HYUNDAI AUTOEVER

AUTOSAR WdgM User Manual

DOC. NO

SCOPE OF APPLICATION All Project/Engineering
Responsibility : Classic AUTOSAR Team

File Name WdgM_UM.docx
Creation HyeonSeok Park
Check HoiMin Kim
Approval InWon Kang
Edition Date: 2023/07/17
Document Management System

HYUNDAI AutoEver

Document Change Histroy

Date (YYYY-MM-DD)	Ver.	Editor	Chap	Description (before -> after revision)
2016-04-06	1.0.0	Kuksun Joo	All	• WdgM User Manual Initial Creation
2016-05-27	1.1.0	Kuksun Joo	All	User manual review results applied
2016-07-29	1.1.1	Kuksun Joo	8.2.3	• Logical Supervision details added
2016-08-11	1.1.2	Kuksun Joo	8.1.2	• RTE BSW Module Instance configuration added
2016-08-25	1.1.3	Kuksun Joo	5.2	• WdgM_DeInit Timeout default value descriptions added
2016-09-06	1.1.4	Kuksun Joo	4.3 8.3.3	• WdgM 1.2.11 revision details added • Cautions about design when using logical supervision added
2016-09-26	1.1.5	Kuksun Joo	4.3 8.1.2.1	• Change Log new format applied • RTE BSW Module Instance unnecessary configuration removed
2016-11-10	1.1.6	Kuksun Joo	8.4.3	• Details of using RH850 Option Byte added
2018-04-18	1.1.7	YongHyun Han	4.3 8	• WdgM 1.2.12 revision details added • SWP Error Code added
2018-12-11	1.1.8	YongHyun Han	4.3 4.4 9.3.7	• WdgM 1.2.13 revision details added • Watchdog triggering in Low Power Mode added
2019-03-25	1.1.9	YongHyun Han	4.3 4.4 5.2.8 9.3.8	• WdgM 1.2.14 revision details added • Unnecessary descriptions removed • WdgM Condition Value meaning added • WdgM Condition Value configuration cautions added
2019-10-16	1.1.10	YongHyun Han	4.3 5 9	• WdgM 1.2.14.0 distributed • Modified parameter category (“Fixed” -> “Changeable”) • File structure changed (Moved lib folder file to delivery)
2020-03-13	1.1.11	YongHyun Han	4.3	• WdgM 1.2.15.0 distributed
2020-11-18	1.1.12	YongHyun Han	4.3 4.4 5.1 5.2 9.2 9.3	• History of WdgM 1.2.16.0 distribution added • Restrictions regarding the use of S32K1xx FTM added • WdgM configuration parameter guide added • WdgM configuration parameter guide added • WdgM configuration parameter guide added • WdgM configuration parameter guide added
2021-01-12	1.1.13	YongHyun Han	4.3	• History of WdgM 1.2.17.0 distribution added
2021-12-21	1.1.14	YongHyun Han	4.3	• History of WdgM 1.2.18.0 distribution added
2022-07-28	1.1.15	YongHyun Han	4.3	• History of WdgM 1.2.19.0 distribution added
2022-08-23	1.1.16	Gongbin Lim	4.3	• History of WdgM 1.2.20.0 distribution added

2023-03-07	1.1.17	YongHyun Han	4.3	• History of WdgM 1.2.20.1 distribution added
2023-07-17	1.1.18	HyeonSeok Park	4.3 4.4	• History of WdgM 1.2.21.0 distribution added • User confirmation about using internal/external watchdog added

Table of contents

1. Overview

This document is created based on the AUTOSAR standard SRS/SWS, and for more functional details when using the module, please see the reference documents below.

Each configuration category is defined as follows.

- Changeable (C) : Items that can be configured by users
- Fixed (F) : Items that cannot be changed by users
- NotSupported (N) : Items that are not used

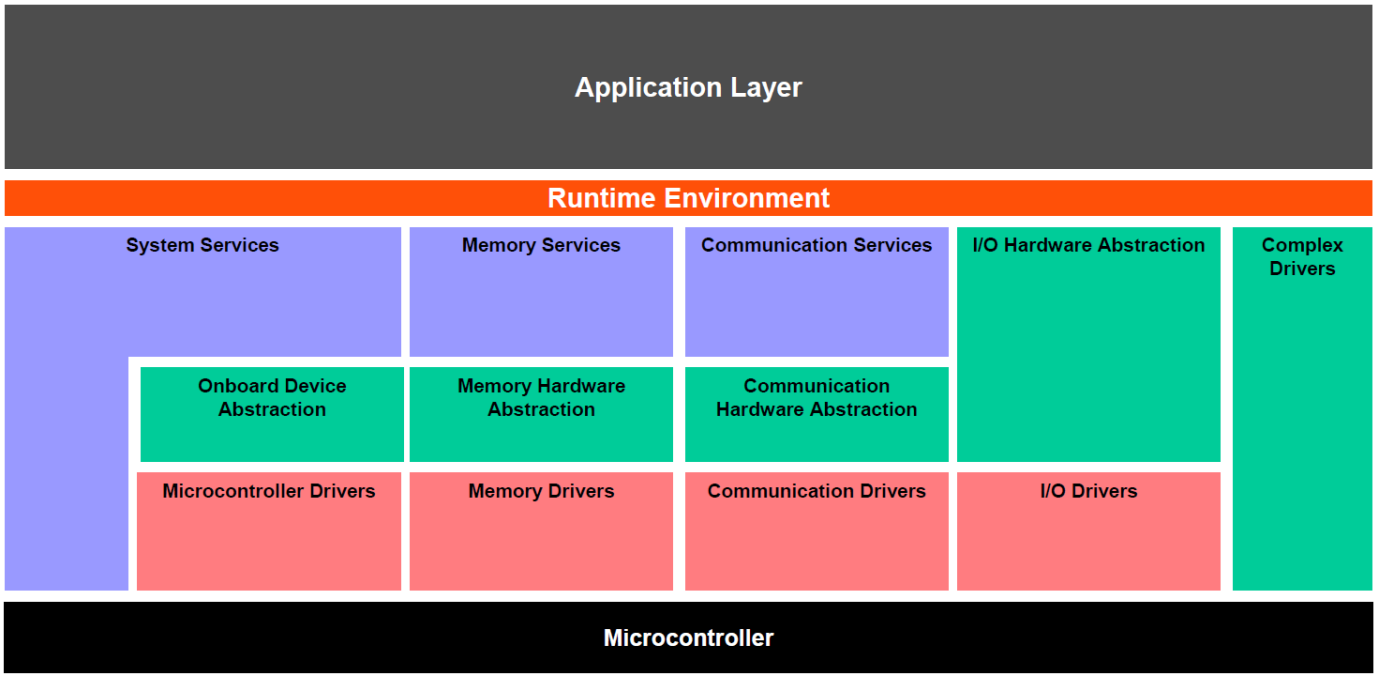
2. Reference

Sl. No.	Title	Version
1.	AUTOSAR BSW Service API Guide.doc	1.0.0 or later
2.	AUTOSAR_SWS_WatchdogManager.pdf	2.2.0
5.	AUTOSAR_TR_BSWModuleList.pdf	1.6.0

3. AUTOSAR System

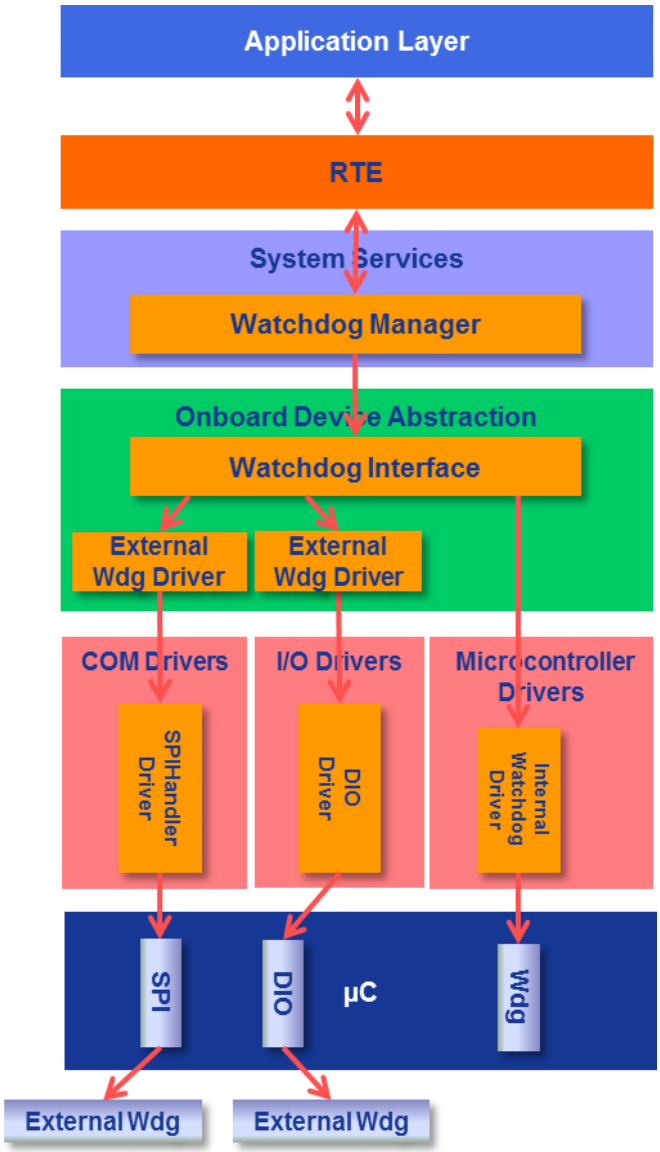
3.1 Overview of Software Layers

The AUTOSAR platform has a layered architecture as illustrated below. The AUTOSAR platform can be divided into Service Layer, ECU Abstraction Layer, Complex Device Drivers and Microcontroller Abstraction Layer.

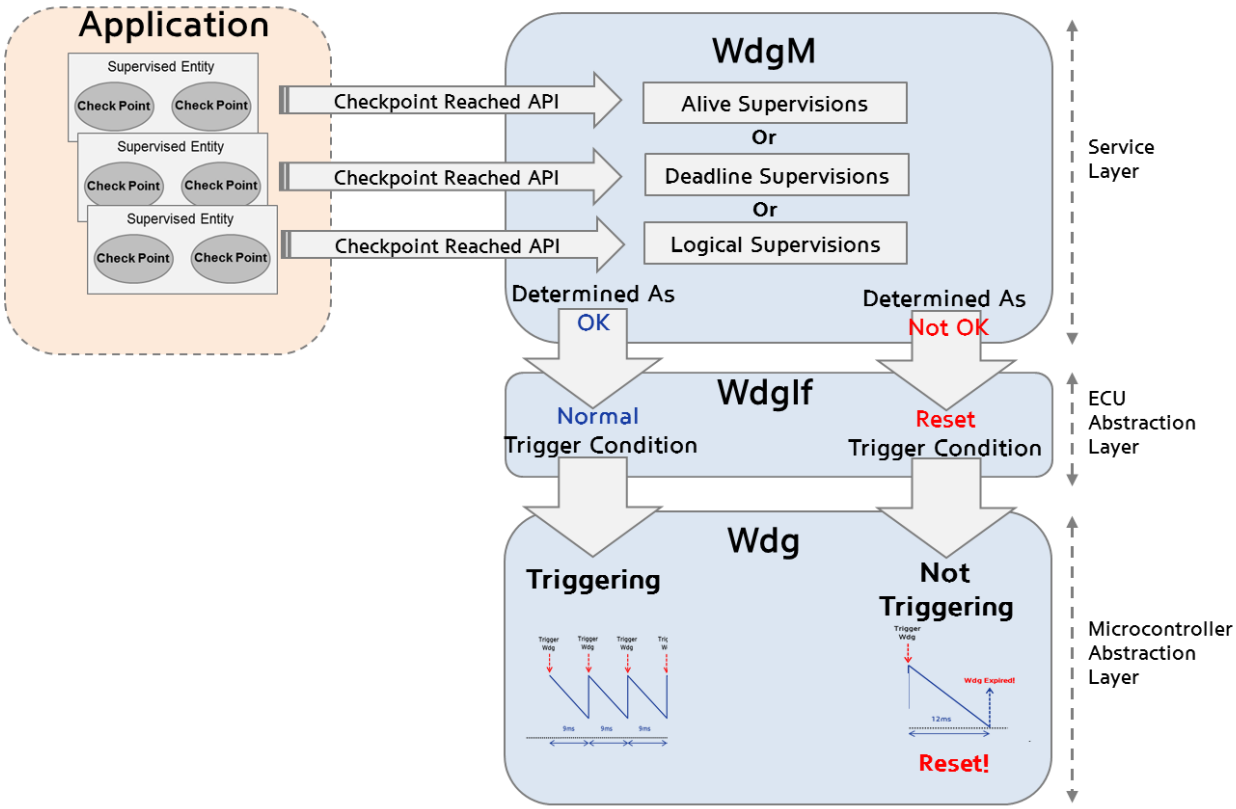


3.2 AUTOSAR WdgM Module

The AUTOSAR layers and the interface between modules for using the Wdg Stack are shown below. The Wdg Stack consists of the following modules: WdgM (Watchdog Manager), WdgIf (Watchdog Interface), and Wdg (Watchdog Driver).
WdgM : monitors the operation of supervised entity, provides triggering conditions for Wdg, makes requests for switching Wdg modes, and handles errors
WdgIf : serves as an abstraction of Wdg
Wdg : triggers HW Wdg, and controls HW Wdg modes



WdgM only monitors whether supervised entities operate according to monitoring conditions while WdgIf abstracts Wdg, and Wdg triggers HW Watchdog.



WdgM monitors whether supervised entities operate according to monitoring conditions (Alive Supervision, Deadline Supervision, Logical Supervision) and provides triggering conditions to Wdg through WdgIf based on monitoring results. In other words, WdgM informs Wdg of normal triggering conditions through WdgIf if supervised entities operate normally according to monitoring conditions; but if it is decided that Wdg reset is necessary because supervised entities violated monitoring conditions, WdgM informs Wdg of triggering conditions as 0 through WdgIf, ultimately leading Wdg to expire and be reset.

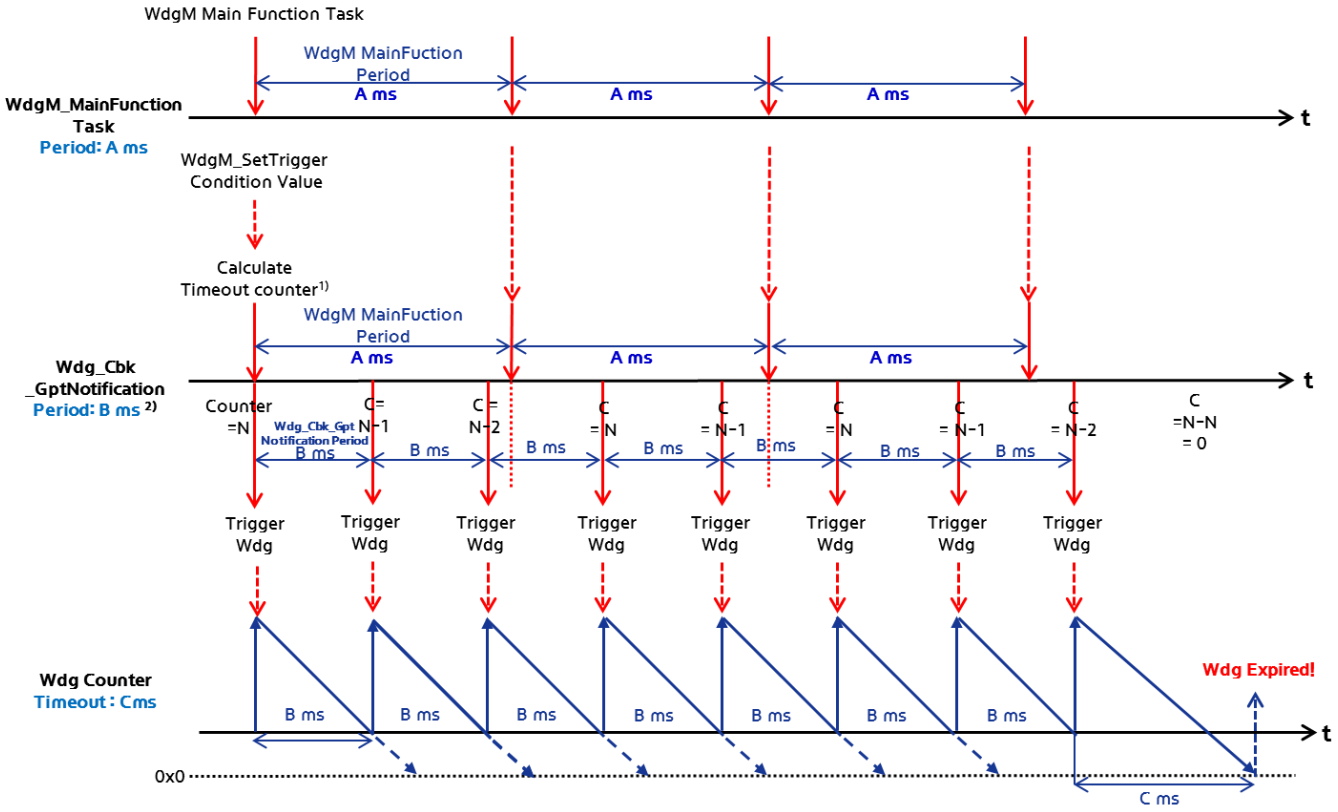
Monitoring conditions provided by WdgM are the following three.

Alive Supervision
Alive Supervision defines one checkpoint for a supervised entity and monitors the supervised entity based on Supervision Cycle (WdgM_MainFunction cycle).

Deadline Supervision
Deadline Supervision defines two checkpoints for supervision and monitors execution time in the execution time range between the two points (minimum execution time <= actual execution time <= maximum execution time).

Logical Supervision
Logical Supervision defines the order of checkpoints of supervised entity and monitors the execution order.

Wdg operation is divided by MCU according to Wdg design. Wdg operation varies according to Freescale and Infineon, and MCAL implementation is also divided by MCU.
(See AUTRON_AUTOSAR_Wdg_Module_UM for Wdg operation per each MCU.)
Wdg module triggers following WdgM MainFunction operate as in the image below.
(MCU: Freescale)



1) Recalculate timeout counter = Int (WdgM SetTrigger Condition Value / Wdg_Cbk_GptNotification Period), Ex) 30ms / 9ms = 3.33 ≒ 3
2) Wdg_Cbk_GptNotification Period = Timeout Period * (3 / 4)

4. Product Release Notes

4.1 Overview

This chapter is intended to provide the release information on the Hyundai AutoEver WdgM module, describing the features and restrictions of different release versions of the WdgM module software product.

4.2 Scope of the release

All content in this document is limited to the following Hyundai AutoEver WdgM module.

Module	Autosar version	Module version
WdgM	4.0.3	1.2.21

※ Module version refers to the SW version of the BswModule Description (Bswmd) file of each module.

4.3 Change Log

4.3.1 Version 1.2.21.0

➢ **Improvements**
- Add user confirmation of integration_Wdg about using internal/external wdg
- Cause: Require WdgStack_Callout.h modification of integration_Wdg when user enables/disables internal/external watchdog driver
- Operation effect: None
- Setting effect: None
- ASW Action: None

➢ **Improvements**
- Improve to sort input file list of generation files
- Cause: Generator results changed even though there was no configuration change
- Operation effect: None
- Setting effect: None
- ASW Action: None

➢ **Improvements**
- Generator changed not to occur generation errors when Dem set to Off
- Cause: Generation error occurs when Dem set to Off
- Operation effect: None
- Setting effect: None
- ASW Action: None

➢ **Improvements**
- Modified MCAL AR Version Check Range
- Cause: New MCAL version released
- Operation effect: None
- Setting effect: None
- ASW Action: None

4.3.2 Version 1.2.20.1

➢ **Improvements**
- An english UM document added
- Cause: Request for english UM docunemt
- Operation effect: None
- Setting effect: None
- ASW Action: None

4.3.3 Version 1.2.20.0

➢ **Improvements**
- Supported R44 OS for Version Check
- Cause: Compile errors occur when combining S32K1x controllers.
- Operation effect: None
- Setting effect: None
- ASW Action: None

4.3.4 Version 1.2.19.0

➢ **Improvements**
- Improved security coding to comply with the UNECE Cyber Security regulations
- Cause: Required to comply with the UNECE Cyber Security regulations
- Operation effect: None
- Setting effect: None
- ASW Action: None

4.3.5 Version 1.2.18.0

➢ **Improvements**
- Conducted static analysis
- Cause: Needed to correct or justify static analysis violations
- Operation effect: None
- Setting effect: None
- ASW Action: None
-
- Improved security coding to comply with the UNECE Cyber Security regulations
- Cause: Violation of the UNECE Cyber Security regulations occurred.
- Operation effect: None
- Setting effect: None
- ASW Action: None

- Applied a new document template
- Cause: The template has been changed due to company merger.
- Operation effect: None

- Setting effect: None
- ASW Action: None

4.3.6 Version 1.2.17.0

➢ Improvements

- Conducted static analysis
- Cause: Needed to correct or justify static analysis violations
- Operation effect: None
- Setting effect: None
- ASW Action: None

4.3.7 Version 1.2.16.0

➢ **Improvements**

- Conducted static analysis
- Cause: Needed to correct or justify static analysis violations
- Operation effect: None
- Setting effect: None
- ASW Action: None
- Restricted the FTM use of S32K1xx
- Cause: Added restrictions due to common use of OsProfiler and FTM unit
- Operation effect: None
- Setting effect: None
- ASW Action: None
- WdgM configuration parameter guide added
- Cause: Guide was necessary for the time taken from the occurrence of an error to the actual reset and parameter configuration such as AliveSupervision Min/Max Margin.
- Operation effect: None
- Setting effect: None
- ASW Action: None
- Changed the logic for checking transfer factor range of WdgM_SetMode()
- Cause: Needed to improve unnecessary operation that occurs when WdgMMode among WdgM_SetMode() transfer parameters is beyond the range.
- Operation effect: None
- Setting effect: None
- ASW Action: None

4.3.8 Version 1.2.15.0

➢ **Improvements**

- Added DataTypeMappings configuration to the Bswmd file
- Cause:Validation check failed due to the omission of the relevant configuration during saferte validation check
- Operation effect: None
- Setting effect: None
- ASW Action: None

4.3.9 Version 1.2.14.0

➢ **Improvements**
- Modified file structure and parameter categories
- Cause: Needed to change file structure and parameter for Code Open
- Operation effect: None
- Setting effect: None
- ASW Action: None

4.3.10 Version 1.2.14

➢ **Improvements**
- WdgM Condition Value configuration cautions added
- Cause: Unwanted watchdog reset could occur due to WdgM Condition Value misconfiguration.
- Operation effect: None
- Setting effect: None
- ASW Action: None
- Modified code coverage test codes of WdgM_Init()
- Cause: Code coverage test failed at WdgM_Init().
- Operation effect: None
- Setting effect: None
- ASW Action: None
- Modified code coverage test codes of WdgM_EvaluateGlobalStatus()
- Cause: Code coverage test failed at WdgM_EvaluateGlobalStatus ()
- Operation effect: None
- Setting effect: None
- ASW Action: None
- Improved WdgM module misra-c 2012 violations
- Cause: WdgM module misra-c 2012 violations occurred.
- Operation effect: None
- Setting effect: None
- ASW Action: None
- Corrected compile errors caused by version check violations of the MCU module (MCU module 4.0 - 4.4 support)
- Cause:During version check, compile when using MCU modules other than 4.0
- Operation effect: None
- Setting effect: None
- ASW Action: None

4.3.11 Previous Version

➢ **Version 1.2.13**
- Specified watchdog triggering considerations at Sleep Mode in the user manual
- Improved MISRA-C 2012 violations
-

➢ **Version 1.2.12**
- Added SWP Error Code to UM

- Improved MISRA-C violations
-

➢ **Version 1.2.11**
- Added Accessed Mode Group, Bsw Mode Receiver Policy, Bsw Service Dependency, Bsw Mode Switch Event when generated
-

➢ **Version 1.2.10**
- Added Disable In mode configuration when generating Bswmd
-

➢ **Version 1.2.9**
- Applied UM review results
-

➢ **Version 1.2.8**
- Applied UM separation
-

➢ **Version 1.2.7**
- Additionally revised generation alignment of the WdgM_Cfg.h/c file
-

➢ **Version 1.2.6**
- Corrected partial generation errors of #define definition
-

➢ **Version 1.2.5**
- Corrected SWC-BSW-MAPPING generation errors
-

➢ **Version 1.2.4**
- Corrected memory section designation errors
- Corrected arrangement size fixation errors when making a library
- Corrected some struct duplicate generation errors of the WdgM_Cfg.h/c file

➢ **Version 1.2.3**
- Added runnable arguments configuration generation of Swcd_WdgM.arxml

➢ **Version 1.2.2**
- Changed WdgM ArPackage route
- Separated Swcd generation mode (fixed part of Swcd_WdgM_Fixed.arxml, changed part of Swcd_WdgM.arxml)

➢ **Version 1.2.1**
- Corrected ModeAccessPoints generation errors of Swcd_WdgM.arxml
- Enhanced AliveSupervision configuration validation

➢ **Version 1.2.0**
- Added Bswmd_WdgM.arxml generation feature
- Changed AliveSupervision PPort naming

4.4 Limitations

- Deadline Max/Min value acceptance range of Deadline Supervision

  Deadline supervision internally uses OsCounter. Thus when the deadline max/min value of deadline supervision is smaller than OsCounter resolution to which WdgMSupervisedEntity refers, deadline supervision execution time cannot be monitored.

- WdgM_MainFunction cycle configuration when using Infineon Wdg

  Infineon Wdg triggering is executed by Gtm timer that Wdg driver implements. This timer stops/restarts after being renewed to new timeout value for every WdgM_MainFunction execution, and if WdgM_MainFunction cycle is set to smaller than the WdgNSlow/FastServiceGtmCbkTime value of Infineon Wdg configuration, reset by watchdog occurs because Gtm timer callback is not called.

- Mode changing restrictions when using Renesas Wdg

  Mode changing of Renesas RH850 Wdg is not allowed after operation.

  WDGIF_SLOW_MODE → WDGIF_FAST_MODE, WDGIF_OFF_MODE unavailable

  WDGIF_FAST_MODE → WDGIF_SLOW_MODE, WDGIF_OFF_MODE unavailable

- Watchdog triggering in Sleep Mode

  Periodic triggering is necessary when using hardware watchdog in sleep mode. (see 9.3.7)

- FTM use restrictions of S32K1xx

  For periodic triggering of intern/external watchdog, S32K1xx uses FTM in GPT module, and the unit is used in OsProfiler as well. Thus changing configuration about FTM3 in OsProfiler or internal/external watchdog may affect other modules.

- User confirmation for using Internal/External Watchdog

  Depending on whether the Internal/External Watchdog Driver is used, the values of WDGSTACK_ENABLE_INTERNAL_WDG and WDGSTACK_INTERNAL_WDG in WdgStack_Callout.h of integration_Wdg should be set to STD_ON/STD_OFF.

4.5 Deviations

- Ecuc Partition Ref feature of WdgMSupervisionEntity (SWS – WDGM360_Conf)

  This is used as information to deactivate a supervised entity located in the relevant partition from the supervised entities of the currently using WdgMMode when restarting partition.

  Feature unsupported

- Os Application Ref feature of WdgMSupervisionEntity (SWS – Chap 7.3.3, 7.9.1.2)

  This is used as information to restart the relevant partition only when the status of a supervised entity located in Non-trusted OsApplication transitions to WDGM_LOCAL_STATUS_FAILED.

  Feature unsupported

- WdgMMode change when calling WdgM_DeInit (SWS – Chap 7.10.3)

  It needs to be changed to WdgMMode configured in advance for DeInit when calling WdgM_DeInit.

  The current implementation mode does not change WdgMMode and executes WdgIf_SetTriggerCondition by using WdgMConfigSet / WdgMDeInitTimeout configuration value for WdgMTrigger of WdgMMode in use.

- PPort Short Name (SWS – WDGM147, WDGM149)

  PPort short names generated by WdgM need to be used as alive000, alive001, …, alive〈#SE-1〉, mode000, mode001, …, mode〈#SE-1〉.

  The current implementation mode generates PPort short names as alive_〈short name of supervised entity〉, mode_〈short name of supervised entity〉 to easily identify the target PPort when connecting with RPort.

5. Configuration Guide

5.1 WdgMGeneral Container

Parameter Name	Value	Category
Defensive Behavior1)	true	C
Dem Stopped Supervision Report2)	true	C
Dev Error Detect	true	C
Immediate Reset3)	false	C
Off Mode Enabled4)	true	C
Version Info Api	false	C
Os Vendor5) (Vendor specific)	Use configuration at platform deployment	C
Error Info Api6) (Vendor specific)	false	N

1) When calling WdgM_SetMode
   - A feature that checks whether CallerID is a caller registered in WdgMCallerIds
2) If Global Supervision Status transitions to WDGM_GLOBAL_STATUS_STOPPED
   - A feature that reports DemEvent (WDGM_E_MONITORING)
3) If Global Supervision Status transitions to WDGM_GLOBAL_STATUS_STOPPED
   - A feature that calls Mcu_PerformReset if MCU provides immediate reset feature when immediate MCU reset is necessary
   - Time taken from error detection of supervised entity to reset may differ according to configuration.
   - If configuration is true, it tries to reset through Mcu_PerformReset when WdgM transitions to WDGM_GLOBAL_STATUS_STOPPED.
   - If configuration is false, a reset occurs after the termination of HW watchdog timeout after WDGM_GLOBAL_STATUS_STOPPED. Thus some delay may occur until the actual reset occurs after WdgM compares with a case where the configuration is true.
4) When calling WdgM_Init, WdgM_SetMode
   - A feature that allows WDGIF_OFF_MODE as WdgMMode / WdgMTrigger / Watchdog Mode configuration
   - If set to true, Disable Allowed in Wdg Driver configuration needs to be set to true as well.
5) Configuration to apply differences caused by Os Vendor
6) If the supervision result is Incorrect or Local Supervisoin Status transitions to WDGM_LOCAL_STATUS_EXPIRED

- Save Supervision type, Supervised entity ID, Checkpoint ID, Local Supervision Status at the time of occurrence.
- The current error status of supervised entity can be obtained through WdgM_GetErrorInfo API.

5.1.1 WdgMCallerIds Configuration

Parameter Name	Value	Category
Short Name	User Defined	C
Caller Id1)	User Defined	C

1) See Defensive Behavior configuration.
   - Caller Id is Supervised Entity Id.
   - See the AUTOSAR_TR_BSWModuleList.pdf document not to use the reserved BswModule Id when allocating Supervised Entity Id.

5.1.2 WdgMWatchdog Configuration

WdgIfDevice configuration that WdgM intends to manage

Parameter Name	Value	Category
Short Name	Use configuration at platform deployment	C
Name	Use configuration at platform deployment	C
Device Ref1)	Use configuration at platform deployment	C

1) WdgIfDevice reference that is referencing Wdg

5.1.3 WdgMSupervisedEntity Configuration

Parameter Name	Value	Category
Short Name1)	User Defined	C
Id2)	User Defined	C
Ecuc Partition Ref3)		N
Os Application Ref4)		N
Internal Checkpoint Initial Ref5)	User Defined	C
Internal Checkpoint Final Ref6)	User Defined	C
Os Counter7) (Vendor specific)	User Defined	C

1) The following P Ports are generated at Swcd_WdgM.arxml as ShortName.

- alive_'ShortName' : WdgM_AliveSupervion(ClientServerInterface)
- mode_'ShortName' : WdgM_IndividualMode(ModeSwitchInterface)

2) Used as API factors as follows
   - WdgM_SetMode : Use Supervised Entity Id for CallerID
   - WdgM_CheckpointReached : The user does not directly use as it is mapped in advance at Port as Port API Option.
   - WgdM_GetLocalStatus : Used as Supervised Entity Id factor to obtain local supervision status
   - WdgM_GetFirstExpiredSEID : Used as a factor to obtain the first expired supervised entity ID
   - WdgM_GetErrorInfo : Used as Supervised Entity Id factor to obtain error information
3) Partition reference at which Supervised Entity is located
   - Used as information to deactivate a supervised entity located in the relevant partition from the supervised entities of the currently using WdgMMode when restarting partition.
   - Feature unsupported
4) OsApplication reference at which Supervised Entity is located
   - Used as information to restart the relevant partition only when the status of a supervised entity located in Non-trusted OsApplication transitions to WDGM_LOCAL_STATUS_FAILED.
   - Feature unsupported
5) Internal Graph Start Checkpoint Reference
6) Internal Graph End Checkpoint Reference
7) OsCounter reference configuration to be used at internal WdgM to obtain and compare timestamp when a supervised entity uses deadline supervision

5.1.4 WdgMCheckpoint Configuration

Targets of Alive / Deadline / Logical Supervion

Parameter Name	Value	Category
Short Name	User Defined	C
Id1)	Increase sequentially starting from 0	C

1) Used as a factor when calling WdgM_CheckpointReached

5.1.5 WdgMInternalTransition Configuration

Set transitions that comprise Internal Graph

Parameter Name	Value	Category
Short Name	User Defined	C
Source Ref1)	User Defined	C
Dest Ref1)	User Defined	C

1) Internal Transition Start/End Checkpoint Reference

5.2 WdgMConfigSet Container

Parameter Name	Value	Category
Short Name	User Defined	C
De Init Timeout1) (Vendor specific)	User Defined	C
Initial Mode2)	User Defined	C

1) Timeout value (msec) to be used when calling WdgM_DeInit

   Remark) DeInit Timeout Default Value is set to 4s when the initial platform is deployed. In case of reprogramming, Positive Response needs to be executed within 5s based on ES specifications after requesting reprogramming from diagnosis device.
   Default value is set by estimating the maximum run time before WdgM_Deinit execution as 1s and the maximum run time after WdgM_Deinit as 4s.
   Set DeInit Timeout Value to adequate 4s or lower in consideration of NVM Write All time.

2) WdgMMode to be used when calling WdgM_Init

5.2.1 WdgMDemEventParameterRefs Configuration

Parameter Name	Value	Category
Short Name	User Defined	C
WDGM_E_IMPROPER_CALLER1)	User Defined	C
WDGM_E_MONITORING1)	User Defined	C
WDGM_E_SET_MODE1)	User Defined	C

1) Set to refer to DemEventParameter for WdgM error report configured in Dem module.

5.2.2 WdgMMode Configuration

Preconfigured set for monitoring conditions, supervised entities, Wdg configuration and can be changed during runtime

Parameter Name	Value	Category
Short Name	User Defined	C
Expired Supervision Cycle Tol1)	User Defined	C
Id2)	Increase sequentially starting from 0	C
Supervision Cycle3)	Set the same as WdgM_MainFunction cycle	C

1) When transitioning to WDGM_GLOBAL_STATUS_EXPIRED, the number of cycle that maintains

the status before transitioning to WDGM_GLOBAL_STATUS_STOPPED (multiples of WdgM_MainFunction cycle)

- Used for obtaining delay before reset request (when transitioning to WDGM_GLOBAL_STATUS_STOPPED)

Ex) Expired Supervision Cycle Tol : 3, Supervision Cycle : 0.01(sec)

WDGM_GLOBAL_STATUS_EXPIRED maintained during 3 x 0.01(s) = 0.03(sec)

* Time taken until watchdog reset may differ according to configuration. See chapter 9.2.1 and revise parameters according to system operation.

2) Used as API factors as follows
   - WdgM_SetMode : Used as change request WdgMMode Id factor
   - WdgM_GetMode : Used as the current WdgMMode Id obtaining factor
3) Set the same as WdgM_MainFunction cycle
   - See Rte / BswInstance_WdgM / RteBswEventToTaskMappin_TE_WdgM / BswEventRef.

5.2.3 WdgMAliveSupervision Configuration

Parameter Name	Value	Category
Short Name	User Defined	C
Expected Alive Indications1)	User Defined	C
Max Margin2)	User Defined	C
Min Margin2)	User Defined	C
Supervision Reference Cycle3)	User Defined	C
Checkpoint Ref4)	User Defined	C

1) The number of WdgM_CheckpointReached callout of target checkpoint required during supervision reference cycle configuration cycle

   * The configuration is value used to decide whether the WdgM system conducts normal/abnormal operation. See chapter 9.2.2 and revise parameters according to system operation.

2) The number of allowable addition(+)/decrease(-) in expected alive indications

   Ex) Expected Alive Indications : 4, Max Margin : 2, Min Margin : 1

   (4 - 1) <= allowable number of alive indication <= (4 + 2)

   * The configuration is value used to decide whether the WdgM system conducts normal/abnormal operation. See chapter 9.2.2 and revise parameters according to system operation.

3) Multiples of WdgM_MainFunction cycle as monitoring cycle for the target checkpoint

Ex) Supervion Reference Cycle : 2, WdgM_MainFunction Cycle : 0.01(sec)

2 x 0.01(s) = 0.02(sec)

* Time taken until watchdog reset may differ according to configuration. See chapter 9.2.1 and revise parameters according to system operation.

4) Checkpoint for Alive Supervision

5.2.4 WdgMDeadlineSupervision Configuration

Parameter Name	Value	Category
Short Name	User Defined	C
Deadline Max1)	User Defined	C
Deadline Min1)	User Defined	C
Deadline Start Ref2)	User Defined	C
Deadline Start Ref2)	User Defined	C

1) Boundary value of allowable execution time between start and end checkpoints

Ex) Deadline Max : 0.035(sec), Deadline Min : 0.005(sec)

0.005(sec) <= execution time <= 0.035(sec)

* The configuration is value used to decide whether the WdgM system conducts normal/abnormal operation. See chapter 9.2.3 and revise parameters according to system operation.

2) Start/End Checkpoint for Deadline Supervision

5.2.5 WdgMExternalLogicalSupervision Configuration

External Graph Configuration

Parameter Name	Value	Category
Short Name	User Defined	C
External Checkpoint Initial Ref1)	User Defined	C
External Checkpoint Final Ref2)	User Defined	C

1) Reference for Start Checkpoints of External Graph
2) Reference for End Checkpoints of External Graph

5.2.6 WdgMExternalTransition Configuration

Set transitions that comprise External Graph

Parameter Name	Value	Category
Short Name	User Defined	C
Source Ref1)	User Defined	C
Dest Ref1)	User Defined	C

1) External Transition Start/End Checkpoint Reference

5.2.7 WdgMLocalStatusParams Configuration

Supervised Entity Configuration in WdgMMode

Parameter Name	Value	Category
Short Name	User Defined	C
Failed Alive Supervision Ref Cycle Tol1)	User Defined	C
Local Status Supervised Entity Ref2)	User Defined	C

1) When the alive supervision result is incorrect, the number of cycle that maintains WDGM_LOCAL_STATUS_FAILED instead of immediately transitioning to WDGM_LOCAL_STATUS_EXPIRED (multiples of WdgM_MainFunction cycle)

   * Time taken until watchdog reset may differ according to configuration. See chapter 9.2.1 and revise parameters according to system operation.

2) Supervised Entity Reference

5.2.8 WdgMTrigger Configuration

Wdg Configuration in WdgMMode

Parameter Name	Value	Category
Short Name	User Defined	C
Condition Value1)	User Defined	C
Watchdog Mode2)	User Defined	C
Watchdog Ref3)	User Defined	C

1) Value that is sent by WdgM to Wdg through WdgIf so that Wdg trigger watchdog normally when WdgM_MainFunction of WdgM is normal. Wdg needs to ensure that watchdog does not expire during the time set by the condition value. Condition value needs to be bigger than watchdog timeout value according to Wdg mode.

   * Time taken until watchdog reset may differ according to configuration. See chapter 9.2.1 and revise parameters according to system operation.

2) Wdg Operation Mode
   - WDGIF_OFF_MODE : Suspend Wdg (WdgMGeneral / Off Mode Enabled : true, WdgGeneral / Disable Allowed : true)
   - WDGIF_SLOW_MODE : Apply WdgSettingsConfig / WdgSettingsSlow configuration
   - WDGIF_FAST_MODE : Apply WdgSettingsConfig / WdgSettingsFast configuration

3) WdgMGeneral / Watchdog Reference

5.3 System Configuration

5.3.1 ApplicationSwComponentType Configuration

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

1) ApplicationSwComponent Generation
2) RPort Generation
   A. For CheckpointReached callout: one is generated per supervised entity.
      i. Corresponding PPort : / WdgM / ServiceSwComponentTypes / WdgM / alive_'Supervised Entity Short Name'
      ii. Required Interface : / WdgM / ClientServerInterfaces / WdgM_AliveSupervision
      iii. Client Com Spec : / WdgM / ClientServerInterfaces / WdgM_AliveSupervision / CheckpointReached
   B. For the reception of changing Local Supervision Status event: one is generated per supervised entity (optional).
      i. Corresponding PPort : / WdgM / ServiceSwComponentTypes / WdgM / mode_'Supervised Entity Short Name'
      ii. Required Interface : / WdgM / ModeSwitchInterfaces / WdgM_IndividualMode
   C. For the reception of changing Global Supervision Status event: only one is generated (optional).
      i. Corresponding PPort : / WdgM / ServiceSwComponentTypes / WdgM / globalMode
      ii. Required Interface : / WdgM / ModeSwitchInterfaces / WdgM_GlobalMode
   D. For the use of WdgM Api: only one is generated.
      i. Corresponding PPort : / WdgM / ServiceSwComponentTypes / WdgM / WdgM_API_P
      ii. Required Interface : / WdgM / ClientServerInterfaces / WdgM_API
      iii. Client Com Spec : selectively add necessary operation only.
3) Internal Behavior Generation
4) Runnalbe Generation
   A. Add Runnable according to design
5) Synchronous Server Call Point or Mode Access Point Generation
   A. Synchronous Server Call Point for Runnable
      i. RPort and Operation configuration intended to use
6) Swc Mode Switch Event Generation
   A. Event configuration intended to receive status change

5.3.2 Assembly Sw Connector Configuration

※ See Help of Odin Studio Assembly Sw Connector configuration.

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_SupervisedEntityIdType

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-〈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

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, AUTOMATIC, WDGM_APPL_CONST) ConfigPtr)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non reentrant
Parameters (In)	ConfigPtr
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, WDGM_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
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, uint16 CallerID)
Service ID	0x03
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	Mode, CallerID
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, AUTOSAR_uint16 usCallerID) <P> : R-Port Name

6.3.5 WdgM_GetMode

Function Name	WdgM_GetMode
Syntax:	FUNC(Std_ReturnType, WDGM_CODE) WdgM_GetMode(P2VAR(WdgM_ModeType, AUTOMATIC, 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> : R-Port Name

6.3.6 WdgM_CheckpointReached

Function Name	WdgM_CheckpointReached
Syntax:	FUNC(Std_ReturnType, WDGM_CODE) WdgM_CheckpointReached(WdgM_SupervisedEntityIdType SEID, WdgM_CheckpointIdType CheckpointID)
Service ID	0x0e
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	SEID, CheckpointID
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> : R-Port Name

6.3.7 WdgM_UpdateAliveCounter

Function Name	WdgM_UpdateAliveCounter
Syntax:	FUNC(Std_ReturnType, WDGM_CODE) WdgM_UpdateAliveCounter(WdgM_SupervisedEntityIdType SEID)
Service ID	0x04
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	SEID
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	This service is deprecated and it is provided only for backward compatibility
Preconditions	Watchdog Manager should be initialized
Configuration Dependency	None
In Communication with application SW-C	Rte_Call_<P>_UpdateAliveCounter(void) <P> : R-Port Name

6.3.8 WdgM_GetLocalStatus

Function Name	WdgM_GetLocalStatus
Syntax:	FUNC(Std_ReturnType, WDGM_CODE) WdgM_GetLocalStatus(WdgM_SupervisedEntityIdType SEID, P2VAR(WdgM_LocalStatusType, AUTOMATIC, WDGM_APPL_DATA) Status)
Service ID	0x0c
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	SEID
Parameters (Inout)	None
Parameters (Out)	Status
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_SupervisedEntityIdType ddSEID, WdgM_LocalStatusType* pStatus) <P> : R-Port Name

6.3.9 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
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> : R-Port Name

6.3.10 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> : R-Port Name

6.3.11 WdgM_GetFirstExpiredSEID

Function Name	WdgM_GetFirstExpiredSEID
Syntax:	FUNC(Std_ReturnType, WDGM_CODE) WdgM_GetFirstExpiredSEID(P2VAR(WdgM_SupervisedEntityIdType, WDGM_DATA, WDGM_APPL_DATA) SEID)
Service ID	0x10
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	SEID
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_SupervisedEntityIdType* pSEID) <P> : R-Port Name

6.3.12 WdgM_PmModeChange

Function Name	WdgM_PmModeChange
Syntax:	FUNC(Std_ReturnType, WDGM_CODE) WdgM_PmModeChange(WdgM_ChangeModeIntReqType ddReqMode)
Service ID	0xa1
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	ddReqMode
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	This service instructs the Watchdog Manager to change Wdg mode depending requested mode.
Preconditions	None
Configuration Dependency	None

6.3.13 Notes

■ In Communication with application SWC

See AUTOSAR BSW Service API Guide.doc for prototypes of RTE-based functions.

7. Generator

7.1 Generator Option

7.1.1 WdgM

Option	Description
-S	Software Component Description generation
-B	Bsw Module Description generation

7.2 Generator message

7.2.1 Error Messages

1) ERR013002: Unexpected Error Found. This error may be due to the incorrect configuration of the element(s) 'Element Name'. If you need any support contact HYUNDAI AUTRON Co., Ltd.
   - This error occurs, if the structure fields that are to be generated in the C Source file are empty. Contact HYUNDAI AUTRON Co., Ltd.

2) ERR013003: 'Component Name' Component is not present in the input file(s).
   - This error occurs, if WDGM component is not present in any of the input ECU Configuration Description File(s).

3) ERR013004: 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 reference path is not provided for the reference parameter.

Container Name	Parameter Name
WdgMConfigSet	WdgMInitialMode
WdgMAliveSupervision	WdgMAliveSupervisionCheckpointRef
WdgMDeadlineSupervision	WdgMDeadlineStartRef
	WdgMDeadlineStopRef
WdgMInternalTransition	WdgMInternalTransitionSourceRef
	WdgMInternalTransitionDestRef
WdgMExternalTransition	WdgMExternalTransitionSourceRef
	WdgMExternalTransitionDestRef
WdgMExternalLogicalSupervision	WdgMExternalCheckpointInitialRef
	WdgMExternalCheckpointFinalRef
WdgMLocalStatusParams	WdgMLocalStatusSupervisedEntityRef

Container Name	Parameter Name
WdgMTrigger	WdgMTriggerWatchdogRef
WdgMSupervisedEntity	WdgMInternalCheckpointInitialRef
	WdgMInternalCheckpointFinalRef
WdgMWatchdog	WdgMWatchdogDeviceRef

4) ERR013005: The parameter 'Parameter Name' in the container 'Container Name' should be configured.
   - This error occurs, if any of the mandatory configuration parameters mentioned below is not configured in ECU Configuration Description File.

Container Name	Parameter Name
WdgMMode	WdgMExpiredSupervisionCycleTol
	WdgMModeId
	WdgMSupervisionCycle
WdgMAliveSupervision	WdgMExpectedAliveIndications
	WdgMMaxMargin
	WdgMMinMargin
	WdgMSupervisionReferenceCycle
WdgMGeneral	WdgMDevErrorDetect
	WdgMVersionInfoApi
	WdgMDefensiveBehavior
	WdgMDemStoppedSupervisionReport
	WdgMImmediateReset
	WdgMOffModeEnabled
	WdgMOsVendor
WdgMSupervisedEntity	WdgMSupervisedEntityId
WdgMCheckpoint	WdgMCheckpointId
WdgMWatchdog	WdgMWatchdogName
WdgMCallerIds	WdgMCallerId
WdgMDeadlineSupervision	WdgMDeadlineMax
	WdgMDeadlineMin
WdgMLocalStatusParams	WdgMFailedAliveSupervisionRefCycleTol
WdgMTrigger	WdgMTriggerConditionValue
	WdgMWatchdogMode
BSW-IMPLEMENTATION	AR-RELEASE-VERSION
	VENDOR-ID
	SW-VERSION
BSW-MODULE-DESCRIPTION	MODULE-ID

5) ERR013006: The value configured for the parameter ‘Parameter Name’ in the container ‘Container Name’ should follow the pattern: <Pattern>.
   - This error occurs, when the parameter ‘Parameter Name’ is not configured as per the pattern.

Parameter Name	Container Name	Pattern	Example
AR-RELEASE-VERSION	BSW-IMPLEMENTATION	4.[0-9]+.[0-9]+	4.0.3
SW-VERSION			
WdgMWatchdogName	WdgMWatchdog	[a-zA-Z][a-zA-Z0-9₩_]*	Watchdog0

6) ERR013013: The reference path <Reference Path> provided for the parameter ‘Parameter Name’ in the container ‘Container Name’, having short name <Container Short Name> is incorrect.
   - This error occurs, if incorrect reference is provided for the reference parameter.

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

7) ERR013017: Value of the parameter ‘WdgMDeadlineMax’ in the container ‘WdgMDeadlineSupervision’ should be greater than the value of the parameter ‘WdgMDeadlineMin’ in the container ‘WdgMDeadlineSupervision’.
   - This error occurs, if value of the parameter ‘WdgMDeadlineMax’ in the container ‘WdgMDeadlineSupervision’ is less than the value of the parameter ‘WdgMDeadlineMin’ in the container ‘WdgMDeadlineSupervision’.

8) ERR013022: Value of the parameter ‘Parameter Name’ in the container ‘Container Name’ should not be configured as <0>.
   - This error occurs, if parameter name in the container name is configured as zero.

Container Name	Parameter Name
WdgMMode	WdgMSupervisionCycle

9) ERR013051: The value configured for the parameter ‘WdgMCheckpointId’ in the container ‘WdgMCheckpoint’ should be unique within the container ‘WdgMSupervisedEntity’.
   - This error occurs, if value configured for the parameter ‘WdgMCheckpointId’ in the container ‘WdgMCheckpoint’ is not unique within the ‘WdgMSupervisedEntity’.

10) ERR013052: Value configured for the ‘WdgMCheckpointId’ in the container ‘WdgMCheckpoint’ should be sequential and starts from <0> within the ‘WdgMSupervisedEntity’.
   - This error occurs, if value configured for the parameter ‘WdgMCheckpointId’ in the container ‘WdgMCheckpoint’ is not sequential and doesn’t start from ‘0’.

11) ERR013053: Reference path configured for the parameters ‘WdgMDeadlineStartRef’ and ‘WdgMDeadlineStopRef’ in the container ‘WdgMDeadlineSupervision’ should refer to same ‘WdgMSupervisedEntity’.
   - This error occurs, if Reference paths configured for the parameters ‘WdgMDeadlineStartRef’ and ‘WdgMDeadlineStopRef’ are not referring to same ‘WdgMSupervisedEntity’.

12) ERR013054: At least one of the containers ‘WdgMAliveSupervision’, ‘WdgMDeadlineSupervision’ and ‘WdgMExternalLogicalSupervision’ should be configured within the mode.
   - This error occurs, if none of the containers ‘WdgMAliveSupervision’, ‘WdgMDeadlineSupervision’ and ‘WdgMExternalLogicalSupervision’ is configured within the mode.

13) ERR013056: The value configured for the parameter ‘WdgMSupervisionCycle’ in the container ‘WdgMMode’ should be same across all the modes.
   - This error occurs, if the value configured for the parameter ‘WdgMSupervisionCycle’ in the container ‘WdgMMode’ is not same across all the modes.

14) ERR013057: At least one instance of the container ‘WdgMLocalStatusParams’ should be configured within the mode.
   - This error occurs, if Container ‘WdgMLocalStatusParams’ is not configured in any of the modes.

15) ERR013058: Reference path configured for the parameter 'Parameter Name' in the container 'Container Name' should refer to 'WdgMCheckpoint' container within 'WdgMSupervisedEntity' container.
   - This error occurs, if Reference path configured for the parameter 'Parameter Name' do not point to 'WdgMCheckpoint' container within 'WdgMSupervisedEntity' container.

16) ERR013059: Container 'WdgMCheckpoint' having ShortName <WdgMCheckpoint Container ShortName> is repeated with different values of parameter 'WdgMCheckpointId' across Supervised Entities. All the container instances having same ShortName should have same symbolic name value.
   - This error occurs, if WdgMCheckpoint is repeated with different values of parameter 'WdgMCheckpointId' across Supervised Entities.

Container Name	Parameter Name
WdgMInternalTransition	WdgMInternalTransitionSourceRef
	WdgMInternalTransitionDestRef

17) ERR013060: The value configured for the parameter 'Parameter Name' in the container 'Container Name' should be unique.
   - This error occurs, if the value configured for the parameter 'Parameter Name' in the container 'Container Name' is not unique.

Container Name	Parameter Name
WdgMSupervisedEntity	WdgMSupervisedEntityId
WdgMMode	WdgMModeId
WdgMWatchdog	WdgMWatchdogName

18) ERR013061: The value configured for the parameter 'WdgMModeId' in the container 'WdgMMode' should be sequential and should start from <0>.
   - This error occurs, if the value configured for the parameter 'WdgMModeId' in the container 'WdgMMode' is not sequential and does not start from <0>.

19) ERR013062: Reference path configured for the parameter 'Parameter Name' should refer to 'WdgMCheckpoint' containers within the 'WdgMSupervisedEntity' container.
   - This error occurs, if Reference path configured for the parameter 'Parameter Name' does not refer to 'WdgMCheckpoint' containers within the 'WdgMSupervisedEntity' container.

Parameter Name
WdgMInternalCheckpointInitialRef
WdgMInternalCheckpointFinalRef

20) ERR013063: <WdgMCheckpoint Container ShortName> of <WdgMSupervisedEntity Container ShortName> is repeated in more than one 'WdgMExternalLogicalSupervision' containers within <WdgMMode Container ShortName>, 'WdgMCheckpoint' containers referred should be unique across 'WdgMExternalLogicalSupervision' containers within a mode.
   - This error occurs, if Checkpoints referred are not unique across

WdgMExternalLogicalSupervision containers within any mode.

21) ERR013064: The ordered set <WdgMSupervisedEntity Container ShortName/ WdgMCheckpoint Container ShortName> and <WdgMSupervisedEntity Container ShortName/WdgMCheckpoint Container ShortName> referred by 'WdgMDeadlineStartRef' and 'WdgMDeadlineStopRef' parameters should be unique across the 'WdgMDeadlineSupervision' containers within a mode.
   - This error occurs, if ordered set of Checkpoints referred by 'WdgMDeadlineStartRef' and 'WdgMDeadlineStopRef' parameters is not unique across the 'WdgMDeadlineSupervision' containers.

22) ERR013065: The Reference path configured for the parameter 'WdgMWatchdogDeviceRef' in the container 'WdgMWatchdog' should be unique.
   - This error occurs, if the Reference path configured for the parameter 'WdgMWatchdogDeviceRef' in the container 'WdgMWatchdog' is not unique.

23) ERR013066: <WdgMCheckpoint Container ShortName> of <WdgMSupervisedEntity Container ShortName> is repeated in containers 'WdgMInternalTransition' and 'WdgMExternalTransition'. 'WdgMCheckpoint' containers referred should be unique across internal and external graphs within a mode.
   - This error occurs, if 'WdgMCheckpoint' containers referred are not unique across internal and external graphs within a mode.

24) ERR013067: At least two 'WdgMCheckpoint' containers referred by the parameters 'WdgMExternalTransitionSourceRef' and 'WdgMExternalTransitionDestRef' of 'WdgMExternalTransition' container should belong to different 'WdgMSupervisedEntity' containers within 'WdgMExternalLogicalSupervision' container.
   - This error occurs, if all the 'WdgMCheckpoint' containers referred by 'WdgMExternalTransition' container belong to the same 'WdgMSupervisedEntity' container.

25) ERR013068: Reference path configured for the parameter 'Parameter1 Name' and 'Parameter2 Name' in the container 'Container Name' should not be equal.
   - This error occurs, if paths provided for the reference parameters Parameter1 and Parameter2 are equal.

Parameter1 Name	Parameter2 Name	Container Name
WdgMDeadlineStartRef	WdgMDeadlineStopRef	WdgMDeadlineSupervision
WdgMExternalCheckpointFinalRef	WdgMExternalCheckpointInitialRef	WdgMExternalLogicalSupervision
WdgMExternalTransitionDestRef	WdgMExternalTransitionSourceRef	WdgMExternalTransition
WdgMInternalCheckpointInitialRef	WdgMInternalCheckpointFinalRef	WdgMSupervisedEntity
WdgMInternalTransitionDestRef	WdgMInternalTransitionSourceRef	WdgMInternalTransition

26) ERR013069: The reference path(s) configured for the parameter 'Parameter Name' in the

container 'Container Name' should be referred by the parameters 'Parameter1 Name' or 'Parameter2 Name' in the container 'Container1 Name'.

- This error occurs, if reference path(s) configured for the parameter 'Parameter Name' in the container 'Container Name' is(are) not referred by the parameters 'Parameter1 Name' or 'Parameter2 Name' in the container 'Container1 Name'.

Parameter Name	Container Name	Parameter1 Name	Parameter2 Name	Container1 Name
WdgMInternalCheckpointInitialRef	WdgMSupervisedEntity	WdgMInternalTransitionSourceRef	WdgMInternalTransitionDestRef	WdgMInternalTransition
WdgMInternalCheckpointFinalRef				
WdgMExternalCheckpointInitialRef	WdgMExternalLogicalSupervision	WdgMExternalTransitionSourceRef	WdgMExternalTransitionDestRef	WdgMExternalTransition
WdgMExternalCheckpointFinalRef				

Warning Messages

1) WRN013051: The ordered set ‹WdgMSupervisedEntity Container ShortName/ WdgMCheckpoint Container ShortName› and ‹WdgMSupervisedEntity Container ShortName/WdgMCheckpoint Container ShortName› referred by ‘WdgMInternalTransitionSourceRef’ and ‘WdgMInternalTransitionDestRef’ parameters should be unique across the ‘WdgMInternalTransition’ containers within the container 'WdgMSupervisedEntity'.
   - This warning occurs, if ordered set of Checkpoints referred by ‘WdgMInternalTransitionSourceRef’ and ‘WdgMInternalTransitionDestRef’ parameters is not unique across the ‘WdgMInternalTransition’ containers.

2) WRN013052: The ordered set ‹WdgMSupervisedEntity Container ShortName/ WdgMCheckpoint Container ShortName› and ‹WdgMSupervisedEntity Container ShortName/WdgMCheckpoint Container ShortName› referred by ‘WdgMExternalTransitionSourceRef’ and ‘WdgMExternalTransitionDestRef’ parameters should be unique across the ‘WdgMExternalTransition’ containers within the container 'WdgMExternalLogicalSupervision'.
   - This warning occurs, if ordered set of Checkpoints referred by parameters ‘WdgMExternalTransitionSourceRef’ and ‘WdgMExternalTransitionDestRef’ is not unique across the ‘WdgMExternalTransition’ containers.

Information Messages

1) INF013015: AUTOSAR Release version ‹Version› configured for the parameter ‘AR-RELEASE-VERSION’ in provided MDT file is not correct. AUTOSAR Release version should be one of the following: ‹Versions›.
   - This information message occurs, if the value of the element AR-RELEASE-VERSION present in the BSW Module Description template is configured other than 4.0.3

8. SWP Error Code

8.1 SWP Error Code List

8.1.1 WDGM_E_IMPROPER_CALLER

ErrorId Symbol	WDGM_E_IMPROPER_CALLER
Description	In case of WdgMGeneral->DefensiveBehavior=true, this occurs when caller id that is not registered in WdgMCallerIds calls WdgM_SetMode.
Cause of error	ASW
Platform default Action	NO RESET
Functional impact	Failure to change WdgMMode
Correlation with other modules	None
MCU	Common
Error type	Configuration, code
Measures to apply to application	Add Caller ID to WdgMGeneral->WdgMCallerIds configuration as Id (Supervised Entity ID) of Caller that calls WdgM_SetMode is missing.

8.1.2 WDGM_E_MONITORING

ErrorId Symbol	WDGM_E_MONITORING
Description	In case of WdgMGeneral->DemStoppedSupervisionReport=true configuration, this occurs when: 1. WdgM_Init execution fails: In case of WdgMGeneral->OffModeEnabled=false configuration, this occurs when the default setting of the relevant WdgM Watchdog is Off Mode. 2. WdgM_SetMode execution fails: When Wdg SetMode execution fails during initialization 3. WdgM_PerformReset is called: When ASW calls the relevant API 4. GlobalSupervisionStatus is changed to WDGM_GLOBAL_STATUS_STOPPED because supervise entity fails to meet monitoring conditions
Cause of error	ASW
Platform default Action	RESET
Functional impact	After an event occurs right before reset, it is reset after delay following WdgMConfigSet->WdgMMode->ExpiredSuspensionCycleTol configuration.
Correlation with other modules	None
MCU	Common
Error type	Configuration, code
Measures to apply to application	1. See the below and examine HW register. 1.1. WdgM_Init execution fails: As this occurs when HW Wdg mode change (register change) process fails in Wdg_SetMode function of MCAL Wdg driver, examination of Wdg register operation is necessary. 1.2. WdgM_SetMode execution fails: As this occurs when HW Wdg mode change (register change) process fails in Wdg_SetMode function of MCAL Wdg driver, examination of Wdg register operation is necessary. 1.3. WdgM_PerformReset is called: Intentional callout of WdgM_PerformRes

	et is normal operation. If it is not an intentional callout, review whether WdgM_PerformReset callout is configured. 1.4. GlobalSupervisionStatus is changed to WDGM_GLOBAL_STATUS_STOPPED: Need to analyze what causes WdgMMode in use to violate monitoring conditions (First Expired Supervised Entity ID, which is a supervised entity that violates monitoring conditions first, can be identified through WdgM_GetFirstExpiredSEID.) 2. Review ASW operation: Related tests need to be conducted as reset occurs when Dem Error occurs during runtime.

8.1.3 WDGM_E_SET_MODE

ErrorId Symbol	WDGM_E_SET_MODE
Description	This occurs in the following cases: 1. Wdg driver fails to change WdgMMode in WdgM_Init execution. 2. Wdg driver fails to change WdgMMode in WdgM_SetMode execution. For your information, WDGM_E_MONITORING event also occurs in case of WdgMGeneral->DemStoppedSupervisionReport->true configuration.
Cause of error	ASW
Platform default Action	RESET
Functional impact	SetTriggerCondition value is sent to Wdg as 0 when failing to change WdgM Mode. In the case of Wdg, Wdg Triggering is suspended when SetTriggerCondition value is 0, resulting in Watchdog Reset.
Correlation with other modules	None
MCU	Common
Error type	Configuration, code
Measures to apply to application	1. See the below and examine HW register. 1.1. Wdg driver fails to change WdgMMode in WdgM_Init execution: As this occurs when HW Wdg mode change (register change) process fails in Wdg_SetMode function of MCAL Wdg driver, examination of Wdg register operation is necessary. 1.2. Wdg driver fails to change WdgMMode in WdgM_SetMode execution: As this occurs when HW Wdg mode change (register change) process fails in Wdg_SetMode function of MCAL Wdg driver, examination of Wdg register operation is necessary. 2. Review ASW operation: Related tests need to be conducted as reset occurs when Dem Error occurs during runtime.

9. Appendix

9.1 Integration Additional Module

9.1.1 Os Module

9.1.1.1 OsTask Container

9.1.1.1.1 OsTask Configuration

OsTask configuration for WdgM module

Parameter Name	Value	Category
Short Name	OsTask_BSW_FG1_10ms	C
Activation	1	C
Priority	Use configuration at platform deployment	C
Schedule	FULL	C

Parameter Name	Value	Category
Short Name	OsTask_BSW_FG2_WdgMAPI	C
Activation	1	C
Priority	Use configuration at platform deployment	C
Schedule	FULL	C

9.1.1.2 OsAlarm Container

9.1.1.2.1 OsAlarm Configuration

OsAlarm configuration for WdgM module

Parameter Name	Value	Category
Short Name	OsAlarm_BSW_10ms	C
Accessing Application	Use configuration at platform deployment	C
Counter Ref	OsCounter reference of OsApplication at which WdgM is located	C

9.1.1.2.2 OsAlarmAction Configuration

Parameter Name	Value	Category
Short Name	Use configuration at platform deployment	C
Ref	OsTask_BSW_FG1_10ms	C

9.1.1.3 OsApplication Container

9.1.1.3.1 OsApplication Configuration

Add OsTask and OsAlarm for WdgM module to OsApplication of Master core.

Parameter Name	Value	Category
Short Name	Use configuration at platform deployment	C
Core Assignment	0	C
Trusted	true	C
App Alarm Ref	Add OsAlarm_BSW_10ms	C
App Counter Ref	Use configuration at platform deployment	C
App Isr Ref	Use configuration at platform deployment	C
App Task Ref	Add OsTask_BSW_FG1_10ms	C

9.1.2 RTE Module

9.1.2.1 RteBswModuleInstance Container

Rte configuration of WdgM as a BSW module

9.1.2.1.1 BswInstance_WdgM Configuration

1) RteBswModuleInstance Configuration

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

2) RteBswEventToTaskMapping Configuration
Task mapping and cycle configuration for WdgM_MainFunction
WdgM_MainFunction Cycle: See Period of / Bsw_WdgM / WdgM /BswInternalBehavior_WdgM / TE_WdgM

Parameter Name	Value	Category
Short Name	RteBswEventToTaskMapping	C
Bsw Immediate Restart	false	C
Bsw Position In Task	1	C
Bsw Event Ref	BswTE_WdgM_MainFunction	C
Bsw Mapped To Task Ref	OsTask_BSW_FG1_10ms	C
Bsw Used Os Alarm Ref	OsAlarm_BSW_10ms	C

3) RteBswExclusiveAreaImpl Configuration

Parameter Name	Value	Category
Short Name	RteBswExclusiveAreaImpl _GLOBALSUPVSNSTATUS_PROTECTION	C
Exclusive Area Impl Mechanism	ALL_INTERRUPT_BLOCKING	C
Bsw Exclusive Area Ref	GLOBALSUPVSNSTATUS_PROTECTION	C

4) Bsw Required Mode Group Connection Configuration

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

9.1.2.2 RteSwComponentInstance Container

9.1.2.2.1 SwcInstance_WdgM Configuration

ServiceSwComponent Instance configuration of WdgM module

1) RteSwComponentInstance Configuration

Parameter Name	Value	Category
Short Name	SwcInstance_WdgM	C
Software Component Instance Ref	Extracted Swc_WdgM reference	C

2) RteEventToTaskMapping Configuration
Task mapping and cycle configuration of WdgM_SetMode, WdgM_PerformReset

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_OIE_SetMode	C
Bsw Position In Task	0	C
Bsw Event Ref	OperationInvokedEvent_SetMode	C
Bsw Mapped To Task Ref	OsTask_BSW_FG2_WdgMAPI	C

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_OIE_PerformReset	C
Bsw Position In Task	1	C

Parameter Name	Value	Category
Bsw Event Ref	OperationInvokedEvent_Perform Reset	C
Bsw Mapped To Task Ref	OsTask_BSW_FG2_WdgMAPI	C

9.1.2.2.2 SwcInstance_WdgMTest Configuration

ServiceSwComponent Instance configuration of WdgM module

1) RteSwComponentInstance Configuration

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

2) RteEventToTaskMapping Configuration
Task mapping and cycle configuration of RE_WdgMTest
RE_WdgMTest Cycle: See Period of / App_WdgM / Swc_WdgMTest /IB_WdgMTest / TE_WdgMTest

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_TE_WdgMTest	C
Bsw Immediate Restart	false	C
Bsw Position In Task	2	C
Bsw Event Ref	TE_WdgMTest	C
Bsw Mapped To Task Ref	OsTask_ASW_FG1_10ms	C
Bsw Used Os Alarm Ref	OsAlarm_ASW_10ms	C

Task mapping of Global Supervision Status change event

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_SMSE_globalMode_OK	C
Bsw Position In Task	0	C
Bsw Event Ref	SwcModeSwitchEvent_globalMode_OK	C
Bsw Mapped To Task Ref	OsTask_ASW_FG2_WdgMTestMode	C

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_SMSE_globalMode_FAILED	C
Bsw Position In Task	1	C
Bsw Event Ref	SwcModeSwitchEvent_globalMode_FAILED	C

Parameter Name	Value	Category
Bsw Mapped To Task Ref	OsTask_ASW_FG2_WdgMTestMode	C

Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_SMSE_globalMode_EXPIRED	C
Bsw Position In Task	2	C
Bsw Event Ref	SwcModeSwitchEvent_globalMode_EXPIRED	C
Bsw Mapped To Task Ref	OsTask_ASW_FG2_WdgMTestMode	C

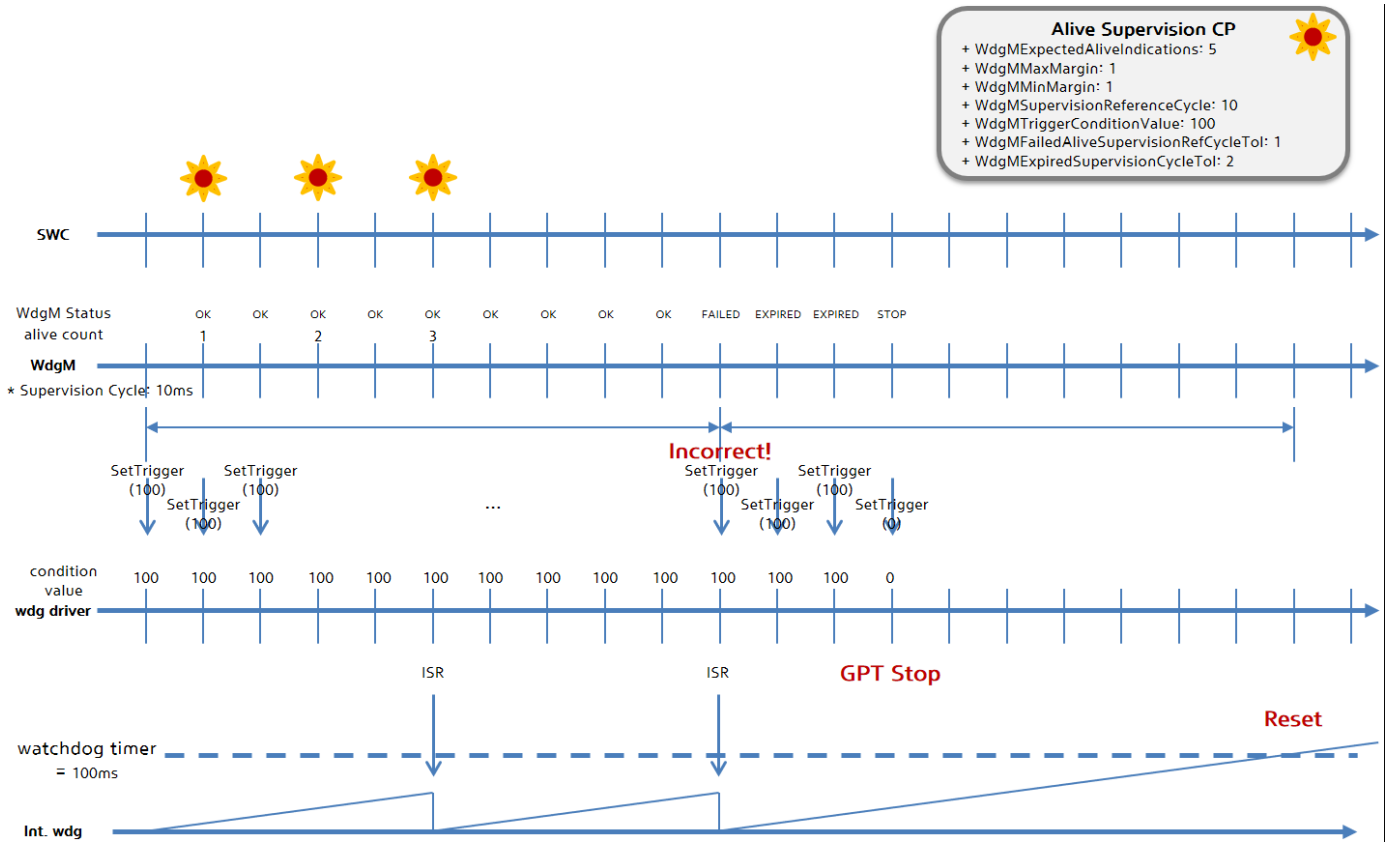
Parameter Name	Value	Category
Short Name	RteEventToTaskMapping_SMSE_globalMode_STOPPED	C
Bsw Position In Task	3	C
Bsw Event Ref	SwcModeSwitchEvent_globalMode_STOPPED	C
Bsw Mapped To Task Ref	OsTask_ASW_FG2_WdgMTestMode	C

9.2 Application User Configuration Guide

To add Alive / Deadline / Logical Supervision to WdgM, a design that suits the purpose needs to be developed in advance.

9.2.1 WdgM Status Transition

The image below shows WdgM status transition and watchdog stack operation during Alive Supervision Fail.



WdgM transitions to OK -> FAILED -> EXPIRED -> STOP when detecting an error of supervised entity. WdgM maintains FAILED status for WdgMFailedAliveSupervisionRefCycleTol and EXPIRED status for WdgMExpiredSupervisionCycleTol likewise. As such, the time taken from when WdgM detects an error of supervised entity to reset through STOP is related to the relevant parameters.

The image above shows WdgM status transition when Alive Supervision monitoring has an error. Unlike the image above, if Deadline Supervision or Logical Supervision detects an error, WdgM transitions to OK -> EXPIRED -> STOP. See 9.2.2 – 9.2.4 for more details of each monitoring technique.

Alive Supervision can decide whether alive supervision has an error per WdgMSupervisionReferenceCycle. Thus setting the relevant parameter to the right value may adjust the time from error occurrence to the actual reset within the time that system allows.

WdgM calls Wdg_SetTriggerCondition() per Supervision Cycle. WdgM allows Wdg driver to trigger watchdog timer for a particular period of time through the relevant function's parameter. Wdg driver

triggers watchdog timer until condition value is valid, and condition value is sent as value other than 0 as long as WdgM status is not STOP. Parameter sent by WdgM through Wdg_SetTriggCondition() may be set at WdgMTriggerConditionValue.

When WdgM status become STOP, WdgM calls the relevant function making Wdg_SetTriggerCondition() parameter 0. When condition value become 0, Wdg driver stops watchdog triggering, resulting in reset. Operation when Wdg_SetTriggerCondition() parameter is 0 varies with MCAL / MCU. For example, there is MCAL / MCU that decreases condition value per watchdog triggering without changing condition value to 0 and generates reset when condition value is not valid. Thus the operation of MCAL / MCU in use needs to be checked, and the time taken from SW error detection to watchdog reset occurrence may differ according to how the operation is processed. In particular, when maintaining condition value during WdgM STOP, WdgMTriggerConditionValue needs to be changed to suit system condition as reset time may be different according to WdgMTriggerConditionValue.

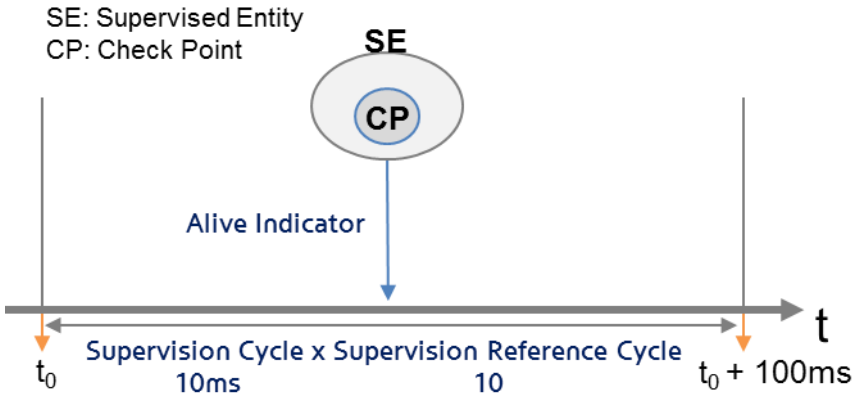
The image above is an example that sets watchdog timeout to 100ms. Watchdog timeout may be randomly set by the user according to MCAL / MCU. Therefore the configuration needs to be changed to suit system condition.

In the case of WdgM, the time taken from error detection to the actual reset is affected by WdgMExpiredSupervisionCycleTol, WdgMTriggerConditionValue, watchdog timeout, etc. as explained above. Additionally, it is affected by WdgMSupervisionReferenceCycle, WdgMFailedAliveSupervisionRefCycleTol, etc. when using Alive Supervision. Thus the relevant configuration values need to be set in consideration of time for the system to tolerate errors.

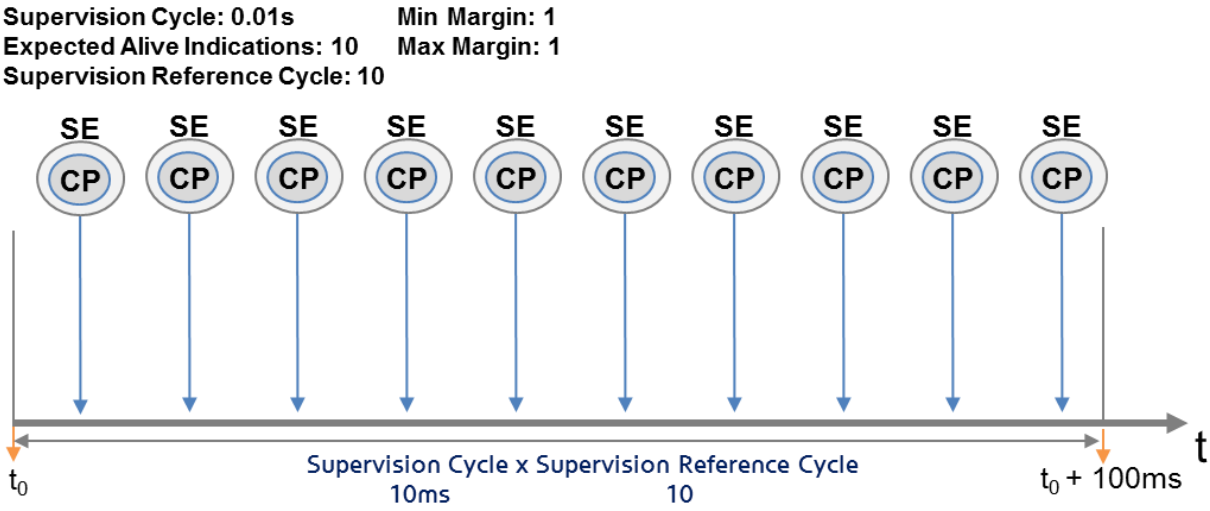
9.2.2 Alive Supervision

Alive Supervision defines one checkpoint for Supervised Entity and defines periodic execution time based on Supervision Cycle (WdgM_MainFunction cycle).

Supervision cycle is based on WdgMMode Supervision Cycle and it is defined as a multiple of this value (Supervision Reference Cycle).



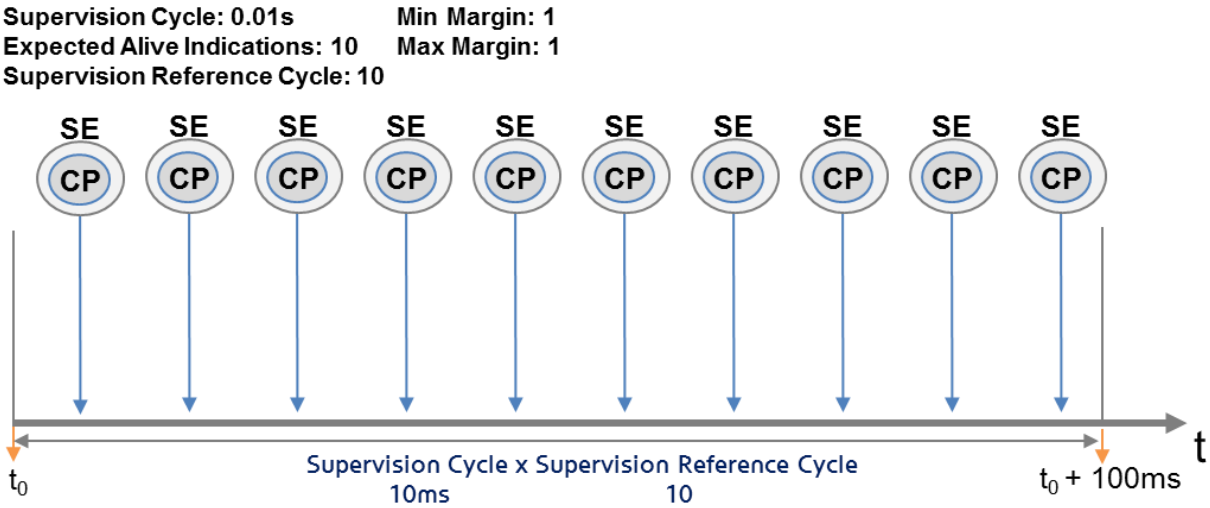
The number of checkpoint execution during the relevant supervision cycle is defined as Expected Alive Indications. The permitted extra number of execution can be defined as Min/Max Margin.



The configuration in the image above is a condition where Indicator needs to occur ten times during supervision cycle 100 ms (periodic execution time 10ms X 10 times). As Min/Max Margin is 1 each, the condition allows Indicator to occur 9 to 11 times.

As for Min/Max Margin parameter, users can set the value after considering a condition where supervised entity can be executed fast or slowly during supervision cycle depending on Application execution timing. If the system must operate in a particular cycle, Min/Max Margin parameter can be set to strict value. On the contrary, if there is no big problem even when the system does not operate in a particular cycle, setting loose Min/Max Margin parameter can prevent unwanted reset. See exercises below for configuration examples.

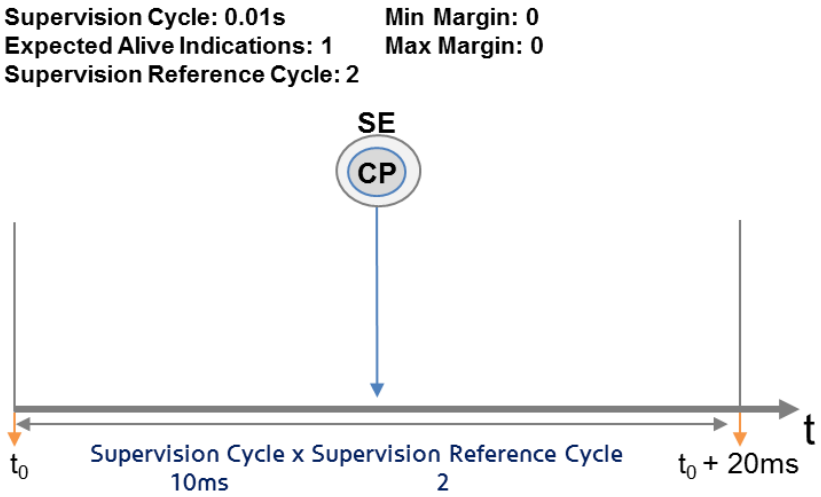
Ex1)
Supervision Cycle(WdgM_MainFunction Cycle) : 0.01 (sec)
Supervision Reference Cycle : 10 (0.1 sec)
Expected Alive Indications : 10
Min Margin : 1
Max Margin : 1
➔ CheckpointReached function with Checkpoint ID as factor configured in Checkpoint Ref needs to be called 9(10-1) - 11(10+1) times per 100ms cycle.



Ex2)

Supervision Cycle(WdgM_MainFunction Cycle) : 0.01 (sec)
Supervision Reference Cycle : 2 (0.02 sec)
Expected Alive Indications : 1
Min Margin : 0
Max Margin : 0

➔ CheckpointReached function with Checkpoint ID as factor configured in Checkpoint Ref needs to be called once per 20ms cycle.

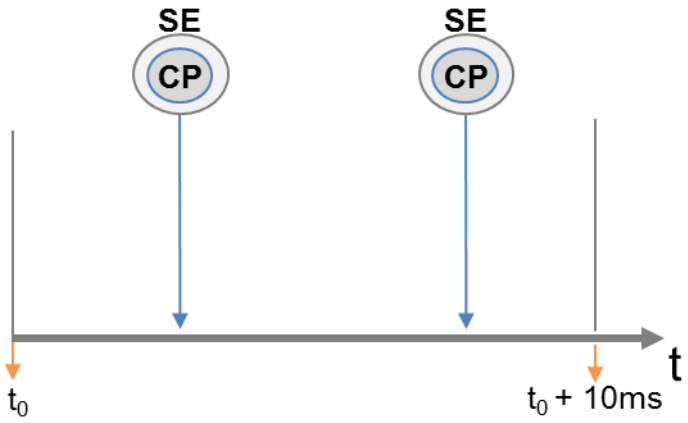


Ex3)

Supervision Cycle(WdgM_MainFunction Cycle) : 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 as factor configured in Checkpoint Ref needs to be called 2 ~ 3(2+1) times per 10ms cycle.

Supervision Cycle: 0.01s **Min Margin: 0**
Expected Alive Indications: 2 **Max Margin: 1**
Supervision Reference Cycle: 1



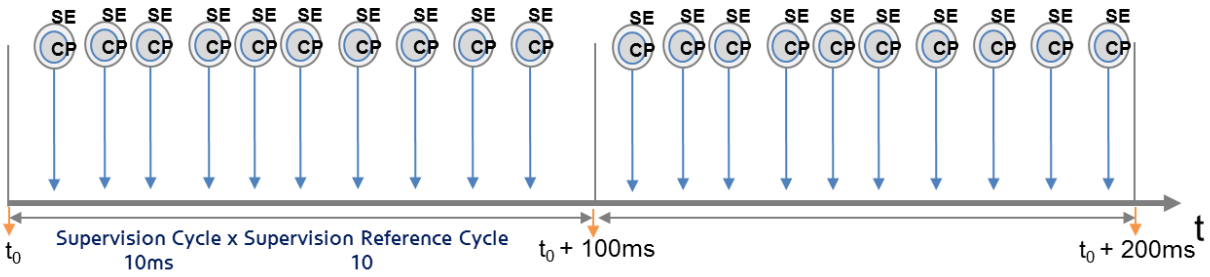
Ex4)

Supervision Cycle(WdgM_MainFunction Cycle) : 0.01 (sec)
* Expired Supervision Cycle Tol: 100 (* See 5.2.2 Wdg Mode.)
Supervision Reference Cycle : 10
Expected Alive Indications : 10
Min Margin : 0
Max Margin : 0

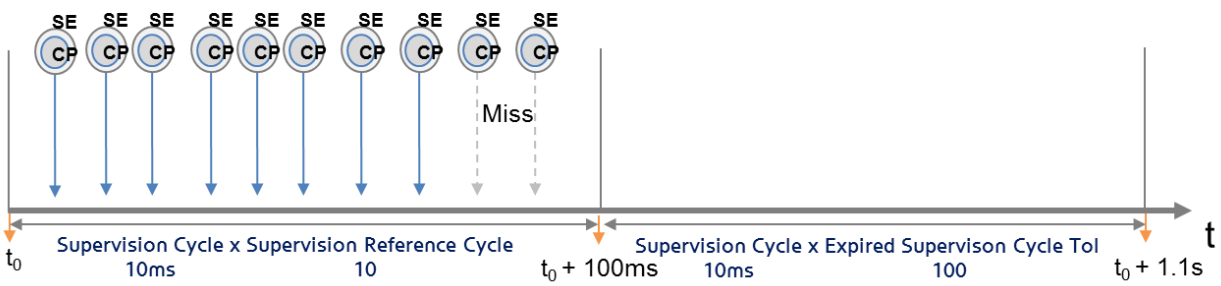
➔ CheckpointReached function with Checkpoint ID as factor configured in Checkpoint Ref needs to be called 10 times per 100ms cycle. If failed, status information is sent to Wdg driver after 1s.

Supervision Cycle: 0.01s **Min Margin: 0**
Expected Alive Indications: 10 **Max Margin: 0**
Supervision Reference Cycle: 10
Expired Supervision Cycle Tol: 100

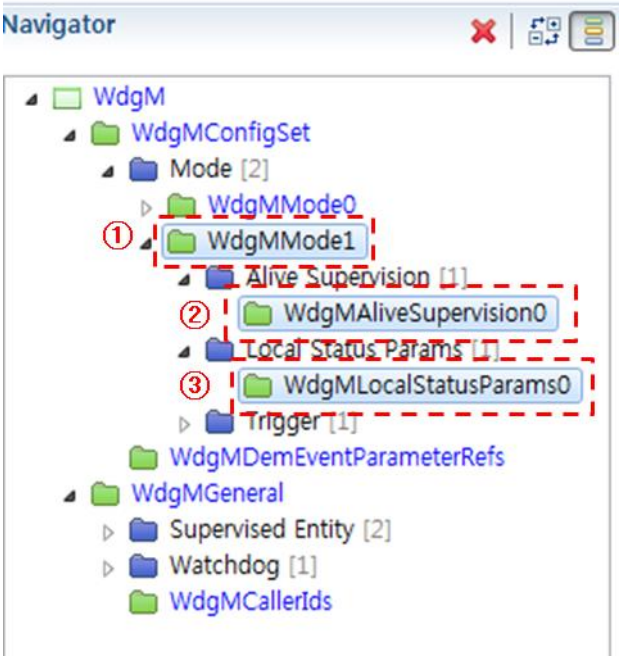
<Normal Condition>

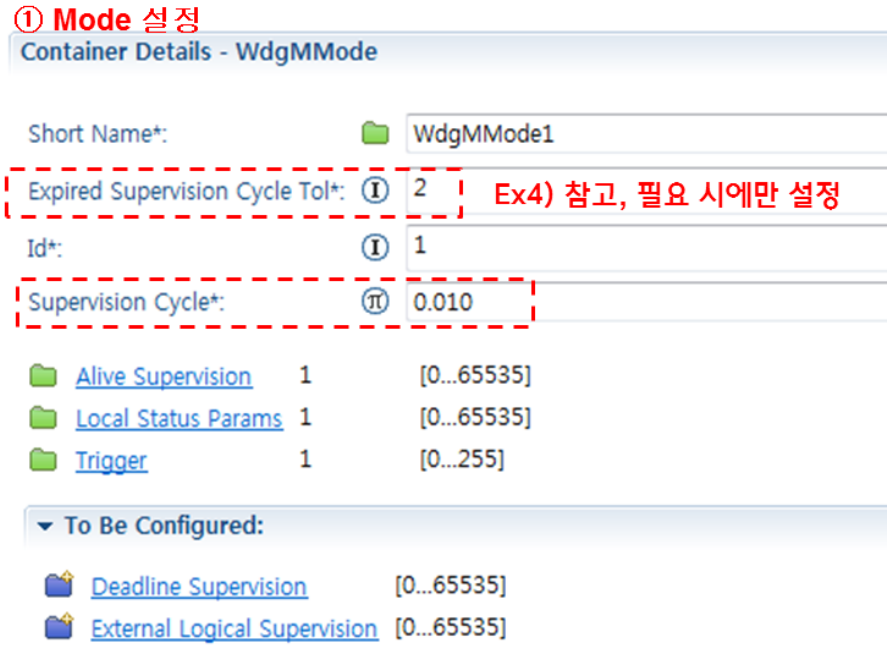


<Reset Condition>

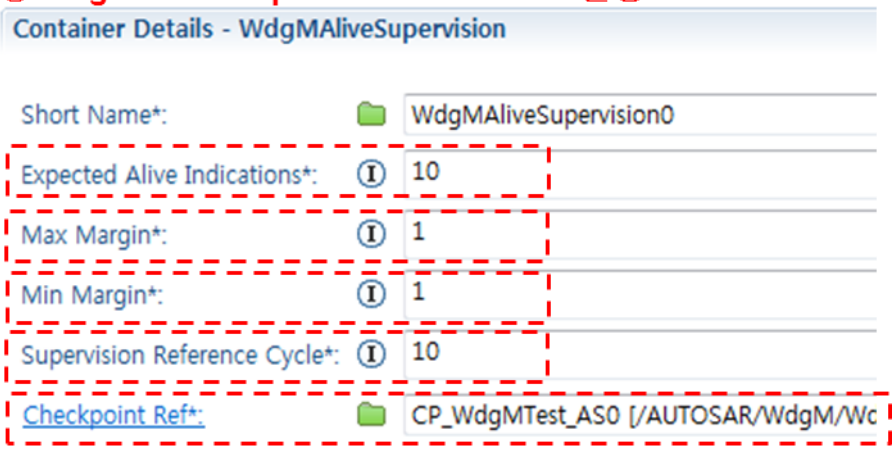


Set the configuration details of Watchdog Mode in the example above for the relevant WdgM Mode, Alive Supervision Container properties, and Local Status Params Container properties.
(* See 5.2.5.2 WdgMLocalStatusParams.)

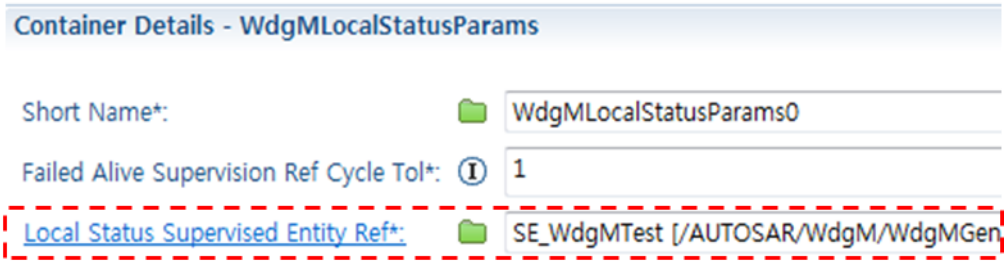




② WdgM Alive Supervision Container 설정



③ WdgM Local Status Parameters 설정



9.2.3 Deadline Supervision

Deadline Supervision defines two checkpoints for supervision and determines the execution time range between the two points (minimum execution time <= actual execution time <= maximum execution time). Defined details are configured in Deadline Supervision properties of WdgMMode and Local Status Params.

See exercises below for configuration examples.

Ex1)

Deadline Min : 0.0

Deadline Max : 0.025

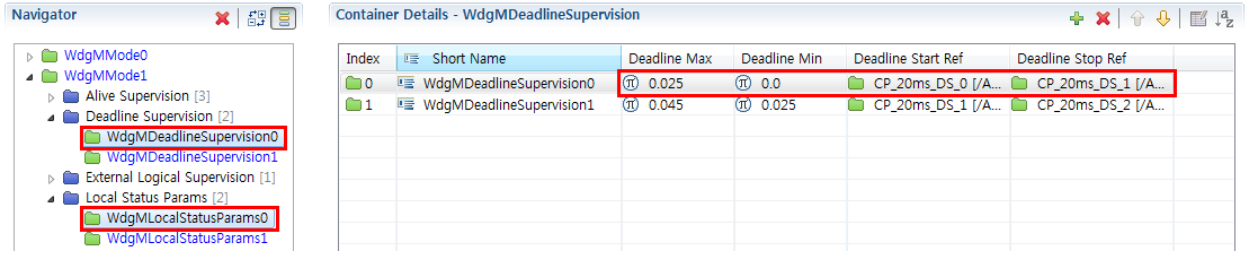
➔ After CheckpointReached function with Checkpoint ID set in Deadline Start Ref as factor is called, CheckpointReached function with Checkpoint ID set in Deadline Stop Ref as factor needs to be called within 25ms.

Ex2)

Deadline Min : 0.025

Deadline Max : 0.045

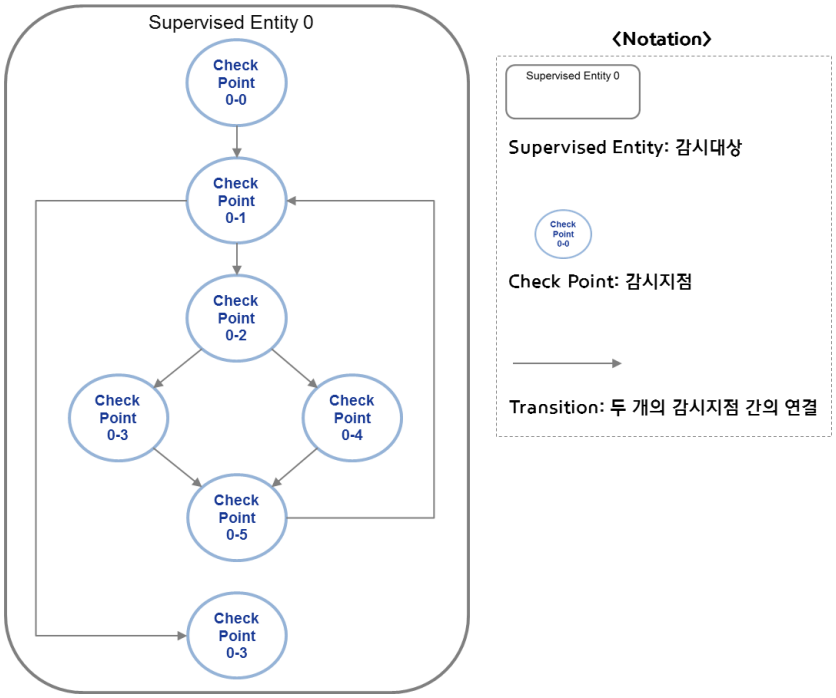
➔ After CheckpointReached function with Checkpoint ID set in Deadline Start Ref as factor is called, CheckpointReached function with Checkpoint ID set in Deadline Stop Ref as factor needs to be called within 45ms after 25ms.



9.2.4 Logical Supervision

Logical Supervision is a basic technique to ensure the right execution of embedded system software, and details of safety standards requiring Logical Supervision (ISO26262 or IEC61508) need to be confirmed.

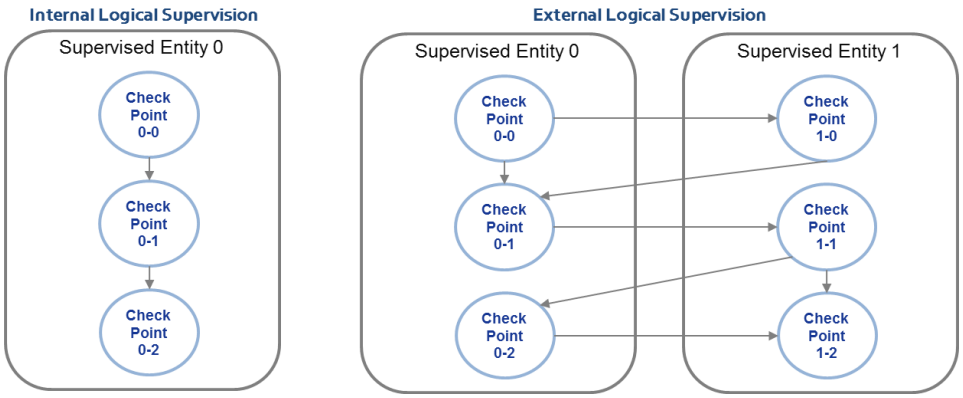
Logical Supervision focuses on Control Flow Error diverged from effective (i.e. coded/compiled) program sequences during Error-Free application program execution. Incorrect control flow causes more than one program command not to be processed or executed in the wrong order. Control flow errors may lead to data damage, micro controller resets, Fail-silence violations, etc. [1]
AUTOSAR_SWS_WatchdogManager



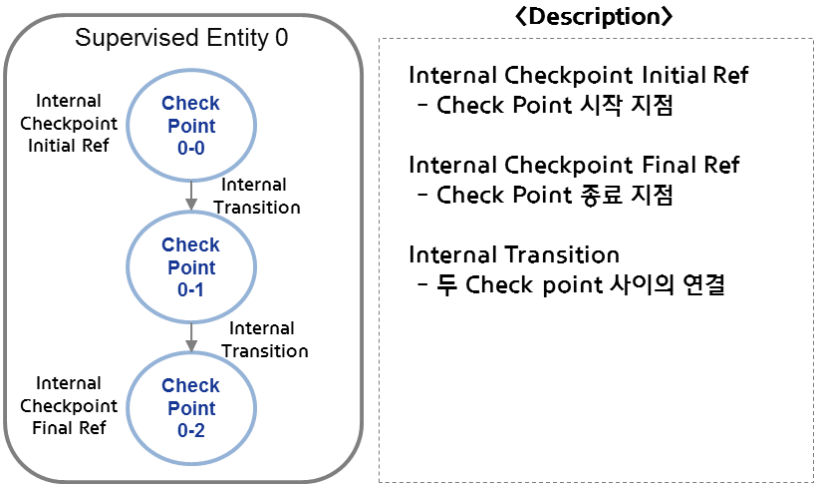
In summary, logical supervision defines transition between checkpoints of supervised entity and supervises the execution order of the transition.

Configuration method of Logical Supervision varies according to whether checkpoints for supervision exist in only one supervised entity or two or more supervised entities.

Internal Logical Supervision monitors control flow of checkpoints in one supervised entity.
External Logical Supervision monitors control flow between checkpoints of two or more supervised entities.

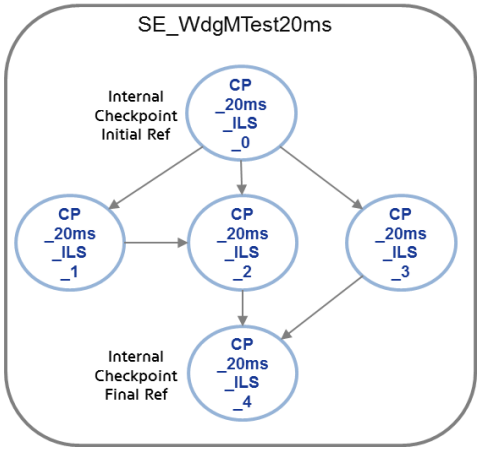


The image below shows a case where checkpoints belong to one supervised entity. The execution order of supervision is illustrated with a graph as below, and Internal Checkpoint Initial Ref, Internal Checkpoint Final Ref and Internal Transition are defined. Defined details are configured in Supervised Entity properties.

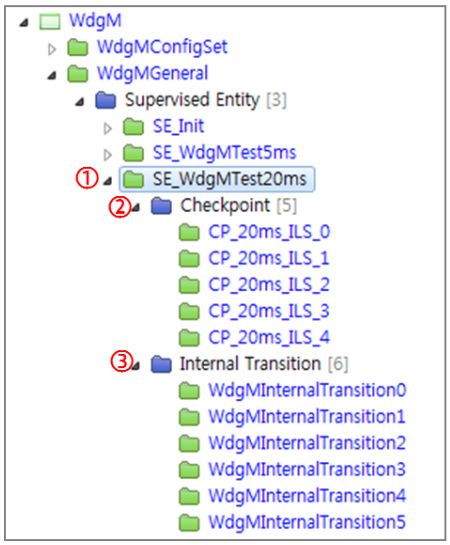


[Example of Internal Logical Supervision Configuration]

1) Control Flow Graph



2) WdgM Configuration



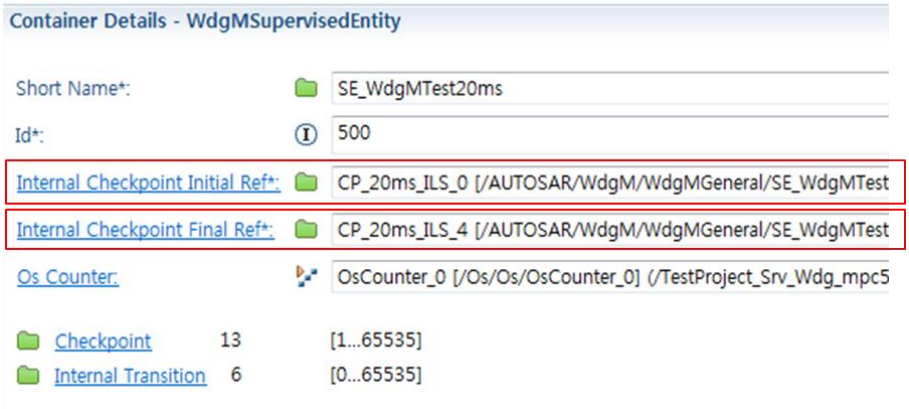
〈Description〉

①: 감시 대상 설정
- 기존에 설정된 감시대상 사용해도 무관함
- 별도 감시대상이 없다면 생성 필요
- Internal Checkpoint Initial Ref 설정
- Internal Checkpoint Final Ref 설정

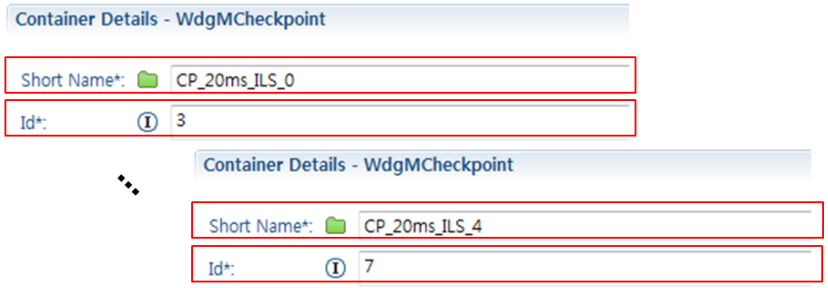
②: 감시 지점 생성
- Control Flow Graph를 기준으로 생성
- 해당 감시 지점의 ID 설정

③: 감시 지점 연결 생성
- Source Ref 설정 (시작 지점)
- Dest Ref 설정 (종료 지점)

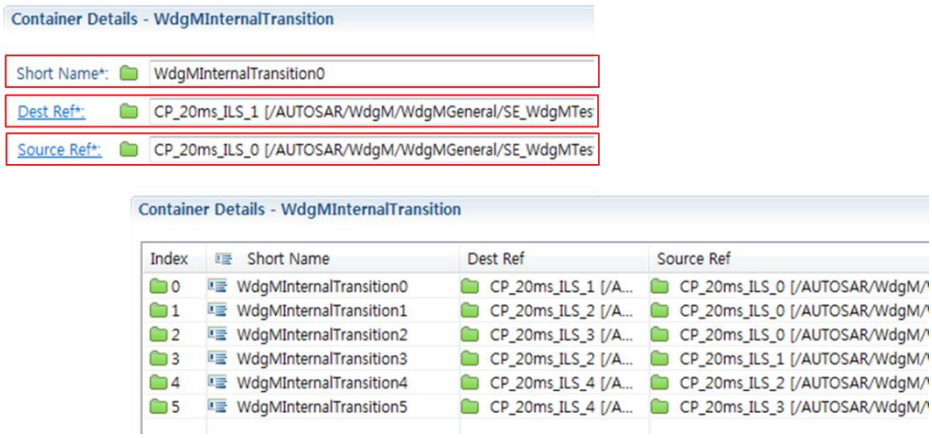
① 감시대상 설정



② 감시지점 설정 – Control Flow Graph에 설정된 모든 감시지점 생성 필요

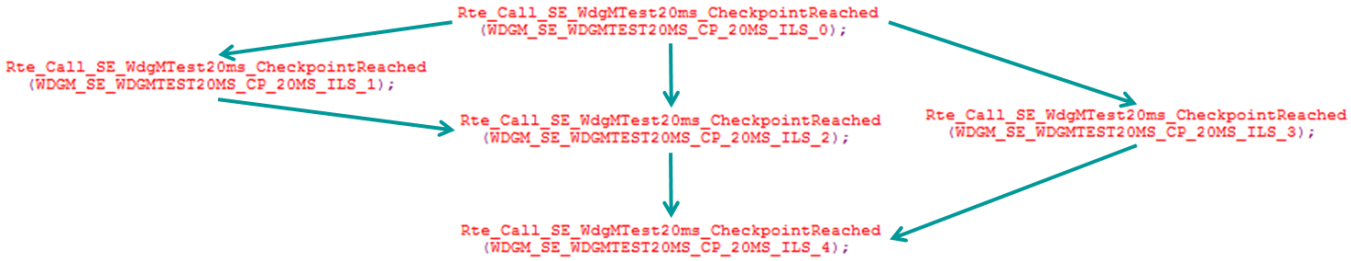


② 감시지점 연결 설정 – Control Flow Graph에 설정된 감시지점 간 연결 설정

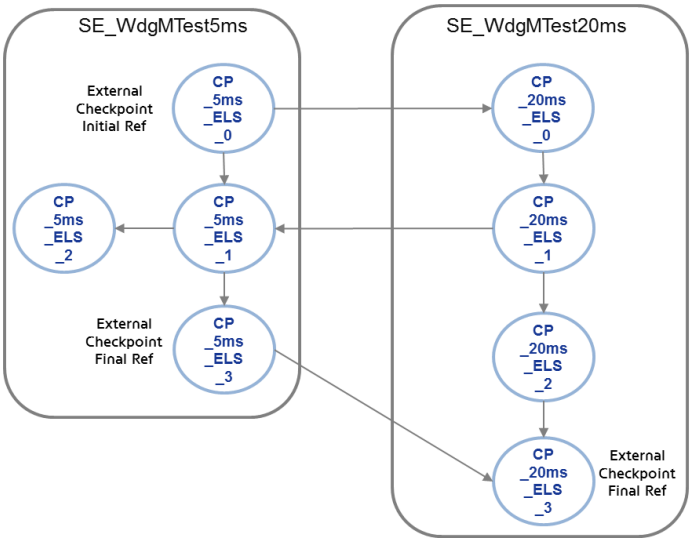


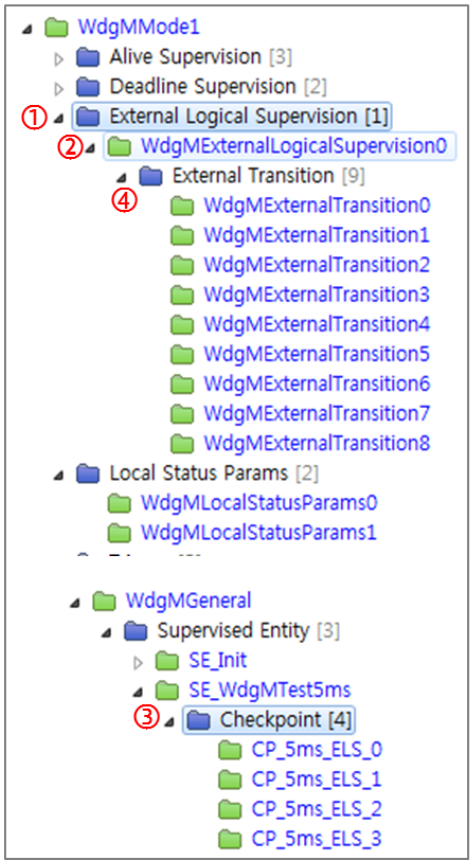
3) CheckpointReached API Call

On the actual codes, CheckpointReached checkpoint ID needs to be called with a factor in the designed checkpoint location as below.



If checkpoints for supervision exist across two or more supervised entities, it is illustrated with a graph as below, and External Checkpoint Initial Ref, Exteral Checkpoint Final Ref and External Transition are defined. Defined details are configured in External Logical Supervision properties of WdgMMode and Local Status Params.





〈Description〉

External Logical Supervision의 경우 필요한 WdgMMode에 추가한다.

①: External Logical Supervision 추가

②: 감시대상 설정
- Internal Checkpoint Initial Ref 설정
- Internal Checkpoint Final Ref 설정

③: 감시 지점 생성
- Control Flow Graph를 기준으로 생성
- 해당 감시 지점의 ID 설정

④: 감시 지점 연결 생성
- Source Ref 설정 (시작 지점)
- Dest Ref 설정 (종료 지점)

On the actual codes, CheckpointReached checkpoint ID needs to be called with a factor in the designed checkpoint location.

9.3 Cautions during Design

* The user needs to see 9.2 and set each supervision technique of WdgM according to the system. As the time necessary to detect an error and correct it varies, the parameters of WdgM supervision technique need to be set after referring to 9.2.

9.3.1 Alive Supervision

1) As for WdgMSupervisionReferenceCycle of Alive Supervision, when setting too large a value for alive supervision normal/abnormal decision cycle, it could take too much time from error occurrence to the actual reset.
2) WdgM determines the number of checkpoint arrival events per WdgMSupervisionReferenceCycle through WdgMExpectedAliveIndications value of alive supervision and WdgMMaxMargin/WdgMMinMargin. Unnecessary reset may occur if WdgMMaxMargin/WdgMMinMargin value is too small, but errors may persist if the configuration is too big. Thus the configuration needs to be modified according to system characteristics.

9.3.2 Deadline Supervision

1) In case of Checkpoint arrival event that belongs to Deadline Supervision configuration
   A. Ignore if end checkpoint reached when start checkpoint has not reached.
   B. Use checkpoint arrival event timestamp that reached later if the same start checkpoint reached again when start checkpoint has already reached.

9.3.3 Logical Supervision

1) In case of Checkpoint arrival event that belongs to Internal / External Graph
   A. Fail if another checkpoint (middle/end) that belongs to Graph reached when start checkpoint has not reached.
2) Execution time of CheckpointReached API may increase if Transition of Logical Supervision increases.

9.3.4 Mode Switch Event

1) Change in Local Supervision Status or Global Supervision Status of Supervised Entity can be sent through Mode switch event.
   A. The problem of multiple activation may arise if mode switch event and mapped task are

overlapped.

9.3.5 WdgM_DeInit

1) Wdg_SetTriggerCondition is not called after WdgM DeInit. Thus an adequate timeout value needs to be set to prevent reset by Wdg that the platform does not aim.
2) Timeout value needs to be set within a maximum of 4s in consideration of NVM Write All time.

9.3.6 Std_ReturnType

1) As N_OK value may be returned if using Std_Return during WdgM API, this needs to be put into adequate consideration during design.

9.3.7 Low Power Mode

1) During Low Power

   A. WdgM module operation is suspended while ECU is in sleep mode. Thus watchdog triggering needs to be executed periodically before hardware watchdog timeout if hardware watchdog keeps operating in sleep mode. Watchdog triggering in sleep mode needs to be reflected in design by referring to EcuM User Manual.

   B. Startup time needs to be included in watchdog timeout if wakeup reset occurs according to MCU. Thus an adequate watchdog timeout time needs to be set in consideration of startup time.

   C. Watchdog triggering cycle affects current consumption. If watchdog timeout is too short, current consumption increases because watchdog triggering needs to be conducted in a short interval. Watchdog timeout time and watchdog triggering cycle need to be set in consideration of this matter.

2) Low Power Transition

   A. In the case of FREESCALE MCU, fixed Watchdog Timeout/Window period (100ms) and GPT Disable used by Internal Watchdog are included in Transition to prevent unintended watchdog reset in High to Low Transition.

   B. When using External Watchdog, GPT Disable used by External Watchdog in High to Low Transition is included in Transition.

   C. Internal / External Watchdog Triggering is executed once at the last phase of Transition.

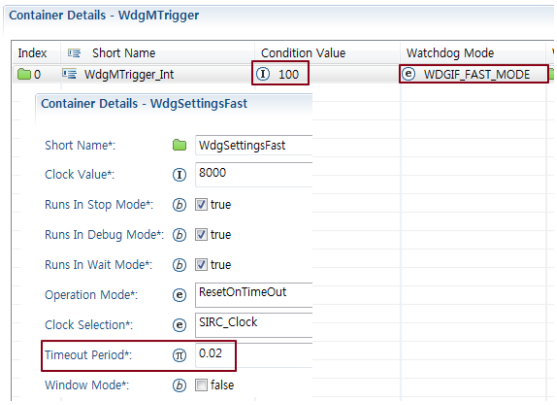
D. In the case of FREESCALE, INFINEON MCU, it is changed to Watchdog Mode before Low Power entry by calling WdgM_PmModeChange() API in Low to High Transition.

E. During application design, using GPT in Low Power Transition and Watchdog Trigger restrictions need to be considered and reflected in the design.

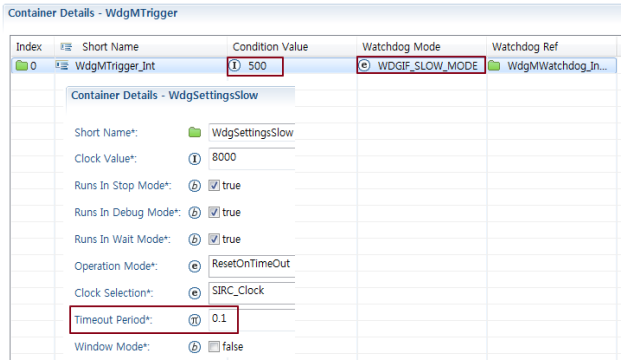
9.3.8 Condition Value

1) Condition value needs to be bigger than Fast mode / Slow mode timeout value according to Wdg operation modes.

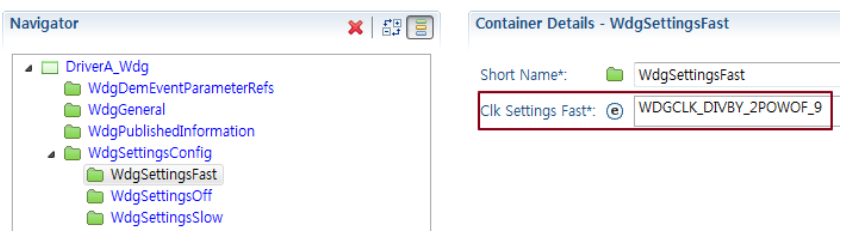
A. If Wdg mode is the fast mode, set condition value to a value bigger than timeout in the fast mode of Wdg module as below.



B. If Wdg mode is the slow mode, set condition value to a value bigger than timeout in the slow mode of Wdg module as below.



C. For some MCU, set Wdg timeout value to a value other than a time unit. For example, timeout value for RH850 can be set through configuration in the image below, and timeout value by the time unit can be estimated through the formula below.



$$\text{Timeout (ms)} = \frac{2^n}{\text{McuAwoWdta}} * 1000 \ (WDGCLK\_DIVBY\_2POWOF\_n \ where \ n = 9 \ to \ 16)$$

Thus set condition value after estimating Wdg timeout by referring to the relevant MCU user manual.

9.4 How to Check Wdg Reset Reason

How to check whether reset is caused by Wdg after reset occurs

9.4.1 Freescale Wdg

T32 development environment standard

1) How to check EcuM variable
   A. After reset occurrence
      i. In case of start address break
         Proceed with Step B below after T32 Go.
      ii. If the target has no running with T32
         Proceed with Step B below after T32 Attach.
   B. Execute Var.Watch EcuM_GddResetReason.
   C. Check EcuM_GddResetReason variable value MCU_WATCHDOG_RESET.
2) How to Check Wdg Reset Callback
   A. Configuration change and build
      i. Change Operation Mode of Ecud_Wdg.arxml / WdgSettingsConfig / WdgSettingsSlow, WdgSettingsFast containers from ResetOnTimeOut to Interrupt.
      ii. Add Isr to Ecud_Os.arxml / Isr as follows and register to Application / AppIsrRef.
         ShortName : Wdg_Isr
         Category: CATEGORY_2
         Priority : 1
         IrqChannel / EXTERNAL / IrqNumber : 28
      iii. Platform build
   B. After running T32 and loading elf
      i. Break.Set RE_Wdg_Cbk
      ii. Reproduce HW Wdg reset.
      iii. Check RE_Wdg_Cbk callout.

9.4.2 Infenion Wdg

T32 development environment standard

1) How to check EcuM variable
   A. After reset occurrence
      i. In case of start address break
         Proceed with Step B below after T32 Go.
      ii. If the target has no running with T32
         Proceed with Step B below after T32 Attach.
   B. Execute Var.Watch EcuM_GddResetReason.
   C. Check EcuM_GddResetReason variable value MCU_SMU_RESET.
2) How to check Register
   A. Check target SYStem.Up status after reset occurrence.
   B. Check items of Peripheral → SCU(System Control Unit) → Reset Operation.
   C. Check SMU flag value “Requested” of SCU_RSTSTAT (Reset Status Register) register.

9.4.3 Renesas Wdg

T32 development environment standard

1) HW Wdg Enable/Disable application (Option Byte application)

A. Apply HW Wdg Enable/Disable by applying cmm file included in the build folder.

Status is maintained with one application (limited to the development stage).

i. RH850_EW.cmm : cmm for HW Wdg Enable

ii. RH850_DW.cmm : cmm for HW Wdg Disable

B. RH850 sets HW Watchdog Enable/Disable by using Option Byte.

HW Watchdog is set by Option Byte using T32 in the development stage, and when mass-producing, additional Option Byte configuration is required according to the Renesas guide in the mass-production stage.

(Consulting Renesas is required in case of mass-production.)

OptionByte configuration set by Cmm script is as below.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTA0.Enable	1	1	1	1	1	0	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1
WDTA0.Disable	1	1	1	1	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1

Start-Up Option	Function	Description	Option Byte
OPWDEN	WDTA setting	Enables/disables the WDTA: 0: WDTA is disabled 1: WDTA is enabled	• WDTA0: OPBT0.OPBT0[19] • WDTA1: OPBT0.OPBT0[23]
OPWDOVF[2:0]	Overflow interval time reset value setting	Specifies the reset value of the overflow interval time control bits WDTAnMD.WDTAnOVF[2:0].	• WDTA0/WDTA1: OPBT0.OPBT0[18:16]
OPWDRUN	Start mode setting	Specifies the start mode: 0: Software trigger start mode 1: Default start mode For details, see **Section 20.5.1, WDTA after Reset Release**.	• WDTA0: OPBT0.OPBT0[20] • WDTA1: OPBT0.OPBT0[24]
OPWDVAC	Variable activation code selection	Specifies the trigger register for the generation of counter re-start triggers to keep the counter from overflowing. 0: WDTAnWDTE (fixed) 1: WDTAnEVAC (variable) When WDTAnWDTE is selected, the value to be written to the register (activation code) is fixed (ACH). When WDTAnEVAC is selected, the activation code to be written to the register is variable. For details, see **Section 20.5.2, WDTA Trigger** and **20.5.2.1, Calculating an Activation Code when the VAC Function is Used**.	• WDTA0: OPBT0.OPBT0[22] • WDTA1: OPBT0.OPBT0[26]

2) How to check EcuM variable

A. After reset occurrence

i. In case of start address break

Proceed with Step B below after T32 Go.

ii. If the target has no running with T32

Proceed with Step B below after T32 Attach.

B. Execute Var.Watch EcuM_GddResetReason.

C. Check EcuM_GddResetReason variable value MCU_WATCHDOG0_RESET.

3) How to check Register

A. T32 Attach after reset occurrence

B. Check items of Peripheral → Reset Controller → Redundant reset.

1) Check RESFR1 flag value “Occurred” of RESFR (Redundant reset source register) register.