

RX65N グループ

RX65N Cloud Kit で FreeRTOS を用いて Amazon Web Services でセンサー情報を可視化および制御する方法

要旨

本ドキュメントでは、ルネサス製ボードとして、RX65N Cloud Kit を使用したシステムについて説明します。このシステムは、RX65N に Amazon FreeRTOS を実装し、Wi-Fi 経由で Amazon Web Services（以下、AWS）上でのセンサー情報（光）の可視化および、shadow を用いたボード上の LED 制御が可能になります。

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。

e² 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、リンカ制御）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。

注：Amazon Elasticsearch Service は今後 Amazon OpenSearch Service への名称変更を予定されていますが、本ドキュメントでは名称を Amazon Elasticsearch Service として説明させていただきます。

本ドキュメントの目的

本ドキュメントでは、Amazon FreeRTOS デモアプリケーションを使用して、AWS との接続/制御を実行する手順（AWS との接続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

動作は以下の環境で確認しました。

統合開発環境	e ² studio 2021-04
ボード	RX65N Cloud Kit
ツールチェーン	CC-RX Compiler v3.03
	GCC for Renesas RX 8.3.0.202004
エミュレータ	E2 エミュレータ Lite (オンボード)

RX クラウドソリューションの開発に必要なボード、関連プログラム、開発環境に関する情報は以下のリンクにまとめられています。

<https://www.renesas.com/rx-cloud>

関連ドキュメント

RX ファミリー Amazon Web Service 使用時のトラブルシューティング集 (R20AN0624)

目次

1. 用語.....	3
2. 準備.....	4
2.1 ハードウェア構成.....	4
2.2 ソフトウェア構成.....	4
2.3 Tera Term 設定.....	5
3. システム図.....	6
4. AWS との接続.....	7
4.1 AWS 準備.....	7
4.2 ハードウェア準備.....	8
4.3 ソフトウェア準備.....	9
5. shadow.....	15
5.1 デバイスプロパティ.....	15
5.2 shadow の操作手順.....	16
6. Elasticsearch.....	19
6.1 Elasticsearch の準備.....	20
6.2 Kibana の準備.....	26
6.3 IoT Rule の準備.....	30
6.4 デモプログラムの実行.....	36
6.5 Kibana でセンサー情報を可視化.....	37
6.6 デモプログラム実行後の注意事項について.....	45
7. ウェブサイトおよびサポート.....	46
改訂記録.....	47

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 Diligent Inc.の商標です。 (<https://store.diligentinc.com/>)

1. 用語

本資料中の用語を説明します。

表 1-1 用語集

用語	意味
AWS	Amazon Web Service
shadow	デバイスの現在の状態情報の保存と取得に使用される。
Amazon Elastic Service	Elasticsearch クラウドの AWS クラスターを、デプロイ、運用、スケールするマネージ型サービス。

2. 準備

本ドキュメントでは、プロジェクトのインポートから RX65N Cloud Kit を使用したデモを実行するまでを説明します。

2.1 ハードウェア構成

本デモプロジェクトのハードウェア構成を下表に示します。

表 2-1 ハードウェア構成

Item	Content	Provider	Description
使用ボード (RX65N Cloud Kit 同梱物)	Target Board for RX65N	Renesas Electronics Corporation	RX65N MCU 搭載の評価ボード ^注
	RX Cloud Option Board		AWS 接続可能なクラウド通信評価ボード ^注
	Silex Pmod Module		無線 LAN モジュール搭載の通信ボード ^注
Wi-Fi	無線ルーター	-	無線 LAN 規格 : IEEE 802.11b/g/n (2.4GHz) 暗号化方式 : ES
PC	Windows 10	-	推奨 OS
	Google Chrome	-	使用するブラウザ

注 Target Board for RX65N と RX65N Cloud Option Board、Silex Pmod Module は、RX65N Cloud kit に同梱されています。

2.2 ソフトウェア構成

本デモプロジェクトのソフトウェア構成を下表に示します。

表 2-2 ソフトウェア構成

Item	Content	Version
統合開発環境	e ² studio	2021-04
コンパイラ	CC-RX	V3.03
	GCC for Renesas RX	8.3.0.202004
通信ソフト	Tera Term	Version 4.71
エミュレータ	E2 Lite(オンボード)	-

本アプリケーションで使用するヒープ領域を確保するため、r_bsp_config.h の BSP_CFG_HEAP_BYTES マクロの値を 0x400 から 0x1000 に変更しています。

2.3 Tera Term 設定

本デモプロジェクトの Tera Term 設定を下表に示します。

表 2-3 Tera Term 設定

Item	Setting
ボー・レート	115200
データ長	8bit
パリティ	none
ストップビット	1bit
フロー制御	none

3. システム図

以下に、光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および、shadow を使用して RX65N Cloud Kit を制御するシステム図を示します。

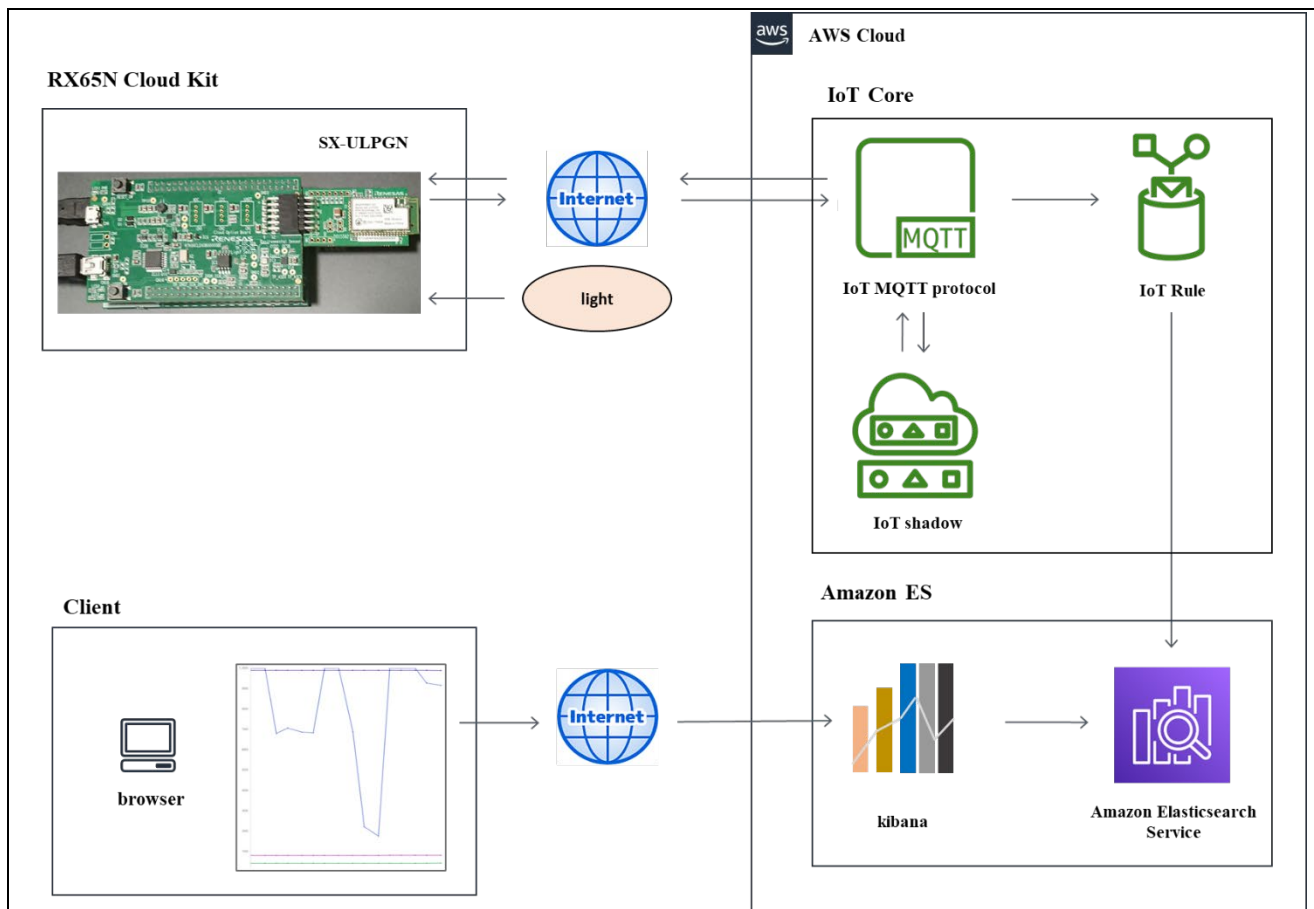


図 3-1 センサー情報を取得してから、可視化するまでのシステム図

4. AWS との接続

RX65N Cloud Kit と AWS を接続させるために必要な準備をします。

4.1 AWS 準備

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

日本語版 : [デバイスを AWS IoT に登録する](#) · [renesas/amazon-freertos Wiki](#) · [GitHub](#)

英語版 : <https://github.com/renesas/amazon-freertos/wiki/Register-device-to-AWS-IoT>

注 : 「AWS IoT のエンドポイントを確認する」まで行ってください。

4.2 ハードウェア準備

デモプログラムを実行するためのハードウェアの準備をします。

1. Target Board (下段ボード) の EJ2 ジャンパピンを外す。
2. Target Board (下段ボード) の ECN1 コネクタと PC を USB ケーブルで接続する。
3. Cloud Option Board (上段ボード) の CN18 コネクタと PC を USB ケーブルで接続する。

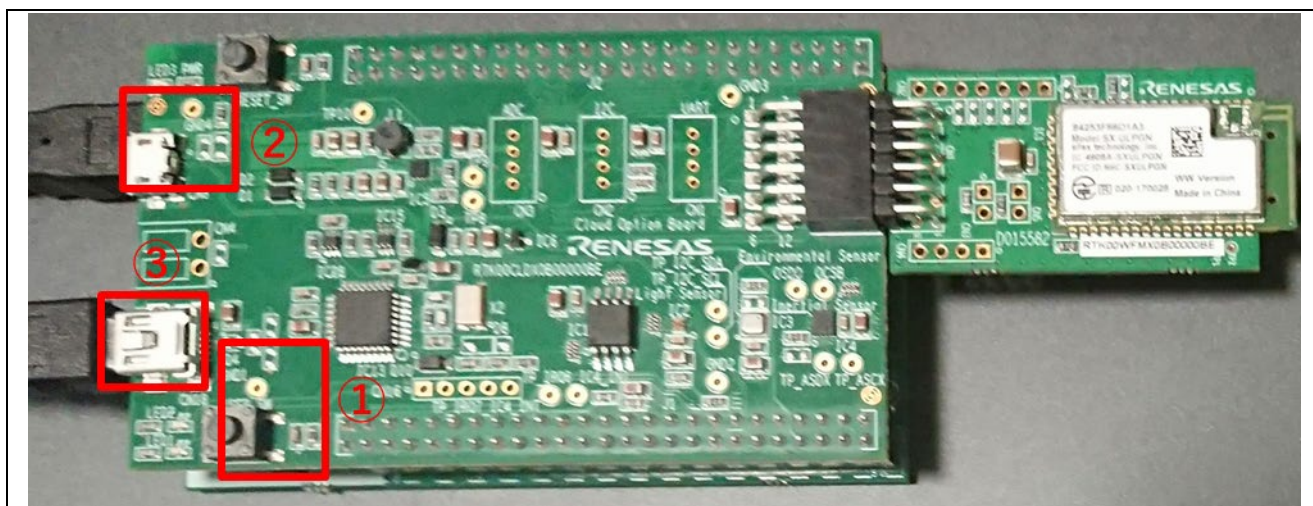


図 4-1 RX65N Cloud Kit (上段)

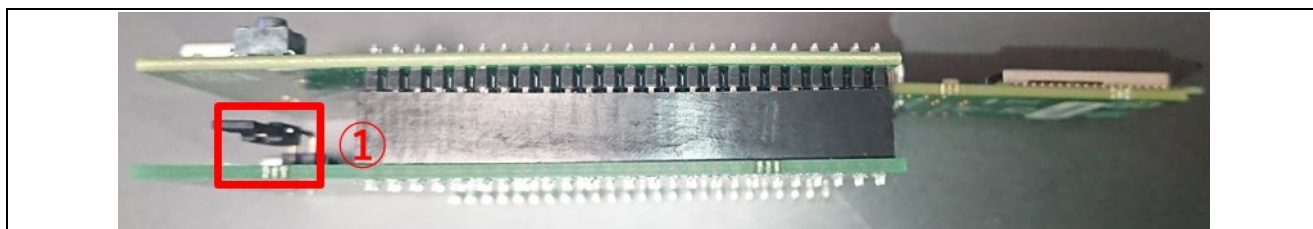


図 4-2 RX65N Cloud Kit (下段)

4.3 ソフトウェア準備

デモプログラムを実行するためのソフトウェアの準備をします。

1. プロジェクトを解凍し、プロジェクトを配置する。(以下、プロジェクトを配置したルートフォルダを $\${base_folder}$ と表記します。)

注：解凍したプロジェクトは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. e²-studio を起動して、ワークスペースを指定する。

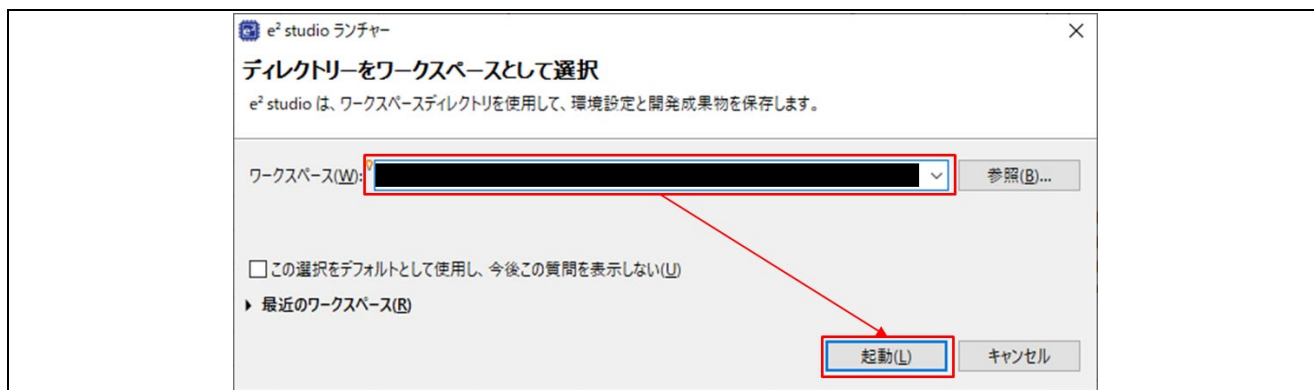


図 4-3 ワークスペース選択画面

3. ファイル -> インポートをクリック。

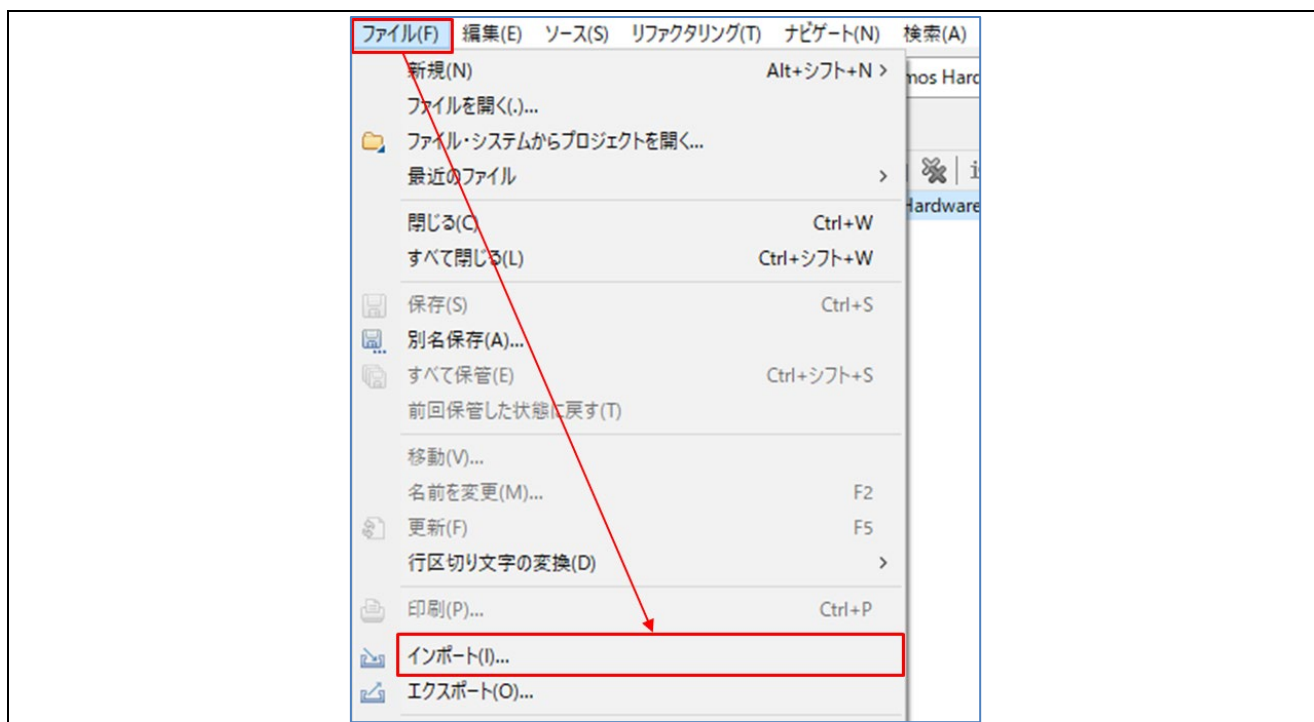


図 4-4 ファイル -> インポート

4. 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。

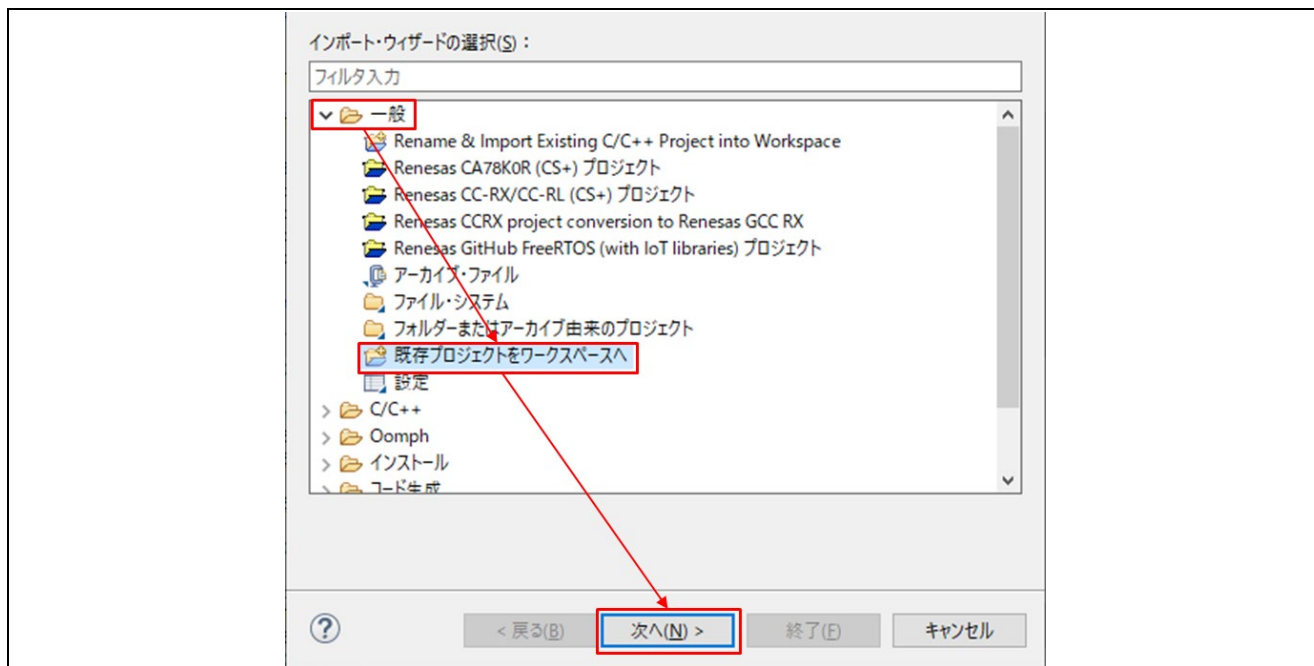


図 4-5 一般 -> 既存プロジェクトをワークスペースへ -> 次へ

5. 「参照」で、以下のルートフォルダを指定する。

・ `${base_folder}\projects\renesas\rx65n-cloud-kit-uart-sx-ulpgn\e2studio\aws_demos`

(使用コンパイラが CC-RX の場合)

・ `${base_folder}\projects\renesas\rx65n-cloud-kit-uart-sx-ulpgn\e2studio-gcc\aws_demos`

(使用コンパイラが GCC for Renesas RX の場合)

その後、「終了」をクリック。

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。

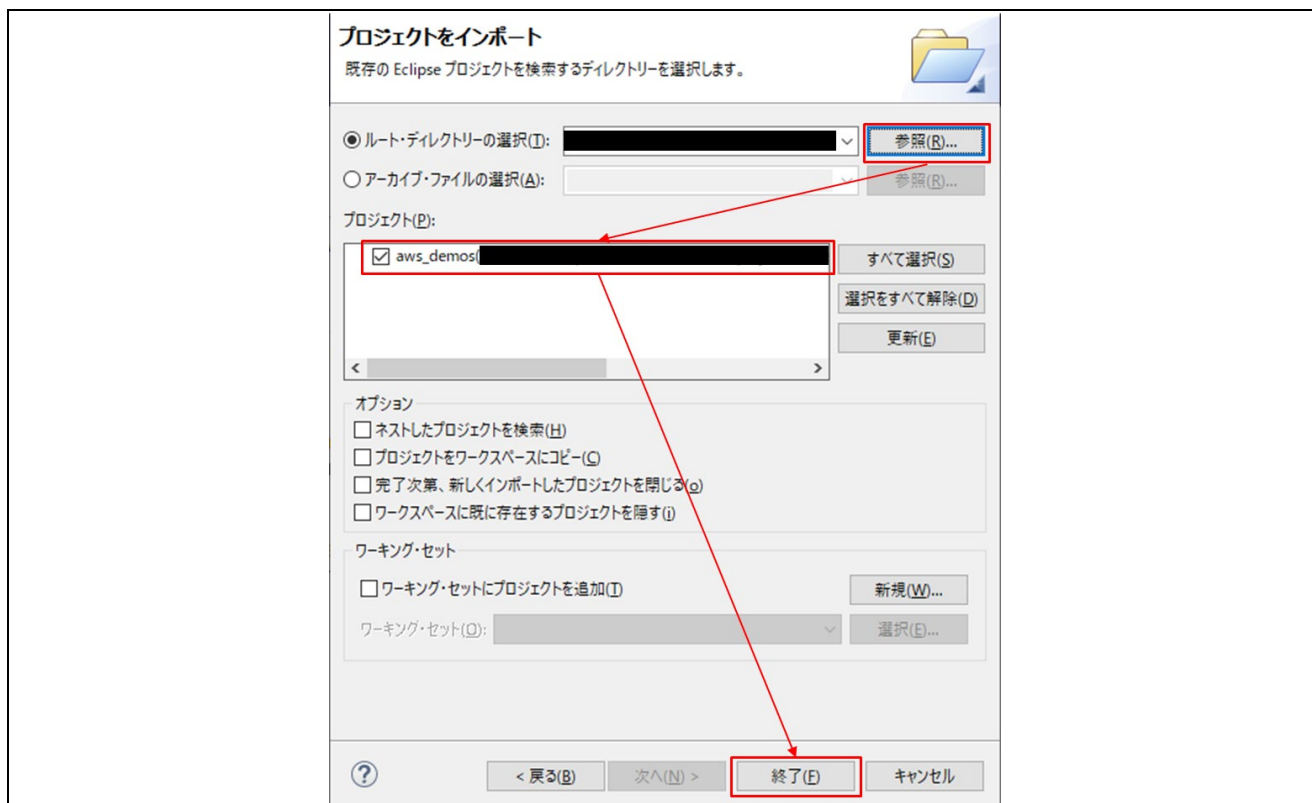


図 4-6 一般 -> 既存プロジェクトをワークスペースへ -> 次へ

6. `$(base_folder)\demos\include\aws_clientcredential.h` にある 4 つのマクロを設定する。

- `clientcredentialMQTT_BROKER_ENDPOINT` -> 「4.1 AWS 準備」で確認した エンドポイント の名前
- `clientcredentialIOT_THING_NAME` -> 「4.1 AWS 準備」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
(上記のマクロは、下図のように、” ”の中に入力してください。)

注 : Wi-Fi の暗号化規格は、「2.1 ハードウェア構成」で示した規格をデフォルトとしています。

暗号化規格が異なる場合は、追加で `clientcredentialWIFI_SECURITY` の設定が必要になります。

```
/*
 * @brief MQTT Broker endpoint.
 *
 * @todo Set this to the fully-qualified DNS name of your MQTT broker.
 */
#define clientcredentialMQTT_BROKER_ENDPOINT "XXXXXXXXXX"

/*
 * @brief Host name.
 *
 * @todo Set this to the unique name of your IoT Thing.
 */
#define clientcredentialIOT_THING_NAME "XXXXXXXXXX"

/*
 * @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.
 *
 * @todo If you are using Wi-Fi, set this to your network name.
 */
#define clientcredentialWIFI_SSID "XXXXXXXXXX"

/*
 * @brief Password needed to join Wi-Fi network.
 * @todo If you are using WPA, set this to your network password.
 */
#define clientcredentialWIFI_PASSWORD "XXXXXXXXXX"

/*
 * @brief Wi-Fi network security type.
 *
 * @see WiFiSecurity_t.
 *
 * @note Possible values are eWiFiSecurityOpen, eWiFiSecurityWEP, eWiFiSecurityWPA,
 * eWiFiSecurityWPA2 (depending on the support of your device Wi-Fi radio).
 */
#define clientcredentialWIFI_SECURITY WiFiSecurityWPA2

#endif /* ifndef _AWS_CLIENTCREDENTIAL_H_ */
```

図 4-7 aws_clientcredential.h

7. `$(base_folder)\tools\certificate_configuration\CertificateConfigurator.html` をダブルクリック。

js	2021/05/10 15:33	ファイル フォルダー	
CertificateConfigurator.html	2021/05/06 11:03	Chrome HTML Do...	5 KB
PEMfileToCString.html	2021/05/06 11:02	Chrome HTML Do...	4 KB

図 4-8 CertificateConfigurator.html の起動

8. 「4.1 AWS 準備」でダウンロードしたモノの証明書/プライベートキーの以下 2 ファイルを指定し、「Generate and save aws_clientcredential_key.h」をクリック。

- xxxxxx-certificate.pem.crt (モノの証明書)
- xxxxxx-private.pem.key (プライベートキー)

Certificate Configuration Tool
Amazon FreeRTOS Developer Demos

Provide client certificate and private key PEM files downloaded from the AWS IoT Console.

Certificate PEM file:
ファイルの選択 [redacted]-certificate.pem.crt

Private Key PEM file:
ファイルの選択 [redacted]-private.pem.key

Generate and save aws_clientcredential_keys.h

Save the generated header file to the `demos/common/include` folder of the demo project.

Copyright (C) 2017 Amazon.com, Inc. or its affiliates. All Rights Reserved.

図 4-9 Certificate Configuration Tool

9. 生成された `aws_clientcredential_key.h` を `$(base_folder)\demos\common\include\aws_clientcredential_key.h` に上書きする。

aws_application_version.h	2021/04/27 12:04	C言語ヘッダファイル	2 KB
aws_ble_gatt_server_demo.h	2021/04/08 10:17	C言語ヘッダファイル	5 KB
aws_clientcredential.h	2021/04/12 9:41	C言語ヘッダファイル	3 KB
aws_clientcredential_keys.h	2021/04/08 16:12	C言語ヘッダファイル	4 KB
aws_demo.h	2021/04/08 10:17	C言語ヘッダファイル	3 KB
aws_iot_demo_network.h	2021/04/08 10:17	C言語ヘッダファイル	3 KB
aws_ota_codesigner_certificate.h	2021/04/08 10:17	C言語ヘッダファイル	2 KB
aws_wifi_connect_task.h	2021/04/08 10:17	C言語ヘッダファイル	2 KB
iot_ble_numericComparison.h	2021/04/08 10:17	C言語ヘッダファイル	3 KB
iot_config_common.h	2021/04/08 10:17	C言語ヘッダファイル	9 KB
iot_demo_logging.h	2021/04/08 10:17	C言語ヘッダファイル	3 KB
iot_demo_runner.h	2021/04/19 17:23	C言語ヘッダファイル	6 KB

図 4-10 aws_clientcredential_key.h の上書き

10. プロジェクト -> 「すべてビルド」をクリックし、0 errors であることを確認する。

注：初回ビルド時はプロジェクトクリーン後に実行してください。また、初回以降でもビルドエラーが出る場合は、再度クリーンを行ってからビルドしてください。

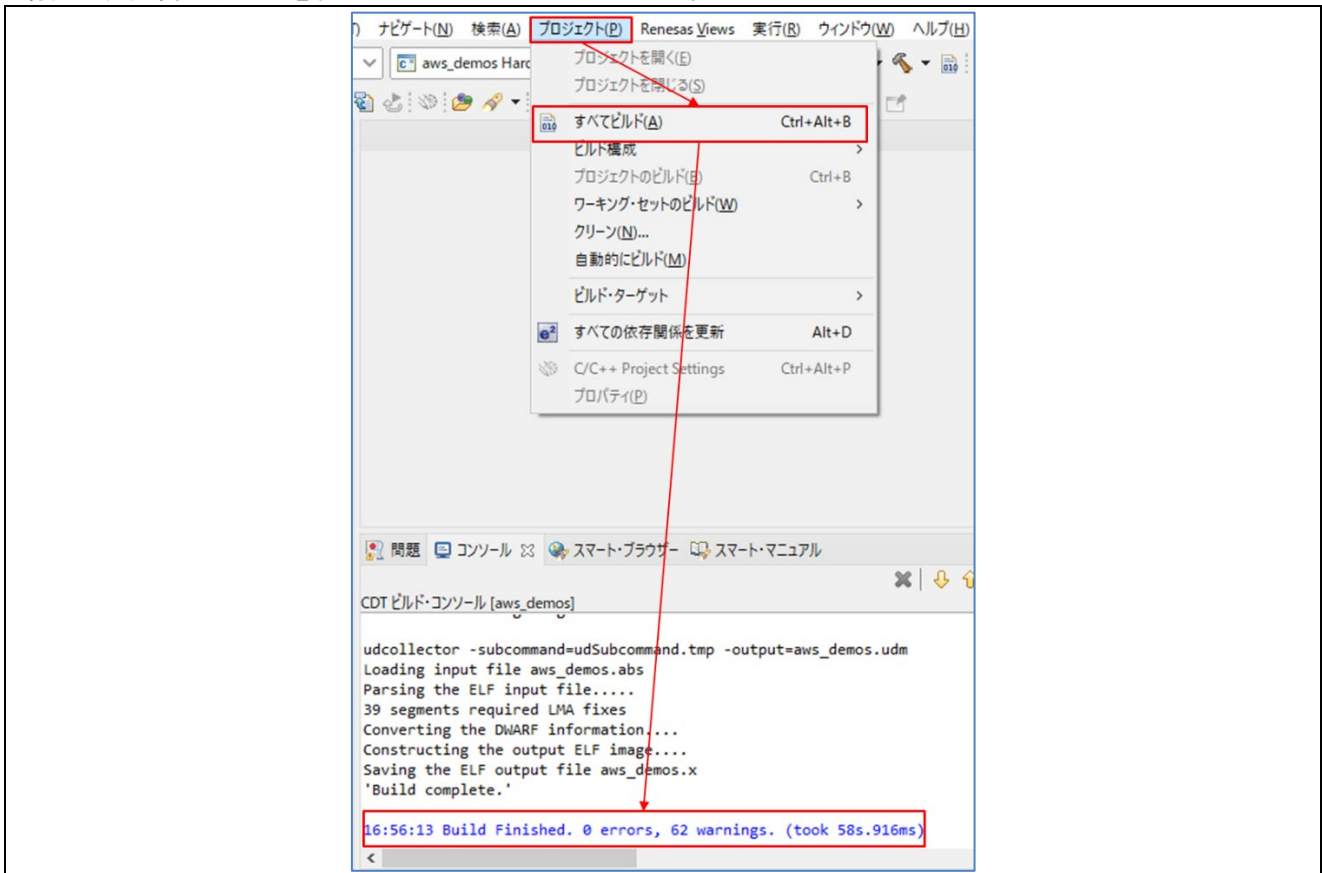


図 4-11 プロジェクト -> すべてビルド -> 0 errors

AWS の接続ができていないか確認したい場合は、参考資料「Amazon Web Service 使用時のトラブルシューティング集 2.AWS 接続確認」を参照してください。

5. shadow

本章では、AWS IoT Core の shadow を使用して、クラウド側から、エッジデバイス（RX65N Cloud Kit）を制御する方法を紹介します。

エッジデバイスからデータを集めるだけでなく、クラウドからエッジデバイスを制御することで、リモートで製品の制御が行え、様々なアプリケーションの要求を実現することができます。

本サンプルコードでは LED やセンサーの取得データに応じた LED 制御に切り替える動作が行えます。

5.1 デバイスプロパティ

本デモで使用できるデバイスプロパティの一覧を下表に示します。

表 5-1 デバイスプロパティ一覧

プロパティ	ステータス	動作
"LEDControl"	"LED_ON"	RX65N Cloud Kit 上段の LED1 (赤) と LED2 (赤) が点灯する。
	"LED_OFF"	RX65N Cloud Kit 上段の LED1 (赤) と LED2 (赤) が消灯する。
	"LED_LIGHT"	光センサーの値が 500 以上 : LED1 (赤) / LED2 (赤) 消灯 光センサーの値が 100 以上 500 未満 : LED1 (赤) 点灯、LED2 (赤) 消灯 光センサーの値が 100 未満 : LED1 (赤) / LED2 (赤) 点灯
	"LED_TEMP"	温度センサーの値が 30 以下 : LED1 (赤) / LED2 (赤) 消灯 温度センサーの値が 30 超過 40 以下 : LED1 (赤) 点灯、LED2 (赤) 消灯 温度センサーの値が 40 超過 : LED1 (赤) / LED2 (赤) 点灯
"SWVersion"	"VER_x.y.z" (注 : x,y,z は任意の正の整数)	SW バージョンを x.y.z に変更する。(初期状態は aws_application_version.h の値を使用) SW バージョンが変更されたとき、LED1 (赤) と LED2 (赤) が 10 秒点滅する。
"IPAddress"	なし	IP アドレスを表示する。 (R_WIFI_SX_ULPGN_GetIpAddress の値を使用)
"sensorDataUpdateOn"	"UpdateOn"	光センサーから読み取った値をアップロードする。
	"UpdateOff"	固定値で 0 をアップロードする。

5.2 shadow の操作手順

shadow を操作する手順を以下に示します。

1. 「4. AWS との接続」の手順をすべて行う。
2. AWS のサービス -> 全てのサービス -> IoT -> IoT Core に移動し、テスト -> トピックへサブスクライブする -> トピックのサブスクリプションに「#」を指定 -> 「サブスクライブ」をクリック。



図 5-1 トピックへのサブスクライブ

3. e²-studio 画面右上の虫アイコンをクリック。

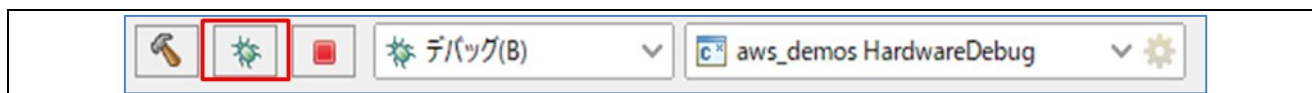


図 5-2 デバッグ

4. パースペクティブ切り替えの確認が出現したら、「切り替え」をクリック。

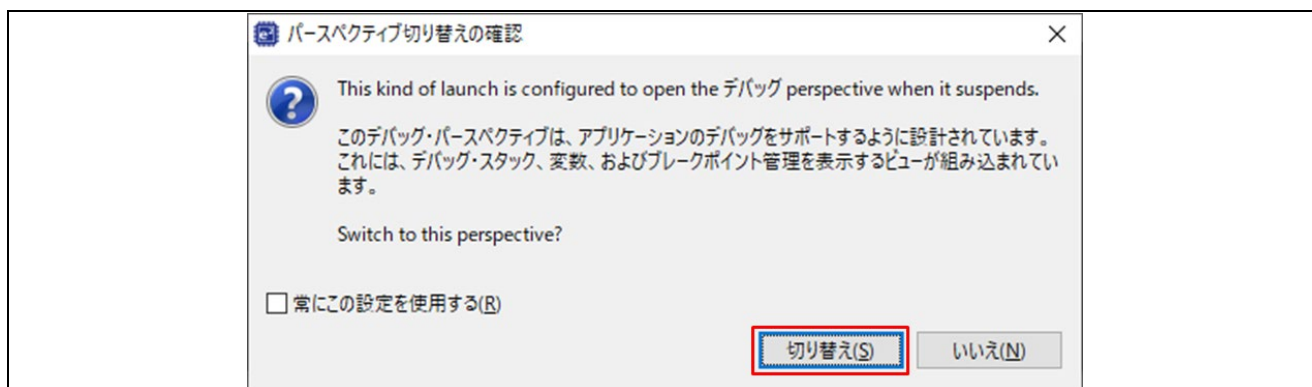


図 5-3 パースペクティブ切り替えの確認

5. 「再開」をクリックする。しばらくして main 関数に止まるので再度「再開」をクリック。

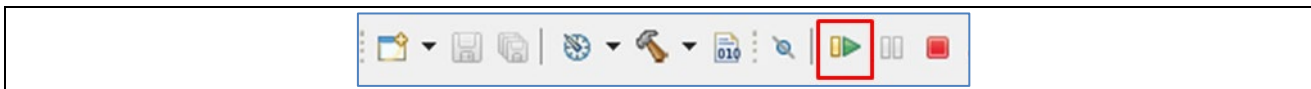


図 5-4 デモプログラムの実行

6. AWS のページに戻り、テスト -> トピックに公開するに移動し、トピック名に以下のコードを入力する
注 : xxxx の部分は、「4.1 AWS 準備」で登録したモノの名前を入力してください。

```
$saws/things/xxxx/shadow/update
```

図 5-5 トピック名

7. 以下のコードをコピーしメッセージペイロードに張り付ける。

(ここでは、例として LED_ON を設定していますが、デバイスプロパティ一覧に記載されたステータスに設定することで、ステータスに対応する動作になります。)

```
{
  "state": {
    "desired": {
      "LEDControl": "LED_ON",
      "SWVersion": "VER_0.9.3",
      "SensorDataUpdateOn": "UpdateOn"
    }
  }
}
```

図 5-6 メッセージペイロード

8. 発行をクリックして、RX65N Cloud Kit 上段の LED1/LED2 が点灯していることを確認する。

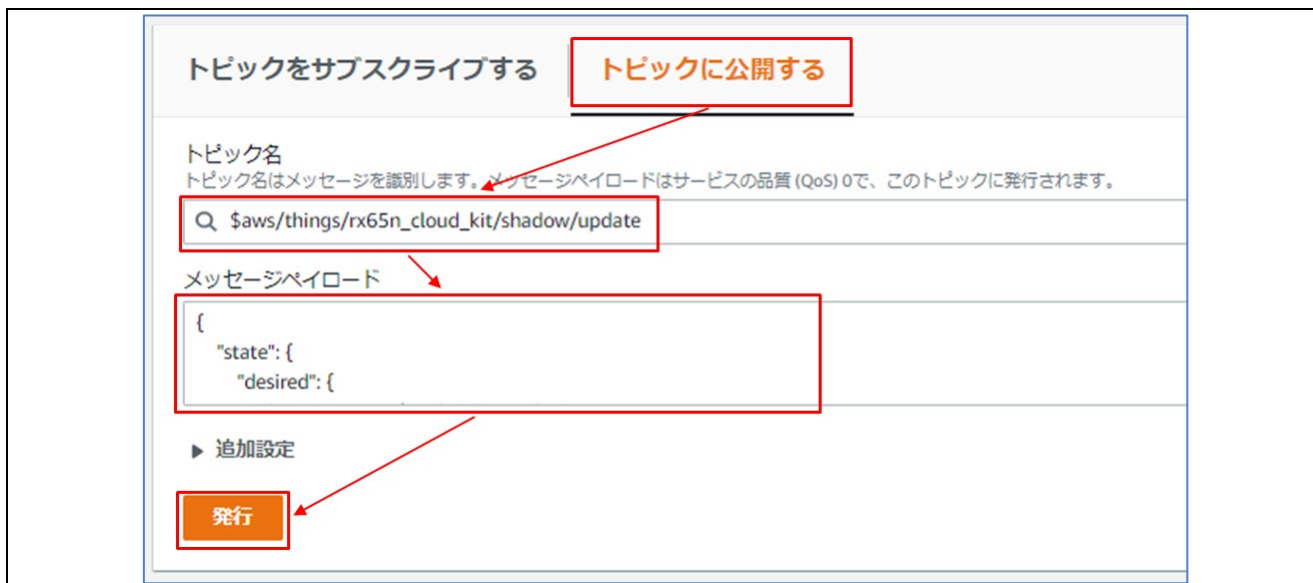


図 5-7 トピックの発行

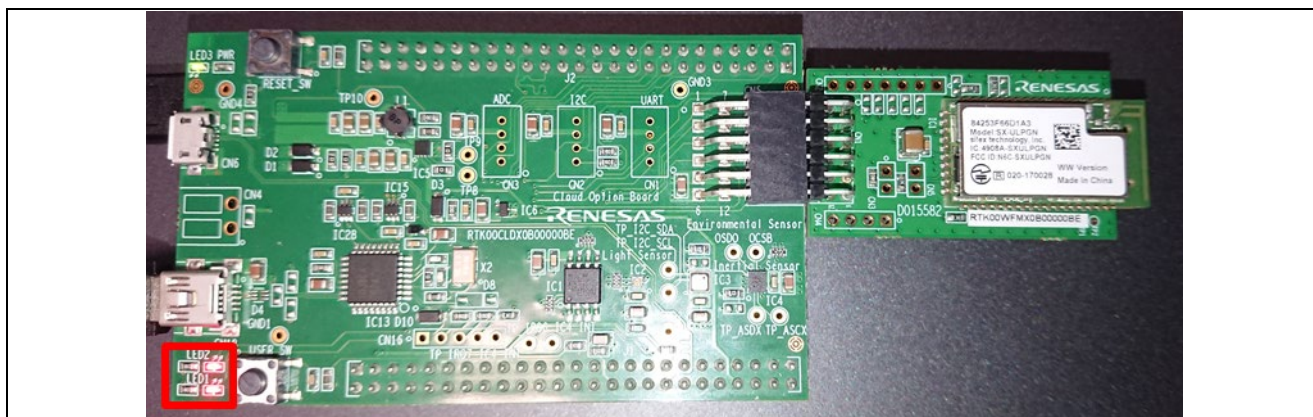


図 5-8 RX65N Cloud Kit LED1/LED2 点灯

6. Elasticsearch

RX65N Cloud Kit に搭載されたセンサーモジュールから取得したデータを、Elasticsearch を使用して AWS で可視化します。

Amazon Elasticsearch 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に、Elasticsearch のドメイン削除を忘れずに行ってください。

6.1 Elasticsearch の準備

デモプログラムを実行するための Elasticsearch の準備をします。

以下の手順で Elasticsearch を設定します。

1. AWS マネジメントコンソールから、「すべてのサービス -> 分析 -> Elasticsearch Service」をクリック。

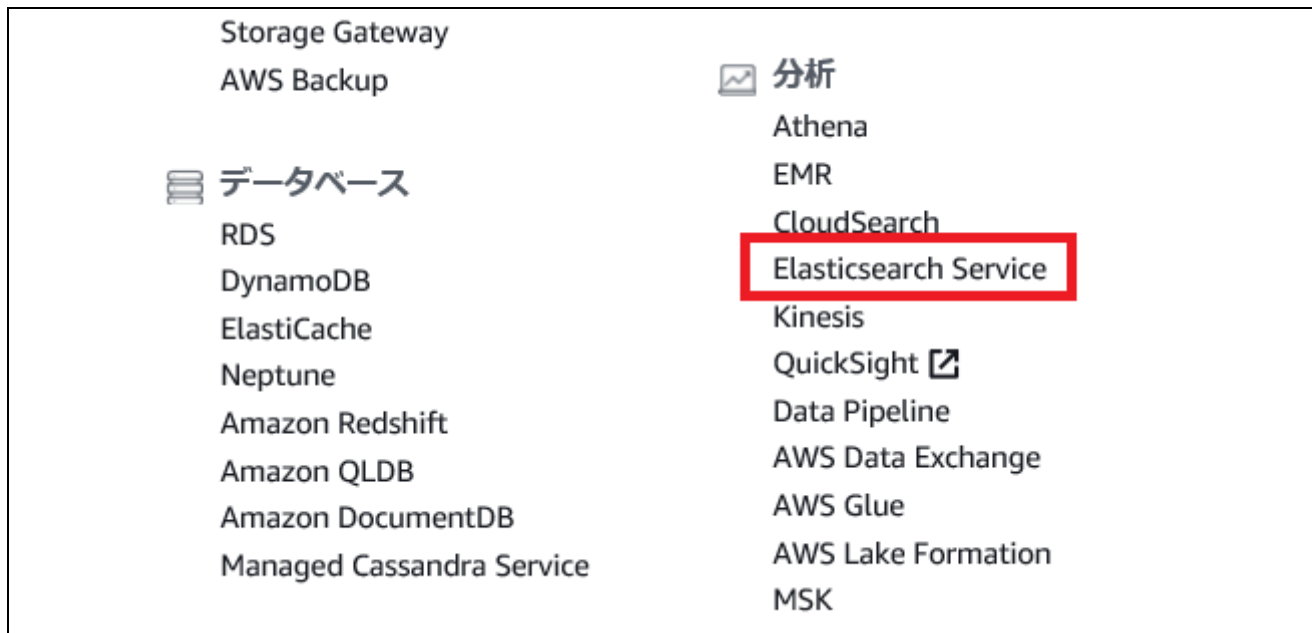


図 6-1 Elasticsearch Service を選択

2. 「新しいドメインの作成」をクリック。



図 6-2 新しいドメインの作成

3. 開発およびテストにチェックを入れて、Elasticsearch のバージョンを 7.1 に設定する。その後、「次へ」をクリック。

デプロイタイプの選択

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。

デプロイタイプ

- 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
Elasticsearch エンドポイントが 1 つのみ必要な場合、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

ドメインの Elasticsearch バージョンを選択します。

Elasticsearch のバージョン 7.1

キャンセル 次へ

図 6-3 デプロイタイプの選択

4. Elasticsearch ドメイン名を入力してから、インスタンスタイプを t2.small.elasticsearch に変更して下へスクロールする。

ドメインの設定

ドメインは、Elasticsearch の実行に必要なリソースのコレクションです。ドメイン名はドメインエンドポイントの一部になります。

Elasticsearch ドメイン名

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。

データノード

アプリケーションのコンピューティング、メモリ、ストレージのニーズに応じたインスタンスタイプを選択します。Elasticsearch インデックスのサイズ、シャードとレプリカの数、クエリのタイプ、およびリクエストのボリュームを考慮します。各インスタンスタイプに割り当てられるハードウェアリソースの概要については、[こちらを参照してください](#)

インスタンスタイプ

AWS 無料利用枠では、t2.micro または t2.small インスタンスを 1 か月あたり最大 750 時間、マグネティックまたは汎用 EBS ストレージを最大 10 GiB ご利用いただけます。
[Amazon Elasticsearch Service 無料利用枠](#)
t2.small.elasticsearch インスタンスタイプには EBS ストレージが必要です。

ノードの数

図 6-4 ドメインの設定

5. 「次へ」をクリックする。

自動スナップショットの開始時間

任意の Elasticsearch クラスター設定

キャンセル

図 6-5 ドメイン設定後、次へ

6. パブリックアクセスを選択してから、下へスクロールする。

ネットワーク構成

インターネットまたは VPC アクセスを選択します。VPC アクセスを有効にするため、お客様の VPC のプライベート IP アドレスが使用されます。これにより、デフォルトでセキュリティが提供されます。お客様は、セキュリティグループを使用して VPC 内でネットワークアクセスを制御します。オプションで、制限付きアクセスポリシーを適用してセキュリティのレイヤーを追加できます。インターネットエンドポイントはパブリックアクセス可能です。パブリックアドレスを選択する場合、ドメインにアクセスする特定のユーザーまたは IP アドレスのみを許可するアクセスポリシーで、ドメインをセキュリティ保護する必要があります。

VPC アクセス (推奨)
 パブリックアクセス




図 6-6 ネットワーク構成の設定

7. ドメインアクセスポリシーをカスタムアクセスポリシーに設定し、IPv4 アドレスを選択する。自身の PC のグローバル IP アドレスを入力してから、許可に設定する。

注：グローバル IP アドレスはインターネットで「グローバル IP アドレス 確認方法」と検索して確認します。

アクセスポリシー

アクセスポリシーは、Amazon Elasticsearch Service のドメインに到達したときにリクエストを受け入れるか拒否するかを制御します。このポリシーでアカウント、ユーザー、またはロールを指定する場合、リクエストに署名する必要があります。[詳細はこちら](#)

カスタムポリシービルダーでは、最大 10 個の要素を許可します。JSON 定義のアクセスポリシーを使用して、10 個を超える要素を持つポリシーを定義します。

ドメインアクセスポリシー **カスタムアクセスポリシー**

AWS アカウント ID、アカウント ARN、IAM ユーザー ARN、IAM ロール ARN、IPv4 アドレス、または CIDR ブロックによるアクセスを許可または拒否します。

[要素を追加](#)

図 6-7 アクセスポリシーの設定

8. 「次へ」をクリック。

キャンセル

図 6-8 アクセスポリシーの設定後、次へ

9. タグの追加画面では何もせずに、「次へ」をクリック。

タグの追加 - オプション

タグを追加してドメインについて説明できます。タグは、大文字と小文字が区別されるキーと値のペアから構成されます。たとえば、Environment Name-Development のキーと値のペアでタグを定義できます。ドメインあたり最大 50 個のタグを作成できます。詳細はこちら

キー	値	削除
<input type="text"/>	<input type="text"/>	

一意のキーを追加してください 値を追加してください

キャンセル 戻る **次へ**

図 6-9 何もせず、次へ

10. 確認画面で内容を確認してから、「確認」をクリック。

キャンセル 戻る **確認**

図 6-10 次へをクリック後、確認

11. Elasticsearch のドメインが作成されるので、ドメインのステータスがアクティブになるまで待つ。

注：ドメインのステータスはアクティブになるまで 1~2 時間かかる場合があります。

ドメインの編集 アクセスポリシーの変更 タグの管理 ドメインの削除 ドメインのアップグレード

ドメインはまだ初期化中で、これには約 10 分かかります。初期化が完了するまでは、データをロードしたり、ドメインに対してはできません。ドメインを使用する準備ができるとすぐに、ドメインのステータスが [アクティブ] に変わります。

概要 クラスタの状態 インスタンスの状態 インデックス ログ アップグレード履歴

ドメインのステータス **読み込み中...**

Elasticsearch のバージョン 7.1

エンドポイント

ドメイン ARN [REDACTED]

Kibana

図 6-11 ドメインのステータスがアクティブになるまで待つ

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



図 6-12 ドメインのステータスがアクティブ

6.2 Kibana の準備

デモプログラムを実行するための Kibana の準備をします。

1. 「Explore on my own」 をクリック。

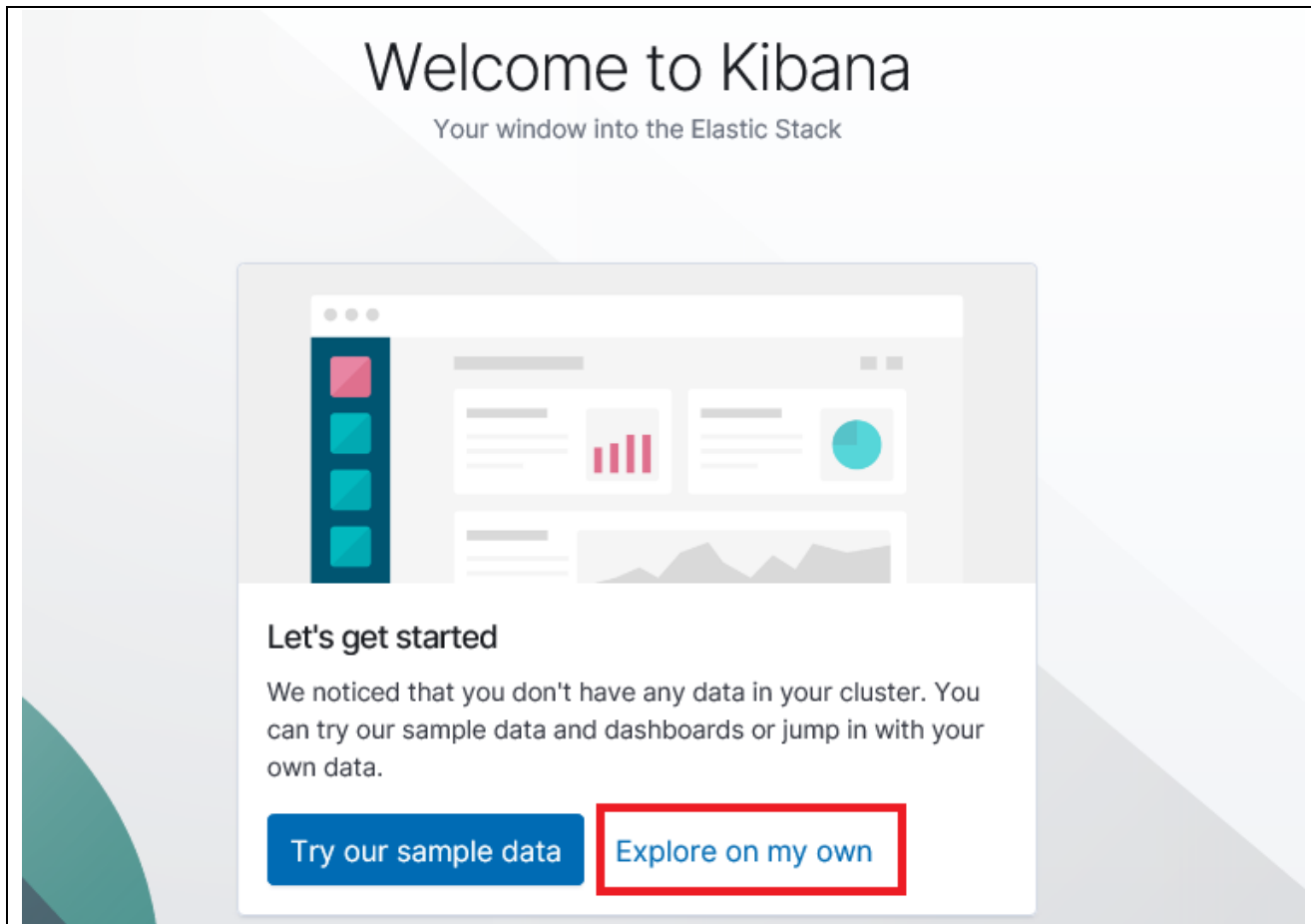


図 6-13 Explore on my own

2. 左側のメニューから、「Dev Tools」をクリック。

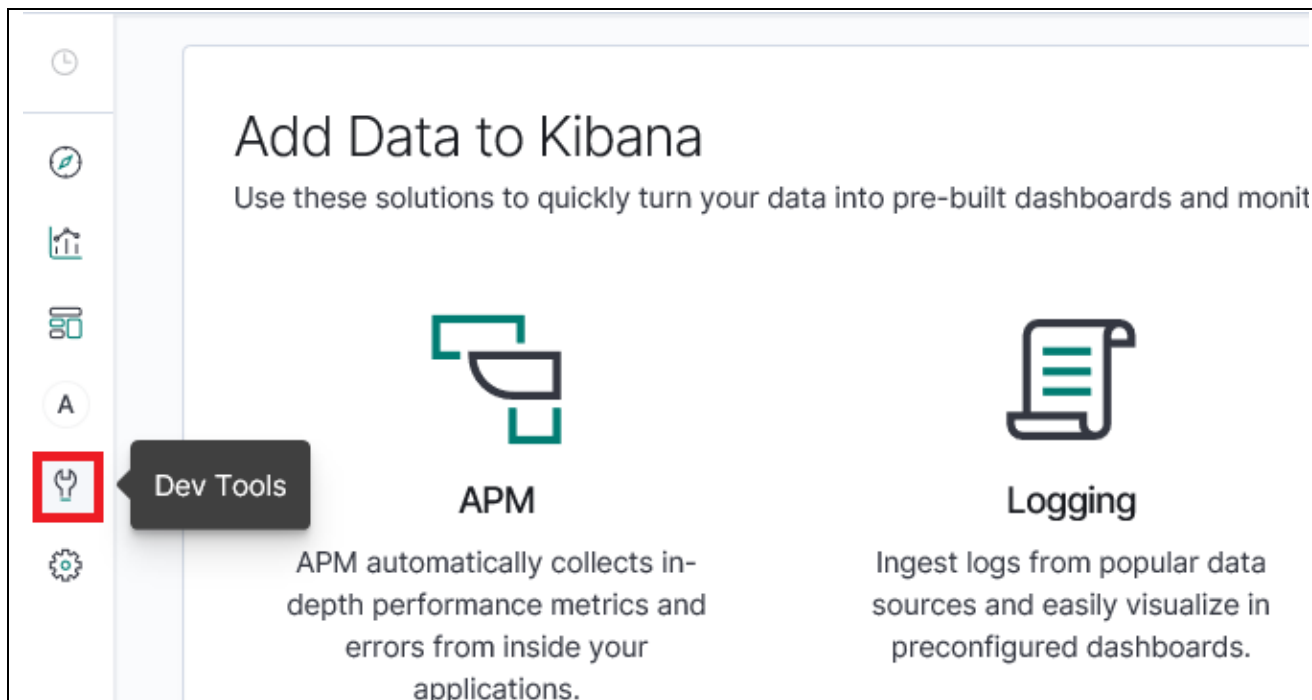


図 6-14 Dev Tools

3. 「Get to work」をクリックします。

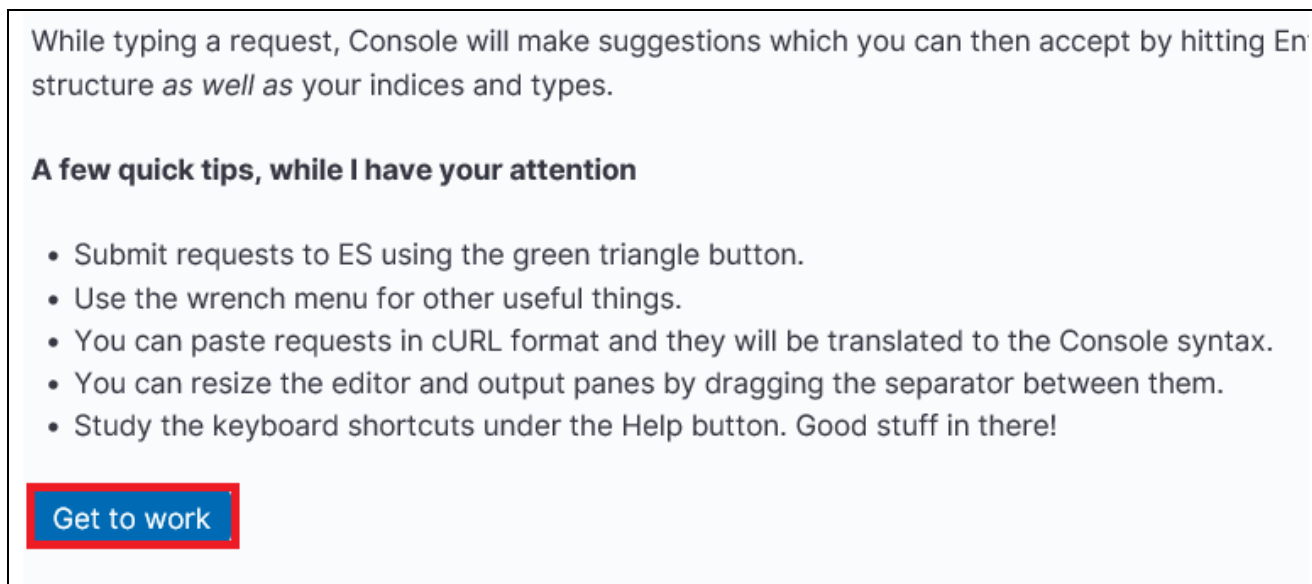


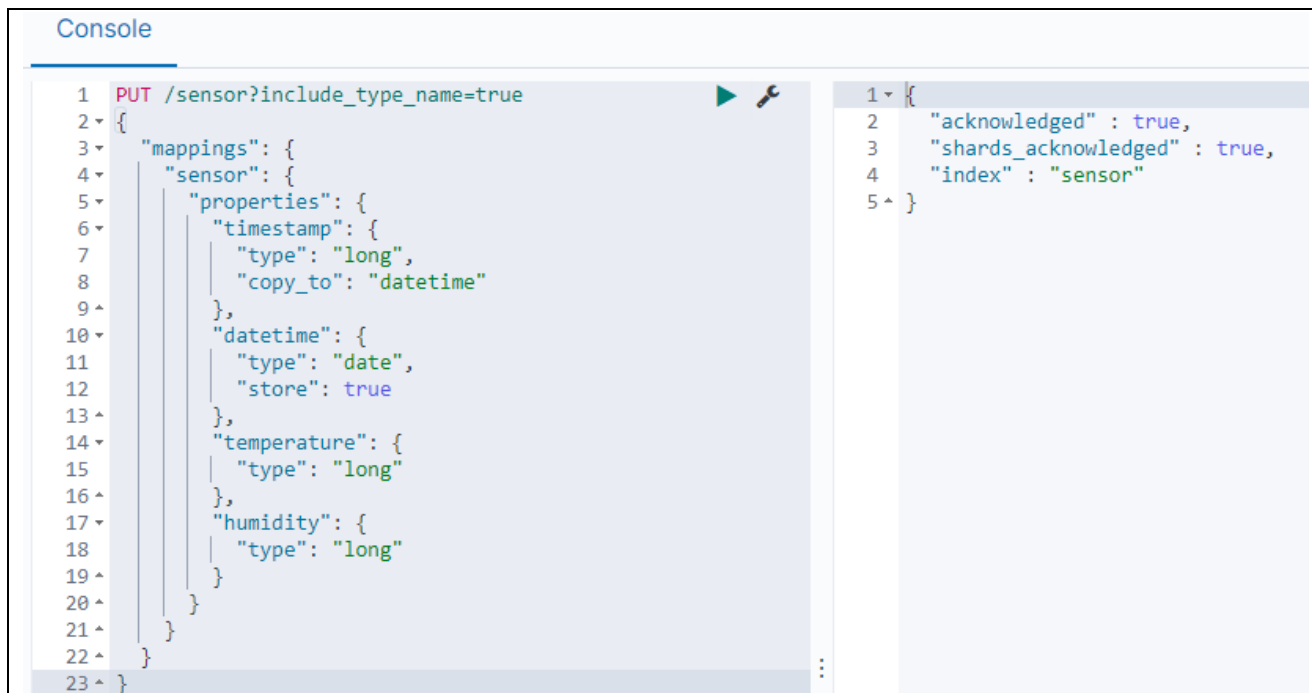
図 6-15 Get to work

4. 左側のコンソールに以下のコードを入力する。

```
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 6-16 コンソールにコードを入力

5. コンソール右上の「click to send request」をクリック。



The screenshot shows a REST client interface with a 'Console' title. On the left, a PUT request is defined with a URL and a JSON body. On the right, the response is shown as a JSON object.

```
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 6-17 click to send request

6. 以下のレスポンスが返却されることを確認する。

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

図 6-18 レスポンスの確認

6.3 IoT Rule の準備

デモプログラムを実行するための、IoT Rule の準備をします。

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



図 6-19 ルールの作成

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する。

```
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 rule configuration page. At the top, there is a '名前' (Name) field and a '説明' (Description) field. Below these is the 'ルールクエリステートメント' (Rule Query Statement) section. It includes a dropdown menu for 'SQLバージョンの使用' (SQL Version) set to '2016-03-23'. Below that is a text area for the 'ルールクエリステートメント' with the following text: 'SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. SQLステートメントを構築する方法については、「AWS IoT SQL リファレンス」を参照してください。' A code editor below shows the SQL query: '1 SELECT *, timestamp() as timestamp FROM 'freertos/demos/sensor'' and '2'. The code editor is highlighted with a red border.

図 6-20 コードの入力

3. 「アクションの追加」をクリック。

The screenshot shows the '1つ以上のアクションを設定する' (Set up one or more actions) section. It contains a paragraph: 'インバウンドメッセージが上記のルールに一致すると、1つ以上のアクションが選択されます。メッセージ受信時に発生する追加アクティビティ(データベースへの格納、クラウド関数の呼び出し、通知の送信など)を定義するアクション。(*必須)' Below this is a button labeled 'アクションの追加' (Add Action), which is highlighted with a red border. Below that is the 'エラーアクション' (Error Action) section, which contains a paragraph: 'ルールの処理中に問題が発生した場合に実行するアクションを任意で設定します。' and another button labeled 'アクションの追加' (Add Action).

図 6-21 アクションの追加

4. 「Amazon Elasticsearch 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。



図 6-22 アクションの設定


5. ドメイン名に 6.1 Elasticsearch の準備 で作成したドメイン名を入力して、ID に\${newuuid()}、索引に sensor、タイプに sensor を入力する。

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

*ドメイン名

*エンドポイント

*ID 

*索引 


*タイプ 

図 6-23 ドメイン名、ID、索引、タイプの設定

6. 「ロールの作成」をクリックしてから、ロールの名前を入力して「ロールの作成」をクリック。

新しいロールの作成

新しい IAM ロールがお客様のアカウントに作成されます。AWS IoT がお客様に代わってリソースにアクセスすることを許可するスコープダウンされたアクセス許可を提供するロールにインラインポリシーがアタッチされます。

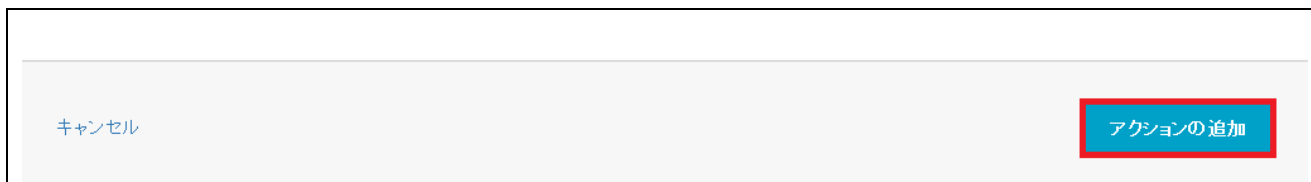
名前

ロールに一意の名前を付ける

キャンセル **ロールの作成**

図 6-24 ロールの作成

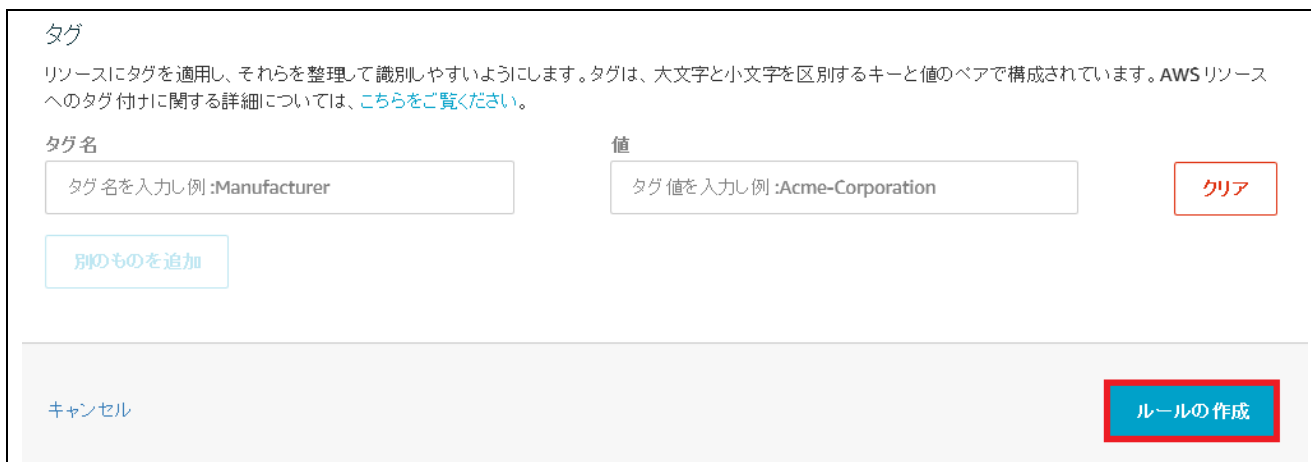
7. 作成したルールが選択されていることを確認してから、「アクションの追加」をクリック。



The screenshot shows a horizontal bar with a light gray background. On the left side, there is a blue button labeled 'キャンセル' (Cancel). On the right side, there is a blue button labeled 'アクションの追加' (Add Action), which is highlighted with a red rectangular border.

図 6-25 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てから、「ルールの作成」をクリック。



The screenshot shows a form for creating a rule. At the top, there is a section titled 'タグ' (Tag) with a paragraph of text: 'リソースにタグを適用し、それらを整理して識別しやすいようにします。タグは、大文字と小文字を区別するキーと値のペアで構成されています。AWS リソースへのタグ付けに関する詳細については、[こちらをご覧ください](#)。' Below this, there are two input fields: 'タグ名' (Tag Name) with the example 'Manufacturer' and '値' (Value) with the example 'Acme-Corporation'. There is a 'クリア' (Clear) button to the right of the value field. Below the input fields, there is a blue button labeled '別のものを追加' (Add another one). At the bottom of the form, there is a horizontal bar with a light gray background. On the left side, there is a blue button labeled 'キャンセル' (Cancel). On the right side, there is a blue button labeled 'ルールの作成' (Create Rule), which is highlighted with a red rectangular border.

図 6-26 ルールの作成

6.4 デモプログラムの実行

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、RX65N Cloud Kit に接続する。

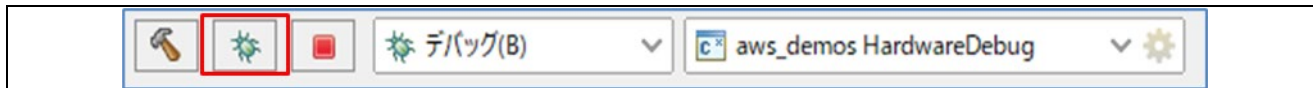


図 6-27 デバッグ

2. 「再開」をクリックする。しばらくして main 関数に止まるので再度「再開」をクリック。



図 6-28 デモプログラムの実行

6.5 Kibana でセンサー情報を可視化

センサー情報を可視化するための Kibana の操作手順を以下に示します。

1. Kibana に移動して、左のメニューから Management をクリック。

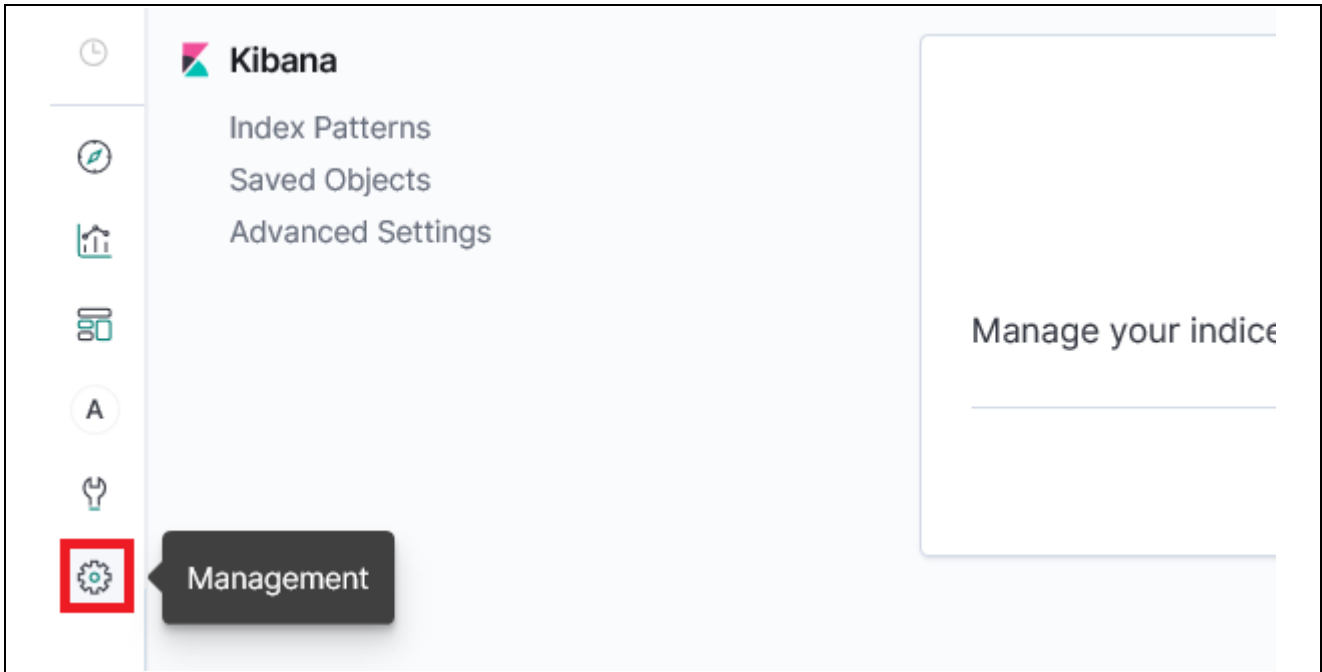


図 6-29 Kibana の設定

2. Index Patterns をクリックしてから、Index pattern に sensor と入力して、> Next step をクリック。

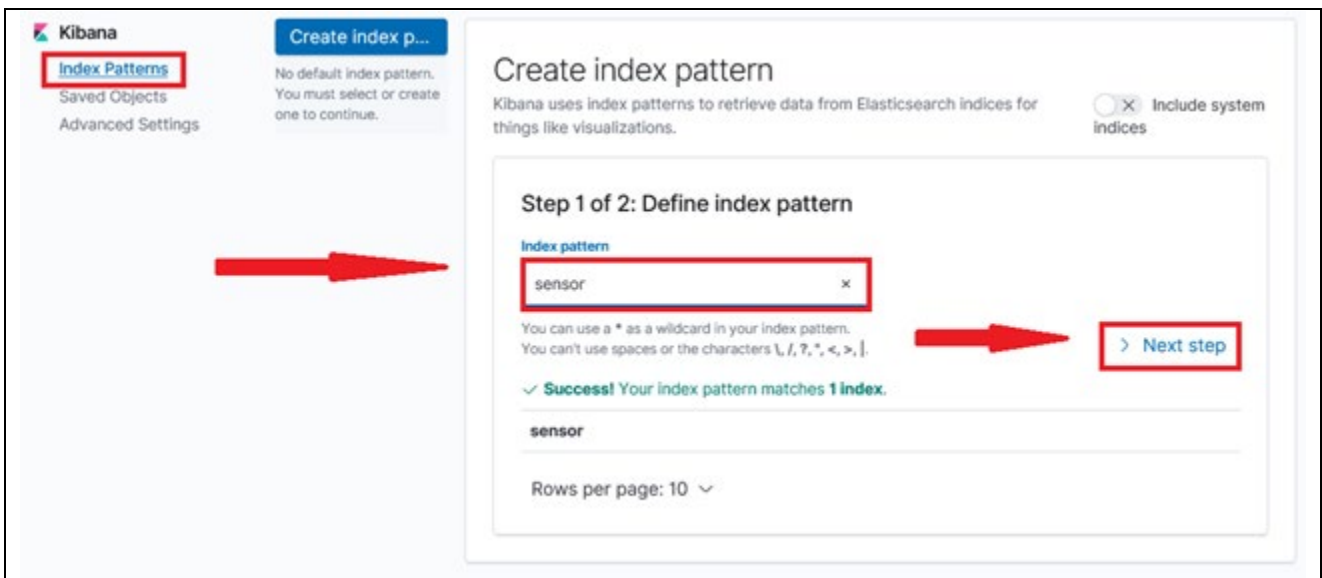


図 6-30 Define index pattern

3. Time Filter field name に datetime を選択してから、Create index pattern をクリック。

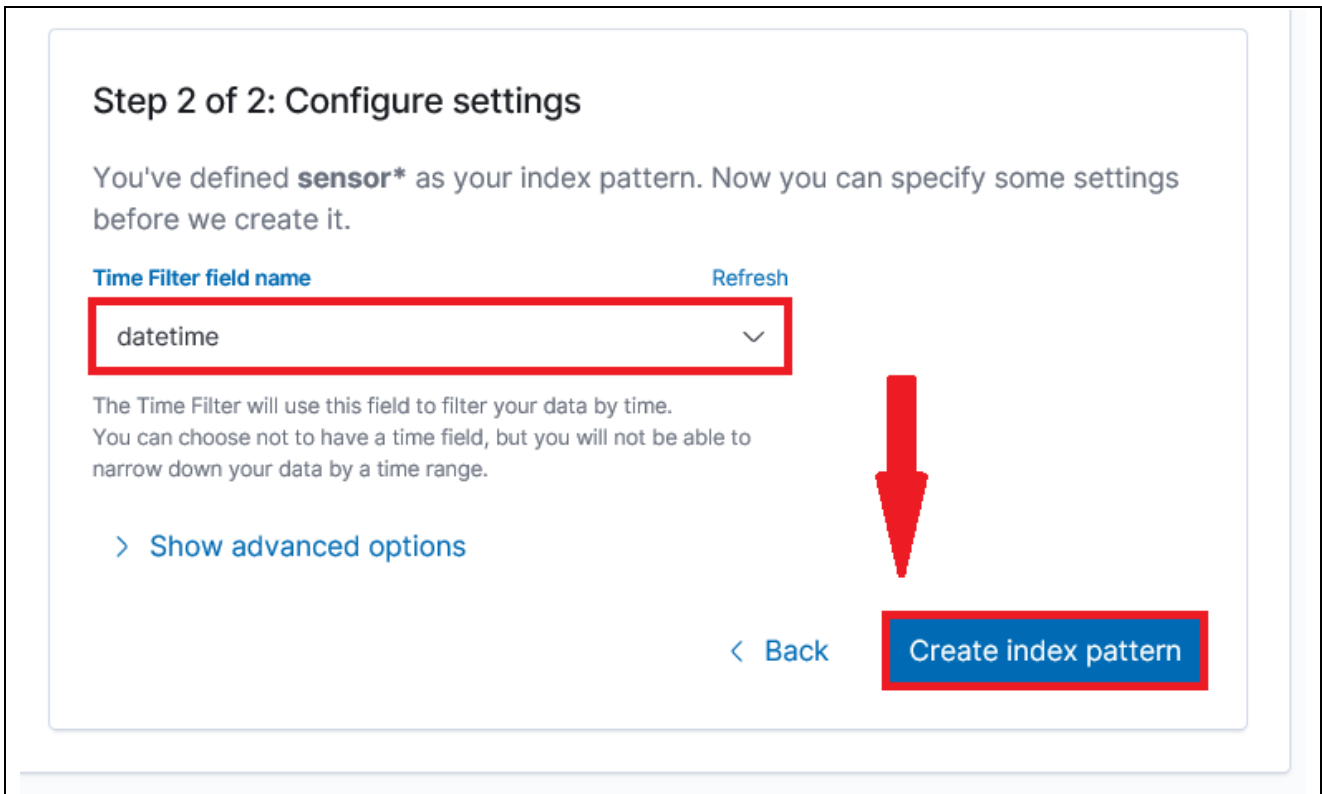


図 6-31 Configure settings

4. 左のメニューから Visualize をクリック。

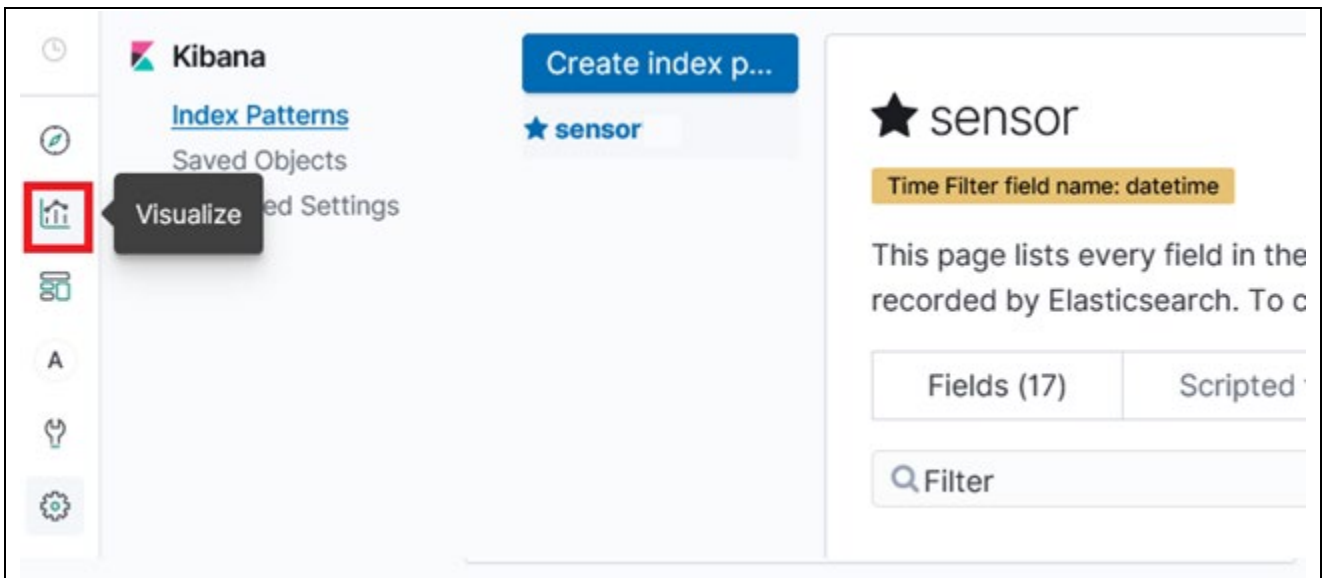


図 6-32 Visualize

5. Create a visualization をクリック。

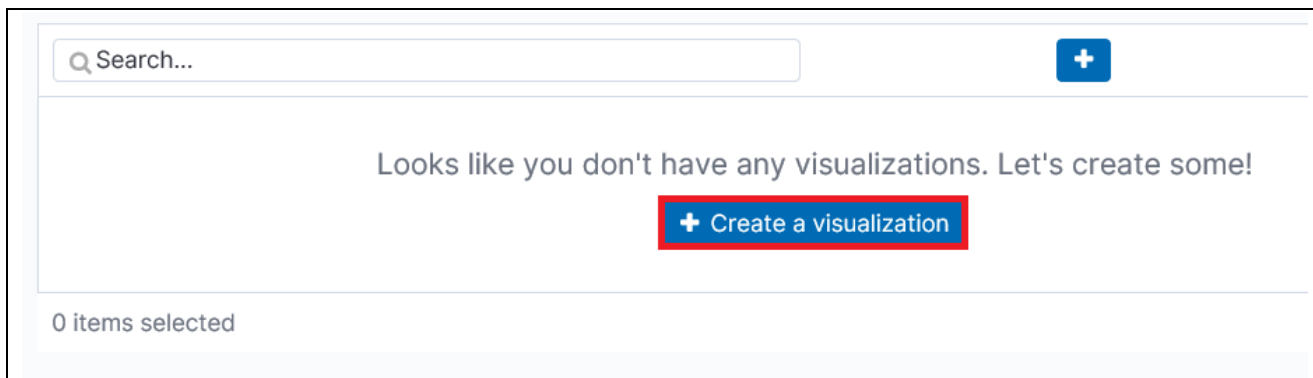


図 6-33 Create a visualization

6. Line をクリック。

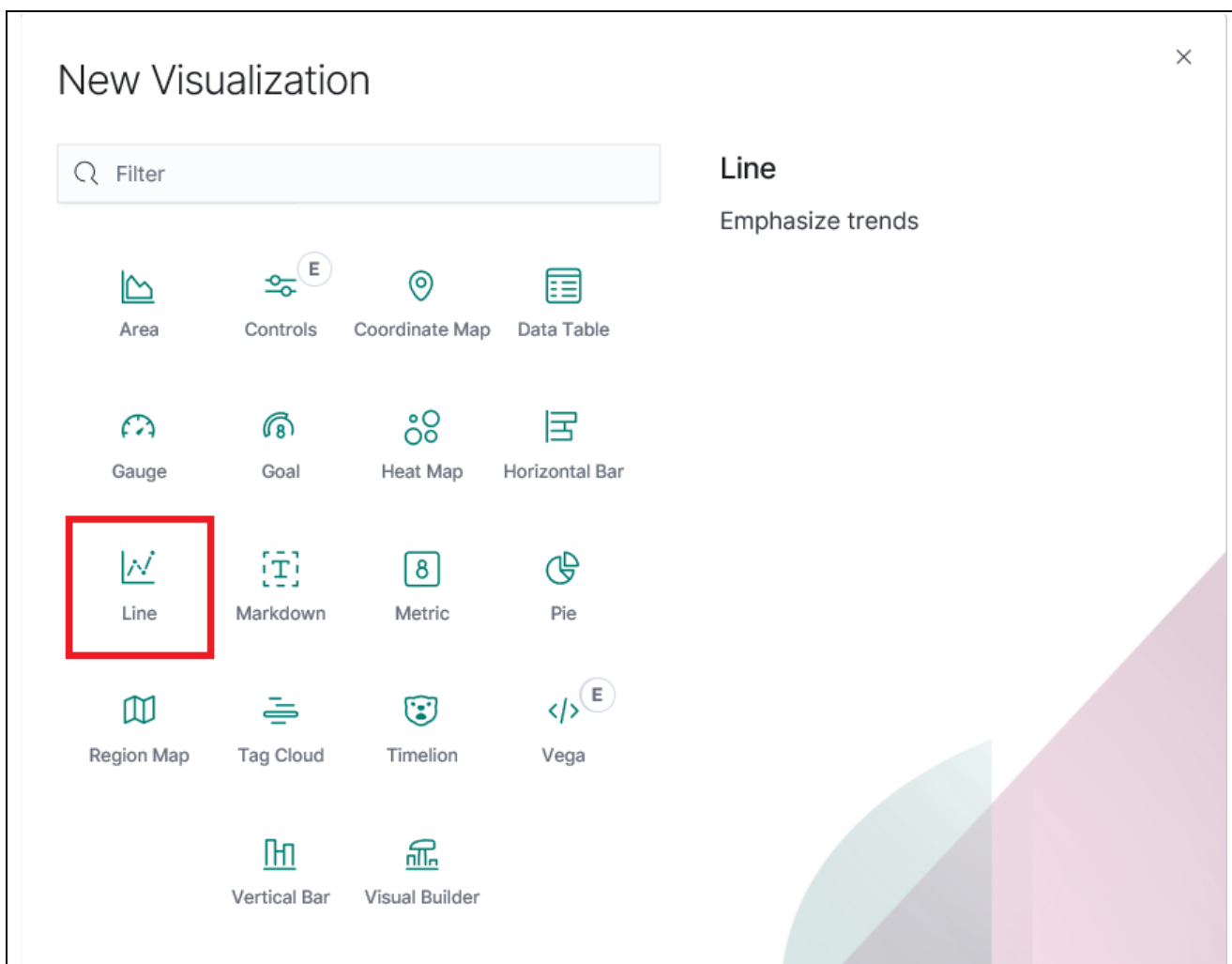


図 6-34 Line

7. sensor をクリック。

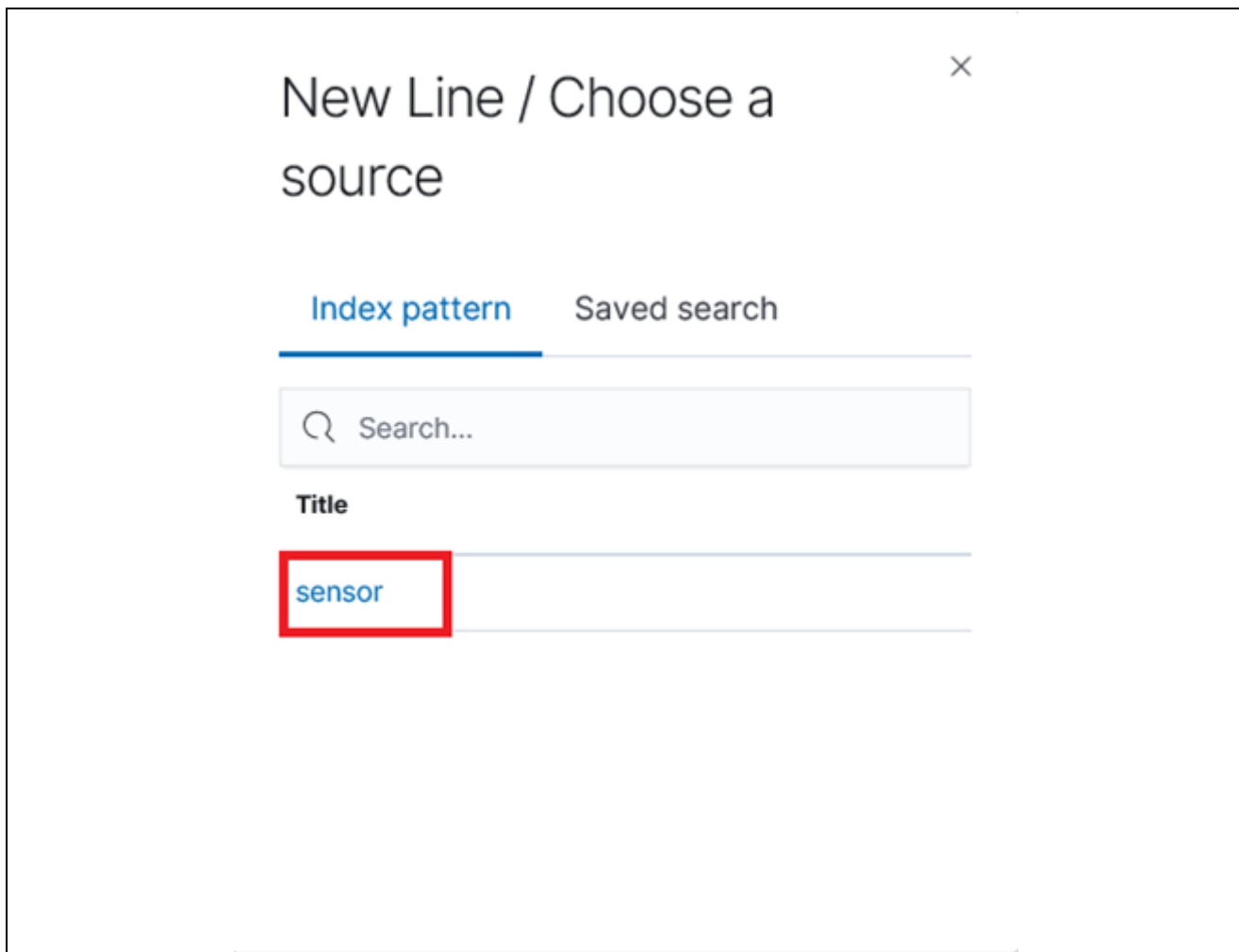


図 6-35 New Line / Choose a source

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。

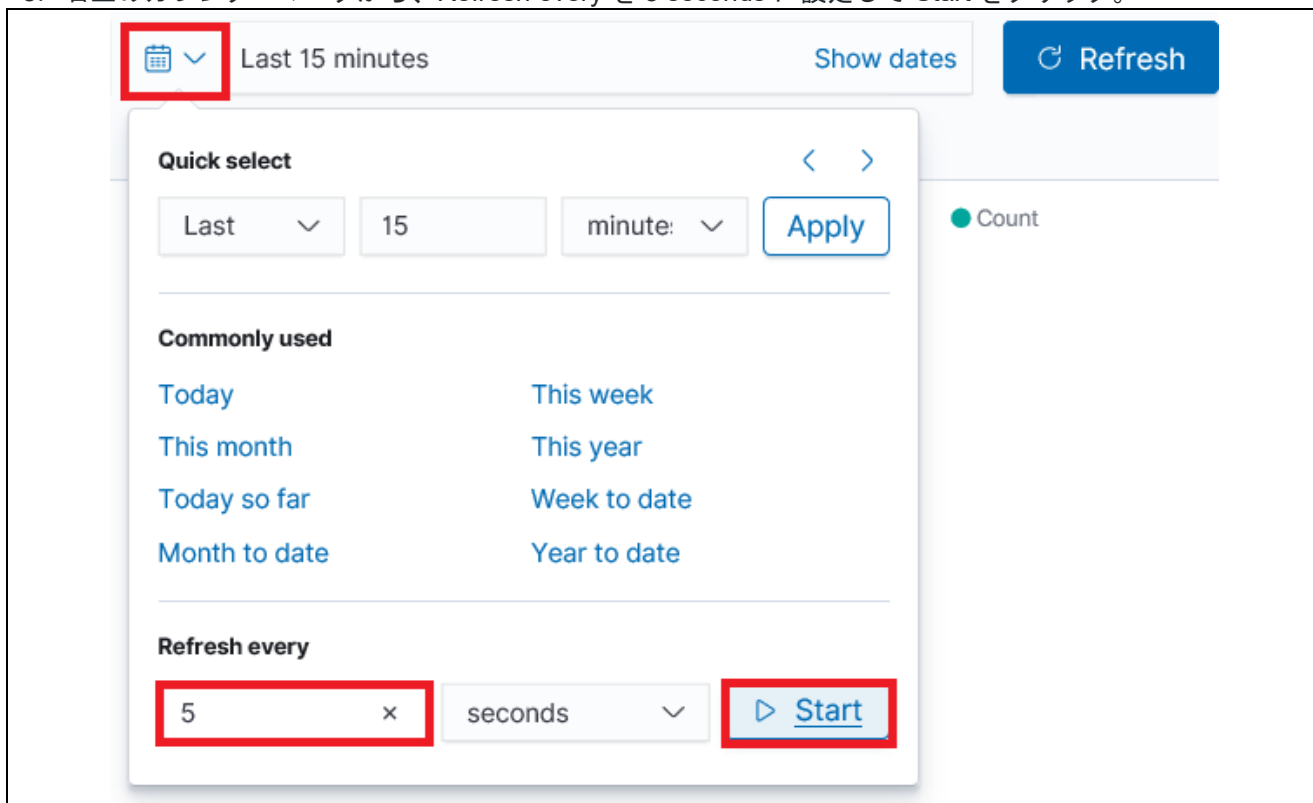


図 6-36 Refresh every の設定

9. Metrics で Y-Axis の Aggregation を Average、Field を light に設定する。

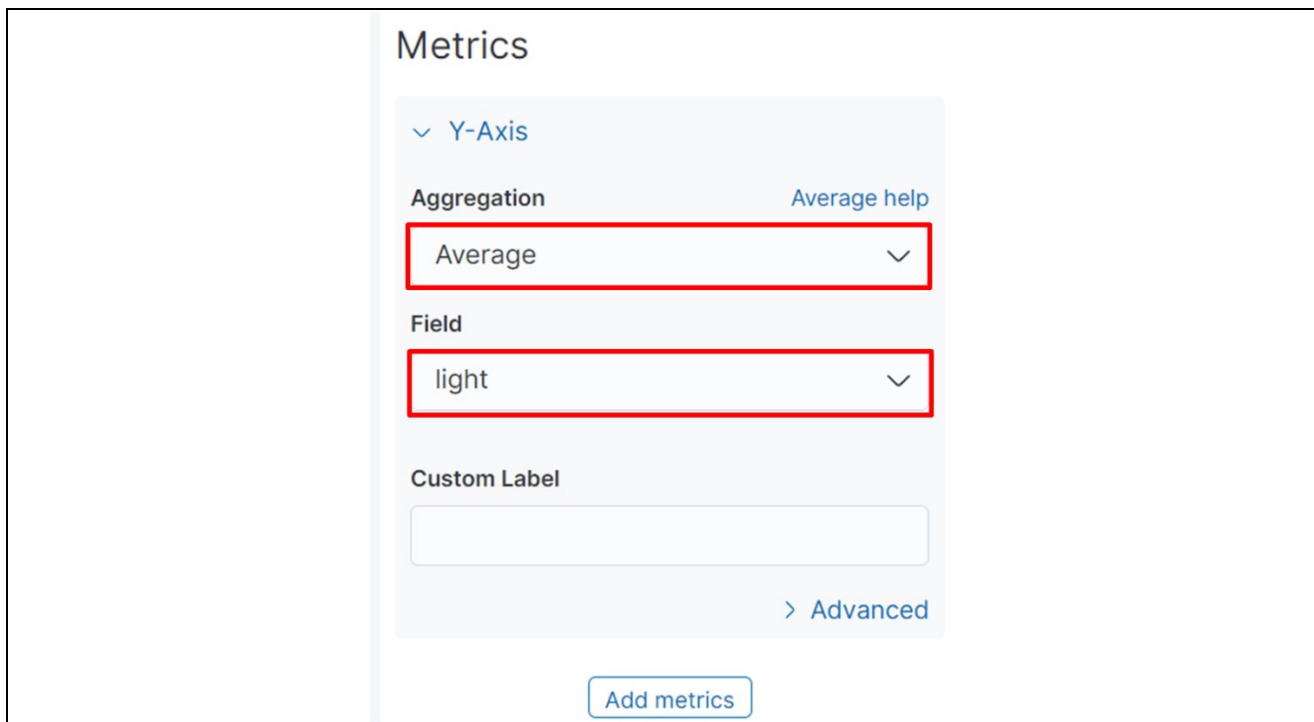


図 6-37 Metrics の設定

10. Buckets で X-Axis の Aggregation を Date Histogram、Field を datetime、Interval を Second に設定する。

The image shows a configuration interface for 'Buckets' under the 'X-Axis' section. The 'Aggregation' dropdown is set to 'Date Histogram', the 'Field' dropdown is set to 'datetime', and the 'Interval' dropdown is set to 'Second'. There is an unchecked checkbox for 'Drop partial buckets' and an empty 'Custom Label' field. An 'Add sub-buckets' button is located at the bottom of the panel.

図 6-38 X-Axis の設定

11. Apply changes をクリック。

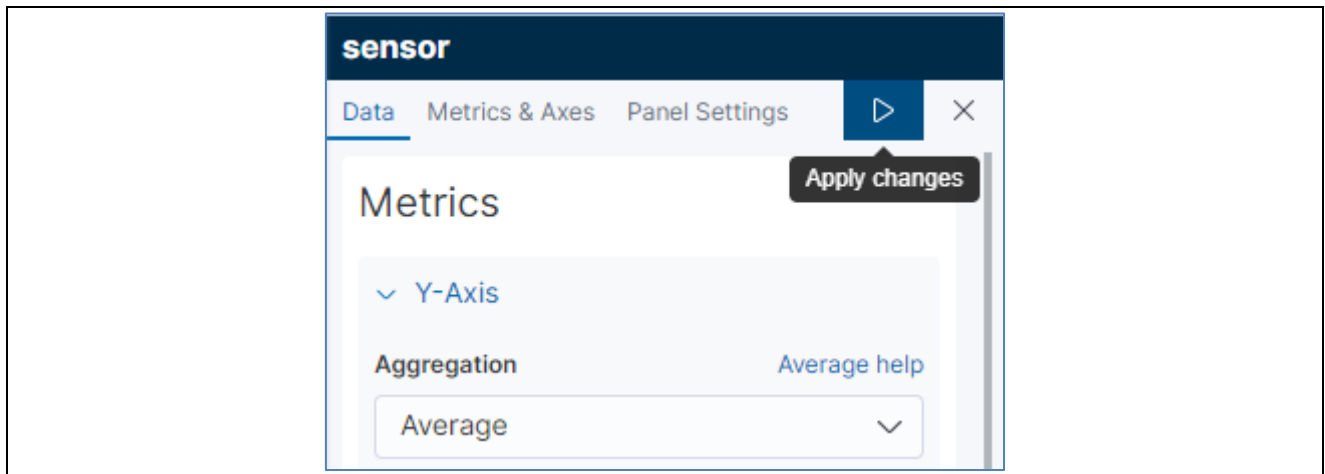


図 6-39 Apply changes

12. センサデータのグラフにて、明度の変化と共にグラフの値が変化することを確認する。

(ボードに手をかざすとグラフが下がり、ボードに光を当てるとグラフが上がります。)

以下に、光のセンサー情報を可視化したときの図を示します。

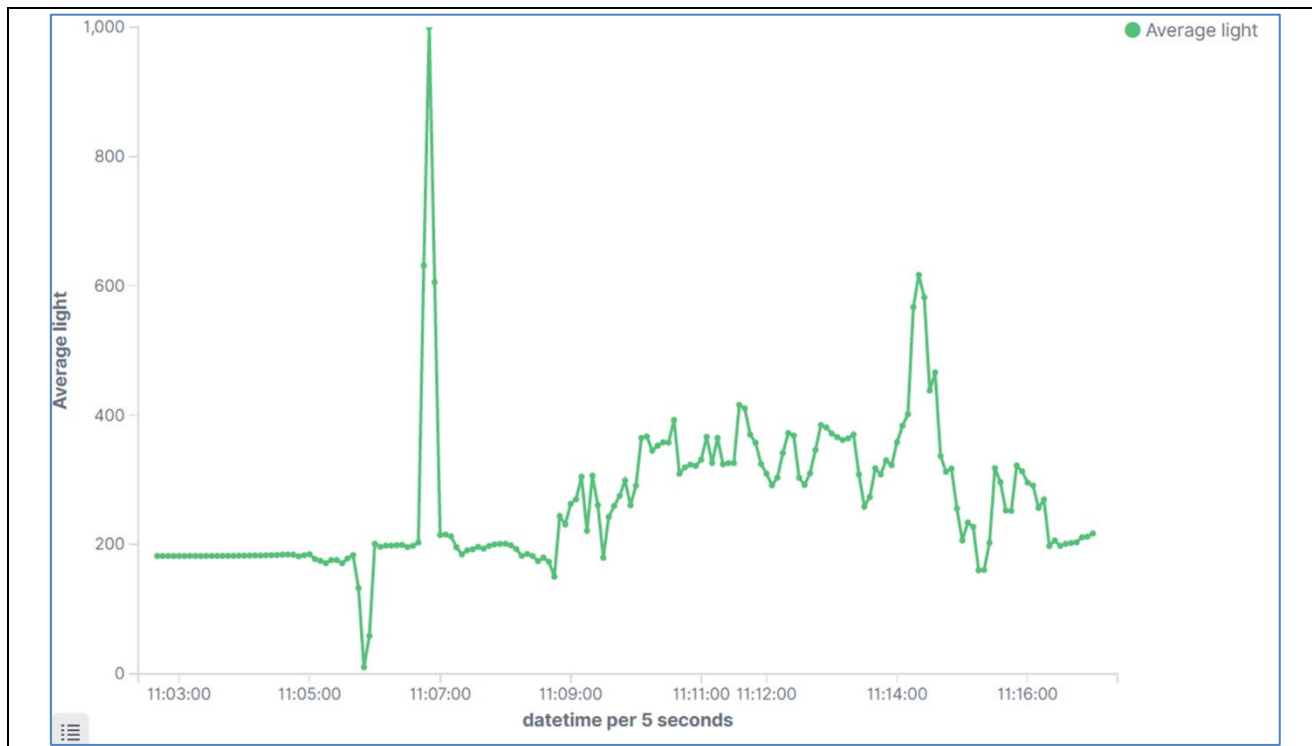


図 6-40 光センサー情報の可視化

6.6 デモプログラム実行後の注意事項について

Amazon Elasticsearch 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に、Elasticsearch のドメイン削除を忘れずに行ってください。



図 6-41 Elasticsearch のドメイン削除を忘れずに行う

7. 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	May.24.21	-	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しないでください。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、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ 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、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ 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、ノイズ放射量などが異なる場合があります。型名が異なる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、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。