# Customer training workshop: Multi-core application debugging in ModusToolbox<sup>™</sup> for TRAVEO<sup>™</sup> T2G



V1.0.0 2022-12

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



- > Multi-core platform support
  - Multi-core application development
    - The ModusToolbox<sup>™</sup> build and program/debug support has been extended to support multi-core devices, such that the code can be developed on all (user-accessible) cores.
  - Multi-core device configuration
    - The Device Configurator has been enhanced to enable the generation of peripheral configuration code for all cores in a multi-core device from one *design.modus* file.
  - Embedded Trace Macrocell (ETM) support in µVision and EW-ARM
     ETM is offered on third-party IDEs only, using the SEGGER J-Trace hardware and the IDE vendor's own tools.

- > There are two types of applications:
  - Single core application

For example: The *Hello\_World* application contains the prebuilt CM0+ image, and the main application function runs on the CM7\_0 core. The prebuilt CM0+ image only starts the CM7 core and puts the CM0+ core into Deep Sleep mode.

For details of the prebuilt CM0+ image, please check <u>CAT1 Cortex<sup>®</sup> M0+ prebuilt images</u>.

- Multi-core application:

For example: The *Multicore\_Empty\_App* application contains the CM0+ project, CM7\_0 project, and CM7\_1 project.

Both CM0+ and CM7 can do normal code execution but from an architectural viewpoint, only CM7 is considered as 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 the **Cy\_SysEnableCM7()** function.



## Build the multi-core application

- a) In the Project Explorer window, click on the Multicore\_Empty\_App project.
- b) Click on the Build 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





## Launch the multi-core application debugging

- In the Quick Panel, click the Multicore\_Empty\_App Debug MultiCore (KitProg3) link under Launches.
  - 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.

#### 

- Multicore\_Empty\_App Program Application (JLink)
- Multicore\_Empty\_App Program Application (KitProg3\_MiniProg4)
- Multicore\_Empty\_App Debug MultiCore (JLink)
- Multicore\_Empty\_App Debug MultiCore (KitProg3\_MiniProg4)

K Generate Launches for Multicore\_Empty\_App

 The CMOP debug session is started and halted at the beginning of the main () function. Note that when the CM7\_0 and CM7\_1 debug sessions start, CPU has not yet started (as shown in the figure).

i mtw - Multicore_Empty_App.proj_cm0p/main.c - Eclipse IDE for ModusToolbox™	-		>	×
File Edit Source Refactor Navigate Search Project Run Window Help				
i 🗂 • 🗐 🐚   🗞 • 🔦 • 励 i 🖳 i 🕨 💷 🛤 🧏 👁 🕼   🗮 🖉 🕹 i 🎋 • 🕻	) = 🂁 = [ 🏄 🖉 = [ 🥒 🖩 🏾 🛛 🖢 = 🖓 = 🏷 =? 🗇 =   📷	9	😰	2
Project Explorer *tp Debug ⊠ III Register * Peripherals      A findhicore_Empty_App Debug MultiCore (KtAProg3_MiniProg4) [Launch Group]     Of Multicore_Empty_App.roj.cmD Debug (KtAProg3_MiniProg4) [GBB OpenOCD Debugging]     Proj.cmOp.eff     Monte 41 1 (Name Current Execution) (Suppended : Breakpoint)     main() at main.c54 0x1000014a     openocd.exe     Aminon-eable.gdb.exe     Multicore_Empty_App.proj.cm7_0.Add CM7_0 to CM0+ (KtAProg3_MiniProg4) [GDB OpenOCD     Prove 20 Auff	BEADMEnd @ READMEnd @ manc 23 * so agrees to indemnify Cypres against all liability. #include "cypld1.h" #include "cyclsp.h" #include "cyclsp.n" #define CM7_DUAL 1		^	8
The projection of the second se	<pre>@int main(void) {     /* enable interrupts */    enable_interrupts */    enable_interrupts */     /* Enable GM7_0/1. CV_CORTEX_M7_APPL_ADDR is calculated in linker script, check it in case of prob     Cy_SystableCM7(CORE_CM7_0, CV_CORTEX_M7_0_APPL_ADDR);     Cy_SystableCM7(CORE_CM7_1, CV_CORTEX_M7_1_APPL_ADDR); #endif /* CM7_DUAL     Cy_SystableCM7(CORE_CM7_1, CV_CORTEX_M7_1_APPL_ADDR); #endif /* CM7_DUAL     SystemCoreClockUpdate();     Cy_Systm_CpuSleepOnExit(true); </pre>	lems. "		
<ul> <li>کواند ( Panel ا الله الله الله الله الله الله الله ا</li></ul>	<pre>for(;;) {     Cy_SysPm_CpuEnterDeepSleep(CY_SYSPM_MAIT_FOR_INTERRUPT); } /* [] END OF FILE */</pre>			
> Start	<	>	. *	
Multicore_Empty_App (APP_KIT_T2G-B-H_EVK)  Launches  Multicore_Empty_App Program Application (/Link)  Multicore_Empty_App Program Application (KtProg3_MiniProg4)  Multicore_Empty_App Debug MultiCore (/Link)  Multicore_Empty_App Debug MultiCore (KtProg3_MiniProg4)  Generate Launches for Multicore_Empty_App  Tools  Classification 1.0	© Console ⊠ Problems → Progress		•	
V V			_	



- Place a break point in the cybsp\_init() API in the main.c of CM7\_0 core. You can also place another break point in the main.c of CM7\_1 core.
- CM7\_0 core and CM7\_1 core start executing after they are enabled by the CM0+ core.
- You can debug three cores simultaneously.

🎬 mtw - Multicore_Empty_App.proj_cm7_0/main.c - Eclipse IDE for ModusToolbox™		- 0	×
File Edit Source Refactor Navigate Search Project Run Window Help			
🖆 🕶 📾 🛯 🥸 🕶 🗞 🕶 📾 🖥 🖳 💷 🛤 🔊 👁 💷 🗮 😒 🕹 🕸 🍷 🛇	) ▼ 💁 ▼ 🕼 ∥ 🖩 🗊 🗄 🖞 ▼ 🕅 ▼ 🗁 라 ♥ 🚽 🗠 ▼ 🚽 📩	۹ 🗄	3   🔛
🏠 Project Explorer 🐐 Debug 🙁 🗱 Registers 🥋 Peripherals 💿 🙀 🔹 🖇 🙄 🗖	🖻 main.c 🛛 🖻 main.c	-	- e
<ul> <li>Multicore_Empty_App.pobug MultiCore (KRProg3_MiniProg4) [Launch Group]</li> <li>Multicore_Empty_App.poi_cm0p Debug (KRProg3_MiniProg4) [GDB OpenOCD Debugging]</li> <li>Proj_cm0p.edf</li> <li>makin0 at main.c35 40:1000014a</li> <li>generative absolution (Suspended E Breakpoint)</li> <li>makin0 at main.c35 40:1000014a</li> <li>generative absolution (Suspended E Breakpoint)</li> <li>arm-non-eabsigdbace</li> <li>Though at main.c35 back</li> </ul>	<pre>* not authorize its products for use in any products where a malfunction or * failure of the Cypress product may reasonably be expected to result in * significant property damage, injury or death ("High Risk Product"). By * including Cypress's product in a High Risk Product, the manufacturer * of such system or application assumes all risk of such use and in doing * so agrees to <u>indemity</u> Cypress against all liability. ************************************</pre>	,	• 8≊
Proj_cm7_1.elf	⊖ int main(void)		
📕 arm-none-eabi-gdb.exe	<pre>{     cy_rslt_t result; </pre>		
	/* Initialize the device and board peripherals */		
	if (result != CY RSLT SUCCESS)		
	{ CY_ASSERT(0); }		
	<pre>/* Enable global interrupts */enable_irq();</pre>		
< > >	for (;;)		
🔐 Quick Panel 🗱 Variables 🙀 Expressions 💁 Breakpoints 📃 🗖			
Eclipse IDE for	}		
ModusToolbox™	/* [] END OF FILE */		
Modustoolbox			
Start		>	
Multicore_Empty_App (APP_KIT_T2G-B-H_EVK)	🕒 Console 🛛 🎦 Problems 🖶 Progress 🚺 Memory 🖑 Terminal 🎬 Disassembly		
▼ Launches	Multicore Empty App.proj cm0p Debug (KitProg3 MiniProg4) (GDB OpenOCD Debugging)	🔤 🖳 ד 🗋	•
Multicore_Empty_App Program Application (JLink)	(51) d9 (/64): 0x0000004c01100040		^
Multicore_Empty_App Program Application (KitProg3_MiniProg4)	(52) d10 (/64): 0x0000180208108184 (53) d11 (/64): 0x08b1000000022445		
Multicore_Empty_App Debug MultiCore (JLink)	(54) d12 (/64): 0x0008100001405002		
Multicore_Empty_App Debug MultiCore (KitProg3_MiniProg4)	(55) d13 (/64): 0xc208400800004108 (56) d14 (/64): 0x0710000805004280		
K Generate Launches for Multicore_Empty_App	(57) d15 (/64): 0x1008200017850000		
▼ Tools	===== Cortex-M DWT registers		
BSP Assistant 1.0	4	>	~
			-
	•		

## Multi-core application debugging (contd.)



- > Now, click the **Resume** icon or press **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 beginning of the CM7 main () function.
- You can click the **Resume** icon or press 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.







- The flash and RAM sections for the CPU are defined in the linker files:
  - Multicore\_Empty\_App\bsps\TARGET\_APP\_KIT\_T2G-B-H EVK\COMPONENT CMOP\TOOLCHAIN GCC ARM\linker.ld
  - Multicore\_Empty\_App\bsps\TARGET\_APP\_KIT\_T2G-B-H EVK\COMPONENT CM7\TOOLCHAIN\_GCC\_ARM\linker.ld
- The following example modifies the flash and RAM size:







## How to modify CM0P and CM7 cores flash and RAM size (2)

1. Modify CM0P core flash and RAM size in the *bsps\TARGET\_APP\_KIT\_T2G-B-H\_EVK\COMPONENT\_CM0P\TOOLCHAIN\_GCC\_ARM\linker.ld* 

<ul> <li>✓          <sup>™</sup> Multicore_Empty_App         <ul> <li>✓              <sup>™</sup> bsps             </li> <li>✓              <sup>™</sup> TARGET_APP_KIT_T2G-B-H_EVK             </li> </ul> </li> </ul>	<pre>sram_start_reserve sram_private_for_srom</pre>	= 0; = 0x00000800; /* Private SRAM for SROM (e.g. API processing). Reserved at the
✓ ➢ COMPONENT_CMOP > ➢ TOOLCHAIN_ARM	<pre>cm0plus_sram_reserve cm0plus_code_flash_reserve</pre>	= 0x00020000; /* cm0 sram size */ = 0x00080000; /* cm7_0 sram size */
<ul> <li>         TOOLCHAIN_GCC_ARM         S startup_cm0plus.S         Inker.Id         TOOLCHAIN_IAR         </li> </ul>	<pre>cm/_0_code_flash_reserve sram_base_address code_flash_base_address code_flash_total_size</pre>	= 0x28000000; = 0x10000000; = 0x00080000;

2. Modify CM7 cores flash and RAM size in the *bsps\TARGET\_APP\_KIT\_T2G-B-H\_EVK\COMPONENT\_CM7*\TOOLCHAIN\_GCC\_ARM\linker.ld

✓ ☑ Multicore_Empty_App	sram_start_reserve	= 0;
<ul> <li>bsps</li> <li>TARGET_APP_KIT_T2G-B-H_EVK</li> <li>COMPONENT_CM0P</li> <li>COMPONENT_CM7</li> </ul>	sram_total_size sram_private_for_srom sram_used_by_boot	= 0x00100000; /* SRAM0 + SRAM1 */ = 0x00000800; /* Private SRAM for SROM (e.g. API processing) */ = 0x0; /* Used during boot by Cypress firmware (content will be overwritten on re
> 👝 TOOLCHAIN_ARM	cm0plus_sram_reserve	= 0x00020000;
Iinker.ld TOOLCHAIN_IAR startup_cm7.c	<pre>code flash total size cm0plus_code_flash_reserve cm7_0_code_flash_reserve</pre>	= 0x00830000; = 0x00080000; = 0x00200000;



3. The CY\_CORTEX\_M7\_0\_APPL\_ADDR and CY\_CORTEX\_M7\_1\_APPL\_ADDR macro must be used as the parameter for the Cy\_SysEnableCM7() function in CM0P *main.c* to enable the CM7 core. It is defined in the bsps\TARGET\_APP\_KIT\_T2G-B-H\_EVK\system\_cat1c.h file.



# The **BASE\_CODE\_FLASH\_CM7\_0** and **BASE\_CODE\_FLASH\_CM7\_1** is defined in the bsps\TARGET\_APP\_KIT\_T2G-B-H\_EVK\xmc7xxx\_partition.h file; user needs to modify the flash and RAM size in this file.

🖌 🗁 bsps	#detine CODE_FLASH_BASE_ADDRESS	0x10000000 /* FLASH START */
TARGET ADD KIT TOG-R-H EVK	#define CM0PLUS_CODE_FLASH_RESERVE	0x80000 /* 512K CMOP FLASH SIZE */
	#define CM7_0_CODE_FLASH_RESERVE	0x200000 /* 2048K CM7_0 FLASH SIZE */
> COMPONENT_CMOP		
> COMPONENT_CM7	/* SRAM reservations */	
> 🕞 config	#define BASE_SRAM_CM0P	<pre>SRAM_BASE_ADDRESS + SRAM_START_RESERVE + SRAM_PRIVATE_FOR_SROM</pre>
> 🕞 deps	#define SIZE_SRAM_CM0P	CM0PLUS_SRAM_RESERVE - SRAM_START_RESERVE - SRAM_PRIVATE_FOR_SROM
> Condess	#define BASE_SRAM_CM7_0	SRAM_BASE_ADDRESS + CM0PLUS_SRAM_RESERVE
	#define SIZE_SRAM_CM7_0	CM7_0_SRAM_RESERVE
> h cybsp_doc.n	#define BASE_SRAM_CM7_1	<pre>SRAM_BASE_ADDRESS + CM0PLUS_SRAM_RESERVE + CM7_0_SRAM_RESERVE</pre>
> h cybsp_types.h		
> 🖻 cybsp.c	<pre>/* Code flash reservations */</pre>	
> b) cybsp.h	#define BASE_CODE_FLASH_CM0P	CODE_FLASH_BASE_ADDRESS
S istartup catte h	#define STZE CODE FLASH CMAD	CMARLIS CODE ELASH RESERVE
> m startup_cuttern	#define BASE_CODE_FLASH_CM7_0	CODE_FLASH_BASE_ADDRESS + CM0PLUS_CODE_FLASH_RESERVE
> in system_catic.n		
h xmc7xxx_partition.h	#define BASE_CODE_FLASH_CM7_1	CODE_FLASH_BASE_ADDRESS + CM0PLUS_CODE_FLASH_RESERVE + CM7_0_CODE_FLASH_RESE
bsp.mk		
EULA		
LICENSE		
propriego	<pre>#endif /* LAYOUT_CAT1C_H */</pre>	
W READIVIE.ma		
RELEASE.md	/* [] END OF FILE */	
and the second se	~	



### 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 technic
- > al reference manual
- **Registers Technical reference manual**
- > TRAVEO™ T2G Automotive body controller high registers technical reference manual
- Training
- → TRAVEO™ T2G Training
- > How to debug on MTB for KIT\_T2G-B-H\_EVK (Doc No. 002-36716)



## **Revision History**

Revision	ECN	Submission Date	Description of Change
**	7847279	2022/12/13	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 002-36718 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.