

SCOPE OF APPLICATION All Project/Engineering	HYUNDAI AutoEver	SHT/SHTS 1 / 70
Responsibility: Classic AUTOSAR team	AUTOSAR WdgM User Manual	DOC. NO
<h1>AUTOSAR WdgM User Manual</h1>		

Document Change History				
Date (YYYY-MM-DD)	Ver.	Editor	Chap	Content (before revision-> after revision)
2020-09-23	1.0.0	ThanhNH5	All	WdgM Initial Release
2020-11-25	1.0.1	ThanhNH5	3.3 4.3 5.2.9 7.1.1 9.1.2.1	Added new Chapter 3.3 4.3.2: Added change log in file UM version 1.0.1 5.2.9: Updated note 2 7.1.1: Added ERR013050 to ERR013054 9.1.2.1: Added exclusive area FIRSTEXPIREDEID_MULTICORE_PROTECTION
2021-03-25	1.0.2	HiepVT1	4.3	4.3.3: Added change log in file UM version 1.0.2
2021-06-30	1.0.3	HiepVT1	4.3	4.3.4: Added change log in file UM version 1.0.3
2021-09-17	1.1.0	HiepVT1	4.3	4.3.4: Added change log in file UM version 1.1.0 Change logo and company name from Hyundai Autron to Hyundai AutoEver
2021-10-25	1.1.1	HiepVT1	4.3	4.3.4: Added change log in file UM version 1.1.1
2021-12-06	1.2.0	HiepVT1	4.3	4.3.4: Added change log in file UM version 1.2.0
2022-01-25	1.3.0	HiepVT1	4.3	4.3.4: Added change log in file UM version 1.3.0
2022-06-29	1.3.0.1	TriBD Gongbin Lim	1 4.3	Clarify copyright of the code Added change log in file UM version 1.3.0.1
2022-08-12	1.3.1.0	TriBD Gongbin Lim	4.3	Added change log in file UM version 1.3.1.0
2022-08-22	1.3.1.1	Gongbin Lim	4.3	Added change log in file UM version 1.3.1.1
2022-12-29	1.3.2.0	NhanNV8	4.3	Added change log in file UM version 1.3.2.0
2023-10-11	1.3.2.1	Hyeonseok Park	4.3	Added change log in file UM version 1.3.2.1
2023-10-25	1.3.3.0	Jeonghyun Kim	4.3	Added change log in file UM version 1.3.3.0

Edition Date: 2023/10/25	File Name WdgM_UM.pdf	Creation Jeonghyun Kim	Check Seungmo Koo	Approval Junho Cho
Document Management System				
		2023/10/25	2023/10/25	2023/10/25

Table of Contents

1. OVERVIEW	5
2. REFERENCE	5
3. AUTOSAR SYSTEM	6
3.1 Overview of Software Layers	6
3.2 AUTOSAR WdgM Module	7
3.3 WdgM Multicore	9
4. PRODUCT RELEASE NOTES	11
4.1 Overview.....	11
4.2 Scope of the release	11
4.3 Change Log.....	11
4.3.1 Version 1.0.0.0	11
4.3.2 Version 1.0.1.0	11
4.3.3 Version 1.0.2.0	11
4.3.4 Version 1.0.3.0	11
4.3.5 Version 1.1.0.0	12
4.3.6 Version 1.1.1.0	12
4.3.7 Version 1.2.0.0	12
4.3.8 Version 1.3.0.0	12
4.3.9 Version 1.3.0.1	12
4.3.10 Version 1.3.1.0	12
4.3.11 Version 1.3.1.1	12
4.3.12 Version 1.3.2.0	12
4.3.13 Version 1.3.2.1	13
4.3.14 Version 1.3.3.0	13
4.4 Limitations	13
4.5 Deviations	13
5. CONFIGURATION GUIDE	15
5.1 WdgMGeneral Container	15
5.1.1 WdgMWatchdog setting	16
5.1.2 WdgMSupervisedEntity setting.....	16
5.1.3 WdgMCheckpoint setting	17
5.1.4 WdgMInternalTransition setting.....	17
5.2 WdgMConfigSet Container.....	18
5.2.1 WdgMDemEventParameterRefs setting.....	18
5.2.2 WdgMMode setting	18

5.2.3 WdgMAliveSupervision setting	19
5.2.4 WdgMDeadlineSupervision setting	19
5.2.5 WdgMAliveSupervision setting	20
5.2.6 WdgMDeadlineSupervision setting	20
5.2.7 WdgMExternalLogicalSupervision setting	20
5.2.8 WdgMExternalTransition setting	21
5.2.9 WdgMLocalStatusParams setting	21
5.2.10 WdgMTrigger setting	22
5.3 System Configuration	23
5.3.1 ApplicationSwComponentType setting	23
5.3.2 Assembly Sw Connector Settings	23
6. APPLICATION PROGRAMMING INTERFACE (API)	24
6.1 Type Definitions	24
6.1.1 WdgM_ModeType	24
6.1.2 WdgM_SupervisedEntityIdType	24
6.1.3 WdgM_CheckpointIdType	24
6.1.4 WdgM_LocalStatusType	24
6.1.5 WdgM_GlobalStatusType	24
6.2 Macro Constants	26
6.3 Functions	27
6.3.1 WdgM_Init	27
6.3.2 WdgM_Delnit	28
6.3.3 WdgM_GetVersionInfo	29
6.3.4 WdgM_SetMode	30
6.3.5 WdgM_GetMode	31
6.3.6 WdgM_CheckpointReached	32
6.3.7 WdgM_GetLocalStatus	33
6.3.8 WdgM_GetGlobalStatus	34
6.3.9 WdgM_PerformReset	35
6.3.10 WdgM_GetFirstExpiredSEID	36
7. GENERATOR	37
7.1 Generator Message	37
7.1.1 Error Messages	37
7.1.2 Warning Messages	49
7.1.3 Information Messages	49
8. SWP ERROR CODE	50
8.1 SWP Error Code List	50
8.1.1 WDGME_SUPERVISION	50
8.1.2 WDGME_SET_MODE	51
9. APPENDIX	52

9.1	Integration Additional Module.....	52
9.1.1	Os module.....	52
9.1.2	Rte module.....	53
9.2	Application User Setting Guide	56
9.2.1	Alive Supervision.....	56
9.2.2	Deadline Supervision.....	61
9.2.3	Logical Supervision	62
9.3	Notes on design.....	67
9.3.1	Alive Supervision.....	67
9.3.2	Deadline Supervision.....	67
9.3.3	Logical Supervision	67
9.3.4	Mode Switch Event	67
9.3.5	WdgM_Delnit	67
9.3.6	Std_ReturnType.....	67
9.3.7	Low Power Mode	67
9.3.8	Condition Value	68
9.4	Add workaround for Initialization function in multicore	69

1. Overview

It is written based on Autosar standard SRS / SWS. To have more detailed functional description when using the module, refer to the reference document below

The interpretation of the category related to setting is as follows

- Changeable (C) : Items that can be set by the user
- Fixed (F) : Items that cannot be changed by user
- Not Supported (N) : Items that not supported

This source code is permitted to be used only in projects contracted with Hyundai Autoever, and any other use is prohibited.

If you use it for other purposes or change the source code, you may take legal responsibility.

In this case, There is no warranty and technical support.

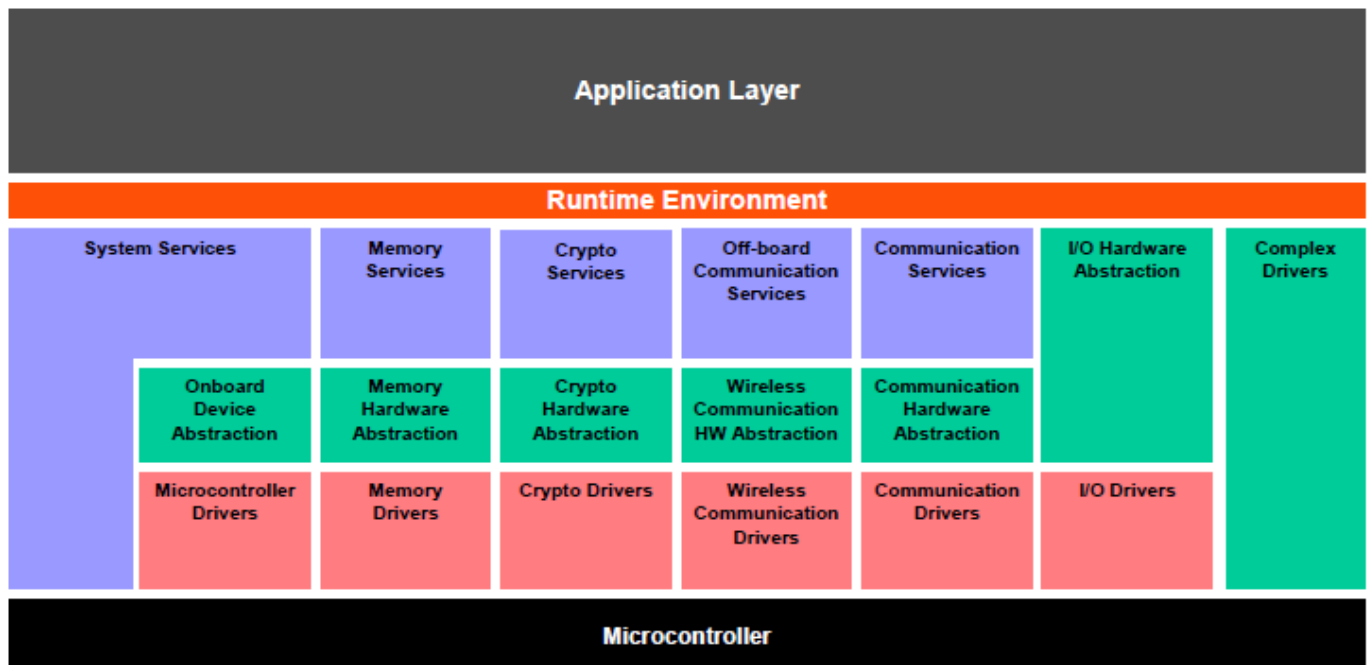
2. Reference

Sl. No.	Title	Version
1.	AUTOSAR_SWS_WatchdogManager.pdf	4.4.0

3. AUTOSAR System

3.1 Overview of Software Layers

The Layered Architecture of the AUTOSAR platform is as follows. The AUTOSAR platform can be divided into Service Layer, ECU Abstraction Layer, Complex Device Drivers and Microcontroller Abstraction Layer.



3.2 AUTOSAR WdgM Module

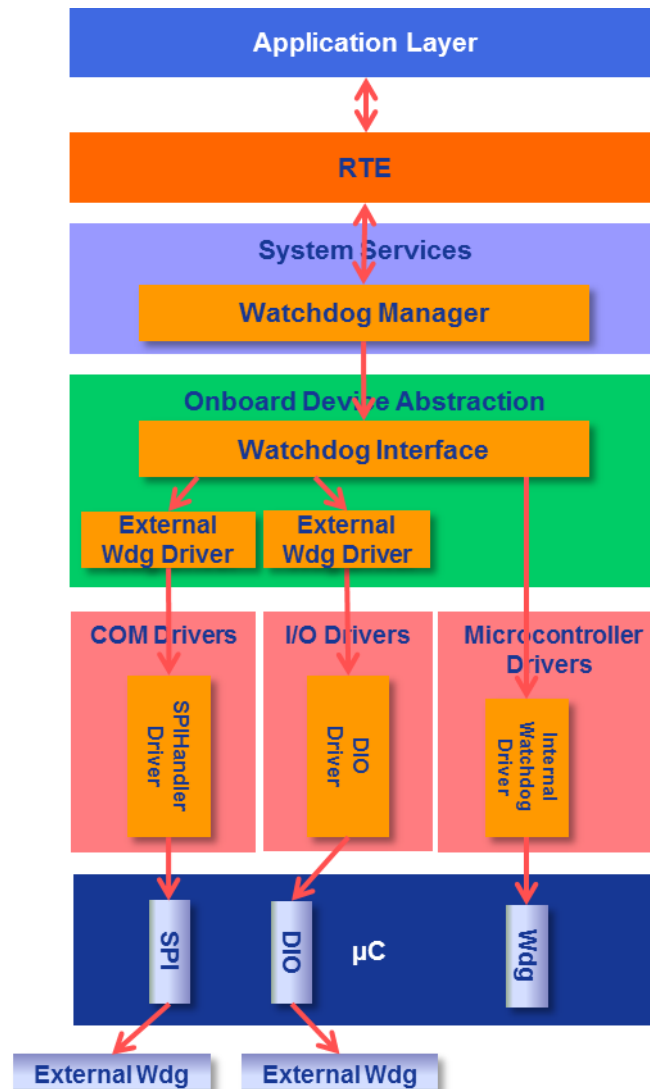
The interface between AUTOSAR layer and each module to use Wdg Stack is as follows.

The modules in the Wdg Stack are Watchdog Manager (WdgM), Watchdog Interface (WdgIf), and Watchdog Driver (Wdg).

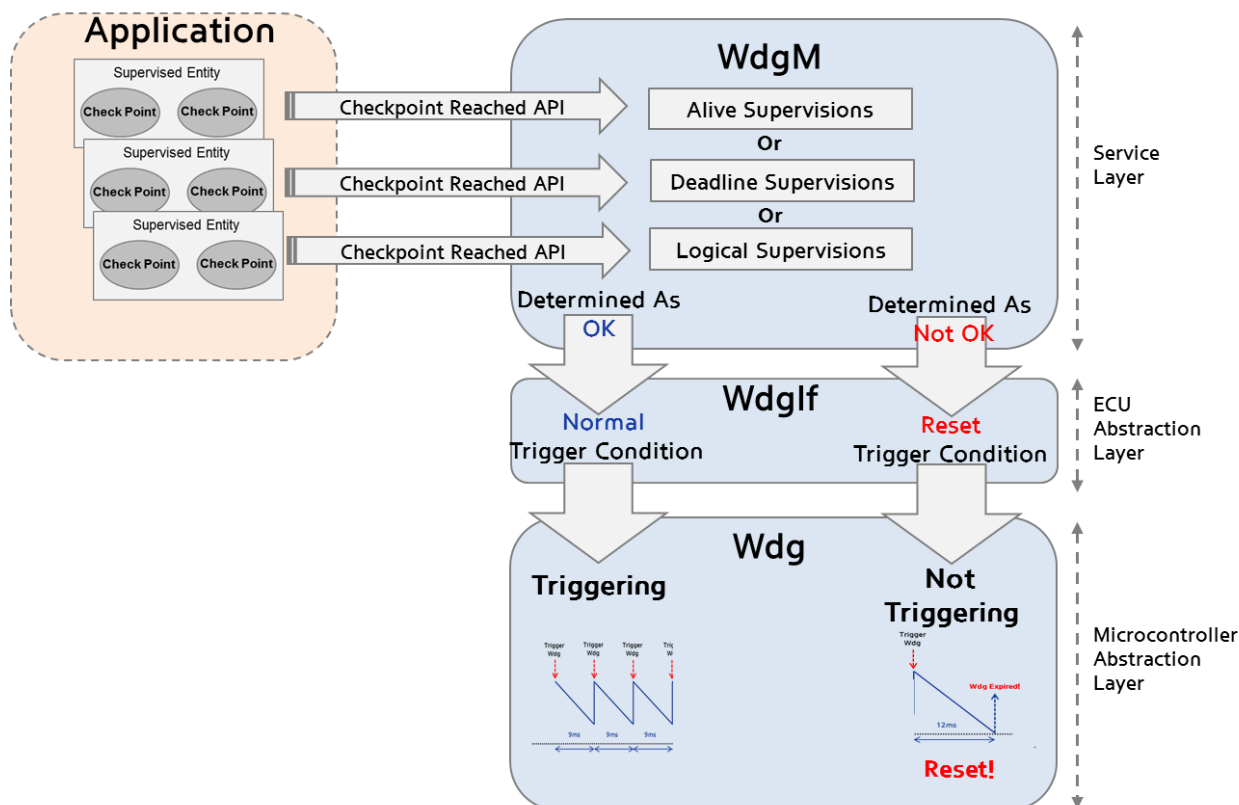
WdgM: Monitoring target monitoring, providing Wdg triggering condition, request to change Wdg mode, error handling

WdgIf : Wdg Interface abstraction

Wdg: HW Wdg trigger, HW Wdg mode control



WdgM only monitors whether the targets are operating according to the monitoring conditions, WdgIf abstracts Wdg, and Wdg performs the actual HW Watchdog triggering.



WdgM monitors whether supervised entities are operating according to the monitoring conditions (Alive Supervision, Deadline Supervision, Logical Supervision) and provides triggering conditions to Wdg through WdgIf based on the monitoring results. In other words, if the monitoring targets are operating normally according to the monitoring conditions, WdgM notifies Wdg of the normal triggering condition through WdgIf, and if it is determined that the monitoring targets need to reset Wdg because of violation of the monitoring conditions, WdgM sends Wdg through WdgIf. By setting the triggering condition to 0, Wdg is ultimately expired and reset.

There are 3 supervision mechanisms in WdgM which are described as below:

Alive Supervision

Alive Supervision defines one checkpoint in the target to be monitored (Supervised Entity) and monitors the monitored target based on the periodic execution time of the Supervision Cycle (WdgM_MainFunction cycle).

Deadline Supervision

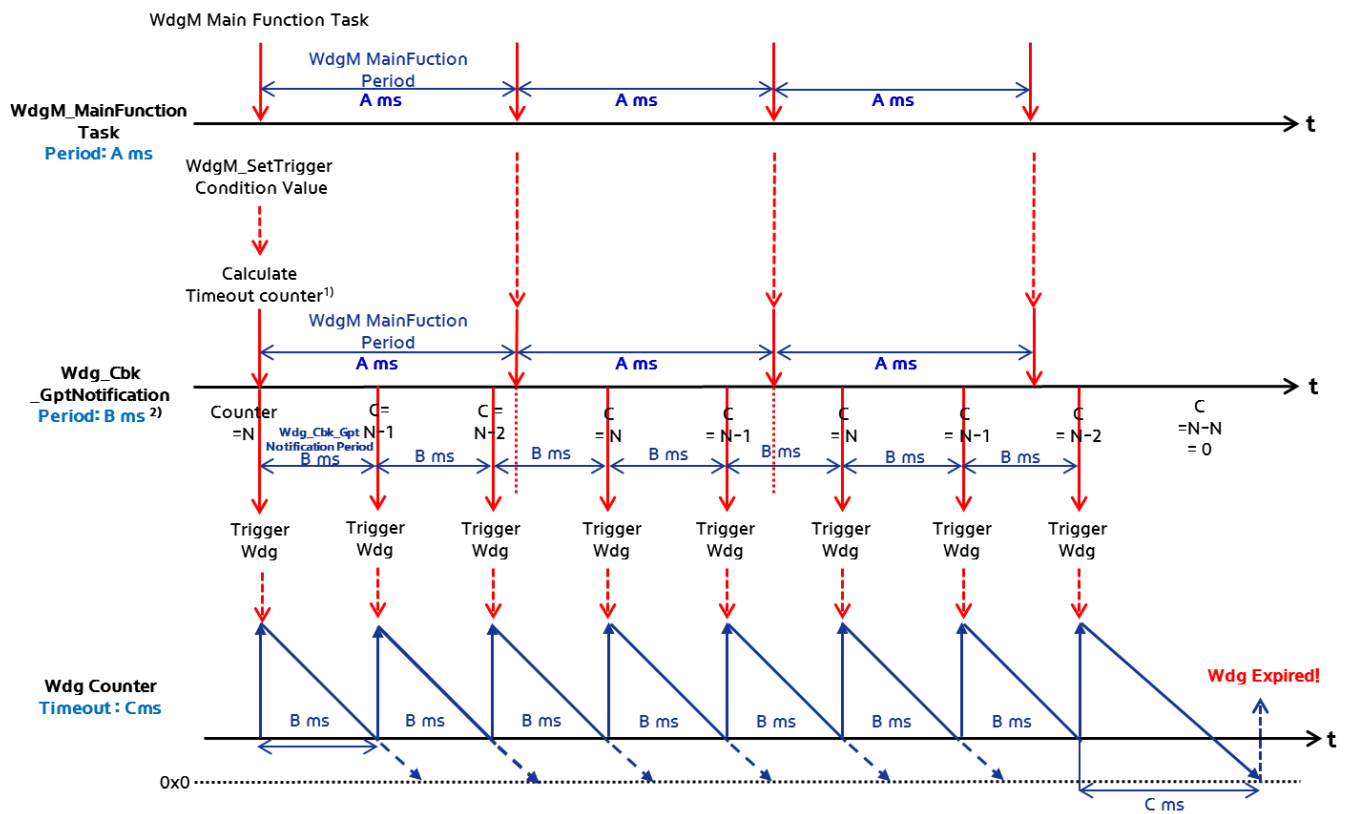
Deadline Supervision defines two checkpoints to be monitored and monitors execution time within the range of execution time between two points (minimum execution time \leq actual execution time \leq maximum execution time).

Logical Supervision

Logical Supervision defines the sequence between the monitoring points of the supervised entity to be monitored, and monitors the execution sequence.

According to Wdg design, Wdg operation is classified by MCU.

Trigger of Wdg module according to WdgM's MainFunction operates as shown below.

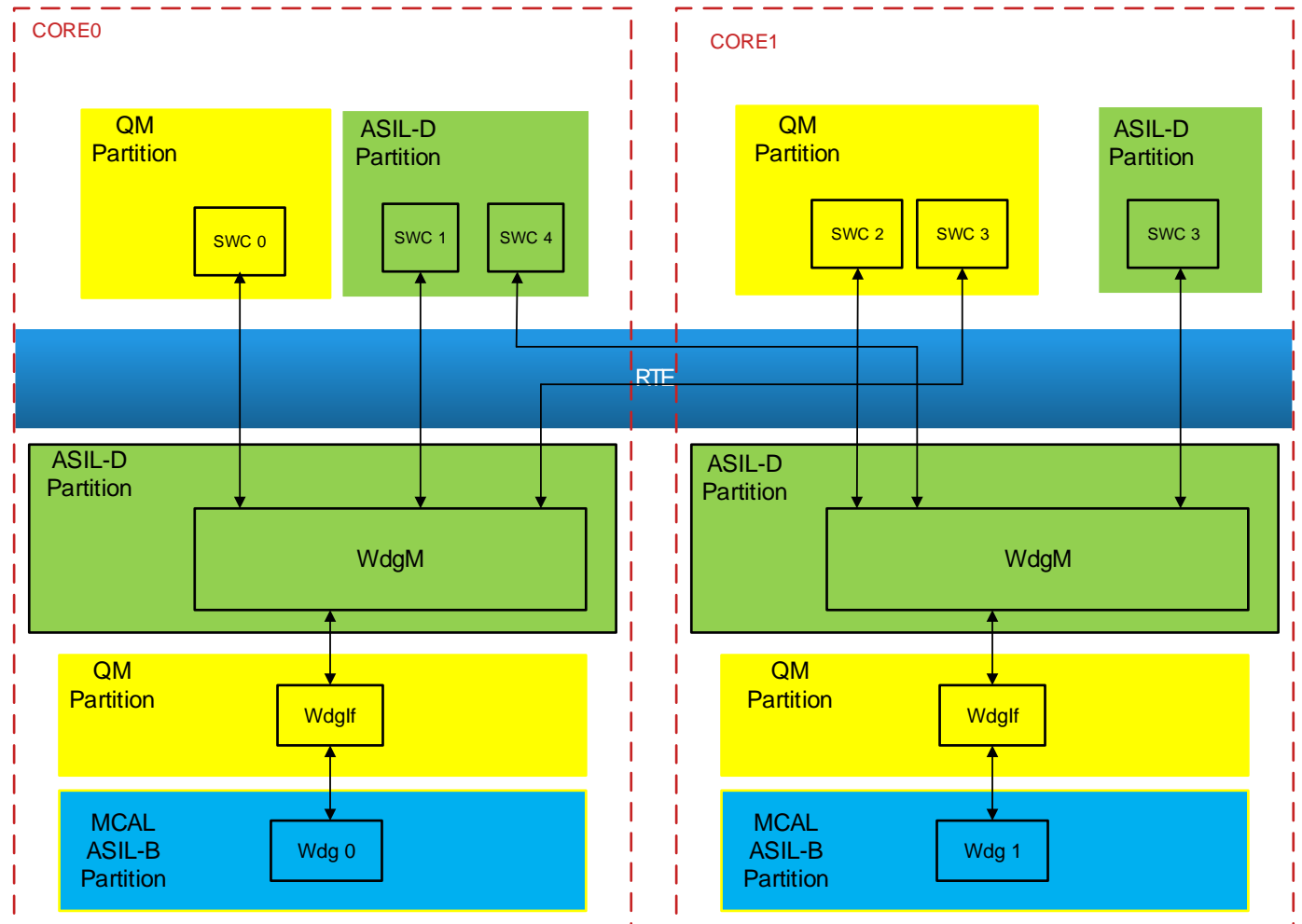


1) Recalculate timeout counter = $\text{Int}(\text{WdgM SetTrigger Condition Value} / \text{Wdg_CbK_GptNotification Period})$, Ex) $30\text{ ms} / 9\text{ ms} = 3.33 \approx 3$

2) $\text{Wdg_CbK_GptNotification Period} = \text{Timeout Period} * (3 / 4)$

3.3 WdgM Multicore

- WdgM module supports to run in multi cores (WdgM's Main functions will be executed in different cores).
- WdgM module supervises for Supervised Entities that run in multi cores. The Supervised Entities can run in the same core or difference core with the WdgM.
- The number of core which the WdgM module executed is configurable.



4. Product Release Notes

4.1 Overview

This chapter aims to provide release-related content for the Hyundai AutoEver WdgM Module, and describes restrictions and specifics for the WdgM Module Software product release version

4.2 Scope of the release

All contents of this document are limited to the following Hyundai AutoEver WdgM modules.

Module	Autosar version	Module version
WdgM	4.4.0	1.3.3

※ Module version means the Sw version of each module's BswModule Description (Bswmd) file.

4.3 Change Log

4.3.1 Version 1.0.0.0

- Version 1.0.0
 - Initial version

4.3.2 Version 1.0.1.0

- Version 1.0.1
 - Updated T-Code Framework: Change from FW_0.0.9 to FW_0.0.15.
 - Updated MemMap section macro as defined by Autosar for ASIL D module.
 - Change the order of Supervised Entity is not sorted to sort Supervised Entity by Supervised Entity ID in ascending order.
 - WdgM_FailedRefCycleCounter is accessed through SE index instead of counter index.
 - Updated T-Code to fix error occurred when compile source code if Alive supervision not configured in arxml file.
 - Corrected number of structure element in WdgM_DelInitTimeout, and WdgM_Config array.
 - Removed redundant code in WdgM_EvaluatePlatformGlobalStatus(): Only master core can calculate new WdgM_PlatformGlobalSupervisionStatus.

4.3.3 Version 1.0.2.0

- Version 1.0.2.0
 - Updated code: Changed code to use Det_ReportError() replace for Det_ReportRuntimeError() dependent on the DET version.

4.3.4 Version 1.0.3.0

➤ Version 1.0.3.0

- Remove code check and report to DET when WdgM in master core need to de-initialized after slave cores in WdgM_DeInit function.

4.3.5 Version 1.1.0.0

➤ Version 1.1.0.0

- Update SQT, SIT, SUT to verify WdgM in TC36x environment.

4.3.6 Version 1.1.1.0

➤ Version 1.1.1.0

- Improvement some work products to fix gaps of ASPICE.

4.3.7 Version 1.2.0.0

➤ Version 1.2.0.0

- Update SQT, SIT, SUT to verify WdgM in TC33x environment.
- Update WdgM to support reset partition.
- Update WdgM for Cyber-Security violations

4.3.8 Version 1.3.0.0

➤ Version 1.3.0.0

- Update SQT, SIT, SUT to verify WdgM in TC37x environment.
- Update WdgM to fix warning Tasking compiler

4.3.9 Version 1.3.0.1

➤ Version 1.3.0.1

- Clarify the copyright of code in E-Code, Generated Code.
- Apply the latest template of DeliveryBoxHistory.

4.3.10 Version 1.3.1.0

➤ Version 1.3.1.0

- Update code for safety coding rule and UNECE standard
- Update generator of SWCD file for fixing error relates to Rte module.

4.3.11 Version 1.3.1.1

➤ Version 1.3.1.1

- Modify justification comment for UNECE

4.3.12 Version 1.3.2.0

- Version 1.3.2.0
 - Update format comment in E-Code to fix violation related 'Comment Out' rule D4.4
 - Add new App_WdgM_GeneratorSampleApplication in Test_Applications
 - Update SRS mapping in Generated Code
 - Update SIT ID in Test_Applications to fix duplicated violation

4.3.13 Version 1.3.2.1

- Version 1.3.2.1
 - Fix Security Coding Report

4.3.14 Version 1.3.3.0

- Version 1.3.3.0
 - Modify OsCounter Destination Path in PDF

4.4 Limitations

- Deadline Max / Min minimum allowable range of Deadline Supervision
Deadline Supervision uses OsCounter internally. Therefore, if Deadline Max / Min value of Deadline Supervision is smaller than the resolution of OsCounter referenced by WdgMSupervisedEntity, Deadline Supervision execution time cannot be monitored.
- WdgM_MainFunction cycle setting when using Wdg
Triggering of Wdg is performed by a Gtm timer driven by the Wdg driver. This timer is updated / stopped / restarted with the new timeout value whenever WdgM_MainFunction is executed. When the WdgM_MainFunction cycle is set smaller than the WdgNSlow / FastServiceGtmCbKTime value of the Wdg setting, the Gtm timer callback is not called, so a watchdog reset occurs.
- Watch Watchdog triggering in Sleep Mode
Periodic triggering is required when using a hardware watchdog in sleep mode.

4.5 Deviations

- **EcuC Partition Ref related function of WdgMSupervisionEntity** (SWS - ECUC_WdgM_00360)
When restarting a partition, it is used as information to exclude the supervised entity located in the partition from the monitoring target of the currently used WdgMMode (deactivate).
Function not supported
- **Os Application Ref related function of WdgMSupervisionEntity** (SWS - Chap 7.2.3, 7.5.1.2)
The status of the Supervised Entity located in the Non-trusted OsApplication
WDGM_LOCAL_STATUS_FAILED is used as information to restart only the partition when transitioning.
Function not supported

➤ **Change WdgMMode when calling WdgM_DeInit** (SWS - Chap 7.4.3)

When WdgM_DeInit is called, it should be changed to WdgMMode set in advance for DeInit. The currently implemented method is to perform WdgM_SetTriggerCondition using WdgMConfigSet / WdgMDeInitTimeout setting value in WdgMTrigger of WdgMMode in use without changing WdgMMode.

➤ **PPort Short Name** (SWS - WDGM147, WDGM149)

In the current implementation, the short name of the PPort generated by WdgM is created as localSupervisionStatus_ <short name of supervised entity>, localSupervision_<short name of supervised entity>_<short name of checkpoint> and mode_ <short name of supervised entity> so that the target PPort can be easily known when connecting to the RPort.

.

5. Configuration Guide

5.1 WdgMGeneral Container

Parameter Name	Value	Category
Dem Stopped Supervision Report ¹⁾	true	F
Dev Error Detect	true	F
Immediate Reset ²⁾	false	F
Off Mode Enabled ³⁾	true	F
Version Info Api	false	F

- 1) When Global Supervision Status transitions to WDGM_GLOBAL_STATUS_STOPPED state
 - DemEvent (WDGM_E_SUPERVISION) reporting function
- 2) When Global Supervision Status transitions to WDGM_GLOBAL_STATUS_STOPPED status
 - A function that calls Mcu_PerformReset when Mcu provides an immediate reset function when an immediate MCU reset is required
- 3) When calling WdgM_Init, WdgM_SetMode
 - A function that allows WDGMIF_OFF_MODE by setting WdgMMode / WdgMTrigger / Watchdog Mode
 - When setting true, Disable Allowed must also be set to true in Wdg Driver setting

5.1.1 WdgMWatchdog setting

WdgIfDevice setting to be managed by WdgM

Parameter Name	Value	Category
Short Name	Use settings when deploying the platform	C
Name	Use settings when deploying the platform	F
Device Ref ¹⁾	Use settings when deploying the platform	F

- 1) WdgIfDevice reference that references Wdg

5.1.2 WdgMSupervisedEntity setting

Parameter Name	Value	Category
Short Name ¹⁾	User Defined	C
Id ²⁾	User Defined	C
Ecuc Partition Ref ³⁾		N
Os Application Ref ⁴⁾		N
Internal Checkpoint Initial Ref ⁵⁾	User Defined	C
Internal Checkpoint Final Ref ⁶⁾	User Defined	C
Os Counter ⁷⁾ (Vendor specific)	User Defined	C

- The following P Port with ShortName name is created in Swcd_WdgM.arxml
 - localSupervision_ 'ShortName': WdgM_AliveSupervision(ClientServerInterface)
 - mode_ 'ShortName': WdgM_IndividualMode(ModeSwitchInterface)
- Used as an argument of the following API
 - WdgM_SetMode: Use Mode ID
 - WdgM_CheckpointReached: It is pre-mapped to Port with Port API Option, so the user does not directly use it
 - WdgM_GetLocalStatus: Used as a Supervised Entity Id argument to get Local Supervision Status
 - WdgM_GetFirstExpiredSEID: Used as an argument to obtain the first expired Supervised Entity Id
- Partition reference where Supervised Entity is located
 - When restarting a partition, it is used as information to exclude the supervised entity located in the partition from the monitoring target of the currently used WdgMMode (deactive)
 - No function**
- OsApplication reference where Supervised Entity is located
 - When the status of Supervised Entity located in Non-trusted OsApplication transitions to WDG_M_LOCAL_STATUS_FAILED, it is used as information to restart only the partition.

- No function

- 5) Internal Graph's Start Checkpoint Reference
- 6) Internal Graph's End Checkpoints Reference
- 7) When the Supervised Entity uses Deadline Supervision, OsCounter reference setting to be used inside WdgM for timestamp acquisition and comparison

5.1.3 WdgMCheckpoint setting

Targets of Alive / Deadline / Logical Supervision

Parameter Name	Value	Category
Short Name	User Defined	C
Id ¹⁾	Sequential increments starting from 0	C

- 1) Used as an argument when calling WdgM_CheckpointReached

5.1.4 WdgMInternalTransition setting

Transitions that make up the internal graph

Parameter Name	Value	Category
Short Name	User Defined	C
Source Ref ¹⁾	User Defined	C
Dest Ref ¹⁾	User Defined	C

- 1) Internal Transition Start / End Checkpoint Reference

5.2 WdgMConfigSet Container

Parameter Name	Value	Category
Short Name	User Defined	C
Initial Mode ¹⁾	User Defined	C
De Init Timeout ²⁾ (Vendor specific)	User Defined	C
CoreID ³⁾ (Vendor specific)	User Defined	C

- 1) WdgMMode to use when calling WdgM_Init
- 2) WdgMDeInitTimeout to use when calling WdgM_DeInit
- 3) WdgMCoreID to use in multicore mode to determines which core will execute the WdgM's Main functions.

5.2.1 WdgMDemEventParameterRefs setting

Parameter Name	Value	Category
Short Name	User Defined	C
WDGM_E_SUPERVISION ¹⁾	User Defined	F
WDGM_E_SET_MODE ¹⁾	User Defined	F

- 1) Set to refer to DemEventParameter for WdgM error report set in Dem module

5.2.2 WdgMMode setting

Presets for monitoring conditions, monitoring targets, and Wdg settings can be changed at runtime

Parameter Name	Value	Category
Short Name	User Defined	C
Expired Supervision Cycle Tol ¹⁾	User Defined	C
Id ²⁾	Sequential increments starting from 0	C
Supervision Cycle ³⁾	Set equal to WdgM_MainFunction cycle	C

- 1) When transitioning to WDGM_GLOBAL_STATUS_EXPIRED state, the number of cycles that maintain the state before transitioning to WDGM_GLOBAL_STATUS_STOPPED (multiple of WdgM_MainFunction cycle)
 - Used to secure delay before requesting reset (when transitioning to WDGM_GLOBAL_STATUS_STOPPED state)
 - Ex) Expired Supervision Cycle Tol: 3, Supervision Cycle : 0.01(sec)
WDGM_GLOBAL_STATUS_EXPIRED state for 3 x 0.01 (s) = 0.03 (sec)
- 2) Used as an argument of the following API

- WdgM_SetMode: Used as a change request WdgMMode Id argument
 - WdgM_GetMode: Currently used as WdgMMode Id acquisition factor
- 3) Same setting as WdgM_MainFunction cycle
- See Rte / BswInstance_WdgM / RteBswEventToTaskMapping_TE_WdgM / BswEventRef

5.2.3 WdgMAliveSupervision setting

Parameter Name	Value	Category
Short Name	User Defined	C
Expected Alive Indications ¹⁾	User Defined	C
Max Margin ²⁾	User Defined	C
Min Margin ²⁾	User Defined	C
Supervision Reference Cycle ³⁾	User Defined	C
Checkpoint Ref ⁴⁾	User Defined	C

- 1) The number of WdgM_CheckpointReached calls of the target checkpoint required during the Supervision Reference Cycle setup cycle
- 2) The number of times allowed by adding (+) / decreasing (-) to Expected alive indications
Ex) Expected Alive Indications: 4, Max Margin : 2, Min Margin : 1
 $(4-1) \leq \text{number of alive indications allowed} \leq (4 + 2)$
- 3) Multiple of WdgM_MainFunction cycle as a monitoring cycle for the target checkpoint
Supervision Reference Cycle: 2, WdgM_MainFunction Cycle: 0.01 (sec)
 $2 \times 0.01(\text{s}) = 0.02(\text{sec})$
- 4) Checkpoint for Alive Supervision

5.2.4 WdgMDeadlineSupervision setting

Parameter Name	Value	Category
Short Name	User Defined	C
Deadline Max ¹⁾	User Defined	C
Deadline Min ¹⁾	User Defined	C
Deadline Start Ref ²⁾	User Defined	C
Deadline Start Ref ²⁾	User Defined	C

- 1) Allowable execution time boundary value between start and end checkpoints
Ex) Deadline Max: 0.035(sec), Deadline Min : 0.005(sec)
 $0.005(\text{sec}) \leq \text{Performance time} \leq 0.035(\text{sec})$
- 2) Deadline Supervision Target Start / End Checkpoint

5.2.5 WdgMAliveSupervision setting

Parameter Name	Value	Category
Short Name	User Defined	C
Expected Alive Indications ¹⁾	User Defined	C
Max Margin ²⁾	User Defined	C
Min Margin ²⁾	User Defined	C
Supervision Reference Cycle ³⁾	User Defined	C
Checkpoint Ref ⁴⁾	User Defined	C

- 1) The number of WdgM_CheckpointReached calls of the target checkpoint required during the Supervision Reference Cycle setup cycle
- 2) The number of times allowed by adding (+) / decreasing (-) to Expected alive indications
Ex) Expected Alive Indications: 4, Max Margin : 2, Min Margin : 1
 $(4-1) \leq \text{number of alive indications allowed} \leq (4 + 2)$
- 3) Multiple of WdgM_MainFunction cycle as a monitoring cycle for the target checkpoint
Supervision Reference Cycle: 2, WdgM_MainFunction Cycle: 0.01 (sec)
 $2 \times 0.01(\text{s}) = 0.02(\text{sec})$
- 4) Checkpoint for Alive Supervision

5.2.6 WdgMDeadlineSupervision setting

Parameter Name	Value	Category
Short Name	User Defined	C
Deadline Max ¹⁾	User Defined	C
Deadline Min ¹⁾	User Defined	C
Deadline Start Ref ²⁾	User Defined	C
Deadline Start Ref ²⁾	User Defined	C

- 1) Allowable execution time boundary value between start and end checkpoints
Ex) Deadline Max: 0.035(sec), Deadline Min : 0.005(sec)
 $0.005(\text{sec}) \leq \text{Performance time} \leq 0.035(\text{sec})$
- 2) Deadline Supervision Target Start / End Checkpoint

5.2.7 WdgMExternalLogicalSupervision setting

Parameter Name	Value	Category
Short Name	User Defined	C
External Checkpoint Initial Ref ¹⁾	User Defined	C
External Checkpoint Final Ref ²⁾	User Defined	C

- 1) External Graph Start Checkpoints Reference

2) External Graph's Exit Checkpoints Reference

5.2.8 WdgMExternalTransition setting

Transitions that make up the external graph

Parameter Name	Value	Category
Short Name	User Defined	C
Source Ref ¹⁾	User Defined	C
Dest Ref ¹⁾	User Defined	C

1) External Transition Start / End Checkpoint Reference

5.2.9 WdgMLocalStatusParams setting

Monitoring target setting in WdgMMode

Parameter Name	Value	Category
Short Name	User Defined	C
Failed Alive Supervision Ref Cycle Tol ¹⁾	User Defined	C
Local Status Supervised Entity Ref ²⁾	User Defined	C

- 1) If the Alive Supervision result is Incorrect, the number of cycles that do not immediately transition to the WDGM_LOCAL_STATUS_EXPIRED state and maintain the WDGM_LOCAL_STATUS_FAILED state (multiple of WdgM_MainFunction cycles)
- 2) Supervised Entity reference to be monitored. A Supervised Entity could be referenced from Modes of a WdgMConfigSet container only, it cannot be referenced from Modes of the different WdgMConfigSet containers.

5.2.10 WdgMTrigger setting

Wdg setting in WdgMMode

Parameter Name	Value	Category
Short Name	User Defined	C
Condition Value ¹⁾	User Defined	C
Watchdog Mode ²⁾	User Defined	C
Watchdog Ref ³⁾	User Defined	C

- 1) WdgM_MainFunction is a value that WdgM sends to Wdg through WdgIf so that WdgM can trigger watchdog normally when WdgM status is normal. Wdg must ensure that the watchdog does not expire for the time set as the Condition Value. Condition Value must be larger than watchdog timeout value according to Wdg mode.
- 2) Wdg operation mode
 - WDGIF_OFF_MODE: Stop Wdg (WdgMGeneral / Off Mode Enabled: true, WdgGeneral / Disable Allowed: true)
 - WDGIF_SLOW_MODE: Apply WdgSettingsConfig / WdgSettingsSlow settings
 - WDGIF_FAST_MODE: Apply WdgSettingsConfig / WdgSettingsFast settings
- 3) WdgMGeneral / Watchdog Reference

5.3 System Configuration

5.3.1 ApplicationSwComponentType setting

Reference file: Configuration / System / Swcd_App / App_WdgM.arxml

- 1) ApplicationSwComponent produce
- 2) RPort produce
 - A. For calling CheckpointReached: One per Supervised Entity
 - i. **Correspondence PPort:**
/ WdgM / ServiceSwComponentTypes / WdgM / localSupervisionStatus_'Supervised Entity Short Name'_ 'Checkpoint Short Name'
 - ii. **Required Interface:**
/ WdgM / ClientServerInterfaces / WdgM_LocalSupervision
 - iii. **Client Com Spec:**
/ WdgM / ClientServerInterfaces / WdgM_LocalSupervision / CheckpointReached
 - B. For receiving Local Supervision Status change event: One per Supervised Entity (Optional)
 - i. **Correspondence PPort:**
/ WdgM / ServiceSwComponentTypes / WdgM / mode_'Supervised Entity Short Name'
 - ii. **Required Interface:**
/ WdgM / ModeSwitchInterfaces / WdgM_LocalMode
 - C. For receiving Global Supervision Status change event: Create only one (Optional)
 - i. **Correspondence PPort:**
/ WdgM / ServiceSwComponentTypes / WdgM / globalMode
 - ii. **Required Interface:**
/ WdgM / ModeSwitchInterfaces / WdgM_GlobalMode
 - D. Using WdgM Api: Create only one
 - i. **Correspondence PPort:**
/ WdgM / ServiceSwComponentTypes / WdgM / globalSupervision
 - ii. **Required Interface:**
/ WdgM / ClientServerInterfaces / WdgM_GlobalSupervision
 - iii. **Client Com Spec:**
Optionally add only necessary operations
- 3) Creating Internal Behavior
- 4) Creating Runnable
 - A. Add Runnable according to design
- 5) Create Synchronous Server Call Point or Mode Access Point
 - A. Runnable Synchronous Server Call Point
 - i. RPort and Operation setting to use
- 6) Create Swc Mode Switch Event
 - A. Event setting to receive status change

5.3.2 Assembly Sw Connector Settings

※ Please refer to Mobilegene Studio's Help for the settings for Assembly Sw Connector

6. Application Programming Interface (API)

6.1 Type Definitions

6.1.1 WdgM_ModeType

Type:	uint8
Range:	0-<Number of Modes>
Description:	This type identifies the different modes that are configured for the Watchdog Manager

6.1.2 WdgM_SupervisedEntityType

Type:	uint16
Range:	0-<Number of Supervised Entities >
Description:	This type identifies an individual SE for the Watchdog Manager in all modes

6.1.3 WdgM_CheckpointIdType

Type:	uint16
Range:	0-<Maximum number of Checkpoints>
Description:	This type identifies a Checkpoint in the context of a SE for the Watchdog Manager. Note that an individual Checkpoint can only be identified by the pair of SE ID and Checkpoint ID.

6.1.4 WdgM_LocalStatusType

Type:	uint8		
Range:	WDGM_LOCAL_STATUS_OK	0	The supervision of this SE has not shown any failures
	WDGM_LOCAL_STATUS_FAILED	1	The supervision of this SE has failed but can still be "healed". I.e., if the SE returns to a normal behaviour, its supervision state will also return to WDGM_LOCAL_STATUS_OK. Furthermore, the number of times that the supervision has failed has not yet exceeded a configurable limit. When this limit has been exceeded the state will change to WDGM_LOCAL_STATUS_EXPIRED
	WDGM_LOCAL_STATUS_EXPIRED	2	The supervision of this SE has failed permanently. This state cannot be left
	WDGM_LOCAL_STATUS_DEACTIVATED	4	The supervision of this SE is temporarily disabled
Description:	This type shall be used for variables that represent the current status of supervision for individual Supervised Entities		

6.1.5 WdgM_GlobalStatusType

AUTOSAR WdgM User Manual

Document
(DOC NO)

number

SHT/SHTS
25 / 70

Type:	uint8		
Range:	WDGM_GLOBAL_STATUS_OK	0	Supervision did not show any failures
	WDGM_GLOBAL_STATUS_FAILED	1	Supervision has failed but is still within the limit of allowed failures
	WDGM_GLOBAL_STATUS_EXPIRED	2	Supervision has failed, the allowed limit of failures has been exceeded, but the Watchdog Driver has not yet been instructed to stop triggering
	WDGM_GLOBAL_STATUS_STOPPED	3	Supervision has failed, the allowed limit of failures has been exceeded, and the Watchdog Driver has been instructed to stop triggering. A watchdog reset is about to happen
	WDGM_GLOBAL_STATUS_DEACTIVATED	4	WdgM is not initialized and therefore will not manage the watchdogs
Description:	This type shall be used for variables that represent the global supervision status of the Watchdog Manager module		

6.2 Macro Constants

None

6.3 Functions

6.3.1 WdgM_Init

Function Name	WdgM_Init
Syntax:	FUNC(void, WDGM_CODE) WdgM_Init(P2CONST(WdgM_ConfigType, WDGM_CONST, WDGM_CONST)ConfigPtr)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non reentrant
Parameters (In)	ConfigPtr: Pointer to post-build configuration data
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	This service performs initialization of WdgM
Preconditions	None
Configuration Dependency	None

6.3.2 WdgM_DeInit

Function Name	WdgM_DeInit
Syntax:	FUNC(void, WDG_CODE) WdgM_DeInit(void)
Service ID	0x01
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	This service performs Deinitialization of WdgM
Preconditions	Watchdog Manager should be initialized
Configuration Dependency	None

6.3.3 WdgM_GetVersionInfo

Function Name	WdgM_GetVersionInfo
Syntax:	FUNC(void, WDGM_CODE) WdgM_GetVersionInfo (P2VAR(Std_VersionInfoType, AUTOMATIC, WDGM_APPL_DATA) VersionInfo)
Service ID	0x02
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	VersionInfo: Pointer to where to store the information of the module WdgM
Return Value	None
Description	This API reads the version information of WdgM
Preconditions	None
Configuration Dependency	None

6.3.4 WdgM_SetMode

Function Name	WdgM_SetMode
Syntax:	FUNC(Std_ReturnType, WDGM_CODE) WdgM_SetMode(WdgM_ModeType Mode)
Service ID	0x03
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	Mode: One of the configured WdgM modes
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	This service performs switching between different modes of WdgM
Preconditions	Watchdog Manager should be initialized
Configuration Dependency	None
In Communication with application SW-C	Rte_Call_<P>_SetMode (WdgM_ModeType ddMode) <P> : <Runnable Name>_<R-Port Name>

6.3.5 WdgM_GetMode

Function Name	WdgM_GetMode
Syntax:	FUNC(Std_ReturnType, WDGM_CODE) WdgM_GetMode(P2VAR(WdgM_ModeType, WDGM_DATA, WDGM_APPL_DATA) Mode)
Service ID	0x0b
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	Mode
Parameters (Inout)	None
Parameters (Out)	Mode
Return Value	Std_ReturnType
Description	This service Returns the current mode of WdgM
Preconditions	Watchdog Manager should be initialized
Configuration Dependency	None
In Communication with application SW-C	Rte_Call_<P>_GetMode (WdgM_ModeType* pMode) <P> : <Runnable Name>_<R-Port Name>

6.3.6 WdgM_CheckpointReached

Function Name	WdgM_CheckpointReached
Syntax:	FUNC(Std_ReturnType, WdGM_CODE) WdgM_CheckpointReached(WdgM_SupervisedEntityType SEID, WdgM_CheckpointIDType CheckpointID)
Service ID	0x0e
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	SEID: Identifier of the Supervised Entity that reports a CheckpointID CheckpointID: Identifier of the Checkpoint within a Supervised Entity that has been reached
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	Indicates to the WdgM that a checkpoint within a supervised entity has been reached
Preconditions	Watchdog Manager should be initialized
Configuration Dependency	None
In Communication with application SW-C	Rte_Call_<P>_CheckpointReached (WdgM_CheckpointIDType ddCheckpointID) <P> : <Runnable Name>_<R-Port Name>

6.3.7 WdgM_GetLocalStatus

Function Name	WdgM_GetLocalStatus
Syntax:	FUNC(Std_ReturnType, WDGM_CODE) WdgM_GetLocalStatus(WdgM_SupervisedEntityType SEID, P2VAR(WdgM_LocalStatusType, WDGM_DATA, WDGM_APPL_DATA) Status)
Service ID	0x0c
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	SEID: Identifier of the supervised entity whose supervision status shall be returned
Parameters (Inout)	None
Parameters (Out)	Status: Supervision status of the given supervised entity
Return Value	Std_ReturnType
Description	This service provides the supervision status of an individual supervised entity
Preconditions	Watchdog Manager should be initialized
Configuration Dependency	None
In Communication with application SW-C	Rte_Call_<P>_GetLocalStatus (WdgM_SupervisedEntityType ddSEID, WdgM_LocalStatusType* pStatus) <P> : <Runnable Name>_<R-Port Name>

6.3.8 WdgM_GetGlobalStatus

Function Name	WdgM_GetGlobalStatus
Syntax:	FUNC(Std_ReturnType, Wdgm_CODE) WdgM_GetGlobalStatus (P2VAR(WdgM_GlobalStatusType, Wdgm_DATA, Wdgm_APPL_DATA) Status)
Service ID	0x0d
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	Status: Global supervision status of the WdgM
Return Value	Std_ReturnType
Description	This service provides global supervision status of WdgM
Preconditions	Watchdog Manager should be initialized
Configuration Dependency	None
In Communication with application SW-C	Rte_Call_<P>_GetGlobalStatus (WdgM_GlobalStatusType* pStatus) <P> : <Runnable Name>_<R-Port Name>

6.3.9 WdgM_PerformReset

Function Name	WdgM_PerformReset
Syntax:	FUNC(void, WDGM_CODE) WdgM_PerformReset(void)
Service ID	0x0f
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	This service instructs the WdgM to initiate a watchdog reset
Preconditions	Watchdog Manager should be initialized
Configuration Dependency	None
In Communication with application SW-C	Rte_Call_<P>_PerformReset(void) <P> : <Runnable Name>_<R-Port Name>

6.3.10 WdgM_GetFirstExpiredSEID

Function Name	WdgM_GetFirstExpiredSEID
Syntax:	FUNC(Std_ReturnType, WDGM_CODE) WdgM_GetFirstExpiredSEID(P2VAR(WdgM_SupervisedEntityType, WDGM_DATA, WDGM_APPL_DATA) SEID)
Service ID	0x10
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	SEID: Identifier of the supervised entity that first reached the state WDGM_LOCAL_STATUS_EXPIRED
Return Value	Std_ReturnType
Description	This service returns SEID that first reached the state WDGM_LOCAL_STATUS_EXPIRED
Preconditions	None
Configuration Dependency	None
In Communication with application SW-C	Rte_Call_<P>_GetFirstExpiredSEID (WdgM_SupervisedEntityType* pSEID) <P> : <Runnable Name>_<R-Port Name>

7. Generator

7.1 Generator Message

7.1.1 Error Messages

- 1) **ERR013001: The reference path is empty for the parameter "Parameter Name" in the container "Container Name", having short name "Container Short Name".**
 - This error occurs, if the Reference path is not configured for the parameter "Parameter Name" in the container "Container Name".

Container Name	Parameter Name
WdgMSupervisedEntity	WdgMInternalCheckpointInitialRef
	WdgMInternalCheckpointFinalRef
	WdgMEcucPartitionRef
	WdgMOSCounter
WdgMInternalTransition	WdgMInternalTransitionSourceRef
WdgMWatchdog	WdgMWatchdogDeviceRef
WdgMConfigSet	WdgMInitialMode
WdgMDemEventParameterRefs	WDGM_E_SET_MODE
	WDGM_E_SUPERVISION
WdgMAliveSupervision	WdgMAliveSupervisionCheckpointRef
WdgMDeadlineSupervision	WdgMDeadlineStartRef
	WdgMDeadlineStopRef
WdgMExternalLogicalSupervision	WdgMExternalCheckpointInitialRef
	WdgMExternalCheckpointFinalRef
WdgMExternalTransition	WdgMExternalTransitionSourceRef
	WdgMExternalTransitionDestRef
WdgMTrigger	WdgMTriggerWatchdogRef
WdgMLocalStatusParams	WdgMLocalStatusSupervisedEntityRef

- 2) **ERR013002: The parameter 'Parameter Name' in the container 'Container Name' should be configured.**
 - This error occurs, if the parameter 'Parameter Name' in the container 'Container Name' is not configured.

AUTOSAR WdgM User Manual

Document
(DOC NO)

number

SHT/SHTS
38 / 70

Container Name	Parameter Name
WdgMGeneral	WdgMVersionInfoApi
	WdgMDevErrorDetect
	WdgMDemStoppedSupervisionReport
	WdgMImmediateReset
	WdgMOffModeEnabled
WdgMSupervisedEntity	WdgMSupervisedEntityId
WdgMCheckpoint	WdgMCheckpointId
WdgMWatchdog	WdgMWatchdogName
WdgMMode	WdgMModelId
	WdgMExpiredSupervisionCycleTol
	WdgMSupervisionCycle
WdgMAliveSupervision	WdgMSupervisionReferenceCycle
	WdgMExpectedAliveIndications
	WdgMMinMargin
	WdgMMaxMargin
WdgMDeadlineSupervision	WdgMDeadlineMin
	WdgMDeadlineMax
WdgMTrigger	WdgMWatchdogMode
	WdgMTriggerConditionValue
WdgMLocalStatusParams	WdgMFailedAliveSupervisionRefCycleTol

container 'Container Name', having short name 'Container Short Name' is incorrect.

- This error occurs, if incorrect reference path is configured for the parameter 'Parameter Name' in the container 'Container Name'.

Container Name	Parameter Name
WdgMGeneral	WdgMVersionInfoApi
	WdgMDevErrorDetect
	WdgMDemStoppedSupervisionReport
	WdgMImmediateReset
	WdgMOffModeEnabled
WdgMSupervisedEntity	WdgMSupervisedEntityId
WdgMCheckpoint	WdgMCheckpointId
WdgMWatchdog	WdgMWatchdogName
WdgMMode	WdgMModelId
	WdgMExpiredSupervisionCycleTol
	WdgMSupervisionCycle
WdgMAliveSupervision	WdgMSupervisionReferenceCycle
	WdgMExpectedAliveIndications
	WdgMMinMargin
	WdgMMaxMargin
WdgMDeadlineSupervision	WdgMDeadlineMin
	WdgMDeadlineMax
WdgMTrigger	WdgMWatchdogMode
	WdgMTriggerConditionValue
WdgMLocalStatusParams	WdgMFailedAliveSupervisionRefCycleTol

4) **ERR013004: The value of the 'Parameter Name' in the 'Container Name' container is out of range.**

- This error occurs, if the value of the 'Parameter Name' in the 'Container Name' container is out of range

Container Name	Parameter Name	Range
WdgMSupervisedEntity	WdgMSupervisedEntityId	0..65535
WdgMCheckpoint	WdgMCheckpointId	0..65535
WdgMMode	WdgMModelId	0..255
	WdgMExpiredSupervisionCycleTol	0..65535
	WdgMSupervisionCycle	0..INF
WdgMAliveSupervision	WdgMSupervisionReferenceCycle	1..65535
	WdgMExpectedAliveIndications	0..65535
	WdgMMinMargin	0..255
	WdgMMaxMargin	0..255
WdgMDeadlineSupervision	WdgMDeadlineMin	0..INF
	WdgMDeadlineMax	0..INF
WdgMTrigger	WdgMWatchdogMode	WDGIF_FAST_MODE WDGIF_SLOW_MODE WDGIF_OFF_MODE
	WdgMTriggerConditionValue	1..65535

- ERR013005: The number of instance of the 'Container Name' is out of range.**
 - This error occurs, If the number of instance of the 'Container Name' container is out of range (lowerMultiplicity-upperMultiplicity)
- ERR013006: The number of instance of the 'Parameter Name' of the 'Container Name' is out of range.**
 - This error occurs, if the number of instance of the Parameter Name' parameter is out of range (lowerMultiplicity-upperMultiplicity).
- ERR013007: The Supervised Entity ID "WdgMSupervisedEntityId.VALUE" of the "WdgMSupervisedEntity.SHORT-NAME" is not unique.**
 - This error occurs:
For all containers "WdgMSupervisedEntity"
if the parameter "WdgMSupervisedEntityId" of each container "WdgMSupervisedEntity" is equal to the parameter "WdgMSupervisedEntityId" of the other "WdgMSupervisedEntity" containers.
- ERR013008: The Check point ID "WdgMCheckpointId.VALUE" of the Supervised Entity "WdgMSupervisedEntity.SHORT-NAME" is not started from zero.**

- This error occurs:
For each container "WdgMSupervisedEntity" in the container "WdgMGeneral"{
With the first container "WdgMCheckpoint", store the parameter "WdgMCheckpointId" to
CheckpointId. if the parameter "WdgMCheckpointId" is not equal to zero.
- 9) **ERR013008: The Check point ID "WdgMCheckpointId.VALUE" of the Supervised Entity "WdgMSupervisedEntity.SHORT-NAME" is not unique or sequential.**
 - This error occurs:
For each container "WdgMSupervisedEntity" in the container "WdgMGeneral"{
For all containers "WdgMCheckpoint" from the second{
CheckpointId = CheckpointId + 1
if the parameter "WdgMCheckpointId" is not equal to (CheckpointId) then error. }
}
- 10) **ERR013009: Value configured for the parameter "WdgMModelId" is not start from <0>.**
 - This error occurs:
For all containers WdgMMode" in all containers "WdgMConfigSet":
With the first container "WdgMMode", store the parameter "WdgMModelId" to ModelId.
If ModelId is not equal to zero.
- 11) **ERR013010: Value configured for the parameter "WdgMModelId" is not sequential.**
 - This error occurs:
For all containers "WdgMMode" from the second:
ModelId = ModelId + 1
If the parameter "WdgMModelId" is not equal to (ModelId).
- 12) **ERR013011: The Mcu_PerformReset API is not available.**
 - This error occurs, if the "WdgM/WdgMGeneral/WdgMImmediateReset" parameter is "True" and the "Mcu/McuGeneralConfiguration/McuPerformResetApi" parameter is "False".
- 13) **ERR013012: Check points of the {WdgM/WdgMConfigSet/WdgMMode/WdgMDeadlineSupervision.SHORT_NAME} in {WdgM/WdgMConfigSet/WdgMMode.SHORT_NAME} are not in the same Supervised Entity..**
 - This error occurs:
For each "WdgMConfigSet" container:
For each "WdgMMode" container:
For each "WdgMDeadlineSupervision" container:
if the check points referenced by "WdgMDeadlineStartRef" and
"WdgMDeadlineStopRef" from the difference "WdgMSupervisedEntity" container.
- 14) **ERR013013: WdgMInternalTransitions configured for {WdgM/WdgMGeneral/WdgMSupervisedEntity.SHORT_NAME} do not form an internal graph.**
 - This error occurs:
For each "WdgMSupervisedEntity" container:
For each "WdgMInternalTransition" container:
if the check points referenced by "WdgMInternalTransitionDestRef" and
"WdgMInternalTransitionSourceRef" from the difference "WdgMSupervisedEntity" container.
- 15) **ERR013014: The check point**

{WdgM/WdgMGeneral/WdgMSupervisedEntity.SHORT_NAME}/{WdgM/WdgMGeneral/WdgMSupervisedEntity/WdgMCheckpoint.SHORT_NAME}) belongs to more than one internal graph.

- This error occurs:
 For each "WdgMSupervisedEntity" container:
 For each internal graph:
 For each "checkpointID" of the internal graph:
 If the "checkpointID" is equal to check point ID of other internal graph then error.
 With internal graph which was checked before, doesn't have to check in the next step of the loop.

16) **ERR013015: The Checkpoint**

({WdgM/WdgMGeneral/WdgMSupervisedEntity/WdgMSupervisedEntityID.VALUE},{WdgM/WdgMGeneral/WdgMSupervisedEntity/WdgMCheckpoint/WdgMCheckpointId.VALUE}) belongs to internal and external graph."})

- This error occurs:
 For each "WdgMSupervisedEntity" container:
 For each internal graph:
 If any check point of the internal graph is same as any check point in the external graphs of all "WdgMMode" containers (check points in the same "WdgMSupervisedEntity" have the same "WdgMCheckpointId").

17) **ERR013016: The checkpoint**

({WdgM/WdgMGeneral/WdgMSupervisedEntity/WdgMSupervisedEntityID.VALUE},{WdgM/WdgMGeneral/WdgMSupervisedEntity/WdgMCheckpoint/WdgMCheckpointId.VALUE}) belong to more than one external graph."})

- This error occurs:
 For each "WdgMConfigSet" container:
 For each "WdgMMode" container:
 List out check points of each external graph.
 If there's any "checkpointID" in an external graph is equal to check point ID in other external graphs.

18) **ERR013019: The value of the WdgMDeadlineMax must be greater than WdgMDeadlineMin in '<WdgMMode.ShortName>/<WdgMDeadlineSupervision.ShortName>'.**

- This error occurs:
 For each "WdgMConfigSet" container:
 For each "WdgMMode" container:
 For each "WdgMDeadlineSupervision" container:
 If WdgMDeadlineMax.VALUE <= WdgMDeadlineMin.VALUE.

19) **ERR013020: A Dead line supervision cannot refer to the same check point.**

- This error occurs:
 For each "WdgMConfigSet" container:
 For each "WdgMMode" container:
 For each "WdgMDeadlineSupervision" container:
 If check point ID referred by "WdgMDeadlineStartRef" has the same checkpoint ID referred by "WdgMDeadlineStopRef".

20) **ERR013023: The expected Alive indication should not lower than Min Margin.**

- This error occurs:
 For each "WdgMConfigSet" container:
 For each "WdgMMode" container:
 For each "WdgMAliveSupervision" container:

if WdgMExpectedAliveIndications.VALUE < WdgMMinMargin.VALUE.

21) **ERR013024: WdgMExternalLogicalSupervision is configured for {/AUTRON/WdgM/WdgMConfigSet/WdgMMode/WdgMExternalLogicalSupervision.SHORT_NAME} but it doesn't form a graph**

- This error occurs:
For each "WdgMConfigSet" container:
For each "WdgMMode" container:
For each "WdgMExternalLogicalSupervision" container:
If WdgMExternalLogicalSupervision is configured but it doesn't form a graph

22) **ERR013025: There is more than one External Logical Supervision graph configured for {/AUTRON/WdgM/WdgMConfigSet/WdgMMode/WdgMExternalLogicalSupervision.SHORT_NAME}**

- This error occurs:
For each "WdgMConfigSet" container:
For each "WdgMMode" container:
For each "WdgMExternalLogicalSupervision" container:
If WdgMExternalLogicalSupervision is configured and there is more than one external graph configured

23) **ERR013026: Maximum Alive Supervision Reference Cycle overflow**

- This error occurs:
For each "WdgMConfigSet" container:
For each "WdgMMode" container:
For each "WdgMAliveSupervision" container:
Calculate least common multiple (LCM) of
WdgMSupervisionReferenceCycle
If LCM value is overflow (value is greater than 2147483647) then report error

24) **ERR013027: There is no Supervised Entity configured for this Mode: "WdgMMode.SHORT-NAME"**

- This error occurs:
For each "WdgMConfigSet" container:
For each "WdgMMode" container:
If there's no "WdgMLocalStatusParams" configured.

25) **ERR013028: Value configured for the parameter "WdgMCoreID" is empty**

- This error occurs in multicore mode:
For each "WdgMConfigSet" container:
If "WdgMCoreID" is empty

26) **ERR013029: Value configured for the parameter "WdgMCoreID" is not unique**

- This error occurs in multicore mode:
For each "WdgMConfigSet" container:
If WdgMCoreID.value of a WdgMConfigSet equals to WdgMCoreID.value of other WdgMConfigSet(s)

27) **ERR013030: The SE names <SE shortname> is referenced in more than 1 ConfigSet**

- This error occurs:
For each "WdgMConfigSet" container:
For each "WdgMMode" container:
For each "WdgMLocalStatusParams" container:
SE = Gets reference of "WdgMLocalStatusSupervisedEntityRef"
Get assigned SupervisedEntity of the mode in the other WdgMConfigSet: SE_NEXT = WdgMConfigSet. WdgMMode.

WdgMLocalStatusParams.
WdgMLocalStatusSupervisedEntityRef.

If SE = SE_NEXT

28) ERR013031: There is no WdgMOsCounter configured for <WdgMSupervisedEntity.ShortName>

- This error occurs:
 - For each "WdgMSupervisedEntity" container:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - Find for DeadLineSupervision configured.
 - If "WdgMSupervisedEntity" is referred in configured DeadLineSupervision and WdgMSupervisedEntity.OsCounter is empty.

29) ERR013032: The <WdgMCheckPoint.ShortName> in < WdgMSupervisedEntity.ShortName > is duplicated in wdgMInternalCheckpointInitialRef and wdgMInternalCheckpointFinalRef

- This error occurs:
 - For each " WdgMSupervisedEntity" container:
 - For each " WdgMInternalCheckpointInitialRef" container:
 - For each " WdgMInternalCheckpointFinalRef" container:
 - If check point of WdgMInternalCheckpointInitialRef is same as WdgMInternalCheckpointFinalRef

30) ERR013033: The <WdgMCheckpoint.ShortName> in wdgMInternalCheckpointInitialRef is not in < WdgMSupervisedEntity.ShortName>

- This error occurs:
 - For each " WdgMSupervisedEntity" container:
 - For each " WdgMInternalCheckpointInitialRef" container:
 - For each "WdgMCheckPoint" container in this WdgMSupervisedEntity:
 - If there's no "WdgMCheckPoint" same as "WdgMInternalCheckpointInitialRef"
 - For each " wdgMInternalCheckpointFinalRef" container:
 - For each "WdgMCheckPoint" container in this WdgMSupervisedEntity:
 - If there's no "WdgMCheckPoint" same as "wdgMInternalCheckpointFinalRef"

31) ERR013034: Destination check point in < WdgMInternalTransitionDestRef.ShortName> does not belong to <WdgMSupervisedEntity.ShortName>

- This error occurs:
 - For each " WdgMSupervisedEntity" container:
 - For each "WdgMInternalTransition" container:
 - Get list check points of "WdgMSupervisedEntity"
 - If List of check points contains "WdgMInternalTransitionDestRef" of "WdgMInternalTransition"

32) ERR013034: Source check point in < WdgMInternalTransitionSourceRef.ShortName> does not belong to <WdgMSupervisedEntity.ShortName>

- This error occurs:
 - For each " WdgMSupervisedEntity" container:
 - For each "WdgMInternalTransition" container:
 - Get list check points of "WdgMSupervisedEntity"
 - If List of check points contains "WdgMInternalTransitionSourceRef" of "WdgMInternalTransition"

33) ERR013035: The device ref < WdgMWatchdogDeviceRef.ShortName> in WdgMWatchdog shall not be duplicated

- This error occurs:
 - For each " WdgMWatchdog" container:
 - If "WdgMWatchdogDeviceRef" in current "WdgMWatchdog" is duplicated with "WdgMWatchdogDeviceRef" in other "WdgMWatchdog"

34) ERR013036: InitialMode of <WdgMConfigSet.ShortName> is not in the modes of <WdgMConfigSet.ShortName>

- This error occurs:
 - For each "WdgMConfigSet" container:
 - If "WdgMInitialMode" doesn't refer to "WdgMMode" in this "WdgMConfigSet"

35) ERR013037: The <WdgMSupervisedEntity.ShortName> is duplicated in WdgMLocalStatusSupervisedEntityRef of < WdgMConfigSet.ShortName> \ < WdgMMode.ShortName>

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMLocalStatusParams" container:
 - If referred "WdgMSupervisedEntity" in "WdgMLocalStatusSupervisedEntityRef" of the "WdgMLocalStatusParams" containers is not duplicated

36) ERR013038: The <WdgMCheckpoint.ShortName> in <WdgMAliveSupervision.ShortName> is not in the supervision entities of < WdgMConfigSet.ShortName> \ < WdgMMode.ShortName>

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMAliveSupervision" container:
 - If "WdgMCheckPoint" referred in "WdgMAliveSupervisionCheckpointRef" is not referred by "WdgMLocalStatusSupervisedEntityRef" of all "WdgMLocalStatusParams" containers of current "WdgMMode"

37) ERR013039: The <WdgMTriggerWatchdogRef.ShortName> shall not be duplicated in WdgMTrigger of < WdgMConfigSet.ShortName> \ < WdgMMode.ShortName>

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMTrigger" container:
 - If Watchdog referred by current "WdgMTriggerWatchdogRef" is contained in "WdgMTriggerWatchdogRef" of the other "WdgMTrigger" of current "WdgMMode"

38) ERR013040: The < WdgMCheckpoint.ShortName> is duplicated in WdgMAliveSupervision of < WdgMConfigSet.ShortName> \ < WdgMMode.ShortName>

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMAliveSupervision" container:
 - If "WdgMCheckPoint" referred by current "WdgMAliveSupervisionCheckpointRef" is contained in "WdgMAliveSupervisionCheckpointRef" of the other "WdgMAliveSupervision" of current "WdgMMode"

39) ERR013041: The internal transition <WdgMInternalTransition.ShortName> in <WdgMSupervisedEntity.ShortName> is duplicated

- This error occurs:
 - For each "WdgMSupervisedEntity" container:
 - For each "WdgMInternalTransition" container:
 - If "WdgMInternalTransitionDestRef" of current "WdgMInternalTransition" equals to "WdgMInternalTransitionDestRef" of other "WdgMInternalTransition" containers and
 - "WdgMInternalTransitionSourceRef" of current "WdgMInternalTransition" equals to "WdgMInternalTransitionSourceRef" of other "WdgMInternalTransition" containers

40) ERR013042: The external transition <WdgMExternalTransition.ShortName> in <WdgMConfigSet.ShortName> \ <WdgMMode.ShortName> is duplicated

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMExternalLogicalSuperVision" container:
 - For each "WdgMExternalTransition" container:
 - If "WdgMExternalTransitionDestRef" of current "WdgMExternalTransition" equals to "WdgMExternalTransitionDestRef" of other "WdgMExternalTransition" containers and
 - "WdgMExternalTransitionSourceRef" of current "WdgMExternalTransition" equals to "WdgMExternalTransitionSourceRef" of other "WdgMExternalTransition" containers

41) ERR013043: The deadline supervision <WdgMDeadlineSuperVision.ShortName> in <WdgMConfigSet.ShortName> \ <WdgMMode.ShortName> is duplicated

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMDeadlineSuperVision" container:
 - If "WdgMDeadlineStartRef" of current "WdgMDeadlineSuperVision" equals to "WdgMDeadlineStartRef" of other "WdgMDeadlineSuperVision" containers and
 - "WdgMDeadlineStopRef" of current "WdgMDeadlineSuperVision" equals to "WdgMDeadlineStopRef" of other "WdgMDeadlineSuperVision" containers

42) ERR013045: There's no OS_SPINLOCK mechanism configured for FIRSTEXPIREDSEID_MULTICORE_PROTECTION in multi cores mode.

- This error occurs:
 - In Rte configuration, find for the RteBswModuleInstance/RteBswModuleConfigurationRef/RteBswExclusiveAreaImpl/RteBswExclusiveAreaRef which refer to FIRSTEXPIREDSEID_MULTICORE_PROTECTION.

If RteBswModuleInstance/RteBswModuleConfigurationRef/RteBswExclusiveAreaImpl/RteBswExclusiveAreaRef/RteExclusiveAreaImplMechanism is not "OS_SPINLOCK"

43) ERR013046: Could not get Rte configuration. Exclusive area cannot be validated.

- This error occurs:

Get the Rte's configuration. If configuration is not available (null)

**44) ERR013047: The <WdgMDeadlineStartRef.ShortName> in
<WdgMDeadlineSupervision.ShortName> is not in the supervision entities of
<WdgMConfigSet.ShortName>**

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMDeadlineSupervision" container:
 - If "WdgMCheckpoint" referred by "WdgMDeadlineStartRef" of current "WdgMDeadlineSupervision" is not contained in list of "WdgMCheckpoint" of "WdgMSupervisedEntity" referred by "WdgMLocalStatusSupervisedEntityRef" of any "WdgMLocalStatusParams" containers of current "WdgMMode"

**45) ERR013047: The <WdgMDeadlineStopRef.ShortName> in
<WdgMDeadlineSupervision.ShortName> is not in the supervision entities of
<WdgMConfigSet.ShortName>**

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMDeadlineSupervision" container:
 - If "WdgMCheckpoint" referred by "WdgMDeadlineStopRef" of current "WdgMDeadlineSupervision" is not contained in list of "WdgMCheckpoint" of "WdgMSupervisedEntity" referred by "WdgMLocalStatusSupervisedEntityRef" of any "WdgMLocalStatusParams" containers of current "WdgMMode"

**46) ERR013048: The Init and Final check points
<WdgMExternalCheckpointInitialRef.ShortName> is duplicated in
<Uri(WdgMExternalLogicalSupervision)>**

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMExternalLogicalSupervision" container:
 - If "WdgMCheckpoint" referred by "WdgMExternalCheckpointInitialRef" equals to "WdgMExternalCheckpointFinalRef"

**47) ERR013049: The initial checkpoint <WdgMExternalCheckpointInitialRef.ShortName> in
<WdgMExternalLogicalSupervision.ShortName> is not in the supervision entities of <
WdgMConfigSet.ShortName>\< WdgMMode.ShortName>**

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMExternalLogicalSupervision" container:
 - If "WdgMCheckpoint" referred by "WdgMExternalCheckpointInitialRef" of current "WdgMExternalLogicalSupervision" is not contained in list of "WdgMCheckpoint" of "WdgMSupervisedEntity" referred by "WdgMLocalStatusSupervisedEntityRef" of any "WdgMLocalStatusParams" containers of current "WdgMMode"

**48) ERR013049: The final checkpoint < WdgMExternalCheckpointFinalRef.ShortName> in
<WdgMExternalLogicalSupervision.ShortName> is not in the supervision entities of <
WdgMConfigSet.ShortName>\< WdgMMode.ShortName>**

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMExternalLogicalSupervision" container:
 - If "WdgMCheckpoint" referred by
"WdgMExternalCheckpointFinalRef" of current "
WdgMExternalLogicalSupervision " is not contained in list of
"WdgMCheckpoint" of "WdgMSupervisedEntity" referred by
"WdgMLocalStatusSupervisedEntityRef" of any
"WdgMLocalStatusParams" containers of current "WdgMMode"

49) ERR013049: The destination checkpoint < WdgMExternalTransitionDestRef.ShortName> in < WdgMExternalTransition.ShortName> is not in the supervision entities of < WdgMConfigSet.ShortName>\< WdgMMode.ShortName>\<WdgMExternalLogicalSupervision.ShortName>

- This error occurs:
 - For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMExternalLogicalSupervision" container:
 - For each "WdgMExternalTransition" container:
 - If "WdgMCheckpoint" referred by
"WdgMExternalTransitionDestRef" of "WdgMExternalTransition"
of current " WdgMExternalLogicalSupervision " is not contained
in list of "WdgMCheckpoint" of "WdgMSupervisedEntity" referred
by "WdgMLocalStatusSupervisedEntityRef" of any
"WdgMLocalStatusParams" containers of current "WdgMMode"

50) ERR013049: The source checkpoint < WdgMExternalTransitionSourceRef.ShortName> in < WdgMExternalTransition.ShortName> is not in the supervision entities of < WdgMConfigSet.ShortName>\< WdgMMode.ShortName>\<WdgMExternalLogicalSupervision.ShortName>

- This error occurs:
- For each "WdgMConfigSet" container:
 - For each "WdgMMode" container:
 - For each "WdgMExternalLogicalSupervision" container:
 - For each "WdgMExternalTransition" container:
 - If "WdgMCheckpoint" referred by "WdgMExternalTransitionSourceRef" of
"WdgMExternalTransition" of current " WdgMExternalLogicalSupervision " is not contained in list of
"WdgMCheckpoint" of "WdgMSupervisedEntity" referred by
"WdgMLocalStatusSupervisedEntityRef" of any "WdgMLocalStatusParams" containers of current
"WdgMMode"

51) ERR013050: The value configured for parameter MODULE-ID in container BSW-MODULE-DESCRIPTION in provided MDT file is not correct. Module ID of WdgM must be 13.

If value of ModuleId in file BSWMDT is not equal with the ModuleId of WdgM

52) ERR013051: The value configured for parameter VENDOR-ID in container BSW-IMPLEMENTATION in provided MDT file is not correct. Vendor ID of WdgM must be 76.

If value of VendorId in file BSWMDT is not equal with the VendorId of WdgM

53) ERR013052: The parameter <Parameter Name> in the container <Container Name> should be configured.

If any of the mandatory configuration parameters mentioned below is not configured in ECU Configuration Description File. (Refer to the list in Table 013005)

Container Name	Parameter Name
BSW-IMPLEMENTATION	AR-RELEASE-VERSION
	VENDOR-ID
	SW-VERSION
BSW-MODULE-DESCRIPTION	MODULE-ID

54) ERR013053: The value configured parameter <Parameter Name> in the container <Container Name> is incorrect. It should be 1.0.0 for example.

If the parameters 'Parameter Name' is not configured as per the pattern.
(Refer to the list in Table 013006)

Table 013006

Parameter Name	Container Name
SW-VERSION	BSW-IMPLEMENTATION

55) ERR013054: AUTOSAR RELEASE VERSION <Version> configured for the parameter <AR-RELEASE-VERSION> in provided MDT file is not correct. AUTOSAR RELEASE VERSION should be 4.4.0.

If the value of the element AR-RELEASE-VERSION present in file BSWMDT is configured other than 4.4.0.

7.1.2 Warning Messages

1) WAR013022: The off mode is not allowed in '<WdgMMode.ShortName>/<WdgMTrigger.ShortName>'.

- This error occurs:

if the Reference path configured for the parameter 'WdgMWatchdogDeviceRef' in the container 'WdgMWatchdog' is not unique.

7.1.3 Information Messages

None

8. SWP Error Code

8.1 SWP Error Code List

8.1.1 WDG_M_E_SUPERVISION

ErrorId Symbol	WDGM_E_SUPERVISION
Description	Occurs when: 1. WdgM_Init execution fails to change WdgMMode of Wdg driver. 2. WdgM_SetMode execution fails to change WdgMMode of Wdg driver. 3. Supervision has failed (Global Supervision Status has reached WDGM_GLOBAL_STATUS_STOPPED) and a watchdog reset will occur
Cause of the problem	ASW
Platform default Action	RESET
Functional impact	When the change of WdgM Mode fails, SetTriggerCondition value is sent to Wdg as 0. Wdg triggering stops when SetTriggerCondition value is 0, resulting in Watchdog Reset..
Other module association	None
MCU	Common
Types of questions	Settings, code
Application applicable measures	1. Need to review HW register with reference to below 1.1. Failed to change WdgMMode of Wdg driver in executing WdgM_Init: It occurs when the mode change (register change) of HW Wdg in the Wdg_SetMode function of MCAL Wdg driver fails. 1.2. Failed to change WdgMMode of Wdg driver in executing WdgM_SetMode: It occurs when the mode change (register change) of HW Wdg in the Wdg_SetMode function of MCAL Wdg driver fails. 2. ASW operation review: When a Dem Error occurs during Runtime, reset occurs, so it is necessary to perform relevant tests.

8.1.2 WDG_M_E_SET_MODE

ErrorId Symbol	WDGM_E_SET_MODE
Description	Occurs when: 1. WdgM_Init execution fails to change WdgMMode of Wdg driver. 2. WdgM_SetMode execution fails to change WdgMMode of Wdg driver.
Cause of the problem	ASW
Platform default Action	RESET
Functional impact	When the change of WdgM Mode fails, SetTriggerCondition value is sent to Wdg as 0. Wdg triggering stops when SetTriggerCondition value is 0, resulting in Watchdog Reset..
Other module association	None
MCU	Common
Types of questions	Settings, code
Application applicable measures	1. Need to review HW register with reference to below 1.1. Failed to change WdgMMode of Wdg driver in executing WdgM_Init: It occurs when the mode change (register change) of HW Wdg in the Wdg_SetMode function of MCAL Wdg driver fails. 1.2. Failed to change WdgMMode of Wdg driver in executing WdgM_SetMode: It occurs when the mode change (register change) of HW Wdg in the Wdg_SetMode function of MCAL Wdg driver fails. 2. ASW operation review: When a Dem Error occurs during Runtime, reset occurs, so it is necessary to perform relevant tests.

9. Appendix

9.1 Integration Additional Module

9.1.1 Os module

9.1.1.1 OsTask Container

9.1.1.1.1 OsTask setting

OsTask configuration for WdgM module

Parameter Name	Value	Category
Short Name	OsTask_WdgM	F
Activation	1	F
Priority	Use settings when deploying the platform	F
Schedule	FULL	F

Parameter Name	Value	Category
Short Name	OsTask_WdgMAPI	F
Activation	1	F
Priority	Use settings when deploying the platform	F
Schedule	FULL	F

9.1.1.2 OsAlarm Container

9.1.1.2.1 OsAlarm setting

OsAlarm configuration for WdgM module

Parameter Name	Value	Category
Short Name	OsAlarm_WdgM	F
Accessing Application	Use settings when deploying the platform	F
Counter Ref	OsCounter reference from OsApplication where WdgM is located	F

9.1.1.2.2 OsAlarmAction setting

Parameter Name	Value	Category
Short Name	Use settings when deploying the platform	F
Ref	OsTask_WdgM	F

9.1.1.3 OsApplication Container

9.1.1.3.1 OsApplication setting

OsTask and OsAlarm for WdgM module added to OsApplication of Master core

Parameter Name	Value	Category
Short Name	Use settings when deploying the platform	F
Core Assignment	0	F
Trusted	true	F
App Alarm Ref	OsAlarm_WdgM added	F
App Counter Ref	Use settings when deploying the platform	F
App Isr Ref	Use settings when deploying the platform	F
App Task Ref	OsTask_WdgM added	F

9.1.2 Rte module

9.1.2.1 RteBswModuleInstance Container

Rte setting of BSW module WdgM

9.1.2.1.1 BswInstance_WdgM setting

1) RteBswModuleInstance setting

Parameter Name	Value	Category
Short Name	BswInstance_WdgM	F
Bsw Implementation Ref	BswImplementation_WdgM	F
Bsw Module Configuration Ref	WdgM	F

2) RteBswEventToTaskMapping setting

Task mapping and period setting of WdgM_MainFunction

Period of WdgM_MainFunction: See Period of / Bsw_WdgM / WdgM / BswInternalBehavior_WdgM / TE_WdgM

Parameter Name	Value	Category
Short Name	RteBswEventToTaskMapping_TE_WdgM	F
Bsw Immediate Restart	false	F
Bsw Position In Task	0	F
Bsw Event Ref	TE_WdgM	F
Bsw Mapped To Task Ref	OsTask_BSW_10ms	F
Bsw Used Os Alarm Ref	OsAlarm_BSW_10ms	F

3) RteBswExclusiveAreaImpl setting

Parameter Name	Value	Category
Short Name	GLOBALSUPVSNSTATUS_PROTECTION	F
Exclusive Area Impl Mechanism	ALL_INTERRUPT_BLOCKING	F
Bsw Exclusive Area Ref	GLOBALSUPVSNSTATUS_PROTECTION	F

Parameter Name	Value	Category
Short Name	FIRSTEXPIREDSEID_MULTICORE_PROTECTION	F
Exclusive Area Impl Mechanism	OS_SPINLOCK	F
Bsw Exclusive Area Ref	FIRSTEXPIREDSEID_MULTICORE_PROTECTION	F

4) Bsw Required Mode Group Connection setting

Parameter Name	Value	Category
Short Name	RteBswRequiredModeGroupConnection0	F
Bsw Required Mode Group Ref	modeNotificationPort_EcuMode	F
Bsw Provided Mode Grp Mod Inst Ref	BswInstance_BswM	F

9.1.2.2 RteSwComponentInstance Container

9.1.2.2.1 SwcInstance_WdgM setting

Setting ServiceSwComponent Instance of WdgM module

1) RteSwComponentInstance setting

Parameter Name	Value	Category
Short Name	SwcInstance_WdgM	F
Software Component Instance Ref	Extracted Svc_WdgM reference	F

2) RteEventToTaskMapping setting

Task mapping and period setting of WdgM_SetMode, WdgM_PerformReset

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_OIE_SetMode	F
Bsw Position In Task	0	F
Bsw Event Ref	OperationInvokedEvent_SetMode	F
Bsw Mapped To Task Ref	OsTask_WdgMAPI	F

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_OIE_PerformReset	F
Bsw Position In Task	1	F
Bsw Event Ref	OperationInvokedEvent_PerformReset	F
Bsw Mapped To Task Ref	OsTask_WdgMAPI	F

9.1.2.2.2 SwcInstance_WdgMTest setting

Setting ServiceSwComponent Instance of WdgM module

1) RteSwComponentInstance setting

Parameter Name	Value	Category
Short Name	SwcInstance_WdgMTest	F
Software Component Instance	Extracted Swc_WdgMTest reference	F

Parameter Name	Value	Category
Ref		

2) RteEventToTaskMapping setting

Task mapping and cycle setting of RE_WdgMTest

Period of RE_WdgMTest: / Period of / App_WdgM / Swc_WdgMTest / IB_WdgMTest / TE_WdgMTest

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_TE_WdgMTest	F
Bsw Immediate Restart	false	F
Bsw Position In Task	0	F
Bsw Event Ref	TE_WdgMTest	F
Bsw Mapped To Task Ref	OsTask_ASW_10ms	F
Bsw Used Os Alarm Ref	OsAlarm_ASW_10ms	F

Task mapping of Global Supervision Status status change event

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_MSE_global Mode_OK	F
Bsw Position In Task	0	F
Bsw Event Ref	SwcModeSwitchEvent_globalMode_O K	F
Bsw Mapped To Task Ref	OsTask_WdgMTestMode	F

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_MSE_global Mode_FAILED	F
Bsw Position In Task	1	F
Bsw Event Ref	SwcModeSwitchEvent_globalMode_F AILED	F
Bsw Mapped To Task Ref	OsTask_WdgMTestMode	F

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_MSE_global Mode_EXPIRED	F
Bsw Position In Task	2	F
Bsw Event Ref	SwcModeSwitchEvent_globalMode_E XPIRED	F
Bsw Mapped To Task Ref	OsTask_WdgMTestMode	F

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_MSE_global Mode_STOPPED	F
Bsw Position In Task	3	F
Bsw Event Ref	SwcModeSwitchEvent_globalMode_S TOPPED	F
Bsw Mapped To Task Ref	OsTask_WdgMTestMode	F

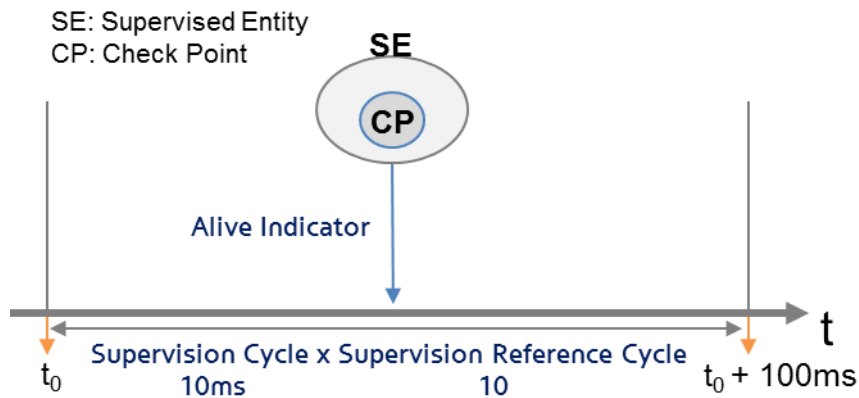
9.2 Application User Setting Guide

Adding Alive / Deadline / Logical Supervision to WdgM requires a design that fits the purpose

9.2.1 Alive Supervision

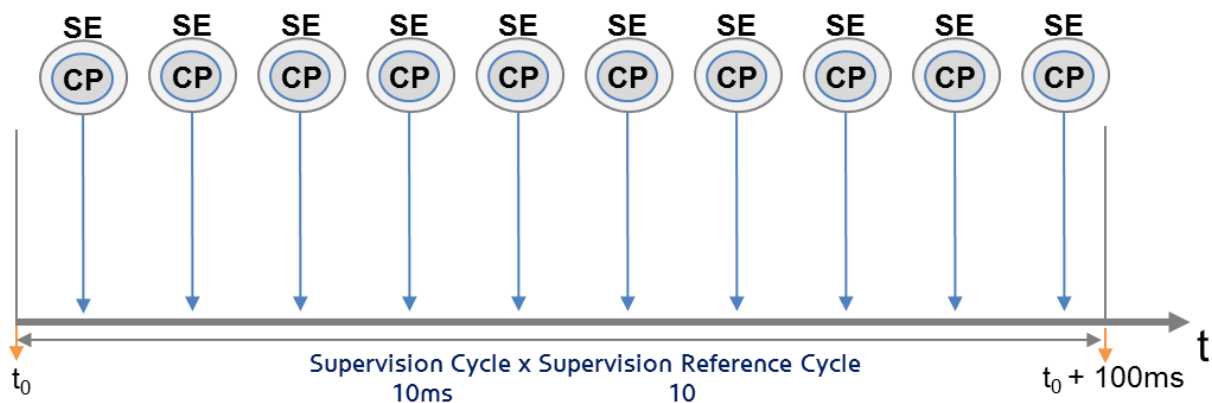
Alive Supervision defines one checkpoint in the target to be monitored (Supervised Entity) and defines the periodic execution time based on the Supervision Cycle (WdgM_MainFunction cycle).

The basis of the monitoring cycle is WdgMMode's Supervision Cycle, and the monitoring cycle is defined as a multiple of this value (Supervision Reference Cycle).



The number of monitoring points performed during the monitoring cycle is defined as Expected Alive Indications. The additional allowable range of execution frequency can be defined as Min / Max Margin.

Supervision Cycle: 0.01s Min Margin: 1
Expected Alive Indications: 10 Max Margin: 1
Supervision Reference Cycle: 10



The setting of the figure above is a condition that the indicator should occur 10 times during the monitoring period 100 ms (periodic execution time 10 ms X times 10 times). Since Min / Max Margin is 1 each, it is a permitted condition even if an indicator occurs 9 to 11 times.

The Min / Max Margin parameter is a value that can be set by the user considering the condition that the monitoring target can be executed quickly or slowly during the monitoring period depending on the execution timing of the application.

Refer to the example below for an example of setting.

Ex1)

Supervision Cycle(Cycle of WdgM_MainFunction) : 0.01 (sec)

Supervision Reference Cycle : 10 (0.1 sec)

Expected Alive Indications : 10

Min Margin : 1

Max Margin : 1

➔ CheckpointReached function that takes Checkpoint ID set in Checkpoint Ref as an argument for every 100ms period should be called 9 (10-1) ~ 11 (10 + 1) times.

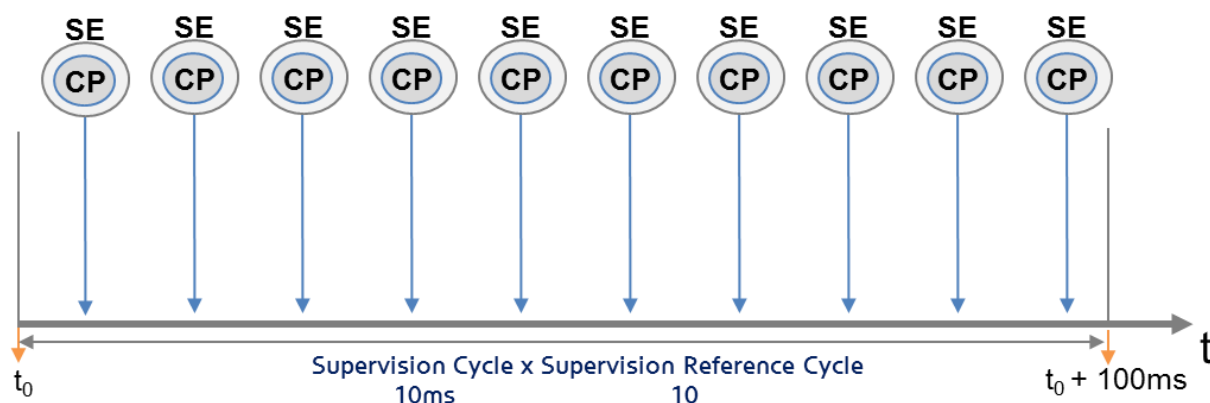
Supervision Cycle: 0.01s

Min Margin: 1

Expected Alive Indications: 10

Max Margin: 1

Supervision Reference Cycle: 10



Ex2)

Supervision Cycle(Cycle of WdgM_MainFunction) : 0.01 (sec)

Supervision Reference Cycle : 2 (0.02 sec)

Expected Alive Indications : 1

Min Margin : 0

Max Margin : 0

➔ Check CheckpointReached function that takes Checkpoint ID set in Checkpoint Ref as an argument should be called once every 20ms period.

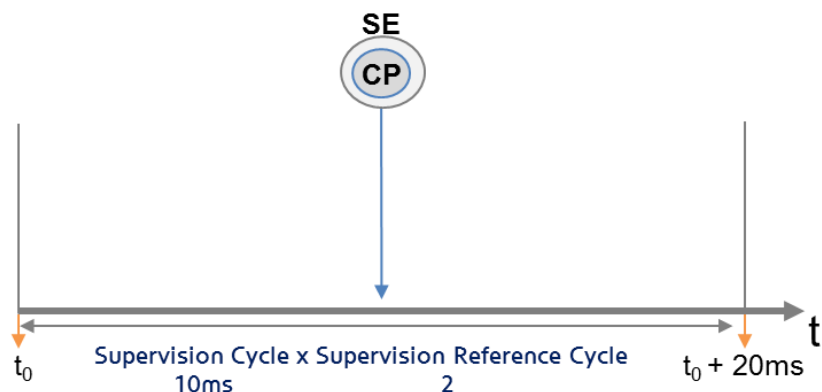
Supervision Cycle: 0.01s

Min Margin: 0

Expected Alive Indications: 1

Max Margin: 0

Supervision Reference Cycle: 2



Ex3)

Supervision Cycle(Cycle of WdgM_MainFunction) : 0.01 (sec)

Supervision Reference Cycle : 1 (0.01 sec)

Expected Alive Indications : 2

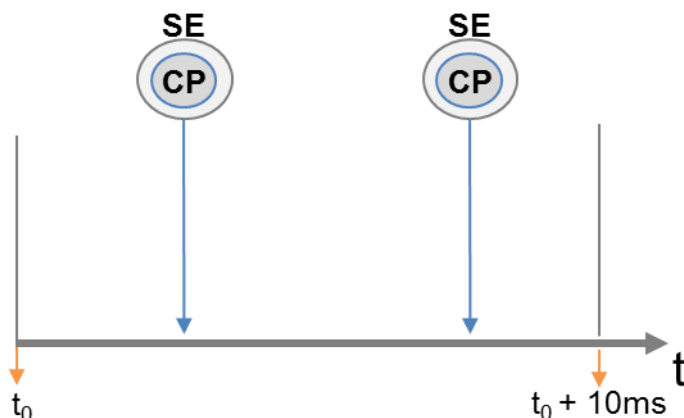
Min Margin : 0

Max Margin : 1

- ➔ CheckpointReached function with Checkpoint ID set in Checkpoint Ref should be called 2 ~ 3 (2 + 1) times every 10ms period.

Supervision Cycle: 0.01s
Expected Alive Indications: 2
Supervision Reference Cycle: 1

Min Margin: 0
Max Margin: 1



Ex4)

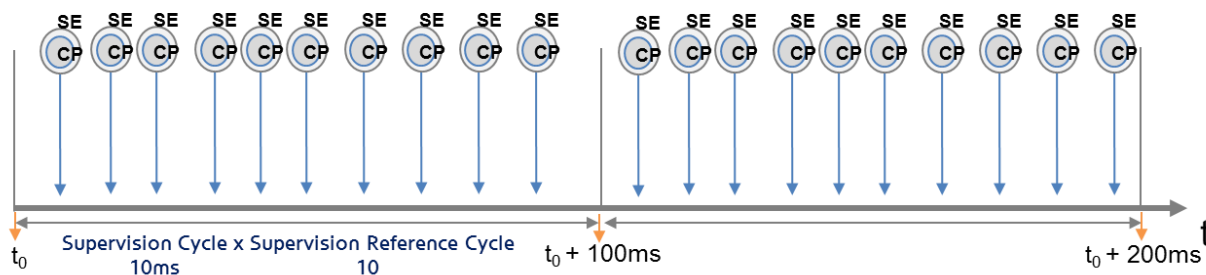
Supervision Cycle(Cycle of WdgM_MainFunction) : 0.1 (sec)
 * Expired Supervision Cycle Tol: 10 (* 5.2.2 Wdg Mode Description Reference)
 Supervision Reference Cycle : 10
 Expected Alive Indications : 10
 Min Margin : 0
 Max Margin : 0

- ➔ CheckpointReached function that takes Checkpoint ID set in Checkpoint Ref as an argument should be called 10 times every 100ms period. In case of failure, after 1s, status information is transmitted to the Wdg driver.

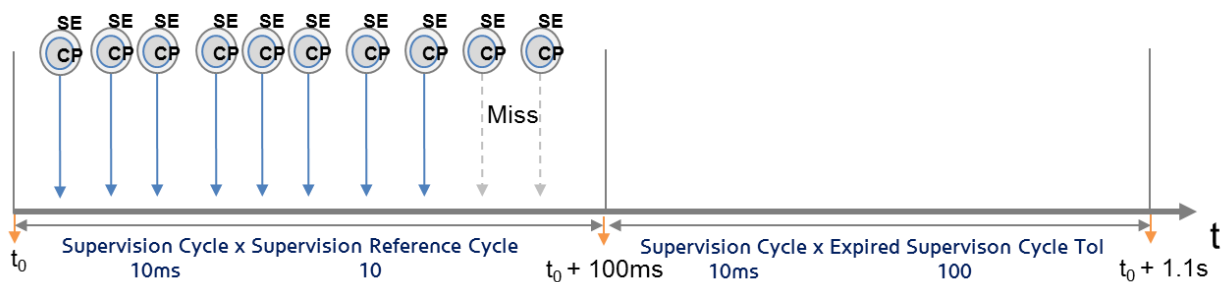
Supervision Cycle: 0.01s
Expected Alive Indications: 10
Supervision Reference Cycle: 10
Expired Supervision Cycle Tol: 100

Min Margin: 0
Max Margin: 0

<Normal Condition>

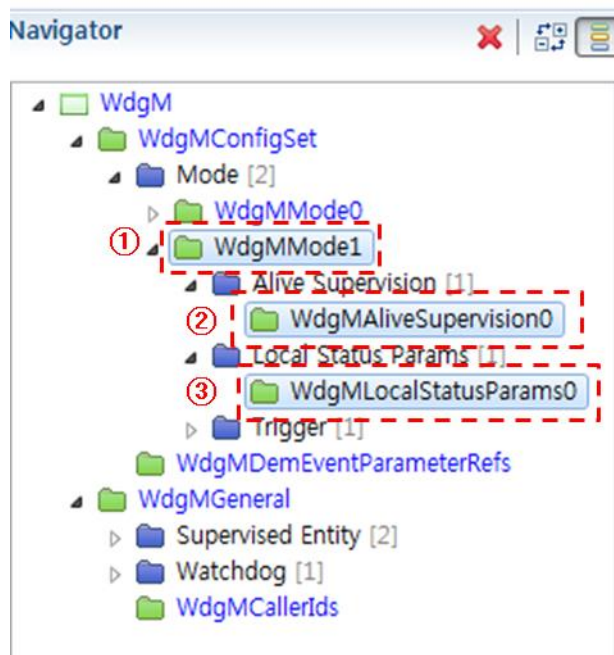


<Reset Condition>










In the above example, the properties set in the corresponding Watchdog Mode are set in the properties of the WdgM Mode and Alive Supervision Container, and the properties of the Local Status Params Container.

(* Refer to 5.2.5.2 WdgMLocalStatusParams)



Container Details - WdgMMode
Config Mode

Short Name*:		WdgMModel
Expired Supervision Cycle Tol*:		2 Ex4,
Id*:		1
Supervision Cycle*:		0.010







	Alive Supervision	1	[0...65535]
	Local Status Params	1	[0...65535]
	Trigger	1	[0...255]

▼ To Be Configured:

-  [Deadline Supervision](#) [0...65535]
-  [External Logical Supervision](#) [0...65535]




WdgM Alive Supervision Container

Container Details - WdgMAliveSupervision

Short Name*:		WdgMAliveSupervision0
Expected Alive Indications*:		10
Max Margin*:		1
Min Margin*:		1
Supervision Reference Cycle*:		10
Checkpoint Ref*:		CP_WdgMTest_AS0 [/AUTOSAR/WdgM/Wc

WdgM Local Status Parameters Setting

Container Details - WdgMLocalStatusParams

Short Name*:		WdgMLocalStatusParams0
Failed Alive Supervision Ref Cycle Tol*:		1
Local Status Supervised Entity Ref*:		SE_WdgMTest [/AUTOSAR/WdgM/WdgMGen

9.2.2 Deadline Supervision

Deadline Supervision defines two checkpoints to be monitored and sets the range of execution time between two points (minimum execution time \leq actual execution time \leq maximum execution time). Defined content is set in the properties of Deadline Supervision of WdgMMode and Local Status Params. Refer to the example below for an example of setting.

Ex1)

Deadline Min: 0.0

Deadline Max: 0.025

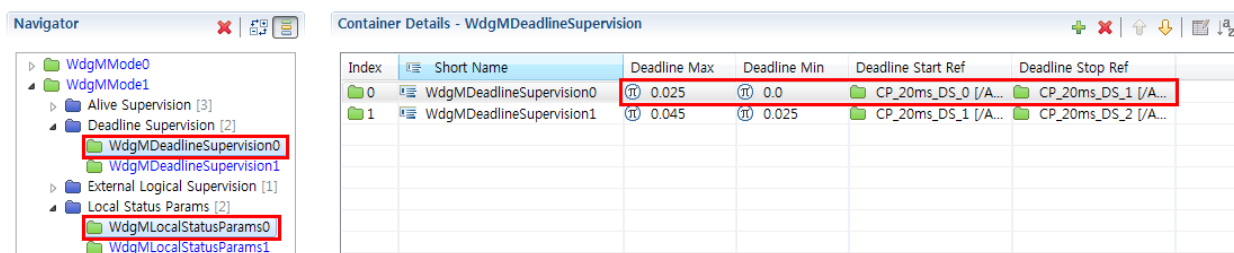
- After CheckpointReached function with Checkpoint ID set in Deadline Start Ref is called, CheckpointReached function with Checkpoint ID set in Deadline Stop Ref should be called within 25ms.

Ex2)

Deadline Min: 0.025

Deadline Max: 0.045

- After CheckpointReached function with Checkpoint ID set in Deadline Start Ref is called, CheckpointReached function with Checkpoint ID set in Deadline Stop Ref should be called within 25ms and within 45ms.

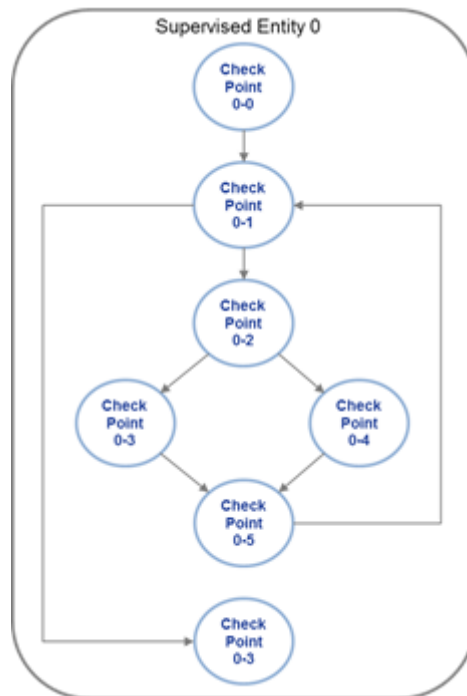


Index	Short Name	Deadline Max	Deadline Min	Deadline Start Ref	Deadline Stop Ref
0	WdgMDeadlineSupervision0	0.025	0.0	CP_20ms_DS_0 [/A...	CP_20ms_DS_1 [/A...
1	WdgMDeadlineSupervision1	0.045	0.025	CP_20ms_DS_1 [/A...	CP_20ms_DS_2 [/A...

9.2.3 Logical Supervision

Logical Supervision is a basic technique for verifying the correct execution of embedded system software, and it is necessary to check the details of the safety standards (ISO26262 or IEC61508) that Logical Supervision requires.

Logical Supervision focuses on the Control Flow Error branching from a valid (i.e. coded / compiled) program sequence while the application is running error-free. Incorrect control flow causes one or more program instructions to be processed or not executed in the wrong order. Control flow errors can cause data corruption, microcontroller resets, fail-silence violations, etc. [1] AUTOSAR_SWS_WatchdogManager

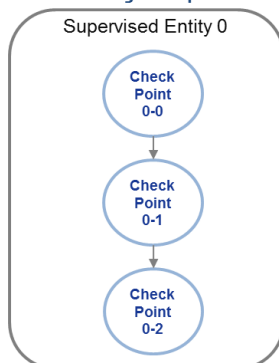


In conclusion, Logical Supervision defines the transition between the check points of the supervised entity, and monitors the execution order of the connection..

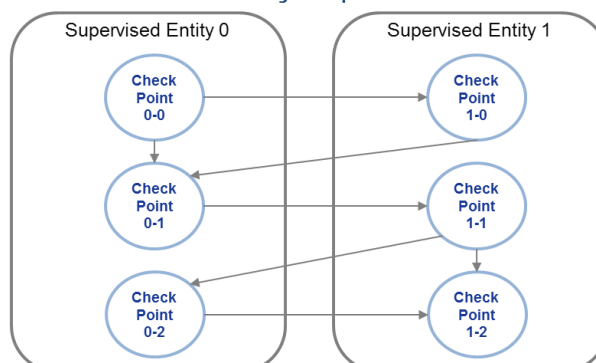
The setting method of Logical Supervision differs depending on whether the monitoring point to be monitored exists only in one supervised entity or in two or more supervised entities..

Internal Logical Supervision monitors the control flow of the monitoring point in one monitoring target. External Logical Supervision monitors the control flow between two or more monitoring targets.

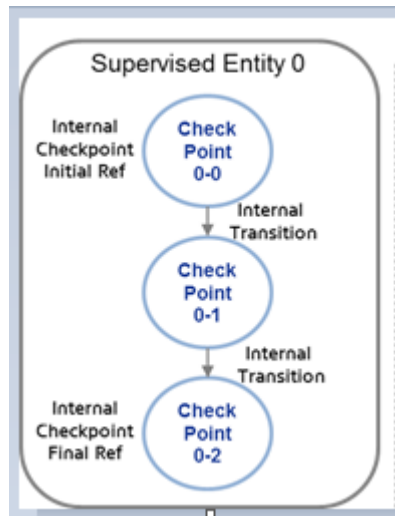
Internal Logical Supervision



External Logical Supervision

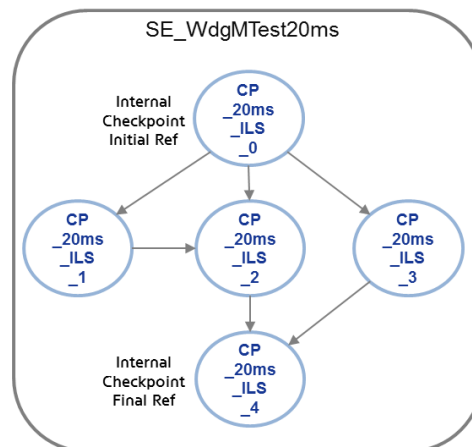


The figure below shows the case where a checkpoint belongs to only one watched object. The execution sequence to be monitored is shown in the graph below, and Internal Checkpoint Initial Ref, Internal Checkpoint Final Ref, and Internal Transitions are defined. Set the defined content in the properties of the relevant Supervised Entity.



[Internal Logical Supervision setting example]

1) Control Flow Graph



2) WdgM Configuration



Monitoring target setting

Container Details - WdgMSupervisedEntity

Short Name*:

Id*:

Internal Checkpoint Initial Ref*:

Internal Checkpoint Final Ref*:

Os Counter:

☐ Checkpoint 13 [1...65535]

☐ Internal Transition 6 [0...65535]

Monitoring target setting - All monitoring point settings in Control Flow Graph are required

Container Details - WdgMCheckpoint

Short Name*:

Id*:

...

Container Details - WdgMCheckpoint

Short Name*:

Id*:

Monitoring point connection setting – Connection setting between points set in Control Flow Graph

Container Details - WdgMInternalTransition

Short Name*: WdgMInternalTransition0

Dest Ref*: CP_20ms_ILS_1 [/AUTOSAR/WdgM/WdgMGeneral/SE_WdgMTes

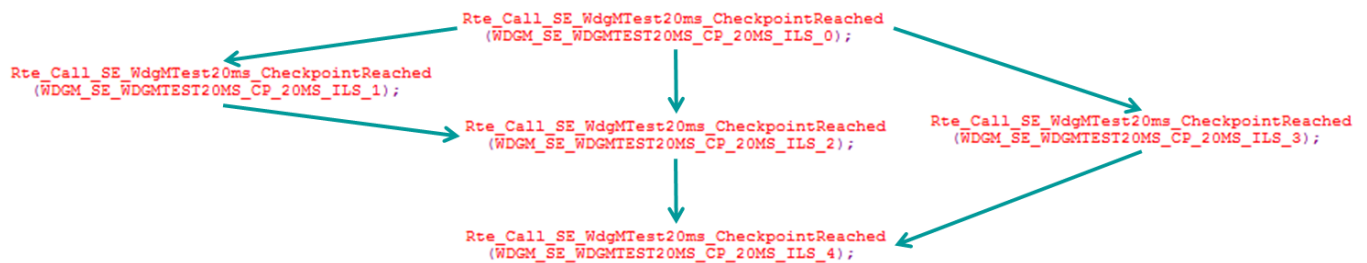
Source Ref*: CP_20ms_ILS_0 [/AUTOSAR/WdgM/WdgMGeneral/SE_WdgMTes

Container Details - WdgMInternalTransition

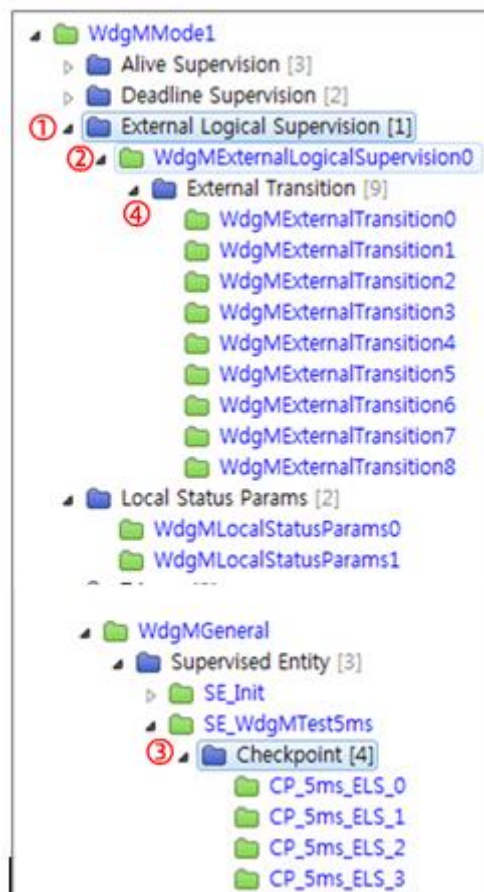
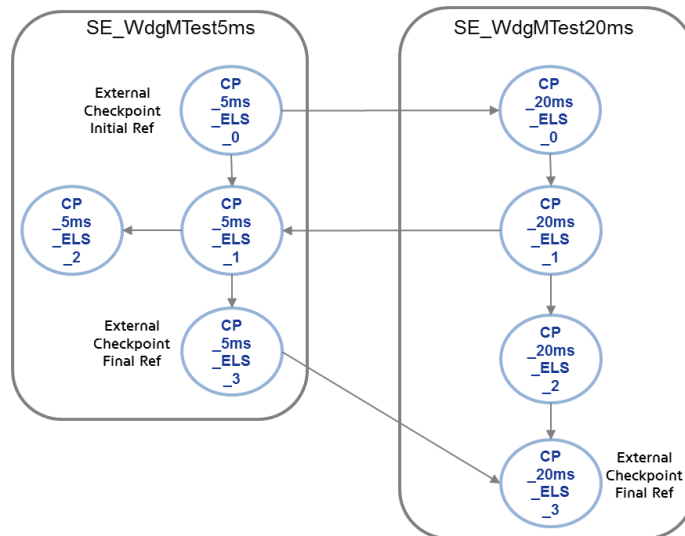
Index	Short Name	Dest Ref	Source Ref
0	WdgMInternalTransition0	CP_20ms_ILS_1 [/A...	CP_20ms_ILS_0 [/AUTOSAR/WdgM/
1	WdgMInternalTransition1	CP_20ms_ILS_2 [/A...	CP_20ms_ILS_0 [/AUTOSAR/WdgM/
2	WdgMInternalTransition2	CP_20ms_ILS_3 [/A...	CP_20ms_ILS_0 [/AUTOSAR/WdgM/
3	WdgMInternalTransition3	CP_20ms_ILS_2 [/A...	CP_20ms_ILS_1 [/AUTOSAR/WdgM/
4	WdgMInternalTransition4	CP_20ms_ILS_4 [/A...	CP_20ms_ILS_2 [/AUTOSAR/WdgM/
5	WdgMInternalTransition5	CP_20ms_ILS_4 [/A...	CP_20ms_ILS_3 [/AUTOSAR/WdgM/

3) CheckpointReached API Call

In the actual code, checkpointReached is called with the argument of the checkpoint ID at the designed checkpoint location as shown below.



If the checkpoint to be monitored exists over two or more monitoring targets, it is represented as a graph as below and defines External Checkpoint Initial Ref, External Checkpoint Final Ref, and External Transitions. Set the defined content in the properties of External Logical Supervision of WdgMMode and Local Status Params.



In the actual code, checkpointReached is called with the argument of checkpoint ID at the designed checkpoint location.

9.3 Notes on design

9.3.1 Alive Supervision

None

9.3.2 Deadline Supervision

- 1) When a Checkpoint Reach Event that belongs to Deadline Supervision setting occurs
 - A. Ignore if the end checkpoint is reached without the start checkpoint being reached
 - B. If the same start checkpoint is reached while the start checkpoint has already been reached, use the checkpoint arrival event timestamp that was reached later.

9.3.3 Logical Supervision

- 1) When a Checkpoint Reach Event that belongs to the Internal / External Graph occurs
 - A. If the start checkpoint has not been reached and another checkpoint in the graph (middle / end) is reached, failure is determined
- 2) If the Transition of Logical Supervision increases, the execution time of CheckpointReached API may increase.

9.3.4 Mode Switch Event

- 1) Change of Local Supervision Status or Global Supervision Status of Supervised Entity can be transmitted through Mode switch event.
 - A. Multiple activation problems may occur when a mode switch event and a mapped task are used in duplicate.

9.3.5 WdgM_DeInit

- 1) Wdg_SetTriggerCondition is not called after WdgM_DeInit. Therefore, it is necessary to set a sufficient timeout value so that the platform does not intentionally reset by Wdg.
- 2) Timeout value should be set within 4s maximum considering NVM Write All time.

9.3.6 Std_ReturnType

- 1) In case of using Std_Return among WdgM API, N_OK value may be returned, so it should be considered sufficiently in design.

9.3.7 Low Power Mode

- 1) During Low Power

- A. WdgM module operation is stopped while the ECU is in the sleep state. Therefore, if the hardware watchdog continues to operate in the sleep state, watchdog triggering must be performed periodically within the hardware watchdog timeout. In the sleep state, watchdog triggering needs to reflect design by referring to EcuM User Manual.
- B. When wakeup reset occurs according to the MCU, it is necessary to include the startup time in the watchdog timeout. Therefore, the watchdog timeout time should be set sufficiently considering the startup time.
- C. The watchdog triggering cycle affects the current consumption. If the watchdog timeout is too short, the current consumption increases because the watchdog triggering must be performed at short intervals. Considering this, the watchdog timeout time and watchdog triggering cycle must be set.

9.3.8 Condition Value

- 1) Condition Value should be larger than the timeout value of Fast mode / Slow mode according to the operation mode of Wdg.

- A. When the Wdg mode is the Fast mode, set a value greater than the timeout in the Wdg module's fast mode as the Condition Value as shown below.

Index	Short Name	Condition Value	Watchdog Mode	Watchdog Ref
0	WdgMTrigger_Int	100	WDGIF_FAST_MODE	WdgMWatchdog_In...

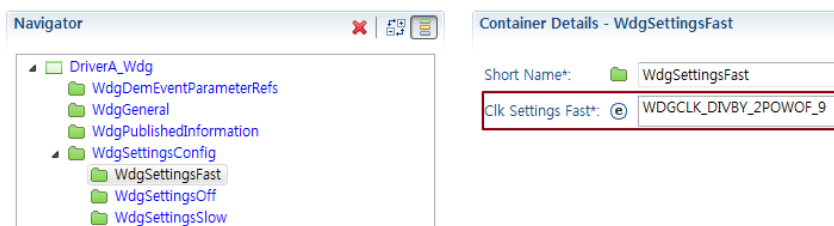
Container Details - WdgSettingsFast	
Short Name*:	WdgSettingsFast
Fast Mode Timeout Value*:	0.01

- B. When the Wdg mode is Slow mode, set a value greater than the timeout in the Slow mode of the Wdg module as the Condition Value as shown below.

Index	Short Name	Condition Value	Watchdog Mode	Watchdog Ref
0	WdgMTrigger_Int	500	WDGIF_SLOW_MODE	WdgMWatchdog_In...

Container Details - WdgSettingsSlow	
Short Name*:	WdgSettingsSlow
Slow Mode Timeout Value*:	0.02

- C. For some MCUs, set the timeout value of Wdg to a value other than a time unit. For example, the RH850 can set the timeout value through the settings in the figure below, and to calculate the timeout value in units of hours, it can be calculated through the following equation.



$$\text{Timeout (ms)} = \frac{2^n \text{McuAwoWdta}}{1000} * 1000 \text{ (WDGCLK_DIVBY_2POWOF}_n \text{ where } n = 9 \text{ to } 16)$$

Therefore, the condition value is set after calculating the timeout of Wdg by referring to the relevant MCU user manual.

9.4 Add workaround for Initialization function in multicore

This workaround is needed because the module initialization has not supported by EcuM. The workaround needs to update the integration code in EcuM_Boot.c for WdgM and Wdg driver Init function as bellow example:

WdgM and Wdg driver Init is could be performed in TASK (OsTask_BSW_Init_Subx) of EcuM_boot.c file. User needs to update the call to Init functions according to the number of configured WdgMConfigSet containers. In case of single core mode (one configured WdgMConfigSet) this workaround is not need.

Example code:

```
#if (ECUM_NO_OF_CORES > 1)
TASK(OsTask_BSW_Init_Sub1)
{
    EcuM_StartupTwo();
    Wdg_17_Scu_Init(&Wdg_17_Scu_Config_1);
    WdgM_Init(NULL_PTR);
    (void)WaitEvent(OsEvent_RteStart_Sub1);
    (void)ClearEvent(OsEvent_RteStart_Sub1);

    Rte_Start();

    (void)TerminateTask();
}
#endif

#if (ECUM_NO_OF_CORES > 2)
TASK(OsTask_BSW_Init_Sub2)
{
    EcuM_StartupTwo();
    Wdg_17_Scu_Init(&Wdg_17_Scu_Config_2);
    WdgM_Init(NULL_PTR);
    (void)WaitEvent(OsEvent_RteStart_Sub2);
    (void)ClearEvent(OsEvent_RteStart_Sub2);
    Rte_Start();

    (void)TerminateTask();
}
#endif

#if (ECUM_NO_OF_CORES > 3)
TASK(OsTask_BSW_Init_Sub3)
{
```

```
EcuM_StartupTwo();  
Wdg_17_Scu_Init(&Wdg_17_Scu_Config_3);  
WdgM_Init(NULL_PTR);  
(void)WaitEvent(OsEvent_RteStart_Sub3);  
(void)ClearEvent(OsEvent_RteStart_Sub3);  
  
Rte_Start();  
  
(void)TerminateTask();  
}  
#endif
```