RZ/T2M CPU BOARD
開発チュートリアル

1.3 版　2024 年 09 月 02 日

1.　概要 2

1.1　概要 2

1.2　使用環境 2

1.3　CPU ボード スイッチ設定 3

1.4　各 CPU ボードの資料に関して 3

2.　機能 4

2.1　サンプルプログラムの使用機能 4

3.　e2 studio を用いた動作方法 5

3.1　サンプルプログラムの動作方法 5

3.1.1　インポート方法 5

3.1.2　ビルド方法 11

3.1.3　RAM デバッグ方法 15

3.1.4　ROM デバッグ方法 19

4.　e2 studio を用いた新規プロジェクト作成方法 26

4.1　UART プログラムの作成例 26

5.　シリアル FlashROM への書き込み 38

5.1　事前準備 39

5.1.1　Python のインストール 39

5.1.2　書き込み時の PC⇔ボード間通信プログラムの用意 40

5.1.3　書き込むプログラムの用意 41

5.2　シリアル FlashROM への書き込み 44

5.2.1　ボード設定 44

5.2.2　書き込み手順 46

5.3　シリアル FlashROM ブートの実行 51

©2023 Alpha Project Co., Ltd.

1.　概要

1.1　概要

本アプリケーションノートは、弊社製 RZ/T2M ファミリ CPU ボードを用いて、Flexible Support Package for Renesas RZ/T シリーズ（以下、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。
各サンプルプログラムで使用されている環境のバージョンが異なる場合に画面の構成など多少違いがございますが、基本的な使用方法は変わりません。
本アプリケーションノートは、AP-RZT2-0A を基準に記載されておりますが、AP-RZT2-1A の場合でも同様にご参照いただけます。
（AP-RZT2-0A の設定のみ記述されている項目につきましては、ボード名、CPU 名をお使いのボードに合わせてください。）

ソフトウェア	バージョン	備考
e2studio	2024-04	–
GCC for Renesas RZ	12.2.1.arm-12-24	–
FSP	2.1.0	Flexible Support Package for Renesas RZ/T シリーズ

デバッガ	ハードウェアバージョン	備考
J-Link	V11	Segger Microcontroller Systems 社 ハードウェアバージョン V10 以下はご使用になれませんのでご注意ください。

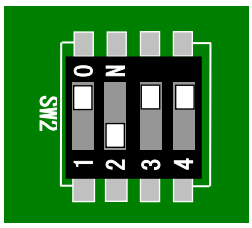
※AP-RZT2-0A と J-Link を直接接続することはできません。
AP-RZT2-0A 側(ハーフピッチコネクタ)と J-Link 側(フルピッチコネクタ)を接続するための変換アダプタが必要となります。
変換アダプタについては、J-Link 取扱店へご確認ください。

©2023 Alpha Project Co., Ltd.

1.3　CPU ボード スイッチ設定

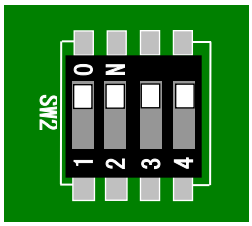
CPU ボードを動作させる際は、動作モードに応じてボード上のディップスイッチを設定する必要があります。
AP-RZT2-0A・AP-RZT2-1A で動作モードに対するスイッチ設定は共通です。

・デバッグをする場合



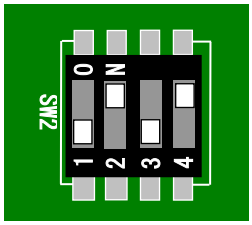
<SW2 設定>	
ブートモード	：シリアル FlashROM ブート/USB ブート/SCI ブートのいずれにも該当しないモード
JTAG Hash モード	：使用する（不問）

・シリアル FlashROM ブートをする場合



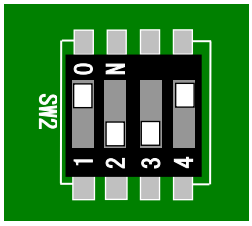
<SW2 設定>	
ブートモード	：シリアル FlashROM ブート
JTAG Hash モード	：使用する（不問）

・SCI ブートをする場合



<SW2 設定>	
ブートモード	：SCI ブート
JTAG Hash モード	：使用する（不問）

・USB ブートをする場合



<SW2 設定>	
ブートモード	：USB ブート
JTAG Hash モード	：使用する（不問）

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。

AP-RZT2-0A　製品ページ　https://www.apnet.co.jp/product/rza/ap-rzt2-0a.html
AP-RZT2-1A　製品ページ　https://www.apnet.co.jp/product/rza/ap-rzt2-1a.html

©2023 Alpha Project Co., Ltd.

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。

・AP-RZT2-0A

ドキュメント番号	サンプルプログラム	機能
AN1648	標準サンプルプログラム	・UART 通信 ・CAN 通信 ・USB ホスト ファイル読み書き ・USB ファンクション 仮想 COM 通信 ・ネットワーク通信
AN1649	PMOD サンプルプログラム	・PMOD 動作 （GPIO/UART/SPI/I2C）
AN1650	EtherCAT サンプルプログラム	・EtherCAT スレーブ通信
AN1651	DualCore サンプルプログラム	・DualCore 動作
AN1652	SSBL サンプルプログラム	・ブート（SSBL）動作

・AP-RZT2-1A

ドキュメント番号	サンプルプログラム	機能
AN1648	標準サンプルプログラム	・UART 通信 ・CAN 通信 ・USB ホスト ファイル読み書き ・USB ファンクション 仮想 COM 通信
AN1649	PMOD サンプルプログラム	・PMOD 動作 （GPIO/UART/SPI/I2C）
AN1652	SSBL サンプルプログラム	・ブート（SSBL）動作

©2023 Alpha Project Co., Ltd.

3.　e2 studio を用いた動作方法

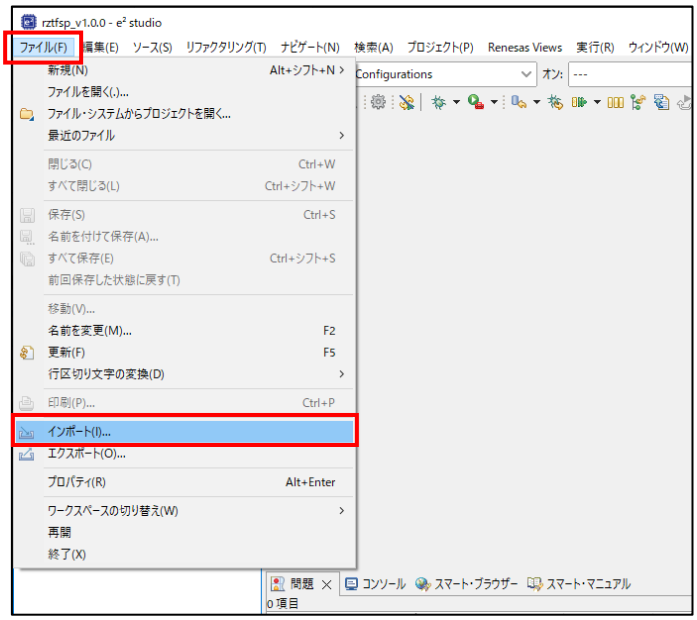
本章では、e2 studio を使用して CPU ボードを動作させる方法を説明します。
e2 studio および FSP はインストール済みであるものとしますので、事前にインストールを行ってください。

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-RZT2-0A 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。

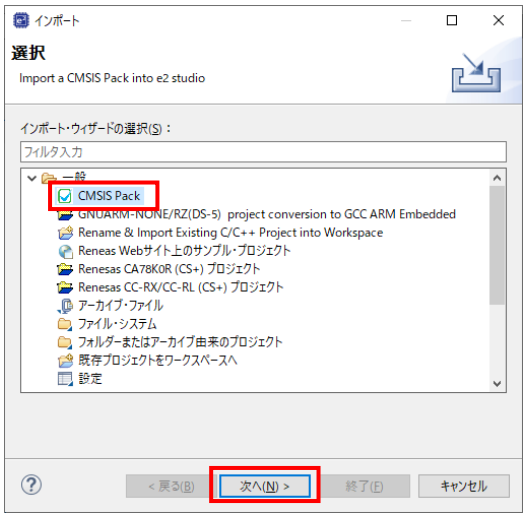
3.1.1　インポート方法

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。

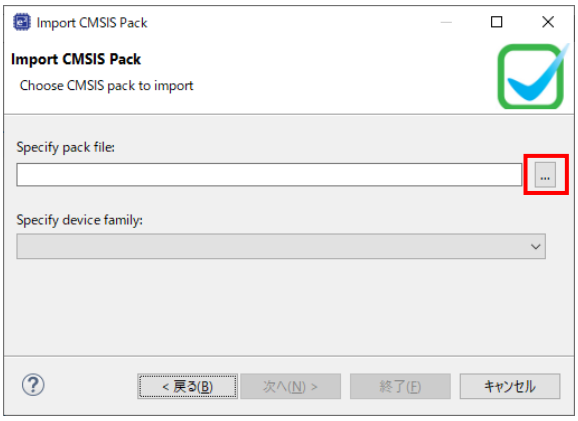


©2023 Alpha Project Co., Ltd.

② ［一般］→［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「AlphaProject.ap_rzt2_0a.1.2.0.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。

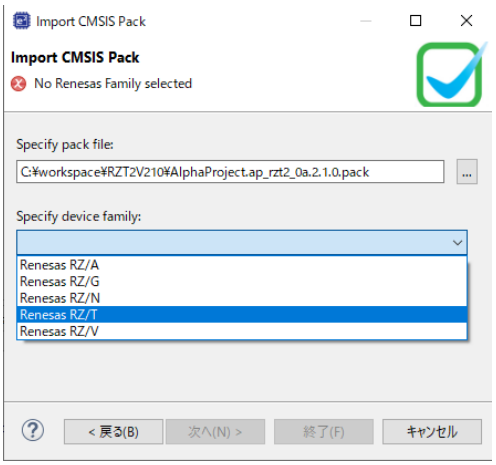


③ ［Import CMSIS Pack ウィンドウ］が表示されましたら、インポートする pack ファイルを選択してください。

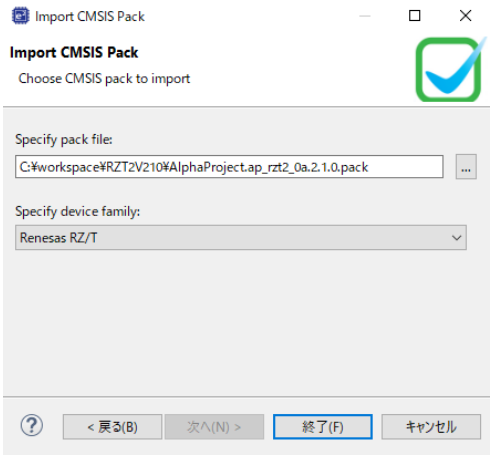


©2023 Alpha Project Co., Ltd.

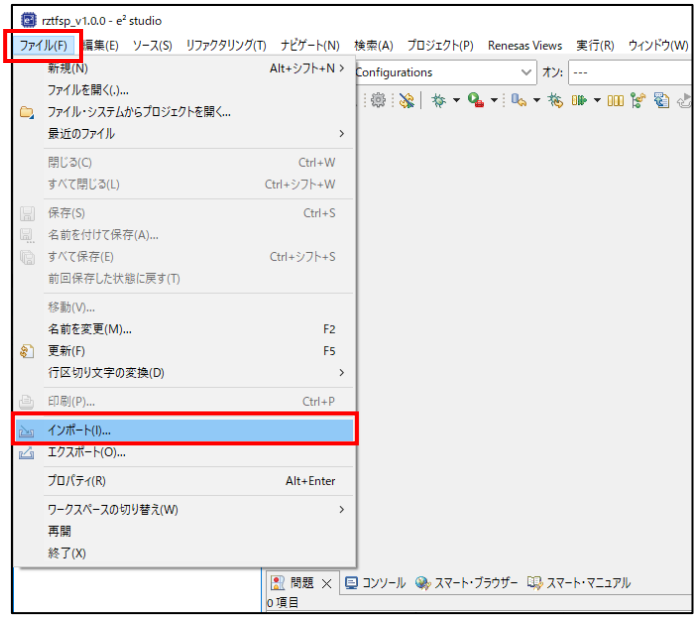
④　pack ファイルを選択後、メッセージ「No Renesas Family selected」が表示されるので、Specify device family から「Renesas RZ/T」を選択してください。



⑤　[終了]を選択します。

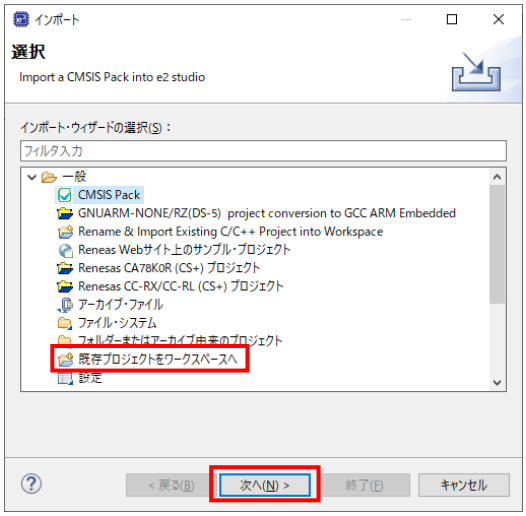


⑥　もう一度ツールバーの［ファイル］→［インポート］を選択します。

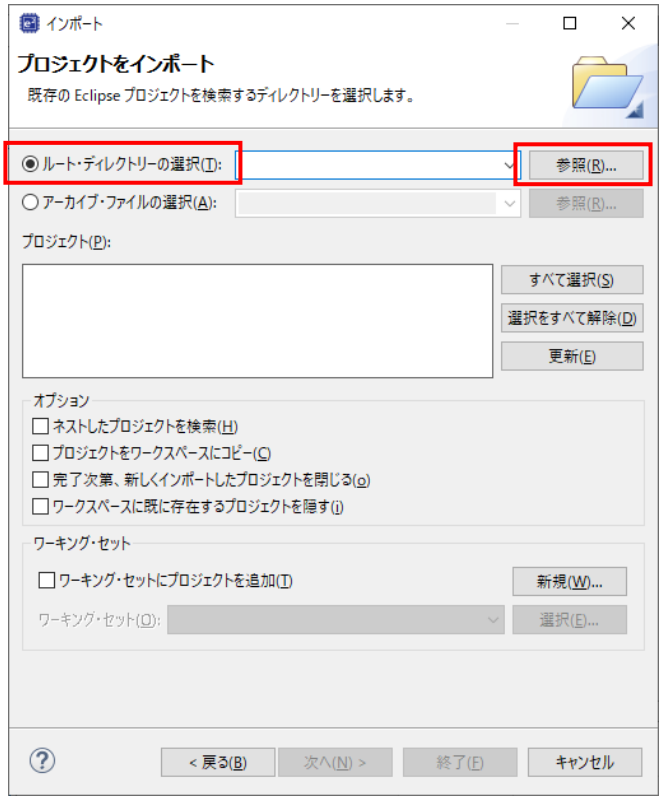


©2023 Alpha Project Co., Ltd.

⑦　［一般］→［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。

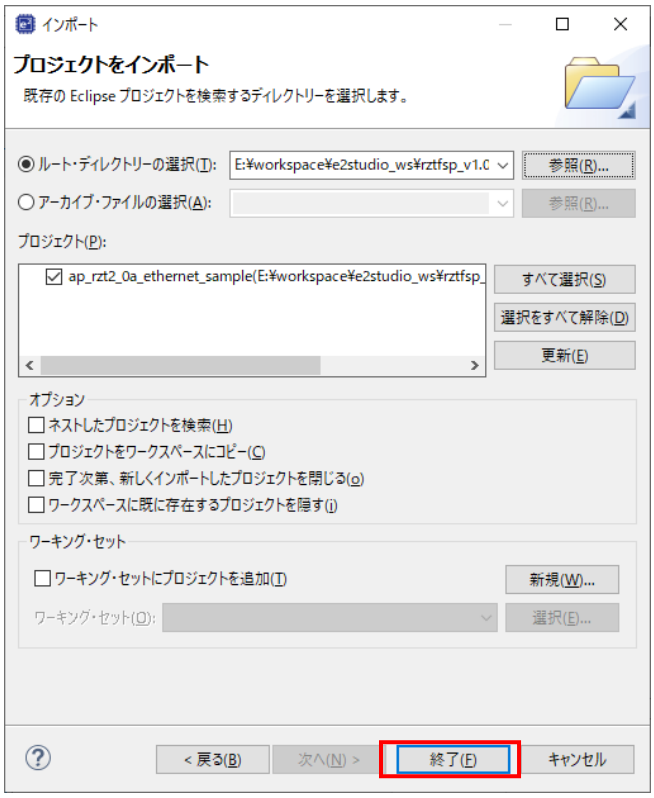


⑧　［ルート・ディレクトリーの選択］を選択し、［参照］からサンプルプログラムのフォルダを選択します。



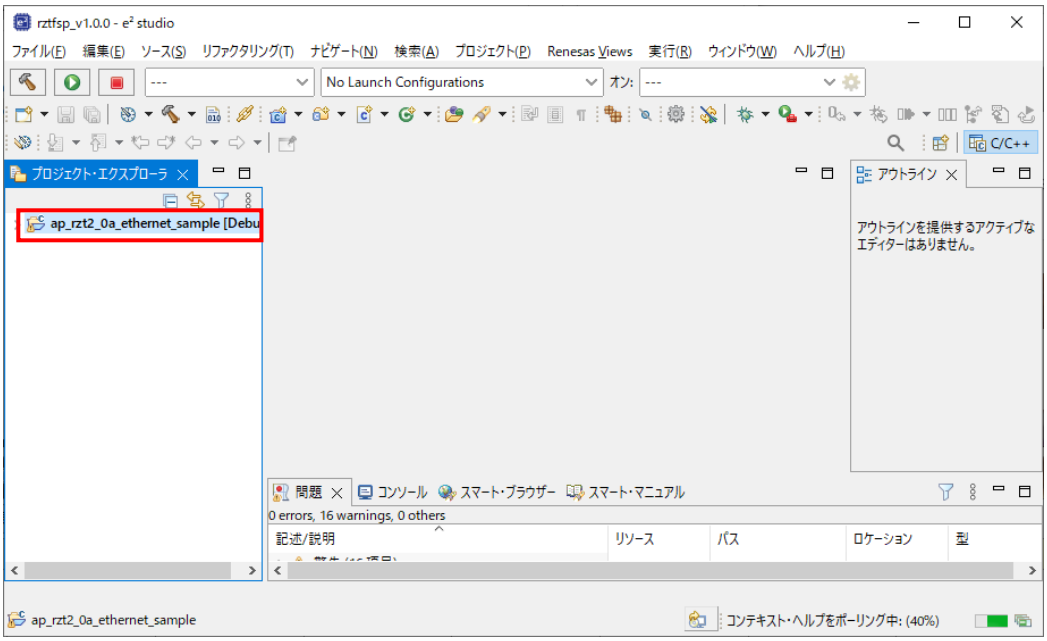
©2023 Alpha Project Co., Ltd.

⑨　［終了］を選択します。



©2023 Alpha Project Co., Ltd.

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。

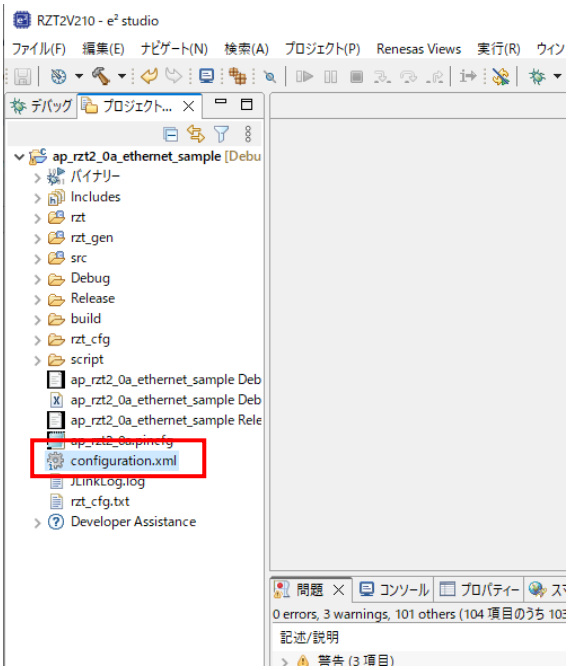


以上でプロジェクトのインポートは完了です。

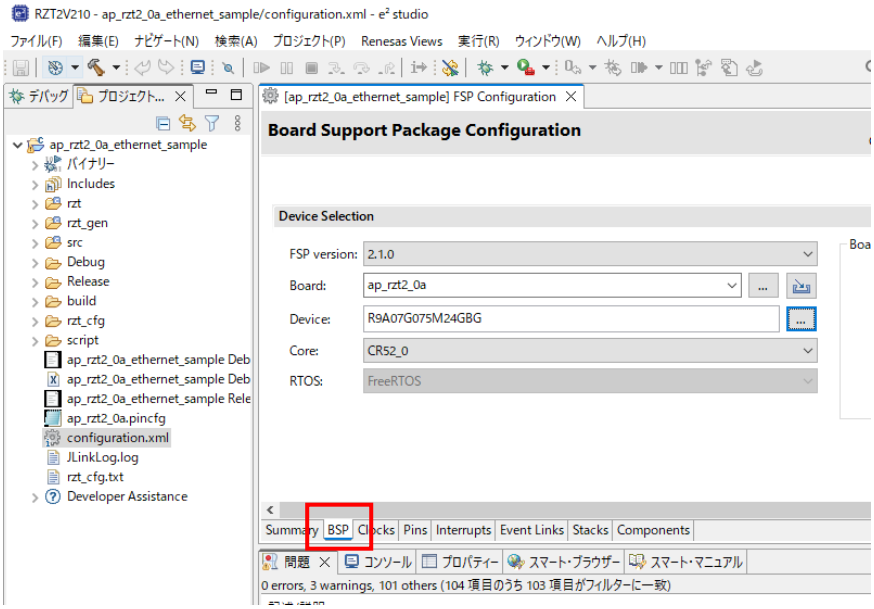
©2023 Alpha Project Co., Ltd.

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。

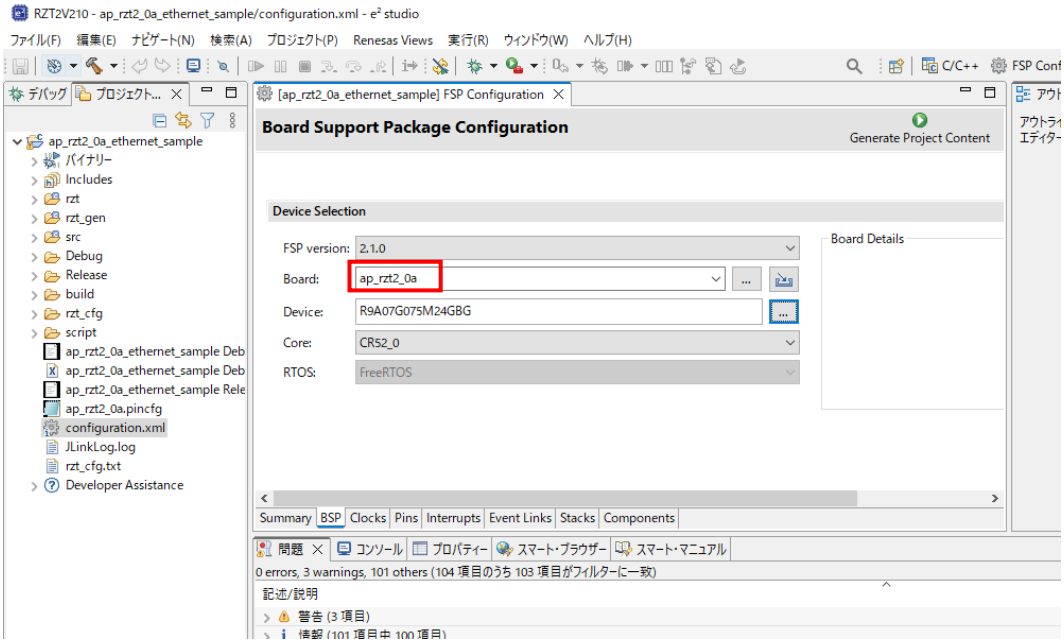


②　［BSP］タブを開きます。

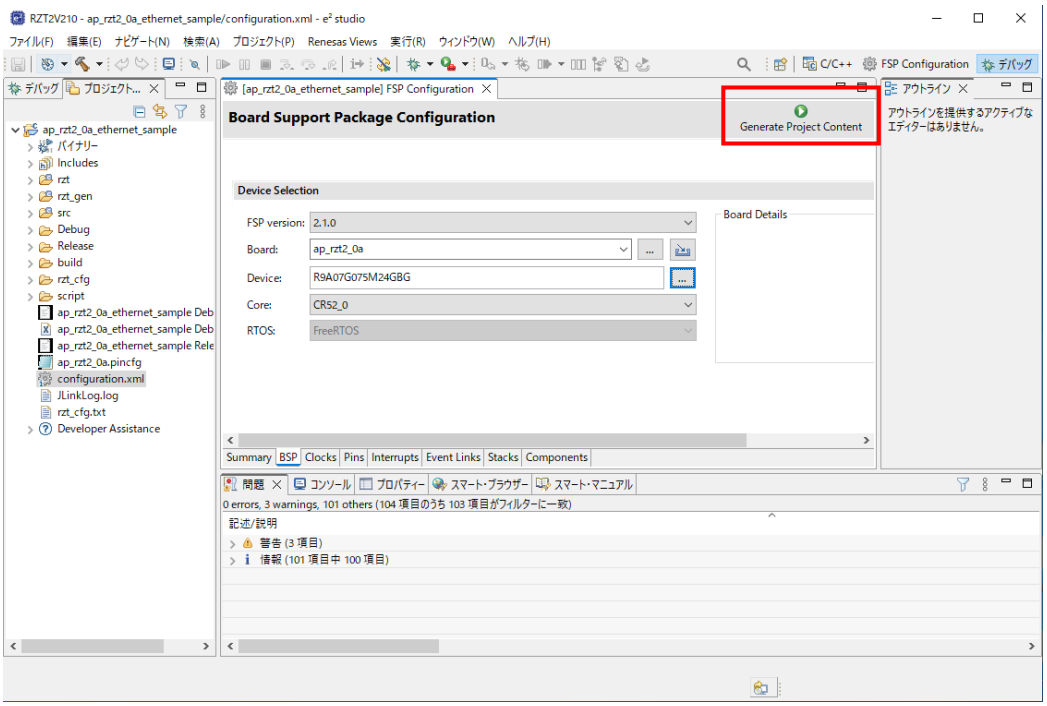


©2023 Alpha Project Co., Ltd.

③　［BSP］タブの［Board］が「ap_rzt2_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。

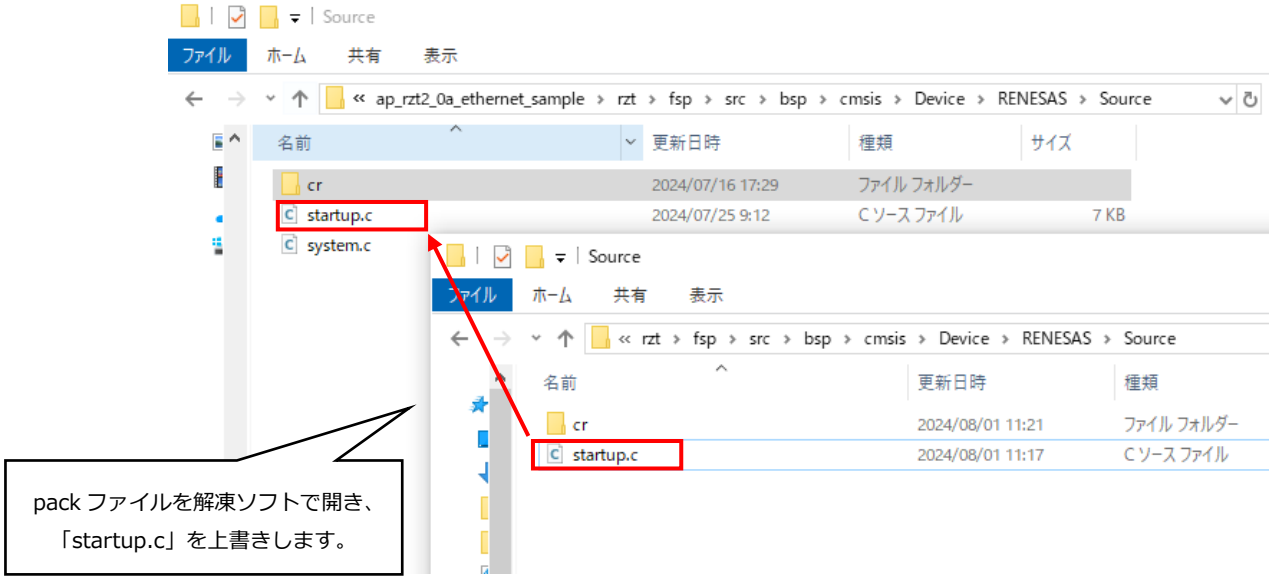


©2023 Alpha Project Co., Ltd.

⑤　プロジェクトに自動作成されたデータのうち、下記のファイルを、「3.1.1 インポート方法」でも使用した pack ファイル（本書の例では「AlphaProject.ap_rzt2_0a.2.1.0.pack」）の内容で上書きをします。

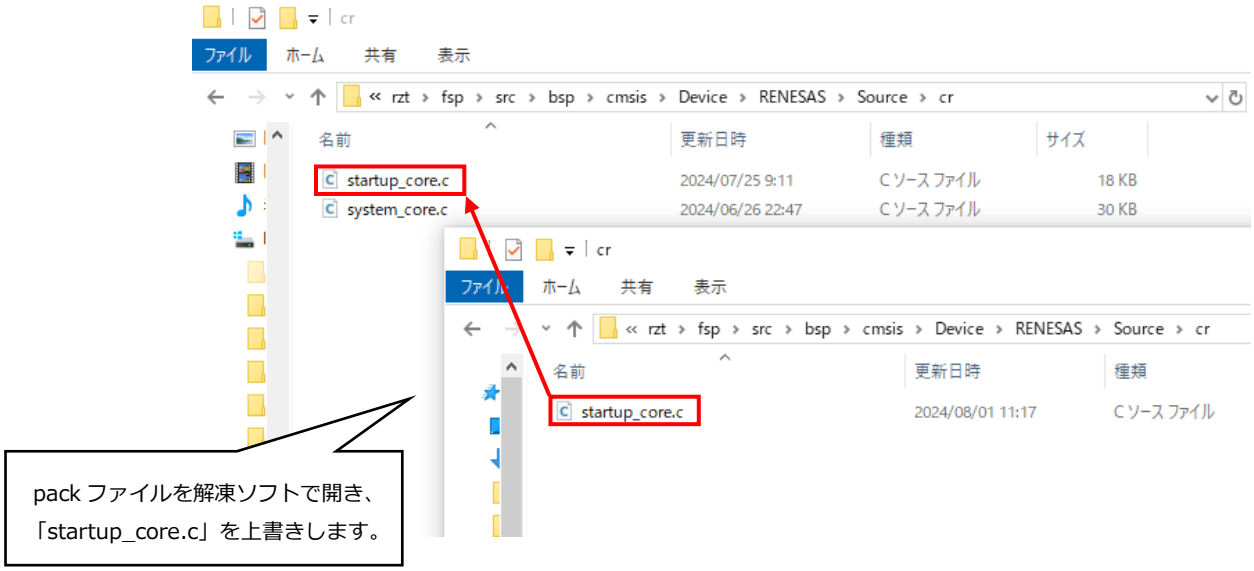
pack ファイルを解凍ソフトで開き、中に含まれる「startup.c」を、プロジェクト内に上書きコピーしてください。

対象ファイル：[プロジェクトフォルダ] ¥rzt ¥fsp ¥src ¥bsp ¥cmsis ¥Device ¥RENESAS ¥Source ¥startup.c
上書き用ファイル：[pack ファイル] ¥rzt ¥fsp ¥src ¥bsp ¥cmsis ¥Device ¥RENESAS ¥Source ¥startup.c



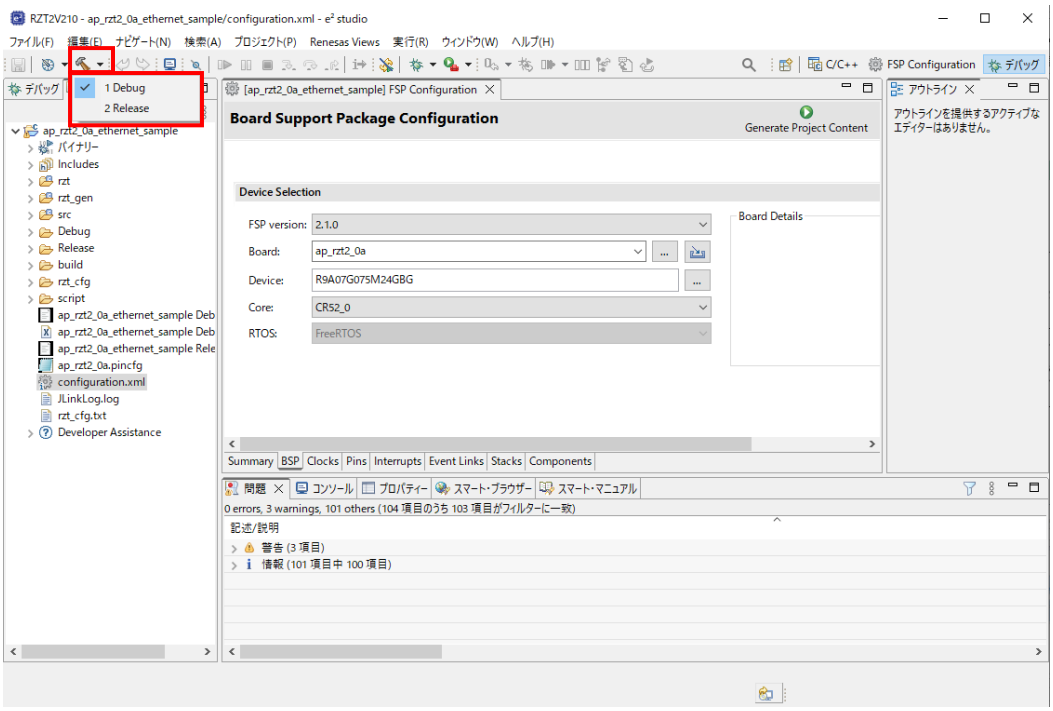
同様に pack ファイル内の「startup_core.c」を、プロジェクトの下記の場所に上書きコピーしてください。

対象ファイル：[プロジェクトフォルダ] ¥rzt ¥fsp ¥src ¥bsp ¥cmsis ¥Device ¥RENESAS ¥Source ¥cr ¥startup_core.c
上書き用ファイル：[pack ファイル] ¥rzt ¥fsp ¥src ¥bsp ¥cmsis ¥Device ¥RENESAS ¥Source ¥cr ¥startup_core.c



©2023 Alpha Project Co., Ltd.

⑥　ツールバーのビルドアイコンからビルドを行います。
弊社サンプルプログラムの場合、基本的に「Debug」「Release」の 2 種類のビルド設定がデフォルトで用意されています。
「Debug」ビルドの場合は ¥Debug ワークフォルダに、「Release」ビルドの場合は ¥Release ワークフォルダにそれぞれオブジェクトファイルが生成されます。



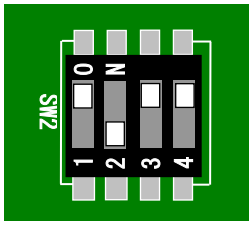
e2 studio の詳細な使用方法に関しては、 e2 studio のマニュアルを参照してください。

©2023 Alpha Project Co., Ltd.

3.1.3　RAM デバッグ方法

①　「3.1.2　ビルド方法」を参考に、デバッグをする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_rzt2_0a_ethernet_sample」とします。）

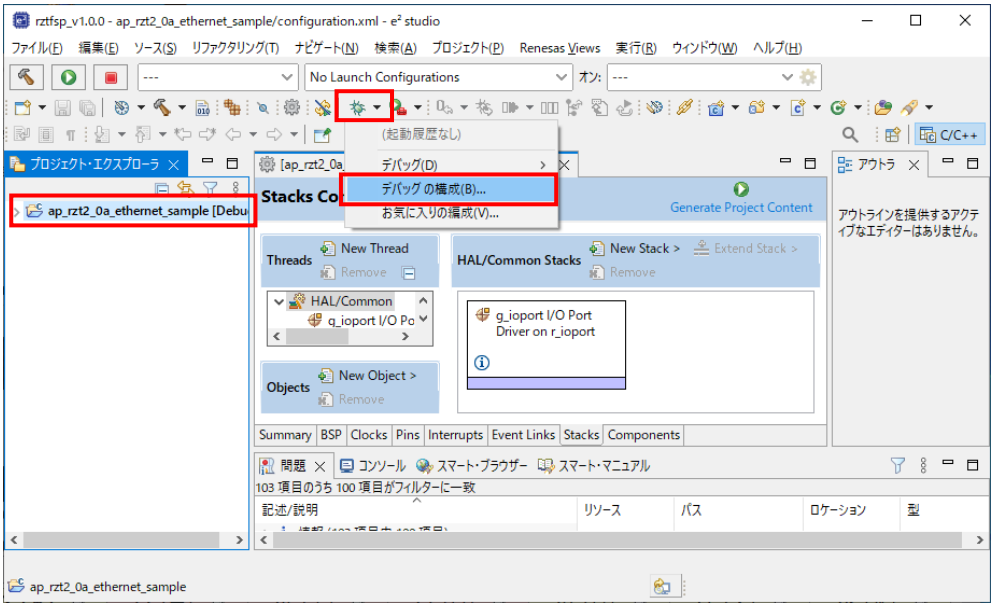
②　ボード上のディップスイッチを設定します。
サンプルプログラム動作時は、AP-RZT2-0A・AP-RZT2-1A 共通で以下の通りに設定してください。



<SW2 設定>	
ブートモード	：シリアル FlashROM ブート/USB ブート/SCI ブートのいずれにも該当しないモード
JTAG Hash モード	：使用する（不問）

③　ボードに電源を投入してください。

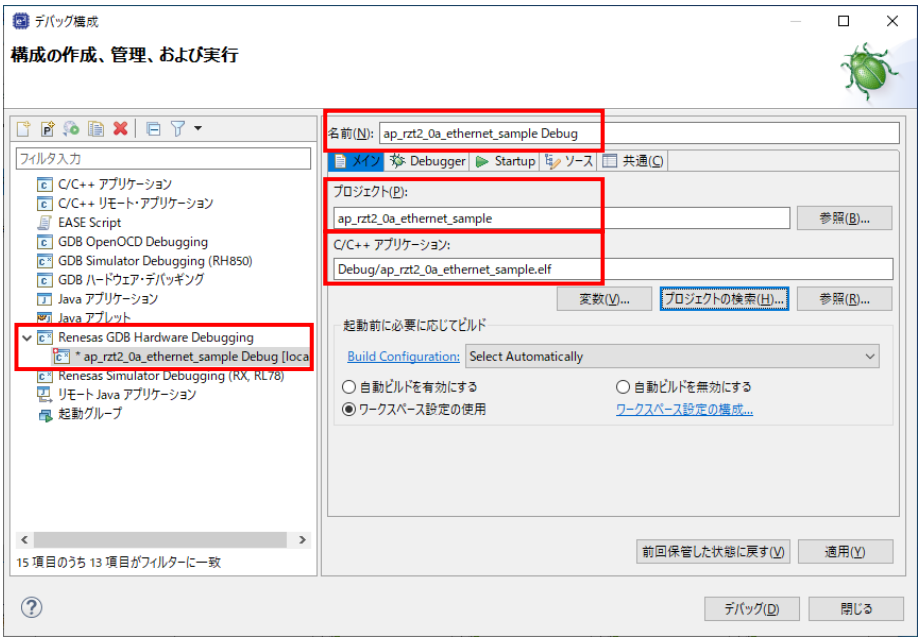
④　プロジェクトを選択し、ツールバーのデバッグアイコンから［デバッグの構成］を開きます。



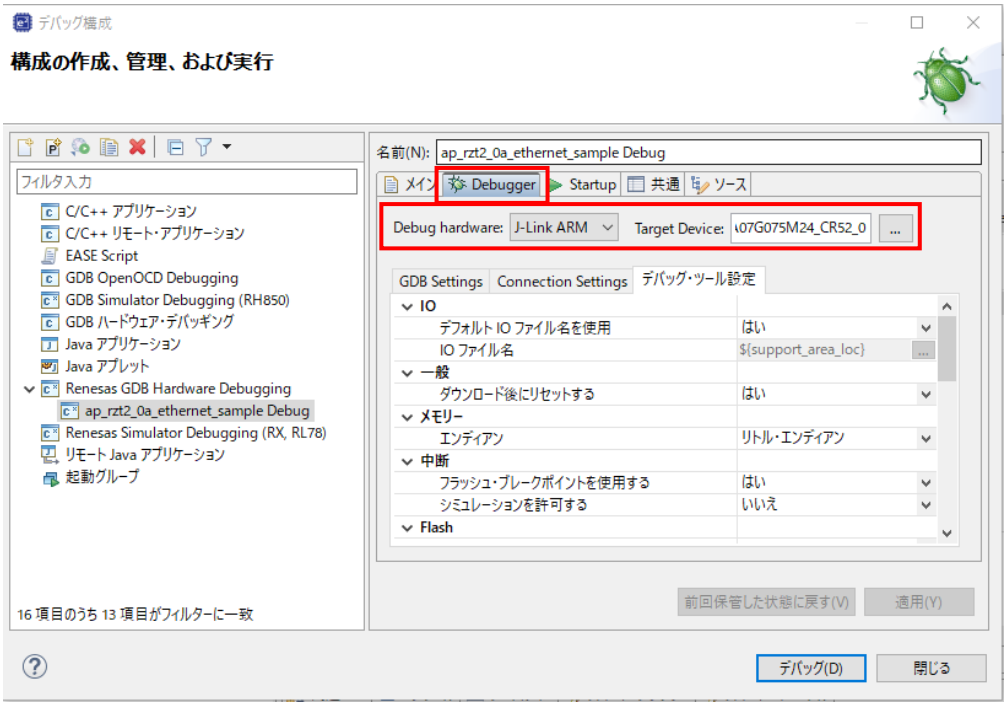
©2023 Alpha Project Co., Ltd.

⑤　［Renesas GDB Hardware Debugging］のデバッグ設定から［<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。
また、デバッグ設定が見つからない場合はデバッグ設定を新規作成してください。

- [名前]：<プロジェクト名> Debug
- [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

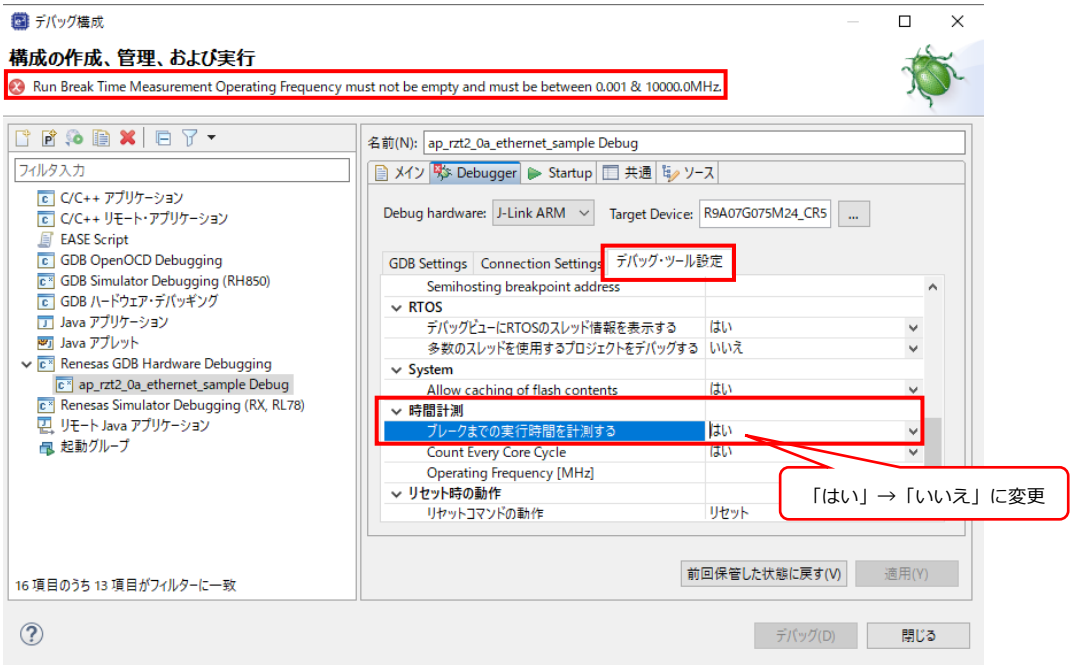


⑥　［Debugger］タブを選択し、［Debug hardware］を［J-Link ARM］、
［Target Device］を「R9A07G075M24_CR52_0」※に設定されていることを確認してください。
※AP-RZT2-1A の場合は「R9A07G075M01」を設定してください。

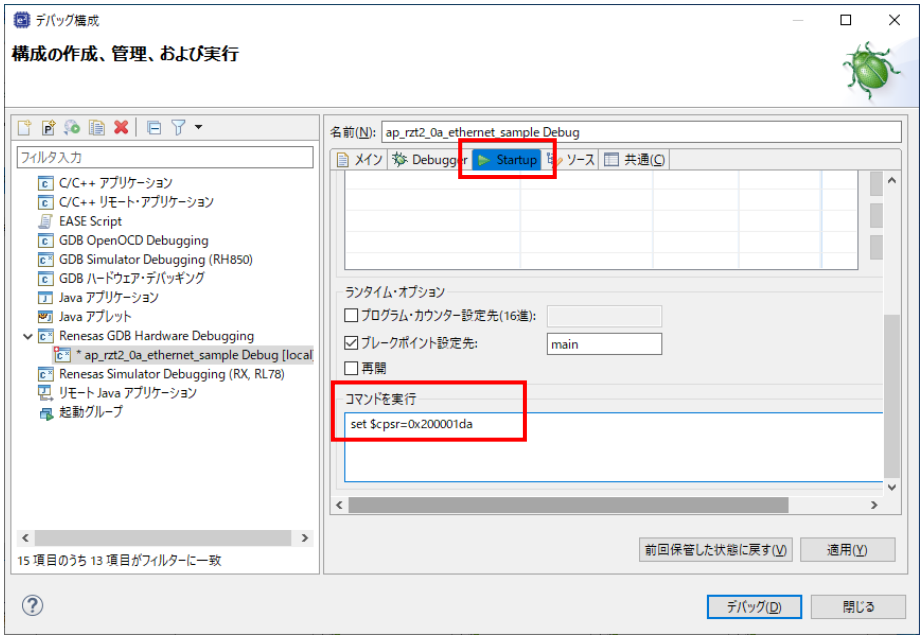


©2023 Alpha Project Co., Ltd.

デバッグ設定を新規作成した場合などに、［Target Device］に「R9A07G075M24_CR52_0］を選択するとウィンドウ上にエラーメッセージ「Run Break Time ～」が表示されることがあります。
エラーメッセージが表示された場合は、「デバッグ・ツール設定」＞「時間計測」＞「ブレークまでの実行時間を計測する」を「いいえ」に変更してください。

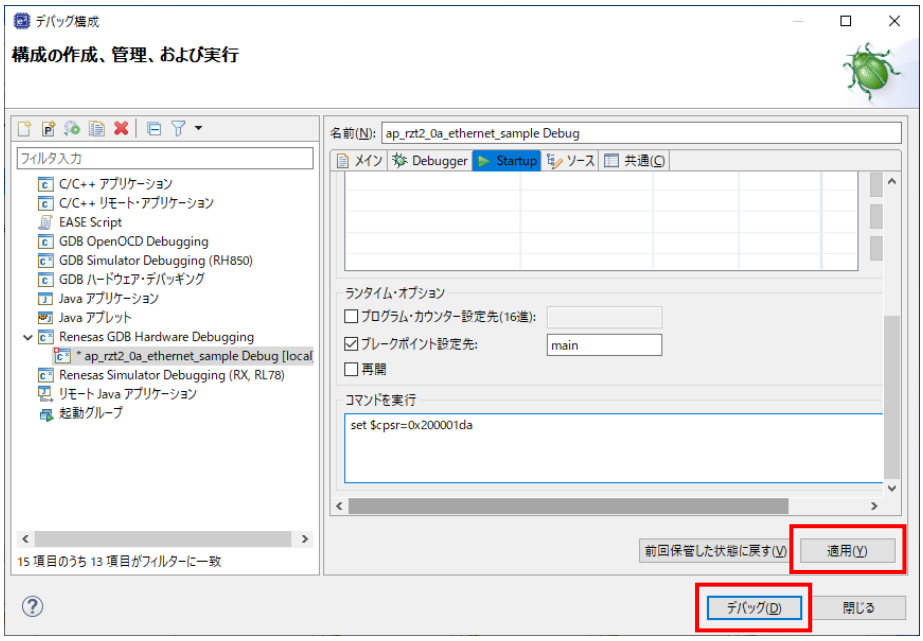


⑦　［Startup］タブを選択し、［コマンドを実行］に「set $cpsr=0x200001da」を設定してください。

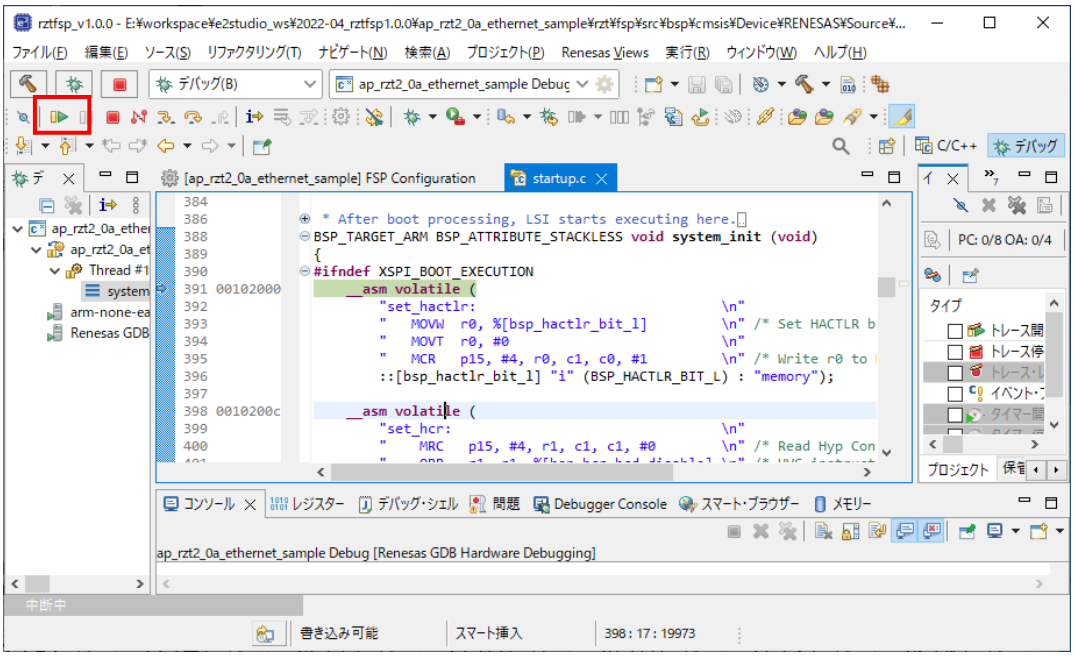


©2023 Alpha Project Co., Ltd.

⑧　「適用」ボタンを押して設定を保存し、続けて「デバッグ」ボタンを押します。



⑨　ボードとの接続が完了したら「再開」ボタンを押し、サンプルプログラムを動作させてください。



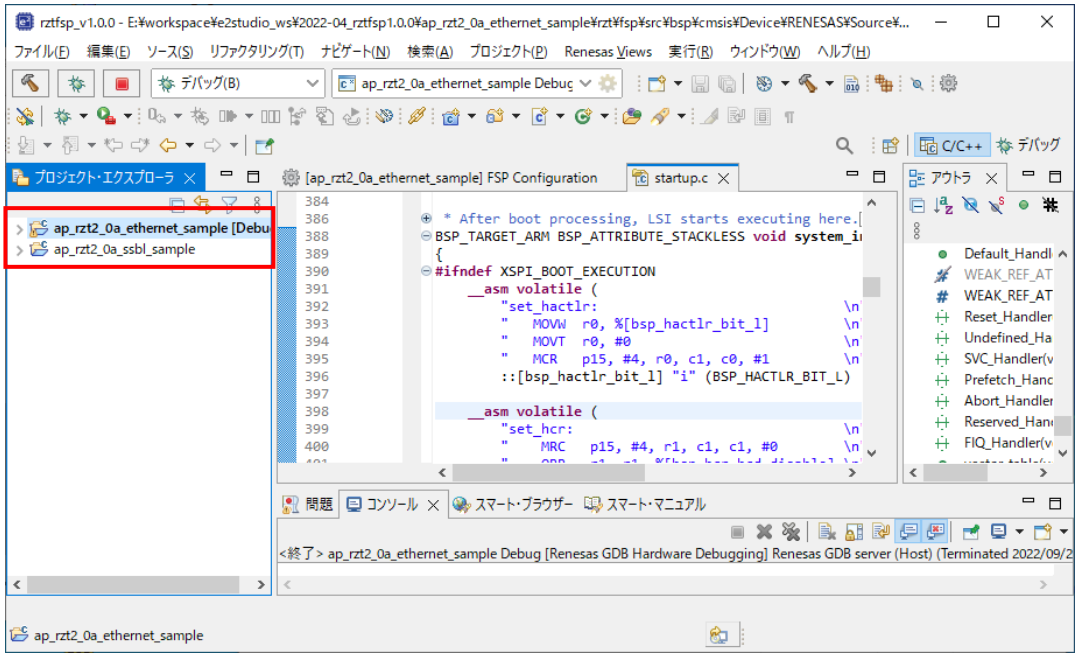
プログラムの動作については、デバッグするサンプルプログラムのアプリケーションノートをご参照ください。

©2023 Alpha Project Co., Ltd.

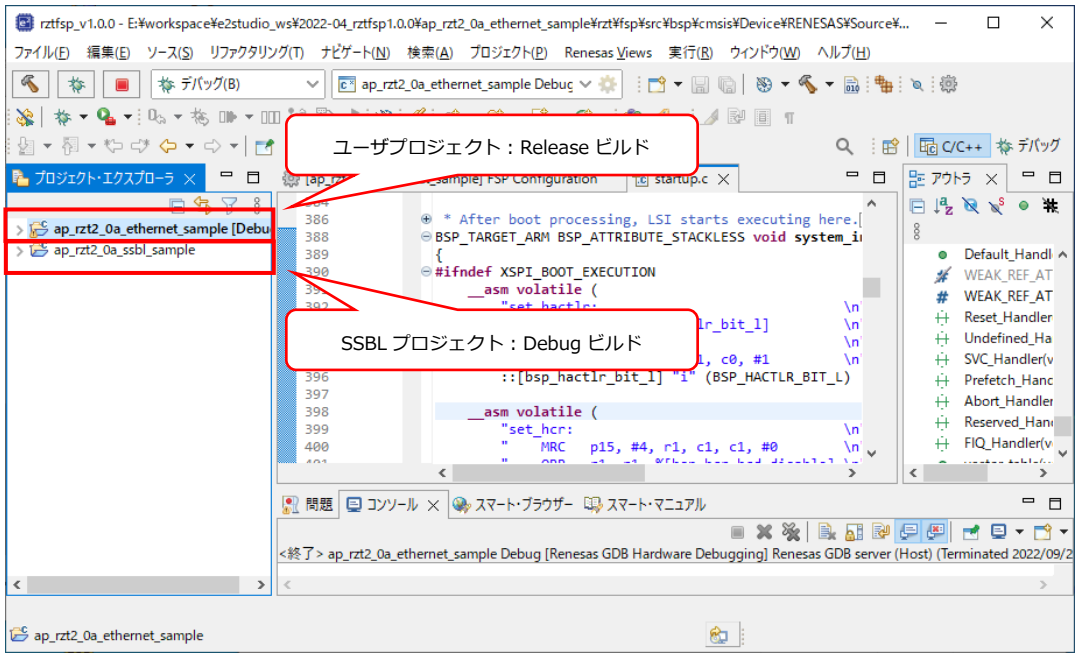
3.1.4　ROM デバッグ方法

本項ではボード上のシリアル FlashROM に書き込まれたプログラムを対象にデバッグをする手順を示します。
後述の「5.　シリアル FlashROM への書き込み」を参考に、あらかじめプログラムをシリアル FlashROM へ書き込んでから読み進めてください。

① ROM デバッグを行うユーザプロジェクトとボード付属の SSBL サンプルプロジェクトを、「3.1.1　インポート方法」を参考に e2studio にインポートします。
（例で使用する画像では、ユーザプロジェクト名は「ap_rzt2_0a_ethernet_sample」とします。）

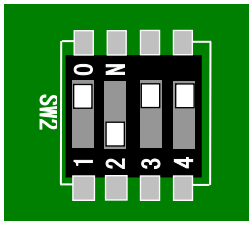


② ROM デバッグを行うユーザプロジェクトには「Release」ビルド、「SSBL サンプルプロジェクト」には「Debug」ビルドをそれぞれ「3.1.2　ビルド方法」を参考にビルドします。



©2023 Alpha Project Co., Ltd.

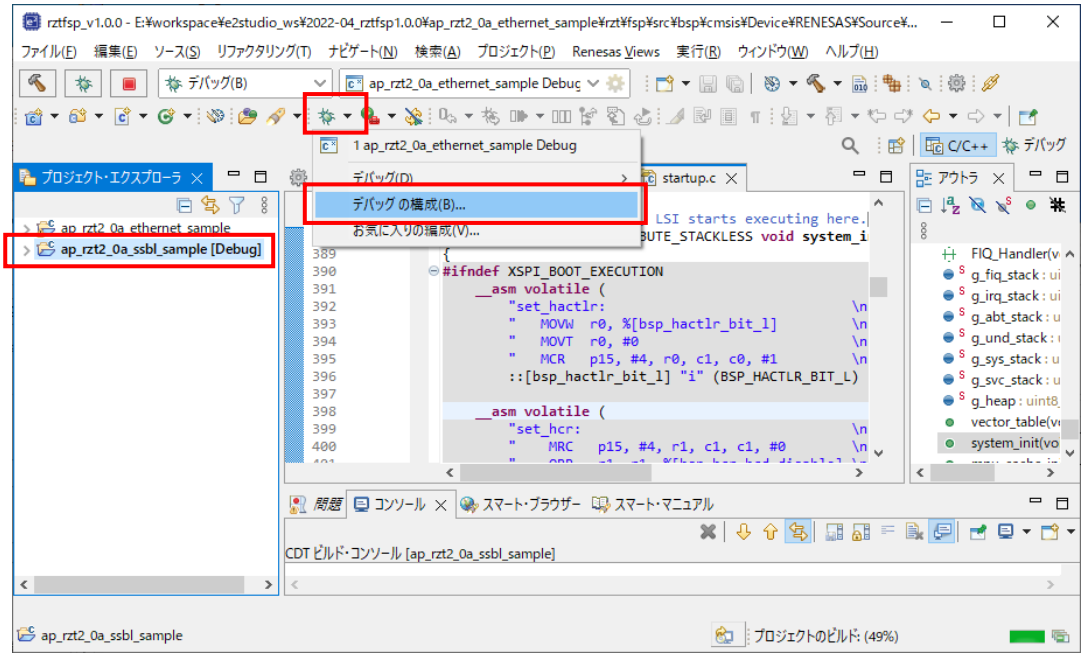
③　ボード上のディップスイッチを設定します。
AP-RZT2-0A・AP-RZT2-1A 共通で以下の通りに設定してください。



<SW2 設定>	
ブートモード	：シリアル FlashROM ブート/USB ブート/SCI ブートのいずれにも該当しないモード
JTAG Hash モード	：使用する（不問）

④　ボードに電源を投入してください。

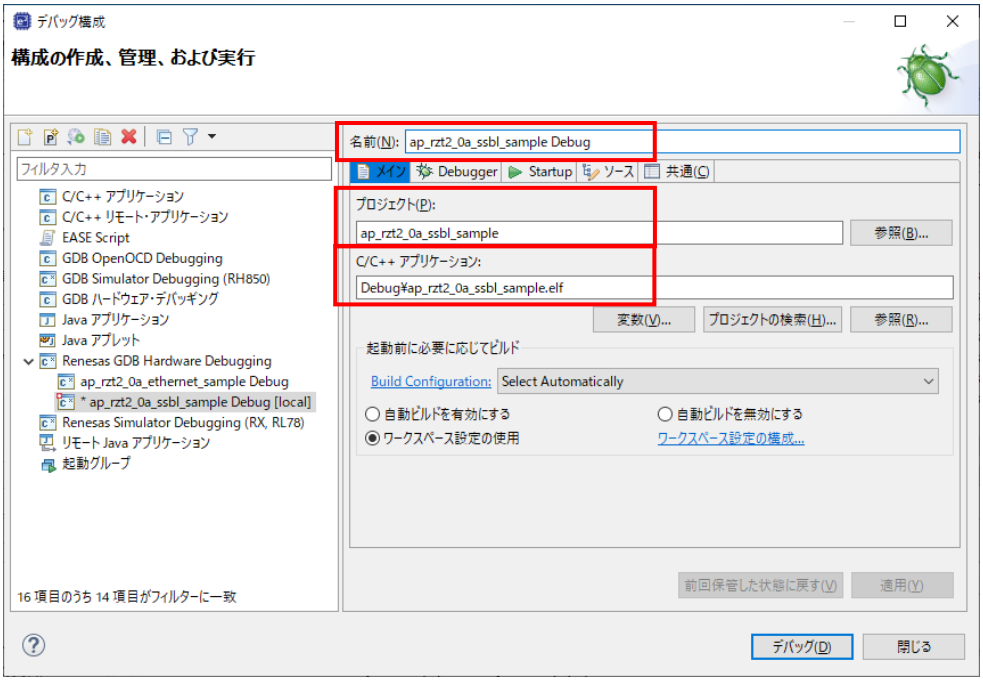
⑤　SSBL プロジェクトを選択し、ツールバーのデバッグアイコンから［デバッグの構成］を開きます。



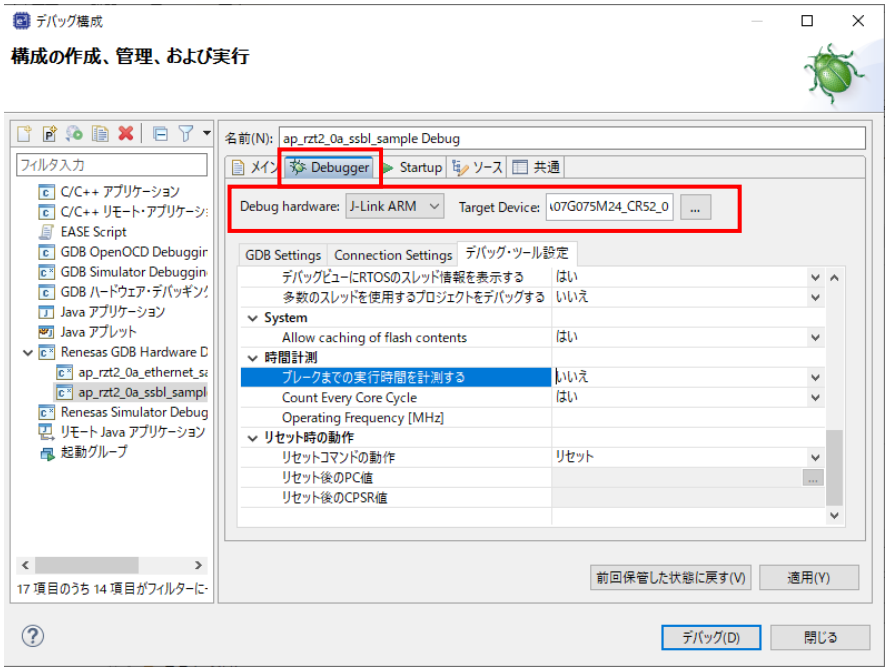
©2023 Alpha Project Co., Ltd.

⑥　［Renesas GDB Hardware Debugging］のデバッグ設定から［ap_rzt2_0a_ssbl_sample 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。
また、デバッグ設定が見つからない場合はデバッグ設定を新規作成してください。

- [名前]：ap_rzt2_0a_ssbl_sample Debug
- [プロジェクト]：ap_rzt2_0a_ssbl_sample
- [C/C++アプリケーション]：Debug¥ ap_rzt2_0a_ssbl_sample.elf

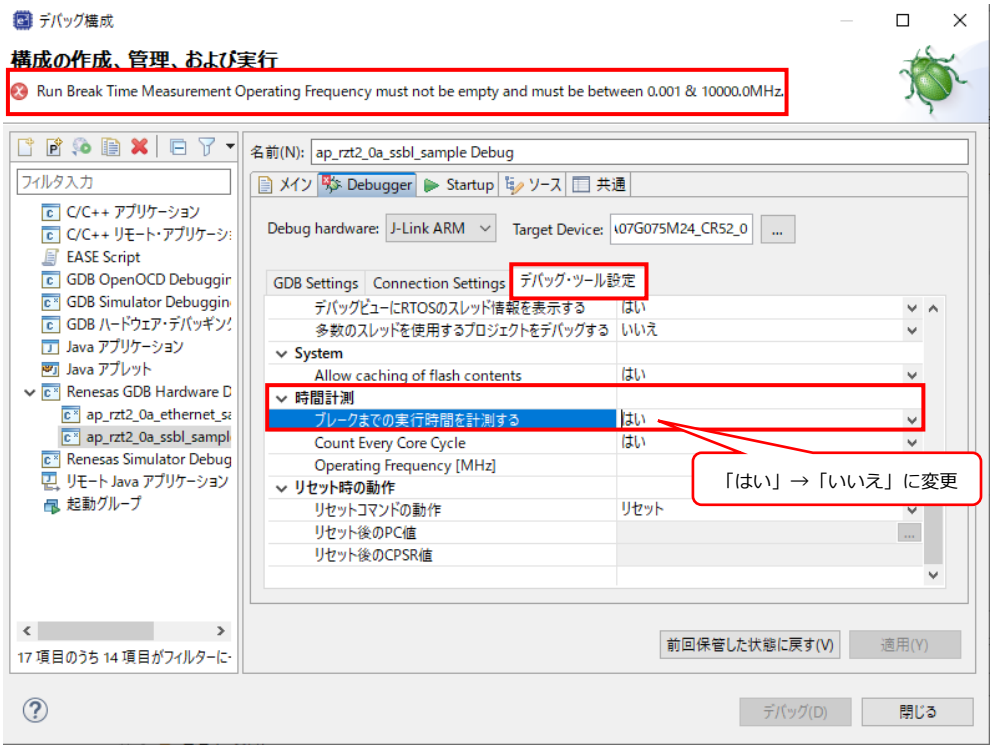


⑦　［Debugger］タブを選択し、［Debug hardware］を［J-Link ARM］、
［Target Device］を［R9A07G075M24_CR52_0］※に設定されていることを確認してください。
※AP-RZT2-1A の場合は［R9A07G075M01］を設定してください

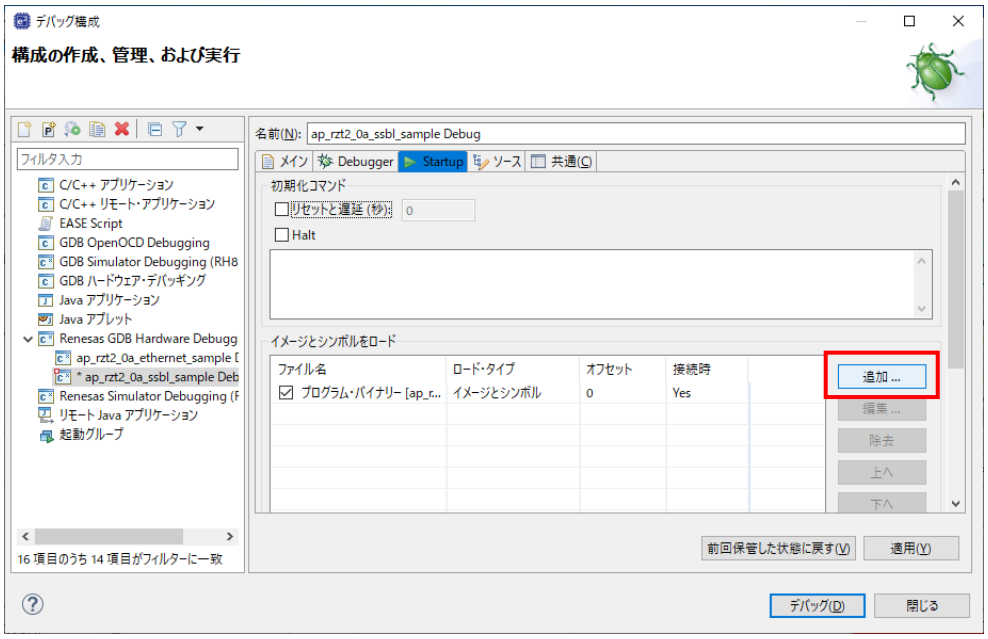


©2023 Alpha Project Co., Ltd.

デバッグ設定を新規作成した場合などに、［Target Device］を設定するとウィンドウ上にエラーメッセージ「Run Break Time ～」が表示されることがあります。
エラーメッセージが表示された場合は、「デバッグ・ツール設定」＞「時間計測」＞「ブレークまでの実行時間を計測する」を「いいえ」に変更してください。

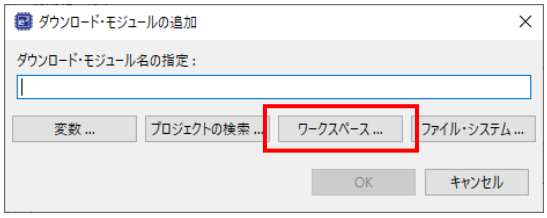


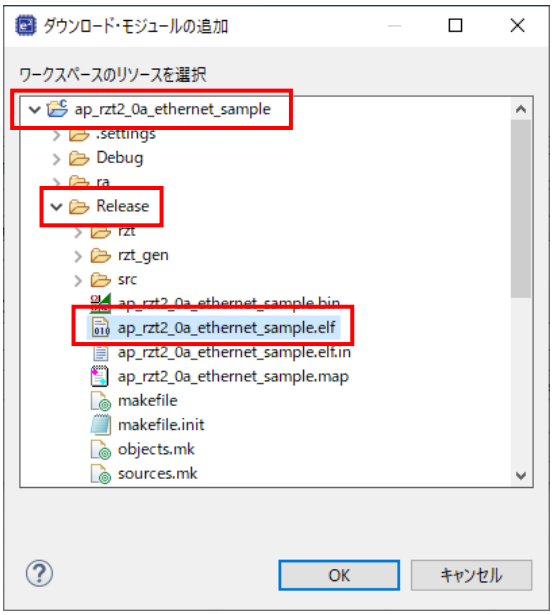
⑧　［Startup］タブを選択し、［イメージとシンボルをロード］の「追加」ボタンを押してください。



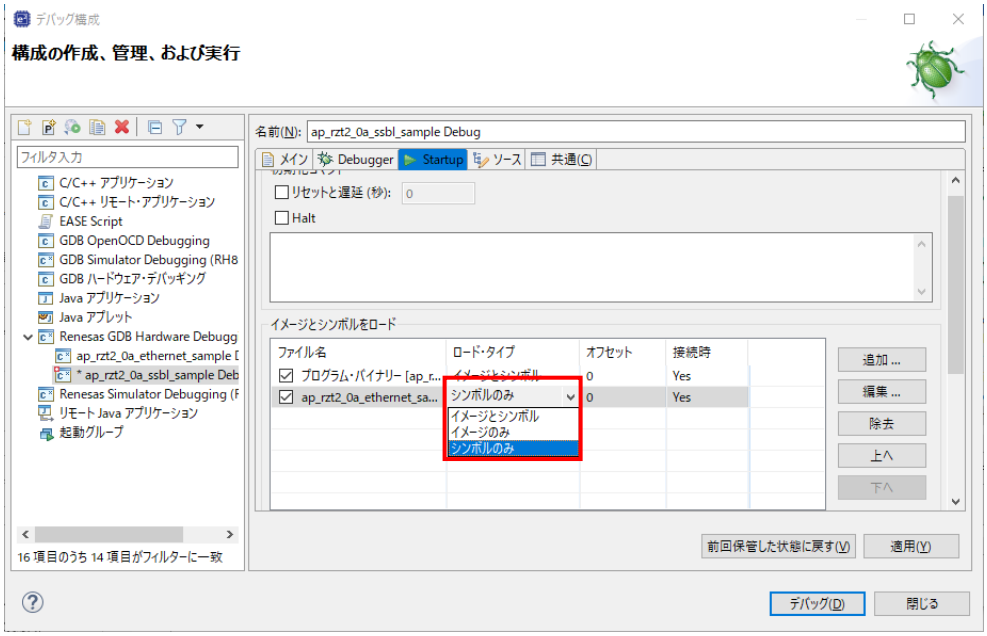
©2023 Alpha Project Co., Ltd.

⑨　「ワークスペース」を押し、ROM デバッグを行いたいプロジェクトの「Release¥ <プロジェクト名>.elf」を選択してウィンドウを閉じてください。



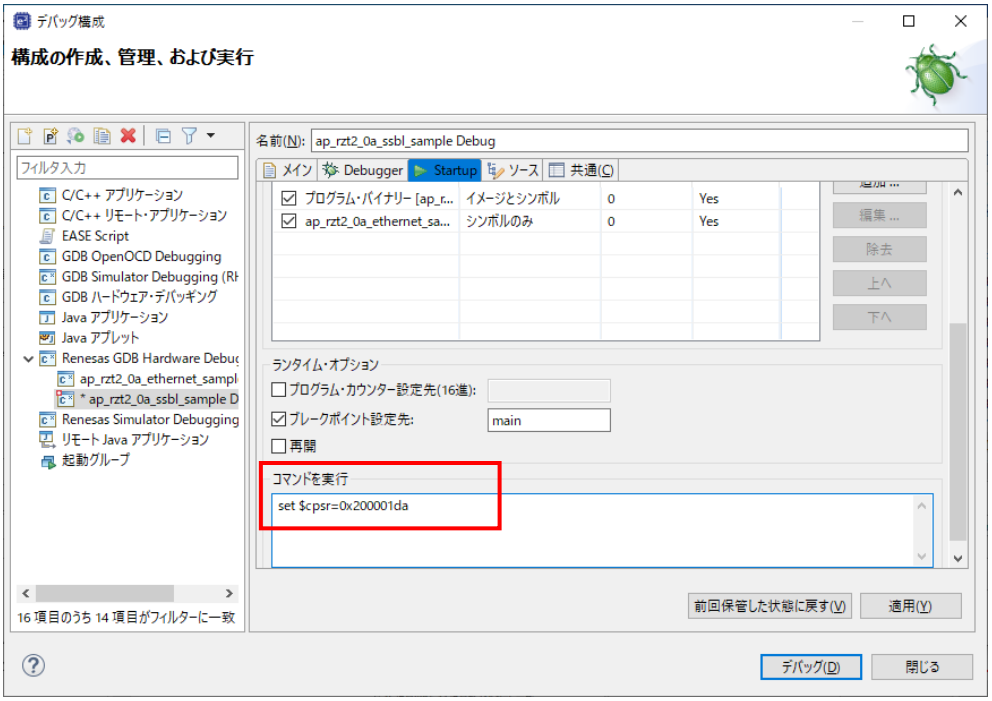


⑩　［イメージとシンボルをロード］に追加できた項目の「ロード・タイプ」を「シンボルのみ」に変更してください。

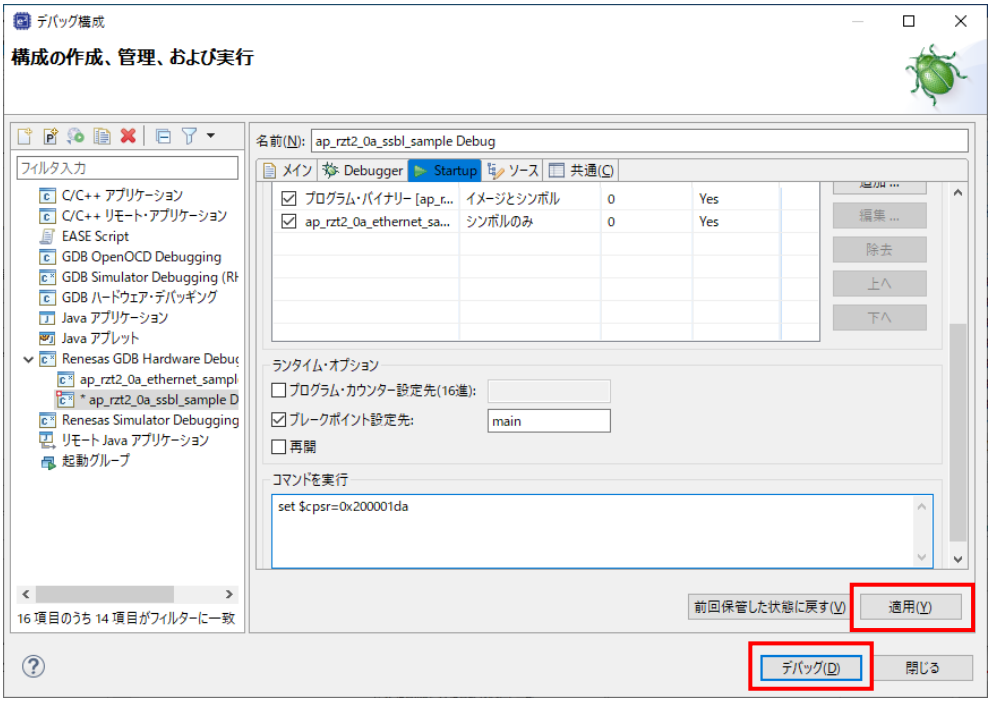


©2023 Alpha Project Co., Ltd.

⑪　［コマンドを実行］に「set $cpsr=0x200001da」を設定してください。

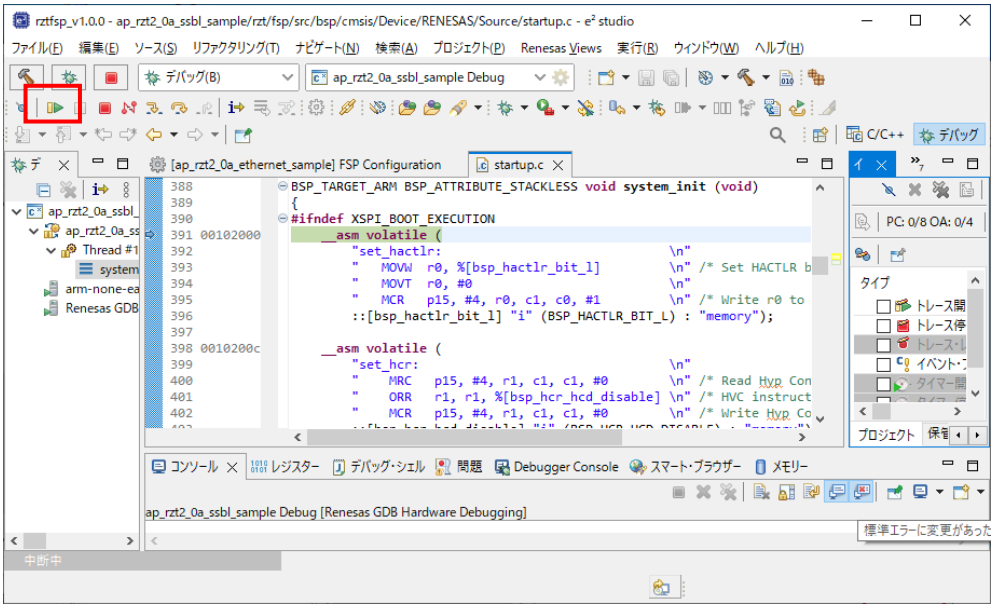


⑫　「適用」ボタンを押して設定を保存し、続けて「デバッグ」ボタンを押します。



©2023 Alpha Project Co., Ltd.

⑬　ボードとの接続が完了したらプログラムを「再開」ボタンを押し、サンプルプログラムを動作させてください。



プログラムの動作については、デバッグす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。

©2023 Alpha Project Co., Ltd.

4.　e2 studio を用いた新規プロジェクト作成方法

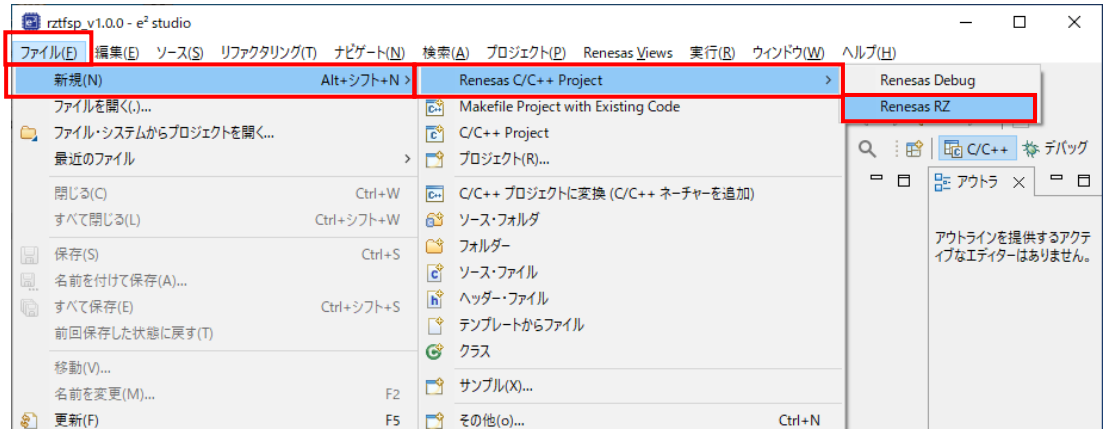
本章では、e2 studio で CPU ボード用プログラムを作成する方法を説明します。
プロジェクトコンフィギュレータの設定を行うプロジェクトとして、「4.1　UART プログラムの作成例」にて、UART を使用したプログラムの作成方法を説明します。
なお、本書で扱っている FSP は バージョン 1.2.0 です。異なるバージョンの FSP をご使用の場合は、本書で解説する項目・設定等が異なる場合がございますのでご注意ください。

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

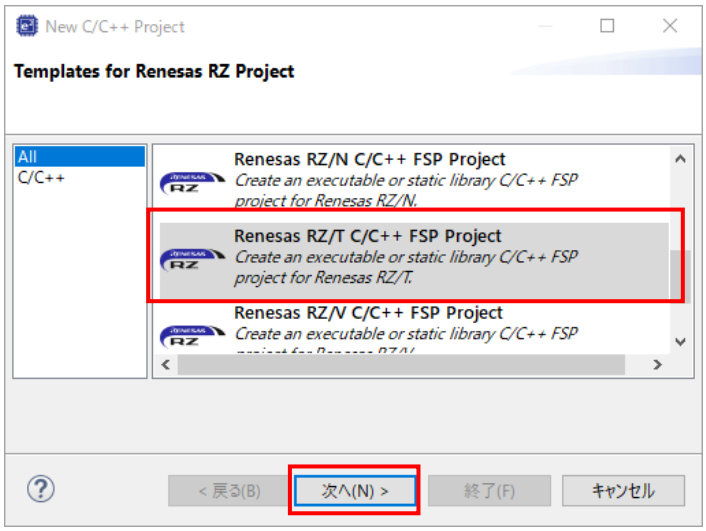
本節では、AP-RZT2-0A で動作する UART のプログラムを例に、プロジェクトの作成方法を説明します。

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

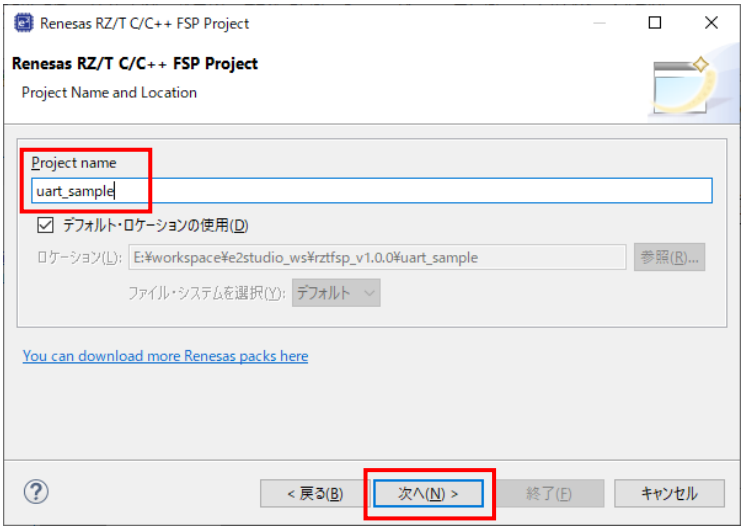


©2023 Alpha Project Co., Ltd.

② ［Renesas RZ/T C/C++ FSP Project］を選択し、[次へ]を押します。



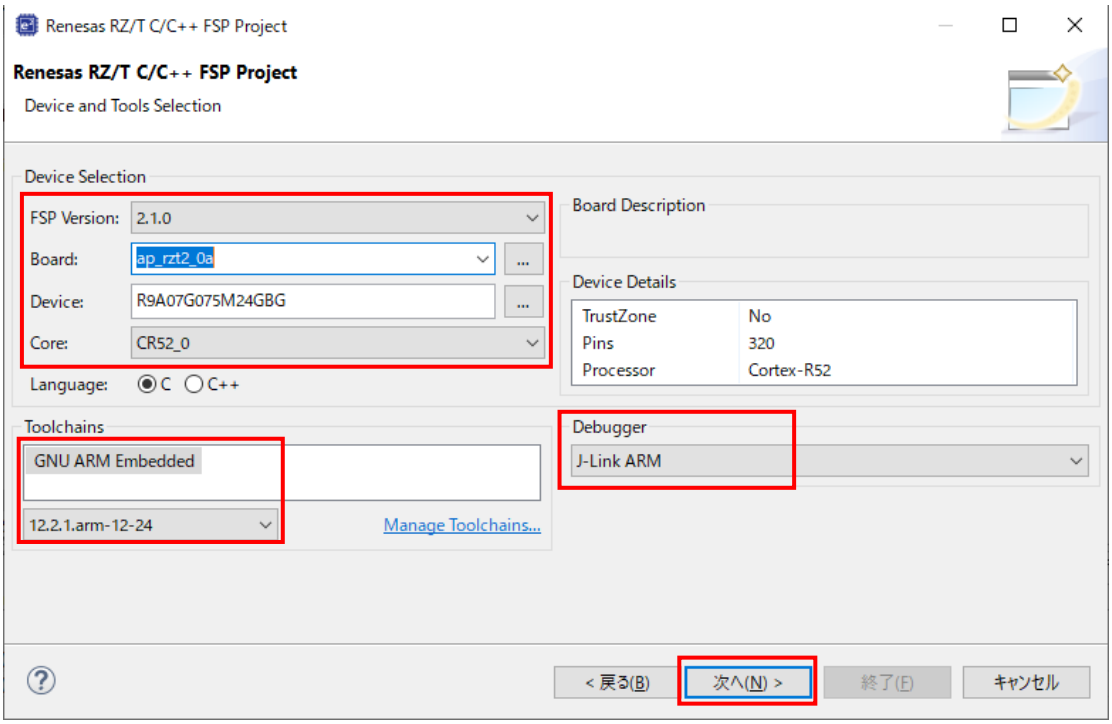
③ ［Project name］に任意のプロジェクト名を入力します（ここでは「uart_sample」とします）。入力後、［次へ］を押して次へ進みます。



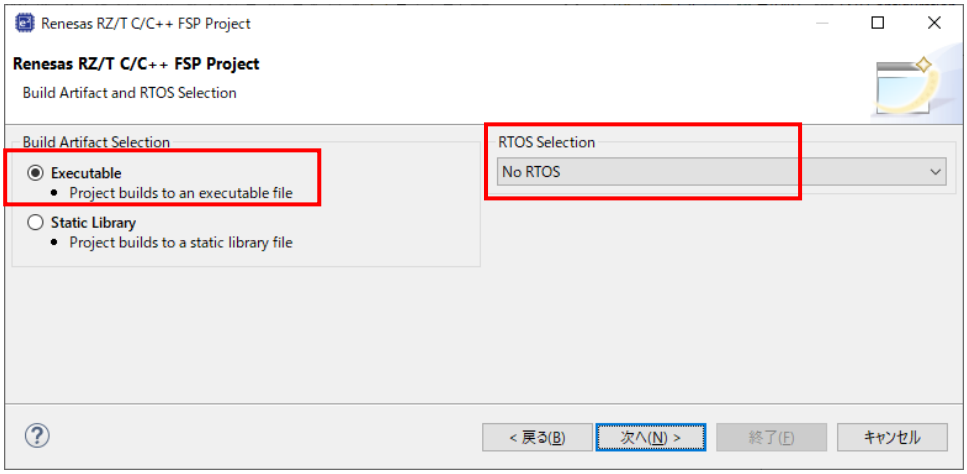
©2023 Alpha Project Co., Ltd.

④　Device Selection 設定では以下の表に従ってそれぞれ設定します。
設定後、［次へ］を押します。

FSP Version	2.1.0
Board	ap_rzt2_0a ※AP-RZT2-1A の場合は「ap_rzt2_1a」を選択してください。
Device	R9A07G075M24GBG ※AP-RZT2-1A の場合は「R9A07G075M01GFA」を選択してください。
Core	CR52_0
Toolchains	GNU ARM Embedded/12.2.1.arm-12-24
Debugger	J-Link ARM

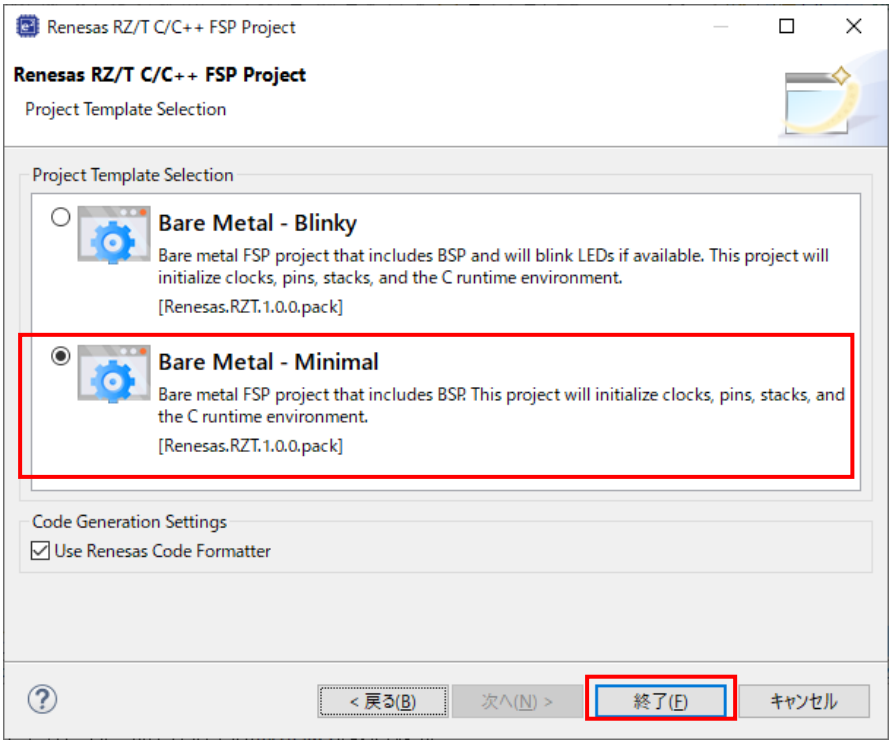


⑤　ビルド生成物と RTOS 設定を設定します。
サンプルでは［Build Artifact Selection］に［Executable］、［RTOS Selection］に［No RTOS］を選択します。
設定後、［次へ］を押します。

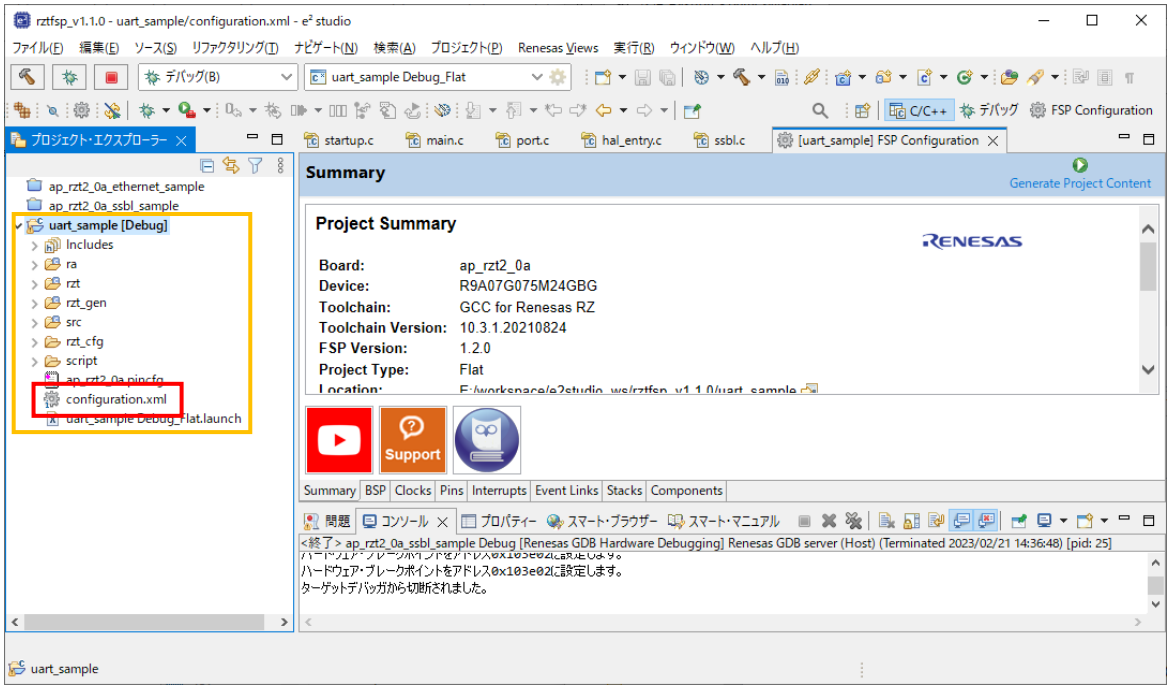


©2023 Alpha Project Co., Ltd.

⑥　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。

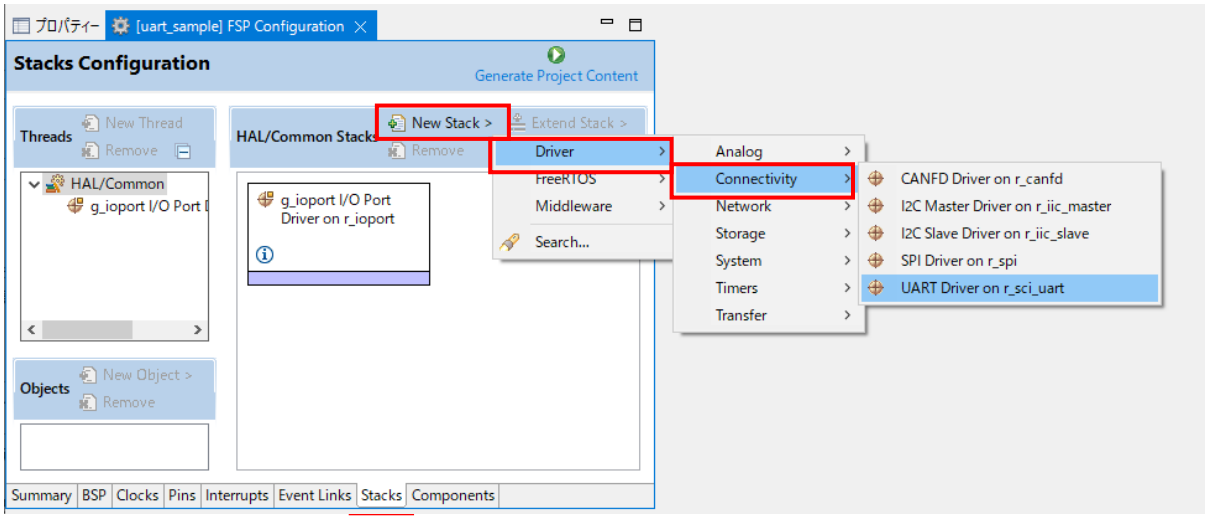


⑦　プロジェクトが作成されました。
以降では、FSP スマートコンフィギュレータでプロジェクトの設定を行います。
プロジェクトのコンフィギュレータファイル（configration.xml）を開いてください。

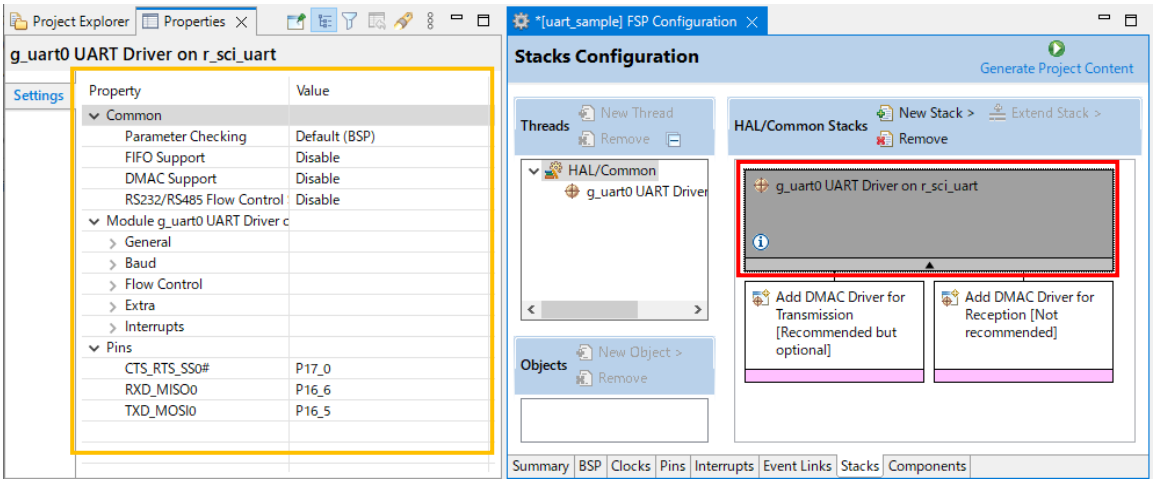


©2023 Alpha Project Co., Ltd.

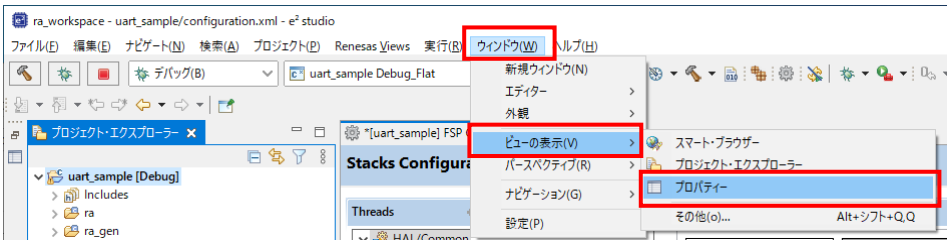
⑧　［Stacks］タブを開きます。この画面で、HAL ドライバやフレームワークの追加や設定をします。
［Driver］-［Connectivity］-［UART Driver on r_sci_uart］を選択します。



⑨　UART のモジュールが追加され、下図のように、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。

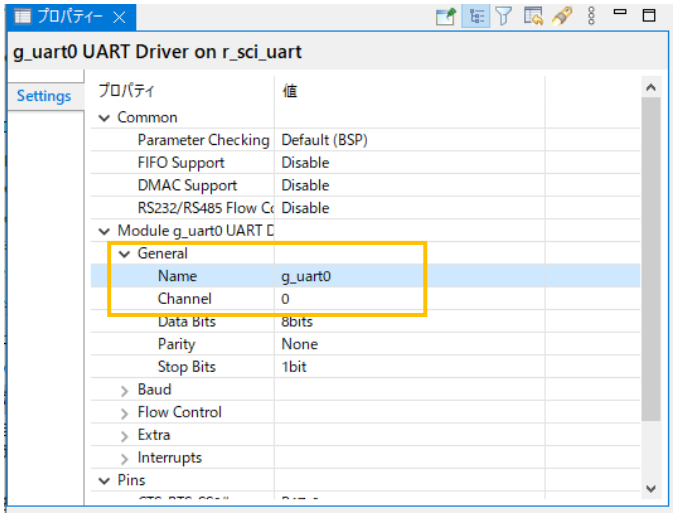


プロパティが見つからない場合は、ツールバーの［ウィンドウ］-［ビューの表示］-［プロパティー］で表示できます。



©2023 Alpha Project Co., Ltd.

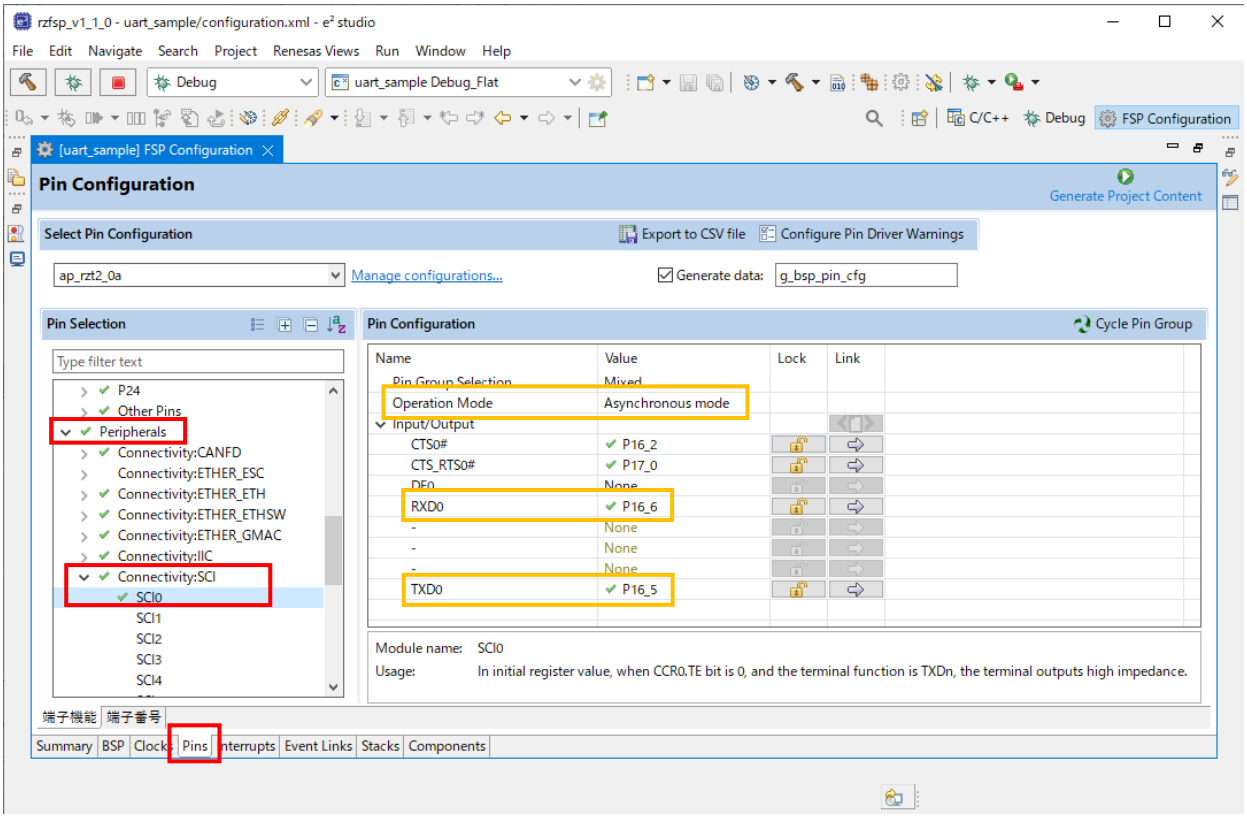
⑩　表示されたプロパティのうち［General］内の設定［Name］が［g_uart0］、［Channel］が［0］であることを確認してください。



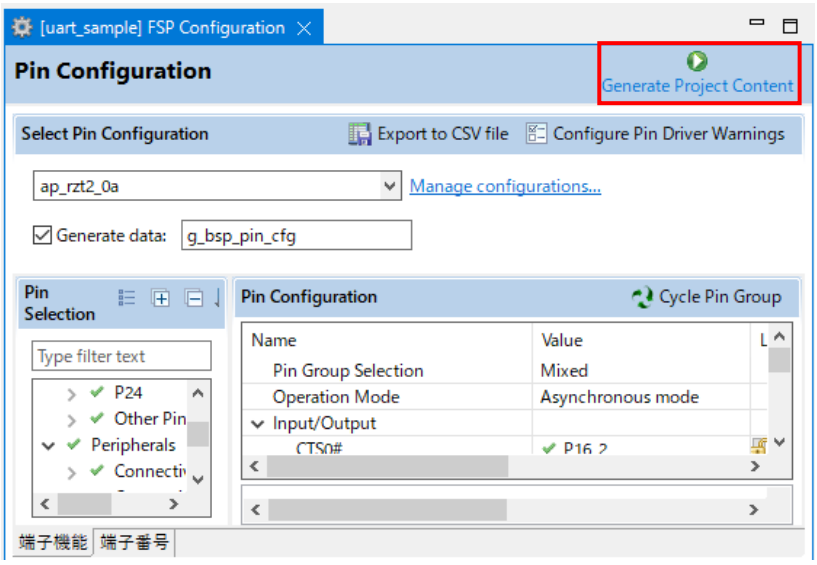
©2023 Alpha Project Co., Ltd.

⑪　［Pins］タブを開きます。この画面で、UART 端子の設定を確認します。
［Peripherals］-［Connectivity:SCI］-［SCI0］を選択した後、以下の設定になっていることを確認します。

- ［Operation Mode］→［Asynchronous mode］
- ［Input/Output］-［RXD0］→［P16_6］、［TXD0］→［P16_5］
  （AP-RZT2-1A の場合は［RXD0］→［P16_1］、［TXD0］→［P16_0］）



⑫　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。

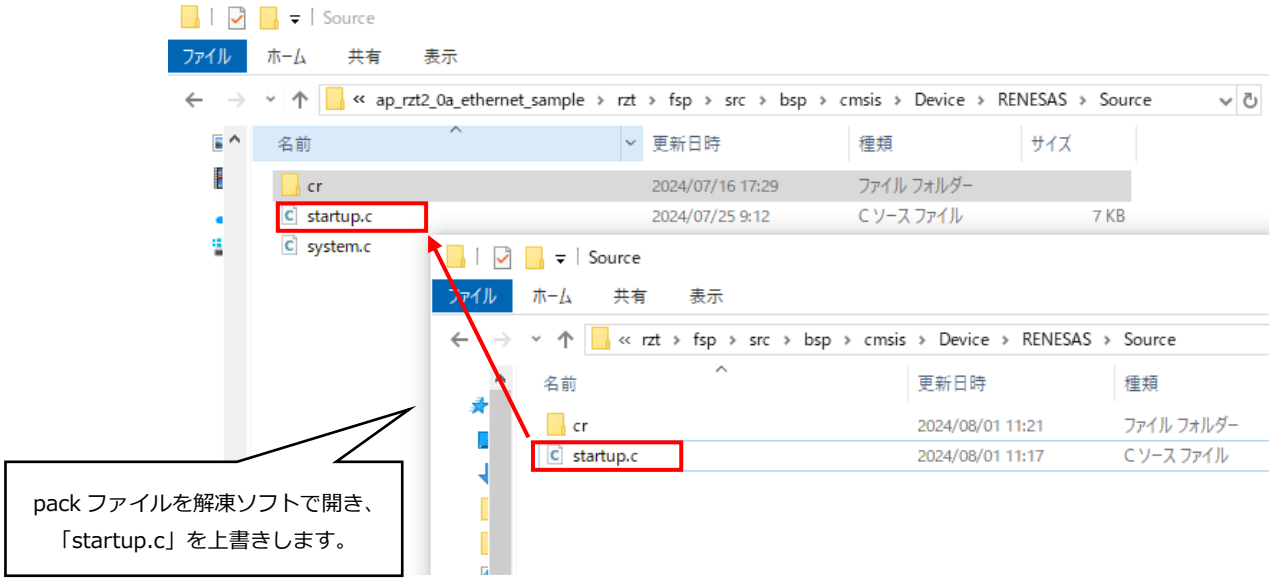


©2023 Alpha Project Co., Ltd.

⑬　プロジェクトに自動作成されたデータのうち、下記のファイルを、「3.1.1 インポート方法」でも使用した pack ファイル（本書の例では「AlphaProject.ap_rzt2_0a.2.1.0.pack」）の内容で上書きをします。

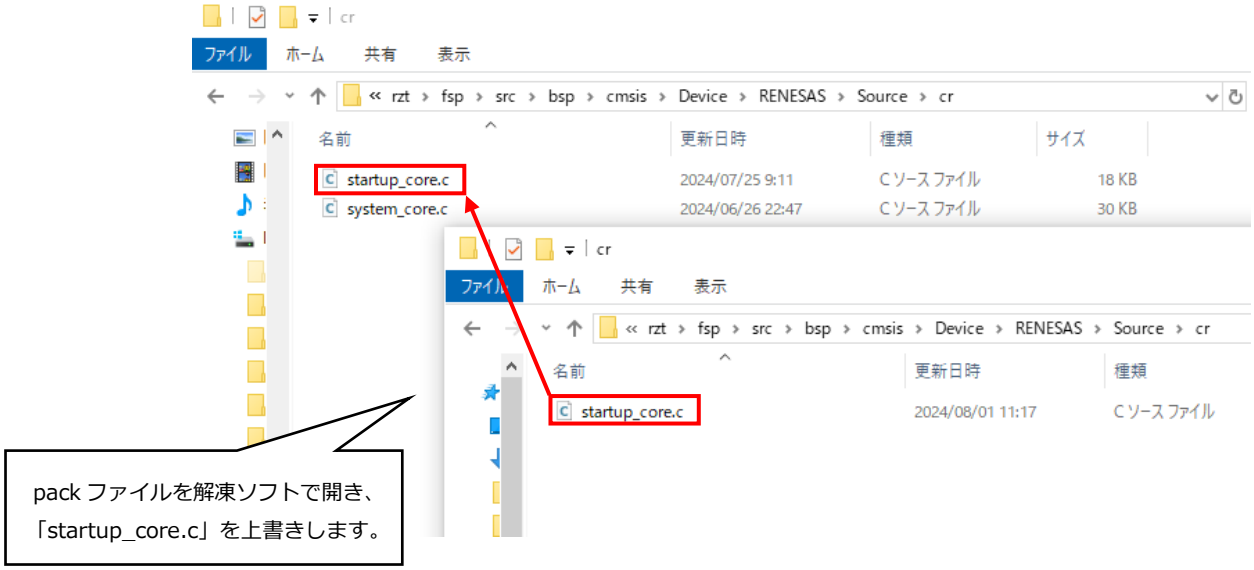
pack ファイルを解凍ソフトで開き、中に含まれる「startup.c」を、プロジェクト内に上書きコピーしてください。

対象ファイル：[プロジェクトフォルダ] ¥rzt ¥fsp ¥src ¥bsp ¥cmsis ¥Device ¥RENESAS ¥Source ¥startup.c
上書き用ファイル：[pack ファイル] ¥rzt ¥fsp ¥src ¥bsp ¥cmsis ¥Device ¥RENESAS ¥Source ¥startup.c



同様に pack ファイル内の「startup_core.c」を、プロジェクトの下記の場所に上書きコピーしてください。

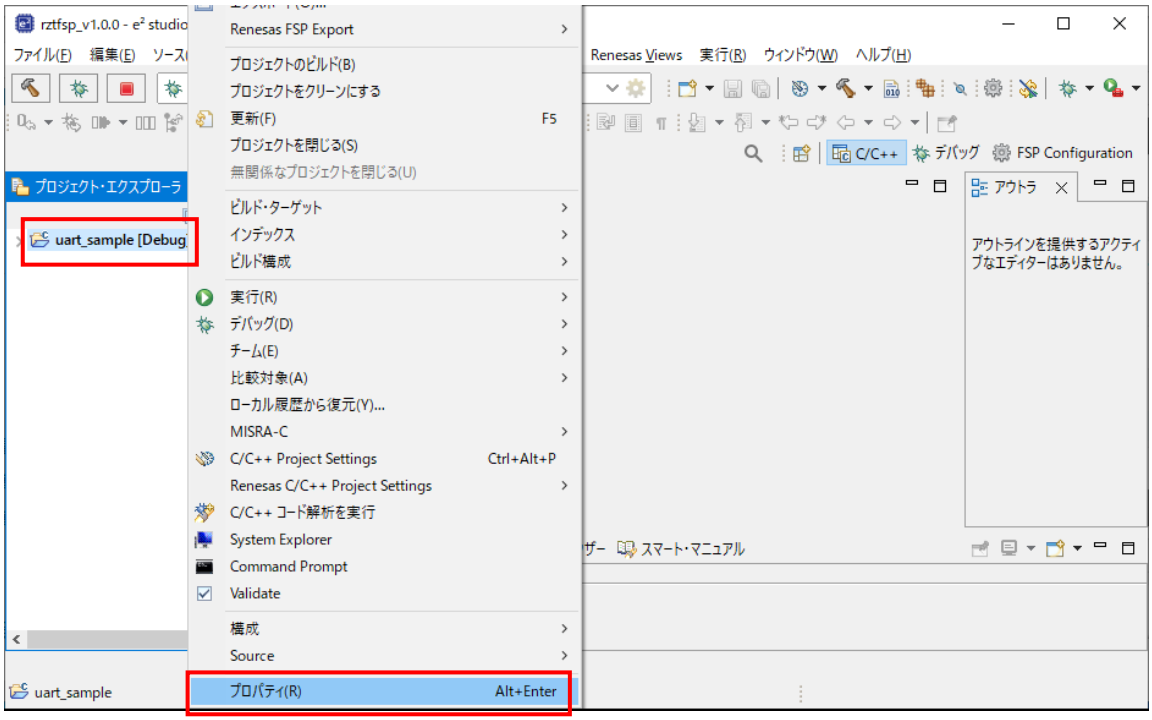
対象ファイル：[プロジェクトフォルダ] ¥rzt ¥fsp ¥src ¥bsp ¥cmsis ¥Device ¥RENESAS ¥Source ¥cr ¥startup_core.c
上書き用ファイル：[pack ファイル] ¥rzt ¥fsp ¥src ¥bsp ¥cmsis ¥Device ¥RENESAS ¥Source ¥cr ¥startup_core.c



©2023 Alpha Project Co., Ltd.

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。

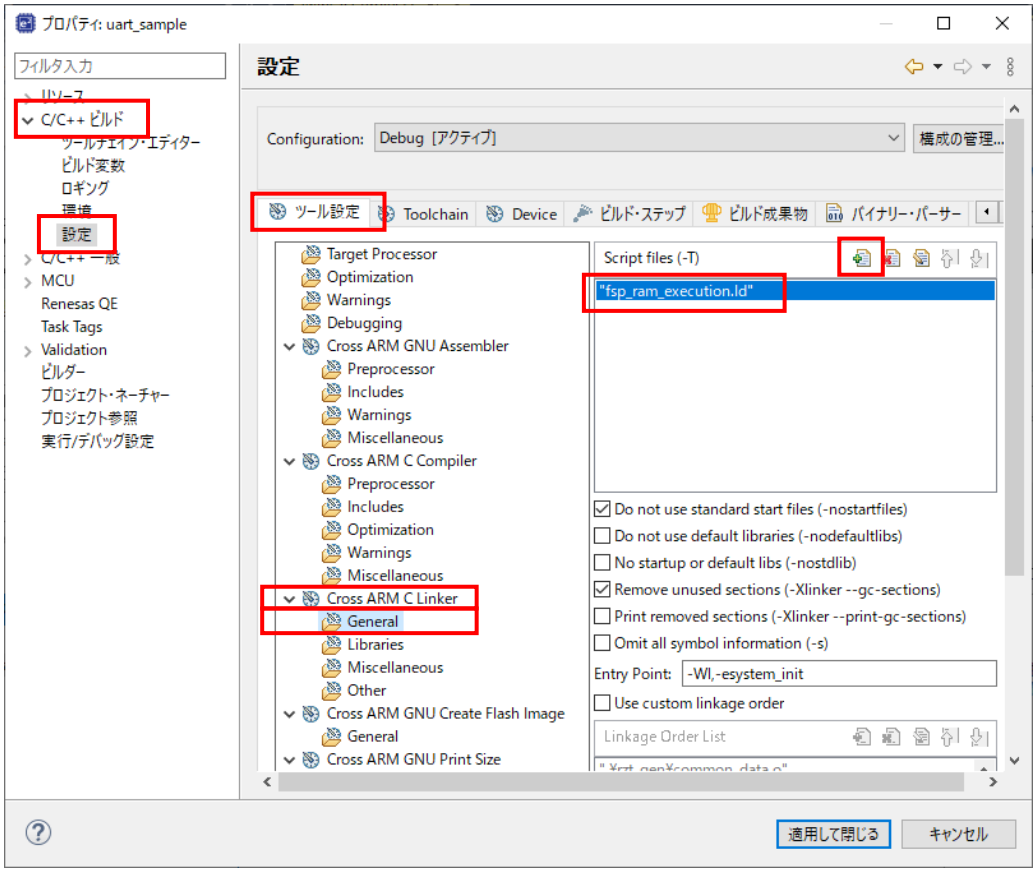
⑮　プロジェクトを選択して右クリックをし、「プロパティ」を選択します。



©2023 Alpha Project Co., Ltd.

⑯　プロパティウィンドウから「C/C++ビルド」＞「設定」を選択した上で、「ツール設定」＞「Cross ARM C Linker」＞「General」を選択します。
「Script files(-T)」横の追加ボタンから以下を設定します。

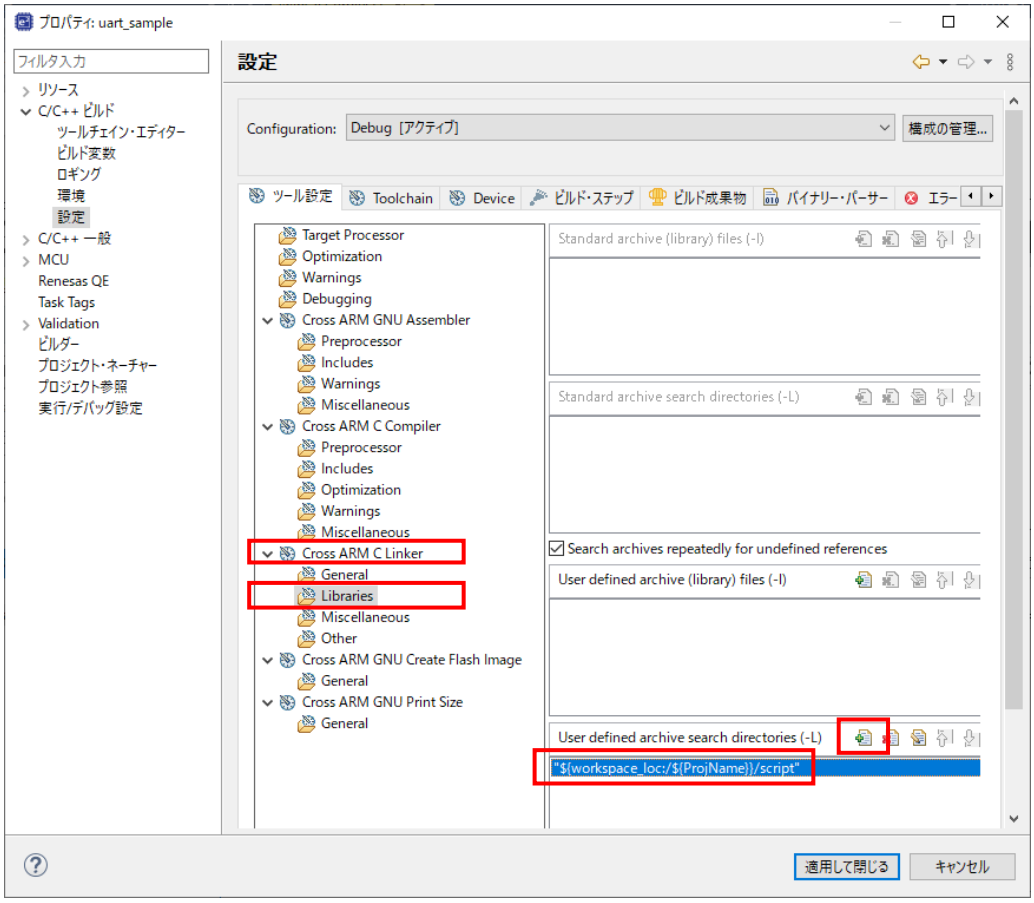
RAM 動作（Debug ビルド）の場合：	*"fsp_ram_execution.ld"*
ROM 動作（Release ビルド）の場合：	*"fsp_rom_execution.ld"*



©2023 Alpha Project Co., Ltd.

⑰　続けて「ツール設定」＞「Cross ARM C Linker」＞「Libraries」を選択します。
「User defined archive search directiories(-L)」横の追加ボタンから以下を設定します。

"${workspace_loc:/${ProjName}}/script"

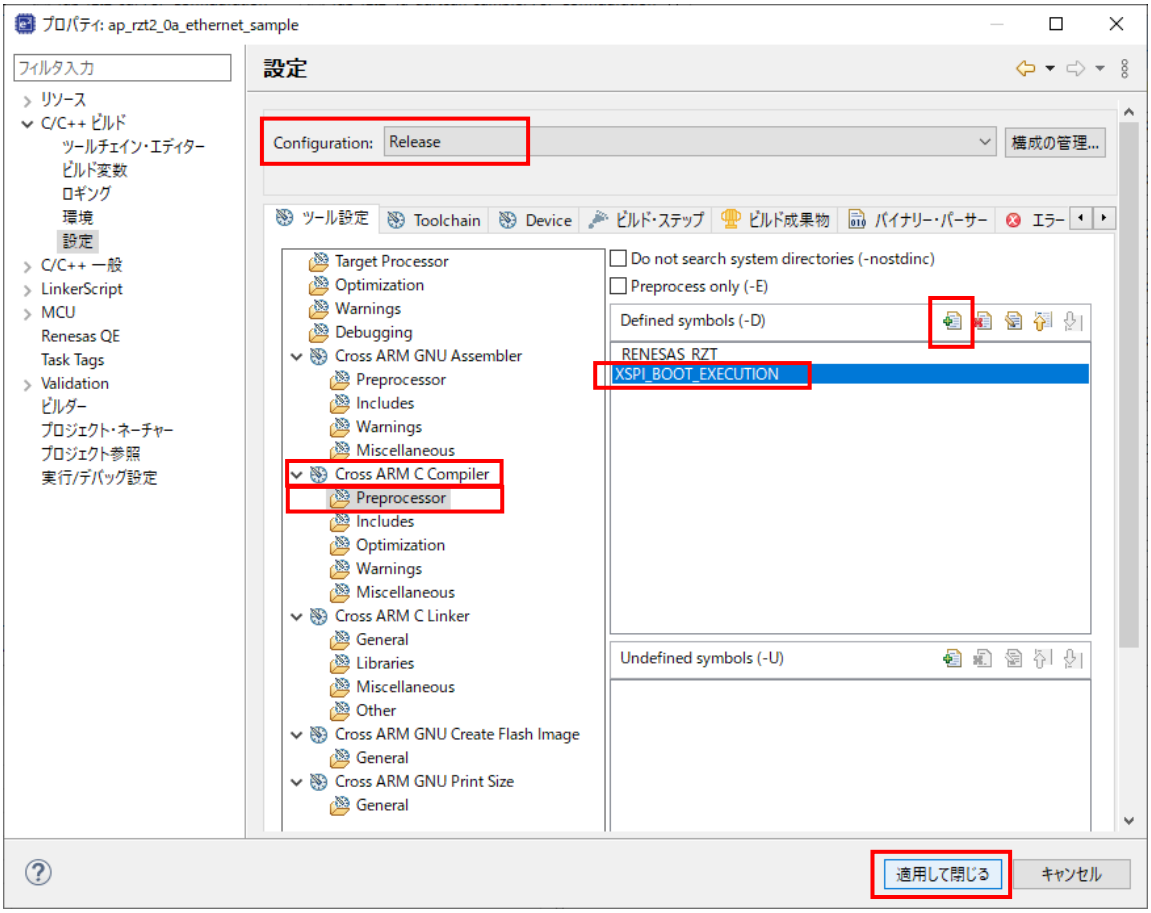


©2023 Alpha Project Co., Ltd.

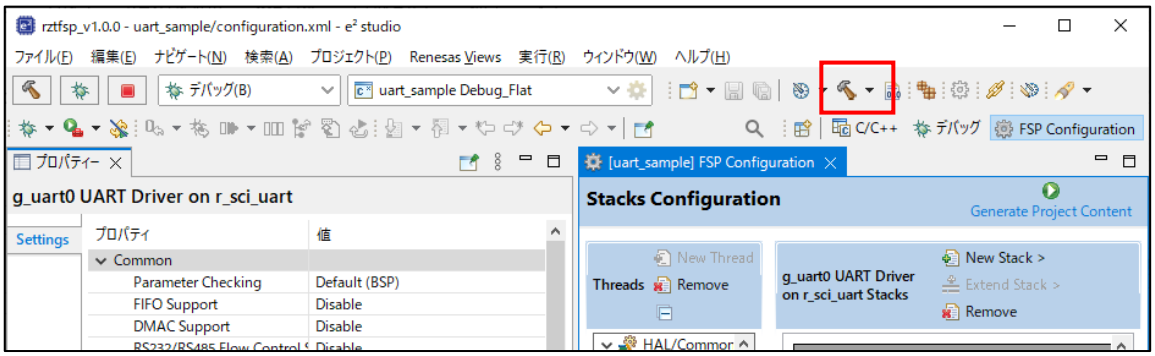
⑱　ROM 動作（「Release」ビルド）を行う場合は、続けて「ツール設定」＞「Cross ARM C Compiler」＞「Preprocessor」を選択します。
ビルド設定を「Release」に変更したうえで、「Defined symbols(-D)」横の追加ボタンから以下を設定します。

XSPI_BOOT_EXECUTION

すべての設定を行ったら「適用して閉じる」ボタンを選択します。
※RAM 動作（「Debug」ビルド）を行う場合は、「XSPI_BOOT_EXECUTION」を設定せずに「適用して閉じる」ボタンを選択してください。



⑲　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。



©2023 Alpha Project Co., Ltd.

5.　シリアル FlashROM への書き込み

AP-RZT2-0A/AP-RZT2-1A は、シリアル FlashROM ブートを用いることで、電源起動後に自動でプログラムをシリアル FlashROM から RAM へ読み出して実行することができます。
本章ではシリアル FlashROM ブートを行うために必要な、プログラムをシリアル FlashROM へ書き込む手順を説明いたします。

シリアル FlashROM への書き込みにはサンプルプログラムに付属する書き込み用バッチファイルを使用します。
書き込み用バッチファイルは、DualCore サンプルプログラム（AP-RZT2-0A のみ）用ファイルとそれ以外のサンプルプログラム用ファイルの 2 種類用意しています。
※DualCore サンプルプログラムは他サンプルプログラムと異なり、CPU コア別に複数のユーザプログラムを操作する必要があるため、別途用意しております。

本書では後者の DualCore サンプルプログラム以外の共通の書き込み用バッチファイルを基に説明を行います。
DualCore サンプルプログラムの書き込み手順については「AN1651 AP-RZT2-0A (RZ/T2M CPU BOARD) DualCore サンプルプログラム解説」も併せてご参照ください。

注意事項

- シリアル FlashROM の書き込むデータ構成について

  書き込み用バッチファイルを使うことで、以下の表に従ってプログラムをシリアル FlashROM に書き込みます。
  ユーザにてプログラムの配置先などを変更した場合には適宜値を読み替えてご参考ください。

項目	設定値
SSBL プログラムのローダ用パラメータの書き込み先アドレス	0x60000000
SSBL プログラムの書き込み先アドレス	0x60000050
ユーザプログラムの書き込み先アドレス	0x6004D000
ユーザプログラムのローダ用パラメータの書き込み先アドレス	0x60007080
ユーザプログラムの展開先先頭アドレス、および、実行開始アドレス	0x00000000

- python スクリプトについて

  書き込み用バッチファイルはルネサス エレクトロニクス社が公開する「RZ/T2,RZ/N2 グループ デバイスセットアップ for フラッシュブート アプリケーションノート」の"2024 年 6 月 14 日"版データを参考に作成しています。
  データの版数が異なる場合、バッチファイル内のコマンドや手順が正常に動作しない場合がございます。
  サンプルプログラム付属のバッチファイルを用いてシリアル FlashROM への書き込みがうまくいかない場合、「RZ/T2M グループ デバイスセットアップ for フラッシュブート サンプルプログラム」の資料をご参照いただき、適宜修正してバッチファイルをお使いください。

©2023 Alpha Project Co., Ltd.

5.1　事前準備

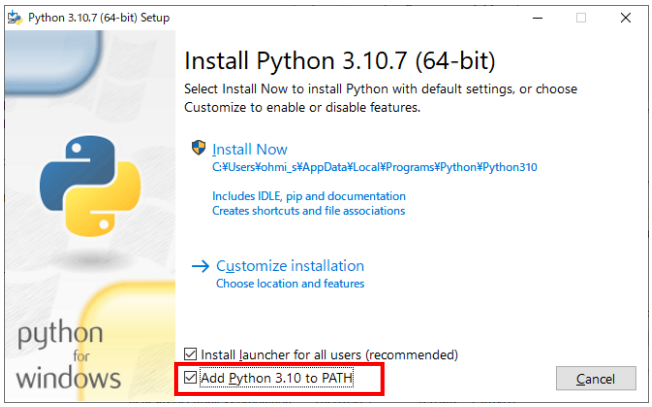
5.1.1　Python のインストール

シリアル FlashROM への書き込みには、PC 上で Python が動作する必要があります。

① Python の Web ページからインストーラをダウンロードします。

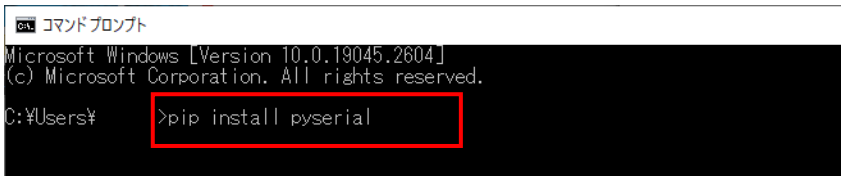
Python Web ページ：https://www.python.org/

② インストーラを起動して、PC に Python をインストールしてください。
なお、インストールの際には環境変数へ Python のパスを追加するようにしてください。
以降、環境変数に Python のパスを追加した状態で説明を行います。



③ Python インストール後、コマンドプロンプトを立ち上げ、以下のコマンドを実行します。

"pip install pyserial"



以上で Python の準備が終了です。

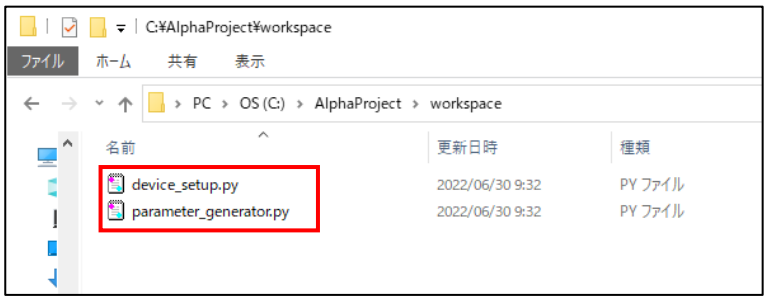
©2023 Alpha Project Co., Ltd.

5.1.2　書き込み時の PC⇔ボード間通信プログラムの用意

① ルネサス社の RZ/T2M CPU の Web ページから「RZ/T2M グループ デバイスセットアップ for フラッシュブート サンプルプログラム」のサンプルコードをダウンロードします。

RZ/T2M CPU Web ページ：
https://www.renesas.com/jp/ja/products/microcontrollers-microprocessors/rz-mpus/rzt2m-high-performance-multi-function-mpu-realizing-high-speed-processing-and-high-precision-control
※コンテンツをダウンロードする際にはルネサス社の My Renesas への登録が必要となります。

② ダウンロードしたコンテンツに含まれる「parameter_generator.py」「device_setup.py」を PC の任意の場所へ展開します。

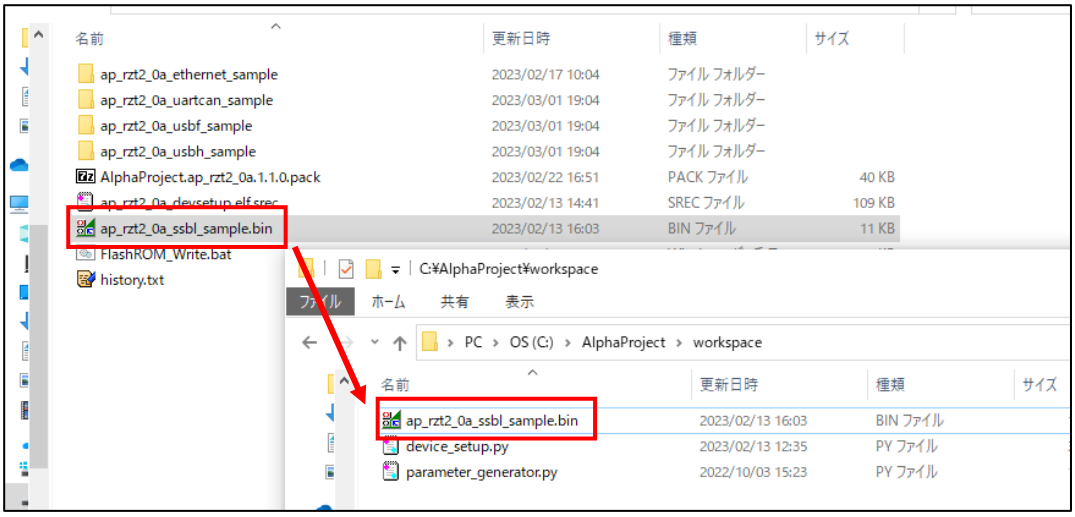


ここで展開した 2 つの python スクリプトファイルは書き込み用バッチファイルによって使用され、シリアル FlashROM の書き込み処理を行います。
また、ここで展開した先のフォルダを以降の説明では「Python ワークフォルダ」と記述します。

©2023 Alpha Project Co., Ltd.

5.1.3　書き込むプログラムの用意

① Python ワークフォルダへ、各サンプルプログラムに付属する「ap_rzt2_0a_ssbl_sample.bin」ファイルを移動します。



ここで移動した「ap_rzt2_0a_ssbl_sample.bin」は、シリアル FlashROM ブート時に最初に起動させる SSBL サンプルプログラムのデータです。
SSBL サンプルプログラムが起動すると、ユーザプログラム（ap_rzt2_0a_ethernet_sample など）をシリアル FlashROM から RAM へと展開し、その後ユーザプログラムの実行を開始します。

シリアル FlashROM ブート時の動作についてはルネサス社の「RZ/T2M グループ ユーザーズマニュアル ハードウェア編」内「3.5.1 ブート機能」をご参照ください。
文中のローダプログラムが SSBL サンプルプログラムに該当します。

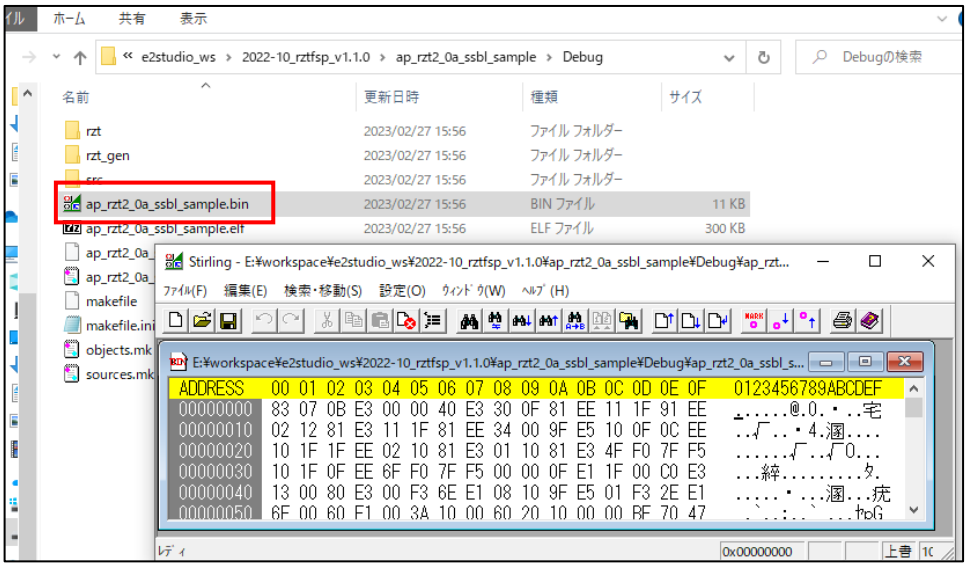
CPU のブート機能を使ってユーザプログラムを実行するのではなく、一旦 SSBL サンプルプログラムを動作させて、ユーザプログラムをシリアル FlashROM から RAM へのプログラムデータの展開・実行する手順を踏むことにより、ユーザプログラムは CPU のブート機能の制限に関係なく、任意のプログラムを作成することができます。
CPU のブート機能の制限については「RZ/T2M グループ ユーザーズマニュアル ハードウェア編」内「3.5.3 ローダプログラム」に記載のローダプログラムの条件をご参照ください。

なお、「ap_rzt2_0a_ssbl_sample.bin」は SSBL サンプルプログラムを Debug ビルドで出力したデータから一部変更を加えています。
もし SSBL サンプルプログラムをユーザ自身でカスタマイズした場合は、以下のファイル変更を行ってください。

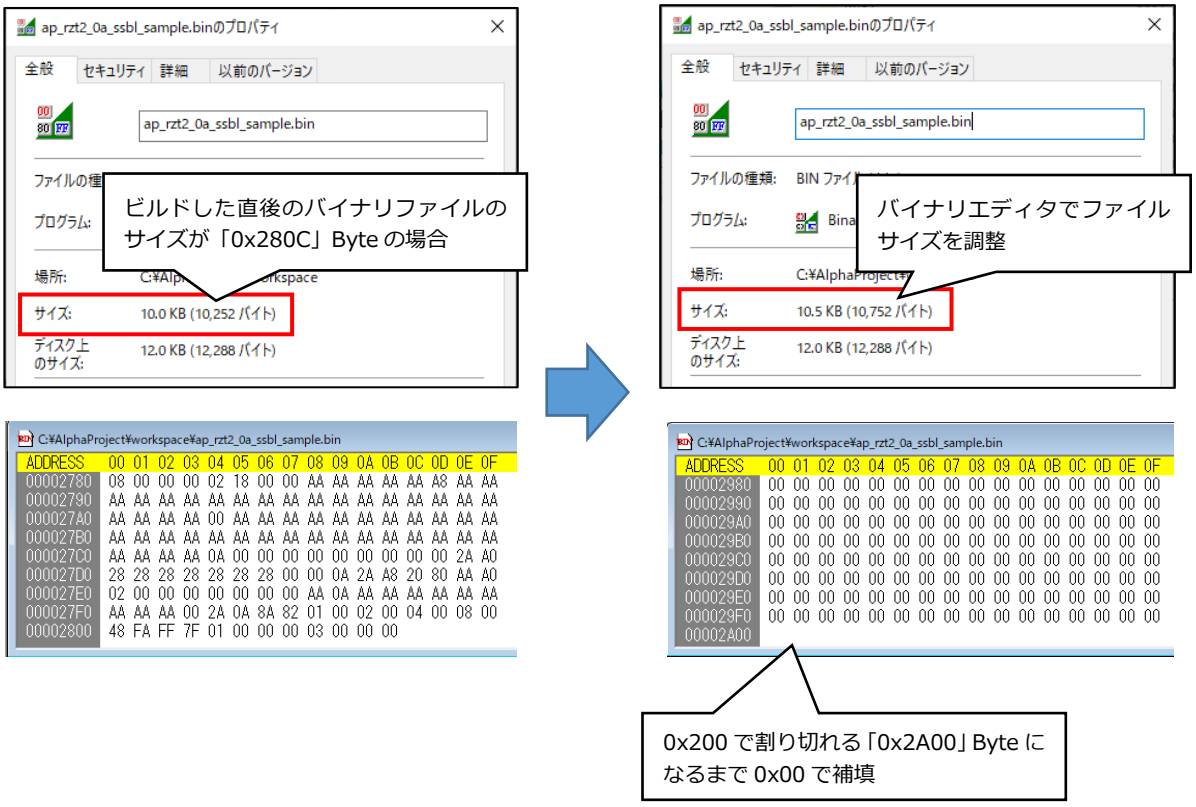
©2023 Alpha Project Co., Ltd.

(1)　SSBL サンプルプログラムの Debug フォルダ内にあるバイナリファイルをバイナリエディタで開きます。

SSBL プログラム：　ap_rzt2_0a_ssbl_sample¥Debug¥ap_rzt2_0a_ssbl_sample.bin
※SSBL サンプルプログラムの Debug ビルドで生成できるバイナリファイルを使用します。



(2)　SSBL プログラム（ap_rzt2_0a_ssbl_sample.bin）のデータサイズが 0x200 単位になるようプログラム末尾に「0x00」などのダミーデータを追加します。その後、上書き保存してください。

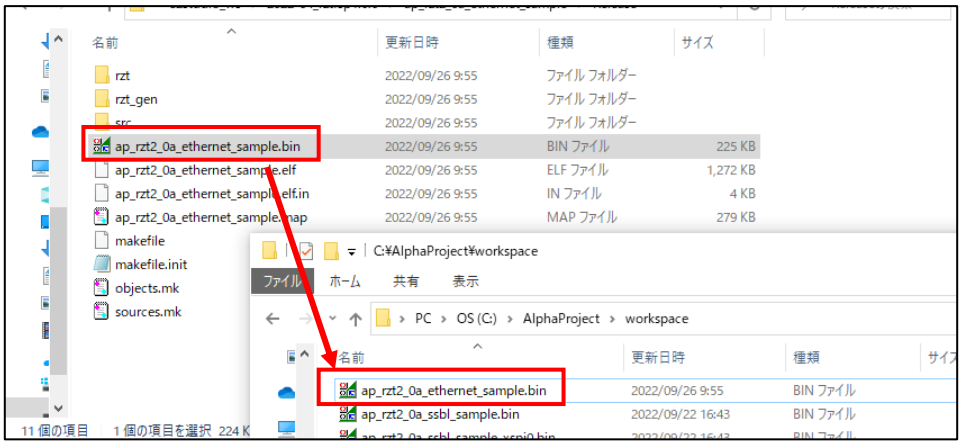


各サンプルプログラムに付属する「ap_rzt2_0a_ssbl_sample.bin」は、SSBL サンプルプログラムの Debug フォルダ内にある「ap_rzt2_0a_ssbl_sample.bin」から上記の変更を加えたものです。

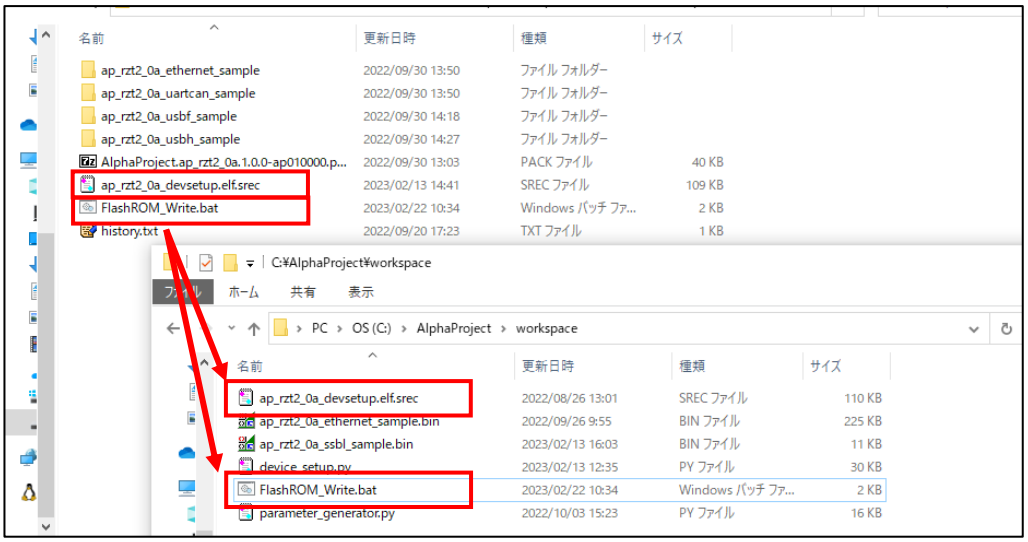
©2023 Alpha Project Co., Ltd.

② Python ワークフォルダへ、ユーザプロジェクト（例：ap_rzt2_0a_ethernet_sample）で生成した以下のプログラムを移動します。

ユーザプログラム（例）：　ap_rzt2_0a_ethernet_sample¥Release¥ap_rzt2_0a_ssbl_sample.bin
※サンプルプログラムの場合は Release ビルドで生成できるバイナリファイルを使用します。



③ Python ワークフォルダへ、サンプルプログラム付属の「ap_rzt2_0a_devsetup.elf.srec」※と「FlashROM_Write.bat」を移動します。
※AP-RZT2-1A の場合は「ap_rzt2_1a_devsetup.elf.srec」です。



©2023 Alpha Project Co., Ltd.

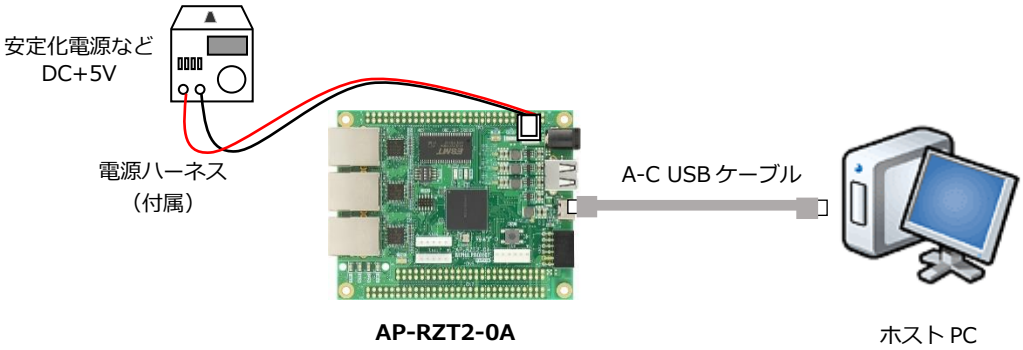
5.2　シリアル FlashROM への書き込み

書き込み用バッチファイルを使用して、CPU ボードのシリアル FlashROM にプログラムを書き込みます。

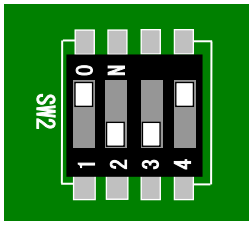
5.2.1　ボード設定

シリアル FlashROM への書き込み方法として USB 通信を使う方法と SCI 通信を使う方法の 2 種類があります。
サンプルプログラムに付属する書き込み用バッチファイルはデフォルトで USB 通信を使って書き込むように設定されています。
ご使用の環境に合わせて以下のようにボードの接続準備を行い、必要に応じてバッチファイルを修正してください。

- USB 通信で書き込む場合

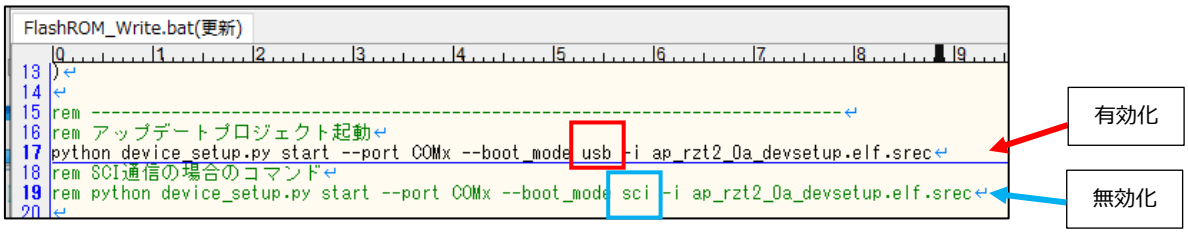


SW2 設定



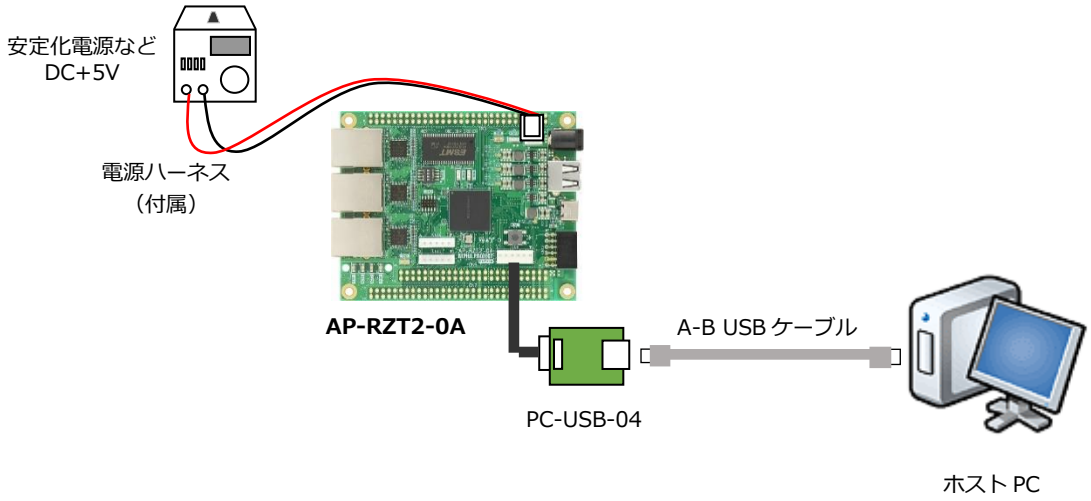
<SW2 設定>	
ブートモード	: USB ブート
JTAG Hash モード	: 使用する（不問）

※バッチファイル内の「python device_setup.py start ～」のコマンドは「usb」を含むコマンド行を有効化し、
「sci」を含むコマンド行は無効化してください。

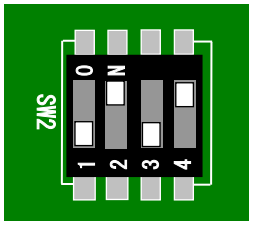


©2023 Alpha Project Co., Ltd.

● SCI 通信で書き込む場合

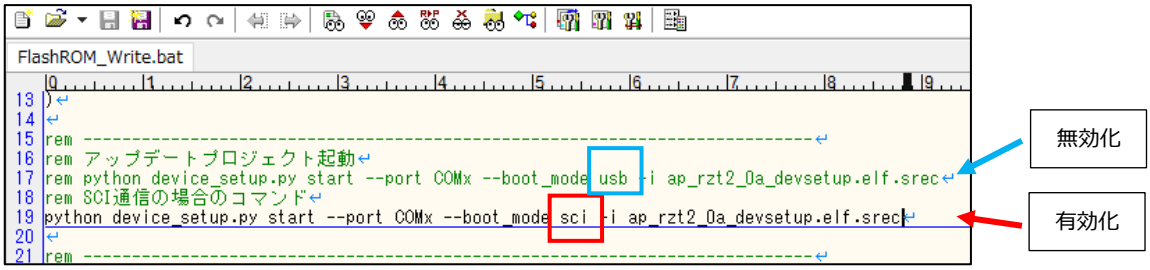


SW2 設定



<SW2 設定>	
ブートモード	: SCI ブート
JTAG Hash モード	: 使用する（不問）

※バッチファイル内の「python device_setup.py start ～」のコマンド行は「sci」を含むコマンド行を有効化し、「usb」を含むコマンド行は無効化してください。



©2023 Alpha Project Co., Ltd.

5.2.2　書き込み手順

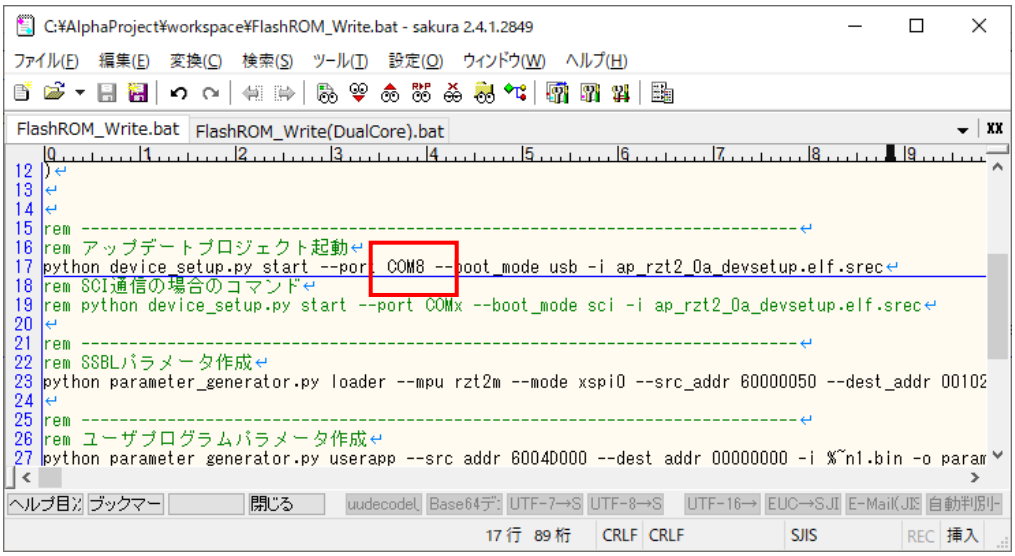
① SCI ブート、あるいは、USB ブートモードに設定した CPU ボードに電源を入れます。

② デバイスマネージャーを使用し、PC とボードを結ぶ COM ポートの番号を確認してください。

※使用する COM ポートの取り違えにご注意ください。
※COM ポートが現れない場合は、ボードのスイッチ設定や USB ケーブルの接続を見直してください。

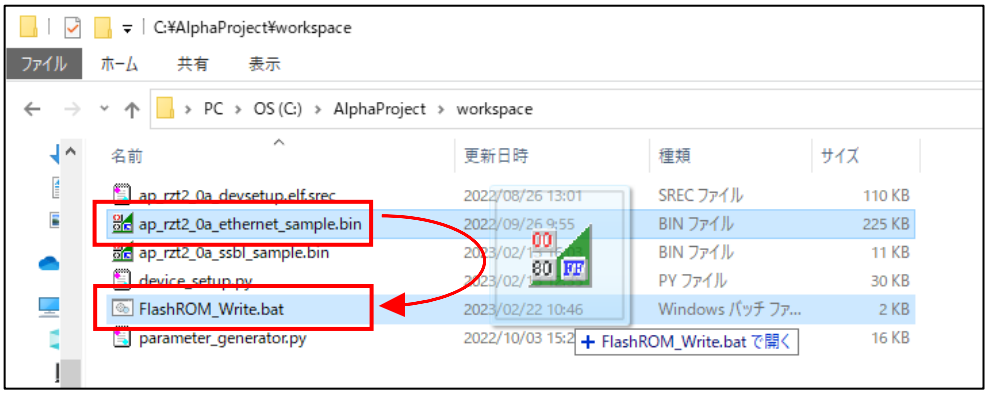
③ 書き込み用バッチファイル「FlashROM_Write.bat」をテキストエディタで開きます。

④ 「python device_setup.py start～」の行中の COM 番号を②で確認した COM ポート番号に変更してください。
変更後、ファイルを保存して閉じます。



©2023 Alpha Project Co., Ltd.

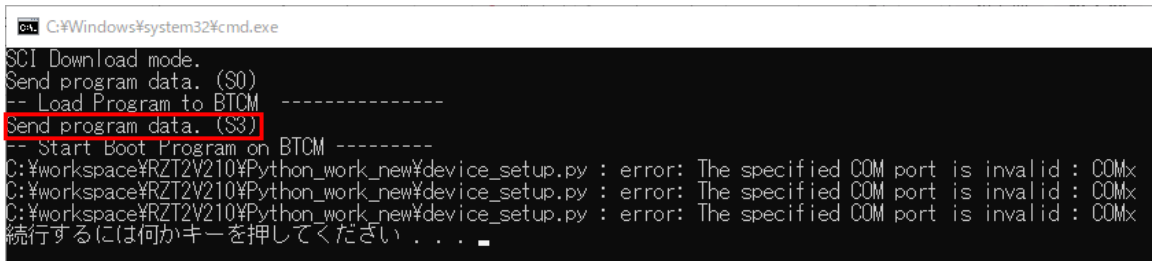
⑤　書き込むユーザプログラムを書き込み用バッチファイルに対してドラッグ＆ドロップをします。



⑥　書き込み用バッチファイルが起動して次のような表示になることを確認します。
エラーメッセージが表示されていますが「Send program data. (S3)」までが表示されれば問題ありません。

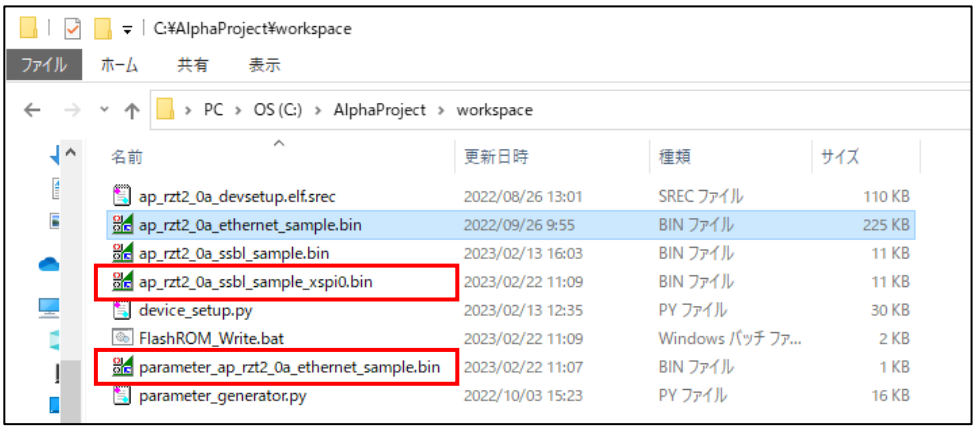
- USB 通信の場合

- SCI 通信の場合



©2023 Alpha Project Co., Ltd.

また、バッチファイルを実行すると python ワークフォルダに「ap_rzt2_0a_ssbl_sample_xspi0.bin」や「parameter_<ユーザプログラム>.bin」のファイルが新規作成されます。



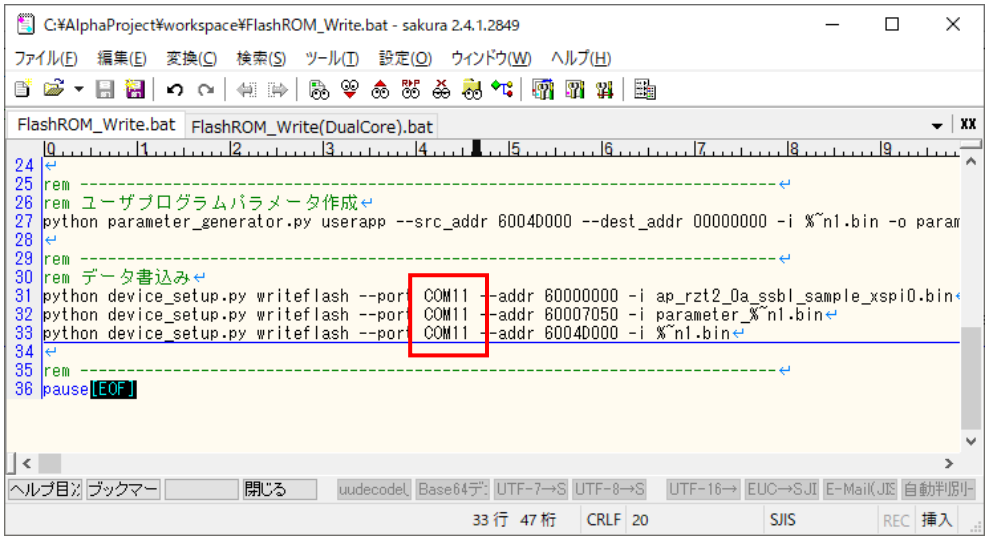
メッセージ「Program send completed」が表示されない場合、バッチファイル内のコマンド「python device_setup.py start ～」が正常に動作していません。
また、ファイル「ap_rzt2_0a_ssbl_sample_xspi0.bin」などが作成されない場合、バッチファイル内のコマンド「python parameter_generator.py ～」が正常に動作していません。

各コマンドが正常に動作しない場合は、python スクリプトの入手元であるルネサス社の「RZ/T2M グループ デバイスセットアップ for フラッシュブート サンプルプログラム」の資料をご確認いただき、各 python スクリプトの仕様に従ってバッチファイル内のコマンドを修正してください。

⑦　CPU ボードの電源を落とさずに、再度デバイスマネージャーを使用し、PC とボードを結ぶ COM ポートの番号を確認してください。
※SCI 通信の場合は COM ポートに変わりはないはずですが、USB 通信の場合 COM ポートが変わることがあります。

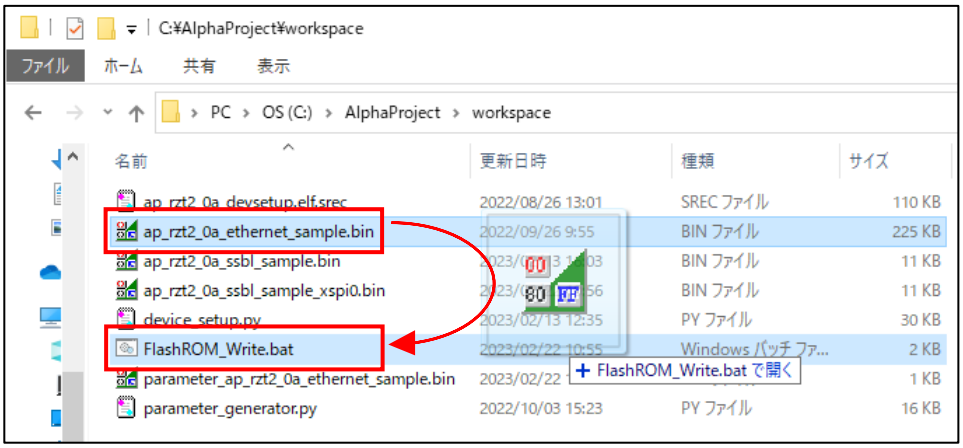
©2023 Alpha Project Co., Ltd.

⑧　再度「FlashROM_Write.bat」をテキストエディタで開き、「python device_setup.py writeflash～」の行中の COM 番号を⑦で確認した COM ポート番号に変更してください。このとき、④で変更した「python device_setup.py start～」の行中の COM ポート番号は変更しないでください。
変更後、ファイルを保存して閉じます。



⑨　CPU ボードの電源を切り、再度電源を入れて再起動します。

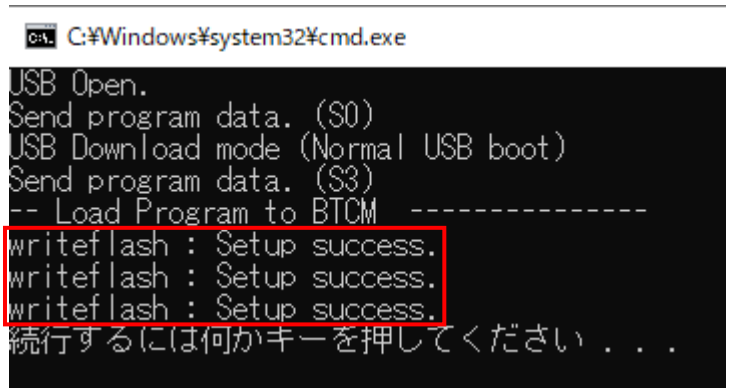
⑩　再度書き込むユーザプログラムを書き込み用バッチファイルに対してドラッグ＆ドロップをします。



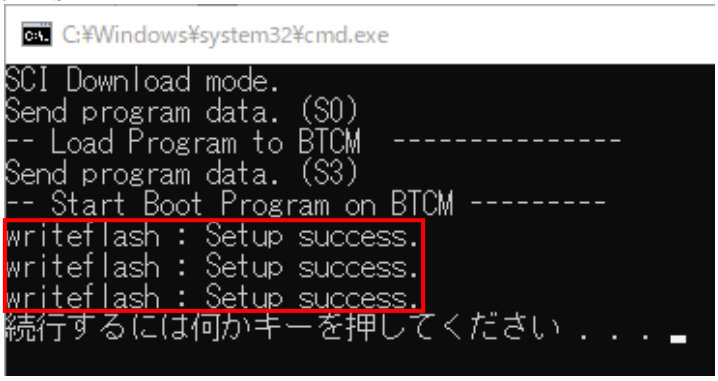
©2023 Alpha Project Co., Ltd.

⑪　書き込み用バッチファイルが起動して次のような表示になることを確認します。
「writeflash：Setup success.」が 3 行表示されればシリアル FlashROM への書き込みは成功です。

- USB 通信の場合



- SCI 通信の場合



以上でシリアル FlashROM への書き込みは終了です。

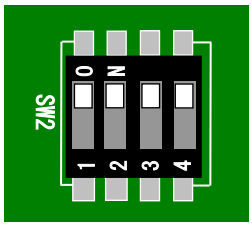
©2023 Alpha Project Co., Ltd.

5.3　シリアル FlashROM ブートの実行

「5.2　シリアル FlashROM の書き込み」を行った後、以下のスイッチ設定を行うことで、シリアル FlashROM ブートを行うことができます。

ボードに電源を投入し、プログラムの動作についてご確認ください。

SW2 設定



<SW2 設定>	
ブートモード	：シリアル FlashROM ブート
JTAG Hash モード	：使用する（不問）

©2023 Alpha Project Co., Ltd.

改定履歴

版数	更新日	改定内容
1 版	2023/03/20	新規作成
1.1 版	2023/04/24	FSP のバージョン 1.2.0 に対応
1.2 版	2023/10/02	住所の更新
1.3 版	2024/09/02	FSP のバージョン 2.1.0 に対応

©2023 Alpha Project Co., Ltd.

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

・RZ ファミリおよび RZ/T2M 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。
・Pmod は、Digilent.Inc.の商標です。
・J Link は、SEGGER Microcontroller GmbH & Co. KG の登録商標もしくは商標です。

・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

©2023 Alpha Project Co., Ltd.