

## RL78/G23

### RL78/G23-128p Fast Prototyping Board で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化する方法

#### はじめに

Amazon FreeRTOS は、接続、セキュリティ、および無線（OTA）アップデートなどの FreeRTOS カーネルを強化するリアルタイムオペレーティングシステムです。Amazon FreeRTOS には、Amazon FreeRTOS 機能のデモを行うデモアプリケーションも含まれています。

e<sup>2</sup> studio は、オープンソースの Eclipse CDT(C/C++ Development Tooling)をベースとした開発環境で、デバッグのインターフェイスに加え、ビルド（エディタ、コンパイラ、リンク制御）をサポートしています。また、Amazon FreeRTOS のデモアプリケーションを統合し、ルネサス製ボード上での動作をサポートしています。

本ドキュメントでは、ルネサス製ボードとして、RL78/G23-128p Fast Prototyping Board と Wi-Fi モジュール（SX-ULPGN(Silex Technology 製）、温湿度センサーPmod ボード(US082-HS3001EVZ (Renesas 製))を組み合わせたシステムについて説明します。このシステムは、RL78/G23-128p に Amazon FreeRTOS を実装し、センサー情報（温度、及び湿度）を Wi-Fi 経由で Amazon Web Services（AWS）上で可視化できます。

#### 本ドキュメントの目的

本ドキュメントでは、e<sup>2</sup> studio を使用して Amazon FreeRTOS デモアプリケーションを実行する手順（Renesas GitHub Amazon FreeRTOS プロジェクトのダウンロードからデモの実行まで）について分かりやすく解説しています。

#### 動作環境

動作環境を以下に示します。

統合開発環境	e <sup>2</sup> studio 2021-07 (21.07.0) <a href="https://www.renesas.com/software-tool/e-studio">https://www.renesas.com/software-tool/e-studio</a>
ボード	RL78/G23-128p Fast Prototyping Board <a href="https://www.renesas.com/rl78g23-128p_fpb">https://www.renesas.com/rl78g23-128p_fpb</a>  Wi-Fi Pmod 拡張ボード <a href="https://www.renesas.com/wi-fi-pmod-expansion-Board">https://www.renesas.com/wi-fi-pmod-expansion-Board</a> 温湿度センサーPmod ボード US082-HS3001EVZ <a href="https://www.renesas.com/us/ja/products/sensor-products/humidity-sensors/us082-hs3001evz-relative-humidity-sensor-pmod-board-renesas-quick-connect-iot">https://www.renesas.com/us/ja/products/sensor-products/humidity-sensors/us082-hs3001evz-relative-humidity-sensor-pmod-board-renesas-quick-connect-iot</a> DIGILENT Pmod USBUART <a href="https://store.digilentinc.com/pmod-usbuart-usb-to-uart-interface/">https://store.digilentinc.com/pmod-usbuart-usb-to-uart-interface/</a>
ツールチェーン	CCRL Compiler v1.10.00 <a href="https://www.renesas.com/software-tool/c-compiler-package-rl78-family">https://www.renesas.com/software-tool/c-compiler-package-rl78-family</a>
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>

## 目次

1. 概要	4
1.1 システム図	4
2. Amazon FreeRTOS プロジェクトの準備	5
2.1 GitHub からソースコードをダウンロード	5
2.2 本アプリケーションノート付属のサンプルコード	5
2.3 デモプロジェクトのインポート	6
3. AWS の準備	9
4. ハードウェアの準備	10
4.1 RL78/G23-128p Fast Prototyping Board	10
4.2 SX-ULPGN	10
4.3 US082-HS3001EVZ	11
4.4 DIGILENT Pmod USBUART	11
4.5 証明書の書き込み	11
4.5.1 SharkSSL のダウンロード	11
4.5.2 証明書データの入手	11
4.5.3 CA リスト (Class 2 Root CA) の入手	12
4.5.4 証明書と秘密鍵を SharkSSL バイナリフォーマットへ変換	15
4.5.5 CA リストを SharkSSLPerseCAList バイナリフォーマットへ変換	15
4.5.6 SX-ULPGN への証明書の書き込み	15
4.5.7 SX-ULPGN の接続	18
4.6 デバッグログ受信のための準備	19
5. デモプロジェクトの準備	21
6. Amazon OpenSearch Service の準備	23
7. Kibana の準備	29
8. IoT Rule の準備	34
9. デモプログラムの実行	42
10. Kibana でセンサー情報を可視化	43
11. デモプログラム実行後の注意事項について	52
12. 付録	53
12.1 ドライバのバージョンアップ方法について	53
改訂記録	60

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>)

Pmod は Digilent Inc.の商標です。 (<https://store.digilentinc.com/>)

## 1. 概要

本ドキュメントでは、ルネサス RL78/G23-128p Fast Prototyping Board で Amazon FreeRTOS のプロジェクト用意からデモを実行するまでを説明します。

### 1.1 システム図

以下に、温度、湿度のセンサー情報を取得してから、可視化するまでのシステム図を示します。

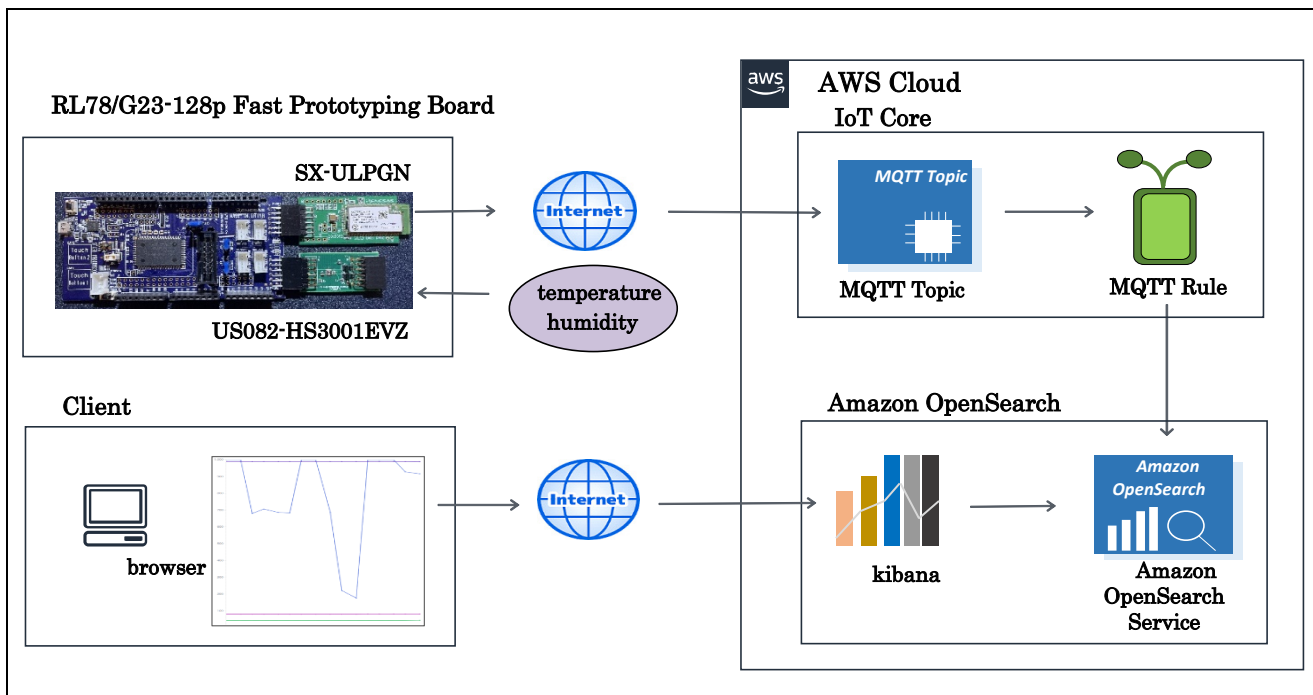


図 1.1 センサー情報を取得してから、可視化するまでのシステム図

## 2. Amazon FreeRTOS プロジェクトの準備

本ドキュメントの Amazon FreeRTOS プロジェクトは、GitHub の以下リポジトリからダウンロードできます。

GitHub リポジトリ

[https://github.com/renesas/amazon-freertos/tree/rl78\\_developmment\\_202012.00\\_sensor](https://github.com/renesas/amazon-freertos/tree/rl78_developmment_202012.00_sensor)

または、本アプリケーションノート付属の zip ファイルに同梱されている amazon-freertos.zip が使用できます。

### 2.1 GitHub からソースコードをダウンロード

GitHub からソースコードをダウンロードし、プロジェクトを e<sup>2</sup> studio のワークスペースにインポートする必要があります。

GitHub からのダウンロードは、Git を使用します。

本ドキュメントでは、Git for Windows (<https://gitforwindows.org/>) の使用を推奨しています。

GitHub からのダウンロード例について、以下に示します。

1. マスターブランチのクローンを作成します。

```
git clone https://github.com/renesas/amazon-freertos.git
```

**注：**クローンしたファイルは C ドライブ直下など、フォルダ階層が浅い階層に配置してください。フォルダパスが長い場合、ビルドエラーになる場合があります。

2. カレントディレクトリをクローンしたディレクトリのルートに変更します。

```
cd amazon-freertos
```

3. リリースタグをチェックアウトします。

```
git checkout v202012.00-rl78-1.0.0-sensor
```

4. サブモジュールを更新します。

```
git submodule update --init --recursive
```

### 2.2 本アプリケーションノート付属のサンプルコード

本アプリケーションノート付属の zip ファイルに同梱されている amazon-freertos.zip を使用できます。

zip ファイルを解凍してください。

**注：**解凍したファイルは C ドライブ直下など、フォルダ階層が浅い階層に配置してください。フォルダパスが長い場合、ビルドエラーになる場合があります。

## 2.3 デモプロジェクトのインポート

e<sup>2</sup> studio で RL78/G23-128p のデモプロジェクトをインポートします。

1. e<sup>2</sup>-studio を起動しワークスペースを指定します。

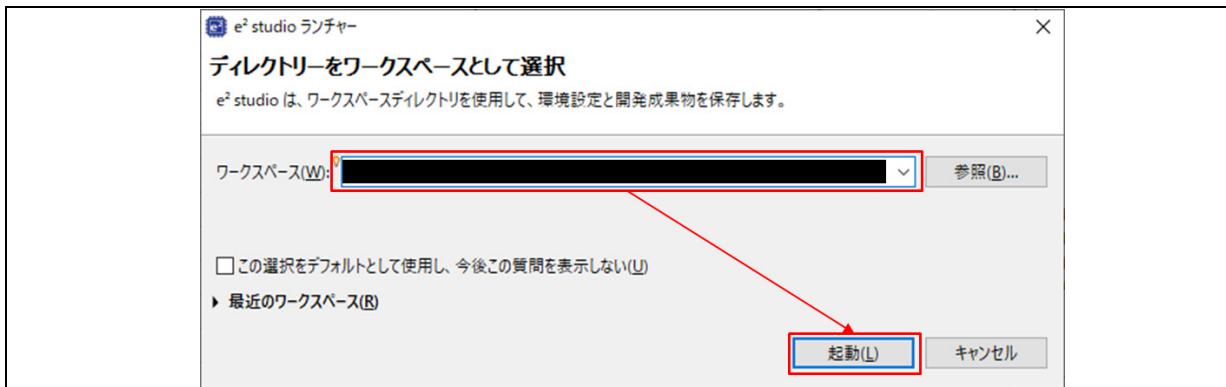


図 2.1 ワークスペース選択画面

2. ファイル -> インポート をクリックします。

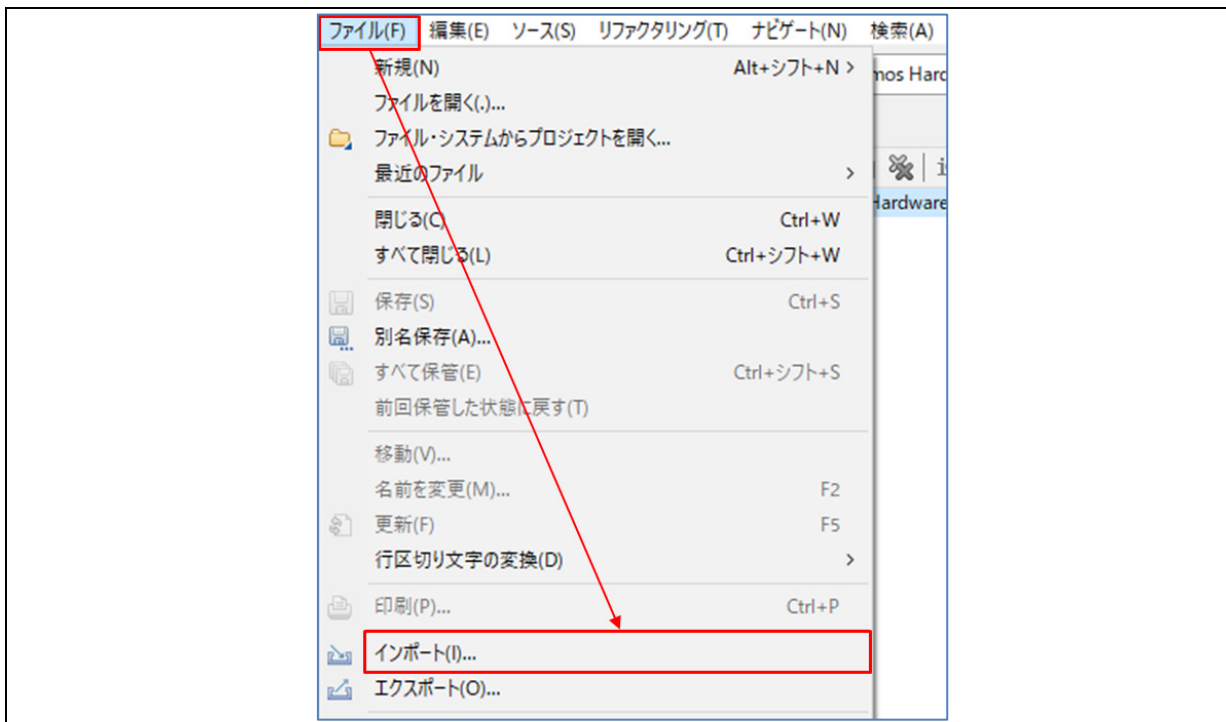


図 2.2 インポート を選択

3. 一般 -> 既存プロジェクトをワークスペースへ -> 次へ をクリックします。

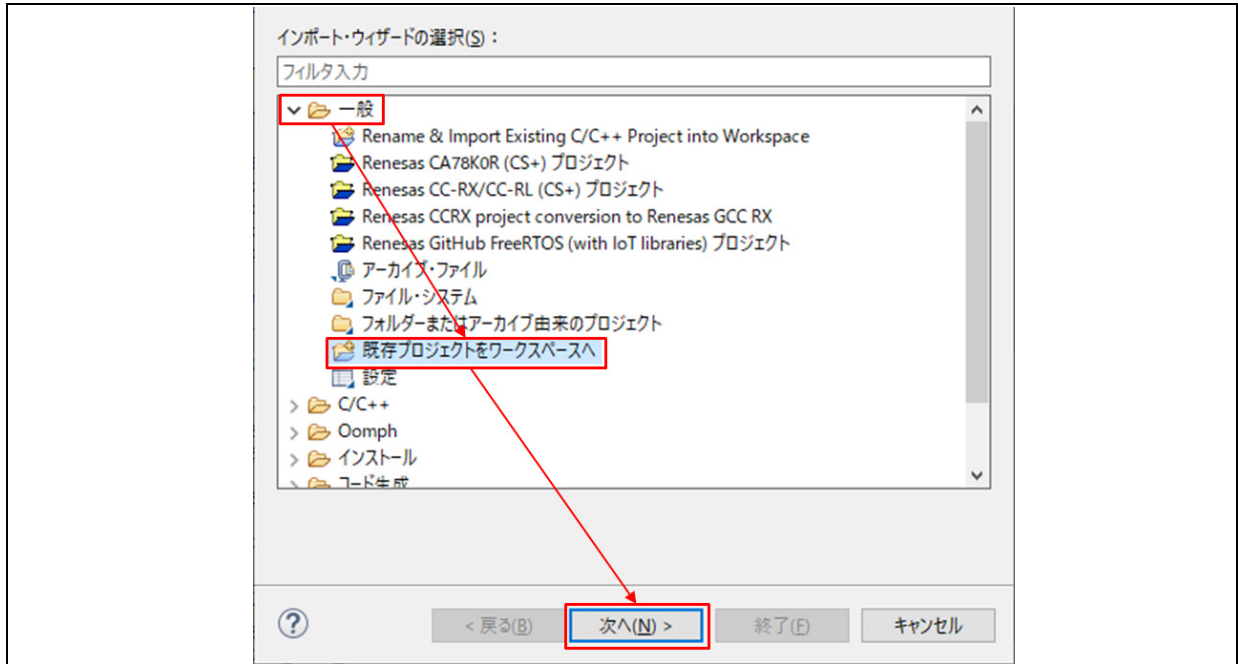


図 2.3 既存プロジェクトをワークスペースへ を選択

## RL78/G23 RL78/G23-128p Fast Prototyping Board で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化する方法

4. [参照]で、以下のルートフォルダを指定します。その後、「終了」をクリックします。

projects -> renesas -> rl78g23-fpb-sx-ulpgn -> e2studio -> aws\_demo

注：「プロジェクトをワークスペースにコピー」にチェックをいれないでください。

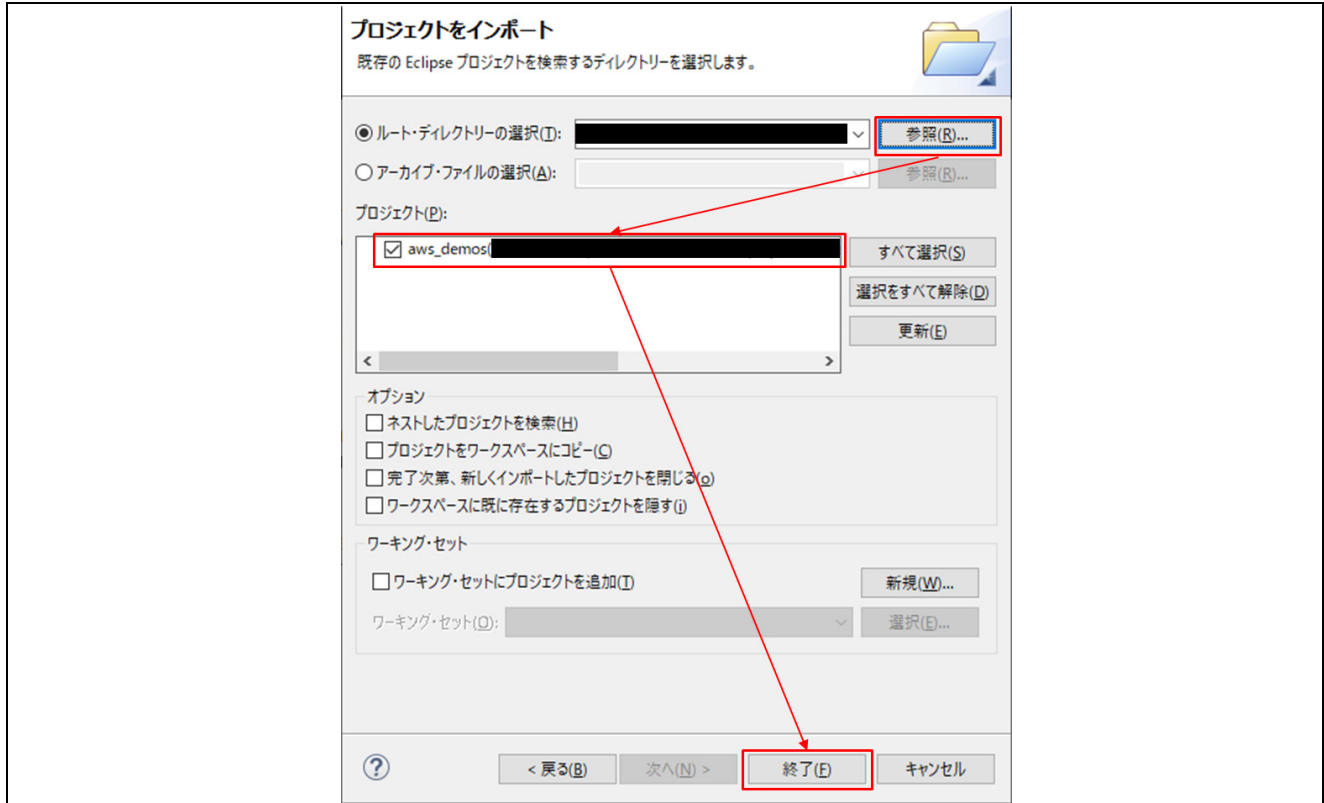


図 2.4 デモプロジェクトのインポート



### 3. AWS の準備

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。

- ・デバイスを AWS IoT に登録する

[https://github.com/renesas/amazon-freertos/wiki/デバイスを AWS-IoT に登録する](https://github.com/renesas/amazon-freertos/wiki/デバイスを-AWS-IoT-に登録する)

また、デモプロジェクトの `aws_demos -> demos -> include -> aws_clientcredential.h` にある 4 つのマクロを設定してください。

- ・ `clientcredentialMQTT_BROKER_ENDPOINT` -> 「デバイスを AWS IoT に登録する」で確認したエンドポイント の名前
- ・ `clientcredentialIOT_THING_NAME` -> 「デバイスを AWS IoT に登録する」で登録した モノ の名前
- ・ `clientcredentialWIFI_SSID` (Wi-Fi 利用の場合) -> 接続するアクセスポイントの SSID
- ・ `clientcredentialWIFI_PASSWORD` (Wi-Fi 利用の場合) -> 接続するアクセスポイントのパスワード

```
+ * FreeRTOS V202002.00[]
- #ifndef _AWS_CLIENTCREDENTIAL_H_
  #define _AWS_CLIENTCREDENTIAL_H_

+ * @brief MQTT Broker endpoint.[]
  #define clientcredentialMQTT_BROKER_ENDPOINT ""

+ * @brief Host name.[]
  #define clientcredentialIOT_THING_NAME ""

+ * @brief Port number the MQTT broker is using.[]
  #define clientcredentialMQTT_BROKER_PORT 8883

+ * @brief Port number the Green Grass Discovery use for JSON retrieval from cloud is using.[]
  #define clientcredentialGREENGRASS_DISCOVERY_PORT 8443

+ * @brief Wi-Fi network to join.[]
  #define clientcredentialWIFI_SSID ""

+ * @brief Password needed to join Wi-Fi network.[]
  #define clientcredentialWIFI_PASSWORD ""
```

図 3.1 マクロの設定

## 4. ハードウェアの準備

デモプロジェクトを実行するためのハードウェアの準備をします。

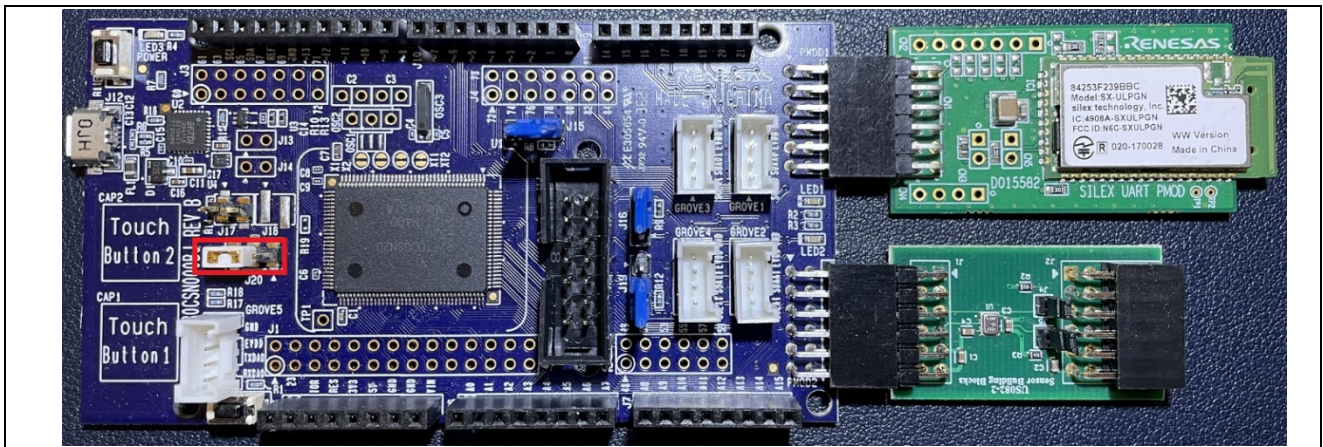


図 4.1 RL78/G23-128p Fast Prototyping Board と Wi-Fi モジュール (SX-ULPGN(Silex Technology 製))

### 4.1 RL78/G23-128p Fast Prototyping Board

デモプロジェクトを実行する場合、RL78/G23-128p Fast Prototyping Board が必要です。

以下の URL から購入できます。

[https://www.renesas.com/rl78g23-128p\\_fpb](https://www.renesas.com/rl78g23-128p_fpb)

電源選択ヘッダ (写真の赤枠 J20) の 2-3 をショートし電源を 3.3V に設定してください。

またデモ プロジェクトは E2 Lite エミュレータで接続するため、エミュレータ用コネクタの実装が必要です。実装方法については「RL78/G23-128p Fast Prototyping Board ユーザーズマニュアル (エミュレータ用コネクタ)」(R20UT4870xxxxxx) を参照ください。

<https://www.renesas.com/document/mat/rl78g23-128p-fast-prototyping-board-users-manual-rev100?language=ja&r=1537821>

### 4.2 SX-ULPGN

RL78/G23-128p Fast Prototyping Board に接続する無線 LAN モジュールが必要です。Amazon FreeRTOS の RL78/G23-128p のデモは SX-ULPGN を使って動作の確認をしています。以下の URL から購入できます。

<https://www.renesas.com/products/software-tools/boards-and-kits/eval-kits/wi-fi-pmod-expansion-board.html>

### 4.3 US082-HS3001EVZ

RL78/G23-128p Fast Prototyping Board の Pmod コネクタに接続する温湿度センサーモジュールです。Amazon FreeRTOS の RL78/G23-128p のデモは US082-HS3001EVZ を使って動作の確認をしています。

以下の URL から購入できます。

<https://www.renesas.com/us/ja/products/sensor-products/humidity-sensors/us082-hs3001evz-relative-humidity-sensor-pmod-board-renesas-quick-connect-iot>

### 4.4 DIGILENT Pmod USBUART

DIGILENT Pmod USBUART は SX-ULPGN への証明書と CA リストの書き込みと RL78/G23-128p のデモを実行するときのデバッグログの受信に使用します。

以下の URL から購入できます。

<https://store.digilentinc.com/pmod-usbuart-usb-to-uart-interface/>

### 4.5 証明書の書き込み

SX-ULPGN に証明書と CA リストを書き込みます。SX-ULPGN へ書き込みを行う証明書データは SharkSSLParseCert バイナリフォーマット、CA リストは SharkSSLPerseCAList バイナリフォーマットに変換しておく必要があります。

以下に、Tera term からの証明書の書き込み手順を示します。

#### 4.5.1 SharkSSL のダウンロード

フォーマット変換は下記のフリーソフトウェアで行うことができます。

SharkSSL <<https://realtimelogic.com/downloads/sharkssl/>>

「SharkSSL for Windows」を選択します。ダウンロード、インストールはソフトウェアの指示に従ってください。

#### 4.5.2 証明書データの入手

「3. AWS の準備」で入手した証明書と秘密鍵を使用します。

#### 4.5.3 CA リスト (Class 2 Root CA) の入手

1. Microsoft Edge -> 設定 -> プライバシー、検索、サービス -> 証明書の管理 -> 証明書 -> 信頼されたルート証明機関 を選択し、「Starfield Class 2 Certification Authority」をエクスポートします。

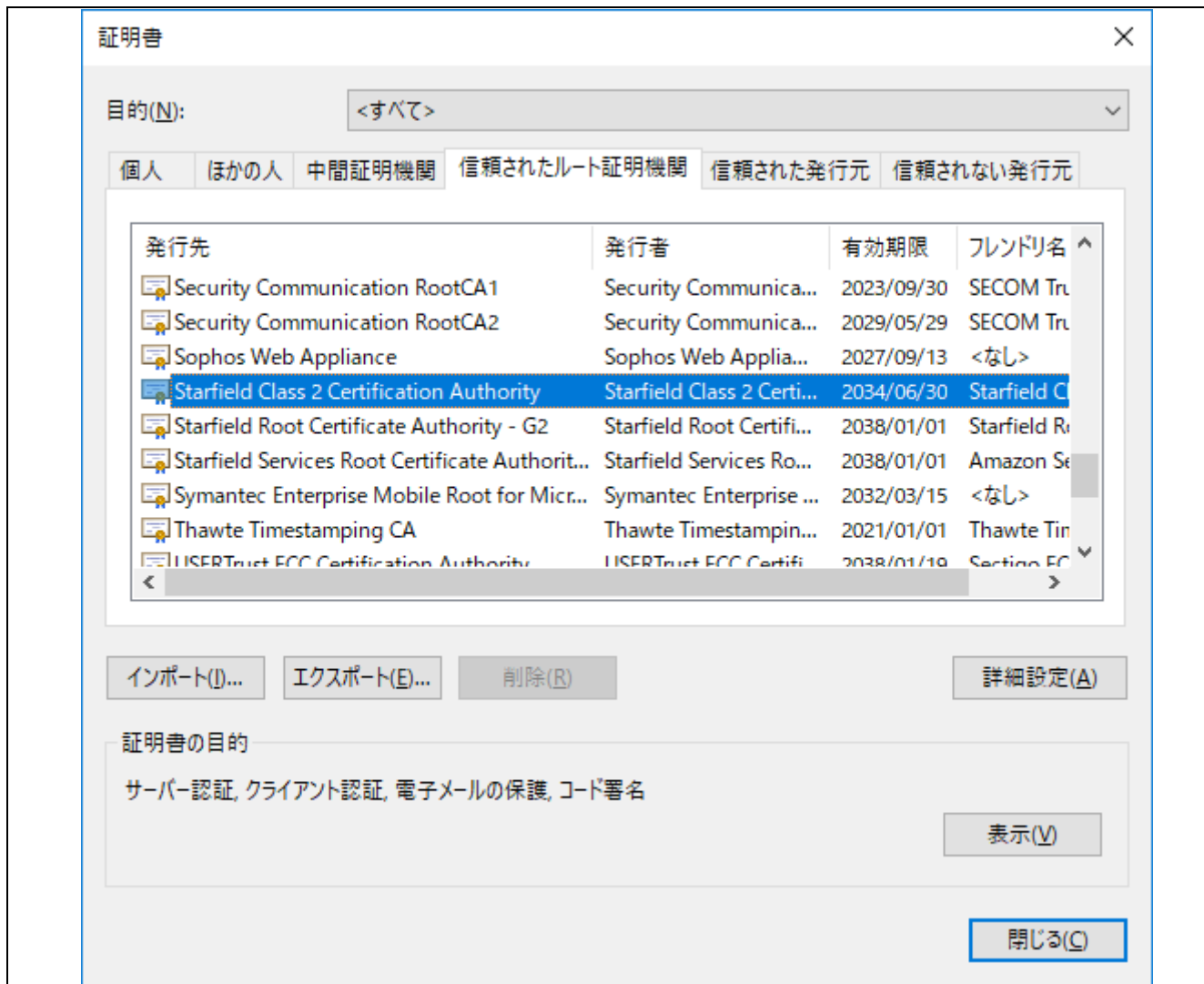


図 4.2 CA リストの入手

2. Base 64 encoded X.509 (.CER)を選択します。

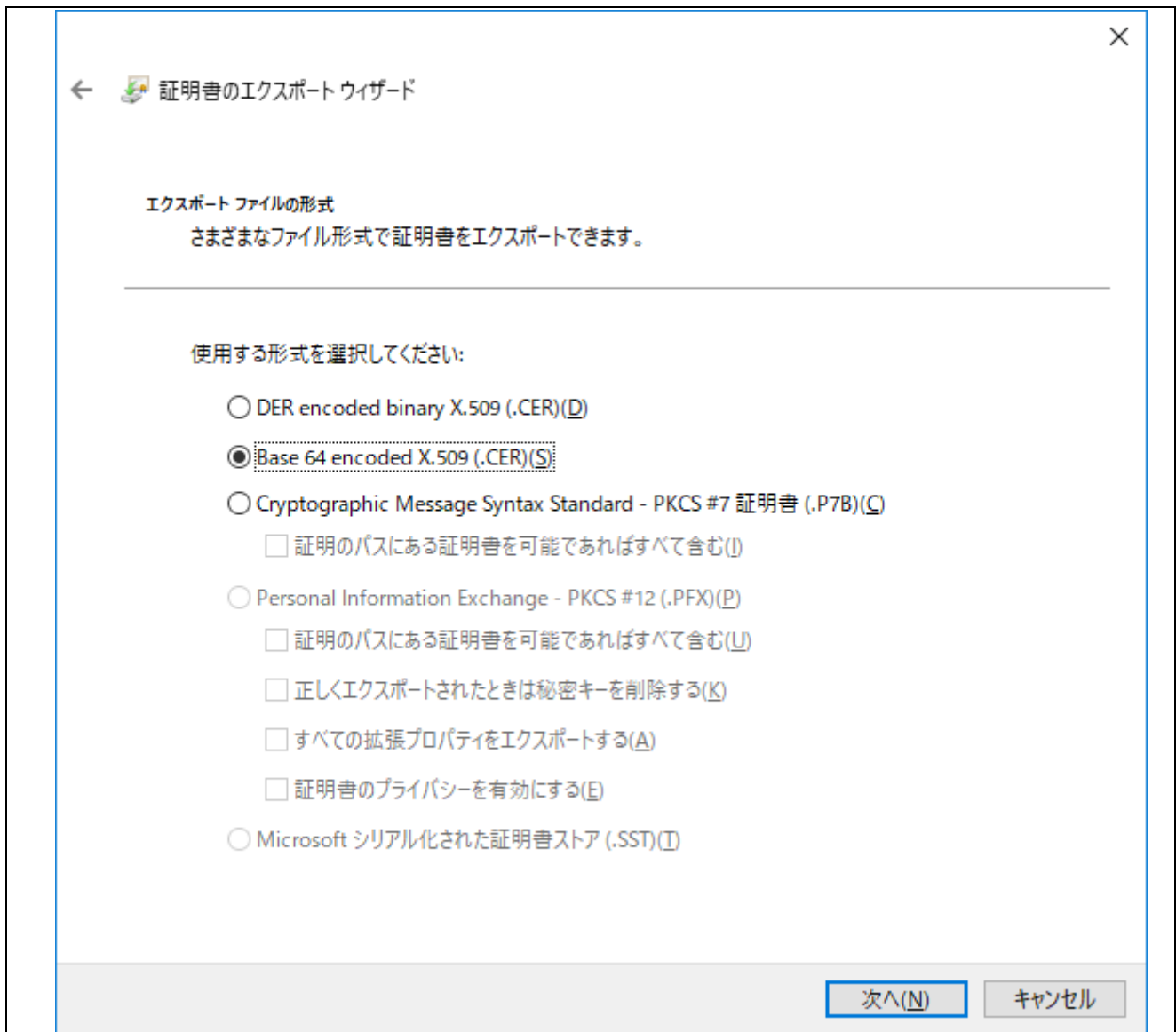


図 4.3 Base 64 encoded X.509 (.CER)を選択

3. ファイル名を「calist1」と入力し、証明書をエクスポートします。エクスポートされたファイルは拡張子 c が追加され、「calist1.cer」になります。

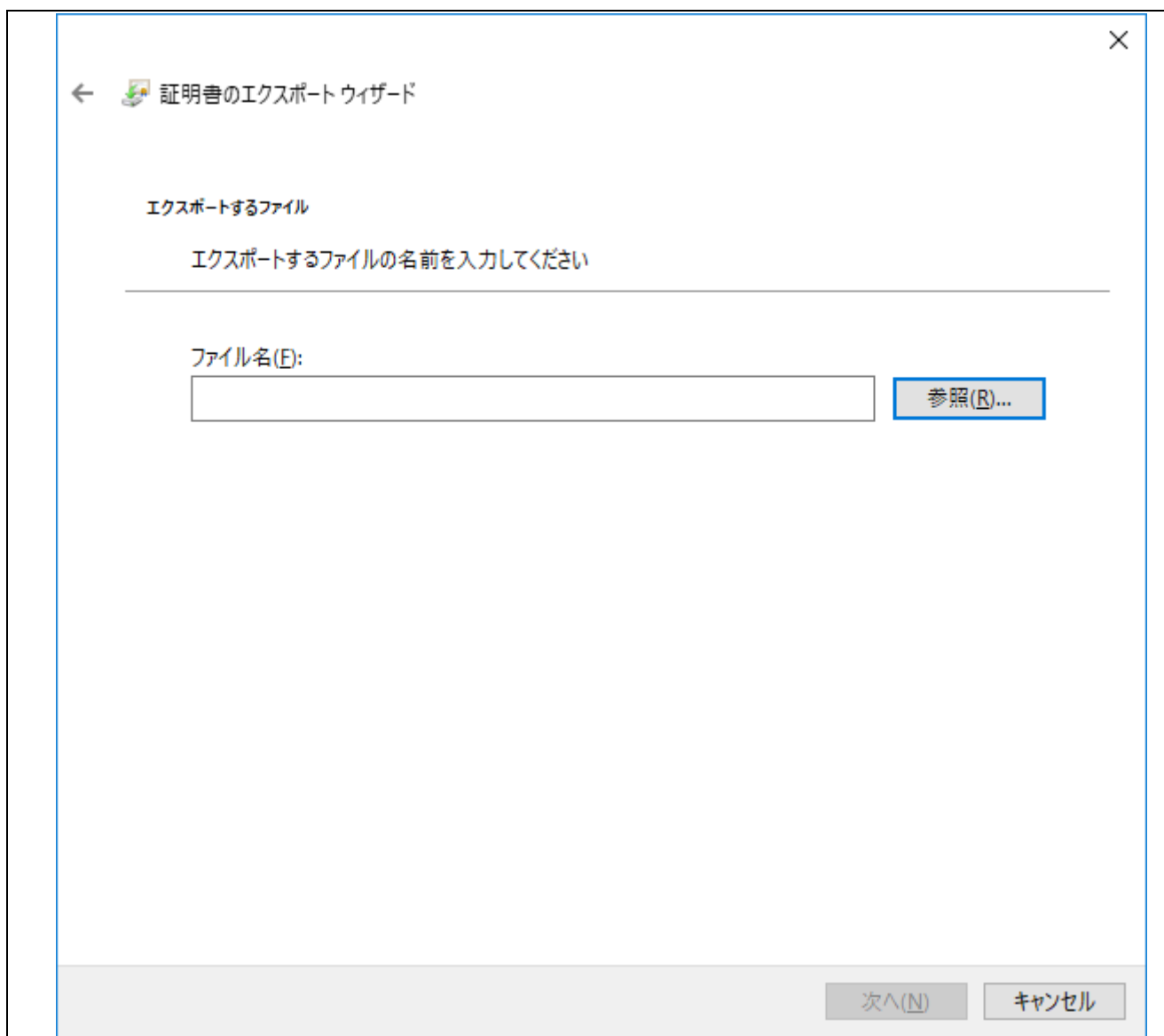


図 4.4 証明書のエクスポート

#### 4.5.4 証明書と秘密鍵を SharkSSL バイナリフォーマットへ変換

1. コマンドプロンプト上で、以下を実行して証明書と秘密鍵を SharkSSI バイナリフォーマット変換します。

```
SharkSSLParseCert xxxxx-certificate.pem.crt xxxxx-private.pem.key -b cert1.bin
```

注：

xxxxx は「3 AWS の準備」で AWS から入手した秘密鍵、デバイス証明書のファイル名にある "d666c26201" のようなハッシュ値の断片です  
「cert1」という正確な名前である必要があります

#### 4.5.5 CA リストを SharkSSLPerseCAList バイナリフォーマットへ変換

1. コマンドプロンプト上で、以下を実行して CA リストを SharkSSLPerseCAList バイナリフォーマット変換します。

```
SharkSSLParseCAList.exe -b calist1.bin yyyyy.cer
```

注：

「calist1」という正確な名前である必要があります  
yyyyy は図 4.4 証明書のエクスポート にて作成した CA ファイルの名前です

#### 4.5.6 SX-ULPGN への証明書の書き込み

変換した証明書と CA リスト（バイナリファイル）を SX-ULPGN に書き込みます。USB シリアル変換を介して PC と Wi-Fi モジュールの TX,RX 端子を接続し、AT コマンドを用いて書き込みを行います。この際、ボーレートは 115200bps としてください。

以下に、例としてターミナルエミュレータ（Tera Term）を使用して証明書と CA リストを書き込むための設定を示します。

このとき、Tera Term は Version 4.105 以上を使用してください。

##### [設定タブのシリアルポートの設定]

- ボーレート: 115200bps
- データ: 8 bit
- パリティ: none
- ストップ: 1 bit
- フロー制御: none

##### [設定タブの端末の設定]

- 改行コード 受信: CR
- 改行コード 送信: CR
- ローカルエコー: チェックを外す



## RL78/G23 RL78/G23-128p Fast Prototyping Board で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化する方法

以下に DIGILENT Pmod USBUART と SX-ULPGN の接続例を示します。

SX-ULPGN 側のコネクタは 2 段あります。DIGILENT Pmod USBUART からの配線は SX-ULPGN の上段に繋いでください。

- DIGILENT Pmod USBUART のジャンパ VCC、SYS をショート (DIGILENT Pmod USBUART から SX-ULPGN に電源供給)
- DIGILENT Pmod USBUART 2pin (RxD) と SX-ULPGN 3pin (TxD)
- DIGILENT Pmod USBUART 3pin (TxD) と SX-ULPGN 2pin (RxD)
- DIGILENT Pmod USBUART 5pin (GND) と SX-ULPGN 5pin (GND)
- DIGILENT Pmod USBUART 6pin (VCC) と SX-ULPGN 6pin (VCC)

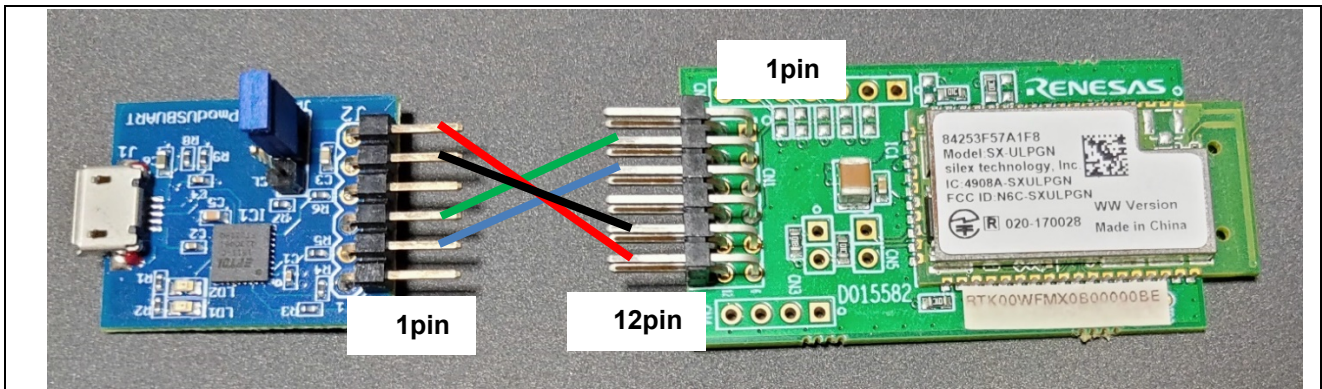


図 4.5 DIGILENT Pmod USBUART と SX-ULPGN の接続



以下に、ターミナルエミュレータ（Tera Term）を使った証明書の登録方法を示します。

1. 以下のコマンドを実行します。

`ATNSSLCERT=cert1.crt,<変換した証明書のバイナリファイルサイズ>`

例：`ATNSSLCERT=cert1.crt,1768`

2. 30 秒以内に、Tera Term のファイル送信から「証明書と秘密鍵を SharkSSL バイナリフォーマットへ変換」で変換したバイナリファイルを送信します。

※この時、オプションのバイナリにチェックを入れます。

※PC 上のファイル名の拡張子は \*.bin であることに注意してください。

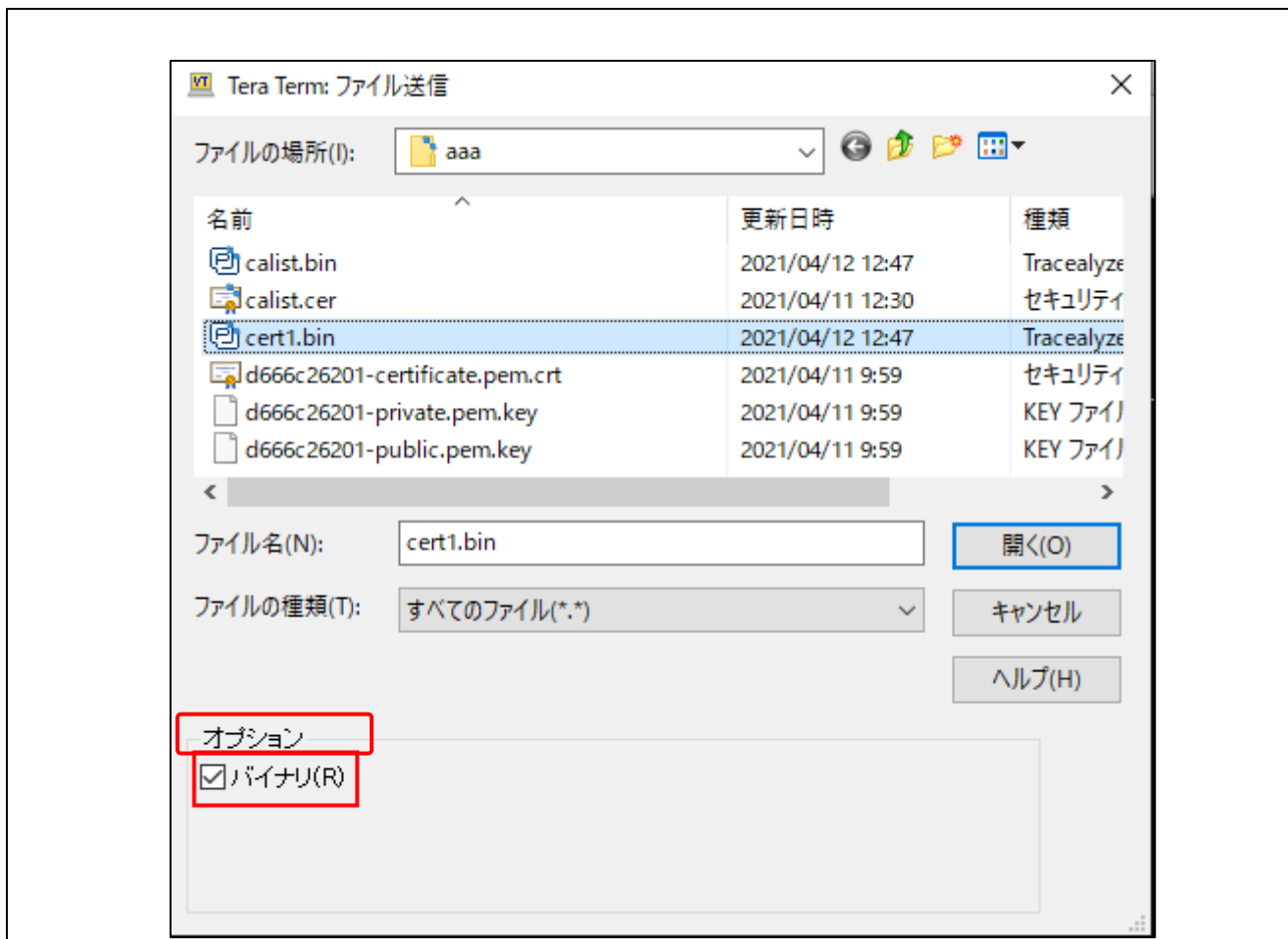


図 4.6 証明書の登録

3. 以下のコマンドを実行します。

`ATNSSLCERT= calist1.crt,<変換した CA リストのバイナリファイルサイズ>`

例：`ATNSSLCERT=calist1.crt,1059`

4. 30 秒以内に、Tera Term のファイル転送から「CA リストを SharkSSLPerseCAList バイナリフォーマットへ変換」で変換したバイナリファイルを送信します。

※オプションのバイナリにチェックを入れます。

5. 「ATNSSLCERT=?」 コマンドを実行し、以下が表示されることを確認します。

calist1.crt

cert1.crt

補足：誤って証明書を登録した場合、「ATNSSLCERT=ファイル名,0」とコマンドを実行することで、登録した証明書を削除することができます。

## 4.5.7 SX-ULPGN の接続

RL78/G23-128p Fast Prototyping Board の PMOD1 に SX-ULPGN を接続します。

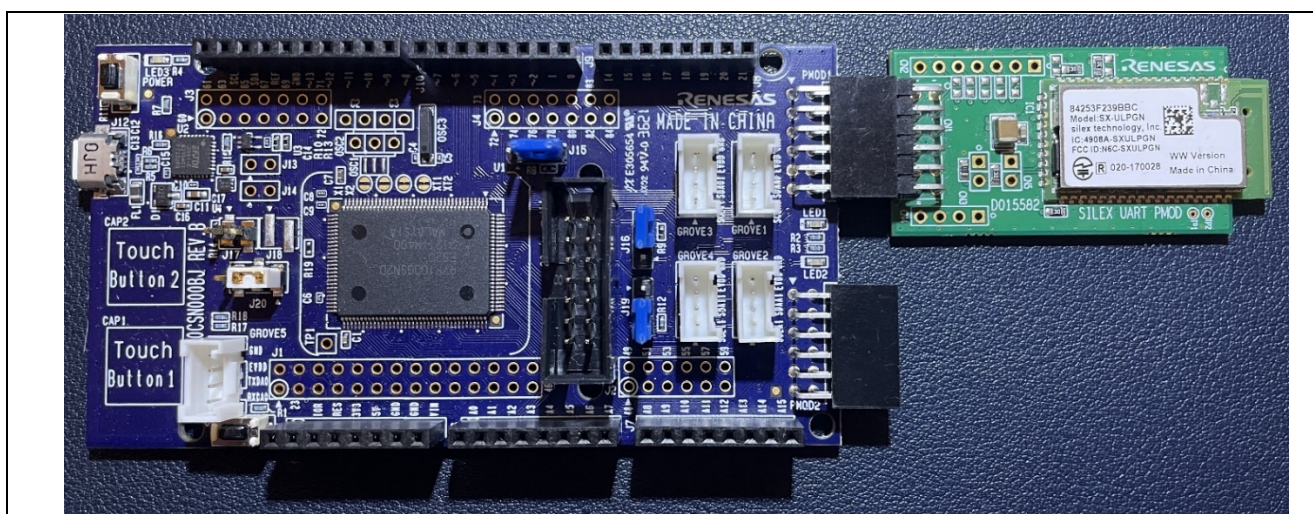


図 4.7 RL78/G23-128p Fast Pototyping Board に SX-ULPGN を接続



#### 4.6 デバッグログ受信のための準備

デバッグログを確認する場合は、ターミナルエミュレータ（Tera Term 等）を SCI ドライバで使用しているシリアルポートに接続します。

以下に DIGILENT Pmod USBUART と RL78/G23-128p Fast Prototyping Board の接続例を示します。

- DIGILENT Pmod USBUART 2pin (RxD) と RL78/G23-128p Fast Prototyping Board J8 4 pin (TxD)
- DIGILENT Pmod USBUART 5pin (GND) と RL78/G23-128p Fast Prototyping Board J5 6 または 7pin (GND)

RL78/G23-128p Fast Prototyping Board は PC から USB ケーブルを介して電源供給されているため、DIGILENT Pmod USBUART からの電源供給は不要です。また、デバッグログは受信のみ必要であるため、DIGILENT Pmod USBUART からの送信も不要です。

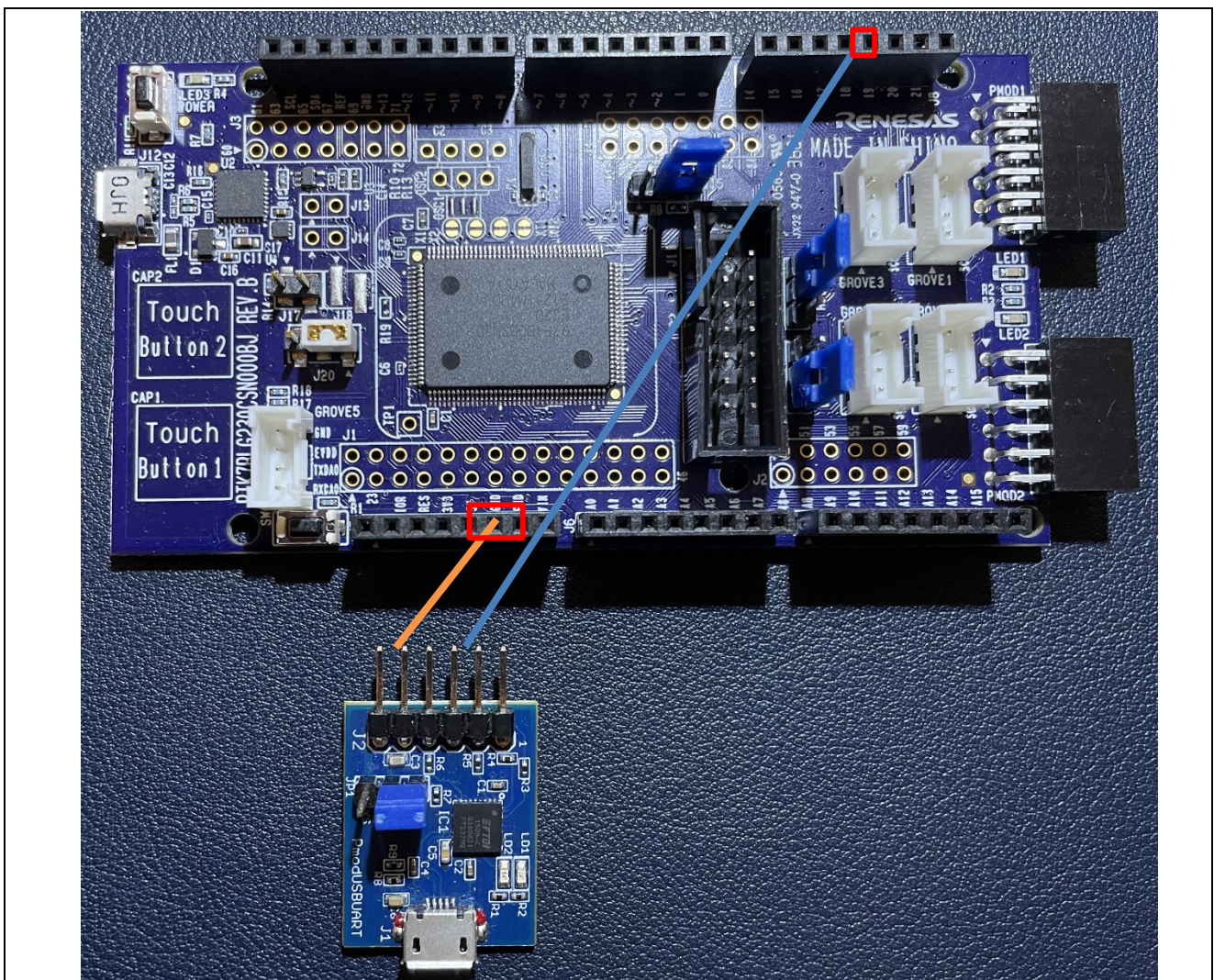


図 4.8 DIGILENT Pmod USBUART と RL78/G23-128p Fast Prototyping Board の接続

デバッグログの受信に Tera Term を使用する場合は Version 4.105 以上を使用してください。

以下に、Tera Term の設定を示します。

[設定タブのシリアルポートの設定]

- ボーレート: 115200bps
- データ: 8 bit
- パリティ: none
- ストップ: 1 bit
- フロー制御: none

[設定タブの端末の設定]

- 改行コード 受信: CR
- 改行コード 送信: CR
- ローカルエコー : チェックを外す

## 5. デモプロジェクトの準備

以下の手順でデモをビルドおよび実行します。

1. Project Explorer のプロジェクトを右クリックし、「ビルド」を選択します。
2. メニューより、[実行] → [デバッグの構成]を選択します。
3. Renesas GDB Hardware Debugging を拡張し、aws\_demos HardwareDebug を選択します。

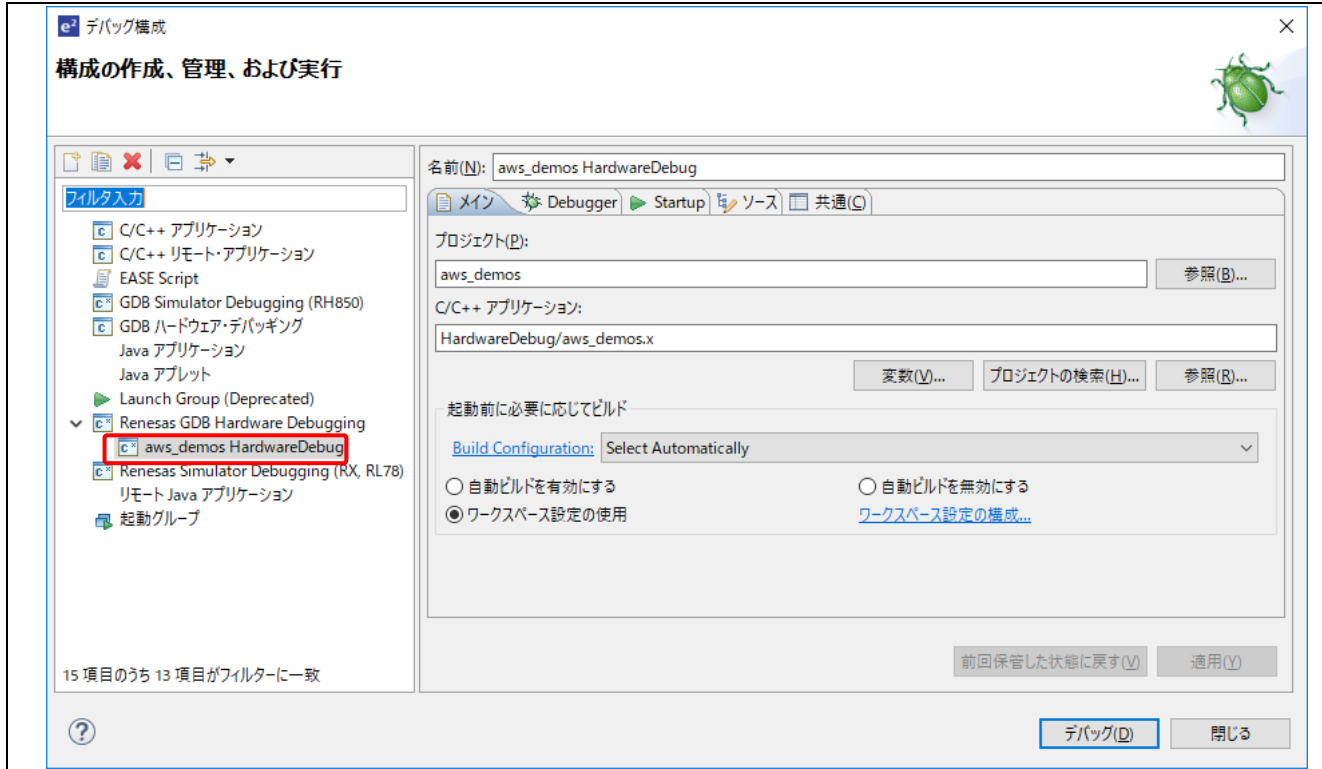


図 5.1 起動設定を選択

# RL78/G23 RL78/G23-128p Fast Prototyping Board で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化する方法

4. Debugger タブ→Connection Settings タブの順に選択します。接続設定が正しいか確認してください。

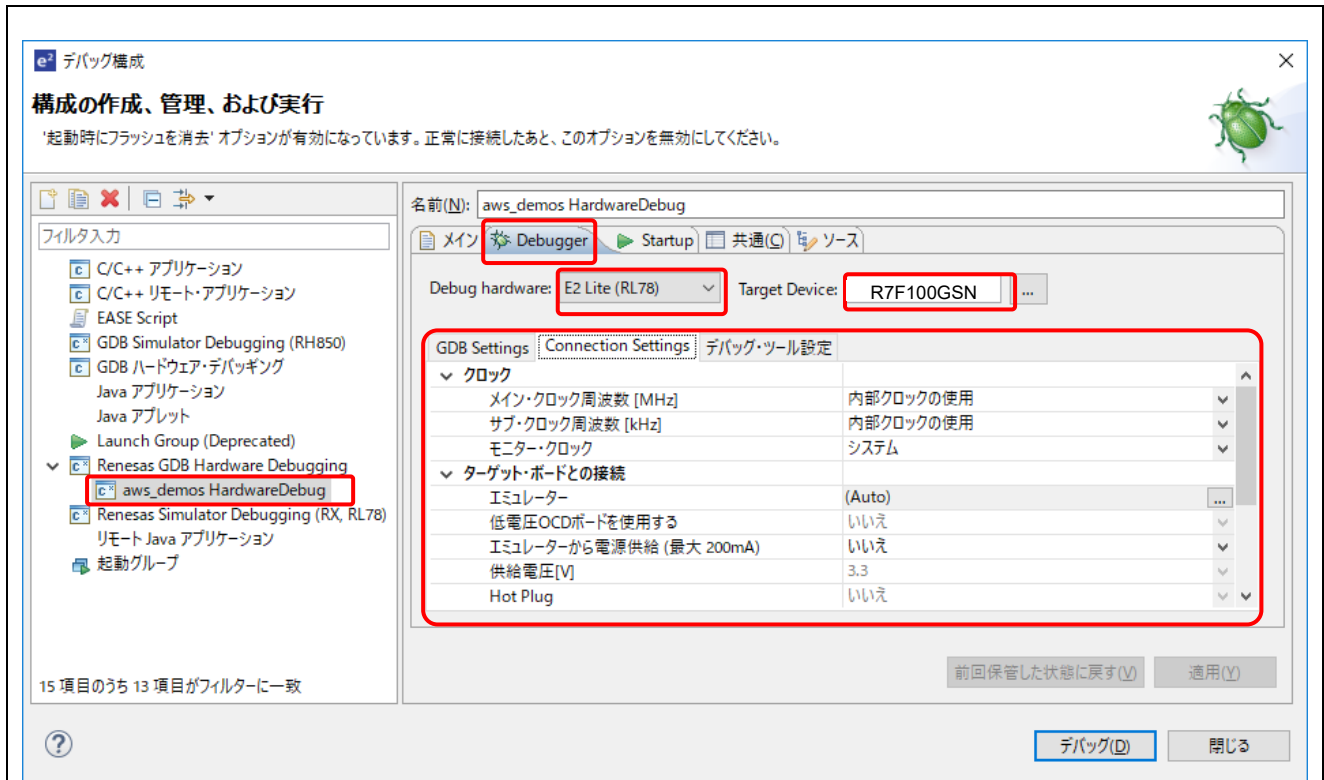


図 5.2 ハードウェアデバッグ設定



## 6. Amazon OpenSearch Service の準備

RL78/G23-128p Fast Prototyping Board に接続されたセンサーモジュールから取得したデータを、Amazon OpenSearch Service を使用して AWS で可視化します。

**Amazon OpenSearch Service は起動していると料金が発生します。  
デモプログラムを終了した際はドメイン削除を忘れずに行ってください。**

以下の手順で Amazon OpenSearch Service を設定します。最新の設定項目は以下の画像と異なる場合があります。示されていない項目はデフォルトで設定してください。

1. Amazon OpenSearch Service のドメインを作成します。AWS マネジメントコンソールから、サービス -> 分析 -> Amazon OpenSearch Service をクリックします。

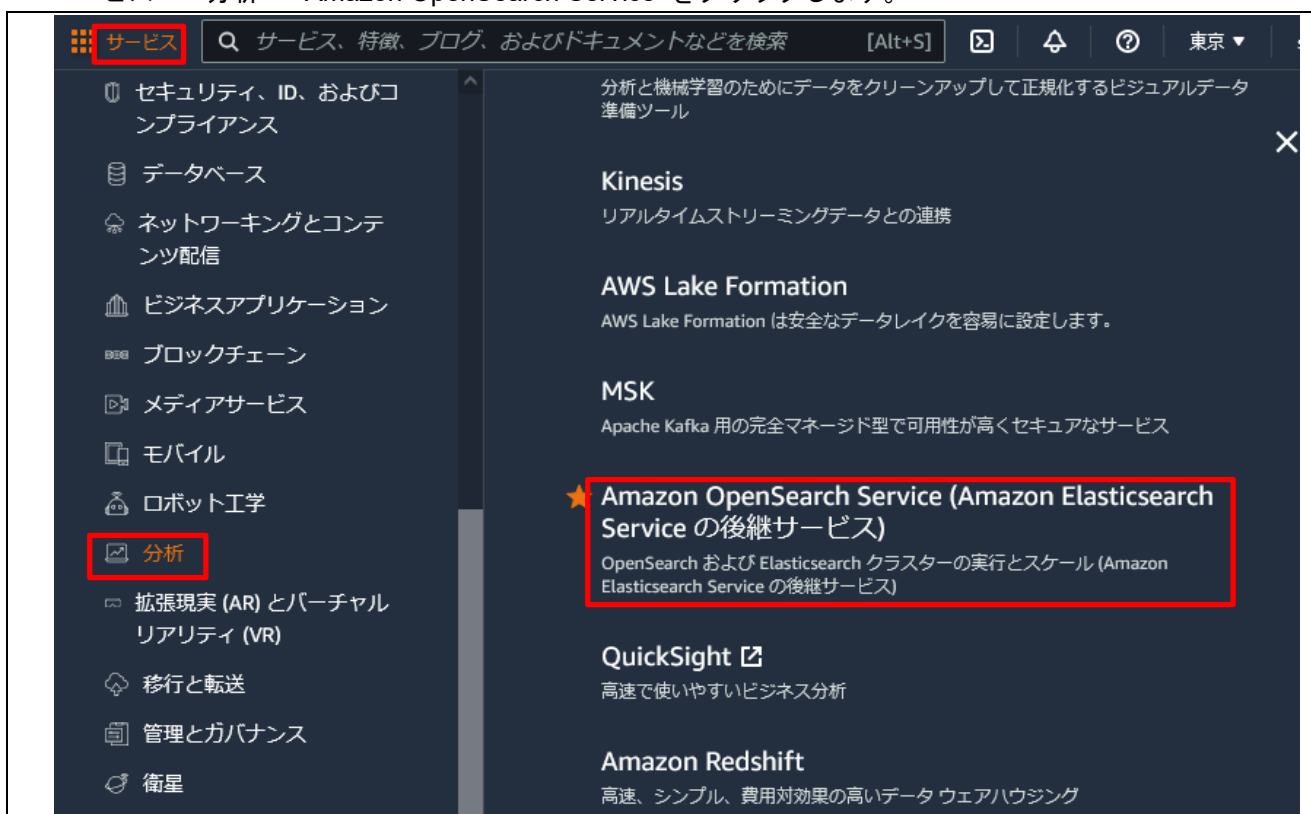


図 6.1 Amazon OpenSearch Service を選択

2. ドメインの作成をクリックします。

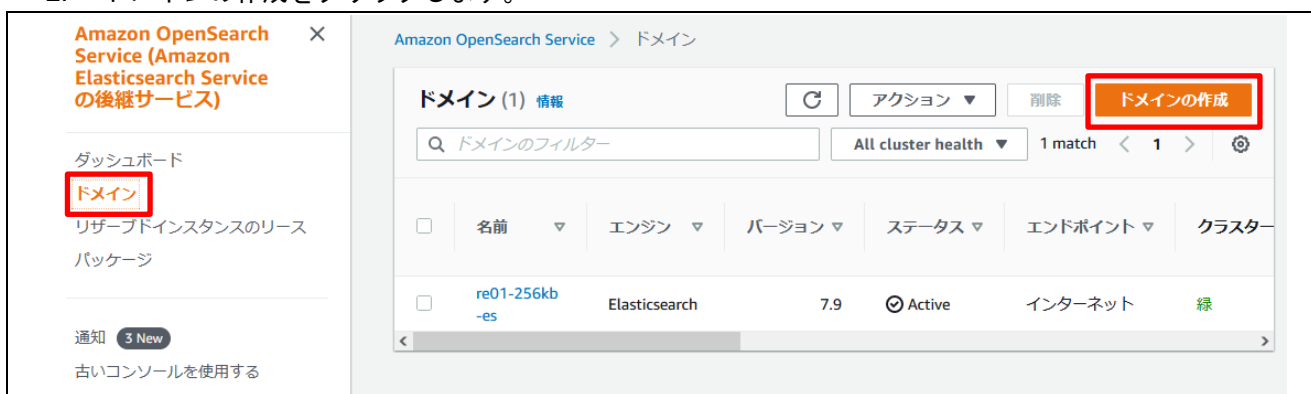


図 6.2 新しいドメインの作成

## RL78/G23 RL78/G23-128p Fast Prototyping Board で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化する方法

- ドメイン名を入力し、開発およびテストにチェックを入れて、バージョンを 7.9 に設定し下へスクロールします。

The screenshot shows the 'ドメインの作成' (Create Domain) page in the Amazon OpenSearch Service console. The left sidebar contains navigation options: 'ダッシュボード', 'ドメイン', 'リザーブドインスタンスのリース', 'パッケージ', '通知 4 New', and '古いコンソールを使用する'. The main content area is titled 'ドメインの作成' and includes the following sections:

- 名前**: A text input field containing 'rl78g23-sensor'. Below it, a note states: '名前は小文字で始まり、3~28 文字で構成される必要があります。有効な文字は、a~z (小文字のみ)、0~9、- (ハイフン) です。'
- カスタムエンドポイント**: A section with a note: '各 Amazon OpenSearch Service ドメインには自動生成されたエンドポイントがありますが、AWS Certificate Manager (ACM) を使用してカスタムエンドポイントを追加することもできます。詳細はこちら' and a checkbox labeled 'カスタムエンドポイントを有効化' which is currently unchecked.
- デプロイタイプ**: A section with a note: 'デプロイタイプはユースケースの一般的な設定を指定します。これらの設定は、ドメインの作成後に、いつでも変更できます。' and three radio button options:
  - 本番稼働用: 安定したパフォーマンスを確保する場合、複数のアベイラビリティゾーンと専用マスターノード。
  - 開発およびテスト: OpenSearch エンドポイントが 1 つのみ必要な場合は、1 つのアベイラビリティゾーン。
  - カスタム: すべての利用可能なオプションから設定を選択します。
- バージョン**: A dropdown menu set to '7.9'. Below it, a note states: '特定の機能には、特定の OpenSearch/Elasticsearch のバージョンが必要です。最新バージョンの選択をお勧めします。詳細はこちら'

図 6.3 ドメイン名とデプロイタイプ設定



## 4. インスタンスタイプを t2.small.search、ノードの数を 1 に変更し下へスクロールします。



図 6.4 データノード設定

## 5. パブリックアクセスをチェックし下にスクロールします。



図 6.5 ネットワーク設定

## RL78/G23 RL78/G23-128p Fast Prototyping Board で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化する方法

- 「ドメインレベルのアクセスポリシーの設定」をチェックし、タイプは IPv4 アドレスを選択します。プリンシパルに RL78/G23-128p Fast Prototyping Board のグローバル IP アドレスを入力し、アクションを許可に設定します。  
グローバル IP アドレスは自身の PC を RL78/G23-128p Fast Prototyping Board と同じネットワークに接続し、インターネットで「グローバル IP アドレス 確認方法」でサーチして確認します。

The screenshot shows the Amazon OpenSearch Service console interface. On the left is a navigation sidebar with options like 'Dashboard', 'Domain', and 'Release instances'. The main content area is titled 'Access Policy' and includes a sub-section 'Domain Access Policy' with three radio button options. The third option, 'Domain Level Access Policy', is selected and highlighted with a red box. Below this, there are tabs for 'Visual Editor' and 'JSON', and a 'Policy Import' button. A section titled 'Elements' explains how to define policies using principals and actions. At the bottom, a table shows a single policy element being added: 'IPv4 Address' as the type, a redacted IP address as the principal, and 'Allow' as the action. A 'Delete' button is also visible next to the element.

図 6.6 アクセスポリシーの設定

7. 暗号化を赤枠の通りに設定し、作成ボタンを押します。

The screenshot shows the Amazon OpenSearch Service console interface. On the left is a navigation sidebar with options like 'ダッシュボード', 'ドメイン', 'リザーブドインスタンスのリース', 'パッケージ', '通知 4 New', and '古いコンソールを使用する'. The main content area is titled 'Amazon OpenSearch Service (Amazon Elasticsearch Service の後継サービス)'. It shows an IPv4 address '217.178.137.172' and a '許可' (Allow) dropdown. Below this is a '新しい要素の追加' (Add new element) button. The '暗号化' (Encryption) section is highlighted with a red box and contains a warning message: '選択されたインスタンスタイプ (t2.small.search) では、保管時の暗号化はサポートされていません。' (The selected instance type (t2.small.search) does not support encryption at rest). Underneath, there are three options: 'ドメインへのすべてのトラフィックに HTTPS を要求する' (checked), 'ノード間の暗号化' (unchecked), and '保管時のデータの暗号化の有効化' (unchecked). Below the encryption section is the 'タグ - オプション' (Tags - Options) section, which is currently empty. At the bottom right, there are 'キャンセル' (Cancel) and '作成' (Create) buttons, with the '作成' button highlighted in red.

図 6.7 暗号化の設定

# RL78/G23 RL78/G23-128p Fast Prototyping Board で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化する方法

8. ドメインのステータスがアクティブになるまで待ちます。

注：通常は 10 分程度ですが状況により時間がかかることがあります。



図 6.8 ドメインのステータスがアクティブになるまで待つ

9. ドメインのステータスがアクティブになったら、Kibana の URL にアクセスします。



図 6.9 ドメインのステータスがアクティブ

## 7. Kibana の準備

1. 赤枠をクリックします。

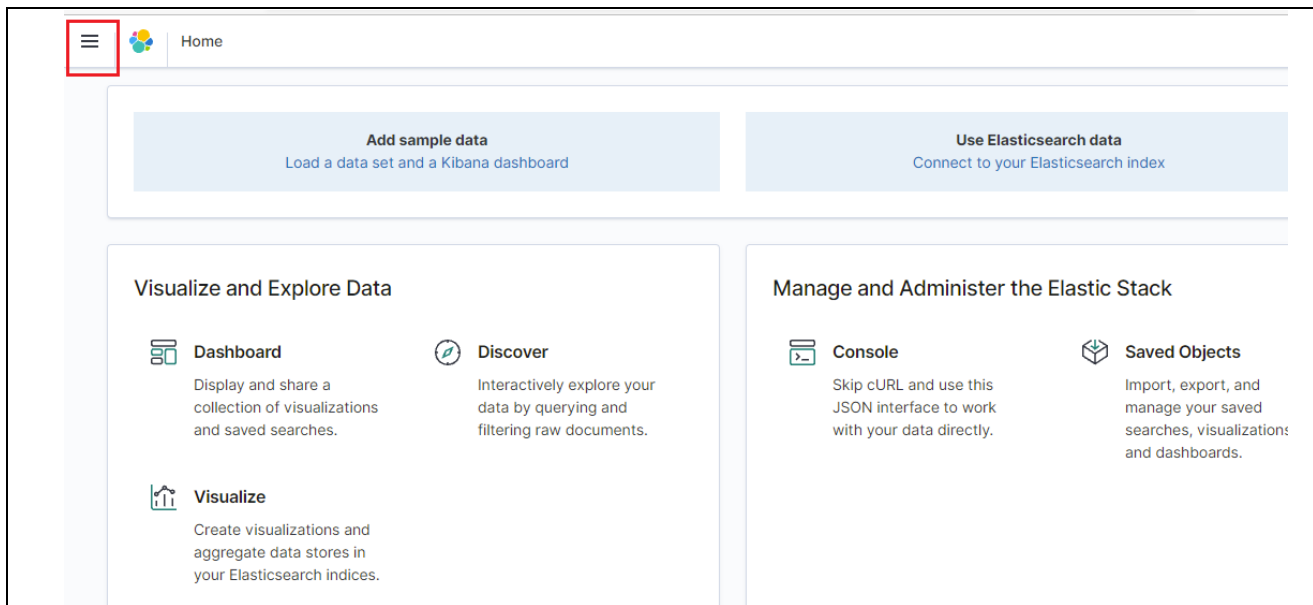


図 7.1 Kibana の準備

2. Dev Tools をクリックします。

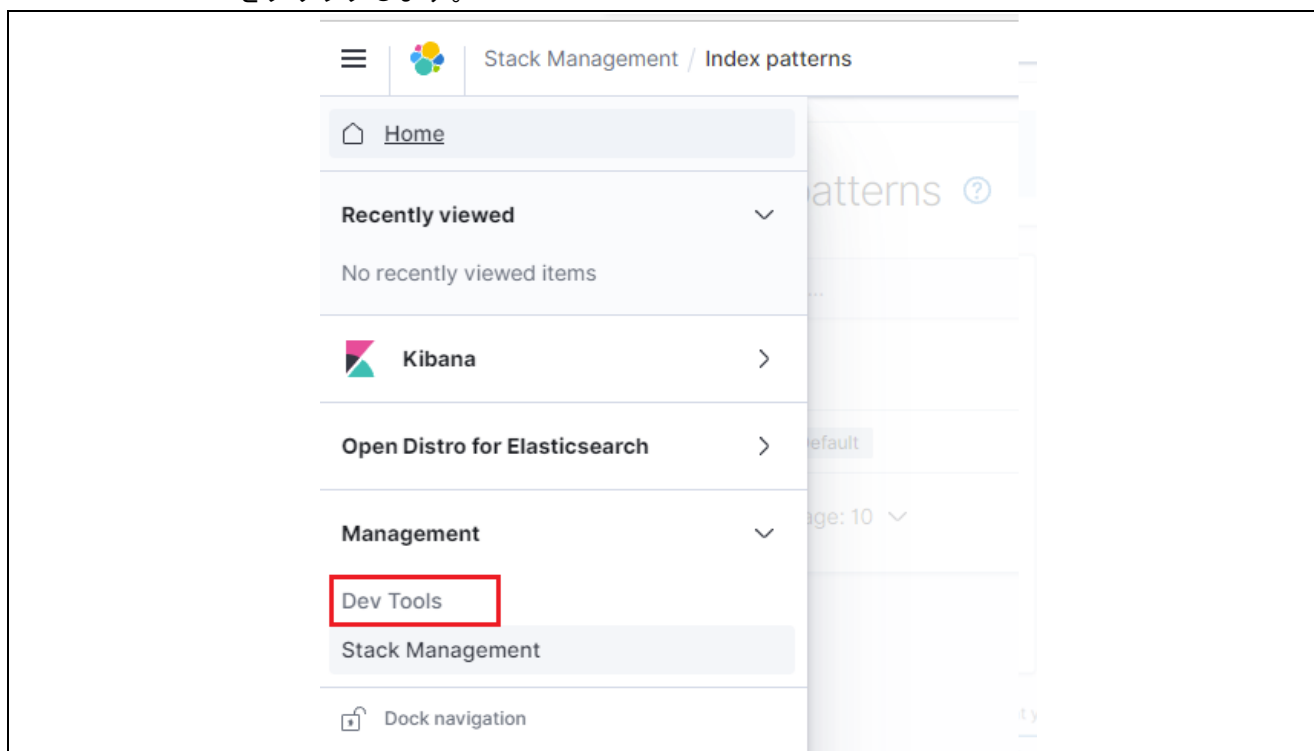


図 7.2 Dev Tools

3. 左側のコンソールに以下のコードを入力します。

```
PUT /sensor?include_type_name=true
{
  "mappings": {
    "sensor": {
      "properties": {
        "timestamp": {
          "type": "long",
          "copy_to": "datetime"
        },
        "datetime": {
          "type": "date",
          "store": true
        },
        "temperature": {
          "type": "long"
        },
        "humidity": {
          "type": "long"
        }
      }
    }
  }
}
```

図 7.3 コンソールにコードを入力

4. コンソール右上の click to send request を 1 回クリックします。

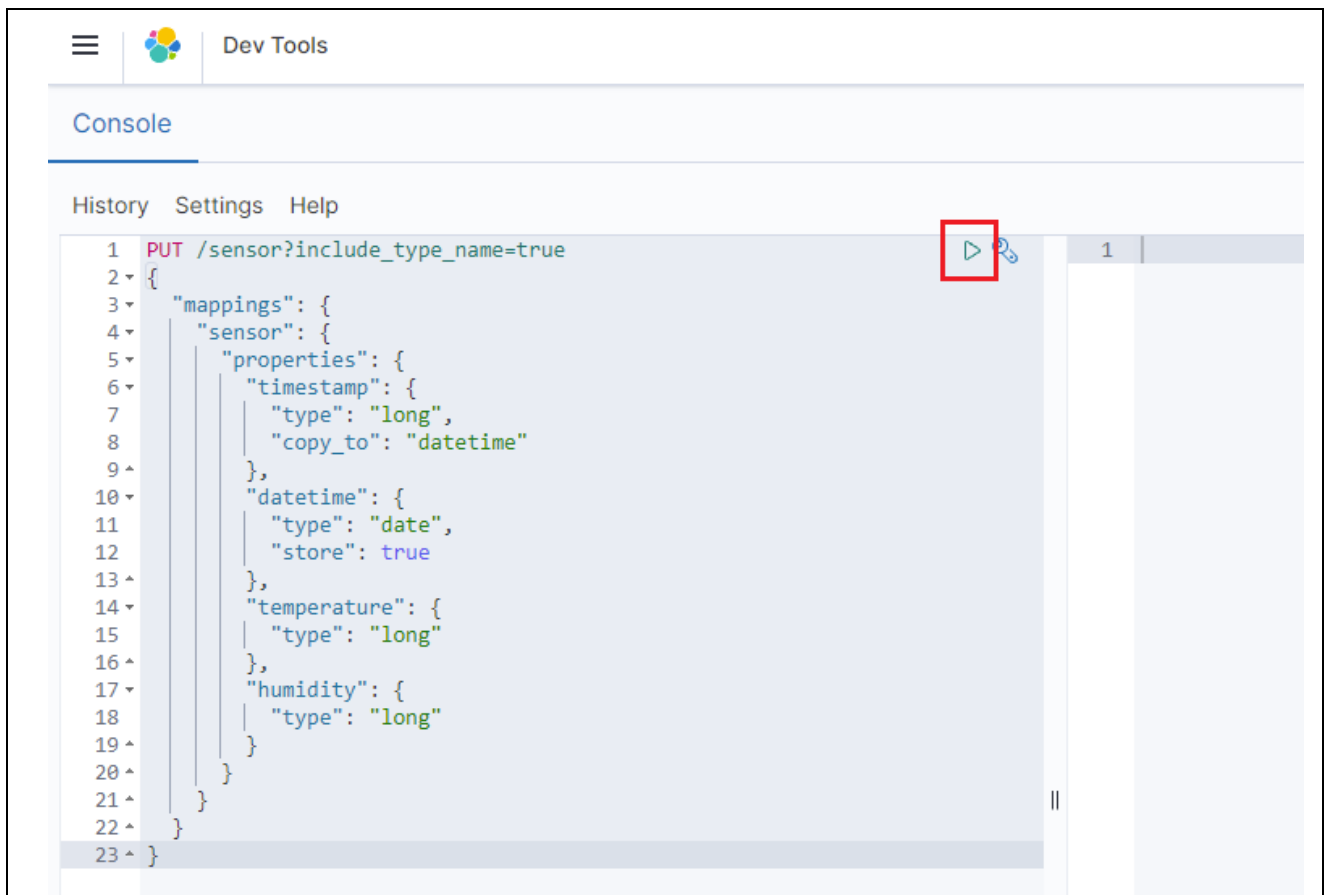


図 7.4 click to send request



5. 以下のレスポンスが返却されることを確認します。

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "sensor"
}
```

Console

History Settings Help

200 - success 773 ms

```
1 PUT /sensor?include_type_name=true
2 {
3   "mappings": {
4     "sensor": {
5       "properties": {
6         "timestamp": {
7           "type": "long",
8           "copy_to": "datetime"
9         },
10        "datetime": {
11          "type": "date",
12          "store": true
13        },
14        "temperature": {
15          "type": "long"
16        },
17        "humidity": {
18          "type": "long"
19        }
20      }
21    }
22  }
23 }
```

```
1 {
2   "acknowledged" : true,
3   "shards_acknowledged" : true,
4   "index" : "sensor"
5 }
```

図 7.5 レスポンスの確認

## 8. IoT Rule の準備

AWS の IoT からルールを作成します。

1. IoT Core コントロールパネルに移動し、ACT -> ルール -> 作成 をクリックします。

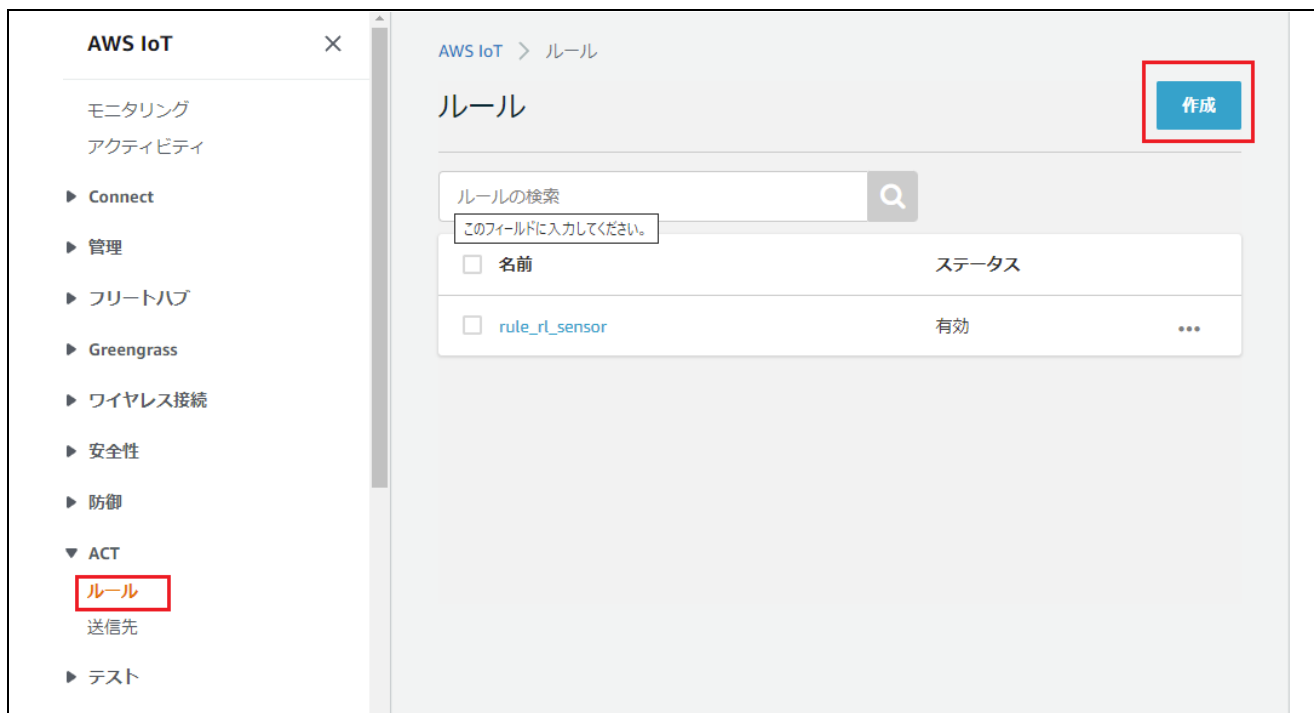


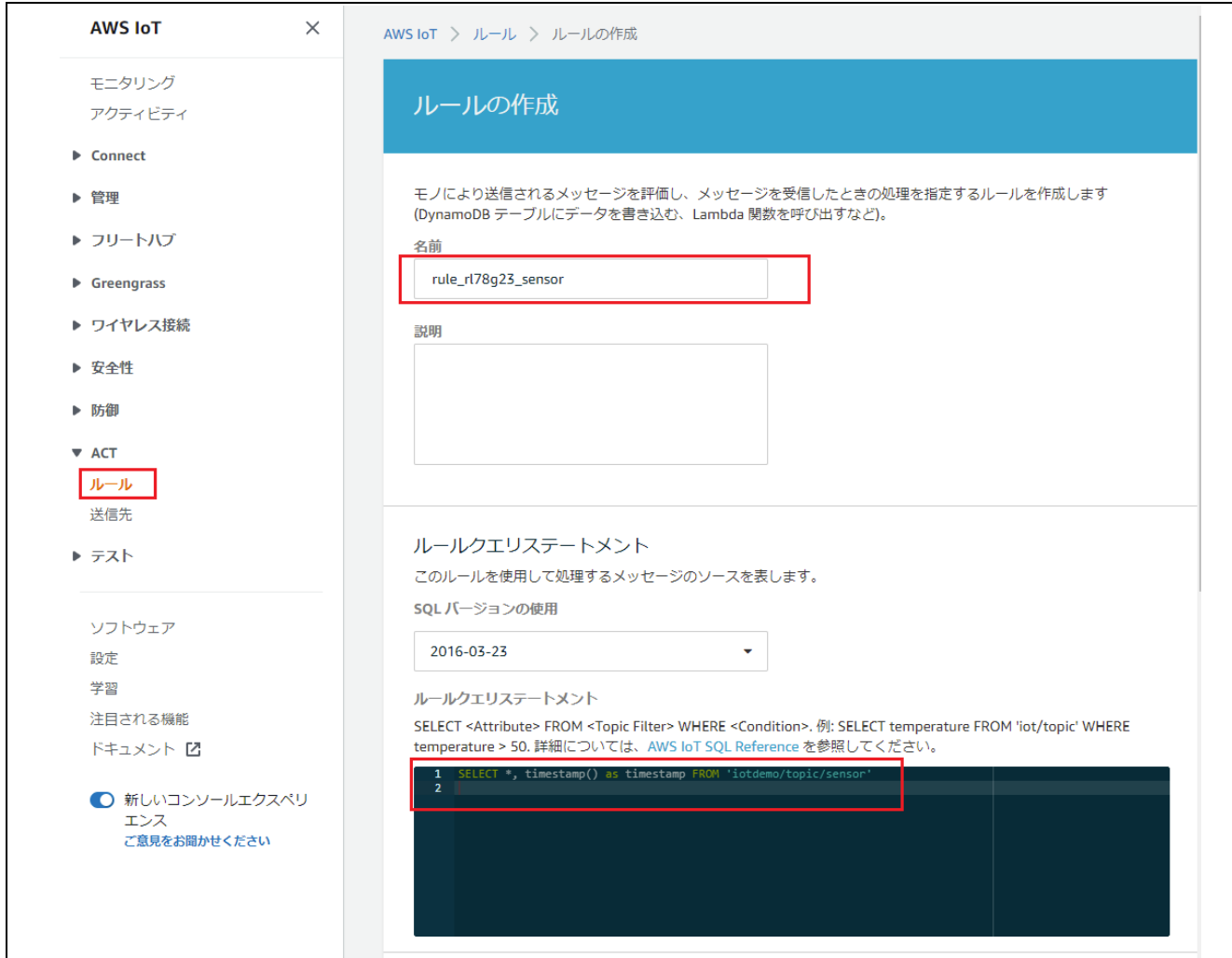
図 8.1 ルールの作成

## RL78/G23 RL78/G23-128p Fast Prototyping Board で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化する方法

2. ルールの名前を入力してから、ルールクエリステートメントに以下のコードを入力します。

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

注：図のようにルールクエリステートメントにコード入力後、必ず改行を行ってください。



The screenshot shows the AWS IoT console interface for creating a rule. On the left is a navigation menu with 'ルール' (Rules) highlighted. The main area is titled 'ルールの作成' (Create Rule). It includes a description of rules, a '名前' (Name) field containing 'rule\_rl78g23\_sensor', an empty '説明' (Description) field, and a 'ルールクエリステートメント' (Rule Query Statement) section. This section has a dropdown for 'SQLバージョンの使用' (SQL Version) set to '2016-03-23' and a text area for the query. The query text is: 'SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. 例: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. 詳細については、AWS IoT SQL Reference を参照してください。' Below this, a code editor shows the SQL code: '1 SELECT \*, timestamp() as timestamp FROM 'iotdemo/topic/sensor'' and '2'. The code editor is highlighted with a red box.

図 8.2 コードの入力

3. アクションの追加をクリックします。

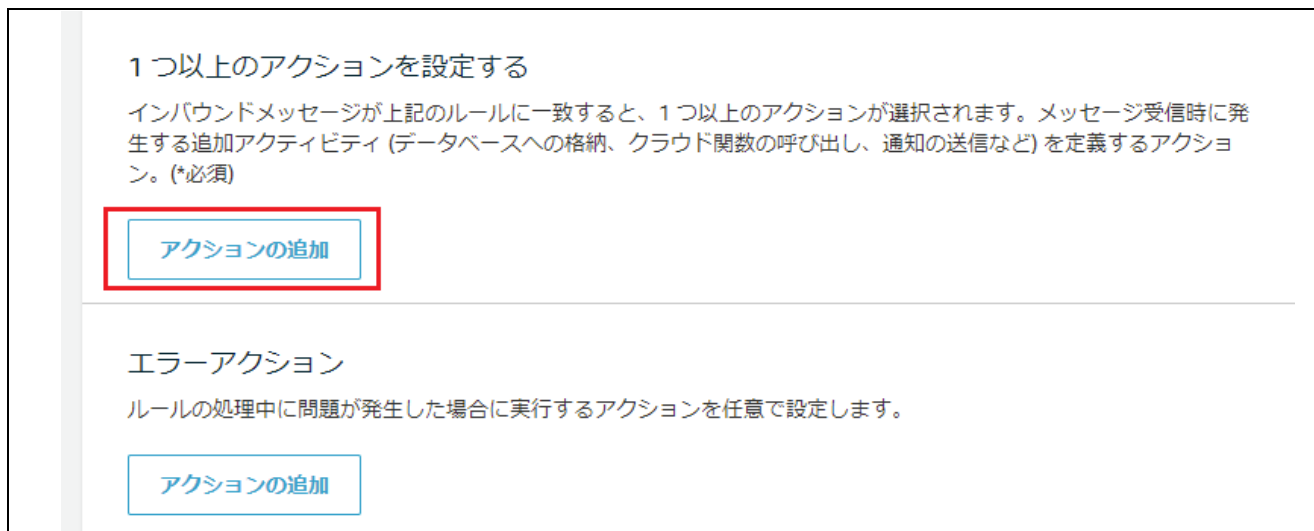


図 8.3 アクションの追加

4. 「Amazon OpenSearch Service にメッセージを送信する」を選択してアクションの設定をクリックする。



図 8.4 アクションの設定

## RL78/G23 RL78/G23-128p Fast Prototyping Board で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化する方法

5. ドメイン名に Amazon OpenSearch Service で作成したドメイン名（6章の手順3を参照）、ID に `${newuid()}`、索引に `sensor`、タイプに `sensor` を入力します。  
その後、ロールの作成をクリックします。

### アクションの設定

Amazon OpenSearch Service (Amazon Elasticsearch Service の後継) にメッセージを送信する  
AMAZON OPENSEARCH SERVICE (AMAZON ELASTICSEARCH SERVICE の後継)

このアクションにより、メッセージは Amazon OpenSearch クラスターに送信されます。

\*ドメイン名

rt78g23-sensor

\*エンドポイント

https://[REDACTED]

\*ID

\*索引

\*タイプ

このアクションを実行するための AWS IoT アクセス権限を付与するロールを選択または作成します。

選択されたロールがありません

図 8.5 ドメイン名、ID、索引、タイプの設定

6. ロールの名前を入力してロールの作成をクリックします。

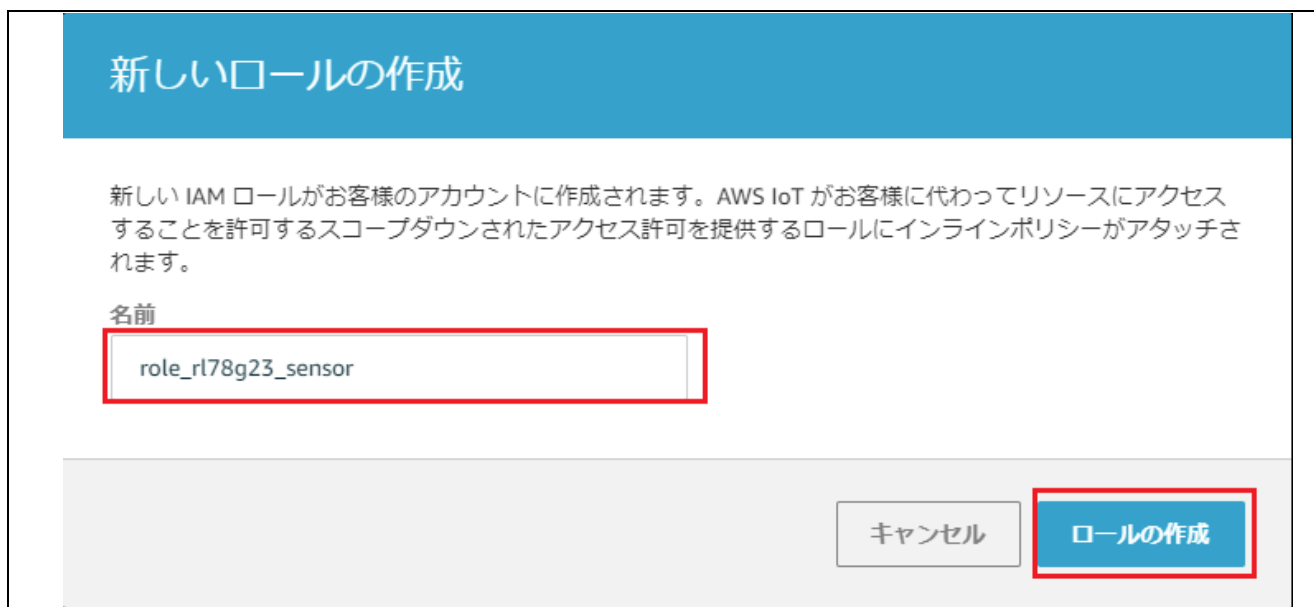


図 8.6 ロールの作成

## RL78/G23 RL78/G23-128p Fast Prototyping Board で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化する方法

7. 作成したロールが選択されていることを確認してから、アクションの追加をクリックします。



図 8.7 アクションの追加

8. アクションが追加されたことを確認し、ルールの作成をクリックします。

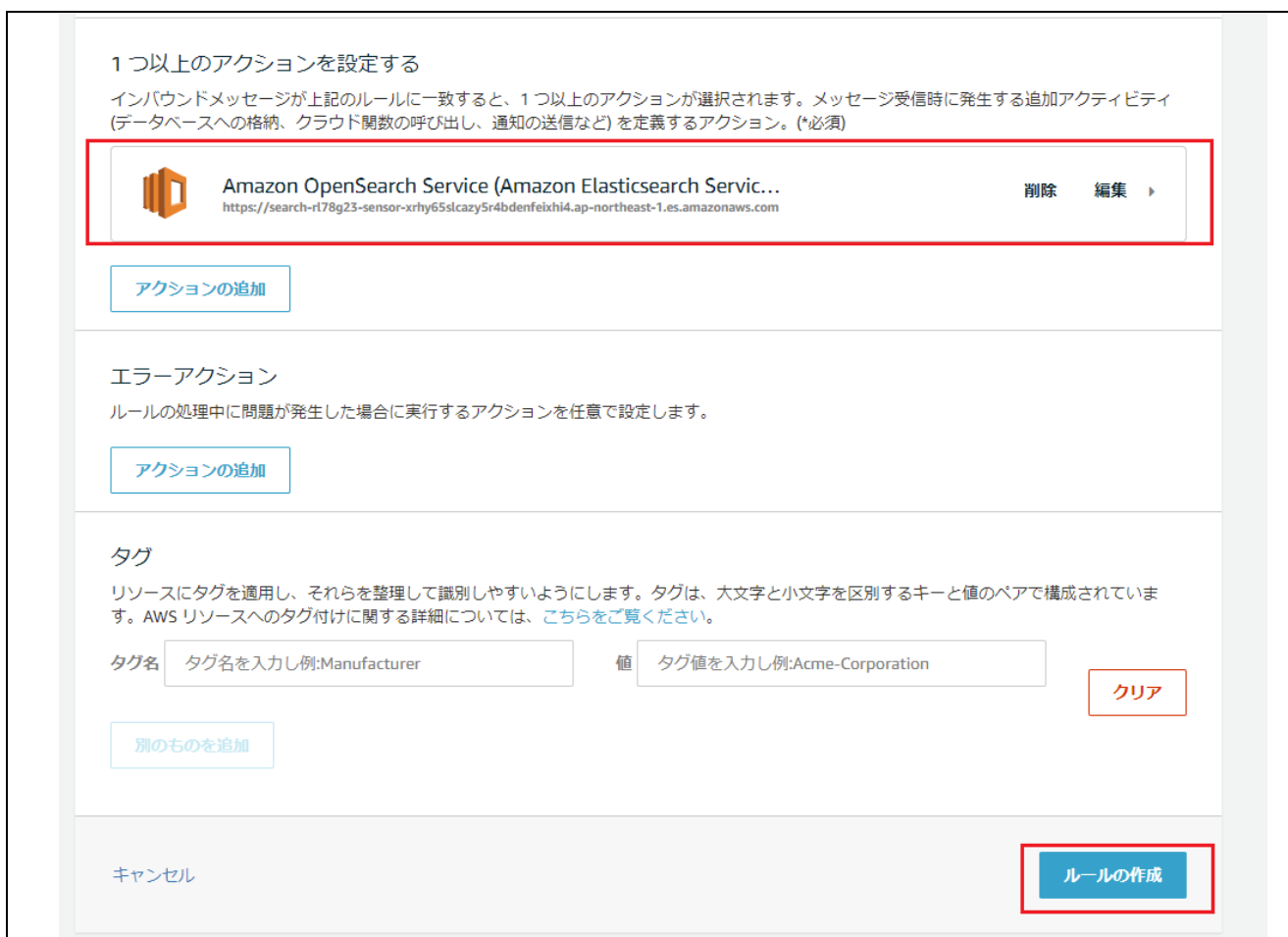


図 8.8 ルールの作成



9. ルールが追加されたことを確認し有効化します。

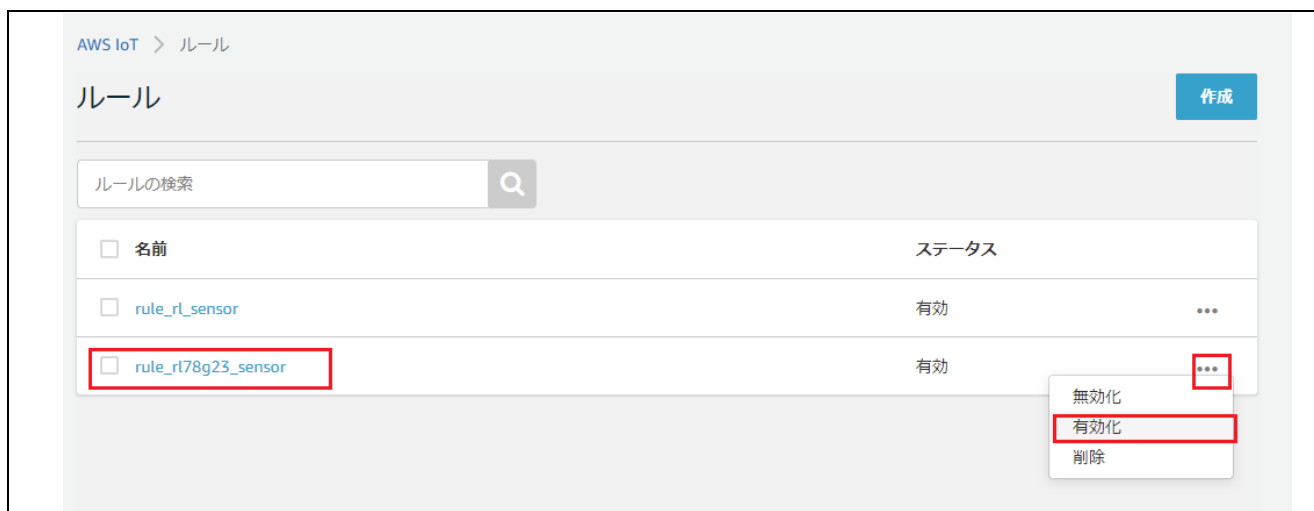


図 8.9 ルールの有効化

## 9. デモプログラムの実行

「5 デモプロジェクトの準備」で準備したプロジェクトのデモプログラムを実行します。

1. デバッグを押して、RL78/G23-128p Fast Prototyping Board に接続します。

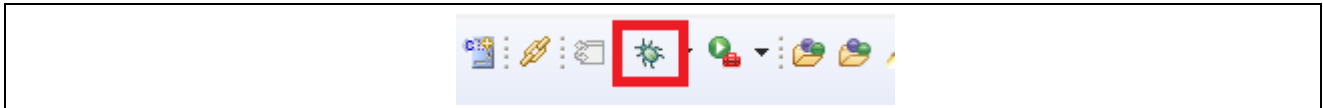


図 9.1 デバッグ

2. スタートを押すと、main 関数で一度停止します。再度スタートを押してデモプログラムを実行します。

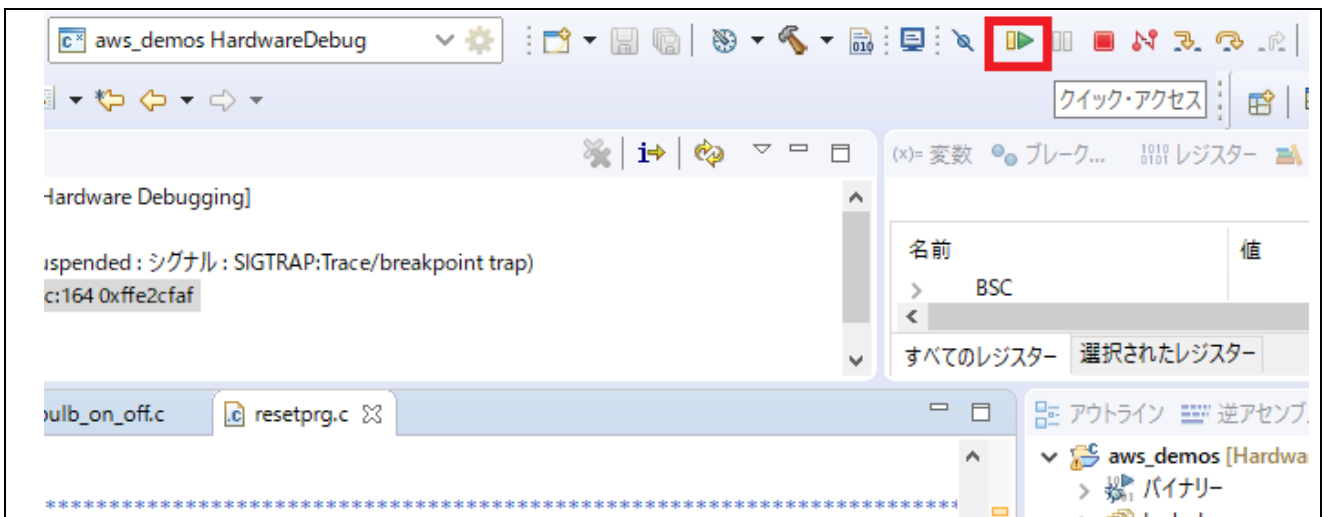


図 9.2 デモプログラムの実行

## 10. Kibana でセンサー情報を可視化

1. Kibana に移動し Stack Management をクリックします。

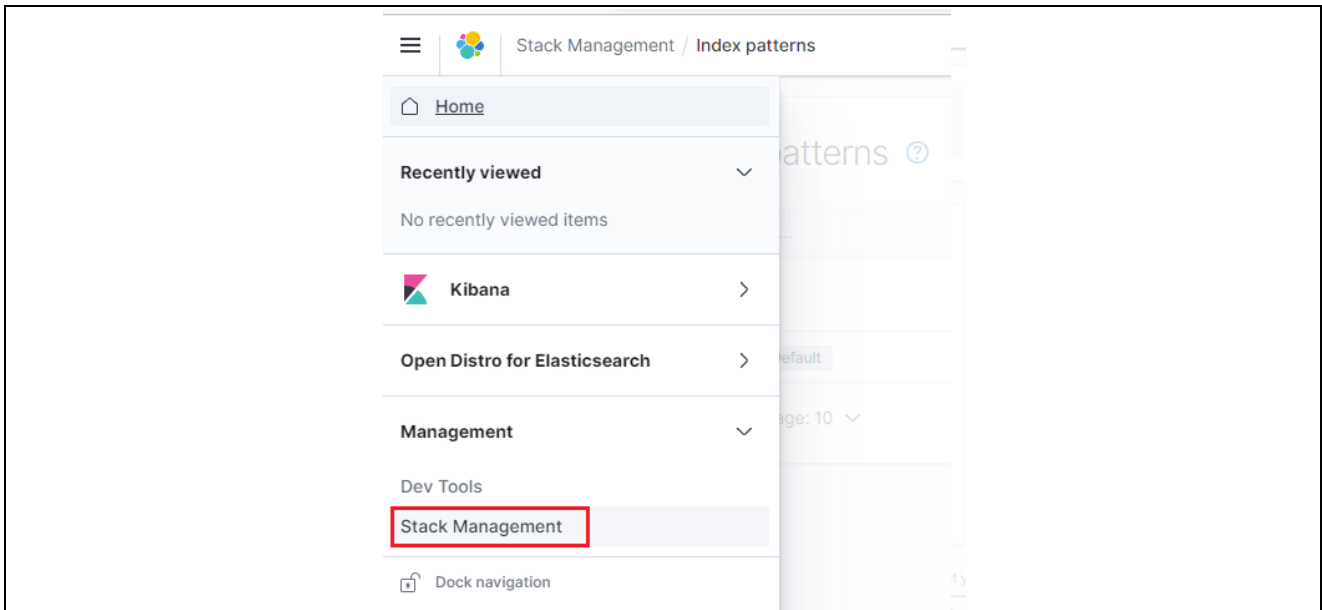


図 10.1 Kibana の設定

2. Index Patterns をクリックしてから、Create Index pattern をクリックします。

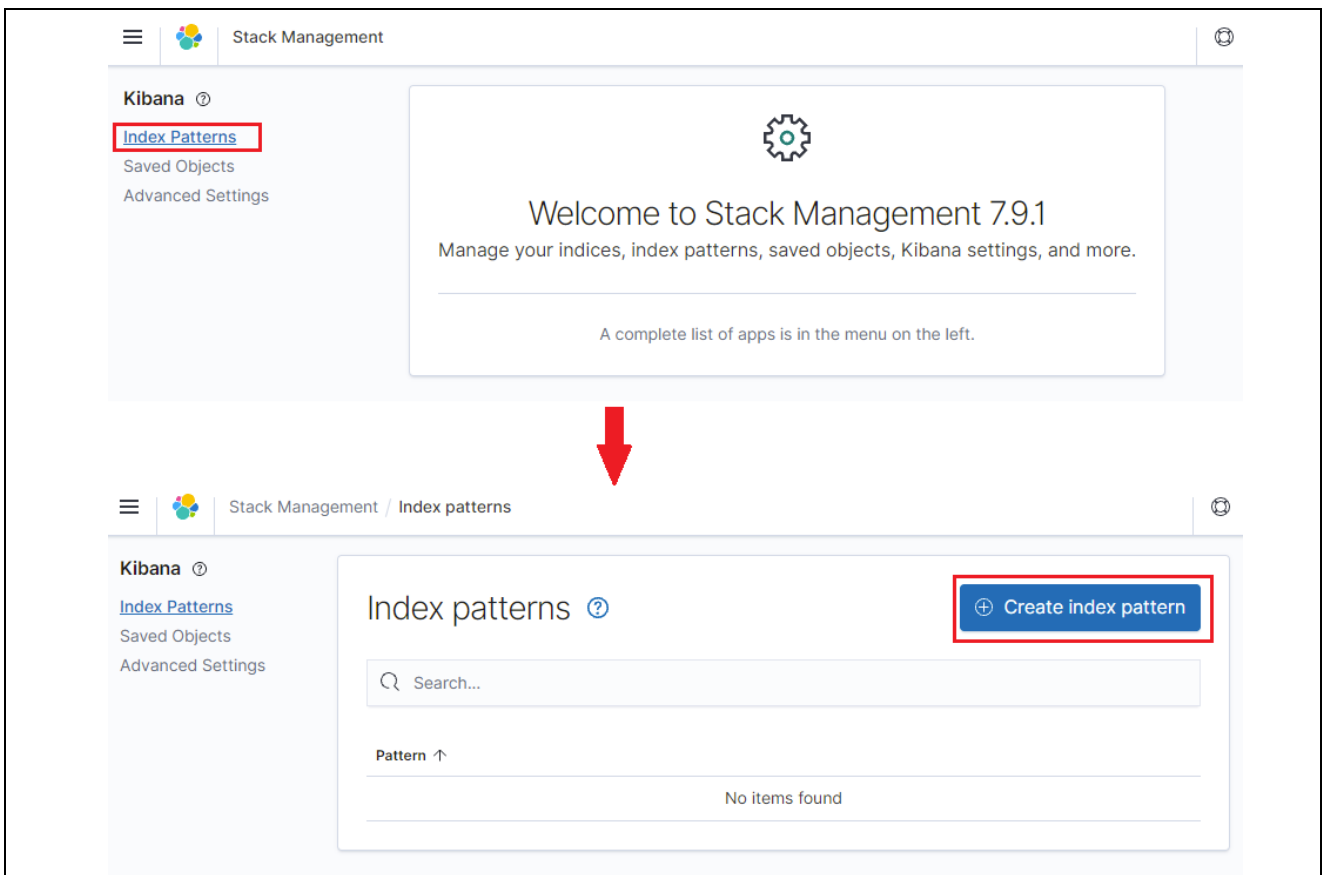


図 10.2 Define index pattern

3. Index pattern name に sensor を選択し Next step をクリックします。Step2 of 2 に切り替わったら datetime を選択し Create index pattern をクリックします。

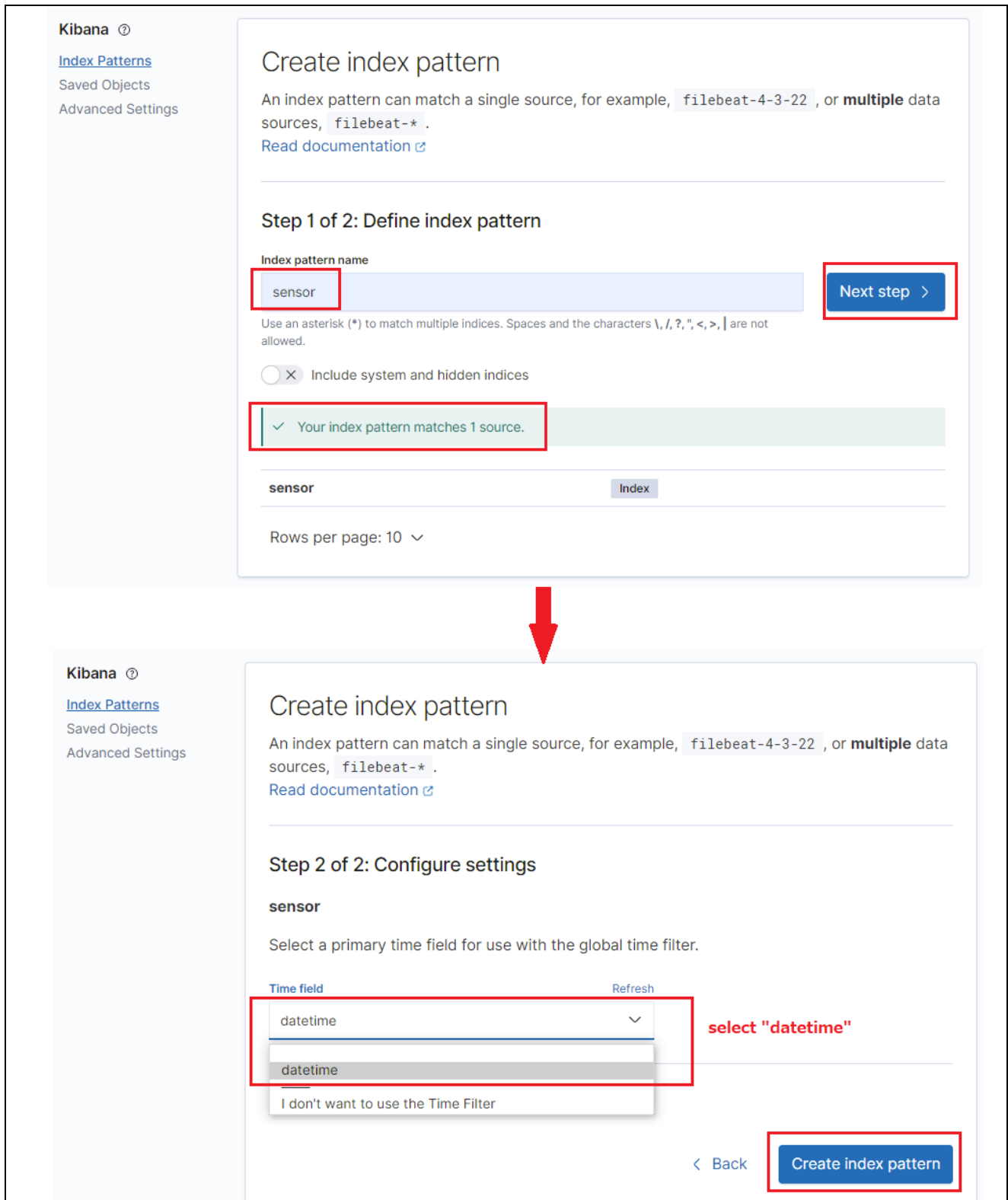


図 10.3 Index pattern 作成

4. 以下の画面が出ていれば Index pattern の作成に成功です。

The screenshot shows the Kibana Index Patterns page for the 'sensor' index pattern. The page title is 'sensor' and it shows the 'Time Filter field name' as 'datetime'. Below the title, there is a description: 'This page lists every field in the sensor index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch Mapping API'. There are three tabs: 'Fields (9)', 'Scripted fields (0)', and 'Source filters (0)'. The 'Fields (9)' tab is active, showing a table of fields. The table has columns for Name, Type, Format, Searchable, Aggregatable, and Excluded. The fields listed are: \_id (string), \_index (string), \_score (number), \_source (\_source), \_type (string), datetime (date), humidity (number), temperature (number), and timestamp (number). At the bottom of the table, there is a 'Rows per page: 10' dropdown and a pagination control showing page 1 of 1.

Name	Type	Format	Searchable	Aggregatable	Excluded
_id	string		●	●	
_index	string		●	●	
_score	number				
_source	_source				
_type	string		●	●	
datetime	date		●	●	
humidity	number		●	●	
temperature	number		●	●	
timestamp	number		●	●	

図 10.4 Index pattern 確認

5. Visualize を選択し Create a visualization をクリックします。

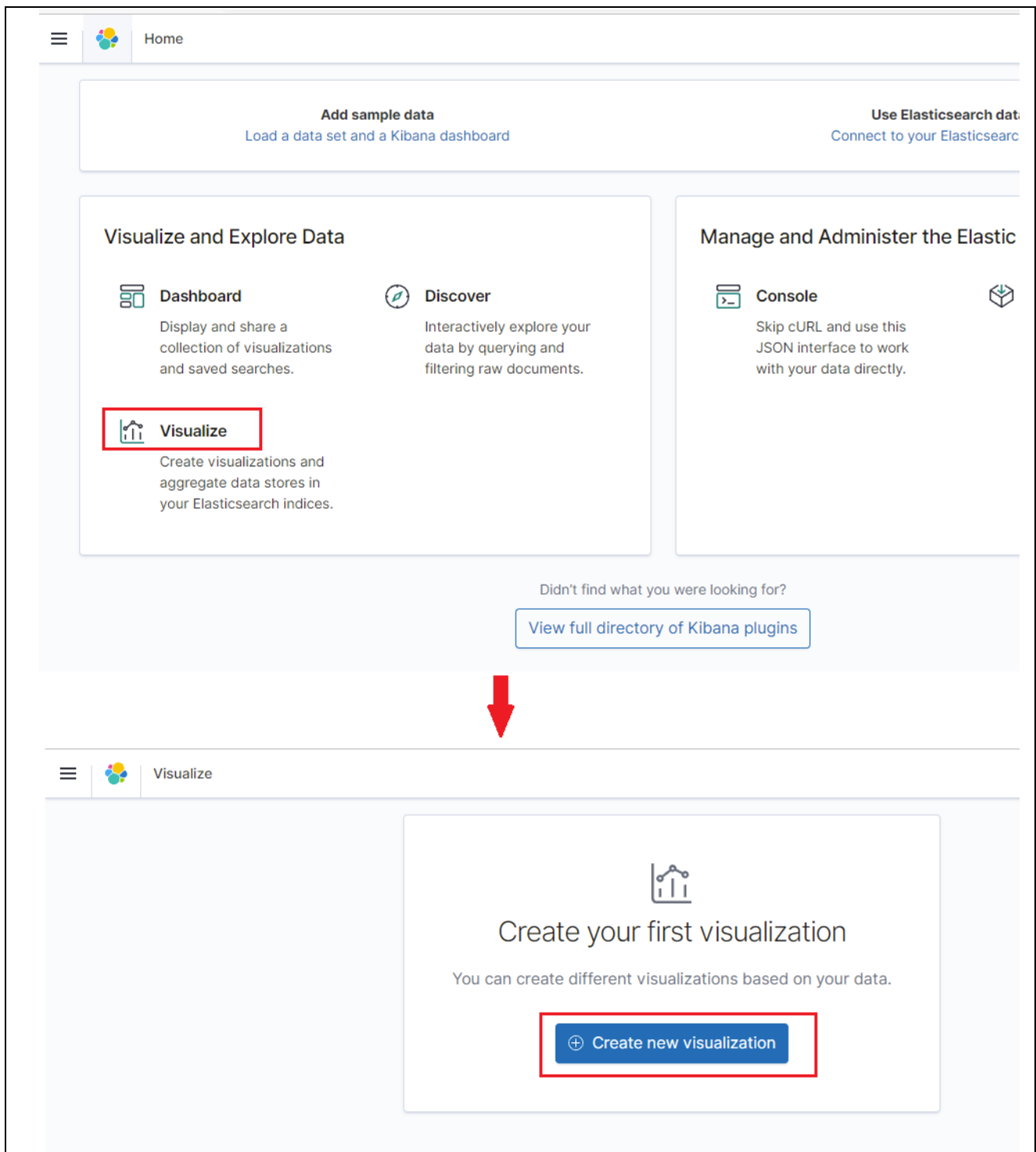


図 10.5 Create a visualization

6. Line をクリックします。

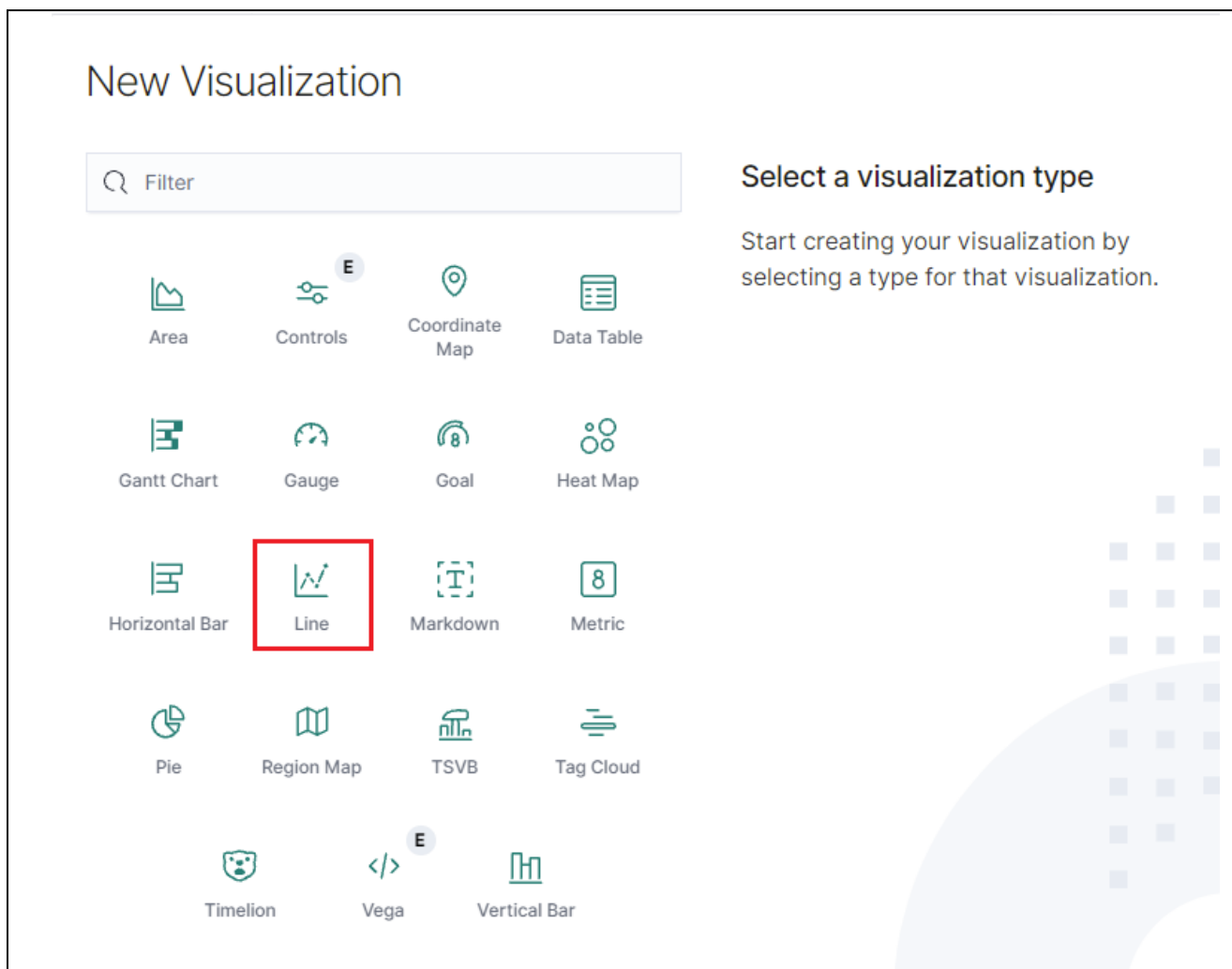


図 10.6 Line

7. sensor をクリックします。

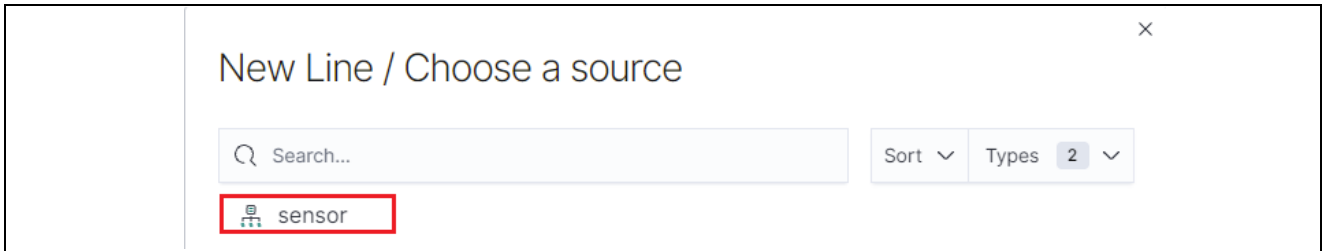


図 10.7 New Line / Choose a source

8. 右上のカレンダーマークから、Refresh every を 5 seconds に設定して Start をクリックします。

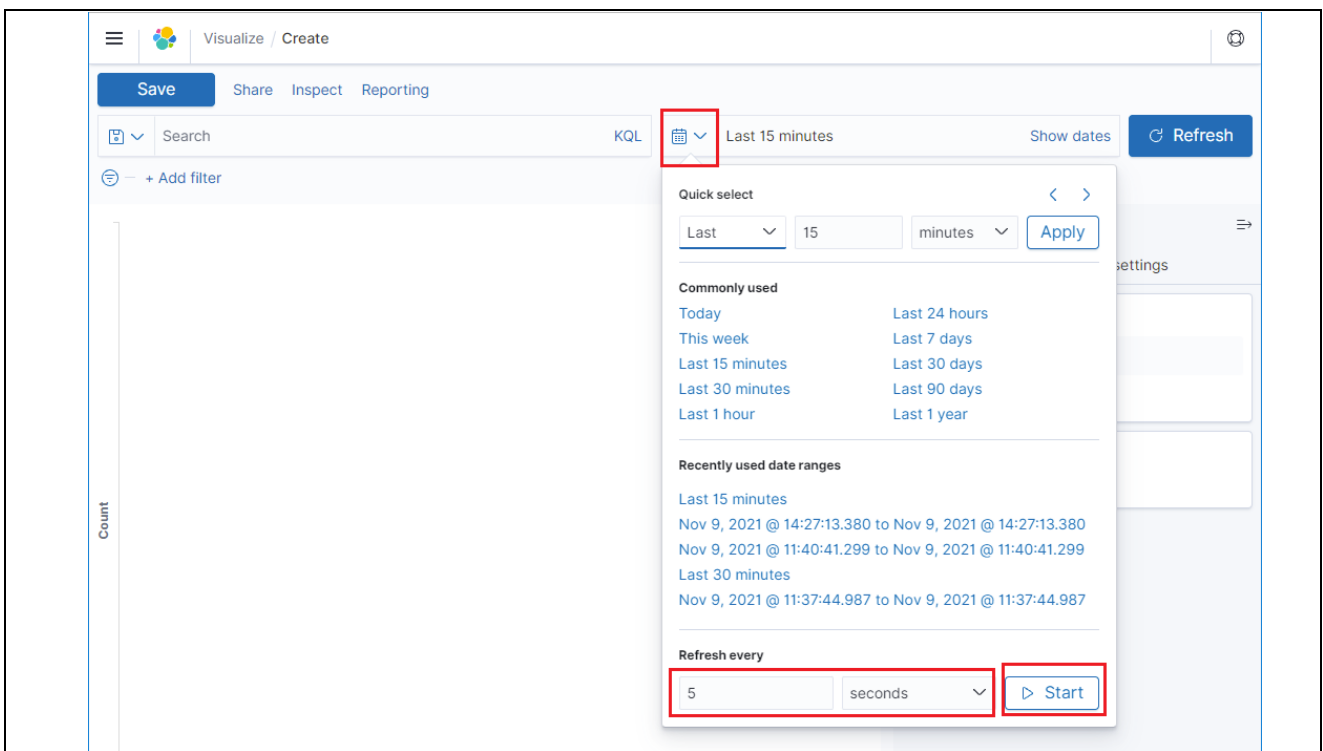


図 10.8 Refresh every の設定



9. Metrics の Y 軸と X 軸を以下の通りに設定します。

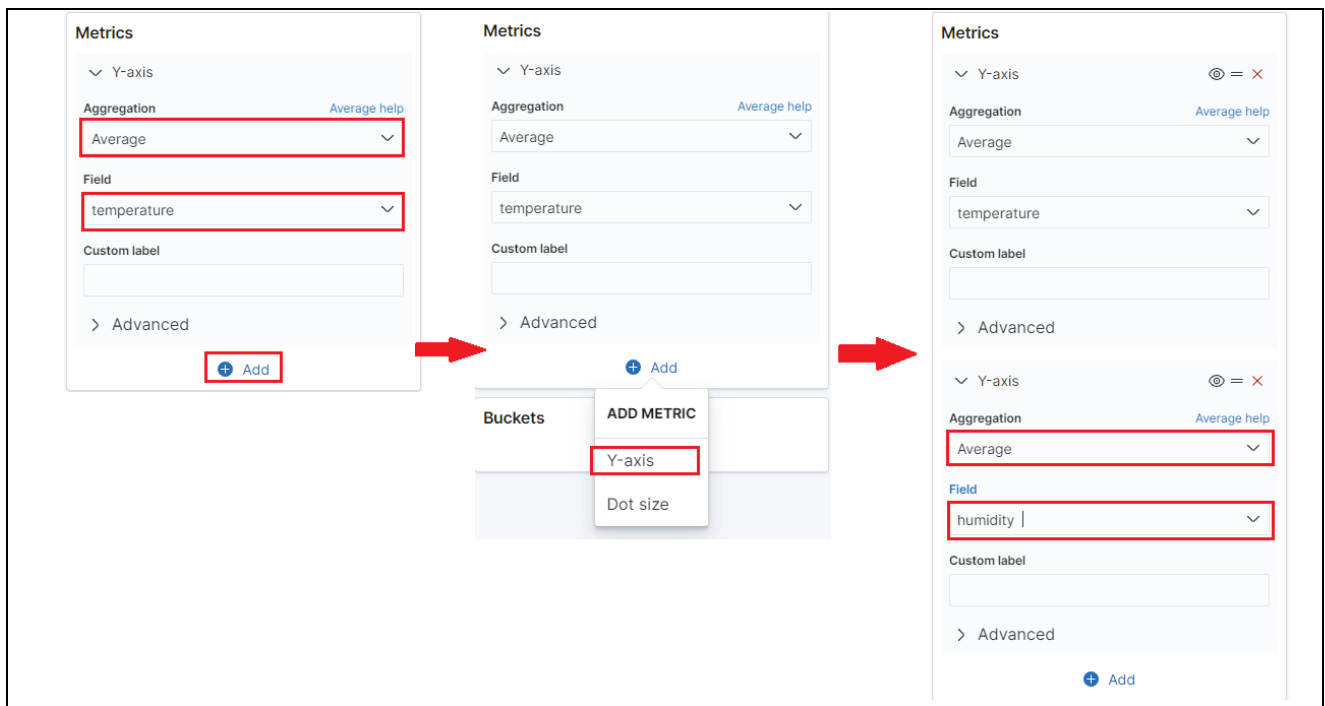


図 10.9 Metrics(Y-axis) の設定

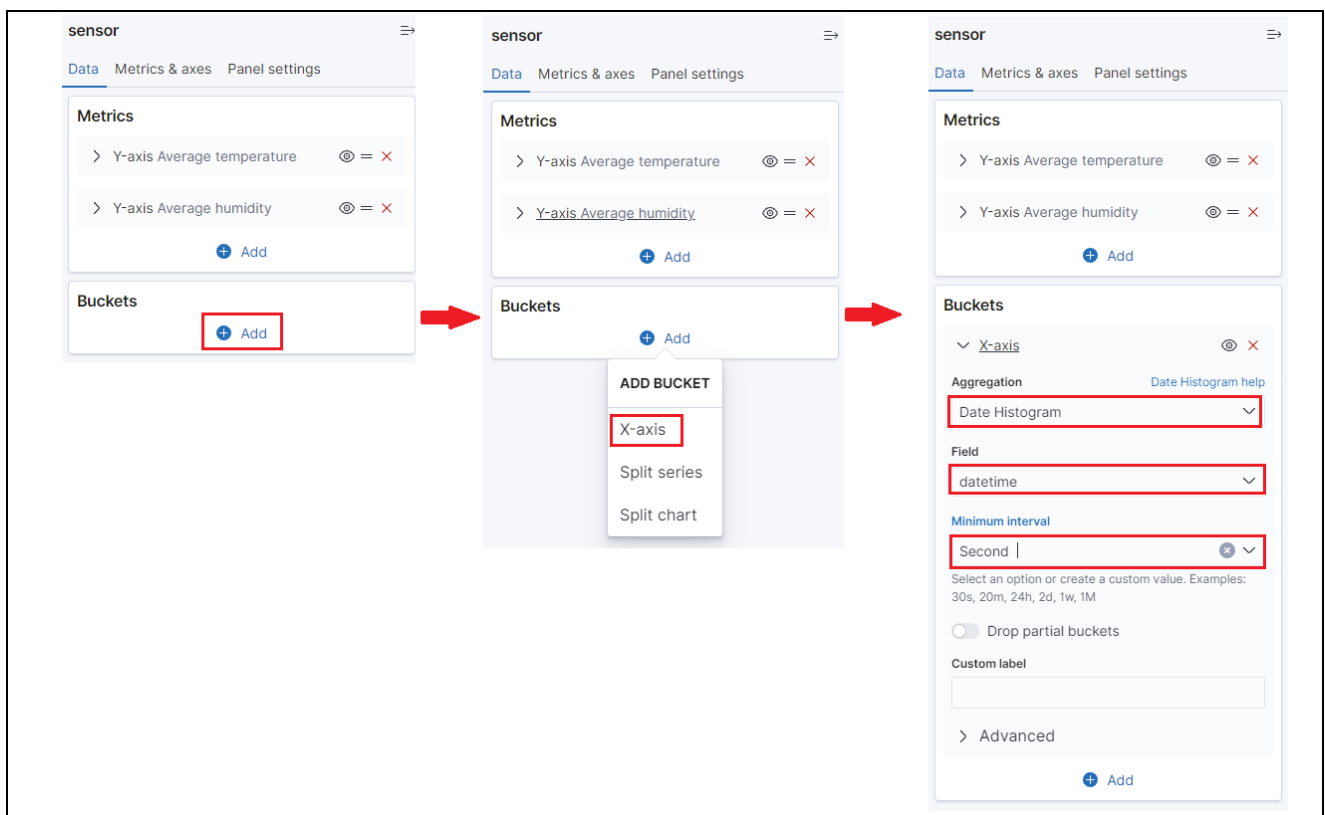


図 10.10 Metrics(X-axis) の設定

10. Update をクリックします。

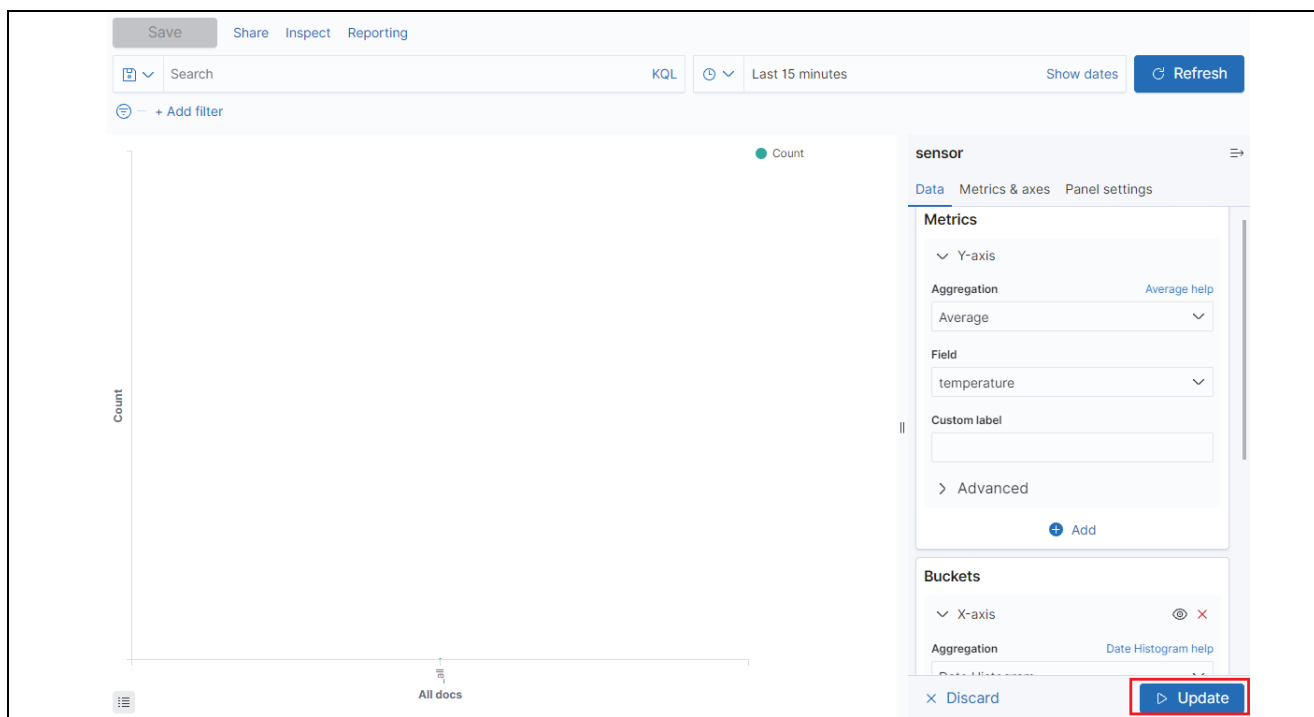


図 10.11 Update

11. 温度や湿度が変化することでグラフが変化することを確認します。以下に、温湿度のセンサー情報を可視化したときの図を示します。

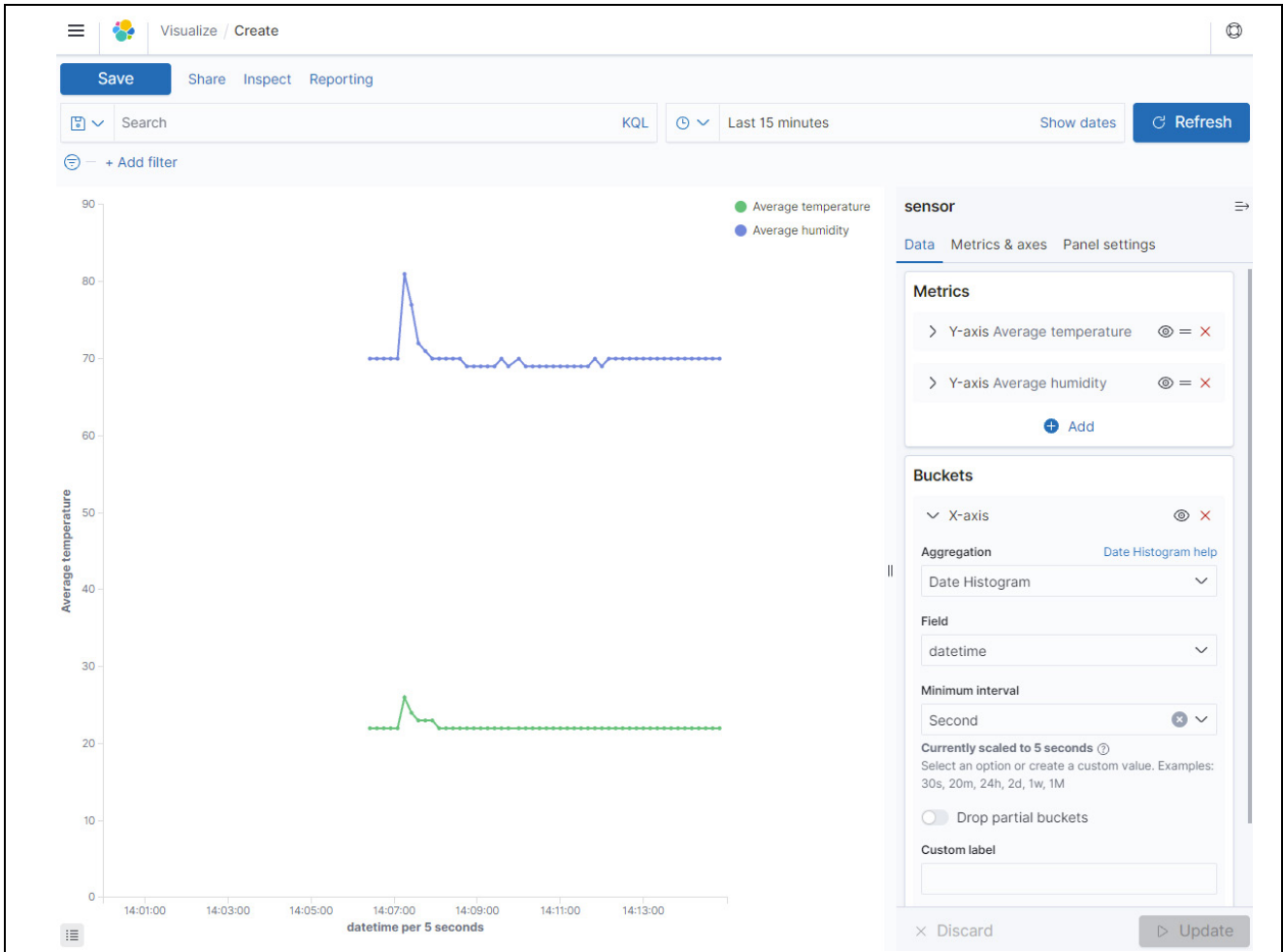


図 10.12 温湿度センサー情報の可視化

## 11. デモプログラム実行後の注意事項について

Amazon OpenSearch Service は起動していると料金が発生します。

デモプログラムを終了した際はドメイン削除を忘れずに行ってください。

The screenshot shows the Amazon OpenSearch Service console interface. At the top, the breadcrumb navigation is 'Amazon OpenSearch Service > ドメイン > rl78g23-sensor'. The main content area displays the details for the domain 'rl78g23-sensor'. A '削除' (Delete) button is highlighted with a red box. Below this, a confirmation dialog box titled 'ドメインを削除しますか?' (Delete domain?) is shown. The dialog contains the text: '次のドメインを削除してよろしいですか? データを復旧することはできなくなります。' (Are you sure you want to delete the following domain? Data cannot be recovered.) Below this, a table lists the domain 'rl78g23-sensor'. At the bottom of the dialog, a text input field contains 'rl78g23-sensor', which is also highlighted with a red box. To the right of the input field are 'キャンセル' (Cancel) and '削除' (Delete) buttons, with the '削除' button highlighted in orange.

一般的な情報			
名前 rl78g23-sensor	ドメインのステータス 🟢 アクティブ	バージョン Info Elasticsearch 7.9	Kibana URL [Redacted]
ドメイン ARN [Redacted]	クラスターのヘルス Info 🟡 黄色	サービスソフトウェアのバージョン Info R20210816-P3 (最新)	ドメインエンドポイント [Redacted]

図 11.1 Amazon OpenSearch Service のドメイン削除を忘れずに行う

## 12. 付録

### 12.1 ドライバのバージョンアップ方法について

デモプロジェクトのドライバをバージョンアップする方法を紹介します。

以下はスマート・コンフィグレータを使用し IICA ドライバをバージョンアップする手順です。

1. 新規にプロジェクトを作成します。この時、Use Smart Configurator にチェックを付けてください。終了ボタンを押すとプロジェクトが作成され、スマート・コンフィグレータの画面が開きます。

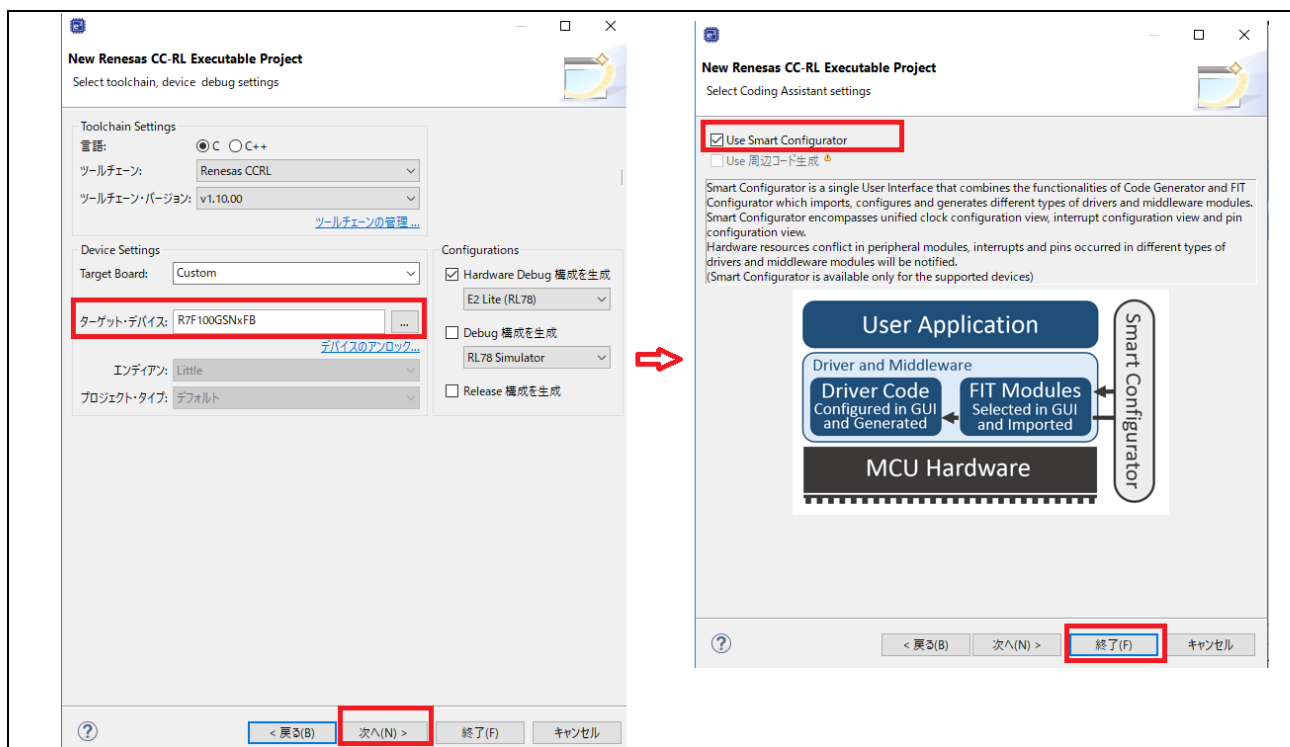


図 12.1 新規プロジェクト作成 (CC-RL)

# RL78/G23 RL78/G23-128p Fast Prototyping Board で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化する方法



図 12.2 スマート・コンフィグレータ起動

## 2. バージョンアップしたいソフトウェアコンポーネントを追加します。(例：IICA1)

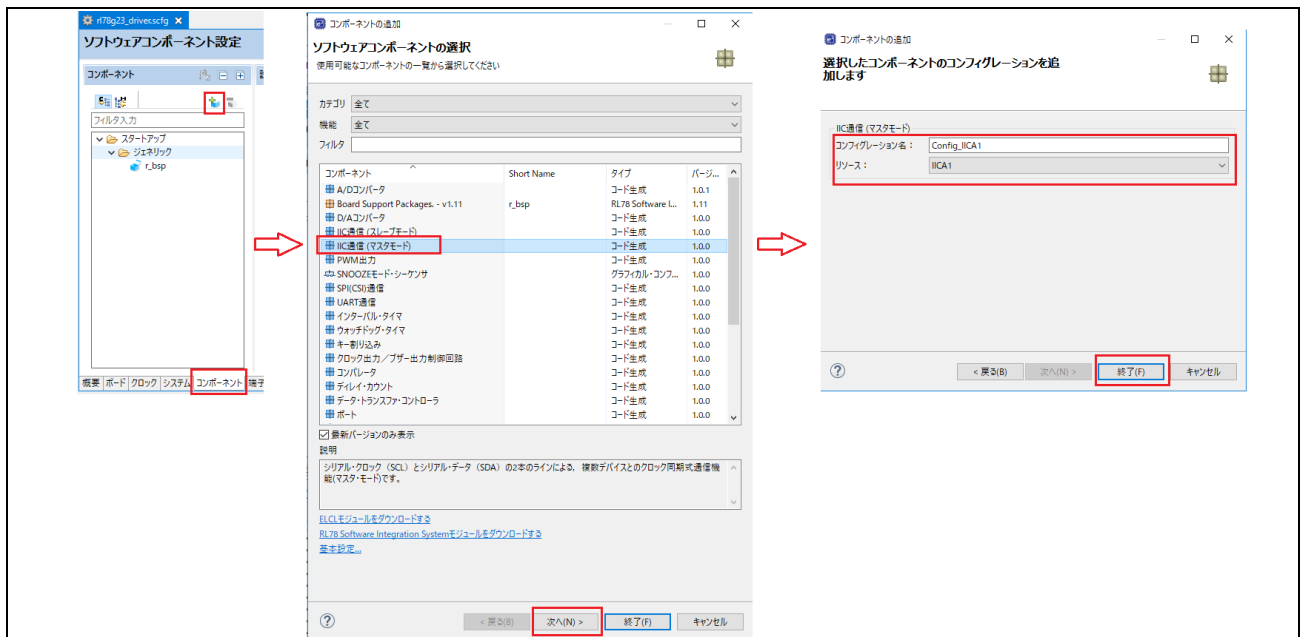


図 12.3 ソフトウェアコンポーネントの追加

3. 赤枠を設定しコードの生成を押すとドライバのコードが生成されます。

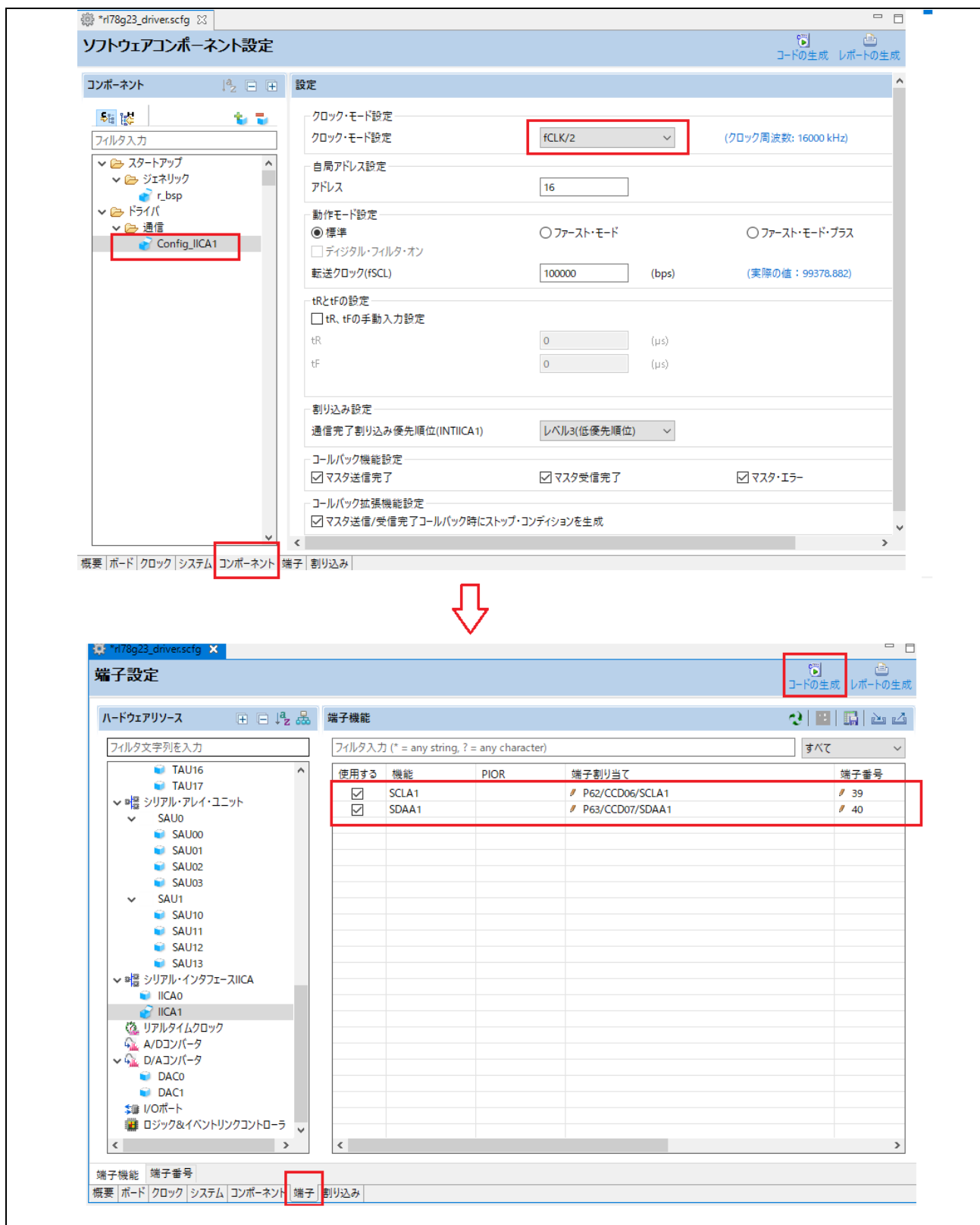


図 12.4 ドライバ設定

4. <Project\_Name>\src\smc\_gen 以下にドライバコードが生成されていることを確認します。

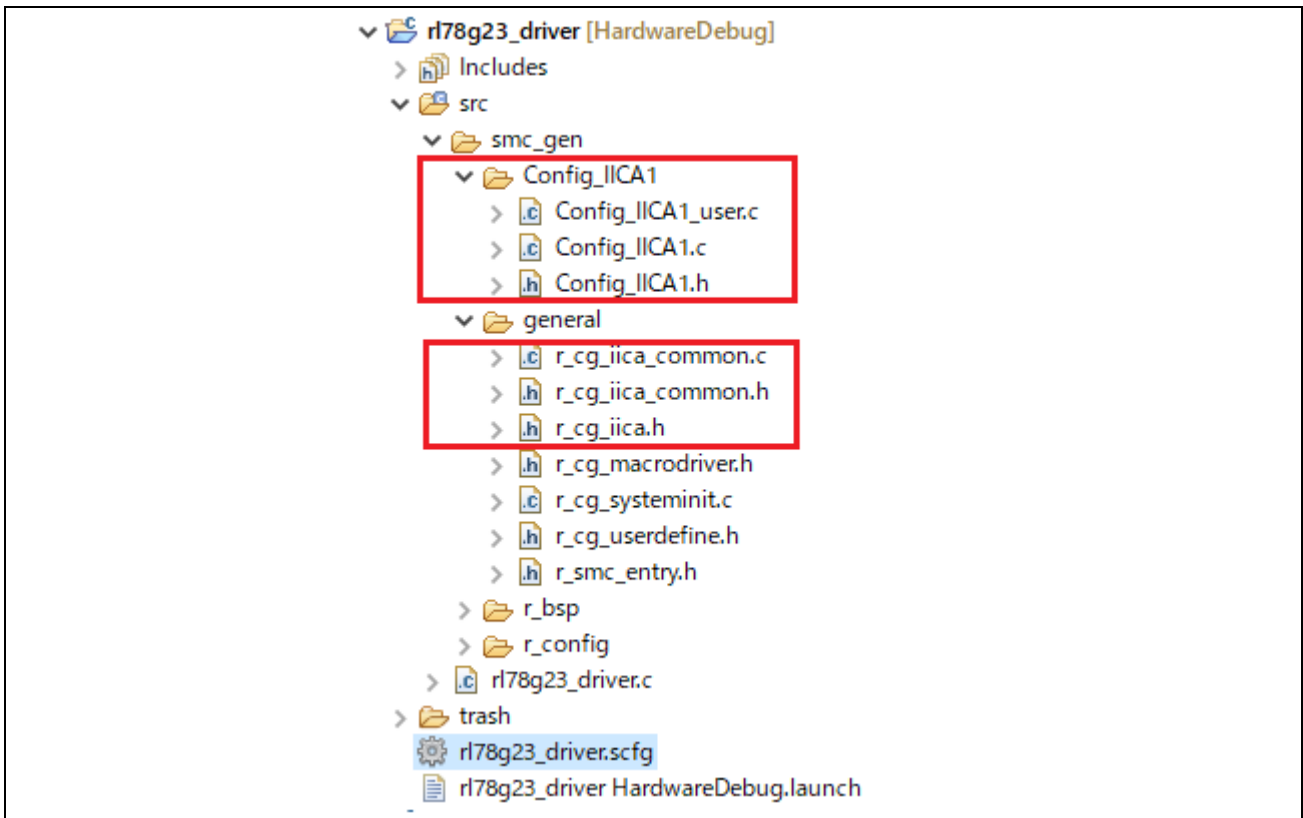


図 12.5 ドライバコード生成



5. 生成されたドライバを本プロジェクト (aws\_demos) にコピーします。

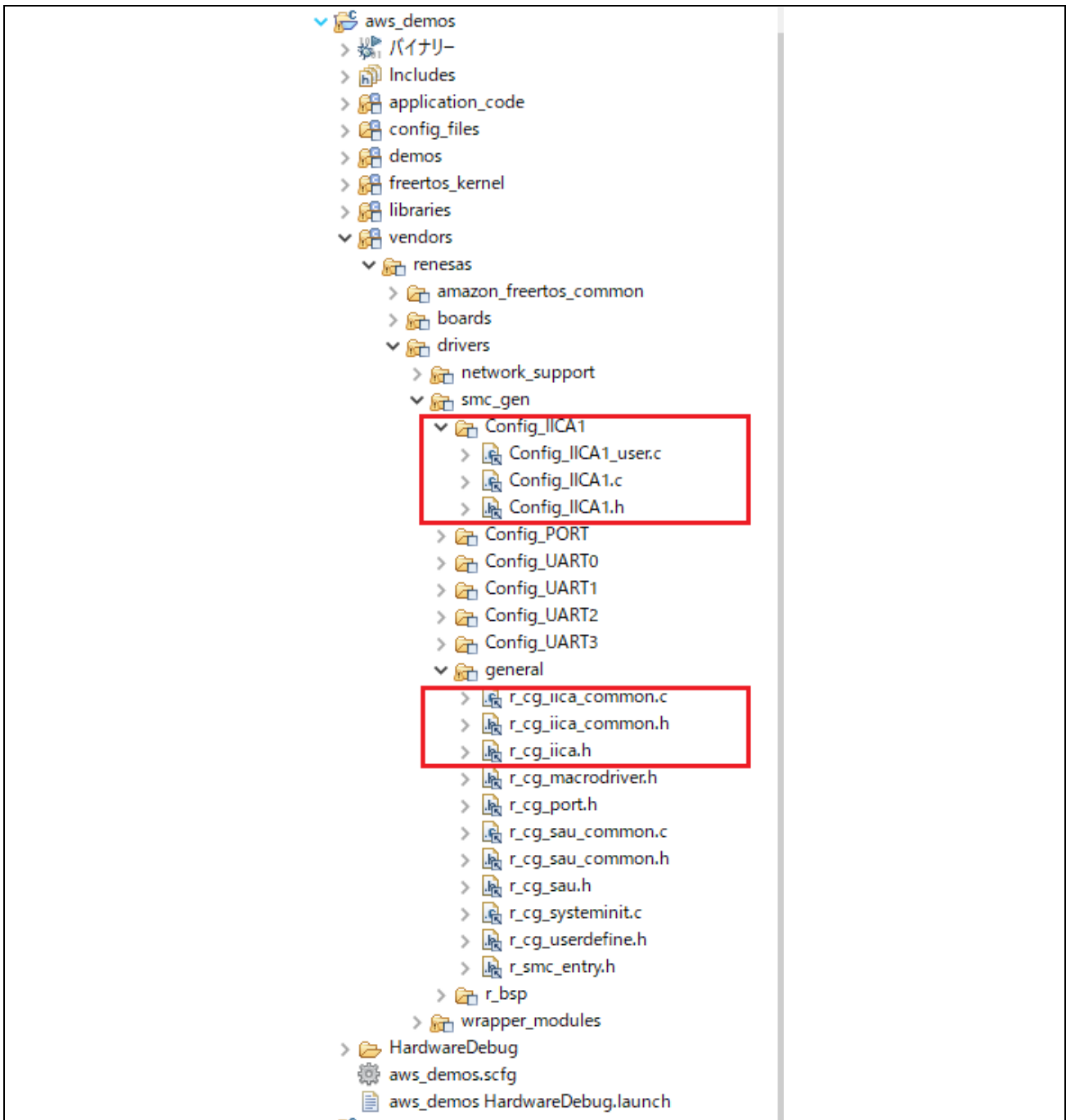


図 12.6 ドライバコードを aws\_demos にコピー

6. コピーしたドライバを起動できるようにします。

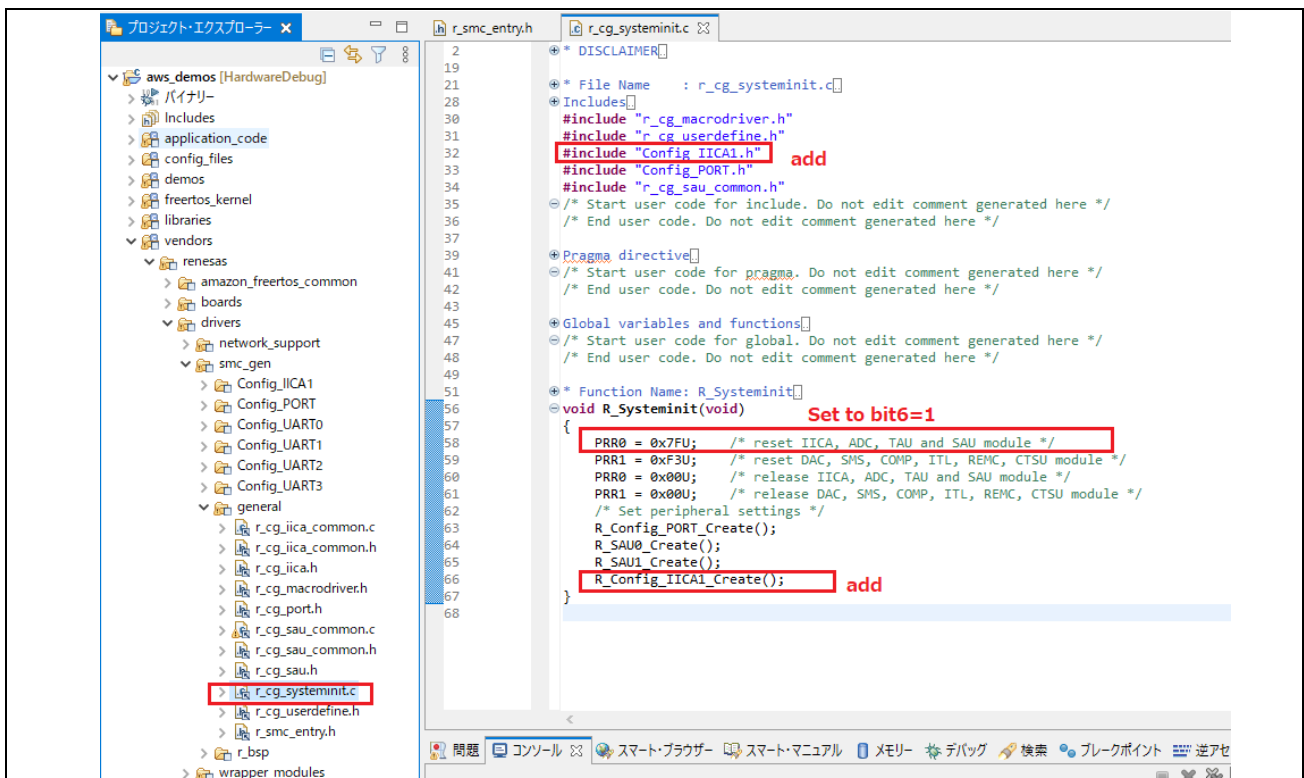


図 12.7 追加したドライバを組み込むー 1

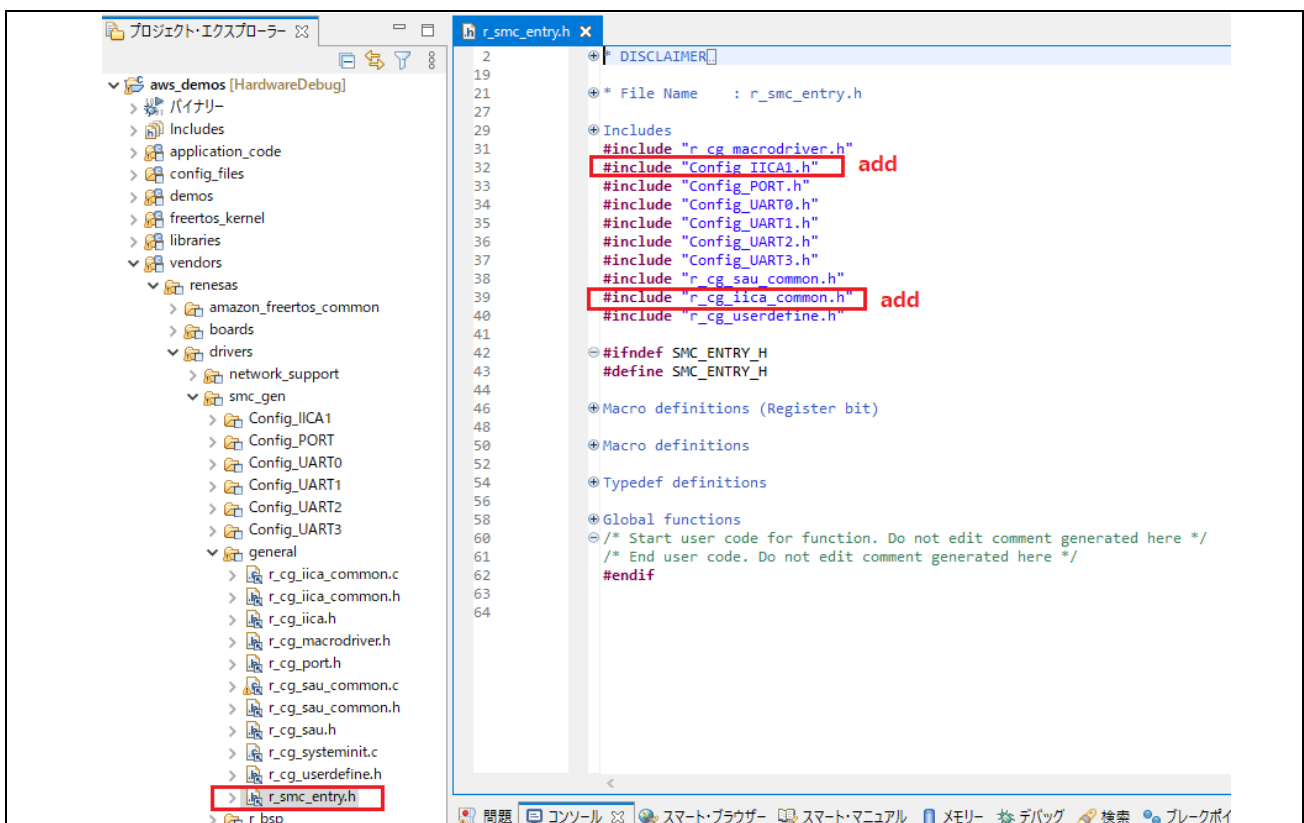


図 12.8 追加したドライバを組み込むー 2

ウェブサイトおよびサポート

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

Renesas Amazon FreeRTOS GitHub: <https://github.com/renesas/amazon-freertos>

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Dec.01.21	-	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っていると、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、変更、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、変更、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、これに限られません。）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](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。