RX ファミリ
Microsoft Azure サービスを利用した OTA の実現方法

R01AN6928JJ0200
Rev.2.00
2024.01.12

Important Notice:

On November 21, 2023, Microsoft announced that they have decided to contribute Azure RTOS to Open Source

under the stewardship of the Eclipse foundation and Azure RTOS becomes Eclipse ThreadX.

For detailed information, please refer to the announcement titled at Microsoft Contributes Azure RTOS to Open Source.

The support strategy scheme for Eclipse ThreadX will be determined and communicated at a later date.

Microsoft will discontinue the Azure RTOS and Azure RTOS Middleware under the existing agreement LICENSED-HARDWARE.txt.

It's important to note that updates for Azure RTOS on these hardware will no longer be provided.

要旨

本書は Microsoft Azure において、IoT デバイスの Over-the-Air (OTA) 更新をデプロイできるようにする環境構築方法について解説します。
OTA は、Azure サービスの一つである Device Update for IoT Hub を利用して行います。本書では本機能を ADU と記載し、手順をガイドします。

また QE for OTA を使用することにより、ADU のプロジェクト構築に必要な工程を簡略化することも可能ですので参考にしてください。

なお、本ドキュメントに記載されている情報は予告なく変更される場合がありますことをご了承ください。

対象デバイス

・CK-RX65N（Ether）
・Renesas Starter Kit+ for RX65N-2MB（Ether）
・RX65N Cloud Kit（Wi-Fi）
・RX72N Envision Kit（Ether）
・RX671 RSK（Ether）

（※）本ドキュメントは CK-RX65N を例として記載しております。

使用する開発環境

統合開発環境（IDE）	e2 studio 2024-01
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 8.3.0.202305-GNURXGCC
ドライバパッケージ(RDP)	RX Driver Package V1.39
ファームウェアアップデートモジュール	Firmware Update module V1.06 Firmware Update module V2.01(※)
Azure RTOS	6.2.1_rel-rx-2.0.0
フラッシュ書き込みツール	Renesas Flash Programmer V3.13
MOT ファイル変換ツール(※)	Renesas Image Generator v3.03 （サンプルプロジェクトに同梱）
	Renesas Secure Flash Programmer (RX MCUs mot file converter 2.0.2) （別途インストール方法を記載）
Python 実行環境	Python 3.12.0（別途インストール方法を記載）
鍵生成ツール	Win32/Win64 OpenSSL v3.1.3 Light（別途インストール方法を記載）

(※)本書でガイドするサンプルプロジェクトは対応するターゲットボードとファームウェアアップデートモジュールのバージョンの組み合わせが存在します。また、ファームウェアアップデートモジュールのバージョンによって使用する MOT ファイル変換ツールが異なります。

これらの組み合わせは以下の表を参照してください。

表 1-1 ターゲットボードとファームウェアアップデートモジュールの組み合わせ

対応するターゲットボード	Firmware Update module	MOT ファイル変換ツール
CK-RX65N	v2.01	Renesas Image Generator
Renesas Starter Kit+ for RX65N-2MB RX65N Cloud Kit RX72N Envision Kit RX671 RSK	v1.06	Renesas Secure Flash Programmer

※本書ではファームウェアアップデートモジュールの v2.xx を v2、v1.xx を v1 と呼称します。

目次

1. ADU のためのメモリ配置と動作 5
1.1 ファームウェアのメモリ配置 5
1.1.1 ファームウェアアップデートモジュール v1 での動作内容 5
1.1.2 ファームウェアアップデートモジュール v2 での動作内容 7
1.2 データフラッシュの配置 8

2. サンプルプロジェクトの構築 9
2.1 ワークスペースの作成 9
2.2 サンプルプロジェクトの生成 9
2.2.1 ADU サンプルプロジェクトの新規作成 9
2.2.2 Bootloader サンプルプロジェクトの新規作成 14
2.3 プロジェクトの設定変更 15
2.3.1 バックグラウンドオペレーション（BGO）モードについて 15
2.3.2 コンポーネントの組み込み 15
2.3.3 ビルド時のコード生成の設定 19
2.4 鍵情報の作成 21
2.4.1 OpenSSL のインストール 21
2.4.2 OpenSSL での ECC 用鍵ペア生成 21
2.4.3 公開鍵の入力 23
2.5 bootloader プロジェクトのビルド 23
2.6 接続情報のマクロ設定 24
2.7 初期ファームウェアのバージョン確認 24
2.8 adu_sample のビルド（初期ファームウェア用） 24
2.9 初期ファームウェアの作成 25
2.9.1 ファームウェアアップデートモジュール v1 を使用する場合 25
2.9.2 ファームウェアアップデートモジュール v2 を使用する場合を 28
2.10 フラッシュ書き込みツールのインストール 30
2.11 初期ファームウェアの書き込み 30
2.12 初期ファームウェアの実行 32
2.13 更新ファームウェアのコード変更 33
2.14 adu_sample のビルド（更新ファームウェア用） 33
2.15 更新ファームウェアの作成 34
2.15.1 ファームウェアアップデートモジュール v1 を使用する場合 34
2.15.2 ファームウェアアップデートモジュール v2 を使用する場合を 36
2.15.3 ブートローダ v1 の環境でファームウェアアップデートモジュール v2 を使用する場合 37

3. Microsoft Azure ポータルでの操作 39
3.1 IoT Hub とデバイスの登録 39
3.2 デバイス更新の作成 39
3.3 ストレージアカウント・コンテナーの作成 44

3.4 更新ファームウェアの準備 48
3.4.1 更新ファームウェアのビルド 48
3.4.2 マニフェストファイルの作成 48
3.5 更新ファームウェアのストレージコンテナーへのアップロード 49
3.6 更新ファームウェアの登録 51
3.7 ADU グループの作成 54
3.8 ファームウェアの更新 55
3.8.1 ターゲットボードの実行 55
3.8.2 更新ファームウェアのデプロイ 55

4. Appendix 59
4.1 Azure ADU とサンプルプロジェクトのコマンド制御 59
4.2 ファームウェアのデバッグ手法 60
4.2.1 初期ファームウェアのデバッグ方法 60
4.2.2 更新ファームウェアのデバッグ方法 63

5. 改定履歴 66

1. ADU のためのメモリ配置と動作

1.1 ファームウェアのメモリ配置

RX65N の ADU サンプルプロジェクトは初期ファームウェア、及び更新ファームウェアは 1M バイトの領域を確保します。
ファームウェアアップデートモジュール v1 と v2 では基本構造は共通ですが、一部 ADU の処理過程で動作が異なる部分があります。

以下にメモリ配置と動作内容について説明します。

1.1.1 ファームウェアアップデートモジュール v1 での動作内容

以下にファームウェアアップデートモジュール v1 での動作の概略図を示します。

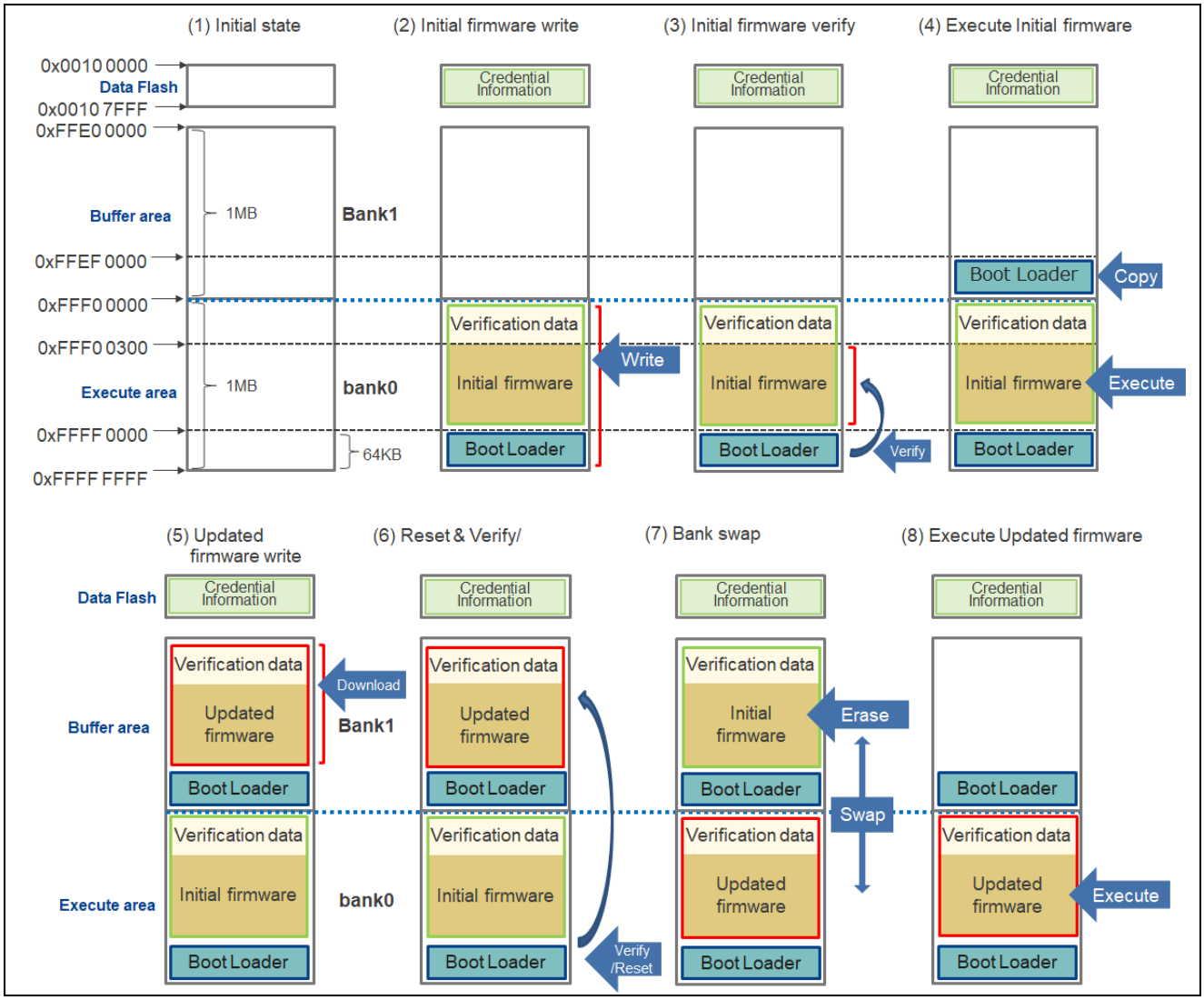

図 1-1 ADU のためのメモリ配置(v1)

(1) 初期状態として初期ファームウェアと更新ファームウェアにように 1M バイトの領域を確保します。
また、64K バイトはセキュアブートローダの領域とします。
メモリの各データのアドレスは以下のようになります。

0xFFF0 0000～0xFFF0 02FF :検証データ（Verification Data）
0xFFF0 0300～0xFFFE FFFF :ファームウェア(Firmware)
0xFFFF 0000～0xFFFF FFFF :セキュアブートローダ（Boot Loader）

(2) 初期ファームウェアとセキュアブートローダを、ルネサスフラッシュプログラマを使用して実行エリアに書き込みを行います。
また、データフラッシュへは資格情報（Credential Information：IoT Hub への接続情報等）が書き込まれます。

(3) セキュアブートローダが実行されると、ブートローダは 0xFFF00000～0xFFF002FF に書き込まれている検証データ（Verification data）を使用し、書き込まれている初期ファームウェアが改ざんされていないかの検証を行います。

(4) 検証が成功すると、初期ファームウェアが実行され、実行後にセキュアブートローダをバッファエリアにコピーします。

(5) Azure より更新ファームウェアをダウンロードし、バッファ領域（0xFFE0000～0xFFEEFFFF）へ書き込みを行います。

(6) ダウンロード完了後、システムのリセットを行います。リセット後はブートローダが Buffer Area にある検証データを使用し、更新ファームウェアが改ざんされていないか検証を行います。

(7) 検証が成功するとバンクスワップ機能を使用し、初期ファームウェアと更新ファームウェアのメモリ領域を切り替えます。バンクスワップ後ブートローダが Buffer Area の初期ファームウェアを削除します。

(8) Execute Area の更新ファームウェアが実行されます。
バンクスワップ機能を使用しているため、ファームウェア更新後もアプリケーションが参照するアドレスは共通で使用することができます。

1.1.2 ファームウェアアップデートモジュール v2 での動作内容

以下にファームウェアアップデートモジュール v2 での動作の概略図を示します。

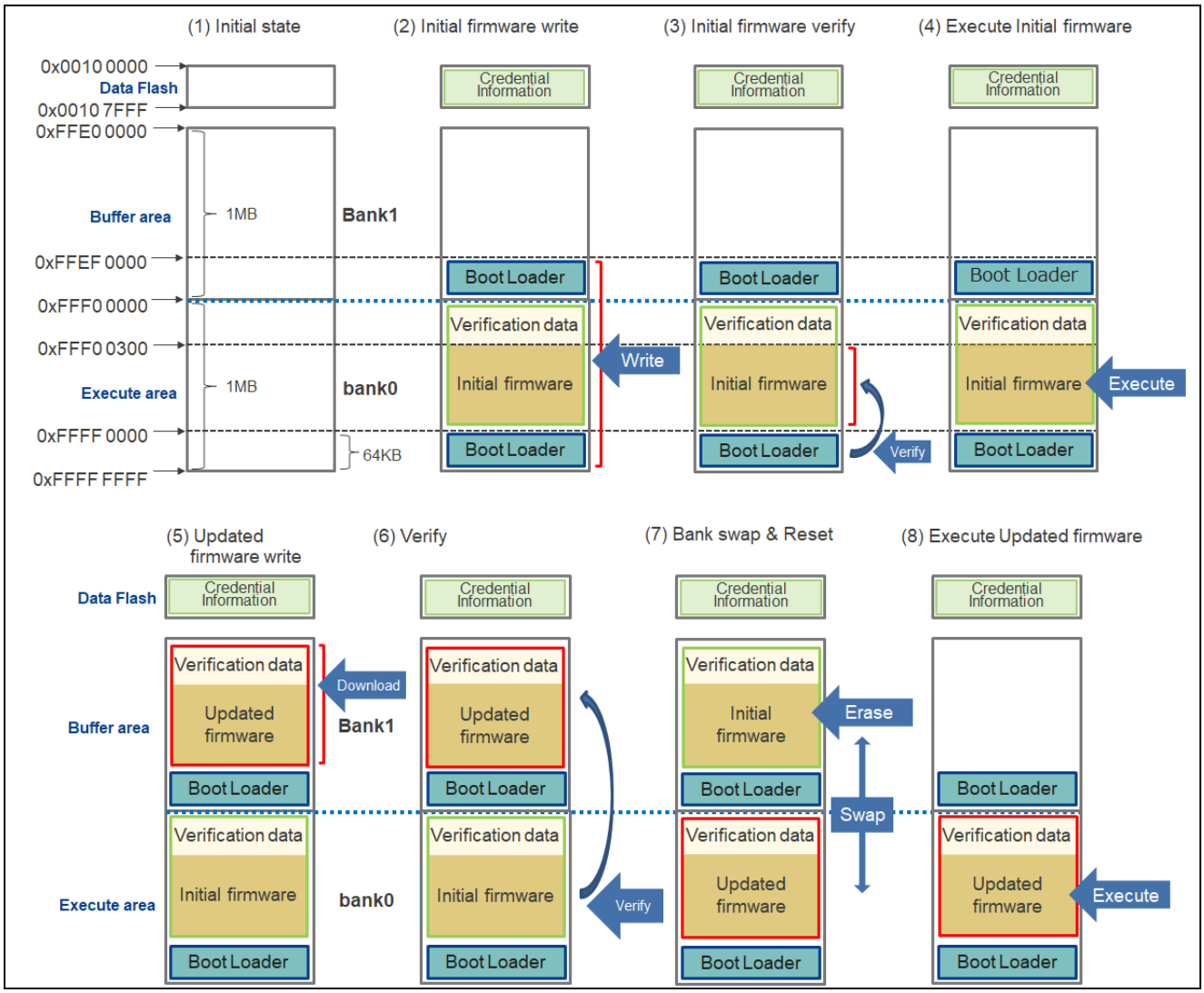

図 1-2 ADU のためのメモリ配置(v2)

(1) 初期状態として初期ファームウェアと更新ファームウェアにように 1M バイトの領域を確保します。
また、64K バイトはセキュアブートローダの領域とします。
メモリの各データのアドレスは以下のようになります。

0xFFF0 0000～0xFFF0 02FF :検証データ(Verification Data)
0xFFF0 0300～0xFFFE FFFF :ファームウェア(Firmware)
0xFFFF 0000～0xFFFF FFFF :セキュアブートローダ(Boot Loader)

(2) 初期ファームウェアとセキュアブートローダを、ルネサスフラッシュプログラマを使用して実行エリアに書き込みを行います。
また、データフラッシュへは資格情報(Credential Information:IoT Hub への接続情報等)が書き込まれます。
この際、バッファエリアのブートローダも同時に書き込みを行います。

(3) セキュアブートローダが実行されると、ブートローダは 0xFFF00000～0xFFF002FF に書き込まれている検証データ(Verification data)を使用し、書き込まれている初期ファームウェアが改ざんされていないかの検証を行います。

(4) 検証が成功すると、初期ファームウェアが実行されます。

(5) Azure より更新ファームウェアをダウンロードし、バッファ領域（0xFFE0000～0xFFEEFFFF）へ書き込みを行います。

(6) ダウンロードが完了後、初期ファームウェアは Buffer Area にある検証データを使用し、更新ファームウェアが改ざんされていないか検証を行います。

(7) 検証が成功するとバンクスワップ機能を使用し、初期ファームウェアと更新ファームウェアのメモリ領域の切り替えとリセットを実行します。リセット後ブートローダが起動し、Buffer Area の初期ファームウェアを削除します。

(8) Execute Area の更新ファームウェアが実行されます。
バンクスワップ機能を使用しているため、ファームウェア更新後もアプリケーションが参照するアドレスは共通で使用することができます。

1.2 データフラッシュの配置

本サンプルプロジェクトでは、データフラッシュへ Azure 接続等に使用するために必要な各種資格情報（Credential Information）を書き込みます。
資格情報は、Azure の接続先情報や MAC アドレス等が書き込まれます。

また、この領域（0x00100000-0x00100EFF）はシステムで予約されています。
このため、ユーザーアプリケーションでデータフラッシュをご利用になる場合は 0x00100F00-0x00107FFF の領域を使用してください。

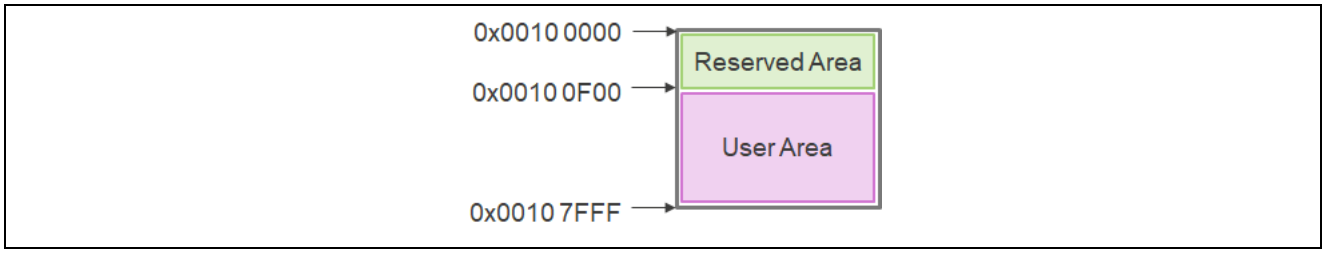

図 1-3 データフラッシュの配置

(※)本サンプルプロジェクトでは資格情報のデータフラッシュへの書き込みは、初期ファームウェアのフラッシュ書き込み時にのみ行われます。
このため、初期ファームウェア作成後はデータフラッシュの資格情報が変化する変更は行わないでください。変更が発生した場合は、初期ファームの書き換えを実施してください。

また、最適化の設定等のビルド方法の変更を行うことでデータフラッシュのメモリ配置が変化する場合がありますので、初期ファームウェアと更新ファームウェアは同一コンパイラ・同じ設定でビルドを実施してください。

2. サンプルプロジェクトの構築

本章では、ADU 実行するためのプロジェクト構築方法をガイドします。
セキュアブートローダを使用する ADU の動作は以下の 2 つのサンプルプロジェクトを使用します。

- Azure Device Update(ADU) sample project
- Secure bootloader sample project

本章の手順に沿って 2 つのプロジェクトを生成してください。
また、生成したプロジェクトは設定やメモリ配置、ソースコードの変更が必要となりますので変更方法についても記載します。

2.1 ワークスペースの作成

e² studio を起動して新しいワークスペースを作成してください。
ワークスペースやプロジェクトファイル名はなるべく短くするようにしてください。最下層のフルパスの長さが 256byte を超えるとビルド時にエラーが発生する場合があります。
またパスに日本語が存在するとエラーとなる場合がありますので、名前は英数字で入力してください。

【例】C:\workspace にワークスペースを作成する場合

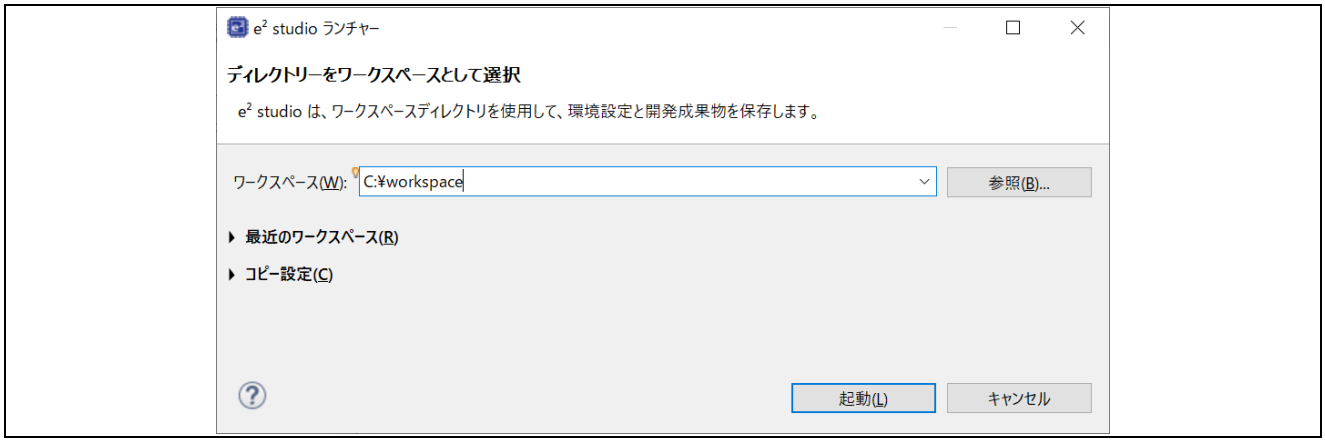

図 2-1 ワークスペース作成画面

2.2 サンプルプロジェクトの生成

2.2.1 ADU サンプルプロジェクトの新規作成

e² studio の起動後、[ファイル]メニュー ⇒ [新規] ⇒ [Renesas C/C++ Project] ⇒ [Renesas RX] を選択し、New C/C++ Project ダイアログを起ち上げてください。

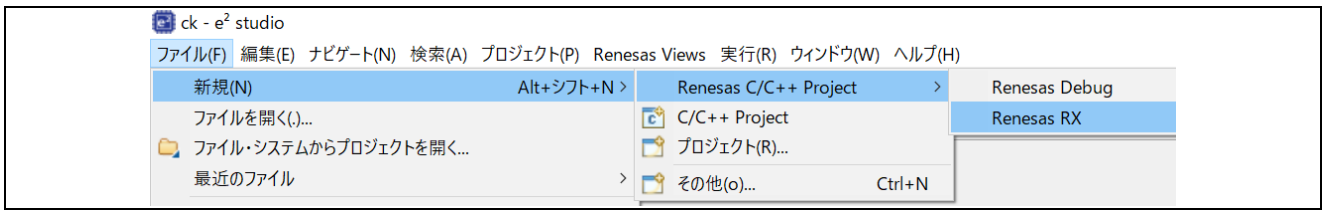

図 2-2 プロジェクトの新規作成メニュー

New C/C++ Project ダイアログで作成するプロジェクトの種類を選択します。ここでは、[All]を選択してから [Renesas CC-RX C/C++ Executable Project]を選択して、[次へ]ボタンを押下してください。
プロジェクトの種類選択（New Renesas CC-RX Executable Project）ダイアログが開かれます。
GCC をご利用になる場合は[GCC for Renesas RX C/C++ Execute Project]を選択してください。

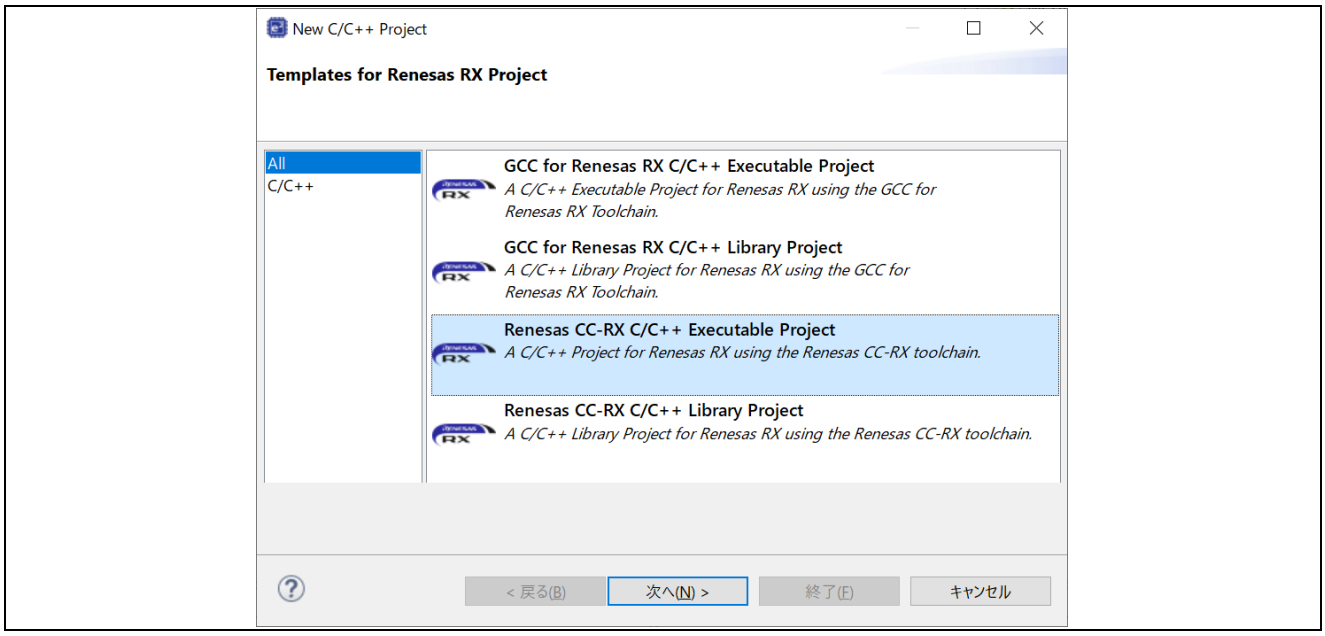

図 2-3 プロジェクトの種類選択画面

ここではプロジェクトの名称を設定します。プロジェクト名に”adu_sample”と入力し、[次へ]ボタンを押下してください。ツールチェイン・デバイスとデバッグ設定ダイアログが開かれます。

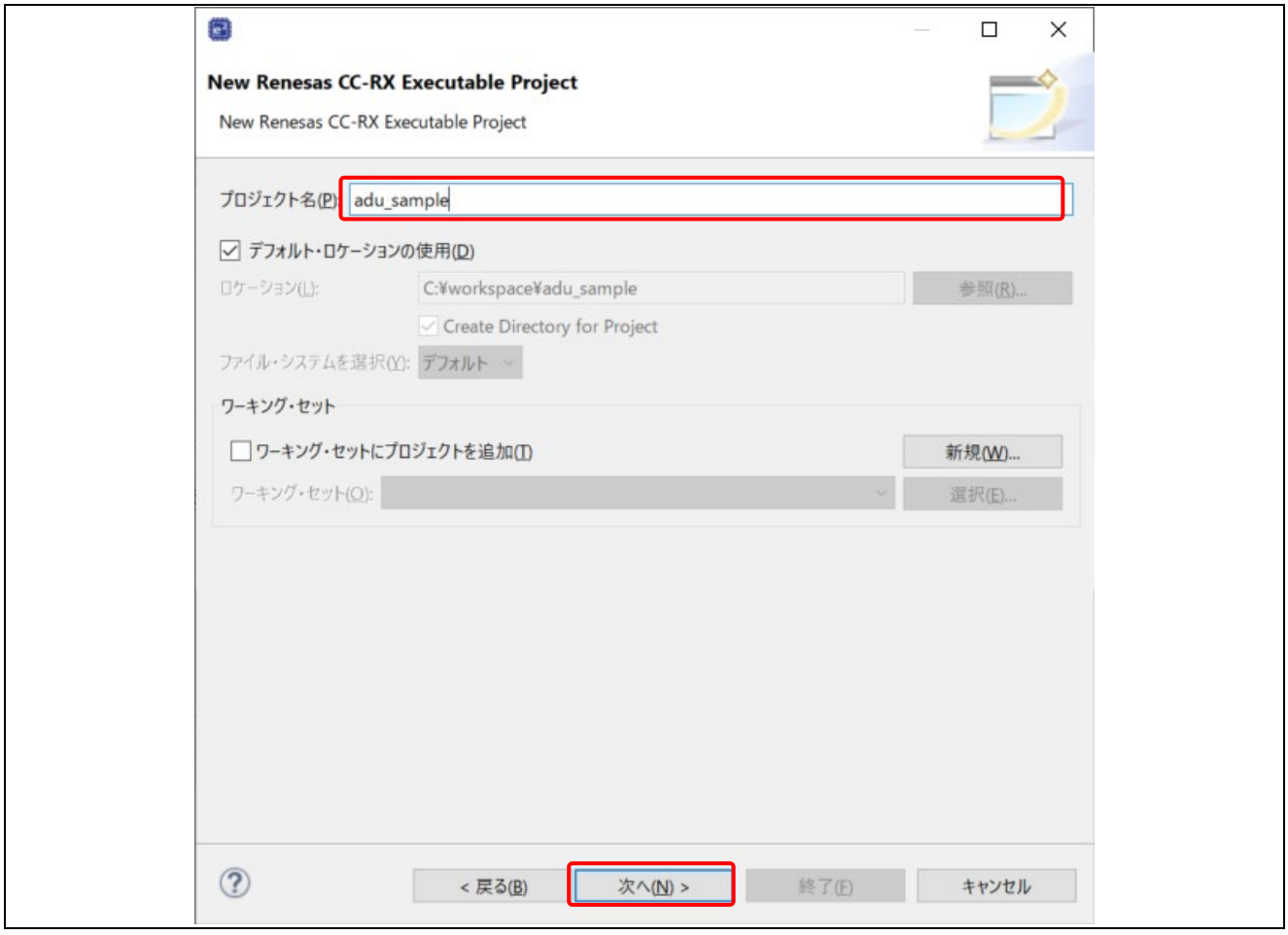

図 2-4 プロジェクトの名称設定画面

プロジェクトに使用するツールチェインとデバイスおよび、デバッグの設定を行います。
Toolchain はプロジェクトの種類で選択されたものが設定されます。ツールチェインのバージョンを変更したい場合は[ツールチェイン・バージョン]より必要なバージョンを選択してください。

RTOS は"Azure RTOS"を選択し、RTOS Version より"6.2.1_rel-rx-2.0.0"を選択してください。初めて e² studio をご利用になる場合やリストに必要なバージョンが表示されない場合は、[Managed RTOS Versions...]をクリックすると RTOS Module Download ダイアログが表示されるので、使用したいバージョンをチェックして[ダウンロード]ボタン押下してダウンロードしてください。

[Target Board]は"CK-RX65N(DUAL)"を選択してください(ターゲット・デバイスは自動でセットされます)。

ADU を行うためには、デバイスのコードフラッシュを 2 つのバンク領域として扱うデュアルモードに設定する必要があります。

すべての設定が完了したら[次へ]ボタンを押下してください。

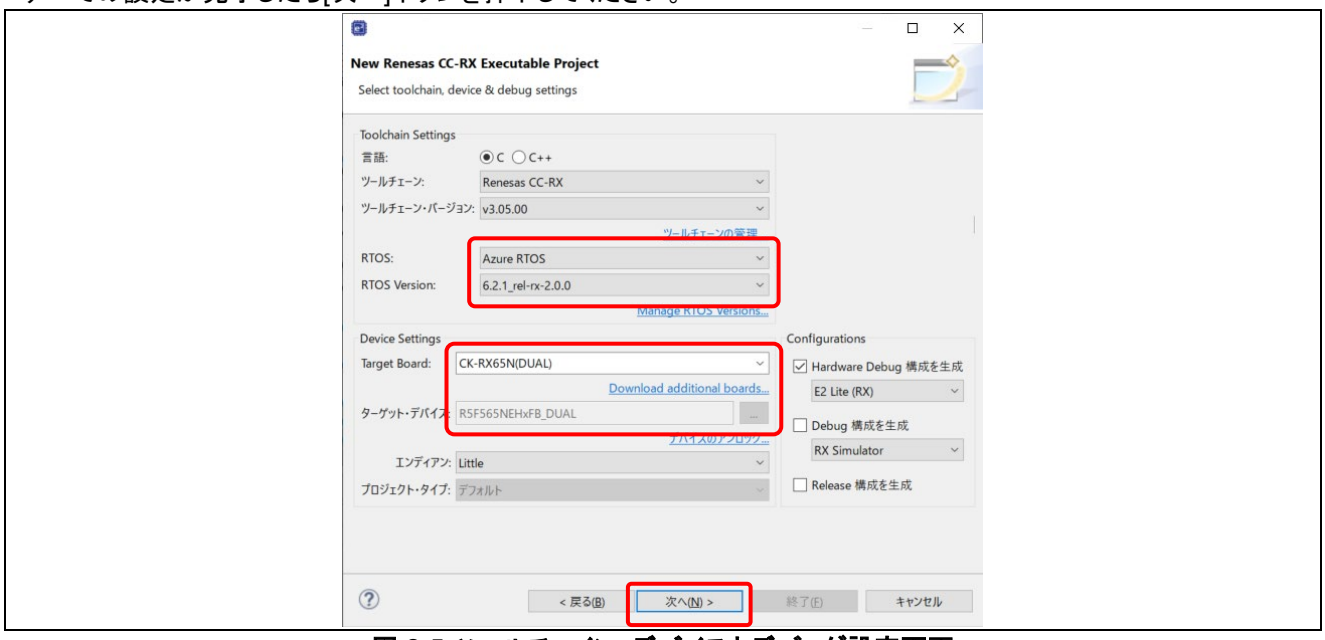

図 2-5 ツールチェイン・デバイスとデバッグ設定画面

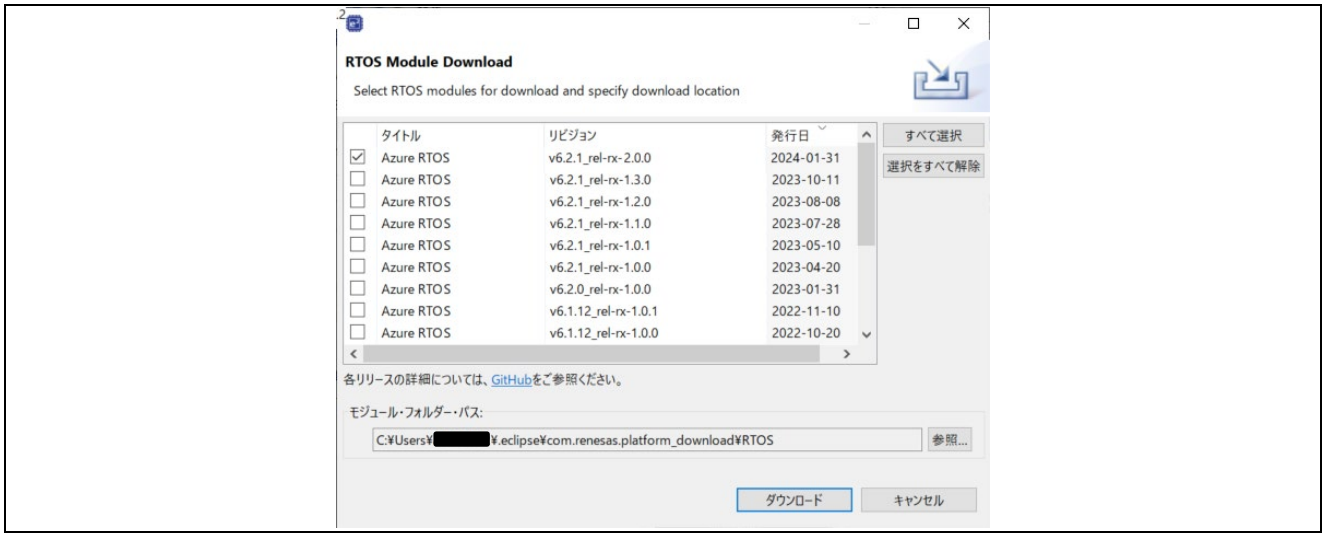

図 2-6 RTOS モジュールダウンロード画面

コーディングアシストツールの選択ダイアログが表示されますのでそのまま[次へ]ボタンを押下してください。

図 2-7 コーディングアシストツールの選択画面

RTOS プロジェクトの選択ダイアログに、サンプルプロジェクトの一覧が表示されます。
スクロールバーを操作し、リストより"Azure Device Update(ADU) sample project"を選択し、[次へ]ボタンを押下してください。

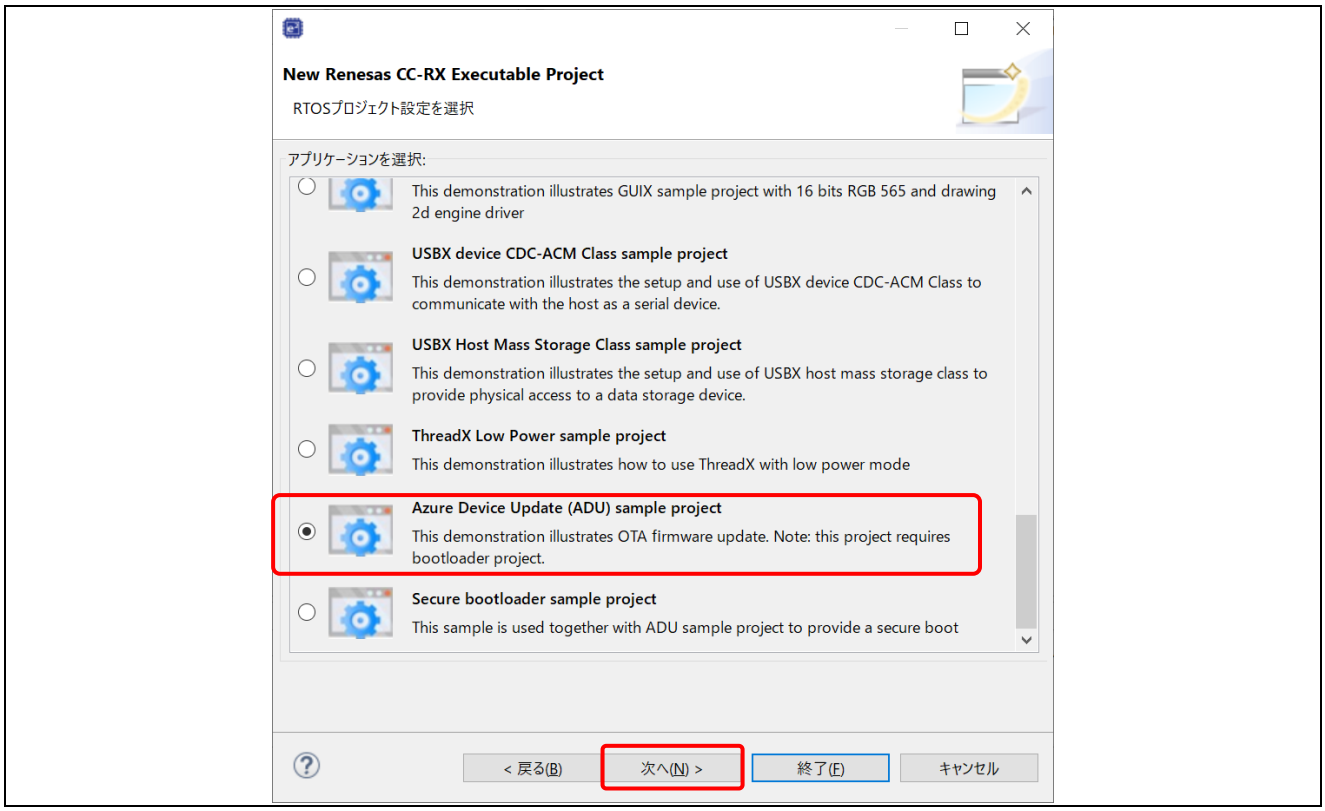

図 2-8 RTOS プロジェクト設定選択画面

生成ファイルの内容設定（Setting The Contents of Files to be Generated）ダイアログが表示されます。そのまま[次へ]ボタンを押下してください。

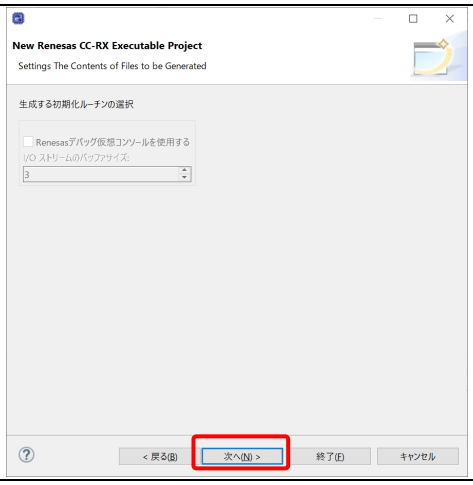

図 2-9 生成ファイルの内容設定画面

プロジェクト生成の終了画面が表示されます。問題なければ[終了]ボタンを押下してください。

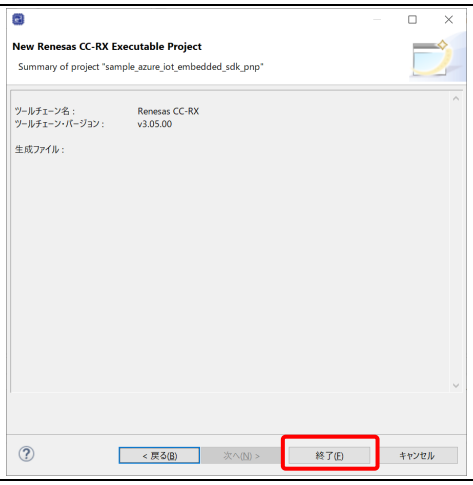

図 2-10 プロジェクト生成の終了画面

マーケットプレイスで利用可能なエディターのダイアログが表示されたら[キャンセル]ボタンを押下して閉じてください。

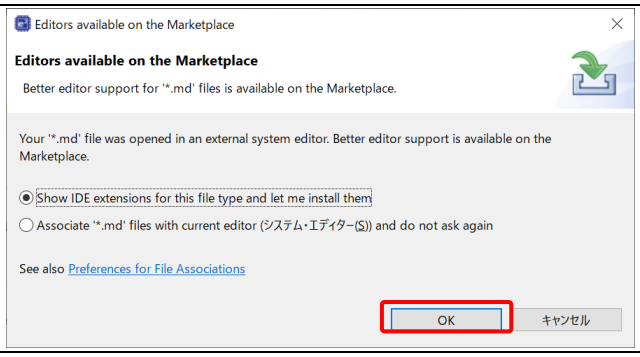

図 2-11 マーケットプレイスで利用可能なエディター画面

以下のように e² studio にプロジェクトが生成されます。
プロジェクト・エクスプローラーが表示されない場合は、画面右上のパースペクティブの[C/C++]を押下してから、[ウィンドウ] ⇒[ビューの表示] ⇒[プロジェクト・エクスプローラー]を選択してください。

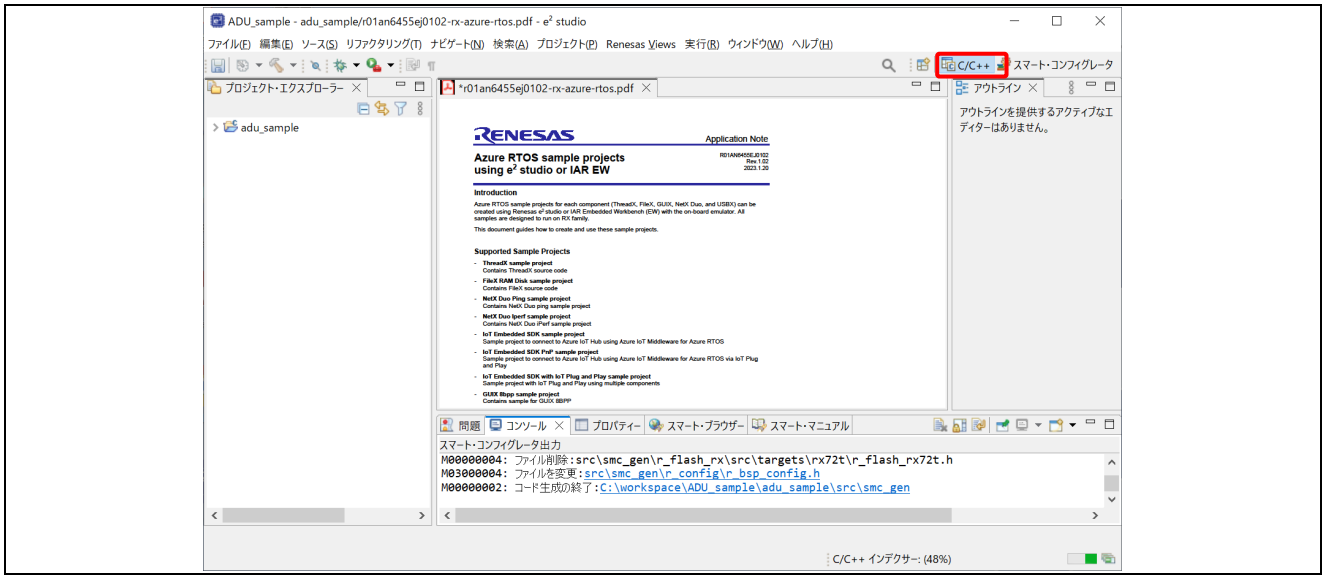

図 2-12 ADU サンプルプロジェクト生成後の画面

2.2.2 Bootloader サンプルプロジェクトの新規作成

ADU サンプルプロジェクトの新規作成の手順と同様に、Bootloader のサンプルプロジェクトを生成します。
基本的な手順は ADU サンプルプロジェクトと同様です。
プロジェクト名は" bootloader "を入力し、RTOS プロジェクト設定を選択では"Secure bootloader sample project" を選択してください。
その他の設定は ADU サンプルプロジェクトと同様です。

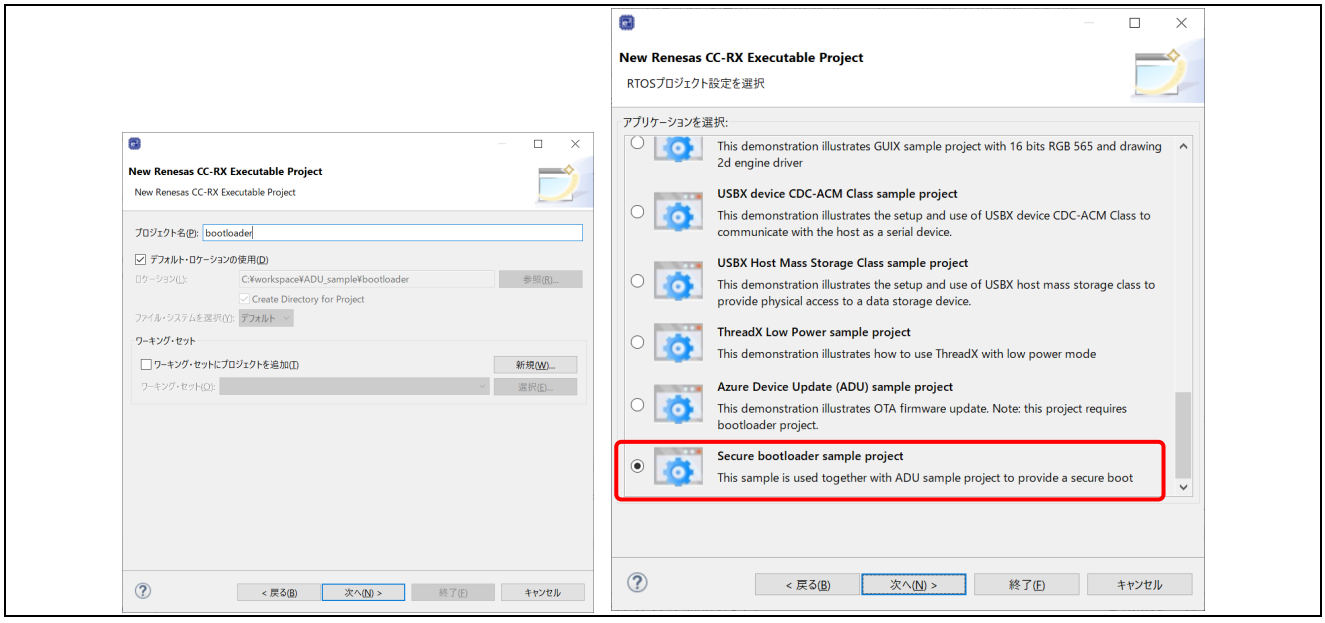

図 2-13 Bootloader プロジェクト生成の設定

2.3 プロジェクトの設定変更

生成したプロジェクトへ、ADU を実施するのに必要な設定の変更を行います。
なお、本節の手順は ADU サンプルプロジェクトと、Bootloader サンプルプロジェクトの両方に実施してください。
以下手順は主に Bootloader サンプルプロジェクトを例として説明します。

2.3.1 バックグラウンドオペレーション（BGO）モードについて

本ターゲットボードに搭載されている RX72N, RX671,RX65N グループは、フラッシュメモリの書き換え処理をバックグラウンドで行える BGO (Back Ground Operation)モードを搭載しています。

本書で説明するサンプルプロジェクトは、ファームウェアアップデートモジュール v1 では **BGO 有効**、v2 では **BGO 無効**にして動作します。(ファームウェアアップデート v.2 は CK-RX65N のみ対応)

BGO を有効にするとフラッシュメモリへの書き込み動作がノンブロッキングモードとなり、無効とするとブロッキングモードとなります。BGO モードの詳細については RX ファミリ フラッシュモジュール Firmware Integration Technology(R01AN2184)、RX65N グループ、RX651 グループ フラッシュメモリ ユーザーズマニュアル ハードウェア インタフェース編(R01UH0602)を参照してください。

2.3.2 コンポーネントの組み込み

初めてセットアップした e² studio の環境では一部コンポーネントが組み込まれていない場合があります。
ここでは ADU の必須コンポーネントのファームウェアアップデートモジュール（FIT）を例に説明します。

プロジェクト・エクスプローラーの bootloader プロジェクトツリーを開いて、"bootloader.scfg"をダブルクリックし、bootloader のスマート・コンフィグレータを開きます。
スマート・コンフィグレータの画面で、[コンポーネント]タブを選択すると"ソフトウェアコンポーネント設定"画面になります。

画面左に組み込みが必要なコンポーネントツリーが表示されます。ファームウェアアップデートモジュールはツリーの"r_fwup"に相当します。

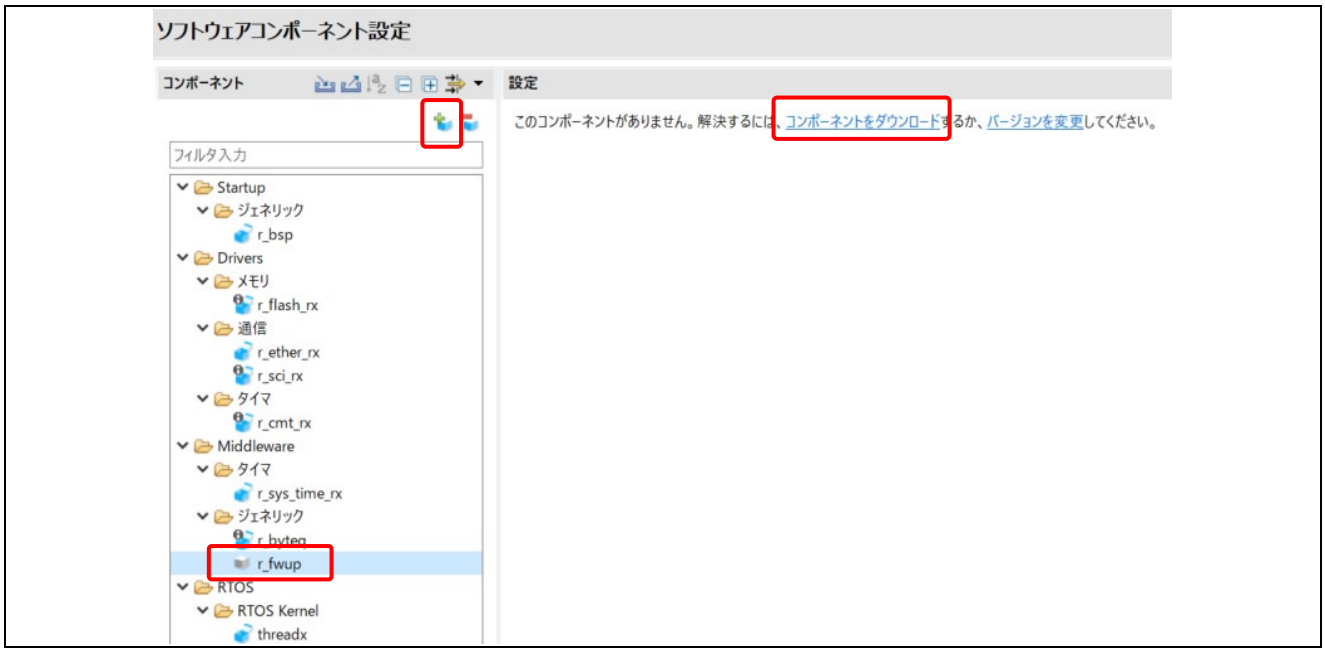

図 2-14 ソフトウェアコンポーネント設定画面

ここでコンポーネント名のアイコンがグレーになっている場合は、e² studio にコンポーネントがダウンロードされていない状態のため、次の手順でコンポーネントをダウンロードしてください。

また、グレーになっているコンポーネントのアイコンをクリックすることで表示される[コンポーネントをダウンロード]で不足コンポーネントを一括でダウンロードすることも可能です。

(1) コンポーネントツリー上の[+]ボタンを押下してください。ソフトウェアコンポーネントの選択画面が表示されるので、コンポーネント一覧から"FWUP Library"を選択し、[終了]ボタンを押下してください。コンポーネントツリーのアイコンが青に変化します（もともとツリーになかったコンポーネントの時はツリーに追加されます）。
FWUP Library は使用するターゲットボードで指定するバージョンが異なります。以下の表の通りのバージョンを指定してください。

表 2-1 ファームウェアアップデートモジュールのバージョン

ターゲットボード	FWUP Library （Firmware Update module）
CK-RX65N	v2.01
Renesas Starter Kit+ for RX65N-2MB RX65N Cloud Kit RX72N Envision Kit RX671 RSK	v1.06

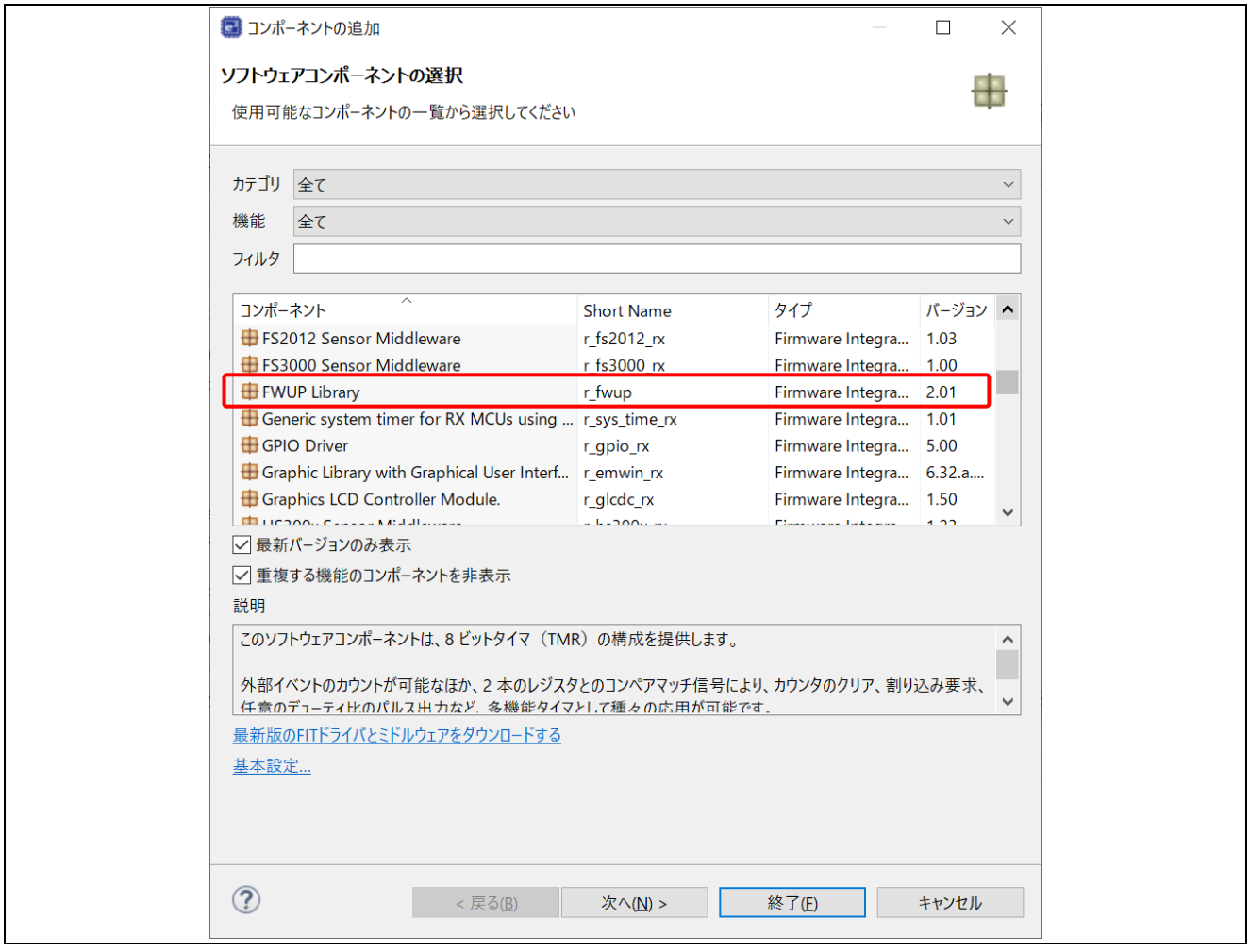

図 2-15 ソフトウェアコンポーネントの選択画面

最新バージョンのコンポーネントは上記ソフトウェアコンポーネントの一覧に表示されない場合があります。この場合は「①」または「②」の手順にてコンポーネントをダウンロードしてください。
ダウンロード後は上記画面で、コンポーネントの登録を行うことができるようになります。
ファームウェアアップデートモジュール V2 の場合、「①」の手順でコンポーネントが表示されない場合は「②」の手順を実施してください。

①スマート・コンフィグレータの FIT モジュールのダウンロード画面にてダウンロードする場合
コンポーネントツリー上の[＋]ボタンを押下してください。ソフトウェアコンポーネントの選択画面が表示されるので、画面下の[最新版の FIT ドライバとミドルウェアをダウンロードする]をクリックしてください。

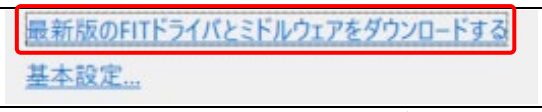

図 2-16 最新版の FIT ドライバのダウンロード

FIT モジュールのダウンロード画面が表示されるので、[RX Driver Package のみ表示する]のチェックを外してください。
FIT モジュールのリストから必要なモジュールにチェックを入れて[ダウンロード]ボタンをクリックしてください。ダウンロード後、”モジュール・フォルダ・パス”に指定されているフォルダに保存され、ソフトウェアコンポーネント設定画面でコンポーネントをサンプルプロジェクトへ組み込むことができるようになります。

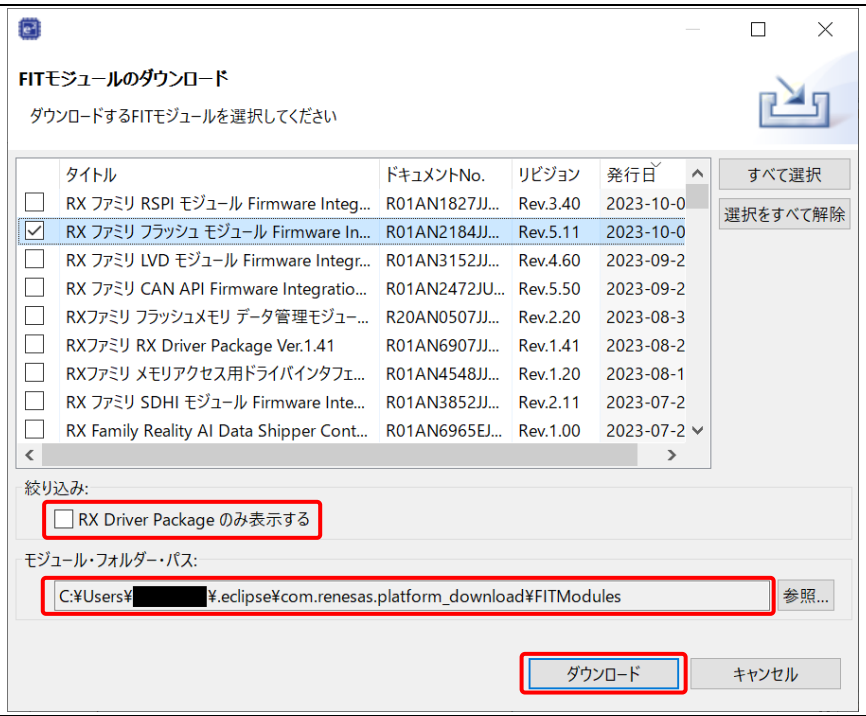

図 2-17 FIT モジュールのダウンロード画面

②Renesas WEB サイトからダウンロードして登録する場合

FIT モジュールのダウンロード画面に登録されていない最新モジュールは、ルネサスの WEB サイトからダウンロードして登録することができます。

以下の手順はファームウェアアップデートモジュール V2.01 の例を示します。

a)ルネサス Web サイトよりファームウェアアップデートモジュールのサンプルコードをダウンロードします。

b)ダウンロードした zip ファイルを展開して、”FITModules”フォルダ内の以下の 3 つのファイルを、①の FIT モジュールのダウンロード画面下部に表示されている、”モジュール・フォルダ・パス”で示されているフォルダへコピーします。
e² studio を再起動すると、ダウンロードした FIT モジュールが使用可能となります。

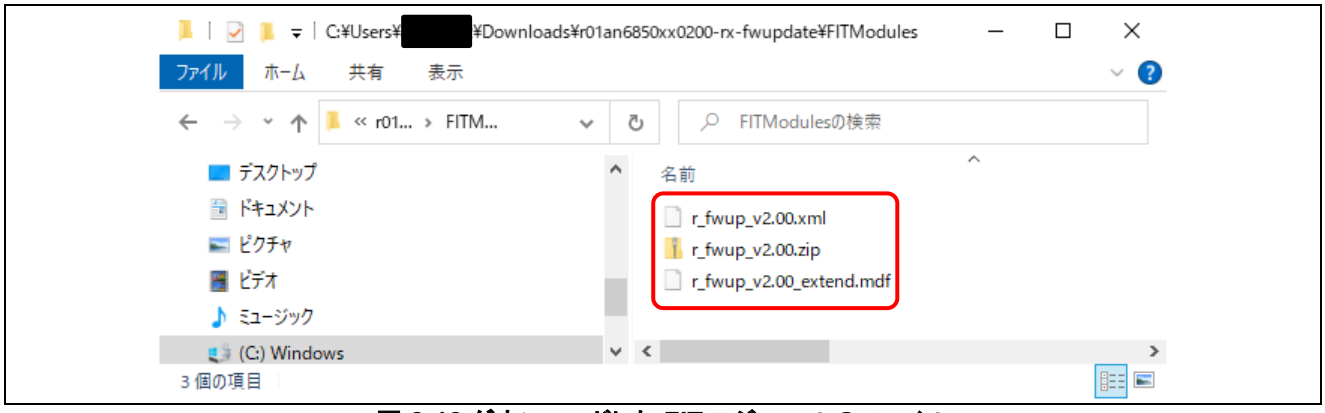

図 2-18 ダウンロードした FIT モジュールのファイル

(2)　コンポーネントの組み込みが完了したら、コンポーネントのコード生成(※)を行います。
ソフトウェアコンポーネント設定画面の右上の[コード生成]ボタンを押下してください。
生成されたコードはプロジェクトフォルダ下の"\src\smc_gen"フォルダに格納されます。

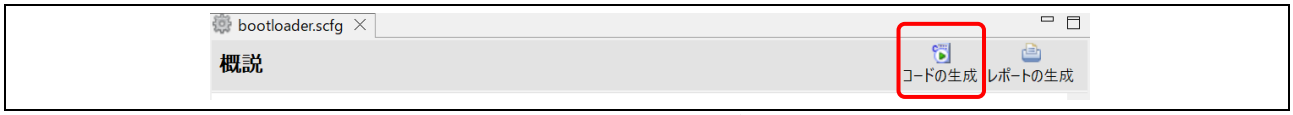

図 2-19 コード生成ボタン

また、その他組み込まれていないアイコンがグレーになっているコンポートネントがある場合は、同様の手順で組み込みを行ってください。

(※)スマート・コンフィグレータの設定を変更した後は最後に必ずコード生成を行ってください。

以上の手順は adu_sample に対しても実施してください。

2.3.3 ビルド時のコード生成の設定

本サンプルプロジェクトでは、コードの生成を行った際に追加している設定が失われる場合があります。
このため、プロジェクトをビルドまたはクリーンする際、すべてのソースファイルが再生成されることを回避するため、次の (1) ～ (3) の処理を実施してください。

(1) e² studio のプロジェクト・エクスプローラーで、bootloader を右クリックし、コンテキストメニューから[プロパティ]を選択してください。
本手順は adu_sample に対しても実施してください。

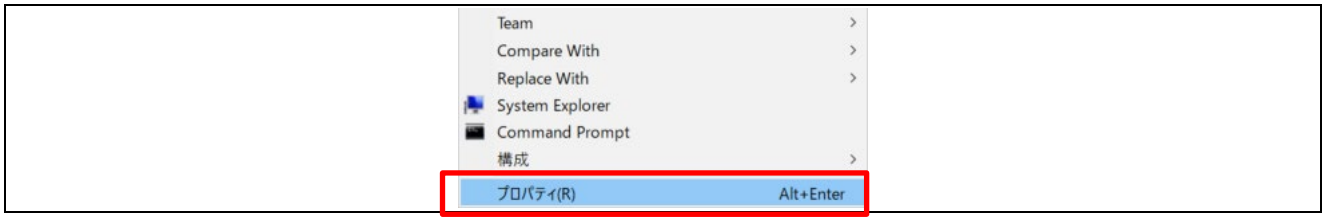

図 2-20 プロジェクトのコンテキストメニュー

(2) 「プロパティ」ダイアログのメニューの[ビルダー]を選択し、[SC Code Generation Builder]を選択後、[編集]をクリックしてください。ビルダーの構成画面が表示されます。

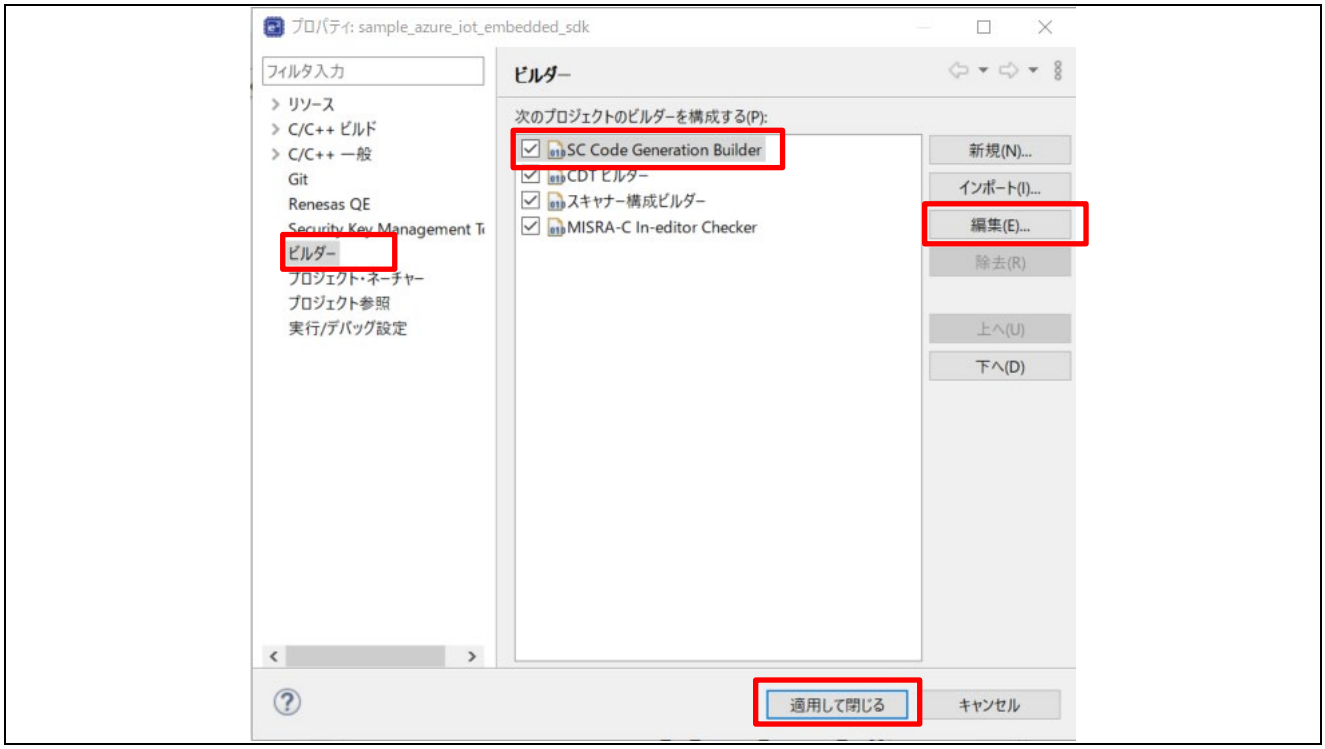

図 2-21 プロジェクトのプロパティダイアログ

(3) ビルダーの構成画面ですべてのチェックボックスのチェックを外して、[OK]をクリックしてください。その後、プロパティダイアログの[適用して閉じる]をクリックしてください。
本設定を実施することで、予期せぬコード生成を抑止することができます。(※)

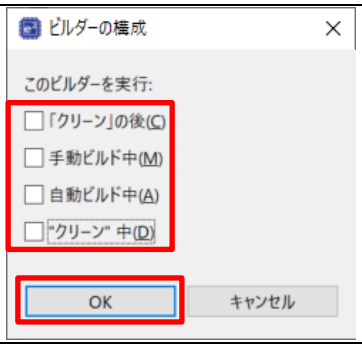

図 2-22 ビルダーの構成画面

(※) 実際にコード生成を行いたい場合は、スマート・コンフィグレータのコード生成ボタンをクリックして実施してください。

2.4 鍵情報の作成

サンプルプロジェクトに設定する鍵情報を生成します。
鍵情報の生成は、OpenSSL を使用してください。

2.4.1 OpenSSL のインストール

Win32/Win64 OpenSSL のダウンロードサイトにアクセスして、使用する OS に合わせて OpenSSL のインストーラーをダウンロードしてインストールしてください。
インストール後、Windows のシステムの環境変数の Path に OpenSSL のインストールフォルダを追加してください。
64bit 版の場合：C:\Program Files\OpenSSL-Win64\bin

システムの環境変数は Windows10 の場合、以下の操作で開くことができます。
[設定]⇒[システム]⇒[詳細情報]⇒[システムの詳細設定]⇒[システムのプロパティ]⇒[詳細設定]タブ⇒[環境変数]⇒[システムの環境変数]

OpenSSL のコマンドを実行する場合は、Windows スタートメニューから、「Win64 OpenSSL Command Prompt」アプリを実行してください。
(※)アプリケーション名は 64bit 版の時の例です。

2.4.2 OpenSSL での ECC 用鍵ペア生成

ファームウェア作成の際に、ECC 用の公開鍵や秘密鍵を使用します。これらは OpenSSL を使用して生成することができます。各種入力値は適宜ご自身のものに置き換えて、コマンドプロンプトで以下青文字のコマンドを実行してください。
コマンドには設定の入力を必要とする項目があるので、青文字の部分を入力してください。

なお、ECC 用の公開鍵と秘密鍵を生成するだけであれば、ステップ B, E, F だけで十分です。

A. CA 証明書の作成

```
openssl ecparam -genkey -name secp256r1 -out ca.key
using curve name prime256v1 instead of secp256r1

openssl req -x509 -sha256 -new -nodes -key ca.key -days 3650 -out ca.crt
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:JP
State or Province Name (full name) [Some-State]:Tokyo
Locality Name (eg, city) []:Kodaira
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Renesas Electronics
Organizational Unit Name (eg, section) []:Software Development Division
Common Name (e.g. server FQDN or YOUR name) []:Renesas Tarou
Email Address []:Tarou.Renesas@sample.com
```

B. 楕円曲線暗号(パラメータは secp256r1)の鍵ペアを生成

```
openssl ecparam -genkey -name secp256r1 -out secp256r1.keypair
using curve name prime256v1 instead of secp256r1
```

C. 鍵ペアの証明書を作成

```
openssl req -new -sha256 -key secp256r1.keypair > secp256r1.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:JP
State or Province Name (full name) [Some-State]:Tokyo
Locality Name (eg, city) []:Kodaira
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Renesas Electronics
Organizational Unit Name (eg, section) []:Software Development Division
Common Name (e.g. server FQDN or YOUR name) []:Renesas Tarou
Email Address []:Tarou.Renesas@sample.com
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

D. CA 証明書を使用して鍵ペアの証明書を作成

```
openssl x509 -req -sha256 -days 3650 -in secp256r1.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out secp256r1.crt
Certificate request self-signature ok
subject=/C=JP/ST=Tokyo/L=Kodaira/O=Renesas Electronics/OU=Software Development
Division/CN= Renesas Tarou/emailAddress= Tarou.Renesas@sample.com
```

E. 楕円曲線暗号(パラメータは secp256r1)の秘密鍵を抽出

```
openssl ec -in secp256r1.keypair -outform PEM -out secp256r1.privatekey
read EC key
writing EC key
```

F. 楕円曲線暗号(パラメータは secp256r1)の公開鍵を抽出

```
openssl ec -in secp256r1.keypair -outform PEM -pubout -out secp256r1.publickey
read EC key
writing EC key
```

2.4.3　公開鍵の入力

bootloader プロジェクトの\src\key\code_signer_public_key.h を開き、OpenSSL にて生成した secp256r1.publickey をテキストエディタで開いて、secp256r1.publickey の内容を CODE_SIGNENR_PUBLIC_KEY_PEM にコピーしてください。

各行は""で囲み、行の最後には \ が必要です。

また、ファームウェアアップデートモジュール v2 を使用している場合は、同様の鍵情報の入力作業を adu_sample プロジェクトの\src\key\code_signer_public_key.h に対しても行ってください。

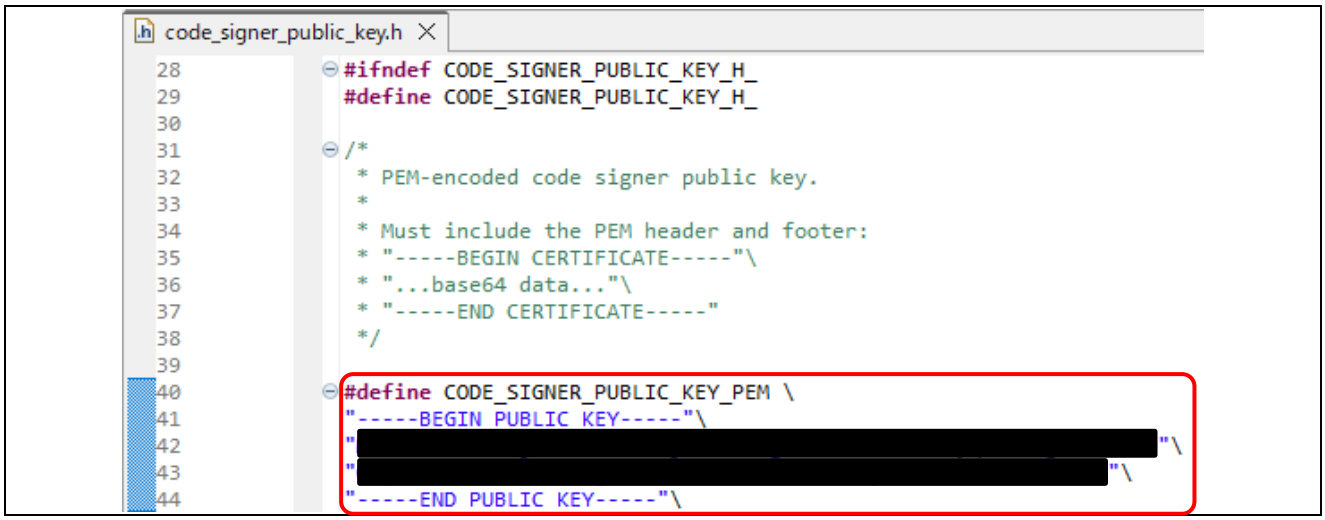

図 2-23 公開鍵の情報を設定

2.5　bootloader プロジェクトのビルド

bootloader プロジェクトをビルドし、bootloader.mot ファイルを生成してください。
MOT ファイルは以下フォルダに生成されます。
\bootloader\HardwareDebug/

2.6 接続情報のマクロ設定

adu_sample プロジェクトの\src\sample_config.h を開き HOST_NAME, DEVICE_ID, DEVICE_SYMMETRIC_KEY の値を設定してください。
各設定値は、"3.1IoT Hub とデバイスの登録"にて Azure ポータルで設定した IoT Hub とデバイスのパラメータとなりますのでお使いの設定を登録してください。

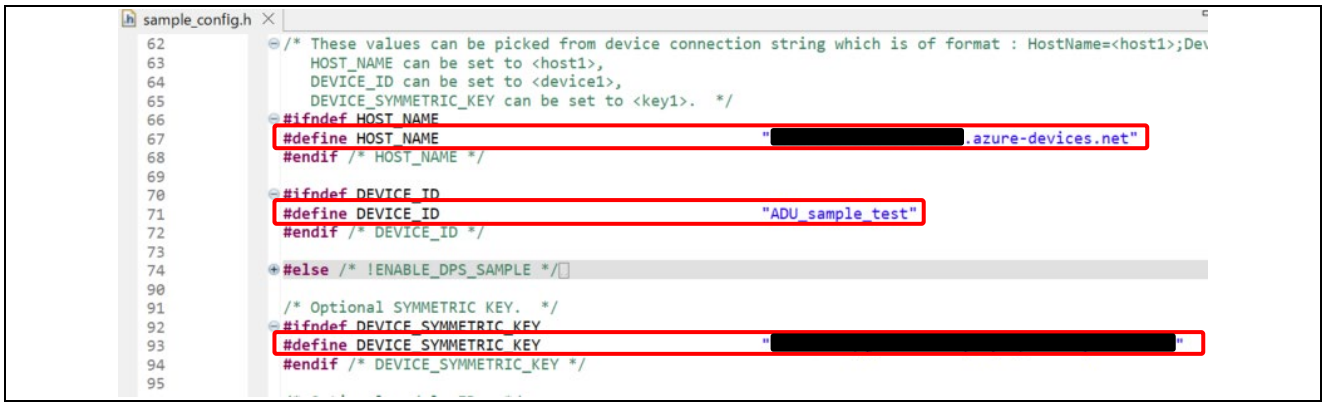

図 2-24 Azure 接続情報を設定

RX65N Cloud Kit をお使いの場合は Wi-Fi の接続設定を行う必要があります。
adu_sample プロジェクトの\src\hardware_setup.h を開き、WIFI_SSID と WIFI_PASSWORD の値を設定してください。
設定値は接続したい Wi-Fi アクセスポイントの SSID とパスワードを参照して登録してください。

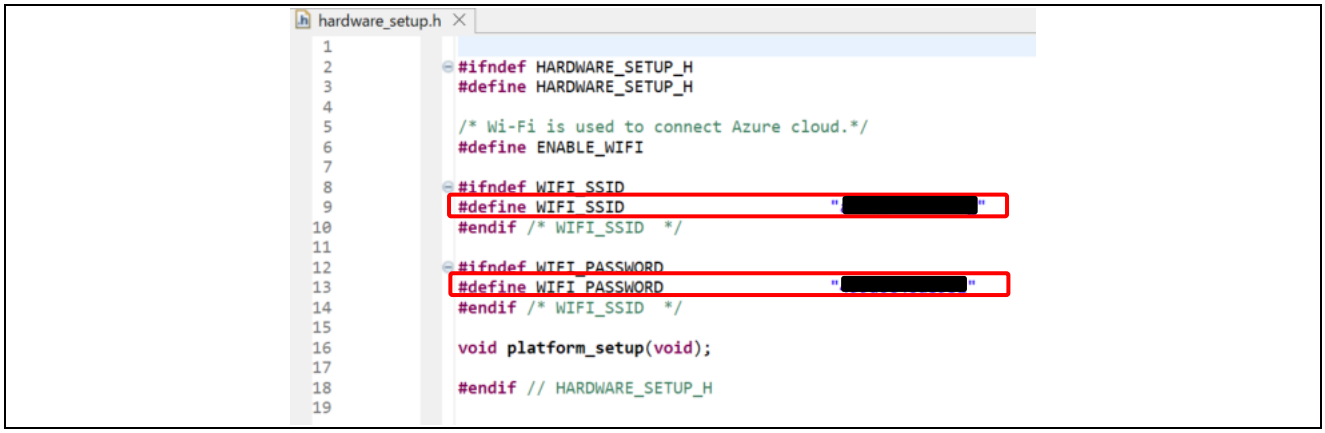

図 2-25 Wi-Fi の接続情報を設定

2.7 初期ファームウェアのバージョン確認

初期ファームウェア用のバージョンが"1.0.0"となっていることを確認します。
adu_sample プロジェクトの\src\sample_azure_iot_embedded_sdk_adu.c を開き、
SAMPLE_DEVICE_INSTALLED_CRITERIA が”1.0.0”となっていることを確認してください。

図 2-26 初期ファームウェアのバージョン

2.8 adu_sample のビルド（初期ファームウェア用）

adu_sample プロジェクトをビルドし、adu_sample.mot ファイルを生成してください。
MOT ファイルは以下フォルダに生成されます。
\adu_sample\HardwareDebug\

2.9 初期ファームウェアの作成

ターゲットボードにダウンロードする初期ファームウェアを作成します。

ファームウェアアップデート v1 と v2 ではファームウェアの作成手順と、使用する MOT ファイル変換ツールが異なります。お使いのファームウェアアップデートモジュールのバージョンに対応する手順を参照してください。

2.9.1 ファームウェアアップデートモジュール v1 を使用する場合

MOT ファイル変換ツールを使用して、bootloader と adu_sample のそれぞれの MOT ファイルを合成し、初期ファームウェアイメージを作成します。
MOT ファイル変換ツールには Renesas Secure Flash Programmer を使用します。

Renesas Secure Flash Programmer（RX MCUs mot file converter 2.0.2）を任意のフォルダにダウンロードして、ファイルを展開してください。

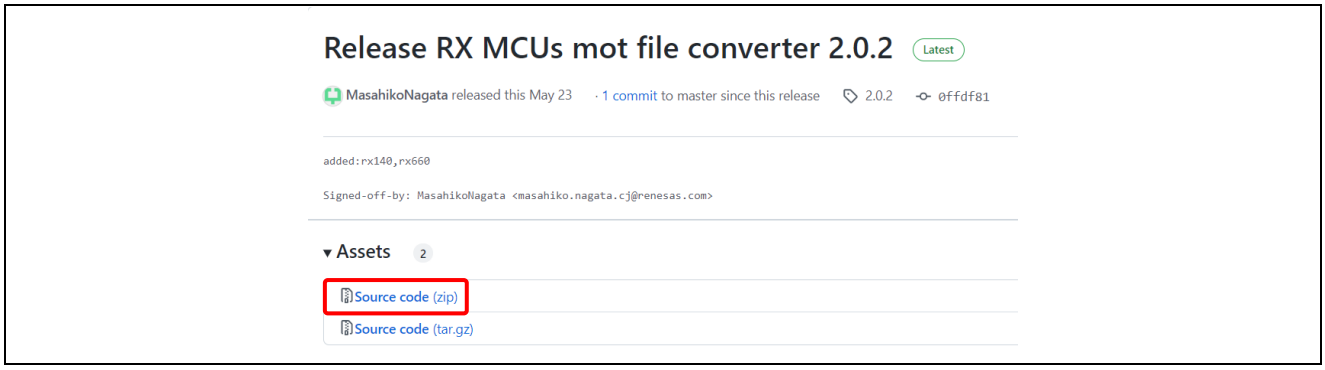

図 2-27 Renesas Secure Flash Programmer ダウンロード画面

ダウンロードが完了したら、\mot-file-converter-2.0.2 \Renesas Secure Flash Programmer\bin\Debug\Renesas Secure Flash Programmer.exe をダブルクリックで起動してください。

Renesas Secure Flash Programmer を起動すると以下のような画面が表示されます。

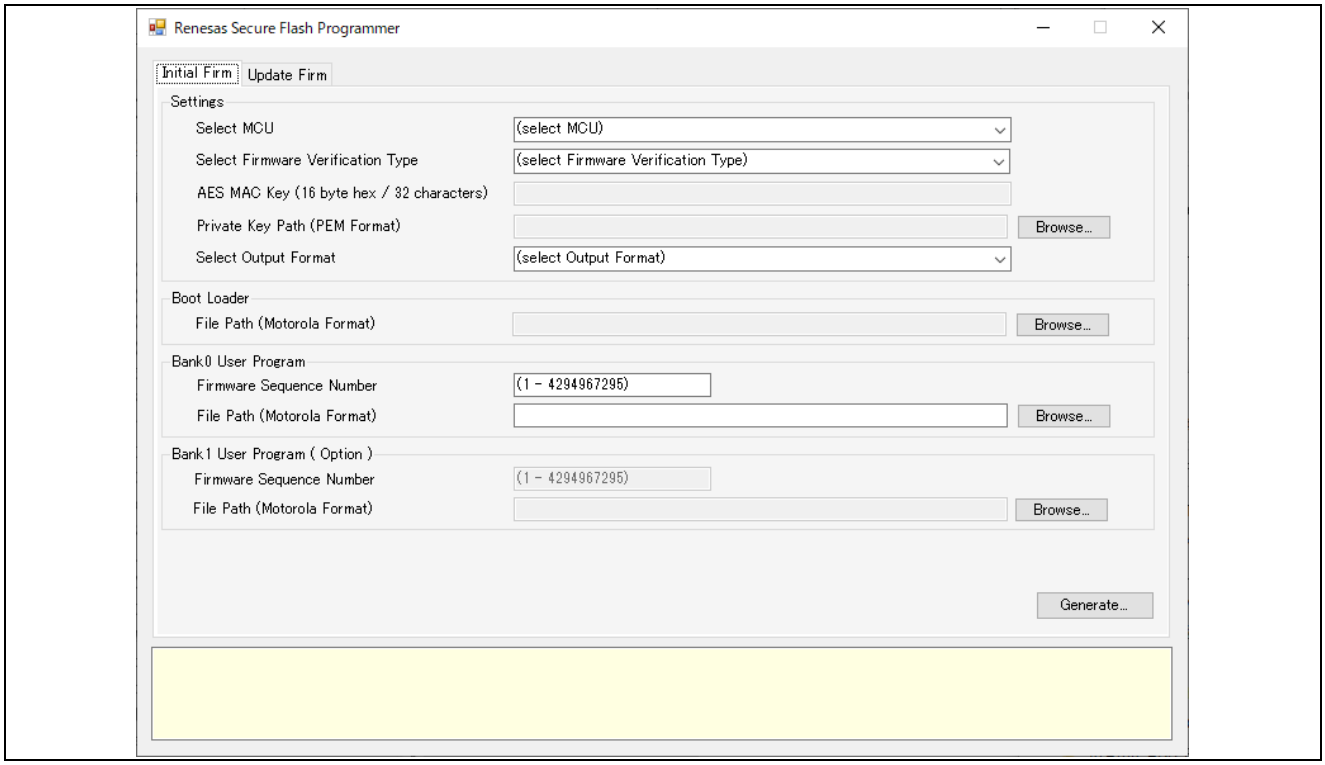

図 2-28 Renesas Secure Flash Programmer の画面

画面の Initial Firm タブをクリックして以下項目を青字の通りに設定してください。

- Select MCU: RX65N Flash(Code=2MB, Data=32KB)/Secure Bootloader=64KB
- Select Firmware Verification Type: sig-sha256-ecdsa
- Private Key Path (PEM format): secp256r1.privatekey generated by OpenSSL in step E
- Select Output Format: Bank 0 User Program + Boot Loader (Motorola S Format)
- Boot Loader File Path (Motorola Format): \bootloader\HardwareDebug\bootloader.mot
- Firmware Sequence Number: 1
- Bank 0 User Program File Path (Motorola format): \adu_sample\HardwareDebug\adu_sample.mot

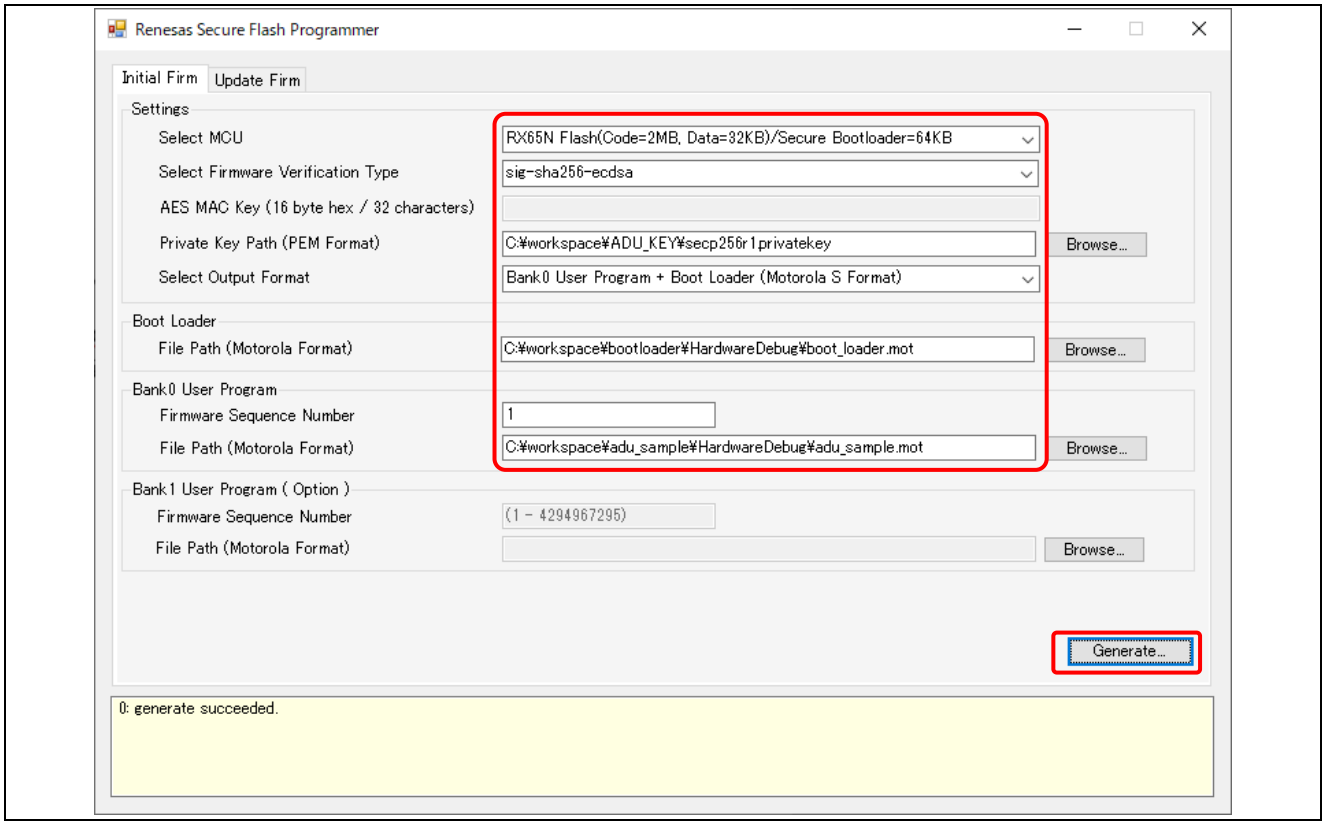

図 2-29 初期ファームウェア作成画面

次に[Generate...]ボタンをクリックして以下の画面で、任意の出力先のフォルダを設定し、出力ファイル名を入力してください。出力ファイル名は「userprog.mot」と入力します。

[保存]ボタンをクリックすると指定したフォルダに初期ファームウェアの MOT ファイルが出力されます。
画面下の表示に"generate succeeded."と表示されたら完了となります。

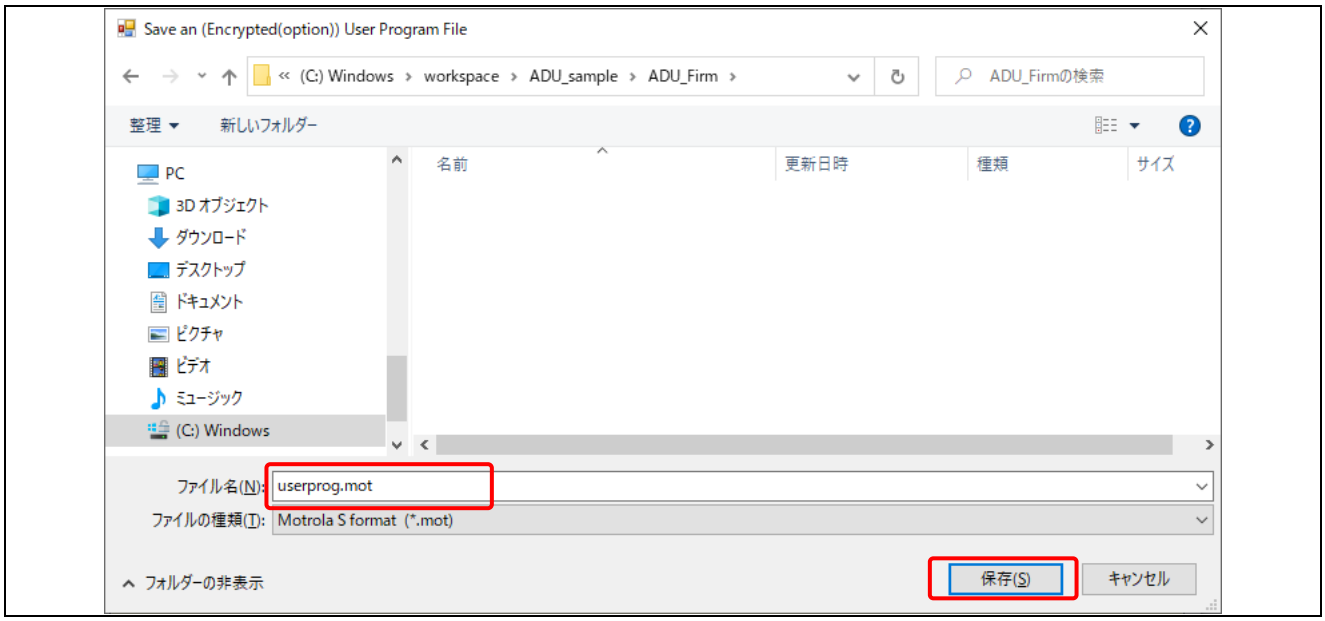

図 2-30 初期ファームウェア MOT ファイルの出力

Renesas Secure Flash Programmer で生成される userprog.mot の構成を以下に示します。
検証データ（Verification data）と初期ファームウェアとブートローダをまとめた MOT ファイルとなります。
Verification data は、ファームウェアのアドレス情報や、検証を行うための情報が格納されています。

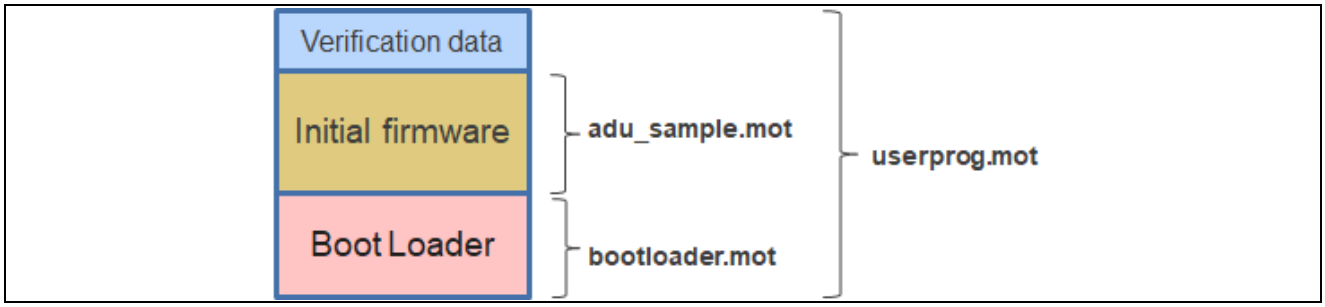

図 2-31 userprog.mot（v1）ファイルの構成

2.9.2　ファームウェアアップデートモジュール v2 を使用する場合を

MOT ファイル変換ツールを使用して、bootloader と adu_sample のそれぞれの MOT ファイルを合成し、初期ファームウェアイメージを作成します。
MOT ファイル変換ツールには Renesas Image Generator を使用します。

1.　Python のインストール

Renesas Image Generator を使用するためには Python の実行環境が必要です。Python のダウンロードサイトにアクセスし、Python のダウンロードとインストールを実施してください。本書では Python 3.12.0 で動作確認を行っています。
インストール後、Windows のシステムの環境変数の Path に以下の Python のインストールフォルダを追加してください。
v3.12.0 の場合の例：C:\Users\xxxx\AppData\Local\Programs\Python\Python312

※xxxx はご使用の Windows のユーザー名。インストール環境によってインストール先は異なる場合があります。

システムの環境変数は Windows10 の場合、以下の操作で開くことができます。
[設定]⇒[システム]⇒[詳細情報]⇒[システムの詳細設定]⇒[システムのプロパティ]⇒[詳細設定]タブ⇒[環境変数]⇒[システムの環境変数]

Python のインストールが完了したら Python 暗号化ライブラリ（pycryptodome）をインストールします。
コマンドラインで以下のコマンドを入力すると、暗号化ライブラリがダウンロード後、インストールされます。

```
> python -m pip install pycryptodome
```

インストールが完了すると"Successfully installed pycryptodome-x.xx.x"と表示されます。

2.　初期ファームウェアイメージの作成

以下手順で初期ファームウェアのイメージファイルを作成します。

①サンプルプロジェクトに同梱されている"RenesasImageGenerator"フォルダを開きます。このフォルダはサンプルプロジェクトの tools フォルダに格納されています。
本フォルダには ADU 用の Renesas Image Generator の 3 つのファイルが格納されています。

```
tools
|--RenesasImageGenerator
   |-- image-gen.py
   |-- RX65N_DualBank_Initial_PRM.csv
   |-- RX65N_DualBank_Update_PRM.csv
```

以下にフォルダ内のファイルの説明を記載します。

ファイル名	内容
image-gen.py	Renesas Image Generator アプリケーション（Python コード）
RX65N_DualBank_Initial_PRM.csv	初期ファーム作成用パラメータファイル
RX65N_DualBank_Update_PRM.csv	更新ファームウェア作成用パラメータファイル

②以下の3つのファイルを RenesasImageGenerator フォルダへコピーします。

秘密鍵	secp256r1.privatekey	"2.4.2 OpenSSL での ECC 用鍵ペア生成"で作成
ブートローダ	bootloader.mot	"2.5 bootloader プロジェクトのビルド"で作成
初期ファームウェア	adu_sample.mot	"2.8 adu_sample のビルド"で作成

コピーするファイルは RenesasImageGenerator フォルダに以下のように配置します。赤字がコピーするファイルです。

```
RenesasImageGenerator
|-- image-gen.py
|-- RX65N_DualBank_Initial_PRM.csv
|-- RX65N_DualBank_Update_PRM.csv
|-- secp256r1.privatekey
|-- bootloader.mot
|-- adu_sample.mot
```

③コマンドラインを起動後 RenesasImageGenerator フォルダに移動します。

④コマンドラインで、以下のコマンドを実行すると、初期ファームウェアイメージが作成されます。

```
> python image-gen.py -iup adu_sample.mot -ip RX65N_DualBank_Initial_PRM.csv -o
userprog -ibp bootloader.mot -vt ecdsa
```

作成にはしばらく時間がかかります。
画面に" Successfully generated the userprog.mot file. "と表示されたら完了となります。
RenesasImageGenerator フォルダに初期ファームウェアイメージ" userprog.mot "が出力されます。

Renesas Image Generator で生成される userprog.mot の構成を以下に示します。

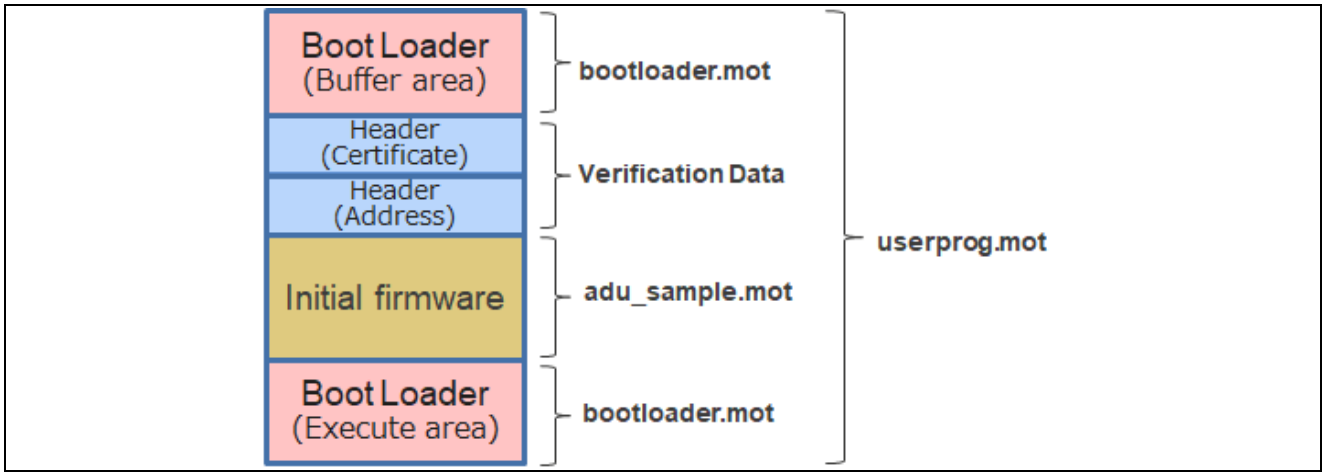

図 2-32 userprog.mot（v2）ファイルの構成

初期ファームウェアイメージ”userprog.mot”は、ヘッダと初期ファームウェアとブートローダをまとめた MOT ファイルとなります。
ヘッダ（署名情報）にはファームウェアの検証を行うための情報が、ヘッダ（アドレス情報）にはファームウェアのアドレス情報が格納されています。

また、Renesas Image Generator はバッファエリアのブートローダも同時に合成します。

2.10 フラッシュ書き込みツールのインストール

フラッシュ書き込みツールのダウンロードサイトにアクセスし、"Renesas Flash Programmer V3.11.02 Windows"をダウンロードしてインストールしてください。

2.11 初期ファームウェアの書き込み

A)　初期ファームウェアの FLASH 書き込み（リニアモード→デュアルモード）

ターゲットボードに初期ファームウェアを書き込みます。
本手順では、初期状態ではリニアモードになっている RX65N をデュアルモードに変更しファームウェアの書き込みを行います。

まず、\adu_sample\tools\Flash_Project\CKRX65N_ADU_Write フォルダにある flash_project.rpj を起動してください。次に作成した初期ファームウェア「userprog.mot」を RX65N に書き込むため、[操作]タブの[プログラムファイル]に「userprog.mot」を指定して[スタートボタン]を押下してください。

画面下に"操作が成功しました。"と表示されたら書き込みは完了です。

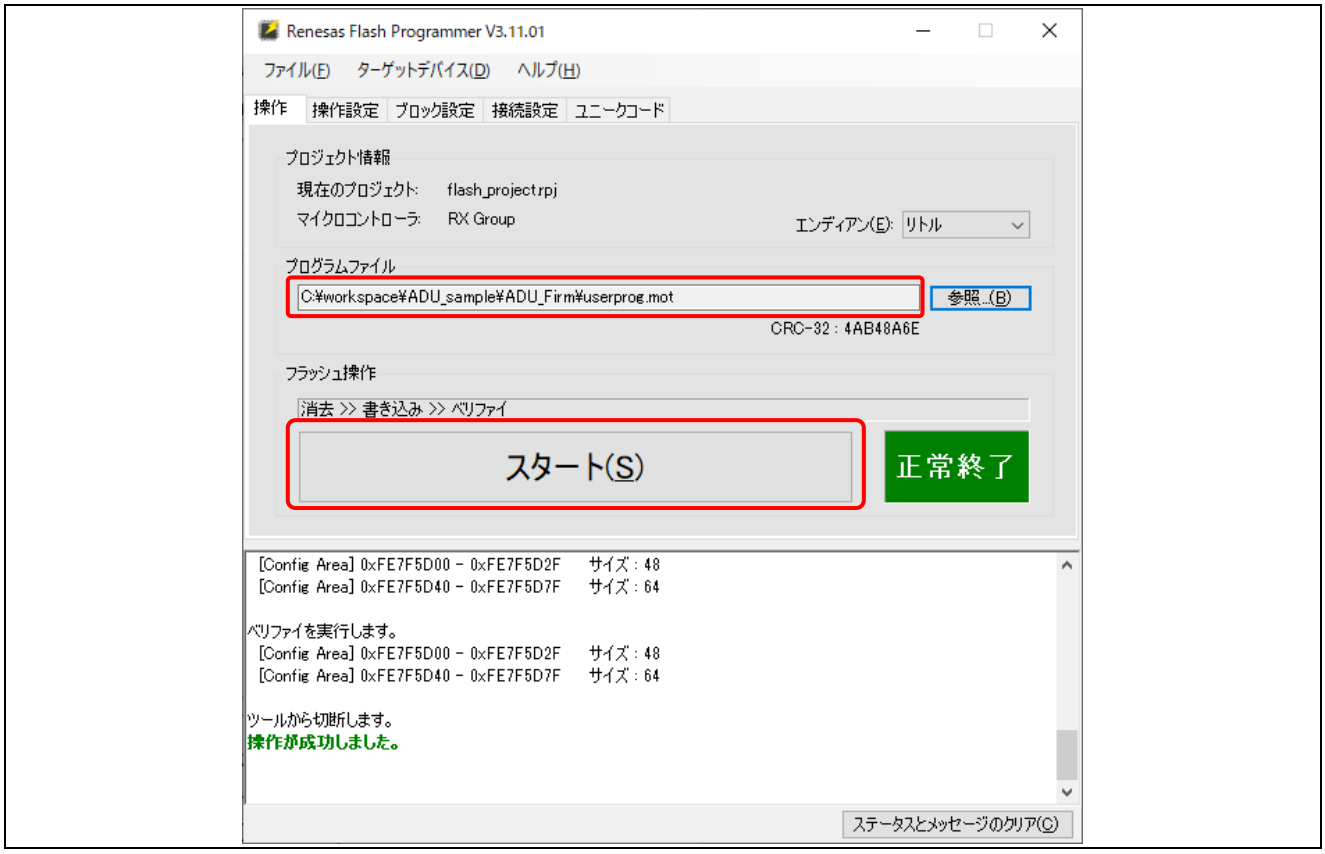

図 2-33 初期ファームウェアをデバイスへ書き込み

初期ファームウェアを実行することで、マイコンはリニアモード（通常のモード）からデュアルモード (コードフラッシュメモリを 2 領域に分割して使用するモード) に遷移します。
次回以降、再度、初期ファームウェアを FLASH 書き込みする場合は、B)の FLASH 消去を実行してからリニアモードに変更後に、A)の FLASH 書き込みを実行してください。

B)　FLASH 消去 (デュアルモード→リニアモード)

チップ消去を実行することで ROM 情報を初期化し、デュアルモードになっている RX65N をリニアモードに変更します。
tools/Flash_Project にある CKRX65N_ ADU_Erase フォルダにある erase_project.rpj を起動し、[操作]タブの「スタート」ボタンを押下することでチップ消去してください。

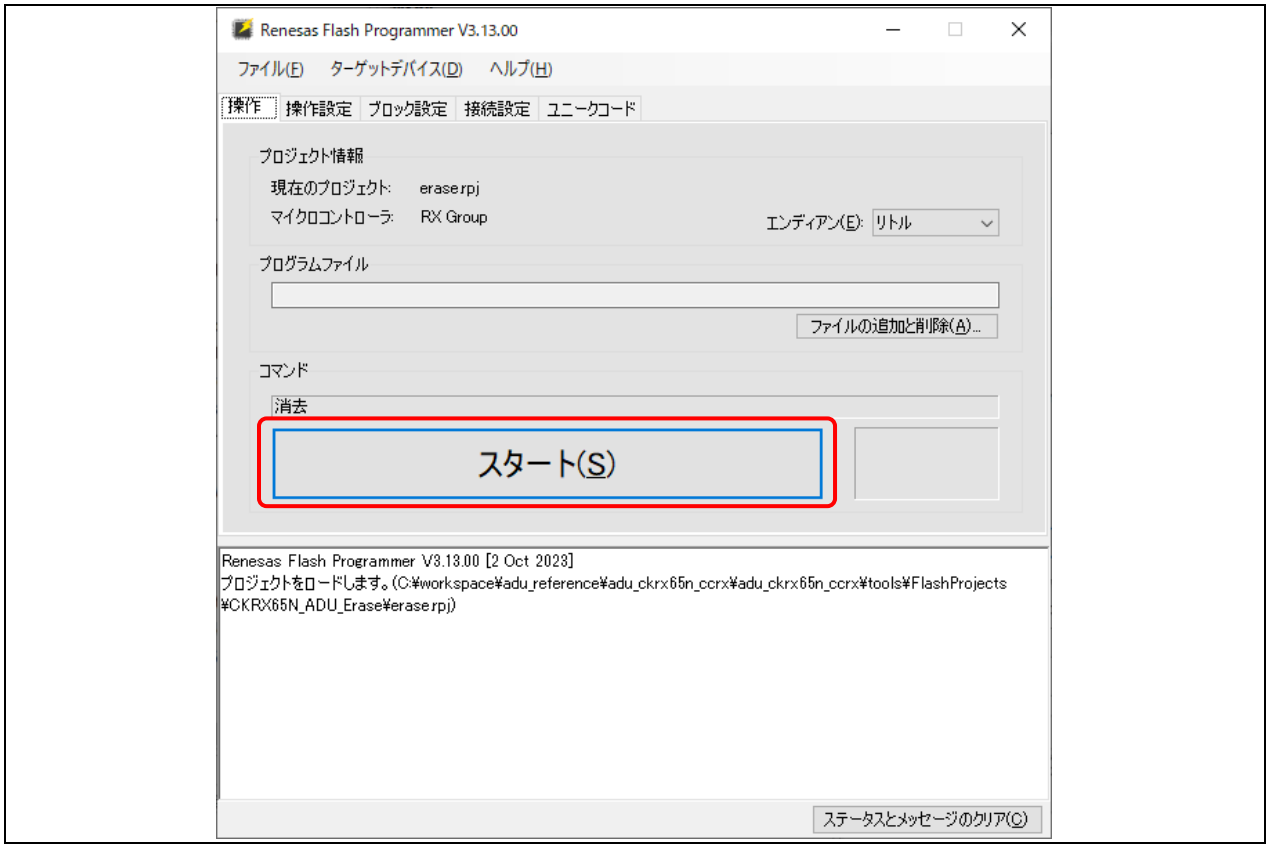

図 2-34 FLASH の消去とリニアモードへの変更

Renesas Flash Programmer は、リニアモードとデュアルモードを別の MCU として認識します。A)の手順はデバイスがリニアモードであることを認識してから FLASH への書き込みを実行します。また、B)の手順はデバイスがデュアルモードであることを認識してから FLASH の消去を実行します。
このため、A)あるいは B)をそれぞれ連続して実行すると、デバイス情報が不一致となり「エラー(E3000107): デバイスが接続情報と一致しません。」が発生します。

2.12 初期ファームウェアの実行

初期ファームウェアの書き込みが完了後、RX65N 上でプログラムが動作します。
実際の動作手順は「3.8.1 ターゲットボードの実行」の節を参照してください。
書き込みからのプログラム起動のシーケンスは以下の通りとなります。

(1)初期ファームウェアの書き込み後、プログラムが動作します。

(2)プログラム実行後はブートローダが起動して Verification data エリアに書き込まれている公開鍵を使用して暗号化されたハッシュ値を復号します。

(3)ファームウェア全体のハッシュ値を計算して、復号したハッシュ値と一致するか確認します。
一致した場合、ファームウェアを起動します。
ファームウェアアップデートモジュール v1 をご利用の場合は、ファームウェア検証後にバッファエリアへ bootloader がコピーされます。

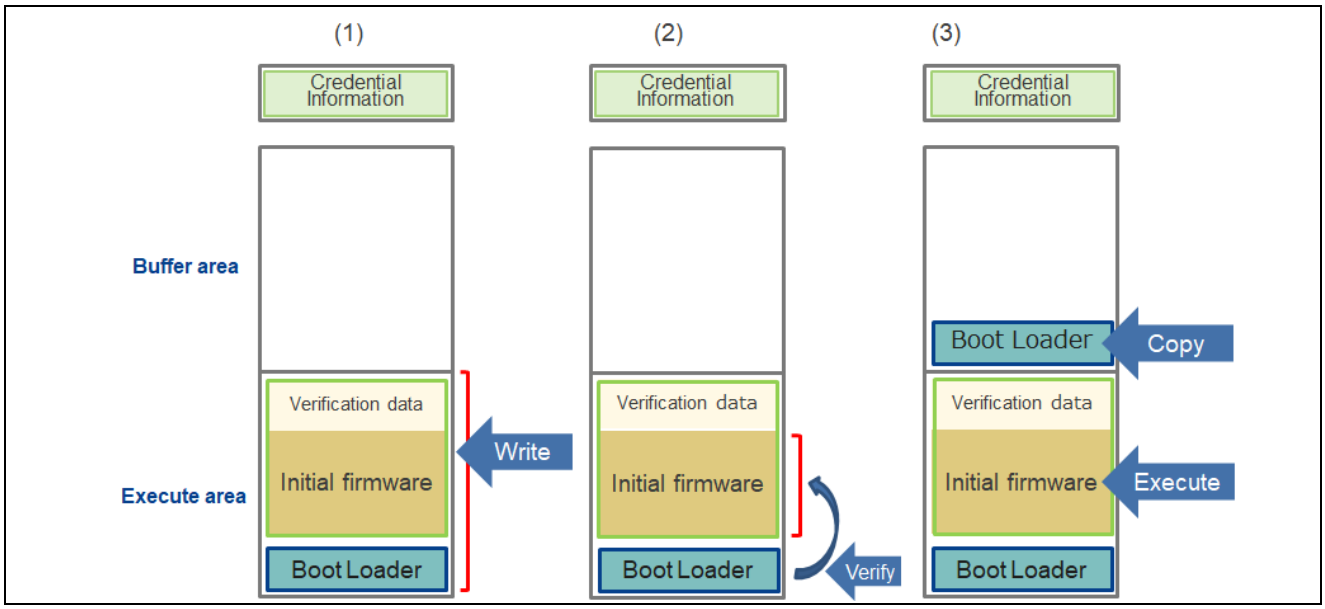

図 2-35 ブートローダの動作（v1）

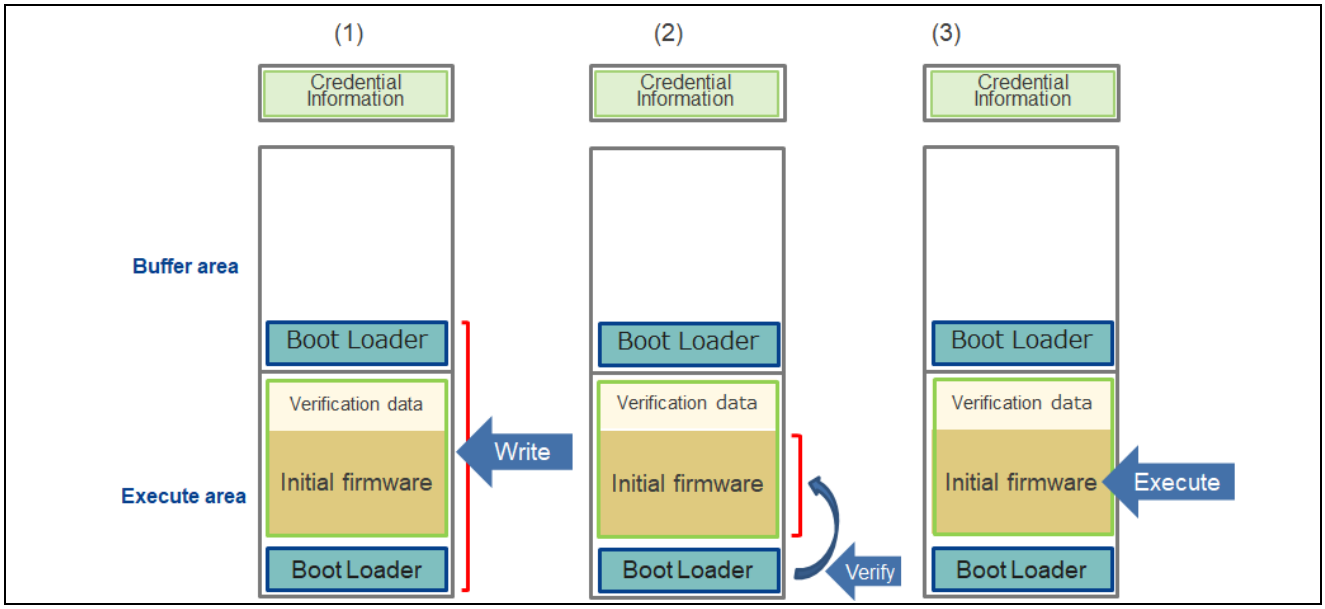

図 2-36 ブートローダの動作（v2）

2.13 更新ファームウェアのコード変更

adu_sample プロジェクトの\src\sample_azure_iot_embedded_sdk_adu.c を開き、
SAMPLE_DEVICE_INSTALLED_CRITERIA を”1.1.0”に更新してください。(※)
(※)Azure IoT Hub に設定したバージョンのファームウェアがすでに存在する場合は、異なるバージョンを設定してください。

その後、必要なアップデート処理を追加してください。

```
#ifndef SAMPLE_DEVICE_INSTALLED_CRITERIA
#define SAMPLE_DEVICE_INSTALLED_CRITERIA                    "1.1.0"
#endif /* SAMPLE_DEVICE_INSTALLED_CRITERIA */

#if (NX_AZURE_IOT_ADU_AGENT_PROXY_UPDATE_COUNT >= 1)
/* Device properties.  */
#ifndef SAMPLE_LEAF_DEVICE_MANUFACTURER
#define SAMPLE_LEAF_DEVICE_MANUFACTURER                     "Contoso"
#endif /* SAMPLE_LEAF_DEVICE_MANUFACTURER*/

#ifndef SAMPLE_LEAF_DEVICE_MODEL
#define SAMPLE_LEAF_DEVICE_MODEL                            "IoTDevice-Leaf"
#endif /* SAMPLE_LEAF_DEVICE_MODEL */

#ifndef SAMPLE_LEAF_DEVICE_INSTALLED_CRITERIA
#define SAMPLE_LEAF_DEVICE_INSTALLED_CRITERIA               "1.0.0"
#endif /* SAMPLE_LEAF_DEVICE_INSTALLED_CRITERIA */
#endif /* NX_AZURE_IOT_ADU_AGENT_PROXY_UPDATE_COUNT */
```

図 2-37 更新ファームウェアの設定

2.14 adu_sample のビルド（更新ファームウェア用）

adu_sample プロジェクトをビルドし、adu_sample.mot ファイルを生成してください。
MOT ファイルは以下フォルダに生成されます。
\adu_sample\HardwareDebug\

2.15 更新ファームウェアの作成

更新ファームウェアを RSU 形式(※)に変換します。

ファームウェアの作成はファームウェアアップデートモジュール v1 と v2 では作成する手順が異なります。
お使いのバージョンに対応する手順を参照してください。
また、ファームウェアアップデートモジュール v1 を使用して作成したブートローダで v2 の更新ファームウェアを適用する場合も手順が異なりますので、対応する手順を参照して下さい。

(※) 一般的なファームウェアのデータフォーマットである MOT ファイルにはデバイスに書き込むための実データ以外のデータ(例えばハッシュ値等の検証用データ)を保持する機構がありません。また MOT ファイルはテキストデータであるため、バイナリデータと比べてデータ量が 2 倍に増える傾向にあります。
これらを回避するため、実データ以外のデータ(Verification data)を保持したルネサス独自のバイナリ方式である RSU(Renesas Secure Update)方式を規定して ADU に使用します。

2.15.1 ファームウェアアップデートモジュール v1 を使用する場合

ファームウェア adu_sample.mot に検証情報を付加して更新ファームウェアの RSU ファイルを作成します。
更新ファームウェアの作成は Renesas Secure Flash Programmer を使用します。
mot-file-converter-2.0.2 \Renesas Secure Flash Programmer\bin\Debug\Renesas Secure Flash Programmer.exe をダブルクリックで起動し、[Update Firm] タブをクリックして以下項目を青字の通りに設定してください。

- Select MCU: RX65N Flash(Code=2MB, Data=32KB)/Secure Bootloader=64KB
- Select Firmware Verification Type: sig-sha256-ecdsa
- Private Key Path (PEM format): secp256r1.privatekey generated by OpenSSL in step E
- Firmware Sequence Number: 1
- File Path (Motorola format): \adu_sample\HardwareDebug\adu_sample.mot

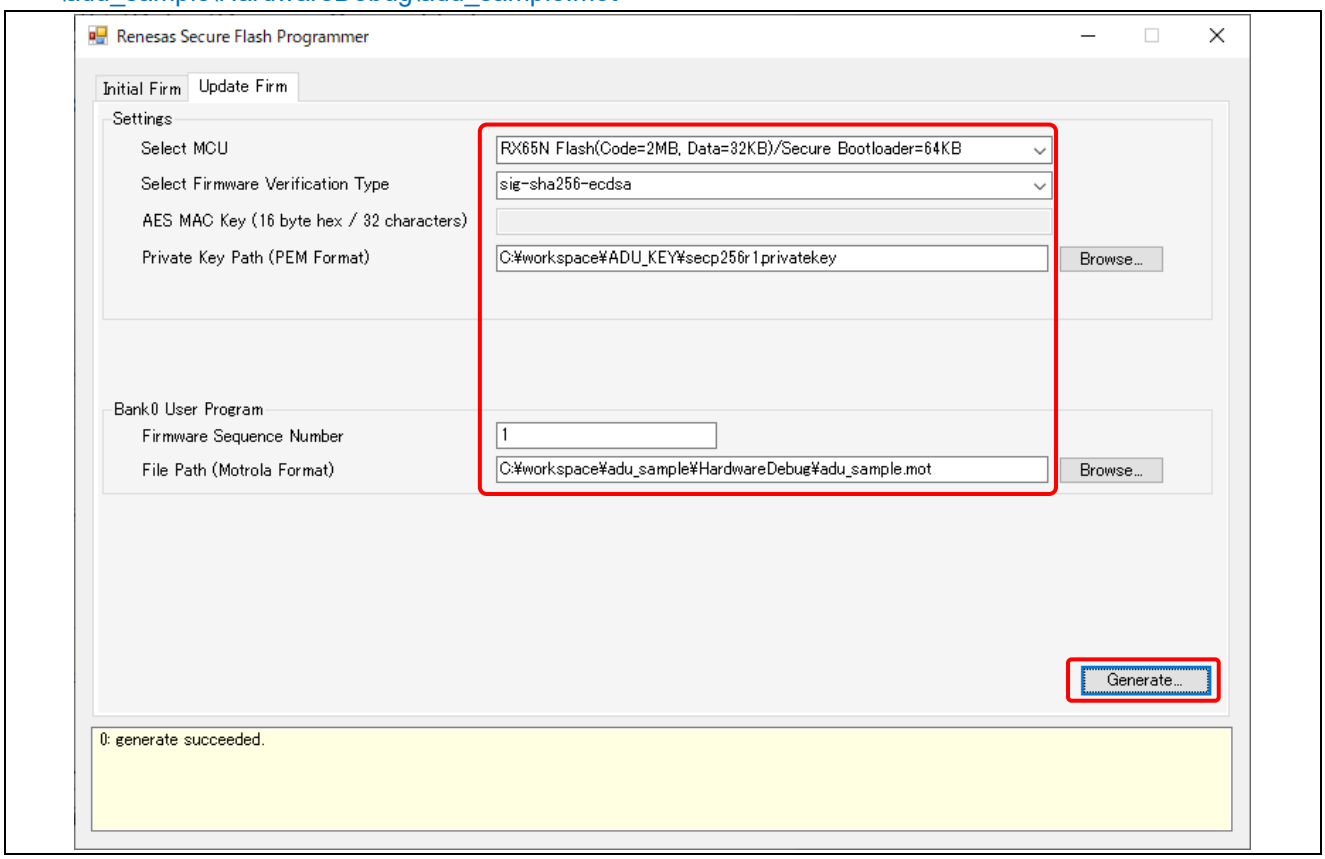

図 2-38 更新ファームウェア作成画面

最後に[Generate...]ボタンをクリックして以下の画面で、任意の出力先のフォルダを設定し、出力ファイル名を入力してください。出力ファイル名は「firmware_1.1.0.rsu」(※)と入力します。
[保存]ボタンを押下すると指定したフォルダに更新ファームウェアの RSU ファイルが出力されます。
画面下の表示に"generate succeeded."と表示されたら完了となります。

(※)ファイル名の"1.1.0"は設定した更新ファームウェアのバージョンに合わせてください。

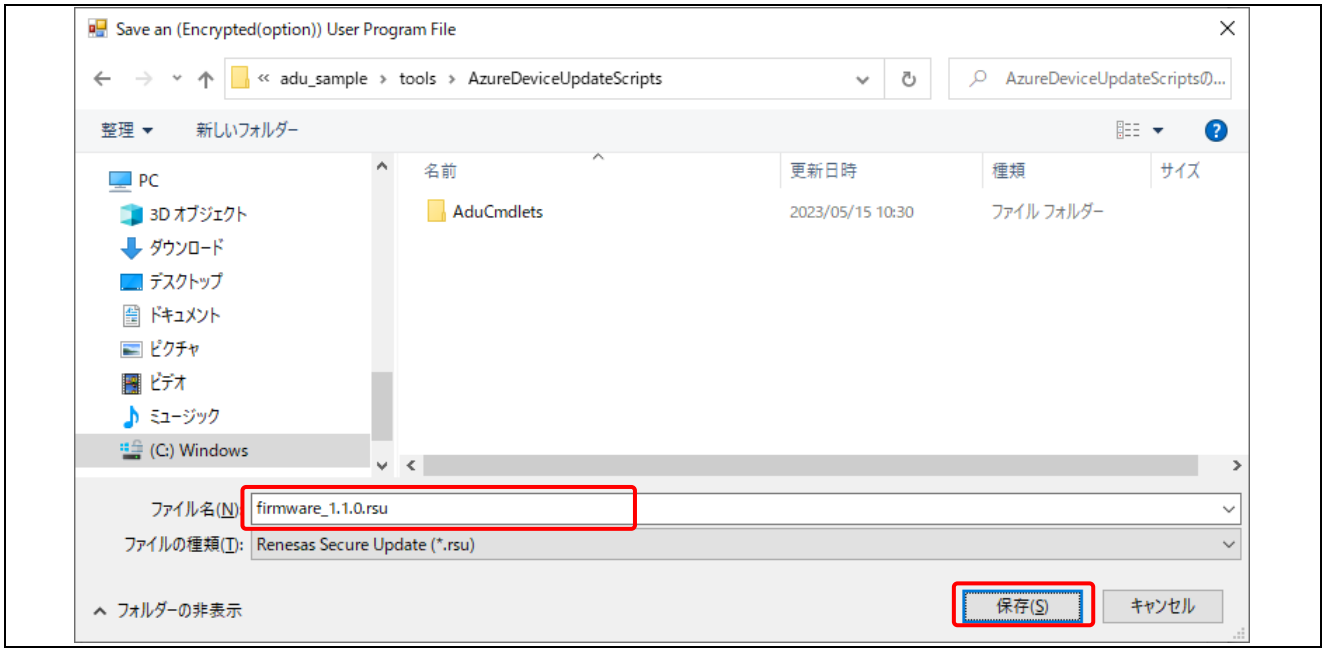

図 2-39 更新ファームウェア RSU ファイルの出力

画面下の表示に"generate succeeded."と表示されたら完了となります。

Renesas Secure Flash Programmer で生成される更新ファームウェア RSU ファイルの構成を以下に示します。

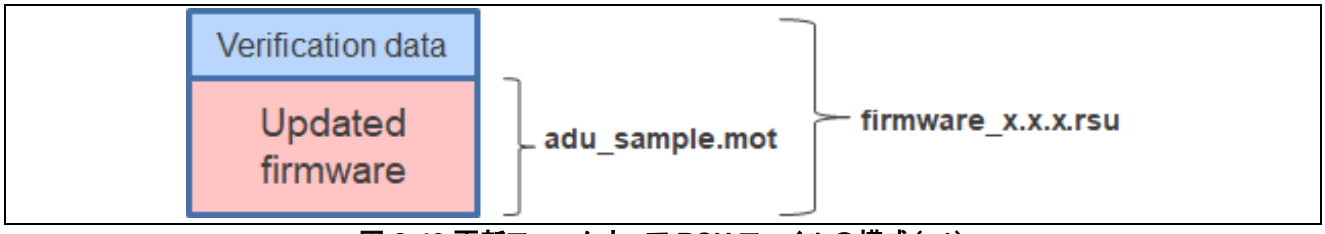

図 2-40 更新ファームウェア RSU ファイルの構成(v1)

更新ファームウェアイメージ"firmware_1.1.0.rsu"は、検証データ(Verification data)と更新ファームウェア(MOT ファイル)をまとめてバイナリファイルとしたものです。

RSU ファイルの詳細は、アプリケーションノート"ルネサス MCU におけるファームウェアアップデートの設計方針"の 7.1 節"ダウンロードデータフォーマット"を参照してください。
また、本項には Verification data(0x00000000～0x000002FF)の詳細も説明しています。

更新ファームウェアが作成できたら、"3.Microsoft Azure ポータルでの操作"の章を参照し、IoT Hub へ更新ファームウェアの登録を行い、ファームウェアの更新を実行してください。

2.15.2 ファームウェアアップデートモジュール v2 を使用する場合を

ファームウェア adu_sample.mot にヘッダ情報を付加して更新ファームウェアの RSU ファイルを作成します。

更新ファームウェアの作成には Renesas Image Generator を使用します。
以下手順で更新ファームウェアのイメージファイルを作成してください。

①サンプルプロジェクトに同梱されている"RenesasImageGenerator"フォルダを開きます。
本フォルダには以下の 3 つのファイルが格納されています。

```
RenesasImageGenerator
|-- image-gen.py
|-- RX65N_DualBank_Initial_PRM.csv
|-- RX65N_DualBank_Update_PRM.csv
```

②以下の 2 つのファイルを RenesasImageGenerator フォルダへコピーします。

秘密鍵	secp256r1.privatekey	"2.4.2 OpenSSL での ECC 用鍵ペア生成"で作成
更新ファームウェア	adu_sample.mot	"2.14 adu_sample のビルド（更新ファームウェア用）"で作成

コピーするファイルは RenesasImageGenerator フォルダに以下のように配置します。赤字がコピーするファイルです。

```
RenesasImageGenerator
|-- image-gen.py
|-- RX65N_DualBank_Initial_PRM.csv
|-- RX65N_DualBank_Update_PRM.csv
|-- secp256r1.privatekey
|-- adu_sample.mot
```

③コマンドラインを起動後 RenesasImageGenerator フォルダに移動します。

④コマンドラインで、以下のコマンドをコマンドラインから実行すると、更新ファームウェアイメージが作成されます。

```
> python image-gen.py -iup adu_sample.mot -ip RX65N_DualBank_Update_PRM.csv -o
firmware_1.1.0 -vt ecdsa
```

上記赤字部が、出力される更新ファームウェアイメージのファイル名となります。
ファイル名の"1.1.0"の部分は設定した更新ファームウェアのバージョンに合わせてください。

ファームウェアの作成にはしばらく時間がかかります。
画面に" Successfully generated the firmware_1.1.0.rsu file."と表示されたら完了となります。
RenesasImageGenerator フォルダに更新ファームウェアイメージの RSU ファイル、”firmware_1.1.0.rsu"が出力されます。

Renesas Image Generator で生成される更新ファームウェア RSU ファイルの構成を以下に示します。

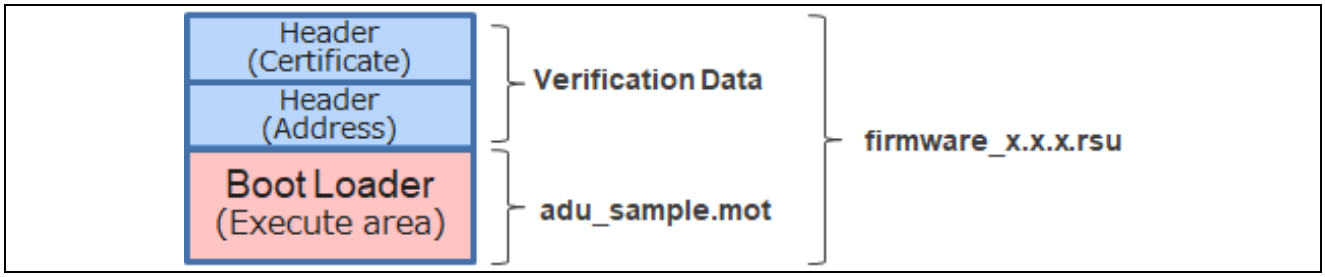

図 2-41 更新ファームウェア RSU ファイルの構成（v2）

更新ファームウェアイメージ”firmware_1.1.0.rsu”は、検証データ（Verification data）と更新ファームウェア（MOT ファイル）をまとめてバイナリファイルとしたものです。
検証データのヘッダ（署名情報）にはファームウェアの検証を行うための情報が、ヘッダ（アドレス情報）にはファームウェアのアドレス情報が格納されています。

RSU ファイルの詳細は、アプリケーションノート”ファームウェアアップデートモジュール (Rev2.01)”の 4.2.1 節の”更新イメージファイル”を参照してください。
また、本項には RSU ヘッダフォーマット（Verification data：0x00000000～0x000002FF）の詳細も説明しています。

更新ファームウェアが作成できたら、"3.Microsoft Azure ポータルでの操作"の章を参照し、IoT Hub へ更新ファームウェアの登録を行い、ファームウェアの更新を実行してください。

2.15.3 ブートローダ v1 の環境でファームウェアアップデートモジュール v2 を使用する場合

ファームウェアアップデートモジュール v1 と v2 では作成される RSU ファイルのフォーマットが異なります。
このため、ファームウェアアップデートモジュール v1 で作成したブートローダではファームウェアアップデートモジュール v2 で作成した更新ファームウェアをそのままでは使用することができません。

この場合は以下の手順で更新ファームウェアイメージを作成してください。(※)

(※)本手順で作成した更新ファームウェアイメージは、ファームウェアアップデート v2 で作成したブートローダでは使用できないため、”2.15.2 ファームウェアアップデートモジュール v2 を使用する場合を”の手順で作成してください。

①更新ファームウェアをビルドする際に以下の設定を行います。
スマート・コンフィグレータのコンポーネントツリーにて r_fwup を選択し、”FWUP v1 compatible Setting”プロパティを”Enabled”にセットして adu_sample をビルドしてください。
スマート・コンフィグレータの設定変更後は[コードの生成]ボタンをクリックしてコード生成を実施して下さい。

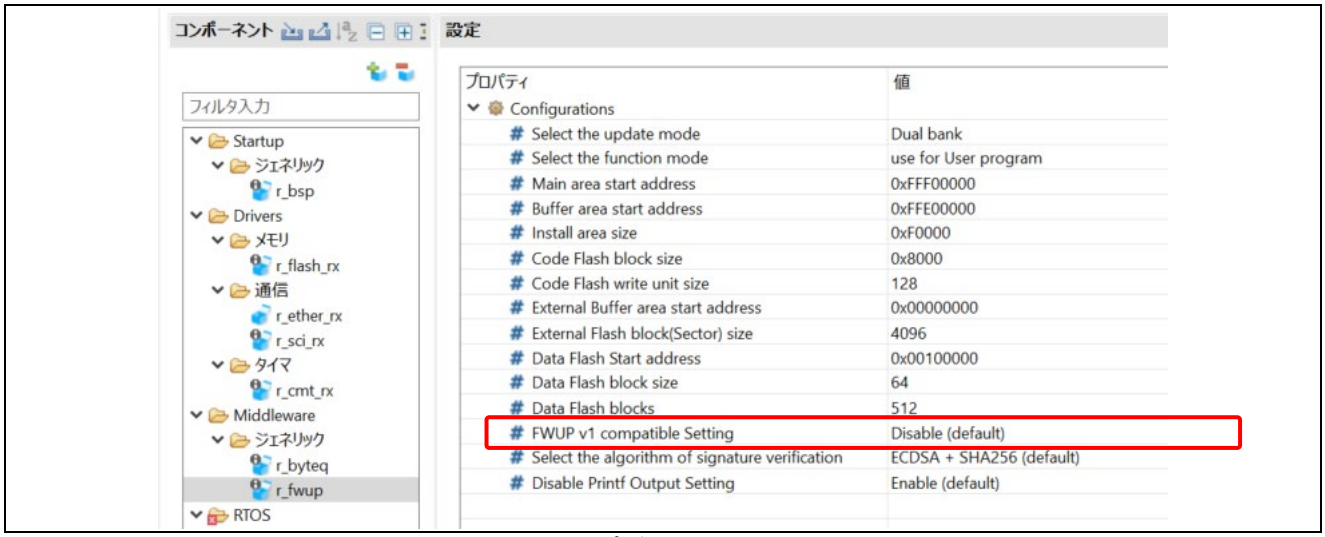

図 2-42 ファームウェアアップデートモジュール v2 の v1 互換設定

②サンプルプロジェクトに同梱されている"RenesasImageGenerator"フォルダを開きます。
本フォルダには以下の 3 つのファイルが格納されています。

```
RenesasImageGenerator
|-- image-gen.py
|-- RX65N_DualBank_Initial_PRM.csv
|-- RX65N_DualBank_Update_PRM.csv
```

③以下の 2 つのファイルを”RenesasImageGenerator” フォルダへコピーします。

秘密鍵	secp256r1.privatekey	"2.4.2 OpenSSL での ECC 用鍵ペア生成"で作成
更新ファームウェア	adu_sample.mot	"2.14 adu_sample のビルド（更新ファームウェア用）"で作成 ”FWUP v1 compatible Setting”を Enable でビルドしたもの

RenesasImageGenerator がフォルダに以下のように配置されます。赤字がコピーするファイルです。

```
RenesasImageGenerator
|-- image-gen.py
|-- RX65N_DualBank_Initial_PRM.csv
|-- RX65N_DualBank_Update_PRM.csv
|-- secp256r1.privatekey
|-- adu_sample.mot
```

④コマンドラインを起動後”RenesasImageGenerator”フォルダに移動します。

⑤コマンドラインで、以下のコマンドをコマンドラインから実行すると、更新ファームウェアイメージが作成されます。

```
> python image-gen.py -iup adu_sample.mot -ip RX65N_DualBank_Update_PRM.csv -o
firmware_1.1.0 -vt ecdsa -ff BAREMETAL_FWUP_V2_V1_DATA
```

上記赤字部が、出力される更新ファームウェアのファイル名となります。
ファイル名の"1.1.0"は設定した更新ファームウェアのバージョンに合わせてください。

作成にはしばらく時間がかかります。
画面に" Successfully generated the firmware_1.1.0.rsu file."と表示されたら完了となります。
RenesasImageGenerator フォルダに更新ファームウェアイメージの RSU ファイル、”firmware_1.1.0.rsu"が出力されます。

更新ファームウェアが作成できたら、"3.Microsoft Azure ポータルでの操作"の章を参照し、IoT Hub へ更新ファームウェアの登録を行い、ファームウェアの更新を実行してください。

3. Microsoft Azure ポータルでの操作

ADU を実施するための Microsoft Azure の操作手順を説明します。

3.1 IoT Hub とデバイスの登録

Azure ポータルにて IoT Hub とデバイスを作成します。(※)
作成にはアプリケーションノート(RX65N Cloud kit で Azure RTOS を用いてセンサデータを可視化する方法)の 3.1 節を参照してください。

(※)参照先の画面例は変更になる場合があります。

(※)ADU を実行するためには IoT Hub の価格オプションをスタンダードレベル S1 以上で選択する必要があり課金が必要です。無料の Free 版では実行できないため注意してください。
以下の IoT Hub 作成画面で、レベルを”Standard”・1 日当たりのメッセージの制限を 400,000 にすることで S1 に設定することができます。

図 3-1 価格オプション設定

ここで作成した IoT Hub の以下の 3 つのパラメータを”2.6 接続情報のマクロ設定”の節を参考にして ADU サンプルプロジェクトのソースコードに記述してください。

- ホスト名:HOST_NAME
- デバイス ID:DEVICE_ID
- 主キー:DEVICE_SYMMETRIC_KEY

3.2 デバイス更新の作成

ADU を実行するための IoT Hub 用のデバイス更新のリソースを作成します。
なお、デバイス更新を利用するためには課金が必要のため、ご注意ください。

(1)Azure ポータルのホーム画面で[リソースの作成]をクリックしてください。リソースの作成画面が開きます。

図 3-2 Azure ホーム画面

(2)IoT ハブ用のデバイス更新を作成します。リソースの作成画面の検索ボックスに”デバイス更新”と入力し、検索してください。Marketplace 画面に”IoT ハブ用のデバイス更新”が検索結果として表示されます。

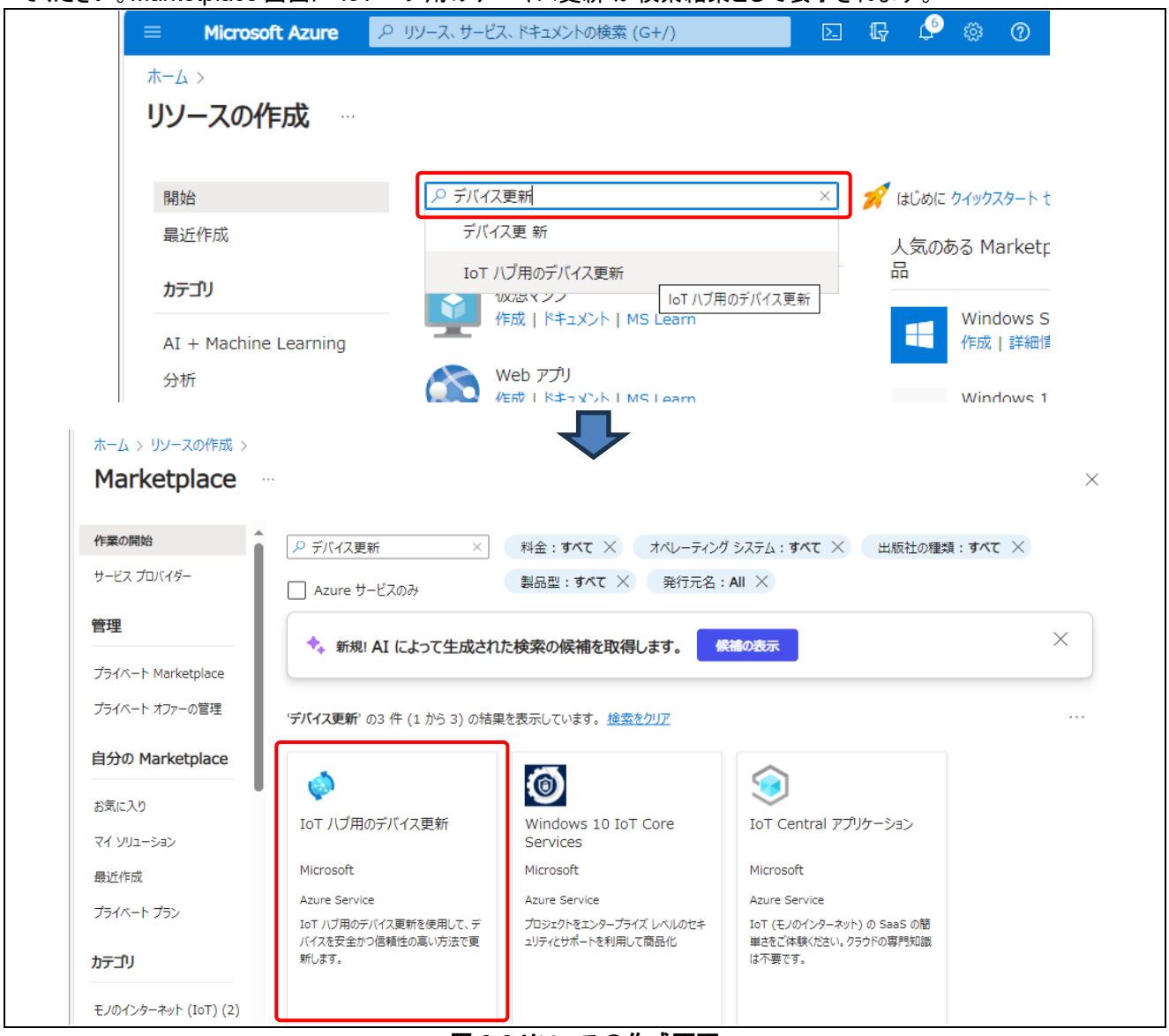

図 3-3 リソースの作成画面

(3)IoT ハブ用のデバイス更新の下部にある[作成]⇒[IoT ハブ用のデバイス更新] をクリックしてください。”デバイス更新の作成”画面が表示されます。

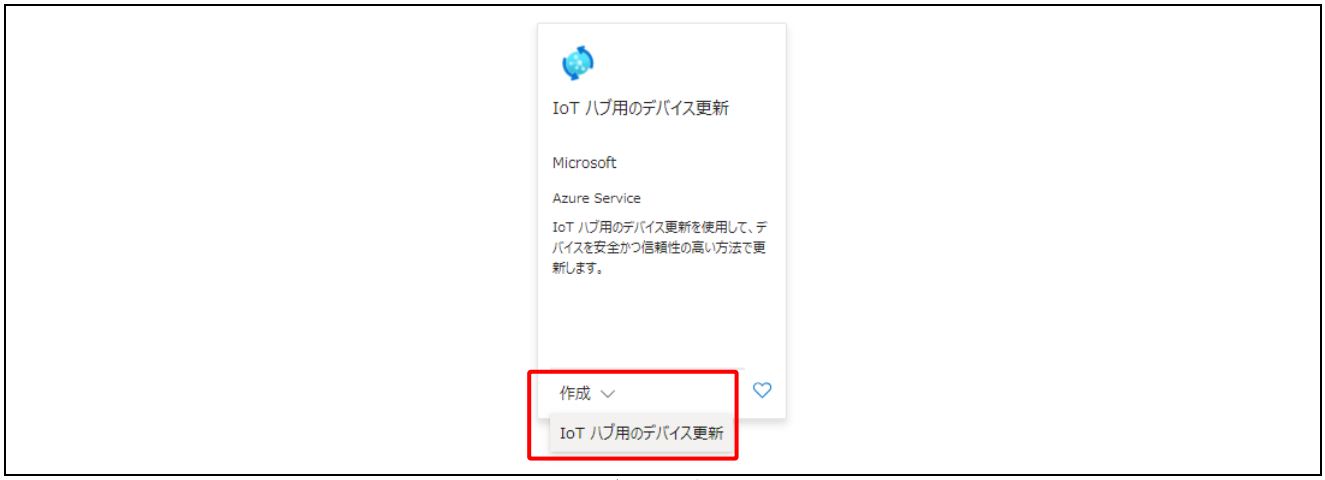

図 3-4 IoT ハブ用のデバイス更新画面

(4)デバイス更新の作成画面で以下の項目を入力してください。設定が完了したら画面下の[次へ：Diagnostics]ボタンをクリックして下さい。

①	サブスクリプション	IoT Hub を作成した際のサブスクリプション ID を指定する
②	リスースグループ	IoT Hub を作成した際のリソースグループを指定する
③	名前（アカウントの詳細）	デバイス更新アカウント名を任意の文字列で入力する
④	場所（アカウントの詳細）	デバイス更新アカウントが配置される Azure の地域となる
⑤	SKU	Standard のままとする
⑥	Grant Access to Account	管理者権限を割り当てるかのチェック。チェックのままとする
⑦	インスタンス名	アカウントの名前がそのまま設定されるので変更しない
⑧	IoT ハブ名	“3.1”節で作成した IoT Hub を指定する

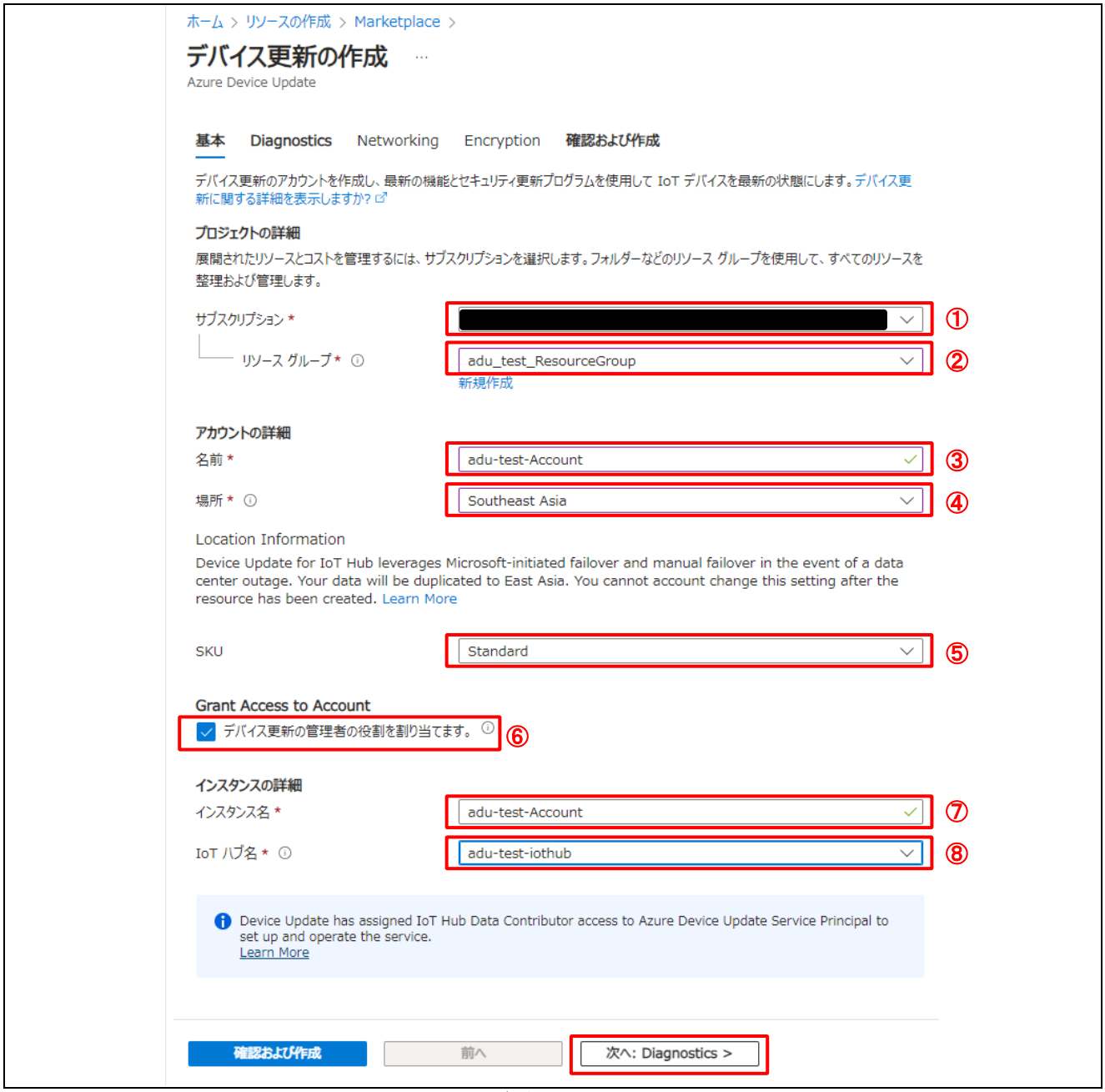

図 3-5 デバイス更新の作成

(5)”Diagnostics”・”Networking”・”Encryption”の各タブの各設定画面は初期値のままとなります。画面下部の[次へ]ボタンをクリックしてください。Encryption 画面で[次へ：確認及び作成]ボタンをクリックすると以下の確認画面が表示されます。
確認画面では登録内容の検証が行われます。検証が完了すると”画面に検証が完了しました”と表示され、画面下部の”作成”ボタンがクリックできるようになります。
設定した内容に問題がないことを確認したら[作成]ボタンをクリックしてください。デバイス更新の作成が開始されます。

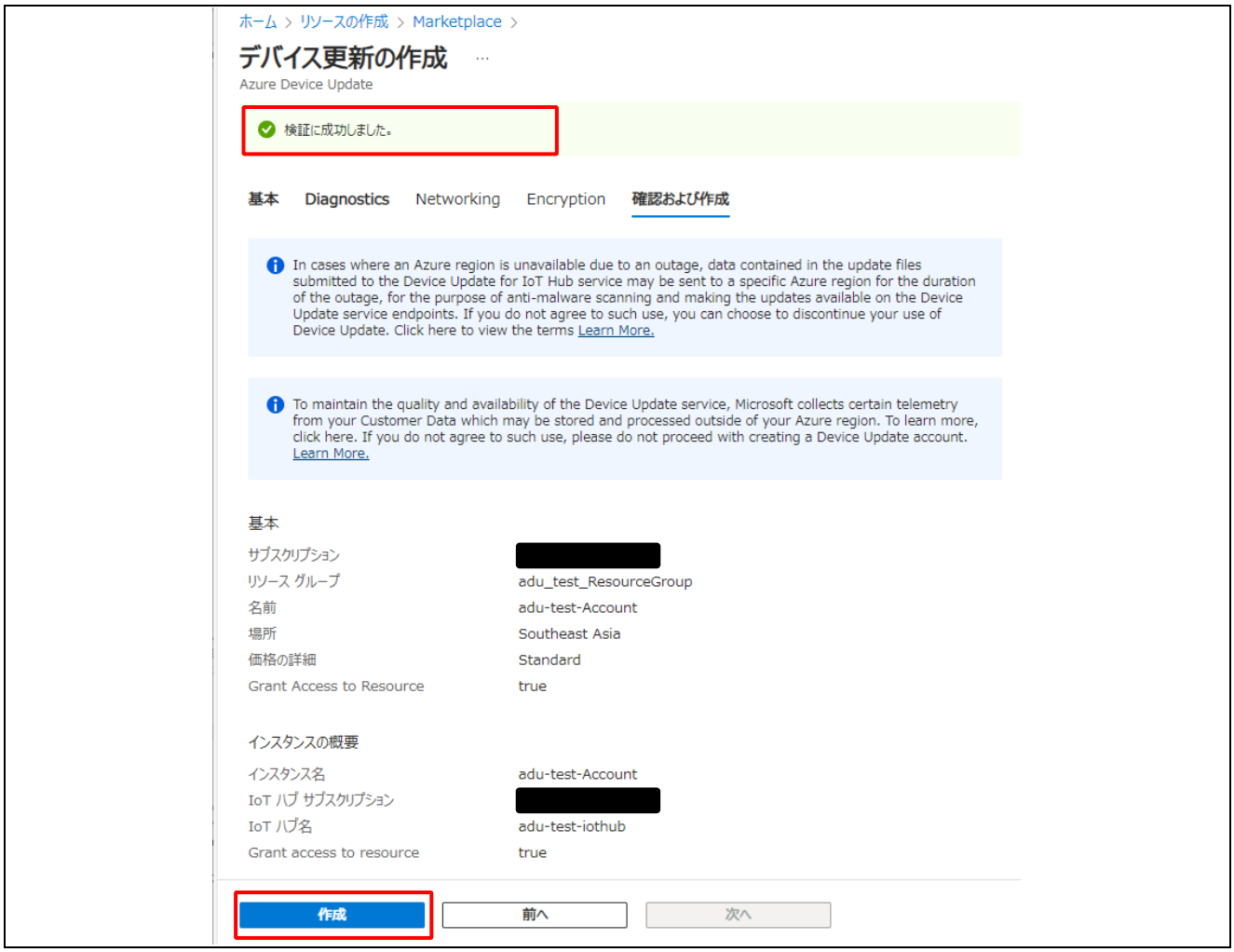

図 3-6 デバイス更新の作成確認画面

(6)デバイス更新の登録（デプロイ）には 10～20 分程度かかることがあります。登録が完了したら”リソースに移動”ボタンをクリックしてください。作成した IoT ハブ用のデバイス更新画面が表示されます。

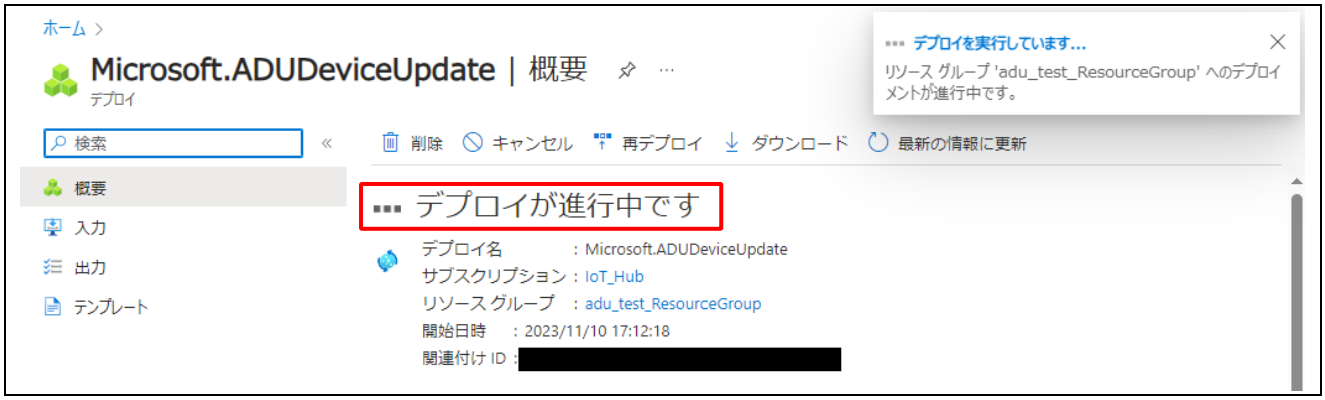

図 3-7 デバイス更新の登録

(7)作成した”IoT Hub のデバイス更新画面”で [デバイス更新のインスタンスの表示]または、画面左側メニューのインスタンスの管理の[インスタンス]をクリックして下さい。インスタンスの登録状況が表示されます。
デバイス更新作成時に設定したインスタンス名とリンクされた IoT Hub 名が表示され、プロビジョニングの状態が成功と表示されればデバイス更新の作成は完了です。

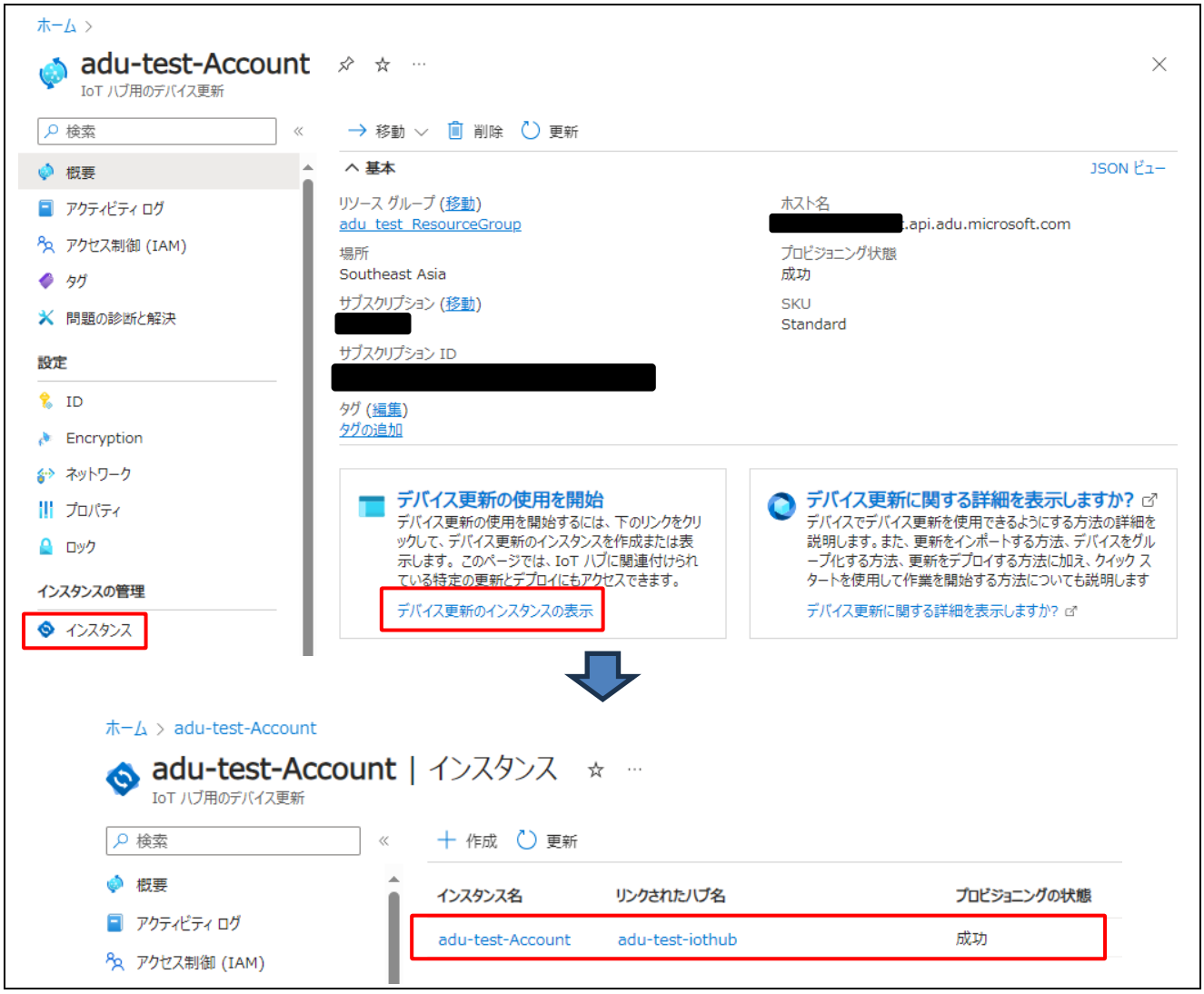

図 3-8 IoT ハブ用のデバイス更新画面

3.3 ストレージアカウント・コンテナーの作成

更新ファームウェアを格納するストレージを作成します。
ストレージを使用するには、ファイルを保存するためのストレージアカウントとコンテナーが必要です。

(1)”3.2 デバイス更新の作成"の節と同様に、Azure ポータルのホーム画面から[リソースの作成]をクリックし、検索ボックスで”ストレージ”を検索してください。
検索結果に”ストレージ アカウント”が表示されるので、[作成]⇒[ストレージアカウント]をクリックしてください。ストレージアカウントの作成画面が表示されます。

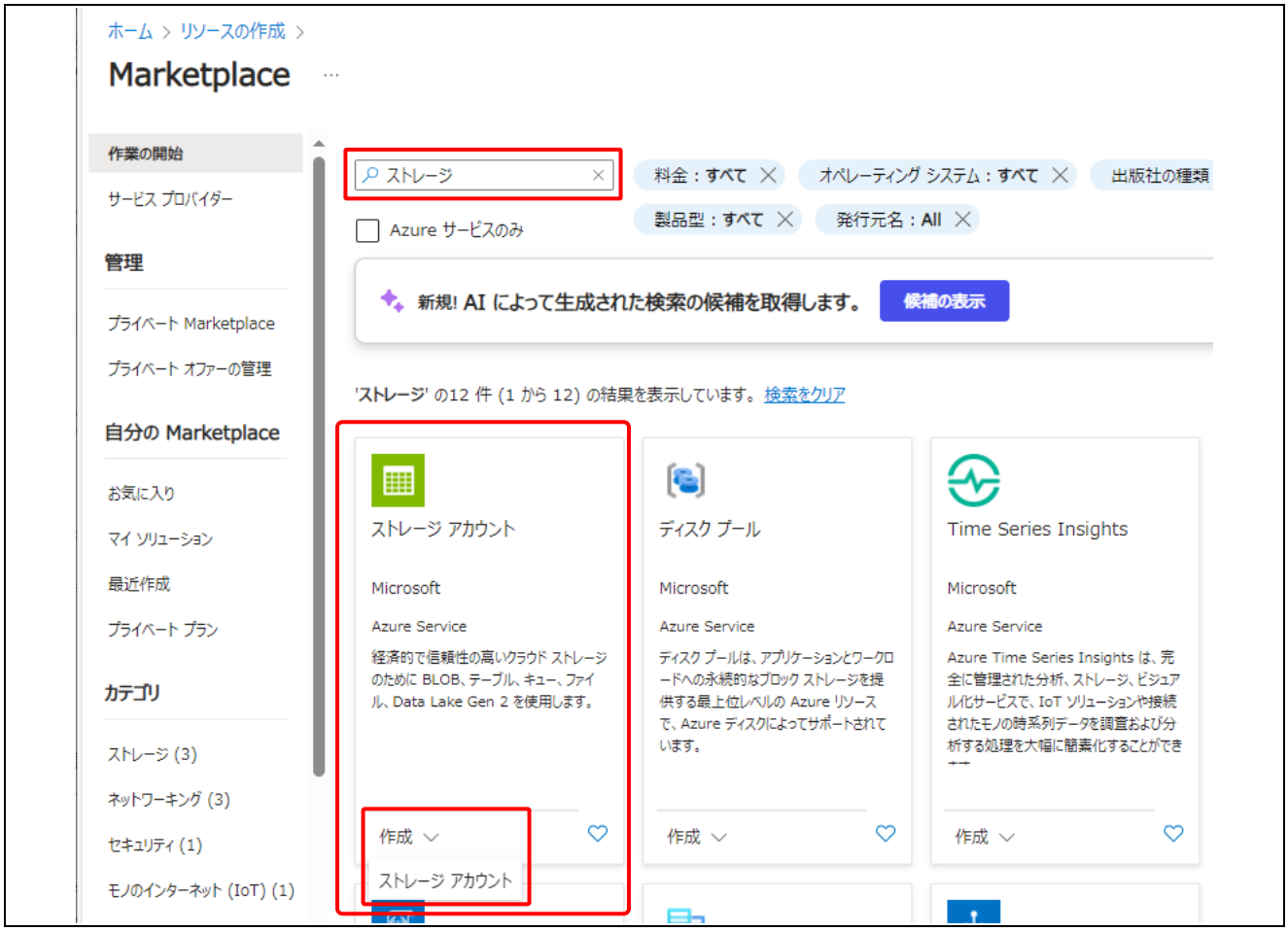

図 3-9 ストレージアカウントの作成

(2)ストレージアカウントの作成画面で以下の項目を入力してください。設定が完了したら画面下の[次へ:詳細設定]ボタンをクリックして下さい。

①	サブスクリプション	IoT Hub を作成した際のサブスクリプション ID を指定する
②	リスースグループ	IoT Hub を作成した際のリソースグループを指定する
③	ストレージアカウント名	ストレージアカウント名を任意の文字列で入力する
④	地域	ストレージアカウントが配置される Azure の地域となる

※その他の設定は初期値のままとしてください。

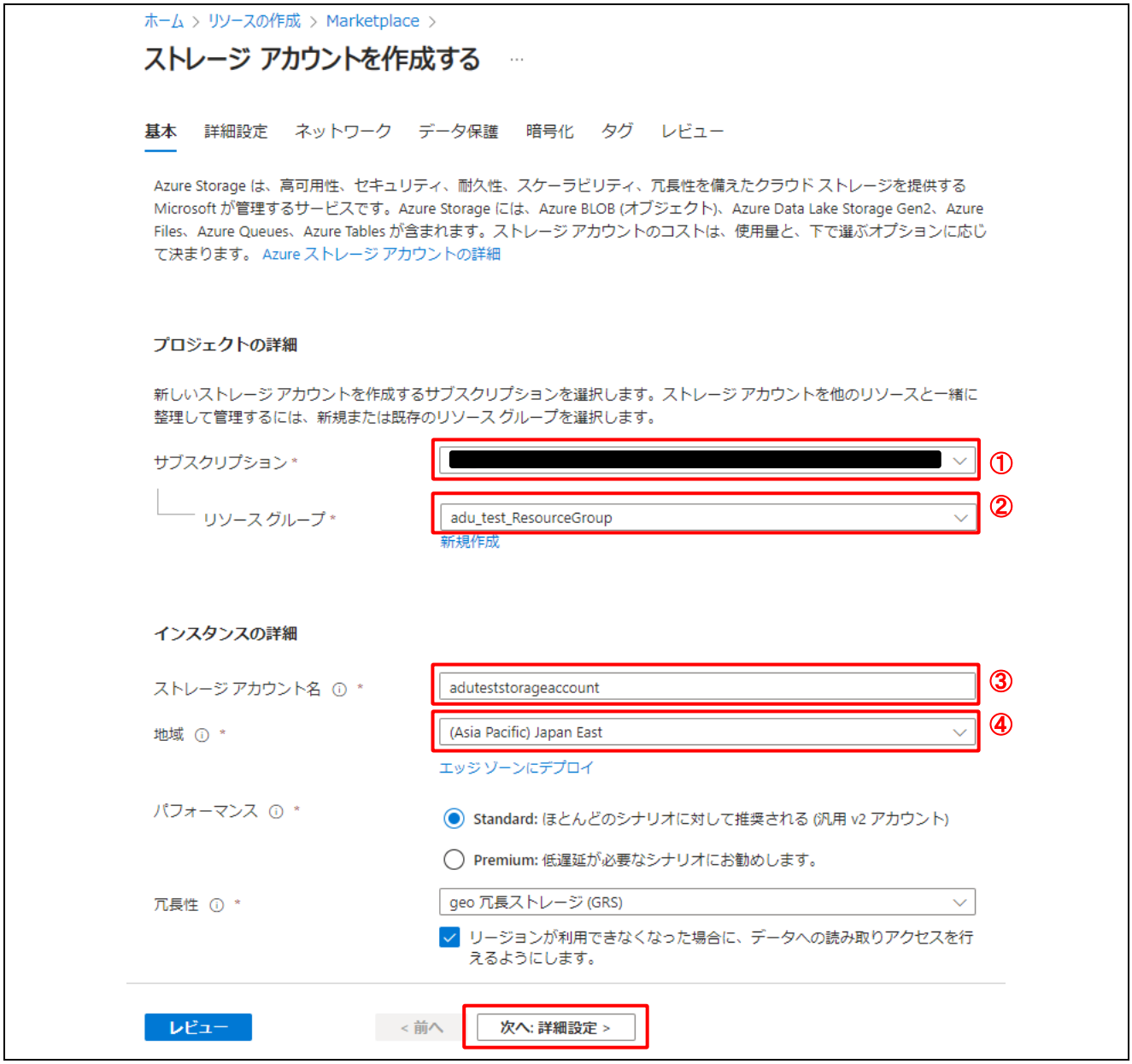

図 3-10 ストレージアカウントを作成する

(3) “詳細設定”、”ネットワーク”・”データ保護”・”暗号化”・”タグ”の各タブの各設定画面は初期値のままとなります。画面下部の[次へ]ボタンをクリックしてください。”タグ”画面で[次へ：レビュー]ボタンをクリックすると以下の確認画面が表示されます。
確認画面では登録内容の検証が行われます。検証が完了すると画面下部の”作成”ボタンがクリックできるようになります。
設定した内容に問題がないことを確認したら[作成]ボタンをクリックしてください。ストレージアカウントの作成が開始されます。

図 3-11 ストレージアカウントを作成する　設定確認画面

(4)ストレージアカウントの登録（デプロイ）にはしばらく時間がかかります。登録が完了したら[リソースに移動]ボタンをクリックしてください。作成したストレージアカウント画面が表示されたら登録は完了です。

図 3-12 ストレージアカウントの登録

(5) 次に更新ファームウェアのファイルを格納する、コンテナーを作成します。
ストレージアカウント画面より左側メニューの[データストレージ]⇒[コンテナー]をクリックしてください。コンテナー画面が表示されます。

図 3-13 ストレージアカウントの登録完了

(6)コンテナー画面で[＋コンテナー]をクリックしてください。新しいコンテナーの作成画面が表示されるので、任意のコンテナー名を入力し、[作成]ボタンをクリックしてください。そのほかの設定は初期値のままとしてください。一覧に作成したコンテナー名が表示されたら完了です。

図 3-14 新しいコンテナーの作成

3.4 更新ファームウェアの準備

3.4.1 更新ファームウェアのビルド

「2.サンプルプロジェクトの構築」の手順で更新ファームウェアを作成し、RSU ファイルを作成してください。

3.4.2 マニフェストファイルの作成

マニフェストファイルは IoT Hub のデバイス更新に必要な更新プログラムに関する情報を定義する JSON ファイルです。
IoT Hub へアップロードする更新ファームウェアはマニフェストファイルとバイナリファイルをペアで使用します。
以下の手順でマニフェストファイルを作成してください。

① マニフェストファイルの作成には PowerShell v7.0 を使用します。ご使用の OS に合わせてインストーラーをダウンロードしインストールしてください。
※通常は v7 の最新リリース版をダウンロードしてください。以下画面は v7.3.4 の例です。

図 3-15 PowerShell のダウンロード

② PowerShell を実行し、カレントディレクトリを ADU プロジェクトのマニフェストファイル作成スプリクトのディレクトリへ移動します。
\adu_sample\tools\AzureDeviceUpdateScripts

③ PowerShell で以下のコマンドを実行します。
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope Process

④ 「②」のフォルダに"2.15 更新ファームウェアの作成"で作成した RSU ファイルをコピーしてください。コピーした RSU ファイルは以下のファイル名に設定してください。
firmware_1.1.0.rsu
ファイル名の"1.1.0"は設定した更新ファームウェアのバージョンに合わせてください。また、マニフェストファイル作成後はファイル名の変更は行わないでください。

⑤ 次のスクリプトを実行してください。実行すると入力が必要な項目があるので、青文字の部分を入力してください。
スクリプト名はご使用のターゲットボードにより異なります。スクリプトファイル名を確認して適宜読み替えてください。
LeafPath の項目は、ターゲットボードに接続されている子デバイスのパス設定のため、何も入力しないで ENTER のみ入力してください。

```
.\CreateCKRX65NUpdate.ps1

cmdlet CreateCKRX65NUpdate.ps1 at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
Version: 1.1.0
HostPath: ./firmware_1.1.0.rsu
LeafPath:
Preparing update RENESAS.CK-RX65N.1.1.0 ...
Preparing parent update RENESAS.CK-RX65N.1.1.0 ...
Generating an import manifest RENESAS.CK-RX65N.1.1.0...
        Saving parent manifest file and payload(s) to .\RENESAS.CK-RX65N.1.1.0...
```

```
cmdlet CreateCKRX65NUpdate.ps1 at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
Version: 1.1.0
HostPath: ./firmware_1.1.0.rsu
LeafPath:
Preparing update RENESAS.CK-RX65N.1.1.0 ...
Preparing parent update RENESAS.CK-RX65N.1.1.0 ...
Generating an import manifest RENESAS.CK-RX65N.1.1.0...
    Saving parent manifest file and payload(s) to .¥RENESAS.CK-RX65N.1.1.0...
```

図 3-16 スクリプト実行画面

⑥ 正常にスクリプトが実行されたら、スクリプトのフォルダ下に"RENESAS.CK-RX65N.1.1.0"フォルダが作成され、以下の RSU ファイルとマニフェストファイルが格納されます。
これら 2 つのファイルをストレージコンテナーにアップロードします。
・RENESAS.CK-RX65N.1.1.0.importmanifest.json
・firmware_1.1.0.rsu

3.5 更新ファームウェアのストレージコンテナーへのアップロード

作成した更新ファームウェアをストレージコンテナーへアップロードします。
Azure ポータルのホーム画面より以下の手順を行ってください。

ホームより[ストレージアカウント]⇒[使用するストレージアカウント名]⇒データストレージ[コンテナー]⇒[使用するコンテナー名]をクリックしてください。コンテナー画面が表示されます。
コンテナー画面で[アップロード]をクリックすると更新ファイルのアップロード画面（BLOB のアップロード）が表示されます。

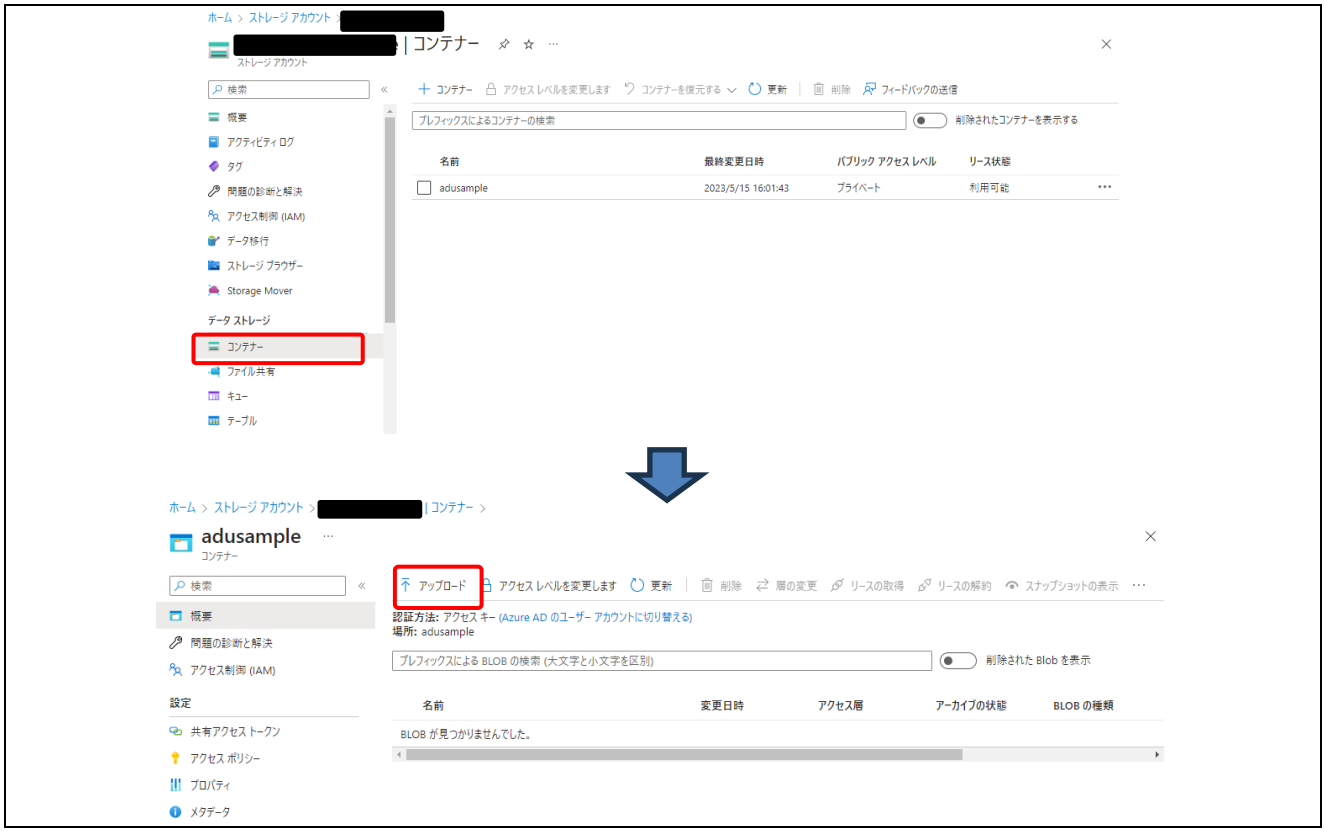

図 3-17 コンテナー画面

BLOB のアップロード画面にて、3.4 節で準備した更新ファームウェアのバイナリファイルとマニフェストファイルが格納された"RENESAS.CK-RX65N.1.1.0"フォルダをドラッグアンドドロップします。
ファイルが登録されたら、[アップロード]ボタンを押してください。アップロード後は[×]ボタンを押下して BLOB のアップロード画面を閉じてください。追加したフォルダ（またはファイル）がコンテナーの一覧リストに登録されます。

図 3-18 更新ファイルのアップロード

3.6 更新ファームウェアの登録

IoT Hub へ、ストレージコンテナーにアップロードした更新ファームウェアを登録します
使用する IoT Hub からデバイス管理[更新プログラム]をクリックすると更新プログラム一覧画面が表示されます。ここで[更新プログラムタブ]⇒[更新プログラムのインポート]をクリックしてください。

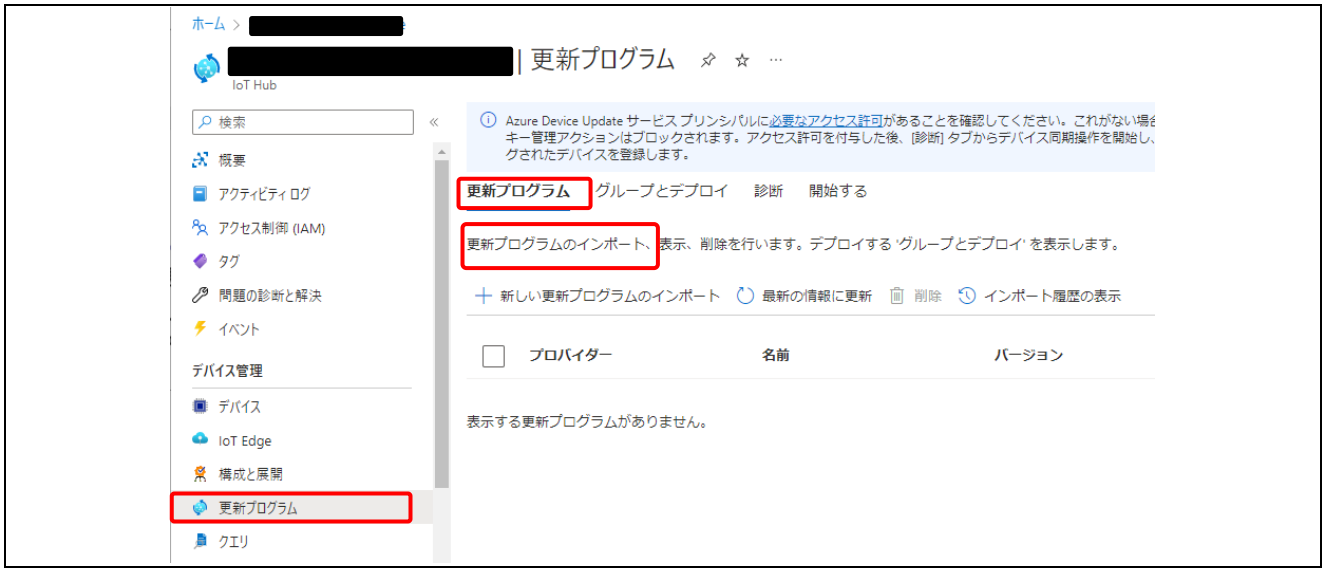

図 3-19 更新プログラム一覧画面

更新プログラムのインポート画面が表示されるので、説明ラベルに更新ファームウェアの内容を記入し、[ストレージコンテナーから選択]をクリックします。

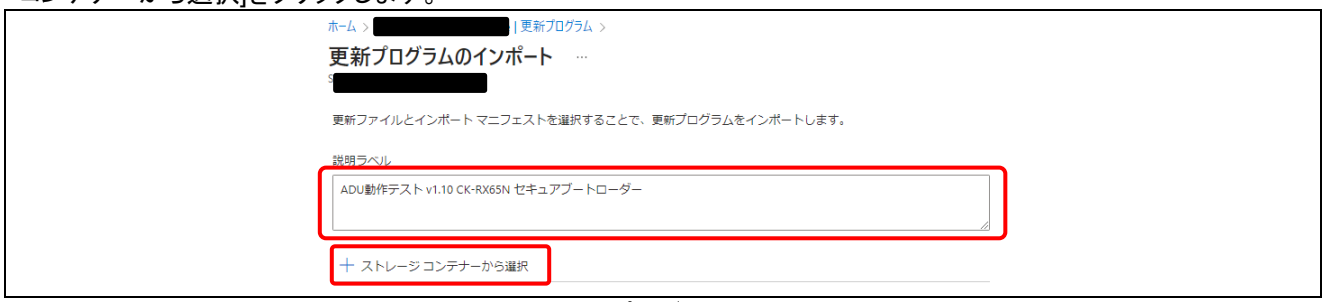

図 3-20 更新プログラムの説明入力

更新ファームウェアを格納した、[ストレージアカウント名]⇒[コンテナー名]をクリックします。
コンテナー画面より登録したいファームウェアのバイナリファイルとマニフェストファイルをチェックし、[選択]ボタンを押下します。

図 3-21 更新プログラムの選択

更新プログラムのインポート画面にチェックしたファームウェアのファイルが登録されるので[更新プログラムのインポート]ボタンを押下してください。

図 3-22 更新プログラムのインポート

インポート開始時に画面右上に以下のような警告のポップアップが表示されることがあります。

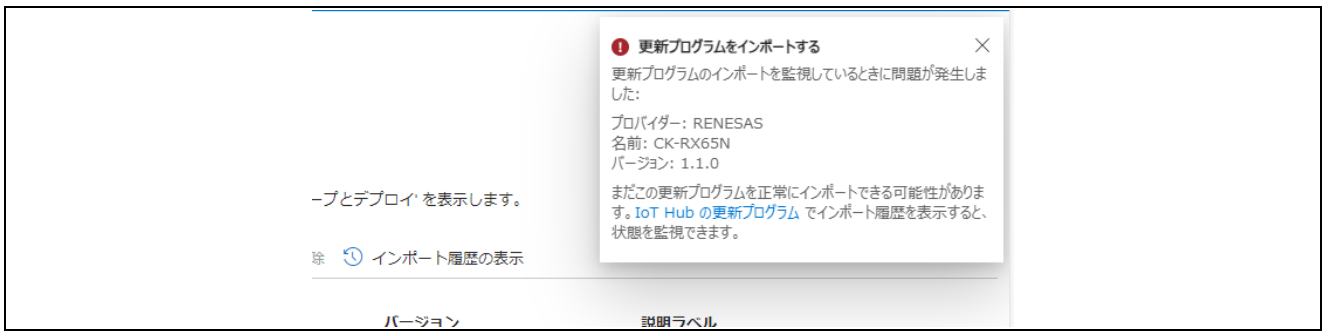

図 3-23 インポートの警告ポップアップ

この場合でも実際にはインポート処理は実行されているので、IoT Hub の更新プログラム画面の[インポート履歴の表示]をクリックしてインポートの状況を確認してください。

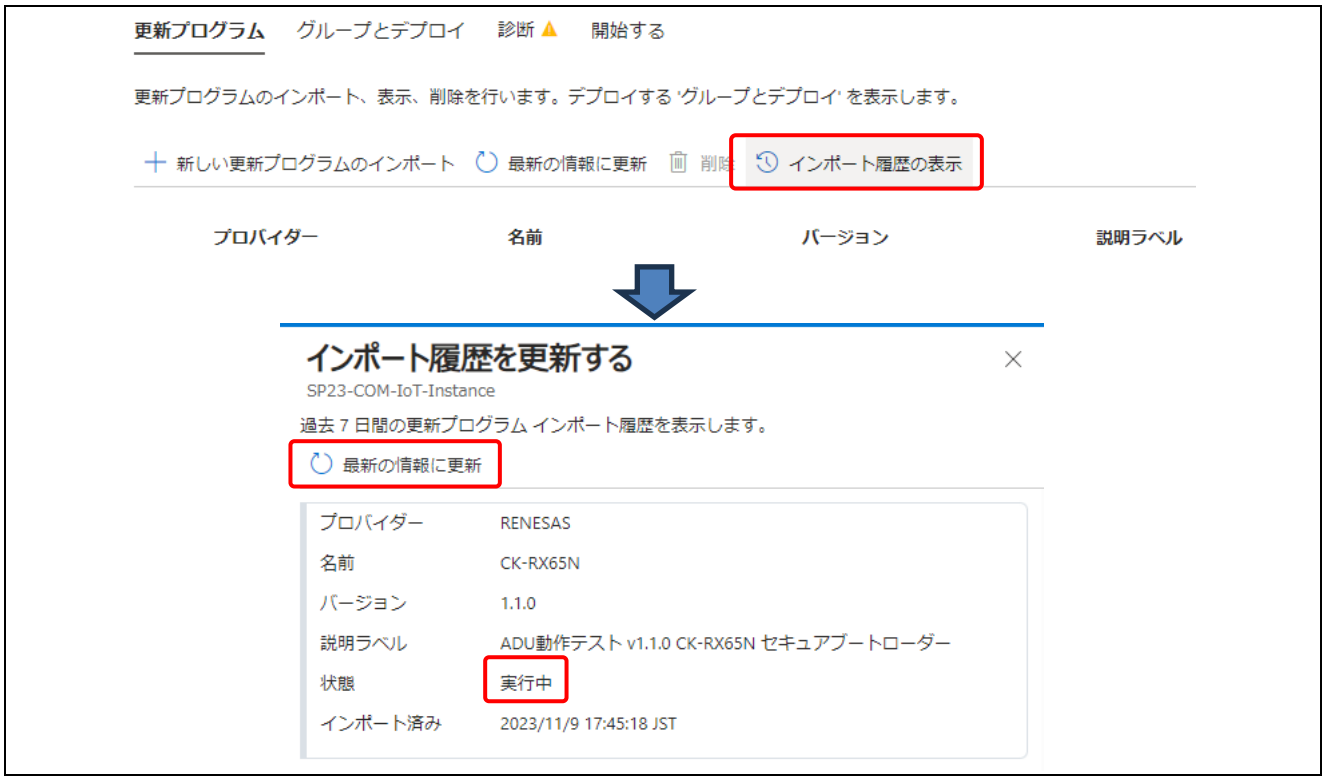

図 3-24 インポート履歴の表示

インポートにはしばらく時間がかかります。
状態の表示が"実行中"の時はインポートが進行しているのでしばらくお待ちください。
また、インポートの履歴を更新する画面は自動で表示が更新されないため、適宜、[最新の状態に更新]をクリックして画面を更新してください。
状態が"成功"となったらインポートは終了です。失敗となる場合は問題を解決してからインポートを再試行してください。

正常にインポートが完了後、[最新の状態に更新]をクリックすると、更新プログラムの一覧にインポートしたファームウェアが追加されます。

図 3-25 更新プログラムの一覧

3.7 ADU グループの作成

更新プログラムに ADU グループを追加します。
IoT Hub のデバイスと ADU グループはリンクしているため、IoT Hub のデバイスに対して ADU グループを追加する設定を行います。
IoT Hub よりデバイス管理[デバイス]をクリックしてください。デバイスの一覧より 3.1 節で作成したデバイスを選択しデバイスの詳細設定画面を開いてください。
デバイスの詳細設定画面からタグ[編集]をクリックするとタグの編集画面が表示されます。

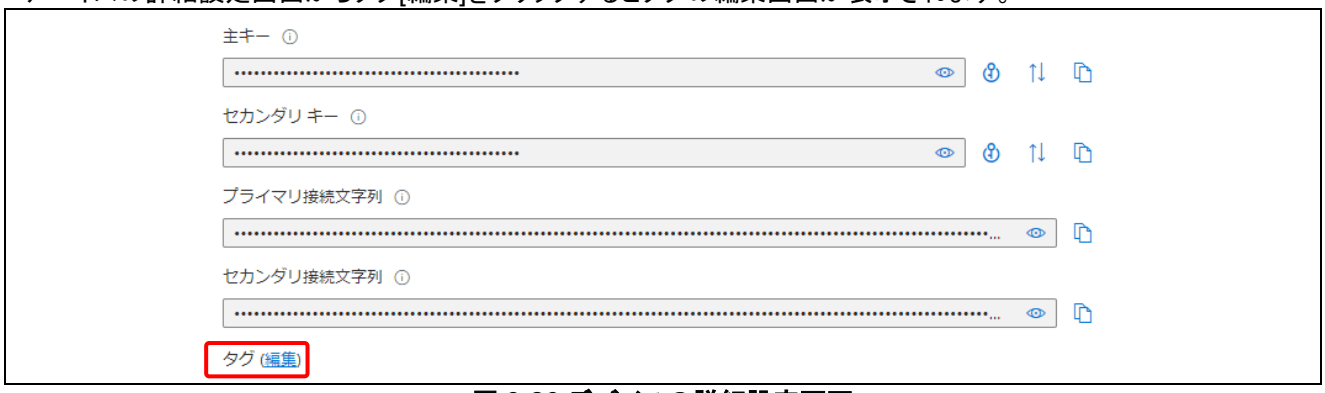

図 3-26 デバイスの詳細設定画面

タグの編集画面で、"名前"と"値"を入力します。名前は「ADUGroup」と入力し、値は任意の文字を入力してください。
入力したら[保存]ボタンを押下してタグ編集画面を閉じてください。
デバイス詳細設定画面でタグに設定した名前と値が登録されていることを確認したら画面左上の[保存]ボタンを押下します。

図 3-27 タグ編集画面

IoT Hub からデバイス管理[更新プログラム]⇒[グループとデプロイ]をクリックしてください。
グループの一覧が表示されるのでタグの"値"に設定されたグループが追加されています。(※)

(※)デバイスに対して一度もターゲットボードと通信を行っていない場合は、ADU グループの一覧に追加されない場合があります。この場合は、一度ターゲットボードから ADU を実行する IoT Hub のデバイスへ接続を行ってください。

3.8 ファームウェアの更新

IoT Hub へ登録した更新ファームウェアをターゲットボードにダウンロードします。

3.8.1 ターゲットボードの実行

まずは、初期ファームウェアを書き込んだターゲットボードを実行してください。
CK-RX65N の場合、JP16 を RUN 側にジャンパーして J14 の USB を PC の USB ポート等と接続し、電源を供給することで動作開始します。

また、J20 の USB を PC と接続することで Tera Term 等のターミナルソフトで動作状態をモニターすることができます。
J20 を PC と接続すると、Windows のデバイスマネージャーのポートに登録されるので、登録された COM ポート番号をターミナルソフトに設定してターゲットボードと接続してください。
シリアルポートの通信設定は以下に設定してください。

- ボーレート:115200bps
- データビット:8
- ストップビット:1
- パリティ:なし

プロジェクトを実行すると、サンプルプログラムは以下の画面例のようにステータス情報をターミナルへシリアル出力します。(※)
ターミナルの表示を確認し、IoT Hub・デバイスへ正常に接続していることを確認します。
また、"Installed Criteria:"の表示で現在のファームウェアバージョンを確認してください。

(※)ターミナルへ出力される情報はビルドしたプロジェクトにより異なる場合があります。

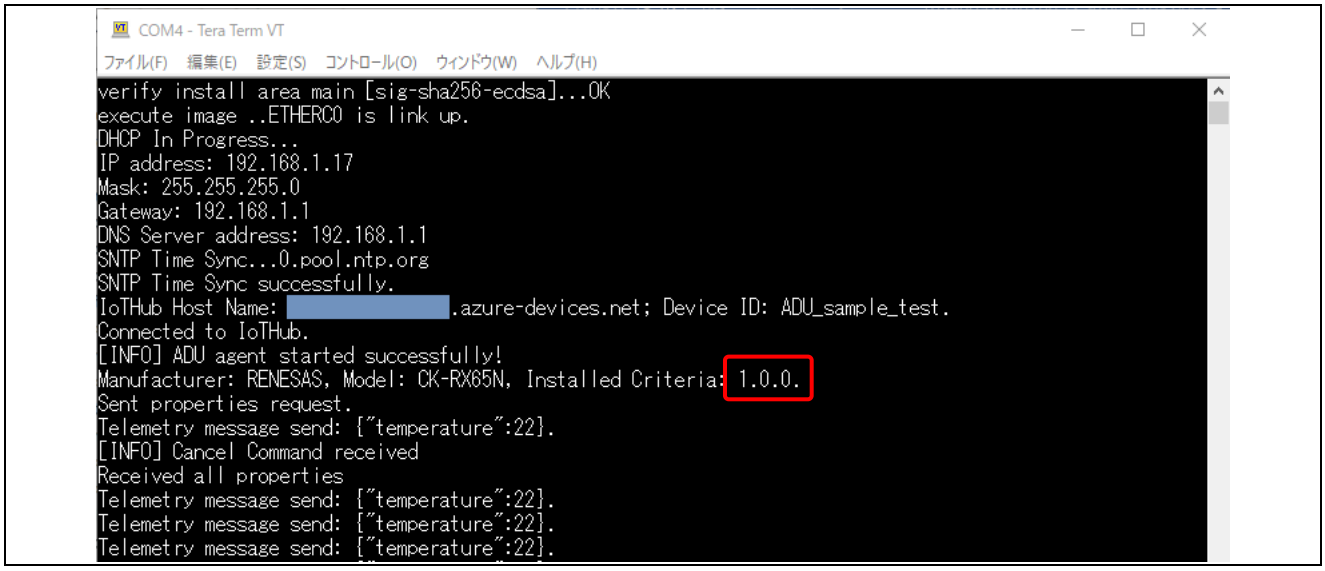

図 3-28 ターミナル出力画面(例)

3.8.2 更新ファームウェアのデプロイ

IoT Hub から更新ファームウェアを環境にデプロイ(配置)し、ターゲットのファームウェアを更新します。
使用する IoT Hub からデバイス管理[更新プログラム]をクリックし、更新プログラム一覧画面より、[グループとデプロイ]をクリックしてください。デバイスグループの一覧画面が表示されます。

図 3-29 グループとデプロイの表示

デバイスグループの一覧に、3.6 節で登録した ADU グループの名前でグループが表示されるので、ファームウェア更新に使用するグループ名をクリックしてください。グループの詳細画面が表示されます。
グループの詳細画面で[グループの基本]タブをクリックし、画面中央の[デプロイする]をクリックしてください。また、グループとデプロイの表示画面で使用するグループ名の横の[デプロイする]をクリックでもデプロイ可能です。

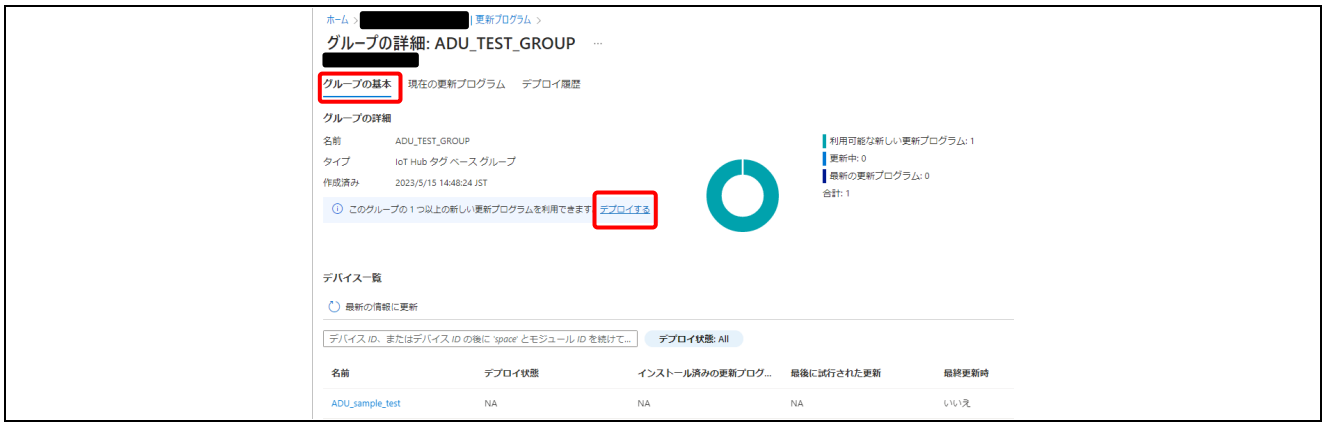

図 3-30 グループの詳細画面

新しい更新の画面が表示されるのでバージョンを確認し、[デプロイ]ボタンを押下してください。
デプロイの作成画面が表示されます。

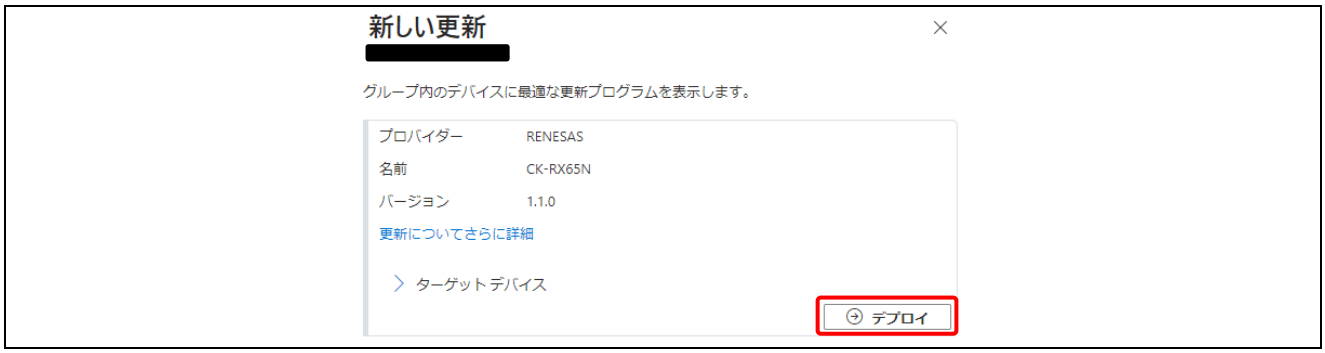

図 3-31 新しい更新画面

デプロイの作成画面で[作成]ボタンを押下してください。デプロイが開始されます。

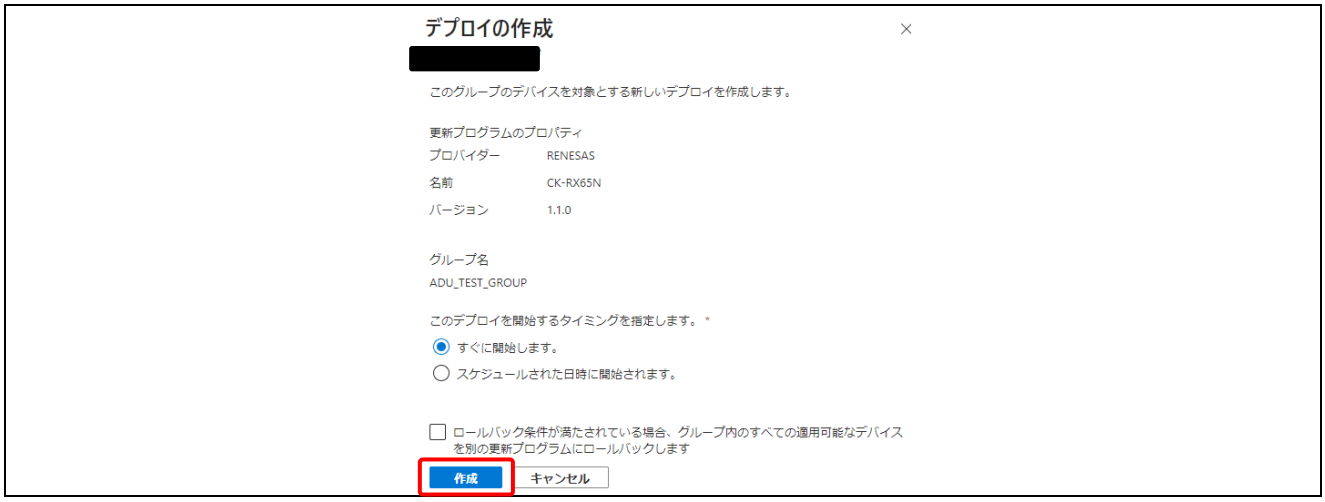

図 3-32 デプロイの作成画面

デプロイが開始されると CK-RX65N のターミナルにてダウンロード状況が出力されます。
ダウンロードされた更新ファームウェアはフラッシュメモリへ書き込まれ、バージョン照合・Verification data によるファームウェアの検証が行われます。その後デバイスのバンクススワップが実行され、ソフトウェアリセット後に更新ファームウェアが適用されます。

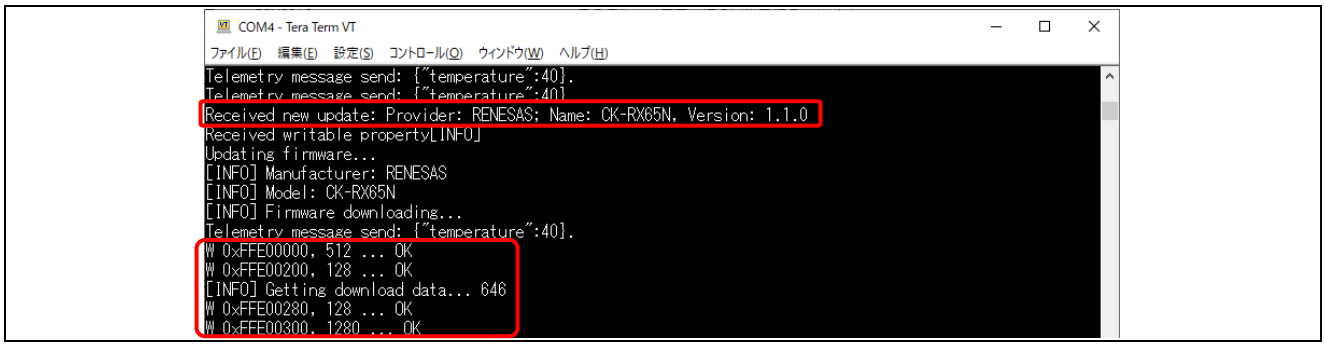

図 3-33 ファームウェアのダウンロード

ファームウェアのダウンロード後、"Installed Criteria:"の表示が更新ファームウェアのバージョンとなれば更新完了です。

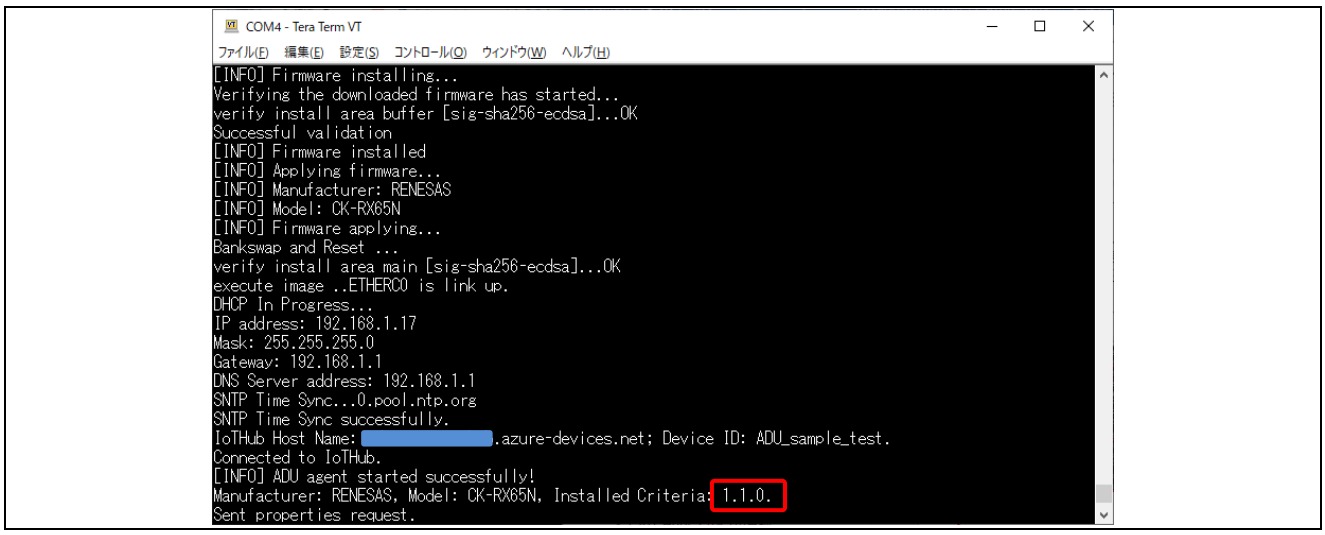

図 3-34 ファームウェアの更新完了

ファームウェアの更新が完了したら、グループの詳細画面でデプロイの状態が"成功"となります。

図 3-35 デプロイの作成画面

最新と異なるバージョンをデプロイしたい場合はデバイスグループ一覧からグループ名をクリックし、グループの詳細画面の[現在の更新プログラム]タブをクリックしてデプロイの詳細表示から[新しい更新プログラムを入手できます]をクリックすると"使用できる更新プログラム"画面が表示されるので、リストから必要なバージョンを選択し、デプロイを実行することができます。

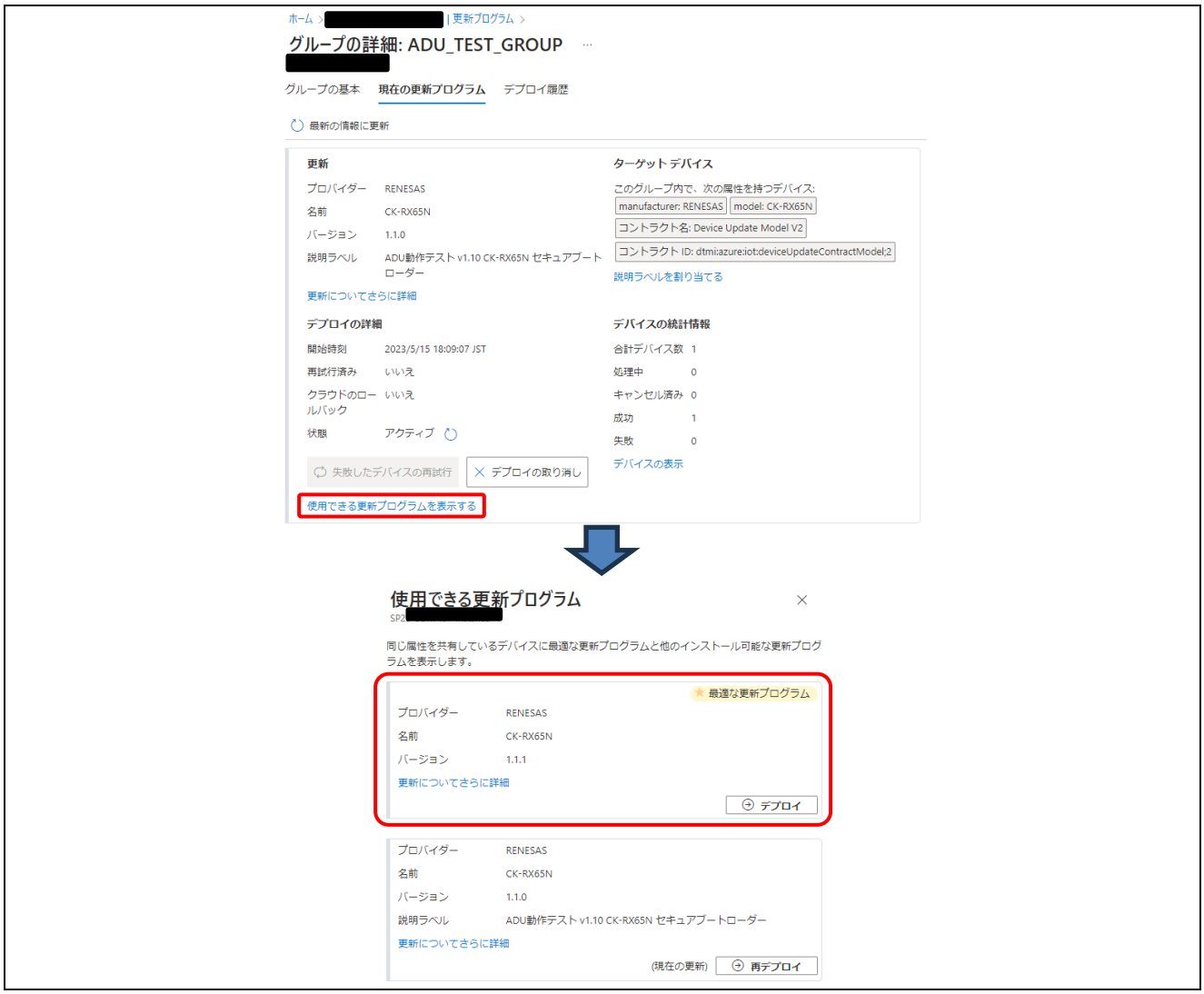

図 3-36 使用できる更新プログラムの選択

4. Appendix

4.1 Azure ADU とサンプルプロジェクトのコマンド制御

本サンプルプロジェクトは Azure からのコマンドを受信し、ファームウェアアップデートを行うための様々な制御を行います。これらを制御するアプリケーションを ADU エージェントと呼び、本サンプルプロジェクトの主となる。制御を行っています。

ADU エージェントは Azure からの要求コマンドを受信することで各処理が実行されます。
本サンプルプロジェクトが対応している要求コマンドの一覧を以下に示します。

コマンド	本サンプルプロジェクトでの機能	対応
INITIALIZE	ファームウェアアップデートモジュール等を初期化する	○
PREPROCESS	本サンプルプロジェクトでは実行する処理はなし	○
WRITE	Azure から送信されたファームウェアデータをフラッシュに書き込みを行う	○
INSTALL	ファームウェアのインストールを実施。 フラッシュへ書き込みを行ったファームウェアの検証を行う 検証に失敗した場合は更新ファームウェアを削除する	○
APPLY	ファームウェアの適用を行う 本サンプルプロジェクトでは起動バンクの切り替え後、ソフトウェアリセットを行う	○
CANCEL	本サンプルプロジェクトは対応する機能なし	×
UPDATE_CHECK	本サンプルプロジェクトは対応する機能なし	×

また、上記要求コマンドはファームウェアアップデートモジュール v1 と v2 に対応するサンプルプロジェクトで同じものとなります。

4.2 ファームウェアのデバッグ手法

e² studio を使用して初期ファームウェアおよび更新ファームウェアをソースデバッグする方法を示します。

4.2.1 初期ファームウェアのデバッグ方法

1. e² studio の bootloader プロジェクトのデバッグ設定を下記の通り変更してください。

   メニューより[実行]⇒[デバッグの構成]をクリックし、デバッグの構成画面の左ペインより"bootloader Hardware Debug"をクリックします。
   [メイン]タブをクリックして、[C/C++アプリケーション]の参照ボタンから、"2.9 初期ファームウェアの作成"で作成した初期ファームウェアの"userprog.mot"を指定してください。

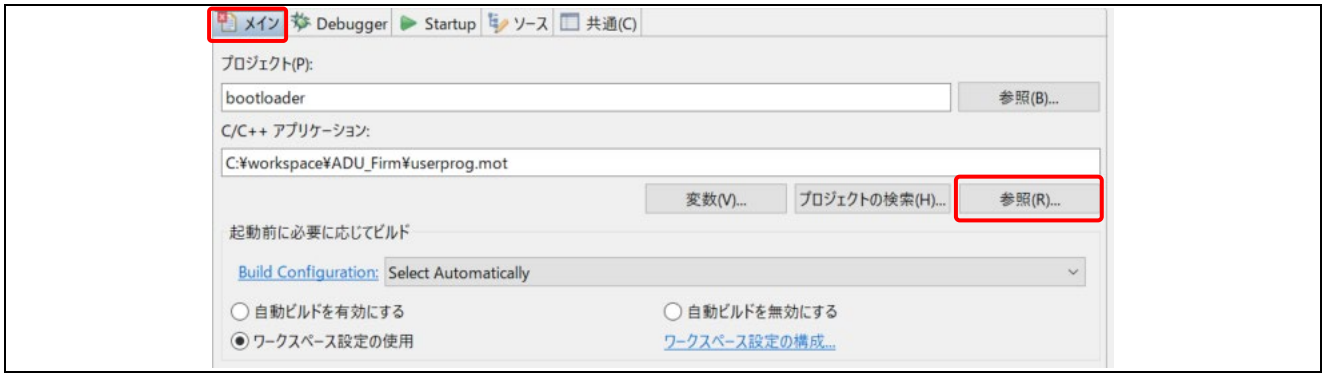

図 4-1 C/C++アプリケーションに userprog.mot を指定

[Debugger]タブ⇒[Connection Settings]をクリックし、”起動バンクを変更する”と、”はい”、"起動バンク"を"バンク 0"に設定します。

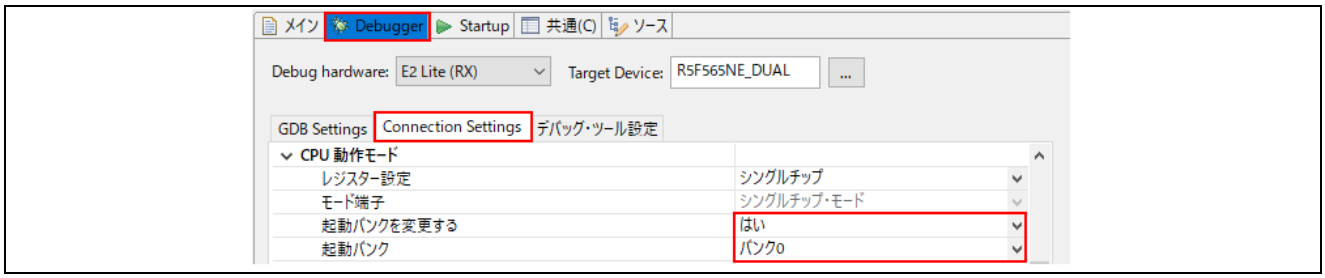

図 4-2 起動バンクを変更

[Debugger]タブ⇒[デバッグ・ツール設定]をクリックし、"内蔵プログラム ROM を書き換えるプログラムをデバッグする"を"はい"に設定します。

図 4-3 デバッグ・ツール設定

2. 同じく bootloader プロジェクトのデバッグ設定で[Startup]タブをクリックし、"イメージとシンボルをロード”を下記のように項目を追加して設定してください。
   - "userprog.mot"の"ロード・タイプ"を"イメージのみ"に変更
   - [追加]⇒[ファイル・システム...]から"adu_sample.x"を追加して"ロード・タイプ"を"シンボルのみ"に変更
   - [追加]⇒[ファイル・システム...]から"bootloader.x"を追加して"ロード・タイプ"を"シンボルのみ"に変更

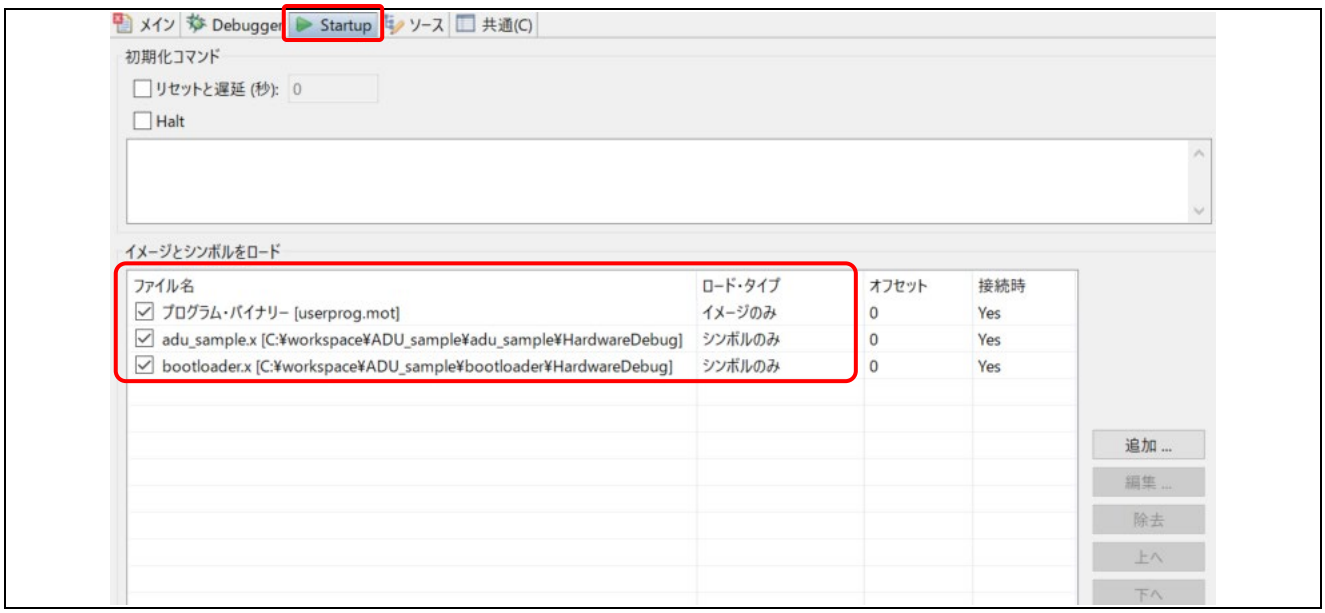

図 4-4 イメージとシンボルをロード を設定

3. bootloader プロジェクトのデバッグを開始して、bootloader/src/bootloader.c の main 関数でブレークすることを確認してください。以下の画面はファームウェアアップデートモジュール v1 の時の例です。

[bootloader.c]

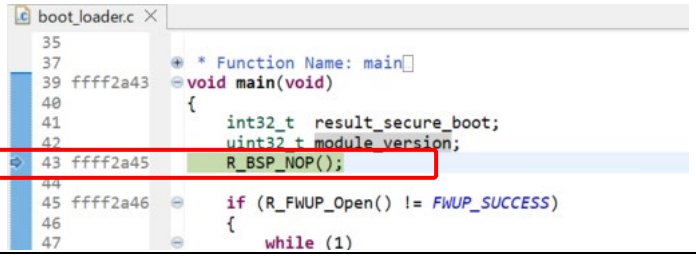

図 4-5 bootloader.c のブレークポイント

また adu_sample/src/main.c の main 関数でブレークすることで、ファームウェアが正常に起動することを確認してください。

[main.c]

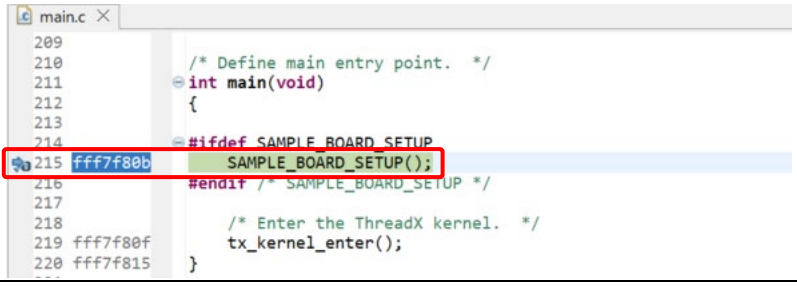

図 4-6 main.c のブレークポイント

main 関数でブレークしないときは上記ポイントにブレークポイントを設定してください。

なお、初期ファームウェア userprog.mot を生成する際、空き領域を 0xFF で充填した MOT ファイルを生成します。つまり、データフラッシュ領域を使用していなくても、この領域を 0xFF で充填することになります。このとき、プログラム中でデータフラッシュ領域のデータを書き換えて、再度ダウンロードしてデバッグ実行すると、0xFF で上書きすることになります。

この場合、初期ファームウェアをダウンロードする前に、書き換えたデータフラッシュ領域をコピーしておき、ダウンロード後に書き戻すことで対応してください。e² studio の場合、dump/restore コマンドを使用することでメモリ内容をファイルに出力し、ファイルから読み込むことが可能です。

例：0x100000-0x107FFF 番地のデータフラッシュ領域を S-format ファイルに出力して読み込む場合、Debugger Console ビューにて下記の GDB コマンドを実行してください。

- memdump.mot ファイルに出力する場合

```
dump srec memory memdump.mot 0x100000 0x107FFF
```

- memdump.mot ファイルからメモリに書き込む場合

```
restore memdump.mot 0 0x100000 0x107FFF
```

なお、memdump.mot はプロジェクトフォルダに生成されます。

4.2.2 更新ファームウェアのデバッグ方法

更新ファームウェアをデバッグする場合は、初期ファームウェアと更新ファームウェアを別のプロジェクトとして作成してください。
bootloader を実行することで、初期ファームウェアが実行されます。

1. 初期ファームウェア起動後、ブートローダがファームウェアを起動する直前にブレークポイントを設定してください。ブレークポイントの設定個所はファームウェアアップデートモジュール v1 と v2 で異なります。以下を参照してください。

   (a) ファームウェアアップデートモジュール v1 の時
   bootloader/src/smc_gen/r_fwup/src/r_fwup_boot_loader.c ファイルの下記処理がファームウェアを起動する処理です。

[r_fwup_boot_loader.c]

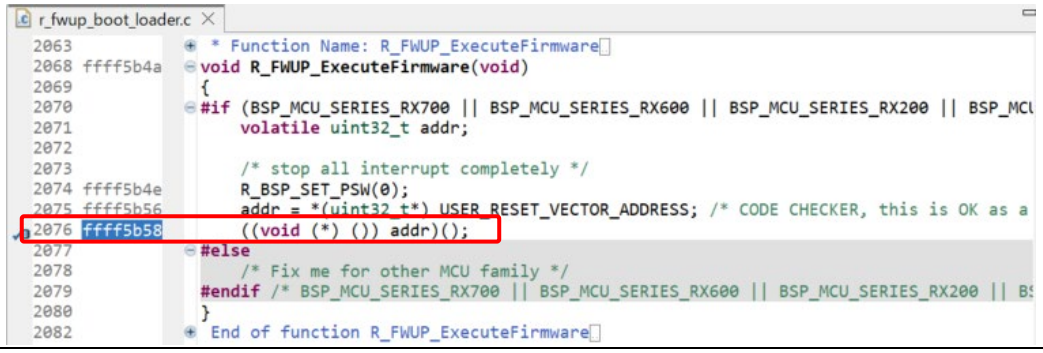

図 4-7 ブレークポイント設定場所（v1）

   (a) ファームウェアアップデートモジュール v2 の時
   bootloader/src/smc_gen/r_fwup/src/r_fwup.c ファイルの下記処理がファームウェアを起動する処理です。

[r_fwup.c]

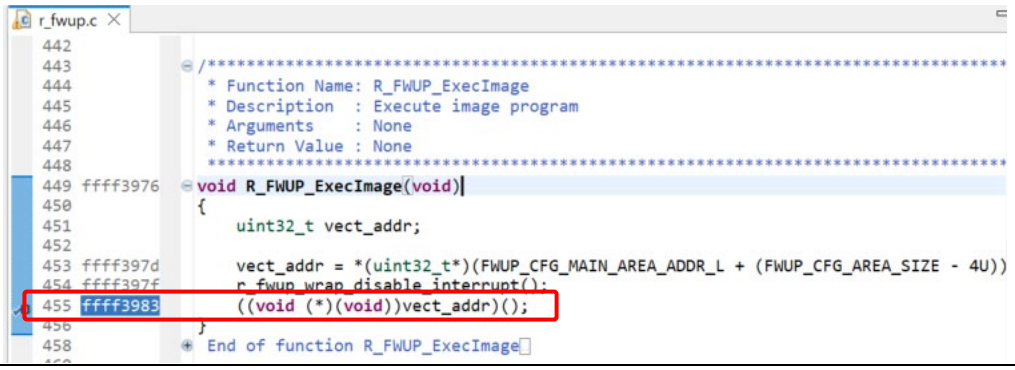

図 4-8 ブレークポイント設定場所（v2）

2. ADU によってファームウェアをアップデート後、バンクスワップとソフトウェアリセットが発生します。
3. 更新ファームウェアが起動する直前に、1.で指定したブレークポイントでブレークします。

4. ブレーク後に下記の GDB コマンドを実行してシンボル情報を更新してください。
   GDB コマンドの実行は e² studio 下部の"Debugger Console"で行ってください。
   例：更新ファームウェアの.x ファイルが C:\workspace\ADU_sample\HardwareDebug 内にある場合(※)
   以下は更新ファームウェアの x ファイル名が"ADU_sample.x"の場合の例です。

   symbol-file C:/\workspace/\ADU_sample/\HardwareDebug/\ADU_sample.x -readnow(※)

   (※)パスの"\"の前には"/"を付けてください。

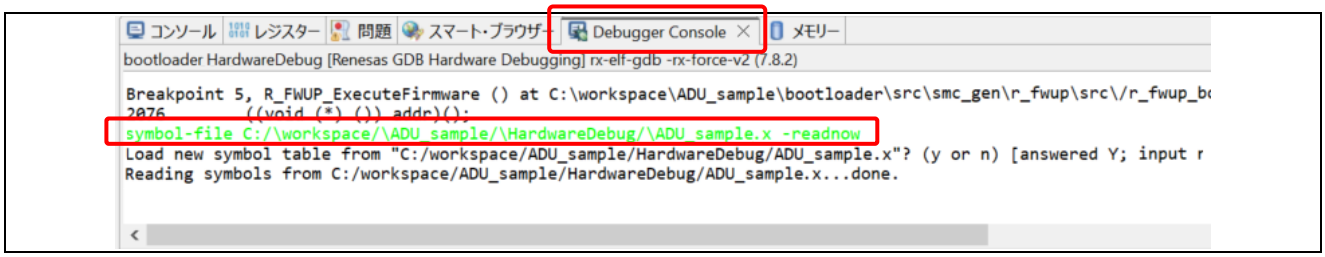

図 4-9 シンボル情報更新コマンド

5. 更新ファームウェアの任意のソース上にブレークポイントを設定してデバッグを再開すると、当該ブレークポイントでブレークすること確認してください。

Online technical support and information is available at: https://en-support.renesas.com/dashboard

Technical Contact Details: https://www.renesas.com/jp/ja/contact-us

5. 改定履歴

Rev.	発行日	改定内容	
		ページ	ポイント
1.00	2023.7.31	-	初版発行
2.00	2024.1.12	1	使用する開発環境を更新
		2	ターゲットボードとファームウェアアップデートモジュール（FWUP）の組み合わせ表を追加
		5-8	メモリ配置及び動作内容の説明を追加（FWUP v1/v2） データフラッシュのメモリ配置説明を追加
		11	指定する Azure RTOS のバージョンとターゲットボードの種類を変更。 DUAL モードの説明を追加
		15	BGO モードの説明追加
		15-18	不足コンポーネントの組み込みについての説明を変更 FWUP v2 の組み込み手順の追加
		19-20	ビルド時のコード生成設定に関する説明の追加 デバイスの DUAL 設定、セクションの設定、シンボルの設定変更が不要となったため手順を削除
		25-29 34-38	FWUP FIT v2 に対応するファームウェア作成手順を追加
		32	初期ファームウェア実行の項を修正
		39-47	IoT Hub 作成時の補足を追加 デバイスアップデート、ストレージの作成手順を追加
		52	ファームウェアのアップロード時のストレージコンテナーの警告メッセージについての説明追加
		59	Azure ADU のコマンド制御に関する説明の追加
		63	更新ファームのデバッグ方法に FWUP FIT v2 時の説明を追加

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。

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うと、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、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因し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いたします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。

© 2024 Renesas Electronics Corporation. All rights reserved.