Document Title	Requirements on PWM Driver
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	113

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R24-11

Document Change History

Date	Release	Changed by	Description
2024-11-27	R24-11	AUTOSAR Release Management	• No content changes
2023-11-23	R23-11	AUTOSAR Release Management	• No content changes
2022-11-24	R22-11	AUTOSAR Release Management	• No content changes
2021-11-25	R21-11	AUTOSAR Release Management	• No content changes
2020-11-30	R20-11	AUTOSAR Release Management	• No content changes
2019-11-28	R19-11	AUTOSAR Release Management	• No content changes • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Editorial changes
2017-12-08	4.3.1	AUTOSAR Release Management	• Editorial changes
2016-11-30	4.3.0	AUTOSAR Release Management	• Added requirements tracing section

▽

△

2015-07-31	4.2.2	AUTOSAR Release Management	• Editorial changes
2014-10-31	4.2.1	AUTOSAR Release Management	• Editorial changes
2013-10-31	4.1.2	AUTOSAR Release Management	• Editorial changes
2013-03-15	4.1.1	AUTOSAR Administration	• Added requirements to support ECU degradation concept • Added links to new features document
2010-09-30	3.1.5	AUTOSAR Administration	• Legal disclaimer revised
2008-08-13	3.1.1	AUTOSAR Administration	• Legal disclaimer revised
2007-12-21	3.0.1	AUTOSAR Administration	• Document meta information extended • Small layout adaptations made
2007-01-24	2.1.15	AUTOSAR Administration	• Modified "[SRS_Pwm_12379] Frequency of PWM channel groups" to support independent PWM frequencies • Synchronized duty cycle description with SW • Legal disclaimer revised • "Advice for users" revised • "Revision Information" added
2006-05-16	2.0	AUTOSAR Administration	• Release as a separate document. The SRS SPAL V1.0.0 has been split into 12 independent documents for Release 2.0. • Modification of configuration of the module
2005-05-31	1.0	AUTOSAR Administration	• Initial release as a part of the SRS SPAL V1.0.0

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Contents

1 Scope of Document

This document specifies requirements on the module PWM Driver.

Constraints

First scope for specification of requirements on basic software module is systems which are not safety relevant. For this reason safety requirements are assigned to medium priority.

2 Conventions to be used

2.1 Document Conventions

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability ([1]).

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([1]).

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as follows.

Note that the requirement level of the document in which they are used modifies the force of these words.

- MUST: This word, or the adjective "LEGALLY REQUIRED", means that the definition is an absolute requirement of the specification due to legal issues.
- MUST NOT: This phrase, or the phrase "MUST NOT", means that the definition is an absolute prohibition of the specification due to legal issues.
- SHALL: This phrase, or the adjective "REQUIRED", means that the definition is an absolute requirement of the specification.
- SHALL NOT: This phrase means that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED", means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY: This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item.

An implementation, which does not include a particular option, SHALL be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, SHALL be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

2.2 Requirements structure

Each module specific chapter contains a short functional description of the Basic Software Module. Requirements of the same kind within each chapter are grouped under the following headlines (where applicable):

Functional Requirements:

- Configuration (which elements of the module need to be configurable)
- Initialisation
- Normal Operation
- Shutdown Operation
- Fault Operation
- ...

Non-Functional Requirements:

- Timing Requirements
- Resource Usage
- Usability
- Output for other WPs (e.g. Description Templates, Tooling,...)
- ...

3 Acronyms and abbreviations

The glossary below includes acronyms and abbreviations relevant to SRS PWM Driver that are not included in the AUTOSAR Glossary [2].

Acronym:	**Description:**
CS	Chip Select
DIO	Digital Input Output
ECU	Electric Control Unit
DMA	Direct Memory Access
ICU	Input Capture Unit
MAL	Old name of Microcontroller Abstraction Layer (replaced by MCAL because 'MAL' is a french term meaning 'bad')
MCAL	Microcontroller Abstraction Layer
MCU	Microcontroller Unit
MISO	Master Input Slave Output
MMU	Memory Management Unit
MOSI	Master Output Slave Input
Master	A device controlling other devices (slaves, see below)
Slave	A device being completely controlled by a master device
NMI	Non Maskable Interrupt
OS	Operating System
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
RX	Reception (in the context of bus communication)
SPAL	The name of this working group
SFR	Special Function Register
RTE	RunTime Environment

Abbreviation:	***Description:***
STD	Standard
REQ	Requirement
UNINIT	Uninitialized (= not initialized)

As this is a document from professionals for professionals, all other terms are expected to be known.

4 Requirements Specification

This chapter describes all requirements driving the work to define the PWM Driver.

4.1 Functional Overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module Pwm driver.

Each PWM channel is linked to a hardware PWM which belongs to the microcontroller. The type of the PWM signal (for example center Align, left Align, Etc..) is not defined within this specification and is left up to the implementation.

The driver provides services for initialization and control of the microcontroller internal PWM stage (pulse width modulation). The PWM module generates pulses with variable pulse width. It allows the selection of the duty cycle and the signal period time.

4.2 Functional Requirements

4.2.1 General

[SRS_Pwm_12459] The PWM Driver shall provide a scaling scheme for duty cycle

Upstream requirements: RS_BRF_01992

⌈

Description:	The PWM Driver shall provide the following scaling scheme for duty cycle: • 0 = 0% • 0x8000 = 100% 0x8000 gives the highest resolution while allowing 100% duty cycle to be represented with a 16 bit value.
Rationale:	The value 0x8000 (32768) is chosen because the following calculation can be implemented efficiently.
Use Case:	Source code example: `AbsoluteDutyCycle = ((uint32)AbsolutePeriodTime * RelativeDutyCycle) >> 15;`
Dependencies:	[SRS_Pwm_12383] Resolution of duty cycle
Supporting Material:	–

⌋

[SRS_Pwm_12383] The PWM driver shall provide a 16 bit interface to set the duty cycle

Upstream requirements: RS_BRF_01992

⌈

Description:	The PWM driver shall provide a 16 bit interface to set the duty cycle.
Rationale:	–
Use Case:	–
Dependencies:	The duty cycle is always the active level of the signal (High or low, depends on idle level configuration, see [SRS_Pwm_12293])
Supporting Material:	–

⌋

4.2.2 Configuration

[SRS_Pwm_12375] The PWM Driver shall allow the module wide configuration of the numbers of PWM channels

Upstream requirements: RS_BRF_01856

⌈

Description:	The PWM Driver shall allow the module wide configuration of the following parameters: • Number of PWM channels
Rationale:	Basic configuration
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Pwm_12293] The PWM driver shall allow the static configuration of PWM channel properties

Upstream requirements: RS_BRF_01856

⌈

Description:	The PWM driver shall allow the static configuration of the following options for each PWM channel. Mandatory parameters: • assigned HW channel • default value for period • default value for duty cycle • Polarity (high or low) • idle state (duty cycle = 0%) high or low • Type of Channel: – Fixed Period – Fixed Period, shifted (if supported by hardware) – Variable Period Optional parameters (if supported by hardware): • channel phase shift • Reference channel for phase shift • microcontroller specific channel properties
Rationale:	Basic channel configuration.
Use Case:	Channel phase shift: to avoid EMC problems
Dependencies:	[SRS_Pwm_12383] [SRS_Pwm_12459]
Supporting Material:	–

⌋

[SRS_Pwm_12378] The PWM driver shall be able to assign notification to each edges of the PWM-signal

Upstream requirements: RS_BRF_02200

⌈

Description:	The PWM driver shall be able to assign notification to each edges of the PWM-signal. This notification shall be statically configurable.
Rationale:	–
Use Case:	• PWM edge triggered ADC conversions • PWM signal diagnosis
Dependencies:	[SRS_Pwm_12299] [SRS_Pwm_12293]

▽

△

Supporting Material:	–

⌋

[SRS_Pwm_12379] All PWM Channels which work with the same MCU Timer shall have either the same frequency or independent frequencies

Upstream requirements: RS_BRF_01008

⌈

Description:	All PWM Channels which work with the same MCU Timer shall have either the same frequency or independent frequencies.
Rationale:	Depending on the microcontroller hardware it can be possible that not every PWM channel has it's own timer. In this case the PWM channels have to share one timer. The frequencies of the PWM channels have to be either the same or independent from the HW timer frequency. The configuration has to take this into account.
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Pwm_12389] The PWM driver shall allow only static configuration of the frequency for some PWM channels

Upstream requirements: RS_BRF_01856

⌈

Description:	The PWM driver shall allow only static configuration of the frequency for some PWM channels.
Rationale:	–
Use Case:	The frequency of some PWM channels shall not be changeable during runtime
Dependencies:	[SRS_Pwm_12293] configuration of the Type of channel.
Supporting Material:	–

⌋

4.2.3 Initialization

[SRS_Pwm_12380] By initializing the PWM driver, all PWM-channels shall be started

Upstream requirements: RS_BRF_01136, RS_BRF_01096

⌈

Description:	By initializing the PWM driver, all PWM-channels are started.
Rationale:	–
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Pwm_12381] By de-initializing the PWM driver, all PWM-channels shall be stop

Upstream requirements: RS_BRF_01096

⌈

Description:	By de-initializing the PWM driver, all PWM-channels shall be stopped.The state of the PWM outputs shall be configurable.
Rationale:	–
Use Case:	–
Dependencies:	[SRS_Pwm_12293] Configuration of PWM channel properties
Supporting Material:	–

⌋

4.2.4 Normal Operation

[SRS_Pwm_12295] The PWM driver shall provide a service for setting the duty cycle of a selected channel

Upstream requirements: RS_BRF_01136

⌈

Description:	The PWM driver shall provide a service for setting the duty cycle of a selected channel. The parameters shall be • PWM channel • PWM duty cycle (range: 0..100%; 0% = Inverted polarity level, 100% = Polarity level, no spikes allowed)
Rationale:	Basic functionality.
Use Case:	–
Dependencies:	[SRS_Pwm_12459] [SRS_Pwm_12383]
Supporting Material:	–

⌋

[SRS_Pwm_12382] The PWM Driver shall wait to the end of the signal period to update the duty cycle of a PWM signal

Upstream requirements: RS_BRF_01136

⌈

Description:	The PWM Driver shall wait to the end of the signal period to update the duty cycle of a PWM signal. This feature shall be configurable.
Rationale:	Duty cycle not validduring a period. Duty cycle changes during a period can lead to undesired artefacts on the output signal (insertion of one pulse that is too short/too long). I know this as "buffered/unbuffered PWM output operation" (Freescale HC08). Buffered operation (which avoids those dirt effects) requires 2 registers instead of one.
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Pwm_12358] The PWM driver shall be capable to set the output of selected channel to a given state immediately

Upstream requirements: RS_BRF_01448

⌈

Description:	The PWM driver shall be capable to set the output of selected channel to a given state immediately.
Rationale:	Allow disabling a PWM output.
Use Case:	Brake slope of an MOSFET controlled DC motor: The duty cycle is reduced step wise within a slope. After having reached the target brake duty cycle value (e.g. 35 %), the MOSFET is switched off by setting the duty cycle immediately to 0%.
Dependencies:	–
Supporting Material:	This shall be a separate interface

⌋

[SRS_Pwm_12385] The PWM driver shall provide a service to get the state of a PWM channel output

Upstream requirements: RS_BRF_01448

⌈

Description:	The PWM driver shall provide a service to get the state of a PWM channel (high or low) if supported by hardware.
Rationale:	Allow to read back the state of the PWM for diagnosis.
Use Case:	Diagnostic of low frequency PWM outputs. Edge detection using polling when modulating Explanation of Hella on 17.02.2005: A power stage is connected to a PWM output pin. The diagnosis output of the power stage is connected to an ADC input pin. The evaluation of the diagnosis output shall be started after the requested level appears on the PWM output pin.
Dependencies:	–
Supporting Material:	–

⌋

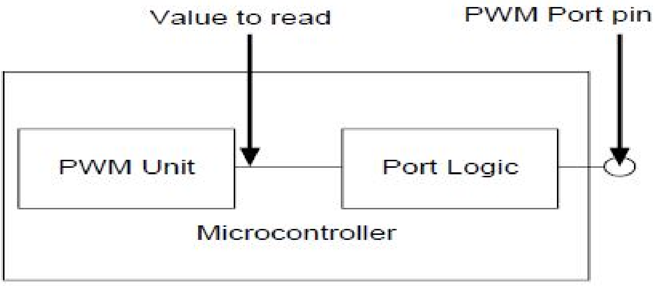

Figure 4.1: Image of the PWM channel for diagnosis

[SRS_Pwm_12297] The PWM driver shall provide a service for setting the period of a selected channel

Upstream requirements: RS_BRF_01992

⌈

Description:	The PWM driver shall provide a service for setting the period of a selected channel. The parameters shall be • PWM channel • PWM period • PWM duty cycle This functionality is available only for the PWM channel configured as type of channels "Variable Period".
Rationale:	The PWM duty cycle parameter is necessary to maintain the consistency between frequency and duty cycle. Otherwise, the effective duty cycle would change when the period is validor the PWM driver would have to recalculate the valid duty cycle by itself.
Use Case:	• Kojak siren with stable 50% duty cycle square wave • LED blinking with different frequencies
Dependencies:	[SRS_Pwm_12459] PWM duty cycle scaling
Supporting Material:	–

⌋

[SRS_Pwm_12299] The PWM driver shall allow to enable/disable the PWM edges notification during runtime

Upstream requirements: RS_BRF_01968

⌈

Description:	The PWM driver shall allow to enable/disable the PWM edges notification during runtime.
Rationale:	Allow synchronization with other modules
Use Case:	• PWM edge triggered ADC conversions • PWM signal diagnosis • Update of duty cycle using the notification in case of single buffer
Dependencies:	[SRS_Pwm_12293] Configuration of PWM channel properties
Supporting Material:	–

⌋

[SRS_Pwm_12460] An API shall be able to read the current power state of the PWM HW module

Upstream requirements: RS_BRF_01448, RS_BRF_01184

⌈

Description:	The PWM driver shall implement an API which allows to read the current power state of the HW perifpheral from outside of the MCAL module. This API can be used by IoHwAbs and CDD SW Components.
Rationale:	It must be possible to gather information on the current power state of a peripheral; this is useful to assert the actual state of the peripheral e.g. in order to know what operations are supported by the peripheral in a given moment or to decide in which power state it should be set.
Use Case:	–
Dependencies:	–
Supporting Material:	Concept n. 596 ECU Degradation

⌋

[SRS_Pwm_12461] An API shall be able to read the target power state of the <PWM/ADC> HW module

Upstream requirements: RS_BRF_01448, RS_BRF_01184

⌈

Description:	The PWM driver shall implement an API which allows to read the target power state of the HW perifpheral from outside of the MCAL module. This API can be used by IoHwAbs and CDD SW Components.
Rationale:	It is necessary to get information on the target power state in order to understand if a power state transition is being executed and, in positive case, which one.
Use Case:	–
Dependencies:	–
Supporting Material:	Concept n. 596 ECU Degradation

⌋

[SRS_Pwm_12462] The PWM Driver shall separate the power state setting procedure in to parts

Upstream requirements: RS_BRF_01448, RS_BRF_01184

⌈

Description:	The PWM Driver shall separate the power state transition sequence in two parts, a preparation phase (preliminary configuration changes to allow the peripheral to enter the target power state) and a setting phase (effective enabling of the valid power state).
Rationale:	Some peripherals may take more time than others to execute all preliminary transitions needed to get into a given power state. Moreover there could be some HW-related dependency between module, requiring to set a different peripheral in a specific power state or HW configuration before being able to transition the target peripheral. By splitting the process in a preparation and a setting phase, it is possible to synchronize different peripherals which than enter their valid power state all at the same time and it is also possible to coordinate HW state intermediate transitions.
Use Case:	–
Dependencies:	–
Supporting Material:	Concept n. 596 ECU Degradation

⌋

[SRS_Pwm_12463] It shall be possible to select a synchronous or asynchronous power state transition process

Upstream requirements: RS_BRF_01448, RS_BRF_01184

⌈

Description:	It shall be possible to configure a synchnous or an asynchronous power state transition behavior. In case a synchronous behavior is configured, the power transition preparation shall happen atomically and the SWC executing it must wait on the result. In case an asynchronous behavior is configured, the power transition preparation shall progress in background once requested and the MCAL module shall notify the registered SWC upon completion.
Rationale:	Some peripheral can be prepared to the valid power state in a negligible or acceptable time, which allows to spare the infrastructure needed to handle notifications, while blocking the caller for the time needed by the preparation. Some other peripheral could require some lengthier time to be prepared for the target power state, in this case the SWCs which expect a notification can instruct different modules to prepare for power state and go on with its execution, until all notifications are issued by the involved MCAL module.
Use Case:	–
Dependencies:	–
Supporting Material:	Concept n. 596 ECU Degradation

⌋

4.3 Non-Functional Requirements (Qualities)

[SRS_Pwm_12386] The PWM driver shall not cover a PWM emulation on general purpose I/O ⌈

Description:	The PWM driver shall only work with the PWM pins on the MCU. The PWM driver shall not cover a PWM emulation on general purpose I/O.
Rationale:	–
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

5 Requirements Tracing

The following table references the features specified in [3] and links to the fulfillments of these.

Requirement	Description	Satisfied by
[RS_BRF_01008]	AUTOSAR shall organize the hardware dependent layer in a microcontroller independent and a microcontroller dependent layer	[SRS_Pwm_12379]
[RS_BRF_01096]	AUTOSAR shall support start-up and shutdown of ECUs	[SRS_Pwm_12380] [SRS_Pwm_12381]
[RS_BRF_01136]	AUTOSAR shall support variants of configured BSW data resolved after system start-up	[SRS_Pwm_12295] [SRS_Pwm_12380] [SRS_Pwm_12382]
[RS_BRF_01184]	AUTOSAR shall support different methods of degradation	[SRS_Pwm_12460] [SRS_Pwm_12461] [SRS_Pwm_12462] [SRS_Pwm_12463]
[RS_BRF_01448]	AUTOSAR services shall support mode and state management	[SRS_Pwm_12358] [SRS_Pwm_12385] [SRS_Pwm_12460] [SRS_Pwm_12461] [SRS_Pwm_12462] [SRS_Pwm_12463]
[RS_BRF_01856]	AUTOSAR microcontroller abstraction shall provide access to internal MCU configuration	[SRS_Pwm_12293] [SRS_Pwm_12375] [SRS_Pwm_12389]
[RS_BRF_01968]	AUTOSAR IO Hardware Abstraction shall support edge triggered I/O signals	[SRS_Pwm_12299]
[RS_BRF_01992]	AUTOSAR IO Hardware Abstraction shall support frequency domain I/O signals	[SRS_Pwm_12297] [SRS_Pwm_12383] [SRS_Pwm_12459]
[RS_BRF_02200]	AUTOSAR diagnostic shall provide external access to internal configuration and calibration data	[SRS_Pwm_12378]

Table 5.1: Requirements Tracing

6 References

[1] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[2] Glossary
AUTOSAR_FO_TR_Glossary

[3] Requirements on AUTOSAR Features
AUTOSAR_CP_RS_Features