Document Title	Requirements on GPT Driver
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Version	2.0.1
Document Status	Final
Part of Release	2.1
Revision	0014

Document Change History

Date	Version	Changed by	Change Description
26.01.2007	2.0.1	AUTOSAR Administration	• Legal disclaimer revised • "Advice for users" revised • "Revision Information" added
21.03.2006	2.0.0	AUTOSAR Administration	Release as a separate document. The SRS SPAL V1.0.0 has been split into 12 independent documents for Release 2.0 Requierments added: • [BSW13601] Wakeup functionality • [BSW13602] Enable/Disable Wakeup • [BSW13603] Wake-up mode selection service
02.06.2005	1.0.0	AUTOSAR Administration	Initial release as a part of the SRS SPAL V1.0.0

Page left intentionally blank

Disclaimer

Advice to users of AUTOSAR Specification Documents:

AUTOSAR Specification Documents may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).
Any such exemplary items are contained in the Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

1 Scope of this 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 Conventions used

In requirements, the following specific semantics are used (taken from Request for Comment RFC 2119 from the Internet Engineering Task Force IETF)

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 described in RFC 2119. Note that the requirement level of the document in which they are used modifies the force of these words.

- MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- MUST NOT: This phrase, or the phrase „SHALL NOT“, means that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", mean 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" mean 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, MUST 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, MUST 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

Acronyms and abbreviations that have a local scope are not contained in the AUTOSAR glossary. These must appear in a local glossary.

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'
HIS	Herstellerinitiative Software
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

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 Requirement Specification

4.1 GPT Driver

4.1.1 Functional Overview

The GPT driver allows generating one-shot or continuous timer notifications. The module uses the hardware timer channels of the general-purpose timer and thus provides exact and short-term timings for use in the Operating System or within other basic software modules where an OS Alarm service has too much overhead.

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

4.1.2 Functional Requirements

4.1.2.1 General

4.1.2.1.1 [BSW12328] GPT driver time unit

Initiator:	WP4.2.2.1.12
Date:	28.09.2004
Short Description:	GPT driver time unit
Type:	Changed (during internal review)
Importance:	High
Description:	All time units used within the API services of the GPT driver shall be of the unit ticks.
Rationale:	Conversions between microseconds and ticks shall be part of the ECU Abstraction Layer.
Use Case:	--
Dependencies:	[[BSW00343] Specification and configuration of time
Conflicts:	--
Supporting Material:	--

4.1.2.2 Configuration

4.1.2.2.1 [BSW12404] Configuration of one-shot/continuous mode

Initiator:	BMW
Date:	05.07.2004
Short Description:	Configuration of one-shot/continuous mode
Type:	New
Importance:	High
Description:	The GPT Driver shall allow the following static configuration for each timer channel: • One-Shot mode: After the timer has reached it's 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:	--
Conflicts:	--
Supporting Material:	BMW Specification MCAL V1.0a, REQ MAL30.1.5

4.1.2.2.2 [BSW12114] Configuration of timer clock source

Initiator:	CAS & BMW
Date:	07.07.2004
Short Description:	Configuration of timer clock source
Type:	Changed (mention of statically configuration)
Importance:	High
Description:	The GPT driver shall allow 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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.2.3 [BSW12460] Configuration of symbolic names for time values

Initiator:	WP4.2.2.1.12
Date:	17.02.2005
Short Description:	Configuration of symbolic names for time values
Type:	New
Importance:	High
Description:	The GPT driver configuration tool shall allow to generate an arbitrary number of user definable symbolic names for time values. Configuration parameters per symbolic name: • Symbolic name itself • Timer channel • Time in µs The values behind the symbolic names shall be calculated from the configuration of the corresponding timer channel.
Rationale:	Abstraction from ticks
Use Case:	Eample: User input to configuration tool: GPT1_INPUT_SAMPLE_RATE, CH1, 224 GPT2_TIME_1_MS, CH2, 1000 GPT2_TIME_2_MS, CH2, 2000

	Configuration tool output:
	`#define GPT1_INPUT_SAMPLE_RATE 814`
	`#define GPT2_TIME_1_MS 94`
	`#define GPT2_TIME_2_MS 188`

Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.2.3 Initialization

4.1.2.3.1 [BSW12116] GPT Deinitialization

Initiator:	CAS
Date:	25.06.04
Short Description:	GPT Deinitialization
Type:	Changed (Addition of a channel reference)
Importance:	High
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 new 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 new prescaler values.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.2.4 Normal Operation

4.1.2.4.1 [BSW12117] Read timer value

Initiator:	CAS
Date:	25.06.04
Short Description:	Read timer value
Type:	New
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.4.2 [BSW12128] Start timer

Initiator:	CAS
Date:	28.09.2004
Short Description:	Start timer

Type:	Changed during Wp4.2.2.1.12 review
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.4.3 [BSW12119] Stop timer

Initiator:	CAS
Date:	25.06.04
Short Description:	Stop timer
Type:	New
Importance:	High
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 new initialization or change of its value to avoid unwanted activities bound to timer values.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.2.4.4 [BSW12120] Provide notification

Initiator:	CAS
Date:	07.07.04
Short Description:	Provide notification
Type:	Changed (Statically configurable only)
Importance:	High
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:	[BSW12128] Start timer
Conflicts:	--
Supporting Material:	--

4.1.2.4.5 [BSW12121] Enable notification

Initiator:	CAS
Date:	07.07.04
Short Description:	Enable notification
Type:	Changed (description and use case)

Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.4.6 [BSW12122] Disable notification

Initiator:	CAS
Date:	07.07.04
Short Description:	Disable notification
Type:	Changed (description and use case)
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.4.7 [BSW13601] Wakeup functionality

Initiator:	WP4.2.2.1.12
Date:	21.07.05
Short Description:	Wakeup functionality
Type:	Changed
Importance:	High
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:	--
Conflicts:	--
Supporting Material:	--

4.1.2.4.8 [BSW13602 Enable/Disable Wakeup

Initiator:	CAS
Date:	21.07.05
Short Description:	Enable/Disable Wakeup
Type:	Changed
Importance:	High
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:	[BSW13601] Wakeup functionality
Conflicts:	--
Supporting Material:	--

4.1.2.4.9 [BSW13603] Wake-up mode selection service

Initiator:	WP4.2.2.1.12
Date:	21.07.05
Short Description:	Wake-up mode selection service
Type:	New
Importance:	High
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:	[BSW13601] Wakeup functionality
Conflicts:	--
Supporting Material:	--

4.1.2.5 Fault Operation

None

5 References

5.1 Deliverables of AUTOSAR

[DOC_LAYERED_ARCH] Layered Software Architecture,
https://svn.autosar.org/repos/10Releases/
AUTOSAR_LayeredSoftwareArchitecture.pdf

[AUTOSAR_GLOSSARY] Glossary,
https://svn.autosar.org/repos/10Releases
AUTOSAR_Glossary.pdf

[SRS_BSW_GENERAL] General Requirements on Basic Software Modules,
https://svn.autosar.org/repos/10Releases
AUTOSAR_SRS_General.pdf

[SRS_BSW_SPAL] General Requirements on SPAL,
https://svn.autosar.org/repos/10Releases
AUTOSAR_SRS_SPAL_General.pdf

5.2 Related standards and norms

[STD_HIS_IO_DRIVER] HIS API IO Driver, V2.1.3, April 29th, 2004,
http://www.automotive-his.de/download
API_IODriver_2_1_3.pdf