Document Title	Specification of Module XCP
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	412

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R24-11

Document Change History

Date	Release	Changed by	Description
2024-11-27	R24-11	AUTOSAR Release Management	• adapted interaction with lower layer to LSduR
2023-11-23	R23-11	AUTOSAR Release Management	• Removed all HandleId configuration parameters
2022-11-24	R22-11	AUTOSAR Release Management	• Editorial changes
2021-11-25	R21-11	AUTOSAR Release Management	• No content changes
2020-11-30	R20-11	AUTOSAR Release Management	• Improve the structure of the 'error sections'
2019-11-28	R19-11	AUTOSAR Release Management	• No content changes • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Update XCP on CAN version to support CAN FD
2017-12-08	4.3.1	AUTOSAR Release Management	• Editorial changes • Update development errors • Adapt XCP with CAN return types • Add new configuration container XcpChannel

△

2016-11-30	4.3.0	AUTOSAR Release Management	• Editorial changes • Harmonize descriptions of identical API functions • Removal of unused artifacts and obsolete elements
2015-07-31	4.2.2	AUTOSAR Release Management	• Debugging support marked as obsolete • Editorial changes • Modifications in some parameters multiplicity of XcpDaqlist container
2014-10-31	4.2.1	AUTOSAR Release Management	• Editorial changes • Minor corrections • Changed the multiplicity of XcpEventChannelTriggeredDaqListRef • Remove limitation “Flash Programming for ECU development purposes”
2014-03-31	4.1.3	AUTOSAR Release Management	• Editorial correction for faulty references links • Minor editorial correction for SWS_Xcp_00841, SWS_Xcp_00844 • Changed Xcp_RxIndication argument from PduInfoType * to const PduInfoType *
2013-10-31	4.1.2	AUTOSAR Release Management	• Minor corrections • Editorial changes • Removed chapter(s) on change documentation
2013-03-15	4.1.1	AUTOSAR Administration	• Reclassify XCP_E_INIT_FAILED from class production error to development

▽

△

2013-03-15	4.1.1	AUTOSAR Administration	• Added parameters for Event Channel and Timestamp configuration • Added possibility to calculate memory consumption for ODT (DAQ & STIM) • Restructuring configuration parameters for static & dynamic ODT • Added support for deactivation of transmission capabilities
2011-12-22	4.0.3	AUTOSAR Administration	• Add chapter 7.8 (Version check), RTE limitation, OS Counter Ref • Remove InstanceID and known limitation (OS)
2010-09-30	3.1.5	AUTOSAR Administration	• Initial Release

Disclaimer

Contents

1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module XCP

XCP is a protocol description (ASAM standard) between a master (tool) and a slave (device), which provides the following basic features:

- Synchronous data acquisition (measurement)
- Synchronous data stimulation (for rapid prototyping)
- Online memory calibration (read / write access)
- Calibration data page initialization and switching
- Flash Programming for ECU development purposes
- Every feature is optional and the access can be restricted
- Various communications busses are supported

XCP was designed according to the following principles:

- Minimal Slave resource consumption (RAM, ROM, runtime)
- Efficient communication
- Simple Slave implementation

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the XCP that are not included in the [1, AUTOSAR glossary].

Acronym:	Description:
AUTOSAR	AUTomotive Open System ARchitecture
A2L	File Extension for an ASAM 2MC Language File
ASAM	Association for Standardization of Automation and Measuring Systems
BSW	Basic Software
CAN	Controller Area Network
CanIf	CAN Interface
CTO	Command Transfer Object
DAQ	Data AcQuisition, Data AcQuisition Packet
DTO	Data Transfer Object
ECU	Electronic Control Unit
FrIf	FlexRay Interface
LPDU	Data Link Layer PDU
MCD	Measurement Calibration and Diagnostics
MISRA	Motor Industry Software Reliability Association
ODT	Object Descriptor Table
PDU	Protocol Data Unit
RAM	Random Access Memory
ROM	Read Only Memory
SchM	Schedule Manager
SVN	Subversion
SRS	Software Requirements Specification
STIM	Data Stimulation packet
SW	Software
SWS	Software Specification
TCP/IP	Transfer Control Protocol / Internet Protocol
TS	Time Stamp
UDP/IP	User Datagram Protocol / Internet Protocol
URL	Uniform Resource Locator
XCP	Universal Calibration Protocol
XML	Extensible Markup Language
ISR	Interrupt Service Routine
DET	Default Error Tracer (AUTOSAR BSW module)

Table 2.1: Acronyms used in the scope of this Document

3 Related documentation

3.1 Input documents & related standards and norms

[1] Glossary
AUTOSAR_FO_TR_Glossary

[2] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[3] ASAM XCP - The Universal Measurement and Calibration Protocol: ASAM_XCP_Part1-Overview - Version 1.1
http://www.asam.net

[4] Specification of PDU Router
AUTOSAR_CP_SWS_PDURouter

[5] Specification of Linklayer Sdu Routing Module
AUTOSAR_CP_SWS_LSduRouter

[6] General Requirements on Basic Software Modules
AUTOSAR_CP_RS_BSWGeneral

[7] Requirements on Module XCP
AUTOSAR_CP_RS_XCP

[8] ASAM XCP - Transport Layer Specification XCP on CAN: ASAM_XCP_Part3 Transport-Layer-Specification_XCPonCAN - Version 1.2
http://www.asam.net

[9] Specification of CAN Interface
AUTOSAR_CP_SWS_CANInterface

[10] ASAM XCP - Transport Layer Specification XCP on FlexRay: ASAM_XCP_Part3-Transport-Layer-Specification_XCPonFlexRay-Version 1.1
http://www.asam.net

[11] Specification of FlexRay Interface
AUTOSAR_CP_SWS_FlexRayInterface

[12] ASAM XCP - Transport Layer Specification XCP on Ethernet: ASAM_XCP_Part3-Transport-Layer-Specification_XCPonEthernet (TCP_IP&UDP_IP) - Version 1.1
http://www.asam.net

[13] Specification of Socket Adaptor
AUTOSAR_CP_SWS_SocketAdaptor

3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [2, SWS BSW General], which is also valid for XCP.

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

4 Constraints and assumptions

4.1 Limitations

The following XCP features are currently out of scope:

- The SET_DAQ_ID command according to the XCP CAN Transport Layer Specification is not part of the AUTOSAR XCP module
- Currently, the AUTOSAR RTE does not offer APIs for direct communication with XCP
- For further details concerning the supported feature set, please refer to [3]
- NAX is only configurable through the ASAM configuration file A2L.

Please note:

For the communications bus LIN, no ASAM XCP is specified.

4.2 Applicability to car domains

n/a

5 Dependencies to other modules

This section describes the relations to other modules and files within the AUTOSAR basic software architecture. It contains brief descriptions of configuration information and services, which are required by the XCP module from other modules.

5.1 AUTOSAR RTE (BSW Scheduler)

The BSW Scheduler calls the main functions of the Xcp, which are necessary for the cyclic processes of the Xcp.

5.2 AUTOSAR PDU Routing module (PduR)

The PDU Routing module (PduR) [4] is the lower layer module of the XCP module for the data flow. Please note: The Linklayer Sdu Routing Module (LSduR) [5] is the lower layer of the PduR. XCP data are excanged via the PduR and LSduR with the according Bus interfaces.

5.3 AUTOSAR FlexRay Interface

The FlexRay Interface is used to transmit and receive XCP PDUs via FlexRay.

5.4 AUTOSAR CAN Interface

The CAN Interface is used to transmit and receive XCP PDUs via CAN.

5.5 AUTOSAR SocketAdaptor

The SocketAdaptor is used to transmit and receive XCP PDUs via Ethernet.

5.6 AUTOSAR RTE

The RTE is used for copying calibration parameters from ROM/FLASH to RAM and to use the double pointered method

5.7 AUTOSAR OS

In order to be able to use the time stamped feature of XCP, an AUTOSAR OS Counter is used.

5.8 AUTOSAR Diagnostic Event Manager

In order to be able to report production errors, the XCP has to have access to the Diagnostic Event Manager.

5.9 AUTOSAR Default Error Tracer

In order to be able to report default errors, the XCP has to have access to the error hook of the Default Error Tracer.

5.10 File structure

5.10.1 Code file structure

[SWS_Xcp_00501]
Upstream requirements: SRS_BSW_00419, SRS_BSW_00383, SRS_BSW_00346, SRS_BSW_-00380

⌈The code file structure shall not be defined within this specification completely. At this point it shall be pointed out that the code-file structure shall include the following files named:

- Xcp.c - general source code file of the module XCP
- Xcp_Cfg.c - for pre-compile time configurable parameters
- Xcp_Lcfg.c - for link time configurable parameters and
- Xcp_PBcfg.c - for post build time configurable parameters.

⌋

These files shall contain all link time and post-build time configurable parameters.

6 Requirements Tracing

The following tables reference the requirements specified in [6] and [7] and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[SRS_BSW_00003]	All software modules shall provide version and identification information	[SWS_Xcp_00807]
[SRS_BSW_00101]	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	[SWS_Xcp_00803]
[SRS_BSW_00159]	All modules of the AUTOSAR Basic Software shall support a tool based configuration	[SWS_Xcp_00102]
[SRS_BSW_00167]	All AUTOSAR Basic Software Modules shall provide configuration rules and constraints to enable plausibility checks	[SWS_Xcp_00103] [SWS_Xcp_00104] [SWS_Xcp_00105]
[SRS_BSW_00318]	Each AUTOSAR Basic Software Module file shall provide version numbers in the header file	[SWS_Xcp_00807]
[SRS_BSW_00327]	Error values naming convention	[SWS_Xcp_00763]
[SRS_BSW_00344]	BSW Modules shall support link-time configuration	[SWS_Xcp_00741]
[SRS_BSW_00345]	BSW Modules shall support pre-compile configuration	[SWS_Xcp_00742]
[SRS_BSW_00346]	All AUTOSAR Basic Software Modules shall provide at least a basic set of module files	[SWS_Xcp_00501]
[SRS_BSW_00358]	The return type of init() functions implemented by AUTOSAR Basic Software Modules shall be void	[SWS_Xcp_00803]
[SRS_BSW_00373]	The main processing function of each AUTOSAR Basic Software Module shall be named according the defined convention	[SWS_Xcp_00823]
[SRS_BSW_00374]	All Basic Software Modules shall provide a readable module vendor identification	[SWS_Xcp_00807]
[SRS_BSW_00379]	All software modules shall provide a module identifier in the header file and in the module XML description file.	[SWS_Xcp_00807]
[SRS_BSW_00380]	Configuration parameters being stored in memory shall be placed into separate c-files	[SWS_Xcp_00501]
[SRS_BSW_00383]	The Basic Software Module specifications shall specify which other configuration files from other modules they use at least in the description	[SWS_Xcp_00501]
[SRS_BSW_00402]	Each module shall provide version information	[SWS_Xcp_00807]
[SRS_BSW_00404]	BSW Modules shall support post-build configuration	[SWS_Xcp_00742]

△

Requirement	Description	Satisfied by
[SRS_BSW_00405]	BSW Modules shall support multiple configuration sets	[SWS_Xcp_00803]
[SRS_BSW_00407]	Each BSW module shall provide a function to read out the version information of a dedicated module implementation	[SWS_Xcp_00807]
[SRS_BSW_00411]	All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API	[SWS_Xcp_00807]
[SRS_BSW_00414]	Init functions shall have a pointer to a configuration structure as single parameter	[SWS_Xcp_00803]
[SRS_BSW_00419]	If a pre-compile time configuration parameter is implemented as `const` it should be placed into a separate c-file	[SWS_Xcp_00501]
[SRS_BSW_00424]	BSW module main processing functions shall not be allowed to enter a wait state	[SWS_Xcp_00823]
[SRS_BSW_00433]	Main processing functions are only allowed to be called from task bodies provided by the BSW Scheduler	[SWS_Xcp_00823]
[SRS_Xcp_29001]	The AUTOSAR XCP module shall be located above the bus interfaces / Socket Adaptor	[SWS_Xcp_00701] [SWS_Xcp_00860]
[SRS_Xcp_29002]	The AUTOSAR XCP shall make use of the data transmit- and receive APIs of the Bus Interfaces	[SWS_Xcp_00712] [SWS_Xcp_00714] [SWS_Xcp_00720] [SWS_Xcp_00734] [SWS_Xcp_00862]
[SRS_Xcp_29003]	The AUTOSAR XCP messages shall be identified by unique PDU-IDs	[SWS_Xcp_00702] [SWS_Xcp_CONSTR_00861]
[SRS_Xcp_29004]	The XCP Specification Version 1.1 shall be used	[SWS_Xcp_00703]
[SRS_Xcp_29005]	XCP on CAN shall be supported	[SWS_Xcp_00713]
[SRS_Xcp_29006]	XCP on FlexRay shall be supported	[SWS_Xcp_00719]
[SRS_Xcp_29007]	XCP on Ethernet shall be supported	[SWS_Xcp_00733]
[SRS_Xcp_29008]	The code generator of the XCP Module shall generate the A2L IF_ DATA section	[SWS_Xcp_00853]
[SRS_Xcp_29009]	The slave shall transfer the contents of the elements defined in each ODT of the DAQ-list to the master	[SWS_Xcp_00705]
[SRS_Xcp_29010]	Synchronous Data Stimulation shall be the inverse mode of Synchronous Data Acquisition	[SWS_Xcp_00707]
[SRS_Xcp_29012]	The XCP master shall already send the next request before having received the response on the previous request	[SWS_Xcp_00710]
[SRS_Xcp_29013]	It shall be possible to configure the DAQ Lists dynamically	[SWS_Xcp_00706]
[SRS_Xcp_29014]	It shall be possible to transmit a timestamp within the XCP packet	[SWS_Xcp_00709]

▽

△

Requirement	Description	Satisfied by
[SRS_Xcp_29015]	It shall be possible to bypass data by making use of Synchronous Data Acquisition and Synchronous Data Stimulation simultaneously	[SWS_Xcp_00761]
[SRS_Xcp_29016]	The feature "Seed&Key" shall be used for protection handling purpose	[SWS_Xcp_00766]
[SRS_Xcp_29017]	The AUTOSAR XCP module shall implement an interface for initialization.	[SWS_Xcp_00803]
[SRS_Xcp_29018]	Page switching shall be supported	[SWS_Xcp_00852]
[SRS_Xcp_29019]	DAQ configuration storing with power-up data transfer (RESUME mode) shall be supported	[SWS_Xcp_00854]
[SRS_Xcp_29020]	Flash Programming for ECU development purposes	[SWS_Xcp_00855] [SWS_Xcp_00856]
[SRS_Xcp_29030]	The AUTOSAR XCP module shall be located above the PduR	[SWS_Xcp_00863]
[SRS_Xcp_29031]	The AUTOSAR XCP shall make use of the data transmit- and receive APIs of the PduR	[SWS_Xcp_00864] [SWS_Xcp_00865] [SWS_Xcp_00866]

Table 6.1: Requirements Tracing

7 Functional specification

The specification of the module XCP shall define all parameters and interfaces, which are required to use the ASAM XCP protocol specification within an AUTOSAR environment.

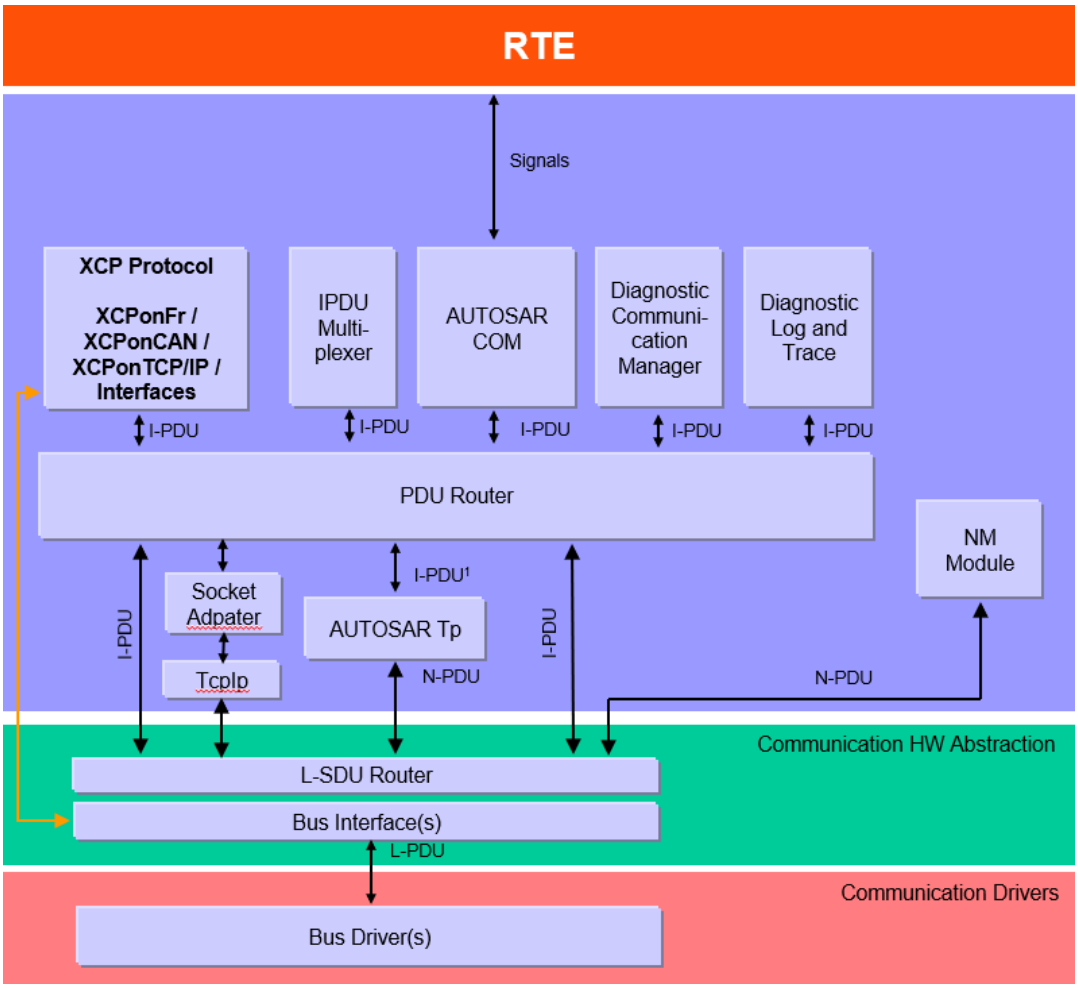

Figure 7.1: Description

Black arrows: Data Path (Signals/Pdus)
Orange arrows: Control Path (FlexRay Interface)

[SWS_Xcp_00701]

Status:	OBSOLETE
Use instead:	SWS_Xcp_00860
Upstream requirements:	SRS_Xcp_29001

⌈The AUTOSAR XCP Module be located above the bus specific Interfaces in case of FlexRay and Can. In case of Ethernet, the AUTOSAR XCP module shall be located above the Socket Adaptor.⌋

[SWS_Xcp_00860] Location of the XCP module in the AUTOSAR communication stack

Status: DRAFT
Replaces: SWS_Xcp_00701
Upstream requirements: SRS_Xcp_29001

⌈The AUTOSAR XCP Module and the bus specific Interfaces shall be located above the AUTOSAR PDU Routing module (PduR) [4].⌋

[SWS_Xcp_00702]

Upstream requirements: SRS_Xcp_29003

⌈For transmitting and receiving of XCP messages, unique PDU-IDs shall be used.⌋

[SWS_Xcp_CONSTR_00861] `XcpRxPdu` and `XcpTxPdu` constraint for keeping the local buffer

Status: DRAFT
Upstream requirements: SRS_Xcp_29003

⌈Each `XcpRxPdu` and `XcpTxPdu` shall refer to global PDU that has `KeepLocalPduBuffer` set to FALSE.⌋

[SWS_Xcp_00703]

Upstream requirements: SRS_Xcp_29004

⌈The AUTOSAR XCP Module shall support the ASAM XCP Specification Version 1.1, except for XCP on CAN where ASAM XCP Specification Version 1.2 shall be supported.⌋

[SWS_Xcp_00705]

Upstream requirements: SRS_Xcp_29009

⌈The AUTOSAR XCP Module shall support the basic feature "Synchronous data acquisition (measurement) ". Please refer to [3]⌋

[SWS_Xcp_00706]

Upstream requirements: SRS_Xcp_29013

⌈The AUTOSAR XCP Module shall support the feature "Dynamic DAQ Configuration". according to [3]⌋

[SWS_Xcp_00707]

Upstream requirements: SRS_Xcp_29010

⌈The AUTOSAR XCP Module shall support the basic feature "Synchronous data stimulation" according to [3]⌋

[SWS_Xcp_00708] ⌈The AUTOSAR XCP Module shall support the basic feature "Online memory calibration (read / write access) ", according to [3]⌋

[SWS_Xcp_00709]

Upstream requirements: SRS_Xcp_29014

⌈The AUTOSAR XCP Module shall support the feature "Timestamped Data Transfer", according to [3]⌋

[SWS_Xcp_00768] ⌈The ECU local time shall be derived from the AUTOSAR OS.⌋

[SWS_Xcp_00711] ⌈The AUTOSAR XCP Module shall support the feature "Block communication mode", according to [3]⌋

[SWS_Xcp_00761]

Upstream requirements: SRS_Xcp_29015

⌈The AUTOSAR XCP Module shall support the feature "Bypassing", according to [3]⌋

[SWS_Xcp_00766]

Upstream requirements: SRS_Xcp_29016

⌈The AUTOSAR XCP Module shall support the feature "Seed & Key" according to [3]⌋

[SWS_Xcp_00712]

Status: OBSOLETE
Use instead: SWS_Xcp_00862
Upstream requirements: SRS_Xcp_29002

⌈For sending and receiving of calibration data, the sending and receiving APIs specified within the AUTOSAR BSW Bus Interfaces (FlexRay Interface, CAN Interface, TCP/IP Socket Adaptor) shall be used. Please refer to chapter 7.1, 7.2 and 7.3.⌋

[SWS_Xcp_00862] Sending and receiving of calibration data

Status: DRAFT
Replaces: SWS_Xcp_00712
Upstream requirements: SRS_Xcp_29002

⌈For sending and receiving of calibration via CAN, FlexRay and Ethernet (i.e. TCP/IP), the sending and receiving APIs specified within the AUTOSAR PDU Routing module (PduR) [4] shall be used. Please refer to chapter 7.1, 7.2 and 7.3.⌋

Note: AUTOSAR PDU Routing module (PduR) [4] and AUTOSAR BSW Linklayer Sdu Routing Module (LSduR) route the XCP data as PDU to the according AUTOSAR BSW Bus Interfaces (CAN Interface, FlexRay Interface and Ethernet Interface (via Socket Adapter and TcpIp module)). The transmission and reception paths differ slighlty.

For example the transmission path is denoted in the following items. Please refer to Chapter 9 for more details:

- XCP data transmitted on CAN or FlexRay: `XCP -> PduR -> LSduR -> <Can|Fr>If`
- XCP data transmitted on Ethernet (via. TcpIp): `XCP -> PduR -> SoAd -> TcpIp -> LSduR -> EthIf`

[SWS_Xcp_00852]

Upstream requirements: SRS_Xcp_29018

⌈The AUTOSAR XCP Module shall support the feature "Page switching", according to [3]⌋

[SWS_Xcp_00853]

Upstream requirements: SRS_Xcp_29008

⌈The code generator of the XCP Module shall generate the A2L IF_DATA section, based on the configuration of XCP⌋

[SWS_Xcp_00854]

Upstream requirements: SRS_Xcp_29019

⌈The AUTOSAR XCP Module shall support the feature "Power-Up data transfer (RESUME MODE)", according to [3]⌋

[SWS_Xcp_00855]

Upstream requirements: SRS_Xcp_29020

⌈The AUTOSAR XCP Module shall support the flash programming (PGM) according to [3]⌋

[SWS_Xcp_00856]

Upstream requirements: SRS_Xcp_29020

⌈Indication the end of a programming sequence is supported using the optional command "PROGRAM_RESET", where the slave will go to disconnected state but without forcing a device reset⌋

[SWS_Xcp_00859]

Status: OBSOLETE
Use instead: SWS_Xcp_00863

⌈The XCP module shall wait for the Xcp_<Lo>TxConfirmation (positive or negative) after each call to <Lo>_Transmit to avoid overwriting previously transmitted data⌋

[SWS_Xcp_00863] Waiting for call of `Xcp_TxConfirmation` to avoid overwritting previously transmitted data

Status: DRAFT
Replaces: SWS_Xcp_00859
Upstream requirements: SRS_Xcp_29030

⌈FThe XCP module shall wait for the `Xcp_TxConfirmation` (positive or negative) after each call to `PduR_XcpTransmit` to avoid overwriting previously transmitted data.⌋

7.1 XCP on CAN

[SWS_Xcp_00713]

Upstream requirements: SRS_Xcp_29005

⌈The AUTOSAR XCP Module shall support the CAN communications bus according to [8]⌋

[SWS_Xcp_00714]

Status: OBSOLETE
Use instead: SWS_Xcp_00864
Upstream requirements: SRS_Xcp_29002

⌈XCP data sent and received via CAN, the PDUs have to be transmitted and received using the transmitting and receive APIs provided by the AUTOSAR CAN Interface, according to [9]⌋

[SWS_Xcp_00864] Use APIs of PduR for exchanging data via CAN

Status: DRAFT
Replaces: SWS_Xcp_00714
Upstream requirements: SRS_Xcp_29031

⌈XCP data sent and received via CAN, the PDUs shall be transmitted and received using the transmit and receive APIs provided by the AUTOSAR PDU Routing module (PduR) [4].⌋

[SWS_Xcp_00715] ⌈For sending and receiving XCP data via CAN, at least two different CAN identifiers have to be configured to be used by XCP.⌋

[SWS_Xcp_00716] ⌈Performance information shall be exchanged between the XCP master and XCP slave using the parameters according to [8]⌋

[SWS_Xcp_00718] ⌈The XCP Module shall support the GET_SLAVE_ID command according to [8]⌋

7.2 XCP on FlexRay

[SWS_Xcp_00719]

Upstream requirements: SRS_Xcp_29006

⌈The AUTOSAR XCP Module shall support the FlexRay communications bus according to [10]⌋

[SWS_Xcp_00720]

Status: OBSOLETE
Use instead: SWS_Xcp_00865
Upstream requirements: SRS_Xcp_29002

⌈XCP data sent and received via FlexRay, the PDUs have to be transmitted and received using the transmit and receive APIs provided by the AUTOSAR FlexRay Interface according to [11].⌋

[SWS_Xcp_00865] Use APIs of PduR for exchanging data via FlexRay

Status: DRAFT
Replaces: SWS_Xcp_00720
Upstream requirements: SRS_Xcp_29031

⌈XCP data sent and received via FlexRay, the PDUs shall be transmitted and received using the transmit and receive APIs provided by the AUTOSAR PDU Routing module (PduR) [4].⌋

[SWS_Xcp_00721] ⌈All XCP on FlexRay LPDUs always are event driven. Please refer to Chapter 1.1.2 "FlexRay Frame Type" of [10]⌋

[SWS_Xcp_00722] ⌈The hardware buffers (of the FlexRay Communication Controller) XCP uses for data transmission and reception are assigned exclusively to the XCP module.⌋

Note:

This restriction prevents disturbances of ongoing FlexRay communication.

[SWS_Xcp_00723] ⌈The usage of FlexRay Communication Controller's hardware buffers shall be configured by the corresponding parameters according to [10]⌋

[SWS_Xcp_00724] ⌈The FlexRay PDU length used by the AUTOSAR XCP module shall be set using the corresponding parameters according to [10]⌋

[SWS_Xcp_00725] ⌈LPDU_IDs which shall be routed to the AUTOSAR XCP module (using the AUTOSAR Bus Interface) have to be defined by the system designer.⌋

[SWS_Xcp_00726] ⌈The ASAM MCD 2MC description file (i.e. A2L file) describes to which extent the XCP-dedicated buffers of a specific slave can be configured for XCP communication.⌋

[SWS_Xcp_00728] ⌈The XCP master gets the information about the XCP dedicated FlexRay Communication Controller buffers from the ASAM MCD 2MC description file.⌋

[SWS_Xcp_00729] ⌈Limitations due to the usage of multiple XCP slaves on the FlexRay communications bus shall be taken into consideration by the system designer. Please refer to [10].⌋

[SWS_Xcp_00730] ⌈Depending upon the requirements on sequencing correctness, alignment and net data throughput, different header types are possible. Please refer to Chapter 1.4.1 "Header" of [10]⌋

[SWS_Xcp_00731] ⌈For XCP on FlexRay, the Tail consists of a Control Field containing optional FILL bytes according to [10].⌋

[SWS_Xcp_00732] ⌈The AUTOSAR XCP module shall be able to pack multiple XCP messages into one FlexRay Frame according to [10].⌋

7.3 XCP on Ethernet

[SWS_Xcp_00733]

Upstream requirements: SRS_Xcp_29007

⌈The AUTOSAR XCP Module shall support the Ethernet communications bus according to [12]⌋

[SWS_Xcp_00734]

Status:	OBSOLETE
Use instead:	SWS_Xcp_00866
Upstream requirements:	SRS_Xcp_29002

⌈XCP data sent and received via Ethernet, the PDUs have to be transmitted and received using the transmitting and receive APIs provided by the AUTOSAR Socket Adaptor according to [13].⌋

[SWS_Xcp_00866] Use APIs of PduR for exchanging data via Ethernet (i.e. TcpIp)

Status: DRAFT
Replaces: SWS_Xcp_00734
Upstream requirements: SRS_Xcp_29031

⌈XCP data sent and received via Ethernet, the PDUs have to be transmitted and received using the transmitting and receive APIs provided by the AUTOSAR PDU Router according to [4].⌋

[SWS_Xcp_00735] ⌈The AUTOSAR XCP slave connected by Ethernet and TCP/IP or UDP/IP is addressed by its IP Address and Port number.⌋

[SWS_Xcp_00736] ⌈The AUTOSAR XCP slave only accepts one connection at the time.⌋

[SWS_Xcp_00737] ⌈If the socket is closed while in XCP connected state, the slave device will perform an XCP disconnect, which means that all data acquisition will be stopped.⌋

[SWS_Xcp_00738] ⌈The addressing scheme is defined according to [12]⌋

.

[SWS_Xcp_00739] ⌈The header and tail of an XCP on Ethernet message have to be set according to [12]⌋

[SWS_Xcp_00740] ⌈The upper performance limit depends on the protocol stack of the host system. The corresponding parameters defined according to [12] have to be set.⌋

[SWS_Xcp_00710]

Upstream requirements: SRS_Xcp_29012

⌈The AUTOSAR XCP Module shall support the feature "Interleaved communication mode", according to according to [3]⌋

7.4 General Requirements

[SWS_Xcp_00741]

Upstream requirements: SRS_BSW_00344

⌈Link-time and post-build-time configuration data shall be implemented as read-only data structures. Link-time configuration data shall be immediately referenced by the

implementation, the start-address of post-build-time configuration data shall be passed during module initialization⌋

[SWS_Xcp_00742]

Upstream requirements: SRS_BSW_00404, SRS_BSW_00345

⌈The XCP module shall support pre-compile time, link-time and post-build-time configuration.⌋

7.5 Error Classification

[SWS_Xcp_00763]

Upstream requirements: SRS_BSW_00327

⌈The error values and EventIds are named in capital letters according to the scheme XCP_E_<NAME>, where NAME describes the error/EventId and may consist of several words separated by underscores.⌋

7.5.1 Development Errors

[SWS_Xcp_00857] Definiton of development errors in module Xcp ⌈

Type of error	*Related error code*	*Error value*
Module not initialized	XCP_E_UNINIT	0x02
API call with wrong PDU ID	CP_E_INVALID_PDUID	0x03
Initialization of XCP failed	XCP_E_INIT_FAILED	0x04
Null pointer has been passed as an argument	XCP_E_PARAM_POINTER	0x12

⌋

7.5.2 Runtime Errors

There are no runtime errors.

7.5.3 Transient Faults

There are no transient faults.

7.5.4 Production Errors

There are no production errors.

7.5.5 Extended Production Errors

There are no extended production errors.

7.6 Version checking

For details refer to the chapter 5.1.8 "Version Check" in SWS_BSWGeneral.

7.7 Security Events

The module does not report security events.

8 API specification

8.1 Imported types

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

[SWS_Xcp_00801] Definition of imported datatypes of module Xcp ⌈

Module	*Header File*	*Imported Type*
Comtype	ComStack_Types.h	NetworkHandleType
	ComStack_Types.h	PduIdType
	ComStack_Types.h	PduInfoType
	ComStack_Types.h	PduLengthType
Fr	Fr_GeneralTypes.h	Fr_ChannelType
Os	Os.h	StatusType
	Os.h	TickRefType
	Os.h	TickType
	Rte_Os_Type.h	CounterType
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType

⌋

8.2 Type definitions

8.2.1 Xcp_ConfigType

[SWS_Xcp_00845] Definition of datatype Xcp_ConfigType ⌈

Name	Xcp_ConfigType	
Kind	Structure	
Elements	implementation specific	
	Type	–
	Comment	The content of the initialization data structure is implementation specific
Description	This is the type of the data structure containing the initialization data for XCP.	
Available via	Xcp.h	

⌋

8.2.2 Xcp_Transmission Mode Type

[SWS_Xcp_00846] Definition of datatype Xcp_TransmissionModeType ⌈

Name	Xcp_TransmissionModeType		
Kind	Enumeration		
Range	XCP_TX_OFF	0x00	Transmission Disabled
	XCP_TX_ON	0x01	Transmission Enabled
Description	Handles the enabling and disabling of the transmission mode		
Available via	Xcp.h		

⌋

8.3 Function definitions

This is a list of functions provided for upper layer modules.

8.3.1 Xcp_Init

[SWS_Xcp_00803] Definition of API function Xcp_Init

Upstream requirements: SRS_BSW_00405, SRS_BSW_00101, SRS_BSW_00358, SRS_BSW_-00414, SRS_Xcp_29017

⌈

Service Name	Xcp_Init	
Syntax	`void Xcp_Init ( const Xcp_ConfigType* Xcp_ConfigPtr )`	
Service ID [hex]	0x00	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	Xcp_ConfigPtr	Pointer to a selected configuration structure
Parameters (inout)	None	
Parameters (out)	None	
Return value	void	–
Description	This service initializes interfaces and variables of the AUTOSAR XCP layer.	
Available via	Xcp.h	

⌋

[SWS_Xcp_00802] ⌈The function `Xcp_Init` shall internally store the configuration address to enable subsequent API calls to access the configuration⌋

8.3.2 Xcp_GetVersionInfo

[SWS_Xcp_00807] Definition of API function Xcp_GetVersionInfo

Upstream requirements: SRS_BSW_00402, SRS_BSW_00407, SRS_BSW_00411, SRS_BSW_-00374, SRS_BSW_00379, SRS_BSW_00003, SRS_BSW_00318

⌈

Service Name	Xcp_GetVersionInfo	
Syntax	`void Xcp_GetVersionInfo ( Std_VersionInfoType* versioninfo )`	
Service ID [hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	versioninfo	Pointer to where to store the version information of this module.
Return value	void	–
Description	Returns the version information of this module.	
Available via	Xcp.h	

⌋

[SWS_Xcp_00825] ⌈If development error detection for the Xcp module is enabled, then the function `Xcp_GetVersionInfo` shall check whether the parameter VersioninfoPtr is a NULL pointer (NULL_PTR). If VersioninfoPtr is a NULL pointer, then the function `Xcp_GetVersionInfo` shall raise the development error XCP_E_PARAM_POINTER and return.⌋

8.3.3 Xcp_SetTransmissionMode

[SWS_Xcp_00844] Definition of callback function Xcp_SetTransmissionMode ⌈

Service Name	Xcp_SetTransmissionMode	
Syntax	`void Xcp_SetTransmissionMode ( NetworkHandleType Channel, Xcp_TransmissionModeType Mode )`	
Service ID [hex]	0x05	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	Channel	The Network channel for the used bus communication
	Mode	Enabled or disabled Transmission mode Parameters
Parameters (inout)	None	

▽

△

Parameters (out)	None
Return value	None
Description	This API is used to turn on and off of the TX capabilities of used communication bus channel in XCP module.
Available via	Xcp.h

⌋

[SWS_Xcp_00848] ⌈The XCP module shall provide this service only if XCP_SUPPRESS_TX_SUPPORT (see [ECUC_Xcp_00169]) equals TRUE.⌋

[SWS_Xcp_00849] ⌈If `Xcp_SetTransmissionMode`(Channel, Mode) is called and parameter Mode equals XCP_TX_OFF, all TxPDUs which are assigned to Channel shall not be transmitted.⌋

Note: It could be derived from <Bus>If configuration and the global PDU parameter, to which specific communication channel the PDU is assigned to.

[SWS_Xcp_00850] ⌈If `Xcp_SetTransmissionMode`(Channel, Mode) is called and parameter Mode equals XCP_TX_ON, all TxPDUs which are assigned to Channel shall be able to be transmitted.⌋

8.4 Callback notifications

[SWS_Xcp_00836] ⌈This is a list of functions provided for other modules.⌋

8.4.1 Xcp_RxIndication

[SWS_Xcp_00813] Definition of callback function Xcp_RxIndication ⌈

Service Name	Xcp_RxIndication

Syntax	`void Xcp_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 PDU.
	PduInfoPtr	Contains the length (SduLength) of the received PDU, a pointer to a buffer (SduDataPtr) containing the PDU, and the MetaData related to this PDU.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Indication of a received PDU from a lower layer communication interface module.	
Available via	Xcp.h	

⌋

The callback function `Xcp_RxIndication` is called by the Bus Interfaces, Ethernet Socket Adaptor or CDD and is implemented by the Xcp module.

[SWS_Xcp_00847]
Status: OBSOLETE
Use instead: SWS_Xcp_00867

⌈The callback function `Xcp_<Lo>RxIndication` shall inform the DET, if development error detection is enabled (XCP_DEV_ERROR_DETECT is set to TRUE) and if function call has failed because of the following reasons:

- Xcp was not initialized (XCP_E_UNINIT)
- PduInfoPtr equals NULL_PTR (XCP_E_PARAM_POINTER)
- Invalid PDUID (XCP_E_INVALID_PDUID)

⌋

[SWS_Xcp_00867] Development error handling in context of `Xcp_RxIndication`
Status: DRAFT
Replaces: SWS_Xcp_00847

⌈The callback function `Xcp_RxIndication` shall inform the DET, if development error detection is enabled (XCP_DEV_ERROR_DETECT is set to TRUE) and if function call has failed because of the following reasons:

- Xcp was not initialized (XCP_E_UNINIT)
- PduInfoPtr equals NULL_PTR (XCP_E_PARAM_POINTER)
- Invalid PDUID (XCP_E_INVALID_PDUID)

⌋

The function `Xcp_RxIndication` shall be called by the Xcp module's environment in an interrupt context.

8.4.2 Xcp_TxConfirmation

[SWS_Xcp_00814] Definition of callback function Xcp_TxConfirmation ⌈

Service Name	Xcp_TxConfirmation	
Syntax	`void Xcp_TxConfirmation ( PduIdType TxPduId, Std_ReturnType result )`	
Service ID [hex]	0x40	
Sync/Async	Synchronous	
Reentrancy	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in)	TxPduId	ID of the PDU that has been transmitted.
	result	E_OK: The PDU was transmitted. E_NOT_OK: Transmission of the PDU failed.
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU.	
Available via	Xcp.h	

⌋

Note:

The callback function `Xcp_TxConfirmation` is called by the Bus Interfaces, Ethernet Socket Adaptor or CDD and is implemented by the Xcp module.

[SWS_Xcp_00840]
Status: OBSOLETE
Use instead: SWS_Xcp_00868

⌈If development error detection for the XCP module is enabled: if the function `Xcp_<Lo>TxConfirmation` is called before the XCP was initialized successfully, the function `Xcp_<Lo>TxConfirmation` shall raise the development error XCP_E_UNINIT and return.⌋

[SWS_Xcp_00868] Development error handling in context of `Xcp_TxConfirmation`
Status: DRAFT
Replaces: SWS_Xcp_00840

⌈If development error detection for the XCP module is enabled: if the function `Xcp_TxConfirmation` is called before the XCP was initialized successfully, the function `Xcp_TxConfirmation` shall raise the development error XCP_E_UNINIT and return.⌋

[SWS_Xcp_00841]
Status: OBSOLETE
Use instead: SWS_Xcp_00869

⌈Caveats of `Xcp_<Lo>TxConfirmation`:

- The call context is either on interrupt level (interrupt mode) or on task level
- The Xcp module is initialized correctly.

⌋

[SWS_Xcp_00869] Caveats on call of `Xcp_TxConfirmation`
Status: DRAFT
Replaces: SWS_Xcp_00841

⌈Caveats of `Xcp_TxConfirmation`:

- The call context is either on interrupt level (interrupt mode) or on task level
- The Xcp module is initialized correctly.

⌋

8.4.3 Xcp_TriggerTransmit

[SWS_Xcp_00835] Definition of callback function Xcp_TriggerTransmit ⌈

Service Name	Xcp_TriggerTransmit	
Syntax	`Std_ReturnType Xcp_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 (inout)	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 number 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.
Available via	Xcp.h

⌋

Note:

The callback function `Xcp_TriggerTransmit` is called by the Bus Interfaces, Ethernet Socket Adaptor or CDD and is implemented by the Xcp module.

[SWS_Xcp_00842]
Status: OBSOLETE
Use instead: SWS_Xcp_00870

⌈If development error detection for the XCP module is enabled: if the function

`Xcp_<Lo>TriggerTransmit` is called before the XCP was initialized successfully, the function `Xcp_<Lo>TriggerTransmit` shall raise the development error XCP_E_UNINIT and return E_NOT_OK.⌋

[SWS_Xcp_00870] Development error handling in context of `Xcp_TriggerTransmit`
Status: DRAFT
Replaces: SWS_Xcp_00842

⌈If development error detection for the XCP module is enabled: if the function

`Xcp_TriggerTransmit` is called before the XCP was initialized successfully, the function `Xcp_TriggerTransmit` shall raise the development error XCP_E_UNINIT and return E_NOT_OK.⌋

[SWS_Xcp_00843]
Status: OBSOLETE
Use instead: SWS_Xcp_00871

⌈Caveats of `Xcp_<Lo>TriggerTransmit`:

- The call context is either on interrupt level (interrupt mode) or on task level
- The Xcp module is initialized correctly.

⌋

[SWS_Xcp_00871] Caveats on call of `Xcp_TriggerTransmit`
Status: DRAFT
Replaces: SWS_Xcp_00843

⌈Caveats of `Xcp_TriggerTransmit`:

- The call context is either on interrupt level (interrupt mode) or on task level
- The Xcp module is initialized correctly.

⌋

8.5 Scheduled functions

The functions are called directly by Basic Software Scheduler. The following functions shall have no return value and no parameter. All functions shall be non reentrant.

8.5.1 Xcp_MainFunction

[SWS_Xcp_00823] Definition of scheduled function Xcp_MainFunction
Upstream requirements: SRS_BSW_00424, SRS_BSW_00433, SRS_BSW_00373

⌈

Service Name	Xcp_MainFunction
Syntax	`void Xcp_MainFunction ( void )`
Service ID [hex]	0x04
Description	Scheduled function of the XCP module
Available via	SchM_Xcp.h

⌋

[SWS_Xcp_00824] ⌈The XCP Main Function shall be called cyclically.⌋

8.6 Expected interfaces

In this chapter all interfaces required from other modules are listed.

8.6.1 Mandatory interfaces

[SWS_Xcp_91001] Definition of mandatory interfaces required by module Xcp ⌈

API Function	Header File	Description
There are no mandatory interfaces.		

⌋

8.6.2 Optional interfaces

[SWS_Xcp_00832] Definition of optional interfaces requested by module Xcp ⌈

API Function	Header File	Description
Det_ReportError	Det.h	Service to report development errors.
FrIf_DisableLPdu	FrIf.h	Wraps the FlexRay Driver Function Fr_DisableLPdu. It disables the hardware resource of an LPdu for transmission/reception.
FrIf_ReconfigLPdu	FrIf.h	Calls the FlexRay Driver's API Fr_ReconfigLPdu. The enum value "FR_CHANNEL_AB" shall not be used.
GetCounterValue	Os.h	This service reads the current count value of a counter (returning either the hardware timer ticks if counter is driven by hardware or the software ticks when user drives counter).
GetElapsedValue	Os.h	This service gets the number of ticks between the current tick value and a previously read tick value.
PduR_XcpTransmit	PduR_Xcp.h	Requests transmission of a PDU.

⌋

8.6.3 Configurable interfaces

In this chapter, all interfaces are listed where the target function could be configured. The target function is usually a call-back function. The names of these kind of interfaces is not fixed because they are configurable.

The XCP module offers configurable interfaces to be used by Complex Driver(s).

9 Sequence diagrams

9.1 XCP on FlexRay

9.1.1 Xcp on FlexRay Transmit

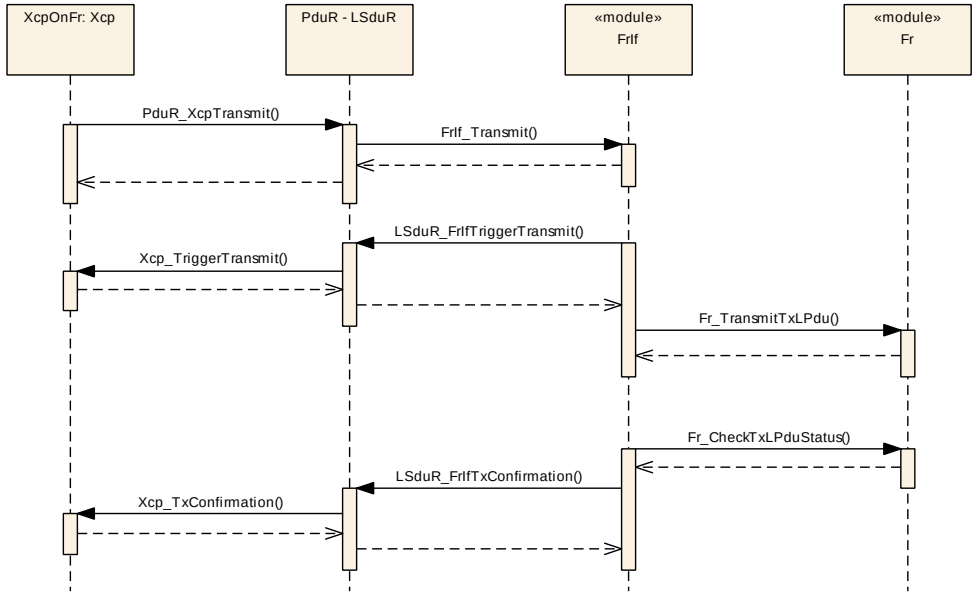

Figure 9.1: Xcp On FlexRay Transmit

9.1.2 Xcp on FlexRay Receive Indication

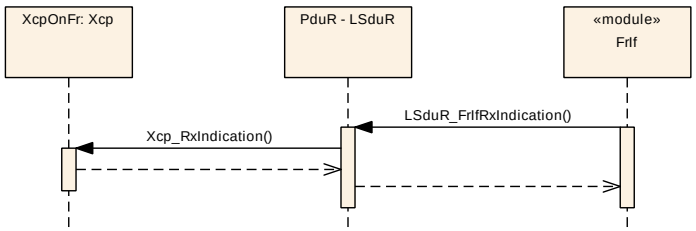

Figure 9.2: Xcp on FlexRay Receive Indication

9.2 XCP on CAN

9.2.1 Xcp on CAN Transmit

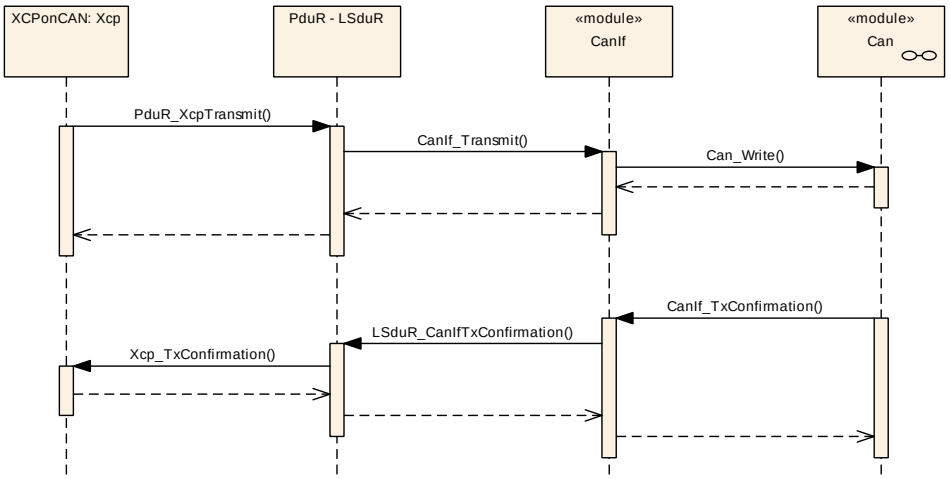

Figure 9.3: Xcp on Can Transmit

9.2.2 Xcp on CAN Receive Indication

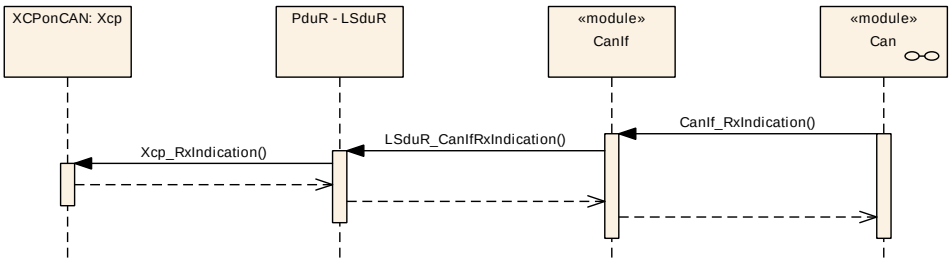

Figure 9.4: Xcp on CAN Receive Indication

9.3 XCP on Ethernet

9.3.1 Xcp on Ethernet Transmit

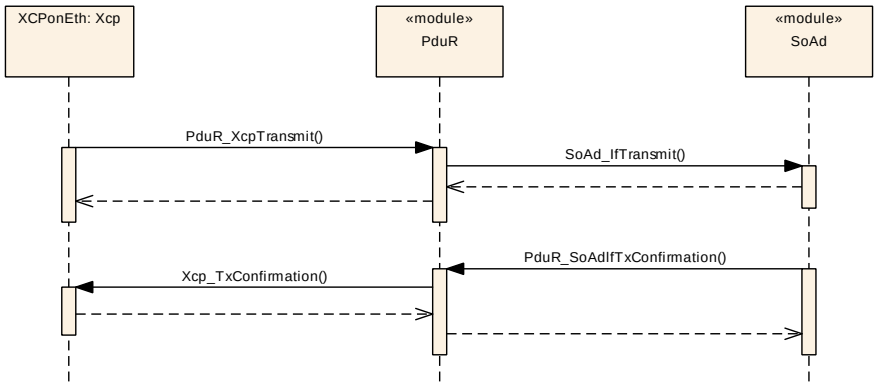

Figure 9.5: Xcp on Ethernet Transmit

9.3.2 Xcp on Ethernet Receive Indication

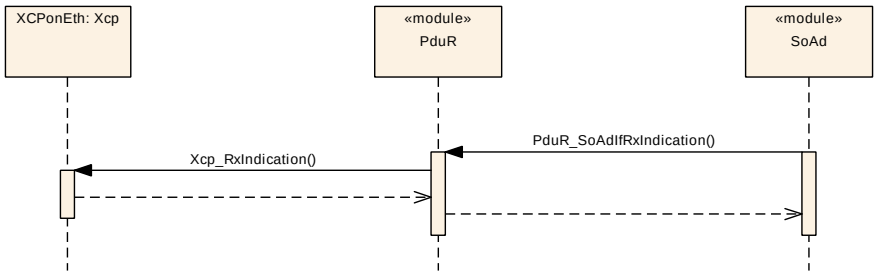

Figure 9.6: Xcp on Ethernet Receive Indication

10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification Chapter 10.1 describes fundamentals. It also specifies a template (table) you shall use for the parameter specification. We intend to leave Chapter 10.1 in the specification to guarantee comprehension.

Chapter 10.2 specifies the structure (containers) and the parameters of the module XCP.

Chapter 10.3 specifies published information of the module XCP.

10.1 How to read this chapter

For details refer to the chapter 10.1 “Introduction to configuration specification” in [2].

10.2 Containers and configuration parameters

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

[SWS_Xcp_00102]

Upstream requirements: SRS_BSW_00159

⌈The listed configuration items can be derived from a network description database, which is based on the EcuConfigurationTemplate. The configuration tool shall extract all information to configure the XCP.⌋

[SWS_Xcp_00103]

Upstream requirements: SRS_BSW_00167

⌈The configuration tool must check the consistency of the configuration at configuration time.⌋

[SWS_Xcp_00104]

Upstream requirements: SRS_BSW_00167

⌈Configuration rules and constraints for plausibility checks shall be performed during configuration time, wherever possible.⌋

[SWS_Xcp_00105]

Upstream requirements: SRS_BSW_00167

⌈These dependencies between FlexRay Interface and FlexRay Driver configuration must be provided at configuration time by the configuration tools.⌋

10.2.1 Xcp

[ECUC_Xcp_00182] Definition of EcucModuleDef Xcp ⌈

Module Name	Xcp
Description	Configuration of the XCP module
Post-Build Variant Support	true
Supported Config Variants	VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
XcpConfig	1	This container contains the configuration parameters and sub containers of the AUTOSAR Xcp module.
XcpGeneral	1	This container contains the general configuration parameters of the XCP.

⌋

10.2.2 XcpGeneral

[ECUC_Xcp_00001] Definition of EcucParamConfContainerDef XcpGeneral ⌈

Container Name	XcpGeneral
Parent Container	Xcp
Description	This container contains the general configuration parameters of the XCP.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpDaqConfigType	1	[ECUC_Xcp_00164]
XcpDaqCount	1	[ECUC_Xcp_00012]
XcpDevErrorDetect	1	[ECUC_Xcp_00003]
XcpFlashProgrammingEnabled	1	[ECUC_Xcp_00181]
XcpIdentificationFieldType	1	[ECUC_Xcp_00170]
XcpMainFunctionPeriod	1	[ECUC_Xcp_00014]

▽

△

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpMaxCto	1	[ECUC_Xcp_00004]
XcpMaxDto	1	[ECUC_Xcp_00005]
XcpMaxEventChannel	1	[ECUC_Xcp_00011]
XcpMinDaq	1	[ECUC_Xcp_00013]
XcpOdtCount	1	[ECUC_Xcp_00054]
XcpOdtEntriesCount	1	[ECUC_Xcp_00059]
XcpOdtEntrySizeDaq	1	[ECUC_Xcp_00177]
XcpOdtEntrySizeStim	1	[ECUC_Xcp_00178]
XcpOnCanEnabled	1	[ECUC_Xcp_00006]
XcpOnCddEnabled	1	[ECUC_Xcp_00009]
XcpOnEthernetEnabled	1	[ECUC_Xcp_00008]
XcpOnFlexRayEnabled	1	[ECUC_Xcp_00007]
XcpPrescalerSupported	1	[ECUC_Xcp_00169]
XcpSuppressTxSupport	1	[ECUC_Xcp_00176]
XcpTimestampTicks	1	[ECUC_Xcp_00167]
XcpTimestampType	1	[ECUC_Xcp_00166]
XcpTimestampUnit	1	[ECUC_Xcp_00168]
XcpVersionInfoApi	1	[ECUC_Xcp_00002]
XcpCounterRef	1	[ECUC_Xcp_00162]
XcpNvRamBlockIdRef	0..1	[ECUC_Xcp_00180]

No Included Containers

⌋

[ECUC_Xcp_00164] Definition of EcucEnumerationParamDef XcpDaqConfigType

⌈

Parameter Name	XcpDaqConfigType		
Parent Container	XcpGeneral		
Description	Sets the DAQ_CONFIG_TYPE bit within the DAQ_PROPERTIES parameter to "static" or to "dynamic". If DAQ_STATIC is selected, the DAQ_CONFIG_TYPE bit is set to "0". If DAQ_DYNAMIC is selected, the DAQ_CONFIG_TYPE bit is set to "1".		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DAQ_DYNAMIC	If DAQ_DYNAMIC is selected, the DAQ_CONFIG_TYPE bit is set to '1'	
	DAQ_STATIC	If DAQ_STATIC is selected, the DAQ_CONFIG_TYPE bit is set to '0'	
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

▽

△

Scope / Dependency	scope: ECU dependency: If DAQ_CONFIG_TYPE = dynamic, MAX_DAQ equals MIN_DAQ+DAQ_COUNT.

⌋

[ECUC_Xcp_00012] Definition of EcucIntegerParamDef XcpDaqCount ⌈

Parameter Name	XcpDaqCount		
Parent Container	XcpGeneral		
Description	Indicates the number of DAQ lists for dynamic configuration.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	–		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU dependency: This parameter is available only if XcpDaqConfigType is set to "1" i.e DAQ_DYNAMIC		

⌋

[ECUC_Xcp_00003] Definition of EcucBooleanParamDef XcpDevErrorDetect ⌈

Parameter Name	XcpDevErrorDetect		
Parent Container	XcpGeneral		
Description	Switches the development error detection and notification on or off. • true: detection and notification is enabled. • false: detection and notification is disabled.		
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		

⌋

[ECUC_Xcp_00181] Definition of EcucBooleanParamDef XcpFlashProgramming Enabled ⌈

Parameter Name	XcpFlashProgrammingEnabled		
Parent Container	XcpGeneral		
Description	Enabling of XCP Flash programming functionality		
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		

⌋

[ECUC_Xcp_00170] Definition of EcucEnumerationParamDef XcpIdentification FieldType ⌈

Parameter Name	XcpIdentificationFieldType		
Parent Container	XcpGeneral		
Description	Type of Identification Field the slave will use when transferring DAQ Packets to the master. The master has to use the same Type of Identification Field when transferring STIM Packets to the slave.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	ABSOLUTE	Absolute ODT number	
	RELATIVE_BYTE	Relative ODT number, absolute DAQ list number (BYTE)	
	RELATIVE_WORD	Relative ODT number, absolute DAQ list number (WORD)	
	RELATIVE_WORD_ALIGNED	Relative ODT number, absolute DAQ list number (WORD, aligned).	
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_Xcp_00014] Definition of EcucFloatParamDef XcpMainFunctionPeriod ⌈

Parameter Name	XcpMainFunctionPeriod
Parent Container	XcpGeneral
Description	The XCP does not require this information but the BSW scheduler, which invokes the main function, needs it in order to plan its tasks.

▽

△

Multiplicity	1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
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		

⌋

[ECUC_Xcp_00004] Definition of EcucIntegerParamDef XcpMaxCto ⌈

Parameter Name	XcpMaxCto		
Parent Container	XcpGeneral		
Description	MAX_CTO shows the maximum length of a CTO packet in bytes.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	8 .. 255		
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		

⌋

[ECUC_Xcp_00005] Definition of EcucIntegerParamDef XcpMaxDto ⌈

Parameter Name	XcpMaxDto		
Parent Container	XcpGeneral		
Description	MAX_DTO shows the maximum length of a DTO packet in bytes.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	8 .. 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		

⌋

[ECUC_Xcp_00011] Definition of EcucIntegerParamDef XcpMaxEventChannel ⌈

Parameter Name	XcpMaxEventChannel		
Parent Container	XcpGeneral		
Description	–		
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: ECU		

⌋

[ECUC_Xcp_00013] Definition of EcucIntegerParamDef XcpMinDaq ⌈

Parameter Name	XcpMinDaq		
Parent Container	XcpGeneral		
Description	Indicates the number of predefined, read only DAQ lists on the XCP slave.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
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		

⌋

[ECUC_Xcp_00054] Definition of EcucIntegerParamDef XcpOdtCount ⌈

Parameter Name	XcpOdtCount		
Parent Container	XcpGeneral		
Description	This parameter indicates the amount of ODTs of a DAQ list using dynamic DAQ list configuration.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 252		
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 dependency: This parameter is available only if XcpDaqConfigType is set to "1" i.e DAQ_DYNAMIC		

⌋

[ECUC_Xcp_00059] Definition of EcucIntegerParamDef XcpOdtEntriesCount ⌈

Parameter Name	XcpOdtEntriesCount		
Parent Container	XcpGeneral		
Description	Indicates the amount of entries into an ODT using dynamic DAQ list configuration.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
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 dependency: This parameter is available only if XcpDaqConfigType is set to "1" i.e DAQ_DYNAMIC		

⌋

[ECUC_Xcp_00177] Definition of EcucIntegerParamDef XcpOdtEntrySizeDaq ⌈

Parameter Name	XcpOdtEntrySizeDaq		
Parent Container	XcpGeneral		
Description	Indicates the size of an element described by an ODT entry to the DaqListType for a DAQ.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
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		

⌋

[ECUC_Xcp_00178] Definition of EcucIntegerParamDef XcpOdtEntrySizeStim ⌈

Parameter Name	XcpOdtEntrySizeStim		
Parent Container	XcpGeneral		
Description	Indicates the size of an element described by an ODT entry to the DaqListType for a stim.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
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		

⌋

[ECUC_Xcp_00006] Definition of EcucBooleanParamDef XcpOnCanEnabled ⌈

Parameter Name	XcpOnCanEnabled		
Parent Container	XcpGeneral		
Description	Enabling of XCPonCAN functionality		
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		

⌋

[ECUC_Xcp_00009] Definition of EcucBooleanParamDef XcpOnCddEnabled ⌈

Parameter Name	XcpOnCddEnabled		
Parent Container	XcpGeneral		
Description	Enabling of XCPonCdd functionality		
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		

⌋

[ECUC_Xcp_00008] Definition of EcucBooleanParamDef XcpOnEthernetEnabled

⌈

Parameter Name	XcpOnEthernetEnabled		
Parent Container	XcpGeneral		
Description	Enabling of XCPonEthernet functionality		
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		

⌋

[ECUC_Xcp_00007] Definition of EcucBooleanParamDef XcpOnFlexRayEnabled

⌈

Parameter Name	XcpOnFlexRayEnabled		
Parent Container	XcpGeneral		
Description	Enabling of XCPonFlexRay functionality		
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		

⌋

[ECUC_Xcp_00169] Definition of EcucBooleanParamDef XcpPrescalerSupported

⌈

Parameter Name	XcpPrescalerSupported		
Parent Container	XcpGeneral		
Description	This parameter enables and disables the support for Prescaler support. True is Enabled, False is disabled		
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		

⌋

[ECUC_Xcp_00176] Definition of EcucBooleanParamDef XcpSuppressTxSupport ⌈

Parameter Name	XcpSuppressTxSupport		
Parent Container	XcpGeneral		
Description	Switches the support of suppressing transmission of PDUs per communication channel on or off. TRUE: Suppressing of TxPDUs supported FALSE: Suppressing of TxPDUs not supported		
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		

⌋

[ECUC_Xcp_00167] Definition of EcucIntegerParamDef XcpTimestampTicks ⌈

Parameter Name	XcpTimestampTicks		
Parent Container	XcpGeneral		
Description	This parameter defines the timestamp that will increment based TIMESTAMP_TICKS per unit and wrap around if an overflow occurs.		
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		

⌋

[ECUC_Xcp_00166] Definition of EcucEnumerationParamDef XcpTimestampType ⌈

Parameter Name	XcpTimestampType		
Parent Container	XcpGeneral		
Description	This parameter indicates the number of bytes used for the timestamp field. In case No_TIME_STAMP is selected the timestamp field is not available.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	FOUR_BYTE	timestamp field has the size of four byte.	
	NO_TIME_STAMP	timestamp field is not available.	
	ONE_BYTE	timestamp field has the size of one byte.	
	TWO_BYTE	timestamp field has the size of two byte.	
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_Xcp_00168] Definition of EcucEnumerationParamDef XcpTimestampUnit ⌈

Parameter Name	XcpTimestampUnit		
Parent Container	XcpGeneral		
Description	This parameter indicates the resolution of the data acquisition clock of the slave when transferring data to master.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	TIMESTAMP_UNIT_100MS	Unit is 100 millisecond.	
	TIMESTAMP_UNIT_100NS	Unit is 100 nanosecond.	
	TIMESTAMP_UNIT_100PS	Unit is 100 picosecond.	
	TIMESTAMP_UNIT_100US	Unit is 100 microsecond.	
	TIMESTAMP_UNIT_10MS	Unit is 10 millisecond.	
	TIMESTAMP_UNIT_10NS	Unit is 10 nanosecond.	
	TIMESTAMP_UNIT_10PS	Unit is 10 picosecond.	
	TIMESTAMP_UNIT_10US	Unit is 10 microsecond.	
	TIMESTAMP_UNIT_1MS	Unit is 1 millisecond.	
	TIMESTAMP_UNIT_1NS	Unit is 1 nonasecond.	
	TIMESTAMP_UNIT_1PS	Unit is 1 picosecond.	
	TIMESTAMP_UNIT_1S	Unit is 1 second.	
	TIMESTAMP_UNIT_1US	Unit is 1 microsecond.	
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

▽

△

Scope / Dependency	scope: local

⌋

[ECUC_Xcp_00002] Definition of EcucBooleanParamDef XcpVersionInfoApi ⌈

Parameter Name	XcpVersionInfoApi		
Parent Container	XcpGeneral		
Description	Enables/disables the existence of the XCP_GetVersionInfo() API service. TRUE: XCP_GetVersionInfo() API service exists FALSE: XCP_GetVersionInfo() API service does not exist		
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		

⌋

[ECUC_Xcp_00162] Definition of EcucReferenceDef XcpCounterRef ⌈

Parameter Name	XcpCounterRef		
Parent Container	XcpGeneral		
Description	This parameter contains a reference to the counter, which is used by XCP.		
Multiplicity	1		
Type	Reference to OsCounter		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_Xcp_00180] Definition of EcucReferenceDef XcpNvRamBlockIdRef ⌈

Parameter Name	XcpNvRamBlockIdRef
Parent Container	XcpGeneral
Description	This reference contains the link to a non-volatile memory block to be used in the feature "RESUME MODE" so this information has to be stored non volatile to be available directly after start-up of the ECU.
Multiplicity	0..1
Type	Symbolic name reference to NvMBlockDescriptor

▽

△

Post-Build Variant Multiplicity	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: local		

⌋

10.2.3 XcpConfig

[ECUC_Xcp_00020] Definition of EcucParamConfContainerDef XcpConfig ⌈

Container Name	XcpConfig
Parent Container	Xcp
Description	This container contains the configuration parameters and sub containers of the AUTOSAR Xcp module.
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
XcpCommunicationChannel	0..*	This container represents the configuration of the communication channel of XCP.
XcpDaqList	1..*	This container contains the configuration of the DAQs.
XcpEventChannel	1..*	This container contains the configuration of event channels on the XCP slave.
XcpPageSwitching	0..1	This container represents configuration of the page switching feature.
XcpPdu	1..*	Contains PDU information. A PDU may be either a transmission PDU or a reception PDU.

⌋

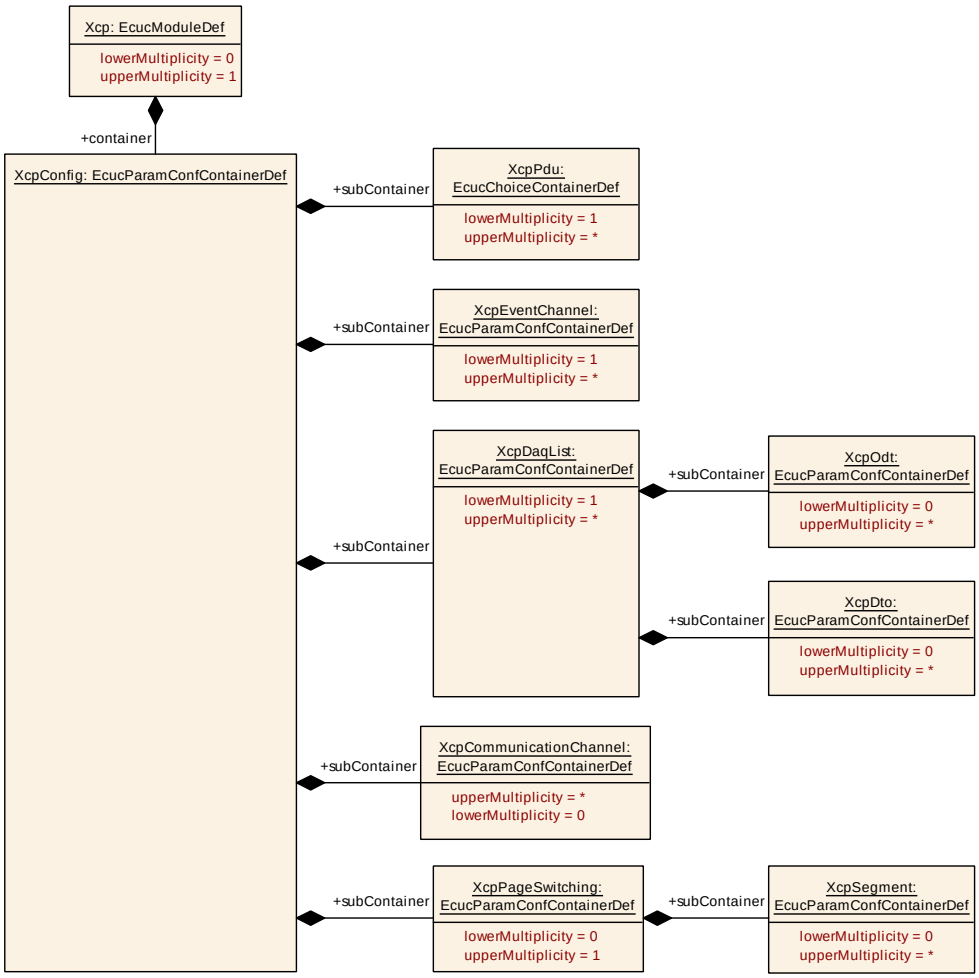

Figure 10.1: Diagram XcpConfig

10.2.4 XcpDaqList

[ECUC_Xcp_00050] Definition of EcucParamConfContainerDef XcpDaqList ⌈

Container Name	XcpDaqList
Parent Container	XcpConfig
Description	This container contains the configuration of the DAQs.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpDaqListNumber	1	[ECUC_Xcp_00051]
XcpDaqListType	1	[ECUC_Xcp_00052]
XcpMaxOdt	1	[ECUC_Xcp_00053]
XcpMaxOdtEntries	1	[ECUC_Xcp_00058]

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
XcpDto	0..*	This container collects data transfer object specific parameters for the DAQ list.
XcpOdt	0..*	This container contains ODT-specific parameter for the DAQ list.

⌋

[ECUC_Xcp_00051] Definition of EcucIntegerParamDef XcpDaqListNumber ⌈

Parameter Name	XcpDaqListNumber		
Parent Container	XcpDaqList		
Description	Index number of the DAQ list		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65534		
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		

⌋

[ECUC_Xcp_00052] Definition of EcucEnumerationParamDef XcpDaqListType ⌈

Parameter Name	XcpDaqListType		
Parent Container	XcpDaqList		
Description	This indicates whether this DAQ list represents a DAQ or a STIM.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DAQ	This DAQ list is a DAQ.	
	DAQ_STIM	This DAQ list can be DAQ or STIM.	
	STIM	This DAQ list is a STIM.	
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

⌋

[ECUC_Xcp_00053] Definition of EcucIntegerParamDef XcpMaxOdt ⌈

Parameter Name	XcpMaxOdt		
Parent Container	XcpDaqList		
Description	MAX_ODT indicates the maximum amount of ODTs in this DAQ list (STATIC configuration)		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 252		
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 dependency: only available if XcpDaqConfigType is "DAQ_STATIC" (bit set to '0')		

⌋

[ECUC_Xcp_00058] Definition of EcucIntegerParamDef XcpMaxOdtEntries ⌈

Parameter Name	XcpMaxOdtEntries		
Parent Container	XcpDaqList		
Description	This parameter indicates the maximum amount of entries in an ODT of this DAQ list (STATIC configuration).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
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 dependency: only available if XcpDaqConfigType is "DAQ_STATIC" (bit set to '0')		

⌋

10.2.5 XcpDto

[ECUC_Xcp_00065] Definition of EcucParamConfContainerDef XcpDto ⌈

Container Name	XcpDto
Parent Container	XcpDaqList
Description	This container collects data transfer object specific parameters for the DAQ list.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpDtoPid	1	[ECUC_Xcp_00066]
XcpDto2PduMapping	1	[ECUC_Xcp_00067]

No Included Containers

⌋

[ECUC_Xcp_00066] Definition of EcucIntegerParamDef XcpDtoPid ⌈

Parameter Name	XcpDtoPid		
Parent Container	XcpDto		
Description	Packet identifier (PID) of the DTO that identifies the ODT the content of the DTO.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 251		
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		

⌋

[ECUC_Xcp_00067] Definition of EcucChoiceReferenceDef XcpDto2PduMapping ⌈

Parameter Name	XcpDto2PduMapping		
Parent Container	XcpDto		
Description	This reference specifies the mapping of the DTO to the PDUs from the lower-layer interfaces (CanIf, FrIf, SoAd and Cdd). A reference to a XcpRxPdu is only feasible if the the DaqListType is DAQ_STIM. A reference to a XcpTxPdu is only feasible if the DaqListType is DAQ.		
Multiplicity	1		
Type	Choice reference to [XcpRxPdu, XcpTxPdu]		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

⌋

10.2.6 XcpOdt

[ECUC_Xcp_00055] Definition of EcucParamConfContainerDef XcpOdt ⌈

Container Name	XcpOdt
Parent Container	XcpDaqList
Description	This container contains ODT-specific parameter for the DAQ list.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpOdtEntryMaxSize	1	[ECUC_Xcp_00060]
XcpOdtNumber	0..1	[ECUC_Xcp_00057]
XcpOdt2DtoMapping	0..1	[ECUC_Xcp_00056]

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
XcpOdtEntry	1..*	This container collects all configuration parameters that comprise an ODT entry.

⌋

[ECUC_Xcp_00060] Definition of EcucIntegerParamDef XcpOdtEntryMaxSize ⌈

Parameter Name	XcpOdtEntryMaxSize		
Parent Container	XcpOdt		
Description	This parameter indicates the upper limit for the size of the element described by an ODT entry. Depending on the DaqListType this ODT belongs to it describes the limit for a DAQ (MAX_ODT_ENTRY_SIZE_DAQ) or a STIM (MAX_ODT_ENTRY_SIZE_STIM).		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 254		
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		

⌋

[ECUC_Xcp_00057] Definition of EcucIntegerParamDef XcpOdtNumber ⌈

Parameter Name	XcpOdtNumber
Parent Container	XcpOdt
Description	Index number of this ODT within the DAQ list.

▽

△

Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 251		
Default value	–		
Post-Build Variant Multiplicity	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		

⌋

[ECUC_Xcp_00056] Definition of EcucReferenceDef XcpOdt2DtoMapping ⌈

Parameter Name	XcpOdt2DtoMapping		
Parent Container	XcpOdt		
Description	This reference maps the ODT to the according DTO in which it will be transmitted.		
Multiplicity	0..1		
Type	Reference to XcpDto		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

⌋

10.2.7 XcpOdtEntry

[ECUC_Xcp_00061] Definition of EcucParamConfContainerDef XcpOdtEntry ⌈

Container Name	XcpOdtEntry
Parent Container	XcpOdt
Description	This container collects all configuration parameters that comprise an ODT entry.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpOdtEntryAddress	0..1	[ECUC_Xcp_00063]
XcpOdtEntryBitOffset	0..1	[ECUC_Xcp_00179]
XcpOdtEntryLength	0..1	[ECUC_Xcp_00064]
XcpOdtEntryNumber	0..1	[ECUC_Xcp_00062]

No Included Containers

⌋

[ECUC_Xcp_00063] Definition of EcucLinkerSymbolDef XcpOdtEntryAddress ⌈

Parameter Name	XcpOdtEntryAddress		
Parent Container	XcpOdtEntry		
Description	Memory address that the ODT entry is referencing to.		
Multiplicity	0..1		
Type	EcucLinkerSymbolDef		
Default value	–		
maxLength	–		
minLength	–		
Regular Expression	–		
Post-Build Variant Multiplicity	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		

⌋

[ECUC_Xcp_00179] Definition of EcucIntegerParamDef XcpOdtEntryBitOffset ⌈

Parameter Name	XcpOdtEntryBitOffset		
Parent Container	XcpOdtEntry		
Description	Represent the bit offset in case of the element represents status bit.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 31		
Default value	–		
Post-Build Variant Multiplicity	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		

⌋

[ECUC_Xcp_00064] Definition of EcucIntegerParamDef XcpOdtEntryLength ⌈

Parameter Name	XcpOdtEntryLength		
Parent Container	XcpOdtEntry		
Description	Length of the referenced memory area that is referenced by the ODT entry.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	–		
Post-Build Variant Multiplicity	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		

⌋

[ECUC_Xcp_00062] Definition of EcucIntegerParamDef XcpOdtEntryNumber ⌈

Parameter Name	XcpOdtEntryNumber		
Parent Container	XcpOdtEntry		
Description	Index number of the ODT entry		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 254		
Default value	–		
Post-Build Variant Multiplicity	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		

⌋

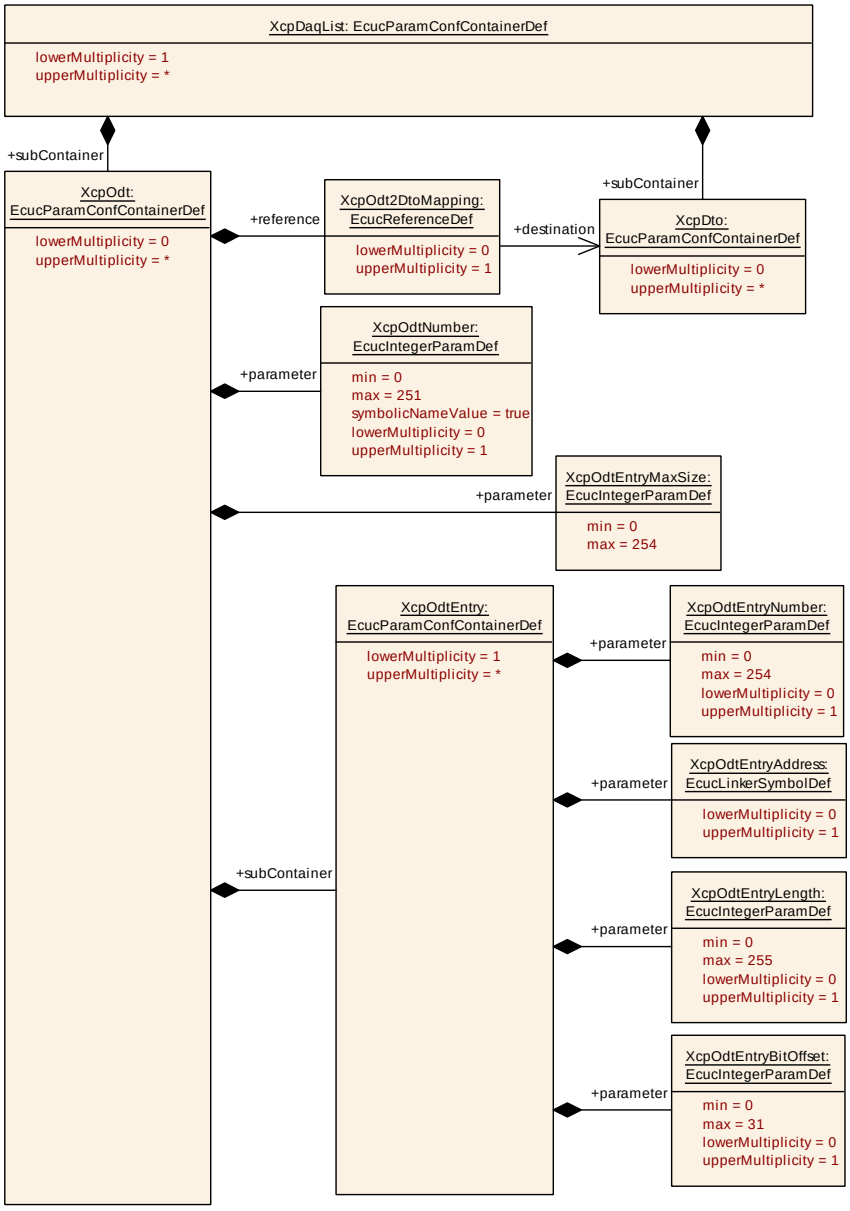

Figure 10.2: Diagram XcpOdtEntry

10.2.8 XcpEventChannel

[ECUC_Xcp_00150] Definition of EcucParamConfContainerDef XcpEventChannel ⌈

Container Name	XcpEventChannel
Parent Container	XcpConfig
Description	This container contains the configuration of event channels on the XCP slave.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpEventChannelConsistency	1	[ECUC_Xcp_00171]
XcpEventChannelMaxDaqList	1	[ECUC_Xcp_00153]
XcpEventChannelNumber	1	[ECUC_Xcp_00152]
XcpEventChannelPriority	1	[ECUC_Xcp_00154]
XcpEventChannelTimeCycle	1	[ECUC_Xcp_00173]
XcpEventChannelTimeUnit	0..1	[ECUC_Xcp_00174]
XcpEventChannelType	1	[ECUC_Xcp_00172]
XcpEventChannelTriggeredDaqListRef	0..*	[ECUC_Xcp_00151]

No Included Containers

⌋

[ECUC_Xcp_00171] Definition of EcucEnumerationParamDef XcpEventChannelConsistency ⌈

Parameter Name	XcpEventChannelConsistency		
Parent Container	XcpEventChannel		
Description	Type of consistency used by event channel		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DAQ	Consistency on DAQ list level	
	EVENT	Consistency on Event Channel Level	
	ODT	Consistency on ODT level (default value).	
Default value	ODT		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_Xcp_00153] Definition of EcucIntegerParamDef XcpEventChannelMaxDaqList ⌈

Parameter Name	XcpEventChannelMaxDaqList		
Parent Container	XcpEventChannel		
Description	Maximum amount of DAQ lists that are handled by this event channel.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
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		

⌋

[ECUC_Xcp_00152] Definition of EcucIntegerParamDef XcpEventChannelNumber ⌈

Parameter Name	XcpEventChannelNumber		
Parent Container	XcpEventChannel		
Description	Index number of the event channel.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65534		
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		

⌋

[ECUC_Xcp_00154] Definition of EcucIntegerParamDef XcpEventChannelPriority ⌈

Parameter Name	XcpEventChannelPriority	
Parent Container	XcpEventChannel	
Description	Priority of the event channel	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 255	
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		

⌋

[ECUC_Xcp_00173] Definition of EcucIntegerParamDef XcpEventChannelTimeCycle ⌈

Parameter Name	XcpEventChannelTimeCycle		
Parent Container	XcpEventChannel		
Description	The event channel time cycle indicates which sampling period is used to process this event channel. A value of 0 means 'Not cyclic'.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
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		

⌋

[ECUC_Xcp_00174] Definition of EcucEnumerationParamDef XcpEventChannelTimeUnit ⌈

Parameter Name	XcpEventChannelTimeUnit	
Parent Container	XcpEventChannel	
Description	This configuration parameter indicates the unit of the event channel time cycle.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	TIMESTAMP_UNIT_100MS	Unit is 100 millisecond.
	TIMESTAMP_UNIT_100NS	Unit is 100 nanosecond.
	TIMESTAMP_UNIT_100PS	Unit is 100 picosecond.
	TIMESTAMP_UNIT_100US	Unit is 100 microsecond.
	TIMESTAMP_UNIT_10MS	Unit is 10 millisecond.
	TIMESTAMP_UNIT_10NS	Unit is 10 nanosecond.
	TIMESTAMP_UNIT_10PS	Unit is 10 picosecond.
	TIMESTAMP_UNIT_10US	Unit is 10 microsecond.
	TIMESTAMP_UNIT_1MS	Unit is 1 millisecond.
	TIMESTAMP_UNIT_1NS	Unit is 1 nonasecond.

▽

△

	TIMESTAMP_UNIT_1PS	Unit is 1 picosecond.	
	TIMESTAMP_UNIT_1S	Unit is 1 second.	
	TIMESTAMP_UNIT_1US	Unit is 1 microsecond.	
Post-Build Variant Multiplicity	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 dependency: Dependent on the Parameter EventChannelTimeCycle. When this parameter is set to 0, the entire event channel time unit parameter shall be ignored.		

⌋

[ECUC_Xcp_00172] Definition of EcucEnumerationParamDef XcpEventChannelType ⌈

Parameter Name	XcpEventChannelType		
Parent Container	XcpEventChannel		
Description	This configuration parameter indicates what kind of DAQ list can be allocated to this event channel.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DAQ	only DAQ supported (default value).	
	DAQ_STIM	Both DAQ and STIM supported (Simultaneously).	
	STIM	only STIM supported	
Default value	DAQ		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_Xcp_00151] Definition of EcucReferenceDef XcpEventChannelTriggeredDaqListRef ⌈

Parameter Name	XcpEventChannelTriggeredDaqListRef
Parent Container	XcpEventChannel
Description	References all DAQ lists that are trigged by this event channel.
Multiplicity	0..*

▽

△

Type	Reference to XcpDaqList		
Post-Build Variant Multiplicity	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		

⌋

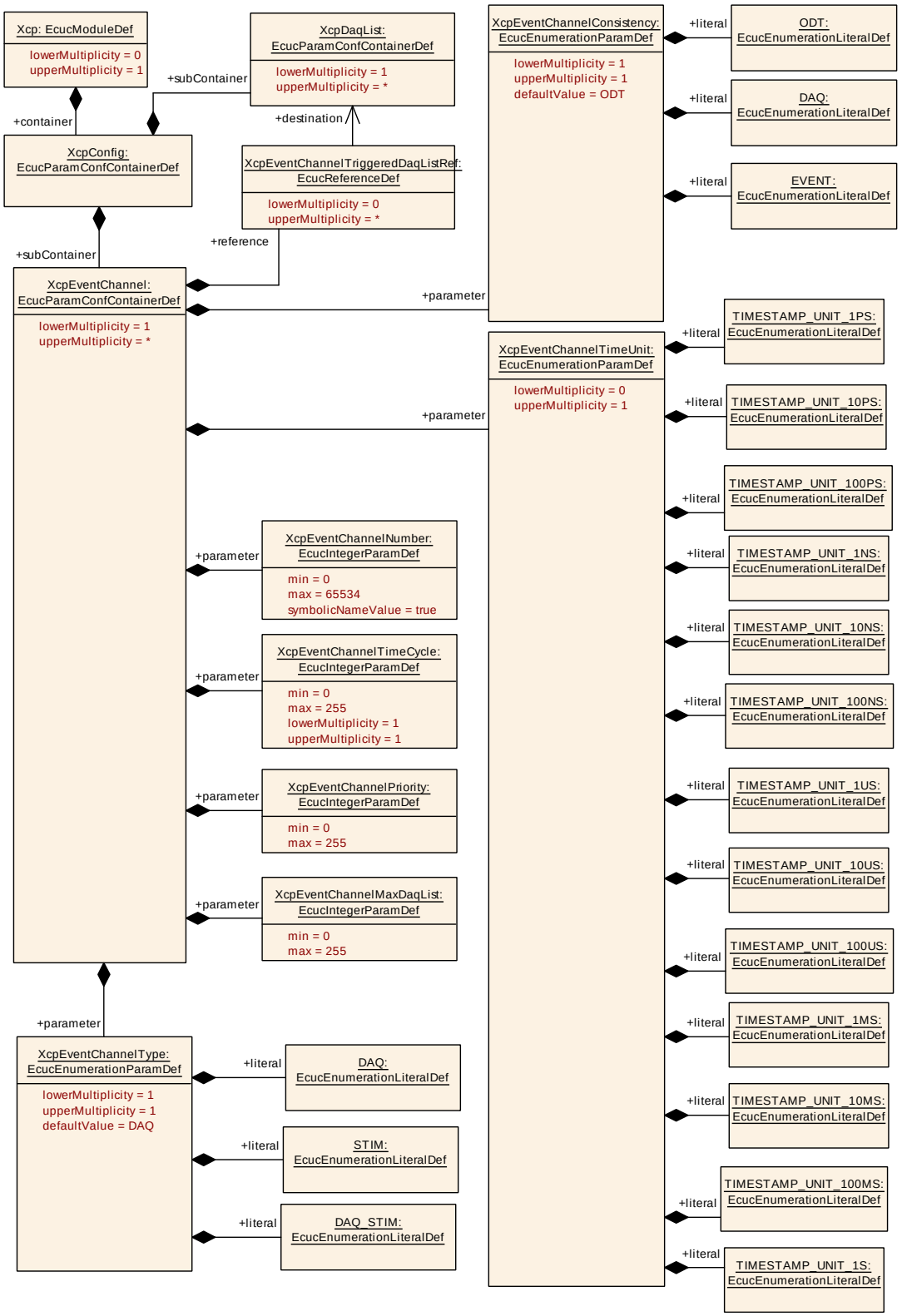

Figure 10.3: Diagram XcpChannel

10.2.9 XcpPdu

[ECUC_Xcp_00100] Definition of EcucChoiceContainerDef XcpPdu ⌈

Choice Container Name	XcpPdu
Parent Container	XcpConfig
Description	Contains PDU information. A PDU may be either a transmission PDU or a reception PDU.

No Included Parameters

Container Choices		
Container Name	**Multiplicity**	**Scope / Dependency**
XcpRxPdu	0..1	This container specifies received PDUs.
XcpTxPdu	0..1	This container specifies transmission PDUs.

⌋

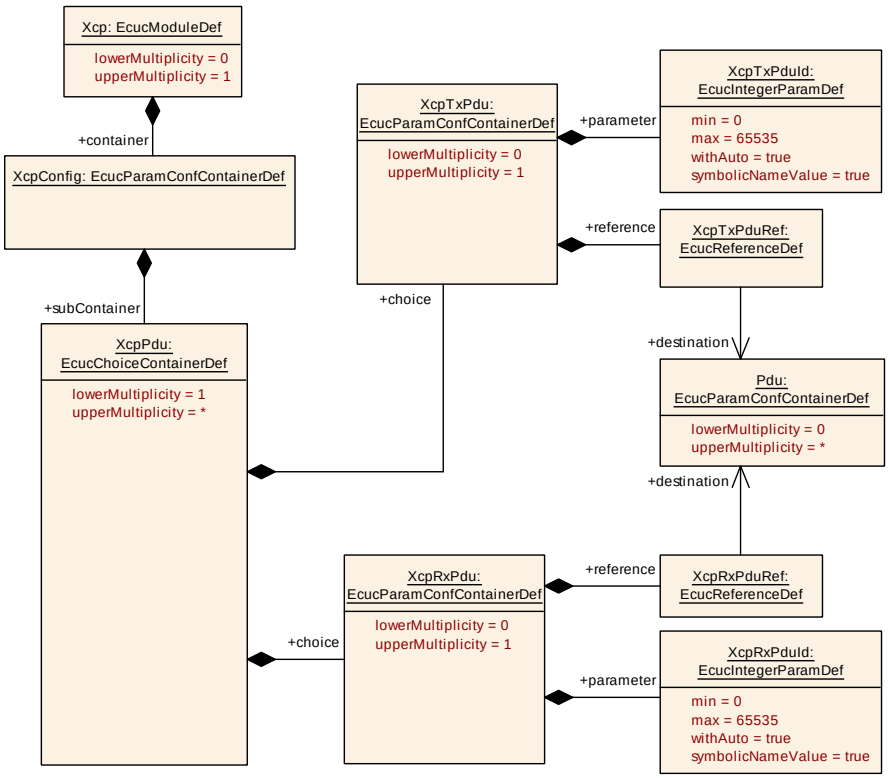

Figure 10.4: Diagram XcpPdu

10.2.10 XcpRxPdu

[ECUC_Xcp_00105] Definition of EcucParamConfContainerDef XcpRxPdu ⌈

Container Name	XcpRxPdu
Parent Container	XcpPdu
Description	This container specifies received PDUs.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpRxPduId	1	[ECUC_Xcp_00106]
XcpRxPduRef	1	[ECUC_Xcp_00107]

No Included Containers

⌋

[ECUC_Xcp_00106] Definition of EcucIntegerParamDef XcpRxPduId ⌈

Parameter Name	XcpRxPduId		
Parent Container	XcpRxPdu		
Description	ID of the PDU that will be received via a Xcp_<module>RxIndication.		
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 withAuto = true		

⌋

[ECUC_Xcp_00107] Definition of EcucReferenceDef XcpRxPduRef ⌈

Parameter Name	XcpRxPduRef		
Parent Container	XcpRxPdu		
Description	–		
Multiplicity	1		
Type	Reference to Pdu		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

⌋

10.2.11 XcpTxPdu

[ECUC_Xcp_00101] Definition of EcucParamConfContainerDef XcpTxPdu ⌈

Container Name	XcpTxPdu
Parent Container	XcpPdu
Description	This container specifies transmission PDUs.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpTxPduId	1	[ECUC_Xcp_00103]
XcpTxPduRef	1	[ECUC_Xcp_00104]

No Included Containers

⌋

[ECUC_Xcp_00103] Definition of EcucIntegerParamDef XcpTxPduId ⌈

Parameter Name	XcpTxPduId		
Parent Container	XcpTxPdu		
Description	The PDU identifier, which has to be used by the lower layer BSW module for Tx Confirmations or TriggerTransmits.		
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 withAuto = true		

⌋

[ECUC_Xcp_00104] Definition of EcucReferenceDef XcpTxPduRef ⌈

Parameter Name	XcpTxPduRef		
Parent Container	XcpTxPdu		
Description	Reference to the external PDU definition.		
Multiplicity	1		
Type	Reference to Pdu		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE

▽

△

	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

⌋

10.2.12 XcpCommunicationChannel

[ECUC_Xcp_00183] Definition of EcucParamConfContainerDef XcpCommunicationChannel ⌈

Container Name	XcpCommunicationChannel
Parent Container	XcpConfig
Description	This container represents the configuration of the communication channel of XCP.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpChannelRxPduRef	0..1	[ECUC_Xcp_00185]
XcpChannelTxPduRef	1	[ECUC_Xcp_00184]
XcpComMChannelRef	1	[ECUC_Xcp_00186]

No Included Containers

⌋

[ECUC_Xcp_00185] Definition of EcucReferenceDef XcpChannelRxPduRef ⌈

Parameter Name	XcpChannelRxPduRef		
Parent Container	XcpCommunicationChannel		
Description	Optional reference to the XCP Rx PDU.		
Multiplicity	0..1		
Type	Reference to XcpRxPdu		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

⌋

[ECUC_Xcp_00184] Definition of EcucReferenceDef XcpChannelTxPduRef ⌈

Parameter Name	XcpChannelTxPduRef		
Parent Container	XcpCommunicationChannel		
Description	Reference to the XCP Tx PDU.		
Multiplicity	1		
Type	Reference to XcpTxPdu		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

⌋

[ECUC_Xcp_00186] Definition of EcucReferenceDef XcpComMChannelRef ⌈

Parameter Name	XcpComMChannelRef		
Parent Container	XcpCommunicationChannel		
Description	Reference to the ComM channel the PDUs belong to.		
Multiplicity	1		
Type	Reference to ComMChannel		
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

⌋

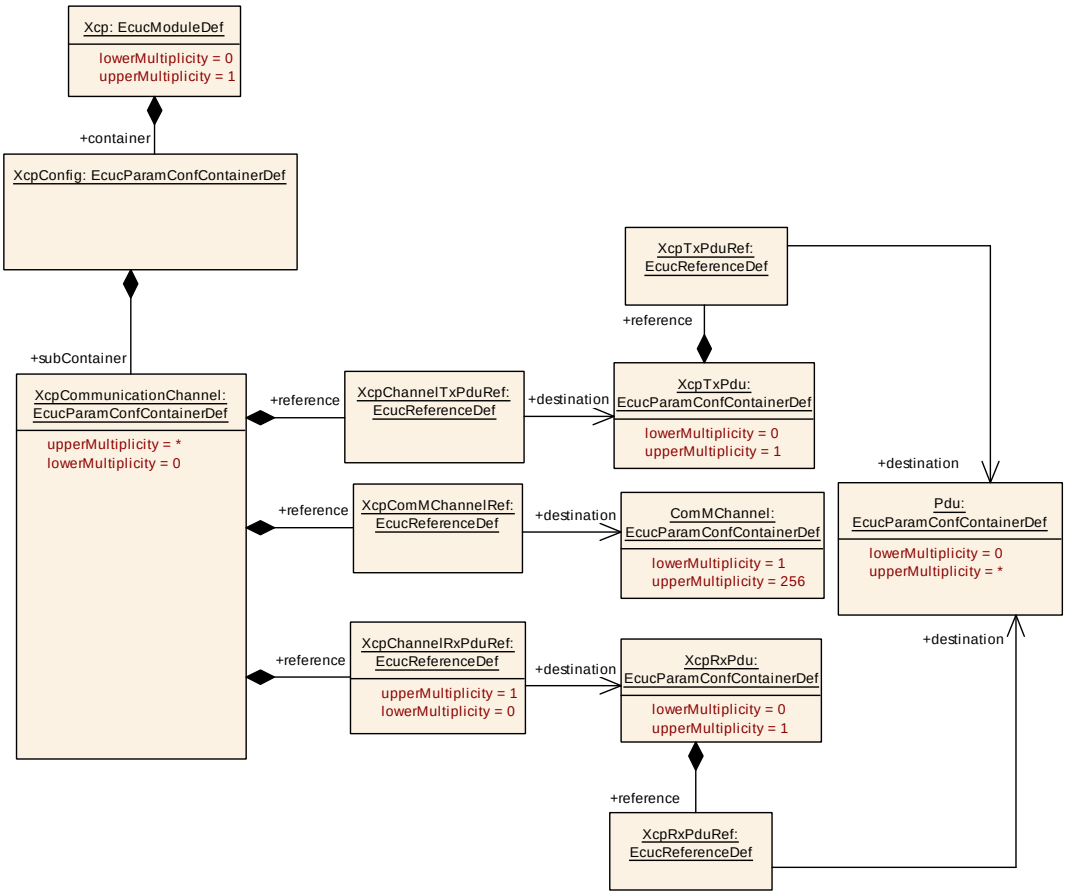

Figure 10.5: Diagram XcpCommunicationChannel

10.2.13 XcpPageSwitching

[ECUC_Xcp_00187] Definition of EcucParamConfContainerDef XcpPageSwitching ⌈

Container Name	XcpPageSwitching
Parent Container	XcpConfig
Description	This container represents configuration of the page switching feature.
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
XcpSegment	0..*	This container represents configuration of the page switching segment element.

⌋

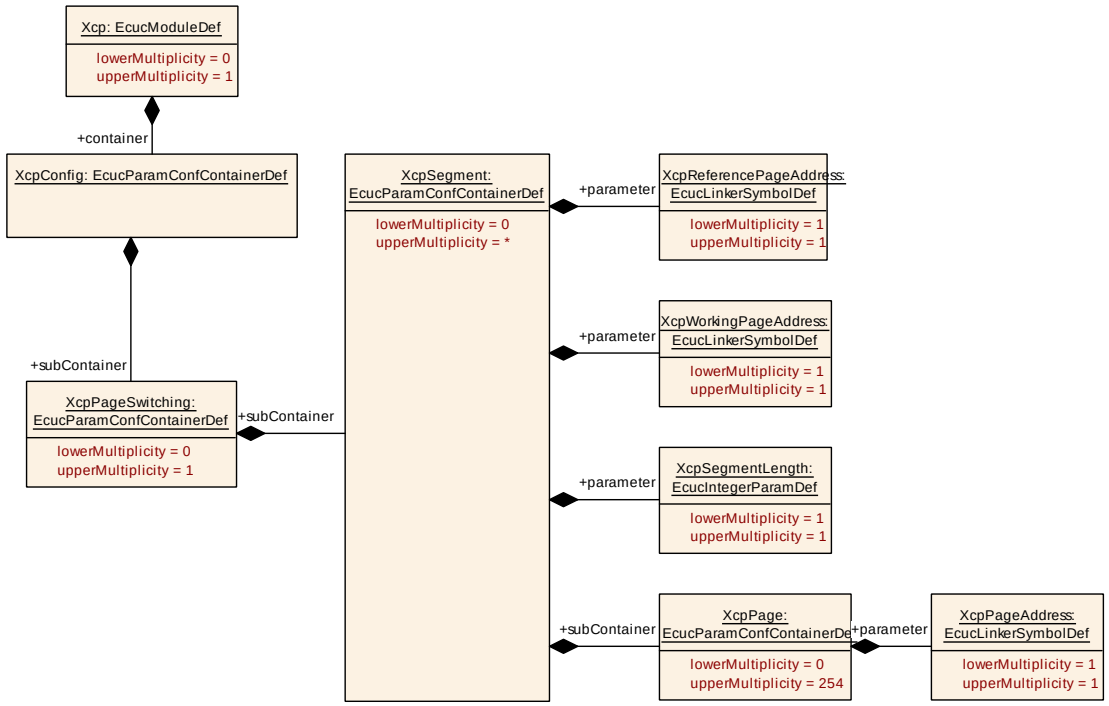

Figure 10.6: Diagram XcpPageSwitching

10.2.14 XcpSegment

[ECUC_Xcp_00188] Definition of EcucParamConfContainerDef XcpSegment ⌈

Container Name	XcpSegment
Parent Container	XcpPageSwitching
Description	This container represents configuration of the page switching segment element.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpReferencePageAddress	1	[ECUC_Xcp_00189]
XcpSegmentLength	1	[ECUC_Xcp_00191]
XcpWorkingPageAddress	1	[ECUC_Xcp_00190]

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
XcpPage	0..254	This container represents configuration of the optional page element.

⌋

[ECUC_Xcp_00189] Definition of EcucLinkerSymbolDef XcpReferencePageAddress ⌈

Parameter Name	XcpReferencePageAddress		
Parent Container	XcpSegment		
Description	Memory address of the reference page (Page ID = 0).		
Multiplicity	1		
Type	EcucLinkerSymbolDef		
Default value	–		
maxLength	–		
minLength	–		
Regular Expression	–		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_Xcp_00191] Definition of EcucIntegerParamDef XcpSegmentLength ⌈

Parameter Name	XcpSegmentLength		
Parent Container	XcpSegment		
Description	Length of the segment in bytes.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 18446744073709551615		
Default value	–		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_Xcp_00190] Definition of EcucLinkerSymbolDef XcpWorkingPageAddress ⌈

Parameter Name	XcpWorkingPageAddress
Parent Container	XcpSegment
Description	Memory address address of the working page (Page ID = 1).
Multiplicity	1
Type	EcucLinkerSymbolDef
Default value	–
maxLength	–
minLength	–

▽

△

Regular Expression	–		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

10.2.15 XcpPage

[ECUC_Xcp_00192] Definition of EcucParamConfContainerDef XcpPage ⌈

Container Name	XcpPage
Parent Container	XcpSegment
Description	This container represents configuration of the optional page element.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
XcpPageAddress	1	[ECUC_Xcp_00193]

No Included Containers

⌋

[ECUC_Xcp_00193] Definition of EcucLinkerSymbolDef XcpPageAddress ⌈

Parameter Name	XcpPageAddress		
Parent Container	XcpPage		
Description	Memory address of the optional page (Page ID = 2 ... 255).		
Multiplicity	1		
Type	EcucLinkerSymbolDef		
Default value	–		
maxLength	–		
minLength	–		
Regular Expression	–		
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

10.3 Published Information

For details refer to the chapter 10.3 “Published Information” in [2].

A Not applicable requirements

[SWS_Xcp_NA_00999]

Upstream requirements: SRS_BSW_00171, SRS_BSW_00170, SRS_BSW_00375, SRS_BSW_-00416, SRS_BSW_00168, SRS_BSW_00423, SRS_BSW_00425, SRS_BSW_00426, SRS_BSW_00427, SRS_BSW_00428, SRS_BSW_-00432, SRS_BSW_00336, SRS_BSW_00417, SRS_BSW_00161, SRS_BSW_00162, SRS_BSW_00005, SRS_BSW_00415, SRS_BSW_-00164, SRS_BSW_00325, SRS_BSW_00413, SRS_BSW_00347, SRS_BSW_00335, SRS_BSW_00410, SRS_BSW_00314, SRS_BSW_-00328, SRS_BSW_00312, SRS_BSW_00006, SRS_BSW_00377, SRS_BSW_00306, SRS_BSW_00309, SRS_BSW_00360, SRS_BSW_-00330, SRS_BSW_00331, SRS_BSW_00009, SRS_BSW_00401, SRS_BSW_00172, SRS_BSW_00010, SRS_BSW_00333, SRS_BSW_-00321, SRS_BSW_00341, SRS_Xcp_29008

⌈These requirements are not applicable to this specification.⌋

B Change history of AUTOSAR traceable items

Please note that the lists in this chapter also include traceable items that have been removed from the specification in a later version. These items do not appear as hyperlinks in the document.

B.1 Traceable item history of this document according to AUTOSAR Release R22-11

B.1.1 Added Specification Items in R22-11

Number	Heading
[SWS_Xcp_00102]	
[SWS_Xcp_00103]	
[SWS_Xcp_00104]	
[SWS_Xcp_00105]	
[SWS_Xcp_00501]	
[SWS_Xcp_00701]	
[SWS_Xcp_00702]	
[SWS_Xcp_00703]	
[SWS_Xcp_00705]	
[SWS_Xcp_00706]	
[SWS_Xcp_00707]	
[SWS_Xcp_00708]	
[SWS_Xcp_00709]	
[SWS_Xcp_00710]	
[SWS_Xcp_00711]	
[SWS_Xcp_00712]	
[SWS_Xcp_00713]	
[SWS_Xcp_00714]	
[SWS_Xcp_00715]	
[SWS_Xcp_00716]	
[SWS_Xcp_00718]	
[SWS_Xcp_00719]	
[SWS_Xcp_00720]	
[SWS_Xcp_00721]	
[SWS_Xcp_00722]	
[SWS_Xcp_00723]	
[SWS_Xcp_00724]	

▽

△

Number	Heading
[SWS_Xcp_00725]	
[SWS_Xcp_00726]	
[SWS_Xcp_00728]	
[SWS_Xcp_00729]	
[SWS_Xcp_00730]	
[SWS_Xcp_00731]	
[SWS_Xcp_00732]	
[SWS_Xcp_00733]	
[SWS_Xcp_00734]	
[SWS_Xcp_00735]	
[SWS_Xcp_00736]	
[SWS_Xcp_00737]	
[SWS_Xcp_00738]	
[SWS_Xcp_00739]	
[SWS_Xcp_00740]	
[SWS_Xcp_00741]	
[SWS_Xcp_00742]	
[SWS_Xcp_00761]	
[SWS_Xcp_00763]	
[SWS_Xcp_00766]	
[SWS_Xcp_00768]	
[SWS_Xcp_00801]	Definition of imported datatypes of module Xcp
[SWS_Xcp_00802]	
[SWS_Xcp_00803]	Definition of API function Xcp_Init
[SWS_Xcp_00807]	Definition of API function Xcp_GetVersionInfo
[SWS_Xcp_00813]	Definition of callback function Xcp_<Lo>RxIndication
[SWS_Xcp_00814]	Definition of callback function Xcp_<Lo>TxConfirmation
[SWS_Xcp_00823]	Definition of scheduled function Xcp_MainFunction
[SWS_Xcp_00824]	
[SWS_Xcp_00825]	
[SWS_Xcp_00832]	Definition of optional interfaces in module Xcp
[SWS_Xcp_00835]	Definition of callback function Xcp_<Lo>TriggerTransmit
[SWS_Xcp_00836]	
[SWS_Xcp_00840]	
[SWS_Xcp_00841]	
[SWS_Xcp_00842]	
[SWS_Xcp_00843]	
[SWS_Xcp_00844]	Definition of callback function Xcp_SetTransmissionMode

▽

△

Number	Heading
[SWS_Xcp_00845]	Definition of datatype Xcp_ConfigType
[SWS_Xcp_00846]	Definition of datatype Xcp_TransmissionModeType
[SWS_Xcp_00847]	
[SWS_Xcp_00848]	
[SWS_Xcp_00849]	
[SWS_Xcp_00850]	
[SWS_Xcp_00852]	
[SWS_Xcp_00853]	
[SWS_Xcp_00854]	
[SWS_Xcp_00855]	
[SWS_Xcp_00856]	
[SWS_Xcp_00857]	Definiton of development errors in module Xcp
[SWS_Xcp_00859]	
[SWS_Xcp_91001]	Definition of mandatory interfaces in module Xcp
[SWS_Xcp_NA_-00999]	

Table B.1: Added Specification Items in R22-11

B.1.2 Changed Specification Items in R22-11

none

B.1.3 Deleted Specification Items in R22-11

none

B.2 Traceable item history of this document according to AUTOSAR Release R23-11

B.2.1 Added Specification Items in R23-11

Number	Heading
[SWS_Xcp_00102]	
[SWS_Xcp_00103]	
[SWS_Xcp_00104]	
[SWS_Xcp_00105]	
[SWS_Xcp_00501]	
[SWS_Xcp_00701]	
[SWS_Xcp_00702]	
[SWS_Xcp_00703]	
[SWS_Xcp_00705]	
[SWS_Xcp_00706]	
[SWS_Xcp_00707]	
[SWS_Xcp_00708]	
[SWS_Xcp_00709]	
[SWS_Xcp_00710]	
[SWS_Xcp_00711]	
[SWS_Xcp_00712]	
[SWS_Xcp_00713]	
[SWS_Xcp_00714]	
[SWS_Xcp_00715]	
[SWS_Xcp_00716]	
[SWS_Xcp_00718]	
[SWS_Xcp_00719]	
[SWS_Xcp_00720]	
[SWS_Xcp_00721]	
[SWS_Xcp_00722]	
[SWS_Xcp_00723]	
[SWS_Xcp_00724]	
[SWS_Xcp_00725]	
[SWS_Xcp_00726]	
[SWS_Xcp_00728]	
[SWS_Xcp_00729]	
[SWS_Xcp_00730]	
[SWS_Xcp_00731]	

▽

△

Number	Heading
[SWS_Xcp_00732]	
[SWS_Xcp_00733]	
[SWS_Xcp_00734]	
[SWS_Xcp_00735]	
[SWS_Xcp_00736]	
[SWS_Xcp_00737]	
[SWS_Xcp_00738]	
[SWS_Xcp_00739]	
[SWS_Xcp_00740]	
[SWS_Xcp_00741]	
[SWS_Xcp_00742]	
[SWS_Xcp_00761]	
[SWS_Xcp_00763]	
[SWS_Xcp_00766]	
[SWS_Xcp_00768]	
[SWS_Xcp_00801]	Definition of imported datatypes of module Xcp
[SWS_Xcp_00802]	
[SWS_Xcp_00803]	Definition of API function Xcp_Init
[SWS_Xcp_00807]	Definition of API function Xcp_GetVersionInfo
[SWS_Xcp_00813]	Definition of callback function Xcp_<Lo>RxIndication
[SWS_Xcp_00814]	Definition of callback function Xcp_<Lo>TxConfirmation
[SWS_Xcp_00823]	Definition of scheduled function Xcp_MainFunction
[SWS_Xcp_00824]	
[SWS_Xcp_00825]	
[SWS_Xcp_00832]	Definition of optional interfaces in module Xcp
[SWS_Xcp_00835]	Definition of callback function Xcp_<Lo>TriggerTransmit
[SWS_Xcp_00836]	
[SWS_Xcp_00840]	
[SWS_Xcp_00841]	
[SWS_Xcp_00842]	
[SWS_Xcp_00843]	
[SWS_Xcp_00844]	Definition of callback function Xcp_SetTransmissionMode
[SWS_Xcp_00845]	Definition of datatype Xcp_ConfigType
[SWS_Xcp_00846]	Definition of datatype Xcp_TransmissionModeType
[SWS_Xcp_00847]	
[SWS_Xcp_00848]	
[SWS_Xcp_00849]	
[SWS_Xcp_00850]	

▽

△

Number	Heading
[SWS_Xcp_00852]	
[SWS_Xcp_00853]	
[SWS_Xcp_00854]	
[SWS_Xcp_00855]	
[SWS_Xcp_00856]	
[SWS_Xcp_00857]	Definiton of development errors in module Xcp
[SWS_Xcp_00859]	
[SWS_Xcp_91001]	Definition of mandatory interfaces in module Xcp
[SWS_Xcp_NA_-00999]	

Table B.2: Added Specification Items in R23-11

B.2.2 Changed Specification Items in R23-11

none

B.2.3 Deleted Specification Items in R23-11

none

B.3 Traceable item history of this document according to AUTOSAR Release R24-11

B.3.1 Added Specification Items in R24-11

Number	Heading
[SWS_Xcp_00860]	Location of the XCP module in the AUTOSAR communication stack
[SWS_Xcp_00862]	Sending and receiving of calibration data
[SWS_Xcp_00863]	Waiting for call of `Xcp_TxConfirmation` to avoid overwritting previously transmitted data
[SWS_Xcp_00864]	Use APIs of PduR for exchanging data via CAN
[SWS_Xcp_00865]	Use APIs of PduR for exchanging data via FlexRay
[SWS_Xcp_00866]	Use APIs of PduR for exchanging data via Ethernet (i.e. TcpIp)
[SWS_Xcp_00867]	Development error handling in context of `Xcp_RxIndication`
[SWS_Xcp_00868]	Development error handling in context of `Xcp_TxConfirmation`

▽

△

Number	Heading
[SWS_Xcp_00869]	Caveats on call of `Xcp_TxConfirmation`
[SWS_Xcp_00870]	Development error handling in context of `Xcp_TriggerTransmit`
[SWS_Xcp_00871]	Caveats on call of `Xcp_TriggerTransmit`

Table B.3: Added Specification Items in R24-11

B.3.2 Changed Specification Items in R24-11

Number	Heading
[SWS_Xcp_00701]	
[SWS_Xcp_00712]	
[SWS_Xcp_00714]	
[SWS_Xcp_00720]	
[SWS_Xcp_00734]	
[SWS_Xcp_00813]	Definition of callback function Xcp_RxIndication
[SWS_Xcp_00814]	Definition of callback function Xcp_TxConfirmation
[SWS_Xcp_00832]	Definition of optional interfaces requested by module Xcp
[SWS_Xcp_00835]	Definition of callback function Xcp_TriggerTransmit
[SWS_Xcp_00840]	
[SWS_Xcp_00841]	
[SWS_Xcp_00842]	
[SWS_Xcp_00843]	
[SWS_Xcp_00847]	
[SWS_Xcp_00859]	

Table B.4: Changed Specification Items in R24-11

B.3.3 Deleted Specification Items in R24-11

none

B.3.4 Added Constraints in R24-11

Number	Heading
[SWS_Xcp_CONSTR_00861]	`XcpRxPdu` and `XcpTxPdu` constraint for keeping the local buffer

Table B.5: Added Constraints in R24-11

B.3.5 Changed Constraints in R24-11

none

B.3.6 Deleted Constraints in R24-11

none