User Manual

for MPC574XG WDG Driver

Document Number: UM35WDGASR4.2 Rev0002 RTM 1.0.0
Rev. 2.3.0

Contents

Section number **Title** **Page**

Chapter 1
Revision History

Chapter 2
Introduction

Chapter 3
Driver

Chapter 4
Tresos Configuration Plug-in

Chapter 1
Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0.0	19/08/2014	Livia Firan	Updated Manual for MPC574XG AUTOSAR 4.0 Beta 0.9.0 Release
2.0.0	24/04/2015	Bach Nguyen	Updated Manual for MPC574XG AUTOSAR 4.0 RTM 1.0.0 Release
2.1.0	12/07/2015	Bach Nguyen	Updated Manual for MPC574XG AUTOSAR 4.0 RTM 1.0.1 Release
2.2.0	12/08/2016	Tuyen Nguyen Duy	Updated Manual for MPC574XG AUTOSAR 4.0 RTM 1.0.2 Release
2.3.0	17/02/2017	Tuyen Nguyen Duy	Updated Manual for MPC574XG AUTOSAR 4.2 RTM 1.0.0 Release

Chapter 2
Introduction

This User Manual describes NXP Semiconductor AUTOSAR Watchdog (Wdg) for MPC574XG .

AUTOSAR Wdg driver configuration parameters and deviations from the specification are described in Wdg 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 intented to be used with the following microcontroller devices of NXP Semiconductor .

Table 2-1. MPC574XG Derivatives

NXP Semiconductor	MPC5748G_LQFP176, MPC5748G_MAPBGA256, MPC5748G_MAPBGA324, MPC5747G_LQFP176, MPC5747G_MAPBGA256, MPC5747G_MAPBGA324, MPC5746G_LQFP176, MPC5746G_MAPBGA256, MPC5746G_MAPBGA324, MPC5748C_LQFP176, MPC5748C_MAPBGA256, MPC5748C_MAPBGA324, MPC5747C_LQFP176, MPC5747C_MAPBGA256, MPC5747C_MAPBGA324, MPC5746C_LQFP176, MPC5746C_MAPBGA256, MPC5746C_MAPBGA324, MPC5746C_MAPBGA100, MPC5745C_LQFP176, MPC5745C_MAPBGA256, MPC5745C_MAPBGA100, MPC5744C_LQFP176, MPC5744C_MAPBGA256,

Table 2-1. MPC574XG Derivatives

	MPC5744C_MAPBGA100, MPC5746B_LQFP176, MPC5746B_MAPBGA256, MPC5746B_MAPBGA100, MPC5744B_LQFP176, MPC5744B_MAPBGA256, MPC5744B_MAPBGA100, MPC5745B_LQFP176, MPC5745B_MAPBGA256, MPC5745B_MAPBGA100

All of the above microcontroller devices are collectively named as MPC574XG .

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 type: Bold is used for important terms, notes and warnings.

Italic font: Italic typeface is 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.

2.4 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

Abbreviation and Definitions	Description
BSW	Basic Software
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECU	Electronic Control Unit
WDG	Watchdog
MCU	MicroController Unit
MCL	MicroController Library
GPT	General Purpose Timers
ISR	interrupt Service Routine
OS	Operating System
RAM	Random Access Memory
ROM	Read-only Memory
GUI	Graphical User Interface
EcuM	ECU state Manager
API	Application Programming Interface
PB Variant	Post Build Variant
PC Variant	Pre Compile Variant

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 4.2 Rev0002Wdg Driver Software Specification Document.	R4.2 Rev 2
2	MPC5748G Reference Manual	Rev. 5, 12/2016
3	MPC5746C Reference Manual	Rev. 4, 12/2016
4	MPC5748G_1N81M_Rev.2 (official document) (1N81M)	Jun-16
5	MPC5748G_1N81M_0N78S_Comparison_Summary_v2_0 (internal document) (1N81M, 0N78S)	31.10.2016

Table continues on the next page...

Table 2-3. Reference List (continued)

#	Title	Version
6	MPC5746C_1N06M_Rev.4 (official document) (1N06M)	Jul-16
7	MPC5746C_cut1.1_cut2.0_cut2.1_comparison_v0 (internal document) (1N06M, 0N84S, 1N84S)	14-Sep-16
8	C3M_cut2.1_new_errata_20170113 (internal document) (1N84S)	13-Jan-17

Chapter 3
Driver

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 4.2 Rev0002Wdg Driver Software Specification document (See Table Reference List).

3.2 Driver Design Summary

Three Software Watchdog Timers (SWT) with programmable interrupt response are available in MPC5748G . The Software Watchdog Timer (SWT) 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 SWT 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 SWT generates an interrupt or hardware reset. The SWT can be configured to generate a reset or interrupt on an initial time-out, a reset is always generated on a second consecutive time-out.

In addition to these modes of operation, the watchdog timer also supports a windowed mode of operation. In this mode, the servicing of the watchdog timer must be performed in the last part of the timeout period defined by the window register. The window is open when the down counter is less than the value in the SWT_WN register. Outside of this window, service sequence writes are invalid accesses and generate a bus error or reset depending on the value of the SWT_CR.RIA. These timeout responses are configurable using the configuration parameter "WdgOperationMode"

The SWT has the following features:

- 32-bit time-out register to set the time-out period
- Selection of oscillator clock for timer operation

- Programmable selection of window mode or regular servicing
- Programmable selection of reset or interrupt on an initial time-out
- Programmable selection of fixed or keyed servicing
- Master access protection

The SWT is started on exit of power-on phase (RGM phase 2) to monitor flash boot sequence phase. It is then reset during (RGM phase 3) and optionally enabled when platform reset is released depending on value of flash user option bit 31 (WATCHDOG_EN)

3.3 Driver Limitations

None.

3.4 Driver Usage and Configuration Tips

1.Configure the WDG reference clock from MCU (see parameter WdgClkSrcRef) according to reference point used by the WDG hardware on the platform. Example: If WDG is clocked by SIRC, then the MCU reference clock must be SIRC.

2.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.If there are multiple WDG hardware instances on the platform, the API names will expand according to AUTOSAR requirement BSW00347. For example, if there are instances 0,1 and 2 available on the hardware,then the name of the init functions will be Wdg_43_Instance0_Init, Wdg_43_Instance1_Init and Wdg_43_Instance2_Init instead of Wdg_Init().

3.5 Requirements

The WDG driver uses the SWT hardware IP.

3.6 Deviation from Requirements

The driver deviates from the AUTOSAR Wdg Driver software specification in some places.

There are also some additional requirements (on top of requirements detailed in AUTOSAR Wdg Driver software specification) which need to be satisfied for correct operation.

Table 3-1. Deviations Status Column Description

Term	Definition
N/A	Not available
N/T	Not testable
N/S	Out of scope
N/I	Not implemented
N/F	Not fully implemented
N/R	Unclear Requirement
N/V	Not Verifiable

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

Table 3-2. Driver Deviations Table

Requirement	Status	Description	Notes
SWS_Wdg_00034	N/A	General design rules: 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 that Configured 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. Rationale: This allows the watchdog device to identify the correct trigger input if supported by the hardware.	Not supported by hardware
SWS_Wdg_00040	N/A	If interrupts have to be disabled in order to ensure data consistency or correct functionality of this module (e.g. while switching the watchdog mode or during the watchdog trigger routine), this shall be done by using the corresponding BSW Scheduler functionality if possible (this means definition of an exclusive area). Theinternal watchdog driver (because it belongs to MCAL) may also directly disable interrupts – see SRS_BSW_00429	
SWS_Wdg_00076	N/S	External watchdog driver To access the external watchdog hardware, the Wdg module	External module is customer dependant and is not yet developed.

Table continues on the next page...

Table 3-2. Driver Deviations Table (continued)

Requirement	Status	Description	Notes
		shall use the functionality and API of the corresponding handler or driver, e.g. the SPI handler or DIO driver.	
SWS_Wdg_00077	N/S	External watchdog driver - 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.	External module is customer dependant and is not yet developed.
SWS_Wdg_00078	N/S	External watchdog driver - 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.	External module is customer dependant and is not yet developed.
ECUC_Wdg_00113	N/S	WdgExternalContainerRef container definition	External module is customer dependant and is not yet developed.
ECUC_Wdg_00118	N/A	Wdg_Configuration - Location (memory address) of the watchdog trigger routine. Dependancy: Only relevant if provided by hardware and needed by the system.	Not supported by hardware.
ECUC_Wdg_00148	N/A	Each variable that shall be accessible by AUTOSAR Debugging shall be defined as global variable.	Avoid global variables or else justify their usage.
ECUC_Wdg_00149	N/A	All type definitions of variables which shall be debugged shall be accessible by the header file Wdg.h.	Avoid global variables or else justify their usage.
ECUC_Wdg_00150	N/A	The declaration of variables in the header file shall be such, that it is possible to calculate the size of the variables by C-"sizeof".	Debugging concept not supported.
SWS_Wdg_00152	N/I	The internal state of the module (which indicates whether it is not initialized, idle or busy) shall be available for debugging.	The internal variables should not be accesible outside the driver - independent of the purpose.
SWS_Wdg_00153	N/I	The internal variable for the watchdog timeout counter shall be available for debugging.	The internal variables should not be accesible outside the driver - independent of the purpose.
SWS_Wdg_00154	N/I	The internal variable for the watchdog mode shall be available for debugging.	Avoid global variables or else justify their usage.
SWS_Wdg_00169	N/F	If more than one watchdog driver instance exists on an ECU (namely an external and an internal one) the implementer shall provide unique code file names by expanding the names according to BSW00347.	A driver for an external watchdog is not provided but for multiple instances file names (only header files) and API names are following the specification. All watchdog instances have same published parameters and Memory allocation keyword.
SWS_Wdg_00061	N/F	The Wdg module shall adhere to the following file structure	Wdg_PBcfg_(variant).c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB). Mdl_Cfg.c file will contain the definition for all parameters that are not variant aware

3.7 Runtime Errors

The driver generates the following DEM errors at runtime.

Table 3-3. Runtime Errors

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
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 Software specification

The following sections contains driver software specifications.

3.8.1 Define Reference

Constants supported by the driver are as per AUTOSAR Wdg Driver software specification Version 4.2 Rev0002 .

3.8.1.1 Define WDG_AR_RELEASE_MAJOR_VERSION

Table 3-4. Define WDG_AR_RELEASE_MAJOR_VERSION Description

Name	WDG_AR_RELEASE_MAJOR_VERSION
Initializer	4

3.8.1.2 Define WDG_AR_RELEASE_MINOR_VERSION

Table 3-5. Define WDG_AR_RELEASE_MINOR_VERSION Description

Name	WDG_AR_RELEASE_MINOR_VERSION
Initializer	2

3.8.1.3 Define WDG_AR_RELEASE_REVISION_VERSION

Violates: MISRA 2004 Rule 1.4, The compiler/linker shall be checked to ensure that 31 character significance and case sensitivity are supported for external identifiers.

Table 3-6. Define WDG_AR_RELEASE_REVISION_VERSION Description

Name	WDG_AR_RELEASE_REVISION_VERSION
Initializer	2

3.8.1.4 Define WDG_MODULE_ID

Table 3-7. Define WDG_MODULE_ID Description

Name	WDG_MODULE_ID
Initializer	102

3.8.1.5 Define WDG_SW_MAJOR_VERSION

Table 3-8. Define WDG_SW_MAJOR_VERSION Description

Name	WDG_SW_MAJOR_VERSION
Initializer	1

3.8.1.6 Define WDG_SW_MINOR_VERSION

Table 3-9. Define WDG_SW_MINOR_VERSION Description

Name	WDG_SW_MINOR_VERSION
Initializer	0

3.8.1.7 Define WDG_SW_PATCH_VERSION

Table 3-10. Define WDG_SW_PATCH_VERSION Description

Name	WDG_SW_PATCH_VERSION
Initializer	0

3.8.1.8 Define WDG_VENDOR_ID

Table 3-11. Define WDG_VENDOR_ID Description

Name	WDG_VENDOR_ID
Initializer	43

3.8.1.9 Define WDG_DEV_ERROR_DETECT

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

Table 3-12. Define WDG_DEV_ERROR_DETECT Description

Name	WDG_DEV_ERROR_DETECT
Initializer	STD_ON

3.8.1.10 Define WDG_DISABLE_ALLOWED

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

Table 3-13. Define WDG_DISABLE_ALLOWED Description

Name	WDG_DISABLE_ALLOWED

Table continues on the next page...

Table 3-13. Define WDG_DISABLE_ALLOWED Description (continued)

Initializer	STD_OFF

3.8.1.11 Define WDG_PRECOMPILE_SUPPORT

Table 3-14. Define WDG_PRECOMPILE_SUPPORT Description

Name	WDG_PRECOMPILE_SUPPORT
Initializer	STD_OFF

3.8.1.12 Define WDG_ROM

This variable will indicate RAM/ROM execution.

Table 3-15. Define WDG_ROM Description

Name	WDG_ROM
Initializer	

3.8.1.13 Define WDG_VERSION_INFO_API

Compile switch to enable/disable the version information.

Table 3-16. Define WDG_VERSION_INFO_API Description

Name	WDG_VERSION_INFO_API
Initializer	STD_ON

3.8.1.14 Define WDG_DISABLE_DEM_REPORT_ERROR_STATUS

Compile switch enable / disable Diagnostic Event Manager for this module.

Table 3-17. Define WDG_DISABLE_DEM_REPORT_ERROR_STATUS Description

Name	WDG_DISABLE_DEM_REPORT_ERROR_STATUS
Initializer	STD_OFF

3.8.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR Wdg Driver software specification Version 4.2 Rev0002 .

3.8.2.1 Enumeration Wdg_ErrorIdType

Indicates the aditional det errors used by the watchdog driver.

Table 3-18. Enumeration Wdg_ErrorIdType Values

Name	Initializer	Description
WDG_E_DRIVER_STATE	0x10	Type of error: API service used in wrong context (e.g. module not initialized).
WDG_E_PARAM_MODE	0x11	Type of error:API service called with wrong / inconsistent parameter(s).
WDG_E_PARAM_CONFIG	0x12	Type of error:API service called with wrong / inconsistent parameter(s).
WDG_E_PARAM_TIMEOUT	0x13	Type of error: The passed timeout value is higher than the maximum timeout value.
WDG_E_PARAM_POINTER	0x14	Type of error: API is called with wrong pointer value (e.g. NULL pointer).
WDG_E_INIT_FAILED	0x15	Type of error: Invalid configuration set selection.

3.8.2.2 Enumeration Wdg_ServiceIdType

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

Pre: To define WDG_GETVERSION_ID, WDG_VERSION_INFO_API has to be equal to STD_ON.

Table 3-19. Enumeration Wdg_ServiceIdType Values

Name	Initializer	Description
WDG_GETVERSION_ID	0x04	The service id for the Wdg_GetVersion function.
WDG_INIT_ID	0x00	The service id for the Wdg_Init function.
WDG_SETMODE_ID	0x01	The service id for the Wdg_SetMode function.
WDG_SETTRIGGERCONDITION_ID	0x02	The service id for the Wdg_SetTriggerCondition function.
WDG_TRIGGER_ID	0x03	The service id for the Wdg_Trigger function.

3.8.3 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR Wdg Driver software specification Version 4.2 Rev0002 .

3.8.3.1 Function Wdg_43_Instance0_GetVersionInfo

Returns the version information of the module.

Details:

The Wdg_43_Instance0_ChannelGetVersionInfo function shall return the version information of this module. The version information includes:

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

Pre: This function is only required if the WDG_VERSION_INFO_API has to be equal STD_ON.

Implements: Wdg_43_Instance0_ChannelGetVersionInfo_Activity

Prototype: `void Wdg_43_Instance0_GetVersionInfo(pVersioninfo);`

Table 3-20. Wdg_43_Instance0_GetVersionInfo Arguments

Type	Name	Direction	Description
	pVersioninfo	**input, output**	Pointer to where to store the version information of this module.

3.8.3.2 Function Wdg_43_Instance0_Init

Include Memory mapping specification.

Details:

TheWdg_43_Instance0_Initfunction shall initialize the Wdg module and the watchdog hardware the pointer to configuration set must be valid pointer, i.e. it shall set the default watchdog mode and timeout period as provided in the configuration set.

The initialization function of this module shall always have a pointer as a parameter.In the supported ConfigVariants VariantPreCompile and VariantLinkTime if only one configuration variant set is used the parameter shall have a NULL_PTR value.

Violates: MISRA 2004 Required Rule 19.15 precautions to prevent the contents of a header file being included twice This function initializes the WDG module.

Implements: Wdg_43_Instance0_Init_Activity

Prototype: `void Wdg_43_Instance0_Init(const Wdg_ConfigType *pConfigPtr);`

Table 3-21. Wdg_43_Instance0_Init Arguments

Type	Name	Direction	Description
	ConfigPtr	**input**	Pointer to configuration set.

3.8.3.3 Function Wdg_43_Instance0_SetMode

Switches the watchdog into the mode Mode.

Details:

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 theWdg_43_Instance0_SetModefunction:

- WDGIF_OFF_MODE,
- WDGIF_SLOW_MODE,
- WDGIF_FAST_MODE.

Return: Std_ReturnType.

Implements: Wdg_43_Instance0_SetMode_Activity

Prototype: `void Wdg_43_Instance0_SetMode(const WdgIf_ModeType Mode);`

Table 3-22. Wdg_43_Instance0_SetMode Arguments

Type	Name	Direction	Description
const WdgIf_ModeType	Mode	**input**	One of the following statically configured modes: WDGIF_OFF_MODE, WDGIF_SLOW_MODE, WDGIF_FAST_MODE.

Table 3-23. Wdg_43_Instance0_SetMode Return Values

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

3.8.3.4 Function Wdg_43_Instance0_SetTriggerCondition

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

Implements: Wdg_43_Instance0_SetTriggerCondition_Activity

Prototype: `void Wdg_43_Instance0_SetTriggerCondition(const uint16 u16Timeout);`

Table 3-24. Wdg_43_Instance0_SetTriggerCondition Arguments

Type	Name	Direction	Description
	Timeout	**input**	Value (milliseconds) for setting the trigger counter.

3.8.3.5 Function Wdg_43_Instance1_GetVersionInfo

Returns the version information of the module.

Details:

The Wdg_43_Instance1_ChannelGetVersionInfo function shall return the version information of this module. The version information includes:

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

Pre: This function is only required if the WDG_VERSION_INFO_API has to be equal STD_ON.

Implements: Wdg_43_Instance1_ChannelGetVersionInfo_Activity

Prototype: `void Wdg_43_Instance1_GetVersionInfo(pVersioninfo);`

Table 3-25. Wdg_43_Instance1_GetVersionInfo Arguments

Type	Name	Direction	Description
	pVersioninfo	**input, output**	Pointer to where to store the version information of this module.

3.8.3.6 Function Wdg_43_Instance1_Init

Include Memory mapping specification.

Details:

TheWdg_43_Instance1_Initfunction shall initialize the Wdg module and the watchdog hardware and the pointer to configuration set must be valid pointer, i.e. it shall set the default watchdog mode and timeout period as provided in the configuration set.

The initialization function of this module shall always have a pointer as a parameter.In the supported ConfigVariants VariantPreCompile and VariantLinkTime if only one configuration variant set is used the parameter shall have a NULL_PTR value.

Violates: MISRA 2004 Required Rule 19.15 precautions to prevent the contents of a header file being included twice This function initializes the WDG module.

Implements: Wdg_43_Instance1_Init_Activity

Prototype: `void Wdg_43_Instance1_Init(const Wdg_ConfigType *pConfigPtr);`

Table 3-26. Wdg_43_Instance1_Init Arguments

Type	Name	Direction	Description
	ConfigPtr	**input**	Pointer to configuration set.

3.8.3.7 Function Wdg_43_Instance1_SetMode

Switches the watchdog into the mode Mode.

Details:

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 theWdg_43_Instance1_SetModefunction:

- WDGIF_OFF_MODE,
- WDGIF_SLOW_MODE,
- WDGIF_FAST_MODE.

Return: Std_ReturnType.

Implements: Wdg_43_Instance1_SetMode_Activity

Prototype: `void Wdg_43_Instance1_SetMode(const WdgIf_ModeType Mode);`

Table 3-27. Wdg_43_Instance1_SetMode Arguments

Type	Name	Direction	Description
const WdgIf_ModeType	Mode	**input**	One of the following statically configured modes: WDGIF_OFF_MODE, WDGIF_SLOW_MODE, WDGIF_FAST_MODE.

Table 3-28. Wdg_43_Instance1_SetMode Return Values

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

3.8.3.8 Function Wdg_43_Instance1_SetTriggerCondition

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

Implements: Wdg_43_Instance1_SetTriggerCondition_Activity

Prototype: `void Wdg_43_Instance1_SetTriggerCondition(const uint16 u16Timeout);`

Table 3-29. Wdg_43_Instance1_SetTriggerCondition Arguments

Type	Name	Direction	Description
	Timeout	**input**	Value (milliseconds) for setting the trigger counter.

3.8.3.9 Function Wdg_43_Instance2_GetVersionInfo

Returns the version information of the module.

Details:

The Wdg_43_Instance2_ChannelGetVersionInfo function shall return the version information of this module. The version information includes:

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

Pre: This function is only required if the WDG_VERSION_INFO_API has to be equal STD_ON.

Implements: Wdg_43_Instance2_ChannelGetVersionInfo_Activity

Prototype: `void Wdg_43_Instance2_GetVersionInfo(pVersioninfo);`

Table 3-30. Wdg_43_Instance2_GetVersionInfo Arguments

Type	Name	Direction	Description
	pVersioninfo	**input, output**	Pointer to where to store the version information of this module.

3.8.3.10 Function Wdg_43_Instance2_Init

Include Memory mapping specification.

Details:

TheWdg_43_Instance2_Initfunction shall initialize the Wdg module and the watchdog hardware the pointer to configuration set must be valid pointer, i.e. it shall set the default watchdog mode and timeout period as provided in the configuration set.

The initialization function of this module shall always have a pointer as a parameter.In the supported ConfigVariants VariantPreCompile and VariantLinkTime if only one configuration variant set is used the parameter shall have a NULL_PTR value.

Violates: MISRA 2004 Required Rule 19.15 precautions to prevent the contents of a header file being included twice This function initializes the WDG module.

Implements: Wdg_43_Instance2_Init_Activity

Prototype: `void Wdg_43_Instance2_Init(const Wdg_ConfigType *pConfigPtr);`

Table 3-31. Wdg_43_Instance2_Init Arguments

Type	Name	Direction	Description
	ConfigPtr	**input**	Pointer to configuration set.

3.8.3.11 Function Wdg_43_Instance2_SetMode

Switches the watchdog into the mode Mode.

Details:

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 theWdg_43_Instance2_SetModefunction:

- WDGIF_OFF_MODE,
- WDGIF_SLOW_MODE,
- WDGIF_FAST_MODE.

Return: Std_ReturnType.

Implements: Wdg_43_Instance2_SetMode_Activity

Prototype: `void Wdg_43_Instance2_SetMode(const WdgIf_ModeType Mode);`

Table 3-32. Wdg_43_Instance2_SetMode Arguments

Type	Name	Direction	Description
const WdgIf_ModeType	Mode	**input**	One of the following statically configured modes: WDGIF_OFF_MODE, WDGIF_SLOW_MODE, WDGIF_FAST_MODE.

Table 3-33. Wdg_43_Instance2_SetMode Return Values

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

3.8.3.12 Function Wdg_43_Instance2_SetTriggerCondition

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

Implements: Wdg_43_Instance2_SetTriggerCondition_Activity

Prototype: `void Wdg_43_Instance2_SetTriggerCondition(const uint16 u16Timeout);`

Table 3-34. Wdg_43_Instance2_SetTriggerCondition Arguments

Type	Name	Direction	Description
	Timeout	**input**	Value (milliseconds) for setting the trigger counter.

3.8.3.13 Function Wdg_Cbk_GptNotification0

This function performs the triggering of the watchdog.In order to have a functional WDG module it is mandatory to configure this API as a GPT callback notification.

There are two configurations needed:

1. The definition of a GPT channel(please see configuration parameter WdgExternalTriggerCounterRef)

2. In GPT configuration set Wdg_Cbk_GptNotification0 as the GptNotification for the respective channel

Violates: MISRA 2004 Required Rule 19.1 , only preprocessor statements and comments before 'include'

Violates: MISRA 2004 Required Rule 19.15 precautions to prevent the contents of a header file being included twice

Violates: MISRA 2004 Required Rule 8.10, All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required.

Prototype: `void Wdg_Cbk_GptNotification0(void);`

3.8.3.14 Function Wdg_Cbk_GptNotification1

This function performs the triggering of the watchdog.In order to have a functional WDG module it is mandatory to configure this API as a GPT callback notification.

There are two configurations needed:

1. The definition of a GPT channel(please see configuration parameter WdgExternalTriggerCounterRef)

2. In GPT configuration set Wdg_Cbk_GptNotification1 as the GptNotification for the respective channel

Violates: MISRA 2004 Required Rule 19.1 , only preprocessor statements and comments before 'include'

Violates: MISRA 2004 Required Rule 19.15 precautions to prevent the contents of a header file being included twice

Violates: MISRA 2004 Required Rule 8.10, All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required.

Prototype: `void Wdg_Cbk_GptNotification1(void);`

3.8.3.15 Function Wdg_Cbk_GptNotification2

This function performs the triggering of the watchdog.In order to have a functional WDG module it is mandatory to configure this API as a GPT callback notification.

There are two configurations needed:

1. The definition of a GPT channel(please see configuration parameter WdgExternalTriggerCounterRef)

2. In GPT configuration set Wdg_Cbk_GptNotification2 as the GptNotification for the respective channel

Violates: MISRA 2004 Required Rule 19.1 , only preprocessor statements and comments before 'include'

Violates: MISRA 2004 Required Rule 19.15 precautions to prevent the contents of a header file being included twice

Violates: MISRA 2004 Required Rule 8.10, All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required.

Prototype: `void Wdg_Cbk_GptNotification2(void);`

3.8.3.16 Function Wdg_Swt0_Isr

This function process the interrupt SWT0.

Details:

This function process the SWT0 interrupt

Prototype: `void Wdg_Swt0_Isr(void);`

3.8.3.17 Function Wdg_Swt1_Isr

This function process the interrupt SWT1.

Details:

This function process the SWT1 interrupt

Prototype: `void Wdg_Swt1_Isr(void);`

3.8.3.18 Function Wdg_Swt2_Isr

This function process the interrupt SWT2.

Details:

This function process the SWT2 interrupt

Prototype: `void Wdg_Swt2_Isr(void);`

3.8.4 Structs Reference

Data structures supported by the driver are as per AUTOSAR Wdg Driver software specification Version 4.2 Rev0002 .

3.8.4.1 Structure Wdg_ConfigType

Defines the configuration structure.

Declaration:

```
typedef struct
                            {
                              const Gpt_ChannelType Wdg_TimerChannel,
                                      const uint32 Wdg_u32TriggerSourceClock,
                                      Wdg_CallbackPtrType Wdg_CallbackPtr,
                                      const WdgIf_ModeType Wdg_DefaultMode,
                                      const Wdg_IPW_InstanceType Wdg_Instance,
                                      constWdg_Swt_ConfigType* Wdg_ModeSettings[3]
                            } Wdg_ConfigType;
```

Table 3-35. Structure Wdg_ConfigType member description

Member	Description
Wdg_TimerChannel	Gpt Channel configured.
Wdg_u32TriggerSourceClock	The frequency of the configured timer channel.
Wdg_CallbackPtr	Pointer to callback notification.
Wdg_DefaultMode	The number of configured channels.
Wdg_Instance	The instance id.
Wdg_ModeSettings	Pointer to Watchdog Specific implementation details.

3.8.5 Types Reference

Types supported by the driver are as per AUTOSAR Wdg Driver software specification Version 4.2 Rev0002 .

3.8.5.1 Typedef Wdg_CallbackPtrType

Type: void(*

3.9 Symbolic Names Disclaimer

All containers having the symbolic name tag set as true in the Autosar schema will generate defines like:

#define <Container_Short_Name> <Container_ID>

For this reason it is forbidden to duplicate the name of such containers across the MCAL configuration, 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 Wdg Driver. The most of the parameters are described below.

4.1 Configuration elements of Wdg

Included forms :

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

Table 4-1. Revision table

Revision	Date
4.1.0	2010-12-03

4.2 Form IMPLEMENTATION_CONFIG_VARIANT

Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION_CONFIG_VARIANT form.

Table 4-2. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description

Property	Value
Label	Config Variant
Type	ENUMERATION
Default	VariantLinkTime
Range	VariantLinkTime VariantPostBuild VariantPreCompile

4.3 Form WdgDemEventParameterRefs

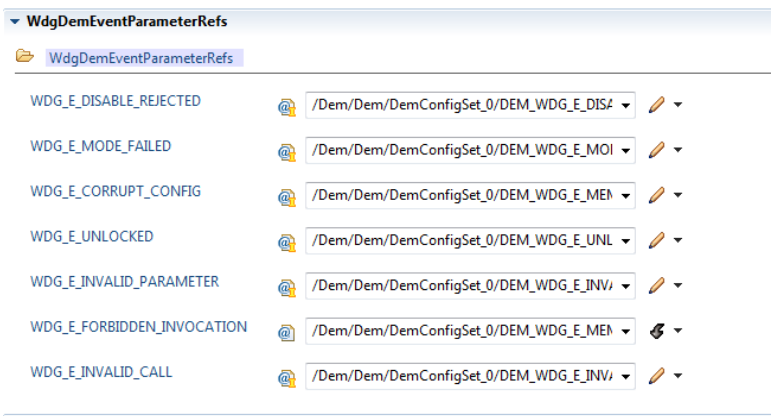

Figure 4-2. Tresos Plugin snapshot for WdgDemEventParameterRefs form.

4.3.1 WDG_E_DISABLE_REJECTED (WdgDemEventParameterRefs)

Table 4-3. Attribute WDG_E_DISABLE_REJECTED (WdgDemEventParameterRefs) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
Origin	AUTOSAR_ECUC
Enable	true

4.3.2 WDG_E_MODE_FAILED (WdgDemEventParameterRefs)

Table 4-4. Attribute WDG_E_MODE_FAILED (WdgDemEventParameterRefs) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
Origin	AUTOSAR_ECUC
Enable	true

4.3.3 WDG_E_CORRUPT_CONFIG (WdgDemEventParameterRefs)

Table 4-5. Attribute WDG_E_CORRUPT_CONFIG (WdgDemEventParameterRefs) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
Origin	Custom
Enable	true

4.3.4 WDG_E_UNLOCKED (WdgDemEventParameterRefs)

Table 4-6. Attribute WDG_E_UNLOCKED (WdgDemEventParameterRefs) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
Origin	Custom
Enable	true

4.3.5 WDG_E_INVALID_PARAMETER (WdgDemEventParameterRefs)

Table 4-7. Attribute WDG_E_INVALID_PARAMETER (WdgDemEventParameterRefs) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
Origin	Custom
Enable	true

4.3.6 WDG_E_FORBIDDEN_INVOCATION (WdgDemEventParameterRefs)

Table 4-8. Attribute WDG_E_FORBIDDEN_INVOCATION (WdgDemEventParameterRefs) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
Origin	Custom
Enable	true

4.3.7 WDG_E_INVALID_CALL (WdgDemEventParameterRefs)

Table 4-9. Attribute WDG_E_INVALID_CALL (WdgDemEventParameterRefs) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
Origin	Custom
Enable	true

4.4 Form WdgGeneral

WdgGeneral

All general parameters of the watchdog driver are collected here.

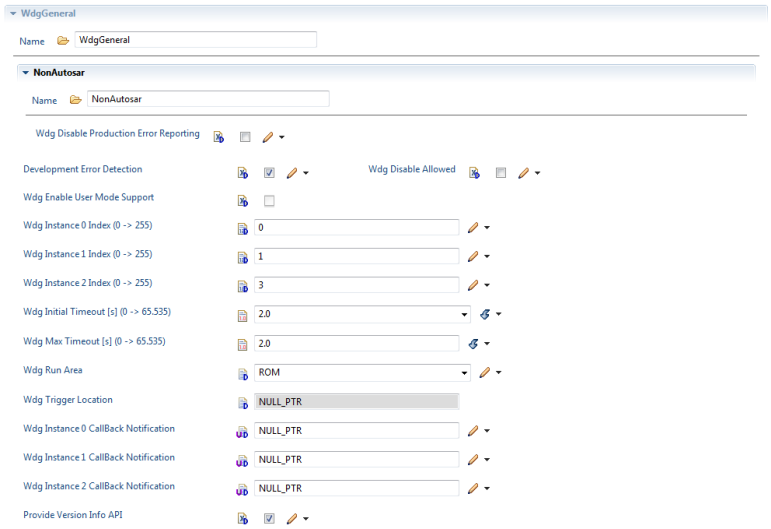

Figure 4-3. Tresos Plugin snapshot for WdgGeneral form.

4.4.1 WdgDisableDemReportErrorStatus (WdgGeneral/ NonAutosar)

Wdg Disable Production Error Reporting

Enable/Disable Dem error reporting. **True**: Dem error reporting enabled **False**: Dem error reporting disabled

Table 4-10. Attribute Wdg Disable Production Error Reporting (WdgGeneral) detailed description

Property	Value
Label	Wdg Disable Production Error Reporting
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false

4.4.2 WdgDevErrorDetect (WdgGeneral)

Wdg Development Error Detect

Compile switch to enable / disable development error detection for this module. **True**: Development error detection enabled **False**: Development error detection disabled

Table 4-11. Attribute WdgDevErrorDetect (WdgGeneral) detailed description

Property	Value
Label	Development Error Detection
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.4.3 WdgDisableAllowed (WdgGeneral)

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

Table 4-12. Attribute WdgDisableAllowed (WdgGeneral) detailed description

Property	Value
Label	Wdg Disable Allowed
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false

4.4.4 WdgEnableUserModeSupport (WdgGeneral)

Wdg Enable User Mode Support

Vendor specific: Wdg can run in user mode without any specific measures. The parameter is not used in the Wdg implementation.

Table 4-13. Attribute WdgEnableUserModeSupport (WdgGeneral) detailed description

Property	Value
Label	Wdg Enable User Mode Support
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false

4.4.5 WdgIndex (WdgGeneral)

Wdg Instance 0 Index

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

Table 4-14. Attribute WdgIndex (WdgGeneral) detailed description

Property	Value
Label	Wdg Instance 0 Index
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	true
Default	0
Invalid	Range <=255 >=0

4.4.6 WdgIndex1 (WdgGeneral)

Wdg Instance 1 Index

Represents the watchdog driver's ID for Instance 1 so that it can be referenced by the watchdog interface.

Table 4-15. Attribute WdgIndex1 (WdgGeneral) detailed description

Property	Value
Label	Wdg Instance 1 Index
Type	INTEGER
Origin	Custom
Symbolic Name	true
Default	0
Invalid	Range <=255 >=0

4.4.7 WdgIndex2 (WdgGeneral)

Wdg Instance 2 Index

Represents the watchdog driver's ID for Instance 2 so that it can be referenced by the watchdog interface.

Table 4-16. Attribute WdgIndex2 (WdgGeneral) detailed description

Property	Value
Label	Wdg Instance 2 Index
Type	INTEGER
Origin	Custom
Symbolic Name	true
Default	0
Invalid	Range <=255 >=0

4.4.8 WdgInitialTimeout (WdgGeneral)

Wdg Initial Timeout

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

Table 4-17. Attribute WdgInitialTimeout (WdgGeneral) detailed description

Property	Value
Label	Wdg Initial Timeout [s]
Type	FLOAT
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	0
Invalid	Range <=65.535 >=0

4.4.9 WdgMaxTimeout (WdgGeneral)

Wdg Max Timeout

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

Table 4-18. Attribute WdgMaxTimeout (WdgGeneral) detailed description

Property	Value
Label	Wdg Max Timeout [s]
Type	FLOAT
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	0
Invalid	Range <=65.535 >=0

4.4.10 WdgRunArea (WdgGeneral)

Wdg Run Area

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

Table 4-19. Attribute WdgRunArea (WdgGeneral) detailed description

Property	Value
Label	Wdg Run Area
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ROM
Range	RAM ROM

4.4.11 WdgTriggerLocation (WdgGeneral)

Wdg Trigger Location

Location (memory address) of the watchdog trigger routine.

Note

Not supported by the current hardware.

Table 4-20. Attribute WdgTriggerLocation (WdgGeneral) detailed description

Property	Value
Label	Wdg Trigger Location
Type	FUNCTION-NAME
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	NULL_PTR
Enable	false

4.4.12 WdgCallbackNotification0 (WdgGeneral)

WdgCallbackNotification0

Callback notification for the ISR Wdg_Swt0_Isr function

Table 4-21. Attribute WdgCallbackNotification0 (WdgGeneral) detailed description

Property	Value
Label	Wdg Instance 0 CallBack Notification
Type	FUNCTION-NAME
Origin	Custom
Symbolic Name	false
Default	NULL_PTR

4.4.13 WdgCallbackNotification1 (WdgGeneral)

WdgCallbackNotification1

Callback notification for the ISR Wdg_Swt1_Isr function

Table 4-22. Attribute WdgCallbackNotification1 (WdgGeneral) detailed description

Property	Value
Label	Wdg Instance 1 CallBack Notification
Type	FUNCTION-NAME
Origin	Custom
Symbolic Name	false
Default	NULL_PTR

4.4.14 WdgCallbackNotification2 (WdgGeneral)

WdgCallbackNotification2

Callback notification for the ISR Wdg_Swt2_Isr function

Table 4-23. Attribute WdgCallbackNotification2 (WdgGeneral) detailed description

Property	Value
Label	Wdg Instance 2 CallBack Notification
Type	FUNCTION-NAME
Origin	Custom
Symbolic Name	false
Default	NULL_PTR

4.4.15 WdgVersionInfoApi (WdgGeneral)

Wdg VersionInfo Api

Compile switch to enable / disable the version information API. **True**: API enabled **False**:API disabled

Table 4-24. Attribute WdgVersionInfoApi (WdgGeneral) detailed description

Property	Value
Label	Provide Version Info API
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.5 Form WdgPublishedInformation

WdgPublishedInformation

Container holding all Wdg specific published information parameters

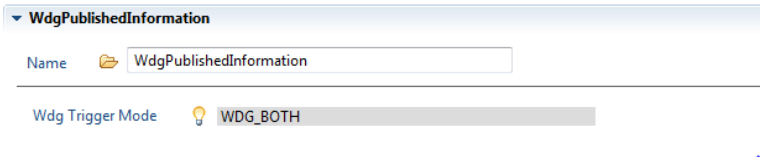

Figure 4-4. Tresos Plugin snapshot for WdgPublishedInformation form.

4.5.1 WdgTriggerMode (WdgPublishedInformation)

Wdg Trigger Mode

Watchdog trigger mode (toggle/window/both).

Table 4-25. Attribute WdgTriggerMode (WdgPublishedInformation) detailed description

Property	Value
Label	Wdg Trigger Mode
Type	ENUMERATION_LABEL
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	WDG_BOTH
Range	WDG_BOTH WDG_TOGGLE WDG_WINDOW

4.6 Form CommonPublishedInformation

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

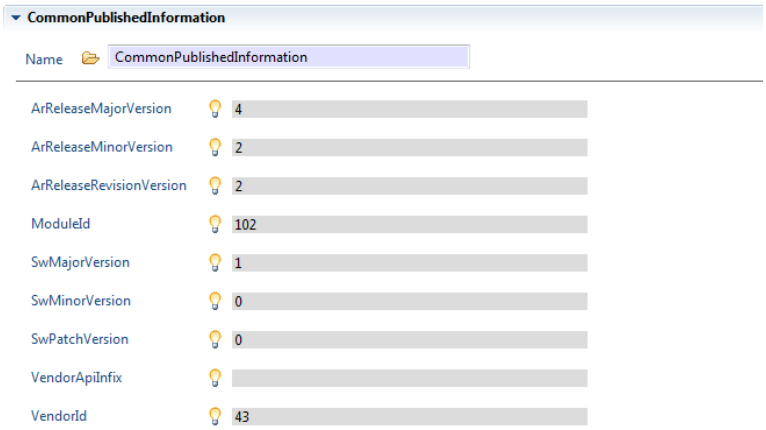

Figure 4-5. Tresos Plugin snapshot for CommonPublishedInformation form.

4.6.1 ArReleaseMajorVersion (CommonPublishedInformation)

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

Table 4-26. Attribute ArReleaseMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Major Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	4
Invalid	Range >=4 <=4

4.6.2 ArReleaseMinorVersion (CommonPublishedInformation)

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

Table 4-27. Attribute ArReleaseMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range >=2 <=2

4.6.3 ArReleaseRevisionVersion (CommonPublishedInformation)

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

Table 4-28. Attribute ArReleaseRevisionVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Release Revision Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range >=2 <=2

4.6.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

Table 4-29. Attribute ModuleId (CommonPublishedInformation) detailed description

Property	Value
Label	Module Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false

Table continues on the next page...

Table 4-29. Attribute ModuleId (CommonPublishedInformation) detailed description (continued)

Property	Value
Default	102
Invalid	Range >=102 <=102

4.6.5 SwMajorVersion (CommonPublishedInformation)

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

Table 4-30. Attribute SwMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Major Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range >=1 <=1

4.6.6 SwMinorVersion (CommonPublishedInformation)

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

Table 4-31. Attribute SwMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range >=0 <=0

4.6.7 SwPatchVersion (CommonPublishedInformation)

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

Table 4-32. Attribute SwPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Patch Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range >=0 <=0

4.6.8 VendorApiInfix (CommonPublishedInformation)

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><Api name from SWS>. 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.

Table 4-33. Attribute VendorApiInfix (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Api Infix
Type	STRING_LABEL
Origin	Custom
Symbolic Name	false
Default	
Enable	false

4.6.9 VendorId (CommonPublishedInformation)

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

Table 4-34. Attribute VendorId (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	43
Invalid	Range >=43 <=43

4.7 Form WdgClockReferencePoint

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 SIRC running at 128KHz frequency), then configure in MCU a reference point of SIRC type with 128KHz frequency. If the cip is not trimmed (SIRC running at frequency different than 128KHz), then configure in MCU a reference point of CUSTOM type with the real SIRC frequency measured on the cip.

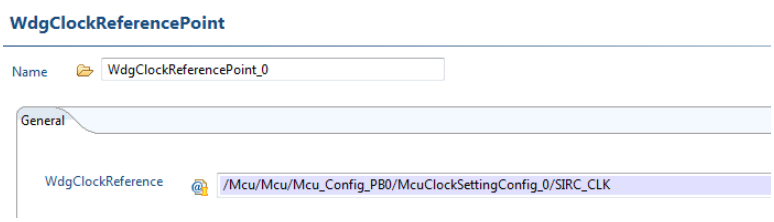

Figure 4-6. Tresos Plugin snapshot for WdgClockReferencePoint form.

4.7.1 WdgClockReference (WdgClockReferencePoint)

WdgClockReference

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

Table 4-35. Attribute WdgClockReference (WdgClockReferencePoint) detailed description

Property	Value
Label	WdgClockReference
Type	REFERENCE
Origin	Custom
Symbolic Name	false

4.8 Form WdgSettingsConfig

WdgSettingsConfig

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

Included forms :

- Form WdgExternalConfiguration
- Form WdgSettingsFast
- Form WdgSettingsOff
- Form WdgSettingsSlow

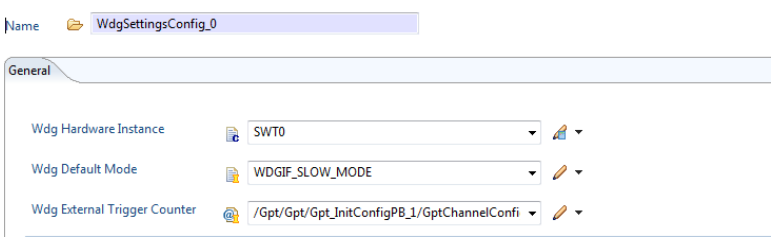

Figure 4-7. Tresos Plugin snapshot for WdgSettingsConfig form.

4.8.1 WdgInstance (WdgSettingsConfig)

Wdg Default Mode

Default mode for watchdog driver initialization.

Table 4-36. Attribute WdgInstance (WdgSettingsConfig) detailed description

Property	Value
Label	Wdg Hardware Instance
Type	ENUMERATION

Table continues on the next page...

Table 4-36. Attribute WdgInstance (WdgSettingsConfig) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false

4.8.2 WdgDefaultMode (WdgSettingsConfig)

Wdg Default Mode

Default mode for watchdog driver initialization.

Table 4-37. Attribute WdgDefaultMode (WdgSettingsConfig) detailed description

Property	Value
Label	Wdg Default Mode
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	WDGIF_SLOW_MODE
Range	WDGIF_FAST_MODE WDGIF_OFF_MODE WDGIF_SLOW_MODE

4.8.3 WdgExternalTriggerCounterRef (WdgSettingsConfig)

Wdg External Trigger Counter

Reference to either **- a GptChannel** Reference sed for the watchdog servicing routine implementation

Table 4-38. Attribute WdgExternalTriggerCounterRef (WdgSettingsConfig) detailed description

Property	Value
Label	Wdg External Trigger Counter
Type	CHOICE-REFERENCE
Origin	Custom

4.8.4 WdgInterruptContentEnable (WdgSettingsConfig)

Wdg Interrupt Enable

This parametter is used to generate interrupt content for each SWT. True = Interrupt content is generated. False = Interrupt content is not generated.

Table 4-39. Attribute WdgExternalTriggerCounterRef (WdgSettingsConfig) detailed description

Property	Value
Label	Wdg Interrupt Enable
Type	BOOLEAN
Origin	Custom
Default	true

4.8.5 Form WdgExternalConfiguration

WdgExternalConfiguration

Configuration items for an external watchdog hardware

Is included by form : Form WdgSettingsConfig

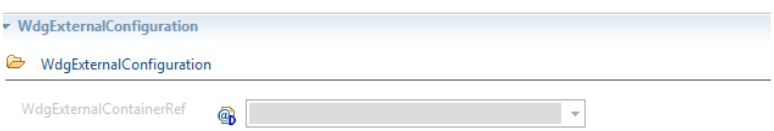

Figure 4-8. Tresos Plugin snapshot for WdgExternalConfiguration form.

4.8.5.1 WdgExternalContainerRef (WdgExternalConfiguration)

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

Table 4-40. Attribute WdgExternalContainerRef (WdgExternalConfiguration) detailed description

Property	Value
Type	CHOICE-REFERENCE
Origin	AUTOSAR_ECUC
Enable	false

4.8.6 Form WdgSettingsFast

WdgSettingsFast

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

Is included by form : Form WdgSettingsConfig

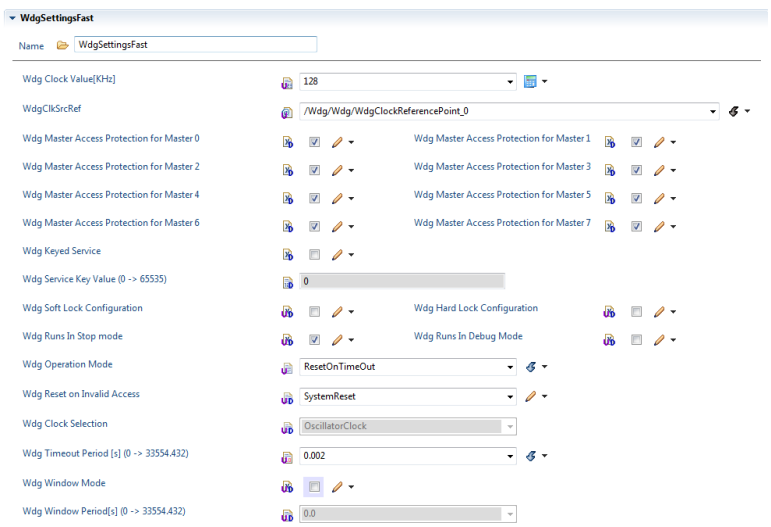

Figure 4-9. Tresos Plugin snapshot for WdgSettingsFast form.

4.8.6.1 WdgClockValue

Wdg Clock Value

This is the Implementation Specific parameter.Indicates Wdg Clock Value in KHz

Table 4-41. Attribute WdgClockValue detailed description

Property	Value
Label	Wdg Clock Value[KHz]
Type	INTEGER

Table continues on the next page...

Table 4-41. Attribute WdgClockValue detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false

4.8.6.2 WdgClkSrcRef

WdgClkSrcRef

Reference to the WdgClockReferencePoint from which the clock is derived

Table 4-42. WdgClkSrcRef detailed description

Property	Value
Label	WdgClkSrcRef
Type	REFERENCE
Origin	Custom
Symbolic Name	false

4.8.6.3 WdgMasterAccessProtectionforMaster0 (WdgSettingsFast)

Master Access Protection for Master 0.

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-43. Attribute WdgMasterAccessProtectionforMaster0 (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 0
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.6.4 WdgMasterAccessProtectionforMaster1 (WdgSettingsFast)

Master Access Protection for Master 1

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-44. Attribute WdgMasterAccessProtectionforMaster1 (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 1
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.6.5 WdgMasterAccessProtectionforMaster2 (WdgSettingsFast)

Master Access Protection for Master 2

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-45. Attribute WdgMasterAccessProtectionforMaster2 (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 2
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.6.6 WdgMasterAccessProtectionforMaster3 (WdgSettingsFast)

Master Access Protection for Master 3

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-46. Attribute WdgMasterAccessProtectionforMaster3 (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 3
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.6.7 WdgMasterAccessProtectionforMaster4 (WdgSettingsFast)

Master Access Protection for Master 4

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-47. Attribute WdgMasterAccessProtectionforMaster4 (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 4
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.6.8 WdgMasterAccessProtectionforMaster5 (WdgSettingsFast)

Master Access Protection for Master 5

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-48. Attribute WdgMasterAccessProtectionforMaster5 (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 5
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.6.9 WdgMasterAccessProtectionforMaster6 (WdgSettingsFast)

Master Access Protection for Master 6

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-49. Attribute WdgMasterAccessProtectionforMaster6 (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 6
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.6.10 WdgMasterAccessProtectionforMaster7 (WdgSettingsFast)

Master Access Protection for Master 7

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-50. Attribute WdgMasterAccessProtectionforMaster7 (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 7
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.6.11 WdgKeyedService (WdgSettingsFast)

Fixed Service Sequence or Keyed Service Mode

. False = Fixed Service Sequence, the fixed sequence 0xA602, 0xB480 is used to service the watchdog. True = Keyed Service Mode, two pseudorandom key values are used to service the watchdog.

Note

This is an Implementation Specific Parameter.

Table 4-51. Attribute WdgKeyedService (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Keyed Service
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.8.6.12 WdgServiceKeyValue (WdgSettingsFast)

Service Key

This value is the initial service key value used in keyed service mode. If SWT_CR[KEY] is set, the next key value to be written to the SWT_SR is (17*SK+3) mod 2^16.

Table 4-52. Attribute WdgServiceKeyValue (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Service Key Value

Table continues on the next page...

Table 4-52. Attribute WdgServiceKeyValue (WdgSettingsFast) detailed description (continued)

Property	Value
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range ../WdgKeyedService <=65535 >=0

4.8.6.13 WdgSoftLockConfiguration (WdgSettingsFast)

SoftLockConfiguration

This is the Implementation Specific parameter.Soft Lock for the Software Watchdog Timer Control (SWTCR) Register

- Enabled : SWTCR can be read or written.
- Disabled: SWTCR can be read only. An unlock sequence should be written into service register before this register can again be written.

The setting of this switch is intended to prevent accidental writes of the SWTCR from changing the defined system watchdog configuration.

Table 4-53. Attribute WdgSoftLockConfiguration (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Soft Lock Configuration
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Enable	true

4.8.6.14 WdgHardLockConfiguration (WdgSettingsFast)

Wdg Read Only

This is the Implementation Specific parameter.HardLock for the Software Watchdog Timer Control (SWTCR) Register

- Enabled : SWTCR can be read or written.
- Disabled: SWTCR can be read only. A system reset is required before this register can again be written.

The setting of this switch is intended to prevent accidental writes of the SWTCR from changing the defined system watchdog configuration.

Table 4-54. Attribute WdgHardLockConfiguration (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Hard Lock Configuration
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Enable	true

4.8.6.15 WdgRunsInStopMode (WdgSettingsFast)

Wdg Runs in Stop Mode

This is the Implementation Specific parameter.

- Enabled: SWT continues to count even while the processor core is in stop mode.
- Disabled: SWT 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.

Table 4-55. Attribute WdgRunsInStopMode (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Runs In Stop mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.6.16 WdgRunsInDebugMode (WdgSettingsFast)

Wdg Runs In Debug Mode

This is the Implementation Specific parameter.

- Enabled: SWT continues to count even while the device enters the debug mode.
- Disabled: SWT stops counting if the processor core when the device enters the debug mode

Table 4-56. Attribute WdgRunsInDebugMode (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Runs In Debug Mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.8.6.17 WdgOperationMode (WdgSettingsFast)

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, reset on a second consecutive time-out.

Table 4-57. Attribute WdgOperationMode (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Operation Mode
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	ResetOnTimeOut
Range	ResetOnTimeOut Interrupt

4.8.6.18 WdgResetOnInvalidAccess (WdgSettingsFast)

Wdg Reset On Invalid Access

If window mode is enabled, the service sequence must be performed in the last part of the window time out period. The window is open when the down counter is less than the value in the SWT_WN register. Outside of this window, service sequence writes are invalid access and generate:

- BUS error: Invalid access to the SWT generated a bus error.
- System Reset: Invalid access to the SWT causes a system reset (if wathdog is enabled).

Table 4-58. Attribute WdgResetOnInvalidAccess (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Reset On Invalid Access
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	BusError
Range	BusError SystemReset

4.8.6.19 WdgClockSelection (WdgSettingsFast)

Wdg Clock Selection

The unique SWT counter clock is the undivided low power internal oscillator (IRC16Mhz).

Table 4-59. Attribute WdgClockSelection (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Clock Selection
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	OscillatorClock
Range	OscillatorClock SystemClock

4.8.6.20 WdgTimeoutPeriod (WdgSettingsFast)

Wdg Timeout Period

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

Table 4-60. Attribute WdgTimeoutPeriod (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Timeout Period [s]
Type	FLOAT
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range <=33554.432 >=0.000016

4.8.6.21 WdgWindowMode (WdgSettingsFast)

Wdg WindowMode.

- Disabled: Regular mode, service sequence can be done at any time.
- Enabled : Windowed mode, the service sequence is only valid when the down counter is less than value in the SWTWN register.

Table 4-61. Attribute WdgWindowMode (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Window Mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.8.6.22 WdgWindowPeriod (WdgSettingsFast)

Wdg Window Period

This is the Implementation Specific parameter. Window start value. When window mode is enabled, the service sequence can only be written when the internal down counter is less than this value.

Table 4-62. Attribute WdgWindowPeriod (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Window Period[s]
Type	FLOAT
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range <=33554.432 >=0

4.8.7 Form WdgSettingsOff

WdgSettingsOff

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

Is included by form : Form WdgSettingsConfig

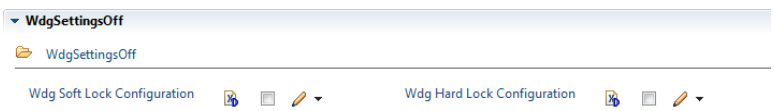

Figure 4-10. Tresos Plugin snapshot for WdgSettingsOff form.

4.8.7.1 WdgSoftLockConfiguration (WdgSettingsOff)

SoftLockConfiguration

This is the Implementation Specific parameter.Soft Lock for the Software Watchdog Timer Control (SWTCR) Register

- Enabled : SWTCR can be read or written.
- Disabled: SWTCR can be read only. An unlock sequence should be written into service register before this register can again be written.

The setting of this switch is intended to prevent accidental writes of the SWTCR from changing the defined system watchdog configuration.

Table 4-63. Attribute WdgSoftLockConfiguration (WdgSettingsOff) detailed description

Property	Value
Label	Wdg Soft Lock Configuration

Table continues on the next page...

Table 4-63. Attribute WdgSoftLockConfiguration (WdgSettingsOff) detailed description (continued)

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Enable	true

4.8.7.2 WdgHardLockConfiguration (WdgSettingsOff)

Wdg Read Only

This is the Implementation Specific parameter.HardLock for the Software Watchdog Timer Control (SWTCR) Register

- Enabled : SWTCR can be read or written.
- Disabled: SWTCR can be read only. A system reset is required before this register can again be written.

The setting of this switch is intended to prevent accidental writes of the SWTCR from changing the defined system watchdog configuration.

Table 4-64. Attribute WdgHardLockConfiguration (WdgSettingsOff) detailed description

Property	Value
Label	Wdg Hard Lock Configuration
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Enable	true

4.8.8 Form WdgSettingsSlow

WdgSettingsSlow

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

Is included by form : Form WdgSettingsConfig

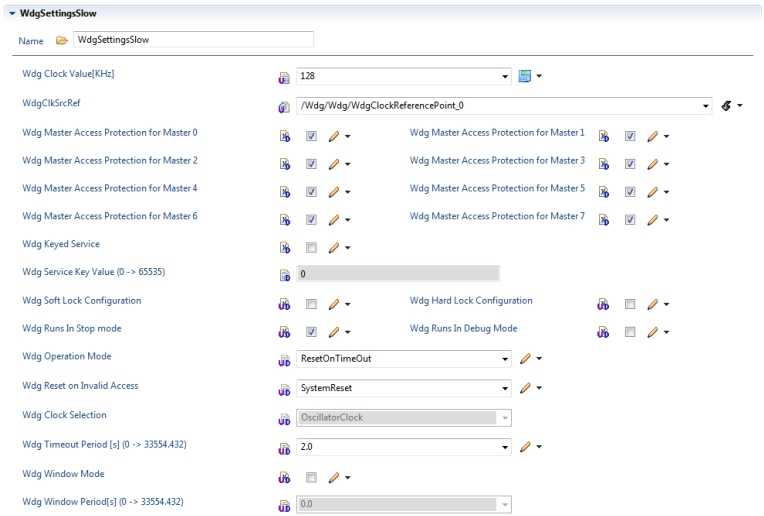

Figure 4-11. Tresos Plugin snapshot for WdgSettingsSlow form.

4.8.8.1 WdgClockValue

Wdg Clock Value

This is the Implementation Specific parameter.Indicates Wdg Clock Value in KHz

Table 4-65. Attribute WdgClockValue detailed description

Property	Value
Label	Wdg Clock Value[KHz]
Type	INTEGER
Origin	Custom
Symbolic Name	false

4.8.8.2 WdgClkSrcRef

WdgClkSrcRef

Reference to the WdgClockReferencePoint from which the clock is derived

Table 4-66. WdgClkSrcRef detailed description

Property	Value
Label	WdgClkSrcRef
Type	REFERENCE
Origin	Custom
Symbolic Name	false

4.8.8.3 WdgMasterAccessProtectionforMaster0 (WdgSettingsSlow)

Master Access Protection for Master 0.

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-67. Attribute WdgMasterAccessProtectionforMaster0 (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 0
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.8.4 WdgMasterAccessProtectionforMaster1 (WdgSettingsSlow)

Master Access Protection for Master 1

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-68. Attribute WdgMasterAccessProtectionforMaster1 (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 1
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.8.5 WdgMasterAccessProtectionforMaster2 (WdgSettingsSlow)

Master Access Protection for Master 2

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-69. Attribute WdgMasterAccessProtectionforMaster2 (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 2
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.8.6 WdgMasterAccessProtectionforMaster3 (WdgSettingsSlow)

Master Access Protection for Master 3

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-70. Attribute WdgMasterAccessProtectionforMaster3 (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 3
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.8.7 WdgMasterAccessProtectionforMaster4 (WdgSettingsSlow)

Master Access Protection for Master 4

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-71. Attribute WdgMasterAccessProtectionforMaster4 (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 4
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.8.8 WdgMasterAccessProtectionforMaster5 (WdgSettingsSlow)

Master Access Protection for Master 5

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-72. Attribute WdgMasterAccessProtectionforMaster5 (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 5
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.8.9 WdgMasterAccessProtectionforMaster6 (WdgSettingsSlow)

Master Access Protection for Master 6

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-73. Attribute WdgMasterAccessProtectionforMaster6 (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 6
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.8.10 WdgMasterAccessProtectionforMaster7 (WdgSettingsSlow)

Master Access Protection for Master 7

. False = Access for the master is not enabled. True = Access for the master is enabled

Note

This is an Implementation Specific Parameter.

Table 4-74. Attribute WdgMasterAccessProtectionforMaster7 (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Master Access Protection for Master 7
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.8.11 WdgKeyedService (WdgSettingsSlow)

Fixed Service Sequence or Keyed Service Mode

. False = Fixed Service Sequence, the fixed sequence 0xA602, 0xB480 is used to service the watchdog. True = Keyed Service Mode, two pseudorandom key values are used to service the watchdog.

Note

This is an Implementation Specific Parameter.

Table 4-75. Attribute WdgKeyedService (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Keyed Service
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.8.8.12 WdgServiceKeyValue (WdgSettingsSlow)

Service Key

This value is the initial service key value used in keyed service mode. If SWT_CR[KEY] is set, the next key value to be written to the SWT_SR is (17*SK+3) mod 2^16.

Table 4-76. Attribute WdgServiceKeyValue (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Service Key Value
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range ../WdgKeyedService <=65535 >=0

4.8.8.13 WdgSoftLockConfiguration (WdgSettingsSlow)

SoftLockConfiguration

This is the Implementation Specific parameter.Soft Lock for the Software Watchdog Timer Control (SWTCR) Register

- Enabled : SWTCR can be read or written.
- Disabled: SWTCR can be read only. An unlock sequence should be written into service register before this register can again be written.

The setting of this switch is intended to prevent accidental writes of the SWTCR from changing the defined system watchdog configuration.

Table 4-77. Attribute WdgSoftLockConfiguration (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Soft Lock Configuration
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Enable	true

4.8.8.14 WdgHardLockConfiguration (WdgSettingsSlow)

Wdg Read Only

This is the Implementation Specific parameter.HardLock for the Software Watchdog Timer Control (SWTCR) Register

- Enabled : SWTCR can be read or written.
- Disabled: SWTCR can be read only. A system reset is required before this register can again be written.

The setting of this switch is intended to prevent accidental writes of the SWTCR from changing the defined system watchdog configuration.

Table 4-78. Attribute WdgHardLockConfiguration (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Hard Lock Configuration
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Enable	true

4.8.8.15 WdgRunsInStopMode (WdgSettingsSlow)

Wdg Runs in Stop Mode

This is the Implementation Specific parameter.

- Enabled: SWT continues to count even while the processor core is in stop mode.
- Disabled: SWT 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.

Table 4-79. Attribute WdgRunsInStopMode (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Runs In Stop mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.8.8.16 WdgRunsInDebugMode (WdgSettingsSlow)

Wdg Runs In Debug Mode

This is the Implementation Specific parameter.

- Enabled: SWT continues to count even while the device enters the debug mode.
- Disabled: SWT stops counting if the processor core when the device enters the debug mode.

Table 4-80. Attribute WdgRunsInDebugMode (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Runs In Debug Mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.8.8.17 WdgOperationMode (WdgSettingsSlow)

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, reset on a second consecutive time-out.

Table 4-81. Attribute WdgOperationMode (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Operation Mode
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	ResetOnTimeOut
Range	ResetOnTimeOut Interrupt

4.8.8.18 WdgResetOnInvalidAccess (WdgSettingsSlow)

Wdg Reset On Invalid Access

If window mode is enabled, the service sequence must be performed in the last part of the window time out period. The window is open when the down counter is less than the value in the SWT_WN register. Outside of this window, service sequence writes are invalid access and generate:

- BUS error: Invalid access to the SWT generated a bus error.
- System Reset: Invalid access to the SWT causes a system reset (if wathdog is enabled).

Table 4-82. Attribute WdgResetOnInvalidAccess (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Reset On Invalid Access
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	BusError
Range	BusError SystemReset

4.8.8.19 WdgClockSelection (WdgSettingsSlow)

Wdg Clock Selection

The unique SWT counter clock is the undivided low power internal oscillator (IRC16Mhz).

Table 4-83. Attribute WdgClockSelection (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Clock Selection
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	OscillatorClock
Range	OscillatorClock SystemClock

4.8.8.20 WdgTimeoutPeriod (WdgSettingsSlow)

Wdg Timeout Period

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

Table 4-84. Attribute WdgTimeoutPeriod (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Timeout Period [s]
Type	FLOAT
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range <=33554.432 >=0.000016

4.8.8.21 WdgWindowMode (WdgSettingsSlow)

Wdg WindowMode.

- Disabled: Regular mode, service sequence can be done at any time.
- Enabled : Windowed mode, the service sequence is only valid when the down counter is less than value in the SWTWN register.

Table 4-85. Attribute WdgWindowMode (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Window Mode

Table continues on the next page...

Table 4-85. Attribute WdgWindowMode (WdgSettingsSlow) detailed description (continued)

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.8.8.22 WdgWindowPeriod (WdgSettingsSlow)

Wdg Window Period

This is the Implementation Specific parameter. Window start value. When window mode is enabled, the service sequence can only be written when the internal down counter is less than this value.

Table 4-86. Attribute WdgWindowPeriod (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Window Period[s]
Type	FLOAT
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range <=33554.432 >=0

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number UM35WDGASR4.2 Rev0002 RTM
1.0.0
Revision 2.3.0

AUTOSAR