| **Document Title** | Specification of Communication |
|---|---|
| **Document Owner** | AUTOSAR |
| **Document Responsibility** | AUTOSAR |
| **Document Identification No** | 015 |
| **Document Classification** | Standard |

| **Document Version** | 3.2.0 |
|---|---|
| **Document Status** | Final |
| **Part of Release** | 3.0 |
| **Revision** | 7 |

## Document Change History

| Date | Version | Changed by | Change Description |
|---|---|---|---|
| 20.09.2010 | 3.2.0 | AUTOSAR Administration | • Added COM572, COM573, COM568, COM569, COM570, COM571, COM574, COM575<br>• Updated COM001, COM314, COM391, COM501, COM100, COM287, COM123, COM001, COM187, COM184<br>• Legal Disclaimer revised |
| 28.01.2010 | 3.1.0 | AUTOSAR Administration | • Added COM558, COM559.<br>• Updated configuration container, due to missing literals in ComGwSourceDescription and ComGwDestinationDescription.<br>• Turned COM385 into a note.<br>• COM_NETWORK_SIGNAL_NAME removed from COM401<br>• Tables were wrongly stating that Com_ReceiveShadowSignal should return COM_SERVICE_NOT_AVAILABLE.<br>• Updated all configuration containers with correctly generated artifacts.<br>• Legal disclaimer revised |
| 22.01.2008 | 3.0.1 | AUTOSAR Administration | Remove Figure 19, which was included twice |

| | | | |
|---|---|---|---|
| 30.11.2007 | 3.0.0 | AUTOSAR Administration | • Removal of Internal Communication feature<br>• Revision of COM configuration<br>• Revision of Signal Gateway configuration<br>• Enhanced maximal I-PDU length for FlexRay<br>• Harmonization of callback-interface to RTE<br>• Clarifications and bug fixes<br>• Document meta information extended<br>• Small layout adaptations made |
| 30.01.2007 | 2.1.0 | AUTOSAR Administration | • 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.<br>• “Advice for users” revised<br>• “Revision Information” added<br>• Legal disclaimer revised |
| 21.04.2006 | 2.0.0 | AUTOSAR Administration | • Document structure adapted to common Release 2.0 SWS Template.<br>• Integration of signal gateway<br>• Major updates in configuration, error handling, filtering, transmission mode switches, callouts, update-bits, deadline monitoring and initialization |
| 31.05.2005 | 1.0.0 | AUTOSAR Administration | Initial release |

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

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

# Table of Contents

# 1 Introduction and functional overview

This specification is the AUTOSAR COM 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 COM module is placed between RTE and the PDU Router, see [1].

AUTOSAR COM is derived from [17]. For details see Chapter 7.2.1.
AUTOSAR COM provides signal gateway functionality. For details see Chapter 7.2.4.

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

# 2 Acronyms, abbreviations and definitions

## 2.1 Acronyms and abbreviations

| *Acronym:* | *Description:* |
|---|---|
| AUTOSAR COM | AUTOSAR COM is derived from OSEK COM [17]. For details see Chapter 7.2.1. |
| DM | Deadline Monitoring. For details see Chapter 7.4.5.4. |
| IL | Interaction Layer. A detailed description can be found in [17]. |
| I-PDU | Interaction Layer Protocol Data Unit<br>An I-PDU carries signals and is defined in [17]. |
| LOM | Listen only mode |
| L-PDU | Data Link Layer PDU. In AUTOSAR the Data Link Layer is equivalent to the Communication Hardware Abstraction and Microcontroller Abstraction Layer. |
| MDT | Minimum Delay Timer. A detailed description can be found in [17]. See also chapter 7.4.5.4. |
| OSEK COM | Open systems and the corresponding interfaces for automotive electronics – communication [17] |
| PCI | Protocol Control Information<br>For a description see [1] page 04-51 ff. |
| PDU Router | Module that transfers I-PDUs from one module to another module. The PDU Router can be utilized for gateway operations and for internal routing purposes. |
| SDT | Specification of System Template [12] |
| SDU | Service Data Unit<br>For a description see [1] Chapter 4. |
| TM | Transmission Mode |
| TMC | Transmission Mode Condition, see Chapter 7.4.3.3 |
| TMS | Transmission Mode Selector, see Chapter 7.4.3.3 |

### 2.1.1 Definitions

| ***Acronym:*** | ***Description:*** |
|---|---|
| Confirmation | Lower layer reports that a request by COM has been completed successfully. It's 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 to indicate that the sender side AUTOSAR Software Component is not able to provide a valid value. |
| Group signal | A group signal is a signal that is contained in a signal group. |
| Indication | Asynchronous information from lower layer to COM, e.g. when something has been received. |
| Init Value | I-PDUs and signals are set to the Init Value by AUTOSAR COM after start-up. This value is used until it is overwritten. |
| I-PDU group | An I-PDU group contains zero or more I-PDUs or I-PDU groups. For a detailed description see [7]. |
| Inter-ECU communication | Communication between two or more ECU for example via a CAN network. |
| Intra-ECU communication | Communication between Software components that reside on the same ECU. |
| Message | OSEK-COM uses always the synonym *message*. In AUTOSAR, *message* is replaced by *signal* but with the same meaning. |
| Notification | Information by COM to upper layer, e.g. when new data is available, an error occurred. |
| Signal | A description can be found in [7]. |
| Signal groups | 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. To ensure data consistency a complex data type must be treated as an atomic unit.<br>The RTE decomposes the complex data type in single signals and sends them to the AUTOSAR COM module. As these signals altogether still have to be treated as atomic, they are called *signal group*.<br>See also [7]. |
| Update-bits | A mechanism supported by AUTOSAR COM with that the receiver of a signal/ signal group can identify whether the sender has updated the data in this signal/ signal group before sending. See Chapter 7.7. |

# 3 Related documentation

## 3.1 Deliverables of AUTOSAR

[1] Layered Software Architecture
AUTOSAR_LayeredSoftwareArchitecture.pdf

[2] Specification of Communication Stack Types
AUTOSAR_SWS_ComStackTypes.pdf

[3] General Requirements on Basic Software Modules
AUTOSAR_SRS_General.pdf

[4] Basic Software UML Model
AUTOSAR_UML_Model.eap

[5] Specification of Standard Types
AUTOSAR_SWS_StandardTypes.pdf

[6] Specification of the Virtual Functional Bus
AUTOSAR_VirtualFunctionBus.pdf

[7] Requirements on Communication
AUTOSAR_SRS_COM.pdf

[8] Software Component Template
AUTOSAR_SoftwareComponentTemplate.pdf

[9] Requirements on Gateway
AUTOSAR_SRS_Gateway.pdf

[10] Specification of PDU Router
AUTOSAR_SWS_PDU_Router.pdf

[11] Specification of Operating System
AUTOSAR_SWS_OS.pdf

[12] Specification of System Template
AUTOSAR_SystemTemplate.pdf

[13] Specification of RTE Software
AUTOSAR_SWS_RTE.pdf

[14] Specification of ECU Configuration
AUTOSAR_ECU_Configuration.pdf

[15] Specification of Generic Network Management
AUTOSAR_SWS_Generic_NM.pdf

[16] Specification of Communication Manager
AUTOSAR_SWS_ComManager.pdf

AUTOSAR Basic Software Module Description Template,
AUTOSAR_BSW_Module_Description.pdf

## 3.2 Related standards and norms

[17] OSEK/ VDX Communication Version 3.0.3
http://www.osek-vdx.org
OSEKCOM303.pdf

[18] OSEK implementation language Version 2.5
http://www.osek-vdx.org
oil25.pdf

# 4 Constraints and assumptions

This document is applicable for AUTOSAR release 3.0. Features for later AUTOSAR releases are not yet included.

## 4.1 Limitations

AUTOSAR COM is based on [17]. Nevertheless not all features of [17] are included and some features are different. See COM013 for a list of not included features.

Within the AUTOSAR COM Stack the I-PDUs of COM are passed via the PDU Router directly to the communication interfaces. Therefore the maximum length of an I-PDU depends of the maximum length of the L-PDU of the underlying communication interface. For CAN and LIN the maximum L-PDU length is 8 bytes. For FlexRay the maximum L-PDU length is 254 bytes.

For AUTOSAR release 3.0 there is no special treatment in COM to handle *floating point* data types. However, they may be handled with existing COM functionality.

The number of IPDU-groups is limited to 64, see COM187.

## 4.2 Applicability to car domains

No restrictions.

# 5 Dependencies to other modules

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

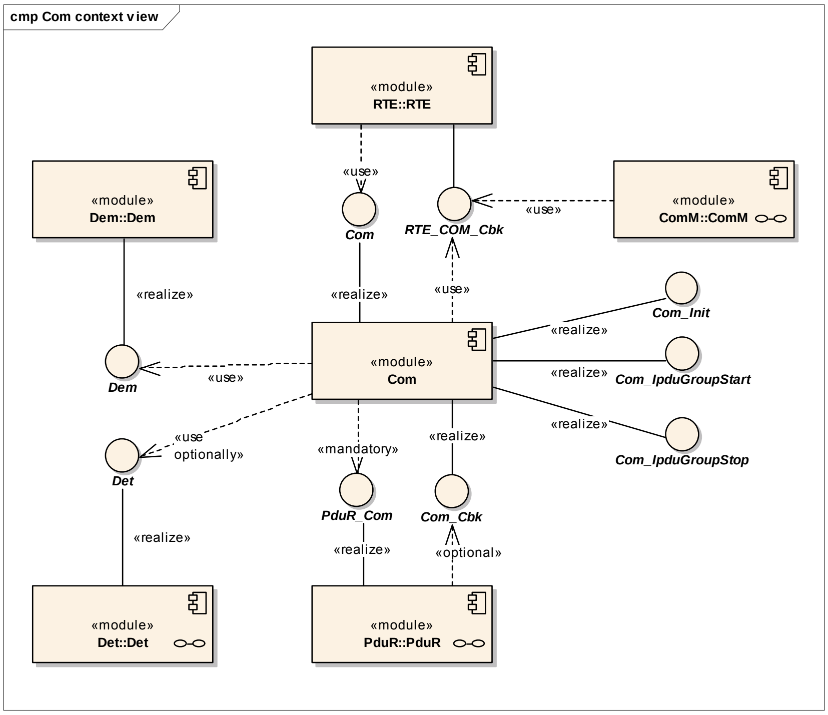

Figure 1: AUTOSAR COM context view

**Note:** RTE_COM_Cbk denotes the interface provided by the RTE that is used by COM to notify about transmission acknowledgements and errors. This must not necessarily result in an RTE_COM_Cbk.h file.

## 5.1 PDU router

The following summarizes the functionality AUTOSAR COM 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 send by the communication controller.
- Trigger interface to enable the PDU router to cause a transmission from AUTOSAR COM

A detailed description of the interface to the lower layer can be found in Chapter 7.6 and in Chapter 9.1. For further information see [10].

## 5.2 Runtime Environment (RTE)

RTE uses the capability to send and receive signals via AUTOSAR COM. In AUTOSAR the RTE is the higher layer above COM. For further information see [13].

## 5.3 COM Manager (ComM)

The COM Manager controls the start and stop of sending and receiving I-PDUs via AUTOSAR COM. For further information see [16].

## 5.4 File structure

### 5.4.1 Code file structure

The code file structure is not completely defined within this specification.

**COM430:** The code-file structure shall include the following files named:

- COM.c – module source file
- COM_Cfg.c – pre-compile-time configurable constant parameters
- COM_Lcfg.c – link-time configurable parameters
- COM_PBcfg.c – post-build time configurable parameters

### 5.4.2 Header file structure

**COM005:** AUTOSAR COM shall offer a header file Com.h which includes all user relevant information and another header file Com_Cbk.h which declares the callback functions and the I-PDU callout prototype.

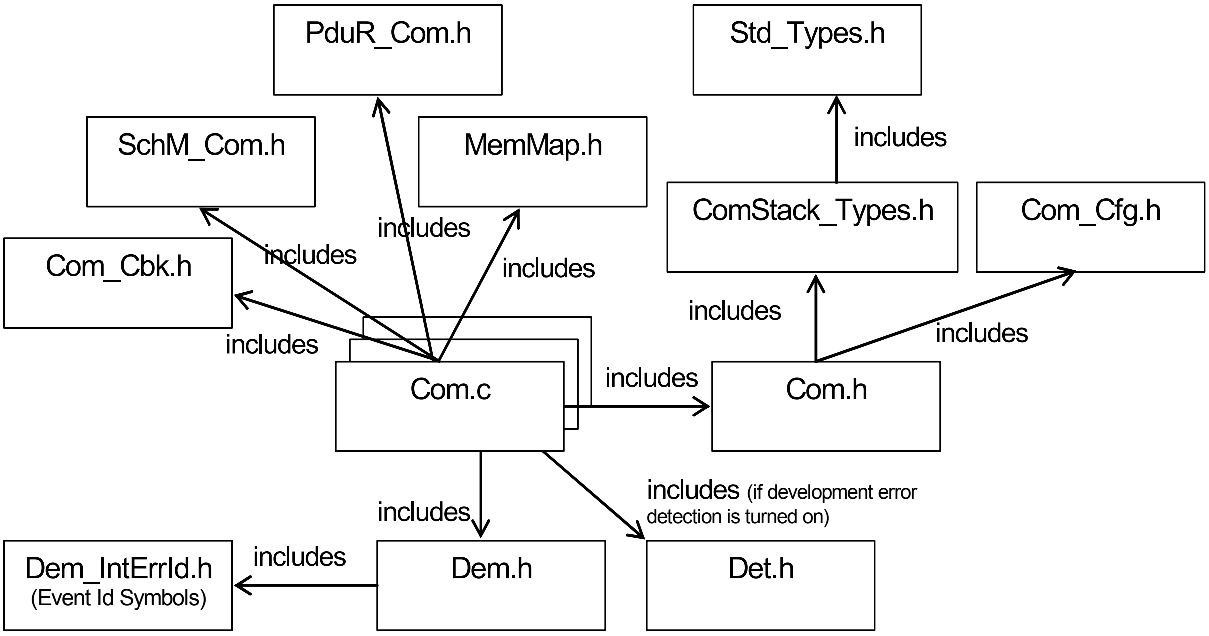

Figure 2: Include file structure

**COM220**: The include file structure regarding the specifics of the COM shall be constructed as shown in **Figure 2**.

**COM431:** The module shall include the Dem.h file. By this inclusion, the APIs to report errors as well as the required Event ID symbols are included. This specification defines the name of the Event ID symbols, which are provided by XML to the DEM configuration tool. The DEM configuration tool assigns ECU dependent values to the Event ID symbols and publishes the symbols in Dem_IntErrId.h.

# 6 Requirements traceability

Document: General Requirements on Basic Software Modules [3]

| *Requirement* | *Satisfied by* |
|---|---|
| [BSW00344]<br>Reference to link-time configuration | Chapter 10.2.1.2 COM432, COM430 |
| [BSW00404]<br>Reference to post build time configuration | Chapter 10.2, COM375, COM432 |
| [BSW00405]<br>Reference to multiple configuration sets | COM432 |
| [BSW00345]<br>Pre-compile-time configuration | Chapter 10.2.1.1 COM430 |
| [BSW159]<br>Tool-based configuration | not applicable<br>(not in scope of this spec) |
| [BSW167]<br>Static configuration checking | not applicable<br>(not scope of this spec) |
| [BSW171]<br>Configurability of optional functionality | not applicable<br>(COM module has no features switches for optional functionality) |
| [BSW170]<br>Data for reconfiguration of AUTOSAR SW-Components | not applicable<br>(not in scope of this spec) |
| [BSW00380]<br>Separate C-files for configuration parameters | COM430 |
| [BSW00419]<br>Separate C-files for pre-compile time □nstanceltion parameters | COM430 |
| [BSW00381]<br>Separate configuration header file for pre-compile time parameters | COM220 |
| [BSW00412]<br>Separate H-file for configuration parameters | COM220 |
| [BSW00383]<br>List dependencies of configuration files | not applicable<br>(implementation specific) |
| [BSW00384]<br>List dependencies to other modules | Chapter 5 |
| [BSW00387]<br>Specify the configuration class of callback function | Chapter 8.4 |
| [BSW00388]<br>Introduce containers | Chapter 10.2, COM541 |
| [BSW00389]<br>Containers shall have names | Chapter 10.2 |
| [BSW00390]<br>Parameter content shall be unique within the module | Chapter 10.2 |
| [BSW00391]<br>Parameter shall have unique names | Chapter 10.2 |
| [BSW00392]<br>Parameters shall have a type | Chapter 10.2 |
| [BSW00393]<br>Parameters shall have a range | Chapter 10.2 |
| [BSW00394]<br>Specify the scope of the parameters | Chapter 10.2 |
| [BSW00395] | Chapter 10 |

| ***Requirement*** | ***Satisfied by*** |
|---|---|
| List the required parameters (per parameter) | (scope and dependency fields) |
| [BSW00396]<br>Configuration classes | Chapter 10.2 |
| [BSW00397]<br>Pre-compile-time parameters | Chapter 10.2 |
| [BSW00398]<br>Link-time parameters | Chapter 10.2 |
| [BSW00399]<br>Loadable post-build time parameters | Chapter 10.2 |
| [BSW00400]<br>Selectable post-build time parameters | Chapter 10.2 |
| [BSW00402]<br>Published information | Chapter 10.2.21 |
| [BSW00375]<br>Notification of wake-up reason | not applicable<br>(not in scope of this spec) |
| [BSW101]<br>Initialization interface | COM128, COM328, COM015, COM098, COM117, COM170, COM217, COM059, COM432 |
| [BSW00416]<br>Sequence of Initialization | not applicable<br>(not in scope of this spec) |
| [BSW00406]<br>Check module initialization | COM433 |
| [BSW00437]<br>NoInit-Area in RAM | not applicable<br>(not in scope of this spec) |
| [BSW168]<br>Diagnostic interface | not applicable<br>(not in scope of this spec) |
| [BSW00407]<br>Function to read out published parameters | COM426 |
| [BSW00423]<br>Usage of SW-C template to describe BSW modules with AUTOSAR Interfaces | not applicable<br>(not valid for AUTOSAR COM module) |
| [BSW00424]<br>BSW main processing function task allocation | not applicable<br>(implementation specific) |
| [BSW00425]<br>Trigger conditions for schedulable objects | COM398, COM399, COM400, COM359 |
| [BSW00426]<br>Exclusive areas in BSW modules | not applicable<br>(implementation specific) |
| [BSW00427]<br>ISR description for BSW modules | not applicable<br>(not valid for AUTOSAR COM module) |
| [BSW00428]<br>Execution order dependencies of main processing functions | not applicable<br>(not valid for AUTOSAR COM module) |
| [BSW00429]<br>Restricted BSW OS functionality access | not applicable<br>(AUTOSAR COM has no interface to OS) |
| [BSW00431]<br>The BSW Scheduler module implements task bodies | not applicable<br>(not in scope of this spec) |
| [BSW00432]<br>Modules should have separate main processing functions for read/receive and write/transmit data path | COM398, COM399, COM400, COM359, COM466 |
| [BSW00433]<br>Calling of main processing functions | not applicable<br>(not in scope of this spec) |
| [BSW00434]<br>The schedule module shall provide an API for exclusive areas | not applicable<br>(not in scope of this spec) |

| ***Requirement*** | ***Satisfied by*** |
|---|---|
| [BSW00336]<br>Shutdown interface | COM129, COM130 |
| [BSW00337]<br>Classification of errors | Chapter 7.10 |
| [BSW00338]<br>Detection and reporting of development errors | COM024, COM028, COM442<br><br>Configuration:<br>COM141 |
| [BSW00369]<br>Do not return development error codes via API | COM442, COM459, Chapter 8 |
| [BSW00339]<br>Reporting of production relevant error status | COM459, COM428 |
| [BSW00417]<br>Reporting of error events by non-basic software | not applicable<br>(not in scope of this spec) |
| [BSW00323]<br>API parameter checking | COM024, COM028 |
| [BSW004]<br>Version check | COM026, COM337 |
| [BSW00409]<br>Header files for production code error IDs | COM431 |
| [BSW00385]<br>List possible error notifications | COM442, COM459 |
| [BSW00386]<br>Configuration for detecting an error | not applicable<br>(not in scope of this spec) |
| [BSW161]<br>Microcontroller abstraction | not applicable<br>(not in scope of this spec) |
| [BSW162]<br>ECU layout abstraction | not applicable<br>(not in scope of this spec) |
| [BSW005]<br>No hard coded horizontal interfaces within MCAL | not applicable |
| [BSW00415]<br>User dependent include files | Chapter 5.4, COM005, COM220 |
| [BSW164]<br>Implementation of interrupt service routines | not applicable<br>(not in scope of this spec) |
| [BSW00325]<br>Runtime of interrupt service routines | not applicable<br>(not in scope of this spec) |
| [BSW00326]<br>Transition from ISRs to OS tasks | not applicable<br>(not in scope of this spec) |
| [BSW00342]<br>Usage of source code and object code | Chapter 10.2 |
| [BSW00343]<br>Specification and configuration of time | Chapter 10.2 |
| [BSW160]<br>Human-readable configuration data | Chapter 10.2 |
| [BSW007]<br>HIS MISRA C | API is MISRA conform other issues are implementation specific |
| [BSW00300]<br>Module naming convention | COM005, COM220, COM540 |
| [BSW00413]<br>Accessing instances of BSW modules | not applicable<br>(not in scope of this spec) |
| [BSW00347]<br>Naming separation of different instances of BSW drivers | not applicable<br>(not in scope of this spec) |
| [BSW00305]<br>Self-defined data types naming convention | see Chapter 8.2 |
| [BSW00307]<br>Global variables naming convention | not applicable<br>(implementation specific) |

| ***Requirement*** | ***Satisfied by*** |
|---|---|
| [BSW00310]<br>API naming convention | Chapter 8.3 and 8.4 |
| [BSW00373]<br>Main processing function naming convention | Chapter 8.5 |
| [BSW00327]<br>Error values naming convention | COM442, COM459 |
| [BSW00335]<br>Status values naming convention | Chapter 8.2.1 |
| [BSW00350]<br>Development error detection keyword | COM028 |
| [BSW00408]<br>Configuration parameter naming convention | Chapter 10.2 |
| [BSW00410]<br>Compiler switches shall have defined values | not applicable<br>(implementation specific) |
| [BSW00411]<br>Get version info keyword | COM425 |
| [BSW00346]<br>Basic set of module files | COM005, COM220 |
| [BSW158]<br>Separation of configuration from implementation | COM005, COM220 |
| [BSW00314]<br>Separation of interrupt frames and service routines | not applicable<br>(not in scope of this spec) |
| [BSW00370]<br>Separation of callback interface from API | Chapter 8 |
| [BSW00435]<br>Module header file structure for the basic software scheduler | COM220 |
| [BSW00436]<br>Module header file Structure for the basic software memory mapping | COM220 |
| [BSW00348]<br>Standard type header | COM220 |
| [BSW00353]<br>Platform specific type header | not applicable<br>(implementation specific) |
| [BSW00361]<br>Compiler specific language extension header | not applicable<br>(implementation specific) |
| [BSW00301]<br>Limit imported information | not applicable<br>(implementation specific) |
| [BSW00302]<br>Limit exported information | not applicable<br>(implementation specific) |
| [BSW00328]<br>Avoid duplication of code | not applicable<br>(implementation specific) |
| [BSW00312]<br>Shared code shall be reentrant | COM320, COM321 |
| [BSW006]<br>Platform independency | not applicable<br>(implementation specific) |
| [BSW00357]<br>Standard API return type | Chapter 8 |
| [BSW00377]<br>Module specific API return types | Chapter 7.10, COM459 |
| [BSW00304]<br>AUTOSAR integer data types | COM220,<br>(implementation specific) |
| [BSW00355]<br>Do not redefine AUTOSAR integer data types | Chapter 7.10 and Chapter 8.2 |
| [BSW00378]<br>AUTOSAR boolean type | not applicable<br>(implementation specific) |
| [BSW00306] | not applicable |

| ***Requirement*** | ***Satisfied by*** |
|---|---|
| Avoid direct use of compiler and platform specific keywords | (implementation specific) |
| [BSW00308]<br>Definition of global data | not applicable<br>(implementation specific) |
| [BSW00309]<br>Global data with read-only constraint | not applicable<br>(implementation specific) |
| [BSW00371]<br>Do not pass function pointers via API | Chapter 8 |
| [BSW00358]<br>Return type of init functions | COM432 |
| [BSW00414]<br>Parameter of init function | COM432 |
| [BSW00376]<br>Return type and parameters of main processing functions | Chapter 8.5 |
| [BSW00359]<br>Return type of callback functions | COM468, COM491, COM536, COM554, COM555, COM556 |
| [BSW00360]<br>Parameters of callback functions | COM468, COM491, COM536, COM554, COM555, COM556 |
| [BSW00329]<br>Avoidance of generic interfaces | Chapter 8 |
| [BSW00330]<br>Usage of macros / inline functions instead of functions | COM434 |
| [BSW00331]<br>Separation of error and status values | Chapter 8, COM194 |
| [BSW009]<br>Module user documentation | not applicable<br>(implementation specific) |
| [BSW00401]<br>Documentation of multiple instances of Configuration parameters | Chapter 10 |
| [BSW172]<br>Compatibility and documentation of scheduling strategy | see Chapter 8.5, COM298 further item are implementation specific |
| [BSW010]<br>Memory resource documentation | not applicable<br>(implementation specific) |
| [BSW00333]<br>Documentation of callback function context | not applicable<br>(not in scope of this spec) |
| [BSW00374]<br>Module vendor identification | COM208 |
| [BSW00379]<br>Module identification | COM417 |
| [BSW003]<br>Version identification | COM426, COM425, COM424, COM026, COM337, COM141, COM208, COM417, COM418, COM419, COM420, COM421, COM422, COM423, COM438 |
| [BSW00318]<br>Format of module version numbers | COM026 |
| [BSW00321]<br>Enumeration of module version numbers | not applicable<br>(not in scope of this spec) |
| [BSW00341]<br>Microcontroller compatibility documentation | not applicable<br>(not in scope of this spec) |
| [BSW00334]<br>Provision of XML file | not applicable<br>(not in scope of this spec) |

Document: Requirements on Communication [7]

| *Requirement* | *Satisfied by* |
|---|---|
| [BSW02037]<br>AUTOSAR COM shall be based on the functionality and APIs of OSEK COM 3.0.3 | COM010, COM011, COM012, COM013, COM396 |
| [BSW02078]<br>Support of endianness conversion | COM007, COM221, COM352, COM472, COM473 |
| [BSW02086] Support of sign-extension for received signals | COM008, COM353, COM352 |
| [BSW02042]<br>Initialization of not used areas of an I-PDU | COM015 |
| [BSW02040]<br>AUTOSAR COM configuration language | COM006, Chapter 10 |
| [BSW177]<br>Configuration of communication parameters | Chapter 10.2 |
| [BSW02067]<br>Rules for checking the consistency of Configuration input | COM101, COM102, COM105, COM319, COM365, COM373, COM384, COM386, COM310, COM401, COM402, COM443, COM474, COM489, COM535, COM553 |
| [BSW02046]<br>Configuration of signal notification | COM298, COM300, COM301, COM393 |
| [BSW02089]<br>Timeout indication mechanism on receiver-side [approved] | [17]<br><br>COM292, COM290, COM291, COM393<br><br>Configuration:<br>COM186, COM263, COM264, COM333, COM552 |
| [BSW02088]<br>Value substitution in case of a signal timeout [approved] | COM393 |
| [BSW02083]<br>Transmission modes [approved] | COM329, COM330, COM135, COM276, COM305, COM392, COM277, COM278, COM307, COM308, COM467, COM478, COM494<br><br>Configuration:<br>COM351, COM178, COM137, COM180, COM281, COM282 |
| [BSW02082]<br>Two different transmission modes | COM032, COM239, COM244, COM238, COM495<br>Configuration:<br>COM454, COM455, COM465 |
| [BSW02084]<br>Transmission mode selection | COM274, COM283, COM241, COM245, COM284, COM255, COM032 |
| [BSW02080]<br>Re-triggering of repetitions of I-PDUs | COM279 |
| [BSW02077]<br>Signal invalidation mechanism on sender-side | COM097, COM099, COM286<br><br>API:<br>COM203, COM288, COM557<br><br>Configuration:<br>COM314, COM315, COM391, COM501 |
| [BSW02079]<br>Signal invalidation mechanism on receiver-side | COM100, COM287, COM323, COM536, COM558, COM559 |
| [BSW02087]<br>Substitution of invalid value by configurable data | COM100, COM412, COM470, COM500 |

| ***Requirement*** | ***Satisfied by*** |
|---|---|
| value | |
| [BSW218]<br>Separate start/stop AUTOSAR COM for separate groups of I-PDUs | COM085, COM114, COM222, COM223, COM228, COM229, COM334, COM090, COM115, COM311, COM187, COM444, COM476, COM479<br><br>API:<br>COM190, COM191<br><br>Configuration:<br>COM341, COM126, COM184, COM185 |
| [BSW192]<br>Disable reception deadline monitoring | COM092, COM225<br><br>API:<br>COM192, |
| [BSW02081]<br>Re-enable reception deadline monitoring | COM095, COM224, COM486, COM534<br><br>API:<br>COM193 |
| [BSW02041]<br>Atomic transfer of complex data types | COM042, COM043, COM047, COM049, COM050, COM051, COM052, COM053, COM327, COM326, COM461, COM464<br><br>API:<br>COM199, COM200, COM201, COM202<br><br>Configuration:<br>COM345, COM044, COM513, COM520, COM521 |
| [BSW02043]<br>Indication service Com_RxIndication | COM574, COM575<br>API:<br>COM123 |
| [BSW02044]<br>Confirmation service Com_TxConfirmation | COM260<br><br>API:<br>COM124 |
| [BSW02045]<br>Function Com_TriggerTransmit | API:<br>COM001, COM260, COM475 |
| [BSW02030]<br>Identify if a signal is updated by the sender | COM054, COM055, COM056, COM057, COM059, COM061, COM062, COM067, COM076, COM324, COM117, COM310<br><br>Configuration:<br>COM257 |
| [BSW02058]<br>Deadline monitoring of receiving updated signals | COM292, COM290, COM291, COM117, COM393 |

Document: OSEK/ VDX Communication Version 3.0.3 [17]

| ***Requirement*** | ***Satisfied by*** |
|---|---|
| Filtering (Section 2.2.2 in [17]) | COM325, COM380, COM272, COM273, COM132, COM230, COM302, COM303, COM231, COM439, COM480<br><br>Configuration:<br>COM339, COM235, COM312, COM313, COM146, COM147, COM317, COM318 |
| Reception deadline monitoring (Section 2.5.1 in [17]) | COM292, COM290, COM291 |
| Transmission deadline monitoring (Section 2.5.2 in [17]) | COM304, COM445, COM481 |
| I-PDU callout (Section 2.9.3.2, Appendix C in [17]) | COM381, COM346, COM347, COM395, COM388, COM492<br><br>API:<br>COM348 |
| OSEK APIs | COM197, COM198, COM469, COM562<br><br>Configuration:<br>COM340, COM174, COM175, COM176, COM017, COM181, COM119, COM387, COM206, COM233, COM234, COM157, COM158, COM259, COM183, COM263, COM264, COM333, COM344, COM163, COM165, COM232, COM170, COM127, COM169, COM275, COM437, COM471, COM493, COM496, COM497, COM518, COM519 |
| Notifications | API:<br>COM123, COM124<br><br>Configuration:<br>COM498, COM499 |

Document: Requirements on Gateway [9]

| ***Requirement*** | ***Satisfied by*** |
|---|---|
| Interface between AUTOSAR COM and the lower layer (PDU-Router) | COM138 |
| [BSW06002]<br>Updateable Configuration | COM373, COM357, COM361, COM544, COM545, COM546, COM547, COM548, COM549, COM550, COM551 |
| [BSW06097]<br>Configuration identification | COM374, COM375, COM394, COM487 |
| [BSW06003]<br>Static Routing Rules | COM376<br><br>Configuration:<br>COM355, COM358 |
| [BSW06055]<br>Signal Based Gateway | COM377, COM539 |
| [BSW06056]<br>Gateway of Signal Groups | COM361, COM383 |
| [BSW06061]<br>Routing operation on Signals | COM371, COM360, COM361, COM362 |
| [BSW06098] | COM024, COM442, COM459 |

| ***Requirement*** | ***Satisfied by*** |
|---|---|
| Signal Gateway Error Handling at signal routing | |
| [BSW06099]<br>Signal Gateway Error Handling at signal routing | COM024, COM442, COM459 |
| [BSW06077]<br>Routing of multiple signals of the same PDU | COM371 |
| [BSW06089]<br>Timeout handling | COM381, COM377, COM568, COM569, COM570, COM571, COM572, COM573 |
| [BSW06064]<br>Signal gateway scalability | COM370 |

# 7 Functional specification

## 7.1 Introduction

### 7.1.1 Signal values

The signals sent by AUTOSAR COM respectively received by AUTOSAR COM could have the values defined in Table 1.

| ***Signal value*** | ***Remark*** |
|---|---|
| init value | See Chapter 7.4.1.4 for details. |
| data invalid value | See Chapter 7.4.3.6 for details. |
| <value> | This is the normal case:<br>A valid value after initialization phase which is sent by AUTOSAR COM respectively received by AUTOSAR COM. |

**Table 1: Possible signal values**

## 7.2 General functionality

### 7.2.1 OSEK-COM

OSEK COM 3.0.3 is the functional basis of AUTOSAR COM.

**COM010:** AUTOSAR COM 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 COM013.

**COM011:** If this specification defines functionality in a different way compared to definitions in [17], the definitions made in this AUTOSAR COM specification shall be used.

**COM012:** AUTOSAR COM shall in addition implement all those features, that are defined in this specification and that are not part of [17].

**COM013:** AUTOSAR COM needs *not* to implement the following features of [17]. If they are implemented in a specific AUTOSAR COM configuration the configuration shall by default disable them. This also applies for all other additional features a specific implementation may offer.

| ***OSEK-COM feature*** | ***Rationale*** | ***related OSEK COM API*** |
|---|---|---|
| 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 mes- | not required, done by the | none |

| *OSEK-COM feature* | *Rationale* | *related OSEK COM API* |
|---|---|---|
| sage to both an external send message object and an internal receive message object (1:n splitting mechanism) | RTE, see [13] | |
| 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<br>functionality is partly covered by Com_TriggerTransmit | SendZeroMessage |
| Dynamic size messages | were introduced for diagnostics, no use case in AUTOSAR | SendDynamicMessage, ReceiveDynamicMessage |
| 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 | not the OSEK OIL shall be used to configure the AUTOSAR COM | none |
| Application modes | not needed | GetApplicationMode |
| Start-up behavior | replaced by<br>• Com_Init<br>• Com_DeInit<br>• Com_IpduGroupStart<br>• Com_IpduGroupStop | 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,<br>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<br>CPU-order message callout | Only I-PDU callouts with a defined AUTOSAR interface are supported by AUTOSAR COM. This is to avoid proprietary solutions. | none |
| Error hook routine | AUTOSAR COM will use a direct interface to DEM/DET instead of using the OSEK COM error hook | COMErrorHook<br>COMError_Name1_Name2 macros<br>COMErrorGetServiceId |

| *OSEK-COM feature* | *Rationale* | *related OSEK COM API* |
|---|---|---|
| Interface for callback routines | The signatures for the used callback function in AUTOSAR COM will be explicitly defined within AUTOSAR COM. | COMCallback |
| Internal communication | not required, ensured by RTE, see [13] | SendMessage,<br>ReceiveMessage |

**Table 2: Excluded OSEK COM features in AUTOSAR COM**

### 7.2.2 Endianness conversion and sign extension

**COM007:** To support the required AUTOSAR data types (signed- and unsigned integer, ASCII, enum, opaque bitfield) the AUTOSAR COM layer shall provide support for endianness conversion of all integer types. Other data types (ASCII, enum, opaque[1]) are either treated as signed or unsigned integer or nothing has to be done, i.e. their contents is not interpreted by the AUTOSAR COM layer.

The supported data types of AUTOSAR COM are:

- boolean
- uint8
- uint16
- uint32
- sint8
- sint16
- sint32

uint8[n]

**COM472:** Opaque data shall always be of uint8[n] and shall always be mapped to an n-bytes sized signal.

**Note:** For opaque data (see [13]) endianness conversion shall be configured to *Opaque* (see COM157).

**COM473:** If the endianness conversion is configured to *Opaque* (see COM157) the parameter ComSignalBitPosition (see COM259) shall define the bit0 of the first byte like in little endian byte order.

**Remark:** [17] Chapter 2.4 defines the endianness conversion for unsigned data types. This endianness conversion shall be extended to signed data types. The associated configurations can be found in the configuration chapter. See also COM127 and COM157.

[1] This Data type represents an array of exactly numberOfBits bits. It is called *opaque* because this array of bits should be transported "as is" by the AUTOSAR communication stack.

**COM008:** To map negative values of signed data correctly, the received data shall be extended to the size of the receiver (sign extension). The platform specific representation of signed data shall be taken into account.

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

**COM353:** On sender side there shall be no sign extensions.

**COM221:** Endianness conversion shall take place before the I-PDU callout on the sender side. (For an overview see Chapter 7.12).

**COM352:** Sign extensions and endianness conversion shall take place before configuration filtering and notification detection on receiver side

### 7.2.3 Filtering

**COM272:** Each filtering condition shall either evaluate to true or false. Filtering out signals shall only take place at receiver side. On sender side the mechanisms are still used for Transmission Mode Conditions (TMC) but no signal filtering shall take place.

**Note:** For Transmission Mode Selection (TMS) see Chapters 7.4.3.4 and 7.4.3.3.

**COM480:** AUTOSAR COM shall only provide the following filter types which are defined in [17]

- F_Always
- F_Never
- F_MaskedNewEqualsX
- F_MaskedNewDiffersX
- F_MaskedNewDiffersMaskedOld
- F_NewIsWithin
- F_NewIsOutside
- F_OneEveryN

**Note:** Some filters defined in [17] are removed from AUTOSAR COM to reduce complexity. The removed filters were either obsolete or special cases of other filters. For example the filter F_NewIsDifferent is a special case of F_MaskedNewDiffersMaskedOld with a fully set mask.

**COM325:** All filter mechanisms defined in COM480 shall be supported for all data types listed in COM007; considering the exceptions defined in COM380 and COM439.

**COM380:** For the type uint8[n] only the filters F_Always and F_Never shall be supported.

**COM439:** For the type boolean only the following filters shall be supported:

- F_Always
- F_Never
- F_MaskedNewEqualsX
- F_MaskedNewDiffersX
- F_MaskedNewDiffersMaskedOld
- F_OneEveryN

**COM273:** If a signal is filtered out on receiver side (i.e. filter condition evaluates to FALSE), it shall be discarded and not be further processed. See also COM303.

**COM327:** Filtering out of signals as specified in COM273 shall not be applied to group signals.

**Note:** Conditions for TMS may be applied to group signals, see COM326.

**COM132:** The filtering mechanisms, defined in [17] and in Chapter 7.4.3.3 shall also be supported for signed data types.

In the case a filter is evaluated before a signal has been written to by a send-API, 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. So it is necessary to define the value used by the filter for *new_value* in the period before the first send takes place.

**COM230:** The *old_value* of the filtering mechanisms is set during start-up to the init-value, see [17]. Up to the point in time where the application has not updated the value, for the *new_value* also the init-value shall be used.

The next two requirements shall clarify the definitions of [17] according to the update of the *old_value* of filters.

**COM302:** If a filter is evaluated to true (value is not filtered out) then the value is placed into *old_value* (as defined in [17]).

**COM303:** 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 value is not placed into *old_value* (as defined in [17]).

**COM231:** In the case of the filter F_OneEveryN,

- OCCURRENCE shall be set to zero by Com_IpduGroupStart
- FILTER shall be set true when OCCURRENCE == OFFSET
- OCCURRENCE shall be incremented after filter processing
- When OCCURRENCE == PERIOD, OCCURRENCE shall be set to zero.

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

This definition of the filter F_OneEveryN has the effect that the signal is passed by the filter (i.e. the filter returns true) once every PERIOD call 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 message must pass through the filter before it returns true.

This definition exists to clarify the description of the F_OneEveryN filter in [17].

The associated configuration items can be found in the configuration chapter, see COM339.

### 7.2.4 Signal Gateway

AUTOSAR COM provides a signal gateway for forwarding signals and signal groups in a 1:n manner.

Signals and signal groups to be routed by the signal gateway are identified and configured by unique static names. The signal gateway determines the destination of a signal or of a signal group by using its name and a configuration table.

**COM370:** Furthermore the signal gateway should scale down to no size if no signal routing functionality is needed.

**COM371:** The signal gateway is placed as shown in Figure 3, Figure 13, Figure 14 and Figure 15.

## 7.3 Configuration

See Chapter 10.

## 7.4 Normal operation

### 7.4.1 Start-up behavior

This chapter describes the actions which shall be performed during Com_Init.

**COM217:** The I-PDU init value (Chapter 7.4.1.3) and the signal / update-bit init value (Chapter 7.4.1.4) together result in one init value for each I-PDU. During Com_Init this init value shall be used to initialize the I-PDU.

#### 7.4.1.1 Preconditions

**Note:** The C initialization code (also known as *start-up code*) which initializes global and static variables with the initial values shall be executed before any call of an AUTOSAR COM service.

#### 7.4.1.2 Initialization

**COM128:** All internal data that is not yet initialized by the *start-up code* e.g. C-structs shall be initialized by Com_Init (see COM432).

**COM328:** The AUTOSAR COM initialization routine shall not enable Inter-ECU communication. Inter-ECU communication requires a separate start (see Chapter 7.4.5 for details).

The API function is defined by COM432.

**Note**: This initialization chapter is not complete. Details about initialization of some COM features are described within the different feature chapters.

#### 7.4.1.3 Initialization of not used areas of an I-PDU

**COM015:** AUTOSAR COM shall fill not used areas within an I-PDU with a value configurable at configuration-time, e.g. 0xFF.

See also COM017 for the associated configuration element.

#### 7.4.1.4 Initialization of signals and Update-bits

**COM098:** AUTOSAR COM shall initialize all signals on sender and receiver side with the configured init values which shall be the same value as used for the initialization of the signal in the related I-PDU (see COM217). For configuration see COM170 in the configuration chapter.

The signal init values may be identical to the AUTOSAR COM data invalid values. These may be different for each signal. For the related configuration items see COM391 and COM501.

**COM117:** AUTOSAR COM shall clear all update-bits during initialization. See also COM059.

#### 7.4.1.5 Initialization of I-PDU groups

**COM444:** By default all I-PDU groups shall be in the state ‘stopped’ and they shall not be started automatically while AUTOSAR COM initialization.

### 7.4.2 Shutdown behavior

#### 7.4.2.1 De-initialization

**COM129:** AUTOSAR COM shall provide a functionality to de-initialize the layer. This means that after de-initialization of the layer no communication via AUTOSAR COM is possible and all started I-PDU groups are stopped.

The API function is defined by COM130.

### 7.4.3 Communication modes

This chapter defines the signal flow in AUTOSAR COM and how the different transmission modes in AUTOSAR COM are defined. Chapter 7.4.3.6 shows exemplary communication use cases that can be solved with AUTOSAR COM. Chapter 7.4.3.3 defines a mechanism to switch between two transmission modes for one I-PDU. The replication of signals is defined in Chapter 7.4.3.5.

#### 7.4.3.1 Transfer Properties and I-PDU Transmission Modes

##### 7.4.3.1.1 Signals

[17] distinguishes between:

- Transfer properties (applies for signals)
- Transmission modes (applies for I-PDUs).

The following two tables are derived from [17]. ***The AUTOSAR extensions are marked with bold and configuration letters.***

| ***TRANSFER PROPERTY***<br>***(per signal)*** | ***Description*** |
|---|---|
| ***TRIGGERED*** | **COM329:** The triggered transfer property causes immediate transmission of the I-PDU, except if transmission mode *PERIODIC* or ***transmission mode NONE*** is defined for the I-PDU (see Table 4). |
| ***PENDING*** | The *PENDING* transfer property does not cause transmission of an I-PDU. |
| ***TRIGGERED_ON_CHANGE*** | ***COM563: 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 initiate ComTxModeNumberOfRepetitions transmissions of the assigned I-PDU if the new value of the sent signal differs to the locally stored (last sent or init) value.*** |

**Table 3: Transfer Properties defined for signals**

**Note:** A pending signal associated with an I-PDU is transmitted if either a signal with *TRIGGERED* transfer property within the same I-PDU is sent or the I-PDU is sent because of a timer (Periodic / Mixed transmission mode).

| ***Transmission mode***<br>***(per I-PDU)*** | ***Description*** |
|---|---|
| ***DIRECT***<br>***(N-Times)*** | **COM330**: Transmission of an I-PDU with Direct transmission mode is caused by the transfer of any signal assigned to the I-PDU with the *TRIGGERED* or *TRIGGERED_ON_CHANGE* transfer property. The transfer of the signal to AUTOSAR COM is immediately followed by ***n transmissions*** (n = 1 … m, m <= 255) on the underlying layer (associated requirements see 7.4.3.5). |
| *PERIODIC* | In Periodic transmission mode, AUTOSAR COM issues periodic transmission requests for an I-PDU to the underlying layer. |
| *MIXED* | Mixed transmission mode is a combination of the “Direct/N-Times” and the Periodic transmission modes. |
| ***NONE*** | **COM135**: ***In transmission mode “None” no transmission is initiated by AUTOSAR COM. It shall only be possibly to request I-PDUs with Com_TriggerTransmit (see COM001). The configuration is defined by COM137.*** |

**Table 4: Transmission modes defined for I-PDUs**

On the one hand the timing of bus messages can be controlled by send requests of the upper layer in combination with the transmission mode and the transfer property as described above. On the other hand it can be controlled by the lower layer (esp. for FlexRay and LIN) with the service Com_TriggerTransmit. In this case the lower layer requests I-PDUs that have to be provided by COM.

**COM260**: It shall be possible to use Com_TriggerTransmit to request the transmission of any I-PDU with any transmission mode.

**Note:** This allows LIN and FlexRay to use all the available transmission modes, particularly for sporadic communication. This mechanism is also used by the NM to send user data.

### 7.4.3.1.2 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.

| ***TRANSFER PROPERTY***<br>***(per signal group)*** | ***Description*** |
|---|---|
| ***TRIGGERED*** | **COM564 :** The triggered transfer property causes immediate transmission of the I-PDU, except if transmission mode *PERIODIC* or transmission mode *NONE* is defined for the I-PDU (see Table 4). |
| ***PENDING*** | **COM565 :** The “Pending” transfer property does not cause transmission of an I-PDU. |
| ***TRIGGERED_ON_CHANGE*** | At a send request of a signal group with ComTransferProperty *TRIGGERED_ON_CHANGE* assigned to an I-PDU with ComTxModeMode *DIRECT* or *MIXED* two cases shall be distinguished: |

| | |
|---|---|
| | **COM566:** If none of the signal group's group signals has a ComTransferProperty specified, the AUTOSAR COM module shall immediately initiate ComTxModeNumberOfRepetitions transmissions of the assigned I-PDU, if at least one new value of the signal group's group signals differs to the locally stored (last sent or init) value.<br>**COM567:** If at least one group signal of a signal group has the ComTransferProperty configured, all other group signals of that signal group shall have configured ComTransferProperty as well. If the ComTransferProperty is set to *PENDING*: a change of the value of this group signal shall trigger the transmission of the assigned I-PDU.<br>If the "ComTransferProperty" is set to *TRIGGERED_ON_CHANGE*, a change of the value of this group signal shall trigger the transmission of the assigned I-PDU. |

#### 7.4.3.2 Link to the VFB specification

**Note:** This chapter is just an illustration how the transfer mode relates to the VFB specification and links to a non normative part of the VFB specification.

Because, from the point of view of the AUTOSAR SW Component, it is not known at implementation time, which communication media is used, all bus specific replications of send requests by a SW component to underlying layers as well as all bus specific timing behavior must be done either by AUTOSAR COM or by the appropriate bus interfaces and drivers.

AUTOSAR COM implements the replication of transmission requests and the bus specific timing behavior by a combination of transfer properties and transmission modes, which is shown in the table below. The entries in the table correspond to the VFB's send modes:

| ***Transfer property (horizontal)***<br>***Transmission mode (vertical)*** | ***TRIGGERED/***<br>***TRIGGERED_ON_CHANGE*** | ***PENDING*** |
|---|---|---|
| Direct/N-Times | N-Times | -- |
| Periodic | -- | Cyclic |
| Mixed | N-Times | Cyclic |
| None | -- | -- |

**Table 5: Mapping of transfer property and transmission Mode (I-PDU) to send modes defined in the AUTOSAR Specification of the Virtual Functional Bus [6]**

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

Signals are carried by I-PDUs. 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.

AUTOSAR COM allows configuring statically two different transmission modes for each I-PDU (see COM032). 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.

**COM326:** For the selection of the transmission mode the group signals shall be treated like normal signals. Therefore each group signal shall contribute, depending on its configuration, to the evaluation of the transmission mode.

The following definitions give an overview about terms used in the following specification articles.

| ***Definitions*** | |
|---|---|
| $s_i$ | Signal in AUTOSAR COM |
| $IPDU_k$ | I-PDU in AUTOSAR COM |
| $C(s_i , IPDU_k )$ | Transmission Mode Condition (TMC), condition attached to a signal ($s_i$) within an I-PDU ($IPDU_k$) |
| $S_k$ | The set of all signals ($s_i$) within one I-PDU ($IPDU_k$) |
| $M_k$ | The set of all signals ($s_i$) within one I-PDU ($IPDU_k$) which contribute according to their configuration to the selection of the transmission mode.<br>$M_k$ has to be a subset of $S_k$. |
| $TMS_k$ | The Transmission Mode Selector (TMS) is calculated from the evaluation of all $C(s_i, IPDU_k )$ for all $s_i$ in $M_k$ |

**Table 6: Definitions for transmission mode selection**

**COM274:** To each signal ($s_i$) mapped to a specific I-PDU ($IPDU_k$) a condition $C(s_i, IPDU_k)$ shall be associated (see COM275).

**COM283:** The conditions $C(s_i, IPDU_k)$ available shall be the same as provided for the filter mechanism (see COM272 and COM480). The definitions made in COM230 and COM231 shall also be applied for the evaluation of the TMS.

**Definition:** (see also [17])
For each I-PDU ($IPDU_k$) a Transmission Mode Selector ($TMS_k$) is defined. $TMS_k$ is calculated by evaluating the conditions $C(s_i , IPDU_k )$ for a configurable subset of signals $M_k$ (see COM275) of the set of all signals $S_k$ mapped to the $IPDU_k$.

$TMS_k$ is defined to be true, if at least one $C(s_i , IPDU_k )$ in $M_k$ evaluates to true and is defined to be false, if all $C(s_i , IPDU_k)$ in $M_k$ evaluate to false.

$$TMS_k = \begin{cases} \text{true, if at least one } C(s_i, IPDU_k) \equiv \text{true for all } s_i \in M_k \\ \text{false, if } C(s_i, IPDU_k) \equiv \text{false for all } s_i \in M_k \end{cases}$$

**Table 7: Calculation of transmission mode selector**

**COM241:** AUTOSAR COM shall be able to define and calculate a TMS for each I-PDU as defined in the definition given above.

**COM245:** The TMS of each I-PDU containing a specific signal shall be re-calculated within a call of Com_SendSignal respective Com_SendSignalGroup by the upper layers for that specific signal.

**COM284:** AUTOSAR COM must provide the mechanisms to specify which signals shall contribute to the evaluation of TMS, i.e. to define the set of signals $M_k$ that contribute to the evaluation of the $TMS_k$.

For the configuration see COM275.

**COM255:** If no signal within an I-PDU is configured to contribute to the calculation of the TMS, the TMS shall be set to true.

**Comment:** Note that a signal with $C(s_i, IPDU_k)$ F_ALWAYS – if it contributes to the selection of the transmission mode – 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.

**COM032:** On sender-side two independent transmission modes shall be configured for each I-PDU. One of those transmission modes shall be valid if the TMS evaluates to true, the other transmission mode shall be valid, if the TMS evaluates to false.

The transmission modes, which can be configured separately for each evaluation result, are defined in the configuration chapter, see COM233 and COM234.

**COM238:** 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.

**COM239:** When the TMS state changes, the now valid transmission mode shall immediately be used. That means, first the mode change shall be done and after that the appropriate requests to the underlying layers, resulting from the request causing the mode change shall be executed.

**COM244:** If a change of the TMS causes a change of the transmission mode for one I-PDU, the timer for the cycle time of the Periodic and the Mixed transmission mode shall be restarted.

**COM495:** By a TMS-switch to the Cyclic or Mixed transmission mode the new cycle shall start with an immediate transmission request to the underlying layers respecting the MDT. See also Figure 6.

**Remark:** In case the TMS evaluates to false, it shall be possible to configure the transmission mode in that way that no transmission takes place (transmission mode

None). This prevents the I-PDU to be transmitted and shall be the default. For the configuration see COM234.

**COM478:** An I-PDU shall be sent out at most once while one call to Com_MainFunctionTx.

#### 7.4.3.4 Signal flow and Transmission Mode Selection

After a send request from an upper layer 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 7.4.3.3.

Figure 3 shows the signal flow:

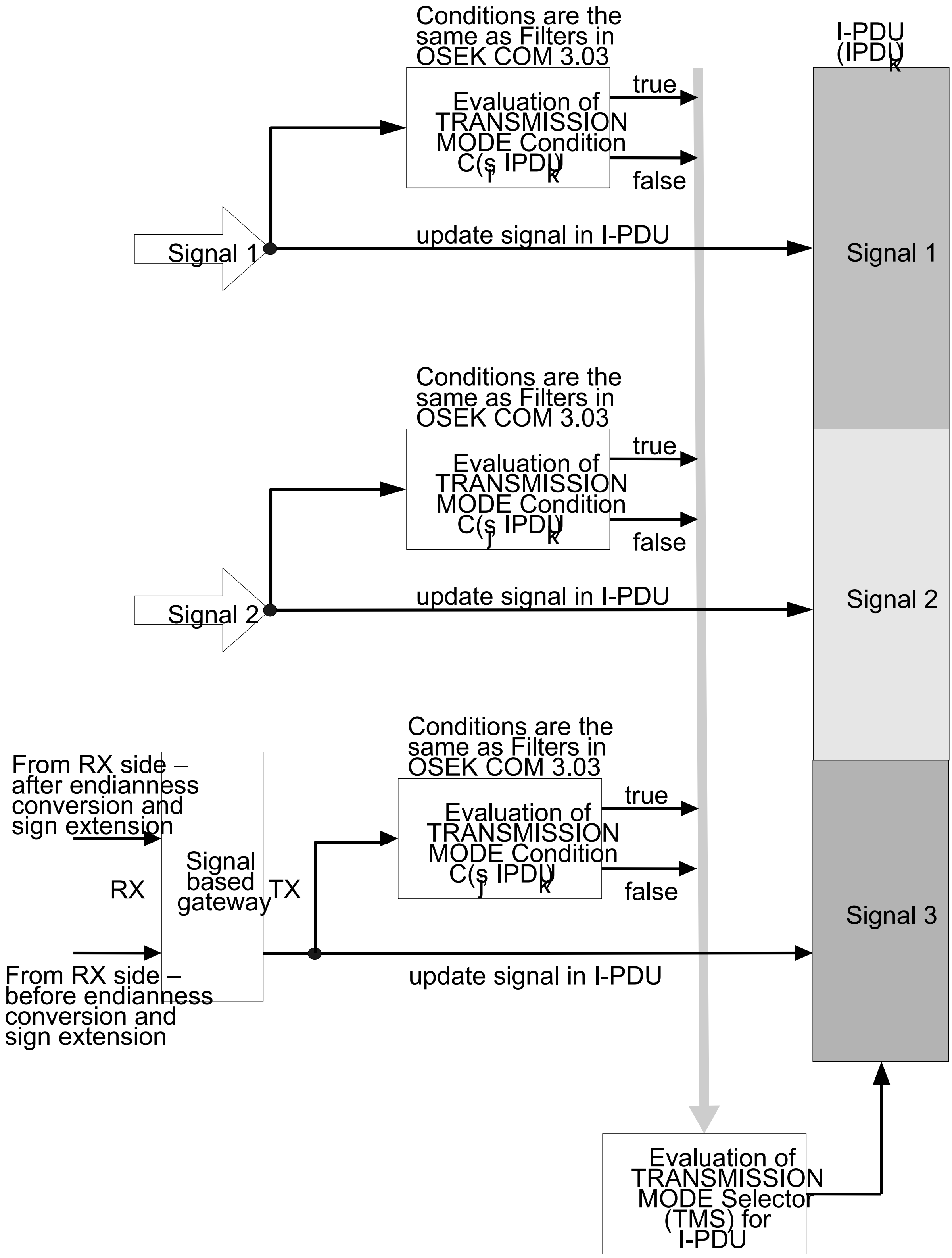
Conditions are the same as Filters in OSEK COM 3.03
I-PDU (IPDU k)
Evaluation of TRANSMISSION MODE Condition C(s i IPDU k)
true
false
Signal 1
update signal in I-PDU
Signal 1
Conditions are the same as Filters in OSEK COM 3.03
Evaluation of TRANSMISSION MODE Condition C(s j IPDU k)
true
false
Signal 2
update signal in I-PDU
Signal 2
Conditions are the same as Filters in OSEK COM 3.03
From RX side – after endianness conversion and sign extension
RX
Signal based gateway
TX
Evaluation of TRANSMISSION MODE Condition C(s j IPDU k)
true
false
From RX side – before endianness conversion and sign extension
update signal in I-PDU
Signal 3
Evaluation of TRANSMISSION MODE Selector (TMS) for I-PDU

**Figure 3: Logical signal flow in AUTOSAR COM shown for two signals (Signal1 and Signal2) that are mapped to one I-PDU ($IPDU_k$)**

### 7.4.3.5 Replication of Signal Transmission Requests

**COM276:** In the Direct/N-Times and Mixed transmission modes AUTOSAR COM shall be able to send n transmission requests (n = 0, 1 … m, m <= 255) to the lower layer for each send request by an upper layer. See also the matching configuration requirement COM281.

**COM467:** If Direct/N-Times or Mixed transmission mode with n == 0 is used this shall result in only one send request without waiting for any confirmation (original OSEK behavior for Direct transmission mode).

**COM279:** If a new send request is received from an upper layer while sending n transmissions belonging together (e.g. after the 3rd of 5 repetitions, see COM276) AUTOSAR COM shall cancel the outstanding transmission repetitions and start processing the new request immediately, see Figure 4.

**COM305:** The confirmation behavior to the upper layer in the Direct/N-Times transmission mode with n > 0 shall be according to the following definition:

1) When an I-PDU is sent by the application with Direct/N-Times transmission mode, n is put into a counter in COM.
2) COM sends to the underlying layer an N-Times I-PDU periodically as long as the counter is non-zero.
3) Whenever a TX confirmation is received and the counter is greater than 0, the counter is decremented. When the counter is 0, transmission confirmations for that I PDU are ignored.
4) When the counter reaches 0 the transmission confirmation is send to the upper layer and transmission deadline monitoring is cancelled (if configured). See COM392 and Chapter 7.4.6.2.

**Note:** The definition in COM305 shall not define a concrete implementation. But every implementation shall implement the confirmation behavior according to the above definition.

**Note:** 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.

**Note:** If the underlying layer returns E_NOT_OK while an N-Times transmission is in progress this error notification shall be ignored (as defined in COM428). As COM305 specifies only confirmed transmissions are counted for the N-Times transmission, erroneous send request can safely be ignored.

**Note:** If the N-Times transmission is requested in Mixed transmission mode 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.

**COM494:** If within the Mixed transmission mode 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.

**COM392:** If a transmission deadline monitoring timeout occurs before the N-Times transmission is complete, the N-Times transmission shall be abandoned.

**Note:** The minimum delay time shall always be taken into account as defined in [17].

**Note:** To avoid bursts in start-up a time offset can be configured per I-PDU. For details see COM180.

**COM277:** The number of repetitions (n) shall be configurable (see COM281).

**COM278:** The time between two repetitions shall be configurable (see COM282).

**Note:** If the transmission mode change leads to the start of the Mixed transmission mode by sending a triggered signal and ComTxModeNumberOfRepetitions is configured > 1 then there will be at least n transmission requests to the lower layer at the beginning of the Mixed transmission mode. See also COM276.

**COM310:** It shall not be possible to use/configure update-bits for signals that are transmitted within I-PDUs having the Direct/N-Times transmission mode with n>=1. (I.e. only n=0 is allowed).

**Note:** There is no common understanding how update-bits and N-Times transmission can be combined. It is unclear when update-bits shall be set and cleared respecting the replication. Therefore this is currently forbidden.

#### 7.4.3.6 Use Cases for communication modes

This chapter shows use cases which have to be realizable by the transmission modes and transfer properties specified in AUTOSAR COM.

**Note:** For a more detailed discussion of the following use cases see Appendix A.

| ***Use case diagram legend*** | |
|---|---|
| $t_c$, $t_{c1}$, $t_{c2}$ | Cycle times |
| $t_d$ | Cycle time of N-Times send signals |
| $t_{r,\ min}$ | Minimum SW reaction time of COM-Layer, e.g. due to internal cycle time |
| V | Value:<br>x stands for an arbitrary value / value range,<br>a…w for specific values / value ranges, defined by the user, with<br>a <> b, *range a* is disjoint from *range b*. |
| ⚡ | Request from upper layers (RTE) to the COM-Layer |
| ↓ | Request from COM-Layer to lower layers (PDU-Router) |
| ⇣ | Potential but skipped request from COM-Layer to lower layers (e.g. because of a new send request by upper layers or delayed due to minimum delay time) |
| dt | Minimum distance between two requests to lower layers (minimum delay time), dt can be set per I-PDU. |
| w/o TMS switch | Without switching of the TMS (see 7.4.3.3) from *true* to *false* or vice versa. |
| w TMS switch | With switching of the TMS (see 7.4.3.3) from *true* to *false* or vice versa (from TM 1 to TM 2)<br>One TM is named before the "+" and one behind in the description. |

**Table 8: Legend for use case diagrams.**

Use case diagrams:

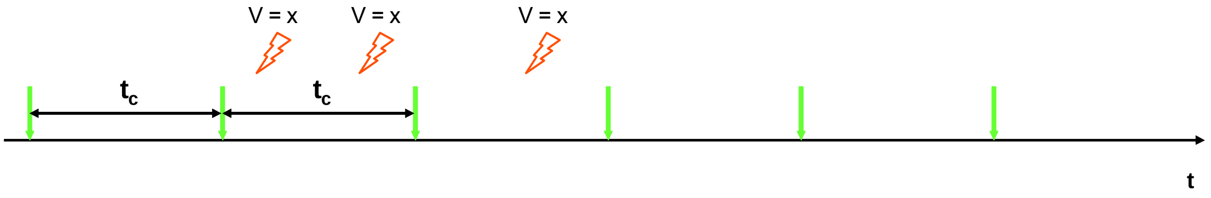

**Figure 4: Use case 1, TM Periodic (without TMS switch)**

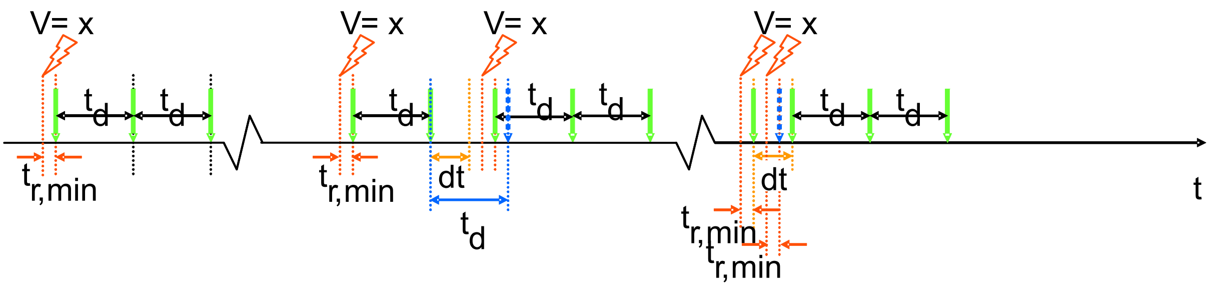

**Figure 5: Use case 2, TM Direct/N-Times, here n = 3 (without TMS switch)**

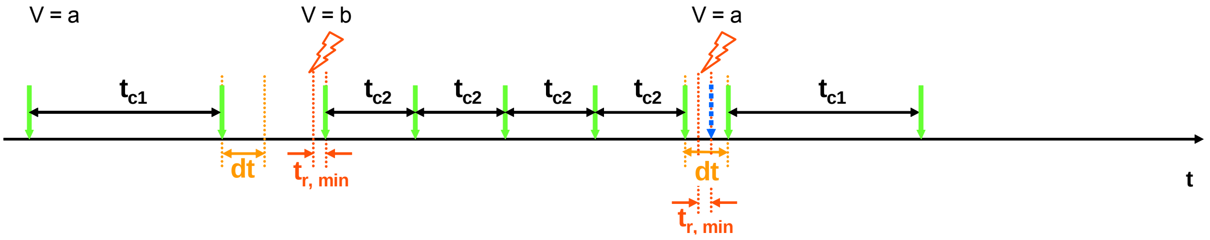

**Figure 6: Use case 3, TM Periodic + Periodic (with TMS switch)**

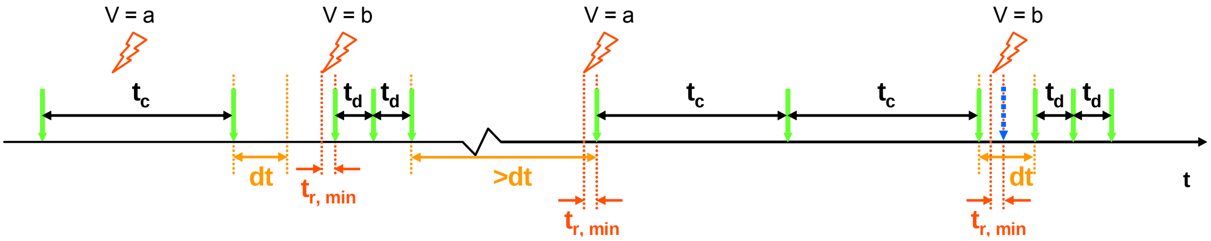

**Figure 7: Use case 4a, TM Periodic + Direct/N-Times, here n = 3 (with TMS switch)**

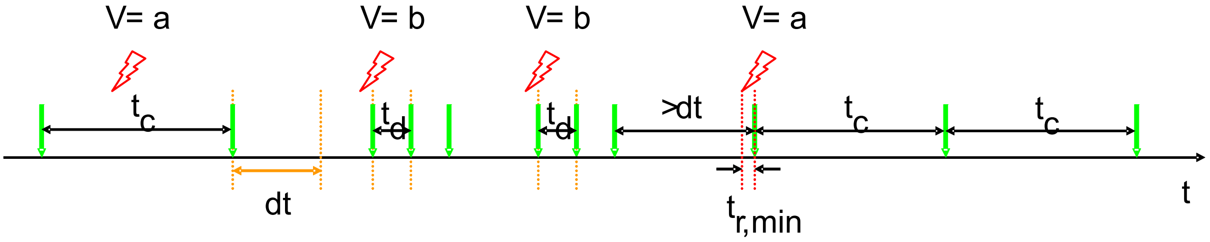

**Figure 8: Use case 4b, TM Periodic + Direct/N-Times, here n = 3 (with TMS switch)**

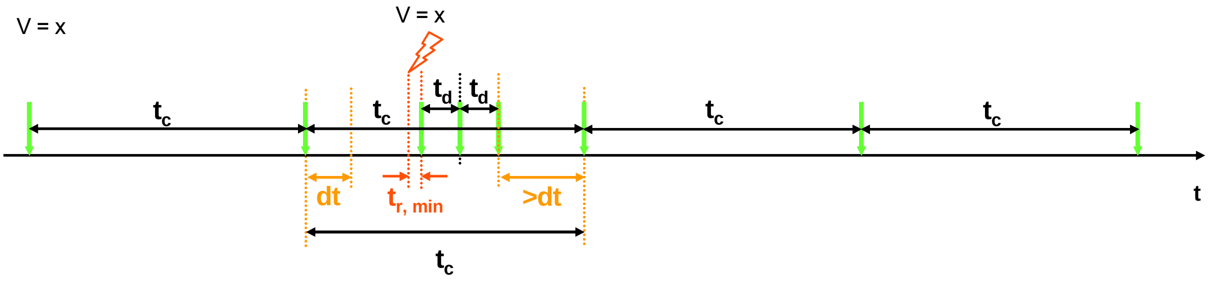

**Figure 9: Use case 5a, TM Mixed, here n = 3 (without TMS switch)**

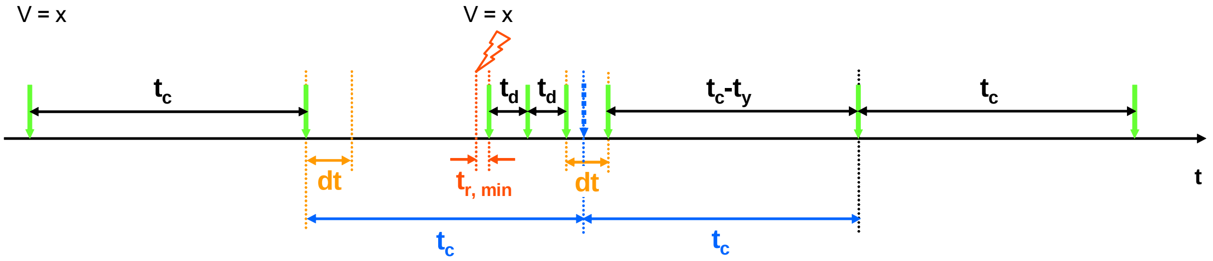

**Figure 10: Use case 5b, TM Mixed, here n = 3 (without TMS switch)**

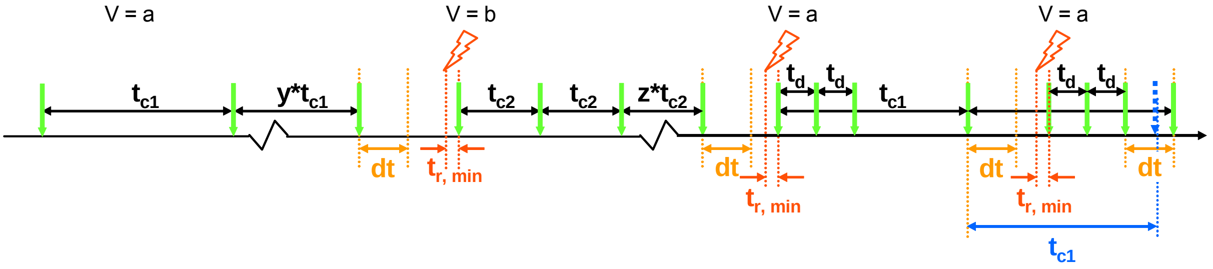

**Figure 11: Use case 6, TM Mixed, here n= 3 + Periodic (with TMS switch)**

### 7.4.4 Signal invalidation mechanism

**COM097:** It shall be possible for the sender side to indicate AUTOSAR COM that it is not able to provide a valid value for a corresponding signal (e.g. sensor is faulty). It shall be possible during configuration to define a *data invalid value* (for configuration see COM501).

#### 7.4.4.1 Transmission of an invalidated signal

**COM099:** AUTOSAR COM shall replace the current value of a signal by the data invalid value if the AUTOSAR software component indicates that it is not able to provide a valid value, see also Com_InvalidateSignal. AUTOSAR COM shall internally perform a Com_SendSignal with the configured data invalid value. For the associated configuration elements see COM501. Therefore the transmission of the data invalid value on the bus is determined by the transfer property and the transmission mode.

**Note:** 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.

**COM286:** By a call of Com_InvalidateShadowSignal AUTOSAR COM shall replace the current value of the signal with the given SignalId within the associated signal group by the configured data invalid value. By this call no send request shall be initiated.

**Note:** Compared to Com_InvalidateSignal, there is no send request associated with Com_InvalidateShadowSignal.

**Note:** The RTE has to call Com_InvalidateShadowSignal for each signal inside a signal group.

Example with 2 signals signal_a and signal_b which belong to group_x:

```
/* invalidate signal a in shadow buffer with configured data invalid value */
Com_InvalidateShadowSignal (signal_a);

/* invalidate signal b in shadow buffer with configured data invalid value */
Com_InvalidateShadowSignal (signal_b);

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

The data invalid values are configured per group signal see COM391.

**Note:** The invalidation of the whole signal group results in a consistent state of the signal group during transmission.

**Note:** 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.

#### 7.4.4.2 Reception of an invalidated signal

**COM100:** Receiver side AUTOSAR COM module shall provide the following configuration options:

- **InvalidNotification**
  The reception of an invalidated signal shall be notified by the function configured by COM315. The normal signal indication shall not take place.
- **ReplaceValue**
  When an invalidated signal is received, this signal value shall be replaced by the init value. Only the normal signal reception indication shall take place (if configured).
- **no invalidation handling**
  In case no ComDataInvalidAction is configured for a (group) signal, the AUTOSAR COM module, shall handle a reception of this signal always like a reception of valid value.

The last option is useful if a system signal has a globally defined invalid value, but shall be handled specially on certain ECUs.

**COM287:** The mechanisms as described in COM100 shall also be applicable for signal groups:

- **InvalidNotification**
  If at least one of the signals inside a signal group is identified as invalid, an invalid notification for the whole signal group shall take place. In this case, the normal signal group indication shall not take place
- **ReplaceValue**
  If at least one of the signals inside a signal group is identified as invalid, all associated signals shall be replaced by their init value. In this case, only the normal signal group reception indication for this signal group shall take place.
- **no invalidation handling**
  In case no ComDataInvalidAction is configured for a signal group, the AUTOSAR COM module, shall handle a reception of this signal group always like a reception of valid signal group.

The last option is useful for example, in case one ECU needs to receive partial values of a signal group, even in case some of the signals are invalidated. In this case the concrete invalidation handling is done in the SWC.

**COM323:** For signals groups it shall be possible to configure an invalid notification for the whole signal group and additionally for each group signal.

The corresponding configuration parameters are ComDataInvalidAction, ComInvalidNotification and ComSignalDataInvalidValue.

**COM558:** 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.

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

**COM559:** 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.

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

### 7.4.5 Handling of I-PDUs

#### 7.4.5.1 Definitions

For the definition of an I-PDU group see [7] and Figure 12. For the configuration of I-PDU groups see COM206 and COM126.

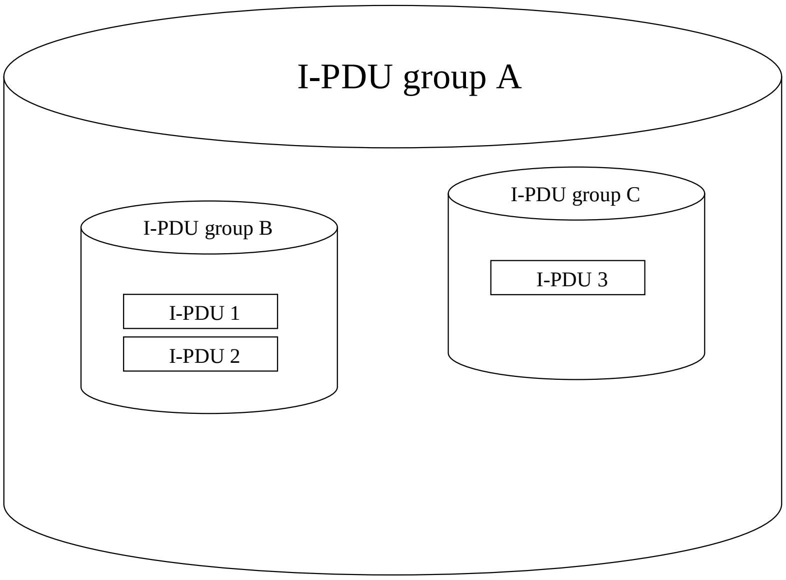

**Figure 12: Grouping of I-PDUs and I-PDU groups**

**COM187:** The number of supported I-PDU groups per ECU shall be limited to a maximum of 64.

### 7.4.5.2 Separate Start/Stop AUTOSAR COM for separate groups of I-PDUs

#### 7.4.5.2.1 Starting of I-PDU groups

By default all I-PDU groups are stopped, see COM444.

**COM085:** AUTOSAR COM shall provide a routine which starts communication per I-PDU group. The name of the routine shall be Com_IpduGroupStart, see COM191. A reference to the I-PDU group shall be transferred in the call to Com_IpduGroupStart.

**COM114:** Starting an I-PDU group shall permit to transmit/receive I-PDUs which belong to the I-PDU group, see also Table 10.

**COM222:** If an I-PDU group is started by Com_IpduGroupStart with parameter Initialize set to true the following attributes (see also Chapter 9.2) for each I-PDU belonging to the group shall be initialized:

- time period and offset attributes of I-PDUs in Periodic or Mixed transmission mode
- the minimum delay time attribute of I-PDUs in Direct/N-Times or Mixed transmission mode
- timeout attributes of I-PDUs for deadline monitoring aspect. All timeout timers (ComSignalFirstTimeoutFactor, ComSignalTimeoutFactor, ComSignalGroup-FirstTimeoutFactor, ComSignalGroupTimeoutFactor) shall restart

- all included update-bits shall be cleared

**COM223:** If an I-PDU group is started, the transmission mode of all associated I-PDUs is determined by the current data content of the I-PDU. E.g. transmission of periodic I-PDUs is started according to their TMS-switch.

**Note:** In order to disable transmission of a group of I-PDUs, the I-PDUs are put into an I-PDU group which is then stopped. This mechanism allows implementing listen-only-mode. Receiving of I-PDUs may be stopped also. Note that an I-PDU group cannot contain a mixture of send I-PDUs and receive I-PDUs.

**COM228:** When an I-PDU group containing an I-PDU is started but one or more signals in that I-PDU have not yet been written to via one of the send-APIs, the initial value is used to determine the transmission mode.

**COM229:** When an I-PDU group containing an I-PDU is started but one or more signals in that I-PDU has been written to via one of the send APIs by the higher layer, the most recently sent values are used to determine the TMS of this I-PDU.

**COM476:** If by starting an I-PDU group (or a contained I-PDU group) an already started I-PDU is started again the further starts shall be ignored.

**COM561:** If an I-PDU is started by a call to Com_IpduGroupStart and the I-PDU contains signals that have deadline monitoring configured (COM183, COM263), the AUTOSAR COM module shall start the deadline monitoring for these signals independently of the value of the initialize parameter.

#### 7.4.5.2.2 Stopping of I-PDU groups

**COM090:** AUTOSAR COM shall provide a routine which supports the possibility of stopping communication per I-PDU group. The name of the routine shall be Com_IpduGroupStop, see COM190. A reference to the I-PDU group shall be transferred in the call to Com_IpduGroupStop.

**COM334:** If an I-PDU-group is stopped, a call of the functions: Com_SendSignal, Com_UpdateShadowSignal, Com_SendSignalGroup, Com_InvalidateSignal and Com_InvalidateShadowSignal shall still update the values in the COM internal buffers, see Table 9.

**Note:** If a signal is written to an I-PDU that belongs to a stopped I-PDU group and this write would trigger the transmission of the I-PDU if it was not stopped then this trigger shall not be stored. After re-starting the corresponding I-PDU group such an old trigger shall not lead to an immediate transmission of the I-PDU.

**COM115:** Stopping I-PDU groups shall disable transmission of I-PDUs which belong to the I-PDU group. Only the Com_TriggerTransmit function is processed because this can not be prohibited by AUTOSAR COM. Deadline monitoring for pending confirmations shall be cancelled. Transmit confirmations shall be ignored.
Stopping I-PDU groups shall disable processing of received I-PDUs which belong to the I-PDU group. Reception deadline monitoring shall be cancelled.

**COM479:** If an I-PDU group is stopped all outstanding not confirmed transmitted signals/ signal groups shall immediately receive an error notification if configured, see COM499 and COM511.

**COM461:** A call to Com_ReceiveSignalGroup shall always copy the last known data (or the init value) of the I PDU to the shadow buffer even if the I-PDU is stopped and COM_SERVICE_NOT_AVAILABLE is returned.

**Note:** If by stopping an I-PDU group (or a contained I-PDU group) an already stopped I-PDU is stopped again this has no effect.

Table 9 gives an overview of the behavior of stopped I-PDU groups:

| ***Behavior on a stopped I PDU group*** | |
|---|---|
| ***Receiver side (RX)*** | ***Transmitter side (TX)*** |
| • disable RX deadline monitoring<br>• no action on a Com_RxIndication to RTE, no storing of the I PDU<br>• 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<br>• normal reaction on Com_ReceiveShadowSignal() | • disable sending<br>• disable TX deadline monitoring<br>• Com_TxConfirmation:<br>if it is for timeout ignore it<br>if it is used by the RTE ignore it.<br>• on a call of Com_SendSignal, Com_UpdateShadowSignal, Com_SendSignalGroup, Com_InvalidateSignal and Com_InvalidateShadowSignal the values in the COM internal buffers are still up-dated but the return code COM_SERVICE_NOT_AVAILABLE is returned<br>• outstanding transmission request (e.g. N-Times) shall be cancelled<br>• normal reaction on Com_TriggerTransmit |
| | ***For periodic (TX)*** |
| | do not send any more |

**Table 9: Behavior of stopped I-PDU-groups**

Table 10 gives an overview of the behavior of started I-PDU groups:

| ***Behavior on a started I-PDU group*** | |
|---|---|
| ***Receiver side (RX)*** | ***Transmitter side (TX)*** |
| • reinitialize timeouts if Initialize==true (ComSignalFirstTimeoutFactor, ComSignalTimeoutFactor, ComSignalGroupFirstTimeoutFactor, ComSignalGroupTimeoutFactor)<br>• normal reaction on Com_RxIndication<br>• normal reaction on Com_ReceiveSignal, Com_ReceiveShadowSignal and Com_ReceiveSignalGroup<br>• deadline monitoring is enabled | • normal reaction on Com_InvalidateSignal, Com_InvalidateShadowSignal, Com_SendSignal, Com_SendSignalGroup, Com_UpdateShadowSignal and Com_SendSignalGroup<br>• no transmission timeout notification until next send<br>• normal reaction on Com_TxConfirmation<br>• normal reaction on Com_TriggerTransmit<br>• deadline monitoring is enabled |
| | ***For periodic (TX)*** |
| | Start at 0 |

**Table 10 Behavior of started I-PDU-groups**

**COM311:** When an I-PDU group containing one or more other I-PDU groups is started the contained I-PDU groups shall also be started. When an I-PDU group containing one or more other I-PDU groups is stopped the contained I-PDU groups shall also be stopped.

**Note:** According to [7] there is only a two level hierarchy of I-PDU groups. That is an I-PDU-group contained in another I-PDU group is not allowed to have any further subgroups. As example see also Figure 12.

With respect to Figure 12 the following examples are given:

- If A is stopped then all I-PDUs are stopped
- If A is started then all I-PDUs are started.
- If A is stopped and then B is started only I-PDU 1 and I-PDU 2 are active.
- If A is started and then B is stopped only I-PDU 3 is active.

### 7.4.5.3 Signal indication (Unpacking of I-PDUs)

**COM298:** In order to support both interrupt-driven and polled system, it shall be configurable when the signal indication takes place. There shall be two configurable signal indication modes:

**Immediate:**
The signal indications/ confirmations are performed in Com_RxIndication/ Com_TxConfirmation

**Deferred:**
Signal indication/ confirmations are deferred for example to a cyclic task

**COM300:** In Immediate mode the signal notification shall be done in Com_RxIndication.

**COM301:** In Deferred mode: First in Com_RxIndication the I-PDU's data shall be copied from the underlying layer into COM. Then Com_RxIndication shall return. Then the signal notification shall be processed asynchronously, for example during the next call to Com_MainFunctionRx.

**Note:** 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 COM119.

**COM574:** Within the function Com_RxIndication, the AUTOSAR COM module shall check the received data length (PduInfoPtr->SduLength) and unpack and notify only completely received signals via ComNotification.

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

**COM575:** Within the function Com_RxIndication, 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.

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

#### 7.4.5.4 Minimum Delay Timer (MDT)

The minimum delay timer shall be defined as in [17].

**Note:** When an I PDU group is started the MDT eventually is re-initialized, depending on the 'Initialize' parameter of Com_IpduGroupStart, see COM222.
Therefore the MDT can be violated by stopping and starting I-PDU groups rapidly.

**Note:** As defined in [17] the MDT is started when the confirmation of the sent I-PDU is received. Therefore exists a time-slot between the send-call and the confirmation where an (erroneous) application could create a burst of I-PDUs by sending rapidly.

**Note:** The behavior of the transmission deadline monitoring timer shall not be affected by any transmission delay caused by the minimum delay time supervision.

### 7.4.6 Deadline Monitoring

Deadline monitoring for signals is defined in [17].

In the context of deadline monitoring a signal group shall be handled like a signal. The deadline monitoring parameters can be configured via ComFirstTimeoutFactor and ComTimeoutFactor in configuration container ComSignal or ComSignalGroup respectively.

The corresponding error notification callback functions can be configured via configuration parameter ComErrorNotification in configuration container ComSignal or ComSignalGroup respectively.

**COM562:** If ComTxIPduMinimumDelayTimeFactor (COM181) of an I PDU is configured greater than 0, 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.

**COM469:** If ComTxIPduMinimumDelayTimeFactor of an I-PDU is configured greater than 0, the AUTOSAR COM module shall (re-)load the already running minimum delay time counter with ComTxIPduMinimumDelayTimeFactor for that I-PDU when Com_TxConfirmation is invoked and the minimum delay time counter of that I-PDU, started at PduR_ComTransmit, is not already elapsed.

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 ComTxIPduMinimumDelayTime 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 COM475 Com_TriggerTransmit does not interfere with the minimum delay timer. Further, the minimum 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.

#### 7.4.6.1 Reception Deadline Monitoring

**COM292:** In the case where reception deadline monitoring is used on signals with update-bits, there shall be a separate reception deadline monitoring for each signal/ signal group with an update-bit. For configuration see COM263 and COM264.

**COM290:** There shall be an I-PDU based reception deadline monitoring for signals without an update-bit.

**COM291:** For all signals/ signal groups without update-bits within the same I-PDU, the smallest configured timeout parameter (ComSignalFirstTimeoutFactor, ComSignalGroupFirstTimeoutFactor / ComSignalTimeoutFactor, ComSignalGroupTimeoutFactor) of the associated signals is chosen as timeout parameter for the reception deadline monitoring of the I-PDU.

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

**COM393:** In case of an Rx-timeout it shall be configurable whether COM replaces the signal/ signal group value with the initial value or keeps it as it is. In case of replacement the *old_value* of the corresponding filter-object (if configured) shall not be replaced.

**Note:** Rx-timeout-indication can be combined and configured separately from COM393.

#### 7.4.6.1.1 En-/Disable Reception Deadline Monitoring

**COM092:** AUTOSAR COM shall provide a routine which supports the possibility to disable reception deadline monitoring for an I-PDU group. The name of the routine shall be Com_DisableReceptionDM. A reference to the I-PDU group shall be transferred in the call to Com_DisableReceptionDM.

Disabling reception deadline monitoring implies that no error indication of reception deadline monitoring expiry is notified to the upper layer for an I-PDU in reception belonging to the concerned I-PDU group.

Disabling reception deadline monitoring shall not stop communication activity for the concerned I-PDU group.

**COM095:** AUTOSAR COM shall provide a routine which supports the possibility to re-enable reception deadline monitoring for an I-PDU group where reception deadline monitoring has been previously disabled. The name of the routine shall be Com_EnableReceptionDM. A reference to the I-PDU group shall be transferred in the call to Com_EnableReceptionDM.

**COM224:** A call to Com_EnableReceptionDM shall reset the reception deadline monitoring timer to ComSignalFirstTimeoutFactor and ComSignalGroupFirstTimeoutFactor if and only if reception deadline monitoring was disabled for the corresponding I-PDU group.

**COM486:** Enabling an already enabled reception deadline monitoring shall have no effect.

Enabling reception deadline monitoring implies that error indications of deadline monitoring expiry are notified to the upper layer for an I-PDU in reception belonging to the concerned I-PDU group.

**COM534:** If Com_EnableReceptionDM is invoked on an I-PDU group containing only/ also Tx-I-PDUs, then the Tx-I-PDUs shall be silently ignored.

**COM225:** Disabling an already disabled reception deadline monitoring shall have no effect.

### 7.4.6.2 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.

**COM481:** Transmission deadline monitoring shall not distinguish between signals with Pending or Triggered transfer property. It shall be performed for all signals and all transmission modes as defined for signals with Triggered transfer property in [17] if it is configured.

**COM445:** If different ComTimeoutFactor parameters of the associated signals/ signal groups of an I-PDU are configured the smallest value shall be used as timeout parameter for the transmission deadline monitoring of the I-PDU.

**Note:** Transmission deadline monitoring should only be configured in COM if the lower layer supports the generation of transmit confirmations. Otherwise the transmission deadline monitoring would always notify a transmission error.

**Note:** Transmission deadline monitoring is not supported for transmission mode None.

**Note:** The PDUR does not support transmit confirmations for PDUs that are configured to be fanned out by the PDUR to multiple receivers.

**Note:** 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 COM345 and COM520.

#### 7.4.6.2.1 Clarification of the OSEK COM specification

The following requirement COM304 states the behavior of the transmission deadline monitoring in the Mixed transmission mode defined in [17] more precisely.

**COM304:** If the transmission does not occur, i.e. if there is no confirmation of the I-PDU’s transmission by the underlying layer, the time-out occurs and the application shall be notified using the appropriate notification mechanism.

**Note:** If the transmission deadline monitoring timer runs out there shall be an error notification regardless of the reason. Even if it was postponed because of the MDT or it was filtered out via an I-PDU callout.

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

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

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

**Note:** As defined in [17] the monitoring timer shall 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 □nstanctively.

**COM307:** In Direct/N-Times transmission mode it must be ensured that all n requests can be made within the configured time period, see Chapter 7.4.3.5.

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

**COM308:** In Direct/N-Times transmission mode with n > 0 the timer shall only be cancelled when n-confirmations are received or if another send request for this I-PDU is initiated.

**Note:** If the timer is cancelled after the n-th confirmation the transmission was successful and then the transmission confirmation is send to the upper layer. See also COM305.

## 7.5 Map Complex Data Types to I-PDUs – Signal Groups

**COM042:** To support the AUTOSAR concept of complex data types the AUTOSAR COM layer provides signal groups (see Chapter 2). A signal group shall be transmitted and received atomically; therefore it provides data consistency for complex data types.

**COM043**: Signal groups are configured statically. For each signal group a symbolic name shall be provided. See COM345 and COM044 for the configuration details.

**COM047:** The atomicity of a signal group shall be achieved by means of a shadow buffer mechanism, i.e. the upper layer uses the group signals in the shadow buffer. If the shadow buffer needs to be synchronized with the I-PDU this is triggered explicitly by the upper layer. Synchronization shall be performed atomically.

### 7.5.1 Initialization

**COM484**: The shadow buffer of a signal group on sender-side shall be initialized by a call to Com_Init.

**Note:** Since it is not suspected that a well-formed SWC tries to read a group signal before a call to Com_ReceiveSignalGroup COM484 applies to the sender side only.

### 7.5.2 Transmission

**COM049:** A group signal in the shadow buffer shall be updated by the call of the service Com_UpdateShadowSignal.

**COM050:** If Com_SendSignalGroup is called for the signal group the AUTOSAR COM layer shall copy the shadow buffer atomically to the I-PDU buffer.

Example with 2 group signals signal_a and signal_b which belong to group_x:

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

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

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

### 7.5.3 Reception

**COM051:** If Com_ReceiveSignalGroup is called for the signal group the AUTOSAR COM layer shall copy the data atomically from the I-PDU buffer to the shadow buffer.

**COM052:** A group signal shall be received from the shadow buffer by means of the service Com_ReceiveShadowSignal.

Example with 2 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_ReceiveShadowSignal (signal_a, &a);

/* copy b from shadow buffer */
Com_ReceiveShadowSignal (signal_b, &b);
```

### 7.5.4 Notifications

It is only possible to configure the signal and error notifications (on sender and receiver side) for the whole signal group.

**COM053:** The notifications shall be sent to the RTE if a whole signal group has been sent to / received or detected to be invalid from the lower layer. See COM345 and COM520 for the configuration details.

### 7.5.5 Collection of the attributes of a signal group

Table 11 gives an overview of the attributes of a signal group:

| ***Attribute*** | ***Per group signal*** | ***Per signal group*** |
|---|---|---|
| Update-bit | No | Yes, associated on the whole group (see 7.7) |
| Signal Notification (sender side) | No | Yes |
| Signal Notification (receiver side) | No | Yes |
| Error Notification (sender side) | No | Yes |
| Error Notification (receiver side) | No | Yes |
| Invalid Notification (receiver side) | Yes (see COM323) | Yes (see COM323) |
| Data access (receiver side) | Yes, with ReceiveSignal API (see 0) | Yes, via shadow buffer (see 8.3.2.5 and 8.3.2.6) |
| Data access (sender side) | No | Yes, via shadow buffer<br>(see 0 and 8.3.2.4) |
| Data Filtering (receiver side) | No (see 7.2.3) | No |
| Data Filtering (sender side) | No | No |
| TMS on sender side | Each signal, according to TMS selection definition.<br>(see 7.4.3.3) | No |

**Table 11: Attributes of signal groups**

## 7.6 Interface between AUTOSAR COM and the lower layer (PDU-Router)

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

The interaction diagram in Chapter 9.1 shows the interaction between the PDU Router and AUTOSAR COM.

A detailed description can be found in the API chapter see Com_RxIndication, Com_TxConfirmation and Com_TriggerTransmit.

**COM138:** When AUTOSAR COM wants to send out an I-PDU, AUTOSAR COM shall use the PduR_ComTransmit function.

## 7.7 Signal status information

### 7.7.1 Identify if a signal is updated by the sender

**COM054:** 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, AUTOSAR COM shall support *update-bits*.

The update-bit shall indicate whether upper layers on sender-side have updated a signal value before lower layers have fetched the I-PDU containing that signal for transmission or before AUTOSAR COM has handed over the I-PDU to lower layers for transmission.

**Note:** Update-bits are not allowed if Direct/N-Times transmission mode with n>1 is used (see COM310).

**COM055**: By configuration on sender- and on receiver-side, it shall be possible to add separately for each signal and/or separately for each signal group at most one additional bit (= update-bit). The update-bit is not part of the signal or signal group itself and shall only be used by AUTOSAR COM itself. The update-bit shall not be visible to or accessible by the AUTOSAR Software Component.

**COM056**:´The position of the update-bit shall be configurable. For configuration parameter see COM257.

**COM057**: A signal/ signal group and the corresponding update-bit shall always be part of the same I-PDU.

**COM059**: The interpretation of the update-bit shall be as follows:

| ***Update-BIT*** | |
|---|---|
| 0 | cleared |
| 1 | set |

**Table 12 update-bit interpretation**

If the value of the update-bit is set, data has been updated; if the value of the update-bit is cleared it has not been updated.

#### 7.7.1.1 Sender Side

The initialization of the update-bit is defined in the Chapter 7.4.1 by COM117.

**COM061:** If upper layers update the value of a signal by calling the AUTOSAR COM API Com_SendSignal, the update-bit for this signal shall be set. For signal groups, the update-bit shall be set, if the upper layers call the AUTOSAR COM API Com_SendSignalGroup.

**COM062:** After an I-PDU is sent to lower layers and no synchronous error is returned by the lower layer the update-bits of all signals and signal groups belonging to the I-PDU sent shall be cleared.

#### 7.7.1.2 Receiver Side

**COM324:** On receiver-side, if there is an update-bit attached to a signal/signal group, AUTOSAR COM shall only process this signal (i.e. filter, notification, signal based, byte swapping), if the signal has been updated. If the signal has not been updated AUTOSAR COM shall discard the signal.

**Note:** If the signals has not been updated the signal will not be routed via the signal gateway. It will only be discarded.

**Remark:** If the upper layer reads a signal with an associated cleared update-bit, the init value or the last received value is returned.

**COM067:** A signal/ signal group shall be interpreted as *updated* if the signal has an update-bit attached and the value of the update-bit is set.

For the behavior of deadline monitoring on signals with update-bits, see Chapter 7.4.5.4.

## 7.8 Callouts

As stated in COM013 *Network-order message callout* and *CPU-order message callout* are not supported in AUTOSAR COM. The only callout method in AUTOSAR COM therefore is the I-PDU callout. AUTOSAR COM supports I-PDU callouts on sender and on receiver side.

### 7.8.1 I-PDU Callout

**COM381:** In an I-PDU callout other COM APIs than Com_TriggerIPDUSend shall not be called.

**COM346:** The I-PDU-Callout API shall be defined as:

boolean <IPDU_CalloutName>(PduIdType ID, uint8* ipduD)

where <IPDU_CalloutName> has to be substituted with the concrete I-PDU callout name.

**Note:** As specified in OSEK COM if the I-PDU callout returns false the I-PDU shall not be processed any further.

**COM347:** It shall be possible to configure separate I-PDU callout function for each I-PDU. Therefore the I-PDU callout function shall be configurable per I-PDU if used.

For configuration see COM387.

**COM395:** When Com_TriggerTransmit is called, COM shall ignore the return value from the I-PDU callout (if configured).

## 7.9 Signal Gateway

The signal gateway is an integrated part of COM. The signal gateway can't be accessed by any external modules, except the cyclic task call.

The signal gateway is working on (group) signals and signal groups.

**COM376:** The signal gateway only supports static routing: All routes shall be used independent of the content of the signals and signal groups to be routed. The destination of a signal or signal group shall only depend on the name of the received signal respectively signal group.

**COM377:** The signal gateway shall copy the value of signals respectively signal groups to be routed to the signals respectively signal groups for transmission according to configuration.

**COM358:** It shall be possible to route a signal/ signal group from one source signal/ signal group to zero (no signal gateway functionality) or more destinations (1:n).

### 7.9.1 Dealing with signals

**COM357:** COM shall forward signals to be routed from received I-PDUs to transmit I-PDUs. For configuration see configuration container ComGwMapping (COM544).

**COM360:** If the endianness of a received signal to be routed differs from the endianness of a related destination signal, its endianness shall be converted using the applicable COM mechanisms.

### 7.9.2 Dealing with signal groups

**COM361:** COM shall forward signal groups to be routed from received I-PDUs to transmit I-PDUs. For configuration see configuration container ComGwMapping (COM544).

**COM383:** Signal groups shall be routed in an atomic manner. I.e. it must be guaranteed that the data within the signal group is transferred as one consistent set of data during the whole routing operation.

**COM362:** It shall be possible to change the endianness of signals contained in signal groups to be routed by signal gateway. The signal gateway shall use the applicable COM mechanisms to do so.

**COM464:** All non-opaque group signals within a signal group shall have the same endianness.

### 7.9.3 Routing of out timed signals and signal groups

**COM568:** The AUTOSAR COM module's signal gateway shall route signal and signal groups even if any configured reception deadline monitoring timeout expired.

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.

#### 7.9.3.1 Handling of update-bits

**COM569:** If both, the received signal/ signal group and the destination signal/ signal group have an update-bit (ComUpdateBitPostition) 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.

**COM570:** If the received signal/ signal group and the destination signal/ signal group have an update-bit (ComUpdateBitPostition) 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.

**COM571:** 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.

**COM572:** 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.

**COM573:** 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.

### 7.9.4 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 signal gateway from interrupt routines.

**COM359:** The functions of the signal gateway shall be executed during a separate function call only. During this function call the signal gateway checks received and to be routed signals and signal groups and forwards them from the related receive I-PDUs to the related transmit I-PDUs (see COM400).

**COM466:** Within Com_MainFunctionRouteSignals the evaluation of transfer properties and transmission mode must be performed in the following sequence (see also Figure 3):

Copy all gated signals from the source to the target I-PDUs
Evaluate the TMC of all gated signals
Evaluate the TMS for the target I-PDUs
For any target I-PDU containing gated signals with Triggered transfer property send it according to its transmission mode

**COM539:** An I-PDU shall be sent out at most once while one call to Com_MainFunctionRouteSignals.

## 7.10 Error classification

### 7.10.1 Development Errors

**COM024:** All input parameters shall be checked for validity during development. The parameter check shall be not contained in the production code. For the configuration of this feature see COM028.

**COM442:** When a development error is detected the function Det_ReportError of the development error tracer shall be called with:

the COM moduleID (see COM417)
the COM instanceID
the service ID of the COM API the error is detected (see Com_ServiceIdType)
the error ID as defined in Table 13

| *Type of development error* | *Related error code* | *Value [hex]* |
|---|---|---|
| API service called with wrong parameter | COM_E_PARAM | 0x01 |
| Error code if any other API service is called before COM was initialized with Com_Init or after a call to Com_Deinit | COM_E_UNINIT | 0x02 |

**Table 13: Mapping of COM development error IDs**

### 7.10.2 Production Errors

Actually no production errors are defined in AUTOSAR COM. If production errors will be defined in later versions AUTOSAR COM shall report them directly to the DEM.

### 7.10.3 Return Codes

AUTOSAR COM 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 14.

**COM459:** Return codes of AUTOSAR Com shall be defined according Table 14.

| *Name* | *Description* | *Type* | *Value* | *Defined in* |
|---|---|---|---|---|
| E_OK | the service has been accepted | #define | 0x00 | Std_Types.h |
| COM_SERVICE_NOT_AVAILABLE | the service is currently not available e.g. the corresponding I-PDU group is stopped<br>(or a development error has been detected) | #define | 0x80 | Com.h |
| COM_TIMEOUT | a timeout has occurred | #define | 0x81 | Com.h |

**Table 14: Mapping of AUTOSAR COM return codes**

## 7.11 Error handling

**COM428:** If not stated otherwise AUTOSAR COM will ignore all errors from the underlying communication layer.

**Note:** AUTOSAR COM supervises the communication with deadline monitoring if configured. The specific error codes from the underlying layer therefore can be ignored. In case of update-bits this error codes are handled see COM062.

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

**COM396:** A signal can be configured to have filtering, data invalidation and notification. The order of execution (if configured) is:

1) Data invalidation
2) Filtering
3) Notification

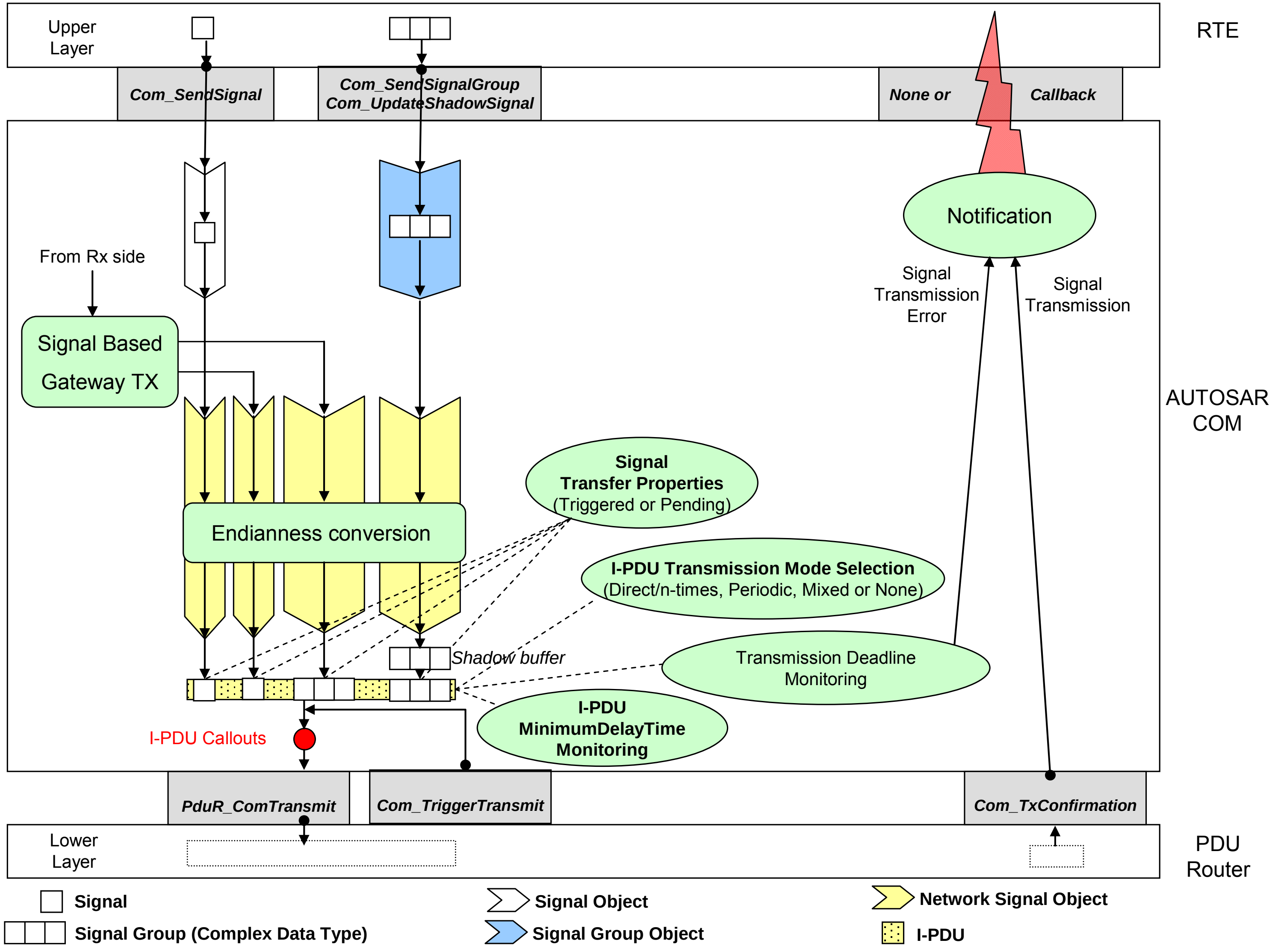

**Figure 13 AUTOSAR COM interaction model for transmission**

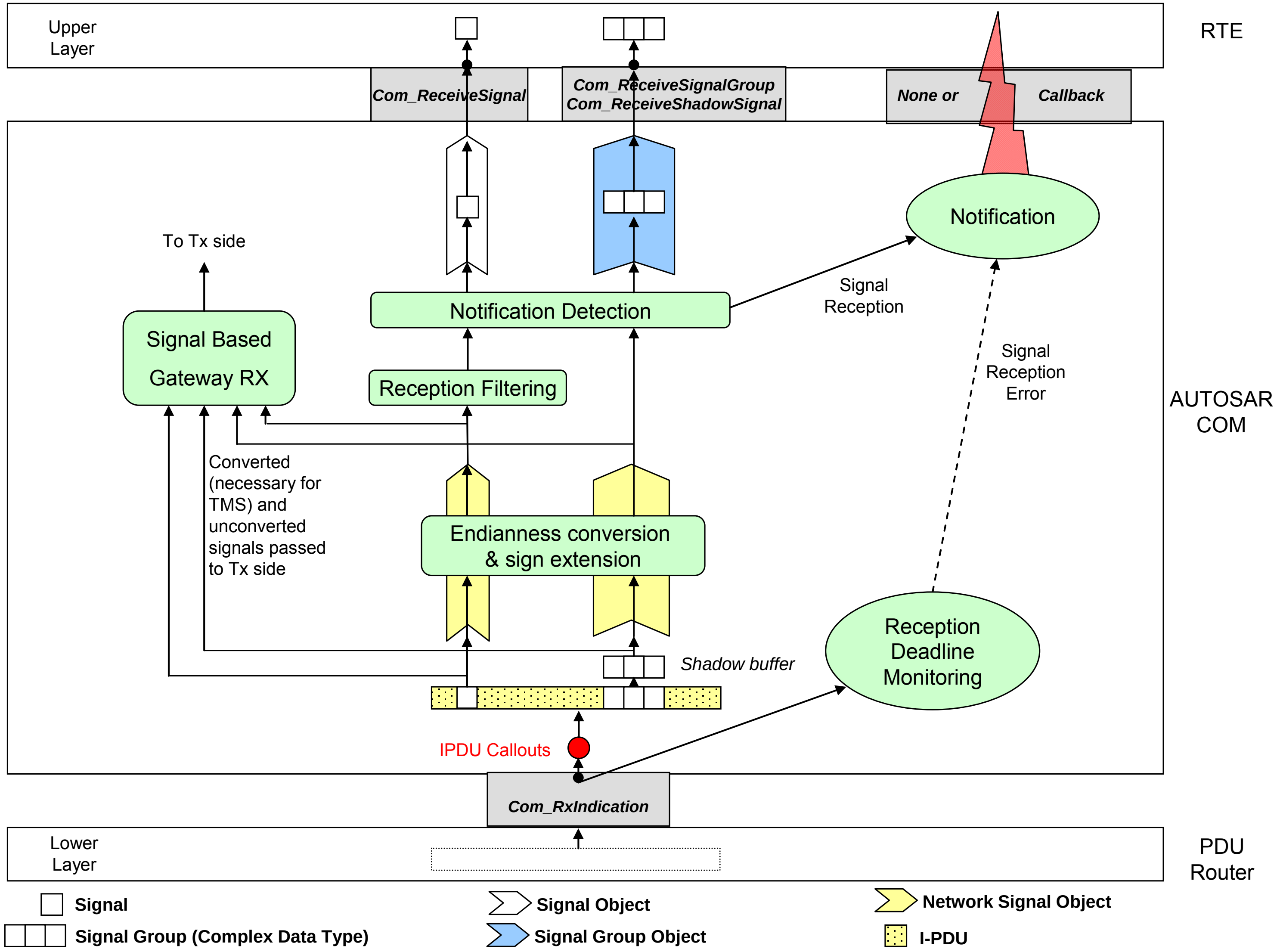

**Figure 14 AUTOSAR COM interaction model for reception**

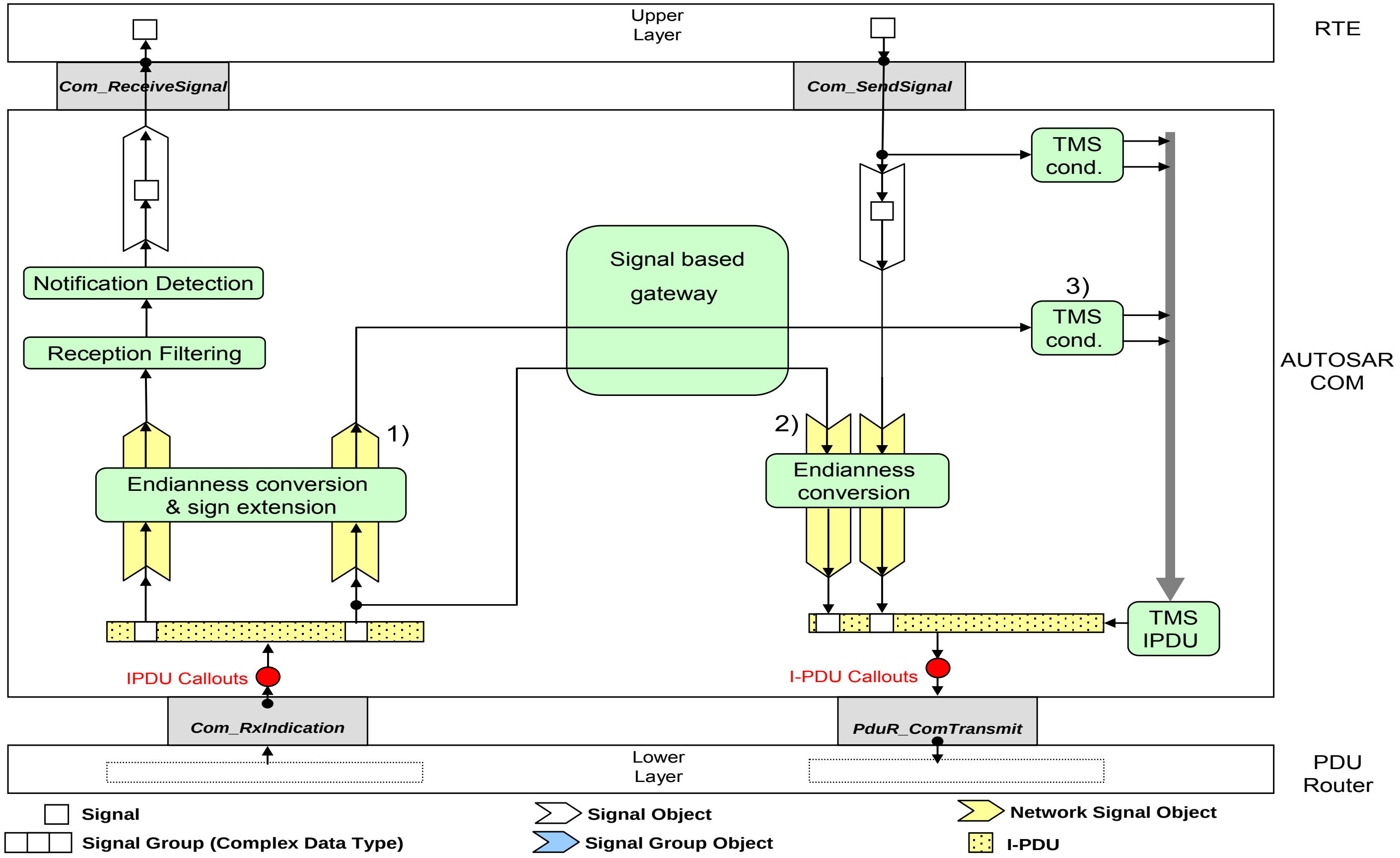

**Figure 15: AUTOSAR COM-interaction model for integrated Signal Gateway**

The endianness conversion and sign extension on receiver side is 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. If a gated signal should not always trigger the associated IPDU to be sent out, one of the configured transmission modes must be None.

# 8 API specification

## 8.1 Imported types

### 8.1.1 PduIdType

The definition of the PduIdType can be found in [2].

### 8.1.2 Std_ReturnType

The definition of the Std_ReturnType can be found in [5].

### 8.1.3 Std_VersionInfoType

The definition of the Std_VersionInfoType can be found in [5].

## 8.2 Type definitions

### 8.2.1 Com_StatusType

| | | |
|---|---|---|
| ***Name:*** | Com_StatusType | |
| ***Type:*** | Enumeration | |
| ***Range:*** | 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. |
| | COM_INIT | The AUTOSAR COM Module is initialized and usable. |
| ***Description:*** | This is a status value returned by the API service Com_GetStatus(). | |

### 8.2.2 Com_SignalIdType

| | | |
|---|---|---|
| ***Name:*** | Com_SignalIdType | |
| ***Type:*** | uint16 | |
| ***Range:*** | 0..<SignalIdmax> | Zero-based integer number |
| ***Description:*** | AUTOSAR COM signal object identifier. | |

### 8.2.3 Com_SignalGroupIdType

| | | |
|---|---|---|
| ***Name:*** | Com_SignalGroupIdType | |
| ***Type:*** | uint16 | |
| ***Range:*** | 0..<SignalGroupIdmax> | Zero-based integer number |
| ***Description:*** | AUTOSAR COM signal group object identifier. | |

### 8.2.4 Com_PduGroupIdType

| | | |
|---|---|---|
| ***Name:*** | Com_PduGroupIdType | |
| ***Type:*** | uint8 | |
| ***Range:*** | 0..<PduGroupId-max> | Zero-based integer number |
| ***Description:*** | AUTOSAR COM PDU group object identifier. | |

### 8.2.5 Com_ServiceIdType

| | | | |
|---|---|---|---|
| ***Name:*** | Com_ServiceIdType | | |
| ***Type:*** | uint8 | | |
| ***Range:*** | COMServiceId_MainFunctionRouteSignals | 0x1A | -- |
| | COMServiceId_MainFunctionTx | 0x19 | -- |
| | COMServiceId_MainFunctionRx | 0x18 | -- |
| | COMServiceId_TriggerIPDUSend | 0x17 | -- |
| | COMServiceId_InvalidateShadowSignal | 0x16 | -- |
| | COMServiceId_TxConfirmation | 0x15 | -- |
| | COMServiceId_RxIndication | 0x14 | -- |
| | COMServiceId_TriggerTransmit | 0x13 | -- |
| | COMServiceId_Error_<Name1>_<Name2>Macros | 0x12 | -- |
| | COMServiceId_ErrorGetServiceId | 0x11 | -- |
| | COMServiceId_InvalidateSignal | 0x10 | -- |
| | COMServiceId_ReceiveShadowSignal | 0x0F | -- |
| | COMServiceId_ReceiveSignalGroup | 0x0E | -- |
| | COMServiceId_SendSignalGroup | 0x0D | -- |
| | COMServiceId_UpdateShadowSignal | 0x0C | -- |
| | COMServiceId_ReceiveSignal | 0x0B | -- |
| | COMServiceId_SendSignal | 0x0A | -- |
| | COMServiceId_GetVersionInfo | 0x09 | -- |
| | COMServiceId_GetConfigurationId | 0x08 | -- |
| | COMServiceId_GetStatus | 0x07 | -- |
| | COMServiceId_DisableReceptionDM | 0x05 | -- |
| | COMServiceId_EnableReceptionDM | 0x06 | -- |
| | COMServiceId_IpduGroupStop | 0x04 | -- |
| | COMServiceId_InvalidateSignalGroup | 0x1B | -- |
| | COMServiceId_IpduGroupStart | 0x03 | -- |
| | COMServiceId_DeInit | 0x02 | -- |
| | COMServiceId_Init | 0x01 | -- |
| ***Description:*** | Unique identifier of an AUTOSAR COM service. Example: COMServiceId_SendSignal 0x0A. | | |

### 8.2.6 Com_ConfigType

| | | |
|---|---|---|
| ***Name:*** | Com_ConfigType | |
| ***Type:*** | Structure | |
| ***Range:*** | implementation specific | The content of the initialization data structure is implementation specific |
| ***Description:*** | This is the type of the data structure containing the initialization data for COM. | |

## 8.3 Function definitions

**COM320:** If a function is marked as non-reentrant the caller of that function shall ensure that this function must not be called while it is running.

**COM321:** Non-reentrant functions do not have to check if they are called reentrant.

**COM434:** It is allowed to use macros instead of functions where source code is used and runtime is critical.

### 8.3.1 Start up and control services

#### 8.3.1.1 Com_Init

**COM432:**

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

**COM433:** If the config parameter does not correspond to a valid configuration then the development error COM_E_PARAM is generated. The behavior of AUTOSAR COM is unspecified until a correct call to Com_Init is made.

**Caveats:** Com_Init shall not pre-empt any COM function. The rest of the system must guarantee that Com_Init is not called in such a way.

#### 8.3.1.2 Com_DeInit

**COM130:**

| | |
|---|---|
| ***Service name:*** | Com_DeInit |
| ***Syntax:*** | void Com_DeInit(<br>) |
| ***Service ID[hex]:*** | 0x02 |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | Non Reentrant |
| ***Parameters (in):*** | None |
| ***Parameters (in-out):*** | None |
| ***Parameters (out):*** | None |

| | |
|---|---|
| ***Return value:*** | None |
| ***Description:*** | 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.<br>By a call to ComDeInit COM is put into an not initialized state. |

**Caveats:** Com_DeInit shall not pre-empt any COM function. The rest of the system must guarantee that Com_DeInit is not called in such a way.

### 8.3.1.3 Com_IpduGroupStart

**COM191:**

| | | |
|---|---|---|
| ***Service name:*** | Com_IpduGroupStart | |
| ***Syntax:*** | void Com_IpduGroupStart(<br>  Com_PduGroupIdType IpduGroupId,<br>  boolean Initialize<br>) | |
| ***Service ID[hex]:*** | 0x03 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant for different I-PDU groups. Non reentrant for the same I-PDU group. | |
| ***Parameters (in):*** | IpduGroupId | Id of I-PDU group to be started |
| | Initialize | flag to request initialization of the data in the I-PDUs of this I-PDU group |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | Starts a preconfigured I-PDU group. For example, cyclic I-PDUs will be sent out cyclically after the call of Com_IpduGroupStart(). See Chapter 7.4.5 for details.<br><br>If Initialize is true all I-PDUs of the I-PDU group shall be (re-)initialized before the I-PDU group is started. That is they shall behave like after a start-up of COM, for example the old_value of the filter objects and shadow buffers of signal groups have to be (re-)initialized. | |

**Caveats:** A call to Com_IpduGroupStart shall not be interrupted by another call to Com_IpduGroupStart or a call to Com_IpduGroupStop. Note that this function is not only called by the COMM but also from other modules e.g. diagnosis.

**Configuration:** An I PDU group must be configured before this call. See COM206 and COM341 for details.

### 8.3.1.4 Com_IpduGroupStop

**COM190:**

| | | |
|---|---|---|
| ***Service name:*** | Com_IpduGroupStop | |
| ***Syntax:*** | void Com_IpduGroupStop(<br>  Com_PduGroupIdType IpduGroupId<br>) | |
| ***Service ID[hex]:*** | 0x04 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant for different I-PDU groups. Non reentrant for the same I-PDU group. | |
| ***Parameters (in):*** | IpduGroupId | Id of I-PDU group to be stopped |
| ***Parameters (in-*** | None | |

| | |
|---|---|
| ***out):*** | |
| ***Parameters (out):*** | None |
| ***Return value:*** | None |
| ***Description:*** | Stops a preconfigured I-PDU group. For example, cyclic I-PDUs will be stopped after the call of Com_IpduGroupStop(). See Chapter 7.4.5 for details. |

**Caveats:** A call to Com_IpduGroupStop shall not be interrupted by another call to Com_IpduGroupStop or a call to Com_IpduGroupStart. Note that this function is not only called by the COMM but also from other modules e.g. diagnosis.

**Configuration:** An I PDU group must be configured before this call. See COM206 and COM341 for details.

#### 8.3.1.5 Com_DisableReceptionDM

**COM192:**

| | | |
|---|---|---|
| ***Service name:*** | Com_DisableReceptionDM | |
| ***Syntax:*** | void Com_DisableReceptionDM(<br>Com_PduGroupIdType IpduGroupId<br>) | |
| ***Service ID[hex]:*** | 0x05 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant for different I-PDU groups. Non reentrant for the same I-PDU group. | |
| ***Parameters (in):*** | IpduGroupId | Id of I-PDU group where reception DM shall be disabled. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | Disables the reception deadline monitoring for the I-PDUs within the given I-PDU group. | |

**Configuration:** An I PDU group must be configured before this call. See COM206 and COM341 for details.

#### 8.3.1.6 Com_EnableReceptionDM

**COM193:**

| | | |
|---|---|---|
| ***Service name:*** | Com_EnableReceptionDM | |
| ***Syntax:*** | void Com_EnableReceptionDM(<br>Com_PduGroupIdType IpduGroupId<br>) | |
| ***Service ID[hex]:*** | 0x06 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant for different I-PDU groups. Non reentrant for the same I-PDU group. | |
| ***Parameters (in):*** | IpduGroupId | Id of I-PDU group where reception DM shall be enabled. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | Enables the reception deadline monitoring for the I-PDUs within the given I-PDU group. | |

**Configuration:** An I-PDU group must be configured before this call. See COM206 and COM341 for details.

### 8.3.1.7 Com_GetStatus

**COM194:**

| | | |
|---|---|---|
| ***Service name:*** | Com_GetStatus | |
| ***Syntax:*** | Com_StatusType Com_GetStatus(<br><br>) | |
| ***Service ID[hex]:*** | 0x07 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant | |
| ***Parameters (in):*** | None | |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | Com_StatusType | COM_UNINIT: AUTOSAR COM is not initialized and not usable<br>COM_INIT: AUTOSAR COM is initialized and usable |
| ***Description:*** | Returns the status of the AUTOSAR COM module. | |

### 8.3.1.8 Com_GetConfigurationId

**COM375:**

| | | |
|---|---|---|
| ***Service name:*** | Com_GetConfigurationId | |
| ***Syntax:*** | uint32 Com_GetConfigurationId(<br><br>) | |
| ***Service ID[hex]:*** | 0x08 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non Reentrant | |
| ***Parameters (in):*** | None | |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | uint32 | configured ConfigurationID, see COM394 |
| ***Description:*** | Provides the unique identifier of the configuration. | |

**Configuration:** The provided Identification shall be set during configuration process and can't be changed by COM.

### 8.3.1.9 Com_GetVersionInfo

**COM426:**

| | |
|---|---|
| ***Service name:*** | Com_GetVersionInfo |
| ***Syntax:*** | void Com_GetVersionInfo(<br>Std_VersionInfoType* versioninfo<br>) |
| ***Service ID[hex]:*** | 0x09 |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | Non Reentrant |

| | | |
|---|---|---|
| ***Parameters (in):*** | None | |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | versioninfo | Pointer to where to store the version information of this module. |
| ***Return value:*** | None | |
| ***Description:*** | Returns the version information of this module. | |

**COM424:** This service returns the version information of this module. The version information includes:

1) Module ID
2) Vendor ID
3) Vendor specific version numbers (BSW00407).

**COM425:** This function shall be pre compile time configurable On/Off by the configuration parameter: COM_VERSION_INFO_API

**Note:** If source code for caller and called of this function is available, this function should be realized as a macro. The macro should be defined in the modules header file.

Configuration: see COM026

### 8.3.2 Communication services

#### 8.3.2.1 Com_SendSignal

**COM197:**

| | | |
|---|---|---|
| ***Service name:*** | Com_SendSignal | |
| ***Syntax:*** | uint8 Com_SendSignal(<br>Com_SignalIdType SignalId,<br>const void* SignalDataPtr<br>) | |
| ***Service ID[hex]:*** | 0x0a | |
| ***Sync/Async:*** | Asynchronous | |
| ***Reentrancy:*** | Non Reentrant for the same signal. Reentrant for different signals. | |
| ***Parameters (in):*** | SignalId | Id of signal to be sent. |
| | SignalDataPtr | Reference to the signal data to be transmitted. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | uint8 | E_OK: service has been accepted<br>COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) |
| ***Description:*** | The service Com_SendSignal updates the signal object identified by SignalId with the signal referenced by the SignalDataPtr parameter.<br><br>If the signal has the Triggered transfer property, the update is followed by immediate transmission of the I-PDU associated with the signal except when the signal is packed into an I-PDU with Periodic transmission mode; in this case, no transmission is initiated by the call to this service.<br>If the signal has the Pending transfer property, no transmission is caused by the update. | |

#### 8.3.2.2 Com_ReceiveSignal

**COM198:**

| | | |
|---|---|---|
| ***Service name:*** | Com_ReceiveSignal | |
| ***Syntax:*** | uint8 Com_ReceiveSignal(<br>Com_SignalIdType SignalId,<br>void* SignalDataPtr<br>) | |
| ***Service ID[hex]:*** | 0x0b | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non Reentrant for the same signal. Reentrant for different signals. | |
| ***Parameters (in):*** | SignalId | Id of signal to be received. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | SignalDataPtr | Reference to the signal data in which to store the re-ceived data. |
| ***Return value:*** | uint8 | E_OK: service has been accepted<br>COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) |
| ***Description:*** | The service Com_ReceiveSignal updates the signal referenced by SignalDataPtr with the data in the signal object identified by SignalId. | |

#### 8.3.2.3 Com_UpdateShadowSignal

**COM199:**

| | | |
|---|---|---|
| ***Service name:*** | Com_UpdateShadowSignal | |
| ***Syntax:*** | void Com_UpdateShadowSignal(<br>Com_SignalIdType SignalId,<br>const void* SignalDataPtr<br>) | |
| ***Service ID[hex]:*** | 0x0c | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non Reentrant for the same group signal. Reentrant for different group signals. | |
| ***Parameters (in):*** | SignalId | Id of group signal to be updated. |
| | SignalDataPtr | Reference to the group signal data to be updated. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | The service Com_UpdateShadowSignal updates a group signal with the data, referenced by SignalDataPtr. The update of the group signal data is done in the shadow buffer, not in the I-PDU. To send out the shadow buffer, Com_SendSignalGroup has to be called.<br><br>Sign extension and byte swapping are performed as the group signal is inserted into the shadow buffer. | |

**Configuration:** A signal group must be configured before this call. See COM345 for details.

#### 8.3.2.4 Com_SendSignalGroup

**COM200:**

| | |
|---|---|
| ***Service name:*** | Com_SendSignalGroup |
| ***Syntax:*** | uint8 Com_SendSignalGroup( |

| | | |
|---|---|---|
| | Com_SignalGroupIdType SignalGroupId<br>) | |
| ***Service ID[hex]:*** | 0x0d | |
| ***Sync/Async:*** | Asynchronous | |
| ***Reentrancy:*** | Non Reentrant for the same group signal. Reentrant for different group signals. | |
| ***Parameters (in):*** | SignalGroupId | Id of signal group to be send. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | uint8 | E_OK: service has been accepted<br>COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) |
| ***Description:*** | The service Com_SendSignalGroup copies the content of the associated shadow buffer to the associated I-PDU. Prior to this call, all group signals should be updated in the shadow buffer by the call of Com_UpdateShadowSignal. | |

**Configuration:** A signal group must be configured before this call. See COM345 for details.

### 8.3.2.5 Com_ReceiveSignalGroup

**COM201:**

| | | |
|---|---|---|
| ***Service name:*** | Com_ReceiveSignalGroup | |
| ***Syntax:*** | uint8 Com_ReceiveSignalGroup(<br>Com_SignalGroupIdType SignalGroupId<br>) | |
| ***Service ID[hex]:*** | 0x0e | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non Reentrant for the same group signal. Reentrant for different group signals. | |
| ***Parameters (in):*** | SignalGroupId | Id of signal group to be received. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | uint8 | E_OK: service has been accepted<br>COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) |
| ***Description:*** | The service Com_ReceiveSignalGroup copies the received signal group from the I-PDU to the shadow buffer. After this call, the group signals could be copied from the shadow buffer to the upper layer by a call of Com_ReceiveShadowSignal. | |

**Configuration:** A signal group must be configured before this call. See COM345 for details.

### 8.3.2.6 Com_ReceiveShadowSignal

**COM202:**

| | |
|---|---|
| ***Service name:*** | Com_ReceiveShadowSignal |
| ***Syntax:*** | void Com_ReceiveShadowSignal(<br>Com_SignalIdType SignalId,<br>void* SignalDataPtr<br>) |
| ***Service ID[hex]:*** | 0x0f |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | Non Reentrant for the same group signal. Reentrant for different group signals. |

| | | |
|---|---|---|
| ***Parameters (in):*** | SignalId | Id of group signal to be received. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | SignalDataPtr | Reference to the group signal data in which to store the received data. |
| ***Return value:*** | None | |
| ***Description:*** | The service Com_ReceiveShadowSignal updates the group signal which is referenced by SignalDataPtr with the data in the shadow buffer. The data in the shadow buffer should be updated before the call of Com_ReceiveShadowSignal by a call of the service Com_ReceiveSignalGroup. | |

### 8.3.2.7 Com_InvalidateSignal

**COM203:**

| | | |
|---|---|---|
| ***Service name:*** | Com_InvalidateSignal | |
| ***Syntax:*** | uint8 Com_InvalidateSignal(<br>Com_SignalIdType SignalId<br>) | |
| ***Service ID[hex]:*** | 0x10 | |
| ***Sync/Async:*** | Asynchronous | |
| ***Reentrancy:*** | Non Reentrant for the same signal. Reentrant for different signals. | |
| ***Parameters (in):*** | SignalId | Id of signal to be invalidated. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | uint8 | E_OK: service has been accepted<br>COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) |
| ***Description:*** | Sender side AUTOSAR Software Component indicates via the RTE to AUTOSAR COM that it is not able to provide a valid value for the corresponding signal (e.g. sensor is faulty). After invaliding the actual signal data a Com_SendSignal is performed internally, for details see COM097 and COM099. | |

**Configuration:** For processing, a Data Invalid Value must have been configured, see COM501.

### 8.3.2.8 Com_InvalidateShadowSignal

**COM288:**

| | | |
|---|---|---|
| ***Service name:*** | Com_InvalidateShadowSignal | |
| ***Syntax:*** | void Com_InvalidateShadowSignal(<br>Com_SignalIdType SignalId<br>) | |
| ***Service ID[hex]:*** | 0x16 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non Reentrant for the same signal. Reentrant for different signals. | |
| ***Parameters (in):*** | SignalId | Id of signal to be sent. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | Sender side AUTOSAR Software Component indicates via the RTE to AUTOSAR COM that it is not able to provide a valid value for the corresponding group signal, | |

| | |
|---|---|
| | e.g. sensor is faulty. See also COM047.<br>The RTE has to call this function for each group signal of a signal group.<br>To send out the invalidated signal group Com_SendSignalGroup must be called separately. |

**Configuration:** For processing, a ComSignalDataInvalidValue must have been configured.

### 8.3.2.9 Com_InvalidateSignalGroup

**COM557:**

| | | |
|---|---|---|
| ***Service name:*** | Com_InvalidateSignalGroup | |
| ***Syntax:*** | uint8 Com_InvalidateSignalGroup(<br>    Com_SignalGroupIdType SignalGroupId<br>) | |
| ***Service ID[hex]:*** | 0x1b | |
| ***Sync/Async:*** | Asynchronous | |
| ***Reentrancy:*** | Non Reentrant for the same signal group. Reentrant for different signal groups. | |
| ***Parameters (in):*** | SignalGroupId | Id of signal group to be invalidated. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | uint8 | E_OK: service has been accepted<br>COM_SERVICE_NOT_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) |
| ***Description:*** | Sender side AUTOSAR Software Component indicates via the RTE to AUTOSAR COM that it is not able to provide a valid value for the corresponding signal group.<br>After invaliding the actual signal group data a Com_SendSignalGroup is performed internally, | |

### 8.3.2.10 Com_TriggerIPDUSend

**COM348:**

| | | |
|---|---|---|
| ***Service name:*** | Com_TriggerIPDUSend | |
| ***Syntax:*** | void Com_TriggerIPDUSend(<br>    PduIdType ComTxPduId<br>) | |
| ***Service ID[hex]:*** | 0x17 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non Reentrant | |
| ***Parameters (in):*** | ComTxPduId | The I-PDU-ID if the I-PDU that shall be triggered for sending |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | By a call to Com_TriggerIPDUSend the I-PDU with the given ID is triggered for transmission. | |

**COM388:** When an I-PDU is sent out because of this API only the minimum delay time has to be taken into account. That is postpone transmissions if necessary and

reset the minimum delay timer in case of transmissions. All other transmission mode related parameters like N-Times shall not be taken into account.

**COM492:** If an I-PDU triggered by Com_TriggerIPDUSend has a configured I-PDU-callout this I-PDU-Callout shall also be called.

**Caveats:** Shall only be used from within an I PDU callout.

## 8.4 Callback notifications

### 8.4.1 Com_TriggerTransmit

**COM001:**

| | | |
|---|---|---|
| ***Service name:*** | Com_TriggerTransmit | |
| ***Syntax:*** | Std_ReturnType Com_TriggerTransmit(<br>PduIdType ComTxPduId,<br>PduInfoType* PduInfoPtr<br>) | |
| ***Service ID[hex]:*** | 0x13 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non reentrant for the same PDU-ID. Reentrant for different PDU-ID. | |
| ***Parameters (in):*** | ComTxPduId | ID of AUTOSAR COM I-PDU that is requested to be transmitted by AUTOSAR COM. |
| ***Parameters (in-out):*** | PduInfoPtr | Contains a pointer to a buffer (SduDataPtr) to where the SDU shall be copied to. On return, the service will indicate the length of the copied SDU data in SduLength. |
| ***Parameters (out):*** | None | |
| ***Return value:*** | Std_ReturnType | E_OK: SDU has been copied and SduLength indicates the number of copied bytes.<br>E_NOT_OK: No SDU has been copied. SduLength has not been set. |
| ***Description:*** | This function is called by the lower layer when an AUTOSAR COM I-PDU shall be transmitted. Within this function, AUTOSAR COM shall copy the contents of its I-PDU transmit buffer to the L-PDU buffer given by SduDataPtr. | |

**COM475:** Com_TriggerTransmit is not interfered by the I-PDU minimum delay time and shall not reset the minimum delay timer, see COM181.

**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.
This function is also used by the FlexRay Interface for requesting PDUs to be sent in static part (synchronous to the FlexRay global time).
Once the I PDU has been successfully sent by the lower layer (PDU Router), the lower layer must call Com_TxConfirmation.

**Caveats:** This function might be called in interrupt context.

### 8.4.2 Com_RxIndication

**COM123:**

| | |
|---|---|
| ***Service name:*** | Com_RxIndication |

| | | |
|---|---|---|
| ***Syntax:*** | void Com_RxIndication(<br>PduIdType ComRxPduId,<br>const PduInfoType* PduInfoPtr<br>) | |
| ***Service ID[hex]:*** | 0x14 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non reentrant for the same PDU-ID. Reentrant for different PDU-ID. | |
| ***Parameters (in):*** | ComRxPduId | ID of AUTOSAR COM I-PDU that has been received. Identifies the data that has been received.<br>Range: 0..(maximum number of I-PDU IDs received by AUTOSAR COM) - 1 |
| | PduInfoPtr | Contains the length (SduLength) of the received I-PDU and a pointer to a buffer (SduDataPtr) containing the I-PDU. |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | This function is called by the lower layer after an I-PDU has been received. | |

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

### 8.4.3 Com_TxConfirmation

**COM124:**

| | | |
|---|---|---|
| ***Service name:*** | Com_TxConfirmation | |
| ***Syntax:*** | void Com_TxConfirmation(<br>PduIdType ComTxPduId<br>) | |
| ***Service ID[hex]:*** | 0x15 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non reentrant for the same PDU-ID. Reentrant for different PDU-ID. | |
| ***Parameters (in):*** | ComTxPduId | ID of AUTOSAR COM I-PDU that has been transmitted.<br>Range: 0..(maximum number of I-PDU IDs transmitted by AUTOSAR COM) - 1 |
| ***Parameters (in-out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | This function is called by the lower layer after the PDU has been transmitted on the network.<br>A confirmation that is received for an I-PDU that does not require a confirmation is silently discarded. | |

**Caveats:** This function might be called in interrupt context (e.g. from transmit interrupt).

## 8.5 Scheduled Functions

### 8.5.1 Com_MainFunctionRx

**COM398:**

| | |
|---|---|
| ***Service name:*** | Com_MainFunctionRx |
| ***Syntax:*** | void Com_MainFunctionRx(<br><br>) |
| ***Service ID[hex]:*** | 0x18 |
| ***Timing:*** | FIXED_CYCLIC |
| ***Description:*** | This function shall perform the processing of the AUTOSAR COM receive processing that are not directly initiated by the calls from the RTE and PDU-R.<br><br>A call to Com_MainFunctionRx shall simply return if COM was not previously initialized with a call to Com_Init. |

Configuration: see COM186.

### 8.5.2 Com_MainFunctionTx

**COM399:**

| | |
|---|---|
| ***Service name:*** | Com_MainFunctionTx |
| ***Syntax:*** | void Com_MainFunctionTx(<br><br>) |
| ***Service ID[hex]:*** | 0x19 |
| ***Timing:*** | FIXED_CYCLIC |
| ***Description:*** | This function shall perform the processing of the AUTOSAR COM transmission activities that are not directly initiated by the calls from the RTE and PDU-R.<br><br>A call to Com_MainFunctionTx shall simply return if COM was not previously initialized with a call to Com_Init. |

Configuration: see COM186.

### 8.5.3 Com_MainFunctionRouteSignals

**COM400:**

| | |
|---|---|
| ***Service name:*** | Com_MainFunctionRouteSignals |
| ***Syntax:*** | void Com_MainFunctionRouteSignals(<br><br>) |
| ***Service ID[hex]:*** | 0x1a |
| ***Timing:*** | FIXED_CYCLIC |
| ***Description:*** | Calls the signal gateway part of COM to forward received signals to be routed. The insertion of this call is necessary for decoupling receive interrupts and signal gateway tasks.<br><br>A call to Com_MainFunctionRouteSignals shall simply return if COM was not previously initialized with a call to Com_Init. |

**Caveat:** The time between to consecutive calls (perhaps the related task/thread cycle) affects directly the signal gateway latency.

**Configuration:** A cyclic task/thread to call this function cyclical shall be configured. The cycle of this task/thread directly affects the latency of the signal gateway, see COM186.

## 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.

| ***API function*** | ***Module*** | ***Description*** |
|---|---|---|
| PduR_ComTransmit | PDU-R | Request the transmission of I-PDU. |
| Dem_ReportErrorStatus | DEM | Routine to report production relevant error events by event ID. |

### 8.6.2 Optional Interfaces

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

| ***API function*** | ***Module*** | ***Description*** | ***Configuration parameter (description see Chapter 10)*** |
|---|---|---|---|
| Det_ReportError | DET | Development error notification | COM_CONFIGURATION_USE_DET |

### 8.6.3 Configurable interfaces

**Caveats:** 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.

**COM468:**

| | |
|---|---|
| ***Service name:*** | Com_CbkTxAck |
| ***Syntax:*** | void Com_CbkTxAck(<br><br>) |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | don't care |
| ***Parameters (in):*** | None |
| ***Parameters (in-out):*** | None |
| ***Parameters (out):*** | None |
| ***Return value:*** | None |
| ***Description:*** | This callback represents notification class 2 of [17]. It is called immediately after successful transmission of the I-PDU containing the message.<br><br>Com_CbkTxAck is called on sender side only. IIt can be configured for signals and |

| | |
|---|---|
| | signal groups.<br>Com_CbkTxAck corresponds to Rte_COMCbkTAck_<sn> or Rte_COMCbkTAck_<sg> repectively.<br><br>For configuration of the callback function names, see COM498. |

**COM491:**

| | |
|---|---|
| ***Service name:*** | Com_CbkTxErr |
| ***Syntax:*** | void Com_CbkTxErr(<br><br>) |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | don't care |
| ***Parameters (in):*** | None |
| ***Parameters (in-out):*** | None |
| ***Parameters (out):*** | None |
| ***Return value:*** | None |
| ***Description:*** | 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.<br><br>Com_CbkTxErr is called on sender side only. This callback function corresponds to Rte_COMCbkTErr_<sn> or Rte_COMCbkTErr_<sg> respectively.<br><br>For configuration of the callback function name, see COM499. |

**COM554:**

| | |
|---|---|
| ***Service name:*** | Com_CbkTxTOut |
| ***Syntax:*** | void Com_CbkTxTOut(<br><br>) |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | don't care |
| ***Parameters (in):*** | None |
| ***Parameters (in-out):*** | None |
| ***Parameters (out):*** | None |
| ***Return value:*** | None |
| ***Description:*** | 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.<br><br>Com_CbkTxTOut is called on sender side only. It can be configured for signals and signal groups. This callback function corresponds to Rte_COMCbkTOut_<sn> or Rte_COMCbkTOut_<sg> respectively.<br><br>For configuration of the callback function names, see COM552. |

**COM555:**

| | |
|---|---|
| ***Service name:*** | Com_CbkRxAck |
| ***Syntax:*** | void Com_CbkRxAck( |

| | |
|---|---|
| | ) |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | don't care |
| ***Parameters (in):*** | None |
| ***Parameters (in-out):*** | None |
| ***Parameters (out):*** | None |
| ***Return value:*** | None |
| ***Description:*** | This callback represents notification class 1 of [17]. It is called immediately after the message has been stored in the receiving message object.<br><br>Com_CbkRxAck is called on receiver side only . It can be configured for signals and signal groups.<br>Com_CbkRxAck corresponds to Rte_COMCbk_<sn> or Rte_COMCbk_<sg> repectively.<br><br>For configuration of the callback function names, see COM498. |

**COM556:**

| | |
|---|---|
| ***Service name:*** | Com_CbkRxTOut |
| ***Syntax:*** | void Com_CbkRxTOut(<br><br>) |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | don't care |
| ***Parameters (in):*** | None |
| ***Parameters (in-out):*** | None |
| ***Parameters (out):*** | None |
| ***Return value:*** | None |
| ***Description:*** | 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.<br><br>Com_CbkRxTOut is called on receiver side only. It can be configured for signals and signal groups. This callback function corresponds to Rte_COMCbkTOut_<sn> or Rte_COMCbkTOut_<sg> respectively.<br><br>For configuration of the callback function names, see COM552. |

**COM536:**

| | |
|---|---|
| ***Service name:*** | Com_CbkInv |
| ***Syntax:*** | void Com_CbkInv(<br><br>) |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | don't care |
| ***Parameters (in):*** | None |
| ***Parameters (in-out):*** | None |
| ***Parameters (out):*** | None |
| ***Return value:*** | None |
| ***Description:*** | This callback function corresponds to COM100. It is called after reception of an invalid signal or signal group respectively. |

| | |
|---|---|
| | Com_CbkInv is called on receiver side only . It can be configured for signals, group signals and signal groups.<br>This callback function corresponds to Rte_COMCbkInv_<sn> (for signals and group signals) and Rte_COMCbkInv_<sg> respectively.<br><br>For configuration of the callback function names, see COM315. |

**Note:** The naming conventions for the RTE callback routines are defined in [13] in Chapter "Naming convention of callbackRoutineName".

**Note:** AUTOSAR COM uses no direct interface of RTE beside the callback functions.

# 9 Sequence diagrams

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

## 9.1 Interface between AUTOSAR COM and the lower layer (PDU Router)

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

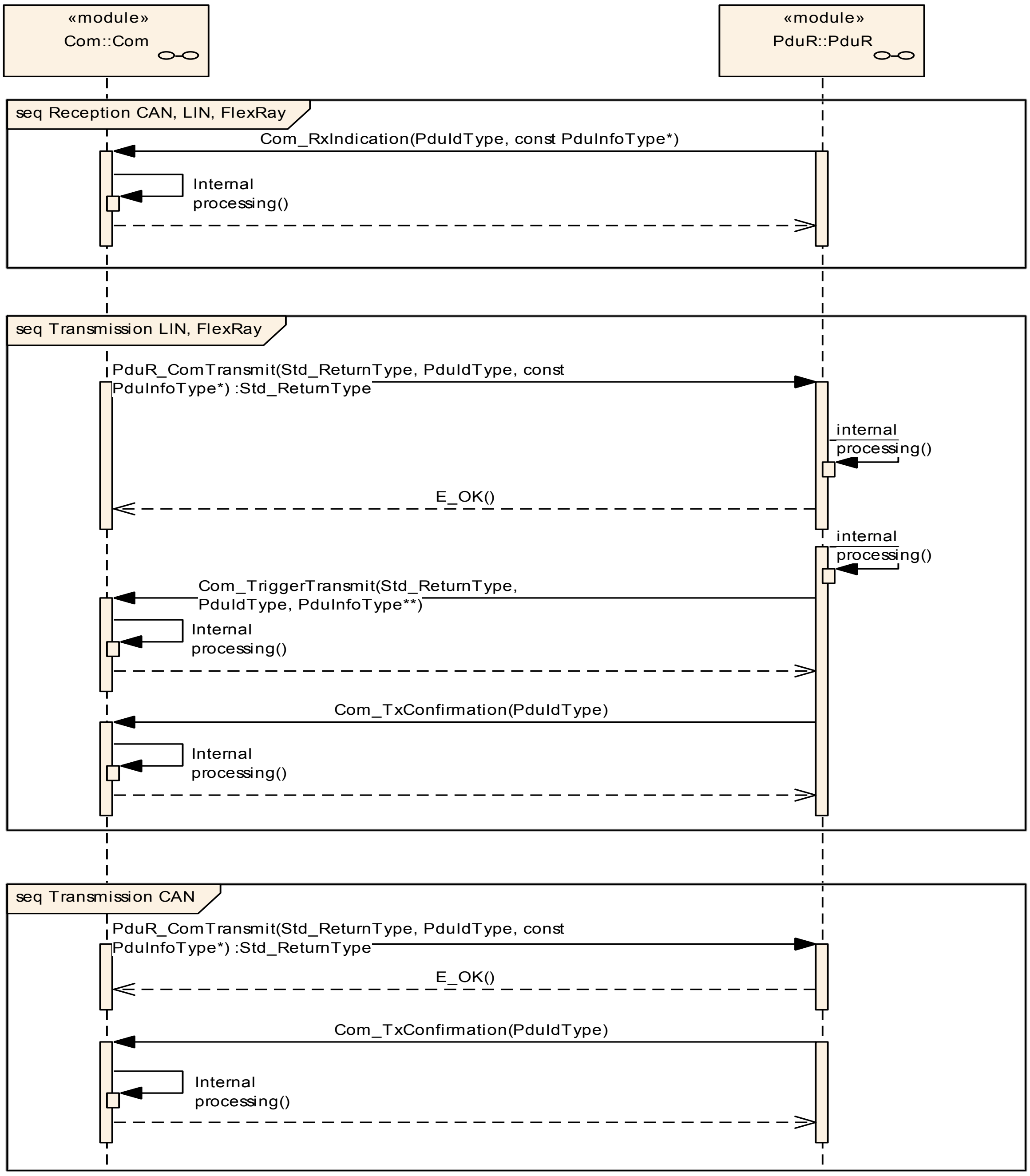
«module»
Com::Com
«module»
PduR::PduR
seq Reception CAN, LIN, FlexRay
Com_RxIndication(PduIdType, const PduInfoType*)
Internal processing()
seq Transmission LIN, FlexRay
PduR_ComTransmit(Std_ReturnType, PduIdType, const PduInfoType*) :Std_ReturnType
internal processing()
E_OK()
internal processing()
Com_TriggerTransmit(Std_ReturnType, PduIdType, PduInfoType**)
Internal processing()
Com_TxConfirmation(PduIdType)
Internal processing()
seq Transmission CAN
PduR_ComTransmit(Std_ReturnType, PduIdType, const PduInfoType*) :Std_ReturnType
E_OK()
Com_TxConfirmation(PduIdType)
Internal processing()

**Figure 16: Interactions between AUTOSAR COM and the PDU router**

## 9.2 Confirmation handling between PDUR, COM and RTE

The following chart shows the confirmation handling with respect to the two different IPDU-processing modes. (See also Chapter 7.4.5.3.)

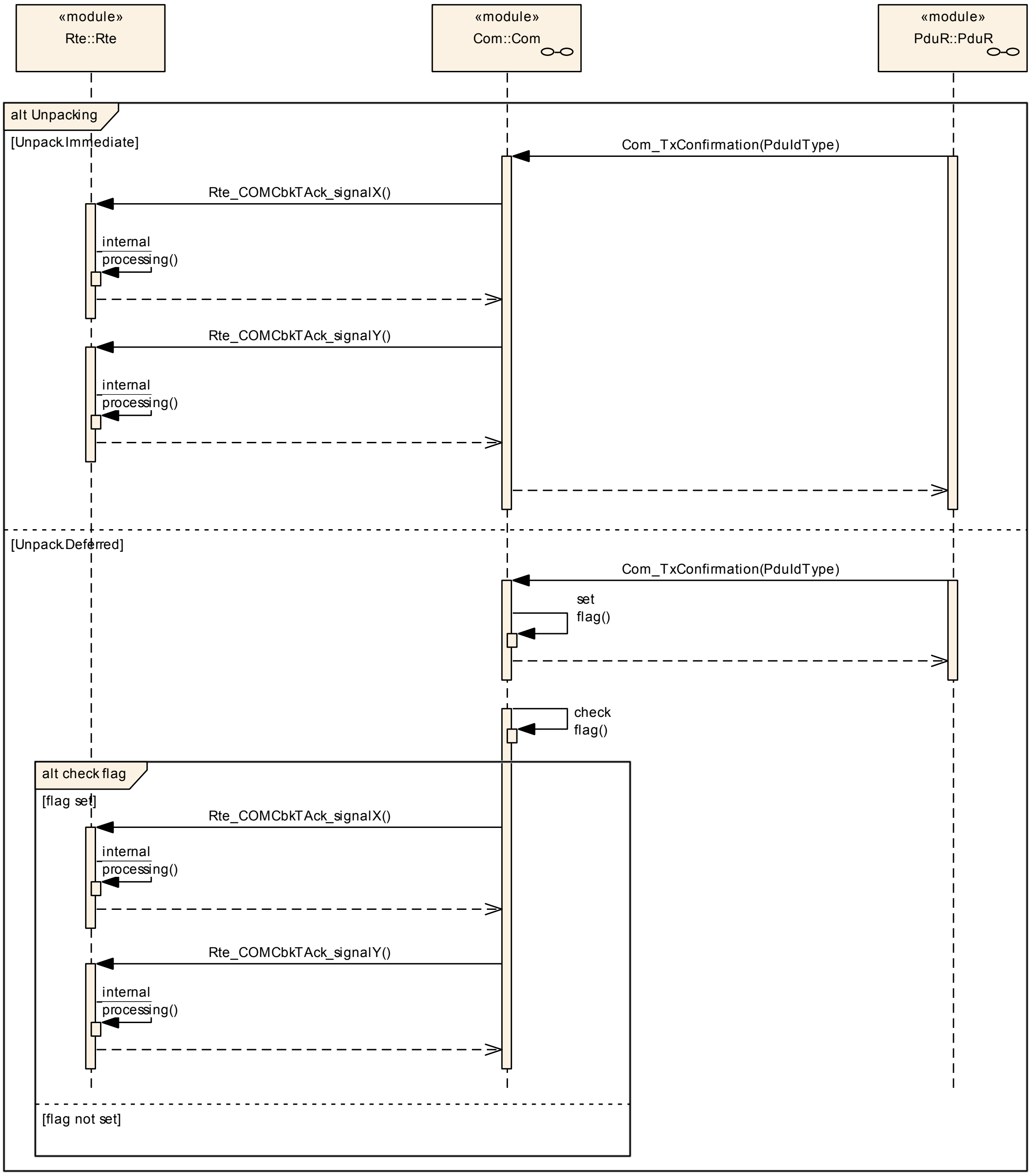

**Figure 17: Confirmation handling between PDUR, COM and RTE**

## 9.3 Indication handling between PDUR, COM and RTE

The following chart shows the indication handling with respect to the two different unpacking modes. (See also Chapter 7.4.5.3.)

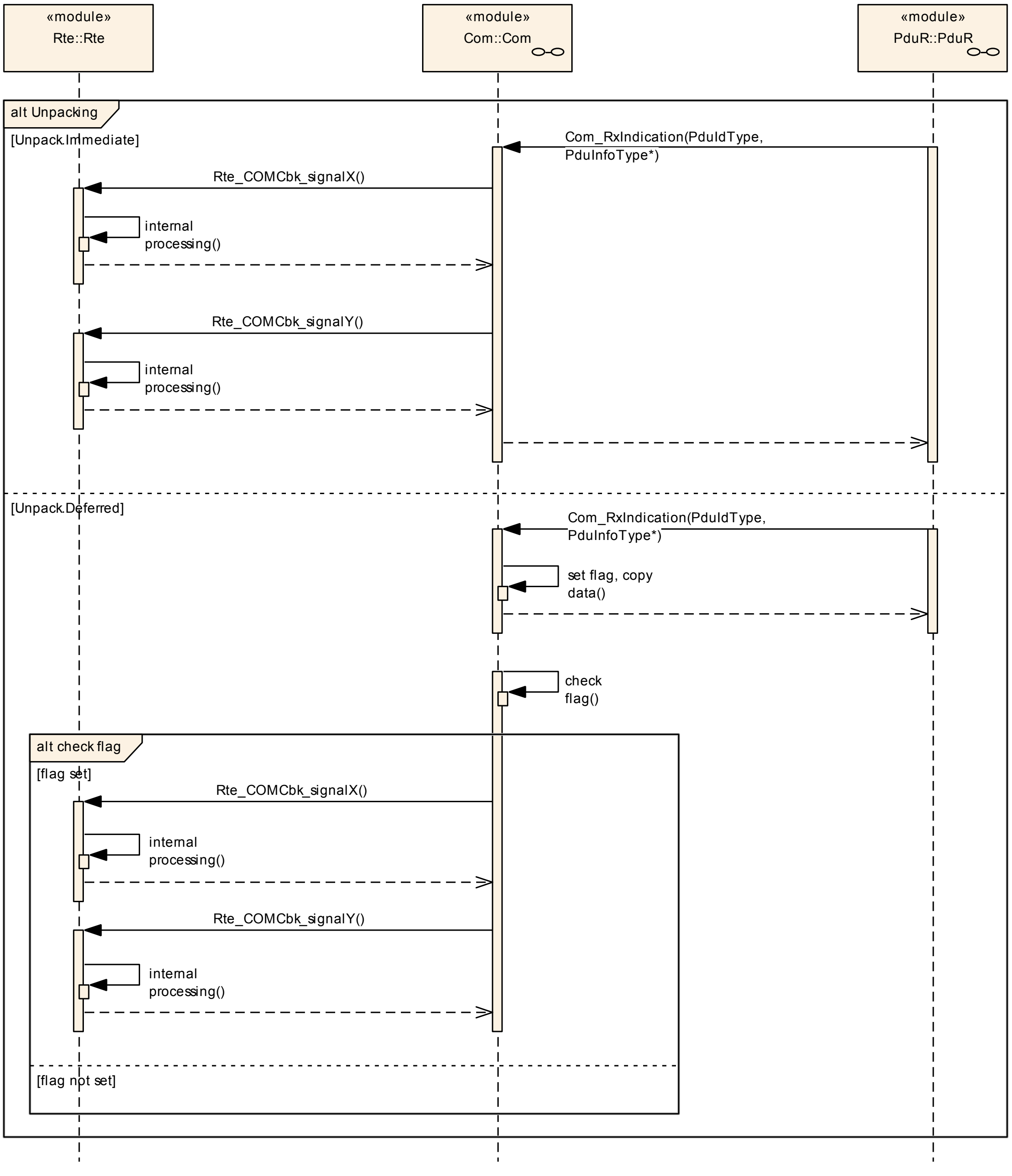

**Figure 18: Indication handling between PDUR, COM and RTE**

# 10 Configuration specification

## 10.1 How to read this chapter

In addition to this section, it is highly recommended to read the documents:

- AUTOSAR Layered Software Architecture [1]
- AUTOSAR ECU Configuration Specification [14]
  This document describes the AUTOSAR configuration methodology and the AUTOSAR configuration meta model in detail.

The following is only a short survey of the topic and it will not replace the ECU Configuration Specification document.

### 10.1.1 Configuration and configuration parameters

Configuration parameters define the variability of the generic part(s) of an □onfiguretation of a module. This means that only generic or configurable module implementation can be adapted to the environment (software/ hardware) in use during system and/ or ECU configuration.

The configuration of parameters can be achieved at different times during the software process: before compile time, before link time or after build time. In the following, the term ***configuration class*** (of a parameter) shall be used in order to refer to a specific configuration point in time.

**COM006:** The configuration parameters are based on [18]. All parameters have to be stored in an XML format.

### 10.1.2 Containers

Containers structure the set of configuration parameters. This means: *all* □onfiguretion parameters are kept in containers. (sub-) containers can reference (sub-) containers. It is possible to assign a multiplicity to these references. The multiplicity then defines the possible number of instances of the contained parameters.

## 10.2 Containers and configuration parameters

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

### 10.2.1 Variants

Currently three configuration variants for AUTOSAR COM are defined.

**COM374:** All configuration sets shall be identifiable by a unique identifier, see COM394.

#### 10.2.1.1 VARIANT-PRE-COMPILE

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.

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

#### 10.2.1.2 VARIANT-LINK-TIME

VARIANT-LINK-TIME includes mainly link-time and some pre-compile configurable parameters. All parameters defined below as link-time configurable shall be □onfigureable 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.

#### 10.2.1.3 VARIANT-POST-BUILD

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.

### 10.2.2 Configuration of the AUTOSAR COM Layer

For an overview of the COM Configuration see Figure 19:

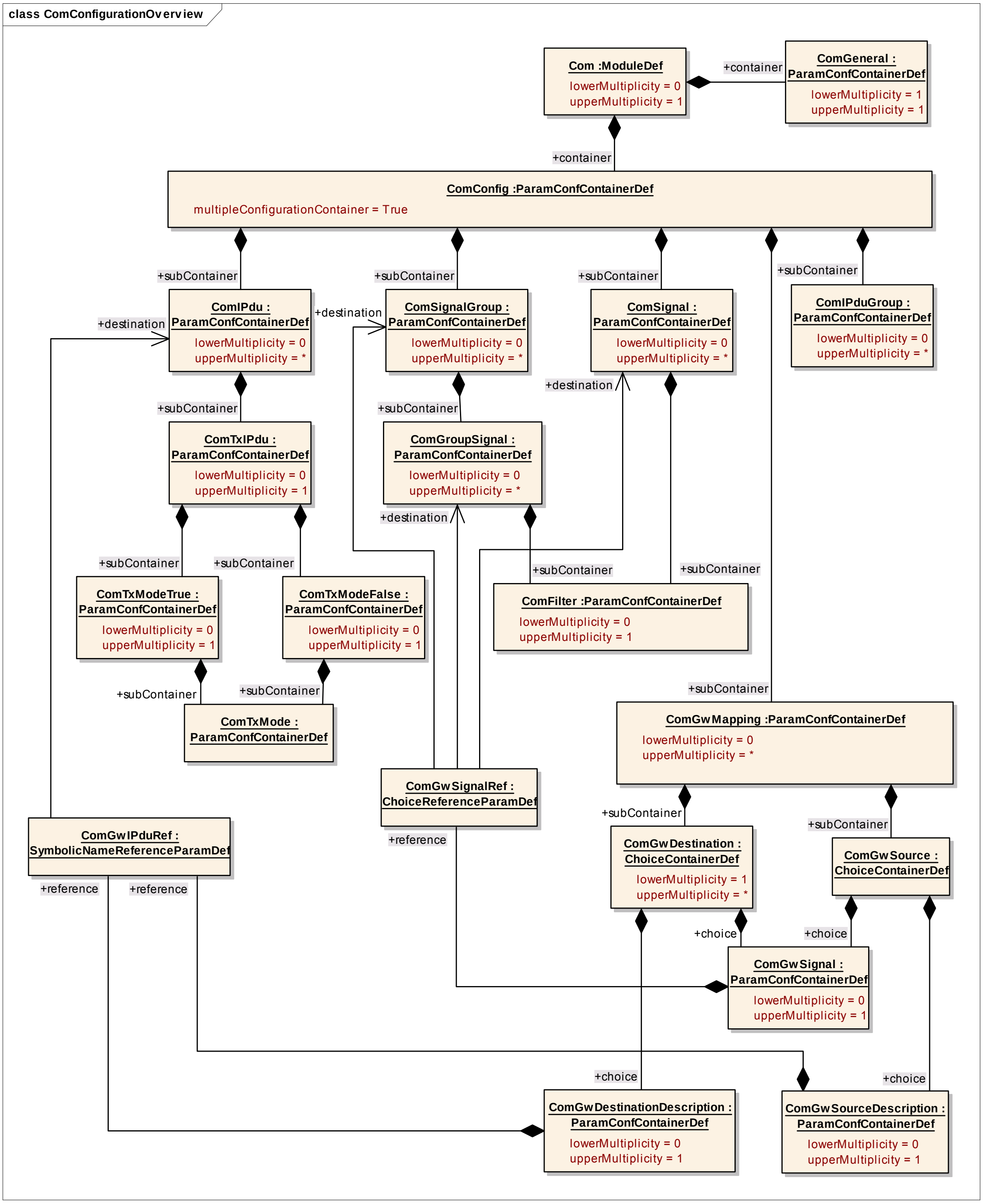

**Figure 19: Com Configuration Overview**

### 10.2.3 Com

| | |
|---|---|
| ***Module Name*** | *Com* |
| ***Module Description*** | COM540: Configuration of the Com module. |

| ***Included Containers*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComConfig | 1 | This container contains the configuration parameters and sub containers of the 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 Com module. |

### 10.2.4 ComConfig

| | |
|---|---|
| ***SWS Item*** | **COM337 :** |
| ***Container Name*** | ComConfig [Multi Config Container] |
| ***Description*** | This container contains the configuration parameters and sub containers of the COM module. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set. |
| ***Configuration Parameters*** | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM394 :** | | |
| ***Name*** | ComConfigurationId | | |
| ***Description*** | This ID is returned by a call to Com_GetConfigurationId. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| ***Included Containers*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComGwMapping | 0..* | Each instance of this container defines one mapping of the integrated Signal Gateway. |
| ComIPdu | 0..* | COM404: See COM340. If there is no such container included no I-PDU is defined. In this case only internal communication is possible. |
| ComIPduGroup | 0..* | COM405: See COM341 If there is no such con-tainer included then no I-PDU group is defined. In this case only internal communication is pos-sible. |
| ComSignal | 0..* | COM407: At least one signal container shall be present, see COM344. |
| ComSignalGroup | 0..* | COM408: If there is no such container included no signal groups are defined. |

**COM404:** See COM340, if there is no ComIpdu container included no I-PDUs are defined. In this case no communication via COM is possible.

**COM405:** See COM341, if there is no ComIPduGroup container included then no I-PDU group is defined. In this case no communication via COM is possible.

**COM407:** See COM344, if there is no ComSignal container included no single signals are defined.

**COM408:** See COM345, if there is no ComSignalGroup container included no signal groups are defined.

### 10.2.5 ComGeneral

| | |
|---|---|
| ***SWS Item*** | **COM541 :** |
| ***Container Name*** | ComGeneral |
| ***Description*** | Contains the general configuration parameters of the Com module. |
| ***Configuration Parameters*** | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM186 :** | | |
| ***Name*** | ComConfigurationTimeBase | | |
| ***Description*** | The period between successive calls to the Main Functions (Rx, Tx, Routing) of AUTOSAR COM in seconds. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | FloatParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | All Variants |
| | ***Link time*** | -- | |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM141 :** | | |
| ***Name*** | ComConfigurationUseDet | | |
| ***Description*** | 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), see COM028. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | BooleanParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | All Variants |
| | ***Link time*** | -- | |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | |
|---|---|
| ***SWS Item*** | **COM438 :** |
| ***Name*** | ComVersionInfoApi |
| ***Description*** | Activate/Deactivate the version information API (Com_GetVersionInfo). |

| | | | |
|---|---|---|---|
| | True: version information API activated False: version information API deactivated | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | BooleanParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | All Variants |
| | ***Link time*** | -- | |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| ***No Included Containers*** |
|---|

## 10.2.6 ComFilter

| | |
|---|---|
| ***SWS Item*** | **COM339 :** |
| ***Container Name*** | ComFilter |
| ***Description*** | This container contains the configuration parameters of COM Filters.<br>Note: On sender side the container is used to specify the transmission mode conditions. |
| ***Configuration Parameters*** | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM146 :** | | |
| ***Name*** | ComFilterAlgorithm | | |
| ***Description*** | The range of values is specified in the [17] specification, chapter 2.2.2, Reception Filtering. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | ALWAYS | | |
| | MASKED_NEW_DIFFERS_MASKED_OLD | | |
| | MASKED_NEW_DIFFERS_X | | |
| | MASKED_NEW_EQUALS_X | | |
| | NEVER | | |
| | NEW_IS_OUTSIDE | | |
| | NEW_IS_WITHIN | | |
| | ONE_EVERY_N | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | |
|---|---|
| ***SWS Item*** | **COM235 :** |
| ***Name*** | ComFilterMask |
| ***Description*** | The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. Only the least significant 32 bits are significant. |
| ***Multiplicity*** | 0..1 |

| | | | |
|---|---|---|---|
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM317 :** | | |
| ***Name*** | ComFilterMax | | |
| ***Description*** | The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. Only the least significant 32 bits are significant. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM318 :** | | |
| ***Name*** | ComFilterMin | | |
| ***Description*** | The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. Only the least significant 32 bits are significant. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM313 :** | | |
| ***Name*** | ComFilterOffset | | |
| ***Description*** | The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. Only the least significant 32 bits are significant.<br>Range = 0..(ComFilterPeriodFactor-1) | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local<br>dependency: COM312 | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM312 :** | | |
| ***Name*** | ComFilterPeriodFactor | | |
| ***Description*** | The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. Only the least significant 32 are significant. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM147 :** | | |
| ***Name*** | ComFilterX | | |
| ***Description*** | The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. Only the least significant 32 bits are significant. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| |
|---|
| ***No Included Containers*** |

## 10.2.7 ComIPdu

| | |
|---|---|
| ***SWS Item*** | **COM340, COM174 :** |
| ***Container Name*** | ComIPdu |
| ***Description*** | Contains the configuration parameters of Com I-Pdus.<br>COM174: The shortName is used as the symbolic name (ComIpduName) of this I-Pdu when communicating with the PduR. Is optional because the Com module might be used for internal communication only. This parameter is only stored in the XML file, and must not be used within the implementation. |
| ***Configuration Parameters*** | |

| | |
|---|---|
| ***SWS Item*** | **COM387 :** |
| ***Name*** | ComIPduCallout |
| ***Description*** | If there is a callout defined for this I-PDU this parameter contains the name of the callout function.. |
| ***Multiplicity*** | 0..1 |
| ***Type*** | FunctionNameDef |
| ***Default value*** | -- |

| | | | |
|---|---|---|---|
| ***ConfigurationClass*** | ***Pre-compile time*** | X | All Variants |
| | ***Link time*** | -- | |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM175 :** | | |
| ***Name*** | ComIPduRxHandleId | | |
| ***Description*** | The numerical value used as the ID of this I-PDU. The Com_IPduRxHandleId is required by the API calls to receive I-PDUs from the PduR. It is only present for I-PDU is received from the PduR, because Com is the starting module for Tx I-PDUs and there is no need to define IDs for Tx I-PDUs in the Com module. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef (Symbolic Name generated for this parameter) | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: external, depends on configuration process | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM119 :** | | |
| ***Name*** | ComIPduSignalProcessing | | |
| ***Description*** | For the definition of the two modes Immediate and Defered, see COM298. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | DEFERED | | |
| | IMMEDIATE | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM176 :** | | |
| ***Name*** | ComIPduSize | | |
| ***Description*** | The size of the I-PDU in bytes.<br>The maximum size is limited by the underlying communication interface.<br>0-8 for CAN and LIN<br>0-254 for FlexRay | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 254 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | |
|---|---|
| ***SWS Item*** | **COM493 :** |
| ***Name*** | ComIpduDirection |
| ***Description*** | The direction defines if this I-PDU, and therefore the contributing signals and |

| | | | |
|---|---|---|---|
| | signal groups, shall be send or received. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | RECEIVE | | |
| | SEND | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: local<br>dependency: If configured to Send also a ComTxIpdu container shall be included, see COM496 | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM206 :** | | |
| ***Name*** | ComIPduGroupRef | | |
| ***Description*** | Reference to the I-PDU group this I-PDU belongs to. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | Reference to ComIPduGroup | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM519 :** | | |
| ***Name*** | ComIPduSignalGroupRef | | |
| ***Description*** | References to all signal groups contained in this I-Pdu | | |
| ***Multiplicity*** | 0..* | | |
| ***Type*** | Reference to ComSignalGroup | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM518 :** | | |
| ***Name*** | ComIPduSignalRef | | |
| ***Description*** | References to all signals contained in this I-PDU. | | |
| ***Multiplicity*** | 0..* | | |
| ***Type*** | Reference to ComSignal | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **--** | | |
| ***Name*** | PduIdRef | | |
| ***Description*** | Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | Reference to Pdu | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |

| | |
|---|---|
| ***Scope / Dependency*** | |

| ***Included Containers*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComTxIPdu | 0..1 | COM496: This container must be included if COM_IPDU_DIRECTION is configured to SEND. |

**COM497:** A ComTxIPdu container must be included if ComIpduDirection is configured to Send.

### 10.2.8 ComTxIPdu

| | |
|---|---|
| ***SWS Item*** | **COM496 :** |
| ***Container Name*** | ComTxIPdu |
| ***Description*** | This container contains additional transmission related configuration parameters of COM I-PDUs |
| ***Configuration Parameters*** | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM181, COM471 :** | | |
| ***Name*** | ComTxIPduMinimumDelayTimeFactor | | |
| ***Description*** | COM181: Minimum delay between successive transmissions of this I-PDU, independent of the transmission mode. There is only one minimum delay time parameter for the I-PDU. This minimum delay time does not change with mode changes. Neither is the timer reset. This means that mode changes are not allowed to violate the minimum delay time. It is not possible to monitor the minumum delay time for I-PDUs that are requested using the Com_TriggerTransmit API.<br>Depending on the implementation, this timeout may be implemented as a 32-bit or a 16-bit counter.<br>COM471: No minimum delay time monitoring shall take place, if ComTxIPduMinimumDelayTimeFactor is omitted or configured to 0. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM017 :** | | |
| ***Name*** | ComTxIPduUnusedAreasDefault | | |
| ***Description*** | AUTOSAR COM fills not used areas of an I-PDU with this bit-pattern. This attribute is mandatory to avoid undefined behaviour. This byte-pattern will be repeated throughout the I-PDU. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 255 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- |

| | | | |
|---|---|---|---|
| | | | BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| ***Included Containers*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComTxModeFalse | 0..1 | COM234: 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 | COM233: The referenced transmission mode object that is used when the filtering state for this I-PDU evaluates to true. |

ComTxModeTrue: COM233: The referenced transmission mode object (see section10.2.2.8) that is used when the filtering state for this I PDU evaluates to true.

ComTxModeFalse: COM234: The referenced transmission mode object (see section 10.2.2.8) that is used when the filtering state for this I PDU evaluates to false

The default is transmission mode None.

### 10.2.9 ComIPduGroup

| ***SWS Item*** | **COM341, COM126 :** |
|---|---|
| ***Container Name*** | ComIPduGroup |
| ***Description*** | Contains the configuration parameters of Com I-Pdu groups.<br>COM126: The shortName is used as the symbolic name of the I-Pdu group (ComIpduGroupName). This parameter is only stored in the XML file, and must not be used within the implementation. |
| ***Configuration Parameters*** | |

| ***SWS Item*** | **COM184 :** | | |
|---|---|---|---|
| ***Name*** | ComIPduGroupHandleId | | |
| ***Description*** | The numerical value used as the ID of this I-PDU Group .<br>The ComIPduGroupHandleId is required by the API calls to start and stop I-PDU Groups.<br>For the rational for the range see COM187. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef (Symbolic Name generated for this parameter) | | |
| ***Range*** | 0 .. 63 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: external, depends on configuration process | | |

| ***SWS Item*** | **COM185 :** |
|---|---|
| ***Name*** | ComIPduGroupGroupRef |
| ***Description*** | If the I-PDU Group belongs to an I-PDU group, this is the name of the I- |

| | | | |
|---|---|---|---|
| | PDU group it belongs to. This I-PDU Group does not belong to another I-PDU group, if this reference is omitted. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | Reference to ComIPduGroup | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| |
|---|
| ***No Included Containers*** |

### 10.2.10 ComSignal

| | |
|---|---|
| ***SWS Item*** | **COM344, COM163 :** |
| ***Container Name*** | ComSignal |
| ***Description*** | Contains the configuration parameters of Com signals.<br>COM163: The shortName is used as the symbolic name of the signal (ComSignalName). This name is also used as the handle name for the signal. This parameter is only stored in the XML file, and must not be used within the implementation. |
| ***Configuration Parameters*** | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM259 :** | | |
| ***Name*** | ComBitPosition | | |
| ***Description*** | Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 63 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM158 :** | | |
| ***Name*** | ComBitSize | | |
| ***Description*** | Size in bits. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 64 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM314 :** | | |
| ***Name*** | ComDataInvalidAction | | |
| ***Description*** | 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. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | NOTIFY | | |
| | REPLACE | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM499 :** | | |
| ***Name*** | ComErrorNotification | | |
| ***Description*** | 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. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | FunctionNameDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM183 :** | | |
| ***Name*** | ComFirstTimeoutFactor | | |
| ***Description*** | Defines the first timeout period for the deadline monitoring. Details can be found in [17].<br>Note: See also COM263 for the configuration of the remaining timeout periods.<br>Depending on the implementation, this timeout may be implemented as a 32-bit or a 16-bit counter. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | |
|---|---|
| ***SWS Item*** | **COM165 :** |
| ***Name*** | ComHandleId |
| ***Description*** | The numerical value used as the ID.<br>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. |

| | | | |
|---|---|---|---|
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef (Symbolic Name generated for this parameter) | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: external, depends on configuration process | | |

| ***SWS Item*** | **COM315 :** | | |
|---|---|---|---|
| ***Name*** | ComInvalidNotification | | |
| ***Description*** | Only valid on receiver side: Name of Com_CbkRxInv callback function to be called.<br>Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComSignalDataInvalidAction is configured to Notify. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | FunctionNameDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| ***SWS Item*** | **COM498 :** | | |
|---|---|---|---|
| ***Name*** | ComNotification | | |
| ***Description*** | On sender side: Name of Com_CbkTxAck callback function to be called.<br>On receiver side: Name of Com_CbkRxAck callback function to be called.<br>If this parameter is omitted no notification shall take place. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | FunctionNameDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| ***SWS Item*** | **COM412, COM470, COM500, COM513 :** | |
|---|---|---|
| ***Name*** | ComRxDataTimeoutAction | |
| ***Description*** | COM412: This parameter defines the action performed upon a reception timeout violation.<br>COM500: If this parameter is omitted or configured to None no replacement shall take place.<br>COM470: Relating to signals: When this parameter is set to Replace, the replacement value used shall be the ComInitValue.<br>COM513: Relating to signal groups: When this parameter is set to Replace, all included signals shall be set to their ComInitValue. | |
| ***Multiplicity*** | 0..1 | |
| ***Type*** | EnumerationParamDef | |
| ***Range*** | NONE | |

| | | | |
|---|---|---|---|
| | REPLACE | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| ***SWS Item*** | **COM391, COM501 :** | | |
|---|---|---|---|
| ***Name*** | ComSignalDataInvalidValue | | |
| ***Description*** | COM391: On receiver side: When this value is received it is recognized as the invalid value and the appropriate invalid action (as specified by Com-DataInvalidAction) is performed.<br>COM501: On sender side: This configures the data invalid value that is used by a call to Com_InvalidateSignal. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| ***SWS Item*** | **COM157 :** | | |
|---|---|---|---|
| ***Name*** | ComSignalEndianess | | |
| ***Description*** | Defines the endianness of the signal's network representation. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | BIG_ENDIAN | | |
| | LITTLE_ENDIAN | | |
| | OPAQUE | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| ***SWS Item*** | **COM170, COM483 :** | | |
|---|---|---|---|
| ***Name*** | ComSignalInitValue | | |
| ***Description*** | COM170: Initial value for this signal. The default value is 0. The lower n-bits of the configured Integer shall be used as init-value for an n-bit sized signal type.<br>COM483: If the signal is of type UINT[n], the Integer's least significant byte shall be assigned to the byte array’s last byte. The second-least significant byte shall be assigned to the byte array’s last but one byte, and so on. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | 0 | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local<br>dependency: IpduM | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM437 :** | | |
| ***Name*** | ComSignalLength | | |
| ***Description*** | The ComSignalLength specifies the n (in Bytes: 1..8) of the type UINT8[n]. For other types it will be ignored. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 1 .. 8 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM127 :** | | |
| ***Name*** | ComSignalType | | |
| ***Description*** | 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. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | BOOLEAN | | |
| | SINT16 | | |
| | SINT32 | | |
| | SINT8 | | |
| | UINT16 | | |
| | UINT32 | | |
| | UINT8 | | |
| | UINT8_N | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM263, COM264, COM333 :** | | |
| ***Name*** | ComTimeoutFactor | | |
| ***Description*** | COM263: Defines the timeout period for the deadline monitoring. Details can be found in [17].<br>Note: The period for the ComFirstTimeoutFactor could differ from the ComTimeoutFactor. Depending on the implementation, this timeout may be implemented as a 32-bit or a 16-bit counter.<br>COM264: If deadline monitoring is used on a signal with an update-bit this defines the timeout for deadline monitoring.<br>COM333: If the timeout is omitted or configured to 0 than no timeout monitoring shall take place. In this case ComFirstTimeoutFactor shall be ignored. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |

| | | | |
|---|---|---|---|
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM552 :** | | |
| ***Name*** | ComTimeoutNotification | | |
| ***Description*** | On sender side: Name of Com_CbkTxTOut callback function to be called.<br>On receiver side: Name of Com_CbkRxTOut callback function to be called. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | FunctionNameDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM232 :** | | |
| ***Name*** | ComTransferProperty | | |
| ***Description*** | 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.<br>TRIGGERED: Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU.<br>PENDING: A write access to this signal never triggers the transmission of the corresponding I-PDU.<br>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 written or init) value. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | PENDING | | |
| | TRIGGERED | | |
| | TRIGGERED_ON_CHANGE | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM257 :** | | |
| ***Name*** | ComUpdateBitPosition | | |
| ***Description*** | Bit position of update-bit inside I-PDU.<br>If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 63 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |

| | | | |
|---|---|---|---|
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **--** | | |
| ***Name*** | SystemTemplateSystemSignalRef | | |
| ***Description*** | Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | Foreign reference to ISignalToIPduMapping | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | -- | |
| | ***Link time*** | -- | |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: system configuration | | |

| ***Included Containers*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComFilter | 0..1 | This container contains the configuration parameters of COM Filters.<br>Note: On sender side the container is used to specify the transmission mode conditions. |

**ComFilter: COM169:** On receiver side:
The name of the filter type as defined in the filter object in section 11.3.1.

ComFilter: COM275: On sender side:
Reference to a filter object which is used to determine the transmission mode selector of the I PDU the signal belongs to.
If this attribute is omitted, the signal does not contribute to the evaluation of the transmission mode of the I PDU the signal belongs to.

### 10.2.11 ComSignalGroup

| | |
|---|---|
| ***SWS Item*** | **COM345, COM044 :** |
| ***Container Name*** | ComSignalGroup |
| ***Description*** | Contains the configuration parameters of Com signal groups.<br>COM044: The shortName is used as the symbolic name of the signal group (ComSignalGroupName). This name is also used as the handle name for the signal group. This parameter is only stored in the XML file, and must not be used within the implementation. |
| ***Configuration Parameters*** | |

| | | |
|---|---|---|
| ***SWS Item*** | **COM259 :** | |
| ***Name*** | ComBitPosition | |
| ***Description*** | Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. | |
| ***Multiplicity*** | 1 | |
| ***Type*** | IntegerParamDef | |
| ***Range*** | 0 .. 63 | |

| | | | |
|---|---|---|---|
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM158 :** | | |
| ***Name*** | ComBitSize | | |
| ***Description*** | Size in bits. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 64 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM314 :** | | |
| ***Name*** | ComDataInvalidAction | | |
| ***Description*** | 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. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | NOTIFY | | |
| | REPLACE | Literal for DataInvalidAction | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM499 :** | | |
| ***Name*** | ComErrorNotification | | |
| ***Description*** | 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. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | FunctionNameDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: local | | |

| | |
|---|---|
| ***SWS Item*** | **COM183 :** |
| ***Name*** | ComFirstTimeoutFactor |
| ***Description*** | Defines the first timeout period for the deadline monitoring. Details can be found in [17]. |

| | | | |
|---|---|---|---|
| | Note: See also COM263 for the configuration of the remaining timeout periods.<br>Depending on the implementation, this timeout may be implemented as a 32-bit or a 16-bit counter. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM165 :** | | |
| ***Name*** | ComHandleId | | |
| ***Description*** | The numerical value used as the ID.<br>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. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef (Symbolic Name generated for this parameter) | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: external, depends on configuration process | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM315 :** | | |
| ***Name*** | ComInvalidNotification | | |
| ***Description*** | Only valid on receiver side: Name of Com_CbkRxInv callback function to be called.<br>Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComSignalDataInvalidAction is configured to Notify. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | FunctionNameDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | |
|---|---|
| ***SWS Item*** | **COM498 :** |
| ***Name*** | ComNotification |
| ***Description*** | On sender side: Name of Com_CbkTxAck callback function to be called.<br>On receiver side: Name of Com_CbkRxAck callback function to be called.<br>If this parameter is omitted no notification shall take place. |
| ***Multiplicity*** | 0..1 |
| ***Type*** | FunctionNameDef |

| | | | |
|---|---|---|---|
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM412, COM470, COM500, COM513 :** | | |
| ***Name*** | ComRxDataTimeoutAction | | |
| ***Description*** | COM412: This parameter defines the action performed upon a reception timeout violation.<br>COM500: If this parameter is omitted or configured to None no replacement shall take place.<br>COM470: Relating to signals: When this parameter is set to Replace, the replacement value used shall be the ComInitValue.<br>COM513: Relating to signal groups: When this parameter is set to Replace, all included signals shall be set to their ComInitValue. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | NONE | Literal for DataTimeoutAction | |
| | REPLACE | Literal for DataTimeoutAction | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM263, COM264, COM333 :** | | |
| ***Name*** | ComTimeoutFactor | | |
| ***Description*** | COM263: Defines the timeout period for the deadline monitoring. Details can be found in [17].<br>Note: The period for the ComFirstTimeoutFactor could differ from the ComTimeoutFactor. Depending on the implementation, this timeout may be implemented as a 32-bit or a 16-bit counter.<br>COM264: If deadline monitoring is used on a signal with an update-bit this defines the timeout for deadline monitoring.<br>COM333: If the timeout is omitted or configured to 0 than no timeout monitoring shall take place. In this case ComFirstTimeoutFactor shall be ignored. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | |
|---|---|
| ***SWS Item*** | **COM552 :** |
| ***Name*** | ComTimeoutNotification |
| ***Description*** | On sender side: Name of Com_CbkTxTOut callback function to be called.<br>On receiver side: Name of Com_CbkRxTOut callback function to be called. |
| ***Multiplicity*** | 0..1 |
| ***Type*** | FunctionNameDef |

| | | | |
|---|---|---|---|
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM232 :** | | |
| ***Name*** | ComTransferProperty | | |
| ***Description*** | 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.<br>TRIGGERED: Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU.<br>PENDING: A write access to this signal never triggers the transmission of the corresponding I-PDU.<br>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 written or init) value. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | PENDING | | |
| | TRIGGERED | | |
| | TRIGGERED_ON_CHANGE | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM257 :** | | |
| ***Name*** | ComUpdateBitPosition | | |
| ***Description*** | Bit position of update-bit inside I-PDU.<br>If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 63 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **--** | | |
| ***Name*** | SystemTemplateSignalGroupRef | | |
| ***Description*** | Reference to the ISignalToIPduMapping that contains a reference to the ISignal (SystemTemplate) which this ComSignalGroup represents. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | Foreign reference to ISignalToIPduMapping | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | -- | |
| | ***Link time*** | -- | |

| | | | |
|---|---|---|---|
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: system configuration | | |

| ***Included Containers*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComGroupSignal | 0..* | COM520: This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.<br>COM521: The shortName is used as the symbolic name of the signal (ComSignalName). This name is also used as the handle name for the signal. This parameter is only stored in the XML file, and must not be used within the implementation. |

## 10.2.12 ComGroupSignal

| | |
|---|---|
| ***SWS Item*** | **COM520, COM521 :** |
| ***Container Name*** | ComGroupSignal |
| ***Description*** | COM520: This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.<br>COM521: The shortName is used as the symbolic name of the signal (ComSignalName). This name is also used as the handle name for the signal. This parameter is only stored in the XML file, and must not be used within the implementation. |
| ***Configuration Parameters*** | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM259 :** | | |
| ***Name*** | ComBitPosition | | |
| ***Description*** | Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 63 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM158 :** | | |
| ***Name*** | ComBitSize | | |
| ***Description*** | Size in bits. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 64 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |

| | | | |
|---|---|---|---|
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM165 :** | | |
| ***Name*** | ComHandleId | | |
| ***Description*** | The numerical value used as the ID.<br>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. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef (Symbolic Name generated for this parameter) | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: external, depends on configuration process | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM315 :** | | |
| ***Name*** | ComInvalidNotification | | |
| ***Description*** | Only valid on receiver side: Name of Com_CbkRxInv callback function to be called.<br>Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComSignalDataInvalidAction is configured to Notify. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | FunctionNameDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM391, COM501 :** | | |
| ***Name*** | ComSignalDataInvalidValue | | |
| ***Description*** | COM391: On receiver side: When this value is received it is recognized as the invalid value and the appropriate invalid action (as specified by ComDataInvalidAction) is performed.<br>COM501: On sender side: This configures the data invalid value that is used by a call to Com_InvalidateSignal. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | |
|---|---|
| ***SWS Item*** | **COM157 :** |
| ***Name*** | ComSignalEndianess |

| | | | |
|---|---|---|---|
| ***Description*** | Defines the endianness of the signal's network representation. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | BIG_ENDIAN | | |
| | LITTLE_ENDIAN | | |
| | OPAQUE | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM170, COM483 :** | | |
| ***Name*** | ComSignalInitValue | | |
| ***Description*** | COM170: Initial value for this signal. The default value is 0. The lower n-bits of the configured Integer shall be used as init-value for an n-bit sized signal type.<br>COM483: If the signal is of type UINT[n], the Integer's least significant byte shall be assigned to the byte array's last byte. The second-least significant byte shall be assigned to the byte array's last but one byte, and so on. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | 0 | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local<br>dependency: IpduM | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM437 :** | | |
| ***Name*** | ComSignalLength | | |
| ***Description*** | The ComSignalLength specifies the n (in Bytes: 1..8) of the type UINT8[n]. For other types it will be ignored. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 1 .. 8 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | |
|---|---|---|
| ***SWS Item*** | **COM127 :** | |
| ***Name*** | ComSignalType | |
| ***Description*** | 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. | |
| ***Multiplicity*** | 1 | |
| ***Type*** | EnumerationParamDef | |
| ***Range*** | BOOLEAN | |
| | SINT16 | |

| | | | |
|---|---|---|---|
| | SINT32 | | |
| | SINT8 | | |
| | UINT16 | | |
| | UINT32 | | |
| | UINT8 | | |
| | UINT8_N | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM560 :** | | |
| ***Name*** | ComTransferProperty | | |
| ***Description*** | 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.<br>PENDING: a change of the value of this group signal shall not be considered in the evaluation of the signal groups ComTransferProperty.<br>TRIGGERED_ON_CHANGE: a change of the value of this group signal shall be considered in the in the evaluation of the signal groups ComTransferProperty. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | PENDING | | |
| | TRIGGERED_ON_CHANGE | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **--** | | |
| ***Name*** | SystemTemplateSystemSignalRef | | |
| ***Description*** | Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | Foreign reference to ISignalToIPduMapping | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | -- | |
| | ***Link time*** | -- | |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: system configuration | | |

| ***Included Containers*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComFilter | 0..1 | This container contains the configuration parameters of COM Filters.<br>Note: On sender side the container is used to specify the transmission mode conditions. |

### 10.2.13 ComTxMode

| | |
|---|---|
| ***SWS Item*** | **COM351 :** |
| ***Container Name*** | ComTxMode |
| ***Description*** | This container contains the configuration parameters of COM transmission modes. |
| ***Configuration Parameters*** | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM137 :** | | |
| ***Name*** | ComTxModeMode | | |
| ***Description*** | The available transmission modes described in [18] shall be extended by the additional mode None.<br>The transmission mode None shall not have any further sub-attributes in the ComTxMode object. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | DIRECT | | |
| | MIXED | | |
| | NONE | Literal for TxMode | |
| | PERIODIC | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM281 :** | | |
| ***Name*** | ComTxModeNumberOfRepetitions | | |
| ***Description*** | Defines the number of repetitions for the Direct/N-Times transmission mode and the event driven part of Mixed transmission mode. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 255 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM282 :** | | |
| ***Name*** | ComTxModeRepetitionPeriodFactor | | |
| ***Description*** | Period of the repetition of the n transmission for the Direct/NTimes transmission mode and the event driven part of the Mixed transmission mode. Depending on the implementation, this timeout may be implemented as a 32-bit or a 16-bit counter. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |

| | | | |
|---|---|---|---|
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM180 :** | | |
| ***Name*** | ComTxModeTimeOffsetFactor | | |
| ***Description*** | Time until first transmission of this I-PDU.<br>ComTxModeTimeOffsetFactor defines the time between Com_IpduGroupStart and the first transmission of the cyclic part of this transmission request for this I-PDU.<br>Depending on the implementation, this timeout may be implemented as a 32-bit or a 16-bit counter. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM178 :** | | |
| ***Name*** | ComTxModeTimePeriodFactor | | |
| ***Description*** | Period of the repetition of cyclic transmissions. Depending on the implementation, this timeout may be implemented as a 32-bit or a 16-bit counter. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| ***No Included Containers*** |
|---|

### 10.2.14 ComTxModeTrue

| | |
|---|---|
| ***SWS Item*** | **COM455 :** |
| ***Container Name*** | ComTxModeTrue |
| ***Description*** | This container contains the configuration parameters of COM transmission modes in the case the ComFilter evaluates to true. |
| ***Configuration Parameters*** | |

| ***Included Containers*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComTxMode | 1 | This container contains the configuration parameters of COM transmission modes. |

### 10.2.15 ComTxModeFalse

| | |
|---|---|
| ***SWS Item*** | **COM454 :** |
| ***Container Name*** | ComTxModeFalse |
| ***Description*** | This container contains the configuration parameters of COM transmission modes in the case the ComFilter evaluates to false. |
| ***Configuration Parameters*** | |

| ***Included Containers*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComTxMode | 1 | This container contains the configuration parameters of COM transmission modes. |

### 10.2.16 ComGwMapping

| | |
|---|---|
| ***SWS Item*** | **COM544 :** |
| ***Container Name*** | ComGwMapping |
| ***Description*** | Each instance of this container defines one mapping of the integrated Signal Gateway. |
| ***Configuration Parameters*** | |

| ***Included Containers*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| 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. |

### 10.2.17 ComGwSource

| | |
|---|---|
| ***SWS Item*** | **COM545 :** |
| ***Choice Container Name*** | ComGwSource |
| ***Description*** | 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. |

| ***Container Choices*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| 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. |

### 10.2.18 ComGwSourceDescription

| | |
|---|---|
| ***SWS Item*** | **COM548 :** |
| ***Container Name*** | ComGwSourceDescription |
| ***Description*** | 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. |
| ***Configuration Parameters*** | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM259 :** | | |
| ***Name*** | ComBitPosition | | |
| ***Description*** | Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 63 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM158 :** | | |
| ***Name*** | ComBitSize | | |
| ***Description*** | Size in bits. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 64 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM157 :** | | |
| ***Name*** | ComSignalEndianess | | |
| ***Description*** | Defines the endianness of the signal's network representation. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | BIG_ENDIAN | | |
| | LITTLE_ENDIAN | | |
| | OPAQUE | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |

| | |
|---|---|
| ***Scope / Dependency*** | scope: Local |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM127 :** | | |
| ***Name*** | ComSignalType | | |
| ***Description*** | 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. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | BOOLEAN | | |
| | SINT16 | | |
| | SINT32 | | |
| | SINT8 | | |
| | UINT16 | | |
| | UINT32 | | |
| | UINT8 | | |
| | UINT8_N | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM257 :** | | |
| ***Name*** | ComUpdateBitPosition | | |
| ***Description*** | Bit position of update-bit inside I-PDU.<br>If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 63 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM550 :** | | |
| ***Name*** | ComGwIPduRef | | |
| ***Description*** | Symbolic reference to an I-PDU of a Signal Gateway source or destination description. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | Reference to ComIPdu | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | | | |

| |
|---|
| ***No Included Containers*** |

### 10.2.19 ComGwDestination

| | |
|---|---|
| ***SWS Item*** | **COM546 :** |
| ***Choice Container Name*** | ComGwDestination |
| ***Description*** | 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. |

| ***Container Choices*** | | |
|---|---|---|
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| 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 ComGroupSignal or a ComSignalGroup. |

### 10.2.20 ComGwDestinationDescription

| | |
|---|---|
| ***SWS Item*** | **COM549 :** |
| ***Container Name*** | ComGwDestinationDescription |
| ***Description*** | 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. |
| ***Configuration Parameters*** | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM259 :** | | |
| ***Name*** | ComBitPosition | | |
| ***Description*** | Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 63 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | |
|---|---|
| ***SWS Item*** | **COM391, COM501 :** |
| ***Name*** | ComSignalDataInvalidValue |
| ***Description*** | COM391: On receiver side: When this value is received it is recognized as the invalid value and the appropriate invalid action (as specified by ComDataInvalidAction) is performed.<br>COM501: On sender side: This configures the data invalid value that is used by a call to Com_InvalidateSignal. |
| ***Multiplicity*** | 0..1 |
| ***Type*** | IntegerParamDef |

| | | | |
|---|---|---|---|
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| | ***Post-build time*** | -- | |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM157 :** | | |
| ***Name*** | ComSignalEndianess | | |
| ***Description*** | Defines the endianness of the signal's network representation. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EnumerationParamDef | | |
| ***Range*** | BIG_ENDIAN | | |
| | LITTLE_ENDIAN | | |
| | OPAQUE | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM170, COM483 :** | | |
| ***Name*** | ComSignalInitValue | | |
| ***Description*** | COM170: Initial value for this signal. The default value is 0. The lower n-bits of the configured Integer shall be used as init-value for an n-bit sized signal type.<br>COM483: If the signal is of type UINT[n], the Integer's least significant byte shall be assigned to the byte array’s last byte. The second-least significant byte shall be assigned to the byte array’s last but one byte, and so on. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Default value*** | 0 | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local<br>dependency: IpduM | | |

| | |
|---|---|
| ***SWS Item*** | **COM232 :** |
| ***Name*** | ComTransferProperty |
| ***Description*** | 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.<br>TRIGGERED: Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU.<br>PENDING: A write access to this signal never triggers the transmission of the corresponding I-PDU.<br>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 written or init) value. |
| ***Multiplicity*** | 0..1 |
| ***Type*** | EnumerationParamDef |

| | | | |
|---|---|---|---|
| ***Range*** | PENDING | | |
| | TRIGGERED | | |
| | TRIGGERED_ON_CHANGE | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM257 :** | | |
| ***Name*** | ComUpdateBitPosition | | |
| ***Description*** | Bit position of update-bit inside I-PDU.<br>If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | IntegerParamDef | | |
| ***Range*** | 0 .. 63 | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: Local | | |

| | | | |
|---|---|---|---|
| ***SWS Item*** | **COM550 :** | | |
| ***Name*** | ComGwIPduRef | | |
| ***Description*** | Symbolic reference to an I-PDU of a Signal Gateway source or destination description. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | Reference to ComIPdu | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | | | |

| |
|---|
| ***No Included Containers*** |

### 10.2.21 ComGwSignal

| | |
|---|---|
| ***SWS Item*** | **COM551 :** |
| ***Container Name*** | ComGwSignal |
| ***Description*** | This container allows specifying a gateway source or destination respectively with a reference to a ComSignal, a ComGroupSignal or a ComSignalGroup. |
| ***Configuration Parameters*** | |

| | |
|---|---|
| ***SWS Item*** | **COM547 :** |
| ***Name*** | ComGwSignalRef |
| ***Description*** | Reference to an object of a gateway relation. Either to a ComSignal, ComGroupSignal or to a SignalGroup. |
| ***Multiplicity*** | 1 |

| | | | |
|---|---|---|---|
| ***Type*** | Choice Reference to ComGroupSignal,ComSignal,ComSignalGroup | | |
| ***ConfigurationClass*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| | ***Link time*** | X | VARIANT-LINK-TIME |
| | ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | | | |

| ***No Included Containers*** |
|---|

## 10.3 Published Information

Published information contains data defined by the implementer of the SW module that does not change when the module is adapted (i.e. configured) to the actual HW/SW environment. It thus contains version and manufacturer information.

**COM026:** The following table specifies parameters that shall be published in the module's header file and also in the module's description file.

The standard common published information like

- vendorId (<Module>_VENDOR_ID),
- □onfigur (<Module>_MODULE_ID),
- arMajorVersion (<Module>_AR_MAJOR_VERSION),
- arMinorVersion (<Module>_ AR_MINOR_VERSION),
- arPatchVersion (<Module>_ AR_PATCH_VERSION),
- swMajorVersion (<Module>_SW_MAJOR_VERSION),
- swMinorVersion (<Module>_ SW_MINOR_VERSION),
- swPatchVersion (<Module>_ SW_PATCH_VERSION),
- vendorApiInfix (<Module>_VENDOR_API_INFIX)

is provided in the BSW Module Description Template (see 0 Figure 4.1 and Figure 7.1).

Additional published parameters are listed below if applicable for this module.

## 10.4 Defines

Besides the configuration the following defines shall be implemented:

**COM028:** If COM_DEV_ERROR_DETECT is set to ON, the detection and reporting of development errors is enabled for the AUTOSAR COM module. For configuration of this parameter see COM141.

**Note:** This parameter shall be an output of the configuration tool. (Input for the source code)

## 10.5 Configuration rules

### 10.5.1 General rules

**COM401:** 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 ComIPdu
- shortName of a ComIPduGroup

**COM560:** 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 four parameters

- shortName of a ComSignal
- shortName of a ComSignalGroup
- shortName of a ComIPdu
- shortName of a ComIPduGroup

**COM402** 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 ComIPdu
- shortName of a ComIPduGroup
- ComNotification
- ComErrorNotification
- ComTimeoutNotification
- ComInvalidNotification
- ComIPduCallout

### 10.5.2 Signal configuration

**COM489:** It shall be ensured, that the data invalid value configured for the sender side is the same as configured for all receiver sides (see also COM097).

**Note:** The data invalid value shall not be within the valid range of the signal. This can not be enforced by COM since knowledge about the application is needed.

More than one signal can be packed into an I-PDU as long as the following packing rules are fulfilled:

**COM101:** No (group) signal shall span more than one I-PDU.

**COM102:** (Group) signals are not allowed to overlap each other.

**COM105:** Signals (no group signals or signal groups) which are represented in I-PDUs as a multiple of 8-bits shall start at byte border only.

**Note:** It is explicitly allowed that a (group) signal may have the size 0, see COM158.

**COM443:** A (group) signal of type uint8[n] shall always be mapped to an n-bytes sized (group) signal.

**COM553:** A (group) signal of type uint8[n] shall be configured to have OPAQUE endianness.

**COM474:** The initial value of a (group) signal shall always be within the possible range of the (group) signal (including the data invalid value).

**Example:** If a signal is of data-type uint8 has a bit-size of 6-bit the possible range is 0..63. In this case it shall not be allowed to configure an init value of 64 or greater.

### 10.5.3 Signal group configuration

**COM365:** It shall not be allowed to configure signal groups for routing with data type differences between receive and transmit signal group.

### 10.5.4 Transmission Mode configuration

**COM319:** It shall not be allowed to configure filter or TMS-conditions that uses floats. Floats are not allowed to be used in filter conditions. See [17] and COM132. Therefore floats are not allowed for conditions of TMS.

**COM465:** Every COM_TRANSMISSION_MODE_TRUE or COM_TRANSMISSION_MODE_FALSE that is a potential result of the configured/ calculated TMS must be configured. Within the COM_IPDU at least one of the containers COM_TRANSMISSION_MODE_FALSE or COM_TRANSMISSION_MODE_TRUE has to be included.

### 10.5.5 Signal Gateway configuration

**COM384:** The bit size of a received and to be routed signal shall not differ.

**COM386:** Optimization issue: In case of an I-PDU containing signals to be routed completely via a transmit I-PDU by retention the signal order and the signals endianness (related use case: rate conversion), it can be configured to be handled en bloc.

In case of a ComSignal to be routed by the Signal Gateway has been configured for deadline monitoring at the receiving node, depending on the use case, it is possible to configure update-bits, via ComUpdateBitPosition, for the transmit signal thus that it can be detected at the receiving node if the signal has really been update by the sender and is not just repeated not updated by the Signal Gateway.

### 10.5.6 Filter Configuration

**COM535:** For the filter F_OneEveryN, the COM_FILTER_OFFSET shall be configured to a value lesser than COM_FILTER_PERIOD_FACTOR.

### 10.5.7 Post Build Configuration

**COM373:** The post-build time configuration part (post-compile and post-link time) can only be updated when it is not in use

**COM487:** The whole post-build time configurable configuration shall be identifiable by a unique identifier.

# 11 Changes to Release 2.1

This chapter lists all modified SWS items. Additionally referenced figures, tables, notes and so on were updated.

## 11.1 Deleted SWS Items

| *SWS Item* | *Rationale* |
|---|---|
| COM045 | removed due to revision of the COM configuration |
| COM076 | interaction of update-bits and Com_IpduGroupStart is now completely defined in COM222 |
| COM116 | deleted due to removal of internal communication |
| COM149 | removed due to revision of the COM configuration |
| COM152 | removed due to revision of the COM configuration |
| COM156 | removed due to revision of the COM configuration |
| COM161 | removed due to revision of the COM configuration |
| COM166 | due to the removal of the internal communication the direction is configured via COM_IPDU_DIRECTION see COM493 |
| COM167 | removed due to revision of the COM configuration |
| COM216 | removed due to revision of the COM configuration |
| COM246 | removed due to revision of the COM configuration |
| COM247 | removed due to revision of the COM configuration |
| COM248 | removed due to revision of the COM configuration |
| COM258 | removed due to revision of the COM configuration |
| COM267 | removed due to revision of the COM configuration |
| COM285 | turned requirement into a note, since it was redundant to N-Times requirements COM276 and accessory |
| COM316 | removed due to revision of the COM configuration |
| COM343 | removed due to revision of the COM configuration |
| COM349 | deleted due to removal of internal communication |
| COM350 | removed COM_SIGNAL_GROUP_TRANSFER_PROPERTY – the transfer property of the signal group shall be configured within the corresponding COM_NETWORK_SIGNAL |
| COM356 | removed due to revision of the COM configuration |
| COM338 | removed due to revision of the COM configuration |
| COM342 | removed due to revision of the COM configuration |
| COM382 | removed due to revision of the COM configuration |
| COM385 | removed, as considered a unnecessary restriction |
| COM389 | removed due to revision of the COM configuration |
| COM406 | COM_NETWORK_SIGNAL was removed by revising the COM □onfiguretion |
| COM410 | removed due to revision of the COM configuration |
| COM411 | removed due to revision of the COM configuration |
| COM415 | removed due to revision of the COM configuration |
| COM416 | COM_RX_DATA_TIMEOUT_SUBSTITUTION shall be configured via the associated COM_NETWORK_SIGNAL |
| COM429 | removed due to revision of the COM configuration |
| COM440 | deleted due to removal of internal communication |
| COM441 | deleted due to removal of internal communication |

| | |
|---|---|
| COM446 | removed due to revision of the COM configuration |
| COM447 | removed due to revision of the COM configuration |
| COM448 | removed due to revision of the COM configuration |
| COM449 | removed due to revision of the COM configuration |
| COM450 | removed due to revision of the COM configuration |
| COM451 | removed due to revision of the COM configuration |
| COM452 | removed due to revision of the COM configuration |
| COM453 | COM_NOTIFICATION_ERROR shall be configured via the associated COM_NETWORK_SIGNAL |
| COM456 | deleted due to removal of internal communication |
| COM457 | deleted due to removal of internal communication |
| COM458 | changed to note due to revision of the COM configuration |
| COM462 | became obsolete because of deleting COM350 |
| COM463 | removed due to revision of the COM configuration |
| COM482 | deleted due to removal of internal communication |
| COM485 | deleted due to removal of internal communication |
| COM490 | deleted due to removal of internal communication |

## 11.2 Replaced SWS Items

| ***SWS Item*** | ***Rationale*** |
|---|---|
| | |
| | |

## 11.3 Changed SWS Items

| ***SWS Item*** | ***Rationale*** |
|---|---|
| COM001 | harmonized TriggerTransmit APIs within the communication stack |
| COM013 | removed internal communication |
| COM017 | updated due to revision of the COM configuration |
| COM044 | updated due to revision of the COM configuration |
| COM053 | updated due to revision of the COM configuration |
| COM119 | updated due to revision of the COM configuration |
| COM126 | updated due to revision of the COM configuration |
| COM127 | updated due to revision of the COM configuration |
| COM130 | changed due to removal of internal communication |
| COM137 | updated due to revision of the COM configuration |
| COM141 | updated due to revision of the COM configuration |
| COM157 | updated due to revision of the COM configuration |
| COM158 | updated due to revision of the COM configuration |
| COM163 | updated due to revision of the COM configuration |
| COM165 | updated due to revision of the COM configuration |
| COM170 | updated due to revision of the COM configuration |
| COM175 | updated due to revision of the COM configuration |
| COM176 | the maximum size of an I-PDU is no longer restricted to 8 bytes for FlexRay |
| COM178 | updated due to revision of the COM configuration |
| COM180 | updated due to revision of the COM configuration |
| COM181 | updated due to revision of the COM configuration |
| COM183 | updated due to revision of the COM configuration |

| | |
|---|---|
| COM184 | extended maximum number of I-PDU groups to 64 |
| COM185 | updated due to revision of the COM configuration |
| COM186 | updated due to revision of the COM configuration |
| COM187 | extended maximum number of I-PDU groups to 64 |
| COM197 | replaced Com_ApplicationDataRefType by void* respectively const void* to allow usage of compiler abstraction macros |
| COM198 | replaced Com_ApplicationDataRefType by void* respectively const void* to allow usage of compiler abstraction macros |
| COM199 | replaced Com_ApplicationDataRefType by void* respectively const void* to allow usage of compiler abstraction macros |
| COM202 | replaced Com_ApplicationDataRefType by void* respectively const void* to allow usage of compiler abstraction macros |
| COM206 | updated due to revision of the COM configuration |
| COM222 | included initialization information of update-bits |
| COM232 | added TRIGGERED_ON_CHANGE |
| COM257 | updated due to revision of the COM configuration |
| COM259 | updated due to revision of the COM configuration |
| COM263 | updated due to revision of the COM configuration |
| COM281 | updated due to revision of the COM configuration |
| COM282 | updated due to revision of the COM configuration |
| COM287 | clarified requirement |
| COM291 | updated due to revision of the COM configuration |
| COM314 | updated due to revision of the COM configuration |
| COM315 | updated due to revision of the COM configuration |
| COM328 | updated due to removal of internal communication |
| COM330 | adapted to new transfer property TRIGGER_ON_CHANGE |
| COM339 | updated due to revision of the COM configuration |
| COM355 | updated due to revision of the COM configuration |
| COM380 | clarification of filter-usage for uint[n] |
| COM384 | clarification of Signal Gateway behavior |
| COM387 | updated due to revision of the COM configuration |
| COM391 | updated due to revision of the COM configuration |
| COM396 | changed order of execution |
| COM401 | removed several items |
| COM404 | updated due to removal of internal communication |
| COM407 | updated due to revision of the COM configuration |
| COM412 | updated due to revision of the COM configuration |
| COM430 | renamed Com_PCcf.c to Com_Cfg.c according to the General SRS |
| COM432 | updated due to removal of internal communication |
| COM437 | updated due to revision of the COM configuration |
| COM438 | updated due to revision of the COM configuration |
| COM442 | added instanceId |
| COM459 | development COM_SERVICE_NOT_AVAILABLE also might indicate development error<br>removed E_NOT_OK since it is no longer used |
| COM468 | harmonization of COM-RTE callback interface: Com_CbkTxAck |
| COM469 | clarified minimum delay time handling |
| COM470 | updated due to revision of the COM configuration |
| COM471 | updated due to revision of the COM configuration |
| COM491 | harmonization of COM-RTE callback interface: Com_CbkTxAck |
| COM493 | updated due to revision of the COM configuration |

## 11.4 Added SWS Items

| *SWS Item* | *Rationale* |
|---|---|
| COM492 | clarification if an I-PDU-Callout has to be called in conjunction with COM_TriggerIPDUSend |
| COM493 | due to the removal of the internal communication the direction is configured within the I-PDU via COM_IPDU_DIRECTION. |
| COM494 | clarification of N-Times request within Mixed transmission mode |
| COM495 | clarification of TMS-switch to Cyclic transmission mode |
| COM496 | added due to revision of the COM configuration |
| COM497 | added due to revision of the COM configuration |
| COM498 | added due to revision of the COM configuration |
| COM499 | added due to revision of the COM configuration |
| COM500 | added due to revision of the COM configuration |
| COM501 | added due to revision of the COM configuration |
| COM513 | added due to revision of the COM configuration |
| COM518 | added due to revision of the COM configuration |
| COM519 | added due to revision of the COM configuration |
| COM520 | added due to revision of the COM configuration |
| COM521 | added due to revision of the COM configuration |
| COM534 | clarification of reception deadline monitoring mechanism |
| COM535 | added configuration rule for F_OneEveryN filter |
| COM536 | harmonization of COM-RTE callback interface: Com_CbkInv |
| COM539 | restrict send out of I-PDUs while one call to Com_MainFunctionRouteSignals |
| COM540 | ComModuleDef |
| COM541 | configuration container ComGeneral added according to actual Meta-Model |
| COM544 | configuration container ComGwMapping added according to actual MetaModel |
| COM545 | configuration container ComGwSource added according to actual Meta-Model |
| COM546 | configuration container ComGwDestination added according to actual MetaModel |
| COM547 | configuration container ComGwSignalRef added according to actual MetaModel |
| COM548 | configuration container ComGwSourceDescription added according to actual MetaModel |
| COM549 | configuration container ComGwDestinationDescription added according to actual MetaModel |
| COM550 | configuration container ComGwIPduRef added according to actual MetaModel |
| COM551 | configuration container ComGwRef added according to actual MetaModel |
| COM552 | configuration container ComTimeoutNotification added according to actual MetaModel |
| COM553 | restricted configuration of uint8[n] signal types |
| COM554 | harmonization of COM-RTE callback interface: Com_CbkTxTOut |
| COM555 | harmonization of COM-RTE callback interface: Com_CbkRxAck |
| COM556 | harmonization of COM-RTE callback interface: Com_CbkRxTOut |
| COM557 | added API Com_InvalidateSignalGroup |
| COM558 | clarified the reception of an invalidated signal |
| COM559 | clarified the reception of an invalidated signal |
| COM560 | removed unnecessary configuration restrictions by splitting COM401 |

| *SWS Item* | *Rationale* |
|---|---|
| COM561 | clarified starting of deadline monitoring |
| COM562 | clarified minimum delay time handling |
| COM563 | added new transfer property TRIGGERED_ON_CHANGE |
| COM564 | transfer properties of signal groups clarified |
| COM565 | transfer properties of signal groups clarified |
| COM566 | transfer properties of signal groups clarified |
| COM567 | transfer properties of signal groups clarified |
| COM568 | handling of update-bits in signal gateway specified |
| COM569 | handling of update-bits in signal gateway specified |
| COM570 | handling of update-bits in signal gateway specified |
| COM571 | handling of update-bits in signal gateway specified |
| COM572 | handling of update-bits in signal gateway specified |
| COM573 | handling of update-bits in signal gateway specified |
| COM574 | defined handling of I-PDUs having a size smaller than expected |
| COM575 | defined handling of I-PDUs having a size smaller than expected |

# 12 Appendix A

In the following use cases with different transmission modes and the necessary configuration for these are shown. For the legend of the pictures see Chapter 7.4.3.6.

Use Case A1 shows an I-PDU which is sent out cyclically with a cycle time $t_c$. This I-PDU consists of signals which all have the Pending transfer property. It is configured that the send out takes place when the TMS evaluates to true.

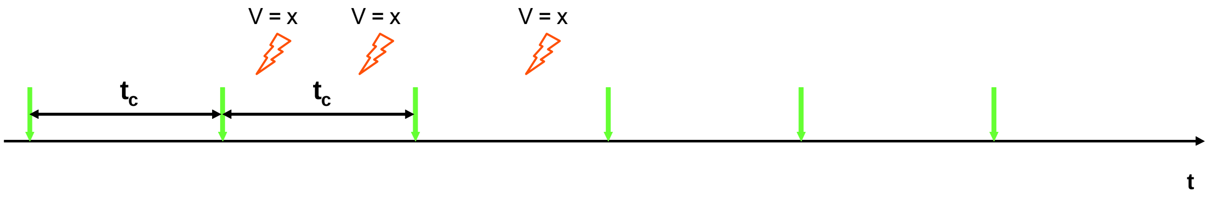

**Figure 20: Use Case A1, TM periodic (without TMS switch (see Chapter 7.4.3.3))**

| ***Relevant configuration items for the I-PDU transmission*** | |
|---|---|
| Object SIGNAL | |
| TRANSFERPROPERTY | Pending or Triggered (Triggered has no influence) |
| Object COM_FILTER | |
| FILTERALGORITHM | F_ALWAYS |
| Object I-PDU | |
| PROPERTY | Sent |
| Object TRANSMISSION MODE on filter true | |
| TIMEPERIOD | $t_c$ |
| MODE | periodic |
| NUMBER_OF_REPETITIONS | N/a |
| REPETITION_PERIOD | N/a |
| Object TRANSMISSION MODE on filter FALSE | |
| | N/a |

Because of the configuration of the parameter FILTERALGORITHM (F_ALWAYS) of the COM_FILTER, 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 FILTERALGORITHM parameter of the COM_FILTER is defined in the configuration for all the signals within the I-PDU with F_ALWAYS or if the COM_FILTER is not defined (shall not contribute to the evaluation of the TMS), see COM255.

Use Case A2 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 send outs is $t_d$. This I-PDU consists of signals which all have the Triggered transfer property. It is configured that the send out takes place when the TMS evaluates to true.

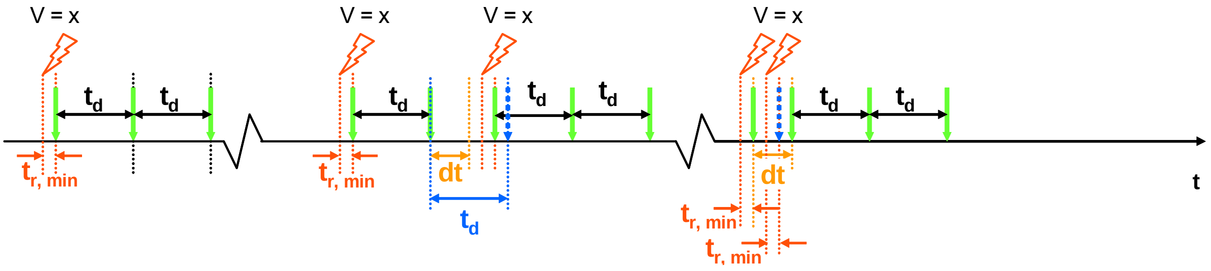

**Figure 21: Use Case A2, TM Direct/N-Times, here n = 3 (without TMS switch)**

| ***Relevant configuration items for the I-PDU transmission*** | |
|---|---|
| Object SIGNAL | |
| TRANSFERPROPERTY | Triggered |
| Object COM_FILTER | |
| FILTERALGORITHM | F_ALWAYS |
| Object I-PDU | |
| PROPERTY | Sent |
| Object TRANSMISSION MODE on filter true | |
| TIMEPERIOD | N/a |
| MODE | Direct/N-Times |
| NUMBER_OF_REPETITIONS | 3 |
| REPETITION_PERIOD | $t_d$ |
| Object TRANSMISSION MODE on filter false | |
| | N/a |

If there is a new send request by the upper layer before the last three sent outs have taken place, the new sent out is started and the rest of the last one is discarded.

Use case A3 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 Pending transfer property.

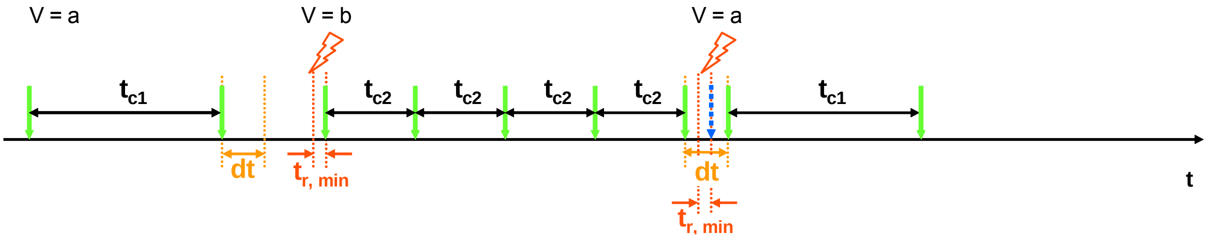

**Figure 22: Use case A3, TM periodic + periodic (with TMS switch)**

| ***Relevant configuration items for the I-PDU transmission*** | |
|---|---|
| Object SIGNAL | |
| TRANSFERPROPERTY | Pending or Triggered (Triggered has no influence) |
| Object COM_FILTER | |
| FILTERALGORITHM | all except F_ALWAYS and F_Never |
| Object I-PDU | |
| PROPERTY | Sent |
| Object TRANSMISSION MODE on filter true | |
| TIMEPERIOD | $t_{c1}$ |
| MODE | periodic |
| NUMBER_OF_REPETITIONS | N/a |
| REPETITION_PERIOD | N/a |
| Object TRANSMISSION MODE on filter false | |
| TIMEPERIOD | $t_{c2}$ |
| MODE | periodic |
| NUMBER_OF_REPETITIONS | N/a |
| REPETITION_PERIOD | N/a |

Because of the TMS switch caused by the new value v = b, the new cycle is started immediately and the new value is sent out. But nevertheless the minimum delay time dt has to be taken into account.
For the parameter FILTERALGORITHM of the configuration object COM_FILTER every in OSEK COM defined item can be used except F_ALWAYS and F_NEVER. These are:

- F_Always
- F_Never
- F_MaskedNewEqualsX
- F_MaskedNewDiffersX
- F_MaskedNewDiffersMaskedOld

- F_NewIsWithin
- F_NewIsOutside
- F_OneEveryN

If the FILTERALGORITHM F_OneEveryN is used not the value of the signal itself has an influence to the TMS but the number of send requests by the upper layer.

Use Case A4 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 upper layer. The time between two send outs is $t_d$. The I-PDU consists of signals which all have the Triggered transfer property.

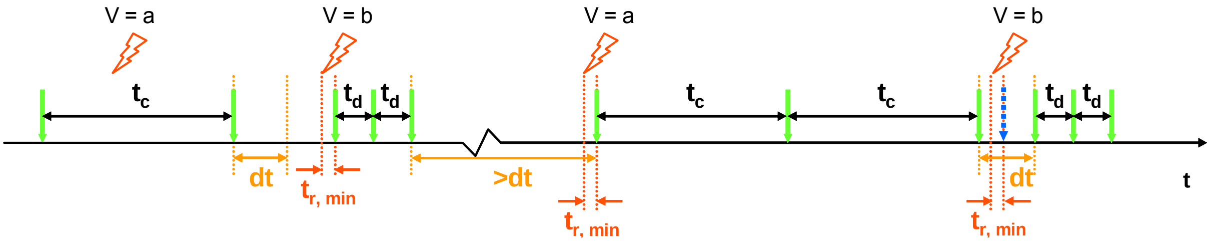

**Figure 23: Use Case 4, TM periodic + Direct/N-Times, here n = 3 (with TMS switch)**

| ***Relevant configuration items for the I-PDU transmission*** | |
|---|---|
| Object SIGNAL | |
| TRANSFERPROPERTY | Triggered |
| Object COM_FILTER | |
| FILTERALGORITHM | all except F_ALWAYS and F_Never |
| Object I-PDU | |
| PROPERTY | Sent |
| Object TRANSMISSION MODE on filter true | |
| TIMEPERIOD | $t_c$ |
| MODE | periodic |
| NUMBER_OF_REPETITIONS | N/a |
| REPETITION_PERIOD | N/a |
| Object TRANSMISSION MODE on filter false | |
| TIMEPERIOD | N/a |
| MODE | Direct/N-Times |
| NUMBER_OF_REPETITIONS | 3 |
| REPETITION_PERIOD | $t_d$ |

After the switch from Direct/N-Times to periodic the cycle is started immediately and the new value a is sent out (with respect of the minimum delay time dt).

Use case A5 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 upper layer it is also sent out directly three times. The time between two of these three send outs is always $t_{d.}$ The I-PDU consists of signals which all have the Triggered transfer property.

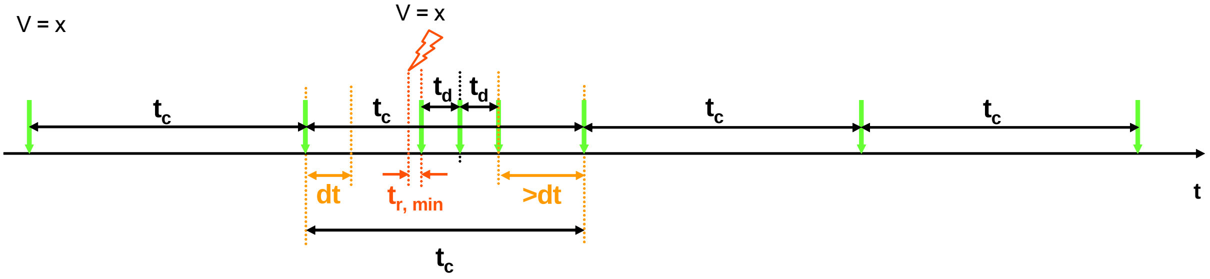

**Figure 24: Use Case 5a, TM Mixed, here n = 3 (without TMS switch)**

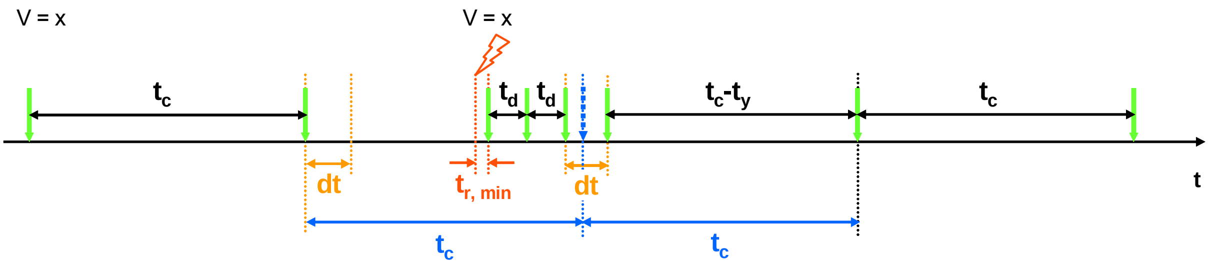

**Figure 25: Use Case 5b, TM Mixed, here n = 3 (without TMS switch), no phase shift**

| ***Relevant configuration items for the I-PDU transmission*** | |
|---|---|
| Object SIGNAL | |
| TRANSFERPROPERTY | Triggered |
| Object COM_FILTER | |
| FILTERALGORITHM | F_ALWAYS |
| Object I-PDU | |
| PROPERTY | Sent |
| Object TRANSMISSION MODE on filter true | |
| TIMEPERIOD | $t_c$ |
| MODE | Mixed |
| NUMBER_OF_REPETITIONS | 3 |
| REPETITION_PERIOD | $t_d$ |
| Object TRANSMISSION MODE on filter false | |
| | N/a |

If the next sent out caused by the periodic part of the Mixed transmission mode should take place within the timeout dt (minimum delay time) after a sent out of the

Direct/N-Times part, this sent out is delayed until the minimum delay time is elapsed. But after that the next time 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 A6 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 upper layer it is also sent out directly three times. The time between two of these three send outs is always $t_{d.}$ The I-PDU consists of signals which all have the Triggered transfer property.

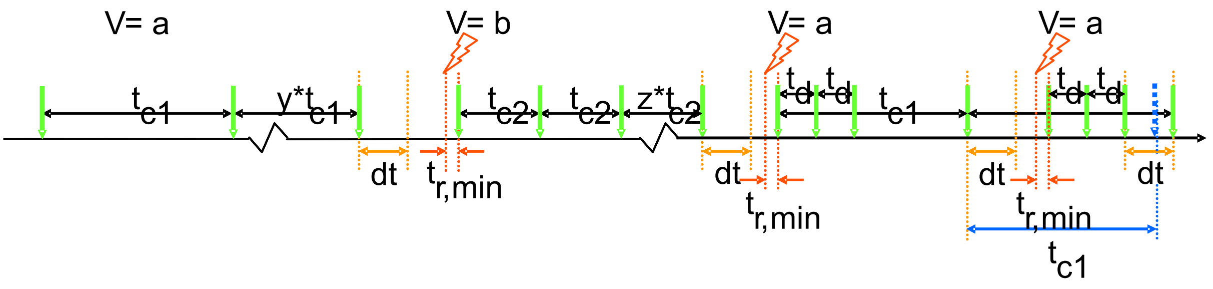

**Figure 26: Use Case 6, TM Mixed, here n= 3 + periodic (with TMS switch)**

| *Relevant configuration items for the I-PDU transmission* | |
|---|---|
| Object SIGNAL | |
| TRANSFERPROPERTY | Triggered |
| Object COM_FILTER | |
| FILTERALGORITHM | all except F_ALWAYS and F_Never |
| Object I-PDU | |
| PROPERTY | Sent |
| Object TRANSMISSION MODE on filter true | |
| TIMEPERIOD | $t_{c1}$ |
| MODE | Mixed |
| NUMBER_OF_REPETITIONS | 3 |
| REPETITION_PERIOD | $t_d$ |
| Object TRANSMISSION MODE on filter false | |
| TIMEPERIOD | $t_{c2}$ |
| MODE | periodic |
| NUMBER_OF_REPETITIONS | N/a |
| REPETITION_PERIOD | N/a |

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, this information is immediately distributed three times with $t_d$ and after that again the long cycle time is used.