

# AP-RA8P-0A

## サンプルプログラム解説

1.0版 2025年12月23日

|  |           |
|--|-----------|
| <b>1. 概要</b> .....                     | <b>2</b>  |
| 1.1 概要.....                            | 2         |
| 1.2 接続概要.....                          | 3         |
| 1.3 本サンプルプログラムについて.....                | 6         |
| 1.4 開発環境について.....                      | 6         |
| 1.5 ワークスペースについて.....                   | 7         |
| <b>2. サンプルプログラムの構成</b> .....           | <b>8</b>  |
| 2.1 フォルダ構成.....                        | 8         |
| 2.2 ファイルの構成.....                       | 9         |
| <b>3. AP-RA8P-0A サンプルプログラム</b> .....   | <b>12</b> |
| 3.1 RTT Viewer 使用方法.....               | 12        |
| 3.2 動作説明.....                          | 15        |
| 3.2.1 CAN サンプルプログラムの動作説明.....          | 15        |
| 3.2.2 Ethernet サンプルプログラムの動作説明.....     | 16        |
| 3.2.3 UART/メモリサンプルプログラムの動作説明.....      | 20        |
| 3.2.4 SDHI サンプルプログラムの動作説明.....         | 19        |
| 3.2.5 USB ホストサンプルプログラムの動作説明.....       | 21        |
| 3.2.6 USB ファンクションサンプルプログラムの動作説明.....   | 22        |
| 3.3 e2 studio を用いたプロジェクトのビルド・デバッグ..... | 24        |
| 3.3.1 インポート方法.....                     | 24        |
| 3.3.2 ビルド方法.....                       | 29        |
| 3.3.3 デバッグ、ダウンロード方法.....               | 32        |

## 1. 概要

### 1.1 概要

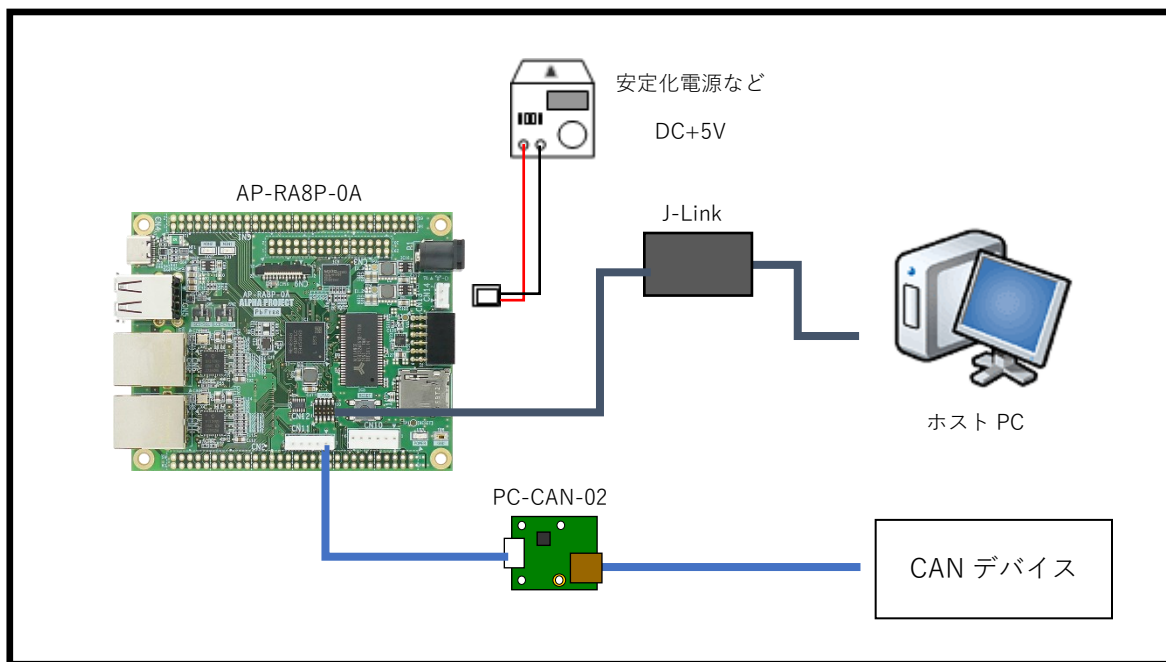
本アプリケーションノートでは、AP-RA8P-0A に付属するサンプルプログラムについて解説します。  
解説するサンプルプログラムは下記のものになります。

| デバイス       | 機能   |
|------------|--|
| AP-RA8P-0A | <ul style="list-style-type: none"><li>・ CAN 通信</li><li>・ ネットワーク通信</li><li>・ UART 通信</li><li>・ SDRAM、OSPI Flash へのリード・ライト</li><li>・ microSD カード読み書き</li><li>・ USB ホスト メモリ読み書き</li><li>・ USB ファンクション 仮想 COM 通信</li></ul> |

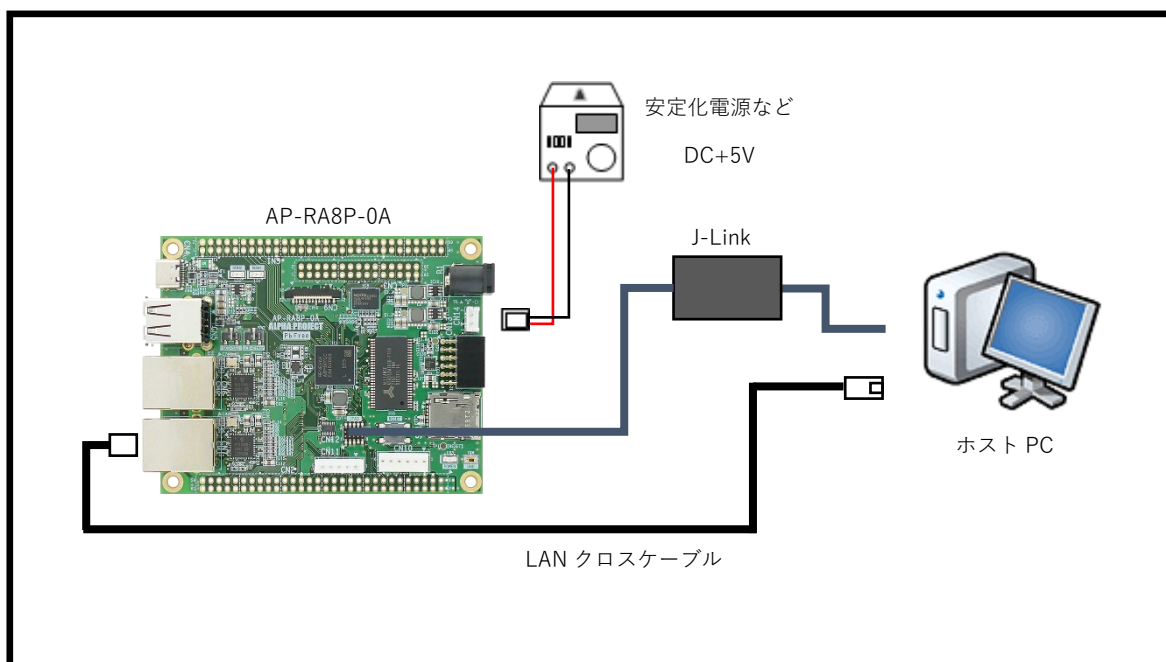
## 1.2 接続概要

本サンプルプログラムの動作を確認する上で必要な CPU ボードの接続例を以下に示します。  
 詳細な接続に関しては後述の「3.2 動作説明」を参照してください。

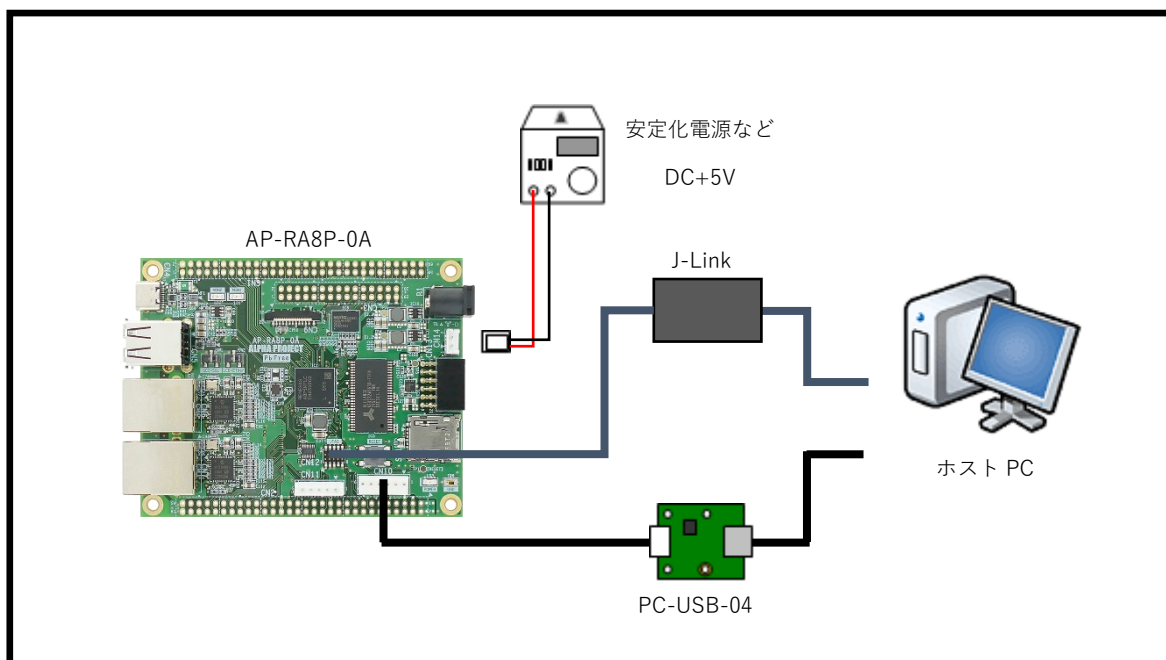
・CAN サンプルプログラム動作時の接続例



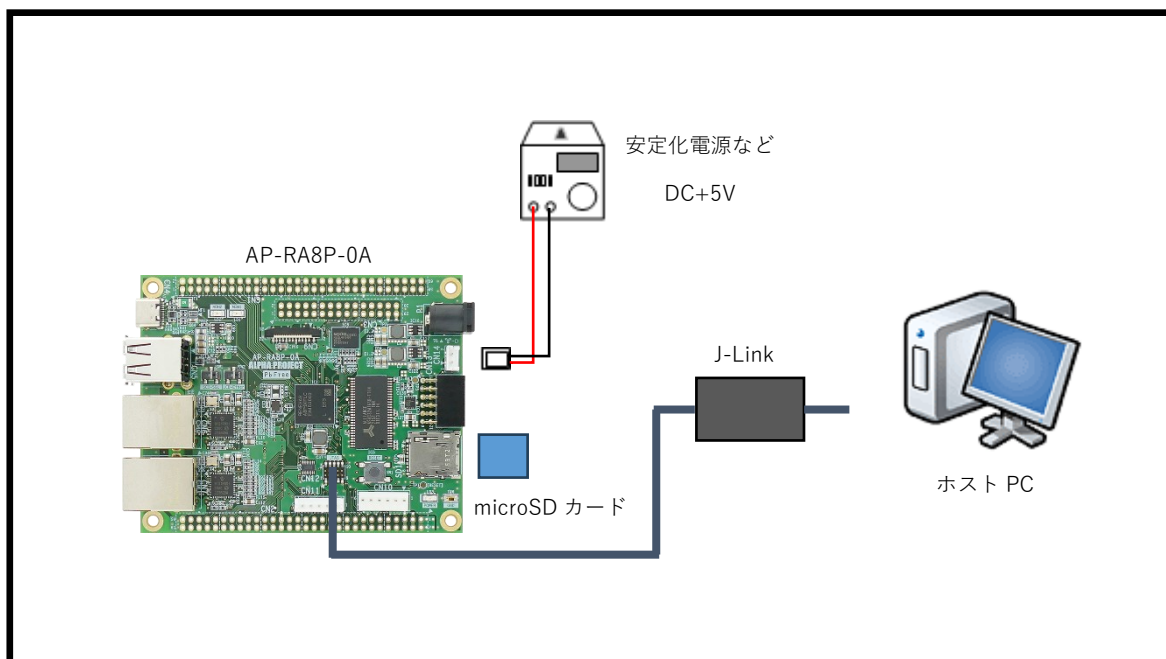
・Ethernet サンプルプログラム動作時の接続例



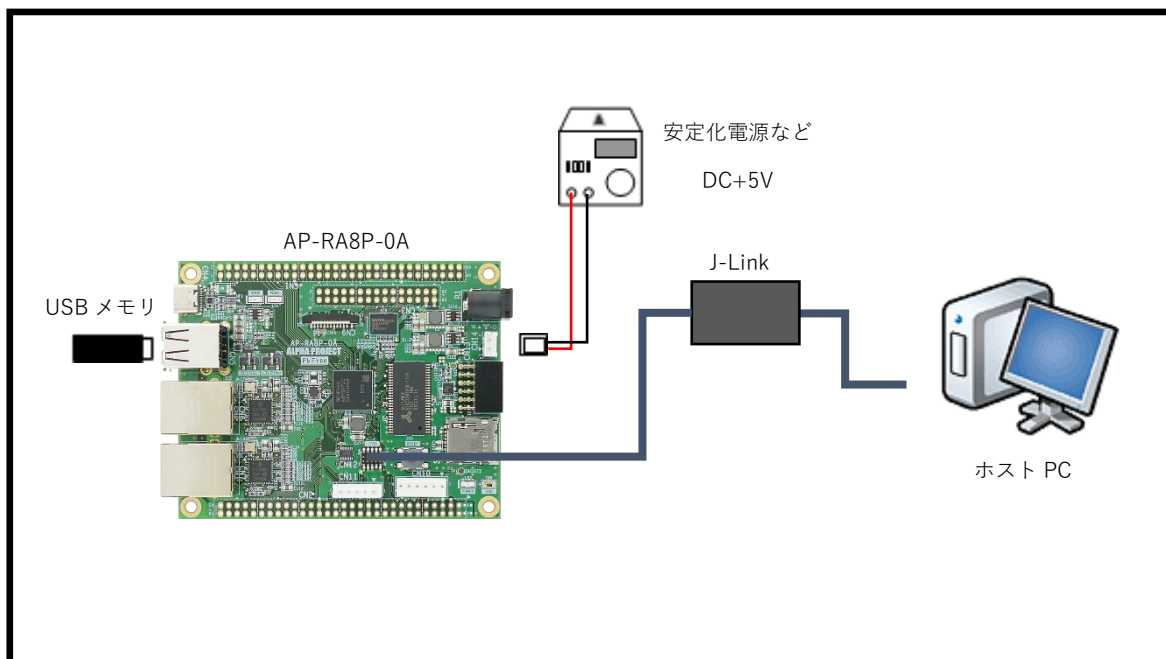
・ UART/メモリサンプルプログラム動作時の接続例



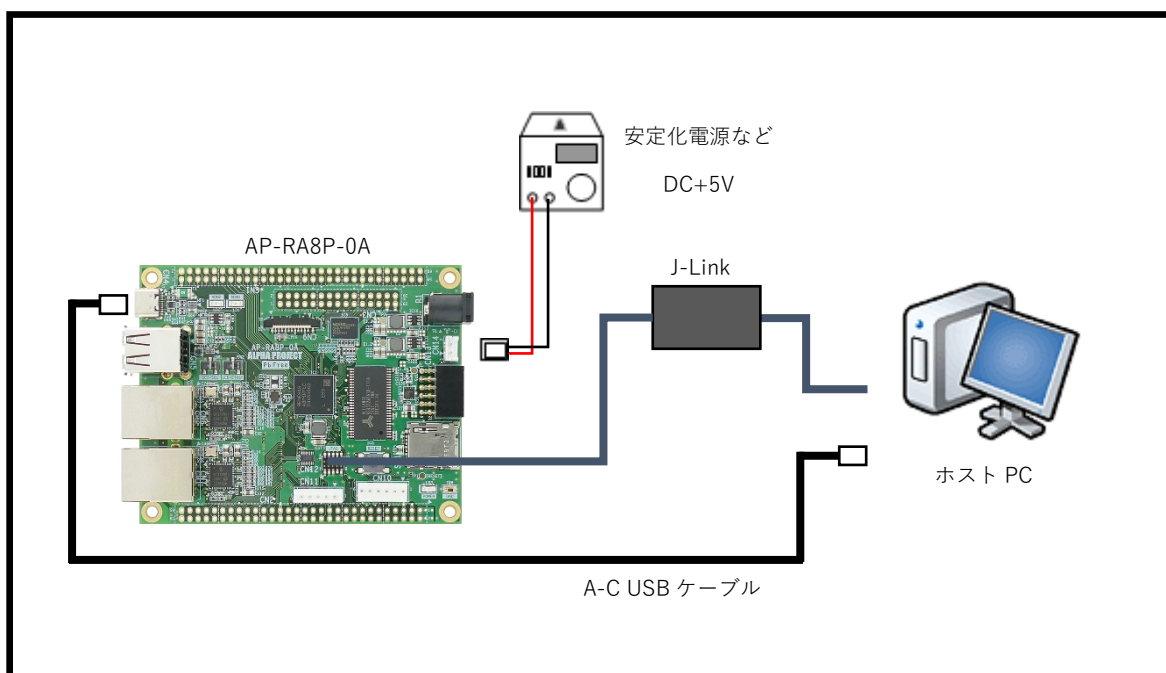
・ SDHI サンプルプログラム動作時の接続例



・USB ホストサンプルプログラム動作時の接続例



・USB ファンクションサンプルプログラム動作時の接続例



### 1.3 本サンプルプログラムについて

本サンプルプログラムおよび本書含むアプリケーションノートは、弊社 Web サイトのボード紹介ページで公開されています。

株式会社アルファプロジェクト

AP-RA8P-0A 製品ページ <https://www.apnet.co.jp/product/ra/ap-ra8p-0a.html>

### 1.4 開発環境について

本サンプルプログラムは統合開発環境「e2 studio」と「Flexible Software Package（以下、FSP）」を用いて開発されています。

本サンプルプログラムに対応する開発環境、FSP、コンパイラ、デバッガのバージョンは次の通りです。

| ソフトウェア                          | バージョン    | 備考                               |
|---------------------------------|----------|----------------------------------|
| e2 studio                       | V2025-10 | –                                |
| FSP                             | 6.2.0    | –                                |
| LLVM Embedded Toolchain for Arm | 18.1.3   | –                                |
| RTTViewer                       | V8.84    | Segger Microcontroller Sysytem 社 |

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

※AP-RA8P-0A と J-Link を直接接続することはできません。

AP-RA8P-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。

## 1.5 ワークスペースについて

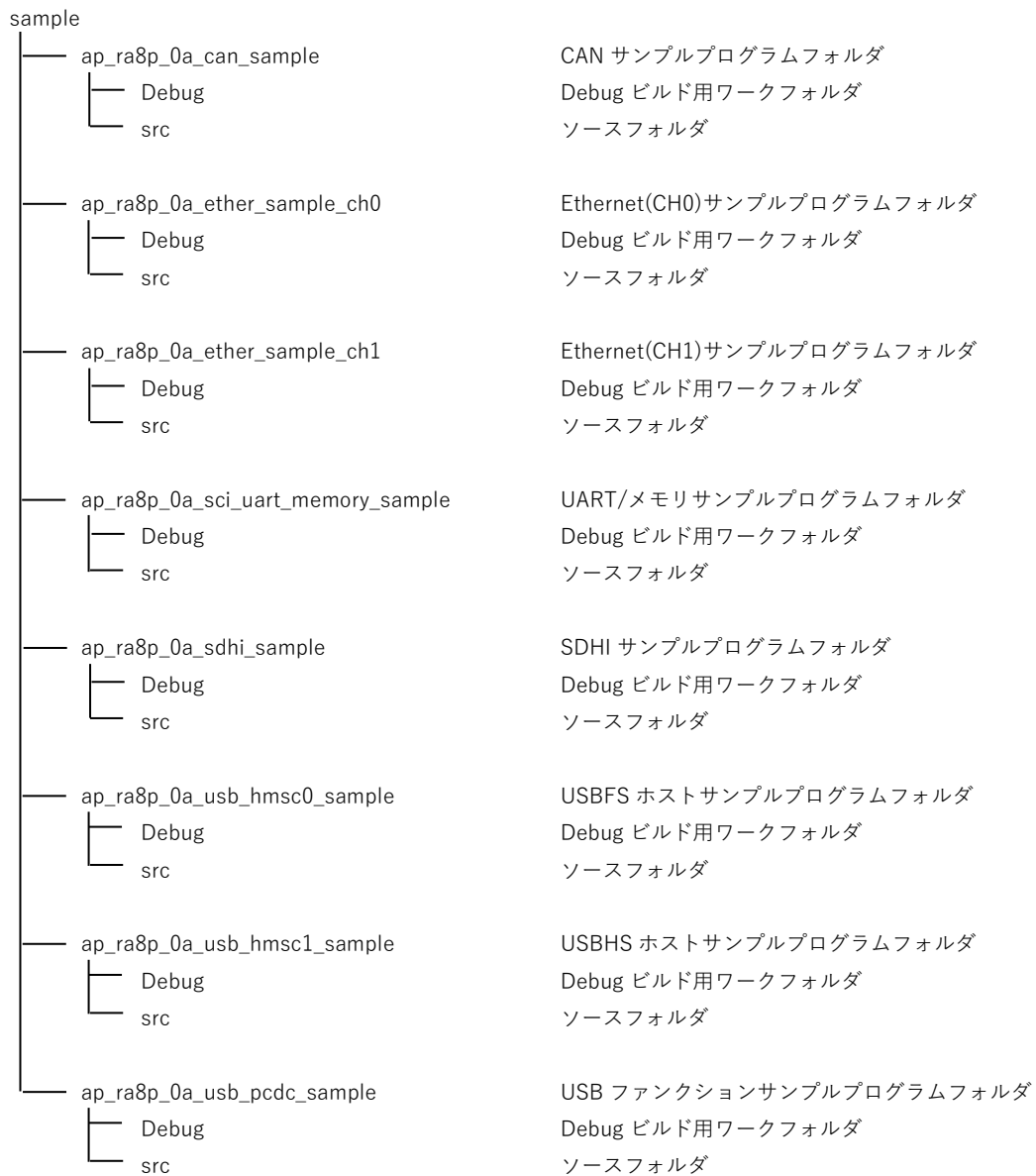
本サンプルプログラムのプロジェクトファイルは次のフォルダに格納されています。

| サンプルプログラム            | フォルダ                                      |
|----------------------|---|
| CAN サンプルプログラム        | ¥sample¥ap_ra8p_0a_can_sample             |
| Ethernet サンプルプログラム   | ¥sample¥ap_ra8p_0a_ether_sample           |
| UART/メモリサンプルプログラム    | ¥sample¥ap_ra8p_0a_sci_uart_memory_sample |
| SDHI サンプルプログラム       | ¥sample¥ap_ra8p_0a_sdhi_sample            |
| USB ホストサンプルプログラム     | ¥sample¥ap_ra8p_0a_usb_hmsc_sample        |
| USB ファンクションサンプルプログラム | ¥sample¥ap_ra8p_0a_usb_pcdc_sample        |

## 2. サンプルプログラムの構成

### 2.1 フォルダ構成

サンプルプログラムは下記のようなフォルダ構成になっています。



## 2.2 ファイルの構成

サンプルプログラムは以下のファイルで構成されています。

本節では、サンプルプログラムの作成にあたって追加したファイルについてのみ記述し、自動生成ファイルなどに関しては説明を省略します。

### ・共通ファイル

<¥sample¥CustomBSP フォルダ内>

AlphaProject.ap\_ra8p\_0a.6.2.0.pack      …      AP-RA8P-0A パックファイル

### ・CAN サンプルプログラム

<¥sample¥ap\_ra8p\_0a\_can\_sample フォルダ内>

.cproject      …      CPROJECT ファイル  
 .project      …      PROJECT ファイル  
 ap\_ra8p\_0a\_can\_sample      …      AP-RA8P-0A CAN サンプルプログラム  
 Debug\_Flat.launch      …      デバッグおよびランタイム設定ファイル  
 configuration.xml      …      FSP コンフィギュレータファイル

<¥sample¥ap\_ra8p\_0a\_can\_sample¥src フォルダ内>

hal\_entry.c      …      hal\_entry 関数ソースファイル  
 hal\_warmstart.c      …      hal\_warmstart 関数ソースファイル

### ・Ethernet サンプルプログラム

Ethernet はチャンネル別にサンプルプログラムを用意していますが、構成は同じため、CH0 のファイル構成を記述します。CH1 サンプルプログラムを扱う際は、チャンネル番号を読み替えてご参照ください。

<¥sample¥ap\_ra8p\_0a\_ether\_sample\_ch0 フォルダ内>

.cproject      …      CPROJECT ファイル  
 .gitignore      …      GITIGNORE ファイル  
 .project      …      PROJECT ファイル  
 ap\_ra8p\_0a.pincfg      …      AP-RA8P-0A ピンコンフィグファイル  
 ap\_ra8p\_0a\_ether\_sample\_ch0      …      AP-RA8P-0A Ethernet (CH0) サンプルプログラム  
 Debug\_Flat.launch      …      デバッグおよびランタイム設定ファイル  
 configuration.xml      …      FSP コンフィギュレータファイル

<¥sample¥ap\_ra8p\_0a\_ether\_sample\_ch0¥src フォルダ内>

SEgger\_RTT      …      RTTViewer ソース格納フォルダ  
 common\_utils.h      …      共通ヘッダファイル  
 hal\_entry.cpp      …      hal\_entry 関数ソースファイル  
 hal\_warmstart.cpp      …      hal\_warmstart 関数ソースファイル  
 net\_thread\_entry.cpp      …      net\_thread\_entry 関数ソースファイル  
 usr\_app.h      …      ユーザーアプリケーションヘッダファイル

### ・UART/メモリサンプルプログラム

<¥sample¥ap\_ra8p\_0a\_sci\_uart\_memory\_sample フォルダ内>

|                                   |     |                              |
|-----------------------------------|-----|------------------------------|
| .cproject                         | ... | CPROJECT ファイル                |
| .project                          | ... | PROJECT ファイル                 |
| ap_ra8p_0a_sci_uart_memory_sample | ... | AP-RA8P-0A UART/メモリサンプルプログラム |
| Debug_Flat.launch                 | ... | デバッグおよびランタイム設定ファイル           |
| configuration.xml                 | ... | FSP コンフィギュレータファイル            |

<¥sample¥ap\_ra8p\_0a\_sci\_uart\_memory\_sample ¥src フォルダ内>

|                       |     |                               |
|-----------------------|-----|-------------------------------|
| SEGGER_RTT            | ... | RTTViewer ソース格納フォルダ           |
| common_utils.h        | ... | 共通ヘッダファイル                     |
| hal_warmstart.c       | ... | hal_warmstart 関数ソースファイル       |
| memory_thread_entry.c | ... | memory_thread_entry 関数ソースファイル |
| ospi_b_commands.c     | ... | OSPI コマンドソースファイル              |
| ospi_b_commands.h     | ... | OSPI コマンドヘッダファイル              |
| ospi_flash.c          | ... | OSPI Flash 処理ソースファイル          |
| ospi_flash.h          | ... | OSPI Flash 処理ヘッダファイル          |
| timer_pwm.c           | ... | PWM タイマ処理ソースファイル              |
| timer_pwm.h           | ... | PWM タイマ処理ヘッダファイル              |
| uart_ep.c             | ... | UART 通信ソースファイル                |
| uart_ep.h             | ... | UART 通信ヘッダファイル                |
| uart_thread_entry.c   | ... | uart_thread_entry 関数ソースファイル   |

### ・SDHI サンプルプログラム

<¥sample¥ap\_ra8p\_0a\_sdhi\_sample フォルダ内>

|                        |     |                           |
|------------------------|-----|---------------------------|
| .cproject              | ... | CPROJECT ファイル             |
| .project               | ... | PROJECT ファイル              |
| ap_ra8p_0a_sdhi_sample | ... | AP-RA8P-0A SDHI サンプルプログラム |
| Debug_Flat.launch      | ... | デバッグおよびランタイム設定ファイル        |
| configuration.xml      | ... | FSP コンフィギュレータファイル         |

<¥sample¥ap\_ra8p\_0a\_sdhi\_sample ¥src フォルダ内>

|                     |     |                             |
|---------------------|-----|-----------------------------|
| SEGGER_RTT          | ... | RTTViewer ソース格納フォルダ         |
| common_utils.h      | ... | 共通ヘッダファイル                   |
| hal_entry.c         | ... | hal_entry 関数ソースファイル         |
| hal_warmstart.c     | ... | hal_warmstart 関数ソースファイル     |
| sdhi_ep.h           | ... | SDHI 情報ヘッダファイル              |
| sdhi_thread_entry.c | ... | sdhi_thread_entry 関数ソースファイル |

### ・USB ホストサンプルプログラム

USB HOST はチャンネル別にサンプルプログラムを用意していますが、構成は同じため、CH0 のファイル構成を記述します。CH1 のサンプルプログラムを扱う際は、チャンネル番号を読み替えてご参照ください。

<¥sample¥ap\_ra8p\_0a\_usb\_hmsc0\_sample フォルダ内>

|                             |     |                               |
|-----------------------------|-----|-------------------------------|
| .cproject                   | ... | CPROJECT ファイル                 |
| .project                    | ... | PROJECT ファイル                  |
| ap_ra8p_0a_usb_hmsc0_sample | ... | AP-RA8P-0A USBFS ホストサンプルプログラム |
| Debug_Flat.launch           | ... | デバッグおよびランタイム設定ファイル            |
| configuration.xml           | ... | FSP コンフィギュレータファイル             |

<¥sample¥ap\_ra8p\_0a\_usb\_hmsc0\_sample¥src フォルダ内>

|                         |     |                                 |
|-------------------------|-----|---------------------------------|
| SEGGER_RTT              | ... | RTTViewer ソース格納フォルダ             |
| common_utils.h          | ... | 共通ヘッダファイル                       |
| hal_entry.c             | ... | hal_entry 関数ソースファイル             |
| hal_warmstart.c         | ... | hal_warmstart 関数ソースファイル         |
| usb_hmsc_ep.h           | ... | USB ホスト情報ヘッダファイル                |
| usb_hmsc_thread_entry.c | ... | usb_hmsc_thread_entry 関数ソースファイル |

### ・USB ファンクションサンプルプログラム

<¥sample¥ap\_ra8p\_0a\_usb\_pcdc\_sample フォルダ内>

|                            |     |                                 |
|----------------------------|-----|---------------------------------|
| .cproject                  | ... | CPROJECT ファイル                   |
| .project                   | ... | PROJECT ファイル                    |
| ap_ra8p_0a_usb_pcdc_sample | ... | AP-RA8P-0A USB ファンクションサンプルプログラム |
| Debug_Flat.launch          | ... | デバッグおよびランタイム設定ファイル              |
| configuration.xml          | ... | FSP コンフィギュレータファイル               |

<¥sample¥ap\_ra8p\_0a\_usb\_pcdc\_sample¥src フォルダ内>

|                         |     |                         |
|-------------------------|-----|-------------------------|
| board_cfg.h             | ... | ボード情報ヘッダファイル            |
| common_init.h           | ... | 共通初期化ヘッダファイル            |
| hal_entry.c             | ... | hal_entry 関数ソースファイル     |
| hal_entry.h             | ... | hal_entry 関数ヘッダファイル     |
| hal_warmstart.c         | ... | hal_warmstart 関数ソースファイル |
| r_usb_pcdc_descriptor.c | ... | USB ファンクション情報ヘッダファイル    |

### 3. AP-RA8P-0A サンプルプログラム

#### 3.1 RTT Viewer 使用方法

サンプルプログラムは、SEGGER 社製ソフトウェア「RTT Viewer」を使用し、動作状況やプログラムの情報が表示される動作があります。

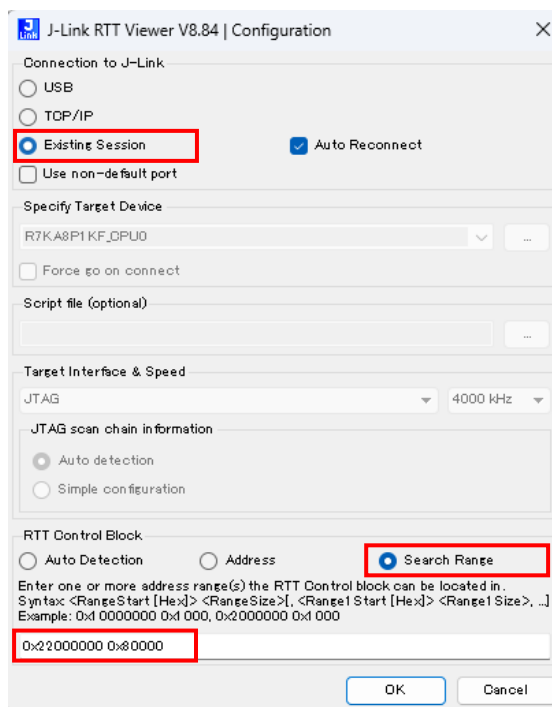
RTT Viewer は、J-Link Software and Documentation Pack に含まれており、下記の SEGGER 社 Web サイトより入手可能です。

SEGGER Microcontroller 社

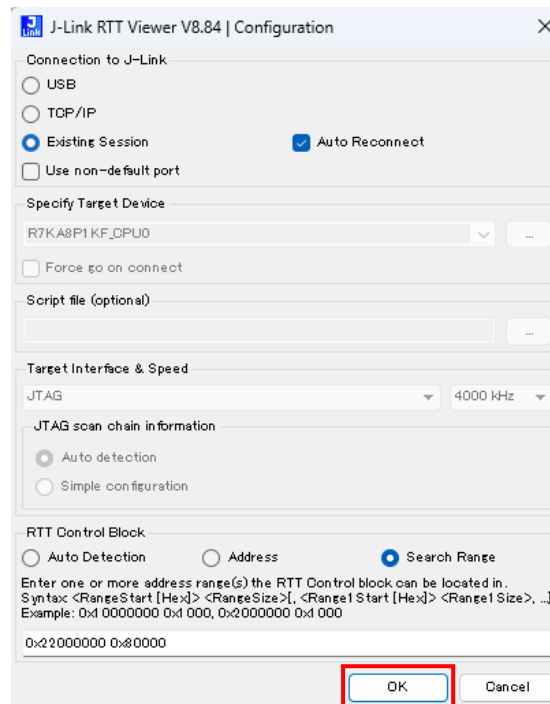
J-Link / J-Trace Downloads ページ <https://www.segger.com/downloads/jlink/>

RTT Viewer を使用した接続は、以下の手順に従って行ってください。

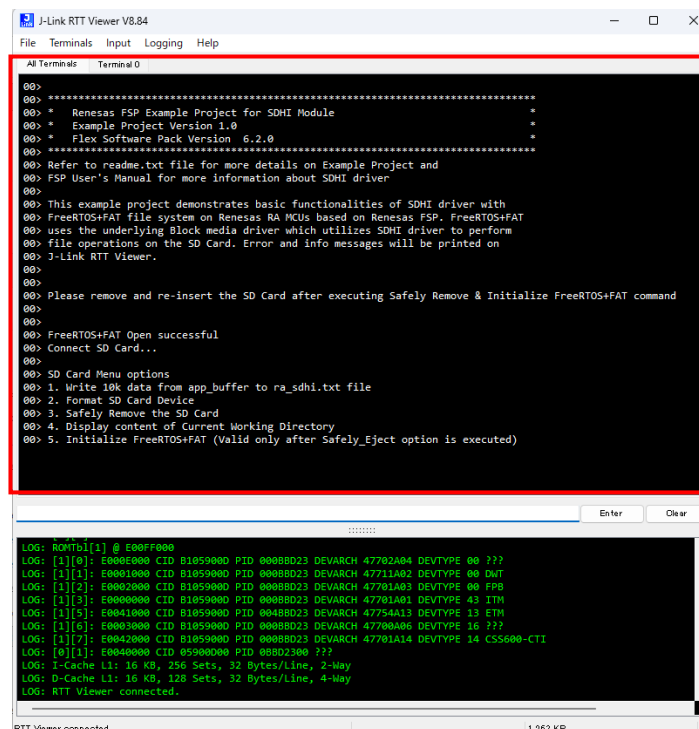
- ① CPU ボードとホスト PC を、J-Link デバッガを使用して接続します。
- ② ホスト PC にて、「JLinkRTTViewer.exe」を起動します。
- ③ Configuration ウィンドウが表示されますので、Connection to J-Link を「Existing Session」に設定し、RTT Control Block を「Search Range」 「0x22000000\_0x80000」に設定します。  
※.画面表示は、RTT Viewer のバージョンにより異なる可能性があります。  
以下の手順では、「V8.84」を使用した場合の手順を記載します。



- ④ CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ⑤ Configuration ウィンドウの[OK]ボタンを押下し、RTTViewer の接続を開始します。



- ⑥ 接続が完了しますと、CPU ボードからの出力内容が、[Terminal] に表示されます。





## 3.2 動作説明

### 3.2.1 CAN サンプルプログラムの動作説明

本サンプルプログラムでは AP-RA8P-0A とともに CAN1 を用いてエコーバック通信を行います。  
CAN の通信設定は、以下のように行われます。

| CPU ボードの設定 |               |                                 |
|------------|---------------|---------------------------------|
| ID         | 受信 Mailbox ID | 任意 (例 : B'00010100001 (0x0A1))  |
|            | 送信 Mailbox ID | 受信データの MailboxID と ID           |
| フォーマット     |               | スタンダードフォーマット、データフレーム、データ長 8 バイト |
| 通信速度       |               | 500kbps                         |

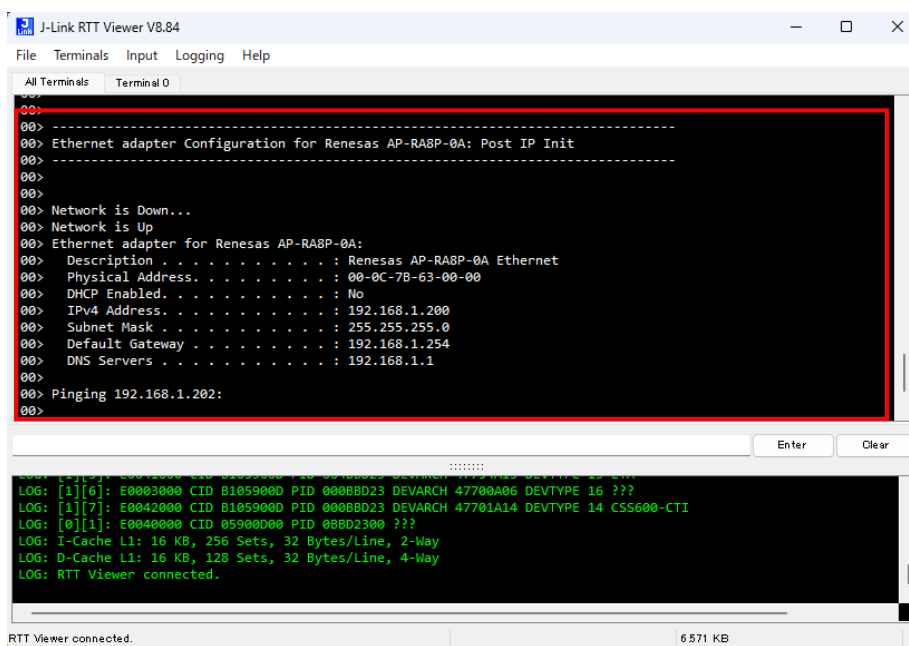
### 3.2.2 Ethernet サンプルプログラムの動作説明

本サンプルプログラムでは、CPU ボードから固定の IP アドレスに Ping 通信を行います。  
 ホスト PC を下記のネットワーク設定で動作させ、ボードへ接続した後、通信動作を行ってください。  
 Ethernet(CH0)サンプルプログラムは CN7 を、Ethernet(CH1)サンプルプログラムは CN6 を使用してください。  
 ネットワーク動作の確認は、以下の手順に従って行ってください。

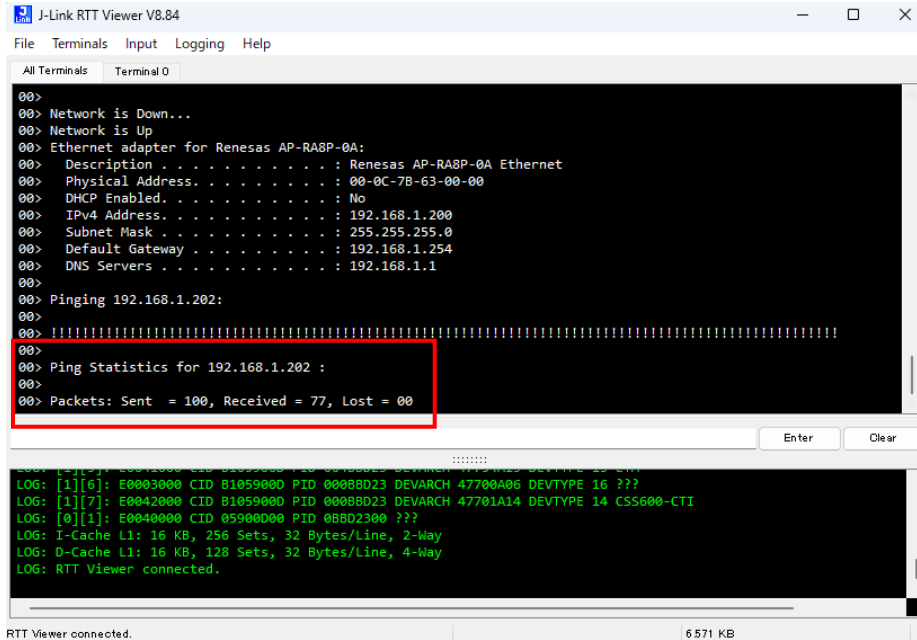
- ① LAN クロスケーブルを用い、CPU ボードの LAN コネクタ (CN7) とホスト PC を接続します。
- ② ホスト PC 上でネットワークの設定を行います。  
 CPU ボードの設定に合わせるため、ホスト PC のネットワーク設定を下記の内容に変更してください。

|          |               |
|----------|---------------|
| IP アドレス  | 192.168.1.202 |
| サブネットマスク | 255.255.255.0 |
| ゲートウェイ   | 192.168.1.254 |

- ③ CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ④ ホスト PC 上で RTTViewer を起動します。接続設定を行い、RTTViewer の接続を確認します。  
 接続完了が確認できた後に、RTTViewer のターミナル画面を開きます。
- ⑤ サンプルプログラムが正常に動作した場合は、以下のように、ターミナル画面にネットワーク設定が表示され、その後、IP アドレス「192.168.1.202」に対し Ping 通信が開始されます。



- ⑥ 正常に Ping 通信が行えた場合、通信結果が表示されます。  
また、この状態で PC より CPU ボードに対して、Ping 応答を確認することも可能です。



### 3.2.2.1 ネットワーク推奨環境

本サンプルプログラムに実装されたネットワーク通信の確認に必要な推奨環境は以下の通りです。

|          |                             |
|----------|-----------------------------|
| ホスト PC   | PC/AT 互換機                   |
| OS       | Windows 10/11               |
| LAN ポート  | 10/100BASE-TX 以上対応の LAN ポート |
| LAN ケーブル | クロスケーブル                     |

3.2.2.2 ネットワーク設定

本 CPU ボードのネットワーク設定は以下の通りです。

|                  | CH0(CN7)                                    | CH1(CN6)      |
|------------------|---|---------------|
| IP アドレス          | 192.168.1.200                               | 192.168.1.201 |
| サブネットマスク         | 255.255.255.0                               | 255.255.255.0 |
| ゲートウェイ           | 192.168.1.254                               | 192.168.1.254 |
| MAC アドレス         | 00-0C-7B-63-XX-XX<br>※ XX-XX の値は製品ごとに異なります。 |               |
| Ping 通信先 IP アドレス | 192.168.1.202                               |               |

上記設定のうち、IP アドレス・サブネットマスク・ゲートウェイの設定は、サンプルプログラム内で定義しています。各設定の定義は以下の通りです。

<¥sample¥ap\_ra8p\_0a\_ether\_sample\_ch0¥src¥net\_thread\_entry.c>

| 設定       | CPU ボードの設定       |
|----------|------------------|
| IP アドレス  | ucIPAddress      |
| サブネットマスク | ucNetMask        |
| ゲートウェイ   | ucGatewayAddress |

<¥sample¥ap\_ra8p\_0a\_ether\_sample\_ch0¥src¥usr\_app.h>

| 設定               | CPU ボードの設定       |
|------------------|------------------|
| Ping 通信先 IP アドレス | USR_TEST_PING_IP |

また、MAC アドレスは EEPROM の先頭 12Byte に格納されています。

| EEPROM アドレス<br>CN7 | EEPROM アドレス<br>CN6 | 格納値  |
|--------------------|--------------------|------|
| 0x00               | 0x06               | 0x00 |
| 0x01               | 0x07               | 0x0C |
| 0x02               | 0x08               | 0x7B |
| 0x03               | 0x09               | 0x63 |
| 0x04               | 0x0A               | 0xXX |
| 0x05               | 0x0B               | 0xXX |

※ 0xXX の値は製品ごとに異なります

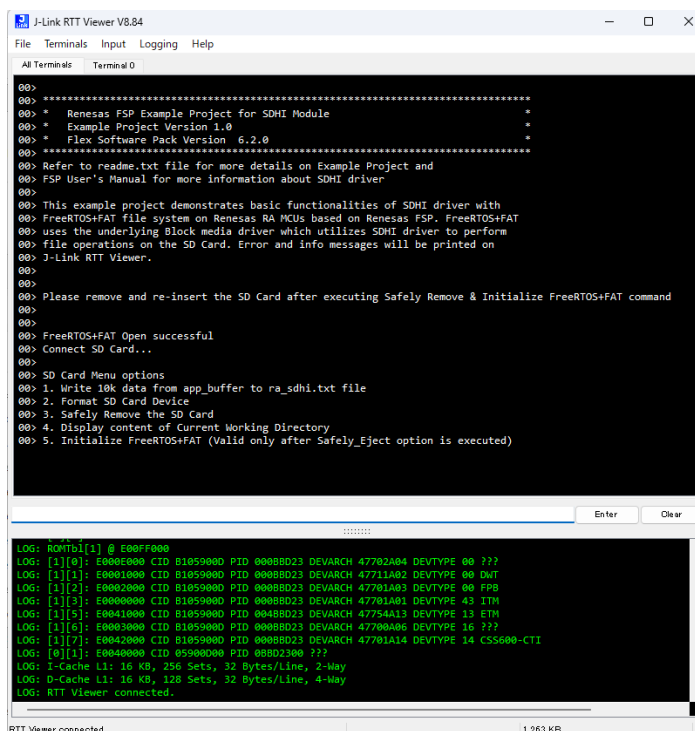
本製品の MAC アドレスは、弊社が米国電気電子学会 (IEEE) より取得したアドレスとなります。MAC アドレスを変更される際は、お客様にて IEEE より MAC アドレスを取得し、設定してください。

### 3.2.3 SDHI サンプルプログラムの動作説明

本サンプルプログラムでは、RTT Viewer を用いてコマンドを送信することで、microSD カードへのデータ読み書きを行うことができます。

SDHI 動作の確認は、以下の手順に従って行ってください。

- ① J-Link を使用し、CPU ボードとホスト PC を接続します。
- ② ホスト PC 上で RTTViewer を起動します。
- ③ CPU ボードの microSD カードを挿入した後、電源を投入し、サンプルプログラムを動作させます。
- ④ ホスト PC 上で RTTViewer の接続動作を行います。  
接続完了が確認できた後に、RTTViewer のターミナル画面を開きます。
- ⑤ サンプルプログラムが正常に動作した場合は、以下のように、ターミナル画面に SD 動作ログが表示され、その後コマンド受信待ち状態が表示されます。



- ⑥ 下記のコマンド一覧を参考に、ターミナルにコマンドを入力します。  
入力されたコマンドに合わせて、下記の動作が行われる事を確認してください。

| コマンド | 説明  |
|------|---|
| 1    | microSD カードにテスト用ファイルの書き込みと、書き込み後の読み出し確認を行います。 |
| 2    | microSD カードのフォーマットを行います。                      |
| 3    | microSD カードの安全な拔出準備を行います。                     |
| 4    | microSD カードのカレントディレクトリ情報を表示します。               |
| 5    | RTOS、FAT の初期化を行います。                           |

## 3.2.4 UART/メモリサンプルプログラムの動作説明

本サンプルプログラムでは、シリアル通信 (SCI1) を用いてコマンドを送信することで、LED の輝度を変更することができます。モニタ LED1 は 1~100% の範囲で 1% 単位で調整が可能で、1% 時に最も暗く、100% 時に最も明るくなります。動作確認は、ホスト PC 上のターミナルソフト (ハイパーターミナルなど) を使用して行ってください。ターミナルソフトの COM ポート設定は、115200bps、ビット長 8、パリティなし、ストップビット 1、フロー制御なしです。また、ターミナルソフトの改行コードの設定は、「CR」 (0x0d) としてください。

電源投入後、SDRAM の初期化を行い、その後、以下の順に SDRAM と OSPI Flash ROM 内のメモリ確認処理を行います。

- ① 1024Byte のダミーデータを SDRAM 上の変数 s\_sdram\_buff1[1024] に、1024Byte の 0xFF データを SDRAM 上の変数 s\_sdram\_buff2[1024] にそれぞれ書き込みます。
- ② 変数 s\_sdram\_buff1[1024] のデータを OSPI Flash ROM の先頭アドレスから順に書き込みます。
- ③ OSPI Flash ROM の先頭アドレスから 1024Byte 分のデータを、SDRAM 上の変数 s\_ospi\_buff1[1024] に読み出します。
- ④ 変数 s\_sdram\_buff1[1024] と s\_ospi\_buff1[1024] を比較し、データが一致した場合は「ospi write check OK.」と表示されます。
- ⑤ OSPI Flash ROM の先頭 1 セクタをイレースします。
- ⑥ OSPI Flash ROM の先頭アドレスから 1024Byte 分のデータを、SDRAM 上の変数 s\_ospi\_buff2[1024] に読み出します。
- ⑦ 変数 s\_sdram\_buff2[1024] と s\_ospi\_buff2[1024] を比較し、データが一致した場合は「ospi erase OK.」と表示されます。

コマンド送信の際は、コマンドの数値と改行コードを送信してください。

| コマンド | モニタ LED1 デューティ比 |
|------|-----------------|
| 1    | 1%              |
| 2    | 2%              |
| .    | .               |
| .    | .               |
| 50   | 50%             |
| .    | .               |
| .    | .               |
| 99   | 99%             |
| 100  | 100%            |

コマンド送信が正常に行われた場合、「Set next value」と表示され、モニタ LED1 が点灯します。コマンド入力で誤った値が入力されたなど、コマンド送信で異常が発生した場合は、「Invalid input. Input range is from 1 - 100」と表示されます。



### 3.2.6 USB ファンクションサンプルプログラムの動作説明

USB ファンクション動作の確認は、以下の手順に従って行ってください。

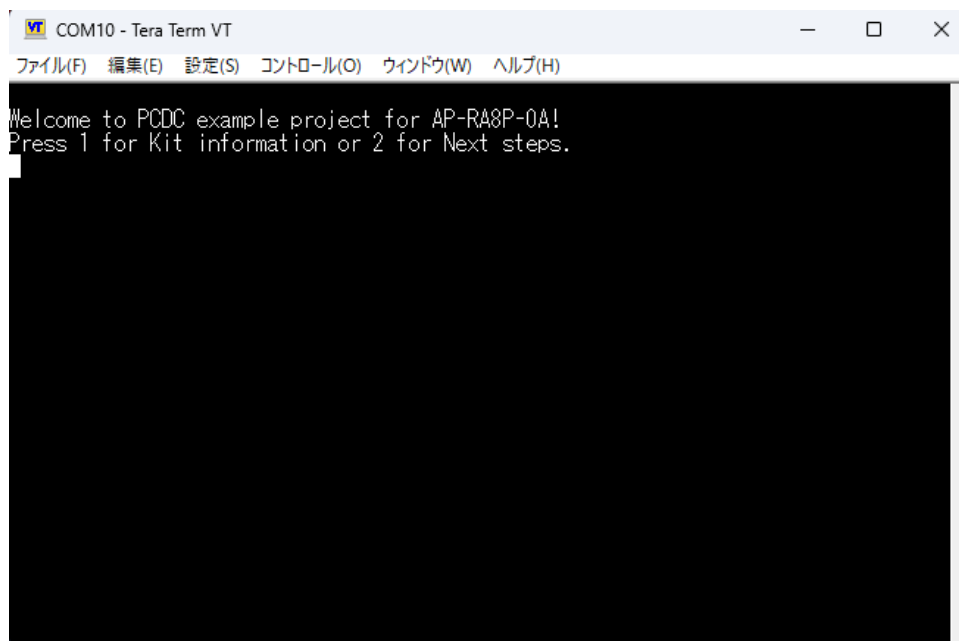
なお、Win10 よりも前の OS での USB ファンクションの動作確認は、あらかじめ USB 仮想シリアルドライバを PC にインストールしておく必要があります。

インストール方法につきましては、「AN178 USB 仮想シリアルドライバ インストールガイド」を参照してください。

- ① USB ケーブルを使い CPU ボードの USB ファンクションポート (CN4) とホスト PC の USB ポートを接続します。
- ② CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ③ ホスト PC 上でターミナルソフト (ハイパーターミナルなど) を起動し、COM ポートの設定を行います。  
その際使用する COM ポートは、「AN178 USB 仮想シリアルドライバインストールガイド」内で確認した仮想 COM ポートを選択してください。  
COM ポートを以下の設定に変更します。

|         |           |
|---------|-----------|
| ボーレート   | 115200bps |
| ビット長    | 8bit      |
| パリティ    | なし        |
| ストップビット | 1bit      |
| フロー制御   | なし        |

- ④ ターミナルソフトで改行コード「CR」 (0x0d) を送信します。  
コマンド入力待ち状態が表示されます。



- ⑤ 下記のコマンドを送信することで、情報の出力が行われます。  
コマンド送信の際は、コマンドの数値を送信してください。

| コマンド | 説明                   |
|------|----------------------|
| 1    | ボードの情報を表示します。        |
| 2    | RA 関係の URL 情報を表示します。 |

```

COM10 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)

Welcome to PCDC example project for AP-RA8P-0A!
Press 1 for Kit information or 2 for Next steps.

1. KIT INFORMATION
a) Kit name:                AP-RA8P-0A
b) Kit ordering part number: AP-RA8P-0A

Press 1 for Kit information or 2 for next steps.

2. Next steps
Visit the following URLs to learn about the kit and the RA family of MCUs, download
tools and documentation, and get support:
a) AP-RA8P-0A resources:    apnet.co.jp/product/ra/ap-ra8p-0a
b) RA product information:  renesas.com/ra
c) RA product support forum: renesas.com/ra/forum
d) Renesas support:        renesas.com/support

Press 1 for Kit information or 2 for next steps.

```

### 3.3 e2 studio を用いたプロジェクトのビルド・デバッグ

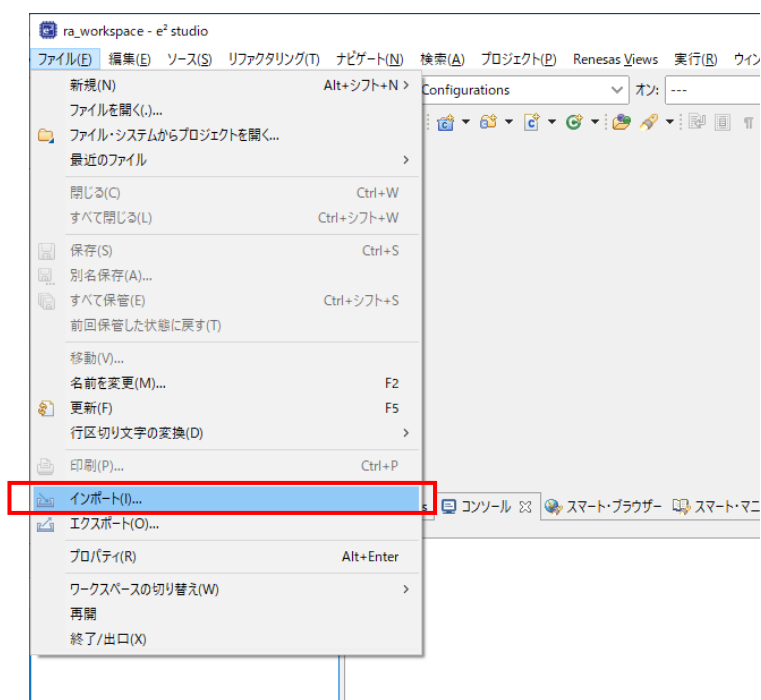
サンプルプログラムを CPU ボード上で実行するためには、e2 studio 上に一度サンプルプログラムをインポートし、ビルドを行う必要があります。

e2 studio 上へのサンプルプログラムのインポート方法、サンプルプログラムのビルド・デバッグ方法については、本節で説明します。

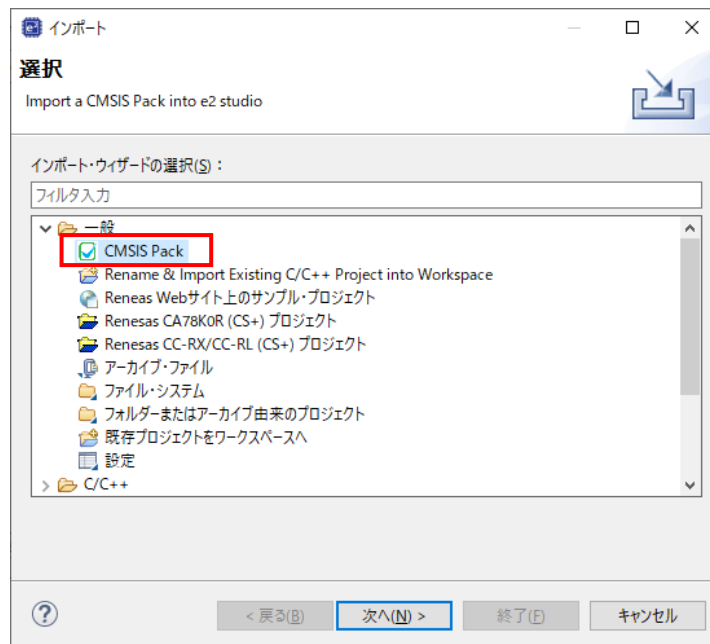
(下記で表示される図は「ap\_ra8p\_0a\_can\_sample」をデバッグ・ビルドする際の例として表示しています。プロジェクト名等は、ビルド・デバッグを行うサンプルプログラムにより変化します。)

#### 3.3.1 インポート方法

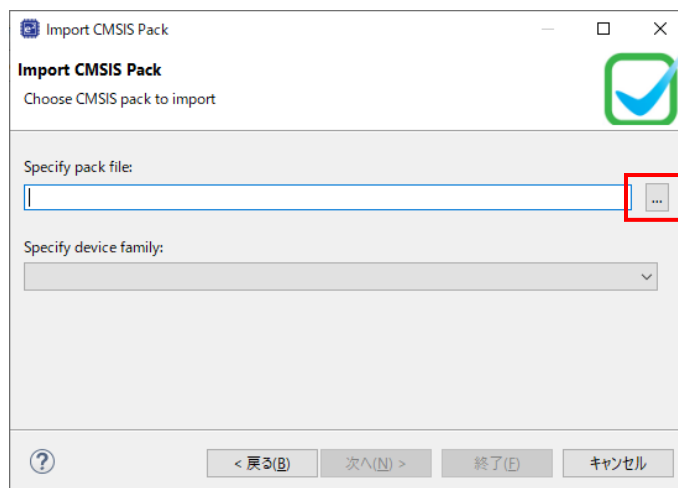
- ① e2 studio を起動し、ツールバーの [ファイル] → [インポート] を選択します。



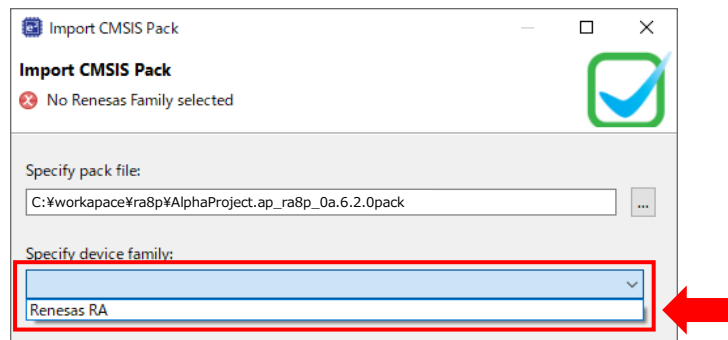
- ② [CMSIS Pack] を選択し [次へ] を選択し、pack ファイル「AlphaProject.ap\_ra8p\_0a.6.2.0.pack」をインポートします。
- すでに開発環境に pack ファイルをインポート済みである場合は、⑥へお進みください。



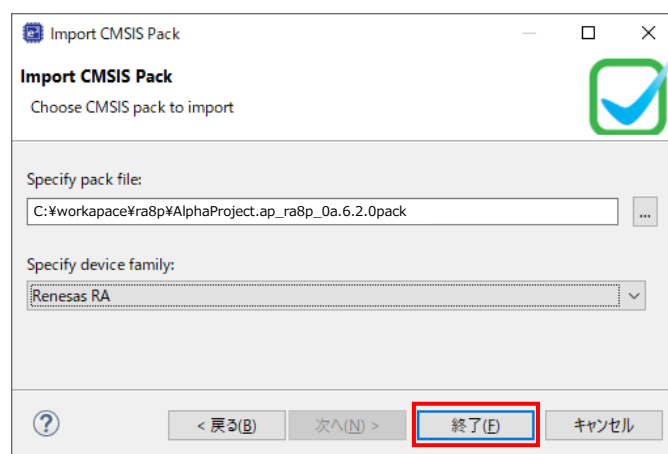
- ③ [Import RA CMSIS Pack ウィンドウ] が表示されましたら、インポートする pack ファイル「sample¥CustomBSP¥AlphaProject.ap\_ra8p\_0a.6.2.0.pack」を選択してください。



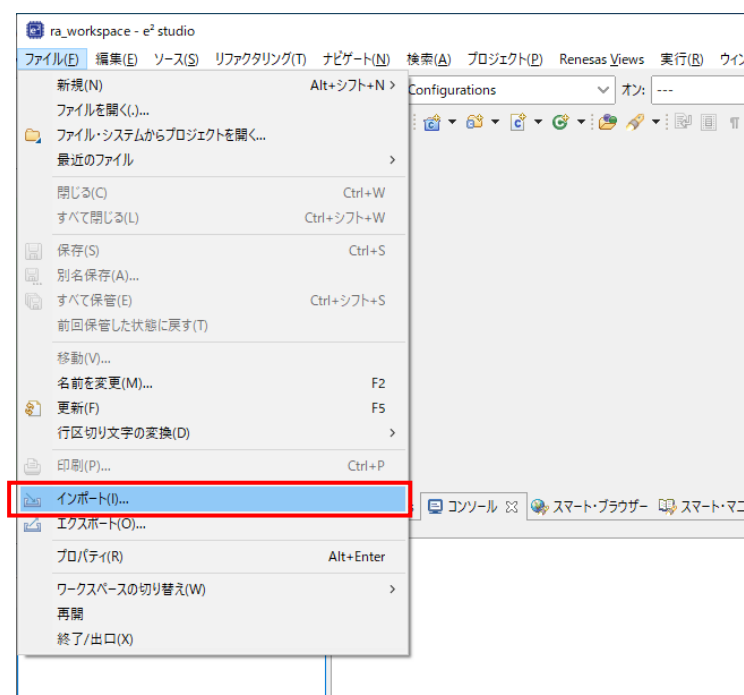
- ④ pack ファイルを選択後、メッセージ「No Renesas Family selected」が表示されるので、Specify device family から「Renesas RA」を選択してください。



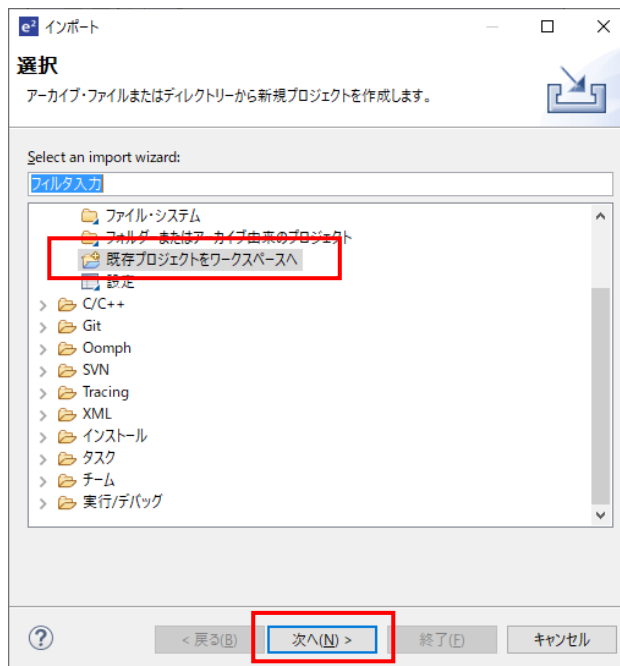
- ⑤ 「終了」を選択してください。



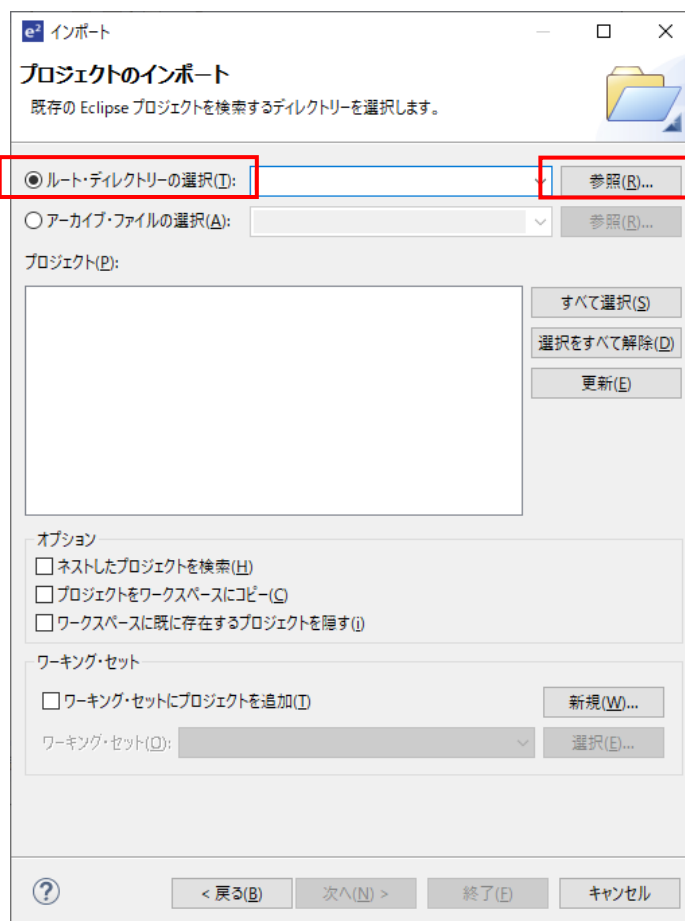
- ⑥ もう一度ツールバーの [ファイル] → [インポート] を選択します。



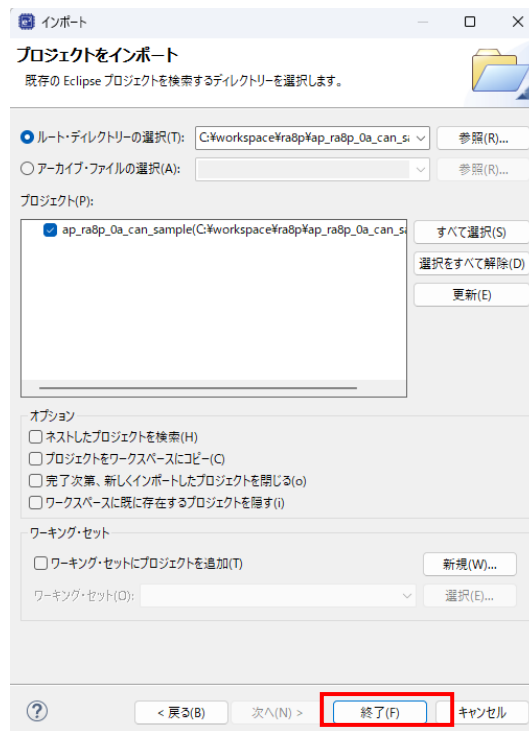
- ⑦ [既存のプロジェクトをワークスペースへ] を選択し [次へ] を選択します。



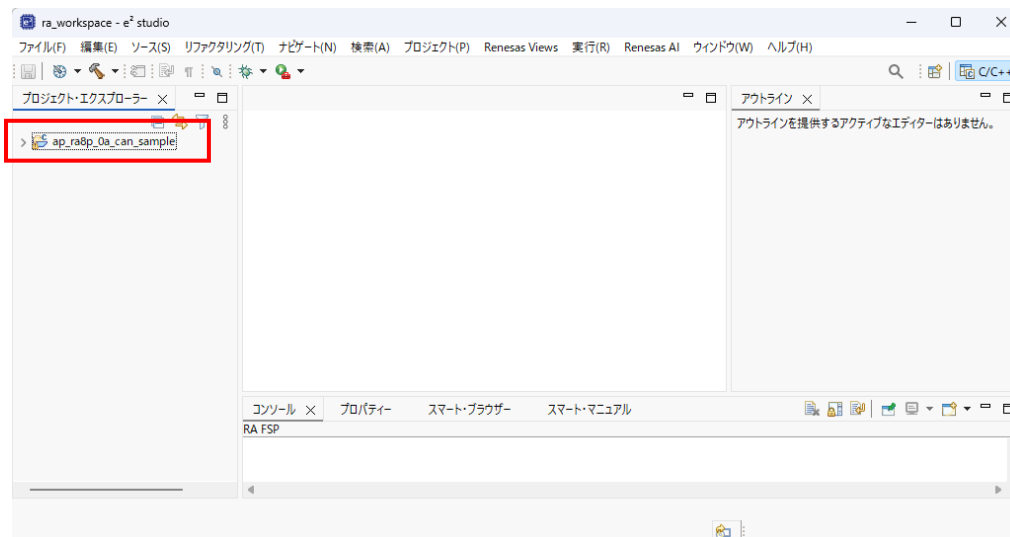
- ⑧ [ルート・ディレクトリの選択] を選択し、[参照] からサンプルプログラムのフォルダを選択します。



- ⑨ [終了] を選択します。



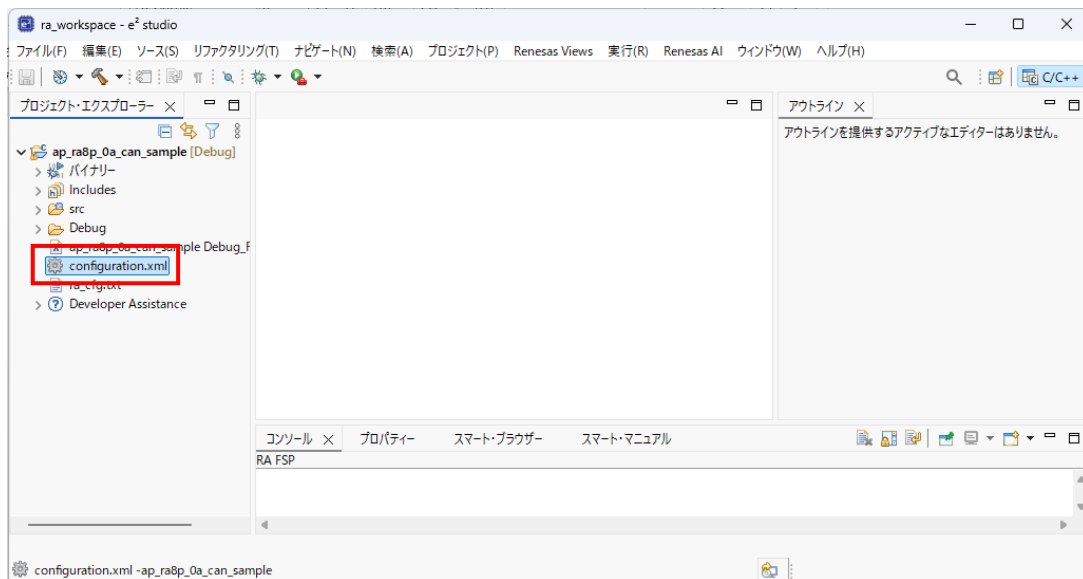
- ⑩ ナビゲーションウィンドウにサンプルプログラムのプロジェクトが追加されていることを確認します。



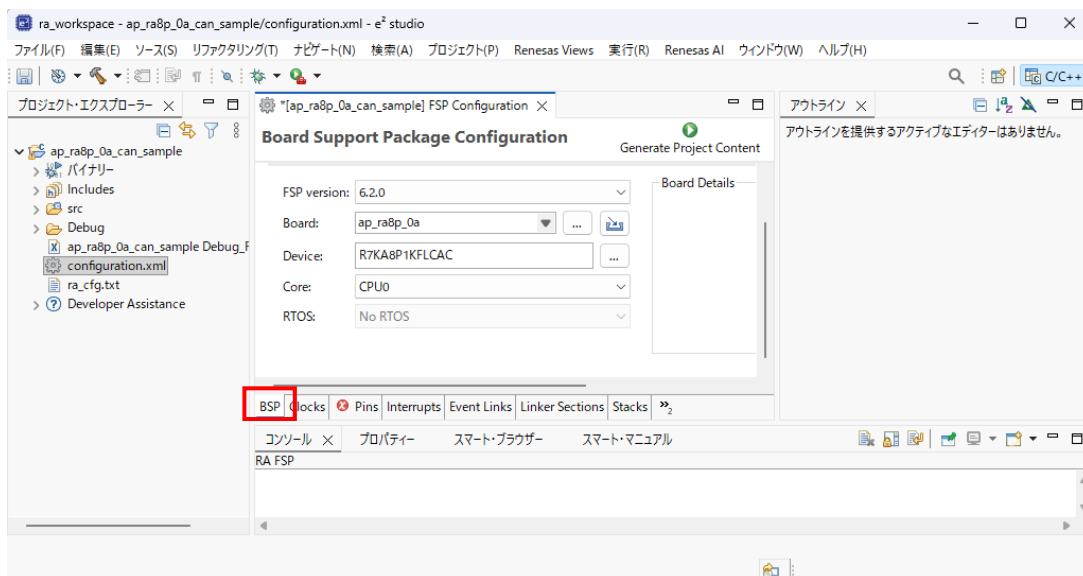
以上でプロジェクトのインポートは完了です。

## 3.3.2 ビルド方法

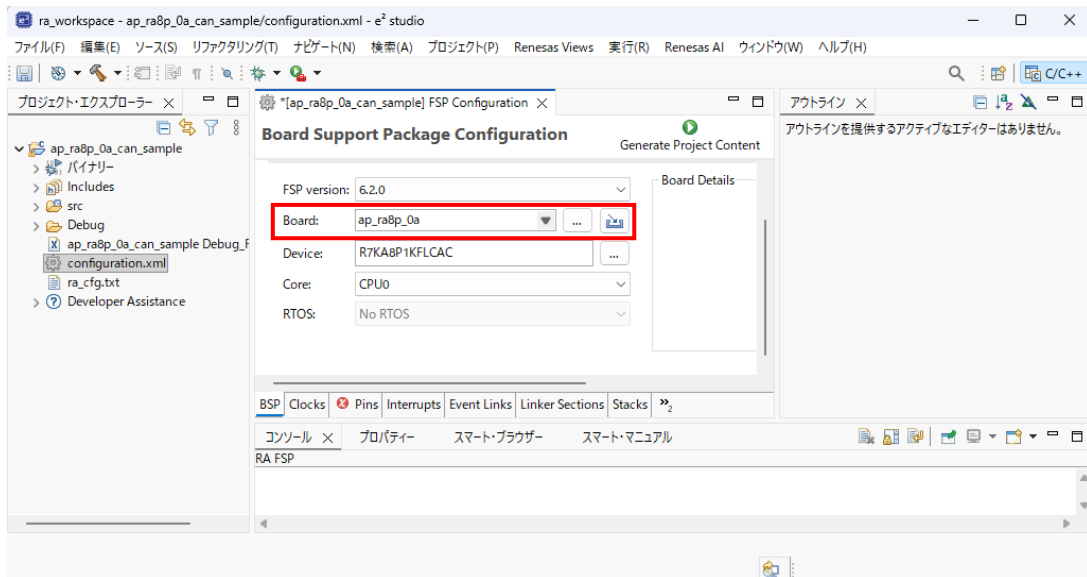
- ① プロジェクトのコンフィギュレータファイルを開きます。



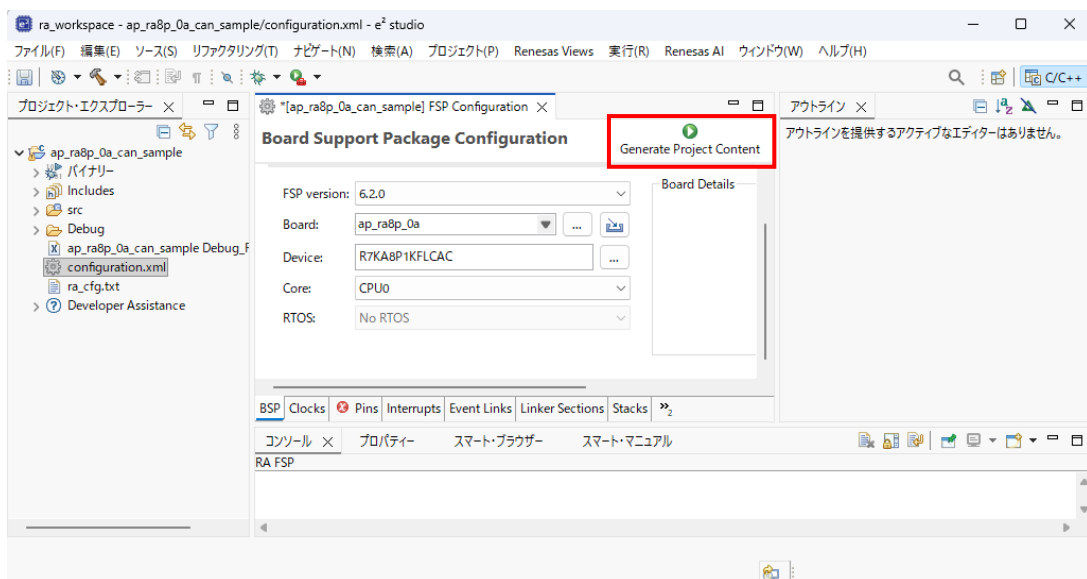
- ② [BSP] タブを開きます。



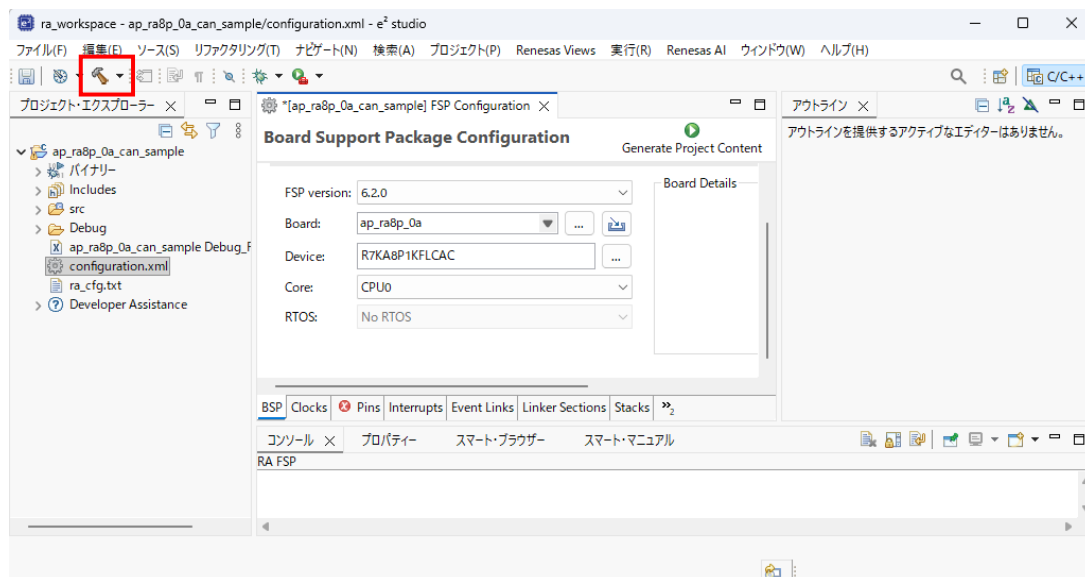
- ③ [BSP] タブで [Board] が「ap\_ra8p\_0a」であることを確認します。



- 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。



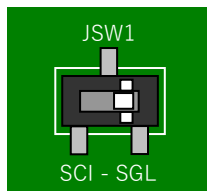
- ⑤ ツールバーからビルドアイコンを選択します。  
ビルドが成功すると、¥Debug ワークフォルダにオブジェクトファイルが生成されます。



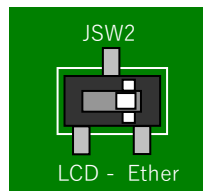
e2 studio の詳細な使用方法に関しては、e2 studio のマニュアルを参照してください。

## 3.3.3 デバッグ、ダウンロード方法

- ① 「3.3.2 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。
- ② ボード上のディップスイッチを以下のように設定してください。

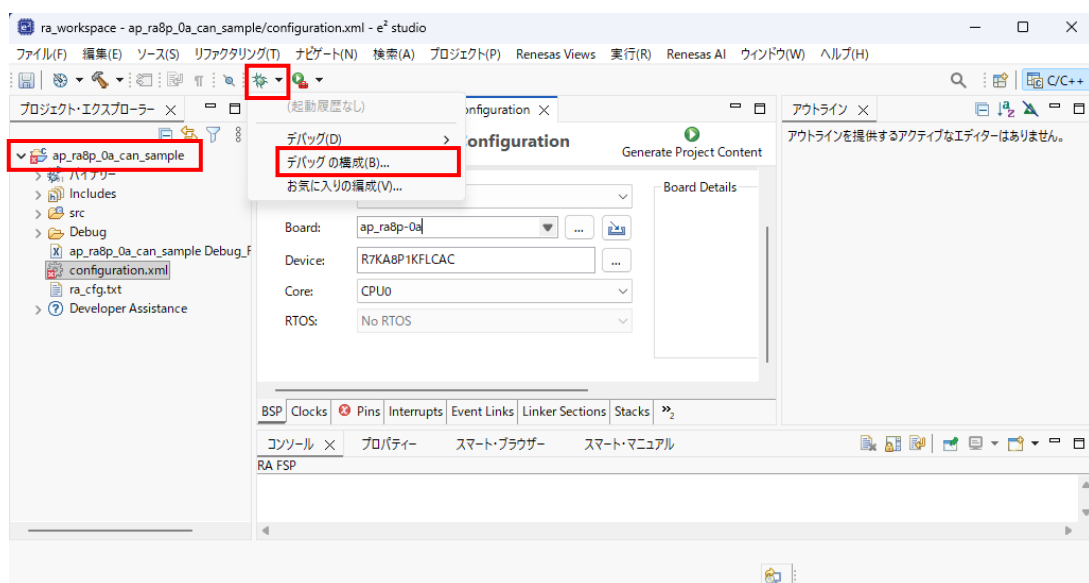


JSW1 : SGL  
シングルチップモード



JSW2  
Ether : Ethernet を使用する  
LCD : LCD を使用する

- 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。

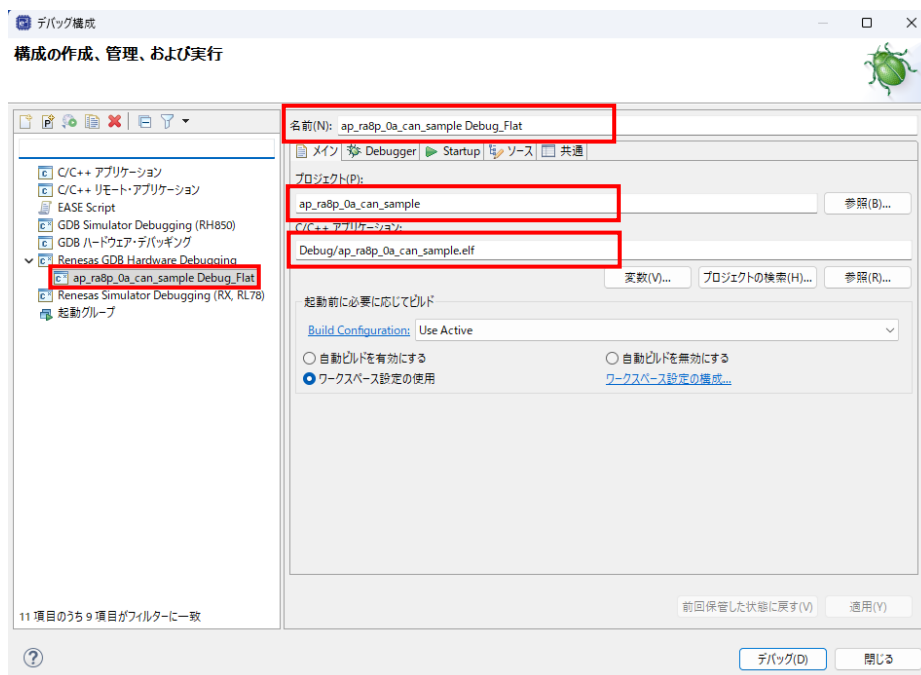


⑤ [Renesas GDB Hardware Debug] の [ap\_ra8p\_0a\_XXXX Debug] を選択し、下記の内容になっていることを確認してください。

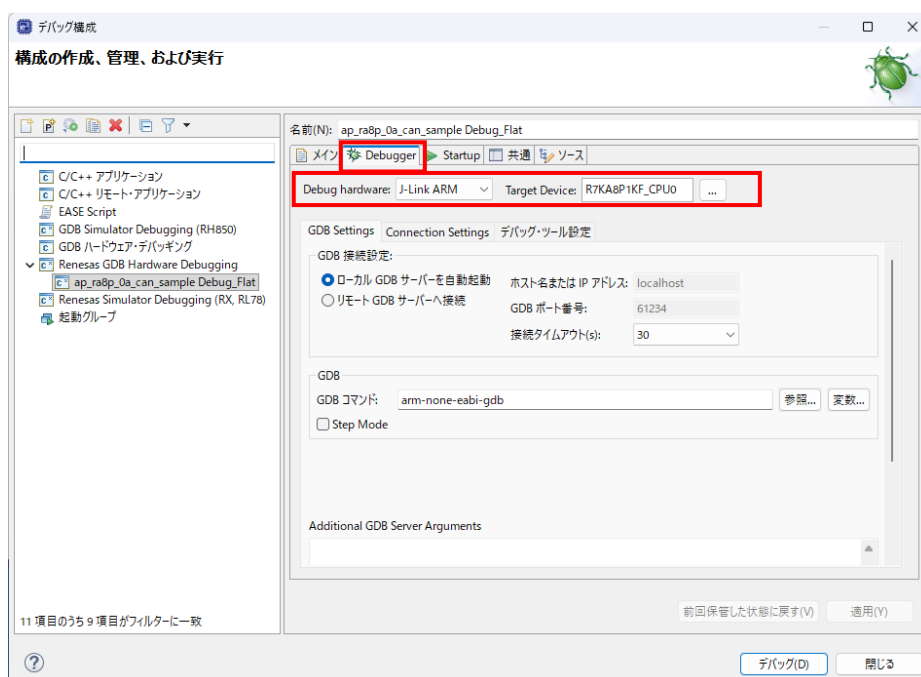
- [名前] : ap\_ra8p\_0a\_XXXX Debug
- [プロジェクト] : ap\_ra8p\_0a\_XXXX
- [C/C++アプリケーション] : Debug\ap\_ra8p\_0a\_XXXX.elf

※.XXXX の個所は、デバッグ対象のサンプルプログラムにより名称が異なります。

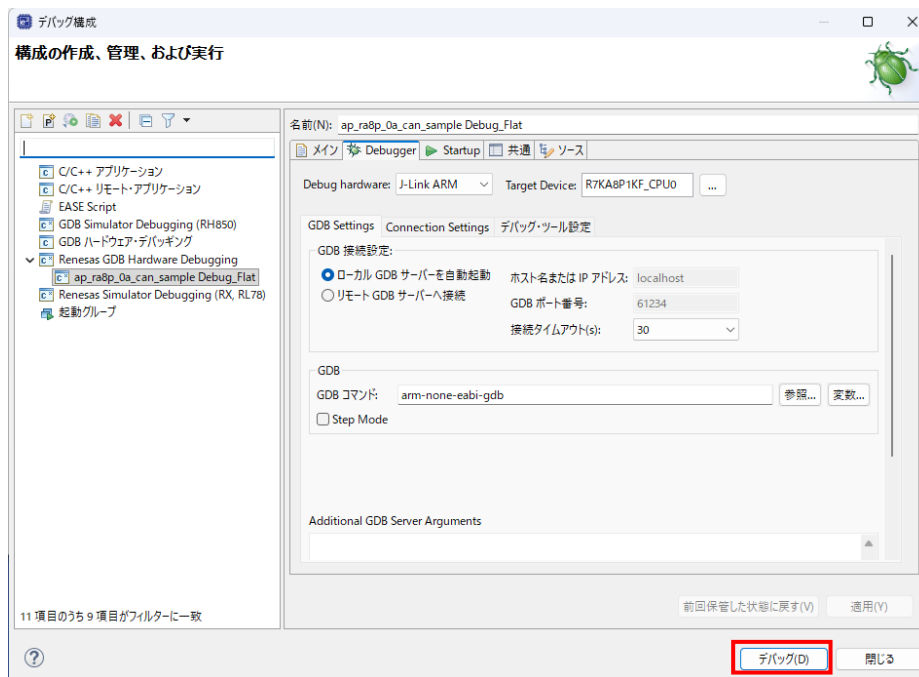
「2.2 フォルダ構成」を参考に、デバッグ対象のサンプルプログラムに合わせたファイルを選択してください。



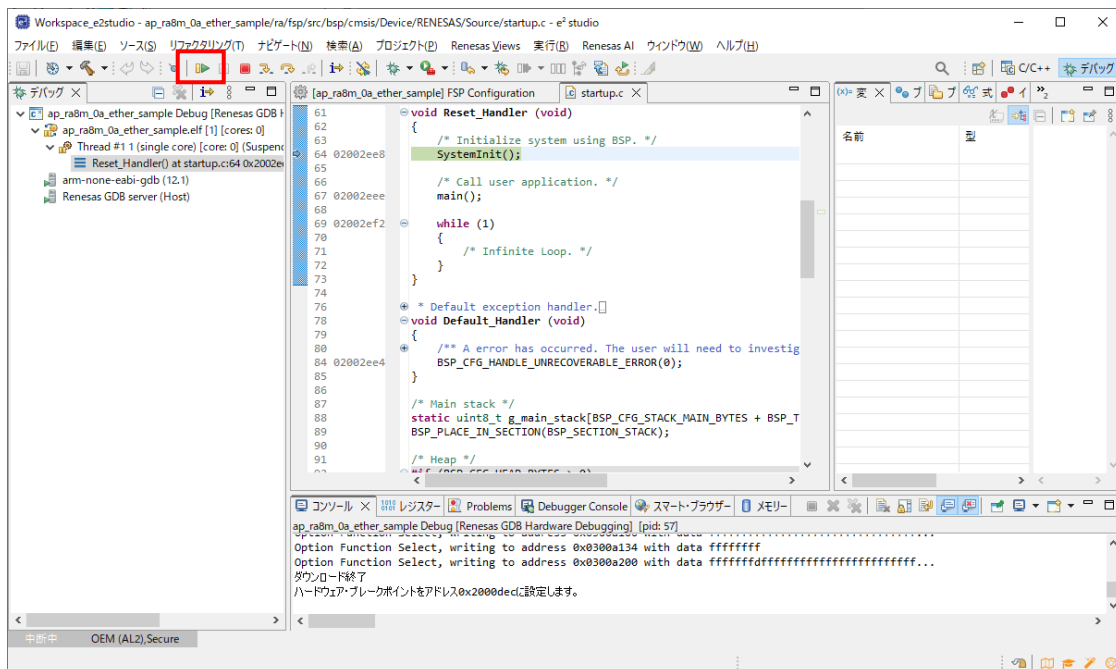
⑥ [Debugger] タブを選択し、[Debug hardware] が [J-Link ARM]、[Target Device] が [R7KA8P1KF\_CPU0] に設定されていることを確認してください。



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。以降、電源投入によりダウンロードされたプログラムの動作が開始されます。

## 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。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

- ・ RA8P1 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。
- ・ 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。

- ・ Arm® は Arm Ltd. の登録商標です。
- ・ J-Link は、SEGGER Microcontroller GmbH & Co. KG の登録商標もしくは商標です。

- ・ Windows® の正式名称は Microsoft® Windows® Operating System です。
  - ・ Microsoft、Windows は、米国 Microsoft Corporation. の米国およびその他の国における商標または登録商標です。
  - ・ Windows® 10、Windows® 11 は、米国 Microsoft Corporation. の商品名称です。
- 本文書では下記のように省略して記載している場合がございます。ご了承ください。

Windows® 10 は Windows 10 もしくは Win10  
Windows® 11 は Windows 11 もしくは Win11

- ・ 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。



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](mailto:query@apnet.co.jp)