Renesas Synergy[™]

LCD-KIT サンプルプログラム解説 (AP-S5D9-0A)

3.2版 2023年10月02日

1. 相	腰	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	3
2. ‡	トンプルプログラムの構成	ł
2.1	フォルダ構成	1
2.2	ファイル構成	5
3. ±	トンプルプログラムについて	7
3. ±	・ンプルプログラムについて	7
3. ± 3.1 3.	トンプルプログラムについて	7 7
3. ± 3.1 3. 3.	トンプルプログラムについて 7 動作説明	7 7 7 3
3. ± 3.1 3. 3. 3.	トンプルプログラムについて 動作説明 1.1 サンプルプログラム動作説明 1.2 LCD-KIT-C01/C02のキャリブレーション 1.3 LCD-KIT の選択・設定	7 7 3 3
3. ± 3.1 3. 3. 3. 3.2	トンプルプログラムについて 動作説明 1.1 サンプルプログラム動作説明 1.2 LCD-KIT-C01/C02のキャリブレーション 1.3 LCD-KIT の選択・設定 メモリマップ	7 7 3 3 9
 3.1 3.1 3. 3.2 3.3 	トンプルプログラムについて 動作説明	7 7 3 3 9 0
 3.1 3.1 3. 3.2 3.3 3.3 3.3 	サンプルプログラムについて 7 動作説明 1.1 1.1 サンプルプログラム動作説明 7 1.2 LCD-KIT-C01/C02 のキャリブレーション 8 1.3 LCD-KIT の選択・設定 8 メモリマップ 9 e2 studio を用いたプロジェクトのビルド・デバッグ 10 3.1 インポート方法 10	7 7 3 3 9 0
 3.1 3.1 3. 3.2 3.3 3.3 3.3 3. 	サンプルプログラムについて	7 7 3 3 9 0 5

1. 概要

1.1 概要

本アプリケーションノートでは、AP-S5D9-0A(S5D9 CPU)を用いて、Renesas Synergy™の Synergy Software Package を 使用した LCD-KIT サンプルプログラムについて解説します。

ソフトウェアは、静電容量式 LCD キット(LCD-KIT-B01、LCD-KIT-B02、LCD-KIT-D02)および抵抗膜式 LCD キット (LCD-KIT-C01、LCD-KIT-C02)に対応しています。

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	動作内容
AP-S5D9-0A	・グラフィック LCD コントローラ	グラフィック表示
	(GLCDC)	
	・シリアルコミュニケーションインタ	LCD-KIT との通信(簡易 I2C 通信)
	フェース (SCI3)	
LCD-KIT	・LCD パネル	・各画面の表示
· LCD-KIT-B01	・タッチパネル(静電容量式・抵抗膜	・タッチ検出
· LCD-KIT-B02	式)	
· LCD-KIT-C01	・バックライト	・バックライトの点灯
· LCD-KIT-C02	・スイッチ	・各種機能の切り替え
· LCD-KIT-D02	・ブザー	・ブザー音出力
	(LCD-KIT-B01/B02/C01/C02)	
	・スピーカ(LCD-KIT-D02)	・音声出力

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。



※AP-S5D9-0A と J-Link を直接接続することはできません。

AP-S5D9-0A 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-S5D9-0A 製品ページ https://www.apnet.co.jp/product/synergy/ap-s5d9-0a.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「Synergy Software Package(以下、SSP)」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、SSP、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	v2021-07	_
SSP	v2.1.0	_
GCC ARM Embedded	v7.2.1	_
AP-S5D9-0A 用 Custom BSP	v2.1.0 - ap010000	_

デバッガ	ファームバージョン	備考
J-Link	V10	Segger Microcontroller Systems 社

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
LCD-KIT-B01/B02/C01/C02	¥sample¥ap_s5d9_0a_sample_lcdkit_b01_c01
サンプルプログラム	
プロジェクトフォルダ	
LCD-KIT-D02	¥sample¥ap_s5d9_0a_sample_lcdkit_d02
サンプルプログラム	
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下記のようなフォルダ構成になっています。

sa	Imple	
-	Custom BSP	AP-S5D9-0A 用 Custom BSP フォルダ
┢	ap_s5d9_0a_sample_lcdkit_b01_c01	LCD-KIT-B01/B02/C01/C02 サンプルプログラムフォルダ
		設定ファイルフォルダ
	data	LCD-KIT-D02 用 音声データ
	script	リンカ設定ファイルフォルダ
	L src	ソースファイルフォルダ
	Llcdkit	LCD-KIT デバイスドライバフォルダ
L	— ap_s5d9_0a_sample_lcdkit_d02	LCD-KIT-D02 サンプルプログラムフォルダ
	.settings	設定ファイルフォルダ
	data	LCD-KIT-D02 用 音声データ
	script	リンカ設定ファイルフォルダ
	src	ソースファイルフォルダ
	Llcdkit	LCD-KIT デバイスドライバフォルダ

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。 <¥sample¥Custom BSP フォルダ内> AlphaProject.ap_s5d9_0a.2.1.0 ··· AP-S5D9-0A 用 Custom BSP -ap010000.pack <¥sample¥ap_s5d9_0a_sample_lcdkit_b01_c01 フォルダ内> • • • .cproject CPROJECT ファイル .project • • • PROJECT ファイル configuration.xml ... Synergy コンフィギュレータファイル ap_s5d9_0a(QSPI_ON)_R7FS5 ···· AP-S5D9-0A 用 QSPI 対応ピンコンフィグファイル D97E3A01CFC.pincfg ap_s5d9_0a(SDHI_ON)_R7FS5 ···· AP-S5D9-0A 用 SDHI 対応ピンコンフィグファイル D97E3A01CFC.pincfg R7FS5D97E3A01CFC.pincfg • • • S5D9 CPU 用 デフォルト ピンコンフィグファイル ※ AP-S5D9-0A 用の設定はしてありません。 ap_s5d9_0a_sample_ ・・・ AP-S5D9-0A LCD-KIT-B01/B02/C01/C02 サンプルプログラム lcdkit b01 c01 Debug.jlink デバッグおよびランタイム設定ファイル ap_s5d9_0a_sample_lcdkit_ ... AP-S5D9-0A LCD-KIT-B01/B02/C01/C02 サンプルプログラム b01_c01 Debug.launch デバッグおよびランタイム設定ファイル

<¥sample¥ap_s5d9_0a_sample_lcdkit_d02 フォルダ内>

.cproject	•••	CPROJECT ファイル
.project	•••	PROJECT ファイル
configuration.xml	•••	Synergy コンフィギュレータファイル
ap_s5d9_0a(QSPI_ON)_R7FS5	•••	AP-S5D9-0A 用 QSPI 対応ピンコンフィグファイル
D97E3A01CFC.pincfg		
ap_s5d9_0a(SDHI_ON)_R7FS5	•••	AP-S5D9-0A 用 SDHI 対応ピンコンフィグファイル
D97E3A01CFC.pincfg		
R7FS5D97E3A01CFC.pincfg	•••	S5D9 CPU 用 デフォルト ピンコンフィグファイル
		※ AP-S5D9-0A 用の設定はしてありません。
ap_s5d9_0a_sample_	•••	AP-S5D9-0A LCD-KIT-D02 サンプルプログラム
lcdkit_d02 Debug.jlink		デバッグおよびランタイム設定ファイル
ap_s5d9_0a_sample_	•••	AP-S5D9-0A LCD-KIT-D02 サンプルプログラム
lcdkit_d02 Debug.launch		デバッグおよびランタイム設定ファイル

※以下、「LCD-KIT-B01/C01/C02 サンプルプログラム」のフォルダおよびファイルに関して記しますが、 特に記載のない場合、「LCD-KIT-D02 サンプルプログラム」のファイルはフォルダ名以外共通となります。

<pre><+sumple+up sous ou sumple icurit boi coi+uutu >//////////////////////////////////</pre>

Sound0.bin	•••	音声データ 0(LCD-KIT-D02 用)
Sound1.bin	•••	音声データ 1(LCD-KIT-D02 用)
Sound2.bin		音声データ 2(LCD-KIT-D02 用)

<¥sample¥ap_s5d9_0a_sample_lcdkit_b01_c01¥script フォルダ内> r7fs5d97e3a01cfc.ld ・・・・ リンカスクリプトファイル

<¥sample¥ap_s5d9_0a_sample_lcdkit_b01_c01¥src フォルダ内>

hal_entry.c		hal_entry 関数ソースファイル
ioport_app.c	•••	I/O ポート制御ソースファイル
ioport_app.h		I/O ポート制御ヘッダファイル
<pre>lcd_thread_entry .c</pre>		lcd_thread_entry 関数ソースファイル
lcd_thread_main .c	•••	LCD アプリケーションソースファイル
common_app.h	•••	共通ヘッダファイル

<¥sample¥ap_s5d9_0a_sample_lcdkit_b01_c01¥src¥lcdkit フォルダ内>

lcdkit.h	•••	LCD-KIT デバイス情報ヘッダファイル
		※このファイル内で使用する LCD-KIT を定義します。
		「LCD-KIT-B01/B02/C01 サンプルプログラム」と
		「LCD-KIT-D02 サンプルプログラム」では
		使用する LCD-KIT のデフォルトの定義が異なります。
lcdkit_app.c		LCD-KIT デバイスドライバソースファイル
lcdkit_app.h	•••	LCD-KIT デバイスドライバヘッダファイル
lcdkit_b01.c	•••	LCD-KIT-B01 デバイスドライバソースファイル
lcdkit_b01.h	•••	LCD-KIT-B01 デバイスドライバヘッダファイル
lcdkit_b02.c	•••	LCD-KIT-B02 デバイスドライバソースファイル
lcdkit_b02.h	•••	LCD-KIT-B02 デバイスドライバヘッダファイル
lcdkit_c01.c	•••	LCD-KIT-C01 デバイスドライバソースファイル
lcdkit_c01.h		LCD-KIT-C01 デバイスドライバヘッダファイル
lcdkit_d02.c		LCD-KIT-D02 デバイスドライバソースファイル
lcdkit_d02.h	•••	LCD-KIT-D02 デバイスドライバヘッダファイル

- 3. サンプルプログラムについて
 - 3.1 動作説明
 - 3.1.1 サンプルプログラム動作説明

サンプルプログラムは、下記の動作を行います。

● グラフィック表示

タッチパネル入力を検出すると、以下の順でグラフィックの切り替えを行います。

- ・カラーバー
- ・グラデーション
- ・市松模様
- ・カラーバー(以下、繰り返し)
- ブザー (LCD-KIT-B01/B02/C01/C02)

SW 入力に	応じて、	ブザー音を出力します。
SW1	•••	ブザー音①を出力します(ブザー音①:低)
SW2		ブザー音②を出力します(ブザー音②:中)
SW3		ブザー音③を出力します(ブザー音③:高)

● 音声出力(LCD-KIT-D02)

SW 入力に応じて、音声を出力します。
 SW1 ・・・ 「ピンポーン」
 SW2 ・・・ 「ブブー」
 SW3 ・・・ 「ありがとうございました」

• LED

LED は LCD-KIT の制御に合わせて点灯・消灯します。

(電源投入		LD1:消灯	LD2 : 消灯)
LCD-KIT 初期化完了		LD1 : 点灯	LD2 : 消灯
- LCD-KIT 初期化失敗	•••	LD1 : 点滅	LD2:点滅
カラーバー表示開始	•••	LD1 : 点灯	LD2 : 点灯

3.1.2 LCD-KIT-C01/C02 のキャリブレーション

LCD-KIT-C01/C02 は、抵抗膜式タッチパネル付き LCD ボードです。 静電容量式の LCD-KIT とは異なり、キャリブレーション処理を行う必要があります。

LCD-KIT-C01/C02 使用時に本サンプルプログラムの動作を開始すると、ピピッと短いブザー音がした後、 黒い画面の左上に白色の十字のポインタが表示されます。

ユーザはそのポインタに対してタッチ&リリースをしてください。

その後も順に右上、左下、右下の順にポインタが表示されますので、同様にユーザは各ポインタに対して タッチ&リリースをしてください。



画面に対して四回タッチ&リリースをするとキャリブレーション処理は終了です。 その後の動作は静電容量式の LCD-KIT と同様です。

・キャリブレーション処理の補正は、リリースする直前のタッチ情報を基に計算されます。

・LCD-KIT-C01はマルチタッチに対応していません。同時に複数箇所に触れると誤動作する場合があります。

3.1.3 LCD-KIT の選択・設定

本サンプルプログラムは、使用する LCD-KIT によって、下記ソースファイルの変更が必要な場合があります。 変更後は、e2 studio 上で [プロジェクトをクリーンにする]を行ってから [プロジェクトのビルド]を選択して、 必ずプロジェクトを再ビルドしてください。

- LCD-KIT-B01/B02/C01/C02 を使用する場合 プロジェクト「¥sample¥ap_s5d9_0a_sample_lcdkit_b01_c01」を使用します。 デフォルトでは LCD-KIT-B01 を使用する設定になっています。
- LCD-KIT-D02 を使用する場合 プロジェクト「¥sample¥ap_s5d9_0a_sample_lcdkit_d02」を使用します。 デフォルトで LCD-KIT-D02 を使用する設定になっています。

<B01/B02/C01/C02>¥sample¥ap_s5d9_0a_sample_lcdkit_b01_c01¥src¥lcdkit¥lcdkit.h(18~28 行目) <D02> ¥sample¥ap_s5d9_0a_sample_lcdkit_d02¥src¥lcdkit¥lcdkit.h(18~28 行目)

18 :	#define LCDKIT_NONE	(0)	
19:	#define LCDKIT_B01	(LCDKIT_NONE + 1)	←LCD-KIT-B01 と接続時の設定値
20:	#define LCDKIT_B02	(LCDKIT_B01 + 1)	←LCD-KIT-B02 と接続時の設定値
21:	#define LCDKIT_C01	(LCDKIT_B02 + 1)	←LCD-KIT-C01/C02 と接続時の設定値
22 :	#define LCDKIT_D02	(LCDKIT_C01 + 1)	←LCD-KIT-D02 と接続時の設定値
		中略	
28 :	#define LCDKIT (LCDKIT	_B01) /* <- Please specify the LCD-KIT to use. */	←赤字の値を適切に設定してください

3.2 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。





3.3 e2 studio を用いたプロジェクトのビルド・デバッグ

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

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

(下記で表示される図は「ap_s5d9_0a_sample_can」をデバッグ・ビルドする際の例として表示しています。 プロジェクト名等は、ビルド・デバッグを行うサンプルプログラムにより変化します。)

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





 [CMSIS Pack]を選択し[次へ]を選択し、pack ファイル「AlphaProject.ap_s5d9_0a.2.1.0-ap010000.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、⑤へお進みください。

インポート	_		×
選択 Import a Synergy CMSIS Pack into e2 studio		Ľ	5
インボート・ウィザードの選択(<u>S</u>):			
7ィルタ入力			
 			< >
(ア) (ア) 終了(E)		キャンセ	IL

③ [Import Synergy 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_s5d9_0a.2.1.0-ap010000.pack」を選択してください。

Import Synergy	CMSIS Pack			_		×
Import Synergy	CMSIS Pack				ſ	
Choose CMSIS pac	k to import					
Specify pack file:						
?	< 戻る(<u>B</u>)	次へ(<u>N</u>) >	終了(<u>F</u>)		キャンセ	JU



④ [終了]を選択してください。



⑤ 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。



⑥ [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。

(1) インポート	_		×
選択 Import a Synergy CMSIS Pack into e2 studio		Ľ	
インポート・ウィザードの選択(<u>S</u>):			
7ብሥንአታ			
 ◇ ご 一般 ◇ CMSIS Pack ◇ Renease Webサイト上のサンブル・プロジェクト ◇ Renesas CA78K0R (CS+) プロジェクト ◇ Renesas CC-RX/CC-RL (CS+) プロジェクト ◇ アーカイブ・ファイル ◇ アーカイブ・ファイル ◇ ファイル・シュテム ◇ フォルダーまたけアーカイブ由来のブロジェクト ◇ 酸存プロジェクトをワークスペースへ ◇ ジ C/C++ > Oomph ◇ Tracing ◇ XMI 			~
(ア) (ア) 終了(E)		キャンセル	

⑦ [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。

📴 インポート			_	- 0	×
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	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(<u>1</u>):			~	参照	쮩(<u>R</u>)
○ ァーカイブ・ファイルの選択(<u>A</u>):			~	参照	즽(<u>R</u>)
プロジェクト(<u>P</u>):					
				すべて選	択(<u>S</u>)
			ž	選択をすべ	て解除(<u>D</u>)
				更新	·(<u>E</u>)
オプション □ ネストレたプロジェクトを検索(H))				
プロジェクトをワークスペースにコ	′ <u>²</u> −(<u>C</u>)				
□ 完了次第、新しくインポートした	プロジェクトを	:閉じる(<u>o</u>) ヨナい			
ローキング・セット	ロジェクトを展	≣9(<u>1</u>)			
フィンフ ビフト □ □ □ = ± \) ガ・セット(= プロジェクト)	を:自力((T)			奈坦の	10
	2,22/JH(<u>1</u>)			388+D//	<u>v</u>)
9-499729P(<u>0</u>);			~	選択(]	_)

アプリケーションノート AN1820

⑧ [終了]を選択します。

ゴンポート		
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。		
 ・ディレクトリーの選択(①: C¥workspace¥Synergy¥ap_s5d9_0a_sz アーカイブ・ファイルの選択(A): プロジェクト(P): 	• ~	参照(<u>R</u>) 参照(<u>R</u>)
ap_s5d9_0a_sample_can(C:¥workspace¥Synergy¥ap_s5d9_0a_sa	選	すべて選択(<u>S</u>) 択をすべて解除(<u>D</u>) 更新(F)
オプション ネストしたプロジェクトを検索(出) 「プロジェクトをワークスペースにコピー(C) 「完了次第、新しくインポートしたプロジェクトを閉じる(o) 「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(<u>□</u>) ワーキング・セット(<u>0</u>):	/	新規(<u>W</u>) 選択(<u>E</u>)
? (N) ※ (N) ※ 終了(E)		キャンセル

⑨ ナビゲーションウィンドウにサンプルプログラムのプロジェクトが追加されていることを確認します。



以上でプロジェクトのインポートは完了です。

3.3.2 ビルド方法

① プロジェクトのコンフィギュレータファイルを開きます。



② [BSP] タブを開きます。





③ [BSP]タブで [Board] が「ap_s5d9_0a」であることを確認します。



④ [Generate Project Content] をクリックし、自動作成ファイルを出力して設定をプロジェクトに適用します。

syn_workspace - ap_s5d9_0a_sample_can/co	nfiguration.xml - e² studio	- 🗆 ×
ファイル(E) 編集(E) ソース(S) 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>) ウィンドウ(<u>W</u>)	ヘルプ(円)
後 一 (アグ(B)	🕐 💽 ap_s5d9_0a_sample_can Debug 🛛 🗸 🏟 🗄 😁 🕶 🔚 🔞 🛛 😵 🕶 🖉	§ ▼ 副 X X 林 ▼ 🂁 ▼ फ ▼ 核 🕪 💷 📽 🖏 🕹
👋 🔅 🥖 💼 + 🛍 + 🖬 + 🚱 + 🖉	9 🛷 ▼ 🗟 🔲 ୩ 🗄 ▼ 🖓 ▼ 🏷 🗇 🗘 ▼ 🔿 ▼ 📑	🔍 🛛 😰 🛛 🔂 C/C++ 🏶 Synergy Configuration 🚸 Debug
🔁 プロジェクト・エクスプローラー 🛛 🖳 🗆	🔅 [ap_s5d9_0a_sample_can] Synergy Configuration 🗙	
E ⊈ 7 8 ✓ 2 ap_s5d9_0a_sample_can	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
> 🔊 Includes > 😂 src		図 Re ターはありません。
> 🗁 script	Device Selection	
ap.5342.00.3ample.can DebugJinik ap.5542.00.3ample.can DebugJinik ap.5542.00.3mple.can DebugJianch ap.5542.004(SDH1_ON)_R7FSSD97E3A01 can provide the state of t	SSP version: 2.1.0 Board Detai Board: ap_s5d9_0a Device: R7FSSD97E3A01CFC Summary BSP Clocks Pins Threads Messaging Components	ls
	Problems ロンソール 23 (余) スマート・ブラウザー ロッスマート・マニュアル	
	<終了> ap_s7g2_0a_sample_can Debug [Renesas GDB Hardware Debugging] Re	nesas GDB server (Host) (Terminated 2021/11/08 9:48:39)
	R7FS7G27H へ接続, ARM Target GDBServer エンディアン : リトル	^
	Iミュレーターからの電源供給 : Off ターゲット接続開始 ターゲット接続於了 GDR - 58463	
< >		>
		8

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。

syn_workspace - ap_s5d9_0a_sample_can/configuration.xml - e	² studio		- 🗆 ×
ファイル(E) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検	素(<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>	<u>ヘルプ(H)</u>	
参 「 「 「 「 「 「 「 」 「 」 「 」 「 」 「 」 「 」 」 「 」 」 「 」 」 」 「 」 」 」 」 「 」 」 」 」 」 」 」 「 」 」 」 」 」 「 」 」 」 」 「 」 」 」 」 」 」 「 」 」 」 」 」 」 」 」 」 」 」 」 」 」 」	a_sample_can Debug 🛛 🗸 🏟 🗄 🔂 🕶 🦓	🗞 🕶 🔒 🔍 🔌 🛛 🚸 🕶 💁	• 🔍 • ち い 🖬 🖏 🕹
🚳 🕼 🖋 📸 • 🛍 • 健 • 🎯 • 🅭 🛷 • 💀 🗐	¶ 월 ▼ 祠 ▼ ♥> <> ♥ <> ▼ 🛃	Q हा दि पि दि दे र दे के	Synergy Configuration 🛛 🎄 Debug
🚹 プロジェクト・エクスプローラー 🛛 👘 🔲 🛱 [ap_s5d9_0a	sample_can] Synergy Configuration 🗙	- 8	ד דאליד 🗉 🗖
► 🛠 🏹 🕴 Board Supp	oort Package Configuration	O Generate Project Content	8
> 🔊 Includes		🛃 Re	ターはありません。
> 🧽 script	0.0		
ap_s5d9_0a_sample_can Debug_jlink	Brand Datail	1-	
ap_s5d9_0a(QSPI_ON)_R7FS5D97E3A01(2.1.0 V	IS	
ap_s5d9_0a(SDHI_ON)_R7FS5D97E3A010 Board:	ap_s5d9_0a 🗸 🔛		
configuration.xml R7555D97E3A01CEC pipefa Device:	R7FS5D97E3A01CFC		
synergy_cfq.txt	L		
> ⑦ Developer Assistance			
<		>	
Summary BSP	Clocks Pins Threads Messaging Components		
R Problems	🛛 コンソール 🙁 🁒 スマート・ブラウザー 👒 スマート・マニュアル	🛛 🗙 💥 🗟 🚮 😥	🗩 🍠 🖻 🕶 😁 🗖
<終了> ap_s7g2_	0a_sample_can Debug [Renesas GDB Hardware Debugging] Re	enesas GDB server (Host) (Termina	ted 2021/11/08 9:48:39)
R7FS7G27H A	接続, ARM Target		^
GDBSe	inver エンディアン : リトル ーカーからの意道(供給 - off		
ターゲット接続開始			
ターゲット接続終了			
< >> <			>
		8	
		Conter .	

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

3.3.3 デバッグ方法

- 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。





- ⑤ [Renesas GDB Hardware Debug]の [ap_s5d9_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_s5d9_0a_XXXX Debug
 - [プロジェクト]: ap_s5d9_0a_XXXX
 - [C/C++アプリケーション]: Debug¥ ap_s5d9_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。

📴 デバッグ構成			— 🗆 X
構成の作成、管理、および実行	Ŧ		Ť.
 ・ ・ ・	名前(N): ap_s5d9_0a_sample_can Debug	(C) 「● ソース」 変数(ゾ) プロジェクトの検索(出) ○ 自動ビルドを無効にする ワークスペース設定の構成	参照(<u>b</u>) 参照(<u>b</u>)
< > 21 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(<u>V</u>)	適用(<u>Y</u>)
?		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「R7FS5D97E」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 Run Break Time Measurement Operation	T requency must not be empty and must be between 0.001_10000.0MHz.
 ・ ・ ・	名前(N): ap_55d9_0a_sample_can Debug アイン ひ Debugger い Startup 二 共通(C) ジ ソース Debug hardware: J-Link ARM ~ Target Device: R7FSSD97E … GDB Settings Connection Settings デバッグ・ツール設定 GDB 接続設定: ④ ローカル GDB サーパーを自動起動 ホスト名または IP アドレス: localhost 〇 リモート GDB サーパーへ接続 GDB ポート番号: 61234 GDB GDB コマンド: arm-none-eabi-gdb 参照… 変数…
< > 21 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(<u>V</u>) 適用(<u>Y</u>)
?	デバッグ(<u>D)</u> 閉じる



⑦ [デバッグ]を選択します。

@ ≓げ⊎ガ進武	
構成の作成、管理、および実 Run Break Time Measurement Operat	Fr Ing Frequency must not be empty and must be between 0.001_10000.0MHz.
 アルタ入力 C (C++ アブリケーション) C (C++ アブリケーション) C (C++ リモート・アブリケーション) E ASE Script C GOB OpenOCD Debugging (R C GOB Simulator Debugging (R C GOB N-ドウェア・デバッギング Java アブリケーション Java アブリケーション Renesas GoB Hardware Debuging C Renesas Simulator Debugging U H=- ト Java アブリケーション E 起動グルーブ 	名前(N): ap_55d9_0a_sample_can Debug メイン ◆ Debugger ◆ Startup 1 共通(① 15/ ソース) Debug hardware: J-Link ARM Target Device: R7FSSD97E GDB Settings Connection Settings デパッグ・ツール設定 GDB 接続設定: ODB 接続設定: ODB サーバーを自動起動 ホスト名または IP アドレス: Iocalhost 〇 リモート GDB サーバーへ接続 GDB ボート番号: 61234 GDB GDB コマンド: arm-none-eabi-gdb 参照 変数 Step Mode
< > 21 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(公) 適用(公)
?	デバッグ(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。



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

- ・Renesas Synergy™および S5D9 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。
- ・Synergy 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。
- ・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