

## RL78/G23

### LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

#### はじめに

本ドキュメントは、ルネサス製マイコンボードとセルラーIoT モジュールを組み合わせ、Amazon Web Services（以降 AWS）に接続する方法を示します。

#### 関連文書

RL78/G23 ユーザーズマニュアル ハードウェア編 (R01UH0896)

RL78/G22, RL78/G23, RL78/G24 ファームウェア アップデート モジュール (R01AN6374)

RL78/G23-128p Fast Prototyping Board ユーザーズマニュアル (R20UT4870)

Renesas Flash Driver RL78 Type01 ユーザーズマニュアル (R20UT4830)

HS300x High Performance Relative Humidity and Temperature Sensor (R36DS0010)

HS40xx High-Performance Relative Humidity and Temperature Sensor with Digital Output (R36DS0036)

RL78/G23 LTE 通信での Amazon Web Services 接続 Getting Started Guide: RL78/G23-128p Fast Prototyping Board + FreeRTOS (R20AN0666)

#### 注意事項：当社での RYZ024A モジュールの取り扱いについて

当社は、RYZ024A 型名の既存 LTE モジュールの製造を中止し、製品の出荷を終了することを発表しました。

現在の設計または生産中に本製品を使用している場合、Sequans 社の製品型名 GM02S が、RYZ024A とピン及び機能コンパティビリティ代替品となります。本ドライバは、以下の組み合わせにて継続してご利用可能です。

- RYZ024A Cellular モジュール制御モジュール：Sequans 社 GM02S が互換のある代替モジュールとなります。

なお、RYZ024A の EOL 通知は [製品ページ内の本リンク](#) をご参照ください。

## 目次

1. 概要	4
1.1 デモプロジェクトの概要	4
1.2 動作確認条件	5
1.3 機材一覧	7
2. ハードウェア説明	8
2.1 システム構成	8
2.2 使用端子一覧	8
3. ソフトウェア説明	9
3.1 デモプロジェクト	9
3.2 オプション・バイトの設定一覧	9
3.3 フォルダ構成	10
3.4 コードサイズ	11
4. デモプロジェクトの動作説明	12
5. デモプロジェクトのセットアップ	12
5.1 ハードウェアのセットアップ	12
5.1.1 全体の構成	12
5.1.2 ハードウェアの接続方法	12
5.2 ソフトウェアのセットアップ	16
5.2.1 ターミナルソフトの設定	16
5.2.2 フラッシュライタのインストール	16
5.2.3 デモプロジェクトに SIM カード情報を追加	16
5.2.4 デモプロジェクトに AWS IoT 接続設定を追加	16
5.2.5 使用するセンサボードを選択	16
5.3 OTA アップデートのための準備	16
5.3.1 ツールのインストール	16
5.3.2 署名生成/検証用鍵の作成	16
5.4 AWS クラウドの準備	17
5.4.1 センサデータ可視化の設定	17
5.4.1.1 Amazon CloudWatch の設定	17
5.4.2 OTA アップデートの設定	28
5.4.2.1 Amazon S3 バケットの作成	28
5.4.2.2 OTA 更新用サービスロールの作成	28
5.4.2.3 OTA 更新用ユーザポリシーの作成	28
5.4.2.4 OTA 更新用ポリシーを IAM ユーザに割り当て	28
5.4.2.5 AWS IoT のコード署名へのアクセス権付与	28
5.5 初期イメージの作成	29
5.5.1 ブートローダを作成	29
5.5.1.1 ブートローダプロジェクトのインポート	29
5.5.1.2 ファームウェア検証用の鍵をブートローダプロジェクトに追加	29
5.5.1.3 ブートローダプロジェクトのビルド	29
5.5.2 初期アプリケーションを作成	30

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

5.5.2.1	初期アプリケーションのインポート	30
5.5.2.2	ファームウェア検証用の鍵を初期アプリケーションに追加	31
5.5.2.3	初期アプリケーションのビルド	31
5.5.3	Renesas Image Generator で初期イメージを生成	32
5.6	更新イメージの作成	33
5.6.1	更新アプリケーションを作成	33
5.6.1.1	アプリケーションのソースコードを変更	33
5.6.1.2	更新アプリケーションのビルド	33
5.6.1.3	更新アプリケーションの MOT ファイルをリネーム	34
5.6.2	Renesas Image Generator で更新イメージを生成	34
5.7	デモプロジェクトの実行	35
5.7.1	初期イメージ (initial_image.mot) をボードに書き込む	35
5.7.2	更新イメージ (aws_ryz024a_sensor_rl78g23-fpb_093.rsu) を OTA ジョブに登録	37
5.8	初期アプリケーションのデバッグ方法	45
6.	Renesas Flash Programmer の使用方法	46
7.	デバッグ手順	46
8.	付録	46
8.1	注意事項 : サードパーティ製ライブラリを RL78 に移植する際の注意点	46
8.1.1	Int の幅が 16bit	46
8.1.2	セクションのサイズ制限	46
8.2	デモプロジェクトで利用するオープンソースソフトウェアのライセンス情報	46
9.	ウェブサイトおよびサポート	47
	改訂記録	48

注 :

AWS™ は Amazon.com, Inc. or its affiliates の商標です。 (<https://aws.amazon.com/trademark-guidelines/>)

FreeRTOS™ は Amazon Web Services, Inc. の商標です。 (<https://freertos.org/copyright.html>)

GitHub® は GitHub, Inc. のトレードマークです。 (<https://github.com/logos>)

## 1. 概要

サンプルプログラム「[iot-reference-rl78](#)」は RL78 ファミリ、AWS、FreeRTOS を使用した IoT ソリューションのリファレンスを提供します。このサンプルプログラムはルネサスエレクトロニクスが提供する他の様々な製品と連携しており、AWS IoT のデモを簡単に試すことができます。

### 1.1 デモプロジェクトの概要

サンプルプログラムは下記のデモプロジェクトを含みます。これらのデモプロジェクトは、ルネサス製マイコンボード RL78/G23-128p Fast Prototyping Board とセルラーIoT モジュールを用いて AWS クラウドに接続する動作を実現します。

表 1-1 デモプロジェクトの一覧

デモプロジェクトの名称	説明
デモプロジェクト(PubSub)	MQTT 通信によるシンプルなデータアップロードを行う。
デモプロジェクト(OTA)	OTA によるファームウェアアップデートを行う。
デモプロジェクト(Sensor)	センサデータの取得、MQTT 通信、センサデータの可視化、OTA アップデートを組み合わせた動作を行う。

このアプリケーションノートはデモプロジェクト(Sensor)について説明します。以降、デモプロジェクト(Sensor)をデモプロジェクトと表します。その他のデモプロジェクトについては [Getting Started Guide \(R20AN0666\)](#) を参照してください。

デモプロジェクトの初期ファームウェアの動作は、温湿度センサボード(以降、センサボードと表します)から取得した温度データを MQTT 通信で AWS クラウドにアップロードするものです。アップロードされたセンサデータは AWS 提供の Amazon CloudWatch サービス(以降、CloudWatch と表します)にて可視化できます。次に、OTA アップデートによって初期ファームウェアを更新ファームウェアに更新し、更新ファームウェアを開始させます。更新ファームウェアの動作は、温度データに加え湿度データも取得およびアップロードする機能を追加したものです。

デモプロジェクトの概要は下記の章を参照してください。

- 2 ハードウェア説明
- 3 ソフトウェア説明

デモプロジェクトの実行方法は下記の章を参照してください。

- 5 デモプロジェクトのセットアップ

## 1.2 動作確認条件

デモプロジェクトは下記の条件で動作を確認しています。

表 1-2 動作確認条件 (RL78/G23)

項目	内容
使用マイコン	RL78/G23 (R7F100GSN CF 768KB)
使用ボード	RL78/G23-128p Fast Prototyping Board (RTK7RLG230CSN000BJ)
動作周波数	高速オンチップ・オシレータ・クロック: 32MHz
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 e <sup>2</sup> studio 2024-04
C コンパイラ	ルネサスエレクトロニクス製 CC-RL V1.12.01
ファームウェア書き込みツール	Renesas Flash Programmer V3.14.00
スマート・コンフィグレータ(SC)	Renesas Smart Configurator for RL78 24.1.0.v20231218-0132
ボードサポートパッケージ(BSP)	v1.60 (r_bsp)
フラッシュライブラリ	RL78/G2x 用 Renesas Flash Driver (RFD) RL78 Type01 V1.20 注: <a href="#">コードフラッシュライブラリ</a> -> <a href="#">RL78/G2x 用 Renesas Flash Driver RL78 Type01 パッケージ V1.20</a>
ファームウェアアップデートモジュール(FWUP)	RL78/G22,RL78/G23,RL78/G24 ファームウェア アップデート モジュール v2.01 ( <a href="#">アプリケーションノート</a> 、 <a href="#">サンプルコード</a> )
ファームウェアイメージ生成ユーティリティツール	Renesas Image Generator V3.03 注: ファームウェアアップデートモジュール(FWUP)に同梱
Python	Python 3.10.1
OpenSSL	OpenSSL 3.1.4

表 1-3 デモ動作確認条件(センサボード)

項目	内容
センサボード(HS3001)	US082-HS3001EVZ Board
センサボード(HS4001)	QCIOT-HS4001POCZ

表 1-4 動作確認条件 (その他、OSS ライブラリ等)

項目	内容
iot-reference-rl78	v202210.01-LTS-rl78-1.0.0 (Based FreeRTOS 202210.01-LTS) <a href="https://github.com/renesas/iot-reference-rl78/tree/v202210.01-LTS-rl78-1.0.0">https://github.com/renesas/iot-reference-rl78/tree/v202210.01-LTS-rl78-1.0.0</a>
FreeRTOS Cellular Interface	1.3.0 <a href="https://github.com/FreeRTOS/FreeRTOS-Cellular-Interface">https://github.com/FreeRTOS/FreeRTOS-Cellular-Interface</a>
FreeRTOS Kernel	10.5.1 <a href="https://github.com/FreeRTOS/FreeRTOS-Kernel">https://github.com/FreeRTOS/FreeRTOS-Kernel</a>
backoffAlgorithm	1.3.0 <a href="https://github.com/FreeRTOS/backoffAlgorithm">https://github.com/FreeRTOS/backoffAlgorithm</a>
coreJSON	3.2.0 <a href="https://github.com/FreeRTOS/coreJSON">https://github.com/FreeRTOS/coreJSON</a>
coreMQTT Client	2.1.1 <a href="https://github.com/FreeRTOS/coreMQTT">https://github.com/FreeRTOS/coreMQTT</a>
coreMQTT Agent	1.2.0 <a href="https://github.com/FreeRTOS/coreMQTT-Agent">https://github.com/FreeRTOS/coreMQTT-Agent</a>
AWS IoT Over-the-air Update	3.4.0 <a href="https://github.com/aws/ota-for-aws-iot-embedded-sdk">https://github.com/aws/ota-for-aws-iot-embedded-sdk</a>
tinycbor	0.5.2 <a href="https://github.com/intel/tinycbor">https://github.com/intel/tinycbor</a>
FreeRTOS-Plus network_transport	バージョンなし <a href="https://www.freertos.org/network-interface.html">https://www.freertos.org/network-interface.html</a>
Logging Interface	1.1.3 <a href="https://github.com/aws/amazon-freertos/tree/main/libraries/logging">https://github.com/aws/amazon-freertos/tree/main/libraries/logging</a>
TinyCrypt Cryptographic Library	0.2.8 <a href="https://github.com/intel/tinycrypt">https://github.com/intel/tinycrypt</a>

### 1.3 機材一覧

デモプロジェクトに必要な機材の一覧を示します。

表 1-5 機材一覧

項目	内容
マイコンボード	RL78/G23-128p Fast Prototyping Board <a href="#">RTK7RLG230CSN000BJ - RL78/G23-128p Fast Prototyping Board</a>
セルラーIoT モジュール	PMOD Expansion Board for RYZ024A (以降、RYZ024A と表します) <a href="#">RTKYZ024A0B00000BE - PMOD Expansion Board for RYZ024A</a> F/W ver: LR8.2
センサボード	US082-HS3001EVZ Board(以降、HS300x と表します) もしくは QCIOT-HS4001POCZ(以降、HS400x と表します)
SIM カード	LTE 通信が可能であること 例 : <a href="#">RTKYZ024A0B00000BE</a> 付属 SIM カード(注) - Truphone 製 <a href="#">DHA-SIM-132</a> - Nippon SIM 製
USB-UART 変換ボード	Pmod USBUART <a href="https://reference.digilentinc.com/reference/pmod/pmodusbuart/start">https://reference.digilentinc.com/reference/pmod/pmodusbuart/start</a>
Micro USB Type-B ケーブル x3	・ USB-UART 変換ボードと PC を接続 ・ マイコンボードと PC を接続 ・ RYZ024A の電源供給用
ジャンパワイヤ x3	USB-UART 変換ボード と マイコンボードを接続
ジャンパピン x3	マイコンボードの電源選択用(J15,J16,J19)

注 :

PMOD Expansion Board for RYZ024A ([RTKYZ024A0B00000BE](#)) に付属の Truphone SIM カードをご使用の場合は、SIM カードの Activation が必要です。  
「[RA6M5 Group RYZ024A PMOD LTE Connectivity with RA6M5 MCU Quick Start Guide](#)」  
(R21QS0007) を参照し SIM カードの Activation を行ってください。

#### 機材接続の全体図

各デモの機材接続の全体図は下記を参照してください。

- デモプロジェクト : 図 5-1 本デモプロジェクトのハードウェア全体構成

#### デバッグ用機材に関する注意事項

本デモプロジェクトは COM port デバッグを使用しますが、エミュレータを用いてデバッグをすることも可能です。

エミュレータを使用する場合、エミュレータ接続用コネクタの実装と回路変更が必要です。詳細は「[Getting Started Guide](#) の 7.2.1 Setting Jumper Pins, Mounting the Connector, and Cutting Patterns」もしくは以下を参照してください。

表 1-6 デバッグ機材の一覧

項目	内容
エミュレータ	E2 エミュレータ Lite <a href="https://www.renesas.com/software-tool/e2-emulator-lite-rte0t0002lkce00000r">https://www.renesas.com/software-tool/e2-emulator-lite-rte0t0002lkce00000r</a>

## 2. ハードウェア説明

### 2.1 システム構成

デモプロジェクトのシステム構成を示します。

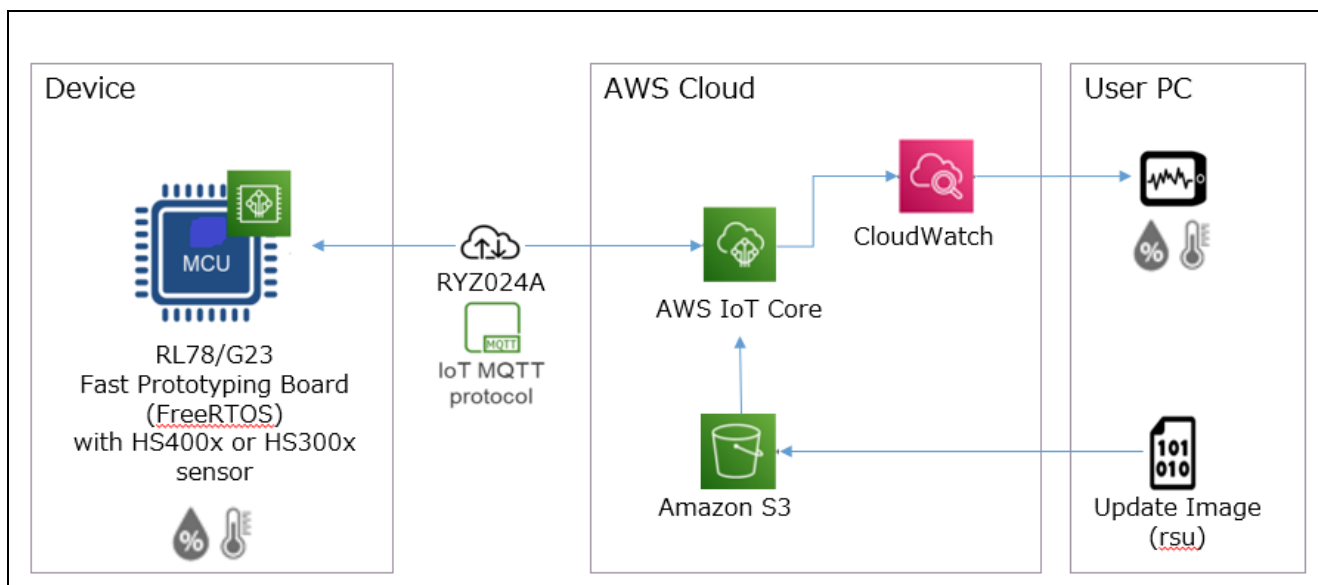


図 2-1 デモプロジェクトのシステム構成

### 2.2 使用端子一覧

デモプロジェクトで使用する端子と機能を示します。

表 2-1 デモプロジェクトの使用端子と機能

端子名	入出力	内容
P143/RxD3	入力	RYZ024A との UART 通信(受信)
P144/TxD3	出力	RYZ024A との UART 通信(送信)
P00	出力	RYZ024A へのリセット
P142	出力	RYZ024A との UART 通信(RTS)
P14/RxD2	入力	ターミナル入力
P13/TxD2	出力	ターミナル出力
P50	出力	LED1
P51	出力	LED2
P62/SCLA1	入出力	センサボードとの I2C 通信(クロック)
P63/SDAA1	入出力	センサボードとの I2C 通信(データ)



### 3. ソフトウェア説明

#### 3.1 デモプロジェクト

[Getting Started Guide](#) の 3.2.1 Demo Project Structure を参照してください。

#### 3.2 オプション・バイトの設定一覧

オプション・バイトの設定を示します。

表 3-1 オプション・バイト設定

アドレス	設定値	内容
000C0H/040C0H	11101111B	ウォッチドッグ・タイマ 動作停止 (リセット解除後、カウント停止)
000C1H/040C1H	00111010B	LVDO オフ ( RESET 端子により外部リセットを使用)
000C2H/040C2H	11101000B	HS (高速メイン) モード & 高速オンチップ・オシレータ・クロック (fIH): 32MHz
000C3H/040C3H	10000100B	オンチップ・デバッグ許可

### 3.3 フォルダ構成

サンプルプログラムのフォルダ構成を以下に示します。

表 3-2 サンプルプログラムのフォルダ構成

Folder Name	Description
iot-reference-rl78	The sample program described in this Getting Started Guide.
├─Common	
└─FreeRTOS_common	
└─ports	
└─ota_pal	
├─Configuration	
└─rl78g23-fpb	
└─sensor	Sensor demo configurations.
├─Demos	
└─common	
└─include	
└─mqtt_agent	
└─OtaOverMqtt	OTA demo source codes.
└─Sensor	Sensor demo source codes.
└─SimplePubSub	PubSub demo source codes.
├─IDT_config	
├─Middleware	
└─3rdparty	
└─Application-Protocols	
└─network_transport	
└─AWS	
└─ota-for-aws-iot-embedded-sdk	
└─FreeRTOS	FreeRTOS Kernel and libraries.
└─backoffAlgorithm	
└─coreJSON	
└─coreMQTT	
└─coreMQTT-Agent	
└─FreeRTOS-Cellular-Interface	
└─FreeRTOS-Kernel	
└─logging	
├─Projects	
└─rl78g23-fpb	
└─application_code	
└─flash_proj	
└─helper	
└─modules	
└─projects	Import below folders to IDE.
└─aws_ryz024a_rl78g23-fpb	PubSub demo and OTA demo. Select by Build Configurations.
└─aws_ryz024a_sensor_rl78g23-fpb	Sensor demo (including OTA demo).
└─boot_loader	Boot loader for OTA demo.
└─test_aws_cellular_ryz024a	
└─rtos_skelton	
├─Test	
└─Tools	

### 3.4 コードサイズ

以下の条件におけるデモプロジェクトの ROM、RAM サイズを下表に示します。

- CC-RL
  - コンパイル・オプション
    - -Odefault : オブジェクト・サイズと実行速度の両方に効果のある最適化
  - リンク・オプション
    - -optimize=symbol\_delete : 一度も参照のない変数／関数を削除

表 3-3 デモプロジェクトの ROM、RAM サイズ

デモプロジェクト名	ROM (byte)	RAM (byte)
aws_ryz024a_sensor_rl78g23-fpb	233229	43204
boot_loader	22147	1348

## 4. デモプロジェクトの動作説明

デモプロジェクトの動作を以下に示します。

- ① 初期ファームウェアはセンサボードから取得した温度データと温度ワーニング(25度未満は False、25度以上は True)を AWS クラウドにアップロードします。
- ② OTA アップデートで更新ファームウェアをダウンロードしファームウェアを更新します。
- ③ ファームウェア更新後は温度データと温度ワーニングに加え湿度データを AWS クラウドにアップロードします。アップロードする温度データ、温度ワーニング、湿度データはログと CloudWatch で確認できます。
- ④ 以下のいずれかの条件を満たした場合、AWS クラウドへのアップロードを終了します。
  - アップロード回数が 2000 回に到達した場合
  - センサデータ取得に失敗した場合

## 5. デモプロジェクトのセットアップ

デモプロジェクトのセットアップについて説明します。

### 5.1 ハードウェアのセットアップ

#### 5.1.1 全体の構成

最初に、本デモプロジェクトを構成するハードウェア全体の構成を示します。

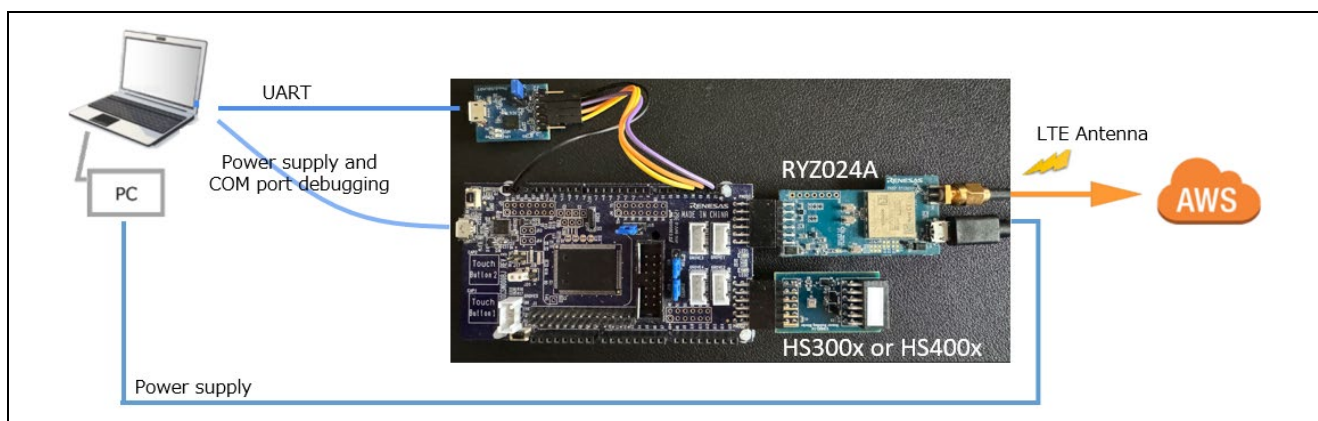


図 5-1 本デモプロジェクトのハードウェア全体構成

#### 5.1.2 ハードウェアの接続方法

ハードウェアの接続方法を以下に示します。

- ① アクティベート済の SIM カードを RYZ024A に挿入します。

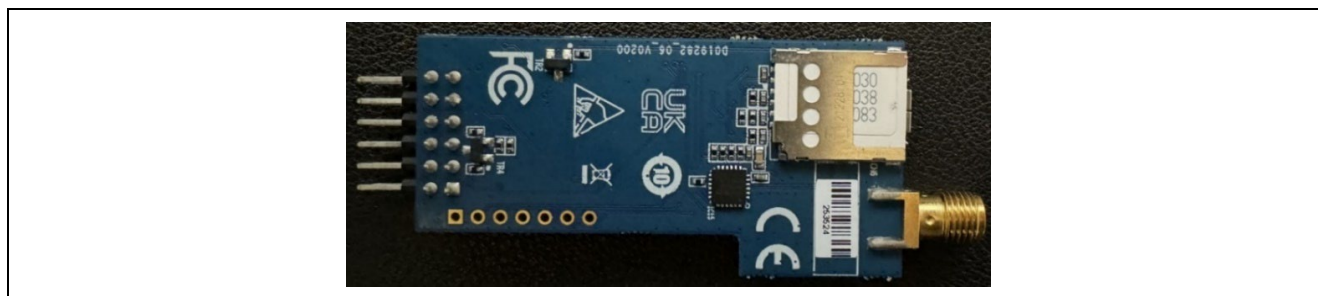


図 5-2 アクティベート済の SIM カードを RYZ024A に挿入

RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法:  
RL78/G23-128p Fast Prototyping Board + FreeRTOS

- ② アンテナと電源供給用の USB ケーブルを RYZ024A に接続します。

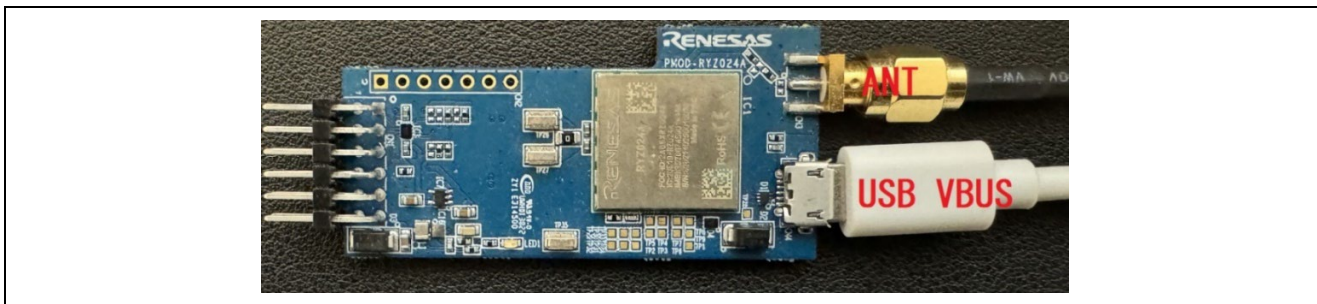


図 5-3 アンテナと電源供給用の USB ケーブルを RYZ024A に接続

- ③ RYZ024A をマイコンボードの PMOD1 に接続します。

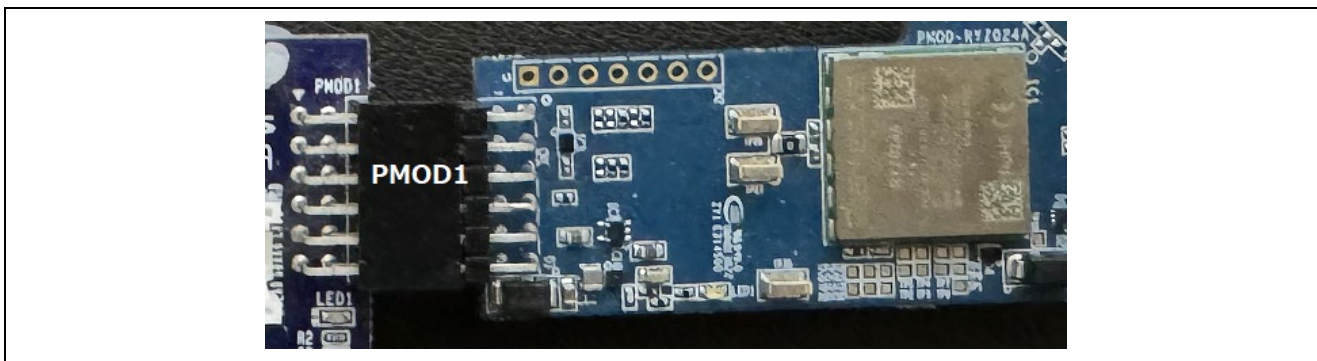


図 5-4 RYZ024A をマイコンボードの PMOD1 に接続

- ④ センサボードをマイコンボードの PMOD2 に接続します。

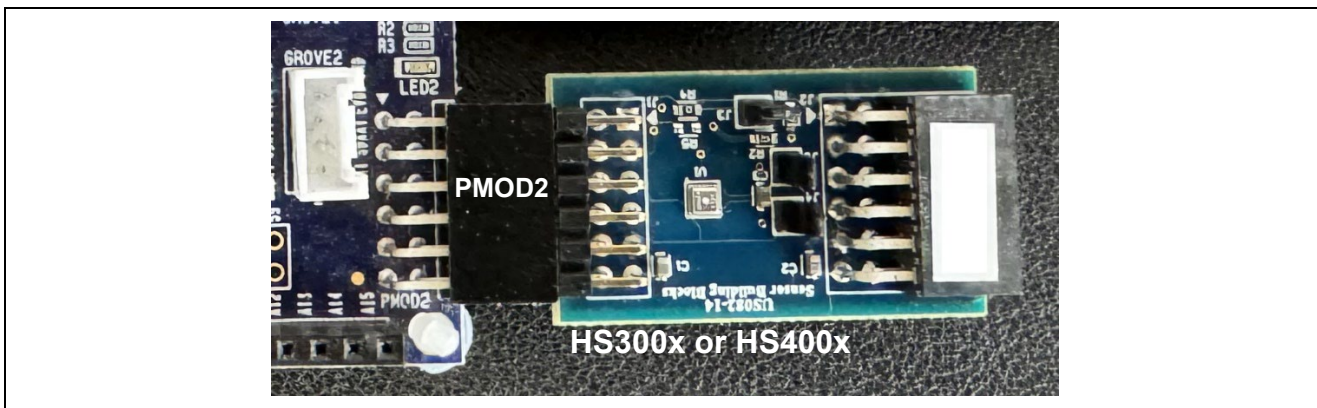


図 5-5 センサボードをマイコンボードの PMOD2 に接続



## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

- ⑤ USB-UART 変換ボードをマイコンボードに接続します。

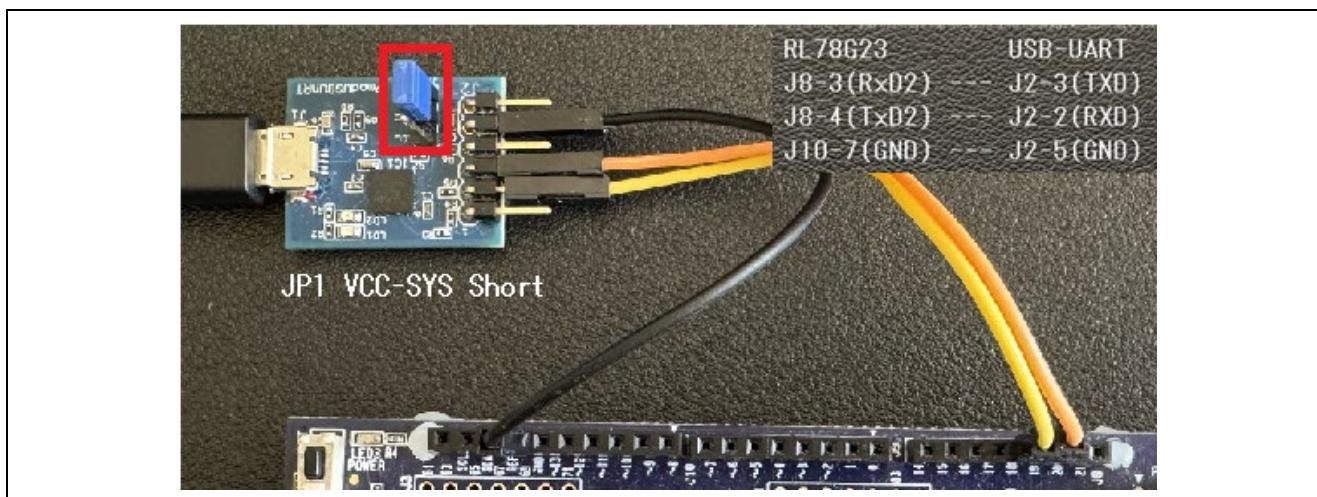


図 5-6 マイコンボードに USB-UART 変換ボードを接続

- ⑥ マイコンボードの電源選択ヘッダを J20 2-3 ショートに設定し 3.3V 電源を選択します。

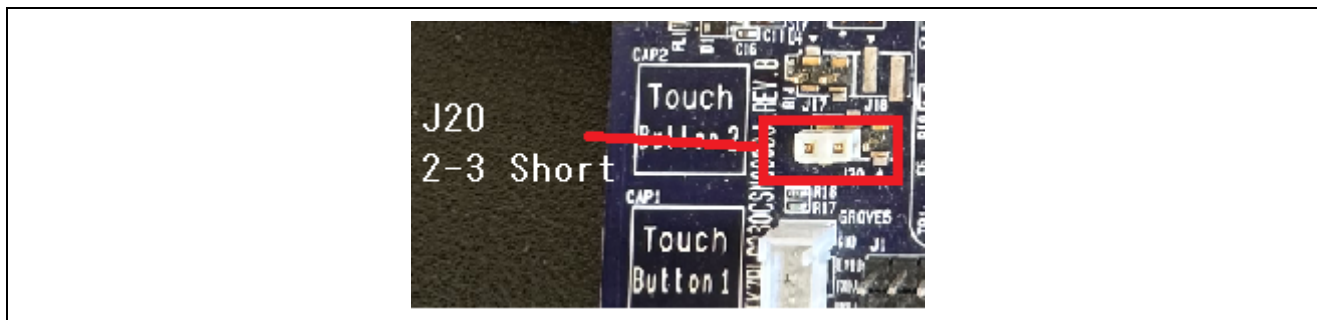


図 5-7 マイコンボード電源を 3.3V に設定

- ⑦ マイコンボードにエミュレータ用コネクタを実装するための回路変更を実施している場合、USB シリアル変換器を使用した COM port デバッグの設定にします。  
回路変更を実施していない場合、この操作は不要です。

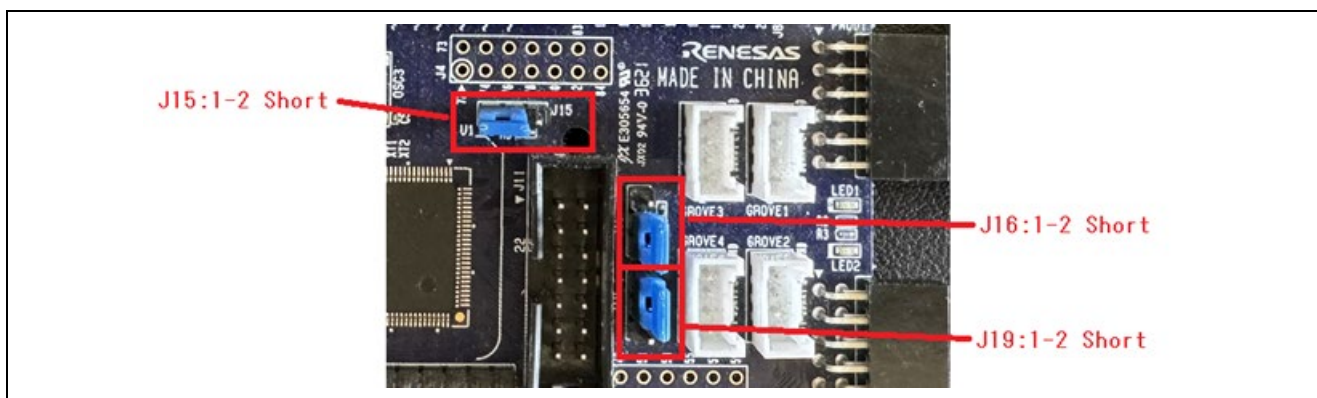


図 5-8 COM port デバッグ使用時設定(部品面)

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

- ⑧ USB ケーブルを接続してマイコンボードへ電源を供給します。

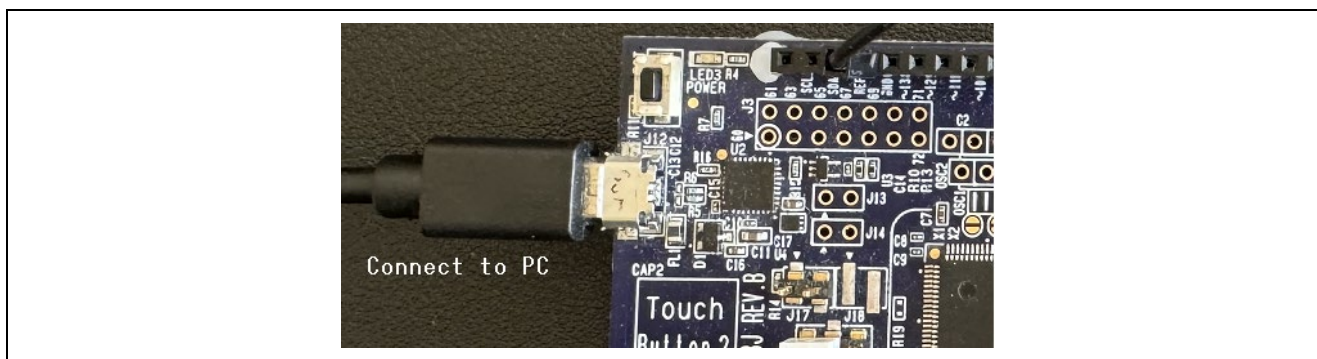


図 5-9 マイコンボードに電源を供給

- ⑨ Windows PC のデバイスマネージャーなどを利用して COM port 番号を確認します。  
ファームウェア書き込みおよびデバッグで COM port 番号を使用します。
- ⑩ USB ケーブルを抜いてマイコンボードへの電源供給を一旦停止します。

## 5.2 ソフトウェアのセットアップ

### 5.2.1 ターミナルソフトの設定

デモプロジェクトのログ出力時にターミナルソフト(例: Tera Term 等)が必要です。以下にシリアルポートの設定を示します。

表 5-1 シリアルポート設定

項目	内容
ボーレート	115200 bps
データ	8 bit
パリティ	なし
ストップビット	1 bit
フロー制御	なし

### 5.2.2 フラッシュライタのインストール

初期イメージの書き込みに使用します。

[Renesas Flash Programmer \(Programming GUI\)](#)

### 5.2.3 デモプロジェクトに SIM カード情報を追加

プロジェクトをインポート後に SIM カード情報を追加します。

[Getting Started Guide](#) の 4.2.3 Adding SIM Card Information to the Demo Project を参照してください。

### 5.2.4 デモプロジェクトに AWS IoT 接続設定を追加

プロジェクトをインポート後に AWS IoT 接続設定を追加します。

[Getting Started Guide](#) の 4.2.4 Adding AWS IoT Connection Settings to the Demo Project を参照してください。

### 5.2.5 使用するセンサボードを選択

使用するセンサボードに合わせて「SENSOR\_CFG\_TYPE」マクロの定義を変更します。

```
iot-reference-rl78\Projects\rl78g23-fpb\projects\aws_ryz024a_sensor_rl78g23-  
fpb\src\application_code\sensor.h  
HS400x の場合  
#define SENSOR_CFG_TYPE (SENSOR_TYPE_HS400X)  
  
HS300x の場合  
#define SENSOR_CFG_TYPE (SENSOR_TYPE_HS300X)
```

## 5.3 OTA アップデートのための準備

### 5.3.1 ツールのインストール

[Getting Started Guide](#) の 6.1.1 Installing Tools を参照してください。

### 5.3.2 署名生成/検証用鍵の作成

[Getting Started Guide](#) の 6.1.2 Generating Keys for Signature Generation and Verification を参照してください。



## 5.4 AWS クラウドの準備

### 5.4.1 センサデータ可視化の設定

センサボードから受信した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。まずは、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ログイン時の設定と同じリージョンを選択します。

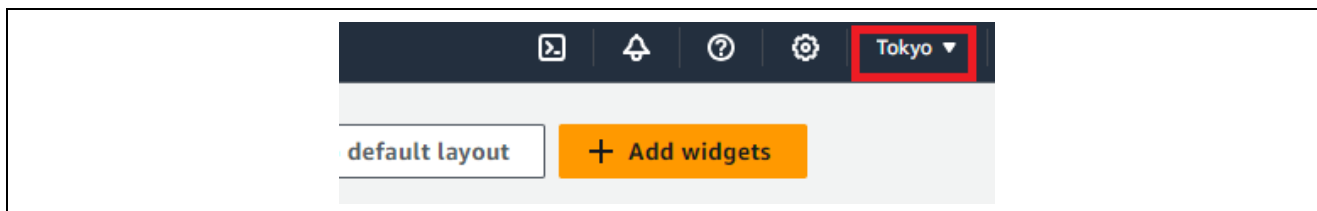


図 5-10 リージョンの確認

#### 5.4.1.1 Amazon CloudWatch の設定

##### (1) AWS IoT でルールを作成

- AWS IoT > Message routing > Rules > “Create rule”をクリック

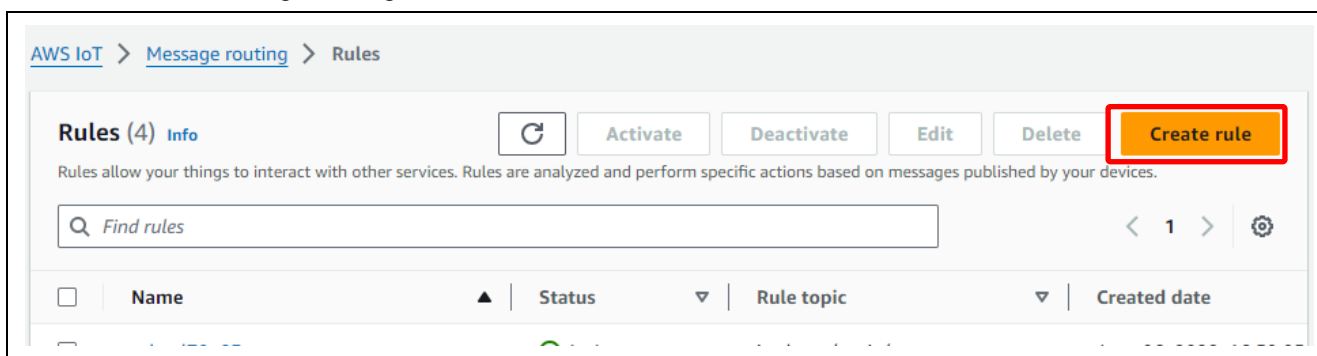


図 5-11 AWS IoT でルールを作成

(2) ルールのプロパティを指定

- Rule name を入力し “Next” をクリック

AWS IoT > Message routing > Rules > Create rule

Step 1  
Specify rule properties

Step 2  
Configure SQL statement

Step 3  
Attach rule actions

Step 4  
Review and create

### Specify rule properties Info

A rule resource contains a list of actions based on the MQTT topic stream.

#### Rule properties

Rule name

ro...e

Enter an alphanumeric string that can also contain underscore ( `_` ) characters, but no spaces.

Rule description - *optional*

Enter a description to provide additional details about the rule to others.

A description of your new rule

▼ Tags - *optional*

No tags associated with the resource.

Add new tag

You can add up to 50 tags.

Cancel **Next**

図 5-12 Rule Name を入力し “Next” をクリック

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

### (3) SQL ステートメントを設定

- SQL statement に以下を入力し“Next”をクリック

```
SELECT *, timestamp() as timestamp FROM '[thing-name]_tempdata'  
(改行が必要)
```

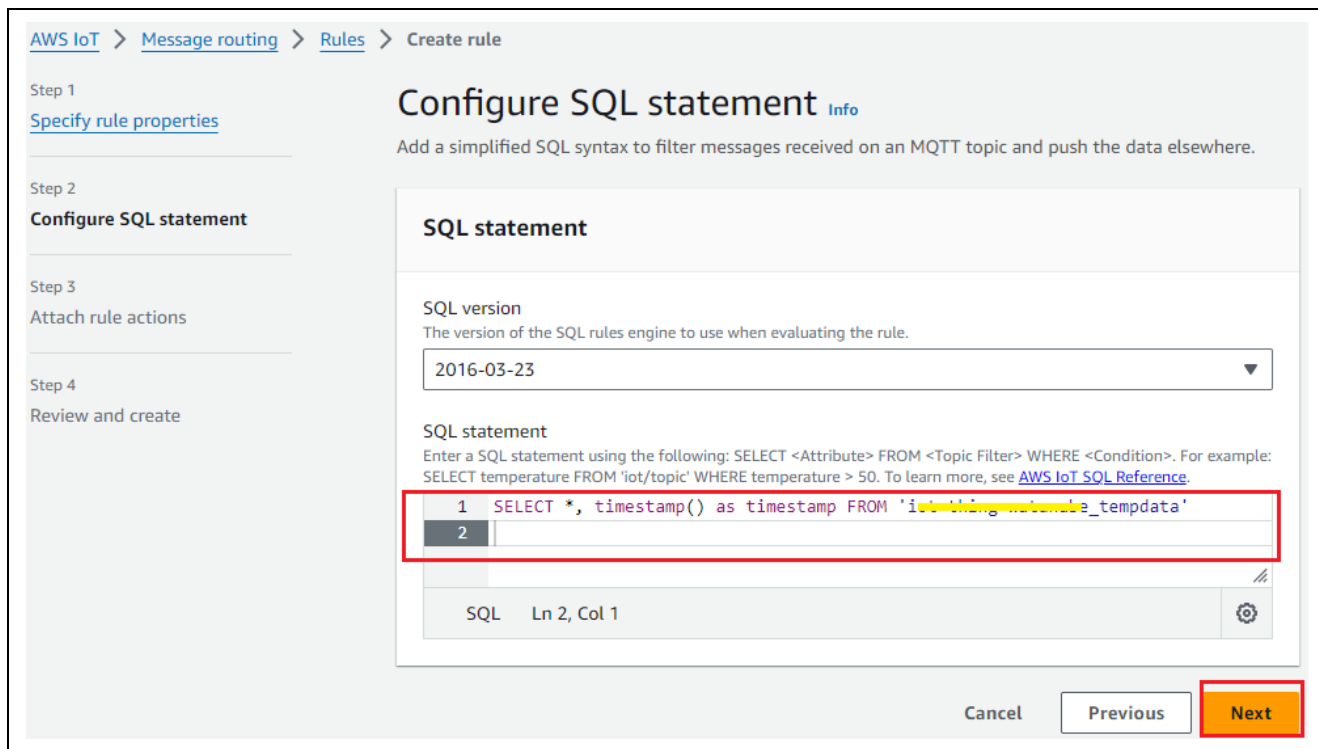


図 5-13 SQL statement を入力し“Next”をクリック

(4) ルールアクションを設定

- Action1 に”CloudWatch logs”を選択し ”Create CloudWatch Log group”をクリック

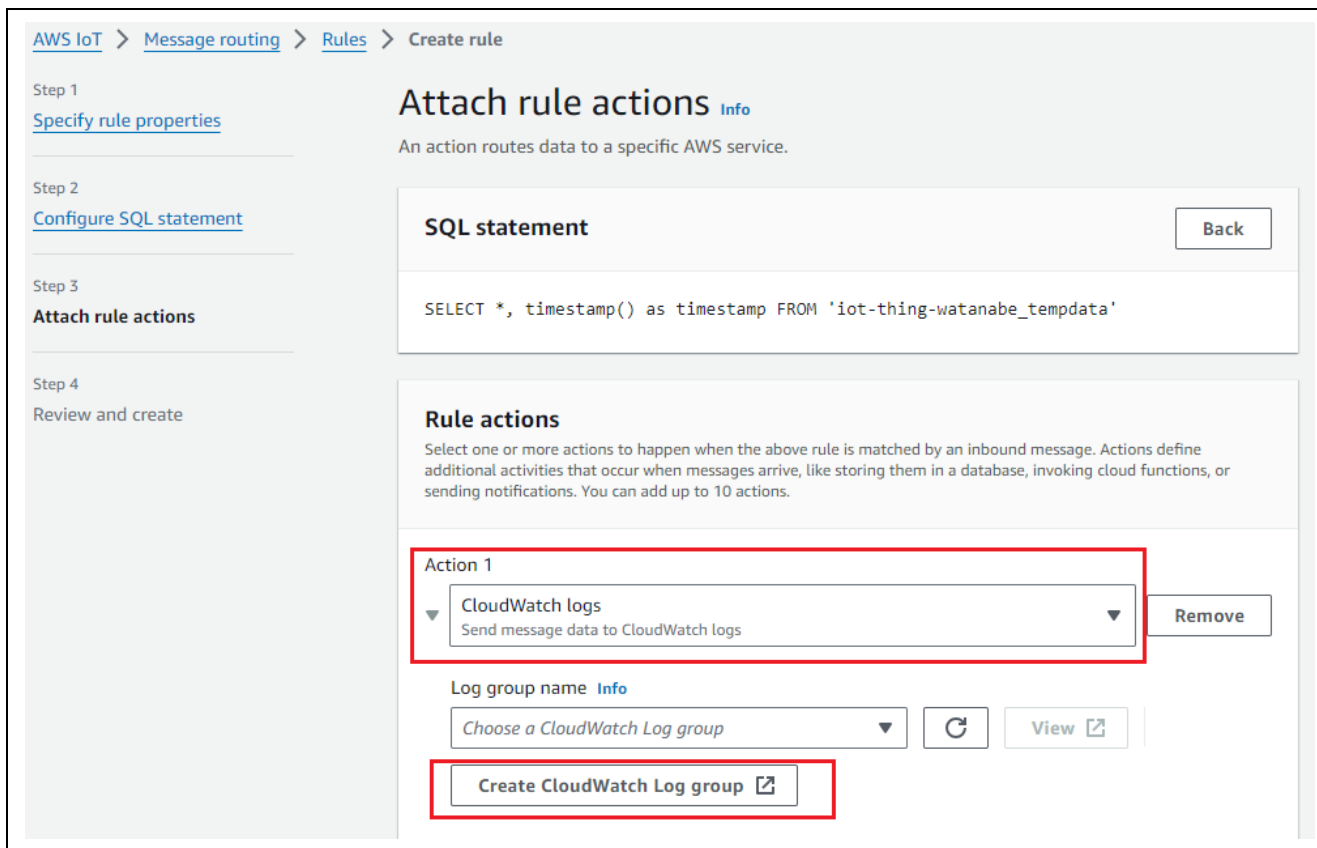


図 5-14 Action1 に”Cloudwatch logs”を選択

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

- 新規の CloudWatch タブが起動し Create log group の画面であることを確認

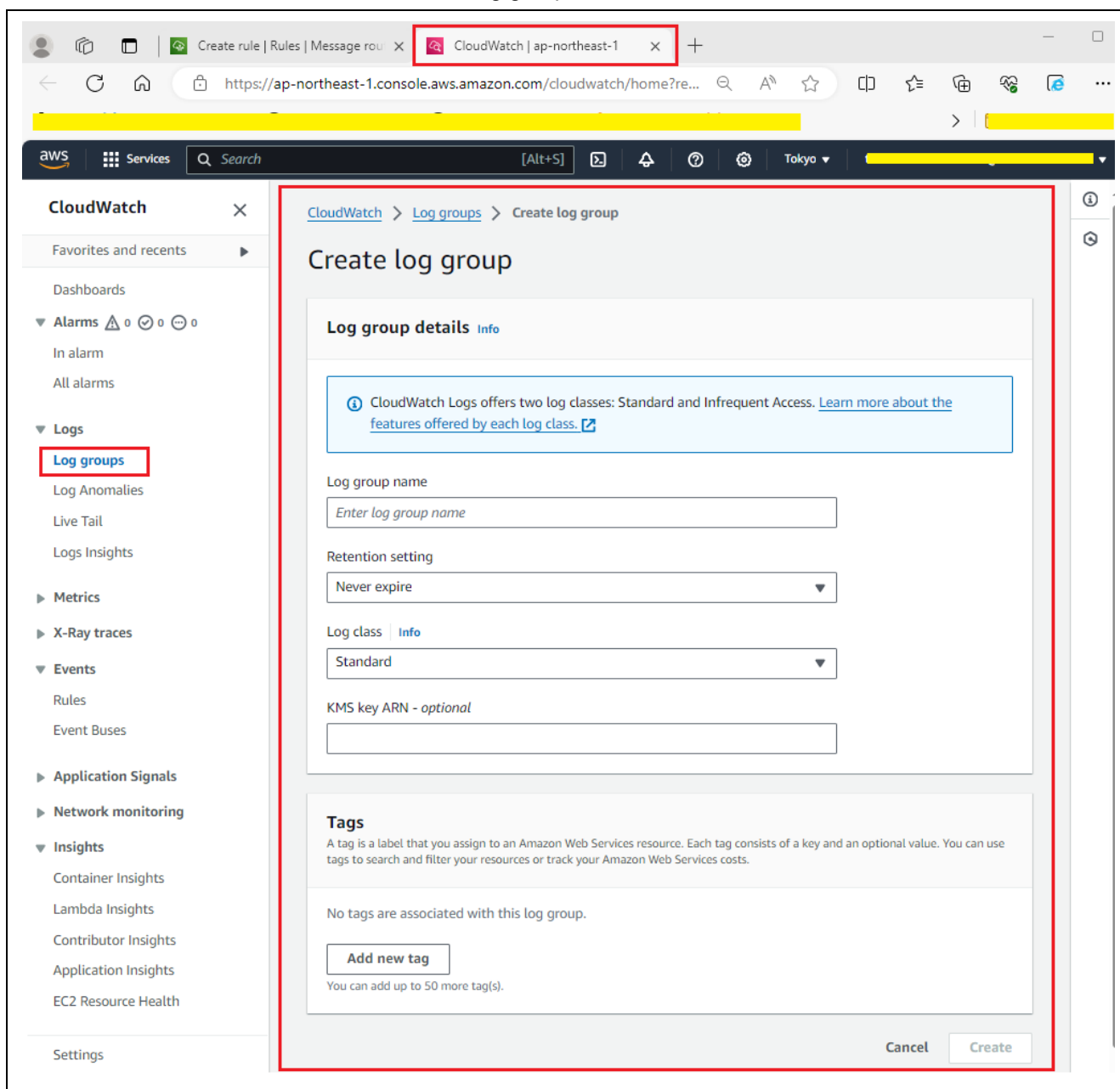


図 5-15 新規の CloudWatch タブが起動し Create log group の画面であることを確認

(5) ロググループを作成

- Log group name を入力し“Create”をクリックし、ロググループが作成されていることを確認

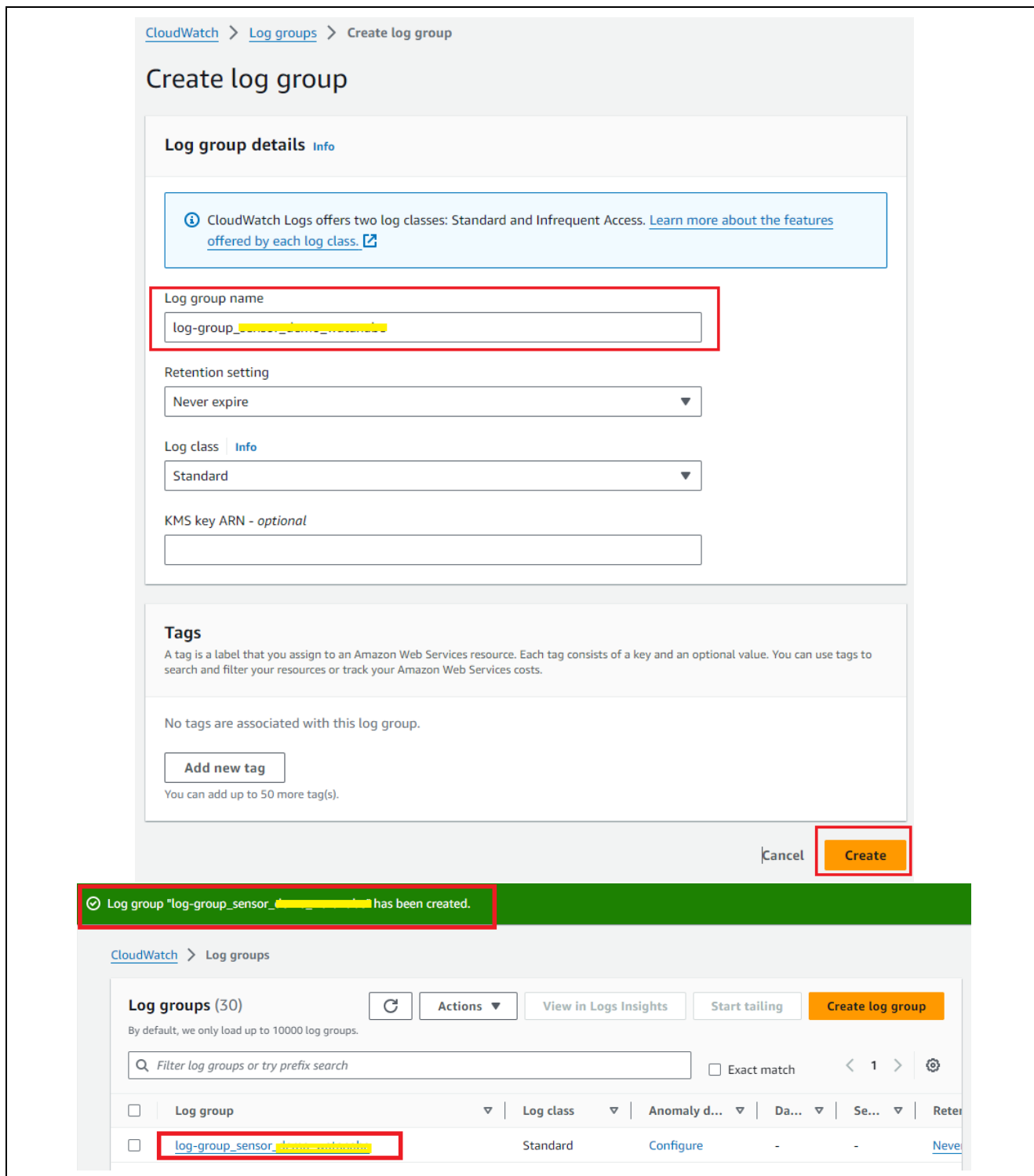


図 5-16 ロググループが正しく作成されることを確認

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

### (6) 作成したロググループを選択

- Rules のタブをクリックしてアクティブ化

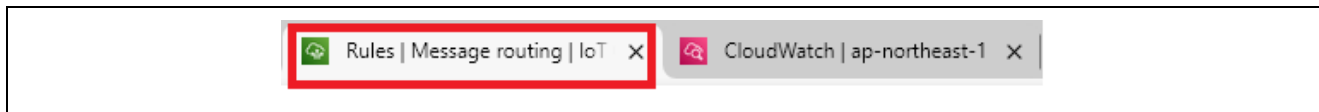


図 5-17 Rules のタブをクリックしてアクティブ化

- Log group name に(5)で作成したロググループを選択し”Create new role”をクリック

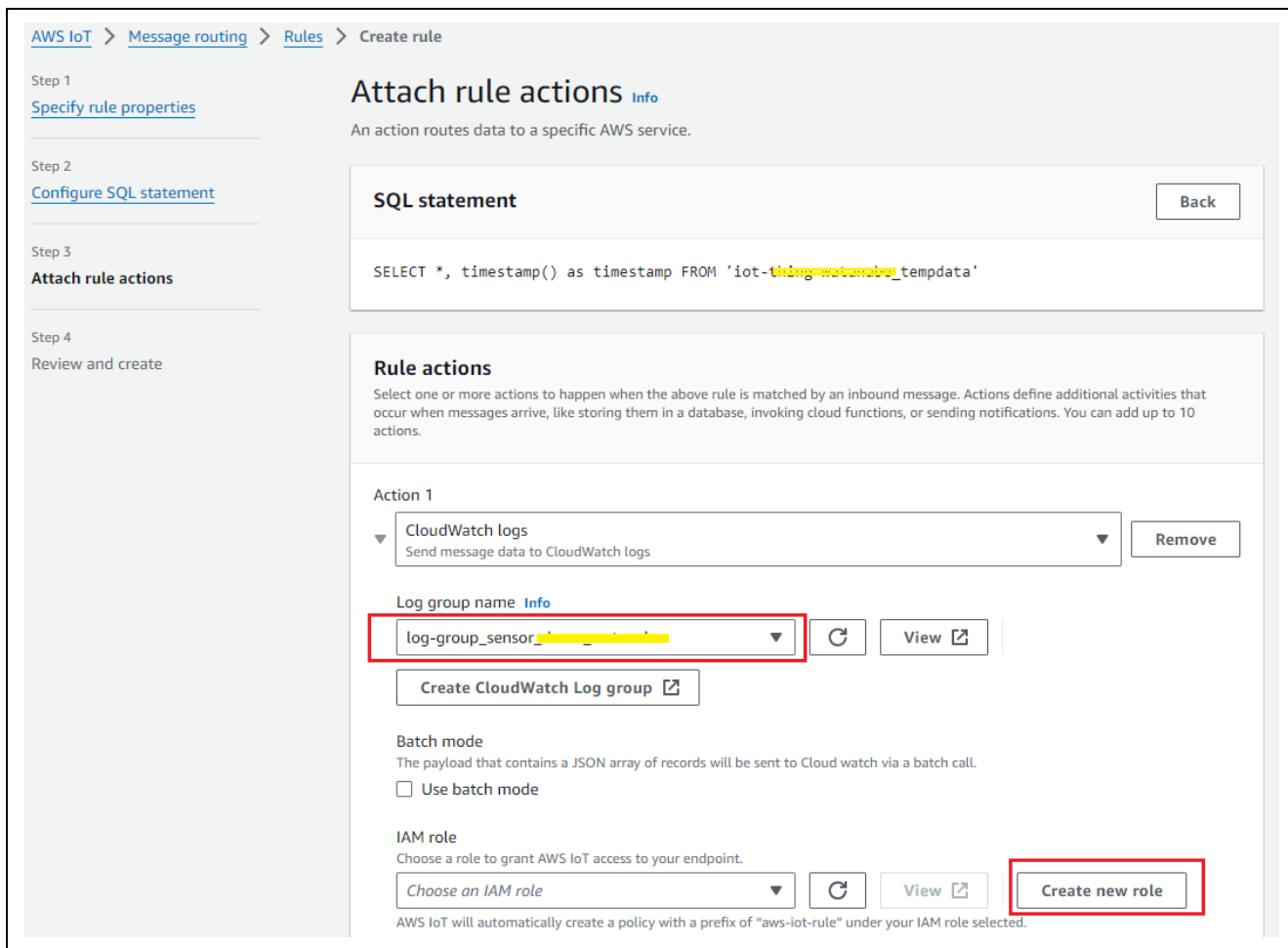


図 5-18 Log group name に作成したロググループを選択

(7) IAM ロールを作成

- Role name を入力し“Create”をクリック

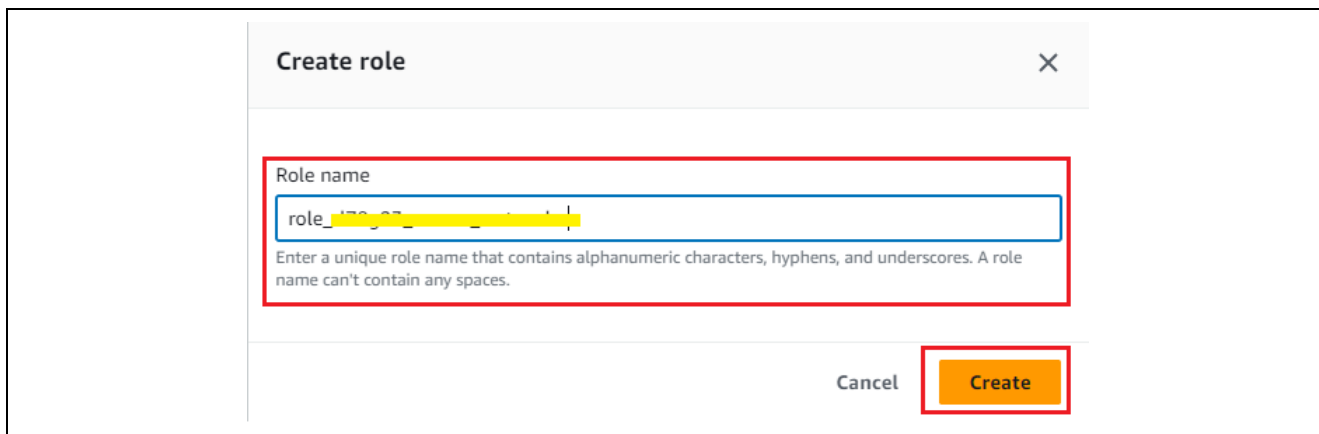
The image shows a screenshot of the AWS IAM console's 'Create role' dialog box. The dialog box has a title bar 'Create role' with a close button (X) in the top right. Below the title bar is a 'Role name' label and a text input field. The input field contains the text 'role\_1234567890123456789012345678901234567890' and is highlighted with a red rectangular border. Below the input field, there is a small instructional text: 'Enter a unique role name that contains alphanumeric characters, hyphens, and underscores. A role name can't contain any spaces.' At the bottom of the dialog box, there are two buttons: 'Cancel' and 'Create'. The 'Create' button is highlighted with a red rectangular border.

図 5-19 Role name を入力し“Create”をクリック



## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

### (8) 作成した IAM ロールを選択

- IAM role に(7)で作成した IAM ロールを選択し“Next”をクリック

Successfully created role role\_rl78g23\_...

AWS IoT > Message routing > Rules > Create rule

Step 1  
Specify rule properties

Step 2  
Configure SQL statement

Step 3  
**Attach rule actions**

Step 4  
Review and create

### Attach rule actions Info

An action routes data to a specific AWS service.

**SQL statement** Back

```
SELECT *, timestamp() as timestamp FROM 'iot-thing-attributes-tempdata'
```

**Rule actions**

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  
log-group\_sensor\_... Refresh View

[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.  
role\_rl78g23\_... Refresh 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**

図 5-20 作成した IAM ロールを選択し“Next”をクリック

(9) ルールの内容を確認しルールを作成

- ルールの内容を確認し“Create”をクリック

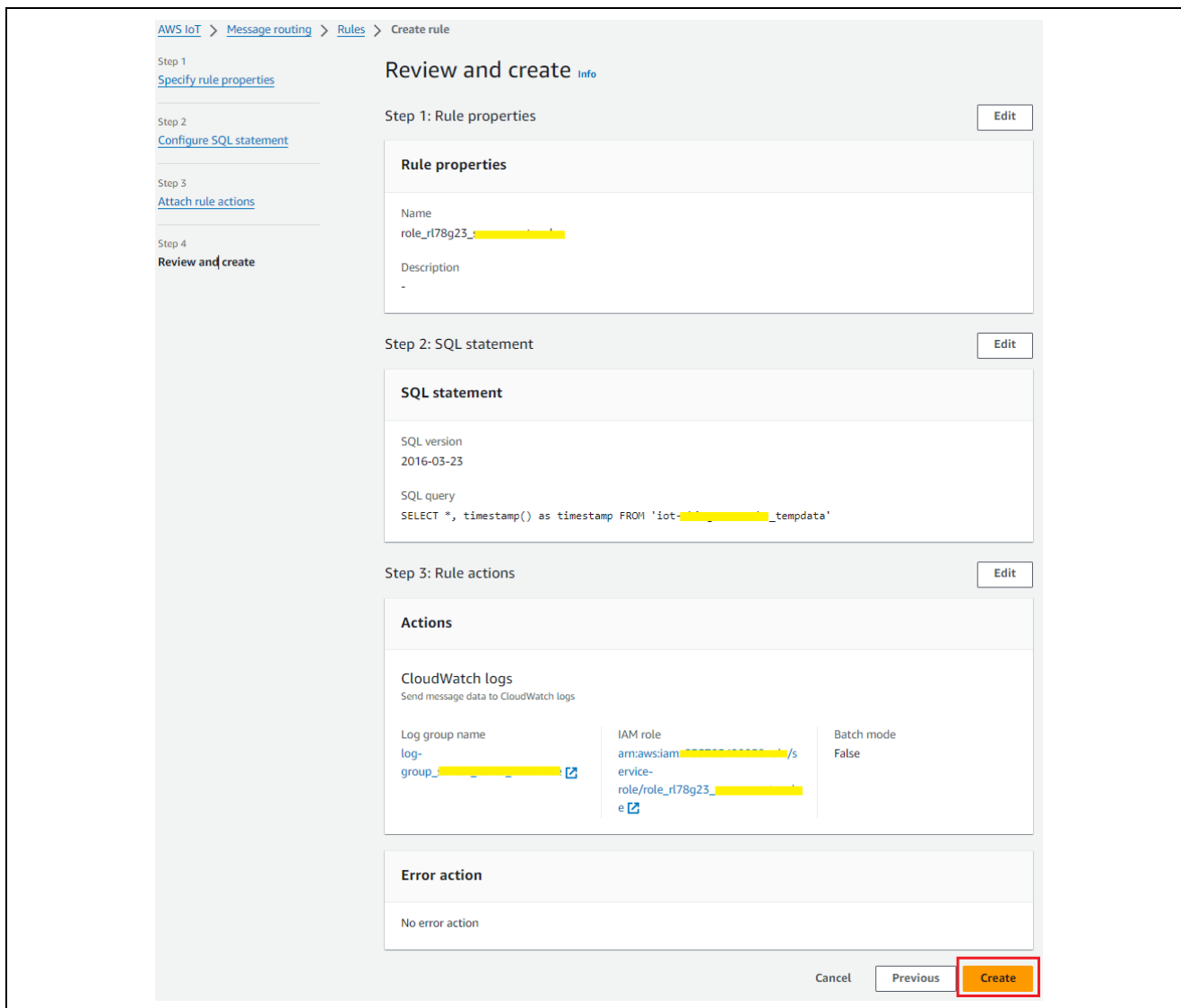


図 5-21 ルールの内容を確認し“Create”をクリック

- ルールが作成されたことを確認

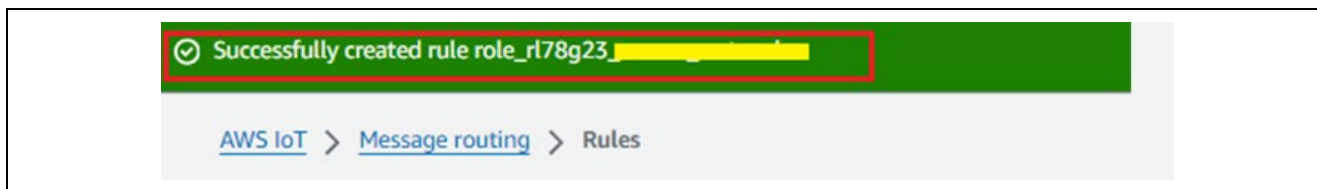


図 5-22 ルールが作成されたことを確認

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

### (10) CloudWatch でグラフを表示

- CloudWatch > Logs Insights
- Log group: 0 で作成したロググループを選択
- Query: 以下のクエリを入力

```
stats avg(humidity), avg(temperature), max(warning) by bin(10s)
```

- Visualization: “Visualization”を選択

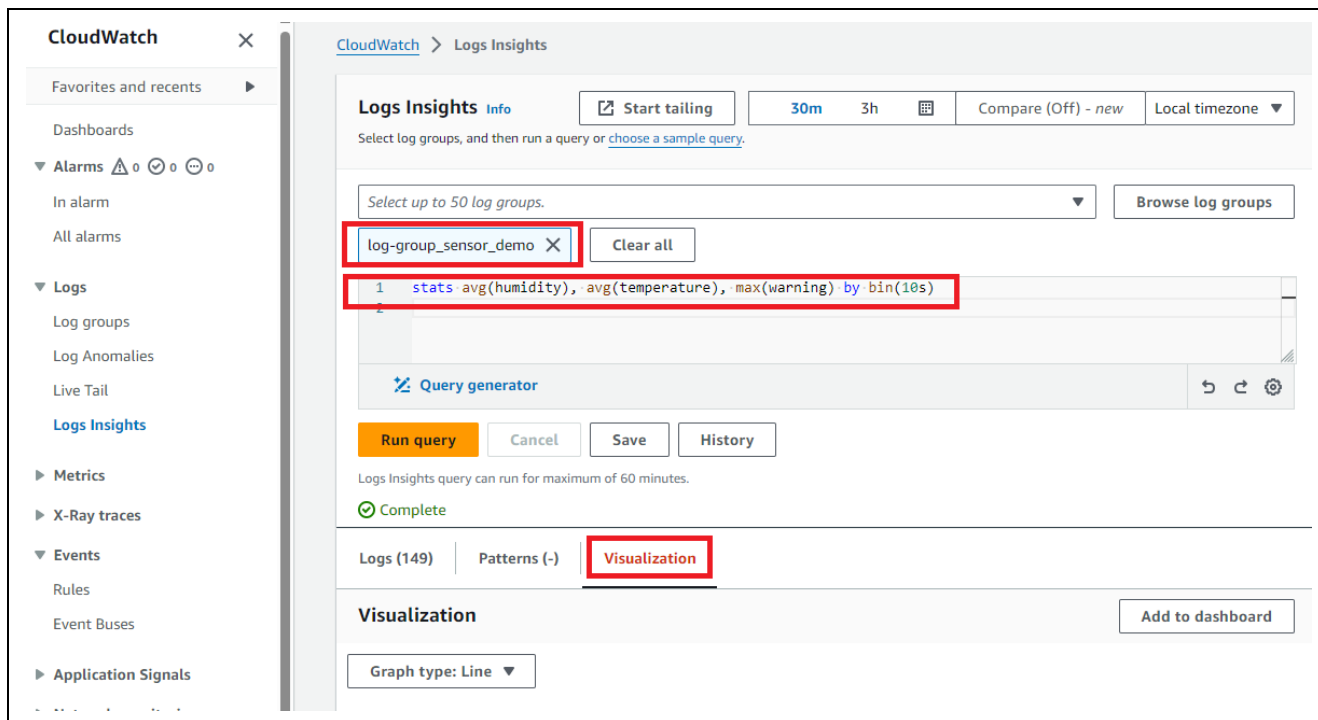


図 5-23 CloudWatch > Logs Insights

- デモプロジェクト実行後に[Run query]を押してグラフを表示

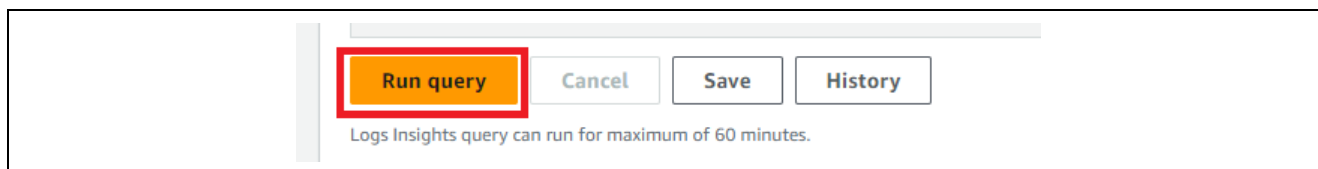


図 5-24 [Run query]を押してグラフを表示

#### 5.4.2 OTA アップデートの設定

##### 5.4.2.1 Amazon S3 バケットの作成

[Getting Started Guide](#) の 6.1.3.1 Creating Amazon S3 Buckets を参照してください。

##### 5.4.2.2 OTA 更新用サービスロールの作成

[Getting Started Guide](#) の 6.1.3.2 Creating an OTA Update Service Role を参照してください。

##### 5.4.2.3 OTA 更新用ユーザポリシーの作成

[Getting Started Guide](#) の 6.1.3.3 Creating an OTA Update User Policy を参照してください。

##### 5.4.2.4 OTA 更新用ポリシーを IAM ユーザに割り当て

[Getting Started Guide](#) の 6.1.3.4 Allocating an OTA Update Policy to IAM User を参照してください。

##### 5.4.2.5 AWS IoT のコード署名へのアクセス権付与

[Getting Started Guide](#) の 6.1.3.5 Granting Access Permissions to AWS IoT Code Signature を参照してください。

## 5.5 初期イメージの作成

初期イメージとは、ブートローダの MOT ファイルと初期アプリケーションの MOT ファイルを「Renesas Image Generator」で結合した MOT ファイルのことです。

Renesas Image Generator は「[RL78/G22,RL78/G23,RL78/G24 ファームウェア アップデート モジュール](#)」に付属するツールです。詳細は上記リンク先のアプリケーションノートを参照してください。

この文書では初期イメージに関連するファイル名を以下とします。

- ブートローダ : boot\_loader.mot
- 初期アプリケーション : aws\_ryz024a\_sonsor\_rl78g23-fpb\_ota.mot
- 初期イメージ : initial\_image.mot

### 5.5.1 ブートローダを作成

本章の手順は [Getting Started Guide](#) と同じです。

#### 5.5.1.1 ブートローダプロジェクトのインポート

[Getting Started Guide](#) の 6.2.1.1 Importing the Bootloader Project を参照してください。

#### 5.5.1.2 ファームウェア検証用の鍵をブートローダプロジェクトに追加

[Getting Started Guide](#) の 6.2.1.2 Adding the Firmware Verification Key to the Bootloader Project を参照してください。

#### 5.5.1.3 ブートローダプロジェクトのビルド

[Getting Started Guide](#) の 6.2.1.3 Building the Bootloader Project を参照してください。

## 5.5.2 初期アプリケーションを作成

本章の手順は [Getting Started Guide](#) と同じです。プロジェクト名やコマンドパラメータ等を本アプリケーションノートのものに読み替えてください。

### 5.5.2.1 初期アプリケーションのインポート

プロジェクト「aws\_ryz024a\_sensor\_rl78g23-fpb」を e2 studio にインポートします。次の手順でインポートウィザードを開いてください。

File > Import... > Existing Projects into Workspace > Next

次に、プロジェクト「aws\_ryz024a\_sensor\_rl78g23-fpb」を選択してください。**この時、ワークスペースにプロジェクトをコピーしないオプション設定にしてください。**その後、Finish ボタンをクリックします。

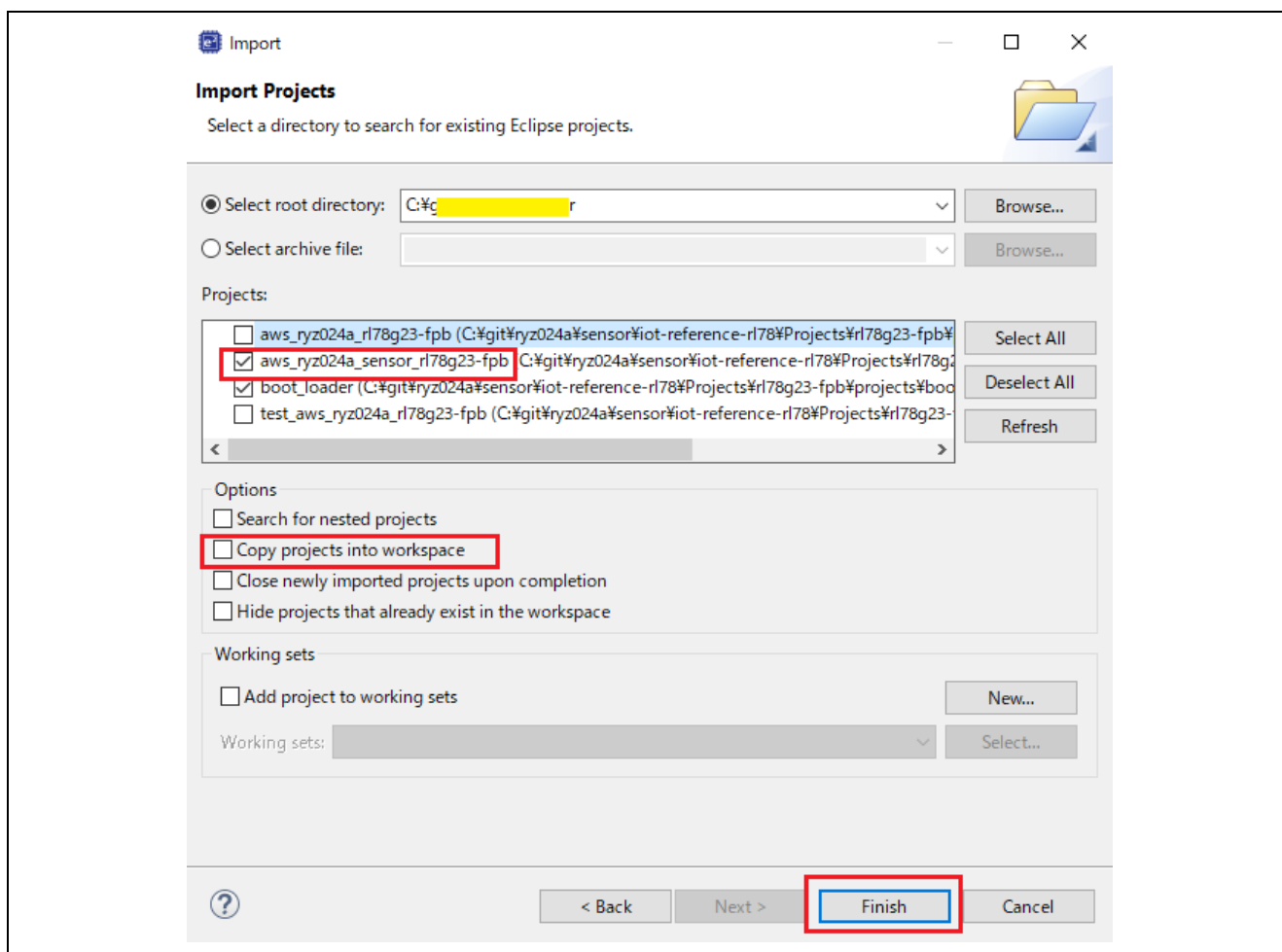


図 5-25 プロジェクト「aws\_ryz024a\_sensor\_rl78g23-fpb」を選択

プロジェクト・エクスプローラー画面にインポートされたプロジェクトが表示されます。

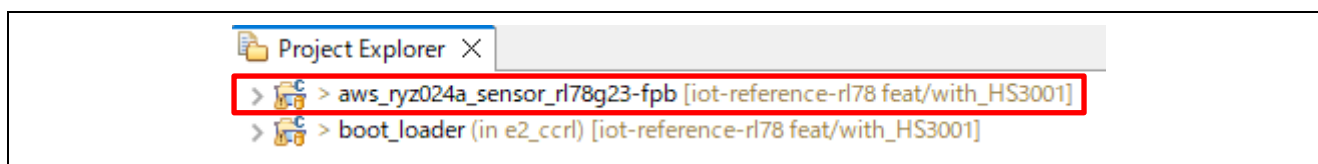


図 5-26 プロジェクト「aws\_ryz024a\_sensor\_rl78g23-fpb」のインポート完了

### 5.5.2.2 ファームウェア検証用の鍵を初期アプリケーションに追加

プロジェクト「aws\_ryz024a\_sensor\_rl78g23-fpb」の code\_signer\_public\_key.h にファームウェア検証用の鍵(secp256r1.publickey)を追加します。

**注意事項：**各行末に \ を付与してください。

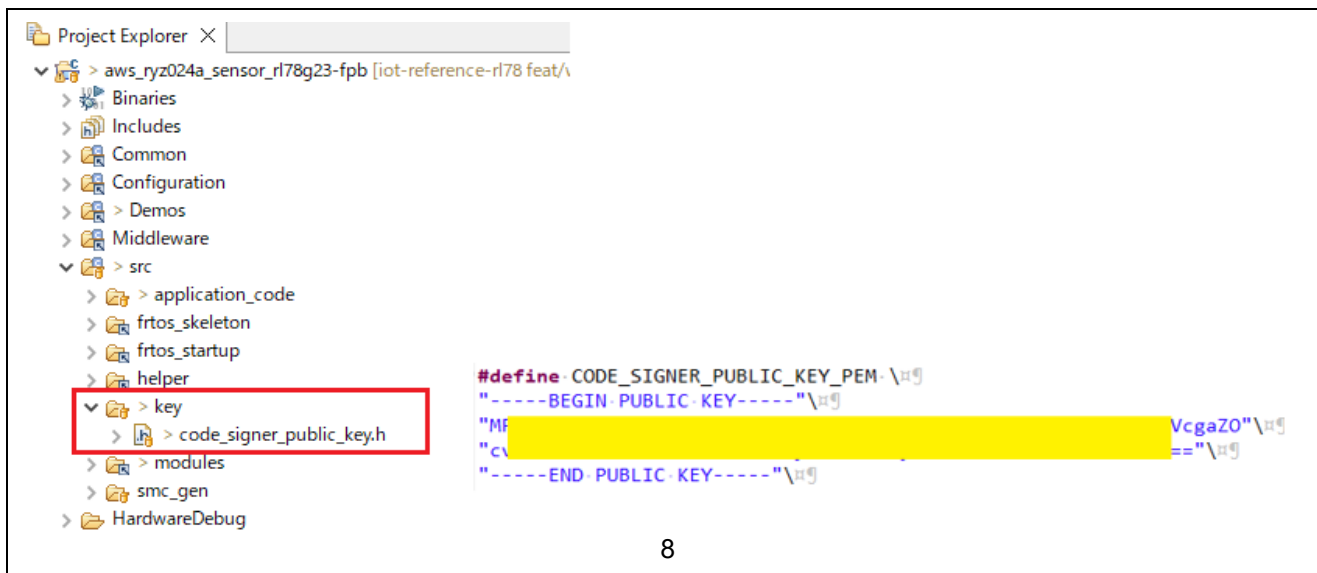


図 5-27 初期アプリケーションにファームウェア検証用の鍵を追加

### 5.5.2.3 初期アプリケーションのビルド

プロジェクト「aws\_ryz024a\_sensor\_rl78g23-fpb」をビルドし、MOT ファイルを作成します。その後、プロジェクトフォルダ直下の HardwareDebug フォルダに aws\_ryz024a\_sensor\_rl78g23-fpb.mot が生成されていることを確認します。

### 5.5.3 Renesas Image Generator で初期イメージを生成

ブートローダと初期アプリケーションを Renesas Image Generator で結合して初期イメージを生成します。

① Renesas Image Generator と同じフォルダに以下のファイルを格納します。

- ブートローダ : boot\_loader.mot
- 初期アプリケーション : aws\_ryz024a\_sensor\_rl78g23-fpb.mot
- 初期アプリケーション検証用の秘密鍵 : secp256r1.privatekey

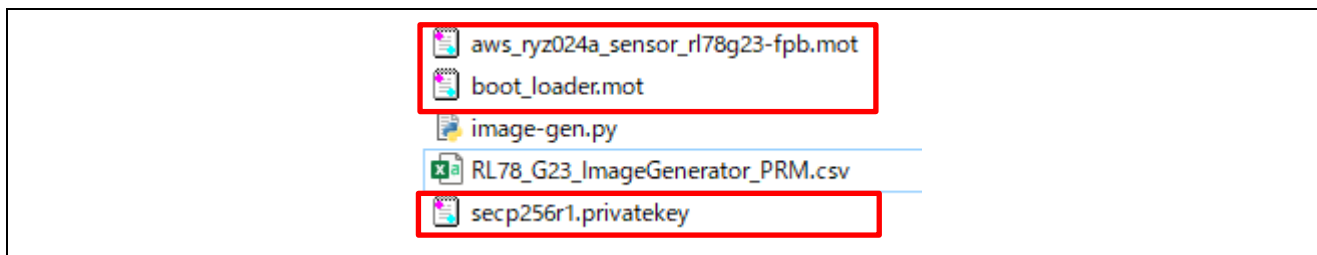


図 5-28 Renesas Image Generator と同じフォルダに必要なファイルを格納

② 以下のコマンドを実行し初期イメージを生成します。

```
python image-gen.py -iup .\aws_ryz024a_sensor_rl78g23-fpb.mot -ibp  
boot_loader.mot -o initial_image -ip .\RL78_G23_ImageGenerator_PRM.csv
```

③ 初期イメージ (initial\_image.mot)が生成されていることを確認してください。

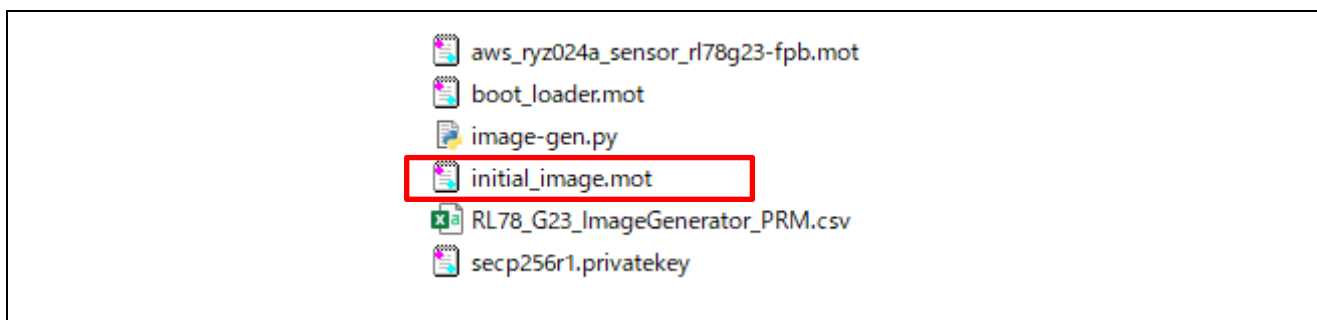


図 5-29 Renesas Image Generator と同じフォルダに初期イメージが生成される



## 5.6 更新イメージの作成

更新イメージとは、更新アプリケーションの MOT ファイルを「Renesas Image Generator」で変換したバイナリ形式 (拡張子 rsu) の更新用ファームウェアのことです。更新イメージのフォーマットについては「[RL78/G22,RL78/G23,RL78/G24 ファームウェア アップデート モジュール](#)」を参照してください。

この文書では更新イメージに関連するファイル名を以下とします。

- 更新アプリケーション : aws\_ryz024a\_sensor\_rl78g23-fpb\_093.mot
- 更新イメージ : aws\_ryz024a\_sensor\_rl78g23-fpb\_093.rsu

### 5.6.1 更新アプリケーションを作成

本章の手順は [Getting Started Guide](#) と同じです。プロジェクト名やコマンドパラメータ等を本アプリケーションノートのものに読み替えてください。

#### 5.6.1.1 アプリケーションのソースコードを変更

更新アプリケーションを作成するために以下を変更します。

iot-reference-rl78\Configuration\rl78g23-fpb\sensor\cellular\ertos\_config\demo\_config.h の「APP\_VERSION\_BUILD」マクロの定義を 2 から 3 に変更します。

```
iot-reference-rl78\Configuration\rl78g23-  
fpb\sensor\cellular\ertos_config\demo_config.h  
  
/**  
 * @brief Build version of the firmware.  
 *  
 * This is used in the OTA demo to set the appFirmwareVersion variable that  
 is  
 * declared in the ota_appversion32.h file in the OTA library.  
 */  
#ifndef APP_VERSION_BUILD  
    #define APP_VERSION_BUILD    3  
#endif
```

iot-reference-rl78\Demos\Sensor\SensorDemo.c の「[SENSOR\\_ENABLE\\_HUMIDITY](#)」マクロの定義を 0 から 1 に変更します。

```
iot-reference-rl78\Demos\Sensor\SensorDemo.c  
  
/* Sensor for HSxxxx series */  
#include "sensor.h"  
#define SENSOR_ENABLE_HUMIDITY    ( 1 )
```

#### 5.6.1.2 更新アプリケーションのビルド

aws\_ryz024a\_sensor\_rl78g23-fpb プロジェクトをビルドし、MOT ファイルを作成します。その後、プロジェクトフォルダ直下の HardwareDebug フォルダに aws\_ryz024a\_sensor\_rl78g23-fpb.mot が上書き生成されていることを確認します。

### 5.6.1.3 更新アプリケーションの MOT ファイルをリネーム

aws\_ryz024a\_sensor\_rl78g23-fpb.mot を aws\_ryz024a\_sensor\_rl78g23-fpb\_093.mot にリネームします。

### 5.6.2 Renesas Image Generator で更新イメージを生成

更新アプリケーションを Renesas Image Generator で更新イメージに変換します。

- ① Renesas Image Generator と同じフォルダに以下のファイルを格納します。
  - 更新アプリケーションの MOT ファイル : aws\_ryz024a\_sensor\_rl78g23-fpb\_093.mot
  - 更新アプリケーション検証用の秘密鍵 : secp256r1.privatekey



図 5-30 Renesas Image Generator と同じフォルダに必要なファイルを格納

- ② 以下のコマンドを実行し RSU 形式の更新イメージ aws\_ryz024a\_sensor\_rl78g23-fpb\_093.rsu を生成します。

```
python image-gen.py -iup .\aws_ryz024a_sensor_rl78g23-fpb_093.mot -o  
aws_ryz024a_sensor_rl78g23-fpb_093 -ip .\RL78_G23_ImageGenerator_PRM.csv -vt  
ecdsa -ff RTOS
```

- ③ aws\_ryz024a\_sensor\_rl78g23-fpb\_093.rsu が生成されていることを確認します。

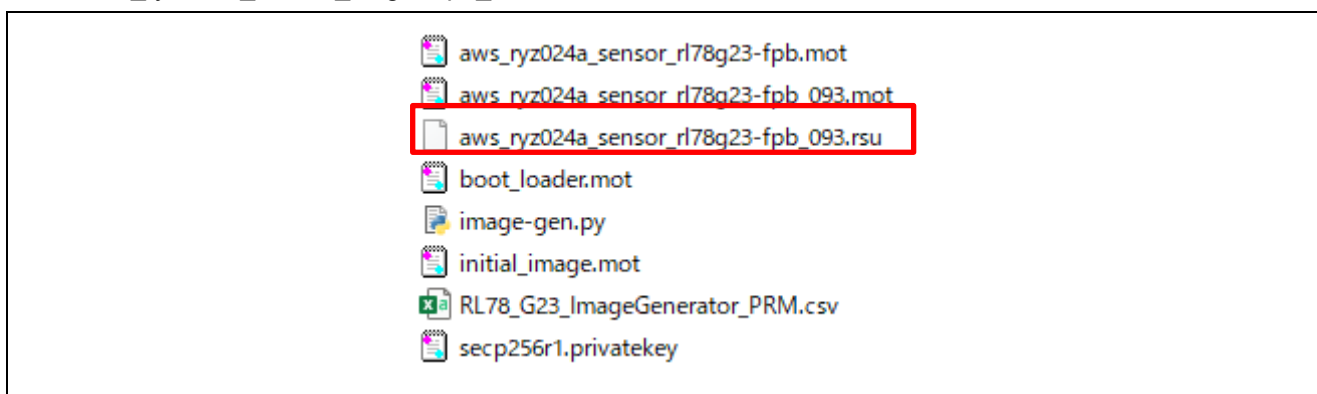


図 5-31 Renesas Image Generator と同じフォルダに更新イメージが生成される

## 5.7 デモプロジェクトの実行

デモプロジェクトの実行手順を説明します。

### 5.7.1 初期イメージ (initial\_image.mot)をボードに書き込む

- ① 初期イメージ (initial\_image.mot)を書き込みます。  
書き込み方法は「6章 Renesas Flash Programmer の使用方法」を参照してください。
- ② USB ケーブルを抜いてマイコンボードへの電源供給を一旦停止します。
- ③ USB ケーブルを再接続し電源供給を再開しデモプロジェクトを起動します。
- ④ ターミナルで初期アプリケーション (バージョン 0.9.2)が起動しセンサボードから取得した温度データを送信していることを確認します。

```
==== RL78G23 : BootLoader [with buffer] ====
verify install area main [sig-sha256-ecdsa]...OK
execute new image ...Hello World.
0 3927 [MAIN_TASK] [INFO] >>> Cellular SIM okay <<<

1 4228 [MAIN_TASK] [INFO] >>> Cellular_GetServiceStatus 0, PS registration status 0 <<<
2 5242 [MAIN_TASK] [INFO] >>> Cellular_GetServiceStatus 0, PS registration status 2 <<<
3 6257 [MAIN_TASK] [INFO] >>> Cellular_GetServiceStatus 0, PS registration status 2 <<<
4 7276 [MAIN_TASK] [INFO] >>> Cellular module registered <<<

-----STARTING DEMO-----
5 8423 [MQTT] [INFO] -----Start MQTT Agent Task-----

6 8424 [MQTT] [INFO] Creating a TLS connection to [REDACTED].ap-northeast-1.amazonaws.com:8883.
7 10893 [MQTT] [INFO] Creating an MQTT connection to the broker.
8 11987 [MQTT] [INFO] MQTT connection established with the broker.
9 11988 [MQTT] [INFO] Successfully connected to MQTT broker.
11 11995 [OTA Demo Ta] [INFO] -----Start OTA Task-----
12 12001 [OTA Demo Ta] [INFO] OTA over MQTT demo, Application version 0.9.2
9 11988 [MQTT] [INFO] Successfully connected to MQTT broker.
13 12015 [PUBSUB] [INFO] Sending subscribe request to agent for topic filter: iot-[REDACTED]-tempdata
14 12022 [OTA Demo Ta] [INFO] Received: 0 Queued: 0 Processed: 0 Dropped: 0
15 12030 [OTA Agent T] [INFO] Current State=[RequestingJob], Event=[Start], New state=[RequestingJob]
16 12870 [PUBSUB] [INFO] Successfully subscribed to topic: iot-[REDACTED]-tempdata
17 12974 [PUBSUB] [INFO] Sending publish request on topic "iot-[REDACTED]-tempdata"
18 13819 [OTA Agent T] [INFO] Subscribed to topic $aws/things/[REDACTED]/jobs/notify-next.

19 13820 [OTA Agent T] [INFO] Subscribed to MQTT topic: $aws/things/[REDACTED]/jobs/notify-next
20 13919 [MQTT] [INFO] Publishing message to iot-[REDACTED]-tempdata.

21 14030 [OTA Demo Ta] [INFO] Received: 0 Queued: 0 Processed: 0 Dropped: 0
22 14547 [PUBSUB] [INFO] Successfully sent QoS 0 publish to topic: iot-[REDACTED]-tempdata (PassCount:1, FailCount:0).
23 14573 [MQTT] [INFO] De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
24 14574 [MQTT] [INFO] State record updated. New state=MQTTPublishDone.
25 14583 [MQTT] [INFO] Received incoming publish message {"temperature": 22.09, "warnings": 0, "things": "[REDACTED]"}
26 14690 [MQTT] [INFO] Publishing message to $aws/things/[REDACTED]/jobs/$next/get.
```

図 5-32 初期アプリケーション (バージョン 0.9.2)起動

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

⑤ CloudWatch でグラフを表示します。

● CloudWatch > Logs Insights > “Run query”をクリック

— temperature: 温度のグラフです。

— warning: 温度のワーニングです。25 度未満の場合は False(値は 0)、25 度以上は True(値は 50)です。

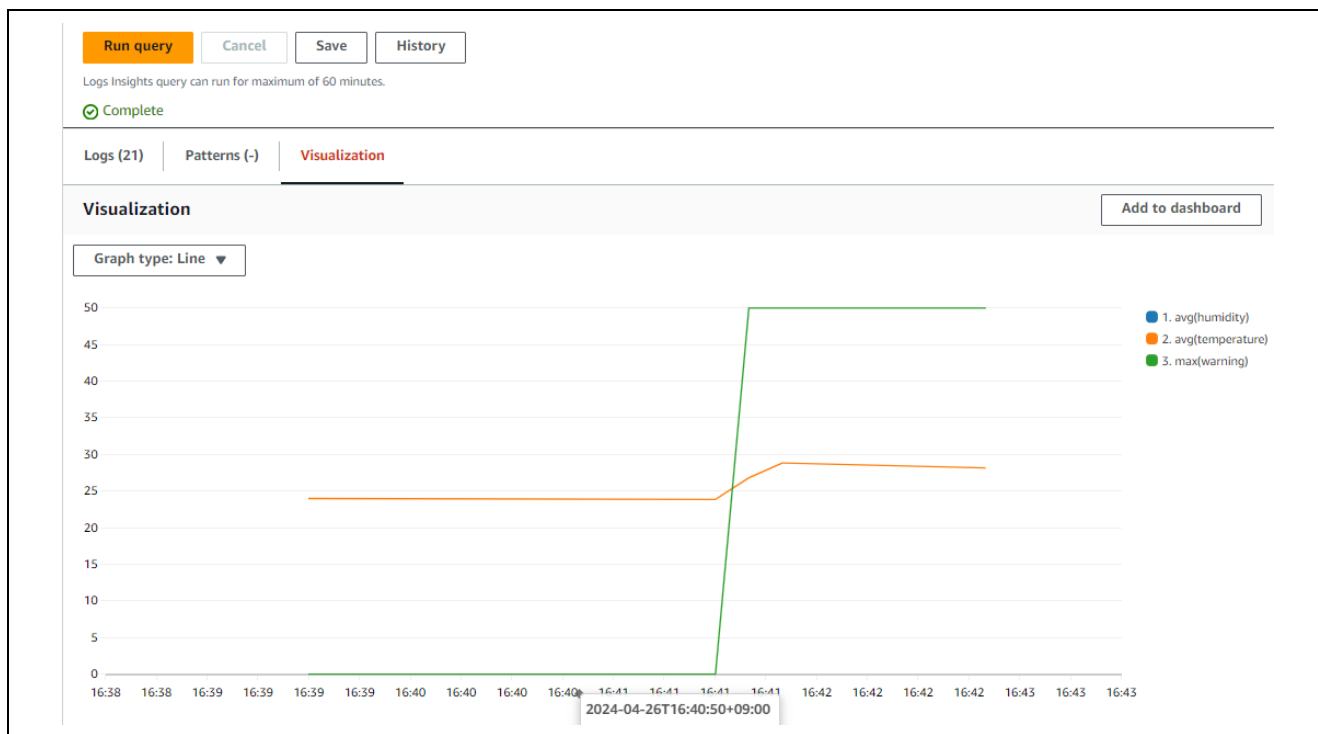


図 5-33 初期アプリケーション(バージョン 0.9.2)起動後に CloudWatch でグラフを表示

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

### 5.7.2 更新イメージ (aws\_ryz024a\_sensor\_rl78g23-fpb\_093.rsu)を OTA ジョブに登録

- ① AWS IoT > Manage > Remote actions > Jobs > “Create job” をクリック

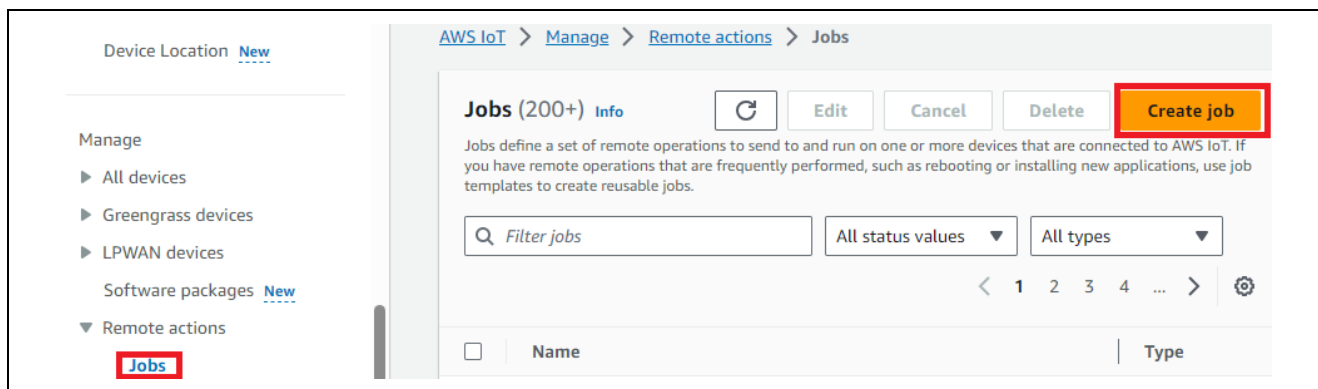


図 5-34 Jobs

- ② Check “Create FreeRTOS OTA update job” > “Next” をクリック

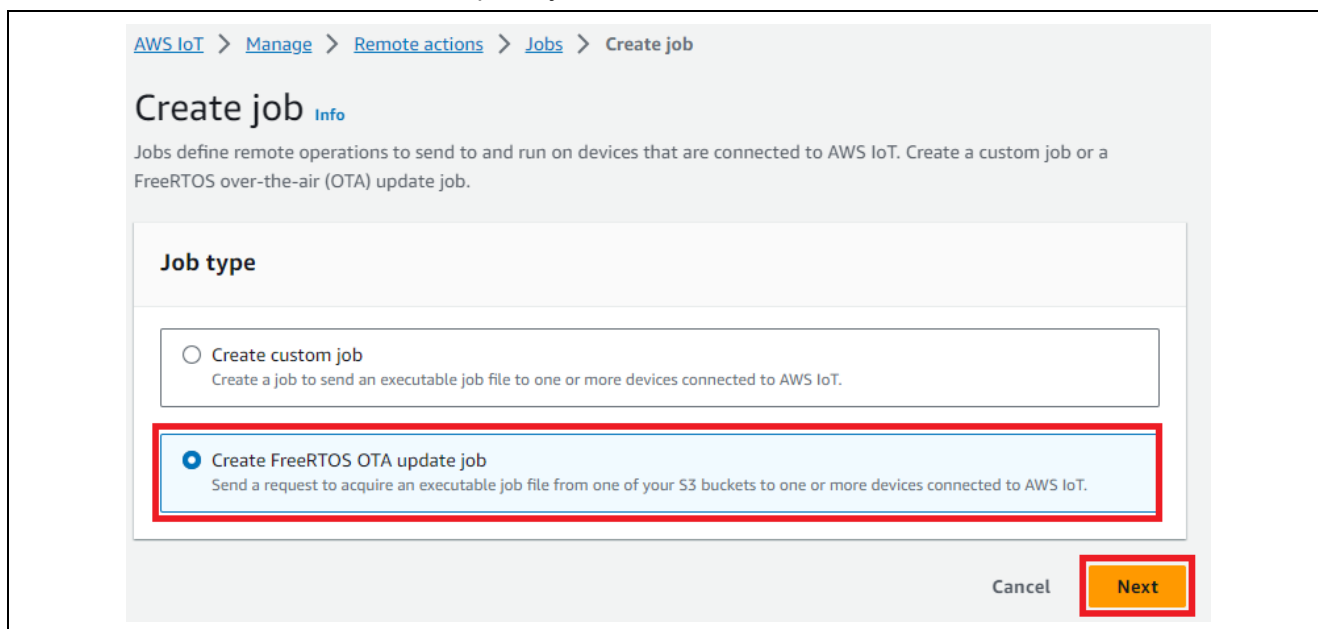
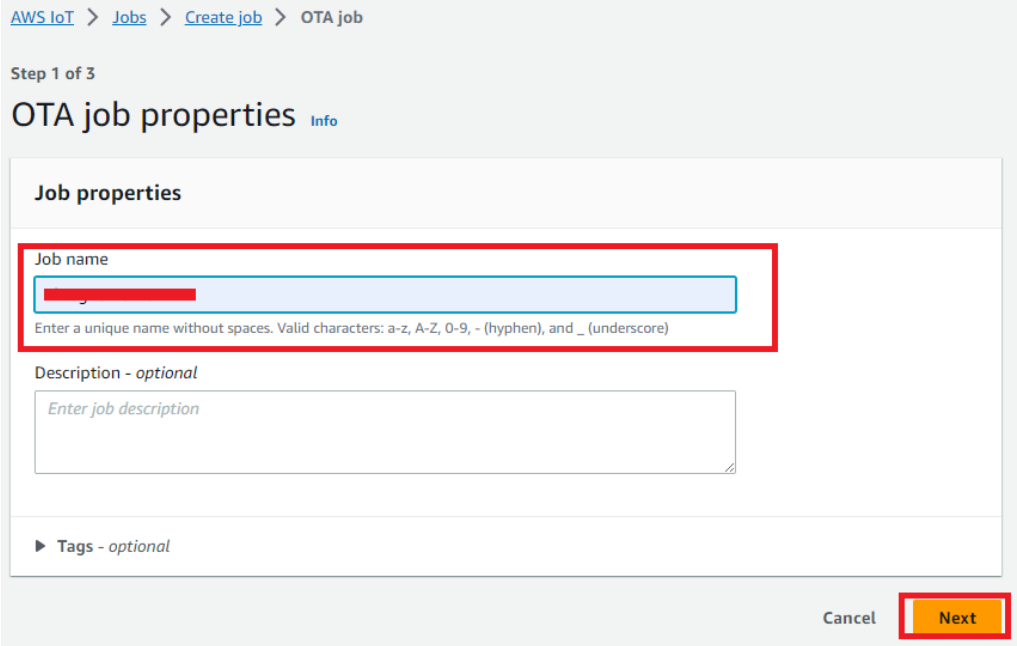


図 5-35 Create Job

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

### ③ Step1: OTA job properties

- Job name: 任意



AWS IoT > Jobs > Create job > OTA job

Step 1 of 3

### OTA job properties [Info](#)

**Job properties**

**Job name**

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

**Description - optional**

Enter job description

**Tags - optional**

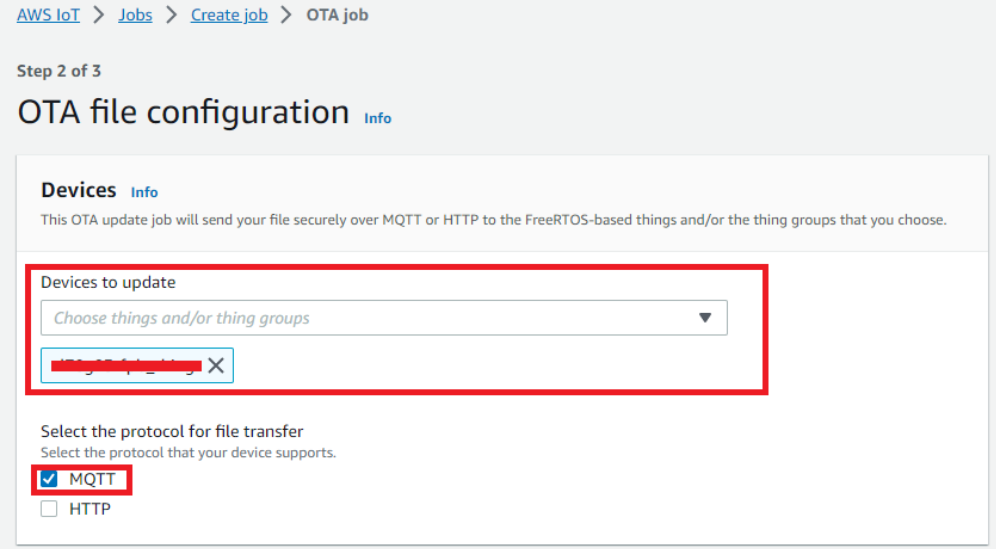
図 5-36 Step1: OTA job properties

### ④ Step2: OTA file configuration > Devices

- Devices to update: aws\_clientcredential.h の”モノの名前”

```
#define clientcredentialIOT_THING_NAME "YOUR THING NAME"
```

- Select the protocol for file transfer: MQTT



AWS IoT > Jobs > Create job > OTA job

Step 2 of 3

### 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**

Select the protocol for file transfer

Select the protocol that your device supports.

MQTT

HTTP

図 5-37 Step2: OTA file configuration > Devices

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

- ⑤ Step2: OTA file configurations > File
  - Sign and choose your file: Sign a new file for me.

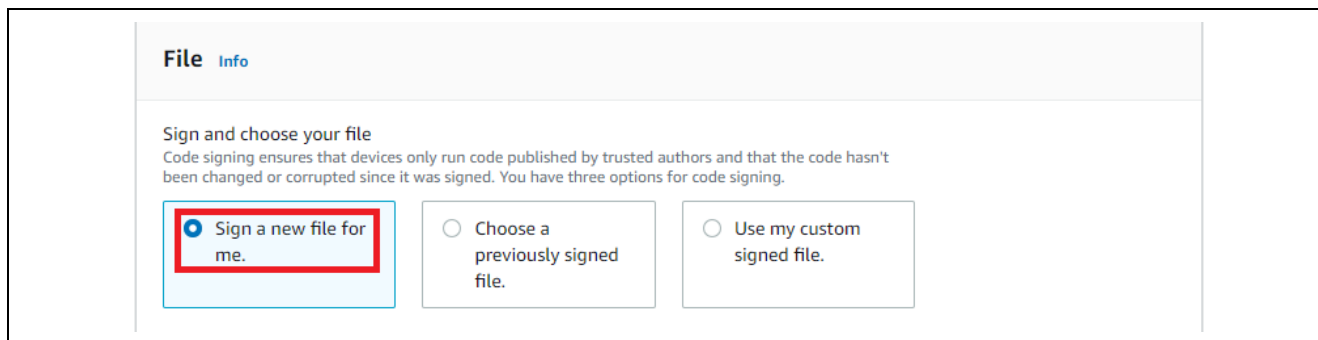


図 5-38 Step2: OTA file configurations > File (1)

- Code signing profile: “Create new profile” をクリック

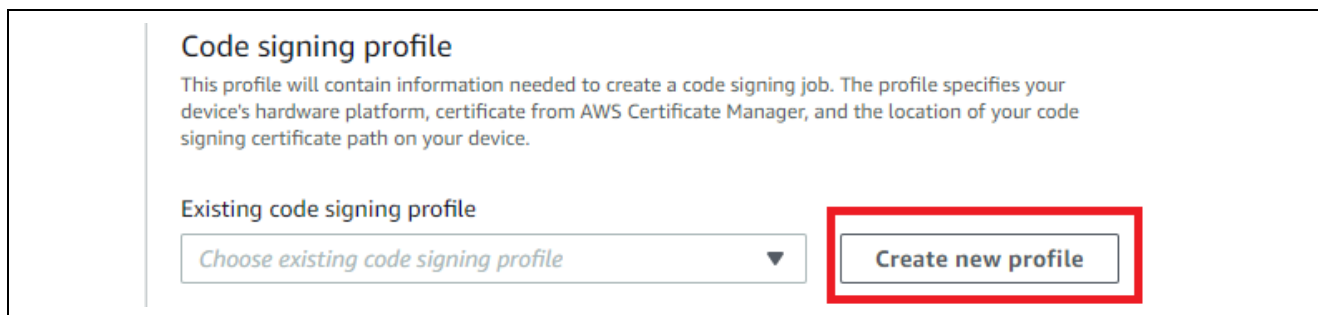


図 5-39 Step2: OTA file configurations > File (2)

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

- Create a code signing profile.
  - Profile name: 任意 (例 : rl78g23\_fpb\_ota\_cert)
  - Device hardware platform: Windows Simulator
  - Code signing certificate: “Import new code signing certificate”
  - Certificate body: secp256r1.crt
  - Certificate private key: secp256r1.privatekey
  - Certificate chain - optional: ca.crt
  - Path name of code signing certificate on device: 任意

**Create a code signing profile** [X]

Profile name  
rl78g23\_fpb\_ota\_cert  
Enter a unique name without spaces. Valid characters: a-z, A-Z, 0-9, and \_ (underscore)

Device hardware platform  
Windows Simulator

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 906 bytes Uploaded
Certificate private key	secp256r1.privatekey 232 bytes Uploaded
Certificate chain - optional	ca.crt 1030 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.  
/certificates/rl78g23\_fpb\_ota\_cert

Cancel **Create**

図 5-40 Create a code signing profile



## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

- File > “Upload a new file.” > “Choose file” > aws\_ryz024a\_sensor\_rl78g23-fpb\_093.rsu

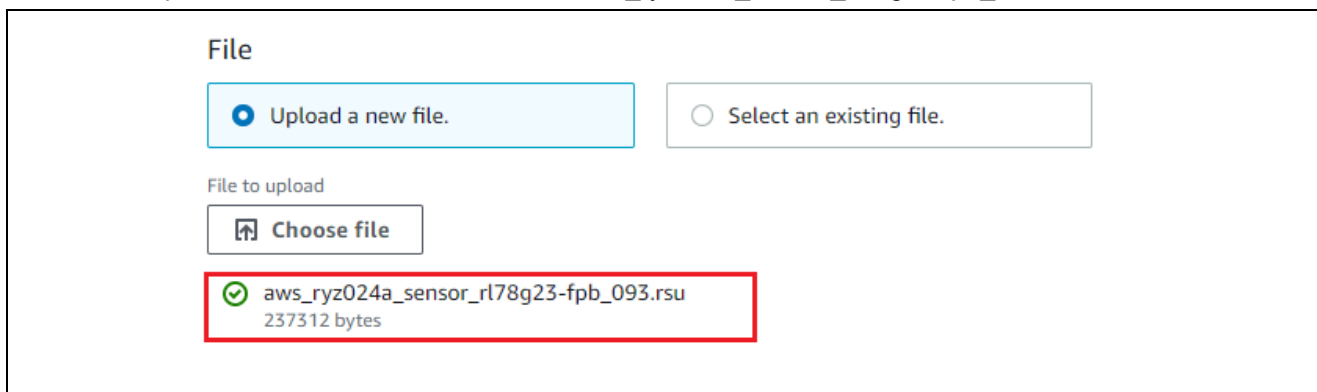


図 5-41 Upload a new file > aws\_ryz024a\_sensor\_rl78g23-fpb\_093.rsu

- File upload location in S3: 作成済のバケットを指定 (「5.4.2.1 章 Amazon S3 バケットの作成」で指定した Bucket name)
- Path name of file on device: 任意

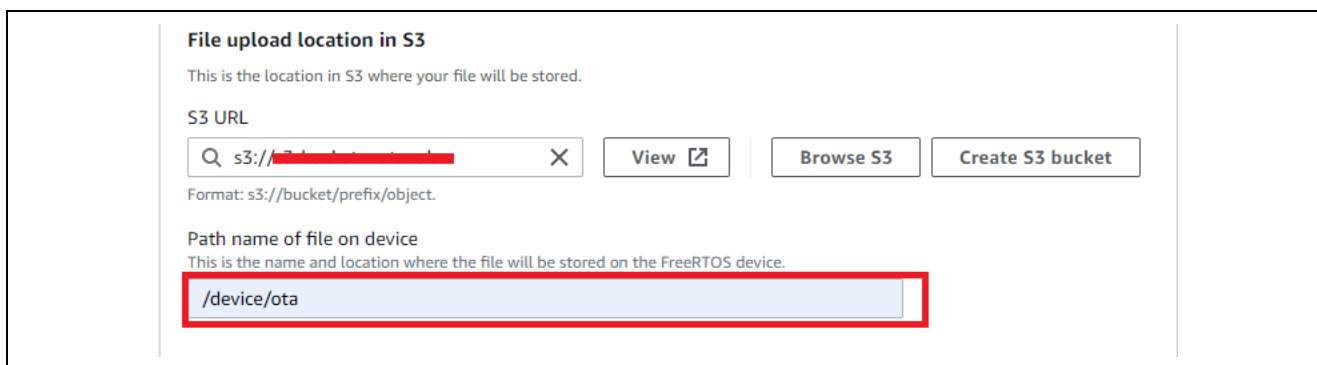


図 5-42 File upload location in S3

### ⑥ Step2: OTA file configurations > IAM role

- Role: 作成済のロールを指定 (「5.4.2.2 章 OTA 更新用サービスロールの作成」で指定した Role name)

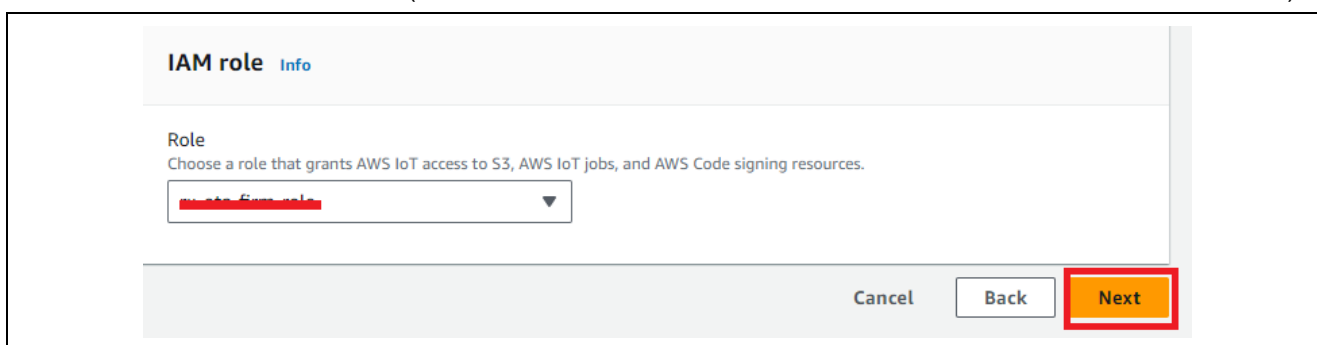


図 5-43 Step2: OTA file configurations > IAM role

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

### ⑦ Step3: OTA job configuration

- Job run type: Your job will complete after deploying to the devices and groups that you chose (snapshot)

The screenshot displays the AWS IoT console interface for configuring an OTA job. The breadcrumb trail is [AWS IoT](#) > [Jobs](#) > [Create job](#) > [OTA job](#). The page is titled "Step 3 of 3" and "OTA job configuration".

**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.

Buttons: Cancel, Back, **Create job**

**Successfully created Job:** [redacted]

[AWS IoT](#) > [Manage](#) > [Remote actions](#) > [Jobs](#)

**Jobs (200+)** Info [Refresh] [Edit] [Cancel] [Delete]

Jobs define a set of remote operations to send to and run on one or more devices that are connected to AWS IoT. If you have remote operations that are frequently performed, or installing new applications, use job templates to create reusable jobs.

Filter jobs: [input] All status values [dropdown] All types [dropdown] < 1 2 3 4

<input type="checkbox"/>	Name	Type	Status	Created date
<input type="checkbox"/>	<a href="#">AFR OTA-[redacted]</a>	Snapshot	In progress - Rollout completed...	November 16, 2023, 14:37:51 (UTC)

図 5-44 Step3: OTA job configuration

- ⑧ しばらく待つ（10 分程度待つ場合があります）と更新イメージをマイコンボードに書き込むログがターミナルに出力されます。

```
425 160746 [MQTT] [INFO] State record updated. New state=MQTTPublishDone.
426 160755 [MQTT] [INFO] Received data message callback, size 1048.

427 160765 [OTA Agent T] [INFO] Received valid file block: Block index=0, Size=0
428 160773 [OTA Agent T] W 0x59200, 256 ... OK

429 160782 [OTA Agent T] W 0x59300, 768 ... OK

430 160786 [OTA Agent T] [INFO] Number of blocks remaining: 231
431 161089 [MQTT] [INFO] De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
```

図 5-45 更新イメージをマイコンボードに書き込む

- ⑨ 書き込みが終了すると更新イメージ (バージョン 0.9.3) が起動しセンサボードから取得した温度データと湿度データを送信していることを確認します。

```
11 11943 [PUBSUB] [INFO] -----Start PubSub Demo task 0-----
10 11942 [OTA Demo Ta] [INFO] -----Start OTA Task-----

12 11955 [OTA Demo Ta] [INFO] OTA over MQTT demo. Application version 0.9.3
13 11961 [PUBSUB] [INFO] Sending subscribe request to agent for topic filter: iot[redacted]_tempdata
14 11971 [OTA Demo Ta] [INFO] Received: 0 Queued: 0 Processed: 0 Dropped: 0
15 11978 [OTA Agent T] [INFO] Current State=[RequestingJob], Event=[Start], New state=[RequestingJob]
16 12819 [PUBSUB] [INFO] Successfully subscribed to topic: iot[redacted]_tempdata
17 12934 [PUBSUB] [INFO] Sending publish request on topic "iot[redacted]_tempdata"
18 13767 [OTA Agent T] [INFO] Subscribed to topic $aws/things/iot[redacted]/jobs/notify-next.

19 13768 [OTA Agent T] [INFO] Subscribed to MQTT topic: $aws/things/iot[redacted]/jobs/notify-next
20 13867 [MQTT] [INFO] Publishing message to iot[redacted]_tempdata.

21 13978 [OTA Demo Ta] [INFO] Received: 0 Queued: 0 Processed: 0 Dropped: 0
22 14497 [PUBSUB] [INFO] Successfully sent QoS 0 publish to topic: iot[redacted]_tempdata (PassCount:1, FailCount:0).
23 14597 [MQTT] [INFO] Publishing message to $aws/things/iot[redacted]/jobs/$next/get.

24 15459 [MQTT] [INFO] De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
25 15460 [MQTT] [INFO] State record updated. new state=MQTTPublishDone.
26 15469 [MQTT] [INFO] Received incoming publish message {"temperature": 22.14, "humidity": 59.01, "warning": 0, "things": "iot[redacted]"}
27 15567 [MQTT] [INFO] Ack packet deserialized with result: MQTTSuccess.
```

図 5-46 書き込み終了後、更新イメージ (バージョン 0.9.3) が起動

## RL78/G23 LTE 通信での Amazon Web Services 接続でセンサー情報を可視化する方法: RL78/G23-128p Fast Prototyping Board + FreeRTOS

⑩ CloudWatch でグラフを表示します。

● CloudWatch > Logs Insights > “Run query”をクリック

— humidity: 湿度のグラフです。

— temperature: 温度のグラフです。

— warning: 温度のワーニングです。25 度未満の場合は False(値は 0)、25 度以上は True(値は 50)です。

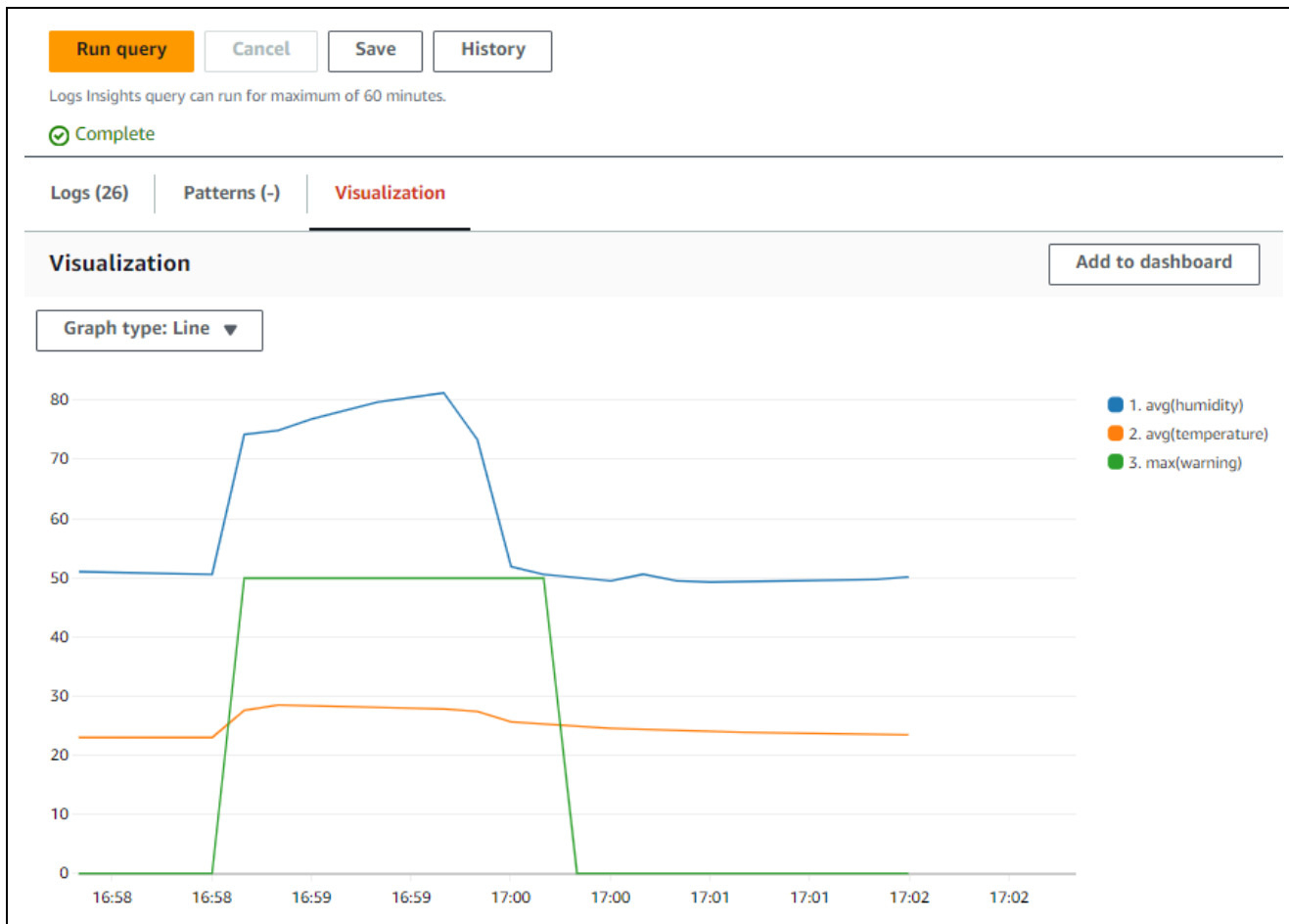


図 5-47 更新イメージ(バージョン 0.9.3)起動後に CloudWatch でグラフを表示

## 5.8 初期アプリケーションのデバッグ方法

初期アプリケーションを e2 studio から起動してデバッグする手順を示します。  
この手順ではブートローダを使用しないため、ダウンロードした更新イメージを起動することはできません。

① ブートローダを使用しない設定に変更します。

プロジェクト「aws\_ryz024a\_sensor\_rl78g23-fpb」の“USE\_BOOTLOADER\_V2”マクロを 0 に設定して“Apply and Close”を押します。

- Configuration: HardwareDebug
- Languages: GNU C
- USE\_BOOTLOADER\_V2: 0

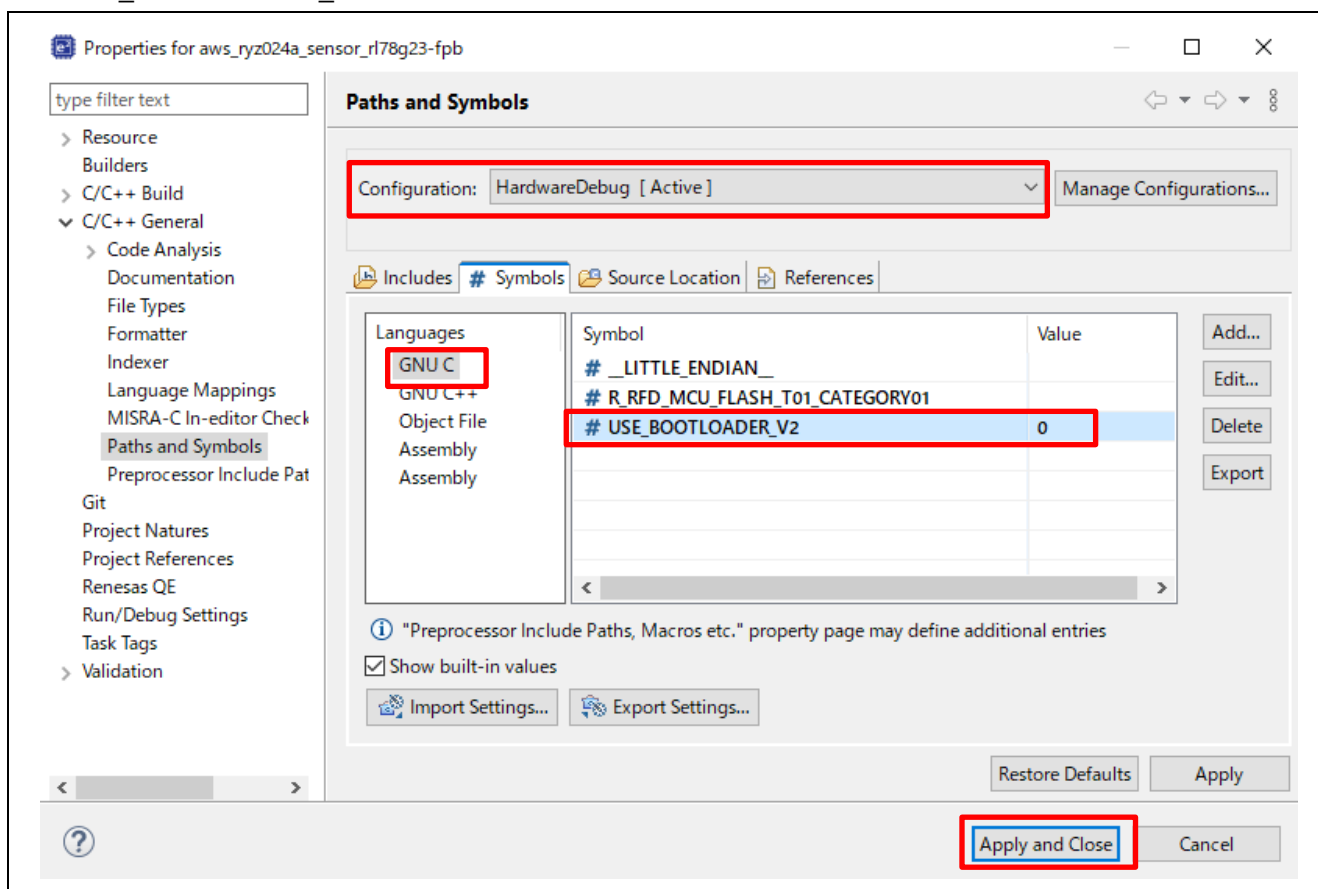


図 5-48 “USE\_BOOTLOADER\_V2”マクロを 0 に設定

② プロジェクト「aws\_ryz024a\_sensor\_rl78g23-fpb」をビルドします。

③ デバッグを開始します。

「7章 デバッグ手順」を参照してください。

## 6. Renesas Flash Programmer の使用方法

[Getting Started Guide](#) の 7. Using Renesas Flash Programmer を参照してください。

## 7. デバッグ手順

[Getting Started Guide](#) の 8. Debug Procedure を参照してください。

## 8. 付録

### 8.1 注意事項：サードパーティ製ライブラリを RL78 に移植する際の注意点

RL78 は 16bit システムのためサードパーティ製のライブラリを適用するときに下記の注意が必要です。

#### 8.1.1 Int の幅が 16bit

[Getting Started Guide](#) の 9.1.1 Width of int Is 16 Bits を参照してください。

#### 8.1.2 セクションのサイズ制限

[Getting Started Guide](#) の 9.1.2 Size Limitation of Section を参照してください。

### 8.2 デモプロジェクトで利用するオープンソースソフトウェアのライセンス情報

本製品のデモプロジェクトはオープンソースソフトウェア（OSS）を使用しており、ユーザは各 OSS が定めるライセンス条項を遵守する必要があります。ライセンス条項は各 OSS の公式 Web ページで確認してください。「表 1-4 動作確認条件 (その他、OSS ライブラリ等)」に各 OSS のリンク先を記載していません。

## 9. ウェブサイトおよびサポート

本 Getting Started Guide のサンプルプログラム: <https://github.com/renesas/iot-reference-rl78>

AWS forum: <http://forums.aws.amazon.com>

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Jul.31.24	-	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、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まれます。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）、交通制御（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なります。
  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（豊洲フォレスト）

[www.renesas.com](http://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/](http://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。