Document Title	Classic Platform Release Overview
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	0

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R21-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2021-11-25	R21-11	AUTOSAR Release Management	Release Life Cycle Status: R21-11 is in Evolution, R21-11 supersedes R20-11

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

References

[1] Specification of LIN Interface
AUTOSAR_SWS_LINInterface

[2] Log and Trace Protocol Specification with protocol version "2"
AUTOSAR_PRS_LogAndTraceProtocol from Release R21-11

[3] Log and Trace Protocol Specification with protocol version "1"
AUTOSAR_PRS_LogAndTraceProtocol from Release R20-11

[4] SOME/IP Protocol Specification
AUTOSAR_PRS_SOMEIPProtocol

[5] SAE J1939-81 Network Management

1 Introduction

1.1 Scope of this document

This document provides an overview of the AUTOSAR standard Classic Platform Release R21-11.

1.2 Terminology and Licenses

1.2.1 Terminology statement

AUTOSAR has identified a use of previously common terminology that can be considered oppressive or racist, such as master/slave and black/white list, or in other contexts such as gender or age as harmful connotations. AUTOSAR has started a discussion with all the working groups to replace these terms. AUTOSAR is committed to provide all specification documents without these terminology in the coming and future releases. Nevertheless, it may take several releases before the terms are completely replaced, as AUTOSAR has to continue its operations and thousands of pages of existing specifications have to be reviewed and updated in parallel.

1.2.2 Usage of W3C XML schema

The AUTOSAR XML Schema requires the XML namespace definition file xml.xsd.

There are several occurrences of the "xml.xsd" file within this release. For all occurrences the W3C license applies which can be found on `https://www.w3.org/Consortium/Legal/2015/copyright-software-and-document`.

License

Disclaimers

1.3 AUTOSAR Standards

1.3.1 Introduction

AUTOSAR addresses a wide range of use cases in automotive software development with its standards. These use cases have different requirements and lead to different technical solutions.

Packaging its deliverables into different "standards"

- eases the access to AUTOSAR solutions for users and
- allows AUTOSAR to scale with market needs.

1.3.2 Definition

An AUTOSAR standard is a consistent set of AUTOSAR deliverables, which are released at the same time. AUTOSAR deliverables can, but are not limited to be of the following kinds:

- textual explanations
- textual specifications
- test specifications
- source code
- other formal or semi-formal textual formats (e.g. ARXML, UML models, XML Schemata)

At the time of release, AUTOSAR ensures that dependencies are fulfilled.

1.3.3 Overview on AUTOSAR's Standards

AUTOSAR delivers the following standards:

Standard	Abbreviation
Adaptive Platform	AP
Classic Platform	CP
Foundation	FO

1.3.3.1 Adaptive Platform

The Adaptive Platform is AUTOSAR's solution for high-performance computing ECUs to build safety-related systems for use cases such as highly automated and autonomous driving.

1.3.3.2 Classic Platform

The Classic Platform is AUTOSAR's solution for embedded systems with hard real-time and safety constraints.

1.3.3.3 Foundation

The purpose of the Foundation standard is to enforce interoperability between the AUTOSAR platforms.

Foundation contains the generic artifacts that are common for AP and CP to ensure compatibility between

- Classic- and Adaptive Platform
- Non-AUTOSAR platforms to AUTOSAR platforms

1.3.4 Dependencies between Standards

Each release of Classic and Adaptive Platform relies on a dedicated version of Foundation. The specific dependency is documented in chapter 1.4.6.

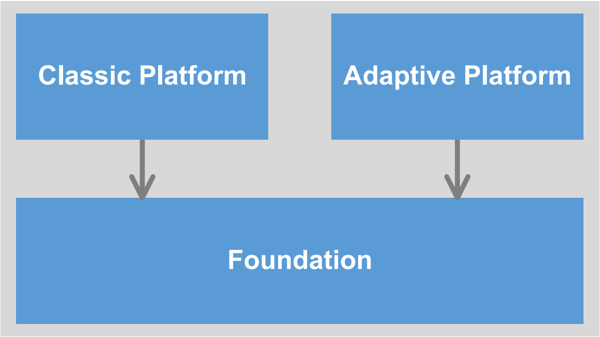


Figure 1.1: Dependencies of AUTOSAR Standards

1.3.5 Dependencies to other Standards

This release of the Classic Platform depends on the standard Foundation in release R21-11, which

- defines protocols implemented by Classic Platform
- contains the project objectives and the common requirements from which the features of the Classic Platform are derived
- contains common specification parts which apply to both, the Adaptive Platform and the Classic Platform.

These dependencies are refined in the trace information of the requirements in the respective specifications.

1.4 Release Numbering and Life Cycle

1.4.1 Platform release number

AUTOSAR applies a four-digit numbering scheme Ryy-mm to identify releases. The identifiers “yy” and “mm” depict the year and month of the release date, e.g. R20-11 for the November 2020 release.

1.4.2 Internal release number

AUTOSAR additionally maintains an internal release number for different purposes (e.g. usage in BSW modules in Classic Platform).
The internal release number is used for all platforms and follows up on the Classic Platform release number. In Adaptive Platform this is newly introduced. In Foundation this leads to a discontinuation of the former numbering pattern (e.g. R1.5.0).
A mapping list between Platform Releases and corresponding internal release num-

bers can be found in chapter 1.4.5. The internal release number uses a three-digit numbering scheme R<major>.<minor>.<revision> to identify releases. Its primary purpose is to identify a release as

- a major release: Valid and draft specification parts may be changed backward incompatibly.
- a minor release: Valid specification parts may only be changed backward compatibly. Draft specification parts may be changed backward incompatibly.
- a revision: Does not contain extensions but only backward compatible bugfixes.

1.4.3 Release life cycle of a major release

Each major release goes through four consecutive steps within its life cycle (examples based on the internal release numbering scheme):

1. Development: Between start of life cycle and the initial release (e.g. R4.0.1)
2. Evolution: Following the initial release with zero, one or several minor releases and/or revisions (e.g. R4.0.2, R4.1.1)
3. Maintenance: No new content is added to a major release but only maintenance of the existing content with zero, one or several revisions (e.g. R3.2.2) is provided
4. Issue Notice: No more revisions but zero, one or several issue notices, i.e. updates of the list of known issues until end of life cycle.

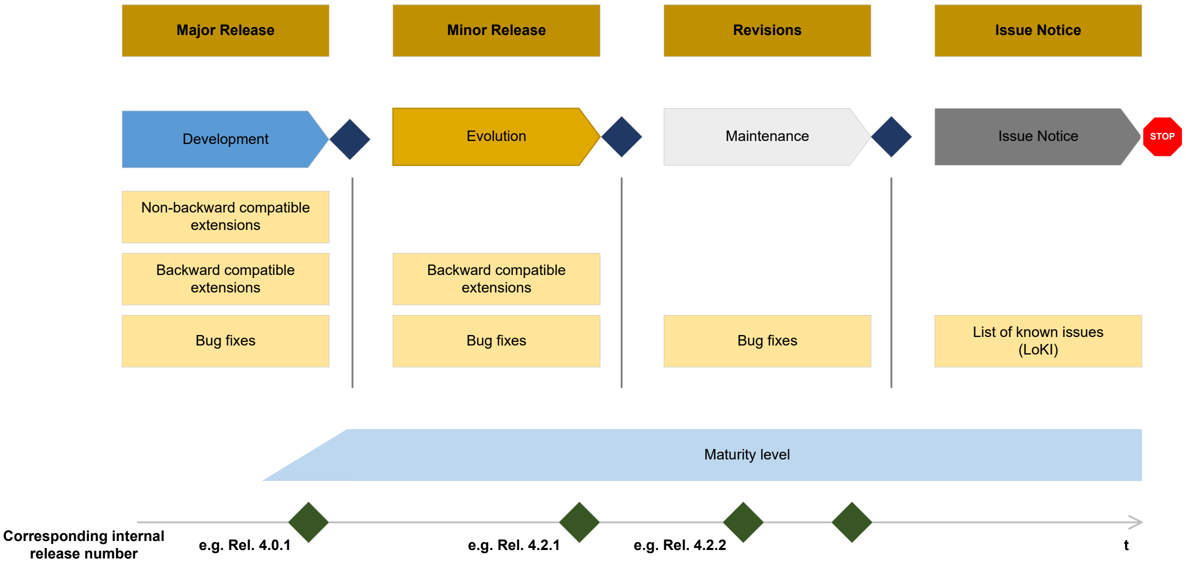


Figure 1.2: Life cycle model of AUTOSAR standards

1.4.4 Life cycle states of specification items and requirements

The life cycle state of a specification item is found after the specification item ID surrounded by curly brackets. The states are:

- {Valid}: This indicates that the related entity is a valid part of the document. This is the default and also applies if no dedicated life cycle status is annotated for the related entity.
- {Draft}: This indicates that the related entity is newly introduced but still experimental. This information is published but is subject to change without backward compatibility guarantee.
- {Obsolete}: This indicates that the related entity is subject to be removed in one of the following releases without further notice.

The life cycle state of a requirement is found in the attribute "type". The states are the same as the specification item states.

1.4.5 Overview of AUTOSAR schema versions and corresponding internal AUTOSAR releases

Schema Version	Platform release	Internal release number
AUTOSAR_00048	R19-11	R4.5.0
AUTOSAR_00049	R20-11	R4.6.0
AUTOSAR_00050	R21-11	R4.7.0

According to the release life cycle of AUTOSAR the release R21-11 is a minor release.

1.4.6 Overview of AUTOSAR schema versions and corresponding valid AUTOSAR releases

The AUTOSAR schema does not have an impact on the Foundation. The Foundation releases are mentioned for the sake of completeness.

Schema Version	Classic Platform release	Adaptive Platform release	Foundation release
AUTOSAR_00042	R4.3.0	R17-03	R1.1.0
AUTOSAR_00043	R4.3.0	R17-10	R1.2.0
AUTOSAR_00044	R4.3.1	R17-10	R1.3.0
AUTOSAR_00045	R4.3.1	R18-03	R1.4.0
AUTOSAR_00046	R4.4.0	R18-10	R1.5.0
AUTOSAR_00047	R4.4.0	R19-03	R1.5.1

Schema Version	AUTOSAR release
AUTOSAR_00048	R19-11
AUTOSAR_00049	R20-11
AUTOSAR_00050	R21-11

1.5 Content of chapters

This document is structured as follows:

- Chapter 1 provides an introduction to AUTOSAR's release strategy and its standardization approach.
- Chapter 2 provides a summary of changes since the previous release of the Classic Platform.
- Chapter 3 contains the overview of specifications comprising the AUTOSAR release R21-11. This chapter is structured according to the clusters of AUTOSAR release R21-11.
- Chapter 4 contains remarks about known technical deficiencies.
- Chapter 5 contains the detailed revision history of all released specifications.

2 Summary of changes

This chapter contains a summary of changes which have been implemented since the previous release R20-11.

2.1 Release R21-11

Several concepts mainly affecting the Classic Platform have been introduced with release R21-11 thereby adding functionalities to the platform (10BASE-T1S, Classic Platform Flexibility, SD Harmonization, Memory Stack Rework, Rework of PNC related ComM and NM).
Additionally one concept targets the Classic and Adaptive Platform, strengthening the interaction between the two platforms.

2.1.1 Concepts

2.1.1.1 Introduced Concepts

The following concepts in 2.1.1.1.1 - 2.1.1.1.6 have been introduced.

2.1.1.1.1 10BASE-T1S

This concept introduces the support of Ethernet 10BASE-T1S specified by IEEE802.3cg and enables bus topologies in Ethernet networks. This new extension localized on layers 1 and 2 of the OSI model is to be supported by Classic Platform as well as Adaptive.

Currently the support is for the two available HW solutions:

- for external MAC controller for 10BASE-T1S over SPI
- PHYs over MII

Additionally the concept provides an optional buffer handling for HW that does not support several HW queues. The handling of link-up for 10BASE-T1S in multidrop mode is also considered.

2.1.1.1.2 Classic Platform Flexibility

The concept supports signal and SOME/IP based communication from Software Components located in an Applicative Software Cluster - independently buildable from the Host Software Cluster with its communication stack. The Software Cluster Connection supports now the connection to diagnostic services as Dem, FiM, Dcm. The supervision of safety related functionality is supported with an individual WdgM instance in each Software Cluster.

2.1.1.1.3 Rework of PNC related ComM and NM

Replaced synchronization of PN information via ComSignals between ComM and Nm with dedicated APIs and introduced simplification of PNC related functionalities in ComM, NmInterface and <Bus>Nms (e.g. PN timer handling). Clarified the supported PNC gateway use cases and introduced according configuration extensions in ComM. Clarified the handling of multiple top-level PNC coordinators within the same PN topology.

2.1.1.1.4 Service Discovery Harmonization

The concept SDHarmonization cleans-up the Service Discovery specification, by removing duplicate content and fixing contradicting statements in the documents FO PRS SOMEIPServiceDiscoveryProtocol and CP SWS ServiceDiscovery.

2.1.1.1.5 Memory Stack Rework

The concept extends the existing memory stack by the lower layer components MemAcc and Mem to support new use cases like over the air (OTA) software update by providing memory access coordination for multiple upper layer modules and a memory technology agnostic memory driver interface.

2.1.1.1.6 E2E For Fields

This concept extends the E2E protection to fields, the publisher subscriber pattern of AUTOSAR applications. The E2E protection is applied to Getter/Setter functions of fields and the notification of subscriber applications.

2.1.1.2 Impact of Concepts

The introduced concepts had impact on several specifications. The following table provides a detailed overview.

Please note that some of the specifications are marked by special text formatting:

- Specifications in **bold** font are completely new specifications originating from the particular concept.

- Specifications in *italic* font are affected indirectly as they provide artefacts for the actually impacted specifications.

Concept Name	Specification Long Name	Standard	Concept Lifecycle
10BASE-T1S	Specification of Ethernet Driver	Classic Platform	draft
	Specification of Ethernet Interface		
	Specification of Ethernet Switch Driver		
	Specification of Ethernet Transceiver Driver		
	Requirements on SPI Handler/Driver		
Classic Platform Flexibility	Requirements on Diagnostics	Foundation	draft
	Requirements on Software Cluster Connection module	Classic Platform	
	Specification of Software Cluster Connection module		
	Explanation of CP Software Cluster Design And Integration Guideline		
	System Template		
	Specification of RTE Software		
	Requirements on Communication		
	Specification of Communication		
	Specification of Large Data COM		
	Specification of Diagnostic Communication Manager		
	Specification of Diagnostic Event Manager		
	Diagnostic Extract Template		
	Specification of PDU Router		
	Specification of Watchdog Manager		
	Specification of Communication Stack Types		
Service Discovery Harmonization	SOME/IP Service Discovery Protocol Specification	Foundation	draft
	Specification of Service Discovery	Classic Platform	
Memory Stack Rework	Specification of NVRAM Manager	Classic Platform	draft
	Layered Software Architecture		
	Requirements on Memory Hardware Abstraction Layer		
	List of Basic Software Modules		

▽

△

Concept Name	Specification Long Name	Standard	Concept Lifecycle
	Specification of Memory Abstraction Interface		
	Specification of Flash EEPROM Emulation		
	Specification of EEPROM Abstraction		
	Guide to BSW Distribution		
	Specification of Memory Access		
	Specification of Memory Driver		
E2E For Fields	Explanation of Diagram Source	Foundation	draft
	Specification of Module E2E Transformer	Classic Platform	
	Specification of Communication Management	Adaptive Platform	
Rework of PNC related ComM and NM	Glossary	Foundation	draft
	Requirements on Network Management	Classic Platform	
	Specification of CAN Network Management		
	Specification of FlexRay Network Management		
	Specification of FlexRay Transport Layer		
	System Template		
	Requirements on Mode Management		
	Specification of Communication Manager		
	Specification of UDP Network Management		
	Specification of Ethernet State Manager		
	Specification of Ethernet Interface		
	Specification of Ethernet Driver		
	Specification of FlexRay Network Management		

Table 2.1: Impact of Concepts

2.1.1.3 Validated Concepts

The following concepts have been validated:

- 10BASE-T1S (Part 1)
- Unified AUTOSAR Timing and Tracing Approach (Part 2)

- Ethernet WakeUp On DataLine
- Rework of PNC related ComM and NM handling (Part 1) - all features have been validated, except feature "SLAVE_PASSIVE"

2.1.2 Specifications

2.1.2.1 New Specifications

The following new specifications have been introduced via concepts:

- Specification of Memory Access (UID 1017, SWS)
- Specification of Memory Driver (UID 1018, SWS)

In addition to the above listed new specifications, the following documents have been added with R21-11:

- Explanation of Classic Platform Software Architectural Decisions (UID 1028, EXP)

2.1.2.2 Migrated Specifications

With this release, the following specifications have been moved from AUTOSAR Classic Platform to the AUTOSAR Foundation standard:

- Requirements on Network Management (UID 3, SRS) to Requirements on AUTOSAR Network Management (UID 927, RS)

2.1.2.3 Obsolete Specifications

The following specifications have been set to status "obsolete" in this release:

- Requirements on Safety Extensions (UID 670, RS)
- Specification of Safety Extensions (UID 671, TPS)
- Specification of Compiler Abstraction (UID 51, SWS)
- Requirements on AUTOSAR Features (UID 294, RS)

2.1.2.4 Removed Specifications

The following specifications have been set to status "removed" in this release and hence are not released anymore:

- General Requirements on Methodology and Templates (UID 604, RS)

2.1.2.5 Reworked Specifications

The following documents have been changed fundamentally in R21-11:

- none

2.1.2.6 Moved Specification parts

The following specification parts have been moved to other documents in R21-11.

- none

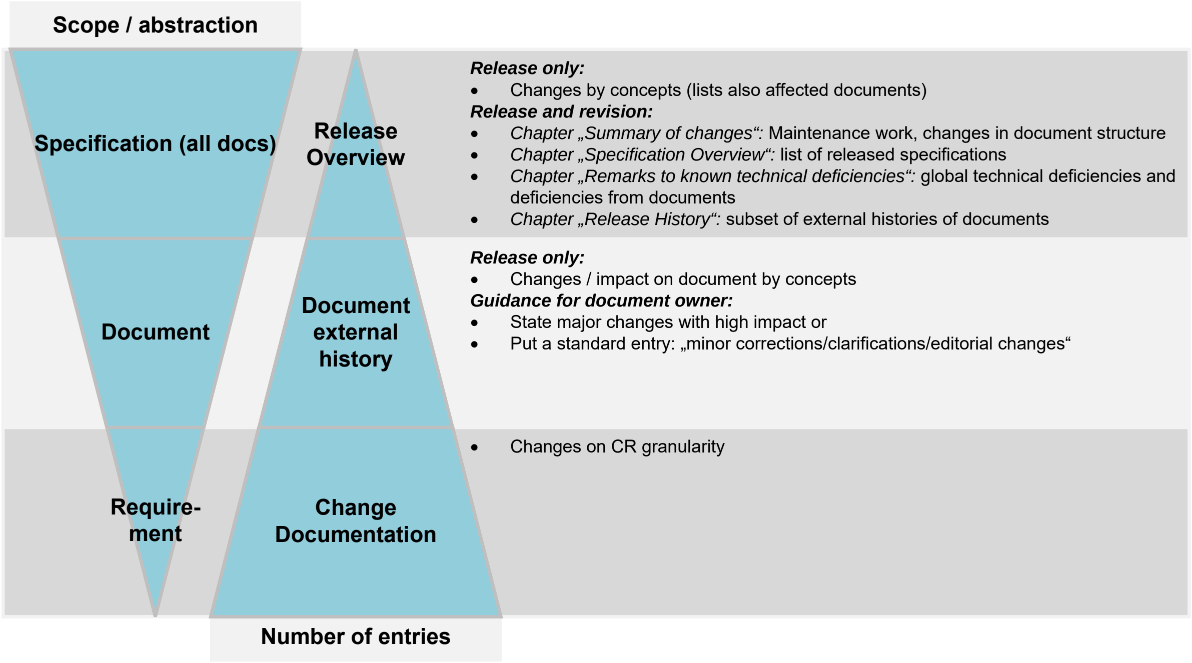
2.1.3 Release Documentation

There are no major changes in the Release Documentation.

2.2 History information in AUTOSAR

The following diagram shows the location of documentation of changes.

The Change Documentation is also available for Adaptive Platform since release R20-11.



Release only:
- Changes by concepts (lists also affected documents)

Release and revision:
- *Chapter „Summary of changes“:* Maintenance work, changes in document structure
- *Chapter „Specification Overview“:* list of released specifications
- *Chapter „Remarks to known technical deficiencies“:* global technical deficiencies and deficiencies from documents
- *Chapter „Release History“:* subset of external histories of documents

Release only:
- Changes / impact on document by concepts

Guidance for document owner:
- State major changes with high impact or
- Put a standard entry: „minor corrections/clarifications/editorial changes“

- Changes on CR granularity

Figure 2.1: History information in AUTOSAR

3 Specification overview

The published specifications are divided into the clusters

- Release Documentation
- Communication
- Memory
- System Services
- MCAL
- IO
- Libraries
- Diagnostics
- Safety
- BSW General
- General
- Methodology and Templates
- Mode Management
- RTE
- Application Interfaces
- Crypto
- Global Time
- SWArch
- Security

The assignment of the specifications to these clusters is shown below.

Long Name	File Name	Life cycle changes
Release Documentation		
Classic Platform Release Overview	AUTOSAR_TR_ClassicPlatformReleaseOverview	
AUTOSAR Classic Platform Specification Hashes	AUTOSAR_TR_ClassicPlatformSpecificationHashes	
Communication		
General Specification of Transformers	AUTOSAR_ASWS_TransformerGeneral	
Requirements on BSW Modules for SAE J1939	AUTOSAR_SRS_SAEJ1939	
Requirements on Bus Mirroring	AUTOSAR_SRS_BusMirroring	
Requirements on CAN	AUTOSAR_SRS_CAN	
Requirements on Communication	AUTOSAR_SRS_COM	
Requirements on Ethernet Support in AUTOSAR	AUTOSAR_SRS_Ethernet	
Requirements on FlexRay	AUTOSAR_SRS_FlexRay	
Requirements on Gateway	AUTOSAR_SRS_Gateway	
Requirements on I-PDU Multiplexer	AUTOSAR_SRS_IPDUMultiplexer	
Requirements on LIN	AUTOSAR_SRS_LIN	
Requirements on Module XCP	AUTOSAR_SRS_XCP	
Requirements on Secure Onboard Communication	AUTOSAR_SRS_SecureOnboardCommunication	
Requirements on SPI Handler/Driver	AUTOSAR_SRS_SPIHandlerDriver	
Requirements on Transformer	AUTOSAR_SRS_Transformer	
Requirements on TTCAN	AUTOSAR_SRS_TTCAN	
Requirements on Vehicle-2-X Communication	AUTOSAR_SRS_V2XCommunication	
Specification of a Request Manager for SAE J1939	AUTOSAR_SWS_SAEJ1939RequestManager	
Specification of a Transport Layer for SAE J1939	AUTOSAR_SWS_SAEJ1939TransportLayer	
Specification of Bus Mirroring	AUTOSAR_SWS_BusMirroring	
Specification of CAN Driver	AUTOSAR_SWS_CANDriver	
Specification of CAN Interface	AUTOSAR_SWS_CANInterface	
Specification of CAN Network Management	AUTOSAR_SWS_CANNetworkManagement	
Specification of CAN State Manager	AUTOSAR_SWS_CANStateManager	
Specification of CAN Transceiver Driver	AUTOSAR_SWS_CANTransceiverDriver	
Specification of CAN Transport Layer	AUTOSAR_SWS_CANTransportLayer	
Specification of COM Based Transformer	AUTOSAR_SWS_COMBasedTransformer	
Specification of Communication	AUTOSAR_SWS_COM	
Specification of Diagnostic Log and Trace	AUTOSAR_SWS_DiagnosticLogAndTrace	
Specification of Diagnostic over IP	AUTOSAR_SWS_DiagnosticOverIP	
Specification of Ethernet Driver	AUTOSAR_SWS_EthernetDriver	
Specification of Ethernet Interface	AUTOSAR_SWS_EthernetInterface	
Specification of Ethernet State Manager	AUTOSAR_SWS_EthernetStateManager	

▽

△

Long Name	File Name	Life cycle changes
Specification of Ethernet Switch Driver	AUTOSAR_SWS_EthernetSwitchDriver	
Specification of Ethernet Transceiver Driver	AUTOSAR_SWS_EthernetTransceiverDriver	
Specification of FlexRay AUTOSAR Transport Layer	AUTOSAR_SWS_FlexRayARTransportLayer	
Specification of FlexRay Driver	AUTOSAR_SWS_FlexRayDriver	
Specification of FlexRay Interface	AUTOSAR_SWS_FlexRayInterface	
Specification of FlexRay ISO Transport Layer	AUTOSAR_SWS_FlexRayISOTransportLayer	
Specification of FlexRay Network Management	AUTOSAR_SWS_FlexRayNetworkManagement	
Specification of FlexRay State Manager	AUTOSAR_SWS_FlexRayStateManager	
Specification of FlexRay Transceiver Driver	AUTOSAR_SWS_FlexRayTransceiverDriver	
Specification of I-PDU Multiplexer	AUTOSAR_SWS_IPDUMultiplexer	
Specification of Large Data COM	AUTOSAR_SWS_LargeDataCOM	
Specification of LIN Driver	AUTOSAR_SWS_LINDriver	
Specification of LIN Interface	AUTOSAR_SWS_LINInterface	
Specification of LIN State Manager	AUTOSAR_SWS_LINStateManager	
Specification of LIN Transceiver Driver	AUTOSAR_SWS_LINTransceiverDriver	
Specification of Module E2E Transformer	AUTOSAR_SWS_E2ETransformer	
Specification of Module XCP	AUTOSAR_SWS_XCP	
Specification of Network Management for SAE J1939	AUTOSAR_SWS_SAEJ1939NetworkManagement	
Specification of Network Management Interface	AUTOSAR_SWS_NetworkManagementInterface	
Specification of PDU Router	AUTOSAR_SWS_PDURouter	
Specification of Secure Onboard Communication	AUTOSAR_SWS_SecureOnboardCommunication	
Specification of Service Discovery	AUTOSAR_SWS_ServiceDiscovery	
Specification of Socket Adaptor	AUTOSAR_SWS_SocketAdaptor	
Specification of SOME/IP Transformer	AUTOSAR_SWS_SOMEIPTransformer	
Specification of SPI Handler/Driver	AUTOSAR_SWS_SPIHandlerDriver	
Specification of TCP/IP Stack	AUTOSAR_SWS_TcpIp	
Specification of TTCAN Driver	AUTOSAR_SWS_TTCANDriver	
Specification of TTCAN Interface	AUTOSAR_SWS_TTCANInterface	
Specification of UDP Network Management	AUTOSAR_SWS_UDPNetworkManagement	
Specification of Vehicle-2-X Basic Transport	AUTOSAR_SWS_V2XBasicTransport	
Specification of Vehicle-2-X Facilities	AUTOSAR_SWS_V2XFacilities	
Specification of Vehicle-2-X Geo Networking	AUTOSAR_SWS_V2XGeoNetworking	
Specification of Vehicle-2-X Management	AUTOSAR_SWS_V2XManagement	

▽

△

Long Name	File Name	Life cycle changes
Specification of Wireless Ethernet Driver	AUTOSAR_SWS_WirelessEthernetDriver	
Specification of Wireless Ethernet Transceiver Driver	AUTOSAR_SWS_WirelessEthernetTransceiverDriver	
Specification on SOME/IP Transport Protocol	AUTOSAR_SWS_SOMEIPTransportProtocol	
Memory		
Explanation of Firmware Over-The-Air	AUTOSAR_EXP_FirmwareOverTheAir	
NV Data Handling Guideline	AUTOSAR_EXP_NVDataHandling	
Requirements on EEPROM Driver	AUTOSAR_SRS_EEPROMDriver	
Requirements on Firmware Over-The-Air	AUTOSAR_RS_FirmwareOverTheAir	
Requirements on Flash Driver	AUTOSAR_SRS_FlashDriver	
Requirements on Flash Test	AUTOSAR_SRS_FlashTest	
Requirements on Memory Hardware Abstraction Layer	AUTOSAR_SRS_MemoryHWAbstractionLayer	
Requirements on Memory Services	AUTOSAR_SRS_MemoryServices	
Requirements on RAM Test	AUTOSAR_SRS_RAMTest	
Specification of EEPROM Abstraction	AUTOSAR_SWS_EEPROMAbstraction	
Specification of EEPROM Driver	AUTOSAR_SWS_EEPROMDriver	
Specification of Flash Driver	AUTOSAR_SWS_FlashDriver	
Specification of Flash EEPROM Emulation	AUTOSAR_SWS_FlashEEPROMEmulation	
Specification of Flash Test	AUTOSAR_SWS_FlashTest	
Specification of Memory Abstraction Interface	AUTOSAR_SWS_MemoryAbstractionInterface	
Specification of Memory Access	AUTOSAR_SWS_MemoryAccess	Initial release
Specification of Memory Driver	AUTOSAR_SWS_MemoryDriver	Initial release
Specification of Memory Mapping	AUTOSAR_SWS_MemoryMapping	
Specification of NVRAM Manager	AUTOSAR_SWS_NVRAMManager	
Specification of RAM Test	AUTOSAR_SWS_RAMTest	
System Services		
Explanation of CP Software Cluster Design And Integration Guideline	AUTOSAR_EXP_CPSwClusterDesignAndIntegrationGuideline	
Requirements on Free Running Timer	AUTOSAR_SRS_FreeRunningTimer	
Requirements on Function Inhibition Manager	AUTOSAR_SRS_FunctionInhibitionManager	
Requirements on Hardware Test Manager on start up and shutdown	AUTOSAR_SRS_HWTestManager	
Requirements on Operating System	AUTOSAR_SRS_OS	
Requirements on Software Cluster Connection module	AUTOSAR_SRS_SoftwareClusterConnection	
Requirements on Time Service	AUTOSAR_SRS_TimeService	
Specification and Integration of Hardware Test Management at start up and shutdown	AUTOSAR_TR_HWTestManagementIntegrationGuide	
Specification of Communication Manager	AUTOSAR_SWS_COMManager	
Specification of Default Error Tracer	AUTOSAR_SWS_DefaultErrorTracer	

▽

△

Long Name	File Name	Life cycle changes
Specification of Function Inhibition Manager	AUTOSAR_SWS_ FunctionInhibitionManager	
Specification of Hardware Test Manager on start up and shutdown	AUTOSAR_SWS_HWTestManager	
Specification of Operating System	AUTOSAR_SWS_OS	
Specification of Software Cluster Connection module	AUTOSAR_SWS_ SoftwareClusterConnection	
Specification of Time Service	AUTOSAR_SWS_TimeService	
MCAL		
General Requirements on SPAL	AUTOSAR_SRS_SPALGeneral	
Requirements on Core Test	AUTOSAR_SRS_CoreTest	
Requirements on GPT Driver	AUTOSAR_SRS_GPTDriver	
Requirements on MCU Driver	AUTOSAR_SRS_MCUDriver	
Specification of Core Test	AUTOSAR_SWS_CoreTest	
Specification of GPT Driver	AUTOSAR_SWS_GPTDriver	
Specification of MCU Driver	AUTOSAR_SWS_MCUDriver	
IO		
Requirements on ADC Driver	AUTOSAR_SRS_ADCDriver	
Requirements on DIO Driver	AUTOSAR_SRS_DIODriver	
Requirements on I/O Hardware Abstraction	AUTOSAR_SRS_IOHWAbstraction	
Requirements on ICU Driver	AUTOSAR_SRS_ICUDriver	
Requirements on OCU Driver	AUTOSAR_SRS_OCUDriver	
Requirements on Port Driver	AUTOSAR_SRS_PortDriver	
Requirements on PWM Driver	AUTOSAR_SRS_PWMDriver	
Specification of ADC Driver	AUTOSAR_SWS_ADCDriver	
Specification of DIO Driver	AUTOSAR_SWS_DIODriver	
Specification of I/O Hardware Abstraction	AUTOSAR_SWS_ IOHardwareAbstraction	
Specification of ICU Driver	AUTOSAR_SWS_ICUDriver	
Specification of OCU Driver	AUTOSAR_SWS_OCUDriver	
Specification of Port Driver	AUTOSAR_SWS_PortDriver	
Specification of PWM Driver	AUTOSAR_SWS_PWMDriver	
Libraries		
Macro Encapsulation of Interpolation Calls	AUTOSAR_EXP_MacroEncapsulationofInterpolationCalls	
Requirements on Libraries	AUTOSAR_SRS_Libraries	
Specification of Basic Software Multicore Library	AUTOSAR_SWS_ BSWMulticoreLibrary	
Specification of Bit Handling Routines	AUTOSAR_SWS_BFXLibrary	
Specification of CRC Routines	AUTOSAR_SWS_CRCLibrary	
Specification of Extended Fixed Point Routines	AUTOSAR_SWS_EFXLibrary	
Specification of Fixed Point Interpolation Routines	AUTOSAR_SWS_IFXLibrary	
Specification of Fixed Point Math Routines	AUTOSAR_SWS_MFXLibrary	
Specification of Floating Point Interpolation Routines	AUTOSAR_SWS_IFLLibrary	

▽

△

Long Name	File Name	Life cycle changes
Specification of Floating Point Math Routines	AUTOSAR_SWS_MFLLibrary	
Specification of SW-C End-to-End Communication Protection Library	AUTOSAR_SWS_E2ELibrary	
Diagnostics		
Specification of a Diagnostic Communication Manager for SAE J1939	AUTOSAR_SWS_ SAEJ1939DiagnosticCommunicationManager	
Specification of Diagnostic Communication Manager	AUTOSAR_SWS_ DiagnosticCommunicationManager	
Specification of Diagnostic Event Manager	AUTOSAR_SWS_ DiagnosticEventManager	
Safety		
Overview of Functional Safety Measures in AUTOSAR	AUTOSAR_EXP_ FunctionalSafetyMeasures	
Requirements on Safety Extensions	AUTOSAR_RS_SafetyExtensions	obsolete
Requirements on Watchdog Driver	AUTOSAR_SRS_WatchdogDriver	
Safety Use Case Example	AUTOSAR_EXP_SafetyUseCase	
Specification of Safety Extensions	AUTOSAR_TPS_SafetyExtensions	obsolete
Specification of Watchdog Driver	AUTOSAR_SWS_WatchdogDriver	
Specification of Watchdog Interface	AUTOSAR_SWS_WatchdogInterface	
Specification of Watchdog Manager	AUTOSAR_SWS_WatchdogManager	
BSW General		
Basic Software UML Model	AUTOSAR_MOD_BSWUMLModel	
Complex Driver design and integration guideline	AUTOSAR_EXP_ CDDDesignAndIntegrationGuideline	
Description of the AUTOSAR standard errors	AUTOSAR_EXP_ErrorDescription	
Explanation of Error Handling on Application Level	AUTOSAR_EXP_ ApplicationLevelErrorHandling	
Explanation of Interrupt Handling within AUTOSAR	AUTOSAR_EXP_ InterruptHandlingExplanation	
General Requirements on Basic Software Modules	AUTOSAR_SRS_BSWGeneral	
General Specification of Basic Software Modules	AUTOSAR_SWS_BSWGeneral	
Guide to BSW Distribution	AUTOSAR_EXP_ BSWDistributionGuide	
List of Basic Software Modules	AUTOSAR_TR_BSWModuleList	
Modeling Guidelines of Basic Software EA UML Model	AUTOSAR_TR_ BSWUMLModelModelingGuide	
Specification of Communication Stack Types	AUTOSAR_SWS_ CommunicationStackTypes	
Specification of Compiler Abstraction	AUTOSAR_SWS_CompilerAbstraction	obsolete
Specification of Platform Types	AUTOSAR_SWS_PlatformTypes	
Specification of Standard Types	AUTOSAR_SWS_StandardTypes	
General		
Explanation of Classic Platform Software Architectural Decisions	AUTOSAR_EXP_ CPSWArchitecturalDecisions	Initial release
Layered Software Architecture	AUTOSAR_EXP_ LayeredSoftwareArchitecture	

▽

△

Long Name	File Name	Life cycle changes
Requirements on AUTOSAR Features	AUTOSAR_RS_Features	obsolete
Specification of Bulk NvData Manager	AUTOSAR_SWS_BulkNvDataManager	
Virtual Functional Bus	AUTOSAR_EXP_VFB	
Methodology and Templates		
Basic Software Module Description Template	AUTOSAR_TPS_ BSWModuleDescriptionTemplate	
Diagnostic Extract Template	AUTOSAR_TPS_ DiagnosticExtractTemplate	
Integration of Franca IDL Software Component Descriptions	AUTOSAR_TR_FrancaIntegration	
Methodology for Classic Platform	AUTOSAR_TR_Methodology	
Modeling Show Cases Examples	AUTOSAR_EXP_ModelingShowCases	
Modeling Show Cases Report	AUTOSAR_TR_ModelingShowCases	
Requirements on Basic Software Module Description Template	AUTOSAR_RS_ BSWModuleDescriptionTemplate	
Requirements on Diagnostic Extract Template	AUTOSAR_RS_ DiagnosticExtractTemplate	
Requirements on ECU Configuration	AUTOSAR_RS_ECUConfiguration	
Requirements on ECU Resource Template	AUTOSAR_RS_ ECUResourceTemplate	
Requirements on Software Component Template	AUTOSAR_RS_ SoftwareComponentTemplate	
Requirements on System Template	AUTOSAR_RS_SystemTemplate	
Software Component Template	AUTOSAR_TPS_ SoftwareComponentTemplate	
Specification of ECU Configuration	AUTOSAR_TPS_ECUConfiguration	
Specification of ECU Configuration Parameters (XML)	AUTOSAR_MOD_ ECUConfigurationParameters	
Specification of ECU Resource Template	AUTOSAR_TPS_ ECUResourceTemplate	
Specification of Timing Extensions	AUTOSAR_TPS_TimingExtensions	
Supplementary material of general blueprints for AUTOSAR	AUTOSAR_TR_ GeneralBlueprintsSupplement	
System Template	AUTOSAR_TPS_SystemTemplate	
Mode Management		
Guide to Mode Management	AUTOSAR_EXP_ ModeManagementGuide	
Requirements on Mode Management	AUTOSAR_SRS_ModeManagement	
Specification of Basic Software Mode Manager	AUTOSAR_SWS_BSWModeManager	
Specification of ECU State Manager	AUTOSAR_SWS_ECUStateManager	
RTE		
Requirements on Runtime Environment	AUTOSAR_SRS_RTE	
Specification of RTE Software	AUTOSAR_SWS_RTE	
Application Interfaces		
Application Design Patterns Catalogue	AUTOSAR_TR_ AIDesignPatternsCatalogue	
Application Interface Examples	AUTOSAR_MOD_ AISpecificationExamples	

▽

△

Long Name	File Name	Life cycle changes
Application Interfaces User Guide	AUTOSAR_EXP_AIUserGuide	back from obsolete to published
Explanation of Application Interface of AD/ADAS vehicle motion control	AUTOSAR_EXP_AIADASAndVMC	
Explanation of Application Interfaces of Occupant and Pedestrian Safety Systems Domain	AUTOSAR_EXP_AIOccupantAndPedestrianSafety	
Explanation of Application Interfaces of the Body and Comfort Domain	AUTOSAR_EXP_AIBodyAndComfort	
Explanation of Application Interfaces of the Chassis Domain	AUTOSAR_EXP_AIChassis	
Explanation of Application Interfaces of the HMI, Multimedia and Telematics Domain	AUTOSAR_EXP_AIHMIMultimediaAndTelematics	
Explanation of Application Interfaces of the Powertrain Engine Domain	AUTOSAR_EXP_AIPowertrain	
Requirements on SW-C and System Modeling	AUTOSAR_RS_SWCModeling	
SW-C and System Modeling Guide	AUTOSAR_TR_SWCModelingGuide	
Unique Names for Documentation, Measurement and Calibration: Modeling and Naming Aspects including Automatic Generation	AUTOSAR_TR_AIMeasurementCalibrationDiagnostics	
XML Specification of Application Interfaces	AUTOSAR_MOD_AISpecification	
Crypto		
Requirements on Crypto Stack	AUTOSAR_SRS_CryptoStack	
Specification of Crypto Driver	AUTOSAR_SWS_CryptoDriver	
Specification of Crypto Interface	AUTOSAR_SWS_CryptoInterface	
Specification of Crypto Service Manager	AUTOSAR_SWS_CryptoServiceManager	
Specification of Key Manager	AUTOSAR_SWS_KeyManager	
Utilization of Crypto Services	AUTOSAR_EXP_UtilizationOfCryptoServices	
Global Time		
Specification of Synchronized Time-Base Manager	AUTOSAR_SWS_SynchronizedTimeBaseManager	
Specification of Time Synchronization over CAN	AUTOSAR_SWS_TimeSyncOverCAN	
Specification of Time Synchronization over Ethernet	AUTOSAR_SWS_TimeSyncOverEthernet	
Specification of Time Synchronization over FlexRay	AUTOSAR_SWS_TimeSyncOverFlexRay	
SWArch		
Explanatory Document for usage of AUTOSAR RunTimeInterface	AUTOSAR_EXP_ClassicPlatformARTI	
Requirements on Debugging, Tracing and Profiling support of AUTOSAR Components	AUTOSAR_RS_ClassicPlatformDebugTraceProfile	
Specification of AUTOSAR Run-Time Interface	AUTOSAR_SWS_ClassicPlatformARTI	
Security		

▽

△

Long Name	File Name	Life cycle changes
Specification of Intrusion Detection System Manager	AUTOSAR_SWS_ IntrusionDetectionSystemManager	

Table 3.1: Specification Overview

4 Remarks to known technical deficiencies

The technical deficiencies per specification are - if applicable - mentioned inside the respective specification in a chapter "Known Limitations" located after the table of contents.

The following technical deficiencies are to be mentioned, where clicking on the section reference will bring you to the respective document:

Document UID	Long Name	Document Type	Section Reference
72	Specification of LIN Driver	SWS	4.1
73	Specification of LIN Interface	SWS	4.2
74	Specification of FlexRay Transceiver Driver	SWS	4.3
80	Specification of Watchdog Manager	SWS	4.4
253	Specification of CAN State Manager	SWS	4.5
254	Specification of FlexRay State Manager	SWS	4.6
313	Specification of Basic Software Mode Manager	SWS	4.7
351	Specification of Diagnostic Log and Trace	SWS	4.8
414	Specification of UDP Network Management	SWS	4.9
416	Specification of Socket Adaptor	SWS	4.10
425	Specification of a Transport Layer for SAE J1939	SWS	4.11
430	Specification of Ethernet Driver	SWS	4.12
431	Specification of Ethernet Transceiver Driver	SWS	4.13
610	Specification of a Diagnostic Communication Manager for SAE J1939	SWS	4.14
611	Specification of a Request Manager for SAE J1939	SWS	4.15
612	Specification of Network Management for SAE J1939	SWS	4.16
617	Specification of TCP/IP Stack	SWS	4.17
631	Guide to BSW Distribution	EXP	4.18
655	Specification of Large Data COM	SWS	4.19
798	Specification of Wireless Ethernet Driver	SWS	4.20
799	Specification of Wireless Ethernet Transceiver Driver	SWS	4.21
873	Specification of Bus Mirroring	SWS	4.22
974	Specification of Software Cluster Connection module	SWS	4.23
988	Explanation of Application Interface of AD/ADAS vehicle motion control	EXP	4.24

4.1 Specification of LIN Driver (UID 72, SWS)

Only one LIN channel of an ECU is allowed to connect to a particular LIN cluster. Unless there are unused (not connected) channels in the ECU, the number of LIN channels is equal to the number of LIN clusters.

Driver scope

- One LIN driver provides access to one LIN hardware unit type (simple UART or dedicated LIN hardware) that may consist of several LIN channels.

- For different LIN hardware units a separate LIN driver needs to be implemented. It is up to the implementer to adapt the driver to the different instances of similar LIN channels.
- In case several LIN driver instances (of same or different vendor) are implemented in one ECU the file names, API names, and published parameters must be modified such that no two definitions with the same name are generated. The name shall be extended according to [SRS_BSW_00347] with a Vendor Id (needed to distinguish LIN drivers from different vendors) and a Vendor specific name (needed to distinguish different hardware units implemented by one Vendor): <Module abbreviation>_<Vendor Id>_<Vendor specific name>.

The LIN Interface is responsible for calling the correct function. The necessary information shall be given in an XML file during configuration. See [1] for description how the LIN Interface handles several LIN drivers.

4.2 Specification of LIN Interface (UID 73, SWS)

The LIN Interface module (LinIf) supports ISO 17987 series of standards (which also covers ISO 14229-7:2015, SAE J2602 and the behavior of previous versions of LIN specifications: LIN 2.2, LIN 2.1, LIN 2.0 and LIN 1.3 by LIN Consortium as far as they are identical to ISO 17987, see Annex B of ISO 17987-3 for compatibility information), but with some limitations, e.g.

- Following services are not supported by the LinIf:
  - ConditionalChangeNAD (SID 0xB3, defined in the LIN 2.1 specification; obsolete in ISO 17987-3)
  - DataDump (SID 0xB4, optional in ISO 17987-3)
  - Transmission of Reserved Frames (defined in the LIN 2.1 specification)
- Following services are not supported by the LinIf, for LIN Slave Nodes:
  - ReadByIdentifier with identifier unequal to 0 and 2 (SID 0xB2, mandatory in ISO 17987-3)
  - the Serial Number (defined in the ISO 17987-3, clause 6.2.2). It means that there's no corresponding configuration nor API for accessing Serial Number
  - AutoAddressingSlave (SID 0xB8, optional in ISO 17987-3), Slave node position detection (SID 0xB5, optional in LIN 2.x specification)
- Following services are not supported by the LinIf, for LIN Master Nodes:
  - ReadByIdentifier (SID 0xB2, mandatory in ISO 17987-3)

The Specification of LIN Interface (SWS LinIf) defines the Behavior of LIN Schedule Table Manager which is not defined in the ISO 17987 series of standards. But if LinTpScheduleChangeDiag was set to TRUE, simultaneous Schedule Table Switch requests originated from LinTp and from Non-LinTp (BswM or CDD) must be avoided, to prevent premature terminations of diagnostic connections. This issue will be fixed in next release(s). The Specification of LIN Interface (SWS LinIf) defines the Behavior of LIN Schedule Table Manager which is not defined in the ISO 17987 series of standards. But if LinTpScheduleChangeDiag was set to TRUE, simultaneous Schedule Table Switch requests originated from LinTp and from Non-LinTp (BswM or CDD) must be avoided, to prevent premature terminations of diagnostic connections. This issue will be fixed in next release(s).

4.3 Specification of FlexRay Transceiver Driver (UID 74, SWS)

The FlexRay Transceiver must provide functionality and an interface, mapped to the operation mode model assumed for the AUTOSAR FlexRay Transceiver Driver. See 7.1 AUTOSAR FlexRay Transceiver Operation Modes.

- The FlexRay Transceiver Driver shall use the APIs of underlying DIO drivers synchronously.
- The FlexRay Transceiver Driver should use the APIs of underlying SPI drivers synchronously if possible and asynchronously where required.
- The FlexRay transceiver requires a LEVEL 2, Enhanced (Synchronous/Asynchronous) SPI Handler_Driver
- The FlexRay Transceiver Driver shall handle the transceiver-specific timing requirements internally.

The communication between the driver and the transceiver is performed via ports or SPI or both. If ports are used, applying values in a predefined sequence and with a given timing to the ports are used to communicate and change the hardware operation modes. These sequences and timings must be handled within the FlexRay Transceiver Driver.

4.4 Specification of Watchdog Manager (UID 80, SWS)

- There're many long-lasting limitations for the Watchdog Manager module (WdgM). For details, see chap. 4 of SWS WdgM.
- Uptraces: AUTOSAR Foundation documents related to Health Monitoring (which are to be applicable to both AP PHM and CP WdgM) are not referred yet.
- Behavior of MainFunciton-based Supervision Algorithms (Alive, Timeout part of Deadline) right after Mode Switch (incl. right after calling WdgM_Init) and resulting updates of the states (Local Supervision Status and Global Supervision Status) requires further clarification.

4.5 Specification of CAN State Manager (UID 253, SWS)

The CanSM module can be used for CAN communication only. Its task is to operate with the CanIf module to control one or multiple underlying CAN Controllers and CAN Transceiver Drivers. Other protocols than CAN (i.e. LIN or FlexRay) are not supported.

4.6 Specification of FlexRay State Manager (UID 254, SWS)

This specification only defines the straightforward case for starting and stopping the communication on a FlexRay cluster.

For the case of multiple CC of one ECU assigned to one FlexRay cluster some items are left open for the implementation:

- Which CC is used to transmit the wakeup pattern
- Handling of inconsistent POC states in the CCs

4.7 Specification of Basic Software Mode Manager (UID 313, SWS)

Maximum one instance of the BSW Mode Manager may be used within a partition.

4.8 Specification of Diagnostic Log and Trace (UID 351, SWS)

VFB Tracing: Currently, VFB Trace only supports the non-verbose mode. I.e., the Dlt module will send out the arguments in a raw format, simply doing a memory copy of the arguments to the trace message.

The Dlt data type model does NOT support arbitrarily nested complex data types, which AUTOSAR does. So there is no generic way to transform arguments given to the VFB Trace hook functions into Dlt data types needed for the verbose mode.

An ASAM Fibex description cannot be generated by the Dlt module as the in-memory representation might not be compliant to the SWCD data type description of the arguments.

Although Log and Trace Protocol version "2" (compare [2]) is already available, the Dlt module currently only supports version "1" of the Log and Trace Protocol [3].

4.9 Specification of UDP Network Management (UID 414, SWS)

- One instance of UdpNm is associated with only one NM-Cluster in one network. One NM-Cluster can have only one instance of UdpNm in one node.
- One instance of UdpNm is associated with only one network within the same ECU.
- UdpNm is only applicable for TCP/IP based systems.

The AUTOSAR UdpNm algorithm shall support up to 250 nodes per NM-Cluster by default.

Note: The AUTOSAR UdpNm algorithm can support an arbitrary number of nodes per NM-cluster (even more than default 250 nodes per cluster, if necessary) - it is only a matter of configuration, since the upper limit is not fixed and depends on the trade off between response time, fault-tolerance and resulted bus load configured for the AUTOSAR UdpNm coordination algorithm. This might depend on the physical layer used.

4.10 Specification of Socket Adaptor (UID 416, SWS)

The transmission of data using TCP/IP over Ethernet requires about 60 bytes of header information. This implies that for small messages the header overhead may reach an unacceptably high percentage.

To avoid further protocol overhead, the use of a single socket connection per PDU is described here. However, this solution is very resource intensive, particularly if many small PDUs are to be transmitted. One solution described here as an option is to add a small PDU header, containing an ID and length information. This enables transmission of multiple PDUs via one socket connection. Additionally, a resource conservation scheme is included in this specification as an option.

This document does not cover the assignment of UDP or TCP port numbers. There is no reserved space within the IANA assigned number range. Each implementer is responsible for managing the used port numbers.

This document does not cover the management of IP addresses. This might be done dynamically, e.g. by using DHCP, or statically. It is the implementers responsibility to prevent address conflicts and achieve compliance with IANA address assignments.

This specification does not prescribe a certain physical layer or data rate.

SOME/IP Protocol Specification [4] specifies to check Protocol Version prior to the check of Service ID and Method ID to be valid. SoAd does not check Protocol Version. Independent of future version It will always interpret the first 4 bytes as message ID (header ID in this document) and the second 4 bytes as payload length.

SOME/IP Protocol Specification [4] specifies to check Interface Version prior to the check of Method ID to be valid. SoAd does not check Interface Version. SoAd performs routing to configured Message IDs independent of the contained Interface Version on dedicated routing paths.

SOME/IP Protocol Specification [4] specifies to check Service ID and Method ID to be valid. Invalid IDs shall be responded with corresponding error codes.

SoAd checks the SOME/IP message ID (header ID in this document) which is a combination of service ID and method ID. If the ID is not valid the module is not capable to respond with corresponding error codes. Instead, runtime error SOAD_E_INV_PDUHEADER_ID is raised.

SOME/IP Protocol Specification [4] specifies with [PRS_SOMEIP_00535] that all Transport Protocol Bindings shall support transporting more than one SOME/IP message in a Transport Layer PDU. In case of TCP this requirement is out of SoAd scope since the message packing and transportation over a TCP stream can not be influenced by SoAd. In case of UDP collecting of SOME/IP message is limited to IF API only. According to [SWS_SoAd_00553] a SOME/IP message will be retrieved from upper layer via TP API and immediately sent over UDP. Packing of messages is not foreseen and not supported in this case.

4.11 Specification of a Transport Layer for SAE J1939 (UID 425, SWS)

The AUTOSAR architecture contains several communication system specific transport layers (J1939Tp, CanTp, FrTp, etc.). All of these modules need to have identical APIs, with the exception of API functions for which the PduR has separate configuration abilities.

The J1939Tp module does not implement the TriggerTransmit API, because it is only needed for time triggered bus architectures.

4.12 Specification of Ethernet Driver (UID 430, SWS)

The Ethernet Driver module is only able to handle a single thread of execution. The execution must not be preempted by itself.
It is not possible to transmit data which exceeds the available buffer size of the used controller. Longer data has to be transmitted using the Internet Protocol (IP) or Transmission Control Protocol (TCP).
Depending on the Ethernet hardware, it may become necessary that implementations deviate from API specifications in respect to the asynchronous/synchronous behavior.

4.13 Specification of Ethernet Transceiver Driver (UID 431, SWS)

The Ethernet Transceiver Driver module is only able to handle a single thread of execution. The execution must not be preempted by itself.

4.14 Specification of a Diagnostic Communication Manager for SAE J1939 (UID 610, SWS)

The J1939 Diagnostic Communication Manager implements only a subset of ’Diagnostic messages’ as defined in Table 1: Supported DMx messages.

The DM13 does not support “Suspend Signal” “Suspend Duration”.

NACK is not provided for received DMx messages that are not supported or not configured. This restriction mainly affects handling of DM07 and DM13.

4.15 Specification of a Request Manager for SAE J1939 (UID 611, SWS)

The J1939 Request Manager only implements Request, Request2, and Acknowledgement PGs. It does not provide support for the Transfer PG.

4.16 Specification of Network Management for SAE J1939 (UID 612, SWS)

The J1939 Network Management module does not support all features defined in [5], especially:

- Changing the address of a node after reception of CommandedAddress or after an address loss.
- Changing the NAME of a node using the Name Management protocol.
- Detection of address violations by messages other than AddressClaimed.

4.17 Specification of TCP/IP Stack (UID 617, SWS)

This document does not cover the assignment of UDP or TCP port numbers. There is no reserved space within the IANA assigned number range. Each implementer is responsible for managing the used port numbers.

This document does not cover the management of IP addresses. This might be done dynamically, e.g. by using DHCP, or statically. It is the implementer's responsibility to prevent address conflicts and achieve compliance with IANA address assignments.

This specification does not prescribe a certain physical layer or data rate.

Although a CDD interface is specified, allowing additional upper layer modules, a fan-out of one socket to multiple upper layer modules is not intended to be supported.

The AUTOSAR TLS implementation has the following limitations:

- A TLS implementation shall not support data compression or decompression.
- Session renegotiation shall not be supported.
- No support for secure connection over UDP (e.g. for DTLS)
- No support of FQDN
- No client Hello padding extension IETF RFC7685
- No session hash and extended master secret IETF RFC 7627
- No support for TLS versions lower than 1.2.
- No support for dynamic “downgrading” of a TCP connection with an established TLS connection to a plain TCP connection (without TLS)
- Static TLS connection assignment is bound to the port configuration of the server. Thus, using different TLS settings for different connections (possibly originating from different clients) to the same server port is not possible.

The AUTOSAR IPsec implementation has the following limitations:

- IPsec in “tunnel mode” is not supported right now. Transport mode only.
- IPv6 is not supported.
- Multicast is not supported.

4.18 Guide to BSW Distribution (UID 631, EXP)

The support for Basic Software Allocation in AUTOSAR is currently limited to backward compatible changes (w.r.t. AUTOSAR 4.0.3). This currently results in the following restrictions, which may not apply to future releases of AUTOSAR:

- Communication between master and satellites is not standardized.
- BSW functional clusters and their AUTOSAR BSW Cluster Interface are not standardized.

Since the concept "Memory Stack Rework" 2.1.1.1.5 is still draft, this also applies to the properties of the Mem driver mentioned in subsection 2.5.8.

4.19 Specification of Large Data COM (UID 655, SWS)

Large data COM supports communication of linear opaque byte wise data in a very resource-saving way. It does so by skipping all functionality not required for event based non-cyclic communication. Large data COM does not apply any changes like for instance endianness conversion to the data it transports. Prerequisites for usage of Efficient COM:

- PDU contains only 1 Signal and no ISignalGroup
- The Signal is of type byte array with either fixed or dynamic length
- Transmission mode is either triggered or triggered without repetition
- Transmission mode selection is not used
- No update bit is used
- No minimum delay time is used
- No timeout supervision is used
- No byte order conversion is used
- No Rx/Tx Filtering
- No Signal Invalidation
- No TP Fan-out

4.20 Specification of Wireless Ethernet Driver (UID 798, SWS)

- It is not possible to transmit data which exceeds the available buffer size of the used controller.
- AUTOSAR supports currently only wireless communication using IEEE 802.11p. Other 802.11 standards (e.g. for infrastructure networks and integration with TCP/IP) can be extended in future releases of the AUTOSAR standard.
- The V2X modules follow the guidance regarding the Day-1 scenarios defined by the Basic System Standards Profile from Car-2-Car-Consortium.
- AUTOSAR R20-11 only focuses on the European version of car-to-car communication as defined by ETSI. Extension to other regions are planned for future releases of the AUTOSAR standard.
- The Microcontroller Abstraction Layer Multi-Core Distribution Concept is implemented as “draft” in this software specification. Refer to chapter 10 for more information.

4.21 Specification of Wireless Ethernet Transceiver Driver (UID 799, SWS)

The Microcontroller Abstraction Layer Multi-Core Distribution Concept is implemented as “draft” in this software specification. Refer to chapter 10 for more information.

4.22 Specification of Bus Mirroring (UID 873, SWS)

The Bus Mirroring module cannot be used to influence the traffic on one of the buses configured as a source bus. To ensure this and to avoid loop-back of messages leading to bus overload, the generation tool shall ensure that no bus is connected to the Bus Mirroring module both as source and destination bus (see [SWS_Mirror_00001]).

The Bus Mirroring module is controlled by a diagnostic control application through the dedicated (service) API listed in chapter 8. The control functionality is made accessible to a diagnostic tester by special diagnostic services, which are handled by the DCM and implemented by the diagnostic control application. The DCM provides the necessary security to exclude inadvertent activation of the Bus Mirroring. The Bus Mirroring module does not provide another control interface, and it does not receive control messages on the destination bus.

In general, the Bus Mirroring module does not support source buses that have a larger frame size or more additional information than the destination bus can carry, e.g. CAN-FD to CAN, CAN to LIN, FlexRay to CAN, Ethernet to CAN, or Ethernet to FlexRay. The Bus Mirroring module does not fragment mirrored frames.

The Bus Mirroring module will only mirror traffic that is actually received or transmitted by the bus interface modules. For CAN this means that besides the transmitted frames only those data frames that pass the hardware filter will be mirrored, and that remote frames and error frames will not be mirrored. For LIN, slave-to-slave communication will not be mirrored by a LIN master. And for FlexRay, only transmitted frames and those received frames for which reception buffers are assigned (possibly as a FIFO) will be mirrored.

Another limitation of the mirroring from a FlexRay source bus concerns the reported time stamps and cycles. The Timestamp reported for a FlexRay frame contains the time when the corresponding job list entry was executed. The actual transmission time has to be calculated from the slot ID contained in the reported FrameID. The cycle contained in the reported FrameID is accurate only for received frames and frames transmitted in the static segment. For frames transmitted in the dynamic segment, the reported cycle can be inaccurate because it can happen that a frame cannot be transmitted in the expected cycle, it is then deferred to the next suitable cycle.

A re-serialization of received serialized frames shall not be done by the Bus Mirroring module, because that would require too much resources. Instead, the serialized PDUs shall be routed directly to the destination bus.

The Bus Mirroring module will also not support the forwarding from Ethernet to Ethernet. This use case is already covered by the Port Mirroring feature of the AUTOSAR Ethernet Switch Driver.

4.23 Specification of Software Cluster Connection module (UID 974, SWS)

The specification currently supports a limited number of BSW modules. Furthermore, for the supported BSW modules, the functionality and APIs available for Applicative Software Clusters is only subset of the overall functionality. In addition, the available VFB communication features are restricted.

4.24 Explanation of Application Interface of AD/ADAS vehicle motion control (UID 988, EXP)

The logical architectures proposed do not restrict the development or products of companies or organizations participating in AUTOSAR.

5 Release history

5.1 Release R21-11

Name	Specification history entry
Classic Platform Release Overview	• Release Life Cycle Status: R21-11 is in Evolution, R21-11 supersedes R20-11
Application Design Patterns Catalogue	• Editorial changes
Application Interfaces User Guide	• No content changes
Basic Software Module Description Template	• Modified compiler abstraction • Minor corrections • Editorial changes
Complex Driver design and integration guideline	• No content changes
Description of the AUTOSAR standard errors	• No content changes
Diagnostic Extract Template	• Support for Authentication service • Support for CpSoftwareCluster • minor corrections / clarifications / editorial changes
Explanation of Application Interface of AD/ADAS vehicle motion control	• Add Curvature request interface • Add interface list as appendix • Expand ADAS function scope: ACC to include standstill, LKA to remove speed limit
Explanation of Application Interfaces of Occupant and Pedestrian Safety Systems Domain	• No content changes
Explanation of Application Interfaces of the Body and Comfort Domain	• No content changes
Explanation of Application Interfaces of the Chassis Domain	• No content changes
Explanation of Application Interfaces of the HMI, Multimedia and Telematics Domain	• No content changes
Explanation of Application Interfaces of the Powertrain Engine Domain	• No content changes
Explanation of Classic Platform Software Architectural Decisions	• Initial release
Explanation of CP Software Cluster Design And Integration Guideline	• Minor corrections / clarifications / editorial changes
Explanation of Error Handling on Application Level	• No content changes
Explanation of Firmware Over-The-Air	• No content changes
Explanation of Interrupt Handling within AUTOSAR	• No content changes
Explanatory Document for usage of AUTOSAR RunTimeInterface	• Added examples showing static debugging, CAT1 interrupts, and VFB Hooks • Minor corrections and updates
General Requirements on Basic Software Modules	• Allowance of function pointers (SRS_BSW_00371) • Exclusive use of AUTOSAR data types (SRS_BSW_00304) • Introduction of C99 standard in AUTOSAR (SRS_BSW_00477)
General Requirements on SPAL	• No content changes

▽

△

Name	Specification history entry
General Specification of Basic Software Modules	• Introduce C99 standard in AUTOSARMinor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation • Changed Document Status from Final to published
General Specification of Transformers	• Clarification of APIs defined as "Synchronous /Asynchronous" • Contradiction solved in SWS_Xfrm_00108
Guide to BSW Distribution	• Clarified partition scope of MCAL • Removed restriction for BSW partitions per core
Guide to Mode Management	• Added chapter on PduR routing path group switching • Editorial Changes
Integration of Franca IDL Software Component Descriptions	• No content changes
Layered Software Architecture	• Incorporated draft concept for new Memory Driver and Memory Access
List of Basic Software Modules	• Added AUTOSAR Run-Time Interface • Added Memory Driver, Memory Access
Macro Encapsulation of Interpolation Calls	• No content changes
Methodology for Classic Platform	• Minor corrections and editorial changes • changed document title to: Methodology for Classic Platform
Modeling Guidelines of Basic Software EA UML Model	• redesigned modeling of Generic Interfaces • redesigned modeling of Virtual Interfaces • described modeling of BSW Module Extensions • described modeling of union datatypes and function pointer datatypes
Modeling Show Cases Report	• No content changes
NV Data Handling Guideline	• No content changes
Overview of Functional Safety Measures in AUTOSAR	• No content changes
Requirements on ADC Driver	• No content changes
Requirements on AUTOSAR Features	• Updated to C9
Requirements on Basic Software Module Description Template	• No content changes
Requirements on BSW Modules for SAE J1939	• Converted to LATEX • Extended acronyms / abbreviations and references • Improved linking of terms
Requirements on Bus Mirroring	• Added missing requirement for CAN ID and LIN PID mapping • Added detailed change history
Requirements on CAN	• Added requirement for hardware time stamping • Editorial changes
Requirements on Communication	• Removed I-PDU counter and I-PDU replication • Added support for independent development of CP Software Clusters
Requirements on Core Test	• No content changes
Requirements on Crypto Stack	• Editorial changes

▽

△

Name	Specification history entry
Requirements on Debugging, Tracing and Profiling support of AUTOSAR Components	• Added functional requirements on Tracing • Removed some superfluous requirements • Adjusted spec items to module name
Requirements on Diagnostic Extract Template	• No content changes
Requirements on DIO Driver	• No content changes
Requirements on ECU Configuration	• No content changes
Requirements on ECU Resource Template	• Editorial changes
Requirements on EEPROM Driver	• No content changes
Requirements on Ethernet Support in AUTOSAR	• Validation of 10BASE-T1S • Validation of Ethernet Wake on data line
Requirements on Firmware Over-The-Air	• No content changes
Requirements on Flash Driver	• No content changes
Requirements on Flash Test	• No content changes
Requirements on FlexRay	• No content changes
Requirements on Free Running Timer	• No content changes
Requirements on Function Inhibition Manager	• No content changes
Requirements on Gateway	• No content changes
Requirements on GPT Driver	• No content changes.
Requirements on Hardware Test Manager on start up and shutdown	• No content changes
Requirements on I/O Hardware Abstraction	• No content changes
Requirements on ICU Driver	• No content changes
Requirements on I-PDU Multiplexer	• No content changes
Requirements on Libraries	• No content changes
Requirements on LIN	• No content changes
Requirements on MCU Driver	• No content changes
Requirements on Memory Hardware Abstraction Layer	• Added MemAcc and Mem related requirements (SRS_MemHwAb_14033 to SRS_MemHwAb_14056) due to Memory stack rework
Requirements on Memory Services	• No content changes
Requirements on Mode Management	• Removed Draft from SRS_ModeMgm_09266 and SRS_ModeMgm_09268 • Removed SRS_ModeMgm_09252 (BswM shall be able to directly request communication modes for the available Partial Networks) • SRS_ModeMgm_09249 (PNC gateway and coordination functionality) • Editorial Changes
Requirements on Module XCP	• No content changes
Requirements on OCU Driver	• No content changes
Requirements on Operating System	• Updated traceability
Requirements on Port Driver	• Editorial Changes
Requirements on PWM Driver	• No content changes
Requirements on RAM Test	• No content changes
Requirements on Runtime Environment	• No content changes
Requirements on Safety Extensions	• Changed Document Status from published to obsolete

▽

△

Name	Specification history entry
Requirements on Secure Onboard Communication	• No content changes
Requirements on Software Cluster Connection module	• Support of Signal and SOME/IP based communication • Support of Diagnostic Services
Requirements on Software Component Template	• Uptrace to RS_Main_00050 corrected. • Editorial changes.
Requirements on SPI Handler/Driver	• Modified requirements: SRS_Spi_12197, SRS_Spi_12256
Requirements on SW-C and System Modeling	• No content changes
Requirements on System Template	• No content changes
Requirements on Time Service	• No content changes
Requirements on Transformer	• No content changes
Requirements on TTCAN	• Editorial changes
Requirements on Vehicle-2-X Communication	• No content changes
Requirements on Watchdog Driver	• No content changes
Safety Use Case Example	• No content changes
Software Component Template	• Introduction of imposition times for constraints • Clean-up of diagnostics service needs • New writing strategy for NvRAM • minor corrections / clarifications / editorial changes
Specification and Integration of Hardware Test Management at start up and shutdown	• No content changes
Specification of a Diagnostic Communication Manager for SAE J1939	• Fixed description of return values
Specification of a Request Manager for SAE J1939	• Converted to LATEX • Fixed UserId parameter range • Extended acronym and related documents tables • Improved linking of terms
Specification of a Transport Layer for SAE J1939	• No content changes
Specification of ADC Driver	• SWS_ADC_00338 modified
Specification of AUTOSAR Run-Time Interface	• Introduced SWS items into specification • Overall review and clarification • ARTI introduced as BSW Module "Arti" • New ARTI API and Errors
Specification of Basic Software Mode Manager	• Adding new mode request for SoAd (BswM_SoAd_SoConModeChg) • Setting BswM_WdgM_RequestPartitionReset and its configuration to obsolete • Editorial Changes
Specification of Basic Software Multicore Library	• No content changes
Specification of Bit Handling Routines	• No content changes (only converted to LaTex) • Artifact inclusion based on ArtifactAnalysis corrected
Specification of Bulk NvData Manager	• No content changes
Specification of Bus Mirroring	• Added detailed change history

▽

△

Name	Specification history entry
Specification of CAN Driver	• Timestamp requirements were added • Removed SWS_Can_00485 and ECUC_Can_00466 • Changed the scope of CanIndex from local to ECU global • Minor corrections / clarifications / editorial changes;
Specification of CAN Interface	• Support hardware-based timestamping • Minor corrections/clarifications • Editorial changes
Specification of CAN Network Management	• Rework of Partial Networking • Fixes for Partial Networking extensions
Specification of CAN State Manager	• Note added for CanSM_TransceiverModeIndication() • Communication mode notification to ComM after initialization clarified • Clean-up in CANSM_BSM regarding REPEAT_MAX / No Never-Give-Up Strategy
Specification of CAN Transceiver Driver	• Updated state machine behavior for CanTrcv_Init • Editorial changes
Specification of CAN Transport Layer	• Improve Error handling • Clarifications
Specification of COM Based Transformer	• Updated buffer reservation in transformer chain
Specification of Communication	• Removed I-PDU counter and I-PDU replication • Added support for independent development of CP Software Clusters • minor corrections / clarifications / editorial changes
Specification of Communication Manager	• Introduced dedicated APIs to synchronize the PNC status with Nm and set the usage of ComSignals to obsolete • Introcuded ComMChannelPerTxOnlyPnc to support transmission-only PNCs • Set requirements to valid which relates to forward an wake up request if an PNC is actively requested • Re-worked the service interfaces to support the Pn learning phase
Specification of Communication Stack Types	• Added CbkHandleIdType in Type definitions
Specification of Compiler Abstraction	• Deleted SWS_COMPILER_00051 • Marked specification as obsolete
Specification of Core Test	• Artifact inclusion based on ArtifactAnalysis corrected.
Specification of CRC Routines	• Minor corrections / clarifications / editorial changes • Changed Document Status from Final to published
Specification of Crypto Driver	• Clarification of Sync/Async for APIs • Clarification of key format description for SHE-Keys • Clarification about key state after Crypto_KeyElementSet() API. • Input and Output be optional for AEAD encrypt and decrypt in update mode

▽

△

Name	Specification history entry
Specification of Crypto Interface	• CryIf_KeyGenerate() and CryIf_RandomSeed() are always synchronous
Specification of Crypto Service Manager	• Harmonize definition of CRYPTO_ALGOMODE between CryptoDrv and Csm • Added key format description in CSM/Crypto Driver for SHE-keys • Added Clarification on seeding and generation of random numbers in the crypto stack • removed superfluous parameter keyId in CsmJobXXX interface operations • Editorial changes
Specification of Default Error Tracer	• Inconsistency between SWS_Det_00024 and SWS_Det_00009 solved. Also SWS_Det_00208 adapted. • Clarification of APIs defined as "Synchronous /Asynchronous" (Det_ReportError) • Editorial change (converted to LaTex)
Specification of Diagnostic Communication Manager	• Incorporation of Concept 670 Classic Platform Flexibility • Separated SAE J2012_4 DTCs and UDS DTCs • OBD on UDS shall be supported • Removed the Mirror Memory following ISO 14229-1:2020 • Support subfunctions 1A and 56 for UDS services 0x19
Specification of Diagnostic Event Manager	• OBD on UDS supported • SAE J2012_4 DTCs and UDS DTCs seperated • Add Pending / Busy in Dem_GetDTCSuppression / Dem_SetDTCSuppression • Removal of the Mirror Memory following ISO 14229-1:2020 • Support subfunctions 1A and 56 for UDS services 0x19 • Limit OBD DTCs to primary fault memory
Specification of Diagnostic Log and Trace	• Bugfixes and corrections • Editorial changes
Specification of Diagnostic over IP	• Most APIs reporting development errors no longer return with E_NOT_OK • Removed obsolete elements • Editorial changes
Specification of DIO Driver	• Cleaned return codes
Specification of ECU Configuration	• Extend EcucParameterDefs with symbolicNameValue to support PublishedInformation. • Added withAuto support to EcucAbstractReferenceDef.
Specification of ECU Resource Template	• Editorial changes

▽

△

Name	Specification history entry
Specification of ECU State Manager	• Updates on wakeup handling (ethernet wakeup) • Updates on error handling • Minor content changes, clarifications
Specification of EEPROM Abstraction	• Ea_SetMode() service is removed. • Ea_Cancel() service is now asynchronous. • Added support for buffer alignment for read and write operations. • Replaced Eep by MemAcc module as lower layer API interface to Ea.
Specification of EEPROM Driver	• Removed SWS_Eep_00047 • EepJobCallCycle renamed to EepMainFunctionPeriod and moved from EepInitConfiguration to EepGeneral
Specification of Ethernet Driver	• New runtime error and return code handling modified • Silent communication added • EthGetRxStatsApi added • Support SPI interface for external devices
Specification of Ethernet Interface	• Updates on 10BASE-T1S • EthernetWakeOnDataLine Specification items valid (no "Draft" tag) • Clarification on Return codes and error reporting • Updates on ReworkofPNCrelatedComM-andNMhandling COncept • Clarification on "Synchronous /Asynchronous" APIs • Removed section EthIfSwitchTimeStampIndication-Config • Removed SWS_EthIf_00248 • Updated uptraces of Security Event tables
Specification of Ethernet State Manager	• Update state machine behaviour in "ETHSM_STATE_ONLINE" and "ETHSM_STATE_WAIT_OFFLINE" • Editorial changes
Specification of Ethernet Switch Driver	• Added 10BASE-T1S support • Clarified return values and development errors • Removed EthSwtMgmtInfoIndicationTimeout related requirements
Specification of Ethernet Transceiver Driver	• New runtime error and return code handling modified • Detailing added in buffer handling • Editorial changes
Specification of Extended Fixed Point Routines	• Service ID field for specific API functions changed • Artifact inclusion based on ArtifactAnalysis corrected • Editorial change (converted to LaTex)

▽

△

Name	Specification history entry
Specification of Fixed Point Interpolation Routines	• No content changes (only converted to LaTex) • Artifact inclusion based on ArtifactAnalysis corrected
Specification of Fixed Point Math Routines	• Missing input parameter and return value description of Mfx_DivShLeft function (SWS_Mfx_00058) added • Editorial change (converted to LaTex)
Specification of Flash Driver	• Removed SWS_Fls_00109 • FlsCallCycle renamed to FlsMainFunctionPeriod and moved it from FlsConfigSet to FlsGeneral
Specification of Flash EEPROM Emulation	• Updated for new memory stack • Removed return codes for Det errors • Removed definitions of NVM functions
Specification of Flash Test	• SWS_FlsTst_00019 removed
Specification of FlexRay AUTOSAR Transport Layer	• No content changes
Specification of FlexRay Driver	• No content changes
Specification of FlexRay Interface	• Clarification on shortening of L-SduLength
Specification of FlexRay ISO Transport Layer	• No content changes
Specification of FlexRay Network Management	• Note added under SWS_FrNm_00492 • Changed occurrences of FRNM_PASSIVE_MODE_ENABLED to FrNmPassiveModeEnabled. • Changes to Synchronized PNC shutdown. • Uptrace from SRS_Nm changed to RS_Nm
Specification of FlexRay State Manager	• No content changes
Specification of FlexRay Transceiver Driver	• Header file cleanup
Specification of Floating Point Interpolation Routines	• Converted to LaTeX
Specification of Floating Point Math Routines	• New requirements SWS_Mfl_00843 and SWS_Mfl_00848 added for "Mfl_FloatToIntCvrt_f32" function. • Editorial changes.
Specification of Function Inhibition Manager	• SWS_Fim_CONSTR_0001 changed to SWS_Fim_CONSTR_00001
Specification of GPT Driver	• Update optional interfaces relative to EcuM
Specification of Hardware Test Manager	• No content changes
Specification of I/O Hardware Abstraction	• No content changes
Specification of ICU Driver	• Clean up of ICU_ConfigType related requirements regarding the data structure. • Correct sequence diagrams in chapter 9.4.
Specification of Intrusion Detection System Manager	• Added subcontainers for Timestamp and Signature • Additional Internal Security Events signals communication error • clarification of Internal Security Event • Replaced handdrawn sequence charts by generated
Specification of I-PDU Multiplexer	• Rework of assembly strategy of a ContainerIpdu that contain ContainedIpdus with Last-is-Best semantic • Shortening of IPDUs if possible • Avoid possible interpretation of queue bypass for containers

▽

△

Name	Specification history entry
Specification of Key Manager	• Editorial changes • Add upstream requirements
Specification of Large Data COM	• Introduced the support for "Software Clusters". Therefore extend the LdComUser configuration and introduced handle-id-based call-back functions • Minor corrections
Specification of LIN Driver	• Clarification of configuration parameter LinChannelWakeupSupport • Cleanup of Error classification chapter • Header file for EcuM_CheckWakeup changed
Specification of LIN Interface	• Added the API table of <User>_GotoSleepIndication • Removed inconsistent requirements regarding availability of LinIf_CheckWakeup and LinIf_WakeupConfirmation APIs
Specification of LIN State Manager	• Corrected Figure 7 and SWS_LinSM_00233
Specification of LIN Transceiver Driver	• Cleaned error codes
Specification of MCU Driver	• Removed SWS_Mcu_00131, SWS_Mcu_00054, SWS_Mcu_00035, SWS_Mcu_00030 and SWS_Mcu_00031 • Cleaned up unresolved references in traceability
Specification of Memory Abstraction Interface	• Improve the structure of the ’error sections’ • Cleanup diagrams in chapter 10
Specification of Memory Access	• Initial release
Specification of Memory Driver	• Initial release
Specification of Memory Mapping	• POWER_ON_INIT behaviour does not match ComputerRuntimeInitialization • Deprecate compiler abstraction • Description regarding alignment is too strict for some targets
Specification of Module E2E Transformer	• Added Concept 700 text and figures (E2E for fields) • Added Description of Profile 8m and 44m (E2E for methods)
Specification of Module XCP	• No content changes
Specification of Network Management for SAE J1939	• No content changes
Specification of Network Management Interface	• Several quality improvements • Removed chapter 10
Specification of NVRAM Manager	• Changes related to the concept 691 MemoryStackRework • Clarification regarding validation in NvM_WriteBlock • Migration from doc to latex
Specification of OCU Driver	• Removed the Ocu_ConfigType data structure specification

▽

△

Name	Specification history entry
Specification of Operating System	• Further updates to ARTI sections • API changes and clarifications (SetScheduleTableAsync, GetNumberOfActivatedCores) • New configuration options for placement of callouts. • Update of RES_SCHEDULER handling. • Minor correction / clarification / editorial changes
Specification of PDU Router	• Added multicast (1:n) support from a transport protocol module to local upper layer modules • Added fan-in (n:1) support for multiple communication interface modules to a local upper layer module • Cleaned up chapter 7 and clarified buffering concept • Same PduRRoutingPath may be assigned to multiple PduRRoutingPathGroups • Inter-Partition Gateway Routing Relations are described in more detail • Clarification and clean up of Multicast TP Tx PDU Forwarding • Editorial changes
Specification of Platform Types	• Editorial changes and clarifications. • Requirements tracing improved.
Specification of Port Driver	• Removed SWS_Port_00072 and SWS_Port_00073 and corresponding notes
Specification of PWM Driver	• SWS_Pwm_00061 removed as <Modul>_ConfigTypes are introduced as implementation specific and therefore additional requirements did not make sense anymore • Runtime error added on SWS_Pwm_00200
Specification of RAM Test	• Clarificaton on APIs defined as Synchronous / Asynchronous (RamTst_ErrorNotification, RamTst_TestCompletedNotification)
Specification of RTE Software	• Independent COM support for Software Clusters • New writing strategy for NvRAM • Minor corrections / clarifications / editorial changes
Specification of Safety Extensions	• Changed Document Status from published to obsolete
Specification of Secure Onboard Communication	• Minor corrections / clarifications / editorial changes
Specification of Service Discovery	• Introduced optional functionality to subscribe to a multicast address pre-defined by a ClientService • Consideration of the connection status of a security associations for clients and servers was added • Harmonization of Specification of Service Discovery and Service Discovery Protocol specification: – removal of duplicate specification items – moving of specification items from Specification of Service Discovery to Service Discovery Protocol specification • Minor bugfixes and editorial change

▽

△

Name	Specification history entry
Specification of Socket Adaptor	• Introduced config parameter SoAdSocketSoConModeChgBswM-Notification • Added limitations for SomeIP protocol handling. • Introduced config parameter SoAdSocketTcpRetransmissionTimeout • Introduced config parameter SoAd-SocketTcpAutoConnectTimeout • Introduced SoAd_IsConnectionReady() to retrieve connection status from TcpIp
Specification of Software Cluster Connection module	added support for • signal and SOME/IP based communication from Software Components located in an Applicative Software Cluster • connection to diagnostic services as Dem, FiM, Dcm. • Software Cluster Service Resources for connection of WdgM instances in different Software Clusters
Specification of SOME/IP Transformer	• Clarification on network representation • SOME/IP Header encoded in network byte order • Clarification on SOMEIPLegacyStringSerialization • Optional method arguments not supported • Clarification on Interface Version • Clarification on processing order of header fields in AUTOSAR CP • Removed SOMEIPXF_E_UNKNOWN_SERVICE and SOMEIPXF_E_UNKNOWN_METHOD • Introduction on External Trigger Events • Clarification on ISignal length of external trigger event • Editorial Changes
Specification of SPI Handler/Driver	• Chapter 10 diagrams updated • New configuration parameter: ECUC_Spi_00249 • Reworked or rephrased requirements: SWS_Spi_00128, SWS_Spi_00382, SWS_Spi_00360, SWS_Spi_00170, SWS_Spi_00150, SWS_Spi_00185, SWS_Spi_00328, SWS_Spi_00329, SWS_Spi_00154, ECUC_Spi_00208, ECUC_Spi_00214, ECUC_Spi_00202, ECUC_Spi_00204, ECUC_Spi_00205, ECUC_Spi_00234, ECUC_Spi_00242, ECUC_Spi_00197, ECUC_Spi_00198, ECUC_Spi_00199, ECUC_Spi_00236 • Removed requirements: SWS_Spi_00108, SWS_Spi_00155, SWS_Spi_00152, SWS_Spi_00271, SWS_Spi_00008, SWS_Spi_00009, SWS_Spi_00010, SWS_Spi_00063, SWS_Spi_00064, SWS_Spi_00344 • Editorial changes, errors descriptions updated, SpiDataWidth up to 64bits

▽

△

Name	Specification history entry
Specification of Standard Types	• Added SWS_Std_00031 (NULL_PTR) • Editorial Changes
Specification of SW-C End-to-End Communication Protection Library	• New profiles 8m, 44m
Specification of Synchronized Time-Base Manager	• Support for CAN HW timestamping added • API for cloning of timebases added • Rate correction of the sync reception delay added • Several minor clarifications and corrections
Specification of TCP/IP Stack	• Add TcpIp_IsConnectionReady() • Minor corrections and clarifications • Editorial changes
Specification of Time Service	• Artefact inclusion based on ArtefactAnalysis corrected.
Specification of Time Synchronization over CAN	• CAN HW timestamping added • Hysteresis added for sequence counter validation
Specification of Time Synchronization over Ethernet	• Origin Time Stamp calculation corrected • Sync reception delay corrected • Sequence Counter specified • Removed De-Init and re-Init requirments • New parameter for handling of • Sequence Counter jumps introduced • Migration to Latex Based document
Specification of Time Synchronization over FlexRay	• Hysteresis added for sequence counter validation • Small enhancement to improve precision of Global Time • Bugfix for Time Validation
Specification of Timing Extensions	• Removed some obsolete elements • Corrected description of the attribute ExecutionOrderConstraint.orderedElement • Corrected spelling errors and incorrect captions
Specification of TTCAN Driver	• No content changes
Specification of TTCAN Interface	• No content changes
Specification of UDP Network Management	• Added handling of internal requested Pnc • Improved synchronized Pnc shutdown • NM PDU filter algorithm and aggregation of internal and external requested partial networks is now obsolete and replaced • Traceability directly to RS_Nm
Specification of Vehicle-2-X Basic Transport	• Editorial changes
Specification of Vehicle-2-X Facilities	• Editorial Changes
Specification of Vehicle-2-X Geo Networking	• Correction in V2xGn_TxConfirmation prototype • Scrubbing and editorial changes
Specification of Vehicle-2-X Management	• Return codes and error reporting added
Specification of Watchdog Driver	• Removed "E_NOT_OK" return code when development error is reported
Specification of Watchdog Interface	• Editorial changes

▽

△

Name	Specification history entry
Specification of Watchdog Manager	• Resolved inconsistency regarding determination of Supervised Entity ID values, between SWS WdgM and TPS • Set "Partition Restart / Shutdown" feature to obsolete • Removed the redundant parameter WdgMDemStoppedSupervisionReport • Extended to support supervision for Clustered Software Architecture (Classic Platform Flexibility), incl. support of multiple main functions
Specification of Wireless Ethernet Driver	• No content changes
Specification of Wireless Ethernet Transceiver Driver	• Header File Cleanup • Fixed configuration ranges • Editorial changes
Specification on SOME/IP Transport Protocol	• Optional parameters to define a BurstSize to specify the number of segments that shall be transmited in a burst and a SeparationTime between these bursts were added • Several minor bugfixes • Editorial changes
Supplementary material of general blueprints for AUTOSAR	• No content changes
SW-C and System Modeling Guide	• No content changes
System Template	• Rework of Log and Trace model • Rework of TLS modeling using IANA Parameters • Introduction of Affinity Constraints • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
Unique Names for Documentation, Measurement and Calibration: Modeling and Naming Aspects including Automatic Generation	• No content changes
Utilization of Crypto Services	• No content changes
Virtual Functional Bus	• No content changes