# COM

|  |
| --- |
| * SRS: Xác định cái gì cần phải được thực hiện. * SWS: Mô tả làm thế nào để thực hiện cái đã được xác định trong SRS. |

## Giới thiệu và Tổng quan về chức năng

Đặc điểm kỹ thuật này là Đặc điểm kỹ thuật phần mềm mô-đun AUTOSAR COM. Nó dựa trên AUTOSAR COM SRS [7]. Nó quy định cách các yêu cầu của AUTOSAR COM SRS sẽ được thực hiện. Điều đó có nghĩa là chức năng và API của mô-đun AUTOSAR COM được mô tả trong tài liệu này.

Trong AUTOSAR Layered Architecture, mô-đun AUTOSAR COM được đặt giữa RTE và Bộ định tuyến PDU, xem [1].

Mô-đun AUTOSAR COM có nguồn gốc từ [17]. Để biết chi tiết, xem Chương 7.2.1. Mô-đun AUTOSAR COM cung cấp chức năng cổng tín hiệu. Để biết chi tiết, xem Chương 7.2.5.

Những đặc điểm chính:

* Cung cấp giao diện dữ liệu theo tín hiệu cho RTE.
* Đóng gói các tín hiệu AUTOSAR vào I-PDU để truyền.
* Giải nén các I-PDU đã nhận và cung cấp các tín hiệu đã nhận cho RTE.
* Định tuyến các tín hiệu từ I-PDU đã nhận vào I-PDU để truyền.
* Định tuyến các nhóm tín hiệu từ I-PDU đã nhận vào I-PDU để truyền.
* Kiểm soát truyền thông (bắt đầu/dừng các nhóm I-PDU).
* Nhân bản các yêu cầu gửi.
* Đảm bảo khoảng cách tối thiểu giữa các I-PDU truyền.
* Giám sát các tín hiệu nhận (hết thời gian chờ của tín hiệu).
* Cơ chế lọc cho các tín hiệu đến.
* Các cơ chế thông báo khác nhau.
* Cung cấp các giá trị khởi tạo và chỉ thị cập nhật.
* Chuyển đổi thứ tự byte.
* Mở rộng dấu.
* Hỗ trợ hai chế độ truyền khác nhau cho mỗi I-PDU.
* Cổng dựa trên tín hiệu.
* Hỗ trợ các kiểu dữ liệu lớn và có độ dài động.
* Hỗ trợ bộ đếm I-PDU và nhân bản I-PDU.

## Từ viết tắt, viết tắt và định nghĩa

### Từ viết tắt và viết tắt

**Chữ viết tắt và Giải thích**

| **Chữ viết tắt** | **Mô tả** |
| --- | --- |
| AUTOSAR COM | Mô-đun AUTOSAR COM được phát triển từ OSEK COM [17]. Chi tiết xem Chương 7.2.1. |
| DM | Giám sát thời hạn (Deadline Monitoring), chi tiết xem Chương 7.3.6 |
| I-PDU | Gói dữ liệu tầng tương tác (Interaction Layer Protocol Data Unit). Một I-PDU chứa các tín hiệu. Nó được định nghĩa trong [17]. |
| L-PDU | Gói dữ liệu tầng liên kết (Data Link Layer Protocol Data Unit). Trong AUTOSAR, Tầng liên kết dữ liệu tương đương với Tầng Trừu tượng hóa Phần cứng Truyền thông và Tầng Trừu tượng hóa Vi điều khiển. |
| MDT | Mô tả chi tiết về Bộ đếm thời gian trễ tối thiểu (Minimum Delay Timer) có thể được tìm thấy trong [17]. Xem thêm Chương 7.3.5.5. |
| OSEK COM | Hệ thống mở và các giao diện tương ứng cho điện tử ô tô – truyền thông [17]. |
| PDU Router | Bộ định tuyến PDU là một mô-đun chuyển các I-PDU từ một mô-đun này sang mô-đun khác. Bộ định tuyến PDU có thể được sử dụng cho các hoạt động cổng và cho các mục đích định tuyến nội bộ. |
| SDU | Đơn vị dữ liệu dịch vụ (Service Data Unit). Mô tả xem [1] Chương 4. |
| TM | Chế độ truyền (Transmission Mode) |
| TMC | Điều kiện chế độ truyền (Transmission Mode Condition), xem Chương 7.3.3.2 |
| TMS | Bộ chọn chế độ truyền (Transmission Mode Selector), xem Chương 7.3.3.2 |

### Định nghĩa

**Định nghĩa**

| **Thuật ngữ** | **Mô tả** |
| --- | --- |
| Confirmation | Với một Confirmation (Xác nhận), Bộ định tuyến PDU báo cáo rằng một yêu cầu bởi mô-đun AUTOSAR COM đã được hoàn thành thành công. Đây là phản ứng đối với một yêu cầu của COM, ví dụ khi một PDU đã được truyền thành công. |
| Data Invalid Value | Giá trị dữ liệu không hợp lệ được gửi bởi mô-đun AUTOSAR COM để chỉ ra rằng thành phần phần mềm AUTOSAR bên gửi không thể cung cấp giá trị hợp lệ. |
| Dynamic Length Signal | Tín hiệu có độ dài động là tín hiệu có độ dài có thể thay đổi trong thời gian chạy. |
| Dynamic Length I-PDU | I-PDU có độ dài động là I-PDU chứa một tín hiệu có độ dài động. Độ dài của nó thay đổi tùy thuộc vào độ dài của tín hiệu có độ dài động bao gồm trong đó. |
| Group Signal | Tín hiệu nhóm là tín hiệu được chứa trong một nhóm tín hiệu. |
| Indication | Một Indication (Chỉ báo) là thông tin không đồng bộ từ Bộ định tuyến PDU đến COM, ví dụ như để xác nhận rằng một cái gì đó đã được nhận. |
| Init Value | Các I-PDU và tín hiệu được đặt về Giá trị khởi tạo bởi mô-đun AUTOSAR COM sau khi khởi động. Giá trị này được sử dụng cho đến khi nó bị ghi đè. |
| I-PDU Group | Nhóm I-PDU là một tập hợp tùy ý của các I-PDU trong mô-đun AUTOSAR COM. |
| Inter-ECU Communication | Giao tiếp giữa hai hoặc nhiều ECU, ví dụ như thông qua mạng CAN. |
| Intra-ECU Communication | Giao tiếp giữa các thành phần phần mềm cư trú trên cùng một ECU. |
| Large Signal | Tín hiệu lớn là tín hiệu quá lớn để vừa với một L-PDU duy nhất của giao thức truyền thông cơ bản. |
| Large I-PDU | I-PDU lớn là I-PDU không vừa với một L-PDU duy nhất của giao thức truyền thông cơ bản. |

## Related Documentation

### Deliverables of AUTOSAR

* [1] AUTOSAR Layered Architecture

AUTOSAR\_EXP\_LayeredSoftwareArchitecture.pdf

* [2] Specification of Communication Stack Types

AUTOSAR\_SWS\_CommunicationStackTypes.pdf

* [3] General Requirements on Basic Software Modules

AUTOSAR\_SRS\_BSWGeneral.pdf

* [4] Basic Software UML Model

AUTOSAR\_MOD\_BSWUMLModel.eap

* [5] Specification of Standard Types

AUTOSAR\_SWS\_StandardTypes.pdf

* [6] Specification of the Virtual Functional Bus

AUTOSAR\_EXP\_VFB.pdf

* [7] Requirements on Communication

AUTOSAR\_SRS\_COM.pdf

* [8] Software Component Template

AUTOSAR\_TPS\_SoftwareComponentTemplate.pdf

* [9] Requirements on Gateway

AUTOSAR\_SRS\_Gateway.pdf

* [10] Specification of PDU Router

AUTOSAR\_SWS\_PDURouter.pdf

* [11] Specification of Operating System

AUTOSAR\_SWS\_OS.pdf

* [12] System Template

AUTOSAR\_TPS\_SystemTemplate.pdf

* [13] Specification of RTE Software

AUTOSAR\_SWS\_RTE.pdf

* [14] Specification of ECU Configuration

AUTOSAR\_TPS\_ECUConfiguration.pdf

* [15] Specification of Communication Manager

AUTOSAR\_SWS\_COMManager.pdf

* [16] AUTOSAR Basic Software Module Description Template

AUTOSAR\_TPS\_BSWModuleDescriptionTemplate.pdf

* [19] Specification of CAN Transport Layer

AUTOSAR\_SWS\_CANTransportLayer.pdf

* [20] Specification of FlexRay Transport Layer

AUTOSAR\_SWS\_FlexRayTransportLayer.pdf

* [21] List of Basic Software Modules

AUTOSAR\_TR\_BSWModuleList.pdf

* [22] Generic Structure Template

AUTOSAR\_TPS\_GenericStructureTemplate.pdf

* [23] General Specification of Basic Software Modules

AUTOSAR\_SWS\_BSWGeneral.pdf

### Related Standards and Norms

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

### Related Specification

AUTOSAR cung cấp một Đặc tả Chung về các mô-đun Phần mềm Cơ bản [23] (SWS BSW General), cũng có hiệu lực đối với COM. Do đó, đặc tả SWS BSW General sẽ được coi là đặc tả bổ sung và bắt buộc cho COM.

## Constraints and Assumptions

Tài liệu này áp dụng cho phiên bản AUTOSAR 4.1.

### Limitations

Mô-đun AUTOSAR COM được dựa trên [17]. Tuy nhiên, không phải tất cả các tính năng của [17] đều được bao gồm và một số tính năng khác biệt. Xem SWS\_Com\_00013 để biết danh sách các tính năng không được bao gồm.

### Applicability to Car Domains

Không có hạn chế.

## Dependencies to Other Modules

Chương này liệt kê tất cả các tính năng từ các mô-đun khác được sử dụng bởi mô-đun AUTOSAR COM và các chức năng mà mô-đun AUTOSAR COM cung cấp cho các mô-đun khác. Để biết vị trí của mô-đun AUTOSAR COM trong ngăn xếp truyền thông, xem Hình 1.

A diagram of a computer

Description automatically generated

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

* **Rte::Rte** và **Com::Com**: Giao tiếp qua giao diện Rte\_Cbk để gửi và nhận tín hiệu.
* **ComM::ComM** và **Com::Com**: Quản lý truyền thông.
* **BswM::BswM** và **Com::Com**: Quản lý nhóm I-PDU tùy chọn.
* **EcuM::EcuM** và **Com::Com**: Khởi tạo I-PDU tùy chọn.
* **PduR::PduR** và **Com::Com**: Chuyển tiếp và callback I-PDU.
* **Det::Det** và **Com::Com**: Báo lỗi phát triển tùy chọn.
* **Dem::Dem** và **Det::Det**: Báo cáo lỗi.

### PDU Router

Mô-đun AUTOSAR COM sử dụng tập hợp các API của các mô-đun lớp trên của PDU Router. Đó là các API cho các mô-đun lớp trên sử dụng TP và các API cho các mô-đun lớp trên không sử dụng TP. Điều này là cần thiết vì mô-đun AUTOSAR COM vận chuyển các I-PDU hoặc không phân mảnh thông qua các L-PDU đơn giản hoặc phân mảnh qua TP. Dưới đây là tóm tắt các chức năng mà mô-đun AUTOSAR COM cần từ lớp dưới của Bộ định tuyến PDU:

* Quy Trình Thông Báo I-PDUs
* **Gửi giao diện cho các I-PDU gửi đi** bao gồm cả xác nhận nếu một I-PDU đã được gửi bởi bộ điều khiển truyền thông.
* **Giao diện kích hoạt** để cho phép bộ định tuyến PDU gây ra việc truyền từ mô-đun AUTOSAR COM.
* **Xử lý bộ đệm cho truyền thông TP**.

Chương 7.5 và Chương 9.1 bao gồm mô tả chi tiết về các giao diện đến Bộ định tuyến PDU. Để biết thêm thông tin, xem [10].

### Runtime Environment (RTE)

RTE sử dụng các khả năng của mô-đun AUTOSAR COM để gửi và nhận tín hiệu. Trong AUTOSAR, RTE là lớp cao hơn trên mô-đun AUTOSAR COM. Để biết thêm thông tin, xem [13].

### File Structure

#### Header File Structure

A diagram of a computer

Description automatically generated

Hình 2: Cấu trúc tệp bao gồm

[**SWS\_Com\_00220**] ⌈Tệp Com.c phải bao gồm:

* Com.h
* Com\_MemMap.h
* PduR\_Com.h
* SchM\_Com.h
* Tất cả các tệp tiêu đề callback người dùng được cấu hình theo ComUserCbkHeaderFile (ECUC\_Com\_10001) ⌋ (SRS\_BSW\_00300, SRS\_BSW\_00415)

[**SWS\_Com\_00796**] ⌈Tệp Com.h phải bao gồm:

* Com\_Cfg.h
* ComStack\_Types.h ⌋ (SRS\_BSW\_00346, SRS\_BSW\_00381, SRS\_BSW\_00412)

## Requirements Traceability

## Functional Specification

### Introduction

### General Functionality

#### OSEK-COM

OSEK COM 3.0.3 là cơ sở chức năng của mô-đun AUTOSAR COM.

[**SWS\_Com\_00010**] Mô-đun AUTOSAR COM phải triển khai tất cả các chức năng và tất cả các API của OSEK/VDX Communication phiên bản 3.0.3 [17], ngoại trừ các tính năng và API được đề cập trong SWS\_Com\_00013.

**Giải thích**: Điều này có nghĩa là AUTOSAR COM sẽ tuân thủ hầu hết các chức năng và giao diện lập trình ứng dụng (API) mà OSEK/VDX Communication phiên bản 3.0.3 đã xác định. Chỉ những tính năng và API được liệt kê cụ thể trong SWS\_Com\_00013 mới không được triển khai.

[**SWS\_Com\_00011**]: Nếu đặc tả AUTOSAR COM này định nghĩa chức năng theo cách khác so với các định nghĩa trong [17], mô-đun AUTOSAR COM phải triển khai chức năng được định nghĩa trong đặc tả AUTOSAR COM này.

**Giải thích**: Khi có sự khác biệt giữa đặc tả của AUTOSAR COM và OSEK/VDX Communication phiên bản 3.0.3, mô-đun AUTOSAR COM sẽ tuân theo các quy định trong đặc tả AUTOSAR COM. Điều này đảm bảo rằng các yêu cầu cụ thể của AUTOSAR COM được ưu tiên và thực hiện.

[**SWS\_Com\_00012**]: Mô-đun AUTOSAR COM phải bổ sung và triển khai tất cả những tính năng được định nghĩa trong đặc tả AUTOSAR COM này và không phải là một phần của [17].  
**Giải thích**: AUTOSAR COM phải thêm vào các tính năng mới mà không có trong OSEK/VDX Communication phiên bản 3.0.3, đảm bảo rằng mô-đun này không chỉ tuân theo các tiêu chuẩn cũ mà còn cải tiến và mở rộng các tính năng mới.

[**SWS\_Com\_00013**]: Mô-đun AUTOSAR COM có thể triển khai các tính năng sau của [17]. Nếu chúng được triển khai trong một mô-đun AUTOSAR COM cụ thể, cấu hình sẽ vô hiệu hóa chúng theo mặc định. Điều này cũng áp dụng cho tất cả các tính năng bổ sung khác mà một triển khai cụ thể có thể cung cấp.

**Giải thích**: Các tính năng từ OSEK/VDX Communication phiên bản 3.0.3 có thể được triển khai trong AUTOSAR COM, nhưng mặc định sẽ bị vô hiệu hóa. Nếu cần sử dụng, chúng phải được kích hoạt thông qua cấu hình. Điều này cũng đúng với các tính năng bổ sung khác có thể có trong các phiên bản triển khai cụ thể.

| **Tính năng OSEK-COM** | **Lý do** | **API OSEK COM liên quan** |
| --- | --- | --- |
| Ánh xạ một tin nhắn mạng nhận được (trong một I-PDU) thành nhiều đối tượng dữ liệu tin nhắn (cơ chế chia tách 1 : n).  **Lý do không cần thiết**: Việc này đã được RTE (Runtime Environment) thực hiện. RTE chịu trách nhiệm chia tách tin nhắn và phân phối chúng tới các đối tượng dữ liệu khác nhau. | Không cần thiết, đã được RTE thực hiện, xem [13] | Không có |
| Ánh xạ một tin nhắn nội bộ thành nhiều đối tượng dữ liệu tin nhắn (cơ chế chia tách 1: n).  **Lý do không cần thiết**: Tương tự như trên, RTE đảm nhiệm việc chia tách tin nhắn nội bộ. | Không cần thiết, đã được RTE thực hiện, xem [13] | Không có |
| Ánh xạ một tin nhắn chỉ gửi nội bộ tới cả đối tượng tin nhắn gửi ra ngoài và đối tượng tin nhắn nhận nội bộ (cơ chế chia tách 1: n).  **Lý do không cần thiết**: RTE đã đảm nhận việc này, do đó không cần thiết trong AUTOSAR COM. | Không cần thiết, đã được RTE thực hiện, xem [13] | Không có |
| Ánh xạ tin nhắn từ nhiều nguồn gửi tới cùng một đối tượng tin nhắn.  **Lý do không cần thiết**: RTE đảm bảo việc này. AUTOSAR COM không cần xử lý ánh xạ từ nhiều nguồn gửi. | Không cần thiết, được đảm bảo bởi RTE, xem [13] | SendMessage |
| Tin nhắn được xếp hàng đợi.  **Lý do không cần thiết**: Việc xếp hàng tin nhắn đã được RTE thực hiện. | Không cần thiết, đã được RTE thực hiện, xem [13] | GetMessageStatus |
| Tin nhắn kích thước bằng không.  **Lý do không cần thiết**: Có thể thiết lập giao tiếp mà không cần tin nhắn kích thước bằng không. | Có thể thiết lập giao tiếp mà không cần các chức năng này, vì chúng đã được bao phủ một phần bởi Com\_TriggerTransmit. | SendZeroMessage |

| **Tính năng OSEK-COM** | **Lý do loại trừ** | **API liên quan của OSEK COM** |
| --- | --- | --- |
| Cơ chế thông báo TASK, FLAG và EVENT | Không yêu cầu, được thực hiện bởi RTE, xem [13] | None |
| Thông điệp trùng lặp trong I-PDU | Không có trường hợp sử dụng, là khái niệm nguy hiểm | None |
| Sử dụng OIL | OSEK OIL không được sử dụng để cấu hình module AUTOSAR COM | None |
| Chế độ ứng dụng | Không cần thiết | GetComApplicationMode |
| Hành vi khởi động | Thay thế bởi:  - Com\_Init  - Com\_DeInit  - Com\_IpduGroupControl | StartCOM, StopCOM, StartCOMExtensions, InitMessage |
| Bắt đầu và dừng thông điệp định kỳ | Không có trường hợp sử dụng, được thực hiện bởi cơ chế nhóm I-PDU | StartPeriodic, StopPeriodic |
| Tính tái nhập | Không phải tất cả các lời gọi API của AUTOSAR đều tái nhập. Xem Chương 8.3. | Xem Chương 8.3 |
| Giao diện với OSEK gián tiếp NM | Không cần thiết | I\_MessageTransfer, I\_MessageTimeOut |
| Lọc phía gửi | Không có trường hợp sử dụng, các điều kiện lọc vẫn được sử dụng trong việc chọn chế độ truyền nhưng không có lọc tín hiệu | None |
| Gọi thông điệp theo thứ tự mạng và CPU | Chỉ các cuộc gọi I-PDU với giao diện AUTOSAR được định nghĩa mới được hỗ trợ bởi module AUTOSAR COM để tránh các giải pháp độc quyền. | None |
| Chức năng hook lỗi | Module AUTOSAR COM sẽ sử dụng giao diện trực tiếp đến DEM/DET thay vì sử dụng hook lỗi OSEK COM | COMErrorHook, COMError\_Name1\_Name2 macros, COMErrorGetServiceId |
| Giao diện cho các hàm callback | Chữ ký cho các hàm callback sẽ được định nghĩa rõ ràng trong đặc tả của module AUTOSAR COM. | COMCallback |
| Giao tiếp nội bộ | Không cần thiết, được đảm bảo bởi RTE, xem [13] | SendMessage, ReceiveMessage |

#### Signal Values

Các tín hiệu được gửi bởi module AUTOSAR COM và nhận bởi module AUTOSAR COM có thể có các giá trị được định nghĩa trong Bảng 2.

|  |  |
| --- | --- |
| **Signal value** | **Remark** |
| Giá trị khởi tạo | Xem Chương 7.3.1.4 để biết chi tiết. |
| Giá trị dữ liệu không hợp lệ | Xem Chương 7.3.4 để biết chi tiết. |
| <Giá trị> | Đây là trường hợp bình thường: Một giá trị hợp lệ sau giai đoạn khởi tạo, được gửi bởi module AUTOSAR COM tương ứng, nhận bởi module AUTOSAR COM. |

**Table 2: Possible signal values**

#### Endianness Conversion and Sign Extension

**[SWS\_Com\_00675]** ⌈ Module AUTOSAR COM phải hỗ trợ các kiểu dữ liệu sau:

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

Kiểu dữ liệu uint8[n] có thể được ánh xạ tới ComSignalType là hai kiểu UINT8\_N hoặc UINT8\_DYN. (SRS\_Com\_02078, SRS\_Com\_02094)

[**SWS\_Com\_00007**] ⌈Mô-đun AUTOSAR COM phải hỗ trợ chuyển đổi Endianness cho tất cả các kiểu dữ liệu số nguyên có dấu và không dấu.⌋ (SRS\_Com\_02078)

[**SWS\_Com\_00810**] ⌈Mô-đun AUTOSAR COM, các kiểu dữ liệu không phải số nguyên như uint8[n] hoặc float32 sẽ được xử lý như các kiểu dữ liệu số nguyên tương ứng hoặc để nguyên nếu ComSignalEndianness của chúng được cấu hình là OPAQUE.⌋ (SRS\_Com\_02078)

[**SWS\_Com\_00472**] ⌈Trong mô-đun AUTOSAR COM, dữ liệu opaque (dữ liệu không được giải mã hoặc can thiệp) được coi là uint8[n] và luôn được ánh xạ đến một tín hiệu có kích thước n-bytes...⌋ (SRS\_Com\_02078)

Đối với dữ liệu opaque, chuyển đổi endianness của dữ liệu này phải được cấu hình là OPAQUE (xem ECUC\_Com\_00157).

[**SWS\_Com\_00674**] ⌈Mô-đun AUTOSAR COM phải mở rộng chuyển đổi endianness được định nghĩa trong [17] Chương 2.4 tới các kiểu dữ liệu có dấu.⌋ (SRS\_Com\_02078)

Trong [17] Chương 2.4 định nghĩa chuyển đổi endianness cho các kiểu dữ liệu không dấu. Các cấu hình liên quan có thể được tìm thấy trong Chương 10. (Xem thêm ECUC\_Com\_00127 và ECUC\_Com\_00157).

[**SWS\_Com\_00829**] ⌈Trong mô-đun AUTOSAR COM, phải mở rộng chuyển đổi endianness được định nghĩa trong [17] Chương 2.4 tới các tín hiệu của nhóm tín hiệu.⌋ (SRS\_Com\_02086)

AUTOSAR COM xử lý các tín hiệu và nhóm tín hiệu một cách đồng đều liên quan đến chuyển đổi endianness.

[**SWS\_Com\_00008**] ⌈ Khi nhận dữ liệu có dấu, giá trị sẽ được mở rộng để phù hợp với kích thước của ComSignalType ⌋ (SRS\_Com\_02086)

Cần phải xem xét biểu diễn cụ thể theo nền tảng của dữ liệu có dấu. Các giá trị âm của dữ liệu có dấu sẽ được ánh xạ đúng cách.

Ví dụ: Một tín hiệu có dấu 10-Bit được nhận và sao chép bởi Com\_ReceiveSignal vào một biến số nguyên có dấu 16-Bit. Nếu (-3)decimal được nhận, tín hiệu 10-Bit nhận được sẽ có giá trị 1111111101b. Khi sao chép nó vào biến số nguyên 16-Bit, giá trị sẽ được mở rộng thành 1111111111111101b.

[**SWS\_Com\_00723**] ⌈Giá trị khởi tạo (ComSignalInitValue) của một tín hiệu cũng được mở rộng để phù hợp với kích thước của ComSignalType của nó⌋ (SRS\_Com\_02086)

**Lưu ý**: Mô-đun AUTOSAR COM sẽ không cần thực hiện mở rộng dấu ở phía người gửi vì đảm bảo rằng các giá trị được truyền sẽ luôn được mở rộng dấu. Xem thêm SWS\_Com\_00785.

[**SWS\_Com\_00579**] ⌈Mô-đun AUTOSAR COM sẽ không hỗ trợ mở rộng dấu cho các kiểu dữ liệu float32 và float64.⌋ (SRS\_Com\_02086)

[**SWS\_Com\_00221**] ⌈Mô-đun AUTOSAR COM phải thực hiện chuyển đổi endianness trước cuộc gọi I-PDU ở phía người gửi để đảm bảo dữ liệu được truyền đúng thứ tự byte. Để có cái nhìn tổng quan, xem Chương 7.15.⌋ (SRS\_Com\_02078)

[**SWS\_Com\_00352**] ⌈Trước khi thực hiện lọc và phát hiện thông báo, mô-đun COM phải thực hiện mở rộng dấu và chuyển đổi endianness ở phía người nhận.⌋ (SRS\_Com\_02078, SRS\_Com\_02086)

[**SWS\_Com\_00580**] ⌈Ngoài chuyển đổi endianness, mô-đun COM sẽ không hỗ trợ các chuyển đổi phức tạp khác cho các tín hiệu với ComSignalType FLOAT32 hoặc FLOAT64. Nghĩa là chỉ hỗ trợ chuyển đổi endianness mà không hỗ trợ các chuyển đổi phức tạp hoặc chuẩn hóa các phần, số mũ, dấu hoặc giá trị lệch.⌋ (SRS\_Com\_02078)

GIẢI THÍCH

* C**omSignalType uint8[n]**: Kiểu dữ liệu uint8[n] có thể được ánh xạ tới UINT8\_N hoặc UINT8\_DYN. Đây là các kiểu dữ liệu số nguyên không dấu với chiều dài cố định hoặc động.
* **Chuyển đổi Endianness**: Mô-đun AUTOSAR COM phải hỗ trợ chuyển đổi endianness (sắp xếp byte) cho tất cả các kiểu dữ liệu số nguyên, bao gồm cả có dấu và không dấu. Điều này đảm bảo rằng dữ liệu được sắp xếp đúng thứ tự byte khi truyền giữa các hệ thống khác nhau.
* **Xử lý Dữ liệu Không Phải Số Nguyên**: Dữ liệu không phải số nguyên như uint8[n] hoặc float32 sẽ được xử lý như các kiểu dữ liệu số nguyên tương ứng hoặc để nguyên nếu cấu hình là OPAQUE.
* **Dữ liệu Opaque**: Dữ liệu opaque được coi là uint8[n] và luôn được ánh xạ đến một tín hiệu có kích thước n-bytes. Endianness của dữ liệu này được cấu hình là OPAQUE.
* **Mở rộng Chuyển đổi Endianness**: Chuyển đổi endianness cũng áp dụng cho các kiểu dữ liệu có dấu và các tín hiệu trong nhóm tín hiệu.
* **Mở rộng Dữ liệu Nhận Được**: Khi nhận dữ liệu có dấu, giá trị sẽ được mở rộng để phù hợp với kích thước của ComSignalType. Ví dụ, tín hiệu 10-Bit có dấu sẽ được mở rộng thành 16-Bit.
* **Giá trị Khởi tạo**: Giá trị khởi tạo của một tín hiệu cũng được mở rộng để phù hợp với kích thước của ComSignalType của nó.
* **Không Hỗ trợ Mở rộng Dấu cho float32 và float64**: Mô-đun không hỗ trợ mở rộng dấu cho các kiểu dữ liệu float32 và float64.
* **Thực hiện Chuyển đổi Trước Khi Gọi I-PDU**: Chuyển đổi endianness phải được thực hiện trước cuộc gọi I-PDU ở phía người gửi.
* **Xử lý Dữ liệu Ở Phía Người Nhận**: Trước khi thực hiện lọc và phát hiện thông báo, mô-đun phải thực hiện mở rộng dấu và chuyển đổi endianness.
* **Không Hỗ trợ Các Chuyển đổi Phức Tạp**: Ngoài chuyển đổi endianness, mô-đun không hỗ trợ các chuyển đổi phức tạp cho các tín hiệu với kiểu dữ liệu FLOAT32 hoặc FLOAT64.

#### Filtering

[**SWS\_Com\_00694**] ⌈Mô-đun AUTOSAR COM phải đánh giá mỗi điều kiện lọc để kết quả là đúng hoặc sai.⌋ (SRS\_Com\_02037)

**Giải thích**: Module AUTOSAR COM cần kiểm tra các điều kiện lọc để xác định xem mỗi điều kiện đó có được thỏa mãn hay không. Kết quả của mỗi lần kiểm tra sẽ là một giá trị boolean: đúng (true) hoặc sai (false). Điều này đảm bảo rằng tín hiệu được xử lý chính xác dựa trên các điều kiện lọc đã được định nghĩa.

[**SWS\_Com\_00695**] ⌈Mô-đun AUTOSAR COM chỉ lọc các tín hiệu ở phía nhận.⌋ (SRS\_Com\_02037)

**Giải thích**: Việc lọc tín hiệu chỉ được thực hiện khi module AUTOSAR COM nhận tín hiệu. Điều này có nghĩa là các tín hiệu khi gửi đi sẽ không trải qua quá trình lọc. Việc này giúp đơn giản hóa việc xử lý tín hiệu ở phía gửi và tập trung vào việc đảm bảo dữ liệu nhận được là chính xác và hợp lệ.

[**SWS\_Com\_00602**] ⌈Mô-đun AUTOSAR COM phải sử dụng các cơ chế lọc ở phía gửi cho các Điều kiện Chế độ Truyền - Transmission Mode Conditions (TMC), nhưng không lọc tín hiệu ở phía gửi.⌋ (SRS\_Com\_02083)

**Giải thích**: Mặc dù tín hiệu không được lọc ở phía gửi, nhưng module AUTOSAR COM phải sử dụng các điều kiện chế độ truyền (TMC) để xác định xem tín hiệu có nên được gửi hay không. Các cơ chế lọc này giúp kiểm soát việc truyền tín hiệu dựa trên các điều kiện cụ thể mà không cần phải thực hiện quá trình lọc tín hiệu thực tế.

**Thông tin thêm về Lựa chọn Chế độ Truyền -** Transmission Mode Conditions (TMC). Để hiểu rõ hơn về cách các chế độ truyền được chọn và sử dụng, bạn cần tham khảo Chương 7.3.3.2 và 7.3.3.3 của tài liệu.

***Thuật toán Lọc ComFilterAlgorithms được hỗ trợ***

Module AUTOSAR COM chỉ hỗ trợ các thuật toán lọc sau đây như được định nghĩa trong [17], xem ECUC\_Com\_00146:

* **ALWAYS (Luôn luôn)**: Điều kiện luôn đúng, tín hiệu luôn được xử lý.
* **NEVER (Không bao giờ)**: Điều kiện luôn sai, tín hiệu không bao giờ được xử lý.
* **MASKED\_NEW\_EQUALS\_X (Mặt nạ mới bằng X)**: Tín hiệu mới sau khi áp dụng mặt nạ bằng giá trị X.
* **MASKED\_NEW\_DIFFERS\_X (Mặt nạ mới khác X)**: Tín hiệu mới sau khi áp dụng mặt nạ khác giá trị X.
* **MASKED\_NEW\_DIFFERS\_MASKED\_OLD (Mặt nạ mới khác mặt nạ cũ)**: Tín hiệu mới sau khi áp dụng mặt nạ khác với tín hiệu cũ sau khi áp dụng mặt nạ.
* **NEW\_IS\_WITHIN (Mới nằm trong)**: Tín hiệu mới nằm trong khoảng giá trị xác định.
* **NEW\_IS\_OUTSIDE (Mới nằm ngoài)**: Tín hiệu mới nằm ngoài khoảng giá trị xác định.
* **ONE\_EVERY\_N (Một trong mỗi N lần)**: Tín hiệu chỉ được xử lý một lần trong mỗi N lần.

Để giảm độ phức tạp, mô-đun AUTOSAR COM không hỗ trợ tất cả các bộ lọc được định nghĩa trong [17]. Các bộ lọc không được hỗ trợ là không cần thiết hoặc là trường hợp đặc biệt của các bộ lọc khác. Ví dụ, bộ lọc NEW\_IS\_DIFFERENT là một trường hợp đặc biệt của MASKED\_NEW\_DIFFERS\_MASKED\_OLD với một mặt nạ đầy đủ.

[SWS\_Com\_00325] ⌈Mô-đun AUTOSAR COM phải hỗ trợ tất cả các cơ chế lọc được liệt kê trong ECUC\_Com\_00146, ngoại trừ các trường hợp ngoại lệ được định nghĩa trong SWS\_Com\_00380 và SWS\_Com\_00439.⌋ (SRS\_Com\_02037)

**Giải thích**: Module AUTOSAR COM phải hỗ trợ các cơ chế lọc chính, ngoại trừ những trường hợp đặc biệt đã được định nghĩa. Điều này đảm bảo rằng phần lớn các bộ lọc cần thiết được hỗ trợ, nhưng loại trừ những bộ lọc không cần thiết hoặc quá phức tạp.

[SWS\_Com\_00380] ⌈Đối với các tín hiệu với ComSignalType là UINT8\_N hoặc UINT8\_DYN, mô-đun AUTOSAR COM chỉ hỗ trợ ComFilterAlgorithm được cấu hình là ALWAYS hoặc NEVER.⌋ (SRS\_Com\_02037)

**Giải thích**: Khi loại tín hiệu là UINT8\_N hoặc UINT8\_DYN, chỉ có hai thuật toán lọc được hỗ trợ: ALWAYS (luôn luôn) hoặc NEVER (không bao giờ). Điều này giúp đơn giản hóa việc xử lý các loại tín hiệu cụ thể này.

[SWS\_Com\_00439] ⌈Đối với các tín hiệu với ComSignalType được cấu hình là BOOLEAN, mô-đun AUTOSAR COM chỉ hỗ trợ ComFilterAlgorithm được cấu hình là:

* ALWAYS (Luôn luôn)
* NEVER (Không bao giờ)
* MASKED\_NEW\_EQUALS\_X (Mặt nạ mới bằng X)
* MASKED\_NEW\_DIFFERS\_X (Mặt nạ mới khác X)
* MASKED\_NEW\_DIFFERS\_MASKED\_OLD (Mặt nạ mới khác mặt nạ cũ)
* ONE\_EVERY\_N (Một trong mỗi N lần) ⌋ (SRS\_Com\_02037)

**Giải thích**: Khi tín hiệu được cấu hình là BOOLEAN, chỉ các thuật toán lọc cụ thể được hỗ trợ. Điều này đảm bảo rằng các loại tín hiệu đơn giản như BOOLEAN được xử lý một cách hiệu quả và không quá phức tạp.

[SWS\_Com\_00764] ⌈Đối với các tín hiệu và nhóm tín hiệu với ComBitSize được cấu hình là 0, mô-đun AUTOSAR COM sẽ không hỗ trợ thuật toán lọc Masked\_New\_Differs\_Masked\_Old. ⌋ (SRS\_Com\_02037)

**Giải thích**: Khi kích thước bit của tín hiệu hoặc nhóm tín hiệu được cấu hình là 0, thuật toán lọc Masked\_New\_Differs\_Masked\_Old sẽ không được hỗ trợ. Điều này hợp lý vì không có bit nào để áp dụng mặt nạ trong trường hợp này.

[SWS\_Com\_00866] ⌈Trong trường hợp điều kiện lọc được cấu hình là MASKED\_NEW\_DIFFERS\_X, MASKED\_NEW\_EQUALS\_X, NEW\_IS\_OUTSIDE hoặc NEW\_IS\_WITHIN, mô-đun AUTOSAR COM phải tính toán bộ lọc nhận bằng cách chỉ sử dụng các bit có giá trị nhỏ nhất của ComBitSize trong các tham số lọc.⌋ (SRS\_Com\_02037)

**Giải thích**: Khi sử dụng các điều kiện lọc như MASKED\_NEW\_DIFFERS\_X, MASKED\_NEW\_EQUALS\_X, NEW\_IS\_OUTSIDE hoặc NEW\_IS\_WITHIN, module AUTOSAR COM phải tính toán các điều kiện lọc dựa trên các bit có giá trị nhỏ nhất của ComBitSize. Điều này đảm bảo rằng chỉ những bit quan trọng nhất mới được xem xét trong quá trình lọc.

[SWS\_Com\_00273] ⌈Nếu mô-đun AUTOSAR COM lọc bỏ một tín hiệu ở phía nhận, tức là điều kiện lọc đánh giá là sai, mô-đun AUTOSAR COM phải loại bỏ tín hiệu đó và không xử lý nó. Xem thêm SWS\_Com\_00303.⌋ (SRS\_Com\_02037)

**Giải thích**: Nếu điều kiện lọc không thỏa mãn (đánh giá là sai), tín hiệu sẽ bị loại bỏ và không được xử lý. Điều này đảm bảo rằng chỉ các tín hiệu hợp lệ và quan trọng mới được xử lý tiếp.

[SWS\_Com\_00836] ⌈Nếu mô-đun AUTOSAR COM lọc bỏ một nhóm tín hiệu ở phía nhận, tức là điều kiện lọc đánh giá là sai, mô-đun AUTOSAR COM phải loại bỏ toàn bộ nhóm tín hiệu và không xử lý nó.⌋ (SRS\_Com\_02037)

**Giải thích**: Nếu điều kiện lọc của một nhóm tín hiệu không thỏa mãn (đánh giá là sai), module AUTOSAR COM phải loại bỏ toàn bộ nhóm tín hiệu đó và không thực hiện bất kỳ xử lý nào đối với nhóm tín hiệu này. Điều này đảm bảo rằng các tín hiệu không hợp lệ hoặc không cần thiết sẽ không được xử lý.

[SWS\_Com\_00132] ⌈Mô-đun AUTOSAR COM phải hỗ trợ các cơ chế lọc như được định nghĩa trong ECUC\_Com\_00146, kể cả đối với các kiểu dữ liệu có dấu.⌋ (SRS\_Com\_02037)

**Giải thích**: Module AUTOSAR COM phải hỗ trợ tất cả các cơ chế lọc được định nghĩa, bao gồm cả các kiểu dữ liệu có dấu (signed data types). Điều này đảm bảo rằng module có thể xử lý một cách chính xác và hiệu quả tất cả các loại tín hiệu, bao gồm cả những tín hiệu có giá trị âm.

Trong trường hợp bộ lọc được đánh giá trước khi API gửi ghi tín hiệu tương ứng, cần có cách để xác định trạng thái bộ lọc của tín hiệu này. Một số bộ lọc yêu cầu giá trị mới để đánh giá bộ lọc. Tuy nhiên, điều này chỉ khả dụng sau khi tín hiệu đã được cập nhật bằng cách sử dụng API gửi. Do đó, cần phải xác định giá trị được sử dụng bởi bộ lọc cho giá trị mới trong khoảng thời gian trước khi gửi lần đầu.

**Giải thích**: Để đánh giá bộ lọc một cách chính xác trước khi gửi tín hiệu, cần phải biết giá trị mới của tín hiệu. Tuy nhiên, giá trị mới chỉ có sau khi tín hiệu đã được cập nhật. Do đó, cần phải có cơ chế để xác định giá trị ban đầu của tín hiệu được sử dụng trong quá trình lọc trước khi tín hiệu được gửi lần đầu tiên.

[SWS\_Com\_00603] ⌈Mô-đun AUTOSAR COM phải đặt giá trị cũ của các cơ chế lọc cho mỗi tín hiệu thành ComSignalInitValue (ECUC\_Com\_00170) trong quá trình khởi động. Xem thêm [17].⌋ (SRS\_Com\_02037)

**Giải thích**: Trong quá trình khởi động, giá trị cũ của mỗi tín hiệu trong các cơ chế lọc phải được thiết lập thành ComSignalInitValue. Điều này đảm bảo rằng các tín hiệu có một giá trị ban đầu được định nghĩa rõ ràng để sử dụng trong quá trình lọc.

[SWS\_Com\_00604] ⌈Cho đến khi ứng dụng cập nhật giá trị mới của các cơ chế lọc cho một tín hiệu, mô-đun AUTOSAR COM phải sử dụng ComSignalInitValue làm giá trị mới cho tín hiệu đó.⌋ (SRS\_Com\_02037)

**Giải thích**: Trước khi giá trị mới của tín hiệu được ứng dụng cập nhật, module AUTOSAR COM phải sử dụng ComSignalInitValue làm giá trị mới cho tín hiệu này. Điều này đảm bảo rằng tín hiệu có một giá trị ban đầu hợp lệ cho quá trình lọc.

[SWS\_Com\_00302] ⌈Nếu mô-đun AUTOSAR COM đánh giá bộ lọc cho một tín hiệu là đúng (giá trị không bị lọc bỏ), thì mô-đun AUTOSAR COM phải đặt giá trị của tín hiệu đó vào giá trị cũ (như được định nghĩa trong [17]).⌋ (SRS\_Com\_02037)

**Giải thích**: Nếu điều kiện lọc thỏa mãn (đánh giá là đúng), giá trị của tín hiệu phải được cập nhật thành giá trị cũ. Điều này giúp giữ lại giá trị của tín hiệu cho các lần lọc tiếp theo.

[SWS\_Com\_00303] ⌈Khi một giá trị bị lọc, nếu bộ lọc không cho phép giá trị đó qua (tức là bộ lọc đánh giá là sai), thì mô-đun AUTOSAR COM không được đặt giá trị đó vào giá trị cũ (như được định nghĩa trong [17]).⌋ (SRS\_Com\_02037)

**Giải thích**: Nếu giá trị bị lọc bỏ (điều kiện lọc đánh giá là sai), giá trị đó không được cập nhật thành giá trị cũ. Điều này ngăn chặn việc lưu trữ các giá trị không hợp lệ hoặc không cần thiết.

[SWS\_Com\_00231] ⌈Trong trường hợp ComFilterAlgorithm được cấu hình là ONE\_EVERY\_N, mô-đun AUTOSAR COM phải:

* đặt OCCURRENCE về không khi OCCURRENCE == PERIOD
* đặt FILTER thành true khi OCCURRENCE == OFFSET
* tăng OCCURRENCE sau khi xử lý bộ lọc ⌋ (SRS\_Com\_02037)

**Giải thích**:

* **ONE\_EVERY\_N** là một thuật toán lọc cho phép tín hiệu vượt qua bộ lọc mỗi lần gọi thứ N.
* **OCCURRENCE**: Số lần bộ lọc đã được gọi.
* **FILTER**: Trạng thái của bộ lọc (true/false).
* **OFFSET**: Số lần gọi bộ lọc trước khi tín hiệu đầu tiên được cho qua.
* **PERIOD**: Số lần gọi bộ lọc trước khi tín hiệu tiếp theo được cho qua.

Khi **OCCURRENCE** đạt đến **PERIOD**, nó được đặt lại về 0. Nếu **OCCURRENCE** bằng **OFFSET**, bộ lọc được đặt thành true, cho phép tín hiệu qua. Sau mỗi lần xử lý bộ lọc, **OCCURRENCE** được tăng lên một.

**Định nghĩa**: ONE\_EVERY\_N cho phép tín hiệu qua mỗi lần gọi thứ N của bộ lọc. Nếu **OFFSET** là 0, tín hiệu đầu tiên sẽ được cho qua ngay lần đầu gọi bộ lọc. Nếu **OFFSET** lớn hơn 0, phải qua một số lần gọi bộ lọc trước khi tín hiệu được cho qua.

[SWS\_Com\_00793] ⌈Đối với một tín hiệu với bộ lọc tiếp nhận được cấu hình là MASKED\_NEW\_DIFFERS\_MASKED\_OLD, mô-đun AUTOSAR COM phải xử lý giá trị đầu tiên nhận được cho tín hiệu này sau khi xảy ra sự cố theo dõi thời hạn tiếp nhận giống như giá trị đó đã vượt qua tiêu chí bộ lọc.⌋ (SRS\_Com\_02037)

**Giải thích**:

* **MASKED\_NEW\_DIFFERS\_MASKED\_OLD** là thuật toán lọc so sánh tín hiệu mới sau khi áp dụng mặt nạ với tín hiệu cũ sau khi áp dụng mặt nạ. Nếu khác nhau, tín hiệu được cho qua.
* Khi thời hạn tiếp nhận kết thúc, tín hiệu đầu tiên nhận được sau thời hạn đó sẽ được coi là đã vượt qua bộ lọc, tức là được cho qua, bất kể giá trị thực của nó.

Các khái niệm và thuật ngữ liên quan

1. **OCCURRENCE**: Số lần bộ lọc đã được gọi.
2. **FILTER**: Trạng thái của bộ lọc.
3. **OFFSET**: Số lần gọi bộ lọc trước khi tín hiệu đầu tiên được cho qua.
4. **PERIOD**: Số lần gọi bộ lọc trước khi tín hiệu tiếp theo được cho qua.
5. **MASKED\_NEW\_DIFFERS\_MASKED\_OLD**: Thuật toán lọc so sánh tín hiệu mới và cũ sau khi áp dụng mặt nạ.

#### Signal Gateway

Gateway Tín Hiệu tích hợp trong mô-đun AUTOSAR COM

**Mô tả**: Module AUTOSAR COM cung cấp một Gateway Tín Hiệu tích hợp để chuyển tiếp các tín hiệu và nhóm tín hiệu theo kiểu 1

. Điều này có nghĩa là một tín hiệu hoặc nhóm tín hiệu có thể được gửi đến nhiều đích khác nhau.

**Cấu hình định tuyến tĩnh**: Các mối quan hệ định tuyến được cấu hình tĩnh qua container cấu hình ComGwMapping (xem ECUC\_Com\_00544). Điều này có nghĩa là các mối quan hệ định tuyến được thiết lập trước và không thay đổi trong quá trình hoạt động.

**Hoạt động của Gateway Tín Hiệu**:

* Gateway Tín Hiệu hoạt động như một bộ nhận cho tất cả các tín hiệu hoặc nhóm tín hiệu được cấu hình làm nguồn cho các tín hiệu chuyển tiếp.
* Sau khi nhận tín hiệu hoặc nhóm tín hiệu để định tuyến, nó ngay lập tức hoạt động như một bộ gửi cho các tín hiệu hoặc nhóm tín hiệu này tương ứng.
* Quá trình xử lý tín hiệu không khác biệt khi Gateway Tín Hiệu tích hợp chuyển tiếp một tín hiệu/nhóm tín hiệu hoặc khi một Thành Phần Phần Mềm gửi nó.

**Tối ưu hóa**:

* Do các mối quan hệ gateway được cấu hình tĩnh, một Gateway Tín Hiệu tối ưu có thể bỏ qua một số giai đoạn xử lý cho các cấu hình cụ thể (ví dụ: chuyển đổi endianness không cần thiết trong một số trường hợp).
* Các tối ưu hóa chỉ được áp dụng để tăng hiệu suất của Gateway Tín Hiệu và không nên dẫn đến bất kỳ hành vi logic khác biệt nào hoặc các biểu diễn khác nhau trên bus mục tiêu.

**Xử lý tín hiệu và nhóm tín hiệu**:

* Một tín hiệu hoặc nhóm tín hiệu nhận được có thể được nhận trên ECU cục bộ và đồng thời là nguồn cho một hành động gateway.
* Các bộ đệm shadow cho các nhóm tín hiệu được nhận bởi RTE (Runtime Environment) được sử dụng để đảm bảo tính nhất quán.
* Gateway Tín Hiệu tích hợp phải đảm bảo xử lý nhóm tín hiệu một cách nhất quán nhưng không có cách nào định trước mà điều này phải được thực hiện.

[SWS\_Com\_00370] ⌈Gateway Tín Hiệu của mô-đun AUTOSAR COM phải thu nhỏ về không kích thước nếu không cần chức năng định tuyến tín hiệu.⌋ (SRS\_PduR\_06064)

**Giải thích**: Nếu chức năng định tuyến tín hiệu không cần thiết, Gateway Tín Hiệu trong mô-đun AUTOSAR COM phải được thu nhỏ về kích thước không. Điều này giúp giảm thiểu tài nguyên hệ thống khi chức năng định tuyến không được sử dụng.

### Normal Operation

#### Start-Up Behavior

Chương này mô tả các hành động sẽ được thực hiện trong quá trình khởi tạo Com\_Init.

[SWS\_Com\_00217] ⌈Mô-đun AUTOSAR COM phải khởi tạo mỗi I-PDU trong quá trình thực thi Com\_Init (SWS\_Com\_00432), đầu tiên theo từng byte với giá trị ComTxIPduUnusedAreasDefault và sau đó theo từng bit dựa trên các giá trị khởi tạo (ComSignalInitValue) của các tín hiệu chứa và các bit cập nhật (xem SWS\_Com\_00117).⌋ (SRS\_BSW\_00101)

**Giải thích**: Trong quá trình khởi tạo Com\_Init, mỗi I-PDU phải được khởi tạo:

1. **Theo từng byte**: Sử dụng giá trị ComTxIPduUnusedAreasDefault để khởi tạo các khu vực không sử dụng.
2. **Theo từng bit**: Sử dụng các giá trị khởi tạo ComSignalInitValue của các tín hiệu chứa và các bit cập nhật.

##### Preconditions

**Giải thích**: Mã khởi tạo C, hay mã khởi động, khởi tạo các biến toàn cục và tĩnh với các giá trị khởi tạo và phải được thực thi trước bất kỳ cuộc gọi nào của dịch vụ mô-đun AUTOSAR COM.

##### Initialization

[SWS\_Com\_00128] ⌈Hàm khởi tạo của mô-đun AUTOSAR COM Com\_Init (SWS\_Com\_00432) phải khởi tạo tất cả dữ liệu nội bộ chưa được khởi tạo bởi mã khởi động, ví dụ như các cấu trúc C.⌋ (SRS\_BSW\_00101)

**Giải thích**: Hàm Com\_Init phải khởi tạo tất cả dữ liệu nội bộ chưa được mã khởi động khởi tạo, như các cấu trúc dữ liệu C.

[SWS\_Com\_00328] ⌈Hàm khởi tạo của mô-đun AUTOSAR COM Com\_Init (SWS\_Com\_00432) không được kích hoạt giao tiếp liên ECU.⌋ (SRS\_BSW\_00101)

**Giải thích**: Hàm Com\_Init không được phép kích hoạt giao tiếp giữa các ECU trong quá trình khởi tạo.

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

[SWS\_Com\_00015] ⌈Mô-đun AUTOSAR COM phải điền các khu vực không sử dụng trong một I-PDU với một giá trị được xác định bởi tham số cấu hình ComTxIPduUnusedAreasDefault (ECUC\_Com\_00017), ví dụ: 0xFF.⌋ (SRS\_BSW\_00101, SRS\_Com\_02042)

**Giải thích**: Các khu vực không sử dụng trong một I-PDU phải được điền với giá trị ComTxIPduUnusedAreasDefault, ví dụ: 0xFF, để đảm bảo rằng các khu vực này không chứa dữ liệu không hợp lệ.

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

[SWS\_Com\_00098] ⌈Mô-đun AUTOSAR COM phải khởi tạo mỗi tín hiệu có kích thước n-bit trên phía gửi và nhận với n-bit thấp hơn của tham số cấu hình ComSignalInitValue (ECUC\_Com\_00170).⌋ (SRS\_BSW\_00101)

**Giải thích**: Mỗi tín hiệu có kích thước n-bit phải được khởi tạo với n-bit thấp hơn của giá trị ComSignalInitValue. Điều này áp dụng cho cả phía gửi và nhận.

**Thông tin thêm**:

* Giá trị ComSignalInitValue của một tín hiệu có thể giống với giá trị ComSignalDataInvalidValue nhưng có thể khác nhau cho từng tín hiệu.
* Giá trị khởi tạo ComSignalInitValues cũng được sử dụng để khởi tạo tín hiệu trong I-PDU liên quan (xem SWS\_Com\_00217).

[SWS\_Com\_00117] ⌈Mô-đun AUTOSAR COM phải xóa tất cả các bit cập nhật trong quá trình khởi tạo. Xem thêm SWS\_Com\_00059.⌋ (SRS\_BSW\_00101, SRS\_Com\_02030, SRS\_Com\_02058)

**Giải thích**: Tất cả các bit cập nhật phải được xóa trong quá trình khởi tạo để đảm bảo rằng không có tín hiệu nào được đánh dấu là cập nhật trước khi hệ thống thực sự hoạt động.

##### Initialization of I-PDU Groups

[SWS\_Com\_00444] ⌈Mặc định, tất cả các nhóm I-PDU phải ở trạng thái dừng và chúng không được khởi động tự động bằng một cuộc gọi đến Com\_Init (SWS\_Com\_00432).⌋ (SRS\_Com\_00218)

**Giải thích**: Mặc định, tất cả các nhóm I-PDU (Interface-PDU) sẽ ở trạng thái dừng khi hệ thống khởi tạo. Chúng không được tự động khởi động khi gọi hàm Com\_Init. Điều này có nghĩa là các nhóm I-PDU chỉ được khởi động khi có yêu cầu cụ thể, giúp tránh việc truyền dữ liệu không cần thiết ngay khi hệ thống khởi động.

#### De-Initialization

**Giải thích**: Mô-đun AUTOSAR COM cung cấp hàm API Com\_DeInit (SWS\_Com\_00130) để hủy khởi tạo lớp COM. Khi lớp này bị hủy khởi tạo, việc giao tiếp qua mô-đun AUTOSAR COM không thể thực hiện và tất cả các nhóm I-PDU đã khởi động sẽ bị dừng lại. Điều này giúp đảm bảo rằng không có hoạt động giao tiếp không mong muốn xảy ra sau khi hủy khởi tạo.

#### Communication Modes

**Giải thích**: Chương này mô tả các luồng tín hiệu và chế độ truyền khác nhau mà mô-đun AUTOSAR COM hỗ trợ, cùng với cơ chế chuyển đổi giữa các chế độ truyền. Nó bao gồm các định nghĩa về sao chép tín hiệu và các trường hợp sử dụng giao tiếp mẫu mà mô-đun có thể xử lý.

Các trường hợp sử dụng giao tiếp mẫu mà mô-đun AUTOSAR COM có thể xử lý được mô tả trong Chương 12.

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

###### Signals

[SWS\_Com\_00330] ⌈Tại bất kỳ yêu cầu gửi nào của một tín hiệu với ComTransferProperty TRIGGERED gán cho một I-PDU với ComTxModeMode DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải ngay lập tức (trong chức năng chính tiếp theo chậm nhất) khởi tạo ComTxModeNumberOfRepetitions cộng thêm một lần truyền của I-PDU được gán.⌋ (SRS\_Com\_02083)

**Giải thích**: Khi có yêu cầu gửi một tín hiệu với thuộc tính TRIGGERED, nếu I-PDU liên quan có chế độ truyền là DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải khởi tạo việc truyền I-PDU ngay lập tức, ít nhất trong lần chức năng chính tiếp theo. Số lần truyền được xác định bởi ComTxModeNumberOfRepetitions cộng thêm một lần.

[SWS\_Com\_00767] ⌈Tại bất kỳ yêu cầu gửi nào của một tín hiệu với ComTransferProperty TRIGGERED\_WITHOUT\_REPETITION gán cho một I-PDU với ComTxModeMode DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải ngay lập tức (trong chức năng chính tiếp theo chậm nhất) khởi tạo một lần truyền của I-PDU được gán.⌋ (SRS\_Com\_02083)

**Giải thích**: Tương tự, nếu tín hiệu có thuộc tính TRIGGERED\_WITHOUT\_REPETITION và I-PDU có chế độ truyền là DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải khởi tạo việc truyền I-PDU ngay lập tức, ít nhất trong lần chức năng chính tiếp theo, nhưng chỉ một lần.

[SWS\_Com\_00734] ⌈Tại bất kỳ yêu cầu gửi nào của một tín hiệu với ComTransferProperty TRIGGERED\_ON\_CHANGE gán cho một I-PDU với ComTxModeMode DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải ngay lập tức (trong chức năng chính tiếp theo chậm nhất) khởi tạo ComTxModeNumberOfRepetitions cộng thêm một lần truyền của I-PDU được gán, nếu tín hiệu mới gửi khác với tín hiệu lưu trữ cục bộ (lần gửi cuối hoặc giá trị khởi tạo) về độ dài hoặc giá trị.⌋ (SRS\_Com\_02083)

**Giải thích**: Nếu thuộc tính của tín hiệu là TRIGGERED\_ON\_CHANGE và I-PDU có chế độ truyền là DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải khởi tạo việc truyền I-PDU ngay lập tức nếu tín hiệu mới khác với tín hiệu lưu trữ cục bộ về độ dài hoặc giá trị. Số lần truyền được xác định bởi ComTxModeNumberOfRepetitions cộng thêm một lần.

[SWS\_Com\_00768] ⌈Tại bất kỳ yêu cầu gửi nào của một tín hiệu với ComTransferProperty TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION gán cho một I-PDU với ComTxModeMode DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải ngay lập tức (trong chức năng chính tiếp theo chậm nhất) khởi tạo một lần truyền của I-PDU được gán, nếu tín hiệu mới gửi khác với tín hiệu lưu trữ cục bộ (lần gửi cuối hoặc giá trị khởi tạo) về độ dài hoặc giá trị.⌋ (SRS\_Com\_02083)

**Giải thích**: Tương tự, nếu thuộc tính của tín hiệu là TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION và I-PDU có chế độ truyền là DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải khởi tạo việc truyền I-PDU ngay lập tức nếu tín hiệu mới khác với tín hiệu lưu trữ cục bộ về độ dài hoặc giá trị, nhưng chỉ một lần.

[SWS\_Com\_00762] ⌈Mô-đun AUTOSAR COM sẽ không hỗ trợ các thuộc tính truyền TRIGGERED\_ON\_CHANGE và TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION cho các tín hiệu và nhóm tín hiệu với ComBitSize được cấu hình là 0.⌋ (SRS\_Com\_02037)

**Giải thích**: Các thuộc tính truyền TRIGGERED\_ON\_CHANGE và TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION sẽ không được hỗ trợ cho các tín hiệu và nhóm tín hiệu có kích thước bit (ComBitSize) bằng 0.

[SWS\_Com\_00135] ⌈Mô-đun AUTOSAR COM sẽ không khởi tạo truyền cho các I-PDU có ComTxModeMode (ECUC\_Com\_00137) NONE.⌋ (SRS\_Com\_02083)

**Giải thích**: Nếu chế độ truyền của I-PDU là NONE, mô-đun AUTOSAR COM sẽ không khởi tạo việc truyền cho I-PDU đó. Tuy nhiên, I-PDU này có thể được yêu cầu truyền qua hàm Com\_TriggerTransmit.

###### Signal Groups

Trong AUTOSAR COM, cả nhóm tín hiệu và tín hiệu nhóm có thể có thuộc tính truyền (ComTransferProperty), kết hợp với chế độ truyền (ComTxModeMode), để xác định xem I-PDU có được gửi ra khi cập nhật một nhóm tín hiệu hoặc tín hiệu nhóm hay không.

[SWS\_Com\_00741] ⌈Tại bất kỳ yêu cầu gửi nào của một nhóm tín hiệu với thuộc tính truyền ComTransferProperty TRIGGERED gán cho một I-PDU với ComTxModeMode DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải ngay lập tức (trong chức năng chính tiếp theo chậm nhất) khởi tạo ComTxModeNumberOfRepetitions cộng thêm một lần truyền của I-PDU được gán.⌋ (SRS\_Com\_02083)

**Giải thích**: Nếu nhóm tín hiệu có thuộc tính truyền là TRIGGERED và I-PDU có chế độ truyền là DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải khởi tạo việc truyền I-PDU ngay lập tức, ít nhất trong lần chức năng chính tiếp theo. Số lần truyền được xác định bởi ComTxModeNumberOfRepetitions cộng thêm một lần.

[SWS\_Com\_00769] ⌈Tại bất kỳ yêu cầu gửi nào của một nhóm tín hiệu với thuộc tính truyền ComTransferProperty TRIGGERED\_WITHOUT\_REPETITION gán cho một I-PDU với ComTxModeMode DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải khởi tạo một lần truyền của I-PDU được gán trong chức năng chính tiếp theo chậm nhất.⌋ (SRS\_Com\_02083)

**Giải thích**: Tương tự, nếu nhóm tín hiệu có thuộc tính truyền là TRIGGERED\_WITHOUT\_REPETITION và I-PDU có chế độ truyền là DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải khởi tạo một lần truyền I-PDU ngay lập tức, ít nhất trong lần chức năng chính tiếp theo.

[SWS\_Com\_00742] ⌈Đối với các nhóm tín hiệu với thuộc tính truyền ComTransferProperty TRIGGERED\_ON\_CHANGE không chứa bất kỳ tín hiệu nào có cấu hình thuộc tính truyền riêng: Tại bất kỳ yêu cầu gửi nào của nhóm tín hiệu như vậy gán cho một I-PDU với ComTxModeMode DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải ngay lập tức (trong chức năng chính tiếp theo chậm nhất) khởi tạo ComTxModeNumberOfRepetitions cộng thêm một lần truyền của I-PDU được gán, nếu ít nhất một tín hiệu nhóm mới gửi khác với tín hiệu lưu trữ cục bộ (lần gửi cuối hoặc giá trị khởi tạo) về độ dài hoặc giá trị.⌋ (SRS\_Com\_02083)

**Giải thích**: Nếu nhóm tín hiệu có thuộc tính truyền TRIGGERED\_ON\_CHANGE và không chứa bất kỳ tín hiệu nào có thuộc tính truyền riêng, và I-PDU có chế độ truyền là DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải khởi tạo việc truyền I-PDU ngay lập tức nếu ít nhất một tín hiệu nhóm mới khác với tín hiệu lưu trữ cục bộ về độ dài hoặc giá trị. Số lần truyền được xác định bởi ComTxModeNumberOfRepetitions cộng thêm một lần.

[SWS\_Com\_00743] ⌈Đối với các nhóm tín hiệu với thuộc tính truyền ComTransferProperty TRIGGERED\_ON\_CHANGE chứa bất kỳ tín hiệu nào có cấu hình thuộc tính truyền riêng: Tại bất kỳ yêu cầu gửi nào của nhóm tín hiệu như vậy gán cho một I-PDU với ComTxModeMode DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải ngay lập tức (trong chức năng chính tiếp theo chậm nhất) khởi tạo ComTxModeNumberOfRepetitions cộng thêm một lần truyền của I-PDU được gán, nếu ít nhất một tín hiệu nhóm mới gửi có cấu hình thuộc tính truyền ComTransferProperty TRIGGERED\_ON\_CHANGE khác với tín hiệu lưu trữ cục bộ (lần gửi cuối hoặc giá trị khởi tạo) về độ dài hoặc giá trị.⌋ (SRS\_Com\_02083)

**Giải thích**: Nếu nhóm tín hiệu có thuộc tính truyền TRIGGERED\_ON\_CHANGE và chứa ít nhất một tín hiệu có thuộc tính truyền riêng, và I-PDU có chế độ truyền là DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải khởi tạo việc truyền I-PDU ngay lập tức nếu ít nhất một tín hiệu nhóm mới có thuộc tính truyền TRIGGERED\_ON\_CHANGE khác với tín hiệu lưu trữ cục bộ về độ dài hoặc giá trị. Số lần truyền được xác định bởi ComTxModeNumberOfRepetitions cộng thêm một lần.

[SWS\_Com\_00770] ⌈Tại bất kỳ yêu cầu gửi nào của một nhóm tín hiệu với thuộc tính truyền ComTransferProperty TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION gán cho một I-PDU với ComTxModeMode DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải ngay lập tức (trong chức năng chính tiếp theo chậm nhất) khởi tạo một lần truyền của I-PDU được gán, nếu ít nhất một tín hiệu nhóm mới gửi khác với tín hiệu lưu trữ cục bộ (lần gửi cuối hoặc giá trị khởi tạo) về độ dài hoặc giá trị.⌋ (SRS\_Com\_02083)

**Giải thích**: Tương tự, nếu nhóm tín hiệu có thuộc tính truyền TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION và I-PDU có chế độ truyền là DIRECT hoặc MIXED, mô-đun AUTOSAR COM phải khởi tạo việc truyền I-PDU ngay lập tức nếu ít nhất một tín hiệu nhóm mới khác với tín hiệu lưu trữ cục bộ về độ dài hoặc giá trị, nhưng chỉ một lần.

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

(Tổng quan về chế độ truyền của I-PDU trong AUTOSAR COM)

I-PDUs (Interface Protocol Data Units) mang các tín hiệu. Vì một I-PDU có thể chứa nhiều tín hiệu, một phương pháp được định nghĩa để xác định chế độ truyền của I-PDU từ trạng thái của các tín hiệu chứa trong I-PDU đó.

Cấu hình chế độ truyền

* **Hai chế độ truyền**: Mô-đun AUTOSAR COM cho phép cấu hình hai chế độ truyền tĩnh cho mỗi I-PDU (SWS\_Com\_00032).
* **Chọn chế độ truyền**: Chế độ truyền của I-PDU tại một thời điểm cụ thể được chọn chỉ sử dụng các giá trị của các tín hiệu được ánh xạ tới I-PDU này.
* **Cấu hình tĩnh**: Các tín hiệu và điều kiện dùng để chọn chế độ truyền được cấu hình tĩnh (SWS\_Com\_00676).
* **Tín hiệu nhóm**: Đối với việc chọn chế độ truyền, mô-đun AUTOSAR COM xử lý tín hiệu nhóm như các tín hiệu bình thường.

[SWS\_Com\_00676] ⌈Để đánh giá một TMS (Transmission Mode Selector), mô-đun AUTOSAR COM phải xem xét tất cả các tín hiệu (bao gồm cả tín hiệu nhóm) có cấu hình chứa container ComFilter. Xem các container của ComSignal và ComGroupSignal.⌋ (SRS\_Com\_02041)

**Giải thích**: Khi đánh giá một TMS (Transmission Mode Selector), mô-đun AUTOSAR COM phải xem xét tất cả các tín hiệu (bao gồm cả tín hiệu nhóm) có cấu hình chứa container ComFilter.

[SWS\_Com\_00677] ⌈Nếu không có tín hiệu nào trong một I-PDU đóng góp vào việc tính toán TMS, mô-đun AUTOSAR COM sẽ đánh giá TMS của I-PDU này là true.⌋ (SRS\_Com\_02084, SRS\_Com\_02041)

**Giải thích**: Nếu không có tín hiệu nào trong một I-PDU đóng góp vào việc tính toán TMS, mô-đun AUTOSAR COM sẽ đánh giá TMS của I-PDU này là true.

[SWS\_Com\_00678] ⌈Nếu mô-đun AUTOSAR COM đánh giá TMC (Transmission Mode Condition) của một tín hiệu đóng góp là true, thì TMS của I-PDU cũng sẽ được đánh giá là true.⌋ (SRS\_Com\_02084, SRS\_Com\_02041)

**Giải thích**: Nếu mô-đun AUTOSAR COM đánh giá TMC (Transmission Mode Condition) của một tín hiệu đóng góp là true, thì TMS của I-PDU cũng sẽ được đánh giá là true.

[SWS\_Com\_00679] ⌈Nếu không có TMC của bất kỳ tín hiệu đóng góp nào được đánh giá là true, thì TMS của I-PDU sẽ được đánh giá là false.⌋ (SRS\_Com\_02084, SRS\_Com\_02041)

**Giải thích**: Nếu không có TMC của bất kỳ tín hiệu đóng góp nào được đánh giá là true, thì TMS của I-PDU sẽ được đánh giá là false.

[SWS\_Com\_00605] ⌈Mô-đun AUTOSAR COM phải định nghĩa một Transmission Mode Selector (TMS) cho mỗi I-PDU.⌋ (SRS\_Com\_02084)

**Giải thích**: Mô-đun AUTOSAR COM phải định nghĩa một Transmission Mode Selector (TMS) cho mỗi I-PDU.

[SWS\_Com\_00245] ⌈Mô-đun AUTOSAR COM phải tính lại TMS cho một I-PDU sau khi một tín hiệu chứa trong I-PDU đó được cập nhật bởi một cuộc gọi tới Com\_SendSignal hoặc Com\_SendSignalGroup.⌋ (SRS\_Com\_02084)

**Giải thích**: Mô-đun AUTOSAR COM phải tính lại TMS cho một I-PDU sau khi một tín hiệu chứa trong I-PDU đó được cập nhật bởi một cuộc gọi tới Com\_SendSignal hoặc Com\_SendSignalGroup.

[SWS\_Com\_00763] ⌈Đối với việc tính toán chế độ truyền với điều kiện cấu hình MASKED\_NEW\_DIFFERS\_MASKED\_OLD, mô-đun AUTOSAR COM chỉ sử dụng các bit ít quan trọng nhất của ComBitSize.⌋ (SRS\_Com\_02084)

**Giải thích**: Đối với việc tính toán chế độ truyền với điều kiện cấu hình MASKED\_NEW\_DIFFERS\_MASKED\_OLD, mô-đun AUTOSAR COM chỉ sử dụng các bit ít quan trọng nhất của ComBitSize.

[SWS\_Com\_00813] ⌈Đối với việc tính toán chế độ truyền với các điều kiện cấu hình MASKED\_NEW\_DIFFERS\_X, MASKED\_NEW\_EQUALS\_X, NEW\_IS\_OUTSIDE hoặc NEW\_IS\_WITHIN, mô-đun AUTOSAR COM phải sử dụng tất cả các bit của ComSignalType.⌋ (SRS\_Com\_02084)

**Giải thích**: Đối với việc tính toán chế độ truyền với các điều kiện cấu hình MASKED\_NEW\_DIFFERS\_X, MASKED\_NEW\_EQUALS\_X, NEW\_IS\_OUTSIDE hoặc NEW\_IS\_WITHIN, mô-đun AUTOSAR COM phải sử dụng tất cả các bit của ComSignalType.

[SWS\_Com\_00032] ⌈Nếu TMS của một I-PDU được đánh giá là true, mô-đun AUTOSAR COM phải sử dụng ComTxModeMode được định nghĩa trong container cấu hình ComTxModeTrue cho I-PDU đó.⌋ (SRS\_Com\_02082, SRS\_Com\_02084)

**Giải thích**: Nếu TMS của một I-PDU được đánh giá là true, mô-đun AUTOSAR COM phải sử dụng ComTxModeMode được định nghĩa trong container cấu hình ComTxModeTrue cho I-PDU đó.

[SWS\_Com\_00799] ⌈Nếu TMS của một I-PDU được đánh giá là false, mô-đun AUTOSAR COM phải sử dụng ComTxModeMode được định nghĩa trong container cấu hình ComTxModeFalse cho I-PDU đó.⌋ (SRS\_Com\_02082, SRS\_Com\_02084)

**Giải thích**: Nếu TMS của một I-PDU được đánh giá là false, mô-đun AUTOSAR COM phải sử dụng ComTxModeMode được định nghĩa trong container cấu hình ComTxModeFalse cho I-PDU đó.

[SWS\_Com\_00238] ⌈Trong mỗi trạng thái TMS, các quy tắc kết hợp thuộc tính truyền của tín hiệu và chế độ truyền của I-PDUs phải được áp dụng như được định nghĩa trong [17] Mục 2.3.⌋ (SRS\_Com\_02082)

**Giải thích**: Trong mỗi trạng thái TMS, các quy tắc kết hợp thuộc tính truyền của tín hiệu và chế độ truyền của I-PDUs phải được áp dụng như được định nghĩa trong [17] Mục 2.3.

[SWS\_Com\_00239] ⌈Khi trạng thái TMS của một I-PDU thay đổi, mô-đun AUTOSAR COM phải sử dụng chế độ truyền mới hợp lệ ngay lập tức (trong chức năng chính tiếp theo chậm nhất). Điều này có nghĩa là, trước tiên mô-đun AUTOSAR COM phải thực hiện thay đổi chế độ và sau đó thực hiện bất kỳ cuộc gọi nào tới PduR\_ComTransmit do thay đổi chế độ gây ra.⌋ (SRS\_Com\_02082)

**Giải thích**: Khi trạng thái TMS của một I-PDU thay đổi, mô-đun AUTOSAR COM phải sử dụng chế độ truyền mới hợp lệ ngay lập tức (trong chức năng chính tiếp theo chậm nhất). Điều này có nghĩa là, trước tiên mô-đun AUTOSAR COM phải thực hiện thay đổi chế độ và sau đó thực hiện bất kỳ cuộc gọi nào tới PduR\_ComTransmit do thay đổi chế độ gây ra.

[SWS\_Com\_00244] ⌈Nếu thay đổi TMS gây ra thay đổi chế độ truyền cho một I-PDU, mô-đun AUTOSAR COM phải khởi động lại bộ đếm thời gian cho chu kỳ truyền của chế độ PERIODIC và MIXED.⌋ (SRS\_Com\_02082)

**Giải thích**: Nếu thay đổi TMS gây ra thay đổi chế độ truyền cho một I-PDU, mô-đun AUTOSAR COM phải khởi động lại bộ đếm thời gian cho chu kỳ truyền của chế độ PERIODIC và MIXED.

[SWS\_Com\_00495] ⌈Khi một cuộc gọi tới Com\_SendSignal hoặc Com\_SendSignalGroup dẫn đến thay đổi chế độ truyền của một I-PDU đã khởi động sang chế độ PERIODIC hoặc MIXED, mô-đun AUTOSAR COM phải bắt đầu chu kỳ truyền mới với một cuộc gọi tới PduR\_ComTransmit trong chức năng chính tiếp theo chậm nhất. Việc truyền phải được khởi tạo bất kể thuộc tính truyền của tín hiệu hoặc nhóm tín hiệu đã gây ra thay đổi chế độ truyền. Thời gian trễ tối thiểu và ComTxModeTimeOffset vẫn phải được tuân thủ.⌋ (SRS\_Com\_02082)

**Giải thích**: Khi một cuộc gọi tới Com\_SendSignal hoặc Com\_SendSignalGroup dẫn đến thay đổi chế độ truyền của một I-PDU đã khởi động sang chế độ PERIODIC hoặc MIXED, mô-đun AUTOSAR COM phải bắt đầu chu kỳ truyền mới với một cuộc gọi tới PduR\_ComTransmit trong chức năng chính tiếp theo chậm nhất. Việc truyền phải được khởi tạo bất kể thuộc tính truyền của tín hiệu hoặc nhóm tín hiệu đã gây ra thay đổi chế độ truyền. Thời gian trễ tối thiểu và ComTxModeTimeOffset vẫn phải được tuân thủ.

[SWS\_Com\_00582] ⌈Nếu thay đổi TMS gây ra thay đổi sang chế độ truyền DIRECT, một lần truyền trực tiếp/n-lần tới lớp dưới sẽ được khởi tạo ngay lập tức (trong chức năng chính tiếp theo chậm nhất trừ khi bị dời lại do MDT).⌋ (SRS\_Com\_02082)

**Giải thích**: Nếu thay đổi TMS gây ra thay đổi sang chế độ truyền DIRECT, một lần truyền trực tiếp/n-lần tới lớp dưới sẽ được khởi tạo ngay lập tức (trong chức năng chính tiếp theo chậm nhất trừ khi bị dời lại do MDT). Điều này làm rõ hành vi trong trường hợp TMS chuyển sang ComTxModeMode DIRECT do tín hiệu với thuộc tính truyền PENDING.

**Lưu ý**: Nếu ComTxModeMode của một I-PDU được cấu hình là NONE, không có truyền nào sẽ được khởi tạo bởi AUTOSAR COM. Điều này có thể được sử dụng để ngăn chặn việc truyền một I-PDU, ví dụ như trong trường hợp TMS được đánh giá là false.

[SWS\_Com\_00478] ⌈Mô-đun AUTOSAR COM phải gửi một I-PDU tối đa một lần trong mỗi lần gọi của Com\_MainFunctionTx.⌋ (SRS\_Com\_02083)

**Giải thích**: Mô-đun AUTOSAR COM phải gửi một I-PDU tối đa một lần trong mỗi lần gọi của Com\_MainFunctionTx. Điều này đảm bảo rằng I-PDU không được gửi nhiều lần trong cùng một chu kỳ chức năng chính, tránh việc truyền dữ liệu không cần thiết.

##### Signal Flow and Transmission Mode Selection

Sau khi có yêu cầu gửi từ RTE cho một tín hiệu cụ thể

**Giải thích**: Khi có một yêu cầu gửi từ Runtime Environment (RTE) đối với một tín hiệu cụ thể, tín hiệu này sẽ được xử lý theo các bước sau:

1. **Ghi vào bộ đệm I-PDU phù hợp**:
   * Tín hiệu được ghi vào bộ đệm của I-PDU tương ứng như được định nghĩa trong cấu hình hệ thống. Điều này đảm bảo rằng dữ liệu của tín hiệu sẽ được lưu trữ đúng vị trí trong I-PDU để truyền đi sau đó.
2. **Chọn chế độ truyền của I-PDU**:
   * Việc chọn chế độ truyền của các I-PDU sẽ được thực hiện theo quy trình định nghĩa trong Chương 7.3.3.2.

A diagram of a computer network

Description automatically generated

[Figure 3](#_bookmark65) shows the signal flow:

Hình 3: Luồng tín hiệu logic trong mô-đun AUTOSAR COM cho hai tín hiệu (Signal1 và Signal2) được ánh xạ vào một I-PDU (IPDUk)

|  |
| --- |
| Mô tả chi tiết  Hình 3 minh họa luồng tín hiệu logic trong mô-đun AUTOSAR COM cho hai tín hiệu (Signal1 và Signal2) được ánh xạ vào một I-PDU (IPDUk). Quy trình này bao gồm các bước chính sau:   1. **Gửi tín hiệu từ RTE**:    * RTE gửi yêu cầu truyền tín hiệu cho Signal1 và Signal2. 2. **Ghi tín hiệu vào bộ đệm I-PDU**:    * Signal1 và Signal2 được ghi vào bộ đệm của IPDUk theo cấu hình đã định. 3. **Chọn chế độ truyền của I-PDU**:    * Chế độ truyền của IPDUk được xác định dựa trên trạng thái của các tín hiệu chứa trong nó (Signal1 và Signal2).   Các bước chi tiết trong quá trình truyền tín hiệu   1. **Gửi tín hiệu từ RTE**:    * RTE gọi hàm Com\_SendSignal(Signal1) và Com\_SendSignal(Signal2) để gửi tín hiệu. 2. **Ghi tín hiệu vào bộ đệm I-PDU**:    * Signal1 và Signal2 được ghi vào các vị trí tương ứng trong bộ đệm của IPDUk. Việc ánh xạ này được định nghĩa trong cấu hình hệ thống. 3. **Đánh giá TMS (Transmission Mode Selector)**:    * **Xác định các tín hiệu có ComFilter**: Kiểm tra xem Signal1 và Signal2 có cấu hình ComFilter hay không.    * **Đánh giá TMC (Transmission Mode Condition)**: Đánh giá các điều kiện truyền của các tín hiệu này.    * **Xác định TMS**: Nếu bất kỳ TMC nào của các tín hiệu được đánh giá là true, TMS sẽ là true. Nếu không, TMS sẽ là false. 4. **Chọn chế độ truyền**:    * **Nếu TMS là true**: Sử dụng chế độ truyền ComTxModeTrue cho IPDUk.    * **Nếu TMS là false**: Sử dụng chế độ truyền ComTxModeFalse cho IPDUk. 5. **Khởi tạo truyền I-PDU**:    * Nếu TMS thay đổi chế độ truyền của IPDUk, mô-đun AUTOSAR COM phải thực hiện việc truyền IPDUk theo chế độ truyền mới. |

##### Replication of Signal Transmission Requests

Số lần lặp lại của yêu cầu truyền trong ComTxModeMode DIRECT hoặc MIXED

**Giải thích**: Số lần lặp lại của yêu cầu truyền cho một I-PDU trong chế độ truyền DIRECT hoặc MIXED được định nghĩa bởi tham số cấu hình ComTxModeNumberOfRepetitions (ECUC\_Com\_00281).

[SWS\_Com\_00467] ⌈Nếu ComRetryFailedTransmitRequests không được đặt thành TRUE và một I-PDU với ComTxModeMode DIRECT hoặc MIXED và ComTxModeNumberOfRepetitions được đặt thành 0 được kích hoạt để gửi, mô-đun AUTOSAR COM sẽ gọi PduR\_ComTransmit cho I-PDU này chỉ một lần, bất kể kết quả của xác nhận.⌋ (SRS\_Com\_02083)

**Giải thích**: Nếu ComRetryFailedTransmitRequests không được đặt thành TRUE và một I-PDU với chế độ truyền DIRECT hoặc MIXED có ComTxModeNumberOfRepetitions bằng 0 được kích hoạt để gửi, mô-đun AUTOSAR COM sẽ gọi PduR\_ComTransmit cho I-PDU này chỉ một lần, bất kể kết quả xác nhận.

[SWS\_Com\_00279] ⌈Nếu một yêu cầu gửi mới được nhận từ RTE trong khi đang gửi n lần truyền liên tiếp, mô-đun AUTOSAR COM phải hủy các lần truyền còn lại và bắt đầu xử lý yêu cầu mới trong chức năng chính tiếp theo chậm nhất.⌋ (SRS\_Com\_02080)

**Giải thích**: Nếu một yêu cầu gửi mới được nhận từ RTE trong khi đang gửi n lần truyền liên tiếp, mô-đun AUTOSAR COM phải hủy các lần truyền còn lại và bắt đầu xử lý yêu cầu mới trong chức năng chính tiếp theo chậm nhất.

[SWS\_Com\_00305] ⌈Mô-đun AUTOSAR COM sẽ thực hiện việc truyền tới PduR và hành vi xác nhận tới RTE trong ComTxModeMode DIRECT hoặc MIXED với ComTxModeNumberOfRepetitions lớn hơn 0 theo các bước sau:⌋ (SRS\_Com\_02083)

**Giải thích**: Mô-đun AUTOSAR COM sẽ thực hiện việc truyền và hành vi xác nhận theo các bước sau:

1. **Đặt bộ đếm**: Khi một I-PDU được gửi bằng cách sử dụng Com\_SendSignal hoặc Com\_SendSignalGroup, ComTxModeMode được đặt thành DIRECT hoặc MIXED, và ComTransferProperty được đặt thành TRIGGERED, mô-đun AUTOSAR COM sẽ đặt một bộ đếm gán cho I-PDU đó thành ComTxModeNumberOfRepetitions cộng thêm một.
2. **Gọi PduR\_ComTransmit định kỳ**: Mô-đun AUTOSAR COM sẽ gọi PduR\_ComTransmit định kỳ, với khoảng thời gian ComTxModeRepetitionPeriod, miễn là bộ đếm không bằng không.
3. **Giảm bộ đếm**: Bất cứ khi nào nhận được xác nhận TX và bộ đếm lớn hơn 0, mô-đun AUTOSAR COM sẽ giảm bộ đếm. Khi bộ đếm bằng 0, mô-đun AUTOSAR COM sẽ bỏ qua các xác nhận truyền cho I-PDU đó.
4. **Gửi xác nhận tới RTE**: Khi bộ đếm đạt đến 0, mô-đun AUTOSAR COM sẽ gửi xác nhận truyền tới RTE và hủy việc giám sát thời hạn truyền (nếu được cấu hình).

[SWS\_Com\_00494] ⌈Nếu trong chế độ truyền MIXED một yêu cầu truyền N-Times chồng lấp với phần chu kỳ của truyền MIXED, thì truyền chu kỳ sẽ được tính là lần truyền tương ứng của yêu cầu truyền N-Times.⌋ (SRS\_Com\_02083)

**Giải thích**: Nếu trong chế độ truyền MIXED, một yêu cầu truyền N-Times chồng lấp với phần chu kỳ của truyền MIXED, thì truyền chu kỳ sẽ được tính là lần truyền tương ứng của yêu cầu truyền N-Times.

[SWS\_Com\_00392] ⌈Nếu thời hạn giám sát truyền xảy ra trước khi hoàn thành truyền N-Times, mô-đun AUTOSAR COM sẽ không khởi tạo thêm lần truyền nào cho truyền N-Times này.⌋ (SRS\_Com\_02083)

**Giải thích**: Nếu thời hạn giám sát truyền xảy ra trước khi hoàn thành truyền N-Times, mô-đun AUTOSAR COM sẽ không khởi tạo thêm lần truyền nào cho truyền N-Times này.

Khoảng thời gian giữa hai lần lặp lại

**Giải thích**: Thời gian giữa hai lần lặp lại được cấu hình bởi tham số cấu hình ComTxModeRepetitionPeriod (ECUC\_Com\_00282). Nếu thay đổi chế độ truyền dẫn đến việc bắt đầu chế độ truyền MIXED bằng cách gửi một tín hiệu kích hoạt và ComTxModeNumberOfRepetitions được cấu hình lớn hơn hoặc bằng 1, thì sẽ có ít nhất ComTxModeNumberOfRepetitions cộng thêm một yêu cầu truyền tới PDU Router ở đầu chế độ truyền MIXED.

##### Retry Failed Transmission Requests

[SWS\_Com\_00773] ⌈Nếu ComRetryFailedTransmitRequests được đặt thành TRUE, giá trị trả về của PduR\_ComTransmit sẽ được đánh giá. Nếu giá trị trả về của PduR\_ComTransmit không bằng E\_OK, mô-đun AUTOSAR COM sẽ gọi lại PduR\_ComTransmit cho I-PDU chưa được gửi trong lần Com\_MainFunctionTx tiếp theo.⌋ (SRS\_Com\_02037)

**Giải thích**: Khi ComRetryFailedTransmitRequests được đặt thành TRUE, mô-đun AUTOSAR COM sẽ kiểm tra giá trị trả về của PduR\_ComTransmit. Nếu giá trị trả về không bằng E\_OK (nghĩa là truyền không thành công), mô-đun AUTOSAR COM sẽ gọi lại PduR\_ComTransmit cho I-PDU chưa được gửi trong lần Com\_MainFunctionTx tiếp theo. Nếu I-PDU được cập nhật trong khoảng thời gian này, dữ liệu mới sẽ được gửi.

**Lưu ý**:

* Có thể dẫn đến nhiều lần thử lại cho cùng một yêu cầu truyền thất bại.
* Không bị ảnh hưởng bởi các thay đổi chế độ truyền.
* Không ảnh hưởng đến chu kỳ lặp lại tiềm năng. Bất kỳ lần lặp lại nào sẽ xảy ra như thể không có lần thử lại nào diễn ra.

[SWS\_Com\_00774] ⌈Mô-đun AUTOSAR COM sẽ bắt đầu giám sát thời hạn truyền độc lập với các lần thử lại có thể có. Các lần thử lại sau đó sẽ không ảnh hưởng đến giám sát thời hạn truyền.⌋ (SRS\_Com\_02037)

**Giải thích**: Mô-đun AUTOSAR COM sẽ bắt đầu giám sát thời hạn truyền ngay khi bắt đầu quá trình truyền, và quá trình giám sát này sẽ hoạt động độc lập với các lần thử lại có thể có. Điều này có nghĩa là các lần thử lại sẽ không làm gián đoạn hoặc ảnh hưởng đến giám sát thời hạn truyền.

[SWS\_Com\_00775] ⌈Nếu ComRetryFailedTransmitRequests được đặt thành TRUE và một thời hạn giám sát truyền xảy ra cho một I-PDU, mô-đun AUTOSAR COM sẽ hủy mọi yêu cầu truyền đang chờ cho I-PDU này.⌋ (SRS\_Com\_02037)

**Giải thích**: Khi ComRetryFailedTransmitRequests được đặt thành TRUE và thời hạn giám sát truyền hết cho một I-PDU, mô-đun AUTOSAR COM sẽ hủy mọi yêu cầu truyền đang chờ cho I-PDU đó.

[SWS\_Com\_00776] ⌈Bộ đếm thời gian chu kỳ cho truyền chu kỳ sẽ luôn bắt đầu với lần thử truyền đầu tiên.⌋ (SRS\_Com\_02037)

**Giải thích**: Bộ đếm thời gian chu kỳ cho truyền chu kỳ sẽ luôn bắt đầu từ lần thử truyền đầu tiên. Điều này đảm bảo rằng chu kỳ thời gian được tính từ lần truyền đầu tiên, không phải từ các lần thử lại.

#### Signal Invalidation

Mô-đun AUTOSAR COM cung cấp khả năng cho người gửi chỉ ra rằng không thể cung cấp giá trị hợp lệ cho tín hiệu tương ứng, ví dụ trong trường hợp cảm biến bị lỗi. Mô-đun AUTOSAR COM cho phép định nghĩa một ComSignalDataInvalidValue (ECUC\_Com\_00391) trong quá trình cấu hình.

##### Transmission of an Invalidated Signal

[SWS\_Com\_00099] ⌈Khi gọi Com\_InvalidateSignal, mô-đun AUTOSAR COM sẽ thực hiện nội bộ một Com\_SendSignal với giá trị ComSignalDataInvalidValue (ECUC\_Com\_00391) được cấu hình.⌋ (SRS\_Com\_02077)

**Giải thích**: Khi hàm Com\_InvalidateSignal được gọi, mô-đun AUTOSAR COM sẽ thực hiện nội bộ một cuộc gọi đến hàm Com\_SendSignal với giá trị ComSignalDataInvalidValue đã được cấu hình. Thuộc tính truyền (ComTransferProperty) và chế độ truyền sẽ xác định việc truyền giá trị dữ liệu không hợp lệ (ComSignalDataInvalidValue) trên bus. Cuộc gọi Com\_SendSignal nội bộ với giá trị dữ liệu không hợp lệ này sẽ dẫn đến việc giá trị dữ liệu không hợp lệ được sử dụng như giá trị hiện tại cho các bộ lọc và TMS (Transmission Mode Selector).

[SWS\_Com\_00286] ⌈Khi gọi Com\_InvalidateShadowSignal, mô-đun AUTOSAR COM sẽ thay thế giá trị hiện tại của tín hiệu nhóm với SignalId đã cho trong nhóm tín hiệu liên quan bằng giá trị ComSignalDataInvalidValue (ECUC\_Com\_00391) của tín hiệu nhóm.⌋ (SRS\_Com\_02077)

**Giải thích**: Khi hàm Com\_InvalidateShadowSignal được gọi, mô-đun AUTOSAR COM sẽ thay thế giá trị hiện tại của tín hiệu nhóm với SignalId đã cho bằng giá trị ComSignalDataInvalidValue của tín hiệu nhóm đó.

**Lưu ý**:

* Com\_InvalidateShadowSignal hiện đã bị loại bỏ, tín hiệu nhóm nên được làm mất hiệu lực bằng Com\_InvalidateSignal hoặc Com\_InvalidateSignalGroup.
* Các giá trị dữ liệu không hợp lệ được cấu hình cho mỗi tín hiệu nhóm, xem ECUC\_Com\_00520.
* VFB (Virtual Function Bus) chỉ định một thuộc tính cho một kiểu dữ liệu phức tạp. Do đó, ánh xạ tốt nhất của một kiểu dữ liệu phức tạp bị làm mất hiệu lực đến một nhóm tín hiệu bị làm mất hiệu lực là làm mất hiệu lực tất cả các tín hiệu nhóm của một nhóm tín hiệu. Vì vậy, RTE cũng có thể yêu cầu làm mất hiệu lực toàn bộ nhóm tín hiệu bằng cách gọi Com\_InvalidateSignalGroup (SWS\_Com\_00557).

##### Reception of an Invalidated Signal

[SWS\_Com\_00680] ⌈Nếu ComSignalDataInvalidValue (ECUC\_Com\_00391) được cấu hình được nhận cho một tín hiệu và ComDataInvalidAction (ECUC\_Com\_00314) được cấu hình để NOTIFY cho tín hiệu này, mô-đun AUTOSAR COM sẽ thông báo cho RTE qua chức năng ComInvalidNotification (ECUC\_Com\_00315) được cấu hình. Trong trường hợp này, không có xử lý tín hiệu nào khác như lọc hoặc chỉ báo tín hiệu thông thường sẽ diễn ra.⌋ (SRS\_Com\_02079)

**Giải thích**: Nếu ComSignalDataInvalidValue được nhận cho một tín hiệu và ComDataInvalidAction được cấu hình là NOTIFY, mô-đun AUTOSAR COM sẽ thông báo cho RTE qua chức năng ComInvalidNotification. Không có xử lý tín hiệu nào khác như lọc hoặc chỉ báo tín hiệu thông thường sẽ diễn ra.

[SWS\_Com\_00681] ⌈Nếu ComSignalDataInvalidValue (ECUC\_Com\_00391) được cấu hình được nhận cho một tín hiệu và ComDataInvalidAction (ECUC\_Com\_00314) được cấu hình để REPLACE cho tín hiệu này, mô-đun AUTOSAR COM sẽ thay thế giá trị của tín hiệu bằng ComSignalInitValue (ECUC\_Com\_00170) được cấu hình của nó. Sau khi thay thế, việc xử lý tín hiệu thông thường như lọc và thông báo sẽ diễn ra như thể ComSignalInitValue đã được nhận thay vì ComSignalDataInvalidValue.⌋ (SRS\_Com\_02079, SRS\_Com\_02087)

**Giải thích**: Nếu ComSignalDataInvalidValue được nhận và ComDataInvalidAction được cấu hình là REPLACE, mô-đun AUTOSAR COM sẽ thay thế giá trị của tín hiệu bằng ComSignalInitValue. Sau đó, việc xử lý tín hiệu thông thường như lọc và thông báo sẽ diễn ra như thể ComSignalInitValue đã được nhận.

[SWS\_Com\_00682] ⌈Nếu ComSignalDataInvalidValue (ECUC\_Com\_00391) được cấu hình được nhận cho ít nhất một tín hiệu nhóm của một nhóm tín hiệu và ComDataInvalidAction (ECUC\_Com\_00314) được cấu hình để NOTIFY cho nhóm tín hiệu này, mô-đun AUTOSAR COM sẽ thông báo cho RTE qua chức năng ComInvalidNotification (ECUC\_Com\_00315) được cấu hình. Trong trường hợp này, không có xử lý tín hiệu nhóm hoặc nhóm tín hiệu nào khác như chỉ báo thông thường sẽ diễn ra.⌋ (SRS\_Com\_02079)

**Giải thích**: Nếu ComSignalDataInvalidValue được nhận cho ít nhất một tín hiệu nhóm của một nhóm tín hiệu và ComDataInvalidAction được cấu hình là NOTIFY, mô-đun AUTOSAR COM sẽ thông báo cho RTE qua chức năng ComInvalidNotification. Không có xử lý tín hiệu nhóm hoặc nhóm tín hiệu nào khác như chỉ báo thông thường sẽ diễn ra.

[SWS\_Com\_00683] ⌈Nếu ComSignalDataInvalidValue (ECUC\_Com\_00391) được cấu hình được nhận cho ít nhất một tín hiệu nhóm của một nhóm tín hiệu và ComDataInvalidAction (ECUC\_Com\_00314) được cấu hình để REPLACE cho nhóm tín hiệu này, mô-đun AUTOSAR COM sẽ thay thế tất cả các tín hiệu nhóm của nhóm tín hiệu này bằng các giá trị ComSignalInitValue được cấu hình của chúng. Sau khi thay thế, việc xử lý tín hiệu nhóm hoặc nhóm tín hiệu thông thường như thông báo nhận sẽ diễn ra như thể ComSignalInitValue đã được nhận cho tất cả các tín hiệu nhóm.⌋ (SRS\_Com\_02087)

**Giải thích**: Nếu ComSignalDataInvalidValue được nhận cho ít nhất một tín hiệu nhóm của một nhóm tín hiệu và ComDataInvalidAction được cấu hình là REPLACE, mô-đun AUTOSAR COM sẽ thay thế tất cả các tín hiệu nhóm của nhóm tín hiệu này bằng các giá trị ComSignalInitValue. Sau đó, việc xử lý tín hiệu nhóm hoặc nhóm tín hiệu thông thường như thông báo nhận sẽ diễn ra như thể ComSignalInitValue đã được nhận cho tất cả các tín hiệu nhóm.

[SWS\_Com\_00717] ⌈Nếu ComSignalDataInvalidValue được nhận cho một tín hiệu và ComDataInvalidAction của nó được cấu hình để NOTIFY, mô-đun AUTOSAR COM sẽ không lưu trữ ComSignalDataInvalidValue đã nhận vào đối tượng tín hiệu.⌋ (SRS\_Com\_02079)

**Giải thích**: Nếu ComSignalDataInvalidValue được nhận cho một tín hiệu và ComDataInvalidAction được cấu hình là NOTIFY, mô-đun AUTOSAR COM sẽ không lưu trữ giá trị không hợp lệ đã nhận vào đối tượng tín hiệu. Cuộc gọi tiếp theo đến Com\_ReceiveSignal sẽ trả về tín hiệu hợp lệ cuối cùng đã nhận hoặc ComSignalInitValue nếu chưa có tín hiệu nào được nhận.

[SWS\_Com\_00718] ⌈Nếu ComSignalDataInvalidValue được nhận cho ít nhất một tín hiệu nhóm của một nhóm tín hiệu và ComDataInvalidAction của nó được cấu hình để NOTIFY, mô-đun AUTOSAR COM sẽ không lưu trữ bất kỳ tín hiệu nhóm nào đã nhận vào các đối tượng tín hiệu.⌋ (SRS\_Com\_02079)

**Giải thích**: Nếu ComSignalDataInvalidValue được nhận cho ít nhất một tín hiệu nhóm của một nhóm tín hiệu và ComDataInvalidAction được cấu hình là NOTIFY, mô-đun AUTOSAR COM sẽ không lưu trữ bất kỳ tín hiệu nhóm nào đã nhận vào các đối tượng tín hiệu. Cuộc gọi tiếp theo đến Com\_ReceiveSignalGroup sẽ sao chép các tín hiệu nhóm hợp lệ cuối cùng đã nhận hoặc ComSignalInitValue nếu nhóm tín hiệu chưa được nhận vào bộ đệm shadow.

#### Handling of I-PDUs

##### I-PDU Group Definitions

Quy tắc cho một nhóm I-PDU

Quy tắc 1: Một I-PDU có thể thuộc bất kỳ nhóm I-PDU nào.

**Giải thích**: Một I-PDU (Interface Protocol Data Unit) có thể được gán vào nhiều nhóm I-PDU khác nhau. Điều này cho phép linh hoạt trong việc quản lý và điều khiển các I-PDU trong hệ thống.

Quy tắc 2: [SWS\_Com\_00771] ⌈Một I-PDU hoạt động (được khởi động) nếu và chỉ nếu ít nhất một nhóm I-PDU mà nó thuộc về đang hoạt động (được khởi động).⌋ (SRS\_Com\_00218)

**Giải thích**: Một I-PDU sẽ ở trạng thái hoạt động (được khởi động) nếu ít nhất một trong các nhóm I-PDU mà nó thuộc về đang hoạt động. Ngược lại, nếu tất cả các nhóm I-PDU mà I-PDU đó thuộc về đều không hoạt động, thì I-PDU đó cũng sẽ không hoạt động.

Quy tắc 3: Số lượng tối đa của các nhóm I-PDU có thể cấu hình trước khi biên dịch.

**Giải thích**: Số lượng tối đa các nhóm I-PDU mà hệ thống có thể hỗ trợ sẽ được xác định và cấu hình trước khi biên dịch mã nguồn. Điều này đảm bảo rằng hệ thống có thể xử lý đúng số lượng nhóm I-PDU đã được xác định.

Cấu hình và hỗ trợ

* **Cấu hình COM**: Quy tắc 1 và 3 được hỗ trợ bởi cấu hình COM. Số lượng tối đa các nhóm I-PDU được hỗ trợ có thể được cấu hình thông qua ComSupportedIpduGroups (ECUC\_Com\_00710).
* **Kích hoạt I-PDU**: Một I-PDU được coi là kích hoạt (hoặc khởi động) nếu bất kỳ nhóm I-PDU nào chứa I-PDU đó được kích hoạt. Ngược lại, nếu không có nhóm I-PDU nào chứa I-PDU đó được kích hoạt, thì I-PDU đó sẽ được coi là dừng hoặc hủy kích hoạt.

Sự lồng ghép của các nhóm I-PDU

**Giải thích**: Sự lồng ghép của các nhóm I-PDU chỉ mang tính khái niệm và phải được giải quyết bằng cấu hình phù hợp và/hoặc sử dụng mô-đun AUTOSAR COM. Ví dụ, nếu một I-PDU thuộc về nhóm I-PDU nhỏ hơn và nhóm I-PDU này lại thuộc về một nhóm I-PDU lớn hơn, thì I-PDU đó cũng phải được bao gồm trong nhóm I-PDU lớn hơn.

Quản lý trạng thái của các nhóm I-PDU

**Giải thích**: Trạng thái hoàn chỉnh của các nhóm I-PDU được xử lý bên ngoài mô-đun AUTOSAR COM, ví dụ trong Trình Quản Lý Chế Độ Phần Mềm Cơ Bản. Khi có sự thay đổi trạng thái, mô-đun quản lý trạng thái nhóm I-PDU sẽ truyền trạng thái nhất quán của tất cả các nhóm I-PDU thông qua Com\_IpduGroupControl tới mô-đun AUTOSAR COM, và sau đó mô-đun này sẽ khởi động hoặc dừng các I-PDU tương ứng.

Ngoại lệ

[SWS\_Com\_00840] ⌈Nếu một I-PDU không được gán cho bất kỳ nhóm I-PDU nào, I-PDU sẽ được khởi động ban đầu (và không bao giờ bị dừng).⌋ (SRS\_Com\_00218)

**Giải thích**: Nếu một I-PDU không được gán cho bất kỳ nhóm I-PDU nào, nó sẽ được khởi động ban đầu và sẽ không bao giờ bị dừng. Điều này đảm bảo rằng các I-PDU không thuộc bất kỳ nhóm nào vẫn có thể hoạt động mà không cần phụ thuộc vào trạng thái của nhóm I-PDU.

A diagram of a business flowchart

Description automatically generated

CAN TX UC1

CAN TX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN TX UC1

CAN TX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN RX UC1

CAN TX UC1

CAN TX UC1

CAN TX UC1

CAN TX UC1

**Figure 4: Example usage of I-PDU groups**

|  |
| --- |
| Mô tả hình ảnh  Hình ảnh minh họa cấu trúc và sự phụ thuộc giữa các I-PDU và nhóm I-PDU trong hệ thống AUTOSAR COM. Các yếu tố chính bao gồm:   1. **I-PDUs và Nhóm I-PDU**:    * I-PDUs được nhóm lại thành các nhóm I-PDU.    * Các nhóm I-PDU có thể bao gồm nhiều I-PDU và có thể lồng ghép nhau. 2. **Ví dụ về Nhóm I-PDU**:    * **BUS1**: Bao gồm các nhóm I-PDU BUS1 RX và BUS1 TX.    * **BUS2**: Bao gồm các nhóm I-PDU BUS2 RX và BUS2 TX. 3. **Chức năng (Function)**:    * Các nhóm I-PDU như BUS1 RX, BUS1 TX, BUS2 RX, và BUS2 TX bao gồm các chức năng cụ thể như Function1, Function2, Function3. 4. **Chế độ (Mode)**:    * Các chức năng này tương tác với các chế độ hoạt động khác nhau như Mode1, Mode2, Mode3. 5. **Kết nối**:    * Các mũi tên thể hiện sự liên kết giữa các nhóm I-PDU và các chức năng, và từ các chức năng tới các chế độ.   Quy tắc và Mô tả   1. **Một I-PDU có thể thuộc bất kỳ nhóm I-PDU nào**:    * Ví dụ, BUS1 RX Function1 thuộc nhóm I-PDU BUS1 RX và BUS2 RX. 2. **Một I-PDU hoạt động nếu ít nhất một nhóm I-PDU mà nó thuộc về đang hoạt động**:    * Nếu BUS1 RX hoặc BUS2 RX đang hoạt động, thì BUS1 RX Function1 cũng sẽ hoạt động. 3. **Số lượng tối đa các nhóm I-PDU có thể cấu hình trước khi biên dịch**:    * Điều này được cấu hình thông qua ComSupportedIpduGroups (ECUC\_Com\_00710).   Quản lý và Điều khiển   * **Cấu hình và hỗ trợ**: Quy tắc 1 và 3 được hỗ trợ bởi cấu hình COM. * **Kích hoạt I-PDU**: Một I-PDU được coi là kích hoạt nếu bất kỳ nhóm I-PDU nào chứa I-PDU đó được kích hoạt. * **Sự lồng ghép của các nhóm I-PDU**: Các nhóm I-PDU có thể lồng ghép nhau, như trong ví dụ BUS1 RX thuộc về nhóm BUS1. * **Quản lý trạng thái của các nhóm I-PDU**: Trạng thái của các nhóm I-PDU được quản lý bởi mô-đun bên ngoài, ví dụ như Trình Quản Lý Chế Độ Phần Mềm Cơ Bản. * **Ngoại lệ**: Nếu một I-PDU không được gán cho bất kỳ nhóm I-PDU nào, nó sẽ được khởi động ban đầu và không bao giờ bị dừng. |

##### Starting of I-PDU Groups

Mặc định và cách khởi động nhóm I-PDU

* **Mặc định tất cả các nhóm I-PDU đều dừng**: Theo quy tắc SWS\_Com\_00444, mặc định tất cả các nhóm I-PDU đều dừng.
* **Khởi động nhóm I-PDU**: Gọi hàm Com\_IpduGroupControl để khởi động một nhóm I-PDU nếu trạng thái kích hoạt yêu cầu của nhóm I-PDU là khởi động và nhóm I-PDU trước đó đã dừng.

[SWS\_Com\_00114] ⌈Nếu một I-PDU được khởi động như là kết quả của cuộc gọi đến Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ cho phép truyền/nhận các tín hiệu và nhóm tín hiệu của nó.⌋ (SRS\_Com\_00218)

**Giải thích**: Khi một I-PDU được khởi động bởi Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ cho phép truyền và nhận các tín hiệu và nhóm tín hiệu của I-PDU đó.

[SWS\_Com\_00787] ⌈Nếu một I-PDU được khởi động bởi Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ luôn khởi tạo các thuộc tính sau của I-PDU này:

1. ComMinimumDelayTime của I-PDUs trong chế độ truyền DIRECT hoặc MIXED
2. Các thuộc tính thời gian chờ của I-PDUs để giám sát thời hạn: tất cả các bộ đếm thời gian chờ (ComFirstTimeout, ComTimeout) sẽ được khởi động lại
3. Tất cả các bit cập nhật được bao gồm sẽ được xóa
4. Đặt lại OCCURRENCE của bộ lọc với ComFilterAlgorithm ONE\_EVERY\_N
5. Đặt bộ đếm I-PDU về 0 cho I-PDUs với ComIPduDirection được cấu hình là SEND
6. Chấp nhận cho I-PDUs với ComIPduDirection được cấu hình là RECEIVED bất kỳ bộ đếm I-PDU tiếp theo nào⌋ (SRS\_Com\_00218)

**Giải thích**: Khi một I-PDU được khởi động bởi Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ khởi tạo các thuộc tính sau của I-PDU này theo các yêu cầu trên.

[SWS\_Com\_00222] ⌈Nếu một I-PDU được khởi động bởi Com\_IpduGroupControl với tham số Initialize được đặt thành true, mô-đun AUTOSAR COM sẽ khởi tạo thêm các thuộc tính sau của I-PDU này:

1. Dữ liệu của I-PDU như được định nghĩa trong SWS\_Com\_00217
2. Các bộ đệm shadow của các nhóm tín hiệu được bao gồm
3. old\_value của các cơ chế lọc cho mỗi tín hiệu thành ComSignalInitValue
4. ComTxModeTimePeriod và ComTxModeTimeOffset của I-PDUs trong chế độ truyền Periodic hoặc MIXED⌋ (SRS\_Com\_00218)

**Giải thích**: Nếu tham số Initialize được đặt thành true, mô-đun AUTOSAR COM sẽ khởi tạo thêm các thuộc tính này cho I-PDU.

[SWS\_Com\_00223] ⌈Nếu một I-PDU được khởi động bởi cuộc gọi tới Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ xác định chế độ truyền của nó dựa trên nội dung dữ liệu hiện tại.⌋ (SRS\_Com\_00218)

**Giải thích**: Khi một I-PDU được khởi động, mô-đun AUTOSAR COM sẽ xác định chế độ truyền của I-PDU dựa trên nội dung dữ liệu hiện tại.

[SWS\_Com\_00228] ⌈Trong một số trường hợp, một I-PDU được khởi động bởi cuộc gọi tới Com\_IpduGroupControl trước khi tất cả các tín hiệu của nó đã được ghi. Trong trường hợp này, mô-đun AUTOSAR COM sẽ sử dụng ComSignalInitValue cho dữ liệu tín hiệu còn thiếu.⌋ (SRS\_Com\_00218)

**Giải thích**: Nếu một I-PDU được khởi động trước khi tất cả các tín hiệu của nó đã được ghi, mô-đun AUTOSAR COM sẽ sử dụng ComSignalInitValue cho các dữ liệu tín hiệu còn thiếu.

[SWS\_Com\_00229] ⌈Khi một I-PDU được khởi động và một hoặc nhiều tín hiệu trong I-PDU đã được ghi qua một trong các API gửi bởi lớp trên, mô-đun AUTOSAR COM sẽ sử dụng các giá trị mới nhất đã nhận được để xác định TMS của I-PDU.⌋ (SRS\_Com\_00218)

**Giải thích**: Khi một I-PDU được khởi động và một hoặc nhiều tín hiệu đã được ghi, mô-đun AUTOSAR COM sẽ sử dụng các giá trị mới nhất đã nhận được để xác định TMS của I-PDU.

[SWS\_Com\_00733] ⌈Nếu một I-PDU được khởi động bởi cuộc gọi tới Com\_IpduGroupControl và I-PDU chứa các tín hiệu có cấu hình giám sát thời hạn, mô-đun AUTOSAR COM sẽ bắt đầu giám sát thời hạn cho các tín hiệu này bất kể giá trị của tham số initialize.⌋ (SRS\_Com\_00218)

**Giải thích**: Nếu một I-PDU được khởi động và có các tín hiệu có cấu hình giám sát thời hạn, mô-đun AUTOSAR COM sẽ bắt đầu giám sát thời hạn cho các tín hiệu này bất kể giá trị của tham số initialize.

Lưu ý về Initialize

**Giải thích**: Nếu một I-PDU được khởi động bởi cuộc gọi tới Com\_IpduGroupControl và tham số initialize được đặt thành false, các bộ đệm nội bộ sẽ không thay đổi và I-PDU sẽ chứa các giá trị cuối cùng đã đặt hoặc giá trị khởi tạo. Đặc biệt, nếu I-PDU được khởi động lần đầu tiên và initialize được đặt thành false, mô-đun COM sẽ hoạt động trên các giá trị khởi tạo được đặt bởi Com\_Init.

##### Stopping of I-PDU Groups

Quy trình dừng nhóm I-PDU

1. **Dừng nhóm I-PDU**:
   * Gọi hàm Com\_IpduGroupControl để dừng một nhóm I-PDU nếu trạng thái kích hoạt yêu cầu của nhóm I-PDU là dừng và nhóm I-PDU trước đó đã được khởi động.
   * Để vô hiệu hóa việc truyền một I-PDU, tất cả các nhóm I-PDU chứa I-PDU đó cần phải được dừng.
2. **Chế độ chỉ lắng nghe (listen-only-mode)**:
   * Để triển khai chế độ chỉ lắng nghe, tất cả các nhóm I-PDU chứa I-PDU truyền cần phải được dừng. Việc nhận I-PDU cũng có thể bị dừng.

Các yêu cầu và hành vi khi dừng I-PDU

[SWS\_Com\_00334] ⌈Bằng một cuộc gọi tới các hàm: Com\_SendSignal, Com\_SendSignalGroup, hoặc Com\_InvalidateSignal, mô-đun AUTOSAR COM sẽ cập nhật các giá trị của bộ đệm nội bộ ngay cả đối với I-PDUs đã dừng.⌋ (SRS\_Com\_00218)

**Giải thích**: Ngay cả khi I-PDU đã dừng, các giá trị của bộ đệm nội bộ vẫn sẽ được cập nhật khi gọi các hàm Com\_SendSignal, Com\_SendSignalGroup, hoặc Com\_InvalidateSignal.

* Nếu một tín hiệu được ghi vào một I-PDU đã dừng và tín hiệu này sẽ kích hoạt việc truyền I-PDU nếu nó không bị dừng, thì kích hoạt này sẽ không được lưu trữ. Sau khi khởi động lại nhóm I-PDU tương ứng, kích hoạt cũ này không dẫn đến việc truyền I-PDU ngay lập tức.

[SWS\_Com\_00777] ⌈Nếu một I-PDU được dừng bởi cuộc gọi tới Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ hủy bỏ bất kỳ yêu cầu truyền nào đang chờ cho I-PDU này. Điều này bao gồm việc hủy bỏ bất kỳ lần thử lại nào liên quan đến ComRetryFailedTransmitRequests.⌋ (SRS\_Com\_00218)

**Giải thích**: Khi một I-PDU được dừng, mô-đun AUTOSAR COM sẽ hủy bỏ tất cả các yêu cầu truyền đang chờ cho I-PDU này, bao gồm cả các lần thử lại.

[SWS\_Com\_00115] ⌈Nếu một I-PDU được dừng bởi cuộc gọi tới Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ hủy bỏ giám sát thời hạn cho tất cả các xác nhận đang chờ.⌋ (SRS\_Com\_00218)

**Giải thích**: Khi một I-PDU được dừng, mô-đun AUTOSAR COM sẽ hủy bỏ giám sát thời hạn cho tất cả các xác nhận đang chờ.

[SWS\_Com\_00800] ⌈Mô-đun AUTOSAR COM sẽ bỏ qua bất kỳ xác nhận truyền nào cho một I-PDU đã dừng.⌋ (SRS\_Com\_00218)

**Giải thích**: Khi một I-PDU đã dừng, mô-đun AUTOSAR COM sẽ bỏ qua mọi xác nhận truyền cho I-PDU đó.

* Mô-đun AUTOSAR COM không thể ngăn chặn việc gọi hàm Com\_TriggerTransmit. Tuy nhiên, trong trường hợp I-PDU đã dừng, hàm Com\_TriggerTransmit sẽ trả về E\_NOT\_OK.

[SWS\_Com\_00684] ⌈Nếu một I-PDU được dừng bởi cuộc gọi tới Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ vô hiệu hóa quá trình xử lý nhận của nó.⌋ (SRS\_Com\_00218)

**Giải thích**: Khi một I-PDU được dừng, mô-đun AUTOSAR COM sẽ vô hiệu hóa quá trình xử lý nhận của I-PDU đó.

[SWS\_Com\_00713] ⌈Nếu một I-PDU lớn được dừng bởi cuộc gọi tới Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ dừng quá trình nhận và bỏ qua I-PDU đã nhận một phần.⌋ (SRS\_Com\_00218)

**Giải thích**: Khi một I-PDU lớn được dừng, mô-đun AUTOSAR COM sẽ dừng quá trình nhận và bỏ qua phần I-PDU đã nhận một phần.

[SWS\_Com\_00685] ⌈Nếu một I-PDU được dừng bởi cuộc gọi tới Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ hủy bỏ giám sát thời hạn của nó.⌋ (SRS\_Com\_00218)

**Giải thích**: Khi một I-PDU được dừng, mô-đun AUTOSAR COM sẽ hủy bỏ giám sát thời hạn của I-PDU đó.

[SWS\_Com\_00479] ⌈Nếu một I-PDU được dừng bởi cuộc gọi tới Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ ngay lập tức gọi hàm ComErrorNotification được cấu hình (ECUC\_Com\_00499) cho các tín hiệu/tín hiệu nhóm truyền chưa được xác nhận của I-PDU đã dừng.⌋ (SRS\_Com\_00218)

**Giải thích**: Khi một I-PDU được dừng, mô-đun AUTOSAR COM sẽ ngay lập tức gọi hàm ComErrorNotification cho các tín hiệu hoặc nhóm tín hiệu truyền chưa được xác nhận của I-PDU đó.

[SWS\_Com\_00714] ⌈Nếu một I-PDU lớn được dừng trong khi quá trình truyền của nó đang diễn ra, mô-đun AUTOSAR COM sẽ dừng quá trình truyền ngay lập tức.⌋ (SRS\_Com\_00218)

**Giải thích**: Nếu một I-PDU lớn được dừng trong khi quá trình truyền đang diễn ra, mô-đun AUTOSAR COM sẽ dừng quá trình truyền ngay lập tức.

|  |  |
| --- | --- |
| Hành vi của các I-PDUs khi dừng | |
| Bên nhận (RX) | Bên truyền (TX) |
|  **Vô hiệu hóa giám sát thời hạn nhận (RX deadline monitoring)**   **Không thực hiện hành động nào khi nhận được Com\_RxIndication từ RTE**, không lưu trữ I-PDU   **Mã trả về**: COM\_SERVICE\_NOT\_AVAILABLE cho Com\_ReceiveSignal và Com\_ReceiveSignalGroup. Giá trị cuối cùng đã biết (hoặc giá trị khởi tạo) được trả về làm dữ liệu. |   **Vô hiệu hóa việc gửi**   **Vô hiệu hóa giám sát thời hạn truyền (TX deadline monitoring)**   **Bỏ qua Com\_TxConfirmation**   Khi gọi Com\_SendSignal, Com\_SendSignalGroup, Com\_InvalidateSignal:   * Giá trị trong bộ đệm nội bộ của mô-đun AUTOSAR COM vẫn được cập nhật nhưng mã trả về là COM\_SERVICE\_NOT\_AVAILABLE    **Hủy bỏ các yêu cầu truyền đang chờ (ví dụ N-Times)**   *  Mã trả về E\_NOT\_OK cho Com\_TriggerTransmit |
|  | Đối với I-PDUs định kỳ (TX) |
|  | không gửi thêm nữa |

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

[Table 4](#_bookmark82) gives an overview of the behavior of started I-PDUs:

|  |  |
| --- | --- |
| Hành vi của các I-PDUs khi khởi động | |
| Bên nhận (RX) | Bên truyền (TX) |
|  **Khởi tạo lại thời gian chờ nếu Initialize == true** (ComSignalFirstTimeout, ComSignalTimeout)   **Phản ứng bình thường với Com\_RxIndication / Com\_TpRxIndication**   *  **Phản ứng bình thường với Com\_ReceiveSignal, và Com\_ReceiveSignalGroup** |  **Phản ứng bình thường với Com\_InvalidateSignal, Com\_SendSignal, và Com\_SendSignalGroup**   **Không có thông báo thời gian chờ truyền cho đến lần gửi tiếp theo**   **Phản ứng bình thường với Com\_TxConfirmation / Com\_TpTxConfirmation**   *  **Phản ứng bình thường với Com\_TriggerTransmit** |
|  | Đối với I-PDUs định kỳ (TX) |
|  | Bắt đầu từ 0 |

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

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

Để hỗ trợ cả hệ thống điều khiển bằng ngắt và thăm dò, có thể cấu hình khi tín hiệu được chỉ định. Có hai chế độ chỉ định tín hiệu có thể cấu hình: **IMMEDIATE** và **DEFERRED** thông qua ComIPduSignalProcessing. Xem ECUC\_Com\_00119.

[SWS\_Com\_00300] ⌈Nếu ComIPduSignalProcessing cho một I-PDU được cấu hình là **IMMEDIATE**, mô-đun AUTOSAR COM sẽ gọi các ComNotifications đã cấu hình cho các tín hiệu và nhóm tín hiệu bao gồm trong hàm Com\_RxIndication hoặc Com\_TpRxIndication tương ứng.⌋ (SRS\_Com\_02046)

**Giải thích**: Khi ComIPduSignalProcessing được cấu hình là **IMMEDIATE**, mô-đun AUTOSAR COM sẽ gọi các thông báo đã cấu hình ngay lập tức trong hàm Com\_RxIndication hoặc Com\_TpRxIndication.

[SWS\_Com\_00301] ⌈Nếu ComIPduSignalProcessing cho một I-PDU được cấu hình là **DEFERRED**, mô-đun AUTOSAR COM sẽ sao chép dữ liệu của I-PDU trong hàm Com\_RxIndication hoặc các hàm nhận TP liên quan từ PduR vào COM. Sau đó, mô-đun AUTOSAR COM sẽ gọi các ComNotifications đã cấu hình cho các tín hiệu và nhóm tín hiệu bao gồm một cách không đồng bộ trong lần gọi tiếp theo tới hàm Com\_MainFunctionRx.⌋ (SRS\_Com\_02046)

**Giải thích**: Khi ComIPduSignalProcessing được cấu hình là **DEFERRED**, mô-đun AUTOSAR COM sẽ sao chép dữ liệu của I-PDU vào COM trong hàm Com\_RxIndication và gọi các thông báo đã cấu hình trong lần gọi tiếp theo tới hàm Com\_MainFunctionRx.

* Nếu trong chế độ **DEFERRED**, một cuộc gọi tới Com\_ReceiveSignal được thực hiện trước khi giải nén hoãn lại, các giá trị trước đó chưa được cập nhật sẽ được trả về.

[SWS\_Com\_00574] ⌈Khi giải nén một I-PDU, mô-đun AUTOSAR COM sẽ kiểm tra độ dài dữ liệu nhận được (PduInfoPtr->SduLength) và chỉ giải nén và thông báo các tín hiệu nhận được hoàn toàn qua ComNotification.⌋ (SRS\_Com\_02046)

**Giải thích**: Mô-đun AUTOSAR COM sẽ kiểm tra độ dài dữ liệu nhận được và chỉ giải nén và thông báo các tín hiệu đã nhận hoàn toàn.

[SWS\_Com\_00794] ⌈Trong trường hợp nhận một I-PDU ngắn hơn so với dự kiến dẫn đến việc nhận một tín hiệu mà không có bit cập nhật đã cấu hình, mô-đun AUTOSAR COM sẽ xử lý tín hiệu này như thể bit cập nhật đã được đặt và coi tín hiệu đó là đã được cập nhật.⌋ (SRS\_Com\_02046)

**Giải thích**: Nếu độ dài I-PDU nhận được ngắn hơn so với cấu hình/ mong đợi, cần phải ngăn chặn việc tín hiệu được cập nhật một phần. Tuy nhiên, tất cả các tín hiệu đã nhận hoàn toàn nên được nhận và thông báo lên lớp trên.

[SWS\_Com\_00575] ⌈Khi giải nén một I-PDU, mô-đun AUTOSAR COM sẽ kiểm tra độ dài dữ liệu nhận được (PduInfoPtr->SduLength) và trong trường hợp một nhóm tín hiệu chỉ được nhận một phần, nhóm tín hiệu đó và tất cả các tín hiệu nhóm bao gồm sẽ không được giải nén hoặc thông báo qua ComNotification.⌋ (SRS\_Com\_02046)

**Giải thích**: Nếu một nhóm tín hiệu chỉ được nhận một phần, nhóm tín hiệu đó và tất cả các tín hiệu nhóm bao gồm sẽ không được giải nén hoặc thông báo để tránh nhận dữ liệu không nhất quán.

* Mô-đun AUTOSAR COM không sao chép hoặc xử lý dữ liệu nhận thêm cho các tín hiệu không được cấu hình trong trường hợp độ dài dữ liệu nhận được lớn hơn so với mong đợi.

##### Minimum Delay Timer (MDT)

Mô-đun AUTOSAR COM thừa kế cơ chế bộ đếm thời gian trễ tối thiểu (MDT) như đã định nghĩa trong [17]. Cơ chế OSEK COM được làm rõ và chi tiết trong chương này. Khi một I-PDU được khởi động, MDT được khởi tạo lại. Do đó, MDT có thể bị vi phạm khi dừng và khởi động I-PDU một cách nhanh chóng. Hành vi của bộ đếm thời gian giám sát thời hạn truyền không bị ảnh hưởng bởi bất kỳ trễ truyền nào do giám sát thời gian trễ tối thiểu gây ra.

Quy tắc và yêu cầu

[SWS\_Com\_00471] ⌈Không có giám sát thời gian trễ tối thiểu nếu ComMinimumDelayTime bị bỏ qua hoặc cấu hình là 0.⌋ (SRS\_Com\_02037)

**Giải thích**: Nếu ComMinimumDelayTime không được cấu hình hoặc cấu hình là 0, không có giám sát thời gian trễ tối thiểu sẽ được thực hiện.

[SWS\_Com\_00789] ⌈Nếu ComEnableMDTForCyclicTransmission (ECUC\_Com\_00788) được cấu hình là false, bộ đếm thời gian trễ tối thiểu sẽ không được (tái) khởi động trong các trường hợp sau:

* I-PDUs với ComTxModeMode PERIODIC
* Phần chu kỳ của I-PDUs với ComTxModeMode MIXED
* Truyền lại cho I-PDUs trong trường hợp ComTxModeNumberOfRepetitions > 0⌋ (SRS\_Com\_02037)

**Giải thích**: Nếu ComEnableMDTForCyclicTransmission được cấu hình là false, bộ đếm thời gian trễ tối thiểu sẽ không được khởi động lại trong các trường hợp nêu trên. Nếu bộ đếm thời gian trễ tối thiểu hiện không khởi động, sẽ không có cách nào để bộ đếm này hết hạn. Do đó, các triển khai tối ưu có thể bỏ qua hoàn toàn việc giám sát thời gian trễ tối thiểu cho các I-PDUs định kỳ nếu ComEnableMDTForCyclicTransmission được cấu hình là false.

[SWS\_Com\_00698] ⌈Nếu ComMinimumDelayTime của một I-PDU được cấu hình lớn hơn 0 và ComRetryFailedTransmitRequests được đặt thành FALSE, mô-đun AUTOSAR COM sẽ nạp và khởi động bộ đếm thời gian trễ tối thiểu khi truyền I-PDU đó tới PDU Router thông qua PduR\_ComTransmit.⌋ (SRS\_Com\_02037)

**Giải thích**: Khi truyền I-PDU tới PDU Router, mô-đun AUTOSAR COM sẽ nạp và khởi động bộ đếm thời gian trễ tối thiểu nếu ComMinimumDelayTime lớn hơn 0 và ComRetryFailedTransmitRequests là FALSE.

[SWS\_Com\_00828] ⌈Nếu ComMinimumDelayTime của một I-PDU được cấu hình lớn hơn 0 và ComRetryFailedTransmitRequests được đặt thành TRUE, mô-đun AUTOSAR COM sẽ nạp và khởi động bộ đếm thời gian trễ tối thiểu khi truyền I-PDU đó tới PDU Router thông qua PduR\_ComTransmit chỉ trong trường hợp PduR\_ComTransmit trả về E\_OK.⌋ (SRS\_Com\_02037)

**Giải thích**: Nếu ComRetryFailedTransmitRequests là TRUE, bộ đếm thời gian trễ tối thiểu sẽ chỉ được khởi động nếu PduR\_ComTransmit trả về E\_OK.

[SWS\_Com\_00469] ⌈Nếu ComMinimumDelayTime của một I-PDU được cấu hình lớn hơn 0, mô-đun AUTOSAR COM sẽ (tái) nạp bộ đếm thời gian trễ tối thiểu đang chạy với ComMinimumDelayTime cho I-PDU đó khi Com\_TxConfirmation được gọi và bộ đếm thời gian trễ tối thiểu bắt đầu tại PduR\_ComTransmit của I-PDU đó chưa hết hạn.⌋ (SRS\_Com\_02037)

**Giải thích**: Bộ đếm thời gian trễ tối thiểu đang chạy sẽ được nạp lại khi nhận được xác nhận truyền của I-PDU, trừ khi truyền đã bị trễ lâu hơn ComMinimumDelayTime khi nhận được xác nhận. Trong trường hợp bình thường, sẽ không có truyền nào nữa của I-PDU đó cho đến khi thời gian trễ tối thiểu đã được nạp và khởi động hết hạn.

* Tuy nhiên, một số ngoại lệ tồn tại: Theo SWS\_Com\_00475, Com\_TriggerTransmit không can thiệp vào bộ đếm thời gian trễ tối thiểu. Thêm vào đó, bộ đếm thời gian trễ tối thiểu sẽ được đặt lại nếu bộ đếm thời gian giám sát thời hạn truyền hết hạn.

[SWS\_Com\_00812] ⌈Nếu bộ đếm thời gian trễ tối thiểu của một I-PDU được đặt lại vì bộ đếm thời gian giám sát thời hạn truyền hết hạn và có yêu cầu hoãn lại cho I-PDU này, mô-đun AUTOSAR COM sẽ ngay lập tức truyền I-PDU này.⌋ (SRS\_Com\_02037)

**Giải thích**: Nếu bộ đếm thời gian trễ tối thiểu được đặt lại do bộ đếm thời gian giám sát thời hạn truyền hết hạn và có yêu cầu hoãn lại cho I-PDU này, mô-đun AUTOSAR COM sẽ ngay lập tức truyền I-PDU.

#### Deadline Monitoring

Khái niệm

Giám sát thời hạn nhận đảm bảo rằng các tín hiệu và nhóm tín hiệu trong hệ thống được nhận trong khoảng thời gian quy định. Nếu không, hệ thống sẽ thực hiện các hành động dựa trên cấu hình.

Cấu hình và cơ chế giám sát

* **ComFirstTimeout** và **ComTimeout**: Được định nghĩa trong container cấu hình ComSignal hoặc ComSignalGroup.
* **ComTimeoutNotification**: Hàm callback thông báo thời gian chờ, được định nghĩa trong container cấu hình ComSignal hoặc ComSignalGroup.

Quy tắc và yêu cầu

[SWS\_Com\_00333] ⌈Nếu tham số cấu hình ComTimeout cho một tín hiệu hoặc nhóm tín hiệu bị bỏ qua hoặc cấu hình là 0, mô-đun AUTOSAR COM sẽ không giám sát tín hiệu hoặc nhóm tín hiệu này. Nó cũng sẽ bỏ qua ComFirstTimeout.⌋ (SRS\_Com\_02089)

**Giải thích**: Không có giám sát thời hạn nếu ComTimeout bị bỏ qua hoặc bằng 0.

##### Reception Deadline Monitoring

Giám sát thời hạn nhận có thể được bật hoặc tắt bằng cách điều khiển các nhóm I-PDU tương tự như Chương 7.3.5.1.

[SWS\_Com\_00772] ⌈Giám sát thời hạn nhận của một I-PDU được bật nếu và chỉ nếu nó nằm trong một nhóm I-PDU có giám sát thời hạn nhận được bật. Nếu không, giám sát thời hạn nhận của I-PDU sẽ bị tắt.⌋ (SRS\_Com\_00192)

**Giải thích**: Giám sát thời hạn nhận của một I-PDU chỉ hoạt động khi nhóm I-PDU chứa nó có giám sát thời hạn nhận được bật.

[SWS\_Com\_00292] ⌈Khi giám sát thời hạn nhận được cấu hình cho các tín hiệu với các bit cập nhật, mô-đun AUTOSAR COM sẽ thực hiện giám sát thời hạn nhận riêng cho mỗi tín hiệu/nhóm tín hiệu với một bit cập nhật.⌋ (SRS\_Com\_02089, SRS\_Com\_02058)

**Giải thích**: Giám sát thời hạn nhận được thực hiện riêng cho mỗi tín hiệu hoặc nhóm tín hiệu có bit cập nhật.

[SWS\_Com\_00290] ⌈Mô-đun AUTOSAR COM sẽ thực hiện giám sát thời hạn nhận dựa trên I-PDU cho các tín hiệu không có bit cập nhật.⌋ (SRS\_Com\_02089, SRS\_Com\_02058)

**Giải thích**: Giám sát thời hạn nhận được thực hiện dựa trên I-PDU cho các tín hiệu không có bit cập nhật.

[SWS\_Com\_00291] ⌈Đối với tất cả các tín hiệu và nhóm tín hiệu không có bit cập nhật trong cùng một I-PDU, mô-đun AUTOSAR COM sẽ thực hiện giám sát thời hạn nhận của I-PDU bằng cách sử dụng tham số thời gian chờ nhỏ nhất không bằng không (ComFirstTimeout, ComTimeout) của các tín hiệu và nhóm tín hiệu liên quan.⌋ (SRS\_Com\_02089, SRS\_Com\_02058)

**Giải thích**: Giám sát thời hạn nhận cho các tín hiệu và nhóm tín hiệu không có bit cập nhật trong cùng một I-PDU sẽ sử dụng tham số thời gian chờ nhỏ nhất không bằng không.

* Nếu tất cả các tín hiệu trong một I-PDU có cấu hình giám sát thời hạn nhận cũng có bit cập nhật, không cần thực hiện giám sát thời hạn nhận trên cơ sở I-PDU.

Hành động khi hết thời gian chờ (Rx-timeout)

* **ComRxDataTimeoutAction**: Quyết định xem mô-đun AUTOSAR COM thay thế giá trị tín hiệu/nhóm tín hiệu bằng giá trị khởi tạo hoặc giữ giá trị cuối cùng nhận được. Xem ECUC\_Com\_00314.

[SWS\_Com\_00470] ⌈Nếu ComRxDataTimeoutAction được đặt thành REPLACE (xem ECUC\_Com\_00412), mô-đun AUTOSAR COM sẽ thay thế giá trị của tín hiệu bằng giá trị khởi tạo (ComSignalInitValue) khi bộ đếm thời gian giám sát thời hạn nhận của tín hiệu hết hạn.⌋ (SRS\_Com\_02087, SRS\_Com\_02088)

**Giải thích**: Giá trị khởi tạo (ComSignalInitValue) sẽ thay thế giá trị cuối cùng nhận được nếu bộ đếm thời gian giám sát thời hạn nhận của tín hiệu hết hạn.

[SWS\_Com\_00500] ⌈Nếu bộ đếm thời gian giám sát thời hạn nhận của một tín hiệu hoặc nhóm tín hiệu hết hạn và tham số cấu hình ComRxDataTimeoutAction cho tín hiệu hoặc nhóm tín hiệu này bị bỏ qua hoặc được cấu hình là NONE, mô-đun AUTOSAR COM sẽ không thay thế tín hiệu hoặc các tín hiệu của nhóm tín hiệu tương ứng.⌋ (SRS\_Com\_02087)

**Giải thích**: Nếu ComRxDataTimeoutAction là NONE, giá trị tín hiệu hoặc nhóm tín hiệu sẽ không được thay thế khi hết thời gian chờ.

[SWS\_Com\_00513] ⌈Nếu ComRxDataTimeoutAction được đặt thành REPLACE, mô-đun AUTOSAR COM sẽ thay thế các giá trị của tất cả các tín hiệu nhóm bao gồm bằng các giá trị khởi tạo (ComSignalInitValues) khi bộ đếm thời gian giám sát thời hạn nhận của nhóm tín hiệu hết hạn.⌋ (SRS\_Com\_02041)

**Giải thích**: Giá trị khởi tạo (ComSignalInitValues) sẽ thay thế giá trị của tất cả các tín hiệu nhóm khi bộ đếm thời gian giám sát thời hạn nhận của nhóm tín hiệu hết hạn.

[SWS\_Com\_00715] ⌈Khi một I-PDU (lớn) với cấu hình giám sát thời hạn nhận được nhận thành công, mô-đun AUTOSAR COM sẽ đặt lại bộ đếm thời gian giám sát thời hạn nhận cho I-PDU này khi gọi hàm Com\_RxIndication hoặc Com\_TpRxIndication tương ứng.⌋ (SRS\_Com\_02058)

**Giải thích**: Bộ đếm thời gian giám sát thời hạn nhận sẽ được đặt lại khi I-PDU được nhận thành công.

[SWS\_Com\_00716] ⌈Nếu tham số cấu hình ComFirstTimeout cho một tín hiệu hoặc nhóm tín hiệu bị bỏ qua hoặc cấu hình là 0, mô-đun AUTOSAR COM sẽ không giám sát việc nhận tín hiệu hoặc nhóm tín hiệu này từ khi bắt đầu I-PDU tương ứng cho đến khi nhận lần đầu.⌋ (SRS\_Com\_02058)

**Giải thích**: Nếu ComFirstTimeout là 0 hoặc bị bỏ qua, không có giám sát việc nhận tín hiệu hoặc nhóm tín hiệu từ khi bắt đầu I-PDU cho đến khi nhận lần đầu.

[SWS\_Com\_00738] ⌈Cơ chế giám sát thời hạn nhận sẽ không tính đến giá trị của các tín hiệu. Do đó, mô-đun AUTOSAR COM sẽ khởi động lại bộ đếm thời gian giám sát thời hạn nhận ngay cả trong trường hợp nhận được giá trị không hợp lệ.⌋ (SRS\_Com\_02089)

**Giải thích**: Bộ đếm thời gian giám sát thời hạn nhận sẽ khởi động lại ngay cả khi nhận được giá trị không hợp lệ.

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

Giám sát thời hạn nhận giúp đảm bảo rằng các tín hiệu và nhóm tín hiệu được nhận trong khoảng thời gian quy định. Nếu không, các hành động cụ thể sẽ được thực hiện dựa trên cấu hình. Dưới đây là các quy tắc và yêu cầu liên quan đến việc quản lý trạng thái giám sát thời hạn nhận trong mô-đun AUTOSAR COM.

Các quy tắc và yêu cầu liên quan

[SWS\_Com\_00224] ⌈Nếu trạng thái giám sát thời hạn nhận của một I-PDU được thay đổi từ tắt sang bật bằng một cuộc gọi tới Com\_ReceptionDMControl, mô-đun AUTOSAR COM sẽ đặt bộ đếm thời gian giám sát thời hạn nhận cho các tín hiệu và nhóm tín hiệu bao gồm trong I-PDU đó thành giá trị ComFirstTimeout đã cấu hình.⌋ (SRS\_Com\_00192)

**Giải thích**: Khi giám sát thời hạn nhận được bật cho một I-PDU, bộ đếm thời gian giám sát sẽ được đặt lại bằng giá trị ComFirstTimeout đã cấu hình.

[SWS\_Com\_00486] ⌈Mô-đun AUTOSAR COM sẽ âm thầm bỏ qua việc thiết lập giám sát thời hạn nhận của một I-PDU thành bật bằng một cuộc gọi tới Com\_ReceptionDMControl, trong trường hợp giám sát thời hạn nhận đã được bật cho I-PDU này.⌋ (SRS\_Com\_00192)

**Giải thích**: Nếu giám sát thời hạn nhận đã được bật cho I-PDU, bất kỳ yêu cầu bật giám sát nào khác sẽ bị bỏ qua mà không có thông báo lỗi.

* **Bật giám sát thời hạn nhận**: Việc bật giám sát thời hạn nhận có nghĩa là các thông báo hết thời gian chờ sẽ được gửi tới RTE cho các tín hiệu hoặc nhóm tín hiệu có ComTimeoutNotification được cấu hình (ECUC\_Com\_00552).

[SWS\_Com\_00534] ⌈Nếu Com\_ReceptionDMControl được gọi trên một nhóm I-PDU chứa chỉ/ cũng Tx-I-PDUs, mô-đun AUTOSAR COM sẽ âm thầm bỏ qua các Tx-I-PDUs.⌋ (SRS\_Com\_00192)

**Giải thích**: Khi Com\_ReceptionDMControl được gọi trên nhóm I-PDU chứa cả Tx-I-PDUs, các Tx-I-PDUs sẽ bị bỏ qua và chỉ các Rx-I-PDUs sẽ được xem xét.

[SWS\_Com\_00225] ⌈Mô-đun AUTOSAR COM sẽ âm thầm bỏ qua việc thiết lập giám sát thời hạn nhận của một I-PDU thành tắt bằng một cuộc gọi tới Com\_ReceptionDMControl, trong trường hợp giám sát thời hạn nhận đã bị tắt cho I-PDU này.⌋ (SRS\_Com\_00192)

**Giải thích**: Nếu giám sát thời hạn nhận đã bị tắt cho I-PDU, bất kỳ yêu cầu tắt giám sát nào khác sẽ bị bỏ qua mà không có thông báo lỗi.

Các điểm quan trọng khác

* **Không có thông báo lỗi khi giám sát thời hạn nhận bị tắt**: Khi giám sát thời hạn nhận của một I-PDU bị tắt và bộ đếm thời gian hết hạn, sẽ không có thông báo lỗi nào được gửi tới RTE.
* **Không dừng việc nhận I-PDU khi giám sát thời hạn nhận bị tắt**: Việc tắt giám sát thời hạn nhận không dừng việc nhận I-PDU.

##### Transmission Deadline Monitoring

Cơ chế giám sát

Giám sát thời hạn truyền đảm bảo rằng các tín hiệu và nhóm tín hiệu được truyền trong khoảng thời gian quy định. Việc giám sát thời hạn truyền được thực hiện dựa trên I-PDU và thông báo về vi phạm thời hạn truyền được thực hiện cho từng tín hiệu.

Quy tắc và yêu cầu

[SWS\_Com\_00481] ⌈Mô-đun AUTOSAR COM phải thực hiện giám sát thời hạn truyền (nếu được cấu hình) cho tất cả các tín hiệu và nhóm tín hiệu độc lập với thuộc tính truyền và độc lập với các chế độ truyền của I-PDU mà tín hiệu/nhóm tín hiệu thuộc về.⌋ (SRS\_Com\_02037)

**Giải thích**: Giám sát thời hạn truyền sẽ được thực hiện cho tất cả các tín hiệu và nhóm tín hiệu bất kể thuộc tính truyền và chế độ truyền của I-PDU.

[SWS\_Com\_00445] ⌈Nếu các tham số ComTimeout khác nhau của các tín hiệu/nhóm tín hiệu liên quan của một I-PDU được cấu hình, mô-đun AUTOSAR COM sẽ sử dụng giá trị nhỏ nhất làm tham số thời gian chờ cho giám sát thời hạn truyền của I-PDU.⌋ (SRS\_Com\_02037)

**Giải thích**: Giá trị nhỏ nhất của ComTimeout được cấu hình cho các tín hiệu và nhóm tín hiệu trong một I-PDU sẽ được sử dụng để giám sát thời hạn truyền của I-PDU.

Cấu hình và yêu cầu giám sát

* Giám sát thời hạn truyền chỉ nên được cấu hình cho các bus hỗ trợ việc tạo ra xác nhận truyền (transmit confirmations). Nếu không, giám sát thời hạn truyền sẽ luôn thông báo lỗi truyền.

[SWS\_Com\_00696] ⌈Trong trường hợp giám sát thời hạn truyền được cấu hình cho một I-PDU với chế độ truyền NONE duy nhất, mô-đun AUTOSAR COM sẽ bắt đầu bộ đếm thời gian giám sát thời hạn truyền cho I-PDU này khi bắt đầu nhóm I-PDU mà I-PDU thuộc về.⌋ (SRS\_Com\_02037)

**Giải thích**: Nếu một I-PDU chỉ có chế độ truyền NONE, bộ đếm thời gian giám sát thời hạn truyền sẽ bắt đầu khi nhóm I-PDU chứa I-PDU này được khởi động.

[SWS\_Com\_00835] ⌈Trong trường hợp giám sát thời hạn truyền được cấu hình cho một I-PDU với chế độ truyền NONE và một chế độ truyền khác, giám sát thời hạn truyền sẽ bị tắt bất cứ khi nào chế độ truyền NONE hoạt động.⌋ (SRS\_Com\_02037)

**Giải thích**: Nếu một I-PDU có cả chế độ truyền NONE và một chế độ truyền khác, giám sát thời hạn truyền sẽ bị tắt khi chế độ truyền NONE hoạt động.

[SWS\_Com\_00697] ⌈Trong trường hợp giám sát thời hạn truyền được cấu hình cho một I-PDU với chế độ truyền NONE duy nhất, mô-đun AUTOSAR COM sẽ đặt lại bộ đếm thời gian giám sát thời hạn truyền cho I-PDU này sau mỗi xác nhận truyền thông qua Com\_TxConfirmation cho I-PDU này.⌋ (SRS\_Com\_02037)

**Giải thích**: Bộ đếm thời gian giám sát thời hạn truyền sẽ được đặt lại sau mỗi xác nhận truyền cho I-PDU với chế độ truyền NONE.

* **Nhóm tín hiệu**: Chỉ có thể cấu hình giám sát thời hạn truyền cho cả nhóm tín hiệu, không thể cấu hình riêng cho các tín hiệu nhóm. Xem ECUC\_Com\_00345 và ECUC\_Com\_00520.

[SWS\_Com\_00708] ⌈Trong trường hợp bộ đếm thời gian giám sát thời hạn truyền hết hạn cho một I-PDU với ComIPduCancellationSupport được cấu hình là TRUE, mô-đun AUTOSAR COM sẽ gọi PduR\_ComCancelTransmit cho I-PDU đó.⌋ (SRS\_Com\_02107)

**Giải thích**: Nếu bộ đếm thời gian giám sát thời hạn truyền hết hạn và I-PDU có ComIPduCancellationSupport là TRUE, mô-đun AUTOSAR COM sẽ gọi PduR\_ComCancelTransmit cho I-PDU. Gọi này được thực hiện ngoài việc xử lý thời gian chờ thông thường của Com, và sẽ thông báo tới SW-C nếu được cấu hình.

* **Bỏ qua mã trả về**: Mô-đun AUTOSAR COM sẽ bỏ qua mã trả về của PduR\_ComCancelTransmit.

###### Clarification of the OSEK COM specification

Giám sát thời hạn truyền đảm bảo rằng các tín hiệu và nhóm tín hiệu được truyền trong khoảng thời gian quy định. Dưới đây là các quy tắc và yêu cầu liên quan đến giám sát thời hạn truyền trong mô-đun AUTOSAR COM, đặc biệt là trong chế độ truyền MIXED và N-Times.

Cơ chế giám sát thời hạn truyền

[SWS\_Com\_00304] ⌈Nếu truyền không xảy ra, tức là nếu không có xác nhận truyền của I-PDU bởi PduR, thì thời gian chờ sẽ hết hạn và mô-đun AUTOSAR COM sẽ thông báo cho RTE bằng cách gọi hàm ComTimeoutNotification đã cấu hình, xem ECUC\_Com\_00552.⌋ (SRS\_Com\_02037)

**Giải thích**: Nếu không có xác nhận truyền của I-PDU, bộ đếm thời gian giám sát sẽ hết hạn và thông báo sẽ được gửi tới RTE qua hàm ComTimeoutNotification.

* **Hành vi giám sát thời hạn trong chế độ truyền MIXED**: Nếu bộ đếm thời gian giám sát hết hạn, sẽ có một thông báo timeout bất kể lý do. Điều này bao gồm các trường hợp truyền bị hoãn lại do MDT hoặc I-PDU bị lọc ra bởi một cuộc gọi I-PDU.
* **Làm rõ từ [17]**: Trong chế độ truyền DIRECT, bộ đếm thời gian giám sát bắt đầu khi hoàn thành cuộc gọi tới dịch vụ API SendMessage, SendDynamicMessage hoặc SendZeroMessage.
* **Lưu ý**: Bộ đếm thời gian giám sát thời hạn truyền chỉ nên được đặt lại nếu tín hiệu tương ứng có thời gian chờ giám sát thời hạn truyền được cấu hình. Các tín hiệu không có giám sát thời hạn truyền không nên can thiệp vào quá trình giám sát dựa trên I-PDU.

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

Như định nghĩa trong [17], bộ đếm thời gian giám sát phải được bắt đầu khi hoàn thành cuộc gọi tới Com\_SendSignal hoặc Com\_SendSignalGroup nếu giám sát thời hạn truyền được cấu hình cho tín hiệu hoặc nhóm tín hiệu tương ứng.

* **Chế độ truyền DIRECT và MIXED**: Cần đảm bảo rằng tất cả các yêu cầu ComTxModeNumberOfRepetitions có thể được thực hiện trong khoảng thời gian cấu hình.
* **Thông báo sự cố**: Nếu bộ đếm thời gian giám sát hết hạn, RTE sẽ được thông báo về sự cố đó bằng cơ chế thông báo đã cấu hình.

[SWS\_Com\_00308] ⌈Đối với một I-PDU với ComTxModeMode DIRECT và ComTxModeNumberOfRepetitions > 0, mô-đun AUTOSAR COM sẽ hủy bộ đếm thời gian giám sát thời hạn truyền sau khi nhận được xác nhận ComTxModeNumberOfRepetitions cộng một.⌋ (SRS\_Com\_02083)

**Giải thích**: Bộ đếm thời gian giám sát sẽ bị hủy sau khi nhận được xác nhận ComTxModeNumberOfRepetitions cộng một. Việc truyền thành công và xác nhận truyền sẽ được gửi tới RTE.

[SWS\_Com\_00739] ⌈Đối với một I-PDU với ComTxModeMode DIRECT và ComTxModeNumberOfRepetitions > 0, mô-đun AUTOSAR COM sẽ đặt lại bộ đếm thời gian giám sát thời hạn truyền đang chạy nếu một yêu cầu gửi khác cho I-PDU này được khởi tạo.⌋ (SRS\_Com\_02083)

**Giải thích**: Bộ đếm thời gian giám sát sẽ được đặt lại nếu một yêu cầu gửi khác cho I-PDU này được khởi tạo trong khi bộ đếm thời gian giám sát đang chạy.

### Signal Groups - Complex Data Types

Tổng quan

Để hỗ trợ khái niệm các kiểu dữ liệu phức tạp của AUTOSAR, mô-đun AUTOSAR COM cung cấp các nhóm tín hiệu. Các nhóm tín hiệu này giúp truyền và nhận các tín hiệu một cách nhất quán, đảm bảo tính toàn vẹn cho các kiểu dữ liệu phức tạp. Các nhóm tín hiệu có thể được cấu hình tĩnh với các tên biểu tượng riêng biệt.

Cơ chế hoạt động

Mô-đun AUTOSAR COM đạt được sự nhất quán của một nhóm tín hiệu thông qua cơ chế bộ đệm shadow. RTE truy cập các tín hiệu nhóm trong bộ đệm shadow và có thể đồng bộ hóa bộ đệm này với I-PDU thông qua các cuộc gọi Com\_SendSignalGroup hoặc Com\_ReceiveSignalGroup.

#### Initialization

[SWS\_Com\_00484] ⌈Bằng một cuộc gọi tới Com\_Init, mô-đun AUTOSAR COM sẽ khởi tạo bộ đệm shadow của một nhóm tín hiệu ở phía gửi.⌋ (SRS\_BSW\_00101)

**Giải thích**: Bộ đệm shadow của nhóm tín hiệu ở phía gửi sẽ được khởi tạo khi gọi Com\_Init. Điều này đảm bảo rằng các tín hiệu nhóm không được đọc trước khi dữ liệu nhóm tín hiệu được sao chép vào bộ đệm shadow.

#### Transmission

* Nếu Com\_SendSignal hoặc Com\_InvalidateSignal được gọi cho một tín hiệu thuộc về một nhóm tín hiệu, mô-đun AUTOSAR COM chỉ cập nhật bộ đệm shadow của nhóm tín hiệu này.
* **Không cần xử lý I-PDU**: Trừ khi nội dung I-PDU thay đổi, không cần thực hiện đánh giá TMS hoặc bất kỳ xử lý I-PDU nào khác.

[SWS\_Com\_00050] ⌈Nếu Com\_SendSignalGroup được gọi cho nhóm tín hiệu, mô-đun AUTOSAR COM sẽ sao chép bộ đệm shadow một cách nguyên tử vào bộ đệm I-PDU.⌋ (SRS\_Com\_02041)

**Ví dụ**:

|  |
| --- |
| /\* Sao chép a vào bộ đệm shadow \*/  Com\_SendSignal(signal\_a, &a);  /\* Sao chép b vào bộ đệm shadow \*/  Com\_SendSignal(signal\_b, &b);  /\* Sao chép bộ đệm shadow vào I-PDU \*/  Com\_SendSignalGroup(group\_x); |

#### Reception

* Một tín hiệu nhóm có thể được nhận từ bộ đệm shadow bằng cách gọi hàm Com\_ReceiveSignal sau khi dữ liệu nhóm tín hiệu đã được sao chép vào bộ đệm shadow bằng Com\_ReceiveSignalGroup.

[SWS\_Com\_00051] ⌈Nếu Com\_ReceiveSignalGroup được gọi cho một nhóm tín hiệu, mô-đun AUTOSAR COM sẽ sao chép dữ liệu nguyên tử từ bộ đệm I-PDU sang bộ đệm shadow.⌋ (SRS\_Com\_02041)

**Ví dụ**:

|  |
| --- |
| /\* Sao chép I-PDU vào bộ đệm shadow \*/  Com\_ReceiveSignalGroup(group\_x);  /\* Sao chép a từ bộ đệm shadow \*/  Com\_ReceiveSignal(signal\_a, &a);  /\* Sao chép b từ bộ đệm shadow \*/  Com\_ReceiveSignal(signal\_b, &b); |

#### Notifications

Bảng 5 cho thấy tất cả các hàm thông báo chỉ có thể cấu hình cho các nhóm tín hiệu chứ không phải cho các tín hiệu nhóm riêng lẻ. Chương 7.3 định nghĩa logic chức năng của các phương pháp thông báo cho các tín hiệu và nhóm tín hiệu. Các nguyên mẫu cho các hàm thông báo có thể cấu hình được định nghĩa trong Chương 8.6.3.1.

Các hàm thông báo chỉ có thể cấu hình cho các nhóm tín hiệu chứ không phải cho các tín hiệu nhóm riêng lẻ. Bảng dưới đây cho thấy các thuộc tính của một nhóm tín hiệu:

#### Attributes of a Signal Group

Bảng 5 cung cấp cái nhìn tổng quan về các thuộc tính của một nhóm tín hiệu:

Các thuộc tính của Nhóm Tín Hiệu (Attributes of a Signal Group)

| **Attribute** | **Per group signal** | **Per signal group** |
| --- | --- | --- |
| Update-bit | No | Yes, associated on the whole group (see Chapter 7.8) |
| Signal Notification (sender side) | No | Yes |
| Signal Notification (receiver side) | No | Yes |
| Error Notification (sender side) | No | Yes |
| Timeout Notification (sender side) | No | Yes |
| Timeout Notification (receiver side) | No | Yes |
| Invalid Notification (receiver side) | No | Yes |
| Data access (receiver side) | Yes, see SWS\_Com\_00202 | Yes, see SWS\_Com\_00201 |
| Data access (sender side) | Yes, see SWS\_Com\_00199, SWS\_Com\_00288 | Yes, see SWS\_Com\_00200, SWS\_Com\_00557 |
| Data Filtering (receiver side) | Yes, see Chapter 7.2.4 | No |
| Data Filtering (sender side) | No | No |
| TMS on sender side | Each signal, according to TMS selection definition. (see Chapter 7.3.3.2) | No |

**Table 5: Attributes of signal groups**

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

Các điều kiện tiên quyết cho truy cập dựa trên mảng UINT8

[SWS\_Com\_00841] ⌈Truy cập dựa trên mảng UINT8 vào các nhóm tín hiệu chỉ nên được sử dụng nếu các điều kiện tiên quyết sau được áp dụng:

* Chỉ sử dụng các kiểu dữ liệu có kích thước cố định cho dữ liệu hợp phần.
* Các nhóm tín hiệu được ánh xạ byte-aligned với I-PDU.
* Các nhóm tín hiệu mà tất cả các tín hiệu nhóm được ánh xạ liên tiếp vào I-PDU ở phía truyền.⌋ (SRS\_Com\_02112)

**Giải thích**: Truy cập dựa trên mảng UINT8 chỉ nên được sử dụng khi các tín hiệu có kích thước cố định, được ánh xạ byte-aligned với I-PDU, và tất cả các tín hiệu nhóm được ánh xạ liên tiếp vào I-PDU.

[SWS\_Com\_00842] ⌈Trong trường hợp một nhóm tín hiệu có ComSignalGroupArrayAccess (ECUC\_Com\_10003) được định nghĩa, không có bộ đệm shadow nào được cấp phát cho nhóm tín hiệu đó.⌋ (SRS\_Com\_02112)

**Giải thích**: Nếu ComSignalGroupArrayAccess được định nghĩa, không có bộ đệm shadow nào được cấp phát cho nhóm tín hiệu đó.

[SWS\_Com\_00843] ⌈Trong trường hợp một nhóm tín hiệu có ComSignalGroupArrayAccess (ECUC\_Com\_10003) được định nghĩa, việc gọi tới các API sau đây không được hỗ trợ cho nhóm tín hiệu đó:

* Com\_SendSignal và Com\_ReceiveSignal cho bất kỳ tín hiệu nhóm nào của nhóm tín hiệu đó.
* Com\_UpdateShadowSignal [đã lỗi thời] và Com\_ReceiveShadowSignal [đã lỗi thời] cho bất kỳ tín hiệu nhóm nào của nhóm tín hiệu đó.
* Com\_SendSignalGroup và Com\_ReceiveSignalGroup cho nhóm tín hiệu đó.⌋ (SRS\_Com\_02112)

**Giải thích**: Các API như Com\_SendSignal, Com\_ReceiveSignal, Com\_UpdateShadowSignal, Com\_ReceiveShadowSignal, Com\_SendSignalGroup và Com\_ReceiveSignalGroup không được hỗ trợ cho các nhóm tín hiệu có ComSignalGroupArrayAccess.

Chức năng cơ bản (Basic functionality)

Trong trường hợp phương pháp AUTOSAR Transformer được sử dụng, việc tuần tự hóa và xử lý dữ liệu tiếp theo đã được thực hiện trong chuỗi transformer. Việc truy cập dựa trên mảng uint8 vào các nhóm tín hiệu cho phép sử dụng dữ liệu đã tuần tự hóa và đặt nó vào I-PDU mà không cần mô-đun COM tuần tự hóa dữ liệu lại.

[SWS\_Com\_00844] ⌈Vị trí bắt đầu của biểu diễn mảng uint8 của dữ liệu truyền thông sẽ là độ lệch byte của tín hiệu nhóm trong nhóm tín hiệu này với ComBitPosition nhỏ nhất.⌋ (SRS\_Com\_02112)

**Giải thích**: Vị trí bắt đầu của mảng uint8 sẽ là độ lệch byte của tín hiệu nhóm có ComBitPosition nhỏ nhất.

[SWS\_Com\_00845] ⌈Chiều dài của biểu diễn mảng uint8 của dữ liệu truyền thông sẽ là số byte mà tất cả các ComGroupSignals trong ComSignalGroup này chiếm trong ComIPdu.⌋ (SRS\_Com\_02112)

**Giải thích**: Chiều dài của mảng uint8 sẽ là tổng số byte mà tất cả các ComGroupSignals trong ComSignalGroup này chiếm trong ComIPdu.

##### Basic functionality

Vị trí bắt đầu và chiều dài của biểu diễn mảng uint8

[SWS\_Com\_00844] ⌈Vị trí bắt đầu của biểu diễn mảng uint8 của dữ liệu truyền thông sẽ là độ lệch byte của tín hiệu nhóm trong nhóm tín hiệu này với ComBitPosition nhỏ nhất.⌋ (SRS\_Com\_02112)

**Giải thích**: Vị trí bắt đầu của mảng uint8 là độ lệch byte của tín hiệu nhóm có ComBitPosition nhỏ nhất.

[SWS\_Com\_00845] ⌈Chiều dài của biểu diễn mảng uint8 của dữ liệu truyền thông sẽ là số byte mà tất cả các ComGroupSignals trong ComSignalGroup này chiếm trong ComIPdu.⌋ (SRS\_Com\_02112)

**Giải thích**: Chiều dài của mảng uint8 là tổng số byte mà tất cả các ComGroupSignals trong ComSignalGroup chiếm trong ComIPdu.

##### Initialization

[SWS\_Com\_00850] ⌈Bằng một cuộc gọi tới Com\_Init, mô-đun AUTOSAR COM sẽ khởi tạo phần bộ đệm I-PDU tương ứng của một nhóm tín hiệu, có ComSignalGroupArrayAccess (ECUC\_Com\_10003) được cấu hình là true, dựa trên ComSignalInitValue của tất cả các ComGroupSignals bao gồm.⌋ (SRS\_Com\_02112, SRS\_BSW\_00101)

**Giải thích**: Khi gọi Com\_Init, mô-đun AUTOSAR COM sẽ khởi tạo phần bộ đệm I-PDU của nhóm tín hiệu có ComSignalGroupArrayAccess được cấu hình là true dựa trên giá trị khởi tạo ComSignalInitValue.

##### Transmission

[SWS\_Com\_00846] ⌈Khi Com\_SendSignalGroupArray được gọi, một con trỏ dữ liệu tới biểu diễn mảng uint8 của nhóm tín hiệu sẽ được cung cấp. Mô-đun AUTOSAR COM sẽ sao chép dữ liệu vào I-PDU dựa trên vị trí bắt đầu được định nghĩa trong SWS\_Com\_00844 và kích thước được định nghĩa trong SWS\_Com\_00845.⌋ (SRS\_Com\_02112)

**Giải thích**: Khi gọi Com\_SendSignalGroupArray, mô-đun AUTOSAR COM sẽ sao chép dữ liệu từ biểu diễn mảng uint8 vào I-PDU dựa trên vị trí bắt đầu và kích thước đã định nghĩa.

[SWS\_Com\_00847] ⌈Mô-đun AUTOSAR COM sẽ xử lý cuộc gọi Com\_SendSignalGroupArray tương đương với Com\_SendSignalGroup liên quan đến:

* Chương 7.3.3.1.2 Nhóm Tín Hiệu (Thuộc tính Truyền và Chế độ Truyền I-PDU)
* Chương 7.3.3.2 Lựa chọn Chế độ Truyền cho một I-PDU cụ thể
* Chương 7.3.3.3 Luồng Tín Hiệu và Lựa chọn Chế độ Truyền ⌋ (SRS\_Com\_02112)

**Giải thích**: Com\_SendSignalGroupArray sẽ được xử lý tương đương với Com\_SendSignalGroup trong các khía cạnh liên quan đến thuộc tính truyền và chế độ truyền của I-PDU.

* **Lựa chọn Chế độ Truyền (TMS)**: Các bộ lọc được hỗ trợ cho Com\_SendSignalGroupArray bao gồm:
  + ALWAYS
  + NEVER
  + MASKED\_NEW\_EQUALS\_X
  + MASKED\_NEW\_DIFFERS\_X
  + MASKED\_NEW\_DIFFERS\_MASKED\_OLD
  + ONE\_EVERY\_N

##### Reception

[SWS\_Com\_00849] ⌈Khi Com\_ReceiveSignalGroupArray được gọi, một con trỏ dữ liệu tới biểu diễn mảng uint8 của nhóm tín hiệu sẽ được cung cấp. Mô-đun AUTOSAR COM sẽ sao chép dữ liệu vào vị trí con trỏ dữ liệu của I-PDU dựa trên vị trí bắt đầu được định nghĩa trong SWS\_Com\_00844 và kích thước được định nghĩa trong SWS\_Com\_00845.⌋ (SRS\_Com\_02112)

**Giải thích**: Khi gọi Com\_ReceiveSignalGroupArray, dữ liệu sẽ được sao chép từ I-PDU vào biểu diễn mảng uint8 dựa trên vị trí bắt đầu và kích thước đã định nghĩa.

### Large Data Types

Một tín hiệu lớn là tín hiệu quá lớn để có thể vừa trong một L-PDU của giao thức truyền thông cơ bản. Ví dụ, một tín hiệu lớn trên CAN sẽ vượt quá 8 byte hoặc 64 byte đối với CAN FD. Đối với FlexRay, tình huống phức tạp hơn vì một khung có thể chứa nhiều I-PDU. Do đó, một tín hiệu có thể là tín hiệu lớn đối với FlexRay ngay cả khi nó không vượt quá kích thước của một khung FlexRay.

Các tín hiệu lớn trong AUTOSAR yêu cầu cấu hình một I-PDU lớn sẽ được truyền qua giao thức vận chuyển của bus cơ bản. Tuy nhiên, không nhất thiết một I-PDU lớn phải chứa một hoặc nhiều tín hiệu lớn.

Hạn chế kích thước tín hiệu và I-PDU

Sử dụng khái niệm này, độ dài I-PDU và do đó kích thước tín hiệu sẽ bị giới hạn theo giao thức vận chuyển cụ thể của bus, xem ECUC\_Com\_00437. UINT8\_N là kiểu tín hiệu duy nhất hỗ trợ các tín hiệu có kích thước tĩnh lớn hơn 8 byte. Đối với các tín hiệu có độ dài động, xem Chương 7.6.

Giới hạn của mô-đun AUTOSAR COM

Do mô-đun AUTOSAR COM không biết về các thuộc tính của bus cơ bản, nó phải được cấu hình nếu một I-PDU có thể được truyền trong một L-PDU đơn, ví dụ một khung CAN, hoặc nếu nó cần được truyền qua TP, xem ECUC\_Com\_00761.

Mô-đun AUTOSAR COM không thể ngăn chặn hoặc phát hiện nếu các I-PDU quá lớn được gửi tới một giao thức vận chuyển bus cụ thể, vì nó sử dụng giao diện PDU Router độc lập với bus. Tuy nhiên, mô-đun AUTOSAR COM cho phép sử dụng chiều dài tối đa của N-SDU của các TP cơ bản và do đó không giới hạn thêm chiều dài.

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

Mô-đun AUTOSAR COM gửi các I-PDU lớn qua các API PduR chung cho các mô-đun lớp trên sử dụng giao thức vận chuyển. Theo [10], trình tự gọi là:

* PduR\_ComTransmit: để khởi tạo yêu cầu gửi
* Một hoặc nhiều callback tới Com\_CopyTxData: để sao chép dữ liệu
* Callback tới Com\_TpTxConfirmation: để xác nhận truyền

[SWS\_Com\_00662] ⌈Sau khi khởi tạo thành công việc truyền một I-PDU lớn qua PduR\_ComTransmit, mô-đun AUTOSAR COM không được thay đổi dữ liệu trong bộ đệm truyền nội bộ của I-PDU này trừ khi được thông báo về việc truyền thành công qua Com\_TpTxConfirmation hoặc được thông báo về một lỗi chỉ ra rằng truyền đã bị hủy.⌋ (SRS\_Com\_02095)

**Giải thích**: Sau khi khởi tạo thành công việc truyền, mô-đun AUTOSAR COM không được thay đổi dữ liệu trong bộ đệm truyền nội bộ cho đến khi nhận được xác nhận truyền thành công hoặc thông báo lỗi.

* **Cấu hình I-PDU tĩnh**: Các I-PDU trong mô-đun AUTOSAR COM được cấu hình tĩnh. Do đó, toàn bộ bộ nhớ của I-PDU, kể cả I-PDU rất lớn, sẽ được cấp phát. Cách tiếp cận này được thực hiện để giảm độ phức tạp của việc xử lý các kiểu dữ liệu lớn.
* **Hỗ trợ giao thức vận chuyển nhiều người dùng**: Nếu giao thức vận chuyển được sử dụng cho chẩn đoán hoặc các dịch vụ khác, nó phải hỗ trợ nhiều người dùng. Điều này ít nhất được hỗ trợ bởi CAN TP và FlexRay TP.

[SWS\_Com\_00863] ⌈Trong trường hợp một I-PDU lớn đang được truyền và cùng I-PDU được kích hoạt để truyền lại, mô-đun AUTOSAR COM sẽ bỏ qua yêu cầu gửi thêm và báo cáo lỗi thời gian chạy COM\_E\_SKIPPED\_TRANSMISSION.⌋ (SRS\_Com\_02095)

**Giải thích**: Nếu một I-PDU lớn đang được truyền và cùng I-PDU được yêu cầu truyền lại, mô-đun AUTOSAR COM sẽ bỏ qua yêu cầu truyền thêm và báo cáo lỗi COM\_E\_SKIPPED\_TRANSMISSION. Điều này có thể xảy ra trong trường hợp một I-PDU lớn được gửi định kỳ và truyền bị trì hoãn qua chu kỳ tiếp theo.

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

Mô-đun AUTOSAR COM nhận các I-PDU lớn qua các API chung của PDU Router dành cho các mô-đun lớp trên sử dụng giao thức vận chuyển. Trình tự gọi như sau:

1. **Bắt đầu tiếp nhận**:
   * Com\_StartOfReception: để khởi tạo yêu cầu RX.
2. **Sao chép dữ liệu nhận**:
   * Gọi một hoặc nhiều callback tới Com\_CopyRxData: sao chép dữ liệu nhận vào bộ đệm nhận nội bộ của mô-đun AUTOSAR COM.
3. **Kết thúc tiếp nhận**:
   * Com\_TpRxIndication: chỉ báo kết thúc quá trình tiếp nhận đúng hoặc sai.

Trình tự gọi API khi nhận I-PDU lớn

Theo [10], trình tự gọi như sau:

* **Com\_StartOfReception**: Khởi tạo yêu cầu RX.
* **Com\_CopyRxData**: Sao chép dữ liệu nhận vào bộ đệm nhận nội bộ của mô-đun AUTOSAR COM.
* **Com\_TpRxIndication**: Chỉ báo kết thúc quá trình tiếp nhận, có thể đúng hoặc sai.

Xử lý tín hiệu không hợp lệ

[SWS\_Com\_00838] ⌈Trong trường hợp Com\_TpRxIndication được gọi với giá trị E\_NOT\_OK cho tham số kết quả, mô-đun AUTOSAR COM sẽ xử lý tất cả các tín hiệu của I-PDU lớn đã nhận giống như các tín hiệu không hợp lệ. Tức là, mô-đun AUTOSAR COM sẽ thực hiện tất cả các ComDataInvalidActions đã cấu hình cho tất cả các tín hiệu và nhóm tín hiệu bao gồm.⌋ (SRS\_Com\_02095)

**Giải thích**: Nếu Com\_TpRxIndication nhận được giá trị E\_NOT\_OK, tất cả các tín hiệu trong I-PDU lớn được xử lý như các tín hiệu không hợp lệ. Mô-đun AUTOSAR COM sẽ thực hiện các hành động ComDataInvalidActions đã cấu hình cho tất cả các tín hiệu và nhóm tín hiệu trong I-PDU.

### Dynamic Length Signals

Định nghĩa tín hiệu có độ dài động

Một tín hiệu có độ dài động là tín hiệu mà độ dài có thể thay đổi trong thời gian chạy. Độ dài tối đa của tín hiệu này phải được chỉ định tại thời điểm cấu hình. Các tín hiệu có độ dài động phải được truyền trong I-PDU có độ dài động.

Hạn chế loại tín hiệu

[SWS\_Com\_00753] ⌈Đối với các tín hiệu có độ dài động, mô-đun AUTOSAR COM chỉ hỗ trợ loại ComSignalType UINT8\_DYN.⌋ (SRS\_Com\_02094, SRS\_Com\_02098)

**Giải thích**: Mô-đun AUTOSAR COM chỉ hỗ trợ tín hiệu có độ dài động dưới dạng mảng UINT8. Điều này giới hạn tín hiệu có độ dài động phải được đặt tại các biên byte.

Đặt tín hiệu có độ dài động

Tất cả các tín hiệu khác, bao gồm cả tín hiệu lớn, phải được đóng gói phía trước tín hiệu có độ dài động. Điều này loại bỏ việc phải mã hóa độ dài của tín hiệu có độ dài động trong I-PDU. Ở phía nhận, độ dài của tín hiệu có độ dài động có thể được tính từ độ dài của I-PDU nhận được.

Bit cập nhật cho tín hiệu có độ dài động

Được phép cấu hình một bit cập nhật cho tín hiệu có độ dài động. Trong trường hợp này, bit cập nhật phải được đặt ở phía trước tín hiệu có độ dài động.

Kết hợp với nhóm tín hiệu và các hạn chế

Không có giới hạn nào trong mô-đun AUTOSAR COM ngăn cản việc sử dụng nhóm tín hiệu kết hợp với các tín hiệu lớn hoặc tín hiệu có độ dài động. Các tín hiệu có độ dài động không bị giới hạn chỉ gửi qua giao thức truyền tải, nhưng độ dài tối đa của I-PDU chứa tín hiệu có độ dài động phải tuân theo các hạn chế cụ thể của bus.

Sử dụng Multi PDU Mapping của IpduM

Trong trường hợp các I-PDU có tín hiệu có độ dài động được gửi qua CAN FD, nên sử dụng tính năng Multi PDU Mapping của IpduM để tránh diễn giải sai độ dài nhận được do các độ dài dữ liệu rời rạc của các khung CAN FD.

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

Để gửi một tín hiệu có độ dài động, RTE cần chỉ định độ dài thực tế khi gọi hàm gửi. API tương ứng là Com\_SendDynSignal.

[SWS\_Com\_00757] ⌈Khi gọi Com\_SendDynSignal, mô-đun AUTOSAR COM phải đặt độ dài của I-PDU có độ dài động tương ứng thành độ dài nhỏ nhất bao gồm tín hiệu có độ dài động. Độ dài của tín hiệu động sẽ bằng giá trị của tham số đầu vào “Length” được truyền trong API Com\_SendDynSignal.⌋ (SRS\_Com\_02093)

**Giải thích**: Khi gọi Com\_SendDynSignal, mô-đun AUTOSAR COM sẽ điều chỉnh độ dài của I-PDU động sao cho vừa đủ để chứa tín hiệu động và độ dài này sẽ bằng với giá trị “Length” truyền vào trong API.

Mô-đun AUTOSAR COM có thể gửi các I-PDUs có độ dài động qua các I-PDUs bình thường hoặc lớn. Trong bất kỳ trường hợp nào, nó cần truyền độ dài của I-PDU động đến các lớp dưới.

[SWS\_Com\_00832] ⌈Khi mô-đun AUTOSAR COM gọi PduR\_ComTransmit cho một I-PDU có độ dài động, SduLength của PduInfoType sẽ được đặt theo lần cập nhật cuối cùng (hoặc giá trị khởi tạo nếu chưa được gửi) của tín hiệu có độ dài động bao gồm.⌋ (SRS\_Com\_02093)

**Giải thích**: Khi mô-đun AUTOSAR COM gọi PduR\_ComTransmit cho một I-PDU động, SduLength sẽ được đặt theo độ dài cập nhật cuối cùng của tín hiệu động hoặc giá trị khởi tạo nếu chưa được gửi.

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

Để nhận một tín hiệu có độ dài động, RTE cần được thông báo về kích thước thực tế của tín hiệu.

[SWS\_Com\_00758] ⌈Khi nhận một I-PDU có độ dài động, mô-đun AUTOSAR COM sẽ tính toán độ dài của tín hiệu có độ dài động chứa bên trong, bằng cách:

|  |
| --- |
| <dynamic signal length in bytes> = <received I-PDU length in bytes> - <ComBitPosition of dynamic length signal> / 8 |

**Giải thích**: Khi nhận một I-PDU có độ dài động, mô-đun AUTOSAR COM sẽ tính toán độ dài của tín hiệu động dựa trên độ dài I-PDU nhận được và vị trí bit của tín hiệu động.

Com\_ReceiveDynSignal trả về độ dài của tín hiệu có độ dài động, được tính như đã định nghĩa bởi SWS\_Com\_00758.

Vì độ dài của tín hiệu có độ dài động được suy ra từ độ dài nhận được của I-PDU, mô-đun AUTOSAR COM không hỗ trợ sự kết hợp giữa các I-PDUs có độ dài động và I-PDUs với metadata, xem SWS\_Com\_00817.

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

Giao diện giữa mô-đun AUTOSAR COM và PDU Router

Tổng quan

Mô-đun AUTOSAR COM giao tiếp với các lớp dưới thông qua PDU Router. Mô-đun AUTOSAR COM sử dụng PDU Router trong hai chế độ khác nhau, tùy thuộc vào loại I-PDU:

1. **I-PDU thông thường (NORMAL)**: Không sử dụng giao thức vận chuyển (TP).
2. **I-PDU sử dụng TP**: Sử dụng giao thức vận chuyển bus-specific.

Cấu hình loại I-PDU được thực hiện thông qua tham số cấu hình ComIPduType.

Gửi I-PDUs

[SWS\_Com\_00138] ⌈Mô-đun AUTOSAR COM sẽ gửi I-PDUs bằng cách gọi hàm PduR\_ComTransmit.⌋ (SRS\_Com\_02037)

**Giải thích**: Để gửi I-PDUs, mô-đun AUTOSAR COM sẽ sử dụng hàm PduR\_ComTransmit.

I-PDUs thông thường

[SWS\_Com\_00759] ⌈Nếu mô-đun AUTOSAR COM gửi một I-PDU với ComIPduType được cấu hình là NORMAL, mô-đun AUTOSAR COM sẽ mong đợi được gọi bởi PDU Router sử dụng các giao diện của COM cho các mô-đun không sử dụng TP. Xem cột normal I-PDUs trong Bảng 6.⌋ (SRS\_Com\_02096)

**Giải thích**: Đối với I-PDUs thông thường, mô-đun AUTOSAR COM mong đợi PDU Router sử dụng giao diện COM cho các mô-đun không sử dụng TP.

I-PDUs sử dụng TP

[SWS\_Com\_00760] ⌈Nếu mô-đun AUTOSAR COM gửi một I-PDU với ComIPduType được cấu hình là TP, mô-đun AUTOSAR COM sẽ mong đợi được gọi bởi PDU Router sử dụng các giao diện của COM cho các mô-đun sử dụng TP. Xem cột TP I-PDUs trong Bảng 6.⌋ (SRS\_Com\_02096)

**Giải thích**: Đối với I-PDUs sử dụng TP, mô-đun AUTOSAR COM mong đợi PDU Router sử dụng giao diện COM cho các mô-đun sử dụng TP.

Bảng 6: Giao diện API giữa mô-đun AUTOSAR COM và PDU Router

|  |  |  |
| --- | --- | --- |
| ***API*** | ***normal I-PDUs*** | ***TP I-PDUs*** |
| PduR\_ComTransmit | **X** | **X** |
| Com\_TxConfirmation | **X** | **-** |
| Com\_RxIndication | **X** | **-** |
| Com\_TriggerTransmit | **X** | **-** |
| Com\_TpTxConfirmation | **-** | **X** |
| Com\_TpRxIndication | **-** | **X** |
| Com\_StartOfReception | **-** | **X** |
| Com\_CopyRxData | **-** | **X** |
| Com\_CopyTxData | **-** | **X** |

Bảng 6 cung cấp cái nhìn tổng quan về các API giữa mô-đun AUTOSAR COM và PDU Router được sử dụng cho một I-PDU cụ thể với giá trị cấu hình ComIPduType.

Sơ đồ tương tác

Sơ đồ tương tác trong Chương 9.1 minh họa sự tương tác giữa mô-đun PDU Router và mô-đun AUTOSAR COM cho các I-PDUs thông thường.

Mô tả API

Mô tả chi tiết của các API được chỉ định trong Chương 8.

**Table 6: API to PDU Router with respect to ComIPduType**

### Update-Bits

Tổng quan về bit cập nhật

Mô-đun AUTOSAR COM hỗ trợ các bit cập nhật để cho phép người nhận xác định liệu người gửi có cập nhật dữ liệu trong tín hiệu hoặc nhóm tín hiệu trước khi gửi hay không. Các bit cập nhật chỉ ra liệu RTE ở phía người gửi có cập nhật giá trị tín hiệu trước khi I-PDU chứa tín hiệu đó được truyền đến PDU Router hay không.

**Giới hạn và cấu hình**

* Các bit cập nhật không được phép nếu chế độ truyền DIRECT được sử dụng với ComTxModeNumberOfRepetitions lớn hơn hoặc bằng 1 (xem SWS\_Com\_00310).
* Tối đa một bit cập nhật riêng lẻ có thể được thêm cho mỗi tín hiệu và/hoặc nhóm tín hiệu.
* Vị trí của bit cập nhật có thể được cấu hình bởi tham số cấu hình ComUpdateBitPosition (ECUC\_Com\_00257).
* ComUpdateBitPosition được bao gồm trong container cấu hình ComSignal. Do đó, tín hiệu/nhóm tín hiệu và bit cập nhật tương ứng luôn là một phần của cùng một I-PDU.

**Xử lý bit cập nhật**

[SWS\_Com\_00055] ⌈Mô-đun AUTOSAR COM sẽ xử lý bit cập nhật chỉ bên trong và không phải là một phần của tín hiệu hoặc nhóm tín hiệu.⌋ (SRS\_Com\_02030)

**Giải thích**: Các bit cập nhật không trực tiếp hiển thị hoặc truy cập được bởi các thành phần phần mềm AUTOSAR.

[SWS\_Com\_00059] ⌈Mô-đun AUTOSAR COM sẽ giải thích bit cập nhật như được định nghĩa trong Bảng 7.⌋ (SRS\_BSW\_00101, SRS\_Com\_02030)

**Giải thích**: Mô-đun AUTOSAR COM sẽ giải thích các giá trị của bit cập nhật như sau:

* **0**: cleared/ dữ liệu chưa được cập nhật.
* **1**: set/ dữ liệu đã được cập nhật.

| **Update-BIT** | **Mô tả** |
| --- | --- |
| 0 | cleared/ dữ liệu chưa cập nhật |
| 1 | set/ dữ liệu đã cập nhật |

**Table 7 update-bit interpretation**

#### Sender Side

**Khởi tạo bit cập nhật**

Việc khởi tạo các bit cập nhật được định nghĩa bởi SWS\_Com\_00117.

[SWS\_Com\_00061] ⌈Nếu RTE cập nhật giá trị của một tín hiệu bằng cách gọi Com\_SendSignal, mô-đun AUTOSAR COM sẽ đặt bit cập nhật của tín hiệu này.⌋ (SRS\_Com\_02030)

**Giải thích**: Khi RTE cập nhật giá trị của một tín hiệu thông qua Com\_SendSignal, mô-đun AUTOSAR COM sẽ đặt bit cập nhật tương ứng.

[SWS\_Com\_00801] ⌈Nếu RTE cập nhật một nhóm tín hiệu bằng cách gọi Com\_SendSignalGroup, mô-đun AUTOSAR COM sẽ đặt bit cập nhật của nhóm tín hiệu này.⌋ (SRS\_Com\_02030)

**Giải thích**: Khi RTE cập nhật giá trị của một nhóm tín hiệu thông qua Com\_SendSignalGroup, mô-đun AUTOSAR COM sẽ đặt bit cập nhật tương ứng của nhóm tín hiệu này.

**Xóa bit cập nhật**

[SWS\_Com\_00062] ⌈Nếu tham số ComTxIPduClearUpdateBit của một I-PDU được cấu hình để Truyền (Transmit), mô-đun AUTOSAR COM sẽ xóa tất cả các bit cập nhật của tất cả các tín hiệu và nhóm tín hiệu chứa trong I-PDU này sau khi I-PDU này được gửi qua PduR\_ComTransmit và PduR\_ComTransmit trả về E\_OK.⌋ (SRS\_Com\_02030)

**Giải thích**: Khi I-PDU được cấu hình để xóa bit cập nhật sau khi Truyền, mô-đun AUTOSAR COM sẽ xóa các bit cập nhật sau khi I-PDU được gửi thành công qua PduR\_ComTransmit.

[SWS\_Com\_00577] ⌈Nếu tham số ComTxIPduClearUpdateBit của một I-PDU được cấu hình để Xác nhận (Confirmation), mô-đun AUTOSAR COM sẽ xóa tất cả các bit cập nhật của tất cả các tín hiệu và nhóm tín hiệu chứa trong I-PDU này sau khi I-PDU này được gửi qua PduR\_ComTransmit, PduR\_ComTransmit trả về E\_OK và I-PDU được xác nhận thành công.⌋ (SRS\_Com\_02030)

**Giải thích**: Khi I-PDU được cấu hình để xóa bit cập nhật sau khi Xác nhận, mô-đun AUTOSAR COM sẽ xóa các bit cập nhật sau khi I-PDU được gửi thành công và xác nhận thành công.

[SWS\_Com\_00578] ⌈Nếu tham số ComTxIPduClearUpdateBit của một I-PDU được cấu hình để Kích hoạt Truyền (TriggerTransmit), mô-đun AUTOSAR COM sẽ xóa tất cả các bit cập nhật của tất cả các tín hiệu và nhóm tín hiệu chứa trong I-PDU này sau khi nội dung của I-PDU này được yêu cầu thành công bởi Com\_TriggerTransmit.⌋ (SRS\_Com\_02030)

**Giải thích**: Khi I-PDU được cấu hình để xóa bit cập nhật sau khi Kích hoạt Truyền, mô-đun AUTOSAR COM sẽ xóa các bit cập nhật sau khi nội dung của I-PDU được yêu cầu thành công bởi Com\_TriggerTransmit.

#### Receiver Side

**Quy định về xử lý tín hiệu và nhóm tín hiệu có bit cập nhật**

[SWS\_Com\_00324] ⌈Nếu một tín hiệu hoặc nhóm tín hiệu có kèm theo bit cập nhật được nhận, mô-đun AUTOSAR COM sẽ chỉ xử lý tín hiệu này, ví dụ như lọc, thông báo, gateway dựa trên tín hiệu, hoán đổi byte, nếu bit cập nhật của tín hiệu được đặt.⌋ (SRS\_Com\_02030)

**Giải thích**: Khi một tín hiệu hoặc nhóm tín hiệu có kèm theo bit cập nhật được nhận và bit cập nhật được đặt, mô-đun AUTOSAR COM sẽ xử lý tín hiệu này theo các quy trình như lọc, thông báo, gateway dựa trên tín hiệu, hoặc hoán đổi byte.

[SWS\_Com\_00802] ⌈Nếu một tín hiệu hoặc nhóm tín hiệu có kèm theo bit cập nhật được nhận và bit cập nhật không được đặt, mô-đun AUTOSAR COM sẽ loại bỏ tín hiệu này.⌋ (SRS\_Com\_02030)

**Giải thích**: Khi một tín hiệu hoặc nhóm tín hiệu có kèm theo bit cập nhật được nhận nhưng bit cập nhật không được đặt, mô-đun AUTOSAR COM sẽ loại bỏ tín hiệu này. Tín hiệu này sẽ không được chuyển qua gateway tín hiệu và sẽ bị loại bỏ.

**Hành vi khi đọc tín hiệu với bit cập nhật đã bị xóa**

Nếu RTE đọc một tín hiệu với bit cập nhật đã bị xóa, giá trị khởi tạo hoặc giá trị nhận được lần cuối sẽ được trả về.

[SWS\_Com\_00067] ⌈Mô-đun AUTOSAR COM sẽ hiểu một tín hiệu/nhóm tín hiệu là đã được cập nhật nếu tín hiệu có kèm theo bit cập nhật và giá trị của bit cập nhật được đặt.⌋ (SRS\_Com\_02030)

**Giải thích**: Mô-đun AUTOSAR COM sẽ coi một tín hiệu hoặc nhóm tín hiệu là đã được cập nhật nếu tín hiệu đó có kèm theo bit cập nhật và giá trị của bit cập nhật được đặt.

**Hành vi giám sát thời hạn nhận**

Để biết hành vi giám sát thời hạn nhận trên các tín hiệu có bit cập nhật, xem Chương 7.3.6.1.

### Data Sequence Control

Mô-đun AUTOSAR COM cung cấp các cơ chế điều khiển chuỗi dữ liệu dưới dạng bộ đếm I-PDU. Các I-PDU ngoài chuỗi, tức là các I-PDU lặp lại hoặc I-PDU nhận được sau khi các I-PDU bị thiếu, sẽ được phát hiện và loại bỏ. Bộ đếm I-PDU là tùy chọn và có thể được cấu hình cho từng I-PDU. Để biết container cấu hình tương ứng, xem ECUC\_Com\_00592.

#### Sender Side (Phía gửi)

Yêu Cầu [SWS\_Com\_00687]

* **Yêu cầu**: Đối với tất cả các I-PDU có hướng ComIPduDirection được cấu hình là SEND và có một ComIPduCounter được cấu hình, mô-đun AUTOSAR COM sẽ đặt bộ đếm I-PDU về 0 trong Com\_Init và trong Com\_IpduGroupControl nếu tham số Initialize là true.
* **Giải thích**: Nếu một I-PDU được cấu hình để gửi dữ liệu (SEND) và có bộ đếm I-PDU (ComIPduCounter) được thiết lập, khi hệ thống khởi động (Com\_Init) hoặc khi gọi hàm Com\_IpduGroupControl với tham số Initialize là true, bộ đếm I-PDU sẽ được đặt lại về 0. Điều này có nghĩa là khi hệ thống khởi động hoặc khi khởi tạo lại, bộ đếm sẽ bắt đầu từ đầu.

Yêu Cầu [SWS\_Com\_00688]

* **Yêu cầu**: Mô-đun AUTOSAR COM sẽ tăng bộ đếm I-PDU của một I-PDU lên một, với điều kiện là bộ đếm vòng, ngay trước khi I-PDU được gửi bởi PduR\_ComTransmit. Trong trường hợp giá trị trả về của PduR\_ComTransmit không bằng E\_OK, việc cập nhật bộ đếm I-PDU sẽ bị thu hồi (bằng cách giảm bộ đếm I-PDU xuống một).
* **Giải thích**: Khi một I-PDU sắp được gửi đi (qua hàm PduR\_ComTransmit), mô-đun AUTOSAR COM sẽ tăng giá trị bộ đếm I-PDU lên 1. Nếu quá trình gửi này không thành công (hàm PduR\_ComTransmit trả về giá trị khác E\_OK), mô-đun sẽ hủy việc tăng bộ đếm bằng cách giảm giá trị bộ đếm xuống 1. Điều này đảm bảo rằng bộ đếm chỉ tăng khi việc gửi thành công.

Lưu ý về Com\_TriggerTransmit

* **Lưu ý**: Nếu Com\_TriggerTransmit lấy một I-PDU chứa bộ đếm I-PDU, sẽ không có sự tăng lên nào xảy ra. Do đó, cần phải cẩn thận nếu sử dụng bộ đếm I-PDU kết hợp với Com\_TriggerTransmit. Nếu Com\_TriggerTransmit được gọi nhiều lần sau PduR\_ComTransmit tương ứng, tất cả các I-PDU sẽ chứa cùng một giá trị bộ đếm. Do đó, việc truyền TriggerTransmit trì hoãn có thể hoạt động, nhưng một trường hợp sử dụng TriggerTransmit chỉ do lớp dưới điều khiển sẽ không hoạt động kết hợp với bộ đếm I-PDU.
* **Giải thích**: Khi sử dụng hàm Com\_TriggerTransmit với I-PDU có bộ đếm, bộ đếm sẽ không tăng. Nếu Com\_TriggerTransmit được gọi nhiều lần sau khi PduR\_ComTransmit được gọi, tất cả các I-PDU sẽ có cùng giá trị bộ đếm. Điều này có thể gây ra vấn đề nếu sử dụng TriggerTransmit để điều khiển việc truyền từ lớp dưới. Chỉ khi truyền TriggerTransmit bị trì hoãn mới có thể hoạt động đúng cách trong trường hợp này.

#### Receiver Side - (Phía nhận)

Yêu Cầu [SWS\_Com\_00587]

* **Yêu cầu**: Đối với tất cả các I-PDU có hướng ComIPduDirection được cấu hình là RECEIVE và có một ComIPduCounter được cấu hình, mô-đun AUTOSAR COM sẽ chấp nhận bất kỳ I-PDU nào đến, bất kể giá trị của bộ đếm I-PDU, sau khi I-PDU được khởi tạo bởi Com\_Init hoặc được khởi tạo lại bởi Com\_IpduGroupControl với tham số Initialize được đặt là true.
* **Giải thích**: Nếu một I-PDU được cấu hình để nhận dữ liệu (RECEIVE) và có bộ đếm I-PDU (ComIPduCounter) được thiết lập, sau khi hệ thống khởi động (Com\_Init) hoặc khi gọi hàm Com\_IpduGroupControl với tham số Initialize là true, mô-đun AUTOSAR COM sẽ chấp nhận bất kỳ giá trị nào của bộ đếm I-PDU trong I-PDU nhận được. Điều này có nghĩa là sau khi khởi động hoặc khởi tạo lại, mọi giá trị của bộ đếm I-PDU đều được chấp nhận.

Yêu Cầu [SWS\_Com\_00588]

* **Yêu cầu**: Khi nhận một I-PDU chứa bộ đếm I-PDU, mô-đun AUTOSAR COM sẽ đặt giá trị mong đợi tiếp theo thành giá trị tiếp theo sau giá trị nhận được, với điều kiện bộ đếm vòng.
* **Giải thích**: Khi mô-đun AUTOSAR COM nhận được một I-PDU có bộ đếm I-PDU, nó sẽ thiết lập giá trị mong đợi tiếp theo thành giá trị kế tiếp sau giá trị bộ đếm vừa nhận, theo quy tắc bộ đếm vòng. Ví dụ, nếu bộ đếm I-PDU có 4 bit (các giá trị từ 0 đến 15), và giá trị nhận được là 15, giá trị mong đợi tiếp theo sẽ là 0.

Yêu Cầu [SWS\_Com\_00590]

* **Yêu cầu**: Khi nhận một I-PDU chứa bộ đếm I-PDU, mô-đun AUTOSAR COM sẽ loại bỏ I-PDU, sau khi đặt giá trị mong đợi tiếp theo như được định nghĩa trong SWS\_Com\_00588, nếu không có sự sao chép I-PDU nào cho I-PDU này được cấu hình và giá trị bộ đếm I-PDU nhận được < giá trị bộ đếm I-PDU mong đợi HOẶC giá trị bộ đếm I-PDU nhận được > giá trị bộ đếm I-PDU mong đợi với điều kiện bộ đếm vòng.
* **Giải thích**: Khi nhận được một I-PDU có bộ đếm I-PDU, nếu giá trị bộ đếm nhận được nhỏ hơn hoặc lớn hơn giá trị mong đợi và không có bản sao của I-PDU được cấu hình, mô-đun AUTOSAR COM sẽ loại bỏ I-PDU đó. Tuy nhiên, trước khi loại bỏ, nó sẽ thiết lập giá trị mong đợi tiếp theo như đã định nghĩa ở yêu cầu [SWS\_Com\_00588].

Yêu Cầu [SWS\_Com\_00727]

* **Yêu cầu**: Trong trường hợp phát hiện sự không khớp giữa giá trị bộ đếm I-PDU mong đợi và giá trị bộ đếm I-PDU nhận được, mô-đun AUTOSAR COM sẽ thông báo sự không khớp này thông qua chức năng thông báo ComIPduCounterErrorNotification (ECUC\_Com\_00003) được cấu hình.
* **Giải thích**: Nếu mô-đun AUTOSAR COM phát hiện giá trị bộ đếm I-PDU nhận được không khớp với giá trị bộ đếm I-PDU mong đợi, nó sẽ kích hoạt chức năng thông báo lỗi ComIPduCounterErrorNotification đã được cấu hình. Thông báo lỗi này sẽ được gửi đi bất kể ngưỡng ComIPduCounterThreshold được cấu hình là gì.

Ví Dụ về Xử Lý Bộ Đếm I-PDU

Trong ví dụ dưới đây, bộ đếm I-PDU rộng 4 bit (các giá trị từ 0-15) và ComIPduCounterThreshold được cấu hình là 1:

| **Expected I-PDU counter** | **Received counter value** | **Action** |
| --- | --- | --- |
| 6 | 6 | normal reception |
| 6 | 7 | normal reception |
| 6 | 8 | discarded |
| 6 | 1 | discarded |
| 15 | 15 | normal reception |
| 15 | 0 | normal reception |
| 15 | 1 | discarded |

* **Giải thích**: Trong bảng này, nếu giá trị bộ đếm I-PDU nhận được nằm ngoài phạm vi mong đợi (giá trị mong đợi ± 1, trong trường hợp này là 1), I-PDU sẽ bị loại bỏ (discarded). Nếu giá trị nhận được nằm trong phạm vi mong đợi, I-PDU sẽ được chấp nhận và xử lý bình thường (normal reception).

**Table 8 Example I-PDU counter handling**

### Communication Protection - Bảo vệ Truyền thông

**I-PDU liên quan đến an toàn có thể cần được sao chép trong nhiều L-PDU để ngăn chặn sự hỏng hóc và mất dữ liệu. Trong AUTOSAR, việc sao chép I-PDU được thực hiện bởi PDU Router. Việc so sánh và bầu chọn của các I-PDU sao chép được thực hiện trong AUTOSAR COM. Vì các I-PDU sao chép không có checksum hoặc chữ ký để so sánh, cần phải lưu trữ và so sánh toàn bộ I-PDU tại phía nhận.**

Giải thích: Các I-PDU quan trọng về an toàn có thể cần phải được sao chép trong nhiều L-PDU để tránh mất mát dữ liệu hoặc lỗi. Trong hệ thống AUTOSAR, việc sao chép này được thực hiện bởi PDU Router, và việc so sánh và bầu chọn các bản sao của I-PDU sẽ được thực hiện bởi mô-đun AUTOSAR COM. Vì các I-PDU sao chép không có mã kiểm tra (checksum) hoặc chữ ký (signature), toàn bộ I-PDU sẽ được lưu trữ và so sánh tại phía nhận.

**Mô-đun AUTOSAR COM cho phép cấu hình hai I-PDU sao chép bổ sung thông qua ComIPduReplicaRef (ECUC\_Com\_00601).**

Giải thích: Mô-đun AUTOSAR COM có thể cấu hình thêm hai I-PDU sao chép thông qua tham chiếu cấu hình ComIPduReplicaRef.

**Số lượng I-PDU nhận được thành công và giống nhau cần thiết cho một quá trình nhận hợp lệ có thể được cấu hình qua ComIPduReplicationQuorum (ECUC\_Com\_00600) từ 1 đến 3.**

Giải thích: Số lượng I-PDU giống nhau và được nhận thành công để quá trình nhận được coi là hợp lệ có thể cấu hình từ 1 đến 3 thông qua ComIPduReplicationQuorum.

**Mô-đun PDU Router không hỗ trợ việc định tuyến 1**

**của các I-PDU được gửi hoặc nhận qua mô-đun TP và yêu cầu nhiều khung để truyền. Do đó, ngăn xếp AUTOSAR COM hiện tại không hỗ trợ sao chép các I-PDU lớn hoặc động.**

* Giải thích: PDU Router không hỗ trợ định tuyến 1

cho các I-PDU được gửi hoặc nhận qua giao thức truyền tải (TP) và yêu cầu nhiều khung để truyền, do đó hiện tại hệ thống AUTOSAR COM không hỗ trợ sao chép các I-PDU lớn hoặc có độ dài động.

#### Sender Side

**Vì việc sao chép được thực hiện bởi PduR, không có xử lý đặc biệt nào trên phía người gửi đối với AUTOSAR COM. Tuy nhiên, cần đảm bảo rằng các I-PDU sao chép có cấu hình bộ đếm I-PDU, xem SWS\_Com\_00834. Điều này cần thiết vì việc bầu chọn tại phía nhận dựa trên bộ đếm I-PDU.**

* Giải thích: Do việc sao chép được thực hiện bởi PduR, phía người gửi của mô-đun AUTOSAR COM không cần xử lý đặc biệt nào. Tuy nhiên, cần đảm bảo rằng các I-PDU sao chép có cấu hình bộ đếm I-PDU vì việc bầu chọn tại phía nhận dựa trên bộ đếm này.

#### Receiver Side

**Tại phía nhận, mô-đun AUTOSAR COM thực hiện việc bầu chọn các I-PDU sao chép. Mô-đun AUTOSAR COM chỉ chuyển các tín hiệu của các I-PDU đã được xác nhận thành công đến RTE. Như đã nói ở trên, các I-PDU sao chép cần phải có bộ đếm I-PDU được cấu hình. Tuy nhiên, ComIPduCounterThreshold phải được cấu hình là 0, xem ECUC\_Com\_00595. Không có trường hợp sử dụng kết hợp hai tính năng này và việc bầu chọn sẽ trở nên phức tạp hơn.**

* Giải thích: Ở phía nhận, mô-đun AUTOSAR COM thực hiện bầu chọn trên các I-PDU sao chép. Chỉ khi I-PDU sao chép được xác nhận thành công thì tín hiệu mới được chuyển đến RTE. Các I-PDU sao chép cần phải có bộ đếm I-PDU và ngưỡng bộ đếm (ComIPduCounterThreshold) phải được cấu hình là 0.

**SWS\_Com\_00596**

Chỉ khi mô-đun AUTOSAR COM nhận được ít nhất ComIPduReplicationQuorum bản sao giống hệt của một I-PDU sao chép, nó mới cung cấp các tín hiệu và/hoặc nhóm tín hiệu từ I-PDU này đến RTE.

* Giải thích: Mô-đun AUTOSAR COM chỉ cung cấp tín hiệu và/hoặc nhóm tín hiệu từ I-PDU sao chép đến RTE khi nó nhận được ít nhất số lượng bản sao giống hệt nhau của I-PDU này bằng ComIPduReplicationQuorum.

**SWS\_Com\_00597**

Khi mô-đun AUTOSAR COM nhận được ít nhất ComIPduReplicationQuorum bản sao giống hệt của một I-PDU sao chép, nó sẽ thông báo cho RTE chỉ một lần về các tín hiệu và/hoặc nhóm tín hiệu từ I-PDU này.

* Giải thích: Khi mô-đun AUTOSAR COM nhận được ít nhất ComIPduReplicationQuorum bản sao giống hệt của một I-PDU sao chép, nó sẽ thông báo cho RTE chỉ một lần về các tín hiệu và/hoặc nhóm tín hiệu từ I-PDU này.

**Bộ đếm I-PDU sẽ được xử lý riêng cho mỗi I-PDU sao chép như được định nghĩa trong Chương 7.9.**

* Giải thích: Bộ đếm I-PDU sẽ được xử lý riêng cho mỗi I-PDU sao chép như đã được định nghĩa trong chương trước.

**Vì các I-PDU không chứa một checksum đặc biệt, việc so sánh dự kiến sẽ được thực hiện bằng cách so sánh toàn bộ I-PDU từng byte một.**

* Giải thích: Vì các I-PDU không chứa mã kiểm tra (checksum) đặc biệt, việc so sánh dự kiến sẽ được thực hiện bằng cách so sánh toàn bộ I-PDU từng byte một.

**Đối với việc sao chép I-PDU, ComIPduCounterThreshold cần được cấu hình là 0, xem SWS\_Com\_00834. Do đó, ComIPduReplicaRef + 1 bộ đệm là đủ để lưu trữ các I-PDU cho một cuộc bầu chọn thành công.**

* Giải thích: Để sao chép I-PDU, ngưỡng bộ đếm I-PDU (ComIPduCounterThreshold) cần được cấu hình là 0. Vì vậy, ComIPduReplicaRef + 1 bộ đệm là đủ để lưu trữ các I-PDU cho một cuộc bầu chọn thành công.

#### Constraints

**Với việc tái sử dụng chức năng đã tồn tại, nghĩa là sử dụng PduR cho phân tán (fan out), cần chú ý thiết lập hệ thống đúng cách. PduR cho phép phân tán các I-PDU sao chép trên các bus khác nhau. Ở phía nhận, các I-PDU này được chỉ định cho COM với cùng một ID I-PDU. Tuy nhiên, Com\_RxIndication (SWS\_Com\_00123) không thể được gọi lại (reentrant) cho cùng một ID I-PDU.**

* Giải thích: Khi tái sử dụng các chức năng đã có sẵn như sử dụng PduR để phân tán các I-PDU sao chép (fan out), cần phải cẩn thận thiết lập hệ thống một cách chính xác. PduR cho phép phân tán các I-PDU sao chép trên nhiều bus khác nhau. Khi nhận, các I-PDU này sẽ được chỉ định cho mô-đun COM với cùng một ID I-PDU. Tuy nhiên, hàm Com\_RxIndication không thể được gọi lại đồng thời (reentrant) cho cùng một ID I-PDU.

**Một cách dễ dàng để ngăn chặn điều này là đảm bảo rằng tất cả các I-PDU được nhận bởi cùng một mô-đun giao diện, dẫn đến việc sử dụng cùng một bus cho tất cả các bản sao. Đối với ngăn xếp AUTOSAR hiện tại, đây là phương pháp được khuyến nghị.**

* Giải thích: Để tránh vấn đề này, một cách đơn giản là đảm bảo rằng tất cả các I-PDU được nhận bởi cùng một mô-đun giao diện, điều này sẽ dẫn đến việc sử dụng cùng một bus cho tất cả các bản sao. Đây là phương pháp được khuyến nghị cho ngăn xếp AUTOSAR hiện tại.

**Tuy nhiên, người tích hợp hệ thống cũng có thể đồng bộ hóa đường dẫn nhận bằng các phương pháp khác.**

* Giải thích: Người tích hợp hệ thống cũng có thể sử dụng các phương pháp khác để đồng bộ hóa đường dẫn nhận.

**Hơn nữa, vì không có cơ chế nào để ngăn chặn các cập nhật từ lớp trên đến một I-PDU, phương pháp bảo vệ truyền thông này không hoạt động kết hợp với việc truyền tải tách biệt, tức là các I-PDU được yêu cầu qua Com\_TriggerTransmit.**

* Giải thích: Ngoài ra, vì không có cơ chế nào để ngăn chặn các cập nhật từ lớp trên đến một I-PDU, phương pháp bảo vệ truyền thông này không thể hoạt động khi kết hợp với phương pháp truyền tải tách biệt, tức là các I-PDU được yêu cầu qua hàm Com\_TriggerTransmit.

### Signal Gateway - Cổng Tín Hiệu

Cổng tín hiệu là một phần tích hợp của mô-đun AUTOSAR COM. Bất kỳ mô-đun bên ngoài nào, ngoại trừ cuộc gọi tác vụ chu kỳ, không thể truy cập cổng tín hiệu. Cổng tín hiệu của mô-đun AUTOSAR COM hoạt động với (nhóm) tín hiệu và nhóm tín hiệu.

**Giải thích:** Cổng tín hiệu này là một phần của mô-đun AUTOSAR COM và chỉ có thể được truy cập bởi các cuộc gọi chu kỳ bên trong hệ thống AUTOSAR. Nó làm việc với các tín hiệu và nhóm tín hiệu.

Cổng tín hiệu của mô-đun AUTOSAR COM chỉ hỗ trợ định tuyến tĩnh cho container cấu hình **ComGwMapping (ECUC\_Com\_00544)**. Chỉ có thể cấu hình các tuyến đường một cách độc lập với nội dung của các tín hiệu và nhóm tín hiệu được định tuyến.

**Giải thích:** Cổng tín hiệu chỉ hỗ trợ định tuyến tĩnh, có nghĩa là các tuyến đường phải được cấu hình trước và không thể thay đổi dựa trên nội dung của các tín hiệu.

Điểm đích của một tín hiệu hoặc nhóm tín hiệu có thể được cấu hình theo mỗi container **ComGwDestinationDescription (ECUC\_Com\_00549)**, hoặc thông qua tham chiếu **ComGwSignal (ECUC\_Com\_00551)**. Chế độ truyền sẽ được chọn dựa trên TMCs của các tín hiệu khác liên quan của I-PDU. Trong trường hợp không có tín hiệu nào đóng góp vào tính toán TMS, chế độ truyền **ComTxModeTrue (SWS\_Com\_00677)** sẽ được sử dụng.

**Giải thích:** Điểm đến của tín hiệu hoặc nhóm tín hiệu có thể được cấu hình cụ thể. Chế độ truyền sẽ được chọn dựa trên các điều kiện truyền của các tín hiệu khác. Nếu không có tín hiệu nào ảnh hưởng đến việc chọn chế độ truyền, chế độ mặc định **ComTxModeTrue** sẽ được sử dụng.

SWS\_Com\_00377

* **Yêu cầu:** ⌈Cổng tín hiệu của mô-đun AUTOSAR COM sẽ sao chép giá trị của các tín hiệu/ nhóm tín hiệu được định tuyến tới các tín hiệu/ nhóm tín hiệu để truyền theo cấu hình, xem **ECUC\_Com\_00544**.⌋ (SRS\_PduR\_06055, SRS\_PduR\_06089)
* **Giải thích:** Cổng tín hiệu sẽ sao chép giá trị của các tín hiệu hoặc nhóm tín hiệu từ nguồn đến đích theo cấu hình đã định trong **ComGwMapping**.

Cấu hình **ComGwMapping** cho phép định tuyến một tín hiệu hoặc nhóm tín hiệu từ một nguồn tín hiệu hoặc nhóm tín hiệu đến không (không có chức năng cổng tín hiệu) hoặc nhiều đích.

**Giải thích:** **ComGwMapping** cho phép bạn định tuyến một tín hiệu hoặc nhóm tín hiệu từ một nguồn đến nhiều đích hoặc không định tuyến nếu không cần thiết.

Cổng tín hiệu của mô-đun AUTOSAR COM không hỗ trợ xử lý các tín hiệu hoặc nhóm tín hiệu nằm trong một **I-PDU lớn**, xem **SWS\_Com\_00598**.

**Giải thích:** Cổng tín hiệu không thể xử lý các tín hiệu hoặc nhóm tín hiệu nằm trong một I-PDU lớn, vì vậy nếu một I-PDU chứa quá nhiều dữ liệu, cổng tín hiệu sẽ không thể xử lý nó.

#### Dealing with Signals (Xử lý Tín Hiệu)

SWS\_Com\_00357

* **Yêu cầu:** ⌈Mô-đun AUTOSAR COM sẽ chuyển tiếp các tín hiệu cần định tuyến từ I-PDU nhận đến I-PDU truyền. Để cấu hình, xem container cấu hình **ECUC\_Com\_00544 ComGwMapping**.⌋ (SRS\_PduR\_06002)
* **Giải thích:** Mô-đun AUTOSAR COM sẽ chuyển các tín hiệu từ I-PDU nhận sang I-PDU truyền theo cấu hình được định nghĩa trong **ComGwMapping**.

Việc dừng I-PDU nhận sau khi các tín hiệu cần định tuyến đã được chuyển tiếp đến I-PDU truyền sẽ không ngăn chặn I-PDU truyền được truyền trừ khi nó cũng bị dừng.

**Giải thích:** Nếu I-PDU nhận bị dừng sau khi các tín hiệu đã được chuyển tiếp, điều này không ngăn chặn I-PDU truyền trừ khi I-PDU truyền cũng bị dừng.

SWS\_Com\_00360

* **Yêu cầu:** ⌈Nếu endianness của một tín hiệu nhận được để định tuyến khác với endianness của tín hiệu đích liên quan, mô-đun AUTOSAR COM sẽ chuyển đổi endianness của nó sang endianness của tín hiệu đích.⌋ (SRS\_PduR\_06061)
* **Giải thích:** Nếu endianness (thứ tự byte) của tín hiệu nhận khác với tín hiệu đích, mô-đun AUTOSAR COM sẽ chuyển đổi endianness của tín hiệu nhận để phù hợp với tín hiệu đích.

#### Dealing with Signal Groups (Xử lý Nhóm Tín Hiệu)

SWS\_Com\_00361

* **Yêu cầu:** ⌈Mô-đun AUTOSAR COM sẽ chuyển tiếp các nhóm tín hiệu cần định tuyến từ I-PDU nhận đến I-PDU truyền. Xem thêm container cấu hình **ComGwMapping (ECUC\_Com\_00544)**.⌋ (SRS\_PduR\_06002, SRS\_PduR\_06056, SRS\_PduR\_06061)
* **Giải thích:** Mô-đun AUTOSAR COM sẽ chuyển các nhóm tín hiệu từ I-PDU nhận sang I-PDU truyền theo cấu hình được định nghĩa trong **ComGwMapping**.

Việc dừng I-PDU nhận sau khi các nhóm tín hiệu cần định tuyến đã được chuyển tiếp đến I-PDU truyền sẽ không ngăn chặn I-PDU truyền được truyền trừ khi nó cũng bị dừng.

**Giải thích:** Nếu I-PDU nhận bị dừng sau khi các nhóm tín hiệu đã được chuyển tiếp, điều này không ngăn chặn I-PDU truyền trừ khi I-PDU truyền cũng bị dừng.

SWS\_Com\_00383

* **Yêu cầu:** ⌈Mô-đun AUTOSAR COM sẽ định tuyến các nhóm tín hiệu một cách nhất quán. Do đó, mô-đun AUTOSAR COM sẽ chuyển dữ liệu của một nhóm tín hiệu như một tập dữ liệu nhất quán trong suốt quá trình định tuyến.⌋ (SRS\_PduR\_06056)
* **Giải thích:** Mô-đun AUTOSAR COM sẽ đảm bảo rằng các nhóm tín hiệu được định tuyến một cách nhất quán, chuyển toàn bộ dữ liệu nhóm tín hiệu như một khối thống nhất.

SWS\_Com\_00735

* **Yêu cầu:** ⌈Mô-đun AUTOSAR COM sẽ hỗ trợ định tuyến một cách nhất quán một tập con các tín hiệu nhóm từ nhóm tín hiệu nguồn vào một nhóm tín hiệu đích giảm bớt.⌋ (SRS\_PduR\_06056)
* **Giải thích:** Mô-đun AUTOSAR COM có thể định tuyến một phần của nhóm tín hiệu nguồn vào một nhóm tín hiệu đích nhỏ hơn, vẫn đảm bảo tính nhất quán.

SWS\_Com\_00833

* **Yêu cầu:** ⌈Mô-đun AUTOSAR COM sẽ hỗ trợ định tuyến các tín hiệu nhóm riêng lẻ từ nhóm tín hiệu nguồn đến các tín hiệu đích riêng lẻ (không phải là một phần của nhóm tín hiệu).⌋ (SRS\_PduR\_06056)
* **Giải thích:** Mô-đun AUTOSAR COM có thể định tuyến các tín hiệu nhóm riêng lẻ từ nhóm tín hiệu nguồn đến các tín hiệu đích riêng lẻ mà không cần nhóm.

SWS\_Com\_00362

* **Yêu cầu:** ⌈Nếu endianness của một tín hiệu nhóm nhận được để định tuyến khác với endianness của tín hiệu (nhóm) đích liên quan, mô-đun AUTOSAR COM sẽ chuyển đổi endianness của nó sang endianness của tín hiệu (nhóm) đích.⌋ (SRS\_PduR\_06061)
* **Giải thích:** Nếu endianness của tín hiệu nhóm nhận khác với tín hiệu (nhóm) đích, mô-đun AUTOSAR COM sẽ chuyển đổi endianness của tín hiệu nhận để phù hợp với tín hiệu đích.

#### Routing of Out-Timed Signals and Signal Groups (Định tuyến Tín Hiệu và Nhóm Tín Hiệu Ngoài Thời Gian)

SWS\_Com\_00701

* **Yêu cầu:** ⌈Cổng tín hiệu của mô-đun AUTOSAR COM sẽ định tuyến tín hiệu và nhóm tín hiệu ngay cả khi bất kỳ thời gian chờ giám sát thời hạn nhận cấu hình nào đã hết hạn.⌋ (SRS\_PduR\_06089)
* **Giải thích:** Mô-đun AUTOSAR COM sẽ định tuyến tín hiệu và nhóm tín hiệu ngay cả khi thời gian chờ nhận của chúng đã hết hạn.

#### Handling of Update-Bits (Xử lý Bit Cập Nhật)

SWS\_Com\_00702

* **Yêu cầu:** ⌈Nếu cả tín hiệu/ nhóm tín hiệu nhận được và tín hiệu/ nhóm tín hiệu đích đều có cấu hình bit cập nhật (**ComUpdateBitPosition**) và bit cập nhật của tín hiệu/ nhóm tín hiệu nhận được được đặt, mô-đun AUTOSAR COM sẽ định tuyến tín hiệu/ nhóm tín hiệu với bit cập nhật đã đặt và xóa bit cập nhật của tín hiệu/ nhóm tín hiệu đích sau khi nó được gửi.⌋ (SRS\_Com\_02030)
* **Giải thích:** Nếu cả tín hiệu nhận và tín hiệu đích đều có bit cập nhật và bit cập nhật của tín hiệu nhận được đặt, mô-đun sẽ định tuyến tín hiệu với bit cập nhật đã đặt và xóa bit cập nhật của tín hiệu đích sau khi gửi.

SWS\_Com\_00703

* **Yêu cầu:** ⌈Nếu tín hiệu/ nhóm tín hiệu nhận được và tín hiệu/ nhóm tín hiệu đích có cấu hình bit cập nhật (**ComUpdateBitPosition**), và bit cập nhật của tín hiệu/ nhóm tín hiệu nhận được không được đặt, mô-đun AUTOSAR COM sẽ không định tuyến tín hiệu/ nhóm tín hiệu này.⌋ (SRS\_Com\_02030)
* **Giải thích:** Nếu tín hiệu nhận và tín hiệu đích đều có bit cập nhật nhưng bit cập nhật của tín hiệu nhận không được đặt, mô-đun sẽ không định tuyến tín hiệu này.

SWS\_Com\_00704

* **Yêu cầu:** ⌈Nếu tín hiệu/ nhóm tín hiệu nhận được có cấu hình bit cập nhật (**ComUpdateBitPosition**), nhưng tín hiệu đích không có cấu hình bit cập nhật và bit cập nhật được đặt, mô-đun AUTOSAR COM sẽ định tuyến tín hiệu/ nhóm tín hiệu này mà không có bit cập nhật.⌋ (SRS\_Com\_02030)
* **Giải thích:** Nếu tín hiệu nhận có bit cập nhật nhưng tín hiệu đích không có, và bit cập nhật của tín hiệu nhận được đặt, mô-đun sẽ định tuyến tín hiệu mà không có bit cập nhật.

SWS\_Com\_00705

* **Yêu cầu:** ⌈Nếu tín hiệu/ nhóm tín hiệu nhận được có cấu hình bit cập nhật (**ComUpdateBitPosition**), nhưng tín hiệu đích không có cấu hình bit cập nhật và bit cập nhật không được đặt, mô-đun AUTOSAR COM sẽ không định tuyến tín hiệu/ nhóm tín hiệu này.⌋ (SRS\_Com\_02030)
* **Giải thích:** Nếu tín hiệu nhận có bit cập nhật nhưng tín hiệu đích không có, và bit cập nhật của tín hiệu nhận không được đặt, mô-đun sẽ không định tuyến tín hiệu này.

SWS\_Com\_00706

* **Yêu cầu:** ⌈Nếu tín hiệu/ nhóm tín hiệu nhận được không có cấu hình bit cập nhật (**ComUpdateBitPosition**) và tín hiệu/ nhóm tín hiệu đích có cấu hình bit cập nhật, mô-đun AUTOSAR COM sẽ đặt bit cập nhật của tín hiệu đích khi một tín hiệu/ nhóm tín hiệu mới được nhận và xóa nó sau khi gửi tín hiệu/ nhóm tín hiệu đích.⌋ (SRS\_Com\_02030)
* **Giải thích:** Nếu tín hiệu nhận không có bit cập nhật nhưng tín hiệu đích có, mô-đun sẽ đặt bit cập nhật của tín hiệu đích khi tín hiệu mới được nhận và xóa nó sau khi gửi.

#### Decoupling Signal Gateway (Tách biệt Cổng Tín Hiệu)

Để bảo vệ các quy trình ngắt được sử dụng cho việc nhận I-PDU khỏi việc sử dụng thời gian không thể tính toán và có thể đắt đỏ, cần phải tách biệt cổng tín hiệu của mô-đun AUTOSAR COM khỏi các quy trình ngắt.

SWS\_Com\_00359

* **Yêu cầu:** ⌈Mô-đun AUTOSAR COM sẽ thực hiện tất cả các chức năng của cổng tín hiệu trong cuộc gọi hàm **Com\_MainFunctionRouteSignals** (SWS\_Com\_00400). Trong cuộc gọi hàm này, cổng tín hiệu của mô-đun AUTOSAR COM sẽ kiểm tra các tín hiệu và nhóm tín hiệu nhận được cần định tuyến và chuyển tiếp chúng từ các I-PDU nhận liên quan đến các I-PDU truyền liên quan.⌋ (SRS\_BSW\_00425, SRS\_BSW\_00432)
* **Giải thích:** Tất cả các chức năng của cổng tín hiệu sẽ được thực hiện trong cuộc gọi hàm **Com\_MainFunctionRouteSignals**. Trong cuộc gọi này, mô-đun sẽ kiểm tra các tín hiệu cần định tuyến và chuyển tiếp chúng.

SWS\_Com\_00466

* **Yêu cầu:** ⌈Trong **Com\_MainFunctionRouteSignals**, mô-đun AUTOSAR COM sẽ đánh giá các thuộc tính truyền và chế độ truyền và thực hiện các hành động định tuyến theo trình tự sau, xem thêm Hình 3:

1. Sao chép tất cả các tín hiệu từ nguồn đến I-PDUs đích
2. Đánh giá TMC của tất cả các tín hiệu
3. Đánh giá TMS cho I-PDUs đích⌋ (SRS\_BSW\_00432)

* **Giải thích:** Trong **Com\_MainFunctionRouteSignals**, mô-đun sẽ đánh giá các thuộc tính truyền và chế độ truyền, thực hiện các bước sao chép tín hiệu, đánh giá TMC và TMS cho I-PDUs đích.

Trong trường hợp, một I-PDU đích chứa các tín hiệu có thuộc tính truyền **ComTransferProperty TRIGGERED** mới nhận được, việc gửi có thể được thực hiện trong **Com\_MainFunctionTx** hoặc **Com\_MainFunctionRouteSignals**.

SWS\_Com\_00539

* **Yêu cầu:** ⌈Mô-đun AUTOSAR COM sẽ gửi một I-PDU nhiều nhất một lần trong một cuộc gọi đến **Com\_MainFunctionRouteSignals**.⌋ (SRS\_PduR\_06055)
* **Giải thích:** Mô-đun AUTOSAR COM sẽ chỉ gửi một I-PDU một lần trong mỗi cuộc gọi đến **Com\_MainFunctionRouteSignals**.

### Error Classification (Phân Loại Lỗi)

Mô-đun AUTOSAR COM hiện tại không hỗ trợ các lỗi sản xuất.

| Type of error | Relevance | Related error code | Value [hex] |
| --- | --- | --- | --- |
| **SWS\_Com\_00803** | **API service called with wrong parameter** | **Development** | **COM\_E\_PARAM** |
| **SWS\_Com\_00804** | **Error code if any other API service, except Com\_GetStatus, is called before the AUTOSAR COM module was initialized with Com\_Init or after a call to Com\_Deinit** | **Development** | **COM\_E\_UNINIT** |
| **SWS\_Com\_00805** | **NULL pointer checking** | **Development** | **COM\_E\_PARAM\_POINTER** |
| **SWS\_Com\_00837** | **Invalid configuration set selection** | **Development** | **COM\_E\_INIT\_FAILED** |
| **SWS\_Com\_00864** | **Transmission request was skipped** | **Runtime** | **COM\_E\_SKIPPED\_TRANSMISSION** |

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

**SWS\_Com\_00803:**

* + **Loại lỗi:** API service called with wrong parameter.
  + **Mức độ liên quan:** Development.
  + **Mã lỗi liên quan:** COM\_E\_PARAM.
  + **Giá trị:** 0x01.
  + **Giải thích:** Lỗi này xảy ra khi một dịch vụ API được gọi với tham số không đúng. Điều này thường được phát hiện trong quá trình phát triển và thử nghiệm, đảm bảo rằng các tham số được truyền vào các hàm API là hợp lệ.

**SWS\_Com\_00804:**

* + **Loại lỗi:** Error code if any other API service, except Com\_GetStatus, is called before the AUTOSAR COM module was initialized with Com\_Init or after a call to Com\_Deinit.
  + **Mức độ liên quan:** Development.
  + **Mã lỗi liên quan:** COM\_E\_UNINIT.
  + **Giá trị:** 0x02.
  + **Giải thích:** Lỗi này xuất hiện khi bất kỳ dịch vụ API nào, ngoại trừ Com\_GetStatus, được gọi trước khi mô-đun AUTOSAR COM được khởi tạo bằng Com\_Init hoặc sau khi gọi Com\_Deinit. Điều này đảm bảo rằng các dịch vụ API chỉ được gọi khi mô-đun đang ở trạng thái hợp lệ.

**SWS\_Com\_00805:**

* + **Loại lỗi:** NULL pointer checking.
  + **Mức độ liên quan:** Development.
  + **Mã lỗi liên quan:** COM\_E\_PARAM\_POINTER.
  + **Giá trị:** 0x03.
  + **Giải thích:** Lỗi này liên quan đến việc kiểm tra các con trỏ NULL. Điều này đảm bảo rằng tất cả các con trỏ được sử dụng trong dịch vụ API đều hợp lệ và không gây ra lỗi bộ nhớ.

**SWS\_Com\_00837:**

* + **Loại lỗi:** Invalid configuration set selection.
  + **Mức độ liên quan:** Development.
  + **Mã lỗi liên quan:** COM\_E\_INIT\_FAILED.
  + **Giá trị:** 0x04.
  + **Giải thích:** Lỗi này xuất hiện khi lựa chọn tập cấu hình không hợp lệ. Điều này đảm bảo rằng các tập cấu hình được sử dụng trong mô-đun AUTOSAR COM đều hợp lệ và đã được kiểm tra trước khi sử dụng.

**SWS\_Com\_00864:**

* + **Loại lỗi:** Transmission request was skipped.
  + **Mức độ liên quan:** Runtime.
  + **Mã lỗi liên quan:** COM\_E\_SKIPPED\_TRANSMISSION.
  + **Giá trị:** 0x05.
  + **Giải thích:** Lỗi này xuất hiện khi một yêu cầu truyền bị bỏ qua. Đây là lỗi thời gian chạy, có thể xảy ra trong quá trình vận hành hệ thống và cần được xử lý thích hợp để đảm bảo quá trình truyền thông diễn ra một cách chính xác.

Những mã lỗi này giúp xác định và xử lý các vấn đề trong quá trình phát triển và vận hành mô-đun AUTOSAR COM một cách hiệu quả.

### Error Notification (Thông báo Lỗi)

[SWS\_Com\_00442]

* **Mô tả:** Khi phát hiện lỗi phát triển, hàm Det\_ReportError của bộ truy vết lỗi phát triển sẽ được gọi với các tham số sau:
  + **ModuleId:** 50 (ModuleId của mô-đun AUTOSAR COM)
  + **InstanceId:** 0
  + **ApiId:** Id dịch vụ của API mô-đun AUTOSAR COM nơi phát hiện lỗi (xem Com\_ServiceIdType)
  + **ErrorId:** Id lỗi như được định nghĩa trong Bảng 9
* **Giải thích:** Khi có lỗi phát triển xảy ra trong mô-đun AUTOSAR COM, hệ thống sẽ gọi hàm Det\_ReportError để báo cáo lỗi này. Hàm này sẽ nhận các tham số gồm:
  + ModuleId là 50, đại diện cho mô-đun AUTOSAR COM.
  + InstanceId là 0, chỉ ra phiên bản của mô-đun.
  + ApiId là Id của dịch vụ API nơi lỗi được phát hiện.
  + ErrorId là Id của lỗi cụ thể, như được định nghĩa trong Bảng 9 (ví dụ: COM\_E\_PARAM, COM\_E\_UNINIT,...).

### Error Handling

AUTOSAR COM hỗ trợ giám sát truyền thông với giám sát thời hạn, xem Chương 7.3.6. Hơn nữa, AUTOSAR COM hỗ trợ kiểm soát thứ tự dữ liệu, xem Chương 7.9 và bảo vệ truyền thông, xem Chương 7.10.

Phản ứng cụ thể của mô-đun AUTOSAR COM đối với lỗi và giá trị trả về của các API được gọi được nêu trong các yêu cầu chức năng tương ứng.

### AUTOSAR COM Module’s Interaction Model

Tổng quan

* **Mô tả:** AUTOSAR COM hỗ trợ giám sát truyền thông với giám sát thời hạn và kiểm soát thứ tự dữ liệu, cũng như bảo vệ truyền thông. Phản ứng cụ thể của mô-đun AUTOSAR COM đối với lỗi và giá trị trả về của các API được gọi được nêu trong các yêu cầu chức năng tương ứng.
* **Giải thích:**
  + **Giám sát truyền thông (Communication Monitoring):** AUTOSAR COM có khả năng giám sát thời hạn (deadline monitoring) để đảm bảo rằng các tín hiệu và I-PDU được truyền và nhận trong khoảng thời gian cho phép. Điều này được mô tả chi tiết trong Chương 7.3.6.
  + **Kiểm soát thứ tự dữ liệu (Data Sequence Control):** Đảm bảo rằng các I-PDU được nhận theo đúng thứ tự và không có dữ liệu bị bỏ sót hoặc trùng lặp. Điều này được mô tả trong Chương 7.9.
  + **Bảo vệ truyền thông (Communication Protection):** Đảm bảo tính toàn vẹn của dữ liệu bằng cách sao chép và so sánh các I-PDU. Điều này giúp ngăn chặn việc mất mát hoặc hỏng dữ liệu. Chi tiết được nêu trong Chương 7.10.

Mô-đun AUTOSAR COM sẽ có các hành động cụ thể đối với từng loại lỗi và các giá trị trả về từ các API sẽ phản ánh các trạng thái lỗi này, giúp hệ thống phát hiện và xử lý lỗi một cách hiệu quả.

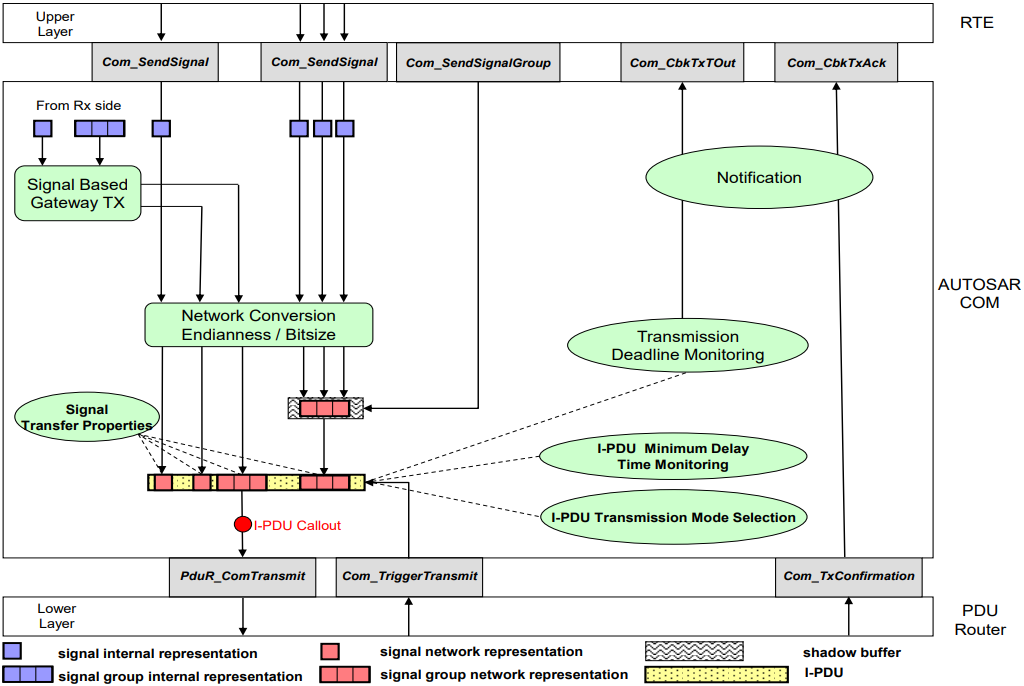


Figure 5 The AUTOSAR COM module’s interaction model for transmission

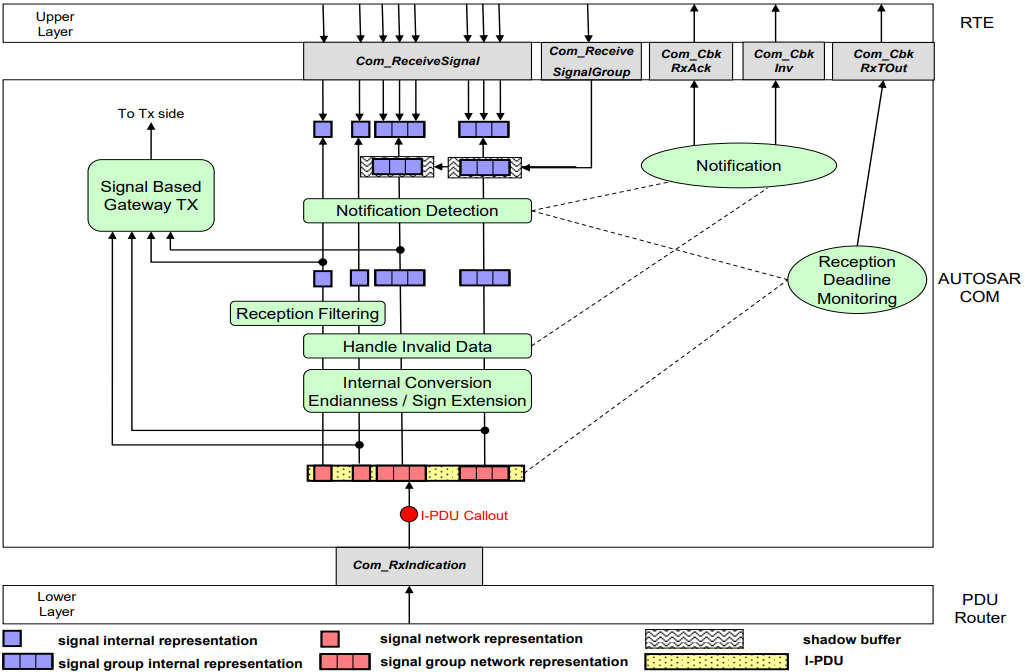


Figure 6 The AUTOSAR COM module’s interaction model for reception

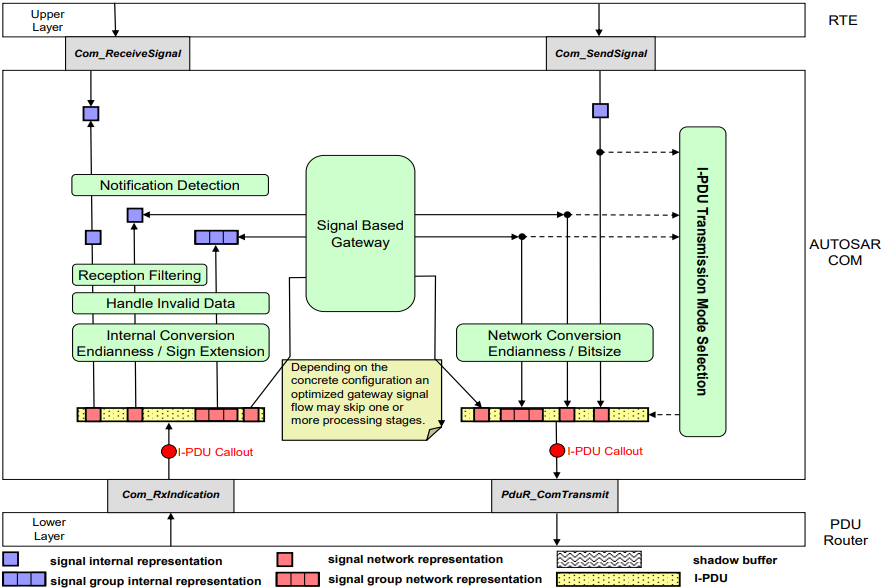


Figure 7: The AUTOSAR COM module’s interaction model for integrated Signal Gateway

Việc chuyển đổi endianness và kéo dài dấu ở phía nhận là cần thiết để cung cấp dữ liệu đúng định dạng cho TMS (Transmission Mode Select). Việc chuyển đổi endianness này chỉ cần thiết nếu endianness của bus nhận khác với endianness của CPU. Việc chuyển đổi endianness ở phía gửi chỉ cần thiết nếu endianness của bus nhận khác với endianness của bus gửi.

**Giải thích:**

* **Chuyển đổi Endianness:** Endianness là thứ tự byte của dữ liệu đa byte (như integer, float) khi nó được lưu trữ trong bộ nhớ hoặc truyền qua mạng. Có hai loại endianness chính:
  + **Big-endian:** Byte quan trọng nhất được lưu ở vị trí thấp nhất.
  + **Little-endian:** Byte quan trọng nhất được lưu ở vị trí cao nhất.

Khi dữ liệu được truyền qua các hệ thống với endianness khác nhau, cần phải chuyển đổi endianness để đảm bảo dữ liệu được hiểu đúng ở cả hai phía.

* **Kéo dài Dấu (Sign Extension):** Đây là quá trình mở rộng số bit của một giá trị số nguyên có dấu để giữ nguyên giá trị của nó khi chuyển đổi giữa các kích thước dữ liệu khác nhau. Ví dụ, khi một số nguyên 8-bit có dấu được mở rộng thành số nguyên 16-bit, bit dấu (bit quan trọng nhất) sẽ được nhân bản để giữ nguyên giá trị.
* **Trường hợp Chuyển đổi Endianness trên phía Nhận:**
  + Khi bus nhận dữ liệu có endianness khác với CPU, dữ liệu cần được chuyển đổi để CPU có thể xử lý đúng. Điều này đảm bảo rằng dữ liệu khi vào TMS có định dạng chính xác.
* **Trường hợp Chuyển đổi Endianness trên phía Gửi:**
  + Khi bus nhận và bus gửi có endianness khác nhau, dữ liệu cần được chuyển đổi để đảm bảo rằng khi dữ liệu đến bus nhận, nó có thể được hiểu đúng. Điều này đảm bảo rằng dữ liệu truyền đi không bị sai lệch do khác biệt về endianness.

Tóm lại, việc chuyển đổi endianness và kéo dài dấu là cần thiết để đảm bảo tính tương thích dữ liệu giữa các hệ thống có endianness khác nhau và để dữ liệu được xử lý đúng đắn trong quá trình truyền thông.

RTE



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Upper Layer | | | | | | | | | | | | | | | |
|  | ***Com\_SendSignal*** | |  | ***Com\_SendSignal*** | | |  | ***Com\_SendSignalGroup*** | |  | ***Com\_CbkTxTOut*** |  | ***Com\_CbkTxAck*** |  | |
| From Rx side  Signal Based Notification  Gateway TX  Network Conversion  Endianness / Bitsize Transmission  Deadline Monitoring  **Signal Transfer Properties**  **I-PDU Minimum Delay Time Monitoring**  I-PDU Callout **I-PDU Transmission Mode Selection** | | | | | | | | | | | | | | | |
|  | | ***PduR\_ComTransmit*** | | |  | ***Com\_TriggerTransmit*** | | |  | | | | ***Com\_TxConfirmation*** | |  |
| Lower Layer | | | | | | | | | | | | | | | |

AUTOSAR COM

**signal internal representation signal group internal representation**

**signal network representation signal group network representation**

**shadow buffer I-PDU**



PDU

Router

**Figure 5 The AUTOSAR COM module’s interaction model for transmission**

RTE



|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Upper Layer | | | | | | | | | | | |
|  | ***Com\_ReceiveSignal*** | | |  | ***Com\_Receive***  ***SignalGroup*** |  | ***Com\_Cbk RxAck*** |  | ***Com\_Cbk Inv*** |  | ***Com\_Cbk***  ***RxTOut*** |
| To Tx side  Notification  Signal Based  Gateway TX Notification Detection  Reception Deadline Monitoring  Reception Filtering  Handle Invalid Data Internal Conversion  Endianness / Sign Extension  I-PDU Callout | | | | | | | | | | | |
|  | | ***Com\_RxIndication*** |  | | | | | | | | |
| Lower Layer | | | | | | | | | | | |

AUTOSAR COM

**signal internal representation signal group internal representation**

**signal network representation signal group network representation**

**shadow buffer I-PDU**



PDU

Router

**Figure 6 The AUTOSAR COM module’s interaction model for reception**

RTE



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Upper Layer | | | | | | | | |
|  | ***Com\_ReceiveSignal*** | |  | | | ***Com\_SendSignal*** | |  |
| Notification Detection  Signal Based Gateway  Reception Filtering Handle Invalid Data  Internal Conversion Network Conversion Endianness / Sign Extension Endianness / Bitsize  Depending on the concrete configuration an optimized gateway signal flow may skip one or more processing stages.  I-PDU Callout I-PDU Callout | | | | | | | | |
|  | | ***Com\_RxIndication*** | |  | ***PduR\_ComTransmit*** | |  | |
| Lower Layer | | | | | | | | |

AUTOSAR COM

**I-PDU Transmission Mode Selection**

**signal internal representation signal group internal representation**

**signal network representation signal group network representation**

**shadow buffer I-PDU**



PDU

Router

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

The endianness conversion and sign extension on receiver side are needed to feed the TMS with a correct data format. This en- dianness 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.

## API Specification

### Imported Types

Trong chương này, tất cả các loại bao gồm từ các tệp sau được liệt kê:

[SWS\_Com\_00609] ⌈

|  |  |
| --- | --- |
| ***Module*** | ***Imported Type*** |
| ComStack\_Types | BufReq\_ReturnType |
| PduIdType |
| PduInfoType |
| PduLengthType |
| RetryInfoType |
| Std\_Types | Std\_ReturnType |
| Std\_VersionInfoType |

⌋ (SRS\_BSW\_00301)

**Giải thích:**

1. **ComStack\_Types:** Đây là module định nghĩa các loại dữ liệu cơ bản được sử dụng trong ngăn xếp truyền thông của AUTOSAR. Các loại dữ liệu này bao gồm:
   * **BufReq\_ReturnType:** Loại này xác định các mã trả về cho các yêu cầu bộ đệm.
   * **PduIdType:** Loại này xác định ID của PDU (Protocol Data Unit).
   * **PduInfoType:** Loại này chứa thông tin về PDU, bao gồm con trỏ đến dữ liệu PDU và độ dài của dữ liệu.
   * **PduLengthType:** Loại này xác định độ dài của PDU.
   * **RetryInfoType:** Loại này chứa thông tin về việc thử lại truyền thông, nếu cần.
2. **Std\_Types:** Đây là module định nghĩa các loại dữ liệu chuẩn được sử dụng trong nhiều module của AUTOSAR. Các loại dữ liệu này bao gồm:
   * **Std\_ReturnType:** Loại này xác định các mã trả về chuẩn cho các hàm.
   * **Std\_VersionInfoType:** Loại này chứa thông tin về phiên bản của một module.

Các loại dữ liệu được liệt kê trên đây là các loại cơ bản được sử dụng trong mô-đun AUTOSAR COM và các mô-đun liên quan khác để đảm bảo tính nhất quán và tương thích giữa các phần khác nhau của hệ thống. Việc nhập khẩu các loại này giúp chuẩn hóa cách truyền thông tin giữa các mô-đun, đảm bảo rằng mọi thành phần của hệ thống có thể giao tiếp với nhau một cách hiệu quả và chính xác.

### Type Definitions

#### Com\_StatusType

[SWS\_Com\_00819] ⌈

|  |  |  |
| --- | --- | --- |
| ***Name:*** | Com\_StatusType | |
| ***Type:*** | Enumeration | |
| ***Range:*** | COM\_INIT | Mô-đun AUTOSAR COM đã được khởi tạo và có thể sử dụng. |
| COM\_UNINIT | Mô-đun AUTOSAR COM chưa được khởi tạo hoặc không thể sử dụng. Đây sẽ là giá trị mặc định sau khi reset. Trạng thái này sẽ có giá trị là 0. |
| ***Description:*** | Đây là giá trị trạng thái được trả về bởi dịch vụ API Com\_GetStatus(). | |

⌋ (SRS\_BSW\_00335)

**Giải thích:** Com\_StatusType là một kiểu dữ liệu liệt kê (enumeration) xác định trạng thái hiện tại của mô-đun AUTOSAR COM. Nó có hai trạng thái chính:

* **COM\_INIT:** Cho biết mô-đun đã được khởi tạo và có thể sử dụng.
* **COM\_UNINIT:** Cho biết mô-đun chưa được khởi tạo hoặc không thể sử dụng, và đây là giá trị mặc định sau khi hệ thống được reset.

#### Com\_SignalIdType

[SWS\_Com\_00820] ⌈

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name:*** | Com\_SignalIdType | | |
| ***Type:*** | uint16 | | |
| ***Range:*** | 0..<SignalIdmax> | -- | Số nguyên không dấu bắt đầu từ 0. |
| ***Description:*** | Định danh đối tượng tín hiệu của mô-đun AUTOSAR COM. | | |

⌋ (SRS\_Com\_02037, SRS\_BSW\_00441)

**Giải thích:** Com\_SignalIdType là một kiểu dữ liệu xác định bằng số nguyên không dấu 16 bit (uint16) được sử dụng để định danh các đối tượng tín hiệu trong mô-đun AUTOSAR COM. Dãy giá trị của nó bắt đầu từ 0 và có thể lên đến một giá trị tối đa được xác định bởi hệ thống.

#### Com\_SignalGroupIdType

[SWS\_Com\_00821] ⌈

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name:*** | Com\_SignalGroupIdType | | |
| ***Type:*** | uint16 | | |
| ***Range:*** | 0..<SignalGroupIdmax> | -- | Số nguyên không dấu bắt đầu từ 0. |
| ***Description:*** | Định danh đối tượng nhóm tín hiệu của mô-đun AUTOSAR COM. | | |

⌋ (SRS\_Com\_02041, SRS\_BSW\_00441)

**Giải thích:** Com\_SignalGroupIdType là một kiểu dữ liệu xác định bằng số nguyên không dấu 16 bit (uint16) được sử dụng để định danh các đối tượng nhóm tín hiệu trong mô-đun AUTOSAR COM. Dãy giá trị của nó bắt đầu từ 0 và có thể lên đến một giá trị tối đa được xác định bởi hệ thống.

#### Com\_IpduGroupIdType

[SWS\_Com\_00822] ⌈

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name:*** | Com\_IpduGroupIdType | | |
| ***Type:*** | uint16 | | |
| ***Range:*** | 0..<IpduGroupId- max> | -- | Số nguyên không dấu bắt đầu từ 0, trong đó IpduGroupId-max nhỏ hơn ComSupportedIpduGroups. |
| ***Description:*** | Định danh đối tượng nhóm I-PDU của mô-đun AUTOSAR COM. | | |

⌋ (SRS\_Com\_00218, SRS\_BSW\_00441)

**Giải thích:** Com\_IpduGroupIdType là một kiểu dữ liệu xác định bằng số nguyên không dấu 16 bit (uint16) được sử dụng để định danh các đối tượng nhóm I-PDU trong mô-đun AUTOSAR COM. Dãy giá trị của nó bắt đầu từ 0 và có thể lên đến một giá trị tối đa được xác định bởi ComSupportedIpduGroups.

#### Com\_IpduGroupVector

[SWS\_Com\_00823] ⌈

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name:*** | Com\_IpduGroupVector | | |
| ***Type:*** | uint8[(ComSupportedIPduGroups-1)/8+1] | | |
| ***Range:*** | bitfield | -- | Bitfield là một mảng uint8[(ComSupportedIpduGroups-1)/8+1], tức là từ bit0 đến bit<ComSupportedIpduGroups - 1>. |
| ***Description:*** | Kiểu dữ liệu này có thể được sử dụng để lưu trữ một cờ (bit) cho mỗi nhóm I-PDU trong hệ thống. Nó được sử dụng để thiết lập trạng thái kích hoạt và trạng thái giám sát thời hạn cho các nhóm I-PDU trong một lần gọi hàm.groups within one function call | | |

⌋ (SRS\_Com\_02090, SRS\_BSW\_00441)

**Giải thích:** Com\_IpduGroupVector là một kiểu dữ liệu được sử dụng để lưu trữ một bit cho mỗi nhóm I-PDU trong hệ thống. Đây là một mảng các giá trị uint8, trong đó mỗi bit đại diện cho một nhóm I-PDU cụ thể. Kiểu dữ liệu này được sử dụng để thiết lập trạng thái kích hoạt và giám sát thời hạn cho các nhóm I-PDU khi gọi hàm.

#### Com\_ServiceIdType

[SWS\_Com\_00824] ⌈

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name:*** | Com\_ServiceIdType | | |
| ***Type:*** | uint8 | | |
| ***Range:*** | COMServiceId\_Init | 0x01 | -- |
| COMServiceId\_DeInit | 0x02 | -- |
| COMServiceId\_IpduGroupControl | 0x03 | -- |
| COMServiceId\_ReceptionDMControl | 0x06 | -- |
| COMServiceId\_GetStatus | 0x07 | -- |
| COMServiceId\_GetVersionInfo | 0x09 | -- |
| COMServiceId\_SendSignal | 0x0A | -- |
| COMServiceId\_ReceiveSignal | 0x0B | -- |
| COMServiceId\_UpdateShadowSignal | 0x0C | -- |
| COMServiceId\_SendSignalGroup | 0x0D | -- |
| COMServiceId\_ReceiveSignalGroup | 0x0E | -- |
| COMServiceId\_ReceiveShadowSignal | 0x0F | -- |
| COMServiceId\_InvalidateSignal | 0x10 | -- |
| COMServiceId\_InvalidateShadowSignal | 0x16 | -- |
| COMServiceId\_TriggerIPDUSend | 0x17 | -- |
| COMServiceId\_MainFunctionRx | 0x18 | -- |
| COMServiceId\_MainFunctionTx | 0x19 | -- |
| COMServiceId\_MainFunctionRouteSignals | 0x1A | -- |
| COMServiceId\_InvalidateSignalGroup | 0x1B | -- |
| COMServiceId\_ClearIpduGroupVector | 0x1C | -- |
| COMServiceId\_SetIpduGroup | 0x1D | -- |
| COMServiceId\_SendDynSignal | 0x21 | -- |
| COMServiceId\_ReceiveDynSignal | 0x22 | -- |
| COMServiceId\_SendSignalGroupArray | 0x23 | -- |
| COMServiceId\_ReceiveSignalGroupArray | 0x24 | -- |
| COMServiceId\_SwitchIpduTxMode | 0x27 | -- |
| COMServiceId\_TriggerIPDUSendWithMetaData | 0x28 | -- |
| COMServiceId\_TxConfirmation | 0x40 | -- |
| COMServiceId\_TriggerTransmit | 0x41 | -- |
| COMServiceId\_RxIndication | 0x42 | -- |
| COMServiceId\_CopyTxData | 0x43 | -- |
| COMServiceId\_CopyRxData | 0x44 | -- |
| COMServiceId\_TpRxIndication | 0x45 | -- |
| COMServiceId\_StartOfReception | 0x46 | -- |
| COMServiceId\_TpTxConfirmation | 0x48 | -- |
| Description: | Định danh duy nhất của dịch vụ trong mô-đun AUTOSAR COM. Ví dụ: COMServiceId\_SendSignal 0x0A. | | |

⌋ (SRS\_BSW\_00462, SRS\_BSW\_00441)

#### Com\_ConfigType

[SWS\_Com\_00825] ⌈

|  |  |  |
| --- | --- | --- |
| ***Name:*** | Com\_ConfigType | |
| ***Type:*** | Structure | |
| ***Range:*** | implementation specific | Nội dung của cấu trúc dữ liệu khởi tạo phụ thuộc vào từng cách triển khai cụ thể. |
| ***Description:*** | Đây là kiểu của cấu trúc dữ liệu chứa dữ liệu khởi tạo cho COM. | |

⌋ (SRS\_BSW\_00404, SRS\_BSW\_00441)

**Giải thích:** Com\_ConfigType là một kiểu cấu trúc dữ liệu được sử dụng để chứa các thông tin khởi tạo cho mô-đun AUTOSAR COM. Nội dung của cấu trúc này phụ thuộc vào từng cách triển khai cụ thể, tức là mỗi nhà cung cấp hoặc hệ thống có thể định nghĩa nội dung này theo cách riêng của họ. Kiểu dữ liệu này rất quan trọng vì nó cung cấp các tham số và cấu hình cần thiết để khởi động và vận hành mô-đun COM trong hệ thống AUTOSAR.

### Function Definitions

**[SWS\_Com\_00321]**

* **Description:** Non-reentrant functions do not have to check if they are called reentrant.
  + **(SRS\_BSW\_00312)**

**Giải thích:** Các hàm không có khả năng tái nhập (non-reentrant) không cần kiểm tra xem chúng có được gọi lại (reentrant) hay không. Điều này có nghĩa là những hàm này không thể được gọi đồng thời từ nhiều nguồn khác nhau mà không gây ra lỗi hoặc hành vi không mong muốn. Ví dụ, hàm Com\_SendSignal không tái nhập cho cùng một tín hiệu, nhưng có thể tái nhập cho các tín hiệu khác nhau. Vì vậy, nếu có nhiều người gọi mô-đun AUTOSAR COM, họ cần được đồng bộ hóa ở cấp hệ thống, chẳng hạn bằng cách sử dụng các tập hợp tín hiệu và nhóm I-PDU riêng biệt.

#### Return Codes

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

**Giải thích:** Mô-đun AUTOSAR COM không định nghĩa một loại trả về COM đặc biệt. Các dịch vụ API trả về lỗi bằng cách sử dụng Std\_ReturnType như được định nghĩa trong tài liệu [5] hoặc thông qua giá trị uint8 được ánh xạ theo Bảng 10.

Các mã trả về của mô-đun AUTOSAR COM được định nghĩa trong Bảng 10.

**[SWS\_Com\_00865]**

* **Description:** This section describes the return codes for the AUTOSAR COM module.

| **Name** | **Description** | **Type** | **Value** | **Defined in** |
| --- | --- | --- | --- | --- |
| E\_OK | The service has been accepted | #define | 0x00 | Std\_Types.h |
| E\_NOT\_OK | Invocation of service failed | #define | 0x01 | Std\_Types.h |
| COM\_SERVICE\_NOT\_AVAILABLE | The service is currently not available, e.g., the corresponding I-PDU group is stopped or a development error has been detected. | #define | 0x80 | Com.h |
| COM\_BUSY | Successful execution of this function is currently not possible because a resource is currently in use. For example, the buffer of a large I-PDU is locked. COM\_BUSY might, but must not, be returned whenever a signal of a large I-PDU is accessed. | #define | 0x81 | Com.h |

**Giải thích:**

1. **E\_OK (0x00):** Dịch vụ đã được chấp nhận.
2. **E\_NOT\_OK (0x01):** Gọi dịch vụ thất bại.
3. **COM\_SERVICE\_NOT\_AVAILABLE (0x80):** Dịch vụ hiện không khả dụng, ví dụ, nhóm I-PDU tương ứng đã bị dừng hoặc có lỗi phát triển đã được phát hiện.
4. **COM\_BUSY (0x81):** Việc thực hiện chức năng này hiện không thể thực hiện được vì tài nguyên đang được sử dụng. Ví dụ, bộ đệm của một I-PDU lớn đang bị khóa. COM\_BUSY có thể, nhưng không nhất thiết, được trả về khi một tín hiệu của I-PDU lớn được truy cập.

**Table 10: Mapping of AUTOSAR COM module’s return codes**

#### Start-Up and Control Services

Com\_Init [SWS\_Com\_00432] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_Init | |
| ***Syntax:*** | void Com\_Init(  const Com\_ConfigType\* config  ) | |
| ***Service ID[hex]:*** | 0x01 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Không gọi lại | |
| ***Parameters (in):*** | config | Con trỏ tới dữ liệu cấu hình của mô-đun AUTOSAR COM. |
| ***Parameters (in- out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | Dịch vụ này khởi tạo các giao diện nội bộ và bên ngoài cùng các biến của tầng mô-đun AUTOSAR COM để xử lý tiếp theo. Sau khi gọi hàm này, giao tiếp giữa các ECU vẫn bị vô hiệu hóa. | |

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

* Hành vi của mô-đun AUTOSAR COM là không xác định cho đến khi một cuộc gọi đúng đến Com\_Init được thực hiện.

Lưu ý khi sử dụng Com\_Init:

* Không dự kiến rằng Com\_Init sẽ ưu tiên bất kỳ chức năng nào khác của mô-đun AUTOSAR COM. Mô-đun AUTOSAR COM có thể dựa vào phần còn lại của hệ thống rằng Com\_Init không được gọi theo cách như vậy.

##### Com\_DeInit

[SWS\_Com\_00130] ⌈

|  |  |
| --- | --- |
| ***Service name:*** | Com\_DeInit |
| ***Syntax:*** | void Com\_DeInit( void  ) |
| ***Service ID[hex]:*** | 0x02 |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | Không gọi lại |
| ***Parameters (in):*** | None |
| ***Parameters (in- out):*** | None |
| ***Parameters (out):*** | None |
| ***Return value:*** | None |
| ***Description:*** | Dịch vụ này dừng giao tiếp giữa các ECU. Tất cả các nhóm I-PDU đã bắt đầu sẽ bị dừng và cần phải khởi động lại, nếu cần thiết, sau khi Com\_Init được gọi. Bằng cách gọi Com\_DeInit, mô-đun AUTOSAR COM sẽ được đặt vào trạng thái chưa khởi tạo. |

⌋ (SRS\_BSW\_00336)

**[SWS\_Com\_00129]** ⌈Hàm Com\_DeInit của mô-đun AUTOSAR COM sẽ dừng tất cả các nhóm I-PDU đã bắt đầu.(SRS\_BSW\_00336)

Chú ý khi sử dụng Com\_DeInit:

* Không dự kiến rằng Com\_DeInit sẽ ưu tiên bất kỳ chức năng nào khác của mô-đun AUTOSAR COM. Mô-đun AUTOSAR COM có thể dựa vào phần còn lại của hệ thống rằng Com\_DeInit không được gọi theo cách như vậy.

##### Com\_IpduGroupControl

[SWS\_Com\_00751] ⌈

|  |  |  |
| --- | --- | --- |
| **Service name:** | Com\_IpduGroupControl | |
| **Syntax:** | void Com\_IpduGroupControl( Com\_IpduGroupVector ipduGroupVector, boolean initialize  ) | |
| **Service ID[hex]:** | 0x03 | |
| **Sync/Async:** | Synchronous | |
| **Reentrancy:** | Không gọi lại | |
| **Parameters (in):** | ipduGroupVector | Vector nhóm I-PDU chứa trạng thái kích hoạt (dừng = 0 / bắt đầu = 1) cho tất cả các nhóm I-PDU. |
| initialize | Cờ để yêu cầu khởi tạo các I-PDU mới bắt đầu. |
| **Parameters (in- out):** | None | |
| **Parameters (out):** | None | |
| **Return value:** | None | |
| **Description:** | Dịch vụ này bắt đầu hoặc dừng các nhóm I-PDU. | |

⌋ (SRS\_Com\_00218)

[SWS\_Com\_00792] ⌈Khi gọi chức năng Com\_IpduGroupControl, mô-đun AUTOSAR COM sẽ khởi động/dừng mọi ComIPdu theo trạng thái đã được truyền của các ComIpduGroups trong tham số ipduGroupVector.⌋ (SRS\_Com\_00218)

ipduGroupVector đúng có thể được xây dựng bằng cách sử dụng các hàm trợ giúp Com\_ClearIpduGroupVector và Com\_SetIpduGroup. Điều này cho phép sử dụng các tên biểu tượng của các ComIPduGroups.

##### Com\_ReceptionDMControl

[SWS\_Com\_00752] ⌈

|  |  |  |
| --- | --- | --- |
| Service name: | Com\_ReceptionDMControl | |
| Syntax: | void Com\_ReceptionDMControl( Com\_IpduGroupVector ipduGroupVector  ) | |
| Service ID[hex]: | 0x06 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | ipduGroupVector | Vector nhóm I-PDU chứa trạng thái giám sát thời hạn yêu cầu (vô hiệu = 0/ kích hoạt = 1) cho tất cả các nhóm I-PDU. |
| Parameters (in - out): | None | |
| Parameters (out): | None | |
| Return value: | None | |
| Description: | Dịch vụ này kích hoạt hoặc vô hiệu hóa giám sát thời hạn của nhóm I-PDU. | |

⌋ (SRS\_Com\_00192)

[SWS\_Com\_00616] ⌈Đầu tiên, chức năng Com\_ReceptionDMControl sẽ đặt trạng thái giám sát thời hạn tiếp nhận của tất cả các nhóm I-PDU theo trạng thái yêu cầu.⌋ (SRS\_Com\_00192)

[SWS\_Com\_00617] ⌈Thứ hai, chức năng Com\_ReceptionDMControl sẽ khởi động hoặc dừng giám sát thời hạn tiếp nhận cho tất cả các I-PDU thay đổi trạng thái giám sát thời hạn tiếp nhận của chúng do lệnh gọi này của Com\_ReceptionDMControl.⌋ (SRS\_Com\_00192)

[SWS\_Com\_00618] ⌈Đối với tất cả các I-PDU không thay đổi trạng thái giám sát thời hạn của chúng, chức năng Com\_ReceptionDMControl sẽ không làm gì.⌋ (SRS\_Com\_00192)

Top of Form

Bottom of Form

##### Com\_GetStatus

[SWS\_Com\_00194] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_GetStatus | |
| ***Syntax:*** | Com\_StatusType Com\_GetStatus( void) | |
| ***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: mô-đun AUTOSAR COM chưa được khởi tạo và không sử dụng được COM\_INIT: mô-đun AUTOSAR COM đã được khởi tạo và sử dụng được |
| ***Description:*** | Trả về trạng thái của mô-đun AUTOSAR COM. | |

⌋ (SRS\_BSW\_00331)

##### Com\_GetVersionInfo

[SWS\_Com\_00426] ⌈

|  |  |  |
| --- | --- | --- |
| Service name: | Com\_GetVersionInfo | |
| Syntax: | void Com\_GetVersionInfo( Std\_VersionInfoType\* versioninfo  ) | |
| Service ID[hex]: | 0x09 | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Reentrant | |
| Parameters (in): | None | |
| Parameters (in-out): | None | |
| Parameters (out): | versioninfo | Con trỏ đến nơi lưu trữ thông tin phiên bản của mô-đun này. |
| Return value: | None | |
| Description: | Trả về thông tin phiên bản của mô-đun này. | |

⌋ (SRS\_BSW\_00407, SRS\_BSW\_00003)

Để cấu hình Com\_GetVersionInfo, xem SWS\_Com\_00026.

##### Com\_ClearIpduGroupVector

[SWS\_Com\_00749] ⌈

|  |  |  |
| --- | --- | --- |
| Service name: | Com\_ClearIpduGroupVector | |
| Syntax: | void Com\_ClearIpduGroupVector( Com\_IpduGroupVector ipduGroupVector  ) | |
| Service ID[hex]: | 0x1c | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | None | |
| Parameters (in- out): | ipduGroupVector | Vector nhóm I-PDU cần được xóa. |
| Parameters (out): | None | |
| Return value: | None | |
| ***Description:*** | Dịch vụ này đặt tất cả các bit của Com\_IpduGroupVector được cho về 0. | |

⌋ (SRS\_Com\_02090)

##### Com\_SetIpduGroup

[SWS\_Com\_00750] ⌈

|  |  |  |
| --- | --- | --- |
| Service name: | Com\_SetIpduGroup | |
| Syntax: | void Com\_SetIpduGroup(  Com\_IpduGroupVector ipduGroupVector, Com\_IpduGroupIdType ipduGroupId, boolean bitval  ) | |
| Service ID[hex]: | 0x1d | |
| Sync/Async: | Synchronous | |
| Reentrancy: | Non Reentrant | |
| Parameters (in): | ipduGroupId | ipduGroup dùng để xác định bit tương ứng trong vector nhóm I-PDU |
| bitval | Giá trị mới của bit tương ứng |
| Parameters (in- out): | ipduGroupVector | Vector nhóm I-PDU cần được sửa đổi |
| Parameters (out): | None | |
| Return value: | None | |
| Description: | ịch vụ này đặt giá trị của một bit trong vector nhóm I-PDU. | |

⌋ (SRS\_Com\_02090)

SWS\_Com\_00623] ⌈Chức năng Com\_SetIpduGroup sẽ đặt bit của vector nhóm I-PDU tương ứng với nhóm I-PDU được cho, tức là bit thứ n cho nhóm I-PDU với ID n, thành bitval.⌋ (SRS\_Com\_02090)

#### Communication Services

##### Com\_SendSignal

**[SWS\_Com\_00197]** ⌈

|  |  |  |
| --- | --- | --- |
| Service name: | Com\_SendSignal | |
| Syntax: | uint8 Com\_SendSignal( Com\_SignalIdType SignalId, const void\* SignalDataPtr  ) | |
| Service ID[hex]: | 0x0a | |
| Sync/Async: | Asynchronous | |
| Reentrancy: | Không tái nhập đối với cùng một tín hiệu. Có thể tái nhập đối với các tín hiệu khác nhau. | |
| Parameters (in): | SignalId | D của tín hiệu cần gửi. |
| SignalDataPtr | Tham chiếu đến dữ liệu tín hiệu cần truyền. |
| Parameters (in- out): | None | |
| Parameters (out): | None | |
| Return value: | uint8 | E\_OK: dịch vụ đã được chấp nhận COM\_SERVICE\_NOT\_AVAILABLE: nhóm I-PDU tương ứng đã dừng (hoặc dịch vụ thất bại do lỗi phát triển) COM\_BUSY: trong trường hợp TP-Buffer bị khóa để xử lý các loại dữ liệu lớn |
| Description: | Dịch vụ Com\_SendSignal cập nhật đối tượng tín hiệu được xác định bởi SignalId với tín hiệu được tham chiếu bởi tham số SignalDataPtr. | |

⌋ (SRS\_Com\_02037)

[SWS\_Com\_00624] ⌈Dịch vụ Com\_SendSignal sẽ cập nhật đối tượng tín hiệu được xác định bởi SignalId với tín hiệu được tham chiếu bởi tham số SignalDataPtr.⌋ (SRS\_Com\_02037)

[SWS\_Com\_00625] ⌈Nếu tín hiệu được cập nhật có ComTransferProperty TRIGGERED và nó được gán cho một I-PDU với ComTxModeMode DIRECT hoặc MIXED, thì Com\_SendSignal sẽ thực hiện việc truyền ngay lập tức (chậm nhất là trong lần chạy chức năng chính tiếp theo) của I-PDU đó, trừ khi việc gửi bị trì hoãn hoặc ngăn cản bởi các cơ chế COM khác.⌋ (SRS\_Com\_02037)

Các cơ chế liên quan ví dụ như bộ hẹn giờ trì hoãn tối thiểu (ComMinimumDelayTime) hoặc độ trễ truyền (ComTxModeTimeOffset).

##### Com\_SendDynSignal

[SWS\_Com\_00627] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_SendDynSignal | |
| ***Syntax:*** | uint8 Com\_SendDynSignal( Com\_SignalIdType SignalId, const void\* SignalDataPtr, uint16 Length  ) | |
| ***Service ID[hex]:*** | 0x21 | |
| ***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. |
| Length | Length of the dynamic length signal |
| ***Parameters (in-***  ***out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | uint8 | E\_OK: dịch vụ đã được chấp nhận E\_NOT\_OK: trong trường hợp Độ dài lớn hơn ComSignalLength được cấu hình của tín hiệu gửi này COM\_SERVICE\_NOT\_AVAILABLE: nhóm I-PDU tương ứng đã dừng (hoặc dịch vụ thất bại do lỗi phát triển) COM\_BUSY: trong trường hợp TP-Buffer bị khóa |
| ***Description:*** | Dịch vụ Com\_SendDynSignal cập nhật đối tượng tín hiệu được xác định bởi SignalId với tín hiệu được tham chiếu bởi tham số SignalDataPtr. | |

⌋ (SRS\_Com\_02095)

[SWS\_Com\_00628] ⌈Dịch vụ Com\_SendDynSignal sẽ cập nhật đối tượng tín hiệu được xác định bởi SignalId với tín hiệu được tham chiếu bởi tham số SignalDataPtr.⌋ (SRS\_Com\_02095)

[SWS\_Com\_00629] ⌈Nếu tín hiệu được cập nhật (xem SWS\_Com\_00628) có ComTransferProperty TRIGGERED, thì chức năng Com\_SendDynSignal sẽ thực hiện việc truyền ngay lập tức (chậm nhất là trong lần chạy chức năng chính tiếp theo) của I-PDU liên kết với tín hiệu đó, ngoại trừ khi tín hiệu được đóng gói vào một I-PDU với chế độ truyền định kỳ.⌋ (SRS\_Com\_02095)

[SWS\_Com\_00630] ⌈Nếu tín hiệu được cập nhật, xem SWS\_Com\_00628, có ComTransferProperty PENDING, thì chức năng Com\_SendDynSignal sẽ không thực hiện việc truyền của I-PDU liên kết với tín hiệu đó.⌋ (SRS\_Com\_02095)

Không có giới hạn nào về độ dài tối thiểu của một tín hiệu có độ dài động. Do đó, tín hiệu có độ dài động được phép có độ dài bằng không.

##### Com\_ReceiveSignal

[SWS\_Com\_00198] ⌈

|  |  |  |
| --- | --- | --- |
| Service name: | Com\_ReceiveSignal | |
| Syntax: | uint8 Com\_ReceiveSignal( Com\_SignalIdType SignalId, void\* SignalDataPtr  ) | |
| 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 location where the received signal data shall be  stored |
| Return value: | uint8 | E\_OK: service has been accepted COM\_SERVICE\_NOT\_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error)  COM\_BUSY: in case the TP-Buffer is locked for large data types handling |
| Description: | Com\_ReceiveSignal copies the data of the signal identified by SignalId to the location specified by SignalDataPtr. | |

⌋ (SRS\_Com\_02037)

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

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

##### Com\_ReceiveDynSignal

[SWS\_Com\_00690] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_ReceiveDynSignal | |
| ***Syntax:*** | uint8 Com\_ReceiveDynSignal( Com\_SignalIdType SignalId, void\* SignalDataPtr,  uint16\* Length  ) | |
| ***Service ID[hex]:*** | 0x22 | |
| ***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):*** | Length | in: maximum length that could be received out: length of the dynamic length signal |
| ***Parameters (out):*** | SignalDataPtr | reference to the location where the received signal data shall be stored |
| ***Return value:*** | uint8 | E\_OK: service has been accepted  E\_NOT\_OK: in case the Length (as in-parameter) is smaller than the received length of the dynamic length signal COM\_SERVICE\_NOT\_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) COM\_BUSY: in case the TP-Buffer is locked |
| ***Description:*** | Com\_ReceiveDynSignal copies the data of the signal identified by SignalId to the location specified by SignalDataPtr and stores the length of the dynamical length signal at the position given by the Length parameter. | |

⌋ (SRS\_Com\_02092, SRS\_Com\_02095)

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

**[SWS\_Com\_00712]** ⌈The service Com\_ReceiveDynSignal shall return the calculated length (see [SWS\_Com\_00758](#_bookmark112)) of the dynamic length signal identified by SignalId in parameter Length.⌋ (SRS\_Com\_02092)

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

##### Com\_UpdateShadowSignal

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

[SWS\_Com\_00199] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_UpdateShadowSignal | |
| ***Syntax:*** | void Com\_UpdateShadowSignal( Com\_SignalIdType SignalId, const void\* SignalDataPtr  ) | |
| ***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. | |

⌋ (SRS\_Com\_02041)

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

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

**[SWS\_Com\_00634]** ⌈Configuration of Com\_UpdateShadowSignal: A signal group must be configured before this call. See [ECUC\_Com\_00345](#_bookmark279) for details.⌋ (SRS\_Com\_02041)

##### Com\_SendSignalGroup

[SWS\_Com\_00200] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_SendSignalGroup | |
| ***Syntax:*** | uint8 Com\_SendSignalGroup( Com\_SignalGroupIdType SignalGroupId  ) | |
| ***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 sent. |
| ***Parameters (in- out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | uint8 | E\_OK: service has been accepted COM\_SERVICE\_NOT\_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) COM\_BUSY: in case the TP-Buffer is locked for large data types  handling |
| ***Description:*** | The service Com\_SendSignalGroup copies the content of the associated shadow buffer to the associated I-PDU. | |

⌋ (SRS\_Com\_02041)

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

**[SWS\_Com\_00637]** ⌈Configuration of Com\_SendSignalGroup: A signal group must be configured before this call. See [ECUC\_Com\_00345](#_bookmark279) for details.⌋ (SRS\_Com\_02041)

##### Com\_ReceiveSignalGroup

[SWS\_Com\_00201] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_ReceiveSignalGroup | |
| ***Syntax:*** | uint8 Com\_ReceiveSignalGroup( Com\_SignalGroupIdType SignalGroupId  ) | |
| ***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 COM\_SERVICE\_NOT\_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error) COM\_BUSY: in case the TP-Buffer is locked for large data types  handling |
| ***Description:*** | The service Com\_ReceiveSignalGroup copies the received signal group from the I-PDU to the shadow buffer. | |

⌋ (SRS\_Com\_02041)

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

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

**[SWS\_Com\_00639]** ⌈Configuration of Com\_ReceiveSignalGroup: A signal group must be configured before this call. See [ECUC\_Com\_00345](#_bookmark279) for details.⌋ (SRS\_Com\_02041)

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

##### Com\_ReceiveShadowSignal

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

[SWS\_Com\_00202] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_ReceiveShadowSignal | |
| ***Syntax:*** | void Com\_ReceiveShadowSignal( Com\_SignalIdType SignalId, void\* SignalDataPtr  ) | |
| ***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 recei- ved data. |
| ***Return value:*** | None | |
| ***Description:*** | The service Com\_ReceiveShadowSignal updates the group signal which is refe- renced by SignalDataPtr with the data in the shadow buffer. | |

⌋ (SRS\_Com\_02041)

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

##### Com\_SendSignalGroupArray

[SWS\_Com\_00851]⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_SendSignalGroupArray | |
| ***Syntax:*** | uint8 Com\_SendSignalGroupArray( Com\_SignalGroupIdType SignalGroupId, const uint8\* SignalGroupArrayPtr  ) | |
| ***Service ID[hex]:*** | 0x23 | |
| ***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 sent. |
| SignalGroupArrayPtr | Reference to the signal group array. |
| ***Parameters (in-***  ***out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | uint8 | E\_OK: service has been accepted COM\_SERVICE\_NOT\_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error)  COM\_BUSY: in case the TP-Buffer is locked for large data types handling |
| ***Description:*** | The service Com\_SendSignalGroupArray copies the content of the provided Sig- nalGroupArrayPtr to the associated I-PDU. The provided data shall correspond to  the array representation of the signal group. | |

⌋(SRS\_Com\_02112, SRS\_Com\_02041)

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

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

##### Com\_ReceiveSignalGroupArray

[SWS\_Com\_00854]⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_ReceiveSignalGroupArray | |
| ***Syntax:*** | uint8 Com\_ReceiveSignalGroupArray( Com\_SignalGroupIdType SignalGroupId, uint8\* SignalGroupArrayPtr  ) | |
| ***Service ID[hex]:*** | 0x24 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non Reentrant for the same signal group. Reentrant for different signal groups. | |
| ***Parameters (in):*** | SignalGroupId | Id of signal group to be received. |
| ***Parameters (in-***  ***out):*** | None | |
| ***Parameters (out):*** | SignalGroupArrayPtr | reference to the location where the received signal group array shall be stored |
| ***Return value:*** | uint8 | E\_OK: service has been accepted  COM\_SERVICE\_NOT\_AVAILABLE: corresponding I-PDU group was stopped (or service failed due to development error)  COM\_BUSY: in case the TP-Buffer is locked for large data types  handling |
| ***Description:*** | The service Com\_ReceiveSignalGroupArray copies the received signal group  array representation from the PDU to the location designated by SignalGroupAr- rayPtr. | |

⌋(SRS\_Com\_02112, SRS\_Com\_02041)

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

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

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

##### Com\_InvalidateSignal

[SWS\_Com\_00203] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_InvalidateSignal | |
| ***Syntax:*** | uint8 Com\_InvalidateSignal( Com\_SignalIdType SignalId  ) | |
| ***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  COM\_SERVICE\_NOT\_AVAILABLE: corresponding I-PDU group is stop- ped, no ComSignalDataInvalidValue is configured for the given signalId or service fails due to development error  COM\_BUSY: in case the TP-Buffer is locked |
| ***Description:*** | The service Com\_InvalidateSignal invalidates the signal with the given SignalId by setting its value to its configured ComSignalDataInvalidValue. | |

⌋ (SRS\_Com\_02077)

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

**[SWS\_Com\_00642]** ⌈After invaliding the signal data, the function Com\_Invalidate- Signal shall perform a call of Com\_SendSignal internally, for details see [SWS\_Com\_00099](#_bookmark73).⌋ (SRS\_Com\_02077)

**[SWS\_Com\_00643]** ⌈Com\_InvalidateSignal shall return COM\_SERVICE\_NOT\_- AVAILABLE in case no ComSignalDataInvalidValue ([ECUC\_Com\_00391](#_bookmark270)) is config- ured for the signal with the given SignalId.⌋ (SRS\_Com\_02077)

##### Com\_InvalidateShadowSignal

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

[SWS\_Com\_00288] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_InvalidateShadowSignal | |
| ***Syntax:*** | void Com\_InvalidateShadowSignal( Com\_SignalIdType SignalId  ) | |
| ***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:*** | The service Com\_InvalidateShadowSignal invalidates the group signal with the given SignalId by setting its value to its configured ComSignalDataInvalidValue. | |

⌋ (SRS\_Com\_02077)

See Chapter [7.3.4](#_bookmark71) for further information about the signal invalidation mechanism.

**[SWS\_Com\_00644]** ⌈The AUTOSAR COM module shall only process Com\_InvalidateShadowSignal only if a ComSignalDataInvalidValue, see [ECUC\_Com\_00391](#_bookmark270), is configured for the signal with the given SignalId.⌋ (SRS\_Com\_02077)

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

The service Com\_InvalidateShadowSignal is included in the AUTOSAR COM mod- ule because of backwards compatibility reasons. In general, the function Com\_- InvalidateSignalGroup should be used instead.

##### Com\_InvalidateSignalGroup

[SWS\_Com\_00557] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_InvalidateSignalGroup | |
| ***Syntax:*** | uint8 Com\_InvalidateSignalGroup( Com\_SignalGroupIdType SignalGroupId  ) | |
| ***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 COM\_SERVICE\_NOT\_AVAILABLE: corresponding I-PDU group is stopped, no ComSignalDataInvalidValue is configured for any of the group signals or service fails due to development error  COM\_BUSY: in case the TP-Buffer is locked |
| ***Description:*** | The service Com\_InvalidateSignalGroup invalidates all group signals of the signal group with the given SignalGroupId by setting their values to their configured ComSignalDataInvalidValues. | |

⌋ (SRS\_Com\_02077)

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

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

##### Com\_TriggerIPDUSend

[SWS\_Com\_00348] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_TriggerIPDUSend | |
| ***Syntax:*** | Std\_ReturnType Com\_TriggerIPDUSend( PduIdType PduId  ) | |
| ***Service ID[hex]:*** | 0x17 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Non Reentrant | |
| ***Parameters (in):*** | PduId | The I-PDU-ID of the I-PDU that shall be triggered for sending |
| ***Parameters (in- out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | Std\_ReturnType | E\_OK: I-PDU was triggered for transmission  E\_NOT\_OK: I-PDU is stopped, the transmission could not be triggered |
| ***Description:*** | By a call to Com\_TriggerIPDUSend the I-PDU with the given ID is triggered for transmission. | |

⌋ (SRS\_Com\_02037)

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

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

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

**[SWS\_Com\_00492]** ⌈If an I-PDU triggered by Com\_TriggerIPDUSend or Com\_TriggerIPDUSendWithMetaData has a configured I-PDU-callout, see [ECUC\_Com\_00387](#_bookmark246), then the function Com\_TriggerIPDUSend or Com\_Trigger- IPDUSendWithMetaData repectively shall also call this I-PDU-Callout.⌋ (SRS\_Com\_02037)

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

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

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

##### Com\_TriggerIPDUSendWithMetaData

[SWS\_Com\_00858]⌈

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

⌋ (SRS\_Com\_02037)

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

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

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

##### Com\_SwitchIpduTxMode

[SWS\_Com\_00784] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_SwitchIpduTxMode | |
| ***Syntax:*** | void Com\_SwitchIpduTxMode( PduIdType PduId,  boolean Mode  ) | |
| ***Service ID[hex]:*** | 0x27 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant for different PduIds. Non reentrant for the same PduId. | |
| ***Parameters (in):*** | PduId | Id of the I-PDU of which the transmission mode shall be changed. |
| Mode | the transmission mode that shall be set. |
| ***Parameters (in- out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | The service Com\_SwitchIpduTxMode sets the transmission mode of the I-PDU referenced by PduId to Mode.  In case the transmission mode changes, the new mode shall immediately be effective (see SWS\_Com\_00239).  In case the requested transmission mode was already active for this I-PDU, the call will have no effect. | |

⌋ (SRS\_Com\_02082)

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

### Callback Functions and Notifications

#### Com\_TriggerTransmt

[SWS\_Com\_00001] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_TriggerTransmit | |
| ***Syntax:*** | Std\_ReturnType Com\_TriggerTransmit( PduIdType TxPduId,  PduInfoType\* PduInfoPtr  ) | |
| ***Service ID[hex]:*** | 0x41 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant for different PduIds. Non reentrant for the same PduId. | |
| ***Parameters (in):*** | TxPduId | ID of the SDU that is requested to be transmitted. |
| ***Parameters (in- out):*** | PduInfoPtr | Contains a pointer to a buffer (SduDataPtr) to where the SDU data shall be copied, and the available buffer size in SduLengh. 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 num- ber of copied bytes.  E\_NOT\_OK: No SDU data has been copied. PduInfoPtr must not  be used since it may contain a NULL pointer or point to invalid data. |
| ***Description:*** | Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by PduInfoPtr->SduLength.  If it fits, it shall copy its data into the buffer provided by PduInfoPtr->SduDataPtr and update the length of the actual copied data in PduInfoPtr->SduLength.  If not, it returns E\_NOT\_OK without changing PduInfoPtr. | |

⌋ (SRS\_Com\_02045)

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

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

**[SWS\_Com\_00475]** ⌈Com\_TriggerTransmit is not interfered by the I-PDU minimum delay time and shall not reset the minimum delay timer, see [ECUC\_Com\_00181](#_bookmark259).⌋ (SRS\_Com\_02045)

The function Com\_TriggerTransmit behaves independent of the configured transmis- sion mode.

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

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

**Caveats of Com\_TriggerTransmit:** This function might be called in interrupt con- text.

#### Com\_RxIndication [SWS\_Com\_00123] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_RxIndication | |
| ***Syntax:*** | void Com\_RxIndication( PduIdType RxPduId,  const PduInfoType\* PduInfoPtr  ) | |
| ***Service ID[hex]:*** | 0x42 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant for different PduIds. Non reentrant for the same PduId. | |
| ***Parameters (in):*** | RxPduId | ID of the received I-PDU. |
| 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:*** | Indication of a received I-PDU from a lower layer communication interface module. | |

⌋ (SRS\_Com\_02043)

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

#### Com\_TpRxIndication [SWS\_Com\_00650] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_TpRxIndication | |
| ***Syntax:*** | void Com\_TpRxIndication( PduIdType id, Std\_ReturnType result  ) | |
| ***Service ID[hex]:*** | 0x45 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant | |
| ***Parameters (in):*** | id | Identification of the received I-PDU. |
| result | Result of the reception. |
| ***Parameters (in- out):*** | None | |
| ***Parameters (out):*** | None | |
| ***Return value:*** | None | |
| ***Description:*** | Called after an I-PDU has been received via the TP API, the result indicates whe- ther the transmission was successful or not. | |

⌋ (SRS\_Com\_02095)

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

#### Com\_TxConfirmation [SWS\_Com\_00124] ⌈

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

⌋ (SRS\_Com\_02044)

**Caveats of Com\_TxConfirmation:** This function might be called in interrupt context,

e.g. from transmit interrupt.

#### Com\_TpTxConfirmation

[SWS\_Com\_00725] ⌈

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

⌋ (SRS\_Com\_02095)

#### Com\_StartOfReception

[SWS\_Com\_00691] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_StartOfReception | |
| ***Syntax:*** | BufReq\_ReturnType Com\_StartOfReception( PduIdType id,  const PduInfoType\* info, PduLengthType TpSduLength,  PduLengthType\* bufferSizePtr  ) | |
| ***Service ID[hex]:*** | 0x46 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant | |
| ***Parameters (in):*** | id | Identification of the I-PDU. |
| info | Pointer to a PduInfoType structure containing the payload data (without protocol information) and payload length of the first frame or single frame of a transport protocol I-PDU re- ception. Depending on the global parameter MetaDataLen- gth, additional bytes containing MetaData (e.g. the CAN ID) are appended after the payload data, increasing the length accordingly. If neither first/single frame data nor MetaData  are available, this parameter is set to NULL\_PTR. |
| TpSduLength | Total length of the N-SDU to be received. |
| ***Parameters (in-***  ***out):*** | None | |
| ***Parameters (out):*** | bufferSizePtr | Available receive buffer in the receiving module. This para- meter will be used to compute the Block Size (BS) in the transport protocol module. |
| ***Return value:*** | BufReq\_ReturnType | BUFREQ\_OK: Connection has been accepted. bufferSizePtr indicates the available receive buffer; reception is continued. If no buffer of the requested size is available, a receive buffer size of 0 shall be indicated by bufferSizePtr.  BUFREQ\_E\_NOT\_OK: Connection has been rejected; recep- tion is aborted. bufferSizePtr remains unchanged.  BUFREQ\_E\_OVFL: No buffer of the required length can be provided; reception is aborted. bufferSizePtr remains unchanged. |
| ***Description:*** | This function is called at the start of receiving an N-SDU. The N-SDU might be  fragmented into multiple N-PDUs (FF with one or more following CFs) or might consist of a single N-PDU (SF). | |

⌋ (SRS\_Com\_02095)

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

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

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

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

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

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

#### Com\_CopyRxData

[SWS\_Com\_00692] ⌈

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

⌋ (SRS\_Com\_02095)

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

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

#### Com\_CopyTxData

[SWS\_Com\_00693] ⌈

|  |  |  |
| --- | --- | --- |
| ***Service name:*** | Com\_CopyTxData | |
| ***Syntax:*** | BufReq\_ReturnType Com\_CopyTxData( PduIdType id,  const PduInfoType\* info, RetryInfoType\* retry, PduLengthType\* availableDataPtr  ) | |
| ***Service ID[hex]:*** | 0x43 | |
| ***Sync/Async:*** | Synchronous | |
| ***Reentrancy:*** | Reentrant | |
| ***Parameters (in):*** | id | Identification of the transmitted I-PDU. |
| info | Provides the destination buffer (SduDataPtr) and the number of bytes to be copied (SduLength).  If not enough transmit data is available, no data is copied by the upper layer module and BUFREQ\_E\_BUSY is returned. The lower layer module may retry the call.  An SduLength of 0 can be used to indicate state changes in the retry parameter or to query the current amount of available data in the upper layer module. In this case, the SduDataPtr may be a NULL\_PTR. |
| retry | This parameter is used to acknowledge transmitted data or to retransmit data after transmission problems.  If the retry parameter is a NULL\_PTR, it indicates that the transmit data can be removed from the buffer immediately after it has been copied. Otherwise, the retry parameter must point to a valid RetryInfoType element.  If TpDataState indicates TP\_CONFPENDING, the previously copied data must remain in the TP buffer to be available for error recovery.  TP\_DATACONF indicates that all data that has been copied before this call is confirmed and can be removed from the TP buffer. Data copied by this API call is excluded and will be confirmed later.  TP\_DATARETRY indicates that this API call shall copy previously copied data in order to recover from an error. In this case TxTpDataCnt specifies the offset in bytes from the  current data copy position. |
| ***Parameters (in-***  ***out):*** | None | |
| ***Parameters (out):*** | availableDataPtr | Indicates the remaining number of bytes that are available in the upper layer module's Tx buffer. availableDataPtr can be used by TP modules that support dynamic payload lengths (e.g. FrIsoTp) to determine the size of the following CFs. |
| ***Return value:*** | BufReq\_ReturnType | BUFREQ\_OK: Data has been copied to the transmit buffer completely as requested.  BUFREQ\_E\_BUSY: Request could not be fulfilled, because the required amount of Tx data is not available. The lower layer module may retry this call later on. No data has been copied.  BUFREQ\_E\_NOT\_OK: Data has not been copied. Request failed. |
| ***Description:*** | This function is called to acquire the transmit data of an I-PDU segment (N-PDU). Each call to this function provides the next part of the I-PDU data unless retry-  >TpDataState is TP\_DATARETRY. In this case the function restarts to copy the data beginning at the offset from the current position indicated by retry-  >TxTpDataCnt. The size of the remaining data is written to the position indicated by availableDataPtr. | |

⌋ (SRS\_Com\_02095)

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

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

### Scheduled Functions

#### Com\_MainFunctionRx

[SWS\_Com\_00398] ⌈

|  |  |
| --- | --- |
| ***Service name:*** | Com\_MainFunctionRx |
| ***Syntax:*** | void Com\_MainFunctionRx( void  ) |
| ***Service ID[hex]:*** | 0x18 |
| ***Description:*** | This function performs the processing of the AUTOSAR COM module's receive processing that are not directly handled within the COM's functions invoked by the PDU-R, for example Com\_RxIndication. |

⌋ (SRS\_BSW\_00425, SRS\_BSW\_00432)

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

Configuration of Com\_MainFunctionRx: see [ECUC\_Com\_00186](#_bookmark241).

#### Com\_MainFunctionTx

[SWS\_Com\_00399] ⌈

|  |  |
| --- | --- |
| ***Service name:*** | Com\_MainFunctionTx |
| ***Syntax:*** | void Com\_MainFunctionTx( void  ) |
| ***Service ID[hex]:*** | 0x19 |
| ***Description:*** | This function performs the processing of the AUTOSAR COM module's transmis- sion activities that are not directly handled within the COM's function invoked by the RTE, for example Com\_SendSignal. |

⌋ (SRS\_BSW\_00425, SRS\_BSW\_00432)

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

**Configuration:** see [ECUC\_Com\_00186](#_bookmark241).

#### Com\_MainFunctionRouteSignals

[SWS\_Com\_00400] ⌈

|  |  |
| --- | --- |
| ***Service name:*** | Com\_MainFunctionRouteSignals |
| ***Syntax:*** | void Com\_MainFunctionRouteSignals( void  ) |
| ***Service ID[hex]:*** | 0x1a |
| ***Description:*** | Calls the signal gateway part of the AUTOSAR COM module to forward received signals to be routed. |

⌋ (SRS\_BSW\_00425, SRS\_BSW\_00432)

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

Caveats of Com\_MainFunctionRouteSignals:

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

### Expected Interfaces

#### Mandatory Interfaces

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

[SWS\_Com\_00669] ⌈

|  |  |
| --- | --- |
| ***API function*** | ***Description*** |
| PduR\_ComTransmit | Requests transmission of an I-PDU. |

⌋ (SRS\_BSW\_00384)

#### Optional Interfaces

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

[SWS\_Com\_00670] ⌈

|  |  |
| --- | --- |
| ***API function*** | ***Description*** |
| Det\_ReportError | Service to report development errors. |
| Det\_ReportRuntimeError | Service to report runtime errors. If a callout has been configured then this callout shall be called. |
| PduR\_ComCancelTransmit | Requests cancellation of an ongoing transmission of an I-PDU in a lower layer communication interface or transport protocol module. |

⌋ (SRS\_BSW\_00384)

The associated configuration parameter for API function Det\_ReportError is ComConfigurationUseDet ([ECUC\_Com\_00141](#_bookmark236)).

The API function PduR\_ComCancelTransmit must be included if at least one I-PDU has configured ComIPduCancellationSupport to TRUE, see [ECUC\_Com\_00709](#_bookmark247).

#### Configurable Interfaces

##### Callback/ Notification Functions

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

[SWS\_Com\_00468] ⌈

|  |  |
| --- | --- |
| ***Service name:*** | Com\_CbkTxAck |
| ***Syntax:*** | void Com\_CbkTxAck( void  ) |
| ***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. |

⌋ (SRS\_BSW\_00359, SRS\_BSW\_00360)

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

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

The callback function name must be configured by ComNotification ([ECUC\_Com\_00498](#_bookmark268)).

[SWS\_Com\_00491] ⌈

|  |  |
| --- | --- |
| ***Service name:*** | Com\_CbkTxErr |
| ***Syntax:*** | void Com\_CbkTxErr( void  ) |
| ***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. |

⌋ (SRS\_BSW\_00359, SRS\_BSW\_00360)

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

The callback function name must be configured by ComErrorNotification ([ECUC\_Com\_00499](#_bookmark265)).

[SWS\_Com\_00554] ⌈

|  |  |
| --- | --- |
| ***Service name:*** | Com\_CbkTxTOut |
| ***Syntax:*** | void Com\_CbkTxTOut( void  ) |
| ***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. |

⌋ (SRS\_BSW\_00359, SRS\_BSW\_00360)

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

This callback function corresponds to Rte\_COMCbkTxTOut\_<sn> or Rte\_COMCbk- TxTOut\_<sg> respectively.

The callback function name must be configured by ComTimeoutNotification ([ECUC\_Com\_00552](#_bookmark276)).

[SWS\_Com\_00555] ⌈

|  |  |
| --- | --- |
| ***Service name:*** | Com\_CbkRxAck |
| ***Syntax:*** | void Com\_CbkRxAck( void  ) |
| ***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. |

⌋ (SRS\_BSW\_00359, SRS\_BSW\_00360)

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

Com\_CbkRxAck corresponds to Rte\_COMCbk\_<sn> or Rte\_COMCbk\_<sg> respec- tively.

The callback function name must be configured by ComNotification ([ECUC\_Com\_00498](#_bookmark268)).

[SWS\_Com\_00556] ⌈

|  |  |
| --- | --- |
| ***Service name:*** | Com\_CbkRxTOut |
| ***Syntax:*** | void Com\_CbkRxTOut( void  ) |
| ***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. |

⌋ (SRS\_BSW\_00359, SRS\_BSW\_00360)

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

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

The callback function name must be configured by ComTimeoutNotification (see [ECUC\_Com\_00552](#_bookmark276)).

[SWS\_Com\_00536] ⌈

|  |  |
| --- | --- |
| ***Service name:*** | Com\_CbkInv |
| ***Syntax:*** | void Com\_CbkInv( void  ) |
| ***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 SWS\_Com\_00680. It is called after recepti- on of an invalid signal or signal group respectively. |

⌋ (SRS\_BSW\_00359, SRS\_BSW\_00360, SRS\_Com\_02079)

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

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

The callback function name must be configured by ComInvalidNotification ([ECUC\_Com\_00315](#_bookmark267)).

[SWS\_Com\_00726] ⌈

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

⌋ (SRS\_Com\_02102)

Com\_CbkCounterErr is called on receiver side only. It can be configured per I-PDU counter. The callback function name must be configured by ComIPduCounterError- Notification ([ECUC\_Com\_00003](#_bookmark253)).

#### I-PDU Callout Functions

As stated in [SWS\_Com\_00013](#_bookmark34) *Network-order message callout* and *CPU-order mes- sage callout* are not supported by the AUTOSAR COM module. The only callout method supported by the AUTOSAR COM module therefore is the I-PDU callout. The AUTOSAR COM module supports I-PDU callouts on sender and on receiver side.

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

#### [SWS\_Com\_00700] ⌈

|  |  |
| --- | --- |
| ***Service name:*** | Com\_RxIpduCallout |
| ***Syntax:*** | boolean Com\_RxIpduCallout( PduIdType PduId,  const PduInfoType\* PduInfoPtr  ) |
| ***Sync/Async:*** | Synchronous |
| ***Reentrancy:*** | don't care |

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

⌋ (SRS\_Com\_02037)

The callout function name must be configured by ComIPduCallout ([ECUC\_Com\_00387](#_bookmark246)).

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

#### [SWS\_Com\_00346] ⌈

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

⌋ (SRS\_Com\_02037)

The callout function name must be configured either by ComIPduCallout ([ECUC\_Com\_00387](#_bookmark246)) or by ComIPduTriggerTransmitCallout ([ECUC\_Com\_00765](#_bookmark249)).

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

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

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

callout within every execution of Com\_TriggerTransmit for this I-PDU.⌋ (SRS\_Com\_02037)

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

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

For TP I-PDUs, the I-PDU callout functions are handled similar to the I-PDU callout functions of normal I-PDUs and are supported both on transmitter and receiver side. Note that for example the requirement [SWS\_Com\_00719](#_bookmark221) does not consider the con- figured ComIPduType.

**[SWS\_Com\_00720]** ⌈When a large I-PDU is received successfully via the TP inter- face, the I-PDU callout shall take place within the invocation of Com\_TpRxIndication.⌋ (SRS\_Com\_02095)

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

## Sequence Diagrams

A sequence diagram of the underlying OSEK COM communication stack can be found in [[17]](#_bookmark17).

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

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



seq Reception CAN, LIN, FlexRay

Com\_RxIndication(PduIdType, PduInfoType\*)

Internal processing()

seq Transmission LIN, FlexRay

PduR\_ComTransmit(Std\_ReturnType, PduIdType, const PduInfoType\*) : Std\_ReturnType

internal processing()

E\_OK()

Com\_TriggerTransmit(Std\_ReturnType, PduIdType, PduInfoType\*)

internal processing()

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()

«module» PduR

«module» Com

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

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

The following chart shows the confirmation handling with respect to the two different IPDU-processing modes. (See also Chapter [7.3.5.4](#_bookmark83).)



[Unpack.Def

Com\_TxConfirmation(PduIdType)

set

flag()

check flag()

alt check flag

[flag set]

Rte\_COMCbkTAck\_signalX()

internal processing()

Rte\_COMCbkTAck\_signalY()

internal processing()

[flag not set]

erred]

Com\_TxConfirmation(PduIdType)

internal processing()

Rte\_COMCbkTAck\_signalY()

internal processing()

Rte\_COMCbkTAck\_signalX()

g

ediate]

alt Unpackin [Unpack.Imm

«module» PduR

«module» Com

«module» Rte

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

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

The following chart shows the indication handling with respect to the two different unpacking modes. (See also Chapter [7.3.5.4](#_bookmark83).)



ng

mediate]

Rte\_COMCbk\_signalX()

internal processing()

Rte\_COMCbk\_signalY()

internal processing()

[Unpack.D

Com\_RxIndication(PduIdType, PduInfoType\*)

set flag, copy data()

check flag()

ck flag

et]

Rte\_COMCbk\_signalX()

internal processing()

Rte\_COMCbk\_signalY()

internal processing()

ot set]

[flag n

alt che [flag s

eferred]

Com\_RxIndication(PduIdType, PduInfoType\*)

alt Unpacki [Unpack.Im

«module» PduR

«module» Com

«module» Rte

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

## Configuration Specification

### Containers and Configuration Parameters

The following chapters summarize all configuration parameters. The detailed mean- ings of the parameters are described in Chapter [7](#_bookmark30) and Chapter [8](#_bookmark144).

**[SWS\_Com\_00006]** ⌈The configuration parameters are based on [[18]](#_bookmark18). All parame- ters shall be stored in an XML format.⌋(SRS\_Com\_02040)

#### Variants

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

##### VARIANT-PRE-COMPILE

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

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

##### VARIANT-LINK-TIME

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

##### VARIANT-POST-BUILD

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

#### Configuration of the AUTOSAR COM Module

For an overview of the AUTOSAR COM module’s configuration, see [Figure 11](#_bookmark233).

Com :EcucModuleDef upperMultiplicity = 1

lowerMultiplicity = 0

ComGeneral :

+container EcucParamConfContainerDef



upperMultiplicity = 1

lowerMultiplicity = 1

+container

ComConfig :EcucParamConfContainerDef

multipleConfigurationContainer = true

+subContainer

ComIPdu :

+subContainer

ComSignalGroup :

+subContainer

ComSignal :

+subContainer

ComIPduGroup :

+destination EcucParamConfContainerDef +destination EcucParamConfContainerDef

EcucParamConfContainerDef

EcucParamConfContainerDef

upperMultiplicity = \* lowerMultiplicity = 0

+subContainer

upperMultiplicity = \* lowerMultiplicity = 0

+subContainer

+destination

upperMultiplicity = \* lowerMultiplicity = 0

upperMultiplicity = \* lowerMultiplicity = 0

+subContainer

ComTimeBase :EcucParamConfContainerDef

ComTxIPdu : EcucParamConfContainerDef

ComGroupSignal : EcucParamConfContainerDef

upperMultiplicity = 1

lowerMultiplicity = 0

upperMultiplicity = \* lowerMultiplicity = 0

+destination

+parameter

ComGwTimeBase :

+parameter ComRxTimeBase :

+subContainer

ComTxModeTrue :

+subContainer

ComTxModeFalse :

+subContainer

ComFilter :

+subContainer

EcucFloatParamDef

min = 0

EcucFloatParamDef

min = 0

EcucParamConfContainerDef

upperMultiplicity = 1

EcucParamConfContainerDef

upperMultiplicity = 1

EcucParamConfContainerDef

upperMultiplicity = 1

max = 3600

lowerMultiplicity = 0

upperMultiplicity = 1

max = 3600

lowerMultiplicity = 0

upperMultiplicity = 1

lowerMultiplicity = 0

lowerMultiplicity = 0

lowerMultiplicity = 0

+subContainer

+subContainer

+parameter

+subContainer

ComTxMode : EcucParamConfContainerDef

+subContainer

ComGwMapping :EcucParamConfContainerDef

upperMultiplicity = \* lowerMultiplicity = 0

ComTxTimeBase :

EcucFloatParamDef

min = 0

max = 3600

lowerMultiplicity = 0

ComGwSignalRef : EcucChoiceReferenceDef

+subContainer

ComGwSource : EcucChoiceContainerDef

+subContainer

ComGwDestination : EcucChoiceContainerDef

upperMultiplicity = 1

ComGwIPduRef :EcucReferenceDef

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

#### Com

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00540 :** |
| ***Module Name*** | *Com* |
| ***Module Description*** | Configuration of the AUTOSAR COM module. |
| ***Post-Build Variant Support*** | true |

|  |  |  |
| --- | --- | --- |
| ***Included Containers*** | | |
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComConfig | 1 | This container contains the configuration parameters and sub containers of the AUTOSAR COM module. This container is a MultipleConfigurationContainer, i.e. this container and its sub-  containers exist once per configuration set. |
| ComGeneral | 1 | Contains the general configuration parameters of the AUTO- SAR COM module. |

#### ComGeneral

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00541 :** |
| ***Container Name*** | ComGeneral |
| ***Description*** | Contains the general configuration parameters of the module. |
| ***Configuration Parameters*** | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_10000 :** | | |
| ***Name*** | ComCancellationSupport | | |
| ***Description*** | This parameter enables/disables the cancellation feature:  true: enabled false: disabled | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | true | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00141 :** |
| ***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). |
| ***Multiplicity*** | 0..1 |
| ***Type*** | EcucBooleanParamDef |
| ***Default value*** | -- |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Post-Build Variant Multipli- city*** | false |  | |
| ***Post-Build Variant Value*** | false |  | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00788 :** | | |
| ***Name*** | ComEnableMDTForCyclicTransmission | | |
| ***Description*** | Enables globally for the whole Com module the minimum delay time moni- toring for cyclic and repeated transmissions (ComTxModeMo-  de=PERIODIC or ComTxModeMode=MIXED for the cyclic transmissions, ComTxModeNumberOfRepetitions > 0 for repeated transmissions). | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | true | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_10002 :** | | |
| ***Name*** | ComEnableSignalGroupArrayApi | | |
| ***Description*** | Activate/Deactivate the signal group array access APIs (Com\_SendSignalGroupArray, Com\_ReceiveSignalGroupArray).  true: signal group array access APIs activated false: signal group array access APIs deactivated | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_10004 :** |
| ***Name*** | ComMetaDataSupport |
| ***Description*** | This parameter enables/disables the support of meta-data feature inclu- ding the API Com\_TriggerIPDUSendWithMetaData.  true: enabled false: disabled |
| ***Multiplicity*** | 0..1 |
| ***Type*** | EcucBooleanParamDef |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Default value*** | false | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00780 :** | | |
| ***Name*** | ComRetryFailedTransmitRequests | | |
| ***Description*** | If this Parameter is set to true, retry of failed transmission requests is enabled. If this Parameter is not present, the default value is assumed. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | false | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00710 :** | | |
| ***Name*** | ComSupportedIPduGroups | | |
| ***Description*** | Defines the maximum number of supported I-PDU groups. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 65535 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Value*** | false | | |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_10001 :** |
| ***Name*** | ComUserCbkHeaderFile |
| ***Description*** | Defines the header files for callback functions which shall be included by the COM module. |
| ***Multiplicity*** | 0..\* |
| ***Type*** | EcucStringParamDef |
| ***Default value*** | -- |
| ***maxLength*** | 32 |
| ***minLength*** | 1 |
| ***regularExpression*** | -- |
| ***Post-Build Variant Multipli- city*** | false |
| ***Post-Build Variant Value*** | false |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local |  | |

***No Included Containers***

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00438 :** | | |
| ***Name*** | ComVersionInfoApi | | |
| ***Description*** | Activate/Deactivate the version information API (Com\_GetVersionInfo).  True: version information API activated False: version information API deactivated | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Value*** | false | | |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local | | |

#### ComConfig

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00337 :** |
| ***Container Name*** | ComConfig |
| ***Description*** | This container contains the configuration parameters and sub containers of the AUTOSAR COM module. |
| ***Configuration Parameters*** | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00783 :** | | |
| ***Name*** | ComDataMemSize | | |
| ***Description*** | Size of internal Com data in units of bytes (static memory allocation) - memory required by post-build configuration must be smaller than this constant. This parameter is needed only in case of post-build loadable implementation using static memory allocation. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 ..  18446744073709551615 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00782 :** | | |
| ***Name*** | ComMaxIPduCnt | | |
| ***Description*** | Maximum number of IPdus. This parameter is needed only in case of post- build loadable implementation using static memory allocation. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 ..  18446744073709551615 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** |
| ComGwMapping | 0..\* | Each instance of this container defines one mapping of the integrated Signal Gateway. |
| ComIPdu | 0..\* | See ECUC\_Com\_00340, if there is no ComIpdu container included no IPDUs are defined. In this case no communication  via the AUTOSAR COM module is possible. |
| ComIPduGroup | 0..\* | See ECUC\_Com\_00341, if there is no ComIPduGroup contai- ner included then no IPDU group is defined. In this case no communication via the AUTOSAR COM module is possible. |
| ComSignal | 0..\* | See ECUC\_Com\_00344, if there is no ComSignal container included no single signals are defined. |
| ComSignalGroup | 0..\* | See ECUC\_Com\_00345, if there is no ComSignalGroup con- tainer included no signal groups are defined. |
| ComTimeBase | 1 | Contains the timebase parameters for Tx, Rx and routing. |

#### ComTimeBase

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00730 :** |
| ***Container Name*** | ComTimeBase |
| ***Description*** | Contains the timebase parameters for Tx, Rx and routing. |
| ***Configuration Parameters*** | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00729 :** |
| ***Name*** | ComGwTimeBase |
| ***Description*** | The period between successive calls to Com\_MainFunctionRouteSignals |

|  |  |  |  |
| --- | --- | --- | --- |
|  | in seconds. This parameter may be used by the COM generator to trans- form the values of the signal gateway related timing configuration parame- ters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific.  The COM module (generator) might rely on the fact that Com\_MainFunctionRouteSignals is scheduled according to the value configured here. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFloatParamDef | | |
| ***Range*** | 0 .. 3600 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli-***  ***city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: ECU | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00728 :** | | |
| ***Name*** | ComRxTimeBase | | |
| ***Description*** | The period between successive calls to Com\_MainFunctionRx in seconds. This parameter may be used by the COM generator to transform the valu- es of the reception related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific.  The COM module (generator) may rely on the fact that Com\_MainFunctionRx is scheduled according to the value configured here. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFloatParamDef | | |
| ***Range*** | 0 .. 3600 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: ECU | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00186 :** |
| ***Name*** | ComTxTimeBase |
| ***Description*** | The period between successive calls to Com\_MainFunctionTx in seconds. |

|  |  |  |  |
| --- | --- | --- | --- |
|  | This parameter may be used by the COM generator to transform the valu- es of the transmission related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific.  The COM module (generator) may rely on the fact that Com\_MainFunctionTx is scheduled according to the value configured here. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFloatParamDef | | |
| ***Range*** | 0 .. 3600 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: ECU | | |

***No Included Containers***

#### ComFilter

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00339 :** |
| ***Container Name*** | ComFilter |
| ***Description*** | This container contains the configuration parameters of the AUTOSAR COM module's Filters.  Note: On sender side the container is used to specify the transmission mode conditions. |
| ***Configuration Parameters*** | |

|  |  |  |
| --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00146 :** | |
| ***Name*** | ComFilterAlgorithm | |
| ***Description*** | The range of values is specified in the [17] specification, chapter 2.2.2, Reception Filtering. | |
| ***Multiplicity*** | 1 | |
| ***Type*** | EcucEnumerationParamDef | |
| ***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 | -- |
| ***Post-Build Vari-*** | false | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***ant Value*** |  |  | |
| ***Value Configura- tion Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Depen- dency*** | scope: local |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00235 :** | | |
| ***Name*** | ComFilterMask | | |
| ***Description*** | The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | -2147483648 ..  4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00317 :** | | |
| ***Name*** | ComFilterMax | | |
| ***Description*** | The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | -2147483648 ..  4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00318 :** |
| ***Name*** | ComFilterMin |
| ***Description*** | The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Multiplicity*** | 0..1 |  | |
| ***Type*** | EcucIntegerParamDef |  | |
| ***Range*** | -2147483648 ..  4294967295 |  | |
| ***Default value*** | -- |  | |
| ***Post-Build Variant Multipli- city*** | false |  | |
| ***Post-Build Variant Value*** | false |  | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00313 :** | | |
| ***Name*** | ComFilterOffset | | |
| ***Description*** | The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.  Range = 0..(ComFilterPeriod-1) | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 4294967294 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local  dependency: ECUC\_Com\_00312 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00312 :** | | |
| ***Name*** | ComFilterPeriod | | |
| ***Description*** | This parameter defines the period of the ComFilterAlgorithm O- NE\_EVERY\_N. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 2 .. 4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00147 :** | | |
| ***Name*** | ComFilterX | | |
| ***Description*** | The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | -2147483648 ..  4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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***

#### ComIPdu

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00340 :** | | |
| ***Container Name*** | ComIPdu | | |
| ***Description*** | Contains the configuration parameters of the AUTOSAR COM module's I- PDUs. | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Configuration Parameters*** | | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00387 :** |
| ***Name*** | ComIPduCallout |
| ***Description*** | This parameter defines the existence and the name of a callout function for the corresponding I-PDU. If this parameter is omitted no I-PDU callout shall take place for the corresponding I-PDU. |
| ***Multiplicity*** | 0..1 |
| ***Type*** | EcucFunctionNameDef |
| ***Default value*** | -- |
| ***maxLength*** | -- |
| ***minLength*** | -- |
| ***regularExpression*** | -- |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Post-Build Variant Multipli- city*** | false |  | |
| ***Post-Build Variant Value*** | false |  | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00709 :** | | |
| ***Name*** | ComIPduCancellationSupport | | |
| ***Description*** | Defines for I-PDUs with ComIPduType NORMAL: If the underlying IF- modul supports cancellation of transmit requests.  Defines for I-PDUs with ComIPduType TP: If the underlying TP-module supports RX and TX cancellation of ongoing requests. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | false | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | -- |  |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | -- |  |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: ECU  dependency: This parameter shall not be set to true if ComCancellation- Support is set to false | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00493 :** | | |
| ***Name*** | ComIPduDirection | | |
| ***Description*** | The direction defines if this I-PDU, and therefore the contributing signals and signal groups, shall be sent or received. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | RECEIVE | -- | |
| SEND | -- | |
| ***Post-Build Variant Value*** | false | | |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT- POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Depen- dency*** | scope: local  dependency: If configured to Sent also a ComTxIpdu container shall be included, see ECUC\_Com\_00496 | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00175 :** |
| ***Name*** | ComIPduHandleId |
| ***Description*** | The numerical value used as the ID of this I-PDU. The ComIPduHandleId is required by the API calls Com\_RxIndication, Com\_TpRxIndication, |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Com\_StartOfReception and Com\_CopyRxData to receive I-PDUs from the PduR (ComIP-duDirection: Receive), as well as the PduId passed to an Rx-I-PDU-callout. For Tx-I-PDUs (ComIPduDirection: Send), this handle Id is used for the APIs calls Com\_TxConfirmation, Com\_TriggerTransmit, Com\_TriggerIPDUSend or Com\_TriggerIPDUSendWithMetaData, Com\_CopyTxData and Com\_TpTxConfirmation to transmit respectively confirm transmissions of I-PDUs, as well as the PduId passed to the Tx-I-  PDU-callout configured with ComIPduCallout and/or ComIPduTrigger- TransmitCallout. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | | |
| ***Range*** | 0 .. 65535 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: ECU | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00119 :** |  | |
| ***Name*** | ComIPduSignalProcessing |  | |
| ***Description*** | For the definition of the two modes Imm | ediate and Deferred. | |
| ***Multiplicity*** | 1 |  | |
| ***Type*** | EcucEnumerationParamDef |  | |
| ***Range*** | DEFERRED | signal indication / confirmations are defer- red for example to a cyclic task | |
| IMMEDIATE | the signal indications / confirmations are performed in Com\_RxIndication/  Com\_TxConfirmation | |
| ***Post-Build Variant Value*** | true |  | |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00765 :** | | |
| ***Name*** | ComIPduTriggerTransmitCallout | | |
| ***Description*** | If there is a trigger transmit callout defined for this I-PDU this parameter contains the name of the callout function. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFunctionNameDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00761 :** | | |
| ***Name*** | ComIPduType | | |
| ***Description*** | Defines if this I-PDU is a normal I-PDU that can be sent unfragmented or if this is a large I-PDU that shall be sent via the Transport Protocol of the underlying bus. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | NORMAL | sent or received via normal L-PDU | |
| TP | sent or received via TP | |
| ***Post-Build Variant Value*** | true | | |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00206 :** | | |
| ***Name*** | ComIPduGroupRef | | |
| ***Description*** | Reference to the I-PDU groups this I-PDU belongs to. | | |
| ***Multiplicity*** | 0..\* | | |
| ***Type*** | Reference to [ ComIPduGroup ] | | |
| ***Post-Build Variant Multipli-***  ***city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00519 :** | | |
| ***Name*** | ComIPduSignalGroupRef | | |
| ***Description*** | References to all signal groups contained in this I-Pdu | | |
| ***Multiplicity*** | 0..\* | | |
| ***Type*** | Reference to [ ComSignalGroup ] | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00518 :** |
| ***Name*** | ComIPduSignalRef |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Description*** | References to all signals contained in this I-PDU. | | |
| ***Multiplicity*** | 0..\* | | |
| ***Type*** | Reference to [ ComSignal ] | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00711 :** | | |
| ***Name*** | ComPduIdRef | | |
| ***Description*** | Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | Reference to [ Pdu ] | | |
| ***Post-Build Variant Value*** | false | | |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** |  | | |

|  |  |  |
| --- | --- | --- |
| ***Included Containers*** | | |
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComIPduCounter | 0..1 | This optional container contains the configuration parameters of PDU Counter. |
| ComIPduReplication | 0..1 | This optional container contains the information needed for each I-PDU replicated. |
| ComTxIPdu | 0..1 | This container must be included if COM\_IPDU\_DIRECTION is configured to SEND. |

**[SWS\_Com\_00497]** ⌈A ComTxIPdu container must be included if ComIPduDirection is configured to SEND.⌋ (SRS\_BSW\_00167)

#### ComIPduCounter

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00592 :** |
| ***Container Name*** | ComIPduCounter |
| ***Description*** | This optional container contains the configuration parameters of PDU Counter. |
| ***Configuration Parameters*** | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00003 :** |
| ***Name*** | ComIPduCounterErrorNotification |
| ***Description*** | Name of Com\_CbkCounterErr callback function to be called. If this  parameter is omitted no I-PDU counter mismatch notification shall take place. |
| ***Multiplicity*** | 0..1 |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Type*** | EcucFunctionNameDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00593 :** | | |
| ***Name*** | ComIPduCounterSize | | |
| ***Description*** | Size of PDU Counter expressed in bits | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 1 .. 8 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Value*** | false | | |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00594 :** | | |
| ***Name*** | ComIPduCounterStartPosition | | |
| ***Description*** | Position of PDU counter expressed in bits from start position of data con- tent  of I-PDU (SDU). Note that PDU counter is not allowed to cross a byte border. The parameter ComIPduCounterStartPosition shall define the bit0  of the first byte like in little endian byte order. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Value*** | false | | |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local | | |

|  |  |  |
| --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00595 :** | |
| ***Name*** | ComIPduCounterThreshold | |
| ***Description*** | Threshold value of I-PDU counter algorithm, see ECUC\_Com\_00590. | |
| ***Multiplicity*** | 0..1 | |
| ***Type*** | EcucIntegerParamDef | |
| ***Range*** | 0 .. 255 |  |
| ***Default value*** | 0 | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local | | |

***No Included Containers***

#### ComIPduReplication

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00599 :** |
| ***Container Name*** | ComIPduReplication |
| ***Description*** | This optional container contains the information needed for each I-PDU replicated. |
| ***Configuration Parameters*** | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00600 :** | | |
| ***Name*** | ComIPduReplicationQuorum | | |
| ***Description*** | The number of identical I-PDUs needed for successful voting. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 1 .. 3 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Value*** | false | | |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local  dependency: This value must be less or equal to the multiplicity of ( Co- mIPduReplicaRef + 1). | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00601 :** | | |
| ***Name*** | ComIPduReplicaRef | | |
| ***Description*** | Reference to replicas PduR PDUs of this IPDU. | | |
| ***Multiplicity*** | 1..2 | | |
| ***Type*** | Reference to [ Pdu ] | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: ECU | | |

***No Included Containers***

#### ComTxIPdu

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00496 :** |
| ***Container Name*** | ComTxIPdu |
| ***Description*** | This container contains additional transmission related configuration para- meters of the AUTOSAR COM module's I-PDUs. |
| ***Configuration Parameters*** | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_10005 :** | | |
| ***Name*** | ComMetaDataDefault | | |
| ***Description*** | In case an I-PDU has a globally configured MetaDataLength distinct from 0 and no explicit meta data is given for a send request (e.g. by using Com\_TriggerIPDUSendWithMetaData ), The AUTOSAR COM module uses this configured default meta data for sending.  The configured string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is  in byte 2 and (highest address). | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucStringParamDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local  dependency: This parameter shall only be used in case ComMetaDa- taSupport is enabled. The length of the default meta data shall match the globally configured MetaDataLength of this I-PDU. | | |

|  |  |  |
| --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00181 :** | |
| ***Name*** | ComMinimumDelayTime | |
| ***Description*** | Defines the Minimum Delay Time (MDT) between successive transmissi- ons of this I-PDU in seconds. The MDT is independent of the possible different transmission modes. There is only one minimum delay time pa- rameter for one I-PDU. The minimum delay timer is not reset by changing the transmission mode. Hence, it is not allowed to violate the minimum delay time by transmission mode changes. It is not possible to monitor the  minimum delay time for I-PDUs that are requested using the Com\_TriggerTransmit API. | |
| ***Multiplicity*** | 0..1 | |
| ***Type*** | EcucFloatParamDef | |
| ***Range*** | 0 .. 3600 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Default value*** | -- |  | |
| ***Post-Build Variant Multipli- city*** | true |  | |
| ***Post-Build Variant Value*** | true |  | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00576 :** |  | |
| ***Name*** | ComTxIPduClearUpdateBit |  | |
| ***Description*** | Defines when the update-bits of signals will be cleared. | or signal groups, contained in this I-PDU, | |
| ***Multiplicity*** | 0..1 |  | |
| ***Type*** | EcucEnumerationParamDef |  | |
| ***Range*** | Confirmation | The update-bits are cleared when the transmission of the I-PDU was confirmed. In case of Direct/N-Times transmission mode the update bits will be cleared with  respect to the confirmation behaviour of ECUC\_Com\_00305. | |
| Transmit | The update-bits are cleared directly after the invocation of PduR\_ComTransmit. | |
| TriggerTransmit | The update-bits are cleared after the I-PDU was fetched via Com\_TriggerTransmit. | |
| ***Post-Build Variant Multiplicity*** | true |  | |
| ***Post-Build Variant Value*** | true |  | |
| ***Multiplicity Confi- guration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00017 :** | | |
| ***Name*** | ComTxIPduUnusedAreasDefault | | |
| ***Description*** | The AUTOSAR COM module fills not used areas of an I-PDU with this byte pattern. This attribute is mandatory to avoid undefined behaviour. This  byte-pattern will be repeated throughout the I-PDU before any init-values or update-bits were set. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 255 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Value*** | false | | |
| ***Value Configuration Class*** | ***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 | The referenced transmission mode object that is used when the filtering state for this I-PDU evaluates to false. The default is transmission mode None. |
| ComTxModeTrue | 0..1 | The referenced transmission mode object that is used when the filtering state for this I-PDU evaluates to true. |

#### ComIPduGroup

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00341 :** | | |
| ***Container Name*** | ComIPduGroup | | |
| ***Description*** | Contains the configuration parameters of the AUTOSAR COM module's I- PDU groups. | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Configuration Parameters*** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00184 :** | | |
| ***Name*** | ComIPduGroupHandleId | | |
| ***Description*** | The numerical value used as the ID of this I-PDU Group .  The ComIPduGroupHandleId is required by the API calls to start and stop I-PDU Groups.  Range: 0 .. (ComSupportedIPduGroups-1) | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | | |
| ***Range*** | 0 .. 65535 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Value*** | false | | |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: ECU | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00185 :** | | |
| ***Name*** | ComIPduGroupGroupRef | | |
| ***Description*** | References to all I-PDU groups that includes this I-PDU group. If this refe- rence is omitted this I-PDU group does not belong to another I-PDU group. | | |
| ***Multiplicity*** | 0..\* | | |
| ***Type*** | Reference to [ ComIPduGroup ] | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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***

#### ComSignal

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00344 :** | | |
| ***Container Name*** | ComSignal | | |
| ***Description*** | Contains the configuration parameters of the AUTOSAR COM module's  signals. | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Configuration Parameters*** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00259 :** | | |
| ***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. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Value*** | true | | |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00158 :** | | |
| ***Name*** | ComBitSize | | |
| ***Description*** | Size in bits, for integer signal types. For ComSignalType UINT8\_N and UINT8\_DYN  the size shall be configured by ComSignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 64 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00314 :** | | |
| ***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 re- placement. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | NOTIFY | -- | |
| REPLACE | Literal for DataInvalidAction | |
| ***Post-Build Variant Multiplicity*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Confi- guration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT- POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT- POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00499 :** | | |
| ***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*** | EcucFunctionNameDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00183 :** | | |
| ***Name*** | ComFirstTimeout | | |
| ***Description*** | Defines the length of the first deadline monitoring timeout period in se- conds. This timeout is used immediately after start (or restart) of the dead- line monitoring service. The timeout period of the successive periods is configured by ECUC\_Com\_00263. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFloatParamDef | | |
| ***Range*** | 0 .. 3600 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00165 :** | | |
| ***Name*** | ComHandleId | | |
| ***Description*** | The numerical value used as the ID.  For signals it is required by the API calls Com\_UpdateShadowSignal, Com\_ReceiveShadowSignal and Com\_InvalidateShadowSignal. For sig-  nals groups it is required by the Com\_SendSignalGroup and Com\_ReceiveSignalGroup calls. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | | |
| ***Range*** | 0 .. 65535 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: ECU | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00811 :** |
| ***Name*** | ComInitialValueOnly |
| ***Description*** | This parameter defines that the respective signal's initial value shall be put into the respective PDU but there will not be any update of the value  through the RTE. Thus the Com implementation does not need to expect any API calls for this signal (group). |
| ***Multiplicity*** | 0..1 |
| ***Type*** | EcucBooleanParamDef |
| ***Default value*** | false |
| ***Post-Build Variant Multipli- city*** | false |
| ***Post-Build Variant Value*** | false |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00315 :** | | |
| ***Name*** | ComInvalidNotification | | |
| ***Description*** | Only valid on receiver side: Name of Com\_CbkInv callback function to be called. Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComDataInvalidAc-  tion is configured to NOTIFY. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFunctionNameDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00498 :** | | |
| ***Name*** | ComNotification | | |
| ***Description*** | On sender side: Name of Com\_CbkTxAck callback function to be called.  On receiver side: Name of Com\_CbkRxAck callback function to be called. If this parameter is omitted no notification shall take place. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFunctionNameDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00412 :** | | |
| ***Name*** | ComRxDataTimeoutAction | | |
| ***Description*** | This parameter defines the action performed upon expiration of the reception dead- line monitoring timer. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | NONE | no replacement shall take place | |
| REPLACE | signals shall be replaced by their ComSig- nalInitValue | |
| ***Post-Build Variant Multiplicity*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Confi- guration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT- POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT- POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00391 :** | | |
| ***Name*** | ComSignalDataInvalidValue | | |
| ***Description*** | Defines the data invalid value of the signal.  In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8\_N, UINT8\_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8\_DYN the dynamic length shall  be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucStringParamDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Class*** | ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local  dependency: In case of UINT8\_N the length of ComSignalDataInvalidVa- lue has to be the same as ComSignalLength. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00157 :** | | |
| ***Name*** | ComSignalEndianness | | |
| ***Description*** | Defines the endianness of the signal's network representation. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | BIG\_ENDIAN | -- | |
| LITTLE\_ENDIAN | -- | |
| OPAQUE | -- | |
| ***Post-Build Variant Value*** | true | | |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00170 :** | | |
| ***Name*** | ComSignalInitValue | | |
| ***Description*** | Initial value for this signal. In case of UINT8\_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of UINT8\_DYN the initial size shall be 0.  In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8\_N, UINT8\_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8\_DYN the dynamic length shall  be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucStringParamDef | | |
| ***Default value*** | 0 | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Class*** | ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: local  dependency: In case of UINT8\_N the length of ComSignalInitValue has to be the same as ComSignalLength. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00437 :** | | |
| ***Name*** | ComSignalLength | | |
| ***Description*** | Description:  For ComSignalType UINT8\_N this parameter specifies the length n in bytes. For ComSignalType UINT8\_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored.  Range: 0..8 for normal CAN/ LIN I-PDUs, 0..64 for CAN FD I-PDUs, 0..254  for normal FlexRay I-PDUs (all of ComIPduType NORMAL), 0..4294967295 for I-PDUs with ComIPduType TP. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00127 :** | |
| ***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*** | EcucEnumerationParamDef | |
| ***Range*** | BOOLEAN | -- |
| FLOAT32 | -- |
| FLOAT64 | -- |
| SINT16 | -- |
| SINT32 | -- |
| SINT64 | -- |
| SINT8 | -- |
| UINT16 | -- |
| UINT32 | -- |
| UINT64 | -- |
| UINT8 | -- |

|  |  |  |  |
| --- | --- | --- | --- |
|  | UINT8\_DYN | -- | |
| UINT8\_N | -- | |
| ***Post-Build Variant Value*** | false |  | |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT- POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Depen- dency*** | scope: local |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00263 :** | | |
| ***Name*** | ComTimeout | | |
| ***Description*** | Defines the length of the deadline monitoring timeout period in seconds.  The period for the first timeout period can be configured separately by ECUC\_Com\_00183. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFloatParamDef | | |
| ***Range*** | 0 .. 3600 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00552 :** | | |
| ***Name*** | ComTimeoutNotification | | |
| ***Description*** | On sender side: Name of Com\_CbkTxTOut callback function to be called.  On receiver side: Name of Com\_CbkRxTOut callback function to be called. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFunctionNameDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli-***  ***city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00232 :** |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Name*** | ComTransferProperty | | |
| ***Description*** | Defines if a write access to this signal can trigger the transmission of the correspon- ding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | PENDING | A write access to this signal never triggers the transmission of the cor- responding I-PDU. | |
| TRIGGERED | Depending on the trans- mission mode, a write access to this signal can trigger the transmission of  the corresponding I-PDU. | |
| TRIGGERED\_ON\_CHANGE | Depending on the trans- mission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU, but only in case the written value is different to the locally stored (last sent or  initial value) in length or value. | |
| TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION | Depending on the trans- mission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repeti- tion, but only in case the written value is different to the locally stored (last sent or initial value) in length or  value. | |
| TRIGGERED\_WITHOUT\_REPETITION | Depending on the trans- mission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU  just once without a repeti- tion. | |
| ***Post-Build Vari- ant Multiplicity*** | true | | |
| ***Post-Build Vari- ant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE- COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configu- ration Class*** | ***Pre-compile time*** | X | VARIANT-PRE- COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00257 :** | | |
| ***Name*** | ComUpdateBitPosition | | |
| ***Description*** | Bit position of update-bit inside I-PDU.  If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side.  Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli-***  ***city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00002 :** | | |
| ***Name*** | ComSystemTemplateSystemSignalRef | | |
| ***Description*** | Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal)  represents. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | Foreign reference to [ I-SIGNAL-TO-I-PDU-MAPPING ] | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: ECU | | |

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

#### ComSignalGroup

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00345 :** |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Container Name*** | ComSignalGroup | | |
| ***Description*** | Contains the configuration parameters of the AUTOSAR COM module's signal groups. | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Configuration Parameters*** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00314 :** | | |
| ***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 re- placement. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | NOTIFY | -- | |
| REPLACE | Literal for DataInvalidAction | |
| ***Post-Build Variant Multiplicity*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Confi- guration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT- POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT- POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00499 :** | | |
| ***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*** | EcucFunctionNameDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00183 :** | | |
| ***Name*** | ComFirstTimeout | | |
| ***Description*** | Defines the length of the first deadline monitoring timeout period in se- conds. This timeout is used immediately after start (or restart) of the dead-  line monitoring service. The timeout period of the successive periods is configured by ECUC\_Com\_00263. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFloatParamDef | | |
| ***Range*** | 0 .. 3600 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00165 :** | | |
| ***Name*** | ComHandleId | | |
| ***Description*** | The numerical value used as the ID.  For signals it is required by the API calls Com\_UpdateShadowSignal, Com\_ReceiveShadowSignal and Com\_InvalidateShadowSignal. For sig- nals groups it is required by the Com\_SendSignalGroup and Com\_ReceiveSignalGroup calls. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | | |
| ***Range*** | 0 .. 65535 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: ECU | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00811 :** |
| ***Name*** | ComInitialValueOnly |
| ***Description*** | This parameter defines that the respective signal's initial value shall be put into the respective PDU but there will not be any update of the value through the RTE. Thus the Com implementation does not need to expect any API calls for this signal (group). |
| ***Multiplicity*** | 0..1 |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | false | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00315 :** | | |
| ***Name*** | ComInvalidNotification | | |
| ***Description*** | Only valid on receiver side: Name of Com\_CbkInv callback function to be called. Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComDataInvalidAc-  tion is configured to NOTIFY. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFunctionNameDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00498 :** |
| ***Name*** | ComNotification |
| ***Description*** | On sender side: Name of Com\_CbkTxAck callback function to be called. On receiver side: Name of Com\_CbkRxAck callback function to be called.  If this parameter is omitted no notification shall take place. |
| ***Multiplicity*** | 0..1 |
| ***Type*** | EcucFunctionNameDef |
| ***Default value*** | -- |
| ***maxLength*** | -- |
| ***minLength*** | -- |
| ***regularExpression*** | -- |
| ***Post-Build Variant Multipli- city*** | false |
| ***Post-Build Variant Value*** | false |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00412 :** | | |
| ***Name*** | ComRxDataTimeoutAction | | |
| ***Description*** | This parameter defines the action performed upon expiration of the reception dead- line monitoring timer. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | NONE | no replacement shall take place | |
| REPLACE | signals shall be replaced by their ComSig- nalInitValue | |
| ***Post-Build Variant Multiplicity*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Confi- guration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT- POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-  POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_10003 :** | | |
| ***Name*** | ComSignalGroupArrayAccess | | |
| ***Description*** | Defines whether the uint8-array based access shall be used for this Com- SignalGroup. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** |  | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00263 :** |
| ***Name*** | ComTimeout |
| ***Description*** | Defines the length of the deadline monitoring timeout period in seconds. The period for the first timeout period can be configured separately by |

|  |  |  |  |
| --- | --- | --- | --- |
|  | ECUC\_Com\_00183. |  | |
| ***Multiplicity*** | 0..1 |  | |
| ***Type*** | EcucFloatParamDef |  | |
| ***Range*** | 0 .. 3600 |  | |
| ***Default value*** | -- |  | |
| ***Post-Build Variant Multipli- city*** | true |  | |
| ***Post-Build Variant Value*** | true |  | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00552 :** | | |
| ***Name*** | ComTimeoutNotification | | |
| ***Description*** | On sender side: Name of Com\_CbkTxTOut callback function to be called.  On receiver side: Name of Com\_CbkRxTOut callback function to be called. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFunctionNameDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli-***  ***city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00232 :** | |
| ***Name*** | ComTransferProperty | |
| ***Description*** | Defines if a write access to this signal can trigger the transmission of the correspon- ding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the  corresponding I-PDU. | |
| ***Multiplicity*** | 0..1 | |
| ***Type*** | EcucEnumerationParamDef | |
| ***Range*** | PENDING | A write access to this signal never triggers the transmission of the cor-  responding I-PDU. |
| TRIGGERED | Depending on the trans-  mission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU. |

|  |  |  |  |
| --- | --- | --- | --- |
|  | TRIGGERED\_ON\_CHANGE | Depending on the trans- mission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU, but only in case the written value is different to the locally stored (last sent or initial value) in length or  value. | |
| TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION | Depending on the trans- mission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repeti- tion, but only in case the written value is different to the locally stored (last sent or initial value) in length or  value. | |
| TRIGGERED\_WITHOUT\_REPETITION | Depending on the trans- mission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repeti-  tion. | |
| ***Post-Build Vari- ant Multiplicity*** | true | | |
| ***Post-Build Vari- ant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE- COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configu- ration Class*** | ***Pre-compile time*** | X | VARIANT-PRE- COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |
| --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00257 :** | |
| ***Name*** | ComUpdateBitPosition | |
| ***Description*** | Bit position of update-bit inside I-PDU.  If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side.  Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP. | |
| ***Multiplicity*** | 0..1 | |
| ***Type*** | EcucIntegerParamDef | |
| ***Range*** | 0 .. 4294967295 |  |
| ***Default value*** | -- | |
| ***Post-Build Variant Multipli- city*** | true | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00001 :** | | |
| ***Name*** | ComSystemTemplateSignalGroupRef | | |
| ***Description*** | Reference to the ISignalToIPduMapping that contains a reference to the ISignalGroup (SystemTemplate) which this ComSignalGroup represents. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | Foreign reference to [ I-SIGNAL-TO-I-PDU-MAPPING ] | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: ECU | | |

|  |  |  |
| --- | --- | --- |
| ***Included Containers*** | | |
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComGroupSignal | 0..\* | This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group. |

#### ComGroupSignal

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00520 :** | | |
| ***Container Name*** | ComGroupSignal | | |
| ***Description*** | This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group. | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Configuration Parameters*** | | | |

|  |  |  |
| --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00259 :** | |
| ***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. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order | |
| ***Multiplicity*** | 1 | |
| ***Type*** | EcucIntegerParamDef | |
| ***Range*** | 0 .. 4294967295 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Default value*** | -- |  | |
| ***Post-Build Variant Value*** | true |  | |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00158 :** | | |
| ***Name*** | ComBitSize | | |
| ***Description*** | Size in bits, for integer signal types. For ComSignalType UINT8\_N and UINT8\_DYN  the size shall be configured by ComSignalLength. For ComSignalTypes  FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 64 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00165 :** | | |
| ***Name*** | ComHandleId | | |
| ***Description*** | The numerical value used as the ID.  For signals it is required by the API calls Com\_UpdateShadowSignal, Com\_ReceiveShadowSignal and Com\_InvalidateShadowSignal. For sig- nals groups it is required by the Com\_SendSignalGroup and Com\_ReceiveSignalGroup calls. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | | |
| ***Range*** | 0 .. 65535 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli-***  ***city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: ECU | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00391 :** |
| ***Name*** | ComSignalDataInvalidValue |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Description*** | Defines the data invalid value of the signal.  In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8\_N, UINT8\_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8\_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be  interpreted as 0-sized dynamic signal. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucStringParamDef | | |
| ***Default value*** | -- | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: local  dependency: In case of UINT8\_N the length of ComSignalDataInvalidVa- lue has to be the same as ComSignalLength. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00157 :** | | |
| ***Name*** | ComSignalEndianness | | |
| ***Description*** | Defines the endianness of the signal's network representation. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | BIG\_ENDIAN | -- | |
| LITTLE\_ENDIAN | -- | |
| OPAQUE | -- | |
| ***Post-Build Variant Value*** | true | | |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00170 :** |
| ***Name*** | ComSignalInitValue |
| ***Description*** | Initial value for this signal. In case of UINT8\_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of |

|  |  |  |  |
| --- | --- | --- | --- |
|  | UINT8\_DYN the initial size shall be 0.  In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8\_N, UINT8\_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8\_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be  interpreted as 0-sized dynamic signal. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucStringParamDef | | |
| ***Default value*** | 0 | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: local  dependency: In case of UINT8\_N the length of ComSignalInitValue has to be the same as ComSignalLength. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00437 :** | | |
| ***Name*** | ComSignalLength | | |
| ***Description*** | Description:  For ComSignalType UINT8\_N this parameter specifies the length n in bytes. For ComSignalType UINT8\_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored.  Range: 0..8 for normal CAN/ LIN I-PDUs, 0..64 for CAN FD I-PDUs, 0..254  for normal FlexRay I-PDUs (all of ComIPduType NORMAL), 0..4294967295 for I-PDUs with ComIPduType TP. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST-  BUILD |
| ***Post-build time*** | -- |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00127 :** | | |
| ***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*** | EcucEnumerationParamDef | | |
| ***Range*** | BOOLEAN | -- | |
| FLOAT32 | -- | |
| FLOAT64 | -- | |
| SINT16 | -- | |
| SINT32 | -- | |
| SINT64 | -- | |
| SINT8 | -- | |
| UINT16 | -- | |
| UINT32 | -- | |
| UINT64 | -- | |
| UINT8 | -- | |
| UINT8\_DYN | -- | |
| UINT8\_N | -- | |
| ***Post-Build Variant***  ***Value*** | false | | |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT- POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00560 :** | | |
| ***Name*** | ComTransferProperty | | |
| ***Description*** | Optionally defines whether this group signal shall contribute to the TRIGGE- RED\_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. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | PENDING | A change of the value of this group signal shall not be considered in the evaluation of the signal groups Com-  TransferProperty. | |
| TRIGGERED\_ON\_CHANGE | A change of the value of this group signal shall be considered in the eva- luation of the signal groups ComTrans- ferProperty. | |
| ***Post-Build Variant Multiplicity*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Confi-*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |

|  |  |  |  |
| --- | --- | --- | --- |
| ***guration Class*** | ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configura- tion Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00002 :** | | |
| ***Name*** | ComSystemTemplateSystemSignalRef | | |
| ***Description*** | Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | Foreign reference to [ I-SIGNAL-TO-I-PDU-MAPPING ] | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: ECU | | |

|  |  |  |
| --- | --- | --- |
| ***Included Containers*** | | |
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComFilter | 0..1 | This container contains the configuration parameters of the AUTOSAR COM module's filters. Note: On sender side the container is used to specify the transmission mode conditions. |

#### ComTxMode

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00351 :** |
| ***Container Name*** | ComTxMode |
| ***Description*** | This container contains the configuration parameters of the AUTOSAR COM module's transmission modes. |
| ***Configuration Parameters*** | |

|  |  |  |
| --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00137 :** | |
| ***Name*** | ComTxModeMode | |
| ***Description*** | The available transmission modes described in [18] shall be extended by the addi- tional mode None.  The transmission mode None shall not have any further sub-attributes in the ComTxMode object. | |
| ***Multiplicity*** | 1 | |
| ***Type*** | EcucEnumerationParamDef | |
| ***Range*** | DIRECT | -- |
| MIXED | -- |
| NONE | Literal for TxMode |
| PERIODIC | -- |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Post-Build Variant Value*** | true |  | |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00281 :** | | |
| ***Name*** | ComTxModeNumberOfRepetitions | | |
| ***Description*** | Defines the number of repetitions for the transmission mode DIRECT and the event driven part of transmission mode MIXED. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 255 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00282 :** | | |
| ***Name*** | ComTxModeRepetitionPeriod | | |
| ***Description*** | Defines the repetition period in seconds of the multiple transmissions in case ComTxModeNumberOfRepetitions is configured greater than or equal to 1 and ComTxModeMode is configured to DIRECT or MIXED. In case of the mixed transmission mode only the event driven part is affected. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFloatParamDef | | |
| ***Range*** | 0 .. 3600 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00180 :** |
| ***Name*** | ComTxModeTimeOffset |
| ***Description*** | Defines the period in seconds between the start of the I-PDU by Com\_IpduGroupControl and the first transmission request in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected.  In case ComTxModeTimeOffset is omitted or configured to 0, the first |

|  |  |  |  |
| --- | --- | --- | --- |
|  | periodic transmission shall be transmitted within the next invocation of Com\_MainFunctionTx. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFloatParamDef | | |
| ***Range*** | 0 .. 3600 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00178 :** | | |
| ***Name*** | ComTxModeTimePeriod | | |
| ***Description*** | Defines the repetition period in seconds of the periodic transmission re- quests in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucFloatParamDef | | |
| ***Range*** | 0 .. 3600 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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***

#### ComTxModeTrue

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00455 :** |
| ***Container Name*** | ComTxModeTrue |
| ***Description*** | This container contains the configuration parameters of the AUTOSAR COM module's 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 the AUTOSAR COM module's transmission modes. |

#### ComTxModeFalse

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00454 :** |
| ***Container Name*** | ComTxModeFalse |
| ***Description*** | This container contains the configuration parameters of the AUTOSAR COM module's 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 the AUTOSAR COM module's transmission modes. |

#### ComGwMapping

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00544 :** | | |
| ***Container Name*** | ComGwMapping | | |
| ***Description*** | Each instance of this container defines one mapping of the integrated Signal Gateway. | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***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 contai- ner. |

#### ComGwSource

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00545 :** |
| ***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 destina-  tion respectively with a reference to a ComSignal, a Com- GroupSignal or a ComSignalGroup. |
| ComGwSourceDescription | 0..1 | Description of a gateway source. This container allows defi- ning 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. |

#### ComGwSourceDescription

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00548 :** |
| ***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*** | **ECUC\_Com\_00259 :** | | |
| ***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. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0  of the first byte like in little endian byte order | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Value*** | true | | |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00158 :** | | |
| ***Name*** | ComBitSize | | |
| ***Description*** | Size in bits, for integer signal types. For ComSignalType UINT8\_N and UINT8\_DYN  the size shall be configured by ComSignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and  therefore may be omitted. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 64 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00157 :** | | |
| ***Name*** | ComSignalEndianness | | |
| ***Description*** | Defines the endianness of the signal's network representation. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | BIG\_ENDIAN | -- | |
| LITTLE\_ENDIAN | -- | |
| OPAQUE | -- | |
| ***Post-Build Variant Value*** | true | | |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00437 :** | | |
| ***Name*** | ComSignalLength | | |
| ***Description*** | Description:  For ComSignalType UINT8\_N this parameter specifies the length n in bytes. For ComSignalType UINT8\_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored.  Range: 0..8 for normal CAN/ LIN I-PDUs, 0..64 for CAN FD I-PDUs, 0..254  for normal FlexRay I-PDUs (all of ComIPduType NORMAL), 0..4294967295 for I-PDUs with ComIPduType TP. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | false | | |
| ***Post-Build Variant Value*** | false | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT-POST- BUILD |
| ***Post-build time*** | -- |  |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00127 :** | |
| ***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*** | EcucEnumerationParamDef | |
| ***Range*** | BOOLEAN | -- |

|  |  |  |  |
| --- | --- | --- | --- |
|  | FLOAT32 | -- | |
| FLOAT64 | -- | |
| SINT16 | -- | |
| SINT32 | -- | |
| SINT64 | -- | |
| SINT8 | -- | |
| UINT16 | -- | |
| UINT32 | -- | |
| UINT64 | -- | |
| UINT8 | -- | |
| UINT8\_DYN | -- | |
| UINT8\_N | -- | |
| ***Post-Build Variant Value*** | false |  | |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME, VARIANT- POST-BUILD |
| ***Post-build time*** | -- |  |
| ***Scope / Depen- dency*** | scope: local |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00257 :** | | |
| ***Name*** | ComUpdateBitPosition | | |
| ***Description*** | Bit position of update-bit inside I-PDU.  If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side.  Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00550 :** | | |
| ***Name*** | ComGwIPduRef | | |
| ***Description*** | Reference to an I-PDU of a Signal Gateway source or destination descrip- tion. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | Reference to [ ComIPdu ] | | |
| ***Post-Build Variant Value*** | true | | |
| ***Value Configuration Class*** | ***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***

#### ComGwDestination

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00546 :** | | |
| ***Choice container Name*** | ComGwDestination | | |
| ***Description*** | Each instance of this choice container allows to define one routing destina- tion either by reference to an already configured COM signal / signal group  or by a destination description container. | | |
| ***Post-Build Variant Multipli-***  ***city*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |

|  |  |  |
| --- | --- | --- |
| ***Container Choices*** | | |
| ***Container Name*** | ***Multiplicity*** | ***Scope / Dependency*** |
| ComGwDestinationDescripti- on | 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 destina- tion respectively with a reference to a ComSignal, a Com- GroupSignal or a ComSignalGroup. |

#### ComGwDestinationDescription

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00549 :** |
| ***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*** | **ECUC\_Com\_00259 :** | | |
| ***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. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0  of the first byte like in little endian byte order | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Value*** | true | | |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00157 :** | | |
| ***Name*** | ComSignalEndianness | | |
| ***Description*** | Defines the endianness of the signal's network representation. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | BIG\_ENDIAN | -- | |
| LITTLE\_ENDIAN | -- | |
| OPAQUE | -- | |
| ***Post-Build Variant Value*** | true | | |
| ***Value Configurati- on Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00170 :** | | |
| ***Name*** | ComSignalInitValue | | |
| ***Description*** | Initial value for this signal. In case of UINT8\_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of UINT8\_DYN the initial size shall be 0.  In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8\_N, UINT8\_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8\_DYN the dynamic length shall  be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucStringParamDef | | |
| ***Default value*** | 0 | | |
| ***maxLength*** | -- | | |
| ***minLength*** | -- | | |
| ***regularExpression*** | -- | | |
| ***Post-Build Variant Multipli- city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** | scope: local  dependency: In case of UINT8\_N the length of ComSignalInitValue has to be the same as ComSignalLength. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00232 :** | | |
| ***Name*** | ComTransferProperty | | |
| ***Description*** | Defines if a write access to this signal can trigger the transmission of the correspon- ding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucEnumerationParamDef | | |
| ***Range*** | PENDING | A write access to this signal never triggers the transmission of the cor-  responding I-PDU. | |
| TRIGGERED | Depending on the trans- mission mode, a write access to this signal can  trigger the transmission of the corresponding I-PDU. | |
| TRIGGERED\_ON\_CHANGE | Depending on the trans- mission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU, but only in case the written value is different to the locally stored (last sent or initial value) in length or  value. | |
| TRIGGERED\_ON\_CHANGE\_WITHOUT\_REPETITION | Depending on the trans- mission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repeti- tion, but only in case the written value is different to the locally stored (last sent  or initial value) in length or value. | |
| TRIGGERED\_WITHOUT\_REPETITION | Depending on the trans- mission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repeti-  tion. | |
| ***Post-Build Vari-***  ***ant Multiplicity*** | true | | |
| ***Post-Build Vari- ant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE- COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configu- ration Class*** | ***Pre-compile time*** | X | VARIANT-PRE- COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Depen- dency*** | scope: local | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00257 :** | | |
| ***Name*** | ComUpdateBitPosition | | |
| ***Description*** | Bit position of update-bit inside I-PDU.  If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side.  Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP. | | |
| ***Multiplicity*** | 0..1 | | |
| ***Type*** | EcucIntegerParamDef | | |
| ***Range*** | 0 .. 4294967295 |  | |
| ***Default value*** | -- | | |
| ***Post-Build Variant Multipli-***  ***city*** | true | | |
| ***Post-Build Variant Value*** | true | | |
| ***Multiplicity Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Value Configuration Class*** | ***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*** | **ECUC\_Com\_00550 :** | | |
| ***Name*** | ComGwIPduRef | | |
| ***Description*** | Reference to an I-PDU of a Signal Gateway source or destination descrip- tion. | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | Reference to [ ComIPdu ] | | |
| ***Post-Build Variant Value*** | true | | |
| ***Value Configuration Class*** | ***Pre-compile time*** | X | VARIANT-PRE-COMPILE |
| ***Link time*** | X | VARIANT-LINK-TIME |
| ***Post-build time*** | X | VARIANT-POST-BUILD |
| ***Scope / Dependency*** |  | | |

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

#### ComGwSignal

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00551 :** |
| ***Container Name*** | ComGwSignal |
| ***Description*** | This container allows specifying a gateway source or destination respec- tively with a reference to a ComSignal, a ComGroupSignal or a ComSig- nalGroup. |
| ***Configuration Parameters*** | |

|  |  |
| --- | --- |
| ***SWS Item*** | **ECUC\_Com\_00547 :** |

|  |  |  |  |
| --- | --- | --- | --- |
| ***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 ] | | |
| ***Post-Build Variant Value*** | true | | |
| ***Value Configuration Class*** | ***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***

### Configuration Rules

#### General Rules

**[SWS\_Com\_00401]** ⌈It is illegal for any two of the following parameters to have the same value:

* shortName of a ComSignal
* shortName of a ComSignalGroup
* shortName of a ComGroupSignal
* shortName of a ComIPdu
* shortName of a ComIPduGroup

⌋ (SRS\_Com\_02067)

**[SWS\_Com\_00732]** ⌈It is illegal for any of the following five parameters:

* ComNotification
* ComErrorNotification
* ComTimeoutNotification
* ComInvalidNotification
* ComIPduCallout

to have the same value as any of the following five parameters

* Com\_<shortName of a ComSignal>
* Com\_<shortName of a ComSignalGroup>
* Com\_<shortName of a ComGroupSignal>
* Com\_<shortName of a ComIPdu>
* Com\_<shortName of a ComIPduGroup>

⌋ (SRS\_Com\_02067)

**[SWS\_Com\_00402]** ⌈It is illegal for any of the following parameters not to be formu- lated according to C’s identifier rules:

* shortName of a ComSignal
* shortName of a ComSignalGroup
* shortName of a ComGroupSignal
* shortName of a ComIPdu
* shortName of a ComIPduGroup
* ComNotification
* ComErrorNotification
* ComTimeoutNotification
* ComInvalidNotification
* ComIPduCallout

⌋ (SRS\_Com\_02067)

#### Signal Configuration

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

**[SWS\_Com\_00102]** ⌈ComSignal/ ComGroupSignal are not allowed to overlap each other.⌋ (SRS\_Com\_02067)

It is explicitly allowed that a ComSignal/ ComGroupSignal may have the size 0, see [ECUC\_Com\_00158](#_bookmark263).

**[SWS\_Com\_00105]** ⌈ComSignal/ ComGroupSignal of ComSignalType UINT8\_N and UINT8\_DYN shall be byte aligned (ComBitPosition is a multiple of 8).⌋ (SRS\_Com\_02067)

**[SWS\_Com\_00443]** ⌈A ComSignal/ ComGroupSignal of type uint8[n] shall always be mapped to an n-bytes sized ComSignal/ ComGroupSignal.⌋ (SRS\_Com\_02067)

**[SWS\_Com\_00553]** ⌈A ComSignal/ ComGroupSignal of type uint8[n] shall be con- figured to have OPAQUE endianness.⌋ (SRS\_Com\_02067)

**[SWS\_Com\_00754]** ⌈A dynamic length I-PDU shall contain at most one dynamic length signal.⌋ (SRS\_Com\_02091, SRS\_Com\_02092)

**[SWS\_Com\_00755]** ⌈The dynamic length signal shall be placed last in a dynamic length I-PDU.⌋ (SRS\_Com\_02091, SRS\_Com\_02093)

**[SWS\_Com\_00756]** ⌈The ComSignalLength parameter shall be configured to the maximum size/ length for dynamic length signals.⌋ (SRS\_Com\_02091, SRS\_Com\_02097)

**[SWS\_Com\_00310]** ⌈For I-PDUs with transmission mode *DIRECT* or *MIXED* with ComTxModeNumberOfRepetitions greater or equal 1 no update-bit (ComUpdateBit- Position) shall be configured.⌋(SRS\_Com\_02067, SRS\_Com\_02030)

**[SWS\_Com\_00785]** ⌈The ComBitSize of a (group) signal shall not extend past the size of its configured ComSignalType.⌋ (SRS\_Com\_02067)

For example, the ComBitSize of a signal with ComSignalType UINT8 shall not ex- ceed 8 bits.

**[SWS\_Com\_00790]** ⌈The configured string of ComSignalInitValues/ ComSignalData- InvalidValue shall be interpreted according to the definitions of [[22]](#_bookmark14) for the boolean and all numerical types.⌋ (SRS\_Com\_02067)

**[SWS\_Com\_00859]** ⌈In case a ComSignalDataInvalidValue is configured for a ComSignal that is included in a ComIPdu with ComIPduDirection *RECEIVE*, the ComDataInvalidAction shall also be configured for this ComSignal.⌋ (SRS\_Com\_02079, SRS\_Com\_02087)

#### Signal Group Configuration

**[SWS\_Com\_00365]** ⌈It shall not be allowed to configure signal groups for routing with data type differences between receive and transmit signal group. A sub-setting of signal groups (as defined in [SWS\_Com\_00735](#_bookmark132)) shall be supported.⌋ (SRS\_Com\_02067)

How the signals of signal groups are placed within an I-PDU is not restricted. It is allowed to define interlaced signal groups, or to place other signals within the wholes of a signal group.

**[SWS\_Com\_00860]** ⌈In case a ComSignalDataInvalidValue is configured for at least one ComGroupSignal of a ComSignalGroup that is included in a ComIPdu with ComIPduDirection *RECEIVE*, the ComDataInvalidAction shall also be configured for this ComSignalGroup.⌋ (SRS\_Com\_02079, SRS\_Com\_02087)

#### Transmission Mode Configuration

**[SWS\_Com\_00319]** ⌈It shall not be allowed to configure a ComFilter (respectively TMS-conditions) for signals with ComSignalType *FLOAT32* or *FLOAT64*.⌋ (SRS\_Com\_02067)

It is not be allowed to configure a ComFilter (respectively TMS-conditions) that uses floats.

Floats are not allowed to be used in filter conditions. See [[17]](#_bookmark17) and [SWS\_Com\_00132](#_bookmark43). Therefore, floats are not allowed for conditions of TMS.

**[SWS\_Com\_00465]** ⌈Every ComTxModeTrue or ComTxModeFalse that is a potential result of the configured/ calculated TMS must be configured. Within the ComTxIPdu, at least one of the containers ComTxModeTrue or ComTxModeFalse has to be in- cluded.⌋ (SRS\_Com\_02067)

#### Signal Gateway Configuration

**[SWS\_Com\_00384]** ⌈The ComBitSize of a received and to be routed ComSignal shall not differ.⌋ (SRS\_Com\_02067)

**[SWS\_Com\_00598]** ⌈A signal or signal group contained in an I-PDU with ComIPdu- Type configured as TP shall not be configured as source or destination within the ComGwMapping configuration container.⌋ (SRS\_PduR\_06055)

Additionally to reception deadline monitoring of a ComSignal, to be routed by the Signal Gateway, it is possible to configure update-bits via ComUpdateBitPosition, for the transmit signal. In this case, the receiving node can detect if the sender has really updated the signal or it is just repeated by the Signal Gateway. If this is necessary depends on the use-case.

#### Filter Configuration

**[SWS\_Com\_00535]** ⌈For the ComFilterAlgorithm *ONE\_EVERY\_N*, the ComFilterOff- set shall be configured to a value lesser than ComFilterPeriod.⌋ (SRS\_Com\_02067)

#### Post Build Configuration

**[SWS\_Com\_00373]** ⌈The post-build time configuration part (post-compile and post- link time) can only be updated when it is not in use.⌋ (SRS\_Com\_02067, SRS\_PduR\_06002)

**[SWS\_Com\_00487]** ⌈The complete post-build time configurable configuration shall be identifiable by a unique identifier.⌋ (SRS\_PduR\_06097)

#### Dynamic Length I-PDUs

**[SWS\_Com\_00817]** ⌈I-PDUs including metadata shall not be configured as dynamic length I-PDUs.⌋ (SRS\_Com\_02067)

The above restriction is necessary, since the length of the dynamic length signal could not be derived correctly, when the I-PDU contains additional metadata.

Since the FlexRay Interface does only support even values for the I-PDU length, it is not recommend to configure dynamic length I-PDU to be send via FrIf. Instead, the FlexRay TP module should be used to transport dynamic length I-PDUs via FlexRay.

#### Replicated I-PDUs

**[SWS\_Com\_00834]** ⌈If an I-PDU is configured for replication (includes a ComIPduReplication), this I-PDU shall also have configured an I-PDU counter (ComIPduCounter) with a counter threshold (ComIPduCounterThreshold) set to 0.⌋ (SRS\_Com\_02102)

For further information on this constraint see [7.10](#_bookmark125).

# Not Applicable Requirements

**[SWS\_Com\_00999]** ⌈These requirements are not applicable to this specification.⌋ (SRS\_BSW\_00171, SRS\_BSW\_00170, SRS\_BSW\_00383, SRS\_BSW\_00375, SRS\_BSW\_00416, SRS\_BSW\_00437, SRS\_BSW\_00168, SRS\_BSW\_00423, SRS\_BSW\_00424, SRS\_BSW\_00426, SRS\_BSW\_00427, SRS\_BSW\_00428, SRS\_BSW\_00429, SRS\_BSW\_00433, SRS\_BSW\_00417, SRS\_BSW\_00409, SRS\_BSW\_00386, SRS\_BSW\_00161, SRS\_BSW\_00162, SRS\_BSW\_00005, SRS\_BSW\_00164, SRS\_BSW\_00325, SRS\_BSW\_00413, SRS\_BSW\_00347, SRS\_BSW\_00307, SRS\_BSW\_00410, SRS\_BSW\_00314, SRS\_BSW\_00353, SRS\_BSW\_00361, SRS\_BSW\_00302, SRS\_BSW\_00328, SRS\_BSW\_00006, SRS\_BSW\_00378, SRS\_BSW\_00306, SRS\_BSW\_00308, SRS\_BSW\_00309, SRS\_BSW\_00009, SRS\_BSW\_00010, SRS\_BSW\_00333, SRS\_BSW\_00321, SRS\_BSW\_00341, SRS\_BSW\_00334)

# Appendix A

This appendix contains use cases with different transmission modes and the neces- sary configuration for these. [Table 11](#_bookmark321) shows the notation of the use case diagrams.

|  |  |
| --- | --- |
| ***Notation*** | ***Description*** |
| tc, tc1, tc2 | cycle times |
| td | cycle time of N-Times sent signals |
| tr | minimum SW reaction time of COM-Layer due to internal main cycle time |
| V | Value:  x stands for an arbitrary value/ value range,  a…w for specific values / value ranges, defined by the user, with a <> b, *range a* is disjoint from *range b*. |
|  | request from RTE to the COM-Layer |
|  | request from COM-Layer to PDU Router |
|  | potential but skipped request from COM-Layer to PDU Router (e.g. because of a new send request by the RTE or delayed due to minimum delay time) |
| dt | minimum delay time: minimum distance between two requests to PDU Router |
| without TMS switch | without switching of the TMS (see [7.3.3.2](#_bookmark60)) from *true* to *false* or vice versa |
| with TMS switch | with switching of the TMS (see [7.3.3.2](#_bookmark60)) from *true* to *false* or vice versa (from  TM 1 to TM 2); one TM is named before the “+” and one behind in the descrip- tion |

**Table 11: Legend for use case diagrams.**

**Use case 1** shows an I-PDU that is sent out cyclically with a cycle time tc. This I-PDU consists of signals that all have the ComTransferProperty *PENDING*. It is configured that the transmission takes place when the TMS evaluates to true.

v=x v=x v=x

tc

tc

tc

tc

tc

t

**Figure 12: Use case 1, TM periodic (without TMS switch, see Chapter** [**7.3.3.2**](#_bookmark60)**)**

|  |  |
| --- | --- |
| ***ComSignal*** |  |
| ***ComTransferProperty*** | PENDING or TRIGGERED  (TRIGGERED has no influence) |
| ***ComFilter*** |  |

|  |  |
| --- | --- |
| ***ComFilterAlgorithm*** | ALWAYS |
| ***ComIPdu*** |  |
| ***ComIPduDirection*** | SEND |
| ***ComTxModeTrue*** |  |
| ***ComTxModeTimePeriod*** | tc |
| ***ComTxModeMode*** | PERIODIC |
| ***ComTxIPdu*** |  |
| ***ComMinimumDelayTime*** | 0 or omitted |

**Table 12: Relevant I-PDU transmission configuration for use case 1**

Because of the configuration of the parameter ComFilterAlgorithm *ALWAYS* of the ComFilter, there is no need to configure a transmission mode for the case that the TMS evaluates to false.

It does not make any difference in the behavior whether the ComFilterAlgorithm pa- rameter of the ComFilter is set for all the signals within the I-PDU to *ALWAYS* or if the ComFilter is not defined (does not contribute to the evaluation of the TMS), see SWS\_Com\_00255.

**Use case 2** shows an I-PDU which is sent out three times whenever a value is given by the upper (Com\_SendSignal or Com\_SendSignalGroup). The time between two transmissions is td. This I-PDU consists of signals, which all have the ComTransfer- Property *TRIGGERED*. It is configured that the transmission takes place when the TMS evaluates to true.

A screenshot of a computer

Description automatically generated

**Figure 13: Use case 2, TM DIRECT (N-Times) here n = 2 (without TMS switch)**

|  |  |
| --- | --- |
| ***ComGeneral*** |  |
| ***ComEnableMDTForCyclicTransmission*** | true |
| ***ComSignal*** |  |
| ***ComTransferProperty*** | TRIGGERED |
| ***ComFilter*** |  |
| ***ComFilterAlgorithm*** | ALWAYS |
| ***ComIPdu*** |  |
| ***ComIPduDirection*** | SEND |
| ***ComTxModeTrue*** |  |
| ***ComTxModeMode*** | DIRECT |
| ***ComTxModeNumberOfRepetitions*** | 2 |
| ***ComTxModeRepetitionPeriod*** | td |
| ***ComTxIPdu*** |  |
| ***ComMinimumDelayTime*** | dt |

**Table 13: Relevant I-PDU transmission configuration for use case 2**

If there is a new send request by the RTE before the ongoing repetition cycle is com- pleted, the new repetition cycle is started and the rest of the last one is discarded.

**Use case 3** shows an I-PDU which is send out cyclically with a cycle time tc1 if value v = a (TMS evaluates to true) and with a cycle time tc2 if value v = b (TMS evaluates to false). The I-PDU consists of signals, which all have the ComTransferProperty *PENDING*.

A screen shot of a computer screen

Description automatically generated

**Figure 14: Use case 3, TM periodic + periodic (with TMS switch)**

|  |  |
| --- | --- |
| ***ComGeneral*** |  |
| ***ComEnableMDTForCyclicTransmission*** | true |
| ***ComSignal*** |  |
| ***ComTransferProperty*** | PENDING or TRIGGERED  (TRIGGERED has no influence) |
| ***ComFilter*** |  |
| ***ComFilterAlgorithm*** | any filter mapping value a to true and value b to false |
| ***ComIPdu*** |  |
| ***ComIPduDirection*** | SEND |
| ***ComTxModeTrue*** |  |
| ***ComTxModeMode*** | PERIODIC |
| ***ComTxModeTimePeriod*** | tc1 |
| ***ComTxModeFalse*** |  |
| ***ComTxModeMode*** | PERIODIC |
| ***ComTxModeTimePeriod*** | tc2 |
| ***ComTxIPdu*** |  |
| ***ComMinimumDelayTime*** | dt |

**Table 14: Relevant I-PDU transmission configuration for use case 3**

The TMS switch caused by the new value v = b, starts a new cycle within the next main function and the new value is sent out. Nevertheless, the minimum delay time dt has to be taken into account, since ComEnableMDTForCyclicTransmission is set to true.

For the parameter ComFilterAlgorithm, the following OSEK COM defined filters can be used for this use-case:

* *MASKED\_NEW\_EQUALS\_X*
* *MASKED\_NEW\_DIFFERS\_X*
* *MASKED\_NEW\_DIFFERS\_MASKED\_OLD*
* *NEW\_IS\_WITHIN*
* *NEW\_IS\_OUTSIDE*

**Use case 4** shows an I-PDU which is send out cyclically with a cycle time tc if value v

= a (TMS evaluates to true) and if value v = b (TMS evaluates to false) it is sent out three times whenever the value is given by the RTE. The time between two transmis- sions is td. The I-PDU consists of signals that all have the ComTransferProperty *TRIGGERED*.

A group of arrows pointing to different directions

Description automatically generated with medium confidence

**Figure 15: Use case 4 scenario a, TM periodic + *DIRECT*/N-Times, here n = 2 (with TMS switch)**

A diagram of different types of numbers

Description automatically generated with medium confidence

**Figure 16: Use case 4 scenario b, TM periodic + *DIRECT*/N-Times, here n = 2 (with TMS switch)**

|  |  |
| --- | --- |
| ***ComGeneral*** |  |
| ***ComEnableMDTForCyclicTransmission*** | True |
| ***ComSignal*** |  |
| ***ComTransferProperty*** | TRIGGERED |
| ***ComFilter*** |  |
| ***ComFilterAlgorithm*** | any filter mapping value a to true and value b to false |
| ***ComIPdu*** |  |
| ***ComIPduDirection*** | SEND |
| ***ComTxModeTrue*** |  |
| ***ComTxModeMode*** | PERIODIC |
| ***ComTxModeTimePeriod*** | tc |
| ***ComTxModeFalse*** |  |
| ***ComTxModeMode*** | DIRECT |
| ***ComTxModeNumberOfRepetitions*** | 2 |
| ***ComTxModeRepetitionPeriod*** | td |
| ***ComTxIPdu*** |  |
| ***ComMinimumDelayTime*** | dt |

**Table 15: Relevant I-PDU transmission configuration for use case 4**

After the switch from ComTxModeMode *DIRECT* to *PERIODIC* the cycle is started within the next main function and the new value a is sent out with respect to ComMin- imumDelayTime dt.

**Use case 5** shows an I-PDU which is send out cyclically with a cycle time tc and if the value (the same or a new one) is given by the RTE it is also sent out directly three times. The time between two of these three transmissions is always td. The I-PDU consists of signals that all have the ComTransferProperty *TRIGGERED*.

A diagram of different types of arrows

Description automatically generated with medium confidence

**Figure 17: Use case 5 scenario a, TM *MIXED*, here n = 2 (without TMS switch)**

A group of arrows pointing at different directions

Description automatically generated with medium confidence

**Figure 18: Use case 5 scenario b, TM *MIXED*, here n = 2 (without TMS switch)**

|  |  |
| --- | --- |
| ***ComGeneral*** |  |
| ***ComEnableMDTForCyclicTransmission*** | true |
| ***ComSignal*** |  |
| ***ComTransferProperty*** | TRIGGERED |
| ***ComFilter*** |  |
| ***ComFilterAlgorithm*** | ALWAYS |
| ***ComIPdu*** |  |
| ***ComIPduDirection*** | SEND |
| ***ComTxModeTrue*** |  |
| ***ComTxModeMode*** | MIXED |
| ***ComTxModeTimePeriod*** | tc |

|  |  |
| --- | --- |
| ***ComTxModeNumberOfRepetitions*** | 2 |
| ***ComTxModeRepetitionPeriod*** | td |
| ***ComTxIPdu*** |  |
| ***ComMinimumDelayTime*** | dt |

**Table 16: Relevant I-PDU transmission configuration for use case 5**

If the next transmission caused by the periodic part of the ComTxModeMode *MIXED* should take place within the timeout dt (ComMinimumDelayTime) after a transmis- sion of the *DIRECT* (N-Times) part, this sent out is delayed until the minimum delay time is elapsed. However, after that the next period of the periodic part is shortened so that there is only an intermediate phase shift of the periodic part but no continuous one.

**Use case 6** shows an I-PDU which is send out cyclically with a cycle time tc2 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 tc1 and whenever the value v = a is given by the RTE it is also sent out directly three times. The time between two of these three transmissions is always td. The I-PDU consists of signals that all have the ComTransferProperty *TRIGGERED*.

A diagram of different types of arrows

Description automatically generated with medium confidence

**Figure 19: Use Case 6, TM *MIXED*, here n= 2 + periodic (with TMS switch)**

|  |  |
| --- | --- |
| ***ComGeneral*** |  |
| ***ComEnableMDTForCyclicTransmission*** | true |
| ***ComSignal*** |  |
| ***ComTransferProperty*** | TRIGGERED |
| ***ComFilter*** |  |
| ***ComFilterAlgorithm*** | any filter mapping value a to true and value b to false |
| ***ComIPdu*** |  |
| ***ComIPduDirection*** | SEND |
| ***ComTxModeTrue*** |  |
| ***ComTxModeMode*** | MIXED |
| ***ComTxModeTimePeriod*** | tc1 |
| ***ComTxModeNumberOfRepetitions*** | 2 |
| ***ComTxModeRepetitionPeriod*** | td |
| ***ComTxModeFalse*** |  |
| ***ComTxModeMode*** | PERIODIC |
| ***ComTxModeTimePeriod*** | tc2 |
| ***ComTxIPdu*** |  |
| ***ComMinimumDelayTime*** | dt |

**Table 17: Relevant I-PDU transmission configuration for use case 6**

A usage of this in practice is for example the signal of the button that controls the window-lift motor. If the button is not pressed, there is a long cycle time tc1 with this information. If it is pressed this information is distributed with a short cycle time tc2. If the button is released again, starting with the next main function this information is distributed three times with td and after that, again the long cycle time is used.

**Use case 7** is similar to use case 5 but with ComEnableMDTForCyclicTransmission disabled and ComTxModeNumberOfRepetitions set to one.

A black background with arrows and symbols

Description automatically generated with medium confidence

**Figure 20: Use case 7 scenario a, TM *MIXED*, here n = 1 (without TMS switch)**

A screen shot of a computer screen

Description automatically generated

**Figure 21: Use case 7 scenario b, TM *MIXED*, here n = 1 (without TMS switch)**

A screenshot of a computer screen

Description automatically generated

**Figure 22: Use case 7 scenario c, TM *MIXED*, here n = 1 (without TMS switch)**

|  |  |
| --- | --- |
| ***ComGeneral*** |  |
| ***ComEnableMDTForCyclicTransmission*** | False |
| ***ComSignal*** |  |
| ***ComTransferProperty*** | TRIGGERED |
| ***ComFilter*** |  |
| ***ComFilterAlgorithm*** | ALWAYS |
| ***ComIPdu*** |  |
| ***ComIPduDirection*** | SEND |
| ***ComTxModeTrue*** |  |
| ***ComTxModeMode*** | MIXED |
| ***ComTxModeTimePeriod*** | tc |
| ***ComTxModeNumberOfRepetitions*** | 1 |
| ***ComTxModeRepetitionPeriod*** | td |
| ***ComTxIPdu*** |  |
| ***ComMinimumDelayTime*** | dt |

**Table 18: Relevant I-PDU transmission configuration for use case 7**

In contrary to use case 5, here the minimum delay timer is neither started for the periodic transmissions of the cyclic part nor for the repeated transmissions of the direct part of the Mixed mode. This can result in two consecutive transmissions within the timespan dt as shown in the scenarios b and c.