Document Title	Requirements on ICU Driver
Document Category	Requirement specification (RS)
Document UID	112
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R26-11

Document Change History

Date	Release	Changed by	Description
2026-11-26	R26-11	AUTOSAR Release Management	• No content changes
2025-11-27	R25-11	AUTOSAR Release Management	• No content changes
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	• 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	• Updated according to FO_TPS_StandardizationTemplate [1] • Formal rework of requirement tracing
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	• "Advice for users" revised • "Revision Information" added
2006-11-28	2.1	AUTOSAR Administration	• Legal disclaimer revised
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. • Removed functionalities: – Hardware gated edge counting – Notification on signal level • Approved requirements for functionality: – Timestamping – Wakeup – Edge-Counting – Get elapsed Signal Times – Get Duty Cycle input values • Added requirements for functionality: – Reset the value of counted edges
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.

Table of Contents

1 Scope of Document 5

2 Conventions to be used 6
2.1 Document Conventions . . . 6
2.2 Requirements structure . . . 7

3 Acronyms and abbreviations 8

4 Requirements Specification 9
4.1 Functional Overview . . . 9
4.1.1 ICU Driver . . . 9
4.2 Functional Requirements . . . 12
4.2.1 ICU Driver . . . 12
4.2.1.1 Configuration . . . 12
4.2.1.2 Initialization . . . 13
4.2.1.3 Normal Operation . . . 14
4.2.1.4 Shutdown Operation . . . 22

5 References 23

A Change history of AUTOSAR traceable items 24
A.1 Traceable item history of this document according to AUTOSAR Release R26-11 . . . 24
A.1.1 Added Requirements in R26-11 . . . 24
A.1.2 Changed Requirements in R26-11 . . . 24
A.1.3 Deleted Requirements in R26-11 . . . 24
A.2 Traceable item history of this document according to AUTOSAR Release R25-11 . . . 24
A.2.1 Added Requirements in R25-11 . . . 24
A.2.2 Changed Requirements in R25-11 . . . 24
A.2.3 Deleted Requirements in R25-11 . . . 24

1 Scope of Document

This document specifies requirements on the module ICU Driver.

Constraints

First scopes 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 Conventions to be used

2.1 Document Conventions

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see [1, Standardization Template].

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

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)
- 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 ICU Driver that are not included in the AUTOSAR Glossary [2].

Abbreviation / 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.
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: • 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
STD	Standard
UNINIT	Uninitialized (= not initialized)

Table 3.1: Acronyms and 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 ICU Driver.

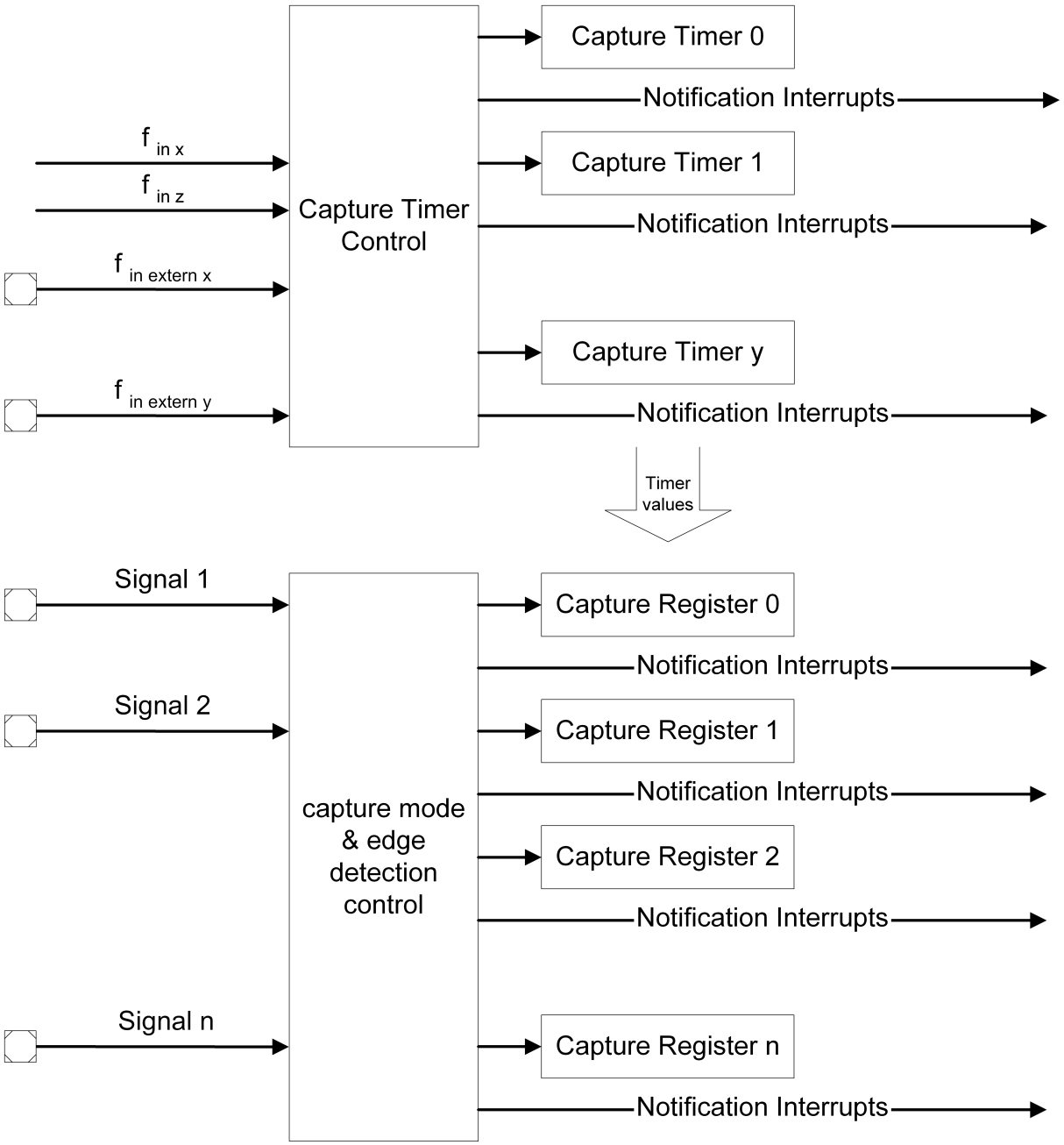
4.1 Functional Overview

4.1.1 ICU Driver

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

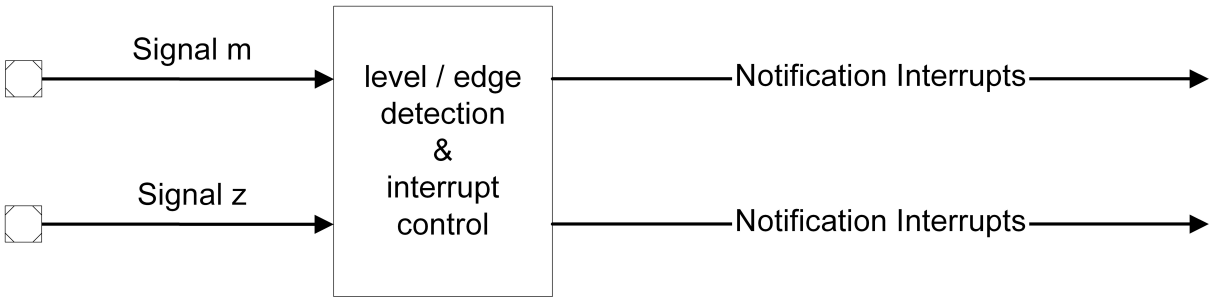
- Period-, Low-, High time measurement
- Edge detection and notification
- Edge counting
- Edge time stamping
- Wake-up interrupts

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.

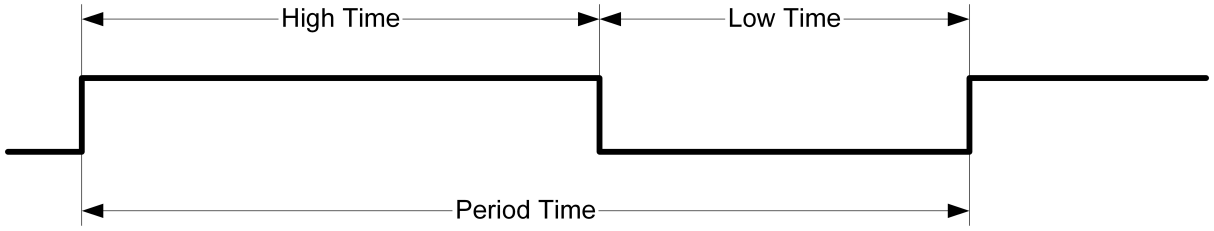


Figure 4.1: ICU time definitions

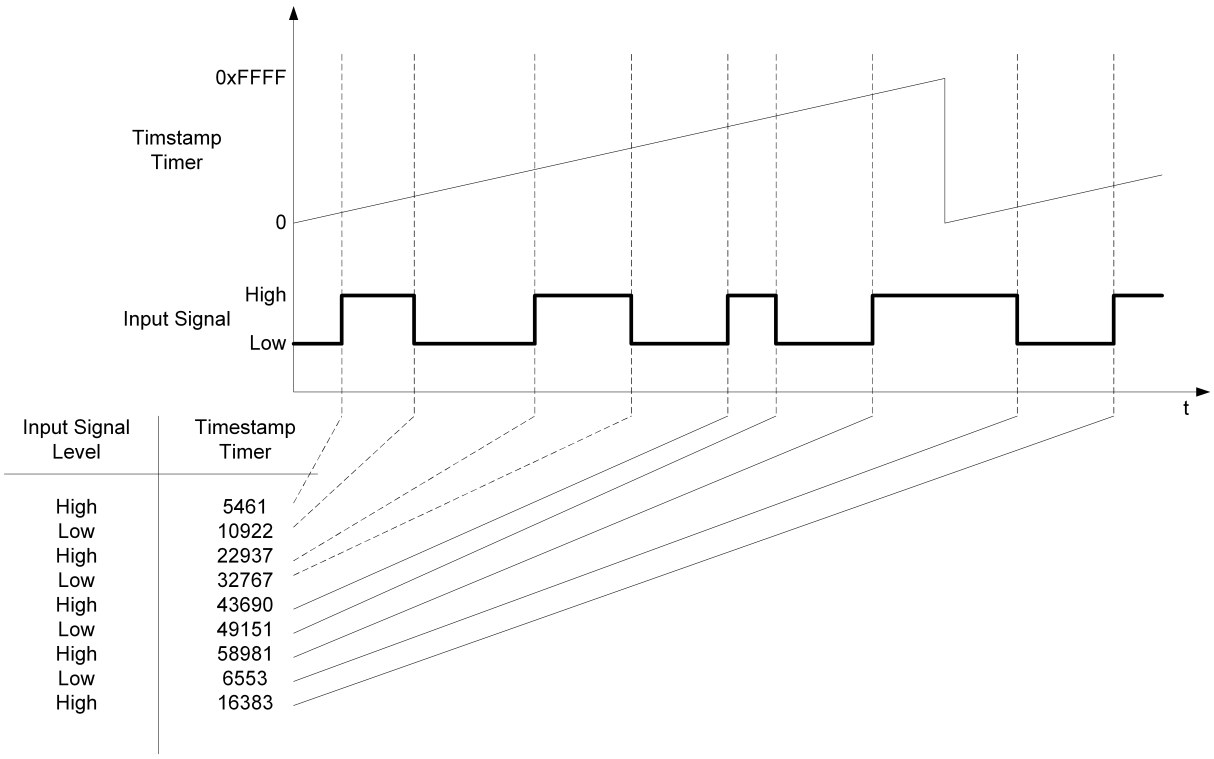


Figure 4.2: ICU time stamp

4.2 Functional Requirements

4.2.1 ICU Driver

4.2.1.1 Configuration

[SRS_Icu_12327] The ICU driver shall allow the configuration of global parameters ⌈

Description:	The ICU driver shall allow the configuration of the following parameters: • 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:	–
Supporting Material:	–

⌋

[SRS_Icu_12368] The ICU driver shall support basic static configurations per channel ⌈

Description:	The ICU driver shall support the following basic static configurations per channel: Mandatory parameters • Symbolic name for channel • Notification function(s) • Wake-up capability Optional parameters • 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) – Signal Edge Detection / Notification – Signal Measurement – Timestamp – Edge Counter • If measurement mode is "timestamp measurement", buffer handling shall be configurable. The values shall be: – Circular buffer handling – Linear buffer handling • 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
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Icu_12425] For each ICU Channel the 'property' that could be measured shall be configurable ⌈

Description:	For each ICU Channel the 'property' that could be measured shall be configurable and the values available (at least) shall be : • High Level • Low Level • Period Time
Rationale:	Definition of the measurement purpose to allocate the required HW resources during configuration.
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

4.2.1.2 Initialization

[SRS_Icu_12407] After initialization of the ICU driver all notifications shall be disabled ⌈

Description:	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. All used registers shall be initialized (including pending flags for interrupts) .
Rationale:	The user of an ICU channel shall be responsible to enable / disable the notification.
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Icu_12429] The ICU Driver shall provide the functionality to deinitialize ICU channels to their power on reset state ⌈

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 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:	–

⌋

4.2.1.3 Normal Operation

[SRS_Icu_12305] The ICU driver shall allow to enable/disable the notification for an ICU channel at runtime ⌈

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: • disable notification • enable notification on – 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:	–
Supporting Material:	–

⌋

[SRS_Icu_12369] The ICU driver shall provide notification for an ICU Channel at the configured signal edge ⌈

Description:	The ICU driver shall provide notification for an ICU Channel at the configured signal edge (rising / falling / both edges) in the following configuration: • Notification function configured as no null pointer • and only if notifications are enabled
Rationale:	Notification on signal edge
Use Case:	Signal edge detection
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Icu_12370] The ICU driver shall provide a service for selecting the sleep mode ⌈

Description:	The ICU driver shall provide a service for selecting the sleep mode: • 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:	–
Supporting Material:	–

⌋

[SRS_Icu_12371] The ICU driver shall provide a synchronous service that returns the status of the ICU input ⌈

Description:	The ICU driver shall provide a synchronous service that returns the status of the ICU input. • If an activation edge has been detected this service will return ACTIVE. Once the service has returned the status ACTIVE the status will be set to IDLE until the next edge is detected • If no activation edge has been detected this service will return IDLE

▽

△

Rationale:	Polled access to input when notification is disabled
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Icu_12438] The ICU driver shall provide the functionality to capture timer values on configurable edges to an external buffer ⌈

Description:	The ICU driver shall provide the functionality to capture timer values on configurable edges (rising edge / falling edge / both edges) to an external buffer. This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".
Rationale:	–
Use Case:	Acquisition of high frequent and non-periodic sensor signals
Dependencies:	–
Supporting Material:	See Figure “ICU time stamp”

⌋

[SRS_Icu_12455] If circular buffer handling is configured, the driver shall restart at the beginning of the external buffer, when the end of the buffer is reached ⌈

Description:	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. This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".
Rationale:	–
Use Case:	High frequency continuous data acquisition
Dependencies:	–
Supporting Material:	See Figure “ICU time stamp”

⌋

[SRS_Icu_12456] If linear buffer handling is configured, the driver shall stop capturing timer values, when the end of the buffer is reached ⌈

Description:	If linear buffer handling is configured, when the capture functionality reaches the end of the buffer, the driver stops capturing timer values. This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".
Rationale:	–

▽

△

Use Case:	High frequency non continuous data acquisition
Dependencies:	–
Supporting Material:	See Figure “ICU time stamp”

⌋

[SRS_Icu_12430] The ICU driver shall provide an asynchronous service for starting the timestamp measurement on an ICU channel ⌈

Description:	The ICU driver shall provide an asynchronous service for starting the timestamp measurement on an ICU channel. The passed parameters shall be • ICU channel • Pointer to data buffer (destination for timestamps and signal levels) • Size of data buffer • Notification interval (events) This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".
Rationale:	Configure and enable timestamp capture
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Icu_12431] The ICU driver shall provide a synchronous service for canceling the timestamp measurement on an ICU channel ⌈

Description:	The ICU driver shall provide a synchronous service for canceling the timestamp measurement on an ICU channel. The passed parameters shall be • ICU channel This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".
Rationale:	Disable timestamp capture
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Icu_12444] The ICU driver shall provide a notification if the number of requested timestamps are acquired ⌈

Description:	The ICU driver shall provide a notification if the number of requested timestamps (Notification interval) are acquired. This functionality shall be available for each ICU channel in Measurement mode "Timestamp".
Rationale:	Enable summarizing of notification during timestamping
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Icu_12453] The Timestamp index service shall be provided by ICU ⌈

Description:	The ICU driver shall provide a synchronous service for reading the current timestamp index of the driver. The passed parameters shall be • ICU channel This functionality shall be available for each ICU channel in Measurement Mode "Timestamp".
Rationale:	Read the current timestamp index within the buffer
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Icu_12439] Edges of a signal shall be counted by the ICU ⌈

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

⌋

[SRS_Icu_12432] Edge counting service shall be available on an ICU channel ⌈

Description:	The ICU driver shall provide a synchronous service for enabling the edge counting on an ICU channel. The passed parameters shall be • ICU channel This functionality shall be available for each ICU channel in Measurement Mode "Edge Counter".
Rationale:	Basic functionality.
Use Case:	Counting of edges over a defined time span.
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Icu_13100] Reseting the value of counted edges of an ICU channel shall be available ⌈

Description:	The ICU driver shall provide a synchronous service for resetting the counted edges of an ICU channel. The passed parameters shall be • ICU channel This functionality shall be available for each ICU channel in Measurement Mode "Edge Counter".
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:	–
Supporting Material:	–

⌋

[SRS_Icu_12433] Edge counting service on a ICU channel shall be disabled ⌈

Description:	The ICU driver shall provide a synchronous service for disabling the edge counting on an ICU channel. The passed parameter shall be • ICU channel This functionality shall be available for each ICU channel in Measurement Mode "Edge Counter".
Rationale:	Basic functionality.
Use Case:	Counting of edges over a defined time span.
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Icu_12434] Edge counting read service shall be available ⌈

Description:	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 • ICU channel This functionality shall be available for each ICU channel in Measurement Mode "Edge Counter".
Rationale:	Reading the number of counted edges after the last call
Use Case:	–
Dependencies:	–
Supporting Material:	–

⌋

[SRS_Icu_12442] The elapsed Signal Low Time for each ICU Channel shall be provided ⌈

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:	–
Supporting Material:	–

⌋

[SRS_Icu_12435] The elapsed Signal High Time for each ICU Channel shall be provided ⌈

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:	–
Supporting Material:	–

⌋

[SRS_Icu_12443] The elapsed Period Time for an ICU Channel shall be provided ⌈

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:	–
Supporting Material:	–

⌋

[SRS_Icu_12436] The High time and Period Time of an ICU Channel shall be provided ⌈

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:	–
Supporting Material:	–

⌋

[SRS_Icu_12437] All time units used within the API services of the ICU driver shall be of the unit ticks ⌈

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:	–
Supporting Material:	–

⌋

4.2.1.4 Shutdown Operation

[SRS_Icu_12408] The ICU driver shall provide a service for enabling / disabling the wake-up capability of single ICU channels ⌈

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:	–
Supporting Material:	–

⌋

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.

5 References

[1] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[2] Glossary
AUTOSAR_FO_TR_Glossary

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 R26-11

A.1.1 Added Requirements in R26-11

none

A.1.2 Changed Requirements in R26-11

none

A.1.3 Deleted Requirements in R26-11

none

A.2 Traceable item history of this document according to AUTOSAR Release R25-11

A.2.1 Added Requirements in R25-11

none

A.2.2 Changed Requirements in R25-11

none

A.2.3 Deleted Requirements in R25-11

none