Elektrobit

EB tresos® AutoCore Generic 8
Com Services documentation

release notes update for the Com module

product release 8.5.1

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
Email: info.automotive@elektrobit.com

Technical support

Europe	Japan	USA
Phone: +49 9131 7701 6060	Phone: +81 3 5577 6110	Phone: +1 888 346 3813

Support URL

https://www.elektrobit.com/support

Legal notice

Confidential and proprietary information

Table of Contents

1. Overview

This document provides you with the release notes to accompany an update to the `Com` module. Refer to the changelog Section 2.1, “Change log” for details of changes made for this update.

2. Com module release notes

- AUTOSAR R4.0 Rev 3
- AUTOSAR SWS document version: 4.2.0
- Module version: 6.3.38.B216231
- Supplier: Elektrobit Automotive GmbH

2.1. Change log

This chapter lists the changes between different versions.

Module version 6.3.38

2018-09-28

- Implemented extended basic support for uint64 and sint64 signal types

Module version 6.3.37

2018-06-22

- Improved handle ID wizard for Tx-I-PDUs which takes the priority of CAN messages into account
- Implemented defer Com transmission into Tx main function support
- Improved usage of critical sections for Com_SendDynSignal
- Implemented version compatibility check for EcuC library

Module version 6.3.36

2018-05-25

- Implemented COM TP-API support
- Implemented COM API Com_GetRxIPduBuffer which returns information about the receive IPdu buffer of a Rx IPdu

Module version 6.3.35

2018-05-07

- ASCCOM-2420 Fixed known issue: Wrongly generated code after reopening the project in EB tresos Studio

Module version 6.3.34

2018-04-20

- Implemented non-functional code improvements to fix Misra violations
- Implemented improvements for flexible allocation of PDUs to multiple Tx respectively Rx main functions including signal gateway
- ASCCOM-2410 Fixed known issue: Wrong COM_EXCLUSIVE_AREA is used

Module version 6.3.33

2018-03-16

- Implemented improvements for PduLengthType uint32 support

Module version 6.3.32

2018-02-16

- Implemented improvements for flexible allocation of PDUs to multiple Tx respectively Rx main functions
- ASCCOM-2359 Fixed known issue: Group signals are not routed by signal gateway
- ASCCOM-2360 Fixed known issue: Group signals are not routed by signal gateway
- ASCCOM-2375 Fixed known issue: Fragmented/interlaced signal groups are embedded wrong in Tx-I-PDUs
- Implemented improvements for routing of I-PDUs with different unused area values in between fragmented/interlaced signal groups
- Implemented improvements for routing of group signals where the signal group has parameter ComSignalGroupArrayAccess set to true
- Implemented improvements for flexible allocation of PDUs to multiple Tx respectively Rx main functions including signal gateway

Module version 6.3.31

2017-12-15

- ASCCOM-2299 Fixed known issue: Compilation error of Com_MainFunctionRouteSignals.c

Module version 6.3.30

2017-09-22

- Implemented improvements for flexible allocation of PDUs to several Tx respectively Rx main functions including signal gateway
- Switch from MISRA-C:2004 to MISRA-C:2012
- Introduced basic support for float64, uint64 and sint64 signal types

Module version 6.3.29

2017-08-25

- ASCCOM-2256 Fixed known issue: Wrong signal packing behaviour for 8-bit unaligned (group) signals with big endian

Module version 6.3.28

2017-07-28

- Implemented non-functional code improvements
- Implemented improvements for flexible allocation of PDUs to several Tx respectively Rx main functions (without singal gateway)

Module version 6.3.27

2017-06-30

- Implemented non-functional code improvements
- Added support for additional ComTransferProperties TRIGGERED_ON_CHANGE_WITHOUT_-REPETITION and TRIGGERED_WITHOUT_REPETITION
- Added support for flexible allocation of PDUs to several Tx respectively Rx main functions

Module version 6.3.26

2017-03-31

- Implemented non-functional code improvements

Module version 6.3.25

2017-03-03

- Improved description of configuration parameter ComTxModeTimeOffset
- Implemented non-functional code improvements
- ASCCOM-2202 Fixed known issue: Wrong invocation of ComNotification callbacks on transmission side (for non-AUTOSAR use-case only)
- Improved usage of critical sections

Module version 6.3.24

2017-02-03

- Implemented non-functional code improvements
- Implemented non-functional code improvements for optimizations
- Implemented non-functional code improvements

Module version 6.3.23

2016-12-02

- Implemented non-functional code improvements

Module version 6.3.22

2016-11-04

- ASCCOM-2157 Fixed known issue: Out of bounds access of unaligned Tx 16 bit signals / group signals
- Implemented non-functional code improvements
- Implemented non-functional code improvements to avoid compiler warnings

Module version 6.3.21

2016-09-23

- Added support for routing of fragmented/interlaced signal groups

Module version 6.3.20

2016-09-09

- Added support for fragmented/interlaced signal groups
- Adapted resource file for the scheduling of main functions to the split of `IpduM_MainFunction()` into `IpduM_MainFunctionRx()` and `IpduM_MainFunctionTx()`.

Module version 6.3.19

2016-08-05

- Implemented non-functional code improvements

Module version 6.3.18

2016-07-01

- Implemented non-functional code improvements

Module version 6.3.17

2016-05-25

- ASCCOM-2104 Fixed known issue: Unintended restarting of reception deadline monitoring with Com_IpduGroupControl()

Module version 6.3.16

2016-04-01

- Implemented non-functional code improvements

Module version 6.3.15

2016-02-05

- ASCCOM-2084 Fixed known issue: Nested MemMap section if TS_MERGED_COMPILE is activated
- Added support for Debug & Trace with custom header file configurable via parameter `BaseDbgHeaderFile`

Module version 6.3.14

2015-11-06

- ASCCOM-2071 Fixed known issue: Missing includes is source files in case option TS_MERGED_COMPILE is disabled

Module version 6.3.13

2015-10-09

- Implemented non-functional code improvements to avoid compiler warnings (Green Hills compiler for RH850 derivative) and static code analysis tools warnings
- ASCCOM-2054 Fixed known issue: Com receives incorrect values for Big-endian (group) signals if their msb is set to a multiple of eight
- Implemented non-functional code improvements to avoid compiler warnings

Module version 6.3.12

2015-06-19

- Implemented non-functional code improvements to avoid compiler warnings for specific optimization configurations
- ASCCOM-2021 Fixed known issue: Wrong signal handling on CPUs with big endianness architecture

Module version 6.3.11

2015-05-22

- ASCCOM-2015 Fixed known issue: Compilation error due to wrong usage of MemMap
- ASCCOM-2018 Fixed known issue: Wrong API name in integration requirement EB_INTREQ_Com_0001

Module version 6.3.10

2015-04-24

- Added support for ACG7 Transformer (COM)

  Note: With this feature, the handle ID policy for signals has changed. The conversion to the new handle ID policy can be simply applied by calling the *Calculate Handle IDs wizard* as described in `EB tresos Studio user's guide`.

Module version 6.3.9

2015-02-20

- Use AUTOSAR 4.x compliant memory section names for section CONFIG_DATA_UNSPECIFIED
- ASCCOM-1981 Fixed known issue: Invalid length calculation for dynamic length signals
- Implemented non-functional code improvements to fix Misra violations
- Implemented non-functional code improvements to ease readability

Module version 6.3.8

2015-01-07

- Added support for configurable mapping of PduR_IsValidConfig function to dedicated memory section
- Implemented non-functional code improvements for optimizations
- Implemented non-functional code improvements to fix Misra violations
- Implemented non-functional code improvements and update integration requirement EB_INTREQ_Com_-0002
- Implemented that reception filter MASKED_NEW_DIFFERS_MASKED_OLD always passes the filter criteria after a reception deadline monitoring timeout. For further details please refer to AUTOSAR Com SWS 4.1.1 SWS_Com_00793 and Bugzilla #52102
- Added support for ComInitialValueOnly
- Implemented range limitations for filter parameters. For further details please refer to AUTOSAR Com SWS 4.1.1, Bugzilla #52038 and #67828
- Changed signature of Com_RxIpduCallout and Com_TxIpduCallout. For further details please refer to AUTOSAR Com SWS 4.1.1, Bugzilla RfC #52342
- Removed AUTOSAR 3.x compliant symbolic name value macros and updated the logic to only provide AUTOSAR 4.0.2 compliant macros if macro COM_PROVIDE_LEGACY_SYMBOLIC_NAMES is defined

Module version 6.3.7

2014-10-03

- Added support for dynamic length signals

Module version 6.3.6

2014-08-07

- ASCCOM-1814 Fixed known issue: Build error due to missing file `Com_PBcfg.c` if code generation for Com is disabled and only post-build configuration is compiled
- Improved optimization: Allow disabling of Transmission Mode Selection
- Added support for signal group array access.
- ASCCOM-1856 Fixed known issue: Transmission of a signal gateway destination I-PDU fails if I-PDU shall be transmitted because of a change of signal group values
- ASCCOM-1836 Fixed known issue: Initial TMS evaluation fails for filter parameter with values larger than 0x7FFFFFFF
- ASCCOM-1874 Fixed known issue: Wrong warning if a byte array is larger than 8 bytes
- ASCCOM-1875 Fixed known issue: Error is issued when config time support is enabled and ComIPduTriggerTransmitCallout is configured
- Added support I-PDUs larger than 254 Bytes

Module version 6.3.5

2014-04-25

- ASCCOM-1770 Fixed known issue: Build fails if source files shall be built separately
- ASCCOM-1781 Fixed known issue: Sign extension for Rx-signals may fail if generated Rx-signal API is used
- ASCCOM-1785 Fixed known issue: Com module configuration generator may generate incorrect compiler abstractions for type definitions of module internal data types
- Implemented non-functional code improvements to fix Misra violations
- ASCCOM-1790 Fixed known issue: Missing checks of configuration parameters for transmission modes
- Implemented non-functional code improvements to avoid compiler warning in `Com_MainFunctionRouteSignals.c`
- ASCCOM-1803 Fixed known issue: Nested `MemMap` section if `TS_MERGED_COMPILE` is activated
- ASCCOM-1813 Fixed known issue: Choice container `ComGwDestination` is not set to changeable at post-build time

Module version 6.3.4

2013-10-11

- Implemented non-functional code improvements to use critical sections symmetrically
- Implemented non-functional code improvements to reduce function parameter in order to meet HIS metrics

- Implemented non-functional code improvements to fix Misra violations
- Replaced Rte memory sections and compile abstractions with Com memory sections and compile abstractions
- Improved allocation of post-build memory to ensure proper alignment
- Improved configuration checks of `ComTransferProperty`
- Implemented non-functional code improvements to defensive programming
- Implemented non-functional code improvements for optimizations
- Changed data type of `Com_StatusType` from `uint8` to an enumeration (used by `API Com_GetStatus()`)
- Updated behavior of `Com_SendSignalGroup()` regarding the calculation of the transmission mode of the related I-PDU
- Updated checks for timing parameters that resulting number of ticks matches are exact to configured values (according to TPS_ECUC_08010 of Specification of ECU Configuration AUTOSAR 4.1.1)
- Updated calculation of mask for filter `MASKED_NEW_DIFFERS_MASKED_OLD`
- Improved MCG to generate XML code for Binary Code Generation
- Added consistency checks for Com configuration
- Implemented non-functional code improvements to clean up service IDs
- Changed VSMDs to adhere to additional VSMD rules specified for AUTOSAR 4.1.1 related to attribute post-build changeable

Module version 6.3.3

2013-06-28

- ASCCOM-1685 Fixed known issue: `Com_TriggerTransmit()` incorrectly returns `E_OK` when all I-PDU groups to which an I-PDU belongs are stopped
- Changed timing behavior of starting of periodic I-PDUs according to the clarification in Bugzilla #52352
- Implemented a default value `0` for `ComTxModeTimeOffset` according to the clarification in Bugzilla #52352
- ASCCOM-1666 Fixed known issue: Restriction on I-PDU Trigger Transmit Callout
- ASCCOM-1699 Fixed known issue: `Com_RxIndication()` accesses invalid memory if called while uninitialized

Module version 6.3.2

2013-05-10

- Implemented check of published information signature to prevent loading of incompatible post-build configuration

Module version 6.3.1

2013-02-08

- Updated default value of filter of Tx-signals according to COM676 and COM677

Module version 6.3.0

2012-10-12

- Updated to AUTOSAR 4.0 Handle ID policy
- Added support of configuration parameter `ComRetryFailedTransmitRequests`
- Changed the top-level structure of the SWC description in the arxml files from `/AUTOSAR/Com` to `/AUTOSAR_Com`
- Added support for extended handling of configuration parameter `ComFirstTimeout`
- Added support of configuration parameter `ComEnableMDTForCyclicTransmission` (disable MDT for cyclic transmission)
- Added new API `Com_SwitchIpduTxMode()`
- Added separate I-PDU callout for `Com_TriggerTransmit()`
- Added support of reception of shorter I-PDUs (see deviation *Restricted support of small Rx-I-PDUs*)

Module version 6.2.2

2012-08-17

- Added definition of Exclusive Area Activation in Basic Software Module Description

Module version 6.2.1

2012-06-20

- Added support of usage of PbcfgM module

Module version 6.2.0

2012-03-16

- Modified `SchM_Enter/Exit()` calls to match AUTOSAR 4.0
- Added support for Tx-timeout handling for transmission mode `NONE`
- Updated naming scheme for `#defines` for symbolic name values to AUTOSAR 4.0 Rev 3 naming scheme
- Updated initial value for Rx-signals and signal groups when Rx Deadline Monitoring expired and I-PDU group is stopped
- Updated Com configuration to AUTOSAR 4.0 Rev 3
- Improved error message in case invalid references are configured

Module version 6.1.1

2012-02-17

- Internal module improvement. This module version update does not affect module functionality

Module version 6.1.0

2012-01-20

- Changed I-PDU group control API (removed AUTOSAR 3.x API, introduced AUTOSAR 4.x API)
- Updated Minimum Delay Time Monitoring behavior according to AUTOSAR 4.0
- ASCCOM-1435 Fixed known issue: Minimum Delay Time Monitoring is not canceled when Transmission Deadline Monitoring expires
- ASCCOM-1437 Fixed known issue: Possible omission of transmission of an I-PDU if `ComTransferProperty = TRIGGERED_ON_CHANGE` is used
- Added generation of BSWMD

Module version 6.0.1

2011-09-30

- ASCCOM-1318 Fixed known issue: Compilation fails when `Com.h` and `PduR.h` is included in the same file
- Changed optimization configuration from `ComTxFilterMaskedNewDiffersMaskOldEnable` to `Com_TxF_MaskNewDiffersMaskOld_En`
- Added reception of Rx-PDUs which are longer than configured

Module version 6.0.0

2011-09-02

- Initial AUTOSAR 4.0 version

2.2. New features

- Introduced extended basic support (filtering) for uint64 and sint64 signal types.

2.3. EB-specific enhancements

This chapter lists the enhancements provided by the module.

- [HisCom0029] Compile-time signal endianness optimization (extension to AUTOSAR specification)

  Description:

  If (at compile time) the ComSignalEndianness (COM157) is identical for every `ComSignal` of the COM module, the Com module does the following:

  - It uses only the code required for this endianness.
  - It does not make runtime checks for the signal endianness.
  - It does not store the endianness of every `ComSignal` individually.

- [HisCom0009] The `Com_ReceiveSignal` API as defined in COM198 is implemented as access macro

  Description:

  If this optimization is used, macros and a function is generated which extract the value of a signal from the I-PDU.

  Rationale:

  If these macros or the generated `Com_ReceiveSignal` API are used, the access to the value of the signal is faster.

- Optional reception filter for Signal Gateway

  Description:

  According to AUTOSAR, an Rx-signal is always gated via the Com Signal Gateway if a gateway relation for that signal exists. The update-bit is not considered here. In this implementation the AUTOSAR-conform behavior is achieved when the vendor-specific parameter `ComSigGwRxFilterEnable` is set to false which is the default value. However, when `ComSigGwRxFilterEnable` is set to true, the signal is only gated via the Com Signal Gateway, when the filter of the Rx-signal evaluates to true.

  Rationale:

Gated signals can be filtered.

- Optional Tx-signals with size zero

  Description:

  A zero size signal is a signal which is not represented in an I-PDU. However, an application can send a value which is evaluated by a filter configured for that signal. This feature is enabled when the vendor-specific parameter `ComTxZeroSignalEnable` is set to true and the default value is false.

  Rationale:

  Trigger-sending of an I-PDU without changing a value within the I-PDU.

- Support for signal group array access

  Description:

  The new APIs `Com_SendSignalGroupArray()` and `Com_ReceiveSignalGroupArray()` access the signal group value in the I-Pdu buffer.

  Rationale:

  Allows efficient access to signal groups, especially if the serialized data are also provide/required by another module, e.g. E2E module.

- Support for I-PDUs larger than specified by AUTOSAR

  Description:

  In contrast to AUTOSAR which restricts the configuration of signals / group signals into I-PDUs to at most 254 Bytes / large I-PDUs to at most 4095 bytes, the COM module supports I-PDUs up to 8191 Bytes. Further, the COM module allows the configuration of opaque signals / group signals (ComSignalType equals OPAQUE) with a length up to 8191 Bytes.

  With EcuC parameter `PduLengthTypeEnum` configured to `UINT32`, the module is basically allowed to handle PDUs with user data of more than 64 KiB. With the length restiction imposed by 8191 bytes neither the `UINT16` nor the `UINT32` range can be exploited.

  Rationale:

  Communication with Ethernet frames requires increased length of I-PDUs.

- Support for ACG7 Transformer (COM)

  Description:

  The ACG7 Transformer (COM) uses the post-build configuration and the serialization / de-serialization functions of the Com module.

Rationale:

Ensures consistent configuration between the Com and ComXf and allows the efficient serialization / de-serialization for signals and group signals due to the updated read / write library.

2.4. Deviations

This chapter lists the deviations of the module from the AUTOSAR standard.

- Only post-build configuration is supported

  Description:

  The Com module only supports configuration variant `VARIANT-POST-BUILD`. `VARIANT-PRE-COMPILE` and `VARIANT-LINK-TIME` are not supported.

  Requirements:

  COM606, COM607

- Signal invalidation is not supported (but is supported via RTE) (reference to product description: ASCPD-15)

  Description:

  Signal invalidation is not supported. However, the EB tresos AutoCore RTE is extended in order to provide the signal invalidation functionality based on the configuration of the `Com` module.

  Requirements:

  COM099, COM286, COM680, COM681, COM736, COM683, COM737, COM717, COM718, COM334, COM024, COM203, COM642, COM643, COM288, COM644, COM557, COM645, COM536, COM315_-Conf, COM391_Conf, COM314_Conf COM738, COM682, COM483, COM396, COM005, COM731

- Signal-based gateway: Optimization issue rate conversion not supported

  Description:

  COM386: Optimization issue: An I-PDU can be configured to be handled en bloc if it contains signals to be routed completely via a transmit I-PDU through a retention of the signal order and the signals endianness (related use case: rate conversion).

  The implementation lacks this optimization since AUTOSAR defines no configuration parameter to define to handle the I-PDU en bloc. Workaround: In case the I-PDU which should be gated is not used on the ECU, the whole I-PDU could be defined as a array. Note: If the size is greater than 8 bytes, the `init` value can not be defined with the AUTOSAR configuration parameters. If you use such a configuration, the PDU is directly copied to the Tx-PDU. Nevertheless, in case the signals shall be extracted by the Com module for an application, rate conversion can also be done. But in this case all signals are extracted from the Rx-PDU and packed into the Tx-PDU.

Requirements:

COM386

- Signal-based gateway: The `ComBitSize` of a received `ComSignal` can differ from the routed `ComSignal`

  Description:

  In contrast to AUTOSAR which specifies that the `ComBitSize` of the received and the routed `ComSignal` shall not differ, the Com module allows a `ComBitSize` of the routed `ComSignal`. This `ComBitSize` is greater than the `ComBitSize` of the received `ComSignal` with the constraint that both must be of the same `DataType`.

  Requirements:

  COM384

- SigGW: `ComGwSourceDescription` and `ComGwDestinationDescription` are not supported (reference to product description: ASCPD-17)

  Description:

  AUTOSAR SWS COM548_Conf and COM549_Conf define a `ComGwSourceDescription` and `ComGwDestinationDescription`. These descriptions allow adding/changing gateway relations post-build without the configuration of new signals `ComGwSourceDescription` and `ComGwDestinationDescription` that are not supported.

  Requirements:

  COM548_Conf, COM550_Conf, COM549_Conf

- `Com_SendSignal()` does not return `COM_SERVICE_NOT_AVAILABLE` in case the value of the signal does not fit into the PDU

  Description:

  The function `Com_SendSignal()` does not return `COM_SERVICE_NOT_AVAILABLE` in case the value of the signal does not fit into the PDU, but an error is reported to DET. However, the SWS states: Return value: `E_OK` - service has been accepted `COM_SERVICE_ NOT_AVAILABLE` - corresponding I-PDU group was stopped (or service failed due to development error). Therefore a `COM_SERVICE_ NOT_AVAILABLE` should be returned.

  Requirements:

  COM197

- The content of unfiltered elements of `ComSignal` in I-PDUs which are received deferred is not preserved until the next call to `Com_MainFunctionRx`

  Description

  In case the upper layer calls `Com_ReceiveSignal` or `Com_ReceiveSignalGroup` after an I-PDU with `ComIPduProcessing` *deferred* was received (a lower layer called `RxIndication()`), but before the deferred indications were signaled to the upper layer in `Com_MainFunctionRx`. This implementation does not behave like described in a note in the Com specification, but as follows: The content of the *old* I-PDU is not preserved until the next call to `Com_MainFunctionRx`. Immediately after the reception of the new I-PDU, the `Com_ReceiveSignal` and `Com_ReceiveSignalGroup` API provides the data of the new I-PDU's content. The only exception are non-group signals with a filter. As the filter is evaluated in the `Com_MainFunctionRx()`, the signal's value is retained until then.

  Requirements:

  COM198, COM201

- The Com does not check if an I-PDU is started if `Com_TxConfirmation` is called

  Description

  In contrast to AUTOSAR which states in Table 5 that a call to `Com_TxConfirmation()` shall be ignored in case of a stopped I-PDU, this version does not check if an I-PDU is started or stopped when the lower layer calls `Com_TxConfirmation()`.

  Rationale:

  In order not to lose speed to this check, it is assumed that no sporadic `Tx-Confirmation` appear once an Tx-I-PDU has been stopped.

  Requirements:

  COM124

- I-PDUs of gated signals are not sent out from the `Com_MainFunctionRouteSignals()` but from `Com_MainFunctionTx()`

  Description:

  In COM466 it is stated that I-PDUs that contain gated signals with triggered transfer property shall be sent from the `Com_MainFunctionRouteSignals()` according to their transmission modes. In the implementation, an I-PDU is never sent from the `Com_MainFunctionRouteSignals()`. In case a gated signal has the triggered transfer property, the I-PDU is sent out in the following `Com_MainFunctionTx()`. However, both `Com_MainFunctionRouteSignals()` and `Com_MainFunctionTx()` are scheduled functions which therefore do not have the *event-based* character as does `Com_SendSignal()` with the triggered transfer property.

  Note: The transmission request from the signal gateway is also cleared, independent if a transmission request was issued and/or a transmission request failed if the following conditions are met:

  - `ComRetryFailedTransmitRequest` is enabled.
  - A transmission deadline monitoring is configured for the I-PDU and the deadline monitoring expires in the following `Com_MainFunctionTx()`.

  Requirements:

  COM539

- Data sequence control and Communication protection not supported (reference to product description: ASCPD-22, ASCPD-23)

  Description:

  Data sequence control (I-PDU counter) and communication protection (replication of I-PDUs) are not supported.

  Requirements:

  COM687, COM688, COM587, COM588, COM590, COM727, COM596, COM597, COM726, COM592_-Conf, COM003_Conf, COM593_Conf, COM594_Conf, COM595_Conf, COM599_Conf, COM600_Conf, COM601_Conf, partly COM787, COM731

- *Transmit Cancellation* not supported (reference to product description: ASCPD-24)

  Description:

  The cancelation of transmission requests is not supported.

  Requirements:

  COM708, COM670, COM709_Conf

- Restricted support of small Rx-I-PDUs

  Description:

  According to the AUTOSAR COM SWS chapter *Signal indication (Unpacking of I-PDUs)* it is specified that it is allowed that smaller than expected Rx-I-PDUs can be received (configured). In such a case partly or not received signals/signal groups shall not be updated and no notification via `ComNotification` shall take place.

  However, the implementation behaves as follows:

  - The received data length (`PduInfoPtr`->`SduLength`) is copied into the Com-internal I-PDU buffer. If a signal or signal groups are received only partly, these are also updated partly. Signals/signal groups which are not received at all are not updated. If the I-PDU contains a dynamic length signal the API

  Com_ReceiveDynSignal() does not copy data and 0 is returned in the length parameter. If a dynamic length signal is used within the signal gateway, the length of the corresponding Tx dynamic length signal is also set to 0.

  - `ComNotification` is invoked for all signals or signal groups that belong to this Rx-I-PDU.

  Workaround 1:

  - For a smaller Rx-I-PDU, for which it is expected that a signal or signal group is only partly updated: Configure an I-PDU callout which updates the partly received signal/signal groups with a proper value (either last received value or initial value).
  - Design the applications in a way that they can handle `ComNotification` of signals which are not or only partly received.

  Workaround 2:

  - Provide an application for each expected size of the Rx-I-PDU.
  - For each expected size of the Rx-I-PDU configure a Rx-I-PDU in the Com module.
  - Create a mapping between the additional applications and Rx-I-PDUs.
  - Configure an Rx-I-PDU callout with the large I-PDU which invocates `Com_RxIndication` of the respective smaller Rx-I-PDU and returns `FALSE` in case a shorter I-PDU is received.

  Rationale:

  In general this limitation allows a more efficient implementation for I-PDUs which are received completely. Workarounds are available if this feature is required.

  Requirements:

  COM574, COM575

- No support of Debug & Trace

  Description:

  Tracing of global variables is not supported.

  Requirements:

  COM745, COM746, COM747, COM748

- Non-compliant deviations in vendor-specific module definition file

  Description:

  The vendor-specific module definition file (VSMD) has non-compliant deviations to the AUTOSAR specification:

Violations against Rule EcucSws_6008: The LOWER-MULTIPLICITY of an element in the VSMD must be bigger or equal and the UPPER-MULTIPLICITY must be equal or less than in the StMD

- StMD-Node: /AUTOSAR/Com/ComConfig/ComIPdu/ComIPduTriggerTransmitCallout

Rationale: Configuration shall be equal with ComCallout, see also http://www.autosar.org/bugzilla/show_bug.cgi?id=53200#c50.

- StMD-Node: /AUTOSAR/Com/ComConfig/ComTimeBase

Rationale: Optionality of the ComTimeBase container is used to enable or disable the multiple main function support.

Violations against Rule EcucSws_1007: For integer and float parameters the MIN values must be >= and the MAX values <= as in the StMD.

- StMD-Node: /AUTOSAR/Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter/ComFilterMask
- StMD-Node: /AUTOSAR/Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter/ComFilterMax
- StMD-Node: /AUTOSAR/Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter/ComFilterMin
- StMD-Node: /AUTOSAR/Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter/ComFilterOffset
- StMD-Node: /AUTOSAR/Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter/ComFilterPeriod
- StMD-Node: /AUTOSAR/Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter/ComFilterX
- StMD-Node: /AUTOSAR/Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode/ComTxModeTimeOffset
- StMD-Node: /AUTOSAR/Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode/ComTxModeTimeOffset
- StMD-Node: /AUTOSAR/Com/ComConfig/ComSignal/ComBitPosition
- StMD-Node: /AUTOSAR/Com/ComConfig/ComSignal/ComSignalLength
- StMD-Node: /AUTOSAR/Com/ComConfig/ComSignal/ComUpdateBitPosition
- StMD-Node: /AUTOSAR/Com/ComConfig/ComSignal/ComFilter/ComFilterMask
- StMD-Node: /AUTOSAR/Com/ComConfig/ComSignal/ComFilter/ComFilterMax
- StMD-Node: /AUTOSAR/Com/ComConfig/ComSignal/ComFilter/ComFilterMin
- StMD-Node: /AUTOSAR/Com/ComConfig/ComSignal/ComFilter/ComFilterX

  - StMD-Node: `/AUTOSAR/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComUpdateBitPosition`
  - StMD-Node: `/AUTOSAR/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComBitPosition`
  - StMD-Node: `/AUTOSAR/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComSignalLength`
  - StMD-Node: `/AUTOSAR/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter/ComFilterMask`
  - StMD-Node: `/AUTOSAR/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter/ComFilterMax`
  - StMD-Node: `/AUTOSAR/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter/ComFilterMin`
  - StMD-Node: `/AUTOSAR/Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter/ComFilterX`

  Rationale: Limitations are necessary to ensure valid configuration of filters (see Limitation: Restriction on `ComFilter` values). The range of `ComTxModeTimeOffset` has been extended to allow a backward compatible configuration for starting of I-PDU groups. Value range for parameters ComBitPosition, ComUpdateBitPosition, and ComSignalLength of ComSignals and ComSignalGroups has been extended to support I-PDUs larger than 254 Bytes.

  Violations against Rule EcucSws_1014: Additional vendor specific parameter definitions (using ParameterTypes), container definitions and references shall be added to the VSMD according to the alphabetical order.

  - StMD-Node: `/AUTOSAR/Com`
  - StMD-Node: `/AUTOSAR/ComConfig/ComSignal`
  - StMD-Node: `/AUTOSAR/ComGeneral`

  Rationale: Additional vendor specific parameter definitions have a specific order in the GUI which may differ to the alphabetical order.

- No consistency check between code files and header files

  Description:

  The inter-module version checks as specified by the Com SWS are not implemented.

  Rationale:

  - The required compile-time version checks would result in an inflexible, hardly integratable basic software stack.
  - EB tresos AutoCore is an already integrated product.

- The project handling of EB tresos Studio provides means to enforce that only modules with the same EB tresos AutoCore release version can be added to the project.

Requirements:

COM673

- Behavior of `Com_IpduGroupControl`

Description:

According to COM787, the Com module shall initialize the following attributes of an I-PDU as result of a call `Com_IpduGroupControl` (independent from parameter `Initialize`):

1. `ComMinimumDelayTime` of I-PDUs in transmission mode `DIRECT` or `MIXED`
2. Timeout attributes of I-PDUs for deadline monitoring aspect: all timeout timers (`ComFirstTimeout`, `ComTimeout`) shall restart.
3. All included update-bits shall be cleared.
4. Reset `OCCURRENCE` of filter with `ComFilterAlgorithm ONE_EVERY_N`.
5. Set the I-PDU counter to `0` for I-PDUs with `ComIPduDirection` configured to `SEND`.
6. Accept for I-PDUs with `ComIPduDirection` configured to `RECEIVED` any next incoming I-PDU counter.

However the implementation does not support I-PDU counter, therefore the items 5 and 6 are not initialized. See also deviation *Data sequence control and Communication protection not supported*. The items 1, 3 and 4 are only initialized as the result of a call `Com_IpduGroupControl` with parameter `Initialize` set to true. In contrast to COM222, the shadow buffers of included `RECEIVED` signal groups keeps unchanged if `Com_IpduGroupControl` is called (independent from parameter `Initialize`).

Additionally to the described behavior above, `ComTxModeTimePeriod` and `ComTxModeTimeOffset` of I-PDUs in `PERIODIC` or `MIXED` transmission mode are always respected (independent from parameter `Initialize`).

Rationale:

Requirement COM787 is not backward compatible, see also Bugzilla issue http://www.autosar.org/bugzilla/show_bug.cgi?id=48891.

The description of parameter `Initialize` of the `API Com_IpduGroupControl` (COM751) says: “flag to request initialization of the I-PDUs which are newly started”. That describes well the implemented behavior, but does not imply that I-PDU attributes are initialized, although parameter `Initialize` set to false.

The description of `ComTxModeTimeOffset` refers to `Com_IpduGroupControl` in general and is not limited to parameter `Initialize` set to true.

Requirements:

COM787, COM222

- No generation of symbolic name value into `Com_Cfg.h`

  Description:

  Several requirements claim that the symbolic names for the Com Handle IDs shall be published via `Com_Cfg.h`. However, the symbolic name values are provided in `Com_SymbolicNames_PBcfg.h` which is also included in `Com.h`.

  Rationale:

  - Requirement is a deviation against TPS_ECUC_02108 of Specification of ECU Configuration which says that the symbolic name values shall be generated into the module header file.
  - Requirement is a deviation against SWS_BSW_00200 of SWS General Specification of Basic Software Modules AUTOSAR 4.1 Rev 1, which says that symbolic name values shall be imported through the header of the BSW module that provides the value.
  - Shall be removed in future AUTOSAR releases, see http://www.autosar.org/bugzilla/show_bug.cgi?id=60888

  Requirements:

  COM174, COM126, COM163, COM044, COM521

- No support of dynamic length signals in signal groups

  Description:

  Dynamic length signals are only supported as signals. They are not supported in a group signal.

  Rationale:

  The implementation uses `Com_UpdateShadowSignal()` and `Com_ReceiveShadowSignal()` for the access of group signals. Since AUTOSAR does not define an equivalent API for access dynamic group signals, it is not possible to support dynamic length signals for group signals.

  Requirements:

  COM127

- No support of zero size signals / group signals with transfer property PENDING

  Description:

In contrast to AUTOSAR which allows zero size signals / group signals for transfer properties TRIGGERED, PENDING, and TRIGGERED_WITHOUT_REPETITION, only transfer property TRIGGERED and TRIGGERED_WITHOUT_REPETITION is supported.

Requirements:

COM762

- No need for configuration of ComTxModeTrue or ComTxModeFalse

  Description:

  In contrast to AUTOSAR which states that every ComTxModeTrue or ComTxModeFalse that is a potential result of a configured/ calculated TMS must be configured, the COM module assumes ComTransmissionMode NONE if one of these transmission modes is not configured but evaluated as a result of TMS. Note that at least one of the containers ComTxModeTrue or ComTxModeFalse must be configured at all.

  Requirements:

  COM465

- Overlapping of ComSignals / ComGroupSignals

  Description:

  In contrast to AUTOSAR which states that ComSignal / ComGroupSignal are not allowed to overlap each other, the COM module allows the configuration of overlapped ComSignals / ComGroupSignals.

  Requirements:

  COM102

- Configurable callback / callout functions are not provided in Com_Cbk.h

  Description:

  In contrast to AUTOSAR which states that the configurable callback and callout functions shall be provided in header file Com_Cbk.h, the COM module does not declare these functions. Instead, it declares and calls these external functions in an internal Com compilation unit.

  Rationale:

  These functions are usually generated / implemented by the Rte which also generates adequate function declarations. The linker then is able to resolve the function calls and the adequate function definitions in Rte. See also RTE Specification 4.2.1 Section 5.9.2.1 Call-backs for communication over AUTOSAR COM.

  Requirements:

  COM731

- Optimization parameter ComSignalGwEnable for scaling down signal gateway to no size

  Description:

  In contrast to AUTOSAR which states that the signal gateway of the AUTOSAR COM module shall scale down to no size if no signal routing functionality is needed, the integrator shall disable the vendor specific optimization parameter ComSignalGwEnable to get the same effect.

  Requirements:

  COM370
- Runtime error COM_E_SKIPPED_TRANSMISSION is not supported

  Description:

  In case a large I-PDU is currently transmitted and the same I-PDU is triggered for transmission again, the AUTOSAR COM skips the additionally send request but does not report the runtime error COM_E_-SKIPPED_TRANSMISSION.

  Requirements:

  SWS_Com_00863
- PduR_ComTpTransmit is called for large I-PDUs

  Description:

  AUTOSAR specifies that for transmissions of large I-PDUs the PduR API PduR_ComTransmit has to be called. PduR_ComTpTransmit is used instead for large I-PDUs and PduR_ComTransmit for normal I-PDUs.

  Requirements:

  COM759, COM760, COM467, COM773, COM698, COM138

2.5. Limitations

This chapter lists the limitations of the module. Refer to the module references chapter *Integration notes*, subsection *Integration requirements* for requirements on integrating this module.

- Implementation-specific restrictions

  Description:

  There are some implementation-specific restrictions which are listed for completeness only, as they are most probably irrelevant for the intended use of the module:

- The maximum number of signals allowed is 65534.
- The maximum number of Rx/Tx I-PDUs allowed is 65534.
- The maximum number of callouts configured is 65534.
- The sum of the lengths of all byte-arrays which are sent via the Com module must not exceed 65535 bytes.
- The number of signals and signal group members, signal groups, notifications per I-PDU must not exceed 254.

- Discrepancy between ISO C90 standard and AUTOSAR ranges for signed integers

  Description:

  C90 only allows signed integer values to have the following range:

  - sint8: -127 / 127, or -(2^7 -1) / 2^7-1
  - sint16 -32767 / 32767, or -(2^15 -1) / 2^15+1
  - sint32 -2147483647 / 214743647, or -(2^31 -1) / 2^31 -1
  - sint64 -9223372036854775807 / 9223372036854775807, or -(2^63 -1) / 2^63 -1

  AUTOSAR, in the opposite, allows negative numbers to be one less:

  - sint8: -128 / 127, or -(2^7) / 2^7-1
  - sint16 -32768 / 32767, or -(2^15) / 2^15+1
  - sint32 -2147483648 / 214743647, or -(2^31) / 2^31-1
  - sint64 -9223372036854775808 / 9223372036854775807, or -(2^63) / 2^63-1

  Rationale:

  In AUTOSAR, it is defined that AUTOSAR only supports platforms which use the 2's complement as basis for their architecture (and therefore support the AUTOSAR number range). This implementation implements the full AUTOSAR number range. C90-compliant compilers may legally facilitate code which shows undefined behavior in case the number-range as defined in C90 is left. That is if the minimum signed integer numbers as defined by AUTOSAR are used. Therefore this implementation relies upon the following:

  - A platform is used which uses the 2's complement or a platform is used which can handle the whole AUTOSAR number range.
  - A C-compiler is used which can handle the AUTOSAR-defined minimum signed integer numbers and facilitates code which behaves properly.

- Limitation/extension on configuration of `ComFirstTimeout`

Description:

According to the AUTOSAR SWS Com requirement COM716, the AUTOSAR Com module shall not monitor the reception of this signal or of a signal group from the start of the corresponding I-PDU until the first reception. This behavior applies if the configuration parameter `ComFirstTimeout` for a signal or signal group is omitted or configured to `0`.

The implementation behaves as following regarding the configuration parameter `ComFirstTimeout` for a signal or signal group:

- If configured to `0`: as defined in COM716
- If omitted: `ComTimeout` is used for `ComFirstTimeout`

The default behavior for `ComFirstTimeout` is disabled.

Rationale:

- Eases configuration (otherwise `ComFirstTimeout` has to be configured for every signal/signal group)
- Does not restrict a use-case since configuration for starting of reception deadline monitoring with first reception of the I-PDU is possible.

Requirements:

COM716

- Restriction on `ComFilter` values

Description:

In contrast to AUTOSAR 4.0 Rev 3 where the `ComFilter` values (i.e. `ComFilterX`, `ComFilterMask`, `ComFilterMax`, `ComFilterMin`) for Com signals/signal groups shall have a configurable value range within [0, 18446744073709551615], the Com supports for `ComFilterMax` and `ComFilterMin` a value range within [-2147483648, 4294967295] and for `ComFilterX` and `ComFilterMask` a value range within [-9223372036854775807, 18446744073709551615]. That is, only the least significant 32 or 64 bits are significant. Note: A Bugzilla item exists for this issue: http://www.autosar.org/bugzilla/show_bug.cgi?id=52038.

Rationale:

- Eases configuration
- For 64 bit signals/group signals the ComFilterAlgorithms are restricted where only `ComFilterX` and `ComFilterMask` are required to be set to a different range than `ComFilterMax` and `ComFilterMin`.

  - There is no use-case for a filter ONE_EVERY_N where ComFilterOffset and ComFilterPeriod have to be higher than 2^32-1.

  Requirements:

  COM147_Conf, COM235_Conf, COM317_Conf, COM318_Conf

- Restriction on 64 bit signals/group signals

  Description:

  The following restrictions for signals/group signals with ComSignalType configured to UINT64 apply:

  - The ComBitPosition is restricted to be byte aligned.
  - The ComBitSize is restricted to be a multiple of 8 bit.
  - The ComFilterAlgorithm is limited to ALWAYS, NEVER, ONE_EVERY_N, MASKED_NEW_DIFFERS_X and MASKED_NEW_EQUALS_X.
  - For the ComFilterAlgorithms MASKED_NEW_DIFFERS_X and MASKED_NEW_EQUALS_X, only the bits with respect to the configured ComBitSize are taken into account for the filter evaluation.
  - The ComFilterAlgorithm for zero size signals / group signals is limited to ALWAYS and NEVER.

  The following restrictions for signals/group signals with ComSignalType configured to SINT64 apply:

  - The ComBitPosition is restricted to be byte aligned.
  - The ComBitSize is restricted to be 64 bit (zero size signals / group signals are not supported).
  - The ComFilterAlgorithm is limited to ALWAYS, NEVER, ONE_EVERY_N, MASKED_NEW_DIFFERS_X and MASKED_NEW_EQUALS_X.

  Requirements:

  COM675, COM602, COM170_Conf, COM352, COM325, COM764, COM273, COM603, COM302, COM303, COM763, COM222 COM324, COM793

- Limitation on transmission behaviour for large Tx I-PDUs

  Description:

  The transmission behaviour of large Tx I-PDUs is limited to the following points:

  - only one transmission mode can be enabled.
  - only transmission mode DIRECT with no repetitions (ComTxModeNumberOfRepetitions set to 0) is allowed.
  - all large Tx I-PDU transmission requests are deferred to the next execution of the Com transmission main function (parameter ComDeferTx2MainFunc must be enabled).

- all update-bits of all contained signals and signal groups of large Tx I-PDUs are only cleared if PduR_ComTpTransmit returned E_OK and the I-PDU was successfully confirmed (parameter `ComTxIPduClearUpdateBit` must be configured to Confirmation).

Due to that limitation no transmission mode selection based on Tx transmission filter evaluation is possible for large Tx I-PDUs.

Requirements:

COM694, COM602, COM325, COM380, COM439, COM231, COM330, COM767, COM734, COM768, COM762, COM135, COM741, COM769, COM742, COM770, COM326, COM676, COM678, COM679, COM245, COM763, COM238, COM244, COM495, COM582, COM467, COM279, COM305, COM494, COM392, COM776, COM787, COM222, COM223, COM228, COM229, COM789, COM696, COM308, COM739, COM388, COM492, COM784, COM813, COM605, COM032, COM779, COM625, COM629