

XMC7200 Development Environment

Infineon Technologies Japan K.K. 2024.5.29





Microcontroller Portfolio



Infineon Microcontroller Portfolio Overview





XMC7000 at a glance

Product features

- Single/ Dual Arm Cortex®-M7 CPU (up to 350-MHz) for primary processing
- Arm Cortex®-M0+ (100-MHz) for peripheral & security processing
- Up to 105x 16-bit & 16x 32-bit timer/counter pulse-width modulator blocks
- Up to 11x reconfigurable Serial Communication Blocks
- Up to Up to 220 Programmable I/Os
- 3x SAR ADC (up to 1Msps) with up to 57 external channels
- Up to 10x CAN FD
- Crypto Engine support
- 5 different low power modes
- Extended voltage operating range: 2.7 to 5.5V
- Extended temperature range: -40 to 125°C

Value proposition

- Optimal solution for motor control
- Cost-optimization & robustness: uses an available 40nm platform
- Scalable solution: 2 series, 17 different package-memory combinations
- ➢ ModusToolbox[™] as software development platform
- Advanced security options
- Power saving for energy-critical applications



Availability & material

- ✓ Samples already available
- ✓ Evaluation board already available
- ✓ Marketing material & Datasheet:

infineon.com/XMC7000





ModusToolbox[™]



- このドキュメントは、XMC7200評価キット (KIT_XMC72_EVK) でポテンショメータ(可変抵抗器)の値に応じ てLEDの輝度を変更するプロジェクトをModusToolbox(開発環境)上で作成する手順を説明するものです。



https://www.infineon.com/cms/jp/product/evaluation-boards/kit_xmc72_evk/

Reference



- XMC7200 Manual

XMC7200 Datasheet 32-bit Arm® Cortex-M7

https://www.infineon.com/dgdl/Infineon-XMC-7200-Datasheet-DataSheet-v01_00-EN.pdf?fileId=8ac78c8c8412f8d30184443173374425

XMC7000 MCU family architecture Technical reference manual https://www.infineon.com/dgdl/Infineon-XMC7000 MCU FAMILY_ARCHITECTURE_TECHNICAL_REFERENCE_MANUAL-UserManual-v03_00-EN.pdf?fileId=8ac78c8c850f4bee01859d6b36155cad

XMC7200 MCU registers Technical reference manual https://www.mouser.com/pdfDocs/Infineon-XMC7200 Technical-reference-manual-UserManual-v01 00-EN.pdf

Application Note

https://www.infineon.com/cms/jp/product/microcontroller/32-bit-industrial-microcontroller-based-on-arm-cortex-m/32-bit-xmc7000-industrial-microcontroller-arm-cortex-m7/#!documents

AN235616 - Debug XMC7000 MCU in ModusToolbox[™] environment <u>https://www.infineon.com/dgdl/Infineon-AN235616_Debug_XMC7000_MCU_in_ModusToolbox_environment-ApplicationNotes-v01_00-</u> <u>EN.pdf?fileId=8ac78c8c8412f8d3018455dc4b2006ef</u>

AN234282 - Using a SAR ADC in XMC7000 family

https://www.infineon.com/dgdl/Infineon-AN234282-Using a SAR ADC in XMC7000 Family-ApplicationNotes-v01_00-EN.pdf?fileId=8ac78c8c8412f8d3018451d1b5930227

AN234119 - Timer, Counter, and PWM (TCPWM) usage in XMC7000 family

https://www.infineon.com/dgdl/Infineon-AN234119 - Timer_Counter_and_PWM_TCPWM_usage_in_XMC7000_family-ApplicationNotes-v03_00-EN.pdf?fileId=8ac78c8c8412f8d301845123daf24f35

AN234226 - XMC7000 MCU: Usage of Interrupts

https://www.infineon.com/dgdl/Infineon-AN234226 - XMC7000 MCU Usage of Interrupts-ApplicationNotes-v03 00-EN.pdf?fileId=8ac78c8c8412f8d3018451e417060290



ModusToolbox[™] overview

ModusToolbox[™]とは?



- > 特徴
 - Eclipseをベースに作成されたInfineonの統合開発環境(IDE)
 - XMCをはじめPSoC[™], AIROC[™]等のInfineon製デバイスをサポート
 - コードサイズや機能等の制限無く無償で利用可能
 - ツールチェーンによる機械学習, Connectivity, セキュリティのサポート
 - GitHubによるサンプルコード,ミドルウェア,デバイスドライバーの提供
 - カスタムBSP(ボードサポートパッケージ)作成のためのBSP Assistant
 - Infineon Developer Communityによるオンライン技術サポート

ModusToolbox

https://www.infineon.com/cms/jp/design-support/tools/sdk/modustoolbox-software/

Infineon Developer Community

https://community.infineon.com/?category.id=jp





Project creation for KIT_XMC72_EVK

Preparation



- > 以下の開発環境をインストールして ください。
 - ModusToolbox 3.2
 <u>https://www.infineon.com/cms/en/design-support/tools/sdk/modustoolbox-software/</u>
- > デザインサポートのPCB設計データ よりKIT_XMC72_EVKの回路図(pdf) をダウンロードしてください。

https://www.infineon.com/cms/jp/product/e valuation-boards/kit_xmc72_evk/

Note: 回路図はマイコンに接続された外部部品 (LED等)の接続ポート番号の確認等で使用します。





- 1. スタートメニューよりEclipse IDE for ModusToolbox 3.2をクリックして ModusToolboxを起動します。
- 2. プロジェクトを格納するワークスペース・フォルダを指定します。 Note: フォルダ名, Path/ニマルチバイト文字, スペースを使用しないでください。
- 3. Launchをクリックします。

Note:End User License Agreementのダイアログが表示された場合はAcceptをクリックしてください。

	ModusToolbox 3.1 (Current user) ^ 新規	Eclipse IDE for ModusToolbox™ 3.1 Launcher	:
	bsp-assistant 1.10.0 新規	Select a directory as workspace	
	bt-configurator 2.80.0 新規	Eclipse IDE for ModusToolbox [™] 3.1 uses the workspace directory to store its preferences and development artifacts.	
	capsense-configurator 6.10.0 新規	2	_
8	capsense-tuner 6.10.0 TUNE 新規	Workspace: C:¥user¥work¥modus¥KIT_XMC72_EVK_WS ~ Browse	
	dashboard 3.1.0 START 新規		
	device-configurator 4.10.0 DEV 新規		
<u>~</u>	dfuh-tool 2.0.0 新規	Use this as the default and do not ask again	
1	Eclipse IDE for ModusToolbox 3.1	<u>R</u> ecent Workspaces	
ch	ez-pd-configurator 1.21.0 新規	3 Launch Cancel	
0	. fw-loader 3 5 0		
Ŧ	₽ ここに入力して検索		

プロジェクトの新規作成(1/3)



1. Quick panelのNew Apprication又は、File → New → ModusToolbox Applicationを 選択します。



プロジェクトの新規作成(2/3)



2. XMC BSPs(Board Support Package)のKIT_XMC72_EVKを選択しNextをクリックします。

s 5	MCU/SOC/SIP	Connectivity	^	The KIT_XMC72_EVK, a 272-pin
s				eraldation board is based on the
				XMC7000 family of devices. XMC7000
PS				MCU is designed for industrial
ging BSPs				carries a XMC7000D microcontroller a
				M.2 interface connector for interfacing
DL7141_FOC_3SH	XMC1404-Q064x0200	<none></none>		radio modules based on AIROC [™] Wi-Fi
)L7141_TRAP_1SH	XMC1404-Q064x0200	<none></none>		and Bluetooth® combos (currently
D700A_FOC_3SH	IMD701A-Q064x128 (XMC1404-Q064x0128)	<none></none>		not supported), SMIF dual header
PLT2GO_XMC4200	XMC4200-F64x256	<none></none>		compatible with Digilent Pmod for
PLT2GO_XMC4400	XMC4400-F100x512	<none></none>		(currently not supported) and headers
11_BOOT_001	XMC1100-T038x0064	<none></none>		compatible with Arduino for
12_BOOT_001	XMC1200-T038x0200	<none></none>		interfacing Arduino shields. In
13_BOOT_001	XMC1302-T038x0200	<none></none>		addition, the board features an on-
14_BOOT_001	XMC1404-Q064x0200	<none></none>		board programmer/debugger
43_RELAX_ECAT_V1	XMC4300-F100x256	<none></none>		(KitProg3), a 512-Mbit QSPI NOR flash,
45_RELAX_V1	XMC4500-F100x1024	<none></none>		CAN FD transceiver, Gigabit Ethernet
47_RELAX_V1	XMC4700-F144x2048	<none></none>		interface a micro-B connector for USB
48 RELAX ECAT V1	XMC4800-F144x2048	<none></none>		device interface, three user LEDs, one
72_EVK	XMC7200D-E272K8384	<none></none>		potentiometer, and two push buttons.
/2_EVK_MUK_43439M2	2 XMC7200D-E272K8384	LBEE5KLTYN (CYW4	43	The board supports operating voltages
IERIC	XMC1404-Q064x0200	<none></none>	\mathbf{v}	from 3.3 V to 5.0 V for XMC7000D
			>	device.
le mannest data (4996	ms)			
, db	*			
	9.119 03-3 DL7141_FOC_3SH DL7141_TRAP_1SH D700A_FOC_3SH PLT2GO_XMC4200 PLT2GO_XMC44000 11_BOOT_001 12_BOOT_001 13_BOOT_001 13_BOOT_001 13_BOOT_001 13_RELAX_ECAT_V1 14_BOOT_001 13_RELAX_V1 15_RELAX_V1	ging 03+3 0L7141_FOC_3SH XMC1404-Q064x0200 0L7141_TRAP_1SH XMC1404-Q064x0200 0700A_FOC_3SH IMD701A-Q064x128 (XMC1404-Q064x0128) PLT2GO_XMC4200 XMC4200-F64x256 PLT2GO_XMC4400 XMC1100-T038x0064 12,800T_001 XMC1100-T038x0200 3_BOOT_001 XMC1404-Q064x0200 13_RELAX_ECAT_V1 XMC4300-F100x256 15_RELAX_V1 XMC4700-F144x2048 12_EVK XMC4700-F144x2048 12_EVK XMC7200D-E272K8384 12_EVK XMC1404-Q064x0200	SING USES DL7141_FOC_3SH XMC1404-Q064x0200 DL7141_TRAP_1SH XMC1404-Q064x0200 D1700A_FOC_3SH IMD701A-Q064x128 (XMC1404-Q064x0128) PLT2GO_XMC4200 XMC4200-F64x256 PLT2GO_XMC4400 XMC4200-F64x256 PLT2GO_XMC4400 XMC1100-T038x0064 11_BOOT_001 XMC1100-T038x0004 2_BOOT_001 XMC1404-Q064x0200 3_BOOT_001 XMC1404-Q064x0200 3_BOOT_001 XMC1404-Q064x0200 3_BOOT_001 XMC1404-Q064x0200	ging USF3 XMC1404-Q064x0200 <none> 0L7141_TRAP_1SH XMC1404-Q064x0200 <none> 02700A_FOC_3SH IMD701A-Q064x128 (XMC1404-Q064x0128) <none> PLT2GO_XMC4200 XMC4200-F64x256 <none> PLT2GO_XMC4200 XMC4400-F100x512 <none> 11_BOOT_001 XMC1100-T038x0064 <none> 12_BOOT_001 XMC1200-T038x0200 <none> 3_BOOT_001 XMC1404-Q064x0200 <none> 13_BOOT_001 XMC1404-Q064x0200 <none> 13_RELAX_ECAT_V1 XMC4500-F100x256 <none> 15_RELAX_V1 XMC4500-F100x1024 <none> 15_RELAX_V1 XMC4500-F104x2048 <none> 12_EVK XMC7000D-E272K8384 <none> 12_EVK_MUR_43439M2 XMC1404-Q064x0200 <none> 12_EVK_MUR_43439M2 XMC1404-Q064x0200 <none></none></none></none></none></none></none></none></none></none></none></none></none></none></none></none>

プロジェクトの新規作成(3/3)



Getting StartedのEmpty Appをチェックします。
 New Application Nameに"ADC_PWM_PJ"を設定します(任意指定可)
 Createをクリックします。

Select Application - Project Creator 2.10	- 🗆 X	
Settings Help Application(s) Root Path: C:/user/work/modus/KIT_XMC72_EVK_WS Target IDE: Eclipse IDE for ModusToolbox™ Enter filter text	Browse This is a minimal starter application template for Infineon MCU devices. For more details, see the <u>README on</u> <u>GitHub</u> .	Note: Peripherals以下/こXMC7200/こ搭 載されたペリフェラル用のテンプレー ト(サンプル)が用意されており、各ペリ フェラルの使用方法を参照する事が出来 ます。 また、WebサイトよりXMC7000シリーズ のアプリケーションノートを参照できます。 https://www.infineon.com/cms/jp/produ ct/microcontroller/32-bit-industrial- microcontroller/32-bit-industrial-
Application(s): Empty App Application(s) Root Path: C:/user/work/modus/KIT_XMC72_EVK_WS Press "Create" to create the selected application(s).	<pre></pre>	<u>m/32-bit-xmc7000-industrial-</u> <u>microcontroller-arm-cortex-m7/</u>

2024-05-29

ModusToolBoxパースペクティブ



> プロジェクトを生成するとModusToolboxパースペクティブ(画面レイアウト)が 表示されます。





Configuring with Device Configurator

Device Configuratorの起動



- > プロジェクトには以下の機能を実装します。
 - ポテンショメータ(可変抵抗器)の値を12bit SAR ADC(A/Dコンバータ)で取得
 - 12bit SAR ADCの値に応じてLEDに接続したTCPWM(PWM)の値を設定
- > 12bit SAR ADCやTCPWMといったペリフェラルや、ペリフェラルを動作させる クロック、LEDに接続するGPIO(I/O)の設定はModusToolboxのDevice Configuratorを使用して行います。



- 1. Project ExplorerでADC_PWM_PJプロ ジェクトをクリックして選択します。
- 2. Quick PanelでDevice Configurator をク リックします。

Note: Project ExplorerでADC_PWM_PJプロジェクト を選択することでQuick Panel/こDevice Configurator が現れます。

12bit SAR ADCの設定(1/5)





1. Device Configurator \mathcal{O} Peripheralsタブをクリックします。 2. 12bit SAR ADC 0をチェックします。 3. Name(s)に"SARADC 0"を設定し ます。 Note: Name(s)に設定した名前は、ソース コード(main.c) からライブラリ(PDL/HAL) を使 用してアクセスする際に、関数の引数として使 用されます(SARADC 0 HW, SARADC 0 config等) 4. Enable SAR Blockをチェックしま す。 5. Clockに"8 bit Divider 0 clk"を設 定します。

12bit SAR ADCの設定(2/5)



rie rait settings view	<u>H</u> elp					
XMC7200D-E272K8384				8 bit Divider 0 - Pa	rameters	8 ×
Peripherals Pins Ana	log-Routing System	Peripheral-Clocks	DMA	Enter filter text		🖉 🗉 🖻
Enter filter text		🖉 🔻 E	🖽 🖌 🗎 🖡	Name	Value	
Resource	Name(s)	Personality		∧ ∨ Overview		
 Peri Clock Group 0 				(?) Config	uration Help Open Peripherals Clock [Dividers Documentation
✓ 8 bit		-		✓ General		
8 bit Divider 0	peri_0_group_0_div_8_0	_		(?) Source	Clock 💾 CLK_HF2 (196 MHz ±	: 4%)
8 bit Divider 1	peri_0_group_0_div_8_1			2 Divider	r [16	
8 bit Divider 2	peri_0_group_0_div_8_2			(?) Freque	ncy 📋 12.25 MHz ± 4%	
🗹 8 bit Divider 3	CYBSP_TRACE_CLK_DIV	Peripheral Clock-1.0		(?) Start o	n Reset 🗹	
> 16 bit				Periph	erals 🖉 🔵 12-bit SAR AD	C 0 clock_sar (SARADC_0) [USED]
> 24.5 bit						
 Peri Clock Group 1 					(9)	
	DEPL CLK 1.0	Davinhaval Clask 1.0				
7	PERICER_1_0	Peripheral Clock-1.0	·			
B bit Divider 1	peri_0_group_1_div_8_1	-				
B bit Divider 2	peri_0_group_1_div_8_2	-				
8 bit Divider 3	peri_0_group_1_div_8_3	-				
8 bit Divider 4	peri_0_group_1_div_8_4	-				
8 bit Divider 5	peri_0_group_1_div_8_5	-				
8 bit Divider 6	peri_0_group_1_div_8_6	-				
8 bit Divider 7	peri_0_group_1_div_8_7			8 bit Divider 0 - P	arameters Code Preview	
Notice List						6 ×
😢 0 Errors 🚺 0 Warnin	gs 🧾 2 Tasks 👔 2	Infos				
Fix Description				Locatio	on	
The file was last saved	with a different version of	f the tools than will be	used to perform	ode generation		
on save. Last saved w	ith: 'Configurator Backeng	3.0.0' from 'ModusToc	bolbox 3.0.0'. Curre	nt: 'Configurator design	modus	



- Peri Clock Group 1の8 bit Divider 0をチェックし、Name(s) に"PERI_CLK_1_0"を設定しま す。
- 8. Divider/c"16"を設定します。 Note:Divider(分周比)/c16を設定する事で 12bit SAR ADC 0 の動作クロックが 12.25MHz/c設定されます。

12bit SAR ADCの設定(3/5)



12-bit SAR ADC 0 (SARADC_0) - Parameters - Device Configurator 4.10 ×								
Enter f	ilter text	🖻 ا 🖸	Ŧ					
Name		Value						
	? Clock	8 bit Divider 0 clk (PERI_CLK_1_0) [USED]	~					
	? Clock Frequency	🖱 12.25 MHz						
✓ Ch	annel 0		_					
	? Channel Name	channel_0						
	? Enabled							
	? HW Trigger	Disabled	~					
	? Trigger Input	<unassigned></unassigned>	~					
	? Trigger Output	<unassigned></unassigned>						
	? Trigger Chanel Input	<unassigned></unassigned>						
	? Voltage Range Trigger Output	<unassigned></unassigned>	•					
? Channel Done Trigger Output		<unassigned></unassigned>						
	⑦ Debug Freeze Input	<unassigned></unassigned>						
	? Priority	0						
(11)	? Preemption Type	Abort ongoing acquisition, do not return	~					
W	⑦ Group End							
	Output Trigger Type	Pulse	~					
12	Input		.]					
	? Enable External Analog Mux							
	Precondition Mode	No Preconditioning	~ ~					

10. Channel 0のEnabledをチェックします。 11. Group Endをチェックします。 12. Inputを"P6[6]"に設定します。



12bit SAR ADCの設定(4/5)



12	2-bit SAR ADC 0 (SARADC_0) - Paramete	rs - Device Configurator 4.10	x			
En	nter filter text	🖉 U 🖻	•			
Na	ame	Value	^			
	⑦ Clock	8 bit Divider 0 clk (PERI_CLK_1_0) [USED]	\sim			
	⑦ Clock Frequency	12.25 MHz				
~	Channel 0		_			
	⑦ Channel Name	channel_0				
	? Enabled					
	? HW Trigger	Disabled	~			
	? Trigger Input	<unassigned></unassigned>	~			
	? Trigger Output	<unassigned></unassigned>				
	? Trigger Chanel Input	<unassigned></unassigned>	~			
	? Voltage Range Trigger Output	<unassigned></unassigned>				
	? Channel Done Trigger Output	<unassigned></unassigned>				
	⑦ Debug Freeze Input	<unassigned></unassigned>	~			
	Priority	0				
	Preemption Type	Abort ongoing acquisition, do not return	\sim			
	⑦ Group End	\checkmark				
	Output Trigger Type	Pulse	\sim			
	Input	P6[6] analog (CYBSP_POT) [USED]				
	② Enable External Analog Mux					
	? Precondition Mode	No Preconditioning	~ ~			

13. P6[6]の設定を行うため 2. P6[6]の設定を行うため 2. P6[6]の設定を行うため 2. P6[6]の設定を行うため 2. P5(6)の 2. P5(7)の 2. P5(7)(7) 2. P5(7)(7)

	P6[6] (CYBSP_PO1) - Parameters - Device Configurator 4.10						
	Ent	ter filter	text	U	₽ ₽		
	Na	me		Value			
	~	Overvie	w				
(5	?	Configuration Help	Open GPIO Documentation			
14	1)	General					
		(?)	Drive Mode	Analog High-Z. Input buffer off	~		
		(?)	Initial Drive State	High (1)	~		
	~	Input		C1405			
		0	Ihreshold	CMOS	~		
		0	Interrupt Irigger lype	None	~		
	Ť	() ()	Slow Pata	East			
			Siew Rate	1/2			
	~	Internal	Connection -	172	~		
	Ť	?	Analog 15	8 12-bit SAR ADC 0 vin[0] (SARADC_0) [USED]			
		?	Digital Input	<unassigned></unassigned>			
		?	Digital Output	<unassigned></unassigned>	~		
		?	Digital InOut	<unassigned></unassigned>	~		
	~	Advanc	ed				
		?	Store Config in Flash				

12bit SAR ADCの設定(5/5)



12-bi	it SAR	ADC 0 (SARADC_0) - Parameter	s - Device Configurator 4.10	x
Enter	filter	text	📃 U 📄	(†
Name	2		Value	^
	?	Input	P6[6] analog (CYBSP_POT) [USED]	
	?	Enable External Analog Mux		
	?	Precondition Mode	No Preconditioning ~	·
10	?	Overlap Diagnostic Mode	No Overlap or SARMUX Diagnostics	·
	?	Sample Time (Aperture)	8	
	?	Selection Of Calibration Values	Regular ~	
	?	Result Data Alignment	The data is right aligned in Result[11:0]	-
	?	Sign Extension	Unsigned ~	r
(17)	?	Post Processing Mode	Averaging ~	
	?	Averaging Count	1	J
	?	Shift Right	0	
	?	Positive Reload	<u></u> 0	
	?	Negative Reload	<u></u>	
	?	Range Detection Mode	Below Low Threshold (Result < Low)	,
	?	Range Detect Low Threshold	0	
	?	Range Detect High Threshold	65535	
✓ A	dvanc	ed		
	?	Store Config in Flash	\checkmark	¥

16. Sample Timeを"8"に設定します。 17. Post Processing Modeを"Averaging"に 設定し、Averaging Countを"1"に設定しま す。

TCPWMの設定(1/3)



 XMC7000にはTCPWMのChannelが多数搭載されており、出力に接続可能なGPIOは Channel毎に異なるため、LEDが接続されているGPIOの設定からTCPWMのChannel を特定します。



1. Device Configurator のPinsタブを
2. LED1が接続されているP16[1]を チェックレます
アエツクしま9。 VTARG_REF User LEDs
R103 LED1 ORANGE R104 IK WP16_1
3. Drive Modeを"Strong Drive, Input
buffer off"に設定します。
4. Digital Outputを"TCPWM[1]
Group[0] Counter 61"に設定します。
5. TCPWMの設定に移るため 🖉 をク
リックします。

TCPWMの設定(2/3)



C:/user/work/modus/KIT_XMC72_EVK_W	/S/ADC_PWM_PJ	/bsps/TARGET_AP	P_KIT_XN	1C72_EVK/config/des	ign.modus* - D	evice Configurator 4.1	o —		×
<u>F</u> ile <u>E</u> dit <u>S</u> ettings <u>V</u> iew <u>H</u> elp									
XMC7200D-E272K8384					TCPWM[1] G	roup[0] 16-bit Counter	61 (PWM_1_0_61) - Parameters		đΧ
Peripherals Pins Analog-Routing	System Per	ripheral-Clocks	DMA		Enter filter te	xt		2 0	⊟ ₽
Enter filter text		R	7 6	• * •	Name		Value		^
Resource		Name(s)	_	Personality ^	✓ Overview				
	16-bit Counter 61	PWM 1 0 61		PWM-1.0 ~	? C	onfiguration Help	Open PWM (TCPWM) Docume	entation	
	it Counter 62	tcpwm 1 aroup	0 cnt 62		✓ General		2		_
TCPWM[1] Group[0] 16-b	it Counter 63	tcpwm 1 group	0 cnt 63		(?) T	CPWM Version	TCPWM_ver2		
TCPWM[1] Group[0] 16-b	it Counter 64	tcpwm 1 group	0 cnt 64		(?) P	WM Mode	PWM		
TCPWM[1] Group[0] 16-b	it Counter 65	tcpwm 1 group	0 cnt 65		(?) C	lock Prescaler	Divide by 1		
TCPWM[1] Group[0] 16-b	it Counter 66	tcpwm 1 group	0 cnt 66		(?) P	WM Resolution	🖰 16-bits		
TCPWM[1] Group[0] 16-b	it Counter 67	tcpwm_1_group	0 cnt 67	7	? P	WM Alignment	Left Aligned		
TCPWM[1] Group[0] 16-b	it Counter 68	tcpwm_1_group	0 cnt 68		? R	un Mode	Continuous		
	it Counter 69	tcpwm_1_group	0_cnt_69		? In	nmediate Kill			
	it Counter 70	tcpwm_1_group	0_cnt_70		✓ Period		_		_
	it Counter 70	tcpwm_1_group	0_cnt_70		7	nable Period Swap			_
	it Counter 71	tepwin_1_group	_0_cnt_71		(? P	eriod	65535		
	it Counter 72	tcpwm_1_group	_0_cnt_72		✓ Compare				
	it Counter 73	tcpwm_1_group	_0_cnt_/3			nable Compare 0 Swap			_
CPWM[1] Group[0] 16-b	it Counter 74	tcpwm_1_group	_0_cnt_74		\circ	ompare 0	0		
TCPWM[1] Group[0] 16-b	it Counter 75	tcpwm_1_group	_0_cnt_75			nable Compare 1 Swap			_
TCPWM[1] Group[0] 16-b	it Counter 76	tcpwm_1_group	_0_cnt_76		9) 🕐 🗘	ompare 1	0		
TCPWM[1] Group[0] 16-b	it Counter 77	tcpwm_1_group	_0_cnt_77		Interrupt	Source			×
TCPWM[1] Group[0] 16-b	it Counter 78	tcpwm_1_group	_0_cnt_78		<				>
			0 1 70	· · · ·	TCPWM[1]	Group[0] 16-bit Count	er 61 (PWM_1_0_61) - Parame	Code	Prev
Notice List									8 ×
😢 0 Errors 👍 0 Warnings 闫 1 Tasl	k 🚺 1 Info								
Fix Description				Locatio	n		~		^
The 'Clock Signal' parameter must no	ot be empty.			XMC720	00D-E272K8384:	TCPWM[1] Group[0] 1	6-bit Counter 61 (PWM_1_0_61)	[Clock Si	ignal]
The WCO is enabled. Chip startup wil	Il be slower beca	use clock configu	ration car	not					
continue until the WCO is ready. See Ready	the device datas	heet for WCO star	tun timin	a If WCO is XMC720	00D-F272K8384	WCO			×

 TCPWM[1] Group[0] 16bit Counter 61をチェックし、 Name(s)に"PWM_1_0_61" を設定します。Personality は"PWM-1.0"に設定します。

- 7. Periodに"65535"を設定し ます。
- 8. Compare 0に"0"を設定しま す。
- 9. Compare 1に"0"を設定しま す。





Enter filter text	U U 🖻 🖻
Name	Value
⑦ Compare 1	0
 Interrupt Source 	
Overflow & Under	rflow
⑦ Compare 0	
(?) Compare 1	
✓ Inputs	_
Clock Signal	<unassigned></unassigned>
② Count Input	<up></up>
 Kill 0 loput 	10 8 bit Divider 0 clk (PERI_CLK_1_0) [USED]
	8 bit Divider 1 clk
? Reload Input	8 bit Divider 2 clk
(?) Start Input	8 bit Divider 5 clk
③ Swap Input	8 bit Divider 5 clk
? Kill 1 Input	8 bit Divider 6 clk
 PWM Output Polarity 	8 bit Divider 7 clk
Invert PWM Outp	ut 🔮 8 bit Divider 8 clk
(?) Invert PWM n Ou	tput
✓ PWM Disabled Output	
PWM Disabled Or	utput High Impedance
✓ Outputs	
PWM (line)	P16[1] digital_out (CYBSP_USER_LED, CYBSP_USE
? PWM_n (line_con	npl) <unassigned></unassigned>
 Trigger Outputs 	
? Trigger 0 Event	Disabled
(?) Trigger 1 Event	Disabled
✓ Advanced	
(?) Store Config in Fl.	ash 🗸
<u> </u>	_

10. Clock Signal に"8 bit Divider 0 clk"を設定します。 Note: 8 bit Divider 0 clk/よSARADCと共用になり、既にDivider(分 周比)の設定は完了しているため、Dividerの設定は必要ありません。 他のClockを設定する場合はSARADCと同様にDividerを設定します。 11. File -> Saveを選択して設定情報を保存します。 12. Device Configurator のウインドウを閉じて ModusToolboxに戻ります。

	File	C:/user/work/modus/KIT_XMC72_EVK_WS/ADC_PWM_PJ/bsps/TARGET_APP_KIT_XMC72_EVK/config/design.modus - Device Configurator 4.10 Edit Settings View Help		- <u>12</u> 🗵)
(11)	2	New Open Close	Ctrl+N Ctrl+O Ctrl+W	ۍ ۲۵ ۲۵ ت 🔎	Ē
		Save As	Ctrl+S		
		Open in System Explorer Change Libraries		~	

Device Configuratorで設定されたデータの格納場所



 > Device Configurator で設定されたデータはC言語コードで以下の場所に格納されます。 <sup>¥ADC_PWM_PJ¥bsps¥TARGET_APP_KIT_XMC72_EVK¥config¥GeneratedSource¥cycfg_peripherals.[c|h]
 > また、Code Previewタブをクリックする事でコードを確認する事が出来ます。
 > ユーザーがCodingする際には、このファイルで定義されたデータを使用します。
</sup>

C:/user/work/modus/KIT_XMC72_EVK_WS/ADC_PWM_PJ/bsps/TARGET_A	APP_KIT_XMC72_EVK/config/design.modus - Device Configurator 4.10 –	×
<u>F</u> ile <u>E</u> dit <u>S</u> ettings <u>V</u> iew <u>H</u> elp		
XMC7200D-E272K8384	Code Preview	e ×
Peripherals Pins Analog-Routing System Peripheral-Cl	Enter search text	0,
Enter filter text 🖉 🏹 🖻 🖽	<pre>#include "cy_sysclk.h" #include "cycfg routing.h"</pre>	^
Resource Name(s) ^	<pre>#define SARADC_0_HW PASS0_SAR0 #define SARADC_0_CH0_HW PASS0_SAR0_CH0 #define SARADC_0_chonnel_0_HW_SARADC_0_CH0_HW</pre>	
✓ 12-bit SAR ADC 0 SARADC_0	<pre>#define SARADC_0_CH0_IRQ pass_0_interrupts_sar_0_IRQn</pre>	
12-bit SAR ADC 1 pass_0_saradc_	#define SARADC_0_channel_0_IRQ_SARADC_0_CH0_IRQ	
12-bit SAR ADC 2 pass_0_saradc_	#ifndef SARMUX0 CHO PORT ADDR	
pass_0_aref_0	<pre>#define SARMUX0_CH0_PORT_ADDR 0</pre>	
✓ Communication	#endif #define SNRNDC 0 charges 0 IDV (000)	
> Controller Area Network FD (CAN FD) 0	#deline SARADC_0_channel_0_lDX (00L)	
Controller Area Network FD (CAN FD) 1	<pre>const cy_stc_sar2_channel_config_t SARADC_0_channel_0_config =</pre>	
Inter-IC Sound Bus (I2S) 0 audioss_0_i2s_	{	
Inter-IC Sound Bus (I2S) 1 audioss_1_i2s_	.channelHwEnable = true, triggerSelection = CV_SAR2_TRIGGER_OFF	
Inter-IC Sound Bus (I2S) 2 audioss_2_i2s_	.channelPriority = 0U,	
Quad Serial Memory Interface (QSPI) 0 smif_0	STOORDETING - CV CAD2 DEFEMOTION ABORT CANCET	> [×]
< >	12-bit SAR ADC 0 (SARADC_0) - Parameters Code Preview	-
Notice List		₽×
😢 0 Errors 🔥 0 Warnings 📔 0 Tasks 🚺 1 Info		
Fix Description	Location	
The WCO is enabled. Chip startup will be slower because clock config datasheet for WCO startup timing. If WCO is not required during start	guration cannot continue until the WCO is ready. See the device XMC7200D-E272K838 tup, consider starting it in main() for faster chip startup.	14: WCO
Ready		

Note: cycfg_peripherals.[c|h]に定義されたコードと 同様のコードを自分でCodingする事で、Device Configuratorを使用せずにプログラムを作成する事 も可能です。 しかし、定義漏れなどが発生する事を避けるため、

Device Configuratorを使用する事を推奨します。



Adding libraries using the Library Manager

ライブラリ(retarget-io)の追加(1/2)



Debug用にシリアルコンソール上への出力を行うため、Library Managerを使用してADC_PWM_PJプロジェクトにretarget-ioライブラリを追加します。



1. Project Explorerで ADC_PWM_PJプロジェクトを クリックして選択します。

 Quick PanelでLibrary Managerをクリックします。

ライブラリ(retarget-io)の追加(2/2)



Settings Help Application Directory: C/user/work/modus/KIT_XMC72_EVK_WS/ADC_PWM_PJ Enter filter text	wse
Application Directory: C/user/work/modus/KIT_XMC72_EVK_WS/ADC_PWM_PJ Enter filter text	wse
Application Directory: C/user/work/modus/KIT_XMC72_EVK_WS/ADC_PWM_PJ Enter filter text Name Update Available Remove * BSPs • APP_KIT_XMC72_EVK (ACTIVE) * ADC_PWM_PJ Libraries catIcm0p cmsis core-lib core-make mtb-hal-cat1 mtb-hal-cat1 mtb-hal-cat1 Add Library 3 Statting to refresh dependencies for project 'inth-example-empty-app'	owse
Enter filter text Name Update Available Remove BSPs ADC_PVM_PJ Libraries cat1cm0p cmsis core-lib core-make mtb-hal-cat1 mtb-pdl-cat1 mtb-pdl-cat1 Cat1 Cat1 Cat1 Cat1 Cat1 Cat1 Cat1 C	
Name Update Available V BSPs	
 BSPs APP_KIT_XMC72_EVK (ACTIVE) ADC_PWM_PJ Libraries cat1cm0p cmsis core-lib core-make mtb-hal-cat1 mtb-pdl-cat1 recipe-make-cat1c Add BSP Add Library 3 	
APP_KIT_XMC72_EVK (ACTIVE) ADC_PWM_PJ Libraries cat1cm0p cmsis core-lib core-make mtb-hal-cat1 mtb-pdl-cat1 recipe-make-cat1c Add BSP Add Library 3 Starting to refresh dependencies for project 'mth-example-empty-app'	
ADC_PWM_PJ Libraries cat1cm0p cmsis core-lib core-make mtb-hal-cat1 mtb-pdl-cat1 recipe-make-cat1c	
cat1cm0p Consist Consi	
cmsis Core-lib Core-make C	
core-lib core-make mtb-hal-cat1 mtb-pdl-cat1 recipe-make-cat1c Add BSP Add Library 3 Starting to refresh dependencies for project 'mth-example-empty-app'	
core-make mtb-hal-cat1 mtb-pdl-cat1 recipe-make-cat1c Add BSP Add Library 3 Starting to refresh dependencies for project 'mtb-example-empty-app'	
mtb-hal-cat1 Image: Constraint of the sympleter of the sympletero of the sympleter of the sympletero of the sympleter of the symp	
mtb-pdl-cat1 Image: Constraint of the second seco	
Add BSP Add Library 3	
Add BSP Add Library 3	
Starting to refresh dependencies for project 'mth-example-empty-app'	
blanding to remeshalependencies for project mas example empty app i	
Refreshed all dependencies for project 'mtb-example-empty-app'.	^
0 error(s), 0 warning(s)	
Successionly acquired the information.	
Update	8
	8 v



- 3. Add Libraryをクリックします。
- 4. 検索窓に"retarget"を設定します。
- 5. Peripheralに表示されるretarget-ioを チェックします。
- 6. OKをクリックします。
- 7. Updateをクリックします。
- ライブラリの追加が完了すると以下の メッセージが表示されるので、Close をクリックしてModusToolboxに戻り ます。

Summary: Successfully updated application "ADC_PWM_PJ" Successfully refreshed dependencies. Done.



Coding (main.c)

main.cを表示





 Project Explorerで
 ADC_PWM_PJプロジェクトの main.cをダブルクリックして main.cファイルを開きます。

main.cのCoding



- 2. main.cに以下の関数をCodingします。
 - main()関数:MCU, ペリフェラルの初期化、割り込み設定、12bit SAR ADCソフトウェアトリガ発行
 - saradcIntHandler():割り込み要因の確認、AD変換値のコンソール出力・TCPWMへの設定、ADCソフトウェアトリガ再発行



main.c(1/4)



* File Name: main.c * Measure potentiometer with SARADC and set the measured value to PWM to change LED brightness * Header Files ****** #include "cy pdl.h" #include "cyhal.h" #include "cybsp.h" #include "cy retarget io.h" * Macros * Global Variables /* SARADC interrupt configuration structure */ cy stc sysint t SARADC0 IRQ cfg = { /* Bit 0-15 of intrSrc is used to store system interrupt value and bit 16-31 to store CPU IRO value */ .intrSrc = ((NvicMux2 IRQn << 16) | pass 0 interrupts sar 0 IRQn), /* Interrupt priority */ .intrPriority = 7};

- cy_pdl.hはPDL, cyhal.hはHALのIncludeファイル です(詳細は後述)
- > Cybsp.hはBSP(ボードサポートパッケージ)に関 連したIncludeファイルです。
- Cy_retarget_io.hはRetarget-ioライブラリをア クセスするために指定します。
- SARADC0_IRQ_cfgはCy_SysInt_Init()関数の引 数として使用されます。
- intrSrcにシステム割り込み元、CPU IRQを指定し、
 intrPriorityにCPU IRQ優先度を設定します。

main.c(2/4)



```
* Interrupt routines for SARADC
******
void saradcIntHandler(void) {
uint32 t val;
   if( Cy SAR2 Channel GetInterruptStatus(SARADC 0 HW, SARADC 0 channel 0 IDX ) ==
CY SAR2 INT GRP DONE){
       // Clear interrupt
       Cy SAR2 Channel ClearInterrupt(SARADC_0_HW, SARADC_0_channel_0_IDX,
CY SAR2 INT GRP DONE);
       val = (uint32_t)Cy_SAR2_Channel_GetResult( SARADC_0_HW, SARADC_0_channel_0_IDX,
NULL );
       printf("SARADC Value:%d¥r", (int)val );
       // Set the measured value to PWM
       Cy_TCPWM_PWM_SetCompare0Val( PWM_1_0_61_HW, PWM_1_0_61_NUM, val<<4);</pre>
       // Start next AD-Conversion
       Cy_SAR2_Channel_SoftwareTrigger(SARADC_0_HW, SARADC_0_channel_0_IDX);
    }else{
       CY_ASSERT(false);
                            // Unexpected interrupt
```

- > ステータスレジスタを参照して割り込み要因が AD変換完了であるかを判定し、割り込み要因の クリアを行います。
- AD変換値(0~4095)を読み出し、コンソールに
 出力します。
- AD変換値(0~4095)を、4bit 左シフトしてPWM
 のCompare値(0~65535)に設定します。
- > ソフトウェアトリガを発行して次のAD変換をスタ ートします。

main.c(3/4)



```
* main
int main(void){
cy rslt t result;
   /* Initialize the device and board peripherals */
   result = cybsp init();
   /* Board init failed. Stop program execution */
   if (result != CY RSLT SUCCESS){
      CY ASSERT(0);
   /* Enable global interrupts */
   __enable_irq();
   /* Initialize retarget-io to use the debug UART port */
   result = cy retarget_io_init(CYBSP_DEBUG_UART_TX, CYBSP_DEBUG_UART_RX,
                           CY RETARGET IO BAUDRATE);
   /* retarget-io init failed. Stop program execution */
   if (result != CY RSLT SUCCESS){
      CY ASSERT(0);
   /* Print message */
   /* ¥x1b[2J¥x1b[;H - ANSI ESC sequence for clear screen */
   printf("¥x1b[2J¥x1b[;H");
   printf("-----
                                           -----¥r¥n");
   printf(" Measure potentiometer with SARADC ¥r¥n");
   printf(" and set the measured value to PWM to change LED brightness¥r¥n");
   printf("------¥r¥n¥n");
```

- 最初にcybsp_init()関数を呼び出すことで
 Deviceの初期化を行います。
- > __enable_irq()関数によりマイクロコントロー ラの割り込みを有効化します。
- > cy_retarget_io_init()関数により標準入
 出力をシリアルポートにマップします。
- > シリアルターミナルを以下のように設定する事でprintf()関数などを使用して出力する事が可能です(Port番号は環境により異なるため省略します)

Baud rate: 115200 Data: 8bit parity: None Stop bit: 1 Flow Control: none main.c(4/4)



/* TCPWM Init */ Cy_TCPWM_PWM_Init(PWM_1_0_61_HW, PWM_1_0_61_NUM, &PWM_1_0_61_config); PWM 1 0_61_HW, PWM_1_0_61_NUM); Cy TCPWM PWM Enable(Cy TCPWM TriggerStart Single(PWM 1 0 61 HW, PWM 1 0 61 NUM); /* SARADC Init */ SARADC 0 HW, &SARADC 0 config); Cy SAR2 Init(Cy_SAR2_Enable(SARADC 0 HW); SARADC 0 HW, SARADC 0 channel 0 IDX, Cy_SAR2_Channel_Init(&SARADC 0 channel 0 config); Cy SAR2 Channel_Enable(SARADC_0_HW, SARADC_0_channel_0_IDX); /* Clear whole interrupt flags */ Cy_SAR2_Channel_ClearInterrupt(SARADC 0_HW, SARADC_0_channel_0_IDX, CY_SAR2_INTR); /* Masking interrupts other than CY SAR2 INT GRP DONE */ Cy SAR2 Channel SetInterruptMask(SARADC_0_HW, SARADC_0_channel_0_IDX, CY SAR2 INT GRP DONE); /* Set Interrupt */ Cy_SysInt_Init(&SARADC0_IRQ_cfg, saradcIntHandler); NVIC ClearPendingIRQ((IRQn Type)(SARADC0 IRQ cfg.intrSrc >> 16)); (IRQn_Type)(SARADC0_IRQ_cfg.intrSrc >> 16)); NVIC EnableIRQ(Cy SAR2 Channel ClearInterrupt(SARADC 0 HW, SARADC 0 channel 0 IDX, CY SAR2 INT GRP DONE); Cy_SAR2_Channel_SoftwareTrigger(SARADC 0_HW, SARADC_0_channel_0_IDX); for (;;){} /* [] END OF FILE */

- TCPWMの初期化を行います。
 Note: 引数に指定されているパラメータは cycfg_peripherals.[c|h]で定義されています。
- > SARADCの初期化を行います。
- > 割り込み要因のクリアを行います。
- > AD変換完了割り込み以外をMaskします。
- > 割り込みハンドラの登録とNVICへの設定を行います。
- ソフトウェアトリガを発行し、AD変換を開始します。

PDL/HAL Reference格納場所について



- Coding を行う際にペリフェラルを制御するライブラリとして、PDL(Peripheral driver library)とHAL(Hardware Abstraction Layer)の2種類が提供されます。
- > PDLはハードウェアを使いやすい API のセットに抽象化したライブラリで、HALはPDLを 使用して構築された更に抽象度の高いライブラリとなります。 Note:PDLとHALは共存が可能です。
- Codingで使用するPDL/HAL APIのReferenceはHTMLファイルとして以下に格納されて おりWebブラウザで参照可能です。

PDL: ¥KIT_XMC72_EVK_WS¥mtb_shared¥mtb-pdl-cat1¥release-v3.8.0¥docs¥pdl_api_reference_manual.html HAL: ¥KIT_XMC72_EVK_WS¥mtb_shared¥mtb-hal-cat1¥release-v2.5.4¥docs¥api_reference_manual.html

Infineon MTB CAT1 Peripheral driver library				
Getting started PDL API Reference	Online Documentation • Device Catalog • Q: Search			
EVTGEN (Event Generator) Flash (Flash System Routine) GPIO (General Purpose Input Out)	SAR2 (Analog to Digital Converter (ADC))			
⊢ I2S (Inter-IC Sound)				
 IPC (Inter Process Communication 	General Description			
 ReyScan (ReyScan) 				
LIN (Local Interconnect Network) LPComp (Low Power Comparator)	LPCom (Local Interconnect Network) The SAR2 driver provides an API to configure the SAR2 ADC.			
⊢ LVD (Low-Voltage-Detect)	The SAP2 has up to 4 ADCs and up to 32 channels for each ADC module. About actual number of supported modules and channels, refer to			
► LVD-HT (Low-Voltage-Detect for H	The technical reference manual (TBM) and device distance in the technical reference in adaptive models and enamels, refer to			
⊢ MCWDT (Multi-Counter Watchdog				
 PDM_PCM (PDM-PCM Converter) 				
► PDM_PCM_v2 (PDM-PCM Conve	Configuration Considerations			
⊢ PRA (Protected Register Access)				
⊢ Profile (Energy Profiler)	▷ Profile (Energy Profiler) Typical usage:			
⊢ Prot (Protection Unit)				
⊢ RTC (Real-Time Clock)	Call Cy_SAR2_Init to initialize the ADC module and its channels			
 SAR (SAR ADC Subsystem) SAR2 (Analog to Digital Converter 	 Call Cy_SAR2_Channel_SetInterruptMask if you need to use the interrupt. After initializing channel(s) call, trigger a call by the software (calling Cy_SAR2_Channel_SoftwareTrigger) or start a peripheral configured for the HW trigger. 			



Build & Program



ProjectのBuildを行った後、Debuggerを起動してBuildしたProjectを KIT_XMC72_EVKに書き込んでDebugを行います。



KIT_XMC72_EVKの準備



- 1. ModusToolboxを動作させているPCとKIT_XMC72_EVKをKit付属のUSB Type-A Mirco-Bケーブルで接続します。
- 2. 付属の12V/3A DC電源アダプタを接続します。



Note: USBからの電力供給は500mA までとなり、電力が不足する場合がある ため、DCアダプタは必ず接続してださい。 Debug(1/12)



Debugアイコン 参の横の▼をクリックしてDebug Configurationを選択します。 GDB OpenOCD DebuggingのADC_PWM_PJ Debugをクリックします。 Debug Configurationsに表示が変化するのでDebugをクリックします。



Note: 2度目以降はDebugアイコン参をクリックする事で Debuggerが起動します。

Pebug Configurations		Pebug Configurations			-	-
Create, manage, and run configurations	T.	Create, manage, and run config	urations			Ť
 Configure launch settings from this dialog: Press the 'New Configuration' button to create a configuration of the selected type. GDB Hardware Debugging Press the 'New Prototype' button to create a launch configuration prototype of the selected configuration. Press the 'Second the Selected configuration. ADC_PWM_PJ Debugging GDB CDD Debugging GDB SEGGER J-Link Debug Press the 'Filter' button to configuration by selecting it. Select launch configuration(s) and then select 'Link Prototype' menu item to unlink - Select launch configuration(s) and then select 'Repe Values' menu item to reset with pro 	lected type. stotype. a prototype. totype values.	Image: Second Secon	Name: ADC_PWM_PJ Debug (KitPr Main Debugger Sta Project: ADC_PWM_PJ C/C++ Application: \$(cy_prj_path:4d7871eb-caad-47e Build (if required) before launchi Build Configuration: Select Aut O Enable auto build © Use workspace settings	og3_MiniProg4) rtup i Source 4-82ad-fd4561284 Variables ing tomatically O Dise <u>Config</u>	Common R S	VD Path Browse XMC72_EVK/Det Browse
Configure launch perspective settings from the <u>'Perspectives'</u> preference page.		Filter matched 9 of 13 items			Revert	Apply
(?) Debug	Close	?		(3)	Debug	Close

Debug(2/12)



- > Debugパースペクティブへの切り替えを行います
- ModusToolboxパースペクティブのままDebugを行う事も可能ですが、Debugger
 使用時に適したDebugパースペクティブへの切り替えを行います。
 - 4. PerspectiveのOpen Perspectiveアイコンをクリックします。
 - 5. Open Perspectiveダイアログが表示されますのでDebugを選択してOpen をクリックします。



Debug(3/12)



> Debug Perspectiveは以下のようになります。





アイコン	名称	説明
Ø	Skip All Breakpoints	ブレークポイントを無効化します。ブレークポイントの削除は行いません。 クリックする度にEnableとDisableをトグルします。
	Resume	プログラムを連続実行します。
00	Suspend	実行中のプログラムを停止します。
	Terminate	デバッガーを停止します。 ModusToolbox™ パースペクティブに移動する前に必ず停止してください。
3	Step Into	クリックする度に1ステートメントを実行しますが、実行するステートメントに関 数が含まれている場合、その関数内部に移動します。
2	Step Over	実行するステートメントに関数が含まれている場合、その関数内部の全てのステー トメントを実行します。
. R	Step Return	現在実行中の関数の呼び出し元まで全てのステートメントを実行します。
2	Restart	Resetが行われ、main()関数先頭のステートメントに戻ります。

Debug(5/12)



> Debugセッションが開始されるとプログラムが書き込まれ、main()関数先頭のス テートメントまで自動的に実行されます。

Note: main() 関数の先頭に自動的にブレークポイントが設定されます。このブレークポイントは Debugger ConfigurationのStartupタブにあるSet Breakpoint at:の項目で指定されています。

💼 main.c 🛛 💽 0x21a 📑 startup_cm7.c 🎽 ⊓ 🗖	Pebug Configurations	- 🗆 X
<pre> • Mainte to oke a peripheral: • main * main ************************************</pre>	Create, manage, and run configurations Create, manage, and run configurations Name: PWM_LED Debug (JLink) Name: PWM_LED Debug (JLink) Main Debugger Startup Source Common Source Co	d at Restart)
<pre>/* Board init failed. Stop program execution ' if (result != CY_RSLT_SUCCESS){ CY_ASSERT(0); }</pre>	© PWM_LED Attach (JLink) © PWM_LED Debug (JLink) © PWM_LED Program (JLink) G PWM_LED Program (JLink) G Launch Group ✓ Set program counter at (hex): ✓ Set breakpoint at: ✓ Continue	
	Filter matched 9 of 13 items	Apply

Debug(6/12)



6. Debugger Action のStep Into 3. アイコンをクリックします main()関数内の最初のステートメントまで実行されたことを確認します。



7. 再度Debugger Action のStep Into 🔜 アイコンをクリックします

Editor Viewにcybsp.cが表示され、cybsp_init()関数内の最初のステートメントまで実行されたことを確認します。



Note: Step Into アイコンはクリックする度に1ステートメントを実行しますが、実行するステートメントに関数が含まれている場合、その関数内部に移動します。

Debug(7/12)



 Debugger Action のStep Return P アイコンをクリックします cybsp_init()関数の残りのステートメントが全て実行され、main()関数のcybsp_init()関数呼び出しステート メントの次まで戻っていることを確認します。ソースコードのresultにマウスカーソルを合わせる事でresult に格納された値を確認する事が可能です。



9. 再度Debugger Action のStep Into 3. アイコンをクリックします main()関数内の次のステートメントまで実行されたことを確認します。



Debug(8/12)



10. Debugger Action のStep Over アイコン 💀 をクリックします __enable_irq() 関数内に移動せず、次のステートメントまで実行されたことを確認します。



11. Editor Viewのスクロールバーをドラッグしてmain.cのsardcIntHandler()関数内の、 Cy_TCPWM_PWM_SetCompare0Val() 関数呼び出し行の先頭をダブルクリックします。 Cy_TCPWM_PWM_SetCompare0Val() 関数呼び出し行にbreakpointが設定されたことを確認します。



Debug(9/12)



12. Debugger Action のResumeアイコン IP をクリックします Breakpointを設定したCy_TCPWM_PWM_SetCompare0Val() 関数呼び出し行まで実行されたことを

確認し、Variable/Breakpoint Viewに変数valが表示されている事を確認します。





Note: 変数valの値(value)はポテンショ メータの位置により、0~4095の値になり ます。



Debug(10/12)



14. Variable/Breakpoint Viewの変数valのValueをクリックし、"4095"を入力して Enterを押した後、 Debugger Action のResumeアイコン № をクリックします 再度breakpointまで実行されたことを確認し、KIT_XMC72_EVKのLED1が消灯している事を確認します。



15. Debugger Action のSkip All Breakpointsアイコン をクリックし、Resumeアイコン をクリックします

ResumeアイコンがDisable IP になり、実行状態が継続している事と、ポテンショメータを回してLED1の輝度が変化する事を確認します。

Note:再度クリックするとアイコンの色が戻りDisableとなります。

Debug(11/12)



- 16. Debugger Action のSuspend アイコン III をクリックします ResumeアイコンがEnable IIN になり、停止状態に移行している事と、ポテンショメータを回してもLED1 の輝度が変化しない事を確認します。
- 17. Debugger Action のRestart アイコン 🕹 をクリックします Resetが行われ、main()関数先頭のステートメントに戻ったことを確認します。



18. Debugger Action のTerminate アイコン Eをクリックします

Debugger ActionのアイコンがDisableし、Debug ViewのスレッドがTerminateされていることを確認し



Debug(12/12)



19. PerspectiveのModusToolbox Perspectiveアイコン Was をクリックしてDebugセッ ションを終了します *Note:Debugセッションを終了する際は必ずTerminateしてください。*





XMC7000 Sensor-less Motor Control



Block Diagram





XMC7000 Peripheral Configuration



Time chart for motor control





- There are three TCPWM to generate PWM for U, V and W
 - The TCPWM for V and W are not shown here
 - Dead time insertion are introduced to line_out and line_compl_out of TCPWM for U, V and W
 - The three ADC are triggered synchronously at under flow of TCPWM for U
 - Once the ADC sampling are done, the ADC ISR (Interrupt Service Routine) will be triggered, and Torque control loop is then performed in the ISR



Appendix

Device Configurator setting value for SARADC



12-b	12-bit SAR ADC 0 (SARADC_0) - Parameters - Device Configurator 4.10			
Enter	filter text		2 5	
Name	2	Value		
~ o	verview			
	⑦ Configuration Help	Open SAR2 Documentation		
∽ G	eneral			
	② Enable SAR Block	\checkmark		
	② Enable The SAR MUX	\checkmark		
	② Enable The SAR ADC			
	Precondition Time	0		
	? Power Up Time	0		
	Power Down If Idle			
	(?) MSB Cycles	Use 1 clock cycles per conversion		\sim
	(?) Half LSB			
	Number Of Channels	1		
~ C	onnections			
	⑦ Clock	8 bit Divider 0 clk (PERI_CLK_1_0) [SHARED]		\sim
	⑦ Clock Frequency	12.25 MHz		
~ C	hannel 0			
	⑦ Channel Name	channel_0		
	? Enabled	\checkmark		
	? HW Trigger	Disabled		\sim
	? Trigger Input	<unassigned></unassigned>		\sim
	? Trigger Output	<unassigned></unassigned>		
	? Trigger Chanel Input	<unassigned></unassigned>		\sim
	? Voltage Range Trigger Output	<unassigned></unassigned>		
	? Channel Done Trigger Output	<unassigned></unassigned>		
	⑦ Debug Freeze Input	<unassigned></unassigned>		\sim
	Priority	0		

12-bit SAR ADC 0 (SARADC_0) - Parameters - Device Configurator 4.10 x					
Enter	nter filter text 🦉 💆 🖻 🖽				
Name		Value	^		
	Priority	0			
	? Preemption Type	Abort ongoing acquisition, do not return \sim			
	⑦ Group End				
	Output Trigger Type	Pulse ~			
	⑦ Input	P6[6] analog (CYBSP_POT) [USED]			
	② Enable External Analog Mux				
	Precondition Mode	No Preconditioning V			
	Overlap Diagnostic Mode	No Overlap or SARMUX Diagnostics 🗸 🗸 🗸			
	③ Sample Time (Aperture)	8			
	? Selection Of Calibration Values	Regular v			
	? Result Data Alignment	The data is right aligned in Result[11:0] V			
	? Sign Extension	Unsigned V			
	? Post Processing Mode	Averaging ~			
	? Averaging Count	1			
	③ Shift Right	0			
	Positive Reload	<u></u> 0			
	? Negative Reload	<u></u>			
	? Range Detection Mode	Below Low Threshold (Result < Low) V			
	? Range Detect Low Threshold	0			
	? Range Detect High Threshold	65535			
✓ Ad	vanced				
	(?) Store Config in Flash		¥		

Device Configurator setting value for TCPWM



the second se		-			
Enter filter text	U	E 4		Enter filter text	
Name	Value	,	•	Name	Value
✓ Inputs				✓ Overview	
⑦ Clock Signal	8 bit Divider 0 clk (PERI_CLK_1_0) [SHARED]	~		⑦ Configuration Help	Open PWM (TCPWM)
⑦ Count Input	Disabled	~		✓ General	
⑦ Kill 0 Input	Disabled	~		(?) TCPWM Version	TCPWM_ver2
? Reload Input	Disabled	~		? PWM Mode	PWM
Start Input	Disabled	~		? Clock Prescaler	Divide by 1
 Swap Input 	Disabled	~		PWM Resolution	🛱 16-bits
Kill 1 Input	Disabled	~		PWM Alignment	Left Aligned
 PWM Output Polarity 	Disabled	_		⑦ Run Mode	Continuous
(?) Invert PWM Output				⑦ Immediate Kill	
Invert PWM n Output		_		✓ Period	_
✓ PWM Disabled Output				② Enable Period Swap	
? PWM Disabled Output	It High Impedance	~		(?) Period	65535
✓ Outputs				✓ Compare	_
PWM (line)	P16[1] digital out (CYBSP USER LED. CYBSP USER LED1. LED1) (USED	01		(?) Enable Compare 0 Swa	р <u>Г</u>
() · · · · · · ((?) Compare 0	0
PWM_n (line_compl)	<unassigned></unassigned>			(?) Enable Compare 1 Swa	р <u></u>
✓ Trigger Outputs		_		(?) Compare 1	0
? Trigger 0 Event	Disabled	~		Interrupt Source Overflow & Underflow	
(?) Trigger 1 Event	Disabled	~		Compare 0	
✓ Advanced		_		Compare 1	
③ Store Config in Flash				✓ Inputs	
				1	

тс	TCPWM[1] Group[0] 16-bit Counter 61 (PWM_1_0_61) - Parameters - Device Configurator 4.10					
En	inter filter text 🧟 💆 🖻					
Na	me		Value		^	
~	Overvie	ew .				
	?	Configuration Help	Open PWM (TCPWM) Documentation			
~	Genera	l				
	?	TCPWM Version	TCPWM_ver2			
	?	PWM Mode	PWM	~		
	?	Clock Prescaler	Divide by 1	``		
	?	PWM Resolution	16-bits			
	?	PWM Alignment	Left Aligned	~	Į.	
	(?)	Run Mode	Continuous	``		
	(?)	Immediate Kill			-	
~	Period					
	?	Enable Period Swap				
	?	Period	65535		-	
~	Compa	re				
	?	Enable Compare 0 Swap			_	
	?	Compare 0	0		_	
	?	Enable Compare 1 Swap				
	?	Compare 1	0			
~	Interru	ot Source				
	?	Overflow & Underflow				
	?	Compare 0				
	?	Compare 1				
~	Inputs				×	
<				>		

Device Configurator setting value for GPIO P6[6], P16[1]



P6[6] (CYBSP_POT) - Parameters - Device Configurator 4.10			
Enter filter text	alian 🖉 🖉 🖉	•	
Name	Value		
✓ Overview			
? Configuration Help	Open GPIO Documentation		
✓ General			
⑦ Drive Mode	Analog High-Z. Input buffer off	\sim	
Initial Drive State	High (1)	~	
✓ Input			
⑦ Threshold	CMOS	\sim	
Interrupt Trigger Type	None	~	
✓ Output			
? Slew Rate	Fast	~	
⑦ Drive Strength	1/2	~	
 Internal Connection 			
⑦ Analog	P 12-bit SAR ADC 0 vin[0] (SARADC_0) [USED]		
⑦ Digital Input	<unassigned></unassigned>		
⑦ Digital Output	<unassigned></unassigned>	~	
⑦ Digital InOut	<unassigned></unassigned>	~	
✓ Advanced			
? Store Config in Flash	\checkmark		

P16[1] (CYBSP_USER_LED, CYBSP_USER_LED1, LED1) - Parameters - Device Configurator 4.10			
Enter filter text	alian 🖂 🗹 📄 🖽		
Name	Value		
✓ Overview			
⑦ Configuration Help	Open GPIO Documentation		
✓ General			
⑦ Drive Mode	Strong Drive. Input buffer off		
Initial Drive State	High (1)		
✓ Input			
⑦ Threshold	CMOS		
Interrupt Trigger Type	None		
✓ Output			
Slew Rate	Fast		
⑦ Drive Strength	1/2		
 Internal Connection 			
Analog	<unassigned></unassigned>		
⑦ Digital Output	PWM_1_0 TCPWM[1] Group[0] 16-bit Counter 61 pwm (PWM_1_0		
⑦ Digital InOut	<unassigned></unassigned>		
✓ Advanced			
⑦ Store Config in Flash			
<	>		

Device Configurator setting value for Peripheral Clock



8 bit Divider 0 (PERI_CLK_1_0) - Parameters - Device Configurator 4.10	
Enter filter text 🦉 🖾 🖻	
Name	Value
 Overview 	
⑦ Configuration Help	Open Peripherals Clock Dividers Documentation
✓ General	2
(?) Source Clock	CLK_HF2 (196 MHz ± 4%)
(?) Divider	16
(?) Frequency	12.25 MHz ± 4%
(?) Start on Reset	
Peripherals	12-bit SAR ADC 0 clock_sar (SARADC_0) [USED]
	TCPWM[1] Group[0] 16-bit Counter 61 clock (PWM_1_0_
<	>

