# AP-RA6M-0A サンプルプログラム解説

3.1版 2024年02月26日

1.	概要 2
1.	1 概要2
1.	2 接続概要
1.	3 本サンプルプログラムについて
1.	4 開発環境について7
1.	5 ワークスペースについて
2.	サンプルプログラムの構成9
2.	1 フォルダ構成
2.	2 ファイルの構成10
2	$\Delta P_{-}P_{A} \in M_{-} \cap A $ $+ \sum_{i=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{j=1}^{n} \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{j=1}^{n$
5.	
3.	1 RTT Viewer 使用方法14
3.	2 動作説明
	3.2.1 CAN サンプルプログラムの動作説明17
	3.2.2 Ethernet サンプルプログラムの動作説明18
	3.2.3 QSPI サンプルプログラムの動作説明21
	3.2.4 UART サンプルプログラムの動作説明21
	3.2.5 SDHI サンプルプログラムの動作説明22
	3.2.6 USB ホストサンプルプログラムの動作説明23
	3.2.7 USB ファンクションサンプルプログラムの動作説明24
3.	3 メモリマップ
3.	4 e2 studio を用いたプロジェクトのビルド・デバッグ27
	3.4.1 インポート方法
	3.4.2 ビルド方法
	3.4.3 デバッグ、ダウンロード方法

# 1. 概要

# 1.1 概要

本アプリケーションノートでは、AP-RA6M-0A(RA6M3 CPU)を用いて、Flexible Software Package を使用したサンプルプロ グラムについて解説します。

AP-RA6M-0A には、下記のサンプルプログラムが付属しています。 本サンプルプログラムで使用する主な機能を以下に記します。

デバイス	機能
AP-RA6M-0A	・CAN 通信
	・ネットワーク通信
	・QSPI FlashROM 読み書き
	・UART 通信
	・SD カード読み書き
	・USB ホスト メモリ読み書き
	・USB ファンクション 仮想 COM 通信

#### 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。

※AP-RA6M-0A と J-Link を直接接続することはできません。

AP-RA6M-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。



・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



#### ・QSPI 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-RA6M-0A 製品ページ https://www.apnet.co.jp/product/ra/ap-ra6m-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	V2023-04	_
FSP	V4.5.0	_
GCC ARM Embedded	V10.3.1.20210824	_
RTTViewer	V7.92b	Segger Microcontroller Systems 社

デバッガ	ハードウェアバージョン	備考
J-Link	V10	Segger Microcontroller Systems 社

※AP-RA6M-0A と J-Link を直接接続することはできません。

AP-RA6M-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_ra6m_0a_can_sample
Ethernet サンプルプログ	¥sample¥ap_ra6m_0a_ether_sample
ラム	
QSPI サンプルプログラム	¥sample¥ap_ra6m_0a_qspi_sample
UART サンプルプログラム	¥sample¥ap_ra6m_0a_sci_uart_sample
SDHI サンプルプログラム	¥sample¥ap_ra6m_0a_sdhi_sample
USB ホストサンプルプロ	¥sample¥ap_ra6m_0a_usb_hmsc_sample
グラム	
USB ファンクションサン	¥sample¥ap_ra6m_0a_usb_pcdc_sample
プルプログラム	



# 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。

sample

ч	linpic	
	— ap_ra6m_0a_can_sample	CAN サンプルプログラムフォルダ
	— .settings	設定ファイルフォルダ
	script	リンカ設定ファイルフォルダ
	src	ソースフォルダ
	L Debug	ワークフォルダ
	— ap_ra6m_0a_ether_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
	— script	リンカ設定ファイルフォルダ
	src	ソースフォルダ
	L Debug	ワークフォルダ
	ap_ra6m_0a_qspi_sample	QSPI サンプルプログラムフォルダ
	— .settings	設定ファイルフォルダ
	script	リンカ設定ファイルフォルダ
	src	ソースフォルダ
	L Debug	ワークフォルダ
	— ap_ra6m_0a_sci_uart_sample	UART サンプルプログラムフォルダ
	— .settings	設定ファイルフォルダ
	— script	リンカ設定ファイルフォルダ
	src	ソースフォルダ
	L Debug	ワークフォルダ
$\left  \right $	— ap_ra6m_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
	— script	リンカ設定ファイルフォルダ
	src	ソースフォルダ
	L Debug	ワークフォルダ
$\left  \right $	ap_ra6m_0a_usb_hmsc_sample	USB ホストサンプルプログラムフォルダ
	— .settings	設定ファイルフォルダ
	script	リンカ設定ファイルフォルダ
	src	ソースフォルダ
	L Debug	ワークフォルダ
$\left  \right $	— ap_ra6m_0a_usb_pcdc_sample	USB ファンクションサンプルプログラムフォルダ
	— .settings	設定ファイルフォルダ
	script	リンカ設定ファイルフォルダ
	src	ソースフォルダ
	L Debug	ワークフォルダ

#### 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\_ra6m\_0a.4.5.0 ··· AP-RA6M-0A パックファイル .pack

#### ・CAN サンプルプログラム

<¥sample¥ap\_ra6m\_0a\_can\_sample フォルダ内> .cproject • • • CPROJECT ファイル .project • • • PROJECT ファイル configuration.xml . . . FSP コンフィギュレータファイル ap\_ra6m\_0a.pincfg • • • AP-RA6M-0A ピンコンフィグファイル ap\_ra6m\_0a\_can\_sample • • • AP-RA6M-0A CAN 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 <¥sample¥ap\_ra6m\_0a\_can\_sample¥script フォルダ内> fsp.ld • • • e2 studio 用 リンカスクリプトファイル

#### <¥sample¥ap\_ra6m\_0a\_can\_sample¥src フォルダ内>

SEGGER_RTT	•••	RTTViewer ソース格納フォルダ
timer	•••	タイマ処理ソース格納フォルダ
common_utils.h	•••	共通ヘッダファイル
hal_entry.c	•••	アプリケーションソースファイル

#### ・Ethernet サンプルプログラム

<¥sa	mple¥ap_ra6m_0a_ether_sample 🗸	オルダ内	
	.cproject	•••	CPROJECT ファイル
	.project		PROJECT ファイル
	configuration.xml	•••	FSP コンフィギュレータファイル
	ap_ra6m_0a.pincfg		AP-RA6M-0A ピンコンフィグファイル
	ap_ra6m_0a_ether_sample		AP-RA6M-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
<¥sa	mple¥ap_ra6m_0a_ether_sample¥s	script ファ	tルダ内>
	fsp.ld	•••	e2 studio 用 リンカスクリプトファイル
<¥sa	mple¥ap_ra6m_0a_ether_sample¥s	src フォル	夕内>
	i2c	• • •	I2C 通信ソース格納フォルダ
	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
	net_thread_entry.c	•••	ネットワークアプリケーションソースファイル
	usr_app.h		ユーザーアプリケーションヘッダファイル
· QSI	PI サンプルプログラム		
<¥sa	mple¥ap_ra6m_0a_qspi_sample フ	ォルダ内ン	>
	.cproject	•••	CPROJECT ファイル
	.project	•••	PROJECT ファイル
	configuration.xml	•••	FSP コンフィギュレータファイル
	APRA6M0A_QSPI.pincfg		AP-RA6M-0A ピンコンフィグファイル (QSPI サンプルプロ グラム専用)
	ap_ra6m_0a_qspi_sample		AP-RA6M-0A QSPI 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
<¥sa	mple¥ap_ra6m_0a_qspi_sample¥sc	ript フォ.	ルダ内>
	fsp.ld		e2 studio 用 リンカスクリプトファイル
<¥sa	mple¥ap_ra6m_0a_qspi_sample¥sr	c フォルタ	ダ内>
	sdram	•••	SDRAM 処理ソース格納フォルダ
	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
	qspi_ep.h		SPIFlashROM 情報ヘッダファイル

#### ・UART サンプルプログラム

	le¥ap_ra6m_0a_sci_uart_sample	フォルタ	「内>
	.cproject	•••	CPROJECT ファイル
	.project	•••	PROJECT ファイル
	configuration.xml	•••	FSP コンフィギュレータファイル
	ap_ra6m_0a.pincfg	•••	AP-RA6M-0A ピンコンフィグファイル
	ap_ra6m_0a_sci_uart_sample		AP-RA6M-0A UART 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
<¥samp	le¥ap_ra6m_0a_sci_uart_sample¥	script 2	フォルダ内>
	fsp.ld	•••	e2 studio 用 リンカスクリプトファイル
<¥samp	le¥ap_ra6m_0a_sci_uart_sample¥	ésrc 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
	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
· SDHI	サンプルプログラム		
<¥samp	le¥ap_ra6m_0a_sdhi_sample フォ	ルダ内>	
<¥samp	le¥ap_ra6m_0a_sdhi_sample フォ .cproject	ルダ内> 	CPROJECT ファイル
<¥samp	le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project	ルダ内> 	CPROJECT ファイル PROJECT ファイル
<¥samp	le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project configuration.xml	ルダ内>  	CPROJECT ファイル PROJECT ファイル FSP コンフィギュレータファイル
<¥samp	le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project configuration.xml ap_ra6m_0a.pincfg	ルダ内>   	・ CPROJECT ファイル PROJECT ファイル FSP コンフィギュレータファイル AP-RA6M-0A ピンコンフィグファイル
<¥samp	le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project configuration.xml ap_ra6m_0a.pincfg ap_ra6m_0a_sdhi_sample	ルダ内>   	CPROJECT ファイル PROJECT ファイル FSP コンフィギュレータファイル AP-RA6M-0A ピンコンフィグファイル AP-RA6M-0A SDHI サンプルプログラム
<¥samp	le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project configuration.xml ap_ra6m_0a.pincfg ap_ra6m_0a_sdhi_sample Debug.launch	ルダ内>   	CPROJECT ファイル PROJECT ファイル FSP コンフィギュレータファイル AP-RA6M-0A ピンコンフィグファイル AP-RA6M-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
<¥samp <¥samp	<pre>le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project configuration.xml ap_ra6m_0a.pincfg ap_ra6m_0a_sdhi_sample Debug.launch le¥ap_ra6m_0a_sdhi_sample¥scri</pre>	ルダ内>    pt フォノ	CPROJECT ファイル PROJECT ファイル FSP コンフィギュレータファイル AP-RA6M-0A ピンコンフィグファイル AP-RA6M-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
<¥samp <¥samp	<pre>le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project configuration.xml ap_ra6m_0a_sdhi_sample Debug.launch le¥ap_ra6m_0a_sdhi_sample¥scri fsp.ld</pre>	ルダ内>    pt フォル 	CPROJECT ファイル PROJECT ファイル FSP コンフィギュレータファイル AP-RA6M-0A ピンコンフィグファイル AP-RA6M-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 レダ内> e2 studio 用 リンカスクリプトファイル
<¥samp <¥samp <¥samp	<pre>le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project configuration.xml ap_ra6m_0a_sdhi_sample Debug.launch le¥ap_ra6m_0a_sdhi_sample¥scri fsp.ld le¥ap_ra6m_0a_sdhi_sample¥src</pre>	ルダ内>    pt フォル フォルタ	CPROJECT ファイル PROJECT ファイル FSP コンフィギュレータファイル AP-RA6M-0A ピンコンフィグファイル AP-RA6M-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 レダ内> e2 studio 用 リンカスクリプトファイル
<¥samp <¥samp <¥samp	<pre>le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project configuration.xml ap_ra6m_0a_sdhi_sample Debug.launch le¥ap_ra6m_0a_sdhi_sample¥scri fsp.ld le¥ap_ra6m_0a_sdhi_sample¥src SEGGER_RTT</pre>	ルダ内>    pt フォル  フォルタ 	CPROJECT ファイル PROJECT ファイル FSP コンフィギュレータファイル AP-RA6M-0A ピンコンフィグファイル AP-RA6M-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 レダ内> e2 studio 用 リンカスクリプトファイル
<¥samp <¥samp <¥samp	<pre>le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project configuration.xml ap_ra6m_0a_sdhi_sample Debug.launch le¥ap_ra6m_0a_sdhi_sample¥scri fsp.ld le¥ap_ra6m_0a_sdhi_sample¥src SEGGER_RTT common_utils.h</pre>	ルダ内>    pt フォル  フォル络 	CPROJECT ファイル PROJECT ファイル FSP コンフィギュレータファイル AP-RA6M-0A ピンコンフィグファイル AP-RA6M-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 レダ内> e2 studio 用 リンカスクリプトファイル
<¥samp <¥samp <¥samp	<pre>le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project configuration.xml ap_ra6m_0a_sdhi_sample Debug.launch le¥ap_ra6m_0a_sdhi_sample¥scri fsp.ld le¥ap_ra6m_0a_sdhi_sample¥src SEGGER_RTT common_utils.h hal_entry.c</pre>	ルダ内>    pt フォル  フォルタ 	CPROJECT ファイル PROJECT ファイル FSP コンフィギュレータファイル AP-RA6M-0A ピンコンフィグファイル AP-RA6M-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 Vダ内> e2 studio 用 リンカスクリプトファイル が入> RTTViewer ソース格納フォルダ 共通ヘッダファイル hal_entry 関数ソースファイル
<¥samp <¥samp <¥samp	<pre>le¥ap_ra6m_0a_sdhi_sample フォ .cproject .project configuration.xml ap_ra6m_0a_sdhi_sample Debug.launch le¥ap_ra6m_0a_sdhi_sample¥scri fsp.ld le¥ap_ra6m_0a_sdhi_sample¥src SEGGER_RTT common_utils.h hal_entry.c sdhi_ep.h</pre>	ルダ内>    pt フォル  フォルタ  	CPROJECT ファイル PROJECT ファイル FSP コンフィギュレータファイル AP-RA6M-0A ピンコンフィグファイル AP-RA6M-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 レダ内> e2 studio 用 リンカスクリプトファイル が入> RTTViewer ソース格納フォルダ 共通ヘッダファイル hal_entry 関数ソースファイル SDHI 情報ヘッダファイル

#### ・USB ホストサンプルプログラム

<¥sample¥ap\_ra6m\_0a\_usb\_hmsc\_sample フォルダ内>

		V3197
.cproject	•••	CPROJECT ファイル
.project	•••	PROJECT ファイル
configuration.xml	•••	FSP コンフィギュレータファイル
ap_ra6m_0a.pincfg		AP-RA6M-0A ピンコンフィグファイル
ap_ra6m_0a_usb_hmsc		AP-RA6M-0A USB ホストサンプルプログラム
_sample Debug.launch		デバッグおよびランタイム設定ファイル
<¥sample¥ap_ra6m_0a_usb_hmsc_sam	ıple¥scrip	it フォルダ内>
fsp.ld		e2 studio 用 リンカスクリプトファイル
<¥sample¥ap_ra6m_0a_usb_hmsc_sam	ıple¥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
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
・ <b>USB ファンクションサンフルフロクラム</b> <¥sample¥ap_ra6m_0a_usb_pcdc_sam	ple フォル	ダ内>
.cproject	•••	CPROJECT ファイル
.project	•••	PROJECT ファイル
configuration.xml	•••	FSP コンフィギュレータファイル
ap_ra6m_0a.pincfg		AP-RA6M-0A ピンコンフィグファイル
ap_ra6m_0a_usb_pcdc	•••	AP-RA6M-0A USB ファンクションサンプルプログラム
_sample Debug.launch		デバッグおよびランタイム設定ファイル
<¥sample¥ap_ra6m_0a_usb_pcdc_sam	ple¥script	フォルダ内>
fsp.ld	•••	e2 studio 用 リンカスクリプトファイル
<¥sample¥ap_ra6m_0a_usb_pcdc_sam	ple¥src フ	オルダ内>
board_cfg.h	•••	ボード情報ヘッダファイル
common_init.c	•••	共通初期化ソースファイル
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
r_usb_pcdc_descriptor.c	•••	USB ファンクション情報ヘッダファイル

# アプリケーションノート AN2003

# 3. AP-RA6M-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、以下の手順に従い行ってください。

- 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、Specify Target Device を「R7FA6M3AH」に設定します。
   ※.画面表示は、RTTViewer 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、「V7.92b」を使用した場合の手順を記載します。

L ink RTT Viewer V7 86b   Configuration
Onnection to J-Link
U Existing Session
Specify Target Device
R7FA6M3AH ~
Force go on connect
Script file (optional)
Target Interface & Speed
JTAG 👻 4000 kHz 👻
JTAG scan chain information
<u>A</u> uto detection
O Simple configuration
RTT Control Block
Auto Detection O Address O Search Range
J-Link automatically detects the RTT control block.
OK Cancel

- 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。

🔜 J-Link RTT Viewer V7.86b   Configuration		
Connection to J-Link		
● <u>U</u> SB	<u>S</u> erial No	
O <u>T</u> CP∕IP		
○ <u>E</u> xisting Session		
Specify Target Device		
R7FA6M3AH	×	
Force go on connect		
Script file (optional)		
Target Interface & Speed		
JTAG	▼ 4000 kHz	•
JTAG scan chain information		
Auto detection		
○ Simple configuration		
RTT Control Block		
Auto Detection O Address	○ Search <u>R</u> ange	
J-Link automatically detects the RTT	control block.	
	OK Cancel	

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。

J-Link RTT Viewer V7.86b	-		×
<u>File Terminals Input Logging H</u> elp			_
All Terminals Terminal 0 Terminal 1 Terminal 2			
<pre>00&gt; 00&gt; 00&gt; 00&gt; 00&gt; 00&gt; 00&gt; 00&gt; 00&gt; 00</pre>	d n Renesas F zes SDHI dr Remove & n is execut	= =SP. =iver t	<b>*</b>
<			>
	Enter	0	ar
	Enter	Cie	
TOO: [a][a]: E000E000 CIN DI02E000 KIN 000DE00C 2C2-MV			<b>^</b>
LOG: [0][1]: 50001000 CID B105F000 PID 0038B002 DWT LOG: [0][2]: E0002000 CID B105F000 PID 0028B003 FPB LOG: [0][3]: E0000000 CID B105F000 PID 0038B001 ITM LOG: [0][3]: E0040000 CID B105F000 PID 0008B0A1 TFIU LOG: [0][5]: E0041000 CID B1059000 PID 0008B04 ESTF LOG: [0][6]: E0044000 CID B1059000 PID 0018B008 CSTF LOG: [0][6]: E0044000 CID B1059000 PID 0018B001 TWC LOG: [0][2]: E0044000 CID B105F000 PID 0018B101 TSG LOG: [0][2]: E0044000 CID B105F000 PID 0018B101 TSG			~
RTT Viewer connected. 16.	571 KB		

⑦ CPU ボードに入力を行う場合は、[Terminal]の入力部にデータを入力します。

🔜 J-Link RTT V	fiewer V7.86t	ogging He	In				-		×
All Terminole	Tuminal O	Truminal 1	P Tomical	0					
Airtenninais	Terminal O	Terminal I	Terminal	2					
00>									í
00> ******		Evample Dec	doct for		dula	********	**********	******	
00> * Exa	mole Proi	ect Version	1.0	- SUNT PIC	Juure				*
00> * Fle	x Softwar	e Pack Vers	ion 4.4	4.0					*
00> ******	*******	*********	******	*******	*******	********	*******	*****	*
00> Refer t	o readme.	txt file fo	r more o	details o	on Exampl	e Project	and		
00> FSP Use	r's Manua.	l for more	informa	tion abou	it SDHI d	river			
00>	1						L 2		
00> driver	with Free	RTOS+FAT fi	le svst	em on Rer	nesas RA	MCUs based	on Renesas	FSP	
00> FreeRTO	S+FAT use	s the under	lving B	lock medi	a driver	which uti	lizes SDHI d	river	to
00> file op	erations (	on the SD C	and dev	ices.					
00> Error a	nd info m	essages wil	1 be pr	inted on	JlinkRTT	Viewer.			
00>							- •		
00> Please	remove and	d re-insert	the SD	Card aft	er execu	ting Safel	y Remove &		
00> Initial	12e Freek	TUS+FAT CON	manu						
00> FreeRTO	S+FAT Oper	n successfu	1						
00> Connec	t SD Card								
00>									
00> SD Card	Menu opt:	ions							
00> 1. Writ	e 10k data	a trom app_	butter 1	to ra_sdr	11.TXT T1	Te			
00> 2. Form 00> 3. Safe	lv Remove	the SD Car	d						
00> 4. Disp	lay conte	nt of Curre	nt Work	ing Direc	tory				
00> 5. Init	ialize Fro	eeRTOS+FAT	(Valid	only afte	r Safely	_Eject opt	ion is execu	ited)	
< .									>
							Enter	Cl	ear
					31.3 - 117				
LOG: [0][1]	E0001000	OCID B105E	00D PID	00388002	DWT				
LOG: [0][2]:	E0002000	O CID B105E	00D PID	00288003	FPB				
LOG: [0][3]	E000000	O CID B105E	00D PID	00388001	ITM				
LOG: [0][4]	E0040000	CID 81059	00D PID	000BB9A1	TPIU				
015	E0041000	CID 81059	000 PID	00068925	CSTE				
	20042000	CID 81059	00D PID	00188961	TMC				
LOG: [0][6] LOG: [0][7]	E0043000								
LOG: [0][5] LOG: [0][6] LOG: [0][7] LOG: [0][8]	E0043000	O CID B105F	00D PID	001BB101	TSG				
LOG: [0][5] LOG: [0][6] LOG: [0][7] LOG: [0][8]	: E0043000 : E0044000	OCID B105F	00D PID	001BB101	TSG				
LOG: [0][5] LOG: [0][6] LOG: [0][7] LOG: [0][8] LOG: RTT Vi	E0043000 E0044000	CID B105F	00D PID	00188101	TSG				>

⑧ 動作確認が完了したら、RTT Viewer を終了した後、CPU ボードの電源を落とします。

RTT Viewerの詳細解説は、「J-Link / J-Trace User Guide」を参照してください。

サンプルプログラムにて RTT Viewer 上に表示される情報は、Renesas サンプルプログラムがベースになっています。 一部表示に関しては、弊社サンプルプログラムの動作と異なる点もございますので、ご注意ください。

サンプルプログラムを実行しても RTT Viewer 上に情報が表示されない場合は、CPU ボードのプログラムをリセットした後、 RTT Viewer のメニューバー [File] - [Disconnect] を選択して通信を切断してから改めて再接続を行い、プログラムを再実行 してください。

#### 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、プログラム開始後、5sec 間隔で固定データ(TX\_MSG)の送信を行いながら、データの受信待ち を行います。

CAN 送信、受信動作の結果は、RTT Viewer に出力します。

※受信データは文字列として RTT Viewer に出力します。受信データに NULL 文字(0x00)が含まれていると、NULL 文字を 文字列の終端とみなし、NULL 文字以前の文字までを RTT Viewer に出力します。

CAN の通信設定は、以下のように設定されています。

	CPU ボードの設定				
ID	受信 Mailbox ID	B'0000000010 (0x002)			
	送信 Mailbox ID	B'0000000011 (0x003)			
	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。 ネットワーク動作の確認は、以下の手順に従い行ってください。

- ① LAN クロスケーブルを用い、CPU ボードの LAN コネクタ(CN5)とホスト PC を接続します。
- 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	192.168.1.201
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。
- 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「192.168.1.201」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。

🔜 J-Link RTT Viewer V7.86b	_		$\times$
<u>File Terminals Input Logging H</u> elp			
All Terminals Terminal O			
007			~
00> Network is Up			
00> Ethernet adapter for Renesas AP-RA6M-0A:			
00> Description Renesas AP-RA6M-0A Ethernet			
00> Physical Address			
00> IPv4 Address			
00> Subnet Mask			
00> Default Gateway : 192.168.1.254			
00> DNS Servers			
00> Pinging 192.168.1.201:			
00>			
00>			
00> 00> Ping Statistics for 192.168.1.201 :			
00>			
00> Packets: Sent = 100, Received = 99, Lost = 00			$\sim$
<			>
	Enter	Cler	ar
	2000	0.0	
LOG. KONTULIEJIZJ. LOGOZOGO, CLD. DIGOLOGO, FID. GOZODOGO FRU			~
LOG: ROMIDI[0][3]: E0000000, CID: BI05E00D, PID: 003BB001 ITM			
LOG: ROMTb1[0][5]: E0041000, CID: B105900D, PID: 000BB925 ETM			
LOG: ROMTb1[0][6]: E0042000, CID: B105900D, PID: 002BB908 CSTF			
LOG: ROMTb1[0][7]: E0043000, CID: B105900D, PID: 001BB961 TMC			
LOG: ROMIDI[0][8]: E0044000, CID: B105F00D, PID: 001BB101 ISG			
Loo. Kit viewei connected.			
	(D		••••
KTT viewer connected. UUU2	лы		

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。

IPアドレス	192.168.1.200
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.1.254
MAC アドレス	00-0C-7B-4D-XX-XX
	※ XX-XX の値は製品ごとに異なります。
Ping 通信先 IP アドレス	192.168.1.201

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_	ra6m_0a	_ether_	_sample¥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\_ra6m\_0a\_ether\_sample¥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 6Byte に格納されています。

アドレス(CH0)		格納値	
先頭アドレス	+	0x00	0×00
	+	0x01	0x0C
	+	0x02	0x7B
	+	0x03	0x4D
	+	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。

#### 3.2.3 QSPI 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、QSPI FlashROM ヘデータ書き込み、読出し、Verify チェックを行います。 Verify チェックの結果を、RTT Viewer に出力します。

また、「sample¥ap\_ra6m\_0a\_qspi\_sample¥src¥hal\_entry.c」内の API「hal\_entry(void)」 61 行目にある #if を有効にす ることで、SDRAM ヘデータ書き込み、読出し、Verify チェックも行えます。 SDRAM 内の Verify チェックで異常を検知すると、RTT Viewer に「SDRAM Check Error」と出力します。

#### 3.2.4 UART 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(SCI4)を用いてコマンドを送信することで、LED の点灯/消灯を変更することができます。

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。

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 (バイナリ値)	説明
1 (0x31, 0x0d)	モニタ LED1 を点灯します。
	モニタ LED2 を消灯します。
2 (0x32, 0x0d)	モニタ LED1 を消灯します。
	モニタ LED2 を点灯します。
3 (0x33, 0x0d)	モニタ LED1 を点灯します。
	モニタ LED2 を点灯します。
4 (0x34, 0x0d)	モニタ LED1 を消灯します。
	モニタ LED2 を消灯します。

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 LED が点灯/消灯します。

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 1-4」と表示されます。

3.2.5 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、SD 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、以下の手順に従い行ってください。

- 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 SD 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。
- 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	SD 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	SD カードのフォーマットを行います。
3	SD カードの安全な抜出準備を行います。
4	SD 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。

3.2.6 USB ホスト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、RTTViewer を用いてコマンドを送信することで、USB メモリへのデータ読み書きを行うことができます。

ネットワーク動作の確認は、以下の手順に従い行ってください。

- 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 USB ホストポート(CN4)に USB メモリを挿入して電源を投入し、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確認します。
- 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 USB 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。

le <u>T</u> erminal	s <u>I</u> nput	Lossins He	2lp								
All Terminals	Termine	al O									
20x file o	peratio	ns on the	HIGE LY	orage devi	incurru	GILVEI N	mach aca		/ 1015		
00> Fric c	and inf	o messages	will	be printed	d on J	linkRTTVi	ewer.				
30>		0									
30>											
00> FreeRT	TOS+FAT	Open succe	ssful								
	ect USB	Device									
00> USB D	Device i	s connecte	d								
30>											
00≻ USB HN	ISC Menu	options									
30>	10	d-4- 6				A.A. 611-					
00> 1. Wr	rite IØK	data from	app b	ourrer to r	a usb	.txt Tile					
20											
00> 00> 2.Fc	ormat US	B Drive									
00> 00> 2. Fc 00>	ormat US	B Drive									
20> 20> 2. Fo 20> 20> 3. Sa	ormat US afely Ej	B Drive ect the US	B Driv	/e							
90> 90> 2. Fo 90> 90> 3. Sa 90>	ormat US afely Ej	B Drive ect the US	B Driv	ve		n Safalu	Eiect or	tion is a		ted)	
90> 90> 2. Fo 90> 90> 3. Sa 90> 90> 4. Ir	ormat US afely Ej nitializ	B Drive ect the US e FreeRTOS	B Driv +FAT (	/e [Valid only	/ afte	r Safely_	_Eject op	rtion is e	execu	ited)	Ì
90> 90> 2. Fo 90> 90> 3. Sa 90> 90> 4. In	ormat US afely Ej nitializ	B Drive ect the US e FreeRTOS	B Driv +FAT (	ve Valid only	/ afte	r Safely_	_Eject op	otion is e	execu	ited)	>
90> 90> 2. Fo 90> 90> 3. Sa 90> 90> 4. In	ormat US afely Ej nitializ	B Drive ect the US e FreeRTOS	B Driv +FAT (	/e [Valid only	/ afte	r Safely_	Eject op	tion is e	er	ited) Cle	> ear
00> 2. Fc 00> 30> 3. Sa 00> 30> 4. Ir	ormat US afely Ej nitializ	B Drive ect the US e FreeRTOS	B Driv	/e (Valid only :	– / afte	r Safely_	_Eject op	tion is e	e <b>xecu</b>	rted) Cia	> ear
00> 2. Fc 00> 2. Fc 00> 3. Sa 00> 3. Sa 00> 4. In c	ormat US afely Ej nitializ	B Drive ect the US e FreeRTOS	B Driv +FAT (	Valid only	/ afte	r Safely_	Eject op	otion is e Ent	er	ited) Cle	> ear
00> 00> 00> 00> 00> 00> 00> 00>	ormat US afely Ej nitializ	B Drive ect the US e FreeRTOS : E0000000 : E0000000 : E0040000	B Driv +FAT ( , CID. , CID: , CID:	/e (Valid only 51051000, 81055000, 81055000,	- / afte PID: PID: PID:	r Safely_ 002050003 00385001 000889A1	Eject op ITM TPIU	etion is e	er	ited)	> ear
00> 2. Fc 00> 2. Fc 00> 3. Sa 00> 4. Ir c 00: ROMTb 0.0G: ROMTb 0.0G: ROMTb 0.0G: ROMTb	ormat US afely Ej nitializ 01[0][3]: 01[0][4] 01[0][5]:	B Drive ect the US e FreeRTOS : E0002000 : E0000000 : E0040000 : E0041000	B Driv +FAT ( , CID. , CID: , CID: , CID:	(Valid only Using the second Biostope, Biostope, Biossoop, Biossoop, Biossoop, Biossoop,	- y afte PID: PID: PID: PID:	r Safely_ 002050005 00368001 00088901 00088925	Eject op ITM ITM TPIU ETM	etion is e	er	ited) Cie	> ear
202> 202> 2. Fc 202> 2. Fc 202> 3. Sa 202> 4. Ir c 200- ROMTb 203- ROMTb 203- ROMTb 203- ROMTb 203- ROMTb 203- ROMTb 203- ROMTb	Drmat US afely Ej nitializ 0[0][3]: 1[0][4] 0[0][5]: 0[0][6]:	B Drive ect the US e FreeRTOS : E0002000 : E0000000 : E0041000 : E0042000	B Driv +FAT ( , CID: , CID: , CID: , CID: , CID:	/e (Valid only 01051000; 01055000, 01055000, 01059000, 01059000,	- / afte PID: PID: PID: PID: PID:	r Safely_ 00388001 00088901 00088925 00288908	Eject op ITM TPIU ETM CSTF	etion is e	er	ited) Cle	> ear
00> 2. Fc 00> 2. Fc 00> 3. Sa 00> 4. Ir 00> 4. Ir 00> 4. Ir 00> 4. Ir 00> 60> 70 00> 70 00 00 00 00 00 00 00 00 00	Artes (US) Afely Ej Antializ (10][3] (10][4] (10][6] (10][7] (10][7] (10][7]	B Drive ect the US E FreeRTOS E E0000000 E E0040000 E E0040000 E E0042000 E E0042000 E E0043000	B Driv +FAT ( , CID: , CID: , CID: , CID: , CID:	Valid only 51051000, 81055000, 81059000, 81059000, 81059000, 81059000, 81059000, 81059000, 81059000,	/ afte	CO2555005 00385001 0008859A1 000885925 00288908 00188961	Eject op ITM TPIU ETM CSTF TMC	ition is e	er	ited) Cie	> ≥ar
00> 2. Fc 00> 2. Fc 00> 3. Sz 00> 3. Sz 00> 4. Iri 00> 4. Iri 00> 4. Iri 00> 60> 4. Iri 00> 60> 60 00> 70 00> 70 00	afely Ej afely Ej nitializ nit	B Drive ect the US e FreeRTOS : E0000000 : E0040000 : E0040000 : E0041000 : E0041000 : E0041000 : E0043000 : E0043000	B Driv +FAT ( , CID: , CID: , CID: , CID: , CID: , CID: , CID: , CID:	ve Valid only Bioscool, Bioscool, Biossool, Biossool, Biossool, Biossool, Biossool, Biossool,	/ afte FID. PID: PID: PID: PID: PID: PID:	00200005 00388001 000889A1 000889A 00288908 00288908 00288908 00288908 00288908	Eject op ITM TPU ETM CSTF TMC TSG	rtion is e	er	ited) Cie	> ear

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	USBメモリ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	USB メモリのフォーマットを行います。
3	USB メモリの安全な抜出準備を行います。
4	RTOS、FAT の初期化を行います。

3.2.7 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、以下の手順に従い行ってください。

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(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。
- 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	38400bps
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。

💻 COM4 - Tera Term VT \_ Х ファイル(<u>F</u>) 編集(<u>E</u>) 設定(<u>S</u>) コントロール(<u>O</u>) ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>) ^ Welcome to Quick Start BLINKY example project for AP-RA6M-OA! Press 1 for Kit Information or 2 for Next Steps. 1. KIT INFORMATION AP-RA6M-OA a) Kit name: b) Kit ordering part number: AP-RA6M-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, downl oad tools and documentation, and get support: a) AP-RA6M-0A resources: apnet.co.jp/product/r b) RA product information: renesas.com/ra c) RA product support forum: renesas.com/ra/forum d) Renesas support: renesas.com/support apnet.co.jp/product/ra/ap-ra6m-0a Press 1 for Kit Information or 2 for Next Steps.

## 3.3 メモリマップ

e2 studio のプロジェクトのメモリマップを以下に示します。 サンプルプログラムは、全て共通のメモリマップを使用しています。



3.4 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\_ra6m\_0a\_ether\_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。)

#### 3.4.1 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。



 [CMSIS Pack]を選択し[次へ]を選択し、pack ファイル「AlphaProject.ap\_ra6m\_0a.4.5.0.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\_ra6m\_0a.4.5.0.pack」を選択してください。

Import CMSIS Pack			_		×
Import CMSIS Pack					2
Choose CMSIS pack to impe	ort				J
Specify pack file:					
Specify device family:					
				~	
?	< 戻る( <u>B</u> )	次へ( <u>N</u> ) >	終了( <u>F</u> )	キャンセル	

9 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。

Import CMSIS Pack 8 No Renesas Family selected		
Snarify nack files		
C:¥workspace¥ra6m¥AlphaProject.ap_ra6m_0a.4.5.0.pack	 	]
Specify device family:		
Renesas RA		~

⑤ [終了]を選択してください。

🔄 Import CMSIS	Pack				×
mport CMSIS P	ack			ſ	
Choose CMSIS p	ack to import				
Specify pack file:					
1 21					
C:¥workspace¥ra	a6m¥AlphaProject.	ap_ra6m_0a.4.5.	0.pack		
C:¥workspace¥ra Specify device fa Renesas RA	a6m¥AlphaProject. mily:	ap_ra6m_0a.4.5.	0.pack		~
C:¥workspace¥ra Specify device fa Renesas RA	a6m¥AlphaProject. mily:	ap_ra6m_0a.4.5.	0.pack		~
C:¥workspace¥ra Specify device fa Renesas RA	a6m¥AlphaProject. mily:	ap_ra6m_0a.4.5.	0.pack		~
C:¥workspace¥ra Specify device fa Renesas RA	a6m¥AlphaProject. mily:	ap_ra6m_0a.4.5.	0.pack		~

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。

	ra_workspace - e² studio							
ファイ	<sup>(</sup> ル( <u>E)</u> 編集( <u>E</u> ) ソース( <u>S</u> )	リファクタリング(T)	ナビゲート( <u>N</u> )	検索( <u>A</u> )	プロジェクト( <u>P</u> )	Renesas <u>V</u> iews	実行( <u>R</u> )	ウイン
<u></u>	新規(N) ファイルを開く(.) ファイル・システムからプロジュ 最近のファイル	」	Alt+シフト+N > >	Configur	rations	〜 オン: ぼ ▼ 29 <i>※</i>	 •   R	Π
	閉じる(C) すべて閉じる(L)	C	Ctrl+W trl+シフト+W					
	保存(S) 別名保存(A)		Ctrl+S					
	すべて保管(E) 前回保管した状態に戻す(1	( D	Ctrl+シフト+S					
	移動(V) 名前を変更(M)		F2					
ลา	电前を変更(W) 更新(F)		F5					
	行区切り文字の変換(D)		>					
ð	印刷(P)		Ctrl+P					
è	インポート(I)				א-עע	。スマート・ブラウザー	-דג 🕮	
4	エクスポート(O)							
	プロパティ(R)		Alt+Enter					
	ワークスペースの切り替え(W 再開	)	>					
_	100 J / HE HOAD							

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。

<b>e</b> インポート			×
<b>選択</b> アーカイブ・ファイルまたけディレクトリーかに新想プロジェクトを作成します			1
) - J1 / JF1/VALUA / U / I / J / J / J / J / J / J / J / J / J			
Select an import wizard:			
7ィルタ入力			
<ul> <li>ファイル・システム         フィルダーまたけアーカイブ由来のプロジェクト         アイルダーまたけアーカイブ由来のプロジェクト         アイルグースへ      </li> <li>読定         既存プロジェクトをワークスペースへ      </li> </ul>			^
> 🗁 C/C++ > 🗁 Git > 🗁 Oomph			
-			
> 25 929 > 25 チーム > 25 実行/デバッグ			*
? < 戻る(B) 次へ(N) > 終了	( <u>F</u> )	キャンセ	Jν

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。

e <sup>2</sup> インポート	– 🗆 X
プロジェクトのインポート 既存の Eclipse プロジェクトを検索するディレクトリーを選択します。	
● ルート・ディレクトリーの選択(小)	大 关昭(P)
	₩
○ 𝒴-𝒴1 𝒴・𝒴=𝒴 𝔅(	∽
プロジェクト( <u>P</u> ):	
	すべて選択( <u>S</u> )
	選択をすべて解除( <u>D</u> )
	更新( <u>E</u> )
□ イストしたノロンエクトを使薬(日)	
ワーキング・ヤット	
	±€+8 () \ 0
	新7元( <u>₩</u> )
ワーキング・セット(の):	✓ 選択( <u>E</u> )
(N) > 終了(E)	キャンセル

⑨ [約	冬了]	を選択	しま	đ	•
------	-----	-----	----	---	---

☑ インポート			×
プロジェクトをインポート 既存の Eclipse プロジェクトを検索するディレクトリーを選択します。			
<ul> <li>・ディレクトリーの選択(①: C¥workspace¥ra6m¥ap_ra6m</li> </ul>	_0a_ether 丶	✓ 参照( <u>R</u> ).	
○ アーカイブ・ファイルの選択( <u>A</u> ):		· 参照( <u>R</u> ).	
プロジェクト( <u>P</u> ):			
ap_ra6m_0a_ether_sample(C:¥workspace¥ra6m¥ap_ra6	m_0a_eth	すべて選択( <u>S</u>	)
	3	選択をすべて解除	余( <u>D</u> )
		更新( <u>E</u> )	
<	>		
オプション			
□ ネストしたプロジェクトを検索( <u>H</u> )			
□ アロフェアイモック フスペースにコニ (G) □ 完了次第、新しくインポートしたプロジェクトを閉じる(@)			
ワークスペースに既に存在するプロジェクトを隠す(j)			
ワーキング・セット			
□ ワーキング・セットにプロジェクトを追加(I)		新規( <u>W</u> )	
ワーキング・セット( <u>O</u> ):	$\sim$	選択( <u>E</u> )	
(?) < 戻る(B) 次へ(N) >	終了( <u>F</u> )	キャンセル	ŀ
	-		

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。

#### 3.4.2 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\_ra6m\_0a」であることを確認します。

-				_	
ra_workspace - ap_ra6m_0a_ether_sample/con	iguration.xml - e² studio		-		×
ファイル(E) 編集(E) ソース(S) リファクタリング(T) :	・ビゲート( <u>N</u> ) 検索( <u>A</u> ) プロジェクト( <u>P</u> ) Renesas <u>V</u> iews 実行( <u>R</u> ) ウィンドウ( <u>W</u> ) ヘルプ( <u>H</u>	Û			
🐔 🎋 🔳 🎋 デバッグ(B) 🗸 🗸	💽 ap_ra6m_0a_ether_sample Debug 🗸 🔅 📑 🖛 🔡 🐑 🕷 🖛 📸	🖷 🖉 🐵 🐝 🗛 🗸	9 <b>4</b> -		
0. • 🗞 🕪 💷 📽 🖏 🍪 💋 📸 •	🚳 • 💣 • 🕝 • 🍃 🛷 • 📴 🗉 🖷 🖢 • 🖓 • や ウ ・ ウ •				
		Q 🔡 🖬 C/C++	- 🌼 FSP Configuration	on 🎄 E	Debug
▶ プロジェクト・エクスプローラー ※ □	🔅 Jap rafm 0a ether sample) ESP Configuration 🗙		■ アウトライン S3		
	Record Connect Reclary Confirmation	0	アウトラインを提供する	アクティブな	にエディタ
✓ ⅔ ap_ra6m_0a_ether_sample	Board Support Package Configuration	Generate Project Content	ーはありません。		
> 🖑 バイナリー		Restore			
> Di Includes					
> 👝 Debug	Device Selection				
> 🧀 script	FSP version: 4.5.0				
ap_ra6m_0a_pincfg	Parente de un con con				
🔅 configuration.xml					
i ra_cfg.txt	Device: R7FA6M3AH3CFC				
> (7) Developer Assistance	RTOS: FreeRTOS ~				
	<	>			
	Summary BSP Clocks Pins Interrupts Event Links Stacks Components				
	🖹 Problems 📮 コンソール 🕴 🦓 スマート・ブラウザー 🖏 スマート・マニュアル	i,	🚮 😼 📄 🖛	📑 👻	- 8
	RA FSP		• .		
	Installing support files for AlphaProject.ap_rabm_0a.4.4.0-ap01000	00.packInstallation	complete.		^
	<				>
		http://tool-su	pport5.1/content.xr	nl.xz	-

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。

ra_workspace - ap_ra6m_0a_ether_sample/conf	iguration.xml - e² studio	- 🗆 ×		
ファイル(E) 編集(E) ソース(S) リファクタリング(T) つ	ビゲート( <u>N</u> ) 検索( <u>A</u> ) プロジェクト( <u>P</u> ) Renesas <u>V</u> iews 実行( <u>R</u> ) ウィンドウ( <u>W</u> ) ヘルプ( <u>H</u> )			
🐔 🎋 🔳 🎄 デバッグ(B) 🗸 🗸	💽 ap_ra6m_0a_ether_sample Debug 🗸 🄅 📑 🕶 🔚 🐘 🛛 👟 🥵 🖛 🚮 🟪 🗽 🤅	) i 🗞   🐐 🕶 💁 🕶		
🔍 • 核 🕪 💷 📽 🖏 🕹 😻 🥖 📸 •	· 🚳 • 🖻 • @ • ! 🕭 🛷 • ! 🖗 🗉 π ! 🖉 • 🖗 • 🖓 • !			
	٩ ا	含 I 昆 C/C++ 徳 FSP Configuration 参 Debug		
🔁 プロジェクト・エクスプローラー 🛛 🖳 🗆	(ap_ra6m_0a_ether_sample) FSP Configuration ×	E P5h512 8		
► 🕏 🍸 🕴	Board Support Package Configuration Generate Proj	マウトラインを提供するアクティブなエディタ ect Content ーはありません。		
> 梁介 バイナリー		Restor		
> 👔 Includes		B <sub>0</sub> Restor		
> 👝 Debug	Device Selection			
> 🧽 script	FSP version: 4.5.0 V Board Details			
ap_ra6m_0a.pincfg	Board: ap ra6m 0a			
💮 configuration.xml	Devices R7FA6M3AH3CFC			
> ⑦ Developer Assistance				
	RIOS: FreeRIOS V			
	·			
	Summary BSP Clocks Pins Interrupts Event Links Stacks Components			
	Problems      Problems      マンソール      X       X       X       X			
	RA FSP			
	Installing support files for AlphaProject.ap_ra6m_0a.4.4.0-ap010000.packI	nstallation complete.		
	<	>		
http://tool-support5.1/content.xml.xz 🔤 🖏				

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。

ra_workspace - ap_ra6m_0a_ether_sample/cont	iguration.xml - e² studio		- 🗆 ×
ファイル(E) 編集(E) ソース(S) リファクタリング(T) :	ŀビゲート( <u>N</u> ) 検索( <u>A</u> ) プロジェクト( <u>P</u> ) Renesas <u>V</u> iews 実行( <u>R</u> ) ウィンドウ(W) ヘルプ(F	D	
🐔 🗱 🔳 🕸 รี่เก็ษที่(B) 🗸	💽 ap_ra6m_0a_ether_sample Debug 🗸 🄅 🗋 🖛 🔚 🐑 🖌 🗞 🖛 💦	🖦 🖉 🗞 💠 🗣 🗸	
🔍 🔻 🎋 🕪 💷 😭 🖏 🕹 😻 🥖 📸 י	· 🚳 🕶 🗃 👻 🐨 🔊 🔗 🕶 💷 💷 📲 🖢 🐨 🖓 🕶 💬 🕶		
		Q 🛛 😰 🗖 C/C++ 🌼 FSP (	Configuration 🛛 🎋 Debug
🔁 プロジェクト・エクスプローラー 🛛 👘 🗖	🔅 [ap_ra6m_0a_ether_sample] FSP Configuration 🗙	□ 🔡 アウトラ	デイン 🛛 🗖 🗖
Image: Section Content in the section of the se	Board Support Package Configuration	Content Project Content アウトライン	/を提供するアクティブなエディタ せん。
> 繰 バイナリー		Reston	
> B) Includes			
> 👝 Debug	Device Selection		
> 🧀 script	FSP version: 4.5.0 V Board Details		
ap_ra6m_0a.pincfg	Board: an rafm 0a		
🔅 configuration.xml			
ra_ctg.txt	Device: K/FA0MSAHSCPC		
y () bereicher Assistance	RTOS: FreeRTOS V		
		>	
	Summary BSP Clocks Pins Interrupts Event Links Stacks Components		
	Problems □ コンソール ※ ▲ スマート・ブラウザー □ スマート・マニュアル	B. A. 19	📑 🗉 🛪 📑 🗖
	KAPSP Installing support files for AlphaProject.ap ra6m 0a.4.4.0-ap0100	00.packInstallation complete	e. ^
	9 - + +		
			~
	<		>
		http://tool-support5.1	/content.xml.xz 📰 🖫

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。



#### 3.4.3 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:ON ボード上の SDRAM を使用する



JSW2 : SGL シングルチップモード

- 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。

ra_workspace - ap_ra6m_0a_ether_sample/conf	iguration.xml - e² studio		– 🗆 X	
ファイル(E) 編集(E) ソース(S) リファクタリング(T) つ	ビゲート( <u>N</u> ) 検索( <u>A</u> ) プロジェクト( <u>P</u> ) Renesas <u>V</u> iews 実行( <u>R</u> ) グ	フィンドウ( <u>₩)</u> ヘルプ( <u>H</u> )		
🐔 🔯 🔳 🔅 デバッグ(B) 🗸 🗸	💽 ap_ra6m_0a_ether_sample Debug 🗸 🔅 🗋 🕶 🗒 🐚	) 🛛 🕶 💊 🕶 📾 📲 🔍 🖾 😵 🔹	· 💁 🗸	
0., • 🗞 🕪 💷 😭 🖏 🕹 🖗 🥖 🖬 🔂	<mark>83 ▼ 63 ▼ 63 ▼ 129                                  </mark>	> c>		
		Q   😰   🗟 📿	🕂 🌐 ESP Configuration 🛛 🌟 Debug	
🔁 לוויד 🗠 🗆 🖻	🔅 [ap_ra6m_0a_ether_sample] FSP Configuration 🗙	= =	📴 アウトライン 🛛 🧧 🗖	
✓ 🚰 אך 8 ✓ 🚰 ap_ra6m_0a_ether_sample	Board Support Package Configuration	Generate Project Content	アウトラインを提供するアクティブなエディタ ーはありません。	
> 繰れたオナリー		Restor		
> Mu includes > 🔑 src				
> 👝 Debug	Device Selection			
> 🧽 script	FSP version: 4.5.0 V	Board Details		
ap_ra6m_0a.pincfg	Board: ap_ra6m_0a 🗸 🚵			
configuration.xml ra cfg.txt	Device: R7FA6M3AH3CFC			
> ⑦ Developer Assistance	PTOS. EmpPTOS			
	NOS. TEENOS			
	<	>		
	Summary BSP Clocks Pins Interrupts Event Links Stacks Con	nponents		
	👔 Problems 📮 コンソール 🙁 🧠 スマート・ブラウザー 👊 スマート・	マニュアル 🖹	· 🔐 🖻 🛃 🖻 🕶 🗂 🗖	
	RA FSP			
	Installing support files for AlphaProject.ap_ra6m	_0a.4.4.0-ap010000.packInstallation	complete.	
			~	
	<		>	
		http://tool-su	ipport5.1/content.xml.xz 📰 💼	

- ⑤ [Renesas GDB Hardware Debug]の [ap\_ra6m\_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\_ra6m\_0a\_XXXX Debug
  - [プロジェクト]: ap\_ra6m\_0a\_XXXX
  - [C/C++アプリケーション]: Debug¥ ap\_ra6m\_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。

圆 デバッグ構成	— 🗆 X
構成の作成、管理、および実行	·
<ul> <li>         ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・</li></ul>	名前(1): [ap_ra6m_0a_ether_sample Debug
< > 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(い) 適用(Y)
?	デバッグ( <u>D)</u> 閉じる

 ⑥ [Debugger] タブを選択し、 [Debug hardware] が [J-Link ARM] 、 [Target Device] が「R7FA6M3AH」に 設定されていることを確認してください。

📴 デバッグ構成	— 🗆 X
構成の作成、管理、および実行	
マイルタスカ         E       C/C++ アブリケーション         C       C/C++ アブリケーション         E       ASE Script         C       GDB OpenOCD Debugging         E       GDB Arby-ション         I Java アブリケーション       I Java アブリケーション         E       Renesas GDB Hardware Debu         C       ap_ra6m Oa_ether_sample         E       Renesas Simulator Debugging         U H=-ト Java アブリケーション       定 起動グループ	名前(M): ap.ra6m_0a_ether_sample Debug ③ メイン ☆ Debugger → Startup ① 共通(C) → ソース Debug hardware: J-Link ARM ✓ Target Device: R7FA6M3AH … GDB Settings Connection Settings デバッグ・ツール設定 GDB 接続設定: ④ 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 サーバーへ接続 GDB ポート番号: 61234 GDB GDB コマンド: arm-none-eabi-gdb 参照… 変数…
< > 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

⑦ [デバッグ]を選択します。

📴 デバッグ構成	— 🗆 X
構成の作成、管理、および実行	T
<ul> <li>         ・         ・         ・</li></ul>	名前(M): ap_ra6m_0a_ether_sample Debug → X12 な Debugger → Startup □ 共通(Q は ソース Debug hardware: J-Link ARM ~ Target Device: R7FA6M3AH … GDB Settings Connection Settings デパッグ・ツール設定 GDB 接続設定: ④ 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 サーバーへ接続 GDB ポート番号: 61234 GDB GDB コマンド: arm-none-eabi-gdb 参照… 変数… Step Mode
< > 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(D) 閉じる

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。

ra_workspace - ap_ra6m_0a_ether_sample/ra/fs	p/src/bsp/cmsis/Device/RENESAS/Source/startup.c - e² studio			_		×
ファイル( <u>F</u> ) 編集( <u>F</u> ) ソース( <u>S</u> ) リファクタリング(T) カ	・ピゲート( <u>N</u> ) 検索( <u>A</u> ) プロジェクト( <u>P</u> ) Renesas <u>V</u> iews 実行( <u>R</u> ) ウィンドウ( <u>W</u> ) ヘルプ( <u>H</u> )					
🔦 🔯 🔳 🏘 รีไไซว์(B) 🗸	💽 ap_ra6m_0a_ether_sample Debug 🗸 🄅 📑 😭 🖛 🔚 🔞 🗸 🗞 🗸 📾 🗄 🏪	`@   [	N 🗉 🛛 🗤	3. 🤉 .e   i>	5 7 6	1
🞉   🎋 🕶 💁 🕶 🛰 🕶 💷 😭 🚱	८ । ≫ : ∅ : ⊵ ⊘ ∕ ▼ : ⊿ : ⊵ ▼ ?! ▼ ! ↓ : □ · !! ⊂! (> → ↔ ▼   🛃	Q I	😰   📴 C/C++	FSP Configurat	ion 🞄 D	Debug
🐞 デバッグ 🗙 📄 🚍 🦌 🖻 🗖	@ [ap_ra6m_0a_ether_sample] FSP Configuration	•	(x)= 変数 💥 🤇	🂊 ブレー 🛯 🔁 プロジ	않.式 '	
✓ ▶ ap_ra6m_0a_ether_sample Debug [Renesas G	61 ©void Reset_Handler (void)	^		<b>‱ ⇒t</b> a	8   📬 เ	8
	63 64 66 65 65 65 66 67 600002ca2 68 69 69 69 69 69 69 60 61 61 70 6 71 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7		名前	ᆋ		~
	<pre>72 00002ca6 } 73 73 74 75</pre>	i				
	· · · · · · · · · · · · · · · · · · ·	, Ť	<	>	<	>
	■ コンソール ☆ ① Debug Shell ② Problems ④ スマート・ブラウザー ap_raGm_0a_ether_sample Debug [Renesas GDB Hardware Debugging] Option Function Select, writing to address 0x00000400 with data ffffff SCCMPUxxx, writing to address 0x00000408 with data fcffffffffffffffff グウンロード終了 ハードウェア・ブレークポイントをアドレス0x106cCi設定します。	<b>* %</b>	dffff fffffffff		•	
< >	<					>
			1			-

⑦ 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

・RA ファミリおよび RA6M3 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<sup>®</sup>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。

# 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