

MICROSAR Wdg

Technical Reference

TLE4278G

Version 4.00.00

Authors	Matthias Scheid
Status	Released

Document Information

History

Author	Date	Version	Remarks
Matthias Scheid	2015-01-26	2.0.0	Creation of Technical Reference for component Wdg_30_TLE4278, based on Watchdog core (only supporting TLE7368)
Matthias Scheid	2015-06-15	2.1.0	Added support of device type Infineon TLE4278G
Matthias Scheid	2015-09-01	2.1.1	Added list of compatible devices
Matthias Scheid	2016-09-05	3.0.0	Adaption to SafeBSW process
Matthias Scheid	2017-04-04	4.0.0	Generic support of IO-based external watchdog peripherals

Reference Documents

No.	Source	Title	Version
[1]	AUTOSAR	AUTOSAR_SWS_WatchdogDriver.pdf	2.5.0
[2]	AUTOSAR	AUTOSAR_BasicSoftwareModules.pdf	1.0.0
[3]	AUTOSAR	AUTOSAR_SWS_SPIHandlerDriver.pdf	3.2.0
[4]	Vector	Technical_Reference_MICROSAR_Wdg_general.pdf	3.1.0
[5]	Infineon	Infineon-TLE4278-DS-v01_04-en.pdf	Rev. 1.4

Scope of the Document

This technical reference describes the specific use of the Wdg_30_TLE4278G driver software. It supplements the general Wdg driver technical reference [4].



Caution

We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector's release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

Contents

1	Component History	5
1.1	Hardware overview	6
2	Introduction.....	7
2.1	Architecture Overview	7
3	Functional Description	8
3.1	Features	8
3.1.1	Limitations.....	8
3.1.1.1	Supported watchdog modes	8
3.1.1.2	Optional irreversible activation of watchdog	8
3.2	Initialization	8
4	Integration.....	9
4.1	Scope of Delivery.....	9
4.2	Hardware Software Interface (HSI)	9
4.3	DIO configuration	9
4.3.1	DIO driver compatibility	9
4.3.1.1	Mapping of API to write DIO level to a channel	9
4.3.1.2	Mapping of API to read DIO level from a channel.....	10
4.4	Configure the module.....	10
4.4.1	Setting up the device.....	10
4.4.1.1	Configuration of device pins.....	10
4.4.1.2	Configuration of DIO Mode Settings.....	11
4.5	Support of watchdog modes	11
4.6	Triggering.....	11
4.6.1	Configuration of Gpt.....	11
4.6.2	Configuration of triggering	11
5	API Description.....	13
5.1	Type Definitions	13
5.2	Services used by Wdg	14
6	Glossary and Abbreviations	16
6.1	Abbreviations	16
7	Contact.....	17

Illustrations

Figure 2-1	AUTOSAR 4.x Architecture Overview	7
------------	---	---

Tables

Table 1-1	Component history.....	5
Table 1-2	Overview of reference hardware	6
Table 5-1	Type Definitions	13
Table 5-2	Wdg_30_TLE4278G_ConfigType	14
Table 5-3	Wdg_30_TLE4278G_ModeDioLevelAssignmentType	14
Table 5-4	Services used by the Wdg driver	15
Table 6-1	Abbreviations.....	16

1 Component History

The component history gives an overview over the important milestones that are supported in the different versions of the component.

Component Version	New Features
1.00.00	Initial creation of watchdog module based on ASR 4.0.3
2.00.00	Development according to SafeBSW process
4.00.00	Breaking Change in GenTool R14
5.00.00	Development according to SafeBSW
6.00.00	Generic support of IO-based external watchdog peripherals

Table 1-1 Component history

1.1 Hardware overview

Table 1-2 shows the reference hardware which is supported by this driver. This table is not complete.

Derivatives	Hardware Manual
Infineon TLE4278G	Infineon-TLE4278-DS-v01_04-en.pdf; Data Sheet Rev 1.4; 2007-02-19
Infineon TLE7368	Infineon-TLE7368-DS-v02_01-en.pdf, Data Sheet Rev 2.1; 2010-11-22
Maxim Integrated MAX6369	MAX6369-MAX6374.pdf, 19-1676; Rev 7; 1/16

Table 1-2 Overview of reference hardware

2 Introduction

This document describes the functionality, API and configuration of the AUTOSAR BSW module Wdg as specified in [1].

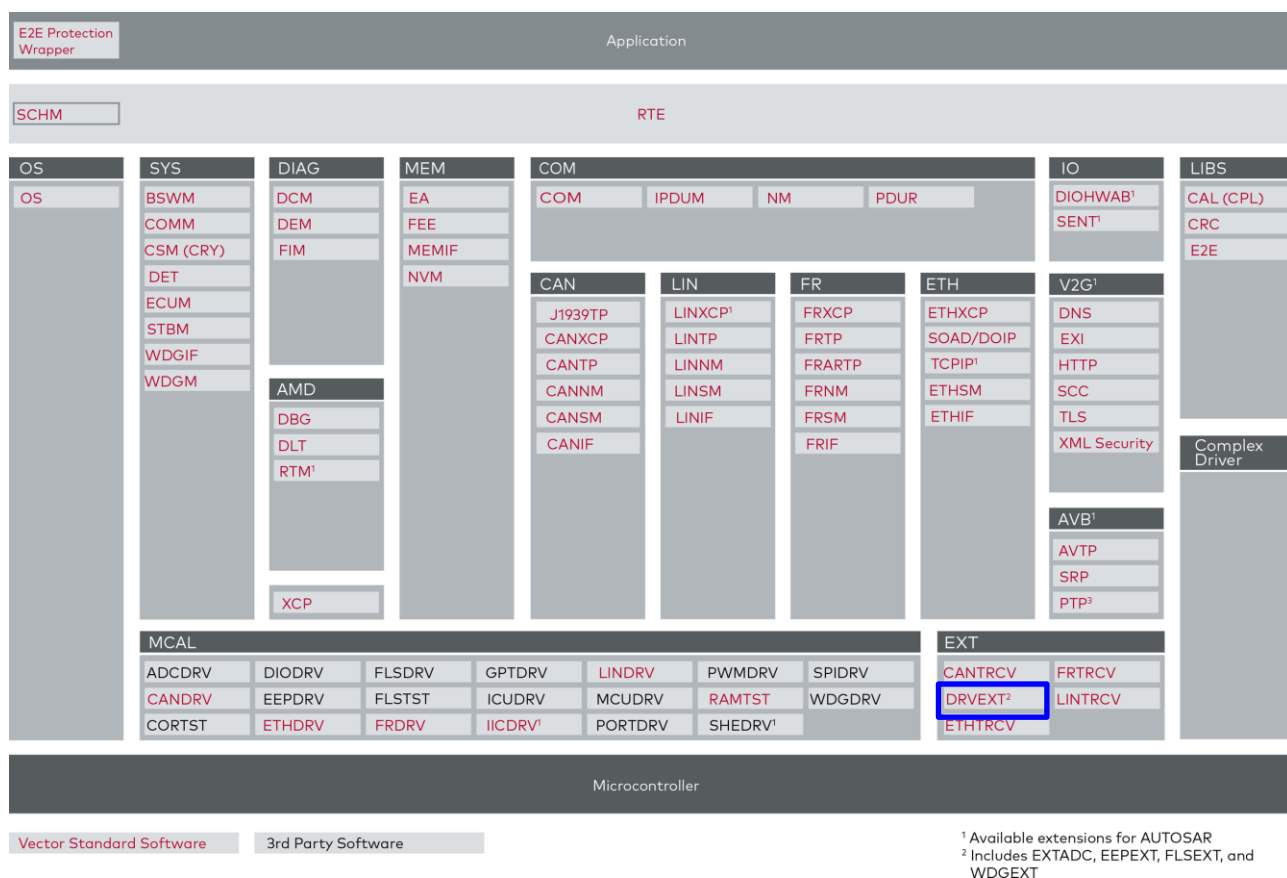
Supported AUTOSAR Release*:	4.0.3	
Supported Configuration Variants:	pre-compile	
Vendor ID:	Wdg_VENDOR_ID	30 decimal (= Vector-Informatik, according to HIS)
Module ID:	Wdg_MODULE_ID	102 decimal (according to ref. [2])

* For the detailed functional specification please also refer to the corresponding AUTOSAR SWS.

The module provides services for initializing the window watchdog of Infineon TLF35584. Furthermore it supports changing watchdog timing configuration and triggering of the watchdog.

2.1 Architecture Overview

The following figure shows where the Wdg is located in the AUTOSAR architecture.



¹ Available extensions for AUTOSAR
² Includes EXTADC, EEPEXT, FLSEXT, and WDGEXT

Figure 2-1 AUTOSAR 4.x Architecture Overview

3 Functional Description

3.1 Features

The module supports the features specified in Autosar (see [1]) as described in [4].

3.1.1 Limitations

3.1.1.1 Supported watchdog modes

Due to hardware limitations the supported watchdog modes depend on the configuration .

Depending on the configuration, the request of an unsupported mode will lead to DEM event `E_MODE_FAILED` or a DET error.

3.1.1.2 Optional irreversible activation of watchdog

Due to hardware limitations of some watchdog peripherals the module supports the an irreversible activation of the watchdog peripheral. If this feature is enabled a request of `WDGIF_OFF_MODE` will result in reporting the DEM event `E_MODE_FAILED` once the was successfully set to `WDGIF_FAST_MODE` or `WDGIF_SLOW_MODE`.

3.2 Initialization

The Wdg driver module is initialized as described in [4].

4 Integration

This chapter gives necessary information for the integration of the MICROSAR Wdg into an application environment of an ECU.

4.1 Scope of Delivery

The delivery of the Wdg contains the files which are described in [4]. An infix to all file names, APIs and memory sections is applied. (E.g. `Wdg.c` is infixed to `Wdg_30_TLE4278G.c`, `Wdg_Init()` is infixed to `Wdg_30_TLE4278G_Init()`).



FAQ

Although the infix TLE4278G implies that this watchdog driver only supports Infineon TLE4278G many other peripherals are supported. This is a generic driver for external DIO watchdog which can be adapted to different hardware requirements via configuration.

4.2 Hardware Software Interface (HSI)

This driver uses DIO Channels to control the external watchdog peripheral. The driver can be adapted to the requirements of the hardware.

Watchdog related DIO Channels must be used exclusively by this watchdog driver.

Chapter 1.1 describes the supported reference hardware.

4.3 DIO configuration

The Wdg driver performs hardware access by calling service functions of the lower layer component DIO driver. Therefore the DIO driver must be configured to allow access to the Wdg driver. This means that DIO channels must be defined prior using the Wdg driver.

4.3.1 DIO driver compatibility

For compatibility reasons the Wdg driver provides the possibility to configure the interface towards the AUTOSAR DIO driver by inserting corresponding defines into the user configuration file of the Wdg driver module (see parameter `WdgUserConfigFile` in DaVinci Configurator 5).

4.3.1.1 Mapping of API to write DIO level to a channel

The Wdg driver calls the AUTOSAR SPI API `Dio_WriteChannel` as default function to set the output for a specific channel. If it is necessary (e.g. for compatibility reasons) to overwrite the function call, the following example code must be added:

```
#define Wdg_30_TLE4278G_DioWriteChannel Appl_DioWriteChannel
```

4.3.1.2 Mapping of API to read DIO level from a channel

The Wdg driver calls the Autosar SPI API `Dio_ReadChannel` as default function to read the output level for a specific channel. If it is necessary (e.g. for compatibility reasons) to overwrite the function call, the following example code must be added:

```
#define Wdg_30_TLE4278G_DioReadChannel Appl_DioReadChannel
```

**Note**

If the described API remapping is used it is required to add the prototypes of the target functions to the module. (E.g. by inclusion of a header file or external declaration.)

4.4 Configure the module

Since different watchdog hardware is supported by this driver it is necessary to adapt the configuration according to the connected external watchdog peripheral.

4.4.1 Setting up the device

**Note**

The module comes with several preconfigured device configurations. These may be compatible with other external watchdog peripherals. Additionally it is possible to use the preconfigured devices as template.

4.4.1.1 Configuration of device pins

The IO pins which are required to operate the external watchdog device have to be added to the watchdog device configuration.

Each device at least requires a trigger pin. Depending on the signal edge that is recognized as a valid trigger signal the initial level of this pin has to be configured.

Depending on the configuration of the initial level the first edge that is produced during trigger routine is a

- ▶ falling edge (`WdgInitilLevel = STD_HIGH`)
- ▶ rising edge (`WdgInitilLevel = STD_LOW`)

**Note**

It is also recommended to set the initial pin levels in the respective port configuration.

Each added pin directly references a Dio Channel from the DIO driver module.

If the external watchdog peripheral can be controlled by additional IO pins (e.g. en-/disabled, set temporal behavior) these also have to be added. The additional pins have to be referenced by `WdgDioModeSettings` which then again get referenced by a watchdog mode `WdgSettings[Fast/Slow/Off]`.

4.4.1.2 Configuration of DIO Mode Settings

For external watchdog devices which can be controlled by IO pins different mode settings (`WdgDioModeSettings`) can be configured.

Therefore different mode settings can be added which reference the configured control pins (see 4.4.1.1).

If the connected external device can only be enabled once (due to hardware limitations) the option `WdgEnableIrreversible` has to be enabled. Any request of `WDGIF_OFF_MODE` after a transition to `WDGIF_[FAST/SLOW]_MODE` will be denied and the DEM event `EModeFailed` reported.

4.5 Support of watchdog modes

To support a mode it is required to configure the associated reference (e.g. `WdgDioModeSettingsRef`) in the appropriate container (`WdgMode[Fast/Slow/Off]`).

`WDGIF_OFF_MODE` and `WDGIF_FAST_MODE` are only available if the above-mentioned reference exists. If the reference is not configured any requests to this mode will be denied with the reporting of a DET error / DEM event.

`WDGIF_FAST_MODE` always is available, even without a referencing an instance of `WdgDioModeSettingsRef`. This is the correct configuration in case the connected external watchdog peripheral always runs automatically and can't be turned off.

4.6 Triggering

4.6.1 Configuration of Gpt

The Wdg driver module uses a Gpt Channel for trigger timing. This `GptChannel` has to be configured in Channel Mode `GPT_CH_MODE_CONTINUOUS`.

The notification parameter (`GptNotification`) has to be configured to the watchdog callback service `Wdg_30_TLE4278G_Cbk_GptNotificationTrigger()`.

4.6.2 Configuration of triggering

The trigger timing is configured in the respective container `WdgSettings[Fast/Slow/Off]`. The configured trigger time value represents the cyclical period of the trigger procedure. This procedure always consists of a falling and a rising edge (triggering and recovering) on the referenced trigger pin (`WdgTriggerPinRef`). The times of the edges within the trigger period are adaptable via the configurable duty cycle.

Depending on which edge is required as trigger pin the initial level has to be set (see 4.4.1.1).

5 API Description

For a detailed API description see [4].

5.1 Type Definitions

Types defined by the Wdg are described in this chapter.

Type Name	C-Type	Description	Value Range
Wdg_30_TLE4278G_ConfigType	struct	This is the configuration structure. A pointer to this structure has to be provided to the API <code>Wdg_Init</code>	-
Wdg_30_TLE4278G_ModeDioLevelAssignmentType	struct	This structure is used to assign a specific physical level to a referenced Dio Channel. Such assignments are used for mode-specific configuration of external watchdog peripheral.	-

Table 5-1 Type Definitions

Table 5-2 describes the members of `Wdg_30_TLE4278G_ConfigType` which also is described

Struct Element Name	C-Type	Description	Value Range
EModeFailed	Dem_EventType	This member is the DEM error corresponding to BSWMD parameter <code>WDG_E_MODE_FAILED</code>	uint16 (0-65535)
EDisableRejected	Dem_EventType	This member is the DEM error corresponding to BSWMD parameter <code>WDG_E_DISABLE_REJECTED</code>	uint16 (0-65535)
DefaultMode	WdgIf_ModeType	Mode which should be set during initialization.	WDGIF_OFF_MODE WDGIF_SLOW_MODE WDGIF_FAST_MODE
DioPinWdiChannel	Dio_ChannelType	Dio Channel that is connected to watchdog input pin of device.	Implementation specific
DioPinWdiInitLevel	Dio_LevelType	Physical level to which DioPinWdiChannel is set during module initialization.	STD_LOW STD_HIGH
DioModeLevelsFast	Wdg_30_TLE4278G_ModeDioLevelAssign	Dio Level assignments to set Wdg hardware to	STD_LOW STD_HIGH

Struct Element Name	C-Type	Description	Value Range
	nmentType	WDGIF_FAST_MODE	
DioModeLevelsSlow	Wdg_30_TLE4278G_ModeDioLevelAssignmentType	Dio Level assignments to set Wdg hardware to WDGIF_SLOW_MODE	STD_LOW STD_HIGH
DioModeLevelsOff	Wdg_30_TLE4278G_ModeDioLevelAssignmentType	Wdg hardware to WDGIF_OFF_MODE	Implementation specific
TriggerTimer	Gpt_ChannelType	GptChannel that is used for trigger timing	Implementation specific
TriggerTimerCounter[2]	Gpt_ValueType	Trigger cycle (in Gpt ticks)	Implementation specific
RecoverTimerCounter	Gpt_ValueType	Recover cycle (in Gpt ticks)	Implementation specific
TriggerCycleDuration[2]	uint16	Duration of trigger cycle (in ms)	uint16 (0-65535)

Table 5-2 Wdg_30_TLE4278G_ConfigType

Table 5-3 describes the members of Wdg_30_TLE4278G_ModeDioLevelAssignmentType.

Type Name	C-Type	Description	Value Range
channel	Dio_ChannelType	Dio Channel which's level needs to be adapted for a mode specific configuration of the external watchdog hardware	Implementation specific
level	Dio_LevelType	Physical level which should be set to referenced Dio Channel for configuration of the external watchdog hardware.	STD_LOW STD_HIGH

Table 5-3 Wdg_30_TLE4278G_ModeDioLevelAssignmentType

5.2 Services used by Wdg

Additional to the services described in see [4] the module uses the following services

Component	API
DIO	Dio_WriteChannel
Gpt	Gpt_StartTimer Gpt_StopTimer Gpt_EnableNotification

Table 5-4 Services used by the Wdg driver

6 Glossary and Abbreviations

6.1 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
CPHA	Clock phase
CPOL	Clock polarity
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DIO	Digital Input / Output
ECU	Electronic Control Unit
GPT	General Purpose Timer
ISR	Interrupt Service Routine
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)
SPI	Serial Peripheral Interface
SRS	Software Requirement Specification
SWC	Software Component
SWS	Software Specification
WEN / WDEN	Watchdog Enable
WDI	Watchdog Input
WDO	Watchdog output

Table 6-1 Abbreviations

7 Contact

Visit our website for more information on

- > News
- > Products
- > Demo software
- > Support
- > Training data
- > Addresses

www.vector.com