User Manual

for S32K1 WDG Driver

Document Number: UM2WDGASR4.4 Rev0000R1.0.1 Rev. 1.0

1 Revision History 2

2 Introduction 3

2.1 Supported Derivatives . . . 3
2.2 Overview . . . 4
2.3 About This Manual . . . 5
2.4 Acronyms and Definitions . . . 6
2.5 Reference List . . . 6

3 Driver 7

3.1 Requirements . . . 7
3.2 Driver Design Summary . . . 7
3.3 Hardware Resources . . . 8
3.4 Deviations from Requirements . . . 8
3.5 Driver Limitations . . . 10
3.6 Driver usage and configuration tips . . . 11
3.7 Runtime errors . . . 11
3.8 Symbolic Names Disclaimer . . . 12

4 Tresos Configuration Plug-in 13

4.1 Module Wdg . . . 15
4.2 Container WdgDemEventParameterRefs . . . 15
4.3 Reference WDG_E_DISABLE_REJECTED . . . 16
4.4 Reference WDG_E_MODE_FAILED . . . 16
4.5 Container WdgGeneral . . . 17
4.6 Parameter WdgDevErrorDetect . . . 17
4.7 Parameter WdgDisableAllowed . . . 18
4.8 Parameter WdgEnableUserModeSupport . . . 18
4.9 Parameter WdgEnableDirectService . . . 19
4.10 Parameter WdgEnableMultiCoreSupport . . . 19
4.11 Parameter WdgTimeoutMethod . . . 20
4.12 Parameter WdgOsifTimeoutVal . . . 21
4.13 Parameter WdgIndex . . . 21
4.14 Parameter WdgInitialTimeout . . . 22
4.15 Parameter WdgMaxTimeout . . . 22
4.16 Parameter WdgRunArea . . . 23
4.17 Parameter WdgTriggerLocation . . . 23
4.18 Parameter WdgCallbackNotification . . . 24
4.19 Parameter WdgVersionInfoApi . . . 24
4.20 Reference WdgEcucPartitionRef . . . 25
4.21 Container AutosarExt . . . 25

4.22 Parameter WdgDisableDemReportErrorStatus . . . 25
4.23 Container WdgClockReferencePoint . . . 26
4.24 Reference WdgClockReference . . . 27
4.25 Container WdgSettingsConfig . . . 27
4.26 Parameter WdgInstance . . . 28
4.27 Parameter WdgDefaultMode . . . 28
4.28 Parameter WdgInterruptContentEnable . . . 29
4.29 Reference WdgExternalTriggerCounterRef . . . 29
4.30 Container WdgExternalConfiguration . . . 30
4.31 Reference WdgExternalContainerRef . . . 30
4.32 Container WdgSettingsFast . . . 31
4.33 Parameter WdgClockValue . . . 31
4.34 Parameter WdgRunsInStopMode . . . 32
4.35 Parameter WdgRunsInDebugMode . . . 32
4.36 Parameter WdgRunsInWaitMode . . . 33
4.37 Parameter WdgOperationMode . . . 33
4.38 Parameter WdgClockSelection . . . 34
4.39 Parameter WdgTimeoutPeriod . . . 35
4.40 Parameter WdgWindowMode . . . 35
4.41 Parameter WdgWindowPeriod . . . 36
4.42 Parameter WdgPrescalerEnabled . . . 36
4.43 Parameter WdgAllowUpdates . . . 37
4.44 Reference WdgClkSrcRef . . . 38
4.45 Container WdgSettingsOff . . . 38
4.46 Parameter WdgAllowUpdates . . . 38
4.47 Container WdgSettingsSlow . . . 39
4.48 Parameter WdgClockValue . . . 39
4.49 Parameter WdgRunsInStopMode . . . 41
4.50 Parameter WdgRunsInDebugMode . . . 42
4.51 Parameter WdgRunsInWaitMode . . . 42
4.52 Parameter WdgOperationMode . . . 43
4.53 Parameter WdgClockSelection . . . 43
4.54 Parameter WdgTimeoutPeriod . . . 44
4.55 Parameter WdgWindowMode . . . 45
4.56 Parameter WdgWindowPeriod . . . 45
4.57 Parameter WdgPrescalerEnabled . . . 46
4.58 Parameter WdgAllowUpdates . . . 46
4.59 Reference WdgClkSrcRef . . . 48
4.60 Container WdgPublishedInformation . . . 48
4.61 Parameter WdgTriggerMode . . . 49

4.62 Container CommonPublishedInformation . . . 49
4.63 Parameter ArReleaseMajorVersion . . . 50
4.64 Parameter ArReleaseMinorVersion . . . 50
4.65 Parameter ArReleaseRevisionVersion . . . 51
4.66 Parameter ModuleId . . . 51
4.67 Parameter SwMajorVersion . . . 52
4.68 Parameter SwMinorVersion . . . 52
4.69 Parameter SwPatchVersion . . . 53
4.70 Parameter VendorApiInfix . . . 53
4.71 Parameter VendorId . . . 54

5 Module Index **55**
5.1 Software Specification . . . 55

6 Module Documentation **56**
6.1 Ewm_Ip . . . 56
6.1.1 Detailed Description . . . 56
6.1.2 Data Structure Documentation . . . 57
6.1.3 Types Reference . . . 57
6.1.4 Enum Reference . . . 57
6.1.5 Function Reference . . . 58
6.2 Wdg . . . 60
6.2.1 Detailed Description . . . 60
6.2.2 Data Structure Documentation . . . 60
6.2.3 Enum Reference . . . 62
6.2.4 Function Reference . . . 63
6.3 Wdog_Ip . . . 66
6.3.1 Detailed Description . . . 66
6.3.2 Data Structure Documentation . . . 67
6.3.3 Types Reference . . . 67
6.3.4 Enum Reference . . . 68
6.3.5 Function Reference . . . 69

Chapter 1

Revision History

Revision	Date	Author	Description
1.0	24.02.2022	NXP RTD Team	Prepared for release RTD S32K1 Version 1.0.1

Chapter 2

Introduction

- Supported Derivatives
- Overview
- About This Manual
- Acronyms and Definitions
- Reference List

This User Manual describes NXP Semiconductor AUTOSAR *Watchdog (Wdg)* for *S32K1*. AUTOSAR *Wdg* driver configuration parameters and deviations from the specification are described in Driver chapter of this document. AUTOSAR *Wdg* driver requirements and APIs are described in the AUTOSAR *Wdg* driver software specification document.

2.1 Supported Derivatives

The software described in this document is intended to be used with the following microcontroller devices of NXP Semiconductors:

- s32k116_qfn32
- s32k116_lqfp48
- s32k118_lqfp48
- s32k118_lqfp64
- s32k142_lqfp48
- s32k142_lqfp64
- s32k142_lqfp100
- s32k142w_lqfp48
- s32k142w_lqfp64
- s32k144_lqfp48

- s32k144_lqfp64
- s32k144_lqfp100
- s32k144_mapbga100
- s32k144w_lqfp48
- s32k144w_lqfp64
- s32k146_lqfp64
- s32k146_lqfp100
- s32k146_mapbga100
- s32k146_lqfp144
- s32k148_lqfp100
- s32k148_mapbga100
- s32k148_lqfp144
- s32k148_lqfp176

All of the above microcontroller devices are collectively named as S32K1.

2.2 Overview

AUTOSAR (AUTomotive Open System ARchitecture) is an industry partnership working to establish standards for software interfaces and software modules for automobile electronic control systems.

AUTOSAR:

- paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.
- is a strong global partnership that creates one common standard: "Cooperate on standards, compete on implementation".
- is a key enabling technology to manage the growing electrics/electronics complexity. It aims to be prepared for the upcoming technologies and to improve cost-efficiency without making any compromise with respect to quality.
- facilitates the exchange and update of software and hardware over the service life of the vehicle.

2.3 About This Manual

This Technical Reference employs the following typographical conventions:

- **Boldface** style: Used for important terms, notes and warnings.
- *Italic* style: Used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

Note

This is a note.

Warning

This is a warning

2.4 Acronyms and Definitions

Term	Definition
API	Application Programming Interface
ASM	Assembler
BSMI	Basic Software Make file Interface
C/CPP	C and C++ Source Code
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DMA	Direct Memory Access
ECU	Electronic Control Unit
LSB	Least Signifigant Bit
MCU	Micro Controller Unit
MIDE	Multi Integrated Development Environment
MSB	Most Significant Bit
RAM	Random Access Memory
SIU	Systems Integration Unit
SWS	Software Specification
VLE	Variable Length Encoding
XML	Extensible Markup Language
EWM	External Watchdog Monitor
WDOG	Watchdog Timer

2.5 Reference List

#	Title	Version
1	S32K1XX Series Reference Manual	Rev. 14, 09/2021
2	Errata S32K116_0N96V	Rev. 22/OCT/2021
3	Errata S32K118_0N97V	Rev. 22/OCT/2021
4	Errata S32K142_0N33V	Rev. 22/OCT/2021
5	Errata S32K144_0N57U	Rev. 22/OCT/2021
6	Errata S32K144W_0P64A	Rev. 22/OCT/2021
7	Errata S32K146_0N73V	Rev. 22/OCT/2021
8	Errata S32K148_0N20V	Rev. 22/OCT/2021
9	S32K1xx Data Sheet	Rev. 14, 08/2021

Chapter 3

Driver

- Requirements
- Driver Design Summary
- Hardware Resources
- Deviations from Requirements
- Driver Limitations
- Driver usage and configuration tips
- Runtime errors
- Symbolic Names Disclaimer

3.1 Requirements

Requirements for this driver are detailed in the Autosar Driver Software Specification document (See Table Reference List).

3.2 Driver Design Summary

The Watchdog Timer(WDOG) and External watchdog monitor(EWM) with programmable interrupt response are available in S32K1:
The Watchdog Timer (WDOG) is a peripheral module that can prevent system lockup in situations such as software getting trapped in a loop or if a bus transaction fails to terminate. When enabled, the WDOG requires periodic execution of a watchdog servicing operation. The servicing operation resets the timer to a specified time-out period. If this servicing action does not occur before the timer expires the WDOG generates an interrupt or hardware reset. The WDOG can be configured to generate a reset or interrupt on an initial time-out. However the WDOG reset is always generated after RCM delay in LPO cycles depending on configuring the RCM_SRIE[DELAY] and RCM_SRIE[WDOG] bits.
In addition to these two modes of operation, the watchdog timer also supports a windowed mode. In this mode, the service sequence must be performed in the last part of the time-out period defined by the window register. The window is open when the down counter is less than the value in the WDOG_WIN register. Outside of this window,

service sequence writes that the Watchdog will reset the MCU.

The WDOG has the following features:
- 16-bit time-out register to set the time-out period
- Programmable selection of window mode or regular servicing
- Programmable selection of reset or interrupt on an initial time-out
- Programmable selection of test mode or user mode

The External Watchdog Monitor (EWM) is designed to monitor external circuits, as well as the microcontroller software flow. This provides aback-up mechanism to the internal watchdog that resets the microcontroller's CPU and peripherals. The overflow of the watchdog counter must not occur if the software code works well and services the watchdog to re-start the actual counter. The EWM differences from the internal watchdog in that it does not reset the microcontroller's CPU and peripherals. The EWM provides an independent EWM_OUT_b signal that when asserted resets or places an external circuit into a safe mode. The EWM_OUT_b signal is asserted upon the EWM counter time-out. An optional external input EWM_in is provided to allow additional control of the assertion of EWM_OUT_b signal actual counter.

The EWM has the following features:
- 8-bit time-out register to set the time-out period
- Programmable selection of window mode or regular servicing
- Programmable selection of reset or interrupt on an initial time-out
- Programmable selection of the logic level voltage of the input pin if enabled and when is asserted

3.3 Hardware Resources

The WDG driver uses the WDOG and EWM hardware IPs.

3.4 Deviations from Requirements

The driver deviates from the AUTOSAR WDG Driver software specification in some places. The table below identifies the AUTOSAR requirements that are not implemented or out of scope for the WDG Driver.

Term	Definition
N/S	Out of scope
N/I	Not implemented
N/F	Not fully implemented

Below table identifies the AUTOSAR requirements that are not fully implemented, implemented differently or out of scope for the WDG driver.

Requirement	Status	Description	Notes
SWS_Wdg_00055	N/S	The Wdg module for an external watchdog driver shall have source code that is independent of the microcontroller platform.	This requirement is rejected because it refers to External Watchdog modules, which are not in scope for the Wdg R↩ TD module. External watchdog must be supported separately
SWS_Wdg_00034	N/S	The start address of the watchdog trigger routine shall be statically configurable to a fixed memory location by the user. The user needs to take care thatConfigured memory location is valid for the platform on which driver is being implemented on.This configuration parameter shall only be given if supported/needed by the hardware.	Rejection Reason: N/A for any of the available platforms
SWS_Wdg_00076	N/S	To access the external watchdog hardware, the corresponding Wdg module instance shall use the functionality and API of the corresponding handler or driver, e.g. the SPI handler or DIO driver.	This requirement is rejected because it refers to External Watchdog modules, which are not in scope for the Wdg R↩ TD module. External watchdog must be supported separately
SWS_Wdg_00162	N/S	The routine servicing an external watchdog shall be implemented by usage of an own internal hardware timer to be independent from other peripherals or by using a GPT driver callback	This requirement is rejected because it refers to External Watchdog modules, which are not in scope for the Wdg R↩ TD module. External watchdog must be supported separately
SWS_Wdg_00077	N/S	A Wdg module for an external watchdog shall satisfy the same functional requirements and offer the same functional scope as a Wdg module for an internal watchdog. Hence their respective APIs are semantically identical.	This requirement is rejected because it refers to External Watchdog modules, which are not in scope for the Wdg R↩ TD module. External watchdog must be supported separately
SWS_Wdg_00078	N/S	The Wdg module shall add all parameters required for accessing the external watchdog hardware, e.g. the used SPI channel or DIO port, to the module's published parameters and to the module's configuration parameters.	This requirement is rejected because it refers to External Watchdog modules, which are not in scope for the Wdg R↩ TD module. External watchdog must be supported separately
SWS_Wdg_00172	N/S	If more than one watchdog driver instance exits on an ECU (namely an external and an internal one) the A↩ PI names and instance specific type names specified in this chapter shall be made unique by expansion according to SRS_BSW_00347.	This requirement is rejected because it refers to External Watchdog modules, which are not in scope for the Wdg R↩ TD module. External watchdog must be supported separately
SWS_Wdg_00175	N/S	These requirements are not applicable to this specification.	This requirement defines a list of requirements marked in the standard as Not applicable

Requirement	Status	Description	Notes
ECUC_Wdg_00118	N/S	Name - WdgTriggerLocation - Parent Container - WdgGeneral - Description - Location (memory address) of the watchdog trigger routine. - Multiplicity - 1 - Type - EcucFunction↩ NameDef - Default value - – - max↩ Length - – - minLength - – - regular↩ Expression - – - Post-Build Variant Value - false - Value Configuration Class - Pre-compile time - X - All Variants - Link time - – - - Post-build time - – - - - Scope / Dependency - scope: localdependency: Only relevant if provided by hardware and needed by the system. -	This paramater functionality is replaced by CPR_RTD_00161.wdg.
ECUC_Wdg_00112	N/S	Container Name - WdgExternal↩ Configuration - Description - Configuration items for an external watchdog hardware - Configuration Parameters -	This requirement is rejected because it refers to External Watchdog modules, which are not in scope for the Wdg R↩ TD module. External watchdog must be supported separately
ECUC_Wdg_00113	N/S	Name - WdgExternalContainerRef - Parent Container - WdgExternal↩ Configuration - Description - Reference to eithera DioChannelGroup container in case the hardware watchdog is connected via DIO pinsan Spi↩ SequenceConfiguration container in case the watchdog hardware is accessed via SPI - Multiplicity - 0..1 - Type - Choice reference to [Dio↩ ChannelGroup , SpiSequence] - Post-Build Variant Multiplicity - true - Post-Build Variant Value - true - Multiplicity Configuration Class - Precompile time - X - VARIANT-P↩ RE-COMPILE - Link time - X - VARIANT-LINK-TIME - Post-build time - X - VARIANT-POST-BU↩ ILD - Value Configuration Class - Pre-compile time - X - VARIAN↩ T-PRE-COMPILE - Link time - X - VARIANT-LINK-TIME - Post-build time - X - VARIANT-POST-BUI↩ LD - Scope / Dependency - scope↩ : localdependency: See DIO resp. SPI SWS -	This requirement is rejected because it refers to External Watchdog modules, which are not in scope for the Wdg R↩ TD module. External watchdog must be supported separately

3.5 Driver Limitations

None

3.6 Driver usage and configuration tips

The Wdg driver can function in either Direct Service Mode or Gpt Triggered mode. In Direct Service Mode, the Wdog and Ewm peripherals can be serviced directly, while in Gpt Triggered mode, a callback notification is set up so that the Gpt will periodically trigger the Wdog and Ewm peripherals.

1.Configure the Wdg reference clock from Mcu (see parameter WdgClkSrcRef) according to reference point used by the Wdg peripherals on the platform.
2.In Gpt Triggered Mode, configure Wdg routine used for triggering as a Gpt callback (Wdg_Cbk_GptNotificationX must be configured as a notification callback for the Gpt channel intended for triggering).
3.The interrupt feature of the Wdog IP (and Wdg Instance 0) can be used to delay the moment until the system is reset, but can not prevent or stop it. This allows some time to prepare the application for the incoming reset. More details on how to configure the Wdog Interrupt feature can be found in the Reference Manual.
4.The Wdg driver can be configured to run from RAM or ROM targets from the WdgRunArea option. The RAM target should be used in case the module dependencies are also run from RAM, such in the case of bootloaders. If the module dependencies code is being run from a flash target, the ROM option needs to be used.
5.When configured for window mode, the Wdg can be reset if it is serviced outside of the window period.
6.In Direct Service Mode, the Gpt module does not need to be configured and the user must call Wdg_43_Instance↩ X_Service periodically to service the Wdg in order to avoid the system is reset. The Wdg_InstanceX_SetTrigger↩ Condition function is unavailable in this mode. Direct Service Mode can not be used in parallel with Gpt Triggered mode.
7.When called for WDG instance over EWM peripheral, the function Wdg_ChannelSetMode raises DET (if enabled) and returns error, because EWM peripheral only supports Slow Mode.
8.In Gpt Triggered Mode, the user must ensure the following conditions are met when servicing the Wdg:

- Watchdog timeout period and Trigger Condition must not have similar values because of interrupt clashing.
- Watchdog timeout period must be less than Trigger Condition, in order for the Wdg to be serviced in time.
- To avoid potential clashing of triggering interrupts with the main loop, the user should avoid choosing a hardware trigger that is equal to or equal to multiples of the Servicing Period.
- Trigger Condition must be greater than the Servicing period with which Wdg_SetTriggerCondition is called, in order for the Wdg not to expire.

3.7 Runtime errors

The driver generates the following DEM errors at runtime.

Function	Error Code	Condition triggering the error
Wdg_43_Instance<Number>_Init	WDG_E_DISABLE_REJECTED	Initialization or mode switch failed because it would disable the watchdog has occurred
Wdg_43_Instance<Number>_↩ SetMode	WDG_E_DISABLE_REJECTED	Initialization or mode switch failed because it would disable the watchdog has occurred

Function	Error Code	Condition triggering the error
Wdg_43_Instance<Number>_Init	WDG_E_MODE_FAILED	Setting a watchdog mode failed (during initialization or mode switch) has occurred
Wdg_43_Instance<Number>_↩ SetMode	WDG_E_MODE_FAILED	Setting a watchdog mode failed (during initialization or mode switch) has occurred

3.8 Symbolic Names Disclaimer

All containers having symbolicNameValue set to TRUE in the AUTOSAR schema will generate defines like:

#define <Mip>Conf_<Container_ShortName>_<Container_ID>

For this reason it is forbidden to duplicate the names of such containers across the RTD configurations or to use names that may trigger other compile issues (e.g. match existing `#ifdefs` arguments).

Chapter 4

Tresos Configuration Plug-in

This chapter describes the Tresos configuration plug-in for the driver. All the parameters are described below.

- Module Wdg
  - Container WdgDemEventParameterRefs
    - Reference WDG_E_DISABLE_REJECTED
    - Reference WDG_E_MODE_FAILED
  - Container WdgGeneral
    - Parameter WdgDevErrorDetect
    - Parameter WdgDisableAllowed
    - Parameter WdgEnableUserModeSupport
    - Parameter WdgEnableDirectService
    - Parameter WdgEnableMultiCoreSupport
    - Parameter WdgTimeoutMethod
    - Parameter WdgOsifTimeoutVal
    - Parameter WdgIndex
    - Parameter WdgInitialTimeout
    - Parameter WdgMaxTimeout
    - Parameter WdgRunArea
    - Parameter WdgTriggerLocation
    - Parameter WdgCallbackNotification
    - Parameter WdgVersionInfoApi
    - Reference WdgEcucPartitionRef
    - Container AutosarExt
      - Parameter WdgDisableDemReportErrorStatus
  - Container WdgClockReferencePoint
    - Reference WdgClockReference
  - Container WdgSettingsConfig
    - Parameter WdgInstance
    - Parameter WdgDefaultMode
    - Parameter WdgInterruptContentEnable
    - Reference WdgExternalTriggerCounterRef

    - Container WdgExternalConfiguration
        - Reference WdgExternalContainerRef
    - Container WdgSettingsFast
        - Parameter WdgClockValue
        - Parameter WdgRunsInStopMode
        - Parameter WdgRunsInDebugMode
        - Parameter WdgRunsInWaitMode
        - Parameter WdgOperationMode
        - Parameter WdgClockSelection
        - Parameter WdgTimeoutPeriod
        - Parameter WdgWindowMode
        - Parameter WdgWindowPeriod
        - Parameter WdgPrescalerEnabled
        - Parameter WdgAllowUpdates
        - Reference WdgClkSrcRef
    - Container WdgSettingsOff
        - Parameter WdgAllowUpdates
    - Container WdgSettingsSlow
        - Parameter WdgClockValue
        - Parameter WdgRunsInStopMode
        - Parameter WdgRunsInDebugMode
        - Parameter WdgRunsInWaitMode
        - Parameter WdgOperationMode
        - Parameter WdgClockSelection
        - Parameter WdgTimeoutPeriod
        - Parameter WdgWindowMode
        - Parameter WdgWindowPeriod
        - Parameter WdgPrescalerEnabled
        - Parameter WdgAllowUpdates
        - Reference WdgClkSrcRef
- Container WdgPublishedInformation
    - Parameter WdgTriggerMode
- Container CommonPublishedInformation
    - Parameter ArReleaseMajorVersion
    - Parameter ArReleaseMinorVersion
    - Parameter ArReleaseRevisionVersion
    - Parameter ModuleId
    - Parameter SwMajorVersion
    - Parameter SwMinorVersion
    - Parameter SwPatchVersion
    - Parameter VendorApiInfix
    - Parameter VendorId

4.1 Module Wdg

Wdg

Configuration of the Wdg (Watchdog driver) module

Included containers:

- WdgDemEventParameterRefs
- WdgGeneral
- WdgClockReferencePoint
- WdgSettingsConfig
- WdgPublishedInformation
- CommonPublishedInformation

Property	Value
type	ECUC-MODULE-DEF
lowerMultiplicity	1
upperMultiplicity	Infinite
postBuildVariantSupport	true
supportedConfigVariants	VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

4.2 Container WdgDemEventParameterRefs

Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.3 Reference WDG_E_DISABLE_REJECTED

Reference to the DemEventParameter which shall be issued when the error "Initialization or mode switch failed because it would disable the watchdog" has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.4 Reference WDG_E_MODE_FAILED

Reference to the DemEventParameter which shall be issued when the error "Setting a watchdog mode failed (during initialization or mode switch)" has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.5 Container WdgGeneral

WdgGeneral

All general parameters of the watchdog driver are collected here.

Included subcontainers:

- AutosarExt

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.6 Parameter WdgDevErrorDetect

Wdg Development Error Detect

Compile switch to enable / disable development error detection for this module.

This switch enables the Development error detection for the individual hardware IP associatted with the Wdg Instance.

If at least one Wdg Instance has this switch enabled, the High Level code of the Wdg Driver will have the development error

detection feature enabled for all enabled Wdg Instances.

True: Development error detection enabled

False: Development error detection disabled

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE VARIANT-POST-BUILD: PRE-COMPILE VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.7 Parameter WdgDisableAllowed

Wdg Disable Allowed

Compile switch to allow / forbid disabling the watchdog driver during runtime.

True: Disabling the watchdog driver at runtime is allowed

False:Disabling the watchdog driver at runtime is not allowed

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.8 Parameter WdgEnableUserModeSupport

Wdg Enable User Mode Support

When this parameter is enabled, the Wdg module will adapt to run from User Mode, with the following measure : configuring REG_PROT for Wdg IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 for more information and availability on this platform, please see chapter "User Mode Support" in IM.

True: Wdg module will adapt to run from User Mode.

False:Wdg module will not apdapt to run from User Mode.

If this parameter is not ediatable, that means Wdg driver can run in User Mode.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.9 Parameter WdgEnableDirectService

Wdg Enable Direct Service

When this parameter is enabled, the Wdg module can be serviced directly,

without using an external hardware trigger.

True: Wdg module can be serviced directly.

False:Wdg module can not be serviced directly.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.10 Parameter WdgEnableMultiCoreSupport

Wdg Enable Multi-Core Support

When this parameter is enabled, the Wdg module will adapt to run in Multi-core

True: Wdg module will adapt to run in Multi-core.

False:Wdg module will not apdapt to in Multi-core.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.11 Parameter WdgTimeoutMethod

WdgTimeoutMethod

Configures the timeout method.

Based on this selection a certain timeout method from OsIf will be used in the driver.

Note: If OSIF_COUNTER_SYSTEM or OSIF_COUNTER_CUSTOM are selected make sure the corresponding timer is enabled in OsIf General configuration.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	OSIF_COUNTER_DUMMY
literals	['OSIF_COUNTER_DUMMY', 'OSIF_COUNTER_SYSTEM', 'OSIF_COUNTER_CUSTOM']

4.12 Parameter WdgOsifTimeoutVal

Wdg Osif timeout value.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	3000
max	4294967295
min	0

4.13 Parameter WdgIndex

Wdg Instance 0 Index

Represents the watchdog driver's ID for Instance 0 so that it can be referenced by the watchdog interface.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	true
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	255
min	0

4.14 Parameter WdgInitialTimeout

Wdg Initial Timeout

The initial timeout (sec) for the trigger condition to be initialized during Init function. It shall be not larger than WdgMaxTimeout.

Property	Value
type	ECUC-FLOAT-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0.0
max	65.535
min	0.0

4.15 Parameter WdgMaxTimeout

Wdg Max Timeout

The maximum timeout (sec) to which the watchdog trigger condition can be initialized.

Property	Value
type	ECUC-FLOAT-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0.0
max	65.535
min	0.0

4.16 Parameter WdgRunArea

Wdg Run Area

Represents the watchdog driver execution area is either from ROM(Flash) or RAM as required with the particular microcontroller.

This should be set to RAM when other Wdg dependecies, such as Dem and Gpt, are also running from RAM. Otherwise, ROM should be selected.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	ROM
literals	['RAM', 'ROM']

4.17 Parameter WdgTriggerLocation

Wdg Trigger Location

Location (memory address) of the watchdog trigger routine.

Note:Not supported by the current hardware.

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	NULL_PTR

4.18 Parameter WdgCallbackNotification

Vendor specific:

WdgCallbackNotification

Callback notification for the ISR Wdg_Ipw_Isr function

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	NULL_PTR

4.19 Parameter WdgVersionInfoApi

Wdg VersionInfo Api

Compile switch to enable or disable the version information API.

True: API enabled

False:API disabled

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.20 Reference WdgEcucPartitionRef

Maps the Wdg driver to zero or one ECUC partition to make the driver API available in this partition.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-LINK-TIME: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.21 Container AutosarExt

Autosar Extension settings.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.22 Parameter WdgDisableDemReportErrorStatus

Enable/Disable Dem error reporting.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.23 Container WdgClockReferencePoint

This container contains a parameter, which represents a reference to a container of the type McuClockReferencePoint (defined in module MCU).

If the cip is trimmed (internal RC oscillator clock FIRC running at 48MHz frequency), then configure in MCU a reference point of FIRC type

with 48MHz frequency. If the cip is not trimmed (FIRC running at frequency different than 48MHz), then configure in MCU a reference point

of CUSTOM type with the real FIRC frequency measured on the cip.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.24 Reference WdgClockReference

Reference to a container of the type McuClockReferencePoint, to select an input clock.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Mcu/McuModuleConfiguration/McuClockSetting↩ Config/McuClockReferencePoint

4.25 Container WdgSettingsConfig

WdgSettingsConfig

Configuration items for the different watchdog settings, including those for external watchdog hardware.

Included subcontainers:

- WdgExternalConfiguration
- WdgSettingsFast
- WdgSettingsOff
- WdgSettingsSlow

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.26 Parameter WdgInstance

Vendor specific:

Wdg Hardware Instance

Select specific hardware instance for watchdog driver initialization.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	WDOG
literals	['WDOG']

4.27 Parameter WdgDefaultMode

Wdg Default Mode

Default mode for watchdog driver initialization.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	WDGIF_SLOW_MODE
literals	['WDGIF_FAST_MODE', 'WDGIF_OFF_MODE', 'WDGIF_SLOW_MODE']

4.28 Parameter WdgInterruptContentEnable

Vendor specific:

Wdg Interrupt Content Enable

This parametter is used to generate interrupt content for each Wdg Instance.

True = Interrupt content is generated.

False = Interrupt content is not generated.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	true

4.29 Reference WdgExternalTriggerCounterRef

Vendor specific:

Wdg External Trigger Counter

Reference to the GptChannel configuration which set for the watchdog servicing routine implementation.

Property	Value
type	ECUC-CHOICE-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destinations	['/AUTOSAR/EcucDefs/Gpt/GptChannelConfigSet/GptChannelConfiguration']

4.30 Container WdgExternalConfiguration

WdgExternalConfiguration

Configuration items for an external watchdog hardware

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-LINK-TIME: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.31 Reference WdgExternalContainerRef

WdgExternalContainerRef

Reference to either

- a DioChannelGroup container in case the hardware watchdog is connected via DIO pins

- a SpiSequenceConfiguration container in case the watchdog hardware is accessed via SPI

Note:This parameter is not used by current implementation

Property	Value
type	ECUC-CHOICE-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destinations	['/AUTOSAR/EcucDefs/Dio/DioConfig/DioPort/DioChannelGroup', '/AUT↩ OSAR/EcucDefs/Spi/SpiDriver/SpiSequence']

4.32 Container WdgSettingsFast

WdgSettingsFast

Hardware dependent settings for the watchdog driver's fast mode.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.33 Parameter WdgClockValue

Vendor specific:

Wdg Clock Value

This is the Implementation Specific parameter.

Indicates Wdg Clock Value in KHz (internal oscilator clock value is by default 128KHz).

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	56000
min	0

4.34 Parameter WdgRunsInStopMode

Vendor specific:

Wdg Runs in Stop Mode

This is the Implementation Specific parameter.

Enabled: Wdg continues to count even while the processor core is in stop mode.

Disabled: Wdg stops counting if the processor core is in stop mode.

Note: The 'WdgRunsInStopMode' parameter specifies if the watchdog timer should run or not while the clock to the core is halted.

This is true only for the STOP0 mode of the controller. It will always run while the controller is in the HALT0 mode.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.35 Parameter WdgRunsInDebugMode

Vendor specific:

Wdg Runs In Debug Mode

This is the Implementation Specific parameter.

Enabled: Wdg continues to count even while the device enters the debug mode.

Disabled: Wdg stops counting if the processor core when the device enters the debug mode

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.36 Parameter WdgRunsInWaitMode

Vendor specific:

Wdg Runs In Wait Mode

This is the Implementation Specific parameter.

Enabled: Watchdog continues to count even while the device enters Wait mode.

Disabled: Watchdog stops counting if the processor core when the device enters Wait mode

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.37 Parameter WdgOperationMode

Vendor specific:

Wdg Operation Mode

This is the Implementation Specific parameter.

ResetOnTimeOut: Generate a reset on a time-out.

Interrupt: Generate an interrupt on an initial time-out.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	ResetOnTimeOut
literals	['ResetOnTimeOut', 'Interrupt']

4.38 Parameter WdgClockSelection

Vendor specific:

Wdg Clock Selection

WDOG clock selection.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	Bus_Clock
literals	['Bus_Clock', 'LPO_Clock', 'SOSC_Clock', 'SIRC_Clock']

4.39 Parameter WdgTimeoutPeriod

Vendor specific:

Wdg Timeout Period

This is the Implementation Specific parameter.Wdg Time-Out Period in seconds. Selects the time-out period for the Wdg.

Property	Value
type	ECUC-FLOAT-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0.5
max	16776.96
min	0.0

4.40 Parameter WdgWindowMode

Vendor specific:

Wdg WindowMode.

Disabled: Regular mode, service sequence can be done at any time.

Enabled : Windowed mode, the service sequence is only valid when performed during the time frame specified by WdgWindowPeriod.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

Property	Value
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.41 Parameter WdgWindowPeriod

Vendor specific:

Wdg Window Period

This is the Implementation Specific parameter.

Wdg Window Value - When WdgWindowMode is enabled, the Wdg counter can be refreshed in the last period of the counter specified by WdgWindowPeriod.

For example, if WdgTimeoutPeriod is 0.4s and WdgWindowPeriod is 0.1s, the Wdg can be refreshed in the last 0.1s before the counter

reaches 0.4s.

Property	Value
type	ECUC-FLOAT-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0.0
max	16776.96
min	0.0

4.42 Parameter WdgPrescalerEnabled

Vendor specific:

Wdg Prescaler Enabled.

Disabled: Prescaler Disabled.

Enabled : Prescaler Enabled.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	true

4.43 Parameter WdgAllowUpdates

Vendor specific:

Wdg Allow Updates.

Disabled: Updates not allowed.

Enabled : Updates allowed.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	true

4.44 Reference WdgClkSrcRef

Reference to the WdgClockReferencePoint from which the clock is derived.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/TS_T40D2M10I1R0/Wdg/WdgClockReferencePoint

4.45 Container WdgSettingsOff

WdgSettingsOff

Hardware dependent settings for the watchdog driver's off mode.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.46 Parameter WdgAllowUpdates

Vendor specific:

Wdg Allow Updates.

Disabled: Updates not allowed.

Enabled : Updates allowed.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	true

4.47 Container WdgSettingsSlow

WdgSettingsSlow

Hardware dependent settings for the watchdog driver's slow mode.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.48 Parameter WdgClockValue

Vendor specific:

Wdg Clock Value

This is the Implementation Specific parameter.

Indicates Wdg Clock Value in KHz (internal oscilator clock value is by default 128KHz).

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	56000
min	0

4.49 Parameter WdgRunsInStopMode

Vendor specific:

Wdg Runs in Stop Mode

This is the Implementation Specific parameter.

Enabled: Wdg continues to count even while the processor core is in stop mode.

Disabled: Wdg stops counting if the processor core is in stop mode.

Note: The 'WdgRunsInStopMode' parameter specifies if the watchdog timer should run or not while the clock to the core is halted.

This is true only for the STOP0 mode of the controller. It will always run while the controller is in the HALT0 mode.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.50 Parameter WdgRunsInDebugMode

Vendor specific:

Wdg Runs In Debug Mode

This is the Implementation Specific parameter.

Enabled: Wdg continues to count even while the device enters the debug mode.

Disabled: Wdg stops counting if the processor core when the device enters the debug mode.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.51 Parameter WdgRunsInWaitMode

Vendor specific:

Wdg Runs In Wait Mode

This is the Implementation Specific parameter.

Enabled: Watchdog continues to count even while the device enters Wait mode.

Disabled: Watchdog stops counting if the processor core when the device enters Wait mode

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.52 Parameter WdgOperationMode

Vendor specific:

Wdg Operation Mode

This is the Implementation Specific parameter.

ResetOnTimeOut: Generate a reset on a time-out.

Interrupt: Generate an interrupt on an initial time-out.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	ResetOnTimeOut
literals	['ResetOnTimeOut', 'Interrupt']

4.53 Parameter WdgClockSelection

Vendor specific:

Wdg Clock Selection

WDOG clock selection.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	Bus_Clock
literals	['Bus_Clock', 'LPO_Clock', 'SOSC_Clock', 'SIRC_Clock']

4.54 Parameter WdgTimeoutPeriod

Vendor specific:

Wdg Timeout Period

This is the Implementation Specific parameter.Wdg Time-Out Period in seconds. Selects the time-out period for the Wdg.

Property	Value
type	ECUC-FLOAT-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0.5
max	16776.96
min	0.0

4.55 Parameter WdgWindowMode

Vendor specific:

Wdg WindowMode.

Disabled: Regular mode, service sequence can be done at any time.

Enabled : Windowed mode, the service sequence is only valid when performed during the time frame specified by WdgWindowPeriod.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.56 Parameter WdgWindowPeriod

Vendor specific:

Wdg Window Period

This is the Implementation Specific parameter.

Wdg Window Value - When WdgWindowMode is enabled, the Wdg counter can be refreshed in the last period of the counter specified by WdgWindowPeriod.

For example, if WdgTimeoutPeriod is 0.4s and WdgWindowPeriod is 0.1s, the Wdg can be refreshed in the last 0.1s before the counter

reaches 0.4s.

Property	Value
type	ECUC-FLOAT-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0.0
max	16776.96
min	0.0

4.57 Parameter WdgPrescalerEnabled

Vendor specific:

Wdg Prescaler Enabled.

Disabled: Prescaler Disabled.

Enabled : Prescaler Enabled.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	true

4.58 Parameter WdgAllowUpdates

Vendor specific:

Wdg Allow Updates.

Disabled: Updates not allowed.

Enabled : Updates allowed.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	true

4.59 Reference WdgClkSrcRef

Reference to the WdgClockReferencePoint from which the clock is derived.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-LINK-TIME: LINK
	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/TS_T40D2M10I1R0/Wdg/WdgClockReferencePoint

4.60 Container WdgPublishedInformation

WdgPublishedInformation

Container holding all Wdg specific published information parameters

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.61 Parameter WdgTriggerMode

Wdg Trigger Mode

Watchdog trigger mode (toggle/window/both).

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	WDG_BOTH
literals	['WDG_BOTH', 'WDG_TOGGLE', 'WDG_WINDOW']

4.62 Container CommonPublishedInformation

Common container, aggregated by all modules. It contains published information about vendor and versions.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.63 Parameter ArReleaseMajorVersion

Vendor specific:

Major version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.64 Parameter ArReleaseMinorVersion

Vendor specific:

Minor version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.65 Parameter ArReleaseRevisionVersion

Vendor specific:

Revision version number of AUTOSAR specification on which the appropriate implementation is based on.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.66 Parameter ModuleId

Vendor specific:

Module ID of this module from Module List.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	102
max	102
min	102

4.67 Parameter SwMajorVersion

Vendor specific:

Major version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	1
max	1
min	1

4.68 Parameter SwMinorVersion

Vendor specific:

Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.69 Parameter SwPatchVersion

Vendor specific:

Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	1
max	1
min	1

4.70 Parameter VendorApiInfix

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name.

This parameter is used to specify the vendor specific name. In total, the implementation specific name is generated as follows:

<ModuleName>_>VendorId>_<VendorApiInfix>

E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a VendorApiInfix of "v11r456" a api name Can_Write defined in the SWS will translate to Can_123_v11r456Write.

This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false

Property	Value
multiplicityConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	Instance0

4.71 Parameter VendorId

Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-LINK-TIME: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	43
max	43
min	43

This chapter describes the Tresos configuration plug-in for the *WDG* Driver. The most of the parameters are described below.

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Ewm_Ip 56
Wdg 60
Wdog_Ip 66

Chapter 6

Module Documentation

6.1 Ewm_Ip

6.1.1 Detailed Description

Data Structures

- struct Ewm_Ip_ConfigType

  EWM configuration structure This structure is used to configure the EWM prescaler, window, interrupt and input pin. More...

Types Reference

- typedef void(* Ewm_Ip_CallbackPtrType) (void)

  EWM callback type Implements : Ewm_Ip_CallbackPtrType_Class.

Enum Reference

- enum Ewm_Ip_StatusType

  Enum defining the possible type values for EWM API @Implements Ewm_Ip_StatusType_enumeration.
- enum Ewm_Ip_Assert_LogicType

  EWM input pin configuration Configures if the input pin is enabled and when is asserted Implements : Ewm_Ip↩ _Assert_LogicType_Class.

Function Reference

- Ewm_Ip_StatusType Ewm_Ip_Init (const uint8 Instance, const Ewm_Ip_ConfigType *const ConfigPtr)

  Init EWM. This method initializes EWM instance to the configuration from the passed structure. The user must make sure that the clock is enabled. This is the only method needed to be called to start the module.
- void Ewm_Ip_Service (const uint8 Instance)

  Refresh EWM. This method needs to be called within the window period specified by the Compare Low and Compare High registers.

6.1.2 Data Structure Documentation

6.1.2.1 struct Ewm_Ip_ConfigType

EWM configuration structure This structure is used to configure the EWM prescaler, window, interrupt and input pin.

Implements : ewm_init_config_t_Class

Definition at line 142 of file Ewm_Ip_Types.h.

Data Fields

Type	Name	Description
Ewm_Ip_Assert_LogicType	AssertLogic	Assert logic for EWM input pin
boolean	InterruptEnable	Enable EWM interrupt
uint8	Prescaler	EWM clock prescaler
uint8	CompareLow	Compare low value
uint8	CompareHigh	Compare high value
Ewm_Ip_CallbackPtrType	pfEwmCallback	Interrupt callback

6.1.3 Types Reference

6.1.3.1 Ewm_Ip_CallbackPtrType

```
typedef void(* Ewm_Ip_CallbackPtrType) (void)
```

EWM callback type Implements : Ewm_Ip_CallbackPtrType_Class.

Definition at line 133 of file Ewm_Ip_Types.h.

6.1.4 Enum Reference

6.1.4.1 Ewm_Ip_StatusType

```
enum Ewm_Ip_StatusType
```

Enum defining the possible type values for EWM API @Implements Ewm_Ip_StatusType_enumeration.

Definition at line 106 of file Ewm_Ip_Types.h.

6.1.4.2 Ewm_Ip_Assert_LogicType

```
enum Ewm_Ip_Assert_LogicType
```

EWM input pin configuration Configures if the input pin is enabled and when is asserted Implements : Ewm_Ip↩ _Assert_LogicType_Class.

Enumerator

EWM_IN_ASSERT_DISABLED	Input pin disabled
EWM_IN_ASSERT_ON_LOGIC_ZERO	Input pin asserts EWM when on logic 0
EWM_IN_ASSERT_ON_LOGIC_ONE	Input pin asserts EWM when on logic 1

Definition at line 117 of file Ewm_Ip_Types.h.

6.1.5 Function Reference

6.1.5.1 Ewm_Ip_Init()

```
Ewm_Ip_StatusType Ewm_Ip_Init (
            const uint8 Instance,
            const Ewm_Ip_ConfigType *const ConfigPtr )
```

Init EWM. This method initializes EWM instance to the configuration from the passed structure. The user must make sure that the clock is enabled. This is the only method needed to be called to start the module.

Example configuration structure:

Ewm_Ip_ConfigType ewmUserCfg = {

.AssertLogic = EWM_IN_ASSERT_ON_LOGIC_ZERO,

.InterruptEnable = true,

.Prescaler = 128,

.CompareLow = 0,

.CompareHigh = 254

};

This configuration will enable the peripheral, with input pin configured to assert on logic low, interrupt enabled, prescaler 128 and maximum refresh window.

The EWM can be initialized only once per CPU reset as the registers are write once.

Parameters

in	*u8Instance*	EWM instance number
in	*config*	Pointer to the module configuration structure.

Returns

Ewm_Ip_StatusType Will return the status of the operation:

- STATUS_SUCCESS if the operation is successful
- STATUS_ERROR if the windows values are not correct or if the instance is already enabled

6.1.5.2 Ewm_Ip_Service()

```
void Ewm_Ip_Service (
             const uint8 Instance )
```

Refresh EWM. This method needs to be called within the window period specified by the Compare Low and Compare High registers.

Parameters

in	*u8Instance*	EWM instance number

Returns

None

6.2 Wdg

6.2.1 Detailed Description

Data Structures

- struct Wdg_ConfigType

  Defines the configuration structure. *More...*

Enum Reference

- enum Wdg_ServiceIdType

  This enumerated type will contain the service ids for the watchodg functions.
- enum Wdg_ErrorIdType

  Indicates the aditional det errors used by the watchdog driver.
- enum Wdg_Ipw_InstanceType

  Wdg_Ipw_InstanceType.
- enum Wdg_Ipw_IpType

  Wdg_Ipw_IpType.

Function Reference

- void Wdg_ChannelInit (const Wdg_Ipw_InstanceType Instance, const Wdg_ConfigType *ConfigPtr)

  This function initializes the WDG module.
- Std_ReturnType Wdg_ChannelSetMode (const Wdg_Ipw_InstanceType Instance, WdgIf_ModeType Mode)

  Switches the watchdog into the mode Mode.
- void Wdg_ChannelSetTriggerCondition (const Wdg_Ipw_InstanceType Instance, uint16 Timeout)

  Reset the watchdog timeout counter according to the timeout value passed.
- void Wdg_ChannelGetVersionInfo (const Wdg_Ipw_InstanceType Instance, Std_VersionInfoType *Version↩ Info)

  Returns the version information of the module.
- void Wdg_ChannelService (const Wdg_Ipw_InstanceType Instance)

  Perform a wdg channel service.

6.2.2 Data Structure Documentation

6.2.2.1 struct Wdg_ConfigType

Defines the configuration structure.

Definition at line 218 of file Wdg_ChannelTypes.h.

Data Fields

- const WdgIf_ModeType Wdg_DefaultMode
  The number of configured channels.
- const Wdg_Ipw_InstanceType Wdg_Instance
  The instance id.
- const Gpt_ChannelType Wdg_TimerChannel
  Gpt Channel configured.
- const uint32 Wdg_u32TriggerSourceClock
  The frequency of the configured timer channel.
- const Wdg_ModeType *const Wdg_ModeSettings [3]
  Pointer to Watchdog Specific implementation details.

6.2.2.1.1 Field Documentation

6.2.2.1.1.1 Wdg_DefaultMode `const WdgIf_ModeType Wdg_DefaultMode`

The number of configured channels.

Definition at line 223 of file Wdg_ChannelTypes.h.

6.2.2.1.1.2 Wdg_Instance `const Wdg_Ipw_InstanceType Wdg_Instance`

The instance id.

Definition at line 227 of file Wdg_ChannelTypes.h.

6.2.2.1.1.3 Wdg_TimerChannel `const Gpt_ChannelType Wdg_TimerChannel`

Gpt Channel configured.

Definition at line 233 of file Wdg_ChannelTypes.h.

6.2.2.1.1.4 Wdg_u32TriggerSourceClock `const uint32 Wdg_u32TriggerSourceClock`

The frequency of the configured timer channel.

Definition at line 238 of file Wdg_ChannelTypes.h.

6.2.2.1.1.5 Wdg_ModeSettings `const Wdg_ModeType* const Wdg_ModeSettings[3]`

Pointer to Watchdog Specific implementation details.

Definition at line 244 of file Wdg_ChannelTypes.h.

6.2.3 Enum Reference

6.2.3.1 Wdg_ServiceIdType

`enum Wdg_ServiceIdType`

This enumerated type will contain the service ids for the watchodg functions.

Precondition

To define WDG_GETVERSION_ID, WDG_VERSION_INFO_API has to be equal to STD_ON

Definition at line 147 of file Wdg_ChannelTypes.h.

6.2.3.2 Wdg_ErrorIdType

`enum Wdg_ErrorIdType`

Indicates the aditional det errors used by the watchdog driver.

Definition at line 183 of file Wdg_ChannelTypes.h.

6.2.3.3 Wdg_Ipw_InstanceType

`enum Wdg_Ipw_InstanceType`

Wdg_Ipw_InstanceType.

Contains the information related to available Wdg Instances.

Definition at line 153 of file Wdg_Ipw_Types.h.

6.2.3.4 Wdg_Ipw_IpType

```
enum Wdg_Ipw_IpType
```

Wdg_Ipw_IpType.

Contains the Ip types available for Wdg.

Definition at line 163 of file Wdg_Ipw_Types.h.

6.2.4 Function Reference

6.2.4.1 Wdg_ChannelInit()

```
void Wdg_ChannelInit (
            const Wdg_Ipw_InstanceType Instance,
            const Wdg_ConfigType * ConfigPtr )
```

This function initializes the WDG module.

The `Wdg_Init` function shall initialize the Wdg module and the watchdog hardware, i.e. it shall set the default watchdog mode and timeout period as provided in the configuration set.

Parameters

in	*ConfigPtr*	Pointer to configuration set.
in	*Instance*	Harwdware instance.

Returns

void

6.2.4.2 Wdg_ChannelSetMode()

```
Std_ReturnType Wdg_ChannelSetMode (
            const Wdg_Ipw_InstanceType Instance,
            WdgIf_ModeType Mode )
```

Switches the watchdog into the mode Mode.

By choosing one of a limited number of statically configured settings (e.g. toggle or window watchdog, different timeout periods) the Wdg module and the watchdog hardware can be switched between the following three different watchdog modes using the `Wdg_SetMode` function:

- WDGIF_OFF_MODE,
- WDGIF_SLOW_MODE,
- WDGIF_FAST_MODE.

Parameters

in	*Mode*	One of the following statically configured modes: 1. WDGIF_OFF_MODE, 2. WDGIF_SLOW_MODE, 3. WDGIF_FAST_MODE.
in	*Instance*	Harwdware instance.

Returns

Std_ReturnType.

Return values

E_OK	Mode switch executed completely and successfully.
E_NOT_OK	The mode switch encountered errors.

6.2.4.3 Wdg_ChannelSetTriggerCondition()

```
void Wdg_ChannelSetTriggerCondition (
            const Wdg_Ipw_InstanceType Instance,
            uint16 Timeout )
```

Reset the watchdog timeout counter according to the timeout value passed.

Parameters

in	*Timeout*	value (milliseconds) for setting the trigger counter.
in	*Instance*	Harwdware instance.

6.2.4.4 Wdg_ChannelGetVersionInfo()

```
void Wdg_ChannelGetVersionInfo (
```

```
            const Wdg_Ipw_InstanceType Instance,
            Std_VersionInfoType * VersionInfo )
```

Returns the version information of the module.

The Wdg_ChannelGetVersionInfo function shall return the version information of this module. The version information includes:

- Module Id,
- Vendor Id,
- Vendor specific version numbers.

Precondition

This function is available if the WDG_VERSION_INFO_API must be equal STD_ON.

Parameters

in,out	*VersionInfo*	Pointer to where to store the version information of this module.
in	*Instance*	Harwdware instance.

6.2.4.5 Wdg_ChannelService()

```
void Wdg_ChannelService (
            const Wdg_Ipw_InstanceType Instance )
```

Perform a wdg channel service.

Precondition

This this function is availble if the WDG_DIRECT_SERVICE must be equal STD_ON.

Parameters

in	*Instance*	Harwdware instance.

Returns

void

6.3 Wdog_Ip

6.3.1 Detailed Description

Data Structures

- struct Wdog_Ip_OpModeType
  WDOG option mode configuration structure Implements : Wdog_Ip_OpModeType_Class. More...
- struct Wdog_Ip_ConfigType
  WDOG user configuration structure Implements : Wdog_Ip_ConfigType_Class. More...

Types Reference

- typedef void(* Wdog_Ip_CallbackPtrType) (void)
  WDOG callback type Implements : Wdog_Ip_CallbackPtrType_Class.

Enum Reference

- enum Wdog_Ip_StatusType
  Enum defining the possible type values for WDOG API @Implements Wdog_Ip_StatusType_enumeration.
- enum Wdog_Ip_ClkSourceType
  Clock sources for the WDOG. Implements : Wdog_Ip_ClkSourceType_Class.
- enum Wdog_Ip_TestModeType
  Test modes for the WDOG. Implements : Wdog_Ip_TestModeType_Class.

Function Reference

- Wdog_Ip_StatusType Wdog_Ip_Init (const uint8 Instance, const Wdog_Ip_ConfigType *const ConfigPtr)
  Initializes the WDOG driver.
- Wdog_Ip_StatusType Wdog_Ip_DeInit (const uint8 Instance)
  De-initializes the WDOG driver.
- void Wdog_Ip_Service (const uint8 Instance)
  Refreshes the WDOG counter.
- Wdog_Ip_StatusType Wdog_Ip_SetTimeout (const uint8 Instance, uint16 Timeout, uint16 WindowValue)
  Sets the value of the WDOG timeout.
- Wdog_Ip_StatusType Wdog_Ip_SetTestMode (const uint8 Instance, Wdog_Ip_TestModeType TestMode)
  Changes the WDOG test mode.
- Wdog_Ip_TestModeType Wdog_Ip_GetTestMode (const uint8 Instance)
  Returns the current WDOG test mode.
- Wdog_Ip_StatusType Wdog_Ip_Config (const uint8 Instance, const Wdog_Ip_ConfigType *const Config↩ Ptr)
  Configures the WDOG driver, but does not enable or disable it.
- Wdog_Ip_StatusType Wdog_Ip_StartTimer (const uint8 Instance)
  Starts the WDOG counter.
- Wdog_Ip_StatusType Wdog_Ip_StopTimer (const uint8 Instance)
  Stops the WDOG counter.

6.3.2 Data Structure Documentation

6.3.2.1 struct Wdog_Ip_OpModeType

WDOG option mode configuration structure Implements : Wdog_Ip_OpModeType_Class.

Definition at line 154 of file Wdog_Ip_Types.h.

Data Fields

Type	Name	Description
boolean	bWait	Wait mode
boolean	bStop	Stop mode
boolean	bDebug	Debug mode

6.3.2.2 struct Wdog_Ip_ConfigType

WDOG user configuration structure Implements : Wdog_Ip_ConfigType_Class.

Definition at line 165 of file Wdog_Ip_Types.h.

Data Fields

Type	Name	Description
Wdog_Ip_ClkSourceType	clkSource	The clock source of the WDOG
Wdog_Ip_OpModeType	opMode	The modes in which the WDOG is functional
boolean	UpdateEnable	If true, further updates of the WDOG are enabled
boolean	IntEnable	If true, an interrupt request is generated before reset
boolean	WinEnable	If true, window mode is enabled
uint16	WindowValue	The window value
uint16	TimeoutValue	The timeout value
boolean	PrescalerEnable	If true, a fixed 256 prescaling of the counter reference clock is enabled
Wdog_Ip_CallbackPtrType	pfWdogCallback	Interrupt callback

6.3.3 Types Reference

6.3.3.1 Wdog_Ip_CallbackPtrType

```
typedef void(* Wdog_Ip_CallbackPtrType) (void)
```

WDOG callback type Implements : Wdog_Ip_CallbackPtrType_Class.

Definition at line 148 of file Wdog_Ip_Types.h.

6.3.4 Enum Reference

6.3.4.1 Wdog_Ip_StatusType

enum Wdog_Ip_StatusType

Enum defining the possible type values for WDOG API @Implements Wdog_Ip_StatusType_enumeration.

Definition at line 106 of file Wdog_Ip_Types.h.

6.3.4.2 Wdog_Ip_ClkSourceType

enum Wdog_Ip_ClkSourceType

Clock sources for the WDOG. Implements : Wdog_Ip_ClkSourceType_Class.

Enumerator

WDOG_IP_BUS_CLOCK	Bus clock
WDOG_IP_LPO_CLOCK	LPO clock
WDOG_IP_SOSC_CLOCK	SOSC clock
WDOG_IP_SIRC_CLOCK	SIRC clock

Definition at line 117 of file Wdog_Ip_Types.h.

6.3.4.3 Wdog_Ip_TestModeType

enum Wdog_Ip_TestModeType

Test modes for the WDOG. Implements : Wdog_Ip_TestModeType_Class.

Enumerator

WDOG_IP_TST_DISABLED	Test mode disabled
WDOG_IP_TST_USER	User mode enabled. (Test mode disabled.)
WDOG_IP_TST_LOW	Test mode enabled, only the low byte is used.
WDOG_IP_TST_HIGH	Test mode enabled, only the high byte is used.

Definition at line 130 of file Wdog_Ip_Types.h.

6.3.5 Function Reference

6.3.5.1 Wdog_Ip_Init()

```
Wdog_Ip_StatusType Wdog_Ip_Init (
            const uint8 Instance,
            const Wdog_Ip_ConfigType *const ConfigPtr )
```

Initializes the WDOG driver.

Parameters

in	*Instance*	WDOG peripheral instance number
in	*ConfigPtr*	pointer to the WDOG user configuration structure

Returns

operation status

- WDOG_IP_STATUS_SUCCESS: WDOG initialization was successful.
- WDOG_IP_STATUS_ERROR: Operation failed. Possible causes: WDOG configuration updates are not allowed; If window mode enabled and window value greater than or equal to the timeout value.
- WDOG_IP_STATUS_TIMEOUT: The unlock operation was unsuccesful and has timed out.

6.3.5.2 Wdog_Ip_DeInit()

```
Wdog_Ip_StatusType Wdog_Ip_DeInit (
            const uint8 Instance )
```

De-initializes the WDOG driver.

Parameters

in	*Instance*	WDOG peripheral instance number

Returns

operation status

- WDOG_IP_STATUS_SUCCESS: If the WDOG module is de-initialized successfulully.
- WDOG_IP_STATUS_ERROR: Operation failed. Possible causes: WDOG updates are not allowed.
- WDOG_IP_STATUS_TIMEOUT: The unlock operation was unsuccesful and has timed out.

6.3.5.3 Wdog_Ip_Service()

```
void Wdog_Ip_Service (
             const uint8 Instance )
```

Refreshes the WDOG counter.

Parameters

in	*Instance*	WDOG peripheral instance number

6.3.5.4 Wdog_Ip_SetTimeout()

```
Wdog_Ip_StatusType Wdog_Ip_SetTimeout (
             const uint8 Instance,
             uint16 Timeout,
             uint16 WindowValue )
```

Sets the value of the WDOG timeout.

This function sets the value of the WDOG timeout and enables the window mode if WindowValue is greater than 0.

Parameters

in	*Instance*	WDOG peripheral instance number.
in	*Timeout*	the value of the WDOG timeout.
in	*WindowValue*	the value of the WDOG window.

Returns

operation status

- WDOG_IP_STATUS_SUCCESS: The WDOG timeout and window are configured succesfully.
- WDOG_IP_STATUS_ERROR: Operation failed. Possible causes: WDOG updates are not allowed.
- WDOG_IP_STATUS_TIMEOUT: The unlock operation was unsuccesful and has timed out.

6.3.5.5 Wdog_Ip_SetTestMode()

```
Wdog_Ip_StatusType Wdog_Ip_SetTestMode (
             const uint8 Instance,
             Wdog_Ip_TestModeType TestMode )
```

Changes the WDOG test mode.

This function changes the test mode of the WDOG. If the WDOG is tested in mode, software should set this field to 0x01U in order to indicate that the WDOG is functioning normally.

Parameters

in	*Instance*	WDOG peripheral instance number
in	*testMode*	Test modes for the WDOG.

Returns

operation status

- WDOG_IP_STATUS_SUCCESS: The WDOG test mode is configured succesfully.
- WDOG_IP_STATUS_ERROR: Operation failed. Possible causes: WDOG updates are not allowed.
- WDOG_IP_STATUS_TIMEOUT: The unlock operation was unsuccesful and has timed out.

6.3.5.6 Wdog_Ip_GetTestMode()

```
Wdog_Ip_TestModeType Wdog_Ip_GetTestMode (
            const uint8 Instance )
```

Returns the current WDOG test mode.

This function returns the test mode of the WDOG.

Parameters

in	*Instance*	WDOG peripheral instance number

Returns

Wdog_Ip_TestModeType the current WDOG test mode.

6.3.5.7 Wdog_Ip_Config()

```
Wdog_Ip_StatusType Wdog_Ip_Config (
            const uint8 Instance,
            const Wdog_Ip_ConfigType *const ConfigPtr )
```

Configures the WDOG driver, but does not enable or disable it.

Parameters

in	*Instance*	WDOG peripheral instance number
in	*ConfigPtr*	pointer to the WDOG user configuration structure

Returns

operation status

- WDOG_IP_STATUS_SUCCESS: WDOG configuration was successful.
- WDOG_IP_STATUS_ERROR: Operation failed. Possible causes: WDOG configuration updates are not allowed; If window mode enabled and window value greater than or equal to the timeout value.
- WDOG_IP_STATUS_TIMEOUT: The unlock operation was unsuccesful and has timed out.

6.3.5.8 Wdog_Ip_StartTimer()

```
Wdog_Ip_StatusType Wdog_Ip_StartTimer (
            const uint8 Instance )
```

Starts the WDOG counter.

Parameters

in	*Instance*	WDOG peripheral instance number

Returns

operation status

- WDOG_IP_STATUS_SUCCESS: WDOG timer was started successfully.
- WDOG_IP_STATUS_ERROR: Operation failed. Possible causes: WDOG updates are not allowed.
- WDOG_IP_STATUS_TIMEOUT: The unlock operation was unsuccesful and has timed out.

6.3.5.9 Wdog_Ip_StopTimer()

```
Wdog_Ip_StatusType Wdog_Ip_StopTimer (
            const uint8 Instance )
```

Stops the WDOG counter.

Parameters

in	*Instance*	WDOG peripheral instance number

Returns

operation status

- WDOG_IP_STATUS_SUCCESS: WDOG timer was stopped successfully.
- WDOG_IP_STATUS_ERROR: Operation failed. Possible causes: WDOG updates are not allowed.
- WDOG_IP_STATUS_TIMEOUT: The unlock operation was unsuccesful and has timed out.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.