
User Manual

for MPC5634M WDG Driver

Document Number: UM14WDGASR3.0R2.0.0

Rev. 1.1





Contents

Section Number	Title	Page
Chapter 1		
Revision History		
Chapter 2		
Introduction		
2.1	Supported Derivatives.....	9
2.2	Overview.....	9
2.3	About this Manual.....	10
2.4	Acronyms and Definitions.....	10
2.5	Reference List.....	11
Chapter 3		
Driver		
3.1	Requirements.....	13
3.2	Driver Design Summary.....	13
3.3	Deviation from Requirements.....	14
3.4	Software specification.....	16
3.4.1	Enum Reference.....	16
3.4.1.1	Enumeration Wdg_StatusType.....	16
3.4.2	Function Reference.....	17
3.4.2.1	Function Wdg_GetVersionInfo.....	17
3.4.2.2	Function Wdg_Init.....	17
3.4.2.3	Function Wdg_SetMode.....	18
3.4.2.4	Function Wdg_Trigger.....	19
3.4.3	Structs Reference.....	19
3.4.3.1	Structure Wdg_ConfigType.....	19
Chapter 4		
Tresos Configuration Plug-in		
4.1	Configuration elements of Wdg.....	21
4.2	Form IMPLEMENTATION_CONFIG_VARIANT.....	21

Section Number	Title	Page
4.3	Form WdgModeConfig.....	22
4.3.1	WdgDefaultMode (WdgModeConfig).....	22
4.3.2	WdgKeyedService (WdgModeConfig).....	23
4.3.3	WdgServiceKeyValue (WdgModeConfig).....	23
4.3.4	Form WdgSettingsOff.....	24
4.3.4.1	WdgReadOnly (WdgSettingsOff).....	24
4.3.5	Form WdgSettingsSlow.....	25
4.3.5.1	WdgReadOnly (WdgSettingsSlow).....	25
4.3.5.2	WdgWindowMode (WdgSettingsSlow).....	26
4.3.5.3	WdgRunsInStopMode (WdgSettingsSlow).....	27
4.3.5.4	WdgRunsInDebugMode (WdgSettingsSlow).....	27
4.3.5.5	WdgOperationMode (WdgSettingsSlow).....	28
4.3.5.6	WdgResetOnInvalidAccess (WdgSettingsSlow).....	28
4.3.5.7	WdgClockSelection (WdgSettingsSlow).....	29
4.3.5.8	WdgTimeoutPeriod (WdgSettingsSlow).....	29
4.3.5.9	WdgWindowPeriod (WdgSettingsSlow).....	30
4.3.6	Form WdgSettingsFast.....	30
4.3.6.1	WdgReadOnly (WdgSettingsFast).....	31
4.3.6.2	WdgWindowMode (WdgSettingsFast).....	31
4.3.6.3	WdgRunsInStopMode (WdgSettingsFast).....	32
4.3.6.4	WdgRunsInDebugMode (WdgSettingsFast).....	32
4.3.6.5	WdgOperationMode (WdgSettingsFast).....	33
4.3.6.6	WdgResetOnInvalidAccess (WdgSettingsFast).....	34
4.3.6.7	WdgClockSelection (WdgSettingsFast).....	34
4.3.6.8	WdgTimeoutPeriod (WdgSettingsFast).....	35
4.3.6.9	WdgWindowPeriod (WdgSettingsFast).....	35
4.4	Form WdgExternalConfiguration.....	36
4.4.1	WdgExternalContainerRef (WdgExternalConfiguration).....	36

Section Number	Title	Page
4.5	Form WdgGeneral.....	37
4.5.1	WdgDevErrorDetect (WdgGeneral).....	37
4.5.2	WdgDisableAllowed (WdgGeneral).....	38
4.5.3	WdgIndex (WdgGeneral).....	38
4.5.4	WdgTriggerLocation (WdgGeneral).....	39
4.5.5	WdgVersionInfoApi (WdgGeneral).....	39
4.6	Form CommonPublishedInformation.....	40
4.6.1	ArMajorVersion (CommonPublishedInformation).....	40
4.6.2	ArMinorVersion (CommonPublishedInformation).....	41
4.6.3	ArPatchVersion (CommonPublishedInformation).....	41
4.6.4	ModuleId (CommonPublishedInformation).....	42
4.6.5	SwMajorVersion (CommonPublishedInformation).....	42
4.6.6	SwMinorVersion (CommonPublishedInformation).....	43
4.6.7	SwPatchVersion (CommonPublishedInformation).....	43
4.6.8	VendorApiInfix (CommonPublishedInformation).....	44
4.6.9	VendorId (CommonPublishedInformation).....	44
4.7	Form WdgPublishedInformation.....	45
4.7.1	WdgMaxTimeout (WdgPublishedInformation).....	45
4.7.2	WdgMinTimeout (WdgPublishedInformation).....	46
4.7.3	WdgResolution (WdgPublishedInformation).....	47
4.7.4	WdgTriggerMode (WdgPublishedInformation).....	47



Chapter 1

Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0	02-Feb-2011	Salvatore Nappi	Updates for Monaco 1.5 MByte automatic documentation
1.1	21-Dec-2011	Swarupa Chaudhury	Updated for Monaco RTM 2.0.0



Chapter 2

Introduction

This User Manual describes Freescale Semiconductor AUTOSAR Watchdog (Wdg) for MPC5634M .

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 intended to be used with the following microcontroller devices of Freescale Semiconductor .

Table 2-1. MPC5634M Derivatives

Freescale Semiconductor	mpc5634m_bga208, mpc5634m_qfp144, mpc5634m_qfp176
-------------------------	--

All of the above microcontroller devices are collectively named as MPC5634M .

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

Term	Definition
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
ASM	Assembler
BSMI	Basic Software Make file Interface
CAN	Controller Area Network
DEM	Diagnostic Event Manager
DET	Development Error Tracer
C/CPP	C and C++ Source Code
VLE	Variable Length Encoding
N/A	Not Applicable
MCU	Micro Controller Unit

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 3.0Wdg Driver Software Specification Document.	V2.2.0 R3.0 Rev 0001
2	MPC5634M Reference Manual	Rev. 6, 4 October 2011

Chapter 3

Driver

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 3.0Wdg Driver Software Specification document (See Table [Reference List](#)).

3.2 Driver Design Summary

A Software Watchdog Timer (SWT) with programmable interrupt response is available in MPC5634M . 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
- 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
- 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 Deviation from Requirements

The driver deviates from the AUTOSAR FR Driver software specification in some places. Table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, or out of scope for the FR driver. Table [Table 3-1](#) provides Status column description.

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
WDG013	N/A	Additional errors that are detected because of specific implementation and/or specific hardware properties shall be added in the module's implementation documentation. The classification and enumeration shall be compatible to the errors listed above [WDG010]	No additional errors are supported by hardware, hence no additional errors are implemented.

Table continues on the next page...

Table 3-2. Driver Deviations Table (continued)

Requirement	Status	Description	Notes
WDG034	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
WDG040	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	
WDG075	N/F	Published information – If the watchdog hardware provides a uniform timeout resolution over the complete range, this resolution and the minimum and maximum timeout periods that can be selected shall be given. If the timeout resolution is not uniform a list of all possible timeout periods has to be provided.	The timeout resolution is not uniform. The minimum timeout period is 1.6E-5 ms, the maximum timeout period is 268.435456 ms.
WDG076	N/S	External watchdog driver To access the external watchdog hardware, the Wdg module shall use the functionality and API of the corresponding handler or driver, e.g. the SPI handler or DIO driver.	External module is customer dependant and is not yet developed.
WDG077	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.
WDG078	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.
WDG104	N/A	The Wdg module's environment shall make sure that the Wdg module has been initialized before the Wdg_Trigger routine is called.	Not a watchdog requirement.
WDG113	N/S	WdgExternalContainerRef container definition	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
WDG118	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.
WDG128	N/A	Description of the WdgTimeoutList container	In conformance with the WDG075, "if the watchdog hardware provides a uniform timeout resolution over the complete range, this resolution and the minimum and maximum timeout periods that can be selected shall be given. If the timeout resolution is not uniform a list of all possible timeout periods has to be provided". The driver support uniform resolution and therefor this parameter is not applicable
WDG129	N/A	Description of the WfgTlmeoutPeriod parameter of the WdgTimeoutList container	In conformance with the WDG075, "if the watchdog hardware provides a uniform timeout resolution over the complete range, this resolution and the minimum and maximum timeout periods that can be selected shall be given. If the timeout resolution is not uniform a list of all possible timeout periods has to be provided". The driver support uniform resolution and therefor this parameter is not applicable

3.4 Software specification

The following sections contains driver software specifications.

3.4.1 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR Wdg Driver software specification Version 3.0 .

3.4.1.1 Enumeration Wdg_StatusType

This enumerated type will contain the watchdog driver's possible states.

Table 3-3. Enumeration Wdg_StatusType Values

Name	Initializer	Description
WDG_UNINIT	0x00	The watchdog driver is not uninitialized.
WDG_IDLE		= 0x01 The watchdog driver is currently idle, i.e not being triggered or mode changed .
WDG_BUSY		= 0x02 The watchdog driver is currently busy, i.e triggered or switched between modes .

3.4.2 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR Wdg Driver software specification Version 3.0 .

3.4.2.1 Function Wdg_GetVersionInfo

Returns the version information of the module.

Details:

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

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

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

Satisfied Requirements: WDG068.

Prototype: `void Wdg_GetVersionInfo(Std_VersionInfoType *versioninfo);`

Table 3-4. Wdg_GetVersionInfo Arguments

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

3.4.2.2 Function Wdg_Init

This function initializes the WDG module.

Details:

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.

Satisfied Requirements: WDG064, WDG063, WDG065.

Prototype: `void Wdg_Init(const Wdg_ConfigType *ConfigPtr);`

Table 3-5. Wdg_Init Arguments

Type	Name	Direction	Description
<code>const Wdg_ConfigType *</code>	<code>ConfigPtr</code>	input	Pointer to configuration set.

3.4.2.3 Function `Wdg_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 the `Wdg_SetMode` function:

- `WDGIF_OFF_MODE`,
- `WDGIF_SLOW_MODE`,
- `WDGIF_FAST_MODE`.

Satisfied Requirements: WDG064, WDG031, WDG063, WDG065.

Prototype: `Std_ReturnType Wdg_SetMode(WdgIf_ModeType Mode);`

Table 3-6. Wdg_SetMode Arguments

Type	Name	Direction	Description
<code>WdgIf_ModeType</code>	<code>Mode</code>	input	One of the following statically configured modes: <code>WDGIF_OFF_MODE</code> , <code>WDGIF_SLOW_MODE</code> , <code>WDGIF_FAST_MODE</code> .

Return : `Std_ReturnType`.

Table 3-7. Wdg_SetMode Return Values

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

3.4.2.4 Function Wdg_Trigger

This function triggers the watchdog hardware.

Details:

It has to be called cyclically by some upper layer function (usually the watchdog manager) in order to prevent the watchdog hardware from expiring.

Pre: The Wdg module's environment shall make sure that the Wdg module has been initialized before the routine is called. .

Satisfied Requirements: WDG036, WDG064, WDG063.

Prototype: `void Wdg_Trigger(void);`

3.4.3 Structs Reference

Data structures supported by the driver are as per AUTOSAR Wdg Driver software specification Version 3.0 .

3.4.3.1 Structure Wdg_ConfigType

Structure to hold the watchdog driver configuration set.

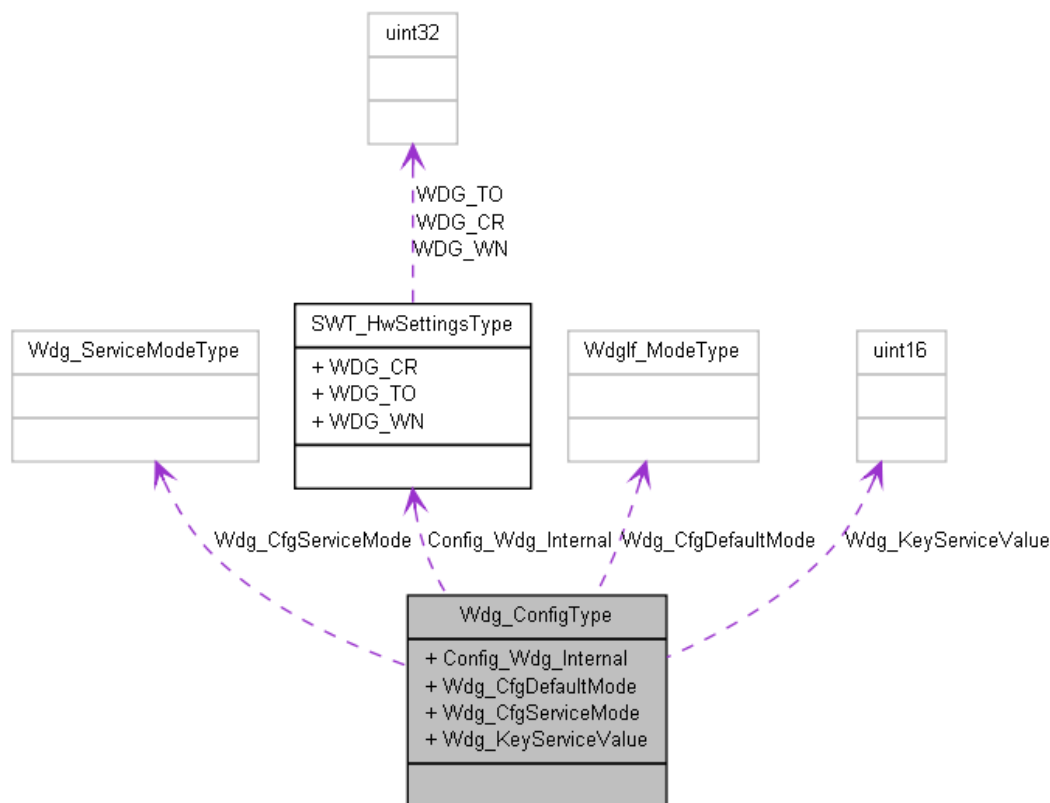


Figure 3-1. Struct Wdg_ConfigType

Details:

Used for pointers to structures holding configuration data provided to the Wdg module initialization routine for configuration of the module and watchdog hardware.

Declaration:

```
typedef struct
{
    SWT_HwSettingsType Config_Wdg_Internal[3],
    WdgIf_ModeType Wdg_CfgDefaultMode,
    Wdg_ServiceModeType Wdg_CfgServiceMode,
    Wdg_ServiceKeyWidthType Wdg_KeyServiceValue
} Wdg_ConfigType;
```

Table 3-8. Structure Wdg_ConfigType member description

Member	Description
Config_Wdg_Internal	The configuration for watchdog driver on SWT.
Wdg_CfgDefaultMode	The default mode for watchdog driver's configuration.
Wdg_CfgServiceMode	The type of the trigger mode for watchdog driver on SWT.
Wdg_KeyServiceValue	Initial value of the service key.

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
- WdgExternalConfiguration
- WdgGeneral
- CommonPublishedInformation
- WdgPublishedInformation
- WdgModeConfig

Table 4-1. Revision table

Revision	Date
1.0	2009-10-13T17:30:00

4.2 Form IMPLEMENTATION_CONFIG_VARIANT

Configuration classes

Enable the parameters that are editable for specific configuration classes

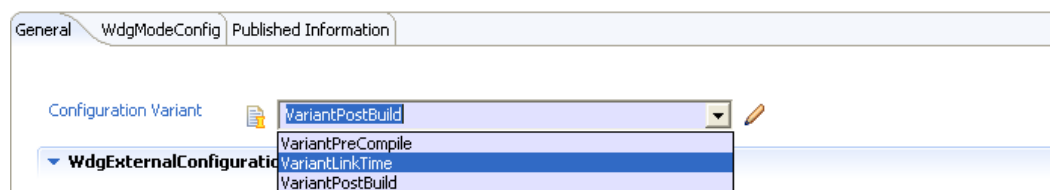


Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION_CONFIG_VARIANT form.

Table 4-2. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description

Property	Value
Label	Configuration Variant
Default	VariantPreCompile
Range	VariantPreCompile VariantLinkTime VariantPostBuild

4.3 Form WdgModeConfig

WdgModeConfig

Configuration items for the different watchdog modes.

Included forms :

- WdgSettingsOff
- WdgSettingsSlow
- WdgSettingsFast

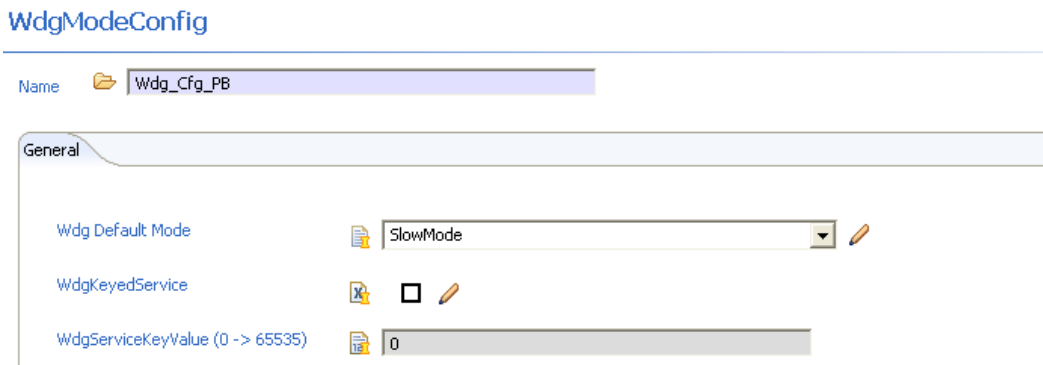


Figure 4-2. Tresos Plugin snapshot for WdgModeConfig form.

4.3.1 WdgDefaultMode (WdgModeConfig)

Wdg Default Mode

Default mode for watchdog driver initialization.

Table 4-3. Attribute WdgDefaultMode (WdgModeConfig) detailed description

Property	Value
Label	Wdg Default Mode
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	SlowMode
Lower Multiplicity	1
Upper Multiplicity	1
Range	FastMode OffMode SlowMode

4.3.2 WdgKeyedService (WdgModeConfig)

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 4-4. Attribute WdgKeyedService (WdgModeConfig) detailed description

Property	Value
Label	Wdg Keyed Service
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.3.3 WdgServiceKeyValue (WdgModeConfig)

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 $(17*SK+3) \bmod 2^{16}$.

Table 4-5. Attribute WdgServiceKeyValue (WdgModeConfig) detailed description

Property	Value
Label	Wdg Service Key Value
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range ../WdgKeyedService ≤65535 ≥0

4.3.4 Form WdgSettingsOff

WdgSettingsOff

Hardware dependent settings for the watchdog driver's off mode

Is included by form : WdgModeConfig

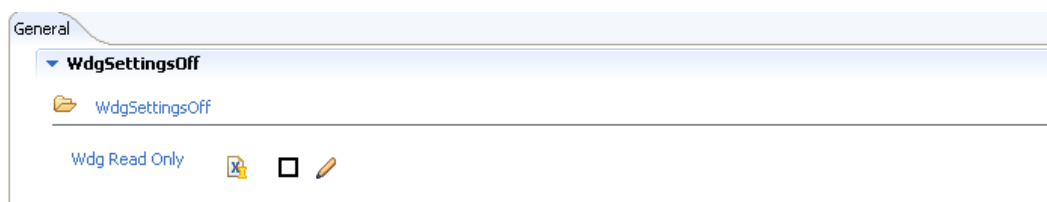


Figure 4-3. Tresos Plugin snapshot for WdgSettingsOff form.

4.3.4.1 WdgReadOnly (WdgSettingsOff)

WdgReadOnly

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. 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 4-6. Attribute WdgReadOnly (WdgSettingsOff) detailed description

Property	Value
Label	Wdg Read Only
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.3.5 Form WdgSettingsSlow

WdgSettingsSlow

Hardware dependent settings for the watchdog driver's slow mode

Is included by form : WdgModeConfig

The screenshot shows the 'WdgSettingsSlow' form with the following settings:

- Wdg Read Only:** ☐
- Wdg Window Mode:** ☐
- Wdg Runs In Stop mode:** ☒
- Wdg Runs In Debug Mode:** ☒
- Wdg Operation Mode:** ResetOnTimeOut
- Wdg Reset On Invalid Access:** BusError
- Wdg Clock Selection:** OscillatorClock
- Wdg Timeout Period [clock cycles] (256 -> 4294967295):** 2500.0
- Wdg Window Period[clock cycles] (0 -> 4294967295):** 0.0

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

4.3.5.1 WdgReadOnly (WdgSettingsSlow)

Wdg Read Only

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. 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 4-7. Attribute WdgReadOnly (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Read Only
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.3.5.2 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 4-8. Attribute WdgWindowMode (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Window Mode
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.3.5.3 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 4-9. Attribute WdgRunsInStopMode (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Runs In Stop mode
Origin	Custom
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.3.5.4 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 4-10. Attribute WdgRunsInDebugMode (WdgSettingsSlow) detailed description

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

Table continues on the next page...

Table 4-10. Attribute WdgRunsInDebugMode (WdgSettingsSlow) detailed description (continued)

Property	Value
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.3.5.5 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 4-11. Attribute WdgOperationMode (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Operation Mode
Origin	Custom
Symbolic Name	false
Default	ResetOnTimeOut
Lower Multiplicity	1
Upper Multiplicity	1
Range	ResetOnTimeOut Interrupt

4.3.5.6 WdgResetOnInvalidAccess (WdgSettingsSlow)

Wdg Reset On Invalid Acces

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 4-12. Attribute WdgResetOnInvalidAccess (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Reset On Invalid Access
Origin	Custom
Symbolic Name	false
Default	BusError
Lower Multiplicity	1
Upper Multiplicity	1
Range	BusError SystemReset

4.3.5.7 WdgClockSelection (WdgSettingsSlow)

Wdg Clock Selection

The SWT counter clock may be the oscillator or the system clock.

Table 4-13. Attribute WdgClockSelection (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Clock Selection
Origin	Custom
Symbolic Name	false
Default	OscillatorClock
Lower Multiplicity	1
Upper Multiplicity	1
Range	OscillatorClock SystemClock

4.3.5.8 WdgTimeoutPeriod (WdgSettingsSlow)

Wdg Timeout Period

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

Table 4-14. Attribute WdgTimeoutPeriod (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Timeout Period [clock cycles]
Origin	Custom
Symbolic Name	false
Default	256
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <=4294967295 >=256

4.3.5.9 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 4-15. Attribute WdgWindowPeriod (WdgSettingsSlow) detailed description

Property	Value
Label	Wdg Window Period[clock cycles]
Origin	Custom
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <=4294967295 >=0

4.3.6 Form WdgSettingsFast

WdgSettingsSlow

Hardware dependent settings for the watchdog driver's fast mode

Is included by form : WdgModeConfig

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

4.3.6.1 WdgReadOnly (WdgSettingsFast)

Wdg Read Only

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. 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 4-16. Attribute WdgReadOnly (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Read Only
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.3.6.2 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 4-17. Attribute WdgWindowMode (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Window Mode
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.3.6.3 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 4-18. Attribute WdgRunsInStopMode (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Runs In Stop mode
Origin	Custom
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.3.6.4 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 4-19. Attribute WdgRunsInDebugMode (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Runs In Debug Mode
Origin	Custom
Symbolic Name	false
Default	false
Lower Multiplicity	1
Upper Multiplicity	1

4.3.6.5 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 4-20. Attribute WdgOperationMode (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Operation Mode
Origin	Custom
Symbolic Name	false
Default	ResetOnTimeOut
Lower Multiplicity	1
Upper Multiplicity	1
Range	ResetOnTimeOut Interrupt

4.3.6.6 WdgResetOnInvalidAccess (WdgSettingsFast)

Wdg Reset On Invalid Acces

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 4-21. Attribute WdgResetOnInvalidAccess (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Reset On Invalid Access
Origin	Custom
Symbolic Name	false
Default	BusError
Lower Multiplicity	1
Upper Multiplicity	1
Range	BusError SystemReset

4.3.6.7 WdgClockSelection (WdgSettingsFast)

Wdg Clock Selection

The SWT counter clock may be the oscillator or the system clock.

Table 4-22. Attribute WdgClockSelection (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Clock Selection
Origin	Custom
Symbolic Name	false
Default	OscillatorClock
Lower Multiplicity	1

Table continues on the next page...

Table 4-22. Attribute WdgClockSelection (WdgSettingsFast) detailed description (continued)

Property	Value
Upper Multiplicity	1
Range	OscillatorClock SystemClock

4.3.6.8 WdgTimeoutPeriod (WdgSettingsFast)

Wdg Timeout Period

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

Table 4-23. Attribute WdgTimeoutPeriod (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Timeout Period [clock cycles]
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	256
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <=4294967295 >=256

4.3.6.9 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 4-24. Attribute WdgWindowPeriod (WdgSettingsFast) detailed description

Property	Value
Label	Wdg Window Period[clock cycles]

Table continues on the next page...

Table 4-24. Attribute WdgWindowPeriod (WdgSettingsFast) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <=4294967295 >=0

4.4 Form WdgExternalConfiguration

WdgExternalConfiguration

Configuration items for an external watchdog hardware

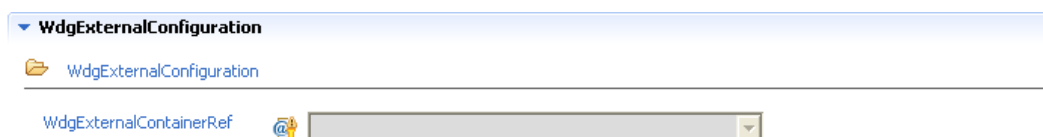


Figure 4-6. Tresos Plugin snapshot for WdgExternalConfiguration form.

4.4.1 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 4-25. Attribute WdgExternalContainerRef (WdgExternalConfiguration) detailed description

Property	Value
Type	CHOICE-REFERENCE
Origin	AUTOSAR_ECUC
Lower Multiplicity	1
Upper Multiplicity	1

4.5 Form WdgGeneral

WdgGeneral

All general parameters of the watchdog driver are collected here

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

4.5.1 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 4-26. Attribute WdgDevErrorDetect (WdgGeneral) detailed description

Property	Value
Label	Development Error Detection
Type	BOOLEAN
Origin	AUTOSAR_ECUC

Table continues on the next page...

Table 4-26. Attribute WdgDevErrorDetect (WdgGeneral) detailed description (continued)

Property	Value
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.5.2 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 allow

Table 4-27. Attribute WdgDisableAllowed (WdgGeneral) detailed description

Property	Value
Label	Wdg Disable Allowed
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.5.3 WdgIndex (WdgGeneral)

Wdg Index

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

Table 4-28. Attribute WdgIndex (WdgGeneral) detailed description

Property	Value
Label	Wdg Index
Type	INTEGER
Origin	AUTOSAR_ECUC

Table continues on the next page...

Table 4-28. Attribute WdgIndex (WdgGeneral) detailed description (continued)

Property	Value
Symbolic Name	true
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range <div style="margin-left: 20px;"><=254</div> <div style="margin-left: 20px;">>=0</div>

4.5.4 WdgTriggerLocation (WdgGeneral)

Wdg Trigger Location

Location (memory address) of the watchdog trigger routine.

Note

Not supported by the current hardware.

Table 4-29. Attribute WdgTriggerLocation (WdgGeneral) detailed description

Property	Value
Label	Wdg Trigger Location
Type	FUNCTION-NAME
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	
Lower Multiplicity	1
Upper Multiplicity	1

4.5.5 WdgVersionInfoApi (WdgGeneral)

Wdg VersionInfo Api

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

Table 4-30. Attribute WdgVersionInfoApi (WdgGeneral) detailed description

Property	Value
Label	Provide Version Info API
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true
Lower Multiplicity	1
Upper Multiplicity	1

4.6 Form CommonPublishedInformation

CommonPublishedInformation

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



CommonPublishedInformation

CommonPublishedInformation

- AUTOSAR Major Version 2
- AUTOSAR Minor Version 2
- AUTOSAR Patch Version 1
- Numeric Module ID 102
- Software Major Version 1
- Software Minor Version 9
- Software Patch Version 0
- Vendor Api Infix
- Vendor ID 27

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

4.6.1 ArMajorVersion (CommonPublishedInformation)

AUTOSAR Major Version

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

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

Property	Value
Label	AUTOSAR Major Version
Origin	Custom
Symbolic Name	false
Default	2
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=2 <=2

4.6.2 ArMinorVersion (CommonPublishedInformation)

AUTOSAR Minor Version

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

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

Property	Value
Label	AUTOSAR Minor Version
Origin	Custom
Symbolic Name	false
Default	2
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=2 <=2

4.6.3 ArPatchVersion (CommonPublishedInformation)

AUTOSAR Patch Version

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

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

Property	Value
Label	AUTOSAR Patch Version
Origin	Custom
Symbolic Name	false
Default	1
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range ≥ 1 ≤ 1

4.6.4 ModuleId (CommonPublishedInformation)

Module ID

Module ID of this module.

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

Property	Value
Label	Numeric Module ID
Origin	Custom
Symbolic Name	false
Default	102
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range ≥ 102 ≤ 102

4.6.5 SwMajorVersion (CommonPublishedInformation)

Software Major Version

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

Table 4-35. Attribute SwMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Major Version
Origin	Custom
Symbolic Name	false
Default	1
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=1 <=1

4.6.6 SwMinorVersion (CommonPublishedInformation)

Software Minor Version

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

Table 4-36. Attribute SwMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Minor Version
Origin	Custom
Symbolic Name	false
Default	9
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=9 <=9

4.6.7 SwPatchVersion (CommonPublishedInformation)

Software Patch Version

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

**Table 4-37. Attribute SwPatchVersion (CommonPublishedInformation)
detailed description**

Property	Value
Label	Software Patch Version
Origin	Custom
Symbolic Name	false
Default	0
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range >=0 <=0

4.6.8 VendorApiInfix (CommonPublishedInformation)

Vendor Api Infix

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 4-38. Attribute VendorApiInfix (CommonPublishedInformation)
detailed description**

Property	Value
Label	Vendor Api Infix
Origin	Custom
Symbolic Name	false
Default	
Lower Multiplicity	0
Upper Multiplicity	1
Enable	false

4.6.9 VendorId (CommonPublishedInformation)

Vendor ID

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

Table 4-39. Attribute VendorId (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor ID
Origin	Custom
Symbolic Name	false
Default	27
Lower Multiplicity	1
Upper Multiplicity	1
Invalid	Range ≥ 27 ≤ 27

4.7 Form WdgPublishedInformation

WdgPublishedInformation

Container holding all Wdg specific published information parameters

Included forms :

- WdgTimeoutList

WdgPublishedInformation

WdgPublishedInformation

Wdg Max Timeout [clock cycles] 4.294967295E9

Wdg Min Timeout [clock cycles] 256.0

Wdg Resolution

Wdg Trigger Mode WDG_BOTH

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

4.7.1 WdgMaxTimeout (WdgPublishedInformation)

Wdg Max Timeout

Maximum timeout period in [clock cycles].

The value of this field is computed to be $(2^{32}-1) \times \text{WdgResolution}$. $2^{32}-1$ (where 2^{32} is 2 power 32) indicates the maximum value of clock cycles that is able to be written as timeout

Table 4-40. Attribute WdgMaxTimeout (WdgPublishedInformation) detailed description

Property	Value
Label	Wdg Max Timeout [clock cycles]
Type	FLOAT_LABEL
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	4294967295
Enable	true
Invalid	Range ≥ 4294967295 ≤ 4294967295

4.7.2 WdgMinTimeout (WdgPublishedInformation)

Wdg Min Timeout

Minimum timeout period in [clock cycles].

The value of this field is computed to be $256 \times \text{WdgResolution}$. 256 indicates the minimum value of clock cycles that can be written as timeout.

Table 4-41. Attribute WdgMinTimeout (WdgPublishedInformation) detailed description

Property	Value
Label	Wdg Min Timeout [clock cycles]
Type	FLOAT_LABEL
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	256
Enable	true

Table continues on the next page...

Table 4-41. Attribute WdgMinTimeout (WdgPublishedInformation) detailed description (continued)

Property	Value
Invalid	Range ≥ 256 ≤ 256

4.7.3 WdgResolution (WdgPublishedInformation)

Wdg Resolution

The Hardware provides a uniform timeout resolution over the complete range. The watchdog clock may be the oscillator clock or the system clock. Depending on the this clock the resolution can be calculated to be $1/\text{Frequency}$, where the frequency is the clock frequency selected to be the clock for the watchdog. Note: The frequency should be in Hz.

Table 4-42. Attribute WdgResolution (WdgPublishedInformation) detailed description

Property	Value
Label	Wdg Resolution
Type	FLOAT_LABEL
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	
Enable	true

4.7.4 WdgTriggerMode (WdgPublishedInformation)

Wdg Trigger Mode

Watchdog trigger mode (toggle/window/both).

Table 4-43. Attribute WdgTriggerMode (WdgPublishedInformation) detailed description

Property	Value
Label	Wdg Trigger Mode
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false

Table continues on the next page...

Table 4-43. Attribute WdgTriggerMode (WdgPublishedInformation) detailed description (continued)

Property	Value
Default	WDG_BOTH
Lower Multiplicity	1
Upper Multiplicity	1
Range	WDG_BOTH WDG_TOGGLE WDG_WINDOW

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor 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. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2011 Freescale Semiconductor, Inc.



AUTOSAR and AUTOSAR logo are registered trademarks of AUTOSAR GbR (www.autosar.org)