

<b>Document Title</b>	Specification of Communication
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	015
<b>Document Classification</b>	Standard
<b>Document Status</b>	Final
<b>Part of AUTOSAR Release</b>	4.2.2

<b>Document Change History</b>		
<b>Release</b>	<b>Changed by</b>	<b>Change Description</b>
4.2.2	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation</li> </ul>
4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• added support for Sender/ Receiver Serialization</li> <li>• updated to support CAN FD</li> <li>• minor corrections</li> </ul>
4.1.3	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Several correction, clarifications and editorial changes</li> <li>• No major functional changes or improvements</li> </ul>
4.1.2	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Removed TP-restriction for dynamic length I-PDUs</li> <li>• Added parameter to globally enable or disable the cancellation support</li> <li>• Many bug fixes, clarifications and harmonization with the other COM-Stack modules</li> <li>• Editorial changes</li> <li>• Removed chapter(s) on change documentation</li> </ul>
4.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Reworked according to the new SWS BSW General</li> <li>• added support for postbuild loadable configuration</li> <li>• harmonization of COM-Stack APIs (TP-APIs &amp; callouts)</li> <li>• many corrections and clarifications</li> </ul>
4.0.3	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Several minor changes and bug-fixes</li> <li>• Clarification and minor extensions of transmission mode usage and activation</li> <li>• Clarification and extension of transfer properties</li> <li>• Added retry mechanism for failed transmission requests</li> </ul>

<b>Document Change History</b>		
<b>Release</b>	<b>Changed by</b>	<b>Change Description</b>
3.1.5	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Update: SWS_Com_00696, SWS_Com_00697, SWS_Com_00442, ECUC_Com_00017 and SWS_Com_00217, ECUC_Com_00550, SWS_Com_00115, SWS_Com_00260, SWS_Com_00708, ECUC_Com_00709, SWS_Com_00469, SWS_Com_00698, SWS_Com_00702, SWS_Com_00703, SWS_Com_00704, SWS_Com_00705, SWS_Com_00706, ECUC_Com_00549, SWS_Com_00700, SWS_Com_00346, SWS_Com_00198, SWS_Com_00691, SWS_Com_00654, SWS_Com_00655, SWS_Com_00692, SWS_Com_00693, SWS_Com_00459 (Table 12), SWS_Com_00005, SWS_Com_00053, SWS_Com_00673, ECUC_Com_00175, SWS_Com_00401, Com_IpduGroupVector in Chapter 8.2.5, SWS_Com_00495, SWS_Com_00469, ECUC_Com_00263, ECUC_Com_00232, SWS_Com_00707</li> <li>• Remove SWS_Com_00626, SWS_Com_00329</li> <li>• Add SWS_Com_00731, SWS_Com_00733, SWS_Com_00734, SWS_Com_00732</li> </ul>
3.1.4	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Added support for large data types</li> <li>• Added support for communication protection</li> <li>• Revised mode management interface</li> <li>• Many clarifications and corrections based on SWS improvement, validation and conformance test activities</li> <li>• Legal disclaimer revised</li> </ul>
3.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Legal disclaimer revised</li> </ul>
3.0.2	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Removed Figure 19, which was included twice</li> </ul>
3.0.1	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Removal of Internal Communication feature</li> <li>• Revision of AUTOSAR COM MODULE configuration</li> <li>• Revision of Signal Gateway configuration</li> <li>• Enhanced maximal</li> <li>• I-PDU length for FlexRay</li> <li>• Harmonization of callback-interface to RTE</li> <li>• Clarifications and bug fixes</li> <li>• Document meta information extended</li> <li>• Small layout adaptations made</li> </ul>

<b>Document Change History</b>		
<b>Release</b>	<b>Changed by</b>	<b>Change Description</b>
2.1.15	AUTOSAR Administration	<ul style="list-style-type: none"><li>• Clarifications of requirements throughout the whole document. Chapter 11 contains a detailed list of the made modifications. No new major features were added since release 2.0.</li><li>•</li><li>• “Advice for users” revised</li><li>• “Revision Information” added</li><li>• Legal disclaimer revised</li></ul>
2.0	AUTOSAR Administration	<ul style="list-style-type: none"><li>• Document structure adapted to common Release 2.0 SWS Template.</li><li>• Integration of signal gateway</li><li>• Major updates in configuration, error handling, filtering, transmission mode switches, callouts, update-bits, deadline monitoring and initialization</li></ul>
1.0	AUTOSAR Administration	<ul style="list-style-type: none"><li>• Initial release</li></ul>

## Disclaimer

This specification 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 specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification 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 specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

## Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

## Table of Contents

1	Introduction and Functional Overview .....	10
2	Acronyms, Abbreviations and Definitions .....	11
2.1	Acronyms and Abbreviations.....	11
2.2	Definitions .....	11
3	Related Documentation .....	13
3.1	Deliverables of AUTOSAR .....	13
3.2	Related Standards and Norms .....	14
3.3	Related Specification.....	14
4	Constraints and Assumptions.....	15
4.1	Limitations .....	15
4.2	Applicability to Car Domains .....	15
5	Dependencies to Other Modules .....	16
5.1	PDU Router.....	16
5.2	Runtime Environment (RTE) .....	17
5.3	File Structure.....	17
5.3.1	Header File Structure .....	17
6	Requirements Traceability .....	19
7	Functional Specification.....	30
7.1	Introduction .....	30
7.2	General Functionality .....	30
7.2.1	OSEK-COM.....	30
7.2.2	Signal Values .....	31
7.2.3	Endianness Conversion and Sign Extension .....	32
7.2.4	Filtering .....	33
7.2.5	Signal Gateway .....	36
7.3	Normal Operation.....	37
7.3.1	Start-Up Behavior.....	37
7.3.1.1	Preconditions.....	37
7.3.1.2	Initialization.....	37
7.3.1.3	Initialization of Not Used Areas of an I-PDU .....	37
7.3.1.4	Initialization of Signals and Update-Bits .....	38
7.3.1.5	Initialization of I-PDU Groups .....	38
7.3.2	De-Initialization.....	38
7.3.3	Communication Modes .....	38
7.3.3.1	Transfer Properties and I-PDU Transmission Mode .....	39
7.3.3.2	Selection of the Transmission Mode for one specific I-PDU .....	41
7.3.3.3	Signal Flow and Transmission Mode Selection .....	43
7.3.3.4	Replication of Signal Transmission Requests.....	45
7.3.3.5	Retry Failed Transmission Requests .....	46
7.3.4	Signal Invalidation .....	47
7.3.4.1	Transmission of an Invalidated Signal .....	47
7.3.4.2	Reception of an Invalidated Signal .....	48
7.3.5	Handling of I-PDUs .....	50
7.3.5.1	I-PDU Group Definitions .....	50

7.3.5.2	Starting of I-PDU Groups.....	51
7.3.5.3	Stopping of I-PDU Groups.....	52
7.3.5.4	Signal Indication (Unpacking of I-PDUs) .....	54
7.3.5.5	Minimum Delay Timer (MDT).....	55
7.3.6	Deadline Monitoring .....	57
7.3.6.1	Reception Deadline Monitoring.....	57
7.3.6.2	Transmission Deadline Monitoring .....	59
7.4	Signal Groups - Complex Data Types .....	61
7.4.1	Initialization .....	62
7.4.2	Transmission.....	62
7.4.3	Reception.....	62
7.4.4	Notifications.....	63
7.4.5	Attributes of a Signal Group.....	64
7.4.6	UINT8-array based access to signal groups .....	64
7.4.6.1	Basic functionality .....	65
7.4.6.2	Initialization.....	65
7.4.6.3	Transmission .....	66
7.4.6.4	Reception .....	66
7.5	Large Data Types.....	66
7.5.1	Transmission of Large Signals/ I-PDUs .....	67
7.5.2	Reception of Large Signals/ I-PDUs.....	68
7.6	Dynamic Length Signals.....	68
7.6.1	Transmission of Dynamic Length Signals/ I-PDUs.....	69
7.6.2	Reception of Dynamic Length Signals/ I-PDUs .....	69
7.7	Interface between AUTOSAR COM Module and the PDU Router .....	70
7.8	Update-Bits .....	70
7.8.1	Sender Side .....	71
7.8.2	Receiver Side.....	72
7.9	Data Sequence Control .....	72
7.9.1	Sender Side .....	72
7.9.2	Receiver Side.....	73
7.10	Communication Protection .....	74
7.10.1	Sender Side .....	74
7.10.2	Receiver Side.....	74
7.10.3	Constraints.....	75
7.11	Signal Gateway .....	75
7.11.1	Dealing with Signals.....	76
7.11.2	Dealing with Signal Groups.....	76
7.11.3	Routing of Out-Timed Signals and Signal Groups.....	77
7.11.4	Handling of Update-Bits .....	77
7.11.5	Decoupling Signal Gateway .....	78
7.12	Error Classification .....	78
7.13	Error Notification.....	79
7.14	Error Handling .....	79
7.15	AUTOSAR COM Module's Interaction Model.....	79
8	API Specification .....	84
8.1	Imported Types .....	84
8.2	Type Definitions.....	84
8.2.1	Com_StatusType .....	84

8.2.2	Com_SignalIdType.....	84
8.2.3	Com_SignalGroupIdType.....	84
8.2.4	Com_IpduGroupIdType.....	85
8.2.5	Com_IpduGroupVector .....	85
8.2.6	Com_ServiceIdType.....	85
8.2.7	Com_ConfigType .....	86
8.3	Function Definitions.....	86
8.3.1	Return Codes.....	86
8.3.2	Start-Up and Control Services.....	87
8.3.2.1	Com_Init .....	87
8.3.2.2	Com_DelInit.....	88
8.3.2.3	Com_IpduGroupControl .....	88
8.3.2.4	Com_ReceptionDMControl.....	89
8.3.2.5	Com_GetStatus .....	89
8.3.2.6	Com_GetVersionInfo .....	90
8.3.2.7	Com_ClearIpduGroupVector .....	90
8.3.2.8	Com_SetIpduGroup.....	90
8.3.3	Communication Services .....	91
8.3.3.1	Com_SendSignal.....	91
8.3.3.2	Com_SendDynSignal .....	92
8.3.3.3	Com_ReceiveSignal .....	93
8.3.3.4	Com_ReceiveDynSignal.....	93
8.3.3.5	Com_UpdateShadowSignal.....	94
8.3.3.6	Com_SendSignalGroup.....	95
8.3.3.7	Com_ReceiveSignalGroup .....	95
8.3.3.8	Com_ReceiveShadowSignal .....	96
8.3.3.9	Com_SendSignalGroupArray .....	97
8.3.3.10	Com_ReceiveSignalGroupArray.....	97
8.3.3.11	Com_InvalidateSignal.....	98
8.3.3.12	Com_InvalidateShadowSignal.....	99
8.3.3.13	Com_InvalidateSignalGroup.....	99
8.3.3.14	Com_TriggerIPDUSend.....	100
8.3.3.15	Com_TriggerIPDUSendWithMetaData .....	101
8.3.3.16	Com_SwitchIpduTxMode.....	102
8.4	Callback Functions and Notifications.....	103
8.4.1	Com_TriggerTransmit .....	103
8.4.2	Com_RxIndication.....	104
8.4.3	Com_TpRxIndication.....	104
8.4.4	Com_TxConfirmation .....	105
8.4.5	Com_TpTxConfirmation .....	105
8.4.6	Com_StartOfReception .....	105
8.4.7	Com_CopyRxData .....	107
8.4.8	Com_CopyTxData.....	107
8.5	Scheduled Functions.....	109
8.5.1	Com_MainFunctionRx.....	109
8.5.2	Com_MainFunctionTx .....	109
8.5.3	Com_MainFunctionRouteSignals.....	109
8.6	Expected Interfaces.....	110
8.6.1	Mandatory Interfaces .....	110
8.6.2	Optional Interfaces .....	110

8.6.3	Configurable Interfaces .....	111
8.6.3.1	Callback/ Notification Functions.....	111
8.6.3.2	I-PDU Callout Functions .....	114
9	Sequence Diagrams .....	117
9.1	Interface between the AUTOSAR COM Module and the PDU Router .....	117
9.2	Confirmation Handling between the PDU Router, the AUTOSAR COM Module and the RTE .....	119
9.3	Indication Handling between the PDU Router, the AUTOSAR COM Module and the RTE .....	120
10	Configuration Specification .....	121
10.1	Containers and Configuration Parameters .....	121
10.1.1	Variants .....	121
10.1.1.1	VARIANT-PRE-COMPILE .....	121
10.1.1.2	VARIANT-LINK-TIME .....	121
10.1.1.3	VARIANT-POST-BUILD .....	121
10.1.2	Configuration of the AUTOSAR COM Module.....	122
10.1.3	Com .....	123
10.1.4	ComGeneral.....	123
10.1.5	ComConfig .....	126
10.1.6	ComTimeBase .....	127
10.1.7	ComFilter.....	129
10.1.8	ComIPdu .....	132
10.1.9	ComIPduCounter .....	136
10.1.10	ComIPduReplication.....	138
10.1.11	ComTxIPdu.....	139
10.1.12	ComIPduGroup.....	141
10.1.13	ComSignal .....	142
10.1.14	ComSignalGroup .....	151
10.1.15	ComGroupSignal .....	158
10.1.16	ComTxMode .....	163
10.1.17	ComTxModeTrue.....	165
10.1.18	ComTxModeFalse .....	166
10.1.19	ComGwMapping.....	166
10.1.20	ComGwSource .....	166
10.1.21	ComGwSourceDescription .....	167
10.1.22	ComGwDestination.....	170
10.1.23	ComGwDestinationDescription.....	170
10.1.24	ComGwSignal.....	173
10.2	Configuration Rules.....	174
10.2.1	General Rules .....	174
10.2.2	Signal Configuration .....	175
10.2.3	Signal Group Configuration .....	176
10.2.4	Transmission Mode Configuration.....	176
10.2.5	Signal Gateway Configuration.....	177
10.2.6	Filter Configuration.....	177
10.2.7	Post Build Configuration.....	177
10.2.8	Dynamic Length I-PDUs.....	177
10.2.9	Replicated I-PDUs.....	177



11	Not Applicable Requirements .....	179
12	Appendix A .....	180

# 1 Introduction and Functional Overview

This specification is the AUTOSAR COM module Software Specification. It is based on the AUTOSAR COM SRS [7]. It specifies how the requirements of the AUTOSAR COM SRS shall be realized. That means that the functionality and the API of the AUTOSAR COM module are described in this document.

Within the AUTOSAR Layered Architecture the AUTOSAR COM module is placed between RTE and the PDU Router, see [1].

The AUTOSAR COM module is derived from [17]. For details, see Chapter 7.2.1. The AUTOSAR COM module provides signal gateway functionality. For details, see Chapter 7.2.5.

Main Features:

- Provision of signal oriented data interface for the RTE
- Packing of AUTOSAR signals to I-PDUs to be transmitted
- Unpacking of received I-PDUs and provision of received signals to RTE
- Routing of signals from received I-PDUs into I-PDUs to become transmitted
- Routing of signal groups from received I-PDUs into I-PDUs to become transmitted
- Communication transmission control (start/ stop of I-PDU groups)
- Replications of send requests
- Guarantee of minimum distances between transmit I-PDUs
- Monitoring of receive signals (signals timeout)
- Filter mechanisms for incoming signals
- Different notification mechanisms
- Provision of init values and update indications
- Byte order conversion
- Sign extension
- Support of two different transmission modes per I-PDU
- Signal based gateway
- Support of large and dynamic length data types
- Support of I-PDU counters and I-PDU replication

## 2 Acronyms, Abbreviations and Definitions

### 2.1 Acronyms and Abbreviations

<b>Acronym:</b>	<b>Description:</b>
AUTOSAR COM	The AUTOSAR COM module is derived from OSEK COM [17]. For details, see Chapter 7.2.1.
DM	Deadline Monitoring, for details see Chapter 7.3.6
I-PDU	Interaction Layer Protocol Data Unit An I-PDU carries signals. It is defined in [17].
L-PDU	Data Link Layer Protocol Data Unit. In AUTOSAR, the Data Link Layer is equivalent to the Communication Hardware Abstraction and Microcontroller Abstraction Layer.
MDT	A detailed description of the Minimum Delay Timer (MDT) can be found in [17]. See also Chapter 7.3.5.5.
OSEK COM	Open systems and the corresponding interfaces for automotive electronics – communication [17].
PDU Router	The PDU Router is a module transferring I-PDUs from one module to another module. The PDU Router can be utilized for gateway operations and for internal routing purposes.
SDU	Service Data Unit For a description see [1] Chapter 4.
TM	Transmission Mode
TMC	Transmission Mode Condition, see Chapter 7.3.3.2
TMS	Transmission Mode Selector, see Chapter 7.3.3.2

### 2.2 Definitions

<b>Term:</b>	<b>Description:</b>
Confirmation	With a Confirmation, the PDU Router reports that a request by the AUTOSAR COM module has been completed successfully. It is a reaction to a request of COM. E.g. when a PDU has been successfully transmitted.
Data Invalid Value	Value sent by the AUTOSAR COM module to indicate that the sender side AUTOSAR Software Component is not able to provide a valid value.
Dynamic Length Signal	A dynamic length signal is a signal which length can vary at run-time.
Dynamic Length I-PDU	A dynamic length I-PDU is an I-PDU containing a dynamic length signal. Its length varies depending on the length of the included dynamic length signal.
Group signal	A group signal is a signal that is contained in a signal group.
Indication	An Indication is asynchronous information from PDU Router to COM, e.g. to acknowledge that something has been received.
Init Value	I-PDUs and signals are set to the Init Value by the AUTOSAR COM module after start-up. This value is used until it is overwritten.
I-PDU group	An I-PDU Group is an arbitrary collection of I-PDUs in the AUTOSAR COM module.
Inter-ECU – communication	Communication between two or more ECUs for example via a CAN network
Intra-ECU – communication	Communication between Software components that reside on the same ECU
Large Signal	A large signal is a signal that is too large to fit into a single L-PDU of the underlying communication protocol.
Large I-PDU	Large I-PDU are I-PDU that do not fit into a single L-PDU of the underlying

<b>Term:</b>	<b>Description:</b>
	communication protocol. Large I-PDU will be transmitted via TP.
Message	OSEK-COM uses always the synonym <i>message</i> . In AUTOSAR, <i>message</i> is replaced by <i>signal</i> but with the same meaning.
Metadata	For some I-PDUs, e.g. J1939 I-PDUs, the payload is extended with additional metadata containing for example the CAN-ID.
Notification	Information by the AUTOSAR COM module to RTE, e.g. when new data is available, an error occurred.
Signal	A signal in the AUTOSAR COM module's context is equal to a message in OSEK COM; see also [7].
Signal group	<p>In AUTOSAR, so called complex data types are used. Inside a complex data type, there are one or more data elements (primitive data types), like in a C struct. The data consistency of such complex data types must be ensured.</p> <p>The RTE decomposes the complex data type in single signals and sends them to the AUTOSAR COM module. As these signals altogether need to be treated consistently, they are called <i>signal group</i>.</p> <p>See also [7].</p>
Update-bit	A mechanism supported by the AUTOSAR COM module with that the receiver of a signal/ signal group could identify whether the sender has updated the data in this signal/ signal group before sending. See Chapter 7.8.

### 3 Related Documentation

#### 3.1 Deliverables of AUTOSAR

- [1] AUTOSAR Layered Architecture  
AUTOSAR\_EXP\_LayeredSoftwareArchitecture.pdf
- [2] Specification of Communication Stack Types  
AUTOSAR\_SWS\_CommunicationStackTypes.pdf
- [3] General Requirements on Basic Software Modules  
AUTOSAR\_SRS\_BSWGeneral.pdf
- [4] Basic Software UML Model  
AUTOSAR\_MOD\_BSWUMLModel.eap
- [5] Specification of Standard Types  
AUTOSAR\_SWS\_StandardTypes.pdf
- [6] Specification of the Virtual Functional Bus  
AUTOSAR\_EXP\_VFB.pdf
- [7] Requirements on Communication  
AUTOSAR\_SRS\_COM.pdf
- [8] Software Component Template  
AUTOSAR\_TPS\_SoftwareComponentTemplate.pdf
- [9] Requirements on Gateway  
AUTOSAR\_SRS\_Gateway.pdf
- [10] Specification of PDU Router  
AUTOSAR\_SWS\_PDURouter.pdf
- [11] Specification of Operating System  
AUTOSAR\_SWS\_OS.pdf
- [12] System Template  
AUTOSAR\_TPS\_SystemTemplate.pdf
- [13] Specification of RTE Software  
AUTOSAR\_SWS\_RTE.pdf
- [14] Specification of ECU Configuration  
AUTOSAR\_TPS\_ECUConfiguration.pdf

- [15] Specification of Communication Manager  
AUTOSAR\_SWS\_COMManager.pdf
- [16] AUTOSAR Basic Software Module Description Template  
AUTOSAR\_TPS\_BSWModuleDescriptionTemplate.pdf
- [19] Specification of CAN Transport Layer  
AUTOSAR\_SWS\_CANTransportLayer.pdf
- [20] Specification of FlexRay Transport Layer  
AUTOSAR\_SWS\_FlexRayTransportLayer.pdf
- [21] List of Basic Software Modules,  
AUTOSAR\_TR\_BSWModuleList.pdf
- [22] Generic Structure Template  
AUTOSAR\_TPS\_GenericStructureTemplate.pdf
- [23] General Specification of Basic Software Modules  
AUTOSAR\_SWS\_BSWGeneral.pdf

### **3.2 Related Standards and Norms**

- [17] OSEK/ VDX Communication Version 3.0.3  
OSEKCOM303.pdf
- [18] OSEK implementation language Version 2.5  
OIL25.pdf

### **3.3 Related Specification**

AUTOSAR provides a General Specification on Basic Software modules [23] (SWS BSW General), which is also valid for COM.

Thus, the specification SWS BSW General shall be considered as additional and required specification for COM.

## 4 Constraints and Assumptions

This document is applicable for AUTOSAR release 4.1.

### 4.1 Limitations

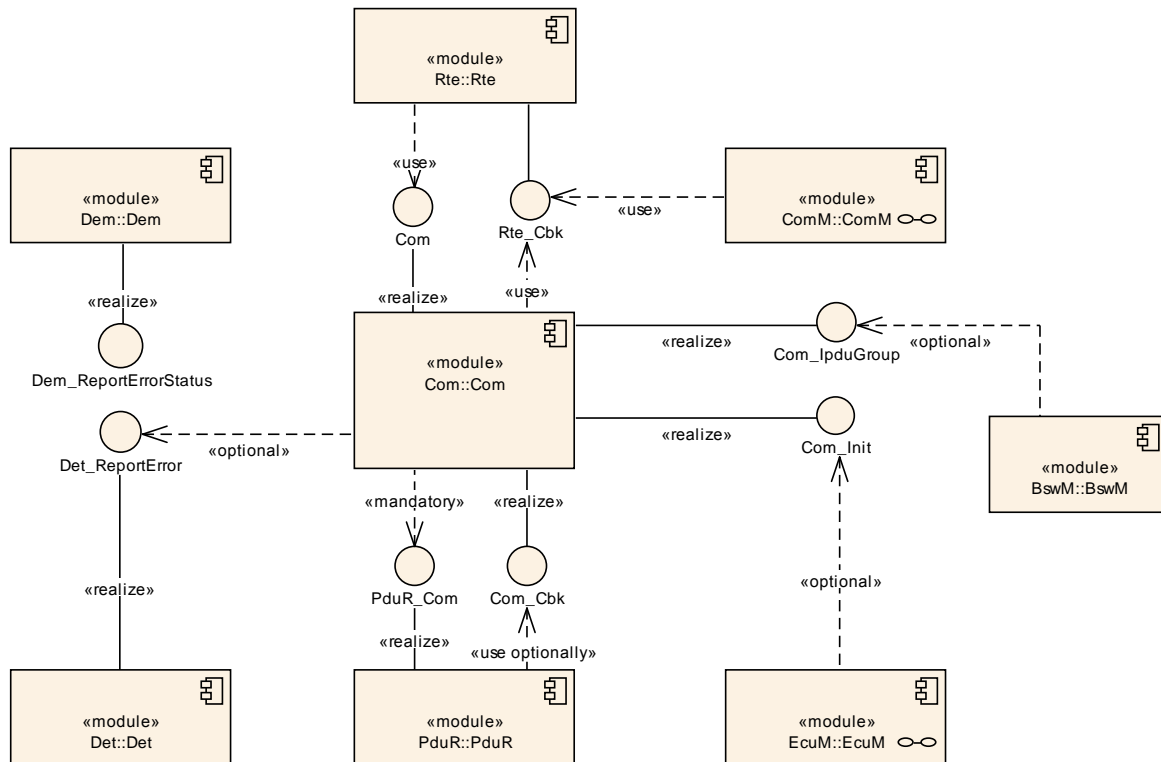
The AUTOSAR COM module is based on [17]. Nevertheless not all features of [17] are included and some features are different. See SWS\_Com\_00013 for a list of not included features.

### 4.2 Applicability to Car Domains

No restrictions.

## 5 Dependencies to Other Modules

This chapter lists all the features from other modules that are used by the AUTOSAR COM module and functionalities that are provided by the AUTOSAR COM module to other modules. For the placement of the AUTOSAR COM module in the communication stack, see **Figure 1**.



**Figure 1: AUTOSAR COM module’s context view**

### 5.1 PDU Router

The AUTOSAR COM module uses the union of both sets of PDU Router’s upper layer module APIs. That is the APIs for upper layer modules that uses TP and the APIs for upper layer modules that do not use TP. This is necessary since the AUTOSAR COM module transports I-PDUs either unfragmented via simple L-PDUs or fragmented via TP.

The following summarizes the functionality of the AUTOSAR COM module needs from the underlying layer PDU Router:



- Indication of incoming I-PDUs
- Sending interface for outgoing I-PDUs including the confirmation if an I-PDU has been sent by the communication controller
- Trigger interface to enable the PDU router to cause a transmission from the AUTOSAR COM module
- Buffer handling for TP communication

Chapter 7.5 and Chapter 9.1 include a detailed description of the interfaces to the PDU Router. For further information, see [10].

## 5.2 Runtime Environment (RTE)

The RTE uses the capabilities of the AUTOSAR COM module to send and receive signals. In AUTOSAR, the RTE is the higher layer above the AUTOSAR COM module. For further information, see [13].

## 5.3 File Structure

### 5.3.1 Header File Structure

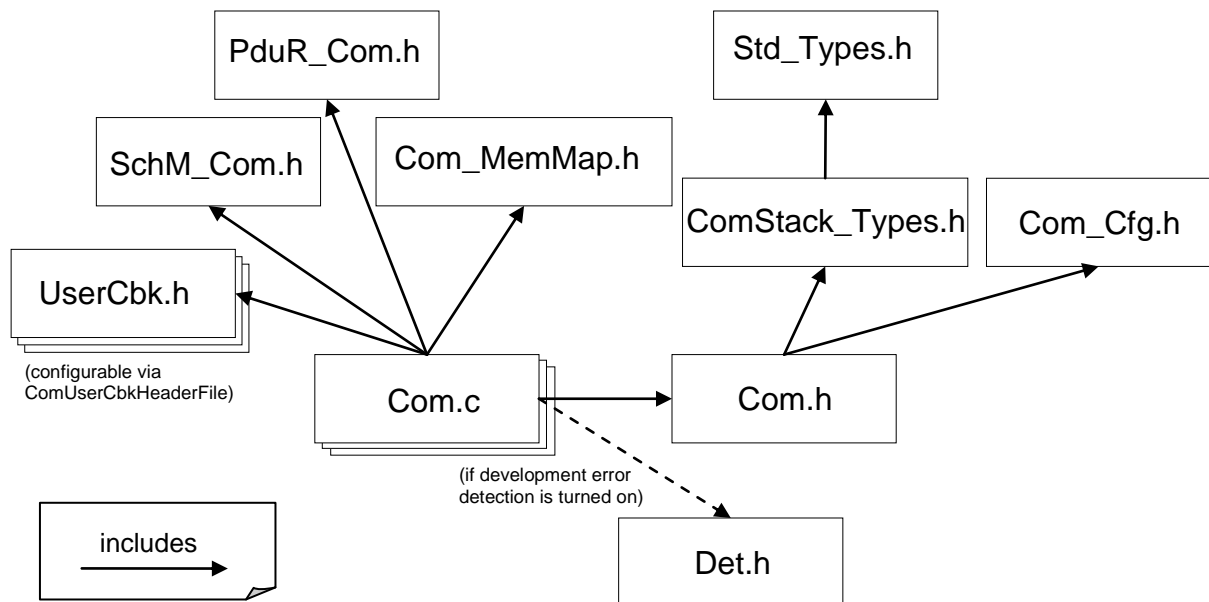


Figure 2: Include file structure

[SWS\_Com\_00220] [The Com.c file shall include:

- Com.h
- Com\_MemMap.h

- PduR\_Com.h
- SchM\_Com.h
- all user callback header files as configured per ComUserCbHeaderFile (ECUC\_Com\_10001)

] (SRS\_BSW\_00300, SRS\_BSW\_00415)

**[SWS\_Com\_00796]** [The Com.h file shall include:

- Com\_Cfg.h
- ComStack\_Types.h

] (SRS\_BSW\_00346, SRS\_BSW\_00381, SRS\_BSW\_00412)

## 6 Requirements Traceability

The following table references the requirements specified in [3], [7], [17] and [9] and links to the fulfillment of these. Requirements that are not fulfilled by this document are linked to SWS\_Com\_00999.

Requirement	Description	Satisfied by
SRS_BSW_00003	All software modules shall provide version and identification information	SWS_Com_00426
SRS_BSW_00005	Modules of the $\hat{\mu}$ C Abstraction Layer (MCAL) may not have hard coded horizontal interfaces	SWS_Com_00999
SRS_BSW_00006	The source code of software modules above the $\hat{\mu}$ C Abstraction Layer (MCAL) shall not be processor and compiler dependent.	SWS_Com_00999
SRS_BSW_00009	All Basic SW Modules shall be documented according to a common standard.	SWS_Com_00999
SRS_BSW_00010	The memory consumption of all Basic SW Modules shall be documented for a defined configuration for all supported platforms.	SWS_Com_00999
SRS_BSW_00101	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	SWS_Com_00015, SWS_Com_00059, SWS_Com_00098, SWS_Com_00117, SWS_Com_00128, SWS_Com_00217, SWS_Com_00328, SWS_Com_00432, SWS_Com_00484, SWS_Com_00850
SRS_BSW_00161	The AUTOSAR Basic Software shall provide a microcontroller abstraction layer which provides a standardized interface to higher software layers	SWS_Com_00999
SRS_BSW_00162	The AUTOSAR Basic Software shall provide a hardware abstraction layer	SWS_Com_00999
SRS_BSW_00164	The Implementation of interrupt service routines shall be done by the Operating System, complex drivers or modules	SWS_Com_00999
SRS_BSW_00167	All AUTOSAR Basic Software Modules shall provide configuration rules and constraints to enable plausibility checks	SWS_Com_00497

SRS_BSW_00168	SW components shall be tested by a function defined in a common API in the Basis-SW	SWS_Com_00999
SRS_BSW_00170	The AUTOSAR SW Components shall provide information about their dependency from faults, signal qualities, driver demands	SWS_Com_00999
SRS_BSW_00171	Optional functionality of a Basic-SW component that is not required in the ECU shall be configurable at pre-compile-time	SWS_Com_00999
SRS_BSW_00300	All AUTOSAR Basic Software Modules shall be identified by an unambiguous name	SWS_Com_00220
SRS_BSW_00301	All AUTOSAR Basic Software Modules shall only import the necessary information	SWS_Com_00609
SRS_BSW_00302	All AUTOSAR Basic Software Modules shall only export information needed by other modules	SWS_Com_00999
SRS_BSW_00306	AUTOSAR Basic Software Modules shall be compiler and platform independent	SWS_Com_00999
SRS_BSW_00307	Global variables naming convention	SWS_Com_00999
SRS_BSW_00308	AUTOSAR Basic Software Modules shall not define global data in their header files, but in the C file	SWS_Com_00999
SRS_BSW_00309	All AUTOSAR Basic Software Modules shall indicate all global data with read-only purposes by explicitly assigning the const keyword	SWS_Com_00999
SRS_BSW_00312	Shared code shall be re-entrant	SWS_Com_00321
SRS_BSW_00314	All internal driver modules shall separate the interrupt frame definition from the service routine	SWS_Com_00999
SRS_BSW_00321	The version numbers of AUTOSAR Basic Software Modules shall be enumerated according specific rules	SWS_Com_00999
SRS_BSW_00325	The runtime of interrupt service routines and functions that are running in	SWS_Com_00999

	interrupt context shall be kept short	
SRS_BSW_00327	Error values naming convention	SWS_Com_00442
SRS_BSW_00328	All AUTOSAR Basic Software Modules shall avoid the duplication of code	SWS_Com_00999
SRS_BSW_00331	All Basic Software Modules shall strictly separate error and status information	SWS_Com_00194
SRS_BSW_00333	For each callback function it shall be specified if it is called from interrupt context or not	SWS_Com_00999
SRS_BSW_00334	All Basic Software Modules shall provide an XML file that contains the meta data	SWS_Com_00999
SRS_BSW_00335	Status values naming convention	SWS_Com_00819
SRS_BSW_00336	Basic SW module shall be able to shutdown	SWS_Com_00129, SWS_Com_00130
SRS_BSW_00337	Classification of development errors	SWS_Com_00803, SWS_Com_00804
SRS_BSW_00341	Module documentation shall contain all needed informations	SWS_Com_00999
SRS_BSW_00344	BSW Modules shall support link-time configuration	SWS_Com_00432, SWS_Com_00607
SRS_BSW_00345	BSW Modules shall support pre-compile configuration	SWS_Com_00606
SRS_BSW_00346	All AUTOSAR Basic Software Modules shall provide at least a basic set of module files	SWS_Com_00796
SRS_BSW_00347	A Naming separation of different instances of BSW drivers shall be in place	SWS_Com_00999
SRS_BSW_00348	All AUTOSAR standard types and constants shall be placed and organized in a standard type header file	SWS_Com_00865
SRS_BSW_00353	All integer type definitions of target and compiler specific scope shall be placed and organized in a single type header	SWS_Com_00999
SRS_BSW_00358	The return type of init() functions implemented by AUTOSAR Basic Software Modules shall be void	SWS_Com_00432
SRS_BSW_00359	All AUTOSAR Basic Soft-	SWS_Com_00468, SWS_Com_00491,

	ware Modules callback functions shall avoid return types other than void if possible	SWS_Com_00536, SWS_Com_00554, SWS_Com_00555, SWS_Com_00556
SRS_BSW_00360	AUTOSAR Basic Software Modules callback functions are allowed to have parameters	SWS_Com_00468, SWS_Com_00491, SWS_Com_00536, SWS_Com_00554, SWS_Com_00555, SWS_Com_00556
SRS_BSW_00361	All mappings of not standardized keywords of compiler specific scope shall be placed and organized in a compiler specific type and keyword header	SWS_Com_00999
SRS_BSW_00369	All AUTOSAR Basic Software Modules shall not return specific development error codes via the API	SWS_Com_00442
SRS_BSW_00375	Basic Software Modules shall report wake-up reasons	SWS_Com_00999
SRS_BSW_00377	A Basic Software Module can return a module specific types	SWS_Com_00865
SRS_BSW_00378	AUTOSAR shall provide a boolean type	SWS_Com_00999
SRS_BSW_00381	The pre-compile time parameters shall be placed into a separate configuration header file	SWS_Com_00796
SRS_BSW_00383	The Basic Software Module specifications shall specify which other configuration files from other modules they use at least in the description	SWS_Com_00999
SRS_BSW_00384	The Basic Software Module specifications shall specify at least in the description which other modules they require	SWS_Com_00669, SWS_Com_00670
SRS_BSW_00385	List possible error notifications	SWS_Com_00442
SRS_BSW_00386	The BSW shall specify the configuration for detecting an error	SWS_Com_00999
SRS_BSW_00404	BSW Modules shall support post-build configuration	SWS_Com_00432, SWS_Com_00608, SWS_Com_00825
SRS_BSW_00405	BSW Modules shall support multiple configuration sets	SWS_Com_00432
SRS_BSW_00407	Each BSW module shall provide a function to read	SWS_Com_00426

	out the version information of a dedicated module implementation	
SRS_BSW_00409	All production code error ID symbols are defined by the Dem module and shall be retrieved by the other BSW modules from Dem configuration	SWS_Com_00999
SRS_BSW_00410	Compiler switches shall have defined values	SWS_Com_00999
SRS_BSW_00412	References to c-configuration parameters shall be placed into a separate h-file	SWS_Com_00796
SRS_BSW_00413	An index-based accessing of the instances of BSW modules shall be done	SWS_Com_00999
SRS_BSW_00414	Init functions shall have a pointer to a configuration structure as single parameter	SWS_Com_00432, SWS_Com_00805, SWS_Com_00837
SRS_BSW_00415	Interfaces which are provided exclusively for one module shall be separated into a dedicated header file	SWS_Com_00220
SRS_BSW_00416	The sequence of modules to be initialized shall be configurable	SWS_Com_00999
SRS_BSW_00417	Software which is not part of the SW-C shall report error events only after the DEM is fully operational.	SWS_Com_00999
SRS_BSW_00423	BSW modules with AUTOSAR interfaces shall be describable with the means of the SW-C Template	SWS_Com_00999
SRS_BSW_00424	BSW module main processing functions shall not be allowed to enter a wait state	SWS_Com_00999
SRS_BSW_00425	The BSW module description template shall provide means to model the defined trigger conditions of schedulable objects	SWS_Com_00359, SWS_Com_00398, SWS_Com_00399, SWS_Com_00400, SWS_Com_00664, SWS_Com_00665, SWS_Com_00666
SRS_BSW_00426	BSW Modules shall ensure data consistency of data which is shared between BSW modules	SWS_Com_00999
SRS_BSW_00427	ISR functions shall be defined and documented in the BSW module description template	SWS_Com_00999

SRS_BSW_00428	A BSW module shall state if its main processing function(s) has to be executed in a specific order or sequence	SWS_Com_00999
SRS_BSW_00429	BSW modules shall be only allowed to use OS objects and/or related OS services	SWS_Com_00999
SRS_BSW_00432	Modules should have separate main processing functions for read/receive and write/transmit data path	SWS_Com_00359, SWS_Com_00398, SWS_Com_00399, SWS_Com_00400, SWS_Com_00466
SRS_BSW_00433	Main processing functions are only allowed to be called from task bodies provided by the BSW Scheduler	SWS_Com_00999
SRS_BSW_00437	Memory mapping shall provide the possibility to define RAM segments which are not to be initialized during startup	SWS_Com_00999
SRS_BSW_00441	Naming convention for type, macro and function	SWS_Com_00820, SWS_Com_00821, SWS_Com_00822, SWS_Com_00823, SWS_Com_00824, SWS_Com_00825, SWS_Com_00865
SRS_BSW_00452	Classification of runtime errors	SWS_Com_00864
SRS_BSW_00462	All Standardized Autosar Interfaces shall have unique requirement Id / number	SWS_Com_00824
SRS_Com_00177	AUTOSAR COM and LargeDataCOM shall support multiple configuration stages	SWS_Com_00853, SWS_Com_00856
SRS_Com_00192	The AUTOSAR COM module shall support enabling and disabling reception deadline monitoring of I-PDU groups	SWS_Com_00224, SWS_Com_00225, SWS_Com_00486, SWS_Com_00534, SWS_Com_00616, SWS_Com_00617, SWS_Com_00618, SWS_Com_00752, SWS_Com_00772
SRS_Com_00218	The AUTOSAR COM module shall support starting and stopping multiple I-PDU groups during runtime	SWS_Com_00114, SWS_Com_00115, SWS_Com_00222, SWS_Com_00223, SWS_Com_00228, SWS_Com_00229, SWS_Com_00334, SWS_Com_00444, SWS_Com_00479, SWS_Com_00684, SWS_Com_00685, SWS_Com_00713, SWS_Com_00714, SWS_Com_00733, SWS_Com_00751, SWS_Com_00771, SWS_Com_00777, SWS_Com_00782, SWS_Com_00783, SWS_Com_00787, SWS_Com_00792, SWS_Com_00800, SWS_Com_00822, SWS_Com_00840
SRS_Com_02030	The AUTOSAR COM module shall support to detect if a received signal or signal group was updated by the	SWS_Com_00055, SWS_Com_00059, SWS_Com_00061, SWS_Com_00062, SWS_Com_00067, SWS_Com_00117, SWS_Com_00310, SWS_Com_00324,



	sender	SWS_Com_00577, SWS_Com_00578, SWS_Com_00702, SWS_Com_00703, SWS_Com_00704, SWS_Com_00705, SWS_Com_00706, SWS_Com_00801, SWS_Com_00802
SRS_Com_02037	AUTOSAR COM module shall be based on the functionality and APIs of OSEK COM 3.0.3	SWS_Com_00010, SWS_Com_00011, SWS_Com_00012, SWS_Com_00013, SWS_Com_00132, SWS_Com_00138, SWS_Com_00197, SWS_Com_00198, SWS_Com_00231, SWS_Com_00273, SWS_Com_00302, SWS_Com_00303, SWS_Com_00304, SWS_Com_00325, SWS_Com_00346, SWS_Com_00348, SWS_Com_00380, SWS_Com_00381, SWS_Com_00388, SWS_Com_00395, SWS_Com_00396, SWS_Com_00439, SWS_Com_00445, SWS_Com_00469, SWS_Com_00471, SWS_Com_00481, SWS_Com_00492, SWS_Com_00603, SWS_Com_00604, SWS_Com_00624, SWS_Com_00625, SWS_Com_00631, SWS_Com_00694, SWS_Com_00695, SWS_Com_00696, SWS_Com_00697, SWS_Com_00698, SWS_Com_00700, SWS_Com_00719, SWS_Com_00762, SWS_Com_00764, SWS_Com_00766, SWS_Com_00773, SWS_Com_00774, SWS_Com_00775, SWS_Com_00776, SWS_Com_00789, SWS_Com_00793, SWS_Com_00812, SWS_Com_00816, SWS_Com_00820, SWS_Com_00828, SWS_Com_00835, SWS_Com_00836, SWS_Com_00858, SWS_Com_00861, SWS_Com_00862, SWS_Com_00866, SWS_Com_00867, SWS_Com_00868
SRS_Com_02040	AUTOSAR COM and LargeDataCOM shall be configured by using XML as configuration language	SWS_Com_00006
SRS_Com_02041	The AUTOSAR module shall handle complex data types as a consistent set of data	SWS_Com_00050, SWS_Com_00051, SWS_Com_00199, SWS_Com_00200, SWS_Com_00201, SWS_Com_00202, SWS_Com_00461, SWS_Com_00513, SWS_Com_00632, SWS_Com_00634, SWS_Com_00635, SWS_Com_00637, SWS_Com_00638, SWS_Com_00639, SWS_Com_00640, SWS_Com_00676, SWS_Com_00677, SWS_Com_00678, SWS_Com_00679, SWS_Com_00821, SWS_Com_00851, SWS_Com_00852, SWS_Com_00854, SWS_Com_00855, SWS_Com_00857
SRS_Com_02042	The AUTOSAR COM module shall fill unused areas/ bits within an I-PDU with a configurable value	SWS_Com_00015
SRS_Com_02043	AUTOSAR COM and	SWS_Com_00123

	LargeDataCOM shall provide a receive indication function	
SRS_Com_02044	AUTOSAR COM and LargeDataCOM shall provide a transmit confirmation function	SWS_Com_00124
SRS_Com_02045	AUTOSAR COM and LargeDataCOM shall provide a function to request the transmit buffer data for lower layer triggered transmission	SWS_Com_00001, SWS_Com_00475, SWS_Com_00647, SWS_Com_00869
SRS_Com_02046	The AUTOSAR COM module shall support immediate and deferred signal based notification to the RTE	SWS_Com_00300, SWS_Com_00301, SWS_Com_00574, SWS_Com_00575, SWS_Com_00794
SRS_Com_02058	The AUTOSAR COM module shall support deadline monitoring for updated signals/signal groups on receiver side	SWS_Com_00117, SWS_Com_00290, SWS_Com_00291, SWS_Com_00292, SWS_Com_00715, SWS_Com_00716
SRS_Com_02067	AUTOSAR COM and LargeDataCOM shall define rules for checking the consistency of configuration data	SWS_Com_00102, SWS_Com_00105, SWS_Com_00310, SWS_Com_00319, SWS_Com_00365, SWS_Com_00373, SWS_Com_00384, SWS_Com_00401, SWS_Com_00402, SWS_Com_00443, SWS_Com_00465, SWS_Com_00535, SWS_Com_00553, SWS_Com_00732, SWS_Com_00785, SWS_Com_00790, SWS_Com_00817
SRS_Com_02077	The AUTOSAR COM module shall support invalidation of signals at sender side	SWS_Com_00099, SWS_Com_00203, SWS_Com_00286, SWS_Com_00288, SWS_Com_00557, SWS_Com_00642, SWS_Com_00643, SWS_Com_00644, SWS_Com_00645
SRS_Com_02078	The AUTOSAR COM module shall support endianness conversion	SWS_Com_00007, SWS_Com_00221, SWS_Com_00352, SWS_Com_00472, SWS_Com_00580, SWS_Com_00674, SWS_Com_00675, SWS_Com_00810
SRS_Com_02079	The AUTOSAR COM module shall support an optional notification when receiving invalidated data	SWS_Com_00536, SWS_Com_00680, SWS_Com_00681, SWS_Com_00682, SWS_Com_00717, SWS_Com_00718, SWS_Com_00859, SWS_Com_00860
SRS_Com_02080	The AUTOSAR COM module shall cancel outstanding repetitions in case of a new send request	SWS_Com_00279
SRS_Com_02082	The AUTOSAR COM module shall support defining two different transmission modes for each I-PDU	SWS_Com_00032, SWS_Com_00238, SWS_Com_00239, SWS_Com_00244, SWS_Com_00495, SWS_Com_00582, SWS_Com_00784, SWS_Com_00799
SRS_Com_02083	The AUTOSAR COM module shall support multiple	SWS_Com_00135, SWS_Com_00305, SWS_Com_00308, SWS_Com_00330,

	transmission modes	SWS_Com_00392, SWS_Com_00467, SWS_Com_00478, SWS_Com_00494, SWS_Com_00602, SWS_Com_00734, SWS_Com_00739, SWS_Com_00741, SWS_Com_00742, SWS_Com_00743, SWS_Com_00767, SWS_Com_00768, SWS_Com_00769, SWS_Com_00770
SRS_Com_02084	The AUTOSAR COM module shall support a configurable signal data based selection mechanism of the two transmission modes	SWS_Com_00032, SWS_Com_00245, SWS_Com_00605, SWS_Com_00677, SWS_Com_00678, SWS_Com_00679, SWS_Com_00763, SWS_Com_00799, SWS_COM_00813
SRS_Com_02086	The AUTOSAR COM module shall support sign-extension	SWS_Com_00008, SWS_Com_00352, SWS_Com_00579, SWS_Com_00723, SWS_Com_00829
SRS_Com_02087	The AUTOSAR COM module shall support an optional substitution of received invalidated data	SWS_Com_00470, SWS_Com_00500, SWS_Com_00681, SWS_Com_00683, SWS_Com_00859, SWS_Com_00860
SRS_Com_02088	The AUTOSAR COM module shall support substituting the last received value by the init value in case of a signal timeout	SWS_Com_00470
SRS_Com_02089	The AUTOSAR COM module shall provide two configurable options to handle signal timeouts	SWS_Com_00290, SWS_Com_00291, SWS_Com_00292, SWS_Com_00333, SWS_Com_00738, SWS_Com_00744
SRS_Com_02090	The AUTOSAR COM module shall define a data-structure allowing efficiently starting and stopping of I-PDU groups	SWS_Com_00623, SWS_Com_00749, SWS_Com_00750, SWS_Com_00823
SRS_Com_02091	AUTOSAR COM and LargeDataCOM shall not support splitting of large signals into different I-PDUs	SWS_Com_00754, SWS_Com_00755, SWS_Com_00756
SRS_Com_02092	The AUTOSAR COM module shall support at most one dynamic length signal per I-PDU	SWS_Com_00690, SWS_Com_00711, SWS_Com_00712, SWS_Com_00724, SWS_Com_00754
SRS_Com_02093	Dynamic length signal must be placed last in I-PDU	SWS_Com_00755, SWS_Com_00757, SWS_Com_00758, SWS_Com_00832
SRS_Com_02094	Dynamic length signals must be of type UINT8[n]	SWS_Com_00675, SWS_Com_00753
SRS_Com_02095	AUTOSAR COM and LargeDataCOM shall use the TP to fragment and reassemble large signals	SWS_Com_00627, SWS_Com_00628, SWS_Com_00629, SWS_Com_00630, SWS_Com_00650, SWS_Com_00654, SWS_Com_00655, SWS_Com_00656, SWS_Com_00657, SWS_Com_00662, SWS_Com_00690, SWS_Com_00691, SWS_Com_00692, SWS_Com_00693, SWS_Com_00720, SWS_Com_00721, SWS_Com_00725, SWS_Com_00818,

		SWS_Com_00838, SWS_Com_00839, SWS_Com_00863
SRS_Com_02096	The AUTOSAR COM module shall not support fragmentation towards the RTE	SWS_Com_00759, SWS_Com_00760
SRS_Com_02097	AUTOSAR COM and LargeDataCOM shall support dynamical signals with a static maximum length	SWS_Com_00756
SRS_Com_02098	The AUTOSAR COM module shall distinct normal and large signals via its configuration	SWS_Com_00753
SRS_Com_02099	The AUTOSAR COM module shall provide a mechanism to detect out of sequence received I-PDUs	SWS_Com_00587
SRS_Com_02101	The AUTOSAR COM module shall support the incrementing and checking of the I-PDU Counter	SWS_Com_00588, SWS_Com_00687, SWS_Com_00688
SRS_Com_02102	The AUTOSAR COM module shall support the detection of out of sequence I-PDUs	SWS_Com_00590, SWS_Com_00726, SWS_Com_00727, SWS_Com_00834
SRS_Com_02103	The AUTOSAR COM module shall provide a mechanism to detect corrupted received I-PDUs and to recover from this failure mode	SWS_Com_00596, SWS_Com_00597
SRS_Com_02105	The AUTOSAR COM module shall support transmission and reception of replicated I-PDUs	SWS_Com_00596, SWS_Com_00597
SRS_Com_02106	The AUTOSAR COM module shall support the detection of failures when receiving replicated I-PDUs	SWS_Com_00596, SWS_Com_00597
SRS_Com_02107	The AUTOSAR COM module shall cancel transmission requests in case of expired transmissions	SWS_Com_00708
SRS_Com_02112	AUTOSAR COM shall provide a uint8-array based API for signal groups	SWS_Com_00841, SWS_Com_00842, SWS_Com_00843, SWS_Com_00844, SWS_Com_00845, SWS_Com_00846, SWS_Com_00847, SWS_Com_00848, SWS_Com_00849, SWS_Com_00850, SWS_Com_00851, SWS_Com_00852, SWS_Com_00854, SWS_Com_00855
SRS_PduR_06002	The routing configuration shall be updateable at post-build time	SWS_Com_00357, SWS_Com_00361, SWS_Com_00373

SRS_PduR_06055	The signal gateway shall provide a mechanism to route individual signals between I-PDUs in a 1:n fashion	SWS_Com_00377, SWS_Com_00539, SWS_Com_00598
SRS_PduR_06056	Signal Groups shall be routed	SWS_Com_00361, SWS_Com_00383, SWS_Com_00735, SWS_Com_00833
SRS_PduR_06061	Routers shall map only signals	SWS_Com_00360, SWS_Com_00361, SWS_Com_00362
SRS_PduR_06064	The signal gateway shall be scalable to zero size and zero resource usage when signal routing is not required	SWS_Com_00370
SRS_PduR_06089	The timeout of a deadline monitored signal shall be ignored by the SigG	SWS_Com_00377, SWS_Com_00701
SRS_PduR_06097	A configuration shall be identified by an unique ID number	SWS_Com_00487
SRS_PduR_06098	Signal Gateway Error shall be handled with signal routing	SWS_Com_00442
SRS_PduR_06099	Signal Gateway Error shall be handled with signal group routing	SWS_Com_00442

## 7 Functional Specification

### 7.1 Introduction

### 7.2 General Functionality

#### 7.2.1 OSEK-COM

OSEK COM 3.0.3 is the functional basis of the AUTOSAR COM module.

**[SWS\_Com\_00010]** [The AUTOSAR COM module shall implement all the functionality and all the APIs of OSEK/ VDX Communication Version 3.0.3 [17] except the features and APIs mentioned in SWS\_Com\_00013.] (SRS\_Com\_02037)

**[SWS\_Com\_00011]** [If this AUTOSAR COM specification defines functionality in a different way compared to definitions in [17], the AUTOSAR COM module shall implement the functionality defined in this AUTOSAR COM specification.] (SRS\_Com\_02037)

**[SWS\_Com\_00012]** [The AUTOSAR COM module shall in addition implement all those features, that are defined in this AUTOSAR COM specification and that are not part of [17].] (SRS\_Com\_02037)

**[SWS\_Com\_00013]** [The AUTOSAR COM module may implement the following features of [17]. If they are implemented in a specific AUTOSAR COM module, the configuration shall disable them by default. This also applies for all other additional features a specific implementation may provide.] (SRS\_Com\_02037)

<i>OSEK-COM feature</i>	<i>Rationale</i>	<i>related OSEK COM API</i>
Mapping of a received network message (within an I-PDU) to more than one message data objects (1:n splitting mechanism)	not required, done by the RTE, see [13]	none
Mapping of an internal message to more than one message data objects (1:n splitting mechanism)	not required, done by the RTE, see [13]	none
Mapping an only locally send message to both an external send message object and an internal receive message object (1:n splitting mechanism)	not required, done by the RTE, see [13]	none
M:1 sending; mapping of messages from multiple senders to one and the same message object	not required, ensured by RTE, see [13]	SendMessage
Queued messages	not required, done by the RTE, see [13]	GetMessageStatus
Zero size messages	it is possible to set up communication without them	SendZeroMessage

<i>OSEK-COM feature</i>	<i>Rationale</i>	<i>related OSEK COM API</i>
	functionality is partly covered by Com_TriggerTransmit	
Notification mechanisms TASK, FLAG and EVENT	not required, done by the RTE, see [13]	none
Overlapping messages in an I-PDU	no use case, dangerous concept	none
Usage of OIL	The OSEK OIL is not used to configure the AUTOSAR COM module.	None
Application modes	not needed	GetComApplicationMode
Start-up behavior	replaced by <ul style="list-style-type: none"> <li>• Com_Init</li> <li>• Com_DeInit</li> <li>• Com_IpduGroupControl</li> </ul>	StartCOM, StopCOM, StartCOMExtensions, InitMessage
Start and stop of periodic messages	no use case, is realized by I-PDU group mechanism	StartPeriodic, StopPeriodic
Reentrancy	Not all of the AUTOSAR API calls are reentrant. See Chapter 8.3.	See Chapter 8.3.
Interface to OSEK indirect NM	not needed	I_MessageTransfer, I_MessageTimeout
Sender side filtering	no use case, the filter conditions are still used in the selection of the transmission mode but there is no signal filtering	none
Network-order message callout CPU-order message callout	Only I-PDU callouts with a defined AUTOSAR interface are supported by the AUTOSAR COM module. This is to avoid proprietary solutions.	None
Error hook routine	The AUTOSAR COM module will use a direct interface to DEM/DET instead of using the OSEK COM error hook	COMErrorHook COMError_Name1_Name2 macros COMErrorGetServiceId
Interface for callback routines	The signatures for the used callback function of the AUTOSAR COM module will be explicitly defined within the AUTOSAR COM module's specification.	COMCallback
Internal communication	not required, ensured by RTE, see [13]	SendMessage, ReceiveMessage

**Table 1: Excluded OSEK COM features in the AUTOSAR COM module**

## 7.2.2 Signal Values

The signals sent by the AUTOSAR COM module respectively received by the AUTOSAR COM module could have the values defined in Table 2.

<i>Signal value</i>	<i>Remark</i>
init value	See Chapter 7.3.1.4 for details.
Data invalid value	See Chapter 7.3.4 for details.

<i>Signal value</i>	<i>Remark</i>
<value>	This is the normal case: A valid value after initialization phase, which is sent by the AUTOSAR COM module respectively, received by AUTOSAR COM module.

**Table 2: Possible signal values**

### 7.2.3 Endianness Conversion and Sign Extension

**[SWS\_Com\_00675]** [The AUTOSAR COM module shall support the following data types:

- boolean
- uint8
- uint16
- uint32
- uint64
- sint8
- sint16
- sint32
- sint64
- uint8[n]
- float32
- float64

The type uint8[n] is mapped to either ComSignalType *UINT8\_N* or *UINT8\_DYN*.] (SRS\_Com\_02078, SRS\_Com\_02094)

**[SWS\_Com\_00007]** [The AUTOSAR COM module shall support endianness conversion for all supported signed and unsigned integer data types (see **SWS\_Com\_00675**).] (SRS\_Com\_02078)

**[SWS\_Com\_00810]** [The AUTOSAR COM module shall treat non-integer data types (e.g. uint8[n] or float32) either like integer data types of the matching size or leave their contents uninterpreted in case their ComSignalEndianness is configured to *OPAQUE*.] (SRS\_Com\_02078)

**[SWS\_Com\_00472]** [The AUTOSAR COM module shall interpret opaque data as uint8[n] and shall always map it to an n-bytes sized signal.] (SRS\_Com\_02078)

For opaque data endianness, conversion has to be configured to *OPAQUE* (see ECUC\_Com\_00157).

**[SWS\_Com\_00674]** [The AUTOSAR COM module shall extend the endianness conversion defined in [17] Chapter 2.4 to signed data types.] (SRS\_Com\_02078)

In [17] Chapter 2.4 defines the endianness conversion for unsigned data types. The associated configurations can be found in Chapter 10. See also ECUC\_Com\_00127 and ECUC\_Com\_00157.



**[SWS\_Com\_00829]** [The AUTOSAR COM module shall extend the endianness conversion defined in [17] Chapter 2.4 to signals of signal groups.] (SRS\_Com\_02086)

AUTOSAR COM handles signals and group signals equally with respect to endianness conversion.

**[SWS\_Com\_00008]** [The AUTOSAR COM module shall extend received data to the size of the ComSignalType of the receive signal (sign extension).] (SRS\_Com\_02086)

The platform specific representation of signed data has to be taken into account. Negative values of signed data will be mapped correctly.

**Example:** A 10-Bit signed signal is received and copied by Com\_ReceiveSignal to a 16-Bit signed integer variable. If  $(-3)_{\text{decimal}}$  is received, the received 10-Bit signal has a value of 1111111101b. While copying it to the 16-Bit integer variable the value will be extended to 1111111111111101b.

**[SWS\_Com\_00723]** [The AUTOSAR COM module shall extend the init value (ComSignalInitValue) of a signal to the size of its ComSignalType.] (SRS\_Com\_02086)

Note that the AUTOSAR COM module will not need to perform sign extensions on sender side as it is guaranteed that the values to be transmitted will always be sign-extended. See also SWS\_Com\_00785.

**[SWS\_Com\_00579]** [The AUTOSAR COM module shall not support sign extension for float32 and float64 data types.] (SRS\_Com\_02086)

**[SWS\_Com\_00221]** [The AUTOSAR COM module shall perform endianness conversion before the I-PDU callout on sender side. For an overview, see Chapter 7.15.] (SRS\_Com\_02078)

**[SWS\_Com\_00352]** [The AUTOSAR COM module shall perform sign extensions and endianness conversion before performing filtering and notification detection on receiver side.] (SRS\_Com\_02078, SRS\_Com\_02086)

**[SWS\_Com\_00580]** [Beside endianness conversion, the AUTOSAR COM module shall not support further conversions for signals with ComSignalType FLOAT32 or FLOAT64. That is endianness conversion shall be supported but complex conversion or normalizations of fractions, exponents, signs or bias values shall not be supported.] (SRS\_Com\_02078)

## 7.2.4 Filtering

**[SWS\_Com\_00694]** [The AUTOSAR COM module shall evaluate each filtering condition to either true or false.] (SRS\_Com\_02037)

**[SWS\_Com\_00695]** [The AUTOSAR COM module shall filter out signals only at receiver side.] (SRS\_Com\_02037)

**[SWS\_Com\_00602]** [The AUTOSAR COM module shall use filtering mechanisms on sender side for Transmission Mode Conditions (TMC) but it shall not filter out signals on sender side.] (SRS\_Com\_02083)

For Transmission Mode Selection (TMS) see Chapters 7.3.3.2 and 7.3.3.3.

The AUTOSAR COM module only provides the following ComFilterAlgorithms of that which are defined in [17], see ECUC\_Com\_00146:

- *ALWAYS*
- *NEVER*
- *MASKED\_NEW\_EQUALS\_X*
- *MASKED\_NEW\_DIFFERS\_X*
- *MASKED\_NEW\_DIFFERS\_MASKED\_OLD*
- *NEW\_IS\_WITHIN*
- *NEW\_IS\_OUTSIDE*
- *ONE EVERY\_N*

To reduce complexity the AUTOSAR COM module does not support all filters defined in [17]. The not supported filters are either obsolete or special cases of other filters. For example, the filter *NEW\_IS\_DIFFERENT* is a special case of *MASKED\_NEW\_DIFFERS\_MASKED\_OLD* with a fully set mask.

**[SWS\_Com\_00325]** [The AUTOSAR COM module shall support all filter mechanisms listed in ECUC\_Com\_00146, considering the exceptions defined in SWS\_Com\_00380 and SWS\_Com\_00439.] (SRS\_Com\_02037)

**[SWS\_Com\_00380]** [For signals with ComSignalType *UINT8\_N* or *UINT8\_DYN*, the AUTOSAR COM module shall only support ComFilterAlgorithm configured to *ALWAYS* or *NEVER*.] (SRS\_Com\_02037)

**[SWS\_Com\_00439]** [For signals with ComSignalType configured to *BOOLEAN*, the AUTOSAR COM module shall only support ComFilterAlgorithm configured to:

- *ALWAYS*
- *NEVER*
- *MASKED\_NEW\_EQUALS\_X*
- *MASKED\_NEW\_DIFFERS\_X*
- *MASKED\_NEW\_DIFFERS\_MASKED\_OLD*
- *ONE EVERY\_N*

] (SRS\_Com\_02037)

**[SWS\_Com\_00764]** [For signals and group signals with ComBitSize configured to 0, the AUTOSAR COM module shall not support the filter algorithm *Masked\_New\_Differs\_Masked\_Old*.] (SRS\_Com\_02037)

**[SWS\_Com\_00866]** [In case the configured filter condition is *MASKED\_NEW\_DIFFERS\_X*, *MASKED\_NEW\_EQUALS\_X*, *NEW\_IS\_OUTSIDE* or *NEW\_IS\_*

WITHIN, the AUTOSAR COM module shall calculate the reception filter by using only the least significant ComBitSize bits of the filter parameters.] (SRS\_Com\_02037)

**[SWS\_Com\_00273]** [If the AUTOSAR COM module filters out a signal on receiver side, i.e. filter condition evaluates to false, the AUTOSAR COM module shall discard that signal and shall not process it. See also SWS\_Com\_00303.] (SRS\_Com\_02037)

**[SWS\_Com\_00836]** [If the AUTOSAR COM module filters out a group signal on receiver side, i.e. filter condition evaluates to false, the AUTOSAR COM module shall discard the whole signal group and shall not process it.] (SRS\_Com\_02037)

**[SWS\_Com\_00132]** [The AUTOSAR COM module shall support the filtering mechanisms as defined in ECUC\_Com\_00146 also for signed data types.] (SRS\_Com\_02037)

In the case a filter is evaluated before a send-API has written the corresponding signal, there needs to be a way to determine the filter state of this signal. Some of the filters require a *new\_value* to evaluate the filter. However, this is only available after the signal has been updated using a send-API. Therefore, it is necessary to define the value used by the filter for *new\_value* in the period before the first send takes place.

**[SWS\_Com\_00603]** [The AUTOSAR COM module shall set the *old\_value* of the filtering mechanisms for each signal to the ComSignalInitValue (ECUC\_Com\_00170) during start-up. See also [17].] (SRS\_Com\_02037)

**[SWS\_Com\_00604]** [Until the application has not updated the *new\_value* of the filtering mechanisms for a signal, the AUTOSAR COM module shall use the ComSignalInitValue as the *new\_value* for that signal.] (SRS\_Com\_02037)

The next two requirements clarify the definitions of [17] according to the update of the *old\_value* of filters.

**[SWS\_Com\_00302]** [If the AUTOSAR COM module evaluates a filter for a signal to true, (value is not filtered out) then the AUTOSAR COM module shall place the value of that signal into *old\_value* (as defined in [17]).] (SRS\_Com\_02037)

**[SWS\_Com\_00303]** [When a value is being filtered, if the filter does not allow the passage of the value (i.e. the filter evaluates to false) then the AUTOSAR COM module shall not place that value into *old\_value* (as defined in [17]).] (SRS\_Com\_02037)

**[SWS\_Com\_00231]** [In the case of ComFilterAlgorithm is configured to *ONE EVERY N*, the AUTOSAR COM module shall

- set OCCURRENCE to zero when OCCURRENCE == PERIOD
- set FILTER to true, when OCCURRENCE == OFFSET
- increment OCCURRENCE after filter processing

] (SRS\_Com\_02037)

For definition of OCCURRENCE, FILTER, OFFSET and PERIOD see [17].

Configuring ComFilterAlgorithm to *ONE EVERY N* for a signal has the effect that the signal is passed by the filter (i.e. the filter returns true) once every PERIOD calls of the filter. If the OFFSET parameter is zero then the first time the filter is used the signal is allowed to pass (i.e. filter returns true). If the OFFSET is greater than zero then more than one signal must pass through the filter before it returns true.

In case the I-PDU is started by Com\_IpduGroupControl with parameter initialize set to true, the OCCURRENCE is also set to zero, see SWS\_Com\_00787.

This definition exists to clarify the description of the *ONE EVERY N* filter in [17].

The associated configuration items can be found in Chapter 10, see ECUC\_Com\_00339.

**[SWS\_Com\_00793]** [For a signal with a configured reception filter MASKED\_NEW\_DIFFERS\_MASKED\_OLD, the AUTOSAR COM module shall treat the first value received for this signal after a reception deadline monitoring timeout occurred for this signal the same way as if the value has passed the filter criteria.] (SRS\_Com\_02037)

Hence, the AUTOSAR COM module will let pass any value for the filter MASKED\_NEW\_DIFFERS\_MASKED\_OLD after an RX deadline timeout for the associated I-PDU.

### 7.2.5 Signal Gateway

The AUTOSAR COM module provides an integrated Signal Gateway for forwarding signals and signal groups in a 1:n manner.

The routing relations are be statically configured via the ComGwMapping configuration container (see ECUC\_Com\_00544).

As shown in Figure 6 and Figure 7, the integrated Signal Gateway acts as a receiver for all signals or signal groups which are configured as a source for gatewayed signals.

After the Signal Gateway received signal or signal groups for routing, it acts immediately as a sender for these signals or signal groups respectively. This is shown in Figure 5 and Figure 7. The signal processing does not differ if the integrated Signal Gateway forwards a signal/ signal group or if a Software Component sends it.

Since the gateway relations are statically configured, an optimized Signal Gateway might skip some processing stages for concrete configurations. For example, the endianness conversion is not necessary in certain cases. Of course, such optimizations should not lead to any different logical behavior or different representations on the target bus. Optimizations may only be introduced to increase the Signal Gateway's performance.

As shown in Figure 6 a received signal or signal group might be received on the local ECU and be a source for a gateway action at the same time.

Figure 5 and Figure 6 show shadow buffers only for signal groups that are received by the RTE. The integrated Signal Gateway has of course to ensure to handle signal groups consistently but there is no predetermined way in which this must be implemented.

**[SWS\_Com\_00370]** [The Signal Gateway of the AUTOSAR COM module shall scale down to no size if no signal routing functionality is needed.] (SRS\_PduR\_06064)

## 7.3 Normal Operation

### 7.3.1 Start-Up Behavior

This chapter describes the actions that will be performed during Com\_Init.

**[SWS\_Com\_00217]** [The AUTOSAR COM module shall initialize each I-PDU during execution of Com\_Init (SWS\_Com\_00432), firstly byte wise with the ComTxIPduUnusedAreasDefault value and then bit wise according to initial values (ComSignalInitValue) of the contained signals and the update-bits (see SWS\_Com\_00117).] (SRS\_BSW\_00101)

#### 7.3.1.1 Preconditions

The C initialization code, also known as *start-up code*, initializes global and static variables with the initial values. It must be executed before any call of an AUTOSAR COM module's service.

#### 7.3.1.2 Initialization

**[SWS\_Com\_00128]** [The AUTOSAR COM module's initialization function Com\_Init (SWS\_Com\_00432) shall initialize all internal data that is not yet initialized by the *start-up code* e.g. C-structs.] (SRS\_BSW\_00101)

**[SWS\_Com\_00328]** [The AUTOSAR COM module's initialization function Com\_Init (SWS\_Com\_00432) shall not enable Inter-ECU communication.] (SRS\_BSW\_00101)

This initialization chapter is not complete. Details about initialization of some AUTOSAR COM module's features are described within the different feature chapters.

#### 7.3.1.3 Initialization of Not Used Areas of an I-PDU

**[SWS\_Com\_00015]** [The AUTOSAR COM module shall fill not used areas within an I-PDU with a value determined by configuration parameter ComTxIPduUnusedAreasDefault (ECUC\_Com\_00017) e.g. 0xFF.] (SRS\_BSW\_00101, SRS\_Com\_02042)

#### 7.3.1.4 Initialization of Signals and Update-Bits

**[SWS\_Com\_00098]** [The AUTOSAR COM module shall initialize each signal of n-bit sized signal type on sender and receiver side with the lower n-bits of its configuration parameter ComSignalInitValue (ECUC\_Com\_00170).] (SRS\_BSW\_00101)

The configured ComSignalInitValues (ECUC\_Com\_00170) are also used for the initialization of the signal in the related I-PDU; see SWS\_Com\_00217.

The ComSignalInitValue (ECUC\_Com\_00170) of a signal can be identical to its ComSignalDataInvalidValue (ECUC\_Com\_00391). These can be different for each signal.

**[SWS\_Com\_00117]** [The AUTOSAR COM module shall clear all update-bits during initialization. See also SWS\_Com\_00059.] (SRS\_BSW\_00101, SRS\_Com\_02030, SRS\_Com\_02058)

#### 7.3.1.5 Initialization of I-PDU Groups

**[SWS\_Com\_00444]** [By default, all I-PDU groups shall be in the state stopped and they shall not be started automatically by a call to Com\_Init (SWS\_Com\_00432).] (SRS\_Com\_00218)

### 7.3.2 De-Initialization

The AUTOSAR COM module provides the API function Com\_DeInit (SWS\_Com\_00130) for de-initialization of the COM layer. This means, after de-initialization of the layer, no communication via the AUTOSAR COM module is possible and all started I-PDU groups are stopped, see also SWS\_Com\_00129.

### 7.3.3 Communication Modes

This chapter defines the signal flow in the AUTOSAR COM module. It further defines the different transmission modes provided by the AUTOSAR COM module. Chapter 7.3.3.2 defines a mechanism to switch between two transmission modes for one I-PDU. The replication of signals is defined in Chapter 7.3.3.4.

Exemplary communication use cases that the AUTOSAR COM module can deal with are shown in Chapter 12.

### 7.3.3.1 Transfer Properties and I-PDU Transmission Mode

#### 7.3.3.1.1 Signals

The AUTOSAR COM module supports several *transfer properties* for signals and several *transmission modes* for I-PDUs. The definitions in this chapters are based on and are to be completed with the definitions in [17] Chapter 2.3.3.

**[SWS\_Com\_00330]** [At any send request of a signal with ComTransferProperty *TRIGGERED* assigned to an I-PDU with ComTxModeMode *DIRECT* or *MIXED*, the AUTOSAR COM module shall immediately (within the next main function at the latest) initiate ComTxModeNumberOfRepetitions plus one transmissions of the assigned I-PDU.] (SRS\_Com\_02083)

**[SWS\_Com\_00767]** [At any send request of a signal with ComTransferProperty *TRIGGERED\_WITHOUT\_REPETITION* assigned to an I-PDU with ComTxModeMode *DIRECT* or *MIXED*, the AUTOSAR COM module shall immediately (within the next main function at the latest) initiate one transmission of the assigned I-PDU.] (SRS\_Com\_02083)

**[SWS\_Com\_00734]** [At a send request of a signal with ComTransferProperty *TRIGGERED\_ON\_CHANGE* assigned to an I-PDU with ComTxModeMode *DIRECT* or *MIXED*, the AUTOSAR COM module shall immediately (within the next main function at the latest) initiate ComTxModeNumberOfRepetitions plus one transmissions of the assigned I-PDU, if the new sent signal differs to the locally stored (last sent or init) in length or value.] (SRS\_Com\_02083)

**[SWS\_Com\_00768]** [At a send request of a signal with ComTransferProperty *TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION* assigned to an I-PDU with ComTxModeMode *DIRECT* or *MIXED*, the AUTOSAR COM module shall immediately (within the next main function at the latest) initiate one transmission of the assigned I-PDU, if the new sent signal differs to the locally stored (last sent or init) in length or value.] (SRS\_Com\_02083)

**[SWS\_Com\_00762]** [The AUTOSAR COM module shall not support the transfer properties *TRIGGERED\_ON\_CHANGE* and *TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION* for signals and group signals with ComBitSize configured to 0.] (SRS\_Com\_02037)

The support of the transfer properties *TRIGGERED\_ON\_CHANGE* and *TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION* is not restricted to certain signal types. Hence, they are supported for all possible signal types. This includes even the byte array types, e.g. *UINT8\_N*.

The details of the transmission replication mechanism are specified in Chapter 7.3.3.4.

**[SWS\_Com\_00135]** [The AUTOSAR COM module shall not initiate transmissions for I-PDUs that have the ComTxModeMode (ECUC\_Com\_00137) *NONE*.] (SRS\_Com\_02083)

It is possible to request I-PDUs with ComTxModeMode *NONE* via Com\_TriggerTransmit.

A pending signal associated with an I-PDU is transmitted if the I-PDU's transmission is triggered for any reason. For example, if a signal with ComTransferProperty *TRIGGERED* within the same I-PDU is sent or the I-PDU sending is scheduled because of ComTxModeMode *PERIODIC* or *MIXED*.

The bus-timing of I-PDUs can be controlled by send requests of the RTE in combination with the transmission mode and the transfer property as described above. Additionally, the PDU Router, especially in case of FlexRay and LIN, can control it with the service Com\_TriggerTransmit. In the latter case, the PDU Router requests I-PDUs to be sent from the AUTOSAR COM module.

The function Com\_TriggerTransmit can be called for any I-PDU regardless of its transmission mode. This allows LIN and FlexRay to use all the available transmission modes, particularly for sporadic communication. The NM also uses this mechanism to send user data.

#### 7.3.3.1.2 Signal Groups

In AUTOSAR COM also signal groups and group signals may have a transfer property, defining in combination with the transmission mode, if the I-PDU is sent out in case of an update of a signal group or group signal, respectively.

**[SWS\_Com\_00741]** [At any send request of a signal group with ComTransferProperty *TRIGGERED* assigned to an I-PDU with ComTxModeMode *DIRECT* or *MIXED*, the AUTOSAR COM module shall immediately (within the next main function at the latest) initiate ComTxModeNumberOfRepetitions plus one transmissions of the assigned I-PDU.] (SRS\_Com\_02083)

**[SWS\_Com\_00769]** [At any send request of a signal group with ComTransferProperty *TRIGGERED\_WITHOUT\_REPETITION* assigned to an I-PDU with ComTxModeMode *DIRECT* or *MIXED*, the AUTOSAR COM module shall initiate one transmission of the assigned I-PDU within the next main function at the latest.] (SRS\_Com\_02083)

**[SWS\_Com\_00742]** [Regarding signal groups with ComTransferProperty *TRIGGERED\_ON\_CHANGE* which do not contain any signals that have an own ComTransferProperty configured:

At any send request of such a signal group assigned to an I-PDU with ComTxModeMode *DIRECT* or *MIXED*, the AUTOSAR COM module shall immediately (within the next main function at the latest) initiate ComTxModeNumberOfRepetitions plus one transmissions of the assigned I-PDU, if at least one new sent group signal differs to the locally stored (last sent or init) in length or value.] (SRS\_Com\_02083)



**[SWS\_Com\_00743]** [Regarding signal groups with ComTransferProperty *TRIGGERED\_ON\_CHANGE* which contain any signals that have an own ComTransferProperty configured:

At any send request of such a signal group assigned to an I-PDU with ComTxModeMode *DIRECT* or *MIXED*, the AUTOSAR COM module shall immediately (within the next main function at the latest) initiate ComTxModeNumberOfRepetitions plus one transmissions of the assigned I-PDU, if at least one new sent group signal configured with ComTransferProperty *TRIGGERED\_ON\_CHANGE* differs to the locally stored (last sent or init) in length or value.] (SRS\_Com\_02083)

**[SWS\_Com\_00770]** [At a send request of a signal group with ComTransferProperty *TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION* assigned to an I-PDU with ComTxModeMode *DIRECT* or *MIXED*, the AUTOSAR COM module shall immediately (within the next main function at the latest) initiate one transmission of the assigned I-PDU, if at least one new sent group signal differs to the locally stored (last sent or init) in length or value.] (SRS\_Com\_02083)

### 7.3.3.2 Selection of the Transmission Mode for one specific I-PDU

I-PDUs carry signals. Because an I-PDU can contain more than one signal, in the following, a method is defined to derive the I-PDU's transmission mode from the state of the signals that are contained in one specific I-PDU.

The AUTOSAR COM module allows configuring statically two different transmission modes for each I-PDU (see SWS\_Com\_00032). The transmission mode of an I-PDU that is valid at a specific point in time is selected using only the values of the signals that are mapped to this I-PDU.

The signals of one I-PDU that contribute to the selection of one of the two transmission modes as well as the conditions used for the selection of the transmission mode are configured statically, see SWS\_Com\_00676.

For the selection of the transmission mode, the AUTOSAR COM module treats group signals like normal signals.

**[SWS\_Com\_00676]** [For the evaluation of a TMS, the AUTOSAR COM module shall take all those (group) signals into account which configuration include a configured ComFilter container. See included containers of ComSignal and ComGroupSignal.] (SRS\_Com\_02041)

If a signal of an I-PDU with ComIPduDirection configured to *SEND* has a configuration container ComFilter included, the signal is named to **contribute** to the TMS of this I-PDU. If the configuration container for that signal has no assigned configuration container ComFilter, then it does **not contribute** to a TMS.

**[SWS\_Com\_00677]** [If no signal within an I-PDU contributes to the calculation of the TMS, then the AUTOSAR COM module shall evaluate the TMS of this I-PDU as true.] (SRS\_Com\_02084, SRS\_Com\_02041)

**[SWS\_Com\_00678]** [If the AUTOSAR COM module evaluates the TMC of a contributing signal as true, then the AUTOSAR COM module shall evaluate the TMS as true.] (SRS\_Com\_02084, SRS\_Com\_02041)

**[SWS\_Com\_00679]** [If the AUTOSAR COM module evaluates no TMC of a contributing signal as true, then the AUTOSAR COM module shall evaluate the TMS as false.] (SRS\_Com\_02084, SRS\_Com\_02041)

**[SWS\_Com\_00605]** [The AUTOSAR COM module shall define a Transmission Mode Selector, for each I-PDU. See definition of TMS above.] (SRS\_Com\_02084)

**[SWS\_Com\_00245]** [The AUTOSAR COM module shall re-calculate the TMS for an I-PDU, after a contained signal was updated by a call to Com\_SendSignal or Com\_SendSignalGroup.] (SRS\_Com\_02084)

**[SWS\_Com\_00763]** [For the calculation of the transmission mode with the configured condition *MASKED\_NEW\_DIFFERS\_MASKED\_OLD*, the AUTOSAR COM module shall use the least significant ComBitSize bits only.] (SRS\_Com\_02084)

**[SWS\_COM\_00813]** [For the calculation of the transmission mode with a configured condition *MASKED\_NEW\_DIFFERS\_X*, *MASKED\_NEW\_EQUALS\_X*, *NEW\_IS\_OUTSIDE* or *NEW\_IS\_WITHIN*, the AUTOSAR COM module shall use all bits of the configured ComSignalType.] (SRS\_Com\_02084)

Note that a signal with ComFilterAlgorithm configured to *ALWAYS*, will always set the TMS of the respective I-PDU to true. Therefore, care must be taken when defining the signals that contribute to the TMS.

**[SWS\_Com\_00032]** [If the TMS of an I-PDU evaluates to true, then the AUTOSAR COM module shall use the ComTxModeMode defined in configuration container ComTxModeTrue (ECUC\_Com\_00455) for that I-PDU.] (SRS\_Com\_02082, SRS\_Com\_02084)

**[SWS\_Com\_00799]** [If the TMS for an I-PDU evaluates to false, then the AUTOSAR COM module shall use the ComTxModeMode defined in configuration container ComTxModeFalse (ECUC\_Com\_00454) for that I-PDU.] (SRS\_Com\_02082, SRS\_Com\_02084)

**[SWS\_Com\_00238]** [In each of the two TMS states, the rules for combination of transfer properties of signals and transmission modes of I-PDUs shall apply as defined in [17] Section 2.3.] (SRS\_Com\_02082)

**[SWS\_Com\_00239]** [When the TMS state of an I-PDU changes, the AUTOSAR COM module shall use the now valid transmission mode immediately (within the next main function at the latest). That means, first the AUTOSAR COM module shall perform the mode change and after that, the AUTOSAR COM module shall execute any resulting calls to PduR\_ComTransmit caused by mode change.] (SRS\_Com\_02082)

**[SWS\_Com\_00244]** [If a change of the TMS causes a change of the transmission mode for an I-PDU, then the AUTOSAR COM module shall restart the timer for the cycle time of the transmission mode PERIODIC and MIXED.] (SRS\_Com\_02082)

**[SWS\_Com\_00495]** [When a call to Com\_SendSignal or Com\_SendSignalGroup results into a change of the transmission mode of a started I-PDU to the transmission mode PERIODIC or MIXED, then the AUTOSAR COM module shall start the new transmission cycle with a call to PduR\_ComTransmit within the next main function at the latest. The transmission shall be initiated regardless of the transfer property of the signal or signal group that caused the transmission mode switch. The minimum delay time and ComTxModeTimeOffset shall still be respected. See also Figure 6 The AUTOSAR COM module's interaction model for reception] (SRS\_Com\_02082)

**[SWS\_Com\_00582]** [If a change of the TMS causes a change to the transmission mode *DIRECT*, an immediate (within the next main function at the latest unless shifted due to the MDT) direct/ n-times transmission to the underlying layer shall be initiated.] (SRS\_Com\_02082)

The above requirement clarifies the behavior in case the TMS-switch to ComTxModeMode *DIRECT* was triggered by a signal with ComTransferProperty *PENDING*.

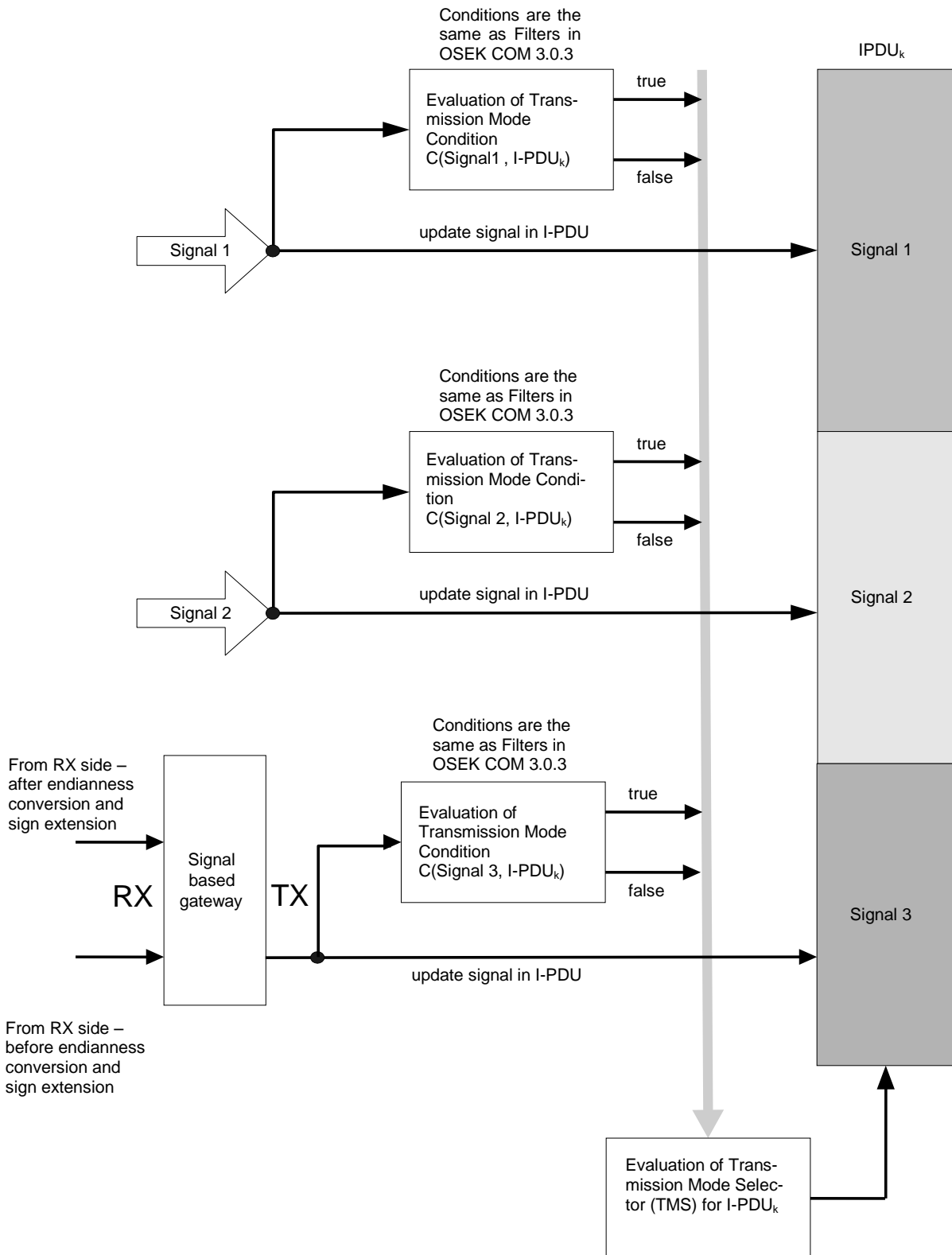
If the ComTxModeMode of an I-PDU is configured to *NONE* no transmission will be initiated by AUTOSAR COM. This can be used to prevent transmitting an I-PDU, for example in case the TMS evaluates to false.

**[SWS\_Com\_00478]** [The AUTOSAR COM module shall send out an I-PDU at most once within one call of Com\_MainFunctionTx.] (SRS\_Com\_02083)

### 7.3.3.3 Signal Flow and Transmission Mode Selection

After a send request from the RTE for a specific signal, the signal is written to the appropriate I-PDU buffer as defined by configuration, and the selection of the transmission mode of the I-PDUs is done according to Chapter 7.3.3.2.

Figure 3 shows the signal flow:



**Figure 3: Logical signal flow in the AUTOSAR COM module shown for two signals (Signal1 and Signal2) that are mapped to one I-PDU (IPDU<sub>k</sub>)**

### 7.3.3.4 Replication of Signal Transmission Requests

The number of repetitions of transmission requests in the ComTxModeMode *DIRECT* or *MIXED* for a send request by the RTE is defined by configuration parameter ComTxModeNumberOfRepetitions (ECUC\_Com\_00281).

**[SWS\_Com\_00467]** [If ComRetryFailedTransmitRequests is not set to TRUE and an I-PDU with ComTxModeMode *DIRECT* or *MIXED* and ComTxModeNumberOfRepetitions set to 0 is triggered for sending, the AUTOSAR COM module shall invoke PduR\_ComTransmit for this I-PDU just once, independently of the result of the confirmation.] (SRS\_Com\_02083)

Configuring ComTxModeNumberOfRepetitions to 0 imitates the original OSEK direct transmission mode.

**[SWS\_Com\_00279]** [If a new send request is received from the RTE while sending *n* transmissions belonging together (e.g. after the 3<sup>rd</sup> of 5 repetitions, see SWS\_Com\_00305) the AUTOSAR COM module shall cancel the outstanding transmission repetitions and start processing the new request within the next main function at the latest, see Figure 4.] (SRS\_Com\_02080)

**[SWS\_Com\_00305]** [The AUTOSAR COM module shall perform the transmission to PduR and confirmation behavior to RTE in the ComTxModeMode *DIRECT* or *MIXED* with ComTxModeNumberOfRepetitions strictly greater than 0 (see ECUC\_Com\_00281) according to the following steps:

- 1) When an I-PDU is sent by using Com\_SendSignal or Com\_SendSignalGroup, the ComTxModeMode is set to *DIRECT* or *MIXED*, and the ComTransferProperty is set to *TRIGGERED*, the AUTOSAR COM module shall set a counter assigned to that I-PDU to ComTxModeNumberOfRepetitions plus one.
- 2) The AUTOSAR COM module shall call PduR\_ComTransmit periodically, with period ComTxModeRepetitionPeriod, as long as the counter is non-zero.
- 3) Whenever a TX confirmation is received and the counter is greater than 0, the AUTOSAR COM module shall decrement the counter. When the counter is 0, the AUTOSAR COM module shall ignore transmission confirmations for that I-PDU.
- 4) When the counter reaches 0, the AUTOSAR COM module shall send the transmission confirmation to the RTE and the AUTOSAR COM module shall cancel transmission deadline monitoring (if configured by ECUC\_Com\_00263), see SWS\_Com\_00392 and Chapter 7.3.6.2.] (SRS\_Com\_02083)

The definition in SWS\_Com\_00305 should not define a concrete implementation. However, every implementation has to implement the confirmation behavior according to the above definition.

This solution allows the violation of the period in certain extreme circumstances when the confirmations arrive late in the period. This solution requires that CAN does not have a queue for these L-PDUs. There is a race condition in the interaction between the CAN driver, interface and hardware that may cause an extra transmission to occur in certain unlikely circumstances.

If the underlying layer returns `E_NOT_OK` while an N-Times transmission is in progress, this error notification will be ignored, unless `ComRetryFailedTransmitRequests` is set to true (see also `SWS_Com_00773`). As `SWS_Com_00305` specifies, only confirmed transmissions are counted for the N-Times transmission, erroneous send request can safely be ignored.

If the N-Times transmission is requested in transmission mode *MIXED* after a cyclic transmission of the *MIXED* transmission mode with a pending confirmation, the confirmation of the cyclic transmission will be assigned to the N-Times transmission. In this case, only n-1 transmissions of the new value of the N-Times request are made, if no confirmation gets lost. The transmission deadline monitoring timer will then be reset earliest after the N-Times request is completed. This must be respected when configuring the transmission deadline monitoring timer in conjunction with the *MIXED* transmission mode and N-Times transmission.

**[SWS\_Com\_00494]** [If within the transmission mode *MIXED* an N-Times transmission request overlaps with the cyclic part of the mixed transmission the cyclic transmission shall be counted as the corresponding transmission of the N-Times transmission request.] (SRS\_Com\_02083)

**[SWS\_Com\_00392]** [If a transmission deadline monitoring timeout occurs before the N-Times transmission is complete, then the AUTOSAR COM module shall not initiate further transmissions for this N-Times transmission.] (SRS\_Com\_02083)

The minimum delay time will always be taken into account as defined in [17] Chapter 2.3.4.

To avoid bursts in start-up a time offset can be configured per I-PDU. See `ECUC_Com_00180` for details.

The time between two repetitions is configured by configuration parameter `ComTxModeRepetitionPeriod` (`ECUC_Com_00282`).

If the transmission mode change leads to the start of the *MIXED* transmission mode by sending a triggered signal and `ComTxModeNumberOfRepetitions` is configured greater than or equal to 1, then there will be at least `ComTxModeNumberOfRepetitions` plus one transmission requests to the PDU Router at the beginning of the *MIXED* transmission mode. See also `SWS_Com_00305`.

### 7.3.3.5 Retry Failed Transmission Requests

**[SWS\_Com\_00773]** [If `ComRetryFailedTransmitRequests` is set to `TRUE`, the return value of `PduR_ComTransmit` shall be evaluated. If the return value of `PduR_ComTransmit` is not equal to `E_OK`, the AUTOSAR COM Module shall invoke `PduR_ComTransmit` for the not sent I-PDU again within the next `Com_MainFunctionTx`.] (SRS\_Com\_02037)

If the I-PDU is updated in between, the new data will be sent.

SWS\_Com\_00773 may lead to multiple retries for the same failed transmission request.

SWS\_Com\_00773 is not affected by transmission mode changes.

SWS\_Com\_00773 has no influence on a potential repetition period. Any repetitions will occur as if no retry had taken place.

**[SWS\_Com\_00774]** [The AUTOSAR COM module shall start the transmission deadline monitoring independently of possible retries. Subsequent retries shall not affect transmission deadline monitoring.] (SRS\_Com\_02037)

**[SWS\_Com\_00775]** [If ComRetryFailedTransmitRequests is set to True and a transmission deadline monitoring timeout occurs for an I-PDU, the AUTOSAR COM module shall expire any pending transmission request for this I-PDU.] (SRS\_Com\_02037)

**[SWS\_Com\_00776]** [The cycle timer for a cyclic transmission shall always start with the first transmit attempt.] (SRS\_Com\_02037)

### 7.3.4 Signal Invalidation

The AUTOSAR COM module provides the possibility for the sender to indicate that it is not able to provide a valid value for a corresponding signal, for example in case a sensor is faulty. The AUTOSAR COM module allows defining a ComSignalDataInvalidValue (ECUC\_Com\_00391) during configuration.

#### 7.3.4.1 Transmission of an Invalidated Signal

**[SWS\_Com\_00099]** [By a call to Com\_InvalidateSignal, the AUTOSAR COM module shall perform internally a Com\_SendSignal with the configured ComSignalDataInvalidValue (ECUC\_Com\_00391).] (SRS\_Com\_02077)

The ComTransferProperty and the transmission mode determine the transmission of the ComSignalDataInvalidValue on the bus. The internally performed Com\_SendSignal with the data invalid value leads to data invalid value to be used as current value for filters and TMS.

**[SWS\_Com\_00286]** [By a call of Com\_InvalidateShadowSignal, the AUTOSAR COM module shall replace the current value of the group signal with the given SignalId within the associated signal group by the group signal's ComSignalDataInvalidValue (ECUC\_Com\_00391).] (SRS\_Com\_02077)

The Com\_InvalidateShadowSignal is deprecated, a group signal should be invalidated either by Com\_InvalidateSignal or Com\_InvalidateSignalGroup.

The data invalid values are configured per group signal see ECUC\_Com\_00520.

The VFB defines only one attribute for a complex data type. Therefore, the best mapping of an invalidated complex data type to an invalidated signal group is to invalidate all group signals of a signal group. Therefore, the RTE can also request to invalidate the complete signal group by a call to `Com_InvalidateSignalGroup` (SWS\_Com\_00557).

### 7.3.4.2 Reception of an Invalidated Signal

**[SWS\_Com\_00680]** [If the configured `ComSignalDataInvalidValue` (ECUC\_Com\_00391) is received for a signal and the `ComDataInvalidAction` (ECUC\_Com\_00314) is configured to **NOTIFY** for this signal, the AUTOSAR COM module shall notify the RTE via the configured `ComInvalidNotification` function (ECUC\_Com\_00315). In this case, no other signal processing like filtering or the normal signal indication shall take place.] (SRS\_Com\_02079)

The reception deadline monitoring timer is also restarted in case of receiving an invalid signal or signal group, see SWS\_Com\_00738.

**[SWS\_Com\_00681]** [If the configured `ComSignalDataInvalidValue` (ECUC\_Com\_00391) is received for a signal and the `ComDataInvalidAction` (ECUC\_Com\_00314) is configured to **REPLACE** for this signal, the AUTOSAR COM module shall replace the signal's value by its configured `ComSignalInitValue` (ECUC\_Com\_00170). After the replacement, the normal signal processing like filtering and notification shall take place as if the `ComSignalInitValue` would have been received instead of the `ComSignalDataInvalidValue`.] (SRS\_Com\_02079, SRS\_Com\_02087)

**[SWS\_Com\_00682]** [If the configured `ComSignalDataInvalidValue` (ECUC\_Com\_00391) is received for at least one group signal of a signal group and the `ComDataInvalidAction` (ECUC\_Com\_00314) is configured to **NOTIFY** for this signal group, the AUTOSAR COM module shall notify the RTE via the configured `ComInvalidNotification` function (ECUC\_Com\_00315). In this case, no other signal group/ group signal processing like the normal indication shall take place.] (SRS\_Com\_02079)

**[SWS\_Com\_00683]** [If the configured `ComSignalDataInvalidValue` (ECUC\_Com\_00391) is received for at least one group signal of a signal group and the `ComDataInvalidAction` (ECUC\_Com\_00314), is configured to **REPLACE** for this signal group, the AUTOSAR COM module shall replace all group signals of this signal group by their configured `ComSignalInitValue` values. After the replacement, the normal signal group/ group signal processing like reception notification shall take place as if the `ComSignalInitValue` would have been received for all group signals.] (SRS\_Com\_02087)

**[SWS\_Com\_00717]** [If the configured `ComSignalDataInvalidValue` is received for a signal and its `ComDataInvalidAction` is configured to **NOTIFY**, the AUTOSAR COM module shall not store the received `ComSignalDataInvalidValue` into the signal object.] (SRS\_Com\_02079)



The next call to `Com_ReceiveSignal` will return the last valid received signal or the `ComSignalInitValue` in case no signal was received yet respectively.

**[SWS\_Com\_00718]** [If the configured `ComSignalDataInvalidValue` is received for at least one group signal of a signal group and its `ComDataInvalidAction` is configured to **NOTIFY**, the AUTOSAR COM module shall not store any of the received group signals into the signal objects.] (SRS\_Com\_02079)

The next call to `Com_ReceiveSignalGroup` will copy the last valid received group signals or the `ComSignalInitValues` in case the signal group was not received yet respectively into the shadow buffer.

## 7.3.5 Handling of I-PDUs

### 7.3.5.1 I-PDU Group Definitions

For an I-PDU group the following rules apply:

1. An I-PDU can belong to any I-PDU group.
2. **[SWS\_Com\_00771]** [An I-PDU is active (started) if and only if at least one I-PDU group is active (started) it belongs to.](SRS\_Com\_00218)
3. The maximum number of I-PDU groups is pre-compile configurable.

Rule 1 and 3 are supported by the COM configuration. The maximum number of supported I-PDU groups can be configured via `ComSupportedIpdudGroups` (ECUC\_Com\_00710).

Up to the definitions above, an I-PDU is named *activated/ started*, if any of the I-PDU groups containing this I-PDU is activated/ started. A not started I-PDU is called to be *stopped* or *deactivated*. An I-PDU must belong to at least one I-PDU group in order to be able to get started.

For the AUTOSAR COM module, the nesting of I-PDU groups is purely conceptual and must be resolved by an appropriate configuration and/ or usage of the AUTOSAR COM module.

For example, if an I-PDU “BUS1 RX Function1” belongs to I-PDU group “BUS1 RX” and I-PDU group “BUS1 RX” is included in I-PDU group “BUS1” then I-PDU “BUS1 RX Function1” must also be included in I-PDU group “BUS1”. Such dependencies need to be resolved at configuration time.

Once again, the COM module does not know or handle any grouping of I-PDUs. The I-PDU grouping must be handled either via configuration or by the calling module. The Figure 4 shows an exemplary usage of the AUTOSAR COM module, but does not define any functionalities of the AUTOSAR COM module.

Further, it is expected that the complete state handling of I-PDU groups is done outside of the AUTOSAR COM module, e.g. within the Basic Software Mode Manager. In case of a state change, the module managing the I-PDU group states passes the consistent state of all I-PDU groups via `Com_IpdudGroupControl` to the AUTOSAR COM module, which then starts and stops the single I-PDUs accordingly.

For I-PDUs that are not assigned to any I-PDU group and therefore which state can not be changed, the following exception to the above rules applies:

**[SWS\_Com\_00840]** [If an I-PDU is not assigned to any I-PDU group, the I-PDU shall be initially started (and never been stopped).](SRS\_Com\_00218)

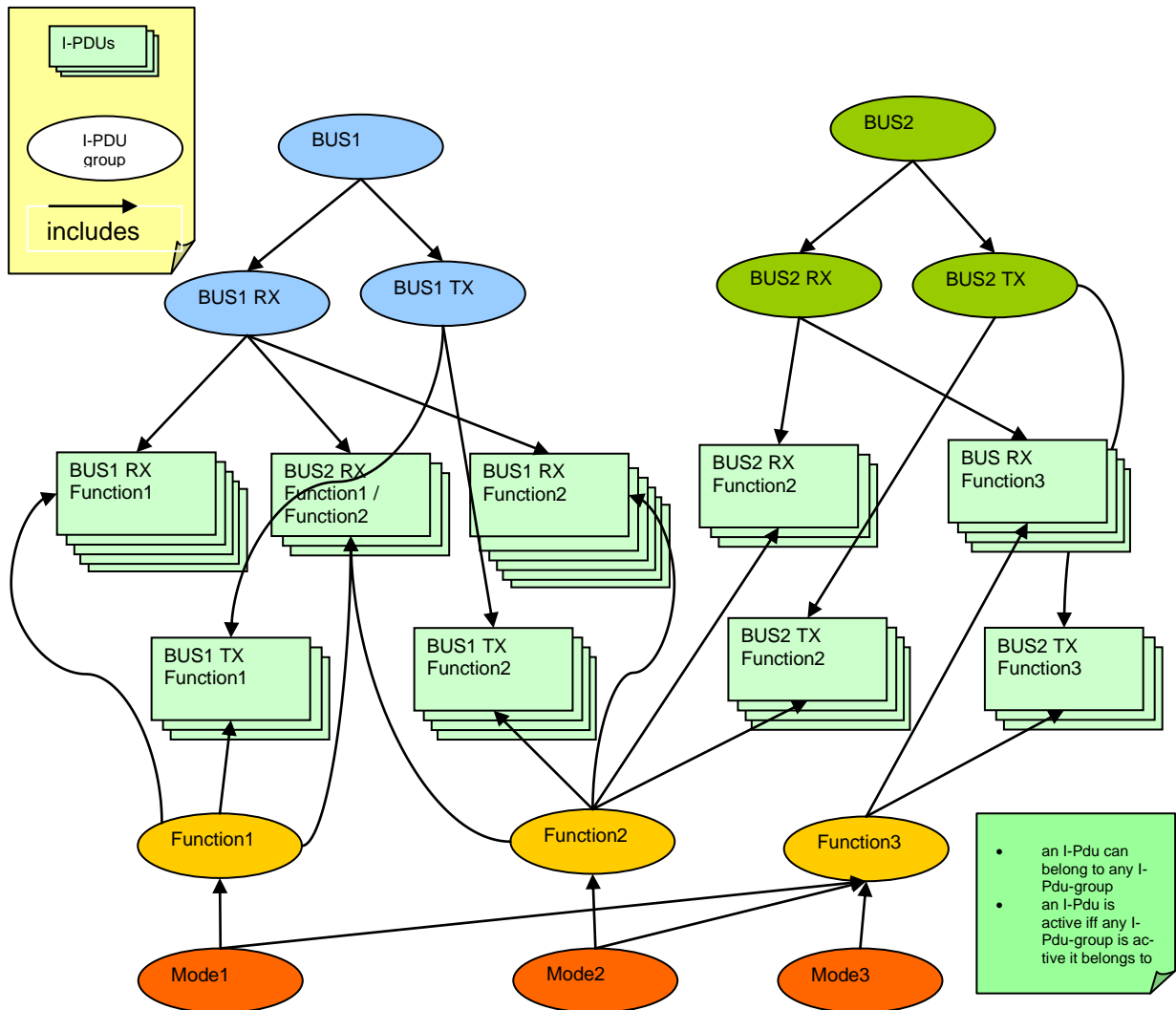


Figure 4: Example usage of I-PDU groups

### 7.3.5.2 Starting of I-PDU Groups

By default all I-PDU groups are stopped, see SWS\_Com\_00444. A call to Com\_IpduGroupControl starts an I-PDU group if its requested activation state is started and the I-PDU group was previously stopped.

**[SWS\_Com\_00114]** [If an I-PDU is started as result of a call to Com\_IpduGroupControl, the AUTOSAR COM module shall permit to transmit/ receive its signals and signal groups, see also Table 4.] (SRS\_Com\_00218)

**[SWS\_Com\_00787]** [If an I-PDU is started as result of a call Com\_IpduGroupControl, the AUTOSAR COM module shall always initialize the following attributes of this I-PDU:

- 1) ComMinimumDelayTime of I-PDUs in transmission mode *DIRECT* or *MIXED*
- 2) timeout attributes of I-PDUs for deadline monitoring aspect: all timeout timers (ComFirstTimeout, ComTimeout) shall restart

- 3) all included update-bits shall be cleared
- 4) reset *OCCURRENCE* of filter with *ComFilterAlgorithm ONE EVERY N*
- 5) set the I-PDU counter to 0 for I-PDUs with *ComIPduDirection* configured to *SEND*
- 6) accept for I-PDUs with *ComIPduDirection* configured to *RECEIVED* any next incoming I-PDU counter] (SRS\_Com\_00218)

**[SWS\_Com\_00222]** [If an I-PDU is started as result of a call *Com\_IpduGroupControl* with parameter *Initialize* set to true, the AUTOSAR COM module shall additionally to *SWS\_Com\_00787* initialize the following attributes of this I-PDU:

- 1) the data of the I-PDU as defined in *SWS\_Com\_00217*
- 2) the shadow buffers of included signal groups
- 3) *old\_value* of the filtering mechanisms for each signal to the *ComSignalInitValue*
- 4) *ComTxModeTimePeriod* and *ComTxModeTimeOffset* of I-PDUs in *Periodic* or *MIXED* transmission mode] (SRS\_Com\_00218)

**[SWS\_Com\_00223]** [If an I-PDU is started as result of a call to *Com\_IpduGroupControl*, the AUTOSAR COM module shall determine its transmission mode according to its current data content.] (SRS\_Com\_00218)

**[SWS\_Com\_00228]** [In some cases, an I-PDU is started as result of a call to *Com\_IpduGroupControl* before all its contained signals have been written. In this case, the AUTOSAR COM module shall use the *ComSignalInitValue* for the missing signal data.] (SRS\_Com\_00218)

**[SWS\_Com\_00229]** [When an I-PDU is started and one or more signals in that I-PDU have already been written via one of the send APIs by the upper layer, the AUTOSAR COM module shall use the most recently obtained values to determine the TMS of the I-PDU.] (SRS\_Com\_00218)

**[SWS\_Com\_00733]** [If an I-PDU is started as result of a call to *Com\_IpduGroupControl* and the I-PDU contains signals that have deadline monitoring configured (*ECUC\_Com\_00183*, *ECUC\_Com\_00263*), the AUTOSAR COM module shall start the deadline monitoring for these signals independently of the value of the *initialize* parameter.] (SRS\_Com\_00218)

Note, in case an I-PDU is started by a call to *Com\_IpduGroupControl* and the parameter *initialize* is set to false, the internal buffers are not changed and the I-PDU contains the last the last set or init values. Particular, if the I-PDU is started for the first time and *initilaize* is set to false, the COM operates on the *init-values* set by *Com\_Init*.

### 7.3.5.3 Stopping of I-PDU Groups

A call to *Com\_IpduGroupControl* stops an I-PDU group if its requested activation state is stopped and the I-PDU group was previously started.

In order to disable the transmission of an I-PDU, all I-PDU groups containing this I-PDU needs to be stopped. Hence, to implement ***listen-only-mode***, all I-PDU groups containing transmission I-PDUs needs to be stopped. Receiving of I-PDUs may also be stopped.

**[SWS\_Com\_00334]** [By a call to the functions: Com\_SendSignal, Com\_SendSignalGroup, or Com\_InvalidateSignal, the AUTOSAR COM module shall update the values of its internal buffers even for stopped I-PDUs. See also Table 3.] (SRS\_Com\_00218)

If a signal written to a stopped I-PDU would trigger the transmission of this I-PDU if it were not stopped, then this trigger is not stored. After re-starting the corresponding I-PDU group, such an old trigger does not lead to an immediate transmission of the I-PDU.

**[SWS\_Com\_00777]** [If an I-PDU is stopped as result of a call to Com\_IpduGroupControl, the AUTOSAR COM module shall cancel any outstanding transmission requests for this I-PDU. This includes cancelling any potential retries with respect to ComRetryFailedTransmitRequests.] (SRS\_Com\_00218)

**[SWS\_Com\_00115]** [If an I-PDU is stopped as result of a call to Com\_IpduGroupControl, the AUTOSAR COM module shall cancel the deadline monitoring for all pending confirmations.] (SRS\_Com\_00218)

**[SWS\_Com\_00800]** [The AUTOSAR COM module shall ignore any transmit confirmations for a stopped I-PDU.] (SRS\_Com\_00218)

The AUTOSAR COM module cannot prohibit the invocation of the Com\_TriggerTransmit function. However, in case of a stopped I-PDU, the function Com\_TriggerTransmit returns E\_NOT\_OK. See SWS\_Com\_00001.

**[SWS\_Com\_00684]** [If an I-PDU is stopped as result of a call to Com\_IpduGroupControl, the AUTOSAR COM module shall disable its reception processing.] (SRS\_Com\_00218)

**[SWS\_Com\_00713]** [If a large I-PDU is stopped as result of a call to Com\_IpduGroupControl, the AUTOSAR COM module shall stop the reception process and ignore the partly received I-PDU. ] (SRS\_Com\_00218)

**[SWS\_Com\_00685]** [If an I-PDU is stopped as result of a call to Com\_IpduGroupControl, the AUTOSAR COM module shall cancel its deadline monitoring.] (SRS\_Com\_00218)

**[SWS\_Com\_00479]** [If an I-PDU is stopped as result of a call to Com\_IpduGroupControl, the AUTOSAR COM module shall immediately invoke the configured ComErrorNotification (ECUC\_Com\_00499), for outstanding not confirmed transmitted signals/ signal groups of the stopped I-PDU.] (SRS\_Com\_00218)

[SWS\_Com\_00714] [If a large I-PDU is stopped while its transmission is already in progress, the AUTOSAR COM module shall stop the transmission process immediately.] (SRS\_Com\_00218)

Table 3 gives an overview of the behavior of stopped I-PDUs:

<b>Behavior of stopped I-PDUs</b>	
<b>Receiver side (RX)</b>	<b>Transmitter side (TX)</b>
<ul style="list-style-type: none"> <li>Disable RX deadline monitoring</li> <li>No action on a Com_RxIndication to RTE, no storing of the I-PDU</li> <li>Return code COM_SERVICE_NOT_AVAILABLE on Com_ReceiveSignal and Com_ReceiveSignalGroup and the last known value (or init value) is given back as data</li> </ul>	<ul style="list-style-type: none"> <li>Disable sending</li> <li>Disable TX deadline monitoring</li> <li>Ignore Com_TxConfirmation:</li> <li>On a call of Com_SendSignal, Com_SendSignalGroup, Com_InvalidateSignal and the values in the AUTOSAR COM module's internal buffers are still up-dated but the return code COM_SERVICE_NOT_AVAILABLE is returned</li> <li>Outstanding transmission request (e.g. N-Times) will be cancelled</li> <li>Return code E_NOT_OK on Com_TriggerTransmit</li> </ul>
	<b>For periodic (TX)</b>
	Do not send any more

**Table 3: Behavior of stopped I-PDUs**

Table 4 gives an overview of the behavior of started I-PDUs:

<b>Behavior of started I-PDUs</b>	
<b>Receiver side (RX)</b>	<b>Transmitter side (TX)</b>
<ul style="list-style-type: none"> <li>Reinitialize timeouts if Initialize==true (ComSignalFirstTimeout, ComSignalTimeout)</li> <li>Normal reaction on Com_RxIndication/Com_TpRxIndication</li> <li>Normal reaction on Com_ReceiveSignal, and Com_ReceiveSignalGroup</li> </ul>	<ul style="list-style-type: none"> <li>Normal reaction on Com_InvalidateSignal, Com_SendSignal, and Com_SendSignalGroup</li> <li>No transmission timeout notification until next send</li> <li>Normal reaction on Com_TxConfirmation/Com_TpTxConfirmation</li> <li>Normal reaction on Com_TriggerTransmit</li> </ul>
	<b>For periodic (TX)</b>
	Start at 0

**Table 4 Behavior of started I-PDUs**

### 7.3.5.4 Signal Indication (Unpacking of I-PDUs)

In order to support both interrupt-driven and polled systems, it can be configured when the signal indication takes place. There are two configurable signal indication

modes *IMMEDIATE* and *DEFERRED* configurable via `ComIPduSignalProcessing`. See `ECUC_Com_00119`.

**[SWS\_Com\_00300]** [If `ComIPduSignalProcessing` for an I-PDU is configured to *IMMEDIATE*, the AUTOSAR COM module shall invoke the configured `ComNotifications` for the included signals and signal groups within the `Com_RxIndication`, or `Com_TpRxIndication` function respectively.] (SRS\_Com\_02046)

**[SWS\_Com\_00301]** [If `ComIPduSignalProcessing` for an I-PDU is configured to *DEFERRED*, the AUTOSAR COM module shall first copy the I-PDU's data within the `Com_RxIndication` function or the related TP reception functions respectively from the PduR into COM. Then the AUTOSAR COM module shall invoke the configured `ComNotifications` for the included signals and signal groups asynchronously during the next call to `Com_MainFunctionRx`.] (SRS\_Com\_02046)

If in *DEFERRED* mode a call to `Com_ReceiveSignal` is made before the deferred unpacking takes place, the previous not updated values are returned.

A sequence chart with both indication options can be found in Chapter 9.3. The configuration of these modes is defined in `ECUC_Com_00119`.

**[SWS\_Com\_00574]** [When unpacking an I-PDU, the AUTOSAR COM module shall check the received data length (`PduInfoPtr->SduLength`) and unpack and notify only completely received signals via `ComNotification`.] (SRS\_Com\_02046)

**[SWS\_Com\_00794]** [In case of receiving of a smaller I-PDU than expected results into receiving a signal without its configured update-bit, the AUTOSAR COM module shall treat this signal as if its update bit was set and interpret such a signal as updated.] (SRS\_Com\_02046)

If the received I-PDU length is smaller than the configured/ expected I-PDU length, it needs to be prevented that signals are updated partially. On the other hand all completely received signals should be received and notified to the upper layer.

**[SWS\_Com\_00575]** [When unpacking an I-PDU, the AUTOSAR COM module shall check the received data length (`PduInfoPtr->SduLength`) and in case a signal group is received only partially, such a signal group and all included group signals shall not be unpacked or notified via `ComNotification`.] (SRS\_Com\_02046)

The above requirement prevents inconsistently received signal groups and therefore inconsistently received complex data types.

The AUTOSAR COM module does not copy or handle additional received data for not configured signals in case the received data length is greater than expected.

### 7.3.5.5 Minimum Delay Timer (MDT)

The AUTOSAR COM module inherits the minimum delay timer mechanism as defined [17]. The OSEK COM mechanism is clarified and detailed in this chapter.

When an I-PDU is started, the MDT is re-initialized. Therefore, the MDT can be violated by stopping and starting I-PDUs rapidly.

The behavior of the transmission deadline monitoring timer is not affected by any transmission delay caused by the minimum delay time supervision.

**[SWS\_Com\_00471]** [No minimum delay time monitoring shall take place, if ComMinimumDelayTime is omitted or configured to 0.] (SRS\_Com\_02037)

**[SWS\_Com\_00789]** [If ComEnableMDTForCyclicTransmission (ECUC\_Com\_00788) is configured to false, the minimum delay timer shall not be (re)started in the following cases:

- I-PDUs with ComTxModeMode PERIODIC
- the cyclic part of I-PDUs with ComTxModeMode MIXED
- repeated transmissions for I-PDUs in case ComTxModeNumberOfRepetitions > 0

] (SRS\_Com\_02037)

In case the minimum delay timer is currently not started, there is no way for the minimum delay timer to expire. Thus, optimized implementations might for example omit monitoring the minimum delay time for periodic I-PDUs completely in case ComEnableMDTForCyclicTransmission is configured to false, since the minimum delay timer never gets started.

**[SWS\_Com\_00698]** [If ComMinimumDelayTime of an I-PDU is configured greater than 0 and ComRetryFailedTransmitRequests is set to *FALSE*, the AUTOSAR COM module shall load and start the minimum delay time counter upon transmission of that I-PDU to the PDU Router via PduR\_ComTransmit.] (SRS\_Com\_02037)

**[SWS\_Com\_00828]** [If ComMinimumDelayTime of an I-PDU is configured greater than 0 and ComRetryFailedTransmitRequests is set to *TRUE*, the AUTOSAR COM module shall load and start the minimum delay time counter upon transmission of that I-PDU to the PDU Router via PduR\_ComTransmit only in case PduR\_ComTransmit returns E\_OK.] (SRS\_Com\_02037)

**[SWS\_Com\_00469]** [If ComMinimumDelayTime of an I-PDU is configured greater than 0, the AUTOSAR COM module shall (re-)load the already running minimum delay time counter with ComMinimumDelayTime for that I-PDU when Com\_TxConfirmation is invoked and the minimum delay time counter started at PduR\_ComTransmit (see SWS\_Com\_00698) of that I-PDU is not already elapsed.] (SRS\_Com\_02037)

The running minimum delay timer is reloaded upon the reception of the TX-confirmation of that I-PDU, unless the transmission was already delayed longer than ComMinimumDelayTime at the reception of the confirmation. In normal case, there will be no further transmission of that I-PDU by the AUTOSAR COM module unless the loaded and started minimum delay has expired. See also Figures 2-4, 2-5 and 2-7 in [17]. However, some exception exists: According to SWS\_Com\_00475 Com\_TriggerTransmit does not interfere with the minimum delay timer. Further, the mini-



imum delay timer is reset if the transmission deadline monitoring timer expires; see Chapter 2.3.4 in [17]. In addition, starting an I-PDU group resets the minimum delay time timer of the included I-PDUs.

**[SWS\_Com\_00812]** [If the minimum delay timer of an I-PDU is reset because the transmission deadline monitoring timer expires and a postponed request for this I-PDU exists, the AUTOSAR COM module shall immediately transmit this I-PDU.] (SRS\_Com\_02037)

### 7.3.6 Deadline Monitoring

Deadline monitoring for signals is defined in [17].

In the context of deadline monitoring for a signal group, it is handled like a signal. The deadline monitoring parameters ComFirstTimeout and ComTimeout can be defined in the configuration container ComSignal or ComSignalGroup.

The corresponding timeout notification callback functions can be defined in parameter ComTimeoutNotification in the configuration container ComSignal or ComSignalGroup.

**[SWS\_Com\_00333]** [If the configuration parameter ComTimeout for a signal or signal group is omitted or configured to 0, the AUTOSAR COM module shall not monitor this signal or signal group. It also shall ignore the ComFirstTimeout.] (SRS\_Com\_02089)

#### 7.3.6.1 Reception Deadline Monitoring

The reception deadline monitoring can be enabled and disabled by the control of I-PDU groups analogous to Chapter 7.3.5.1.

**[SWS\_Com\_00772]** [The reception deadline monitoring of an I-PDU is enabled if and only if it is contained in an I-PDU group that has reception deadline monitoring enabled. Otherwise, the reception deadline monitoring of the I-PDU is *disabled*.] (SRS\_Com\_00192)

**[SWS\_Com\_00292]** [In the case where reception deadline monitoring is configured for signals with update-bits, the AUTOSAR COM module shall perform a separate reception deadline monitoring for each signal/ signal group with an update-bit. For configuration of ComTimeout, see ECUC\_Com\_00263.] (SRS\_Com\_02089, SRS\_Com\_02058)

**[SWS\_Com\_00290]** [The AUTOSAR COM module shall perform an I-PDU based reception deadline monitoring for signals without an update-bit.] (SRS\_Com\_02089, SRS\_Com\_02058)

**[SWS\_Com\_00291]** [For all signals and signal groups without update-bits within the same I-PDU, the AUTOSAR COM module shall perform the reception deadline monitoring of the I-PDU using the smallest configured non-zero timeout parameter (Com-

FirstTimeout, ComTimeout) of the associated signals and signal groups.] (SRS\_Com\_02089, SRS\_Com\_02058)

If all signals within an I-PDU with configured reception deadline monitoring have also an update-bit configured, no reception deadline monitoring on I-PDU base needs to be performed.

In case of an Rx-timeout, the ComRxDataTimeoutAction parameter determines whether the AUTOSAR COM module replaces the signal/ signal group value with the initial value or maintains the last received value. See ECUC\_Com\_00314.

**[SWS\_Com\_00470]** [If ComRxDataTimeoutAction is set to *REPLACE* (see ECUC\_Com\_00412), the AUTOSAR COM module shall replace the signal's value by its ComSignalInitValue (see ECUC\_Com\_00170) when the reception deadline monitoring timer of a signal expires.] (SRS\_Com\_02087, SRS\_Com\_02088)

In case the ComSignalInitValue replaces the last received value, the last received value is overwritten and gets lost. Therefore, the AUTOSAR COM module returns the ComSignalInitValue for this signal until a new value is received.

**[SWS\_Com\_00500]** [If the reception deadline monitoring timer of a signal or signal group expires and the configuration parameter ComRxDataTimeoutAction for this signal or signal group is either omitted or configured to *NONE*, the AUTOSAR COM module shall not replace the signal or the signals of the signal groups respectively.] (SRS\_Com\_02087)

**[SWS\_Com\_00513]** [If ComRxDataTimeoutAction is set to *REPLACE* (see ECUC\_Com\_00412), the AUTOSAR COM module shall replace the values of all included group signals by their ComSignalInitValues (see ECUC\_Com\_00170), when the reception deadline monitoring timer of a signal group expires.] (SRS\_Com\_02041)

The Rx-timeout-indication can be combined and configured separately from ECUC\_Com\_00412.

**[SWS\_Com\_00715]** [When a (large) I-PDU with a configured reception deadline monitoring is received successfully, the AUTOSAR COM module shall reset the reception deadline monitoring timer for this (large) I-PDU at invocation of the function Com\_RxIndication or Com\_TpRxIndication respectively.] (SRS\_Com\_02058)

**[SWS\_Com\_00716]** [If the configuration parameter ComFirstTimeout for a signal or signal group is omitted or configured to 0, the AUTOSAR COM module shall not monitor the reception of this signal or signal group respectively from the start of the corresponding I-PDU until the first reception.] (SRS\_Com\_02058)

**[SWS\_Com\_00738]** [The reception deadline monitoring timer mechanism shall not take the values of the signals into account. Hence, the AUTOSAR COM module shall restart the reception deadline monitoring timer also in case of receiving an invalid value.] (SRS\_Com\_02089)

### 7.3.6.1.1 En-/Disable Reception Deadline Monitoring

When reception deadline monitoring of an I-PDU is disabled and the timer expires, no error indication will be given to the RTE.

Disabling reception deadline monitoring does not stop the reception of an I-PDU.

**[SWS\_Com\_00224]** [If the reception deadline monitoring state of an I-PDU is changed by a call to `Com_ReceptionDMControl` from disabled to enabled, the AUTOSAR COM module shall set the reception deadline monitoring timer for the included signals and signal groups to the configured `ComFirstTimeout` value.] (SRS\_Com\_00192)

**[SWS\_Com\_00486]** [The AUTOSAR COM module shall silently ignore setting the reception deadline monitoring of an I-PDU to enabled by a call to `Com_ReceptionDMControl`, in case the reception deadline monitoring is already enabled for this I-PDU.] (SRS\_Com\_00192)

Enabling reception deadline monitoring implies that timeout notifications of deadline monitoring expiry are notified to the RTE for signals or signal groups with a configured `ComTimeoutNotification` (ECUC\_Com\_00552).

**[SWS\_Com\_00534]** [If `Com_ReceptionDMControl` is invoked on an I-PDU group containing only/ also Tx-I-PDUs, then the AUTOSAR COM module shall silently ignore the Tx-I-PDUs.] (SRS\_Com\_00192)

**[SWS\_Com\_00225]** [The AUTOSAR COM module shall silently ignore setting the reception deadline monitoring of an I-PDU to disabled by a call to `Com_ReceptionDMControl`, in case the reception deadline monitoring is already disabled for this I-PDU.] (SRS\_Com\_00192)

### 7.3.6.2 Transmission Deadline Monitoring

For transmission deadline monitoring, there is no difference between signals with update-bits and signals without update-bits. Therefore, transmission deadline monitoring can be performed on I-PDU base. Nevertheless, notification about detected transmission deadline violations on sender side is done per signal. See [17] for further details.

**[SWS\_Com\_00481]** [The AUTOSAR COM module shall perform the transmission deadline monitoring (if configured) for all signals and signal groups independently from the transfer property and independently from the transmission modes of the I-PDU the signal / signal group belongs to.] (SRS\_Com\_02037)

The AUTOSAR COM module makes no distinction between signals with `ComTransferProperty` *PENDING* or *TRIGGERED* with respect to transmission deadline monitoring.

**[SWS\_Com\_00445]** [If different `ComTimeout` parameters of the associated signals/ signal groups of an I-PDU are configured, the AUTOSAR COM module shall use the

smallest value as timeout parameter for the transmission deadline monitoring of the I-PDU.] (SRS\_Com\_02037)

Transmission deadline monitoring should only be configured in the AUTOSAR COM module for busses that support the generation of transmit confirmations. Otherwise, the transmission deadline monitoring would always notify a transmission error.

**[SWS\_Com\_00696]** [In case transmission deadline monitoring is configured for an I-PDU with transmission mode *NONE* only, the AUTOSAR COM module shall start the transmission deadline monitoring timer for this I-PDU upon the start of the I-PDU group to which the I-PDU belongs to.] (SRS\_Com\_02037)

**[SWS\_Com\_00835]** [In case transmission deadline monitoring is configured for an I-PDU with transmission mode *NONE* and another transmission mode, the transmission deadline monitoring shall be disabled whenever the transmission mode *NONE* is active.] (SRS\_Com\_02037)

**[SWS\_Com\_00697]** [In case transmission deadline monitoring is configured for an I-PDU with transmission mode *NONE* only, the AUTOSAR COM module shall reset the transmission deadline monitoring timer for this I-PDU upon each transmission confirmation via Com\_TxConfirmation for this I-PDU.] (SRS\_Com\_02037)

In case of a signal group, it is only possible to configure transmission deadline monitoring for the whole signal group and not for group signals, see ECUC\_Com\_00345 and ECUC\_Com\_00520.

**[SWS\_Com\_00708]** [In case the transmission deadline monitoring timer expires for an I-PDU with ComIPduCancellationSupport configured to *TRUE*, the AUTOSAR COM module shall invoke PduR\_ComCancelTransmit for that I-PDU.] (SRS\_Com\_02107)

The invocation of PduR\_ComCancelTransmit is done in addition to the normal timeout handling of Com, which notifies the SW-C, if configured.

The AUTOSAR COM ignores the return code of PduR\_ComCancelTransmit.

#### 7.3.6.2.1 Clarification of the OSEK COM specification

The following requirement SWS\_Com\_00304 states the behavior of the transmission deadline monitoring in the *MIXED* transmission mode defined in [17] more precisely.

**[SWS\_Com\_00304]** [If the transmission does not occur, i.e. if there is no confirmation of the I-PDU's transmission by PduR, then the time-out occurs and the AUTOSAR COM module shall notify the RTE by invoking the configured ComTimeoutNotification, see ECUC\_Com\_00552.] (SRS\_Com\_02037)

If the transmission deadline monitoring timer runs out there will be a timeout notification regardless of the reason. The notification will take place even if the transmission was postponed because of the MDT or if the I-PDU was filtered out by an I-PDU callout.

In the case that there are any contradictions between text and diagrams in [17] the text is the normative part.

In [17] is defined that in *DIRECT* transmission mode (here *DIRECT/N-times* with  $n == 0$ ): “The monitoring timer is started upon completion of the call to the `SendMessage`, `SendDynamicMessage` or `SendZeroMessage` API service.”

**Clarification:** The transmission deadline monitoring timer should only be reset if the corresponding signal has transmission deadline monitoring timeouts configured. Signals that have not configured transmission deadline monitoring should not interfere in the I-PDU based monitoring process.

#### 7.3.6.2.2 Transmission Deadline Monitoring with N-Times Transmission Mode

As defined in [17] the monitoring timer has to be started upon completion of a call to `Com_SendSignal` or `Com_SendSignalGroup` respectively if transmission deadline monitoring is configured for the corresponding signal or signal group respectively.

For the transmission mode *DIRECT* and *MIXED*, it should be ensured that all `ComTxModeNumberOfRepetitions` requests could be made within the configured period (see Chapter 7.3.3.4).

As defined in [17], if the monitoring timer expires the RTE is notified with the configured notification mechanism about that failure.

**[SWS\_Com\_00308]** [For an I-PDU with `ComTxModeMode` *DIRECT* and `ComTxModeNumberOfRepetitions`  $> 0$ , the AUTOSAR COM module shall cancel the transmission deadline monitoring timer after `ComTxModeNumberOfRepetitions` plus one received confirmations.] (SRS\_Com\_02083)

If the timer is cancelled after the `ComTxModeNumberOfRepetitions` plus one confirmations, the transmission was successful and then the transmission confirmation is send to the RTE. See also SWS\_Com\_00305.

**[SWS\_Com\_00739]** [For an I-PDU with `ComTxModeMode` *DIRECT* and `ComTxModeNumberOfRepetitions`  $> 0$ , the AUTOSAR COM module shall reset an already running transmission deadline monitoring timer in case another send request for this I-PDU is initiated.] (SRS\_Com\_02083)

## 7.4 Signal Groups - Complex Data Types

To support the AUTOSAR concept of complex data types the AUTOSAR COM module provides signal groups. The AUTOSAR COM module transmits and receives signal groups consistently to provide the necessary consistency for complex data types.

Signal groups can be configured statically. For each signal group a symbolic name can be configured. See ECUC\_Com\_00345 for the configuration details.

The AUTOSAR COM module achieves the consistency of a signal group by means of a shadow buffer mechanism, i.e. the RTE accesses the group signals in the shadow buffer. If the shadow buffer needs to be synchronized with the I-PDU, the RTE can trigger this explicitly with `Com_SendSignalGroup` or `Com_ReceiveSignalGroup`. The synchronization is performed atomically.

An alternative approach to signal group consistency is to enable the signal group array APIs `Com_SendSignalGroupArray` (SWS\_Com\_00851) and `Com_ReceiveSignalGroupArray` (SWS\_Com\_00854). The deviations of this approach are described in Chapter 7.4.6.

### 7.4.1 Initialization

**[SWS\_Com\_00484]** [By a call to `Com_Init`, the AUTOSAR COM module shall initialize the shadow buffer of a signal group on sender-side.] (SRS\_BSW\_00101)

Since it is not suspected that a well-formed SWC tries to read a group signal before a call to `Com_ReceiveSignalGroup`, SWS\_Com\_00484 applies to the sender side only.

### 7.4.2 Transmission

If `Com_SendSignal` or `Com_InvalidateSignal` is called for a signal that belongs to a signal group, then the AUTOSAR COM will only update the shadow buffer of this signal group. There is no need for any further I-PDU processing like TMS evaluation, unless the I-PDU contents changed.

**[SWS\_Com\_00050]** [If `Com_SendSignalGroup` is called for the signal group, the AUTOSAR COM module shall copy the shadow buffer atomically to the I-PDU buffer.] (SRS\_Com\_02041)

Example with two group signals `signal_a` and `signal_b`, which belong to `group_x`:

```
/* copy a to shadow buffer */
Com_SendSignal (signal_a, &a);

/* copy b to shadow buffer */
Com_SendSignal (signal_b, &b);

/* copy shadow buffer to I-PDU */
Com_SendSignalGroup (group_x);
```

### 7.4.3 Reception

A group signal can be received from the shadow buffer by calling the function `Com_ReceiveSignal` after the signal group data has been copied to the shadow buffer by `Com_ReceiveSignalGroup`.

**[SWS\_Com\_00051]** [If Com\_ReceiveSignalGroup is called for a signal group, the AUTOSAR COM module shall copy the data atomically from the I-PDU buffer to the shadow buffer.] (SRS\_Com\_02041)

Example with two group signals signal\_a and signal\_b, which belong to group\_x:

```
/* copy I-PDU to shadow buffer */  
Com_ReceiveSignalGroup (group_x);  
  
/* copy a from shadow buffer */  
Com_ReceiveSignal (signal_a, &a);  
  
/* copy b from shadow buffer */  
Com_ReceiveSignal (signal_b, &b);
```

#### 7.4.4 Notifications

Table 5 shows that all notification functions are only configurable for signal groups but not for individual group signals. Chapter 7.3 defines the functional logic of the notification methods for signals and signal groups. The prototypes for the configurable notification functions are defined in Chapter 8.6.3.1.

### 7.4.5 Attributes of a Signal Group

Table 5 gives an overview of the attributes of a signal group:

<b>Attribute</b>	<b>Per group signal</b>	<b>Per signal group</b>
Update-bit	No	Yes, associated on the whole group (see Chapter 7.8)
Signal Notification (sender side)	No	Yes
Signal Notification (receiver side)	No	Yes
Error Notification (sender side)	No	Yes
Timeout Notification (sender side)	No	Yes
Timeout Notification (receiver side)	No	Yes
Invalid Notification (receiver side)	No	Yes
Data access (receiver side)	Yes, see SWS_Com_00202	Yes, see SWS_Com_00201
Data access (sender side)	Yes, see SWS_Com_00199, SWS_Com_00288	Yes, see SWS_Com_00200, SWS_Com_00557
Data Filtering (receiver side)	Yes, see Chapter 7.2.4	No
Data Filtering (sender side)	No	No
TMS on sender side	Each signal, according to TMS selection definition. (see Chapter 7.3.3.2)	No

**Table 5: Attributes of signal groups**

### 7.4.6 UINT8-array based access to signal groups

An alternative approach to signal group consistency is to enable the signal group array APIs `Com_SendSignalGroupArray` (SWS\_Com\_00851) and `Com_ReceiveSignalGroupArray` (SWS\_Com\_00854) via the COM configuration parameter `ComSignalGroupArrayAccess` (ECUC\_Com\_10003). The serialization of the composite data is then done outside of the COM module but resulting in the same layout as COM would use to place the group signals in the I-PDU.

In this approach, no shadow buffer is used. The serialized data is directly provided to the COM module as a uint8-array representation of the signal group.

**[SWS\_Com\_00841]** [The UINT8-array based access to signal groups shall only be used if the following preconditions apply:

- Uses only fix sized data types for the composite data.
- Signal groups, which are mapped byte aligned to the I-PDU.
- Signal groups where all group signals are mapped consecutively to the I-PDU on transmission side.] (SRS\_Com\_02112)

**[SWS\_Com\_00842]** [In case a signal group has `ComSignalGroupArrayAccess` (ECUC\_Com\_10003) defined, no shadow buffer shall be allocated for that signal group.] (SRS\_Com\_02112)



**[SWS\_Com\_00843]** [In case a signal group has ComSignalGroupArrayAccess (ECUC\_Com\_10003) defined, the call to the following APIs is not supported for that signal group:

- Com\_SendSignal and Com\_ReceiveSignal for any of the group signals of that signal group
- Com\_UpdateShadowSignal [deprecated] and Com\_ReceiveShadowSignal [deprecated] for any of the group signals of that signal group
- Com\_SendSignalGroup and Com\_ReceiveSignalGroup for that signal group

] (SRS\_Com\_02112)

The above requirement defines that certain APIs are not supported for signal groups with array access. That is, these APIs must not be called for signal groups with array access. If they are called illegally, the behavior is undefined.

#### 7.4.6.1 Basic functionality

In case the AUTOSAR Transformer approach is used the serialization and further data processing is already done in the transformer chain. Since the transformer chain already deals with a uint8-array representation of the communication data, the uint8-array based access to signal groups allows to use the already serialized data and to place it into the I-PDU without the need for COM to serialize the data again.

The AUTOSAR System Template [12] defines which signal group shall be handled in the uint8-array based approach. If the System Template enables the uint8-array based approach the Ecu configuration parameter ComSignalGroupArrayAccess (ECUC\_Com\_10003) is set to true.

**[SWS\_Com\_00844]** [The start position of the uint8-array representation of the communication data shall be the byte offset of the group signal in this signal group with the smallest ComBitPosition.] (SRS\_Com\_02112)

**[SWS\_Com\_00845]** [The length of the uint8-array representation of the communication data shall be the number of bytes which all the ComGroupSignals in this ComSignalGroup occupy in the ComIPdu.] (SRS\_Com\_02112)

Please note that for reception, the signal group may not be consecutive and other signals may be interleaved in the uint8-array representation of the received signal group.

#### 7.4.6.2 Initialization

**[SWS\_Com\_00850]** [By a call to Com\_Init, the AUTOSAR COM module shall initialize the respective I-PDU buffer section of a signal group, that has ComSignalGroupArrayAccess (ECUC\_Com\_10003) configured *true*, based on the ComSignalInitValue of all included ComGroupSignals.] (SRS\_Com\_02112, SRS\_BSW\_00101)

### 7.4.6.3 Transmission

**[SWS\_Com\_00846]** [When `Com_SendSignalGroupArray` is called a data pointer to the uint8-array representation of the signal group is provided. The AUTOSAR COM module shall copy the data into the I-PDU based on the start position defined in `SWS_Com_00844` and the size defined in `SWS_Com_00845`.] (SRS\_Com\_02112)

**[SWS\_Com\_00847]** [The AUTOSAR COM module shall handle a call of `Com_SendSignalGroupArray` equally to `Com_SendSignalGroup` with respect to

- Chapter 7.3.3.1.2 Signal Groups (Transfer Properties and I-PDU Transmission Mode)
- Chapter 7.3.3.2 Selection of the Transmission Mode for one specific I-PDU
- Chapter 7.3.3.3 Signal Flow and Transmission Mode Selection

] (SRS\_Com\_02112)

Since the signal group is provided in the uint8-array representation the transmission mode selection needs to cope with this data representation when analyzing the group signals.

In order to ease the Transmission Mode Selection only a subset of the filters from `SWS_Com_00602` are supported.

Two filters are excluded from the support of TMS for `Com_SendSignalGroupArray`:

- `NEW_IS_WITHIN`
- `NEW_IS_OUTSIDE`

**[SWS\_Com\_00848]** [The call of `Com_SendSignalGroupArray` shall support the following filters for Transmission Mode Selection (TMS)

- `ALWAYS`
- `NEVER`
- `MASKED_NEW_EQUALS_X`
- `MASKED_NEW_DIFFERS_X`
- `MASKED_NEW_DIFFERS_MASKED_OLD`
- `ONE EVERY_N`

] (SRS\_Com\_02112)

### 7.4.6.4 Reception

**[SWS\_Com\_00849]** [When `Com_ReceiveSignalGroupArray` is called a data pointer to the uint8-array representation of the signal group is provided. The AUTOSAR COM module shall copy the data to the data pointer position of the I-PDU based on the start position defined in `SWS_Com_00844` and the size defined in `SWS_Com_00845`.] (SRS\_Com\_02112)

## 7.5 Large Data Types

A **large signal** is a signal that is too large to fit into a single L-PDU of the underlying communication protocol. For example, a large signal on CAN would exceed 8 bytes or 64 bytes for CAN FD. For FlexRay the situation is more complex, since one frame

can contain several I-PDUs. Hence, a signal may be a large signal for FlexRay even if it does not exceed the size of a FlexRay frame.

Large signals in AUTOSAR require configuring a **large I-PDU** that will be transmitted via the transport protocol of the underlying bus. It is not imperative that a large I-PDU contains one or more large signals.

Using this concept the I-PDU length and hence the signal size will be limited according to the bus-specific transport protocol, see ECUC\_Com\_00437.

UINT8\_N is the only signal type that supports signals of a static size larger than 8 bytes. For dynamic length signals see Chapter 7.6.

Since the AUTOSAR COM module is not aware of the underlying bus properties, it has to be configured if an I-PDU can be transmitted within a single L-PDU, e.g. one CAN frame, or if it needs to be transmitted via TP, see also ECUC\_Com\_00761.

The AUTOSAR COM module is not able to prevent or to detect if too large I-PDUs are sent to a specific bus transport protocol, because the AUTOSAR COM module uses the bus-independent PDU Router interface. However, the AUTOSAR COM module allows that the maximum length of the N-SDU of the underlying TPs can be used and therefore the AUTOSAR COM module does not introduce any additional length restrictions.

### 7.5.1 Transmission of Large Signals/ I-PDUs

The AUTOSAR COM module sends large I-PDUs via the generic PduR APIs for upper layer modules that use transport protocol. For details of the AUTOSAR COM module – PDU Router interface see Chapter 7.7. According to [10] the call-sequence is:

- PduR\_ComTransmit: for initiating the send request
- one or more callbacks to Com\_CopyTxData: for copying the data
- callback to Com\_TpTxConfirmation: for confirming the transmission

**[SWS\_Com\_00662]** [After successful initiation of a transmission of a large I-PDU via PduR\_ComTransmit, the AUTOSAR COM module shall not modify the data of its internal transmit buffer of this I-PDU unless the AUTOSAR COM module is notified about the successful transmission via Com\_TpTxConfirmation or it is notified by an error indicating that the transmission was aborted.] (SRS\_Com\_02095)

The I-PDUs within the AUTOSAR COM module are statically configured. Therefore, the complete memory of the I-PDU, even for very large I-PDUs, will be allocated. This approach was taken in order to reduce the complexity of the large data type handling.

If the utilized transportation protocol is also used for diagnosis, or other services, it must support multiple users. This is at least supported by CAN TP and FlexRay TP.

**[SWS\_Com\_00863]** [In case a large I-PDU is currently transmitted and the same I-PDU is triggered for transmission again, the AUTOSAR COM shall skip the addi-

tionally send request and report the runtime error `COM_E_SKIPPED_TRANSMISSION.]` (SRS\_Com\_02095)

This could for example happen in case a large I-PDU is sent out periodically and the transmission is delayed over the next cycle.

### 7.5.2 Reception of Large Signals/ I-PDUs

The AUTOSAR COM module receives large I-PDUs via the generic PDU Router's APIs for upper layer modules that use transport protocol. For details of the AUTOSAR COM module – PDU Router interface see Chapter 7.8. According to [10] the call-sequence is:

- call to `Com_StartOfReception`: for initiating the RX request
- one or more callbacks to `Com_CopyRxData`: copying the receive data to the AUTOSAR COM module's internal receive buffer
- call to `Com_TpRxIndication`: indicating the correct, or incorrect, end of the reception process

**[SWS\_Com\_00838]** [In case `Com_TpRxIndication` is called with value `E_NOT_OK` for the parameter result, the AUTOSAR COM module shall treat all signals of the received large I-PDU exactly like invalidated signals. That is, the AUTOSAR COM module shall execute all configured `ComDataInvalidActions` for all included signals and signal groups.] (SRS\_Com\_02095)

## 7.6 Dynamic Length Signals

A **dynamic length signal** is a signal which length can vary at run-time. The maximal length of a dynamic length signal needs to be specified at configuration time. A dynamic length signal has to be transmitted in a **dynamic length I-PDU**.

**[SWS\_Com\_00753]** [For dynamic length signals, the AUTOSAR COM module shall only support the `ComSignalType UINT8_DYN.`] (SRS\_Com\_02094, SRS\_Com\_02098)

Restricting the type to an `UINT8`-array type also restricts the placement of the dynamic length signal to byte boundaries.

All other signals, potentially including large signals, must be packed in front of the dynamic length signal (see `SWS_Com_00754`, `SWS_Com_00755` and `SWS_Com_00756`). Therefore, it is not required to encode the length of the dynamic length signal within the I-PDU. On receiver side, the length of the dynamic length signal can be calculated from the length of the received I-PDU.

It is allowed to configure an update-bit for a dynamic length signal. In this case, the update-bit must be located in front of the dynamic length signal, see also `SWS_Com_00755`.

There is no restriction in the AUTOSAR COM module preventing the usage of signal groups in combination with large or dynamic length signals.

Dynamic length signals are not restricted to be sent via transport protocol. However, the maximum length of the I-PDU containing the dynamic length signal must respect the bus-specific restrictions.

In case of I-PDUs with dynamic length signals that shall be sent via CAN FD it is recommended to use the Multi PDU Mapping feature of the IpduM. This avoids misinterpretation of the received length due to the discrete data lengths of CAN FD frames.

### 7.6.1 Transmission of Dynamic Length Signals/ I-PDUs

For sending a dynamic length signal, the RTE needs to specify the actual length at the send call. The corresponding API is Com\_SendDynSignal (SWS\_Com\_00627).

**[SWS\_Com\_00757]** [At a call to Com\_SendDynSignal, the AUTOSAR COM module shall set the length of the corresponding dynamic length I-PDU to the smallest length that includes the dynamic length signal. The length of the dynamic signal shall be equal to the value of input parameter “Length” passed in the API Com\_SendDynSignal.] (SRS\_Com\_02093)

The AUTOSAR COM module is able to send dynamic length I-PDUs via normal or large I-PDUs. In any case, it needs to pass the length of the dynamic I-PDU to the lower layers.

**[SWS\_Com\_00832]** [When the AUTOSAR COM module invokes PduR\_ComTransmit for a dynamic length I-PDU, the SduLength of the PduInfoType shall be set according the last update (or init value if not yet sent) of the included dynamic length signal.] (SRS\_Com\_02093)

### 7.6.2 Reception of Dynamic Length Signals/ I-PDUs

For receiving a dynamic length signal, the RTE needs to be informed about the actual size.

**[SWS\_Com\_00758]** [At reception of a dynamic length I-PDU, the AUTOSAR COM module shall calculate the length of the contained dynamic length signal, by:

<dynamic signal length in bytes> = <received I-PDU length in bytes>

- <ComBitPosition of dynamic length signal> / 8

] (SRS\_Com\_02093)

Com\_ReceiveDynSignal (SWS\_Com\_00690) returns the length of a dynamic length signal, calculated as defined by SWS\_Com\_00758.

Since, the length of a dynamic length signal is derived from the received length of the I-PDU, the AUTOSAR COM module does not support the combination of dynamic length I-PDUs and I-PDUs with metadata, see SWS\_Com\_00817.

## 7.7 Interface between AUTOSAR COM Module and the PDU Router

OSEK COM leaves the interface between OSEK COM and the lower layers undefined. In AUTOSAR, the only lower layer that the AUTOSAR COM module interfaces to is the PDU Router.

The AUTOSAR COM module uses the PDU Router in two different modes, depending on the type of the I-PDU it will be transported with the bus-specific transport protocol or without. If an I-PDU is supposed to be sent via TP, is configured per configuration parameter ComIPduType.

**[SWS\_Com\_00138]** [The AUTOSAR COM module shall send out I-PDUs by a calling the PduR\_ComTransmit function.] (SRS\_Com\_02037)

**[SWS\_Com\_00759]** [If the AUTOSAR COM module sends out an I-PDU with ComIPduType configured to *NORMAL*, the AUTOSAR COM module shall expect to be called by the PDU Router using Com's interfaces for modules that do not use TP. See column *normal I-PDUs* in Table 6.] (SRS\_Com\_02096)

**[SWS\_Com\_00760]** [If the AUTOSAR COM module sends out an I-PDU with ComIPduType configured to *TP*, the AUTOSAR COM module shall expect to be called by the PDU Router using Com's interfaces for modules that use TP. See column *TP I-PDUs* in Table 6.] (SRS\_Com\_02096)

The interaction diagram in Chapter 9.1 shows the interaction between the PDU Router module and the AUTOSAR COM module for normal I-PDUs.

Table 6 gives an overview, which API between the AUTOSAR COM module and the PDU Router is used for a concrete I-PDU with respect to the configured value of ComIPduType. The API description itself is specified in Chapter 8.

API	<i>normal I-PDUs</i>	<i>TP I-PDUs</i>
PduR_ComTransmit	X	X
Com_TxConfirmation	X	-
Com_RxIndication	X	-
Com_TriggerTransmit	X	-
Com_TpTxConfirmation	-	X
Com_TpRxIndication	-	X
Com_StartOfReception	-	X
Com_CopyRxData	-	X
Com_CopyTxData	-	X

Table 6: API to PDU Router with respect to ComIPduType

## 7.8 Update-Bits

To enable the receiver of a signal/ signal group to identify whether the sender has updated the data in this signal/ signal group before sending, the AUTOSAR COM module supports *update-bits*.

The update-bits indicate whether the RTE on sender-side has updated a signal value, before the I-PDU, containing that signal, was transmitted to the PDU Router.

Update-bits are not allowed if transmission mode *DIRECT* is used with ComTxMode-NumberOfRepetitions greater or equal 1 (see SWS\_Com\_00310).

By configuration on sender- and on receiver-side, it is possible to add separately for each signal and/or separately for each signal group at most one update-bit.

The position of the update-bit is configurable by configuration parameter ComUpdateBitPosition (ECUC\_Com\_00257). ComUpdateBitPosition is included within the configuration container ComSignal. Hence, it is ensured that the signal/ signal group and the corresponding update-bit are always part of the same I-PDU.

**[SWS\_Com\_00055]** [The AUTOSAR COM module shall handle the update-bit only internally and not as part of the signal or signal group.] (SRS\_Com\_02030)

The update-bits are not directly visible or accessible by AUTOSAR Software Components.

**[SWS\_Com\_00059]** [The AUTOSAR COM module shall interpret the update-bit as defined in Table 7.] (SRS\_BSW\_00101, SRS\_Com\_02030)

<i>Update-BIT</i>	
0	cleared/ data has not been updated
1	set/ data has been updated

**Table 7 update-bit interpretation**

### 7.8.1 Sender Side

The initialization of update-bits is defined by SWS\_Com\_00117.

**[SWS\_Com\_00061]** [If the RTE updates the value of a signal by calling Com\_SendSignal, the AUTOSAR COM module shall set the update-bit of this signal.] (SRS\_Com\_02030)

**[SWS\_Com\_00801]** [If the RTE updates a signal group by calling Com\_SendSignalGroup, the AUTOSAR COM module shall set the update-bit of this signal group.] (SRS\_Com\_02030)

**[SWS\_Com\_00062]** [If the parameter ComTxIPduClearUpdateBit of an I-PDU is configured to *Transmit*, the AUTOSAR COM module shall clear all update-bits of all contained signals and signal groups after this I-PDU was sent out via PduR\_ComTransmit and PduR\_ComTransmit returned E\_OK.] (SRS\_Com\_02030)

**[SWS\_Com\_00577]** [If the parameter ComTxIPduClearUpdateBit of an I-PDU is configured to *Confirmation*, the AUTOSAR COM module shall clear all update-bits of all contained signals and signal groups after this I-PDU was sent out via PduR\_ComTransmit, PduR\_ComTransmit returned E\_OK and the I-PDU was successfully confirmed.] (SRS\_Com\_02030)

**[SWS\_Com\_00578]** [If the parameter ComTxIPduClearUpdateBit of an I-PDU is configured to *TriggerTransmit*, the AUTOSAR COM module shall clear all update-bits of all contained signals and signal groups after the contents of this I-PDU was successfully requested by Com\_TriggerTransmit.] (SRS\_Com\_02030)

## 7.8.2 Receiver Side

**[SWS\_Com\_00324]** [If a signal or signal group with an attached update bit is received, the AUTOSAR COM module shall only process this signal, i.e. filter, notification, signal based gateway, byte swapping, if the signal's update bit is set.] (SRS\_Com\_02030)

**[SWS\_Com\_00802]** [If a signal or signal group with an attached update bit is received and the update-bit is not set, the AUTOSAR COM module shall discard this signal.] (SRS\_Com\_02030)

If the signal has not been updated, the signal will not be routed via the signal gateway. It will only be discarded.

If the RTE reads a signal with an associated cleared update-bit, the init value or the last received value is returned.

**[SWS\_Com\_00067]** [The AUTOSAR COM module shall interpret a signal/ signal group as *updated* if the signal has an update-bit attached, and the value of the update-bit is set.] (SRS\_Com\_02030)

For the behavior of reception deadline monitoring on signals with update-bits, see Chapter 7.3.6.1.

## 7.9 Data Sequence Control

The AUTOSAR COM module provides mechanisms of data sequence control in the form of I-PDU Counters. *Out of sequence* I-PDUs, those are repeated I-PDUs or I-PDUs received after missing I-PDUs, will be detected and discarded.

I-PDU counters are optional and can be configured per I-PDU. For the corresponding configuration container see ECUC\_Com\_00592.

### 7.9.1 Sender Side

**[SWS\_Com\_00687]** [For all I-PDUs with ComIPduDirection configured to SEND that have a configured ComIPduCounter, the AUTOSAR COM module shall set the I-PDU counter to 0 within Com\_Init and within Com\_IpduGroupControl if parameter Initialize is true.] (SRS\_Com\_02101)

**[SWS\_Com\_00688]** [The AUTOSAR COM module shall increase the I-PDU counter of an I-PDU by one, with respect to counter wrap around, directly before the I-PDU is



sent out by PduR\_ComTransmit. In case the return value of PduR\_ComTransmit is not equal to E\_OK the update of the I-PDU counter shall be revoked (by decrementing the I-PDU counter by one).] (SRS\_Com\_02101)

If Com\_TriggerTransmit fetches an I-PDU containing an I-PDU counter, no incrementation takes place. Hence, care must be taken if I-PDU counters are used in combination with Com\_TriggerTransmit. If Com\_TriggerTransmit is called multiple times after the corresponding PduR\_ComTransmit all I-PDUs will contain the same counter value. Thus, a deferred TriggerTransmit transmission can work, but a pure lower layer driven TriggerTransmit use-case would not work in combination with an I-PDU counter.

## 7.9.2 Receiver Side

**[SWS\_Com\_00587]** [For all I-PDUs with ComIPduDirection configured to RECEIVE that have a configured ComIPduCounter, the AUTOSAR COM module shall accept any incoming I-PDU, regardless of the value of the I-PDU counter, after the I-PDU was initialized by Com\_Init or reinitialized by Com\_IpduGroupControl with parameter Initialize set to true.] (SRS\_Com\_02099)

**[SWS\_Com\_00588]** [At reception of an I-PDU containing an I-PDU counter, the AUTOSAR COM module shall set the next expected value to the value following the received value with respect to counter wrap-around.] (SRS\_Com\_02101)

**Example:** If the I-PDU counter has 4 bits its possible values are 0 – 15. If an I-PDU with a 4-bit sized counter is received and the received counter value is 15 the next expected value is 0. The new expected value needs to be updated regardless if the received value matches the actual expected value or not.

**[SWS\_Com\_00590]** [At reception of an I-PDU containing an I-PDU counter, the AUTOSAR COM module shall discard the I-PDU, after setting the next expected value as defined in SWS\_Com\_00588, if no I-PDU replication for this I-PDU is configured and:

received I-PDU counter < expected I-PDU counter OR  
received I-PDU counter > expected I-PDU counter  
+ ComIPduCounterThreshold

with respect to counter wrap-around.] (SRS\_Com\_02102)

**[SWS\_Com\_00727]** [In case a mismatch of the expected and the received I-PDU counter is detected, the AUTOSAR COM module shall notify this mismatch by the configured ComIPduCounterErrorNotification (ECUC\_Com\_00003) notification function.] (SRS\_Com\_02102)

The counter error notification is triggered independently of the configured ComIPduCounterThreshold.

The signature of this I-PDU counter error notification function is defined in SWS\_Com\_00726.

**Example:** In the example in Table 8, the I-PDU counter is 4-bit wide (values 0-15) and ComIPduCounterThreshold is configured to 1:

<i>Expected I-PDU counter</i>	<i>Received counter value</i>	<i>Action</i>
6	6	<i>normal reception</i>
6	7	<i>normal reception</i>
6	8	<i>discarded</i>
6	1	<i>discarded</i>
15	15	<i>normal reception</i>
15	0	<i>normal reception</i>
15	1	<i>discarded</i>

Table 8 Example I-PDU counter handling

## 7.10 Communication Protection

Safety related I-PDUs might need to be replicated in multiple L-PDUs in order to prevent corruption and loss of data. In AUTOSAR the replication of I-PDUs is realized by the PDU Router. The comparison and voting of replicated I-PDUs is performed in AUTOSAR COM. Since the replicated I-PDUs have no checksum or signature for comparison it is needed to store and compare the complete I-PDUs at the receiver. The AUTOSAR COM allows configuring two additional replicated I-PDUs via ComIPduReplicaRef (ECUC\_Com\_00601).

The number of successfully equally received I-PDUs needed for a qualified reception can be configured via ComIPduReplicationQuorum (ECUC\_Com\_00600) from 1 to 3. The PDU Router module does not support 1:n routing of I-PDUs which are sent or received via a TP module and require multiple frames for transmission. Hence, the AUTOSAR COM stack does currently not support replication of large or dynamic I-PDUs or data types.

### 7.10.1 Sender Side

Since the replication itself is done by the PduR, there is no special handling on sender side for AUTOSAR COM. However, it must be ensured that replicated I-PDUs have configured an I-PDU counter, see SWS\_Com\_00834. This is required since the voting at the receiver relies on the I-PDU counter.

### 7.10.2 Receiver Side

At receiver side, the AUTOSAR COM module performs a voting about the multiple replicated I-PDUs. The AUTOSAR COM module passes only signals of successfully confirmed I-PDUs to the RTE. As stated above replicated I-PDUs need to have an I-PDU counter configured. However, the ComIPduCounterThreshold must be configured to 0, see ECUC\_Com\_00595. There is no use-case of mixing these two features and the voting would be more complex.

**[SWS\_Com\_00596]** [Only when the AUTOSAR COM module has received at least *ComIPduReplicationQuorum* identical copies of a replicated I-PDU, it shall provide the signals and/ or signal groups out of this I-PDU to the RTE.] (SRS\_Com\_02103, SRS\_Com\_02105, SRS\_Com\_02106)

A replicated I-PDU is configured as a normal I-PDU within the AUTOSAR COM module's configuration. The fan-out on sender-side is configured within the PduR configuration. On reception-side, additionally up to two replicas of PDU-R PDUs (see ECUC\_Com\_00601 *ComIPduReplicaRef*) and the necessary number of identically received I-PDUs for successfully voting (see ECUC\_Com\_00600 *ComIPduReplicationQuorum*) can be configured within the AUTOSAR COM module's configuration.

**[SWS\_Com\_00597]** [When the AUTOSAR COM module has received at least *ComIPduReplicationQuorum* identical copies of a replicated I-PDU, it shall notify the RTE only once with respect to the signals and/or signal groups out of this I-PDU.] (SRS\_Com\_02103, SRS\_Com\_02105, SRS\_Com\_02106)

The I-PDU counter will be handled for each replicated I-PDU separately as defined in Chapter 7.9.

Since the I-PDUs do not contain a special checksum, it is expected, that the comparison is performed by comparing the complete I-PDU byte by byte.

For I-PDU replication, the *ComIPduCounterThreshold* needs to be configured to 0, see SWS\_Com\_00834. Hence, *ComIPduReplicaRef* + 1 buffers are sufficient to store the I-PDUs for a successful voting.

### 7.10.3 Constraints

Due to the approach of reusing already existing functionality, that is using the PduR for the fan out, care must be taken to set up the system correctly. The PduR allows fanning out the replicated I-PDUs on different busses. On the receiver side, these I-PDUs are indicated to COM with the same I-PDU id. However, *Com\_RxIndication* (SWS\_Com\_00123) may not be called reentrant for the same I-PDU id. An easy way to prevent this is to ensure that all I-PDUs are received by the same Interface module, which results in using the same bus for all replicas. For the current AUTOSAR stack, this is the recommend method. However, the system integrator could also synchronize the reception path by other means.

Furthermore, since there is no mechanism to prevent updates from the upper layer to an I-PDU, this approach of communication protection does not work in combination of decoupled transmission, i.e. I-PDUs requested via *Com\_TriggerTransmit*.

## 7.11 Signal Gateway

The signal gateway is an integrated part of the AUTOSAR COM module. Any external modules, except the cyclic task call, cannot access the signal gateway.

The AUTOSAR COM module's signal gateway works with (group) signals and signal groups.

The AUTOSAR COM module's signal gateway supports only static routing for the ComGwMapping configuration container (ECUC\_Com\_00544). It is only possible to configure routes independently of the contents of the routed signals and signal groups.

The destination of a signal or signal group can be configured either per ComGwDestinationDescription container (ECUC\_Com\_00549), or via ComGwSignal reference (ECUC\_Com\_00551). The transmission mode will be selected based on the TMCs of other associated signals of the I-PDU. In case no signal contributes to the TMS calculation, the transmission mode ComTxModeTrue (SWS\_Com\_00677) will be used.

**[SWS\_Com\_00377]** [The AUTOSAR COM module's signal gateway shall copy the value of signals/ signal groups to be routed to the signals/ signal groups for transmission according to configuration, see ECUC\_Com\_00544.] (SRS\_PduR\_06055, SRS\_PduR\_06089)

The ComGwMapping configuration allows to configure routing a signal/ signal group from one source signal/ signal group to zero (no signal gateway functionality) or more destinations (1:n).

The AUTOSAR COM module's signal gateway does not support to process signals or signal groups that are contained within a large I-PDU, see SWS\_Com\_00598.

### 7.11.1 Dealing with Signals

**[SWS\_Com\_00357]** [The AUTOSAR COM module shall forward signals to be routed from received I-PDUs to transmit I-PDUs. For configuration, see ECUC\_Com\_00544 configuration container ComGwMapping.] (SRS\_PduR\_06002)

Stopping the receive I-PDU after the signals to be routed have been forwarded to the transmit I-PDU will not prevent the transmit I-PDU from being transmitted unless it is stopped as well.

**[SWS\_Com\_00360]** [If the endianness of a received signal to be routed differs from the endianness of a related destination signal, the AUTOSAR COM module shall convert its endianness to the endianness of the destination signal.] (SRS\_PduR\_06061)

### 7.11.2 Dealing with Signal Groups

**[SWS\_Com\_00361]** [The AUTOSAR COM module shall forward to be routed signal groups from received I-PDUs to transmit I-PDUs. See also configuration container ComGwMapping (ECUC\_Com\_00544).] (SRS\_PduR\_06002, SRS\_PduR\_06056, SRS\_PduR\_06061)

Stopping the receive I-PDU after the signals groups to be routed have been forwarded to the transmit I-PDU will not prevent the transmit I-PDU from being transmitted unless it is stopped as well.

**[SWS\_Com\_00383]** [The AUTOSAR COM module shall route signal groups in a consistent manner. Therefore, the AUTOSAR COM module shall transfer the data of a signal group as one consistent set of data during a routing operation.] (SRS\_PduR\_06056)

**[SWS\_Com\_00735]** [The AUTOSAR COM module shall support routing consistently a subset of group signals of a source signal group into a reduced target signal group.] (SRS\_PduR\_06056)

**[SWS\_Com\_00833]** [The AUTOSAR COM module shall support routing of individual group signals of a source signal group to a individual target signals (which are not part of signal group).] (SRS\_PduR\_06056)

**[SWS\_Com\_00362]** [If the endianness of a received group signal to be routed differs from the endianness of a related destination (group) signal, the AUTOSAR COM module shall convert its endianness to the endianness of the destination (group) signal.] (SRS\_PduR\_06061)

### 7.11.3 Routing of Out-Timed Signals and Signal Groups

**[SWS\_Com\_00701]** [The AUTOSAR COM module's signal gateway shall route signal and signal groups even if any configured reception deadline monitoring timeout expired.] (SRS\_PduR\_06089)

In case of a not in time received signal or signal group the AUTOSAR COM module's signal gateway will route these signal or signal group anyway.

### 7.11.4 Handling of Update-Bits

**[SWS\_Com\_00702]** [If both, the received signal/ signal group and the destination signal/ signal group have an update-bit (ComUpdateBitPosition) configured and the update-bit of the received signal/ signal group is set, the AUTOSAR COM module shall route the signal/ signal group with the set update-bit and clear the update-bit of the destination signal/ signal group after it was sent.] (SRS\_Com\_02030)

**[SWS\_Com\_00703]** [If the received signal/ signal group and the destination signal/ signal group have an update-bit (ComUpdateBitPosition) configured, and the update-bit of the received signal/ signal group is not set, the AUTOSAR COM module shall not route this signal/ signal group. ] (SRS\_Com\_02030)

**[SWS\_Com\_00704]** [If the received signal/ signal group has an update-bit (ComUpdateBitPosition) configured, but the destination signal has no update-bit configured, and the update-bit is set, the AUTOSAR COM module shall route this signal/ signal group without the update-bit.] (SRS\_Com\_02030)

**[SWS\_Com\_00705]** [If the received signal/ signal group has an update-bit (ComUpdateBitPosition) configured, but the destination signal has no update-bit configured, and the update-bit is not set, the AUTOSAR COM module shall not route this signal/ signal group.] (SRS\_Com\_02030)

**[SWS\_Com\_00706]** [If the received signal/ signal group has no update-bit (ComUpdateBitPosition) configured and the destination signal/ signal group has an update-bit configured, the AUTOSAR COM module shall set the update-bit of the destination signal when a new signal/ signal group was received and clear it after sending of the destination signal/ signal group. ] (SRS\_Com\_02030)

### 7.11.5 Decoupling Signal Gateway

To protect interrupt routines used for I-PDU reception from incalculable and perhaps expensive time usage, it is necessary to decouple the AUTOSAR COM module's signal gateway from interrupt routines.

**[SWS\_Com\_00359]** [The AUTOSAR COM module shall execute all functions of its signal gateway during the function call Com\_MainFunctionRouteSignals (SWS\_Com\_00400). During this function call, the AUTOSAR COM module's signal gateway shall check received and to be routed signals and signal groups and forward them from the related receive I-PDUs to the related transmit I-PDUs.] (SRS\_BSW\_00425, SRS\_BSW\_00432)

**[SWS\_Com\_00466]** [Within Com\_MainFunctionRouteSignals, the AUTOSAR COM module shall evaluate the transfer properties and transmission modes and perform its routing actions in the following sequence, see also Figure 3:

1. copy all gated signals from the source to the target I-PDUs
2. evaluate the TMC of all gated signals
3. evaluate the TMS for the target I-PDUs

] (SRS\_BSW\_00432)

In the case, that a target I-PDU contains gated signals with ComTransferProperty TRIGGERED that are newly received, the sending may be performed in either Com\_MainFunctionTx or Com\_MainFunctionRouteSignals.

**[SWS\_Com\_00539]** [The AUTOSAR COM module shall sent out an I-PDU at most once while one call to Com\_MainFunctionRouteSignals.] (SRS\_PduR\_06055)

### 7.12 Error Classification

The AUTOSAR COM module supports currently no production errors.

Type of error	Relevance	Related error code	Value [hex]
[SWS_Com_00803] [API service called with wrong parameter](SRS_BSW_00337)	Development	COM_E_PARAM	0x01
[SWS_Com_00804] [Error code if any other API service, except Com_GetStatus, is called before the AUTOSAR COM module was initialized with Com_Init or after a call to Com_Deinit](SRS_BSW_00337)	Development	COM_E_UNINIT	0x02
[SWS_Com_00805] [NULL pointer checking](SRS_BSW_00414)	Development	COM_E_PARAM_POINTER	0x03
[SWS_Com_00837] [Invalid configuration set selection](SRS_BSW_00414)	Development	COM_E_INIT_FAILED	0x04
[SWS_Com_00864] [Transmission request was skipped] (SRS_BSW_00452)	Runtime	COM_E_SKIPPED_TRANSMISSION	0x05

Table 9: Mapping of the AUTOSAR COM module's development error IDs

## 7.13 Error Notification

[SWS\_Com\_00442] [When a development error is detected, the function Det\_ReportError of the development error tracer shall be called with:

- 50 as the AUTOSAR COM's ModuleId
- 0 as InstanceId
- the service ID of the AUTOSAR COM module's API in which the error was detected (see Com\_ServiceIdType) as ApiId
- the error ID as defined in Table 9 as ErrorId

] (SRS\_BSW\_00369, SRS\_BSW\_00385, SRS\_BSW\_00327, SRS\_PduR\_06098, SRS\_PduR\_06099)

## 7.14 Error Handling

AUTOSAR COM supports supervision of the communication with deadline monitoring, see Chapter 7.3.6. Further, the AUTOSAR COM supports data sequence control see Chapter 7.9 and communication protection see Chapter 7.10.

The concrete reaction of the AUTOSAR COM module with respect to error and return values of called APIs is stated within the corresponding functional requirements.

## 7.15 AUTOSAR COM Module's Interaction Model

This chapter corresponds to the chapter *Functional Model of Interaction Layer* of [17]. The following figures illustrate the behavior of the Interaction layer for external recep-

tion and external transmission. The complete functionality is shown but it depends on the configuration what parts are present/ used in a concrete implementation.

**[SWS\_Com\_00396]** [A received signal can be configured to have filtering, data invalidation and notification. The AUTOSAR COM module shall execute these services, if configured, in the following order:

- 1) Data invalidation
- 2) Filtering
- 3) Notification

] (SRS\_Com\_02037)

**[SWS\_Com\_00744]** [In case both a ComRxDataTimeoutAction and a ComTimeoutNotification is configured for a ComSignal or a ComSignalGroup, the AUTOSAR COM module shall first call the configured ComRxDataTimeoutAction and then call the configured ComTimeoutNotification.] (SRS\_Com\_02089)



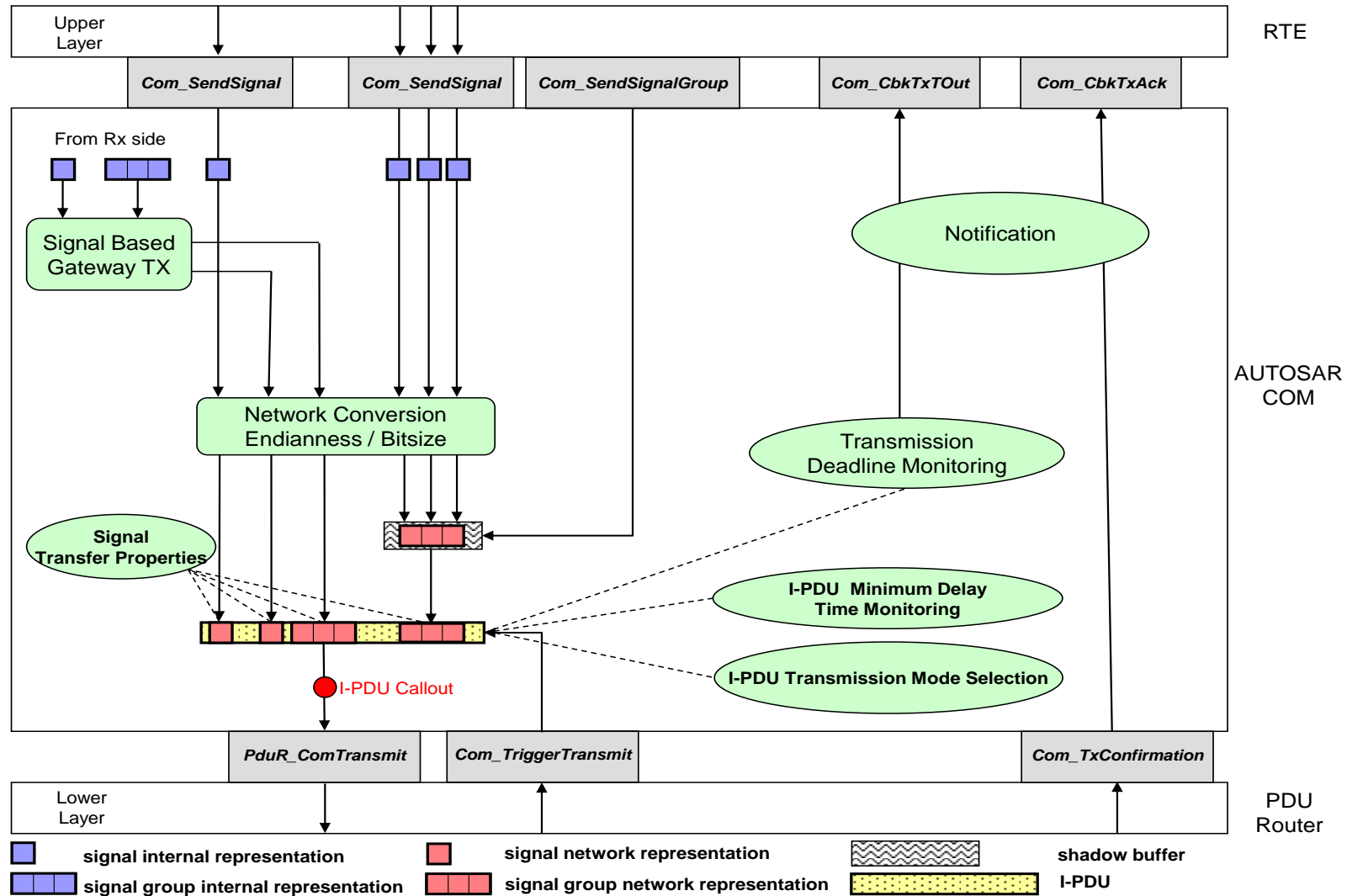


Figure 5 The AUTOSAR COM module's interaction model for transmission

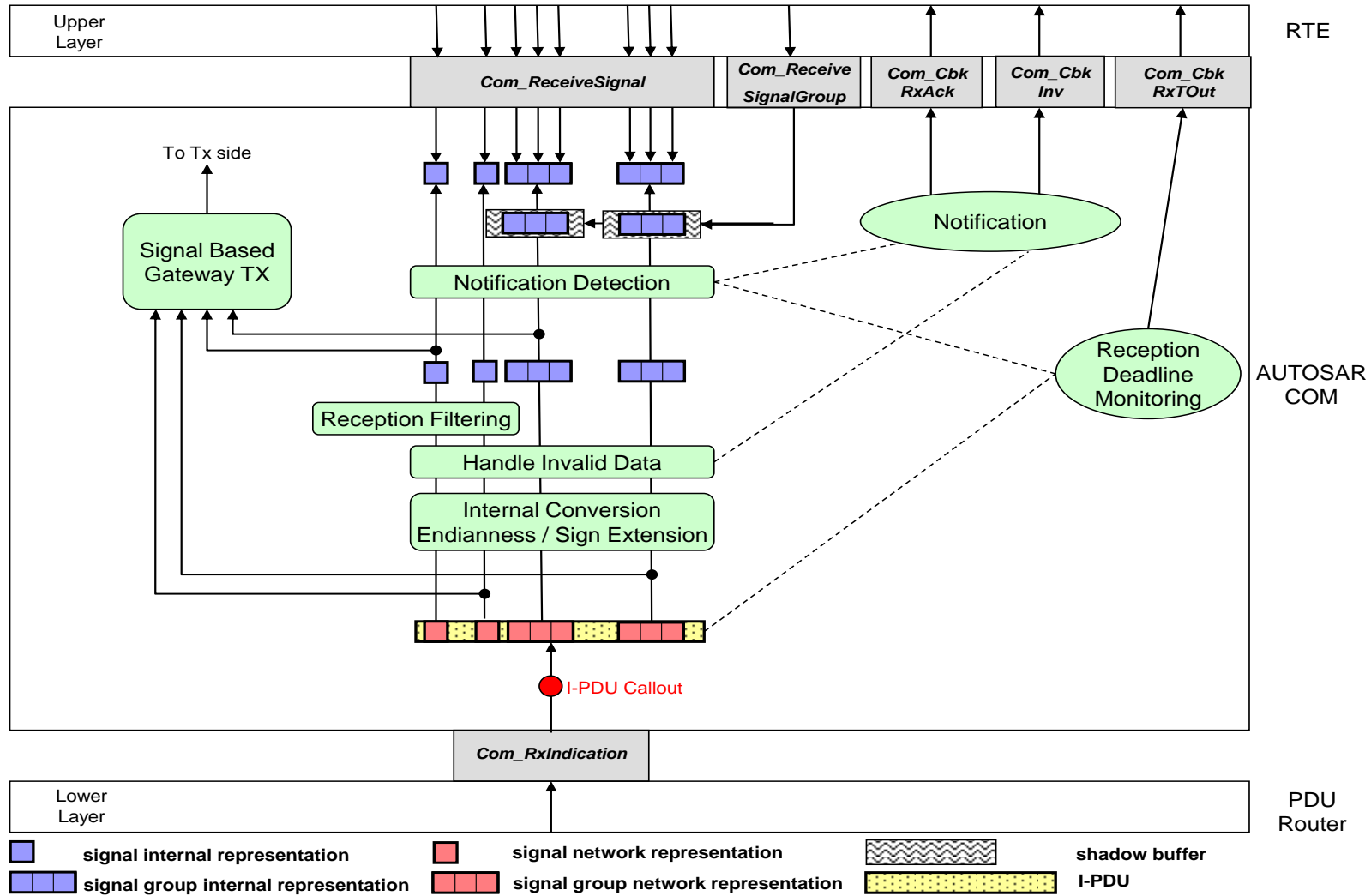


Figure 6 The AUTOSAR COM module's interaction model for reception

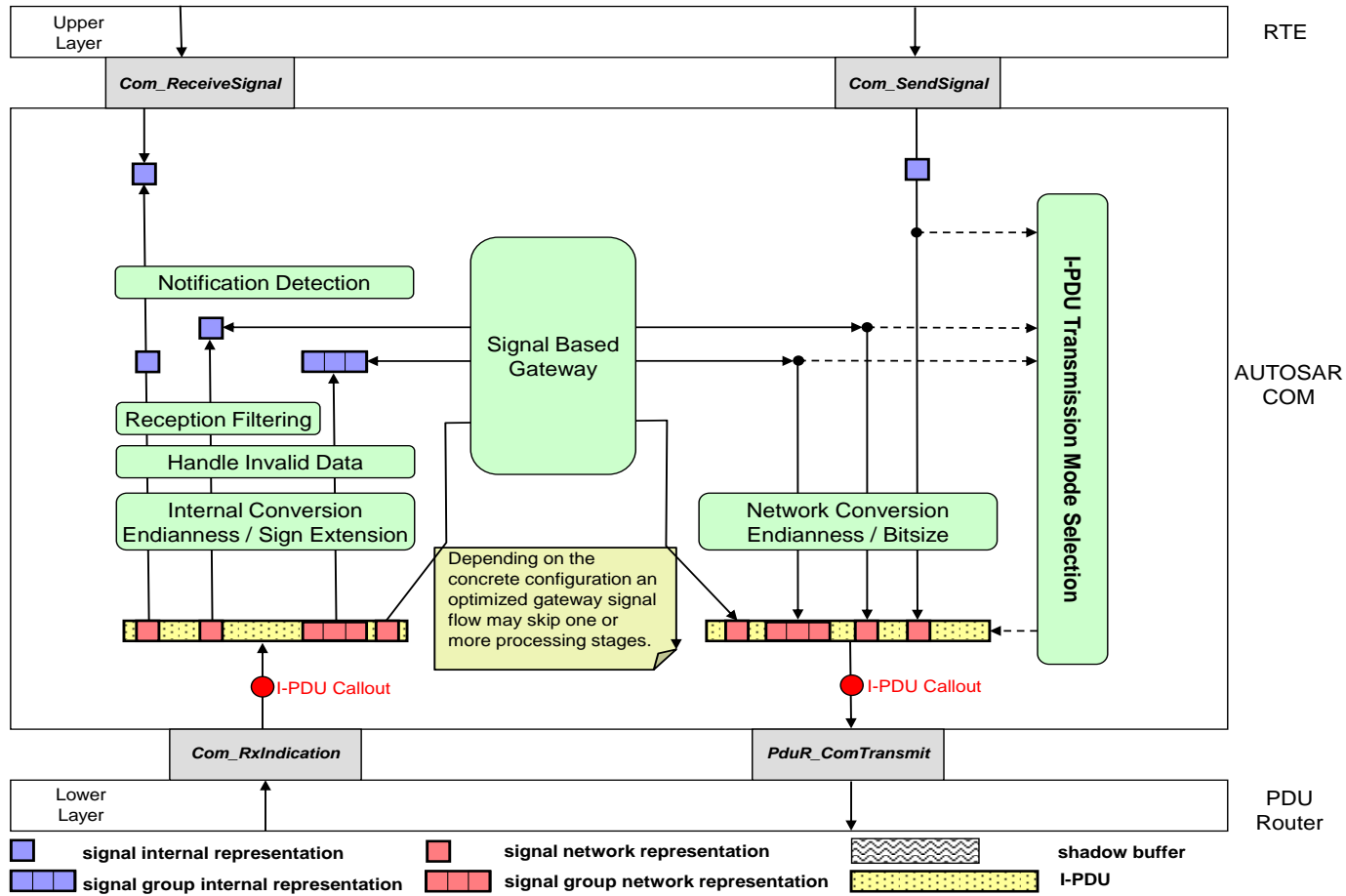


Figure 7: The AUTOSAR COM module's interaction model for integrated Signal Gateway

The endianness conversion and sign extension on receiver side are needed to feed the TMS with a correct data format. This endianness conversion is only necessary if the endianness of the Rx-bus differs from the endianness of the CPU. The endianness conversion on the sender side is only necessary if the endianness of the Rx-bus differs from the endianness of the Tx-bus.

## 8 API Specification

### 8.1 Imported Types

In this chapter, all types included from the following files are listed:

#### [SWS\_Com\_00609] [

<i>Module</i>	<i>Imported Type</i>
ComStack_Types	BufReq_ReturnType
	PduIdType
	PduInfoType
	PduLengthType
	RetryInfoType
Std_Types	Std_ReturnType
	Std_VersionInfoType

] (SRS\_BSW\_00301)

### 8.2 Type Definitions

#### 8.2.1 Com\_StatusType

##### [SWS\_Com\_00819] [

<b>Name:</b>	Com_StatusType	
<b>Type:</b>	Enumeration	
<b>Range:</b>	COM_INIT	The AUTOSAR COM module is initialized and usable.
	COM_UNINIT	The AUTOSAR COM module is not initialized or not usable. This shall be the default value after reset. This status shall have the value 0.
<b>Description:</b>	This is a status value returned by the API service Com_GetStatus().	

] (SRS\_BSW\_00335)

#### 8.2.2 Com\_SignalIdType

##### [SWS\_Com\_00820] [

<b>Name:</b>	Com_SignalIdType	
<b>Type:</b>	uint16	
<b>Range:</b>	0..<SignalIdmax>	-- Zero-based integer number
<b>Description:</b>	The AUTOSAR COM module's signal object identifier.	

] (SRS\_Com\_02037, SRS\_BSW\_00441)

#### 8.2.3 Com\_SignalGroupIdType

##### [SWS\_Com\_00821] [

<b>Name:</b>	Com_SignalGroupIdType	
<b>Type:</b>	uint16	
<b>Range:</b>	0..<SignalGroupIdmax>	-- Zero-based integer number

<b>Description:</b>	The AUTOSAR COM module's signal group object identifier.
---------------------	--

] (SRS\_Com\_02041, SRS\_BSW\_00441)

## 8.2.4 Com\_IpduGroupIdType

[SWS\_Com\_00822] [

<b>Name:</b>	Com_IpduGroupIdType		
<b>Type:</b>	uint16		
<b>Range:</b>	0..<IpduGroupId-max>	--	Zero-based integer number; where IpduGroupId-max < ComSupportedIPduGroups
<b>Description:</b>	The AUTOSAR COM I-PDU module's group object identifier.		

] (SRS\_Com\_00218, SRS\_BSW\_00441)

## 8.2.5 Com\_IpduGroupVector

[SWS\_Com\_00823] [

<b>Name:</b>	Com_IpduGroupVector		
<b>Type:</b>	uint8[(ComSupportedIPduGroups-1)/8+1]		
<b>Range:</b>	bitfield	--	The bitfield is an array of uint8[(ComSupportedIPduGroups - 1)/8 + 1], i.e. there are bit0 - bit<ComSupportedIPduGroups - 1>
<b>Description:</b>	This type can be used to store a flag (bit) for each I-PDU group within the system. It is used for setting the activation state and deadline monitoring state for I-PDU groups within one function call		

] (SRS\_Com\_02090, SRS\_BSW\_00441)

## 8.2.6 Com\_ServiceIdType

[SWS\_Com\_00824] [

<b>Name:</b>	Com_ServiceIdType		
<b>Type:</b>	uint8		
<b>Range:</b>	COMServiceId_Init	0x01	--
	COMServiceId_DeInit	0x02	--
	COMServiceId_IpduGroupControl	0x03	--
	COMServiceId_ReceptionDMControl	0x06	--
	COMServiceId_GetStatus	0x07	--
	COMServiceId_GetVersionInfo	0x09	--
	COMServiceId_SendSignal	0x0A	--
	COMServiceId_ReceiveSignal	0x0B	--
	COMServiceId_UpdateShadowSignal	0x0C	--
	COMServiceId_SendSignalGroup	0x0D	--
	COMServiceId_ReceiveSignalGroup	0x0E	--
	COMServiceId_ReceiveShadowSignal	0x0F	--
	COMServiceId_InvalidSignal	0x10	--
	COMServiceId_InvalidShadowSignal	0x16	--
	COMServiceId_TriggerIPDUSend	0x17	--
	COMServiceId_MainFunctionRx	0x18	--
	COMServiceId_MainFunctionTx	0x19	--
	COMServiceId_MainFunctionRouteSignals	0x1A	--
	COMServiceId_InvalidSignalGroup	0x1B	--

	COMServiceId_ClearIpduGroupVector	0x1C	--
	COMServiceId_SetIpduGroup	0x1D	--
	COMServiceId_SendDynSignal	0x21	--
	COMServiceId_ReceiveDynSignal	0x22	--
	COMServiceId_SendSignalGroupArray	0x23	--
	COMServiceId_ReceiveSignalGroupArray	0x24	--
	COMServiceId_SwitchIpduTxMode	0x27	--
	COMServiceId_TriggerIPDUSendWithMetaData	0x28	--
	COMServiceId_TxConfirmation	0x40	--
	COMServiceId_TriggerTransmit	0x41	--
	COMServiceId_RxIndication	0x42	--
	COMServiceId_CopyTxData	0x43	--
	COMServiceId_CopyRxData	0x44	--
	COMServiceId_TpRxIndication	0x45	--
	COMServiceId_StartOfReception	0x46	--
	COMServiceId_TpTxConfirmation	0x48	--
<b>Description:</b>	Unique identifier of an AUTOSAR COM module's service. Example: COMServiceId_SendSignal 0x0A.		

] (SRS\_BSW\_00462, SRS\_BSW\_00441)

### 8.2.7 Com\_ConfigType

[SWS\_Com\_00825] [

<b>Name:</b>	Com_ConfigType	
<b>Type:</b>	Structure	
<b>Range:</b>	implementation specific	The content of the initialization data structure is implementation specific
<b>Description:</b>	This is the type of the data structure containing the initialization data for COM.	

] (SRS\_BSW\_00404, SRS\_BSW\_00441)

## 8.3 Function Definitions

[SWS\_Com\_00321] [Non-reentrant functions do not have to check if they are called reentrant.] (SRS\_BSW\_00312)

Most of the COM functions are non-reentrant with respect to a concrete object. E.g. Com\_SendSignal is not reentrant for the same signal, but reentrant for different signals. Hence, if there are multiple callers of the AUTOSAR COM module they need to be synchronized on a system level, for example by using disjoint sets of signals and I-PDU groups.

### 8.3.1 Return Codes

The AUTOSAR COM module does not define a special COM return type. The API services return errors either by using the Std\_ReturnType as defined in [5] or via a uint8 value mapped according to Table 10.

Return codes of the AUTOSAR COM module are defined in Table 10.

[SWS\_Com\_00865]

Name	Description	Type	Value	Defined in
E_OK	the service has been accepted	#define	0x00	Std_Types.h
E_NOT_OK	invocation of service failed	#define	0x01	Std_Types.h
COM_SERVICE_NOT_AVAILABLE	the service is currently not available e.g. the corresponding I-PDU group is stopped (or a development error has been detected)	#define	0x80	Com.h
COM_BUSY	Successful execution of this function is currently not possible, because a resource is currently in use. For example, the buffer of a large I-PDU is locked.  COM_BUSY might, but must not, be returned whenever a signal of a large I-PDU is accessed.	#define	0x81	Com.h

] (SRS\_BSW\_00348, SRS\_BSW\_00377, SRS\_BSW\_00441)

Table 10: Mapping of AUTOSAR COM module's return codes

### 8.3.2 Start-Up and Control Services

#### 8.3.2.1 Com\_Init

[SWS\_Com\_00432]

<b>Service name:</b>	Com_Init
<b>Syntax:</b>	void Com_Init( const Com_ConfigType* config )
<b>Service ID[hex]:</b>	0x01
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Non Reentrant
<b>Parameters (in):</b>	config   Pointer to the AUTOSAR COM module's configuration data.
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	This service initializes internal and external interfaces and variables of the AUTOSAR COM module layer for the further processing. After calling this function the inter-ECU communication is still disabled.

] (SRS\_BSW\_00344, SRS\_BSW\_00404, SRS\_BSW\_00405, SRS\_BSW\_00101, SRS\_BSW\_00358, SRS\_BSW\_00414)

The behavior of the AUTOSAR COM module is unspecified until a correct call to Com\_Init is made.

**Caveats of Com\_Init:** It is not foreseen that Com\_Init pre-empts any other AUTOSAR COM module's function. The AUTOSAR COM module can rely on the rest of the system that Com\_Init is not called in such a way.

### 8.3.2.2 Com\_Delnit

#### [SWS\_Com\_00130] [

<b>Service name:</b>	Com_Delnit
<b>Syntax:</b>	void Com_Delnit( void )
<b>Service ID[hex]:</b>	0x02
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Non Reentrant
<b>Parameters (in):</b>	None
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	This service stops the inter-ECU communication. All started I-PDU groups are stopped and have to be started again, if needed, after Com_Init is called. By a call to Com_Delnit the AUTOSAR COM module is put into an not initialized state.

] (SRS\_BSW\_00336)

[SWS\_Com\_00129] [The AUTOSAR COM module's function Com\_Delnit shall stop all started I-PDU groups.] (SRS\_BSW\_00336)

**Caveats of Com\_Delnit:** It is not foreseen that Com\_Delnit pre-empts any AUTOSAR COM module's function. The AUTOSAR COM module can relies on the rest of the system that Com\_Delnit is not called in such a way.

### 8.3.2.3 Com\_IpduGroupControl

#### [SWS\_Com\_00751] [

<b>Service name:</b>	Com_IpduGroupControl
<b>Syntax:</b>	void Com_IpduGroupControl( Com_IpduGroupVector ipduGroupVector, boolean initialize )
<b>Service ID[hex]:</b>	0x03
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Non Reentrant
<b>Parameters (in):</b>	ipduGroupVector   I-PDU group vector containing the activation state (stopped = 0/ started = 1) for all I-PDU groups. initialize   flag to request initialization of the I-PDUs which are newly started
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	This service starts I-PDU groups.

] (SRS\_Com\_00218)

[SWS\_Com\_00792] [At invocation of the function Com\_IpduGroupControl, the AUTOSAR COM module shall start/stop every ComIPdu according to the passed



states of the ComIpduGroups in the parameter ipduGroupVector.] (SRS\_Com\_00218)

The correct ipduGroupVector can be constructed with the helper functions Com\_ClearIpduGroupVector and Com\_SetIpduGroup. This allows using the symbolic names of the ComIPduGroups.

### 8.3.2.4 Com\_ReceptionDMControl

[SWS\_Com\_00752] [

<b>Service name:</b>	Com_ReceptionDMControl
<b>Syntax:</b>	void Com_ReceptionDMControl( Com_IpduGroupVector ipduGroupVector )
<b>Service ID[hex]:</b>	0x06
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Non Reentrant
<b>Parameters (in):</b>	ipduGroupVector   I-PDU group vector containing the requested deadline monitoring state (disabled = 0/ enabled = 1) for all I-PDU groups.
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	This service enables or disables I-PDU group Deadline Monitoring.

] (SRS\_Com\_00192)

[SWS\_Com\_00616] [First, the function Com\_ReceptionDMControl shall set the reception deadline monitoring state of all I-PDU groups to the requested state.] (SRS\_Com\_00192)

[SWS\_Com\_00617] [Second, the function Com\_ReceptionDMControl shall start or stop the reception deadline monitoring for all I-PDUs that change their reception deadline monitoring state because of this call of Com\_ReceptionDMControl respectively.] (SRS\_Com\_00192)

[SWS\_Com\_00618] [For all I-PDUs that do not change their deadline monitoring state the function, Com\_ReceptionDMControl shall do nothing.] (SRS\_Com\_00192)

### 8.3.2.5 Com\_GetStatus

[SWS\_Com\_00194] [

<b>Service name:</b>	Com_GetStatus
<b>Syntax:</b>	Com_StatusType Com_GetStatus( void )
<b>Service ID[hex]:</b>	0x07
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Reentrant
<b>Parameters (in):</b>	None
<b>Parameters (in-out):</b>	None

<b>Parameters (out):</b>	None	
<b>Return value:</b>	Com_StatusType	COM_UNINIT: the AUTOSAR COM module is not initialized and not usable COM_INIT: the AUTOSAR COM module is initialized and usable
<b>Description:</b>	Returns the status of the AUTOSAR COM module.	

] (SRS\_BSW\_00331)

### 8.3.2.6 Com\_GetVersionInfo

[SWS\_Com\_00426] [

<b>Service name:</b>	Com_GetVersionInfo	
<b>Syntax:</b>	void Com_GetVersionInfo( Std_VersionInfoType* versioninfo )	
<b>Service ID[hex]:</b>	0x09	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	versioninfo	Pointer to where to store the version information of this module.
<b>Return value:</b>	None	
<b>Description:</b>	Returns the version information of this module.	

] (SRS\_BSW\_00407, SRS\_BSW\_00003)

For configuration of Com\_GetVersionInfo see SWS\_Com\_00026.

### 8.3.2.7 Com\_ClearIpduGroupVector

[SWS\_Com\_00749] [

<b>Service name:</b>	Com_ClearIpduGroupVector	
<b>Syntax:</b>	void Com_ClearIpduGroupVector( Com_IpduGroupVector ipduGroupVector )	
<b>Service ID[hex]:</b>	0x1c	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (in-out):</b>	ipduGroupVector	I-PDU group vector to be cleared
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	This service sets all bits of the given Com_IpduGroupVector to 0.	

] (SRS\_Com\_02090)

### 8.3.2.8 Com\_SetIpduGroup

[SWS\_Com\_00750] [

<b>Service name:</b>	Com_SetIpduGroup	
<b>Syntax:</b>	void Com_SetIpduGroup( 	

	Com_IpduGroupVector ipduGroupVector, Com_IpduGroupIdType ipduGroupId, boolean bitval )	
<b>Service ID[hex]:</b>	0x1d	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	ipduGroupId	ipduGroup used to identify the corresponding bit in the I-PDU group vector
	bitval	New value of the corresponding bit
<b>Parameters (in-out):</b>	ipduGroupVector	I-PDU group vector to be modified
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	This service sets the value of a bit in an I-PDU group vector.	

] (SRS\_Com\_02090)

**[SWS\_Com\_00623]** [The function Com\_SetIpduGroup shall set the bit of the given I-PDU group vector that corresponds to the given I-PDU group, that is the n-th bit for the I-PDU group with ID n, to bitval.] (SRS\_Com\_02090)

### 8.3.3 Communication Services

#### 8.3.3.1 Com\_SendSignal

**[SWS\_Com\_00197]** [

<b>Service name:</b>	Com_SendSignal	
<b>Syntax:</b>	uint8 Com_SendSignal( Com_SignalIdType SignalId, const void* SignalDataPtr )	
<b>Service ID[hex]:</b>	0x0a	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant for the same signal. Reentrant for different signals.	
<b>Parameters (in):</b>	SignalId	Id of signal to be sent.
	SignalDataPtr	Reference to the signal data to be transmitted.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) COM_BUSY: in case the TP-Buffer is locked for large data types handling
<b>Description:</b>	The service Com_SendSignal updates the signal object identified by SignalId with the signal referenced by the SignalDataPtr parameter.	

] (SRS\_Com\_02037)

**[SWS\_Com\_00624]** [The service Com\_SendSignal shall update the signal object identified by SignalId with the signal referenced by the SignalDataPtr parameter.] (SRS\_Com\_02037)

**[SWS\_Com\_00625]** [If the updated signal has the ComTransferProperty TRIGGERED and it is assigned to an I-PDU with ComTxModeMode DIRECT or MIXED, then Com\_SendSignal shall perform an immediate transmission (within the next main function at the latest) of that I-PDU, unless the sending is delayed or prevented by other COM mechanisms.] (SRS\_Com\_02037)

Related mechanisms are for example the minimum delay timer (ComMinimumDelayTime) or the transmission offset (ComTxModeTimeOffset).

### 8.3.3.2 Com\_SendDynSignal

**[SWS\_Com\_00627]** [

<b>Service name:</b>	Com_SendDynSignal	
<b>Syntax:</b>	uint8 Com_SendDynSignal( Com_SignalIdType SignalId, const void* SignalDataPtr, uint16 Length )	
<b>Service ID[hex]:</b>	0x21	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant for the same signal. Reentrant for different signals.	
<b>Parameters (in):</b>	SignalId	Id of signal to be sent.
	SignalDataPtr	Reference to the signal data to be transmitted.
	Length	Length of the dynamic length signal
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	uint8	E_OK: service has been accepted E_NOT_OK: in case the Length is greater than the configured ComSignalLength of this sent signal COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) COM_BUSY: in case the TP-Buffer is locked
	<b>Description:</b>	The service Com_SendDynSignal updates the signal object identified by SignalId with the signal referenced by the SignalDataPtr parameter.

] (SRS\_Com\_02095)

**[SWS\_Com\_00628]** [The service Com\_SendDynSignal shall update the signal object identified by SignalId with the signal referenced by the SignalDataPtr parameter.] (SRS\_Com\_02095)

**[SWS\_Com\_00629]** [If the updated signal (see SWS\_Com\_00628) has the ComTransferProperty TRIGGERED, then the function Com\_SendDynSignal shall perform an immediate transmission (within the next main function at the latest) of the I-PDU associated with the signal except when the signal is packed into an I-PDU with Periodic transmission mode.] (SRS\_Com\_02095)

**[SWS\_Com\_00630]** [If the updated signal, see SWS\_Com\_00628, has the ComTransferProperty PENDING, then the function Com\_SendDynSignal shall not perform a transmission of the I-PDU associated with the signal.] (SRS\_Com\_02095)

There is no restriction regarding the minimal length of a dynamic length signal. Hence, it is allowed that a dynamic length signal has zero length.

### 8.3.3.3 Com\_ReceiveSignal

[SWS\_Com\_00198] [

<b>Service name:</b>	Com_ReceiveSignal	
<b>Syntax:</b>	uint8 Com_ReceiveSignal( Com_SignalIdType SignalId, void* SignalDataPtr )	
<b>Service ID[hex]:</b>	0x0b	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant for the same signal. Reentrant for different signals.	
<b>Parameters (in):</b>	SignalId	Id of signal to be received.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	SignalDataPtr	Reference to the location where the received signal data shall be stored
<b>Return value:</b>	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) COM_BUSY: in case the TP-Buffer is locked for large data types handling
<b>Description:</b>	Com_ReceiveSignal copies the data of the signal identified by SignalId to the location specified by SignalDataPtr.	

] (SRS\_Com\_02037)

[SWS\_Com\_00631] [The service Com\_ReceiveSignal shall copy the data of the signal object identified by SignalId to the position referenced by SignalDataPtr.] (SRS\_Com\_02037)

**Caveats of Com\_ReceiveSignal:** The AUTOSAR COM module relies on the fact the SignalDataPtr is properly aligned with respect to the data type of the signal with the given SignalId. The caller of Com\_ReceiveSignal has to ensure this.

### 8.3.3.4 Com\_ReceiveDynSignal

[SWS\_Com\_00690] [

<b>Service name:</b>	Com_ReceiveDynSignal	
<b>Syntax:</b>	uint8 Com_ReceiveDynSignal( Com_SignalIdType SignalId, void* SignalDataPtr, uint16* Length )	
<b>Service ID[hex]:</b>	0x22	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant for the same signal. Reentrant for different signals.	
<b>Parameters (in):</b>	SignalId	Id of signal to be received.
<b>Parameters (in-out):</b>	Length	in: maximum length that could be received out: length of the dynamic length signal
<b>Parameters (out):</b>	SignalDataPtr	reference to the location where the received signal data shall be

		stored
<b>Return value:</b>	uint8	E_OK: service has been accepted E_NOT_OK: in case the Length (as in-parameter) is smaller than the received length of the dynamic length signal COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) COM_BUSY: in case the TP-Buffer is locked
<b>Description:</b>	Com_ReceiveDynSignal copies the data of the signal identified by SignalId to the location specified by SignalDataPtr and stores the length of the dynamical length signal at the position given by the Length parameter.	

] (SRS\_Com\_02092, SRS\_Com\_02095)

**[SWS\_Com\_00711]** [The service Com\_ReceiveDynSignal shall copy the data of the signal object identified by SignalId to the position referenced by SignalDataPtr.] (SRS\_Com\_02092)

**[SWS\_Com\_00712]** [The service Com\_ReceiveDynSignal shall return the calculated length (see SWS\_Com\_00758) of the dynamic length signal identified by SignalId in parameter Length.] (SRS\_Com\_02092)

**[SWS\_Com\_00724]** [In case, the Length (as in-parameter) is smaller than the received length of the dynamic length signal, Com\_ReceiveDynSignal shall not copy any data to the position referenced by SignalDataPtr.] (SRS\_Com\_02092)

### 8.3.3.5 Com\_UpdateShadowSignal

**DEPRECATED:** This function will be removed in the next major release.

**[SWS\_Com\_00199]** [

<b>Service name:</b>	Com_UpdateShadowSignal	
<b>Syntax:</b>	void Com_UpdateShadowSignal( Com_SignalIdType SignalId, const void* SignalDataPtr )	
<b>Service ID[hex]:</b>	0x0c	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant for the same group signal. Reentrant for different group signals.	
<b>Parameters (in):</b>	SignalId	Id of group signal to be updated.
	SignalDataPtr	Reference to the group signal data to be updated.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	The service Com_UpdateShadowSignal updates a group signal with the data, referenced by SignalDataPtr.	

] (SRS\_Com\_02041)

**[SWS\_Com\_00632]** [The service Com\_UpdateShadowSignal shall update a group signal with the data, referenced by SignalDataPtr. The update of the group signal data shall be done in the shadow buffer, not in the I-PDU.] (SRS\_Com\_02041)

To send out the shadow buffer, Com\_SendSignalGroup has to be called.

**[SWS\_Com\_00634]** [Configuration of Com\_UpdateShadowSignal: A signal group must be configured before this call. See ECUC\_Com\_00345 for details.] (SRS\_Com\_02041)

### 8.3.3.6 Com\_SendSignalGroup

**[SWS\_Com\_00200]** [

<b>Service name:</b>	Com_SendSignalGroup	
<b>Syntax:</b>	uint8 Com_SendSignalGroup( Com_SignalGroupIdType SignalGroupId )	
<b>Service ID[hex]:</b>	0x0d	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant for the same group signal. Reentrant for different group signals.	
<b>Parameters (in):</b>	SignalGroupId   Id of signal group to be sent.	
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) COM_BUSY: in case the TP-Buffer is locked for large data types handling
<b>Description:</b>	The service Com_SendSignalGroup copies the content of the associated shadow buffer to the associated I-PDU.	

] (SRS\_Com\_02041)

**[SWS\_Com\_00635]** [The service Com\_SendSignalGroup shall copy the content of the shadow buffer referenced by parameter SignalGroupId to the associated I-PDU.] (SRS\_Com\_02041)

**[SWS\_Com\_00637]** [Configuration of Com\_SendSignalGroup: A signal group must be configured before this call. See ECUC\_Com\_00345 for details.] (SRS\_Com\_02041)

### 8.3.3.7 Com\_ReceiveSignalGroup

**[SWS\_Com\_00201]** [

<b>Service name:</b>	Com_ReceiveSignalGroup	
<b>Syntax:</b>	uint8 Com_ReceiveSignalGroup( Com_SignalGroupIdType SignalGroupId )	
<b>Service ID[hex]:</b>	0x0e	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant for the same group signal. Reentrant for different group signals.	
<b>Parameters (in):</b>	SignalGroupId   Id of signal group to be received.	
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group

		was stopped (or service failed due to development error) COM_BUSY: in case the TP-Buffer is locked for large data types handling
<b>Description:</b>	The service Com_ReceiveSignalGroup copies the received signal group from the I-PDU to the shadow buffer.	

] (SRS\_Com\_02041)

**[SWS\_Com\_00638]** [The service Com\_ReceiveSignalGroup shall copy the received signal group from the I-PDU to the shadow buffer.] (SRS\_Com\_02041)

After this call, the group signals could be copied from the shadow buffer to the RTE by calling Com\_ReceiveSignal.

**[SWS\_Com\_00639]** [Configuration of Com\_ReceiveSignalGroup: A signal group must be configured before this call. See ECUC\_Com\_00345 for details.] (SRS\_Com\_02041)

**[SWS\_Com\_00461]** [The AUTOSAR COM module shall always copy the last known data, or the ComSignalInitValue(s) if not yet written, of the I-PDU to the shadow buffer by a call to Com\_ReceiveSignalGroup even if the I-PDU is stopped and COM\_SERVICE\_NOT\_AVAILABLE is returned.] (SRS\_Com\_02041)

### 8.3.3.8 Com\_ReceiveShadowSignal

**DEPRECATED:** This function will be removed in the next major release.

**[SWS\_Com\_00202]** [

<b>Service name:</b>	Com_ReceiveShadowSignal	
<b>Syntax:</b>	void Com_ReceiveShadowSignal( Com_SignalIdType SignalId, void* SignalDataPtr )	
<b>Service ID[hex]:</b>	0x0f	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant for the same group signal. Reentrant for different group signals.	
<b>Parameters (in):</b>	SignalId	Id of group signal to be received.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	SignalDataPtr	Reference to the group signal data in which to store the received data.
<b>Return value:</b>	None	
<b>Description:</b>	The service Com_ReceiveShadowSignal updates the group signal which is referenced by SignalDataPtr with the data in the shadow buffer.	

] (SRS\_Com\_02041)

**[SWS\_Com\_00640]** [The service Com\_ReceiveShadowSignal shall update the group signal that is referenced by SignalDataPtr with the data in the shadow buffer.] (SRS\_Com\_02041)



### 8.3.3.9 Com\_SendSignalGroupArray

#### [SWS\_Com\_00851]

<b>Service name:</b>	Com_SendSignalGroupArray	
<b>Syntax:</b>	uint8 Com_SendSignalGroupArray( Com_SignalGroupIdType SignalGroupId, const uint8* SignalGroupArrayPtr )	
<b>Service ID[hex]:</b>	0x23	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant for the same signal group. Reentrant for different signal groups.	
<b>Parameters (in):</b>	SignalGroupId	Id of signal group to be sent.
	SignalGroupArrayPtr	Reference to the signal group array.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) COM_BUSY: in case the TP-Buffer is locked for large data types handling
<b>Description:</b>	The service Com_SendSignalGroupArray copies the content of the provided SignalGroupArrayPtr to the associated I-PDU. The provided data shall correspond to the array representation of the signal group.	

](SRS\_Com\_02112, SRS\_Com\_02041)

**[SWS\_Com\_00852]** [The service Com\_SendSignalGroupArray shall copy the content designated by the SignalGroupArrayPtr to the associated I-PDU.] (SRS\_Com\_02112, SRS\_Com\_02041)

**[SWS\_Com\_00853]** [The Com\_SendSignalGroupArray API shall only be available when the configuration switch ComEnableSignalGroupArrayApi is enabled.] (SRS\_Com\_00177)

### 8.3.3.10 Com\_ReceiveSignalGroupArray

#### [SWS\_Com\_00854]

<b>Service name:</b>	Com_ReceiveSignalGroupArray	
<b>Syntax:</b>	uint8 Com_ReceiveSignalGroupArray( Com_SignalGroupIdType SignalGroupId, uint8* SignalGroupArrayPtr )	
<b>Service ID[hex]:</b>	0x24	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant for the same signal group. Reentrant for different signal groups.	
<b>Parameters (in):</b>	SignalGroupId	Id of signal group to be received.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	SignalGroupArrayPtr	reference to the location where the received signal group array shall be stored
<b>Return value:</b>	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development

		error) COM_BUSY: in case the TP-Buffer is locked for large data types handling
<b>Description:</b>	The service Com_ReceiveSignalGroupArray copies the received signal group array representation from the PDU to the location designated by SignalGroupArrayPtr.	

](SRS\_Com\_02112, SRS\_Com\_02041)

**[SWS\_Com\_00855]** [The service Com\_ReceiveSignalGroupArray shall copy the array representation of the received signal group from the I-PDU to the location designated by SignalGroupArrayPtr.] (SRS\_Com\_02112, SRS\_Com\_02041)

**[SWS\_Com\_00856]** [The Com\_ReceiveSignalGroupArray API shall only be available when the configuration switch ComEnableSignalGroupArrayApi is enabled.] (SRS\_Com\_00177)

**[SWS\_Com\_00857]** [The AUTOSAR COM module shall always copy the last known data, or the ComSignalInitValues if not yet written, of the I-PDU to the SignalGroupArrayPtr by a call to Com\_ReceiveSignalGroupArray even if the I-PDU is stopped and COM\_SERVICE\_NOT\_AVAILABLE is returned.] (SRS\_Com\_02041)

### 8.3.3.11 Com\_InvalidateSignal

**[SWS\_Com\_00203]** [

<b>Service name:</b>	Com_InvalidateSignal	
<b>Syntax:</b>	uint8 Com_InvalidateSignal( Com_SignalIdType SignalId )	
<b>Service ID[hex]:</b>	0x10	
<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant for the same signal. Reentrant for different signals.	
<b>Parameters (in):</b>	SignalId Id of signal to be invalidated.	
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group is stopped, no ComSignalDataInvalidValue is configured for the given signalId or service fails due to development error COM_BUSY: in case the TP-Buffer is locked
<b>Description:</b>	The service Com_InvalidateSignal invalidates the signal with the given SignalId by setting its value to its configured ComSignalDataInvalidValue.	

](SRS\_Com\_02077)

See Chapter 7.4.4 for further information about the signal invalidation mechanism.

**[SWS\_Com\_00642]** [After invaliding the signal data, the function Com\_InvalidateSignal shall perform a call of Com\_SendSignal internally, for details see SWS\_Com\_00099.] (SRS\_Com\_02077)

**[SWS\_Com\_00643]** [Com\_InvalidateSignal shall return COM\_SERVICE\_NOT\_AVAILABLE in case no ComSignalDataInvalidValue (ECUC\_Com\_00391) is configured for the signal with the given SignalId.] (SRS\_Com\_02077)

### 8.3.3.12 Com\_InvalidateShadowSignal

**DEPRECATED:** This function will be removed in the next major release.

**[SWS\_Com\_00288]** [

<b>Service name:</b>	Com_InvalidateShadowSignal
<b>Syntax:</b>	void Com_InvalidateShadowSignal( Com_SignalIdType SignalId )
<b>Service ID[hex]:</b>	0x16
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Non Reentrant for the same signal. Reentrant for different signals.
<b>Parameters (in):</b>	SignalId   Id of signal to be sent.
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	The service Com_InvalidateShadowSignal invalidates the group signal with the given SignalId by setting its value to its configured ComSignalDataInvalidValue.

] (SRS\_Com\_02077)

See Chapter 7.3.4 for further information about the signal invalidation mechanism.

**[SWS\_Com\_00644]** [The AUTOSAR COM module shall only process Com\_InvalidateShadowSignal only if a ComSignalDataInvalidValue, see ECUC\_Com\_00391, is configured for the signal with the given SignalId.] (SRS\_Com\_02077)

In contrast to Com\_InvalidateSignal and Com\_InvalidateSignalGroup, the AUTOSAR COM module does not perform an send request by a call to Com\_InvalidateShadowSignal. Hence, if the RTE uses this function to invalidate a signal group, the RTE has to invoke Com\_SendSignalGroup after the invalidation of the group signals to send out the invalidated signal group.

The service Com\_InvalidateShadowSignal is included in the AUTOSAR COM module because of backwards compatibility reasons. In general, the function Com\_InvalidateSignalGroup should be used instead.

### 8.3.3.13 Com\_InvalidateSignalGroup

**[SWS\_Com\_00557]** [

<b>Service name:</b>	Com_InvalidateSignalGroup
<b>Syntax:</b>	uint8 Com_InvalidateSignalGroup( Com_SignalGroupIdType SignalGroupId )
<b>Service ID[hex]:</b>	0x1b

<b>Sync/Async:</b>	Asynchronous	
<b>Reentrancy:</b>	Non Reentrant for the same signal group. Reentrant for different signal groups.	
<b>Parameters (in):</b>	SignalGroupId	Id of signal group to be invalidated.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	uint8	E_OK: service has been accepted COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group is stopped, no ComSignalDataInvalidValue is configured for any of the group signals or service fails due to development error COM_BUSY: in case the TP-Buffer is locked
<b>Description:</b>	The service Com_InvalidateSignalGroup invalidates all group signals of the signal group with the given SignalGroupId by setting their values to their configured ComSignalDataInvalidValues.	

] (SRS\_Com\_02077)

For further information about the signal invalidation mechanism, see Chapter 7.4.4.

**[SWS\_Com\_00645]** [After invaliding the signal group data, the function Com\_InvalidateSignalGroup shall perform a call of Com\_SendSignalGroup internally.] (SRS\_Com\_02077)

### 8.3.3.14 Com\_TriggerIPDUSend

**[SWS\_Com\_00348]** [

<b>Service name:</b>	Com_TriggerIPDUSend	
<b>Syntax:</b>	Std_ReturnType Com_TriggerIPDUSend( PduldType Pduld )	
<b>Service ID[hex]:</b>	0x17	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	Pduld	The I-PDU-ID of the I-PDU that shall be triggered for sending
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: I-PDU was triggered for transmission E_NOT_OK: I-PDU is stopped, the transmission could not be triggered
<b>Description:</b>	By a call to Com_TriggerIPDUSend the I-PDU with the given ID is triggered for transmission.	

] (SRS\_Com\_02037)

**[SWS\_Com\_00861]** [By a call to Com\_TriggerIPDUSend, the AUTOSAR COM shall trigger the I-PDU with the given ID for transmission only if the I-PDU is started.] (SRS\_Com\_02037)

In case a stopped I-PDU is triggered for transmission, just E\_NOT\_OK is returned. For stopped I-PDUs no triggers are stored for a later execution after a potential (re-) starting of the I-PDU.

**[SWS\_Com\_00388]** [When an I-PDU is transmitted because of the function Com\_TriggerIPDUSend or Com\_TriggerIPDUSendWithMetaData, the AUTOSAR COM module shall only take the minimum delay time into account. That is the AUTOSAR COM module shall postpone transmissions if necessary and reset the minimum delay timer in case of transmissions. The AUTOSAR COM module shall not take into account all other transmission mode related parameters like ComTxMode-NumberOfRepetitions.] (SRS\_Com\_02037)

**[SWS\_Com\_00492]** [If an I-PDU triggered by Com\_TriggerIPDUSend or Com\_TriggerIPDUSendWithMetaData has a configured I-PDU-callout, see ECUC\_Com\_00387, then the function Com\_TriggerIPDUSend or Com\_TriggerIPDUSendWithMetaData respectively shall also call this I-PDU-Callout.] (SRS\_Com\_02037)

If within the Com\_TriggerIPDUSend function, another call to Com\_TriggerIPDUSend is invoked the application programmer must take care to not create infinite recursions. Both, direct recursions by invoking Com\_TriggerIPDUSend with the same ComPduId again and indirect recursions with longer recursion paths are theoretically possible.

The previous paragraph is also true for Com\_TriggerIPDUSendWithMetaData as well as for combinations of these two I-PDU triggering functions.

**[SWS\_Com\_00867]** [In case a large I-PDU is triggered for transmission by Com\_TriggerIPDUSend and the buffer of the large I-PDU is currently locked, the AUTOSAR COM shall store the transmit request and transmit the I-PDU when the buffer is available again.] (SRS\_Com\_02037)

### 8.3.3.15 Com\_TriggerIPDUSendWithMetaData

**[SWS\_Com\_00858]**

<b>Service name:</b>	Com_TriggerIPDUSendWithMetaData	
<b>Syntax:</b>	Std_ReturnType Com_TriggerIPDUSendWithMetaData( PduIdType PduId, uint8* MetaData )	
<b>Service ID[hex]:</b>	0x28	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	PduId	The I-PDU-ID of the I-PDU that shall be triggered for sending
	MetaData	A pointer to the metadata for the triggered send-request
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: I-PDU was triggered for transmission E_NOT_OK: I-PDU is stopped, the transmission could not be triggered
	<b>Description:</b> By a call to Com_TriggerIPDUSendWithMetaData the AUTOSAR COM module updates its internal metadata for the I-PDU with the given ID by copying the metadata from the given position and with respect to the globally configured metadata length of the I-PDU. Then the I-PDU is triggered for transmission.	

] (SRS\_Com\_02037)

**[SWS\_Com\_00862]** [By a call to Com\_TriggerIPDUSendWithMetaData, the AUTOSAR COM shall trigger the I-PDU with the given ID for transmission only if the I-PDU is started.] (SRS\_Com\_02037)

In case a stopped I-PDU is triggered for transmission, just E\_NOT\_OK is returned. There is no mechanism in COM to store the triggering and to execute it after a potential (re-)starting of the I-PDU.

**[SWS\_Com\_00868]** [In case a large I-PDU is triggered for transmission by Com\_TriggerIPDUSendWithMetaData and the buffer of the large I-PDU is currently locked, the AUTOSAR COM shall store the transmit request with the given MetaData and transmit the I-PDU when the buffer is available again.] (SRS\_Com\_02037)

### 8.3.3.16 Com\_SwitchIpdUTxMode

**[SWS\_Com\_00784]** [

<b>Service name:</b>	Com_SwitchIpdUTxMode	
<b>Syntax:</b>	void Com_SwitchIpdUTxMode( PduldType Pduld, boolean Mode )	
<b>Service ID[hex]:</b>	0x27	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different Pdulds. Non reentrant for the same Pduld.	
<b>Parameters (in):</b>	Pduld	Id of the I-PDU of which the transmission mode shall be changed.
	Mode	the transmission mode that shall be set.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	The service Com_SwitchIpdUTxMode sets the transmission mode of the I-PDU referenced by Pduld to Mode. In case the transmission mode changes, the new mode shall immediately be effective (see SWS_Com_00239). In case the requested transmission mode was already active for this I-PDU, the call will have no effect.	

] (SRS\_Com\_02082)

By mixing the signal based TMS and explicit TMS by Com\_SwitchIpdUTxMode for the same I-PDU, it must be noted that the signal base TMS might switch back the manual set mode during a call to Com\_SendSignal or Com\_SendSignalGroup for this I-PDU. Therefore, mixing explicit and manual TMS for one I-PDU must be carefully designed, if used at all.

## 8.4 Callback Functions and Notifications

### 8.4.1 Com\_TriggerTransmit

[SWS\_Com\_00001] [

<b>Service name:</b>	Com_TriggerTransmit	
<b>Syntax:</b>	Std_ReturnType Com_TriggerTransmit( PduldType TxPduld, PdulInfoType* PdulInfoPtr )	
<b>Service ID[hex]:</b>	0x41	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different Pdulds. Non reentrant for the same Pduld.	
<b>Parameters (in):</b>	TxPduld	ID of the SDU that is requested to be transmitted.
<b>Parameters (in-out):</b>	PdulInfoPtr	Contains a pointer to a buffer (SduDataPtr) to where the SDU data shall be copied, and the available buffer size in SduLength. On return, the service will indicate the length of the copied SDU data in SduLength.
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Std_ReturnType	E_OK: SDU has been copied and SduLength indicates the number of copied bytes. E_NOT_OK: No SDU data has been copied. PdulInfoPtr must not be used since it may contain a NULL pointer or point to invalid data.
<b>Description:</b>	Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by PdulInfoPtr->SduLength. If it fits, it shall copy its data into the buffer provided by PdulInfoPtr->SduDataPtr and update the length of the actual copied data in PdulInfoPtr->SduLength. If not, it returns E_NOT_OK without changing PdulInfoPtr.	

] (SRS\_Com\_02045)

[SWS\_Com\_00647] [Within the function Com\_TriggerTransmit, the AUTOSAR COM module shall copy the contents of its I-PDU transmit buffer to the L-PDU buffer given by PdulInfoPtr->SduDataPtr and update PdulInfoPtr->SduLength with length of the copied data accordingly.] (SRS\_Com\_02045)

[SWS\_Com\_00869] [In case the given PdulInfoPtr->SduLength is smaller than the actual PDU-length, Com\_TriggerTransmit shall not copy any data and return E\_NOT\_OK.] (SRS\_Com\_02045)

[SWS\_Com\_00475] [Com\_TriggerTransmit is not interfered by the I-PDU minimum delay time and shall not reset the minimum delay timer, see ECUC\_Com\_00181.] (SRS\_Com\_02045)

The function Com\_TriggerTransmit behaves independent of the configured transmission mode.

**Use case:** This function is used e.g. by the LIN Master for sending out a LIN frame. In this case, the trigger transmit can be initiated by the Master schedule table itself or a received LIN header. The FlexRay Interface for requesting PDUs to be sent in static part (synchronous to the FlexRay global time) also uses this function.

Once the PDU Router has successfully sent the I-PDU, the PDU Router will call Com\_TxConfirmation.

**Caveats of Com\_TriggerTransmit:** This function might be called in interrupt context.

### 8.4.2 Com\_RxIndication

[SWS\_Com\_00123] [

<b>Service name:</b>	Com_RxIndication	
<b>Syntax:</b>	void Com_RxIndication( PduldType RxPduld, const PdulInfoType* PdulInfoPtr )	
<b>Service ID[hex]:</b>	0x42	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant for different Pdulds. Non reentrant for the same Pduld.	
<b>Parameters (in):</b>	RxPduld	ID of the received I-PDU.
	PdulInfoPtr	Contains the length (SduLength) of the received I-PDU and a pointer to a buffer (SduDataPtr) containing the I-PDU.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Indication of a received I-PDU from a lower layer communication interface module.	

] (SRS\_Com\_02043)

**Caveats of Com\_RxIndication:** This function might be called in interrupt context.

### 8.4.3 Com\_TpRxIndication

[SWS\_Com\_00650] [

<b>Service name:</b>	Com_TpRxIndication	
<b>Syntax:</b>	void Com_TpRxIndication( PduldType id, Std_ReturnType result )	
<b>Service ID[hex]:</b>	0x45	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	id	Identification of the received I-PDU.
	result	Result of the reception.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Called after an I-PDU has been received via the TP API, the result indicates whether the transmission was successful or not.	

] (SRS\_Com\_02095)

**Caveats of Com\_TpRxIndication:** This function might be called in interrupt context.



#### 8.4.4 Com\_TxConfirmation

##### [SWS\_Com\_00124] [

<b>Service name:</b>	Com_TxConfirmation
<b>Syntax:</b>	void Com_TxConfirmation( PduIdType TxPduId )
<b>Service ID[hex]:</b>	0x40
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Reentrant for different PduIds. Non reentrant for the same PduId.
<b>Parameters (in):</b>	TxPduId ID of the I-PDU that has been transmitted.
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	The lower layer communication interface module confirms the transmission of an I-PDU.

] (SRS\_Com\_02044)

**Caveats of Com\_TxConfirmation:** This function might be called in interrupt context, e.g. from transmit interrupt.

#### 8.4.5 Com\_TpTxConfirmation

##### [SWS\_Com\_00725] [

<b>Service name:</b>	Com_TpTxConfirmation
<b>Syntax:</b>	void Com_TpTxConfirmation( PduIdType id, Std_ReturnType result )
<b>Service ID[hex]:</b>	0x48
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Reentrant
<b>Parameters (in):</b>	id Identification of the transmitted I-PDU. result Result of the transmission of the I-PDU.
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	This function is called after the I-PDU has been transmitted on its network, the result indicates whether the transmission was successful or not.

] (SRS\_Com\_02095)

#### 8.4.6 Com\_StartOfReception

##### [SWS\_Com\_00691] [

<b>Service name:</b>	Com_StartOfReception
<b>Syntax:</b>	BufReq_ReturnType Com_StartOfReception( PduIdType id, const PduInfoType* info, PduLengthType TpSduLength, PduLengthType* bufferSizePtr

	)	
<b>Service ID[hex]:</b>	0x46	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	id	Identification of the I-PDU.
	info	Pointer to a PduInfoType structure containing the payload data (without protocol information) and payload length of the first frame or single frame of a transport protocol I-PDU reception. Depending on the global parameter MetaDataLength, additional bytes containing MetaData (e.g. the CAN ID) are appended after the payload data, increasing the length accordingly. If neither first/single frame data nor MetaData are available, this parameter is set to NULL_PTR.
	TpSduLength	Total length of the N-SDU to be received.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	bufferSizePtr	Available receive buffer in the receiving module. This parameter will be used to compute the Block Size (BS) in the transport protocol module.
<b>Return value:</b>	BufReq_ReturnType	BUFREQ_OK: Connection has been accepted. bufferSizePtr indicates the available receive buffer; reception is continued. If no buffer of the requested size is available, a receive buffer size of 0 shall be indicated by bufferSizePtr. BUFREQ_E_NOT_OK: Connection has been rejected; reception is aborted. bufferSizePtr remains unchanged. BUFREQ_E_OVFL: No buffer of the required length can be provided; reception is aborted. bufferSizePtr remains unchanged.
<b>Description:</b>	This function is called at the start of receiving an N-SDU. The N-SDU might be fragmented into multiple N-PDUs (FF with one or more following CFs) or might consist of a single N-PDU (SF).	

] (SRS\_Com\_02095)

**[SWS\_Com\_00654]** [If ComRxPduId corresponds to a large I-PDU and TpSduLength is greater than the configured length of this I-PDU then the function Com\_StartOfReception shall return BUFREQ\_E\_OVFL.] (SRS\_Com\_02095)

**[SWS\_Com\_00655]** [If ComRxPduId corresponds to a dynamic length I-PDU and TpSduLength is greater than the configured maximum length of this I-PDU then the function Com\_StartOfReception shall return BUFREQ\_E\_OVFL.] (SRS\_Com\_02095)

**[SWS\_Com\_00656]** [If ComRxPduId corresponds to a dynamic length I-PDU, the function Com\_StartOfReception shall calculate the length of the included dynamic length signal with respect to TpSduLength.] (SRS\_Com\_02095)

**[SWS\_Com\_00657]** [If the internal AUTOSAR COM module's buffer is currently not locked and the parameter TpSduLength does not exceed the (maximum) expected length, the function Com\_StartOfReception shall write the complete size of the AUTOSAR COM module's internal I-PDU buffer of the I-PDU with handle ComRxPduId to the position given by RxBufferSizePtr.] (SRS\_Com\_02095)

**[SWS\_Com\_00721]** [Com\_StartOfReception shall return BUFREQ\_E\_NOT\_OK, in case ComRxPduId refers to a stopped I-PDU.] (SRS\_Com\_02095)

[SWS\_Com\_00818] [In case Com\_StartOfReception returns BUFREQ\_E\_NOT\_OK or BUFREQ\_E\_OVFL, the AUTOSAR COM module shall not expect a call to Com\_TpRxIndication and hence shall not lock its internal buffer.] (SRS\_Com\_02095)

### 8.4.7 Com\_CopyRxData

[SWS\_Com\_00692] [

<b>Service name:</b>	Com_CopyRxData	
<b>Syntax:</b>	BufReq_ReturnType Com_CopyRxData( PduIdType id, const PduInfoType* info, PduLengthType* bufferSizePtr )	
<b>Service ID[hex]:</b>	0x44	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	id	Identification of the received I-PDU.
	info	Provides the source buffer (SduDataPtr) and the number of bytes to be copied (SduLength). An SduLength of 0 can be used to query the current amount of available buffer in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	bufferSizePtr	Available receive buffer after data has been copied.
<b>Return value:</b>	BufReq_ReturnType	BUFREQ_OK: Data copied successfully BUFREQ_E_NOT_OK: Data was not copied because an error occurred.
<b>Description:</b>	This function is called to provide the received data of an I-PDU segment (N-PDU) to the upper layer. Each call to this function provides the next part of the I-PDU data. The size of the remaining data is written to the position indicated by bufferSizePtr.	

] (SRS\_Com\_02095)

**Caveats of Com\_CopyRxData:** This function might be called in interrupt context. Therefore, data consistency must be ensured.

[SWS\_Com\_00782] [Com\_CopyRxData shall copy no data and return BUFREQ\_E\_NOT\_OK in case the given PduId refers to a stopped I-PDU.] (SRS\_Com\_00218)

### 8.4.8 Com\_CopyTxData

[SWS\_Com\_00693] [

<b>Service name:</b>	Com_CopyTxData	
<b>Syntax:</b>	BufReq_ReturnType Com_CopyTxData( PduIdType id, const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr )	
<b>Service ID[hex]:</b>	0x43	

<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	id	Identification of the transmitted I-PDU.
	info	Provides the destination buffer (SduDataPtr) and the number of bytes to be copied (SduLength). If not enough transmit data is available, no data is copied by the upper layer module and BUFREQ_E_BUSY is returned. The lower layer module may retry the call. An SduLength of 0 can be used to indicate state changes in the retry parameter or to query the current amount of available data in the upper layer module. In this case, the SduDataPtr may be a NULL_PTR.
	retry	This parameter is used to acknowledge transmitted data or to retransmit data after transmission problems.  If the retry parameter is a NULL_PTR, it indicates that the transmit data can be removed from the buffer immediately after it has been copied. Otherwise, the retry parameter must point to a valid RetryInfoType element.  If TpDataState indicates TP_CONFENDING, the previously copied data must remain in the TP buffer to be available for error recovery. TP_DATACONF indicates that all data that has been copied before this call is confirmed and can be removed from the TP buffer. Data copied by this API call is excluded and will be confirmed later. TP_DATARETRY indicates that this API call shall copy previously copied data in order to recover from an error. In this case TxTpDataCnt specifies the offset in bytes from the current data copy position.
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	availableDataPtr	Indicates the remaining number of bytes that are available in the upper layer module's Tx buffer. availableDataPtr can be used by TP modules that support dynamic payload lengths (e.g. FrIsoTp) to determine the size of the following CFs.
<b>Return value:</b>	BufReq_ReturnType	BUFREQ_OK: Data has been copied to the transmit buffer completely as requested. BUFREQ_E_BUSY: Request could not be fulfilled, because the required amount of Tx data is not available. The lower layer module may retry this call later on. No data has been copied. BUFREQ_E_NOT_OK: Data has not been copied. Request failed.
<b>Description:</b>	This function is called to acquire the transmit data of an I-PDU segment (N-PDU). Each call to this function provides the next part of the I-PDU data unless retry->TpDataState is TP_DATARETRY. In this case the function restarts to copy the data beginning at the offset from the current position indicated by retry->TxTpDataCnt. The size of the remaining data is written to the position indicated by availableDataPtr.	

] (SRS\_Com\_02095)

**Caveats of Com\_CopyTxData:** This function might be called in interrupt context.

**[SWS\_Com\_00783]** [Com\_CopyTxData shall copy no data and return BUFREQ\_E\_NOT\_OK in case the given Pduld refers to a stopped I-PDU.] (SRS\_Com\_00218)

## 8.5 Scheduled Functions

### 8.5.1 Com\_MainFunctionRx

[SWS\_Com\_00398] [

<b>Service name:</b>	Com_MainFunctionRx
<b>Syntax:</b>	void Com_MainFunctionRx( void )
<b>Service ID[hex]:</b>	0x18
<b>Description:</b>	This function performs the processing of the AUTOSAR COM module's receive processing that are not directly handled within the COM's functions invoked by the PDU-R, for example Com_RxIndication.

] (SRS\_BSW\_00425, SRS\_BSW\_00432)

[SWS\_Com\_00664] [A call to Com\_MainFunctionRx shall simply return if the AUTOSAR COM module was not previously initialized with a call to Com\_Init.] (SRS\_BSW\_00425)

Configuration of Com\_MainFunctionRx: see ECUC\_Com\_00186.

### 8.5.2 Com\_MainFunctionTx

[SWS\_Com\_00399] [

<b>Service name:</b>	Com_MainFunctionTx
<b>Syntax:</b>	void Com_MainFunctionTx( void )
<b>Service ID[hex]:</b>	0x19
<b>Description:</b>	This function performs the processing of the AUTOSAR COM module's transmission activities that are not directly handled within the COM's function invoked by the RTE, for example Com_SendSignal.

] (SRS\_BSW\_00425, SRS\_BSW\_00432)

[SWS\_Com\_00665] [A call to Com\_MainFunctionTx shall simply return if the AUTOSAR COM module was not previously initialized with a call to Com\_Init.] (SRS\_BSW\_00425)

Configuration: see ECUC\_Com\_00186.

### 8.5.3 Com\_MainFunctionRouteSignals

[SWS\_Com\_00400] [

<b>Service name:</b>	Com_MainFunctionRouteSignals
<b>Syntax:</b>	void Com_MainFunctionRouteSignals( void )
<b>Service ID[hex]:</b>	0x1a
<b>Description:</b>	Calls the signal gateway part of the AUTOSAR COM module to forward received signals to be routed.

] (SRS\_BSW\_00425, SRS\_BSW\_00432)

**[SWS\_Com\_00666]** [A call to Com\_MainFunctionRouteSignals shall simply return if the AUTOSAR COM module was not previously initialized with a call to Com\_Init.] (SRS\_BSW\_00425)

**Caveats of Com\_MainFunctionRouteSignals:**

- The time between two consecutive calls (perhaps the related task/thread cycle) affects directly the signal gateway latency.
- The insertion of this call is necessary for decoupling receive interrupts and signal gateway tasks.
- Normally a cyclic task/ thread to call this function is configured. The cycle of this task/thread directly affects the latency of the signal gateway.

## 8.6 Expected Interfaces

### 8.6.1 Mandatory Interfaces

This chapter defines all interfaces, which are required to fulfill the core functionality of the module.

**[SWS\_Com\_00669]** [

<i>API function</i>	<i>Description</i>
PduR_ComTransmit	Requests transmission of an I-PDU.

] (SRS\_BSW\_00384)

### 8.6.2 Optional Interfaces

This chapter defines all interfaces that are required to fulfill an optional functionality of the module.

**[SWS\_Com\_00670]** [

<i>API function</i>	<i>Description</i>
Det_ReportError	Service to report development errors.
Det_ReportRuntimeError	Service to report runtime errors. If a callout has been configured then this callout shall be called.
PduR_ComCancelTransmit	Requests cancellation of an ongoing transmission of an I-PDU in a lower layer communication interface or transport protocol module.

] (SRS\_BSW\_00384)

The associated configuration parameter for API function Det\_ReportError is ComConfigurationUseDet (ECUC\_Com\_00141).

The API function PduR\_ComCancelTransmit must be included if at least one I-PDU has configured ComIPduCancellationSupport to TRUE, see ECUC\_Com\_00709.

## 8.6.3 Configurable Interfaces

### 8.6.3.1 Callback/ Notification Functions

**Caveats of configurable interfaces:** A callback routine runs either on interrupt level or on task level. Thus, the OS restrictions of usage of system functions for interrupt service routines as well as for tasks apply.

#### [SWS Com\_00468] [

<b>Service name:</b>	Com_CbkTxAck
<b>Syntax:</b>	void Com_CbkTxAck( void )
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	don't care
<b>Parameters (in):</b>	None
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	This callback represents notification class 2 of [17]. It is called immediately after successful transmission of the I-PDU containing the message.

] (SRS\_BSW\_00359, SRS\_BSW\_00360)

Com\_CbkTxAck is called on sender side only. It can be configured for signals and signal groups.

Com\_CbkTxAck corresponds to Rte\_COMCbktAck\_<sn> or Rte\_COMCbktAck\_<sg> respectively.

The callback function name must be configured by ComNotification (ECUC\_Com\_00498).

#### [SWS Com\_00491] [

<b>Service name:</b>	Com_CbkTxErr
<b>Syntax:</b>	void Com_CbkTxErr( void )
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	don't care
<b>Parameters (in):</b>	None
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	This callback corresponds to notification class 4 of [17]. It is called in case the transmission is not possible because the corresponding I-PDU group is stopped.

] (SRS\_BSW\_00359, SRS\_BSW\_00360)

Com\_CbkTxErr is called on sender side only. This callback function corresponds to Rte\_COMCbKTErr\_<sn> or Rte\_COMCbKTErr\_<sg> respectively.

The callback function name must be configured by ComErrorNotification (ECUC\_Com\_00499).

**[SWS\_Com\_00554] [**

<b>Service name:</b>	Com_CbkTxTOut
<b>Syntax:</b>	void Com_CbkTxTOut( void )
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	don't care
<b>Parameters (in):</b>	None
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	This callback corresponds to notification class 4 of [17]. It is called immediately after a message transmission error has been detected by the deadline monitoring mechanism.

] (SRS\_BSW\_00359, SRS\_BSW\_00360)

Com\_CbkTxTOut is called on sender side only. It can be configured for signals and signal groups.

This callback function corresponds to Rte\_COMCbKTxTOut\_<sn> or Rte\_COMCbKTxTOut\_<sg> respectively.

The callback function name must be configured by ComTimeoutNotification (ECUC\_Com\_00552).

**[SWS\_Com\_00555] [**

<b>Service name:</b>	Com_CbkRxAck
<b>Syntax:</b>	void Com_CbkRxAck( void )
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	don't care
<b>Parameters (in):</b>	None
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	This callback represents notification class 1 of [17]. It is called immediately after the message has been stored in the receiving message object.

] (SRS\_BSW\_00359, SRS\_BSW\_00360)

Com\_CbkRxAck is called on receiver side only. It can be configured for signals and signal groups.



Com\_CbkRxAck corresponds to Rte\_COMCbk\_<sn> or Rte\_COMCbk\_<sg> respectively.

The callback function name must be configured by ComNotification (ECUC\_Com\_00498).

**[SWS\_Com\_00556] [**

<b>Service name:</b>	Com_CbkRxTOut
<b>Syntax:</b>	void Com_CbkRxTOut( void )
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	don't care
<b>Parameters (in):</b>	None
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	This callback corresponds to notification class 3 of [17]. It is called immediately after a message reception error has been detected by the deadline monitoring mechanism.

**] (SRS\_BSW\_00359, SRS\_BSW\_00360)**

Com\_CbkRxTOut is called on receiver side only. It can be configured for signals and signal groups.

This callback function corresponds to Rte\_COMCbkRxTOut\_<sn> or Rte\_COMCbkRxTOut\_<sg> respectively.

The callback function name must be configured by ComTimeoutNotification (see ECUC\_Com\_00552).

**[SWS\_Com\_00536] [**

<b>Service name:</b>	Com_CbkInv
<b>Syntax:</b>	void Com_CbkInv( void )
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	don't care
<b>Parameters (in):</b>	None
<b>Parameters (in-out):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	This callback function corresponds to SWS_Com_00680. It is called after reception of an invalid signal or signal group respectively.

**] (SRS\_BSW\_00359, SRS\_BSW\_00360, SRS\_Com\_02079)**

Com\_CbkInv is called on receiver side only. It can be configured for signals, group signals and signal groups.

This callback function corresponds to Rte\_COMCbkInv\_<sn> (for signals and group signals) and Rte\_COMCbkInv\_<sg> respectively.

The callback function name must be configured by ComInvalidNotification (ECUC\_Com\_00315).

**[SWS\_Com\_00726] [**

<b>Service name:</b>	Com_CbkCounterErr	
<b>Syntax:</b>	void Com_CbkCounterErr( PduldType ComPduld, uint8 ExpectedCounter, uint8 ReceivedCounter )	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	don' t care	
<b>Parameters (in):</b>	ComPduld	Id of the I-PDU for that a counter mismatch was detected
	ExpectedCounter	The expected counter value
	ReceivedCounter	The received counter value
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	This function is called immediately, if the AUTOSAR COM module receives an I-PDU counter that differs from the expected one.	

] (SRS\_Com\_02102)

Com\_CbkCounterErr is called on receiver side only. It can be configured per I-PDU counter. The callback function name must be configured by ComIPduCounterErrorNotification (ECUC\_Com\_00003).

### 8.6.3.2 I-PDU Callout Functions

As stated in SWS\_Com\_00013 *Network-order message callout* and *CPU-order message callout* are not supported by the AUTOSAR COM module. The only callout method supported by the AUTOSAR COM module therefore is the I-PDU callout. The AUTOSAR COM module supports I-PDU callouts on sender and on receiver side.

As specified in OSEK COM, if the I-PDU callout returns false the I-PDU will not be processed any further.

**[SWS\_Com\_00700] [**

<b>Service name:</b>	Com_RxlpduCallout	
<b>Syntax:</b>	boolean Com_RxlpduCallout( PduldType Pduld, const PdulInfoType* PdulInfoPtr )	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	don't care	

<b>Parameters (in):</b>	Pduld	ID of the received I-PDU.
	PdulInfoPtr	Contains the length (SduLength) of the received I-PDU and a pointer to the data of the I-PDU (SduDataPtr).
<b>Parameters (in-out):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	boolean	true: I-PDU will be processed normal false: I-PDU will not be processed any further
	<b>Description:</b> The I-PDU callout on receiver side can be configured to implement user-defined receive filtering mechanisms.	

] (SRS\_Com\_02037)

The callout function name must be configured by ComIPduCallout (ECUC\_Com\_00387).

**[SWS\_Com\_00816]** [The AUTOSAR COM module shall forward all data of the received I-PDU (i.e. the complete I-PDU as provided by the PduR) in the Com\_RxIpdudCallout.] (SRS\_Com\_02037)

**[SWS\_Com\_00346]** [

<b>Service name:</b>	Com_TxIpdudCallout	
<b>Syntax:</b>	boolean Com_TxIpdudCallout( PduldType Pduld, PdulInfoType* PdulInfoPtr )	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	don't care	
<b>Parameters (in):</b>	Pduld	ID of the transmitted I-PDU.
	PdulInfoPtr	Contains the length (SduLength) of the transmitted I-PDU and a pointer to the data of the I-PDU (SduDataPtr).
<b>Parameters (out):</b>	None	
<b>Return value:</b>	boolean	true: I-PDU will be processed normal false: I-PDU will not be processed any further
	<b>Description:</b> The I-PDU callout on sender side can be configured for example to implement user-defined transmission filtering or user-defined pre-transmission-processing of the outgoing I-PDU.	

] (SRS\_Com\_02037)

The callout function name must be configured either by ComIPduCallout (ECUC\_Com\_00387) or by ComIPduTriggerTransmitCallout (ECUC\_Com\_00765).

The AUTOSAR COM module implements no means to detect or prevent possible recursions that may occur when I-PDUs are triggered by actions within an I-PDU callout. It is solely in the responsibility of the implementer of an I-PDU callout and the system integrator that unwanted or endless chains of I-PDU callouts are avoided.

**[SWS\_Com\_00381]** [The AUTOSAR COM module shall not support that other AUTOSAR COM module's APIs than Com\_TriggerIPDUSend, Com\_TriggerIPDUSend-WithMetaData, Com\_SendSignal and Com\_SendSignalGroup can be called out of an I-PDU callout.] (SRS\_Com\_02037)

**[SWS\_Com\_00766]** [In case a ComIPduTriggerTransmitCallout is configured for an I-PDU and the I-PDU is started, the AUTOSAR COM module shall invoke this I-PDU

callout within every execution of `Com_TriggerTransmit` for this I-PDU.] (SRS\_Com\_02037)

**[SWS\_Com\_00395]** [When `Com_TriggerTransmit` is called, the AUTOSAR COM module shall ignore the return value from the `ComIPduTriggerTransmitCallout`.] (SRS\_Com\_02037)

**[SWS\_Com\_00719]** [In case a `ComIPduCallout` is configured for an I-PDU with `ComIPduDirection` configured to *SEND*, the AUTOSAR COM module shall invoke this I-PDU callout directly before the I-PDU is transmitted via `PduR_ComTransmit`.] (SRS\_Com\_02037)

For TP I-PDUs, the I-PDU callout functions are handled similar to the I-PDU callout functions of normal I-PDUs and are supported both on transmitter and receiver side. Note that for example the requirement `SWS_Com_00719` does not consider the configured `ComIPduType`.

**[SWS\_Com\_00720]** [When a large I-PDU is received successfully via the TP interface, the I-PDU callout shall take place within the invocation of `Com_TpRxIndication`.] (SRS\_Com\_02095)

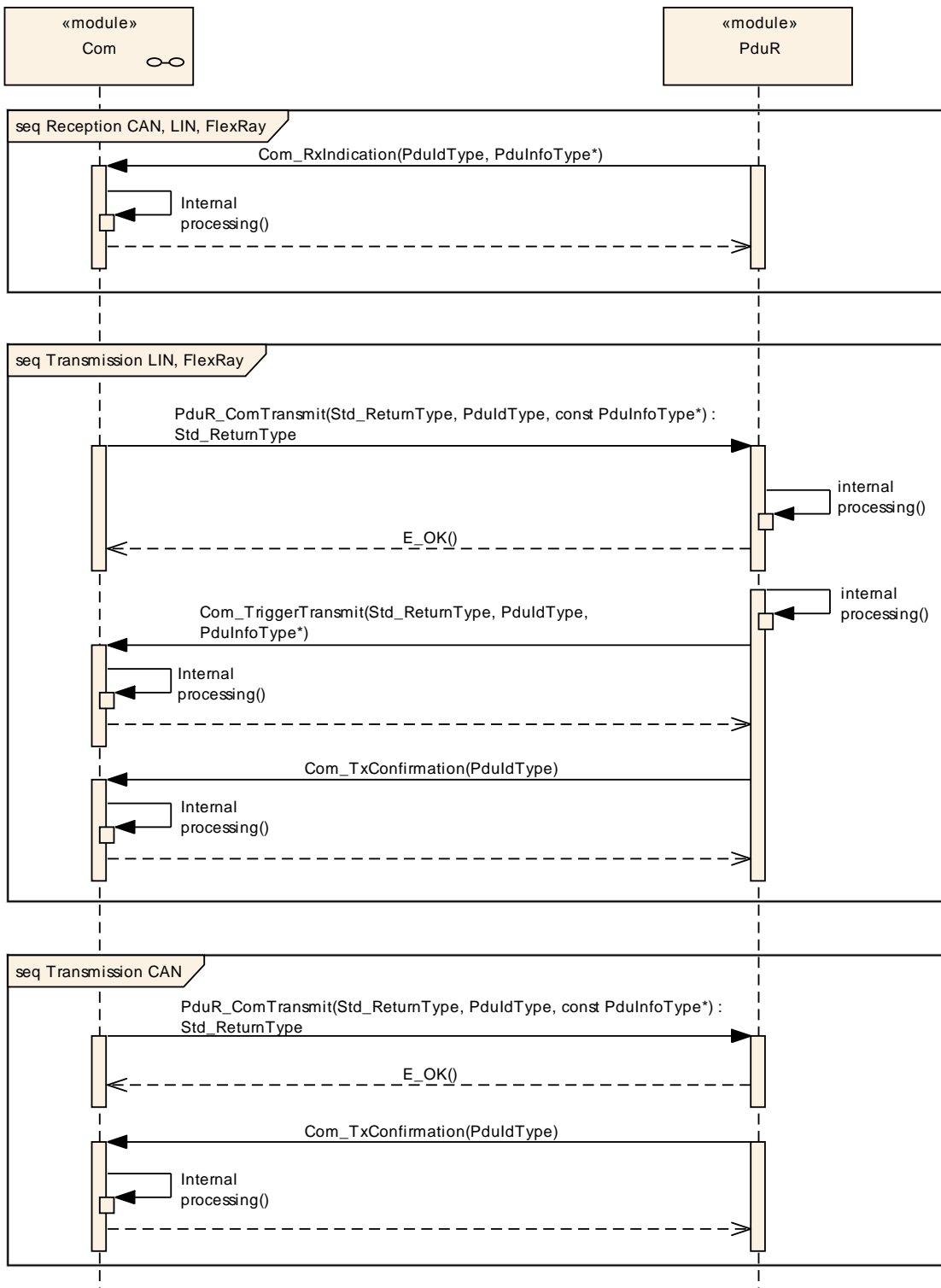
**[SWS\_Com\_00839]** [When the I-PDU callout in context of `Com_TpRxIndication` returns false, the I-PDU shall be considered as incorrectly received and therefore all included signals shall be treated exactly like invalidated signals. That is, the AUTOSAR COM module shall execute all configured `ComDataInvalidActions` for all included signals and signal groups.] (SRS\_Com\_02095)

## 9 Sequence Diagrams

A sequence diagram of the underlying OSEK COM communication stack can be found in [17].

### 9.1 Interface between the AUTOSAR COM Module and the PDU Router

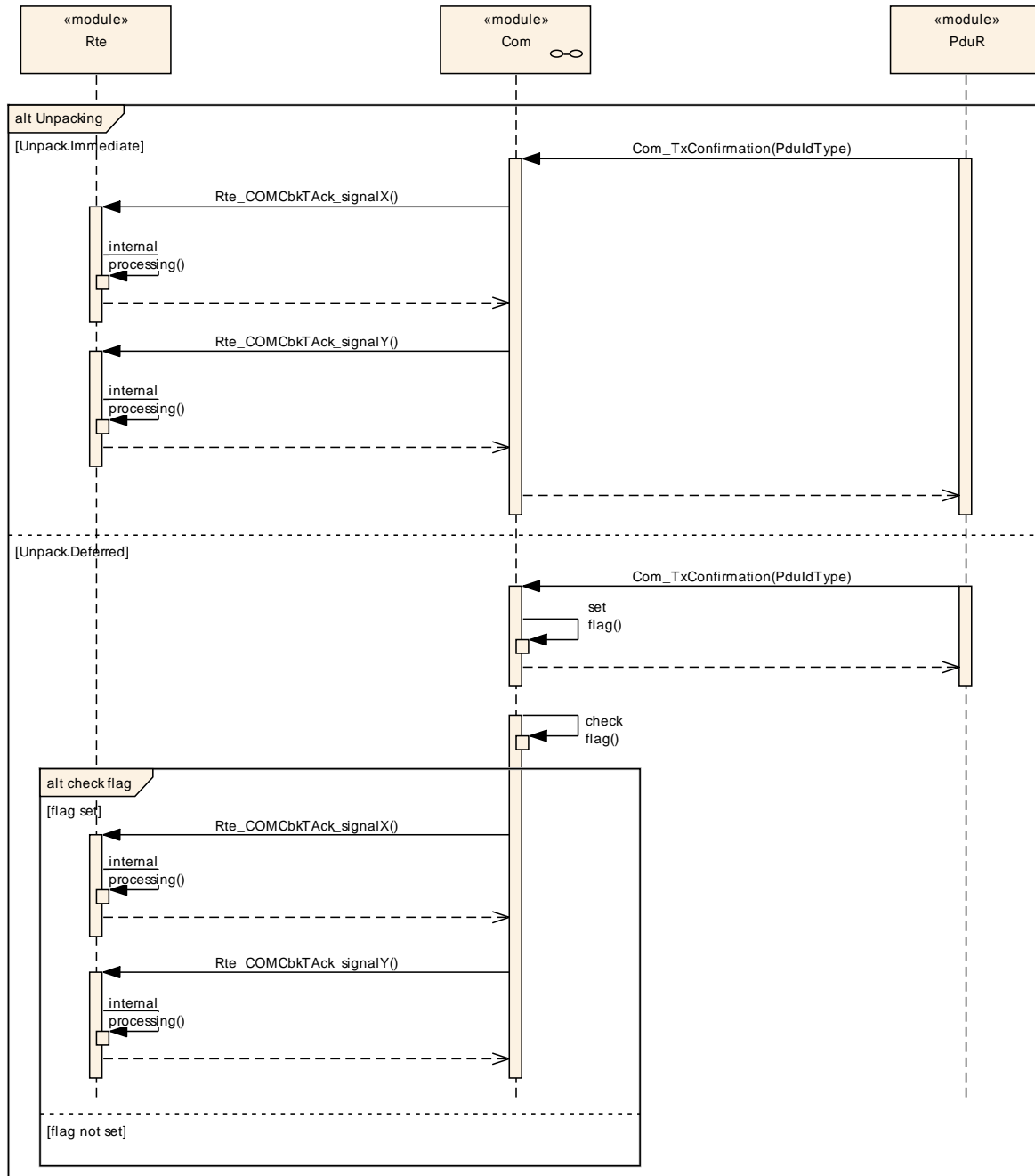
The following chart shows the communication between the AUTOSAR COM module and the PDU Router.



**Figure 8: Interactions between the AUTOSAR COM module and the PDU router**

## 9.2 Confirmation Handling between the PDU Router, the AUTOSAR COM Module and the RTE

The following chart shows the confirmation handling with respect to the two different IPDU-processing modes. (See also Chapter 7.3.5.4.)



**Figure 9: Confirmation handling between the PDU Router, the AUTOSAR COM module and the RTE**

### 9.3 Indication Handling between the PDU Router, the AUTOSAR COM Module and the RTE

The following chart shows the indication handling with respect to the two different unpacking modes. (See also Chapter 7.3.5.4.)

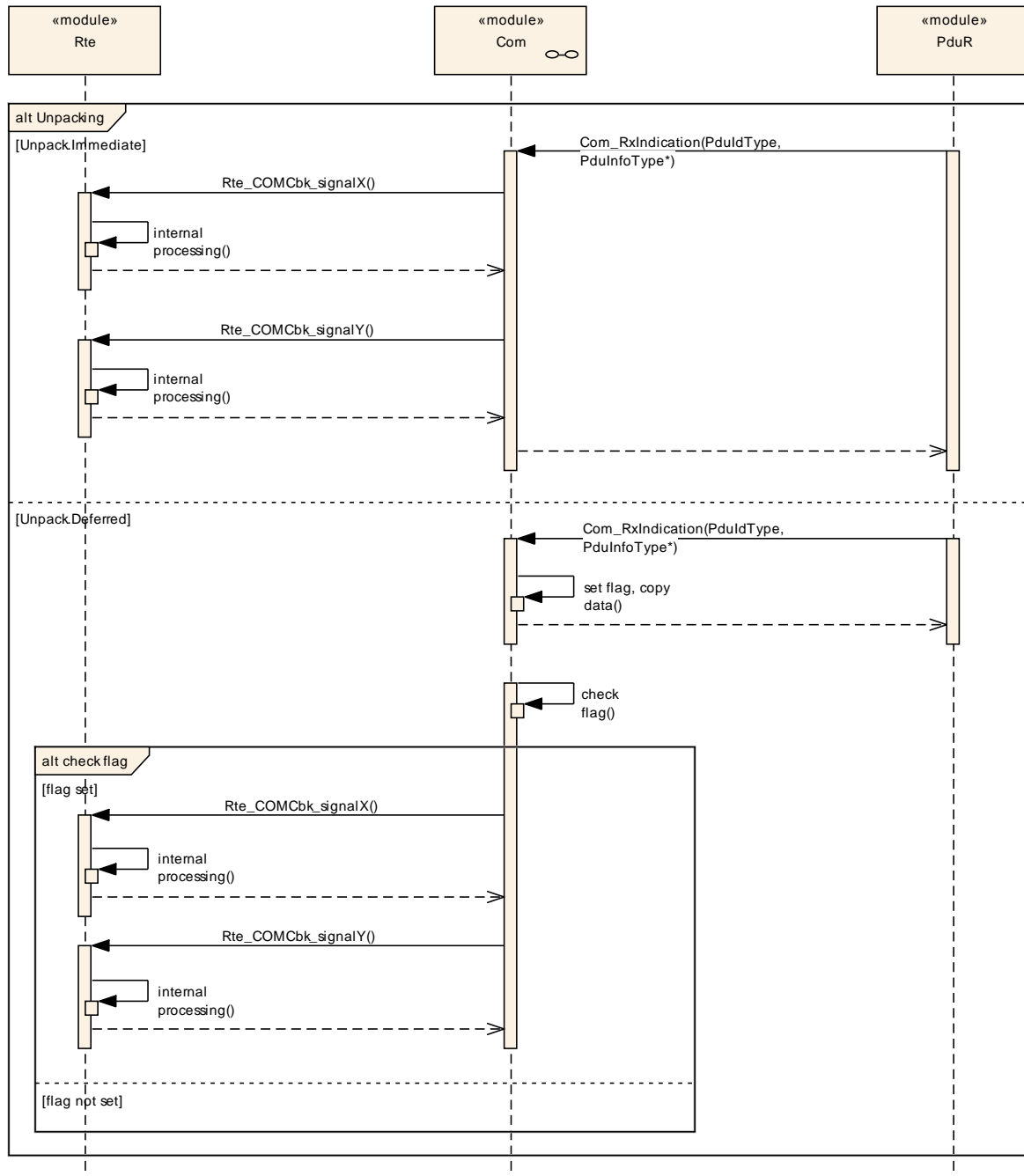


Figure 10: Indication handling between the PDU Router, the AUTOSAR COM module and the RTE



## 10 Configuration Specification

### 10.1 Containers and Configuration Parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters are described in Chapter 7 and Chapter 8.

**[SWS\_Com\_00006]** [The configuration parameters are based on [18]. All parameters shall be stored in an XML format.](SRS\_Com\_02040)

#### 10.1.1 Variants

Currently three configuration variants for the AUTOSAR COM module are defined:

##### 10.1.1.1 VARIANT-PRE-COMPILE

**[SWS\_Com\_00606]** [VARIANT-PRE-COMPILE only supports pre-compile configurable parameters. Parameters below that are marked as Pre-compile configurable shall be configurable in a pre-compile manner, for example as #defines. A VARIANT-PRE-COMPILE module is most likely delivered as source code.] (SRS\_BSW\_00345)

**Remark:** Even though the module is delivered as source code, the implementation might use techniques similar to link time, i.e. table driven configuration.

##### 10.1.1.2 VARIANT-LINK-TIME

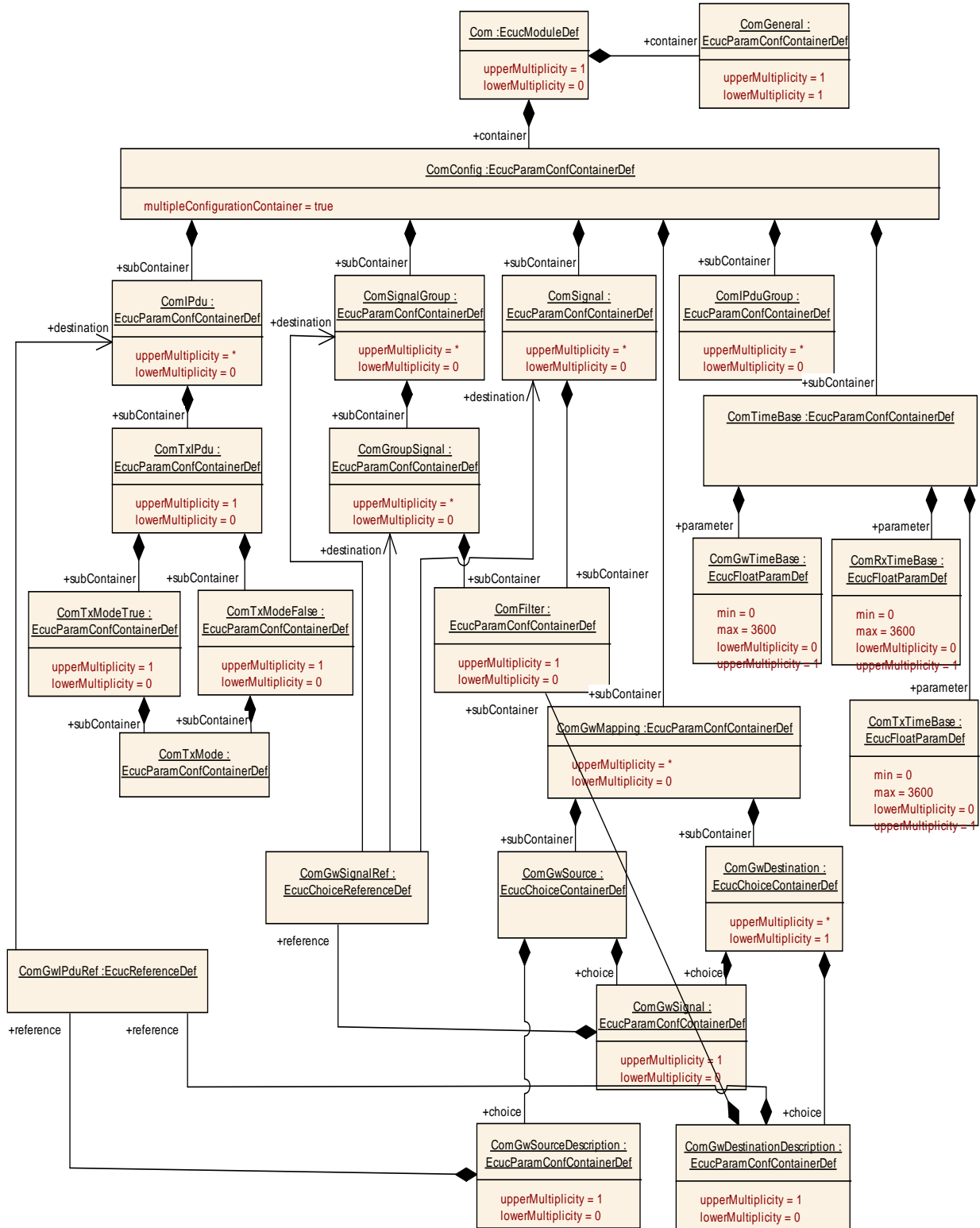
**[SWS\_Com\_00607]** [VARIANT-LINK-TIME includes mainly link-time and some pre-compile configurable parameters. All parameters defined below as link-time configurable shall be configurable at link time for example by linking a special configured parameter object file. A VARIANT-LINK-TIME module is most likely delivered as object code.] (SRS\_BSW\_00344)

##### 10.1.1.3 VARIANT-POST-BUILD

**[SWS\_Com\_00608]** [VARIANT-POST-BUILD includes post-build-time, link-time and some pre-compile configurable parameters. All parameters defined below as post build configurable shall be configurable post build for example by flashing configuration data. A VARIANT-POST-BUILD configurable module is most likely delivered as object code.] (SRS\_BSW\_00404)

### 10.1.2 Configuration of the AUTOSAR COM Module

For an overview of the AUTOSAR COM module's configuration, see Figure 11.



**Figure 11: The AUTOSAR COM module's Configuration Overview**

### 10.1.3 Com

<b>SWS Item</b>	<b>ECUC_Com_00540 :</b>
<b>Module Name</b>	Com
<b>Module Description</b>	Configuration of the AUTOSAR COM module.
<b>Post-Build Variant Support</b>	true

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComConfig	1	This container contains the configuration parameters and sub containers of the AUTOSAR COM module. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.
ComGeneral	1	Contains the general configuration parameters of the AUTOSAR COM module.

### 10.1.4 ComGeneral

<b>SWS Item</b>	<b>ECUC_Com_00541 :</b>
<b>Container Name</b>	ComGeneral
<b>Description</b>	Contains the general configuration parameters of the module.
<b>Configuration Parameters</b>	

<b>SWS Item</b>	<b>ECUC_Com_10000 :</b>		
<b>Name</b>	ComCancellationSupport		
<b>Description</b>	This parameter enables/disables the cancellation feature:  true: enabled false: disabled		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	true		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00141 :</b>		
<b>Name</b>	ComConfigurationUseDet		
<b>Description</b>	The error hook shall contain code to call the Det.  If this parameter is configured COM_DEV_ERROR_DETECT shall be set to ON as output of the configuration tool. (as input for the source code).		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		

<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00788 :</b>		
<b>Name</b>	ComEnableMDTForCyclicTransmission		
<b>Description</b>	Enables globally for the whole Com module the minimum delay time monitoring for cyclic and repeated transmissions (ComTxModeMode=PERIODIC or ComTxModeMode=MIXED for the cyclic transmissions, ComTxModeNumberOfRepetitions > 0 for repeated transmissions).		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	true		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_10002 :</b>		
<b>Name</b>	ComEnableSignalGroupArrayApi		
<b>Description</b>	Activate/Deactivate the signal group array access APIs (Com_SendSignalGroupArray, Com_ReceiveSignalGroupArray).  true: signal group array access APIs activated  false: signal group array access APIs deactivated		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_10004 :</b>		
<b>Name</b>	ComMetaDataSupport		
<b>Description</b>	This parameter enables/disables the support of meta-data feature including the API Com_TriggerIPDUSendWithMetaData.  true: enabled false: disabled		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		

<b>Default value</b>	false		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00780 :</b>		
<b>Name</b>	ComRetryFailedTransmitRequests		
<b>Description</b>	If this Parameter is set to true, retry of failed transmission requests is enabled. If this Parameter is not present, the default value is assumed.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00710 :</b>		
<b>Name</b>	ComSupportedIPduGroups		
<b>Description</b>	Defines the maximum number of supported I-PDU groups.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 65535		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_10001 :</b>		
<b>Name</b>	ComUserCbkHeaderFile		
<b>Description</b>	Defines the header files for callback functions which shall be included by the COM module.		
<b>Multiplicity</b>	0..*		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	--		
<b>maxLength</b>	32		
<b>minLength</b>	1		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		

<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00438 :</b>		
<b>Name</b>	ComVersionInfoApi		
<b>Description</b>	Activate/Deactivate the version information API (Com_GetVersionInfo).  True: version information API activated False: version information API deactivated		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**

### 10.1.5 ComConfig

<b>SWS Item</b>	<b>ECUC_Com_00337 :</b>		
<b>Container Name</b>	ComConfig		
<b>Description</b>	This container contains the configuration parameters and sub containers of the AUTOSAR COM module.		
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_Com_00783 :</b>		
<b>Name</b>	ComDataMemSize		
<b>Description</b>	Size of internal Com data in units of bytes (static memory allocation) - memory required by post-build configuration must be smaller than this constant. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 ..	18446744073709551615	
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE

	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00782 :</b>		
<b>Name</b>	ComMaxIPduCnt		
<b>Description</b>	Maximum number of IPdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 18446744073709551615		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComGwMapping	0..*	Each instance of this container defines one mapping of the integrated Signal Gateway.
ComIPdu	0..*	See ECUC_Com_00340, if there is no ComIpdu container included no IPDUs are defined. In this case no communication via the AUTOSAR COM module is possible.
ComIPduGroup	0..*	See ECUC_Com_00341, if there is no ComIPduGroup container included then no IPDU group is defined. In this case no communication via the AUTOSAR COM module is possible.
ComSignal	0..*	See ECUC_Com_00344, if there is no ComSignal container included no single signals are defined.
ComSignalGroup	0..*	See ECUC_Com_00345, if there is no ComSignalGroup container included no signal groups are defined.
ComTimeBase	1	Contains the timebase parameters for Tx, Rx and routing.

### 10.1.6 ComTimeBase

<b>SWS Item</b>	<b>ECUC_Com_00730 :</b>		
<b>Container Name</b>	ComTimeBase		
<b>Description</b>	Contains the timebase parameters for Tx, Rx and routing.		
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_Com_00729 :</b>		
<b>Name</b>	ComGwTimeBase		
<b>Description</b>	The period between successive calls to Com_MainFunctionRouteSignals		

	<p>in seconds. This parameter may be used by the COM generator to transform the values of the signal gateway related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific.</p> <p>The COM module (generator) might rely on the fact that Com_MainFunctionRouteSignals is scheduled according to the value configured here.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0 .. 3600		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_Com_00728 :</b>		
<b>Name</b>	ComRxTimeBase		
<b>Description</b>	<p>The period between successive calls to Com_MainFunctionRx in seconds. This parameter may be used by the COM generator to transform the values of the reception related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific.</p> <p>The COM module (generator) may rely on the fact that Com_MainFunctionRx is scheduled according to the value configured here.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0 .. 3600		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_Com_00186 :</b>		
<b>Name</b>	ComTxTimeBase		
<b>Description</b>	The period between successive calls to Com_MainFunctionTx in seconds.		



	<p>This parameter may be used by the COM generator to transform the values of the transmission related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific.</p> <p>The COM module (generator) may rely on the fact that Com_MainFunctionTx is scheduled according to the value configured here.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0 .. 3600		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

**No Included Containers**

### 10.1.7 ComFilter

<b>SWS Item</b>	<b>ECUC_Com_00339 :</b>
<b>Container Name</b>	ComFilter
<b>Description</b>	<p>This container contains the configuration parameters of the AUTOSAR COM module's Filters.</p> <p>Note: On sender side the container is used to specify the transmission mode conditions.</p>
<b>Configuration Parameters</b>	

<b>SWS Item</b>	<b>ECUC_Com_00146 :</b>	
<b>Name</b>	ComFilterAlgorithm	
<b>Description</b>	The range of values is specified in the [17] specification, chapter 2.2.2, Reception Filtering.	
<b>Multiplicity</b>	1	
<b>Type</b>	EcucEnumerationParamDef	
<b>Range</b>	ALWAYS	--
	MASKED_NEW_DIFFERS_MASKED_OLD	--
	MASKED_NEW_DIFFERS_X	--
	MASKED_NEW_EQUALS_X	--
	NEVER	--
	NEW_IS_OUTSIDE	--
	NEW_IS_WITHIN	--
	ONE EVERY_N	--
<b>Post-Build Vari</b>	false	

<b>ant Value</b>			
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00235 :</b>		
<b>Name</b>	ComFilterMask		
<b>Description</b>	The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	-2147483648 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00317 :</b>		
<b>Name</b>	ComFilterMax		
<b>Description</b>	The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	-2147483648 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00318 :</b>		
<b>Name</b>	ComFilterMin		
<b>Description</b>	The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		

<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	-2147483648 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00313 :</b>		
<b>Name</b>	ComFilterOffset		
<b>Description</b>	The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.  Range = 0..(ComFilterPeriod-1)		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 4294967294		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local dependency: ECUC_Com_00312		

<b>SWS Item</b>	<b>ECUC_Com_00312 :</b>		
<b>Name</b>	ComFilterPeriod		
<b>Description</b>	This parameter defines the period of the ComFilterAlgorithm ONE_EVERY_N.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	2 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00147 :</b>		
<b>Name</b>	ComFilterX		
<b>Description</b>	The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcuIntegerParamDef		
<b>Range</b>	-2147483648 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**

### 10.1.8 ComIPdu

<b>SWS Item</b>	<b>ECUC_Com_00340 :</b>		
<b>Container Name</b>	ComIPdu		
<b>Description</b>	Contains the configuration parameters of the AUTOSAR COM module's I-PDUs.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_Com_00387 :</b>		
<b>Name</b>	ComIPduCallout		
<b>Description</b>	This parameter defines the existence and the name of a callout function for the corresponding I-PDU. If this parameter is omitted no I-PDU callout shall take place for the corresponding I-PDU.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcuFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		

<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00709 :</b>		
<b>Name</b>	ComIPduCancellationSupport		
<b>Description</b>	Defines for I-PDUs with ComIPduType NORMAL: If the underlying IF-modul supports cancellation of transmit requests.  Defines for I-PDUs with ComIPduType TP: If the underlying TP-module supports RX and TX cancellation of ongoing requests.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	--	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	--	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU dependency: This parameter shall not be set to true if ComCancellationSupport is set to false		

<b>SWS Item</b>	<b>ECUC_Com_00493 :</b>		
<b>Name</b>	ComIPduDirection		
<b>Description</b>	The direction defines if this I-PDU, and therefore the contributing signals and signal groups, shall be sent or received.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	RECEIVE	--	
	SEND	--	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local dependency: If configured to Sent also a ComTxIpdu container shall be included, see ECUC_Com_00496		

<b>SWS Item</b>	<b>ECUC_Com_00175 :</b>		
<b>Name</b>	ComIPduHandleId		
<b>Description</b>	The numerical value used as the ID of this I-PDU. The ComIPduHandleId is required by the API calls Com_RxIndication, Com_TpRxIndication,		

	Com_StartOfReception and Com_CopyRxData to receive I-PDUs from the PduR (ComIP-duDirection: Receive), as well as the PduId passed to an Rx-I-PDU-callout. For Tx-I-PDUs (ComIPduDirection: Send), this handle Id is used for the APIs calls Com_TxConfirmation, Com_TriggerTransmit, Com_TriggerIPDUSend or Com_TriggerIPDUSendWithMetaData, Com_CopyTxData and Com_TpTxConfirmation to transmit respectively confirm transmissions of I-PDUs, as well as the PduId passed to the Tx-I-PDU-callout configured with ComIPduCallout and/or ComIPduTrigger-TransmitCallout.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 65535		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_Com_00119 :</b>		
<b>Name</b>	ComIPduSignalProcessing		
<b>Description</b>	For the definition of the two modes Immediate and Deferred.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	DEFERRED	signal indication / confirmations are deferred for example to a cyclic task	
	IMMEDIATE	the signal indications / confirmations are performed in Com_RxIndication/ Com_TxConfirmation	
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00765 :</b>		
<b>Name</b>	ComIPduTriggerTransmitCallout		
<b>Description</b>	If there is a trigger transmit callout defined for this I-PDU this parameter contains the name of the callout function.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	

	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00761 :</b>		
<b>Name</b>	ComIPduType		
<b>Description</b>	Defines if this I-PDU is a normal I-PDU that can be sent unfragmented or if this is a large I-PDU that shall be sent via the Transport Protocol of the underlying bus.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	NORMAL		sent or received via normal L-PDU
	TP		sent or received via TP
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00206 :</b>		
<b>Name</b>	ComIPduGroupRef		
<b>Description</b>	Reference to the I-PDU groups this I-PDU belongs to.		
<b>Multiplicity</b>	0..*		
<b>Type</b>	Reference to [ ComIPduGroup ]		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00519 :</b>		
<b>Name</b>	ComIPduSignalGroupRef		
<b>Description</b>	References to all signal groups contained in this I-Pdu		
<b>Multiplicity</b>	0..*		
<b>Type</b>	Reference to [ ComSignalGroup ]		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00518 :</b>		
<b>Name</b>	ComIPduSignalRef		

<b>Description</b>	References to all signals contained in this I-PDU.		
<b>Multiplicity</b>	0..*		
<b>Type</b>	Reference to [ ComSignal ]		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00711 :</b>		
<b>Name</b>	ComPduIdRef		
<b>Description</b>	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to [ Pdu ]		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>			

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComIPduCounter	0..1	This optional container contains the configuration parameters of PDU Counter.
ComIPduReplication	0..1	This optional container contains the information needed for each I-PDU replicated.
ComTxIPdu	0..1	This container must be included if COM_IPDU_DIRECTION is configured to SEND.

[SWS\_Com\_00497] [A ComTxIPdu container must be included if ComIPduDirection is configured to SEND.] (SRS\_BSW\_00167)

### 10.1.9 ComIPduCounter

<b>SWS Item</b>	<b>ECUC_Com_00592 :</b>	
<b>Container Name</b>	ComIPduCounter	
<b>Description</b>	This optional container contains the configuration parameters of PDU Counter.	
<b>Configuration Parameters</b>		

<b>SWS Item</b>	<b>ECUC_Com_00003 :</b>	
<b>Name</b>	ComIPduCounterErrorNotification	
<b>Description</b>	Name of Com_CbkCounterErr callback function to be called. If this parameter is omitted no I-PDU counter mismatch notification shall take place.	
<b>Multiplicity</b>	0..1	



<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00593 :</b>		
<b>Name</b>	ComIPduCounterSize		
<b>Description</b>	Size of PDU Counter expressed in bits		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 8		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00594 :</b>		
<b>Name</b>	ComIPduCounterStartPosition		
<b>Description</b>	Position of PDU counter expressed in bits from start position of data content of I-PDU (SDU). Note that PDU counter is not allowed to cross a byte border. The parameter ComIPduCounterStartPosition shall define the bit of the first byte like in little endian byte order.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00595 :</b>		
<b>Name</b>	ComIPduCounterThreshold		
<b>Description</b>	Threshold value of I-PDU counter algorithm, see ECUC_Com_00590.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	0		

<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**

### 10.1.10 ComIPduReplication

<b>SWS Item</b>	<b>ECUC_Com_00599 :</b>		
<b>Container Name</b>	ComIPduReplication		
<b>Description</b>	This optional container contains the information needed for each I-PDU replicated.		
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_Com_00600 :</b>		
<b>Name</b>	ComIPduReplicationQuorum		
<b>Description</b>	The number of identical I-PDUs needed for successful voting.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	1 .. 3		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local dependency: This value must be less or equal to the multiplicity of ( ComIPduReplicaRef + 1).		

<b>SWS Item</b>	<b>ECUC_Com_00601 :</b>		
<b>Name</b>	ComIPduReplicaRef		
<b>Description</b>	Reference to replicas PduR PDUs of this IPDU.		
<b>Multiplicity</b>	1..2		
<b>Type</b>	Reference to [ Pdu ]		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

**No Included Containers**

## 10.1.11 ComTxIPdu

<b>SWS Item</b>	<b>ECUC_Com_00496 :</b>
<b>Container Name</b>	ComTxIPdu
<b>Description</b>	This container contains additional transmission related configuration parameters of the AUTOSAR COM module's I-PDUs.
<b>Configuration Parameters</b>	

<b>SWS Item</b>	<b>ECUC_Com_10005 :</b>		
<b>Name</b>	ComMetaDataDefault		
<b>Description</b>	<p>In case an I-PDU has a globally configured MetaDataLength distinct from 0 and no explicit meta data is given for a send request (e.g. by using Com_TriggerIPDUSendWithMetaData ), The AUTOSAR COM module uses this configured default meta data for sending.</p> <p>The configured string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address).</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local dependency: This parameter shall only be used in case ComMetaDataSupport is enabled. The length of the default meta data shall match the globally configured MetaDataLength of this I-PDU.		

<b>SWS Item</b>	<b>ECUC_Com_00181 :</b>		
<b>Name</b>	ComMinimumDelayTime		
<b>Description</b>	Defines the Minimum Delay Time (MDT) between successive transmissions of this I-PDU in seconds. The MDT is independent of the possible different transmission modes. There is only one minimum delay time parameter for one I-PDU. The minimum delay timer is not reset by changing the transmission mode. Hence, it is not allowed to violate the minimum delay time by transmission mode changes. It is not possible to monitor the minimum delay time for I-PDUs that are requested using the Com_TriggerTransmit API.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0 .. 3600		

<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00576 :</b>		
<b>Name</b>	ComTxIPduClearUpdateBit		
<b>Description</b>	Defines when the update-bits of signals or signal groups, contained in this I-PDU, will be cleared.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	Confirmation	The update-bits are cleared when the transmission of the I-PDU was confirmed. In case of Direct/N-Times transmission mode the update bits will be cleared with respect to the confirmation behaviour of ECUC_Com_00305.	
	Transmit	The update-bits are cleared directly after the invocation of PduR_ComTransmit.	
	TriggerTransmit	The update-bits are cleared after the I-PDU was fetched via Com_TriggerTransmit.	
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00017 :</b>		
<b>Name</b>	ComTxIPduUnusedAreasDefault		
<b>Description</b>	The AUTOSAR COM module fills not used areas of an I-PDU with this byte pattern. This attribute is mandatory to avoid undefined behaviour. This byte-pattern will be repeated throughout the I-PDU before any init-values or update-bits were set.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComTxModeFalse	0..1	The referenced transmission mode object that is used when the filtering state for this I-PDU evaluates to false. The default is transmission mode None.
ComTxModeTrue	0..1	The referenced transmission mode object that is used when the filtering state for this I-PDU evaluates to true.

### 10.1.12 ComIPduGroup

<b>SWS Item</b>		<b>ECUC_Com_00341 :</b>		
<b>Container Name</b>		ComIPduGroup		
<b>Description</b>		Contains the configuration parameters of the AUTOSAR COM module's I-PDU groups.		
<b>Post-Build Variant Multiplicity</b>		true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE	
	<b>Link time</b>	X	VARIANT-LINK-TIME	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD	
<b>Configuration Parameters</b>				

<b>SWS Item</b>		<b>ECUC_Com_00184 :</b>		
<b>Name</b>		ComIPduGroupHandleId		
<b>Description</b>		The numerical value used as the ID of this I-PDU Group .  The ComIPduGroupHandleId is required by the API calls to start and stop I-PDU Groups.  Range: 0 .. (ComSupportedIPduGroups-1)		
<b>Multiplicity</b>		1		
<b>Type</b>		EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>		0 .. 65535		
<b>Default value</b>		--		
<b>Post-Build Variant Value</b>		false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants	
	<b>Link time</b>	--		
	<b>Post-build time</b>	--		
<b>Scope / Dependency</b>		scope: ECU		

<b>SWS Item</b>		<b>ECUC_Com_00185 :</b>		
<b>Name</b>		ComIPduGroupGroupRef		
<b>Description</b>		References to all I-PDU groups that includes this I-PDU group. If this reference is omitted this I-PDU group does not belong to another I-PDU group.		
<b>Multiplicity</b>		0..*		
<b>Type</b>		Reference to [ ComIPduGroup ]		
<b>Post-Build Variant Multiplicity</b>		true		
<b>Post-Build Variant Value</b>		true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE	
	<b>Link time</b>	X	VARIANT-LINK-TIME	
	<b>Post-build time</b>	X	VARIANT-POST-BUILD	
<b>Value Configuration Class</b>		<b>Pre-compile time</b> X VARIANT-PRE-COMPILE		

	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**

### 10.1.13 ComSignal

<b>SWS Item</b>	<b>ECUC_Com_00344 :</b>		
<b>Container Name</b>	ComSignal		
<b>Description</b>	Contains the configuration parameters of the AUTOSAR COM module's signals.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_Com_00259 :</b>		
<b>Name</b>	ComBitPosition		
<b>Description</b>	Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order		
<b>Multiplicity</b>	1		
<b>Type</b>	EcuIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00158 :</b>		
<b>Name</b>	ComBitSize		
<b>Description</b>	Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN  the size shall be configured by ComSignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcuIntegerParamDef		
<b>Range</b>	0 .. 64		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00314 :</b>		
<b>Name</b>	ComDataInvalidAction		
<b>Description</b>	This parameter defines the action performed upon reception of an invalid signal.  Relating to signal groups the action in case if one of the included signals is an invalid signal. If Replace is used the ComSignalInitValue will be used for the replacement.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	NOTIFY	--	
	REPLACE	Literal for DataInvalidAction	
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00499 :</b>		
<b>Name</b>	ComErrorNotification		
<b>Description</b>	Only valid on sender side: Name of Com_CbkTxErr callback function to be called.  If this parameter is omitted no error notification shall take place.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00183 :</b>		
<b>Name</b>	ComFirstTimeout		
<b>Description</b>	Defines the length of the first deadline monitoring timeout period in seconds. This timeout is used immediately after start (or restart) of the deadline monitoring service. The timeout period of the successive periods is configured by ECUC_Com_00263.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0 .. 3600		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00165 :</b>		
<b>Name</b>	ComHandleId		
<b>Description</b>	The numerical value used as the ID.  For signals it is required by the API calls Com_UpdateShadowSignal, Com_ReceiveShadowSignal and Com_InvalidateShadowSignal. For signals groups it is required by the Com_SendSignalGroup and Com_ReceiveSignalGroup calls.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 65535		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_Com_00811 :</b>		
<b>Name</b>	ComInitialValueOnly		
<b>Description</b>	This parameter defines that the respective signal's initial value shall be put into the respective PDU but there will not be any update of the value through the RTE. Thus the Com implementation does not need to expect any API calls for this signal (group).		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		



<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00315 :</b>		
<b>Name</b>	ComInvalidNotification		
<b>Description</b>	Only valid on receiver side: Name of Com_CbkInv callback function to be called. Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComDataInvalidAction is configured to NOTIFY.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00498 :</b>		
<b>Name</b>	ComNotification		
<b>Description</b>	On sender side: Name of Com_CbkTxAck callback function to be called.  On receiver side: Name of Com_CbkRxAck callback function to be called.  If this parameter is omitted no notification shall take place.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE

	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00412 :</b>		
<b>Name</b>	ComRxDataTimeoutAction		
<b>Description</b>	This parameter defines the action performed upon expiration of the reception deadline monitoring timer.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	NONE	no replacement shall take place	
	REPLACE	signals shall be replaced by their ComSignalInitValue	
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00391 :</b>		
<b>Name</b>	ComSignalDataInvalidValue		
<b>Description</b>	<p>Defines the data invalid value of the signal.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE

<b>Class</b>	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local dependency: In case of UINT8_N the length of ComSignalDataInvalidValue has to be the same as ComSignalLength.		

<b>SWS Item</b>	<b>ECUC_Com_00157 :</b>		
<b>Name</b>	ComSignalEndianness		
<b>Description</b>	Defines the endianness of the signal's network representation.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	BIG_ENDIAN	--	
	LITTLE_ENDIAN	--	
	OPAQUE	--	
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00170 :</b>		
<b>Name</b>	ComSignalInitValue		
<b>Description</b>	<p>Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of UINT8_DYN the initial size shall be 0.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0 (lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	0		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE

<b>Class</b>	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local dependency: In case of UINT8_N the length of ComSignalNitValue has to be the same as ComSignalLength.		

<b>SWS Item</b>	<b>ECUC_Com_00437 :</b>		
<b>Name</b>	ComSignalLength		
<b>Description</b>	<p>Description:</p> <p>For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored.</p> <p>Range: 0..8 for normal CAN/ LIN I-PDUs, 0..64 for CAN FD I-PDUs, 0..254 for normal FlexRay I-PDUs (all of ComIPduType NORMAL), 0..4294967295 for I-PDUs with ComIPduType TP.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00127 :</b>		
<b>Name</b>	ComSignalType		
<b>Description</b>	<p>The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute.</p> <p>This type could also be used to reserved appropriate storage in AUTOSAR COM.</p>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	BOOLEAN	--	
	FLOAT32	--	
	FLOAT64	--	
	SINT16	--	
	SINT32	--	
	SINT64	--	
	SINT8	--	
	UINT16	--	
	UINT32	--	
	UINT64	--	
UINT8	--		

	UINT8_DYN	--	
	UINT8_N	--	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00263 :</b>		
<b>Name</b>	ComTimeout		
<b>Description</b>	Defines the length of the deadline monitoring timeout period in seconds. The period for the first timeout period can be configured separately by ECUC_Com_00183.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0 .. 3600		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00552 :</b>		
<b>Name</b>	ComTimeoutNotification		
<b>Description</b>	On sender side: Name of Com_CbkTxTOut callback function to be called.		
	On receiver side: Name of Com_CbkRxTOut callback function to be called.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00232 :</b>		
-----------------	-------------------------	--	--

<b>Name</b>	ComTransferProperty	
<b>Description</b>	Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	EcucEnumerationParamDef	
<b>Range</b>	PENDING	A write access to this signal never triggers the transmission of the corresponding I-PDU.
	TRIGGERED	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU.
	TRIGGERED_ON_CHANGE	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.
	TRIGGERED_ON_CHANGE_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.
	TRIGGERED_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition.
<b>Post-Build Variant Multiplicity</b>	true	
<b>Post-Build Variant Value</b>	true	
<b>Multiplicity Configuration Class</b>	<i>Pre-compile time</i>	X VARIANT-PRE-COMPILE
	<i>Link time</i>	X VARIANT-LINK-TIME
	<i>Post-build time</i>	X VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<i>Pre-compile time</i>	X VARIANT-PRE-COMPILE
	<i>Link time</i>	X VARIANT-LINK-TIME
	<i>Post-build time</i>	X VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local	

<b>SWS Item</b>	<b>ECUC_Com_00257 :</b>		
<b>Name</b>	ComUpdateBitPosition		
<b>Description</b>	<p>Bit position of update-bit inside I-PDU.</p> <p>If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side.</p> <p>Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcuIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00002 :</b>		
<b>Name</b>	ComSystemTemplateSystemSignalRef		
<b>Description</b>	Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Foreign reference to [ I-SIGNAL-TO-I-PDU-MAPPING ]		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU		

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComFilter	0..1	<p>This container contains the configuration parameters of the AUTOSAR COM module's Filters.</p> <p>Note: On sender side the container is used to specify the transmission mode conditions.</p>

### 10.1.14 ComSignalGroup

<b>SWS Item</b>	<b>ECUC_Com_00345 :</b>
-----------------	-------------------------

<b>Container Name</b>	ComSignalGroup		
<b>Description</b>	Contains the configuration parameters of the AUTOSAR COM module's signal groups.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_Com_00314 :</b>		
<b>Name</b>	ComDataInvalidAction		
<b>Description</b>	<p>This parameter defines the action performed upon reception of an invalid signal.</p> <p>Relating to signal groups the action in case if one of the included signals is an invalid signal. If Replace is used the ComSignalInitValue will be used for the replacement.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	NOTIFY	--	
	REPLACE		Literal for DataInvalidAction
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00499 :</b>		
<b>Name</b>	ComErrorNotification		
<b>Description</b>	<p>Only valid on sender side: Name of Com_CbkTxErr callback function to be called.</p> <p>If this parameter is omitted no error notification shall take place.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE



	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00183 :</b>		
<b>Name</b>	ComFirstTimeout		
<b>Description</b>	Defines the length of the first deadline monitoring timeout period in seconds. This timeout is used immediately after start (or restart) of the deadline monitoring service. The timeout period of the successive periods is configured by ECUC_Com_00263.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0 .. 3600		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00165 :</b>		
<b>Name</b>	ComHandleId		
<b>Description</b>	The numerical value used as the ID.  For signals it is required by the API calls Com_UpdateShadowSignal, Com_ReceiveShadowSignal and Com_InvalidateShadowSignal. For signals groups it is required by the Com_SendSignalGroup and Com_ReceiveSignalGroup calls.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 65535		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_Com_00811 :</b>		
<b>Name</b>	ComInitialValueOnly		
<b>Description</b>	This parameter defines that the respective signal's initial value shall be put into the respective PDU but there will not be any update of the value through the RTE. Thus the Com implementation does not need to expect any API calls for this signal (group).		
<b>Multiplicity</b>	0..1		

<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00315 :</b>		
<b>Name</b>	ComInvalidNotification		
<b>Description</b>	Only valid on receiver side: Name of Com_CbkInv callback function to be called. Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComDataInvalidAction is configured to NOTIFY.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00498 :</b>		
<b>Name</b>	ComNotification		
<b>Description</b>	On sender side: Name of Com_CbkTxAck callback function to be called.  On receiver side: Name of Com_CbkRxAck callback function to be called.  If this parameter is omitted no notification shall take place.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		

<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00412 :</b>		
<b>Name</b>	ComRxDataTimeoutAction		
<b>Description</b>	This parameter defines the action performed upon expiration of the reception deadline monitoring timer.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	NONE	no replacement shall take place	
	REPLACE	signals shall be replaced by their ComSignalInitValue	
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_10003 :</b>		
<b>Name</b>	ComSignalGroupArrayAccess		
<b>Description</b>	Defines whether the uint8-array based access shall be used for this ComSignalGroup.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>			

<b>SWS Item</b>	<b>ECUC_Com_00263 :</b>		
<b>Name</b>	ComTimeout		
<b>Description</b>	Defines the length of the deadline monitoring timeout period in seconds. The period for the first timeout period can be configured separately by		

	ECUC_Com_00183.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0 .. 3600		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00552 :</b>		
<b>Name</b>	ComTimeoutNotification		
<b>Description</b>	On sender side: Name of Com_CbkTxTOut callback function to be called. On receiver side: Name of Com_CbkRxTOut callback function to be called.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFunctionNameDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00232 :</b>		
<b>Name</b>	ComTransferProperty		
<b>Description</b>	Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	PENDING		A write access to this signal never triggers the transmission of the corresponding I-PDU.
	TRIGGERED		Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU.

	TRIGGERED_ON_CHANGE	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.
	TRIGGERED_ON_CHANGE_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.
	TRIGGERED_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition.
<b>Post-Build Variant Multiplicity</b>	true	
<b>Post-Build Variant Value</b>	true	
<b>Multiplicity Configuration Class</b>	<i>Pre-compile time</i>	X VARIANT-PRE-COMPILE
	<i>Link time</i>	X VARIANT-LINK-TIME
	<i>Post-build time</i>	X VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<i>Pre-compile time</i>	X VARIANT-PRE-COMPILE
	<i>Link time</i>	X VARIANT-LINK-TIME
	<i>Post-build time</i>	X VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local	

<b>SWS Item</b>	<b>ECUC_Com_00257 :</b>	
<b>Name</b>	ComUpdateBitPosition	
<b>Description</b>	<p>Bit position of update-bit inside I-PDU.</p> <p>If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side.</p> <p>Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.</p>	
<b>Multiplicity</b>	0..1	
<b>Type</b>	EcucIntegerParamDef	
<b>Range</b>	0 .. 4294967295	
<b>Default value</b>	--	
<b>Post-Build Variant Multiplicity</b>	true	

<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00001 :</b>		
<b>Name</b>	ComSystemTemplateSignalGroupRef		
<b>Description</b>	Reference to the ISignalToIPduMapping that contains a reference to the ISignalGroup (SystemTemplate) which this ComSignalGroup represents.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Foreign reference to [ I-SIGNAL-TO-I-PDU-MAPPING ]		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU		

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComGroupSignal	0..*	This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.

### 10.1.15 ComGroupSignal

<b>SWS Item</b>	<b>ECUC_Com_00520 :</b>		
<b>Container Name</b>	ComGroupSignal		
<b>Description</b>	This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_Com_00259 :</b>		
<b>Name</b>	ComBitPosition		
<b>Description</b>	Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 4294967295		

<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00158 :</b>		
<b>Name</b>	ComBitSize		
<b>Description</b>	Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN  the size shall be configured by ComSignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 64		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00165 :</b>		
<b>Name</b>	ComHandleId		
<b>Description</b>	The numerical value used as the ID.  For signals it is required by the API calls Com_UpdateShadowSignal, Com_ReceiveShadowSignal and Com_InvalidateShadowSignal. For signals groups it is required by the Com_SendSignalGroup and Com_ReceiveSignalGroup calls.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
<b>Range</b>	0 .. 65535		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: ECU		

<b>SWS Item</b>	<b>ECUC_Com_00391 :</b>		
<b>Name</b>	ComSignalDataInvalidValue		

<b>Description</b>	<p>Defines the data invalid value of the signal.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0 (lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured signal characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	--		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	<p>scope: local</p> <p>dependency: In case of UINT8_N the length of ComSignalDataInvalidValue has to be the same as ComSignalLength.</p>		

<b>SWS Item</b>	<b>ECUC Com 00157 :</b>		
<b>Name</b>	ComSignalEndianness		
<b>Description</b>	Defines the endianness of the signal's network representation.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	BIG_ENDIAN	--	
	LITTLE_ENDIAN	--	
	OPAQUE	--	
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC Com 00170 :</b>		
<b>Name</b>	ComSignalInitValue		
<b>Description</b>	Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of		



	<p>UINT8_DYN the initial size shall be 0.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	0		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	<p>scope: local</p> <p>dependency: In case of UINT8_N the length of ComSignalNitValue has to be the same as ComSignalLength.</p>		

<b>SWS Item</b>	<b>ECUC_Com_00437 :</b>		
<b>Name</b>	ComSignalLength		
<b>Description</b>	<p>Description:</p> <p>For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored.</p> <p>Range: 0..8 for normal CAN/ LIN I-PDUs, 0..64 for CAN FD I-PDUs, 0..254 for normal FlexRay I-PDUs (all of ComIPduType NORMAL), 0..4294967295 for I-PDUs with ComIPduType TP.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcuIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	

<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00127 :</b>		
<b>Name</b>	ComSignalType		
<b>Description</b>	The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute.  This type could also be used to reserved appropriate storage in AUTOSAR COM.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	BOOLEAN	--	
	FLOAT32	--	
	FLOAT64	--	
	SINT16	--	
	SINT32	--	
	SINT64	--	
	SINT8	--	
	UINT16	--	
	UINT32	--	
	UINT64	--	
	UINT8	--	
	UINT8_DYN	--	
	UINT8_N	--	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00560 :</b>		
<b>Name</b>	ComTransferProperty		
<b>Description</b>	Optionally defines whether this group signal shall contribute to the TRIGGERED_ON_CHANGE transfer property of the signal group. If at least one group signal of a signal group has the "ComTransferProperty" configured all other group signals of that signal group shall have the attribute configured as well.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	PENDING		A change of the value of this group signal shall not be considered in the evaluation of the signal groups ComTransferProperty.
	TRIGGERED_ON_CHANGE		A change of the value of this group signal shall be considered in the evaluation of the signal groups ComTransferProperty.
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE

<b>Duration Class</b>	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00002 :</b>		
<b>Name</b>	ComSystemTemplateSystemSignalRef		
<b>Description</b>	Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	Foreign reference to [ I-SIGNAL-TO-I-PDU-MAPPING ]		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: ECU		

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComFilter	0..1	This container contains the configuration parameters of the AUTOSAR COM module's filters. Note: On sender side the container is used to specify the transmission mode conditions.

### 10.1.16 ComTxMode

<b>SWS Item</b>	<b>ECUC_Com_00351 :</b>	
<b>Container Name</b>	ComTxMode	
<b>Description</b>	This container contains the configuration parameters of the AUTOSAR COM module's transmission modes.	
<b>Configuration Parameters</b>		

<b>SWS Item</b>	<b>ECUC_Com_00137 :</b>	
<b>Name</b>	ComTxModeMode	
<b>Description</b>	The available transmission modes described in [18] shall be extended by the additional mode None.	
	The transmission mode None shall not have any further sub-attributes in the ComTxMode object.	
<b>Multiplicity</b>	1	
<b>Type</b>	EcucEnumerationParamDef	
<b>Range</b>	DIRECT	--
	MIXED	--
	NONE	Literal for TxMode
	PERIODIC	--

<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00281 :</b>		
<b>Name</b>	ComTxModeNumberOfRepetitions		
<b>Description</b>	Defines the number of repetitions for the transmission mode DIRECT and the event driven part of transmission mode MIXED.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00282 :</b>		
<b>Name</b>	ComTxModeRepetitionPeriod		
<b>Description</b>	Defines the repetition period in seconds of the multiple transmissions in case ComTxModeNumberOfRepetitions is configured greater than or equal to 1 and ComTxModeMode is configured to DIRECT or MIXED. In case of the mixed transmission mode only the event driven part is affected.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0 .. 3600		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00180 :</b>		
<b>Name</b>	ComTxModeTimeOffset		
<b>Description</b>	Defines the period in seconds between the start of the I-PDU by Com_IpduGroupControl and the first transmission request in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected.  In case ComTxModeTimeOffset is omitted or configured to 0, the first		

	periodic transmission shall be transmitted within the next invocation of Com_MainFunctionTx.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0 .. 3600		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00178 :</b>		
<b>Name</b>	ComTxModeTimePeriod		
<b>Description</b>	Defines the repetition period in seconds of the periodic transmission requests in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	0 .. 3600		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**

### 10.1.17 ComTxModeTrue

<b>SWS Item</b>	<b>ECUC_Com_00455 :</b>		
<b>Container Name</b>	ComTxModeTrue		
<b>Description</b>	This container contains the configuration parameters of the AUTOSAR COM module's transmission modes in the case the ComFilter evaluates to true.		
<b>Configuration Parameters</b>			

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComTxMode	1	This container contains the configuration parameters of the AUTOSAR COM module's transmission modes.

### 10.1.18 ComTxModeFalse

<b>SWS Item</b>	<b>ECUC_Com_00454 :</b>
<b>Container Name</b>	ComTxModeFalse
<b>Description</b>	This container contains the configuration parameters of the AUTOSAR COM module's transmission modes in the case the ComFilter evaluates to false.
<b>Configuration Parameters</b>	

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComTxMode	1	This container contains the configuration parameters of the AUTOSAR COM module's transmission modes.

### 10.1.19 ComGwMapping

<b>SWS Item</b>	<b>ECUC_Com_00544 :</b>		
<b>Container Name</b>	ComGwMapping		
<b>Description</b>	Each instance of this container defines one mapping of the integrated Signal Gateway.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Configuration Parameters</b>			

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComGwDestination	1..*	Each instance of this choice container allows to define one routing destination either by reference to an already configured COM signal / signal group or by a destination description container.
ComGwSource	1	This choice container allows the definition of the gateway source signal either by reference to an already configured COM signal / signal group or by a source description container.

### 10.1.20 ComGwSource

<b>SWS Item</b>	<b>ECUC_Com_00545 :</b>
<b>Choice container Name</b>	ComGwSource
<b>Description</b>	This choice container allows the definition of the gateway source signal either by reference to an already configured COM signal / signal group or by a source description container.

<b>Container Choices</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>

ComGwSignal	0..1	This container allows specifying a gateway source or destination respectively with a reference to a ComSignal, a ComGroupSignal or a ComSignalGroup.
ComGwSourceDescription	0..1	Description of a gateway source. This container allows defining a gateway source without the configuration of a complete COM signal. This allows adding / changing gateway relations post build without the configuration of new signals.

### 10.1.21 ComGwSourceDescription

<b>SWS Item</b>	<b>ECUC_Com_00548 :</b>		
<b>Container Name</b>	ComGwSourceDescription		
<b>Description</b>	Description of a gateway source. This container allows defining a gateway source without the configuration of a complete COM signal. This allows adding / changing gateway relations post build without the configuration of new signals.		
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_Com_00259 :</b>		
<b>Name</b>	ComBitPosition		
<b>Description</b>	Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00158 :</b>		
<b>Name</b>	ComBitSize		
<b>Description</b>	Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN  the size shall be configured by ComSignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 64		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME

	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00157 :</b>		
<b>Name</b>	ComSignalEndianness		
<b>Description</b>	Defines the endianness of the signal's network representation.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	BIG_ENDIAN	--	
	LITTLE_ENDIAN	--	
	OPAQUE	--	
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00437 :</b>		
<b>Name</b>	ComSignalLength		
<b>Description</b>	Description:  For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored.  Range: 0..8 for normal CAN/ LIN I-PDUs, 0..64 for CAN FD I-PDUs, 0..254 for normal FlexRay I-PDUs (all of ComIPduType NORMAL), 0..4294967295 for I-PDUs with ComIPduType TP.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00127 :</b>		
<b>Name</b>	ComSignalType		
<b>Description</b>	The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute.  This type could also be used to reserved appropriate storage in AUTOSAR COM.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	BOOLEAN	--	



	FLOAT32	--	
	FLOAT64	--	
	SINT16	--	
	SINT32	--	
	SINT64	--	
	SINT8	--	
	UINT16	--	
	UINT32	--	
	UINT64	--	
	UINT8	--	
	UINT8_DYN	--	
	UINT8_N	--	
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00257 :</b>		
<b>Name</b>	ComUpdateBitPosition		
<b>Description</b>	Bit position of update-bit inside I-PDU.  If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side.  Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcuIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00550 :</b>		
<b>Name</b>	ComGwIPduRef		
<b>Description</b>	Reference to an I-PDU of a Signal Gateway source or destination description.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to [ ComIPdu ]		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>			

<b>No Included Containers</b>
-------------------------------

### 10.1.22 ComGwDestination

<b>SWS Item</b>	<b>ECUC_Com_00546 :</b>		
<b>Choice container Name</b>	ComGwDestination		
<b>Description</b>	Each instance of this choice container allows to define one routing destination either by reference to an already configured COM signal / signal group or by a destination description container.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD

<b>Container Choices</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComGwDestinationDescription	0..1	Description of a gateway destination. This container allows defining a gateway destination without the configuration of a complete COM signal. This allows adding / changing gateway relations post build without the configuration of new signals.
ComGwSignal	0..1	This container allows specifying a gateway source or destination respectively with a reference to a ComSignal, a Com-GroupSignal or a ComSignalGroup.

### 10.1.23 ComGwDestinationDescription

<b>SWS Item</b>	<b>ECUC_Com_00549 :</b>		
<b>Container Name</b>	ComGwDestinationDescription		
<b>Description</b>	Description of a gateway destination. This container allows defining a gateway destination without the configuration of a complete COM signal. This allows adding / changing gateway relations post build without the configuration of new signals.		
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_Com_00259 :</b>		
<b>Name</b>	ComBitPosition		
<b>Description</b>	Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD

<b>Scope / Dependency</b>	scope: local
---------------------------	--------------

<b>SWS Item</b>	<b>ECUC_Com_00157 :</b>		
<b>Name</b>	ComSignalEndianness		
<b>Description</b>	Defines the endianness of the signal's network representation.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	BIG_ENDIAN	--	
	LITTLE_ENDIAN	--	
	OPAQUE	--	
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00170 :</b>		
<b>Name</b>	ComSignalInitValue		
<b>Description</b>	<p>Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of UINT8_DYN the initial size shall be 0.</p> <p>In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.</p>		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcucStringParamDef		
<b>Default value</b>	0		
<b>maxLength</b>	--		
<b>minLength</b>	--		
<b>regularExpression</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local dependency: In case of UINT8_N the length of ComSignalInitValue has to be the same as ComSignalLength.		

<b>SWS Item</b>	<b>ECUC Com_00232 :</b>	
<b>Name</b>	ComTransferProperty	
<b>Description</b>	Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.	
<b>Multiplicity</b>	0..1	
<b>Type</b>	EcucEnumerationParamDef	
<b>Range</b>	PENDING	A write access to this signal never triggers the transmission of the corresponding I-PDU.
	TRIGGERED	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU.
	TRIGGERED_ON_CHANGE	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.
	TRIGGERED_ON_CHANGE_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.
	TRIGGERED_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition.
<b>Post-Build Variant Multiplicity</b>	true	
<b>Post-Build Variant Value</b>	true	
<b>Multiplicity Configuration Class</b>	<i>Pre-compile time</i>	X VARIANT-PRE-COMPILE
	<i>Link time</i>	X VARIANT-LINK-TIME
	<i>Post-build time</i>	X VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<i>Pre-compile time</i>	X VARIANT-PRE-COMPILE
	<i>Link time</i>	X VARIANT-LINK-TIME
	<i>Post-build time</i>	X VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local	

<b>SWS Item</b>	<b>ECUC_Com_00257 :</b>		
<b>Name</b>	ComUpdateBitPosition		
<b>Description</b>	Bit position of update-bit inside I-PDU.  If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side.  Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.		
<b>Multiplicity</b>	0..1		
<b>Type</b>	EcuIntegerParamDef		
<b>Range</b>	0 .. 4294967295		
<b>Default value</b>	--		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Post-Build Variant Value</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	<b>ECUC_Com_00550 :</b>		
<b>Name</b>	ComGwIPduRef		
<b>Description</b>	Reference to an I-PDU of a Signal Gateway source or destination description.		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to [ ComIPdu ]		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>			

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
ComFilter	0..1	This container contains the configuration parameters of the AUTOSAR COM module's Filters.  Note: On sender side the container is used to specify the transmission mode conditions.

### 10.1.24 ComGwSignal

<b>SWS Item</b>	<b>ECUC_Com_00551 :</b>		
<b>Container Name</b>	ComGwSignal		
<b>Description</b>	This container allows specifying a gateway source or destination respectively with a reference to a ComSignal, a ComGroupSignal or a ComSignalGroup.		
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>ECUC_Com_00547 :</b>		
-----------------	-------------------------	--	--

<b>Name</b>	ComGwSignalRef		
<b>Description</b>	Reference to an object of a gateway relation. Either to a ComSignal, ComGroupSignal or to a SignalGroup.		
<b>Multiplicity</b>	1		
<b>Type</b>	Choice reference to [ ComGroupSignal , ComSignal , ComSignalGroup ]		
<b>Post-Build Variant Value</b>	true		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	X	VARIANT-LINK-TIME
	<b>Post-build time</b>	X	VARIANT-POST-BUILD
<b>Scope / Dependency</b>			
<b>No Included Containers</b>			

## 10.2 Configuration Rules

### 10.2.1 General Rules

**[SWS\_Com\_00401]** [It is illegal for any two of the following parameters to have the same value:

- shortName of a ComSignal
- shortName of a ComSignalGroup
- shortName of a ComGroupSignal
- shortName of a ComIPdu
- shortName of a ComIPduGroup

] (SRS\_Com\_02067)

**[SWS\_Com\_00732]** [It is illegal for any of the following five parameters:

- ComNotification
- ComErrorNotification
- ComTimeoutNotification
- ComInvalidNotification
- ComIPduCallout

to have the same value as any of the following five parameters

- Com\_<shortName of a ComSignal>
- Com\_<shortName of a ComSignalGroup>
- Com\_<shortName of a ComGroupSignal>
- Com\_<shortName of a ComIPdu>
- Com\_<shortName of a ComIPduGroup>

] (SRS\_Com\_02067)

**[SWS\_Com\_00402]** [It is illegal for any of the following parameters not to be formulated according to C's identifier rules:

- shortName of a ComSignal
- shortName of a ComSignalGroup
- shortName of a ComGroupSignal
- shortName of a ComIPdu
- shortName of a ComIPduGroup
- ComNotification
- ComErrorNotification
- ComTimeoutNotification
- ComInvalidNotification
- ComIPduCallout

] (SRS\_Com\_02067)

### 10.2.2 Signal Configuration

More than one signal can be packed into an I-PDU as long as the following packing rules are fulfilled:

**[SWS\_Com\_00102]** [ComSignal/ ComGroupSignal are not allowed to overlap each other.] (SRS\_Com\_02067)

It is explicitly allowed that a ComSignal/ ComGroupSignal may have the size 0, see ECUC\_Com\_00158.

**[SWS\_Com\_00105]** [ComSignal/ ComGroupSignal of ComSignalType UINT8\_N and UINT8\_DYN shall be byte aligned (ComBitPosition is a multiple of 8).] (SRS\_Com\_02067)

**[SWS\_Com\_00443]** [A ComSignal/ ComGroupSignal of type uint8[n] shall always be mapped to an n-bytes sized ComSignal/ ComGroupSignal.] (SRS\_Com\_02067)

**[SWS\_Com\_00553]** [A ComSignal/ ComGroupSignal of type uint8[n] shall be configured to have OPAQUE endianness.] (SRS\_Com\_02067)

**[SWS\_Com\_00754]** [A dynamic length I-PDU shall contain at most one dynamic length signal.] (SRS\_Com\_02091, SRS\_Com\_02092)

**[SWS\_Com\_00755]** [The dynamic length signal shall be placed last in a dynamic length I-PDU.] (SRS\_Com\_02091, SRS\_Com\_02093)

**[SWS\_Com\_00756]** [The ComSignalLength parameter shall be configured to the maximum size/ length for dynamic length signals.] (SRS\_Com\_02091, SRS\_Com\_02097)

**[SWS\_Com\_00310]** [For I-PDUs with transmission mode *DIRECT* or *MIXED* with ComTxModeNumberOfRepetitions greater or equal 1 no update-bit (ComUpdateBitPosition) shall be configured.] (SRS\_Com\_02067, SRS\_Com\_02030)

**[SWS\_Com\_00785]** [The ComBitSize of a (group) signal shall not extend past the size of its configured ComSignalType.] (SRS\_Com\_02067)

For example, the ComBitSize of a signal with ComSignalType UINT8 shall not exceed 8 bits.

**[SWS\_Com\_00790]** [The configured string of ComSignalInitValues/ ComSignalDataInvalidValue shall be interpreted according to the definitions of [22] for the boolean and all numerical types.] (SRS\_Com\_02067)

**[SWS\_Com\_00859]** [In case a ComSignalDataInvalidValue is configured for a ComSignal that is included in a ComIPdu with ComIPduDirection *RECEIVE*, the ComDataInvalidAction shall also be configured for this ComSignal.] (SRS\_Com\_02079, SRS\_Com\_02087)

### 10.2.3 Signal Group Configuration

**[SWS\_Com\_00365]** [It shall not be allowed to configure signal groups for routing with data type differences between receive and transmit signal group. A sub-setting of signal groups (as defined in SWS\_Com\_00735) shall be supported.] (SRS\_Com\_02067)

How the signals of signal groups are placed within an I-PDU is not restricted. It is allowed to define interlaced signal groups, or to place other signals within the wholes of a signal group.

**[SWS\_Com\_00860]** [In case a ComSignalDataInvalidValue is configured for at least one ComGroupSignal of a ComSignalGroup that is included in a ComIPdu with ComIPduDirection *RECEIVE*, the ComDataInvalidAction shall also be configured for this ComSignalGroup.] (SRS\_Com\_02079, SRS\_Com\_02087)

### 10.2.4 Transmission Mode Configuration

**[SWS\_Com\_00319]** [It shall not be allowed to configure a ComFilter (respectively TMS-conditions) for signals with ComSignalType *FLOAT32* or *FLOAT64*.] (SRS\_Com\_02067)

It is not be allowed to configure a ComFilter (respectively TMS-conditions) that uses floats.

Floats are not allowed to be used in filter conditions. See [17] and SWS\_Com\_00132. Therefore, floats are not allowed for conditions of TMS.

**[SWS\_Com\_00465]** [Every ComTxModeTrue or ComTxModeFalse that is a potential result of the configured/ calculated TMS must be configured. Within the ComTxIPdu, at least one of the containers ComTxModeTrue or ComTxModeFalse has to be included.] (SRS\_Com\_02067)



### 10.2.5 Signal Gateway Configuration

**[SWS\_Com\_00384]** [The ComBitSize of a received and to be routed ComSignal shall not differ.] (SRS\_Com\_02067)

**[SWS\_Com\_00598]** [A signal or signal group contained in an I-PDU with ComIPdu-Type configured as TP shall not be configured as source or destination within the ComGwMapping configuration container.] (SRS\_PduR\_06055)

Additionally to reception deadline monitoring of a ComSignal, to be routed by the Signal Gateway, it is possible to configure update-bits via ComUpdateBitPosition, for the transmit signal. In this case, the receiving node can detect if the sender has really updated the signal or it is just repeated by the Signal Gateway. If this is necessary depends on the use-case.

### 10.2.6 Filter Configuration

**[SWS\_Com\_00535]** [For the ComFilterAlgorithm *ONE\_EVERY\_N*, the ComFilterOffset shall be configured to a value lesser than ComFilterPeriod.] (SRS\_Com\_02067)

### 10.2.7 Post Build Configuration

**[SWS\_Com\_00373]** [The post-build time configuration part (post-compile and post-link time) can only be updated when it is not in use.] (SRS\_Com\_02067, SRS\_PduR\_06002)

**[SWS\_Com\_00487]** [The complete post-build time configurable configuration shall be identifiable by a unique identifier.] (SRS\_PduR\_06097)

### 10.2.8 Dynamic Length I-PDUs

**[SWS\_Com\_00817]** [I-PDUs including metadata shall not be configured as dynamic length I-PDUs.] (SRS\_Com\_02067)

The above restriction is necessary, since the length of the dynamic length signal could not be derived correctly, when the I-PDU contains additional metadata.

Since the FlexRay Interface does only support even values for the I-PDU length, it is not recommend to configure dynamic length I-PDU to be send via Frlf. Instead, the FlexRay TP module should be used to transport dynamic length I-PDUs via FlexRay.

### 10.2.9 Replicated I-PDUs

**[SWS\_Com\_00834]** [If an I-PDU is configured for replication (includes a ComIPduReplication), this I-PDU shall also have configured an I-PDU counter (ComIPduCounter) with a counter threshold (ComIPduCounterThreshold) set to 0.] (SRS\_Com\_02102)

For further information on this constraint see 7.10.

## 11 Not Applicable Requirements

**[SWS\_Com\_00999]** [These requirements are not applicable to this specification.]  
(SRS\_BSW\_00171, SRS\_BSW\_00170, SRS\_BSW\_00383, SRS\_BSW\_00375,  
SRS\_BSW\_00416, SRS\_BSW\_00437, SRS\_BSW\_00168, SRS\_BSW\_00423,  
SRS\_BSW\_00424, SRS\_BSW\_00426, SRS\_BSW\_00427, SRS\_BSW\_00428,  
SRS\_BSW\_00429, SRS\_BSW\_00433, SRS\_BSW\_00417, SRS\_BSW\_00409,  
SRS\_BSW\_00386, SRS\_BSW\_00161, SRS\_BSW\_00162, SRS\_BSW\_00005,  
SRS\_BSW\_00164, SRS\_BSW\_00325, SRS\_BSW\_00413, SRS\_BSW\_00347,  
SRS\_BSW\_00307, SRS\_BSW\_00410, SRS\_BSW\_00314, SRS\_BSW\_00353,  
SRS\_BSW\_00361, SRS\_BSW\_00302, SRS\_BSW\_00328, SRS\_BSW\_00006,  
SRS\_BSW\_00378, SRS\_BSW\_00306, SRS\_BSW\_00308, SRS\_BSW\_00309,  
SRS\_BSW\_00009, SRS\_BSW\_00010, SRS\_BSW\_00333, SRS\_BSW\_00321,  
SRS\_BSW\_00341, SRS\_BSW\_00334)

## 12 Appendix A

This appendix contains use cases with different transmission modes and the necessary configuration for these. Table 11 shows the notation of the use case diagrams.




Notation	Description
$t_c, t_{c1}, t_{c2}$	cycle times
$t_d$	cycle time of N-Times sent signals
$t_r$	minimum SW reaction time of COM-Layer due to internal main cycle time
V	Value: x stands for an arbitrary value/ value range, a..w for specific values / value ranges, defined by the user, with $a <> b$ , range a is disjoint from range b.
	request from RTE to the COM-Layer
	request from COM-Layer to PDU Router
	potential but skipped request from COM-Layer to PDU Router (e.g. because of a new send request by the RTE or delayed due to minimum delay time)
dt	minimum delay time: minimum distance between two requests to PDU Router
without TMS switch	without switching of the TMS (see 7.3.3.2) from true to false or vice versa
with TMS switch	with switching of the TMS (see 7.3.3.2) from true to false or vice versa (from TM 1 to TM 2); one TM is named before the "+" and one behind in the description

Table 11: Legend for use case diagrams.

**Use case 1** shows an I-PDU that is sent out cyclically with a cycle time  $t_c$ . This I-PDU consists of signals that all have the ComTransferProperty *PENDING*. It is configured that the transmission takes place when the TMS evaluates to true.

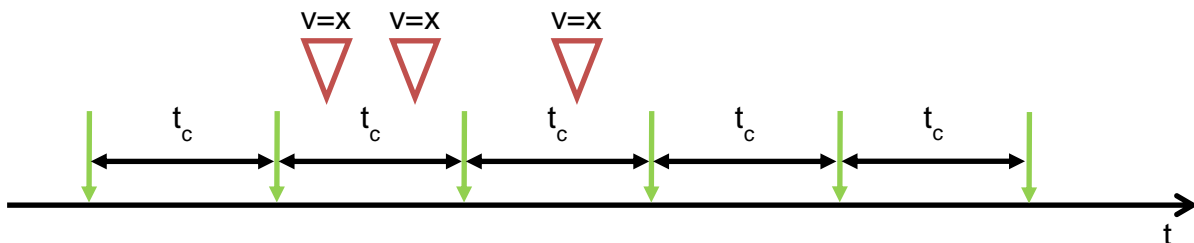


Figure 12: Use case 1, TM periodic (without TMS switch, see Chapter 7.3.3.2)

<b>ComSignal</b>	
<b>ComTransferProperty</b>	PENDING or TRIGGERED (TRIGGERED has no influence)
<b>ComFilter</b>	

<b>ComFilterAlgorithm</b>	ALWAYS
<b>ComIPdu</b>	
<b>ComIPduDirection</b>	SEND
<b>ComTxModeTrue</b>	
<b>ComTxModeTimePeriod</b>	tc
<b>ComTxModeMode</b>	PERIODIC
<b>ComTxIPdu</b>	
<b>ComMinimumDelayTime</b>	0 or omitted

**Table 12: Relevant I-PDU transmission configuration for use case 1**

Because of the configuration of the parameter *ComFilterAlgorithm* *ALWAYS* of the *ComFilter*, there is no need to configure a transmission mode for the case that the TMS evaluates to false.

It does not make any difference in the behavior whether the *ComFilterAlgorithm* parameter of the *ComFilter* is set for all the signals within the I-PDU to *ALWAYS* or if the *ComFilter* is not defined (does not contribute to the evaluation of the TMS), see SWS\_Com\_00255.

**Use case 2** shows an I-PDU which is sent out three times whenever a value is given by the upper (Com\_SendSignal or Com\_SendSignalGroup). The time between two transmissions is  $t_d$ . This I-PDU consists of signals, which all have the ComTransfer-Property *TRIGGERED*. It is configured that the transmission takes place when the TMS evaluates to true.

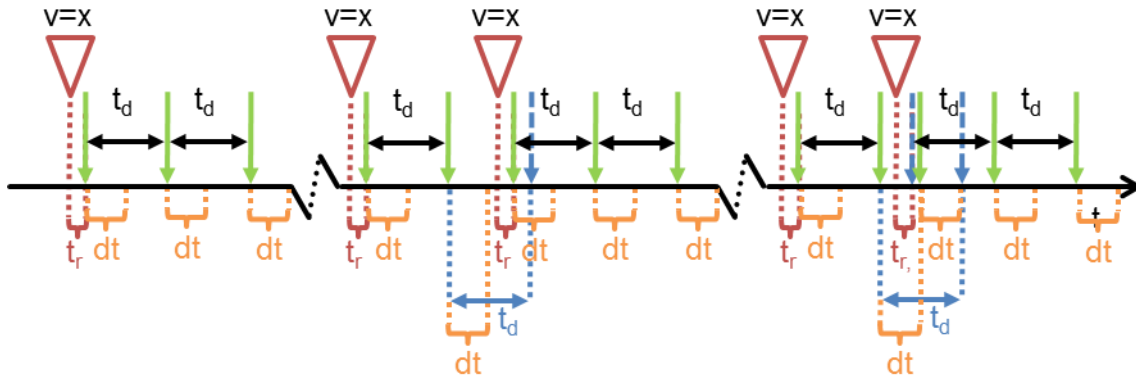


Figure 13: Use case 2, TM DIRECT (N-Times) here n = 2 (without TMS switch)

<b>ComGeneral</b>	
<i>ComEnableMDTForCyclicTransmission</i>	true
<b>ComSignal</b>	
<i>ComTransferProperty</i>	TRIGGERED
<i>ComFilter</i>	
<i>ComFilterAlgorithm</i>	ALWAYS
<b>ComIPdu</b>	
<i>ComIPduDirection</i>	SEND
<i>ComTxModeTrue</i>	
<i>ComTxModeMode</i>	DIRECT
<i>ComTxModeNumberOfRepetitions</i>	2
<i>ComTxModeRepetitionPeriod</i>	$t_d$
<b>ComTxIPdu</b>	
<i>ComMinimumDelayTime</i>	dt

Table 13: Relevant I-PDU transmission configuration for use case 2

If there is a new send request by the RTE before the ongoing repetition cycle is completed, the new repetition cycle is started and the rest of the last one is discarded.

**Use case 3** shows an I-PDU which is send out cyclically with a cycle time  $t_{c1}$  if value  $v = a$  (TMS evaluates to true) and with a cycle time  $t_{c2}$  if value  $v = b$  (TMS evaluates to false). The I-PDU consists of signals, which all have the ComTransferProperty *PENDING*.

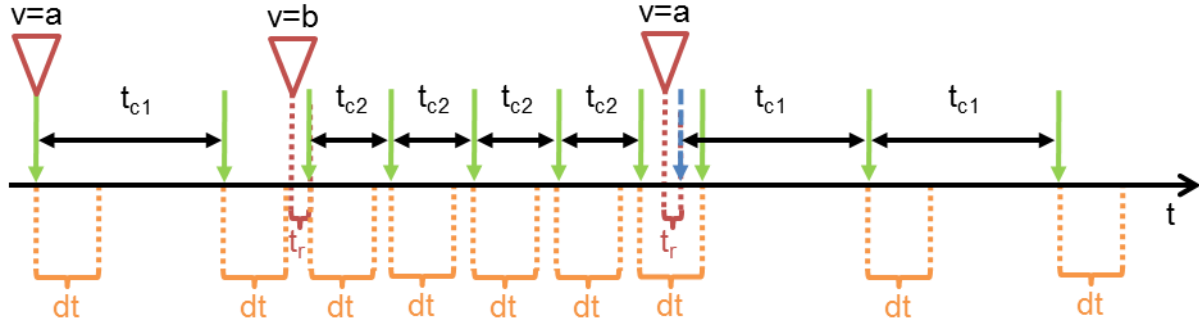


Figure 14: Use case 3, TM periodic + periodic (with TMS switch)

<b>ComGeneral</b>	
<b>ComEnableMDTForCyclicTransmission</b>	true
<b>ComSignal</b>	
<b>ComTransferProperty</b>	PENDING or TRIGGERED (TRIGGERED has no influence)
<b>ComFilter</b>	
<b>ComFilterAlgorithm</b>	any filter mapping value a to true and value b to false
<b>ComIPdu</b>	
<b>ComIPduDirection</b>	SEND
<b>ComTxModeTrue</b>	
<b>ComTxModeMode</b>	PERIODIC
<b>ComTxModeTimePeriod</b>	$t_{c1}$
<b>ComTxModeFalse</b>	
<b>ComTxModeMode</b>	PERIODIC
<b>ComTxModeTimePeriod</b>	$t_{c2}$
<b>ComTxIPdu</b>	
<b>ComMinimumDelayTime</b>	dt

Table 14: Relevant I-PDU transmission configuration for use case 3

The TMS switch caused by the new value  $v = b$ , starts a new cycle within the next main function and the new value is sent out. Nevertheless, the minimum delay time  $dt$  has to be taken into account, since ComEnableMDTForCyclicTransmission is set to true.

For the parameter ComFilterAlgorithm, the following OSEK COM defined filters can be used for this use-case:

- MASKED\_NEW\_EQUALS\_X
- MASKED\_NEW\_DIFFERS\_X
- MASKED\_NEW\_DIFFERS\_MASKED\_OLD
- NEW\_IS\_WITHIN
- NEW\_IS\_OUTSIDE

**Use case 4** shows an I-PDU which is send out cyclically with a cycle time  $t_c$  if value  $v = a$  (TMS evaluates to true) and if value  $v = b$  (TMS evaluates to false) it is sent out three times whenever the value is given by the RTE. The time between two transmissions is  $t_d$ . The I-PDU consists of signals that all have the ComTransferProperty *TRIGGERED*.

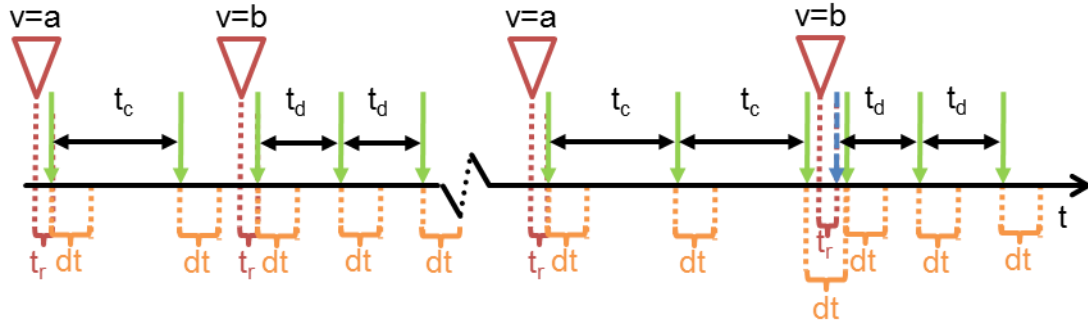


Figure 15: Use case 4 scenario a, TM periodic + *DIRECT/N*-Times, here  $n = 2$  (with TMS switch)

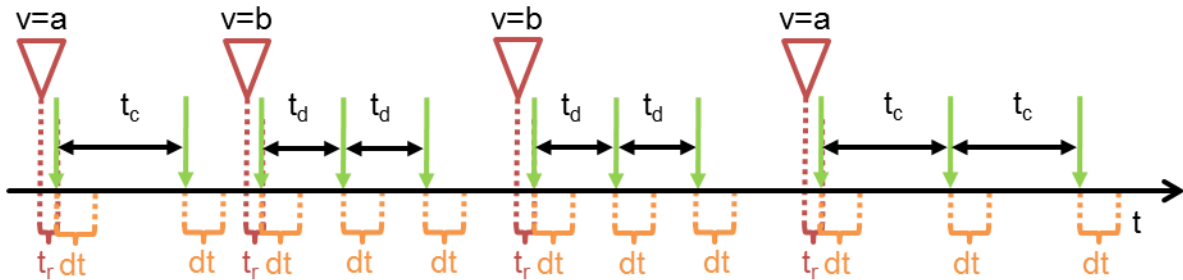


Figure 16: Use case 4 scenario b, TM periodic + *DIRECT/N*-Times, here  $n = 2$  (with TMS switch)

<b>ComGeneral</b>	
<i>ComEnableMDTForCyclicTransmission</i>	True
<b>ComSignal</b>	
<i>ComTransferProperty</i>	TRIGGERED
<i>ComFilter</i>	
<i>ComFilterAlgorithm</i>	any filter mapping value a to true and value b to false
<b>ComIPdu</b>	
<i>ComIPduDirection</i>	SEND
<i>ComTxModeTrue</i>	
<i>ComTxModeMode</i>	PERIODIC
<i>ComTxModeTimePeriod</i>	$t_c$
<i>ComTxModeFalse</i>	
<i>ComTxModeMode</i>	DIRECT
<i>ComTxModeNumberOfRepetitions</i>	2
<i>ComTxModeRepetitionPeriod</i>	$t_d$
<b>ComTxIPdu</b>	
<i>ComMinimumDelayTime</i>	dt

Table 15: Relevant I-PDU transmission configuration for use case 4



After the switch from ComTxModeMode *DIRECT* to *PERIODIC* the cycle is started within the next main function and the new value *a* is sent out with respect to ComMinimumDelayTime *dt*.

**Use case 5** shows an I-PDU which is send out cyclically with a cycle time  $t_c$  and if the value (the same or a new one) is given by the RTE it is also sent out directly three times. The time between two of these three transmissions is always  $t_d$ . The I-PDU consists of signals that all have the ComTransferProperty *TRIGGERED*.

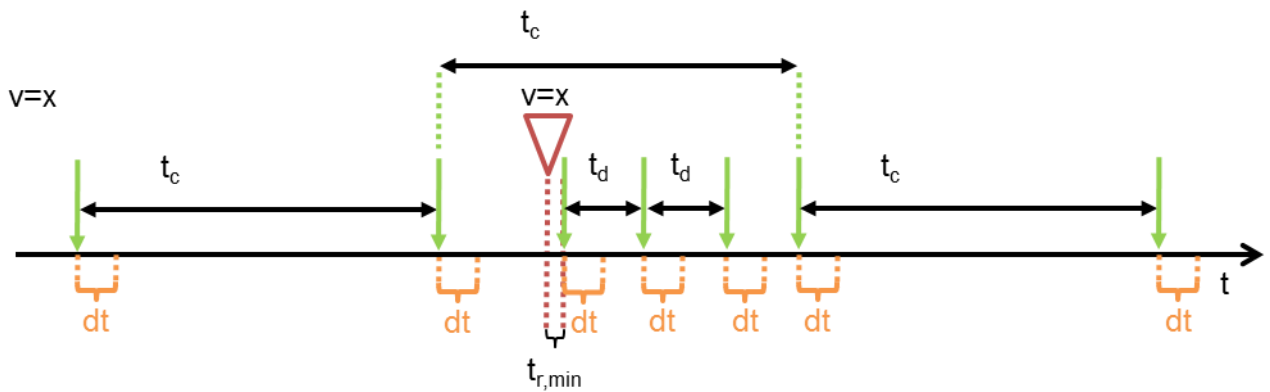


Figure 17: Use case 5 scenario a, TM *MIXED*, here  $n = 2$  (without TMS switch)

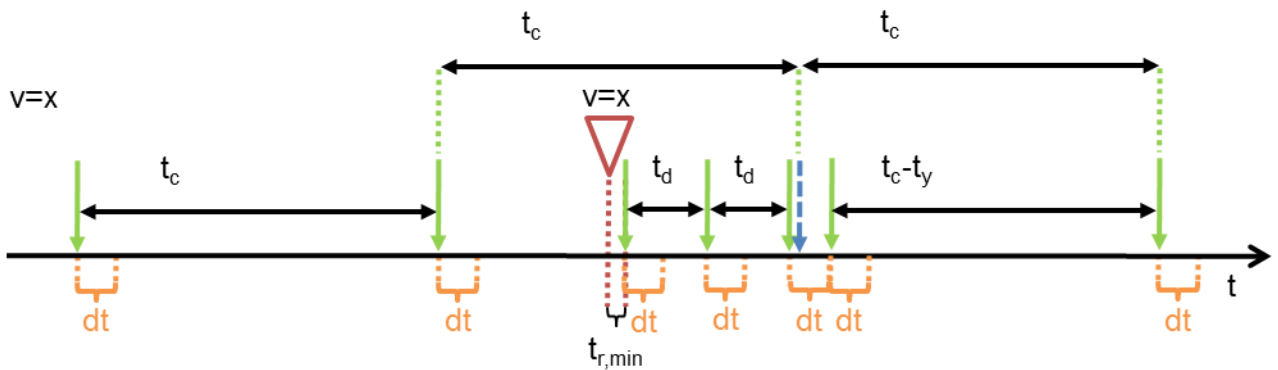


Figure 18: Use case 5 scenario b, TM *MIXED*, here  $n = 2$  (without TMS switch)

<b>ComGeneral</b>	
<i>ComEnableMDTForCyclicTransmission</i>	true
<b>ComSignal</b>	
<i>ComTransferProperty</i>	TRIGGERED
<i>ComFilter</i>	
<i>ComFilterAlgorithm</i>	ALWAYS
<b>ComIPdu</b>	
<i>ComIPduDirection</i>	SEND
<i>ComTxModeTrue</i>	
<i>ComTxModeMode</i>	MIXED
<i>ComTxModeTimePeriod</i>	$t_c$

<b>ComTxModeNumberOfRepetitions</b>	2
<b>ComTxModeRepetitionPeriod</b>	$t_d$
<b>ComTxIPdu</b>	
<b>ComMinimumDelayTime</b>	$dt$

Table 16: Relevant I-PDU transmission configuration for use case 5

If the next transmission caused by the periodic part of the ComTxModeMode *MIXED* should take place within the timeout  $dt$  (ComMinimumDelayTime) after a transmission of the *DIRECT* (N-Times) part, this sent out is delayed until the minimum delay time is elapsed. However, after that the next period of the periodic part is shortened so that there is only an intermediate phase shift of the periodic part but no continuous one.

**Use case 6** shows an I-PDU which is send out cyclically with a cycle time  $t_{c2}$  if value  $v = b$  (TMS evaluates to false). If value  $v = a$  (TMS evaluates to true) it is sent out cyclically with a cycle time  $t_{c1}$  and whenever the value  $v = a$  is given by the RTE it is also sent out directly three times. The time between two of these three transmissions is always  $t_d$ . The I-PDU consists of signals that all have the ComTransferProperty *TRIGGERED*.

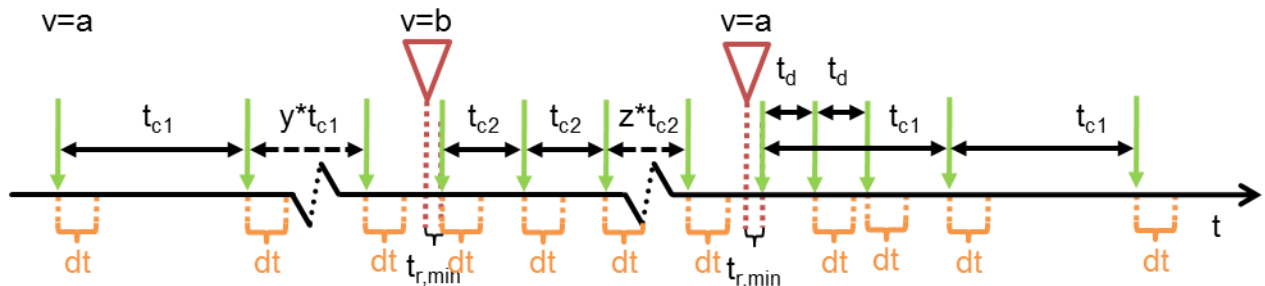


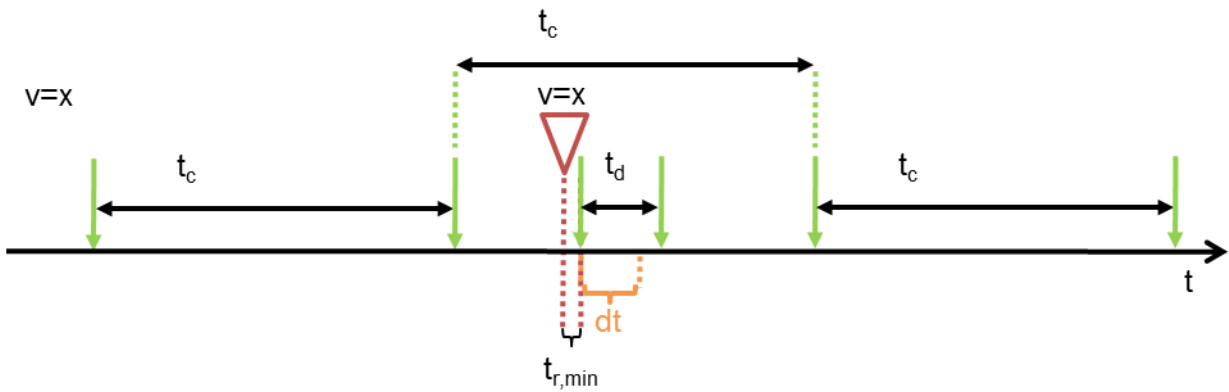
Figure 19: Use Case 6, TM *MIXED*, here  $n = 2 + \text{periodic (with TMS switch)}$

<b>ComGeneral</b>	
<b>ComEnableMDTForCyclicTransmission</b>	true
<b>ComSignal</b>	
<b>ComTransferProperty</b>	TRIGGERED
<b>ComFilter</b>	
<b>ComFilterAlgorithm</b>	any filter mapping value a to true and value b to false
<b>ComIPdu</b>	
<b>ComIPduDirection</b>	SEND
<b>ComTxModeTrue</b>	
<b>ComTxModeMode</b>	MIXED
<b>ComTxModeTimePeriod</b>	$t_{c1}$
<b>ComTxModeNumberOfRepetitions</b>	2
<b>ComTxModeRepetitionPeriod</b>	$t_d$
<b>ComTxModeFalse</b>	
<b>ComTxModeMode</b>	PERIODIC
<b>ComTxModeTimePeriod</b>	$t_{c2}$
<b>ComTxIPdu</b>	
<b>ComMinimumDelayTime</b>	$dt$

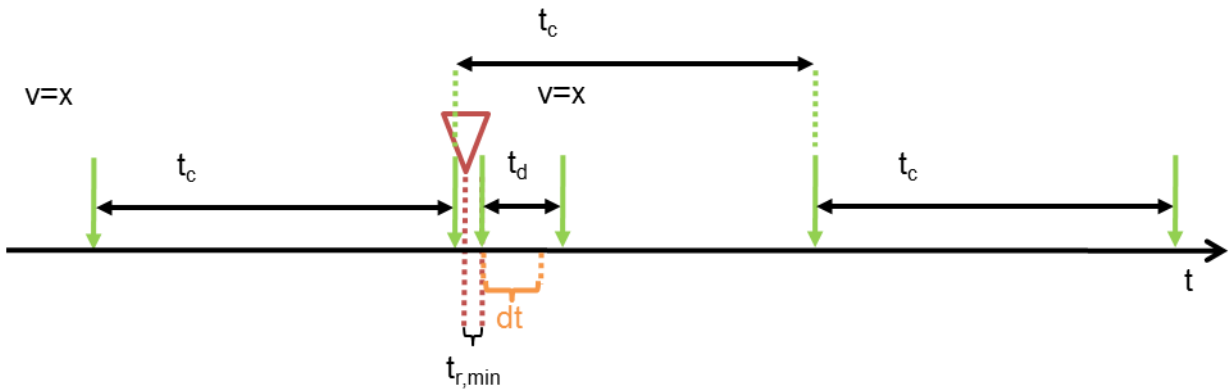
**Table 17: Relevant I-PDU transmission configuration for use case 6**

A usage of this in practice is for example the signal of the button that controls the window-lift motor. If the button is not pressed, there is a long cycle time  $t_{c1}$  with this information. If it is pressed this information is distributed with a short cycle time  $t_{c2}$ . If the button is released again, starting with the next main function this information is distributed three times with  $t_d$  and after that, again the long cycle time is used.

**Use case 7** is similar to use case 5 but with ComEnableMDTForCyclicTransmission disabled and ComTxModeNumberOfRepetitions set to one.



**Figure 20: Use case 7 scenario a, TM MIXED, here  $n = 1$  (without TMS switch)**



**Figure 21: Use case 7 scenario b, TM MIXED, here  $n = 1$  (without TMS switch)**

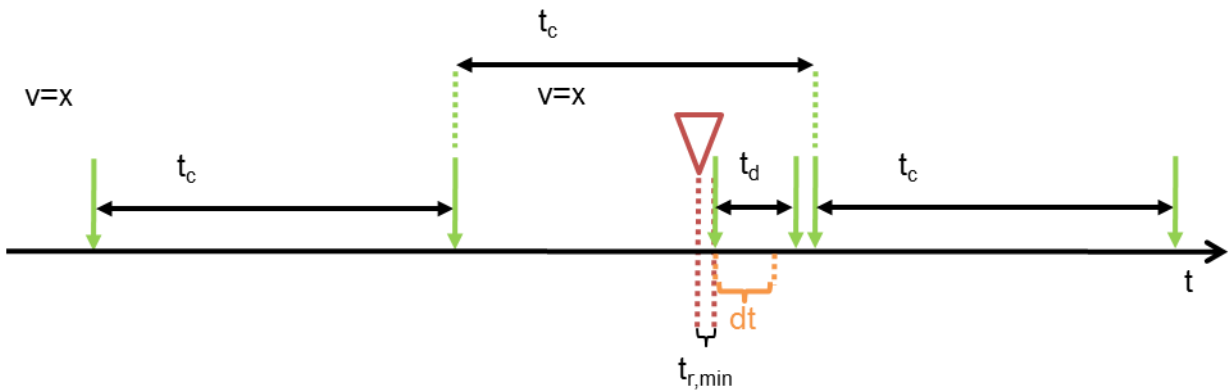


Figure 22: Use case 7 scenario c, TM MIXED, here n = 1 (without TMS switch)

<b>ComGeneral</b>	
<i>ComEnableMDTForCyclicTransmission</i>	False
<b>ComSignal</b>	
<i>ComTransferProperty</i>	TRIGGERED
<b>ComFilter</b>	
<i>ComFilterAlgorithm</i>	ALWAYS
<b>ComIPdu</b>	
<i>ComIPduDirection</i>	SEND
<i>ComTxModeTrue</i>	
<i>ComTxModeMode</i>	MIXED
<i>ComTxModeTimePeriod</i>	$t_c$
<i>ComTxModeNumberOfRepetitions</i>	1
<i>ComTxModeRepetitionPeriod</i>	$t_d$
<b>ComTxIPdu</b>	
<i>ComMinimumDelayTime</i>	dt

Table 18: Relevant I-PDU transmission configuration for use case 7

In contrary to use case 5, here the minimum delay timer is neither started for the periodic transmissions of the cyclic part nor for the repeated transmissions of the direct part of the Mixed mode. This can result in two consecutive transmissions within the timespan  $dt$  as shown in the scenarios b and c.