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

第 1.2 版　2024 年 09 月 02 日

1.　概要

1.1　概要

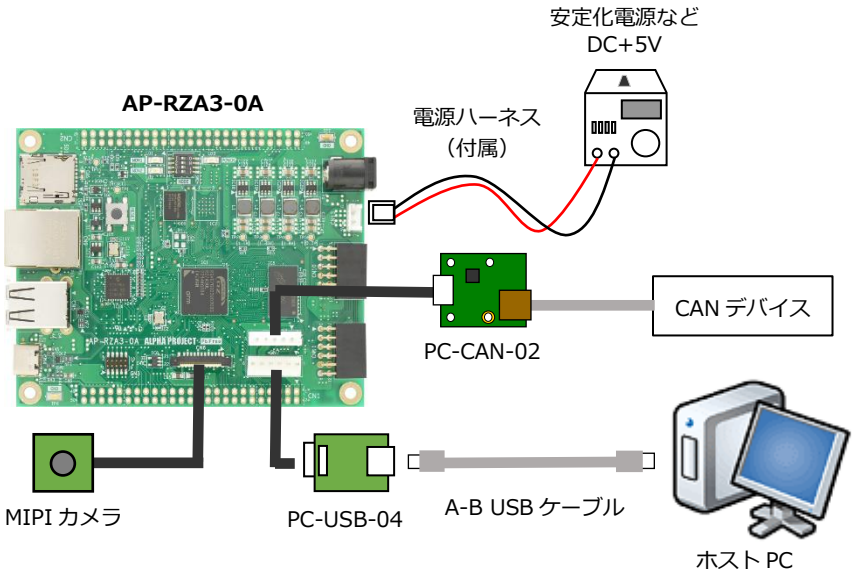
本アプリケーションノートでは、AP-RZA3-0A に付属するサンプルプログラムについて解説します。
解説するサンプルプログラムは下記のものになります。

サンプルプログラム	動作内容
UART・CAN・カメラサンプルプログラム	・UART 動作 エコーバック ・CAN 通信 エコーバック ・MIPI カメラ（Raspberry Pi Camera Module V2 ＋ Raspberry Pi Zero Camera Cable）のデータ受信 ・GTM 動作 LED の点滅
USB HOST サンプルプログラム （ch0、ch1 別個用意）	・USB HOST 動作 ファイル読み書き ・GTM 動作 LED の点滅
USB FUNCTION サンプルプログラム	・USB FUNCTION 動作 仮想 COM 通信 ・GTM 動作 LED の点滅
Ethernet サンプルプログラム	・TCP 通信 エコーバック ・GTM 動作 LED の点滅
SDHI サンプルプログラム	・SD カード ファイル読み書き ・GTM 動作 LED の点滅

1.2　接続概要

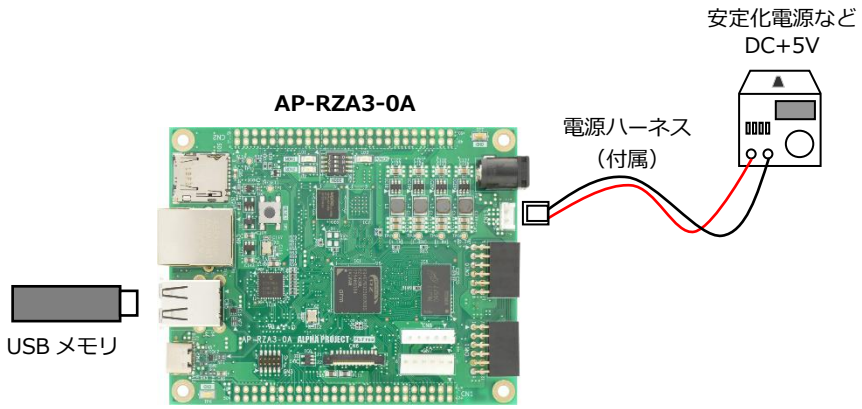
1.2.1　UART・CAN・カメラサンプルプログラムの接続概要

「UART・CAN・カメラ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。
詳細な接続に関しては後述の「３．動作説明」を参照してください。



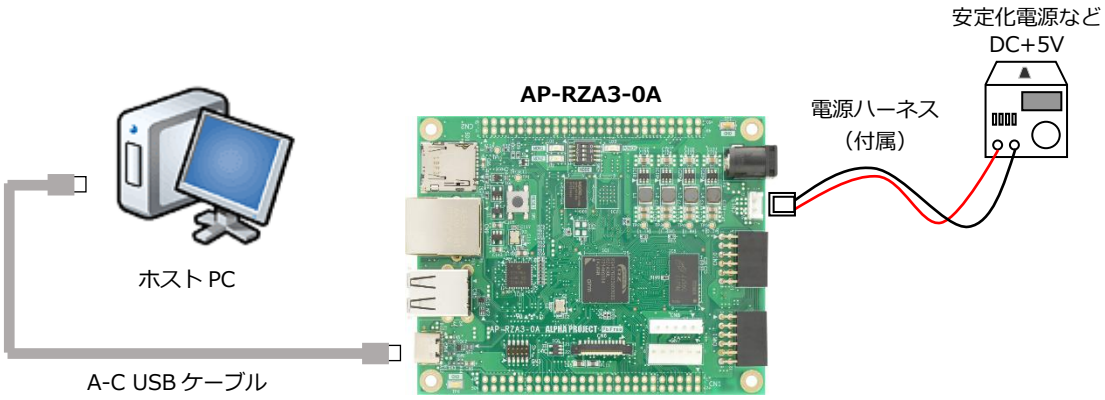
1.2.2　USB HOST サンプルプログラムの接続概要

「USB HOST サンプルプログラム」の動作を確認する上で必要な CPU ボード同士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．動作説明」を参照してください。



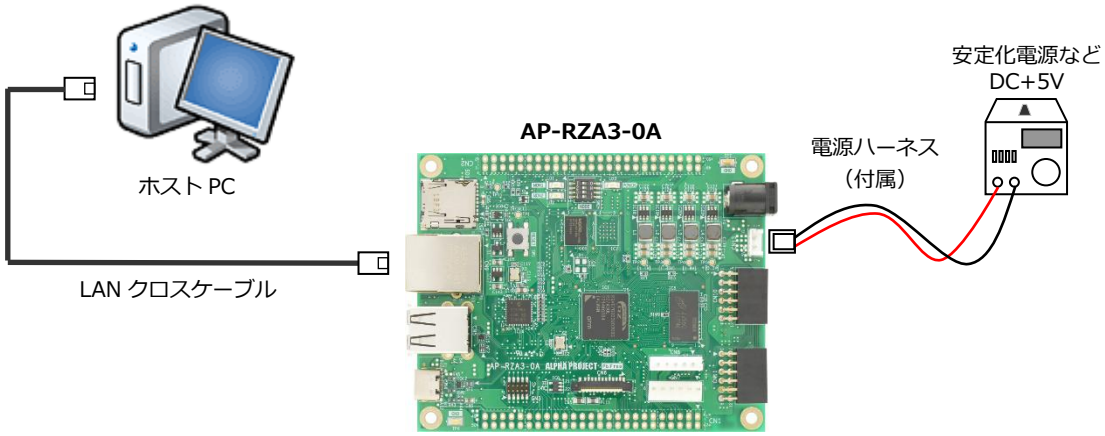
1.2.3　USB FUNCTION サンプルプログラムの接続概要

「USB FUNCTION 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。
詳細な接続に関しては後述の「３．動作説明」を参照してください。



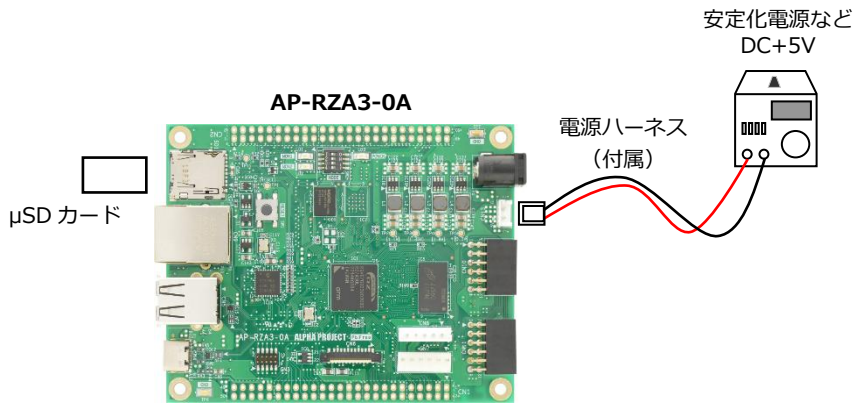
1.2.4　Ethernet サンプルプログラムの接続概要

「Ethernet 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。
詳細な接続に関しては後述の「３．動作説明」を参照してください。



1.2.5　SDHI サンプルプログラムの接続概要

「SDHI 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。
詳細な接続に関しては後述の「３．動作説明」を参照してください。



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

1.4　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
e2studio	2024-07	-
GCC	13.2.1.20231009	A-Profile AArch64 bare-metal
FSP	3.1.0	Flexible Support Package for Renesas RZ/A 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-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。
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
UART・CAN・カメラサンプルプログラム プロジェクトフォルダ	¥sample¥ap_rza3_0a_uartcancam_sample
USB HOST サンプルプログラム プロジェクトフォルダ	¥sample¥ap_rza3_0a_usb_hmsc0_sample ¥sample¥ap_rza3_0a_usb_hmsc1_sample
USB FUNCTION サンプルプログラム プロジェクトフォルダ	¥sample¥ap_rza3_0a_usb_pcdc_sample
Ethernet サンプルプログラム プロジェクトフォルダ	¥sample¥ap_rza3_0a_ethernet_sample
SDHI サンプルプログラム プロジェクトフォルダ	¥sample¥ap_rza3_0a_sdhi_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以下のフォルダで構成されています。

2.1.1　UART・CAN・カメラサンプルプログラムのフォルダ構成

¥ sample	AP-RZA3-0A サンプルプログラムフォルダ
└ ¥ ap_rza3_0a_uartcancam_sample	UART・CAN・カメラサンプルプログラムフォルダ
├ ¥ .settings	設定ファイルフォルダ
├ ¥ ipl	IPL(SPI ブート用プログラム）フォルダ
├ ¥ Debug	デバッグビルド用フォルダ
├ ¥ Release	リリースビルド用フォルダ
├ ¥ script	スクリプト用フォルダ
└ ¥ src	ソースファイル用フォルダ
└camera	カメラデバイス用フォルダ

2.1.2　USB HOST サンプルプログラムのフォルダ構成

・USB HOST ch0

¥ sample	AP-RZA3-0A サンプルプログラムフォルダ
└ ¥ ap_rza3_0a_usb_hmsc0_sample	USB HOST(ch0)サンプルプログラムフォルダ
├ ¥ .settings	設定ファイルフォルダ
├ ¥ ipl	IPL(SPI ブート用プログラム）フォルダ
├ ¥ Debug	デバッグビルド用フォルダ
├ ¥ Release	リリースビルド用フォルダ
├ ¥ script	スクリプト用フォルダ
└ ¥ src	ソースファイル用フォルダ

・USB HOST ch1

¥ sample	AP-RZA3-0A サンプルプログラムフォルダ
└ ¥ ap_rza3_0a_usb_hmsc1_sample	USB HOST(ch1)サンプルプログラムフォルダ
├ ¥ .settings	設定ファイルフォルダ
├ ¥ ipl	IPL(SPI ブート用プログラム）フォルダ
├ ¥ Debug	デバッグビルド用フォルダ
├ ¥ Release	リリースビルド用フォルダ
├ ¥ script	スクリプト用フォルダ
└ ¥ src	ソースファイル用フォルダ

2.1.3　USB FUNCTION サンプルプログラムのフォルダ構成

¥ sample	AP-RZA3-0A サンプルプログラムフォルダ
└ ¥ ap_rza3_0a_usb_pcdc_sample	USB FUNCTION サンプルプログラムフォルダ
├ ¥ .settings	設定ファイルフォルダ
├ ¥ ipl	IPL(SPI ブート用プログラム）フォルダ
├ ¥ Debug	デバッグビルド用フォルダ
├ ¥ Release	リリースビルド用フォルダ
├ ¥ script	スクリプト用フォルダ
└ ¥ src	ソースファイル用フォルダ

2.1.4　Ethernet サンプルプログラムのフォルダ構成

¥ sample	AP-RZA3-0A サンプルプログラムフォルダ
└ ¥ ap_rza3_0a_ethernet_sample	Ethernet サンプルプログラムフォルダ
├ ¥ .settings	設定ファイルフォルダ
├ ¥ ipl	IPL(SPI ブート用プログラム）フォルダ
├ ¥ Debug	デバッグビルド用フォルダ
├ ¥ Release	リリースビルド用フォルダ
├ ¥ script	スクリプト用フォルダ
└ ¥ src	ソースファイル用フォルダ
└ ¥ i2c	I2C 通信用フォルダ

2.1.5　SDHI サンプルプログラムのフォルダ構成

¥ sample	AP-RZA3-0A サンプルプログラムフォルダ
└ ¥ ap_rza3_0a_sdhi_sample	SDHI サンプルプログラムフォルダ
├ ¥ .settings	設定ファイルフォルダ
├ ¥ ipl	IPL(SPI ブート用プログラム）フォルダ
├ ¥ Debug	デバッグビルド用フォルダ
├ ¥ Release	リリースビルド用フォルダ
├ ¥ script	スクリプト用フォルダ
└ ¥ src	ソースファイル用フォルダ

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 フォルダ内>

AlphaProject.ap_rza3_0a.3.1.0.pack	･･･	AP-RZA3-0A パックファイル

2.2.1　UART・CAN・カメラサンプルプログラムのファイル構成

<¥sample¥ap_rza3_0a_uartcancam_sample フォルダ内>

.cproject	･･･	CPROJECT ファイル
.project	･･･	PROJECT ファイル
ap_rza3_0a.pincfg	･･･	AP-RZA3-0A ピンコンフィグファイル
ap_rza3_0a_uartcancam_sample	･･･	AP-RZA3-0A UART・CAN・カメラサンプルプログラム
Debug.jlink		J-Link 設定ファイル（Debug ビルド用）
ap_rza3_0a_uartcancam_sample	･･･	AP-RZA3-0A UART・CAN・カメラサンプルプログラム
Debug.launch		デバッグおよびランタイム設定ファイル（Debug ビルド用）
ap_rza3_0a_uartcancam_sample	･･･	AP-RZA3-0A UART・CAN・カメラサンプルプログラム
Release.jlink		J-Link 設定ファイル（Release ビルド用）
ap_rza3_0a_uartcancam_sample	･･･	AP-RZA3-0A UART・CAN・カメラサンプルプログラム
Release.launch		デバッグおよびランタイム設定ファイル(Release ビルド用)
configuration.xml	･･･	FSP コンフィギュレータファイル

<¥sample¥ap_rza3_0a_uartcancam_sample¥script フォルダ内>

postbuild.sh	･･･	バイナリファイル作成用スクリプトファイル
rza3ul_smarc_qspi.ld	･･･	リンカスクリプトファイル

<¥sample¥ap_rza3_0a_uartcancam_sample¥src フォルダ内>

can_thread_entry.c	･･･	CAN 通信ソースファイル
common_app.h	･･･	サンプルプログラム用定義ヘッダファイル
hal_entry.c	･･･	アプリケーションソースファイル
main_thread_entry.c	･･･	メイン処理ソースファイル
mmu_page_table.c	･･･	MMU ページテーブル定義ソースファイル
sections.c	･･･	セクション定義ソースファイル
syscalls.c	･･･	システムコール関数ソースファイル
uart_thread_entry.c	･･･	シリアル通信ソースファイル

<¥sample¥ap_rza3_0a_uartcancam_sample¥src¥camera フォルダ内>

camera_imx219.c	･･･	カメラ（IMX219）用ソースファイル
camera_imx219.h	･･･	カメラ（IMX219）用ヘッダファイル
camera_imx219_regdata.c	･･･	カメラ（IMX219）用レジスタ設定ソースファイル
camera_imx219_regdata.h	･･･	カメラ（IMX219）用レジスタ設定ヘッダファイル

2.2.2　USB HOST サンプルプログラムのファイル構成

USB HOST はチャ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、チャネル番号を読み替えてご参照ください。

<¥sample¥ap_rza3_0a_usb_hmsc0_sample フォルダ内>

.cproject	･･･	CPROJECT ファイル
.project	･･･	PROJECT ファイル
ap_rza3_0a.pincfg	･･･	AP-RZA3-0A ピンコンフィグファイル
ap_rza3_0a_usb_hmsc0_sample	･･･	AP-RZA3-0A USB HOST(0ch)サンプルプログラム
Debug.jlink		J-Link 設定ファイル（Debug ビルド用）
ap_rza3_0a_usb_hmsc0_sample	･･･	AP-RZA3-0A USB HOST(0ch)サンプルプログラム
Debug.launch		デバッグおよびランタイム設定ファイル（Debug ビルド用）
ap_rza3_0a_usb_hmsc0_sample	･･･	AP-RZA3-0A USB HOST(0ch)サンプルプログラム
Release.jlink		J-Link 設定ファイル（Release ビルド用）
ap_rza3_0a_usb_hmsc0_sample	･･･	AP-RZA3-0A USB HOST(0ch)サンプルプログラム
Release.launch		デバッグおよびランタイム設定ファイル(Release ビルド用)
configuration.xml	･･･	FSP コンフィギュレータファイル

<¥sample¥ap_rza3_0a_usb_hmsc0_sample¥script フォルダ内>

postbuild.sh	･･･	バイナリファイル作成用スクリプトファイル
rza3ul_smarc_qspi.ld	･･･	リンカスクリプトファイル

<¥sample¥ap_rza3_0a_usb_hmsc0_sample¥src フォルダ内>

common_utils.h	･･･	サンプルプログラム用定義ヘッダファイル
hal_entry.c	･･･	アプリケーションソースファイル
mmu_page_table.c	･･･	MMU ページテーブル定義ソースファイル
sections.c	･･･	セクション定義ソースファイル
stdio_uart0.c	･･･	UART 通信用ソースファイル
syscalls.c	･･･	システムコール関数ソースファイル
usb_hmsc_ep.h	･･･	USB HMSC 通信用アプリケーションヘッダファイル
usb_hmsc_thread_entry.c	･･･	USB HMSC 通信用アプリケーションソースファイル

2.2.3　USB FUNCTION サンプルプログラムのファイル構成

<¥sample¥ap_rza3_0a_usb_pcdc_sample フォルダ内>

.cproject	･･･	CPROJECT ファイル
.project	･･･	PROJECT ファイル
ap_rza3_0a.pincfg	･･･	AP-RZA3-0A ピンコンフィグファイル
ap_rza3_0a_usb_pcdc_sample	･･･	AP-RZA3-0A USB FUNCTION サンプルプログラム
Debug.jlink		J-Link 設定ファイル（Debug ビルド用）
ap_rza3_0a_usb_pcdc_sample	･･･	AP-RZA3-0A USB FUNCTION サンプルプログラム
Debug.launch		デバッグおよびランタイム設定ファイル（Debug ビルド用）
ap_rza3_0a_usb_pcdc_sample	･･･	AP-RZA3-0A USB FUNCTION サンプルプログラム
Release.jlink		J-Link 設定ファイル（Release ビルド用）
ap_rza3_0a_usb_pcdc_sample	･･･	AP-RZA3-0A USB FUNCTION サンプルプログラム
Release.launch		デバッグおよびランタイム設定ファイル(Release ビルド用)
configuration.xml	･･･	FSP コンフィギュレータファイル

<¥sample¥ap_rza3_0a_usb_pcdc_sample¥script フォルダ内>

postbuild.sh	･･･	バイナリファイル作成用スクリプトファイル
rza3ul_smarc_qspi.ld	･･･	リンカスクリプトファイル

<¥sample¥ap_rza3_0a_usb_pcdc_sample¥src フォルダ内>

common_app.h	･･･	サンプルプログラム用定義ヘッダファイル
hal_entry.c	･･･	アプリケーションソースファイル
hal_entry.h	･･･	アプリケーションヘッダファイル
mmu_page_table.c	･･･	MMU ページテーブル定義ソースファイル
r_usb_pcdc_apl.h	･･･	USB エコーバックアプリケーションヘッダファイル
r_usb_pcdc_apl_config.h	･･･	USB エコーバックアプリケーション設定ヘッダファイル
r_usb_pcdc_descriptor.c	･･･	USB PCDC ディスクリプタソースファイル
r_usb_pcdc_echo_apl.c	･･･	USB エコーバックアプリケーションソースファイル
sections.c	･･･	セクション定義ソースファイル
syscalls.c	･･･	システムコール関数ソースファイル

2.2.4　Ethernet サンプルプログラムのファイル構成

<¥sample¥ap_rza3_0a_ethernet_sample フォルダ内>

.cproject	･･･	CPROJECT ファイル
.project	･･･	PROJECT ファイル
ap_rza3_0a.pincfg	･･･	AP-RZA3-0A ピンコンフィグファイル
ap_rza3_0a_ethernet_sample	･･･	AP-RZA3-0A Ethernet サンプルプログラム
Debug.jlink		J-Link 設定ファイル（Debug ビルド用）
ap_rza3_0a_ethernet_sample	･･･	AP-RZA3-0A Ethernet サンプルプログラム
Debug.launch		デバッグおよびランタイム設定ファイル（Debug ビルド用）
ap_rza3_0a_ethernet_sample	･･･	AP-RZA3-0A Ethernet サンプルプログラム
Release.jlink		J-Link 設定ファイル（Release ビルド用）
ap_rza3_0a_ethernet_sample	･･･	AP-RZA3-0A Ethernet サンプルプログラム
Release.launch		デバッグおよびランタイム設定ファイル(Release ビルド用)
configuration.xml	･･･	FSP コンフィギュレータファイル

<¥sample¥ap_rza3_0a_ethernet_sample¥script フォルダ内>

postbuild.sh	･･･	バイナリファイル作成用スクリプトファイル
rza3ul_smarc_qspi.ld	･･･	リンカスクリプトファイル

<¥sample¥ap_rza3_0a_ethernet_sample¥src フォルダ内>

common_app.h	･･･	サンプルプログラム用定義ヘッダファイル
hal_entry.c	･･･	アプリケーションソースファイル
mmu_page_table.c	･･･	MMU ページテーブル定義ソースファイル
net_thread_entry.c	･･･	ネットワーク通信ソースファイル
sections.c	･･･	セクション定義ソースファイル
syscalls.c	･･･	システムコール関数ソースファイル
tcp_echo_server_thread_entry.c	･･･	TCP サーバ（エコーバック）通信ソースファイル
udp_echo_server_thread_entry.c	･･･	UDP（エコーバック）通信ソースファイル

<¥sample¥ap_rza3_0a_ethernet_sample¥src¥i2c フォルダ内>

eeprom.c	･･･	EEPROM 通信用ソースファイル
eeprom.h	･･･	EEPROM 通信用ヘッダファイル
i2c.c	･･･	I2C 通信用ソースファイル
i2c.h	･･･	I2C 通信用ヘッダファイル

2.2.5　SDHI サンプルプログラムのファイル構成

<¥sample¥ap_rza3_0a_sdhi_sample フォルダ内>

.cproject	･･･	CPROJECT ファイル
.project	･･･	PROJECT ファイル
ap_rza3_0a.pincfg	･･･	AP-RZA3-0A ピンコンフィグファイル
ap_rza3_0a_sdhi_sample Debug.jlink	･･･	AP-RZA3-0A SDHI サンプルプログラム J-Link 設定ファイル（Debug ビルド用）
ap_rza3_0a_sdhi_sample Debug.launch	･･･	AP-RZA3-0A SDHI 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（Debug ビルド用）
ap_rza3_0a_sdhi_sample Release.jlink	･･･	AP-RZA3-0A SDHI サンプルプログラム J-Link 設定ファイル（Release ビルド用）
ap_rza3_0a_sdhi_sample Release.launch	･･･	AP-RZA3-0A SDHI 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(Release ビルド用)
configuration.xml	･･･	FSP コンフィギュレータファイル

<¥sample¥ap_rza3_0a_sdhi_sample¥script フォルダ内>

postbuild.sh	･･･	バイナリファイル作成用スクリプトファイル
rza3ul_smarc_qspi.ld	･･･	リンカスクリプトファイル

<¥sample¥ap_rza3_0a_sdhi_sample¥src フォルダ内>

common_app.h	･･･	サンプルプログラム用定義ヘッダファイル
hal_entry.c	･･･	アプリケーションソースファイル
mmu_page_table.c	･･･	MMU ページテーブル定義ソースファイル
sdhi_thread_entry.c	･･･	SDHI アプリケーションエントリソースファイル
sections.c	･･･	セクション定義ソースファイル
syscalls.c	･･･	システムコール関数ソースファイル

3.　動作説明

本章ではサンプルプログラムの各動作について説明をしていきます。

3.1　UART・CAN・カメラサンプルプログラムの動作

本サンプルプログラムでは UART、CAN、タイマ(GTM)、カメラの動作を行います。

- UART の動作

  SCI0 を用いてエコーバック通信を行います。
  シリアルの設定は、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等）を使用してください。

- CAN の動作

  CAN0 を用いてエコーバック通信を行います。

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 と同一 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

- タイマ(GTM)の動作

  1msec(1000usec)の割込みを使用し、500msec 周期で LD1、1000msec 周期で LD2 の点滅を行います。

- カメラの動作

  MIPI カメラ（Raspberry Pi Camera Module V2 + Raspberry Pi Zero Camera Cable）と CPU ボードを接続することで、MIPI カメラが撮影した画像データを取得することができます。
  取得する画像データは RAW8 フォーマットで、縦 640：横 480px のデータです。
  LCD に表示するなどして画像データを確認する場合は、ユーザー自身で RGB565 フォーマットや YUV422 フォーマットなどに変換してご確認ください。

  なお、本機能はデフォルトで無効化されています。
  本機能の動作確認を行う場合は、プログラム中の下記を変更してビルドした後にデバッグ動作を行ってください。

  sample¥ap_rza3_0a_uartcancam_sample¥src¥main_thread_entry.c　（8 行目）

```
#define CAMERA_ENABLE (0)      //カメラ機能の動作確認時は "1"に変更してください
```

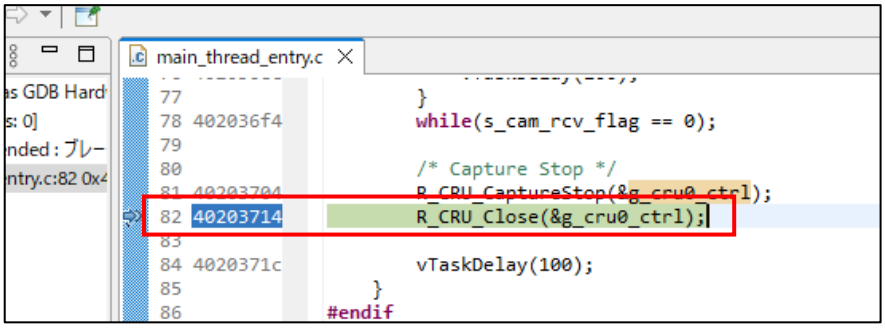
以下に e2studio でデバッグ中に、画像データを PC へ保存する手順を説明いたします。

① 資料「AN1653 RZ/A3UL CPU BOARD 開発チュートリアル」の「3. e2studio を用いた動作方法」を参考に、前述のマクロ「CAMERA_ENABLE」を"1"に設定したうえで、ビルド・デバッグを始めます。

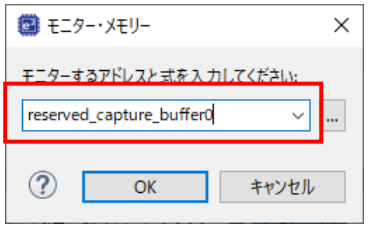
② プログラム中の以下にブレークポイントを配置して、プログラムをブレークします。

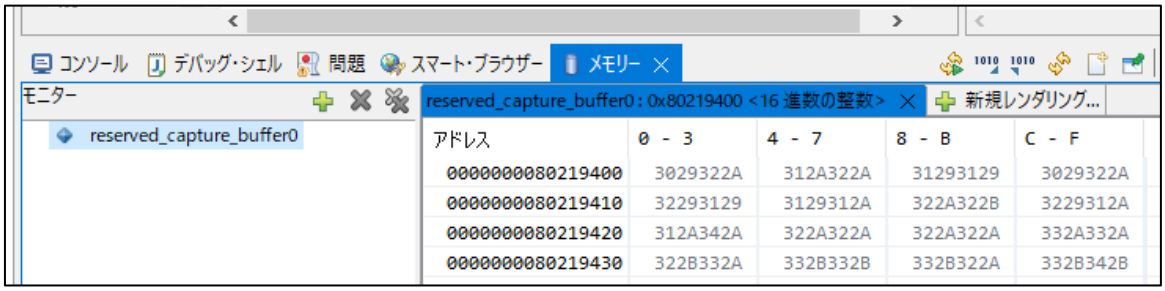
sample¥ap_rza3_0a_uartcancam_sample¥src¥main_thread_entry.c （82 行目）

```
R_CRU_Close(&g_cru0_ctrl);
```

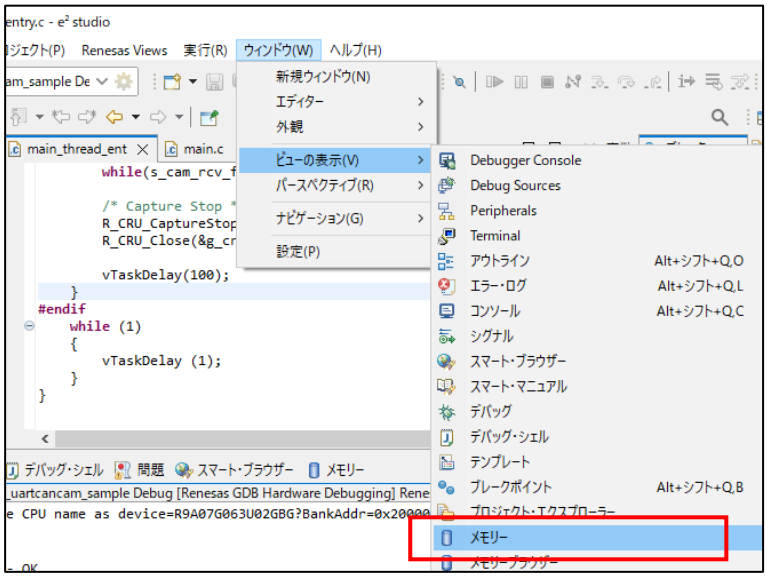


③ e2studio 内のメモリーウィンドウを開き、MIPI カメラからの受信画像データ用バッファ変数「reserved_capture_buffer0」をモニターに追加します。

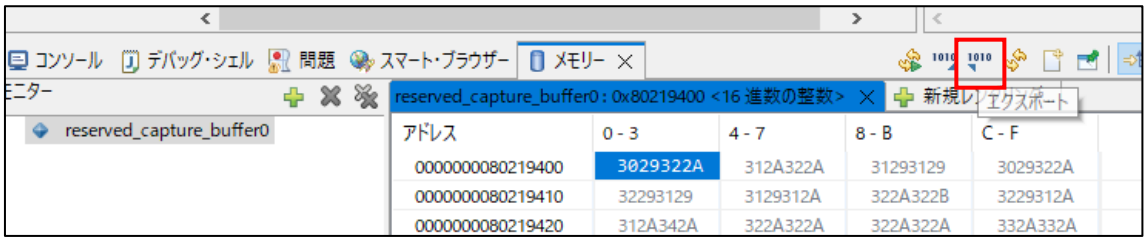


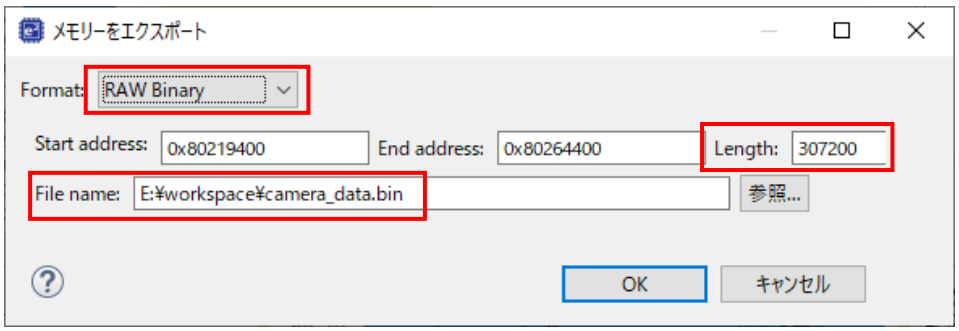


※ メモリーウィンドウが表示されない場合は、e2studio のメニューバー「ウィンドウ」＞「ビューの表示」＞「メモリー」を選択してください。



④　メモリーウィンドウの「エクスポート」ボタンを押し、Format を「RAW Binary」、Length に「307200」(640×480)、File name に任意の画像データ保存先を指定します。





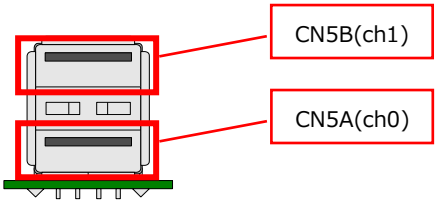
以上で MIPI カメラから受信した RAW8 フォーマットの画像データを PC に保存できます。

3.2　USB HOST サンプルプログラムの動作

USB ホスト（CN5）※に USB メモリを接続すると、USB メモリ上にテキストファイル「rza_usb.txt」を開き、「A」を 10240 バイト分書き込みます。
ファイル書き込み後は、ファイル内容を読み出して比較し、書き込みデータと読み出しデータが一致した場合は LD2 を点灯します。

また、GTM タイマ機能で 1msec(1000usec)の割込みを使用し、500msec 周期で LD1 の点滅を行います。

※ USB HOST 0ch サンプルプログラムはコネクタ下段を、1ch サンプルプログラムはコネクタ上段を使用してください。
また、USB HOST 0ch は USB Function と排他制御となっています。
必ず USB ファンクションコネクタには何も接続しない状態で USB HOST 動作をご確認ください。



3.3　USB FUNCTION サンプルプログラムの動作

USB ファンクション（CN4）に USB ケーブルで PC に接続すると、仮想 COM ポートとしてホスト PC の OS に認識され USB シリアルポートとして動作し、エコーバックを行います。
Win10 よりも前の OS での USB ファンクションの動作確認は、サンプルプログラム内の「COM_Class」フォルダのデータを用いてあらかじめ 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。
動作の確認は、下記の手順で行います。

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(CN7/CN3)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。
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。
④ ターミナルソフトから任意のデータを送信しますと送信した内容がエコーバックで受信されます。

また、GTM タイマ機能で 1msec(1000usec)の割込みを使用し、500msec 周期で LD1 の点滅を行います。

3.4　Ethernet サンプルプログラムの動作

Ethernet 通信によるエコーバックを行います。
CPU ボードは、TCP Server が動作しますので、TCP Client にて、接続を行って下さい。
また、UDP 通信も行い、受信したデータをエコーバックします。

- 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。

設定	プロパティ
IP アドレス	192.168.0.100
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.0.3
ポート番号	10000
MAC アドレス	00-0C-7B-56-XX-XX ※　XX-XX の値は製品ごとに異なります。

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の
¥ap_rza3_0a_ethernet_sample¥src¥net_thread_entry.c ソースファイル内で定義しています。
各設定の定義は以下の通りです。

設定	プロパティ
IP アドレス	static　uint8_t ucIPAddress[4] = {192, 168, 0, 100};
サブネットマスク	static　uint8_t ucNetMask[4] = {255, 255, 255, 0};
ゲートウェイ	static　uint8_t ucGatewayAddress[4] = {192, 168, 0, 3};

また、MAC アドレスはデータ用 EEPROM の先頭 6Byte に格納されています。

アドレス	格納値
先頭アドレス　+　0x00	0x00
+　0x01	0x0C
+　0x02	0x7B
+　0x03	0x56
+　0x04	0xXX
+　0x05	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。

また、GTM タイマ機能で 1msec(1000usec)の割込みを使用し、500msec 周期で LD1 の点滅を行います。

3.5　SDHI サンプルプログラムの動作

microSD カードスロットに microSD カードを挿し込むと、microSD カード上にテキストファイル「test_file.txt」を開き、「0x00, 0x01, 0x02 ... 0xFF」といったバイナリデータを 256 バイト分書き込みます。
ファイル書き込み後は、ファイル内容を読み出して比較し、書き込みデータと読み出しデータが一致した場合は LD2 を点灯します。

また、GTM タイマ機能で 1msec(1000usec)の割込みを使用し、500msec 周期で LD1 の点滅を行います。

3.6　サンプルプログラムのダウンロード

サンプルプログラムを CPU ボード上で実行するためには、ビルドしたサンプルプログラムの実行ファイルを CPU ボードにダウンロードする必要があります。
サンプルプログラムのビルド方法、CPU ボードにサンプルプログラムをダウンロードする方法、ボードのシリアル FlashROM へ書き込んで実行する方法については、以下のアプリケーションノートに詳細な手順が記されています。

・AN1653 RZ/A3UL 開発チュートリアル

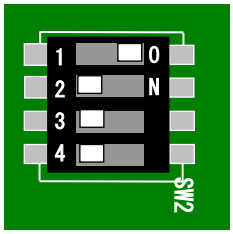
4.　開発環境使用時の各設定値

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。
なお、各ファイル名・フォルダ名につきましては、UART・CAN・カメラサンプルプログラムの内容（ap_rza3_0a_uartcancam_sample）で記載されておりますので、使用するサンプルプログラムに合わせて、赤文字の箇所を読み替えてください。

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_uartcancam_sample
プロジェクト	ap_rza3_0a_uartcancam_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_uartcancam_sample¥Debug
デバッグ用実行ファイル	ap_rza3_0a_uartcancam_sample.elf
Debug hardware	J-Link ARM
Target Device	R9A07G063U02GBG
SerialFlash 書き込み用フォルダ	ap_rza3_0a_uartcancam_sample¥Release
書き込みファイル	ap_rza3_0a_uartcancam_sample.srec

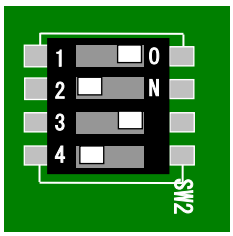
4.1　スイッチ設定

・デバッグをする場合



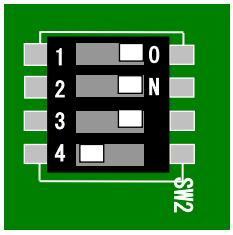
<SW2 設定>
SSCG 設定　　：不問
ブートモード　　：SPI Flash ブートモード(OFF)
デバッグ設定　　：デバッグモード(OFF)

・SPI Flash ブートをする場合



<SW2 設定>
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　　：ノーマルモード(ON)

・SD ブートをする場合



<SW2 設定>
SSCG 設定　　：不問
ブートモード　　：SD ブートモード
デバッグ設定　　：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®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。
・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。

・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 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

・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