Document Title	Requirements on GPT Driver
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	187

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R23-11

Document Change History

Date	Release	Changed by	Description
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	• Editorial changes
2015-07-31	4.2.2	AUTOSAR Release Management	• Editorial changes

△

2014-10-31	4.2.1	AUTOSAR Release Management	• New RS feature linked to GPT Predef Timer requirementss
2013-10-31	4.1.2	AUTOSAR Release Management	• Editorial changes
2013-03-15	4.1.1	AUTOSAR Release Management	• Requirements for GPT Predef Timer functionality added
2011-12-22	4.0.3	AUTOSAR Release Management	• Requirements tracing reworked
2010-09-30	3.1.5	AUTOSAR Release Management	• BSW12460 removed • Legal disclaimer revised
2008-08-13	3.1.1	AUTOSAR Release Management	• Legal disclaimer revised
2007-12-21	3.0.1	AUTOSAR Release Management	• Document meta information extended • Small layout adaptations made
2007-01-24	2.1.15	AUTOSAR Release Management	• Legal disclaimer revised • Advice for users revised • Revision information revised
2006-05-16	2.0	AUTOSAR Release Management	• Release as a separate document. The SRS SPAL V1.0.0 has been split into 12 independent documents for Release 2.0 • Requirements added: [SRS_Gpt_13601], [SRS_Gpt_13602], [SRS_Gpt_13603]
2005-05-31	1.0	AUTOSAR Release Management	• Initial release as a part of the SRS SPAL V1.0.0

Disclaimer

Contents

1 Scope of Document

This document specifies requirements on the module GPT Driver.

Constraints

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

2 How to read this document

Each requirement has its unique identifier starting with the prefix "BSW" (for "Basic Software"). For any review annotations, remarks or questions, please refer to this unique ID rather than chapter or page numbers!

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 market-

place 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)
- Initialization
- 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 GPT driver that are not included in the AUTOSAR Glossary [2].

Acronym:	Description:
CS	Chip select
DIO	Digital Input Output
ECU	Electric Control Unit
EOL	End Of Line Often used in the term 'EOL Programming' or 'EOL Configuration'
ICU	Input Capture Unit
MAL	Old name of Microconroller Abstraction Layer (replaced by MCAL because 'MAL' is a french term meaning 'bad')
MCAL	Microconroller Abstraction Layer
MCU	Microcontroller Unit
MMU	Memory Management Unit
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 (Standard Peripheral Abstraction Layer)
SFR	Special Function Register
RTE	Runtime environment
WP	Work Package

Table 3.1: Acronyms used in the scope of this Document

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

Table 3.2: Abbreviations used in the scope of this Document

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 GPT driver specification.

4.1 Functional Overview

The GPT driver is part of the microcontroller abstraction layer (MCAL). It initializes and controls the internal General Purpose Timer(s) (GPT) of the microcontroller.

The GPT driver provides services and configuration parameters for

- Starting and stopping hardware timers
- Getting timer values
- Controlling time triggered interrupt notifications
- Controlling time triggered wakeup interrupts

The GPT driver is able to provide exact and short-term timings. The one-shot or continuous interrupt notifications of the GPT driver can be used where the OS Alarm service has too much overhead.

An example of a typical period time range is 50μs ... 5 ms.

Some free running up counters - so-called GPT Predef Timers - are defined. These timers have predefined tick durations and predefined number of bits (physical time units and ranges). The GPT Predef Timers are used by the Time Service module.

4.2 Functional Requirements

4.2.1 General

4.2.1.1 [SRS_Gpt_12328]

[SRS_Gpt_12328] The GPT driver shall use the time unit ticks for all API services which are related to GPT timer channels ⌈

Description:	The GPT driver shall use the time unit ticks for all API services which are related to GPT timer channels.
Rationale:	Conversions between physical time unit and ticks shall be part of the user software.
Use Case:	–
Dependencies:	[SRS_BSW_00343] Specification and configuration of time

▽

△

Supporting Material:	–

⌋(*RS_Main_00130*, *RS_Main_00435*)

4.2.1.2 [SRS_Gpt_13604]

[SRS_Gpt_13604] The GPT driver shall support special free running up counters, so-called GPT Predef Timers ⌈

Description:	The GPT driver shall support free running up counters (GPT Predef Timers) with predefined tick durations and predefined number of bits (physical time units and ranges). The functionality of the GPT Predef Timers shall be separated from the functionality relating to GPT timer channels.
Rationale:	The GPT driver shall provide the hardware time bases for the Time Service module.
Use Case:	Time measurement, time based state machine, timeout supervision, busy waiting
Dependencies:	[SRS_BSW_00343] Specification and configuration of time
Supporting Material:	–

⌋(*RS_Main_00100*, *RS_Main_00130*, *RS_Main_00435*)

4.2.1.3 [SRS_Gpt_13605]

[SRS_Gpt_13605] Different types of GPT Predef Timers shall be supported by the GPT driver ⌈

Description:	The following types of GPT Predef Timers shall be supported by the GPT driver: • Timer 1μs 16bit • Timer 1μs 24bit • Timer 1μs 32bit • Timer 100μs 32bit
Rationale:	1μs: high resolution timer. 16bit timer: To support 16bit hardware timers. 24bit timer: To support 24bit hardware timers. 32bit timer: To support 32bit hardware timers. 100μs32bit timer: covers automotive use cases (time span 4.9 days)

▽

△

Use Case:	Time measurement, time based state machine, timeout supervision, busy waiting
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00100*, *RS_Main_00130*, *RS_Main_00435*)

4.2.2 Configuration

4.2.2.1 [SRS_Gpt_12404]

[SRS_Gpt_12404] Configuration of one-shot/continuous mode for each timer channel shall be available ⌈

Description:	The GPT Driver shall allow the following static configuration for each timer channel: • One-Shot mode: After the timer has reached its end value, the timer is stopped • Continuous mode: After the timer has reached it's end value, the timer is restarted automatically
Rationale:	Provision of guaranteed minimum delay time or guaranteed frequency.
Use Case:	One-shot mode: Stepper motor control, where coil driver pulses must have a defined minimum duration. The timer is restarted after the output signal is set. Even if one output pulse is delayed (e.g. by interrupt disabling), the next pulse does not occur too early. Continuous mode: ADC conversion triggering. The ADC is triggered continuously at a fixed rate without the need of restarting the timer. Input signal sampling. An input signal is sampled at a fixed rate.
Dependencies:	–
Supporting Material:	BMW Specification MCAL V1.0a, REQ MAL30.1.5

⌋(*RS_Main_00130*, *RS_Main_00435*)

4.2.2.2 [SRS_Gpt_12114]

[SRS_Gpt_12114] Each timer channel shall be configured in a way that the timer can use different clock sources ⌈

Description:	The GPT driver shall make it possible to configure statically each timer channel in a way that the timer can use different clock sources if provided by hardware.
Rationale:	To provide general purpose functionality
Use Case:	The clock source is different in normal and power save mode.
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00100*, *RS_Main_00130*, *RS_Main_00435*)

4.2.2.3 [SRS_Gpt_13606]

[SRS_Gpt_13606] The GPT driver shall make it possible to configure statically which GPT Predef Timers are enabled ⌈

Description:	The GPT driver shall make it possible to configure statically which GPT Predef Timers are enabled.
Rationale:	To disable GPT Predef Timers if timers can not be supported by hardware reasons.
Use Case:	Hardware does not support a GPT Predef Timer
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00100*, *RS_Main_00130*, *RS_Main_00435*)

4.2.3 Initialization

4.2.3.1 [SRS_Gpt_12116]

[SRS_Gpt_12116] The GPT Driver shall provide the functionality to deinitialize timer channels to their power on reset state ⌈

Description:	The GPT Driver shall provide the functionality to deinitialize timer channels to their power on reset state.
Rationale:	It is necessary to reset all hardware registers to the same state before a valid initialization can be done. Otherwise the code for the initialization is different for initialization after power on reset or after a mode change.

▽

△

Use Case:	After changing internal clock frequency for power save modes it might be necessary to initialize the timer module with valid prescaler values.
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00100*, *RS_Main_00130*, *RS_Main_00435*)

4.2.4 Normal Operation

4.2.4.1 [SRS_Gpt_12117]

[SRS_Gpt_12117] The GPT Driver shall provide a synchronous service for reading the current timer value of each timer channel ⌈

Description:	The GPT Driver shall provide a synchronous service for reading the current timer value of each timer channel.
Rationale:	–
Use Case:	Some signals need a time stamp.
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00100*, *RS_Main_00130*, *RS_Main_00435*)

4.2.4.2 [SRS_Gpt_12128]

[SRS_Gpt_12128] The GPT driver shall provide a service for starting a timer with specific parameters ⌈

Description:	The GPT driver shall provide a service for starting a timer with the following parameters: • timer channel • time period (number of ticks after the notification shall occur)
Rationale:	Basic functionality.
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00100*, *RS_Main_00130*, *RS_Main_00435*)

4.2.4.3 [SRS_Gpt_12119]

[SRS_Gpt_12119] The GPT driver shall provide the service for stopping each channel of the timer ⌈

Description:	The GPT driver shall provide the service for stopping each channel of the timer.
Rationale:	Without control the timer runs as long as power is supplied.
Use Case:	The timer has to be stopped before validinitialization or change of its value to avoid unwanted activities bound to timer values.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00100, RS_Main_00130, RS_Main_00435)*

4.2.4.4 [SRS_Gpt_12120]

[SRS_Gpt_12120] The GPT Driver shall provide a notification per channel that is called when the time period has elapsed ⌈

Description:	The GPT Driver shall provide a notification per channel that is called when the time period has elapsed. This callback shall be statically configurable per channel.
Rationale:	A timer is normally connected
Use Case:	1. A functionality needs the information that a certain amount of time has passed. 2. To synchronize another action from a user function
Dependencies:	[SRS_Gpt_12128] Start timer
Supporting Material:	–

⌋*(RS_Main_00130, RS_Main_00400, RS_Main_00435)*

4.2.4.5 [SRS_Gpt_12121]

[SRS_Gpt_12121] The GPT Driver shall provide the functionality to enable the call of a notification function per channel during the runtime ⌈

Description:	The GPT Driver shall provide the functionality to enable the call of a notification function per channel during the runtime.
Rationale:	A notification function has to be declared explicitly.

▽

△

Use Case:	When the timer rolls over. Roll over means that the timer reaches its maximum value and starts from zero or that it reaches a predefined value and starts from zero.
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00100*, *RS_Main_00130*, *RS_Main_00400*, *RS_Main_00435*)

4.2.4.6 [SRS_Gpt_12122]

[SRS_Gpt_12122] The GPT Driver shall provide the functionality to disable the call of a notification function per channel during the runtime ⌈

Description:	The GPT Driver shall provide the functionality to disable the call of a notification function per channel during the runtime.
Rationale:	Without disabling the notification would be active as long as the timer is active.
Use Case:	When the timer rolls over. (see enable notification)
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00100*, *RS_Main_00130*, *RS_Main_00400*, *RS_Main_00435*)

4.2.4.7 [SRS_Gpt_13601]

[SRS_Gpt_13601] The GPT Driver shall be capable of performing wakeup events, whenever a predefined wakeup period has expired ⌈

Description:	The GPT Driver shall be capable of performing wakeup events, whenever a predefined wakeup period has expired. This feature shall only be available, if supported by hardware
Rationale:	Reducing power consumption
Use Case:	Flashing LED. The ECU is put in sleep mode in the time between the flashes and woken up, when the LED should be turned on again.
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00130*, *RS_Main_00435*, *RS_Main_00460*)

4.2.4.8 [SRS_Gpt_13602]

[SRS_Gpt_13602] The GPT driver shall provide a service for enabling / disabling the wake-up capability of single timer channels ⌈

Description:	The GPT driver shall provide a service for enabling / disabling the wake-up capability of single timer channels. Related notifications for this channel shall be enabled / disabled.
Rationale:	Controlling the wake-up conditions of a MCU needs to enable or disable the notifications.
Use Case:	–
Dependencies:	[SRS_Gpt_13601] Wakeup functionality
Supporting Material:	–

⌋*(RS_Main_00100, RS_Main_00130, RS_Main_00400, RS_Main_00435)*

4.2.4.9 [SRS_Gpt_13603]

[SRS_Gpt_13603] The GPT driver shall provide a service for selecting the Wake-up mode ⌈

Description:	The GPT driver shall provide a service for selecting the Wake-up mode: • Normal mode (mandatory) • Wake-up mode In normal mode all notifications are available as configured. In Wake-up mode only those notifications, which cause wake-up capable notifications, are available. All other notifications are disabled and must not lead to an exit of the reduced power mode state (e.g. idle, halt) of the MCU if the event occurs.
Rationale:	Allow enabling / disabling of all notifications which are not required for the ECU wake-up.
Use Case:	During entry in the reduced power mode of an ECU all notifications of the MCU shall be disabled without disabling the wake-up sources in between. Otherwise wake-up events can be lost.
Dependencies:	[SRS_Gpt_13601] Wakeup functionality
Supporting Material:	–

⌋*(RS_Main_00100, RS_Main_00130, RS_Main_00400, RS_Main_00435, RS_Main_-00460)*

4.2.4.10 [SRS_Gpt_13607]

[SRS_Gpt_13607] The GPT Predef Timers shall be started/stopped automatically by the GPT driver ⌈

Description:	The GPT Predef Timers shall be started/stopped automatically by the GPT driver.
Rationale:	To ensure that all enabled GPT Predef Timers run whenever possible (after initialization/deinitialization, after entering normal/sleep mode).
Use Case:	Avoiding start of GPT Predef Timers by upper layer module
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00100*, *RS_Main_00130*, *RS_Main_00435*)

4.2.4.11 [SRS_Gpt_13608]

[SRS_Gpt_13608] The GPT driver shall provide a synchronous service for reading the current timer value of each GPT Predef Timer ⌈

Description:	The GPT driver shall provide a synchronous service for reading the current timer value of each GPT Predef Timer.
Rationale:	To get the timer values.
Use Case:	Time measurement, time based state machine, timeout supervision, busy waiting
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00100*, *RS_Main_00130*, *RS_Main_00435*)

4.2.5 Fault Operation

None

4.3 Non-Functional Requirements (Qualities)

No content.

5 Requirements Tracing

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

Requirement	Description	Satisfied by
[RS_Main_00100]	AUTOSAR shall provide standardized Basic Software	[SRS_Gpt_12114] [SRS_Gpt_12116] [SRS_Gpt_12117] [SRS_Gpt_12119] [SRS_Gpt_12121] [SRS_Gpt_12122] [SRS_Gpt_12128] [SRS_Gpt_13602] [SRS_Gpt_13603] [SRS_Gpt_13604] [SRS_Gpt_13605] [SRS_Gpt_13606] [SRS_Gpt_13607] [SRS_Gpt_13608]
[RS_Main_00130]	Hardware Abstraction Layer	[SRS_Gpt_12114] [SRS_Gpt_12116] [SRS_Gpt_12117] [SRS_Gpt_12119] [SRS_Gpt_12120] [SRS_Gpt_12121] [SRS_Gpt_12122] [SRS_Gpt_12128] [SRS_Gpt_12328] [SRS_Gpt_12404] [SRS_Gpt_13601] [SRS_Gpt_13602] [SRS_Gpt_13603] [SRS_Gpt_13604] [SRS_Gpt_13605] [SRS_Gpt_13606] [SRS_Gpt_13607] [SRS_Gpt_13608]
[RS_Main_00400]	AUTOSAR shall provide a layered software architecture	[SRS_Gpt_12120] [SRS_Gpt_12121] [SRS_Gpt_12122] [SRS_Gpt_13602] [SRS_Gpt_13603]
[RS_Main_00435]	AUTOSAR shall support automotive microcontrollers	[SRS_Gpt_12114] [SRS_Gpt_12116] [SRS_Gpt_12117] [SRS_Gpt_12119] [SRS_Gpt_12120] [SRS_Gpt_12121] [SRS_Gpt_12122] [SRS_Gpt_12128] [SRS_Gpt_12328] [SRS_Gpt_12404] [SRS_Gpt_13601] [SRS_Gpt_13602] [SRS_Gpt_13603] [SRS_Gpt_13604] [SRS_Gpt_13605] [SRS_Gpt_13606] [SRS_Gpt_13607] [SRS_Gpt_13608]
[RS_Main_00460]	AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level	[SRS_Gpt_13601] [SRS_Gpt_13603]

Table 5.1: RequirementsTracing

6 References

[1] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[2] Glossary
AUTOSAR_FO_TR_Glossary

[3] Requirements on AUTOSAR Features
AUTOSAR_CP_RS_Features

A Change history of AUTOSAR traceable items

Please note that the lists in this chapter also include traceable items that have been removed from the specification in a later version. These items do not appear as hyperlinks in the document.

A.1 Traceable item history of this document according to AUTOSAR Release R23-11

A.1.1 Added Requirements in R23-11

none

A.1.2 Changed Requirements in R23-11

none

A.1.3 Deleted Requirements in R23-11

none