Renesas Synergy™ 開発チュートリアル

3.1版 2023年10月02日

1.	概要	2
1	1 概要	2
1.	2 使用環境	2
1.		3
1.	4 各 CPU ボードの資料に関して	4
2.	機能	5
		_
2.	1 サンブルプログラムの使用機能	5
3.	e2 studio を用いた動作方法	6
3.	1 サンプルプログラムの動作方法	6
	3.1.1 インポート方法	6
	3.1.2 ビルド方法1	1
	3.1.3 デバッグ方法1	5
4.	e2 studio を用いた新規プロジェクト作成方法1	9
4	1 JED 占減プログラムの作成例(Blinky) 1	٩
ч. 4		3
ч. 4	2 USB FUNCTION プログラムの作成例 2	8
4	4 補足(端子設定ファイルの選択) 3	4
		'
5.	Renesas Flash Programmer を使用した書き込み3	5
5.	1 USB インタフェースを使用した書き込み方法	5

1. 概要

1.1 概要

本アプリケーションノートは、弊社製 Renesas Synergy™ CPU ボードを用いて、Renesas Synergy™プラットフォーム対応 のプログラムを作成・動作確認するまでのチュートリアル資料です。 各サンプルプログラムの内容や CPU ボードの特性については、各サンプルプログラム解説やハードウェアマニュアル、 回路図をご覧ください。

1.2 使用環境

本アプリケーションノートの解説で用いる開発環境を以下に示します。

各サンプルプログラムで使用されている環境のバージョンが異なる場合に画面の構成など多少違いがございますが、基本的な 使用方法は変わりません。

本アプリケーションノートは、AP-S7G2-0A を基準に記載されておりますが、AP-S5D9-0A, AP-S3A7-0A の場合でも同様に ご参照いただけます。

(AP-S7G2-0A の設定のみ記述されている項目につきましては、ボード名、CPU 名をお使いのボードに合わせてください。)

<e2 studio>

ソフトウェア	バージョン	備考
e2 studio	v2021-07	_
SSP	v2.1.0	Synergy Support Package
GCC ARM Embedded	v7.2.1	_
AP-S7G2-0A 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

<Renesas Flash Programmer>

ソフトウェア	バージョン	備考
Renesas Flash Programmer	V.3.06.00	評価版使用可能

1.3 CPU ボード・スイッチ設定

CPU ボードを動作させる際は、動作モードに応じてボード上のディップスイッチを設定する必要があります。

(1) AP-S7G2-0A / AP-S5D9-0A

出荷時設定:「ボード上の SDRAM を使用する」および「シングルチップモード」



SDRAM ON :ボード上の SDRAM を使用する OFF : ボード上の SDRAM を使用しない



BootMode SCI/USB: SCI ブートモード・USB ブートモード SGL :シングルチップモード (プログラム動作時やデバッガ使用時)

(2) AP-S3A7-0A

出荷時設定:「シングルチップモード」



BOOT SC/USB: SCI ブートモード・USB ブートモード SGL : シングルチップモード (プログラム動作時やデバッガ使用時)

1.4 各 CPU ボードの資料に関して

資料・サンプルプログラムは、弊社 Web サイトのボード紹介ページで公開されています。

Synergy CPUボードシリーズhttps://www.apnet.co.jp/product/synergy/index.htmlAP-S7G2-0A製品ページAP-S5D9-0A製品ページAP-S3A7-0A製品ページhttps://www.apnet.co.jp/product/synergy/ap-s5d9-0a.htmlhttps://www.apnet.co.jp/product/synergy/ap-s3a7-0a.html

2. 機能

2.1 サンプルプログラムの使用機能

以下のサンプルプログラムを用いて、各機能の確認をすることができます。

٢ΔÞ	-57	G2-	$< \Delta \Omega$
	<i>J</i>	<u>ے</u> ں	07-

ドキュメント番号	サンプルプログラム	機能
AN1802	UART サンプルプログラム	UART 通信
AN1803	USB FUNCTION サンプルプログラム	USB FUNCTION
AN1804	USB HOST サンプルプログラム	USB HOST
AN1805	CAN サンプルプログラム	CAN 通信
AN1806	Ethernet サンプルプログラム	Ethernet 通信
AN1807	QSPI サンプルプログラム	QSPI 通信による QSPI Flash へのアクセス
AN1808 SD サンプルプログラム		SD カード
AN1809	GUIX サンプルプログラム	GLCDC、GUIX を用いた LCD 表示
		I2C による LCD-KIT 制御
AN1811 LCD-KIT サンプルプログラム		GLCDC、I2C による LCD-KIT 制御
AN1827	WM-RP-10 サンプルプログラム	SPI 通信による無線モジュール制御
AN1831	TCP/IP サンプルプログラム	TCP/IP ソケット通信

<AP-S5D9-0A>

ドキュメント番号	サンプルプログラム	機能
AN1812	UART サンプルプログラム	UART 通信
AN1813	USB FUNCTION サンプルプログラム	USB FUNCTION
AN1814	USB HOST サンプルプログラム	USB HOST
AN1815	CAN サンプルプログラム	CAN 通信
AN1816	Ethernet サンプルプログラム	Ethernet 通信
AN1817	QSPI サンプルプログラム	QSPI 通信による QSPI Flash へのアクセス
AN1818	SD サンプルプログラム	SD カード
AN1820	LCD-KIT サンプルプログラム	GLCDC、I2C による LCD-KIT 制御
AN1828	WM-RP-10 サンプルプログラム	SPI 通信による無線モジュール制御
AN1830	TCP/IP サンプルプログラム	TCP/IP ソケット通信

<AP-S3A7-0A>

ドキュメント番号	サンプルプログラム	機能
AN1821	UART サンプルプログラム	UART 通信
AN1822	USB FUNCTION サンプルプログラム	USB FUNCTION
AN1823	CAN サンプルプログラム	CAN 通信
AN1824	SD サンプルプログラム	SD カード
AN1829	WM-RP-10 サンプルプログラム	SPI 通信による無線モジュール制御

3. e2 studio を用いた動作方法

本章では、e2 studio を使用して CPU ボードを動作させる方法を説明します。 e2 studio および SSP(Synergy Software Package)につきましては、事前にインストール行ってください。

3.1 サンプルプログラムの動作方法

本節では、弊社で作成したサンプルプログラムの動作方法について説明いたします。

- 3.1.1 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。



 ② [CMSIS Pack] を選択し [次へ] を選択し、サンプルプログラム内「¥sample¥CutomBSP」にある pack ファイル「AlphaProject.ap_s7g2_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。

(3) インポート		×
選択		1
インポート・ウィザードの選択(<u>S</u>):		
フィルタ入力		
		•
CMSIS Pack		
🔗 Rename & Import Existing C/C++ Project into Workspace		
Reneas Webサイト上のサンプル・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		
つオルターまたはアーカイフ由来のフロジェクト デディングレーターをした。 マックト マッ		
○ 既存ノロンエクトをリークスペースペ		
> C Comph		
> Carling		
С ХМІ		Υ
(?) < 戻る(B) 次へ(N) > 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_s7g2_0a.2.1.0-ap010000.pack」を選択してください。

Import Synergy	/ CMSIS Pack			_		×
Import Synergy	Import Synergy CMSIS Pack					
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	.JL

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。

インポート	_		×
選択 Import a Synergy CMSIS Pack into e2 studio		Ľ	5
インポート・ウィザードの選択(<u>S</u>):			
7ብሥፅኢታ			
 ◇ 一般 ◇ CMSIS Pack ◇ Rename & Import Existing C/C++ Project into Workspace ◇ Reneas Webサイト上のサンプル・プロジェクト ◇ Renesas CA78KOR (CS+) プロジェクト ◇ Pーカイブ・ファイル ◇ アーカイブ・ファイル ◇ フィルターまたはアーカイブ由来のブロジェクト ◇ 設定 > ◇ C/C++ ◇ Oomph ◇ Tracing ◇ XMI 			<
(ア) (ア) 終了(E)		キャンセノ	ŀ

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				—		×
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	⁺るティレクトリ	ーを選択します。				
◉ ルート・ディレクトリーの選択(<u>I</u>):					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>):						
				đ	べて選択(<u>S)</u>
				選択	をすべて解	除(<u>D</u>)
					更新(<u>E</u>)	
オプション						
□ ポストしたフロジェクトを使業(<u>日</u> □ 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>)				
□ ワークスペースに既に存在するフ	ロジェクトを隠	र् <u>र</u> (<u>i</u>)				
ワーキンク・セット					_	
」ワーキンク・セットにフロジェクト	を追加(1)			Ť	新規(<u>W</u>)	_
ワーキング・セット(<u>0</u>):				\sim	選択(<u>E</u>)	
?	戻る(<u>B</u>)	次へ(<u>N</u>) >	終了(<u>F</u>)	キャンセ	IL

アプリケーションノート AN1826

⑧ [終了]を選択します。

インポート	— 🗆 X
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。	
 ・ ルート・ディレクトリーの選択(①: C:¥workspace¥Synergy¥ap_s7g2_0a_sample_ ・ アーカイブ・ファイルの選択(<u>A</u>): プロジェクト(<u>P</u>): 	 ✓ 参照(R) ✓ 参照(R)
ap_s7g2_0a_sample_can(C:¥workspace¥Synergy¥ap_s7g2_0a_sample_	すべて選択(<u>S</u>) 選択をすべて解除(<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>U</u>) □ プロジェクトをワークスペースにコピー(<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>) □ 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) ワーキング・セット(Q): ~	新規(<u>₩</u>) 選択(<u>E</u>)
< 戻3(B) 次へ(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。

3.1.2 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。

syn_workspace - ap_s7g2_0a_sample_can/co	onfiguration.xml - 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)	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(<u>H</u>)	
🔦 🎋 🔳 🏘 デバッグ(B)	🗸 💽 ap_s7g2_0a_sample_can Debug 🛛 🗸 🔅 📑 🖛 📄	💿 🚳 = 🔦 = 🗟 🗙 🎄 -	• 9 <u>4</u> •
0 16 🗰 💷 📽 🖏 🖑 🤌	ं 📸 र 🛍 र 🖸 र 🞯 र 🅭 🛷 र 🕅 🗉 🖷 🧏 र	휘 • 🏷 라 🔶 • 🖒 • 🛛 🛃	
		Q 😰 🗟 C/C++ 戀 🗄	Synergy Configuration 🛛 🎄 Debug
☐ プロジェクト・エクスプローラー ※ □ □	🕸 Jap s7g2 0a sample can] Synergy Configuration 🗙		📴 アウトライン 🛛 🗖 🗖
		0	8
✓ 2 ap_s7g2_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
> 🗊 Includes		^	イターはありません。
> 🚰 src			
ap s7q2 0a sample can Debug.jlink	Device Selection		
ap_s7g2_0a_sample_can Debug.launcl	SSP version: 210	Board Details	
ap_s7g2_0a(QSPI_ON)_R7FS7G27H3A(SSF VEISION. 21.0		
ap_s/g2_0a(SDHI_ON)_K/FS/G2/H3A(Board: ap_s7g2_0a 🗸 🔛		
R7FS7G27H3A01CFC.pincfg	Device: R7FS7G27H3A01CFC		
synergy_cfg.txt			
> (?) Developer Assistance		×	
	Cummer DCD Castle Dire Threads Mercering Commerced	>	
	Summary BSP Cocks Pins Threads Messaging Components		
	Problems ヨンソール ※ ※ スマート・ブラウザー ※ スマート・		<u> </u>
	Synergy		A
< >	<		>

③ [BSP]タブで [Board] が動作確認するサンプルプログラムに従って、下記の項目が選択されていることを確認します。

AP-S7G2-0A	:	ap_s7g2_0a
AP-S5D9-0A	:	ap_s5d9_0a
AP-S3A7-0A	:	ap_s3a7_0a

syn_workspace - ap_s7g2_0a_sample_can/o	onfiguration.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	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(<u>H</u>)	
🔦 🎋 🔳 🎋 デバッグ(B)	✓ I ap_s7g2_0a_sample_can Debug ✓ ♦	📄 🛞 🕶 🐔 🕶 🔜 🗞 🎋 :	• Q ₄ •
u 🗞 🕪 💷 😭 🖏 🧔 💋	🖉 📸 🕶 🖻 🕶 🞯 🕶 🎥 🛷 🕶 🗟 🗉 🔮 🖉 🖛] ▼ <> <> <> = 🛃	
		२ 🗄 🖬 c/c++ 🔅 ९	Synergy Configuration 🛛 🐐 Debug
□ □ □	🔅 [ap_s7q2_0a_sample_can] Synergy Configuration 🗙		アウトライン ※ □ □
日 気 7 兆	Reard Support Deckage Configuration	0	8
✓	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
> 🔊 Includes		^	ィターはありません。
> 🗁 src			
ap_s7g2_0a_sample_can Debug.jlink	Device Selection		
ap_s7g2_0a_sample_can Debug.launc	SSP vertion: 210	Board Details	
ap_s7g2_0a(QSPI_ON)_R7FS7G27H3A			
configuration.xml	Board: ap_s7g2_0a		
R7FS7G27H3A01CFC.pincfg	Device: R7FS7G27H3A01CFC		
synergy_cfg.txt			
> (?) Developer Assistance		×	
	Summany BSD Clocks Dire Threads Messaging Components	>	
	Summary DSF Clocks Fins mreads messaging components		
	Problems ■ コンソール ☆ スマート・フラウザー ↓↓ スマート・マ	ニュアル 🖳	11 🕪 🔜 🖻 + 🖸 + 🗖 - 🗆
	synergy		^
< >	<		>

AP-S3A7-0A

 ④ [Pins]のタブを開き、端子設定ファイルの設定をします。
 動作確認するサンプルプログラムに従って、[Select pin configuration]に下記のように端子設定ファイルが 選択されていることを確認してください。
 ・QSPIのサンプルを動作する場合
 AP-S7G2-0A : 「ap_s7g2_0a(QSPI_ON)_R7FS7G27H3A01CFC.pincfg」
 AP-S5D9-0A : 「ap_s5d9_0a(QSPI_ON)_R7FS5D97E3A01CFC.pincfg」
 ※ SDHI用端子設定ファイルではサンプルプログラムが動作しません。
 ・QSPI以外のサンプルを動作する場合
 AP-S7G2-0A : 「ap_s7g2_0a(SDHI_ON)_R7FS7G27H3A01CFC.pincfg」
 AP-S5D9-0A : 「ap_s5d9_0a(SDHI_ON)_R7FS7G27H3A01CFC.pincfg」

[Generate data] にチェックが入っており、「g_bsp_pin_cfg」が入力されていることを確認します。

: [ap_s3a7_0a_R7FS3A77C3A01CFP.pincfg]

syn_workspace - ap_s7g2_0a_sample_can/con	figuration.xml - e² studio	– 🗆 ×
ファイル(<u>F</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>) ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>)	
🔦 🕸 🔳 🕸 テバッグ(B) 🗸	💽 ap_s7g2_0a_sample_can Debug 🛛 🗸 🎄 : 😭 🕶 🔚 🐘 🐘 😵 🗸 🍇 🗼 🔻 💊 🔹	• 0₀ • ₺ ₪ ш 🗑 🖏
🚳 🙆 🥖 📸 ד 🛍 ד 🚱 ד 🧐	🔗 ▼ 🗟 🗉 ୩ ½ ▼ 🖏 ▼ 🏷 ♂ 🗘 マ 🖓 🛃 🕜 ♥ 🖓 🛃	Synergy Configuration 🛛 🎄 Debug
🔁 プロジェクト・エクスプローラー 🛛 🖳 🗖	🔅 [ap_s7g2_0a_sample_can] Synergy Configuration 🗙	
► Sq2_0a_sample_can	Pins Configuration	O Generate Project Content
> ﷺ //(1/1)- >	Select pin configuration [ap_s7g2_0a(SDHLON)_R7FS7G27H3A01CFC.pincfg] >>	Pins Tutorial 🔏 👻 🗟
> 🗁 script	Pin Selection Pin Configuration	
 e synergy_crig ap_s7g2_0a_sample_can DebugJlink ap_s7g2_0a_sample_can DebugJaunch ap_s7g2_0a(SPI_ON)_R7FS7G27H3A011 configuration.xml R7FS7G27H3A01CC_Dincfg synergy_cfg.txt 	フォルタスカ > ✓ Ports > ✓ Peripherals Summary, IBSP Clock Pins Threads Messaging Components	(B)
> (7) Developer Assistance	Problems 回 ソソール 窓 @ スマート・ブラウザー 印 スマート・マニュアル	
	Synergy	
< >	<	>

⑤ [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_s7g2_0a_sample_can/c	onfiguration.xml - e² studio	- C
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(<u>H</u>)	
🔦 🎄 🔳 🏘 デバッグ(B)	∨ 💽 ap_s7g2_0a_sample_can Debug 🛛 ∨ 🌼 🕴 🖶 🔚 🎲 🗸 🐘 🛛 😵 🗸 🐝 🖗 😵 🐇 🌺	r 💁 🕶
🔍 • 核 🗰 💷 😭 🖏 🧔 🎉	🕴 📸 두 📸 두 🚱 두 🎒 🚀 두 🔛 🗐 🖷 🖢 🖉 두 🏷 르카 🖓 두 🖒 두 📑	
	Q、 118 116 c/c++ 徽 S	ynergy Configuration
🔁 วือรัรวระวาน 🙄 🗆	🔅 [ap_s7g2_0a_sample_can] Synergy Configuration 🗙 📃 📃	🧱 アウトライン 🙁
► 🕏 🏹 🖇	Board Support Package Configuration	アウトラインを提供するア!
> 🔊 Includes > 😂 src		イターはありません。
> 🗁 script	Device Selection	
 ap.<i>srg2</i>, oug. sample_can bebug.inkn ap.<i>srg2</i>.0a, sample_can bebug.iaunci ap.<i>srg2</i>.0a(SDH_ON)_R7FS7G27H3A(ap.<i>srg2</i>.0a(SDH_ON)_R7FS7G27H3A(configuration.xml R7FS7G27H3A01CFC.pincfg ynergy_cfq.tt 	SSP version: 21.0 Board Details Board: ap_s7g2_0a im Device: R7FS7G27H3A01CFC im	
> ⑦ Developer Assistance	v	
	< >	
	Summary BSP Clocks Pins Threads Messaging 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 🛛 🖳 🧕] 🚱 🗧 🚍 🖬 😭
	Synergy	

⑥ 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_s7g2_0a_sample_can/ca	onfiguration.xml - e² studio		- 🗆 X
ファイル(<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>) ウィンドウ(<u>W) ヘルプ(H</u>)	
🔦 🕸 🔳 🎋 デバッグ(B)	∨ 💽 ap_s7g2_0a_sample_can Debug 🛛 ∨ 🔅 🗄 ▼ 🔚	🕼 🛞 • 🔦 • 🔒 🔌 🏀	- Q
🕓 🛪 🕪 🎟 😭 🖏 💩 🧔 🖉	' 📸 🕶 🔂 🕶 🕃 🕶 🚱 🛩 📴 🛷 🕶 💷 🔳 📲 🖢 🖷	श र र दे दे र र र वि	
		Q 🗄 💼 C/C++ 🕸 S	ynergy Configuration 🛛 🐐 Debug
🔁 プロジェクト・エクスプロー フ ー 🛛 🕓 🗖	🔅 [ap_s7g2_0a_sample_can] Synergy Configuration 🗙		アウトライン ※ □ □
	Board Support Package Configuration	Ø	- 8
✓	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
> 🛐 Includes		^	ィターはありません。
> 🗁 src			
ap_s7g2_0a_sample_can Debug.jlink	Device Selection		
ap_s7g2_0a_sample_can Debug.launcl	SSP version: 2.1.0	Board Details	
ap_s7g2_0a(QSPI_ON)_R7FS7G27H3A(
i up_srgz_ca(osn_on)_crrsrdzmiska i configuration.xml	Board: ap_s/g2_0a V		
R7FS7G27H3A01CFC.pincfg	Device: R7FS7G27H3A01CFC		
synergy_cfg.txt			
> (7) Developer Assistance		×	
	Summary BSP Clocks Pins Threads Messaging Components	>	
			▝▖▙▖▖▖▖▖▖▖▖▖
	Superny	· vite_v	
	syncigy		^
			~
< >	<		>

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。

3.1.3 デバッグ方法

- ① 「3.1.2 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設定します。
 サンプルプログラム動作時は、BootModeは「SGL」、SDRAM 搭載 CPU ボードは「SDRAM を使用する」に設定します。

<AP-S7G2-0A / AP-S5D9-0A>





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

<AP-S3A7-0A>



JSW1 : 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。

syn_workspace - ap_s7g2_0a_sample_can/co	onfiguration.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)	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>) ウィント	[×] ウ(<u>W)</u> ヘルプ(<u>H</u>)
🔏 🏘 🔳 🏘 デバッグ(B)	🗸 💽 ap_s7g2_0a_sample_can Debug 🛛 🗸 🔅 📑 🕶 🔚 🗞 🛛	Ð ▼ ≪ ▼ 📾 🔍 💥 🗱 🔺 💊 ▼
0 15 🗰 💷 😭 🖏 🏟 🤌	¹ 💼 र 🛍 र 🖻 र 🞯 र 😕 🔗 र 💷 🗉 👖 🖢 र 🖗	(□ -(*
		Q 記 C/C++ デパッグ(D) > ig
陷 プロジェクト・エクスプローラ- 🗙 🖳 🗖	[ap_s7g2_0a_sample_can] Synergy Configuration	ー デバッグの構成(B)
	Board Support Package Configuration	お気に入りの編成(V)… 8
v 🚰 ap_s7g2_0a_sample_can [Debug]		Generate Project Content アウトラインを提供するアクティブなエデ
> 🔊 Includes		▲ 1少−はめりません。
> 🔑 src	Device Selection	
> 🚰 synergy	Device Selection	
> 🔁 Debug	SSP version: 2.1.0 V	Details
> 🦳 synergy_cfg	Board: ap_s7g2_0a V	
ap_s7g2_0a_sample_can Debug.jlink	Project P7ES7G27H2A01CEC	
ap_s/g2_0a_sample_can Debug.launce	Device:	
ap_s7g2_0a(SDHI_ON)_R7FS7G27H3A		
🔅 configuration.xml	<	>
K7FS7G27H3A01CFC.pinctg syneray_cfa.txt	Summary BSP Clocks Pins Threads Messaging Components	
> ⑦ Developer Assistance	🖹 Problems 📃 コンソール 🙁 🦦 スマート・ブラウザー 👊 スマート・マニュアル	
		🗙 🕂 🗘 🔁 🔜 🖓 = 🗟 🗲 😾 🖵 🕇 🖛
	CDT ビルド・コンソール [ap_s7g2_0a_sample_can]	
	on the puild Sinished in second 2 second as (tesh 12)	A (17)
	09:42:56 Build Finished. @ errors, 5 Warnings. (took 15:	s. 655ms)
		¥
< >>	<	>
1 ⁻⁶ 7-2 0		
ap_srg2_ua_sample_can		

⑤ [Renesas GDB Hardware Debug] を選択後、 [新規の起動構成] をクリックします。

💿 デバッグ構成	— 🗆 X
構成の作成、管理、および実行	
	 このダイアログから起動設定を構成します: ご - 選択した種類の構成を作成するには、「新規構成」ボタンを押します。 ご - 選択した種類の起動構成プロトタイブを作成するには、「新規構成」ボタンを押します。 ご - 選択した種類を起うスポートするには、「エクスポート」ボタンを押します。 ご - 選択した構成をロクスポートするには、「エクスポート」ボタンを押します。 ご - 選択した構成をコピーするには、「花袋」ボタンを押します。 ※ - 選択した構成を削除するには、「削除」ボタンを押します。 デ - アイルタ・オブションを構成するには、「フイルタ」ボタンを押します。 ・ 既存構成を選択して、それを編集または表示します。 ・ 取存構成を選択して、それを編集または表示します。 ・ プロトタイブをリンクするには、起動構成を選択して「プロトタイブをリンク」メニュー項目を選択してください。 ・ プロトタイブのリンクを解除するには、起動構成を選択して、「プロトタイブ値でリセット」メニュー項目を選択してください。 ▶ アロトタイブ値でリセットするには、起動構成を選択して「プロトタイプ値でリセット」メニュー項目を選択してください。
21 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

- ⑥ 新規作成されたデバッグ構成を選択し[メイン] タブを開き、各項目を以下のように設定してください。
 (図は「ap_s7g2_0a_sample_can」をデバッグする際の例です。)
 - [名前]:任意の値を入力。
 - [プロジェクト]:「参照」ボタンを押し、プロジェクトを選択。
 - [C/C++アプリケーション]:「プロジェクトの検索」ボタンを押し、ビルドで生成した elf ファイルを選択。

📴 デバッグ構成				
構成の作成、管理、および実	行			Ť.
 ・ ・ ・	名前(<u>U</u>): ap_s7g2_0a_sample_can Debug コメイン 弥 Debugger ▶ Startup □ 3 プロジェクト(<u>P</u>): ap_s7g2_0a_sample_can C(C++ アブルアーション・ Debug/ap_s7g2_0a_sample_can.elf 起動前に必要に応じてビルド <u>Build Configuration</u> : Use Active ○ 自動じルドを有効にする ● ワークスペース設定の使用	共通(C) 頃 ソース 変数(V) ○ 自動ビルドを: ワークスペース設定	プロジェクトの検索(<u>H</u>) 無効にする 定の場成	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		10	カ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

⑦ [Debugger] タブを選択し、 [Debug hardware] に [J-Link ARM] を設定し、 [Target Device] に以下を選択します。
 AP-S7G2-0A の場合: R7FS7G27H

AP-S5D9-0A の場合: **R7FS5D97E** AP-S3A7-0A の場合: **R7FS3A77C**

デバッグ構成	_		Х
構成の作成、管理、および実行 Run Break Time Measurement Operating Frequency must not be empty and must be between 0.001_10000.0MHz.		Ŕ	r
アルタンカ マハクシカ C (7C++ アブリケ-ション) C (7C++ リモート・アブリケ-ション) E ASE Script C GDB OpenOCD Debugging C GDB Simulator Debugging C GDB N=(トウェア・デバッギング) Java アブリケーション) M V (*) Debug B + ##@① Startup The Assess Gript C GDB Simulator Debugging C GDB /L-(?ウェア・デ(?)ギング) Java アブリケーション) M V (*) Debug B + ##B2 OB # ##B2: OB # ##B2: <	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!	こ戻す(<u>V</u>)	適用(<u>Y</u>)	
? 	(ッグ(<u>D</u>)	閉じる	

⑧ [デバッグ]を選択します。

📴 デバッグ構成	- D X
構成の作成、管理、および実行 Run Break Time Measurement Operati	Ting Frequency must not be empty and must be between 0.001_10000.0MHz.
ご 診 診 論 ¥ □ マ ▼ フィルタ入力	名前(M): ap_s7g2_0a_sample_can Debug
 C/C++ アノリリーション C/C++ リモート・アプリケーション EASE Script GDB OpenOCD Debugging 	Debug hardware: J-Link ARM V Target Device: R7FS7G27H GDB Settings Connection Settings デパッグ・ツール設定
ご GDB Simulator Debugging (R C GDB ハードウエア・デバッギング J Java アブリケーション 図 Java アブレット C Renesas GDB Hardware Debu	GDB 接続設定: ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
 C) ap_s/g_ua_sample_can u C) Renesas Simulator Debuggin Uモート Java アブリケーション 最勤グループ 	GDB GDB J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(火) 適用(火)
?	デバッグ(<u>D)</u> 閉じる

⑨ 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。



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。

各プログラムの動作については、デバッグしているサンプルプログラムのアプリケーションノートをご参照ください。

4. e2 studio を用いた新規プロジェクト作成方法

本章では、e2 studio で CPU ボード用プログラムを作成する方法を説明します。 「4.1 LED 点滅プログラムの作成例(Blinky)」では、LED の点滅を行う Blinky のプロジェクトを作成します。 また、「4.2 UART プログラムの作成例」および「4.3 USB FUNCTION プログラムの作成例」では、Synergy プロジェク トコンフィギュレータを使用したプロジェクトの作成例として、弊社製サンプルプログラムの作成方法を説明します。 また、本章では、図中の選択・実行する必要のある箇所を赤色枠_____、結果の確認をする箇所を橙色枠_____で 示します。

4.1 LED 点滅プログラムの作成例 (Blinky)

本節では、AP-S7G2-0A の Blinky プロジェクト(LED の点滅を行うプログラム)の作成方法を説明します。 AP-S5D9-0A や AP-S3A7-0A をご使用の場合は、適宜読み替えて行ってください。

① e2 studio のメニューから、 [ファイル] – [新規] – [Renesas C/C++ Project] - [Renesas Synergy] を選択します。

	syn_workspace - e² studio									
ファイ	ル(F) 編集(E) ソース(S)	リファクタリング(T) ナビ	ゲート(N)	検索(A	A) プロジェクト(P)	Renesas Views	実行(R)	ウィンドウ(W)	ヘルプ(H)	
	新規(N)	Alt+S	/フト+N >	F	Renesas C/C++ Pro	oject		>	Renesas Debug	。 ▼ 槗 ⊪
۵,	ファイルを開く(.) ファイル・システムからプロジェ! 最近のファイル	7トを開く	>		Synergy C/C++ Pro Makefile Project w C/C++ Project	oject ith Existing Code			Renesas Synergy	Synergy Con
	閉じる(C) すべて閉じる(L)	Ctrl+シ	Ctrl+W フト+W	□ 4	フロジェクト(R) C/C++ プロジェクトに	変換 (C/C++ ネー	チャーを追加])	-	
	保存(S) 別名保存(A) すべて保管(E) 前回保管した状態に戻す(T)	Ctrl+:	Ctrl+S シフト+S		ソース・フォルダ フォルダー ソース・ファイル ヘッダー・ファイル					
62	移動(V) 名前を変更(M) 更 新(F)		F2 F5	© 2	ゲンプレートからファイ. クラス サンプル(X)	ν 			-	
	行区切り文字の変換(D)		>	📑 ÷	その他(o)			Ctrl+N		
۵	印刷(P)		Ctrl+P							
2	インポート(I) エクスポート(O)				א א-עעב	マート・ブラウザー	10. zv-h	・マニュアル		
	プロパティ(R)	Alt	+Enter		••••					
	ワークスペースの切り替え(W) 再開		>							

② [Renesas Synergy C Executable Project] を選択し、 [次へ] を押します。

New C/C+	Project – 🗆 🗙	(
Templates fo	Renesas Synergy Project	
All C/C++	Renesas Synergy C Executable Project RENESAS A C Executable Project for Renesas Synergy.	^
	Renesas Synergy C Library Project RENESAS A C Library Project for Renesas Synergy.	
	Renesas Synergy C Project Using Synergy Library RENESAS Creates a Capplication project which uses an existing Synergy library project	
	Renesas Svnerov C++ Executable Proiect	*
?	< 戻る(<u>B</u>) 次へ(<u>N</u>) > 終了(E) キャンセル	

③ [Project name] に任意のプロジェクト名を入力します(ここでは「blinky_test」とします)。
 入力後、 [次へ] を押して次へ進みます。

uration (Synergy C Executable Project) — 🗆 🗙
uration (Synergy C Executable Project)
Toolchains
GNU ARM Embedded
使用(<u>D</u>)
ace¥WMRP_20211014¥syn_workspace¥blir 参照(<u>R</u>)
Fムを選択(Y): デフォルト ~
Change license file
$\label{eq:com.renesas.platform_57210792} \end{tabular} internal \end{tabular} project \end{tabular} general \end{tabular} are also \end{tabular} and \end{tabular} are also tabu$
DN: ronics America Inc. rgy Evaluation User com
icense file and Pack file downloads < 戻る(B) 次へ(N) > 終了(E) キャンセル
Change license fil ¥com.renesas.platform_57210792¥internal¥projectgen¥arm¥licenses¥SSP_License_Example_EvalL DN: onics America Inc. rgy Evaluation User com icense file and Pack file downloads <

 ④ [SSP version] が [2.1.0] であることを確認し、 [Board] から [ap_s7g2_0a] を選択します。
 (選択肢に [ap_s7g2_0a] が見つからない場合、「3.2.1 インポート方法」を参考にサンプルプログラム内の Custom BSP をインポートしてください。)

Toolchain の設定では、 [Toolchain version] に使用する Toolchain バージョン(図中では [7.2.1.20170904])を、 [Debugger] には [J-Link ARM] をそれぞれ設定してください。

設定後、 [次へ] を押します。

evice Selection	support that you require.		
SSP version: Board: Device:	2.1.0 v ap_s7g2_0a v R7FS7G27H3A01CFC	Board Details	
elect Tools oolchain: oolchain vers ebugger:	GNU ARM Embedded ion: 7.2.1.20170904 J-Link ARM	v	Available Tools GNU ARM Embedded 9.2.1.20191025 8.3.1.20190703 7.2.1.20170904 6.3.1.20170904 6.3.1.20170620 5.4.1.20160919 4.9.3.20150529 V Debuggers J-Link ARM RTOS Express Logic ThreadX Smart Manual IO Registers Supported Software Manual Supported

⑤ テンプレートとして [Blinky with ThreadX] を選択し、 [終了] を押して設定を終了します。

 e2 studio - Project Configuration (Synergy C Executable Project) e2 studio - Project Configuration (Synergy C Executable Project) Select the type of project you wish to create. 	_		×
Project Template Selection BSP Base Board Support Package for the chosen Synergy family. [Renesas.Synergy.2.1.0.pack] Blinky Blinky Blinky			
(Renesas:Synergy.2.1.0.pack) Silinky with ThreadX Threaded version of Blinky project. [Renesas:Synergy.2.1.0.pack]			
Code Generation Settings ✓ Use Synergy Code Formatter ⑦		キャンセノ	ŀ

⑥ プロジェクトが作成されました。
 AP-S7G2-0A 上の LED が点滅するプログラムが生成されていますので、ビルドを行います。
 以上で Blinky プロジェクトの作成は完了です。

「3.1.3 デバッグ方法」を参考に、ボード上の LED が点滅することを確認してください。



4.2 UART プログラムの作成例

本節では、AP-S7G2-0A で動作する UART のプログラムを例に、プロジェクトの作成方法を説明します。 AP-S5D9-0A や AP-S3A7-0A をご使用の場合は、適宜読み替えて行ってください。 なお、プログラムのアプリケーション部のソースコードは、サンプルプログラムのソースコードを利用します。

① プロジェクトを作成します。

「4.1 LED 点滅プログラムの作成例(Blinky)」①~④を参考に、Custom BSP を用いて弊社ボード用のプロジェクトを 作成してください(プロジェクト名は「sample_uart」として説明します)。

プロジェクトのテンプレートは [BSP] を選択し、 [終了] を押してプロジェクトを作成します。

e2 studio - Project Configuration (Synergy C Executable Project)							
e2 studio - Project Configuration (Synergy C Executable Project) Select the type of project you wish to create.							
Project Template Selection							
BSP Base Board Support Package for the chosen Synergy family. [Renesas.Synergy.2.1.0.pack]							
Blinky Blinky project. [Renesas.Synergy.2.1.0.pack]							
Blinky with ThreadX Threaded version of Blinky project. [Renesas.Synergy.2.1.0.pack]							
Code Generation Settings Use Synergy Code Formatter							
(?) <戻3(B)	(E)	キャンセル					

② プロジェクトが作成されました。

以降では、Synergy プロジェクトコンフィギュレータでプロジェクトの設定を行います。



③ [Threads] タブを開きます。この画面で、HAL ドライバやフレームワークの追加や設定をします。
 [New Stack] を押して、 [Driver] - [Connectivity] - [UART Driver on r_sci_uart] を選択します。

🔅 Isample uant Synergy Configuration 🗙		
Threads Configuration	Generate Project Cont	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。
Threads New Thread Remove HAL/Common	Estend Stack > 12 Remov	Analog > Connectivity > Crypto > Graphics > Hull C Master Driver on r_cin Graphics > Hull C Master Driver on r_sic_i2c Input > Hull C Master Driver on r_sic_i2c Storage > Storage > Storage > Storage > Storage > Storage > Hull Driver on r_sci_uart Transfer > Hull Driver on r_sci_uart

④ UART のモジュールが追加され、下図の橙色の枠のように、UART Driver が画面に表示されます。
 [g_uart0 UART Driver on r_sci_uart]のモジュールをクリックすると、モジュールのプロパティがプロパティビューに表示されます。

🔅 *[sample_uart] Synergy Configuration 🎗	<u>د</u>		- 8	
Threads Configuration			Generate Project Content	
Threads 🕢 New Thread	HAL/Common Stacks	🔄 New Stack > 🐣	Extend Stack > 👔 Remove	
✓ ✓ ✓ HAL/Common	}_fmi FMI Driver on _fmi	g_uart0 UART Driver on r_sci_uart		
g_ioport I/O Port Driver or g_fmi FMI Driver on r_fmi		⊕ g_transfer0 Transfer Driver on r dtr SCI0	Add DTC Driver for	
HAL/Common New Object >		TXI	recommended]	
	<		>	
Summary BSP Clocks Pins Threads Mess	saging Components			
🍸 🖇 🗖 🔂 🎦 Pin Co	onflicts 📃 コンソール 😒		- 6	

プロパティが見つからない場合は、ツールバーの [ウィンドウ] - [ビューの表示] - [プロパティー] で表示できます。

syn_workspace - sample_uart/configuration.xml - e ² studio			- 🗆 X
ファイル(F) 編集(E) ナピゲート(N) 検索(A) プロジェクト(P) Renesas V	/iews 実行(R)	ウィンドウ(W) ヘルプ(H)	
🔦 🐞 🔳 漆 デバッグ(B) 🗸 🖻 sample_uart De	ebug	新規ウィンドウ(N)	👋 ▼ 🗞 ▼ 📾 🔌 🎋 ▼ 💁 ▼ ! 0₀ ▼ 🎋 🕪 💷 😭 🖏
@ ∥ タ・ カ ・ ?		⊥テ19- 外観	> Q i III III C/C++ ∰ Synergy Configuration ☆ Debug
	🌣 *[sample_ua	ビューの表示(V)	→ Q スマート・ブラウザー
g_uart0 UART Driver on r_sci_uart	Threads Co	パースペクティブ(R)	プロジェクト・エクスプローラー enerate Project Content
Settings プロパティ 値		ナビゲーション(G)	プロパティー
API Info V Common	Threads 💀 N	設定(0)	その他(o) Alt+シフト+Q,Q 🖞 Stack > 👘 Remove
External RTS Disable	Re.	EX.AC.(F)	
Reception Enable			

⑤ [g_uart0 UART Driver on r_sci_uart]のモジュールをクリックすると、そのプロパティがプロパティビューに 表示されます。

表示されたプロパティのうち、以下の太字・赤文字の箇所を変更してください。

		P			
syn_workspa	ace - sample_uart/configuration.xml - e*	studio			
イル(E) 編集	i(E) ナビゲート(<u>N</u>) 検索(<u>A</u>) ブロジェクト	(P) Renesas <u>V</u> iews 実行(<u>F</u>	3) ウィンドウ(W) ヘルブ(<u>H</u>)		
*	📕 🐐 デバッグ(B) 🗸 💽	sample_uart Debug		- % - ₿ % ‡ - % - 05 - 5 m 0	16800
1 19 -	• 월 • 전 • ♡ ♡ ◇ • ○ •			🔍 🗄 🔡 🗟 C/C++ 🔯 Synergy Confi	guration 🞋 Deb
🔳 วือパร	F1- X	🛃 🕴 🗖 🖬	德 *[sample_uart] Synergy Configurat	tion 🖂	-
g_uart0	UART Driver on r_sci_uart		Threads Configuration	Gen	erate Project Contr
Settings	プロパティ	值 ^			,
ADUING	✓ Common		New Thread	New Stack > Area Extend	Stack >
APTIMO	External RTS Operation	Disable	Threads Remove	HAL/Common Stacks	
	Reception	Enable		a nemore	
	Transmission	Enable	V 🔐 HAL/Common 🔥		
	Parameter Checking	Default (BSP)	g_cgc CGC Driver on	er on g_uarto OAKT Driver on r_sci_uart	
	Module g_uart0 UART Driver on r_	si	g_elc ELC Driver on r		
	Name	g_uart0	g_ioport I/O Port Dri	(i)	
	Channel	0	🗳 g_fmi FMI Driver on 🧹	A	
	Baud Rate	9600	< >	A a transfer Transfer	C Driver fee
	Data Bits	8bits		Driver on r. dtc SCI0 Recent	Reception [Not
	Parity	None	HAL/Commo 🐔 New Object >	TXI recom	mended]
	Stop Bits	1bit	n Objects 🔬 Remove	(1)	
	CTS/RTS Selection	RTS (CTS is disabled)			
	Name of UART callback function	n user_uart_callback		<	>
	Name of UART callback function	n NULL Y			
	<	>	Summary BSP Clocks Pins Threads	Messaging Components	
Probler	ms 🖾 🛞 スマート・ブラウザー		🍸 🖇 🗖 🗖 🎇 Pin Conflicts 🛽	או-עעב	-
0 errors. 1 v	warning. 0 others				🔲 🚽 🖂 🖉 🖻
記述/説明	^		<終了> blinky test [Debug [Renesas GDB Hardware Debugging] Renesas GD	DB server (Host) (Te
> 💧 똘생	告 (1 項目)		GDBServ Iミュレー5	ver エンディアン : リトル ターからの電源供給 : Off	
<			h_H_Ltaszpana		
nmon			2	R :	

プロパティ	値	備考
▼Common		
External RTS Operation	Disable	
Reception	Enable	
Transmission	Enable	
Parameter Checking	Default (BSP)	
▼Module g_uart0_UART Driver on		以下の設定をすると名前が変わります
Name	g_uart4	default : g_uart0
Channel	4	default : 0
Baud Rate	9600	
Data Bits	8bits	
Parity	None	
Stop Bits	1bit	
CTS/RTS Selection	RTS (CTS is disabled)	
Name of UART callback function to be	user_uart4_callback	default : user_uart_callback
defined by user		
Name of UART callback function for the	NULL	
RTS external pin control to be defined		
Clock Source	Internal Clock	
Baudrate Clock Output from SCK pin	Disable	
Start bit detection	Falling Edge	
Noise Cancel	Disable	
Bit Rate Modulation Enable	Enable	
Receive FIFO Trigger Level	Мах	
Receive Interrupt Priority	Priority 4	default : Priority 12
Transmit Interrupt Priority	Priority 4	default : Priority 12
Transmit End Interrupt Priority	Priority 4	default : Priority 12
Error Interrupt Priority	Priority 4	default : Disabled

プロパティ	値	備考
Baudrate Percent Error	2.0	
UART Communication Mode	RS232	
UART RS485 Communication Mode	Half Duplex	
RS485 DE Port	09	
RS485 DE Pin	14	

⑤ 今回は特に設定する必要はありませんが、続けて他のモジュールを設定することもあります。
 適宜必要な設定を全て行った後に、[Generate Project Content] をクリックしてください。
 自動作成ファイルが出力されて設定がプロジェクトに適用されます。



 ⑦ プロジェクトの「src」フォルダの「~.c」「~.h」ファイルを、UART サンプルプログラムの「src」フォルダ内に コピーします。



⑧ 以上で UART のプロジェクトが完成しました。ビルド後、動作を確認してください。 なお、このプログラムの動作に関しては、UART サンプルプログラム解説のアプリケーションノートを参照してください。



4.3 USB FUNCTION プログラムの作成例

本節では、AP-S7G2-0A で動作する USB FUNCTION のプログラムを例に、プロジェクトの作成方法を説明します。 AP-S5D9-0A や AP-S3A7-0A をご使用の場合は、適宜読み替えて行ってください。 なお、プログラムのアプリケーション部のソースコードは、サンプルプログラムのソースコードを利用します。

- 「4.2 UART プログラムの作成例」①を参考に新規プロジェクトを作成してください。
 (プロジェクト名は「sample_usbf」として説明します。)
- Synergy プロジェクトコンフィギュレータの設定を行います。
 [Threads] タブを開き、 [New Thread] をクリックして新しいスレッドを作成します。

🔅 [sample_usbf] Synergy Configuration	on 🗙		
Threads Configuration		Generate Pr	Distance
Threads New Thread	HAL/Common Stacks	w Stack > 🔮 Extend Stack : move	>
 ✓ MAL/Common ♥ g_elc ELC Driver on r_elc ♥ g_fmi FMI Driver on r_fr ♥ g_ioport I/O Port Driver ♥ g_cgc CGC Driver on r_c < > HAL/Commo New Object > New Object > New Object > 	g_elc ELC Driver on r_elc	 g_fmi FMI Driver on r_fmi i 	g_iopo Driver
	<		>
Summary BSP Clocks Pins Threads	Messaging Components		

③ 新しいスレッドが追加され、Threads ウィンドウ内に [New Thread] が表示されます。



③ [New Thread] に HAL ドライバやフレームワークを追加していきます。
 [New Thread] を選択した上で [New Stack] を押して、 [X-ware] - [USBX] - [Device] - [Classes] - [CDC-ACM] - [USBX Device Class CDC-ACM] を選択します。

🔅 *[sample_usbf] Synergy Configura	ition 🗙 🗧	- 🗆 📴 Półsł	2 23			
Threads Configuration	Generate Project Cor	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。		
Threads New Thread Image: HAL/Common Image: BLC Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn Image: grain FMI Driver on r_fn	New Thread State Image: State St	Nove w FileX > GUIX > NetX > NetX > NetX > US8X >	Common Device Host	Classes Common Synergy Port	> CDC-ACM > HID > Mass Storage	

⑤ 「New Thread」に USBX のモジュールが追加され、下図の橙色の枠のように USBX Device Class CDC-ACM が画面に表示されます。

Sample_usoff Synergy Configuration X		- <i>e</i>
Threads Configuration		Generate Project Content
Threads New Thread Remove # HAL/Common # g_cle ELC Driver on r_elc # g_inpit I/O Port Driver on r_ioport # g_cle CCC Driver on r_cgc Wew Thread # g_ux_device_class_cdc_acm0 USBX Device Class Mew Thread # g_ux_device_class_cdc_acm0 USBX Device Class New Thread Objects New Object > Remove 	New Thread Stacks	New Stack > ▲ Extend Stack > ▲ Remove
Summary BSP Clocks Pins Threads Messaging Component	ts	

⑥ [USBX Device Configuration] のプロパティを開き、以下の赤文字の項目を設定してください。

🔲 プロパティ	(- 🛛	📑 🕴 🗖 🖬	🌣 "[sample_usbf] Synergy Configuration 🗙	- 0
USBX De	vice Configuration		Threads Configuration	ct Content
Settings	[†] J□/(5.1 [×] Common ^{Composite Device [×] Module USX Device Configuration ^{Vendor ID} ^{Product ID} ^{Device Release Number ^{Index of Product String Descriptor ^{Index of Product String Descriptor ^{Index of Product String Descriptor describing this co ^{String Descriptor describing this co ^{String Descriptor describing this co ^{String Code} ^{Index of String Descriptor describing this co ^{String Code} ^{Index of String Descriptor describing this co ^{String Code} ^{Index of String Descriptor describing this co ^{String Code} ^{Index of String Descriptor describing this co ^{String Code} ^{Index of String Descriptor describing this co ^{String Code} ^{Index of String Descriptor describing this co ^{String Code} ^{Index of String Descriptor describing the solution} ^{String Descriptor descriptor in bytes for this confil ^{String Code} ^{Index of USP String Community} ^{Index Of Descriptor descriptor descriptor} ^{Index of USP String Community} ^{Index Of USP Stri}}}}}}}}}}}}}}}	信 ^ Disable ^ 0x0D91 2028 0 0x000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Imetadus Comignation Generate Project Imetadus Comignation New Thread @ Remove Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation Imetadus Comignation	Remove
	Total index number of USB String Descriptor Name of USBX1 annuage Framework	0 NIIII	Summary BSP Clocks Pins Threads Messaging Components	>

プロパティ	値	備考
▼Common		
Composite Device	Disable	
▼Module USBX Device Configuration		
Vendor ID	0x0D91	default : 0x045B
Product ID	0x2028	default : 0x0000
Device Release Number	0x0000	
Index of Manufacturer String Descriptor	0x00	
Index of Product String Descriptor	0x00	
Index of Serial Number String Descriptor	0x00	
Class Code	Miscellaneous	default : Communications(CDC)
Index of String Descriptor describing	0x00	
Size of USB Descriptor in bytes for this	0x00	
Number of Interfaces (Modify this	0x00	
Self-Powered	Enable	
Remote Wakeup	Disable	
Maximum Power Consumption (in	50	
Supported Language Code	0x0409	
Name of USBX String Framework	NULL	
Total index number of USB String	0	
Name of USBX Language Framework	NULL	
Number of Languages to support (US	0	
Name of generated initialization function	ux_device_init0	
Auto Initialization	Enable	

⑦ [Add USBX Port DCD] を選択し、[New] - [USBX Port DSD on sf_el_ux for USBFS] を追加します。

🔅 *[sample_usbf] Synergy Configuration 🗙			
Threads Configuration		Gener	Attended of the Project Content
Threads 🕢 New Thread 👔 Remove 📄	New Thread Stacks	🔄 New Stack > 🛛 🐣 Extend Stac	tk > 🔊 Remove
✓ ALL/Common ♥ g_elc ELC Driver on r_elc ♥ g_fmi FMI Driver on r_fmi ♥ g_ioport I/O Port Driver on r_ioport	<pre> g_ux_device_class_cdc_</pre>	.acm0 USBX Device Class CDC-ACM	^
 ♥ g_cgc coc biter on f_cgc ♥ @ New Thread ♥ g_ux_device_class_cdc_acm0 USBX Devic 	Add USBX Device Class CDC-ACM Source [Optional]	Image: State of the state o	g_usb_ir _cdcacm Interface () Configu
New Object > New Object > Remove		Add USBX Port DCD	
	<	New USBX Port DCD on sf_el_ux	t for USBFS
Summary BSP Clocks Pins Threads Messaging Con	nponents		

⑧ 追加された [USBX Port DSD on sf_el_ux for USBFS] のプロパティを設定します。表示されたプロパティのうち、 以下の赤文字の項目を変更してください。

🔲 プロ	∜ ≂ ≺- ×	2 8 🗆 🗖	🌼 *[sample_usbf] Synergy Configuration 🔀	- 8
g_sf_e	I_ux_dcd_fs_0 USBX Port DCD on sf_el_ux f	or USBFS	Threads Configuration	Generate Project Content
Setting	s JD/(7-1 ✓ Common Full Speed Interrupt Priority LDD Regulator(Only for S3 and S1 part MC ✓ Module g_sf_el_ux_dcd_fs_0 USBX Port DCD o Name USB Controller Selection	值 Priority 3 U Disable n g_sf_el_ux_dcd_s USBFS	Image: Second	New Thread Stacks New Stack Stack New Stack Stack New Stack
💦 Pro	olems 🖄 🦚 スマート・ブラウザー		🍸 🕴 🗖 🛛 🔀 Pin Conflicts 📃	א-רעב 🔳 🗶 🍇 🗟 🚮 🛃 🛃 🛃 🛃 🛨 🖬 🕇

プロパティ	値	備考
▼Common		
Full Speed Interrupt Priority	Priority 3	default : Disabled
LDO Regulator(Only for S3 and S1	Disabled	
▼Module g_sf_el_ux_dsd_fs_0 USBX Port		
Name	g_sf_el_ux_dcd_fs_0	
USB Controller Selection	USBFS	

設定する必要のあるモジュールは以上です。その他のモジュールはデフォルト設定で使用します。

⑨ [New Thread] にオブジェクトを追加します。

[New Thread] をクリックした後、 [New Thread Objects] の [New Object] - [Event Flags] を選択してください。

🗱 *[sample_usbf] Synergy Configu	ration X	- 8
Threads Configuration	Generate Pr	O oject Content
Threads	New Thread Stacks New Stack > Area Extend Stack >	Remove
✓ ✓ ✓ HAL/Common	ss_cdc_acm0 USBX Device Class CDC-ACM	49 USB)
Objects Remove	👄 Event Flags	(i)
	Mutex Queue Semaphore	► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ► ►
Summary BSP Clocks Pins Thread	ds Messaging Components	

 ・追加した [Event Flag] のプロパティを設定します。表示されたプロパティのうち、以下の赤文字の項目を変更してください。

	8 👘 🔲 🌐 *[sample_usbf] Synergy Configuration 🔅	
g_new_event_flags0 Event Flags	Threads Configuration	Generate Project Content
Settings プロパティ 値 Name CDCACM Active Eve Symbol g_cdcacm_activate_i	Flags New Thread New Thread ent_flags0	I Stacks New Stack > Extend Stack > Remove 0 USBX Device Class CDC-ACM USBX Device Configuration USBX Device Configuration g. sf. el_ux_dcd_fs_0 USBX Port DCD on sf_el_ux for USBFS Add Transfer Madula PC Add Transfer Madula

プロパティ	値	備考
Name	CDCACM Active Event Flags	default : New Event Flags
Synbol	g_cdcacm_activate_event_flags0	default : g_new_event_flags0

1 [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 worken	ace - sample ushf/configurat	tion xml - e ² studio		- n x
、この Jyn_worksp ファイル(E) 編集	ace sample_asbi/comigata	ゴロジェクト(P) Paparas Views 実行(P)	0.2.150000 All.3700	
		A second web Debug		tons table to all
1	※ デバック(B)	Sample_usbf Debug	<u>· · · · · · · · · · · · · · · · · · · </u>	S
🛷 🔻 🖢 👻	श • ७ ०१ ५ • ० •		🔍 🛛 📅 🖓 Synergy Config	juration 🐐 Debug
🐺 🔳 プロパラ	Fr- ×	📑 8 🗖 🗖	∰ *[sample_usbf] Synergy Configuration ⊠	
🚡 g_new_e	event_flags0 Event Flag	js	Threads Configuration Gene	rate Project Content
Settings	プロパティ	值		
	Name	CDCACM Active Event Flags	New Thread Stacks 🚱 New Stack > 🐣 Extend Sta	ck > 👔 Remove
	Symbol	g_cdcacm_activate_event_flags0	Remove E	~
			ss_cdc_acm0 USBX Device Class CDC-ACM	
			I g_ioport I/O Port Driver on r	
			New Thread	
			⊕ g_ux_device_class_cuc_acm []	
				for 🕀 USB)
			New Thread See New Object > USBFS	
			Objects Remove	í
			g_cdcacm_activate_event_flags0 E	
			Add Teansfor Modula	un 🔫 🗛 🗸
	<	>	Summary BSP Clocks Pins Threads Messaging Components	
🖳 Proble	ms 🖾 🁒 スマート・ブラウザー		🝸 🕴 🗖 🔀 Pin Conflicts 📮 א א א א א א א א א א א א א א א א א א	3 • 📸 • 🗆 🗖
0 errors, 1 v	warning, 0 others	<u>^</u>	<終了> sample_uart Debug [Renesas GDB Hardware Debugging] Renesas GDB serve	er (Host) (Terminated 2
記述/説明			リソーコ GDBServer エンティアン : リトル Tミュレーターからの電源供給 : Off	^
> 💧 😤	동(1項目)		ターゲット接続開始	
			ターゲット接続終了 GDB・56632	
			ターゲット接続状況 - OK	~
<			> <	>
			8	

 ② プロジェクトの「src」フォルダに、USB FUNCTION サンプルプログラムの「src」フォルダ内のファイルを コピーします。



 ③ 以上で USB FUNCTION プロジェクトが完成しました。ビルド後、動作を確認してください。
 なお、このプログラムの動作に関しては、USB FUNCTION サンプルプログラム解説のアプリケーションノートを 参照してください。

syn_workspace - sample_usbf/configuration.xml - e ² studio	- 🗆 X
ファイル(E) 編集(E) ナビゲート(N) 検索(A) プロジェクト(P) Renesas <u>V</u> iews 実行(R)	ウィンドウ(W) ヘルプ(<u>H</u>)
🔦 🔹 📕 🎄 デパッグ(B) 🗸 🕞 sample_usbf Debug	✓ ♣ [☐ ▼ 圖 ⓑ] ※ ▼ ≪ ▼ 월 [※] 巻 ▼ ▲ ▼ ! % ▼ 巻 빠 Ⅲ ※ ② ② [※] ∅] ∅]
🔗 🕶 🌆 🔻 🏷 🕫 👉 🔹 🔿 💌 📷	🔍 📴 🔤 C/C++ 👹 Synergy Configuration 🎄 Debug
📻 🔲 70/(71- ×	🌼 *[sample_usbf] Synergy Configuration 🛛 🗖 🗖
<pre>g_new_event_flags0 Event Flags</pre>	Threads Configuration Generate Project Content
Settings プロパティ 値	
Name CDCACM Active Event Flags	🔄 New Thread Stacks 🔄 New Stack > 🐣 Extend Stack > 👘 Remove

4.4 補足(端子設定ファイルの選択)

AP-S7G2-0A と AP-S5D9-0A では、QSPI 用と SDHI 用の 2 つのピンコンフィグファイルを用意しています。 端子設定ファイルの設定は、 [Pins] タブで行います。 QSPI Flash を使用する場合は QSPI 用を、使用しない場合は SDHI 用を選択してください。 (Custom BSP を使用したプロジェクトは、デフォルトでは SDHI 用が設定されています。)

端子設定ファイルの変更方法を、SDHI 用端子設定ファイルから QSPI 用端子設定ファイルに変更する例を元に説明します。

			0
Pins Configuration			Generate Project Content
Select pin configuration			Pins Tut
ap_s7g2_0a(SDHI_ON)_R7FS7G	27H3A01CFC.pincfg \v Generate data:	g_bsp_pin_cfg	
Pin Selection	Pin Configuration		
フィルタ入力]		
> ✓ Ports > ✓ Peripherals			

① [Generate data] の入力欄をクリアし、チェックを外します。

🔅 *[sample_usbf] Synergy Configuration 🗙	- 8
Pins Configuration	Generate Project Content
Select pin configuration ap_s7g2_0a(SDHI_ON)_R7FS7G27H3A01CFC.pincfg	Cenerate data: g_bsp_pin_cfg

② QSPI 用の端子設定ファイルを選択し、[Generate data] にチェックを入れて入力欄に「g_bsp_pin_cfg」を入力します。

🔅 *[sample_usbf] Synergy Configuration 🗙	
Pins Configuration	Generate Project Content
Select pin configuration ap_s7g2_0a(QSPI_ON)_R7FS7G27H3A01CFC.pincfg Generate data:	Pins Tuto

③ 設定は完了です。 [Generate Project Content] をクリックして、設定をプロジェクトに反映させてください。

🔅 *[sample_usbf] Synergy Configuration 🗙			- 0
Pins Configuration		Generate P	Project Content
Select pin configuration ap_s7g2_0a(QSPI_ON)_R7FS7G27H3A01CFC.pincfg	Generate data: g_bsp_pin_cfg		<u>Pins Tuto</u>

5. Renesas Flash Programmer を使用した書き込み

本章では、CPU ボードに Renesas Flash Programmer を使用してプログラムを書き込む方法を説明します。

- 5.1 USB インタフェースを使用した書き込み方法
 - ① CPU ボードを USB ブートモードのボード設定に変更し、CPU ボードと PC を USB ケーブルで接続します。
 - (1) AP-S7G2-0A / AP-S5D9-0A



AP-S7G2-0A / AP-S5D9-0A の JSW1の設定は 不問 です。

- (2) AP-S3A7-0A



JSW1:SCI/USB

- ② CPU ボードの電源を入れます。
- ③ Renesas Flash Programmer (以下、RFP と記述します)を起動します。
- ④ RFP を起動すると、以下のようなウィンドウが表示されますので、「ファイル(F)」メニューの「新しいプロジェクトを 作成(N)…」を選択します。

	ファイル(F) デバイス情報(D) ヘルプ(H)	
	新しいプロジェクトを作成(N) ユニークコート	8
	プロジェクトを開く(O) プロジェクトを保存(S)	
	イメージファイルを保存(I) ファイルチェックサム(C)	エンディアン(<u>E</u>): リトル 〜
	ファイルパスワード設定(P)	
	1 test.rpj 2 erase.rpj 3 NBK_EPU-210_FW_v0107.rpj 4 62t sample.rpj	参照 (B)
	終了(X) スタート(<u>S</u>)	
6	Renesas Flash Programmer V3.06.01 [1 Oct 2019] (無(餌	ال

⑤ 新しいプロジェクトの作成ウィンドウが表示されますので、プロジェクトの設定を行います。
 設定後、[接続(O)]を押して接続を開始してください。

<プロジェクト情報>

・ マイクロコントローラ(M)	:	[Renesas Synergy] を選択します。
・ プロジェクト名(N)	:	任意のプロジェクト名を設定します。
		(ここでは例として、「ap_s7g2_rfp」を入力しています。)
・ 作成場所(F)	:	[参照(B)]ボタンを押し、任意の保存先を設定します。
		(ここでは例として「C:¥workspace¥synergy」を選択しています。)
<通信>		
・ ツール詳細(D)	:	[ツール詳細(D)] ボタンを押して設定ウィンドウを開きます。
		COM の一覧から [Synergy USB Boot] を選択し、 [OK] を押します。
		(COM ポートの番号は環境により異なります。)

USB Direct が動作しない場合、USB ブート用ドライバが誤認識されている可能性があります。 詳細は RFP のユーザーズマニュアルをご覧ください。

_				
	プロジェクト情報			
	マイクロコントロ <i>ーラ(<u>M</u>):</i>	Renesas Synergy 🗸 🗸		
	プロジェクト名(<u>N</u>):	ap_s7g2_rfp		
	作成場所(<u>F</u>):	C:¥workspace¥synergy		参照(<u>B</u>)
	·●			
	ッール(T): COM port	 インタフェース(1): 2 wire U 	IART 🗸	
	<u>)</u> //a∓inm(<u>D</u>)			
			<u> 接続(O</u>)	キャンセル(0)
-				
	ツール選択			
	COM15 : Synergy	USB Boot		
		<u></u> K キャン	パロル(<u>C</u>)	

⑥ RFP のメイン画面が表示されます。

プログラムファイルの[参照]ボタンをクリックし、書き込むファイルを選択します。 FlashROM に書き込むファイルは、実行したいビルドモードのフォルダ(Debug)内の**.srec ファイル**を指定します。 (図は、AP-S7G2-0A 用 UART サンプルプログラムの.srec ファイルです)

ap_s7	7g2_0a_sample_uart > Debug >	~	ට 🔎 Debugの	検索	
^	名前		更新日時	種類	サイズ
r i	src		2020/10/14 17:22	ファイル フォルダー	
	synergy		2020/10/14 17:22	ファイル フォルダー	
	ap_s7g2_0a_sample_uart.elf		2020/10/14 17:22	ELF ファイル	9
·	ap_s7g2_0a_sample_uart.elf.in		2020/10/14 17:22	IN ファイル	
	🖺 ap_s7g2_0a_sample_uart.map		2020/10/14 17:22	MAP ファイル	4
	🔄 ap_s7g2_0a_sample_uart.srec		2020/10/14 17:22	SREC ファイル	
	makefile		2020/10/14 17:22	ファイル	
	🖺 makefile.init		2020/10/14 17:22	INIT ファイル	
	🔂 objects.mk		2020/10/14 17:22	Makefile	
	🔂 sources.mk		2020/10/14 17:22	Makefile	

- ⑦ ファイルの指定後、一度電源を切断し、再投入します。
- ⑧ [スタート] ボタンを押して書き込みを開始します。

ファイル(E) デバイス情報(D) ヘルプ(H)
操作 操作設定 ブロック設定 接続設定 ユニークコード
プロジェクト情報 現在のプロジェクト: ap_s7を2_rfp.rpj マイクロコントローラ: Renesas Synerey プログラムファイル C:¥Renesas¥workspace¥ap_s7 <u>e2_0a_sample_uart*Debug¥ap_s7e2_0a_sample_uart.srec</u> CRC-32: 55A2F645 フラッシュ操作 消去 >> 書き込み >> パリファイ スタート(S)
Device Code : 01
Hirmware Version: V1.5 Code Flash 1 (アドレス: 0x00000000、サイズ: 64 K、消去サイズ: 8 K) Code Flash 1 (アドレス: 0x00010000、サイズ: 4032 K、消去サイズ: 32 K) Data Flash 1 (アドレス: 0x40100000、サイズ: 64 K、消去サイズ: 64)
Config Area (アドレス:0x40120040、サイズ:128、消去サイズ:0) ツールから切断します。 抹作が成功しました。
マ ステータスとメッセージのクリア(C)

- ⑨ プログレスバーが表示され、書き込みの進捗に合わせて変化します。
- ① プログレスバーの表示が消え、メイン画面に [正常終了] と表示されれば、正常に書き込みが終了しています。
 CPU ボードの電源を切り、その後 RFP を終了します。
 以上で FlashROM の書き込みは終了です。

ファイル(<u>F</u>)) デバイス情報(<u>D</u>) ヘルプ(<u>H</u>)				
操作操	作設定 ブロック設定 接続設定 ユニークコード				
プロジュ 現在 マイク プログラ	プロジェクト情報 現在のプロジェクト: マイクロコントローラ: Renesas Synergy プログラムファイル				
C:¥F	Renesas¥workspace¥ap_s7g2_0a_sample_uart¥Debug¥ap_s7g2_0a_sample_uartsrec 参照…(B) CRC-32:55A2F64B				
- フラッシ 消去	「山井作 こ >> 書き込み >> ベリファイ				
	スタート(<u>S</u>) 正常終了				

① CPU ボードをサンプルプログラム動作時のボード設定に変更し、再度電源を投入してサンプルプログラムが動作することを確認します

<AP-S7G2-0A / AP-S5D9-0A>



JSW1:ON ボード上の SDRAM を使用する

<AP-S3A7-0A>



JSW2 : SGL シングルチップモード



JSW2 : SGL シングルチップモード

再度同じ設定でファイルを書き込む際は、同じプロジェクトを開くことで®から作業を開始できます。 その他の機能など RFP の詳細につきましては、RFP のユーザーズマニュアルを参照してください。

注意:プロジェクトは、設定時に接続していた CPU に対応しています。異なる CPU の書き込みには使用できません。 例えば、AP-S7G2-0A と接続して作成した RFP のプロジェクトを使用して AP-S5D9-0A に書き込むことは できませんので、AP-S5D9-0A 用にプロジェクトを新規作成してください。

改定履歴

版数	更新日	改定内容
1.0 版	2018/04/23	新規作成
2.0 版	2020/10/14	EWARM 環境の記述を削除
2.1 版	2020/10/15	SSP バージョンを v1.4.0 から v1.7.8 へ更新
3.0 版	2021/11/10	SSP バージョンを v1.7.8 から v2.1.0 へ更新
3.1 版	2023/10/02	住所を更新
	1	

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 S7G2、S5D9、S3A7 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。
- ・Renesas Flash Programmer 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。

