Rev 3.00.00

エーワン製サンプルを e2studio にインポートする方法

(e2studio Version:2022-7/SSP Version 2.3.0)

1. サンプルのワークスペース名とプロジェクト名の説明

1-1. MP-S3A7-01 用サンプルの説明

御購入 CD に収めてある「S3A7_e2std_SSPx_x_x.zip」を適当なフォルダに Copy して解 凍する。

フォルダ名・	プロジェクト名	説明
ワークスペース名		
EV-SYNERGY-S3A7		S3A7 用評価基板側の共通 C
		ソース
MP-S3A7-01		S3A7 基板上デバイスの共通
		Cソース
S3A7_e2std_BLDC	BLDC_Drive_HAL	HAL オンリー DC ブラシ
		レスモータ制御
	BLDC_Drive_STD	Azure RTOS 対応 DC ブラ
		シレスモータ制御
S3A7_e2std_CAN	CAN_HAL	HALオンリー CAN ループ
		バック通信
	CAN_STD	Azure RTOS 対応 CAN ル
		ーブバック通信
S3A7_e2std_CharLCD	CharLCD_HAL	HAL オンリー キャラクタ
		LCD 表示
	CharLCD_STD	Azure RIOS 対応 キャフク
S3A7_e2std_demo	S3A7_demo_S1D_F	Azure RIUS 対応 MP-S3A7-
S3A7_e2std_ENC	ENC_HAL	HAL $\pi y y = \pm y = -y$
	ENIC COD	
	ENC_SID	Azure RIUS 対応 エンヨー
		クハル A PTOC 社会 証価世紀
S3A1_e2sta_Ev_Sw_LED	EV_SW_LED_SID	Azure RIOS 対応 評価基似 トSW オカトIFD 出力
S2A7 o2ctd IVD		
SOAT_e2su_LVD	MIKDOF 1994 IDO STD	
SJA7_e2su_wiki0D05	MIIMOL_1024_IIW2_51D	の LED 表示・ IRQ
S3A7_e2std_NMI_FRAM	NMI_HAL	HALオンリー NMI
	NMI_STD	Azure RTOS 対応 NMI
S3A7_e2std_RTC	RTC_HAL	HALオンリー RTC
	RTC_STD	Azure RTOS 対応 RTC
S3A7_e2std_SCI	SCI_HAL	HALオンリー SCI

	SCI_STD	Azure RTOS 対応 SCI		
S3A7_e2std_sFLASH	sFlash_SPI_DTC	Azure RTOS 対応		
		DTC 対応 SerialFlashROM-		
		Read/Write		
	sFlash_SPI_HAL	HALオンリー		
		SerialFlashROM-Read/Write		
	sFlash_SPI_STD	Azure RTOS 対応		
		SerialFlashROM-Read/Write		
S3A7_e2std_sFRAM	sFram_SPI_DTC	Azure RTOS 対応		
		DTC 対応 FRAM-Read/Write		
	sfram_SPI_HAL			
		FRAM—Read/Write		
	sfram_SPI_SID	Azure RIOS 对応		
COA7 operal LICDV	LICE CDCACM Derice	FRAM-Read/Write		
S3A7_e2sta_USBA	USB_CDCACM_Device	Azure RIOS 约応 USP-Eurotion CDC-ACM 通		
		CSD-Function CDC-ACM 通 信		
	USB HMSC Host	口 Azume RTOS 対応		
		USB-Host USBメエリ		
		Read/Write		
S3A7 e2std WiFi NetX 1	WiFi DHCP TCP GT202 PMO	Azure RTOS 対応 WiFi モ		
	DA	ジュール GT202 を PMODA		
		に接続した DHCP・TCP 通		
		信		
	WiFi_DHCP_TCP_GT202_PMO	Azure RTOS 対応 WiFi モ		
	DB	ジュール GT202 を PMODB		
		に接続した DHCP・TCP 通		
		信		
	WiFi_DHCP_TCP_GT202_PMO	Azure RTOS 対応 WiFi モ		
	DC	ジュール GT202 を PMODC		
		に接続した DHCP・TCP 通		
	WiFi_DHCP_UDP_G1202_PMO	Azure RIOS 対応 WiFi モ		
	DA	ンユール G1202 を PMODA		
		に安祝した DHCP・UDP 通 信		
S347 oPetd WiFi NotV 9	WIE DHOP HTTP OTOO DM	旧 Azumo RTOS 対応 WEF エ		
DOM/_E4SUL_VVII'I_IVELA_2	ODA			
	ODA	に接続した DHCP・HTTP		
		如理		
	WiFi DHCP TCP HTTP GT202	Azure RTOS 対応 WiFi モ		
	PMODA	ジュール GT202 を PMODA		
		に接続した DHCP・TCP 通		
		信・HTTP処理		
	WiFi_DHCP_UDP_HTTP_GT20	Azure RTOS 対応 WiFi モ		
	2_PMODA	ジュール GT202 を PMODA		
		に接続した DHCP・UDP 通		
		信・HTTP処理		
S3A7_e2std_WiFi_NetX_3	WiFi_HTTP_GT202_PMODA	Azure RTOS 対応 WiFi モ		
		ジュール GT202 を PMODA		
		に接続した固定 IP・HTTP		



	処理
WiFi_TCP_HTTP_GT202_PMO	Azure RTOS 対応 WiFi モ
DA	ジュール GT202 を PMODA
	に接続した固定 IP・TCP 通
	信・HTTP処理
WiFi_UDP_HTTP_GT202_PMO	Azure RTOS 対応 WiFi モ
DA	ジュール GT202 を PMODA
	に接続した固定 IP・UDP 通
	信・HTTP処理

1-2. MP-S5D9-01 用サンプルの説明

御購入 CD に収めてある「S5D9_e2std_SSPx_x_x.zip」を適当なフォルダに Copy して解 凍する。

フォルダ名・	プロジェクト名	説明
ワークスペース名		
EV-SYNERGY-S5D9		S5D9 用評価基板側の共通 C
		ソース
MP-S5D9-01		S7G2 基板上デバイスの共通
S5D9_e2std_BLDC	BLDC_Drive_HAL	HAL オンリー DC フラシレ
	DIDC Drive CTD	スモーク制御 Amme PTOC 対応 DC ブラシノ
	BLDC_Drive_STD	Azure KIOS 利心 DC ノノン
S5D9 e2std CAN	CAN HAL	HAL オンリー CAN ループ
		バック通信
	CAN STD	Azure RTOS 対応 CAN ルー
		プバック通信
S5D9_e2std_demo	S5D9_demo_STD	Azure RTOS 対応 MP-S5D9-01
		のデモプログラム
S5D9_e2std_E2P	e2p_I2C_DTC	Azure RTOS 対応 DTC 対応
		E2PROM © Read/Write
	e2p_I2C_HAL	HAL オンリー E2PROM の
		Read/Write
	e2p_I2C_STD	Azure RIOS 对応 E2PROM
SEDO ageta ENIC	ENIC HAL	U Kead/Write
S5D9_e2su_ENC	ENC_HAL	
	EINC_SID	Azure KTOS 刘応 Endoder 八 力
S5D9 e2std EV SW LED	EV SW LED STD	Azure RTOS 対応 評価基板
		上SW入力とLED出力
S5D9_e2std_GLCDC	GLCDC_I2C_SPI_STD	Azure RTOS 対応 グラフィ
		ック LCD 表示
S5D9_e2std_LVD	LVD_HAL	HALオンリー IVD
S5D9_e2std_MikroBUS	MIKROE_1824_STD	Azure RTOS 対応 MikroBus
		の LED 表示・IRQ
S5D9_e2std_NMI_FRAM	NMI_HAL	HALオンリー NMI
	NMI_STD	Azure RTOS 対応 NMI
S5D9_e2std_RTC	RTC_HAL	HALオンリー RTC
	RTC_STD	Azure RTOS 対応 RTC
S5D9_e2std_SCI	SCI_HAL	HALオンリー SCI
	SCI_STD	Azure RTOS 対応 SCI
S5D9_e2std_sFLASH	sFlash_SPI_DTC	ThreadX対応
		DTC 対応 SerialFlashROM-
		Read/Write
	st'lash_SPI_HAL	HALオンリー
	aFlach SPI STID	Serialli TashKUM-Kead/Write
	STIASILOLI OLI	AZULEINIUG 对心

		SeriallFlashROM-Read/Write
S5D9_e2std_sFRAM	sFram_SPI_DTC	Azure RTOS 対応
		DTC 対応 FRAM-Read/Write
	sFram_SPI_HAL	HALオンリー
		FRAM-Read/Write
	sFram_SPI_STD	Azure RTOS 対応
		FRAM-Read/Write
S5D9_e2std_USBX	USB_CDCACM_Device	Azure RTOS 対応
		USB-Function CDC-ACM 通信
	USB_HMSC_Host	Azure RTOS 対応
		USB-Host USBメモリ
		Read/Write
S5D9_e2std_WiFi_NetX_1	WiFi_DHCP_TCP_GT202_P	Azure RTOS 対応 WiFi モジ
	MODA	ュール GT202 を PMODA に
		接続した DHCP・TCP 通信
	WiFi DHCP TCP GT202 P	Azure RTOS 対応 WiFi モジ
	MODB	ュール GT202 を PMODB に
		接続した DHCP・TCP 通信
	WiFi DHCP TCP GT202 P	Azure RTOS 対応 WiFi モジ
	MODC	ュール GT202 を PMODC に
		接続した DHCP・TCP 通信
	WiFi DHCP UDP GT202 P	Azure RTOS 対応 WiFi モジ
	MODA	ュール GT202 を PMODA に
		接続した DHCP・UDP 通信
S5D9 e2std WiFi NetX 2	WiFi DHCP HTTP GT202 P	Azure RTOS 対応 WiFi モジ
	MODA	$\pi - \mu$ GT202 & PMODA
		接続した DHCP・HTTP 処理
	WIFI DHCP TCP HTTP GT	Azure BTOS 対応 WiFi モジ
	202 PMODA	$\pi - \mu$ GT202 & PMODA
		接続した DHCP・TCP 通信・
		HTTP 如理
	WiFi DHCP LIDP HTTP GT	Azure BTOS 対応 WiFi モジ
	202 PMODA	$\pi - \mu$ GT202 & PMODA
		接続した DHCP・IIDP 通
		信・HTTP 机 で た BHO ED E D E E E E E E E E E E E E E
S5D9 e2std WiFi NetX 3	WiFi HTTP GT202 PMODA	Azura BTOS 対応 WiFi モジ
DDDJ_e2su_will1_ivetx_5	WIT_IIIII_01202_1WODA	$\pi - \eta CT202 \approx PMODA$
	WIEI TOP HTTP GT202 PM	Azure BTOS 対応 WiFi チジ
	ODA	r = h GT202 $E PMODA$
	WIEI LIDP HTTP CTOOP P	Azume RTOS 対応 WiFi チジ
	MODA	$\gamma - \mu GT909 \Rightarrow PMODA \downarrow \gamma$
		接続した固定 IP・IIDP 通
		信·HTTP机理
S5D9 e2std WIRE Not¥ 1	WIRE DHCP HTTP FTH	Azume BTOS 対応 右線 LAN
		接続したDHCP・HTTP加理
	WIKE_DHCP_ICP_AES_ET	Azure KIUS 灯心 有線 LAN
	11	仮祝した DHCP・ICP 通信に

		おいて SCE7[暗号 AES]を使 用
	WIRE_DHCP_TCP_ETH	Azure RTOS 対応 有線 LAN 接続した DHCP・TCP 通信
	WIRE_DHCP_TCP_HTTP_E TH	Azure RTOS 対応 有線 LAN 接続した DHCP・TCP 通信・ HTTP 処理
	WIRE_DHCP_UDP_AES_ET H	Azure RTOS 対応 有線 LAN 接続した DHCP・UDP 通信 において SCE7[暗号 AES]を 使用
	WIRE_DHCP_UDP_ETH	Azure RTOS 対応 有線 LAN 接続した DHCP・UDP 通信
	WIRE_DHCP_UDP_HTTP_E TH	Azure RTOS 対応 有線 LAN 接続した DHCP・UDP 通 信・HTTP処理
S5D9_e2std_WIRE_NetX_2	WIRE_HTTP_ETH	Azure RTOS 対応 有線 LAN 接続した固定 IP・HTTP 処理
	WIRE_TCP_ETH	Azure RTOS 対応 有線 LAN 接続した固定 IP・TCP 通信処 理
	WIRE_TCP_HTTP_ETH	Azure RTOS 対応 有線 LAN 接続した固定 IP・TCP 通信・ HTTP 処理
	WIRE_UDP_ETH	Azure RTOS 対応 有線 LAN 接続した固定 IP・UDP 通信 処理
	WIRE_UDP_HTTP_ETH	Azure RTOS 対応 有線 LAN 接続した固定 IP・UDP 通 信・HTTP処理

1-3. MP-S7G2-01 用サンプルの説明

御購入 CD に収めてある「S7G2_e2std_SSPx_x_x.zip」を適当なフォルダに Copy して解 凍する。

フォルダ名・	プロジェクト名	説明
ワークスペース名		
EV-SYNERGY-S7G2		S7G2 用評価基板側の共通 C
		ソース
MP-S7G2-01		S7G2 基板上デバイスの共通
		Cソース
S7G2_e2std_BLDC	BLDC_Drive_HAL	HAL オンリー DC ブラシレ
		スモータ制御
	BLDC_Drive_STD	Azure RTOS 対応 DC ブラシ
		レスモータ制御
S7G2_e2std_CAN	CAN_HAL	HAL オンリー CAN ループ
		バック通信
	CAN_STD	Azure RIOS 対応 CAN ルー
		フパック通信
S7G2_e2std_E2P	e2p_12C_D1C	Azure RIUS 对心 DIC 对心
		E2PROM () Read/Write
	e2p_12C_HAL	HAL X V E2PROM ()
	om IOC STD	Kead/Write Arumo PTOS 社内 F2DPOM
	e2p_120_51D	D Road/Write
S7G2 e2std demo	S7G2 demo STD	Azume BTOS 対応 MP-S7G2-01
		のデモプログラム
S7G2 e2std ENC	ENC HAL	HALオンリー Encoder 入力
	ENC STD	Azure BTOS 対応 Encoder 入
		力
S7G2 e2std EV SW LED	EV SW LED STD	Azure RTOS 対応 評価基板
		上 SW 入力と LED 出力
S7G2_e2std_GLCDC	GLCDC_I2C_SPI_STD	Azure RTOS 対応 グラフィ
		ック LCD 表示
S7G2_e2std_LVD	LVD_HAL	HALオンリー LVD
S7G2_e2std_MikroBUS	MIKROE_1824_STD	Azure RTOS 対応 MikroBus
		の LED 表示・IRQ
S7G2_e2std_NMI_FRAM	NMI_HAL	HALオンリー NMI
	NMI STD	Azure RTOS 対応 NMI
S7G2_e2std_RTC	RTC_HAL	HALオンリー RTC
	RTC_STD	Azure RTOS 対応 RTC
S7G2_e2std_SCI	SCI_HAL	HALオンリー SCI
	SCI STD	Azure RTOS 対応 SCI
S7G2 e2std sFLASH	sFlash SPI DTC	Azure RTOS 対応
		DTC 対応 SerialFlashROM-
		Read/Write
	sFlash_SPI_HAL	HALオンリー
		SeriallFlashROM-Read/Write
	sFlash_SPI_STD	AzureRTOS対応

		SeriallFlashROM-Read/Write					
S7G2_e2std_sFRAM	sFram_SPI_DTC	Azure RTOS 対応					
		DTC 対応 FRAM-Read/Write					
	sFram_SPI_HAL	HALオンリー					
		FRAM-Read/Write					
	sFram_SPI_STD	Azure RTOS 対応					
		FRAM-Read/Write					
S7G2_e2std_USBX	USB_CDCACM_Device	Azure RTOS 対応					
		USB-Function CDC-ACM 通信					
	USB_HMSC_Host	Azure RTOS 対応					
		USB-Host USBメモリ					
		Read/Write					
S7G2_e2std_WiFi_NetX_1	WiFi_DHCP_TCP_GT202_P	Azure RIOS 対応 WiFi モジ					
	MODA	ュール GT202 を PMODA に					
		接続した DHCP・TCP 通信					
	WiFi_DHCP_TCP_GT202_P	Azure RIOS 対応 WiFi モジ					
	MODB	ュール GI202 を PMODB に					
		接続した DHCP・TCP 通信					
	WiFi_DHCP_TCP_GT202_P	Azure RIOS 対応 WiFi モジ					
	MODC	ュール GT202 を PMODC に					
		接続した DHCP・TCP 通信					
	WiFi_DHCP_UDP_GT202_P	Azure RTOS 対応 WiFi モジ					
	MODA	ュール GT202 を PMODA に					
		接続した DHCP・UDP 通信					
S7G2_e2std_WiFi_NetX_2	WiFi_DHCP_HTTP_GT202_P	Azure RTOS 対応 WiFi モジ					
	MODA	ュール GT202 を PMODA に					
		接続した DHCP・HTTP 処理					
	WiFi_DHCP_TCP_HTTP_GT	Azure RIOS 対応 WiFi モジ					
	202_PMODA	$\neg -\mu$ GI202 & PMODA C					
		接続した DHCP・TCP 通信・					
		HTTP処理					
	WiFi_DHCP_UDP_HTTP_GT	Azure RTOS 対応 WiFi モジ					
	202_PMODA	ュール GT202 を PMODA に					
		接続した DHCP・UDP 通					
		信・HTTP処理					
S7G2_e2std_WiFi_NetX_3	WiFi_HTTP_GT202_PMODA	Azure RTOS 対応 WiFi モジ					
		ュール GT202 を PMODA に					
		接続した固定 IP・HTTP 処理					
	WiFi_TCP_HTTP_GT202_PM	Azure RTOS 対応 WiFi モジ					
	ODA	ュール GT202 を PMODA に					
		接続した固定 IP・TCP 通信・					
		HITP処理					
	WiFi_UDP_HTTP_GT202_P	Azure RTOS 対応 WiFi モジ					
	MODA	ュール GT202 を PMODA に					
		接続した固定 IP・UDP 通					
		信・HTTP処理					
S7G2_e2std_WIRE_NetX_1	WIRE_DHCP_HTTP_ETH	Azure RTOS 対応 有線 LAN					
		接続した DHCP・HTTP 処理					
	WIRE_DHCP_TCP_AES_ET	Azure RTOS 対応 有線 LAN					
	H	接続した DHCP・TCP 通信に					

		おいて SCE7[暗号 AES]を使 用
	WIRE_DHCP_TCP_ETH	Azure RTOS 対応 有線 LAN 接続した DHCP・TCP 通信
	WIRE_DHCP_TCP_HTTP_E TH	Azure RTOS 対応 有線 LAN 接続した DHCP・TCP 通信・ HTTP 処理
	WIRE_DHCP_UDP_AES_ET H	Azure RTOS 対応 有線 LAN 接続した DHCP・UDP 通信 において SCE7[暗号 AES]を 使用
	WIRE_DHCP_UDP_ETH	Azure RTOS 対応有線 LAN接続した DHCP・UDP 通信
	WIRE_DHCP_UDP_HTTP_E TH	Azure RTOS 対応 有線 LAN 接続した DHCP・UDP 通 信・HTTP処理
S7G2_e2std_WIRE_NetX_2	WIRE_HTTP_ETH	Azure RTOS 対応有線 LAN接続した固定 IP・HTTP 処理
	WIRE_TCP_ETH	Azure RTOS 対応 有線 LAN 接続した固定 IP・TCP 通信処 理
	WIRE_TCP_HTTP_ETH	Azure RTOS 対応 有線 LAN 接続した固定 IP・TCP 通信・ HTTP 処理
	WIRE_UDP_ETH	Azure RTOS 対応 有線 LAN 接続した固定 IP・UDP 通信 処理
	WIRE_UDP_HTTP_ETH	Azure RTOS 対応 有線 LAN 接続した固定 IP・UDP 通 信・HTTP処理

2. e2studio へのインポート方法

ワークスペース名「S7G2_e2std_BLDC」、 プロジェクト名「BLDC_Drive_HAL・BLDC_Drive_STD」を一例としてインポート方法を記 述します。なお、他のワークスペースではこの説明を参考にしてインポートして下さい。

2-1. ワークスペース名の指定





2-2. プロジェクトのインポート「BLDC_Drive_HAL・BLDC_Drive_STD」



	バ(F) 福来(E) ノース(3) リノアワラリノン (5) (1) (1) (3) (1) (1) (1) (1) (1) (1) (1) (1) (1) (1	(1) JCJ=P(IN)	(使飛(A) ,	U919F(P)	Kenesas	Views	夫1](K)	912190	VV) (\)			
	新規(N) ファイルを問(()	Alt+97F+N>	Configurat	ons	~	オン:				× 🌣		
5	ファイル・システムからプロジェクトを開く				<u>,#</u> ; }	ÌQ	※ 袋	: • % •		- · - • •	_	
	最近のファイル	>	- 47 (F	/ ; /	10				(Q : 🖽 🖪	10 C/C	++
	閉じる(C)	Ctrl+W								アウトライ ×		E
	すべて閉じる(L)	Ctrl+シフト+W										
	保存(S)	Ctrl+S							アウブな	ドラインを提供す	ョアク ません。	τ1
	名前を付けて保存(A)											
	すべて保存(E)	Ctrl+シフト+S										
	前回保存した状態に戻す(T)		_									
	移動(V)											
	名前を変更(M)	F2										
2	更新(F)	F5										
	行区切り文字の変換(D)	>	_									
Ð	印刷(P)	Ctrl+P			1.0	1	21	4~/- ^{1_} +	٦			
21	インポート(I) <u>インポート</u>			12	ホー	<u> </u>	ど	首正				
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		ル 🧠 スマ	ート・ブラウザー	🖓 77-	ŀ•₹⊒⊐	アル			7		E
	終了(X)		_									-
	EL 3	业/1元9月	-			リソース		77		07-932		垩
	5											

el 12#−ト – □ ×	
選択 アーカイブ・ファイルまたはディレクトリーから新規プロジェクトを作成します。	
インポート・ウィザードの選択(S):	
フィルタ入力	
 ← 一般 CMSIS Pack]
 □ フォルダ-またはア-カイブ由来のプロジェクトをワークスペースへ □ 設定 > > C/C++ 	
②「次へ」クリック	
(P) < 戻る(B) 次へ(N) > 終了(F) キャンセル	

		– D X
プロジェクトをインボート 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(T):		→ 参照(R)
ブロジェクト(P):	「参照」クリック	✓ ≫ Яस(K)
		すべて選択(S) 選択をすべて解除(D)
		更新(E)
オブション ネストし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ピー(C) 一完了次第、新しくインポートしたプロジェクトを閉じる ロワークスペースに既に存在するプロジェクトを隠す(i)	(o)	
ワーキング・セット		
ロ ワーキング・セットにプロジェクトを追加(T) ワーキング・セット(O):		新規(₩) → 選択(E)
? < 戻る(B)	次へ(N) > 終了(F)	キャンセル





🖻 インポート		– D 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。	
 ・ディレクトリーの選択(T): C:¥WorkSp アーカイブ・ファイルの選択(A): プロジェクト(P): 	ace¥S7G2_e2std_SSP2_3_0_Sample¥S7G	2. ✔ 参照(R) ✔ 参照(R)
BLDC_Drive_HAL(C:¥WorkSpace¥S70 BLDC_Drive_STD(C:¥WorkSpace¥S70	G2_e2std_SSP2_3_0_Sample¥S7G2_e2std ;2_e2std_SSP2_3_0_Sample¥S7G2_e2std_	すべて選択(S) 選択をすべて解除(D)
<	>	更新(E)
オブション ネストし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ピー(C) 一完了次第、新しくインポートしたプロジェクトを 「ワークスペースに既に存在するプロジェクトを願	■確認 ^{閉じる(o)} 討(i)	
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(T)		新規(W) ※ 選択(E)
? < 戻る	「終了」クリック (B) 次へ(N) > 終了(F)	クキャンセル



Image: S7G2_e2std_BLDC - 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) プロジェクト(P) Renesas Views 実行(R) ウィンドウ(W) ヘルプ(H)
🐔 🐐 🔳 漆 デパッグ(B) 🗸 🖻 BLDC_Drive_STD Debug 🗸 🄅
▼ = = = = < < < < < < < < < < < < < < <
E 気 Y * >
 ・・ブラウザー 印 スマート・マニュアル □ □ ・・マニュアル □ □ ・・・マニュアル □ □ ・・・マニュアル □ □ ・・・マニュアル □ □ ・・・・マニュアル □ □ ・・・・マニュアル □ □ ・・・・・・・・・・・・・・・・・・・・・・・・・・
Syntergy

2-3. プロジェクトのコンフィグレーション登録

「BLDC_Drive_HAL」「BLDC_Drive_STD」の「コンフィグレーション登録」は同操作の ため、「BLDC_Drive_STD」を一例にして説明します。1



・「BLDC_Drive_HAL」は、上記と同じ操作をして下さい。

2-4. ビルドオール

「BLDC_Drive_HAL」「BLDC_Drive_STD」の「ビルドオール」は同操作のため、 「BLDC_Drive_STD」を一例にして説明します。

S7G2_e2std_BLDC - BLDC_E	Drive_STD/configuration.xml - e	² 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) プロジェ	ːクト(P) Renesas Views	実行(R)	
ウィンドウ(W) ヘルプ(H)		フ	プロジェクトを開く(E)		
🐔 🎋 🔳 🎄 デバ	ッグ(B) 🗸 💽 BLDC	_Drive_STD 7	プロジェクトを閉じる(S)		
	🔜 : 🥖 : 📸 👻 🚳 👻 🔞 .	- 😪 - i C	Open Synergy Configurat	tion	
: <u>````````````````````````````````````</u>	. m • m • ¶	: @ : b 🔜 🕯	すべてドルド(A)	Ctrl-	+Alt+B
• Max 177		· •••• · <u>- •••</u>	「ルド林」成		>
			プロジェートのビルド(B)		Ctrl+B
🛅 プロジェクト・ X 🕒 🗖	[BLDC_Drive_STD] Synergy	Configura	フーキン マットのドルド(W)		>
	Threads Configuration	on n	711-20		
> 🚰 BLDC_Drive_HAL		é	与新的((M)		
BLDC_Drive_STD [Debu	New Thread				
> P src	Threads	HAL/G	ビルド・ター		>
> 🔑 synergy	Remove L	C	C/C++ 1		>
> 🗁 script	✓ 💒 HAL/Common ∧	🐠 e² 🦻	すべての依新		Alt+D
> 🗁 synergy_cfg	⊕ g_cgc coc Dr ⊕ g ioport I/O F		Change D		
BLDC_Drive_STD Debi	🖑 g_fmi FMI Driv ,	「プロジェク	クトーメニュー		
configuration.xml	< >	「オベてを」	ー ビルドレカリック	7 04	A 14 - D
PIN-EV-SYNERGY-S70				Ctri	+AIt+P
R7FS7G27H3A01CFC.	HAL/Comm 🐑 New Objec				_
synergy_cfg.txt	on Objects 🛛 🔬 Remove				~
> (?) Developer Assistance		<			>
	Summany BSD Clocks Dins	Threads Messagin	on Components		
		inclus messagin			
	- 🦉 問題 🕒 コンソール 🗙 👒	> スマート・フラウサー	□□□ スマート・マニュアル		
				V 🛃 🖬 🖌	- 🖾 -
	Synergy Extracting file: synerg	w/sen/inc/fram	nework/el/ty nort /		
	Extracting file: synerg	gy/ssp/inc/fram	nework/el/tx_trace	.h	
	Extracting file: synerg	gy/ssp/src/fram	nework/el/tx/cm4_go	cc/libtx.a	
					~
< >	<				>
		A			
		<u> </u>			
		B. 77-6.7	שר		
				-	
CDT หั้มหังอ่างหลาย เกิด	Drive STD1				
Building target: BL	DC_Drive_STD.elf				~
arm-none-eabi-objco	py -O srec "BLDC_Dr	ive_STD.elf	" "BLDC_Drive_	STD.srec	
arm-none-eabi-size	TORMAT=DERKELEY " bss dec be	BLDC_Drive_9 x filename	SID.elt"		
62824 224	9104 72152 119d	8 BLDC_Drive	e_STD.elf		
11.00.42 Puild Sini	shed 0 enters 0 to	annings (*	ook 45 406m-		
11:09:45 BUILD F101	sned. 0 errors, 0 w	arnings. (to	00K 45.496MS)		~
					7

Build Finished 0 errors. 0 warnings.(.....)確認

・「BLDC_Drive_HAL」は、上記と同じ操作をして下さい。



2-5. MP-S3A7-01 のジャンパー設定と JTAG デバッガとの接続

2-6. MP-S5D9-01 のジャンパー設定と JTAG デバッガとの接続





2-7. MP-S7G2-01 のジャンパー設定と JTAG デバッガとの接続

- 3. デバッグ操作
 - 3-1. デバッグ構成の設定(S7G2)

「BLDC_Drive_HAL」「BLDC_Drive_STD」の「デバッグの構成」は同操作のため、 「BLDC_Drive_STD」を一例にして説明します。





ー 構成の作成、管理、および実行	Ŕ	5
 ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	名前(N): BLDC_Drive_STD Debug ■ Xイン 弥 Debugger ▶ Startup ッソース ■ 共通(C) プロジェクト(P): BLDC_Drive_STD 参照(B). C/C++ アプリケーション: Debug/BLDC_Drive_STD.elf 参照(V) プロジェクトの検索(H) 参照(R). 起動前に必要に応じてビルド Build Configuration: Use Active ① 「Renesas GDB Hardware Debugging」の 「BLDC_Drive_STD_Debug」 を選択 前回保管した状態に戻す(V) 適用(Y)	······································

 「「「」」」」」 「」」」 「」」」	Ś
	参照(B) 参照(R) 多 5



 デバッグ構成 構成の作成、管理、および実行 Run Break Time Measurement Operating 	Frequenc ①「Debugger」タグ選択 .001 & 10000.0MHz
 □ 図 20 回 × □ マ □ □ □ □ <th>名前(N): BLDC ● STD Debug ● Startup 「シソース □ 共通(C) Debug hardware: J-Link ARM ✓ Target Device: R7FS7G27H … GDB Setting nection Settings デバッグ・ツール ② 「J-Link ARM」確認 ③ 「R7FS7G2H」確認 ○ リモート GDB サーバーへ接続 GDB ポート番号: 61234 接続タイムアウト(s): 30 GDB</th>	名前(N): BLDC ● STD Debug ● Startup 「シソース □ 共通(C) Debug hardware: J-Link ARM ✓ Target Device: R7FS7G27H … GDB Setting nection Settings デバッグ・ツール ② 「J-Link ARM」確認 ③ 「R7FS7G2H」確認 ○ リモート GDB サーバーへ接続 GDB ポート番号: 61234 接続タイムアウト(s): 30 GDB
< > > 16 項目のうち 14 項目がフィルターに一致	前回保管した状態に戻す(V) 適用(Y)
?	デバッグ(D) 閉じる



<u>3-2.</u>デバッグ開始

e S7G2_e2std_BLDC - BLDC_Drive_STD/synergy/ssp/src/bsp/cmsis/Device/RENESAS/S7G2/Source/startup_S7G2.c - e ² 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) プロジェクト(P) Renesas Views 実行(R) ウィンドウ(W) ヘルプ	(H)	
🔦 🗱 🔳 操デバッグ(B) 🗸 🖻 BLDC_Drive_STD Debug 🗸 🔅 🖬 😨 🗞 🗸 🐻	. ø 🔊	ج 🤌 🖻
[⊿] \$∰ ! `\ <mark> </mark>		
	C/C++	参 デバッグ
★ デパッグ ×	(x)= 🔀	» ₆ □ □
ELDC_Drive_STD Debug [Renesas GDB Hardware Debugging] StD of [1] [cores: 0]	1 1	3 📑 🖻
Conversion of the state of	00	
Reset_Handler() at startup_S7G2.c:60 0xa7ac	名前 型	^
arm-none-eabi-gdb (7.8.2)		
Kenesas GDB server (Host)		
(BLDC_Drive_STD] Synergy Configuration Istartup_STG2.c ×		
⇒ 60 0000a7ac { /* T_itialize surface DCD */		
62 0000a7b0 SystemInit();		
63 64 /* Call user application */		
65 0000a7b4 main();		
66 67 0000a7b8 ⊖ while (1)		
		v
<pre>/* Intibits Incom */ /* Intibits Incom */ >></pre>	<	\rightarrow \leftrightarrow
🗐 コンソール 🗙 🔠 レジスター 📋 デパッグ・シェル 🖹 問題 📓 Debugger Console 🦓 スマート・ブラウザー 📋 メモリー		- 0
🗏 🗙 💥 🗟 🛃 🚱 🛃	- 🗶 🛃	💷 🕶 📑 🕶
BLDC_Drive_STD Debug [Renesas GDB Hardware Debugging] [pid: 32]		
		Ŷ
		>

😨 S7G2_e2std_BLDC - BLDC_Drive_STD/src/synergy_gen/main.c - 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) プロジェクト(P) Renesas Views 実行(R) ウィンドウ(W) ヘルフ	^β (H)
🔏 🔺 🔳 🚸 デバッグ(B) 🗸 🖻 BLDC_Drive_STD Debug 🗸 🌼 🗄 🐑 🖷 🐘 🛞 💌 🐇 🗸	1) 🖉 😂 🎘 🗸 🗸
· / : ≒ : ` ▶ = X 3. ?3. i→ ≂ ∞ : 2 3 + • 4 + : 4 + 5 m + 12 3 42 : 30 : 30	
	(x)= × », □
▼ EBLDC_Drive_STD Debug [Reness]	
V 🔐 BLDC_Drive_STD.elf [1] [cores: 0]	
✓ m [®] Thread #1 1 (single core) [core: 0] (Suspended : ブレークポイント)	~ 名前 型 ^
arm-none-eabi-gdb (7.8.2)	
Renesas GDB server (Host)	
🕮 (BLDC: Drive STD) Synergy Configuration 🗋 startup S7G2.c 🔲 main.c 🗙	
	1
127 ⊖ int main(void)	
129	
130 0000abe8 tx_kernel_enter ();	
132 0000abec return 0;	
133 0000abee }	
	< > <> ×
😑 コンソール 🗙 🚟 レジスター 🗊 デパッグ・シェル 🔝 問題 💀 Debugger Console 🛞 スマート・ブラウザー 👖 メモリー	- 8
BLDC_Drive_STD Debug [Renesas GDB Hardware Debugging] [pid: 32]	
ターゲット接続状況 - OK ダウンロード開始	<u></u>
	>
+br+ → 0x0000abe4 🧭 0 ns 🌐 DWT	4 <mark>8</mark> 0



【サンプル実行画面】

M	COM	I - Tera Ter	m VT							—		×
ファ	イル(F)	編集(E)	設定(S)	コントロール(0)	ウィンドウ(W)	ヘルプ(H)						
												^
<st< td=""><td>art E</td><td>SLDC S7</td><td>G2LSTD.</td><td>Sequence</td><td>Sample_LE</td><td>V-JP3:1J></td><td></td><td></td><td></td><td></td><td></td><td></td></st<>	art E	SLDC S7	G2LSTD.	Sequence	Sample_LE	V-JP3:1J>						
511/2 292	(- LUP (f+) E	-F:Soft 2E000	UN:Hai	raj SWB3LUN E0000]	∷Start」S zop+E0000	₩B4L++duty」] du+vEnnn1	200201 200202	ULOUSOJ ALAD PARR	ME21107		U₩] √f+∖ Þ∕	
KSc	oft> F	os[000 os[000	1 00000	rpm[00000]	zcnt[0000] dutv[0001	1 UE20691	V[2062]	W[2054]		OFF1	JSLV
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	002000]	1 Pin200003	LOUICECCOO] 0[2000]	1650053		0011		~

3-3. デバッグ終了

S7G2_e2std_BLDC - BLDC_Drive_STD/src/synergy_gen/main.c - e ² 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) プロジェクト(P) Renesas Views 実行(R) ウィンドウ(W) ヘルプ(H)	
🔦 🚸 🔳 株 デバッグ(B) 🗸 🖻 BLDC_Drive_STD Debug 🗸 🄅 📩 マ 🗟 😨 😽 マ 🗟	Ø 🖉	🗢 🔗 🗢
⊿ ! ₦ : ¤ ■ = = <u>**</u> i+ ≂ ∞ ! ¾ i+ ~ ¶ + ¶ + ¶ + ! □, + * m + == *? ¶ & \$? ¶ &		
1 - 型 - や マ ↓ □ クリックで終了 1 - ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	C/C++	参 デバッグ
※デバッグ×	(x)= ×	» ₆ □ □
✓ 💽 BLDC_Drive_STD Debug [Renesas GDB Hardware Debugging]	<u>ﷺ</u> ⇒ta [3 🖸 🖻
V 况 BLDC_Drive_STD.elf [1] [cores: 0]	8	
Thread #1 1 (single core) [core: 0] (Running)	名前君	J ^
arm-none-eabl-gdb (7.8.2) □ Repeats GDR server (Hott)		
Nellesas duo server (nost)		
(BLDC_Drive_STD) Synergy Configuration Istartup_STG2.c Imain.c X		_
126		_
127 ⊖ int main(void)		_
129		
130 0000abe8 tx_kernel_enter ();		
131 132 0000abec return 0:		
133 0000abee }		
134		
()	<	\rightarrow \bigcirc
🖳 ユンソール 🗙 🔠 レジスター 🕕 デパッグ・シェル 🖹 問題 🙀 Debugger Console 🦓 スマート・ブラウザー 🏮 メモリー		- 8
🗏 X 🔆 🗎 🖬 🖓 🖉	2 🖉 🛃	📃 🕶 📑 🕶
BLDC_Drive_STD Debug [Renesas GDB Hardware Debugging] [pid: 32]		
ターケット接続決定 - OK ガウンロード開始		$\hat{\mathbf{v}}$
		>
実行中 🔶	🗞	

・「BLDC_Drive_HAL」は、上項と同じ操作をして下さい。

- 4. 注意事項
 - ・本文書の著作権は、エーワン(株)が保有します。
 - ・本文書を無断での転載は一切禁止します。
 - ・本文書に記載されている内容についての質問やサポートはお受けすることが出来ません。
 - ・本文章に関して、ルネサス エレクトロニクス社への問い合わせは御遠慮願います。
 - ・本文書の内容に従い、使用した結果、損害が発生しても、弊社では一切の責任は負わないものとします。
 - ・本文書の内容に関して、万全を期して作成しましたが、ご不審な点、誤りなどの点がありましたら弊社までご連絡くだされば幸いです。
 - ・本文書の内容は、予告なしに変更されることがあります。

5. 商標

- ・e2studioは、ルネサス エレクトロニクス株式会社の登録商標、または商品名称です。
- ・Renesas Synergy[™]および S3A7/S5D9/S7G2 は、ルネサス エレクトロニクス株式会社の登録商 標、または商品名です。
- ・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。

6. 参考文献

- ・「S3A7 ユーザーズマニュアル ハードウェア編」 ルネサス エレクトロニクス株式会社
 ・「S5D9 ユーザーズマニュアル ハードウェア編」 ルネサス エレクトロニクス株式会社
 ・「S7G2 ユーザーズマニュアル ハードウェア編」 ルネサス エレクトロニクス株式会社
 ・ルネサス エレクトロニクス株式会社提供のサンプル集
 ・「e2studioユーザーズマニュアル 入門ガイド」 ルネサス エレクトロニクス株式会社
 ・「SSP vx.xx User's Manual」 ルネサス エレクトロニクス株式会社
 ・「X-Ware Component Documents for Renesas Synergy™」 ルネサス エレクトロニクス株式会社
- ・その他

 $\mp 486-0852$

愛知県春日井市下市場町6-9-20 エーワン株式会社 https://www.robin-w.com

