TRAVEO™ T2G family AUTOSAR MCAL WDG release notes

SRN223363 version 1.16

About this document

Scope and purpose

Thank you for your interest in the TRAVEO™ T2G family AUTOSAR MCAL WDG driver version 1.16. This document lists the installation requirements, software changes, limitations, and known issues.

Intended audience

This document is intended for anyone who uses the watchdog (WDG) driver of the TRAVEO™ T2G family.

Table of contents

Table of contents

1 System requirements and recommendations

Software prerequisites	Supported version
EB tresos Studio package for Infineon	26.2.0

1.1 Supported compilers

Green Hills Software, compiler v2017.1.4

IAR Embedded Workbench 8.0, EWARM FS 8.22.3

1.2 Compiler options

This section summarizes the compiler options used to build and test the module. When changing the compiler options, the module must be considered untested.

Compiler	Option (Cortex®-M4F core)
Green Hills Software, compiler v2017.1.4	`-cpu=cortexm4f -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fsingle`

Compiler	Option (Cortex®-M7 core)
Green Hills Software, compiler v2017.1.4	`-cpu=cortexm7 -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fhard`

Compiler	Option (Cortex®-M0+ core)
Green Hills Software, compiler v2017.1.4	`-cpu=cortexm0plus -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fsoft`

Compiler	Option (Cortex®-M4F core)
IAR Embedded Workbench 8.0, EWARM FS 8.22.3	`--debug --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4_sp -Ohs --no_size_constraints`

Compiler	Option (Cortex®-M7 core)
IAR Embedded Workbench 8.0, EWARM FS 8.22.3	`--debug --endian=little --cpu=Cortex-M7 -e --fpu=VFPv5_d16 -Ohs --no_size_constraints`

Compiler	Option (Cortex®-M0+ core)
IAR Embedded Workbench 8.0, EWARM FS 8.22.3	`--debug --endian=little --cpu=Cortex-M0+ -e -Ohs --no_size_constraints`

1.3 Library compiler options

If a binary library has been delivered with this module, it has been built using the following options:

Compiler	Option (Cortex®-M4F core)
Green Hills Software, compiler v2017.1.4	`-cpu=cortexm4f -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fsingle`

Compiler	Option (Cortex®-M7 core)
Green Hills Software, compiler v2017.1.4	`-cpu=cortexm7 -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fhard`

Compiler	Option (Cortex®-M0+ core)
Green Hills Software, compiler v2017.1.4	`-cpu=cortexm0plus -thumb -thumb_lib -C99 --short_enum -align4 --no_commons --no_alternative_tokens -asm3g -preprocess_assembly_files -nostartfiles -globalcheck=normal -globalcheck_qualifiers --prototype_errors -Wformat -Wimplicit-int -Wshadow -Wtrigraphs -Wundef -reject_duplicates -c -list -Ospeed -OI -Olink -Ointerproc -Omax -fsoft`

Compiler	Option (Cortex®-M4F core)
IAR Embedded Workbench 8.0, EWARM FS 8.22.3	`--debug --endian=little --cpu=Cortex-M4 -e --fpu=VFPv4_sp -Ohs --no_size_constraints`

Compiler	Option (Cortex®-M7 core)
IAR Embedded Workbench 8.0, EWARM FS 8.22.3	`--debug --endian=little --cpu=Cortex-M7 -e --fpu=VFPv5_d16 -Ohs --no_size_constraints`

Compiler	Option (Cortex®-M0+ core)
IAR Embedded Workbench 8.0, EWARM FS 8.22.3	`--debug --endian=little --cpu=Cortex-M0+ -e -Ohs --no_size_constraints`

1.4 Memory consumption

GHS (Wdg_lib) section	Size (in bytes)
.text	2170
.data	3
.bss	2
Combined	2175

GHS (Wdg_src) section	Size (in bytes)
.text	934
.bss	4
.rodata	116
Combined	1054

IAR (Wdg_lib) section	Size (in bytes)
.text	4942
.bss	1
.data	4
.rodata	4
Combined	4951

IAR (Wdg_src) section	Size (in bytes)
.text	1030
.bss	4
.rodata	108
Combined	1142

Note: *The memory consumption of *_src.lib depends on the configuration.*

Note: *The listed memory consumption will vary depending on customer configuration.*

Explanatory notes for this section

Section	Description
.text	Program code
.data	Variables with explicitly initialized values
.bss	Variables that are not explicitly initialized
.rodata	Read-only data

1.5 Stack consumption

1.5.1 Green Hills Software

Function	Max stack usage (in bytes)
`Wdg_66_IA_Init`	220
`Wdg_66_IA_SetMode`	204
`Wdg_66_IA_SetTriggerCondition`	196
`Wdg_66_IA_GetVersionInfo`	12
`Wdg_66_IA_WarnIntWDT_Cat1`	28
`Wdg_66_IA_WarnIntWDT_Cat2`	28
`Wdg_66_IA_WarnIntMCWDT0_Cat1`	28
`Wdg_66_IA_WarnIntMCWDT0_Cat2`	28

Note: *Stack consumption has been evaluated using the gstack utility program, which is part of the Green Hills release package. To enable the measurement of stack consumption in your project, build the source code according to the instructions given in the "Measuring Stack Consumption" section of the module's user guide.*

Note: *The listed stack consumption will vary depending on customer configuration.*

Note: *The GHS stack consumption listed in the release notes was measured using the additional compile option "`-gs`". The GHS compiler cannot measure stack consumption for the selected optimization level (see compilation options). Green Hills cannot exclude possible effects of "`-gs`" on optimization and stack consumption. Therefore, Infineon cannot guarantee the accuracy of these values. For more information on measuring GHS stack consumption, see the section gstack utility program in Build_arm.pdf.*

1.5.2 IAR Embedded Workbench

Function	Max stack usage (in bytes)
`Wdg_66_IA_Init`	120
`Wdg_66_IA_SetMode`	112
`Wdg_66_IA_SetTriggerCondition`	108
`Wdg_66_IA_GetVersionInfo`	16
`Wdg_66_IA_WarnIntWDT_Cat1`	32
`Wdg_66_IA_WarnIntWDT_Cat2`	32
`Wdg_66_IA_WarnIntMCWDT0_Cat1`	32
`Wdg_66_IA_WarnIntMCWDT0_Cat2`	32

Note: *To enable the measurement of stack consumption in your project, build the source code with the linker option "`--enable_stack_usage --log call_graph`". See stack usage analysis of the IAR C/C++ development guide for details.*

Note: *The listed stack consumption will vary depending on customer configuration.*

1.6 Note on "*_Bswmd.arxml"

Note that the *<Module>_Bswmd.arxml* files are templates that can be freely modified by the customer or RTE vendor.

These are in the *output\generated\swcd* subfolder of your project folder.

Named files are not tested.

1.7 Release details

Module software version
1.16.x (x=software patch version; see the delivery notes for details)

AUTOSAR specification version (ASR)
4.2.2

Target
MXS40

MCAL configuration settings	Supported derivatives
See the resource release notes	See the resource release notes

Corresponding Wdg_66_IA_MemMap.h stub file version
1.1.1

2 Installation

See the installation manual for EB tresos Studio for INFINEON AUTOSAR software products and installation manual for MCAL42-TRAVEO.

3 Deviations from AUTOSAR

T2MC-14562 - [SWS_Wdg_00018] Function definitions : Wdg_SetMode behavior : State control

Title: [SWS_Wdg_00018] Function definitions : `Wdg_SetMode` behavior : State control

Description: [SWS_Wdg_00018] [When development error detection or debugging support is enabled for the Wdg module: The `function Wdg_SetMode` shall set the Wdg module's state to WDG_BUSY during its execution (indicating, that the module is busy) and shall reset the Wdg module's state to WDG_IDLE as last operation before it returns to the caller.] (SRS_BSW_00335)

Reason for rejection: This requirement may cause unexpected behavior because the BUSY status disturbs other service executions. It is difficult to handle it by application. The WDG driver can control hardware even if the BUSY status is not used. This is a reason for rejection.

T2MC-14529 - [SWS_Wdg_00052] Triggering concept to support windowed watchdogs : Wdg module's state

Title: [SWS_Wdg_00052] Triggering concept to support windowed watchdogs : Wdg module's state

Description: [SWS_Wdg_00052] [When development error detection or debugging support is enabled for the Wdg Driver module: the watchdog servicing routine shall set the Wdg module's state to WDG_BUSY during its execution (indicating, that the module is busy) and shall reset the module's state to WDG_IDLE (indicating, that the module is initialized and not busy) as last operation before it returns.] (SRS_BSW_00337)

Reason for rejection: This requirement may cause unexpected behavior because BUSY status disturbs other service executions. It is difficult to handle it by application. WDG driver can control hardware even if BUSY status is not used. This is a reason for rejection.

T2MC-14480 - [SWS_Wdg_00055] External watchdog source code

Title: [SWS_Wdg_00055] External watchdog source code

Description: [SWS_Wdg_00055] [The Wdg module for an external watchdog driver shall have source code that is independent of the microcontroller platform.] ()

Reason for rejection: External watchdog cannot be supported due to driver limitation.

T2MC-14514 - [SWS_Wdg_00076] External watchdog driver : Access the external watchdog hardware

Title: [SWS_Wdg_00076] External watchdog driver : Access the external watchdog hardware

Description: [SWS_Wdg_00076] [To access the external watchdog hardware, the corresponding Wdg module instance shall use the functionality and API of the corresponding handler or driver, e.g. the SPI handler or DIO driver.] (SRS_SPAL_12092)

Reason for rejection: External watchdog cannot be supported due to driver limitation.

T2MC-14516 - [SWS_Wdg_00077] External watchdog driver : External watchdog driver functionality

Title: [SWS_Wdg_00077] External watchdog driver : External watchdog driver functionality

Description: [SWS_Wdg_00077] [A Wdg module for an external watchdog shall satisfy the same functional requirements and offer the same functional scope as a Wdg module for an internal watchdog. Hence their respective APIs are semantically identical.] (SRS_Wdg_12165)

Reason for rejection: External watchdog cannot be supported due to driver limitation.

T2MC-14517 - [SWS_Wdg_00078] External watchdog driver : Additional parameter for external watchdog driver

Title: [SWS_Wdg_00078] External watchdog driver : Additional parameter for external watchdog driver

Description: [SWS_Wdg_00078] [The Wdg module shall add all parameters required for accessing the external watchdog hardware, e.g. the used SPI channel or DIO port, to the module's published parameters and to the module's configuration parameters.] (SRS_Wdg_12166)

Reason for rejection: External watchdog cannot be supported due to driver limitation.

T2MC-14525 - [SWS_Wdg_00093] Triggering concept to support windowed watchdogs : Activation code

Title: [SWS_Wdg_00093] Triggering concept to support windowed watchdogs : Activation code

Description: [SWS_Wdg_00093] [If the watchdog hardware requires an activation code which can be configured or changed, the Wdg Driver shall handle the activation code internally. In this case, the Wdg Driver shall pass the correct activation code to the watchdog hardware and the watchdog hardware in turn shall update the Wdg module's internal variable where the next expected access code is stored.] (SRS_Wdg_12019)

Reason for rejection: This feature (activation code) is not supported by the hardware.

T2MC-14526 - [SWS_Wdg_00094] Triggering concept to support windowed watchdogs : Trigger cycle

Title: [SWS_Wdg_00094] Triggering concept to support windowed watchdogs : Trigger cycle

Description: [SWS_Wdg_00094] [If the watchdog hardware requires an activation code which can be configured or changed, the trigger cycle of the Wdg Driver shall be defined with a value so that updating the activation code by the watchdog hardware can be guaranteed (see **Figure 3**).] (SRS_Wdg_12019)

Reason for rejection: This feature (activation code) is not supported by the hardware.

T2MC-14527 - [SWS_Wdg_00095] Triggering concept to support windowed watchdogs : Configure the activation code

Title: [SWS_Wdg_00095] Triggering concept to support windowed watchdogs : Configure the activation code

Description: [SWS_Wdg_00095] [If the watchdog hardware requires an activation code which can be configured or changed and the initial activation code can be configured, the activation code shall be provided in the Wdg Driver's configuration set. If the activation code is fixed for a particular hardware the above requirement can be ignored.] (SRS_Wdg_12019)

Reason for rejection: This feature (activation code) is not supported by the hardware.

T2MC-14555 - [SWS_Wdg_00145] Function definitions : Wdg_SetMode behavior : Reset the watchdog timeout counter

Title: [SWS_Wdg_00145] Function definitions : `Wdg_SetMode` behavior : Reset the watchdog timeout counter

Description: [SWS_Wdg_00145] [The `Wdg_SetMode` function shall reset the watchdog timeout counter based on the new watchdog mode i.e. the timeout frame remaining shall be recalculated based on a changed trigger period.] ()

Reason for rejection: To avoid complexity, `Wdg_SetMode` resets the watchdog timer counter to 0 and restarts.

T2MC-14531 - [SWS_Wdg_00152] Debugging : Internal state of the module

Title: [SWS_Wdg_00152] Debugging : Internal state of the module

Description: [SWS_Wdg_00152] {OBSOLETE} [The internal state of the module (which indicates whether it is not initialized, idle or busy) shall be available for debugging.] ()

Reason for rejection: Requirements on debugging in AUTOSAR are set to status "obsolete" from this R4.2.2 and removed in R4.3.0.

T2MC-14532 - [SWS_Wdg_00153] Debugging : Internal variable for the watchdog timeout counter

Title: [SWS_Wdg_00153] Debugging : Internal variable for the watchdog timeout counter

Description: [SWS_Wdg_00153] {OBSOLETE} [The internal variable for the watchdog timeout counter shall be available for debugging.] ()

Reason for rejection: Requirements on debugging in AUTOSAR are set to status "obsolete" from this R4.2.2 and removed in R4.3.0.

T2MC-14533 - [SWS_Wdg_00154] Debugging : Internal variable for the watchdog mode

Title: [SWS_Wdg_00154] Debugging : Internal variable for the watchdog mode

Description: [SWS_Wdg_00154] {OBSOLETE} [The internal variable for the watchdog mode shall be available for debugging.] ()

Reason for rejection: Requirements on debugging in AUTOSAR are set to status "obsolete" from this R4.2.2 and removed in R4.3.0.

T2MC-14587 - [SWS_Wdg_00157] Configuration specification: Variants VARIANT-PRE-COMPILE

Title: [SWS_Wdg_00157] Configuration specification: Variants VARIANT-PRE-COMPILE

Description: [SWS_Wdg_00157] [This module shall support the configuration variant VARIANT- PRE-COMPILE. Only parameters with "Pre-compile time" configuration are allowed in this variant.] ()

Reason for rejection: Only post-build time is supported.

T2MC-14588 - [SWS_Wdg_00158] Configuration specification: Variants VARIANT-LINK-TIME

Title: [SWS_Wdg_00158] Configuration specification: Variants VARIANT-LINK-TIME

Description:
[SWS_Wdg_00158] [This module shall support the configuration variant VARIANT- LINK-TIME. Parameters with "Pre-compile time" and "Link time" are allowed in this variant.] ()

Reason for rejection: VARIANT-LINK-TIME is not supported in this product because VARIANT-POST-BUILD is supported and both parameters "Pre-compile time" and "Link time" are allowed in the variant.

T2MC-14515 - [SWS_Wdg_00162] External watchdog driver : Routine servicing an external watchdog

Title: [SWS_Wdg_00162] External watchdog driver : Routine servicing an external watchdog

Description: [SWS_Wdg_00162] [The routine servicing an external watchdog shall be implemented by usage of an own internal hardware timer to be independent from other peripherals or by using a GPT driver callback]

Reason for rejection: External watchdog cannot be supported due to driver limitation.

T2MC-14520 - [SWS_Wdg_00166] Internal watchdog driver : Routine servicing an internal watchdog

Title: [SWS_Wdg_00166] Internal watchdog driver : Routine servicing an internal watchdog

Description: [SWS_Wdg_00166] [The routine servicing an internal watchdog shall be implemented as an interrupt routine driven by a hardware timer] (SRS_BSW_00427, SRS_BSW_00164, SRS_BSW_00325, SRS_BSW_00326, SRS_BSW_00439, SRS_SPAL_12129, SRS_Wdg_12019)

Reason for rejection: WDT and MCWDT support variable servicing cycles. Therefore “decoupling of hardware and logical control” is not necessary. It can be mapped directly. Then there is no issue with jitter or latency.

T2MC-14623 - [SWS_Wdg_00175] Not applicable requirements

Title: [SWS_Wdg_00175] Not applicable requirements

Description: [SWS_Wdg_00175] [These requirements are not applicable to this specification.] (SRS_BSW_00344, SRS_BSW_00404, SRS_BSW_00405, SRS_BSW_00170, SRS_BSW_00419, SRS_BSW_00383, SRS_BSW_00375, SRS_BSW_00416, SRS_BSW_00437, SRS_BSW_00168, SRS_BSW_00423, SRS_BSW_00424, SRS_BSW_00425, SRS_BSW_00428, SRS_BSW_00432, SRS_BSW_00433, SRS_BSW_00450, SRS_BSW_00339, SRS_BSW_00422, SRS_BSW_00417, SRS_BSW_00161, SRS_BSW_00162, SRS_BSW_00005, SRS_BSW_00415, SRS_BSW_00007, SRS_BSW_00413, SRS_BSW_00441, SRS_BSW_00307, SRS_BSW_00373, SRS_BSW_00410, SRS_BSW_00447, SRS_BSW_00348, SRS_BSW_00353, SRS_BSW_00361, SRS_BSW_00302, SRS_BSW_00328, SRS_BSW_00312, SRS_BSW_00006, SRS_BSW_00449, SRS_BSW_00377, SRS_BSW_00304, SRS_BSW_00378, SRS_BSW_00306, SRS_BSW_00308, SRS_BSW_00309, SRS_BSW_00371, SRS_BSW_00359, SRS_BSW_00360, SRS_BSW_00440, SRS_BSW_00330, SRS_BSW_00009, SRS_BSW_00401, SRS_BSW_00172, SRS_BSW_00010, SRS_BSW_00333, SRS_BSW_00321, SRS_BSW_00341, SRS_BSW_00334, SRS_SPAL_12056, SRS_SPAL_12267, SRS_SPAL_12462, SRS_SPAL_12463, SRS_SPAL_12068, SRS_SPAL_12069, SRS_SPAL_00157, SRS_SPAL_12063, SRS_SPAL_12075, SRS_SPAL_12067, SRS_SPAL_12077, SRS_SPAL_12078, SRS_SPAL_12265, SRS_Wdg_12167, SRS_Wdg_12168)

Reason for rejection: Named RQMs are not applicable.

---

4 Limitations

T2MC-14616 - [ECUC_Wdg_00112] Configuration specification: WdgExternalConfiguration

Title: [ECUC_Wdg_00112] Configuration specification: `WdgExternalConfiguration`

Description:

SWS Item	ECUC_Wdg_00112 :
Container Name	`WdgExternalConfiguration`
Description	Configuration items for an external watchdog hardware
Configuration Parameters	

Limitation: External watchdog cannot be supported due to driver limitation.

T2MC-14617 - [ECUC_Wdg_00113] Configuration specification: WdgExternalConfiguration WdgExternalContainerRef

Title: [ECUC_Wdg_00113] Configuration specification: WdgExternalConfiguration `WdgExternalContainerRef`

Description:

SWS Item	ECUC_Wdg_00113 :		
Name	`WdgExternalContainerRef`		
Description	Reference to either - a `DioChannelGroup` container in case the hardware watchdog is connected via DIO pins - an `SpiSequenceConfiguration` container in case the watchdog hardware is accessed via SPI.		
Multiplicity	0..1		
Type	Choice reference to [DioChannelGroup , SpiSequence].		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	***Pre-compile time***	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: See DIO resp. SPI SWS.		

Limitation: External watchdog cannot be supported due to driver limitation.

T2MC-14599 - [ECUC_Wdg_00116] Configuration specification: WdgGeneral WdgDisableAllowed

Title: [ECUC_Wdg_00116] Configuration specification: WdgGeneral `WdgDisableAllowed`

Description:

SWS Item	ECUC_Wdg_00116 :		
Name	`WdgDisableAllowed`		
Description	Compile switch to allow / forbid disabling the watchdog driver during runtime. True: Disabling the watchdog driver at runtime is allowed. False: Disabling the watchdog driver at runtime is not allowed.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local dependency: Safety relevant compile switch; this has to be in accordance with the corresponding settings for the watchdog manager.		

Limitation: WDG functionality is not allowed to be disabled from functional safety point of view.

T2MC-14601 - [ECUC_Wdg_00130] Configuration specification: WdgGeneral WdgInitialTimeout

Title: [ECUC_Wdg_00130] Configuration specification: WdgGeneral `WdgInitialTimeout`

Description:

SWS Item	ECUC_Wdg_00130 :		
Name	`WdgInitialTimeout`		
Description	The initial timeout (sec) for the trigger condition to be initialized during Init function. It shall be not larger than `WdgMaxTimeout`.		
Multiplicity	1		
Type	`EcucFloatParamDef`		
Range	0..65.535		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

Limitation: This parameter is not used. `WdgFastTimeoutValue` and `WdgSlowTimeoutValue` are used instead of this parameter.

T2MC-14561 - [SWS_Wdg_00017] Function definitions: Wdg_SetMode behavior : Raise development error WDG_E_DRIVER_STATE

Title: [SWS_Wdg_00017] Function definitions: `Wdg_SetMode` behavior : Raise development error WDG_E_DRIVER_STATE

Description: [SWS_Wdg_00017] [When development error detection is enabled for the Wdg module: The `Wdg_SetMode` function shall check that the Wdg module's state is WDG_IDLE (meaning the Wdg module and the watchdog hardware are initialized and the watchdog is currently not being triggered or switched). If this is not the case, the function shall not execute the mode switch but raise the development error WDG_E_DRIVER_STATE and return with the value E_NOT_OK.] (SRS_BSW_00335, SRS_SPAL_12064, SRS_SPAL_12448)

Limitation: DET error detection mechanism is used as safety mechanism (fault detection), so detection of development errors cannot be disabled.

T2MC-14550 - [SWS_Wdg_00019] Function definitions: Wdg_Init behavior : State control

Title: [SWS_Wdg_00019] Function definitions: `Wdg_Init` behavior : State control

Description: [SWS_Wdg_00019] [When development error detection or debugging support is enabled for the Wdg module: The `Wdg_Init` function shall set the Wdg module's internal state from WDG_UNINIT (the default state indicating a non-initialized module) to WDG_IDLE if the initialization was successful.] (SRS_BSW_00406, SRS_BSW_00335)

Limitation: DET error detection mechanism is used as safety mechanism (fault detection), so detection of development errors cannot be disabled. And debug function is not supported.

T2MC-14549 - [SWS_Wdg_00090] Function definitions: Wdg_Init behavior : Raise the development error WDG_E_PARAM_CONFIG.

Title: [SWS_Wdg_00090] Function definitions: Wdg_Init behavior : Raise the development error WDG_E_PARAM_CONFIG.

Description: [SWS_Wdg_00090] [When development error detection is enabled for the Wdg module: The Wdg_Init function shall check that the (hardware specific) contents of the given configuration set is within the allowed boundaries. If this error is detected, the function Wdg_Init shall not execute the initialization but raise the extended error WDG_E_PARAM_CONFIG.] (SRS_BSW_00323, SRS_SPAL_12448)

Limitation: DET error detection mechanism is used as safety mechanism (fault detection), so detection of development errors cannot be disabled.

T2MC-14559 - [SWS_Wdg_00091] Function definitions: Wdg_SetMode behavior : Raise development error WDG_E_PARAM_MODE(out of range)

Title: [SWS_Wdg_00091] Function definitions: `Wdg_SetMode` behavior : Raise development error WDG_E_PARAM_MODE(out of range)

Description: [SWS_Wdg_00091] [When development error detection is enabled for the Wdg module: The `Wdg_SetMode` function shall check that the parameter Mode is within the allowed range. If this is not the case, the function shall not execute the mode switch but raise development error WDG_E_PARAM_MODE and return with the value E_NOT_OK] (SRS_BSW_00323, SRS_SPAL_12448)

Limitation: DET error detection mechanism is used as safety mechanism (fault detection), so detection of development errors cannot be disabled.

T2MC-14560 - [SWS_Wdg_00092] Function definitions: Wdg_SetMode behavior : Raise development error WDG_E_PARAM_MODE(not allowed mode)

Title: [SWS_Wdg_00092] Function definitions: `Wdg_SetMode` behavior : Raise development error WDG_E_PARAM_MODE(not allowed mode)

Description: [SWS_Wdg_00092] [When development error detection is enabled for the Wdg module: The `Wdg_SetMode` function shall check that the (hardware specific) settings for the requested mode are within the allowed boundaries. If this is not the case, the function shall not execute the mode switch but raise the development error WDG_E_PARAM_MODE and return with the value E_NOT_OK.] (SRS_BSW_00323, SRS_SPAL_12448)

Limitation: DET error detection mechanism is used as safety mechanism (fault detection), so detection of development errors cannot be disabled.

T2MC-14569 - [SWS_Wdg_00146] Function definitions: Wdg_SetTriggerCondition behavior : Raise development error WDG_E_PARAM_TIMEOUT

Title: [SWS_Wdg_00146] Function definitions: `Wdg_SetTriggerCondition` behavior : Raise development error WDG_E_PARAM_TIMEOUT

Description: [SWS_Wdg_00146] [When development error detection is enabled for the module: The function `Wdg_SetTriggerCondition` shall check that the timeout parameter given is less or equal to the maximum timeout value (WdgMaxTimeout). If this is not the case the function shall not reload the timeout counter but raise the development error WDG_E_PARAM_TIMEOUT and return to the caller.] ()

Limitation: DET error detection mechanism is used as safety mechanism (fault detection), so detection of development errors cannot be disabled.

T2MC-14589 - [SWS_Wdg_00159] Configuration specification: Variants VARIANT-POST-BUILD

Title: [SWS_Wdg_00159] Configuration specification: Variants VARIANT-POST-BUILD

Description: [SWS_Wdg_00159] [This module shall support the configuration variant VARIANT- POST-BUILD. Parameters with "Pre-compile time", "Link time" and "Post-build time" are allowed in this variant.] ()

Limitation: WDG driver does not provide link time configuration.

T2MC-14572 - [SWS_Wdg_00174] Function definitions : Wdg_GetVersionInfo behavior : Raise development error WDG_E_PARAM_POINTER

Title: [SWS_Wdg_00174] Function definitions : `Wdg_GetVersionInfo` behavior : Raise development error WDG_E_PARAM_POINTER

Description: [SWS_Wdg_00174] [If development error detection is enabled for the Wdg Driver module, the function `Wdg_GetVersionInfo` shall raise WDG_E_PARAM_POINTER, if the argument is a NULL pointer and return without any action.] ()

Limitation: DET error detection mechanism is used as safety mechanism (fault detection), so detection of development errors cannot be disabled.

T2MC-17865 - [WDG] AUTOSAR C implementation rules

Title: [WDG] AUTOSAR C implementation rules

Description: The MCAL modules shall fulfill all design and implementation guidelines as described in specification of C implementation rules AUTOSAR_TR_CImplementationRules.pdf.

Limitation: Out of scope: keyword macros 'CONST' and 'VAR' are not required for declaration/definition of the local variable, function parameter, and structure/union fields.

5 Known defects

The listed issues were known at the day this release note was generated. Further problems may have been discovered in the meantime. For an up-to-date list of known issues, contact your Infineon sales representative.

This release has no known issues at the time of release.

6 Documentation

All user guides for MCAL drivers are in the *doc* subdirectory of the *installation* directory. The default location is:

C:\INFINEON_ESDB\Tresos26_2_0\doc

7 Technical support

If you have questions related to the driver, contact the local support application engineer.

8 Version history

8.1 Module SW-Version 1.4

Initial module setup.

8.2 Module SW-Version 1.5

T2MC-39177 - [WDG] CONFIG register is not initialized when WdgDefaultMode is WDGIF_OFF_MODE

Title: [WDG] CONFIG register is not initialized when WdgDefaultMode is WDGIF_OFF_MODE

Description:

1. If configuration "WdgDefaultMode" is WDGIF_OFF_MODE, CONFIG register will not be initialized. When mode is set to fast or slow by Wdg_66_IA_SetMode(), CONFIG register (LOWER_ACTION, UPPER_ACTION...) will not be set and HW default action will be triggered.
2. When mode is set to WDGIF_OFF_MODE by Wdg_66_IA_SetMode(), INTR_MASK register is not cleared.
3. The generated value of Wait1usCount is wrong.

T2MC-38074 - File extension should be changed from .bmd to .arxml

Title: File extension should be changed from .bmd to .arxml

Description: The file extension should be changed from *.bmd to *.arxml.
Each module still has an *autosar/<module>.bmd* file.

8.3 Module SW-Version 1.6

T2MC-39747 - [All] Checking for valid C function name and including filename in configuration parameters

Title: [All] Checking for valid C function name and including filename in configuration parameters

Description: Checking for valid C function name:
Check all configuration parameters related to the function name to see if it is a valid C function name.
A part of parameters are not checked.
If an invalid function name is set, a compile error will occur during the build process, which is inconvenient for users.
Therefore, it is better to check whether the configured function names are valid C function names in advance (i.e. during configuration phase).

Checking for valid filename:
Check all configuration parameters related to the file name to see if it is valid.
A part of parameters cannot check the fact that empty file name (i.e. ".h") is wrong.
If an invalid file name is set, a compile error will occur during the build process, which is inconvenient for users.
Therefore, it is better, to check in advance, whether the configured file names are valid.

This CR is intended to solve the inconvenience.

T2MC-41850 - [General] <CODE-DESCRIPTORS> Node should be added to the *arxml* files of all modules

Title: [General] <CODE-DESCRIPTORS> Node should be added to the *arxml* files of all modules

Description: For all modules, the <CODE-DESCRIPTORS> Node needs to be added for the RTE within the BSWMD *arxml* file.

T2MC-40977 - [WDG] MCDC coverage does not reach 100%

Title: [WDG] MCDC coverage does not reach 100%

Description: MCDC coverage does not reach 100%.

Following is the reason.

WDG_66_IA_POSSIBLE_MODES_ARCH is used to detect whether the selected mode is valid, but all modes are determined to be valid, so the check result would be always "true".
Therefore, "false" condition cannot be realized. This check must be deleted.

T2MC-46441 - [Wdg] Wdg_66_IA_SetTriggerCondition fail to change trigger period

Title: [Wdg] Wdg_66_IA_SetTriggerCondition fail to change trigger period

Description: Wdg_66_IA_SetTriggerCondition will fail in some cases as below.

1. Running with "Fast" mode -> change the mode to "Slow" -> call Wdg_66_IA_SetTriggerCondition with timeout value which is same as "Fast" mode.
2. Running with "Fast" mode -> call Wdg_66_IA_SetTriggerCondition with different timeout value "A" -> change the mode to "Slow" -> change the mode to "Fast" -> call Wdg_66_IA_SetTriggerCondition with timeout value "A"

8.4 Module SW-Version 1.7

T2MC-39589 - [WDG] Support TRAVEO™ T2G-B-H-8M

Title: [WDG] Support TRAVEO™ T2G-B-H-8M
Description: AUTOSAR MCAL supports the TRAVEO™ T2G-B-H-8M.

Regarding WDG, following will be changed.

- MCWDT's functionalities must be supported by WDG driver.
- WDT needs to be disabled when it is not used because its default status is ENABLED.

In addition to the above, users guide needs update.

T2MC-50612 - [General] Delete device-dependent information from the user guide

Title: [General] Delete device-dependent information from the user guide

Description: Any device-dependent information should not be included in the user guide.
Therefore, delete the datasheet name from the related documentation in the user guide.

T2MC-56922 - [Wdg] “WdgFast/SlowTimeoutValue” is set in case it is bigger than max value

Title: [Wdg] “WdgFast/SlowTimeoutValue” is set in case it is bigger than max value

Description: "WdgFast/SlowTimeoutValue is set to HW without detection of error even though the value is bigger than maximum timeout value.

This problem occurs in Wdg_66_IA_Init() and Wdg_66_IA_SetMode().

There are two types of maximum timeout value:

- WdgMaxTimeout: configuration parameter which can be set by user
- HW specific max timeout: In case MCWDT, it has 16-bit counter. Then the timeout value is limited to 0 - 0xFFFF (1999ms).

This problem causes two illegal behaviors:

1. Unexpected value exceeding WdgMaxTimeout will be set to HW.
2. Unexpected value exceeding 0xFFFF will be set to MCWDT.
   For example, if 2000ms is requested, WDG driver sets 0x10000 to HW and the HW recognizes it as 0x0000.

T2MC-55592 - [WDG] Problem in WDG BSWMD file

Title: [WDG] Problem in WDG BSWMD file

Description: *Wdg_Bswmd.arxml* does not generate IRQs correctly.

T2MC-52871 - WDG: Build fail of assembler code for Cortex®-0+

Title: WDG: Build fail of assembler code for Cortex®-0+

Description: Wdg driver has an assembler code and its build with options for Cortex®-0+ fails.

It comes from differences of instruction set.

In addition, a warning message appears in Wdg_66_IA_HwAccess_*Hwdg.c*.

This also should be disappeared.

8.5 Module SW-Version 1.8

T2MC-59614 - [WDG] ECUC-MULTIPLICITY-CONFIGURATION-CLASS definition is missing

Title: [WDG] ECUC-MULTIPLICITY-CONFIGURATION-CLASS definition is missing

Description: The following messages should be solved if they are detected by new AMDC tool version 1.0.17:

A container has no 'ECUC-MULTIPLICITY-CONFIGURATION-CLASS' elements. (warning of rule A205)

A parameter has no 'ECUC-MULTIPLICITY-CONFIGURATION-CLASS' elements. (warning of rule A205)

T2MC-63194 - [WDG] Wdg_66_IA_Init fail to check the status of WDT when MCWDT is configured and WdgStopWDT = false

Title: [WDG] Wdg_66_IA_Init fail to check the status of WDT when MCWDT is configured and WdgStopWDT = false

Description: WDG driver checks the statuses of MCWDT and WDT.
When only MCWDT is configured, expected MCWDT's status is decided based on the mode but expected WDT's status is always "DISABLED" in this case.
WdgStopWDT is used to disable WDT by WDG driver.
When this parameter is "true", WDG driver must check WDT's status is "DISABLED".
Otherwise, WDG driver must not check WDT's status because it is expected that another WDG driver uses WDT.

T2MC-61740 - [WDG] Word alignment for literal pool

Title: [WDG] Word alignment for literal pool

Description: Following instructions in *Wdg_66_IA_Trigger_Asm_GHS.s* file might load wrong address:
LDR r2, =#0x40268000
LDR r2, =#0x4026c000

8.6 Module SW-Version 1.9

T2MC-65901 - [WDG] Basic WDT registers are not initialized by reset of MCWDT

Title: [WDG] Basic WDT registers are not initialized by reset of MCWDT

Description: WDT reset happens when MCWDT causes a reset.

It also triggers unexpected rapid interruption.

MCAL Wdg module cannot pass CYT4BF8CES-Wdg-008 by the CDT.

Workaround:

Initialize WDT registers by default value at the top of application program.

Limitation:

When WDT and MCWDT is running, this problem will occur.

There is no problem if only WDT or only MCWDT is running.

T2MC-68246 - [WDG] Rejected requirement is missing in Wdg release notes

Title: [WDG] Rejected requirement is missing in Wdg release notes

Description: [SWS_Wdg_00166] Internal watchdog driver: Routine servicing an internal watchdog

This AUTOSAR requirement is rejected but it is not in the Wdg release notes.

To fix this issue, the visibility field must be set correctly.

T2MC-66351 - [WDG] warning on build when no interrupt is configured

Title: [WDG] warning on build when no interrupt is configured

Description: A warning is detected during the Wdg module build if the notification function is not configured.

There is no effect on the Wdg driver's behavior.

T2MC-65914 - [WDG] WDT generates unexpected interrupt during the disabled status

Title: [WDG] WDT generates unexpected interrupt during the disabled status

Description: When the default mode is "OffMode", and changed to "FastMode" or "SlowMode", WDT generates an interrupt before the counter is running (disabled).

It may cause LOWER_LIMIT violation by servicing after detecting the interrupt.

Workaround:

In case WDT is used and "OffMode" is set as the default mode, this issue occurs.

You can use any of the following three workarounds.

- Only MCWDT is used.
- Set "OffMode" as the default mode, and do not change to another mode.
- Set "FastMode" or "SlowMode" as the default mode. In this case, you can change the mode to another mode.

T2MC-77594 - Support IAR compiler

Title: Support IAR compiler

Description: Support IAR compiler (IAR EWARM FS 8.22.3.15992).

8.7 Module SW-Version 1.10

T2MC-84061 - [WDG] Typo in XDM error message

Title: [WDG] Typo in XDM error message

Description: There are several error messages such as:
"WdgXXX has not enough chilren! [...]"
"chilren" should be "children".

T2MC-50674 - WDG: Reset by lower limit violation

Title: WDG: Reset by lower limit violation

Description: HW errata of errata ID #51 (CDT-314555),

The MCAL WDG driver encounters a reset when running changes of “LOWER_LIMIT” register (> CNT register).

To avoid this reset, the MCAL WDG driver has already implemented a workaround code since MCAL V1e.1.1.

Hardware will be fixed for TRAVEO™ T2G-C-2D, TRAVEO™ T2G-B-H-8M based on HW IP SRSS version 3, after which the workaround code will be removed from the WDG driver.

(Note: TRAVEO™ T2G-B-E-1M/2M use version 2 of SRSS IP)

Workaround:

The MCAL WDG driver has already implemented the following workaround (no action is required from the user).

Wait for the SERVICE register status before changing the values of Lower/Upper/Warn limits.

It takes up to three “lf_clk” cycles (assumed max 91.5 µs).

8.8 Module SW-Version 1.11

T2MC-90368 - [WDG] Generate error when multiple Wdg drivers are configured in the same project

Title: [WDG] Generate error when multiple Wdg drivers are configured in the same project

Description: The following line used in the Infineon Wdg driver causes a generation error.
SELECT "as:modconf('Wdg')[1]

It implies that the generators will use the first element in the list of Wdg configurations in the project.

However, if there are other WDGs in the same project, the above code will refer to wrong WDG configuration.

T2MC-91257 - [WDG] Wdg.xdm is inconsistent with Wdg.arxml

Title: [WDG] Wdg.xdm is inconsistent with Wdg.arxml

Description: There are some inconsistent descriptions in *Wdg.xdm* and *Wdg.arxml*.

- Some of CONFIG-CLASS in *Wdg.arxml* are inconsistent with *Wdg.xdm*
- Some of MULTIPLICITY-CONFIGURATION-CLASS in both files are incorrect

Wdg.arxml: incorrect LOWER_MULTIPLICITY and lack of MULTIPLICITY-CONFIGURATION-CLASS

Wdg.xdm: lack of <icc:v mclass description

- *Wdg.xdm* has unnecessary SUPPORTED-CONFIG-VARIANT (VariantLinkTime, VariantPreCompile)
- Incorrect reference path in *Wdg.xdm*
- There are some typos in both files

8.9 Module SW-Version 1.12

T2MC-97382 - Macro definition at variable declaration is missing and the limitation is not mentioned in release notes

Title: Macro definition at variable declaration is missing and the limitation is not mentioned in release notes

Description: Macro definitions are not used when declaring some variables and pointers (in FLS, MCU, PORT, SPI, and WDG).

According to AUTOSAR specification:
[SWS_COMPILER_00026]
#define VAR(vartype, memclass)

True:
volatile P2VAR(Spi_DmaChannelRegsType, AUTOMATIC, REGSPACE) retPtr;

False:
volatile Spi_DmaChannelRegsType * retPtr;

This issue is present in the following cases:

- All types of pointer declaration/definition are defined without macros.

These contain the function parameter/global variable/local variable/structure field/union field.

- All types of function declaration/definition are defined without macros.
- When there is nested macro usage in function macros.
- Raw pointer is used in the function macro:

e.g., FUNC(int *, memclass) function(void);

- Global variable or static variable in the function is not defined with macros.

To fully comply with the above cases, change variable and function definitions in FLS, MCU, PORT, SPI, and WDG.

In requirements, keyword macros 'CONST' and 'VAR' are not required for declaration/definition of the local variable, function parameter, and structure/union fields.

The information must be described in all release notes.

T2MC-39519 - Support EB tresos V26.2.0

Title: Support EB tresos V26.2.0

Description: Support EB tresos V26.2.0
[Impact]

Strict AUTOSAR specification and check for parameter configuration errors are implemented in EB tresos V26.2.0.

In addition, handling of reference paths (relative paths) such as system description file (ARXML) is changed in EB tresos V26.2.0.

Therefore, if the current ECUC configuration definitions XML file contains deviations or errors, you may find errors during import to tresos26. In that case, the ECUC configuration definitions XML file must be modified appropriately.

In addition, if the current ARXML file contains unresolvable paths, you may find errors during import to tresos26. In that case, ARXML file must be modified.

The SW has been tested; no risks except for the low-level risk listed above were found.

T2MC-163914 - [WDG] Incorrect range of WdgCPUSelect

Title: [WDG] Incorrect range of WdgCPUSelect

Description: The range of WdgCPUSelect in the user guide and the design document is incorrect.

0 to 3 is correct range but those documents state as 0 to 4.

8.10 Module SW-Version 1.13

T2MC-164408 - Improvement of interrupt register clear processing

Title: Improvement of interrupt register clear processing

Description: Some modules clear the interrupt register by read modify write (RMW). However, there is a possibility that unintended bits might also be cleared, if some bits are already set before clearing, because the attribute of the interrupt register is RW1C (every bit is cleared upon writing 1).

Also, unnecessary read access to the register reduces performance.

Therefore, change the clearing process to write intended bit only.

T2MC-164778 - Support MISRA C:2012 coding rule

Title: Support MISRA C:2012 coding rule

Description: Support MISRA C:2012 coding rule.

The MISRA C:2012 coding rule checks the source code.

If a deviation from the rules is required, add the deviation comment to the code and report the result.

If a deviation is for MISRA-C:2004 only, remove the deviation comment.

8.11 Module SW-Version 1.14

T2MC-164831 - [ALL] Misleading comment in Module_MemMap.h

Title: [ALL] Misleading comment in Module_MemMap.h

Description: *{Mip}_MemMap.h* files are provided as sample template files. But, the file header comment cannot be modified, which is a contradiction. To resolve this contradiction, change the file header comment to allow user modification.

Also, to make sure that the file is not a part of the commercial product, move the *{Mip}_MemMap.h* files to the *MemMap* stub folder.

T2MC-165895 - [WDG] Unused variable existence

Title: [WDG] Unused variable existence

Description: As the result of horizontal expansion of AAPS-135 (related AI: T2MC-97162), there is an unused variable “Wdg_66_IA_TriggerFuncPtr” in WDG code.

When WdgRunArea=RAM is configured, the pointer to the watchdog trigger routine on RAM area is set to Wdg_66_IA_TriggerFuncPtr but this variable is never used.

8.12 Module SW-Version 1.15

T2MC-167142 - [WDG] Incorrect declaration of constant value

Title: [WDG] Incorrect declaration of constant value

Description: Wdg_66_IA_TriggerFuncSize is defined as a read-only value in Wdg_66_IA_Trigger_Asm_GHS.s and Wdg_66_IA_Trigger_Asm_IAR.s.

But, it is declared as a variable in Wdg_66_IA_Types.h as shown below.extern VAR(uint32, WDG_VAR_NO_INIT) Wdg_66_IA_TriggerFuncSize;It should be changed to a constant value, as shown below, to maintain consistency:.extern CONST(uint32, WDG_APPL_CONST) Wdg_66_IA_TriggerFuncSize;

If the Wdg driver tries to write a value to this variable, an error would happen.

However, there is no code to change this variable. Therefore, there is no possibility of the current Wdg code causing an error.

Following is supported in release V1.9.0.

T2MC-170544 - [WDG] Unclear description of error condition of WDG_66_IA_E_PARAM_MODE in user guide

Title: [WDG] Unclear description of error condition of WDG_66_IA_E_PARAM_MODE in user guide

Description: The WDG driver reports WDG_66_IA_E_PARAM_MODE to DET if the target mode is not in the list of supported modes which is held in the WDG driver. However, this error condition is described as below in the user guide. It would make user confused.

"If the new mode is not supported by hardware, ..."

Therefore, description of this error should be corrected.

Following is supported in release V1.12.0.

T2MC-178688 - Addition of the notice for Arm® errata and workaround in the user guide

Title: Addition of the notice for Arm® errata and workaround in the user guide

Description: Add a notice for Arm® Cortex®-M4 errata 838869 and software workaround in the user guide.

8.13 Module SW-Version 1.16

T2MC-183981 - [WDG] Improvement of interrupt handler to prevent spurious interrupts

Title: [WDG] Improvement of interrupt handler to prevent spurious interrupts

Description: INTR is recommended to be read back to ensure the update in the register immediately. (This recommendation is from application note AN224432 *E, which was recently updated.) This is because there is a possibility that a spurious interrupt occurs when leaving an interrupt handler after clearing the INTR without register readback.

Currently, there is no INTR readback operation in the WDG interrupt handler. If a spurious interrupt occurs, the WDG module calls the notification function.

This can be resolved by adding a readback operation to ensure that the write buffer is drained. Also, if there is no interrupt factor, it is necessary to exit the interrupt handler without calling the notification function.

T2MC-183983 - Update copyright notice and disclaimer statement

Title: Update copyright notice and disclaimer statement

Description: Copyright notice and disclaimer statement in the file header comment are updated to follow the up-to-date specifications.

T2MC-184009 - [WDG] Add a note of caution in the user guide that servicing the WDT after DeepSleep may be ignored

Title: [WDG] Add a note of caution in the user guide that servicing the WDT after DeepSleep may be ignored

Description: HW erratum "129. WDT service can be missed" has impact on the WDG/WDGMC driver.
This erratum describes that "If a servicing of the WDT occurs within 4 ILO clock cycles before DeepSleep entry, it clears the counter but does not fully complete an internal handshake. A servicing of the WDT after DeepSleep wakeup may then be missed if it occurs less than 2 ILO clock cycles after the processor resumes clocking."
If this error condition occurs, the WDT may cause an undesired WARN_ACTION or UPPER_ACTION.

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

Edition 2022-07-12

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Go to www.infineon.com/support

Document reference

002-23363 Rev. *N

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.