Safe Watchdog Manager
Safety Manual

Author:	TTTech Automotive GmbH
Security:	Company Confidential
Document number:	D-SAFEX-S-70-001
Version:	2.3.28
Date:	26.05.2014
Status:	ALM_Published
MKS ID:	228403

TTTech Automotive GmbH
Schoenbrunner Str. 7, A-1040 Vienna, Austria, Tel. + 43 1 585 34 34-0, Fax +43 1 585 34 34-90, office@tttech-automotive.com

Revision History

17.06.2011 V0.9.4 Draft
15.07.2011 V1.0.0 Safe Watchdog Manager Series Release
12.08.2011 V1.1.0 Safe Watchdog Manager Series Release
07.09.2011 V1.2.0 Safe Watchdog Manager Series Release
16.09.2011 V1.3.0 Safe Watchdog Manager Series Release
16.11.2011 V1.3.1 Safe Watchdog Manager - separating S-Wdg drivers
13.12.2011 V1.4.0 Safe Watchdog Manager - Series Release
10.02.2012 V1.5.0 Safe Watchdog Manager - Series Release
17.02.2012 V1.5.1 Safe Watchdog Manager - Series Release (Patch Release)
09.03.2012 V1.6.0 Safe Watchdog Manager - Series Release
13.04.2012 V1.7.0 Safe Watchdog Manager - Series Release
08.05.2012 V1.7.1 Safe Watchdog Manager
05.05.2012 V2.0.0 Copied from MKS 64019 to MKS 228043. Hierarchie restructured. Labeled for review.
24.05.2012 V2.0.1 Labeled for review.
25.05.2012 V2.0.2 Safe Watchdog Manager - Series Release V1.8.0
27.06.2012 V2.1.0 Reviewed. Some information still open
03.07.2012 V2.2.0 Added config generation and verification process
03.07.2012 V2.2.1 Added timing constraints (issue47259)
05.07.2012 V2.3.0 Added requirements from ETA and Check against System Specification
06.07.2012 V2.3.1 Ready for Release 1.8.2
07.08.2012 V2.3.2 Feedback from Hella-Audit, some texts more precise
23.08.2012 V2.3.3 added system assumptions, S-WdgM requ., AUTOSAR 3.1 info, manual checks
10.09.2012 V2.3.4 Traced requirements from ETA. Dissolved section "Requirements derived from ETA process"
13.09.2012 V2.3.5 After walkthrough review
13.09.2012 V2.3.6 Added manual tests
15.09.2012 V2.3.7 Safe Watchdog Manager ASIL Release
15.10.2012 V2.3.8 Added system assumption regarding critical sections (297946,297948), issue49890
Added reentrancy, issue49459 (WDGM_E_REENTRANCY)
05.12.2012 V2.3.9 228523 - Added Safety Manager
313849 - Added the 'Safety related requirement' behavior
315317, 315319 - Additional requirements (Safe Execution, Lock Step)
230020 - Relation to the SEooC
14.01.2013 V2.3.10 324187, XSLT processor, issue51325
239057, 239065, 239067 corrected
313849 'S-Wdg' corrected to 'S-WdgM'
24.04.2013 V2.3.11 issue 53646: 358190 - Alive counter necessary
07.11.2013 V2.3.12 In the item 230126 the missing ISO 'part 6' was added.
02.04.2014 V2.3.13 Issue 59785 (partly): After discussion with customer following comments added: 542988, 544495
Issue 58655: 228813, 228815, 260615, 260617 (Win7 test)
Issue 52760, 62290, 61812, 59931
05.05.2014 V2.3.14 Changed points according EEB remarks, issue 52087
05.05.2014 V2.3.15 Improvements base is the customer OIL list, issue 59785
07.05.2014 V2.3.16 Issue 52087, 52760, 59785 : review points corrected
13.05.2014 V2.3.17 Issue 52087, 52760 corrected
14.05.2014 V2.3.18 Issue 59785 corrected
14.05.2014 V2.3.19 Issue 62591 corrected
14.05.2014 V2.3.20 Issue 62589 corrected

15.05.2014 V2.3.21 Issue 62290 corrected
15.05.2014 V2.3.22 Issue 53646 corrected
15.05.2014 V2.3.23 Issue 62290 corrected
15.05.2014 V2.3.24 Issue 59785, 62589, 62591corrected
16.05.2014 V2.3.25 Issue 62724 corrected
22.05.2014 V2.3.26 Issue 63131: Language Review
23.05.2014 V2.3.27 Issue 62724 corrected
26.05.2014 V2.3.28 Issues corrected:52168, 62591, 53646, 50833, 58842

Table of Contents

Category:	Comment	Keywords:		ID:	545205

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS SAFETY MANUAL IS GIVEN AS SUPPORT FOR THE INTEGRATION OF THE TTTECH SAFETY MODULE INTO A SYSTEM ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE TTTECH SAFETY MODULE. THE RECIPIENT OF THIS SAFETY MANUAL MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION.

TTTECH PROVIDES THE SAFETY MANUAL FOR THE SAFETY MODULE "AS IS" AND WITH ALL FAULTS AND HEREBY DISCLAIMS ALL WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OR COMPLETENESS, OR OF RESULTS TO THE EXTENT PERMITTED BY APPLICABLE LAW. THE ENTIRE RISK, AS TO THE QUALITY, USE OR PERFORMANCE OF THE SAFETY MANUAL, REMAINS WITH THE RECIPIENT. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW TTTECH SHALL IN NO EVENT BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOSS OF DATA, DATA BEING RENDERED INACCURATE, BUSINESS INTERRUPTION OR ANY OTHER PECUNIARY OR OTHER LOSS WHATSOEVER) ARISING OUT OF THE USE OR INABILITY TO USE THE SAFETY MANUAL, EVEN IF TTTECH HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TTTECH MAKES NO WARRANTY OF ITS PRODUCTS, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW DISCLAIMS ALL LIABILITIES OR DAMAGES RESULTING FROM OR ARISING OUT OF THE APPLICATION OR USE OF THESE PRODUCTS.

Category:	Comment	Keywords:		ID:	545219

Legal Notice

The information contained in this safety manual does not affect or change any General Terms and Conditions of TTTech and/or any agreements existing between TTTech and the recipient regarding the product concerned.

The reader acknowledges that this safety manual may not be reproduced, stored in a retrieval system, transmitted, changed, or translated, in whole or in part, without the express written consent of TTTech.

The reader acknowledges that any and all of the copyrights, trademarks, trade names, patents (whether registrable or not) and other intellectual property rights embodied in or in connection with this safety manual are and will remain the sole property of TTTech or the respective right holder. Nothing contained in this legal notice, the safety manual or in any TTTech web site shall be construed as conferring to the recipient any license under any intellectual property rights, whether explicit, by estoppel, implication, or otherwise.

This safety manual and respective products are subject to change.

The product is only allowed to be used in the scope as described in section "System Assumptions". Please note, that based on the current state of the arts in science it is impossible to develop software that is bug-free in all applications.

Category:	Comment	Keywords:		ID:	545221

We Listen to Your Comments

Is there any information in this document that you feel is wrong, unclear or missing?
Your feedback will help us to continuously improve the quality of this document. Please contact TTTech Automotive support if you have questions, change requests or suggestions for improvement related to the SCM product or documentation. TTTech Automotive support can be reached via the following e-mail address: support@tttech.com.

1 Purpose of this Document

Category:	Comment	Keywords:		ID:	228517
This document is the Software Safety Manual for the software component Safe Watchdog Manager (S-WdgM). The S-WdgM was developed by TTTech as an SEooC according to ISO 26262 (2011) for use in safety related items up to ASIL D (see [ISO26262]). This document contains the requirements that have to be met to integrate and apply the S-WdgM into a safety-related item.					

Category:	Comment	Keywords:		ID:	228519
The S-WdgM is part of the S-WdgM Stack. It contains also a S-WdgM Configuration Generator and a S-WdgM Verifier to generate and verify configuration dependent S-WdgM code.					

Category:	Comment	Keywords:		ID:	228521
The document contains the requirements that have to be satisfied to • install the S-WdgM Generator, • generate S-WdgM code with the S-WdgM Configuration Generator, • integrate the S-WdgM code into an AUTOSAR system, and • to apply the S-WdgM within an AUTOSAR system.					

Category:	Comment	Keywords:		ID:	228533
Note: The document describes requirements for the S-WdgM only. It does not provide a full description of how to create a safe system. For example, it is not concerned with hardware architectural metrics that may have an influence on software running on that hardware. These considerations are not specific to the S-WdgM and are thus beyond the scope of this manual.					

Category:	Comment	Keywords:		ID:	231307
The S-WdgM was developed according to AUTOSAR version 4.0.1 [AS_WDGM_SWS] and adapted for the AUTOSAR 3.1.4 [AS_WDGM_SWS_3_1] environment, too. The S-WdgM is compatible with both AUTOSAR versions but not fully compliant. For the deviations see [TT_WDGM_UM].					

2 Introduction

2.1 Target Audience and Responsibilities

Category:	Comment	Keywords:		ID:	228523

This document addresses the Safety Manager and (system) integrator. The integrator is the person who implements the requirements, is responsible for the generation of S-WdgM Configuration code, the integration of the S-WdgM into a safety-related item and its application.

Category:	Requirement	Keywords:		ID:	228525
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall be an expert in the area of functional safety with deep knowledge of ISO 26262 (see [ISO26262]).

Moreover, the integrator needs to know

- the AUTOSAR architecture,
- the ANSI C programing language, and
- the S-WdgM User Manual [TT_WDGM_UM]).

Category:	Requirement	Keywords:		ID:	228529
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall ensure that all requirements defined in this Safety Manual are fulfilled in the integrated item.

Category:	Requirement	Keywords:		ID:	228537
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall also follow the instructions in

- the Safety Manual for the S-WdgIf (see [TT_WDGIF_SM]) and
- the Safety Manual for the used S-Wdg drivers (see the driver specific Safety Manual. Safety Manuals for some drivers can be found in section "References" at the end of this document)

which describe the other components of the S-WdgM Stack.

2.2 Structure of this Document

Category:	Requirement	Keywords:		ID:	228527
Label:		Safety relevant:			
Related To:		Related To':			

Requirements are explicitly marked as "Requirement" in this document. All requirements described in this document shall be considered by the integrator. Explanatory text that does not represent an explicit requirement is marked as "Comment".

Category:	Comment	Keywords:		ID:	314003

Note: The document items of type "Comment" do not represent explicit action items for the integrator, however, the integrator has to ensure that there are no contradictions between the comment and the intend S-WdgM usage.

Category:	Comment	Keywords:		ID:	313849

Note: Requirements in this document shall be treated either as safety related or need not be treated as safety related, depending on the S-WdgM use case:

- If the S-WdgM is used to monitor a safety related application, then for each used S-WdgM functionality all corresponding requirements in this document shall be treated as safety related.
- If the S-WdgM is used to monitor a QM application then the requirements in this document need not be treated as safety related.

As a consequence, the field "Safety relevant" in the requirements are empty.

Category:	Comment	Keywords:		ID:	555645

The list shows some keywords used in requirements and their explanation:

Key Word	**Description**
Must, Shall, Required, Is responsible for, Is the responsibility of	Requirement is mandatory.
Shall not	Requirement is a prohibition.
May	Requirement is optional.

table 1

3 Terms

Category:	Comment	Keywords:		ID:	228565

Alive Supervision	A kind of monitoring that checks whether a Checkpoint in the application code has been passed an allowed number of times (with tolerances) within a time interval.
Application Context	An Application Context is the smallest set of data used by an application that must be saved to allow application interruption at a given time, and a continuation of this application at the point where it has been interrupted.
Checkpoint	A point in the control flow of a Supervised Entity which reports to the Safe Watchdog Manager when it is passed.
Configuration Tool	A tool (like DaVinci Configurator Pro) that creates a Safe Watchdog Manager configuration.
Deadline Monitoring	A kind of monitoring that checks whether the execution time between two Checkpoints is within expected limits (with tolerances).
End Checkpoint	The last Checkpoint in the program flow of a Supervised Entity. When the End Checkpoint has been passed, the S-WdgM assumes that the Supervised Entity has been left. An entity can have more than one End Checkpoint (e.g, in the "then" and "else" clause of an "if" statement).
Error Escalation	The escalation of a detected fault to the WD by a Watchdog reset by calling a S-WdgIf API function or omittance of the Watchdog trigger. The Error Escalation marks the point in time when the S-WdgM Fault Reaction Time ends and the reaction time of the WD driver and WD itself starts.
S-WdgM Fault Detection Time	The time from the occurrence of a fault to the detection by the S-WdgM. The detection is indicated by a status change from WDGM_LOCAL_STATUS_OK or WDGM_GLOBAL_STATUS_OK to another state. The duration of the S-WdgM Fault Detection Time in dependence of the S-WdgM Configuration is explained in this document. The S-WdgM Fault Detection Time is also called "diagnostic test interval" in [ISO26262].
S-WdgM Fault Reaction Time	The time from fault detection to the error escalation to the WD driver (through the S-WdgIf). The duration of the S-WdgM Fault Reaction Time in dependence of the S-WdgM Configuration is explained in this document. Note: The S-WdgM Safety Manual can only discuss the part of the Fault Reaction Time interval at the S-WdgM level. This part of the Fault Reaction Time is prefixed with "S-WdgM". The S-WdgM Fault Reaction Time is • the Fault Reaction Time according to [ISO26262] minus • the reaction time of the WD driver and the WD itself. For calculation of the WD driver see the according Safety Manual.

Freedom from interference	The absence of cascading failures between two or more elements that could lead to the violation of a safety requirement. See [ISO26262], part1.
Global Monitoring Status	The status that summarizes the Local Monitoring Status of all Supervised Entities. It indicates whether the S-WdgM has found an error so far.
Global Transition	In the context of this document a Global Transition is a transition between two Checkpoints of two different Supervised Entities.
Initial Checkpoint	The first Checkpoint in the control flow of a Supervised Entity. The monitoring of a Supervised Entitiy starts when the Initial Checkpoint is passed. A Supervised Entity has exactly one Initial Checkpoint.
Local Monitoring Status	A status that represents the current state of supervision of a single Supervised Entity. It indicates whether the S-WdgM has found an error so far.
Local Transition	In the context of this document a Local Transition is a transition between two Checkpoints of the same Supervised Entity.
Monitoring / Supervision	In the context of the S-WdgM Stack the terms Monitoring and Supervision are synonyms.
Monitoring Feature	The generic term for Alive Supervision, Deadline monitoring and Program Flow Monitoring.
Local/Global OK-Status	The Local OK-Status is present, when the local status is WDGM_LOCAL_STATUS_OK. The Global OK-Status is present, when the global status is WDGM_GLOBAL_STATUS_OK
Program Flow Monitoring	A kind of monitoring that checks whether the Checkpoints in a Supervised Entity are passed in an expected order.
Safe Watchdog Driver	The lower and hardware dependent software layer of the S-WdgM Stack. It controls the Watchdog device.
Safe Watchdog Interface	The middle and hardware independent software layer of the S-WdgM Stack.
Safe Watchdog Manager Configuration	The part of the S-WdgM code that is generated by the S-WdgM Generator out of an ECU description file.
Safe Watchdog Manager Configuration Generator	This TTTech tool generates a S-WdgM Configuration out of an ECU description file. In this document the name is abbreviated to "S-WdgM Generator". The tool is part of the S-WdgM package.
Safe Watchdog Manager	The upper and hardware independent software layer of the S-WdgM Stack. It communicates with the application through RTE.
Safe Watchdog Manager Stack	The stack comprises the S-WdgM, the Safe Watchdog Interface and the Safe Watchdog driver(s).
Supervised Entity	A software entity that is monitored by the S-WdgM. Each Supervised Entity has an identifier. A Supervised Entity is defined as a set of Checkpoints that are (directly or indirectly) connected by Local Transitions within a software component or basic software module. There may be zero, one or more Supervised Entities in a software component or basic software module. Additional TTTech note:Each Supervised Entity has a state that is based on the reports from all its Checkpoints.
Supervision Cycle	The time period of the S-WdgM in which the cyclic supervision

	algorithm is executed. At the end of a cycle, the function WdgM_MainFunction () is called and - depending on the configuration - Alive Supervision, Deadline Supervision and/or Program Flow Supervision are performed. See also "Reference Cycle".
System	A set of elements that relates at least a sensor, a controller and an actuator with one another (see [ISO26262], part1). In this document, the MCU is part of the system.
Reference Cycle	Each kind of monitoring has its own Reference Cycle, which is a multiple of the Supervision Cycle. At the end of the Reference Cycle, the according kind of monitoring checks whether an error has occured. For example: If the Reference Cycle for Deadline Supervision is 5 times the Supervision Cycle, then every 5th call of WdgM_MainFunction () checks for deadline violations.
Timebase Tick	The S-WdgM measures the deadline of a Transition in Timebase Ticks. It is also called S-WdgM Tick. The Timebase Tick can be provided either by the S-WdgM itself or by an external source.
Timing Fault	The generic term for the different kinds of fault that the S-WdgM can detect using a Monitoring Feature: • omittance of an operation, • unrequested execution of an operation, • operation executed too early, • operation executed too late, and • operations executed in the wrong sequence.
Watchdog (device)	A Watchdog device is the hardware part that provides the Watchdog function. It can be an internal watchdog (on the MCU) or an external device.
WD Mode	The "WD Mode" represents watchdog property. According AUTOSAR it can have the value: • "slow", • "fast", and • "off" (WD disabled).
WD Trigger Mode	The "WD Trigger Mode" defines the WD trigger window and consist of: • the window start time, • the window end time, and • the WD mode (slow, fast, off) It can be set with the function WdgM_SetMode (). For details see [TT_WDGM_UM] and [TT_WDGDR_*platform*_UM] (where *platform* is the used platform).

table 2

4 Notations

Category:	Comment	Keywords:		ID:	228609

Notation	Description
text	Italic text is a placeholder for a certain name or pattern. E.g.: In Wdg_*platform*_Init (), the text *platform* is a placeholder for the name of (a) specific platform(s).
AS3: *text*	The text after "AS3:" is relevant for AUTOSAR 3.1 environments only.
AS4: *text*	The text after "AS4:" is relevant for AUTOSAR 4.0 environments only.

table 3

5 Abbreviations

Category:	Comment	Keywords:		ID:	228549

API	Application Programming Interface
AS3	AUTOSAR 3.1 (environment)
AS4	AUTOSAR 4.0 (environment)
ASIL	Automotive Safety Integrity Level
AUTOSAR	Automotive Open System Architecture
BSW	Basic Software (AUTOSAR term)
BswM	BSW module
CP	Checkpoint
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECC	Error Checking (and) Correction
ECU	Engine Control Unit
ISO	International Organization for Standardization
MCU	Microcontroller Unit
MPU	Memory Protection Unit. Usually it is a part of the Microcontroller.
MemMap	Memory Mapping (for Memory Management)
QM	Quality Managed (Software)
RTE	Run-Time Environment
SC	SupervisionCycle
SchM	Schedule Manager module according to AUTOSAR 4.0 specification
SE	Supervised Entity
SM	Safety Manual
SW-C, SWC	Software Component
S-Wdg	Safe Watchdog Driver (from TTTech)
S-WdgM	Safe Watchdog Manager (from TTTech)
S-WdgIf	Safe Watchdog Interface (from TTTech)
WD	Watchdog
WdgM	Watchdog Manager according to the AUTOSAR 4.0 specification
WdgIf	Watchdog Interface according to the AUTOSAR 4.0 specification

table 4

6 Safe Watchdog Manager Overview

Category:	Comment	Keywords:		ID:	228613

For an overview of and more details about

- the S-WdgM,
- the other S-WdgM Stack components,
- the S-WdgM Generator, and
- the S-WdgM Verifier

see the according user manuals and Safety Manuals:

- for the S-WdgM: [TT_WDGM_UM] and this document,
- for the S-WdgIf: [TT_WDGIF_UM] and [TT_WDGIF_SM], and
- for the S-Wdg drivers: the according Safety Manual. See also section "References" at the end of this document.

Category:	Comment	Keywords:		ID:	555650

The Safe Watchdog Manager can be integrated into AUTOSAR 3.1.4 and AUTOSAR 4.0.1 environments. The S-WdgM code differs between the AUTOSAR versions.
The S-WdgM must be configured for the used AUTOSAR version with the preprocessor switch WDGM_AUTOSAR_4_x. This switch is automatically generated by the S-WdgM Configuration Generator.

Category:	Comment	Keywords:		ID:	559886

The S-WdgM is designed for integration into an AUTOSAR version 3.1.4 or AUTOSAR version 4.0.1 system. However, the S-WdgM is not restricted to this AUTOSAR versions. The software module can also be integrated into other versions of AUTOSAR and other system SW architectures, provided that the integration related requirements listed in the Safety Manual are satisfied.

Category:	Comment	Keywords:		ID:	562764

The Safe Watchdog Manager can also be switched to a "S-WdgM AUTOSAR 3.1 compatibility mode".
In this mode the behaviour of S-WdgM functions is as defined for the AUTOSAR 3.1 Watchdog Manager.
The mode is set with the preprocessor switch WDGM_AUTOSAR_3_1_X_COMPATIBILITY. The default value is STD_OFF. On the ECU description file level, the WdgMSupportedAutosarAPI parameter is used.

7 System Assumptions

Category:	Comment	Keywords:		ID:	270633

The S-WdgM module has been developed as a Safety Element out of Context (SEooC) according to ISO 26262. This means that the development was based on assumptions about the target environment where it shall be integrated. The integrator has to assure that these assumptions are fulfilled by the system.

The assumptions are listed as requirements in this section. Further requirements in this Safety Manual that may be considered assumptions (depending on the application of the system) are listed in section "Assumptions in this Document" below.

Category:	Requirement	Keywords:		ID:	282827
Label:		Safety relevant:			
Related To:	__MKSID__283135	Related To':			

The system specification shall be designed to tolerate the occurrence of timing faults. Also a certain (configurable but always greater than 0) time delay from the occurrence of faults to the safe state must be acceptable.

Category:	Comment	Keywords:		ID:	282829

The S-WdgM reacts on timing faults after they occurred. The detection and reaction time also depends on the S-WdgM Configuration.
The S-WdgM is not designed for systems where timing fault shall not occur at all.

Category:	Requirement	Keywords:		ID:	282805
Label:		Safety relevant:			
Related To:	__MKSID__262696,__MKSID__263095	Related To':			

The MCU shall provide computational resources to execute software components within their specification.

Category:	Requirement	Keywords:		ID:	282785
Label:		Safety relevant:			
Related To:	__MKSID__262682,__MKSID__262690,__MKSID__263089,__MKSID__263091,__MKSID__283504,__MKSID__283399,__MKSID__283508	Related To':			

The software execution environment shall be able to run software according to requirements of up to the system's required ASIL.
This also includes:

- free from interference among the SW components (see 282807),
- supervision by an extern measures (see 282795),
- the hardware shall consist of an MCU with all required hardware to run according to system specifications (i.e. safe HW to detect/avoid e.g. bit-flips by means of start up checks, cyclical checks, ECC check, ...), and
- the hardware shall be composed of components that are qualifiable up to the desired ASIL of the system.

Category:	Requirement	Keywords:		ID:	297946
Label:		Safety relevant:			
Related To:		Related To':			

The software execution environment shall provide methods for mutual exclusion.

Category:	Comment	Keywords:		ID:	297948

Such methods are disabling of interrupts, locks, semaphores etc.
Especially disabling of interrupts is often used to gain exclusive access to resources or perform multiple operations atomically.

Category:	Requirement	Keywords:		ID:	282807
Label:		Safety relevant:			
Related To:	__MKSID__263115,__MKSID__283536,__MKSID__261192	Related To':			

The software platform shall provide an execution environment that is capable of running multiple software components with freedom from interference from each other.

Category:	Comment	Keywords:		ID:	282809

The S-WdgM and the supervised application are considered as separate SW components with freedom from (unintended) interference. Freedom from interference can be achieved by e.g. a microcontroller with MPU.

Category:	Requirement	Keywords:		ID:	282795
Label:		Safety relevant:			
Related To:	__MKSID__262661,__MKSID__263099,__MKSID__263109,__MKSID__283504,__MKSID__283508	Related To':			

The integrator shall analyze, what safety measures are required in case of timing violations

- of the calls of the S-WdgM and
- during execution of the S-WdgM.

Category:	Comment	Keywords:		ID:	561887

The timing violations described above are not handled by S-WdgM internally and must be handled externally if necessary.

The timing violation can be caused by e.g.

- slower/faster running MCU oscillator or
- a delay by too many high priority tasks.

Category:	Comment	Keywords:		ID:	282797

An internal WD can detect timing violations of S-WdgM calls and S-WdgM executions. However, an internal WD may have the same time base (oscillator) as the CPU that executes the S-WdgM and therefore may not be able to detect failures of the time base.
An external WD with an independent time base may be necessary.

Category:	Requirement	Keywords:		ID:	315317
Label:		Safety relevant:			
Related To:		Related To':			

The MCU shall execute the given software correctly.

Category:	Comment	Keywords:		ID:	315319

This requirement can be achieved e.g. by using a lockstep MCU.

Category:	Requirement	Keywords:		ID:	282791
Label:		Safety relevant:			
Related To:	__MKSID__262674	Related To':			

In case a software timing fault has been detected and escalated to the system by the S-WdgM, the system shall initiate the safe state within acceptable time tolerances.

Category:	Comment	Keywords:		ID:	282793

The S-WdgM initiates a fault reaction by discontinuation of WD triggering or by a WD reset. It is the integrators responsibility to ensure that the WD itself leads to a safe state in time.
Note; The S-WdgM detection and reaction time is also delayed depending on the S-WdgM Configurations.

Category:	Requirement	Keywords:		ID:	283375
Label:		Safety relevant:			
Related To:	__MKSID__283514,__MKSID__283518	Related To':			

The connected (used) Watchdog (or a hardware that provide the watchdog function) shall work correctly.

Category:	Requirement	Keywords:		ID:	282789
Label:		Safety relevant:			
Related To:	__MKSID__262604,__MKSID__263117,__MKSID__283508,__MKSID__283504,__MKSID__261244	Related To':			

The MCU shall be able to perform a safe startup to the point of where the S-WdgM is safely initialized.

Category:	Requirement	Keywords:		ID:	566080
Label:		Safety relevant:			
Related To:		Related To':			

The RAM memory correctness shall be checked at ECU startup time. An ECC or comparable check shall be used at run-time.

Category:	Requirement	Keywords:		ID:	265876
Label:		Safety relevant:			
Related To:	__MKSID__283397	Related To':			

The FLASH memory correctness shall be checked at ECU startup time. An ECC or comparable check shall be used at run-time.

Category:	Comment	Keywords:		ID:	263975

The generated code contains a checksum over some significant fields (e.g. version) to check that:

- the generated code belongs to the S-WdgM code according to version information and
- the generated code is not overwritten by other code at the flashing process.

The checksum is checked with every run of the function WdgM_Init (). A failed check yields WDGM_E_PARAM_CONFIG.

Note: The checksum does not cover the complete configuration and cannot thoroughly detect when the configuration memory is corrupted (like bitflips).

7.1 Assumptions in this Document

Category:	Requirement	Keywords:		ID:	282887
Label:		Safety relevant:			
Related To:		Related To':			

The following requirements are located in the according context in this document. They may be interpreted as system assumptions or not - depending on the circumstances the system is developed and applied:

Requirement	Description
231900, 230957	Chosen monitoring features and configuration meet the system's safety requirements.
260470, 231825, 229211, 236796, 230793	Quality level degradation by external interfaces.
230494	S-WdgM functionality affected by other SW.
260490, 231403, 231419	Quality level degradation by SE deactivation.
260207, 231823, 231547, 231549, 231609	WD driver and WD device.
231277, 231281, 231454, 231462, 231972, 231203	Memory sections, access rights.
231480	Memory corruption.
231207	WdgM_MainFunction () in separated task.

table 5

8 S-WdgM Function Requirements

Category:	Comment	Keywords:		ID:	270655
The section lists the system requirements that the S-WdgM Stack fulfills. They are derived from [TT_WDGM_TSR] and [TT_WDGM_SD]. Since the S-WdgM function requirements are not requirements for the system or integrator, they are put here as comments and marked with "S-WdgM Requirement".					

Category:	Comment	Keywords:		ID:	282811
(S-WdgM Requirement) The S-WdgM shall be able to detect software timing faults: • There shall be methods to detect timing faults within a software components. • There shall be methods to detect timing faults among software components.					

Category:	Comment	Keywords:		ID:	282813
The S-WdgM is able to detect program flow violations, Alive Counter violations and deadline violations. They cover the following kinds of faults: • omittance of an operation (program flow, Alive Counter), • unrequested execution of an operation (program flow, Alive Counter), • operation executed too early (Alive Counter, deadline), • operation executed too late (Alive Counter, deadline), and • operations executed in the wrong sequence (program flow).					

Category:	Comment	Keywords:		ID:	282815
(S-WdgM Requirement) The S-WdgM shall escalate a detected SW timing fault to the system: There shall be methods to escalate detected faults so that a corresponding safety measure is triggered.					

Category:	Comment	Keywords:		ID:	282817
The S-WdgM initiates a fault reaction by discontinuation of WD triggering or by a WD reset. It is the integrators responsibility to ensure that the WD itself leads to a safe state in time.					

9 S-WdgM Configuration

Category:	Comment	Keywords:		ID:	228629
The S-WdgM Configuration code is the part of the S-WdgM code that is generated with the S-WdgM Generator out of a given ECU description file. This section lists the safety requirements for the creation of S-WdgM Configuration code.					

Category:	Comment	Keywords:		ID:	228631
For a description of • the configuration fields in the ECU description file and • how to generate S-WdgM code out of the ECU description file see [TT_WDGM_UM].					

9.1 Configuration Check-List

Category:	Comment	Keywords:		ID:	228713
The S-WdgM Generator performs basic checks on the contents of the ECU description file when generating the S-WdgM Configuration code. The following sections provide instructions for manual checks of safety relevant configuration values that cannot be performed by the S-WdgM Generator itself.					

Category:	Requirement	Keywords:		ID:	231900
Label:		Safety relevant:			
Related To:		Related To':			
If a subset of the S-WdgM monitoring features is used, then the integrator shall verify that the chosen monitoring features satisfy the system's safety requirements.					

9.1.1 General Requirements

Category:	Requirement	Keywords:		ID:	228717
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall set the configuration parameters according to the project specification.					

Category:	Requirement	Keywords:		ID:	260470
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall verify that no non-S-WdgM function that is called from within the S-WdgM degrades the quality level of the S-WdgM below the required quality level.					

Category:	Comment	Keywords:		ID:	544495
The used non-S-WdgM functions are listed in section "Expected Interface" below.					

Category:	Comment	Keywords:		ID:	260476
Example: If the functions GlobalSuspendInterrupts () and GlobalRestoreInterruts () are implemented for QM					

level and the S-WdgM calls these functions, then the S-WdgM is degraded to QM level.

Category:	Requirement	Keywords:		ID:	284187
Label:		Safety relevant:			
Related To:		Related To':			

The ECU description file that serves as input for the generation of the S-WdgM Configuration code shall follow the XML schema of the used AUTOSAR version. The supported AUTOSAR versions are defined in the 231307.

Category:	Comment	Keywords:		ID:	284517

The corresponding XML schema can be found in www.autosar.org.

9.1.2 Pre-Compile Settings

Category:	Requirement	Keywords:		ID:	228722
Label:		Safety relevant:			
Related To:		Related To':			

The following fields in the ECU description file shall be "true" if the according feature shall be enabled, otherwise "false":

Field	**Feature**
WdgMVersionInfoApi	Enable Version API.
WdgMDevErrorDetect	Enable Development error detection.
WdgMDemReport	Enable DEM calls in case of production errors.
WdgMDefensiveBehavior	Check whether a caller of WdgM_SetMode () is authorized to call the function. Also check that the S-WdgM was initialized when the function WdgM_MainFunction () is called. Note: The AUTOSAR 3.1 version of WdgM_SetMode () does not check the caller.
WdgMImmediateReset	Enable an immediate WD reset in case of a Alive Supervision violation, a Deadline violation or a ProgramFlow violation.
WdgMOffModeEnabled	Enable deactivation of a WD device.
WdgMUseOsSuspendInterrupt	**AS3**: Call SchM_Enter_WdgM () and SchM_Exit_WdgM () **AS4**: Call SchM_Enter_WdgM_WDGM_EXCLUSIVE_AREA_0() and SchM_Exit_WdgM_WDGM_EXCLUSIVE_AREA_0() The functions suspend and resume interrupts.
WdgMSecondResetPath	Call Mcu_PerformReset () if a WD trigger or a WD reset fails.
WdgMTickOverrunCorrection	Correct the tick counter when the value overflows.
WdgMEntityDeactivationEnabled	Enable deactivation and activation of SEs.
WdgMStateChangeNotification	Invoke a callback function when local or global state changes.
WdgMUseRte	Use the RTE-generated defines and typedefs.
WdgMDemSupervisionReport	Make a DEM call when global state WDGM_GLOBAL_STATUS_STOPPED is reached.
WdgMFirstCycleAliveCounterReset	Do not evaluate Alive Counters from the first SC (in the first call of WdgM_MainFunction ()).

table 6

Category:	Requirement	Keywords:		ID:	228883
Label:		Safety relevant:			
Related To:		Related To':			

The value of WdgMTimebaseSource shall be set according to the required source of time ticks:

WdgMTimebaseSource	Description
WDGM_INTERNAL_SOFTWARE_TICK (0)	An internal time source for Deadline Monitoring is selected. The tick counter is incremented each time the WdgM_MainFunction() is invoked.
WDGM_INTERNAL_HARDWARE_TICK (1)	An internal time source for Deadline Monitoring is selected. The tick counter value is read from an MCU's internal hardware counter.
WDGM_EXTERNAL_TICK (2)	An external time source for Deadline Monitoring is selected. The tick counter is incremented each time the WdgM_UpdateTickCount() function is invoked. The function is implemented in the S-WdgM.

table 7

Category:	Comment	Keywords:		ID:	239167

The field WdgMTimebaseSource is a WdgM information. If it is set to WDGM_INTERNAL_HARDWARE_TICK, then the configuration generator checks whether the referred driver has an active tick counter.

Category:	Requirement	Keywords:		ID:	230215
Label:		Safety relevant:			
Related To:		Related To':			

In case the S-WdgM internal hardware tick counter is used, the integrator shall make sure that the MCU's internal hardware counter updates the tick counter according to the system specifications.

Category:	Comment	Keywords:		ID:	270693

In case of an internal hardware tick counter, the S-WdgM updates the tick counter using the MCU's internal hardware counter.

Category:	Requirement	Keywords:		ID:	238968
Label:		Safety relevant:			
Related To:		Related To':			

If UseOSsuspendinterrupts is "false", then the integrator is responsible for the implementation of the functions

- GlobalSuspendInterrupts () and
- GlobalRestoreInterrupts ().

Category:	Requirement	Keywords:		ID:	260490
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall consider:
If WdgMEntityDeactivationEnabled is "true",

then a SW component that calls the functions

- WdgM_DeactivateSupervisionEntity() and
- WdgM_ActivateSupervisionEntity()

degrade the quality level of the S-WdgM to the quality level of their caller(s).

Category:	Comment	Keywords:		ID:	260491

Example: If two components are used with quality level ASIL-B and QM, then the S-WdgM is degraded to QM level.

Category:	Comment	Keywords:		ID:	260496

The functions WdgM_DeactivateSupervisionEntity() and WdgM_ActivateSupervisionEntity() degrade because a faulty activation or deactivation process for a SE call may compromise the monitoring features.

Category:	Comment	Keywords:		ID:	261042

A partition reset with BswM_WdgM_RequestPartitionReset () is not supported by the S-WdgM.

9.1.3 Post Build Configuration and Application Settings

Category:	Comment	Keywords:		ID:	239045

This section provides a check list for the various aspects and configuration fields that must be considered for implementation and post build configuration of the monitoring features.

Category:	Comment	Keywords:		ID:	239073

For further information on configuration fields see [TT_WDGM_UM]. For information on configuration of S-WdgM Fault Detection Times and S-WdgM Fault Reaction Times, see section "S-WdgM Fault Detection Time and S-WdgM Fault Reaction Time Evaluation" below.

Category:	Requirement	Keywords:		ID:	260207
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall make sure that the configuration defines

- only one WD driver and
- only one WD device for the driver.

Category:	Comment	Keywords:		ID:	260209

The current implementation of the S-WdgM Stack supports only one WD device per WD driver. If configured otherwise, the S-WdgM Generator yields an error message.

Category:	Comment	Keywords:		ID:	260211

The current implementation of the S-WdgM Stack supports one WD driver and one WD device per driver. If configured otherwise, the S-WdgIf Generator yields an error message.

Category:	Requirement	Keywords:		ID:	260219
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall make sure that all API functions of the S-WdgIf that require a device index, use 0 as device index.

Category:	Comment	Keywords:		ID:	260221

The index counting for the WD device starts with 0.

Category:	Requirement	Keywords:		ID:	238981
Label:		Safety relevant:			
Related To:	__MKSID__261186	Related To':			

The integrator shall

- partition the supervised application code into SEs,
- configure the OSApplication ID per SE,
- place CPs per SE (including Initial CPs and - if necessary - End CPs),
- place global CP (including Initial CPs and - if necessary - End CPs),
- configure Deadline Monitoring,
- configure Alive Supervision, and
- configure Program Flow Monitoring

according to the system requirements for S-WdgM monitoring.

Category:	Requirement	Keywords:		ID:	358190
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall be aware that, if

- the execution does not hit any CP in a SE and
- no Alive Supervision is configured for this SE,

then the S-WdgM will not detect this violation.

Category:	Comment	Keywords:		ID:	565654

For periodic SE, this can be solved by configuration of Alive Supervision for the SE.
For non periodic SE, Alive Supervision can not be used.

Category:	Requirement	Keywords:		ID:	239047
Label:		Safety relevant:			
Related To:		Related To':			

For the notification of state changes, the integrator shall set

- WdgMLocalStateChangeCbk (per SE) and
- WdgMGlobalStateChangeCbk

according to the system requirements.

Category:	Requirement	Keywords:		ID:	239049
Label:		Safety relevant:			
Related To:		Related To':			

For the activation/deactivation of SEs, the integrator shall set

- WdgMEnableEntityDeactivation (per SE) and
- WdgMInitialStatus (per SE)

according to the system requirements.

Category:	Requirement	Keywords:		ID:	239051
Label:		Safety relevant:			
Related To:	__MKSID__283870,__MKSID__284614,__MKSID__261172,__MKSID__261176,__MKSID__261174,__MKSID__261178	Related To':			

For the scheduling of WdgM_MainFunction () calls, the integrator shall set

- WdgMTicksPerSecond,
- WdgMSupervisionCycle,
- WdgMTriggerWindowStart (per WD Trigger Mode), and
- WdgMTriggerConditionValue (per WD Trigger Mode)

according to the system requirements.

Category:	Requirement	Keywords:		ID:	239053
Label:		Safety relevant:			
Related To:		Related To':			

For correct handling of WD Trigger Modes the integrator shall set

- WdgMAllowedCallers,
- WdgMInitialTriggerModeId (for SetMode ()), and
- WdgMWatchdogMode

according to the system requirements.

9.1.3.1 Alive Monitoring

Category:	Requirement	Keywords:		ID:	239055
Label:		Safety relevant:			
Related To:	__MKSID__261186	Related To':			

The integrator shall

- define Alive Supervision for every CP,
- set WdgMExpectedAliveIndications per WdgMSupervisionReferenceCycle properly, and
- set the interval [WdgMMinMargin, WdgMMaxMargin] narrow enough

so that Alive Supervision violations are detected according to system requirements.

Category:	Requirement	Keywords:		ID:	239057
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall make sure that the following values are set correctly:

- WdgMTicksPerSecond,
- WdgMSupervisionCycle,
- WdgMSupervisionReferenceCycle (perCP),
- WdgMFailedSupervisionRefCycleTol (per SE), and
- WdgMExpiredSupervisionCycleTol,

so that the WD is reset after a time delay according to system requirements.

9.1.3.2 Deadline Monitoring

Category:	Requirement	Keywords:		ID:	239063
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall

- define Deadline Monitoring for every CP and
- set the interval [WdgMDeadlineMin, WdgMDeadlineMax] narrow enough,

so that Deadline violations are detected according to system requirements.

Category:	Requirement	Keywords:		ID:	239065
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall make sure that the following values are set correctly:

- WdgMTicksPerSecond,
- WdgMSupervisionCycle,
- WdgMDeadlineReferenceCycle (per SE),
- WdgMFailedDeadlineRefCycleTol (per SE), and
- WdgMExpiredSupervisionCycleTol,

so that the WD is reset after a time delay according to system requirements.

9.1.3.3 Program Flow Monitoring

Category:	Requirement	Keywords:		ID:	239071
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall define Program Flow Monitoring for every CP, so that program flow violations are detected according to system requirements.

Category:	Requirement	Keywords:		ID:	239067
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall make sure that the following values are set correctly:

- WdgMTicksPerSecond,
- WdgMSupervisionCycle,
- WdgMProgramFlowReferenceCycle (per SE),
- WdgMFailedProgramFlowRefCycleTol (per SE), and
- WdgMExpiredSupervisionCycleTol,

so that the WD is reset after a time delay according to system requirements.

9.1.3.4 Configuration Restrictions for S-WdgM AUTOSAR 3.1 Compatibility Mode

Category:	Comment	Keywords:		ID:	284790

If WDGM_AUTOSAR_3_1_X_COMPATIBILITY is set to STD_ON, then the S-WdgM behaves as defined for the AUTOSAR 3.1 Watchdog Manager. In this case further configuration restrictions shall be considered.

Note: The S-WdgM Generator or S-WdgM Verifier do not check the following restrictions.

Category:	Requirement	Keywords:		ID:	284792
Label:		Safety relevant:			
Related To:		Related To':			

If WDGM_AUTOSAR_3_1_X_COMPATIBILITY is set to STD_ON,
then the following restrictions must be considered:

- for all SEs WdgMSupportedAutosar is set to API_3_1 (in the ECU description file),
- there is only exactly one CP allowed for each SE,
- this CP must be defined as Initial CP and as End CP,
- every CP must have a Alive Supervision defined, and
- there are no local and global transitions allowed.

9.1.4 S-WdgM Fault Detection Time and S-WdgM Fault Reaction Time Evaluation

Category:	Comment	Keywords:		ID:	231587

The time span from a fault occurrence to the system's reaction depends on the S-WdgM Configuration parameters. This section shows how the different configuration timing parameters add up to the actual delay from the fault occurrence to the error escalation.

Category:	Comment	Keywords:		ID:	239236

A further description of the configuration parameters and examples can be found in [TT_WDGM_UM].

Category:	Comment	Keywords:		ID:	231597

Definition:
The time span from the fault occurrence to the error escalation by the S-WdgM to the WD driver (through S-WdgIf) is the sum of

1. the S-WdgM Fault Detection Time and
2. the S-WdgM Fault Reaction Time.

In [ISO26262], the S-WdgM Fault Detection Time is called "diagnostic test interval".

Category:	Comment	Keywords:		ID:	239636

The time spans of the different monitoring features do not affect each other (except of course, that the error escalation of one monitoring violation aborts the monitoring of all other violations.)

9.1.4.1 S-WdgM Fault Detection Time

Category:	Comment	Keywords:		ID:	260591

The S-WdgM Fault Detection Time is evaluated differently for the various monitoring features as shown in this section.

Category:	Comment	Keywords:		ID:	239252
The S-WdgM Fault Detection Time spans • from fault occurrence • to fault detection (when the S-WdgM switches from a Local or Global OK-Status to another state). The state change happens within the WdgM_MainFunction ().					

Category:	Comment	Keywords:		ID:	239560
The S-WdgM Fault Detection Time is differently defined for the various monitoring features.					

9.1.4.1.1 Alive Supervision

Category:	Comment	Keywords:		ID:	239284
Assume that a fault occurs that leads to an Alive Counter violation: The S-WdgM Fault Detection Time is the sum of the time spans • from the fault to the call of the next CP that monitors the alive count and • from the call of this CP to the next call of WdgM_MainFunction() at the end of the current SupervisionReferenceCycle.					

Category:	Comment	Keywords:		ID:	239300
Because a SupervisionReferenceCycle is a multiple of the SC, there may be other call(s) of WdgM_MainFunction () between the CP call and the end of the SupervisionReferenceCycle, but only the WdgM_MainFunction () call at the end of the SupervisionReferenceCycle detects the Alive Counter violation.					

Category:	Comment	Keywords:		ID:	239285
In the best case, the S-WdgM Fault Detection Time is less or equal a SupervisionReferenceCycle. This is when • the fault occurs, • the according CP is called afterwards, and • the WdgM_MainFunction is called at the end of the SupervisionReferenceCycle within the same SupervisionReferenceCycle.					

Category:	Comment	Keywords:		ID:	239286
Note: Depending on the locations of CPs, the time span from the fault occurrence to the CP call may include several SupervisionReferenceCycles. That is, when the CP is not called within every SupervisionReferenceCycle.					

9.1.4.1.2 Deadline Supervision

Category:	Comment	Keywords:		ID:	239240
Assume that a fault occurs that leads to a Deadline Violation: The S-WdgM Fault Detection Time is the sum of the time spans • from the fault to the call of the next CP that monitors the deadline and • from call of this CP to the next call of WdgM_MainFunction () at the end of the current SC.					

Category:	Comment	Keywords:		ID:	239242

In the best case, the S-WdgM Fault Detection Time is less or equal a SC. This is when

- the fault occurs,
- the CP that checks for Deadline Violation*) is called afterwards and
- the WdgM_MainFunction () is called at the end of the SC

within the same SC.

*) Deadline Monitoring includes at least 2 CPs: The first CP starts the timer, the second CP checks the timer for violation of the deadline constraints.

Category:	Comment	Keywords:		ID:	239244

Note: Depending on the locations of CPs, the time span from the fault occurrence to the CP call may include several SCs. That is, when the CP is not called within every SC.

9.1.4.1.3 Program Flow Supervision

Category:	Comment	Keywords:		ID:	239268

Assume that a fault occurs that leads to a Program Flow violation:
The S-WdgM Fault Detection Time is the sum of the time spans

- from the fault to the call of the next CP that monitors the program flow and
- from the call of this CP to the next call of WdgM_MainFunction () at the end of the current SC.

Category:	Comment	Keywords:		ID:	239269

In the best case, the S-WdgM Fault Detection Time is less or equal a SC. This is when

- the fault occurs,
- the according CP is called afterwards and
- WdgM_MainFunction () is called at the end of the SC

within the same SC.

Category:	Comment	Keywords:		ID:	239270

Note: Depending on the locations of CPs, the time span from the fault occurrence to the CP call may include several SCs. That is, when the CP is not called within every SC.

9.1.4.2 S-WdgM Fault Reaction Time

Category:	Comment	Keywords:		ID:	231805

The S-WdgM Fault Reaction Time spans

- from the end of the S-WdgM Fault Detection Time
- to the error escalation to the WD driver (through the S-WdgIf) (by trigger omittance or invokation of a WD reset by calling WdgIf_SetTriggerWindow(*driver*, 0, 0) for each *driver*).

Category:	Comment	Keywords:		ID:	239578

Note: This section does not discuss WD resets due to a S-WdgM error (like DET errors). S-WdgM errors always lead to immediate WD resets by call of ImmediateWatchdogReset ().

Category:	Comment	Keywords:		ID:	239580

Note: In the context of the S-WdgM, the S-WdgM Fault Reaction Time ends with the call of the according S-WdgIf functions

- WdgIf_SetTriggerWindow () and
- Mcu_PerformReset () (if the WD cannot be served correctly).

Be aware that there may be some (configured or HW related) delay from a function call to the actual system reset. See the manuals of the according S-Wdg drivers.

Category:	Comment	Keywords:		ID:	239616

The following assumptions take place here:

- A violation continues from one Reference Cycle (according to the monitoring feature) to the next until the error is escalated. Discontinuation of a violation before error escalation results in a recovery to the OK-Status.
- The monitored SEs are always active. Deactivation of a SE aborts the S-WdgM monitoring of this SE. Activation of a SE resumes the monitoring with OK-Status.

Category:	Comment	Keywords:		ID:	239658

There are two kinds of tolerances involved in the S-WdgM fault reaction time span:

- the number of tolerated Reference Cycles per monitoring feature (defined by WdgMFailedSupervisionRefCycleTol, WdgMFailedDeadlineRefCycleTol and WdgMFailedProgramFlowRefCycleTol, respectively) and
- the number of SupervisionCycles waiting until the actual error escalation takes place (defined by WdgMExpiredSupervisionCycleTol).

Category:	Comment	Keywords:		ID:	239662

Once the S-WdgM Fault Reaction Time has expired, the error escalation is performed as follows:
If WDGM_IMMEDIATE_RESET is set to STD_ON,
then by the call of WdgIf_SetTriggerWindow(*driver*, 0, 0) for each WdgM *driver* to invoke an immediate WD reset,
otherwise by omittance of the WD trigger.

Note: Some WDs do no support an immediate reset. If not supported, then the WD trigger is still omitted and the system resets after the WD timeout expired.

Category:	Comment	Keywords:		ID:	239634

The S-WdgM Fault Reaction Times of the different monitoring features do not affect each other (except of course, that the error escalation of one monitoring violation aborts all other monitoring violations.)

Category:	Comment	Keywords:		ID:	239582

Notation:
Within this section, the following notation is introduced:

"MF(*i*) is the *i*-th run of MainFunction () from the begin of the S-WdgM Fault Reaction Time."

MF(0) is the run of MainFunction () where the S-WdgM Fault Detection Time ends and the Fault Reaction Time starts.
MF(1) is 1 SC later.
MF(*sc*) is *sc* SCs after MF(0).

Category:	Comment	Keywords:		ID:	239584
The S-WdgM Fault Reaction Time is evaluated differently for the various monitoring features as shown in the following sections.					

9.1.4.2.1 Alive Supervision

Category:	Comment	Keywords:		ID:	239644

The error escalation is conducted in
  MF (*i*), which is *i* SCs after MF(0),
where
  i = (WdgMSupervisionReferenceCycle * WdgMFailedSupervisionRefCycleTol) + WdgMExpiredSupervisionCycleTol

This is after *i* SCs.

9.1.4.2.2 Deadline Supervision

Category:	Comment	Keywords:		ID:	239650

The error escalation is conducted in
  MF (*i*), which is *i* SCs after MF(0),
where
  i = (WdgMDeadlineReferenceCycle * WdgMFailedDeadlineRefCycleTol) + WdgMExpiredSupervisionCycleTol

This is after *i* SCs.

9.1.4.2.3 Program Flow Supervision

Category:	Comment	Keywords:		ID:	239654

The error escalation is conducted in
  MF (*i*), which is *i* SupervisionCycles after MF(0),
where
  i = (WdgMProgramFlowReferenceCycle * WdgMProgramFlowDeadlineRefCycleTol) + WdgMExpiredSupervisionCycleTol

This is after *i* SCs.

10 S-WdgM Configuration Generator

Category:	Comment	Keywords:		ID:	228807

This section lists the safety requirements for the installation and application of the S-WdgM Generator. It also lists the safety requirements for the verification of the S-WdgM Generators results.

Category:	Comment	Keywords:		ID:	228809

For information on how to use the S-WdgM Generator, see [TT_WDGM_UM].

Category:	Comment	Keywords:		ID:	228635

Note: The S-WdgM Generator is **not** ASIL-D. Its output cannot be trusted, hence additional checks are required by use of the S-WdgM Verifier, which is part of the S-WdgM package.

10.1 S-WdgM Generator - Installation

Category:	Requirement	Keywords:		ID:	228813
Label:		Safety relevant:			
Related To:		Related To':			

If the S-WdgM Generator is installed and used on a different OS than Windows 7 with Service Pack 1, the integrator is responsible for ensuring that the change of the underlying OS does not affect the behavior and output of the S-WdgM Generator.

Category:	Comment	Keywords:		ID:	228815

The S-WdgM Generator has been tested on Windows 7 with Service Pack 1.

10.2 S-WdgM Generator - Application

Category:	Requirement	Keywords:		ID:	228823
Label:		Safety relevant:			
Related To:		Related To':			

The selected output path for the generated S-WdgM code (runtime argument "OUTPUT-DIRECTORY") shall be empty before the S-WdgM Generator is started.

Category:	Comment	Keywords:		ID:	228825

If the output path is not empty, code from previous generation runs may be accidentally integrated into the AUTOSAR system.

Category:	Comment	Keywords:		ID:	263300

The generated files are listed on standard error (stdout).

Category:	Requirement	Keywords:		ID:	228827
Label:		Safety relevant:			
Related To:		Related To':			

If the S-WdgM Generator aborts the generation process with an error, the (partially) generated output files shall not be used in an AUTOSAR system.

Category:	Comment	Keywords:		ID:	228829

Error messages start with "Error" and are displayed on standard error (stderr).
If successful, the S-WdgM Generator returns error level 0, otherwise an error level higher than 0 is returned.

Category:	Requirement	Keywords:		ID:	228831
Label:		Safety relevant:			
Related To:		Related To':			

If the S-WdgM Generator displays a warning message, the integrator shall ensure that the cause of the warning does not invalidate the generated S-WdgM Configuration.

Category:	Comment	Keywords:		ID:	228833

Warning messages start with "Warning" and are displayed on standard error (stderr).
If successful (even with warning), the S-WdgM Generator returns error level 0, otherwise an error level higher than 0 is returned.

Category:	Comment	Keywords:		ID:	229689

In case of an error free application of the generator, the generated S-WdgM Configuration files in the output directory are:

- WdgM_PBCfg.c
- WdgM_PBCfg.h
- **AS3:** WdgM_MemMap.h, or
- **AS4:** WdgM_OSMemMap.h
- WdgM_Cfg_Features.h

Category:	Comment	Keywords:		ID:	231187

TTTech provides a sample demonstration configuration with four SEs. The files may be used by the integrator, but are intended for demonstration only.

Category:	Comment	Keywords:		ID:	228837

The S-WdgM Generator is not configurable. The S-WdgM Generator process is controlled by the input arguments only.

10.3 S-WdgM Generator - S-WdgM Configuration Verification

Category:	Comment	Keywords:		ID:	229705

This section lists the safety requirements for the verification of the S-WdgM Configuration (i.e. the generated C- and Header-files) of the S-WsgM Generator run.

Category:	Comment	Keywords:		ID:	228843

This section describes how the output of the S-WdgM Generator is to be checked so that the output has ASIL-D quality.

Category:	Comment	Keywords:		ID:	290318
The verification process consists of the following steps, which are explained in details in the following sections: • creation of S-WdgM Info files out of the ECU Description file (for the Verifier build), • build (compilation) of the Verifier, • Verifier run and manual check of Verifier report, • manual checks (which can not be performed by the Verifier) and • check of system specifications against the S-WdgM Info files.					

Category:	Requirement	Keywords:		ID:	291126
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall use the same ECU Description file for verification that was used for the generation of the S-WdgM Configuration files, which are verified.					

Category:	Requirement	Keywords:		ID:	260615
Label:		Safety relevant:			
Related To:		Related To':			
If the S-WdgM Verification process is performed on a different OS than Windows 7 with Service Pack 1, the integrator is responsible for ensuring that the change of the underlying OS does not affect the behavior and output of the S-WdgM Verification process.					

Category:	Comment	Keywords:		ID:	260617
The S-WdgM has been tested on Windows 7 with Service Pack 1.					

10.3.1 Check S-WdgM Configuration against ECU Configuration

Category:	Requirement	Keywords:		ID:	228865
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall ensure that all applied files in the verification process are of the same delivered S-WdgM package.					

Category:	Comment	Keywords:		ID:	228871
Do not use files of different S-WdgM package versions.					

Category:	Requirement	Keywords:		ID:	228877
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall make sure that all files that are applied in the verification process are unaltered: • files that are delivered by TTTech are unaltered, • files created during the verification process are unaltered from creation to application.					

10.3.1.1 Creation of S-WdgM Info Files

Category:	Requirement	Keywords:		ID:	232265
Label:		Safety relevant:			
Related To:		Related To':			

The S-WdgM Info files are a header and a C file with the ECU Description information as C code which is checked against the generated files.

They shall be named

- wdgm_verifier_info.h and
- wdgm_verifier_info.c

(See Requirement 229681and Comment 263659 for details)

Category:	Requirement	Keywords:		ID:	229673
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall use an XSLT Processor, which fulfills the requirements in [ISO26262], part 8, clause 11.4.

Category:	Comment	Keywords:		ID:	324187

The S-WdgM package of TTTech contains an ISO26262 classified XSLT processor named "xsltproc.exe".

Category:	Comment	Keywords:		ID:	263574

The verifier has been tested with xsltproc.exe which uses libxslt V1.1.26 (Win32).

Category:	Comment	Keywords:		ID:	269546

The required XSL transformations do not use any XSLT 2.0 features; therefore, a XSLT 1.0 compliant processor can be used; e.g., XML Spy, xsltproc or Xalan.

Category:	Comment	Keywords:		ID:	269548

The following examples assume that xsltproc is being used. The command-line syntax for Xalan is very similar. XML Spy is a GUI program.

Category:	Requirement	Keywords:		ID:	229681
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall perform two XSL transformations:

The integrator shall call the XSLT processor to apply the verify_wdgm_header.xsl stylesheet (part of the package) to the ECU description file and store the transformation's result in the file wdgm_verifier_info.h.

The integrator shall call the XSLT processor to apply the verify_wdgm_source.xsl stylesheet (part of the package) to the ECU description file and store the result in the file wdgm_verifier_info.c.

Category:	Comment	Keywords:		ID:	263659

If xstlproc.exe is used as XSLT processor, the syntax for the two calls is:

- xsltproc.exe verify_wdgm_header.xsl *ECU-description-file* >wdgm_verifier_info.h
- xsltproc.exe verify_wdgm_source.xsl *ECU-description-file* >wdgm_verifier_info.c

10.3.1.2 Verifier Compilation

Category:	Comment	Keywords:		ID:	228857

The S-WdgM Verifier executable is created as follows:

Category:	Requirement	Keywords:		ID:	229683
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall use a compiler/linker for compilation/linkage, which fulfills the requirements in [ISO26262], part 8, clause 11.4.

Category:	Comment	Keywords:		ID:	232263

TTTech has tested with gcc 3.4.5.

Category:	Requirement	Keywords:		ID:	270666
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall make sure that the AUTOSAR- and S-WdgM Stack files used for compilation of the Verifier are the files used in the system where the S-WdgM is integrated.

Category:	Comment	Keywords:		ID:	263812

This is a list of files needed for building the Verifier (other files may be required for compilation depending on the environment and configuration options):

S-WdgM header files:

- WdgM.h
- WdgM_Cfg.h

S-WdgIf header files:

- WdgIf_Cfg.h
- WdgIf_Types.h

Created S-WdgM "Info file" (XSLT result):

- wdgm_verifier_info.h

Generated S-WdgM header files:

- WdgM_Cfg_Features.h
- **AS3**: WdgM_MemMap.h, or
- **AS4**: WdgM_OSMemMap.h
- WdgM_PBcfg.h

Files from the S-WdgM Stack package:

- wdgm_verifier.h
- wdgm_verifier_types.h
- wdgm_verifier_version.h

List of platform specific files:

- Compiler.h
- Compiler_Cfg.h
- MemMap.h
- Os.h
- Os_MemMap.h
- Platform_Types.h
- Std_Types.h
- Rte_Compiler_Cfg.h (if RTE is used)
- Rte_MemMap.h (if RTE is used)
- Rte_Type (if RTE is used)

Category:	Comment	Keywords:		ID:	263833

The set of include commands (-I*path*) for all include paths to these files is referred to *verify-includes*.

Category:	Requirement	Keywords:		ID:	263825
Label:		Safety relevant:			
Related To:		Related To':			

For the compilation process, the following files must be compiled and linked:
The generated C file:

- WdgM_PBcfg.c

Created S-WdgM Info file:

- wdgm_verifier_info.c

Files from the S-WdgM Stack package:

- wdgm_verifier.dll
- libwdgm_verifierdll.a

Category:	Requirement	Keywords:		ID:	269558
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall ensure that the output files of the S-WdgM Generator are used as input for the S-WdgM Verifier executable - and no other file.

Category:	Requirement	Keywords:		ID:	269560
Label:		Safety relevant:			
Related To:		Related To':			

Do not use S-WdgM Generator output files from previous generation processes, like from former versions of the S-WdgM package.

Category:	Comment	Keywords:		ID:	264066

The syntax for the compilation call is:

gcc -Wall wdgm_verifier_info.c callbacks.c WdgM_PBcfg.c *verify-includes* -L*dll-path* -lwdgm_verifierdll -o wdgm_verifier.exe

where

- *verify-includes* is a placeholder for the path(s) of include files as described above and
- *dll-path* is a placeholder for the path where wdgm_verifier.dll and libwdgm_verifierdll.a are located.

Category:	Comment	Keywords:		ID:	229699

In case of an error free application of the compiler/linker the output is a S-WdgM Verifier executable (wdgm_verifier.exe).

10.3.1.3 Verifier Run

Category:	Comment	Keywords:		ID:	229691

When the S-WdgM Verifier executable has been built, it has to be executed.
The S-WdgM Verifier writes a verification report to standard output 'stdout'.
This report must be reviewed as stated in this section and section "Manual Verification Checks" below.

Category:	Requirement	Keywords:		ID:	229695
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall run the S-WdgM Verifier executable as follows:

```
    wdgm_verifier.exe > verifier_report.txt.
```

Category:	Requirement	Keywords:		ID:	228861
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall review the output report of the S-WdgM Verifier executable run as follows:

If

- there is a summary titled "S U M M A R Y" at the end of the verification result and
- the summary shows all tests as PASSED,

then

the verification process ends with no error and the generated files can be considered correct

otherwise

the verification failed.

Category:	Comment	Keywords:		ID:	263882

If a test in the summary shows FAILED, then check the test information in the result:
Each test shows

- a description and
- the test result.

10.3.2 Manual Verification Checks

Category:	Comment	Keywords:		ID:	284770

The following checks can not be performed automatically but need to be done manually as described here.

Category:	Requirement	Keywords:		ID:	284772
Label:		Safety relevant:			
Related To:		Related To':			

For the following arrays in WdgM_PBcfg.c, the array length must match the number of items in the array:

- WdgMTransition
- WdgMGlobalTransition
- all arrays named StartsGlobalTransition_*se*_*cp*_*i* (for a SE *se*, a CP *cp* and an integer *i*)
- WdgMCheckPoint
- WdgMSupervisedEntity
- WdgMTriggerMode
- WdgMWatchdogDevice

Category:	Comment	Keywords:		ID:	284774

Some array lengths are encapsulated with defines like "WdgMCheckPoint [NR_OF_CHECKPOINTS]". The defines can be found at the top of file WdgM_PBcfg.c.

Category:	Requirement	Keywords:		ID:	290776
Label:		Safety relevant:			
Related To:		Related To':			

In WdgM_PBcfg.c, WdgMTicksPerSecond and WdgMTriggerWindowStart in array WdgMTriggerMode shall meet the condition

round (WdgMTicksPerSecond * WdgMTriggerWindowStart * 0.001) <= 65535

where

round (*x*) rounds *x* to the closest integer value (e.g. round(3.3)=3, round(3.5)=4, round(3.7)=4).

Category:	Requirement	Keywords:		ID:	290778
Label:		Safety relevant:			
Related To:		Related To':			

In WdgM_PBcfg.c, WdgMTicksPerSecond and WdgMTriggerTimeout in array WdgMTriggerMode shall meet the condition

round (WdgMTicksPerSecond * WdgMTriggerTimeout * 0.001) <= 65535

where

round (*x*) rounds *x* to the closest integer value (e.g. round(3.3)=3, round(3.5)=4, round(3.7)=4).

Category:	Requirement	Keywords:		ID:	290780
Label:		Safety relevant:			
Related To:		Related To':	__MKSID__294315		

In WdgM_PBcfg.c, check the array WdgMTransition:

For each item in the array:

CheckpointSourceId shall be set to an index that is in the range 0..*NrOfCheckpoints*-1;
where *NrOfCheckpoints* is the value of the struct member "NrOfCheckpoints" of the corresponding Supervised Entity; i.e., that Supervised Entity where the local transition starts and ends.

Category:	Comment	Keywords:		ID:	290782

For example: If WdgMCheckPoint has length 3, then only the indices 0, 1 and 2 are valid.

Category:	Requirement	Keywords:		ID:	290784
Label:		Safety relevant:			
Related To:		Related To':	__MKSID__294323		

In WdgM_PBcfg.c, check the array WdgMGlobalTransition:
For each item in the array:
CheckpointSourceId shall be set to an index that is in the range 0..*NrOfCheckpoints*-1;
where *NrOfCheckpoints* is the value of the struct member "NrOfCheckpoints" of the corresponding Supervised Entity; i.e. that Supervised Entity where the global transition starts.

Category:	Comment	Keywords:		ID:	290788

For example: If WdgMCheckPoint has length 3, then only the indices 0, 1 and 2 are valid.

Category:	Requirement	Keywords:		ID:	290790
Label:		Safety relevant:			
Related To:		Related To':	__MKSID__294313		

In WdgM_PBcfg.c, check the array WdgMGlobalTransition:
For each item in the array:
EntitySourceId shall be set to an index that is in the range 0..WDGM_NR_OF_ENTITIES-1.

Category:	Comment	Keywords:		ID:	290801

For example: If WdgMCheckPoint has length 3, then only the indices 0, 1 and 2 are valid.

Category:	Requirement	Keywords:		ID:	290792
Label:		Safety relevant:			
Related To:		Related To':			

In WdgM_PBcfg.c, check the array WdgMGlobalTransition:
For each item in the array:
Field WdgMCheckpointLocInitialId shall be set to 0.

Category:	Requirement	Keywords:		ID:	290804
Label:		Safety relevant:			
Related To:		Related To':	__MKSID__294082		

In WdgM_PBcfg.c, check the array WdgMSupervisedEntity :
For each item in the array:
Field WdgMCheckpointRef shall have a value of form &WdgMCheckPoint [*i*], where *i* is in range 0..WDGM_NR_OF_CHECKPOINTS-1.

Category:	Comment	Keywords:		ID:	290806

For example: If WdgMCheckPoint has length 3, then only the indices 0, 1 and 2 are valid.

Category:	Requirement	Keywords:		ID:	290808
Label:		Safety relevant:			
Related To:		Related To':			

In WdgM_PBcfg.c, check the array WdgMSupervisedEntity :
For each item in the array:
WdgMCheckpointLocInitialId shall be set to an index that is within the length of array WdgMCheckPoint.

Category:	Comment	Keywords:		ID:	290812

For example: If WdgMCheckPoint has length 3, then only the indices 0, 1 and 2 are valid.

Category:	Requirement	Keywords:		ID:	290814
Label:		Safety relevant:			
Related To:		Related To':			

In wdgm_verifier_info.c, check the array triggers:
For each item in the array:
Field WdgMTriggerModeId shall be equal to the position of the item in the array, where the first item is considered to have position 0.

Category:	Comment	Keywords:		ID:	290816

I.e. the first item has WdgMTriggerModeId set to 0, the next item has WdgMTriggerModeId set to 1, and so on.

Category:	Requirement	Keywords:		ID:	290818
Label:		Safety relevant:			
Related To:		Related To':			

In wdgm_verifier_info.c, check the array deadline_supervisions:
There shall be no two items in the array with

- the same source entity and
- the same source CP and
- the same destination entity and
- the same destination CP.

Category:	Requirement	Keywords:		ID:	290820
Label:		Safety relevant:			
Related To:		Related To':			

In wdgm_verifier_info.c, check the array deadline_supervisions:
For each item in the array, there shall exist a transition

- in local_transitions or
- in global_transitions

so that all for fields

- source entity
- source CP
- destination entity
- destination CP

are pairwise equal.

Category:	Comment	Keywords:		ID:	290794

That is: for every deadline supervision item there shall be a Local Transition or Global Transition defined.

Category:	Requirement	Keywords:		ID:	290796
Label:		Safety relevant:			
Related To:		Related To':			

Check if
- array WdgMCheckPoint in WdgM_PBcfg.c and
- array alive_supervisions in wdgm_verifier_info.c

match to each other:

```
For each item CP_item in WdgMCheckPoint:
  If WdgMAliveLRef is unequal NULL_PTR (i.e. Alive Supervision is configured),
  then
    there shall be an item AS_item in array alive_supervisions so that:
• source entity in AS_item matches the SE to which the CP in CP_item belongs,
• source CP in AS_item matches the CP referred in CP_item
• alive indications in AS_item matches WdgMExpectedAliveIndications in CP_item,
• minimum margin in AS_item matches WdgMMinMargin in CP_item
• maximum margin in AS_item matches WdgMMaxMargin in CP_item
• supervision Reference Cycle in AS_item matches WdgMSupervisionReferenceCycle in CP_item
 Otherwise (if WdgMAliveLRef is equal NULL_PTR i.e. no Alive Supervision is configured),
 then
     no AS_item in array alive _supervision shall exist that matches CP_item in all 6 fields as described
     below.
```

Category:	Requirement	Keywords:		ID:	555550
Label:		Safety relevant:			
Related To:		Related To':	__MKSID__552565		

In wdgm_verifier_info.c, check the line "AUTOSAR Version: *AUTOSAR namespace*"

```
If the ECU description file is AUTOSAR 4.0 compliant then
  AUTOSAR namespace shall be a 4.0 namespace
else If the ECU description file is AUTOSAR 3.1 compliant then
  AUTOSAR namespace shall be a 3.1 namespace
```

Category:	Comment	Keywords:		ID:	560002

An example for an *AUTOSAR namespace:*

AS4: "http://autosar.org/schema/r4.0"
AS3: "http://autosar.org/3.1.4"

Category:	Requirement	Keywords:		ID:	555591
Label:		Safety relevant:			
Related To:	__MKSID__304557,__MKSID__304553,__MKSID__304567	Related To':			

In WdgM_PBcfg.c, check that the declarations of the following identifiers are placed into the global memory segment of the S-WdgM:
- StatusG,
- EntityStatusG_*seid*, for every defined SE *seid*, and
- Alive_CounterG_*acid*, for every Alive Counter *acid* if Alive Counters are configured for the respective

supervised entity.

The declarations must be memory mapped using the following defines:

- WDGM_GLOBAL_START_SEC_VAR_NOINIT_UNSPECIFIED and
- WDGM_GLOBAL_STOP_SEC_VAR_NOINIT_UNSPECIFIED.

Category:	Requirement	Keywords:		ID:	555593
Label:		Safety relevant:			
Related To:		Related To':	__MKSID__304565,__MKSID__304563,__MKSID__304561		

In WdgM_PBcfg.c, check that the declarations of the following identifiers are placed into the global shared memory segment of the S-WdgM:

- StatusGS,
- EntityGS, and
- GlobalTransitionFlagsGS, which exists only if Global Transitions are defined in the system.

The declarations must be memory mapped using the following defines:

- WDGM_GLOBAL_SHARED_START_SEC_VAR_NOINIT_UNSPECIFIED and
- WDGM_GLOBAL_SHARED_STOP_SEC_VAR_NOINIT_UNSPECIFIED.

Category:	Requirement	Keywords:		ID:	555599
Label:		Safety relevant:			
Related To:	__MKSID__304559,__MKSID__304555	Related To':			

In WdgM_PBcfg.c, check that the declarations of the following identifiers are placed into the entity local data memory segment of the S-WdgM:

- EntityStatusL_*seid*, for every defined SE *seid*, and
- Alive_CounterL_*acid*, for every Alive Counter *acid* if Alive Counters are configured for the respective SE.

The declaration of EntityStatusL_*seid* must be memory mapped using the following defines:

- WDGM_*seid*_START_SEC_VAR_NOINIT_UNSPECIFIED and
- WDGM_*seid*_STOP_SEC_VAR_NOINIT_UNSPECIFIED

The declaration of AliveCounterL_*acid* must be memory mapped using the following defines:

- WDGM_*acid*_START_SEC_VAR_NOINIT_32BIT and
- WDGM_*acid*_STOP_SEC_VAR_NOINIT_32BIT.

Category:	Requirement	Keywords:		ID:	565665
Label:		Safety relevant:			
Related To:		Related To':			

In WdgM_PBcfg.h, check that constant value WDGM_NR_OF_WATCHDOGS matches the actual number of configured Watchdog devices.

Category:	Requirement	Keywords:		ID:	565673
Label:		Safety relevant:			
Related To:		Related To':			

In WdgM_PBcfg.h, check that constant value WDGM_NR_OF_TRIGGER_MODES matches the actual number of configured Watchdog Manager Trigger Modes.

Category:	Requirement	Keywords:		ID:	566072
Label:		Safety relevant:			
Related To:		Related To':			
Check that the constant value WDGM_NR_OF_ALLOWED_CALLERS matches the number of IDs of modules which the WdgM_SetMode function.					

Category:	Requirement	Keywords:		ID:	566082
Label:		Safety relevant:			
Related To:		Related To':			
If WDGM_NR_OF_ALLOWED_CALLERS is greater than zero, check that the struct member WdgMCallersRef` in WdgM_ConfigType points to an array of WdgM_CallersType which has a length of WDGM_NR_OF_ALLOWED_CALLERS					

Category:	Requirement	Keywords:		ID:	566084
Label:		Safety relevant:			
Related To:		Related To':			
If WDGM_NR_OF_ALLOWED_CALLERS is zero, check that that the struct member WdgMCallersRef` in WdgM_ConfigType is set to NULL.					

10.3.3 Check System Specifications against S-WdgM Info Files

Category:	Comment	Keywords:		ID:	265499
As part of the verification process, the generated files wdgm_verifier_info.c must be checked against the system specification, which served as base for the ECU description.					

Category:	Comment	Keywords:		ID:	265501
The following instructions show how to extract the data to be checked from the wdgm_verifier_info.c file. This involves analysis of C-source code and assumes basic knowledge in the programming language.					

Category:	Comment	Keywords:		ID:	265504
Check the generated Local Transitions as follows:					

Category:	Comment	Keywords:		ID:	265508
Find the C-struct array named "local_transition".					

Category:	Comment	Keywords:		ID:	265522
The array holds all Local Transitions of all SEs. Each Local Transition *lt* is given as a C-struct containing the following values (in this order): • the name of the source SE of *lt* • the name of the source CP of *lt* • the name of the destination SE of *lt* and • the name of the destination CP of *lt.*					

Category:	Requirement	Keywords:		ID:	265526
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall check that each *lt* is defined as stated in the System Specification.					

Category:	Requirement	Keywords:		ID:	265528
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall check also that no local transition stated in the System Specification is missing in the array "local_transitions".					

Category:	Comment	Keywords:		ID:	265587
Check the generated Global Transitions as follows:					

Category:	Comment	Keywords:		ID:	265589
Find the C-struct array named "global_transition".					

Category:	Comment	Keywords:		ID:	265591
The array holds all Global Transitions of all SEs. Each Global Transition *gt* is given as a C-struct containing the following values (in this order): • name of the source SE of *gt* • name of the source CP of *gt* • name of the destination SE of *gt* and • name of the destination CP of *gt.*					

Category:	Requirement	Keywords:		ID:	265593
Label:		Safety relevant:			
Related To:		Related To':			
Check that each *gt* is defined as stated in the System Specification.					

Category:	Requirement	Keywords:		ID:	265595
Label:		Safety relevant:			
Related To:		Related To':			
Check also that no Global Transition stated in the System Specification is missing in the array "global_transitions".					

Category:	Comment	Keywords:		ID:	265597
Check the CPs as follows:					

Category:	Comment	Keywords:		ID:	265599
For each defined SE named *se* find the C-struct array named "se_*se*_cp_list".					

Category:	Comment	Keywords:		ID:	265601
The array holds all CPs of all SEs. Within se_*se*_cp_list, each CP *cp* that is associated to *se* is given as a C-struct containing the following values (in this order):					

- ID of *se*
- ID of *cp*
- name of *se* and
- name of *cp.*

Category:	Requirement	Keywords:		ID:	265603
Label:		Safety relevant:			
Related To:		Related To':			

Check that each *cp* is defined in *se* as stated in the System Specification.

Category:	Requirement	Keywords:		ID:	265605
Label:		Safety relevant:			
Related To:		Related To':			

Check also that no CP for *se* stated in the System Specification is missing in the array "se_*se*_cp_list".

Category:	Comment	Keywords:		ID:	265607

At the end you have checked all CPs of all SEs.

Category:	Comment	Keywords:		ID:	265611

Check the SEs as follows:

Category:	Comment	Keywords:		ID:	265613

Find the C-struct array named "entities".

Category:	Comment	Keywords:		ID:	265615

The array holds information about all SEs.
Each SE *se* is given as a C-struct containing the following values (in this order):

- ID of *se*
- name of *se*
- number of CPs associated to *se* and
- a reference se_*se*_cp_list, which refers to a list of CPs for *se* that has been checked in step "Check the CPs as follows" (265597) above.

Category:	Requirement	Keywords:		ID:	265617
Label:		Safety relevant:			
Related To:		Related To':			

Check that each *se* is defined as stated in the System Specification.

Category:	Requirement	Keywords:		ID:	265619
Label:		Safety relevant:			
Related To:		Related To':			

Check also that no SE stated in the System Specification is missing in the array "entities".

Category:	Comment	Keywords:		ID:	265621

Check the deadline supervisions as follows:

Category:	Comment	Keywords:		ID:	265623

Find the C-struct array named "deadline_supervisions".

Category:	Comment	Keywords:		ID:	265625

The array holds information about all transitions with Deadline Supervision.
Each deadline supervision *dl* is given as a C-struct containing the following values (in this order):

- name of the source SE of *dl*
- name of the source CP of *dl*
- name of the destination SE of *dl*
- name of the destination CP of *dl*
- minimum value of the deadline interval of *dl* and
- maximum value of the deadline interval of *dl.*

Category:	Requirement	Keywords:		ID:	265627
Label:		Safety relevant:			
Related To:		Related To':			

Check that each defined *dl* is as stated in the System Specification.

Category:	Requirement	Keywords:		ID:	265629
Label:		Safety relevant:			
Related To:		Related To':			

Check also that no deadline supervision stated in the System Specification is missing in the array "deadline_supervisions".

Category:	Comment	Keywords:		ID:	265639

Check the Alive Supervision as follows:

Category:	Comment	Keywords:		ID:	265641

Find the C-struct array named "alive_supervisions".

Category:	Comment	Keywords:		ID:	265643

The array holds information about all transitions with Alive Supervision.
Each Alive Supervision *as* is given as a C-struct containing the following values (in this order):

- name of the source SE of *al*
- name of the source CP of *al*
- number of expected alive indications per Reference Cycle of *al*
- minimum value of the alive indication margin of *al* and
- maximum value of the alive indication margin of *al.*

Category:	Requirement	Keywords:		ID:	265645
Label:		Safety relevant:			
Related To:		Related To':			

Check that each defined *al* is as stated in the System Specification.

Category:	Requirement	Keywords:		ID:	265647
Label:		Safety relevant:			
Related To:		Related To':			

Check also that no Alive Supervision stated in the System Specification is missing in the array "alive_supervisions".

11 Safe Watchdog Manager

Category:	Comment	Keywords:		ID:	228907

This section lists the safety requirements for the integration and application of the S-WdgM code in(to) an AUTOSAR system.

11.1 API Specification

Category:	Comment	Keywords:		ID:	228909

This section describes the imported types and definitions and the expected interface. It also describes safety related aspects of types, definitions and functions implemented in the S-WdgM.
Some types, definitions and interfaces depend on the used S-WdgM Configuration.

Category:	Comment	Keywords:		ID:	229196

For a detailed description of types, definitions and functions implemented in S-WdgM, see [TT_WDGM_UM].
For a detailed description of types, definitions and functions imported from S-WdgIf, see [TT_WDGIF_UM].

Category:	Comment	Keywords:		ID:	229302

For further requirements related to imported types, definitions and interfaces, see section "Integration".

Category:	Requirement	Keywords:		ID:	229304
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for the correct import of the types and definitions that are listed in this section.

Category:	Requirement	Keywords:		ID:	229306
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for the correct application of the interface functions.

Category:	Comment	Keywords:		ID:	542988

Correct in this context means that the interface functions are used in accordance with the requirements given in this document. See also section "Application Level API Functions" below.

Category:	Requirement	Keywords:		ID:	229744
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for ensuring that all external functions that are called from within the S-WdgM code are imported from the correct versions of AUTOSAR.

Category:	Comment	Keywords:		ID:	558694

The external functions are listed in section "Expected Interface" below. The correct AUTOSAR version is defined in 231307.

Category:	Requirement	Keywords:		ID:	229746
Label:		Safety relevant:			
Related To:		Related To':			

The inclusion of AUTOSAR files or any other files different from S-WdgM files shall not redefine any identifier that is defined in the S-WdgM code. E.g., redefinitions with #define macros.

Category:	Requirement	Keywords:		ID:	231825
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall verify that no external interface with the S-WdgM degrades the quality level of the S-WdgM below the required quality level.

Category:	Comment	Keywords:		ID:	231827

For example, if an external function of quality level ASIL C is called by the S-WdgM, it degrades the quality level of the S-WdgM to ASIL C (if no precautions were taken), although the required quality level is ASIL D.

Category:	Comment	Keywords:		ID:	558698

The external interface is listed in section "Expected Interface" below.

11.1.1 Expected Interface

Category:	Comment	Keywords:		ID:	229201

This section lists external functions that are called by the S-WdgM.

Category:	Comment	Keywords:		ID:	229715

For a scheme with interaction of the S-WdgM with external functions, see [TT_WDGM_UM].

Category:	Comment	Keywords:		ID:	234840

The following functions of the lower WdgIf layer are called independent to the chosen S-WdgM configuration:

Function	Module
WdgIf_SetMode ()	WdgIf
WdgIf_SetTriggerWindow ()	WdgIf

table 8

Category:	Comment	Keywords:		ID:	229726

Some functions are called by the S-WdgM depending on the compiler switches as listed here:

Compiler Switch	Function	Module
WDGM_DEM_REPORT is set to STD_ON	Appl_Dem_ReportErrorStatus () **)	DEM
WDGM_DEV_ERROR_DETECT is set to STD_ON	Appl_Det_ReportError () **)	DET
WDGM_SECOND_RESET_PATH is set to STD_ON	Appl_Mcu_PerformReset () **)	Mcu
WDGM_USE_OS_SUSPEND_INTERRUPT is set to STD_ON	**AS3:** SchM_Enter_WdgM () and SchM_Exit_WdgM () **AS4:** SchM_Enter_WdgM_WDGM_EXCLUSIVE_AREA_0 () and SchM_Exit_WdgM_WDGM_EXCLUSIVE_AREA_0 ()	SchM
WDGM_STATE_CHANGE_NOTIFICATION is set to STD_ON	*WdgM_GlobalStateChangeCbk* () *), *WdgM_LocalStateChangeCbk* ()	*)
WDGM_TIMEBASE_SOURCE is set to WDGM_INTERNAL_HARDWARE_TICK	WdgIf_GetTickCounter ()	WdgIf

table 9

If a compiler switch is set differently, the according function is not called by the S-WdgM.

*) The actual name of the function is defined by the S-WdgM configuration fields WdgM_GlobalStateChangeCbk and WdgM_LocalStateChangeCbk, respectively. The actual module depends on the system architecture.

**) This is a wrapper function. See the next section for information.

11.1.1.1 Implementation of Wrapper Functions for the Expected Interface

Category:	Comment	Keywords:		ID:	238249

Some functions of the expected interface may not meet the required quality level and need to be wrapped so that freedom from interference with the S-WdgM is guaranteed. These functions are:

Function	Wrapper function
Dem_ReportErrorStatus ()	Appl_Dem_ReportErrorStatus ()
Det_ReportError ()	Appl_Det_ReportError ()
Mcu_PerformReset ()	Appl_Mcu_PerformReset ()

table 10

Category:	Comment	Keywords:		ID:	260668

Note: Whether a function is called or not depends on the configuration's compiler switches.

Category:	Requirement	Keywords:		ID:	238245
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for the implementation of each wrapper function as follows:

1. the wrapper function serves as wrapper for the call of the according external function,
2. the wrapper function guarantees freedom from interference with the S-WdgM code and data when the according function is called, and
3. the quality level of the wrapper function is sufficient for the required quality level of the system.

Category:	Requirement	Keywords:		ID:	259941
Label:		Safety relevant:			
Related To:		Related To':			

The wrapper function shall be declared in a separate header-file, which shall include the header file for wrapped AUTOSAR function as follows:

Wrapper Function	Declared In Header File	Header File includes
Appl_Dem_ReportErrorStatus ()	Appl_Dem.h	Dem.h
Appl_Det_ReportError ()	Appl_Det.h	Det.h
Appl_Mcu_PerformReset ()	Appl_Mcu.h	Mcu.h

table 11

Category:	Requirement	Keywords:		ID:	229211
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall verify:
If a function in 234840, 229726, 238249, and 259941above is called, then the quality level of the S-WdgM is not degraded below the required quality level.

Category:	Comment	Keywords:		ID:	260560

If a subset of these functions is called, then the quality level of the S-WdgM is degraded to the quality level of the function in this subset that has the lowest quality level.

Category:	Comment	Keywords:		ID:	229728

For this reason, the integrator is advised to revise the necessity of the expected interfaces.

11.1.2 Imported Types and Definitions

Category:	Comment	Keywords:		ID:	229213

This section lists the types and definitions that are imported by the S-WdgM.

Category:	Comment	Keywords:		ID:	229296

The following types and definitions are imported from Platform_Types.h and used:
Types:
uint8
uint16

uint32
boolean
Definitions:
TRUE
FALSE

Category:	Comment	Keywords:		ID:	229310

The following types and definitions are imported from Std_Types.h and used:
Types:
Std_ReturnType
Definitions:
STD_ON
STD_OFF

Category:	Comment	Keywords:		ID:	235906

The type Std_VersionInfoType is not included, because the WdgM_GetVersionInfo () is implemented as macro.

Category:	Comment	Keywords:		ID:	229312

The following definitions are imported from "Compiler.h" and used:
Definitions:
AUTOMATIC
CONST
FUNC
NULL_PTR
P2CONST
P2FUNC
P2VAR
VAR

Category:	Comment	Keywords:		ID:	229318

The following definitions are imported from "Compiler_Cfg.h" and used:
WDGM_CODE
WDGM_CONST
WDGM_APPL_CONST
WDGM_APPL_DATA
WDGM_APPL_VAR
WDGM_VAR

Category:	Comment	Keywords:		ID:	290334

The following definitions are imported from " SchM_WdgM.h" and used:
WDGM_EXCLUSIVE_AREA_0

Category:	Comment	Keywords:		ID:	290336

The following definitions are imported from " WdgIf_Types.h" and used:
WDGIF_OFF_MODE

Category:	Comment	Keywords:		ID:	290332

If WDGM_USE_RTE is set to STD_ON, then the following definitions are imported from "Rte_Type.h" (for **AS3)** or **"**Rte_WdgM_Type.h" (for **AS4)**:
WDGM_LOCAL_STATUS_OK
WDGM_LOCAL_STATUS_FAILED
WDGM_LOCAL_STATUS_EXPIRED
WDGM_LOCAL_STATUS_DEACTIVATED
WDGM_GLOBAL_STATUS_OK
WDGM_GLOBAL_STATUS_FAILED
WDGM_GLOBAL_STATUS_EXPIRED
WDGM_GLOBAL_STATUS_STOPPED
WDGM_GLOBAL_STATUS_DEACTIVATED

Category:	Comment	Keywords:		ID:	229314

The following definitions are imported from "MemMap.h" (and indirectly from "WdgM_MemMap.h" (for **AS3)** or "WdgM_OSMemMap.h" (for **AS3)**) and used:

In WdgM.c:
WDGM_GLOBAL_START_SEC_VAR_32BIT
WDGM_GLOBAL_STOP_SEC_VAR_32BIT
WDGM_GLOBAL_START_SEC_VAR_BOOLEAN
WDGM_GLOBAL_STOP_SEC_VAR_BOOLEAN
WDGM_START_SEC_CODE
WDGM_STOP_SEC_CODE

In WdgM_Checkpoint.c:
WDGM_START_SEC_CODE
WDGM_STOP_SEC_CODE

In WdgM_PBcfg.c (generated):
WDGM_SE*seid*_START_SEC_VAR_NOINIT_UNSPECIFIED
WDGM_SE*seid*_STOP_SEC_VAR_NOINIT_UNSPECIFIED
WDGM_SE*seid*_START_SEC_VAR_NOINIT_32BIT
WDGM_SE*seid*_STOP_SEC_VAR_NOINIT_32BIT
(for a SE with WdgMSupervisedEntityId *seid*) and
WDGM_GLOBAL_START_SEC_VAR_NOINIT_UNSPECIFIED
WDGM_GLOBAL_STOP_SEC_VAR_NOINIT_UNSPECIFIED
WDGM_GLOBAL_SHARED_START_SEC_VAR_NOINIT_UNSPECIFIED
WDGM_GLOBAL_SHARED_STOP_SEC_VAR_NOINIT_UNSPECIFIED
WDGM_START_SEC_CONST_UNSPECIFIED
WDGM_STOP_SEC_CONST_UNSPECIFIED

Category:	Comment	Keywords:		ID:	290088

If a SE with WdgMSupervisedEntityId *seid* belongs to an application (WdgMAppTaskRef for SE *seid* is set to *appl_name*),
then the following defines in WdgM_MemMap.h (for **AS3**) or WdgM_OSMemMap.h (for **AS4**) are redefined:
WDGM_SE*seid*_START_SEC_VAR_NOINIT_UNSPECIFIED
WDGM_SE*seid*_STOP_SEC_VAR_NOINIT_UNSPECIFIED
WDGM_SE*seid*_START_SEC_VAR_NOINIT_32BIT
WDGM_SE*seid*_STOP_SEC_VAR_NOINIT_32BIT
is redefined to

*appl_name*_START_SEC_VAR_NOINIT_UNSPECIFIED
*appl_name*_STOP_SEC_VAR_NOINIT_UNSPECIFIED
*appl_name*_START_SEC_VAR_NOINIT_32BIT
*appl_name*_STOP_SEC_VAR_NOINIT_32BIT
respectively.

Category:	Comment	Keywords:		ID:	290118

If the S-WdgM component belongs to an application (WdgMGlobalMemoryAppTaskRef is set to *appl_name*),
then the following defines in WdgM_MemMap.h (for **AS3**) or WdgM_OSMemMap.h (for **AS4**) are redefined:
WDGM_GLOBAL_START_SEC_VAR_NOINIT_UNSPECIFIED
WDGM_GLOBAL_STOP_SEC_VAR_NOINIT_UNSPECIFIED
WDGM_GLOBAL_START_SEC_VAR_32BIT
WDGM_GLOBAL_STOP_SEC_VAR_32BIT
WDGM_GLOBAL_START_SEC_VAR_BOOLEAN
WDGM_GLOBAL_STOP_SEC_VAR_BOOLEAN
is redefined to
*appl_name*_GLOBAL_START_SEC_VAR_NOINIT_UNSPECIFIED
*appl_name*_GLOBAL_STOP_SEC_VAR_NOINIT_UNSPECIFIED
*appl_name*_GLOBAL_START_SEC_VAR_32BIT
*appl_name*_GLOBAL_STOP_SEC_VAR_32BIT
*appl_name*_GLOBAL_START_SEC_VAR_BOOLEAN
*appl_name*_GLOBAL_STOP_SEC_VAR_BOOLEAN
respectively.

Category:	Comment	Keywords:		ID:	290889

Defines for global shared data are also redefined:
WDGM_GLOBAL_SHARED_START_SEC_VAR_NOINIT_UNSPECIFIED
WDGM_GLOBAL_SHARED_STOP_SEC_VAR_NOINIT_UNSPECIFIED
is redefined to
GlobalShared_START_SEC_VAR_NOINIT_UNSPECIFIED
GlobalShared_STOP_SEC_VAR_NOINIT_UNSPECIFIED

Category:	Comment	Keywords:		ID:	229730

The following types are imported from "WdgIf_Types.h" (through "WdgM_Cfg.h") and used:
Type:
WdgIf_ModeType

Category:	Requirement	Keywords:		ID:	229235
Label:		Safety relevant:			
Related To:		Related To':			

If the configuration parameter WDGM_USE_RTE is set to STD_ON, then the integrator shall ensure that the following types are defined as shown in this table:

Type	Allowed Value
WdgM_SupervisedEntityIdType	uint8, uint16
WdgM_CheckpointIdType	uint8, uint16
WdgM_ModeType	uint8
WdgM_LocalStatusType	uint8

WdgM_GlobalStatusType	uint8

table 12

No other value is allowed.

Category:	Comment	Keywords:		ID:	229707

The S-WdgM assumes that "Rte_Type.h" (for **AS3**) or **"**Rte_WdgM_Type.h" (for **AS4**) is the source of these types and includes "Rte_Type.h" (for **AS3**) or **"**Rte_WdgM_Type.h" (for **AS4**) if - and only if - WDGM_USE_RTE is set to STD_ON.

Category:	Comment	Keywords:		ID:	237635

See [AS_RTE_SWS] for information on AUTOSAR RTE.

Category:	Comment	Keywords:		ID:	229288

If the configuration parameter WDGM_USE_RTE is set to STD_OFF, then the types are defined by the S-WdgM as shown in this table:

Type	Value
WdgM_SupervisedEntityIdType	uint16
WdgM_CheckpointIdType	uint16
WdgM_ModeType	uint8
WdgM_LocalStatusType	uint8
WdgM_GlobalStatusType	uint8

table 13

Category:	Requirement	Keywords:		ID:	229264
Label:		Safety relevant:			
Related To:		Related To':			

If the configuration parameter WDGM_USE_RTE is set to STD_ON, then the integrator shall ensure that the following definitions are set as shown in the following table:

Definition	Value
WDGM_LOCAL_STATUS_OK	0
WDGM_LOCAL_STATUS_FAILED	1
WDGM_LOCAL_STATUS_EXPIRED	2
WDGM_LOCAL_STATUS_DEACTIVATED	4
WDGM_GLOBAL_STATUS_OK	0
WDGM_GLOBAL_STATUS_FAILED	1
WDGM_GLOBAL_STATUS_EXPIRED	2
WDGM_GLOBAL_STATUS_STOPPED	3
WDGM_GLOBAL_STATUS_DEACTIVATED	4

table 14

Category:	Comment	Keywords:		ID:	229709

The S-WdgM assumes that "Rte_Type.h" (for **AS3**) or **"**Rte_WdgM_Type.h" (for **AS4**) is the source of these types and includes "Rte_Type.h" (for **AS3**) or **"**Rte_WdgM_Type.h" (for **AS4**) if - and only if - WDGM_USE_RTE is set to STD_ON.

Category:	Comment	Keywords:		ID:	237637

See [AS_RTE_SWS] for information on AUTOSAR RTE.

Category:	Comment	Keywords:		ID:	229292

If the configuration parameter WDGM_USE_RTE is set to STD_OFF then the status definitions are implemented by the S-WdgM with the values shown in the table above in requirement 229264.

11.1.3 Error Handling

Category:	Comment	Keywords:		ID:	229752

This section describes the error codes set by the S-WdgM using the DET or DEM mechanism and the return values from S-WdgM API functions.

11.1.3.1 DET Errors

Category:	Comment	Keywords:		ID:	229766

DET Errors are intended to support the development of an application. During software development, the compiler directive WDGM_DEV_ERROR_DETECT is usually set to STD_ON. Once the software is safe enough so that no further DET error can occur, the option is deactivated. For safety reasons the DET defines are listed here.

Category:	Comment	Keywords:		ID:	229742

If the compiler switch WDGM_DEV_ERROR_DETECT is set to STD_ON, then the S-WdgM reports the following development errors through the function Appl_Det_ReportError ():

Error	Code	Description
WDGM_E_NO_INIT	0x10	Uninitialized S-WdgM.
WDGM_E_PARAM_CONFIG	0x11	Invalid S-WdgM Configuration.
WDGM_E_PARAM_MODE	0x12	Invalid mode parameter (currently not used by the S-WdgM).
WDGM_E_PARAM_SEID	0x13	Wrong ID number of the SE.
WDGM_E_NULL_POINTER	0x14	Null pointer parameter.
WDGM_E_DISABLE_NOT_ALLOWED	0x15	Disabled Watchdog is not allowed.
WDGM_E_CPID	0x16	Invalid CP ID number.
WDGM_E_DEPRECATED	0x17	Using deprecated API service (currently not used by S-WdgM).
WDGM_E_TIMEBASE	0x28	Timebase counter failure.
WDGM_E_PARAM_STATE	0x29	Invalid S-WdgM state.
WDGM_E_WDGIF_MODE	0x2A	The WdgIf_SetMode(*mode*) function was called with an invalid *mode* parameter.
WDGM_E_MEMORY_FAILURE	0x2B	Corrupted S-WdgM memory.
WDGM_E_REENTRANCY	0x2C	Reentrancy not allowed.

table 15

These definitions are defined in WdgM.h.

Category:	Comment	Keywords:		ID:	229750

The definitions from 0x10 to 0x17 are AUTOSAR definitions (see [AS_WDGM_SWS]).
The definition from 0x28 to 0x2B are TTTech specific.

Category:	Requirement	Keywords:		ID:	229760
Label:		Safety relevant:			
Related To:	__MKSID__284531,_ _MKSID__284549,__ MKSID__261279,__ MKSID__261146,__ MKSID__261148,__ MKSID__261150,__ MKSID__263904,__ MKSID__283924,__ MKSID__261198,__ MKSID__261210,__ MKSID__261212,__ MKSID__268923,__ MKSID__284038,__ MKSID__284042,__ MKSID__268925,__ MKSID__284050,__ MKSID__268927,__ MKSID__284054,__ MKSID__268929,__ MKSID__284056,__ MKSID__268931,__ MKSID__284062,__ MKSID__268933,__ MKSID__284066,__ MKSID__268935	Related To':			

The integrator is responsible to make sure that - once the compiler switch WDGM_DEV_ERROR_DETECT is set to STD_OFF - no DET related error can occur.

11.1.3.2 DEM Errors

Category:	Comment	Keywords:		ID:	229748

ECU description fileIf the compiler switch WDGM_DEM_REPORT is set to STD_ON, then the S-WdgM reports the following production errors through the function Appl_Dem_ReportErrorStatus():

Error	**Code**	**Description**
AS3: WDGM_E_MONITORING *) **AS4:** DemConf_DemEventParameter_WDGM_E_MONITORING **)	0x30u	The system reached status WDGM_GLOBAL_STATUS_STOPPED
AS3: WDGM_E_IMPROPER_CALLER *) **AS4:** DemConf_DemEventParameter_WDGM_E_IMPROPER_CALLER **)	0x33u	The function is not permitted to call WdgM_SetMode ().

table 16

*) Note: The error definitions are defined in Dem.h
**) Note: The error definition and error code are defined by the user in the ECU description file and can

vary.

Category:	Requirement	Keywords:		ID:	229756
Label:		Safety relevant:			
Related To:	__MKSID__261188,__MKSID__261190	Related To':			

The integrator is responsible for correct handling and escalation of errors related to DEM according to the system requirements.

11.1.3.3 Return Values

Category:	Comment	Keywords:		ID:	229772

The following functions return E_NOT_OK in case an error occured:

Function	Comment
WdgM_CheckpointReached ()	Monitoring update failed.
WdgM_GetLocalStatus ()	Returning current monitoring status failed.
WdgM_GetGlobalStatus ()	Returning current monitoring status failed.
WdgM_PerformReset ()	Immediate reset of at least one Watchdog failed (if WDGM_SECOND_RESET_PATH is set to STD_ON).
WdgM_GetMode ()	Returning current WD Trigger Mode failed.
WdgM_SetMode ()	Changing to new WD Trigger Mode failed.
WdgM_DeactivateSupervisionEntity ()	Deactivating SE failed.
WdgM_ActivateSupervisionEntity ()	Activating SE failed.

table 17

Category:	Requirement	Keywords:		ID:	229782
Label:		Safety relevant:			
Related To:	__MKSID__284531,__MKSID__261188,__MKSID__261190	Related To':			

The integrator is responsible for correct handling and escalation of errors (according to the system requirements) indicated by the return value E_NOT_OK.

11.2 Functional Specification

Category:	Comment	Keywords:		ID:	283403

A detailed functional specification of the S-WdgM module is provided in [TT_WDGM_UDD].

Category:	Requirement	Keywords:		ID:	230494
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for ensuring that the S-WdgM functionality is not unintentionally affected by other software (especially the AUTOSAR application). This is, e.g., modification of data like tolerance value, counters, etc. that are used by the S-WdgM.

Category:	Comment	Keywords:		ID:	287738

This includes:

- memory corruption (see section "S-WdgM Application"),
- source code modification (intended and unintended), and
- API function calls with wrong parameters (see sections "Requirements For All Application Level API Functions" and "Requirements For All System Level API Functions" below).

11.3 S-WdgM Configuration

Category:	Comment	Keywords:		ID:	230543

The S-WdgM differs between two kinds of configuration:

- pre-processor options and
- post-build configuration data.

Category:	Comment	Keywords:		ID:	230545

The pre-processor options are generated out of an ECU configuration using the S-WdgM Generator (coded in the generated file WdgM_Cfg_Features.h).
They activate or deactivate certain S-WdgM features and cannot be altered during runtime.
See section "S-WdgM Configuration Generator" above for details on the S-WdgM Generator and its application.
See [TT_WDGM_UM] for details on the pre-processor options.

Category:	Comment	Keywords:		ID:	230547

The post-build configuration data is also generated out of the ECU configuration using the S-WdgM Generator (coded in the files WdgM_PBcfg.h and WdgM_PBcfg.c).
It defines certain values that affect the S-WdgM functionality (like tolerances or cycle length).
The S-WdgM can switch among these configurations at runtime. However, the current version of the S-WdgM supports only one mode. The configuration data itself can not be altered at runtime.
See section "S-WdgM Configuration Generator" above for details on the S-WdgM Generator and its application.
See [TT_WDGM_UM] for details on the post-build configuration data.

Category:	Requirement	Keywords:		ID:	230549
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for checking the pre-processor and post-build configuration values for the S-WdgM for plausibility and suitability for the system requirements (concerning correct function and timing behaviour) as depicted in section "Configuration Check-List" above.

Category:	Requirement	Keywords:		ID:	230532
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for generation and verification of configuration data as depicted in section "S-WdgM Configuration Generator" above.

Category:	Requirement	Keywords:		ID:	230551
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall guarantee that the configuration data is not altered at runtime, e.g. by erroneous HW.

Category:	Comment	Keywords:		ID:	230553

This can be realized - for example - with ECC ROM checks, cyclical ROM checks, and start up ROM checks.

11.4 File Structure

Category:	Comment	Keywords:		ID:	230234

For information about the S-WdgM file structure, see [TT_WDGM_UM].

Category:	Comment	Keywords:		ID:	230236

The following table shows the files that are only included when the according compiler directive is set to STD_ON:

Include File	**Compiler Directive**
Mcu.h	WDGM_SECOND_RESET_PATH
Det.h	WDGM_DEV_ERROR_DETECT
Dem.h	WDGM_DEM_REPORT
AS3: Rte_Type.h **AS4:** Rte_WdgM_Type.h	WDGM_USE_RTE
SchM_WdgM.h	WDGM_USE_OS_SUSPEND_INTERRUPT

table 18

Category:	Comment	Keywords:		ID:	230373

Also note that the configuration dependent memory mapping definitions for the S-WdgM are defined in the file WdgM_MemMap.h (for **AS3**) or WdgM_OSMemMap.h (for **AS4**), which is generated by the S-WdgM Generator. The configuration independent memory mapping definitions are defined in MemMap.h

The file WdgM_MemMap.h (for **AS3**) or WdgM_OSMemMap.h (for **AS4**) is included into MemMap.h, which is itself included into the S-WdgM source code.

Using the definitions in WdgM_MemMap.h (for **AS3**) or WdgM_OSMemMap.h (for **AS4**), the integrator can place the status variables of each SE in a separate address space (e.g., if the SE is part of an OS application then its data is placed in the same context as the application's data).

Category:	Comment	Keywords:		ID:	230242

See also the requirement 229746 for File inclusion.

11.5 S-WdgM Integration

Category:	Comment	Keywords:		ID:	230951

This section describes how to integrate the S-WdgM into a safety-relevant system.

Category:	Requirement	Keywords:		ID:	230957
Label:		Safety relevant:			
Related To:		Related To':			

It is the responsibility of the integrator to demonstrate that

- the failure detection mechanisms provided by the S-WdgM and
- the generated S-WdgM configuration

are sufficient for the considered system.

Category:	Requirement	Keywords:		ID:	230953
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for a correct integration of the S-WdgM code

- on application level and
- on system level.

Category:	Comment	Keywords:		ID:	558706

The integration of the S-WdgM is correct, when all system requirements are satisfied.

Category:	Requirement	Keywords:		ID:	231823
Label:		Safety relevant:			
Related To:	__MKSID__283518,__MKSID__283514	Related To':			

The integrator shall verify that the chosen WD device - internal or external - meets the system's safety requirements.

Category:	Comment	Keywords:		ID:	231896

For single oscillator MCU's (where the watchdog clock is derived from CPU main clock) it is recommended to use an external watchdog device with its own oscillator as well.

11.5.1 Import from AUTOSAR Definitions into S-WdgM

Category:	Requirement	Keywords:		ID:	230955
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for the correct implementation of all types and definitions that are imported from AUTOSAR header files and used by the S-WdgM code according to AUTOSAR specifications.

Category:	Requirement	Keywords:		ID:	230969
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for providing the AUTOSAR header files for the import of the AUTOSAR types and definitions.

Category:	Comment	Keywords:		ID:	230971

For a list of imported AUTOSAR types and definitions and the related header files, see section "Imported Types and Definitions" above.

Category:	Requirement	Keywords:		ID:	230979
Label:		Safety relevant:			
Related To:		Related To':			

The inclusion of AUTOSAR header files into S-WdgM code shall not redefine any identifier that is defined within the S-WdgM code. This prohibits, e.g., redefinitions with #define macros.

Category:	Requirement	Keywords:		ID:	230981
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for providing the correct code of used AUTOSAR functions. That is, correct in version and functionality.

Category:	Comment	Keywords:		ID:	230983

For a list of used AUTOSAR functions, see section "Expected Interface" above.
For the AUTOSAR version see 231307.

Category:	Requirement	Keywords:		ID:	231015
Label:		Safety relevant:			
Related To:		Related To':			

It is the responsibility of the integrator to provide a file Std_Types.h according to the descriptions and requirements in section "Imported Types and Definitions" above.

Category:	Requirement	Keywords:		ID:	231069
Label:		Safety relevant:			
Related To:		Related To':			

It is the responsibility of the integrator to provide a file Platform_Types.h according to the descriptions and requirements in section "Imported Types and Definitions" above.

Category:	Requirement	Keywords:		ID:	231017
Label:		Safety relevant:			
Related To:		Related To':			

It is the responsibility of the integrator to provide a file Compiler.h and a file Compiler_Cfg.h according to the descriptions and requirements in section "Imported Types and Definitions" above.

Category:	Comment	Keywords:		ID:	230977

Note that some other integrated products, provide their own contents for Compiler_Cfg.h. They need to be merged into the systems Compiler_Cfg.h.

Category:	Requirement	Keywords:		ID:	231025
Label:		Safety relevant:			
Related To:		Related To':			
It is the responsibility of the integrator to provide a file MemMap.h according to AUTOSAR specifications.					

Category:	Comment	Keywords:		ID:	231075
Some other integrated products, provide their own contents for MemMap.h. They need to be merged into the system's MemMap.h file.					

Category:	Requirement	Keywords:		ID:	260767
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall include the generated file WdgM_MemMap.h (for **AS3**) or WdgM_OSMemMap.h (for **AS4**) in the file MemMap.h.					

Category:	Requirement	Keywords:		ID:	260828
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall place the inclusion of WdgM_MemMap.h (for **AS3**) or WdgM_OSMemMap.h (for **AS4**) before Os_MemMap.h in MemMap.h.					

Category:	Comment	Keywords:		ID:	260769
WdgM_MemMap.h (for **AS3**) or WdgM_OSMemMap.h (for **AS4**) contains S-WdgM configuration dependent definitions. See also section "Memory Mapping" below.					

Category:	Comment	Keywords:		ID:	231077
TTTech provides example files for MemMap.h (with include commands of WdgM_MemMap.h (for **AS3**) or WdgM_OSMemMap.h (for **AS4**)) and a file demo_MemMap.h (with the memory mapping definitions of the complete S-WdgM Stack).					

11.5.2 Memory Mapping

Category:	Comment	Keywords:		ID:	231283
This section lists the requirements for the memory mapping of the S-WdgM data and code (also the generated S-WdgM code). For a detailed description on how to manage S-WdgM memory sections, see [TT_WDGM_UM].					

Category:	Requirement	Keywords:		ID:	231029
Label:		Safety relevant:			
Related To:		Related To':			
The integrator is responsible for • the generation of the file WdgM_MemMap.h (for **AS3**) or WdgM_OSMemMap.h (for **AS3**) as described in section "S-WdgM Configuration Generator" above and • its inclusion into the file MemMap.h which is itself included into the S-WdgM code.					

Category:	Comment	Keywords:		ID:	231484

TTTech provides a sample file WdgM_MemMap.h (for **AS3**) or WdgM_OSMemMap.h (for **AS4**).

Category:	Requirement	Keywords:		ID:	231277
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is also responsible for the correct assignment of data and code of the S-WdgM (including the generated S-WdgM code) to the various memory sections according to the memory mapping keywords provided by the S-WdgM.

Category:	Comment	Keywords:		ID:	231289

For the memory sections that are supported by the S-WdgM see comment 229314 in section "Imported Types and Definitions" above.

Category:	Requirement	Keywords:		ID:	231281
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall assign the data for each SE to the corresponding address space of the SWC address area where the SE is located.

Category:	Comment	Keywords:		ID:	290510

See parameter WdgMAppTaskRef in [TT_WDGM_UM].

Category:	Requirement	Keywords:		ID:	231454
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall assign global data to a address space with

- read access for all tasks and applications and
- read/write access for the S-WdgM.

Category:	Requirement	Keywords:		ID:	231462
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall assign global shared data to an address space with read/write access for all tasks and applications.

Category:	Comment	Keywords:		ID:	290512

See parameter WdgMGlobalMemoryAppTaskRef in [TT_WDGM_UM].

Category:	Comment	Keywords:		ID:	231474

All S-WdgM global shared data is protected by the S-WdgM against corruption

Category:	Requirement	Keywords:		ID:	231972
Label:		Safety relevant:			
Related To:	__MKSID__261238,_ _MKSID__261216	Related To':			

In a system that uses MCU memory protection, the S-WdgM global data and variables shall be placed in a separate memory section that can not be corrupted by other software modules or hardware failures.

11.5.3 S-WdgM Files

Category:	Requirement	Keywords:		ID:	231035
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall ensure that only

- files of a single delivered package and
- files generated with tools of this package

are installed:

These are the files:

- WdgM_PBCfg.h (generated),
- WdgM_PBCfg.c (generated),
- WdgM_Cfg_Features.h (generated),
- WdgM_Cfg.h,
- WdgM.h,
- WdgM.c, and
- WdgM_Checkpoint.c

Category:	Requirement	Keywords:		ID:	230229
Label:		Safety relevant:			
Related To:		Related To':			

The loaded S-WdgM Configuration shall be compatible with the S-WdgM embedded code.

Category:	Comment	Keywords:		ID:	289588

The S-WdgM performs a version check with every call of WdgM_Init ().

11.5.4 Compilation and Linkage

Category:	Requirement	Keywords:		ID:	230959
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for compilation of the S-WdgM code with a compiler that is compliant to ANSI ISO/IEC 9899:1990.

Category:	Comment	Keywords:		ID:	230963

The generated code is compliant to ANSI ISO/IEC 9899:1990. It is also known as "ANSI C (C89)" and "ISO C (C90)".

Category:	Requirement	Keywords:		ID:	230991
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for correct compilation and linkage of the S-WdgM into the AUTOSAR system.

Category:	Requirement	Keywords:		ID:	231079
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall guarantee that the compiled and linked target binary is correctly loaded into the target system.

11.5.5 S-WdgM Stack Requirements

Category:	Requirement	Keywords:		ID:	231547
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall make sure that the S-WdgM communicates with least

- an internal WD device (MCU inside) or
- an external WD device.

Category:	Requirement	Keywords:		ID:	231549
Label:		Safety relevant:			
Related To:		Related To':			

For ASIL D systems, an external monitoring facility shall be used.

Category:	Comment	Keywords:		ID:	231551

This is highly recommended in ISO 26262 (see [ISO26262], part 6, section 7.4.14, table 4/1d).

Category:	Requirement	Keywords:		ID:	236796
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall verify that the communication path to the external WD does not degrade the quality level below the required quality level.

11.6 S-WdgM Application

Category:	Comment	Keywords:		ID:	230581

This section lists the requirements for the application of the S-WdgM.
For requirements for the S-WdgM Generator see section "S-WdgM Generator" above.

Category:	Comment	Keywords:		ID:	230584

For an overview of the application of the S-WdgM monitoring features see [TT_WDGM_UM].

Category:	Requirement	Keywords:		ID:	230164
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for the correct inclusion of all S-WdgM header files in the AUTOSAR application that declare the S-WdgM API functions.

Category:	Comment	Keywords:		ID:	230586

This includes:

- WdgM_PBCfg.h (generated),
- WdgM_Cfg_Features.h (generated),
- WdgM_Cfg.h, and
- WdgM.h.

Category:	Requirement	Keywords:		ID:	230588
Label:		Safety relevant:			
Related To:		Related To':			

The application shall check the return values (if any) of the S-WdgM API functions to detect errors.

Category:	Comment	Keywords:		ID:	230609

In case a S-WdgM API function call fails, a DET report is made (if configured so) and an error code is returned.

Category:	Requirement	Keywords:		ID:	230597
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for correct handling and escalation of errors that are detected by the S-WdgM code. This includes:

- error codes indicating that a S-WdgM API function was not successful and
- application errors releaved by S-WdgM monitoring features.

Category:	Requirement	Keywords:		ID:	230226
Label:		Safety relevant:			
Related To:	__MKSID__283536,__MKSID__261228	Related To':			

The following memory sections shall not be corrupted or manipulated neither by a HW failure nor by a SW bug in any SW other than S-WdgM:

- S-WdgM local entity data memory and
- S-WdgM global data memory.

Category:	Comment	Keywords:		ID:	289546

This can be achieved by using e.g. ECC and placing the data to a trusted memory area protected by the MPU.

Category:	Comment	Keywords:		ID:	558862

For the memory section description of

- local entity memory section,
- global memory section, and

- global shared memory section

see section "Memory Sections" in [TT_WDGM_UM].

Category:	Requirement	Keywords:		ID:	230607
Label:		Safety relevant:			
Related To:		Related To':			

The following memory sections shall not be corrupted or manipulated neither by a HW failure nor by a SW bug in any SW other than S-WdgM:

- S-WdgM configuration memory and
- S-WdgM program code memory.

Category:	Comment	Keywords:		ID:	558768

This can be achieved by using e.g. ECC, startup and run-time memory checks.

Category:	Comment	Keywords:		ID:	230617

It shall be considered that the S-WdgM code has no mechanism for detecting and/or correcting the following errors:

- corruption of the Local Entity memory,
- corruption of the Global S-WdgM memory,
- corruption of the S-WdgM memory for constants,
- corruption of the S-WdgM code memory, and
- corruption of the used hardware registers.

Note: The S-WdgM itself has no direct access to hardware registers. The registers can be accessed by calls of external functions. These functions are listed in section "Expected Interfaces" above.

Category:	Requirement	Keywords:		ID:	231480
Label:		Safety relevant:			
Related To:	__MKSID__283399,__MKSID__261192	Related To':			

The integrator shall guarantee that address spaces for which the S-WdgM offers no mechanism for error detection and error correction can not be corrupted.

Category:	Comment	Keywords:		ID:	231317

The S-WdgM has mechanisms for detection of unintended manipulations of its own variables placed in the Global Shared memory. If the memory is manipulated, then a reset is performed.

Category:	Comment	Keywords:		ID:	230615

If a mechanism for detection/correction of such manipulations is implemented in the application level or system level, then it should also cover the S-WdgM code.

11.6.1 Application Level API Functions

Category:	Comment	Keywords:		ID:	230729

This section lists the requirements for the S-WdgM API functions on application level.

11.6.1.1 WdgM_GetMode ()

Category:	Requirement	Keywords:		ID:	230813
Label:		Safety relevant:			
Related To:		Related To':			
The application developer shall retrieve the current WD Trigger Mode using WdgM_GetMode () only.					

Category:	Comment	Keywords:		ID:	236520
The WD trigger mode is not fully AUTOSAR 4.0.1 and AUTOSAR 3.1.4 compatible. It considers only the following configuration fields: • WdgMTriggerConditionValue • WdgMTriggerWindowStart • WdgMWatchdogMode					

11.6.1.2 WdgM_SetMode ()

Category:	Requirement	Keywords:		ID:	231776
Label:		Safety relevant:			
Related To:		Related To':			
The application developer shall set the current WD Trigger Mode using WdgM_SetMode () only.					

Category:	Comment	Keywords:		ID:	231778
The WD Trigger Mode is not fully AUTOSAR 4.0.1 and AUTOSAR 3.1.4 compatible. The function WdgM_SetMode () considers only the following configuration fields for a new configuration: • WdgMTriggerConditionValue • WdgMTriggerWindowStart • WdgMWatchdogMode					

Category:	Comment	Keywords:		ID:	283836
Note: The function WdgM_SetMode () can also be used in AUTOSAR 3.1 compatibility mode. See [TT_WDGM_UM].					

Category:	Requirement	Keywords:		ID:	289522
Label:		Safety relevant:			
Related To:	__MKSID__284058	Related To':			
If WdgMDefensiveBehavior is set to "true", then the integrator shall check the DEM reports for the error WDGM_E_IMPROPER_CALLER, which indicates calls of WdgM_SetMode () by unauthorized callers. Otherwise the integrator shall make sure that unauthorized calls of WdgM_SetMode () can not occur.					

11.6.1.3 WdgM_CheckpointReached ()

Category:	Requirement	Keywords:		ID:	230815
Label:		Safety relevant:			
Related To:		Related To':			
The application developer shall indicate to the S-WdgM that a certain point in application code has been reached using WdgM_CheckpointReached () only.					

Category:	Comment	Keywords:		ID:	230817

WdgM_CheckpointReached () performs the following steps:

- all defined Alive Supervision counters are updated,
- Deadline Monitoring is performed, and
- Program Flow Monitoring is performed.

Category:	Comment	Keywords:		ID:	283838

Note: The function WdgM_CheckpointReached () is not defined in AUTOSAR 3.1 compatibility mode and replaced by the function WdgM_UpdateAliveCounter ().

11.6.1.4 WdgM_GetLocalStatus ()

Category:	Requirement	Keywords:		ID:	230733
Label:		Safety relevant:			
Related To:		Related To':			

The application developer shall retrieve the current local monitoring status using WdgM_GetLocalStatus () only.

11.6.1.5 WdgM_GetGlobalStatus ()

Category:	Requirement	Keywords:		ID:	230739
Label:		Safety relevant:			
Related To:		Related To':			

The application developer shall retrieve the current global monitoring status using WdgM_GetGlobalStatus () only.

11.6.1.6 WdgM_PerformReset ()

Category:	Requirement	Keywords:		ID:	230757
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall initiate an immediate Watchdog reset from application level only using WdgM_PerformReset ().

Category:	Comment	Keywords:		ID:	230761

Note: This function is hardware dependent. Some WD drivers do not support an immediate reset. Check the according S-Wdg driver documentation (see also the reference list for example drivers in this document).

11.6.1.7 WdgM_LocalStateChangeCbk, WdgM_GlobalStateChangeCbk

Category:	Comment	Keywords:		ID:	231768

The identifiers WdgM_LocalStateChangeCbk and WdgM_GlobalStateChangeCbk are not function names. They are fields of the S-WdgM Configuration holding pointers to the actual callback functions.

The functions are implemented by the integrator. They are the alternative to RTE notification. RTE notifications are not supported by the S-WdgM.

Category:	Comment	Keywords:		ID:	237639

See [AS_RTE_SWS] for information on AUTOSAR RTE.

Category:	Requirement	Keywords:		ID:	230793
Label:		Safety relevant:			
Related To:		Related To':			

The SW component that implements the callback functions shall be developed with at least the same quality level as required for the system.

Category:	Comment	Keywords:		ID:	230801

Note: The quality level of the S-WdgM is degraded to the quality level of the callback function. An error in the callback function may corrupt the function integrity of the S-WdgM.

Category:	Comment	Keywords:		ID:	231877

If the application that calls the callback function is in a different memory section than the S-WdgM, then the OS feature "Trusted Function" may be necessary to perform the callback.

Category:	Comment	Keywords:		ID:	230891

The function referred to by WdgM_LocalStateChangeCbk is only invoked if WDGM_STATE_CHANGE_NOTIFICATION is set to STD_ON.

Category:	Comment	Keywords:		ID:	239606

The function referred to by WdgM_GlobalStateChangeCbk is only invoked,
if WDGM_STATE_CHANGE_NOTIFICATION is set to STD_ON,
except when the new status is WDGM_GLOBAL_STATUS_STOPPED and WDGM_IMMEDIATE_RESET is set to STD_ON (an immediate system reset need not be notified).

11.6.1.8 WdgM_ActivateSupervisionEntity ()

Category:	Requirement	Keywords:		ID:	231399
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall activate the monitoring of a SE using WdgM_ActivateSupervisionEntity () only.

Category:	Requirement	Keywords:		ID:	231400
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible that the activation of a SE does not

- compromise the systems performance or
- the systems availability (i.e. no unintended resets)

Category:	Comment	Keywords:		ID:	231401
The activation is performed from within WdgM_MainFunction () at the end of a SC.					

Category:	Requirement	Keywords:		ID:	231403
Label:		Safety relevant:			
Related To:		Related To':			
The software component(s) that call WdgM_ActivateSupervisionEntity () shall be developed with at least the same quality level as required by the system safety requirements.					

Category:	Comment	Keywords:		ID:	231404
A missing activation of a SE may violate safety requirements.					

Category:	Comment	Keywords:		ID:	231402
For more information on WdgM_ActivateSupervisionEntity (), see [TT_WDGM_UM].					

Category:	Comment	Keywords:		ID:	231405
WdgM_ActivateSupervisionEntity () is only available if WDGM_ENTITY_DEACTIVATION_ENABLED is set to STD_ON.					

11.6.1.9 WdgM_DeactivateSupervisionEntity ()

Category:	Requirement	Keywords:		ID:	231415
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall deactivate the monitoring of a SE using WdgM_DeactivateSupervisionEntity () only.					

Category:	Requirement	Keywords:		ID:	231416
Label:		Safety relevant:			
Related To:		Related To':			
The integrator is responsible that deactivation of a SE does not compromise system safety requirements.					

Category:	Comment	Keywords:		ID:	231417
The deactivation is performed from within WdgM_MainFunction () at the end of a SC.					

Category:	Requirement	Keywords:		ID:	231419
Label:		Safety relevant:			
Related To:		Related To':			
The software component(s) that call WdgM_DeactivateSupervisionEntity () shall be developed with at least the same quality level as required by the system safety requirements.					

Category:	Comment	Keywords:		ID:	231420
An unintended deactivation of a SE may violate safety requirements.					

Category:	Requirement	Keywords:		ID:	288603
Label:		Safety relevant:			
Related To:	__MKSID__284070	Related To':			
The integrator shall guarantee that a SE is *not* deactivated while its local Initial CP has been hit but one of its local End CP has not yet been hit.					

Category:	Comment	Keywords:		ID:	288605
That is, the program flow of the SE is currently monitored somewhere between the local Initial CP and a local End CP. A deactivation in this moment may corrupt data that is used to monitor the SE.					

Category:	Comment	Keywords:		ID:	231418
For more information on WdgM_DeactivateSupervisionEntity, () see [TT_WDGM_UM].					

Category:	Comment	Keywords:		ID:	231421
WdgM_DeactivateSupervisionEntity () is only available if WDGM_ENTITY_DEACTIVATION_ENABLED is set to STD_ON.					

11.6.1.10 S-WdgM AUTOSAR 3.1 compatibility mode Functions

Category:	Comment	Keywords:		ID:	231387
This section lists safety requirements of functions that are only available in AUTOSAR 3.1 compatibility mode.					

Category:	Comment	Keywords:		ID:	562709
In the "S-WdgM AUTOSAR 3.1 compatibility mode" the S-WdgM emulates the functionality of the AUTOSAR 3.1 Watchdog Manager. This mode is active when the parameter WDGM_AUTOSAR_3_1_X_COMPATIBILITY is set to STD_ON.					

11.6.1.10.1 WdgM_UpdateAliveCounter ()

Category:	Requirement	Keywords:		ID:	283846
Label:		Safety relevant:			
Related To:		Related To':			
The application developer shall indicate to the S-WdgM that a certain point in application code has been reached using WdgM_UpdateAliveCounter () only.					

Category:	Comment	Keywords:		ID:	283852
This function replaces WdgM_CheckpointReached ().					

11.6.1.10.2 WdgM_SetMode ()

Category:	Requirement	Keywords:		ID:	283850
Label:		Safety relevant:			
Related To:		Related To':			
The application developer shall set the current WD Trigger Mode using WdgM_SetMode () only.					

Category:	Requirement	Keywords:		ID:	283856
Label:		Safety relevant:			
Related To:		Related To':			

Note: The AUTOSAR 3.1 version of this function has not parameter CallerID, hence there is no check whether the caller is authorized to call the function or not.

11.6.1.11 Requirements For All Application Level API Functions

Category:	Requirement	Keywords:		ID:	230613
Label:		Safety relevant:			
Related To:	__MKSID__284040,__MKSID__284048,__MKSID__284052	Related To':			

It is the responsibility of the integrator to verify the correctness of parameters passed to S-WdgM application level API functions.

Category:	Requirement	Keywords:		ID:	230735
Label:		Safety relevant:			
Related To:	__MKSID__284040,__MKSID__284048,__MKSID__284052	Related To':			

Some S-WdgM API function have a pointer to data as argument. The integrator is responsible that such data is not modified by the application or code other than the S-WdgM.

Category:	Comment	Keywords:		ID:	230737

This includes:

- WdgM_GetMode (),
- WdgM_GetLocalStatus (), and
- WdgM_GetGlobalStatus ().

Category:	Requirement	Keywords:		ID:	230751
Label:		Safety relevant:			
Related To:	__MKSID__284060,__MKSID__284064,__MKSID__284068,__MKSID__284072,__MKSID__283934	Related To':			

The integrator is responsible for a correct error escalation if a S-WdgM API function returns E_NOT_OK.

Category:	Comment	Keywords:		ID:	230753

For the list of functions that return E_NOT_OK, see comment 229772 in subsection "Return Values" in section "Error Handling" above.

Category:	Requirement	Keywords:		ID:	230222
Label:		Safety relevant:			
Related To:		Related To':			

If the RTE invokes an W-SgdM API function, the RTE code shall not corrupt SWC's memory.

11.6.2 System Level API Functions

Category:	Comment	Keywords:		ID:	230731
This section lists the requirements for the S-WdgM API functions in the system layer.					

Category:	Comment	Keywords:		ID:	230819
Note: The system level API functions are not visible in the application layer. The system functions are invoked by the BSW modules. The RTE does not generate interfaces for these functions.					

11.6.2.1 WdgM_Init ()

Category:	Requirement	Keywords:		ID:	230821
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall initialize (all parts of the) the S-WdgM (data) using WdgM_Init () only.					

Category:	Requirement	Keywords:		ID:	265946
Label:		Safety relevant:			
Related To:	__MKSID__261148,__MKSID__261150	Related To':			
WdgM_Init () shall be called with correct parameter (i.e. the pointer to the according configuration).					

Category:	Comment	Keywords:		ID:	290640
Besides the DET reports, a WdgM_Init() function failure can be checked indirectly by reading the global pointer variable g_wdgm_cfg_ptr. In case of an error the pointer is NULL					

Category:	Requirement	Keywords:		ID:	265886
Label:		Safety relevant:			
Related To:	__MKSID__261062,__MKSID__261130	Related To':			
The integrator shall check the integrity of the S-WdgM Configuration before invoking the WdgM_Init() function.					

Category:	Requirement	Keywords:		ID:	265884
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall check the loaded S-WdgM code for manipulation before invoking the WdgM_Init() function.					

Category:	Comment	Keywords:		ID:	270674
This includes - for example - checks for bitflips.					

Category:	Requirement	Keywords:		ID:	230841
Label:		Safety relevant:			
Related To:		Related To':			

Any S-WdgM monitoring (e.g. any call of WdgM_CheckpointReached ()) shall be performed after the S-WdgM initialization.

Category:	Requirement	Keywords:		ID:	230843
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for passing the appropriate S-WdgM Configuration to WdgM_Init () (i.e. so that no safety requirement is violated).

Category:	Requirement	Keywords:		ID:	231163
Label:		Safety relevant:			
Related To:		Related To':			

The WdgM_Init() function shall be called after the initialization functions of the used S-Wdg drivers (named Wdg_*platform*_Init (), where *platform* is the used platform).

Category:	Comment	Keywords:		ID:	231179

The initialization function(s) of the S-Wdg driver(s) activate the WD device.

Category:	Comment	Keywords:		ID:	264615

Note: Some platforms activate the WD automatically once it is powered.

Category:	Requirement	Keywords:		ID:	231169
Label:		Safety relevant:			
Related To:	__MKSID__261244	Related To':			

The function WdgM_Init() shall be called after the memory protection is activated.

Category:	Requirement	Keywords:		ID:	231171
Label:		Safety relevant:			
Related To:		Related To':			

All other S-WdgM API functions shall only be called after WdgM_Init() has successfully initialized the S-WdgM.

Category:	Requirement	Keywords:		ID:	265944
Label:		Safety relevant:			
Related To:	__MKSID__261279	Related To':			

The function WdgM_Init () shall be called after Wdg_*platform*_Init ().

Category:	Comment	Keywords:		ID:	231181

After execution of WdgM_Init() all monitoring features are fully operational.

Category:	Requirement	Keywords:		ID:	264609
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall be aware that the system's SW is not monitored by the S-WdgM until the S-Wdg device is initialized.

Category:	Requirement	Keywords:		ID:	264611
Label:		Safety relevant:			
Related To:	__MKSID__283878,__MKSID__261176,__MKSID__261170	Related To':			

The integrator is responsible that the initialization of the WD device and the S-WdgM is performed correctly and in time.

Category:	Requirement	Keywords:		ID:	289548
Label:		Safety relevant:			
Related To:	__MKSID__285029	Related To':			

The integrator shall consider:
If WdgM_Init () is called during monitoring by the S-WdgM (i.e. after the initial SC),
then all information about pending violations gets lost.
There will be no further DEM report for pending violations.

Category:	Comment	Keywords:		ID:	289550

In this context, "pending violations" are violations that have already been detected by the S-WdgM but have not yet been escalated to the lower S-WdgM Stack levels and no DEM error has been reported so far.
The time duration of pending depends on the S-WdgM Configuration fields, like the number of tolerated Reference Cycles.

11.6.2.2 WdgM_MainFunction ()

Category:	Requirement	Keywords:		ID:	265950
Label:		Safety relevant:			
Related To:	__MKSID__261170,__MKSID__261172,__MKSID__261174,__MKSID__261176	Related To':			

The function WdgM_MainFunction () shall be called at the end of every SC.

Category:	Requirement	Keywords:		ID:	231209
Label:		Safety relevant:			
Related To:		Related To':			

The integrator shall make sure that WdgM_MainFunction () is correctly scheduled by the operating system (if used) and is always called as scheduled.

Category:	Comment	Keywords:		ID:	231780

If WdgM_MainFunction () is not called in time then the WD is not triggered in time and performs a system reset.

Category:	Requirement	Keywords:		ID:	231183
Label:		Safety relevant:			
Related To:		Related To':			

The first call of WdgM_MainFunction () shall be inside the initial trigger window of the WD.

Category:	Comment	Keywords:		ID:	264607

The time between the WD initialization and its first trigger by function WdgM_MainFunction (SC #0) shall match the system requirements. This time can be configured in the S-Wdg driver configuration (see the User Manual of the according S-Wdg driver. Not all platforms support the configuration of the time for the first S-Wdg trigger.

Category:	Comment	Keywords:		ID:	231185

Otherwise the safe state is initiated.

Category:	Comment	Keywords:		ID:	232459

For details on the initial trigger window see [TT_WDGM_UM].

Category:	Requirement	Keywords:		ID:	231609
Label:		Safety relevant:			
Related To:	__MKSID__283870	Related To':			

The integrator shall guarantee that the WdgM_MainFunction() is not executed faster as defined by the system design.

Category:	Comment	Keywords:		ID:	231191

This can be achieved e.g. by using a windowed watchdog device.
When the WdgM_MainFunction() is executed faster as defined, then the S-WdgM reaction times (reset) are not as expected.
A trigger of the Watchdog outside the defined window leads to a reset. This feature is HW dependent. See the Safety Manual for the WD driver. Safety Manuals for some drivers can be found in section "References" at the end of this document.

Category:	Requirement	Keywords:		ID:	231207
Label:		Safety relevant:			
Related To:		Related To':			

The function WdgM_MainFunction() shall be executed in a task that is different to the tasks that are monitored by the S-WdgM.

Category:	Comment	Keywords:		ID:	231370

Avoid influence or corruption of WdgM_MainFunction() by another task.

11.6.2.3 WdgM_UpdateTickCount ()

Category:	Comment	Keywords:		ID:	231611

This function has been added by TTTech and not part of AUTOSAR.

Category:	Requirement	Keywords:		ID:	230857
Label:		Safety relevant:			
Related To:	__MKSID__284091,_ _MKSID__261214	Related To':			

If the configuration parameter WDGM_TIMEBASE_SOURCE is set to WDGM_EXTERNAL_TICK,
then the Time Base Tick Counter shall be updated using WdgM_UpdateTickCount () every 1/WdgMTicksPerSecond part of a second.

Category:	Requirement	Keywords:		ID:	230873
Label:		Safety relevant:			
Related To:	__MKSID__284091,_ _MKSID__261214	Related To':			

If the configuration parameter WDGM_TIMEBASE_SOURCE is set to WDGM_EXTERNAL_TICK,
then the developer is responsible for calling WdgM_UpdateTickCount () periodically in an interval that is short enough for successful Deadline Monitoring and long enough so that the system safety is not compromised.

Category:	Requirement	Keywords:		ID:	230213
Label:		Safety relevant:			
Related To:		Related To':			

In case an external tick counter is used, the integrator shall avoid

- forward jumps,
- stuck-at,
- negative counting, and
- jitter

of the S-WdgM Timebase Tick counter.

Category:	Comment	Keywords:		ID:	290532

They can influence the expected accuracy of the deadline measurement.

Category:	Comment	Keywords:		ID:	230875

The Timebase Tick counter delivers the time base for Deadline Monitoring. It can be - for example - called from a task with fixed time period and high priority.

Category:	Comment	Keywords:		ID:	230877

If WDGM_TIMEBASE_SOURCE is set to WDGM_INTERNAL_SOFTWARE_TICK,
then WdgM_UpdateTickCount () is called from within WdgM_MainFunction () once at every call of WdgM_MainFunction ().

Category:	Comment	Keywords:		ID:	236538

If WDGM_TIMEBASE_SOURCE is set to WDGM_INTERNAL_HARDWARE_TICK,
then the S-WdgM does not provide the function WdgM_UpdateTickCount (). The counter value is read from the hardware through the S-WdgIf function WdgIf_GetTickCounter (). See [TT_WDGIF_UM] and [TT_WDGIF_SM].
This feature is HW dependent. See the Safety Manual specific for the driver. Safety Manuals for some drivers can be found in section "References" at the end of this document.

11.6.2.4 WdgM_GetVersionInfo ()

Category:	Requirement	Keywords:		ID:	230895
Label:		Safety relevant:			
Related To:		Related To':			
The integrator shall retrieve the current version of the S-WdgM using WdgM_GetVersionInfo () only.					

Category:	Comment	Keywords:		ID:	230897
WdgM_GetVersionInfo () is only available if WDGM_VERSION_INFO_API is set to STD_ON.					

Category:	Comment	Keywords:		ID:	230899
WdgM_GetVersionInfo () is a C macro.					

11.6.2.5 Requirements For All System Level API Functions

Category:	Requirement	Keywords:		ID:	231321
Label:		Safety relevant:			
Related To:		Related To':			
It is the responsibility of the integrator to verify the correctness of parameters that are passed to the S-WdgM system level API functions.					

Category:	Requirement	Keywords:		ID:	230831
Label:		Safety relevant:			
Related To:		Related To':			
Some S-WdgM API functions have a pointer to data as argument. The integrator is responsible that such data is not modified by the system or code other than the S-WdgM.					

Category:	Comment	Keywords:		ID:	230832
This includes: • WdgM_Init () • WdgM_GetVersionInfo () • WdgM_GetLocalStatus() • WdgM_GetGlobalStatus() • WdgM_GetMode()					

Category:	Requirement	Keywords:		ID:	230833
Label:		Safety relevant:			
Related To:		Related To':			
The integrator is responsible for a correct error escalation if a S-WdgM API function returns E_NOT_OK.					

Category:	Comment	Keywords:		ID:	230835
For the list of functions that return E_NOT_OK, see comment 229772 in subsection "Return Values" in section "Error Handling" above.					

Category:	Requirement	Keywords:		ID:	230885
Label:		Safety relevant:			
Related To:		Related To':			

The following functions - although available - are for S-WdgM internal processing and shall not be used:

- GlobalSuspendInterrupts ()
- GlobalRestoreInterrupts ()
- WdgM_SetTickCount ()
- WdgM_WriteRememberedEntityId ()
- WdgM_WriteGlobalActivityFlag ()
- WdgM_WriteGlobalTransitionFlag ()
- WdgM_ReadGlobalTransitionFlag ()
- WdgM_ReadRememberedEntityId ()

11.6.3 Memory Access

Category:	Comment	Keywords:		ID:	231145

This section lists the requirements related to memory access of the various S-WdgM API functions.

Category:	Requirement	Keywords:		ID:	231203
Label:		Safety relevant:			
Related To:	__MKSID__261230	Related To':			

The S-WdgM API functions shall be granted the required access rights to the various memory sections as depicted in the following table.

Category:	Comment	Keywords:		ID:	231147

The following table shows the required access rights for each S-WdgM API function according to the memory sections.
A description of the memory sections can be found in [TT_WDGM_UM].

	Memory Section			
Function	**S-WdgM local SE memory**	**S-WdgM global memory**	**S-WdgM global shared memory**	**MCU Register (3)**
WdgM_Init () (1)	read, write	read, write	read, write	read, write
WdgM_MainFunction ()	read	read, write	read	read, write
WdgM_CheckpointReached ()	read, write	read	read, write	-----
WdgM_UpdateTickCount () (2)	-----	read, write	-----	-----
WdgM_PerformReset ()	-----	write	-----	read, write
WdgM_GetLocalStatus ()	read	-----	-----	-----
WdgM_GetGlobalStatus ()	-----	read	-----	-----
WdgM_GetMode ()	-----	read	-----	-----
WdgM_SetMode ()	-----	write	-----	-----
WdgM_DeactivateSupervisionEntity ()	-----	-----	write	-----
WdgM_ActivateSupervisionEntity ()	-----	-----	write	-----

table 19

(1) The function WdgM_Init () initializes all internal S-WdgM variables and the S-WdgM variables in the contexts of the SEs.

(2) The Timebase Tick counter belongs to the S-WdgM global variables.
(3) MCU Register access. The S-WdgM does not access the hardware registers directly. The hardware is accessed by calling the WD driver or MCU driver functions. The register access is platform and implementation dependent and may imply "supervisor MCU mode" or "privileged MCU mode". See the driver's User Manual and Safety Manual for details.

Category:	Comment	Keywords:		ID:	231149

Note: The MMU or MPU - if running on the target system - need to be configured accordingly.

Category:	Requirement	Keywords:		ID:	284909
Label:		Safety relevant:			
Related To:	__MKSID__261230	Related To':			

The integrator shall check the MMU/MPU error messages if MMU or MPU is used.

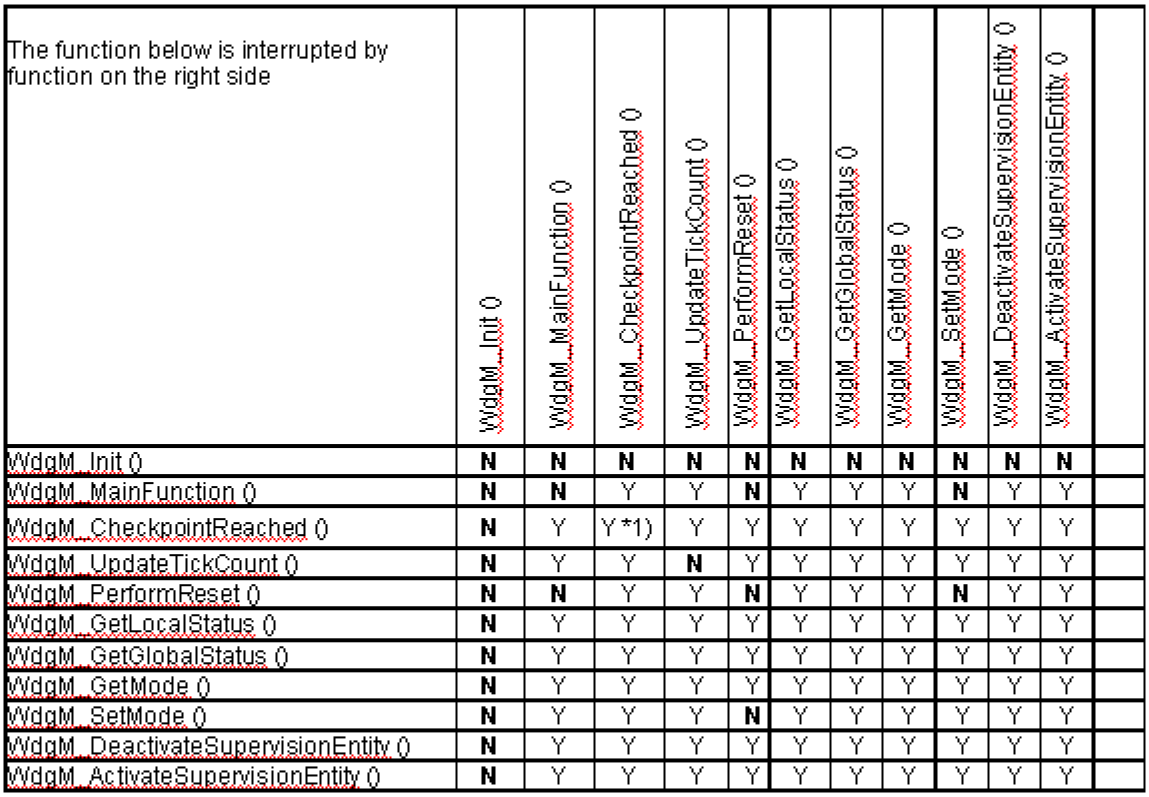
Category:	Comment	Keywords:		ID:	284911

For the case that a S-WdgM API function is denied required memory access.

11.6.4 Concurrent Function Calls

Category:	Requirement	Keywords:		ID:	283147
Label:		Safety relevant:			
Related To:	__MKSID__284600,_ _MKSID__284608	Related To':			

The following table shows which functions may run concurrently:



The function below is interrupted by function on the right side	WdgM_Init ()	WdgM_MainFunction ()	WdgM_CheckpointReached ()	WdgM_UpdateTickCount ()	WdgM_PerformReset ()	WdgM_GetLocalStatus ()	WdgM_GetGlobalStatus ()	WdgM_GetMode ()	WdgM_SetMode ()	WdgM_DeactivateSupervisionEntity ()	WdgM_ActivateSupervisionEntity ()	
WdgM_Init ()	N	N	N	N	N	N	N	N	N	N	N	
WdgM_MainFunction ()	N	N	Y	Y	N	Y	Y	Y	N	Y	Y	
WdgM_CheckpointReached ()	N	Y	Y *1)	Y	Y	Y	Y	Y	Y	Y	Y	
WdgM_UpdateTickCount ()	N	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	
WdgM_PerformReset ()	N	N	Y	Y	N	Y	Y	Y	N	Y	Y	
WdgM_GetLocalStatus ()	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
WdgM_GetGlobalStatus ()	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
WdgM_GetMode ()	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
WdgM_SetMode ()	N	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	
WdgM_DeactivateSupervisionEntity ()	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
WdgM_ActivateSupervisionEntity ()	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	

figure 1

"Y" is for "Yes" (may run concurrently) and
"N" is for "No" (may not run concurrently)
*1) Allowed only if running in different application contexts.

12 Safety Lifecycle Tailoring

Category:	Comment	Keywords:		ID:	230008

This section describes which phases of the S-WdgM product safety lifecycle according to [ISO26262] were executed by TTTech during the development and which phases have to be executed by the integrator.

Category:	Comment	Keywords:		ID:	230016

The S-WdgM is a software unit representing a safety element out of context (SEooC) according to [ISO26262], part 10. The SW requirements of the S-WdgM are based on [AS_WDGM_SWS] and [TT_WDGM_SRD] with deviations listed in [TT_WDGM_UM]. The architectural design is documented in [TT_WDGM_UDD].

Category:	Comment	Keywords:		ID:	230020

The following ISO 26262 phases that are relevant for the integrator were executed by TTTech:

- 3-7 Hazard analysis and risk assessment *)
- 3-8 Functional Safety Concept *)
- 4-6 Technical Safety Concept *)
- 4-7 System Design *)
- 6-5 Initiation of product development at SW level *),
- 6-8 Software unit design and implementation *) and
- 6-9 Software unit tests *).

*) As far as related to the S-WdgM as SEooC.

Category:	Requirement	Keywords:		ID:	230022
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for the execution of ISO 26262 phase 6-6 (Specification of SW safety requirements) to identify the system's SW safety requirements.

Category:	Requirement	Keywords:		ID:	230024
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for the execution of ISO 26262 phase 6-7 (SW architectural design) that covers S-WdgM code.

Category:	Comment	Keywords:		ID:	230026

The S-WdgM code does not impose any special restrictions on the SW architecture design except for the requirements in this document.

Category:	Requirement	Keywords:		ID:	230030
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for the execution of ISO 26262, part 6, clause 8.4.5, b) to verify that the software unit design of the S-WdgM is complete with respect to the software safety requirements and the software architecture through traceability.

Category:	Requirement	Keywords:		ID:	230040
Label:		Safety relevant:			
Related To:		Related To':			
The integrator is responsible for the execution of ISO 26262 phase 6-10 (SW integration and testing) to verify that S-WdgM code is correctly integrated into the system.					

Category:	Requirement	Keywords:		ID:	230042
Label:		Safety relevant:			
Related To:		Related To':			
The integrator is responsible for the execution of phase ISO 26262 6-11 (Verification of SW safety requirements) to verify the safety requirements that are related to S-WdgM code.					

13 Qualification

Category:	Comment	Keywords:		ID:	230060

The S-WdgM has been developed according to the requirements in [ISO26262] as specified in section "Safety Lifecycle Tailoring" above. It can be integrated in systems up to ASIL D, provided that all requirements in this document are fulfilled.

Category:	Comment	Keywords:		ID:	228543

The hardware dependent qualification data and required resources for each platform are part of the WD drivers' Safety Manual.

Category:	Comment	Keywords:		ID:	230093

The S-WdgM Stack Safety Case [TT_WDGS_SC] lists all S-WdgM qualification documents.

Category:	Comment	Keywords:		ID:	230128

om The S-WdgM unit tests are specified in [TT_WDGM_UTS].
The S-WdgM tests of the unit test framework are specified in [TT_WDGS_UTS].
The integration tests of the S-Wdg Stack are specified in [TT_WDGS_ITS].

Category:	Comment	Keywords:		ID:	260892

The environments and S-WdgM Configurations of integration tests that have been conducted by TTTech can be found in the Safety Manual of the various S-Wdg drivers (e.g. [TT_WDGDR_*platform*_SM], where *platform* is the used platform. See also section "References" at the end of the document).

Category:	Requirement	Keywords:		ID:	230124
Label:		Safety relevant:			
Related To:		Related To':			

The integrator is responsible for the qualification of the S-WdgM code for the used environment. This means that the S-WdgM code must be integration tested against these environment.

The environment comprises:

- the target CPU,
- the compiler and linker,
- the compiler and linker settings,
- S-WdgM Stack pre-compile configurations,
- the used WDs and S-Wdg drivers, and
- the AUTOSAR software stack.

Category:	Requirement	Keywords:		ID:	283952
Label:		Safety relevant:			
Related To:		Related To':			

Integration tests shall also cover the detection and escalation of all kinds of violations (by means of "negative tests").

This comprises:

- deadline violations (Local and Global Transitions, min.deadline violations, max. deadline violations),

- program flow violations (Local and Global Transitions), and
- Alive Counter violations (min. Alive Counter violation, max. Alive Counter violations).

Category:	Requirement	Keywords:		ID:	230126
Label:		Safety relevant:			
Related To:		Related To':			

If the S-WdgM is used in an environment that differs in any way from the environment it has been tested with (see the list below), then the integrator shall analyze the consequences of the differences and conduct corresponding tests (see [ISO26262] part 6, clause 9, in particular [ISO26262] part 6, clause 9.4.6).

The TTTech test environments are defined in

- the S-Wdg driver Safety Manual [TT_WDGDR_*platform*_SM] (if a TTTech driver for this *platform* exists),

(and in detail in:)

- the Integration Test Specification [TT_WDGS_ITS], and
- the Unit Test Specification [TT_WDGM_UTS].

Category:	Comment	Keywords:		ID:	231613

TTTech offers qualification of the S-WdgM for customer-specific configurations.

14 Resource Requirements

Category:	Comment	Keywords:		ID:	230150
The memory consumption and runtime consumption of the S-WdgM depends on the chosen HW, which itself is chosen by the used S-Wdg driver. The resource requirements of the complete S-WdgM Stack can be found in the according S-Wdg Safety Manual.					

15 Constraints And Known Problems

Category:	Comment	Keywords:		ID:	290553
For known problem see the Release Notes delivered with this software module.					

16 References

<table>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>229559</td></tr>
<tr><td colspan="6">[ISO26262] ISO26262, Internation Standard, Road vehicles- Functional safety, First edition 2011-11-15</td></tr>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>229814</td></tr>
<tr><td colspan="6">[TT_WDGIF_SM] TTTech Automotive GmbH, Safe Watchdog Interface - Safety Manual, D-SAFEX-S-70-003</td></tr>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>229604</td></tr>
<tr><td colspan="6">[TT_WDGDR_MPC56xx_SM] TTTech Automotive GmbH, Safe Watchdog Driver for MPC56xx - Safety Manual, D-MSP-M-70-022</td></tr>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>229606</td></tr>
<tr><td colspan="6">[TT_WDGDR_SAFETCORE_SM] TTTech Automotive GmbH, Safe Watchdog Driver for TriCore and SafeTcore - Safety Manual, D-SAFEX-S-70-013</td></tr>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>229612</td></tr>
<tr><td colspan="6">[TT_WDGDR_TMS570LS3x_SM] TTTech Automotive GmbH, Safe Watchdog Driver for TMS570LS3x - Safety Manual, D-SAFEX-S-70-015</td></tr>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>230103</td></tr>
<tr><td colspan="6">[TT_WDGS_SC] TTTech Automotive GmbH, Safe Watchdog Manager Stack - Safety Case, D-SAFEX-IN-70-001</td></tr>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>229551</td></tr>
<tr><td colspan="6">[TT_WDGM_UM] TTTech Automotive GmbH, Safe Watchdog Manager - User Manual, D-MSP-M-70-001</td></tr>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>229626</td></tr>
<tr><td colspan="6">[TT_WDGIF_UM] TTTech Automotive GmbH, Safe Watchdog Interface - User Manual, D-MSP-M-70-006</td></tr>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>229628</td></tr>
<tr><td colspan="6">[TT_WDGDR_MPC56xx_UM] TTTech Automotive Gmbh, Safe Watchdog Driver (MPC56xx) - User Manual, D-MSP-M-70-008</td></tr>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>229630</td></tr>
<tr><td colspan="6">[TT_WDGDR_SAFETCORE_UM] Safe Watchdog Driver (SafeTcore) - User Manual, D-MSP-M-70-007</td></tr>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>229634</td></tr>
<tr><td colspan="6">[TT_WDGDR_TMS570LS3x_UM] TTTech Automotive GbmH, Safe Watchdog Driver (TMS570LS3x) - User Manual, D-MSP-M-70-010</td></tr>
<tr><td>Category:</td><td>Comment</td><td>Keywords:</td><td></td><td>ID:</td><td>229521</td></tr>
<tr><td colspan="6">[AS_WDGM_SWS] AUTOSAR, Specification of Watchdog Manager, Version 2.0.0, Release 4.0, Revision 1</td></tr>
</table>

Category:	Comment	Keywords:		ID:	555639
[AS_WDGM_SWS_3_1] AUTOSAR, Specification of Watchdog Manager, Version 1.2.2, Release 3.1, Revision 1					

Category:	Comment	Keywords:		ID:	229535
[AS_WDGIF_SWS] AUTOSAR, Specification of Watchdog Interface, Version 2.3.0, Release 4.0, Revision 1					

Category:	Comment	Keywords:		ID:	229537
[AS_WDGDR_SWS] AUTOSAR, Specification of Watchdog Driver, Version 2.3.0, Release 4.0, Revision 1					

Category:	Comment	Keywords:		ID:	237643
[AS_RTE_SWS] AUTOSAR, Specification of RTE, Version 3.0.0, Release 4.0, Revision 1					

Category:	Comment	Keywords:		ID:	230108
[AS_STDTYP_SWS] AUTOSAR, Specification of Standard Types, Version 1.3.0, Release 4.0, Revision 1					

Category:	Comment	Keywords:		ID:	230110
[AS_COMABS_SWS] AUTOSAR, Specification of Compiler Abstraction, Version 3.0.0, Release 4.0, Revision 1					

Category:	Comment	Keywords:		ID:	230112
[AS_PLTFM_SWS] AUTOSAR, Specification of Platform Types, Version 2.3.0, Release 4.0, Revision 1					

Category:	Comment	Keywords:		ID:	230114
[AS_MEM_SWS] AUTOSAR, Specification of Memory Mapping, Version 1.2.0, Release 4.0, Revision 1					

Category:	Comment	Keywords:		ID:	229557
[TI_SPNU511_UM] Texas Instruments, Safety Manual for TMS570LS31x/21x and RM48x Hercules™ ARM® Safety Critical Microcontrollers - User's Guide, Literature Number: SPNU511A, February 2012					

16.1 Internal Documents

Category:	Comment	Keywords:		ID:	283456
The following referenced documents are internal TTTech Automotive GmbH document. For inspection, please contact TTTech Automotive GmbH:					

Category:	Comment	Keywords:		ID:	283458
[TT_WDGM_ETA] TTTech Automotive GmbH, Safe Watchdog Manager - Event Tree Analysis, S-SAFEX-S-70-001					

Category:	Comment	Keywords:		ID:	283460
[TT_WDGM_SD] TTTech Automotive GmbH, Safe Watchdog Manager - System Design, D-SAFEX-D-70-007					

Category:	Comment	Keywords:		ID:	283476
[TT_WDGM_TSR] TTTech Automotive GmbH, Safe Watchdog Manager - Technical Safety Requirements, D-SAFEX-S-70-021					

Category:	Comment	Keywords:		ID:	283462
[TT_WDGM_SRD] TTTech Automotive GmbH, Safe Watchdog Manager - Software Requirements Document, D-SAFEX-S-70-004					

Category:	Comment	Keywords:		ID:	283464
[TT_WDGM_UDD] TTTech Automotive GmbH, Safe Watchdog Manager - Unit Design Document, D-SAFEX-D-70-002					

Category:	Comment	Keywords:		ID:	283468
[TT_WDGM_UTS] TTTech Automotive Gmbh, Safe Watchdog Manager - Unit Test Specification, D-SAFEX-V-70-001					

Category:	Comment	Keywords:		ID:	283472
[TT_WDGS_ITS] TTTech Automotive GmbH, Safe Watchdog Manager Stack - Integration Test Specification, D-SAFEX-V-01-001					

Category:	Comment	Keywords:		ID:	283474
[TT_WDGS_ITR] TTTech Automotive GmbH, Safe Watchdog Manager Stack - Integration Test Report, D-SAFEX-V-01-002					