Customer training workshop: Multi-core application debugging in ModusToolbox™ for TRAVEO™ T2G

V1.0.0 2022-12

ModusToolbox™ software version 3.0

- Multi-core platform support

  - Multi-core application development
    The ModusToolbox™ 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.

Introduction to application types

- 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 **CAT1 Cortex® M0+ prebuilt images**.

  - 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).

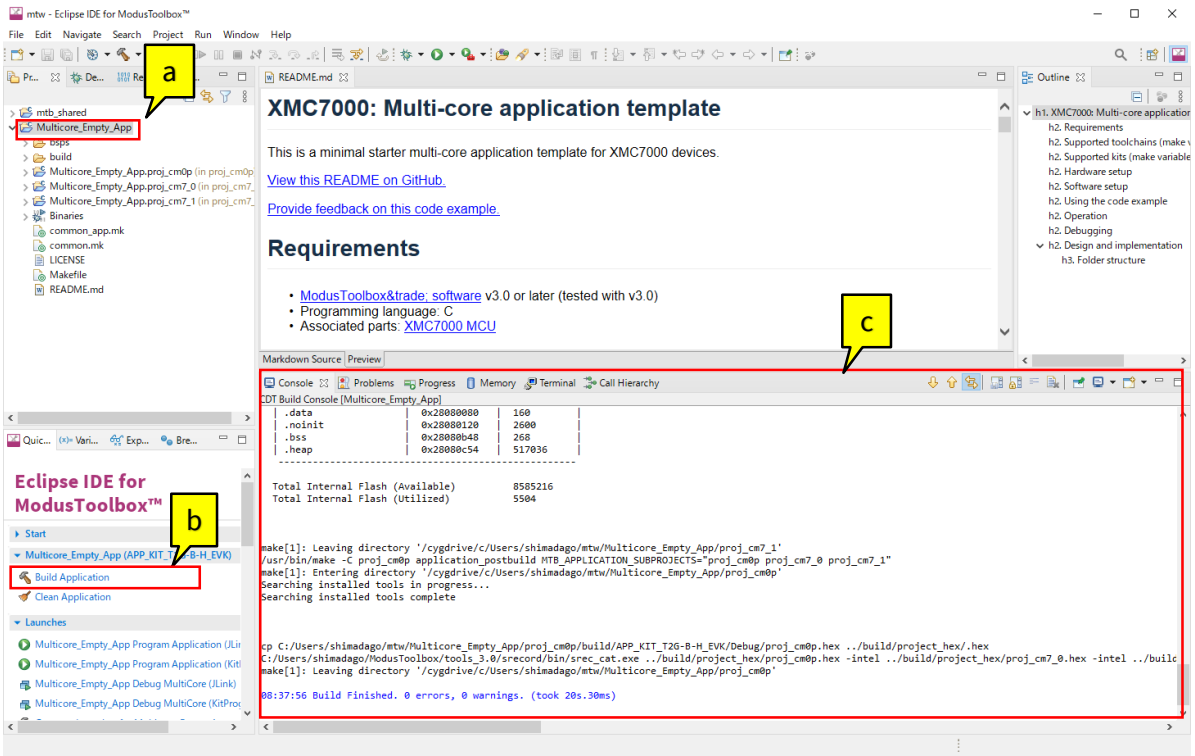
    After a reset, the default core is always the CM0+ core. To enable the CM7 core, CM0+ must call the ***Cy_SysEnableCM7()*** function.

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.

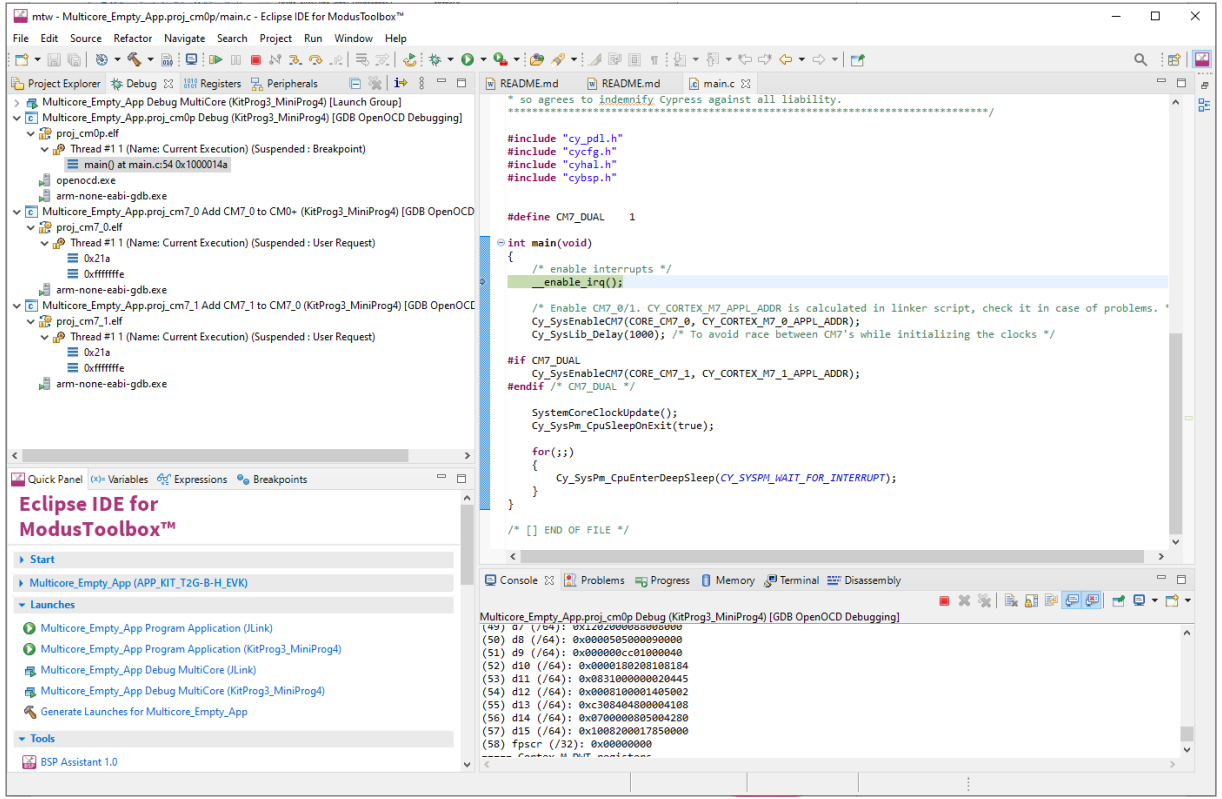
c) The **Console** view lists the results of the build operation



Launch the multi-core application debugging

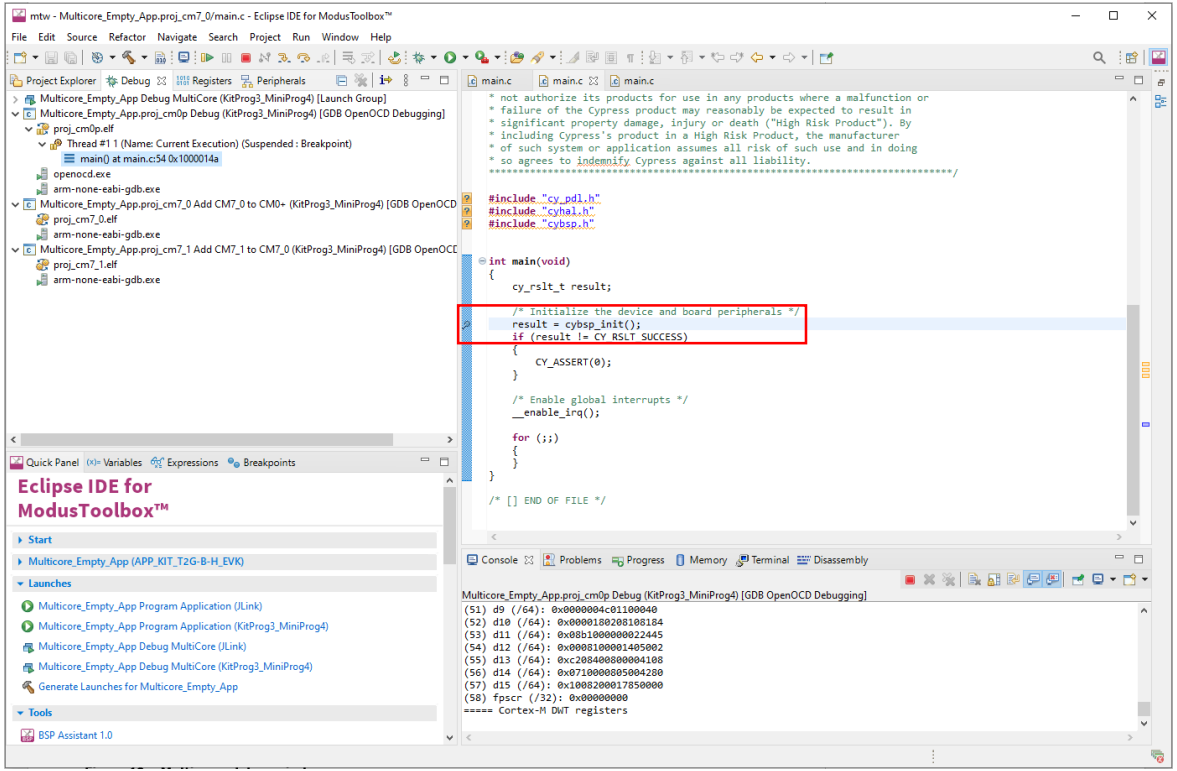
- In the **Quick Panel**, click the **Multicore_Empty_App Debug MultiCore (KitProg3)** link under **Launches.**
  - This will automatically program the CM0P, CM7_0, and CM7_1 code into the flash region of respective cores; then the IDE switches to debug mode automatically.

  - The CM0P 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).



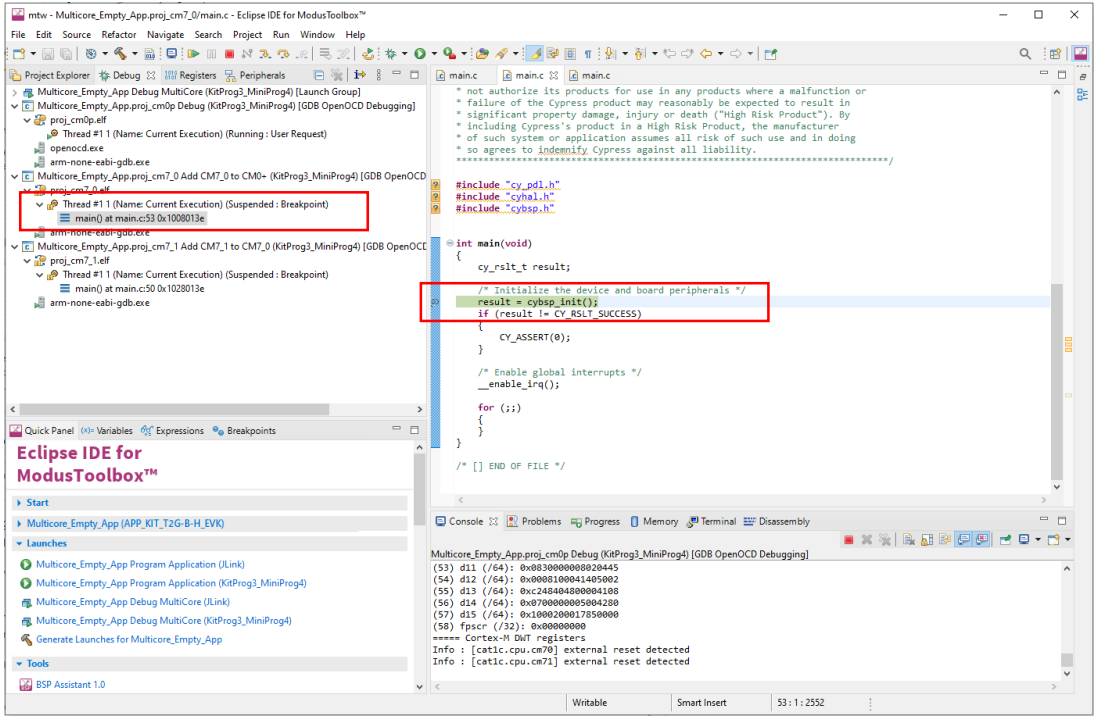
Multi-core application debugging

- 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.



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.



How to modify CM0P and CM7 cores flash and RAM size (1)

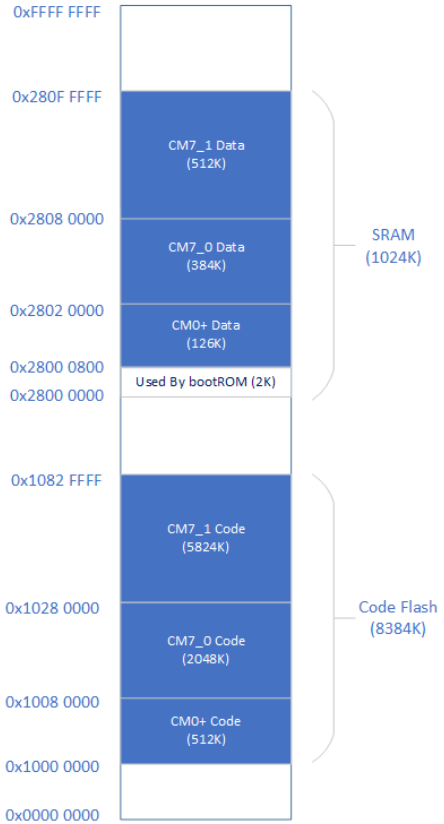
› The picture on the right describes the memory layout of CM0+ and CM7_0/CM7_1 applications for BSP default setting.

› 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_CM0P**\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:

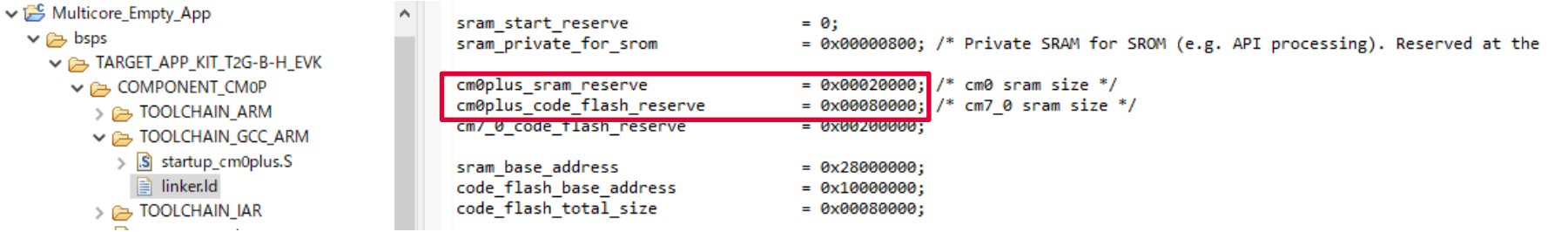
	CM0+	CM7_0	CM7_1
Flash	1024K (0x100000)	4096K (0x400000)	3264K (0x330000)
RAM	256K* (0x40000)	512K (0x80000)	256K (0x40000)

* First 2KB of SRAM is reserved, not available for users.

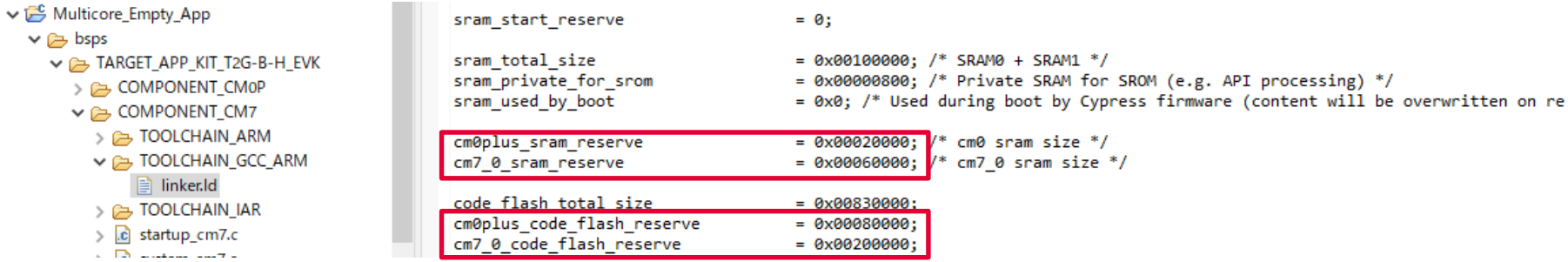


How to modify CM0P and CM7 cores flash and RAM size (2)

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*

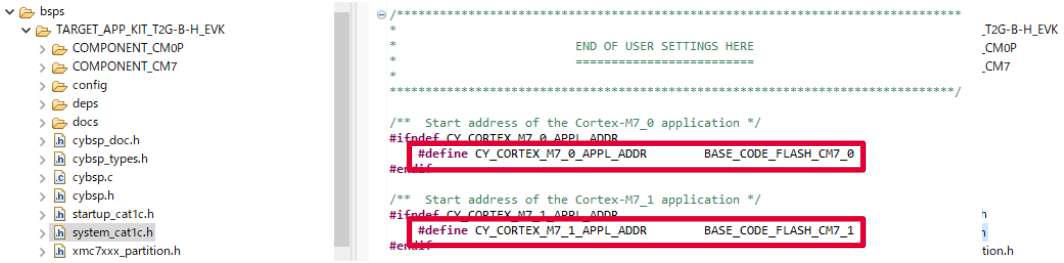


2. Modify CM7 cores flash and RAM size in the *bsps\TARGET_APP_KIT_T2G-B-H_EVK**COMPONENT_CM7***\TOOLCHAIN_GCC_ARM\linker.ld*

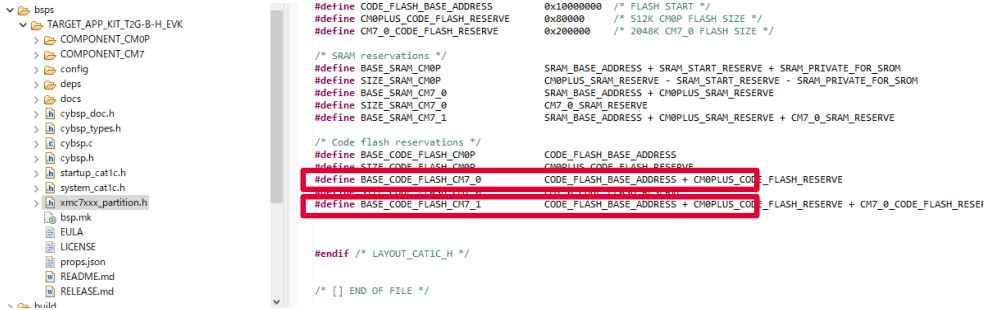


How to modify CM0P and CM7 cores' flash and RAM size (3)

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.



References

Datasheet

› CYT4BF datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family

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

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.

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 such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

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.