

Document Title	Requirements on ICU Driver
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Identification No	112
Document Classification	Auxiliary

Document Version	2.0.4
Document Status	Final
Part of Release	3.1
Revision	0001

Document Change History			
Date	Version	Changed by	Change Description
23.06.2008	2.0.4	AUTOSAR Administration	Legal disclaimer revised
31.10.2007	2.0.3	AUTOSAR Administration	<ul style="list-style-type: none"> • Document meta information extended • Small layout adaptations made
24.01.2007	2.0.2	AUTOSAR Administration	<ul style="list-style-type: none"> • “Advice for users” revised • “Revision Information” added
28.11.2006	2.0.1	AUTOSAR Administration	Legal disclaimer revised
25.04.2006	2.0.0	AUTOSAR Administration	<p>Release as a separate document. The SRS SPAL V1.0.0 has been split into 12 independent documents for Release 2.0.</p> <ul style="list-style-type: none"> • Removed functionalities: <ul style="list-style-type: none"> - Hardware gated edge counting - Notification on signal level • Approved requirements for functionality: <ul style="list-style-type: none"> - Timestamping - Wakeup - Edge-Counting - Get elapsed Signal Times - Get Duty Cycle input values • Added requirements for functionality: <ul style="list-style-type: none"> - Reset the value of counted edges
11.07.2005	1.0.0	AUTOSAR Administration	Initial release as a part of the SRS SPAL V1.0.0

Page left intentionally blank

Disclaimer

This document of a specification as released by the AUTOSAR Development Partnership is intended **for the purpose of information only**. The commercial exploitation of material contained in this specification requires membership of the AUTOSAR Development Partnership or an agreement with the AUTOSAR Development Partnership. The AUTOSAR Development Partnership will not be liable for any use of this specification. Following the completion of the development of the AUTOSAR specifications commercial exploitation licenses will be made available to end users by way of written License Agreement only.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher." The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Copyright © 2004-2008 AUTOSAR Development Partnership. All rights reserved.

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.....	6
2	How to read this document.....	7
2.1	Conventions used.....	7
2.2	Requirements structure	8
3	Acronyms and abbreviations	9
4	Requirement Specification.....	10
4.1	ICU Driver	10
4.1.1	Functional Overview.....	10
4.1.2	Constrains to other functions.....	13
4.1.3	Functional Requirements	13
4.1.3.1	Configuration.....	13
4.1.3.1.1	[BSW12327] ICU global configuration	13
4.1.3.1.2	[BSW12368] ICU channel/group configuration	13
4.1.3.1.3	[BSW12425] Measured property of ICU Channel	14
4.1.3.2	Initialization	14
4.1.3.2.1	[BSW12407] Initialization of ICU.....	14
4.1.3.2.2	[BSW12429] ICU Deinitialization	14
4.1.3.3	Normal Operation.....	14
4.1.3.3.1	[BSW12305] Enable/Disable notification during runtime.....	14
4.1.3.3.2	[BSW12369] Notification on signal edge.....	16
4.1.3.3.3	[BSW12370] Sleep mode selection service	16
4.1.3.3.4	[BSW12371] ICU Channel status function	16
4.1.3.3.5	[BSW12438] Timestamps of elapsed time	16
4.1.3.3.6	[BSW12455] External circular buffer handling	16
4.1.3.3.7	[BSW12456] External linear buffer handling	18
4.1.3.3.8	[BSW12430] ICU start timestamp service.....	18
4.1.3.3.9	[BSW12431] ICU cancel timestamp service	18
4.1.3.3.10	[BSW12444] ICU timestamp notification.....	19
4.1.3.3.11	[BSW12453] ICU get timestamp index service	19
4.1.3.3.12	[BSW12439] Edge counting.....	19
4.1.3.3.13	[BSW12432] Enable ICU edge counting service.....	20
4.1.3.3.14	BSW13100 Reset the value of counted edges	20
4.1.3.3.15	[BSW12433] Disable ICU edge counting service.....	21
4.1.3.3.16	[BSW12434] ICU edge counting read service.....	21
4.1.3.3.17	[BSW12442] Get elapsed Signal Low Time for an ICU Channel	22
4.1.3.3.18	[BSW12435] Get elapsed Signal High Time for an ICU Channel	22
4.1.3.3.19	[BSW12443] Get elapsed Period Time for an ICU Channel....	22
4.1.3.3.20	[BSW12436] Get Duty Cycle input values for an ICU Channel	23
4.1.3.3.21	[BSW12437] ICU driver time unit.....	23
4.1.3.4	Shutdown Operation	23
4.1.3.4.1	[BSW12408] Wake-up enable / disable service	23

5	Related Documentation	25
5.1	Related standards and norms	25
5.1.1	HIS	25

1 Scope of this document

This document specifies requirements on the module ICU Driver.

Constraints

First scope for specification of requirements on basic software modules are 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
Circular buffer	An area of memory used to store a continuous stream of data by starting again at the beginning of the buffer after reaching the end.
DIO	Digital Input Output
Duty Cycle	Percentage of High Time to Period Time (High Time / Period Time) * 100%
ECU	Electric Control Unit
High Time	See Figure "ICU time definitions". The standard type STD_HIGH shall be used.
HIS	Herstellerinitiative Software
ICU	Input Capture Unit
ICU channel	Represents a logical ICU entity bound to one input signal and the hardware resources for the configured measurement mode.
Linear buffer	An area of memory used to store a stream of data by starting at the beginning of the buffer and stopping at the latest on reaching the end.
Low Time	See Figure "ICU time definitions"
MAL	Old name of Microcontroller Abstraction Layer (replaced by MCAL because 'MAL' is a French term meaning 'male')
MCAL	Microcontroller Abstraction Layer
MCU	Microcontroller Unit
Measurement mode	The measurement mode defines the capability for signal acquisition and evaluation. Possible modes: <ul style="list-style-type: none"> • Signal Edge Detection / Notification • Signal Measurement • Timestamp Edge Counter
Measurement mode, Edge counter	Functionality of an Edge Counter, counting of external edges
Measurement mode, Signal Edge Detection	Notification on signal edges.
Measurement mode, Signal Measurement	Measurement of elapsed High Time, elapsed Low Time, elapsed Period Time and Duty Cycle of an input signal.
Measurement mode, Timestamp	Generation of timestamps for signal edges, see Figure "ICU time stamp"
Period Time	See Figure "ICU time definitions"
PWD	Pulse width demodulation
SPAL	Standard Peripheral Abstraction Layer

Abbreviation	Description
STD	Standard
UNINIT	Uninitialized (= not initialized)

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

4 Requirement Specification

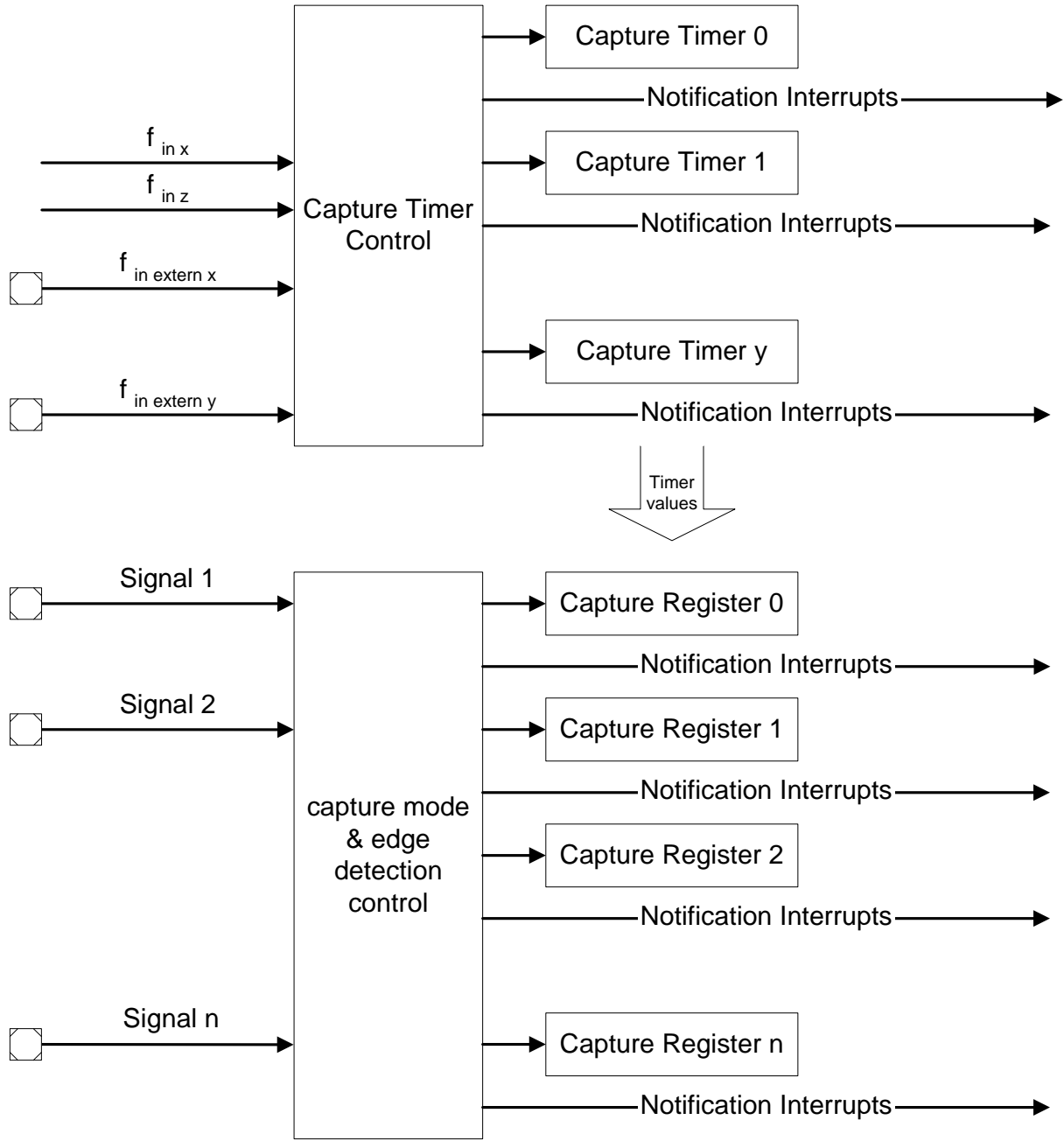
4.1 ICU Driver

4.1.1 Functional Overview

The ICU Driver controls the input capture unit of the microcontroller. It provides the following features:

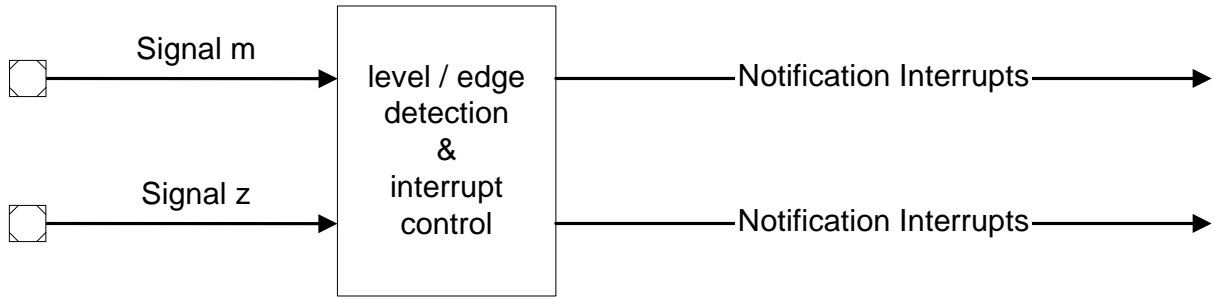
- Period-, Low-, High time measurement
- Edge detection and notification
- Edge counting
- Edge time stamping
- Wake-up interrupts

The picture below shows the typical key resources of a Capture Compare Unit:



For signal edge detection the edge detector of a capture compare unit or the interrupt controller for external events is used.

For signal measuring a capture timer and at least one capture register is needed. Simple signal edge detection without time measurement can also be implemented with external interrupt control units:



However Non Maskable Interrupts (NMI) are not in the scope of this module, because there is nothing which can be controlled.

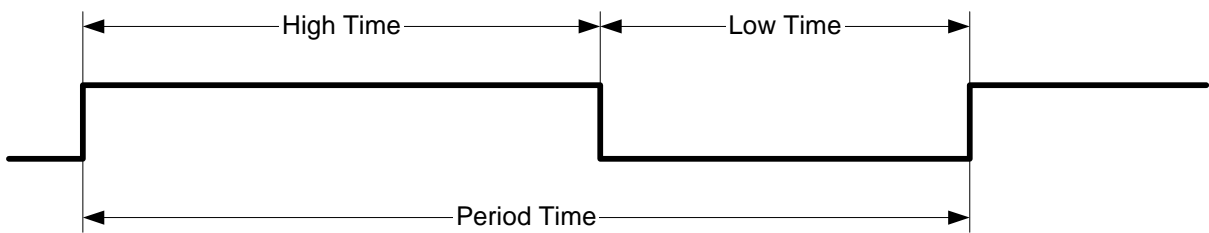


Figure ICU time definitions

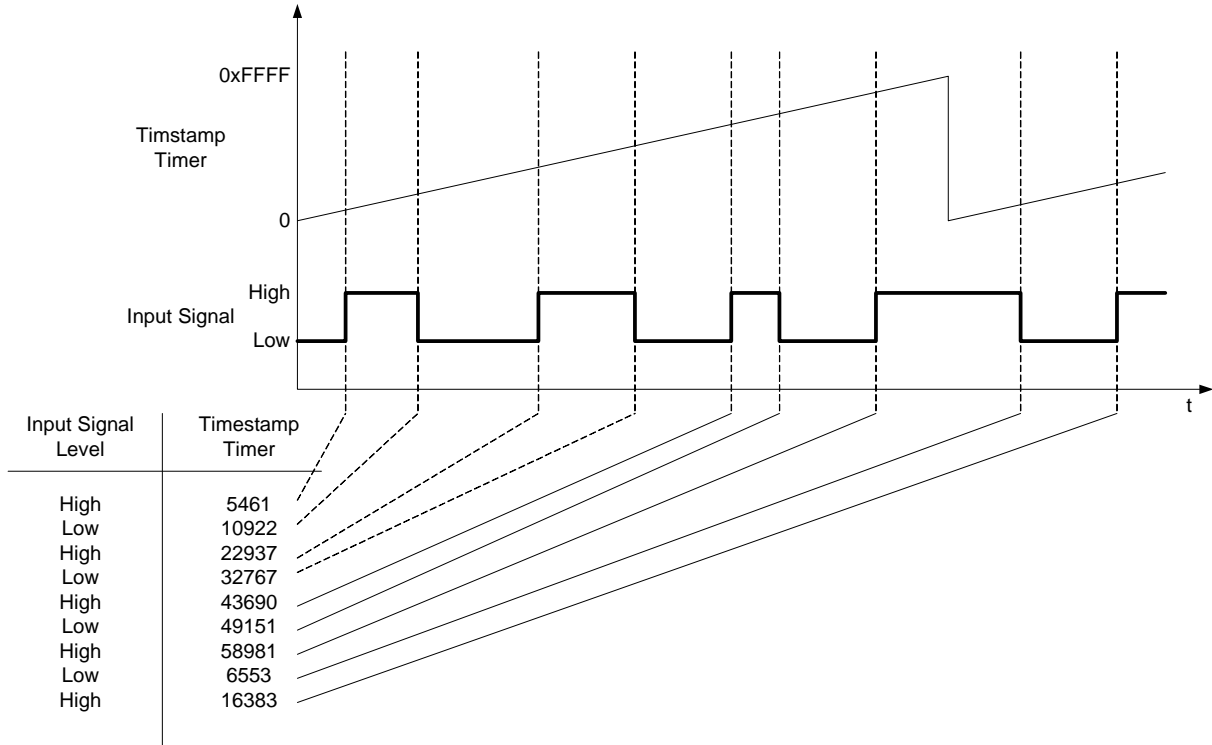


Figure 1: ICU time stamp

4.1.2 Constrains to other functions

Constrains to Mode Management:

After a wake-up condition, the ECU state manager shall take care about the wake-up information before it calls the initialization service of the ICU driver.

4.1.3 Functional Requirements

4.1.3.1 Configuration

4.1.3.1.1 [BSW12327] ICU global configuration

Initiator:	WP4.2.2.1.12
Date:	27.09.2004
Short Description:	ICU global configuration
Type:	New
Importance:	High
Description:	The ICU driver shall allow the configuration of the following parameters: <ul style="list-style-type: none"> • Clock source with optional prescaler (module wide) • MCU Hardware dependent settings (only ICU-peripheral specific settings)
Rationale:	Configuration of microcontroller specific ICU features
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.2 [BSW12368] ICU channel/group configuration

Initiator:	MEDAG
Date:	16.11.2004
Short Description:	ICU channel/group configuration
Type:	Changed
Importance:	High
Description:	The ICU driver shall support the following basic static configurations per channel: <p>Mandatory parameters</p> <ul style="list-style-type: none"> • Symbolic name for channel • Notification function(s) • Wake-up capability <p>Optional parameters</p> <ul style="list-style-type: none"> • Signal Source (Configured Port Pin or Input from a Signal Matrix) if provided by hardware (if multiple sources are selectable) • Measurement Mode: (if multiple modes are selectable) <ul style="list-style-type: none"> • Signal Edge Detection / Notification • Signal Measurement • Timestamp • Edge Counter • If measurement mode is "timestamp measurement", buffer handling

	<p>shall be configurable. The values shall be:</p> <ul style="list-style-type: none"> - Circular buffer handling - Linear buffer handling <ul style="list-style-type: none"> • Assigned capture register(s) (can also be none for channels which provide only edge detection like an external interrupt) • Assigned capture timer (can also be none for channels which provide only edge detection like an external interrupt) • Additional hardware dependent settings (e.g. glitch filters, prescalers)
Rationale:	To allow different usage for each channel
Use Case:	
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.1.3 [BSW12425] Measured property of ICU Channel

Initiator:	Valeo
Date:	02.12.2004
Short Description:	Measured property of ICU Channel
Type:	New
Importance:	High
Description:	<p>For each ICU Channel the 'property' that could be measured shall be configurable and the values available (at least) shall be :</p> <ul style="list-style-type: none"> • High Level • Low Level • Period Time
Rationale:	Definition of the measurement purpose to allocate the required HW resources during configuration.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.2 Initialization

4.1.3.2.1 [BSW12407] Initialization of ICU

Initiator:	WP4.2.2.1.12
Date:	25.11.2004
Short Description:	Initialization of ICU
Type:	Changed
Importance:	High
Description:	<p>After initialization of the ICU driver all notifications shall be disabled. All ICU channel status shall be set to INACTIVE. The wakeup-capability of a channel shall be disabled after the initialization.</p> <p>All used registers shall be initialized (including pending flags for interrupts) .</p>
Rationale:	The user of an ICU channel shall be responsible to enable / disable the

	notification.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.2.2 [BSW12429] ICU Deinitialization

Initiator:	Hella / MEDAG
Date:	29.11.2004
Short Description:	ICU Deinitialization
Type:	New
Importance:	High
Description:	The ICU Driver shall provide the functionality to deinitialize ICU channels to their power on reset state (not including registers which are not writeable).
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:	--

4.1.3.3 Normal Operation

4.1.3.3.1 [BSW12305] Enable/Disable notification during runtime

Initiator:	CAS
Date:	23,09.2004
Short Description:	Enable/Disable notification during runtime
Type:	Changed
Importance:	High
Description:	The ICU driver shall allow to enable/disable the notification for an ICU channel at runtime. The following options shall be available for each selected channel: <ul style="list-style-type: none"> • disable notification • enable notification on <ul style="list-style-type: none"> ○ rising edge (if supported by hardware) ○ falling edge (if supported by hardware) ○ both edges (if supported by hardware)
Rationale:	Adaptable notification according to the next expected edge(s).
Use Case:	Edge detection of a hall sensor. The disable Notification can be used to implement an anti saturation mechanism to avoid to jeopardize the entire system in case of too many interrupts coming to ICU.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.2 [BSW12369] Notification on signal edge

Initiator:	MEDAG
Date:	16.12.2005
Short Description:	Notification on signal edge
Type:	Changed
Importance:	High
Description:	The ICU driver shall provide notification for an ICU Channel at the configured signal edge (rising / falling / both edges) in the following configuration: <ul style="list-style-type: none"> • Notification function configured as no null pointer • and only if notifications are enabled
Rationale:	Notification on signal edge
Use Case:	Signal edge detection
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.3 [BSW12370] Sleep mode selection service

Initiator:	MEDAG
Date:	16.11.2004
Short Description:	Wake-up mode selection service
Type:	New
Importance:	High
Description:	The ICU driver shall provide a service for selecting the sleep mode: <ul style="list-style-type: none"> • Normal mode (mandatory) • sleep mode In Normal mode all notifications are available as configured. In sleep mode only those interrupts which cause wake-up capable notifications are available. All other interrupts 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 interrupts which are not required for the ECU wake-up.
Use Case:	During entry in the reduced power mode of an ECU all interrupts of the MCU shall be disabled without disabling the wake-up sources in between. Otherwise wake-up events can be lost.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.4 [BSW12371] ICU Channel status function

Initiator:	Renesas
Date:	15.11.2004
Short Description:	ICU Channel status function
Type:	New
Importance:	High
Description:	The ICU driver shall provide a synchronous service that returns the status of the ICU input. <ul style="list-style-type: none"> • If an activation edge has been detected this service will return ACTIVE. Once the service has returned the status ACTIVE the status

	<p>will be set to IDLE until the next edge is detected</p> <ul style="list-style-type: none"> If no activation edge has been detected this service will return IDLE
Rationale:	Polled access to input when notification is disabled
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.5 [BSW12438] Timestamps of elapsed time

Initiator:	MEDAG
Date:	06.12.2004
Short Description:	Timestamp of elapsed time
Type:	New
Importance:	High
Description:	<p>The ICU driver shall provide the functionality to capture timer values on configurable edges (rising edge / falling edge / both edges) to an external buffer.</p> <p>This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".</p>
Rationale:	--
Use Case:	Acquisition of high frequent and non-periodic sensor signals
Dependencies:	--
Conflicts:	--
Supporting Material:	See Figure 1: ICU time stamp

4.1.3.3.6 [BSW12455] External circular buffer handling

Initiator:	MEDAG
Date:	06.12.2004
Short Description:	External circular buffer handling
Type:	New
Importance:	High
Description:	<p>If circular buffer handling is configured, when the capture functionality reaches the end of the buffer, the driver restarts at the beginning of the external buffer.</p> <p>This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".</p>
Rationale:	--
Use Case:	High frequency continuous data acquisition
Dependencies:	--
Conflicts:	--
Supporting Material:	See Figure 1: ICU time stamp

4.1.3.3.7 [BSW12456] External linear buffer handling

Initiator:	MEDAG
Date:	06.12.2004
Short Description:	External Linear buffer handling
Type:	New
Importance:	High
Description:	<p>If linear buffer handling is configured, when the capture functionality reaches the end of the buffer, the driver stops capturing timer values.</p> <p>This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".</p>
Rationale:	--
Use Case:	High frequency non continuous data acquisition
Dependencies:	--
Conflicts:	--
Supporting Material:	See Figure 1: ICU time stamp

4.1.3.3.8 [BSW12430] ICU start timestamp service

Initiator:	MEDAG
Date:	06.12.2004
Short Description:	ICU start timestamp service
Type:	Changed
Importance:	High
Description:	<p>The ICU driver shall provide an asynchronous service for starting the timestamp measurement on an ICU channel. The passed parameters shall be</p> <ul style="list-style-type: none"> • ICU channel • Pointer to data buffer (destination for timestamps and signal levels) • Size of data buffer • Notification interval (events) <p>This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".</p>
Rationale:	Configure and enable timestamp capture
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.9 [BSW12431] ICU cancel timestamp service

Initiator:	MEDAG
Date:	06.12.2004
Short Description:	ICU cancel timestamp service
Type:	New
Importance:	High
Description:	<p>The ICU driver shall provide a synchronous service for canceling the timestamp measurement on an ICU channel. The passed parameters shall be</p>

	<ul style="list-style-type: none"> ICU channel <p>This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".</p>
Rationale:	Disable timestamp capture
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.10 [BSW12444] ICU timestamp notification

Initiator:	MEDAG
Date:	06.12.2004
Short Description:	ICU timestamp notification
Type:	New
Importance:	High
Description:	<p>The ICU driver shall provide a notification if the number of requested timestamps (Notification interval) are acquired.</p> <p>This functionality shall be available for each ICU channel in Measurement mode "Timestamp".</p>
Rationale:	Enable summarizing of notification during timestamping
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.11 [BSW12453] ICU get timestamp index service

Initiator:	WP4.2.2.1.12
Date:	16.02.2005
Short Description:	ICU get timestamp index service
Type:	New
Importance:	High
Description:	<p>The ICU driver shall provide a synchronous service for reading the current timestamp index of the driver. The passed parameters shall be</p> <ul style="list-style-type: none"> ICU channel <p>This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".</p>
Rationale:	Read the current timestamp index within the buffer
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.12 [BSW12439] Edge counting

Initiator:	MEDAG
Date:	06.12.2004

Short Description:	Edge counting
Type:	Changed
Importance:	High
Description:	<p>The ICU driver shall provide the functionality to count edges of a signal.</p> <p>Only the configured edges are counted (rising edge / falling edge / both edges)</p> <p>This functionality shall be available for each ICU channel in Measurement Mode "Edge Counter"</p>
Rationale:	--
Use Case:	Counting of high frequency events.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.13 [BSW12432] Enable ICU edge counting service

Initiator:	MEDAG
Date:	06.12.2004
Short Description:	Enable ICU edge counting service
Type:	Changed
Importance:	High
Description:	<p>The ICU driver shall provide a synchronous service for enabling the edge counting on an ICU channel. The passed parameters shall be</p> <ul style="list-style-type: none"> • ICU channel <p>This functionality shall be available for each ICU channel in Measurement Mode "Edge Counter".</p>
Rationale:	Basic functionality.
Use Case:	Counting of edges over a defined time span.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.14 BSW13100 Reset the value of counted edges

Initiator:	MEDAG
Date:	29.11.2005
Short Description:	Reset the value of counted edges
Type:	New
Importance:	High
Description:	<p>The ICU driver shall provide a synchronous service for resetting the counted edges of an ICU channel. The passed parameters shall be</p> <ul style="list-style-type: none"> • ICU channel <p>This functionality shall be available for each ICU channel in Measurement Mode "Edge Counter".</p>
Rationale:	Separate the start of the counting from the reset of the counted edges.
Use Case:	Pulse counting for electrical seat positioning (if the seat stops and then moves again, the position (equivalent to the counted edges) shall be kept.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.15 [BSW12433] Disable ICU edge counting service

Initiator:	MEDAG
Date:	06.12.2004
Short Description:	Disable ICU edge counting service
Type:	New
Importance:	High
Description:	<p>The ICU driver shall provide a synchronous service for disabling the edge counting on an ICU channel. The passed parameter shall be</p> <ul style="list-style-type: none"> • ICU channel <p>This functionality shall be available for each ICU channel in Measurement Mode "Edge Counter".</p>
Rationale:	Basic functionality.
Use Case:	Counting of edges over a defined time span.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.16 [BSW12434] ICU edge counting read service

Initiator:	MEDAG
Date:	06.12.2004
Short Description:	ICU edge counting read service
Type:	New
Importance:	High
Description:	<p>The ICU driver shall provide a synchronous service reading the number of counted edges after the last call of the "Enable ICU edge counting service". The passed parameter shall be</p> <ul style="list-style-type: none"> • ICU channel <p>This functionality shall be available for each ICU channel in Measurement Mode "Edge Counter".</p>
Rationale:	Reading the number of counted edges after the last call
Use Case:	--
Dependencies:	--

Conflicts:	--
Supporting Material:	--

4.1.3.3.17 [BSW12442] Get elapsed Signal Low Time for an ICU Channel

Initiator:	Valeo
Date:	02.12.2004
Short Description:	Get elapsed Signal Low Time for an ICU Channel
Type:	New
Importance:	High
Description:	The ICU driver shall provide a service to get the elapsed Signal Low Time for each ICU Channel that is configured in Measurement Mode "Signal Measurement, Signal Low Time". The elapsed time is measured between a falling edge and the consecutive rising edge of the channel.
Rationale:	get the elapsed Signal Low Time
Use Case:	PWD
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.18 [BSW12435] Get elapsed Signal High Time for an ICU Channel

Initiator:	Valeo
Date:	02.12.2004
Short Description:	Get elapsed Signal High Time for an ICU Channel
Type:	New
Importance:	High
Description:	The ICU driver shall provide a service to get the Signal High Time for each ICU Channel that is configured in Measurement Mode "Signal Measurement, Signal High Time". The elapsed time is measured between a rising edge and the consecutive falling edge of the channel.
Rationale:	--
Use Case:	PWD
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.19 [BSW12443] Get elapsed Period Time for an ICU Channel

Initiator:	Valeo
Date:	02.12.2004
Short Description:	Get elapsed Period Time for an ICU Channel
Type:	New
Importance:	High
Description:	The ICU driver shall provide a service to get the elapsed Period time for each ICU Channel that is configured in Measurement Mode "Signal Measurement, Period Time". The elapsed time is measured between two consecutive rising (or falling) edges of the channel.

Rationale:	--
Use Case:	PWD
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.20 [BSW12436] Get Duty Cycle input values for an ICU Channel

Initiator:	WP 4.2.1.1.12
Date:	07.12.2004
Short Description:	Get Duty Cycle input values for an ICU Channel
Type:	Changed
Importance:	High
Description:	The ICU driver shall provide a service to get the coherent High Time and Period Time for each ICU Channel that is configured in Measurement Mode "Signal Measurement, Duty Cycle.
Rationale:	Basic functionality, Service delivers the values for Duty cycle calculation. The correct calculation and scaling is done by the user of ICU module.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.3.21 [BSW12437] ICU driver time unit

Initiator:	WP4.2.2.1.12
Date:	07.12.2004
Short Description:	ICU driver time unit
Type:	New
Importance:	High
Description:	All time units used within the API services of the ICU driver shall be of the unit ticks.
Rationale:	Conversions between microseconds and ticks shall be part of the ECU Abstraction Layer.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--

4.1.3.4 Shutdown Operation

4.1.3.4.1 [BSW12408] Wake-up enable / disable service

Initiator:	WP4.2.2.1.12
Date:	25.11.2004
Short Description:	Wake-up enable / disable service
Type:	New
Importance:	High
Description:	The ICU driver shall provide a service for enabling / disabling the wake-up capability of single ICU channels.

Rationale:	Controlling the wake-up conditions of a MCU needs to enable or disable the HW interrupts, not only some notification conditions.
Use Case:	Limitation of the wake-up occurrences in the case of error condition.
Dependencies:	--
Conflicts:	--
Supporting Material:	--

5 Related Documentation

- [1] Glossary
https://svn2.autosar.org/repos2/22_Releases/AUTOSAR_Glossary.pdf
- [2] Layered Software Architecture
https://svn2.autosar.org/repos2/22_Releases/AUTOSAR_LayeredSoftwareArchitecture.pdf
- [3] General Requirements on Basic
https://svn2.autosar.org/repos2/22_Releases/AUTOSAR_SRS_General.pdf
- [4] General Requirements on SPAL
https://svn2.autosar.org/repos2/22_Releases/AUTOSAR_SRS_SPAL_General.pdf

5.1 Related standards and norms

5.1.1 HIS

- [5] HIS API I/O Driver Specification V2.1.3