Document Title	Specification of Vehicle-2-X Basic Transport
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	794

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	• No content changes
2023-11-23	R23-11	AUTOSAR Release Management	• Removal of Tx confirmation and transaction ID • Clarification of upper layer interface of <User>_RxIndication() • Renamed V2Btp_GetVersionInfo to V2xBtp_GetVersionInfo
2022-11-24	R22-11	AUTOSAR Release Management	• Editorial changes to reflect the introduction of V2xDM
2021-11-25	R21-11	AUTOSAR Release Management	• Editorial Changes
2020-11-30	R20-11	AUTOSAR Release Management	• Editorial Changes
2019-11-28	R19-11	AUTOSAR Release Management	• Added support for C2CCC BSP 1.3 • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Corrections on init and configuration • Update to ETSI EN 302 636-5-1 V2.1.1 • Editorial Changes

▽

$\triangle$

2017-12-08	4.3.1	AUTOSAR Release Management	● Editorial Changes
2016-11-30	4.3.0	AUTOSAR Release Management	● Initial Release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Contents

1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module Vehicle-2-X Basic Transport (called "V2xBtp module" in this document).

V2xBtp module together with Vehicle-2-X Facilities [1] (V2xFac), Vehicle-2-X GeoNetworking [2] (V2xGn), Vehicle-2-X Management [3] (V2xM), Vehicle-2-X Data Manager [4] (V2xDM) and AUTOSAR BSW module Ethernet Interface [5] (EthIf) forms the V2X stack within the AUTOSAR architecture.

The base for this document is the BTP specification [6]. It is assumed that the reader is familiar with this specification. This document will not redefine BTP functionality, but it will try to follow the same order as the BTP specification.

1.1 Architectural overview

V2xBtp module provides services to and is dependent on the upper V2xFac module and uses the services of and gets services from the lower V2xGn module to realize its functionality explained in sections 1.2, 7.1, 7.2, 7.3 of this document.

Positioning of the V2xBtp module within the AUTOSAR BSW and the Layered Software architecture is shown in the figure below.
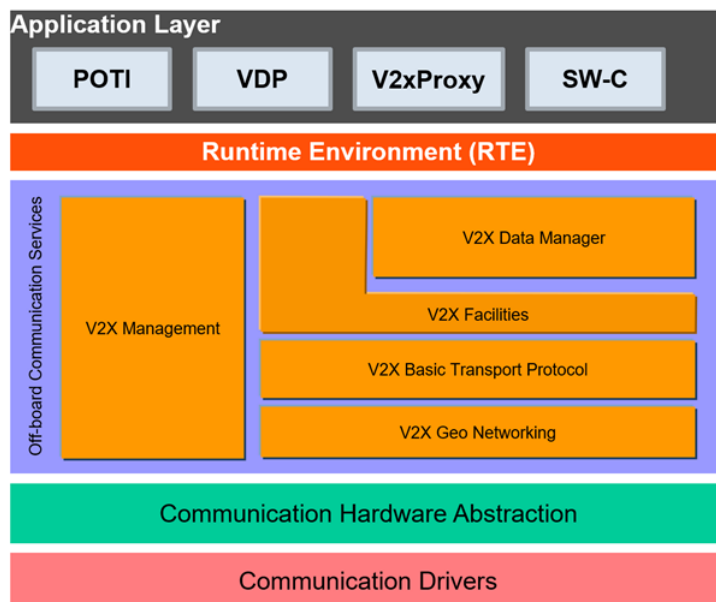

Figure 1.1: AUTOSAR BSW software architecture - V2xBtp module scope

1.2 Functional overview

V2xBtp module implements the BTP protocol specified in [6], which is a lightweight protocol that requires minimal processing.

V2xBtp module resides on top of the V2xGn module [2], which implements the GeoNetworking protocol specified in [7] and [8] and below the V2xFac module specified in [1]. It multiplexes/demultiplexes messages from different processes at the V2xFac module and provides a connectionless, unreliable (i.e. packets can arrive out-of-order, appear duplicated or can be lost), end-to-end packet transport service similar to UDP [9] in the ITS ad hoc network [10]. The design of BTP assumes that entities using the protocol are either tolerant against the unreliable packet transport or provide appropriate mechanisms for reliable communication themselves. It adopts the concept of ports from the IP suite and assigns well-known ports to specific ITS facilities layer protocols. The usage of ports is similar to the two-stage packet transport in the IP protocol suite but in the case of BTP, the GeoNetworking protocol transports the packets among the ITS stations and the BTP protocol delivers the packets to the entities at the ITS facilities layer.

Upper layers can access the transport services provided by the V2xBtp module via the API primitive `V2xBtp_Transmit()` by supplying the necessary data. In order to provide its packet transport services, V2xBtp module relies on the services of the V2xGn module (i.e. calls V2XGn_Transmit()). V2xBtp module also provides services for the V2xFac module by providing message receive notification and payload of the BTP packet that is received from the peer BTP entity by calling V2xFac_RxIndication() (as described in [SWS_V2xBtp_00015]). It also provides the callback `V2xBtp_RxIndication()` to the V2xGn module so that it can get message receive notification and payload of the GN packet that is received from the peer GeoNetworking entity.

According to the explanations above and the specification in [6], responsibilities of the V2xBtp module can be summarized as follows:

- Multiplexes messages from different processes at the V2xFac Module, e.g. CAM and DENM from the cooperative awareness basic service [11] and the decentralized environmental notification basic service [12] for the transmission of packets via the V2xGn module as well as de-multiplexing at the destination BTP protocol entity.

- Accepting message transmit requests along with protocol control information from the upper layer (V2xFac module) and transmitting the data using the services of the V2xGn module.

- Providing message receive notification and payload of the received packet to the upper layer (i.e. V2xFac module) when a BTP packet is received.

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the V2XBasicTransport module that are not included in the Autosar Glossary [13].

Abbreviation / Acronym:	Description:
BTP	Basic Transport Protocol
BTP-A	BTP header type denoting interactive packet transport
BTP-B	BTP header type denoting non-interactive packet transport
CAM	Cooperative Awareness Message
DENM	Decentralized Environmental Notification Message
DET	Default Error Tracer
GN	GeoNetworking
ITS	Intelligent Transport System
IP	Internet Protocol
PDU	Protocol Data Unit
UDP	User Datagram Protocol

Table 2.1: Acronyms and abbreviations used in the scope of this Document

3 Related documentation

3.1 Input documents & related standards and norms

[1] Specification of Vehicle-2-X Facilities
AUTOSAR_CP_SWS_V2XFacilities

[2] Specification of Vehicle-2-X Geo Networking
AUTOSAR_CP_SWS_V2XGeoNetworking

[3] Specification of Vehicle-2-X Management
AUTOSAR_CP_SWS_V2XManagement

[4] Specification of Vehicle-2-X Data Manager
AUTOSAR_CP_SWS_V2XDataManager

[5] Specification of Ethernet Interface
AUTOSAR_CP_SWS_EthernetInterface

[6] EN 302 636-5-1 V2.1.1: Vehicular Communication; Geonetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocols

[7] EN 302 636-4-1 V1.3.1: Vehicular Communication; Geonetworking; Part 4 Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality

[8] TS 102 636-4-2 V1.1.1: Intelligent Transport Systems (ITS); GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 2: Media-dependent functionalities for ITS-G5

[9] User Datagram Protocol
https://rfc-editor.org/rfc/rfc768.txt

[10] EN 302 665 V1.1.1: Intelligent Transport Systems (ITS); Communications Architecture

[11] EN 302 637-2 V1.4.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service

[12] EN 302 637-3 V1.3.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service

[13] Glossary
AUTOSAR_FO_TR_Glossary

[14] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[15] Specification of Default Error Tracer

AUTOSAR_CP_SWS_DefaultErrorTracer

[16] Specification of ECU State Manager
AUTOSAR_CP_SWS_ECUStateManager

[17] General Requirements on Basic Software Modules
AUTOSAR_CP_RS_BSWGeneral

[18] Requirements on Vehicle-2-X Communication
AUTOSAR_CP_RS_V2XCommunication

[19] TS 302 636-1 V1.2.1: Intelligent Transport Systems (ITS); GeoNetworking; Part 1: Requirements

[20] TS 302 636-2 V1.2.1: Intelligent Transport Systems (ITS); GeoNetworking; Part 2: Scenarios

3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [14], which is also valid for V2xBtp.

Thus, the specification SWS BSW General shall be considered as additional and required specification for V2xBtp.

4 Constraints and assumptions

4.1 Limitations

- The V2X modules follow the guidance regarding the Day-1 scenarios defined by the Basic System Standards Profile from Car-2-Car-Consortium.

- AUTOSAR R19-11 only focuses on the European version of car-to-car communication as defined by ETSI. Extension to other regions are planned for future releases of the AUTOSAR standard.

4.2 Applicability to car domains

This specification is applicable to all car domains.

5 Dependencies to other modules

This section describes the relations of the V2xBtp module to other modules within the AUTOSAR basic software architecture. It outlines the modules that are required or optional for the realization of the V2xBtp module and the V2xBtp services that these modules use.

5.1 AUTOSAR DET (Default Error Tracer)

In development mode, the V2xBtp module reports errors through the Det_ReportError function of the DET Module [15].

5.2 AUTOSAR EcuM (Ecu State Manager)

The EcuM [16] initializes the V2xBtp module by calling `V2xBtp_Init()` specified in 8.3.1.

5.3 AUTOSAR V2xFac (Vehicle-2-X Facilities)

The API used by the V2xBtp module consists of a notification service as basic agent for the transfer of BTP related data to the target upper layer namely the V2xFac Module. The callback service called by the V2xBtp module is declared and placed inside the V2xFac module. This callback provides receive indication (V2xFac_RxIndication, see [SWS_V2xBtp_00015]) service for the V2xFac Module.

5.4 AUTOSAR V2xGn (Vehicle-2-X GeoNetworking)

The V2xBtp module assumes a transmit request primitive (V2xGn_Transmit [2], see [SWS_V2xBtp_00017]) to be provided by the V2xGn module.

The V2xBtp module provides a notification service used by the V2xGn module. The callback called by the V2xGn module is declared and placed inside the V2xBtp module. This callback provides receive indication (V2xBtp_RxIndication specified in 8.4.1, see [SWS_V2xBtp_00016]) service for the V2xBtp module.

6 Requirements Tracing

The following tables reference the requirements specified in [17] and [18] 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_00345]	BSW Modules shall support pre-compile configuration	[SWS_V2xBtp_00036]
[SRS_V2X_00631]	The V2X system shall support an ETSI compliant Basic Transport Protocol	[SWS_V2xBtp_00011] [SWS_V2xBtp_20273] [SWS_V2xBtp_20274]

Table 6.1: Requirements Tracing

7 Functional specification

7.1 General Functionality

[SWS_V2xBtp_00011]

Upstream requirements: SRS_V2X_00631

⌈The V2xBtp module shall implement the Basic Transport Protocol (BTP) as specified in [6], unless specified otherwise in this document⌋

[SWS_V2xBtp_00012] ⌈The BTP protocol shall meet the BTP related requirements specified in [19] and support the scenarios specified in [20].⌋

[SWS_V2xBtp_00013] ⌈The V2xBtp Module shall encapsulate the payload from the V2xFac module with a BTP B header only as specified in [6] chapter 6 with the header format and structure as specified in chapter 7.⌋

7.2 Message Reception

[SWS_V2xBtp_00015] ⌈If a packet received by the V2xBTP with a port number that matches a port number contained in one of the containers configured in `V2xBtpDestinationPort`, the V2xBtp module shall call the function with the corresponding name configured in this container `V2xBtpRxIndicationFunctionName` to pass the payload and the GeoNetworking parameters to the V2xFac module.

If no configuration item for `V2xBtpDestinationPort` is found, the packet shall be dropped silently.⌋

Note: To call the V2xFacilities, one container per BTP port should be configured with the function name V2xFac_RxIndication added in the V2xBtpRxIndicationFunctionName.

[SWS_V2xBtp_00016] ⌈The V2xBtp module shall get the data (BTP packet) and the GeoNetworking parameters of a received GeoNetworking packet via the `V2xBtp_-RxIndication()` callback. This callback shall be triggered by the V2xGn module once a GN packet is received from the peer GN entity.⌋

7.3 Message Transmission

[SWS_V2xBtp_00017] ⌈The V2xBtp module shall provide the API `V2xBtp_Transmit()` to service transmit requests from the V2xFac module.⌋

[SWS_V2xBtp_00018] ⌈Source protocol operations when `V2xBtp_Transmit()` is triggered shall be as specified in [6] chapter 8.2.⌋

[SWS_V2xBtp_00019] ⌈The V2xBtp module transmits BTP packets to a peer BTP entity using the V2xGn_Transmit() API provided by the V2xGn module. The parameters for V2xGn_Transmit() shall be set according to [6] chapter 8.2. The parameters are provided to the V2xBtp module by the V2xFac module with a call to `V2xBtp_Transmit()`.⌋

[SWS_V2xBtp_20273]

Upstream requirements: SRS_V2X_00631

⌈The V2xBtp module shall employ BTP-B headers. Consequently, the GeoNetworking common header shall use a value of 2 for the NH field.⌋

[SWS_V2xBtp_20274]

Upstream requirements: SRS_V2X_00631

⌈The V2xBtp module shall set the destination port info field to the value 0.⌋

7.4 Error Classification

Section "Error Handling" of the document [14] "General Specification of Basic Software Modules" describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below.

7.4.1 Development Errors

[SWS_V2xBtp_00037] ⌈In case development error detection is enabled for the BTP module, the BTP module shall check API parameters for validity and report detected errors to the DET.⌋

[SWS_V2xBtp_00023] Definiton of development errors in module V2xBtp ⌈

Type of error	Related error code	Error value
API service called with invalid parameter	V2XBTP_E_PARAM	0x01
API service called with invalid pointer	V2XBTP_E_PARAM_POINTER	0x02
API service used without module initialization	V2XBTP_E_UNINIT	0x03
V2xBtp initialization failed	V2XBTP_E_INIT_FAILED	0x04

⌋

7.4.2 Runtime Errors

There are no runtime errors.

7.4.3 Production Errors

There are no production errors.

7.4.4 Extended Production Errors

There are no extended production errors.

8 API specification

8.1 Imported types

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

[SWS_V2xBtp_00024] Definition of imported datatypes of module V2xBtp ⌈

Module	Header File	Imported Type
Std	Std_Types.h	Std_ReturnType
	Std_Types.h	Std_VersionInfoType
V2x_GeneralTypes	V2x_GeneralTypes.h	V2x_GnAddressType
	V2x_GeneralTypes.h	V2x_GnAreaShapeType
	V2x_GeneralTypes.h	V2x_GnDestinationAreaType
	V2x_GeneralTypes.h	V2x_GnDestinationType
	V2x_GeneralTypes.h	V2x_GnLongPositionVectorType
	V2x_GeneralTypes.h	V2x_GnPacketTransportType
	V2x_GeneralTypes.h	V2x_GnTxResultType
	V2x_GeneralTypes.h	V2x_GnUpperProtocolType
	V2x_GeneralTypes.h	V2x_SecProfileType
	V2x_GeneralTypes.h	V2x_SecReportType
	V2x_GeneralTypes.h	V2x_TrafficClassIdType
V2xFac	V2xFac.h	V2xFac_RxParamsType
V2xGn	V2xGn.h	V2xGn_TxParamsType

⌋

8.2 Type definitions

8.2.1 V2xBtp_TxParamsType

[SWS_V2xBtp_00027] Definition of datatype V2xBtp_TxParamsType ⌈

Name	V2xBtp_TxParamsType	
Kind	Structure	
Elements	upperProtocol	
	Type	V2x_GnUpperProtocolType
	Comment	Specifies whether the BTP is interactive (BTP-A) or non-interactive (BTP-B).
	destinationPort	
	Type	uint16

▽

△

	Comment	Identifies the protocol entity at the ITS facilities layer at the destination of a BTP packet.
	transportType	
	Type	V2x_GnPacketTransportType
	Comment	Packet transport type (GeoUnicast, SingleHopBroadcast, Topology ScopedBroadcast, GeoBroadcast, GeoAnycast).
	destinationAddress	
	Type	V2x_GnAddressType
	Comment	Destination address for GeoUnicast.
	destinationArea	
	Type	V2x_GnDestinationAreaType
	Comment	Destination area for GeoBroadcast/GeoAnycast
	destinationType	
	Type	V2x_GnDestinationType
	Comment	Select which destination type (destinationAddress or destinationArea is used for this packet).
	secProfile	
	Type	V2x_SecProfileType
	Comment	Security Profile Information (Determines the security service to invoke).
	maxPacketLifetime	
	Type	uint16
	Comment	Maximum tolerable time in [s] a GeoNetworking packet can be buffered until it reaches its destination.
	trafficClassId	
	Type	V2x_TrafficClassIdType
	Comment	Traffic class id for the message.
Description	Structure containing BTP and GeoNetworking related parameters that should be passed to the transmit request function (i.e. V2xBtp_Transmit defined in 8.3.3) by the V2xFac module.	
Available via	V2xBtp.h	

⌋

8.2.2 V2xBtp_RxParamsType

[SWS_V2xBtp_00028] Definition of datatype V2xBtp_RxParamsType ⌈

Name	V2xBtp_RxParamsType	
Kind	Structure	
Elements	upperProtocol	
	Type	V2x_GnUpperProtocolType
	Comment	The protocol entity that processes the service primitive (BTP-A, BTP-B).
	packetTransportType	

▽

△

	Type	V2x_GnPacketTransportType
	Comment	Packet transport type of the received packet.
destinationAddress		
	Type	V2x_GnAddressType
	Comment	Destination address for GeoUnicast packet
destinationArea		
	Type	V2x_GnDestinationAreaType
	Comment	Destination area for GeoBroadcast/GeoAnycast packet.
destinationType		
	Type	V2x_GnDestinationType
	Comment	Select which destination type (destinationAddress or destinationArea is used for this packet).
sourcePositionVector		
	Type	V2x_GnLongPositionVectorType
	Comment	Geographical position for the source of the received GeoNetworking packet.
securityReport		
	Type	V2x_SecReportType
	Comment	Result information from the security operations for decryption and verification. This parameter is supplied by the V2xM module and forwarded up to the ITS Facilities layer passing through the Geo Networking and BTP layers.
certificateId		
	Type	uint64
	Comment	Identification of source certificate, for example the certificate hash. This parameter is supplied by the V2xM and forwarded up to the ITS Facilities layer passing through the GeoNetworking and BTP layers.
sspBits		
	Type	Array of uint8
	Size	4
	Comment	Sender permissions.
sspLength		
	Type	uint8
	Comment	Sender permissions length
trafficClass		
	Type	V2x_TrafficClassIdType
	Comment	Traffic class, with which the GeoNetworking packet was generated by the source.
remPacketLifetime		
	Type	uint16
	Comment	Remaining lifetime of the packet in [s].
remHopLimit		
	Type	uint8
	Comment	Remaining hop limit of the packet.
itsAid		
	Type	uint32
	Comment	The numerical value of the ITS-AID (Application Identifer)

▽

$\triangle$

Description	Structure containing BTP and GeoNetworking parameters related to a received BTP packet. This structure should be passed to the V2xBtp module from the V2xGn Module when the V2xGn module receives a GN packet from the peer GeoNetworking entity.
Available via	V2xBtp.h

$\rfloor$

8.3 Function definitions

8.3.1 V2xBtp_Init

[SWS_V2xBtp_00029] Definition of API function V2xBtp_Init $\lceil$

Service Name	V2xBtp_Init	

Syntax	`void V2xBtp_Init (`
	`    void* CfgPtr`
	`)`	

Service ID [hex]	0x01	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	CfgPtr	Pointer to configuration set
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Initializes the V2xBtp module.	
Available via	V2xBtp.h	

$\rfloor$

[SWS_V2xBtp_00059] $\lceil$ If development error detection is enabled: the function shall check the parameter CfgPtr for containing a valid configuration. If the check fails, the function shall raise the development error `V2XBTP_E_INIT_FAILED`. $\rfloor$

8.3.2 V2xBtp_GetVersionInfo

[SWS_V2xBtp_00030] Definition of API function V2xBtp_GetVersionInfo ⌈

Service Name	V2xBtp_GetVersionInfo	

Syntax	`void V2xBtp_GetVersionInfo (`
	`  Std_VersionInfoType* VersionInfoPtr`
	`)`	

Service ID [hex]	0x02	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (inout)	None	
Parameters (out)	VersionInfoPtr	Pointer to where to store the version information of this module.
Return value	None	
Description	Returns the version information of this module.	
Available via	V2xBtp.h	

⌋

[SWS_V2xBtp_00041] ⌈If development error detection is enabled: the function shall check the parameter VersionInfoPtr for being valid. If the check fails, the function shall raise the development error V2XBTP_E_PARAM_POINTER.⌋

8.3.3 V2xBtp_Transmit

[SWS_V2xBtp_00031] Definition of API function V2xBtp_Transmit ⌈

Service Name	V2xBtp_Transmit	

Syntax	`Std_ReturnType V2xBtp_Transmit (`
	`  const V2xBtp_TxParamsType* TransmitParams,`
	`  uint16 Length,`
	`  const uint8* DataPtr`
	`)`	

Service ID [hex]	0x03	
Sync/Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	TransmitParams	Structure containing all the BTP and GeoNetworking parameters used for the transmit request.
	Length	Length of the data pointed by DataPtr.
	DataPtr	Payload of the BTP packet to be transmitted
Parameters (inout)	None	
Parameters (out)	None	

▽

△

Return value	Std_ReturnType	E_OK: Transmit request has been accepted.
		E_NOT_OK: Transmit request has not been accepted.
Description	This API is called by the V2xFac module to request sending a BTP-PDU to the peer BTP entity.	
Available via	V2xBtp.h	

⌋

[SWS_V2xBtp_00042] ⌈If development error detection is enabled: the function shall check that the service `V2xBtp_Init()` was previously called. If the check fails, the function shall raise the development error `V2XBTP_E_UNINIT`.⌋

[SWS_V2xBtp_00043] ⌈If development error detection is enabled: the function shall check the parameter TransmitParams for being valid. If the check fails, the function shall raise the development error `V2XBTP_E_PARAM_POINTER`.⌋

[SWS_V2xBtp_00044] ⌈If development error detection is enabled: the function shall check the parameter DataPtr for being valid. If the check fails, the function shall raise the development error `V2XBTP_E_PARAM_POINTER`.⌋

8.4   Callback notifications

8.4.1   V2xBtp_RxIndication

[SWS_V2xBtp_00032] Definition of callback function V2xBtp_RxIndication ⌈

Service Name	V2xBtp_RxIndication	

Syntax	`void V2xBtp_RxIndication (`
	`  const V2xBtp_RxParamsType* ReceiveParams,`
	`  uint16 Length,`
	`  const uint8* DataPtr`
	`)`	

Service ID [hex]	0x04	
Sync/Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	ReceiveParams	Structure containing all the BTP and GeoNetworking parameters of a received BTP packet.
	Length	Length of the data pointed by dataPtr.
	DataPtr	Payload of the received GeoNetworking packet (BTP-PDU).
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	

▽

△

Description	Via this API, the V2xBtp module gets the data (BTP-PDU) and the GeoNetworking parameters of a received GeoNetworking packet.
Available via	V2xBtp.h

⌋

[SWS_V2xBtp_00046] ⌈If development error detection is enabled: the function shall check that the service `V2xBtp_Init()` was previously called. If the check fails, the function shall raise the development error `V2XBTP_E_UNINIT`.⌋

[SWS_V2xBtp_00047] ⌈If development error detection is enabled: the function shall check the parameter ReceiveParams for being valid. If the check fails, the function shall raise the development error `V2XBTP_E_PARAM_POINTER`.⌋

[SWS_V2xBtp_00048] ⌈If development error detection is enabled: the function shall check the parameter DataPtr for being valid. If the check fails, the function shall raise the development error `V2XBTP_E_PARAM_POINTER`.⌋

8.4.2 V2xBtp_CopyTxData

[SWS_V2xBtp_00052] Definition of API function V2xBtp_CopyTxData ⌈

Service Name	V2xBtp_CopyTxData	

Syntax	`Std_ReturnType V2xBtp_CopyTxData (`
	`  uint8* DestPtr,`
	`  uint16* Length`
	`)`	

Service ID [hex]	0x06	
Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	DestPtr	Pointer to the buffer where the BTP module shall copy the data to.
Parameters (inout)	Length	In direction: Maximum length available for data to be copied. Out direction: The actual length of the copied data.
Parameters (out)	None	
Return value	Std_ReturnType	`E_OK`: Data has been copied `E_NOT_OK`: Length does not match
Description	This API is called by the V2xGn module to request the V2xBtp module to copy the transmission data to a specific location.	
Available via	V2xBtp.h	

⌋

[SWS_V2xBtp_00053] ⌈If development error detection is enabled: the function shall check that the service `V2xBtp_Init()` was previously called. If the check fails, the function shall raise the development error `V2XBTP_E_UNINIT`.⌋

[SWS_V2xBtp_00054] ⌈If development error detection is enabled: the function shall check the parameter DestPtr for being valid. If the check fails, the function shall raise the development error `V2XBTP_E_PARAM_POINTER`.⌋

[SWS_V2xBtp_00057] ⌈If development error detection is enabled: the function shall check the parameter Length for being valid (Pointer is not NULL_PTR and value matches to data length). If the check fails, the function shall raise the development error `V2XBTP_E_PARAM_POINTER`.⌋

8.5 Expected interfaces

8.5.1 Mandatory Interfaces

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

[SWS_V2xBtp_00034] Definition of mandatory interfaces required by module V2x Btp ⌈

API Function	Header File	Description
V2xFac_RxIndication	V2xFac.h	This API primitive is called by the V2xBtp module providing the data and the GeoNetworking parameters of a received BTP packet to V2xFac module.
V2xGn_Transmit	V2xGn.h	Is called by V2x_Btp to send a message.

⌋

8.5.2 Optional interfaces

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

[SWS_V2xBtp_00035] Definition of optional interfaces requested by module V2x Btp ⌈

API Function	Header File	Description
Det_ReportError	Det.h	Service to report development errors.

⌋

8.5.3   Configurable Interfaces

8.5.3.1   <User>_RxIndication

[SWS_V2xBtp_00056] Definition of configurable interface <User>_RxIndication ⌈

Service Name	<User>_RxIndication	

Syntax	`void <User>_RxIndication (`
	`  const V2xBtp_RxParamsType* ReceiveParams,`
	`  uint16 Length,`
	`  const uint8* DataPtr`
	`)`	

Sync/Async	Synchronous	
Reentrancy	Non Reentrant	
Parameters (in)	ReceiveParams	Structure containing all the BTP and GeoNetworking parameters of a received BTP packet.
	Length	Length of the data pointed by dataPtr.
	DataPtr	Payload of the received GeoNetworking packet (BTP packet).
Parameters (inout)	None	
Parameters (out)	None	
Return value	None	
Description	Via this API, the BTP upper layer modules gets the data (BTP-SDU) and the GeoNetworking parameters of a received BTP packet.	
Available via	Configuration parameter V2xBtpCddHeaderFile	

⌋

[SWS_V2xBtp_00058] ⌈The callback function shall be configurable by the configuration container: `V2xBtpRxIndicationFunction`.⌋

9 Sequence diagrams

The following sequence diagrams show the interactions between the V2xBtp module and its lower and upper modules (V2xGn and V2xFac respectively).

Please note that the sequence diagrams are an extension for illustrational purposes to ease understanding of the specification and to support the functional specification described in chapter 7 and API specification described in chapter 8.

Note that all parameters and return types are left out to make the diagrams easier to read and understand.

9.1 Transmit Request

Figure 9.1 shows transmission of an ITS-FPDU from the V2xFac module to V2xGn module using `V2xBtp_Transmit()`. When V2xBtp gets a transmission request from the upper V2xFac module, it will validate the input parameters, construct a BTP packet by adding a BTP header to the payload it received from the V2xFac module (i.e. ITS-FPDU), then calls V2xGn_Transmit() to request sending a GeoNetworking packet setting the parameters for V2xGn_Transmit() as specified in [6] clause 8.2.
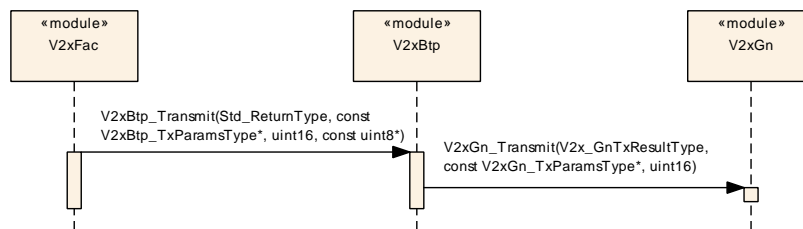

Figure 9.1: Transmission request to V2xBtp

9.2 Receive Indication

As shown in Figure 9.2, the reception of a BTP packet is indicated to the V2xBtp module by the V2xGn module by via a call to `V2xBtp_RxIndication()`. V2xBtp module then calls the corresponding receive indication of the V2xFac module to signal a reception to the upper layer. Packet payload, GeoNetworking related parameters pertaining to the received packet and a handle to be used for verification on demand is the information that is passed to the upper layers through the use of `V2xBtp_-RxIndication()` and V2xFac_RxIndication() primitives.
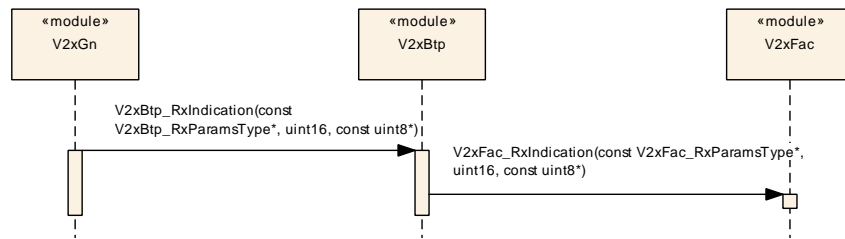
Figure 9.2: Receive Indication to V2xBtp

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 V2xBtp.

Chapter 10.3 specifies published information of the module V2xBtp.

10.1 How to read this chapter

For details refer to the chapter 10.1 "Introduction to configuration specification" in [14].

10.2 Containers and configuration parameters

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

10.2.1 Variants

Currently one configuration variant is defined for the V2xBtp module:

10.2.1.1 VARIANT-PRE-COMPILE

[SWS_V2xBtp_00036]
  Upstream requirements: SRS_BSW_00345

⌈The V2xBtp module only supports VARIANT-PRE-COMPILE⌋

10.2.2 V2xBtp

[ECUC_V2xBtp_00001] Definition of EcucModuleDef V2xBtp ⌈

Module Name	V2xBtp
Description	Configuration of the V2xBtp (Vehicle-2-X Basic Transport) module.
Post-Build Variant Support	false
Supported Config Variants	VARIANT-PRE-COMPILE

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
V2xBtpGeneral	1	This container contains the general configuration parameters of the Vehicle-2-X Basic Transport.
V2xBtpRxIndicationFunction	1..*	This container contains the RxIndication functions for V2xBtp upper layers.

Figure 10.1: ECU Configuration V2xBtp

10.2.3 V2xBtpGeneral

[ECUC_V2xBtp_00002] Definition of EcucParamConfContainerDef V2xBtpGeneral ⌈

Container Name	V2xBtpGeneral
Parent Container	V2xBtp
Description	This container contains the general configuration parameters of the Vehicle-2-X Basic Transport.
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**ECUC ID**
V2xBtpDevErrorDetect	1	[ECUC_V2xBtp_00005]
V2xBtpVersionInfoApi	1	[ECUC_V2xBtp_00004]

No Included Containers

⌟

[ECUC_V2xBtp_00005] Definition of EcucBooleanParamDef V2xBtpDevErrorDetect ⌈

Parameter Name	V2xBtpDevErrorDetect		
Parent Container	V2xBtpGeneral		
Description	Switches the Default Error Tracer (Det) detection and notification ON or OFF. • true: enabled (ON) • false: disabled (OFF)		
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_V2xBtp_00004] Definition of EcucBooleanParamDef V2xBtpVersionInfoApi ⌈

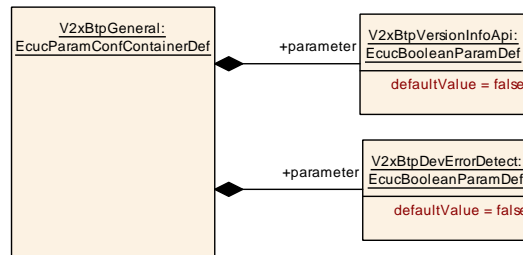| Parameter Name	V2xBtpVersionInfoApi		
Parent Container	V2xBtpGeneral		
Description	Enable/disables the API for reading the version information of the V2xBtp Module. • true: enabled (ON) • false: disabled (OFF)		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌟

Figure 10.2: ECU Configuration V2xBtpGeneral

10.2.4 V2xBtpRxIndicationFunction

[ECUC_V2xBtp_00009] Definition of EcucParamConfContainerDef V2xBtpRxIndicationFunction ⌈

Container Name	V2xBtpRxIndicationFunction		
Parent Container	V2xBtp		
Description	This container contains the RxIndication functions for V2xBtp upper layers.		
Post-Build Variant Multiplicity	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Configuration Parameters			

Included Parameters		
Parameter Name	Multiplicity	ECUC ID
V2xBtpCddHeaderFile	1	[ECUC_V2xBtp_00008]
V2xBtpDestinationPort	1	[ECUC_V2xBtp_00006]
V2xBtpRxIndicationFunctionName	1	[ECUC_V2xBtp_00007]

No Included Containers

⌋

[ECUC_V2xBtp_00008] Definition of EcucStringParamDef V2xBtpCddHeaderFile ⌈

Parameter Name	V2xBtpCddHeaderFile
Parent Container	V2xBtpRxIndicationFunction
Description	This parameter specifies the name of the header file containing the definition of the V2x BtpRxIndicationFunction module functions.
Multiplicity	1
Type	EcucStringParamDef
Default value	–

▽

△

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

⌋

[ECUC_V2xBtp_00006] Definition of EcucIntegerParamDef V2xBtpDestination Port ⌈

Parameter Name	V2xBtpDestinationPort		
Parent Container	V2xBtpRxIndicationFunction		
Description	Destination port for the <User>_RxIndication function.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	0		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

[ECUC_V2xBtp_00007] Definition of EcucFunctionNameDef V2xBtpRxIndication FunctionName ⌈

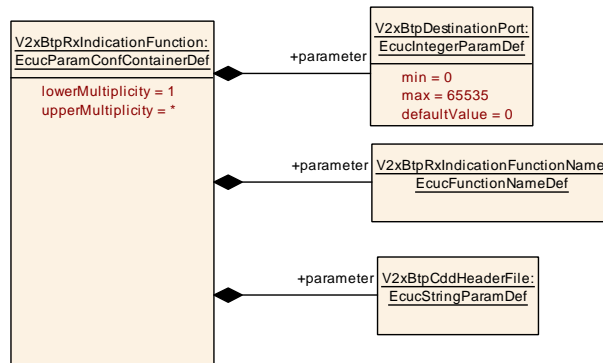| Parameter Name	V2xBtpRxIndicationFunctionName		
Parent Container	V2xBtpRxIndicationFunction		
Description	Name of the <User>_RxIndication callback function.		
Multiplicity	1		
Type	EcucFunctionNameDef		
Default value	–		
Regular Expression	–		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

⌋

Figure 10.3: ECU Configuration V2xBtpRxIndicationFunction

10.3 Published Information

For details refer to the chapter 10.3 "Published Information" in [14].

A   History of Specification Items

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

A.1   Specification Item History of this document compared to AUTOSAR R24-11.

A.1.1   Added Specification Items in R24-11

A.1.2   Changed Specification Items in R24-11

A.1.3   Deleted Specification Items in R24-11

A.2   Specification Item History of this document compared to AUTOSAR R23-11.

A.2.1   Added Specification Items in R23-11

Number	Heading
[SWS_V2xBtp_-00011]	
[SWS_V2xBtp_-00012]	
[SWS_V2xBtp_-00013]	
[SWS_V2xBtp_-00015]	
[SWS_V2xBtp_-00016]	

▽

△

Number	Heading
[SWS_V2xBtp_-00017]	
[SWS_V2xBtp_-00018]	
[SWS_V2xBtp_-00019]	
[SWS_V2xBtp_-00023]	Definiton of development errors in module V2xBtp
[SWS_V2xBtp_-00024]	Definition of imported datatypes of module V2xBtp
[SWS_V2xBtp_-00027]	Definition of datatype V2xBtp_TxParamsType
[SWS_V2xBtp_-00028]	Definition of datatype V2xBtp_RxParamsType
[SWS_V2xBtp_-00029]	Definition of API function V2xBtp_Init
[SWS_V2xBtp_-00030]	Definition of API function V2xBtp_GetVersionInfo
[SWS_V2xBtp_-00031]	Definition of API function V2xBtp_Transmit
[SWS_V2xBtp_-00032]	Definition of callback function V2xBtp_RxIndication
[SWS_V2xBtp_-00034]	Definition of mandatory interfaces in module V2xBtp
[SWS_V2xBtp_-00035]	Definition of optional interfaces in module V2xBtp
[SWS_V2xBtp_-00036]	
[SWS_V2xBtp_-00037]	
[SWS_V2xBtp_-00041]	
[SWS_V2xBtp_-00042]	
[SWS_V2xBtp_-00043]	
[SWS_V2xBtp_-00044]	
[SWS_V2xBtp_-00046]	
[SWS_V2xBtp_-00047]	
[SWS_V2xBtp_-00048]	

▽

△

Number	Heading
[SWS_V2xBtp_-00052]	Definition of API function V2xBtp_CopyTxData
[SWS_V2xBtp_-00053]	
[SWS_V2xBtp_-00054]	
[SWS_V2xBtp_-00056]	Definition of configurable interface <User>_RxIndication
[SWS_V2xBtp_-00057]	
[SWS_V2xBtp_-00058]	
[SWS_V2xBtp_-00059]	
[SWS_V2xBtp_-20273]	
[SWS_V2xBtp_-20274]	

Table A.1: Added Specification Items in R23-11

A.2.2 Changed Specification Items in R23-11

A.2.3 Deleted Specification Items in R23-11

A.3 Specification Item History of this document compared to AUTOSAR R22-11.

A.3.1 Added Specification Items in R22-11

Number	Heading
[SWS_V2xBtp_-00011]	
[SWS_V2xBtp_-00012]	
[SWS_V2xBtp_-00013]	
[SWS_V2xBtp_-00015]	
[SWS_V2xBtp_-00016]	
[SWS_V2xBtp_-00017]	
[SWS_V2xBtp_-00018]	
[SWS_V2xBtp_-00019]	
[SWS_V2xBtp_-00020]	
[SWS_V2xBtp_-00024]	
[SWS_V2xBtp_-00027]	
[SWS_V2xBtp_-00028]	
[SWS_V2xBtp_-00029]	
[SWS_V2xBtp_-00030]	
[SWS_V2xBtp_-00031]	
[SWS_V2xBtp_-00032]	
[SWS_V2xBtp_-00033]	
[SWS_V2xBtp_-00034]	

▽

△

Number	Heading
[SWS_V2xBtp_-00035]	
[SWS_V2xBtp_-00036]	
[SWS_V2xBtp_-00037]	
[SWS_V2xBtp_-00041]	
[SWS_V2xBtp_-00042]	
[SWS_V2xBtp_-00043]	
[SWS_V2xBtp_-00044]	
[SWS_V2xBtp_-00046]	
[SWS_V2xBtp_-00047]	
[SWS_V2xBtp_-00048]	
[SWS_V2xBtp_-00050]	
[SWS_V2xBtp_-00052]	
[SWS_V2xBtp_-00053]	
[SWS_V2xBtp_-00054]	
[SWS_V2xBtp_-00056]	
[SWS_V2xBtp_-00057]	
[SWS_V2xBtp_-00058]	
[SWS_V2xBtp_-00059]	
[SWS_V2xBtp_-20273]	
[SWS_V2xBtp_-20274]	

Table A.2: Added Specification Items in R22-11

A.3.2 Changed Specification Items in R22-11

A.3.3 Deleted Specification Items in R22-11

A.4 Constraint Item History of this document compared to AUTOSAR R22-11.

A.4.1 Added Constraints in R22-11

A.4.2 Changed Constraints in R22-11

A.4.3 Deleted Constraints in R22-11