# AP-RZA3-0A (RZ/A3UL CPU BOARD) IPL サンプルプログラム解説

第1.1版 2023年10月02日

## 1. 概要

## 1.1 概要

本アプリケーションノートでは、AP-RZA3-0A に付属する「Initial Program Loader (以下、IPL) 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。

サンプルプログラム	動作内容
IPL サンプルプログラム	・SPI Flash ブートの動作



## 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-RZA3-0A 製品ページ https://www.apnet.co.jp/product/rza/ap-rza3-0a.html

1.3 ライセンスについて

本サンプルプログラムは、ルネサス エレクトロニクス社が公開する「Initial Program Loader (IPL) for Renesas RZ/A Series」 (IPL v1.2.0)を AP-RZA3-0A 用に移植して作成されています。

rza-initial-program-loader 公開ページ https://github.com/renesas/rza-initial-program-loader

本サンプルプログラムは MIT ライセンス等のソフトウェアライセンスが適用されています。 ライセンスについてはプログラム 内の 「sample¥ap\_rza3\_0a\_ipl\_sample¥docs¥license.rst」 をご参照ください。

### 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。 本サンプルプログラムに対応する開発環境、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
e2studio	2023-01	-
GCC	10.3.1.20210621	A-Profile AArch64 bare-metal
MSYS2	2023/03/18 版	

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-RZA3-0A と J-Link を直接接続することはできません。 AP-RZA3-0A 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。 本地マグプタについては、1 Link 即時たるご確認ください

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。 ご使用のワークスペースにコピーして使用してください。

サンプルプログラム	フォルダ
IPL サンプルプログラム	¥sample¥ap_rza3_0a_ipl_sample
プロジェクトフォルダ	

## 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以下のフォルダで構成されています。 なお、フォルダ数が多いため一部を省略して記述します。

¥ sample	AP-RZA3-0A サンプルプログラムフォルダ
¥ ap_rza3_0a_ipl_sample	IPL サンプルプログラムフォルダ
- ¥ .settings	設定ファイルフォルダ
— ¥ bl1	BL1 フォルダ
— ¥ bl2	BL2 フォルダ
— ¥ bl2u	BL2u フォルダ
— ¥ bl31	BL31 フォルダ
— ¥ build	ビルドフォルダ
└──¥ a3ul	AP-RZA3-0A ビルドフォルダ
- ¥ debug	AP-RZA3-0A デバッグビルドフォルダ
¥ release	AP-RZA3-0A リリースビルドフォルダ
— ¥ common	共通フォルダ
— ¥ docs	資料フォルダ
— ¥ drivers	ドライバフォルダ
— ¥ fdts	FDTs フォルダ
— ¥ include	インクルードフォルダ
— ¥ lib	ライブラリフォルダ
— ¥ licenses	ライセンスフォルダ
— ¥ make_helpers	Make 補助フォルダ
— ¥ plat	プラットフォームフォルダ
└─ ¥ renesas	ルネサス社製 CPU 用プラットフォームフォルダ
— ¥ services	サービスコールフォルダ
L ¥ tools	ツールフォルダ

## 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	च .
なお、ファイル数が多いため一部を省略して記述	します	- •
<¥sample¥ap_rza3_0a_ipl_sampleフォルダP	内>	
.cproject		CPROJECT ファイル
.project		PROJECT ファイル
ap_rza3_0a_ipl_sample Debug.launch		AP-RZA3-0A IPL サンプルプログラム
		デバッグおよびランタイム設定ファイル
readme.rst		リードミーファイル
<¥sample¥ap rza3 0a ipl sample¥build¥a3	3ul¥de	bug フォルダ内>
rza3ul ap rza3 0a ipl.bin		- サンプルプログラム デバッグビルド バイナリファイル
rza3ul_ap_rza3_0a_ipl.elf		サンプルプログラム デバッグビルド ELF ファイル
rza3ul_ap_rza3_0a_ipl.map		サンプルプログラム デバッグビルド マップファイル
rza3ul_ap_rza3_0a_ipl.srec		サンプルプログラム デバッグビルド モトローラSファイル
<pre><ycampleyan 0a="" inl="" pre="" rza3="" sampleyhuildya3<=""></ycampleyan></pre>	Rul¥rol	eace フォルズ広ン
rza3ul an rza3 0a inl hin		サンプルプログラム リリースビルド バイナリファイル
$r_2a_3ul_ap_r_2a_3_0a_iplebilit$		
$r_2a_3ul_ap_12a_3_0a_ipl.en$		
rza3ul ap rza3 0a inl srec		サンプルプログラム リリースビルド モトローラ S ファイル
·		
<¥sample¥ap_rza3_0a_ipl_sample¥plat¥ren	iesas¥	rz¥board¥a3ul_ap_rza3_0a フォルダ内>
rz_board.mk		AP-RZA3-0A 用 make ファイル
<¥sample¥ap rza3 0a ipl sample¥plat¥ren	ıesas¥	rz¥common¥drivers¥ddr フォルダ内>
ddr.c		DDR 共通ファイル
param_swizzle_ap_rza3_0a.c		AP-RZA3-0A DDR ファイル
<¥sample¥an rza3 0a inl sample¥plat¥ren	nesas¥	rz¥common¥drivers¥xspidevice¥octaflash mx25um フォルダ内>
octaflash_mx25um_c		AP-RZA3-0A OctaElashROM ドライバファイル
octaflash mx25um api.h		AP-RZA3-0A OctaElashROM ドライバ API ファイル
rz_xspidevice.mk		AP-RZA3-0A OctaFlashROM ドライバ用 make ファイル
<¥sample¥ap_rza3_0a_ipl_sample¥plat¥ren	iesas¥	rz¥soc¥a3ul¥drivers¥ddr フォルダ内>
ddr_a3ul.c		RZ/A3UL DDR レジスタ制御ファイル
param_mc_ap_rza3_0a.c		AP-RZA3-0A DDR 設定ファイル

## 3. 動作説明

3.1 サンプルプログラムの動作

本サンプルプログラムは、SPI Flash ブート時に最初に起動するプログラムとして働き、ユーザアプリケーションプログラム(以 下、ユーザプログラム)を起動するための動作を行います。

本サンプルプログラムは主に以下の動作をします。

- 1 DDR SDRAM の初期化します。
- シリアル FlashROM 空間内のアドレス「0x20020000」から 480Byte のデータを読み出します。 2 480Byteのデータは以下で構成されるユーザプログラムのヘッダ情報として扱います。

オフセット	サイズ (Byte)	説明
+0x00	8	ユーザプログラムのエントリアドレス
+0x08	8	「ユーザプログラムのエントリアドレス」の逆数パラメータ
+0x10	8	ユーザプログラムの転送先アドレス
+0x18	8	「ユーザプログラムの転送先アドレス」の逆数パラメータ
+0x20	8	ユーザプログラムのサイズ
+0x28	8	「ユーザプログラムのサイズ」の逆数パラメータ
+0x30	432	ダミーデータ(0x00 埋め)
+0x1E0	32	文字列「RZ FSP Application」(残領域は 0x00 埋め)

3 読み出した 480Byte の各パラメータについて、逆数パラメータやプログラム内文字列と比較することでデータに異常が 無いかを確認します。

データが異常であった場合、処理を終了します。

- シリアル FlashROM 空間内のアドレス「0x20020200」から「ユーザプログラムのサイズ」の量だけ「ユーザプログラ **(**4**)** ムの転送先アドレス」にメモリデータを転送していきます。
- RAM 空間に転送したユーザプログラムヘジャンプし、ユーザプログラムの動作に移ります。 (5)

本サンプルプログラムの動作については、ルネサス社の「Initial Program Loader (IPL) for Renesas RZ/A Series」と、資 料「Initial Program Loader Application Note (Octa Flash Boot Edition)」も併せてご参照ください。

3.2 サンプルプログラムのダウンロード

本サンプルプログラムは上述した通り、SPI Flash ブート時に最初に起動し、シリアル FlashROM 空間の指定アドレスに書き 込まれたユーザプログラムを任意の RAM 空間にコピーし、実行するプログラムです。

本サンプルプログラム単独で動作を確認することはできず、ユーザプログラムをあらかじめシリアル FlashROM へ書き込んでおくことで動作確認ができます。

本節では IPL サンプルプログラムを CPU ボードヘダウンロードして実行するための手順を説明します。

## 3.2.1 サンプルプログラムのビルド

「AN1653 RZ/A3UL CPU BOARD 開発チュートリアル」を参考に、本サンプルプログラムを e2studio にインポートしてビルドしようとした場合、デバッグ設定が 3 種類用意されています。

それぞれの設定について説明します。



デバッグモード	説明
Debug	デバッグ情報あり(DEBUG=1)でビルドします。
	IPL サンプルプログラムのデバッグをする場合に使用します。
Release	デバッグ情報なし(DEBUG=0)でビルドします。また、デバッグ用機能を無効
	化することで本サンプルプログラムのサイズや処理速度を最適化します。
	IPL サンプルプログラムのリリース時に使用します。
デフォルト	「Debug」と同じ。



e2studioのインストール時に、インストール先をデフォルト設定から変更した場合、本プロジェクトをビルドすると下記のようなエラーメッセージが表示されることがあります。



上記のようなエラーメッセージが表示された場合、以下の手順に従ってプロジェクト設定を変更してください。



① 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。

 ② プロパティウィンドウから変更が必要なビルド設定を選び(例として Debug ビルド)、「C/C++ビルド」>「環境」を選 択したうえで、「PATH」を選択して「編集」ボタンを押します。

圆 プロパティ: ap_rza3_0a_ipl_samp	le			
7ብሥልን አ	環境		¢	▼ □> ▼ 8
> リソース Arm Tealsheing Paths				
✓ C/C++ ビルド	Configuration: Debug		× :	構成の管理
ノールテェイノ・エティター ビルド変数				
ロビング	設定する環境変数			追加
設定	変数	値	由来	and the
> C/C++ 一般	CROSS_COMPILE	aarch64-none-elf-	ユーザー: 構成	A22.0 (
Renesas QE	CWD	E:¥workspace¥e2studio_ws¥2022-10_rzafsp_v1.2.0	ビルド・システム	編集
Task Tags	MSVSTEM	MSYS	ユーザー: 構成	削除
> Validation	PATH	C:¥msys64¥usr¥bin;C:¥Renesas¥rza¥e2studio_v202	ユーザー: 構成	H11MA
ビルツー プロジェクト・ウーチャー	PWD	E:¥workspace¥e2studio_ws¥2022-10_rzafsp_v1.2.0	ビルド・システム	定義解除
プロジェクト・ホーフャー				
実行/デバッグ設定				
	● ネイティブ環境へ変数を追加	מל		
	○ ネイティブ環境を指定された	環境と置換		
			ニュートの復去の	**田(1)
			テノオルトの復元(1)	迥州( <u>L</u> )
?			適用して閉じる	キャンセル

#### ③ 「値」に含まれる内容を下記の赤字の通りに編集して、OK ボタンを押します。

3	数の編集	×
名前:	PATH	
值:	C:¥msys64¥usr¥bin;C:¥Renesas¥rza¥e2studio_v202.	変数
	ОК	キャンセル

C:¥msys64¥usr¥bin;C:¥Renesas¥rza¥e2studio\_v2023-01\_fsp\_v1.2.1¥toolchains¥gcc\_arm\_aarch64¥10.3-2021.07¥gcc-arm-10.3-2021.07-mingw-w64-i686-aarch64-none-elf¥bin;

 $\downarrow \downarrow \downarrow \downarrow$ 

C:¥msys64¥usr¥bin; [ユーザが e2studio をインストールしたフォルダパス] ¥toolchains¥gcc\_arm\_aarch64¥10. 3-2021.07¥gcc-arm-10.3-2021.07-mingw-w64-i686-aarch64-none-elf¥bin;

※使用するコンパイラのバージョンが異なる場合は、青字部分の変更も必要です。使用する環境に従ってコンパイラのパスを修正してください。

④ 「適用して閉じる」を押し、プロパティウィンドウを閉じ、再度ビルドをお試しください。



#### 3.2.2 サンプルプログラムのデバッグ

本サンプルプログラムをデバッグする場合は上述の通り、「Debug」ビルドでなければデバッグ情報が出力されません。 必ず「Debug」ビルドを行った後、以下の順にデバッグを行ってください。

① ボード上のディップスイッチを設定します。



<sw2 設定=""></sw2>	
SSCG 設定	: 不問
ブートモード	: SPI Flash ブートモード(OFF)
デバッグ設定	: デバッグモード(OFF)

- ② ボードに電源を投入します。
- ③ 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。



 ④ [Renesas GDB Hardware Debugging]のデバッグ設定から [ap\_rza3\_0a\_ipl\_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\_rza3\_0a\_ipl\_sample Debug
- [プロジェクト]: ap\_rza3\_0a\_ipl\_sample
- [C/C++アプリケーション]: build/a3ul/debug¥rza3ul\_ap\_rza3\_0a\_ipl.elf

🚳 デバッグ構成		- <b>D</b> X
構成の作成、管理、および実行		The second se
	名前( <u>N</u> ): ap_rza3_0a_ipl_sample Debug	
フィルタ入力	📄 メイン 🏇 Debugger 🐌 Startup 🧤 ソース	Ⅱ 共通(C)
C/C++ アプリケーション	プロジェクト( <u>P</u> ):	
C/C++ UE-F· P/UU-SEP     EASE Script	ap_rza3_0a_ipl_sample	参照( <u>B</u> )
C GDB OpenOCD Debugging	C/C++ アプリケーション:	
CP GDB Simulator Debugging (RH850) C GDB ハードウェア・デバッギング	build/a3ul/debug/rza3ul_ap_rza3_0a_ipl.elf	
Java アブリケーション     Java アブリケーション     For Java アブリット     C Renesas GDB Hardware Debugging     C ap.rza3_06_jpl_sample Debug	起動前に必要に応じてビルド <u>Build Configuration:</u> Select Automatically	変数( <u>V</u> ) プロジェクトの検索( <u>H</u> ) 参照( <u>R</u> )
Renesas Simulator Debugging (RX, RE78)	○ 自動ビルドを有効にする	○ 自動ビルドを無効にする
eg ジェード Java アフリゾージョン 電 起動グループ	● ワークスペース設定の使用	<u>ワークスペース設定の構成</u>
15 項目のうち 13 項目がフィルターに一致		前回保管した状態に戻す(火) 適用(火)
?		デパッグ( <u>D</u> ) 閉じる

⑤ [Debugger] タブを選択し、[Debug hardware] を [J-Link ARM]、

[Target Device] に「R9A07G063U02GBG」、[GDB コマンド]に「aarch64-elf-gdb」がそれぞれ設定されていることを確認してください。

💽 デバッグ構成	- D X
構成の作成、管理、および実行	- the second sec
○       ○       ○       ○       ○         ○       C/C++ アブリケーション       ○       ○       ○       ○         ○       C/C++ リモート・アブリケーション       ○       EASE Script       ○	名前(W): [ap_rza3_0a_ipl_sample Debug ● パ ● Debugger ● Startup 「シリース □ 共通① Debug hardware:  -Link ARM ▼ Target Device: R9A07G063U02GBG GDB Settings Connection Settings デパッグ・ツール設定 GDB 接続設定: ● ローカル GDB サーバーを自動起動 ホスト名または IP アドレス: localhost 〇 リモート GDB サーバーを自動起動 ホスト名または IP アドレス: localhost 〇 リモート GDB サーバーへ接続 GDB ボート番号: 61234 接続タイムアウト(s): 30 ▼ GDB コマンド: aarch64-elf-gdb ● Step Mode 面回保生II.た大坂(三本40) ④ 田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田
15 項目のうち 13 項目がフィルターに一致	
?	デパッグ( <u>D</u> ) 閉じる

デバッグ設定を新規作成した場合などに、[Target Device] に「R9A07G063U02GBG]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。

(国) デバッグ構成			_	
<mark>構成の作成、管理、および主行</mark> ⑧ Run Break Time Measurement Operating Frequency mus	t not be empty and must be between 0.001 & 10000.0MH	iz.		Ť.
<ul> <li>○ ○ ○ ○ × ○ × ○</li> <li>○ (C++ ) ブリケ-ション</li> <li>○ C(C++ ) ブーリケ-ション</li> <li>○ ○ ○ C(C++ ) ブート・ブリケ-ション</li> <li>○ GDB Simulator Debugging</li> <li>○ GDB Simulator Debugging (RH850)</li> <li>○ GDB (-トラコア・デ(ペギング)</li> <li>○ Java アブリケ-ション</li> <li>○ Java アブリケ-ション</li> <li>○ ap.rza3 (0a.jpl sample Debug</li> <li>○ Renesas Simulator Debugging (RX, RL78)</li> <li>○ リート Java アブリケ-ション</li> <li>■ 起動グルーブ</li> </ul>	<ul> <li>名前(M): ap_rza3_0a_ipl_sample Debug</li> <li>メイン ※ Debugger ▶ Startup は ソース 二 共</li> <li>Debug hardware: J-Link ARM ▼ Target Device:</li> <li>GDB Settings Connection Setting デパッグ・ルージ</li> <li>Semihosting breakpoint address</li> <li>× RTOS</li> <li>デパッグピューにRTOSのスレッド情報を表示する 多数のスレッドを使用するプロジェクトをデパッグする</li> <li>&gt; System Allow caching of flash contents</li> <li>&gt; 時間計測 プレークまでの実行時間を計測する</li> <li>Count Every Core Cycle Operating Frequency [MHz]</li> </ul>	通(C) R9A07G063U02GBG 防定 はい いいえ はい はい はい		
15 項目のうち 13 項目がフィルターに一致			前回保管した状態に戻す(⊻)	適用( <u>Y</u> )
?			デバッグ( <u>D</u> )	閉じる

#### ⑥ [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。

📴 デバッグ構成					— 🗆 X
構成の作成、管理、および実行					TO-
	名前( <u>N</u> ): ap_rza3_0a_ipl_sample D	ebug			
フィルタ入力	📄 メイン 🏇 Debugger ⊳ Star	up 🧤 ソース 🔲 共	通( <u>C</u> )		
C/C++ アプリケーション	初期化コマンド				^
■ C/C++ リモート・アプリケーション	□リセットと遅延(秒): 0				
GDB OpenOCD Debugging	Halt				
C GDB Simulator Debugging (RH850)					^
© GDB ハードウェア・テバッキンク ■ Java アプリケーション					
I Java アプレット					<u> </u>
✓ ☑ Renesas GDB Hardware Debugging	イメージとシンボルをロード				
Renesas Simulator Debugging (RX, RL78)	ファイル名	ロード・タイプ	オフセット	接続時	追加
型 リモート Java アプリケーション	✓ プログラム・バイナリー [rza3	イメージとシンボル	0	Yes	
┃   ■ 起動グループ					福未 …
					除去
15 項目のうち 13 項目がフィルターに一致					前回保管した状態に戻す(火) 適用(火)
?					デバッグ( <u>D)</u> 閉じる

「ワークスペース」を押し、サンプルプログラムの「build¥a3ul¥debug¥rza3ul\_ap\_rza3\_0a\_ipl.srec」を選択してウィンドウを閉じてください。

😰 ダウンロード・モジュールの追加			×
ダウンロード・モジュール名の指定:			
赤粉 プロパックしの絵書		7=71.20	1=1
変数 フロンエクトの使来	リークスペース	ליערייל.	λτ <u>μ</u>
	OK	キャン	セル
📴 ダウンロード・モジュールの編集			$\times$
ワークスペースのリソースを選択			
x 🕰 an rza3 0a int sample			•
• ep_reas_od_pr_sample			
application note			
> 🕞 bl1			
> 🕞 bl2			
> 🕞 bl2u			
> 🕞 bl31			
> 🗁 bl32			
V 🕞 build			
v 🅞 a3ul			
V 🗁 debug			
, 🗁 biz			
> 🗁 lib			
> 🗁 libc			
🔁 libwrapper			
🗁 romlib			
😹 bl2.bin			
rz_bl2_xspi_def.c	:		
rz_bl2_xspi.env			
Ra3ul_ap_rza3_0	Da_ipl.bin		
📄 rza3ul_ap_rza3_0	Da_ipl.dump		
🗟 rza3ul_ap_rza3_0	Da_ipl.elf		
🗒 rza3ul_ap_rza3_(	a_ipl map		
🖺 rza3ul_ap_rza3_0	Da_ipl.srec		~
L			
?	ОК	キャンセ	JL
$\odot$	OK	7172	N

⑧ 追加した「rza3ul\_ap\_rza3\_0a\_ipl.srec」の接続時設定を「No」に設定します。

😰 デバッグ構成							×
構成の作成、管理、および実行						Ŕ	J.
1 🖻 🕫 🗎 🗶 🖻 🍸 🗸	名前( <u>N</u> ): ap_rza3_0a_ipl_sample Debug						
<ul> <li>フイルタ入力</li> <li>C / C(++ アブリケーション)</li> <li>C / C(++ リモート・アブリケーション)</li> <li>E ASE Script</li> <li>E GDB OpenOCD Debugging</li> </ul>	汁ン 珍 Debugger <mark>&gt; Startup</mark> 頃 □ Halt	╱ ソース   <u>□</u> 共通( <u>C</u> )				^	^
<ul> <li>CDB Simulator Debugging (RH850)</li> <li>GDB ハードウェア・デバッギング</li> </ul>	イメージとシンボルをロード					*	
「」 Java アブレット ■」 Java アブレット マ Cel Renesas GDB Hardware Debugging	ファイル名 アゴログラム・バイナリー [rza3ul ap r	ロード・タイプ イメージとシンボル	オフセット o	接続時 Yes	追り	Ođ	
<ul> <li>ご ap_rza3_0a_ipl_sample Debug</li> <li>C Renesas Simulator Debugging (RX, RL78)</li> <li>リモート Java アプリケーション</li> <li>記動グループ</li> </ul>	Image: Image	イメージとシンボル	0	No	編:	€ 法	
					T	~	~
15 項目のうち 13 項目がフィルターに一致				前回保管し	た状態に戻す(⊻)	適用( <u>Y</u> )	
?					デバッグ( <u>D</u> )	閉じる	i

⑨ 「ブレークポイント設定先」を無効化します。

🙆 デバッグ構成						×
構成の作成、管理、および実行					Ť	2
C 2 0 0 0 × 0 √ ·	名前(N): ap_rza3_0a_ipl_sample Debug	a \/7   ■ 共通(O)				
<ul> <li>C (/C++アブリケ-ション</li> <li>C /(C++ アブリケ-ション</li> <li>E (/C++ リモート・アブリケ-ション</li> <li>E EASE Script</li> <li>G OB OpenOCD Debugging</li> <li>G OB Simulator Debugging (RH850)</li> <li>G OB /(-F<sup>5</sup>) 27・デ(マギング</li> <li>Java アブリア-ション</li> <li>Java アブリアト</li> <li>Reneasa GOB Hardware Debugging</li> <li>G ap.rza3_0a.jpl_sample Debug</li> <li>Reneasa Simulator Debugging (RX, RL78)</li> <li>U U U T + Java アブリケーション</li> <li>E 起動 グルーブ</li> </ul>	ファイル・名     ブログラム・オブション     フログラム・オブション     フログラム・オブション     フログラム・カウンター設定先(16進):     □ブレークボイント設定先:     □ブレークボイント設定先:	<ul> <li>ロード・タイプ</li> <li>・イメージとシンボル</li> <li>・イメージとシンボル</li> </ul>	オフセット 0 0	接続時 Yes No	追加… 編集… 除去 上へ 下へ	~
15 項目のうち 13 項目がフィルターに一致				前回保管した	と状態に戻す(⊻) 適用(⊻)	]
?					デバッグ( <u>D)</u> 閉じる	

⑩ 「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。



① ボードとの接続が完了したら「再開」ボタンや「中断」ボタンを使い、サンプルプログラムのデバッグを行ってください。



なお、プログラムを実行すると「3.1 サンプルプログラムの動作」の通り、シリアル FlashROM 内のユーザプログラム を RAM 領域にコピーし、実行します。

「AN1653 RZ/A3UL CPU BOARD 開発チュートリアル」や「AN1658 AP-RZA3-0A シリアル FlashROM の書き込み 方法」を参考にユーザプログラムの書き込みを予め行ってください。

なお、書き込みを行っていない場合は、次の手順を行うことでユーザプログラムを書き込むことができます。

① ボードとの接続が完了した直後の状態で、「ファイルのロード」ボタンを押します。



「ファイルシステム」を選択し、ユーザプログラムのSモトローラファイルを選択します。

下図は AP-RZA3-0A Ethernet サンプルプログラムの Debug ビルドのデータ (ap\_rza3\_0a\_ethernet\_sample¥Debu g¥ap\_rza3\_0a\_ethernet\_sample.srec)を選択した場合のイメージ例です。

3		×
ファイルをロードする		
ロードするファイルを選択してください		
ファイリレ: E:¥workspace¥e2studio_ws¥2022-10_rzafsp_v1.2.0¥ap_rza3_0a_ethernet_sample¥Debug¥ap_rza3_0a_ethernet_sample.srec く)	ワークスペース	ファイルシステム
□ パイナリー・イメージとしてロードする		
アドレス: 0x0000000		
	OK	キャンセル

③ 「バイナリー・イメージとしてロードする」を無効化した状態で「OK」ボタンを押します。

3		×
ファイルをロードする		
ロードするファイルを選択してください		
ファイル: E:¥workspace¥e2studio_ws¥2022-10_rzafsp_v1.2.0¥ap_rza3_0a_ethernet_sample¥Debug¥ap_rza3_0a_ethernet_sample.sree	: > ワークスペース	ファイルシステム
□パイナリー・イメージとしてロードする		
アドレス: 0x00000000		
	ОК	キャンセル

以上の作業をすることで、J-LinkからユーザプログラムをCPUボードのシリアルFlashROMへ書き込みが開始されます。 その後、IPLサンプルプログラムを実行し、書き込んだユーザプログラムが動作することを確認してください。

#### 3.2.3 リリースデータの反映

「Debug」ビルドで本サンプルプログラムのデバッグを終えた後、IPL サンプルプログラムの「Release」ビルドで出力した リリースデータをユーザプログラムと組み合わせるための手順を説明します。

#### ① 「Release」ビルドを行います。

サンプルプログラムの「build¥a3ul¥release」を開き、「rza3ul\_ap\_rza3\_0a\_ipl.srec」「rza3ul\_ap\_rza3\_0a\_ipl.elf」
 をユーザプログラムのプロジェクトヘコピーします。

下図は AP-RZA3-0A Ethernet サンプルプログラム内の IPL フォルダに移動する場合のイメージ例です。



以上で IPL サンプルプログラムのリリースデータがユーザプログラムに反映されました。

「AN1653 RZ/A3UL CPU BOARD 開発チュートリアル」の「3.2.3 デバッグ方法」を参考に、ユーザプログラムの動作確認を行い、CPU リセット解除直後の IPL サンプルプログラムの動作が更新されていることを確認してください。



## 4. 開発環境使用時の各設定値

開発環境を使用する際の、AP-RZA3-0A 固有の設定を以下に示します。

ビルド・動作確認方法	
項目名	設定値
サンプルプログラムフォルダ	sample¥ap_rza3_0a_ipl_sample
プロジェクト	ap_rza3_0a_ipl_sample
デバッグ時のボード設定	「4.1 スイッチ設定」 参照
デバッグ用出力フォルダ	ap_rza3_0a_ipl_sample¥build¥rza3¥debug
デバッグ用実行ファイル	rza3ul_ap_rza3_0a_ipl.elf
Debug hardware	J-Link ARM
Target Device	R9A07G063U02GBG
SerialFlash 書き込み用フォルダ	ap_rza3_0a_ipl_sample¥build¥rza3¥release
書き込みファイル	rza3ul_ap_rza3_0a_ipl.srec



## 4.1 スイッチ設定

・デバッグをする場合		
	<sw2 設定=""> SSCG 設定 ブートモード デバッグ設定</sw2>	: 不問 : SPI Flash ブートモード(OFF) : デバッグモード(OFF)
・SPI Flash ブートをする場合		
	<sw2 設定=""> SSCG 設定 ブートモード デバッグ設定</sw2>	: 不問 : SPI Flash ブートモード(OFF) : ノーマルモード(ON)
・SD ブートをする場合		
	<sw2 設定=""> SSCG 設定 ブートモード デバッグ設定</sw2>	: 不問 : SD ブートモード : ノーマルモード(ON)

Fig4.1-1 デバッグ・各ブート時のボード設定

## 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 および RZ/A3UL 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<sup>®</sup>は Arm Ltd.の登録商標です。
- ・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。
- ・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。

# ALPHAPROJECT

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