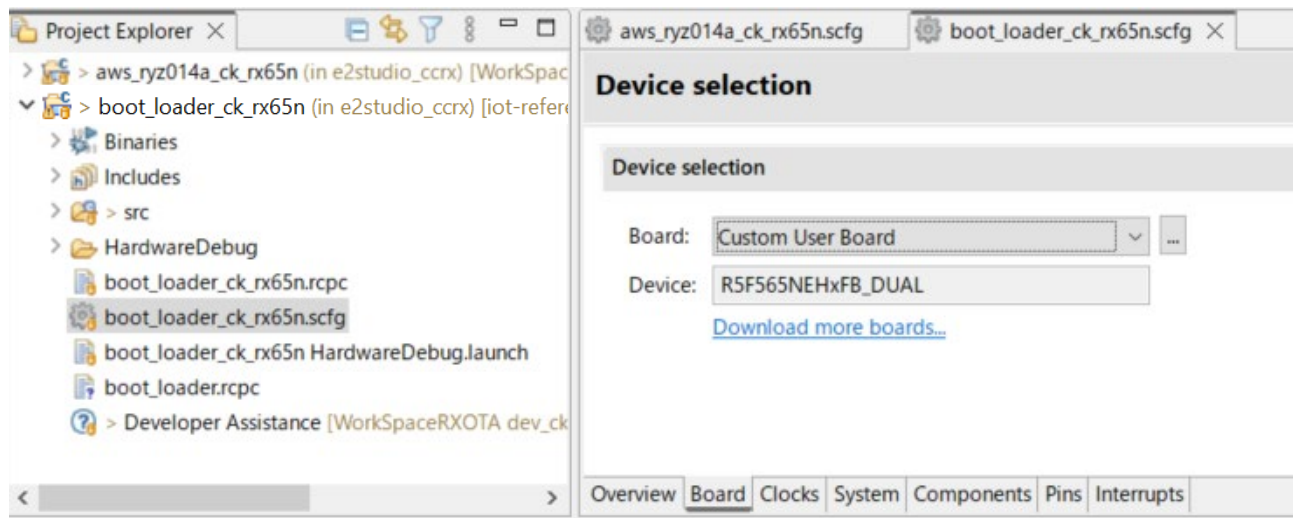
[Build Settings]>[HardwareDebug]>[Toolchain Settings]にある、[ROM から RAM へマップするセクション (-rom)]と[セクション (-start)]のチェックを外し、“終了”をクリックしてください。



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(9) ブートローダのデバイス確認

boot_loader_ck_rx65n.scfg を開き、Board タグを選択。Device が” R5F565NEHxFB_DUAL”になっていることを確認してください。



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

4.2.4 初期ファームウェアの作成

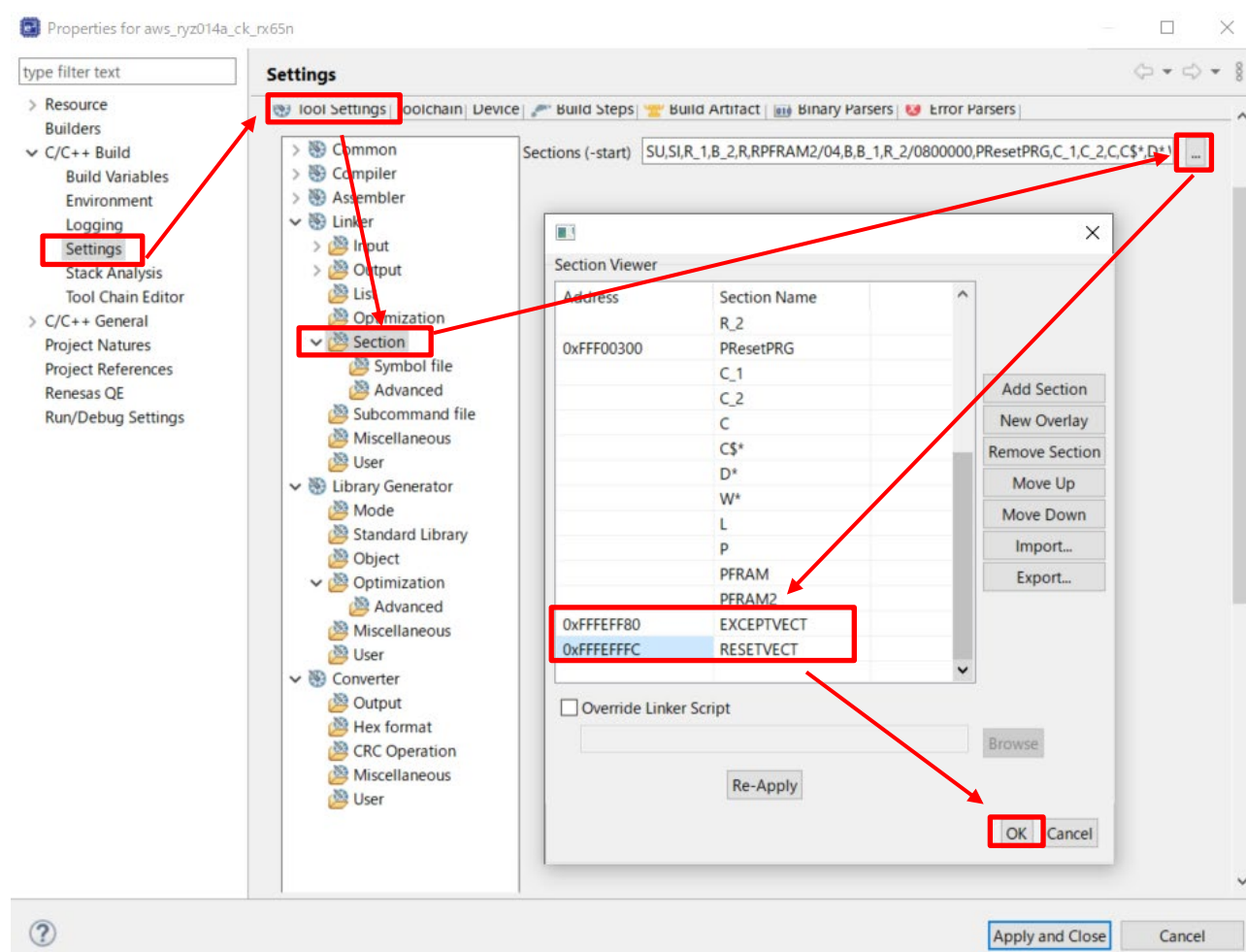
ブートローダ(boot_loader_ck_rx65n)とファームウェア(aws_ryz014a_ck_rx65n)を結合して初期ファームウェアを作成します。

(1) ファームウェア(aws_ryz014a_ck_rx65n)のペクタ変更

aws_ryz014a_ck_rx65n のプロジェクトから、[Project] > [Properties]を選択。

C/C++ Build > Settings から、Tool Settings の Linker > Section から Section Viewer を開き、EXCEPTVECT を 0xFFFFF80 に、RESETVECT を 0xFFFFEFC に割り当てます。

その後、ビルドを行います。



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(2) Renesas Image Generator を使用して初期ファームウェアを生成

Renesas Image Generator フォルダに以下のファイルを格納します。

- 4.2.4(1)でビルド実施結果 aws_ryz014a_ck_rx65n.mot
- ブートローダのビルド実施結果 boot_loader_ck_rx65n.mot
- 4.1(7)で作成した秘密鍵 secp256r1.privatekey

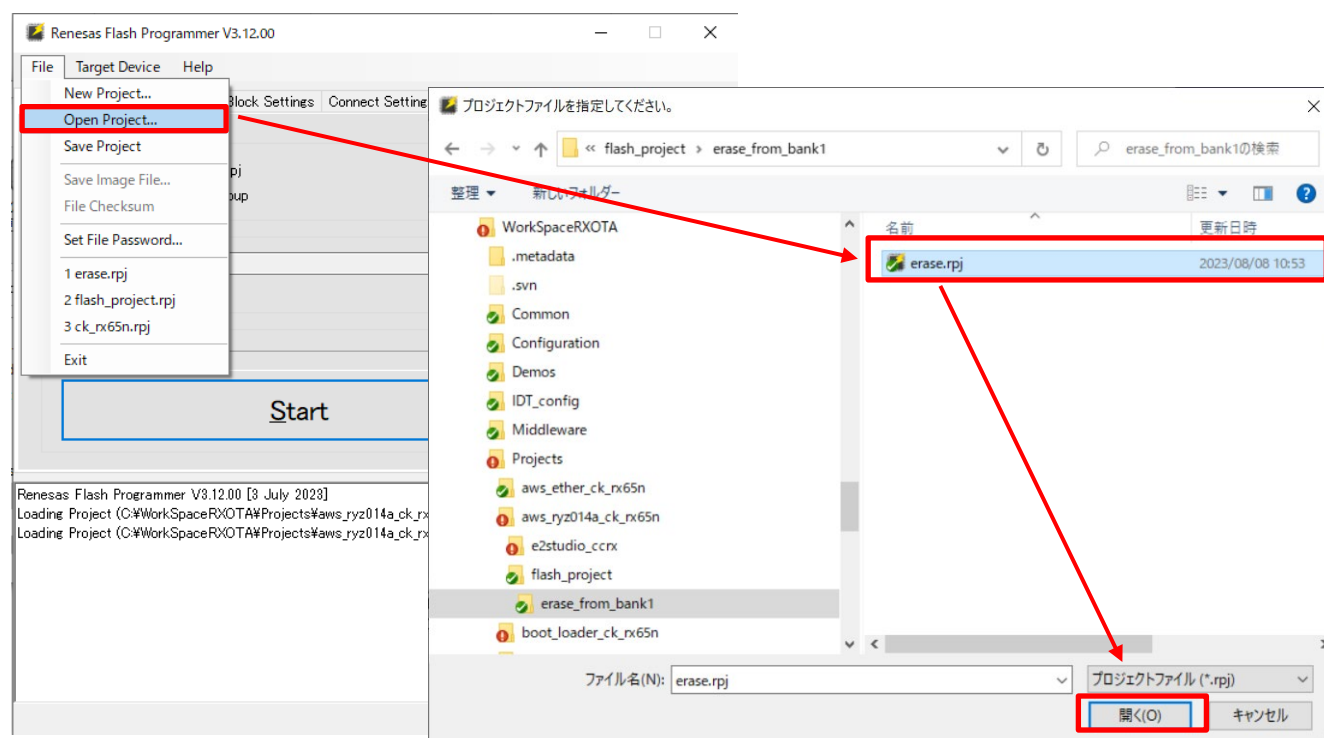
コマンドプロンプトを起動し、Renesas Image Generator フォルダへ移動して以下のコマンドを実行すると、userprog.mot ファイルが生成されます。

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

(3) Renesas Flash Programmer を起動し、erase.rpj プロジェクトを開く

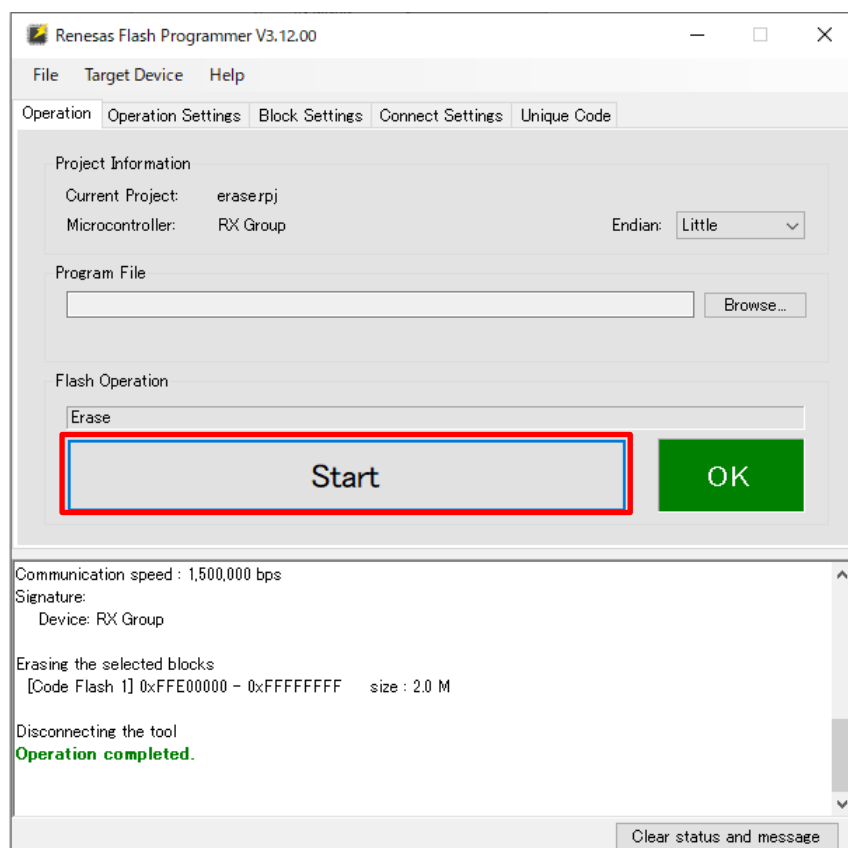
erase.rpj プロジェクトは、本サンプルプログラムの以下フォルダにあります。

¥Projects¥aws_ryz014a_ck_rx65n¥flash_project¥erase_from_bank1



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

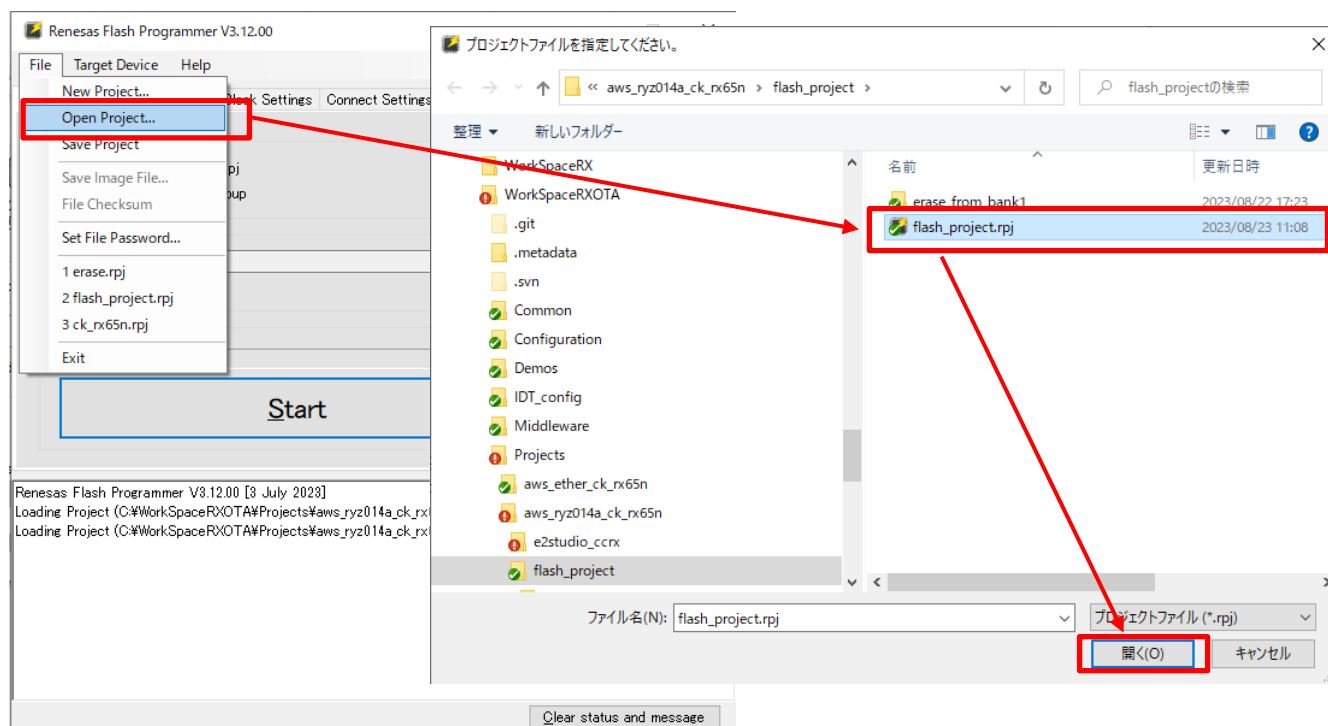
(4) スタートを押し、デバイスのイレーズを実施



(5) flash_project.rpj プロジェクトを開く

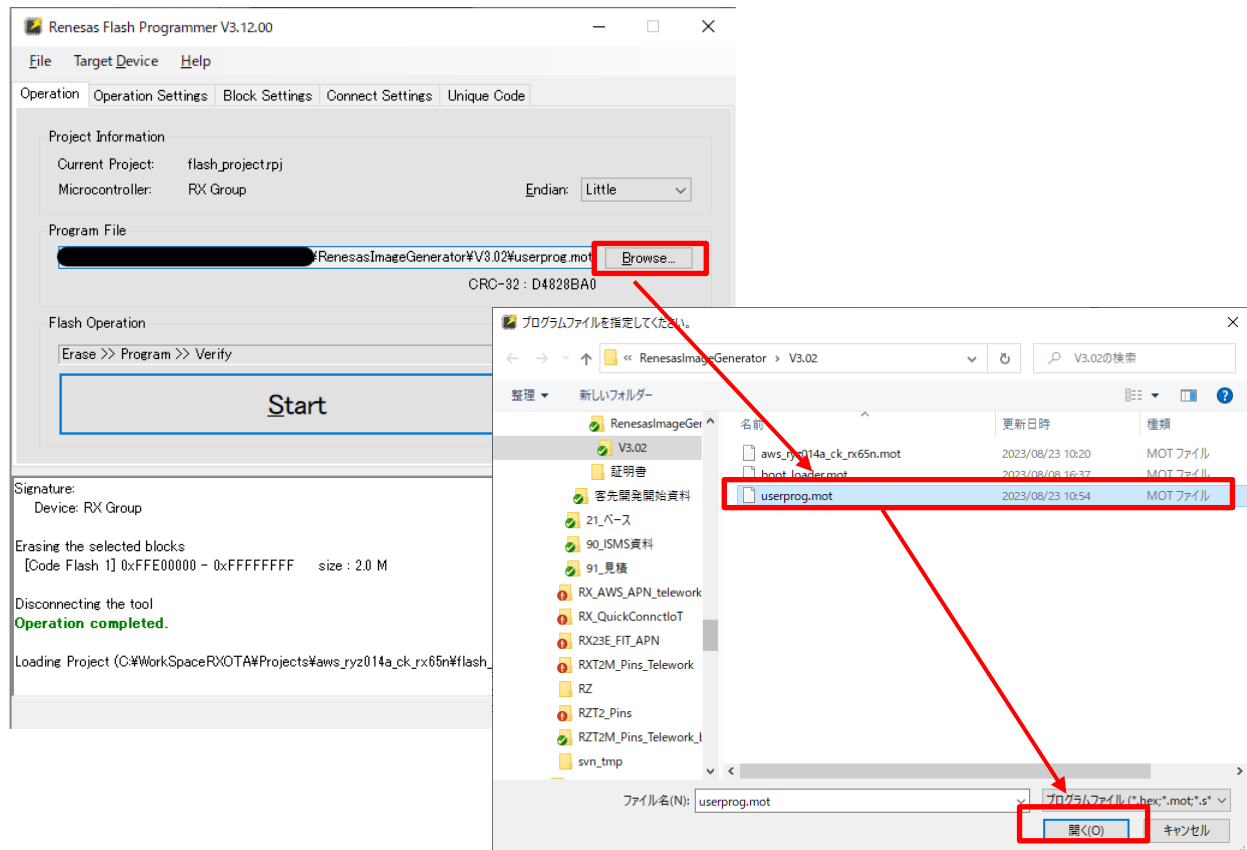
flash_project.rpj プロジェクトは、本サンプルプログラムの以下フォルダにあります。

¥Projects¥aws_ryz014a_ck_rx65n¥flash_project¥

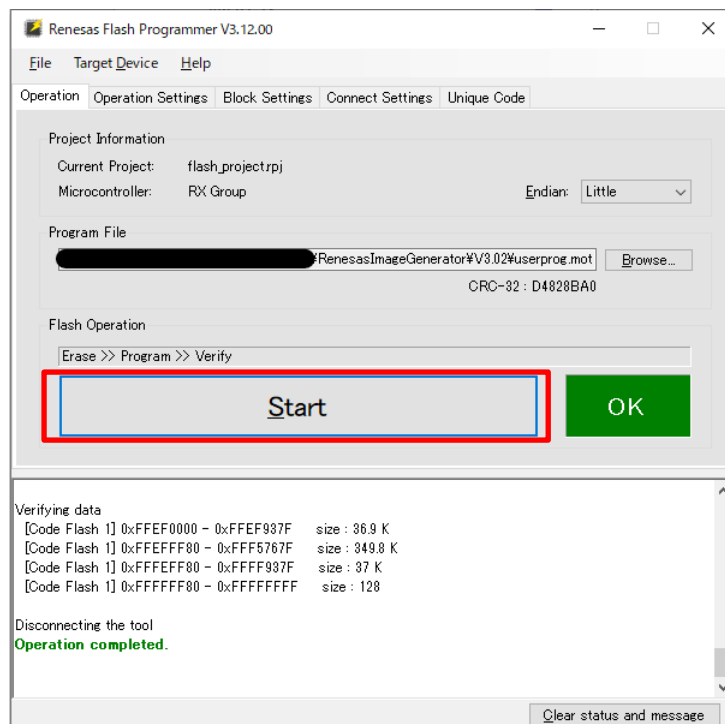


RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(6) 4.2.4(2)で作成した初期ファームウェア(userprog.mot)を選択する



(7) ファームウェアの書き込みを行う

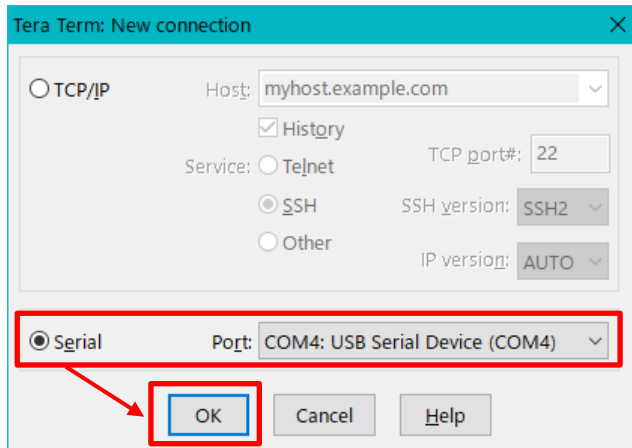


RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

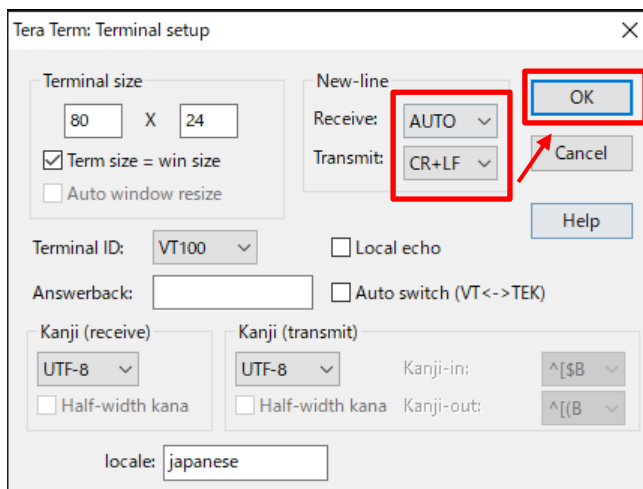
4.2.5 AWS IoT 情報の登録

AWS IoT の情報は aws_ryz014a_ck_rx65n を動作させて、TeraTerm にて設定します。設定した情報はデータフラッシュに書き込まれます。

- (1) TeraTerm を起動して、メニューの File > New Connection... から、Serial を選択して OK をクリック

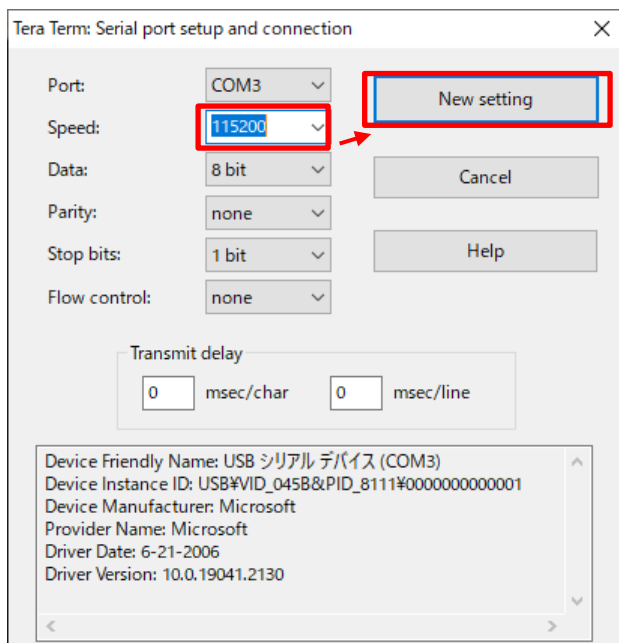


- (2) メニューの Setup > Terminal...を開き、New-line の Receive を Auto、Transmit を CR+LF を選択して OK をクリック

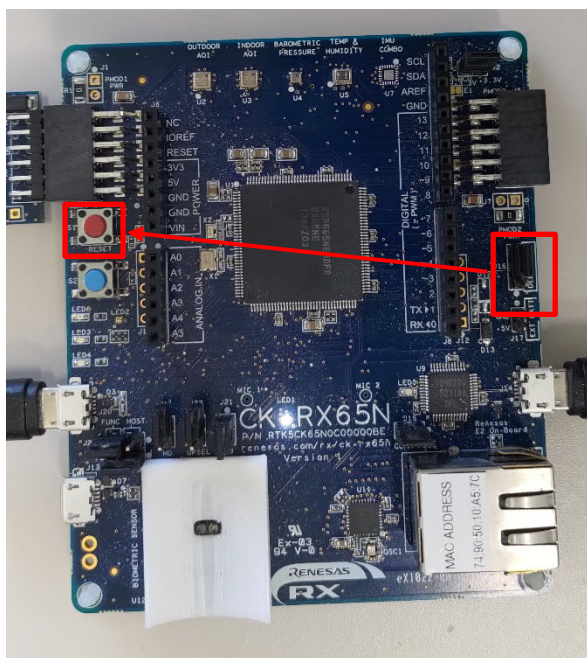


RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(3) メニューの Setup > Serial port...を開いて Speed を 115200 に設定して New setting をクリック



(4) CK-RX65N の J16 を RUN 側に接続し、RESET SW を押す



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(5) TeraTerm の画面にメニューが表示されたのち、10 秒以内に「CLI」と入力して Enter キーを押す

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

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

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

>CLI

Going to FreeRTOS-CLI !
```

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(6) 3.3.2(6)でダウンロードした証明書を登録する

TeraTerm にて「conf set cert」と入力したのち、証明書ファイル(****-certificate.pem.crt)を TeraTerm にドラッグアンドドロップ（ファイル送信）してください。最後に TeraTerm 上で「Enter」を押してください。

**“conf set cert”を入力した後に、
証明書ファイルをドラッグアンドドロップする
その後”Enter”を押す**

```
>conf set cert -----BEGIN CERTIFICATE-----
MIIDWjCCAkKgAwIBAgIIVAPNArvr2KKsndktYgzI9g1yKM5epMA0GCSqG
SwwUAME0xSzBjBjNBBAQwMkFtYXpvcjBjZWIgU2VydmljZXMgTz1BbWF6b
IEIuYy4gTD1TZWF0dGxIIFNUPVdhc2hpbmd0b24gQz1VUzAeFw0yMzA4M
MTNaFw000TEyMzEyMzU5NTIaMB4xHDAaBgNVBAMMEQFXUyBj1QgQ2Vyc
dGUwggeEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAAoIBAQDQWdPrR7e7hF
7w9oRyG6pe3OSgUkYzb18Pz9m6dioAaz1W4I8f0Xs/V+fX3bQYi+zCZfE
rjSX+cJ0TVo+kIEsqvqLyk8n2GZeyMfCIhaSHUej3WDhS15eQBAsPGPQU
HfaT/wgGZy6UiUekD5DsfWDmpgCWrf7kUiqW2HRAN4vhjQAhdBSe/g1um
+b1PogxvVomqW5GgZEzoGgKQybkNUpdRWZ4nei63ownEZ10mMufBiigE
5Koc0ejAKd7VuKLTCT+WmsIBmE4NnT+Eb28zjFy1IDka4uPUSS0EC+s2dZ
8UaxAgMBAAAgYDBEMB8GA1UdIwQYMBaAFMv9mHrTMHUNxc9NW2Y4bAeZ
A1UdDgQWBBSFCoJQ1BwgVG1kD2zbQKsCHR7X9DAMBgNVHRMBAf8EAjAAM
DwEB/wQEAwIHgDANBgkqhkiG9w0BAQsFAAOCAQEAV57iI29MZxRJ5rQfI
EJM9S0dB8jVZrRwr4Xi6b0EQ9Ejo2KQp59EsPL+cmEBwHUSzQIh3gukr4
BXS+Gue3c3cFsn8AOCaCJbBkmMJW38RWekEJ6yudUSfIbMoNOG1Ps7z9A
7v12ThNPEdMX/KxMbXzIBW6e3njMh+4RaWeRUK4M8j03af87V/G1j1t2
GDuhUXomBtcbhBP2UiIDP7GwcaW+eQgTu+BARnFs1PkTEUCWfQLNkkcbxtoIh
VnPhnz385Wni8RUbSTrKmtroJRngYM0EgLTxXrn+Zew+b61/U7r/R1bKhJ0m
-----END CERTIFICATE-----

OK.
>
```

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(7) 3.3.2(6)でダウンロードしたプライベートキーを登録する

TeraTerm にて「conf set key」と入力したのち、プライベートキーファイル(****-private.pem.key)を TeraTerm にドラッグアンドドロップ（ファイル送信）してください。最後に TeraTerm 上で「Enter」を押してください。

“conf set key “を入力した後に、プライベートキーファイルをドラッグアンドドロップするその後”Enter”を押す

```
>conf set key -----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA1nT60e3u4T1RRdyjBu8PaEchuqXtzkoFJGM25fD8
M5VuCPH9F7P1fn8d20GIvswmXwScWArnyq40I/nCdE1aPpCBLKr6i8iv
wiIwkh1Ho91g4UpeXkAWrDxj0FI6vvBGLx32k/8IBmcu1I1HpA+Q7H1g
FIg1th0QDeL4Y0AB3W0nv4Nbpj0uAxeMfPm8T6IMb1ajJuRoGRM6BoC
XUvmeJ3out6MUjGdTpjLnwYooBE3Mm88h+ScHNHocCne1bii0wrVprCA
G9vM4xctS45GuLj1EktBAvrNnWRjpnqSwfFGsQIDAQAIBAIEFk0kZx
rE7jH8y+wl0hQZqp3JDsh/1P9qZ76mVUwWDTIPImQskhqb+AMiXb+Pcm
dN+s6CRRDuV9dPIRY5Km56sBSZqW/uviszatzCJm0baBIwrUapbF-cGG1
pmYr2bPLVLUmWv8sIJyphJULwyHVX478WYE8QyIzyKm8iyaCU8Fh7tHF
InhkUZh49nwEgtQgEA7tfrVDGYE84i+RVWjdHmptxRpAKEo/KYnT7s1s
oNOB9VYU1mw8VoVewC6B19f1cT6v8bSMY1Cy6jviH36Qy4VCj1ZXmsQD
scSLP0ECgYEA/C2Z19PNu9ZHV+/uvEdK90tCNaG0nc9Q3TL+G3rsp5zt
tMhMQ3zQkfW8sFTwFQLascUdWn+zGs5MDjY2sMHEaNAwNGAg+FPzr1ig
LDbanGp0vEmCtAo2b202Be6yxcX2P1swcMjQC1tRaV3LxIBMCyC9akC
f0pWf6J+tpRA3ukKgx0vJLkVWAtVaoRxxUVL8JiQPNGhD5qvHzBe/ThV
26EhQfePFdh2yYyCiXnL6DgV2IDENEuzsB06QtEZTg6w6HUWSBWeEgN+
wqtCKDuomkakUJIfJWcPQKS5Sap5UcAo0AajgXckCgYEAYb1QTGsY/1CA
1jT6WXxKY01dhMckDUdfVxSotPOxKZJwLvqRLHrBVbVArhwk94comdS+7I6w86R
+CxVwz68pjsi1EKxHPPK0IDKowS6GwN076Fe0pDNoHAM06g+hS7DIriYqyx13yJm
ZPh7kjLEXhnmJLj6S1bQy0ECgYEAvgq6XzQG34+78tU8fz2QNE4oBTyHwe1jYIzQ
ZFhtctDa+jkPXC44E100zYFPXN8DF106rSkwRCEYziKiGgj5y0hdUaxc0ULEVX/
kEUmyuAAjoi1TJFHk2irYNFo1EBjVnliTIjazu41SrzOZ1P590D0bSyhh7ku8G8
vL4ZamkCgYAHIdoyaOPP9PuX5K8IKb7Sgpg7Zy3tTsCKtDASWhB3vLiLWMhQ9B
YN56j1BcRcBIRQcXE5bfAg2uyoqnvhf5/VoYochUZ7/qFkBgpiar1Vhz6Vy9jSrP
Iqkj9SEp+kJ01xCRLLVh8++pDmgpFbdgnp/1DndAmHJfJxfxRvoA==
-----END RSA PRIVATE KEY-----
```

OK.

RXファミリ RX65NにおけるAmazon Web Servicesを利用したFreeRTOS OTAの実現方法(v202210.01-LTS-rx-1.1.0以降対応版)

(8) 3.3.2(3)で設定したモノの名称、および3.3.3(1)で控えたエンドポイントを登録する

TeraTerm で以下のコマンドを実行する

conf set thingname [モノの名前]

conf set endpoint [エンドポイント名]

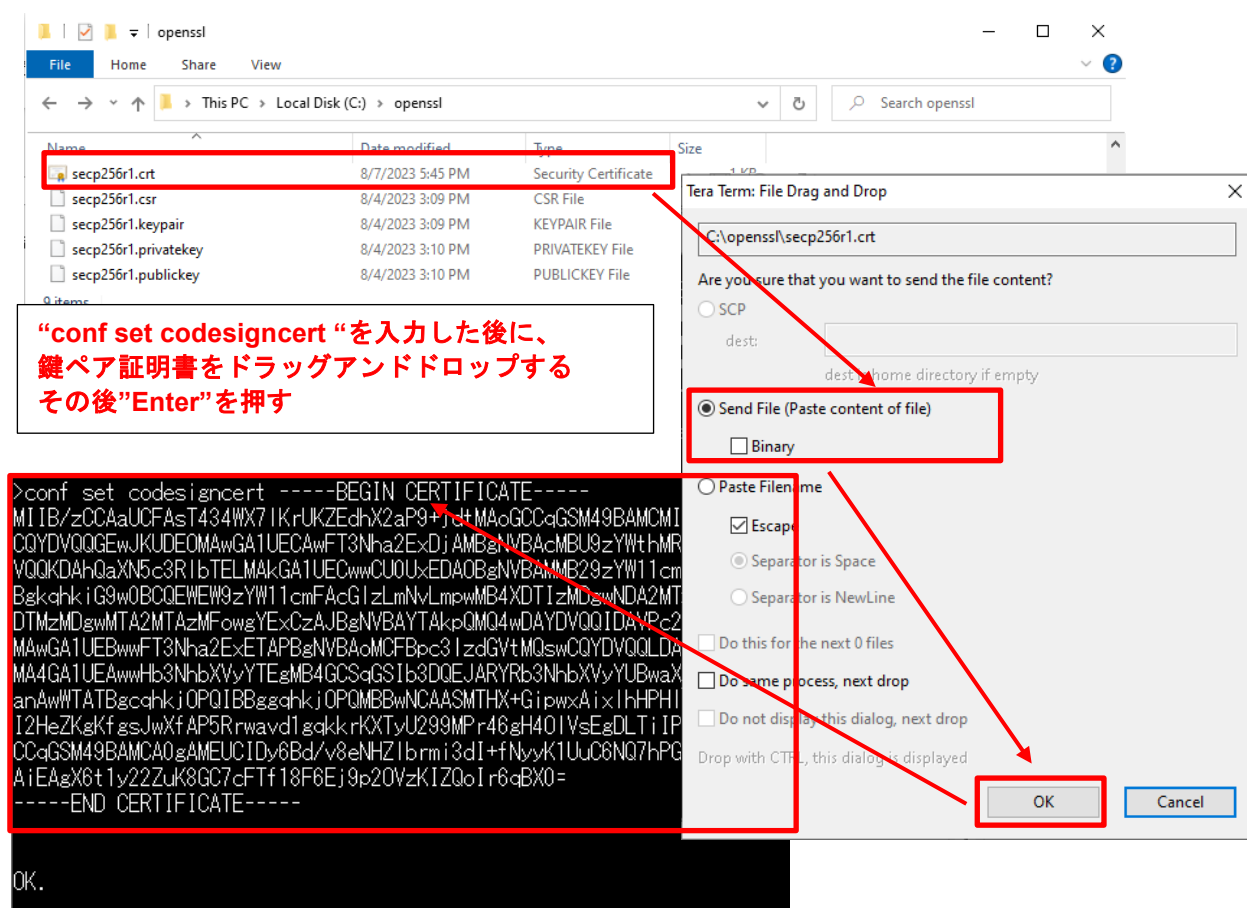
```
>conf set thingname rx65n_ota_demo_thing
OK.

>conf set endpoint <redacted>:ap-northeast-1.amazonaws.com
OK.
```

(9) 4.1(6)で生成した鍵ペア証明書(secp256r1.crt)を登録する

TeraTermにて「conf set codesigncert」と入力したのち、鍵ペア証明書(secp256r1.crt)をTeraTermにドラッグアンドドロップ(ファイル送信)してください。

※証明書ファイルの改行コードはLFに変更してから張り付けてください



(10) AWS IoT の設定を Commit (データフラッシュに書き込み)

TeraTerm で以下のコマンドを実行する

conf commit

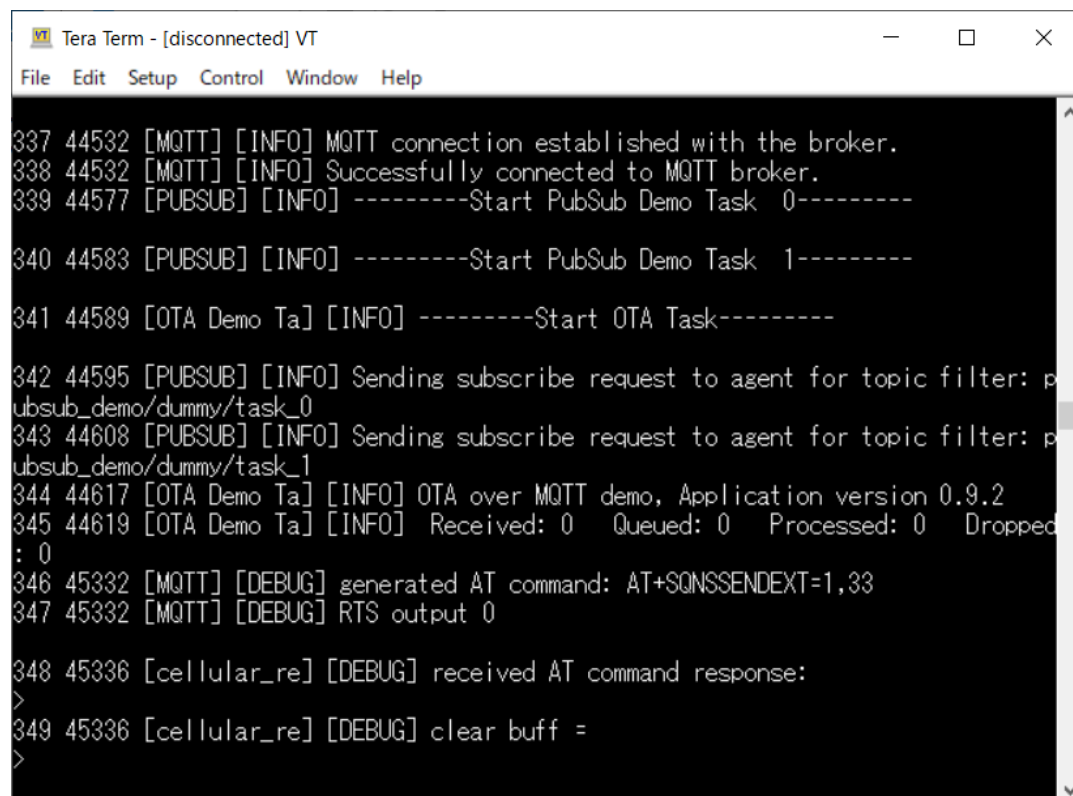
```
>conf commit
0 4472481 [CLI] Destroyed Certificate.
1 4472485 [CLI] Write certificate...
2 4472545 [CLI] Destroyed Private key.
3 4472685 [CLI] Write Private key...
Configuration saved to Data Flash and used 2879 bytes.
```

(11) Reset 実行

TeraTerm で以下のコマンドを実行する

reset

リセット実行後、TeraTerm に通信ログが表示され、OTA ジョブ待ちになっていることを確認する。



The screenshot shows a TeraTerm window titled "Tera Term - [disconnected] VT". The menu bar includes File, Edit, Setup, Control, Window, and Help. The terminal output displays the following log entries:

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

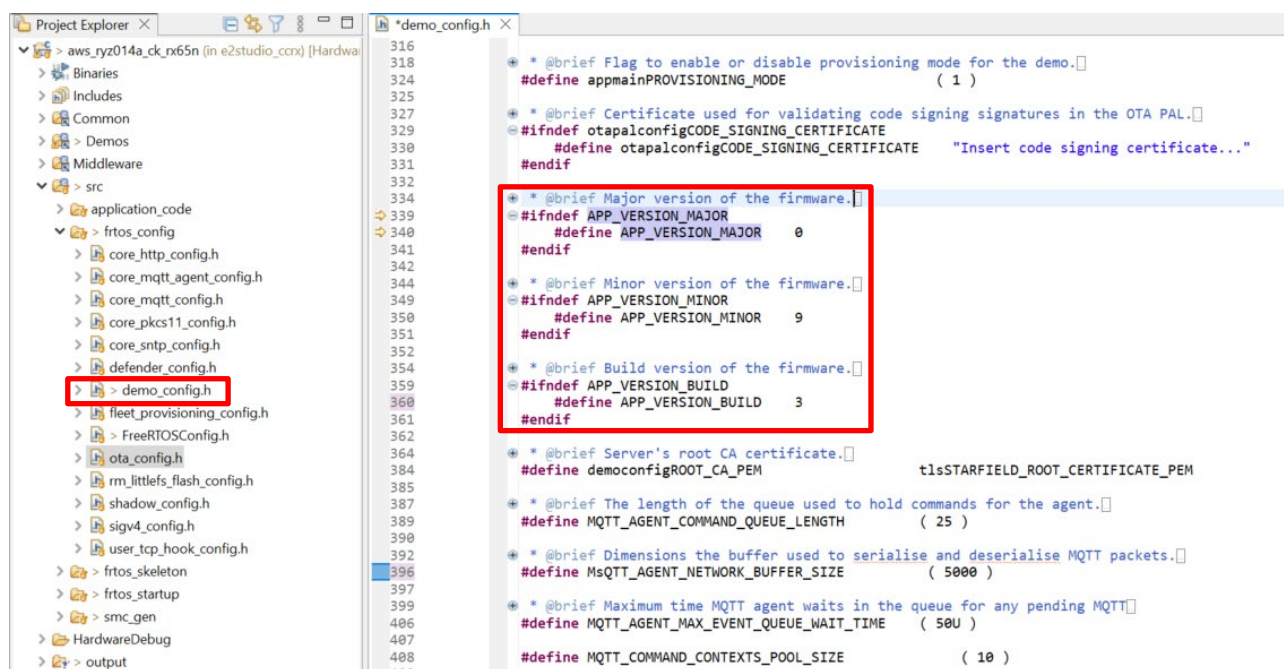

5. ファームウェアの更新

5.1 更新用ファームウェア構築

5.1.1 バージョンの変更

(1) ファームウェアのバージョンを v0.9.3 に変更する

aws_ryz014a_ck_rx65n¥src¥rtos_config¥demo_config.h の APP_VERSION_BUILD 定義を 3 にしてビルドを再実行する。



(2) Renesas Image Generator を使用して更新ファームウェアを生成

5.1.1(1)で再ビルドしたファームウェア(aws_ryz014a_ck_rx65n.mot)を Renesas Image Generator フォルダに上書きし、コマンドプロンプトで以下コマンドを実行します。

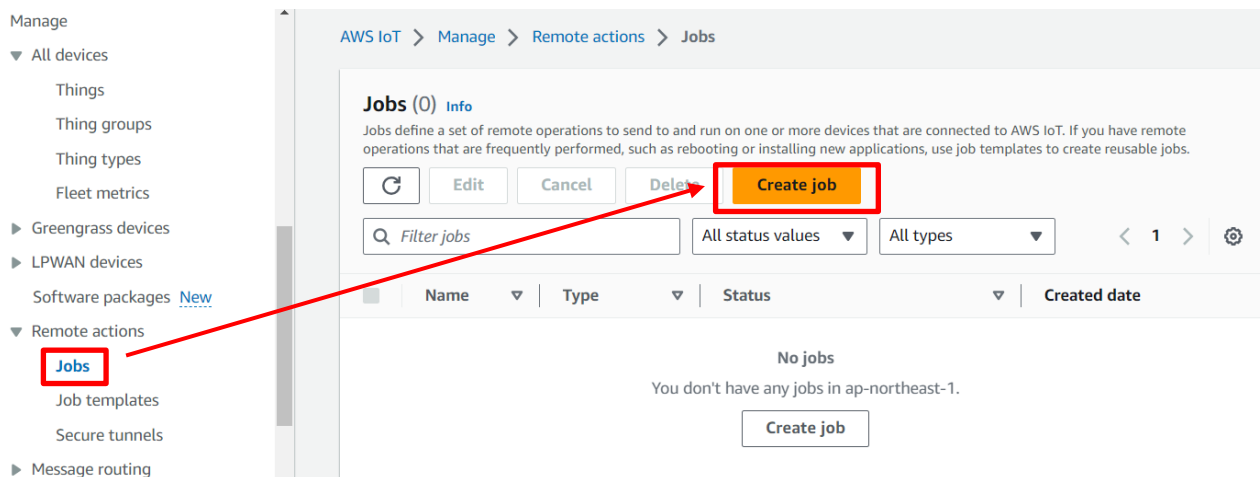
```
python image-gen.py -iup aws_ryz014a_ck_rx65n.mot -ip RX65N_DualBank_ImageGenerator_PRM.csv -o user_093 -key secp256r1.privatekey -vt ecdsa -ff RTOS
```

上記コマンドで、user_093.rsu ファイルが生成されます。

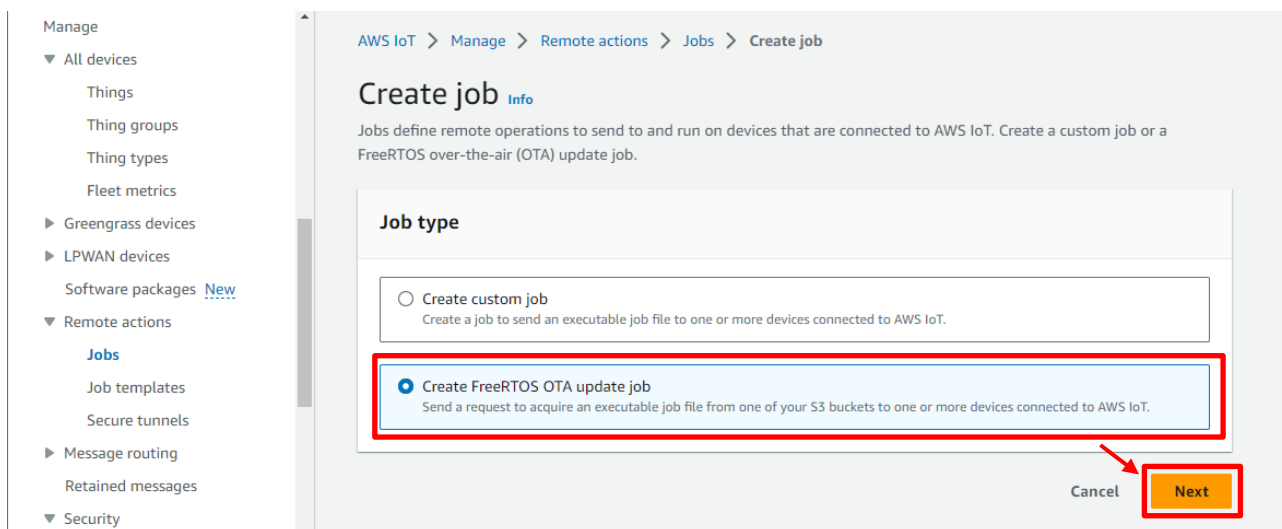
5.2 ファームウェアの更新

AWS にて、ファームウェアの更新を行うための OTA 更新ジョブを作成します。

- (1) IoT Core のメニューから Manage(管理) > Remote actions(リモートアクション) > Jobs(ジョブ)から Create job(ジョブを作成)を選択



- (2) Create FreeRTOS OTA update job (FreeRTOS OTA 更新ジョブを作成)を選択して次へ



RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(3) ジョブ名を入力(例 : rx65n_ota_demo_job)して次へ

AWS IoT > Jobs > Create job > OTA job

Step 1
OTA job properties

Step 2
OTA file configuration

Step 3
OTA job configuration

OTA job properties [Info](#)

Job properties

Job name

rx65n_ota_demo_job

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

Description - optional

Enter job description

► **Tags - optional**

Cancel **Next**

(4) Devices to update(更新するデバイス)をクリックして、更新するデバイスを選択

OTA file configuration [Info](#)

Devices [Info](#)

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

Devices to update

Choose things and/or thing groups

Q

Thing groups

Things

☒ rx65n_ota_demo_thing

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(5) Create new profile(新しいプロファイルの作成)を選択

File Info

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

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

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

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

過去にプロファイルを作成していた場合は、(5)~(9)は省略できます。Choose existing code signing profile (既存のコード署名プロファイル)で過去に作成したプロファイルを選択してください。

Existing code signing profile

Choose existing code signing profile ▲

Q

rx65n_ota_demo_profile2	/dummy
SHA256	ECDSA
rx65n_ota_demo_profile	dummy
SHA256	ECDSA

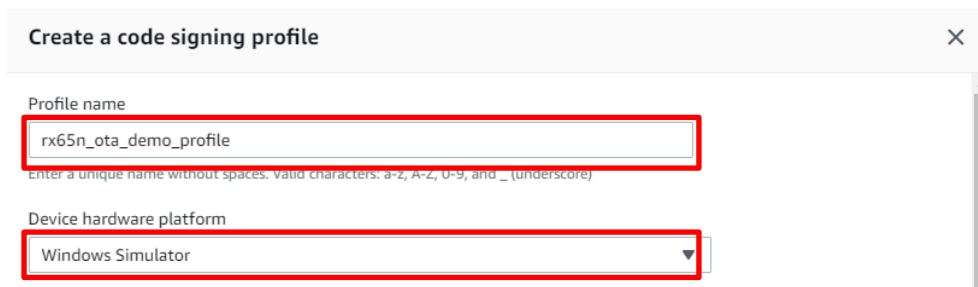
Create new profile

Select an existing file.

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(6) プロファイル作成(1) プロファイル名、デバイスハードウェアプラットフォーム

- プロファイル名を入力(例: rx65n_ota_demo_profile)
- デバイスハードウェアプラットフォームに「Windows Simulator」を選択



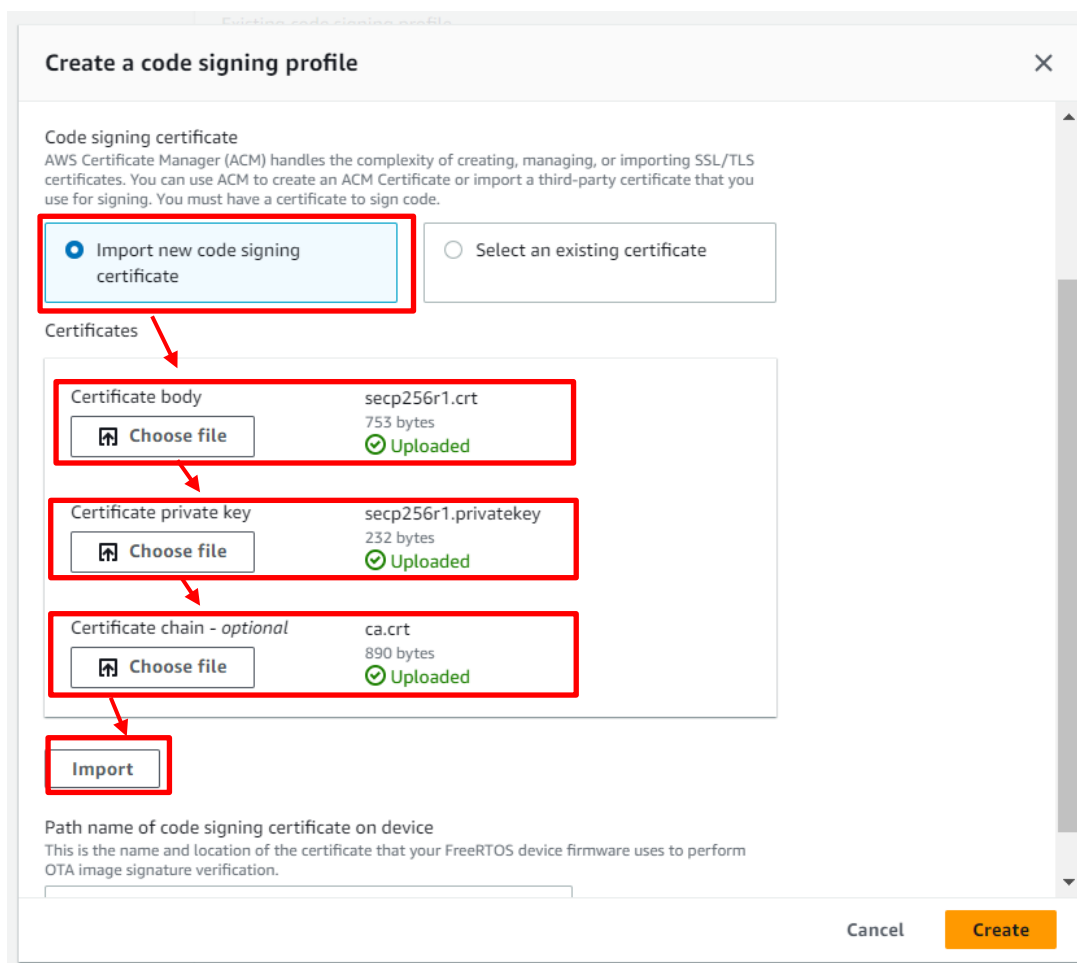
Create a code signing profile

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

Device hardware platform
Windows Simulator

(7) プロファイル作成(2) 証明書のインポート

- コード署名証明書で新しい署名証明書のインポートをクリック
- Certificate body(証明書本文)に 4.1(6) で作成した secp256r1.crt を選択
- Certificate private key(証明書のプライベートキー)に 4.1 で作成した secp256r1.privatekey を選択
- Certificate chain(証明書チェーン)に 4.1(3)で作成した ca.crt を選択
- Import(インポート)をクリック



Create a code signing profile

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

☒ Import new code signing certificate ☐ Select an existing certificate

Certificates

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

Import

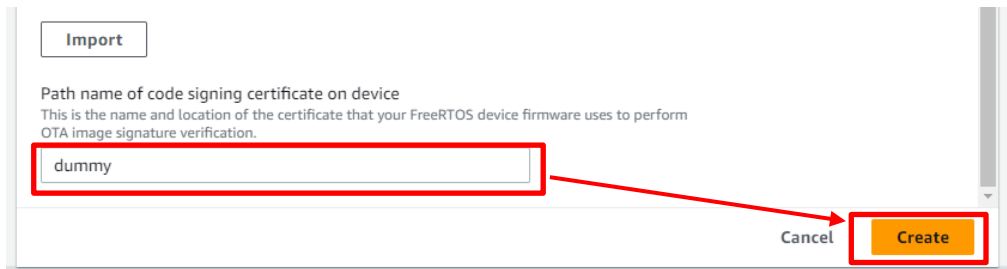
Path name of code signing certificate on device
This is the name and location of the certificate that your FreeRTOS device firmware uses to perform OTA image signature verification.

Cancel Create

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(8) プロファイル作成(3) デバイスのコード署名証明書のパス名を入力して Create をクリック

パス名は任意です。(例 : dummy)



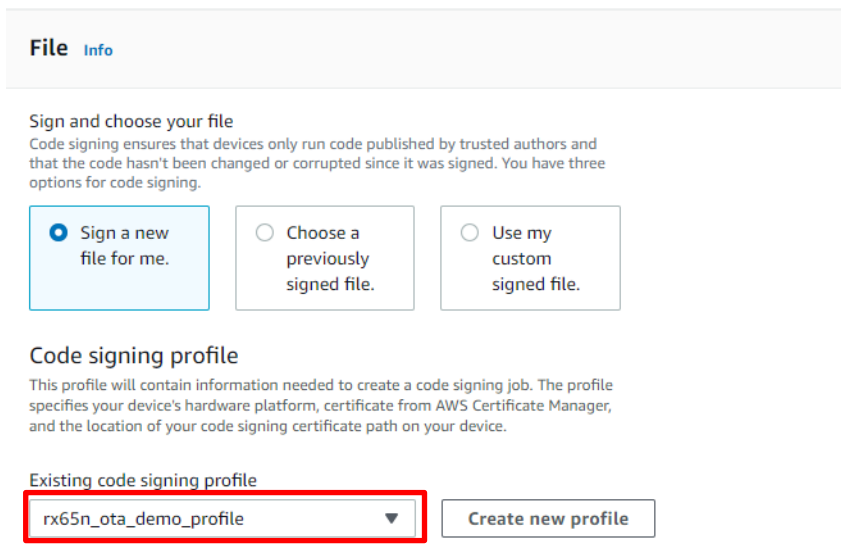
Import

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

dummy

Cancel Create

(9) プロファイルに先ほど作成したプロファイル名が選択されているのを確認



File Info

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

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

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

Existing code signing profile
rx65n_ota_demo_profile ▼

Create new profile

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(10) 更新用ファームウェアのアップデート

- Upload a new file (新しいファイルをアップロードする)を選択
- File to upload (アップロードするファイル)に 5.1.1(2)で作成した usr093.rsu を選択
- Browse S3(S3 をブラウズ)に 3.4 で作成した S3 バケットを選択
- Path name of file on device(デバイス上のファイルのパス名)を入力
(パス名は任意「例: /device/updates」)

File

☒ Upload a new file. ☐ Select an existing file.

File to upload

usr093.rsu
357888 bytes

File upload location in S3

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

S3 URL

s3://s3test-rx65n

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

Path name of file on device

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

/device/updates

► File type - optional

(11) Role (ロール)に 3.5(5)で作成したロールを選択して次へ

IAM role Info

Role

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

ota_role_rx65n ▼

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

(12) Create job (ジョブの作成)をクリック

OTA job configuration [Info](#)

Job run type
Choose how to run this job.

☒ Your job will complete after deploying to the devices and groups that you chose (snapshot)

☐ Your job will continue to deploy to any devices added to the groups that you chose (continuous)

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

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

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

Cancel Back **Create job**

(13) ファームウェアの受信が完了するまで待つ

ジョブが開始されると、受信およびファームウェアの書き込みを行います。

```
COM3 - Tera Term VT
File Edit Setup Control Window KanjiCode Help

4049 186594 [cellular_re] [DEBUG] clear buff =
OK

4050 186663 [OTA Demo Ta] [INFO] Received: 11 Queued: 11 Processed: 11 Dr
opped: 0

4051 187230 [cellular_re] [DEBUG] URC =
+SQNSRING: 1,1024

4052 187231 [MQTT] [DEBUG] generated AT command: AT+SQNSRECV=1,5

4053 187231 [MQTT] [DEBUG] RTS out

4054 187236 [cellular_re] [DEBUG] URC =
+SQNSRECV: 1,5

4055 187238 [cellular_re] [DEBUG] received AT command response:
OK

4056 187238 [cellular_re] [DEBUG] clear buff =
OK
```

受信を開始すると、Received の回数が加算されます

RX ファミリ RX65N における Amazon Web Services を利用した FreeRTOS OTA の実現方法(v202210.01-LTS-rx-1.1.0 以降対応版)

更新が完了すると、リセットがかかり最初のメニューが表示されます。

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

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

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

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

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

(14) ファームウェアのバージョンが Ver0.9.3 になっていることを確認

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

6. トラブルシューティング

本サンプルを実行する際に想定されうるトラブルおよびその解決策を下表に示します。

表 6-1 トラブルシューティング

No	トラブル内容	原因	解決策	参照
1	初期ファームウェアの作成コマンドが失敗する	Python のパスが通っていない	Python を再インストールしてください。また、2.2(3)の手順で、「Add python.exe to PATH」にチェックが入っていることを確認してください。	2.2
2		暗号化ライブラリがインストールされていない	暗号化ライブラリをインストールしてください。	2.2(5)
3	初期ファームウェアが書き込めない	CK-RX65N がデバッグ設定になっていない	CK-RX65N の J16 の設定が 1-2 ショート（デバッグ）になっていることを確認してください。	2.5
4	初期ファームウェアが起動しない	CK-RX65N が RUN 設定になっていない	CK-RX65N の J16 の設定が 2-3 ショート（RUN）になっていることを確認してください。	4.2.5(4)
5	セルラー通信が開始できない	RYZ014A PMOD が正しく接続されていない	RYZ014A PMOD の接続を見直してください。	2.5
6		SIM カードが挿入されていない	SIM カードを挿入してください。	2.5
7		SIM カードの設定が正しくされていない	r_cellular のコンフィグ設定を見直してください。	4.2.3(4)
8		CK-RX65N 付属の SIM カードを使用しており、かつ SIM カードのアクティベーションができていない	SIM カードのアクティベーションを行ってください。	4.2.3(4)
9	セルラー通信中にエラーが発生する	通信環境が悪い	RYZ014A PMOD にアンテナおよび電源の接続を行ってください。また、アンテナを窓際など通信環境の良い場所においてください。	2.5
10	AWS への接続でエラーが発生する	AWS IoT 情報が設定されていない、または間違えている	再度、AWS IoT 情報の設定を行ってください。	4.2.5
11	ブートローダ起動後にファームウェアが起動しない	ブートローダに公開鍵が正しく設定がされていない	ブートローダの公開鍵設定を見直してください。	4.2.3(1)
12	OTA アップデート後にファームウェアが起動しない	ファームウェアに公開鍵が正しく設定されていない	ファームウェアの公開鍵設定を見直してください。	4.2.3(1)
13		デバイス選択が正しく設定されていない	ファームウェアおよびブートローダのデバイス設定を見直してください。	4.2.3(5) ~ 4.2.3(9)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Sep.15.23	-	初版発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものとしします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。