RA ファミリ
開発チュートリアル

6 版　2024 年 12 月 02 日

1. 概要

1.1 概要

本アプリケーションノートは、弊社製 RA ファミリ CPU ボードを用いて、フレキシブル ソフトウェア パッケージ(FSP)を使用したプログラムを作成・動作確認するまでのチュートリアル資料です。
各機能に関する詳細は、サンプルプログラムのソースコードやサンプルプログラム解説、ハードウェアマニュアル、回路図をご覧ください。

1.2 使用環境

本アプリケーションノートの解説で用いる開発環境を以下に示します。

<AP-RA6M-0A/1A>

ソフトウェア	バージョン	備考
e2 studio	V2023-04	–
FSP	V4.5.0	Flexible Support Package
GCC ARM Embedded	V10.3.1.20210824	–
Renesas Flash Programmer	V3.13.02	評価版使用可能

デバッガ	ハードウェアバージョン	備考
J-Link	V10	Segger Microcontroller Systems 社

<AP-RA8M-0A / AP-RA8D-0A>

ソフトウェア	バージョン	備考
e2 studio	V2023-10	–
FSP	V5.1.0	Flexible Support Package
GCC ARM Embedded	V13.2.1.arm-13-7	–
Renesas Flash Programmer	V3.13.00	評価版使用可能

デバッガ	ハードウェアバージョン	備考
J-Link	V11	Segger Microcontroller Systems 社

<AP-RA8T-0A >

ソフトウェア	バージョン	備考
e2 studio	V2024-07	–
FSP	V5.5.0	Flexible Support Package
GCC ARM Embedded	V13.2.1.arm-13-7	–
Renesas Flash Programmer	V3.16.00	評価版使用可能

デバッガ	ハードウェアバージョン	備考
J-Link	V11	Segger Microcontroller Systems 社

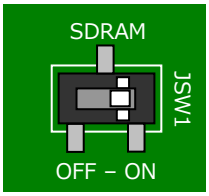
※AP-RA6M-0A/1A, AP-RA8M-0A, AP-RA8D-0A, AP-RA8T-0A, と J-Link を直接接続することはできません。
CPU ボード側(ハーフピッチコネクタ)と J-Link 側(フルピッチコネクタ)を接続するための変換アダプタが必要となります。
変換アダプタについては、J-Link 取扱店へご確認ください。

1.3　CPU ボード スイッチ設定

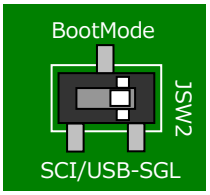
CPU ボードを動作させる際は、動作モードに応じてボード上のディップスイッチを設定する必要があります。

(1)　AP-RA6M-0A

出荷時設定：「ボード上の SDRAM を使用する」および「シングルチップモード」



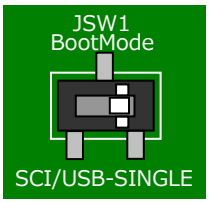
SDRAM
ON ：ボード上の SDRAM を使用する
OFF：ボード上の SDRAM を使用しない



BootMode
SCI/USB：SCI ブートモード・USB ブートモード
SGL　　：シングルチップモード
　　（プログラム動作時やデバッガ使用時）

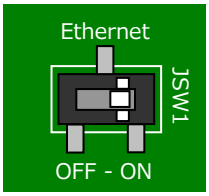
(2)　AP-RA6M-1A

出荷時設定：「シングルチップモード」



BootMode
SCI/USB：SCI ブートモード・USB ブートモード
SINGLE ：シングルチップモード
　　（プログラム動作時やデバッガ使用時）

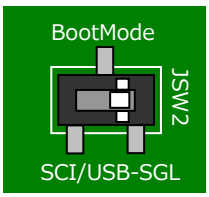
(3)　AP-RA8M-0A, AP-RA8D-0A



Ethernet

Ethernet を使用する場合: ON

CAMERA を使用する場合: OFF

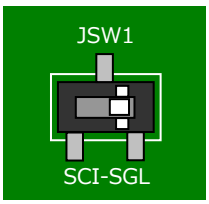


BootMode

SCI/USB：SCI ブートモード・USB ブートモード
SGL　　：シングルチップモード
　　（プログラム動作時やデバッガ使用時）

(4)　AP-RA8T-0A

出荷時設定：「シングルチップモード」



BootMode
SCI：SCI ブートモード・USB ブートモード
SGL：シングルチップモード
（プログラム動作時やデバッガ使用時）

1.4　各 CPU ボードの資料に関して

資料・サンプルプログラムは、弊社 Web サイトのボード紹介ページで公開されています。

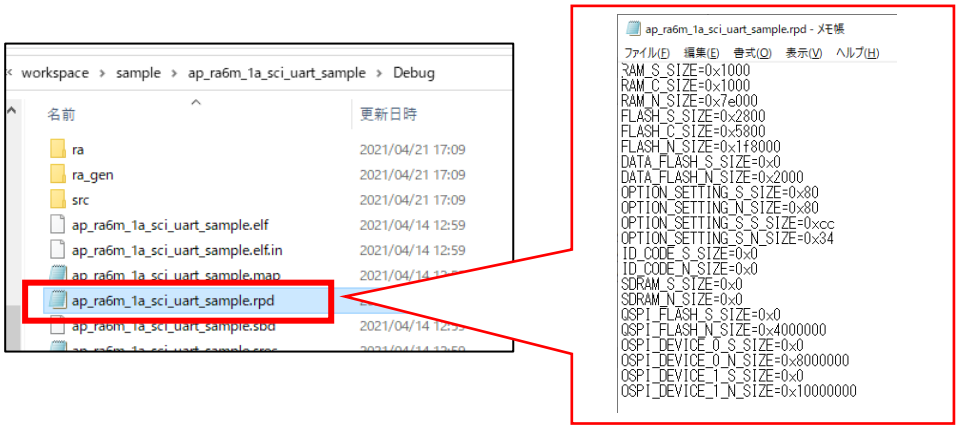
RA ファミリ CPU ボードシリーズ	https://www.apnet.co.jp/product/ra/index.html
AP-RA6M-0A　製品ページ	https://www.apnet.co.jp/product/ra/ap-ra6m-0a.html
AP-RA6M-1A　製品ページ	https://www.apnet.co.jp/product/ra/ap-ra6m-1a.html
AP-RA8M-0A　製品ページ	https://www.apnet.co.jp/product/ra/ap-ra8m-0a.html
AP-RA8D-0A　製品ページ	https://www.apnet.co.jp/product/ra/ap-ra8d-0a.html
AP-RA8T-0A　製品ページ	https://www.apnet.co.jp/product/ra/ap-ra8t-0a.html

1.5　各 CPU ボードの Arm TrustZone セキュリティ機能について

弊社製 RA ファミリ CPU ボードは搭載している CPU によって、Arm TrustZone セキュリティ機能（以下、セキュリティ機能と記述します）の有無があります。
各 CPU ボードのセキュリティ機能の有無を以下に示します。

セキュリティ機能	CPU ボード名(CPU 種類)	参考資料
なし	AP-RA6M-0A(RA6M3)	–
あり	AP-RA6M-1A(RA6M5) AP-RA8M-0A(RA8M1) AP-RA8D-0A(RA8D1) AP-RA8T-0A(RA8T1)	「RA6M5 グループ ユーザーズマニュアル　ハードウェア編」 「RA8M1 グループ ユーザーズマニュアル　ハードウェア編」 「RA8D1 グループ ユーザーズマニュアル　ハードウェア編」 「RA8T1 グループ ユーザーズマニュアル　ハードウェア編」 >「セキュリティ機能」章

セキュリティ機能がある CPU のプログラム開発を e2studio で行う際、デフォルトでセキュリティ機能のメモリセキュリティ属性の設定がビルド時に自動で行われます。（設定内容は Debug フォルダの rpd ファイルとして出力されます）



AP-RA6M-1A の UART サンプルプログラムの場合

また、メモリセキュリティ属性の設定はデフォルトで e2studio のプログラムダウンロード時に自動で CPU に書き込まれます。ただし、Renesas Flash Programmer（以下、RFP と記述します）でプログラムをダウンロードする際には、セキュリティ設定をユーザが行う必要があります。
本書の「5.1 Renesas Flash Programmer を使用した書き込み」では、メモリセキュリティ属性の設定を、全領域非セキュアとする場合の手順も記述してありますので、ご参考ください。

2.　機能

2.1　サンプルプログラムの使用機能

以下のサンプルプログラムを用いて、各機能の確認をすることができます。
（2024 年 2 月現在の公開データの状況です）

<AP-RA6M-0A>

ドキュメント番号	サンプルプログラム	機能
AN2003	標準サンプルプログラム	・CAN 通信 ・ネットワーク通信 ・QSPI FlashROM 読み書き ・UART 通信 ・SD カード読み書き ・USB ホスト メモリ読み書き ・USB ファンクション 仮想 COM 通信
AN2004	LCD-KIT サンプルプログラム	・LCD 出力 ・タッチパネル操作
AN2005	WM-RP-10 サンプルプログラム	・無線 LAN 通信
AN2006	wolfSSL サンプルプログラム	・セキュリティ（wolfSSL）ネットワーク通信

<AP-RA6M-1A>

ドキュメント番号	サンプルプログラム	機能
AN2007	標準サンプルプログラム	・CAN 通信 ・ネットワーク通信 ・UART 通信 ・SD カード読み書き ・USB ファンクション 仮想 COM 通信 ・無線 LAN （ESP32-WROOM-32E）制御

<AP-RA8M-0A>

ドキュメント番号	サンプルプログラム	機能
AN2009	標準サンプルプログラム	・CAN 通信 ・ネットワーク通信 ・UART 通信 ・SD カード読み書き ・USB ホスト メモリ読み書き ・USB ファンクション 仮想 COM 通信
AN2010	Pmod サンプルプログラム	・Pmod 動作 （GPIO/UART/SPI/I2C）

<AP-RA8D-0A>

ドキュメント番号	サンプルプログラム	機能
AN2009	標準サンプルプログラム	・CAN 通信 ・ネットワーク通信 ・UART 通信 ・SD カード読み書き ・USB ホスト メモリ読み書き ・USB ファンクション 仮想 COM 通信
AN2010	Pmod サンプルプログラム	・Pmod 動作 （GPIO/UART/SPI/I2C）
AN2011	LCD-KIT サンプルプログラム	・LCD 出力 ・タッチパネル操作
AN2013	LCD-KIT+カメラサンプルプログラム	・LCD 出力 ・CMOS カメラ入力

<AP-RA8T-0A>

ドキュメント番号	サンプルプログラム	機能
AN2015	標準サンプルプログラム	・CAN 通信 ・ネットワーク通信 ・UART 通信 ・USB ホスト メモリ読み書き ・USB ファンクション 仮想 COM 通信
AN2016	Pmod サンプルプログラム	・Pmod 動作 （GPIO/UART/SPI/I2C）

3.　e2 studio を用いた動作方法

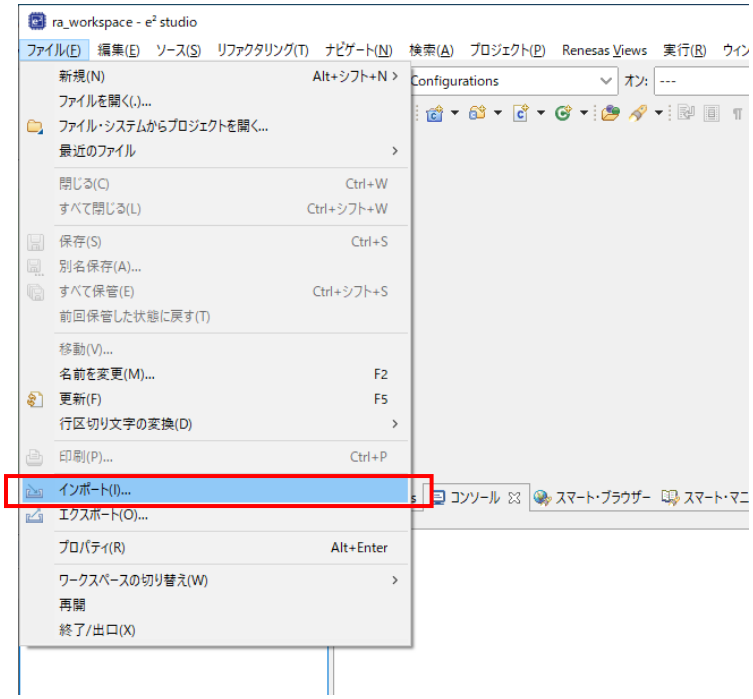
本章では、e2 studio を使用して CPU ボードを動作させる方法を説明します。
e2 studio および FSP（Flexible Software Package）はインストール済みであるものとしますので、事前にインストールを行ってください。

3.1　サンプルプログラムの動作方法

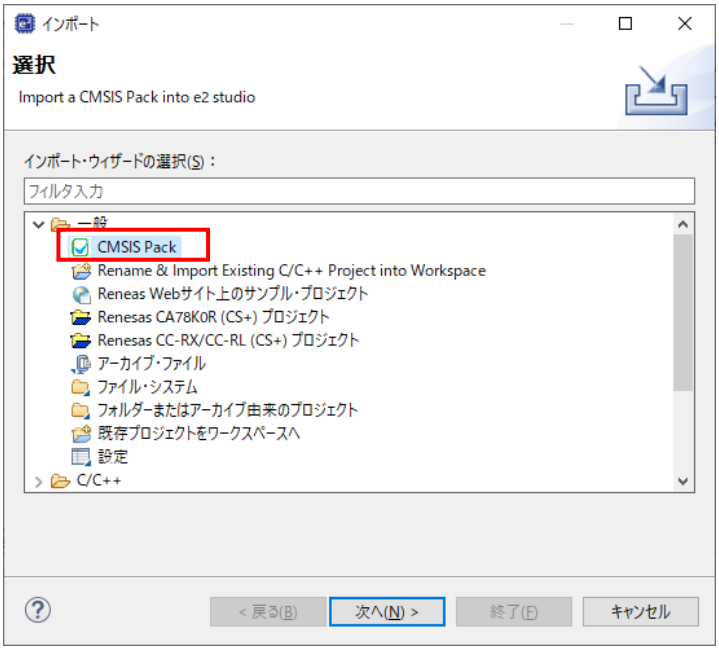
本節では、弊社で作成した AP-RA6M-0A サンプルプログラムを例に動作方法の説明をいたします。

3.1.1　インポート方法

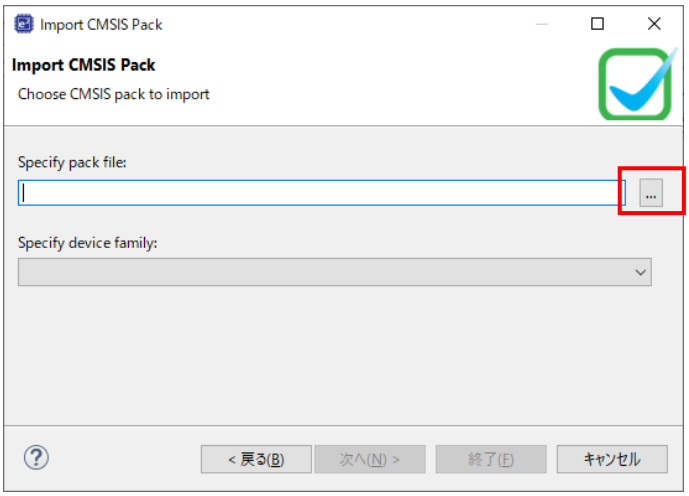
①　e2 studio を起動し、ツールバーの［ファイル］→［インポート］を選択します。



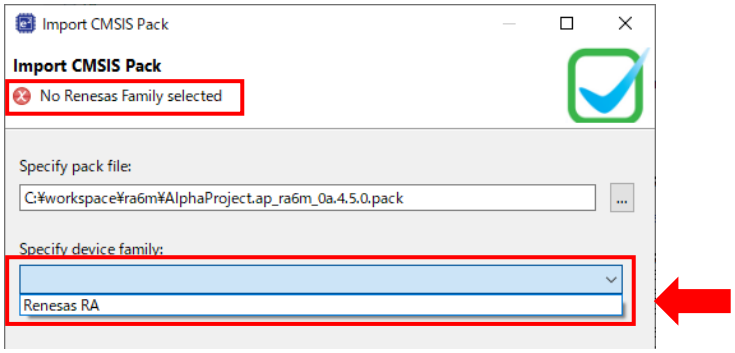
②　［CMSIS Pack］を選択し［次へ］を選択し、pack ファイルをインポートします。
（本書では例として「sample¥ CustomBSP¥AlphaProject.ap_ra6m_0a.4.5.0.pack」をインポートする図を示します）
すでに開発環境に pack ファイルをインポート済みである場合は、⑥へお進みください。



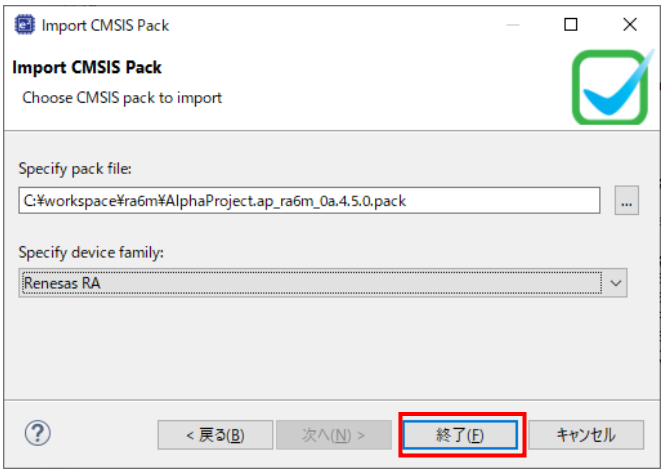
③　［Import RA CMSIS Pack ウィンドウ］が表示されましたら、インポートする pack ファイルを選択してください。



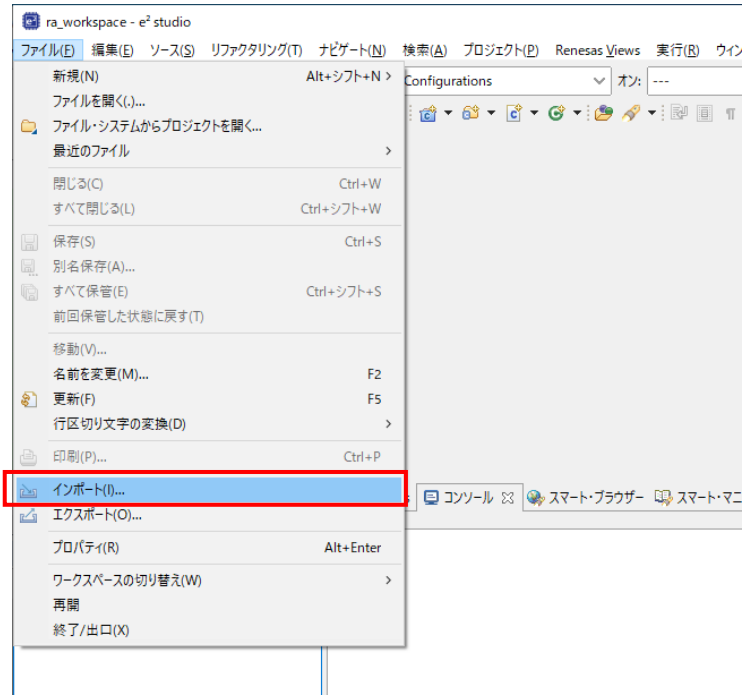
④　pack ファイルを選択後、メッセージ「No Renesas Family selected」が表示されるので、Specify device family から「Renesas RA」を選択してください。



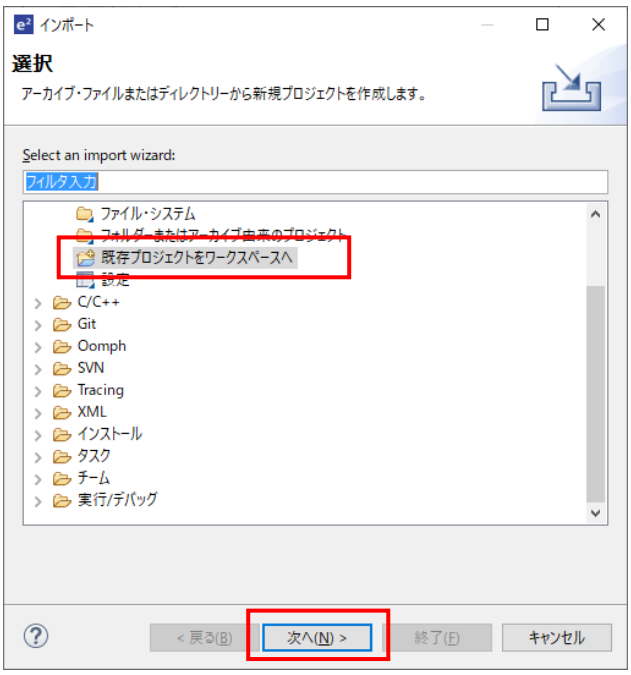
⑤　［終了］を選択してください。



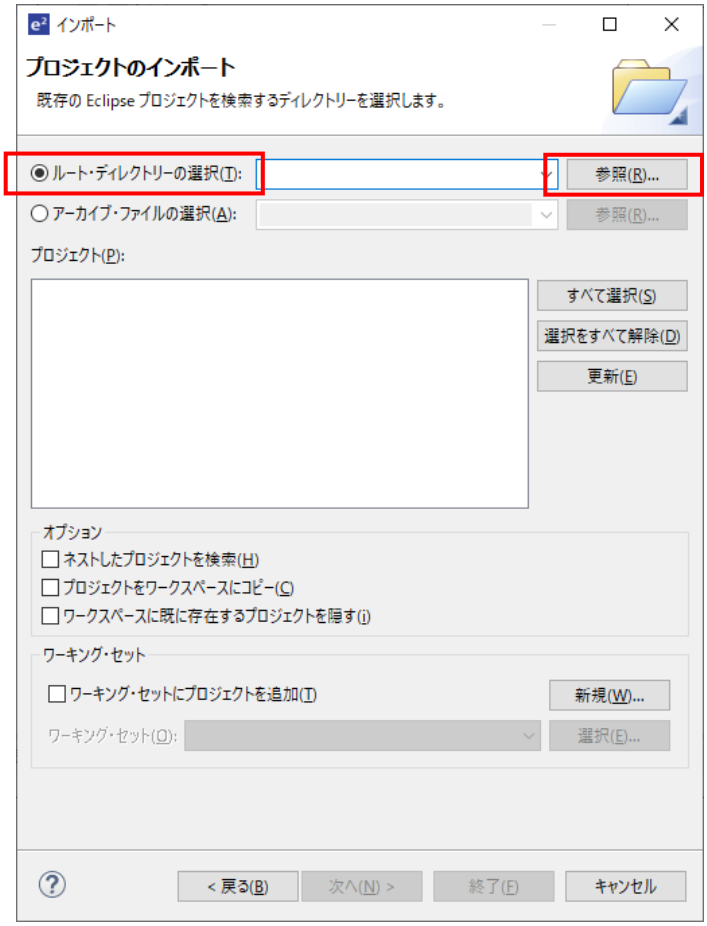
⑥　もう一度ツールバーの［ファイル］→［インポート］を選択します。



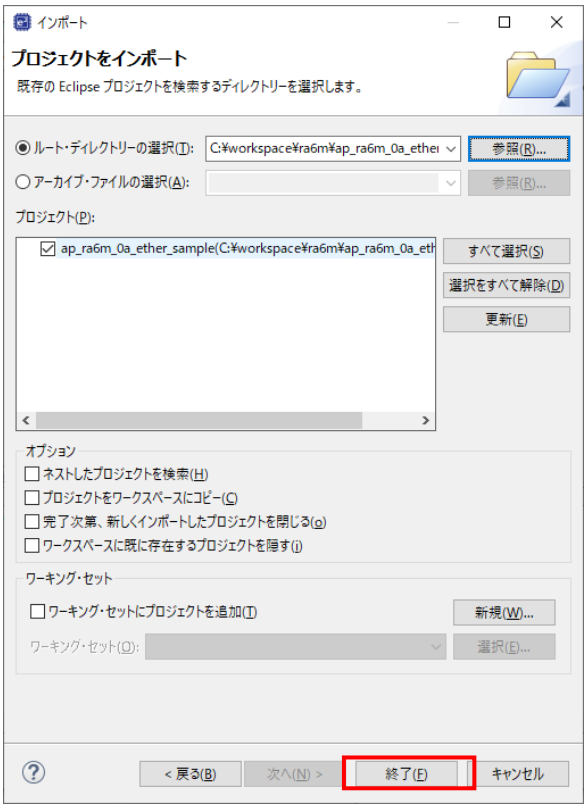
⑦　［既存のプロジェクトをワークスペースへ］を選択し［次へ］を選択します。



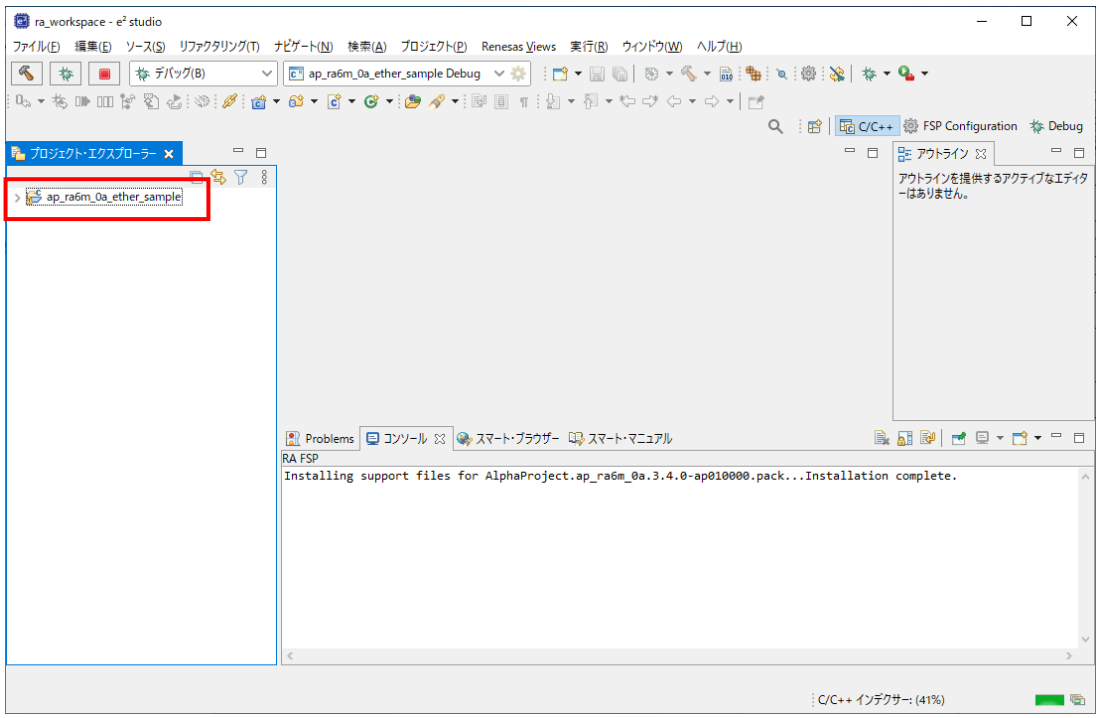
⑧　［ルート・ディレクトリーの選択］を選択し、［参照］からサンプルプログラムのフォルダを選択します。



⑨　[終了]を選択します。



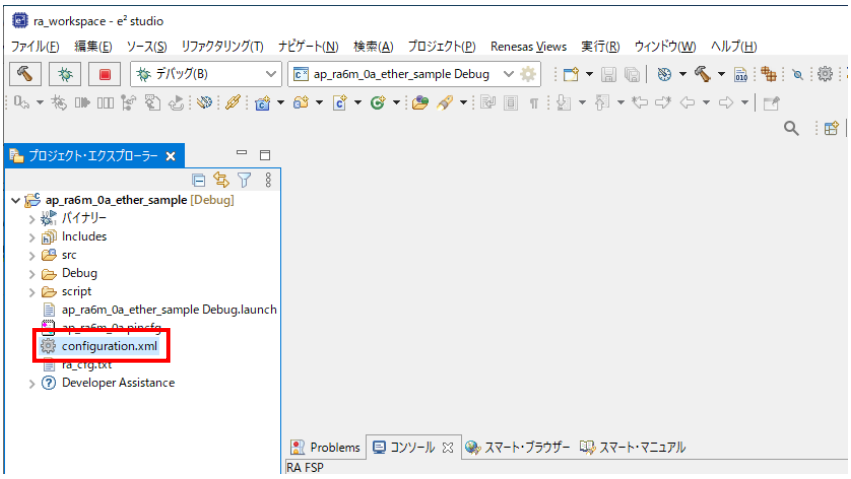
⑩　ナビゲーションウィンドウにサンプルプログラムのプロジェクトが追加されていることを確認します。



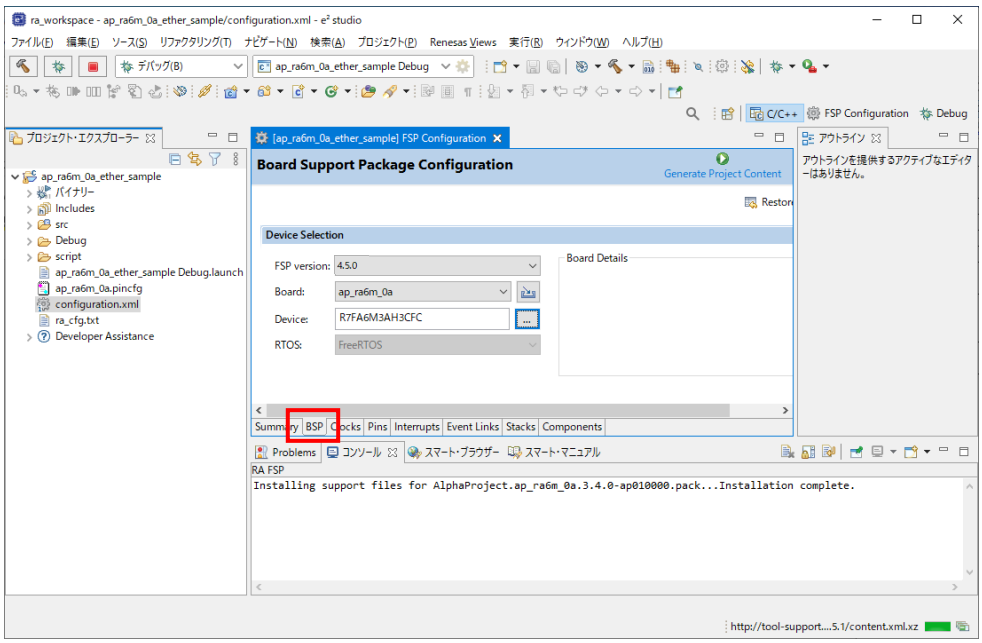
以上でプロジェクトのインポートは完了です。

3.1.2　ビルド方法

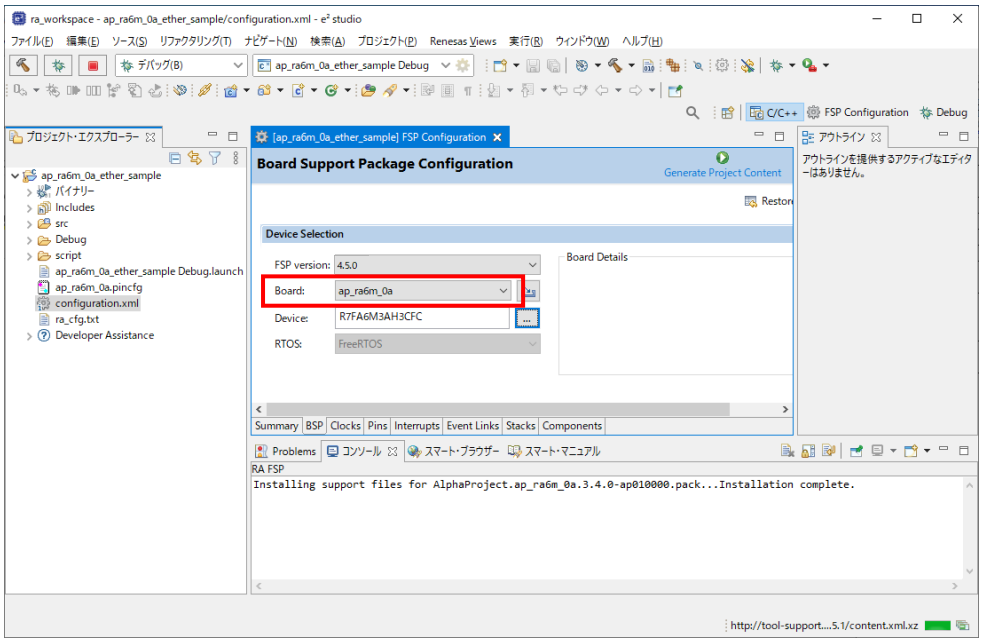
①　プロジェクトのコンフィギュレータファイルを開きます。



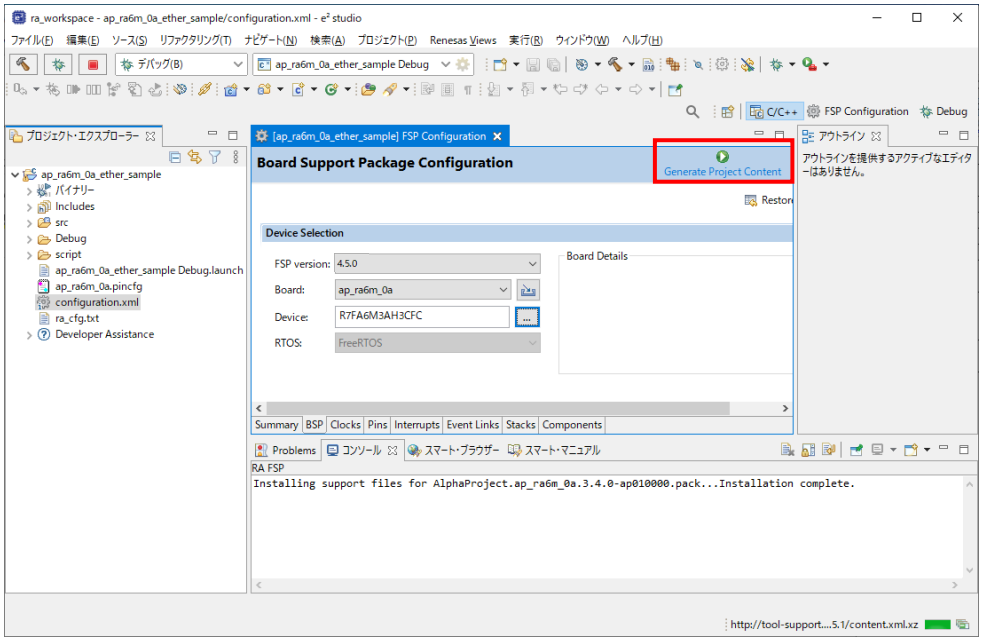
②　[BSP] タブを開きます。



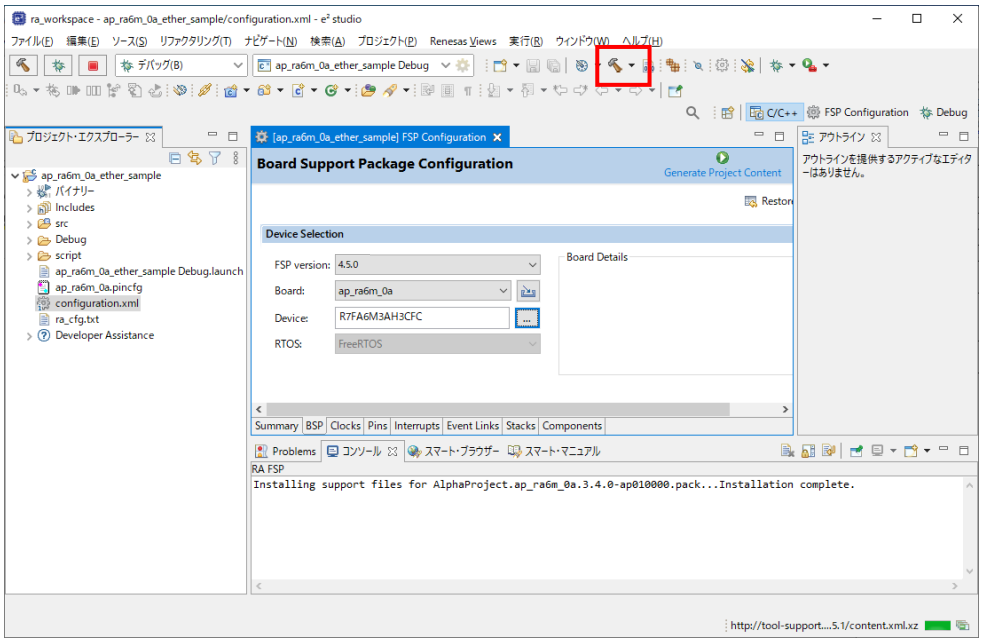
③　[BSP]タブで [Board] が「ap_ra6m_0a」であることを確認します。



④　[Generate Project Content] をクリックし、自動作成ファイルを出力して設定をプロジェクトに適用します。



⑤　ツールバーからビルドアイコンを選択します。
ビルドが成功すると、¥Debug ワークフォルダにオブジェクトファイルが生成されます。



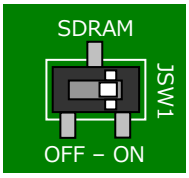
e2 studio の詳細な使用方法に関しては、 e2 studio のマニュアルを参照してください。

3.1.3　デバッグ方法

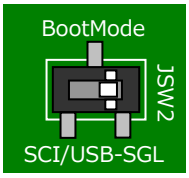
①　「3.1.2　ビルド方法」を参考に、プロジェクトをビルドしてください。
（例で使用する画像では、プロジェクト名は「ap_ra6m_0a_ether_sample」となっています。）

②　ボード上のディップスイッチを設定します。
サンプルプログラム動作時は、BootMode を「SGL（SINGLE）」に設定します。

<AP-RA6M-0A>

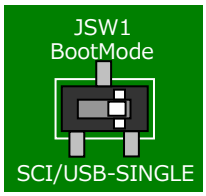


JSW1：ON
ボード上の SDRAM を使用する



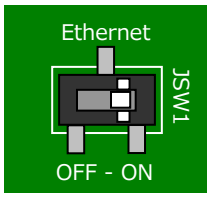
JSW2：SGL
シングルチップモード

<AP-RA6M-1A>



JSW1:SINGLE

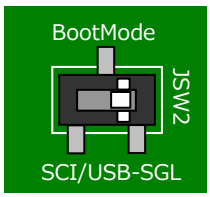
<AP-RA8M-0A、AP-RA8D-0A>



JSW1

Ethernet を使用する場合: ON

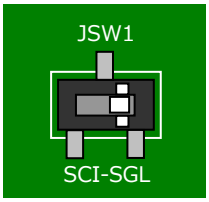
CAMERA を使用する場合: OFF



JSW2：SGL

シングルチップモード

<AP-RA8T-0A>

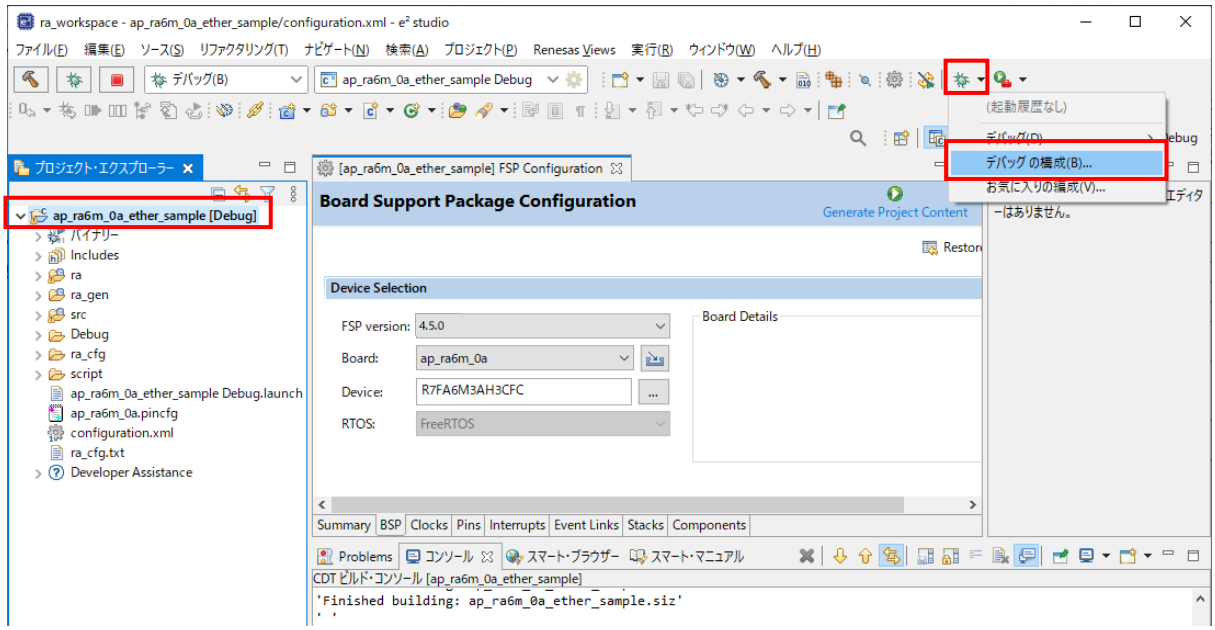


JSW1：SGL

シングルチップモード

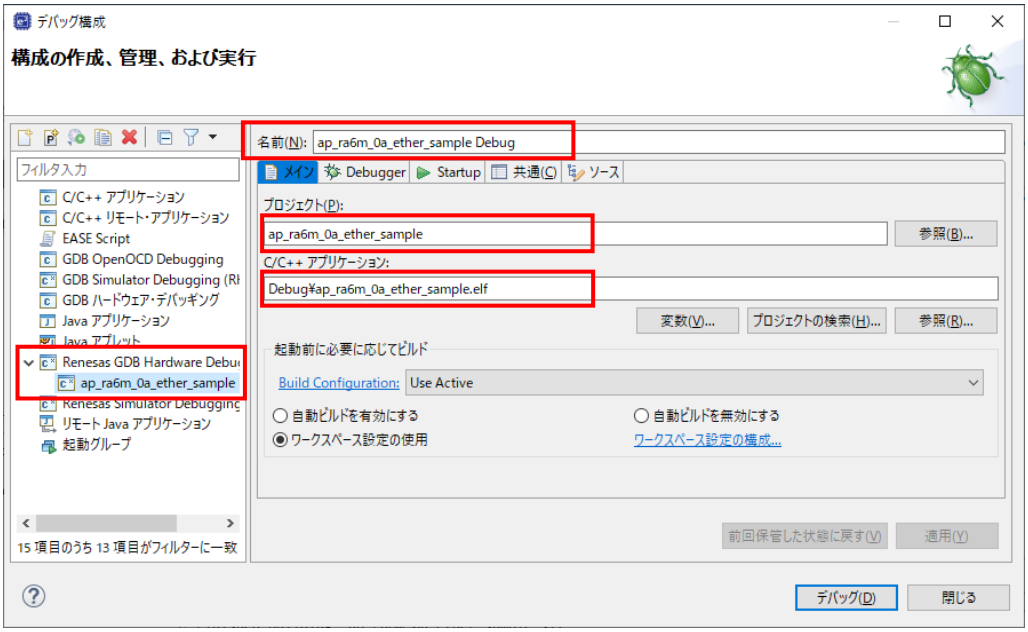
③　ボードに電源を投入してください。

④　プロジェクトを選択し、メニューバーから [デバッグの構成] を開きます。



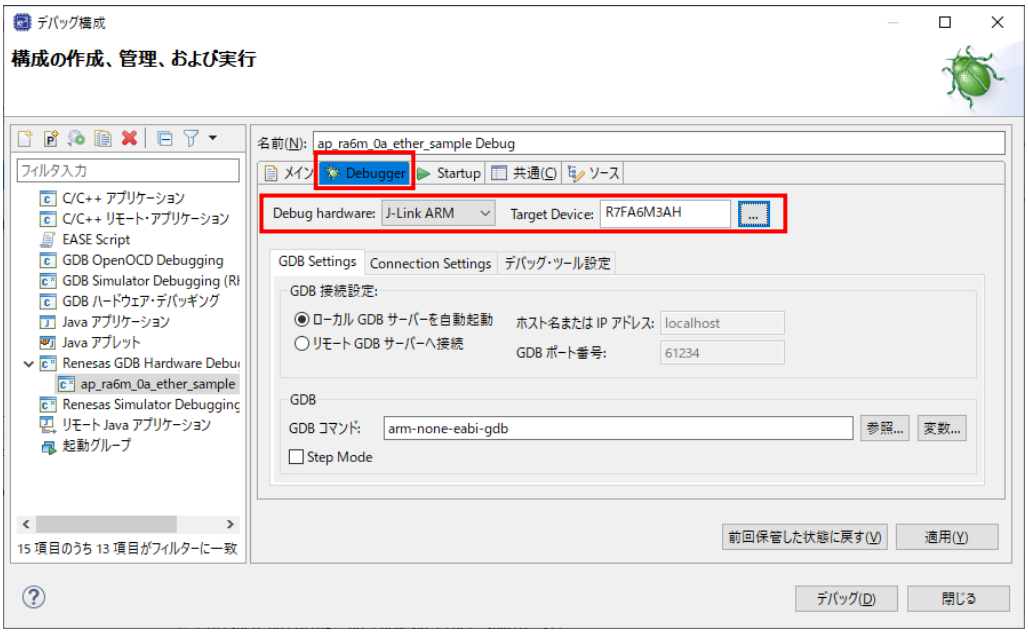
⑤ [Renesas GDB Hardware Debug] の[<プロジェクト名> Debug]を選択し、下記の内容になっていることを確認してください。

- [名前]：<プロジェクト名> Debug
  （AP-RA6M-1A の場合：<プロジェクト名> Debug_Flat）
- [プロジェクト]：<プロジェクト名>
- [C/C++アプリケーション]：Debug¥ <プロジェクト名>.elf

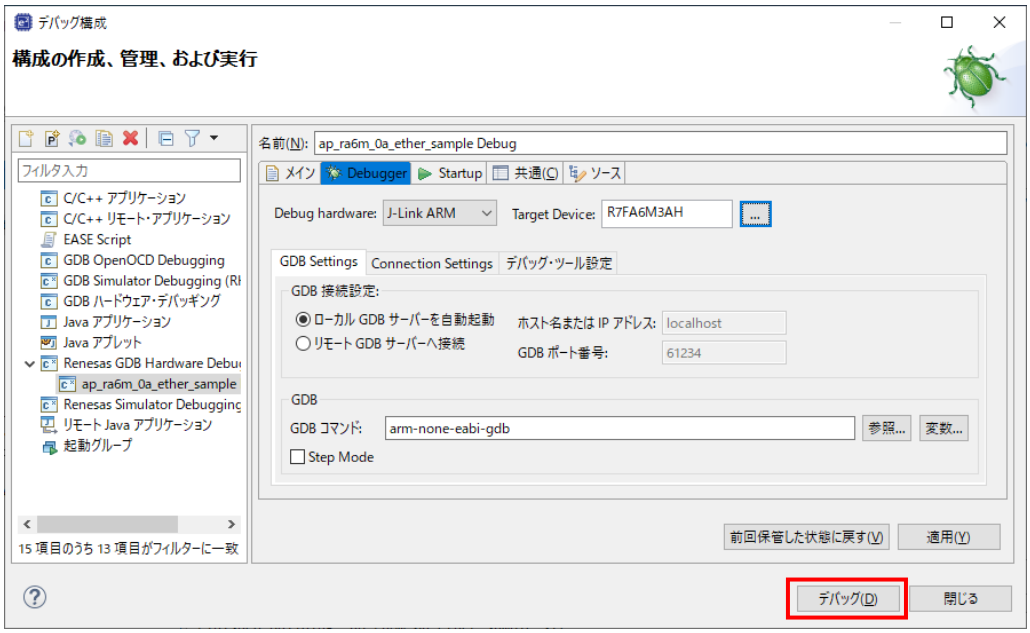


⑥　[Debugger] タブを選択し、 [Debug hardware] が [J-Link ARM] 、 [Target Device] が「R7FA6M3AH」（※）に設定されていることを確認してください。

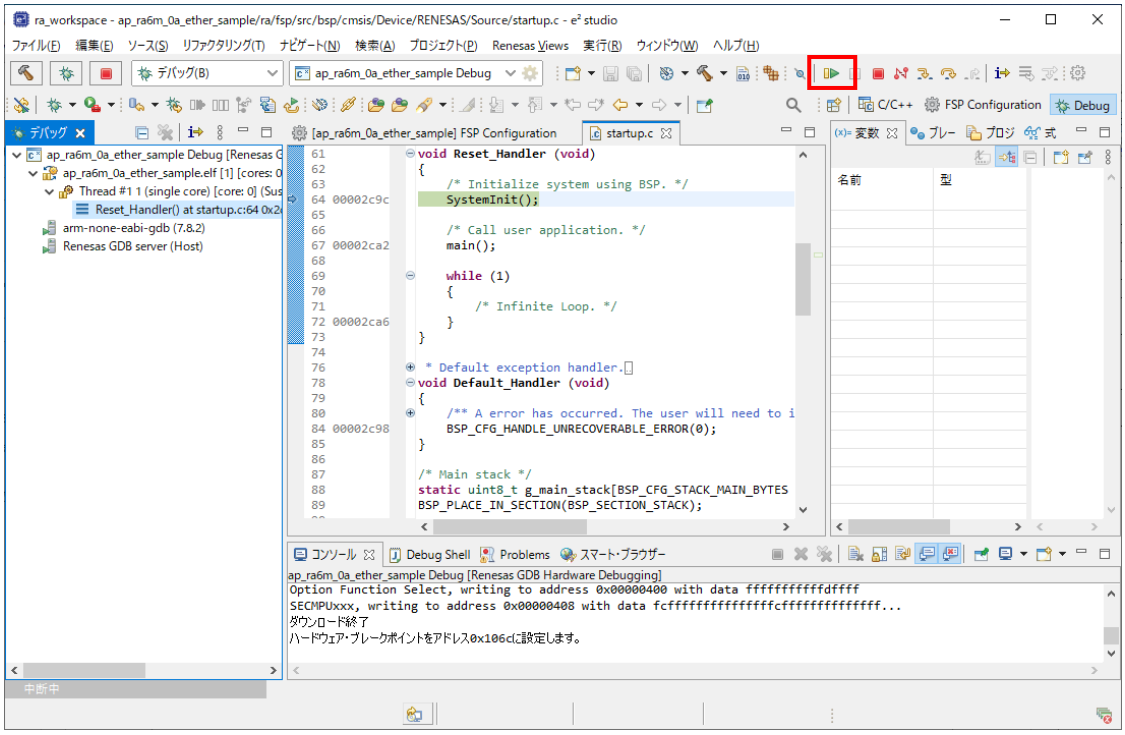
※ AP-RA6M-1A の場合は [CPU] から「R7FA6M5BH」
AP-RA8M-0A の場合は [CPU] から [R7FA8M1AH]
AP-RA8D-0A の場合は [CPU] から [R7FA8D1BH]
AP-RA8T-0A の場合は [CPU] から [R7FA8T1AH] を選択してください。



⑦　[デバッグ] を選択します。



⑧　ボードとの接続が完了したらプログラムを実行し、サンプルプログラムを動作させてください。



⑨　プログラムの動作が確認できましたら、CPU ボードへのプログラムのダウンロードも完了しています。

プログラムの動作については、デバッグしているサンプルプログラムのアプリケーションノートをご参照ください。

4.　e2 studio を用いた新規プロジェクト作成方法

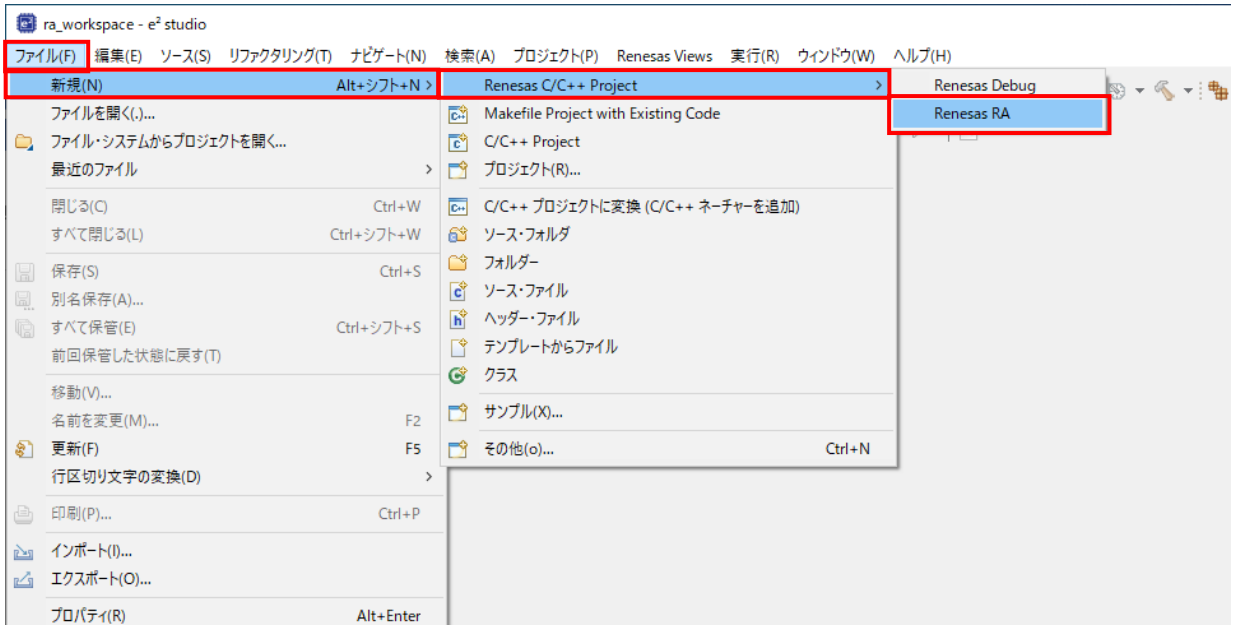
本章では、e2 studio で CPU ボード用プログラムを作成する方法を説明します。
プロジェクトコンフィギュレータの設定を行うプロジェクトとして、「4.1　UART プログラムの作成例」にて、UART を使用したプログラムの作成方法を説明します。
なお、本書で扱っている FSP は バージョン 4.5.0 です。異なるバージョンの FSP をご使用の場合は、本書で解説する項目・設定等が異なる場合がございます。ご注意ください。

また、本章では、図中の選択・実行する必要のある箇所を赤色枠、内容の確認をする箇所を橙色枠で示します。

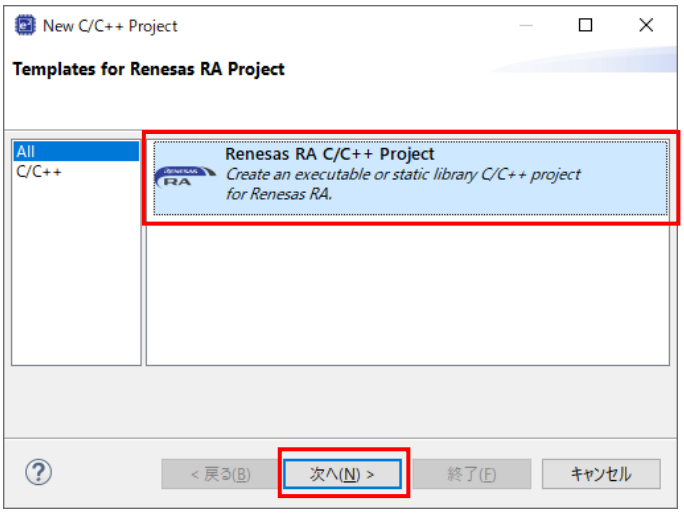
4.1　UART プログラムの作成例

本節では、AP-RA6M-0A で動作する UART のプログラムを例に、プロジェクトの作成方法を説明します。

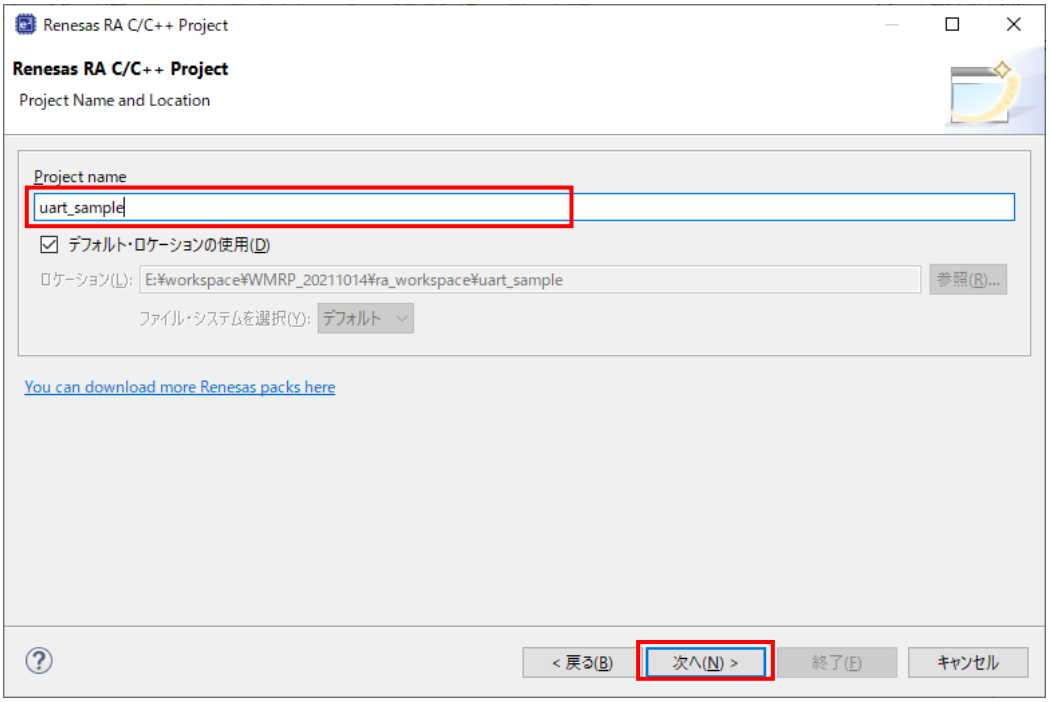
①　e2 studio のメニューから、 [ファイル] – [新規] – [Renesas C/C++ Project] - [Renesas RA] を選択します。



② [Renesas RA C Executable Project] を選択し、 [次へ] を押します。



③ [Project name] に任意のプロジェクト名を入力します（ここでは「uart_sample」とします）。
入力後、 [次へ] を押して次へ進みます。

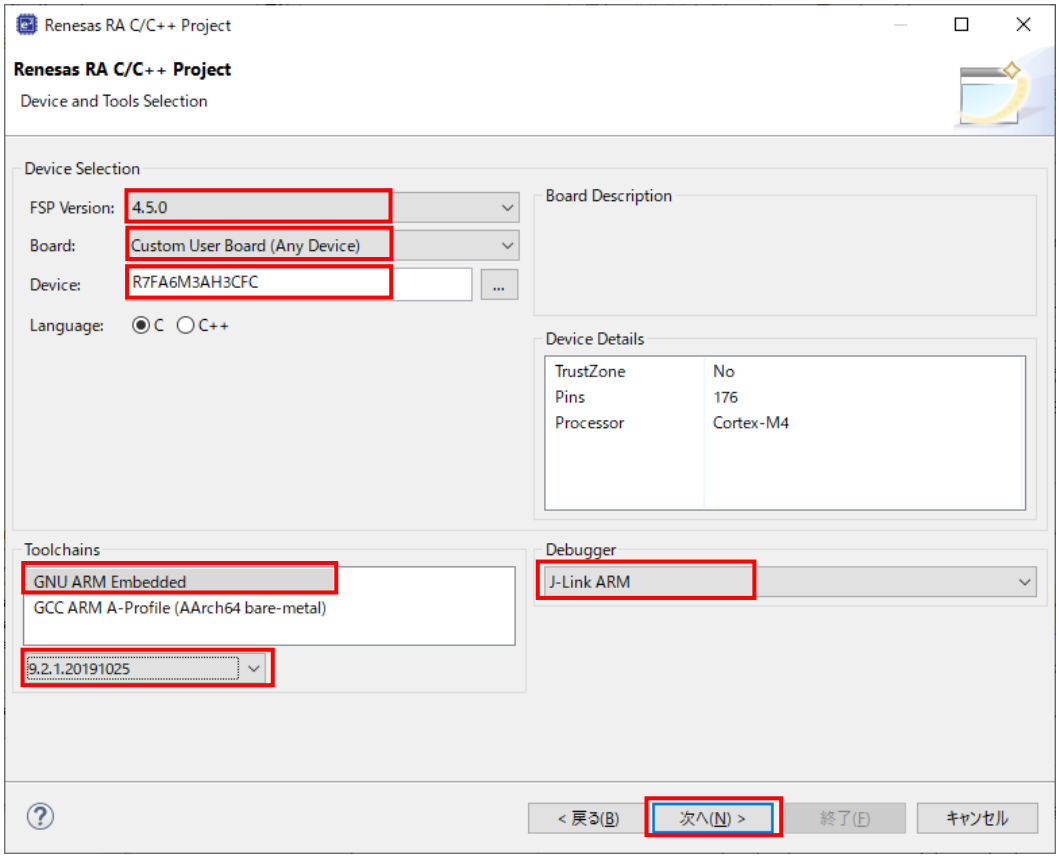


④ AP-RA6M-0A/1A の場合

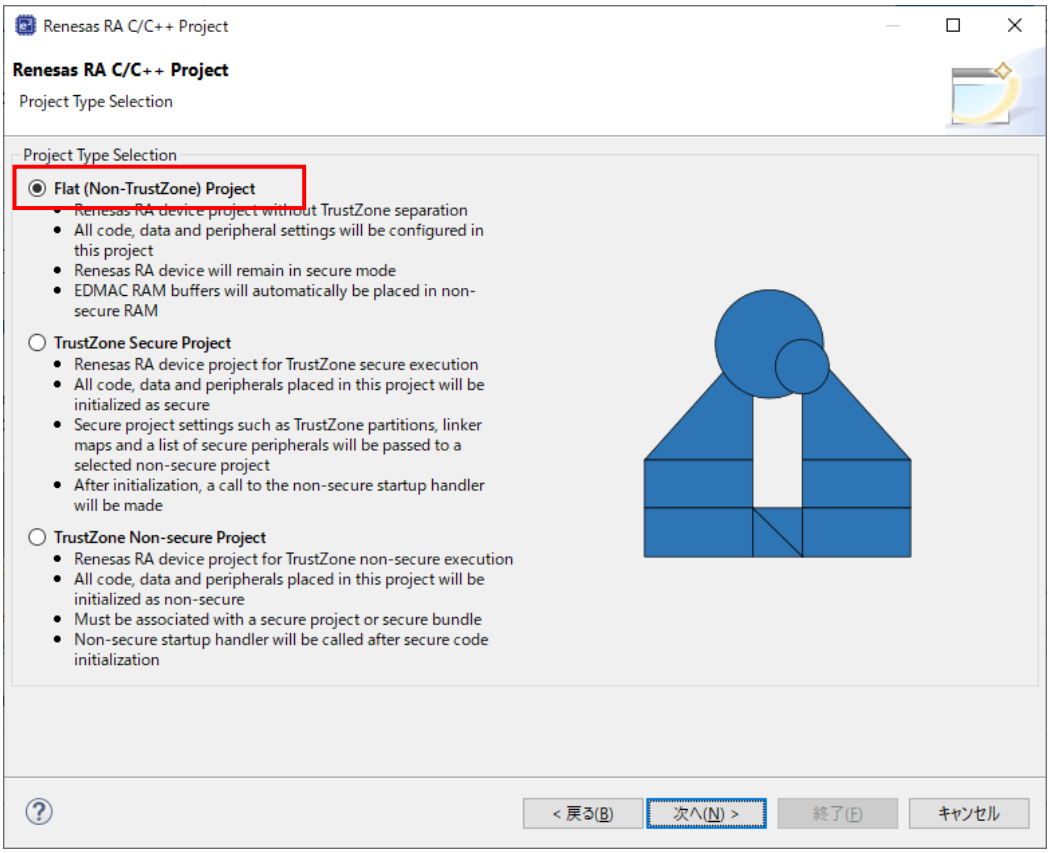
Device Selection 設定では、[FSP version] から[4.5.0]、[Board]から[Custom User Board]、
[CPU] から [R7FA6M3AH3CFC]（※）をそれぞれ設定します。
Toolchain 設定では「GNU ARM Embedded」を選択のうえ、 [Toolchain version] に[10.3.1.20210824]を設定します。
Debugger 設定では、 [Debugger] には [J-Link ARM] を設定します。
設定後、 [次へ] を押します。
※ AP-RA6M-1A の場合は [CPU] から [R7FA6M5BH3CFB] を選択してください。

AP-RA8M-0A、AP-RA8D-0A の場合
Device Selection 設定では、[FSP version] から[5.1.0]、[Board]から[Custom User Board]、
[CPU] から [R7FA8M1AHDCBD]（※）をそれぞれ設定します。
Toolchain 設定では「GNU ARM Embedded」を選択のうえ、 [Toolchain version] に[13.2.1.arm-13-7]を設定します。
Debugger 設定では、 [Debugger] には [J-Link ARM] を設定します。
設定後、 [次へ] を押します。
※ AP-RA8D-0A の場合は [CPU] から [R7FA8D1BHDCBD] を選択してください。

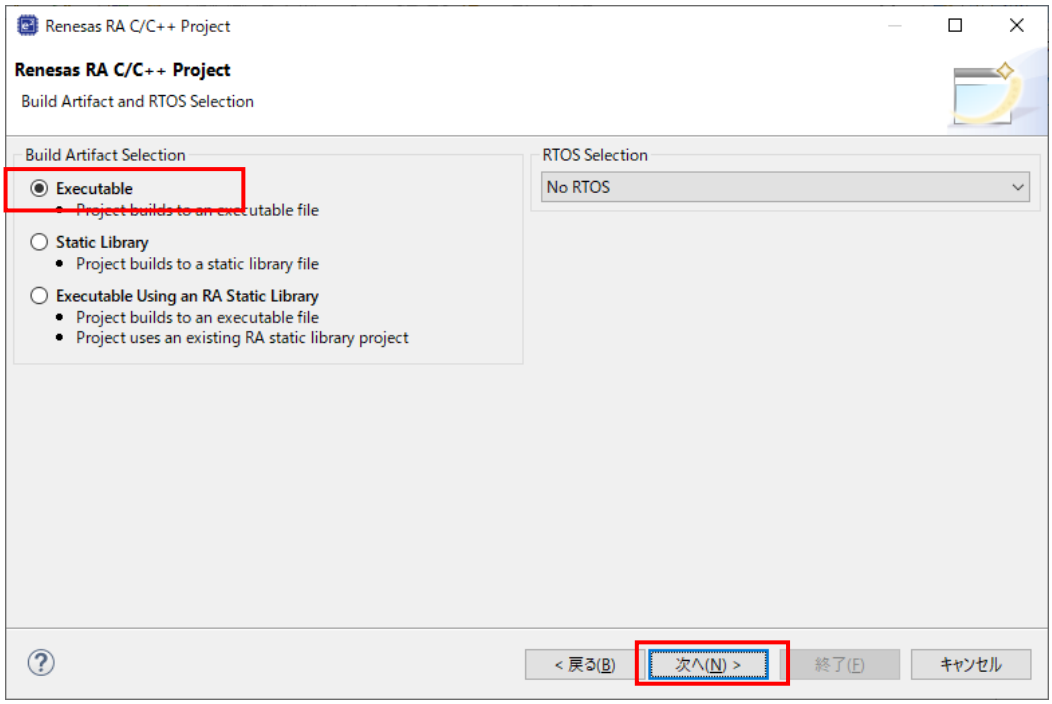
AP-RA8T-0A の場合
Device Selection 設定では、[FSP version] から[5.5.0]、[Board]から[Custom User Board]、
[CPU] から [R7FA8T1AHECFB]を設定します。
Toolchain 設定では「GNU ARM Embedded」を選択のうえ、 [Toolchain version] に[13.2.1.arm-13-7]を設定します。
Debugger 設定では、 [Debugger] には [J-Link ARM] を設定します。
設定後、 [次へ] を押します。



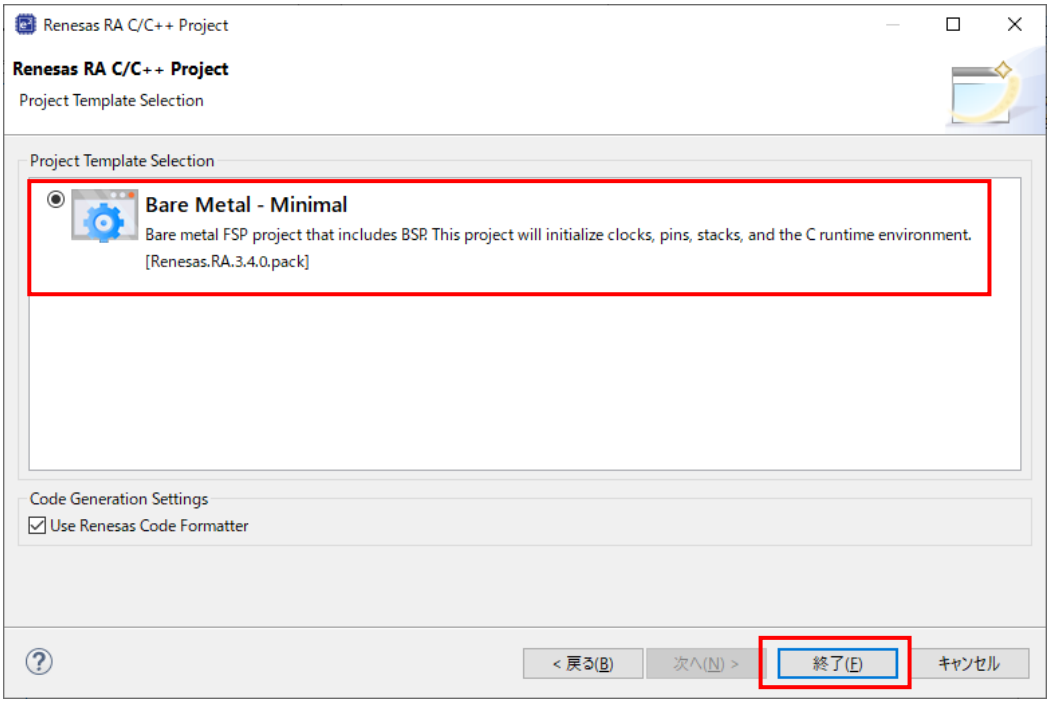
⑤　（AP-RA6M-1A/AP-RA8M-0A/AP-RA8D-0A/AP-RA8T-0A の場合）プロジェクトタイプを設定します。
サンプルでは [Flat (Non-TrustZone) Project] を選択します。
（セキュリティ機能がない CPU ボードの場合は表示されませんので⑥へお進みください）



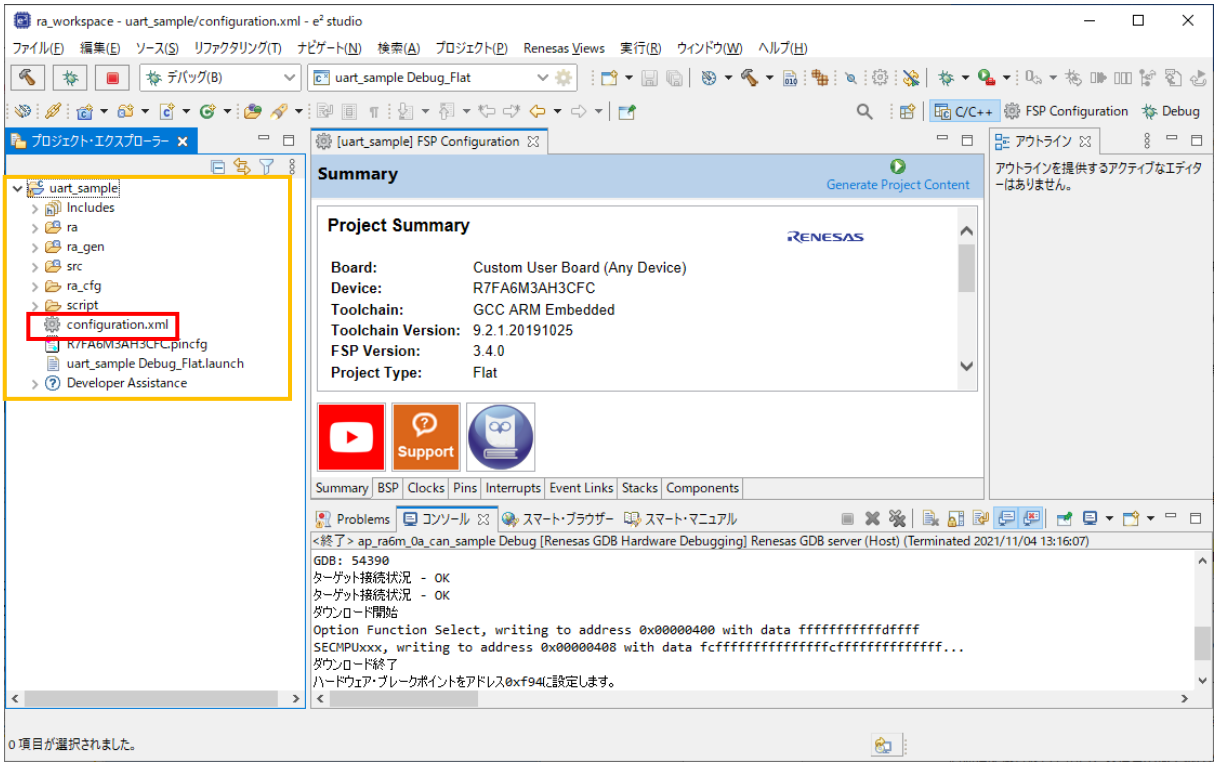
⑥　ビルド生成物と RTOS 設定を設定します。
サンプルでは [Build Artifact Selection] に [Executable] 、[RTOS Selection] に [No RTOS] を選択します。



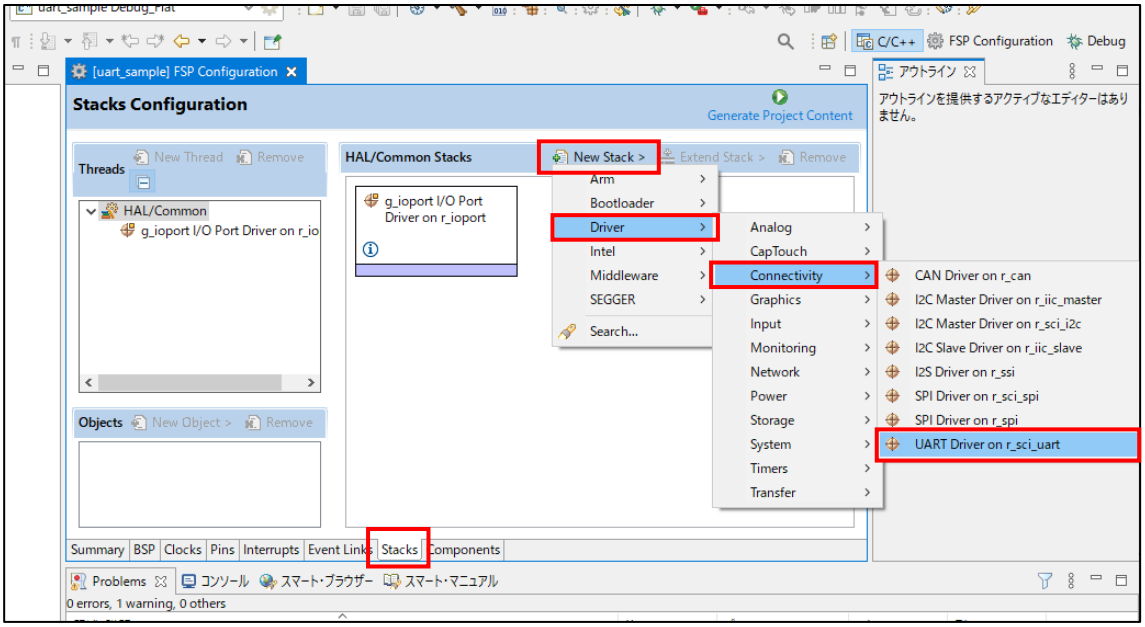
⑦　プロジェクトを作成します。
プロジェクトのテンプレートは [Bare Metal - Minimal] を選択し、 [終了] を押してプロジェクトを作成します。



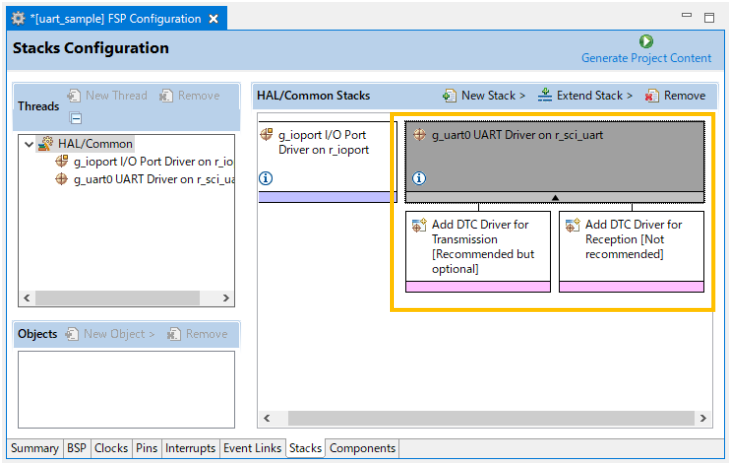
⑧　プロジェクトが作成されました。
以降では、RA プロジェクトコンフィギュレータでプロジェクトの設定を行います。
プロジェクトのコンフィギュレータファイルを開いてください。



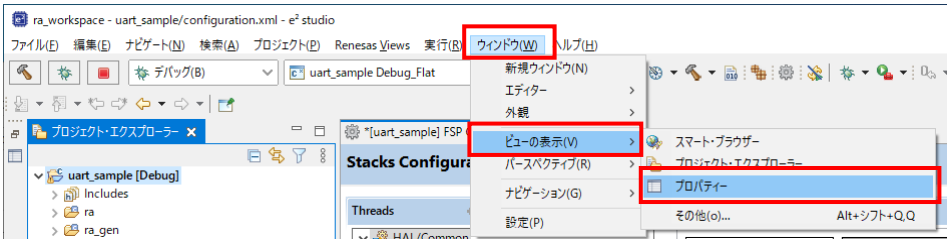
⑨　[Stacks] タブを開きます。この画面で、HAL ドライバやフレームワークの追加や設定をします。
[New Stack] を押して、 [Driver] - [Connectivity] - [UART Driver on r_sci_uart] を選択します。



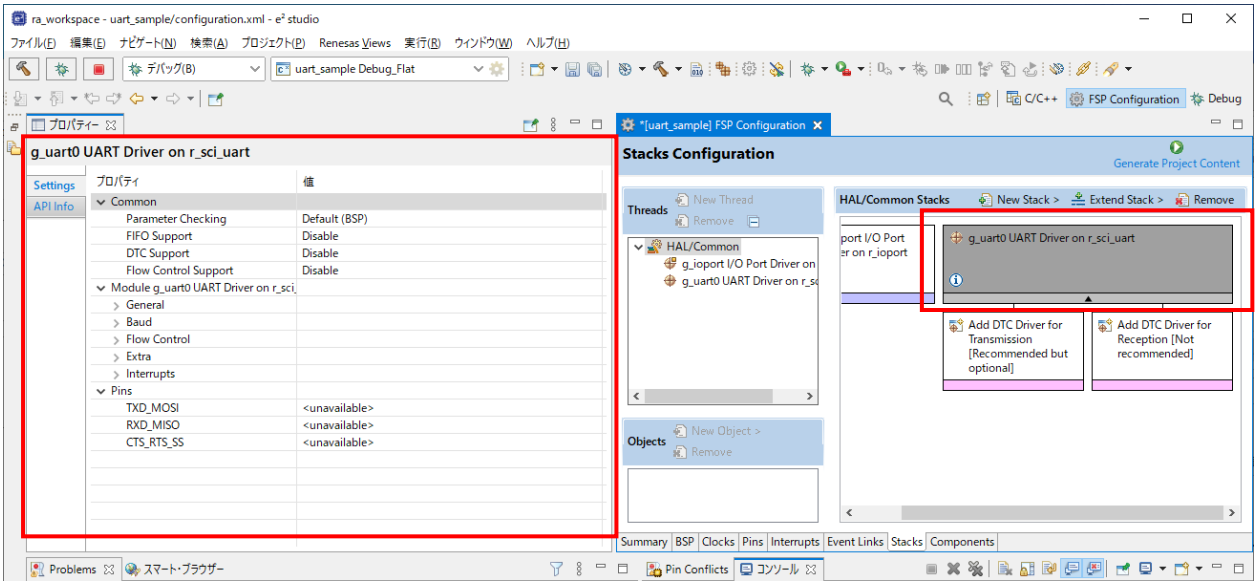
⑩　UART のモジュールが追加され、下図の橙色の枠のように、UART Driver が画面に表示されます。
[g_uart0 UART Driver on r_sci_uart] のモジュールをクリックすると、そのプロパティがプロパティビューに表示されます。



プロパティが見つからない場合は、ツールバーの [ウィンドウ] - [ビューの表示] - [プロパティー] で表示できます。



⑪　[g_uart0 UART Driver on r_sci_uart] のモジュールをクリックすると、そのプロパティがプロパティビューに表示されます。



AP-RA6M-0A/1A の場合

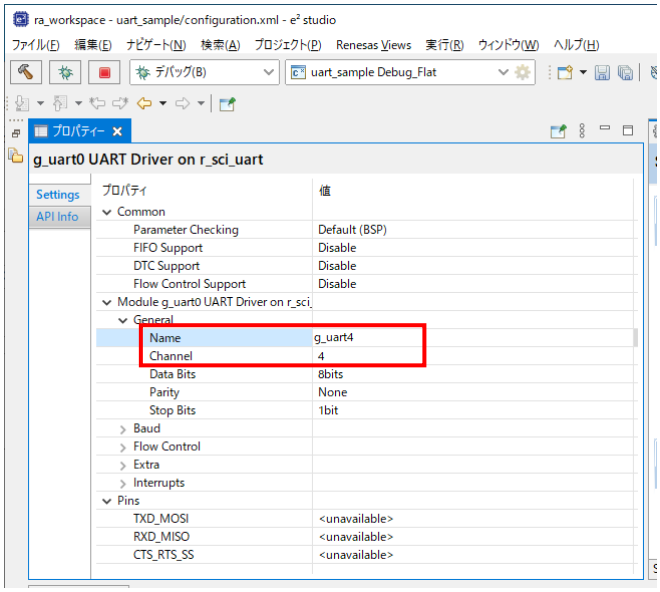
表示されたプロパティのうち、[General]内の 設定[Name]を[g_uart4]、[Channel]を[4]にそれぞれ変更してください。

AP-RA8M-0A、AP-RA8D-0A の場合

表示されたプロパティのうち、[General]内の 設定[Name]を[g_uart1]、[Channel]を[1]にそれぞれ変更してください。

AP-RA8T-0A の場合

表示されたプロパティのうち、[General]内の 設定[Name]を[g_uart0]、[Channel]を[0]にそれぞれ変更してください。



⑫　[Pins] タブを開きます。この画面で、UART 端子の設定をします。

AP-RA6M-0A/1A の場合

[Peripherals]- [Connectivity:SCI]-[SCI4]を選択した後、

[Pin Group Selection]から[_A only]、[Operation Mode]から[Asynchronous UART]を選択します。

AP-RA8M-0A、AP-RA8D-0A の場合

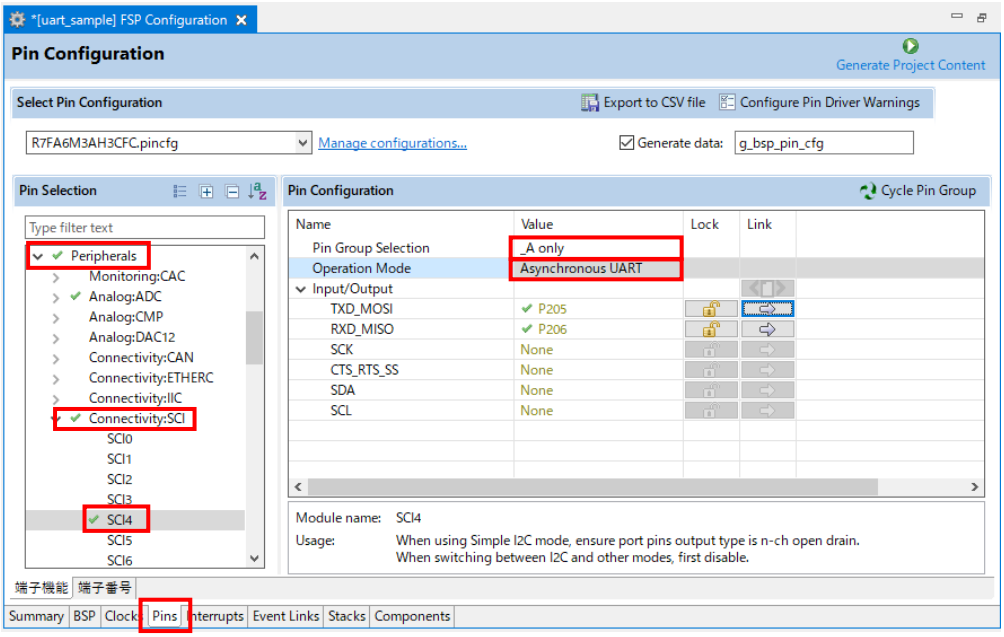
[Peripherals]- [Connectivity:SCI]-[SCI1]を選択した後、

[Pin Group Selection]から[_B only]、[Operation Mode]から[Asynchronous UART]を選択します。

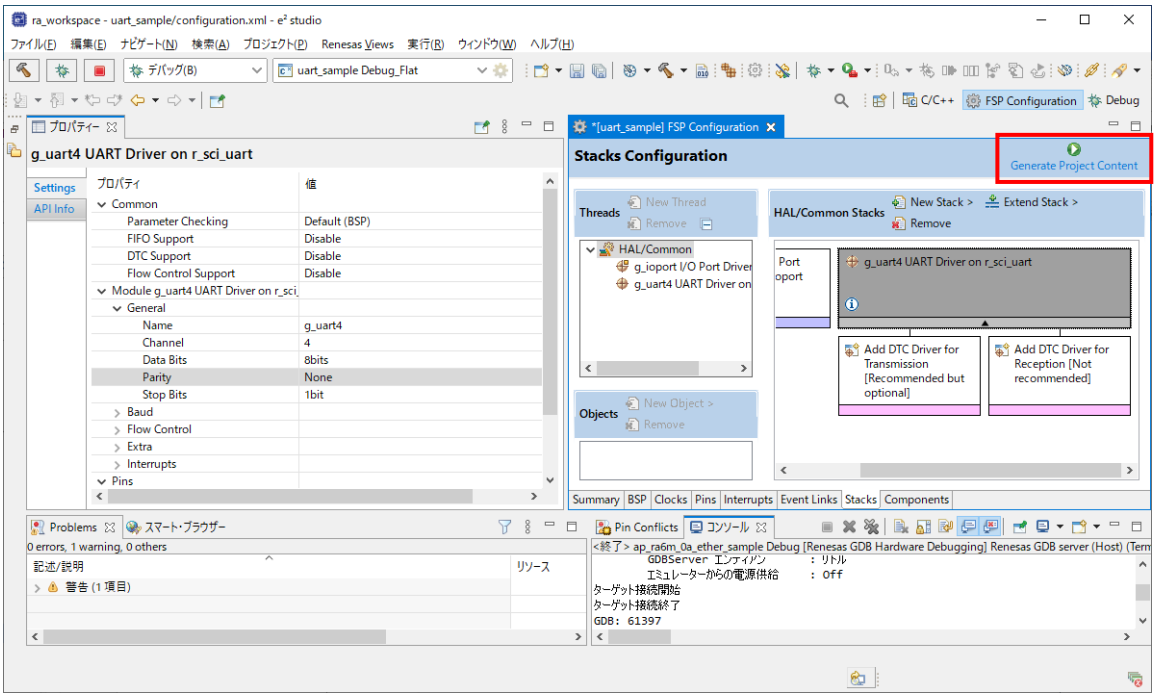
AP-RA8T-0A の場合

[Peripherals]- [Connectivity:SCI]-[SCI0]を選択した後、

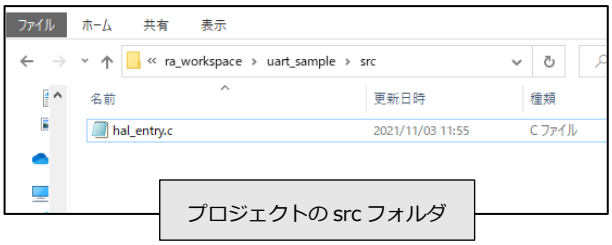
[Pin Group Selection]から[_C only]、[Operation Mode]から[Asynchronous UART]を選択します。



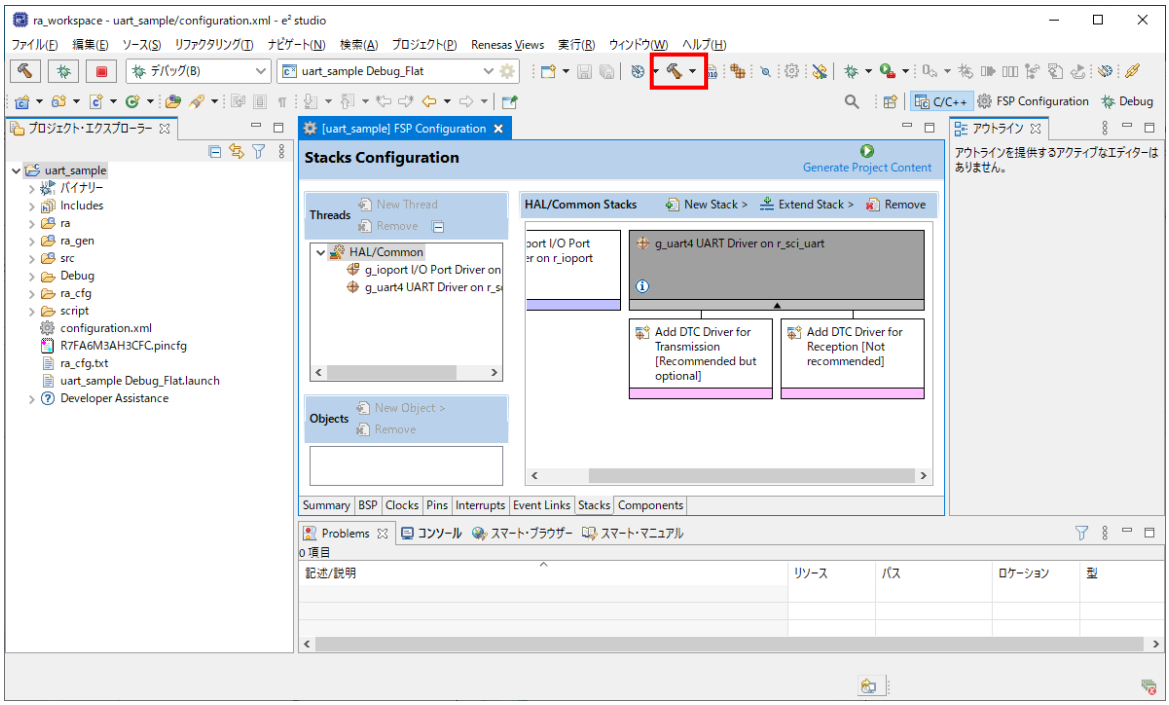
⑬　今回は特に設定する必要はありませんが、続けて他のモジュールを設定することもあります。
適宜必要な設定を全て行った後に、[Generate Project Content] をクリックしてください。
自動作成ファイルが出力されて設定がプロジェクトに適用されます。



⑭　プロジェクトの「src」フォルダに、アプリケーション部分を作成します。



⑮　以上で UART のプロジェクトが完成しました。ビルド後、動作を確認してください。



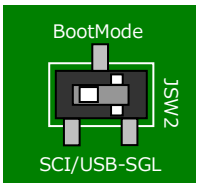
5.　Renesas Flash Programmer を使用した書き込み

本章では、CPU ボードに Renesas Flash Programmer を使用してプログラムを書き込む方法を説明します。

5.1　USB インタフェースを使用した書き込み方法

①　CPU ボードの設定を SCI/USB ブートモードに変更し、CPU ボードと PC を USB ケーブルで接続します。

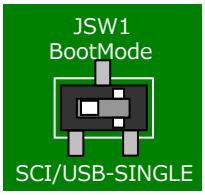
<AP-RM6A-0A>



AP-RA6M-0A の JSW1 の設定は　不問　です。

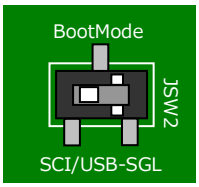
JSW2 : SCI/USB

<AP-RM6A-1A>



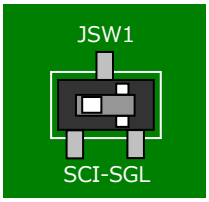
JSW1:SCI/USB

<AP-RM8A-0A, AP-RA8D-0A>



JSW2 : SCI/USB

<AP-RA8T-0A>

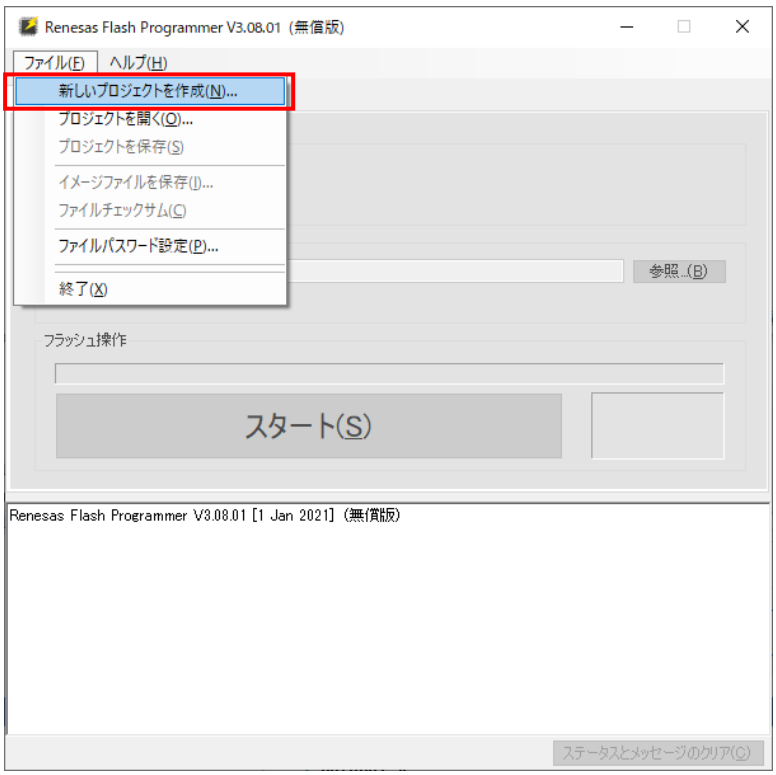


JSW1 : SCI

② CPU ボードの電源を入れます。

③ Renesas Flash Programmer を起動します。

④ RFP を起動すると、以下のようなウィンドウが表示されますので、「ファイル(F)」メニューの「新しいプロジェクトを作成(N)...」を選択します。

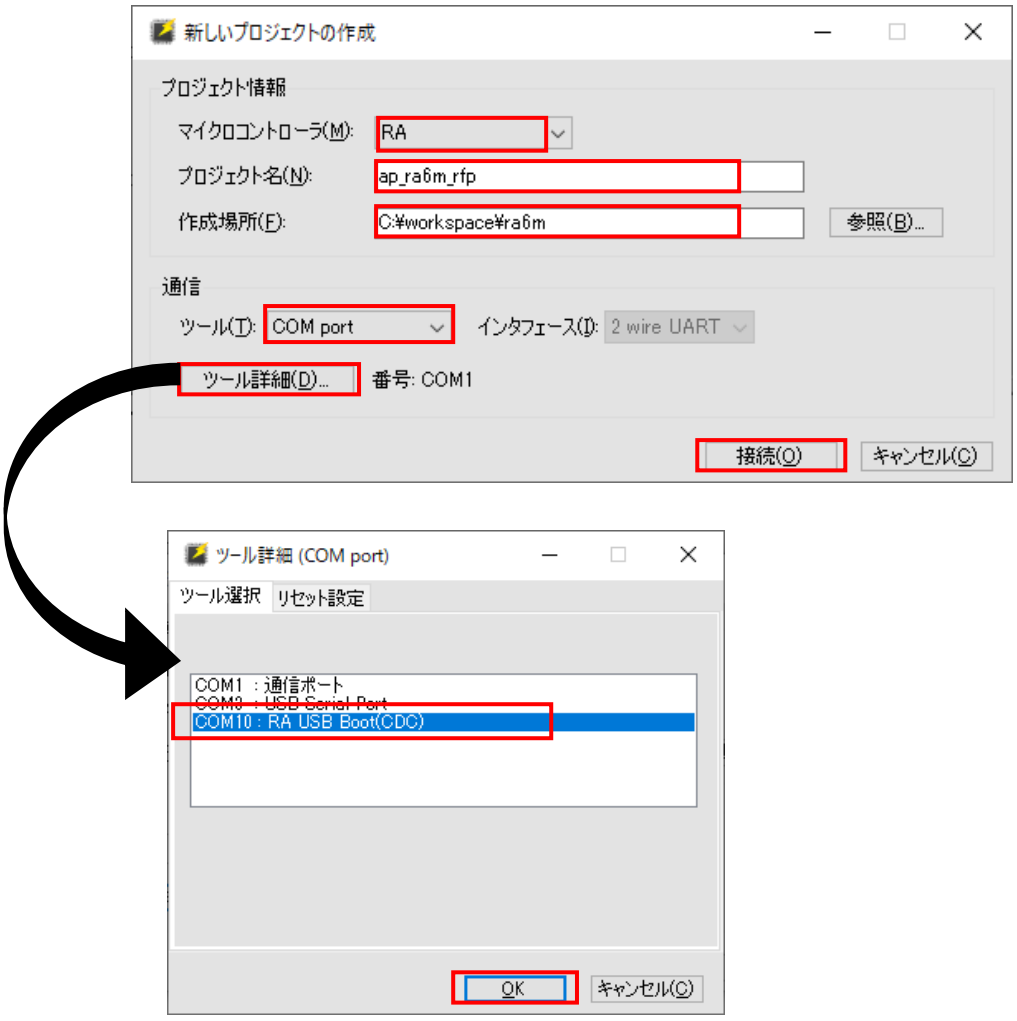


⑤　新しいプロジェクトの作成ウィンドウが表示されますので、プロジェクトの設定を行います。
設定後、 [接続(O)] を押して接続を開始してください。

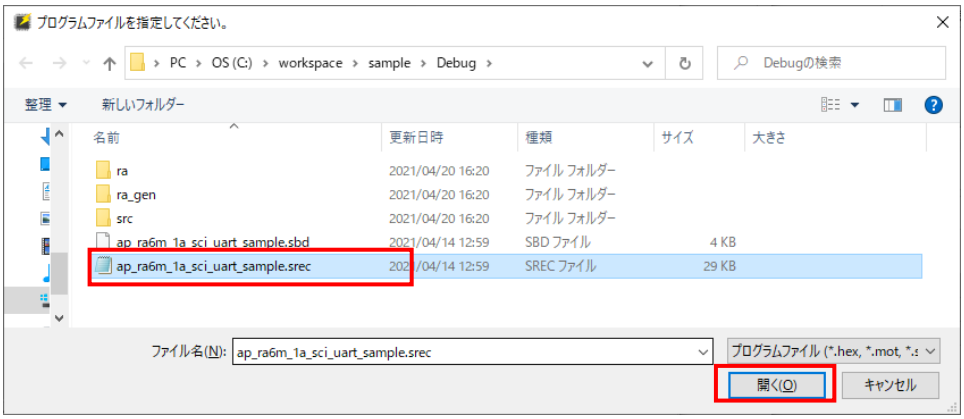
<プロジェクト情報>
・ マイクロコントローラ(M)　：　[RA] を選択します。
・ プロジェクト名(N)　：　任意のプロジェクト名を設定します。
（ここでは例として、「ap_ra6m_rfp」を入力しています。）
・ 作成場所(F)　：　[参照(B)]ボタンを押し、任意の保存先を設定します。
（ここでは例として「C:¥workspace¥ra6m」を選択しています。）

<通信>
・ ツール(T)　：　[COM port] を選択します。
ツール詳細(D)...　[ツール詳細(D)] ボタンを押して設定ウィンドウを開きます。
COM の一覧から [RA USB Boot(CDC)] を選択し、 [OK] を押します。
（COM ポートの番号は環境により異なります。）

COM port が動作しない場合、USB ブート用ドライバが誤認識されている可能性があります。
詳細は RFP のユーザーズマニュアルをご覧ください。

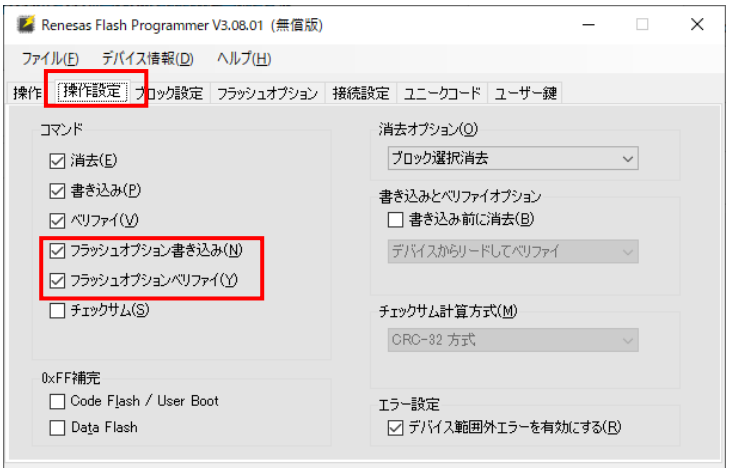


⑥　RFP のメイン画面が表示されます。
プログラムファイルの[参照]ボタンをクリックし、書き込むファイルを選択します。
FlashROM に書き込むファイルは、実行したいビルドモードのフォルダ（Debug）内の**.srec ファイル**を指定します。
（図は、AP-RA6M-1A 用 UART サンプルプログラムの.srec ファイルです）



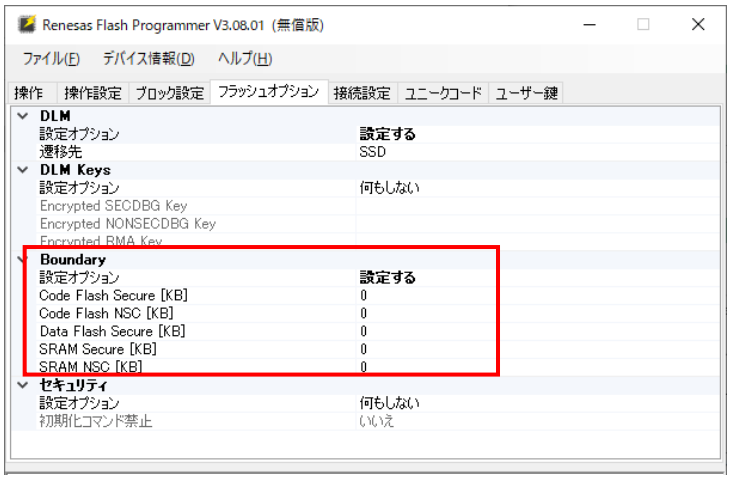
⑦　ファイルの指定後、一度 CPU ボードの電源を切断し、再投入します。
この後は CPU のセキュリティ機能に関する設定に続きます。
セキュリティ機能がない CPU ボードの場合は、⑧～⑩のステップは不要ですので⑪にお進みください。

⑧　[操作設定] タブを開いて、[コマンド] 内の [フラッシュオプション書込み] と [フラッシュオプションベリファイ] のそれぞれにチェックマークを付けて有効化します。

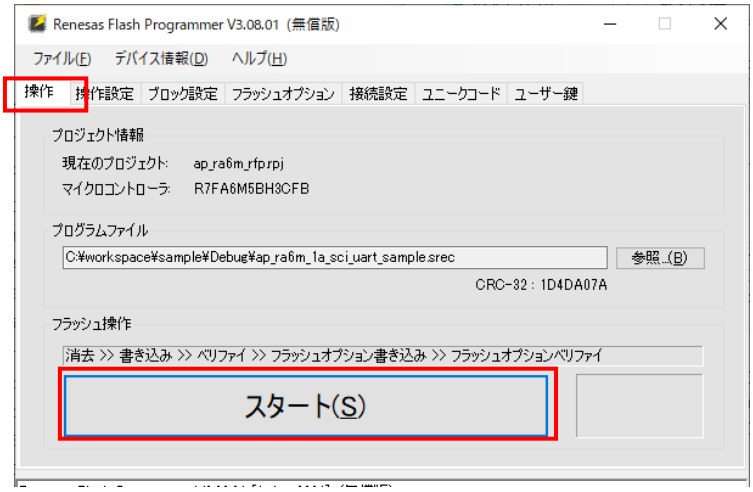


⑨ [フラッシュオプション] タブを開いて、設定「DLM」の [設定オプション] を [設定する]、 [遷移先] を開発で指定する値に設定します。
（ここでは例として、[遷移先] を [SSD] に設定しています）

⑩ [フラッシュオプション] タブ内の設定「Boundary」の [設定オプション] を [設定する] に設定し、さらに、ユーザが作成したプログラムに従ったセキュア領域をそれぞれ設定します。
（ここではメモリの全領域を非セキュアとする例として、[Code Flash Secure [KB]] ～ [SRAM NSC [KB]]まですべてを [0] に設定しています）

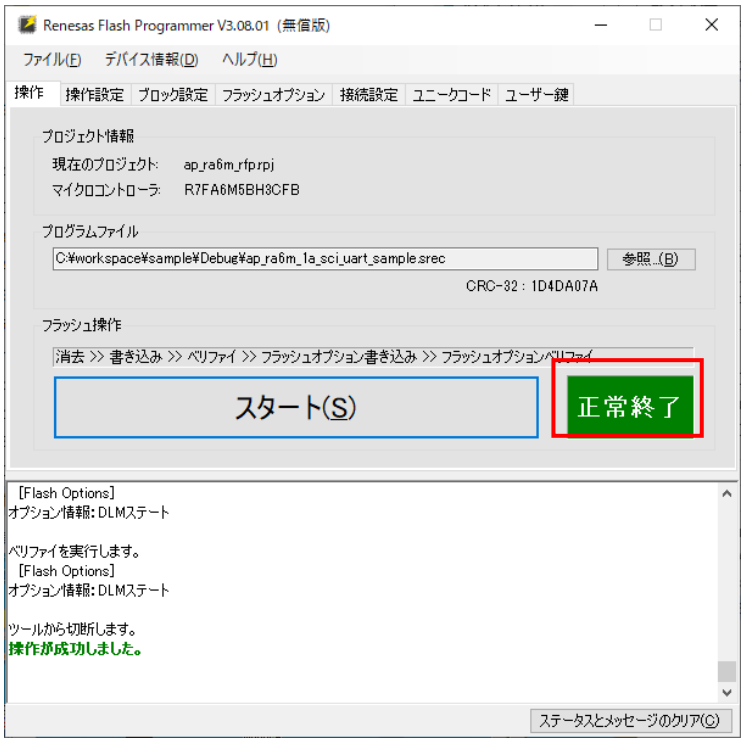


⑪　[操作]タブを開いて [スタート] ボタンを押して書き込みを開始します。



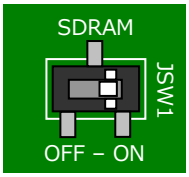
⑫　プログレスバーが表示され、書き込みの進捗に合わせて変化します。

⑬　プログレスバーの表示が消え、メイン画面に [正常終了] と表示されれば、正常に書き込みが終了しています。
CPU ボードの電源を切り、その後 RFP を終了します。
以上で FlashROM への書き込みは終了です。

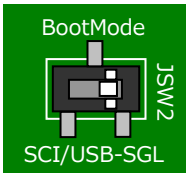


⑭ CPU ボードをサンプルプログラム動作時のボード設定に変更し、再度電源を投入してサンプルプログラムが動作することを確認します

<AP-RA6M-0A>

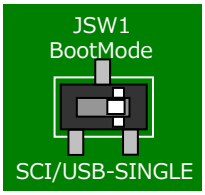


JSW1：ON
ボード上の SDRAM を使用する



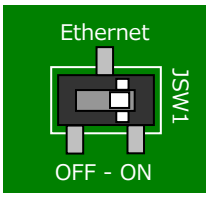
JSW2：SGL
シングルチップモード

<AP-RA6M-1A>



JSW1:SINGLE

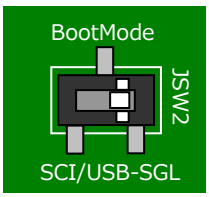
<AP-RA8M-0A, AP-RA8D-0A>



JSW1

Ethernet を使用する場合: ON

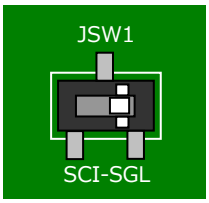
CAMERA を使用する場合: OFF



JSW2：SGL

シングルチップモード

<AP-RA8T-0A>



JSW1：SGL

シングルチップモード

再度同じ設定でファイルを書き込む際は、同じプロジェクトを開くことで⑧から作業を開始できます。
その他の機能など RFP の詳細につきましては、RFP のユーザーズマニュアルを参照してください。

注意：プロジェクトは、設定時に接続していた CPU に対応しています。異なる CPU の書き込みには使用できません。
例えば、AP-RA6M-1A と接続して作成した RFP のプロジェクトを使用して他の RA ファミリが実装されたボードに書き込むことはできませんので、使用する CPU に合わせたプロジェクトを新規作成してください。

改定履歴

版数	更新日	改定内容
1 版	2020/02/21	新規作成
2 版	2020/07/22	FSP のバージョン 1.2.0 に対応
3 版	2021/11/10	FSP のバージョン 3.4.0 に対応 AP-RA6M-1A に関する記述追加 「3.1 サンプルプログラムの動作方法」の pack ファイルインポート方法を更新
4 版	2023/10/02	住所の更新
5 版	2024/02/07	AP-RA8M-0A, AP-RA8D-0A に関する記述追加
5.1 版	2024/03/01	AP-RA6M-0A/1A　FSP のバージョン 4.5.0 に対応
6 版	2024/12/02	AP-RA8T-0A に関する記述追加

ご注意

・本文書の著作権は株式会社アルファプロジェクトが保有します。
・本文書の内容を無断で転載することは一切禁止します。
・本文書に記載されているサンプルプログラムの著作権は株式会社アルファプロジェクトが保有します。
・本サンプルプログラムで使用されているミドルウェアおよびドライバの著作権はルネサス エレクトロニクス株式会社が保有します。
・本文書に記載されている内容およびサンプルプログラムについてのサポートは一切受け付けておりません。
・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承ください。
・本文書の内容については、万全を期して作成いたしましたが、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡ください。
・本文書の内容は、将来予告なしに変更されることがあります。

商標について

・RA ファミリおよび RA6M3, RA6M5, RA8M1, RA8D1, RA8T1 は、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
・e2 studio は、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
・Renesas Flash Programmer は、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
・Flexible Software Package は、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。

・その他の会社名、製品名は、各社の登録商標または商標です。

ALPHA PROJECT
株式会社アルファプロジェクト
〒431-3114
静岡県浜松市中央区積志町８３４
https://www.apnet.co.jp
E-MAIL：query@apnet.co.jp