

AP-RZT-0A (RZ/T1 CPU BOARD) USB ホスト サンプルプログラム (GCC) 解説

3.1 版 2023年10月02日

| | |
|--------------------------------------|----------|
| 1. 概要 | 2 |
| 1.1 概要 | 2 |
| 1.2 接続概要 | 2 |
| 1.3 本サンプルプログラムについて | 3 |
| 1.4 開発環境について | 3 |
| 1.5 ワークスペースについて | 4 |
| 2. サンプルプログラムの構成 | 5 |
| 2.1 フォルダ構成 | 5 |
| 2.2 ファイル構成 | 6 |
| 3. AP-RZT-0A サンプルプログラム | 9 |
| 3.1 動作説明 | 9 |
| 3.2 メモリマップ | 12 |
| 3.2.1 RAM 動作時のメモリマップ | 12 |
| 3.2.2 シリアル FlashROM 動作時のメモリマップ | 13 |
| 3.3 ビルド・デバッグ方法 | 14 |
| 3.3.1 プロジェクトのインポート | 14 |
| 3.3.2 プロジェクトのビルド | 18 |
| 3.3.3 RAM 動作 | 19 |
| 3.3.4 シリアル FlashROM 動作 | 21 |

1. 概要

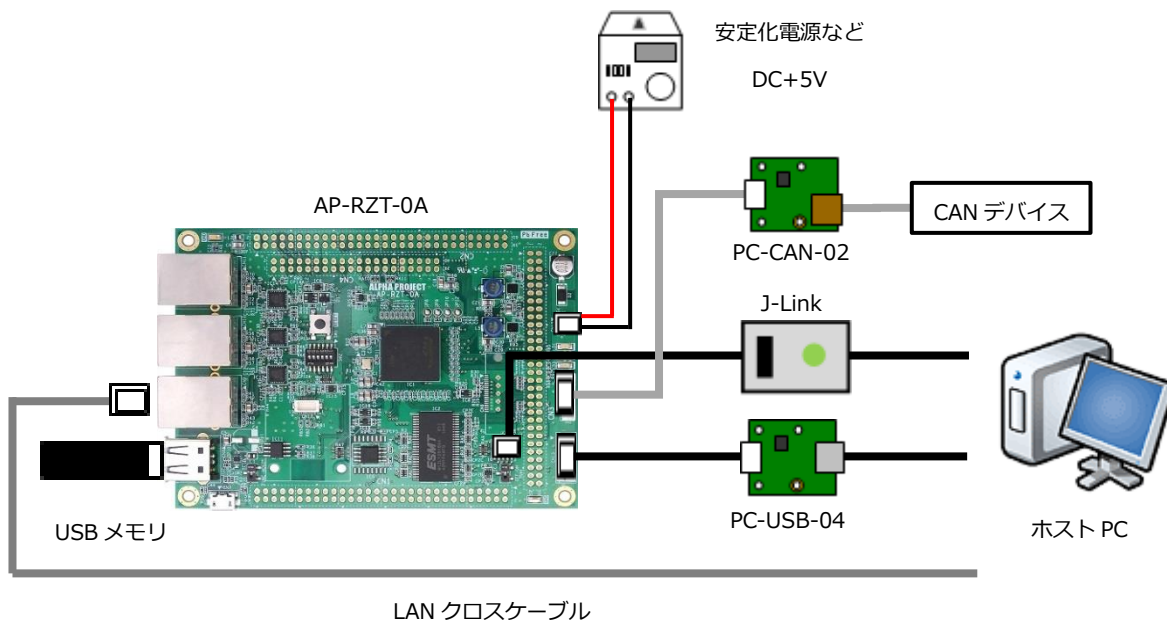
1.1 概要

本アプリケーションノートでは、AP-RZT-0A に付属するサンプルプログラムについて解説します。
 本サンプルプログラムの概要を以下に記します。

| サンプルプログラム | 動作内容 |
|-------------------|---|
| USB ホスト サンプルプログラム | <ul style="list-style-type: none"> ・ USB ホスト ファイル書き込み ・ シリアル通信 ・ CAN 通信 ・ ネットワーク通信 ・ タイマ割り込み |

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 ボードとホスト PC 間の接続例を以下に示します。



※AP-RZT-0A と J-Link を直接接続することはできません。
 AP-RZT-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。

1.3 本サンプルプログラムについて

本サンプルプログラムは、ルネサス エレクトロニクス株式会社提供のミドルウェア及びドライバを AP-RZT-0A に移植しています。

各ミドルウェア及びドライバの詳細については、以下の資料を参照してください。

ルネサス エレクトロニクス社 RZ/T1

<https://www.renesas.com/jp/ja/products/microcontrollers-microprocessors/rz-cortex-a-mpus/rzt1-microprocessors-real-time-control-industrial-equipment-and-networking-same-time#documents>

| |
|---|
| ● CMT |
| ・資料名 RZ/T1 グループ コンペアマッチタイマ(CMT) |
| ● SCIFA |
| ・資料名 RZ/T1 グループ FIFO 内蔵シリアルコミュニケーションインタフェース(SCIFA) |
| ● USB HMSC |
| ・資料名 RZ/T1 グループ USB Host Basic Firmware RZ/T1 グループ USB Host Mass Storage Class Driver(HMSC) |
| ● RIIC |
| ・資料名 RZ/T1 グループ RIIC サンプルプログラム |
| ● ETHERNET |
| ・資料名 RZ/T1 グループ マルチポート対応 ETHERNET ドライバ |

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 を用いて開発されています。

サンプルプログラムに対応する開発環境、ツールチェーンのバージョン、デバッガは次のようになります。

| 開発環境 | バージョン | ツールチェーン名 | バージョン | デバッガ |
|-----------------------|-----------|----------------------------|-------|--------|
| e ² studio | 5.3.0.023 | GNUARM-NONE-EABI Toolchain | 16.01 | J-Link |

1.5 ワークスペースについて

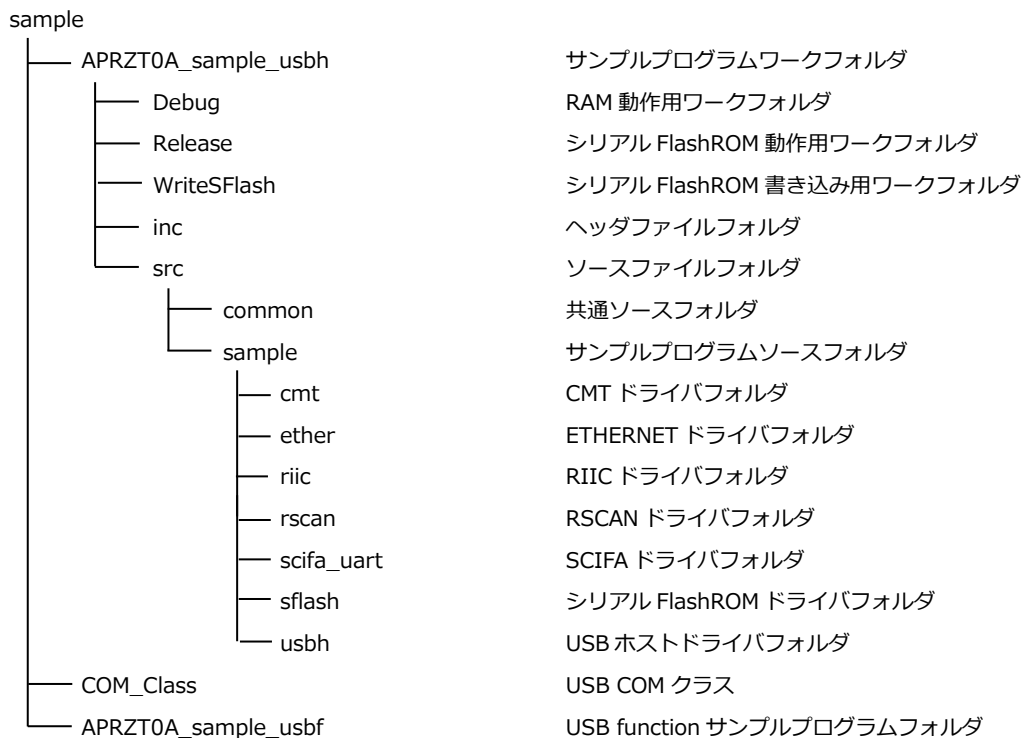
本サンプルプログラムの統合開発環境 e² studio ワークスペースは次のフォルダに格納されています。

| サンプルプログラム | フォルダ |
|---------------------------------|-----------------------------|
| USB ホスト サンプルプログラム プロジェクトフォルダ | ¥sample¥APRZT0A_sample_usbh |

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。



- サンプルプログラムワークフォルダ
 - RAM 動作用ワークフォルダ
 - シリアル FlashROM 動作用ワークフォルダ
 - シリアル FlashROM 書き込み用ワークフォルダ
 - ヘッダファイルフォルダ
 - ソースファイルフォルダ
 - 共通ソースフォルダ
 - サンプルプログラムソースフォルダ
 - CMT ドライバフォルダ
 - ETHERNET ドライバフォルダ
 - RIIC ドライバフォルダ
 - RSCAN ドライバフォルダ
 - SCIFA ドライバフォルダ
 - シリアル FlashROM ドライバフォルダ
 - USB ホストドライバフォルダ
 - USB COM クラス
 - USB function サンプルプログラムフォルダ
- ※ USB function サンプルプログラムに関しては
 アプリケーションノート「**AN1613 AP-RZT-0A
 USB ファンクション サンプルプログラム (GCC) 解説**」を
 参照してください。

2.2 ファイル構成

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。

<¥sample¥APRZT0A_sample_usbh フォルダ内>

| | | |
|--|-----|------------------------|
| .cproject | ... | CPROJECT ファイル |
| .Debuglinker | ... | DEBUGLINKER ファイル |
| .info | ... | INFO ファイル |
| .project | ... | PROJECT ファイル |
| .Releaselinker | ... | RELEASELINKER ファイル |
| .WriteSFlashlinker | ... | WRITESFLASHLINKER ファイル |
| APRZT0A_sample_usbh Debug.jlink | ... | JLINK ファイル |
| APRZT0A_sample_usbh Debug.launch | ... | LAUNCH ファイル |
| APRZT0A_sample_usbh WriteSFlash.jlink | ... | JLINK ファイル |
| APRZT0A_sample_usbh WriteSFlash.launch | ... | LAUNCH ファイル |
| custom.bat | ... | カスタムバッチファイル |
| makefile.init | ... | メイクファイル |

<¥sample¥APRZT0A_sample_usbh¥Debug フォルダ内>

| | | |
|-----------------------|-----|-----------------|
| APRZT0A_sample_usbh.x | ... | RAM 動作用実行可能ファイル |
|-----------------------|-----|-----------------|

<¥sample¥APRZT0A_sample_usbh¥Release フォルダ内>

| | | |
|-------------------------|-----|---------------------------|
| APRZT0A_sample_usbh.bin | ... | シリアル FlashROM 動作用バイナリファイル |
|-------------------------|-----|---------------------------|

<¥sample¥APRZT0A_sample_usbh¥WriteSFlash フォルダ内>

| | | |
|-----------------------|-----|-----------------------------|
| APRZT0A_sample_usbh.x | ... | シリアル FlashROM 書き込み用実行可能ファイル |
|-----------------------|-----|-----------------------------|

<¥sample¥APRZT0A_sample_usbh¥inc フォルダ内>

| | |
|-----------------------|-----------------------------------|
| common.h | … 共通ヘッダファイル |
| eth_hwfnc.h | … ETHERNET ハードウェアファンクション定義ヘッダファイル |
| iodefinc.h | … IO レジスタ定義ヘッダファイル |
| platform.h | … プラットフォームヘッダファイル |
| r_atcm_init.h | … ATCM 初期化ヘッダファイル |
| r_cmt.h | … CMT ドライバヘッダファイル |
| r_cpg.h | … CPG 設定ヘッダファイル |
| r_ecm.h | … ECM 設定ヘッダファイル |
| r_eth.h | … Ethernet 関連ドライバ公開ヘッダファイル |
| r_eth_mac.h | … EthernetMAC ドライバ定義ヘッダファイル |
| r_eth_phy.h | … EthernetPHY ドライバ定義ヘッダファイル |
| r_eth_sw.h | … EthernetSwitch ドライバ定義ヘッダ |
| r_icu_init.h | … ICU 初期化ヘッダファイル |
| r_mpc.h | … MPC 設定ヘッダファイル |
| r_port.h | … ポート設定ヘッダファイル |
| r_ram_init.h | … RAM 初期化ヘッダファイル |
| r_reset.h | … リセット処理ヘッダファイル |
| r_riic_rzt1_config.h | … RIIC ドライバ設定ヘッダファイル |
| r_riic_rzt1_if.h | … RIIC ドライバ API ヘッダファイル |
| r_scifa_uart.h | … SCIFA ドライバヘッダファイル |
| r_system.h | … システム設定定義ヘッダファイル |
| r_typedefs.h | … 基本型定義ヘッダファイル |
| r_usb_basic_config.h | … USB ユーザ定義ヘッダファイル |
| r_usb_basic_if.h | … USB basic API ヘッダファイル |
| r_usb_hatapi_define.h | … USB 共通定義ヘッダファイル |
| r_usb_hmsc_config.h | … USB HMSC 設定ヘッダファイル |
| r_usb_hmsc_if.h | … USB HMSC ドライバヘッダファイル |
| sio_char.h | … シリアル IO 文字制御ヘッダファイル |

<¥sample¥APRZT0A_sample_usbh¥src¥common フォルダ内>

| | | |
|-----------------|---|-------------------------------|
| ap_rzt_0a_ram.x | … | RAM 動作用リンクスクリプトファイル |
| ap_rzt_0a_rom.x | … | シリアル FlashROM 動作用リンクスクリプトファイル |
| exit.c | … | 終了処理 |
| loader_init.asm | … | ローダープログラム 1 |
| loader_init2.c | … | ローダープログラム 2 |
| loader_param.c | … | SPI ブートモード用ローダーパラメータ |
| r_atcm_init.c | … | ATCM 初期化 |
| r_cpg.c | … | CPG 設定 |
| r_ecm.c | … | ECM 設定 |
| r_icu_init.c | … | ICU 初期化 |
| r_mpc.c | … | MPC 設定 |
| r_ram_init.c | … | RAM 初期化 |
| r_reset.c | … | リセット処理 |
| vector.asm | … | ベクタテーブルファイル |

<¥sample¥APRZT0A_sample_usbh¥src¥sample フォルダ内>

| | | |
|------------------|---|----------------------|
| init_main.c | … | メイン処理 |
| eeeprom.c | … | EEPROM ドライバ |
| eeeprom.h | … | EEPROM ドライバヘッダファイル |
| ethernet_app.c | … | Ethernet アプリケーション処理 |
| r_usb_hmsc_apl.c | … | USB HMSC ドライバ実行処理 |
| r_usb_hmsc_apl.h | … | USB HMSC ドライバヘッダファイル |
| r_usb_main.c | … | USB メイン処理 |
| sdram.c | … | SDRAM ドライバ |
| sdram.h | … | SDRAM ドライバヘッダファイル |
| siochar.c | … | シリアル IO 文字制御 |

3. AP-RZT-0A サンプルプログラム

3.1 動作説明

サンプルプログラムは、下記の動作を行います。

- USB ホスト

USB ホストポートに USB メモリを挿入すると、FAT ファイルシステムを利用して USB メモリにテキストファイルの書き込みが行なわれます。

PC に USB メモリを挿入し、USB メモリのルートディレクトリに「HMSCDEM0.TXT」という名前のテキストファイルが作成されていることを確認してください。

「HMSCDEM0.TXT」を開き、「a」が 512 バイト書き込まれていることを確認してください。

- シリアル通信

SCIFA2 でエコーバックを行います。(送受信割り込み使用)

COM ポートの設定は、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。

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。

- CAN 通信

CAN1 でエコーバックを行います。(受信 FIFO 割り込み・送信完了割り込み使用)

スタンダードフォーマット、データフレーム、データ長 1~8 バイト、通信速度 500kbps の設定で動作し、受信したデータを、そのまま送信 ID:B' 10101010100 に対して送信します。

- ネットワーク通信

Ethernet (CN9) で、Ping 応答を行います。

CPU ボードの CN9 と PC を LAN クロスケーブルで接続し、PC から IP アドレス「192.168.1.100」に対して Ping を送信してください。

(本サンプルプログラムでは、CN10、CN11 は動作しません。)

- タイマ割り込み

LD1 を 100msec 間隔で、LD2、LD4、LD5 を 200msec 間隔で ON/OFF します。(CMT 割り込み使用)

また、拡張 I/O ポートより方形波を出力します。周期とピン番号を次ページの表に示します。

タイミングの生成は CMT 割り込みを使用します。

方形波出力端子一覧(1)

| ピン番号 | ピン名 | 周期 | 備考 |
|--------|--|--------|----|
| CN1.5 | P21/IRQ1/CS0#/MTIC5V/TIOCB1/CTS0# | 20msec | |
| CN1.4 | P22/IRQ2/RD#/MTIOC7B/TIOCD0/SCK0 | 20msec | |
| CN1.3 | P23/A0/MTIC5U/TXD0/DACK1 | 20msec | |
| CN1.58 | P67/IRQ15/GTIOC3B/CTXD0/TEND0/USB_OVRCUR | 20msec | |
| CN2.26 | P11/IRQ9/MTIOC4D/GTIOC2B | 40msec | |
| CN2.25 | P12/MTIOC4B/GTIOC2A | 40msec | |
| CN2.24 | P13/RAS#/MTIOC4C/GTIOC1B | 40msec | |
| CN2.23 | P14/CAS#/MTIOC4A/GTIOC1A | 40msec | |
| CN2.21 | P16/CS4#/CS2#/MTIOC3B/GTIOC0A | 40msec | |
| CN2.20 | P17/CS5#/ETH1_TXER/PHYRESETOUT#/ADTRG0 | 40msec | |
| CN2.58 | P70/IRQ0/D16/MTIOC6D/RTS1#/USB_OVRCUR/TRACECLK/ENCIF00 | 40msec | |
| CN2.57 | P71/D17/POE0#/POE10#/TOC2/SCK1/TRACECTL/ENCIF00 | 40msec | |
| CN2.56 | P72/D18/MTIOC1A/TIC2/TXD1/SSITXD0/TRACEDATA0/ENCIF02 | 40msec | |
| CN2.55 | P73/IRQ3/D19/MTCLKB/RXD1/SSIRXD0/TRACEDATA1/ENCIF03 | 40msec | |
| CN2.54 | P74/D20/MTCLKA/CTS1#/SSL03/SSISCK0/TRACEDATA2 | 40msec | |
| CN2.53 | P75/IRQ13/D21/MTIOC4D/GTIOC2B/SSL00/TRACEDATA3/ENCIF04 | 40msec | |
| CN2.52 | P76/D22/MTIOC4B/GTIOC2A/SSL01/SSIWS0/TRACEDATA4 | 40msec | |
| CN2.51 | P77/D23/MTIOC4C/GTIOC1B/RSPCK0/TRACEDATA5 | 40msec | |
| CN2.40 | P91/AN101/CAS#/TXD2/ENCIF06 | 40msec | |
| CN2.39 | P92/AN102/CS5#/TOC3/RXD2 | 40msec | |
| CN2.38 | P93/AN103/MTIOC1A/TIC3/SCK2/ENCIF07 | 40msec | |
| CN2.37 | P94/AN104/IRQ4/MTCLKB/RTS2# | 40msec | |
| CN2.36 | P95/AN105/IRQ13/MTCLKA/CTS2# | 40msec | |
| CN2.35 | P96/AN106/POE0#/POE10# | 40msec | |
| CN2.34 | P97/AN107/IRQ7/A25/ADTRG1 | 40msec | |
| CN2.43 | PA0/D24/MTIOC4A/GTIOC1A/MOSI0_RED/TRACEDATA6/MDAT3 | 40msec | |
| CN2.44 | PA1/D25/MTIOC3D/GTIOC0B/MISO0/AUDIO_CLK/TRACEDATA7/MCLK3 | 40msec | |
| CN2.45 | PA2/D26/MTIOC3B/GTIOC0A/SSL02/DREQ2/MDAT2/ENCIF05 | 40msec | |
| CN2.46 | PA3/D27/ETHSWSECOUT/GTETRIG/TIOCA2/SCK2/DACK2/MCLK2 | 40msec | |
| CN2.30 | PD0/AN108/CS4# | 40msec | |
| CN2.29 | PD1/AN109/CS1# | 40msec | |
| CN2.28 | PD2/AN110/WAIT# | 40msec | |
| CN2.27 | PD4/AN112/ETH2_INT | 40msec | |
| CN2.31 | PK1/CS5# | 40msec | |
| CN2.32 | PK2/A23 | 40msec | |
| CN2.33 | PK3/A24 | 40msec | |

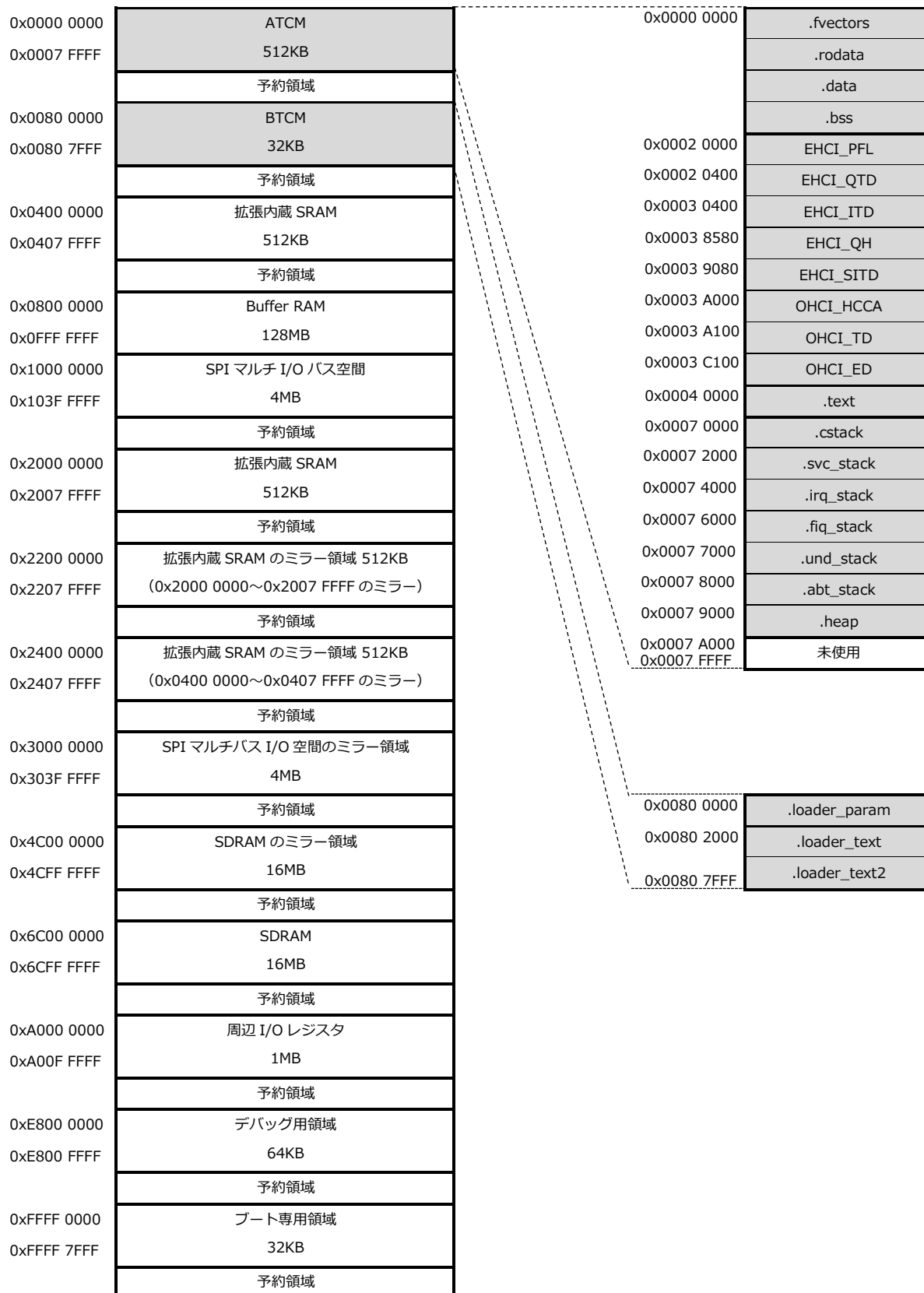
方形波出力端子一覧(2)

| ピン番号 | ピン名 | 周期 | 備考 |
|--------|--|--------|----|
| CN2.10 | PM2/CATSYNC1/CATLATCH1/TCLKE/RTS4# | 40msec | |
| CN2.9 | PM3/CATSYNC0/CATLATCH0/PO16 | 40msec | |
| CN2.7 | PM5/CATLEDSTER/PO18 | 40msec | |
| CN2.6 | PM6/IRQ6/CATLINKACT0/PO19 | 40msec | |
| CN2.5 | PM7/CATLINKACT1/PO20 | 40msec | |
| CN2.12 | PU7/CATIRQ/RXD4 | 40msec | |
| CN3.30 | P20/A17/MTCLKD | 20msec | |
| CN3.29 | P25/A18/MTCLKC/TEND1 | 20msec | |
| CN3.28 | P26/A19/MTIOC8D/DREQ1 | 20msec | |
| CN3.27 | P27/A20/MTIOC8C/TIOCB0/RTS0# | 20msec | |
| CN3.21 | P40/MTIOC8A/TXD0 | 20msec | |
| CN3.22 | P41/BS#/SCK0 | 20msec | |
| CN3.23 | P42/MTIOC7C/RXD0 | 20msec | |
| CN3.24 | P43/WE2#/DQMUL/MTIOC8B/USB_VBUSEN | 20msec | |
| CN3.25 | P44/IRQ12/WAIT#/TCLKD/ADTRG0/CTS0# | 20msec | |
| CN3.26 | P47/WE3#/DQMUU/AH#/MTIOC6C | 20msec | |
| CN3.53 | PP0/POE8#/TEND0/MCLK2 | 20msec | |
| CN3.54 | PP1/MTIOC0D/DACK0/MDAT2 | 20msec | |
| CN3.55 | PP2/MTIOC0C/TCLKH/MCLK1 | 20msec | |
| CN3.56 | PP3/MTIOC0B/TCLKC/MDAT1 | 20msec | |
| CN3.57 | PP4/MTIOC0A/MCLK0 | 20msec | |
| CN3.58 | PP5/PO22/MDAT0 | 20msec | |
| CN3.20 | PS0/MTIOC7D/AUDIO_CLK | 20msec | |
| CN3.19 | PS1/IRQ1/MTIOC7B/SSISCK0 | 20msec | |
| CN3.18 | PS2/MTIOC7C/SSIWS0 | 20msec | |
| CN3.17 | PS3/MTIOC7A/SSIRXD0 | 20msec | |
| CN3.16 | PS4/MTIOC6D/SSITXD0 | 20msec | |
| CN3.5 | PT0/IRQ0/TIOCA3/TIOCB3/PO25/SCK2/ENCIF07 | 20msec | |
| CN3.6 | PT1/TIOCA2/TIOCB2/PO26/RTS2# | 20msec | |
| CN3.7 | PT2/TIOCA1/TIOCB1/PO27 | 20msec | |
| CN3.8 | PT3/IRQ11/TIOCA0/TIOCB0/PO28/CTS2# | 20msec | |
| CN3.9 | PT4/CS3#/PO29 | 20msec | |
| CN3.10 | PT5/BS#/PO30/TEND2 | 20msec | |
| CN3.11 | PT6/A21/DREQ2 | 20msec | |
| CN3.12 | PT7/A22/DACK2 | 20msec | |
| CN4.8 | P55/IRQ5/A24/ETHSWSECOUT | 40msec | |
| CN4.18 | PF7/IRQ7/A25/ETH0_TXER/RTS3#/SSL30 | 40msec | |

3.2 メモリマップ

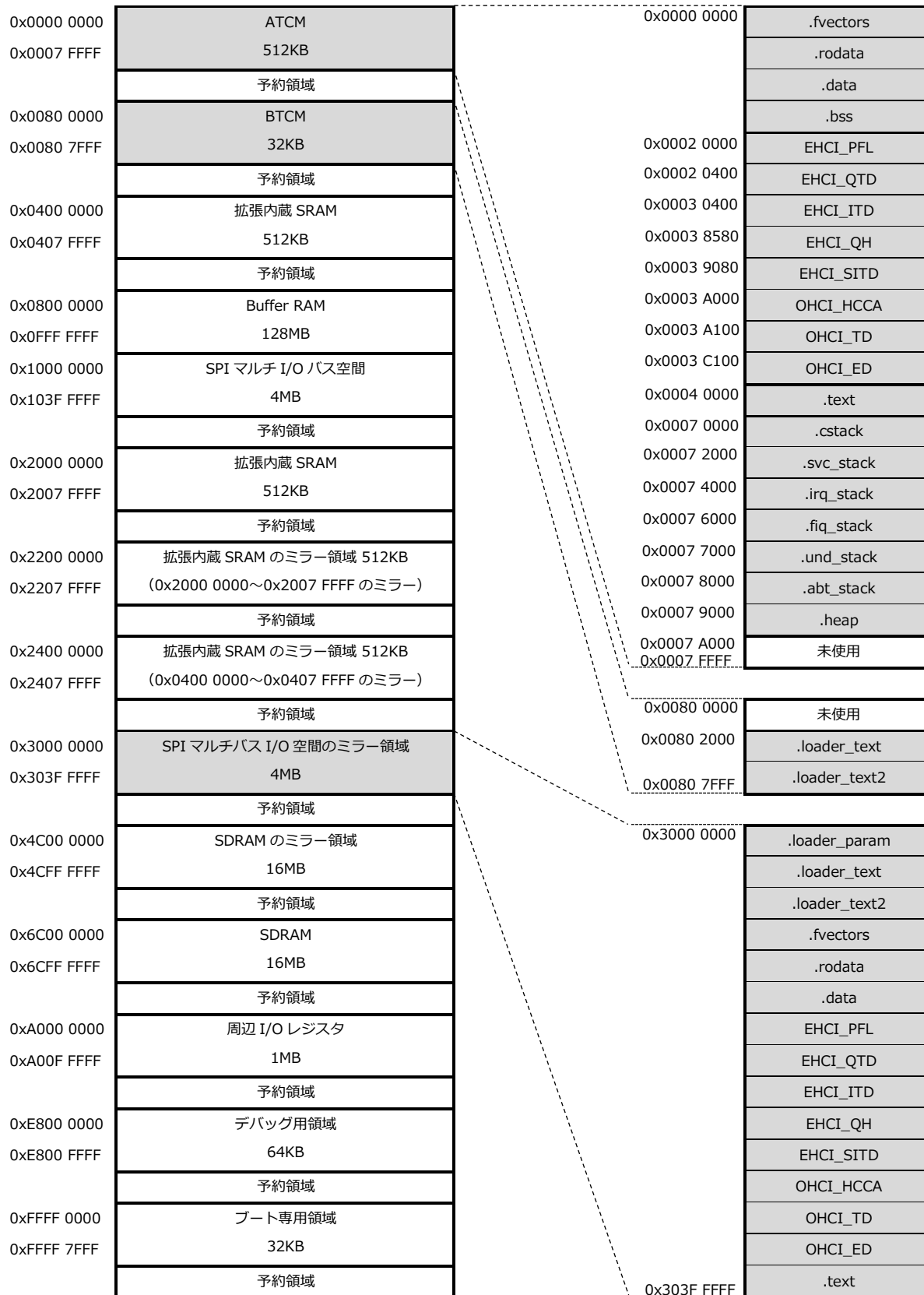
3.2.1 RAM 動作時のメモリマップ

RAM 動作時のメモリマップを以下に示します。



3.2.2 シリアル FlashROM 動作時のメモリマップ

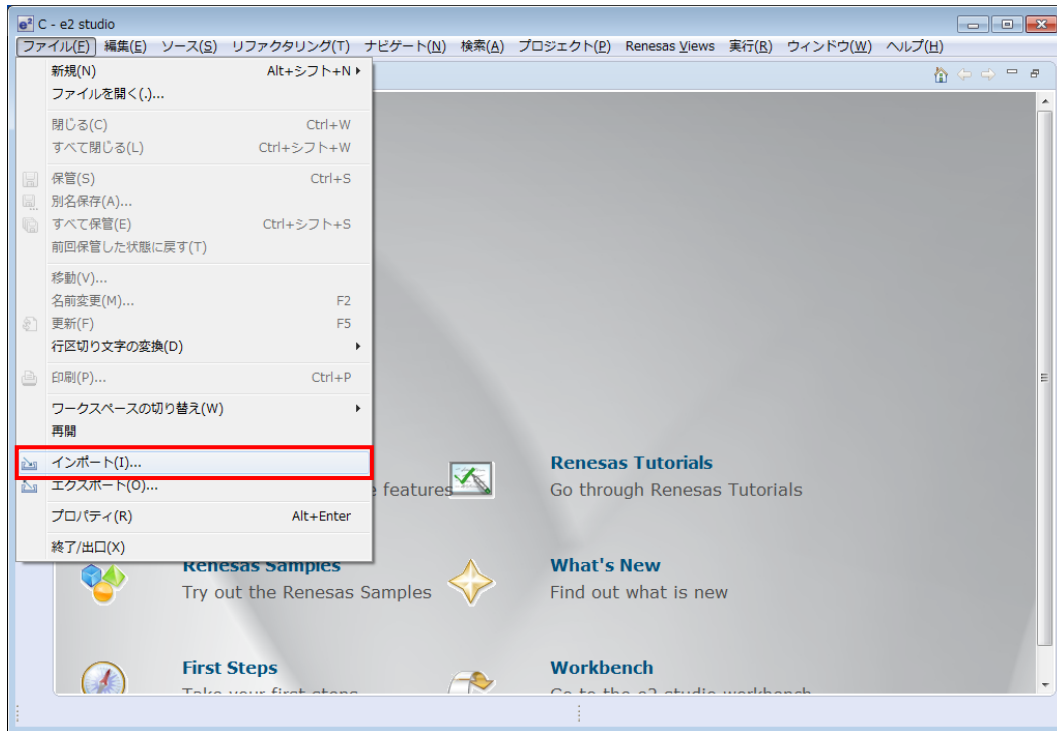
シリアル FlashROM 動作時のメモリマップを以下に示します。



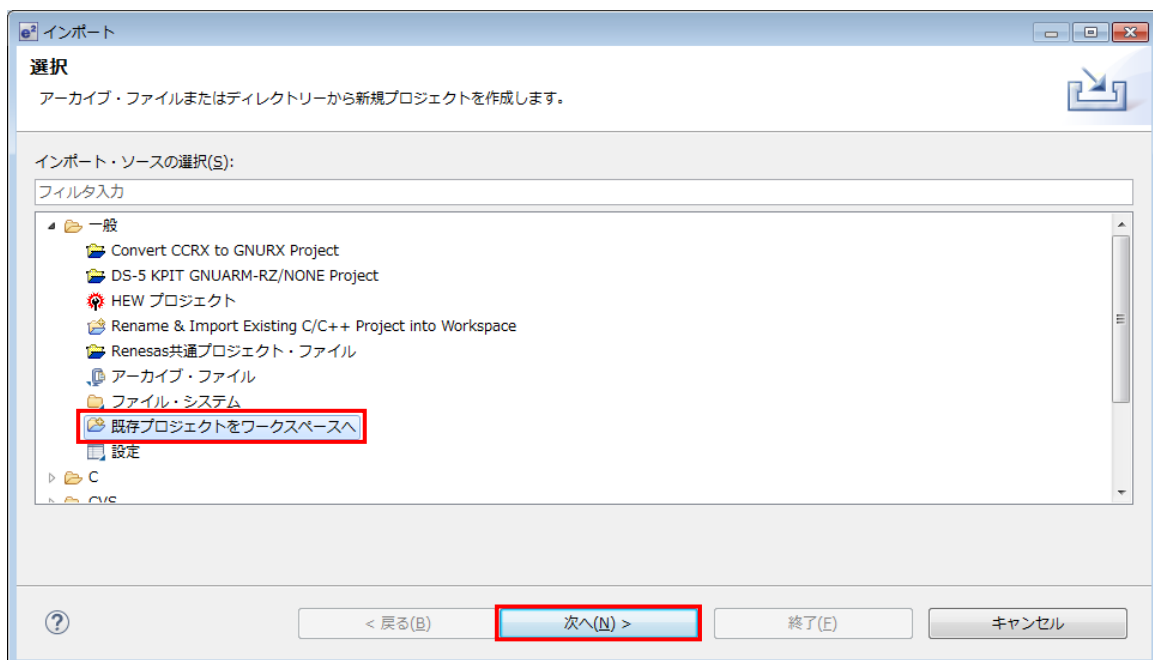
3.3 ビルド・デバッグ方法

3.3.1 プロジェクトのインポート

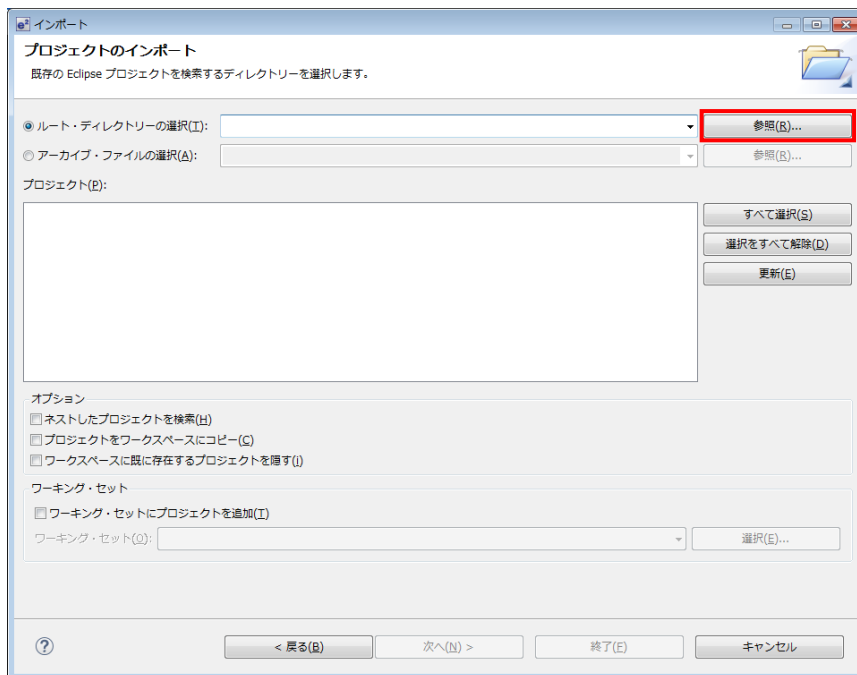
- ① e² studio を起動し、メニューバーの[ファイル]→[インポート]を選択します。



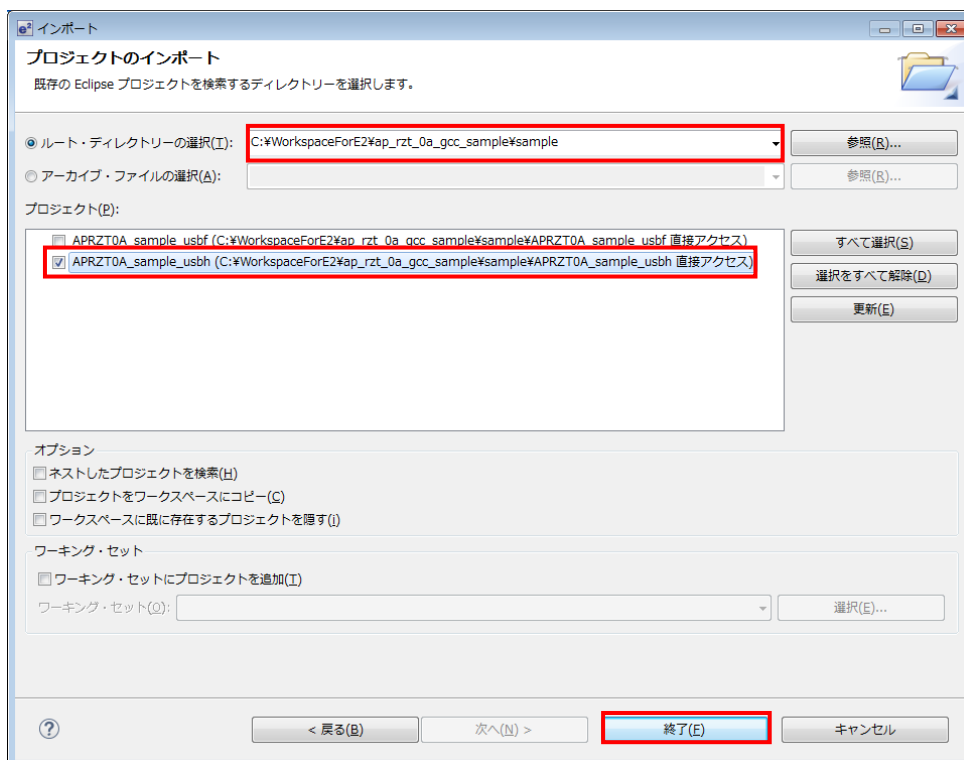
- ② [既存のプロジェクトをワークスペースへ]を選択し[次へ]を選択します。



- ③ [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。

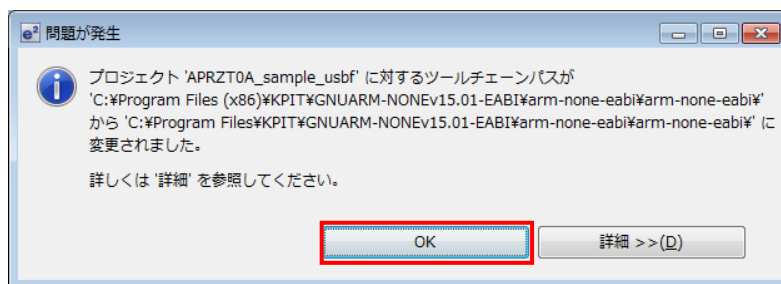


- ④ [APRZT0A_sample_usbh]にチェックを入れ[終了]を選択します。

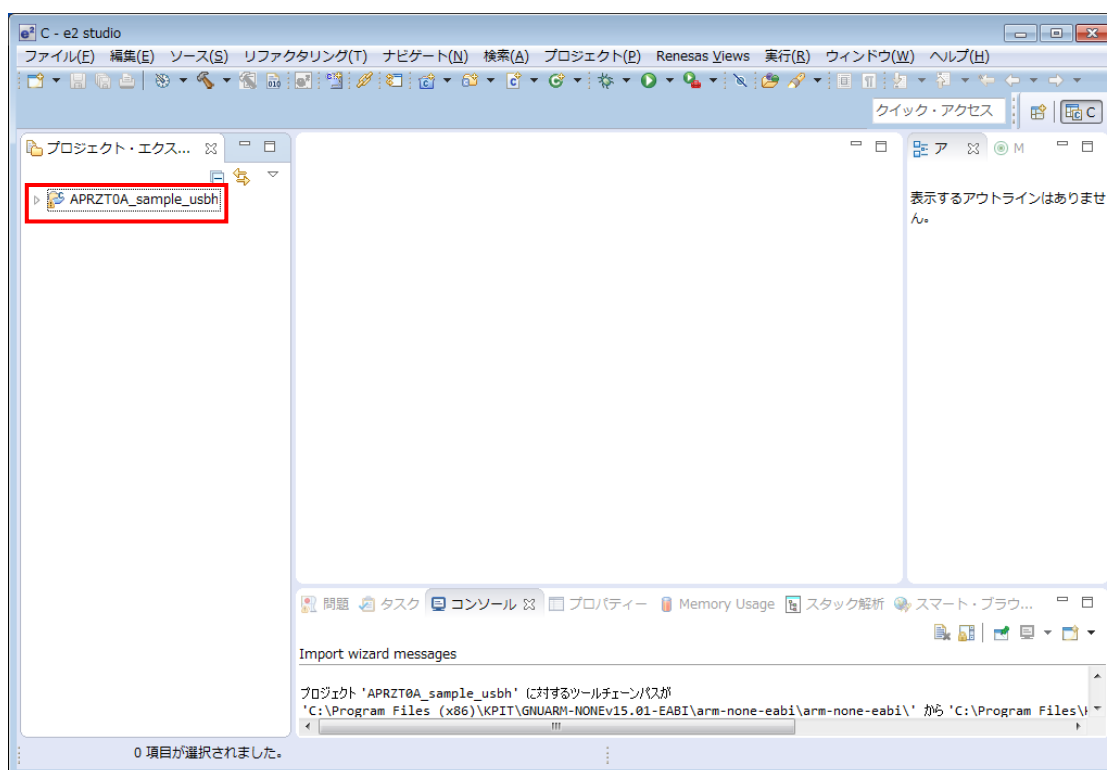


USB ファンクションのサンプルプログラムを使用される場合は[APRZT0A_sample_usbh]にもチェックを入れてください。USB ファンクションのサンプルプログラムについてはアプリケーションノート「AN1613 AP-RZT-0A USB ファンクション サンプルプログラム (GCC) 解説」を参照してください。

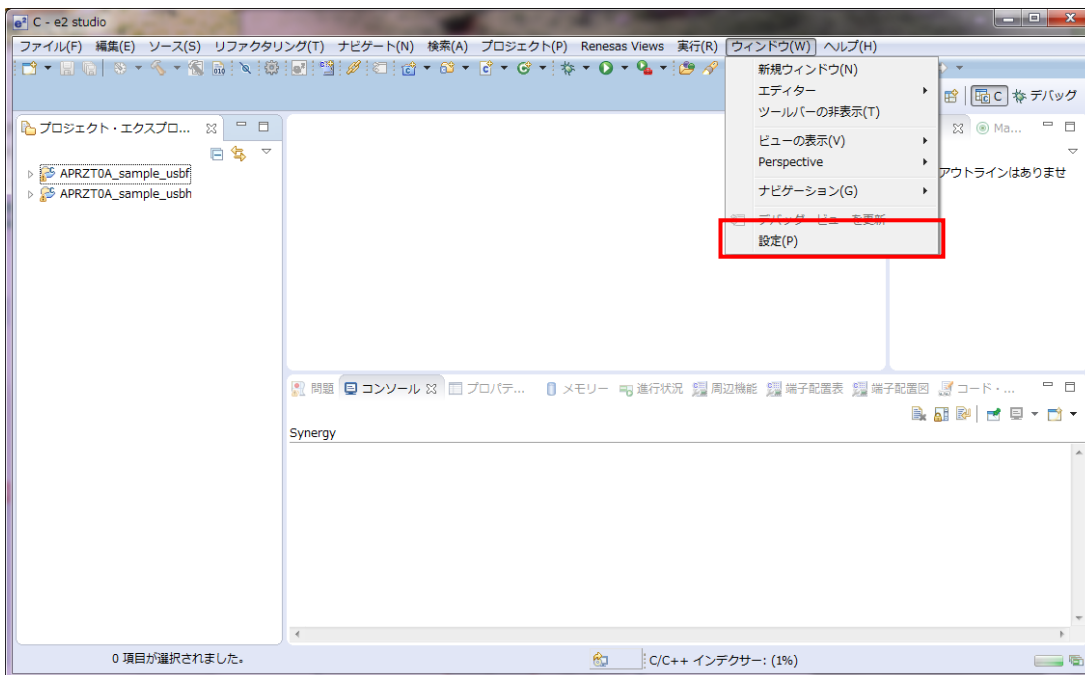
- ⑤ ツールチェーンのパスが変更された場合は、[OK]を選択してください。



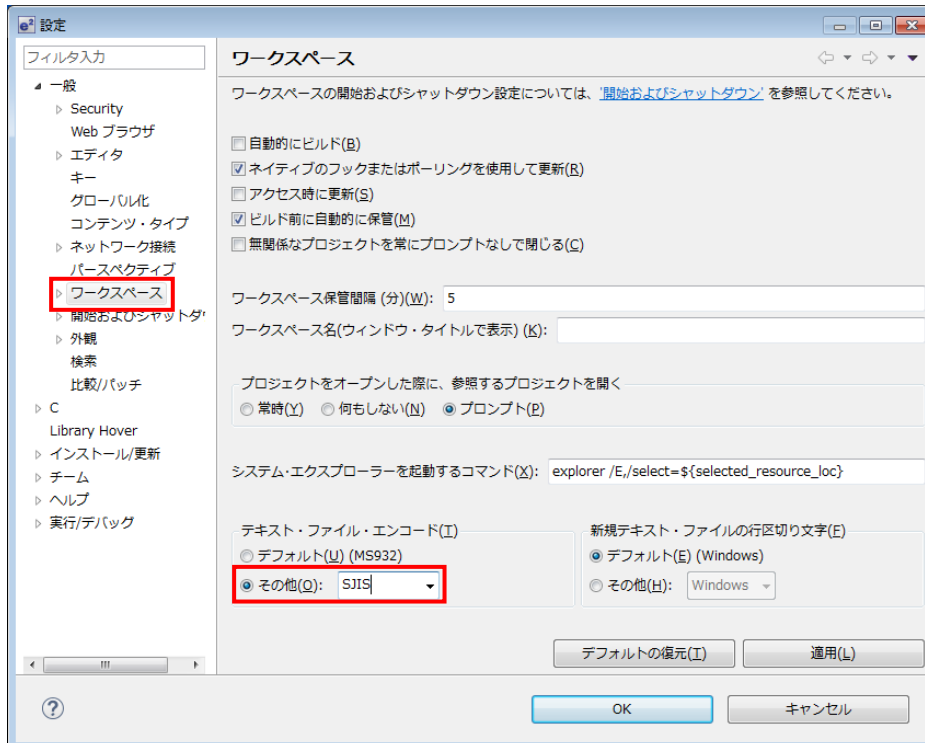
- ⑥ ナビゲータウィンドウにサンプルプログラムのプロジェクトが追加されていることを確認します。



- ⑦ 文字コードの変更を行います。メニューバーの[ウィンドウ]→[設定]を選択します。



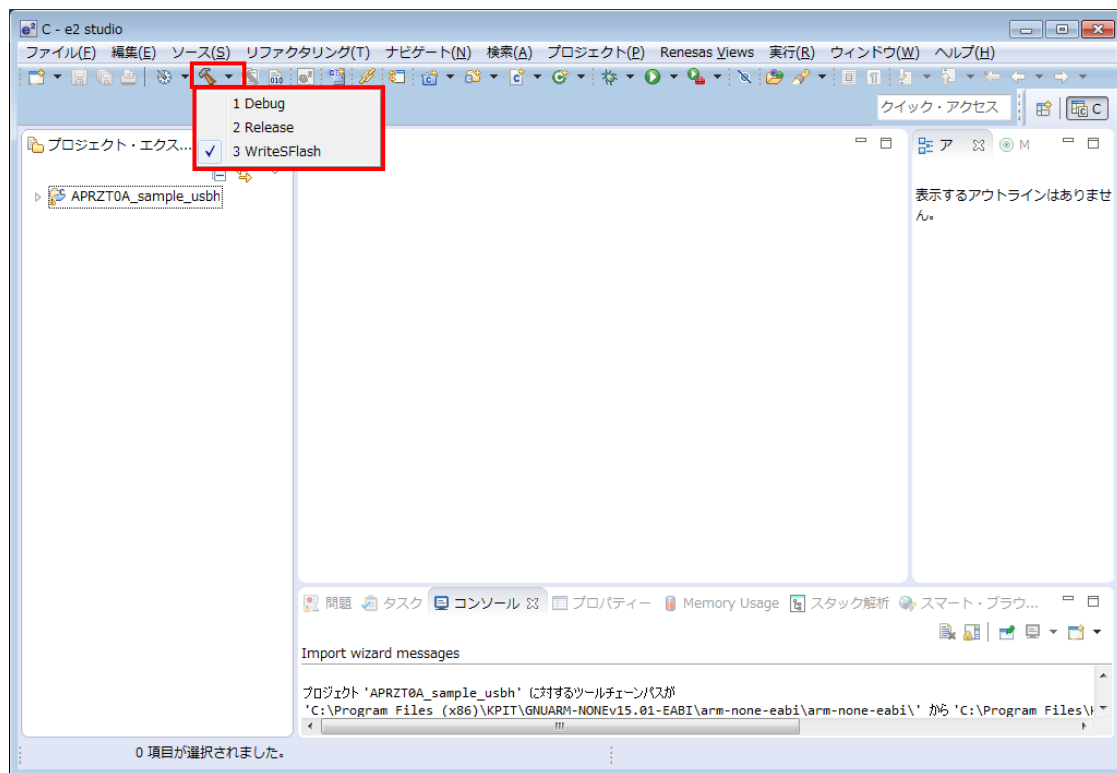
- ⑧ [一般]→[ワークスペース]を選択し、テキスト・ファイル・エンコードをその他にし、[SJIS]を選択します。(SJIS が選択できない場合は直に[SJIS]と打ち込んでください)



3.3.2 プロジェクトのビルド

- ① ツールバーからビルドアイコンを選択します。

アイコン横のした矢印をクリックすることでコンフィグレーションごとのビルドが可能です。

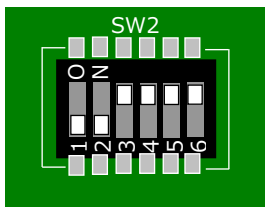


構成が Debug の場合、¥Debug ワークフォルダ内に RAM 動作用の実行可能ファイルが、
Release の場合、¥Release ワークフォルダ内にシリアル FlashROM 動作用のバイナリファイルが、
WriteSFlash の場合、¥WriteSFlash ワークフォルダ内にシリアル FlashROM 書き込み用の実行可能ファイルが
生成されます。

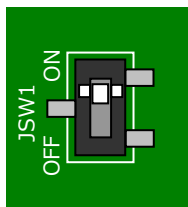
e² studio の詳細な使用方法に関しては e² studio のマニュアルを参照してください。

3.3.3 RAM 動作

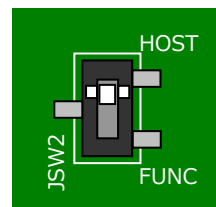
- ① まずは前節[プロジェクトのビルド]で、プロジェクトを Debug 構成にてビルドしてください。
- ② ボード上のディップスイッチを以下に示すように設定してください。



32 ビットバスブートモード
通常動作モード
水晶振動子入力

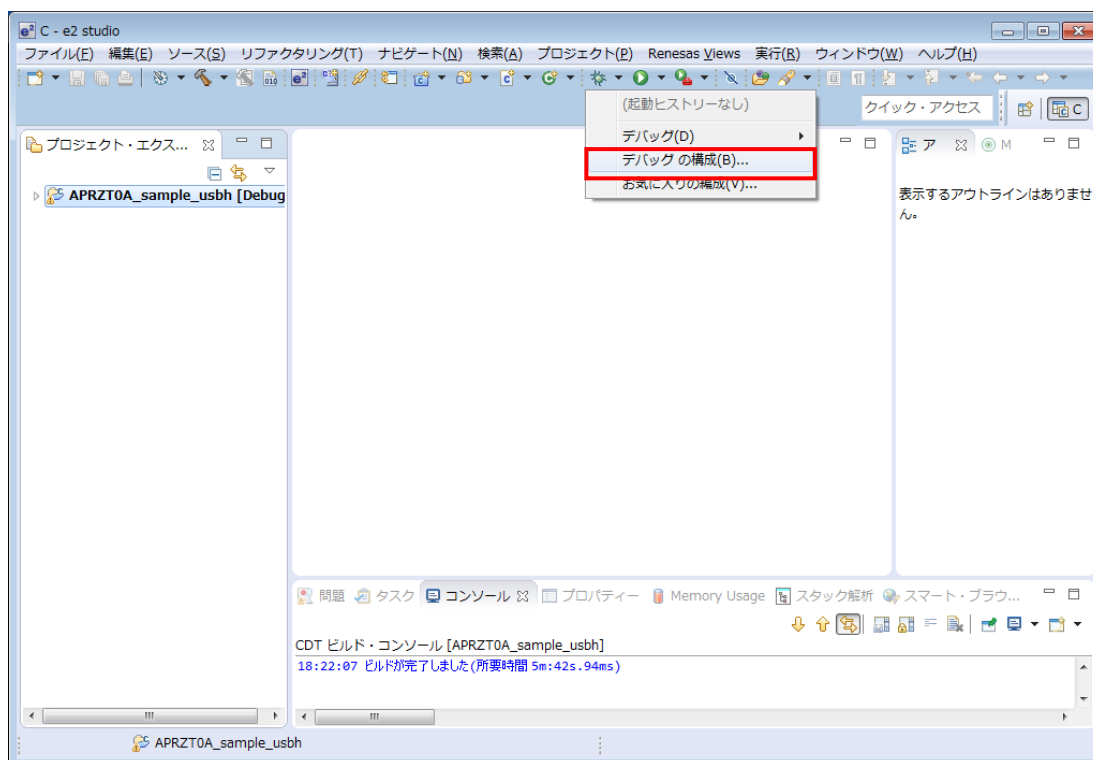


ボード上の SDRAM を使用する

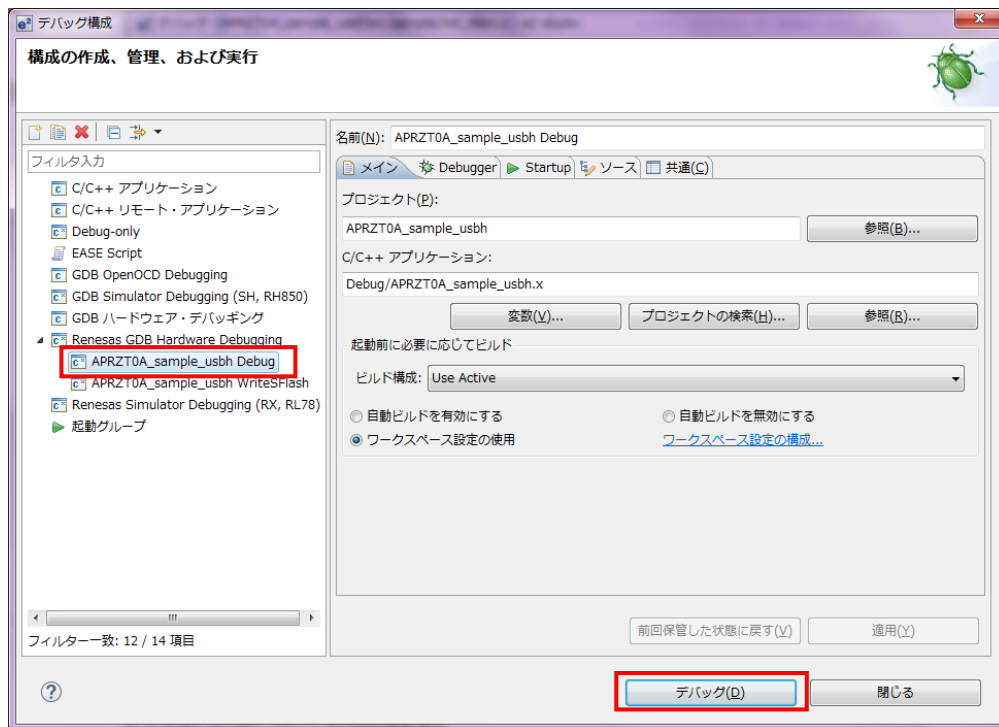


USB ホストで動作

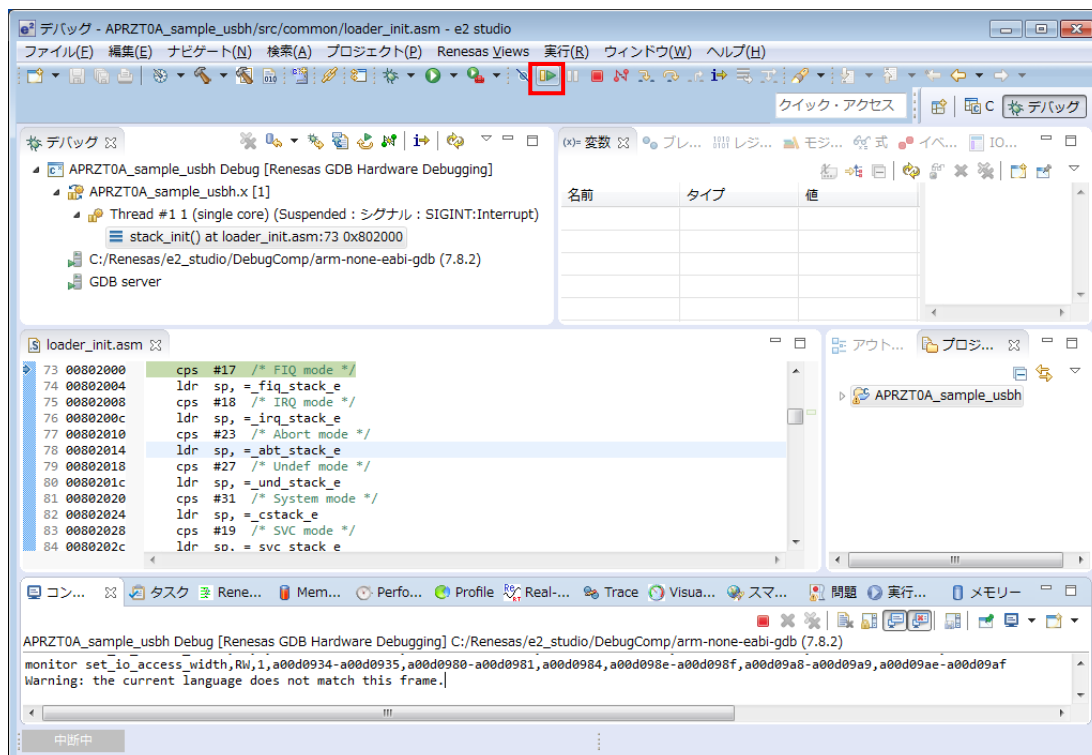
- ③ ボードに電源を投入してください。
- ④ メニューバーから[実行]→[デバッグの構成]を選択します。



- ⑤ [APRZT0A_sample_usbh Debug]をクリックし、[デバッグ]を選択してください。

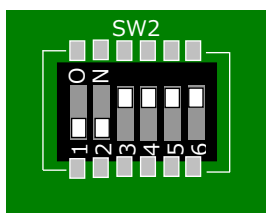


- ⑥ ボードとの接続が完了したらプログラムを実行し、サンプルプログラムを動作させてください。

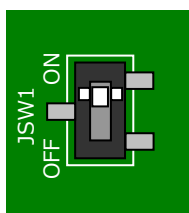


3.3.4 シリアル FlashROM 動作

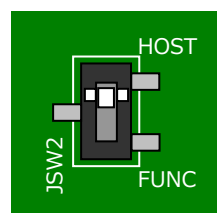
- ① まずは前節[プロジェクトのビルド]で、プロジェクトを Release 構成にてビルドしてください。
- ② [init_main.c]ファイルの 65 行目にある[USER_PRG_SIZE]を Release 構成にてビルドしたバイナリファイルのサイズに変更してください。
 なお、弊社のサンプルプログラムをそのまま動作させる場合は変更する必要はありません。
 お客様がサンプルプログラムを修正された場合は、生成されたバイナリファイルおよびマップファイルを参考に、[USER_PRG_SIZE]を変更してください。
- ③ 続いて前節[プロジェクトのビルド]で、プロジェクトを WriteSFlash 構成にてビルドしてください。
- ④ ボード上のディップスイッチを以下に示すように設定してください。



32 ビットバスブートモード
通常動作モード
水晶振動子入力

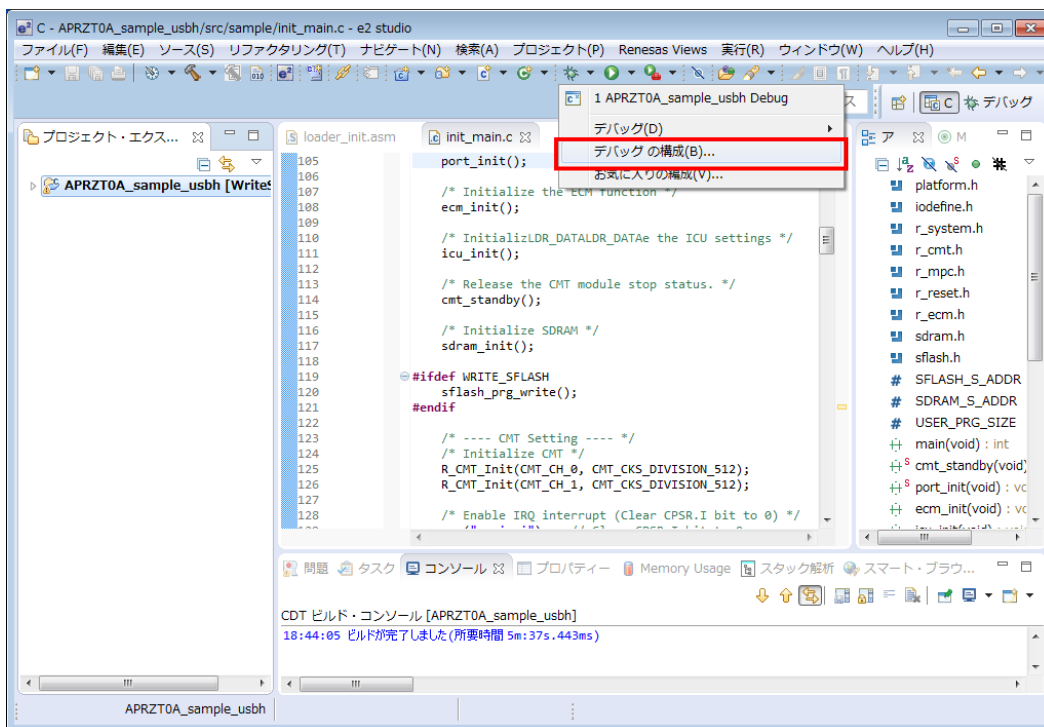


ボード上の SDRAM を使用する

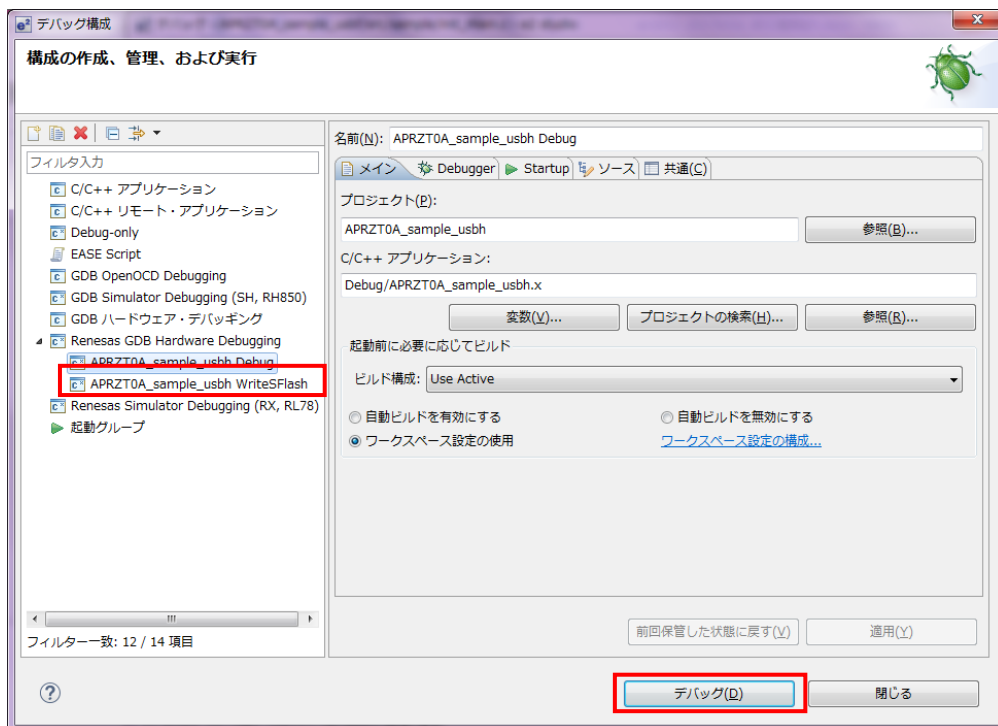


USB ホストで動作

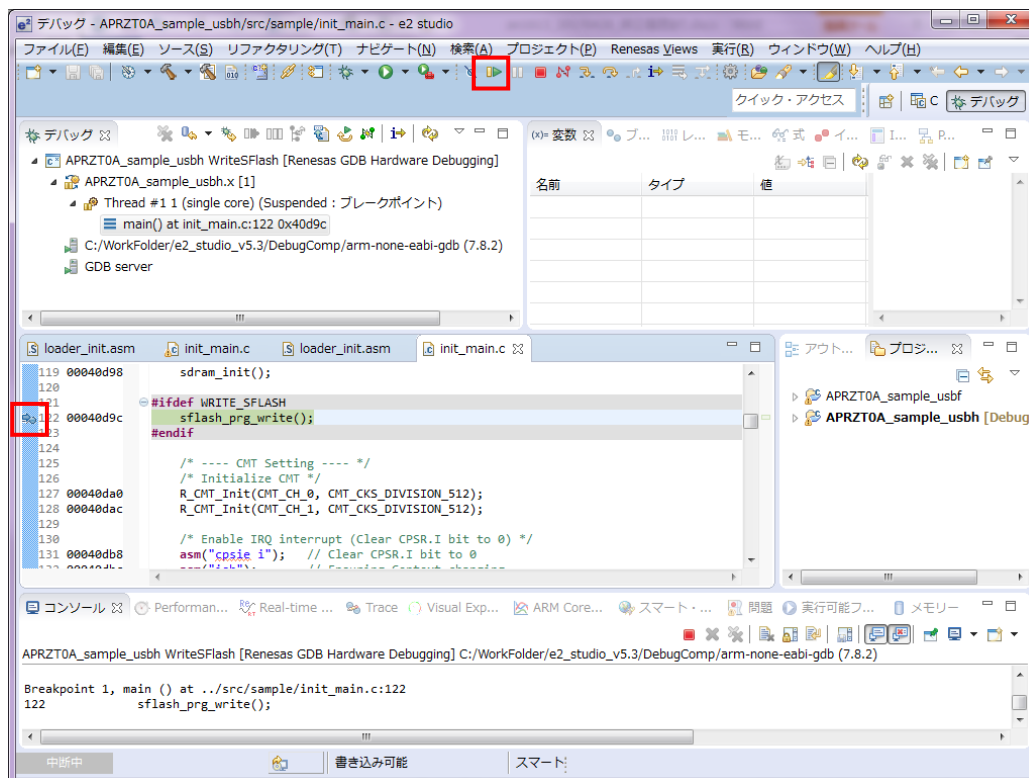
- ⑤ ボードに電源を投入してください。
- ⑥ メニューバーから[実行]→[デバッグの構成]を選択します。



⑦ [APRZT0A_sample_usbh WriteSFlash]をクリックし、[デバッグ]を選択してください。

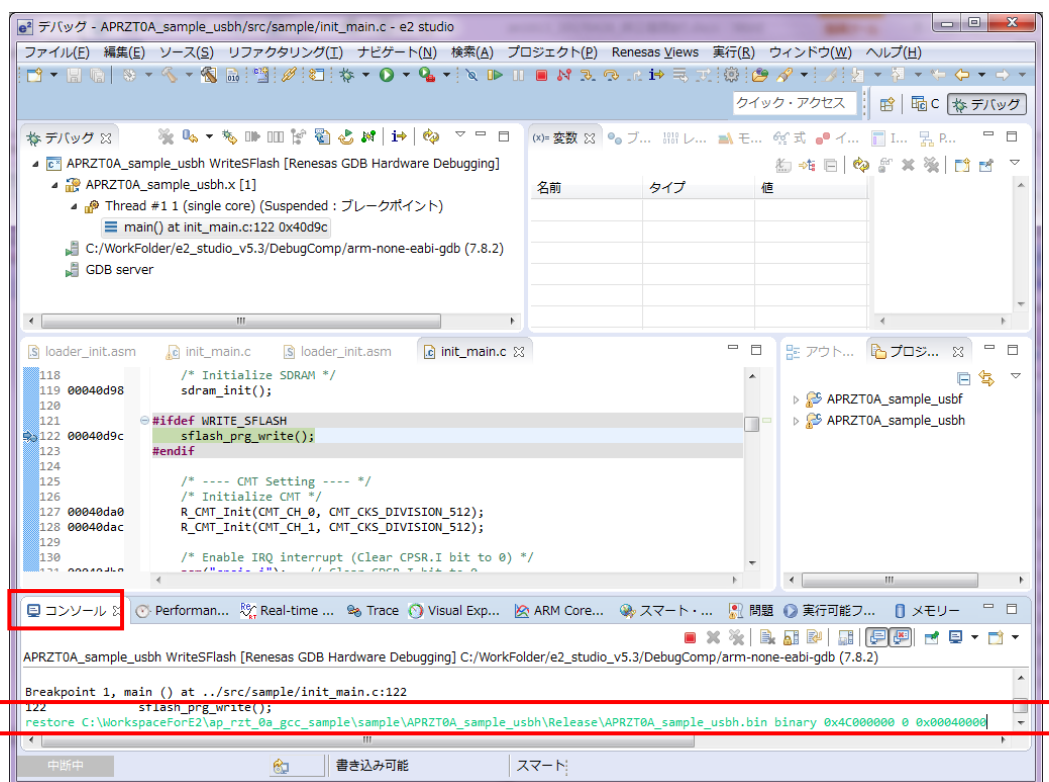


⑧ ボードとの接続が完了したら[init_main.c]の 129 行目の左隅をダブルクリックすることでブレークポイントを設定し、プログラムを実行してください。

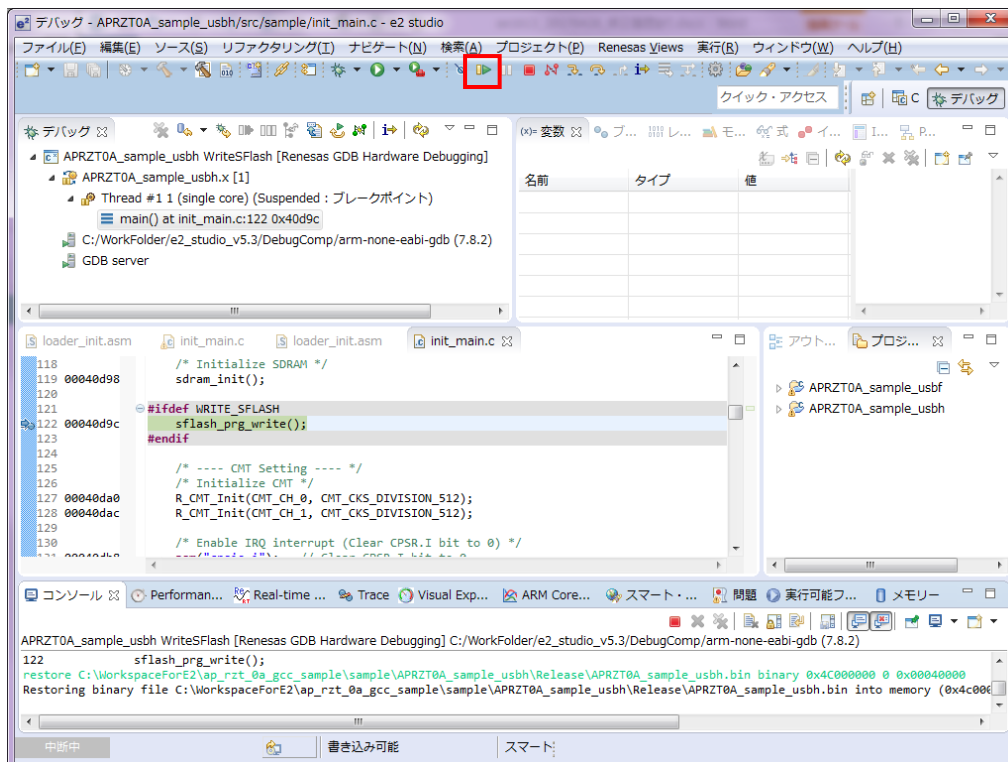


- ⑨ ブレークポイントで停止したらボード上の SDRAM に Release 構成で生成したバイナリファイルを展開します。
 [コンソール] (e² studio V6.0.0 以降では[Debugger Console]) から
 [restore filename binary bias start end]と入力しエンターを押します。
 (下線が引かれた項目については以下を参考に、ユーザの環境に合わせて入力してください。)

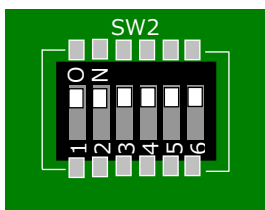
| | |
|----------|--|
| filename | Release 構成で生成したバイナリファイルのパスを指定してください。 |
| bias | SDRAM の先頭アドレス[0x4C000000]を指定してください。 |
| start | [0]を指定してください。 |
| end | [init_main.c]ファイルの 65 行目にある[USER_PRG_SIZE]を指定してください。 |



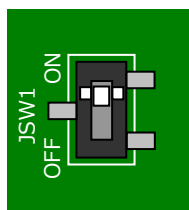
- ⑩ SDRAM ヘバイナリファイルの展開が完了したら、再びプログラムを実行してシリアル FlashROM へプログラムを書き込んでください。書き込みが完了するとボード上の LD4 と LD5 が点灯します。



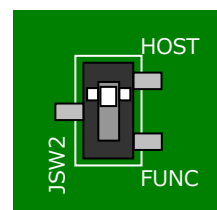
- ⑪ LD4 と LD5 が点灯したら[終了]アイコンをクリックし、ボードとの接続を切り、電源を切ってください。電源が切れたらボード上のディップスイッチを以下に示すように設定してください。



SPI ブートモード
通常動作モード
水晶振動子入力



ボード上の SDRAM を使用する



USB ホストで動作

- ⑫ 再びボードへ電源を投入することで、シリアル FlashROM に書き込まれたサンプルプログラムが動作することを確認してください。

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内容についての質問等のサポートは一切受け付けており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。万が一不審な点、誤りなどお気づきの点がありましたら弊社までご連絡ください。
- ・本サンプルプログラムに関して、ルネサス エレクトロニクス株式会社への問い合わせはご遠慮ください。
- ・本文書の内容に基づき、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

- ・e² studio、RZ および RZ/T1 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。



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