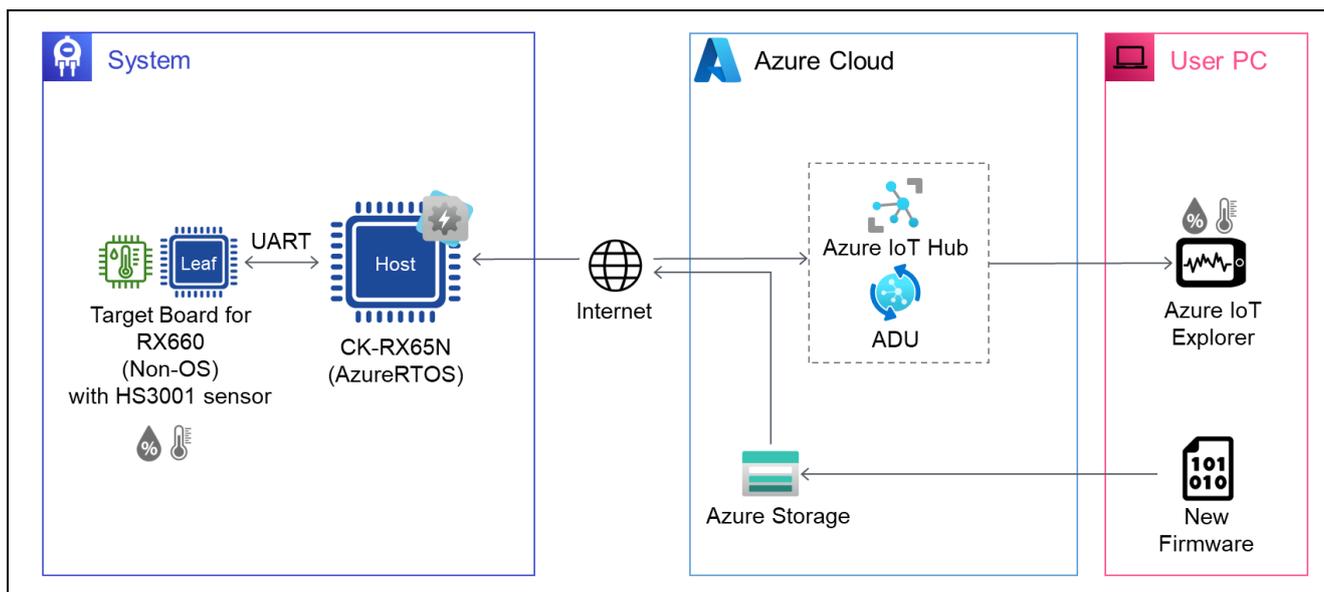


RX ファミリ

Microsoft Azure ADU を使ったセカンダリ OTA アップデート用サンプルコード

要旨

本アプリケーションノートでは、Microsoft Azure（以下、Azure と略します）との通信を行う RX65N を用いて、RX65N と UART で接続された 2 つ目のマイコンに対して、Azure の IoT サービスを利用した OTA ファームウェアアップデートを行うデモについて説明します。



動作確認デバイス

ホスト MCU : RX65N
 セカンダリ MCU : RX660
 温湿度センサ : HS3001

動作確認ボード

ホスト MCU : CK-RX65N (RTK5CK65N0S04000BE)
 セカンダリ MCU : Target Board for RX660 (RTK5RX6600C00000BJ)
 温湿度センサ : 温湿度センサ Pmod™ Board (US082-HS3001EVZ)

関連文書

[RX ファミリ Microsoft Azure サービスを利用した OTA の実現方法 \(R01AN6928\)](#)

[RX ファミリ ファームウェアアップデートモジュール Firmware Integration Technology \(R01AN6850\)](#)

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

[RX65N Group CK-RX65N v1 User's Manual \(R20UT5100\)](#)

[RX660 グループ Target Board for RX660 ユーザーズマニュアル \(R20UT5068\)](#)

目次

1. 概要	4
2. 動作確認条件	4
3. ハードウェア説明	5
4. ソフトウェア説明	6
4.1 サンプルプログラム構成	6
4.2 フォルダ/ファイル構成	7
4.3 コードサイズ	9
5. デモの動作説明	10
6. デモのセットアップ	10
6.1 ハードウェアのセットアップ	10
6.1.1 全体の構成	10
6.1.2 CK-RX65N のセットアップ方法	11
6.1.3 TB-RX660 のセットアップ方法	13
6.2 ソフトウェアのセットアップ	17
6.2.1 事前準備	17
6.2.2 ターミナルソフトの設定	17
6.2.3 CK-RX65N 用の初期ファームウェアの作成と実行	18
6.2.4 TB-RX660 用の初期ファームウェアの作成と実行	21
6.3 センサデータ可視化のためのセットアップ	23
6.3.1 Azure IoT Explorer のインストール	23
6.3.2 IoT Plug and Play Settings の設定	23
6.3.3 IoT Hub の設定	24
6.3.4 Telemetry データの受信開始	26
7. デモの実行手順	27
7.1 初期状態の動作確認	27
7.2 OTA アップデートの実行	29
7.2.1 更新ファームウェアの作成	29
7.2.1.1 CK-RX65N の更新ファームウェアの作成	29
7.2.1.2 TB-RX660 の更新ファームウェアの作成	29
7.2.2 マニフェストファイルの作成	30
7.2.3 更新のデプロイ	31
7.3 セカンダリ OTA アップデート実行中の動作確認	41
7.4 セカンダリ OTA アップデート後の動作確認	42
7.5 Azure クラウドリソースのクリーンアップ	43
8. 備考	45
8.1 使用するソフトウェアのライセンス情報	45
8.2 ファームウェアアップデートモジュール変更箇所	45
8.3 Leaf バージョン情報の制限事項	45

Microsoft、Azure は、米国 Microsoft Corporation.の米国およびその他の国における登録商標もしくは商標です。

Pmod、PmodUSBUART は Digilent Inc.の商標です。

すべての商標および登録商標は、それぞれの所有者に帰属します。

1. 概要

デモでは、Microsoft が提供する IoT 機器の OTA アップデートサービスである Azure Device Update (ADU) を利用し、Azure IoT Hub と Ethernet 経由でインターネット接続された RX65N マイコンおよび、そのマイコンと UART で接続された 2 つ目のマイコンのファームウェアの OTA アップデート（以下、セカンダリ OTA アップデートと呼称します）の動作を確認することができます。

ここで、本アプリケーションノートでは、Azure IoT Hub とインターネット接続されたマイコンを「Host」、Host に UART 接続された 2 つ目のマイコンを「Leaf」と呼称します。

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

2. 動作確認条件

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

表 2-1 デモ動作確認条件 (RX65N)

項目	内容
使用マイコン	RX65N (R5F565NEHDFB)
使用ボード	CK-RX65N (RTK5CK65N0S04000BE)
動作電圧	3.3V
RTOS	Azure RTOS 6.2.1_rel-rx-1.2.0
統合開発環境 (IDE)	e² studio 2023-07 QE for OTA v1.10
C コンパイラ	ルネサス製 RX ファミリ用 C/C++コンパイラ CC-RX v3.05.00 GCC for Renesas RX 8.3.0.202202
フラッシュ書き込みツール	Renesas Flash Programmer V3.12.00

表 2-2 デモ動作確認条件 (RX660)

項目	内容
使用マイコン	RX660 (R5F56609BDFFP)
使用ボード	Target Board for RX660 (RTK5RX6600C00000BJ)
動作電圧	3.3V
統合開発環境 (IDE)	e² studio 2023-07
C コンパイラ	ルネサス製 RX ファミリ用 C/C++コンパイラ CC-RX v3.05.00 GCC for Renesas RX 8.3.0.202202
フラッシュ書き込みツール	Renesas Flash Programmer V3.12.00
MOT ファイル変換ツール	Renesas Image generator (RX660 プロジェクトに付属)
USB-UART コンバータ	PmodUSBUART™

表 2-3 デモ動作確認条件 (センサ)

項目	内容
温湿度センサボード	US082-HS3001EVZ Board

表 2-4 デモ動作確認条件 (その他)

項目	バージョン
Python	3.10.4
Azure IoT Explorer (preview)	0.15.8

3. ハードウェア説明

本システムは、Host に RX65N を搭載した CK-RX65N、Leaf に RX660 を搭載した Target Board for RX660 (以下、TB-RX660 と略します)を使用します。

本デモでは、マイコン間の UART 通信を介して、センサが接続された Leaf のセカンダリ OTA アップデート([proxy updates](#))やセンサボードで取得したセンサデータのクラウドへのアップロードができます。

図 3-1 にシステム構成を示します。

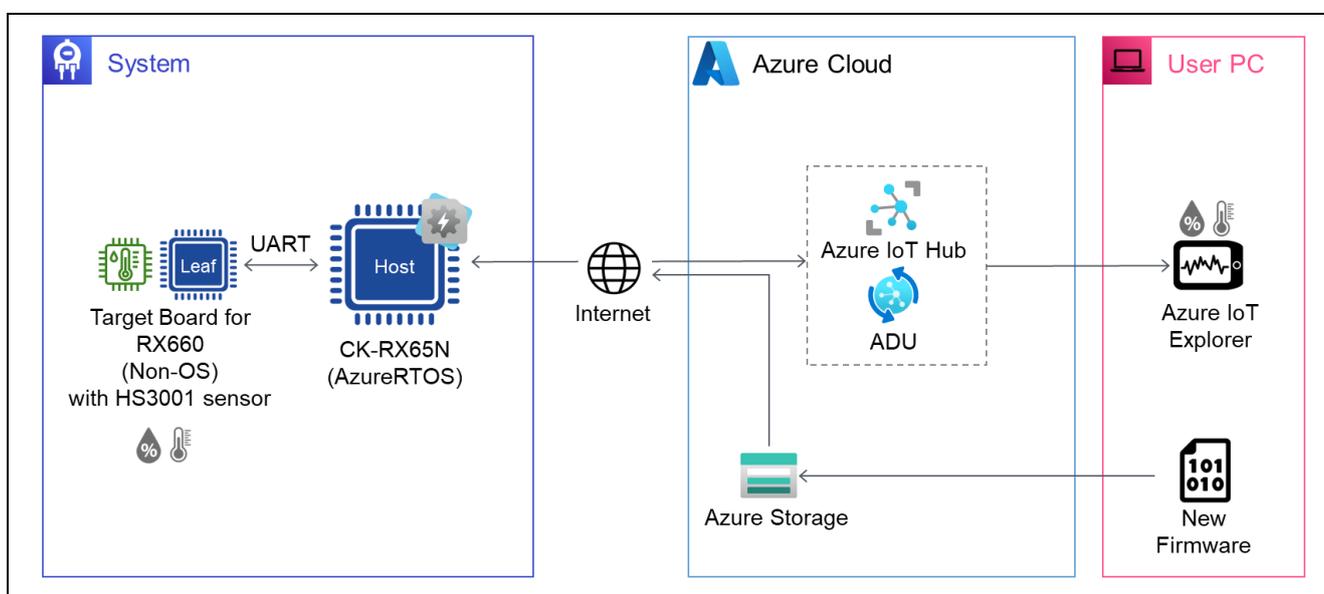


図 3-1 本デモのシステム構成

4. ソフトウェア説明

Host 側の RX65N のファームウェアには、Azure RTOS が実装されています。そのため、Microsoft が Azure クラウドで提供している IoT 向けサービスである Azure IoT Hub と Device Update for IoT Hub を利用して、OTA によるファームウェアアップデートや MQTT 通信によるクラウドへのデータのアップロードが実行可能です。

セカンダリ OTA アップデートの制御には、Host 側の RX65N マイコンでは [Azure ADU の proxy updates 機能](#) を利用し、Azure から受信した RX660 用の更新ファームウェアをさらに Leaf の RX660 マイコンに送信し、ファームウェア更新を実現しています。

Leaf 側の RX660 マイコンのファームウェアアップデートの制御は「[RX ファミリ ファームウェアアップデート モジュール Firmware Integration Technology Rev.2.00](#)」を使用します。なお、RX660 上でファームウェア アップデート モジュール Rev.2.00 を使用するために、ソースコード上のマクロ値に変更を加え、本デモの Happy path でのみ動作を確認しています。詳細は 8.2 章をご参照ください。

また、センサボードで取得したセンサデータのクラウドへのアップロード、およびセンサデータの可視化に関しては、「[RX65N グループ RX65N Cloud kit で Azure RTOS を用いてセンサデータを可視化する方法](#)」を参考に実現しています。

4.1 サンプルプログラム構成

本サンプルプログラムの Leaf のファームウェア更新の仕組みは、ファームウェアアップデートモジュールが提供している方式のうち、「リニアモードの半面更新方式」を使用しています。この方式の詳細は、「[RX ファミリ ファームウェアアップデート モジュール Firmware Integration Technology Rev.2.00](#)」の 1.3 章「各ファームウェアアップデート方式について」の「リニアモードの半面更新方式」をご参照ください。

以下に本サンプルプログラムのメモリマップを示します。

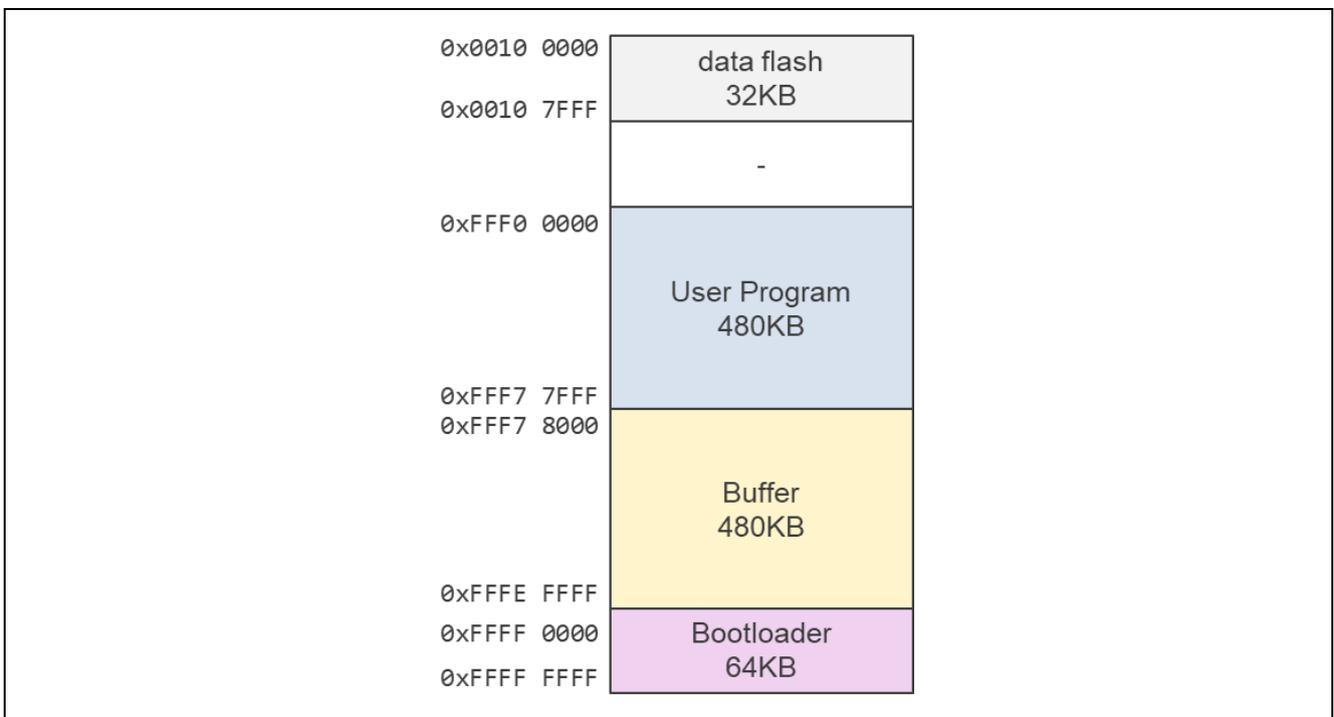


図 4-1 TB-RX660 サンプルプログラムのメモリマップ

4.2 フォルダ/ファイル構成

以下にサンプルプログラムのフォルダ構成を示します。

```
r01an6925xx0100-rx-communication
├─ Demo
│   ├── ccrx
│   │   ├── ck_rx65n_azure_2ndota_demo
│   │   ├── ck_rx65n_demo_bootloader
│   │   ├── rx660_tb_2ndota_demo
│   │   └── rx660_tb_demo_bootloader
│   └── gcc
│       ├── ck_rx65n_azure_2ndota_demo
│       ├── ck_rx65n_demo_bootloader
│       ├── rx660_tb_2ndota_demo
│       └── rx660_tb_demo_bootloader
├─ r01an6925ej0100-rx-communication.pdf
└─ r01an6925jj0100-rx-communication.pdf
```

ck_rx65n_demo_bootloader フォルダと ck_rx65n_azure_2ndota_demo フォルダには CK-RX65N 用のプロジェクトファイルが格納されています。

rx660_tb_demo_bootloader フォルダと rx660_tb_2ndota_demo フォルダには TB-RX660 用のプロジェクトファイルが格納されています。

以下に ck_rx65n_azure_2ndota_demo フォルダ(CC-RX)の主なファイル構成を示します。

```
ck_rx65n_azure_2ndota_demo
├─ .cproject
├─ .project
├─ ck_rx65n_azure_2ndota_demo HardwareDebug.launch
├─ ck_rx65n_azure_2ndota_demo.rcpc
├─ ck_rx65n_azure_2ndota_demo.scfg
├─ libs
├─ model
│   └─ thermostat-4.json
├─ src
│   ├── smc_gen
│   ├── main.c
│   ├── nx_azure_iot_adu_agent_proxy_simulator_driver.c
│   ├── nx_azure_iot_adu_agent_rx_driver.c
│   ├── sample_azure_iot_embedded_sdk_adu.c
│   └─ sample_config.h
└─ tools
    ├── AzureDeviceUpdateScripts
    └─ FlashProjects
```

また、以下に rx660_tb_2ndota_demo フォルダ(CC-RX)の主なファイル構成を示します。

```
rx660_tb_2ndota_demo
├─.cproject
├─.project
├─rx660_tb_2ndota_demo HardwareDebug.launch
├─rx660_tb_2ndota_demo.rcpc
├─rx660_tb_2ndota_demo.scfg
├─rfp
│   └─rx660_program.rpj
├─RenesasImageGenerator
│   ├──keys
│   └─image-gen.py
├─src
│   ├──base64
│   ├──cmdresp
│   ├──key
│   ├──sensor
│   ├──smc_gen
│   ├──tinycrypt
│   └─rx660_tb_2ndota_demo.c
```

4.3 コードサイズ

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

CC-RX

Compiler

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

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

Linker

最適化タイプ (-nooptimize/-optimize): All

Library Generator

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

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

表 4-1 コードサイズ (CC-RX)

プロジェクト	ROM	RAM
ck_rx65n_demo_bootloader	35KB	45KB
ck_rx65n_azure_2ndota_demo	570KB	224KB
rx660_tb_demo_bootloader	27KB	13KB
rx660_tb_2ndota_demo	49KB	22KB

GCC

最適化レベル: Optimize for debug (-Og)

表 4-2 コードサイズ (GCC)

プロジェクト	ROM	RAM
ck_rx65n_demo_bootloader	62KB	47KB
ck_rx65n_azure_2ndota_demo	652KB	234KB
rx660_tb_demo_bootloader	59KB	17KB
rx660_tb_2ndota_demo	91KB	37KB

5. デモの動作説明

- (1) デモの初期状態では TBRX660 は接続されている HS3001 センサを使って湿度データのみ取得します。
- (2) セカンダリ OTA アップデートの仕組みを用いて、Azure から CK-RX65N 経由で Leaf の TB-RX660 の更新ファームウェアをダウンロードし、ファームウェアの更新を行います。同時に Host の CK-RX65N のファームウェア更新も行います。
- (3) ファームウェア更新後では TB-RX660 は HS3001 センサから湿度データに加えて温度データも取得します。

一連の流れで、取得しているセンサデータの種類とその値は、両マイコンからの PC へのログ出力と Azure IoT Explorer から確認できます。

6. デモのセットアップ

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

CK-RX65N と TB-RX660 の配線や HS3001 センサの接続方法等のハードウェアのセットアップ、それぞれのマイコンボード用の初期ファームウェアの作成と書き込み等のソフトウェアのセットアップ、そして OTA アップデートの実行や Azure IoT Explorer でのセンサデータ確認のための Azure クラウド側の準備が必要です。

6.1 ハードウェアのセットアップ

6.1.1 全体の構成

最初に、本デモを構成するハードウェア全体の構成を示します。実際にセットアップ後の画像は図 6-12 を参照してください。以降でそれぞれのボードのセットアップ方法について詳しく説明します。

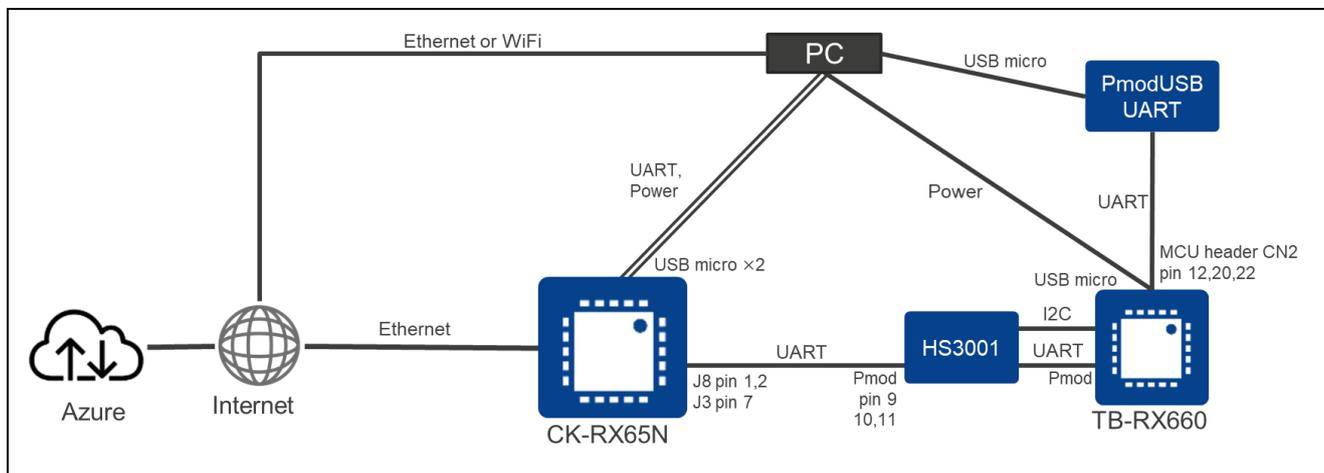


図 6-1 本デモのハードウェア全体構成

6.1.2 CK-RX65N のセットアップ方法

CK-RX65N のセットアップ方法を示します。

(1) TB-RX660 との UART 通信用ケーブル接続

TB-RX660 と UART 通信を行うための TXD, RXD, GND は CK-RX65N の J8, J3 コネクタの以下の端子に割り当てています。6.1.3(3)に示す TB-RX660 側の端子と以下表のように UART 信号の対応を取って接続してください。

表 6-1 マイコン間 UART 接続方法 (CK-RX65N ⇔ Pmod センサボード ⇔ TB-RX660)

CK-RX65N	(注 1)	HS3001 Pmod I/F		TB-RX660 Pmod I/F
J8 Pin 1: D0/RX (RXD7)	⇔	Pin 9: TXD (注 1)	⇔	Pin 9: TXD9
J8 Pin 2: D1/TX (TXD7)	⇔	Pin 10: RXD (注 1)	⇔	Pin 10: RXD9
J3 Pin 7: GND	⇔	Pin 11: GND	⇔	Pin 11: GND

注 1 : HS3001 センサボードの両端子は直結処理のため、TB-RX660 の Pmod I/F の入出力信号を扱うことができます。

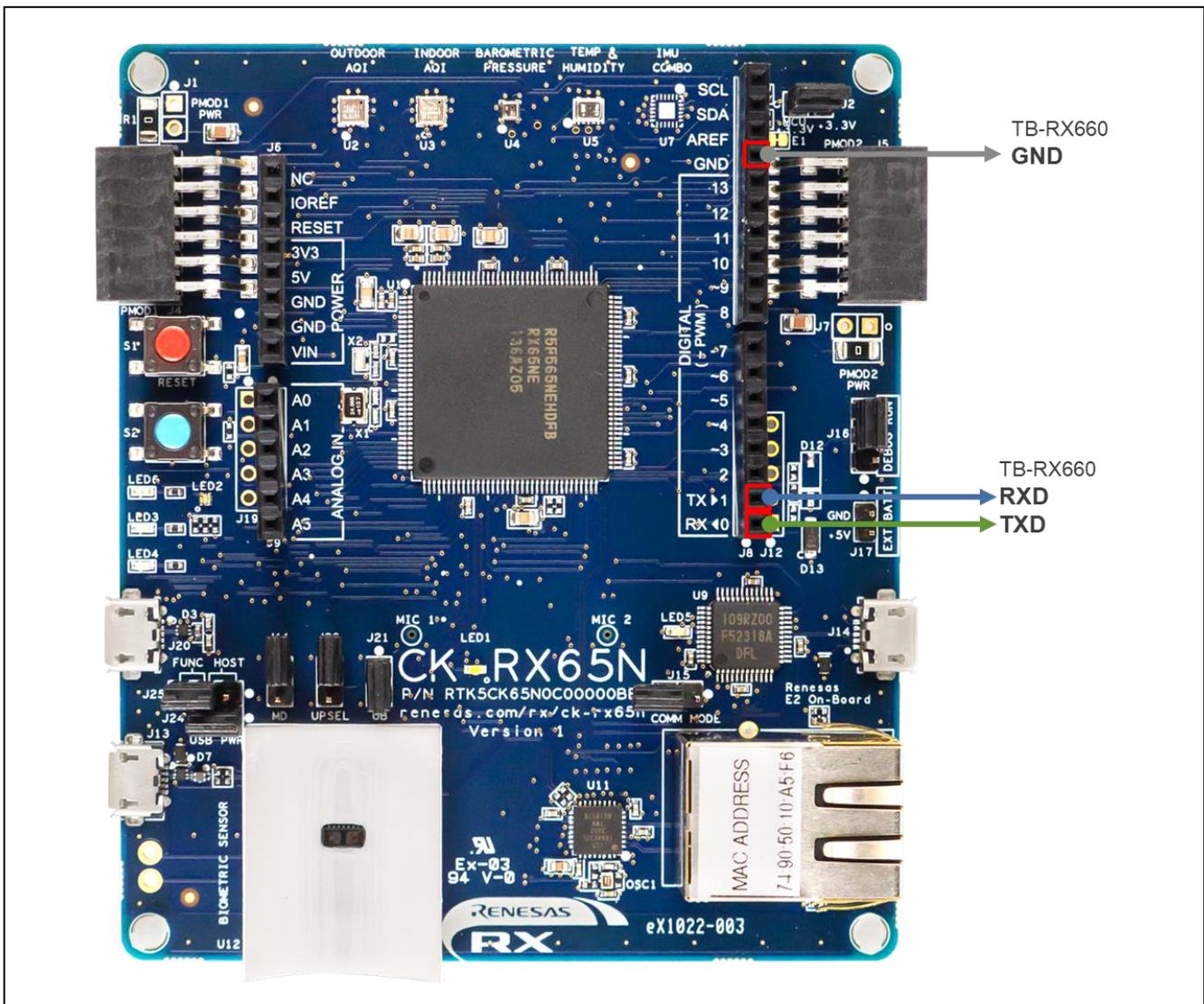


図 6-2 CK-RX65N のマイコン間 UART 通信に使用する端子位置

(2) PC へのログ出力用ケーブル接続

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

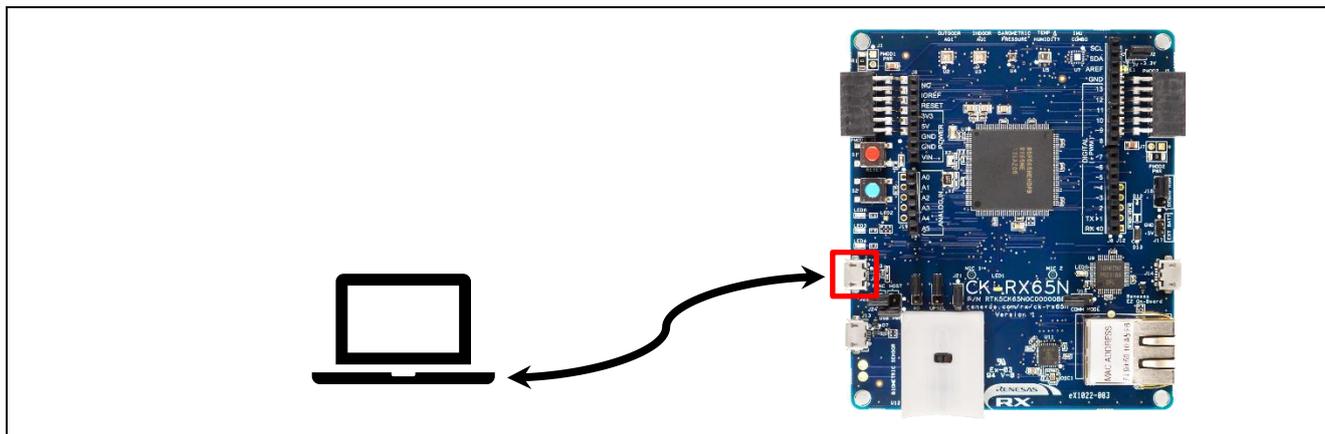


図 6-3 PC へのログ出力用の USB 接続

(3) 電源供給・デバッガとの接続

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

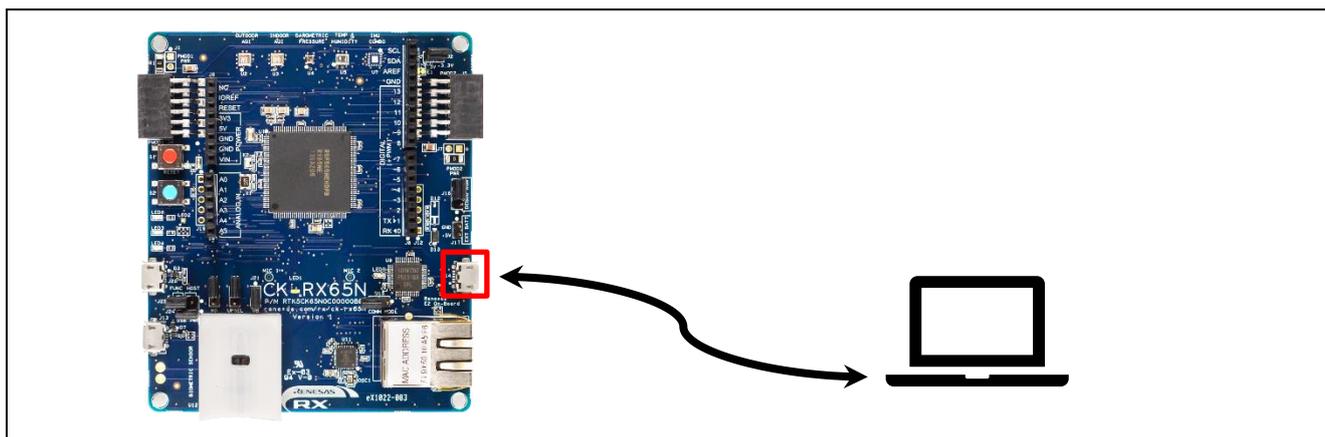


図 6-4 電源供給・エミュレータ用の USB 接続

(4) インターネット接続用の LAN ケーブル接続

CK-RX65N のイーサネットコネクタにインターネットに接続されている LAN ケーブルを接続します。

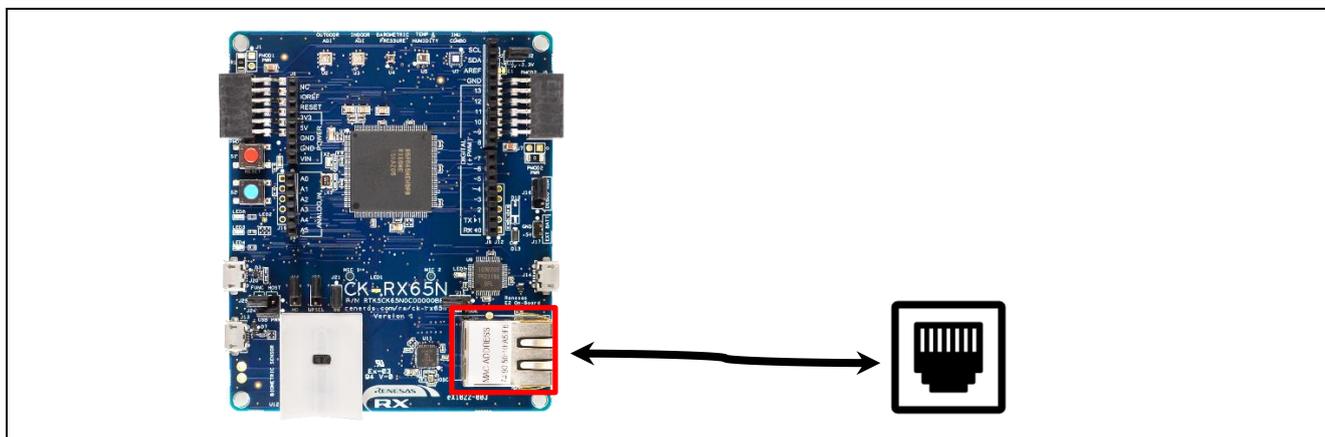


図 6-5 イーサネットによる有線インターネット接続

(5) ジャンパ J16 を DEBUG 側に短絡する

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

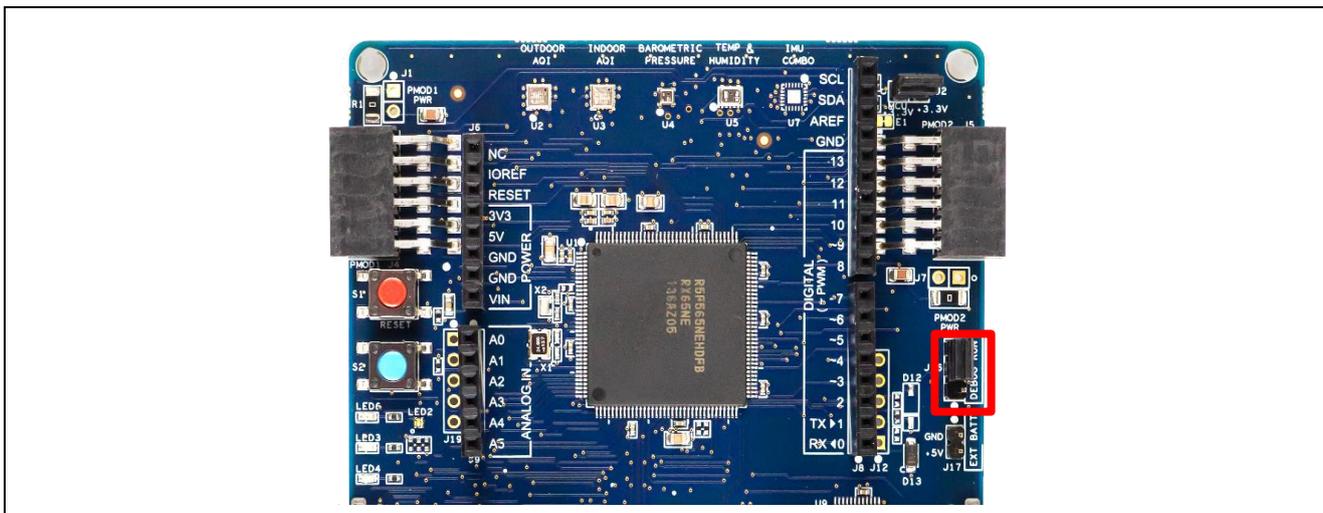


図 6-6 ジャンパ J16 の位置

6.1.3 TB-RX660 のセットアップ方法

TB-RX660 のセットアップ方法を示します。

(1) 事前準備

出荷状態のボードは、スルーホールにヘッダが取り付けられていないため、以下の事前準備をしてください。

- [TB-RX660 ユーザーズマニュアル](#)の「5.13 エミュレタリセットヘッダ」を参照し、ヘッダピンを付けてください。
- TB-RX660 ユーザーズマニュアルの「5.14 電源選択ヘッダ」を参照し、3.3V 電圧供給ができるようにしてください。本デモでは RX660 を 3.3V で動作させます。
- TB-RX660 に CN2 コネクタを付けてください。

(2) HS3001 ボードの接続

TB-RX660 の Pmod コネクタに HS3001 ボードを接続します。

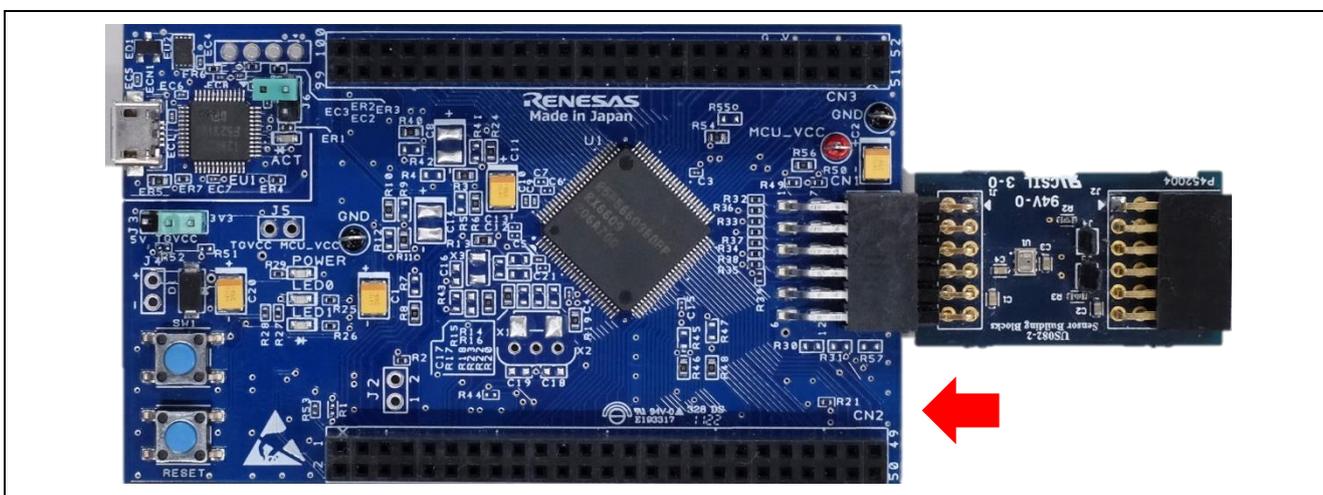


図 6-7 TB-RX660 の Pmod コネクタとセンサボードの接続

(3) CK-RX65N との UART 通信用ケーブル接続

CK-RX65N と UART 通信を行うための TXD, RXD, GND は TB-RX660 の Pmod コネクタの以下の端子に割り当てています。6.1.2(1)に示す CK-RX65N の端子と表 6-1 のように UART 信号の対応を取って接続してください。

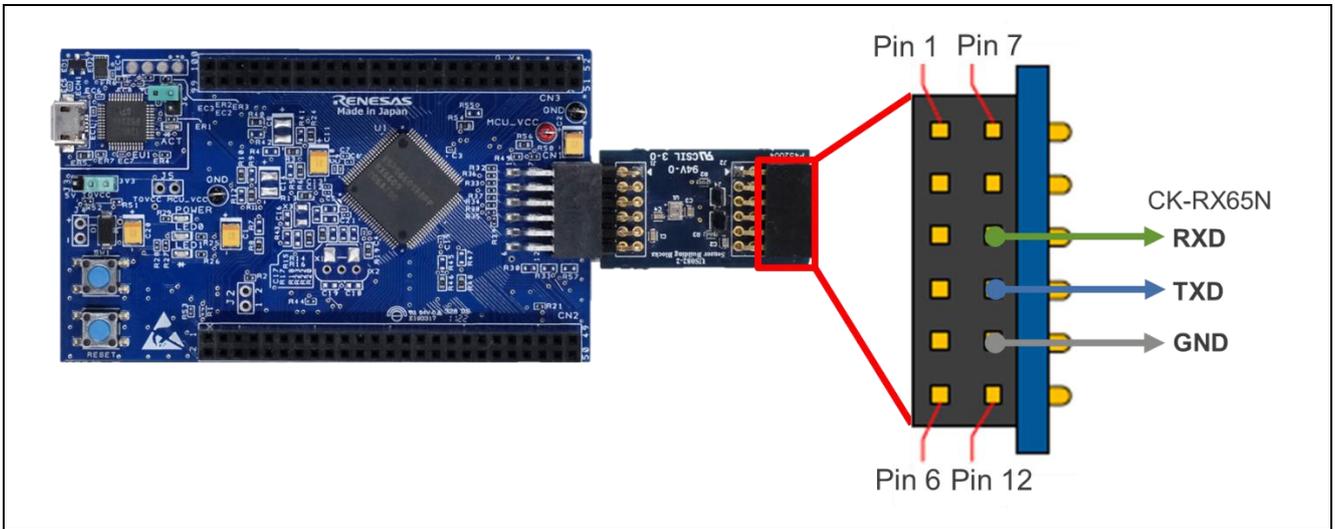


図 6-8 TB-RX660 のマイコン間 UART 通信に使用する端子位置

(4) PC へのログ出力用シリアル通信のケーブル接続

PC へシリアル接続でログ出力するための TXD, RXD, GND は、TB-RX660 の MCU ヘッダ CN2 の以下の端子に割り当てています。PmodUSBUART コンバータの端子と以下の表および図のように UART 信号の対応を取って接続してください。

PmodUSBUART のジャンパ JP1 は VCC-LCL 側を短絡してください。これにより、micro USB 側からの電源により UART I/F 電圧を 3.3V にします。

さらに、PC と PmodUSBUART コンバータの micro USB Type-B コネクタを USB ケーブルで接続します。

表 6-2 TB-RX660 と PmodUSBUART コンバータの接続方法

TB-RX660 MCU Header CN2		PmodUSBUART Pmod I/F
Pin 12: GND	↔	Pin 5: GND
Pin 20: RXD1 (MCU P30)	↔	Pin 3: TXD
Pin 22: TXD2 (MCU P26)	↔	Pin 2: RXD

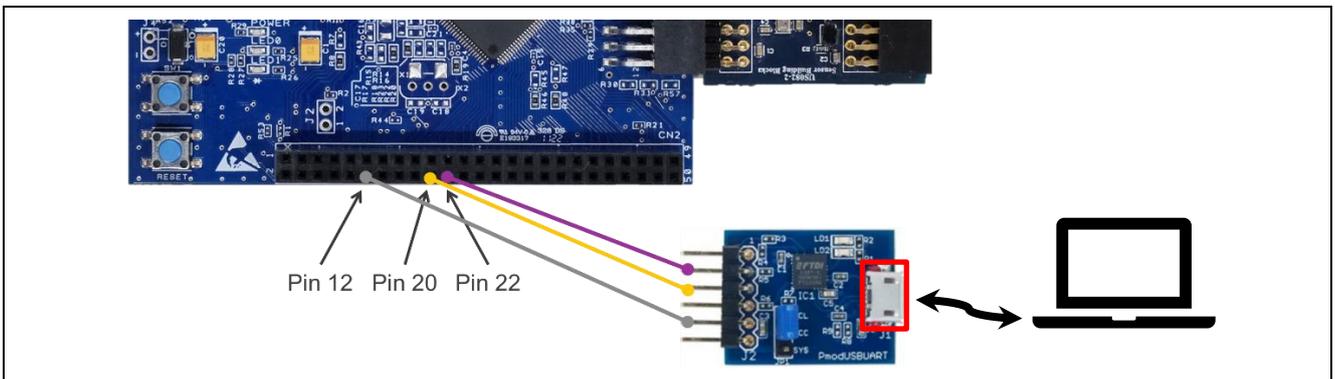


図 6-9 TB-RX660 の PC へのログ出力用のシリアル通信のケーブル接続方法

(5) 電源供給用ケーブル接続

PC と TB-RX660 の micro USB Type-B コネクタを USB ケーブルで接続します。

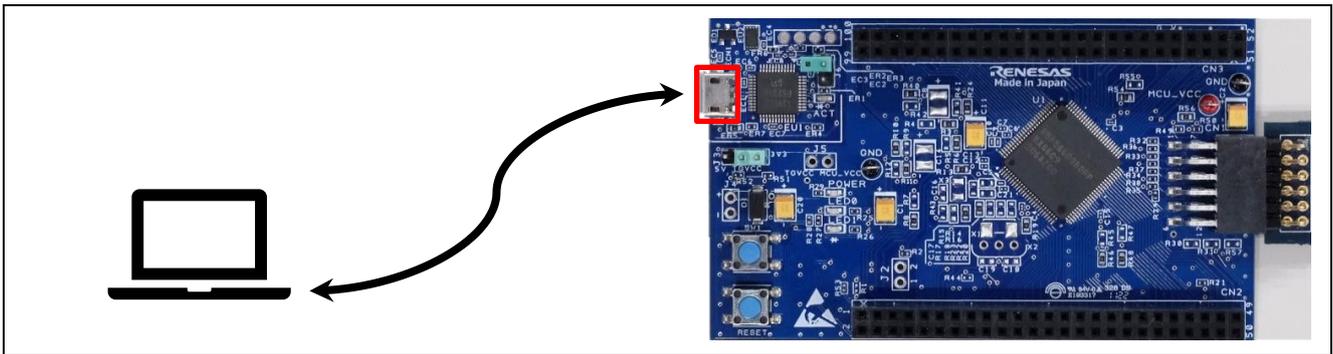


図 6-10 電源供給・エミュレータ用の USB 接続

(6) エミュレータリセットヘッダ(J6)を開放する

TB-RX660 のエミュレータリセットヘッダ(J6)を開放します。

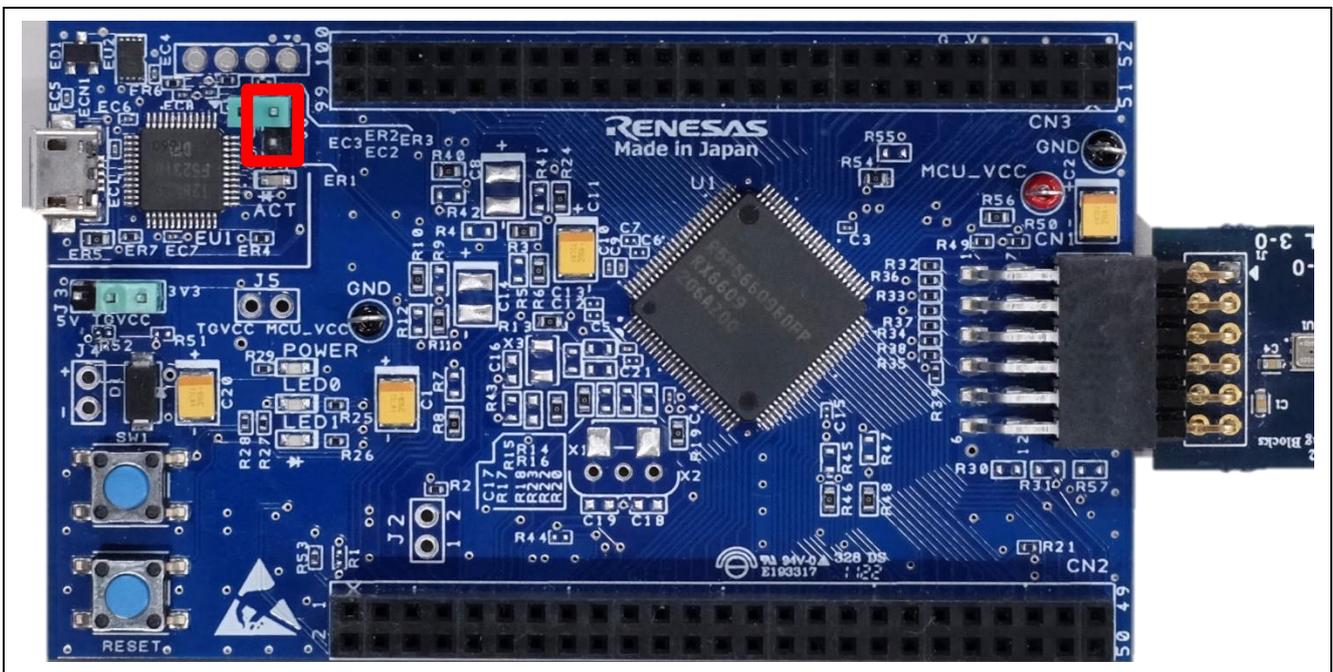


図 6-11 エミュレータリセットヘッダ(J6)の位置

以上でデモを実施するためのハードウェアのセットアップは完了です。図 6-12 にデモ構成の全体画像を示します。

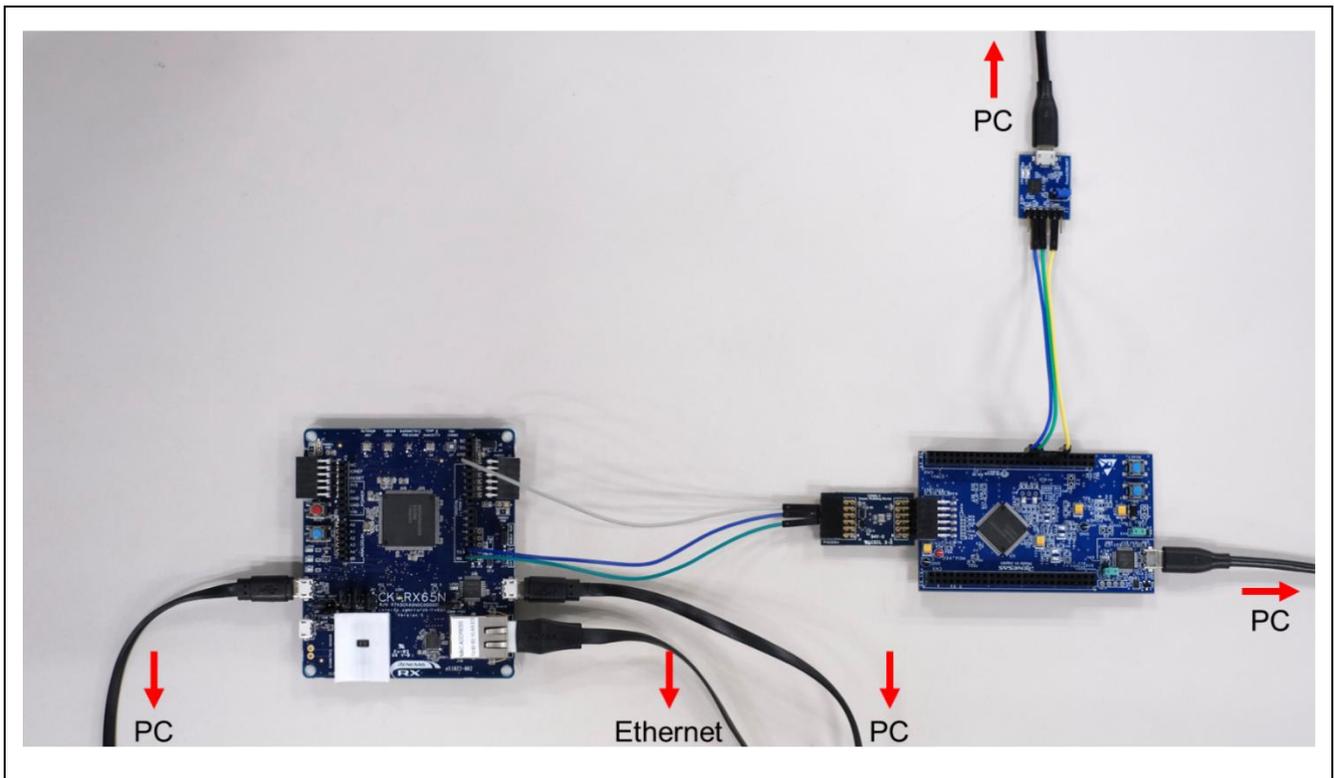


図 6-12 デモ構成の全体画像

6.2 ソフトウェアのセットアップ

以降の手順は、本アプリケーションノートで提供している CC-RX コンパイラ用のサンプルプロジェクトで実施しますが、GCC 用のプロジェクトでも手順は同様です。

6.2.1 事前準備

デモで使用するソフトウェアのインストールを行います。動作確認済みのそれぞれのソフトウェアのバージョンは表 2-4 をご参照ください。

(1) QE for OTA のインストール

e²studio のメニューバーから、[Renesas Views] → [Renesas QE]を開き、QE for OTA がインストールされているか確認します。OTA Main (QE), OTA Manage IoT Device (QE)があればインストール済みです。

無い場合は「[RX ファミリ AWS/Azure を利用したファームウェア更新ソフトの開発ガイド QE for OTA](#)」の「2.1 QE for OTA のインストール」章を参照し、QE for OTA をインストールしてください。

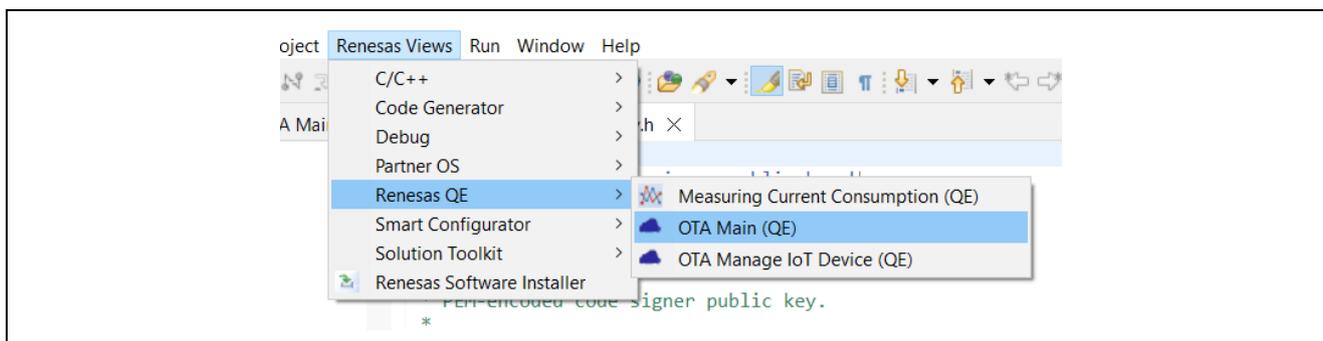


図 6-13 QE for OTA がインストールされているか確認

(2) Python 実行環境のインストール

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

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

```
> pip install pycryptodome
```

(3) Renesas Flash Programmer のインストール

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

6.2.2 ターミナルソフトの設定

シリアル通信を使ったログ出力を利用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 6-3 シリアルポート設定

項目	設定
ボーレート	115,200 bps
データ	8-bit
パリティ	なし
ストップ	1-bit
フロー制御	なし

6.2.3 CK-RX65N 用の初期ファームウェアの作成と実行

QE for OTA を使って CK-RX65N 用の初期ファームウェアを作成し、e² studio でデバッグ実行します。以下に手順を示します。

(1) プロジェクトのインポート

本アプリケーションノートでサンプルコードとして提供している、CK-RX65N 用のブートローダである **ck_rx65n_demo_bootloader** プロジェクトと、ユーザプログラムである **ck_rx65n_azure_2ndota_demo** プロジェクトを e² studio にインポートします。

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

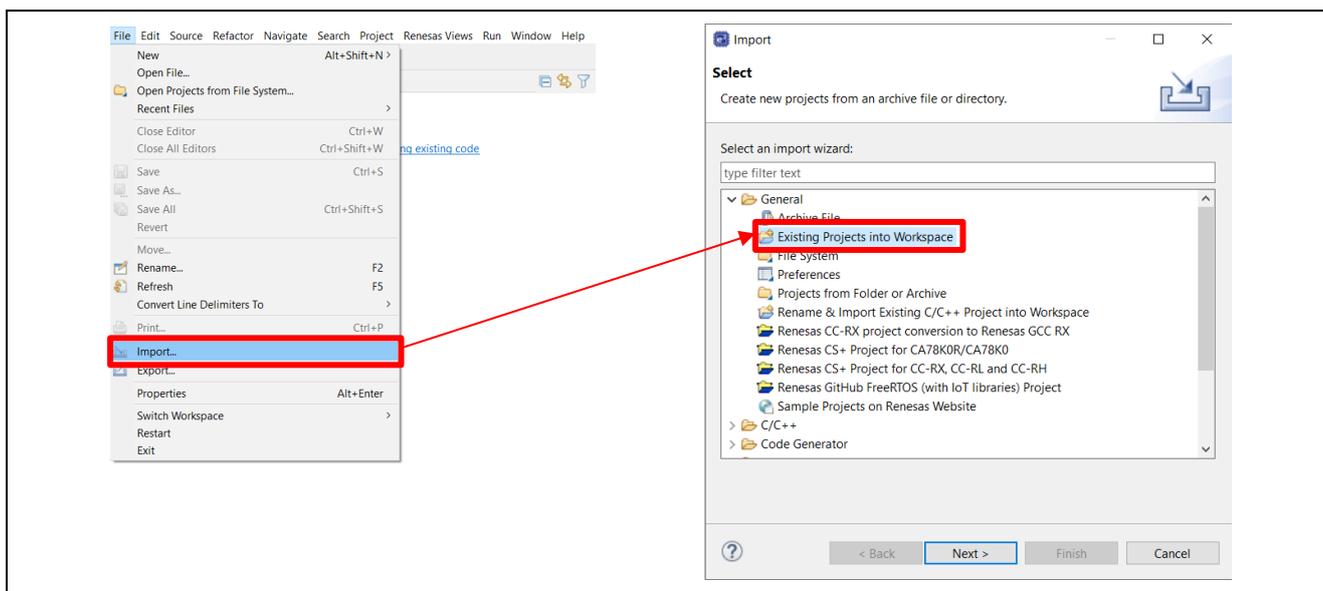


図 6-14 プロジェクトのインポート①

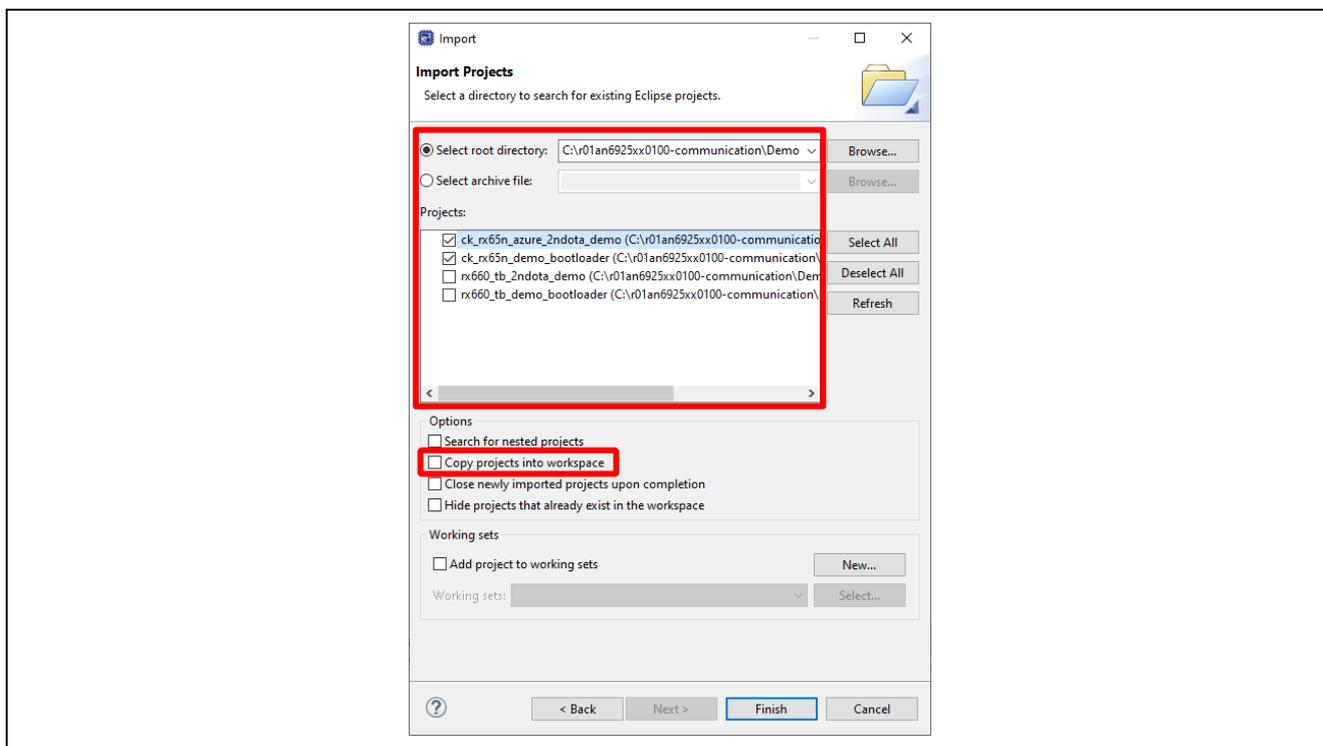


図 6-15 プロジェクトのインポート②

(2) QE for OTA 画面を開く

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

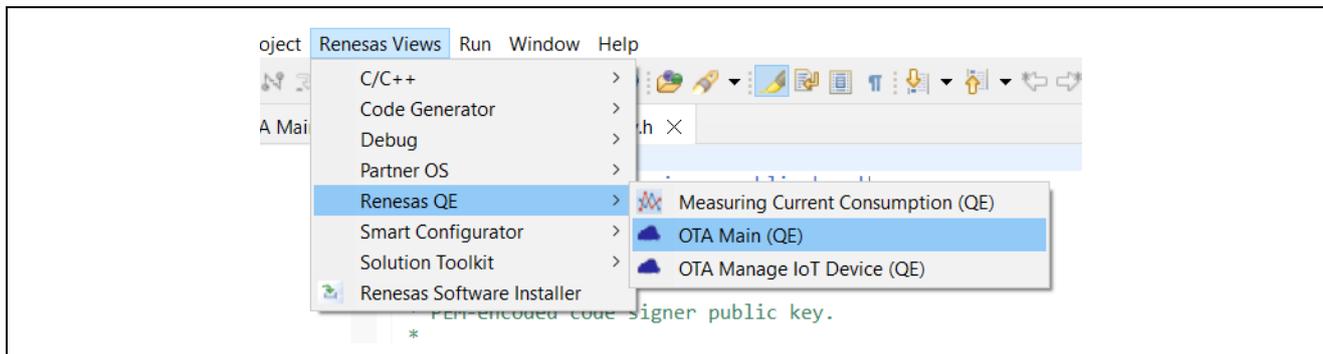


図 6-16 QE for OTA 画面を開く

(3) [QE for OTA] 1. Cloud Settings

「[RX ファミリ AWS/Azure を利用したファームウェア更新ソフトの開発ガイド QE for OTA](#)」の「4.3 Cloud の設定」章の Azure の手順を実行します。これにより Azure クラウド側の設定が完了します。

(4) [QE for OTA] 2. Prepare projects – Select projects

先ほど e² studio にインポートした ck_rx65n_demo_bootloader プロジェクトと ck_rx65n_azure_2ndota_demo プロジェクトを選択します。新たに OTA プロジェクトをインポートする必要はありません。

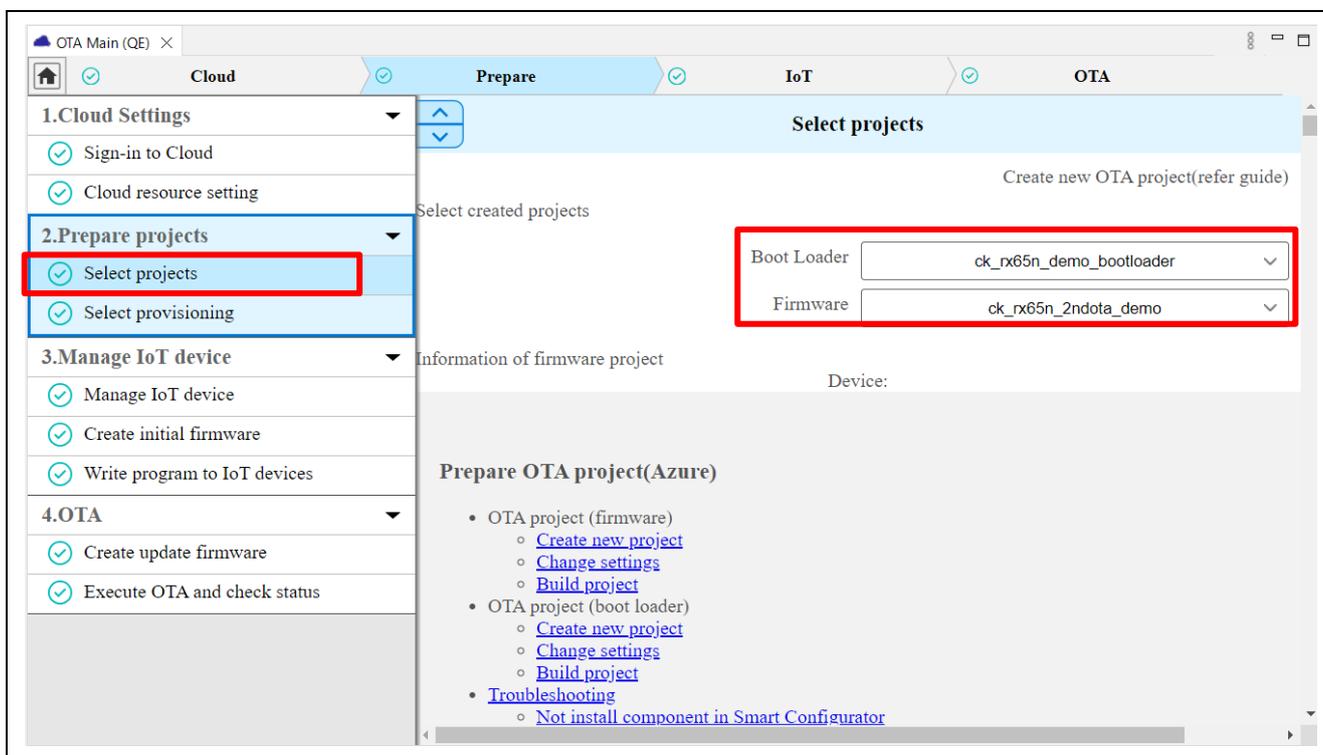


図 6-17 プロジェクトの選択

(5) [QE for OTA] 2. Prepare projects – Select provisioning

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



図 6-18 プロビジョニング方法の選択

(6) [QE for OTA] 3. Manage IoT device

「[RX ファミリ AWS/Azure を利用したファームウェア更新ソフトの開発ガイド QE for OTA](#)」の「4.5 IoT デバイスの管理」章の Azure の手順を実行します。これにより、Azure クラウドへのデバイスの登録と CK-RX65N の初期ファームウェアの書き込みが完了します。

(7) 動作確認

図 6-6 のジャンパ J16 を RUN 側(pin 2-3)に短絡します。

ターミナルソフトを起動し、図 6-19 のようにログが出力されていれば CK-RX65N の準備は完了です。

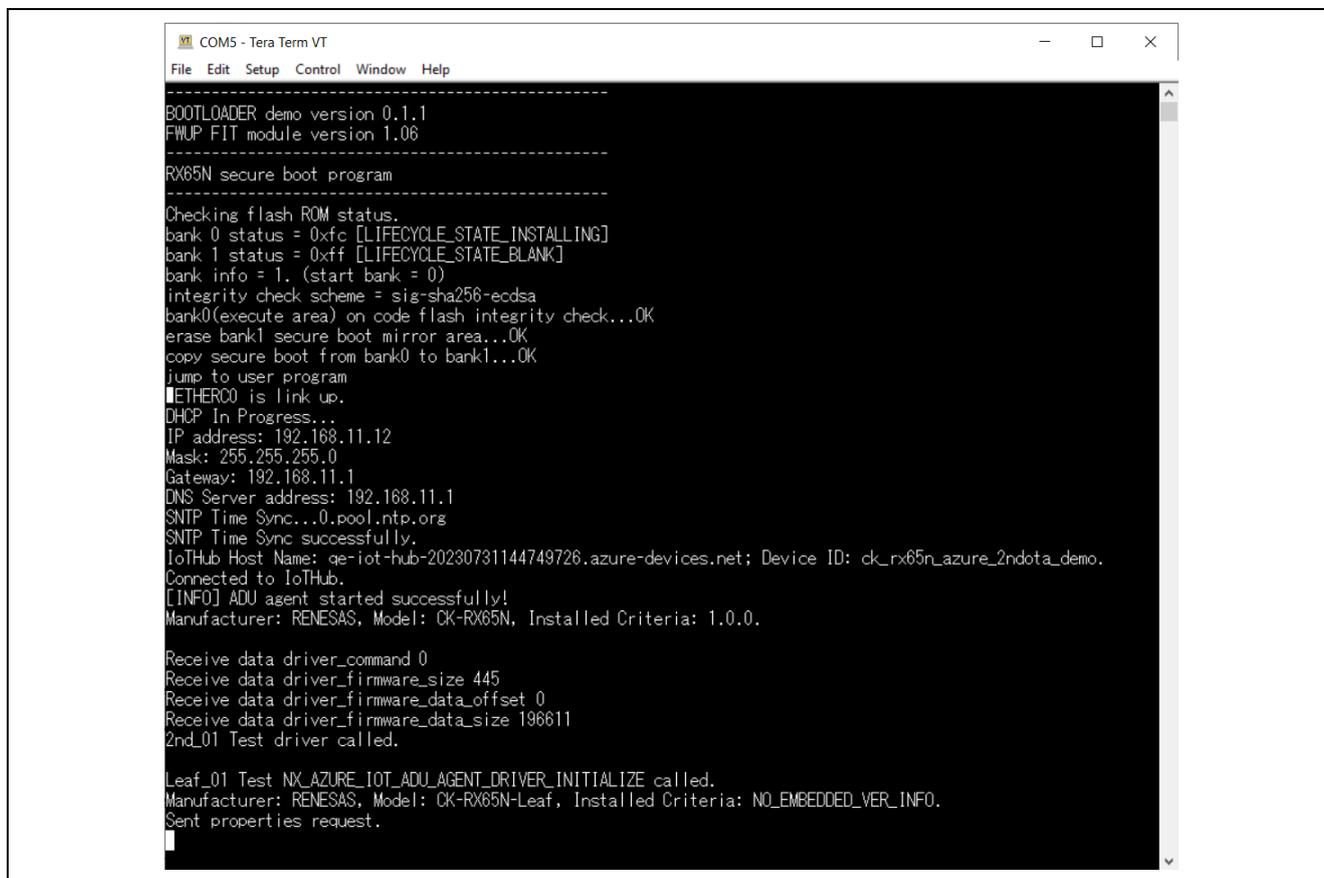


図 6-19 CK-RX65N のログ画面

6.2.4 TB-RX660 用の初期ファームウェアの作成と実行

TB-RX660 用の初期ファームウェアを作成し、Renesas Flash Programmer でマイコンに書き込みます。以下に手順を示します。

(1) プロジェクトのインポート

先ほどの CK-RX65N 用のプロジェクトと同様に、本アプリケーションノートでサンプルコードとして提供している、TB-RX660 用のブートローダである **rx660_tb_demo_bootloader** プロジェクトと、ユーザプログラムである **rx660_tb_2ndota_demo** プロジェクトを e² studio にインポートします。

(2) プロジェクトのビルド

rx660_tb_demo_bootloader プロジェクトと rx660_tb_2ndota_demo プロジェクトをビルドし、MOT ファイルを作成します。MOT ファイルはプロジェクトフォルダ直下の HardwareDebug フォルダに作成されません。

(3) 初期ファームウェアの作成

作成された rx660_tb_demo_bootloader と rx660_tb_2ndota_demo の MOT ファイルを結合して TB-RX660 用の初期ファームウェアを作成します。MOT ファイルの結合には Renesas Image Generator を使用します。Renesas Image Generator は「[RX ファミリ ファームウェアアップデート モジュール Firmware Integration Technology Rev.2.00](#)」に付属するツールです。詳細については、上記リンク先のアプリケーションノートの「Renesas Image Generator」章をご参照ください。

rx660_tb_2ndota_demo¥RenesasImageGenerator フォルダで以下のコマンドを実行し、初期ファームウェア initial_firm.mot を作成します。

```
> python .¥image-gen.py -iup ..¥HardwareDebug¥rx660_tb_2ndota_demo.mot -  
ibp ..¥..¥rx660_tb_demo_bootloader¥HardwareDebug¥rx660_tb_demo_bootloader.mot -  
o .¥initial_firm -key .¥keys¥secp256r1.privatekey -  
ip .¥RX660_Linear_Half_ImageGenerator_PRM_2ndota_demo.csv -vt ecdsa
```

(4) 初期ファームウェアの書き込み

Renesas Flash Programmer を用いて、上で作成した初期ファームウェア initial_firm.mot を TB-RX660 に書き込みます。

rx660_tb_2ndota_demo¥rfp フォルダ内の Renesas Flash Programmer プロジェクト rx660_program.rpj を開き、Program File に先ほど作成した RX660 用の初期ファームウェア initial_firm.mot を指定して、Start をクリックしてください。

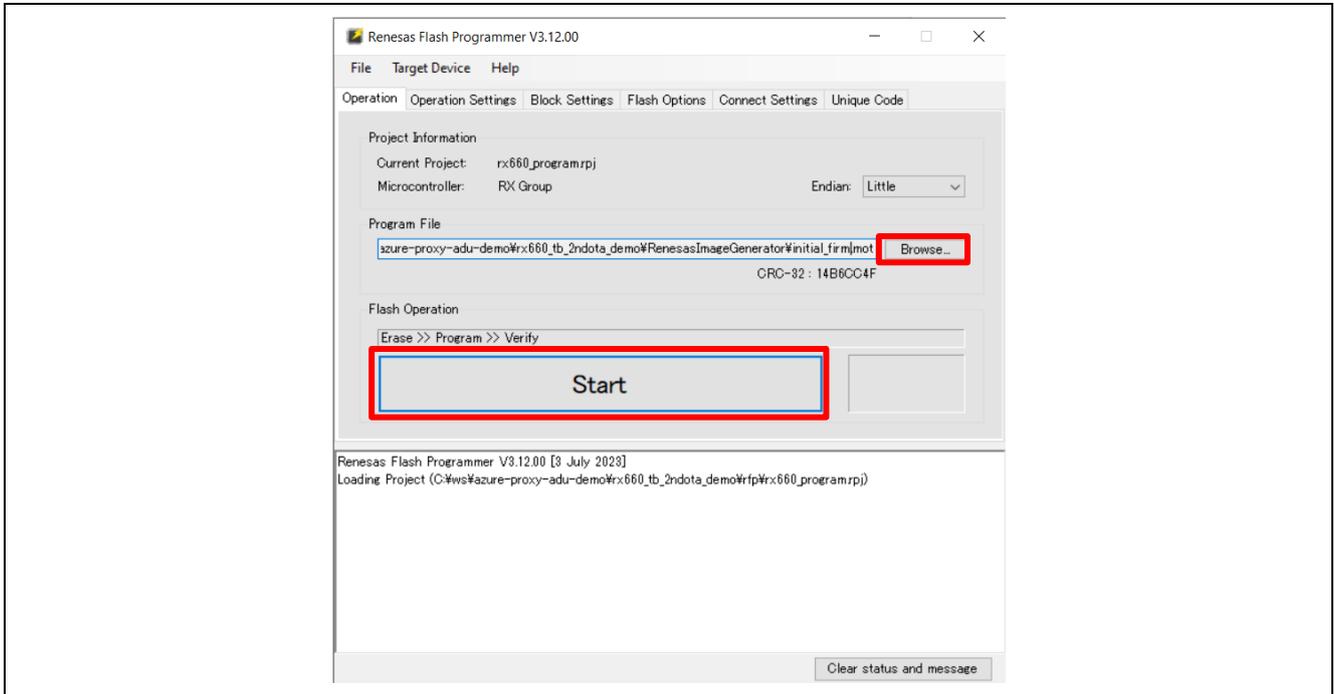


図 6-20 Renesas Flash Programmer での初期ファームウェアの書き込み

(5) 動作確認

TB-RX660 のエミュレータリセットヘッダ(J6)を短絡します。

ターミナルソフトを起動し、図 6-21 のように湿度の値がログ出力されていれば TB-RX660 の準備は完了です。

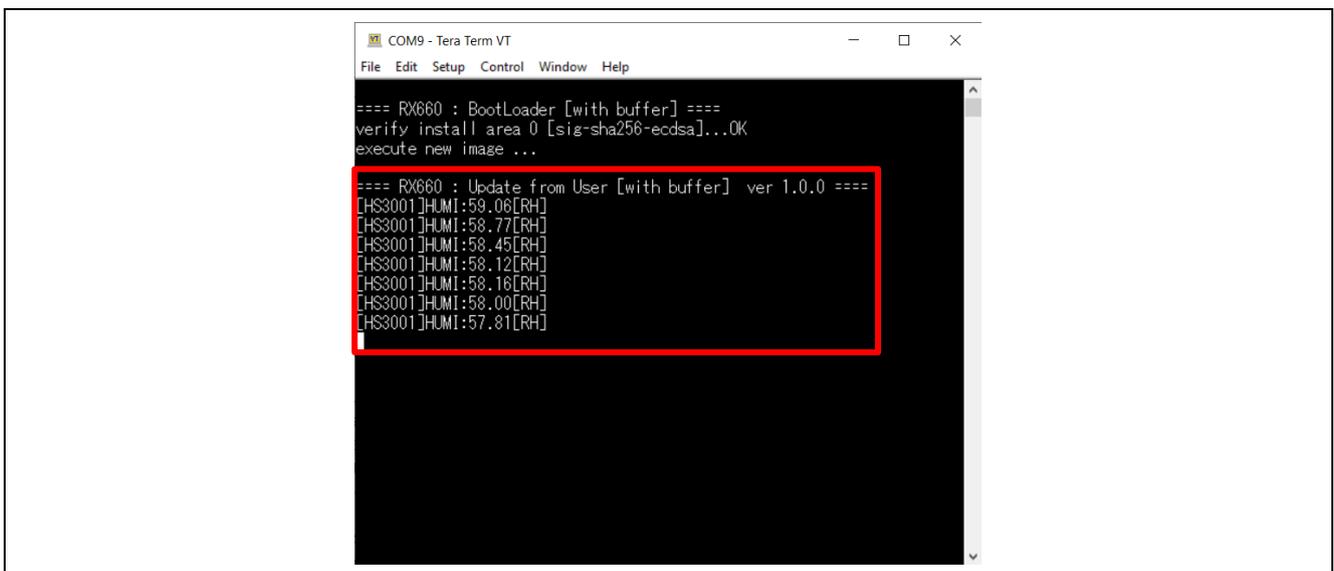


図 6-21 TB-RX660 のログ画面

6.3 センサデータ可視化のためのセットアップ

6.3.1 Azure IoT Explorer のインストール

Azure IoT Explorer のインストール方法および使用方法については下記リンクを参照してください。

[Azure IoT エクスプローラーをインストールして使用する - Azure IoT | Microsoft Learn](#)

本アプリケーションノートでは Host が送信するセンサデータの確認方法のみを記載しています。

Azure IoT Explorer は下記リンクから入手して下さい。

[Releases · Azure/azure-iot-explorer \(github.com\)](#)

6.3.2 IoT Plug and Play Settings の設定

Azure IoT Explorer を起動します。

左側のメニューから IoT Plug and Play Settings を選択し、上部メニューから Add ⇒ Local folder を選択します。

Pick a folder をクリックし、ck_rx65n_azure_2ndota_demo¥model フォルダを選択します。

Save をクリックし設定を保存します。

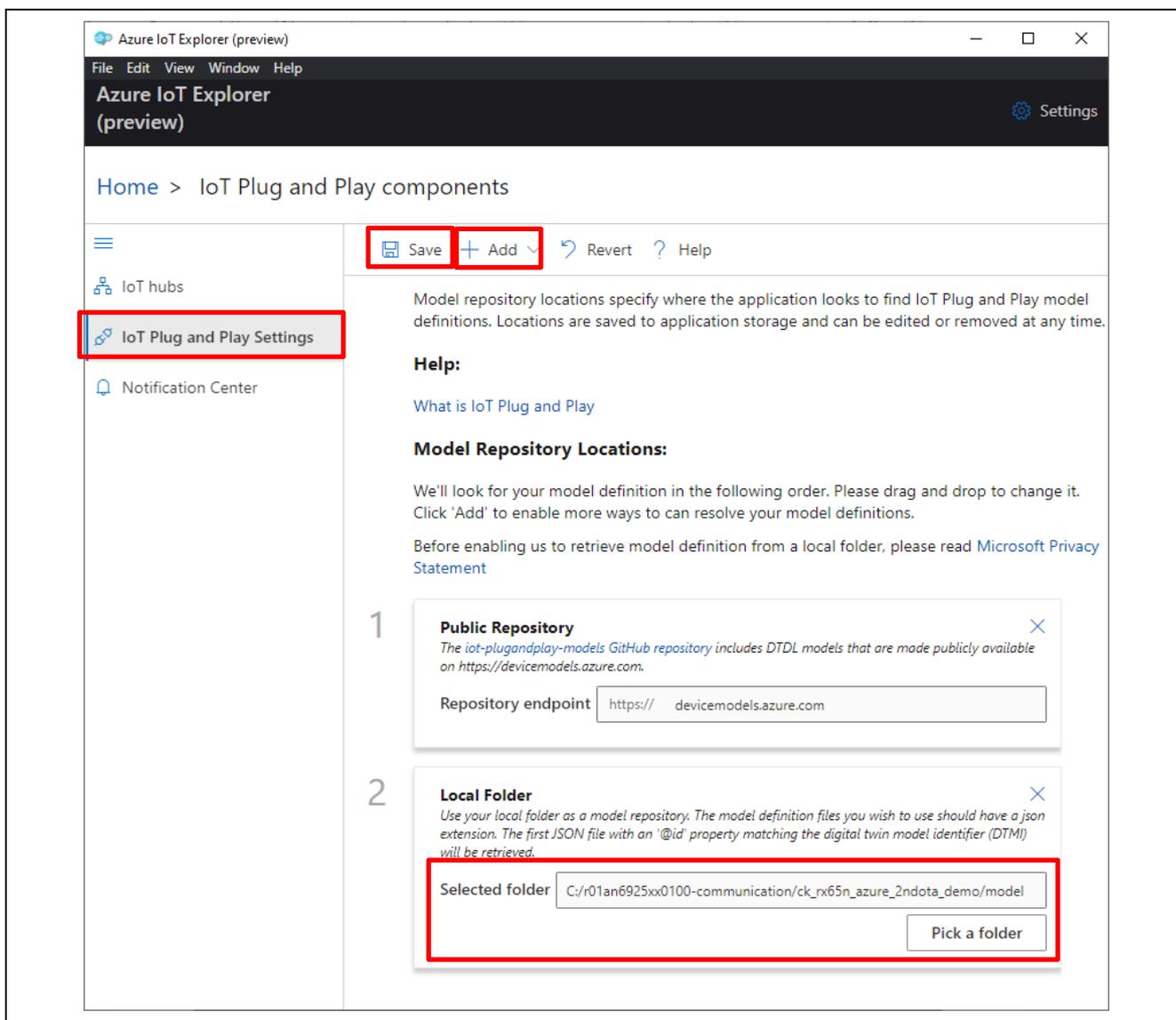


図 6-22 IoT Plug and Play Settings の設定

6.3.3 IoT Hub の設定

ブラウザで Azure ポータルを開き、作成した IoT Hub からプライマリ接続文字列を入手します。

Azure ポータルで以下の順に選択します。

Azure IoT Hub ⇒ Shared access policies ⇒ iothubowner ⇒ Primary connection string

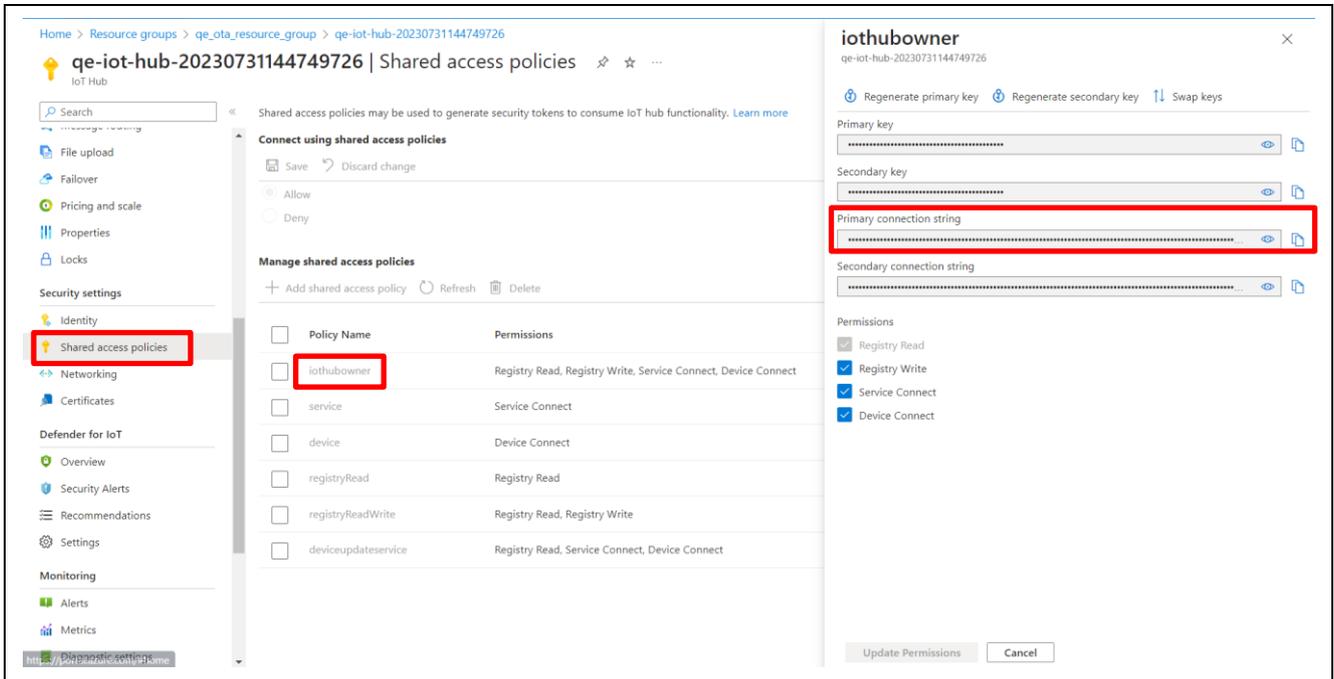


図 6-23 プライマリ接続文字列の入手

Azure IoT Explorer の左側メニューから IoT hubs を選択します。

「Connect via IoT Hub connection string」をクリックします。

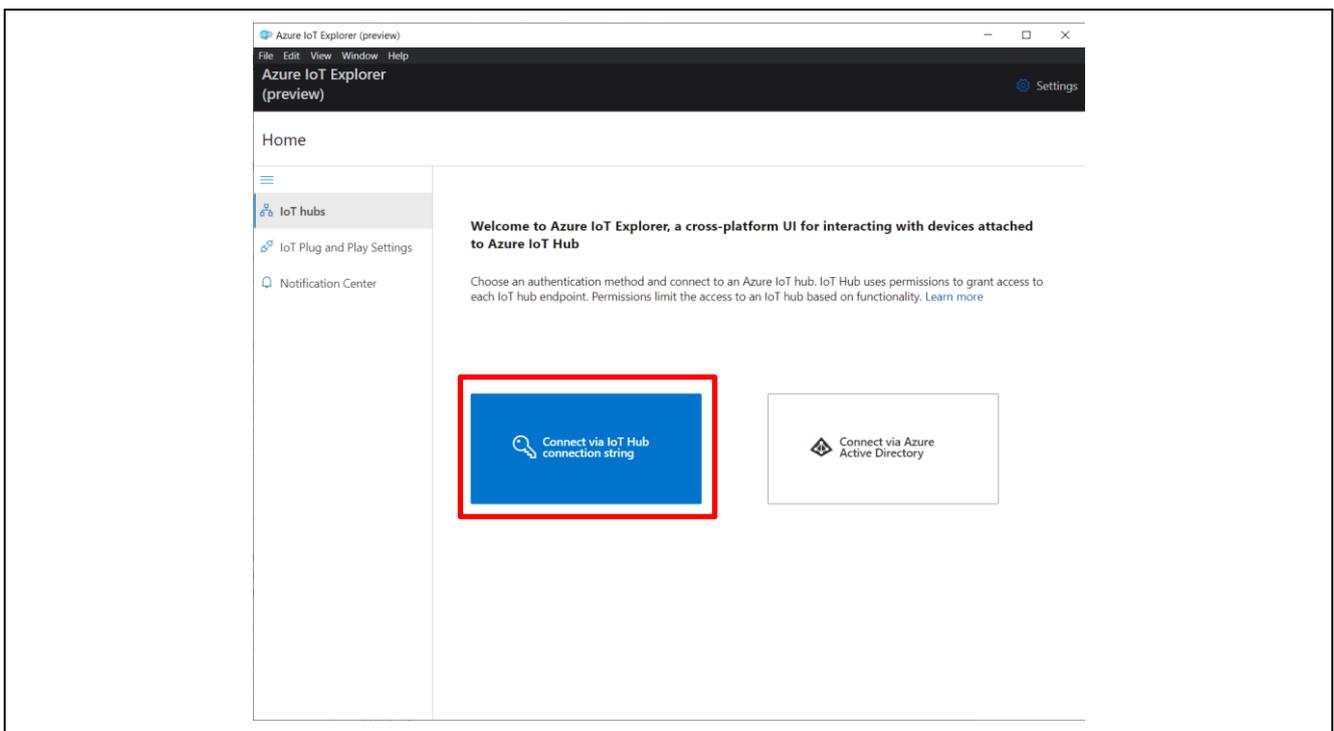


図 6-24 IoT Hub との接続方法選択

「Add connection」をクリックします。

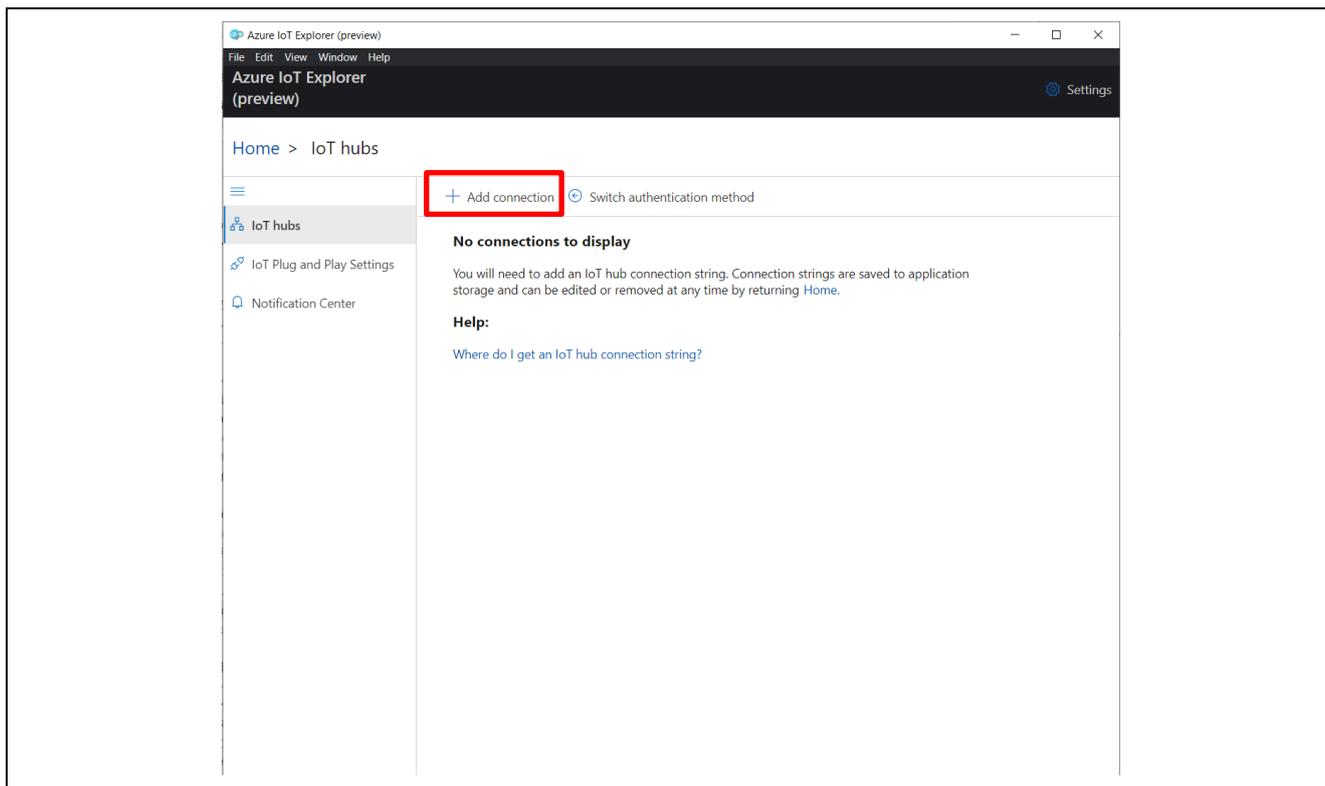


図 6-25 IoT Hub との接続設定

Connection string に入手したプライマリ接続文字列を入力します。

Save をクリックします。

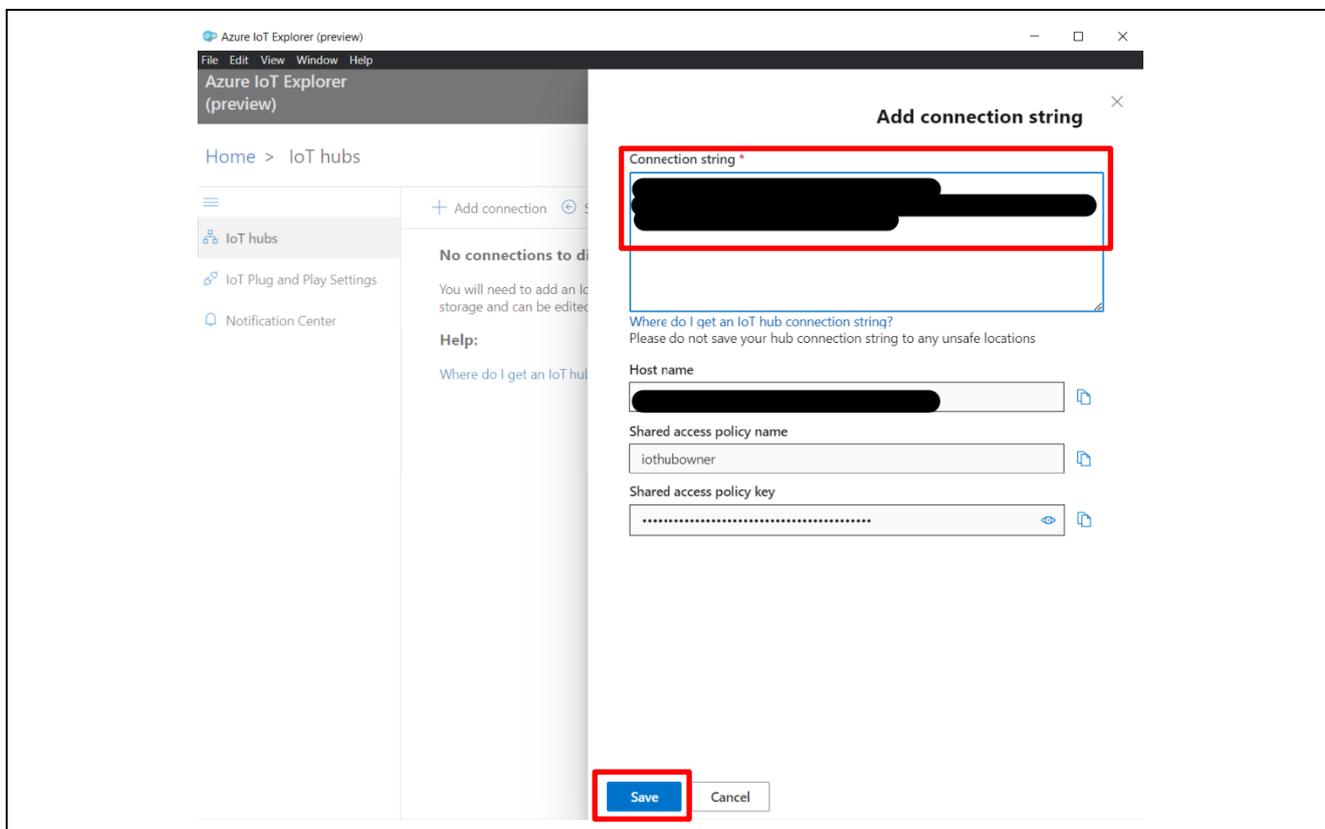


図 6-26 プライマリ接続文字列の入力

Azure IoT hub に登録されているデバイス一覧が表示されます。

表示された Device ID のリンクをクリックします。

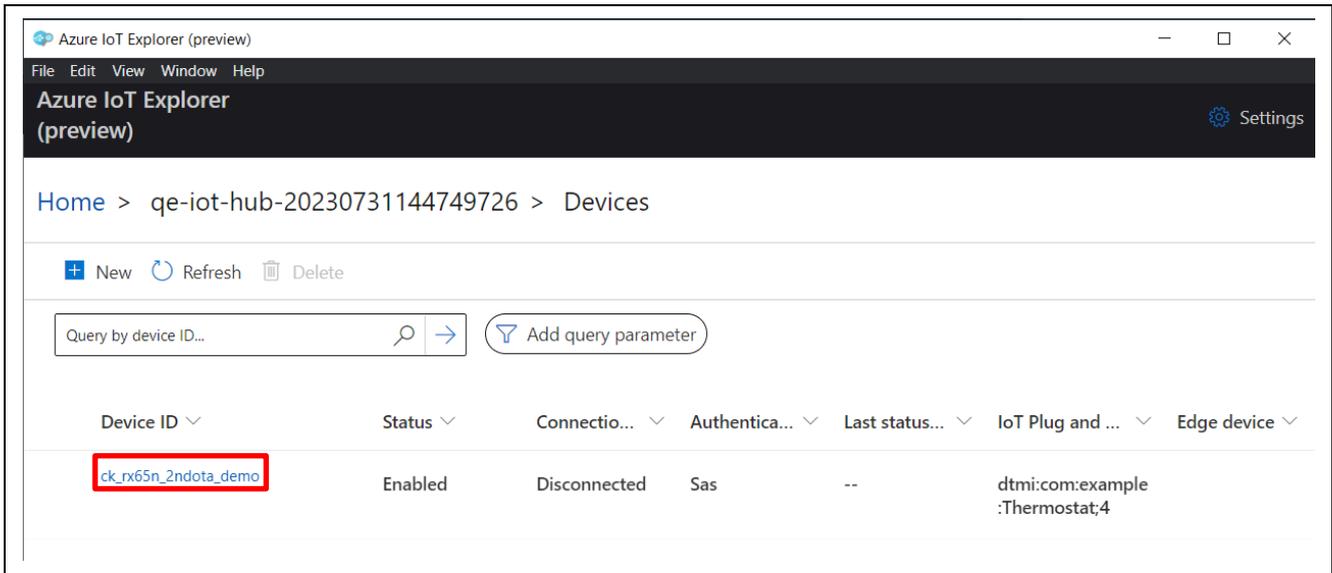


図 6-27 IoT Hub の設定完了

6.3.4 Telemetry データの受信開始

左側メニューから Telemetry を選択します。

Start をクリックすると Host から送られてきた Telemetry データの表示を開始します。

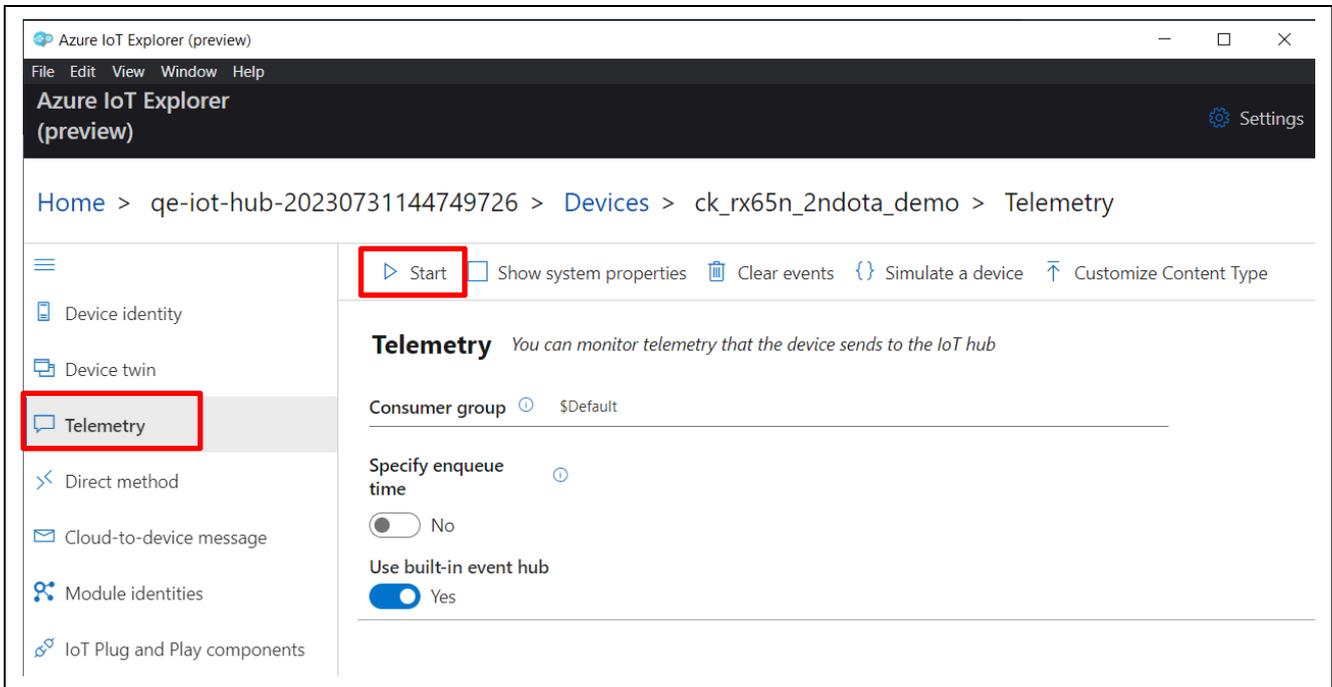


図 6-28 Telemetry データの受信開始

7. デモの実行手順

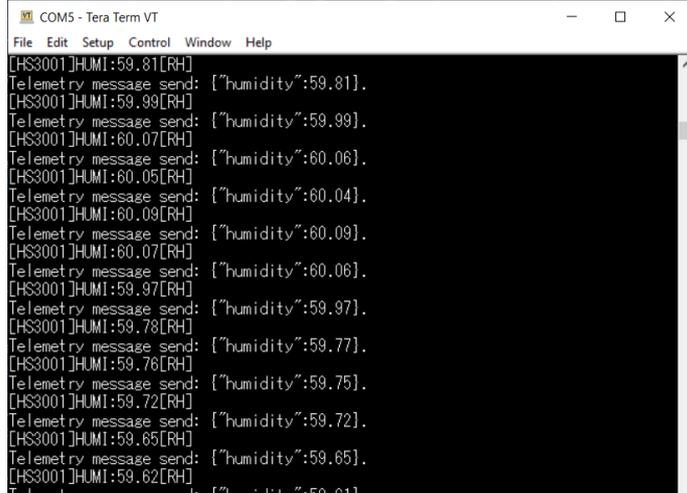
デモを実行する手順について説明します。

7.1 初期状態の動作確認

6章のデモのセットアップが完了している状態で、TB-RX660 のリセットスイッチ(RESET)を押下しハードウェアリセットを実行します。同様に CK-RX65N もリセットスイッチ(S1)を押下しハードウェアリセットを実行します。

ターミナルソフトで各マイコンからのログを確認します。

図 7-1 に CK-RX65N のログ画面を示します。HS3001 センサの湿度データが出力されていることを確認します。

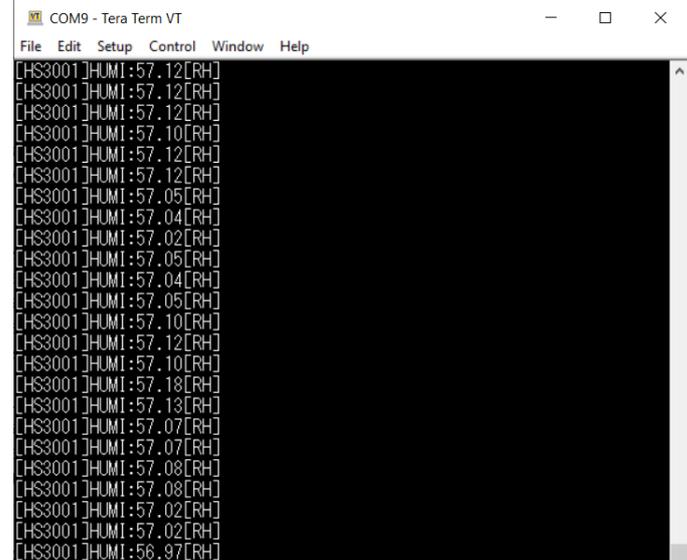


```
COM5 - Tera Term VT
File Edit Setup Control Window Help
[HS3001]HUMI:59.81[RH]
Telemetry message send: ["humidity":59.81].
[HS3001]HUMI:59.99[RH]
Telemetry message send: ["humidity":59.99].
[HS3001]HUMI:60.07[RH]
Telemetry message send: ["humidity":60.06].
[HS3001]HUMI:60.05[RH]
Telemetry message send: ["humidity":60.04].
[HS3001]HUMI:60.09[RH]
Telemetry message send: ["humidity":60.09].
[HS3001]HUMI:60.07[RH]
Telemetry message send: ["humidity":60.06].
[HS3001]HUMI:59.97[RH]
Telemetry message send: ["humidity":59.97].
[HS3001]HUMI:59.78[RH]
Telemetry message send: ["humidity":59.77].
[HS3001]HUMI:59.76[RH]
Telemetry message send: ["humidity":59.75].
[HS3001]HUMI:59.72[RH]
Telemetry message send: ["humidity":59.72].
[HS3001]HUMI:59.65[RH]
Telemetry message send: ["humidity":59.65].
[HS3001]HUMI:59.62[RH]
Telemetry message send: ["humidity":59.61].
```

図 7-1 CK-RX65N のログ画面

次に、図 7-2 に TB-RX660 のログ画面を示します。HS3001 センサの湿度データのみ表示されていることを確認します。

また、初期状態では TB-RX660 の LED0 が点滅します。



```
COM9 - Tera Term VT
File Edit Setup Control Window Help
[HS3001]HUMI:57.12[RH]
[HS3001]HUMI:57.12[RH]
[HS3001]HUMI:57.12[RH]
[HS3001]HUMI:57.10[RH]
[HS3001]HUMI:57.12[RH]
[HS3001]HUMI:57.12[RH]
[HS3001]HUMI:57.05[RH]
[HS3001]HUMI:57.04[RH]
[HS3001]HUMI:57.02[RH]
[HS3001]HUMI:57.05[RH]
[HS3001]HUMI:57.04[RH]
[HS3001]HUMI:57.05[RH]
[HS3001]HUMI:57.10[RH]
[HS3001]HUMI:57.12[RH]
[HS3001]HUMI:57.10[RH]
[HS3001]HUMI:57.18[RH]
[HS3001]HUMI:57.13[RH]
[HS3001]HUMI:57.07[RH]
[HS3001]HUMI:57.07[RH]
[HS3001]HUMI:57.08[RH]
[HS3001]HUMI:57.08[RH]
[HS3001]HUMI:57.02[RH]
[HS3001]HUMI:57.02[RH]
[HS3001]HUMI:56.97[RH]
```

図 7-2 TB-RX660 のログ画面

最後に、図 7-3 に Azure IoT Explorer の画面を示します。HS3001 センサから取得した湿度データが表示されていることを確認します。

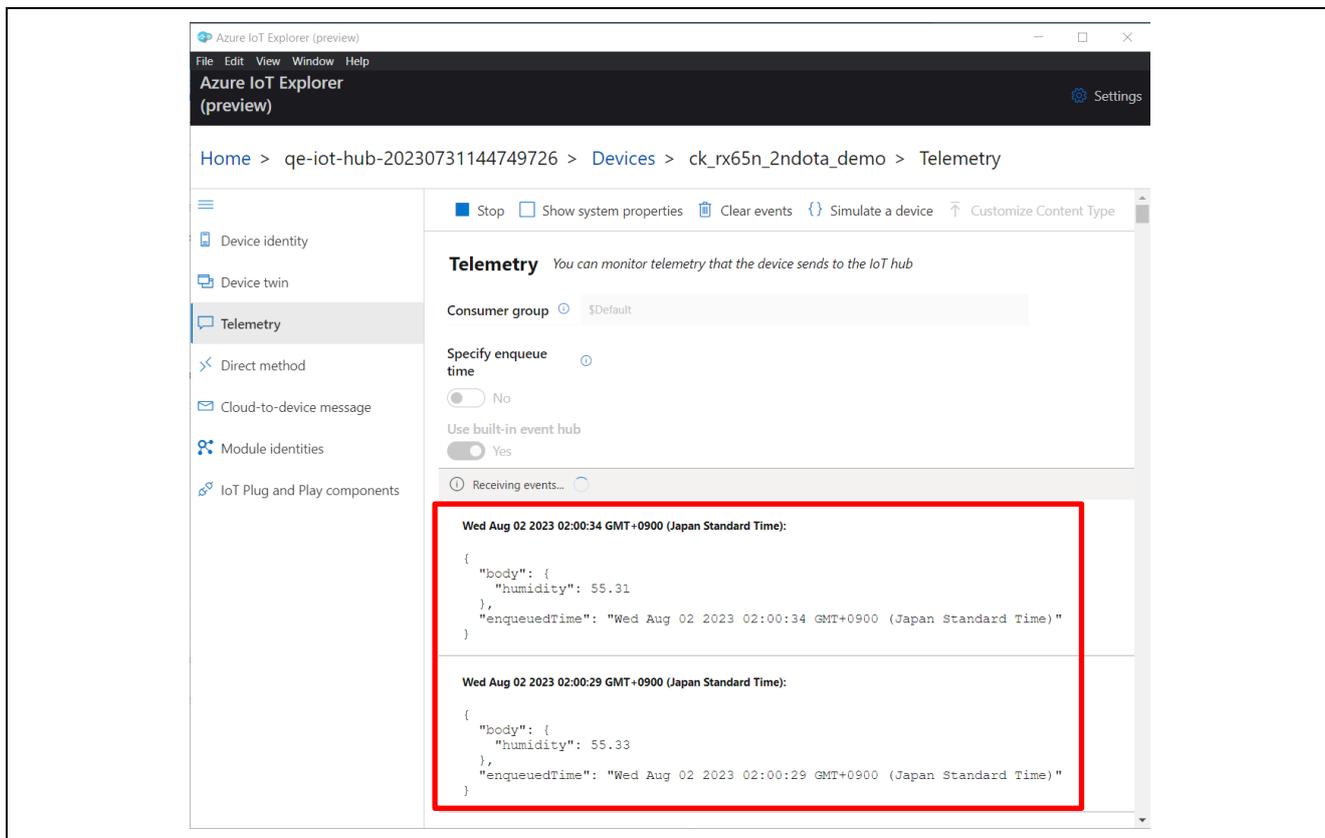


図 7-3 セカンダリ OTA アップデート前の Azure IoT Explorer のテレメトリ表示

この状態がセカンダリ OTA アップデート実行前の初期状態です。

7.2 OTA アップデートの実行

7.2.1 更新ファームウェアの作成

7.2.1.1 CK-RX65N の更新ファームウェアの作成

「[RX ファミリ AWS/Azure を利用したファームウェア更新ソフトの開発ガイド QE for OTA](#)」の「4.6.1 更新ファームウェアの作成」章の Azure の手順を実行します。更新ファームウェアのバージョンは「2.0.0」としてください。

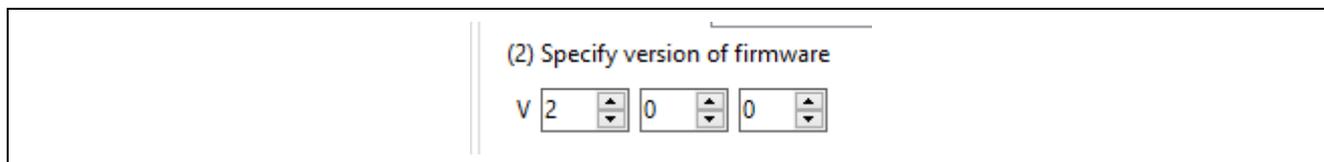


図 7-4 更新ファームウェアのバージョン設定

これにより CK-RX65N の更新ファームウェアが作成できます。

「ck_rx65n_azure_2ndota_demo¥QE-OTA¥apl¥v2.0.0¥<IoT デバイス名>」フォルダにある「<IoT デバイス名>.rsu」ファイルを、

ck_rx65n_azure_2ndota_demo¥tools¥AzureDeviceUpdateScripts フォルダ

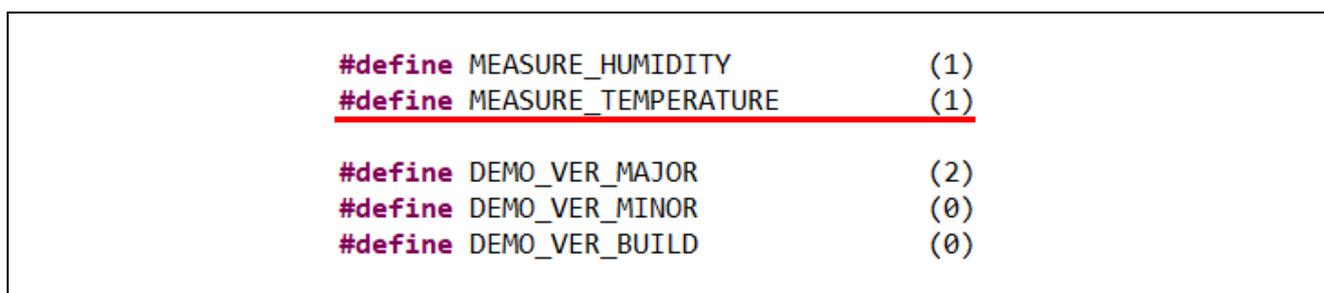
にコピーし、ファイルの名前を「host_update_firm_2.0.0.rsu」に変更します。

7.2.1.2 TB-RX660 の更新ファームウェアの作成

(1) rx660_tb_2ndota_demo プロジェクトのソースコードの変更

rx660_tb_2ndota_demo/src/rx660_tb_2ndota_demo.c の MEASURE_TEMPERATURE のマクロを(1)にセットします。

また、その下の DEMO_VER_MAJOR, DEMO_VER_MINOR, DEMO_VER_BUILD を変更することで、実行時にログ表示されるバージョン値を変更できます。



(2) 更新ファームウェア(MOT ファイル形式)の作成

rx660_tb_2ndota_demo プロジェクトをビルドし、MOT ファイルを作成します。

(3) 更新ファームウェア(RSU ファイル形式)の作成

作成された rx660_tb_2ndota_demo の MOT ファイルを、Renesas Image Generator を使用して RSU 形式の更新ファームウェアに変換します。

rx660_tb_2ndota_demo¥RenasasImageGenerator フォルダで以下のコマンドを実行し、RSU 形式の更新ファームウェア leaf_update_firm_2.0.0.rsu を作成します。

```
> python .¥image-gen.py -iup ..¥HardwareDebug¥rx660_tb_2ndota_demo.mot -
o ..¥..¥ck_rx65n_azure_2ndota_demo¥tools¥AzureDeviceUpdateScripts¥leaf_update_firm_2.0.
0 -key .¥keys¥secp256r1.privatekey -
ip .¥RX660_Linear_Half_ImageGenerator_PRM_2ndota_demo.csv -vt ecdsa
```

7.2.2 マニフェストファイルの作成

Windows PowerShell を起動し、ck_rx65n_azure_2ndota_demo¥tools¥AzureDeviceUpdateScripts フォルダで以下のコマンドを実行し、マニフェストファイルを作成します。

```
.¥CreateRX65NRSKUpdate.ps1
```

Version, HostPath, LeafPath の入力を求められるため、それぞれ以下のように入力します。

- Version: 2.0.0
マニフェストファイルのバージョンです。各ファームウェアのバージョンとは独立しています。
- HostPath: host_update_firm_2.0.0.rsu
7.2.1.1 で作成した RSU ファイルです。
- LeafPath: leaf_update_firm_2.0.0.rsu
7.2.1.2(3) で作成した RSU ファイルです。

```
# .¥CreateCKRX65NUpdate.ps1

cmdlet CreateCKRX65NUpdate.ps1 at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
Version: 2.0.0
HostPath: host_update_firm_2.0.0.rsu
LeafPath: leaf_update_firm_2.0.0.rsu
Preparing update RENESAS.CK-RX65N.2.0.0 ...
Preparing child update RENESAS.CK-RX65N-Leaf.2.0.0 ...
Preparing child update manifest RENESAS.CK-RX65N-Leaf.2.0.0 ...
Saving child manifest files and payload to .¥RENESAS.CK-RX65N.2.0.0 ...

Preparing parent update RENESAS.CK-RX65N.2.0.0 ...
Generating an import manifest RENESAS.CK-RX65N.2.0.0 ...
Saving parent manifest file and payload(s) to .¥RENESAS.CK-RX65N.2.0.0 ...
```

図 7-5 マニフェストファイルの作成

実行完了すると【RENESAS.CK-RX65N.2.0.0】フォルダが作成され、フォルダ内にマニフェストファイルと更新ファームウェアが生成されます。

7.2.3 更新のデプロイ

(1) デバイスに ADUGroup タグを追加

ブラウザで Azure ポータルを開き、IoT Hub の左メニューから「Devices」を選択します。

6.2.3(6)で QE for OTA で作成した Device ID にチェックを入れ、「Assign tags」をクリックします。

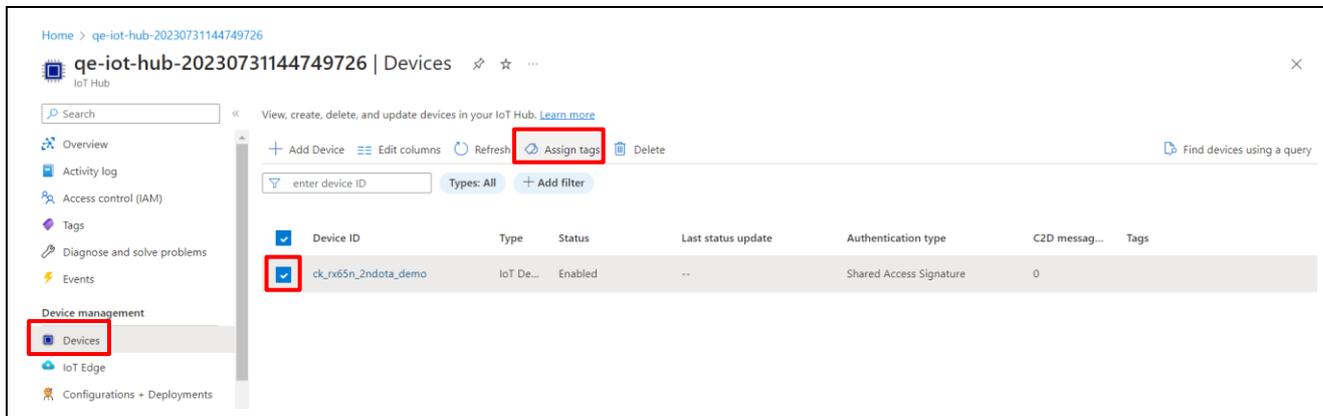


図 7-6 デバイスを選択しタグを追加

Name に「ADUGroup」、Value に「OTADemoGroup」と入力し、Save をクリックします。

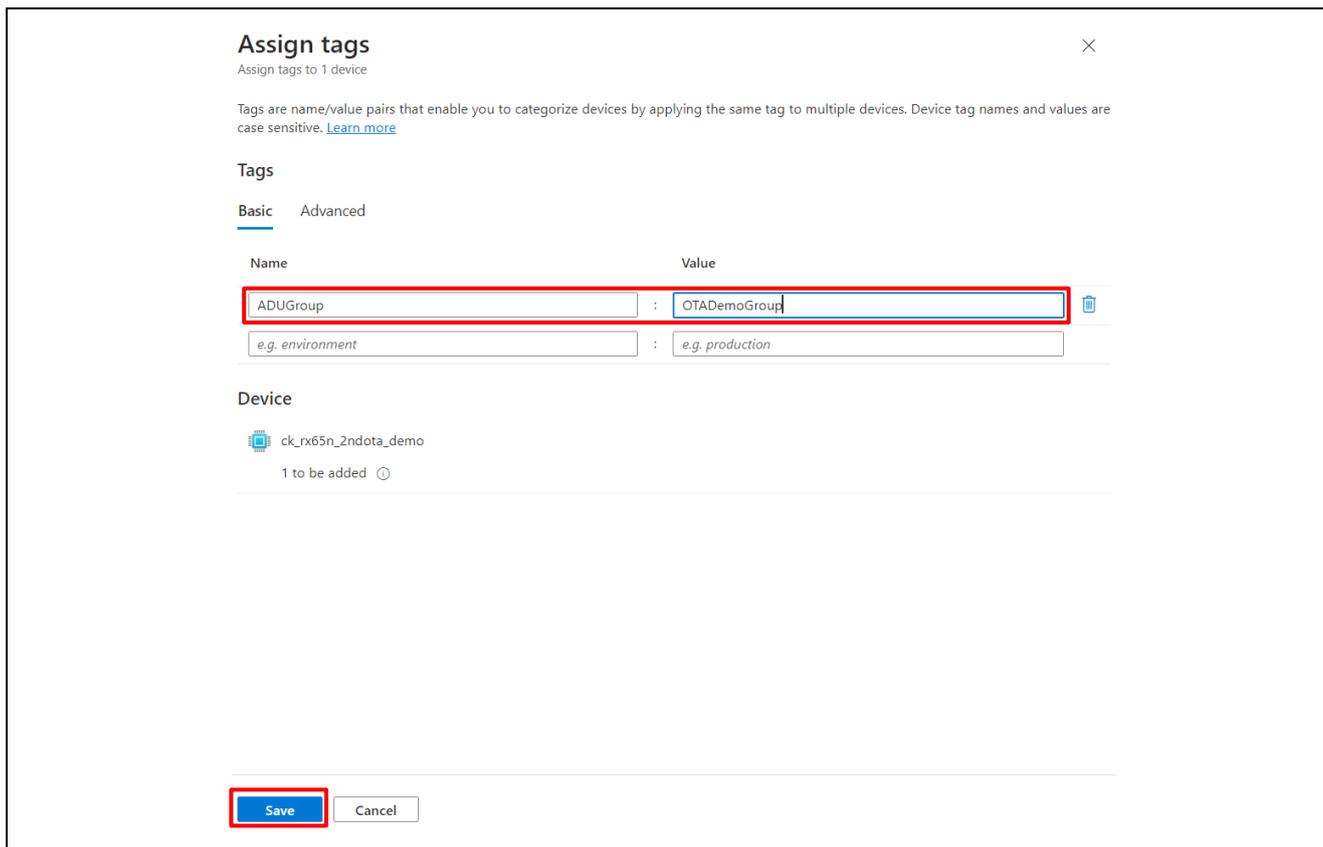


図 7-7 ADUGroup タグの追加

(2) 更新の権限の追加

IoT Hub のページに戻り、左側メニューから「Updates」を選択します。

右上の歯車アイコンをクリックし、表示された吹き出しの「View account details」をクリックします。

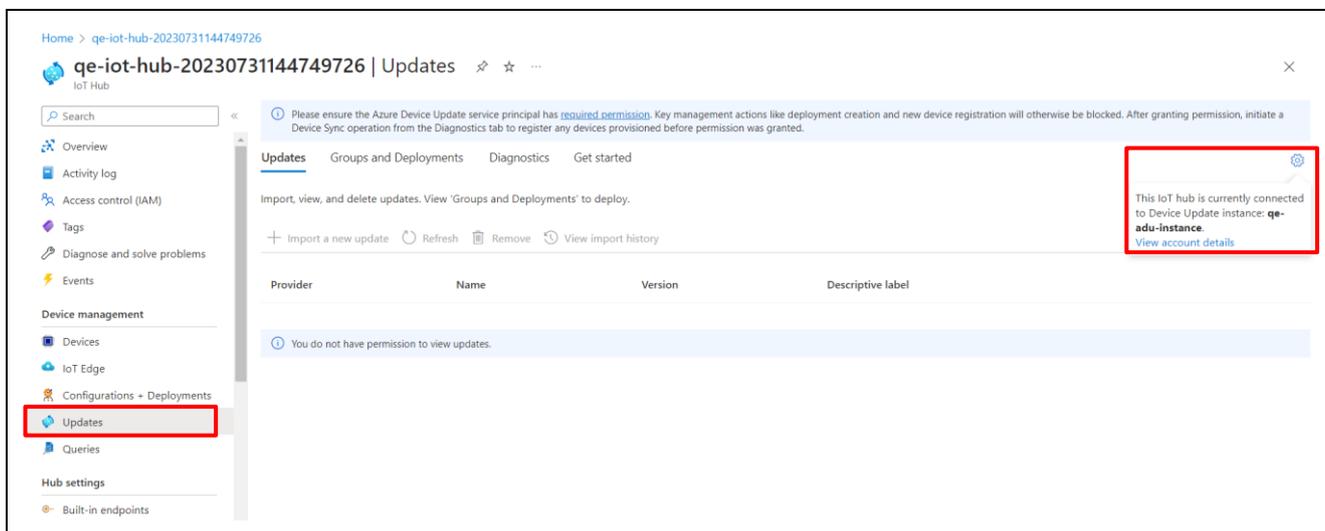


図 7-8 Updates ページから Device update アカウントページに移動

Device update アカウントのページが開くため、左側メニューから「Access control(IAM)」をクリックし、「Add」タブから「Add role assignment」を選択します。

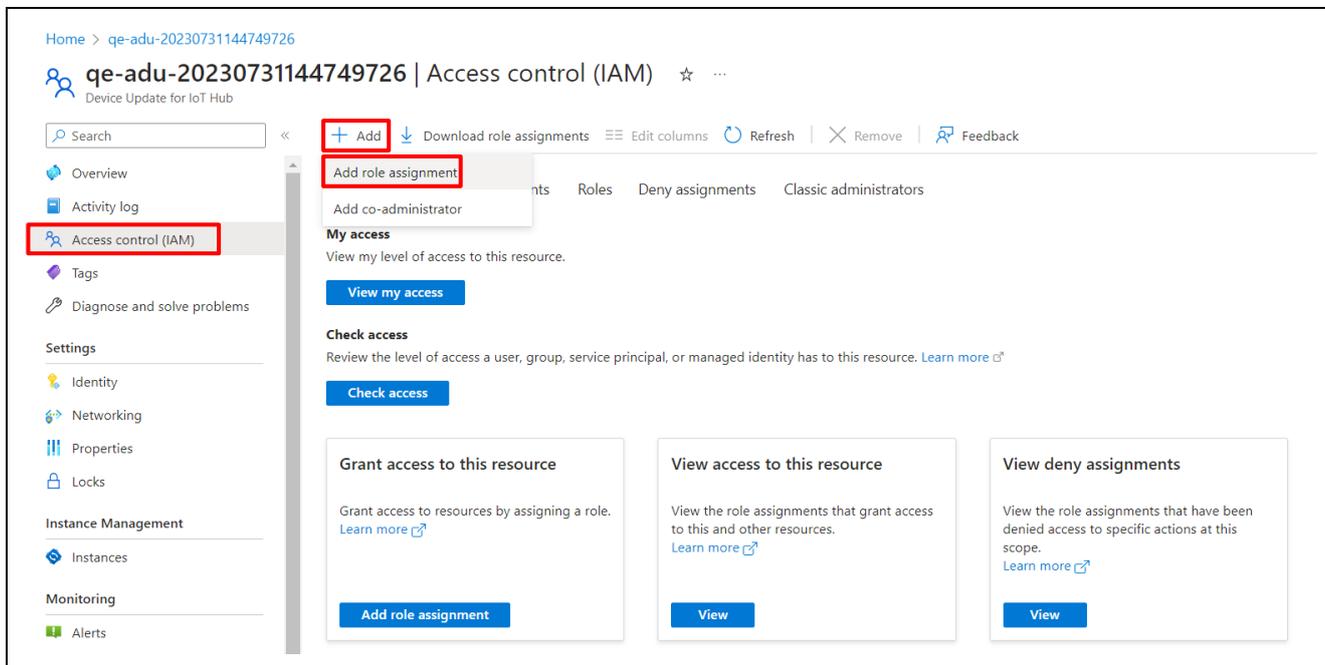


図 7-9 IAM 画面からロールを追加

ロールの一覧から「Device Update Administrator」を選択し、「Next」をクリックします。

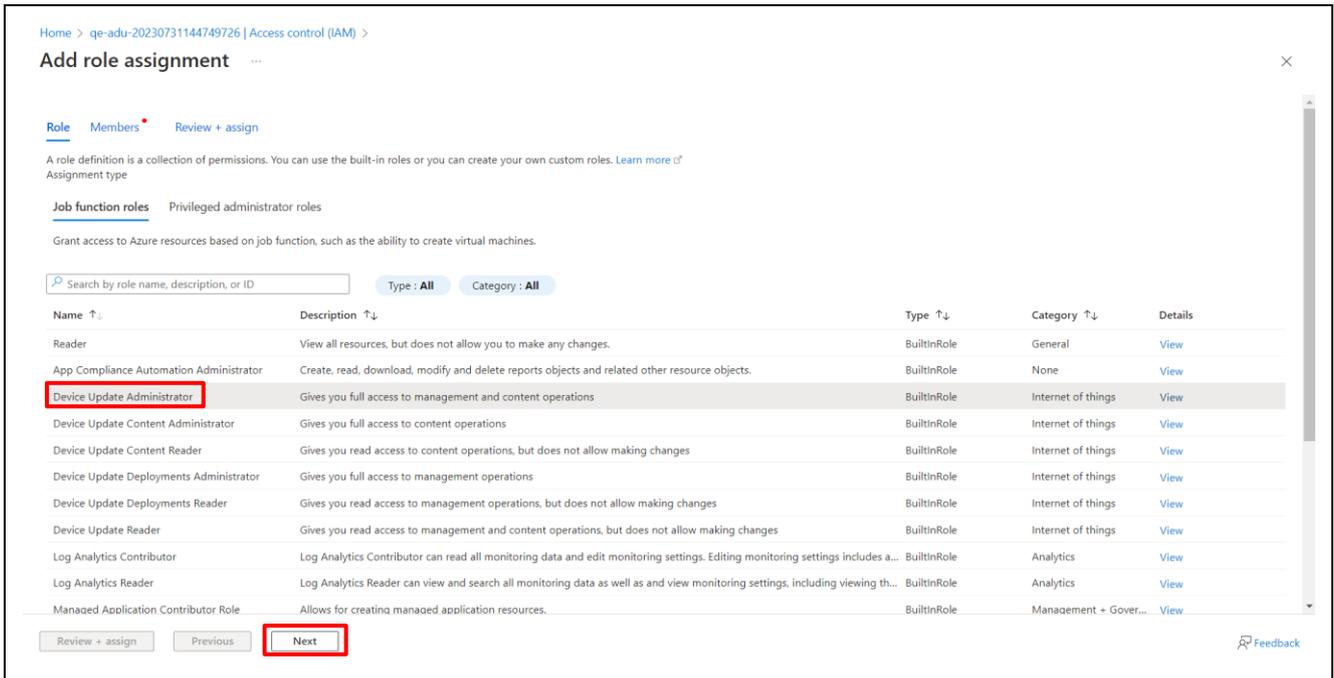


図 7-10 ロール一覧から Device Update Administrator ロールを選択

Select members をクリックし、自分のアカウント名を検索してクリックします。

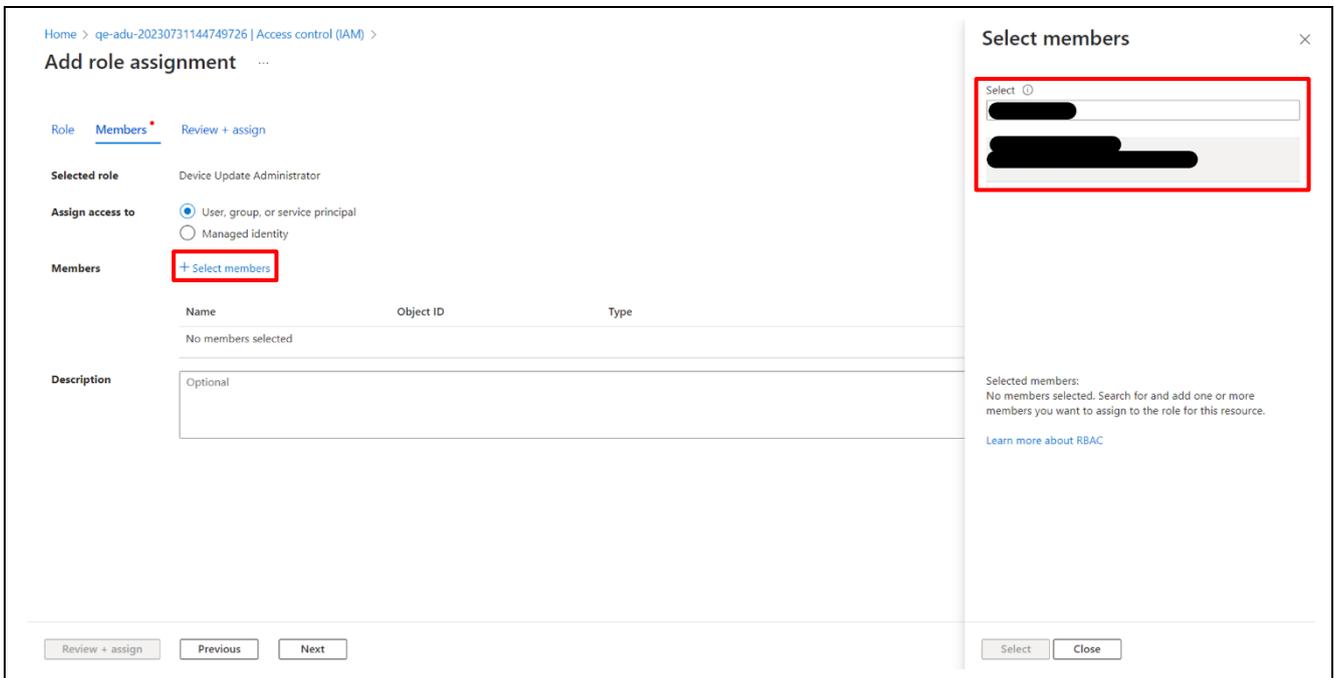


図 7-11 ロールをアサインするメンバーの選択①

自分のアカウントが選択されていることを確認し、「Select」をクリックします。

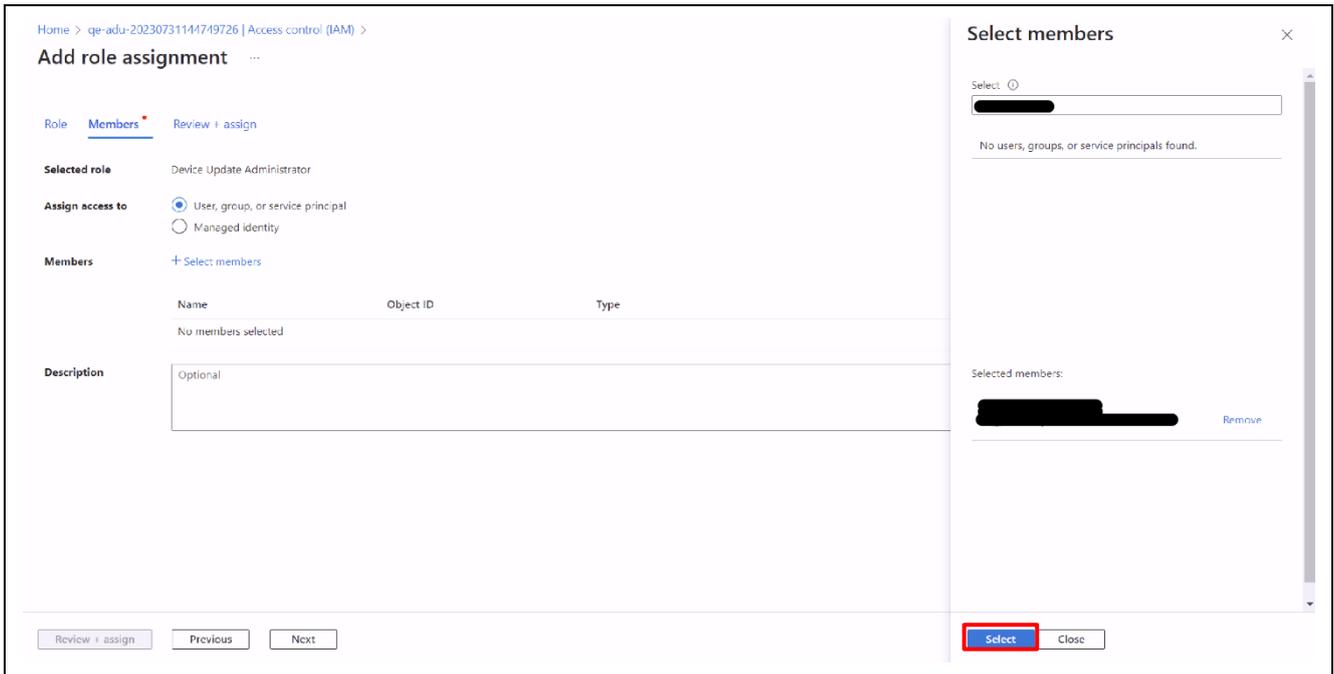


図 7-12 ロールをアサインするメンバーの選択②

自分のアカウントが Members に表示されていることを確認し、「Review + assign」をクリックします。

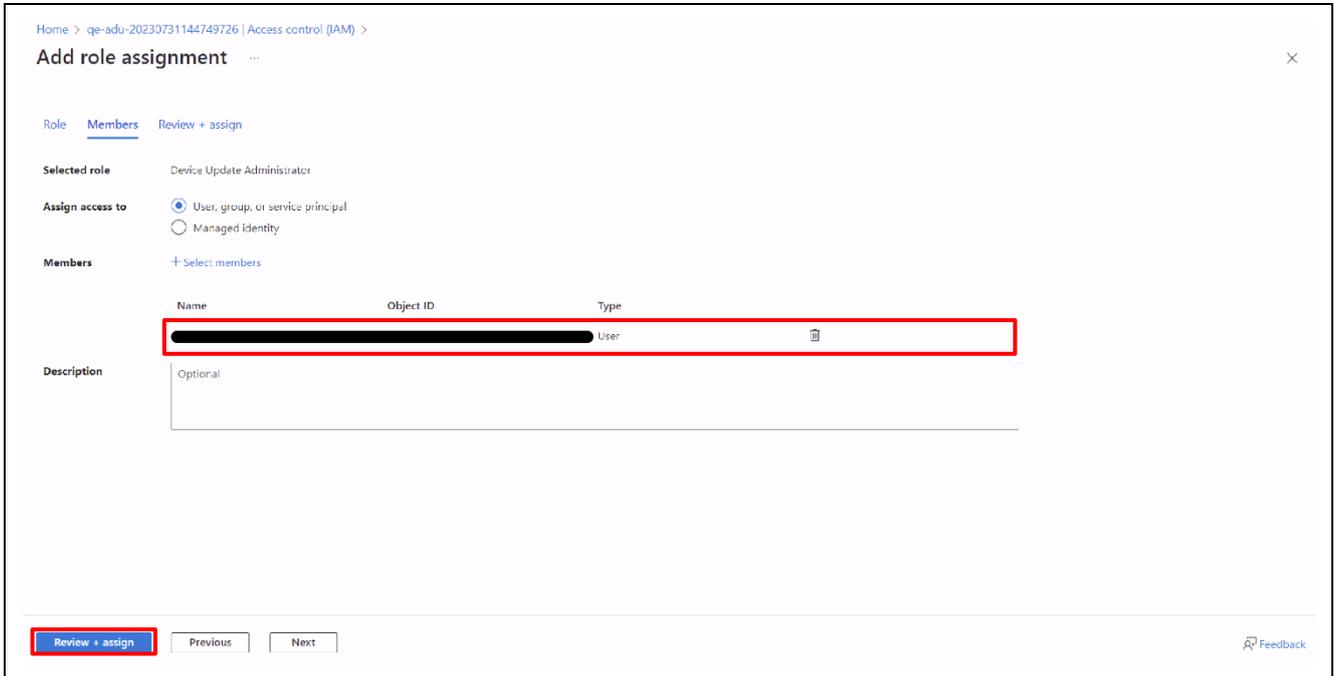


図 7-13 Device Update Administrator ロールのアサイン

(3) 更新のインポート

IoT Hub のページに戻り、左メニューから「Updates」を選択します。

「Import a new update」をクリックします。

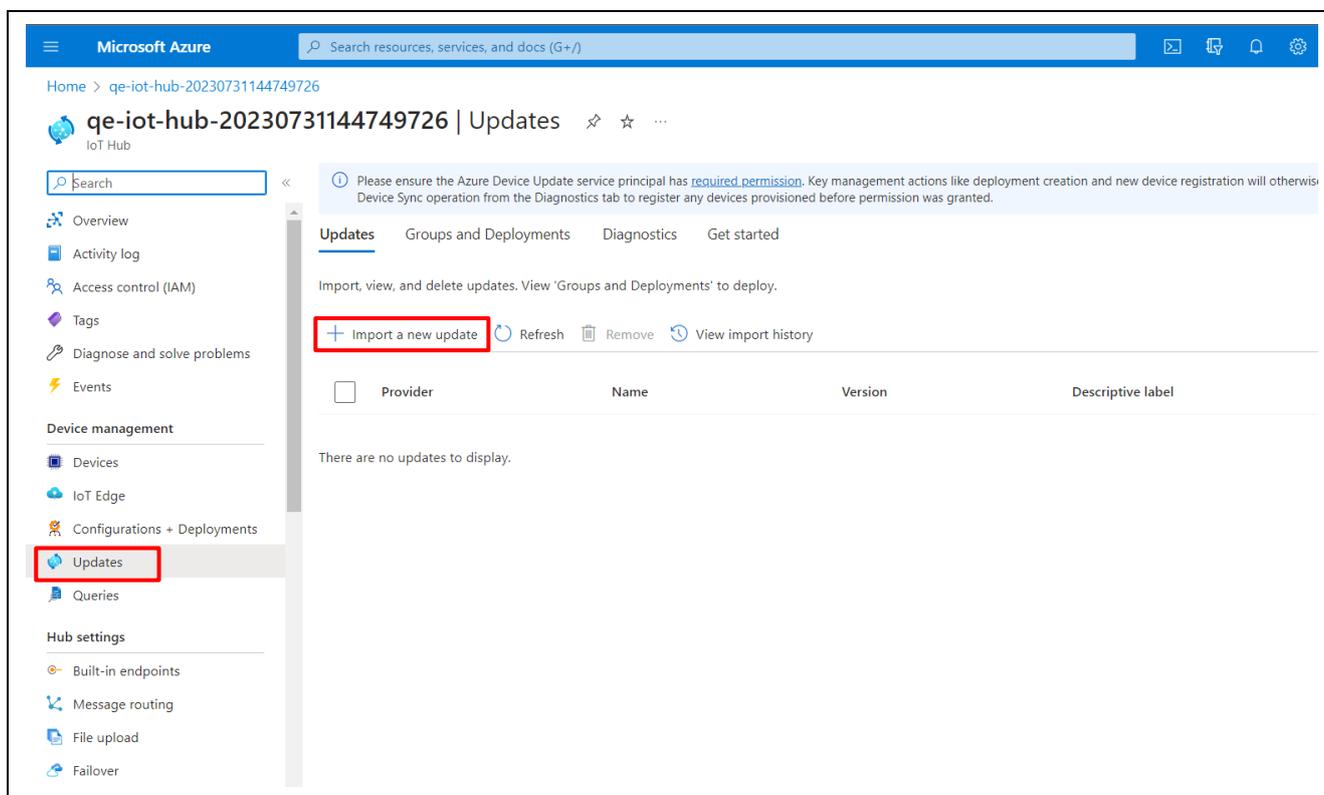


図 7-14 新しい更新プログラムのインポート

「Select from storage container」をクリックします。

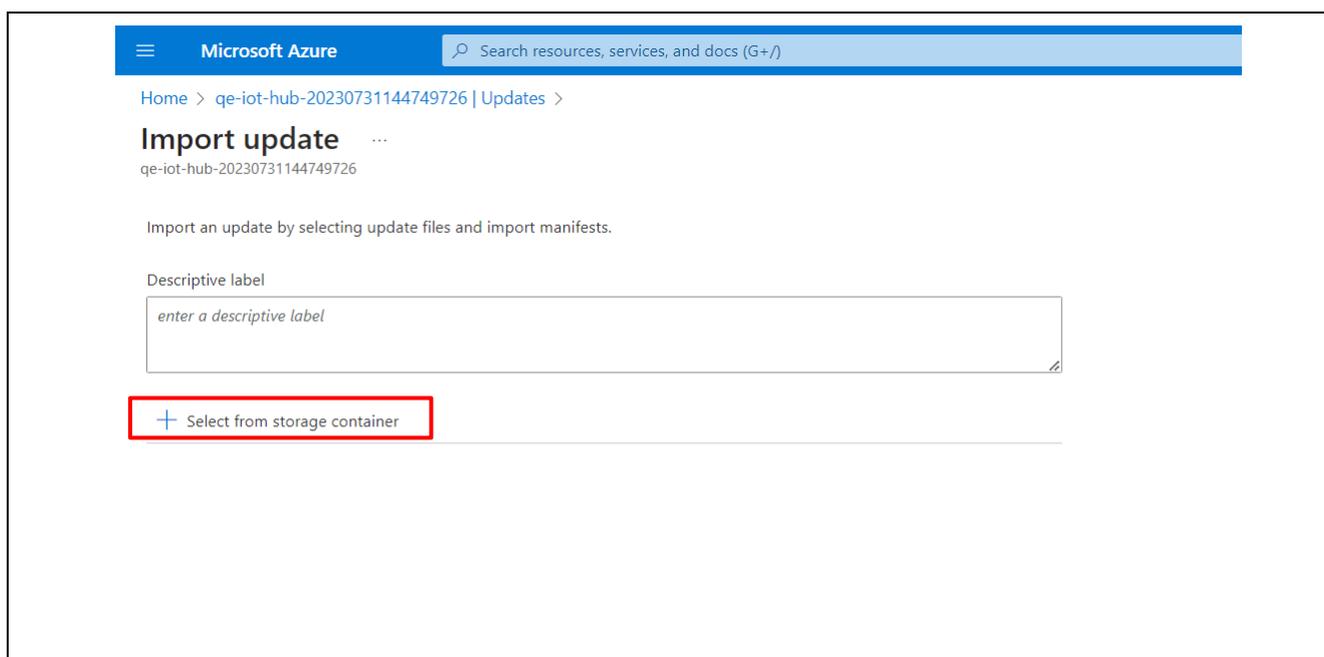


図 7-15 ストレージコンテナから選択

6.2.3(3)で QE for OTA で作成した「Storage account」を選択し、表示された Containers 画面で Container を選択します。

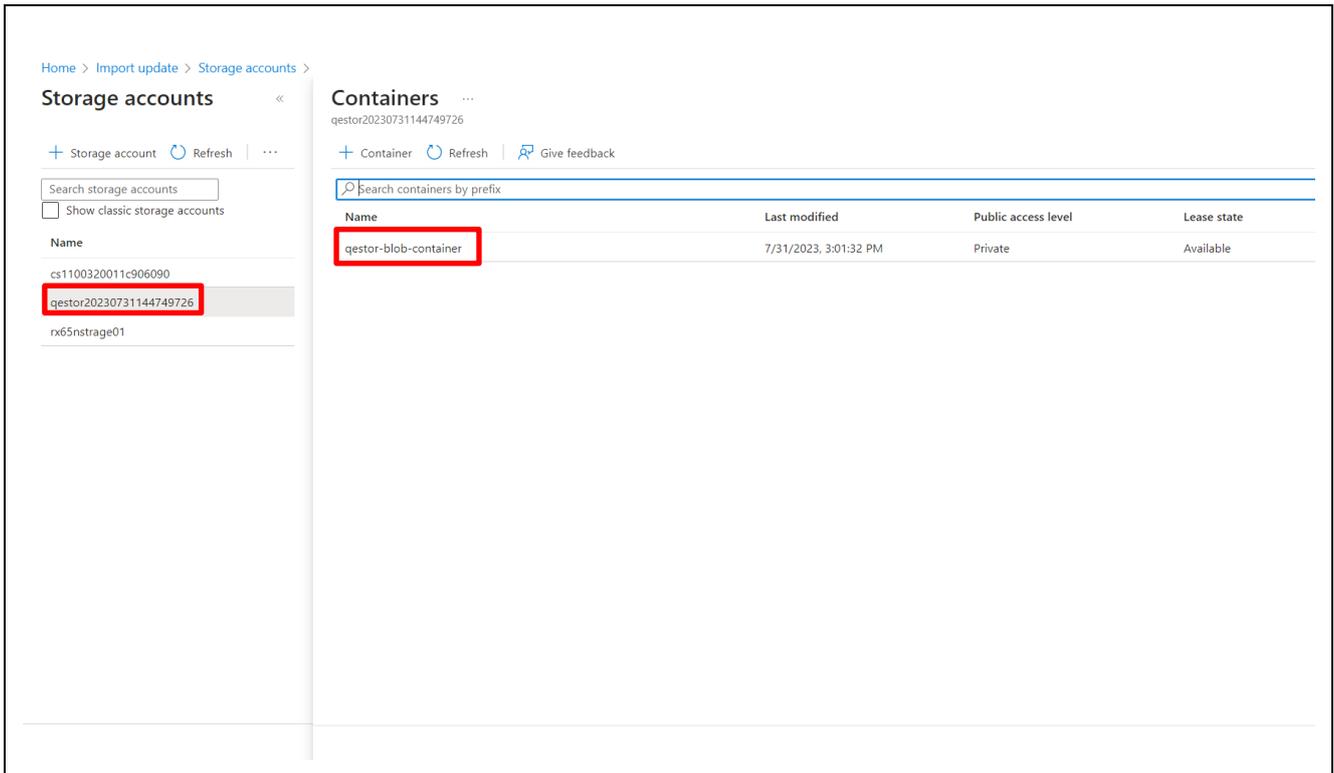


図 7-16 Container の選択

表示された Container 画面で Upload をクリックし、7.2.2 章で作成した【RENESAS.CK-RX65N.2.0.0】フォルダ内のファイル一式をアップロードします。

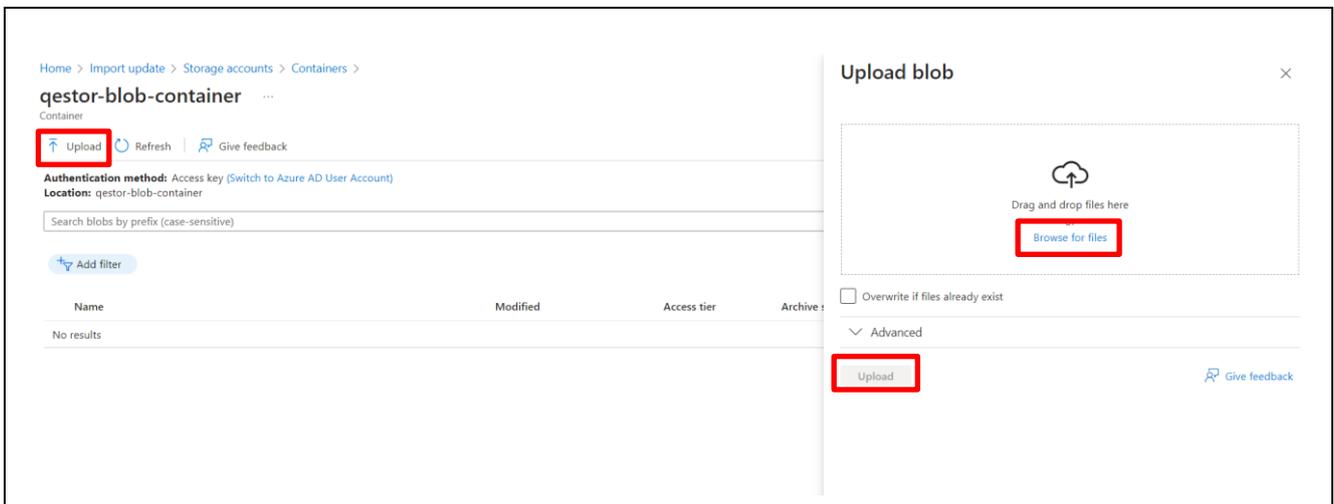


図 7-17 マニフェストファイルと更新ファームウェアのアップロード

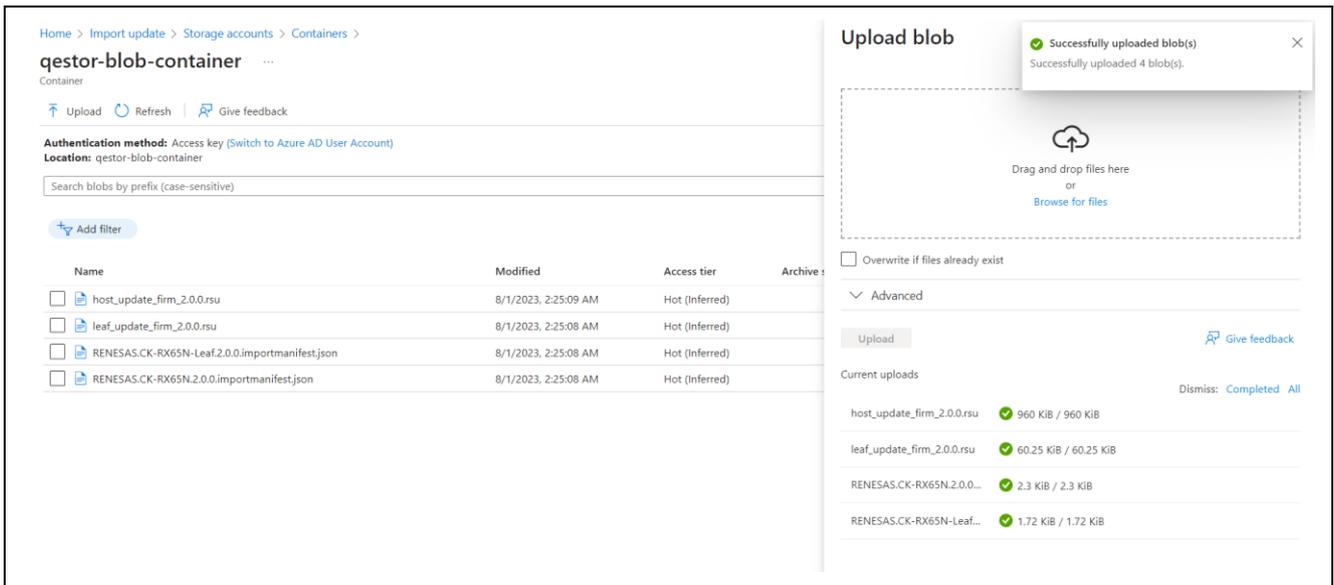


図 7-18 マニフェストファイルと更新ファームウェアのアップロード成功時

アップロードされた全てのファイルのチェックボックスにチェックを入れ、Select をクリックします。

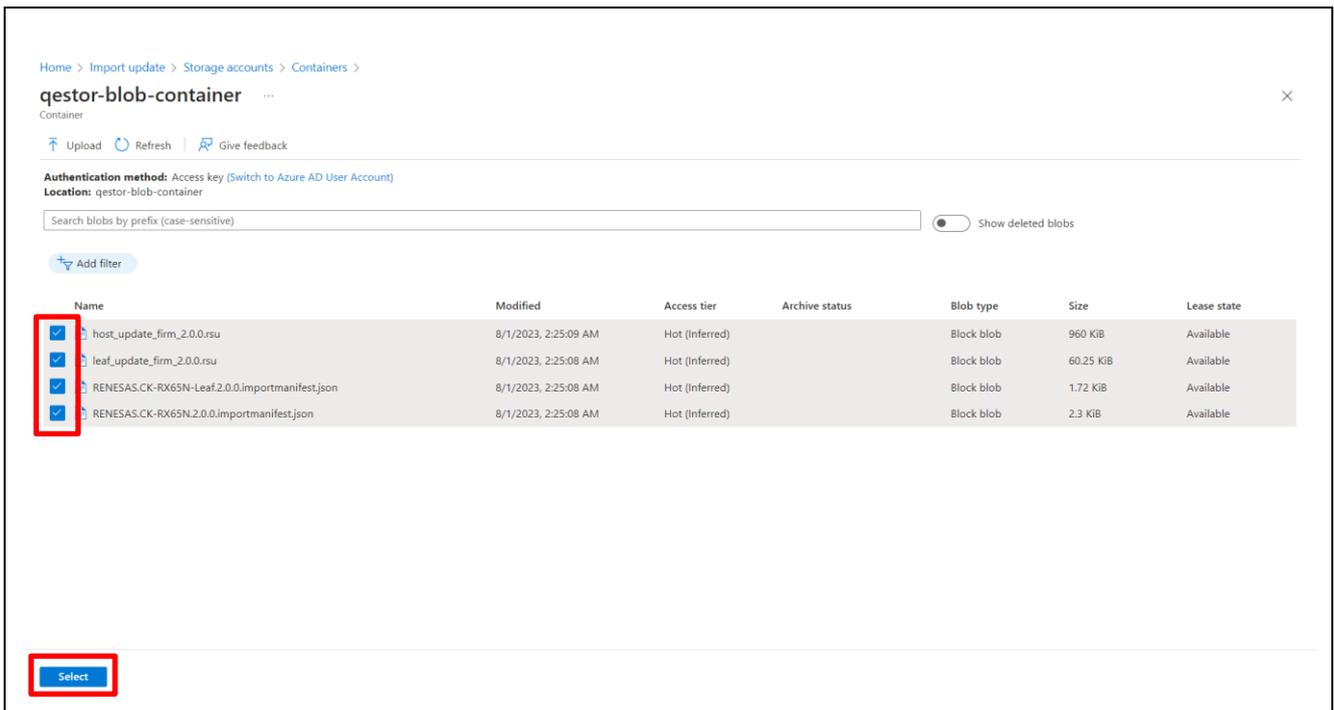


図 7-19 マニフェストファイルと更新ファームウェアの選択

Import update をクリックします。

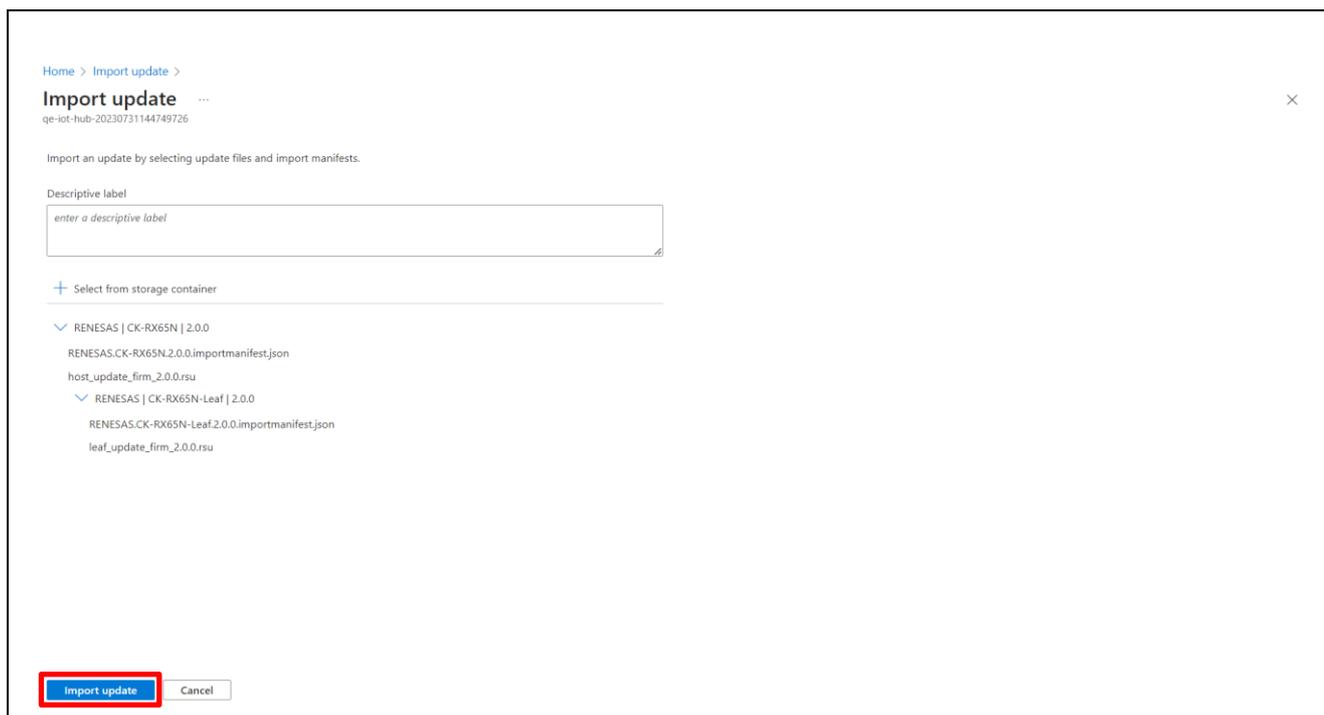


図 7-20 更新ファームウェアのインポート

インポートが完了すると更新の情報が表示されます。処理が完了するまで数分かかることがあります。

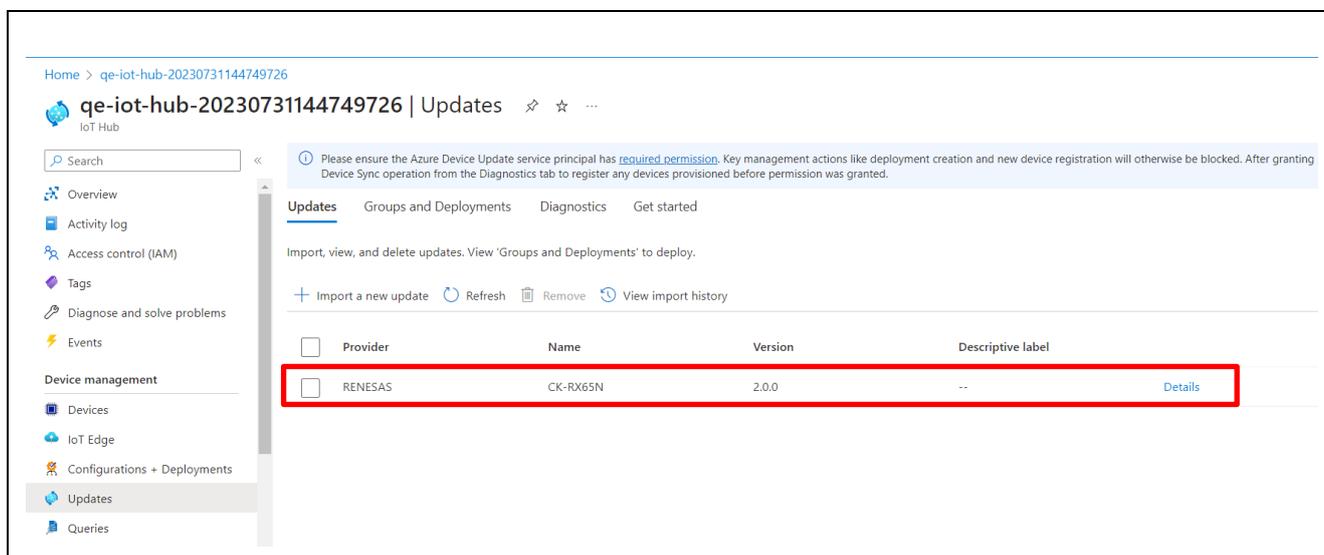


図 7-21 更新のインポート成功時

(4) 更新のデプロイ

Groups and Deployments タブを選択し、7.2.3(1)で作成した「OTADemoGroup」グループをクリックします。グループが表示されない場合は、CK-RX65N の RESET ボタンを押下し、デバイスを IoT Hub に再接続後、画面上の Refresh をクリックしてください。

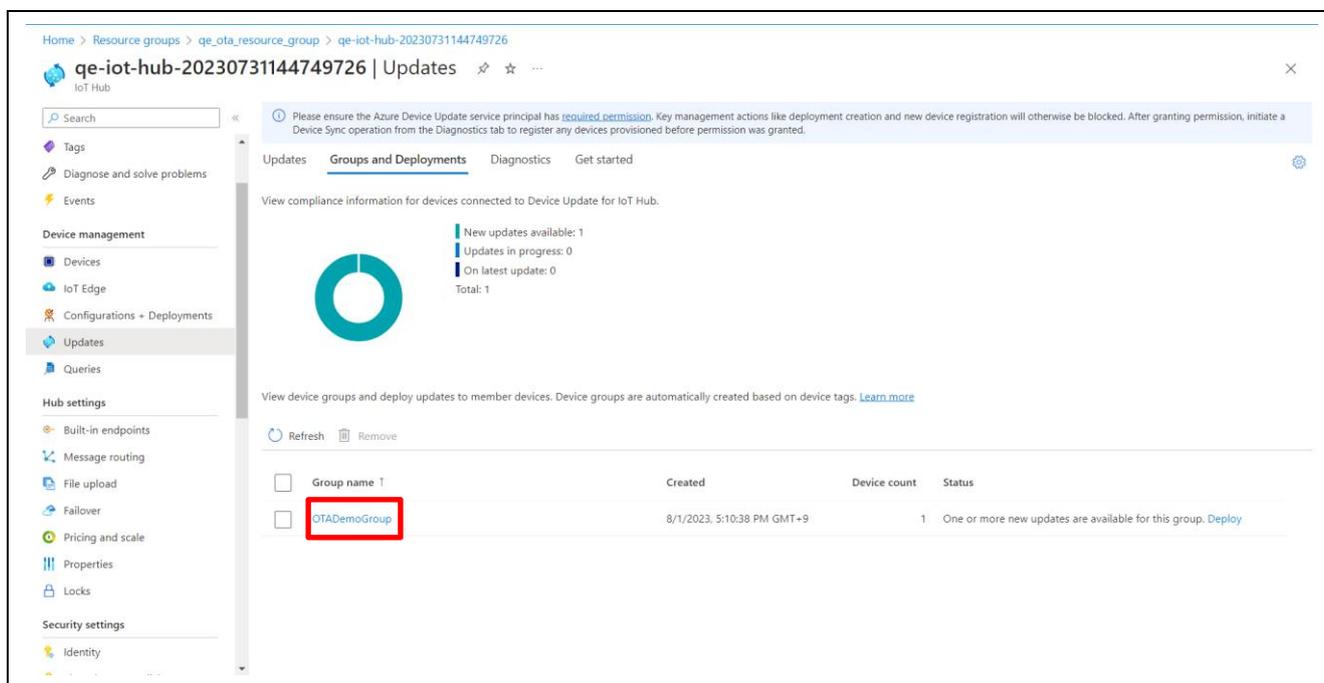


図 7-22 グループとデプロイ画面でグループを選択

Deploy をクリックします。

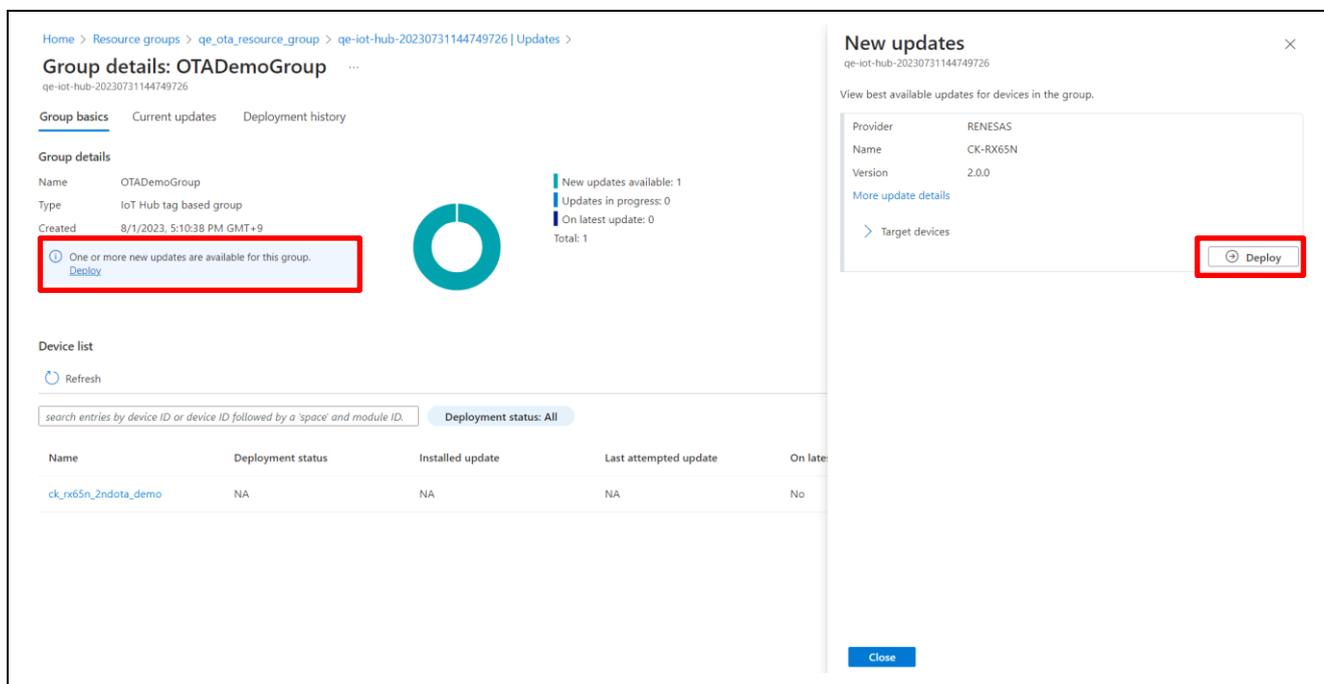


図 7-23 更新のデプロイ

Create をクリックし、更新をデプロイします。

Create deployment ×
qe-iot-hub-20230731144749726

Create a new deployment targeting devices in this group.

Update properties

Provider	RENESAS
Name	CK-RX65N
Version	2.0.0

Group name
OTADemoGroup

Specify when this deployment should start. *

Start immediately.

Start at a scheduled date and time.

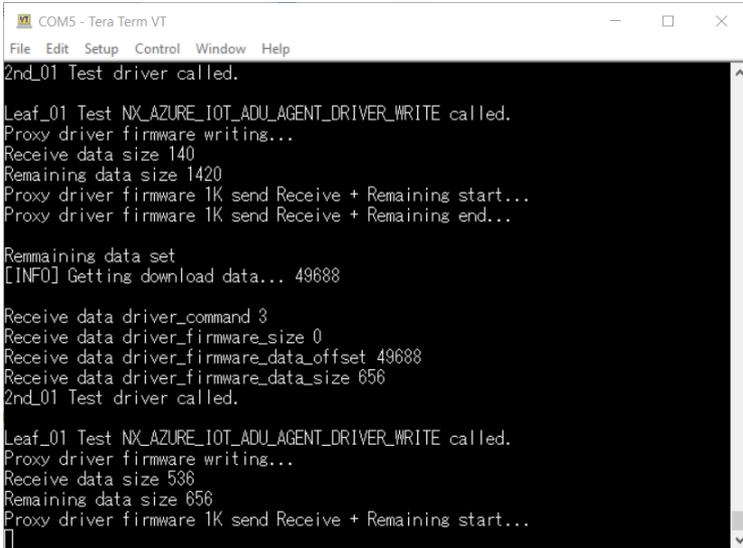
Rollback all applicable devices in the group to a different update if rollback criteria is met

Create Cancel

図 7-24 デプロイの作成

7.3 セカンダリ OTA アップデート実行中の動作確認

デプロイの作成から数秒~数十秒ほどで OTA アップデートが始まります。CK-RX65N と TB-RX660 の両方から、セカンダリ OTA アップデートの進捗がログ出力されます。



```
COM5 - Tera Term VT
File Edit Setup Control Window Help
2nd_01 Test driver called.

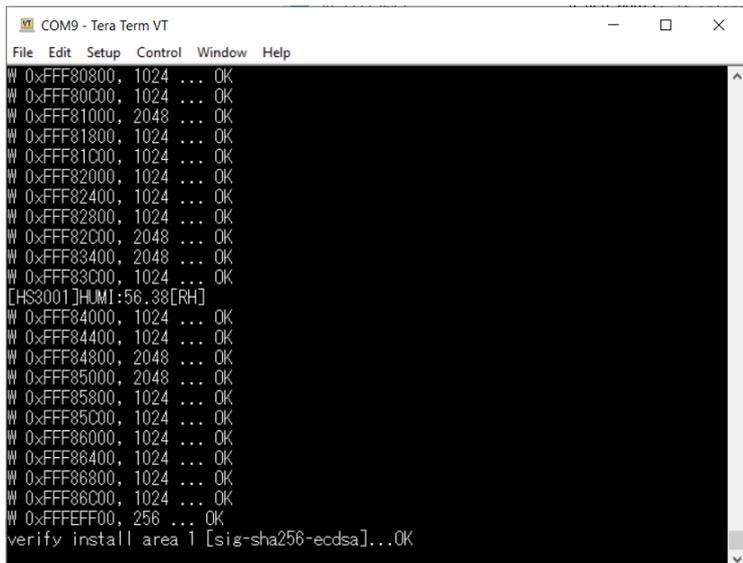
Leaf_01 Test NX_AZURE_IOT_ADU_AGENT_DRIVER_WRITE called.
Proxy driver firmware writing...
Receive data size 140
Remaining data size 1420
Proxy driver firmware 1K send Receive + Remaining start...
Proxy driver firmware 1K send Receive + Remaining end...

Remmaining data set
[INFO] Getting download data... 49688

Receive data driver_command 3
Receive data driver_firmware_size 0
Receive data driver_firmware_data_offset 49688
Receive data driver_firmware_data_size 656
2nd_01 Test driver called.

Leaf_01 Test NX_AZURE_IOT_ADU_AGENT_DRIVER_WRITE called.
Proxy driver firmware writing...
Receive data size 536
Remaining data size 656
Proxy driver firmware 1K send Receive + Remaining start...
```

図 7-25 アップデート実行中の CK-RX65N のログ

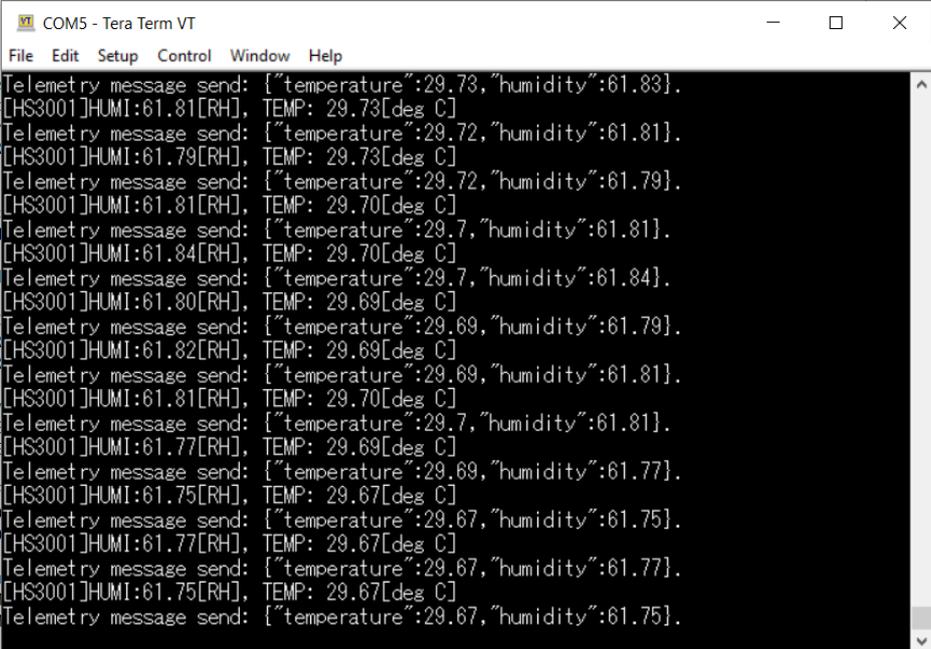


```
COM9 - Tera Term VT
File Edit Setup Control Window Help
W 0xFFFF80800, 1024 ... OK
W 0xFFFF80C00, 1024 ... OK
W 0xFFFF81000, 2048 ... OK
W 0xFFFF81800, 1024 ... OK
W 0xFFFF81C00, 1024 ... OK
W 0xFFFF82000, 1024 ... OK
W 0xFFFF82400, 1024 ... OK
W 0xFFFF82800, 1024 ... OK
W 0xFFFF82C00, 2048 ... OK
W 0xFFFF83400, 2048 ... OK
W 0xFFFF83C00, 1024 ... OK
[HHS3001]HUMI:56.38[RH]
W 0xFFFF84000, 1024 ... OK
W 0xFFFF84400, 1024 ... OK
W 0xFFFF84800, 2048 ... OK
W 0xFFFF85000, 2048 ... OK
W 0xFFFF85800, 1024 ... OK
W 0xFFFF85C00, 1024 ... OK
W 0xFFFF86000, 1024 ... OK
W 0xFFFF86400, 1024 ... OK
W 0xFFFF86800, 1024 ... OK
W 0xFFFF86C00, 1024 ... OK
W 0xFFFFEF00, 256 ... OK
verify install area 1 [sig-sha256-ecdsa]...OK
```

図 7-26 アップデート実行中の TB-RX660 のログ

7.4 セカンダリ OTA アップデート後の動作確認

図 7-27 にアップデート後の CK-RX65N のログ画面を示します。HS3001 センサで取得した湿度データに加えて、新たに温度データが表示されていることを確認できます。

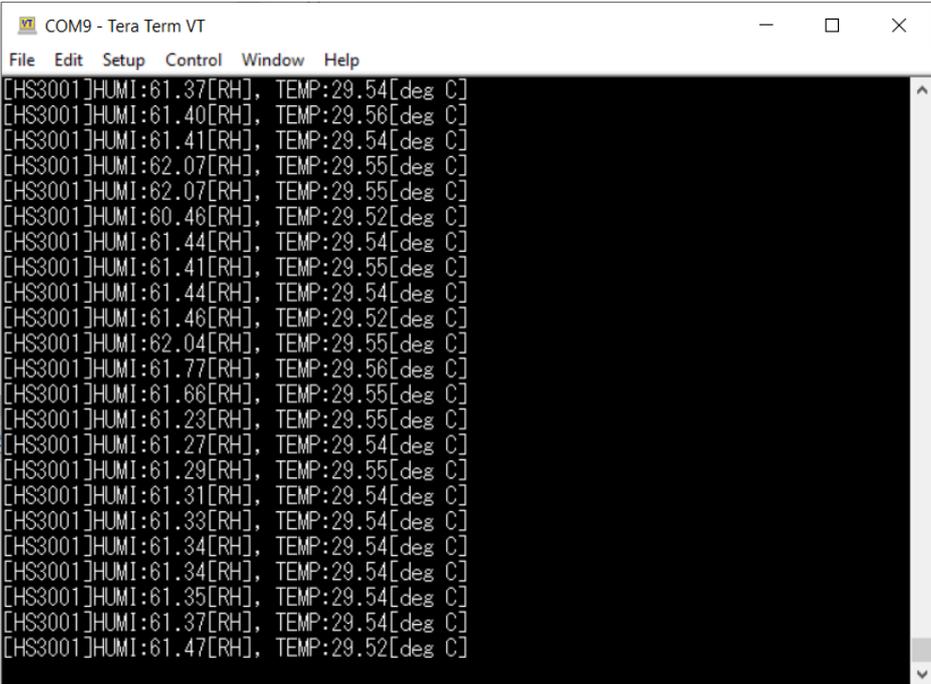


```
COM5 - Tera Term VT
File Edit Setup Control Window Help
Telemetry message send: [{"temperature":29.73,"humidity":61.83}.
[HS3001]HUMI:61.81[RH], TEMP: 29.73[deg C]
Telemetry message send: [{"temperature":29.72,"humidity":61.81}.
[HS3001]HUMI:61.79[RH], TEMP: 29.73[deg C]
Telemetry message send: [{"temperature":29.72,"humidity":61.79}.
[HS3001]HUMI:61.81[RH], TEMP: 29.70[deg C]
Telemetry message send: [{"temperature":29.7,"humidity":61.81}.
[HS3001]HUMI:61.84[RH], TEMP: 29.70[deg C]
Telemetry message send: [{"temperature":29.7,"humidity":61.84}.
[HS3001]HUMI:61.80[RH], TEMP: 29.69[deg C]
Telemetry message send: [{"temperature":29.69,"humidity":61.79}.
[HS3001]HUMI:61.82[RH], TEMP: 29.69[deg C]
Telemetry message send: [{"temperature":29.69,"humidity":61.81}.
[HS3001]HUMI:61.81[RH], TEMP: 29.70[deg C]
Telemetry message send: [{"temperature":29.7,"humidity":61.81}.
[HS3001]HUMI:61.77[RH], TEMP: 29.69[deg C]
Telemetry message send: [{"temperature":29.69,"humidity":61.77}.
[HS3001]HUMI:61.75[RH], TEMP: 29.67[deg C]
Telemetry message send: [{"temperature":29.67,"humidity":61.75}.
[HS3001]HUMI:61.77[RH], TEMP: 29.67[deg C]
Telemetry message send: [{"temperature":29.67,"humidity":61.77}.
[HS3001]HUMI:61.75[RH], TEMP: 29.67[deg C]
Telemetry message send: [{"temperature":29.67,"humidity":61.75}.
```

図 7-27 ファームウェア更新後の CK-RX65N のログ画面

次に、図 7-28 に TB-RX660 のログ画面を示します。TB-RX660 ファームウェア更新に成功すると、HS3001 センサで湿度と温度の測定データを取得します。

また、初期状態で点滅していた LED0 に加えて LED1 も点滅します。



```
COM9 - Tera Term VT
File Edit Setup Control Window Help
[HS3001]HUMI:61.37[RH], TEMP:29.54[deg C]
[HS3001]HUMI:61.40[RH], TEMP:29.56[deg C]
[HS3001]HUMI:61.41[RH], TEMP:29.54[deg C]
[HS3001]HUMI:62.07[RH], TEMP:29.55[deg C]
[HS3001]HUMI:62.07[RH], TEMP:29.55[deg C]
[HS3001]HUMI:60.46[RH], TEMP:29.52[deg C]
[HS3001]HUMI:61.44[RH], TEMP:29.54[deg C]
[HS3001]HUMI:61.41[RH], TEMP:29.55[deg C]
[HS3001]HUMI:61.44[RH], TEMP:29.54[deg C]
[HS3001]HUMI:61.46[RH], TEMP:29.52[deg C]
[HS3001]HUMI:62.04[RH], TEMP:29.55[deg C]
[HS3001]HUMI:61.77[RH], TEMP:29.56[deg C]
[HS3001]HUMI:61.66[RH], TEMP:29.55[deg C]
[HS3001]HUMI:61.23[RH], TEMP:29.55[deg C]
[HS3001]HUMI:61.27[RH], TEMP:29.54[deg C]
[HS3001]HUMI:61.29[RH], TEMP:29.55[deg C]
[HS3001]HUMI:61.31[RH], TEMP:29.54[deg C]
[HS3001]HUMI:61.33[RH], TEMP:29.54[deg C]
[HS3001]HUMI:61.34[RH], TEMP:29.54[deg C]
[HS3001]HUMI:61.34[RH], TEMP:29.54[deg C]
[HS3001]HUMI:61.35[RH], TEMP:29.54[deg C]
[HS3001]HUMI:61.37[RH], TEMP:29.54[deg C]
[HS3001]HUMI:61.47[RH], TEMP:29.52[deg C]
```

図 7-28 ファームウェア更新後の TB-RX660 のログ画面

最後に、図 7-29 に Azure IoT Explorer の画面を示します。HS3001 センサから取得した湿度と温度の測定データが表示されていることを確認できます。

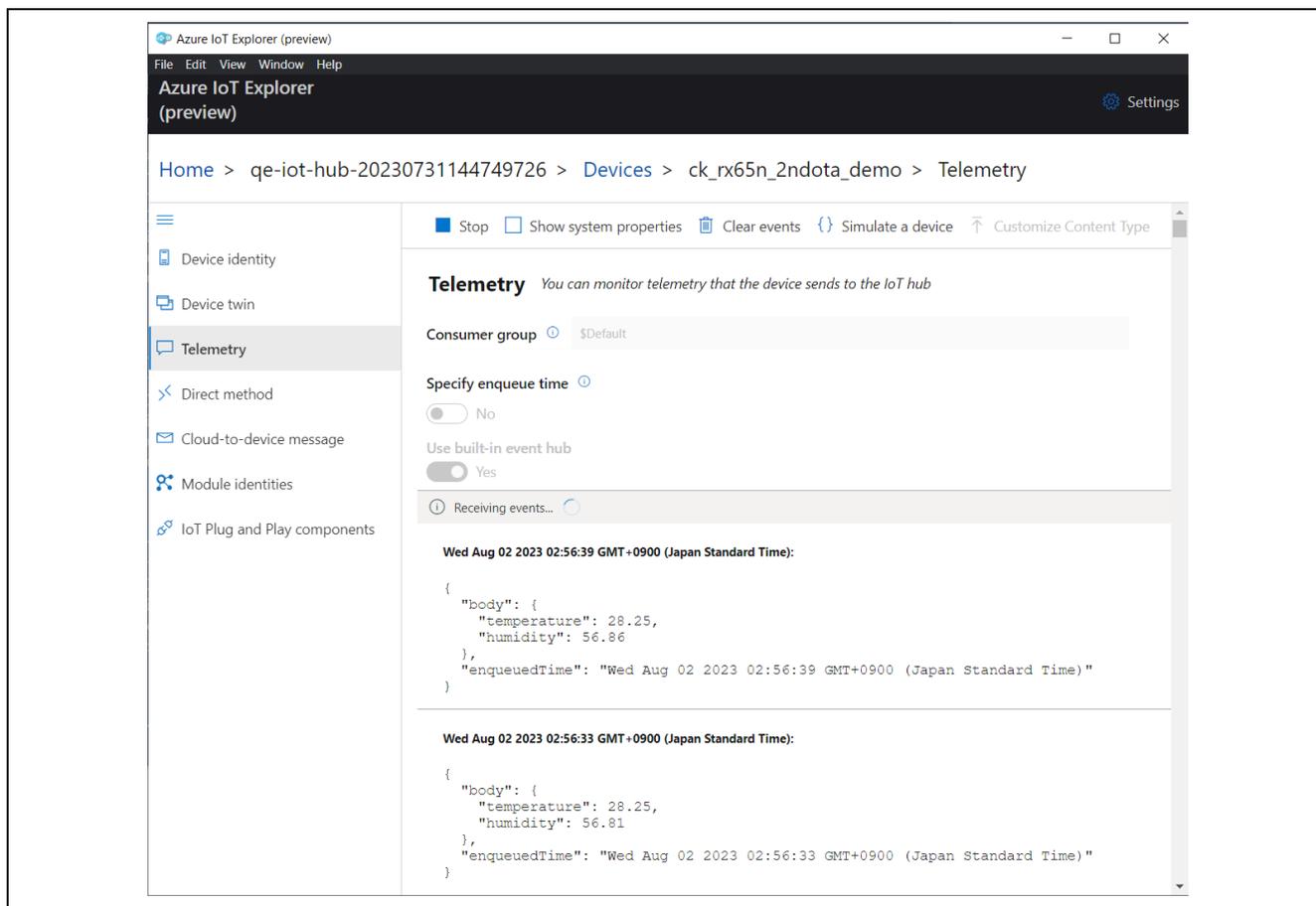


図 7-29 セカンダリ OTA アップデート後の Azure IoT Explorer の Telemetry 表示

デモ動作は以上となります。

7.5 Azure クラウドリソースのクリーンアップ

以上のデモ動作で作成した Azure のクラウドリソースを削除します。

Azure ポータルから Resource groups サービス画面を開きます。リソースグループが一覧表示されるため、デモで作成したリソースグループをクリックし、表示された画面で「Delete resource group」をクリックします。

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

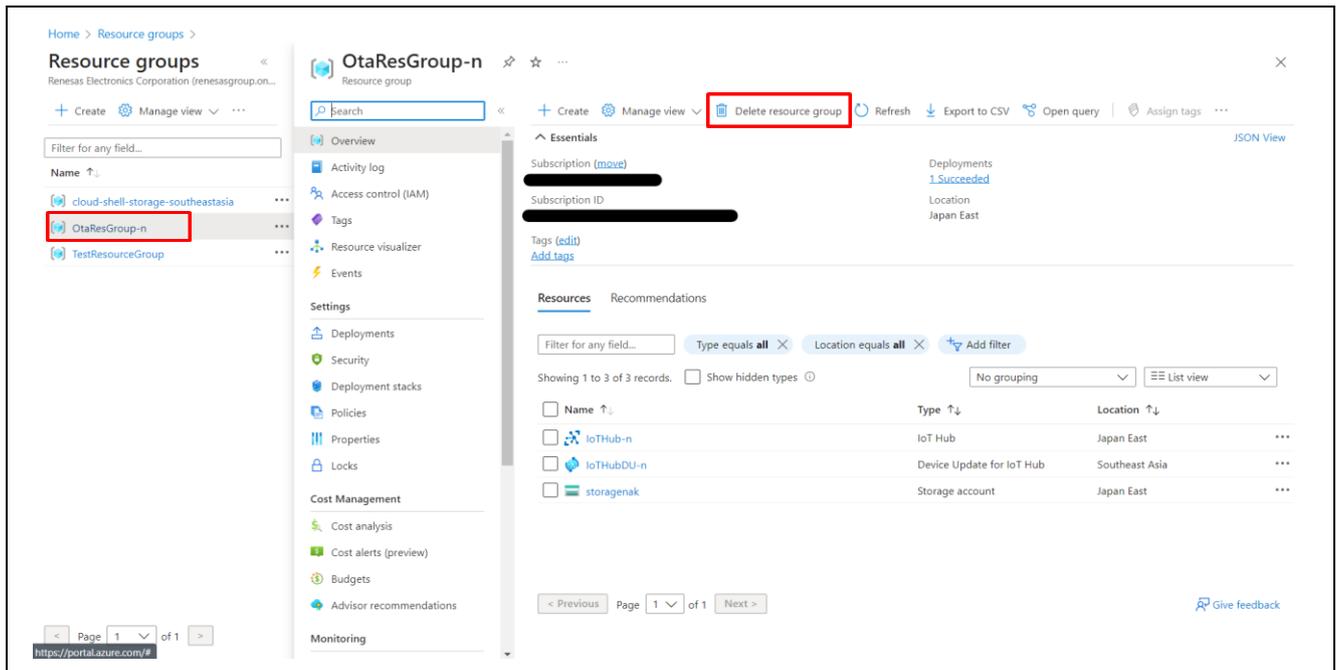


図 7-30 Azure クラウドリソースの削除

8. 備考

8.1 使用するソフトウェアのライセンス情報

以下のソフトウェアを使用しています。

- TinyCrypt Cryptographic Library
— ライセンス <https://github.com/intel/tinycrypt/blob/master/LICENSE>
- Azure RTOS
— ライセンス <https://github.com/azure-rtos/threadx/blob/master/LICENSED-HARDWARE.txt>

8.2 ファームウェアアップデートモジュール変更箇所

rx660_tb_demo_bootloader プロジェクトで使用しているファームウェアアップデートモジュールの r_fwup.c 内、37 行目の FWUP_COPY_BUF_SIZE を、RX660 のフラッシュ書き込み単位に合わせて 256U に変更しています。

```

32
34      + Macro definitions
36      #define FWUP_READ_BUF_SIZE                (128U)
37      #define FWUP_COPY_BUF_SIZE                (256U) // modified for RX660
38      #define FWUP_VERIFY_BUF_SIZE            (128U)
39      #define FWUP_WRITE_HEADER_BUF_SIZE      (128U)
40      #define FWUP_IMAGE_FLAG                  (0xFEU)
41      #define FWUP_HASH_BYTES                  (32U)

```

図 8-1 ファームウェアアップデートモジュール変更箇所

8.3 Leaf バージョン情報の制限事項

7.2.1.2(3)で生成する RSU ファイル名には以下の制限があります。

ファイル名 : leaf_update_firm_2.0.0.rsu の「2.0.0」は Leaf のファームウェアのバージョンを表しています。本サンプルコードではバージョンの各桁を 0~9 までに制限しています。そのため 2 桁以上の数字は使用しないでください。

誤った使用例 :

leaf_update_firm_10.0.0.rsu / leaf_update_firm_1.10.0.rsu / leaf_update_firm_1.0.10.rsu

以下のコマンドを変更することで出力される RSU ファイル名を変更できます。

```

> python .\image-gen.py -iup ..\HardwareDebug\rx660_tb_2ndota_demo.mot -
o ..\..\ck_rx65n_azure_2ndota_demo\tools\AzureDeviceUpdateScripts\leaf_update_firm_2.0.0 -
key .\keys\secp256r1.privatekey -ip .\RX660_Linear_Half_ImageGenerator_PRM_2ndota_demo.csv -vt
ecdsa

```

図 8-2 Renesas Image Generator コマンド変更箇所

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2023/08/31	—	初版

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、 $V_{IL}(\text{Max.})$ 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

- 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。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。