Customer training workshop: How to Debug on ModusToolBox™ for TRAVEO™ T2G

TRAVEO[™] T2G CYT4BF series Microcontroller Training V1.0.0 2022-12



Please read the Important notice and warnings at the end of this document



Scope of work

- > This document explains how to set up and use the CYT4BF evaluation kit, which mounts the CYT4BFBCH device. It also describes debugging with single-core and multi-core applications in the ModusToolbox[™] (MTB) environment.
- > ModusToolbox[™] tools package version

- 3.0.0

- > Device
 - The TRAVEO[™] T2G CYT4BFBCH device is used in this code example.
- Board
 - The TRAVEO[™] T2G KIT_T2G-B-H_EVK board is used for testing.



> This section explains the hardware setup. The following table lists the prerequisites for the setup.

Quantity	Description	Remarks
1	KIT_T2G-B-H_EVK	CYT4BF evaluation kit
1	Micro USB cable	For power and communication
1	PC	With USB port
1	ModusToolbox 3.0	Downloaded from the web



Getting started (contd.)

Connection setup

 Connect the USB cable from the PC to the evaluation kit. The PC powers the evaluation kit via the USB cable (5V).





› Launch ModusToolbox™

- The Eclipse IDE is installed in the following directory by default:
 <install path>\ModusToolbox\ide <version>\eclipse\
- To launch the Eclipse IDE:
 - -On Windows, select the Eclipse IDE for ModusToolbox™ <version> item from the Start menu
 - -For other operating systems, run the "modustoolbox" executable file
- When launching the Eclipse IDE, there is an option to select the workspace location on your machine. This location is used by the IDE for creating and storing the files as part of application creation for a particular platform. The default workspace location is a folder called "mtw" in your home directory. You may add additional folders under the "mtw" folder or to choose any other location for each workspace.
- For more details about Eclipse, see the Eclipse documentation and the Eclipse survival guide.



- > Create an application
- > Choose the board support package (BSP)
 - To choose the BSP, do one of these:
 - Click the New Application link in the Eclipse IDE Quick Panel



– Select File > New > ModusToolbox™ Application





 These commands launch the Project Creator tool, which provides several applications for use with different development kits, and the kits available may change over time. This example uses the KIT_T2G-B-H_EVK kit.

Choose Board Support Package Settings Help	e (BSP) - Project Creator 2.0		- 0	×
Source Template				
Enter filter text		Browse	KIT T2G-B-H EVK	^
Kit Name ✓ TRAVEO™ BSPs	MCU/SOC/SIP Connectivity	(The KIT_T2G-B-H_EVK, a 272-pin evaluation board is based on the TRAVEO™ T2G family of devices. TRAVEO™ T2G MCU is designed for industrial applications. The	
KIT_T2G-B-H_E	EVK CYT4BFBCHE <none></none>		evaluation board carries a TRAVEO [™] T2G microcontroller, a M.2 interface connector for interfacing radio modules based on AIROC [™] Wi-Fi and Bluetooth® combos (currently not supported), SMIF dual header compatible with Digilent Pmod for interfacing HYPERBUS [™] memories (currently not supported), and headers compatible with Arduino for interfacing Arduino shields. In addition, the board features an on-board programmer/debugger (KitProg3), a 512-Mbit QSPI NOR flash, CAN FD transceiver, Gigabit Ethernet PHY transceiver with RJ45 connector interface, a micro-B connector for USB device interface, three user LEDs, one potentiometer, and two push buttons. The board supports operating voltages from 3.3 V to 5.0 V for TRAVEO [™] T2G device.	
			Kit Features:	~
Summary: BSP: KIT_T2G-B-H_EVK Press "Next" to select ar	polication		Next > Close	
Ð				

- For more details about using this tool, see the Project Creator user guide



> Select application

- On the Choose Board Support Package (BSP) Project Creator 2.0 window, click Next > to open the Select Application page
- This page lists various applications available for the selected kit. As you choose an application, a description displays on the right. You can select multiple applications for the selected BSP by enabling the check box next for those applicable.

> For this example:

- Select the check box next to the "Hello World" application and "Multicore Empty APP" application
- If desired, type a name for the application under New Application Name. Do not use spaces in the application name. In this case, we use the default "Hello_World" and "Multicore_Empty_App" as the name.

Settings Help	
Application(s) Root Path: C:/Users/UmenoKazuo/mtw	owse
Target IDE: Eclipse IDE for ModusToolbox™	
Search V 🚱 👫 🕒 This code example demonstrates simple UART communication by printing "Hollo World" message on a terminal and blinks an LED using a Times on	g a
Template Application New Application Name	ource.
 Getting Started For more details, see the <u>README on GitHub</u>. 	
Empty App	
FreeRTOS Blinky	
Hello_World Hello_World_1	
✓ Multicore Empty App Multicore_Empty_App	
Switching Power Modes	
XMC7000 OOB Demo	
✓ Peripherals	
ADC basic	
Asymptric PWM Generation	
BSP: NI_12G-B-M_EVK	^
Application(s) Root Path: C:/User/UmenoKazio/mtw	- 61
Press "Create" to create the selected application(s).	
	~
< Back Create	lose
\oplus	



> Create application

- Click **Create** to begin the project creation process.



 When complete, the Project Creator tool closes automatically. After several moments, the application opens with the Hello_World and Multicore_Empty_App in Project Explorer, and the README.md file opens in the file viewer.





Download and debug with evaluation kit

> The following types of applications are created:

- Single core application (Hello_World):
 - This application contains the prebuilt CM0+ image, and the main application function runs on the CM7_0 core. The prebuilt CM0P image only starts CM7 cores and puts CM0+ core into Deep Sleep mode.
 - For details on the prebuilt CM0+ image, please refer to CAT1 Cortex M0+ prebuilt images.

– Multi-core application (Multicore_Empty_App):

- This application contains the CM0+ project, CM7_0 project, and CM7_1 project.
- CM0+ and CM7 can do normal code execution, but from an architectural point, only CM7 is considered the application core (CM7 cores for primary processing and CM0+ core for peripheral and security processing). After a reset, the default core is always the CM0+ core. To enable the CM7 core, CM0+ must call Cy_SysEnableCM7().



> Debugging with single core application:

1. Build the application

- a) Select the Hello_World project in the Project Explorer window and click on the Build Hello_World Application shortcut under the Hello_World group in the Quick Panel. It selects the Debug build configuration and compiles/links all projects that constitute the application.
- b) The **Console** view lists the results of the build operation, as the following figure shows.
- If you encounter errors, revisit previous steps to ensure that you accomplished all the required tasks.

T-T				-	
mtw - Hello_World/main.c - Eclipse IDE for Mo	us loolbox."		-		, x
File Edit Source Refactor Navigate Search	Project Run Window Help				_
📑 🕶 🔛 🚱 🖉 📲 🖓 🕶 🔛 💷 🕪 💷	🖩 14 고, 오그리 국 🕱 🖉 🐇 🕶 🔾 📲 🧶 🖉 🖬 🖉 🖬 🗉 원 * 원 * 번 다	((+ ()	> + 🛃 👘	Q	8
陷 P 🛛 🎋 D 🎹 R 🧏 P 🖓 🗖	README.md README.md Readme.md		E Outline	3	- 8
E S Hello_World	⊖/*File Name: main.c	^	⊫ ↓ª _≥ ⊗ 8	× ^{\$} • `	# . \$*
> SE Branies > De beps > De beps > De build > De deps > De lines > De l	 Description: This is the source code for Hello World Example using HAL APIs. Related Document: See README.md Copyright 2022, Cypress Semiconductor Corporation (an Infineon company) or an affiliate of Cypress Semiconductor Corporation. All rights reserved. This software, including source code, documentation and related materials ("Software") is owned by Cypress Semiconductor Corporation or one of its affiliates ("Cypress") and is protected by and subject to werladde patter protection (Whited States and foreign), information (Software") is orthware package from which you obtained this Software CPULAP. The of ULA applies, Cypress hereby grants you a personal, non-exclusive, non-transferable license to copy, modify, and compile the Software source code solely for us in connection with Cypress's 		Synthesis and Synthesis an	al.h sp.h etarget_ic _BLINK_T _BLINK_T r_init(voi er_init(voi loink_act _read_val blink_tim n(void) := r_init(voi imer(void)	b.h IMER_CLO IMER_PERI id): void ive_flag: bd ive_flag: bd ive_flag: b ue: uint8_ ive: cyhal_t int id): void if, cyhal_ti
Suild Hello_World Application	* compilation, or representation of this Software except as specified	~			
Sclean Hello_World Application		>	<		-
▼ Launches	📮 Console 🔀 🚼 Problems 🖏 Progress 🚺 Memory 🖑 Terminal 🛛 🕹 😚 😫		- 🔍 🛃 🕻] - 📑	• • •
株 Hello_World Debug (/Link) 株 Hello_World Debug (KitProg3_MiniProg4)	Lui suu console (relio_wong) .noinit 0x28020358 .bss 0x280203ec .heap 0x2802125c				
Hello_World Program (LLink) Hello_World Program (KitProg3_MiniProg Generate Launches for Hello_World	Total Internal Flash (Avsilable) 8585216 Total Internal Flash (Utilized) 52240				
 ▼ Tools ☑ Library Manager 2.0 ☑ Device Configurator 4.0 (new configuration) 	Total Internal SRAM (Avsilable) 1048576 Total Internal SRAM (Utilized with heap) 389120				
Device Firmware Update Host Tool 1.60	Torrerso opera remainder o criora, o marmelligar (cook 253.444015)				-
CODI Configurator 4.0	<				>
😂 Hello_World					



- 2. Debug the application:
 - The CYT4BF evaluation kit has a KitProg3 onboard programmer/debugger. It supports Cortex[®] Microcontroller Software Interface Standard - Debug Access Port (CMSIS-DAP). See the KitProg3 user guide for details.
 - ModusToolbox[™] software uses the OpenOCD protocol to program and debug applications on the CYT4BF MCU devices. ModusToolbox[™] software will identify the device on the kit only if the kit is running KitProg3.
 - In the Quick Panel, click the Hello_World Debug (KitProg3) link under Launches.

mtw - Hello_World/main.c - Eclipse IDE for Mo	dusToolbox™		-			
e Edit Source Refactor Navigate Search	Project Run Window Help					
• 🔛 🕲 • 🗞 • 🔛 🛄 🕨 🗉	■ 24 24 10 11 11 11 12	⇔ • ⇔	- 1	م ∶⊯∣		
P 🛛 🎄 D 🛲 R 🔜 P 🖓 🗖	🖬 README.md 💽 README.md 💽 main.c 🔀		📴 Outline 🖂	-		
E 🕏 🍸 🕴	⊖ /************************************	^	🗏 🗐 🖓 😿	● 推 ○		
Hello_World	* File Name: main.c		8			
> 🖑 Binaries	* * Description: This is the source code for Hello World Example using HAL APTs.		o cyhal.h			
> 😝 bsps	*		sa cyhann			
> 📴 build	* Related Document: See README.md		A cy retar	aet io.h		
> 🗁 deps	*		# LED BLI	NK TIMER C		
> 🗁 images	-		# LED BLI	NK TIMER P		
> 🗁 libs	* Copyright 2022, Cypress Semiconductor Corporation (an Infineon company) or		i timer_in	it(void) : voi		
> 🚘 release-v4.0.0	* an affiliate of Cypress Semiconductor Corporation. All rights reserved.		++ ^s isr_timer	r(void*, cyha		
> 🖻 main.c			timer_in	terrupt_flag		
LICENSE	* Inis software, including source code, documentation and related * materials ("Software") is owned by Synness Semiconductor Corporation		led_blink	k_active_flag		
🚡 Makefile	* or one of its affiliates ("Cypress") and is protected by and subject to		uart_rea	d_value : uin		
README.md	* worldwide patent protection (United States and foreign),		led_blink	k_timer : cyh		
🚔 mtb_shared	* United States copyright laws and international treaty provisions.		main(vo	id) : int		
Multicore_Empty_App	* Therefore, you may use this Software only as provided in the license * agreement accompanying the software package from which you		timer_in	it(void) : voi		
	* obtained this Software ("EULA").		S isr_time	r(void*, cyha		
Qu (x)= Va 🥸 Ex 💁 Br 🗀 🗖	* If no EULA applies, Cypress hereby grants you a personal, non-exclusive,					
	* non-transferable license to copy, modify, and compile the Software					
Hello_World (APP_KIT_XMC72_EVK)	 source code solely for use in connection with Cypress's integrated circuit products Any reproduction modification translation 					
Build Hello World Application	* compilation, or representation of this Software except as specified	~				
	< C C C C C C C C C C C C C C C C C C C	>	<			
Clean Helio_World Application	Concela 😒 💌 Dephamer 💷 Descarere 🕕 Mamoner 💭 Terminal		- 🔍 🚽 🗖 🗸			
Launches	CDT Build Concole (Hello World)	198 (NR -		L2 ·		
Ka Malla Wardd Dabwa (Iliala)	con build console (nello_worka)					
Se Hello_world Debug (JEllik)	Prebuild operations complete					
体 Hello_World Debug (KitProg3_MiniProg4)	Commencing build operations					
D Hello_World Program (JLink)	Tools Directory, C. /Users/Tuedavid/MedusTeelbay/tools 2.0					
Hello World Program (KitProg3 MiniProg						
Consider the formation for a little billion billion	Initializing build: mtb-example-psoc6-hello-world Debug APP_KIT_XMC72_EVK GCC_ARM					
Generate Launches for Hello_world						
Tools						
Z Library Manager 2.0	Constructing build rules					
Library Manager 2.0	Build rules construction complete					
Device Configurator 4.0 (new configuratio	Auto-discovery in progress					
Device Firmware Update Host Tool 1.60	Auto-discovery complete					
▼ OSPI Configurator 4.0						
>	٢			3		
PHelic, World Program (KiRVpog3_MiniProg Melic, World Program (KiRVpog3_MiniProg Generate Launches for Helic, World Tools Device Configurator 4.0 (new configuratio Device Firmware Update Host Tool 1.60 OSIP Configurator 4.0	Tools Directory: C:/Users/zuodavid/ModusToolbox/tools_3.0 Initializing build: mtb-example-psoc6-hello-world Debug APP_KIT_0KC72_EVK GCC_ARM Constructing build rules Build rules construction complete Auto-discovery in progress Auto-discovery complete	ching Hello	· WorldniProa4):	(4%)		



- If needed, the IDE builds the application, and messages display in the Console. If the build is successful, the IDE switches to debug mode automatically, as the following figure shows.
- If you want to open the disassembly window, click Window > Show View > Disassembly.



- 3. Click the **Resume** icon or press the **F8** to start execution. LED1 should start blinking.
 - You can also use the function keys in the Debug window: Resume (F8), Step Into (F5), Step Over (F6), Terminate (Ctrl+F2).



> Debugging with multi-core application:

1. Build the multi-core application

- a) In the Project Explorer window, click the Multicore_Empty_App project.
- b) Click on the Build Multicore_Empty_App Application shortcut under the Multicore_Empty_App group in the Quick Panel. It selects the Debug build configuration and compiles/links all projects that constitute the application.
- c) The **Console** view lists the results of the build operation, as the following figure shows.
- If you have selected Multicore_Empty_App, you can also select Build Project from the Project menu or the right-click menu.





2. In the **Quick Panel**, click the **Multicore_Empty_App MultiCore (KitProg3)** link under **Launches**.

-		
Witcore_Empty_App.proj_cm0p/main.c - Eclips	e IDE for ModusToolbox™	– 🗆 ×
File Edit Source Refactor Navigate Search Project	t Run Window Help	
🔁 🕶 🔚 🐚 🛛 🐨 🗞 🕶 🔛 💷 🖿 🗤	과 ④ 과 국 🕱 🕹 🕸 = 🔕 = 🏰 = 🎒 🖋 = 🗐 💷 🖩 👔 = 친 = 친 - 친 - 〇 -	- 🛃 🔍 📴 📔
🖒 Proj 🔀 🎄 Deb 🚟 Regi 🔒 Peri 🖓 🗖	README.md Comain.c 🛛 Comain.c	📴 Outline 🕴 👘 🗖
		59 D I4 N X ⁸ A
> 🛱 Hello World		<i>π</i>
> 👺 mtb_shared	<pre># #include "cy_pdl.h" # #include "cycfg.h"</pre>	8
✓ ☑ Multicore_Empty_App	<pre>#include "cyhal.h"</pre>	8 El cy ndl h
> 🗁 bsps	#include. "cybsp.h"	L cycfg.h
> 😹 release-v4.0.0		🛃 cyhal.h
S Multicore_Empty_App.proj_cm0p (in proj_cm0p)	#define CM7_DUAL 1	🛃 cybsp.h
See Multicore Empty App.proj cm7 1 (in proj cm7 1)	⊖ int main(void)	# CM7_DUAL
> 🛞 Binaries	{	main(void) : int
ommon_app.mk	<pre>init_cycfg_all(); /* enable interrunts */</pre>	
lo common.mk	enable_irq();	
LICENSE National		
README.md	Cy SysEnableCM7(CORE CM7 0, CY CORTEX M7 0 APPL ADDR is calculated in linker script, (
	Cy_SysLib_Delay(1000); /* To avoid race between CM7's while initializing th	
	#if CM7 DUAL	
	Cy_SysEnableCM7(CORE_CM7_1, CY_CORTEX_M7_1_APPL_ADDR);	
🔛 Quick (x)= Varia 😚 Expr 💁 Break 🖓 🗖		
	😑 Console 🛛 💦 Problems 🖏 Progress 👖 Memory 🧶 Terminal 🛛 🐥 🏠 🚮 🚮 = 🔩	
Eclipse IDE for	CDT Build Console [Multicore_Empty_App]	
MedusTeelbey™	.noinit 0x28080100 148	
Modusiotbox	.bss 0x28080194 2304 ,heap 0x28080a94 517484	
Start		
Multicore Empty App	Total Internal Flash (Available) 8585216	
* Launcher	Total Internal Flash (Utilized) 3456	
· Laurences	Total Internal SPAM (Available) 1948576	
Multicore_Empty_App Debug MultiCore (JLink)	Total Internal SRAM (Utilized with heap) 520192	
Multicore_Empty_App Debug MultiCore (KitProg3_Iv		
K Generate Launches for Multicore_Empty_App	make[1]: Leaving directory '/cygdrive/c/Users/zuodavid/mtw/Multicore Empty App/proj cm7	1'
▼ Tools	15:12:47 Ruild Einished A annens A uppnings (teak 1m:42s 14Ems)	
X Library Manager 2.0	10:12:47 build Finished. @ errors, @ warnings. (took im:45s.145ms)	
Elonory Manager 2.0		
	<	>



- This will automatically program the CM0P, CM7_0, and CM7_1 code into the flash region of respective cores; then, the IDE switches to debug mode automatically, as the following figure shows.
- CM0P debug session is started and halted at the beginning of the main () function, CM7_0 and CM7_1 debug session started, and CPU is not yet started in the following figure.





3. Place a breakpoint in the cybsp_init() API in the mian.c of CM7_0 core, you can also place another breakpoint in main.c of CM7_1 core. CM7_0 core and CM7_1 core will start executing after being enabled by CM0+ core. You can debug three cores simultaneously.

 To place a breakpoint at the target instruction, click the white space between the editor window (left pane).

mtw - Multicore_Empty_App.proj_cm7_0/main.c - Eclipse IDE	for ModusToolb	ox [™]							-	D X
File Edit Source Refactor Navigate Search Project R	un Window H	lelp								
📑 • 🔄 🐚 🗞 • 🗞 • 🛗 🎋 • 🛈 • 🏊 • 🥭 /	🔗 🛨 📮 🕪	0 🔳 🖓 :	3. 10. 10.	30.	۵ 😼 🇾 🕹	n i 😫 🔻 👌	i • 🏷 🕫 🔶 •	- 🗢 - 🛃 👘	Q	: 🗈 🎴
🏠 Projec 🐐 Debug 🙁 🔠 Regist 😤 Periph 🖓 🕻	README	.md 💽	main.c	💼 main.c 🖇	🕄 🚺 main.c	C 0x366	💲 startup_cm0p	⁻ -	🗄 Outline 🛛	- 0
A Multicore, Empty, App Debug MultiCore (KitProg3, MiniPro Ceterminated Multicore, Empty, App proj.cm7, D Program Ceterminated Multicore, Empty, App proj.cm7, D Program Multicore, Empty, App proj.cm7, D Program Multicore, Empty, App proj.cm7, D Proga main() at main.c53 0x10000eas openco.cd.ee arm-none-cabi-gdb.ce Multicore, Empty, App proj.cm7, D Add CM7,0 to CM0+ (K Multicore, Empty, App proj.cm7, D Add CM7,0 to CM0+ (K Wilticore, Empty, App proj.cm7, D Add CM7,1 to CM7,0 (k Wilticore, Empty, App proj.cm7, D Add CM7,1 to CM7,0 (k Wilticore, They App proj.cm7, D Add CM7,1 to CM7,0 (k Wilticore, T), Leff Prosed =1 (Running : User Request) arm-none-cabi-gdb.ce	3	<pre>ade "cy_pdl ade "cy_bal ade "cybal ade "cybap ain(void) /_rslt_t re " Initializ sult = cyb E (result - 1 CY_ASSEF " Enable gl enable_irc or (;;)</pre>	<pre>k.h." .h." esult; te the dev ssp_init() = (Y_RSIT_ T(0); lobal inter {();</pre>	ice and by Success)	bard periphen	als */			 In the second sec	Lh h .h vvoid)∶int
<	> /* []	END OF FIL	LE */							
🗳 Quick (x)= Variables 🙀 Expres 💁 Breakp 🗖 🗖	3 <							>		
_	Console	23 💽 Prob	lems 📖 Pr	oaress 👖	Memory 🧔 Ter	minal	🔳 🗙 🖗 🛛		🗐 🚽 🖸 🕶 🛛	9 • • •
multicore_cmpty_App	^ Multicore_En	pty_App.proj	_cm0p Debu	g (KitProg3_I	MiniProg4) [GDB	OpenOCD Deb	ugging]			
▼ Launches	(46) 04 () (47) d5 ()	64): 0X008	112020000	1100						
Multicore_Empty_App Debug MultiCore (JLink)	(48) d6 ()	64): 0x209	0000248402	2000						
Multicore_Empty_App Debug MultiCore (KitProg3_MiniPro	(49) d7 () (50) d8 ()	'64): 0x120 '64): 0x000	200208c008	3000 3000						
K Generate Launches for Multicore_Empty_App	(51) d9 (/ (52) d10	64): 0x000 /64): 0x80	000cc01100	9040 98184						
▼ Tools	(53) d11 (/64): 0x08	110000080	20445						
Library Manager 2.0	(55) d13	/64): 0xc2	08404c0000	94108						
Device Configurator 4.0 (new configuration)	(56) d14 ((57) d15	/64): 0x07	0000080500	34280 50000						
SPI Configurator 4.0	✓ (58) fpscr	(/32): 0x	00000000							
< >>	<									>
			Writable		Smart Insert	53:1:2	2557			



4. Click the **Resume** icon or press the **F8** in the CM0P project to start execution.

> After executing

Cy_SysEnableCM7 (CORE_CM7_0, CY_CORTEX_M7_0_APPL_ADDR) and Cy_SysEnableCM7 (CORE_CM7_1, CY_CORTEX_M7_1_APPL_ADDR), CM7_0 core and CM7_1 core will be enabled and the execution will be halted at the very beginning of CM7's main () function.

You can click the Resume icon or press the F8 in the CM7_0 and CM7_1 project, the execution will reach the breakpoint in the CM7_0 project and CM7_1 project. You can now continue to debug the code from CM7 cores.





Troubleshooting

> Connection troubleshooting

- Error: Evaluation kit is not detected on the target system
 - Connect the USB cable that comes with the evaluation kit. Other USB cables may not connect data lines
 - Make sure LED D5 is ON (CMSIS-DAP mode). If not, press SW3 to change the KitProg3 device mode
 - If LED D5 is ON (CMSIS-DAP mode), change the KitProg3 device mode by pressing SW3. Now LED3 blinks smoothly. Then try to reconnect the debug session. Later, check if this works independently; stop the debug session again and switch the KitProg3 device mode to "LED D5 is always ON" (CMSIS-DAP mode)

> Driver troubleshooting

- Error: Driver is not detected on the target system or "KitProg3" is not visible
 - For more information on the supported driver, see the KitProg3 user guide

> Debugger troubleshooting

- Error: While programming the XMC7000 device, the CMSIS-DAP device is not found
 - Check the USB cable connection and the state of LED3 (LED should be ON for CMSIS-DAP mode)

> Key points

 The prebuilt CM0+ image should be disabled with the multi-core application; you must add the XMC7xDUAL_CM0P_SLEEP to DISABLE_COMPONENTS for XMC7xxxD device in the project CM7_0 Makefile

Datasheet

- > <u>CYT4BF datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family</u>
- Architecture Technical reference manual
- > TRAVEO™ T2G automotive body controller high family architecture technical reference manual
- **Registers Technical reference manual**
- > TRAVEO[™] T2G Automotive body controller high registers technical reference manual

PDL/HAL

- > <u>PDL</u>
- > <u>HAL</u>

Training

→ TRAVEO™ T2G Training

Application note

> AN235305 - Getting started for ModusToolbox™ in TRAVEO™ T2G family MCUs

User guide

> KitPro3 user guide



Revision History

Revision	ECN	Submission Date	Description of Change
**	7845778	2022/12/9	Initial release



Important notice and warnings

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-12 Published by Infineon Technologies AG 81726 Munich, Germany

© 2022 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document? Go to: www.infineon.com/support

Document reference 002-36716 Rev. **

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics

("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

WARNINGS

The data contained in this document is

exclusively intended for technically

trained staff. It is the responsibility of

customer's technical departments to

evaluate the suitability of the product for

the intended application and the

completeness of the product information

given in this document with respect to

For further information on the product,

conditions and prices please contact

your nearest Infineon Technologies

office (www.infineon.com).

terms

and

delivery

such application.

technology.

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.