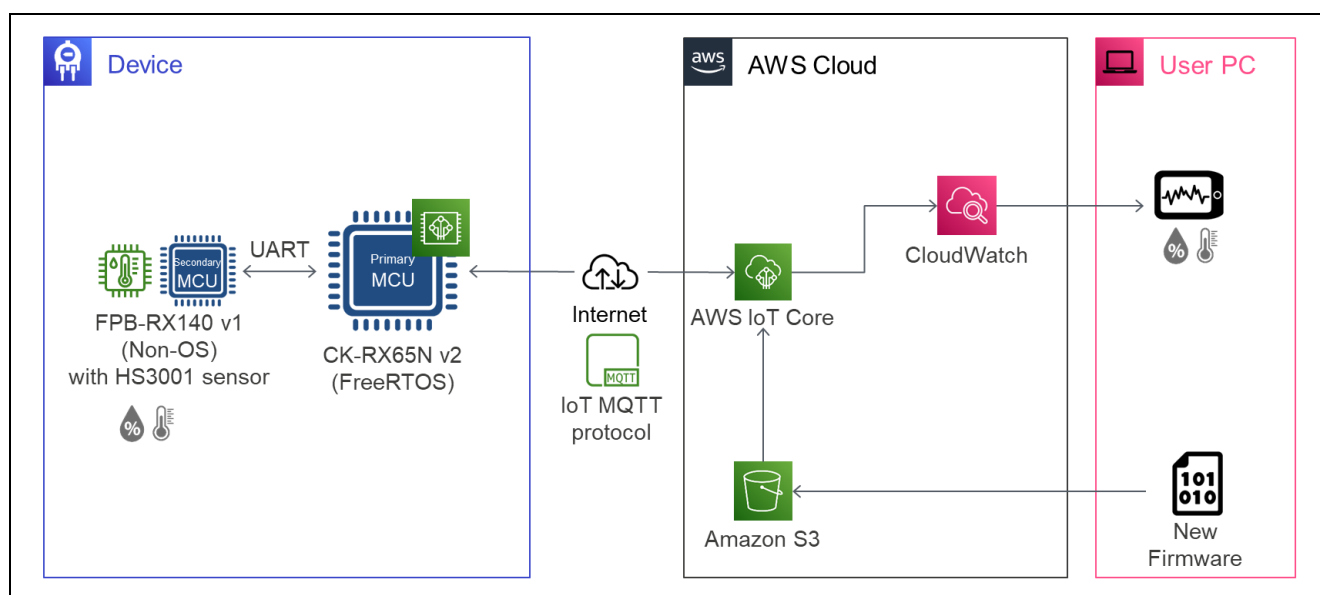


RX65N グループ

FreeRTOS を用いた Amazon Web Services によるセカンダリデバイスの OTA アップデートサンプルコード

要旨

本アプリケーションノートでは、Amazon Web Services™（以下、AWS と略します）との通信を行うプライマリ MCU の RX65N を用いて、センサデータの測定機能を持ったセカンダリ MCU の RX マイコンに対して、AWS のサービスを利用した OTA アップデート（以下、セカンダリ OTA アップデートと略します）を行うデモについて説明します。



動作確認デバイス

プライマリ MCU : RX65N
 セカンダリ MCU : RX140
 温湿度センサ : HS3001 高性能 相対湿度・温度センサ

動作確認ボード

プライマリ MCU: CK-RX65N v2 (RTK5CK65N0S08001BE)
 セカンダリ MCU : FPB-RX140 v1 (RTK5FP1400S00001BE)
 温湿度センサ : 温湿度センサ Pmod™ ボード (US082-HS3001EVZ)

関連文書

本アプリケーションノートは、以下の文書を参照し、説明しています。文書更新の場合に章構成等が変わる場合があります。参照の際に注意してください。

[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 \(R01AN6850\)](#)

[RX ファミリ ファームウェアアップデート通信モジュール Firmware Integration Technology \(R01AN7757\)](#)

[RX ファミリ AWS/Azure を利用したファームウェア更新ソフトの開発ガイド QE for OTA \(R20AN0712\)](#)

[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\(v202210.01-LTS-rx-1.1.0 以降対応版\) \(R01AN7037\)](#)

[RX65N Group CK-RX65N v2 User's Manual \(R20UT5366\)](#)

[RX140 グループ FPB-RX140 v1 ユーザーズマニュアル \(R20UT5376\)](#)

(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート／テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

Amazon Web Services、“Powered by AWS”ロゴ、およびかかる資料で使用されるその他の AWS 商標は、米国および／またはその他の国における Amazon com, Inc. またはその関連会社の商標です。

FreeRTOS™ および FreeRTOS.org™ は Amazon Web Services, Inc. の商標です。

Pmod は Digilent Inc. 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。

目次

1. 概要	4
2. 動作確認条件	4
3. ハードウェア説明	5
4. ソフトウェア説明	6
4.1 クラウド接続、OTA 動作の制御について	6
4.2 ファームウェアアップデート方式	6
4.3 マイコン間 UART 通信	8
4.3.1 UART 通信設定	8
4.4 フォルダ/ファイル構成	9
4.5 コードサイズ	10
5. デモの動作説明	11
6. デモのセットアップ	11
6.1 ハードウェアのセットアップ	11
6.1.1 CK-RX65N のセットアップ方法	12
6.1.2 FPB-RX140 のセットアップ方法	14
6.2 ソフトウェアのセットアップ	18
6.2.1 事前準備	18
6.2.2 QE for OTA で AWS にログイン	20
6.2.3 CK-RX65N 用の初期ファームウェアの作成と実行	21
6.2.4 FPB-RX140 用の初期ファームウェアの作成と実行	27
6.3 AWS クラウド上でセンサデータを可視化するための準備	32
7. デモの実行手順	40
7.1 初期状態の動作確認	40
7.2 FPB-RX140 の OTA アップデートの実行	42
7.2.1 更新ファームウェアの作成	42
7.2.2 OTA ジョブの作成	44
7.2.3 セカンダリ OTA アップデート実行中の動作確認	45
7.3 OTA アップデート後の動作確認	46
8. センサを使用せずデモを実施する方法	48
8.1 デモ実施手順の変更箇所	48
8.2 センサを使用しない場合のデモ動作の確認方法	48
9. 注意事項	49
9.1 使用するオープンソースソフトウェアのライセンス情報	49
9.2 デモを実施する AWS のリージョンとユーザ権限について	49
9.3 AWS の利用料金について	49
改訂記録	50

1. 概要

デモでは、セカンダリ OTA アップデートによって稼働センサを追加し、ブラウザ上の AWS 画面でセンサデータを可視化することで取得するセンサデータの追加を確認できます。

IoT 機器にはセキュリティの脆弱性の適宜修正や、お客様要望に応じた機能のアップデートが求められます。従来、提供しているプライマリ MCU の OTA アップデートだけでなく、セカンダリ OTA アップデートを実現することで、セカンダリ MCU の脆弱性への対応やフレキシブルなサービスのアップデートが可能な製品開発を実現できます。

2. 動作確認条件

本アプリケーションノートのサンプルプログラムは以下に示す条件でデモの動作を確認しています。

表 2-1 デモ動作確認条件 (RX65N)

項目	内容
使用マイコン	RX65N (R5F565NEHDFB)
使用ボード	CK-RX65N v2 (RTK5CK65N0S08001BE)
動作電圧	3.3V
RTOS	FreeRTOS v202210.01-LTS-1.3.1
統合開発環境 (IDE)	e² studio 2025-04 QE for OTA v2.2.0
C コンパイラ	ルネサス製 RX ファミリー用 C/C++コンパイラ CC-RX V3.07.00 GCC for Renesas RX 8.3.0.202411
フラッシュ書き込みツール	Renesas Flash Programmer V3.19.00

表 2-2 デモ動作確認条件 (RX140)

項目	内容
使用マイコン	RX140 (R5F51406BGFN)
使用ボード	FPB-RX140 v1 (RTK5FP1400S00001BE)
動作電圧	3.3V
統合開発環境 (IDE)	e² studio 2025-04 QE for OTA v2.2.0
C コンパイラ	ルネサス製 RX ファミリー用 C/C++コンパイラ CC-RX V3.07.00 GCC for Renesas RX 8.3.0.202411
フラッシュ書き込みツール	Renesas Flash Programmer V3.19.00
USB-UART 変換ボード	Pmod USBUART™

表 2-3 デモ動作確認条件 (センサ)

項目	内容
温湿度センサボード	US082-HS3001EVZ Board
変換ボード	US082-INTERPEVZ

※ HS3001 センサは 2025/9/30 をもちまして EOL となります。代替品およびその他の詳細情報は以下の EOL 通知資料をご確認ください。本アプリケーションノートのデモはセンサなしでも実施可能です。センサを使用しない実施方法を「8 センサを使用せずデモを実施する方法」で説明します。
<https://www.renesas.com/document/elc/plc-250010-end-life-eol-process-select-part-numbers>

表 2-4 デモ動作確認条件 (その他)

項目	バージョン
Python	3.12.6

QE for OTA は、<https://www.renesas.com/qe-ota/> から入手できます。

Python は、<https://www.python.org/> から入手できます。

3. ハードウェア説明

本システムは、AWS との通信制御機能を持つ RX65N マイコン(プライマリ MCU)と、HS3001 センサが接続された RX140 マイコン(セカンダリ MCU)で構成されています。2つのマイコンは UART による相互通信を行います。

図 3-1 にシステム構成を示します。

プライマリ MCU として、RX65N を搭載した CK-RX65N v2 (以降、CK-RX65N)を使用します。

セカンダリ MCU として、RX140 を搭載した FPB-RX140 v1 (以降、FPB-RX140)を使用します。

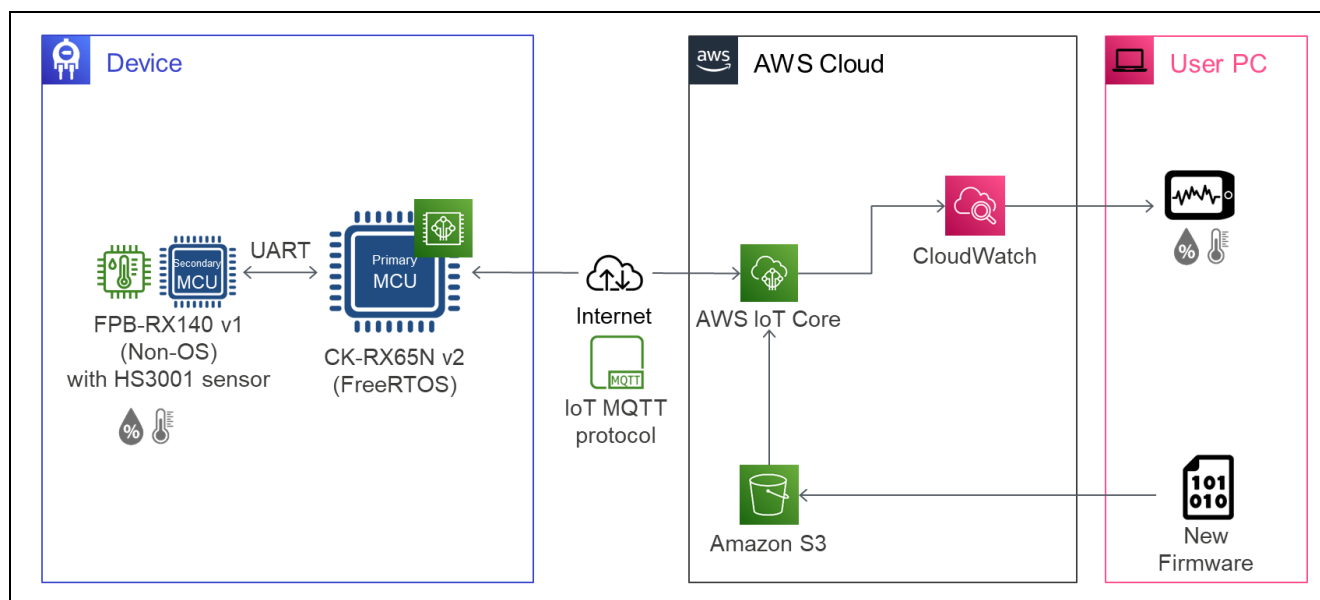


図 3-1 本デモのシステム構成

4. ソフトウェア説明

4.1 クラウド接続、OTA 動作の制御について

AWS 認定済プログラムを使用した RX65N のファームウェアには、FreeRTOS™ with IoT Library が実装されています。そのため、AWS が提供しているマネージドサービスである AWS IoT Core および AWS IoT Device Management を利用して、OTA によるファームウェアアップデートや MQTT 通信によるクラウドへのデータのアップロードが実行可能です。

セカンダリ OTA アップデートの制御には、プライマリ MCU 側の RX65N マイコンでは AWS IoT Over-the-air Update Library を利用し、AWS から受信したセカンダリ MCU 用の更新ファームウェアをセカンダリ MCU に転送し、ファームウェア更新を実現しています。

プライマリ MCU とセカンダリ MCU 間のデータ通信には「[RX ファミリ ファームウェアアップデート通信モジュール Firmware Integration Technology](#)」を使用します。

セカンダリ MCU 側の RX マイコンのファームウェアアップデートの制御は「[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](#)」を使用します。

4.2 ファームウェアアップデート方式

本サンプルプログラムのセカンダリ MCU のファームウェア更新の仕組みは、ファームウェアアップデートモジュールが提供している方式のうち、「リニアモードの半面更新方式」を使用しています。この方式の詳細は、「[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](#)」の「1.3 各ファームウェアアップデート方式について」の「リニアモードの半面更新方式」をご参照ください。

以下の図 4-1 にセカンダリ OTA アップデートの動作概要を示します。また、図 4-2 にアップデートの実行中の各フェーズでの ROM の状態を示します。なお、図 4-2 の赤枠はその時実行されているプログラムを表しています。

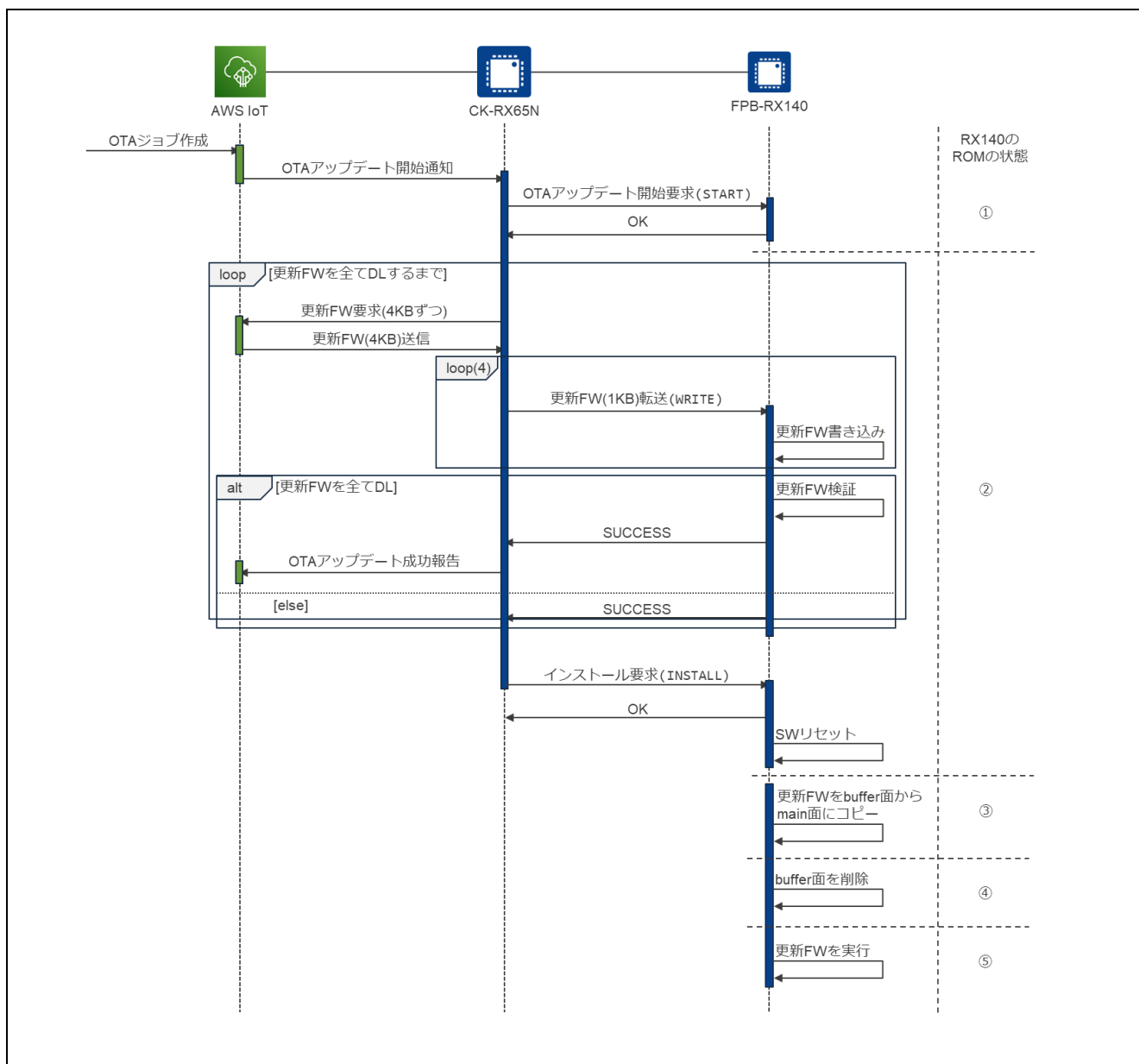


図 4-1 セカンダリ OTA アップデートの動作概要

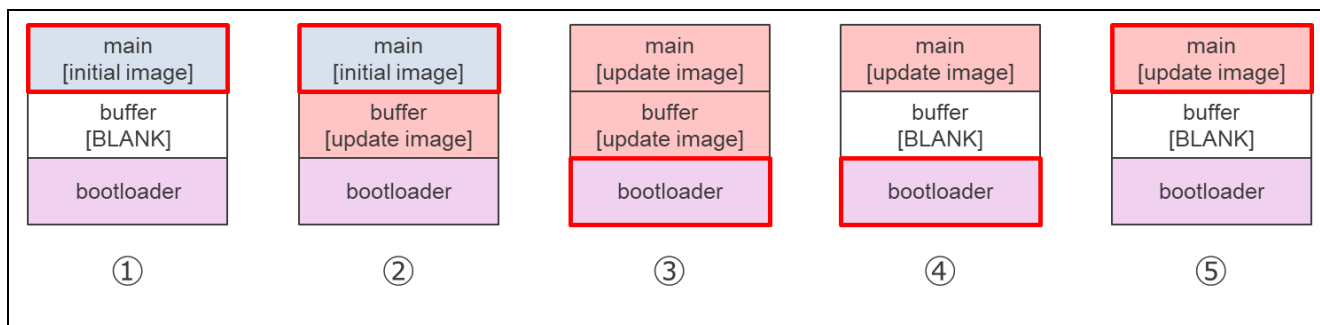


図 4-2 アップデート実行中のセカンダリ MCU の ROM の状態

4.3 マイコン間 UART 通信

プライマリ MCU とセカンダリ MCU 間の UART 通信は、「[RX ファミリ ファームウェアアップデート通信モジュール Firmware Integration Technology](#)」を使用して実現しています。

セカンダリ OTA アップデートに関連する通信(図 4-1 のシーケンス図の CK-RX65N と FPB-RX140 間の通信)には、FWUP コマンド群を使用しています。

また、センサデータの送受信には、Common コマンド群の DATA_RECV コマンドを使用し、Command argument が 1 の場合を、センサデータの送信要求として用いています。

各コマンドの詳細は、「[RX ファミリ ファームウェアアップデート通信モジュール Firmware Integration Technology](#)」の「1.6 コマンド仕様」をご参照ください。

4.3.1 UART 通信設定

マイコン間 UART 通信設定を表 4-1 に示します。

表 4-1 ファームウェアアップデート通信モジュールの UART 通信設定

項目	内容
Data Length	8-bit
Parity	None
Stop Bits	1-bit
Flow Control	None
Bitrate	1Mbps

4.4 フォルダ/ファイル構成

図 4-3 にフォルダ/ファイル構成を示します。

```
r01an6220xx0300-rx-2nd-ota-apl
├─ Demo
│   ├── afr-v202210.01-LTS-rx-1.3.1
│   ├── ccrx
│   │   ├── demo_rx65n_ck_primary
│   │   ├── demo_bl_rx65n_ck_primary
│   │   ├── demo_app_rx140_fpb_w_buffer
│   │   └── bootloader_rx140_fpb_w_buffer
│   └── gcc
│       ├── demo_rx65n_ck_primary
│       ├── demo_bl_rx65n_ck_primary
│       ├── demo_app_rx140_fpb_w_buffer
│       └── bootloader_rx140_fpb_w_buffer
├─ r01an6220ej0300-rx-2nd-ota-apl.pdf
└─ r01an6220jj0300-rx-2nd-ota-apl.pdf
```

図 4-3 フォルダ/ファイル構成

demo_rx65n_ck_primary フォルダと demo_bl_rx65n_ck_primary フォルダには CK-RX65N 用のプロジェクトファイルが格納されています。

demo_app_rx140_fpb_w_buffer フォルダと bootloader_rx140_fpb_w_buffer フォルダには、FPB-RX140 用のプロジェクトファイルが格納されています。

4.5 コードサイズ

本アプリケーションのサンプルコードに含まれるプロジェクトの ROM, RAM サイズを下表に示します。
下表の値は以下の条件で確認しています。

CC-RX

Compiler

最適化レベル (-optimize): Level 2: Performs whole module optimization

最適化タイプ (-speed/-size): Optimizes with emphasis on code size

Linker

最適化タイプ (-nooptimize/-optimize): All

Library Generator

最適化レベル (-optimize): Level 2: Performs whole module optimization

最適化タイプ (-speed/-size): Optimizes with emphasis on code size

表 4-2 コードサイズ (CC-RX)

プロジェクト	ROM	RAM
demo_bl_rx65n_ck_primary	33 KB	8 KB
demo_rx65n_ck_primary	601 KB	382 KB
bootloader_rx140_fpb_w_buffer	29 KB	6 KB
demo_app_rx140_fpb_w_buffer	50 KB	13 KB

GCC

最適化レベル: Optimize for debug (-Og)

表 4-3 コードサイズ (GCC)

プロジェクト	ROM	RAM
demo_bl_rx65n_ck_primary	51 KB	10 KB
demo_rx65n_ck_primary	654 KB	383 KB
bootloader_rx140_fpb_w_buffer	24 KB	13 KB
demo_app_rx140_fpb_w_buffer	41 KB	12 KB

5. デモの動作説明

- (1) デモの初期状態では、FPB-RX140 は接続されている HS3001 センサを使って湿度データのみ取得します。
- (2) セカンダリ OTA アップデートの仕組みを用いて、AWS から CK-RX65N 経由で FPB-RX140 の更新ファームウェアをダウンロードし、ファームウェアの更新を行います。
- (3) ファームウェア更新後は、FPB-RX140 は HS3001 センサから湿度データに加えて温度データも取得します。

一連の流れで、取得しているセンサデータの種類とその値は、両マイコンからの PC へのログ出力と AWS 上のダッシュボードから確認できます。

6. デモのセットアップ

本アプリケーションノートのデモを実行するために必要なセットアップについて説明します。

CK-RX65N と FPB-RX140 の配線や HS3001 センサの接続方法等のハードウェアのセットアップ、それぞれのマイコンボード用の初期ファームウェアの作成と書き込み等のソフトウェアのセットアップ、そして OTA アップデートの実行や AWS 上でのセンサデータの可視化のための AWS クラウド側の準備が必要です。

6.1 ハードウェアのセットアップ

最初に、本デモを構成するハードウェア全体の構成を示します。実際にセットアップ後の画像は図 6-2 を参照してください。以降でそれぞれのボードのセットアップ方法について詳しく説明します。

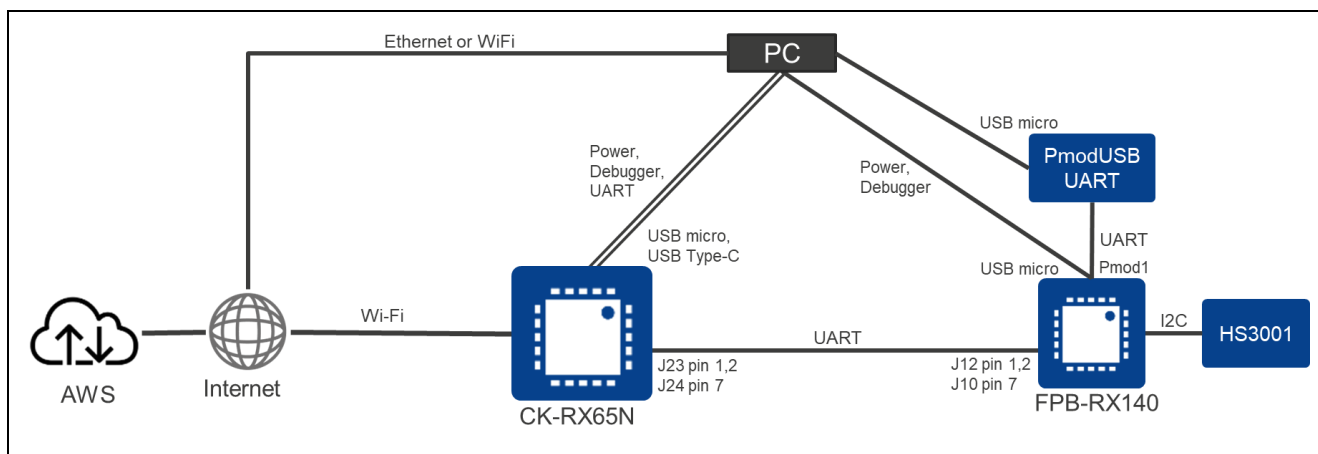


図 6-1 本デモのハードウェア全体構成

6.1.1 CK-RX65N のセットアップ方法

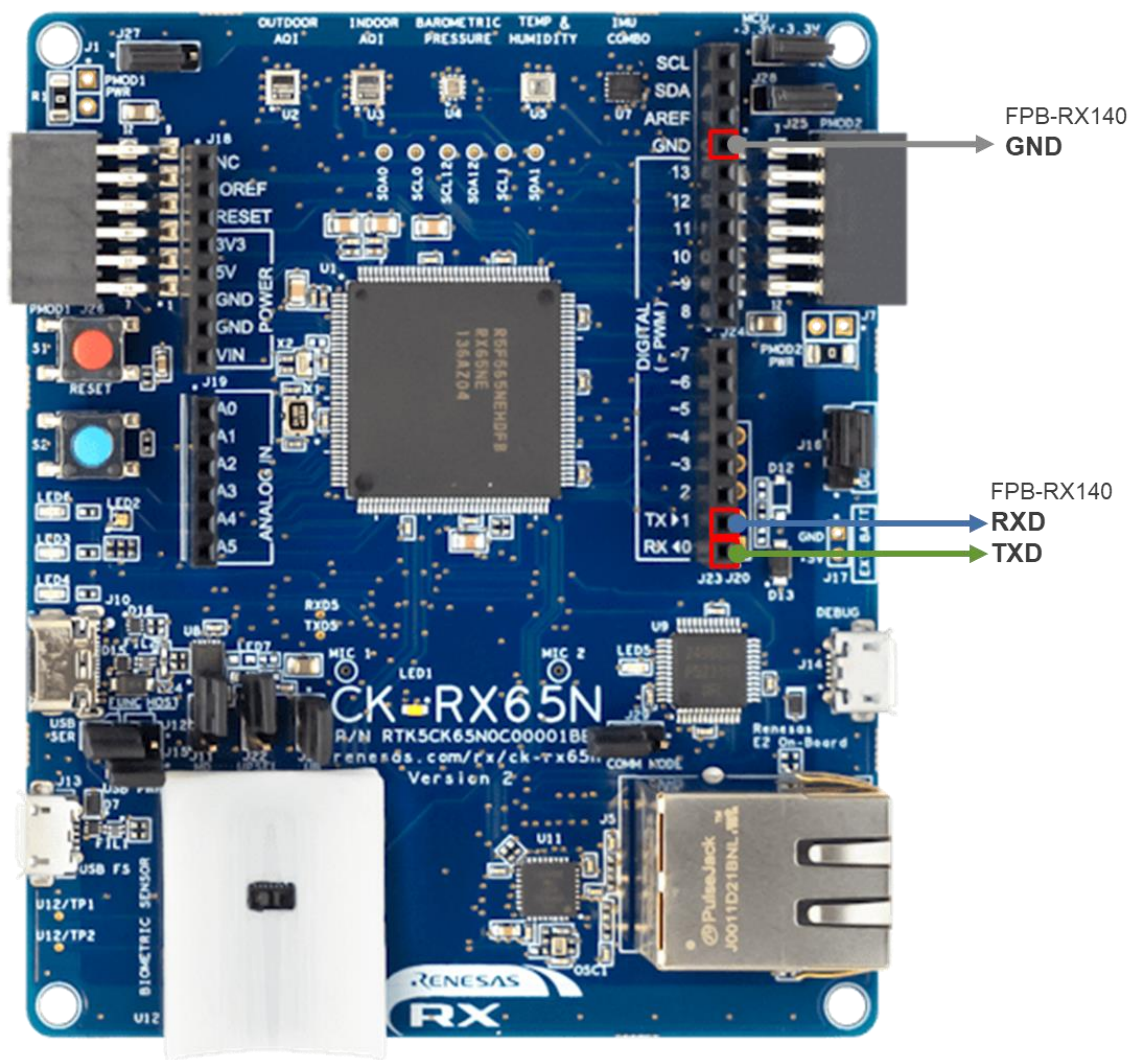
CK-RX65N のセットアップ方法を示します。

(1) FPB-RX140 との UART 通信用ケーブル接続

FPB-RX140 と UART 通信を行うための TXD, RXD, GND は CK-RX65N の J23, J24 コネクタの以下の端子に割り当てています。6.1.2(2)に示す FPB-RX140 側の端子と以下の表のように UART 信号の対応を取って接続してください。

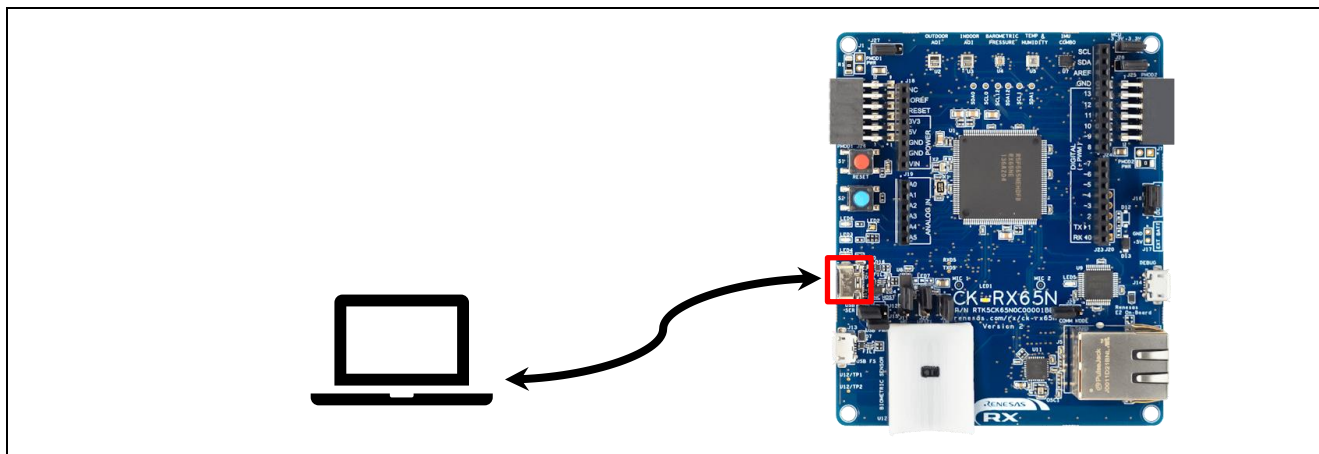
表 6-1 マイコン間 UART 接続方法 (CK-RX65N ⇄ FPB-RX140)

CK-RX65N		FPB-RX140
J23 Pin 1: D0/RX	⇄	J12 Pin 2: D1/TX
J23 Pin 2: D1/TX	⇄	J12 Pin 1: D0/RX
J24 Pin 7: GND	⇄	J10 Pin 7: GND



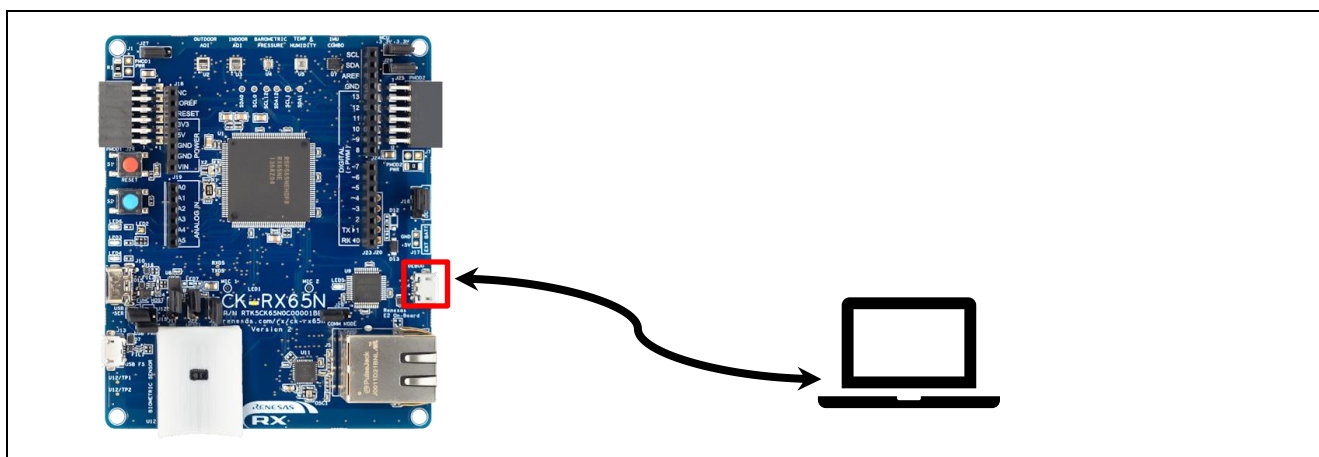
(2) PC へのログ出力用ケーブル接続

PC と CK-RX65N の USB シリアルコネクタ(USB Type-C)を USB ケーブルで接続します。



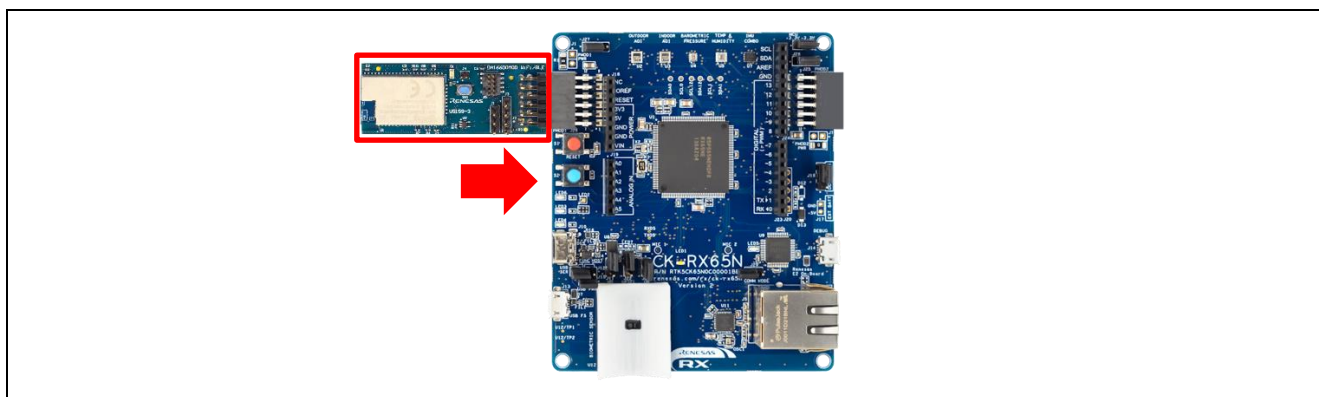
(3) 電源供給・デバッガとの接続

PC と CK-RX65N の E2OB Debugger コネクタ (micro USB Type-B)を USB ケーブルで接続します。



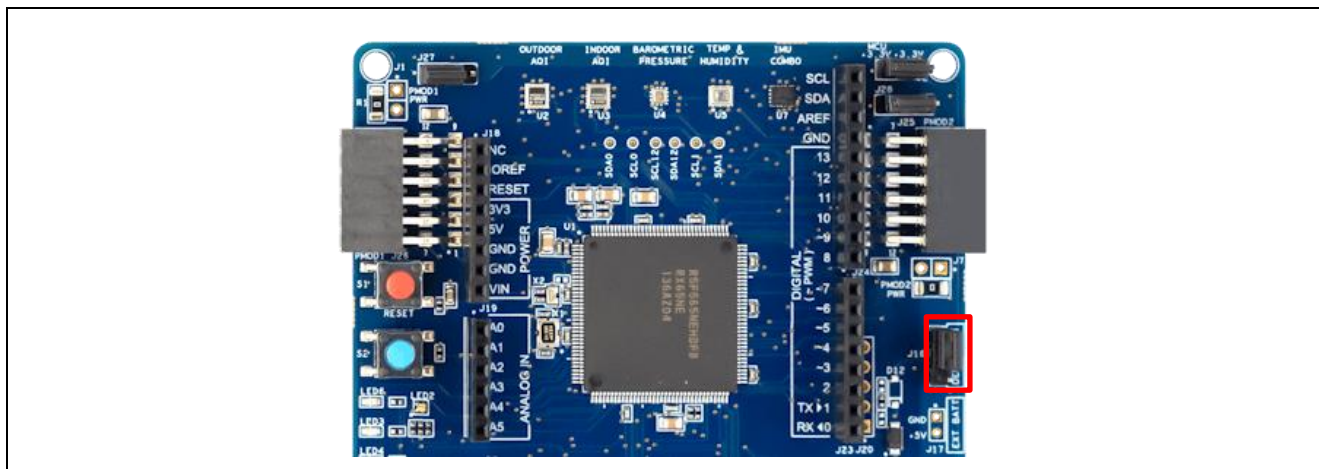
(4) DA16600 Wi-Fi モジュールの接続

CK-RX65N の Pmod1 コネクタに DA16600 Pmod モジュールを接続します。



(5) ジャンパ J16 を DEBUG 側に短絡する

CK-RX65N をデバッグモードにするためにジャンパ J16 を DEBUG 側(pin 1-2)に短絡します。

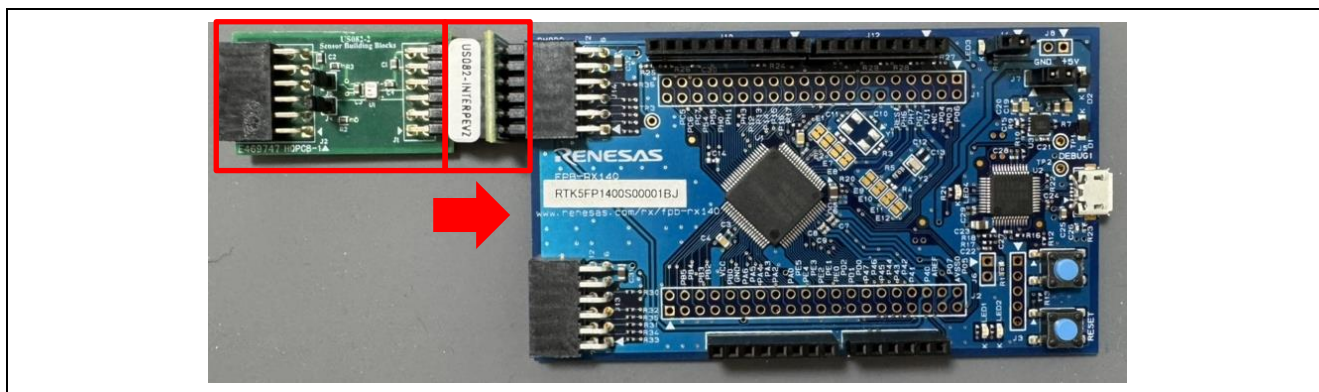


6.1.2 FPB-RX140 のセットアップ方法

FPB-RX140 のセットアップ方法を示します。

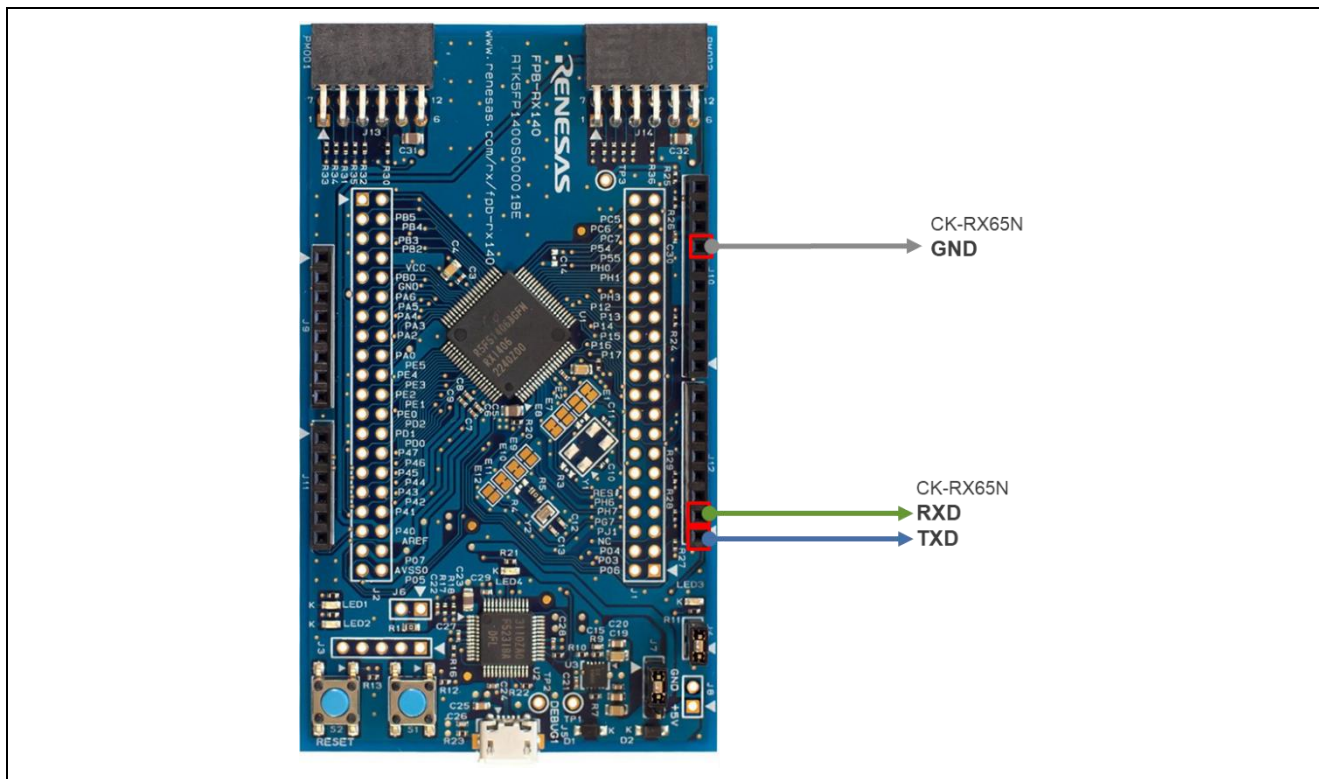
(1) HS3001 ボードの接続

FPB-RX140 の Pmod2 コネクタに HS3001 ボードと変換ボードを接続します。



(2) CK-RX65N との UART 通信用ケーブル接続

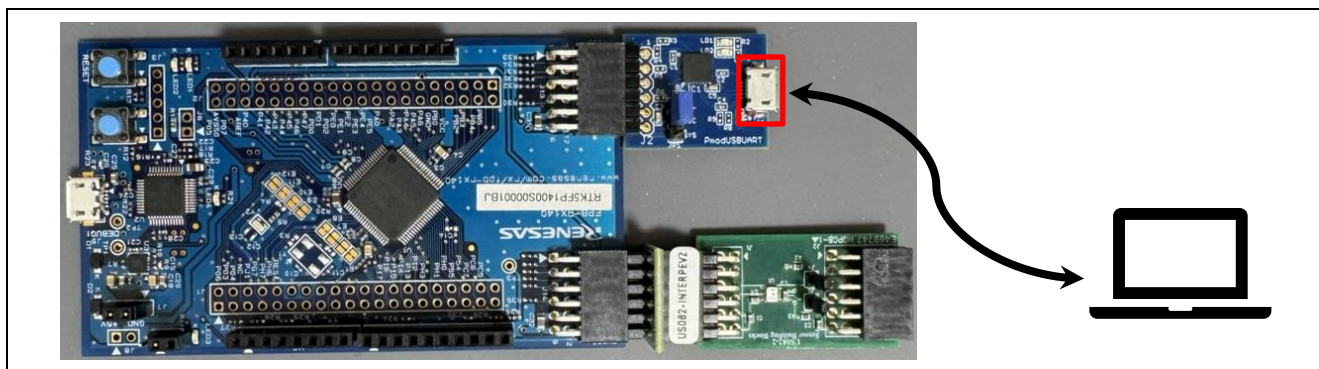
CK-RX65N と UART 通信を行うための TXD, RXD, GND は FPB-RX140 の J12, J10 コネクタの以下の端子に割り当てています。6.1.1(1)に示す CK-RX65N の端子と表 6-1 のように UART 信号の対応を取って接続してください。



(3) PC へのログ出力用ケーブル接続

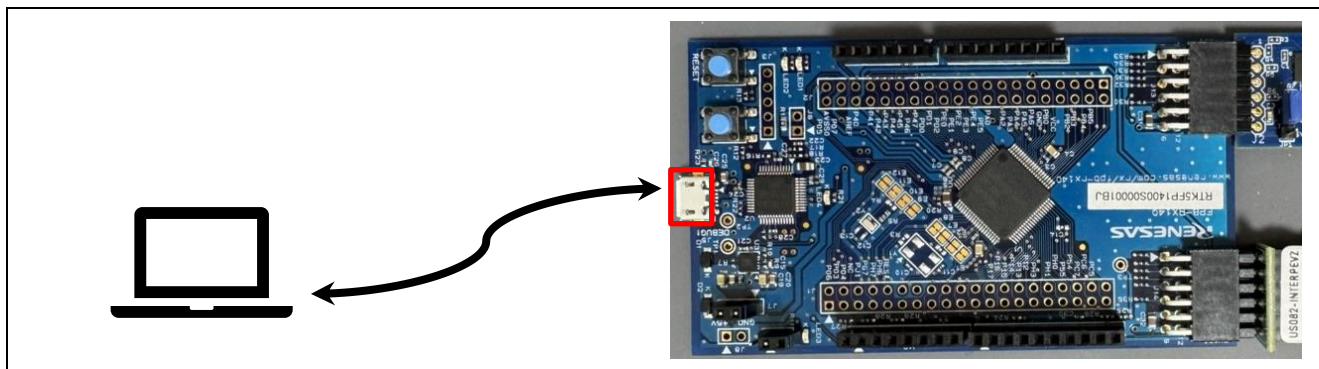
FPB-RX140 の Pmod1 コネクタの Pin1-6 に Pmod USBUART 変換ボードを接続します。

さらに、PC と Pmod USBUART 変換ボードの micro USB Type-B コネクタを USB ケーブルで接続します。



(4) 電源供給・デバッグとの接続

PC と FPB-RX140 の micro USB Type-B コネクタを USB ケーブルで接続します。



(5) エミュレータリセットヘッダ(J4)を開放する

FPB-RX140 のエミュレータリセットヘッダ(J4)を開放します。



以上でデモを実施するためのハードウェアのセットアップは完了です。図 6-2 にデモ構成の全体画像を示します。

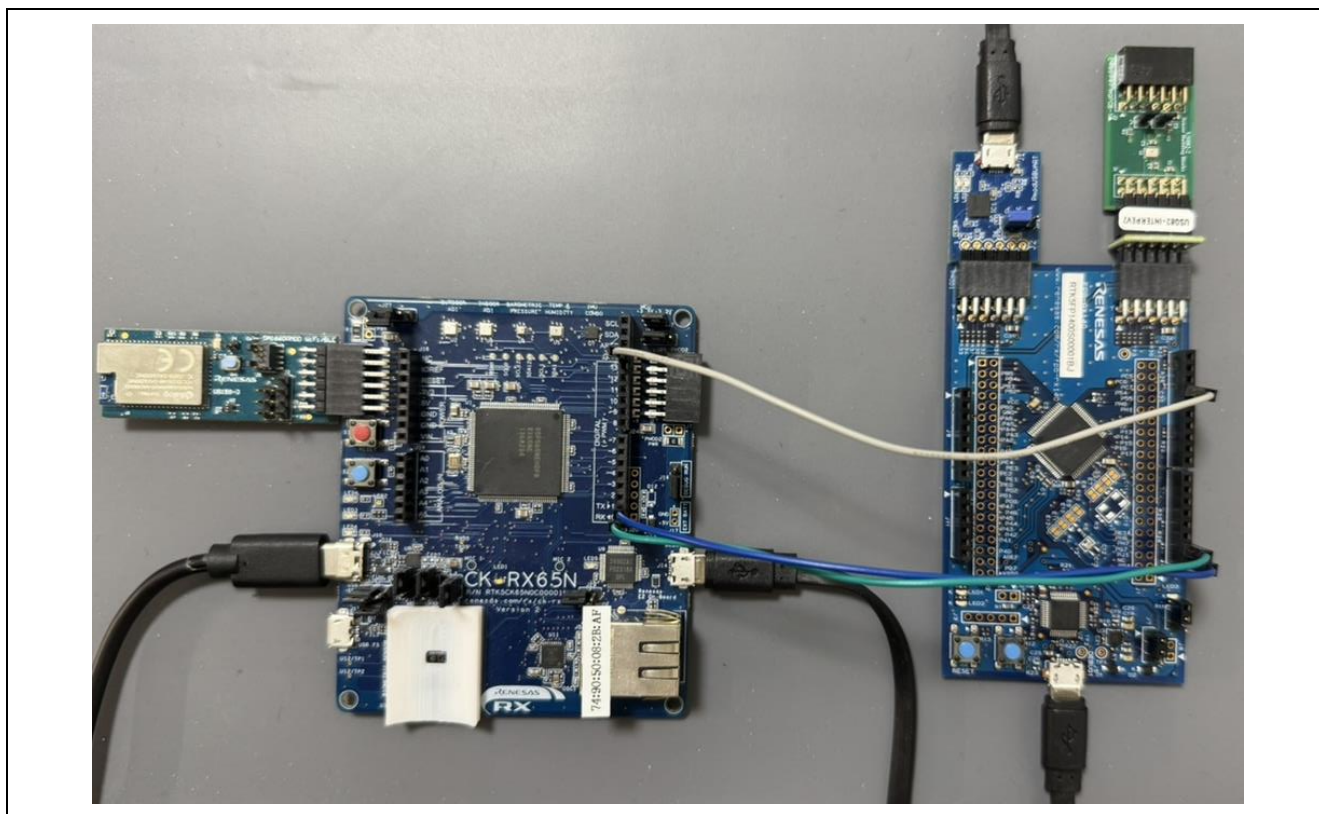


図 6-2 デモ構成の全体画像

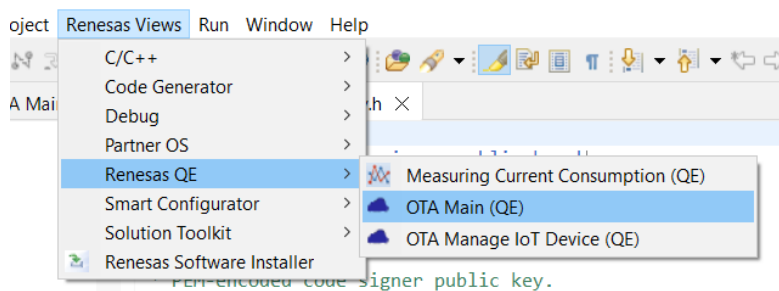
6.2 ソフトウェアのセットアップ

6.2.1 事前準備

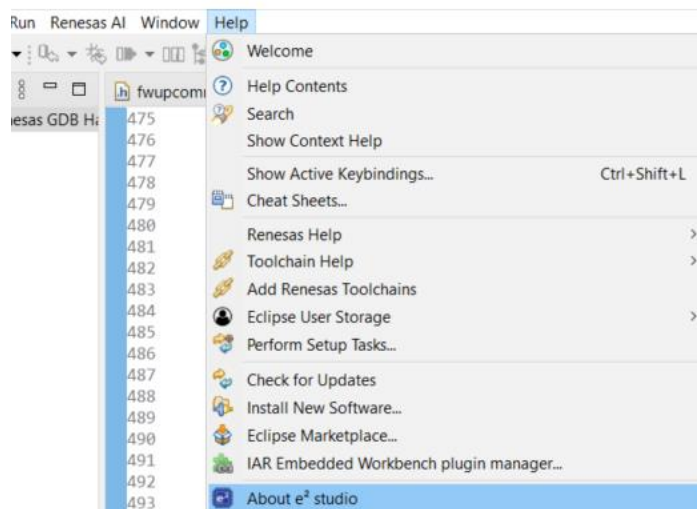
動作確認済みのそれぞれのソフトウェアのバージョンは表 2-1、表 2-2、表 2-3、表 2-4 をご参照ください。

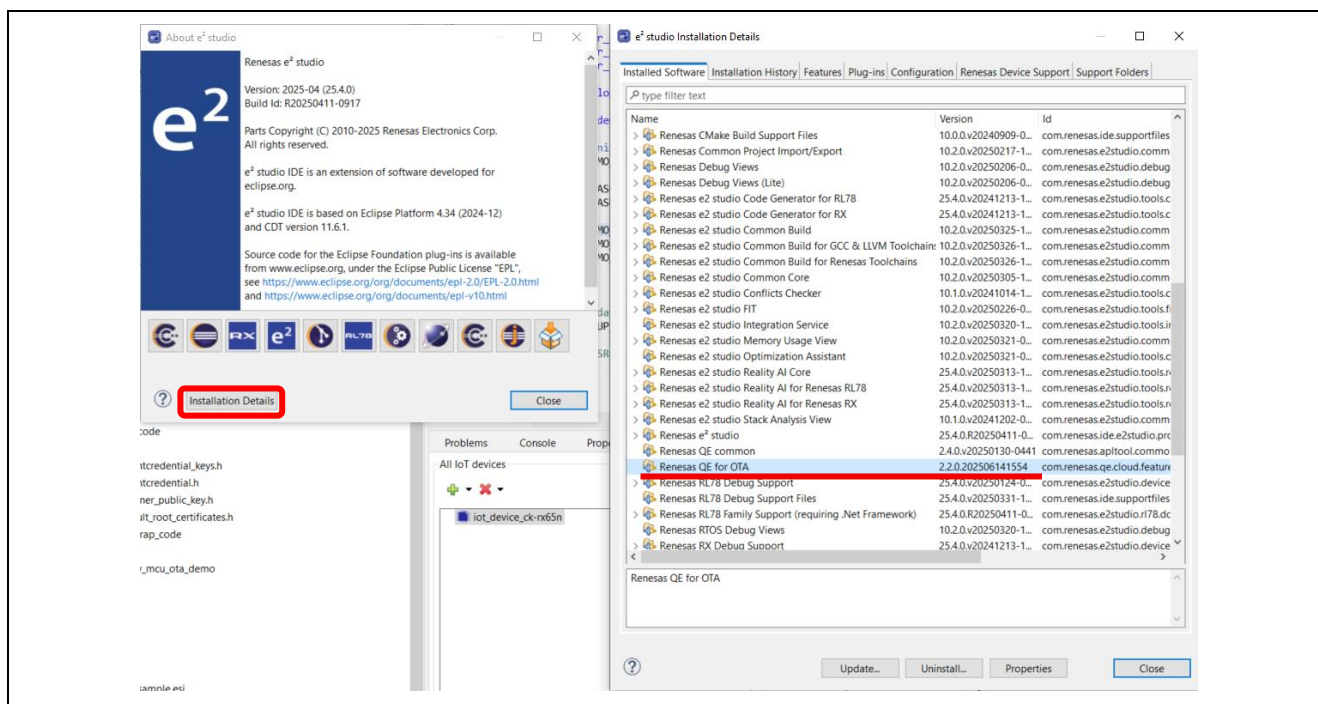
(1) QE for OTA のインストール

e² studio のメニューバーから、[Renesas Views] → [Renesas QE]を開き、QE for OTA がインストールされているか確認します。OTA Main (QE), OTA Manage IoT Device (QE)があればインストール済みです。



また、バージョンに関しては、e² studio の[Help] → [About e² studio] → [Installation Details]をクリックし、「Renesas QE for OTA」のバージョンが「2.2.0.~」以上となっていることを確認してください。





QE for OTA がインストールされていない、もしくは古いバージョンがインストールされている場合は「[RX ファミリー AWS/Azure を利用したファームウェア更新ソフトの開発ガイド QE for OTA \(R20AN0712\)](#)」の「2.1 QE for OTA のインストール」を参照し、QE for OTA をインストールしてください。

(2) Python 実行環境のインストール

Python は、<https://www.python.org/> から入手できます。

また、Python の pycryptodome ライブラリを使用します。Python をインストール後、以下の pip コマンドを実行し、インストールしてください。

```
> pip install pycryptodome
```

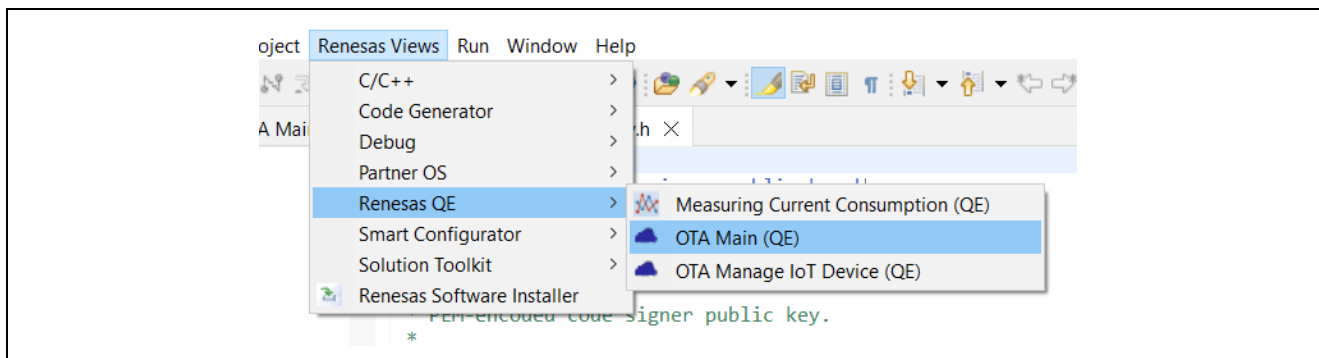
(3) Renesas Flash Programmer のインストール

Renesas Flash Programmer は、[Renesas Flash Programmer \(Programming GUI\) | Renesas](#) から入手できます。

6.2.2 QE for OTA で AWS にログイン

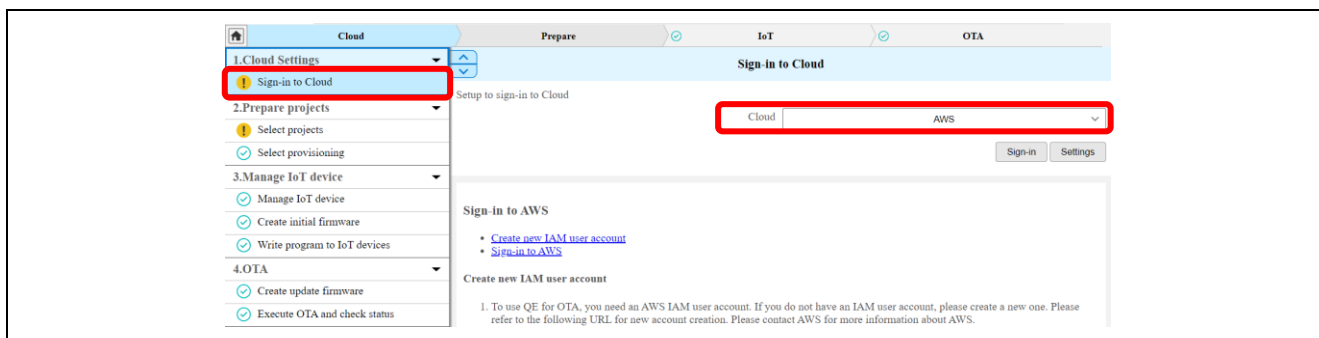
(1) QE for OTA 画面を開く

e² studio のメニューバーから、[Renesas Views] → [Renesas QE] → [OTA Main (QE)]を選択します。



(2) [QE for OTA] 1. Cloud Settings – Sign-in to Cloud

Cloud に「AWS」を選択しサインインします。ログイン時に選択したリージョンに AWS のリソースが生成されます。



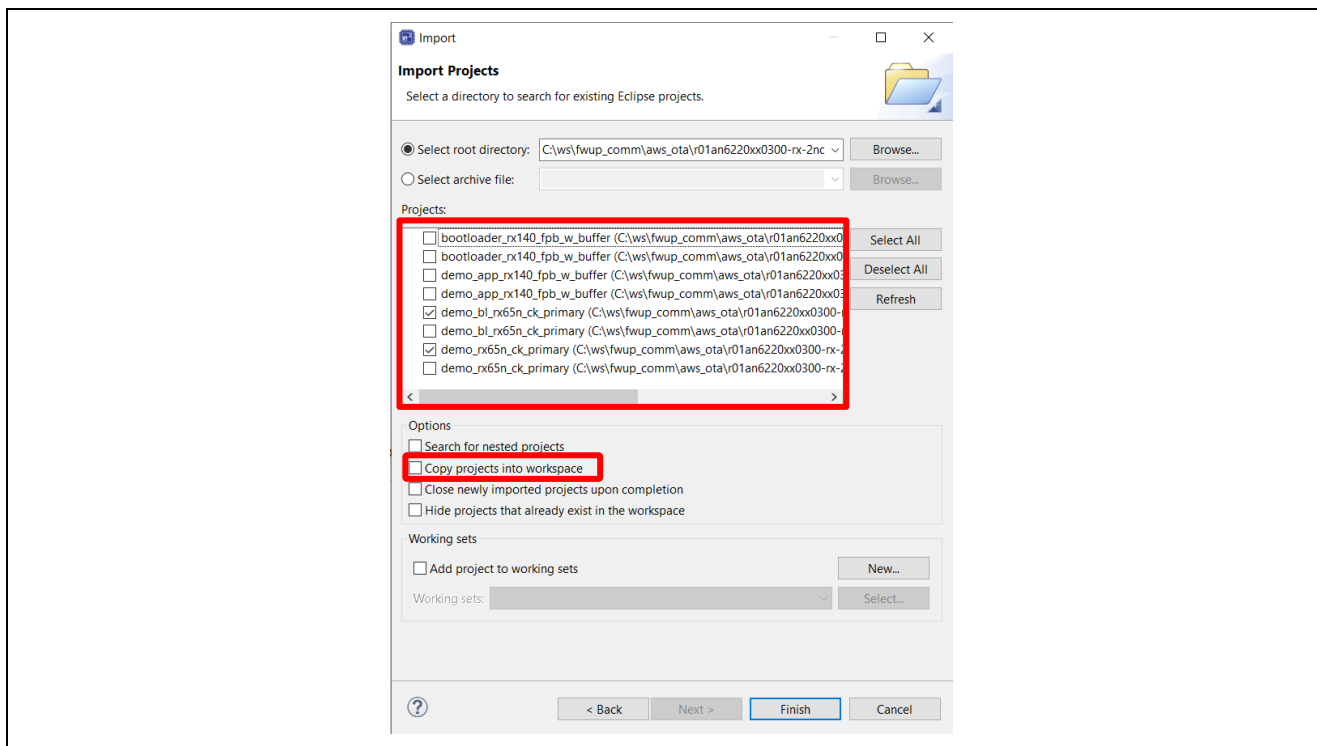
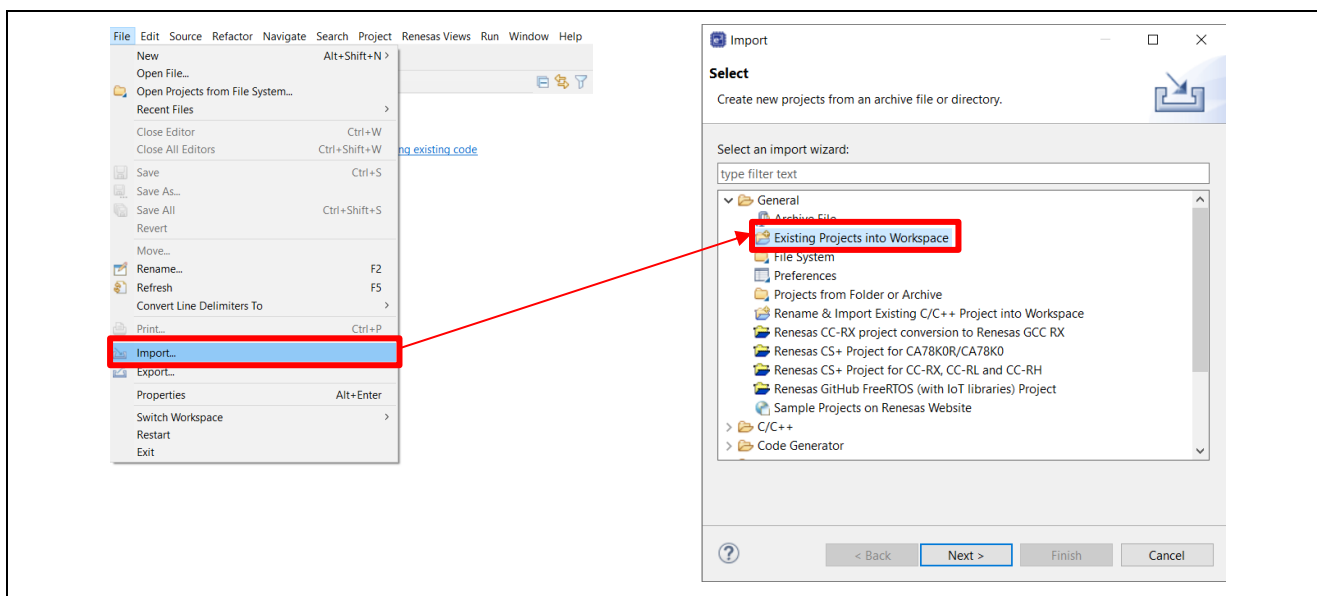
6.2.3 CK-RX65N 用の初期ファームウェアの作成と実行

QE for OTA を使って CK-RX65N 用の初期ファームウェアを作成し、書き込み・実行します。以下に手順を示します。

(1) プロジェクトのインポート

CK-RX65N 用のブートローダである **demo_bl_rx65n_ck_primary** プロジェクトと、ユーザプログラムである **demo_rx65n_ck_primary** プロジェクトを e² studio にインポートします。CK-RX65N のプロジェクトは CC-RX と GCC 用がありますが、ここでは CC-RX 用プロジェクトを使用して説明します。

インポート時は、オプションの「Copy projects into workspace」のチェックを外してください。



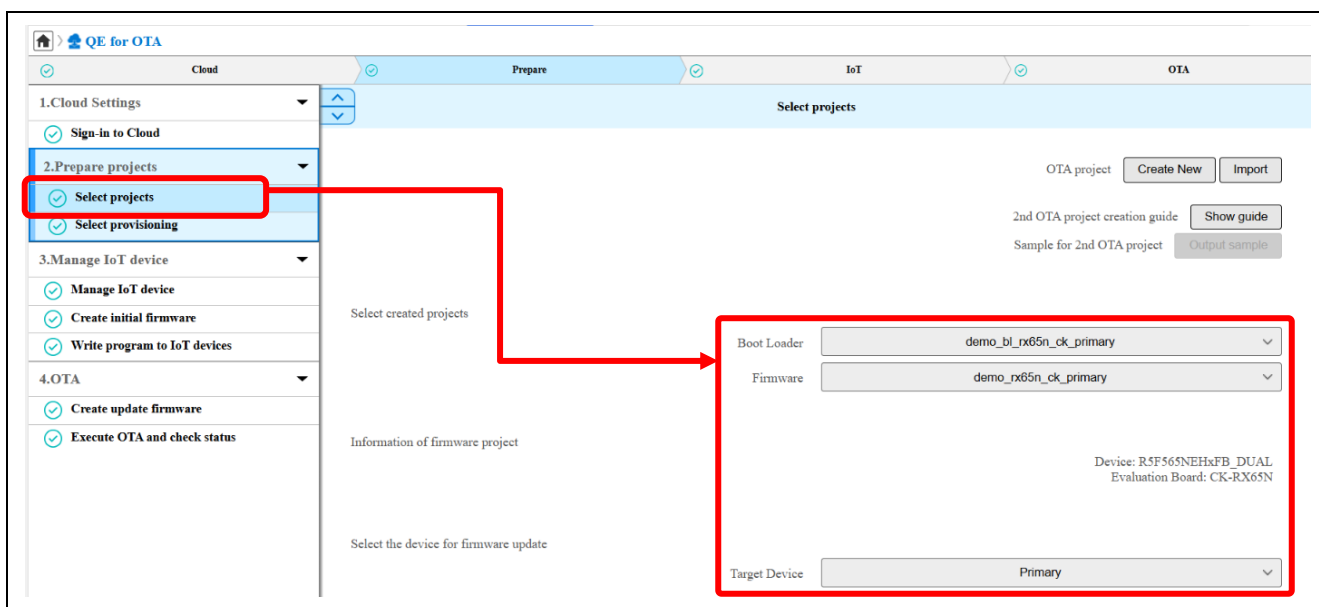
(2) Wi-Fi の接続情報の入力

demo_rx65n_ck_primary プロジェクトの「src/application_code/include/aws_clientcredential.h」で定義されている clientcredentialWIFI_SSID に、使用する Wi-Fi アクセスポイントの SSID を、clientcredentialWIFI_PASSWORD にパスワードを入力します。



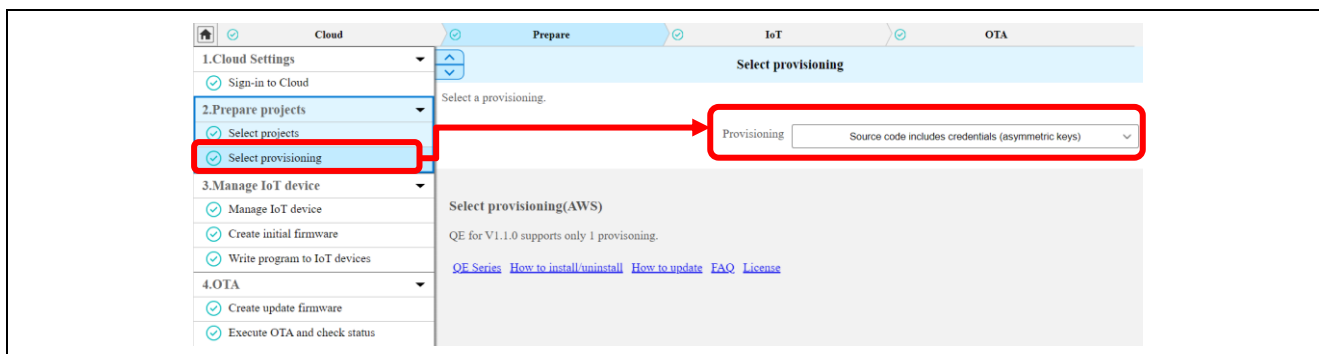
(3) プロジェクトの選択

QE for OTA の「Select projects」で、Boot Loader に先ほど e² studio にインポートした demo_bl_rx65n_ck_primary プロジェクト、Firmware に demo_rx65n_ck_primary プロジェクトを選択します。また、Target Device には「Primary」を選択します。



(4) プロビジョニング方法の選択

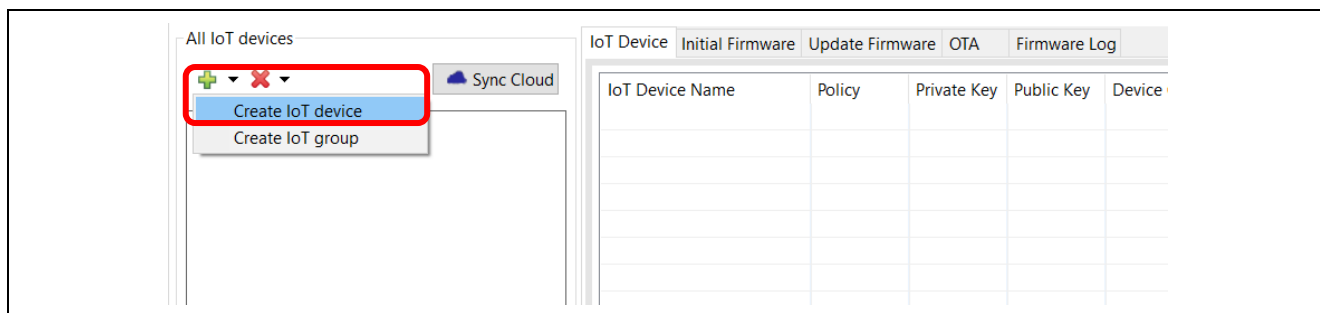
プロビジョニング方法として、「Source code includes credentials (asymmetric keys)」を選択します。



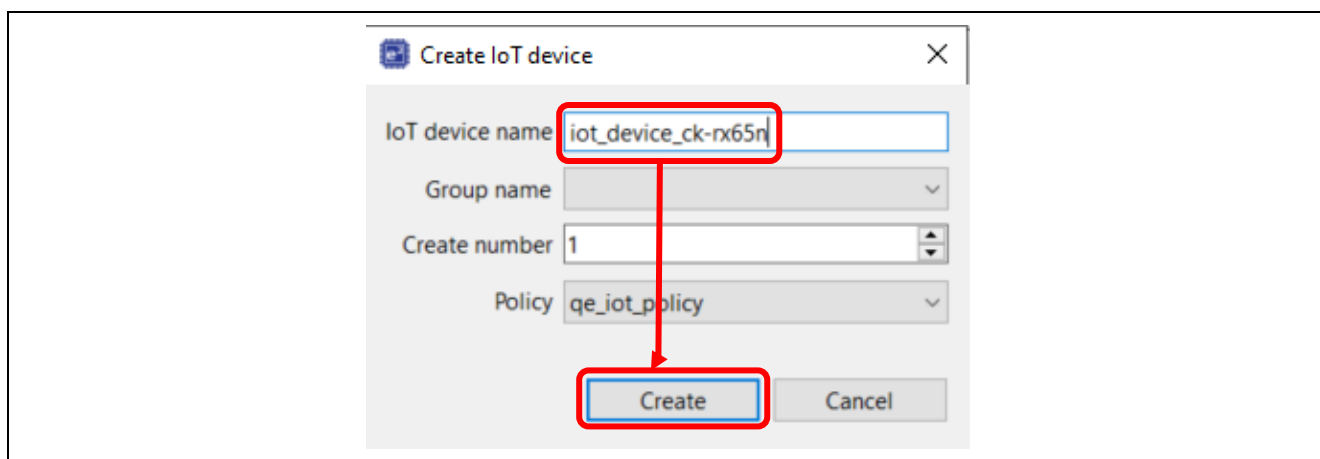
(5) IoT device の作成

Manage IoT device の「Open view」をクリックし、「OTA Manage IoT Device (QE)」画面を開きます。

「All IoT devices」の下の「+」ボタンをクリックし、「Create IoT device」を選択します。

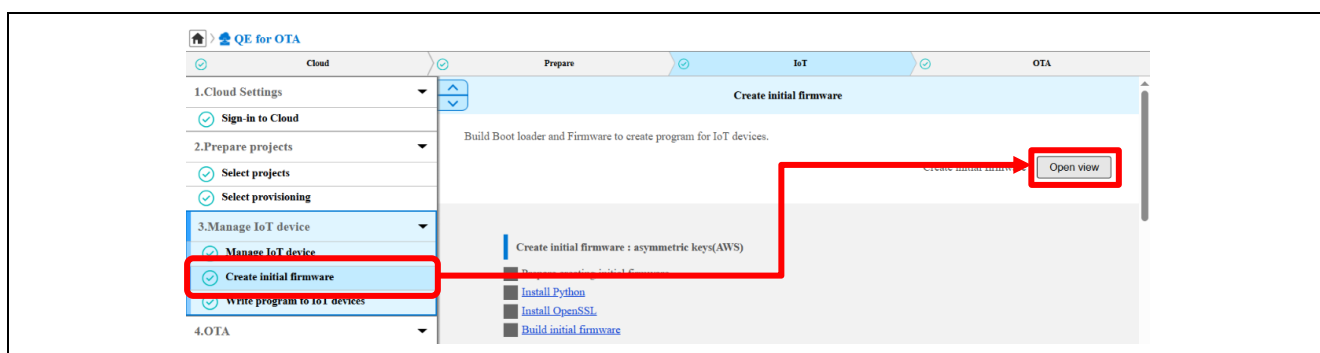


表示されたダイアログの「IoT device name」に適切な名前を入力し、「Create」をクリックします。

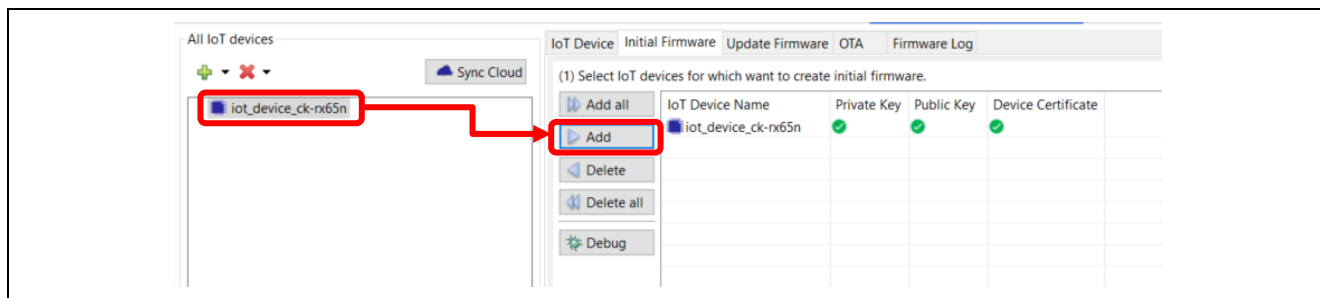


(6) 初期ファームウェアの作成

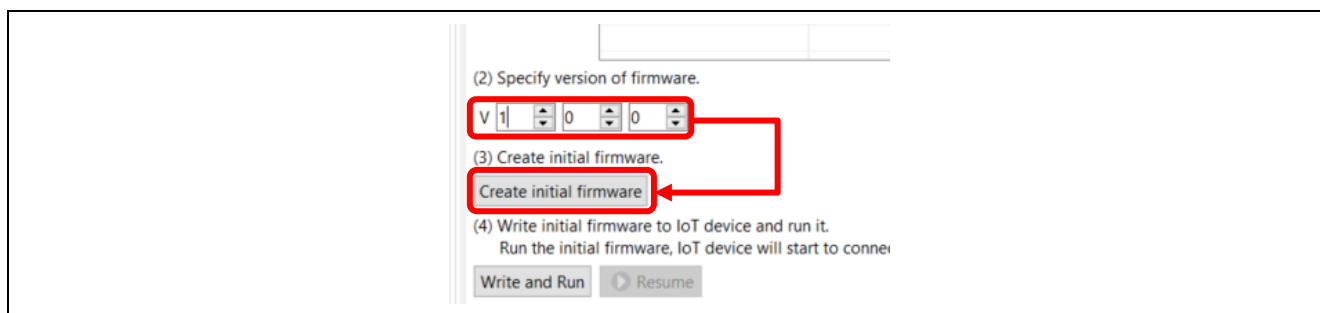
Create initial firmware の「Open view」をクリックします。



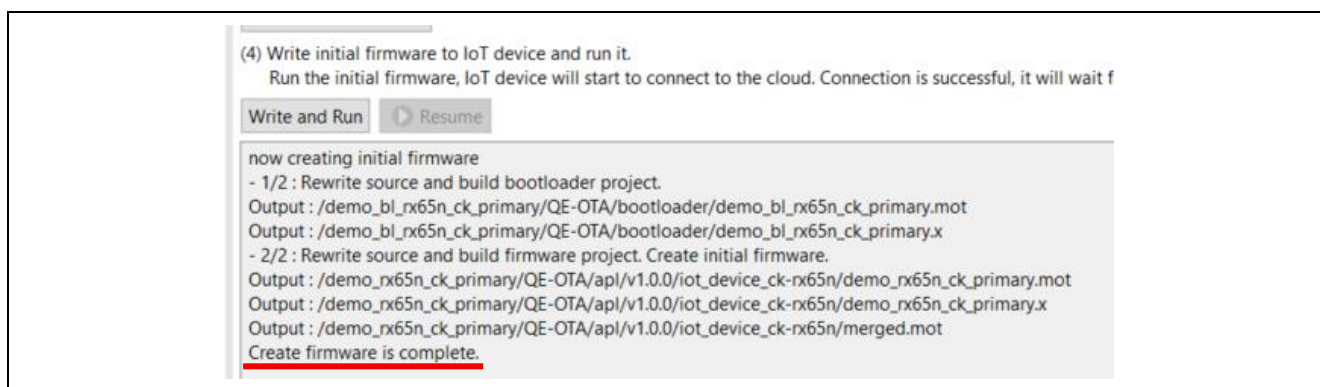
「Initial Firmware タブ」で、先ほど作成した IoT device を選択し、「▶Add」をクリックします。



バージョンに「1.0.0」を入力し、「Create initial firmware」をクリックします。

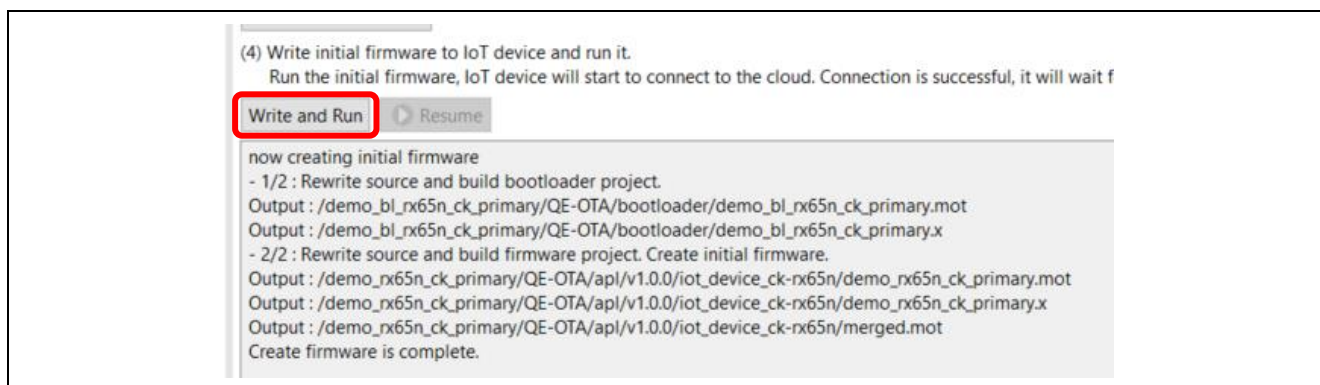


demo_bl_rx65n_ck_primary プロジェクトと demo_rx65n_ck_primary プロジェクトのビルドが実行され、「Initial Firmware」タブに「Create firmware is complete.」と表示されれば成功です。

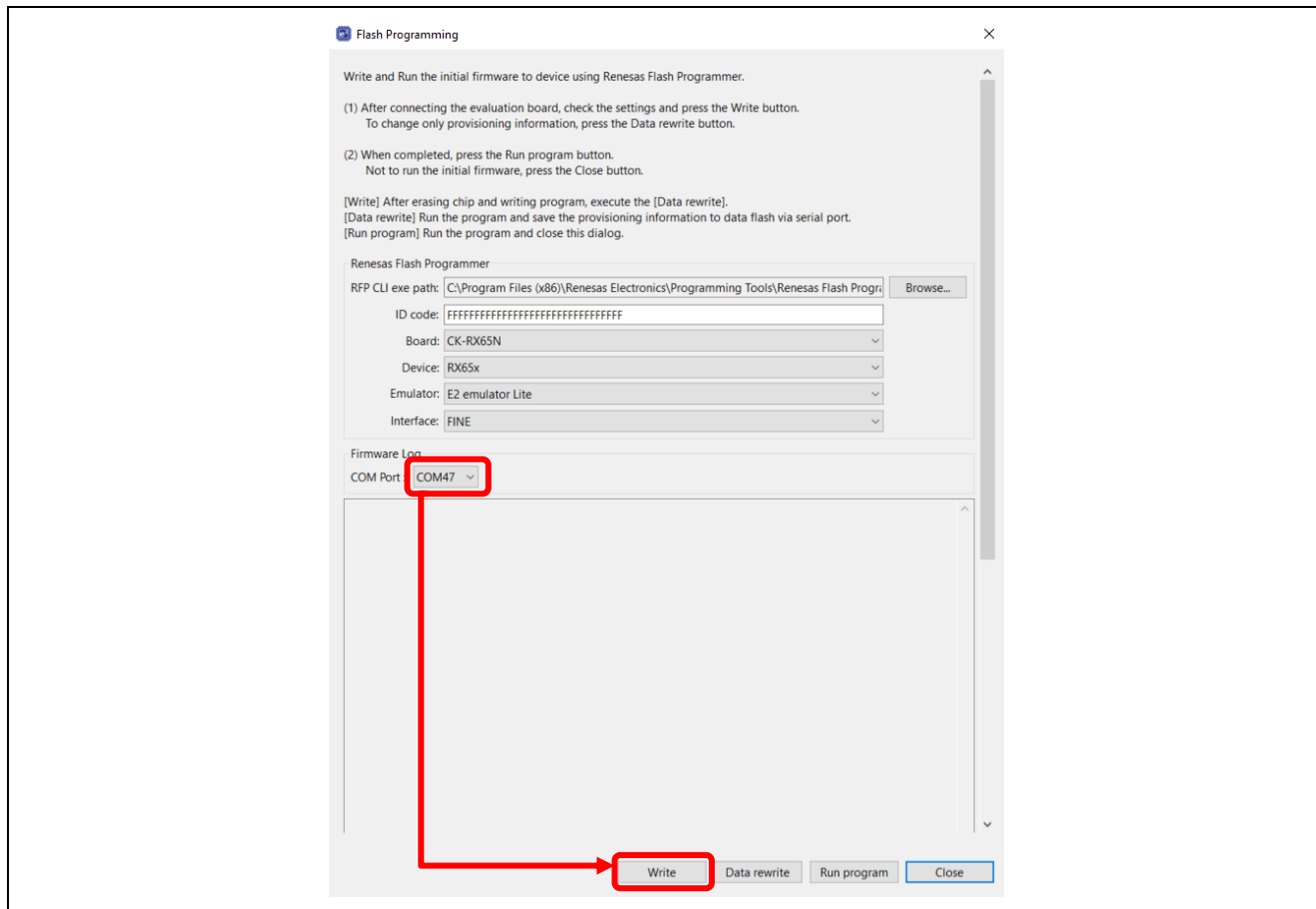


(7) CK-RX65N に初期ファームウェアを書き込み実行する

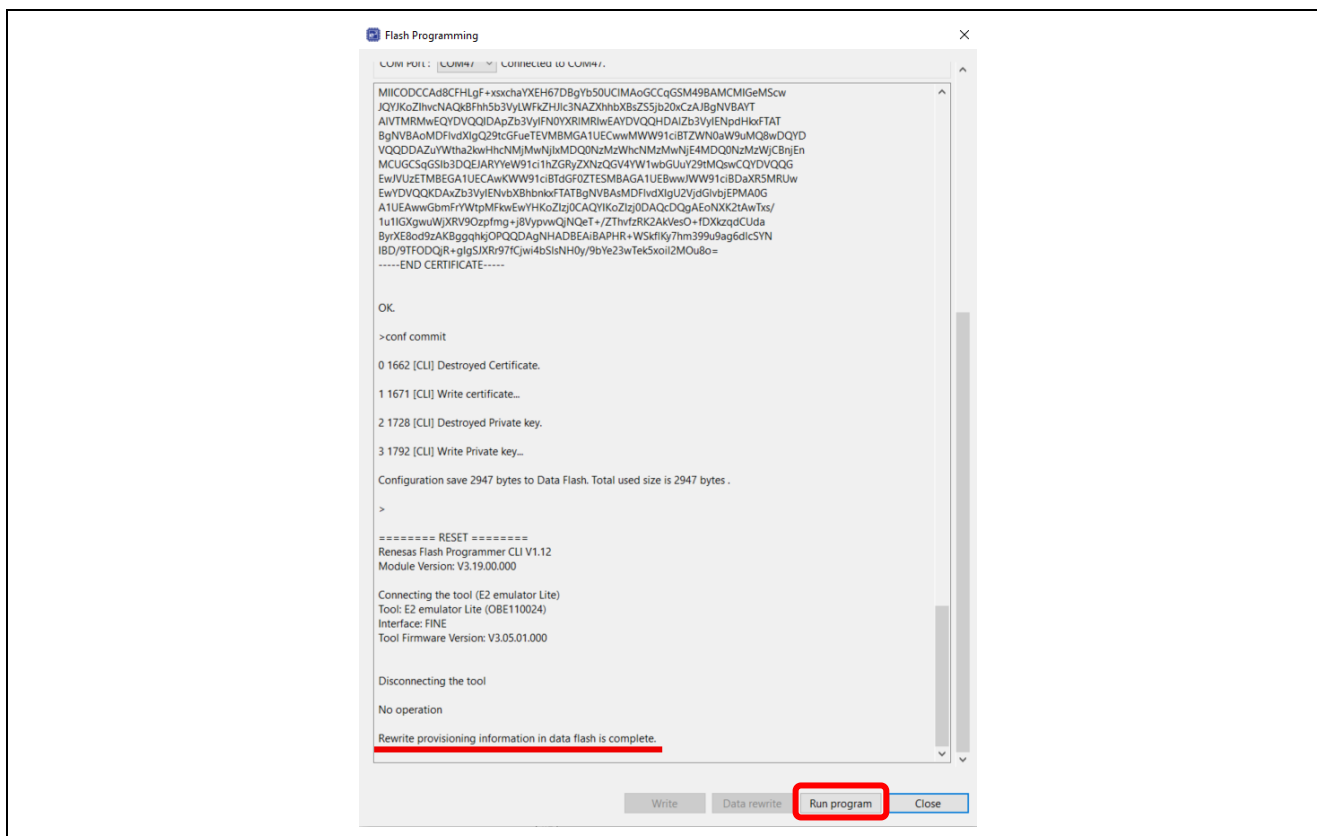
「Initial Firmware」タブの「Write and Run」をクリックします。



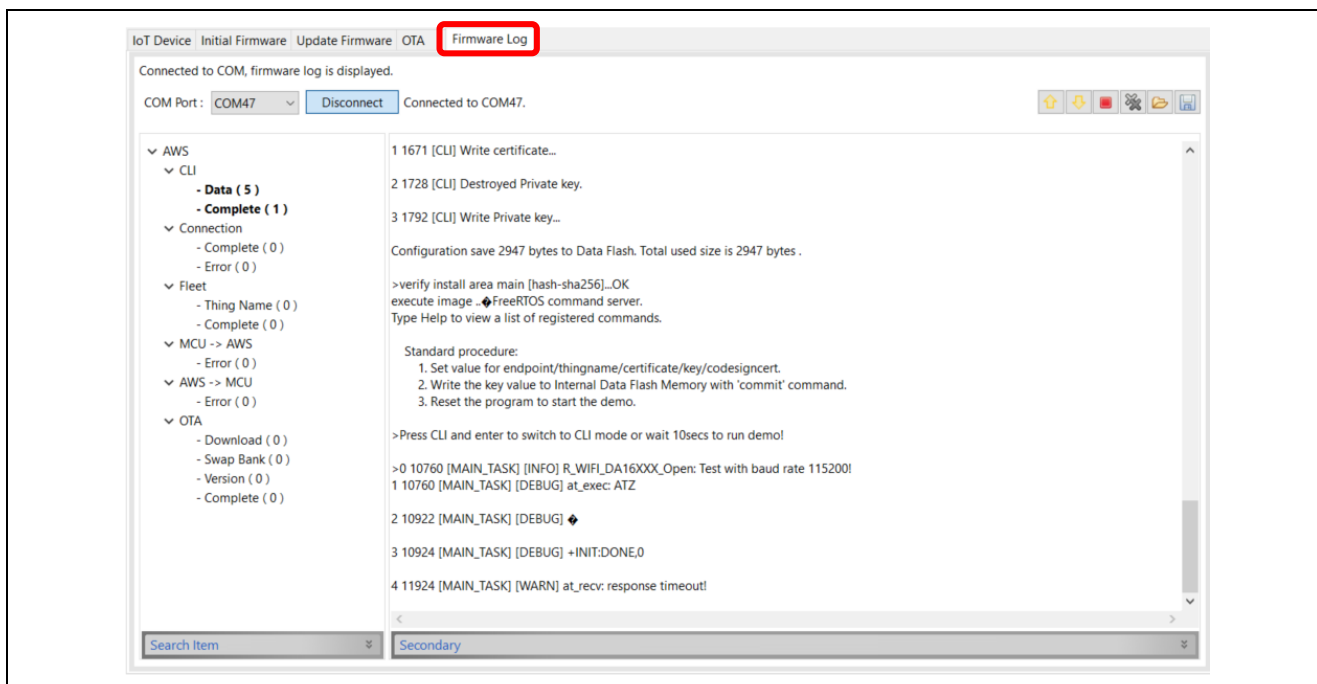
CK-RX65N の COM ポート番号を選択し、「Write」をクリックします。Renesas Flash Programmer の各パラメータは、CK-RX65N のオンボードエミュレータが接続されていれば自動入力されます。空欄になっている場合は、CK-RX65N と PC が接続されていること、ジャンパ J16 が DEBUG 側に接続されていることを確認してください。



「Rewrite provisioning information in data flash is complete.」と表示されれば成功です。「Run program」をクリックします。

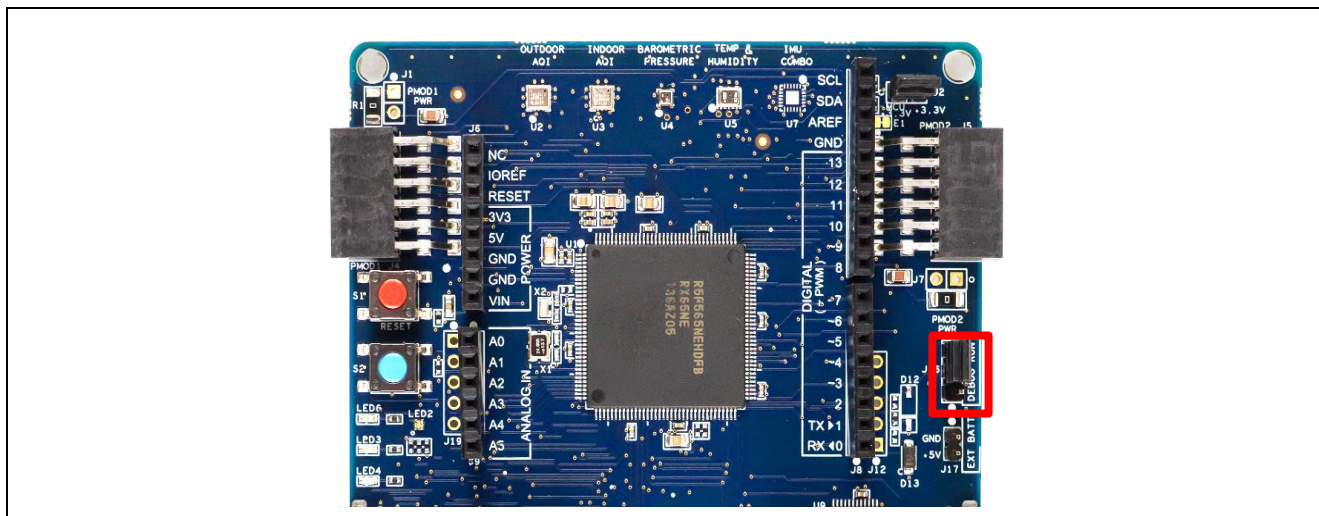


「Firmware Log」タブを開きます。ここには CK-RX65N から出力されるログが表示されます。



(8) CK-RX65N を RUN モードに変更

CK-RX65N のジャンパ J16 を RUN 側(pin 2-3)に短絡します。



6.2.4 FPB-RX140 用の初期ファームウェアの作成と実行

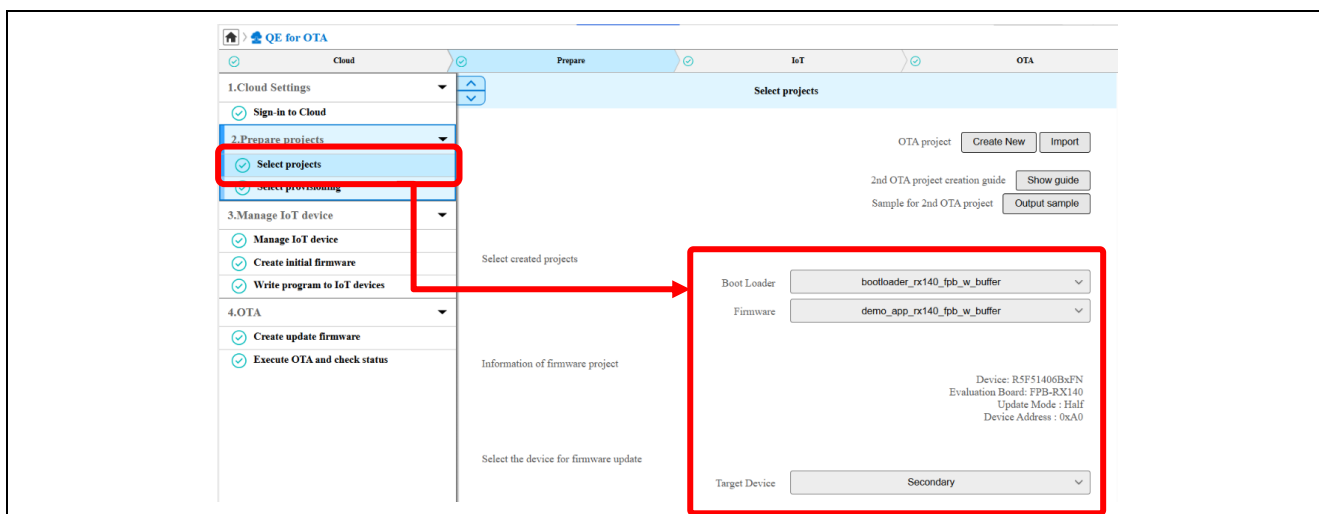
QE for OTA を使って FPB-RX140 用の初期ファームウェアを作成し、書き込み・実行します。以下に手順を示します。

(1) プロジェクトのインポート

先ほどの CK-RX65N 用のプロジェクトと同様に、本アプリケーションノートでサンプルコードとして提供している、FPB-RX140 用のブートローダである **bootloader_rx140_fpb_w_buffer** プロジェクトと、ユーザプログラムである **demo_app_rx140_fpb_w_buffer** プロジェクトを e² studio にインポートします。

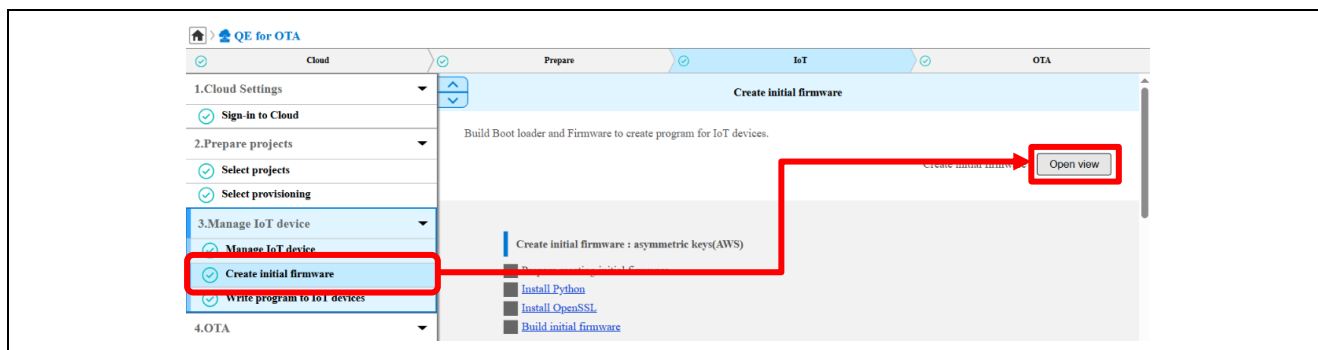
(2) プロジェクトの選択

QE for OTA の「Select projects」で、Boot Loader に先ほど e² studio にインポートした **bootloader_rx140_fpb_w_buffer** プロジェクト、Firmware に **demo_app_rx140_fpb_w_buffer** プロジェクトを選択します。また、Target Device には「Secondary」を選択します。

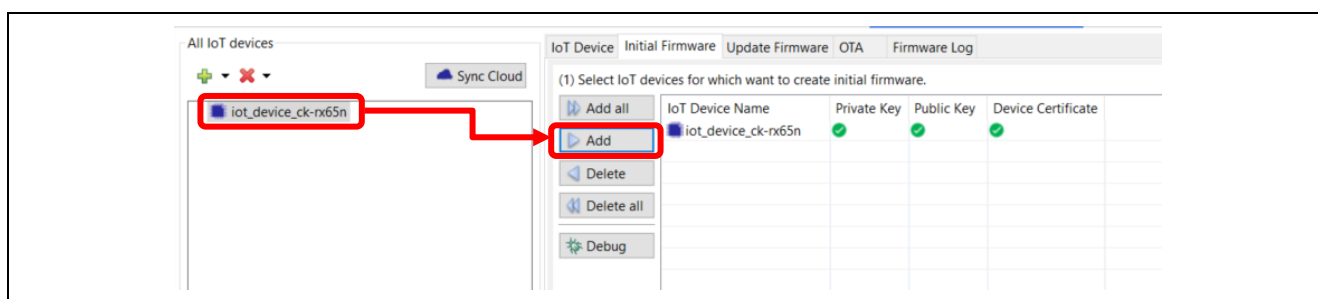


(3) 初期ファームウェアの作成

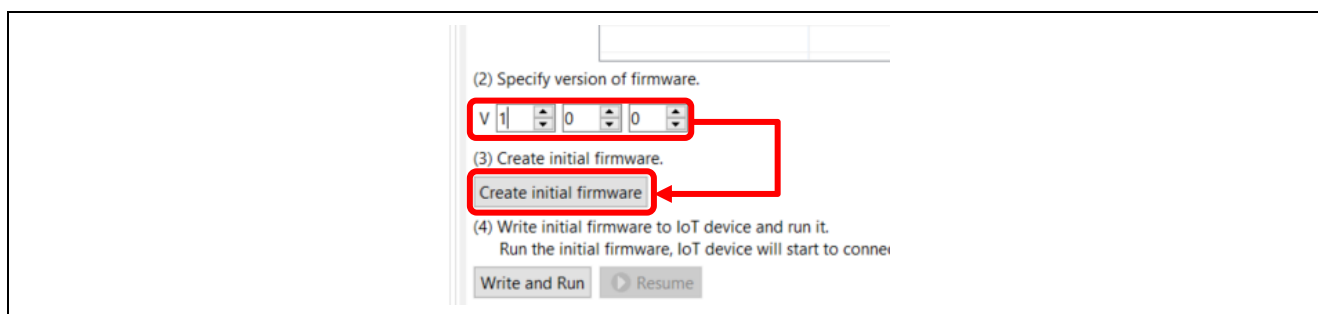
Create initial firmware の「Open view」をクリックし、「OTA Manage IoT Device (QE)」画面を開きます。



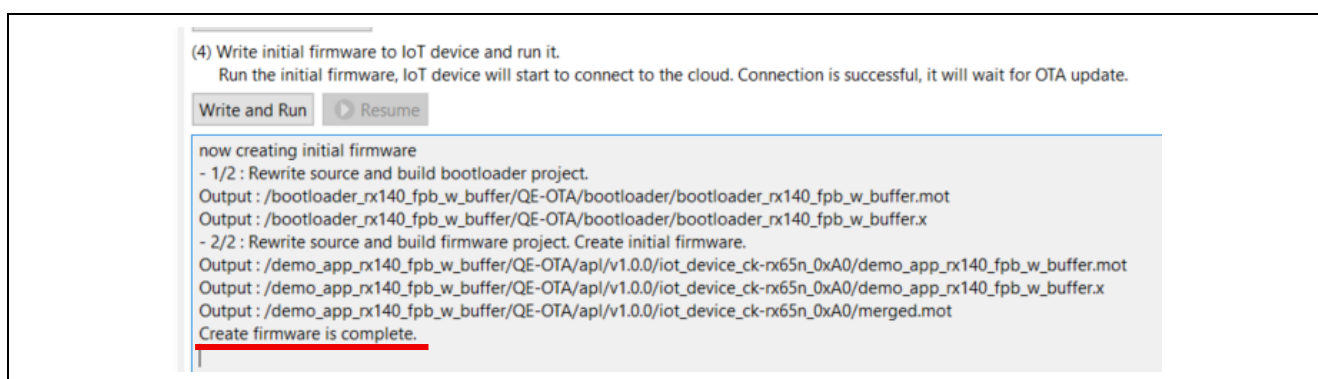
「Initial Firmware タブ」で、6.2.3(5)で作成した IoT device を選択し、「▶Add」をクリックします。



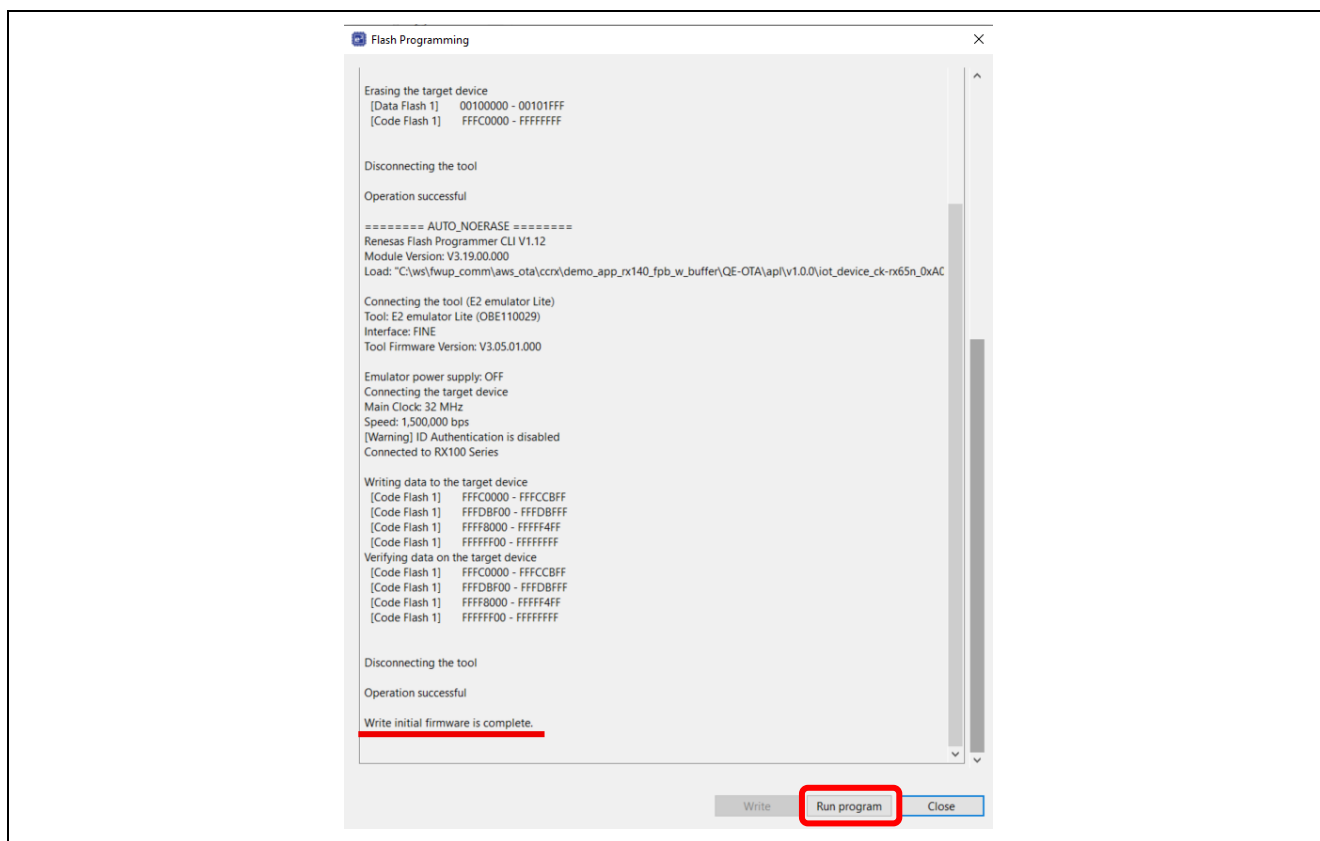
バージョンに「1.0.0」を入力し、「Create initial firmware」をクリックします。



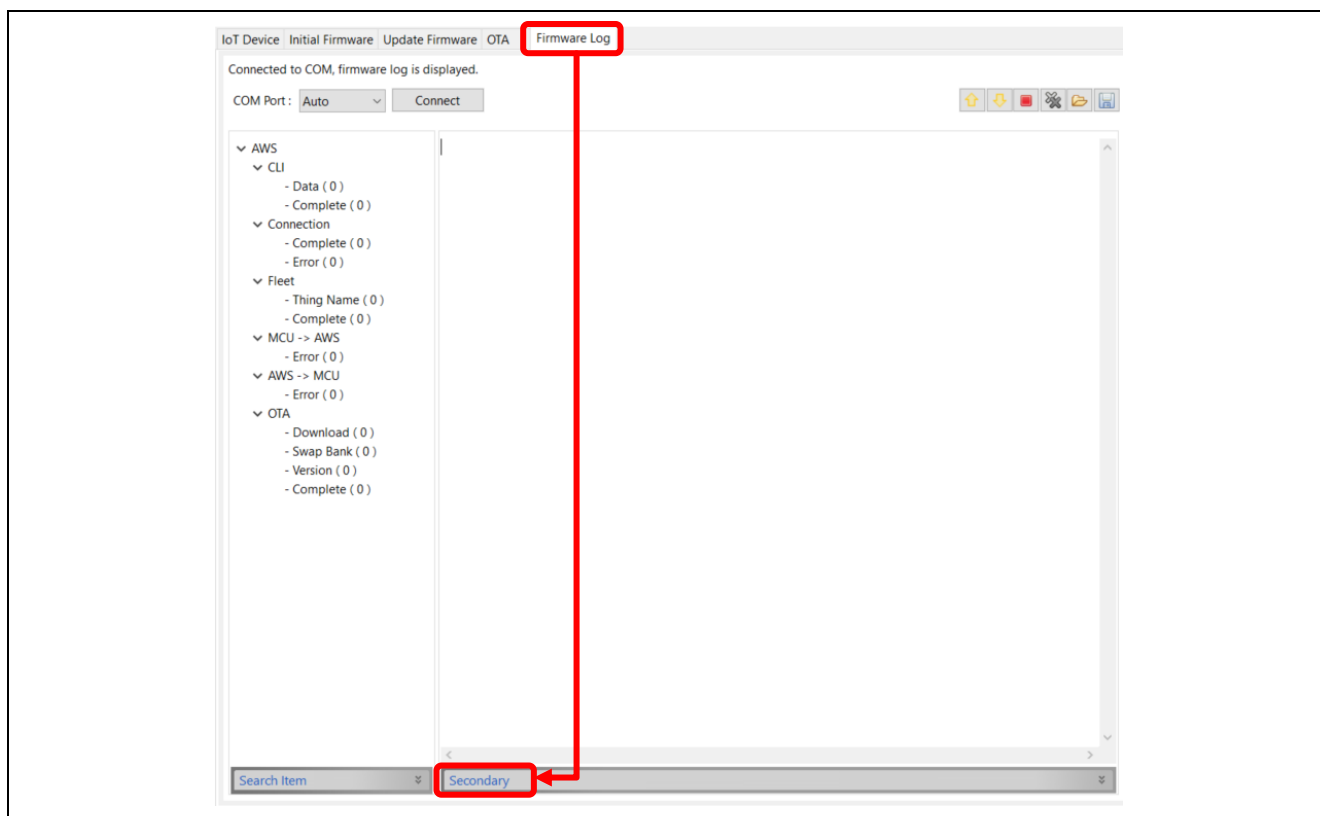
bootloader_rx140_fpb_w_buffer プロジェクトと demo_app_rx140_fpb_w_buffer プロジェクトのビルドが実行され、「Initial Firmware」タブに「Create firmware is complete.」と表示されれば成功です。



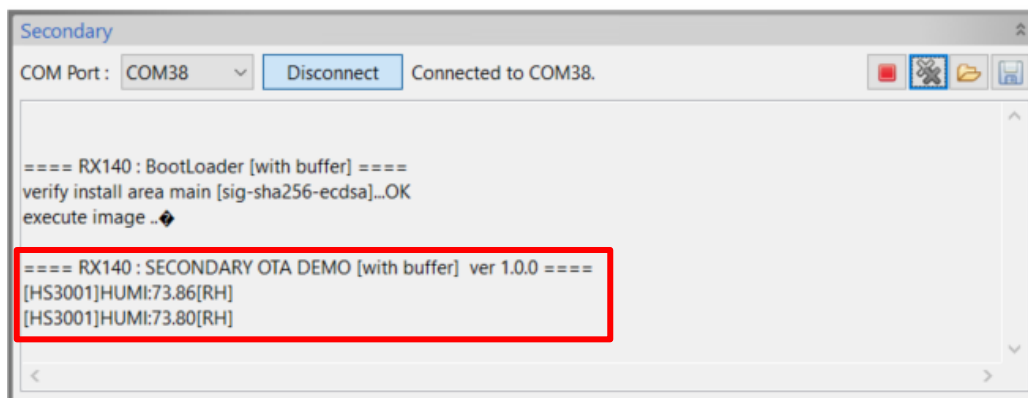
「Write initial firmware is complete.」と表示されれば成功です。「Run program」をクリックします。



「Firmware Log」タブを開いて、「Secondary」をクリックします。



ここには FPB-RX140 から出力されるログが表示されます。「ver 1.0.0」の表示のあとに、HS3001 センサで測定した湿度データが表示されれば成功です。



6.3 AWS クラウド上でセンサデータを可視化するための準備

受信したセンサデータをグラフ形式で可視化するために、Amazon CloudWatch および AWS IoT Core で以下の手順で設定を行います。

Note グラフ形式での可視化は必要なく、データが AWS に届いていることをブラウザ上で確認できればいいという場合は本 6.3 章の作業は丸ごと省略可能です。

この場合、図 6-3 のように AWS IoT の「MQTT テストクライアント」で「iotdemo/topic/sensor」をサブスクライブすることでセンサデータが期待通り受信できていることをテキスト形式で確認できます。

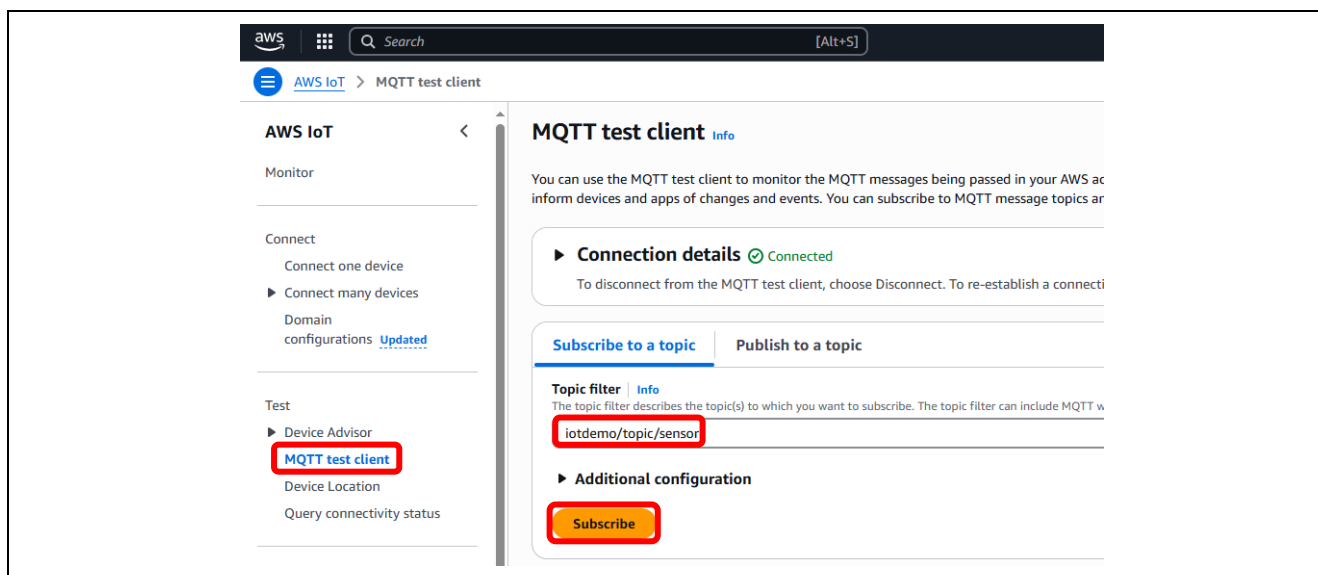


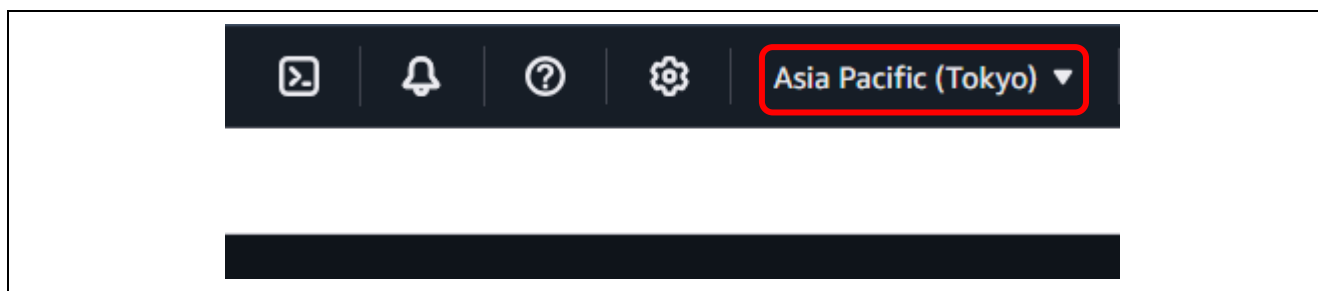
図 6-3 MQTT テストクライアントでのデータ受信確認

(1) AWS マネジメントコンソールへのログイン

AWS マネジメントコンソールにログインします。

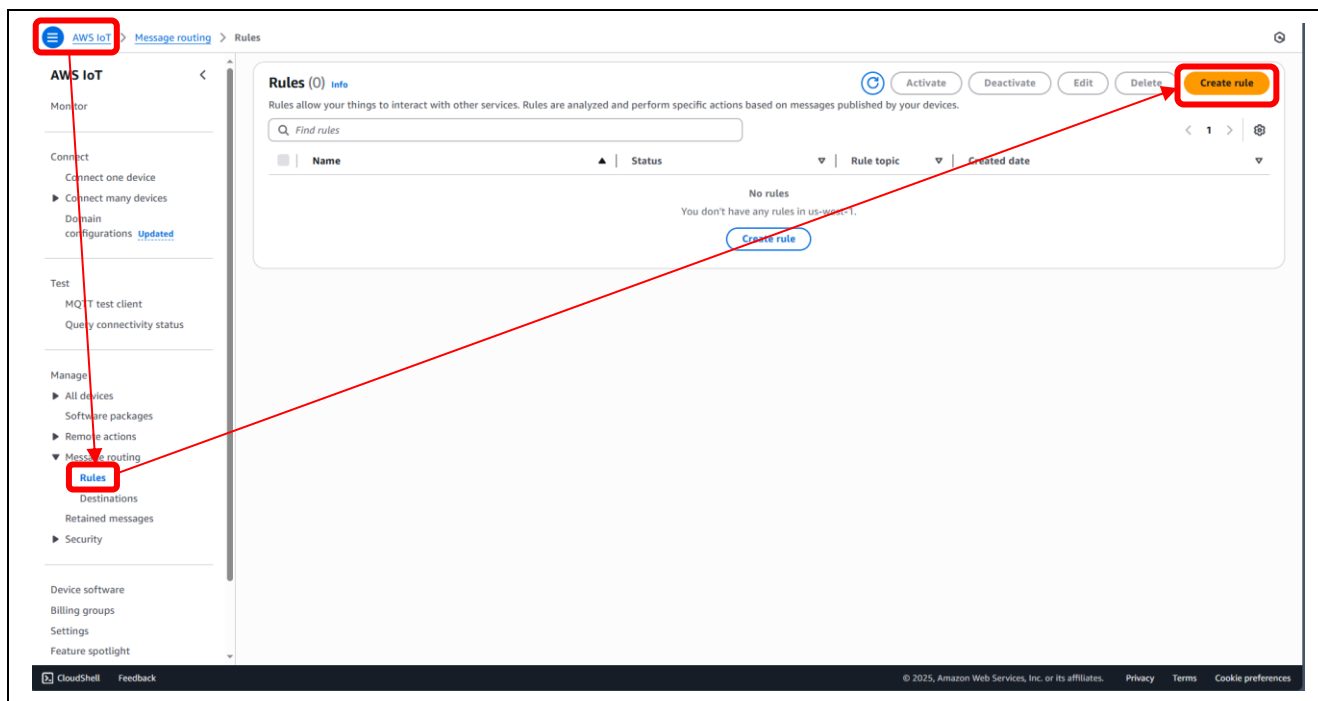
[AWS マネジメントコンソール | AWS \(amazon.com\)](#)

マネジメントコンソール画面の右上に表示されているリージョンを確認し、QE for OTA ログイン時の設定と同じリージョンを選択します。



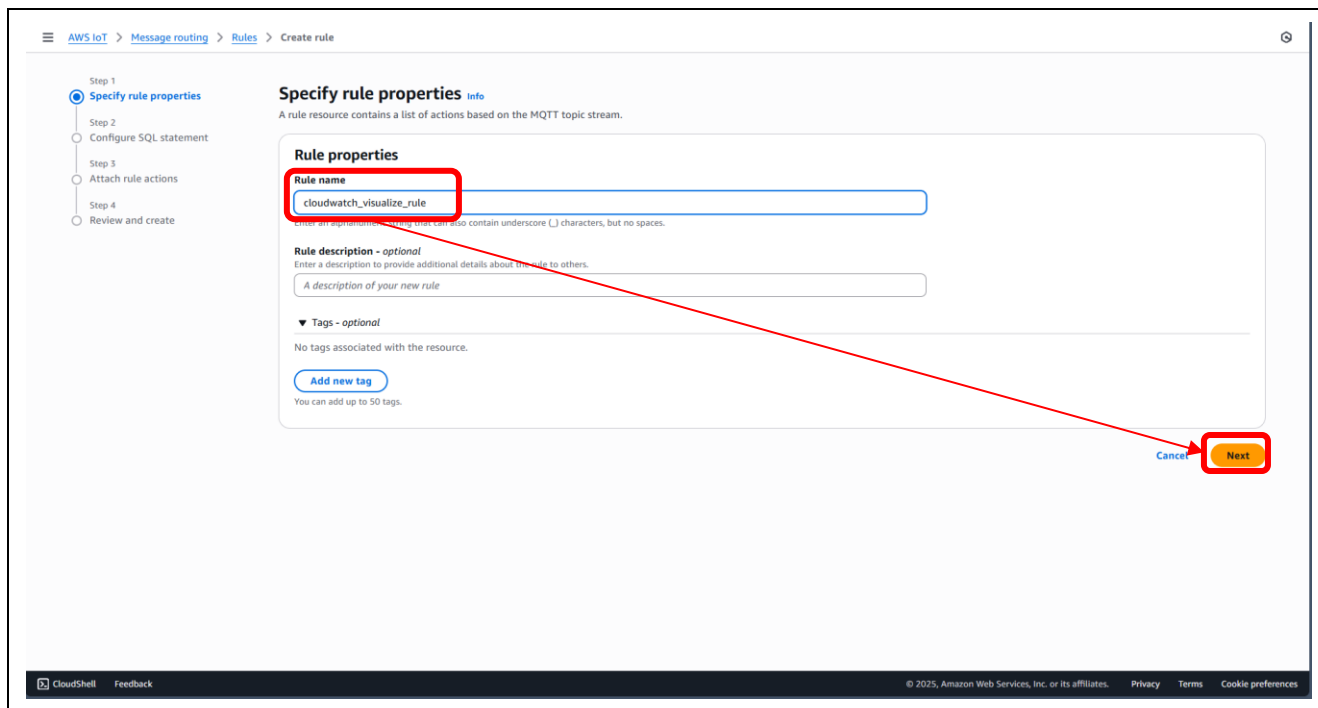
(2) AWS IoT でルールを作成

[AWS IoT] ⇒ [ルール] ⇒ [ルールを作成]をクリックします。



(3) ルールのプロパティを指定

ルール名を入力し、[次へ]をクリックします。

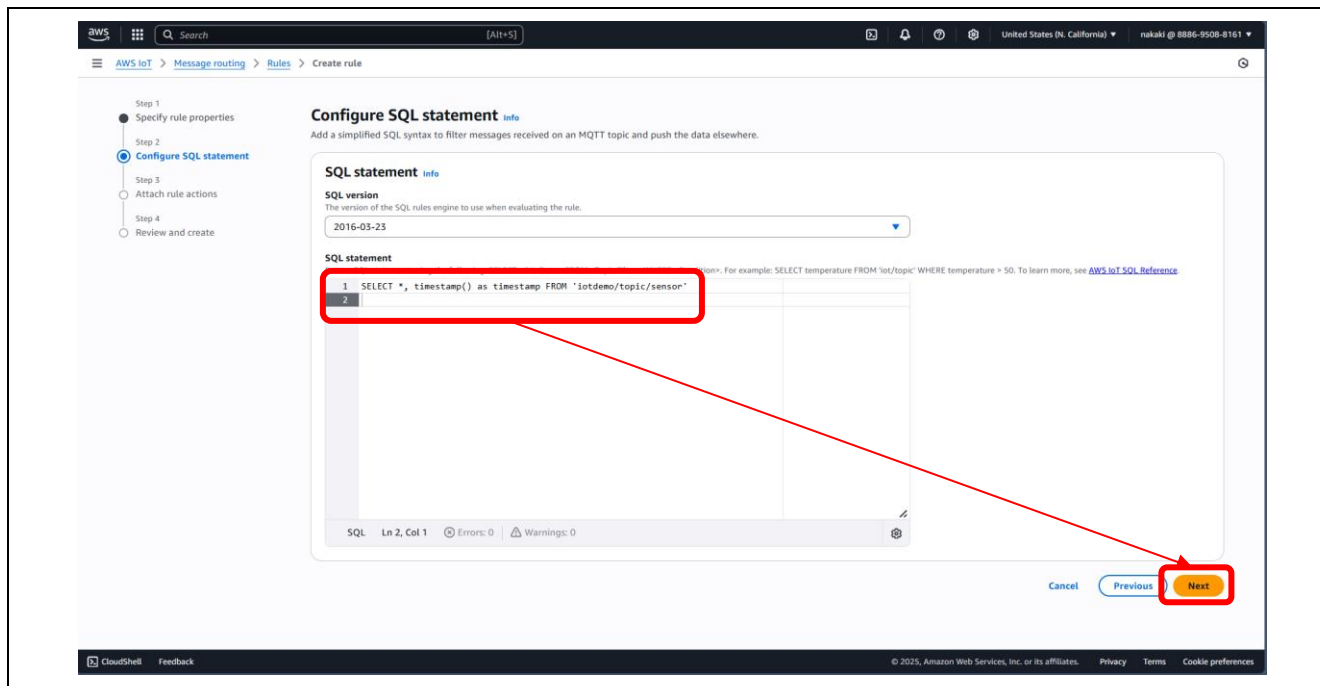


(4) SQL ステートメントを設定

SQL ステートメントを入力します。テキストエディタに以下のように入力します。最終行に改行が必要な点に注意してください。

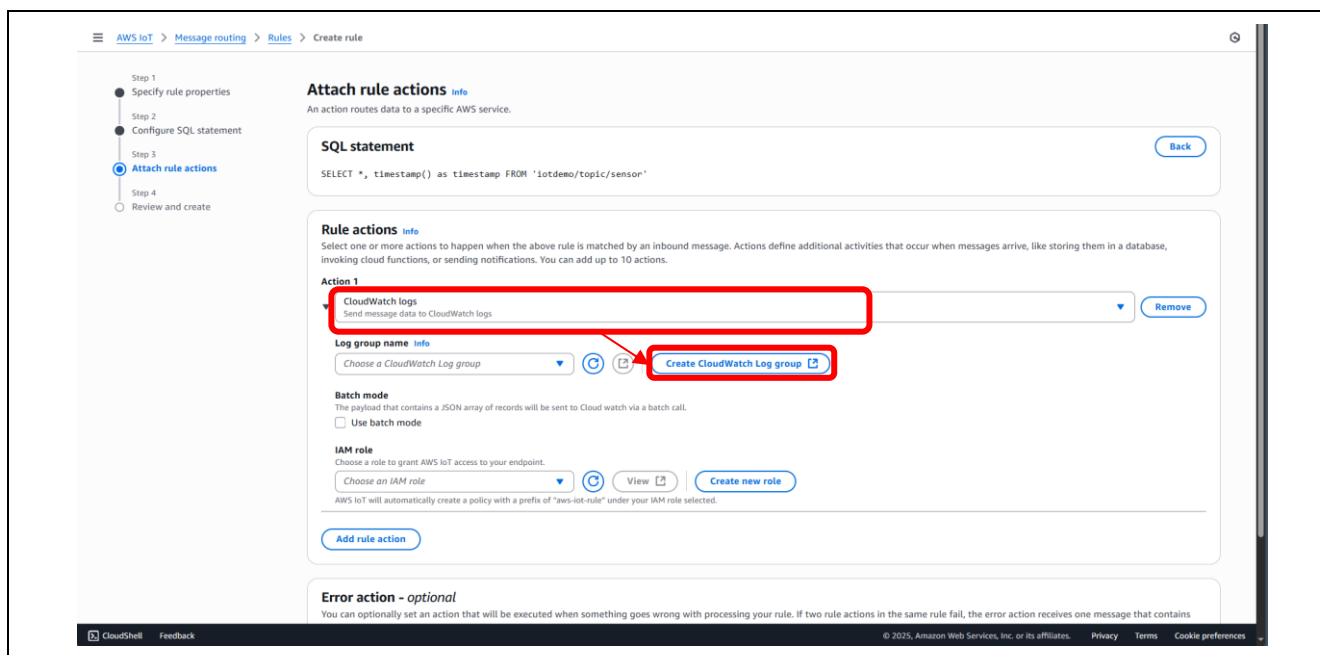
```
SELECT *, timestamp() as timestamp FROM 'iotdemo/topic/sensor'
```

(改行が必要)



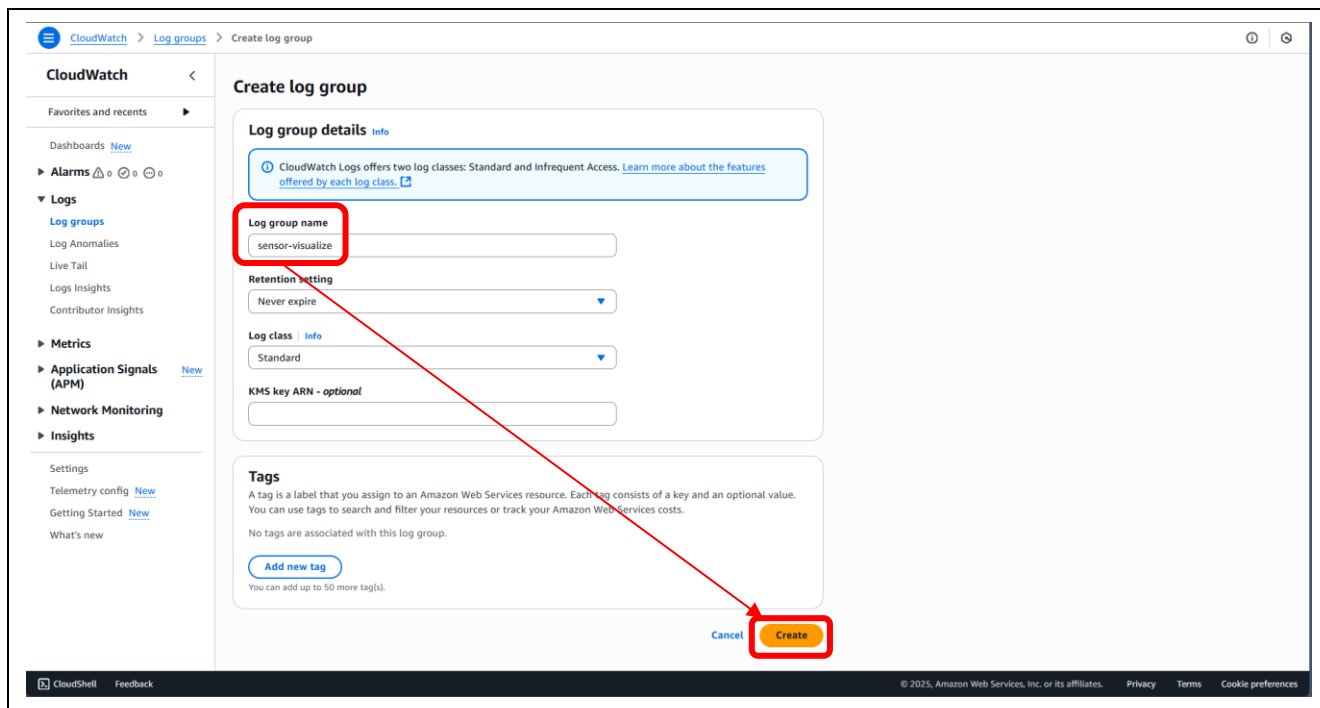
(5) 「ルールアクションをアタッチ」ステップでルールアクションを選択

アクション 1 に「CloudWatch logs」を選択し、[CloudWatch Log グループを作成]をクリックします。



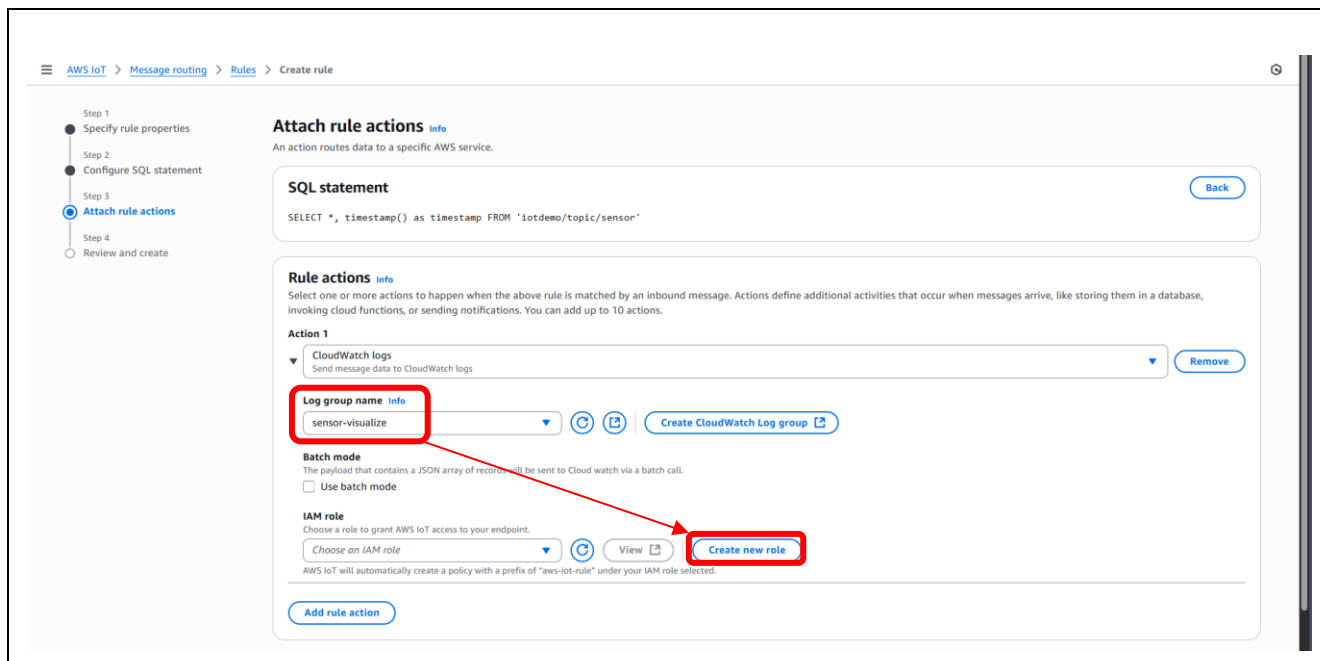
(6) ロググループを作成

ロググループ名を入力し、[作成]をクリックします。

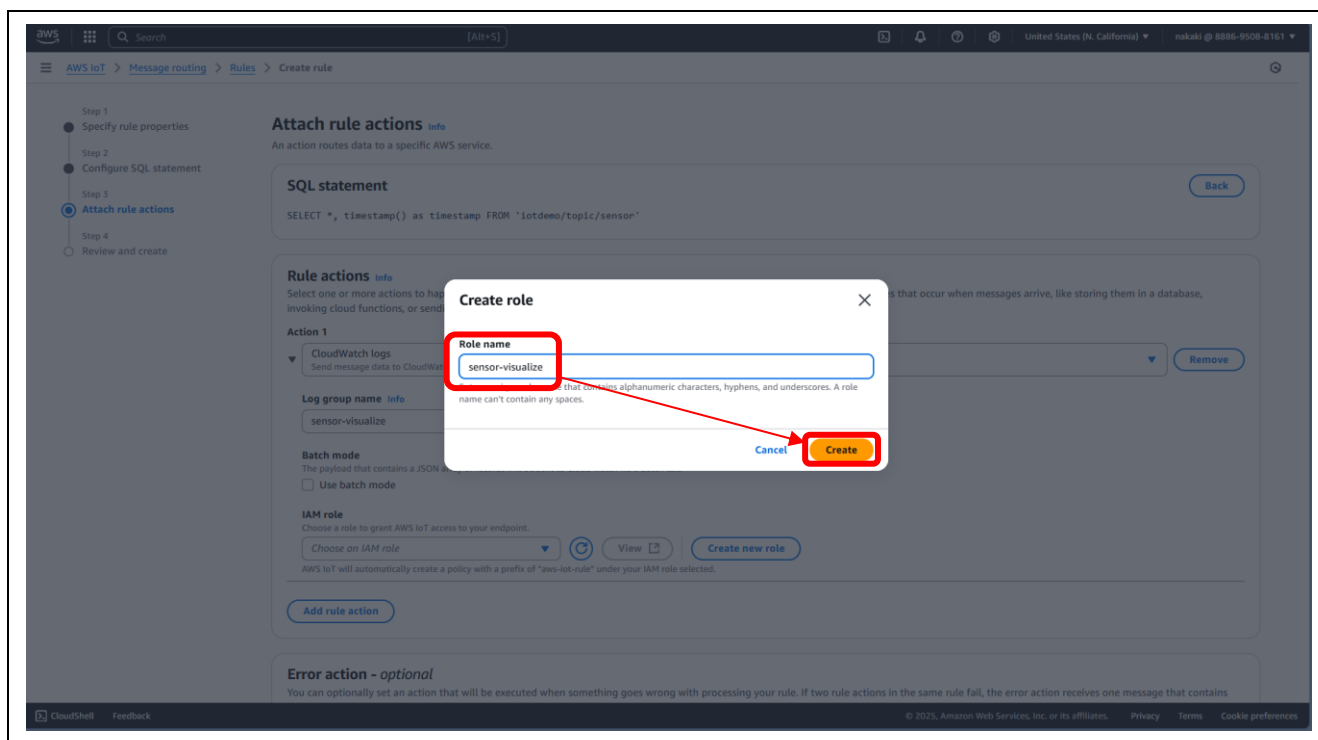


(7) 新しいロールを作成

ルール of 作成画面に戻り、ログのグループ名に、作成したロググループを選択し、[新しいロールを作成]をクリックします。

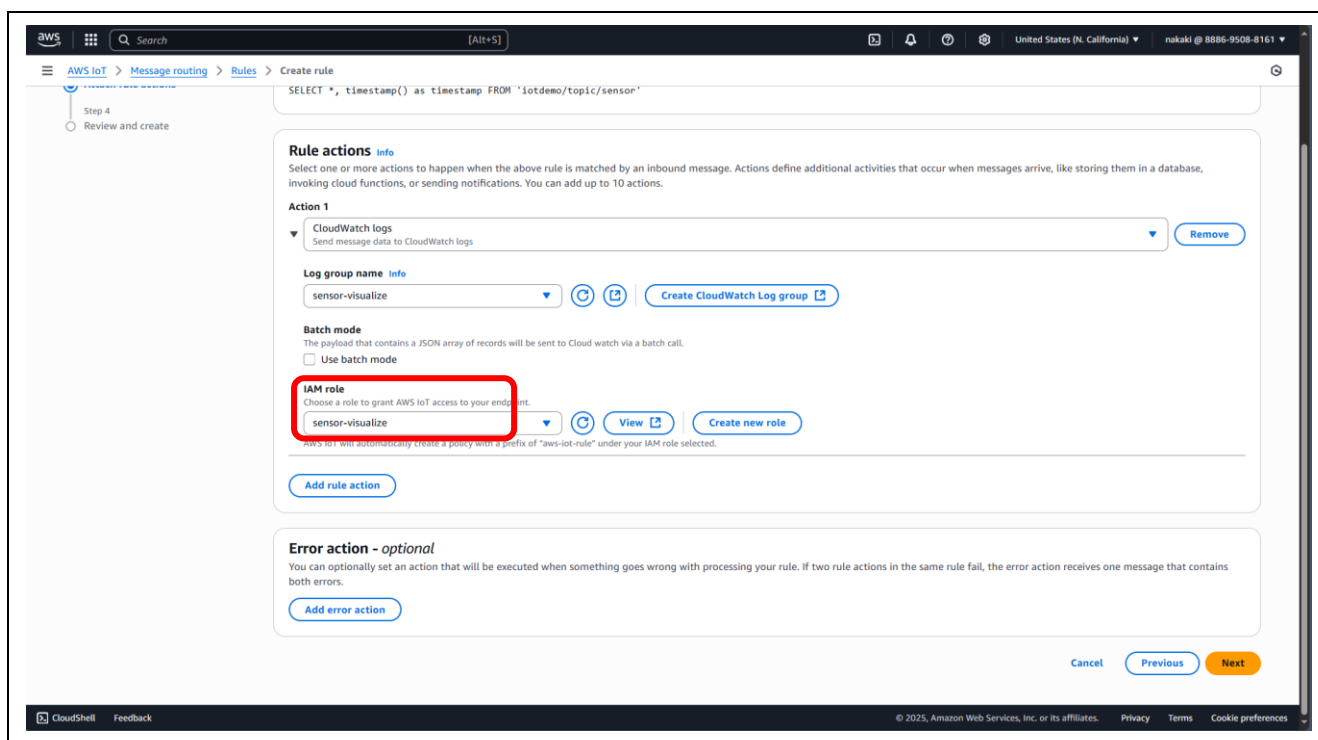


ロール名を入力し、[作成]をクリックします。



(8) 作成した IAM ロールを選択

IAM ロールに、作成したロールを選択します。



(9) ルール作成に成功したことを確認

[次へ] ⇒ [作成]と進み、ルールの一覧に、作成したルールが表示されていることを確認します。

Step 4: Review and create

SELECT *, timestamp() as timestamp FROM 'iotdemo/topic/sensor'

Rule actions Info
Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. You can add up to 10 actions.

Action 1
CloudWatch logs
Send message data to CloudWatch logs. [Remove](#)

Log group name Info
sensor-visualize [Create CloudWatch Log group](#)

Batch mode
The payload that contains a JSON array of records will be sent to Cloud watch via a batch call.
☐ Use batch mode

IAM role
Choose a role to grant AWS IoT access to your endpoint.
sensor-visualize [View](#) [Create new role](#)
AWS IoT will automatically create a policy with a prefix of "aws-iot-rule" under your IAM role selected.

[Add rule action](#)

Error action - optional
You can optionally set an action that will be executed when something goes wrong with processing your rule. If two rule actions in the same rule fail, the error action receives one message that contains both errors.
[Add error action](#)

[Cancel](#) [Previous](#) [Next](#)

Step 3: Rule actions

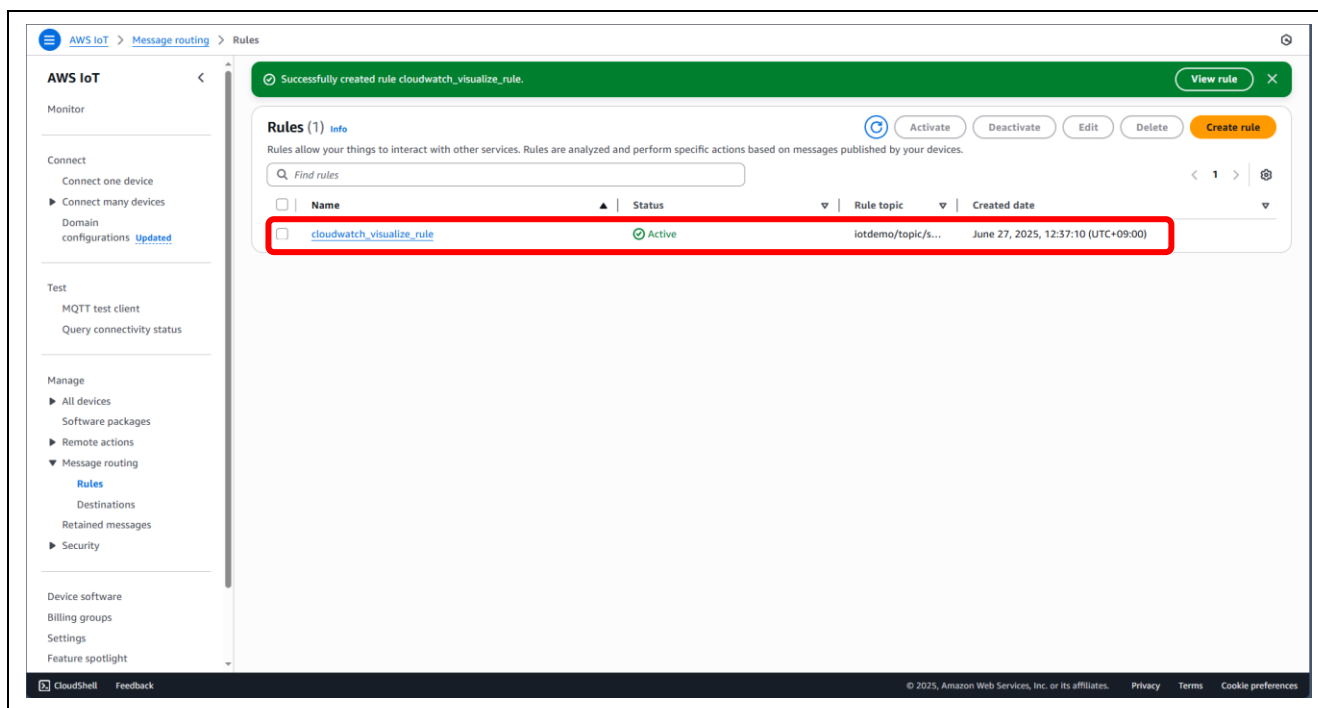
Rule properties
Name: cloudwatch_visualize_rule
Description: -

Step 2: SQL statement [Edit](#)
SQL version: 2016-03-23
SQL query: SELECT *, timestamp() as timestamp FROM 'iotdemo/topic/sensor'

Step 3: Rule actions [Edit](#)
Actions:
Log group name: sensor-visualize [View](#)
IAM role: arn:aws:iam::888695088161:role/service-role/sensor-visualize [View](#)
Batch mode: False

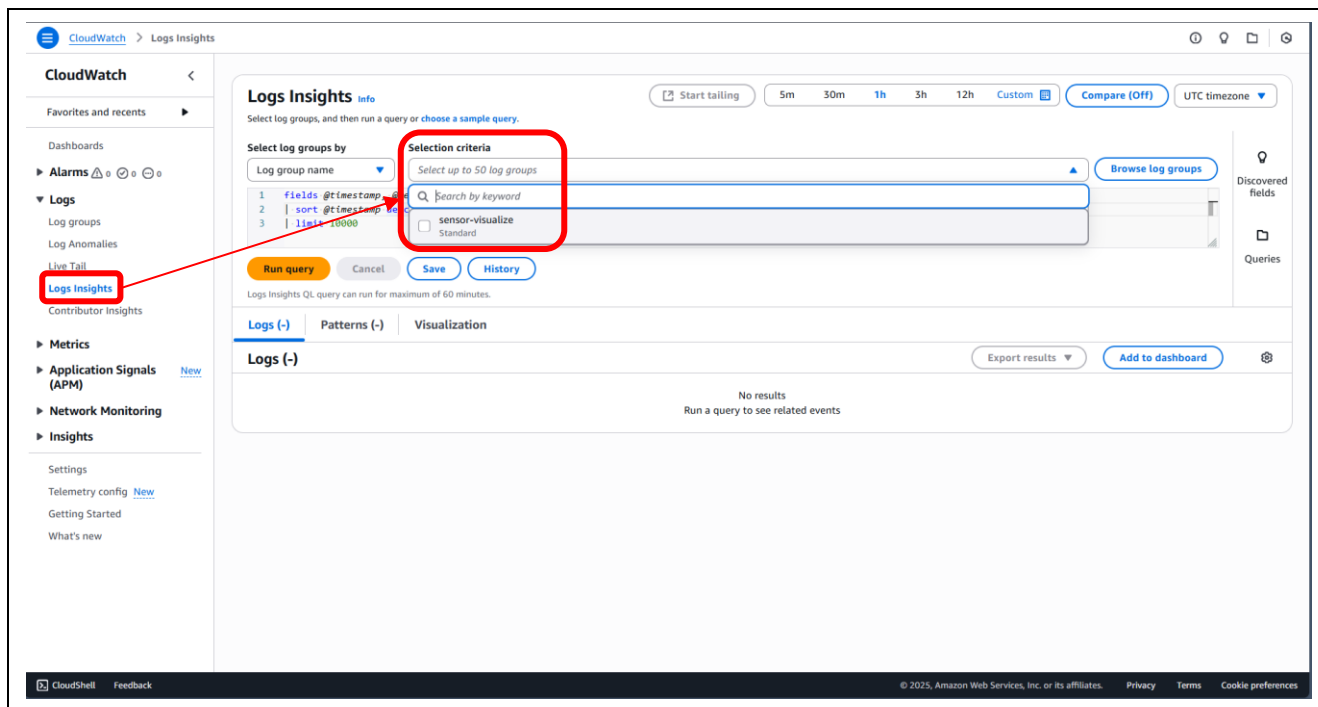
Error action
No error action

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



(10) CloudWatch でグラフ表示を確認

CloudWatch ページを表示し、左メニューの「ログのインサイト」をクリックします。ロググループの「選択基準」に先ほど作成したロググループ名を選択します。



以下のクエリを入力し、[クエリの実行]をクリックします。

```
stats avg(hs300x_humidity), avg(hs300x_temperature) by bin(1m)
```

可視化タブにグラフが表示されます。

The screenshot displays the AWS CloudWatch Logs Insights interface. On the left, the navigation pane shows 'Logs Insights' selected. The main panel is titled 'Logs Insights' and includes a 'Start tailing' button and a time range selector set to '5m'. Below this, the 'Select log groups by' section is empty, and the 'Selection criteria' section shows a query: `1 stats avg(hs300x_humidity), avg(hs300x_temperature) by bin(1m)`. A red box highlights this query, and a red arrow points from it to the 'Run query' button, which is also highlighted with a red box. To the right of the 'Run query' button are 'Cancel', 'Save', and 'History' buttons. Below the query, a status message indicates 'Completed. Query executed for 1 log group.' and a 'Visualization' tab is selected, highlighted with a red box. The 'Visualization' section shows a line graph with two data series: '1. avg(hs300x_humidity)' (blue line) and '2. avg(hs300x_temperature)' (orange line). The y-axis ranges from 20 to 60. The graph is currently empty, showing only the axes and the legend.

7. デモの実行手順

デモを実行する手順について説明します。

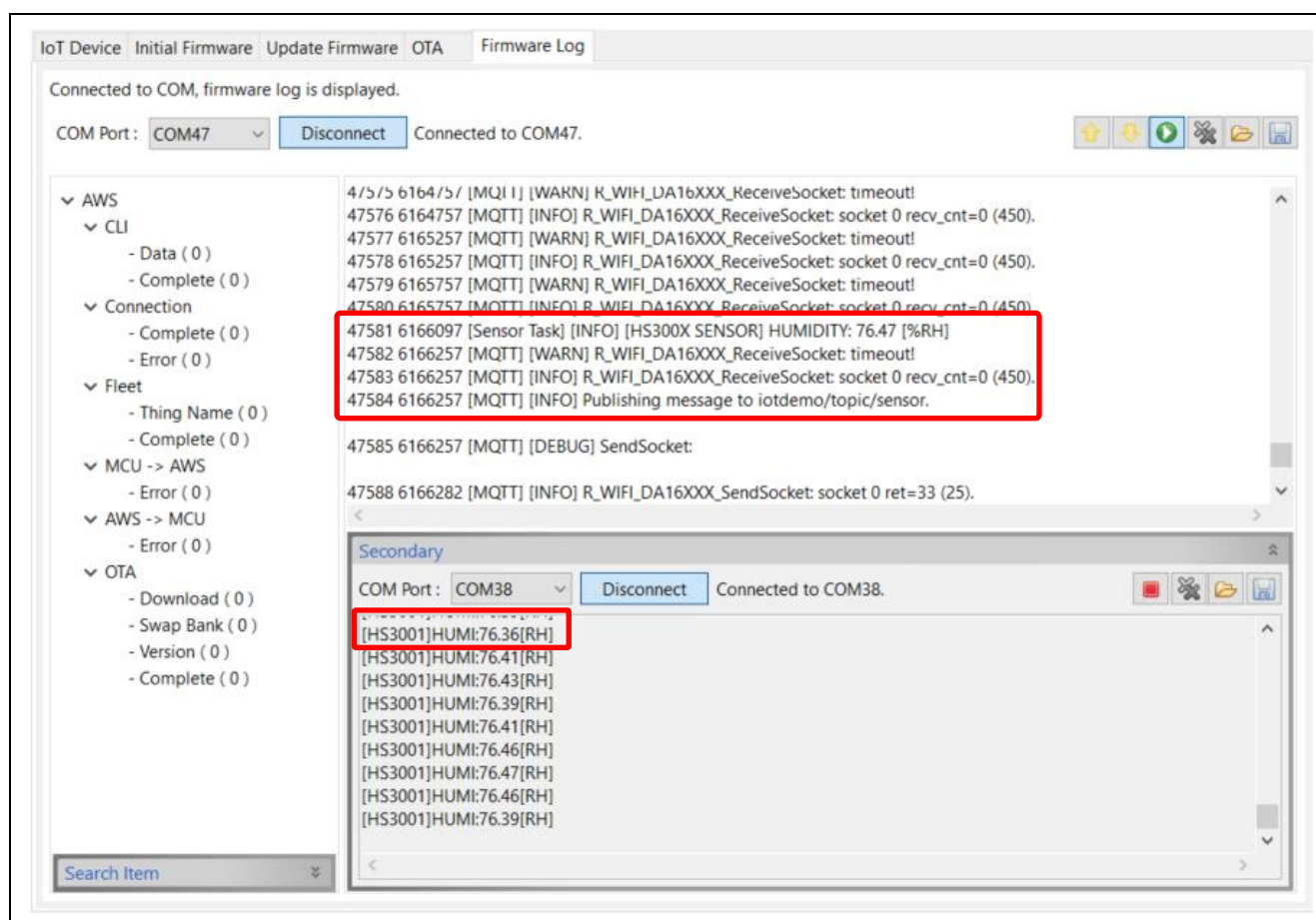
7.1 初期状態の動作確認

6 章のデモのセットアップが完了している状態で、QE for OTA の「Firmware Log」タブで各マイコンからのログを確認します。

CK-RX65N 側のログに、HS3001 センサの湿度データが出力されていることを確認します。また、その下には MQTT 通信でセンサデータを AWS へ送信しているログが表示されます。

次に、FPB-RX140 側のログに、HS3001 センサの湿度データのみ出力されていることを確認します。また、初期状態では FPB-RX140 の LED1 が点滅します。

正しく表示されていない場合は、FPB-RX140 のリセットスイッチ(RESET)を押下しハードウェアリセットを実行します。同様に CK-RX65N もリセットスイッチ(S1)を押下しハードウェアリセットを実行します。



最後に、Amazon CloudWatch の画面を示します。[クエリの実行]をクリックし、HS3001 センサから取得した湿度データがグラフ化されていることを確認します。



この状態がセカンダリ OTA アップデート実行前の初期状態です。

7.2 FPB-RX140 の OTA アップデートの実行

7.2.1 更新ファームウェアの作成

(1) demo_app_rx140_fpb_w_buffer プロジェクトのソースコードの変更

demo_app_rx140_fpb_w_buffer/src/fwupcomm_demo_main.h の MEASURE_TEMPERATURE マクロを(1)に変更します。

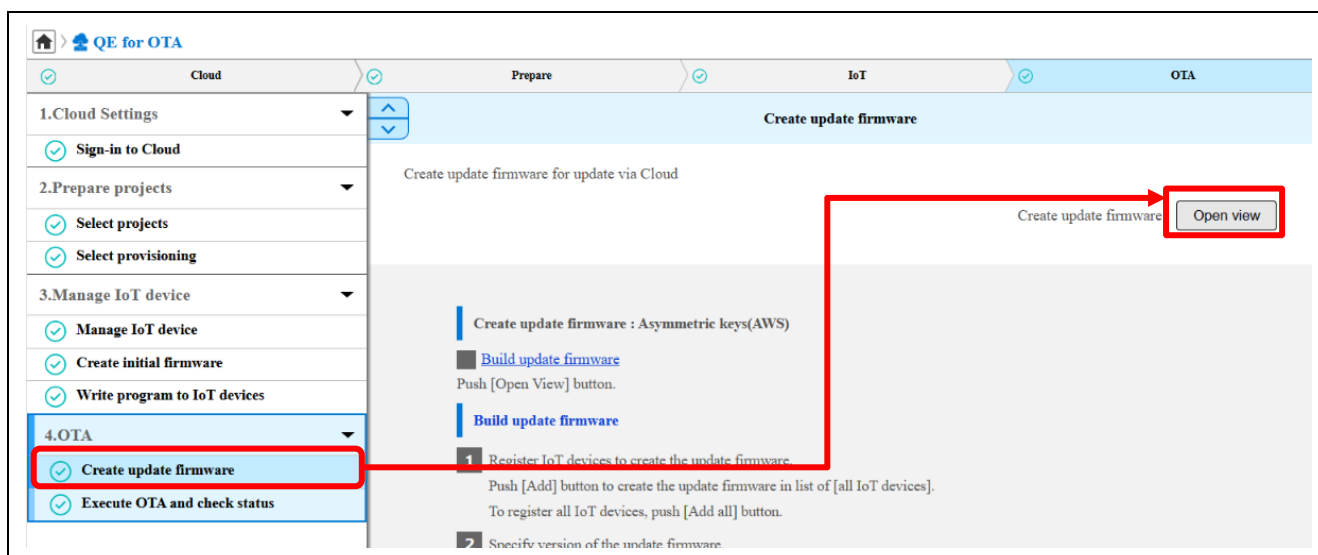
```

32      #define MEASURE_HUMIDITY          (1)
33      #define MEASURE_TEMPERATURE      (1)
34
35      #define DEMO_VER_MAJOR            (1)
36      #define DEMO_VER_MINOR           (0)
37      #define DEMO_VER_BUILD            (0)

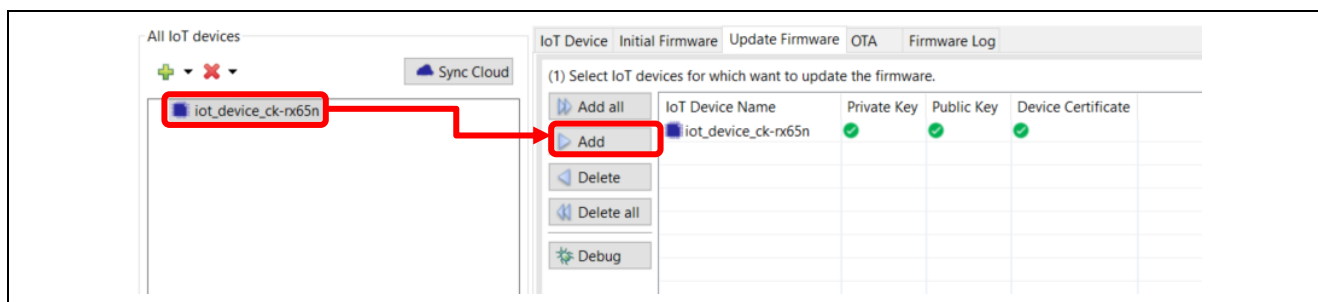
```

(2) 更新ファームウェアの作成

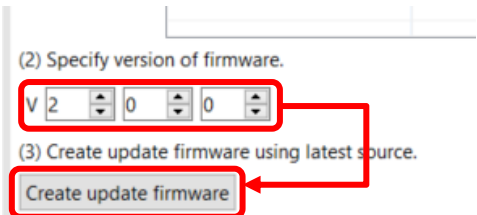
QE for OTA の「Create update firmware」の「Open view」をクリックします。



「Update Firmware タブ」で、6.2.3(5)で作成した IoT device を選択し、「▶Add」をクリックします。



バージョンに「1.0.0」を入力し、「Create update firmware」をクリックします。



(2) Specify version of firmware.

V 2 0 0

(3) Create update firmware using latest source.

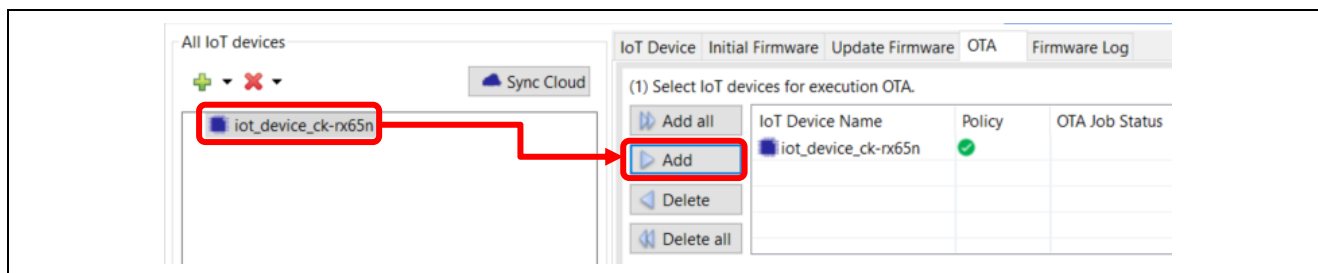
Create update firmware

demo_app_rx140_fpb_w_buffer プロジェクトのビルドが実行され、「Update Firmware」タブに「Create firmware is complete.」と表示されれば成功です。

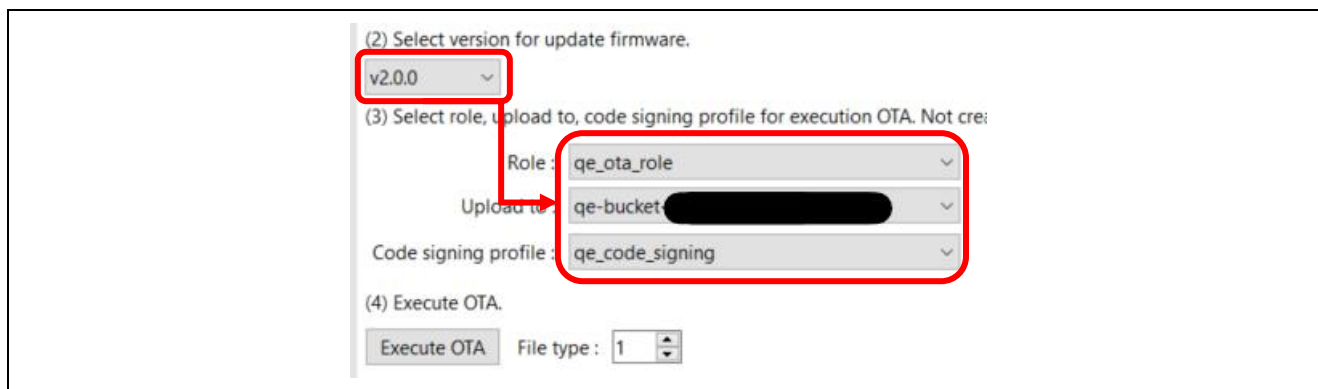
7.2.2 OTA ジョブの作成

(1) OTA ジョブの作成

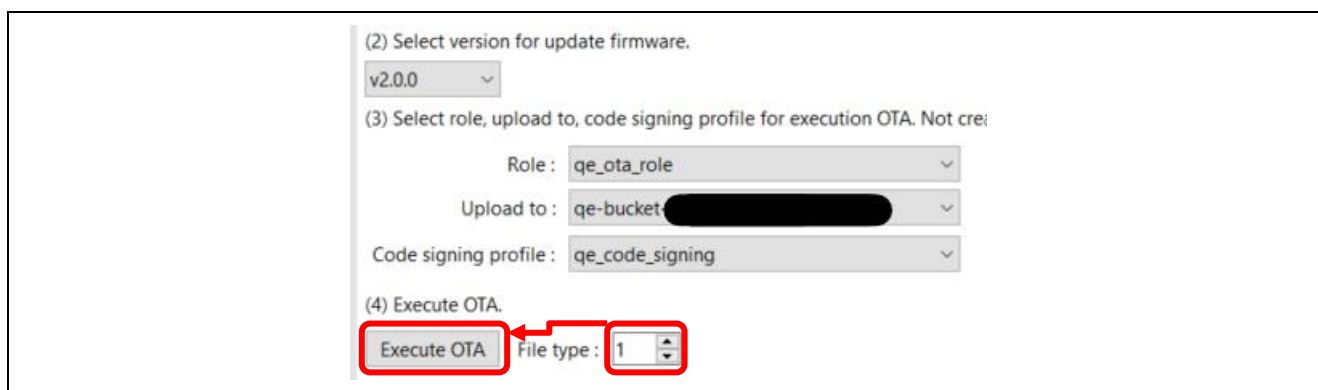
QE for OTA の「OTA タブ」を開き、6.2.3(5)で作成した IoT device を選択し、「▶Add」をクリックします。



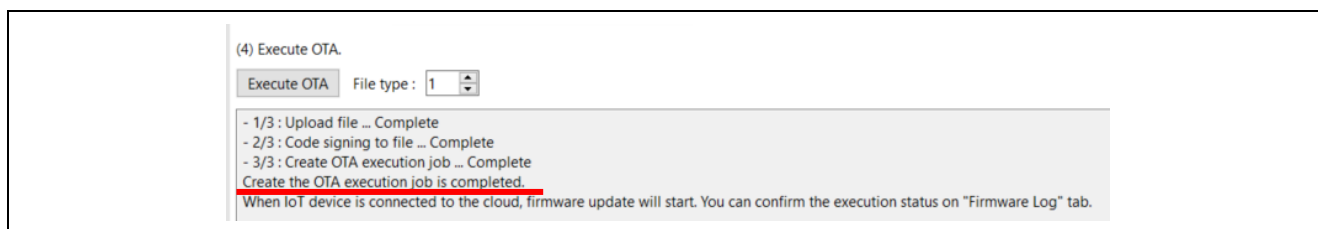
更新ファームウェアのバージョンは「v2.0.0」を選択し、ロール、アップロード先、コード署名プロファイルが入力されていることを確認します。



File type は「1」を入力し、「Execute OTA」をクリックします。

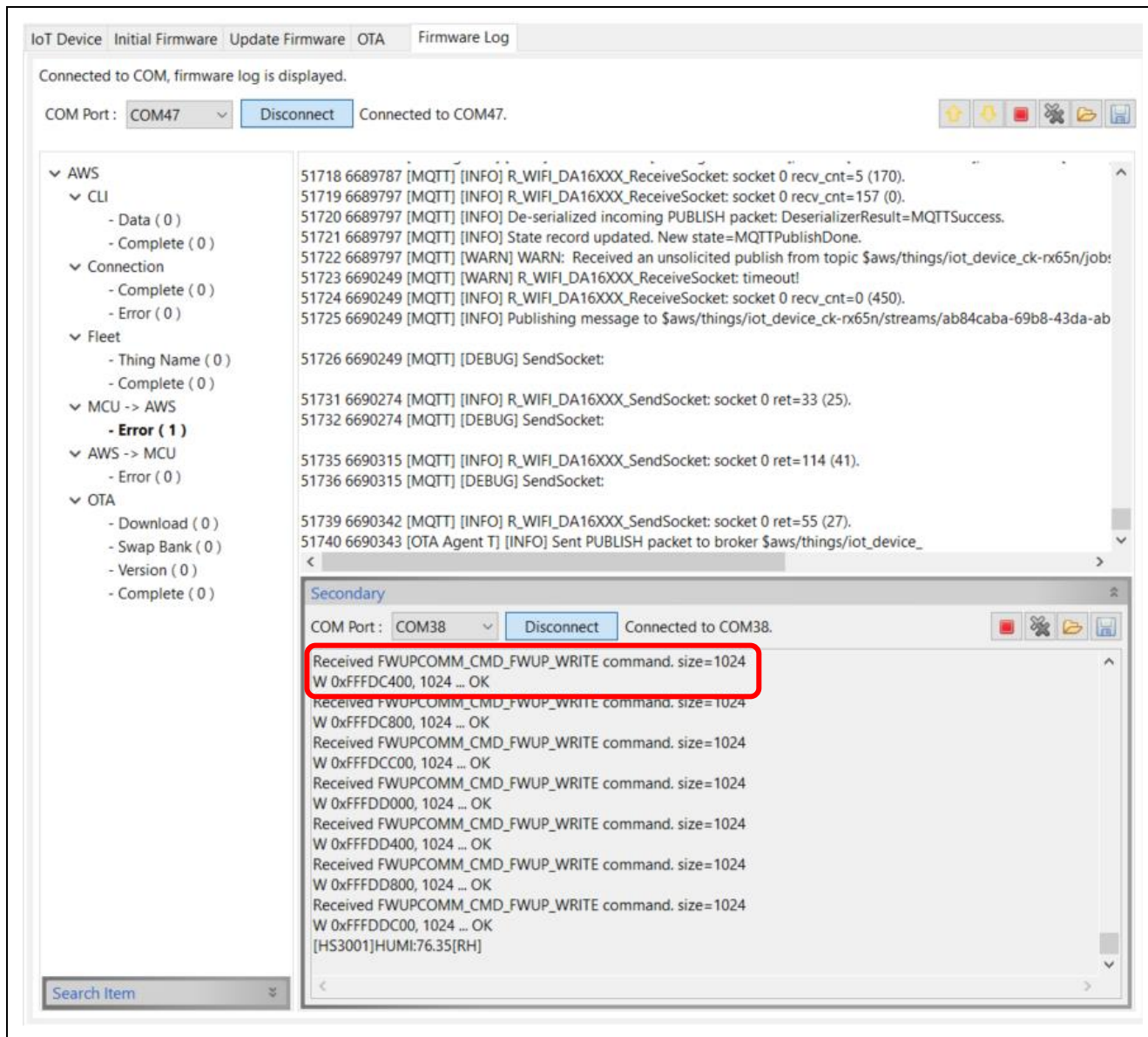


「Create the OTA execution job is completed.」と表示されれば成功です。以上の手順でセカンダリ OTA アップデート用の OTA ジョブが作成され、選択した IoT device に向けて OTA ジョブが配信されます。



7.2.3 セカンダリ OTA アップデート実行中の動作確認

ジョブの作成から数秒で OTA アップデートが始まります。CK-RX65N と FPB-RX140 の両方から、セカンダリ OTA アップデートの進捗がログ出力されます。



7.3 OTA アップデート後の動作確認

図 7-1 にアップデート後の CK-RX65N と FPB-RX140 のログ画面を示します。HS3001 センサで取得した湿度データに加えて、新たに温度データが表示されていることを確認できます。

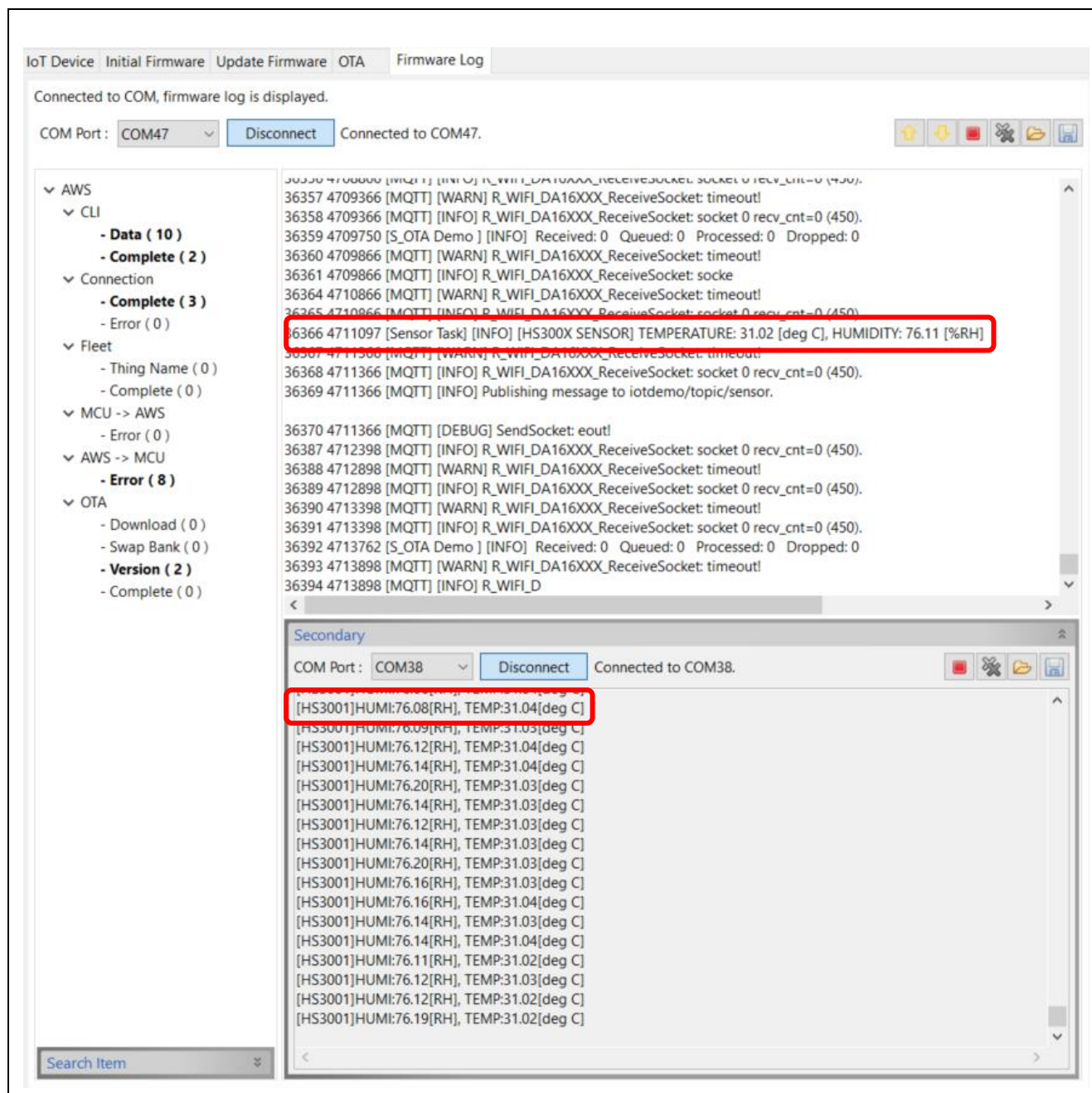


図 7-1 ファームウェア更新後のログ画面

また、FPB-RX140 では、初期状態で点滅していた LED1 に加えて LED2 も点滅します。

最後に、図 7-2 に Amazon CloudWatch の画面を示します。HS3001 センサから取得した湿度と温度の測定データがグラフ化されていることを確認します。



図 7-2 セカンダリ OTA アップデート後の Amazon CloudWatch のグラフ表示

デモ動作は以上となります。

※ 7.2.2(1)で、セカンダリ OTA アップデートの対象プロジェクトの更新方式が「リニアモードの半面更新方式」の場合は、File type の値を「1」に、「リニアモードの全面更新方式」の場合は、File type の値を「2」に設定してください。

8. センサを使用せずデモを実施する方法

FPB-RX140 に HS3001 センサを接続せずにセカンダリ OTA アップデートのデモを実施する方法を説明します。この場合、デモ内のセンサに関連する以下の内容は実施できなくなります。

- FPB-RX140 に接続された HS3001 センサから温湿度データを取得し、ログに出力する
- 温湿度データを CK-RX65N から AWS に送信し、AWS 上でグラフ表示する

8.1 デモ実施手順の変更箇所

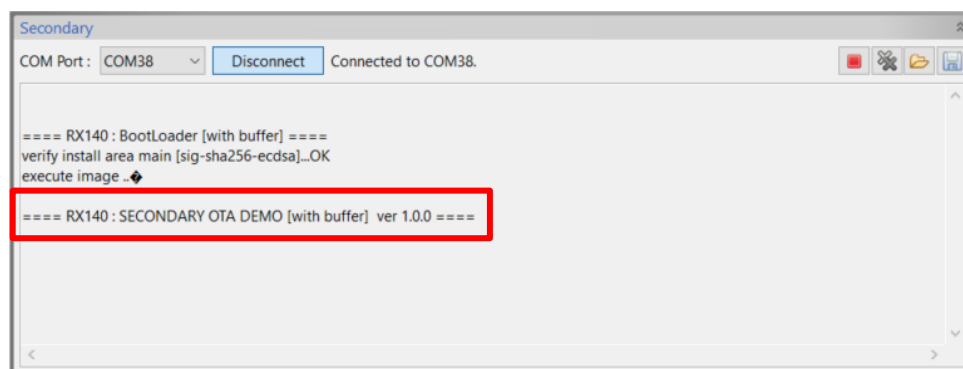
6~7 章のデモ実施手順のうち、以下の箇所を変更してください。それ以外の手順はセンサを使用する場合と同様です。

- (1) 「6.1.2(1) HS3001 ボードの接続」の手順を省略します。
- (2) 「6.2.4(2) FPB-RX140 用の初期ファームウェアの作成と実行 — プロジェクトの選択」を実施後に、demo_app_rx140_fpb_w_buffer/src/fwupcomm_demo_main.h の MEASURE_HUMIDITY マクロを(0)に変更します。
- (3) 「6.3 AWS クラウド上でセンサデータを可視化するための準備」の手順を省略します。
- (4) 「7.2.1(1) 更新ファームウェアの作成 — demo_app_rx140_fpb_w_buffer プロジェクトのソースコードの変更」の手順を省略します。

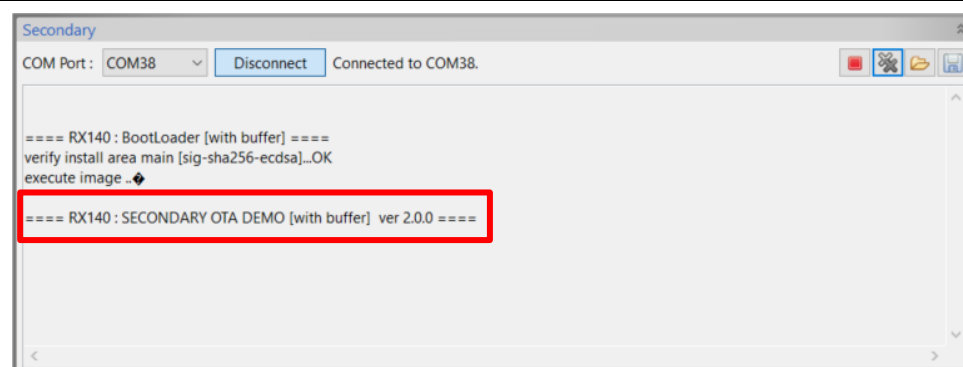
8.2 センサを使用しない場合のデモ動作の確認方法

センサを使用しない場合は、取得するセンサデータの種別でファームウェア更新が実行されたかどうかを確認できないため、FPB-RX140 起動時にログ上に表示されるバージョン情報で確認してください。

【更新前】



【更新後】



9. 注意事項

9.1 使用するオープンソースソフトウェアのライセンス情報

以下のオープンソースソフトウェアを使用しています。

- TinyCrypt Cryptographic Library
 - URL <https://github.com/intel/tinycrypt>
 - ライセンス <https://github.com/intel/tinycrypt/blob/master/LICENSE>
- FreeRTOS
 - URL <https://www.freertos.org/>
 - ライセンス [FreeRTOS open source licensing, FreeRTOS license description, FreeRTOS license terms and OpenRTOS commercial licensing options.](#)

9.2 デモを実施する AWS のリージョンとユーザ権限について

デモ実施の際の AWS の設定について、使用リージョンとユーザ権限についての注意事項を以下に示します。

【使用リージョンについて】

このデモは AWS の ap-northeast-1（アジアパシフィック（東京））リージョンで実施しています。
他のリージョンでこのデモを実施する場合は、デモで使用しているサービスがそのリージョンで提供されているか事前にご確認ください。

【ユーザ権限について】

このデモは AWS Identity and Access Management (IAM) で AdministratorAccess の権限が付与されたユーザで実行しています。そのため、各種サービスを使用する際の IAM での必要権限の付与に関して未記載です。

9.3 AWS の利用料金について

AWS の利用状況によっては、デモで作成・使用したクラウドリソースによって料金が発生する場合があります。意図しない課金を防ぐために、デモ実施後は作成したクラウド上のリソースを削除することをお勧めします。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2021/12/29	—	新規作成
1.01	2022/01/24	3	商標記載を目次ページに移動
		5	1.2.1 フォルダ／ファイル構成 を更新
		8	1.3 動作確認環境 IDE 欄を e2 studio 2021-07 に修正 表 1-3 デモ動作確認条件（センサボード） I/F 変換ボード欄 を見直し
		8	1.4 コードサイズ タイトルの誤記を修正
		9	1.5 1.5 デモ実施上 AWS のリージョンとユーザ権限について を追加
		10 - 12	2. ファームウェア書き込み 文章を見直し
		13	3.1 デモ概要 表 3-1 デモ内容概要 記載内容を見直し
		15	3.2.2 TB-RX23W ボードの構成 (1) RSK+RX65N-2MB ボード との UART 通信用の端子を接続 文章を見直し
		18	3.3.1 RSK+RX65N-2MB ボード用の初期ファームウェアの作成と書き込み 表 3-2 変更が必要なファイルの格納場所 記載 内容を見直し 表下のフォルダ記載箇所の情報を詳細に追記
		18	3.3.1 RSK+RX65N-2MB ボード用の初期ファームウェアの作成と書き込み (4) 手順 4：初期ファームウェアの結合 タ イトルの誤記を修正
		20	3.3.2 TB-RX23W ボード用の初期ファームウェアの作成と書き込み (2) 手順 2：公開鍵情報入力によるプロジェクトの 変更 タイトルに「公開鍵情報入力による」を追記
		20	3.3.2 TB-RX23W ボード用の初期ファームウェアの作成と書き込み (4) 手順 4：初期ファームウェアの結合 タイトル の誤記を修正
		21	3.3.4 AWS 側の設定 「AWS CLI を利用するための設定」を 「3.Python スクリプトを実行するための設定」に変更
		22 - 45	3.3.4 AWS 側の設定 (2) センサデータの可視化のための設定 説明を見直し
		46	3.3.4 AWS 側の設定 「(3) AWS CLI を利用するための設定」を「(3) Python スクリプトを実行するための設定」に 変更
		50	3.4.2 TB-RX23W ボードのファームウェアの 2nd OTA アップ デート実行 (2) 手順 2：Renesas Secure Flash Programmer でアップデート用のファームウェアを作成 内容を見直し
		50	3.4.2 TB-RX23W ボードのファームウェアの 2nd OTA アップ デート実行 (3) 手順 3：Python スクリプトを実行するた めに、PC 上でコマンドプロンプトを起動 タイトルを更新
		50	3.4.2 TB-RX23W ボードのファームウェアの 2nd OTA アップ デート実行 (4) 手順 4：コマンドプロンプト上で script フ ォルダをカレントフォルダに設定 タイトルを更新
		51	3.4.2 TB-RX23W ボードのファームウェアの 2nd OTA アップ デート実行 (5) 手順 5：2nd OTA アップデートを実行 表 3-7 コマンドに記述する入力値の参照先 USER_REGION 欄 を更新

		51 - 52	3.4.2 TB-RX23W ボードのファームウェアの 2nd OTA アップデート実行 (6) 手順 6 : 2nd OTA アップデート完了を確認 文章を見直し 注意 を追加
		53	3.4.3 HS3001 センサと ZMOD4410 センサのデータのクラウドへのアップロードとデータ可視化 文章を見直し
1.10	2022/03/31	—	AWS IoT Over-the-air Update Library v3.0.0 に対応
		5 - 7	RX65N 用プロジェクト変更につき、パッケージ及びフォルダ構成を修正
		5 - 7	各プロジェクトのフォルダ構成に.settings フォルダを追加
		9	動作確認環境の IDE を e2studio 2022-01 に更新、Toolchain を CC-RX V3.04.00 に更新、RX65N 用プロジェクトの FreeRTOS を Version 2021.07 に更新
		9	コードサイズを更新
		19 - 20	RX65N 用プロジェクト変更に伴い、初期ファームウェアの作成方法を更新
		47	RX65N 用プロジェクト変更に伴い、出力ログのスクリーンショットを更新
		51 - 55	OTA アップデートの実行方法を更新
		56	RX65N 用プロジェクト変更に伴い、出力ログのスクリーンショットを更新
2.00	2024/03/31	—	使用ボードを CK-RX65N と TB-RX660 に変更。
		—	RX65N 用のプロジェクトの FreeRTOS パッケージを更新
		—	RX660 用のプロジェクトの FWUP FIT のバージョンを更新
		—	使用ボードとプロジェクトの変更に伴い、アプリケーションノートを全面的に改訂
3.00	2025/06/30	—	使用ボードを CK-RX65N v2 と FPB-RX140 v1 に変更
		—	マイコン間通信制御をファームウェアアップデート通信モジュールに変更
		—	デモの実施手順を QE for OTA V2.2.0 のセカンダリ OTA 機能を使用した方法に変更

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、リセットしてください。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えば、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いたします。
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。